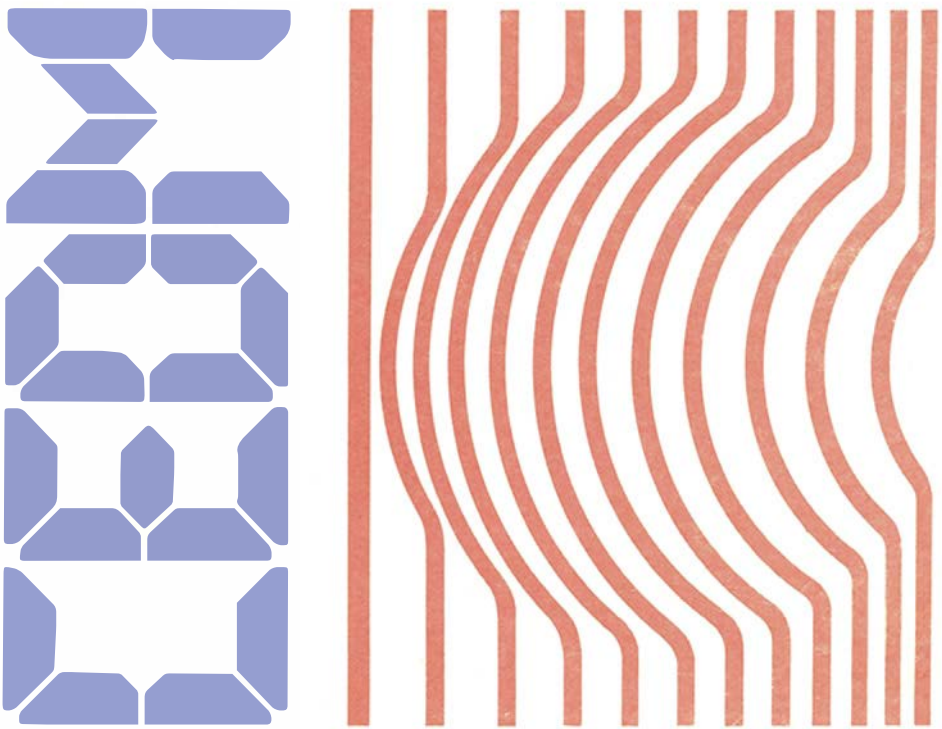
CURSO DE ELECTRÓNICA DIGITAL E MICROPROCESSADORES



CURSOS DE APERFEIÇOAMENTO

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANA

LIÇÃO MC-1

HISTÓRIA DOS COMPUTADORES

Desde os tempos mais remotos, o homem necessita dos números e das informações para o seu desenvolvimento pleno. O homo sapiens é, por natureza, um ser que processa dados, devido a sua capacidade de identificar, comparar, raciocinar e estabelecer grandezas para os objetos, necessitando, assim, de elementos de comunicação, a fim de organizar-se individual e socialmente. A curta e rápida história do desenvolvimento da computação vem refletir o contínuo esforço do homem em encontrar meios mais efetivos ao seu auxílio, na resolução de problemas e no armazenamento das informações essenciais para o equilíbrio das diversas áreas relacionadas a ele.

Por séculos o homem, sem qualquer recurso auxiliar, foi obrigado a utilizar seus dedos ou pequenos pedaços de pedras, rabiscos sobre rochas, ou mesmo, nós em uma corda, para representar os carneiros de seu rebanho, os animais de sua propriedade ou as medidas de cereais armazenados. Cada dedo representava uma unidade e, para efetuar-se uma adição, por exemplo, quatro mais três, estendiam-se quatro dedos de uma mão e três da outra, bastando contar os dedos estendidos. A partir daí, surgiu o nosso sistema decimal. Embora, para nós, uma soma como esta pareça trivial, esse método foi de extrema

utilidade, tanto que,através dele, desenvolveram-se operações mais complexas, tais como a multiplicação e a divisão, que eram ensinadas nas escolas romanas. Por exemplo, o aluno era obrigado a aprender a tabuada de multiplicação até 5 x 5 somente. Com os dedos das mãos, podia-se calcular o produto de quaisquer números entre cinco e dez. Suponhamos que desejássemos multiplicar 8 x 9. Para isso, estenderíamos três dedos de uma das mãos (8-5 = 3),e quatro da outra mão (9-5=4). O resultado seria obtido somando-se os dedos estendidos (3+4=7), que corresponde à dezena (70),e multiplicando-se os dedos restantes (2x1=2), correspondendo à unidade. Fazendo-se a soma dos resultados, obtemos o produto de 8 x 9, que é igual a 72. Prático, não é mesmo? Entretanto, esses métodos eram muito limitados face ao crescente aumento dos rebanhos e à introdução do comércio.

O ANTECESSOR DAS MÁQUINAS DE CALCULAR

O primeiro dispositivo de cálculos foi a "placa de contagem", originária da China e do Egito, por volta de 500 a.C. Estas eram placas com pedaços de madeira, metal ou rocha, semelhantes aos ábacos, com a diferença de que os contadores podiam ser acrescentados ou removidos durante os cálculos.

A origem do ábaco é desconhecida. Acredita-se que deva ter aparecido no Egito ou na Babilônia. Os antigos hindus empregavam um tipo de ábaco denominado "tabuleiro de areia" ou "tábua de poeira", o qual era também conhecido pelas civilizações romana e grega. Alguns estudiosos atribuem a origem do termo "abacus" ao semítico "Abai", que significa "poeira". Entretanto, outros são de opinião que o grego "abax", que significa "tabuazinha", é uma origem mais provável.

O ábaco é um dos dispositivos de cálculo manual mais remotos, introduzido a mais de 2000 anos, e é constituído por uma placa, contendo seixos, que representam as contas, colocados em ranhuras ou enfiados em cordéis, dispostos em fileiras. A palavra "cálculo" é derivada do latim "calculus", que significa "seixo" ou "pedaço de mármore". Um dos ábacos de maior repercussão nos tempos antigos foi o ábaco romano, apresentado na figura 1.

Atualmente, embora existam o Suan-pan da China, o Choreb da Armênia e o Coulba da Turquia, os ábacos mais utilizados são: o Schoty da Rússia e o Soroban do Japão. O ábaco russo tem uma estrutura retangular com fios. Cada um contendo dez

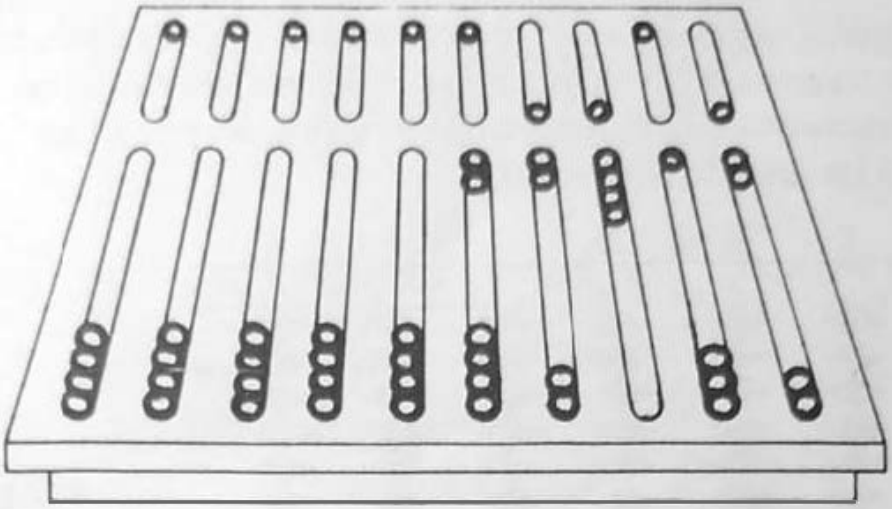


FIGURA 01

contas, das quais nove são requeridas para cálculos e armazenamento, enquanto uma é utilizada para o transporte de um dígito para o próximo fio. Cada fio representa o valor decimal das contas nele contidas, de tal forma que o fio A representa a unidade, o fio B a dezena, o fio C a centena e assim por diante. A figura 2 apresenta um exemplo de "armazenamento" do número 8.372.

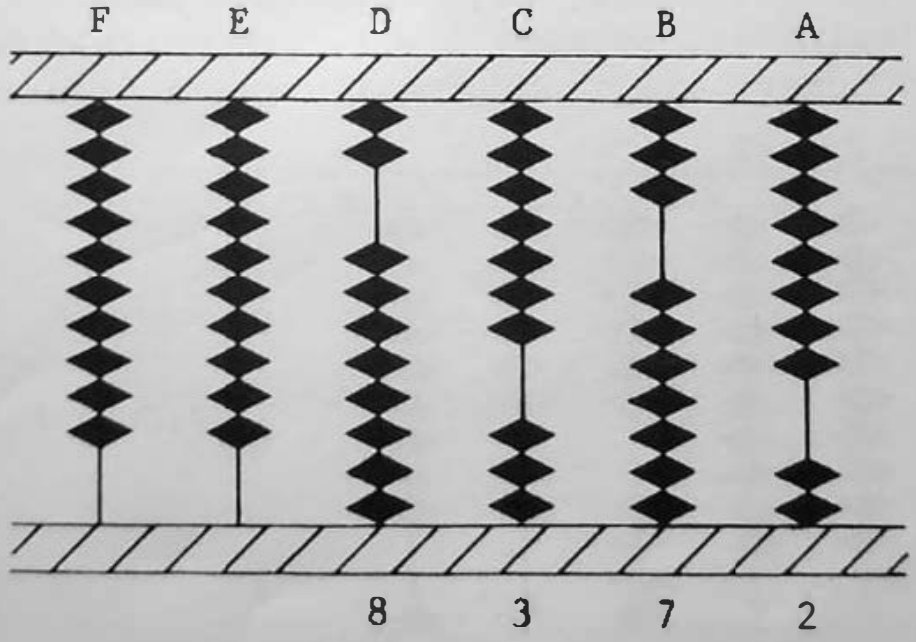


FIGURA 02

A operação aritmética mais comumente executada é a adição. Esta é realizada, somando-se sucessivamente os valores representados pelas contas nas diversas fileiras. Para somarmos 103 com 308, movemos três contas da fileira A para baixo; como temos somente sete contas restantes nessa fileira, a soma 3 + 8 é realizada movendo-se todas as contas para cima no-

vamente, e continuando a contagem do ponto interrompido (7). Assim, movemos apenas uma para baixo, correspondendo à oitava conta. O movimento de todas as contas para cima, indica que deve ser transportada uma conta para a próxima fileira, no caso B. Assim sendo, temos:

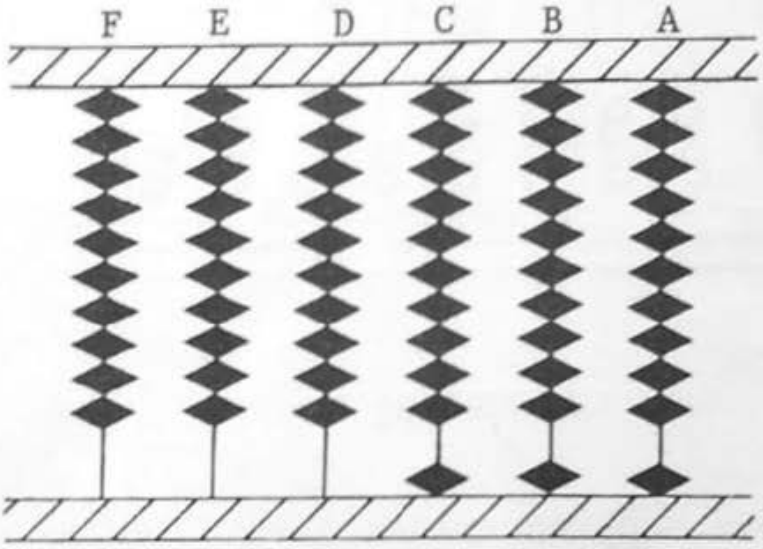


FIGURA 03

1 1 1 = 103 + 8

3 + 8 = 1 + transporte

0 + 1 (transporte) = 1

O próximo passo consiste em somarmos a fileira da dezena (B): zero mais o transporte (1), cujo resultado pode ser visto na figura 3. Finalmente, somamos 3 e 1, que corresponde à centena, obtendo o número 411, conforme ilustra a figura 4.

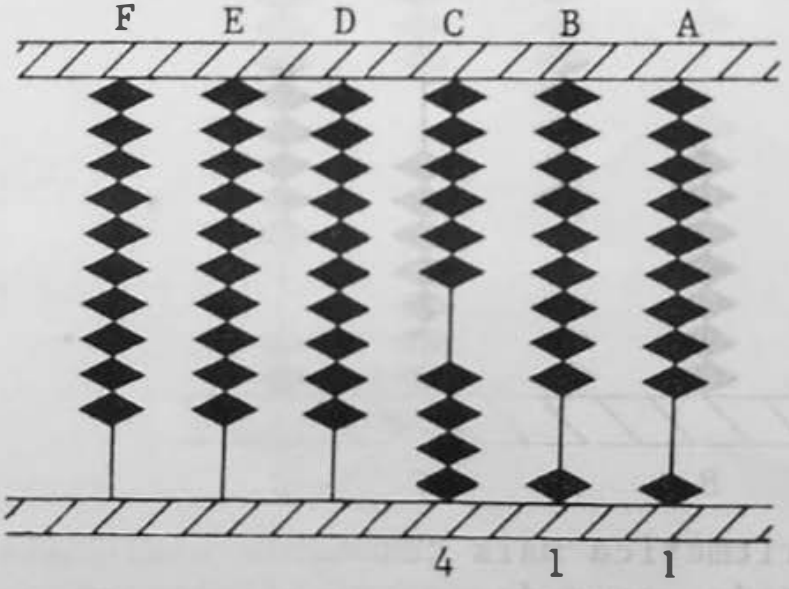


FIGURA 04

A multiplicação e a divisão eram realizadas, respectivamente, através de somas e subtrações sucessivas.

Uma outra versão muito difundida é o Soroban japonês, o qual, na figura 5, exibe o número 5.473.

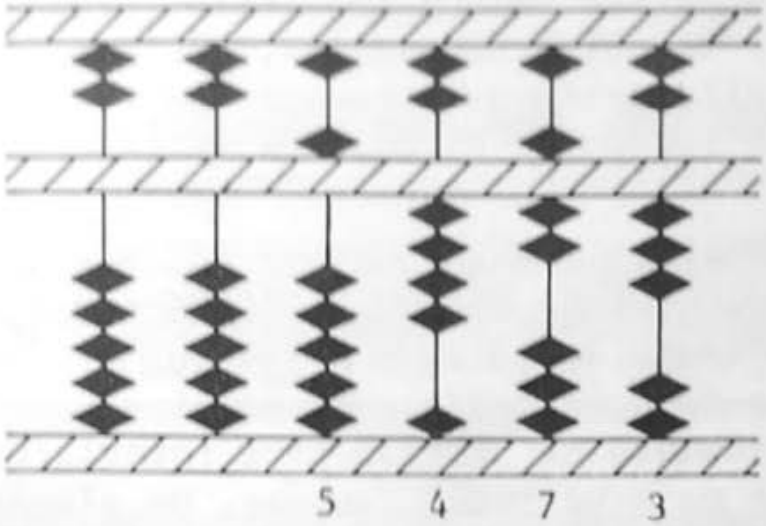


FIGURA 05

Esse ábaco japonês é dividido em duas seções por uma barra. A seção superior contém duas contas por fio e a inferior cinco por fio. Cada conta superior equivale a cinco unidades e estas, por sua vez, são representadas pelas contas inferiores.

A evolução do ábaco se deve, principalmente, ao progresso do comércio, cujas necessidades aritméticas foram amplamente satisfeitas com a sua utilização por um longo tempo. Entretanto, um grande auxílio na resolução de problemas numéricos foi a adoção dos algarismos arábicos (imagine calcular uma divisão entre MDCCXLIV e CXVIII).

Uma sensível simplificação dos cálculos se deve à invenção do algarismo zero, introduzido pelos matemáticos hindus, por volta de 400 a.C. Os numerais arábicos, atualmente usados, foram passados dos árabes para os hindus, cerca de 800 anos a.C. e destes para toda a Europa durante os anos de 1.400. Estes numerais permitem que qualquer número seja expresso por uma combinação de dez dígitos. Combinados com o ponto decimal, os algarismos arábicos tornam possível uma escrita aritmética bastante simples. É muito provável que, não fosse a adoção desses números, o desenvolvimento da tecnologia sofreria um grande atraso.

Um outro motivo, pelo qual o ábaco teve grande aceitação, foi a escassez de material de escrita. O papel, embora fosse inventado pelos chineses antes da era cristã, foi realmente difundido a partir do século XIV. Conta-se que a fabricação do papel teve sua origem com os Mouros da Espanha, no começo do século XII. A manufatura do papel em larga escala, na Itália ocorreu em 1.276. Mais tarde, as fábricas começa-

ram a se espalhar em países como a Inglaterra, França e Alemanha. O papel feito de polpa de madeira, contudo, foi uma invenção do século XIX, sendo que, somente em tempos relativamente recentes é que tornou-se disponível na base de produção em massa.

Esse advento possibilitou que muitos métodos de cálculos auxiliares se desenvolvessem. O crescimento das descobertas científicas, no tempo de Galileu e Newton, produziu muitos problemas, cujas soluções exigiram cálculos numéricos extensos, criando um aumento de demanda para formulações matemáticas, destinadas às mais variadas funções. Os alemães, no século XV, introduziram as tabelas de funções trigonométricas. É um importante evento na história da matemática foi arquitetado, em 1.617, pelo escocês John Napier: a tabela logarítmica. Esta ajudou muito na redução de cálculos monótonos de grandes números.

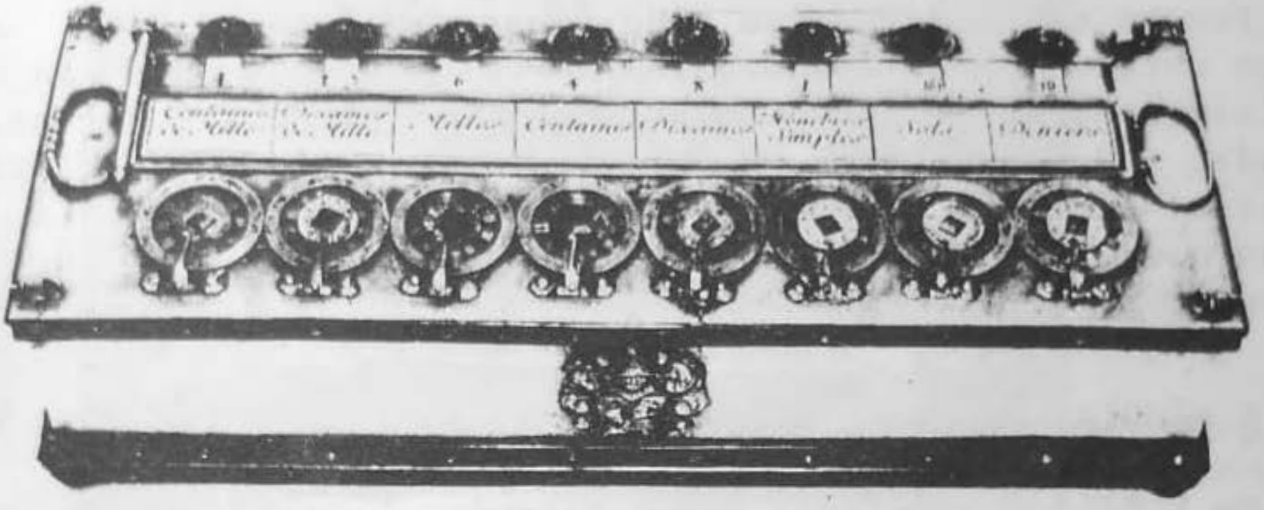
EVOLUÇÃO DAS MÁQUINAS DE CALCULAR.

A PRIMEIRA CALCULADORA-PASCAL.

Embora muitas tabelas e fórmulas já estivessem sendo desenvolvidas para o auxílio do homem na resolução dos problemas cada vez mais complexos e extensos, estas não eram suficientemente hábeis para contorná-los. Iniciou-se, então, através do matemático e filósofo francês, Blaise Pascal, uma jornada em busca da automatização das informações.

Em 1.642, com apenas 19 anos, Pascal inventou a primeira máquina capaz de realizar as operações de soma e subtração com números de 8 algarismos, quando estas operações estavam apenas a nível de adultos razoavelmente educados. Pascal gastou 10 anos aperfeiçoando a sua máquina, sem, no entanto, encontrar compradores, pois podia-se conseguir um pequeno grupo de empregados com placas de contagem pelo preço de uma destas elegantes calculadoras. Essa máquina era constituída de engrenagens numeradas de 0 a 9, ligadas por uma alavanca que acionava a engrenagem seguinte, sempre que uma passava para o dígito 0 (zero), (figura 6).

FIGURA 06



A MÁQUINA DE CALCULAR DE LEIBNITZ

Por volta dos anos de 1.670, Gottfried Wilhelm Von Leibnitz, um filósofo e matemático alemão de 25 anos, foi o primeiro a projetar uma máquina de calcular, na qual as multiplicações podiam ser realizadas rapidamente, através de somas sucessivas e as divisões, através de subtrações sucessivas, incluindo, assim, as quatro operações fundamentais (figura 7). Essa máquina só foi fabricada em 1.694, não chegando a se propagar, devido a sua imprecisão nos cálculos.

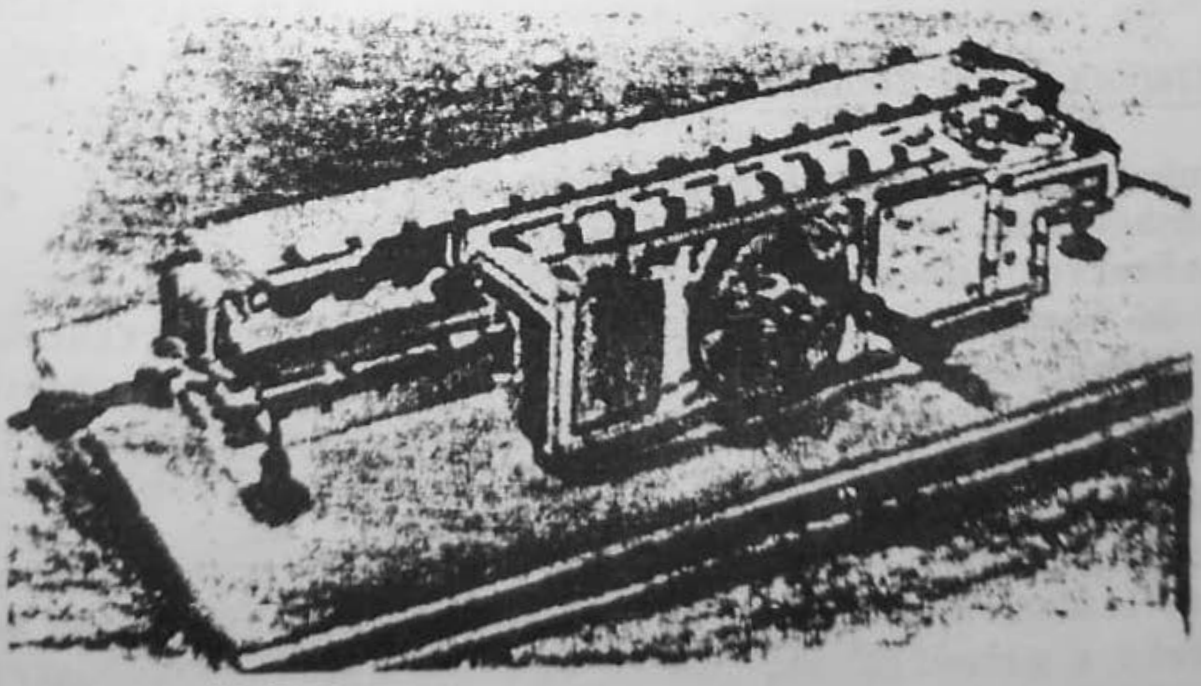


FIGURA 07

A MÁQUINA DE CALCULAR DE THOMAS

Poucas máquinas haviam sido construídas até 1.820, ano em que surgiu a primeira máquina de calcular bem-sucedida comercialmente, projetada pelo francês Charles Xavier Thomas. Essa podia somar, subtrair, multiplicar e dividir com relativa precisão. A exemplo das máquinas de Pascal e Leibnitz, a de Thomas operava manualmente com o sistema decimal, através de engrenagens.

A MÁQUINA DE CALCULAR DE BALDWIN

Em 1.875, o americano Frank Baldwin, baseando-se na máquina de Thomas, introduziu um princípio diferente na construção de sua calculadora. Essa passou a usar engrenagens, cujos dentes eram pinos móveis, que se retraíam para dentro do corpo da engrenagem. Cada engrenagem possuía uma alavanca, que ficava exposta externamente, sendo que a posição da alavanca móvel controlava o número de pinos salientes. Quando a manivela era acionada, a máquina contava os pinos salientes em cada engrenagem e mostrava o resultado, semelhante à técnica de contagem dos dedos. Esse sistema foi tão bem sucedido que foi adotado na maioria das dez milhões de calculadoras mecânicas, vendidas durante o século seguinte. (figura 8).

INTRODUÇÃO DO TECLADO

Nenhuma das máquinas anteriores possuía teclados como nós a conhecemos. A primeira máquina operada por tecla foi o computômetro, inventado por Dorr Eugene Felt, em 1.887. O teclado do computômetro representou um impulso importante no progresso das máquinas de calcular, pois simplificou bastante os trabalhosos sistemas de alavanca. Muitas máquinas semelhantes à de Felt se seguiram posteriormente. Versões melhoradas são ainda hoje largamente utilizadas. Para minimizar o trabalho penoso da contabilidade da época, W.S.Burroughs construiu, em 1.890, uma máquina de adição e impressão, que mais tarde seria a primeira máquina de somar e imprimir industrializada pela Burroughs Corporation. Essa máquina (figura 9) acionada por uma manivela, era projetada para registrar, totalizar e calcular.

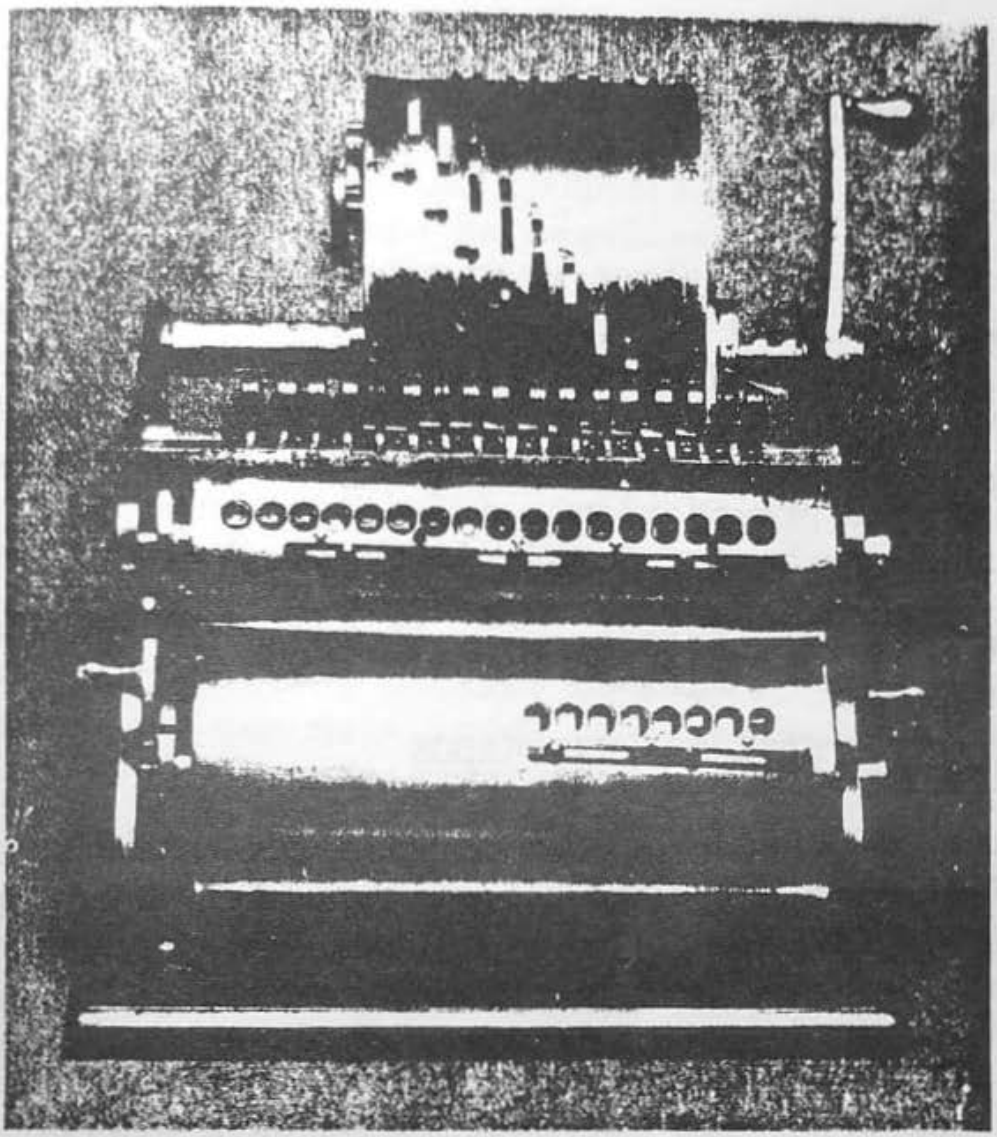


FIGURA 08

Contudo, a primeira máquina de teclado bem sucedida comercialmente,apareceu por volta de 1.911,quando Jay R. Monroe e Baldwin lançaram a calculadora Monroe. A partir de 1.920, as caixas registradoras, máquinas de calcular, e várias

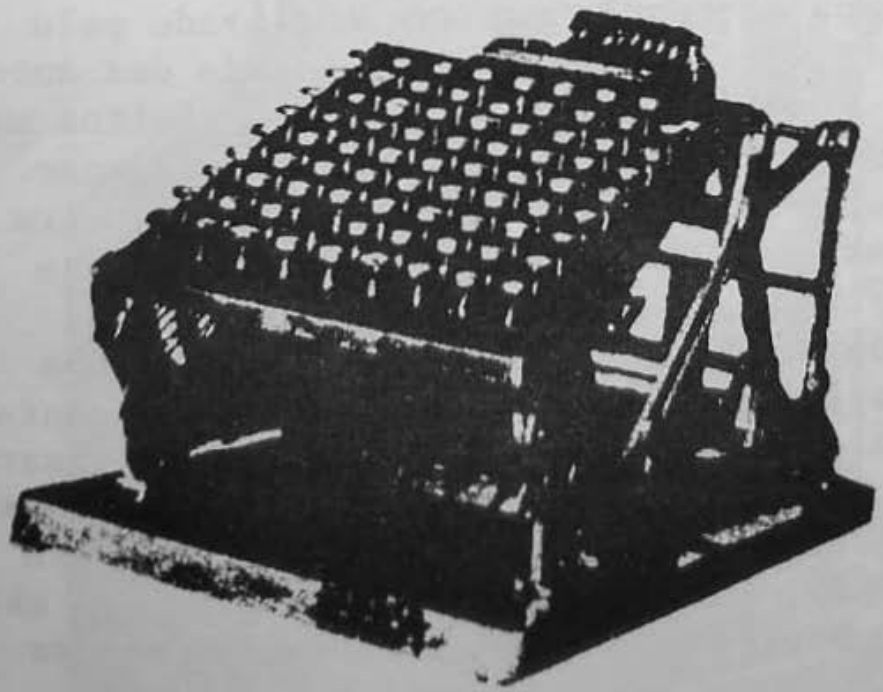


FIGURA 09

máquinas de faturamento e de contas, do tipo de teclado, tinham transformado a aritmética comercial em operação completamente mecanizada.

Porém, esses dispositivos eram máquinas não automáticas, isto é, faziam uma simples operação aritmética de cada vez. Após a Primeira Guerra Mundial, as máquinas de contabilidade sofreram grandes avanços sem, no entanto, suprirem a velocidade e a simplificação que a ciência e a tecnologia impunham. Dois acontecimentos provocariam o início da automatização: o aparecimento dos cartões perfurados e os projetos das máquinas automáticas de Babbage.

INTRODUÇÃO DOS CARTÕES PERFURADOS

Os sistemas de cartão perfurado apareceram comercialmente a partir de 1.930. Entretanto a invenção do cartão perfurado se deve a um tecelão francês, chamado Joseph Marie Jacquard, no final da revolução americana, em 1.801. Jacquard,ao sentir a dificuldade de tecer desenhos complexos em fazendas, construiu a primeira máquina capaz de seguir uma série de instruções perfuradas em cartões. Porém esse grande salto no desenvolvimento das máquinas atemorizou as pessoas leigas e desorientadas, sem conseguir a aceitação pública. Ainda que os cartões perfurados continuassem a ser usados no controle de processo, essa aceitação só foi conseguida a partir de 1.890, por intermédio de Herman Hollerith.

Este fato se deve ao censo, que era realizado pelo Bureau de Recenseamento dos Estados Unidos, a cada dez anos. No início, o cálculo e a compilação dos dados eram feitos manualmente, possibilitado pela relativamente baixa população. Contudo, a população norte-americana cresceu de tal maneira que as informações de natureza mais complexa, requeridas em cada censo, tornaram o método manual impraticável. Para se ter uma idéia, o censo de 1.880 levou sete anos e meio para ser concluído. Quando as informações desejadas estavam prontas para ser publicada, já eram obsoletas e inúteis. Para sanar essa deficiência, o Bureau de Recenseamento procurou o auxílio de um notável estatístico, como agente especial do censo de 1.890, Herman Hollerith. Na época, Hollerith já estava estudando uma maneira de processar os dados mais rapidamente e

com maior eficiência que o sistema manual. Em 1.887, Hollerith desenvolveu o seu conceito de máquina de cartão legível, projetou um aparelho conhecido como a "máquina de censo" e concluiu um grupo de máquinas aptas a processar o censo de 1.890. Foi a primeira instalação em larga escala, de máquinas de processamento de dados por cartões perfurados. O sistema de Hollerith desenvolveu-se com sucesso. O tempo de tabulação diminuiu em oito vezes. Mesmo com um aumento de população de 50 para 63 milhões de pessoas, o censo de 1.890 foi completado em menos de três anos. (Um selecionador de cartões é apresentado pelo Dr. Hollerith) (figura 10).

Devido ao êxito de suas máquinas, Hollerith fundou, em 1.896, "a Tabulating Machine Company" - Companhia de Máquina de Tabulação, que mais tarde, unindo-se a outras, formaram a "Internacional Business Machines Corporation" (IBM).

Contudo, as pesquisas para se desenvolverem máquinas mais eficazes tiveram seqüência através de James Powers, o sucessor de Hollerith no cargo de estatístico do Bureau de

FIGURA 10

Recenseamento norte-americano. Powers aperfeiçoou várias tabuladoras e outras máquinas de cartão perfurado, utilizadas com êxito no censo de 1.910.

No ano seguinte, ele fundou a "Powers Accounting Machines Company" para explorar as vantagens de suas máquinas classificadora e perfuradora. Em 1.927, essa empresa se fundiria com outras, resultando na Remington Rand Corporation.

AS MÁQUINAS AUTOMÁTICAS DE BABBAGE

Charles P. Babbage, nascido em 1.792, foi filho de um banqueiro inglês, e sempre foi aficcionado pela ciência tecnológica. Quando garoto, aprendeu matemática sozinho e, na Universidade de Cambridge, junto com alguns amigos, fundou uma organização chamada Sociedade Analítica. Foi durante sua época em Cambridge que Babbage,com apenas 20 anos, teve as primeiras idéias para a construção de uma máquina de calcular, a fim de sanar os erros, que normalmente e em grande quantidade, existiam nas tabelas logarítmicas. Entretanto, somente em 1.822, a sua máquina começaria a ser construída. Ela seria capaz de operar 17 diferenças, armazenando números de até 20 casas decimais. Isso iria permitir que um polinômio de até 17 graus fosse resolvido com aproximações, relativamente satisfatórias. Tal projeto foi financiado pelo governo britânico, durante dez anos, com cerca de 17.000 libras. Porém,com a complexidade e a excessiva demora na sua conclusão, tanto Babbage quanto o governo britânico foram perdendo o interesse e, depois de dez anos de paralização, o projeto foi definitivamente abandonado em 1.842. Posteriormente, Georg e Eduard Scheutz,de Stockolmo,assimilaram as idéias de Babbage e construíram uma máquina de diferença que trabalhava com 14 diferenças e armazenava números de até 14 casas decimais.

Uma das razões, pela qual Babbage havia perdido o interesse pela máquina de diferença, foi a idéia de se desenvolver uma máquina mais ambiciosa, aplicável a qualquer tipo de cálculo. Em 1.839, ele renunciaria ao posto de professor de Lucassian em Cambridge, posto este que fora formalmente mantido por Newton até 1.828, e passaria a se dedicar exclusivamente a essa máquina, que ele denominou de Máquina Analítica. Essa possuiria todos os ingredientes de um computador atual: um

supervisor, uma área de memória, um processador, dispositivos de comunicação e um selecionador de alternativas. O programa de controle e o dado a ser processado pela máquina sob a direção de um programa, seriam introduzidos na máquina, através dos cartões perfurados de Jacquard. Os números seriam armazenados na forma decimal e transmitidos para o processador por meio de barras e alavancas. Tal máquina teria a capacidade para armazenar 1.000 números decimais de 50 dígitos. Entretanto, devido aos poucos recursos tecnológicos da época e à descredibilidade da população, Babbage faleceu em 1.871, sem ter conseguido o seu grande sonho. A despeito disso, ele é considerado um dos grandes pioneiros no campo da computação. Mais tarde, seu filho iria prosseguir o seu trabalho, também, sem conseguir terminá-lo. Essa obra incompleta permanece atualmente no museu de ciência de Kensington do Sul em Londres, simbolizando a coragem e a imaginação de um homem cujas idéias estavam além do seu tempo. (figura 11)

COMPUTADORES ANALÓGICOS

Acredita-se que os primeiros cálculos analógicos foram realizados através de gráficos para resolver problemas de agrimensura, pelos povos mais antigos. Muitos dispositivos analógicos foram utilizados na história antiga da Europa Ocidental. Contudo o primeiro dispositivo universalmente conhecido foi a régua de cálculo, inventada no início do século XVII. (figura 12).

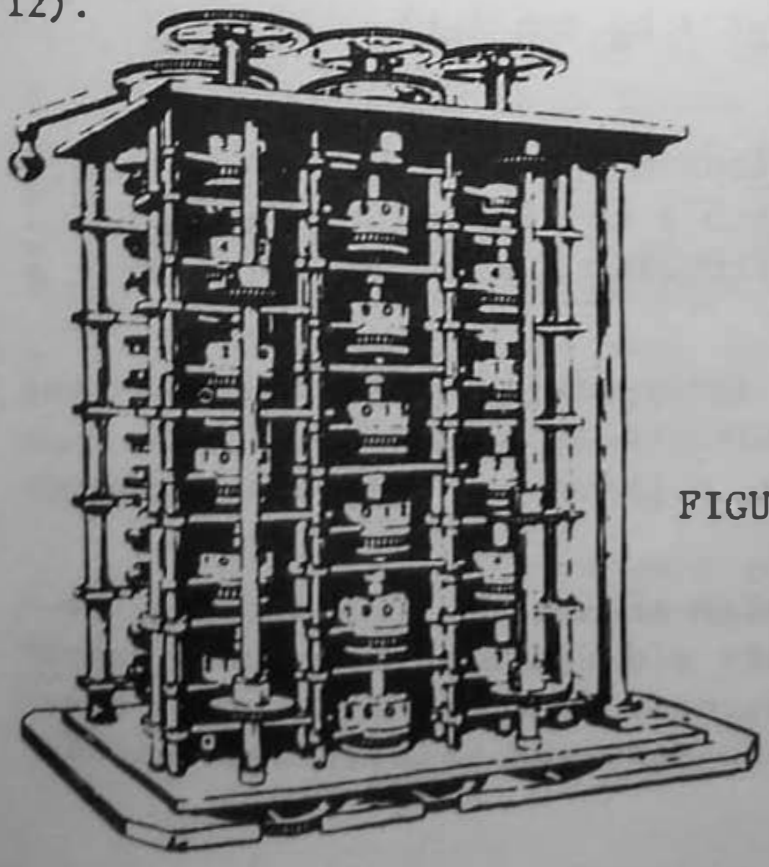


FIGURA 11

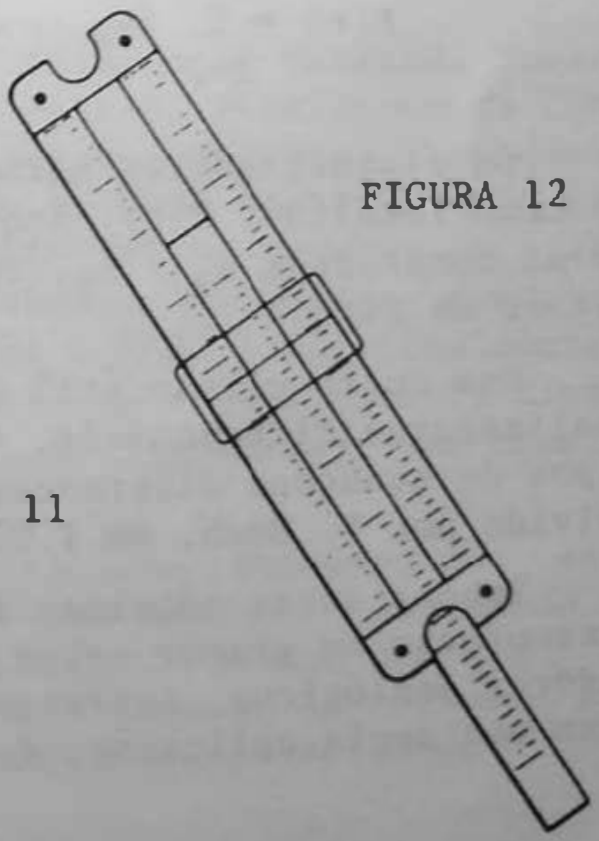


FIGURA 12

Entre outras operações aritméticas, a multiplicação e a divisão eram e ainda são executadas simplesmente somando-se - ou subtraindo-se a distância na régua externa àquela marcada na régua interna deslizante.

Um dos computadores mais antigos foi o Planímetro, construído por J. H. Hermann, em 1.814. O Planímetro era um dispositivo analógico que calculava a área sob uma curva, $\int_a^b F(t)\,dt$. A partir dele, muitos outros surgiram, destacando-se a máquina de James Thomson, em 1.876 e o computador de C.V. Boys, designado de Integraph, projetado em 1.881, para avaliar a integralidade de certas funções.

Em 1.872, Lorde Kelvin construiu uma máquina que gerava oito termos trigonométricos e somava-os. A máquina de Kelvin foi usada para prever a altura das marés nos portos ingleses. Ela foi a primeira a ser conhecida como Sintetizador Harmônico. Um outro instrumento semelhante, construído por Fry e Dietzold, em 1.937, foi o Isograph, usado na resolução de equações polinominais do tipo: $A_0X^n + A_1X^{n-1} + \ldots + An = 0$.

Kelvin estava também relacionado com o que é essencialmente o inverso do sintetizador harmônico: o analisador harmônico. Esse, construído em 1.876, foi baseado no mecanismo básico do Planímetro de seu irmão, James Thomson. Os estudos de Kelvin foram baseados na resolução da série de Fourier, cuja forma corresponde a:

$$F(t) = \sum_{R=0}^{n} (a_R \cos \lambda_R t + b_R \operatorname{sen} \lambda_R t)$$

Os sintetizadores harmônicos calculam a somatória, obtendo como resultado F(t). Enquanto o analisador harmônico calcula as constantes a_R e b_R, construindo a série de Fourier, a partir de F(t).

Uma outra classe útil de instrumentos é conhecida como analisadores diferenciais, que fornecem a resolução de certos tipos de equações diferenciais. O primeiro desses foi desenvolvido por V. Bush, em 1.927.

Embora essas máquinas fossem cheias de imperfeições, representaram um grande salto para o desenvolvimento dos computadores analógicos. Entretanto, esses computadores não possuem uma ampla aplicação, devido aos seus valores imprecisos.

Isso fez com que a tecnologia concentrasse seus interesses no desenvolvimento dos computadores digitais.

DESENVOLVIMENTO DOS COMPUTADORES DIGITAIS

Dois desenvolvimentos na eletrônica prepararam a chegada do computador digital automático. O primeiro ocorreu em 1.919, quando Eccles e Jordan demonstraram que a válvula termiônica poderia possuir dois estados estáveis comutáveis entre si. O segundo foi em 1.931, quando Wynn-Willians descreveu como utilizar essas válvulas eletrônicas como dispositivos de contagem. Estas idéias juntas com aquelas de Babbage foram incorporadas em um projeto de um computador digital, proposto por Willian Phillips em 1.934, exatamente cem anos depois que a Máquina Analítica de Babbage fora construída. No ano seguinte, Phillips ofereceu as patentes provisórias do seu projeto ao governo Britânico, sem, no entanto, receber estímulo algum. Até que, em 1943, ele foi homenageado por John Womersley pelo seu projeto que iria se transformar no computador piloto ACE (Automatic Calculating Engine) da "National Physical Laboratory".

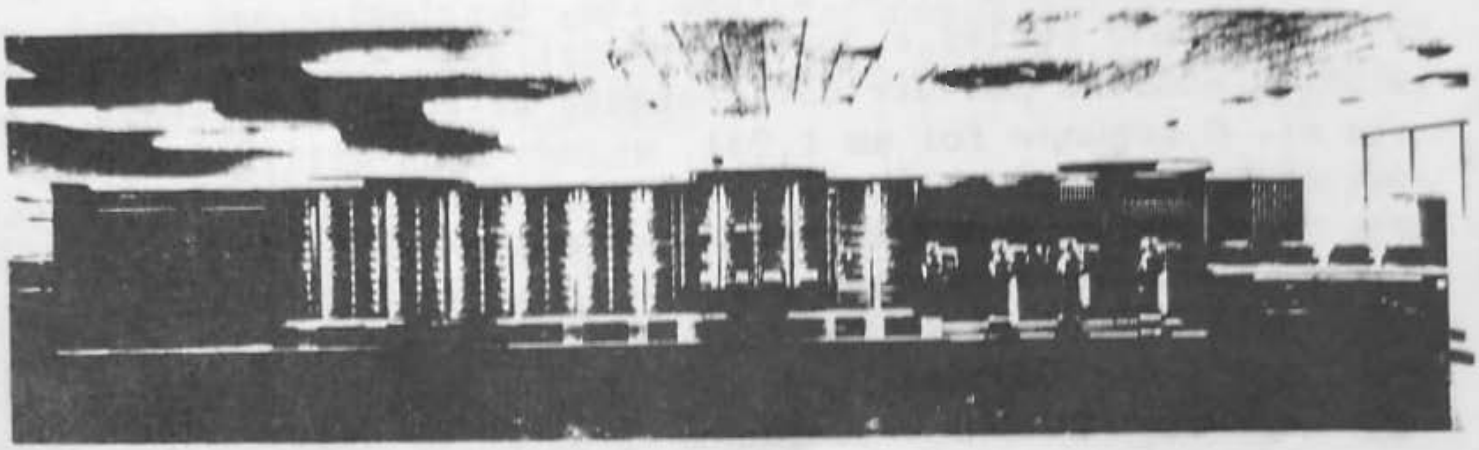
O PRIMEIRO COMPUTADOR COM RELÊS

Em 1.937, o professor da Universidade de Harvard, Howard G. Aiken associando-se a Thomas J. Watson, presidente da IBM, projetou uma máquina de função geral: o computador seqüencial, mais tarde conhecida por MARK I. Onde Babbage havia usado engrenagens dentadas para representar os números, a IBM usou combinações de chaves eletricamente operadas, denominadas de relês eletromecânicos. Quando o MARK I foi finalmente instalado na Universidade de Harvard, em 1.944, possuía dimensões espantosas, com milhares de relês ligados por uma complexidade de fios. (figura 13)

O MARK I foi um sucesso e um fracasso. Sucesso, por ser o primeiro computador completamente automático, trabalhando exatamente como Babbage havia imaginado. Fracasso por que o MARK I apresentou-se muito lento. Essa máquina de milhões de dolares, na sua melhor performance, era apenas cerca de uma

centena de vezes mais rápida do que o homem, usando uma calculadora. Mas vagaroso ou não, o MARK I serviu ao seu propósito, que era o de auxiliar as linhas de artilharia norte-americana durante a II Guerra Mundial.

FIGURA 13



O PRIMEIRO COMPUTADOR COM TUBOS A VÁCUO

O MARK I não era um computador eletrônico, mas sim, um eletromecânico. O primeiro computador digital eletrônico foi criado entre 1.939 e 1.946 na Escola de Engenharia Elétrica Moore da Universidade da Pensilvânia, por John W. Mauchly e J. Presper Eckert Jr. Os tubos a vácuo, ao todo 18.000, foram usados substituindo os relês. O ENIAC (Electronic Numerical Integrator And Calculator), como era chamado, pesava cerca de 30 toneladas e foi construído para o exército dos Estados Unidos a fim de calcular as tabelas da trajetória de sua artilharia. O ENIAC podia fazer em um dia o que uma operação manual requereria 300 dias para executar. Podia fazer 300 multiplicações em um segundo, enquanto os dispositivos eletromecânicos mais rápidos podiam executar, somente, uma multiplicação por segundo. A sua principal desvantagem era que ele fora projetado para um conjunto particular de problemas, tornando, por isso, a mudança de programação muito lenta. O ENIAC foi transladado para os campos de prova de Aberdeen, Maryland, em 1.947 e, em 1.956, foi mantido pela Instituição de Smithsonian. (figura 14).

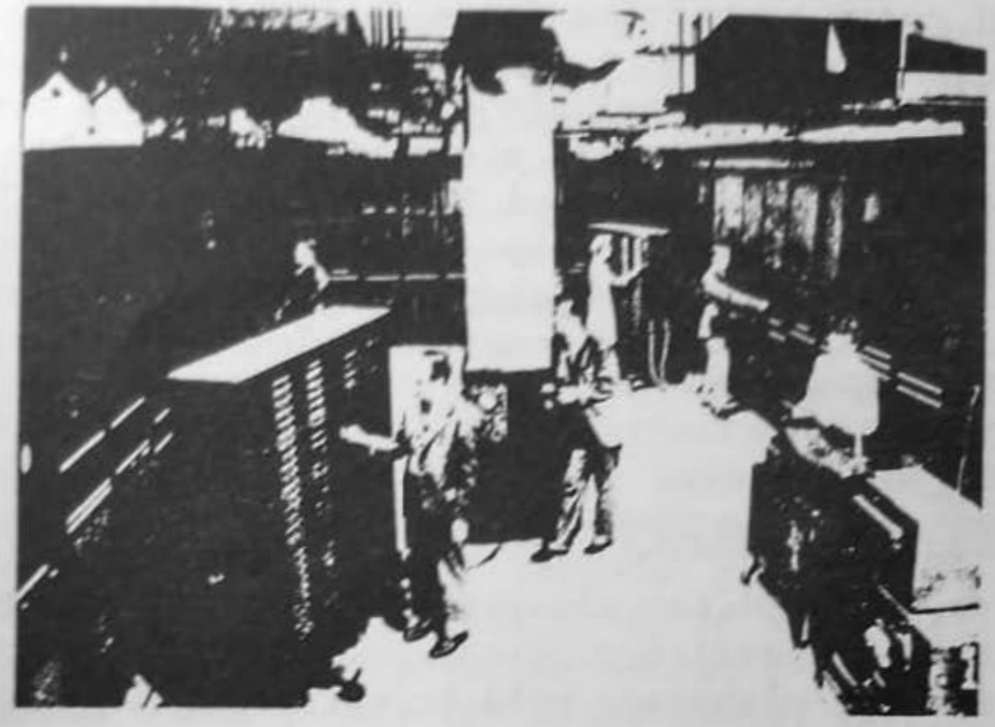


FIGURA 14

O PRIMEIRO COMPUTADOR BINÁRIO COM PROGRAMAS ARMAZENADOS

Em 1.945, um matemático chamado John Von Neumann, enviou um relatório a um grupo ligado à Escola Moore, sugerindo que o sistema de numeração binário fosse usado nos computadores e que as instruções, bem como os dados, fossem compilados e armazenados internamente no computador. Estas sugestões tornaram-se uma parte básica da filosofia do projeto dos computadores. Uma vez que os componentes eletrônicos necessários para atender a apenas duas condições (ligado ou desligado, conduzindo ou não conduzindo, magnetizado ou não magnetizado) fossem requeridas, a velocidade de execução do computador seria grandemente aumentada. Em maio de 1.945, o Dr. Von Neumann demonstrou como as instruções do computador poderiam ser codificadas para números binários e então armazenadas internamente no computador, com os respectivos dados.

Com o auxílio desses novos conceitos, Mauchly e Eckert, juntamente com um grupo de estudantes da Universidade da Pensilvânia, começaram a construir a máquina sugerida por von Neumann, a qual seria mais tarde conhecida como EDVAC (Electronic Discrete Variable Automatic Computer). O projeto do

EDVAC foi posto em prática antes da complementação do ENIAC, porém só foi completado a partir de 1.952. Essa máquina foi considerada, nos Estados Unidos, o protótipo do computador de programas armazenados e ela ainda está sendo usada pelo exército de Aberdeen, para o qual foi designada. O EDVAC utiliza como entrada a fita de papel perfurada e um programa, que controla a seqüência das operações, foi colocado em sua memória. É a primeira máquina comercial eletrônica de processamento de dados do mundo.

OUTROS COMPUTADORES

Desde então muitas máquinas foram desenvolvidas. O EDSAC foi a última das quatro máquinas produzidas nos anos de 1.940. Seu projeto foi elaborado pela Universidade de Cambridge, Inglaterra, em 1.946, entrando em operação no ano de 1.949. Em 1.947, Eckert e Mauchly deixaram a Universidade da Pensilvânia e negociaram um contrato com o "National Bureau of Standards", constituindo a Eckert - Mauchly Corporation, onde construíram o primeiro computador comercial de grande escala: o UNIVAC I (Universal Automatic Computer). Esse computador ficou famoso por ter previsto a vitória do Presidente Dwight D. Eisenhower na eleição de 1.952. A Eckert-Mauchly Corporation tornou-se mais tarde uma divisão da Sperry Rand Corporation.

Na Inglaterra, o MADAM (Manchester Automatic Digital Machine) da Universidade de Manchester foi construído pela Ferranti Ltda. Na Universidade de Londres, o SEC (Simple Eletronic Computer) e o APEC (All-Purpose Eletronic Computer) foram desenvolvidos.

Desde 1.958, milhares de grandes e pequenos computadores têm sido fabricados comercialmente. Entre os fabricantes estão a "International Busines Machines Corporation", a "General Eletric Corporation", a "Radio Corporation of America", a "Sperry Rand Corporation". a "Honeywell, Inc.", a "Burroughs Corporation", a "National Cash Register", a "Control Data Corporation" e muitos outros.

Comparados aos recentes computadores, os da primeira geração (computadores a válvula) eram volumosos, inflexíveis e requeriam grande controle de temperatura, através de ar condicionado. Contudo, possuíam vantagens sobre as máquinas ele-

tromecânicas de processamento de dados. Sua velocidade era atribuída ao uso de válvulas eletrônicas e foram também os primeiros a permitirem programação interna por circuito. Foi essa última característica a responsável pela capacidade dos computadores de comparar e "tomar decisões lógicas", as quais seriam aplicadas nos cálculos e que, desde então, tornaram-se conhecidos como "máquinas pensantes".

Desde o ENIAC, a arquitetura dos computadores tem evoluído espantosamente. Os computadores tornaram-se menores, mais rápidos, mais seguros e incrivelmente mais baratos. O ENIAC pesava 30 toneladas e consumia 150 kW de potência, energia suficiente para iluminar duas mil lâmpadas de 75 watts. Os computadores da segunda geração (computadores a transistor, a partir de 1950) possuem uma capacidade maior, pesando cerca de 150 kgs, com um consumo inferior a 150 watts. Isto se deve à substituição das válvulas por transistores, que são mais eficientes e menores. No início de 1.960, começava-se a terceira geração de computadores, quando os transistores foram miniaturizados e incorporados em circuitos integrados. Cada circuito integrado equivale a cerca de dez milhões de transistores, agrupados em um chip de silício, correspondendo a 1/8 de uma caixa de fósforo. Com isso, a capacidade de memória do computador cresceu enormemente. O ENIAC podia armazenar vinte números de dez dígitos cada um, enquanto os computadores de terceira geração (computadores com circuitos integrados) podem armazenar milhões de números.

Se um matemático egípcio da antigüidade pudesse imaginar dois números de dez dígitos, ele desanimaria só em pensar na multiplicação desses números. Neste século, uma pessoa de nível médio levaria cerca de cinco minutos para multiplicar esses números. O MARK I requeria cerca de cinco segundos. O ENIAC podia multiplicar esses números em cerca de dois milésimos de segundo, enquanto um computador transistorizado podia executar essa operação em cerca de quatro bilhonésimos de segundo. Atualmente, a grande evolução dos circuitos integrados propiciou a introdução dos mini e microcomputadores e possibilitou a grande popularidade do computador nas diversas áreas da sociedade, não somente a nível comercial e científico, mas também como instrumento caseiro e pessoal.

O MINI E O MICROCOMPUTADOR

O primeiro minicomputador surgiu em 1965 exatamente quando os computadores digitais completavam vinte anos de existência. Esse minicomputador, o PDP-5 foi lançado no mercado pela empresa norte-americana DEC (Digital Equipment Corporation) e a partir desta data os minicomputadores evoluiram de forma surpreendente, passando a ter aplicações nunca antes imaginada, não exigindo assim mais altos investimentos para suas instalações e mesmo para sua compra.

O mundo técnico passou a utilizar mais o computador, tirando-o do ambiente de ar condicionado do bureau de processamento de dados e aplicando-o em oficinas, aviões, balanças, robôs etc.

Assim, em 1970 a empresa norte-americana Intel Corporation desenvolveu o primeiro microprocessador, o qual era constituído de um processador completo em uma única pastilha de circuito integrado. Esse microprocessador foi denominado de 4004 por possuir uma palavra de 4 bits de largura. Essa mesma empresa vendo o sucesso de seu lançamento, introduziu em 1972 um modelo mais evoluido o 8008 com oito bits e posteriormente outros como o 8080, 8085 etc.

Atualmente já se encontra no mercado mais de 150 chips de diferentes microprocessadores.

Esses microprocessadores vieram possibilitar o desenvolvimento dos microcomputadores pessoais a um baixo custo, como por exemplo, o TK-82C, CEDM Z80 BASIC, CP-200, DGT-100 etc. Hoje, já são disponíveis no mercado mais de 40 modelos de microcomputadores e várias centenas no mercado mundial.

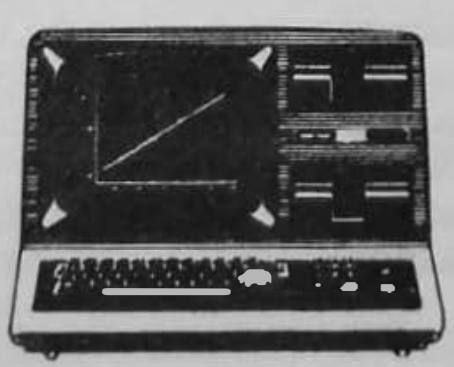
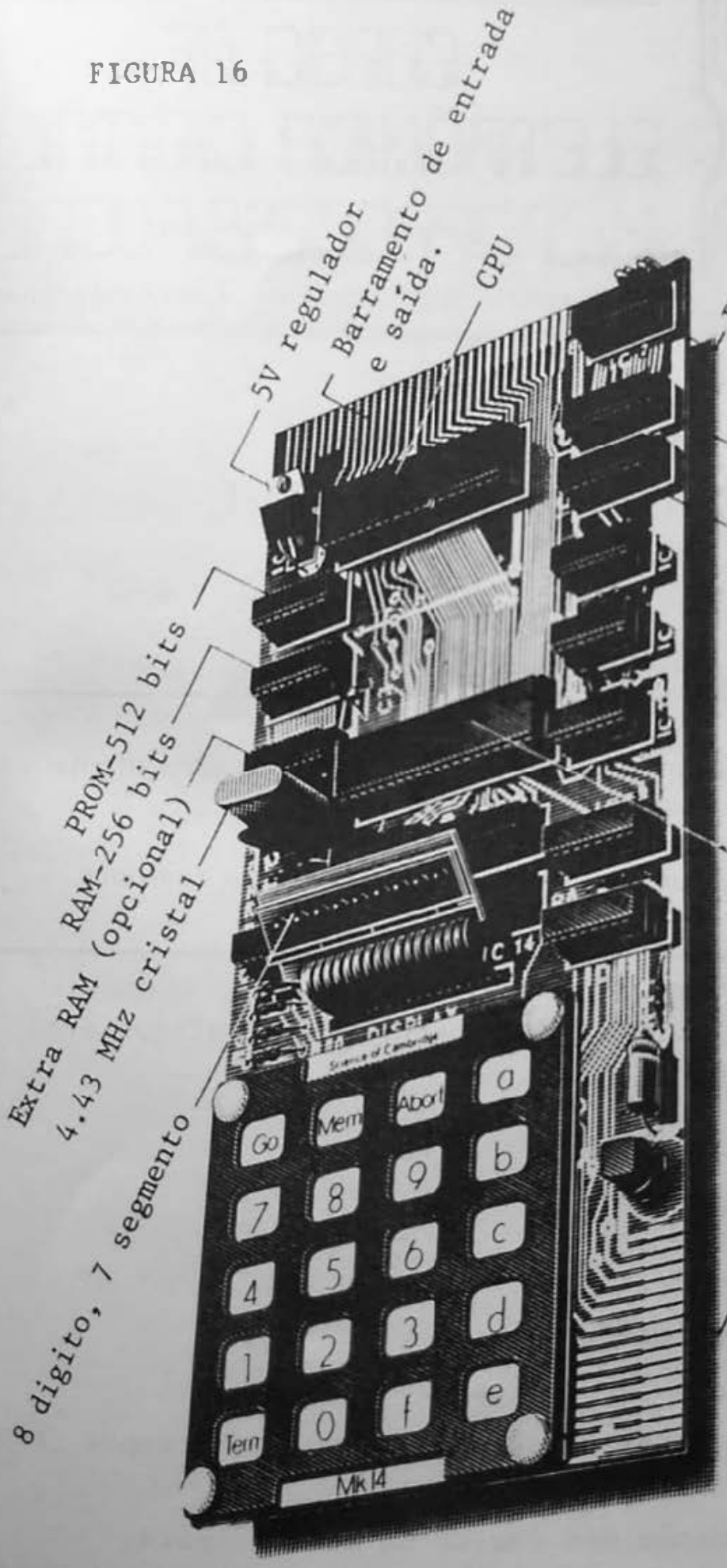


FIGURA 15

A figura 15 ilustra um microcomputador pessoal.

FIGURA 16



A figura 16 ilustra a placa de circuito impresso de um microcomputador.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 · CEP 86100 · LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-1

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) O instrumento de cálculo usado pelos povos antigos era:

   a) o computador analógico;

   b) o computador digital;

   c) a máquina de calcular;

   d) o ábaco.

2) O nosso sistema decimal surgiu devido:

   a) à máquina de calcular de Pascal;

   b) à contagem dos dedos usados pelo homem nos tempos remotos;

   c) ao ábaco que contém dez contas em cada fileira.

3) A primeira máquina de calcular foi inventada por:

a) Pascal; (P. 1642 D.C)

b) Babbage;

c) Hollerith.

4) O computômetro, inventado por Don Eugene Felt, em 1.887, foi a primeira máquina com um sistema de entrada de dados que hoje é usado por quase todas as máquinas de calcular. Esse é o sistema de:

a) Alavancas;

b) Engrenagem;

c) Teclado.

5) Dois fatores importantes antecederam o aparecimento dos computadores. São eles:

a) a construção do MARK 1, pelo professor Howard Aiken e a invenção do ábaco;

b) a invenção do cartão perfurado, por Jacquard e os projetos das máquinas de Babbage;

c) a introdução do sistema binário, por von Neumann e a invenção da válvula termiônica, por Eccles e Jordan.

6) Assinale a alternativa correta:

a) o primeiro dispositivo analógico foi o ábaco;

b) Babbage foi o primeiro a esboçar um computador;

c) Herman Hollerith foi o primeiro a usar o cartão perfurado;

d) o planímetro é um dos mais modernos computadores digitais.

7) O primeiro computador digital de grande repercussão foi projetado em 1.937 e sua constituição foi à base de relês. Seu nome era:

a) MARK I;

b) ENIAC;

c) EDVAC;

d) EDSAC.

8) A adoção do sistema binário e do armazenamento de programas nos computadores, que são usados atualmente, se deve a:

a) Lorde Kelvin;

b) Willian Phillips;

c) John von Neumann.

9) Mauchy e Eckert construíram o EDVAC que foi considerado, nos Estados Unidos:

a) o primeiro computador transistorizado;

b) o protótipo do computador de programas armazenados;

c) o computador mais veloz construído até hoje.

10) O primeiro computador produzido, comercialmente, em grande escala, foi o:

a) ENIAC;

b) EDVAC;

c) UNIVAC I;

d) EDSAC.

CEAM
CURSOS DE APERFEIÇOAMENTO

CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 · CEP86100 · LONDRINA · PARANÁ

LIÇÃO MC-2

ORGANIZAÇÃO DOS COMPUTADORES

INTRODUÇÃO:

Não intencionamos iniciar esta lição com a descrição detalhada do complexo físico que constitui o computador e, menos ainda, tratar de questões referentes às possíveis configurações ou aos sofisticados processos de fabricação de seus componentes. O nosso objetivo é justamente mostrar a forma como o computador se apresenta, ou seja, sua evolução e sua classificação.

CLASSIFICAÇÃO DOS COMPUTADORES:

Até alguns anos atrás, a palavra "computador" tinha um sentido muito vago e distante, apenas com significado para um limitado número de pessoas (geralmente cientistas e especialistas). Foi a época do "cérebro eletrônico", o que equivale a dizer que o computador era uma invenção exótica, logo, fora do alcance da grande maioria da população.

Hoje, devido ao grande avanço tecnológico experimentado por este ramo da técnica, verifica-se justamente o contrário, ou seja, a popularização do computador. O desenvolvimento dos computadores eletrônicos tem sido considerado como o começo da segunda revolução industrial.

Acompanhando o desenvolvimento da tecnologia, a arquitetura dos sistemas tem recebido constantemente uma série de melhoramentos. Também evoluíram sistemas de programação (software), onde os programas monitores fazem a "escalação" dos serviços e regulam o fluxo de trabalho através do sistema. Por exemplo: os conceitos que se tinham sobre as máquinas de "programas armazenados" foram desenvolvidos somente a partir do surgimento dos componentes eletrônicos. Desse modo, podemos dizer que a evolução dos computadores está fortemente ligada ao desenvolvimento da eletrônica. Considerando esses fatos, é comum classificarem-se os computadores em classes ou gerações distintas, a saber:

1a. GERAÇÃO (1946-1956)

Na primeira geração, a tecnologia era caracterizada pela válvula a vácuo, memória de linhas de atraso ou de tubo de raios catódicos eletrostáticos. Já, no final dessa geração, foi introduzida a memória de núcleo de ferrite. Alguns sistemas (como no caso do ENIAC) utilizavam tambores magnéticos para memória principal.

É nessa geração que encontramos o UNIVAC I, o primeiro computador a ser comercializado.

2a. GERAÇÃO (1956-1963)

Como conseqüência do descobrimento do transistor, em 1948, se inicia a segunda geração de computadores, onde a válvula eletrônica é substituída pelo transistor, reduzindo consideravelmente o tamanho e também o consumo de energia. Transistores, capacitores e resistores eram então montados em placas de circuito impresso. A memória principal era de núcleo de ferrite, e o armazenamento secundário feito com tambores e fitas magnéticas.

Muito embora em 1956 já existissem computadores com transistores, sua entrada no mercado mundial se deu pouco antes de 1960. Os computadores dessa geração dividem-se em duas

categorias: "comercial" (mais lento, utiliza números decimais) e o "científico" (mais veloz, utiliza números binários, com instruções que facilitam operações com ponto flutuante). Nessa época, começou-se a empregar o FORTRAN, de tipo elementar, como linguagem de programação.

3a. GERAÇÃO (1964-1971)

Nessa época, inicia-se a aplicação dos circuitos integrados na construção dos computadores. São vários os aspectos que se podem apontar para classificar um computador da terceira geração. Entre esses aspectos destacamos:

a) Utilização de memórias mais rápidas (ciclo básico da ordem de um microssegundo por palavra);

b) Fabricação de memórias especiais (núcleos de ferrite de dimensões reduzidas);

c) Utilização de circuitos integrados;

d) Execução simultânea de programas, através da evolução do sistema operacional, também conhecido por software. O software são programas do sistema que apoiam os programas de aplicação do usuário;

início do CP/M

e) Introdução da multiprogramação (onde pode haver mais de um programa na memória principal ao mesmo tempo);

f) Cablagens especiais programadas;

g) Utilizam-se na programação o FORTRAN, o COBOL e os sistemas operativos.

Um dos primeiros computadores de 3a. geração, lançado no mercado foi o IBM 360. Um outro exemplo seriam os sistemas - UNIVAC 90/30.

A idéia de "geração" vem sendo duramente criticada, pois a noção de gerações distintas esbarra em muitas exceções. Por exemplo, o computador CDC 6600 é um sistema em grande parte de arquitetura típica da terceira geração, porém, foi implementado com transistores discretos, o que o caracteriza como um computador de segunda geração. Apesar dessas críticas, hoje em dia já se fala em quarta geração.

4ª. GERAÇÃO (1971)

Em 1971, a INTEL fabrica, em um só chip de circuito integrado, LSI (alta escala de integração), a "Unidade Central de Processamento" (UCP) de um computador, utilizando a tecnologia MOS. Nasce assim, o "MICROPROCESSADOR". A partir desse momento, esta e outras fabricas voltaram-se para a pesquisa e o desenvolvimento de microprocessadores cada vez mais potentes e velozes.

Na quarta geração se empregam largamente as memórias de semicondutores integrados, com capacidades de armazenamento cada vez maiores. É nessa mesma época em que começam aparecer novas linguagens de programação como, por exemplo, o BASIC, o ALGOL, o PL/1, etc.

A introdução do microprocessador permitiu a construção do computador pessoal ou doméstico.

O microprocessador permitiu o emprego de sistemas de processamento em todas as áreas e a todos os níveis. Tanto é que, hoje em dia, podemos considerar o microprocessador como sendo um dispositivo eletrônico com características muito sofisticadas.

A concepção de um microcomputador, construído em um único chip, nos leva a considerar o aparecimento, em um futuro bem próximo, da 5a. geração, a qual, ao que tudo indica, possibilitará a aplicação em massa dos microcomputadores em setores ainda inexplorados.

Uma segunda classificação, muito utilizada, é feita com base na constituição interna da Unidade Central de Processamento (UCP), a qual pode ser considerada como sendo a parte fundamental do computador. Assim, os computadores podem ser classificados como: Analógicos ou Digitais.

Os computadores analógicos são implementados com dispositivos eletrônicos, capazes de manipular magnetudes variáveis. Os amplificadores operacionais, devido à grande capacidade de realizarem operações matemáticas, alterando sua realimentação, são os elementos mais usados neste tipo de computador. Uma das características do computador analógico é a sua facilidade em simular processos reais. A causa do progressivo abandono dessa tecnologia na fabricação de computadores se deve à insuficiente precisão dos resultados bem como a sua con

cepção orientada para um campo de aplicações muito reduzido.

Quanto aos computadores digitais, esses são construídos, basicamente, de componentes e circuitos da eletrônica digital, os quais caracterizam-se por manipular apenas dois níveis de magnetude, ou seja, nível alto (1) e nível baixo (0). A utilização dessa tecnologia possibilita a construção de computadores com estrutura mais simples, mais confiável e com maior precisão do que a tecnologia dos computadores analógicos.

Nos computadores digitais as operações são realizadas descontinuamente, por etapas distintas, muito embora ligadas entre si. Podemos dizer que um computador digital pode executar qualquer trabalho, desde que esse seja susceptível de decomposição em partes elementares, as quais deverão estar ao nível das instruções ou do conjunto delas. O problema ficará resolvido, quando se associarem esses conjuntos pela ordem correta (a associação desses conjuntos forma os programas).

Dentro dos computadores digitais, levando-se em conta o momento atual, podemos fazer a seguinte divisão:

1º) Sistemas com microprocessador;

2º) Microcomputadores;

3º) Minicomputadores;

4º) Computadores.

Os microcomputadores são máquinas que processam dados mediante uma UCP, constituída de um microprocessador segundo um programa de instruções armazenados na memória. A principal diferença entre "sistema com microprocessador" e "microcomputadores" é que no primeiro, o programa de instruções é fixo, sendo que a concepção física de seu circuito eletrônico é desenvolvido para resolver apenas um determinado tipo de problema, por exemplo, o controle da velocidade versus consumo de combustível de um certo tipo de automóvel. Já, o microcomputador foi concebido com propósito mais amplo, ou seja, não possui um programa fixo de instruções, podendo, portanto, ser "carregado" com qualquer programa desde que esse esteja de acordo com suas especificações.

Tanto os sistemas com microprocessador como os microcom-

putadores são enquadrados na 4a. geração, pois são organizados em torno de uma CPU, fabricada em um único chip. As principais características desses dois primeiros tipos (1º e 2º) são:

a) Velocidade do ciclo de instrução da ordem de 0,5μs(microssegundos);

b) Trabalham com palavras de vários bits (4, 8, 16 e 32 bits);

c) Capacidade de direcionamento da memória da ordem de 64K bytes;

d) Baixo preço. Tendência de fabricação de todo o sistema em um só chip.

Já os minicomputadores são máquinas similares aos grandes computadores surgidos na 3a. geração. Suas características mais importantes são:

a) Preço bastante inferior ao dos computadores;

b) Velocidade do ciclo de instrução maior que a dos microcomputadores;

c) Trabalham com palavras de 16 e 24 bits;

d) Linguagem de programação de alto nível;

e) Elevada capacidade de memória (64K bytes de memória central e até 30M bytes de memória periférica, à base de disco).

Finalmente, o grupo dos computadores propriamente dito - se distingue por possuir características muito superiores às dos grupos anteriores (1º, 2º e 3º grupos). Eis algumas dessas características:

a) Elevado preço de custo;

b) Volume (dimensões) superior ao dos demais grupos;

c) Capacidade de memória praticamente ilimitada;

d) Velocidade de trabalho inferior a 30 ns (nanossegundos);

e) Devido à grande velocidade e capacidade, o computador pode trabalhar para vários usuários ao mesmo tempo;

f) Utilização de sistemas conversáveis entre o computador e o usuário, permitindo a este controlar todas as fases de desenvolvimento do processamento;

g) Possibilidade de controlar, com um grande computador, outros de categoria inferior;

h) Linguagem de programação de alto nível, orientado para resolver problemas de diferentes áreas.

Outra classificação importante é feita com base na aplicação dos computadores os quais podem ser divididos em:

- Científicos e
- Comerciais.

De uma maneira bem resumida, podemos dizer que um computador científico se caracteriza por:

a) Pequeno volume de dados (entrada-INPUT);
b) Pequeno volume de resultados (saída-OUTPUT);
c) Grande volume de cálculos de natureza aritmética;

Em contrapartida, um computador comercial, para a mesma "grandeza", será caracterizado por:

a) Grande volume de dados (entrada-INPUT);
b) Grande volume de resultados (saída-OUTPUT);
c) Pequeno volume de cálculos de natureza aritmética.

Visto como se classificam os computadores, passemos então ao estudo dos elementos básicos que os constituem.

ORGANIZAÇÃO INTERNA DOS COMPUTADORES:

Com o objetivo de dar uma visão geral do desenvolvimento e aplicação da eletrônica digital no campo da computação, apresentaremos doravante uma série de informações correspondentes à 4a. e 5a. gerações (microprocessadores e microcomputadores).

Fundamentalmente, todos os sistemas de computadores são iguais, pois as partes elementares que os constituem são as mesmas.

Todo computador digital é formado por três blocos funda-

mentais, ou seja:

a) UCP (Unidade Central de Processamento);
b) Memória;
c) Unidades de Entradas e Saídas (E/S).

Além desses, temos ainda um gerador de pulsos de relógio, mais conhecido como CLOCK, o qual é responsável pela sincronização do trabalho da máquina. A arquitetura (circuito) fundamental de um computador digital é mostrada na figura 1.

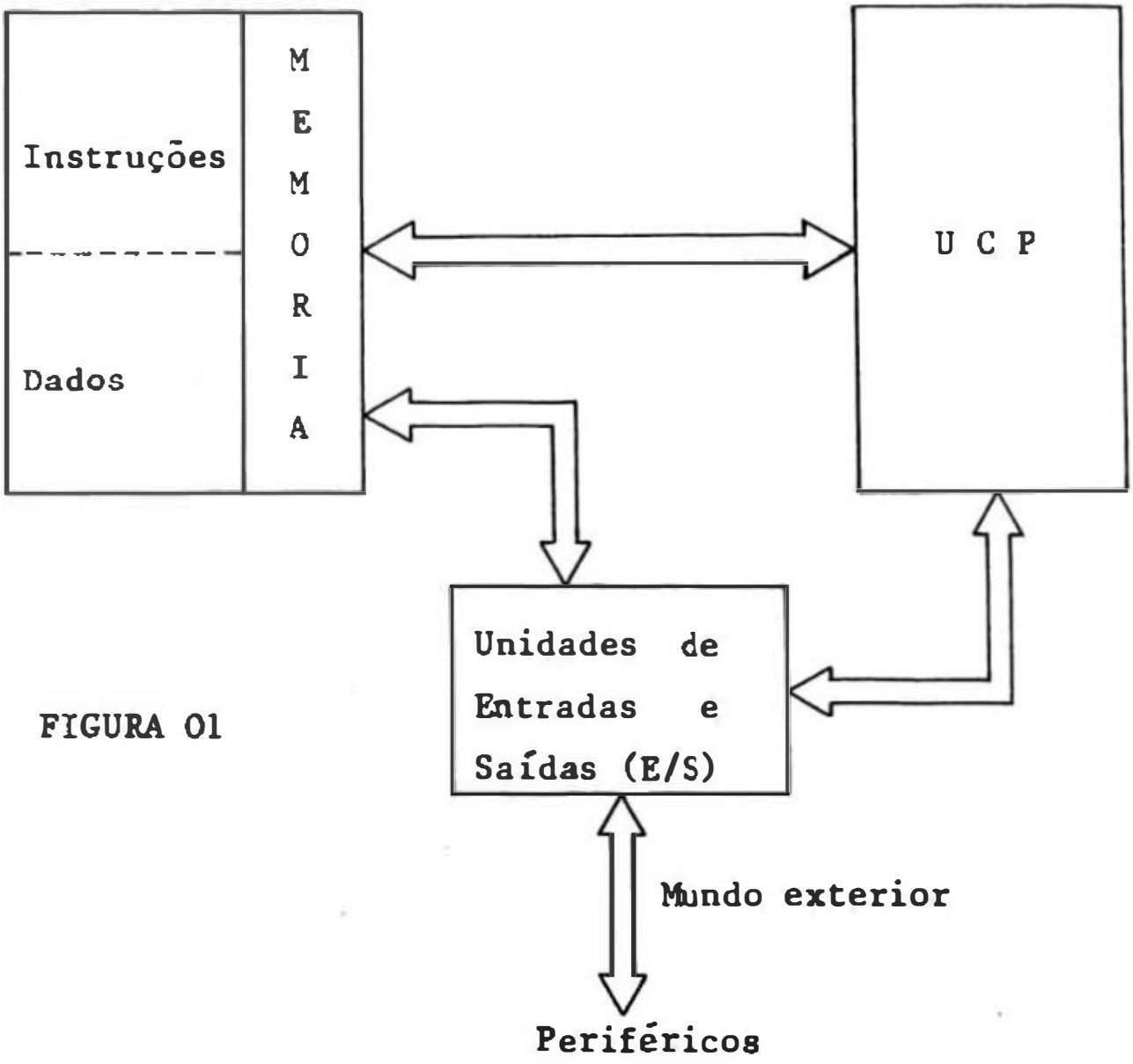


FIGURA 01

A UCP é formada por uma série de elementos auxiliares, - tais como: Unidade de Controle, Unidade Lógica Aritmética (ULA), registros de trabalho, gerador de clock, etc. Quando todo esse conjunto é fabricado sobre um único chip de um cir-

cuito integrado, esse passa a ser chamado de MICROPROCESSADOR.

É a própria UCP que cria e controla o fluxo de dados que circula pelo computador, a partir das instruções recebidas da memória, as quais servem para indicar as operações ou tratamento a serem realizados sobre os dados recebidos do exterior ou da própria memória. Portanto,podemos considerar a memória, juntamente com a UCP, como sendo a parte mais importante do computador, faltando para completar a sua estrutura apenas as Unidades de Entradas e Saídas (E/S), cuja função é adaptar os códigos das informações permutadas pelo processador e os periféricos, os quais relacionam o sistema com o mundo exterior.

O programa (ou seqüência de instruções) que a máquina deve seguir para realizar o processamento dos dados está armazenado em uma parte da memória, denominada "memória de instruções". A outra parte da memória é utilizada para armazenar dados e resultados de forma temporal, ou seja, por um tempo determinado.

A Unidade de Controle decodifica as instruções recebidas da memória e gera, em conformidade, os sinais necessários para que a ULA, bem como o restante do sistema, execute as suas funções. Por fim, os conjuntos de registros são destinados ao armazenamento temporário de informações (dados, instruções e endereços).

O grande desenvolvimento tecnológico,verificado nos últimos anos no campo da eletrônica digital nos levou ao "microcomputador", objeto de nosso estudo, de agora em diante.

OS MICROCOMPUTADORES:

Um microcomputador possui, em princípio, a mesma construção e opera de maneira semelhante ao computador. De uma maneira bem resumida, podemos dizer que um microcomputador é um sistema eletrônico que incorpora um microprocessador (unidade central)e,em torno do qual,existem memórias, interfaces,gerador de clock, etc. A figura 2 nos mostra os componentes básicos de um microcomputador.

A estrutura mais comum de um microcomputador é organizada em barramentos (BUS) de dados e endereço, aos quais se li-

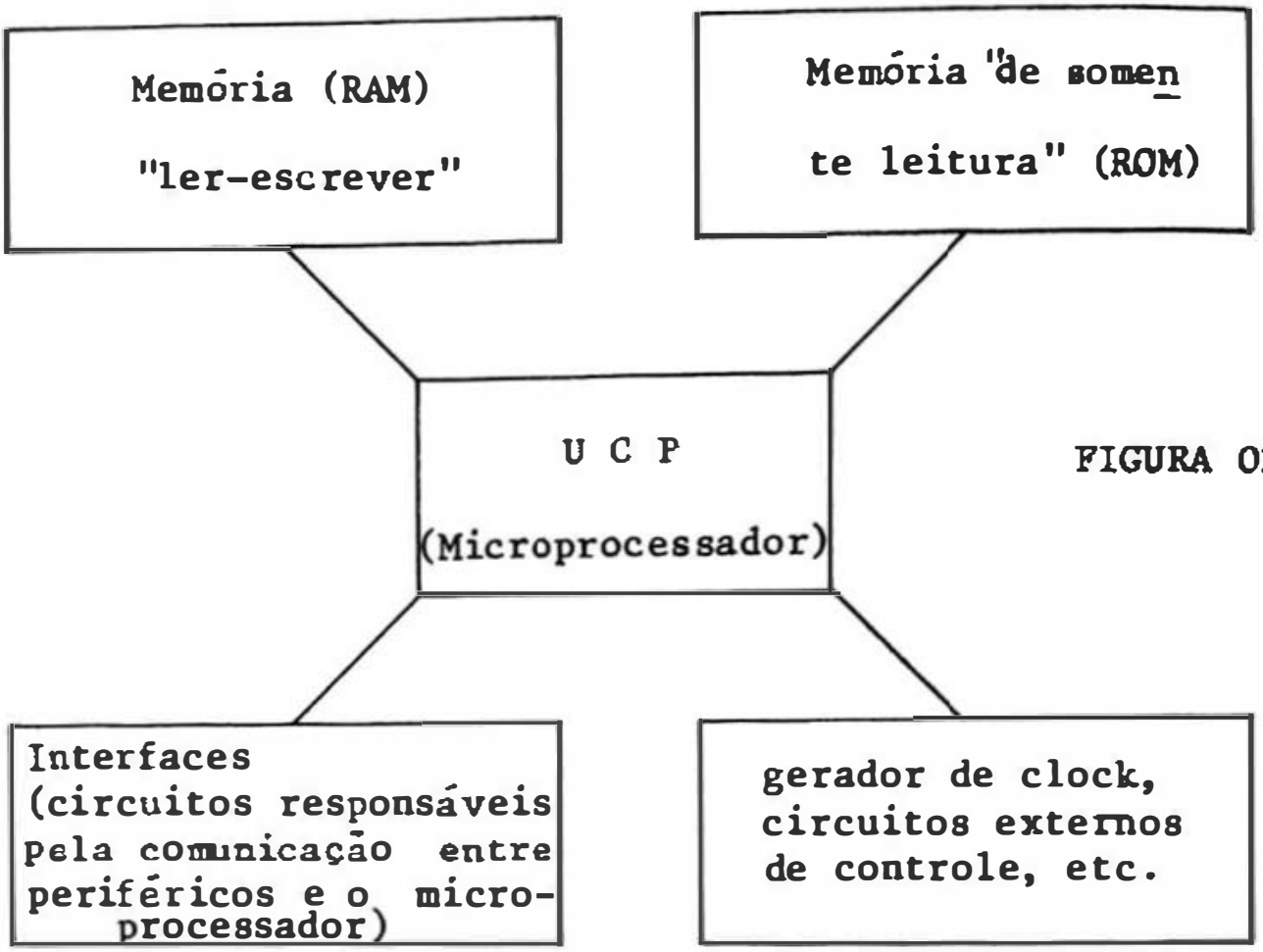


FIGURA 02

gam os vários componentes (ou unidades). Pelo barramento, estas unidades recebem dados, endereços de memória e sinais de controle. A largura do barramento de dados, que varia de 4 a 16 bits, determina o tamanho da palavra do microprocessador. Em estruturas desse tipo, existem dois aspectos básicos na administração geral do sistema:

a) O microprocessador controla os barramentos, informando para as demais unidades o que devem fazer a cada instante;

b) Uma das unidades requisita o controle dos barramentos para se comunicar com outra unidade, que não o microprocessador.

A figura 3 nos mostra o sistema de microcomputador, cuja estrutura é organizada em barramentos (ou vias).

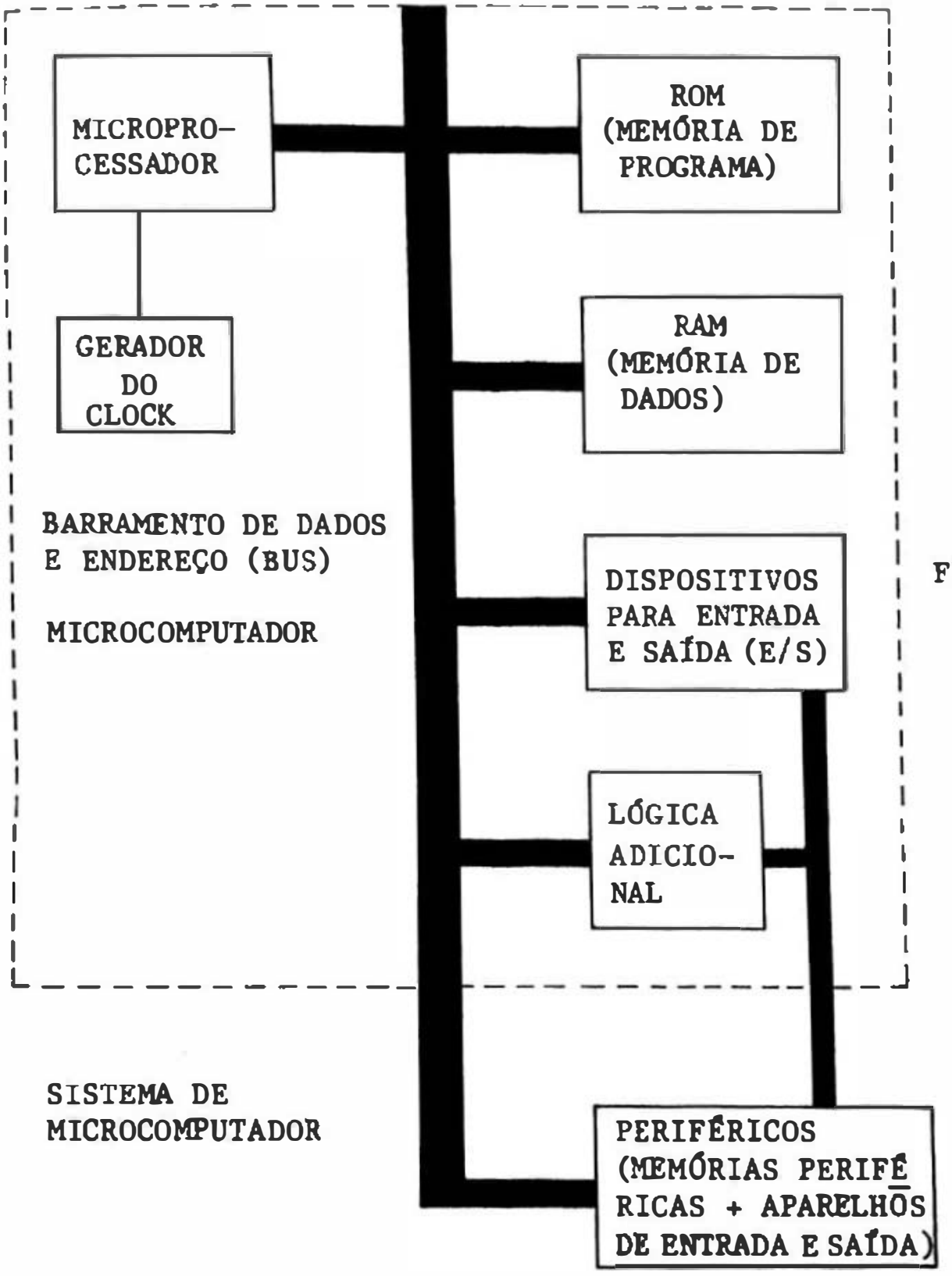


FIGURA 03

Devido ao fato de que a maior parte do intercâmbio de informações que ocorrem nos barramentos de endereço e dados serem comunicações entre o microprocessador e as demais unidades, é natural que o controle do barramento fique sob responsabilidade do microprocessador. Nas raras situações em que as vias são usadas para intercâmbio entre duas unidades, sem a participação do microcomputador, é necessário que uma dessas

unidade assuma o controle do barramento. Nesse caso, o microprocessador sinaliza a permissão do uso dos barramentos e os libera.

A informação vinda da UCP, ou de outros subsistemas do microcomputador, pode ser de três tipos: dados, controle e direção; pelo que, funcionalmente se encontram três tipos de barramentos (buses) no microcomputador, muito embora não estejam, sempre, fisicamente separados, pois podem estar multiplexados ao mesmo tempo sobre linhas comuns. O diagrama de bloco apresentado na figura 3 se refere a um sistema de microcomputador com barramento (BUS) multiplexado. Já, o diagrama de blocos, apresentado na figura 4, nos mostra as interconexões entre os diversos subsistemas de um microcomputador através de barramentos independentes.

Os barramentos de direção são linhas que vão da UCP para a memória e para o sistema ou unidade de Entradas e Saídas (E/S). Essas linhas serão sempre controladas pela UCP, salvo em caso de transferências por acesso direto à memória. Sua função é enviar a direção do elemento selecionado para uma transferência.

Já o barramento de dados é formado por linhas bidirecionais (ou dois barramentos unidirecionais independentes) que servem para a transmissão de informações (instruções ou dados) entre subsistemas. Sua largura, em geral, é igual ao comprimento da "palavra" do microcomputador (4, 8, 16 bits).

Por fim, os barramentos de controle são linhas de entrada ou saída da UCP, de ordenação ou de informação de estado, que permitem coordenar a operação de todo o sistema.

O sistema de memória é organizado em palavras. O comprimento da palavra, em geral, é o mesmo que o da palavra do microprocessador e do barramento de dados.

As memórias utilizadas em microcomputadores são, normalmente, de semicondutores.

Outros tipos de memória utilizados para aplicações especiais são as memórias de acesso seqüencial, as quais possuem uma grande capacidade de armazenamento e um tempo de acesso relativamente alto. Existem também as memórias auxiliares de grande capacidade e baixo custo, como, por exemplo, as cintas magnéticas, discos e fitas cassetes, que por sua forma de

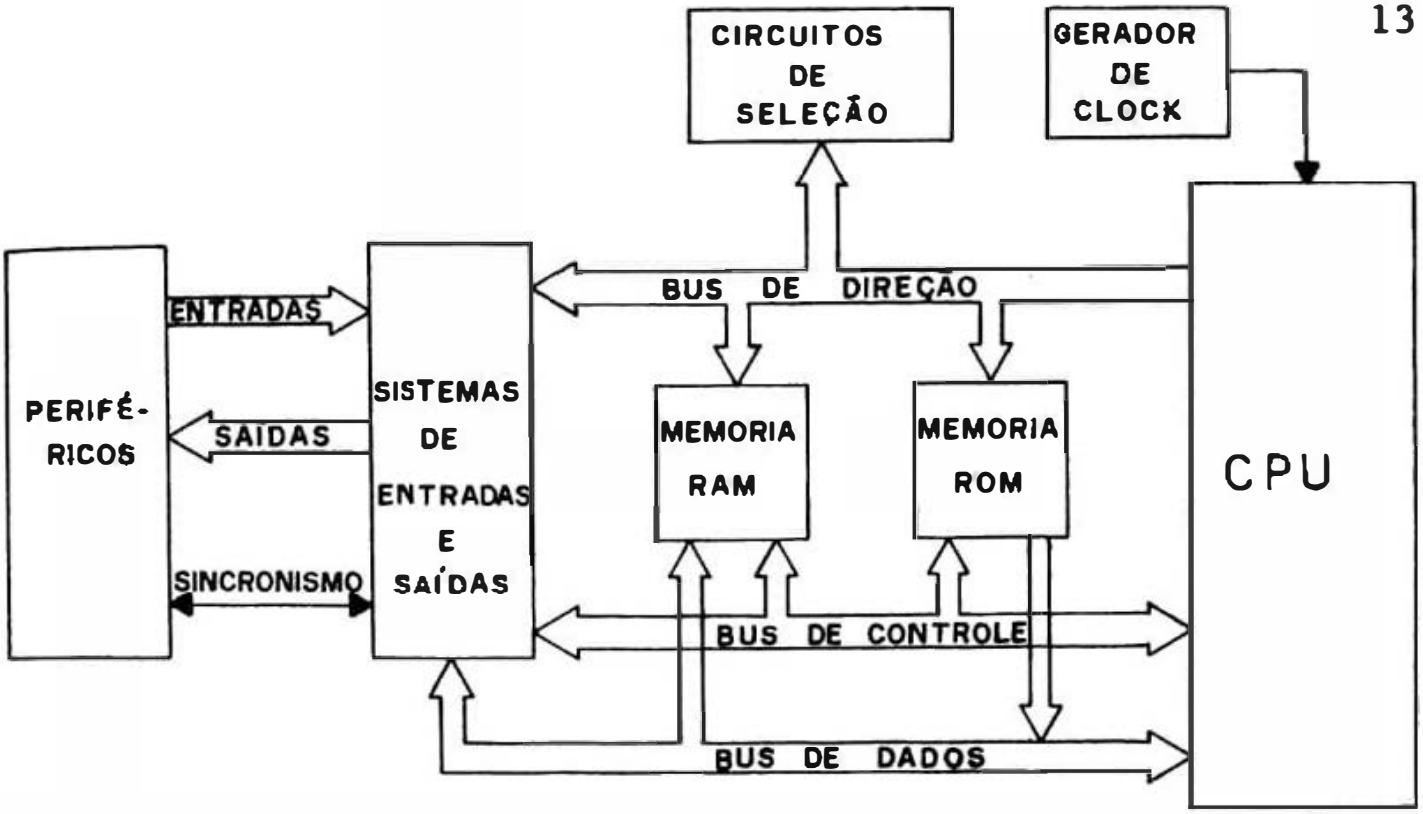


FIGURA 04

acesso são consideradas como periféricos do microcomputador.

As memórias de semicondutores de acesso aleatório são fabricadas, utilizando-se de várias tecnologias (TTL, PMOS, CMOS, NMOS, I^2L, ECL...) com características distintas em tempo de acesso e possibilidades de integração e consumo.

As memórias do tipo "apenas leitura" são utilizadas nos microcomputadores para armazenarem programas, tabelas e dados fixos (constantes). A grande vantagem desse tipo de memória é que o usuário não tem que se preocupar em carregar o programa cada vez que liga o sistema. Além disso, essas memórias não são voláteis, ou seja, não perdem seu conteúdo quando a alimentação é desligada. Existem três tipos básicos de memórias de "apenas leitura": ROM, PROM e EPROM.

Na memória ROM (read only memory), a programação da memória é feita durante a sua fabricação. A memória do tipo PROM (Programmable read only memory) são memórias de apenas leitura, porém programáveis pelo usuário e uma vez programada não pode ser modificada já que o processo de programação utilizado é irreversível. Finalmente, a EPROM (erasable programmable ready-only memory) apresenta a característica de ser programável, sendo portanto, muito recomendada quando da realização de um protótipo, pois,nessa fase, devem-se esperar muitas al-

terações nos programas, até se conseguir a solução final.

Para armazenar dados são utilizadas memórias do tipo "Ler-escrever" ou RAM (Randow acccss memory). Essa é a memória convencional onde se grava e se lê informação. As memórias RAM são voláteis, ou seja, a informação armazenada será perdida - quando houver falha na alimentação do sistema. Para não se perder a informação por falta de alimentação, deve-se dispor de um sistema de baterias, especialmente,para a alimentação da RAM. Basicamente, existem dois tipos de memórias RAM: estáticas e dinâmicas.

Na memória do tipo RAM estática, a informação é armazenada em um biestável, de forma que o conteúdo se mantém desde que haja alimentação. Já nas memórias do tipo RAM dinâmica a informação é mantida sob a forma de carga de uma junção porta-substrato de um transistor MOS. Essa carga, do tipo capacitiva, dissipa-se com o tempo (da ordem de milissegundos), requerendo,portanto,um reforço periódico (a cada ciclo de leitura escrita) em cada bit de memória, o que constitui uma desvantagem desse tipo de memória. A principal vantagem das memórias dinâmicas é que elas permitem uma maior densidade de integração.

Um outro bloco básico utilizado nos microcomputadores é o Sistema de Entrada e Saída (E/S). Esse sistema permite a comunicação do microcomputador com o mundo exterior. De modo idêntico ao que se passa com as posições de memória, as unidades E/S devem ser endereçáveis, permitindo ao sistema escolher uma fonte de informação, tal como um teclado, ou escolher o destino de informação, tal como um "display" de 7 segmentos. As fontes e os destinos de informação são designados por "periféricos".

Uma das unidades E/S mais simples é a de entrada direta, cujo diagrama de blocos é mostrado na figura 5.

Uma outra unidade de E/S, também muito simples, é a de saída direta (veja figura 6). A sua constituição difere da anterior (unidade de entrada direta) por possuir, como porta, (de saída) um conjunto de trincos bi-estáveis (latches), em lugar do conjunto de tri-state + buffers.

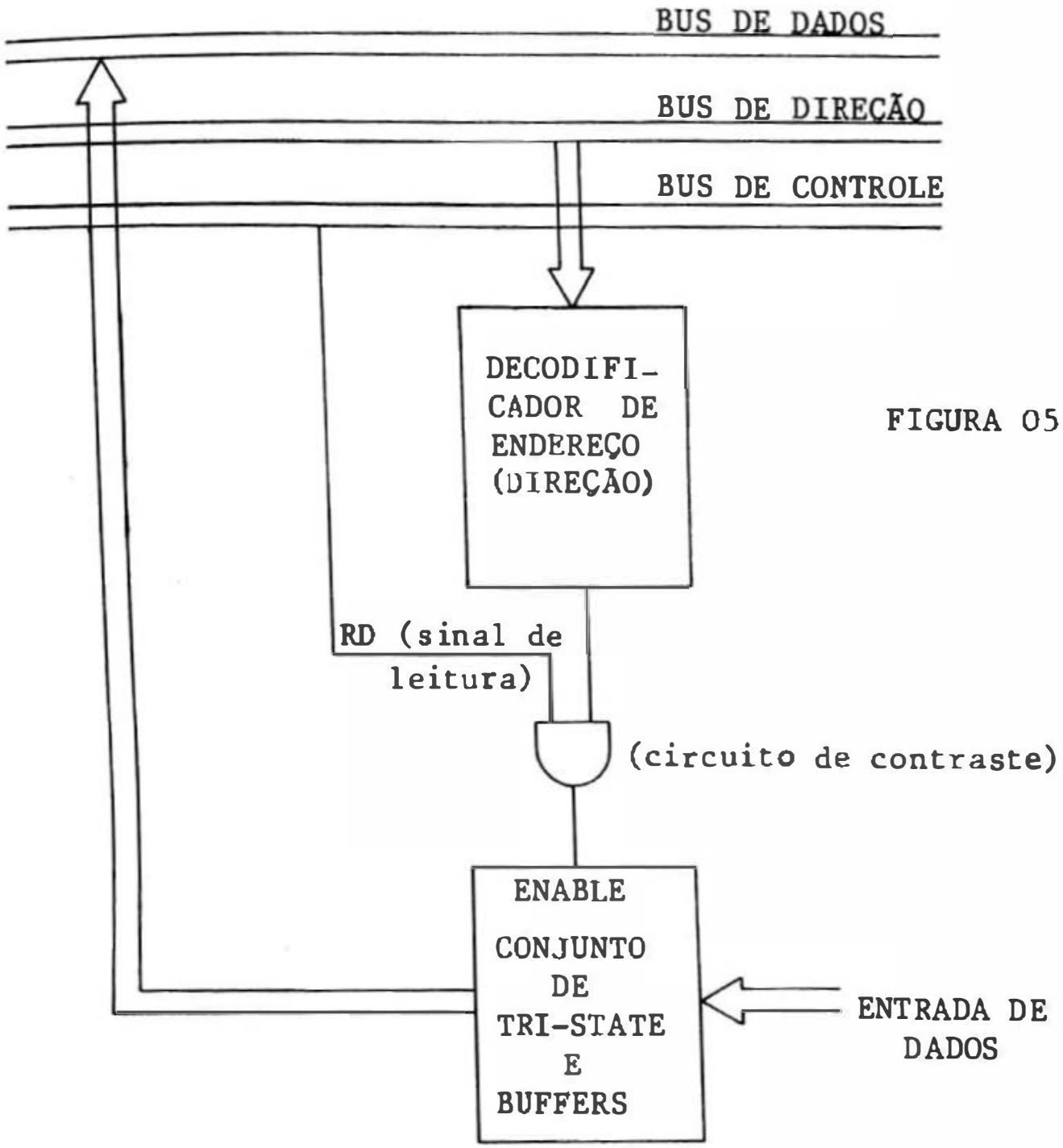


FIGURA 05

Uma unidade de entrada direta e uma unidade de saída direta podem ser agregadas de modo a formarem uma única unidade de entrada/saída, com um endereço único.

Existem duas técnicas de organização das portas E/S de um microcomputador:

a) Organização por endereçamento de memória(Memory Mapped I/O), representada na figura 7.

b) Organização por endereçamento próprio (Isolated I/O), representada na figura 8.

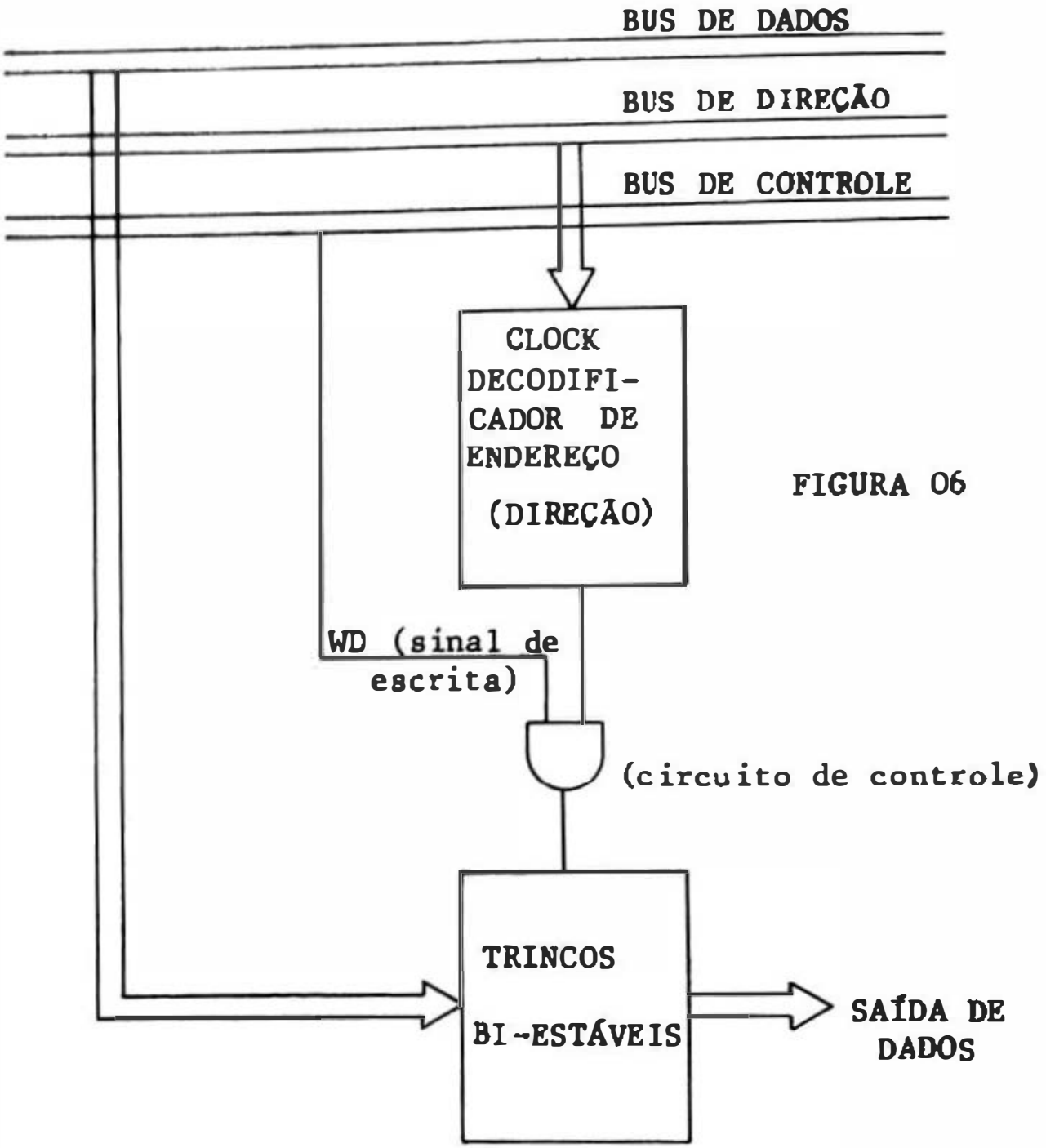


FIGURA 06

Na técnica de endereçamento de memória, as unidades E/S são tratadas exatamente como se fossem posições de memória, - permitindo assim que qualquer instrução de leitura ou escrita da memória seja aplicada a qualquer unidade E/S. Já na técnica de endereçamento próprio, utiliza-se uma linha especial do barramento de controle, designada por ES/$\overline{M}$ (Veja figura 8), para distinguir a memória das unidades E/S. Quando tivermos nessa linha o valor lógico 0, teremos um acesso à memória e, quando o valor lógico, nessa linha, for igual a 1, implica em um acesso às unidades E/S.

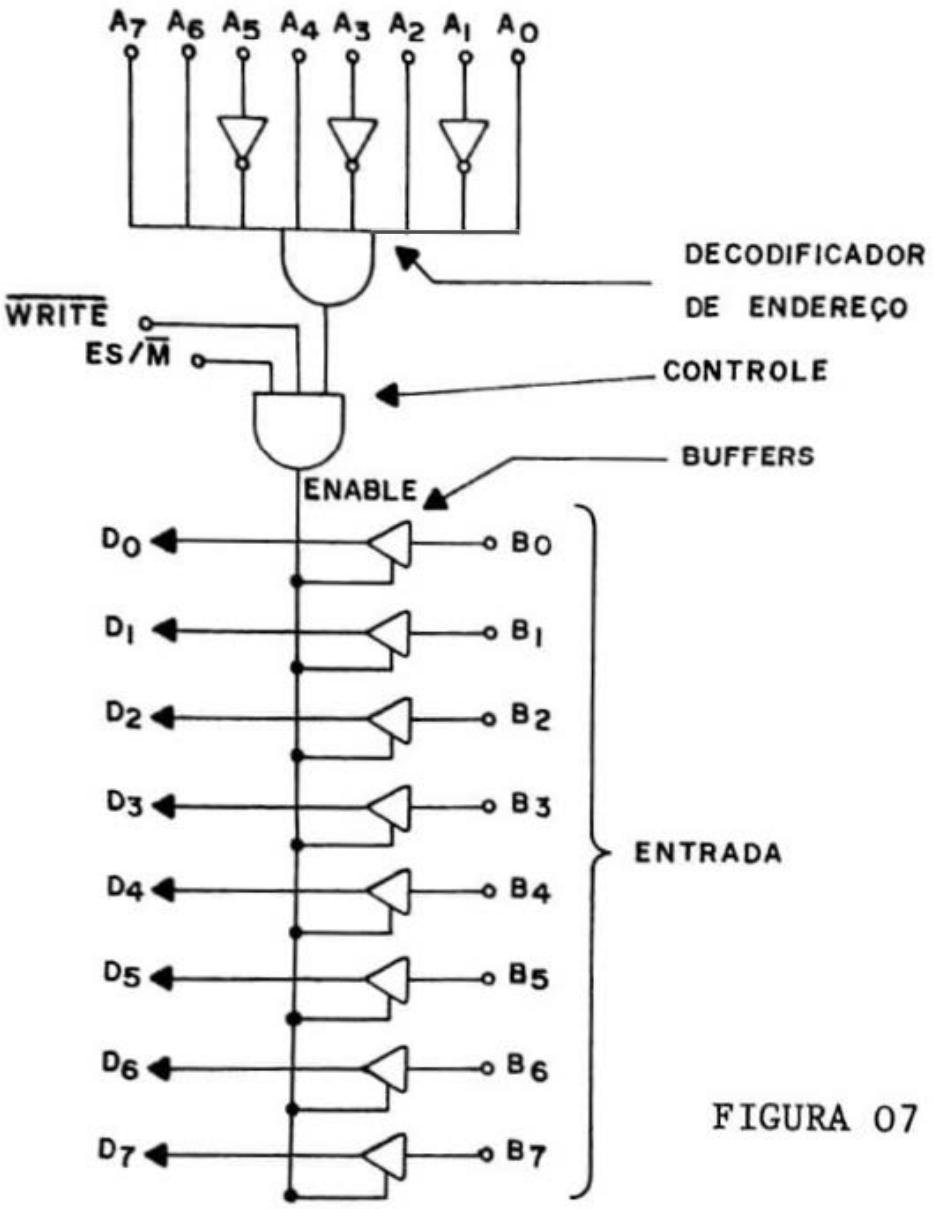


FIGURA 07

A técnica por endereçamento próprio é conhecida ainda por "standard I/O" (E/S estander), "Dedicated I/O" (E/S decodificada) e "I/O Mapped I/O" (E/S mapiada E/S).

As unidades E/S, apresentadas até o momento, possuem as linhas de saídas distintas das linhas de entrada, no entanto, já dispomos de unidades E/S onde as mesmas linhas servem tanto para a entrada como para a saída de informações no microcomputador. Como exemplo, podemos citar as linhas de direção de entrada/saída programável, em que cada bit, ou mesmo grupos de bits, pode ser programado como entrada ou saída.

Existem unidades E/S mais complexas que as apresentadas até aqui, algumas com a complexidade equivalente à do próprio microcomputador.

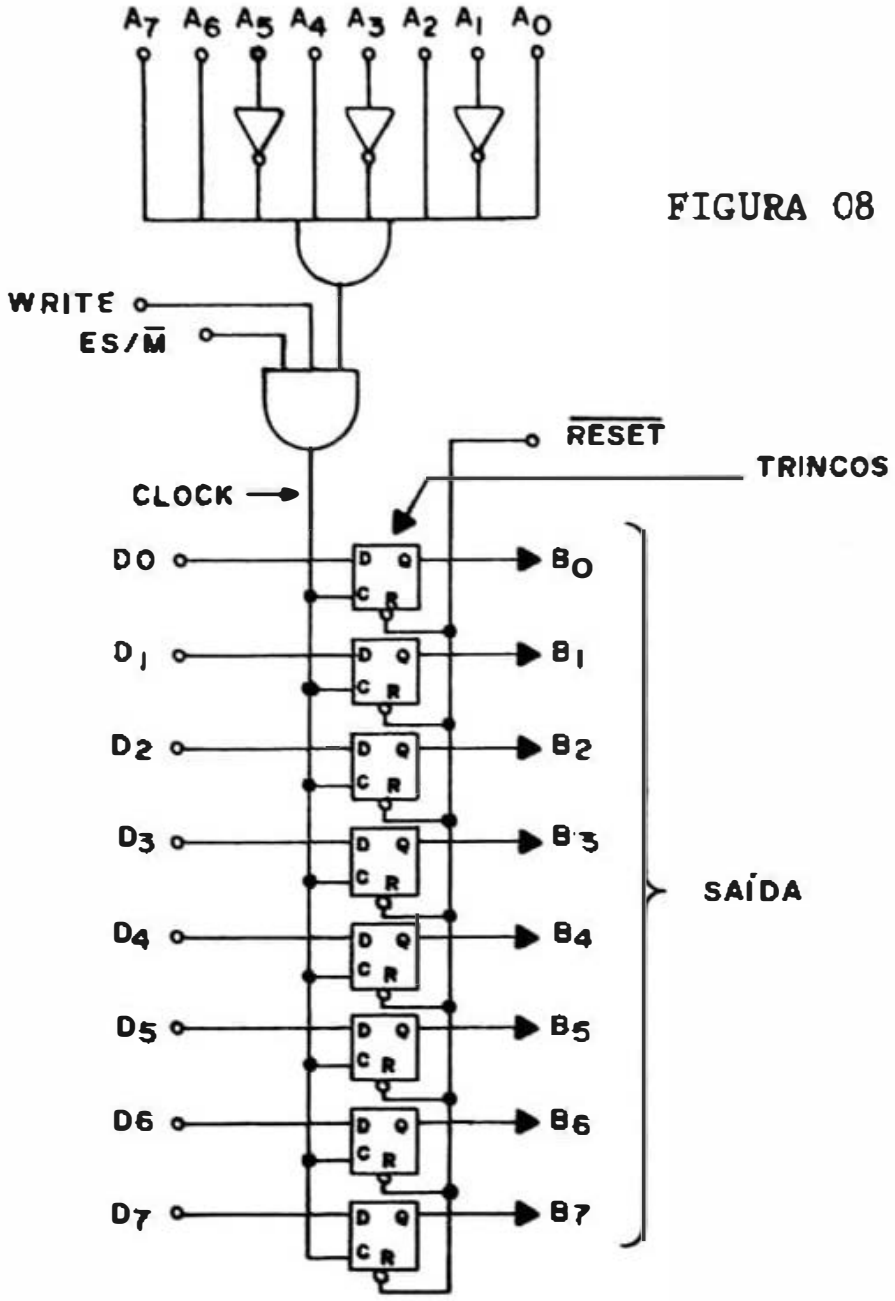


FIGURA 08

COMENTÁRIOS FINAIS:

Tendo como objetivo mostrar a classificação dos computadores, bem como a organização básica dos mesmos (veja figura 1) ressaltando os aspectos construtivos, temos a dizer que as informações prestadas nesta lição são apenas de caráter introdutório, uma vez que esses assuntos serão abordados mais detalhadamente nas próximas lições.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-2

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Na primeira geração, os computadores caracterizavam-se por utilizarem:

a) circuitos integrados;
b) transistores;
c) válvula a vácuo;
d) N.R.A.

2) Com a substituição da válvula a vácuo pelo transistor, teve início a:

a) segunda geração dos computadores;
b) primeira geração dos computadores;
c) terceira geração dos computadores;
d) quarta geração dos computadores;

3) Os computadores da terceira geração caracterizam-se:

a) por utilizarem circuitos integrados;
b) pela execução simultânea de programas, através da evolução do sistema operacional;
c) pela introdução da multiprogramação;
d) os ítens a, b e c estão corretos.

4) A introdução do microprocessador permitiu a construção do:

a) computador de segunda geração; — 4ª Geração
b) computador pessoal ou doméstico;
c) os itens a e b estão corretos;
d) N.R.A.

5) As linguagens Basic, Algol e PL/1 surgiram juntamente com:

a) a quarta geração;
b) a terceira geração;
c) a primeira geração;
d) N.R.A.

6) Uma das características dos computadores analógicos é:

a) utilizar válvulas a vácuo;
b) a sua facilidade em simular processos reais;
c) os itens a e b estão corretos;
d) N.R.A.

7) Os computadores digitais caracterizam-se por:

a) manipularem magnetudes variáveis;
b) manipularem mais de dois níveis de magnetude;
c) manipularem apenas dois níveis de magnetude;
d) N.R.A.

8) Um computador científico possui:

a) pequeno volume de dados (entrada);
b) pequeno volume de resultados (saída);
c) grande volume de cálculos de natureza aritmética;
d) as respostas a, b e c estão corretas.

9) Todo computador digital possui:

a) uma unidade central de processamento;
b) unidades de entradas e saídas (E/S);
c) os itens a e b estão corretos;
d) N.R.A.

10) As memórias utilizadas em microcomputadores são, em geral, de:

a) semicondutores e de acesso aleatório;
b) núcleos de ferrite e de acesso aleatório;
c) os itens a e b estão corretos;
d) N.R.A.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-3

MEMÓRIAS I

INTRODUÇÃO:

Genericamente, podemos dizer que os computadores digitais são constituídos de dois tipos básicos de componentes: os elementos lógicos e os elementos de memória. Entende-se por elementos de memória todo o dispositivo eletrônico capaz de armazenar informações e, por elementos lógicos, todos os dispositivos de comutação.

Os comandos, números ou símbolos (também chamados de palavra) necessários para um processamento são armazenados na memória, cada qual com um certo endereço. O endereço, por sua vez, identifica a palavra armazenada e determina uma posição física definida dentro do dispositivo de memória. Praticamente a totalidade das memórias empregam o armazenamento binário, onde a informação mais elementar é o bit, cujo suporte físico (posição que o bit está ocupando no interior do dispositivo de memória) é denominado "ponto de memória".

O ARMAZENAMENTO EM UM SISTEMA DE COMPUTAÇÃO:

Na realidade, a unidade central de processamento (UCP)

retira e executa instruções da memória principal. A figura 1 nos mostra um diagrama de blocos do armazenamento num sistema de computação.

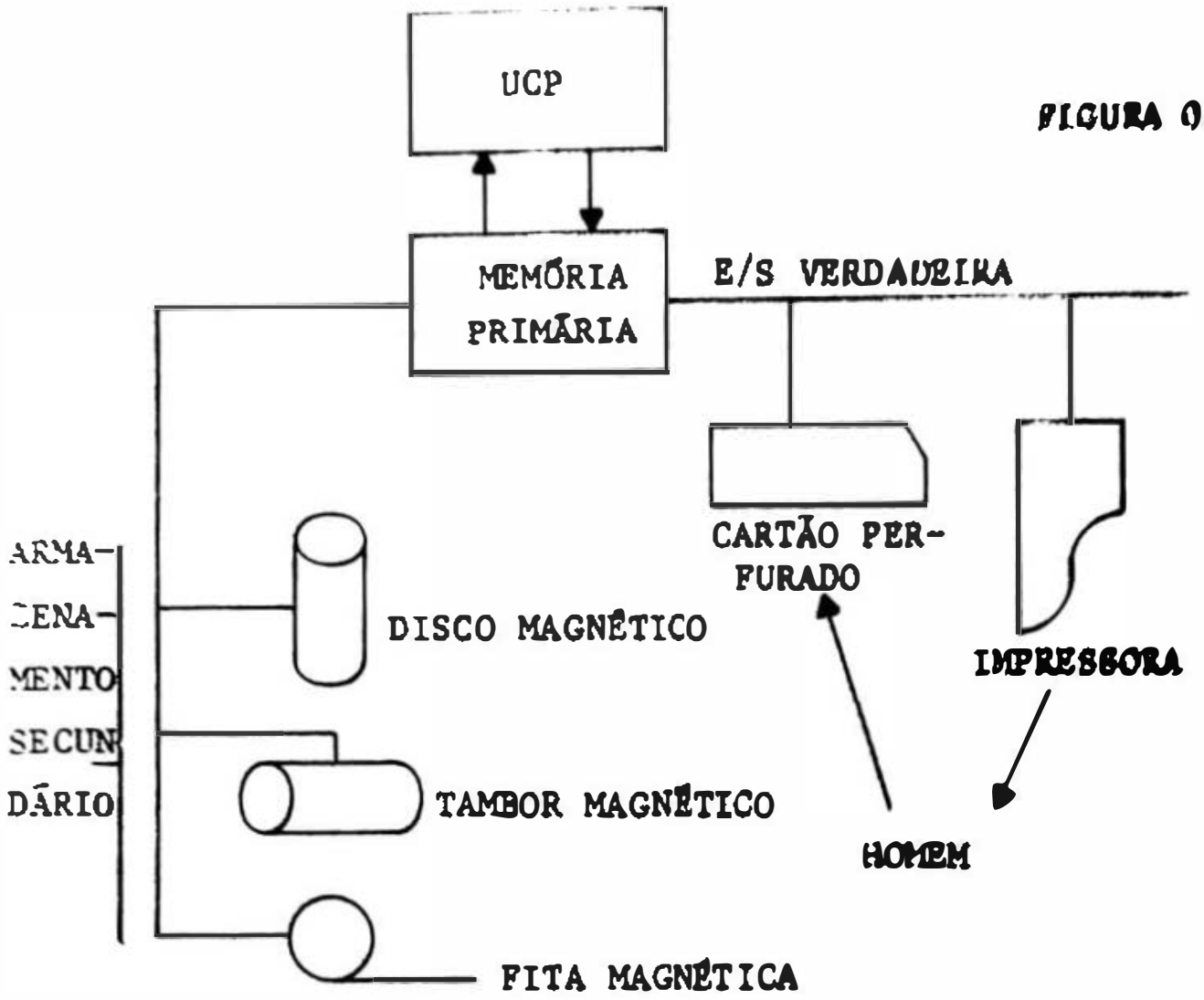


FIGURA 01

A maioria dos computadores da segunda e terceira gerações utilizam-se deste sistema de armazenamento.

Podemos verificar, através da figura 1, que as entradas e saídas verdadeiras passam pela memória principal. Também, antes de serem utilizadas, informações (tanto dados como programas) residentes no armazenamento secundário, são transferidas à memória principal. Com base nessas observações podemos considerar a memória principal (primária) como sendo o ponto focal do sistema. O armazenamento pode ser visto como uma hierarquia onde, na parte inferior ficam as "bibliotecas de fitas" (armazenamento secundário) e, na parte superior, os registradores centrais da UCP. As informações passam para cima para serem processadas. Essa hierarquia possui duas barreiras

naturais: a primeira é a interface entre o homem e o sistema, enquanto que a segunda é a interface entre a memória primária e o armazenamento secundário.

O armazenamento secundário é composto fisicamente, de dispositivos que, sob o controle do sistema, podem fornecer ou aceitar dados da memória primária.

DEFINIÇÃO DE MEMÓRIA PRIMÁRIA:

Trata-se, como já dissemos, do ponto focal do sistema, uma vez que é a memória primária que armazena os programas que dirigem o sistema, pois, sem um programa, o sistema nada faria. Também as características da memória primária, o tempo de ciclo e a capacidade são parâmetros importantes na performace do sistema inteiro.

Na grande maioria dos sistemas, é comum as memórias primária e principal se confundirem. A memória primária é aquela "endereçada pela unidade central de processamento (UCP)", enquanto que a principal, que está contida na primária, faz parte da memória primária de acesso direto e rápido.

Como já foi dito anteriormente, a unidade básica de informação é o bit. Uma palavra é definida como sendo um conjunto de bits. Assim, a palavra da memória principal é geralmente a quantidade de bits retirada por acesso. O acesso a uma palavra armazenada na memória principal é efetuada através do seu endereço, sendo que o endereço nada mais é do que a localização da informação.

Devemos tomar muito cuidado para não confundirmos palavra de informação com endereço da informação, uma vez que tanto a palavra como o endereço são formados por conjuntos de bits.

A memória principal, utilizada nos computadores da primeira geração era do tipo "dinâmica", porém, hoje em dia, a memória principal é, geralmente, do tipo estática.

CLASSIFICAÇÃO:

Tendo em vista que diversos fenômenos físicos podem ser utilizados para a obtenção do "ponto de memória" (o que corresponde a diversas tecnologias) distinguimos três grandes classes de memórias, a saber:

1º) Memórias Estáticas:

Essa classe pode ser dividida em dois grupos: as memórias cujo acesso é feito através de impulsos elétricos, como por exemplo os biestáveis de semicondutores, os núcleos de ferrite, as memórias de película magnética delgadas na unidade central e as memórias cujo acesso é feito através de raios luminosos ou eletrônicos, de possível utilização em um futuro bem próximo.

As memórias estáticas possuem uma forma de trabalho pela qual a memória não necessita de pulsos de clock constantes para manter a informação e não se faz necessário recirculá-la. Esta forma de trabalho vem constituir-se numa das principais características desta classe de memórias.

2º) Memórias de Propagação:

Neste tipo de memória as descontinuidades físicas se propagam de maneira cíclica. Como exemplo, podemos citar as vibrações eletromagnéticas em linhas de retardo (é um tipo de memória). Podemos citar também o deslocamento de bolhas magnéticas sobre o efeito de um campo rotatório. Fenômeno este que já esta sendo utilizado para a fabricação de memórias.

3º) Memórias Dinâmicas:

Esta classe de memória utiliza uma superfície magnética, a qual passa, permanentemente, entre umas cabeças de leitura e escrita, de tal modo a poder registrar a informação sobre a mesma.

A forma de trabalho, utilizada por essa classe de memórias, é contrária à forma de trabalho, utilizada pelas memórias estáticas, pois, neste caso, a memória necessita de pulsos de clock constantes, e a informação precisa ser recirculada para que não se perca.

Esse tipo de memória (memória dinâmica) é utilizada, na maioria das vezes, como "shift register", nos quais as informações ficam sempre girando em torno de si mesmas.

Como exemplos de memórias pertencentes à esta classe, podemos citar as atuais memórias auxiliares: fitas magnéticas, discos magnéticos, tambores magnéticos e lâminas magnéticas.

Os sistemas de fitas magnéticas alcançaram um alto grau de desenvolvimento: são capazes de passar a fita pela cabeça da máquina a uma velocidade de mais de 100 polegadas por segundo e de dar a partida ou determinar a parada da fita em poucos milissegundos.

Já, nos sistemas de discos e tambores magnéticos, estes giram a uma velocidade em torno de 1000 a 2000 polegadas por segundo. Com o avanço tecnológico, uma memória de tambor moderna tem uma capacidade de 262000 palavras e gira a 7200 rotações por minuto; tem um tempo de acesso direto em torno de quatro (4) milissegundos e a taxa de transferência de informação é de 11.200.000 bits por segundo.

CARACTERÍSTICAS DAS MEMÓRIAS:

Quando tivermos que optar por este ou aquele tipo de memória, devemos sempre ter em mente se as características da memória pela qual optamos é adequada, ou seja, é compatível, tanto técnica como economicamente, com o sistema que se está trabalhando. Vejamos então algumas das principais características das memórias.

A) Volatilidade:

Dizemos que uma memória é volátil quando a informação armazenada em seu interior é perdida, devido à falta de alimentação (energia elétrica). Já, uma memória não-volátil se caracteriza pelo fato de não perder a informação armazenada - quando, por exemplo, ocorrer uma falha no sistema de alimentação da mesma.

Como exemplos de memórias voláteis podemos citar as memórias constituídas por biestáveis eletrônicos, as memórias monolíticas, etc... Como exemplo de memórias não voláteis podemos citar as memórias de bolhas magnéticas e as atuais memórias auxiliares (fitas, discos e tambores magnéticos).

B) Leitura e escrita:

Trata-se das operações básicas inerentes às memórias. Realiza-se uma "escrita" quando se registram algumas informações na memória e "leitura" quando se extraem informações previamente registradas. A leitura pode ser destrutiva(neste caso a

informação lida é apagada da memória) ou não destrutiva. A escrita, por sua vez, pode exigir ou não um apagamento prévio da memória. As memórias feitas de elementos semicondutores são de leitura não destrutiva e não exigem, forçosamente, um apagamento prévio para a realização da escrita. Entretanto, nas memórias de núcleo de ferrite, a leitura é destrutiva e a escrita exige um apagamento prévio da memória.

C) Direcionamento:

Para localizar uma informação no interior de uma memória, faz-se necessário saber onde esta informação foi colocada anteriormente. Esta é a função do direcionamento.

Nas memórias em que o "ponto de memória" está perfeitamente individualizado, o alocamento das informações e, por conseguinte, sua direção exata, preexiste a toda operação de armazenamento, ou seja, toda a informação, ao ser armazenada, recebe uma certa direção. Como exemplo podemos citar as memórias de núcleo de ferrite, onde cada célula possue sua direção. O mesmo já não ocorre em memórias do tipo fita magnética, pois o alocamento das informações depende da operação de registro das mesmas. Uma informação não é individualmente direcionável e não existe relação entre a direção e o alocamento. As informações se agrupam por blocos de registros.

A direção da informação consta de dois elementos: o primeiro elemento é o nome do registro (ou indicativo) e o segundo elemento é o número de ordem da informação dentro do registro. De fato, podemos considerar que se tem acesso tão somente ao conjunto de registros.

D) Modo de acesso:

A maioria das memórias recai em uma das três categorias de acesso: direta, periódica ou seqüencial.

Nas memórias de acesso direto, o tempo de acesso independe da seqüência na qual as palavras são introduzidas ou da qual são extraídas. Como exemplo de memórias de acesso direto podemos citar as memórias de núcleo magnético (ou anéis). As memórias com curto tempo de acesso direto são as mais requisitadas, porém custam mais por bit de capacidade de armazenamento.

Um exemplo de dispositivo de memória, que fornece acesso periódico, é o tambor magnético, no qual a informação é registrada na circunferência de um cilindro que gira a uma velocidade constante. Desta forma, após cada período de revolução do tambor, a informação pode ser extraída da memória. Assim, palavras seqüencialmente localizadas podem ser lidas muito rapidamente enquanto passam pelo sensor. O tempo de acesso máximo é igual a um período de revolução do tambor, e o tempo médio de acesso a palavras selecionadas ao acaso é igual à metade do período de revolução do tambor.

Nas memórias de acesso seqüêncial, para se ter acesso a uma determinada informação, é necessário localizar os nomes dos registros até se encontrar a informação desejada. Como exemplo, podemos citar as memórias de fitas magnéticas, nas quais, para se ter acesso a uma determinada informação, é necessário desenrolar a fita, fazendo com que esta passe pela cabeça da leitora até que se encontre a informação desejada.

Apenas a título de informação, o acesso a certos discos com cabeças móveis é do tipo direto, se considerarmos que as cabeças da leitora têm acesso sobre todas as pistas do disco e,do tipo seqüencial no interior da pista. (Veja figura 2).

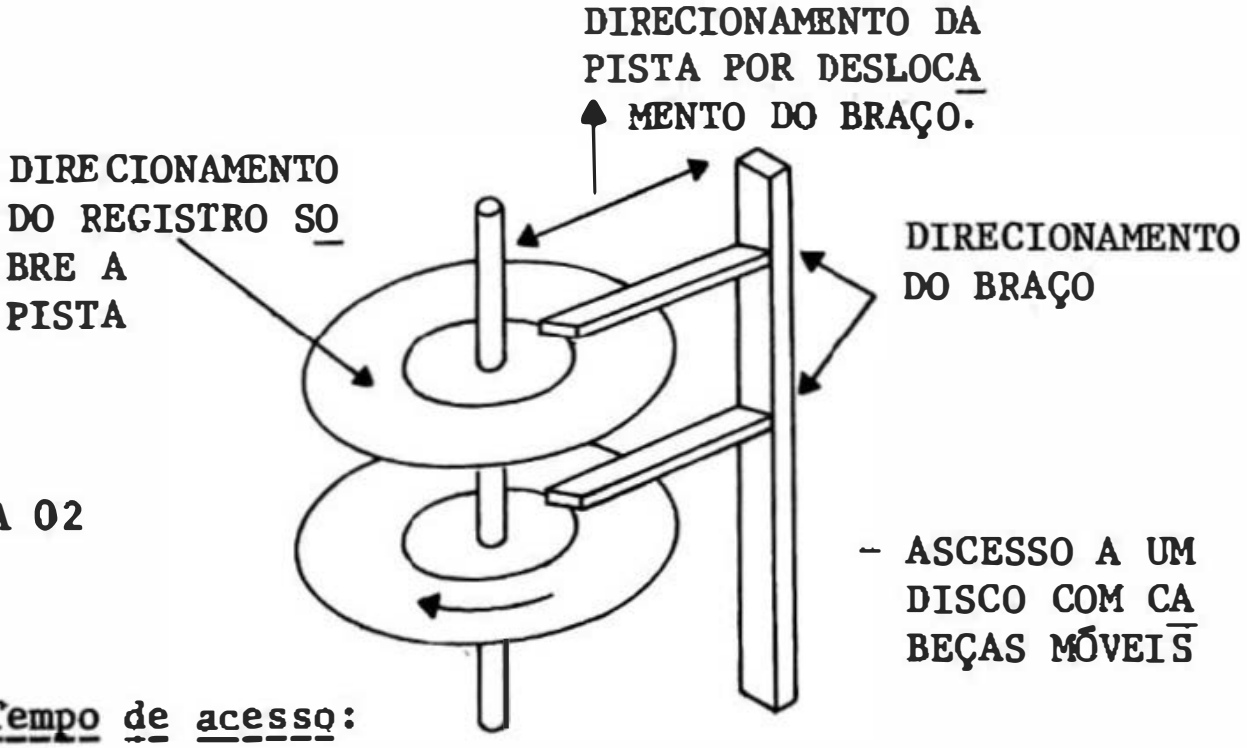


FIGURA 02

E) Tempo de acesso:

Tempo de acesso é o tempo que transcorre entre o instante em que se inicia uma operação de leitura na memória e o instante em que se dispõe da primeira informação que se procura. Para as memórias estáticas, o tempo de acesso, geralmente, independe da localização (aloçação) das informações. Para

calcular o tempo de acesso nas memórias magnéticas dinâmicas, podemos desprezar o tempo de duração das comutações eletrôni cas equivalentes aos tempos de posicionamentos. É o caso, por exemplo, do acesso em um tambor magnético, possuidor de uma cabeça (de leitura) por pista, onde o tempo de acesso depende das posições relativas da informação que se busca e da infor mação situada no momento da leitura ou da escrita.

A noção de tempo de acesso não tem muito sentido quando se trata de fita magnética, pois este pode chegar a ser de vá rios minutos.

F) Caudal:

Dá-se o nome de Caudal ao número máximo de informações lidas ou escritas por unidade de tempo. Geralmente o caudal é expresso em Kilo-informações ou Mega-informações por segundo.

O caudal de um tambor magnético, por exemplo, é igual ao número de palavras por pista, dividido pelo tempo de rotação. A duração de uma operação de entrada/saída sobre um tambor magnético é igual à soma do tempo de acesso e do tempo real de transferência, sendo este último (tempo real de transferên cia) inversamente proporcional ao caudal.

G) Capacidade:

A capacidade de uma memória pode ser entendida como sen do a quantidade máxima de informações que a memória pode arma zenar. Como sabemos, todas as memórias possuem uma certa quan tidade de bits, nos quais irão ser gravadas e lidas as infor- mações. Essa quantidade de bits é que define a capacidade da memória, a qual pode ser expressa em bits ou bytes.

Um bit é o equivalente a um mínimo elemento dentro da me mória (corresponde a um ponto de memória), capaz de armazenar informações.

O byte, por sua vez, é definido como sendo uma certa quantidade de bits. Um byte varia de memória para memória e até hoje ainda não se chegou a um consenso sobre quantos bits tem um byte. Sabe-se, no entanto, que existem bytes de 4, 8, 16, 32, ... etc. bits.

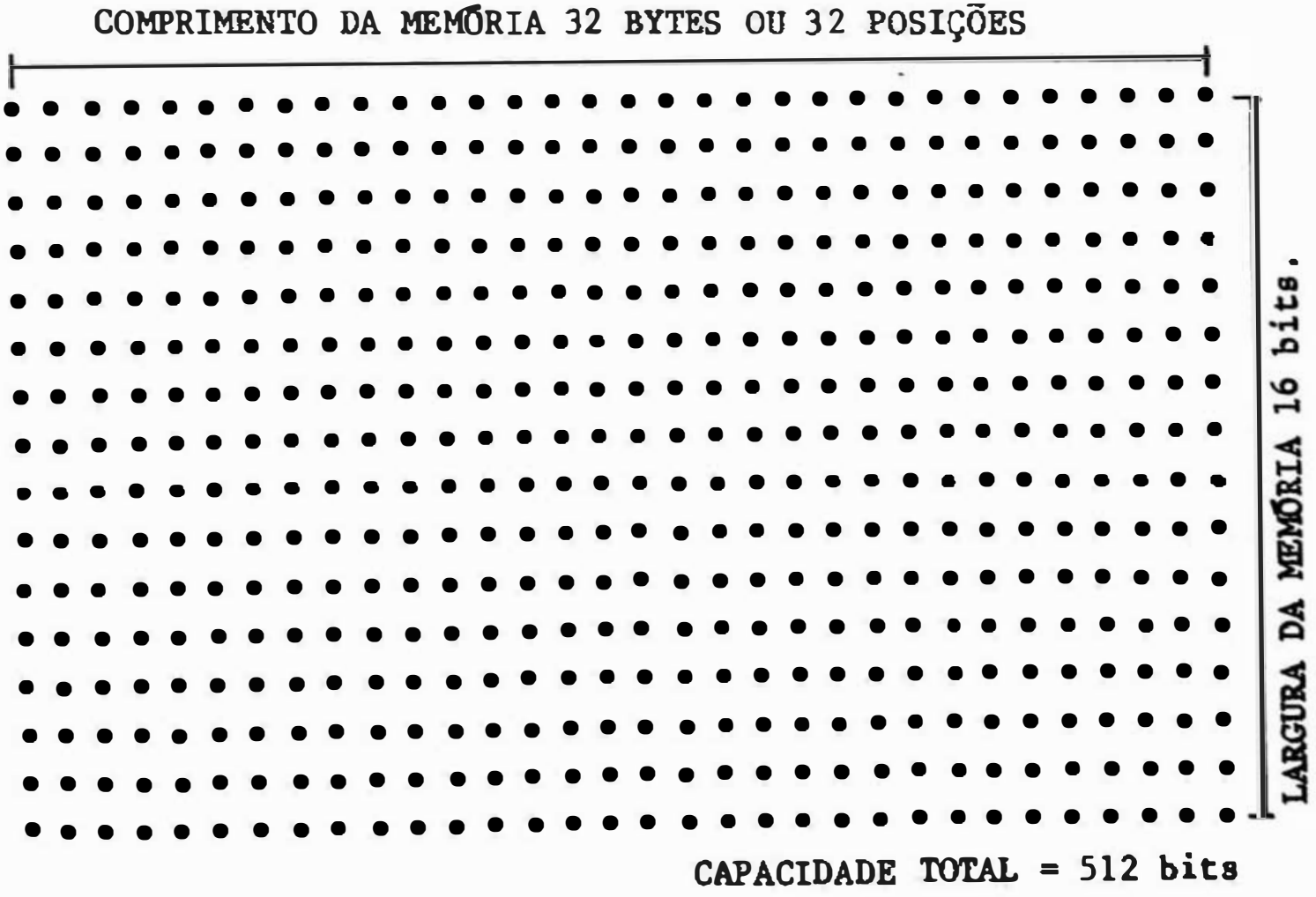
A tendência atual é estabelecer um byte como sendo igual a 8 bits, devido à grande demanda de microprocessadores que u

tilizam 8 bits como sendo igual a um byte.

Devido à grande diversificação de tipos de memórias, os fabricantes especificam as memórias em número exatos e em números de bits. Normalmente essas especificações são dadas em dois planos, ou seja, na largura e no comprimento das memórias. A largura da memória vem a ser a quantidade de bits necessários para formar 1 byte. Já o comprimento é definido como sendo a quantidade de bytes de uma (1) memória.

A capacidade total de uma memória, em termos de bits, pode ser calculada, multiplicando-se a largura pelo comprimento ou seja, multiplicando-se o número de bits pelo número de bytes. Por exemplo, uma memória que possui 16 bits de largura e 32 bytes de comprimento terá uma capacidade total de 512 bits. A figura 3 exemplifica uma memória com capacidade de 512 bits no total.

FIGURA 03



Na figura 4, mostramos uma outra memória, com a mesma capacidade total (512 bits), porém com largura e comprimento diferentes.

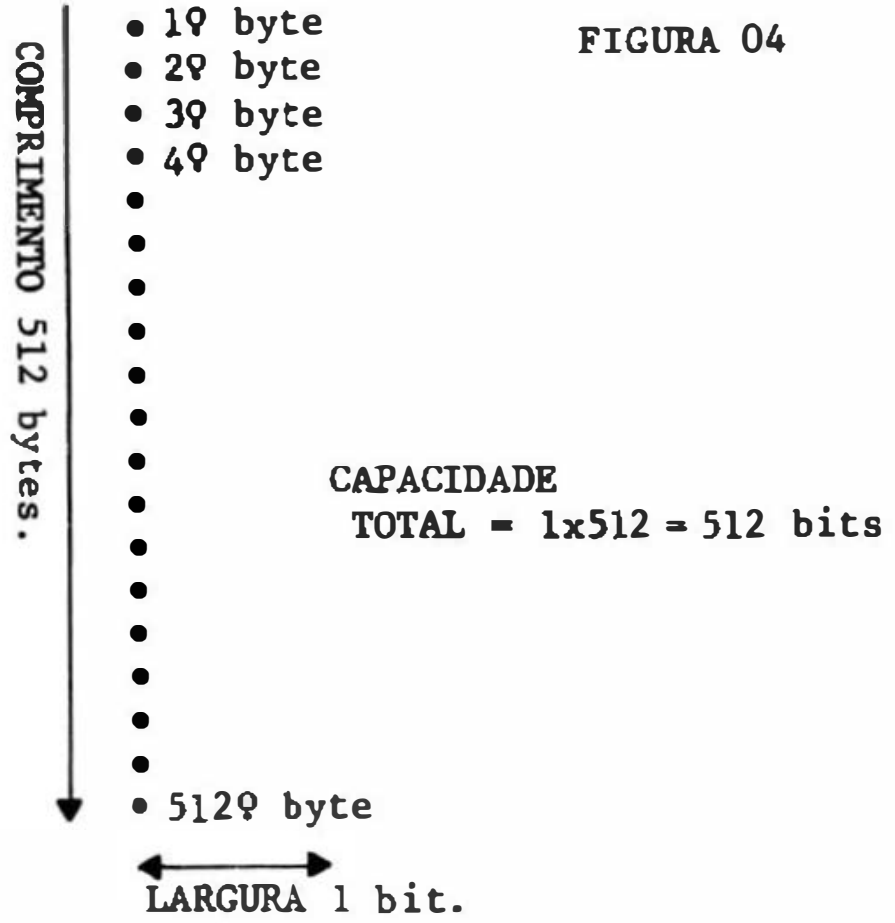


FIGURA 04

Para simplificar, as capacidades totais das memórias são expressas por potenciais de dois (2): 512, 1024, 4096, 8192 bits, etc.. Com base nesse fato, convencionou-se que 2^{10}=1024 unidades de informação é aproximadamente igual a um kilo-in - formação (1K). Dessa forma, uma memória com uma capacidade to tal de 8192 bits será identificada como sendo uma memória de 8K bits. Seguindo esse mesmo raciocínio, podemos concluir que uma memória de 16K possui, nada mais nada menos, que 16.384 bits.

ENDEREÇAMENTO:

Endereço de uma memória nada mais é do que a localização de cada byte ao longo da mesma. Logo, em uma memória de 10 bytes, existem 10 endereços, um para cada byte.

O conteúdo da memória é a informação contida em cada byte. Muitas vezes, a informação contida num byte é maior ou menor, de onde decorre o fato de um byte variar a sua largura.

Na figura 3, temos o exemplo de uma memória de 32 bytes x 16 bits, ou seja, para cada byte (1 endereço) podemos gravar um dado diferente e distinto.

O endereçamento das memórias, geralmente, é feito em forma binária, sendo que a quantidade de fios necessária para endereçar uma memória depende do número de bytes nela existente.

Na figura 5 apresentamos uma memória de 1024 bytes (de 0 a 1023) de 4 bits cada, o que nos leva a concluir que a sua capacidade total é de 4096 bits. Para endereçar esta memória, (veja figura 5) precisamos de 10 fios, ou seja: 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 = 1023 = $2^{10} - 1$.

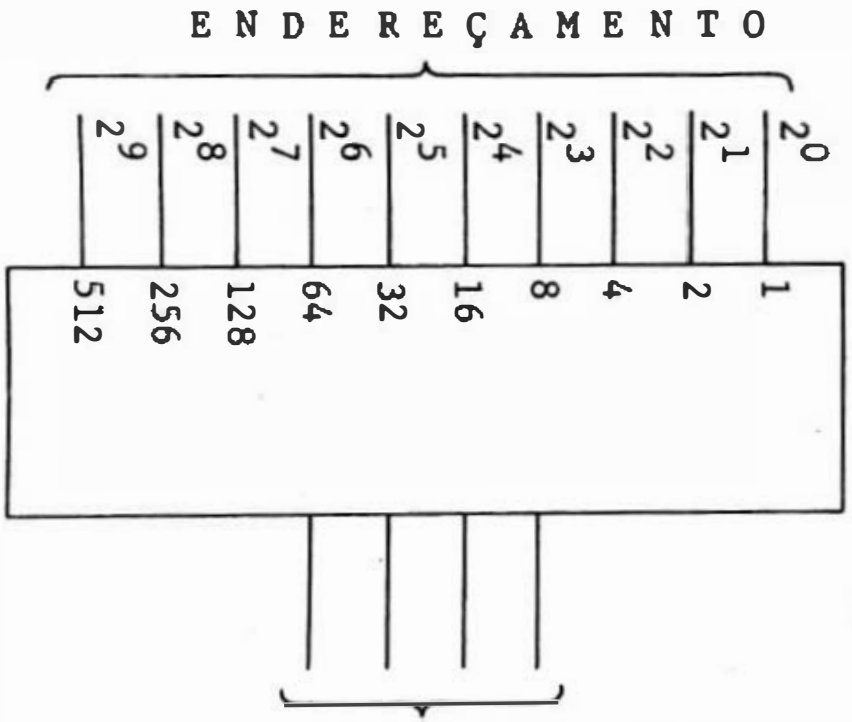


FIGURA 05

Note que na saída dos conteúdos dos endereços, temos apenas 4 linhas, as quais correspondem ao resultado do endereçamento dos 10 fios de entrada (de 2^0 a 2^9) e nele está o conteúdo do endereço selecionado. Para melhor compreensão, imagine uma chave rotativa de 4 polos por 10 posições (veja figura 6).

Observe que, cada vez que mudarmos a posição da chave, estaremos mudando o endereço, ou seja, estaremos endereçando um novo byte e o conteúdo de cada um deles aparece na saída dos 4 polos da chave.

Existem memórias que não possuem o número de bytes exigidos. Por isso, às vezes, somos obrigados a utilizar mais de

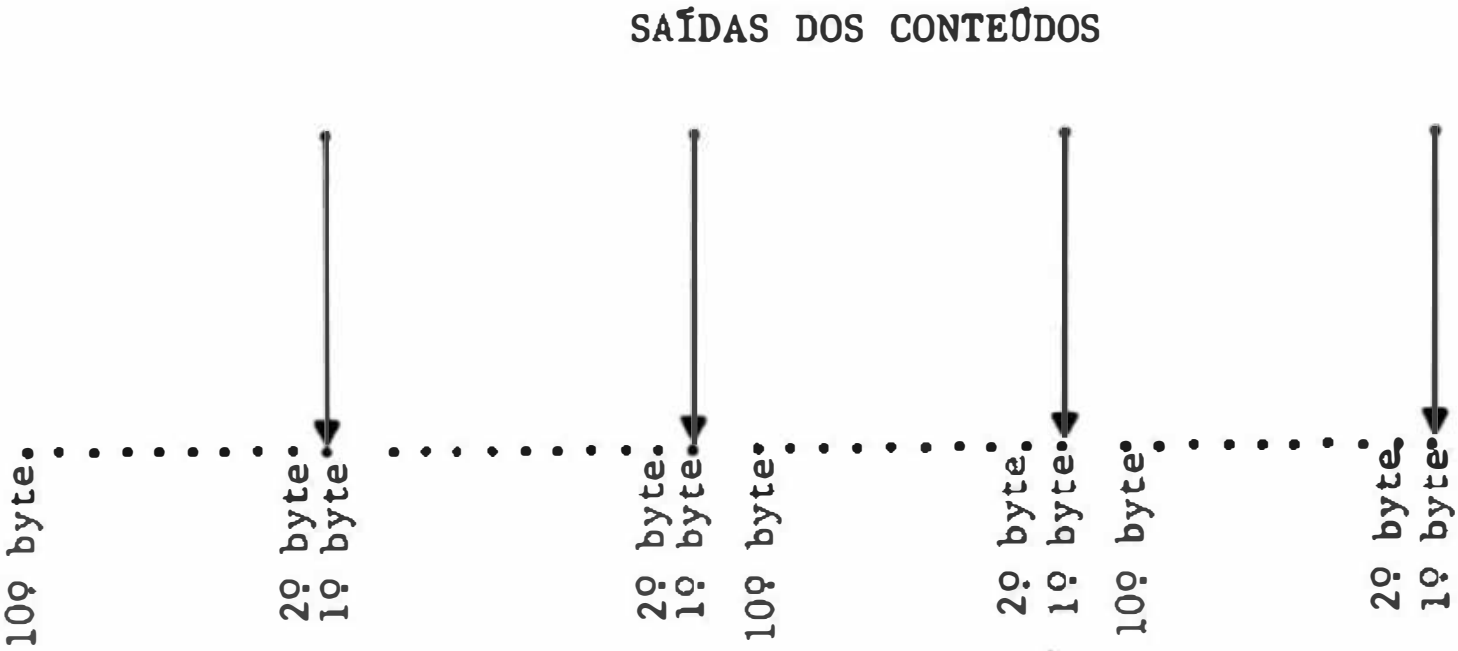


FIGURA 06

uma memória. Na figura 7, temos alguns detalhes de um endereçamento externo (o endereço é dito externo quando endereça vários grupos de circuitos integrados) para duas memórias de 1024 bytes. Neste caso não nos interessa o número de bits por byte, pois o endereçamento é o mesmo para qualquer largura de byte.

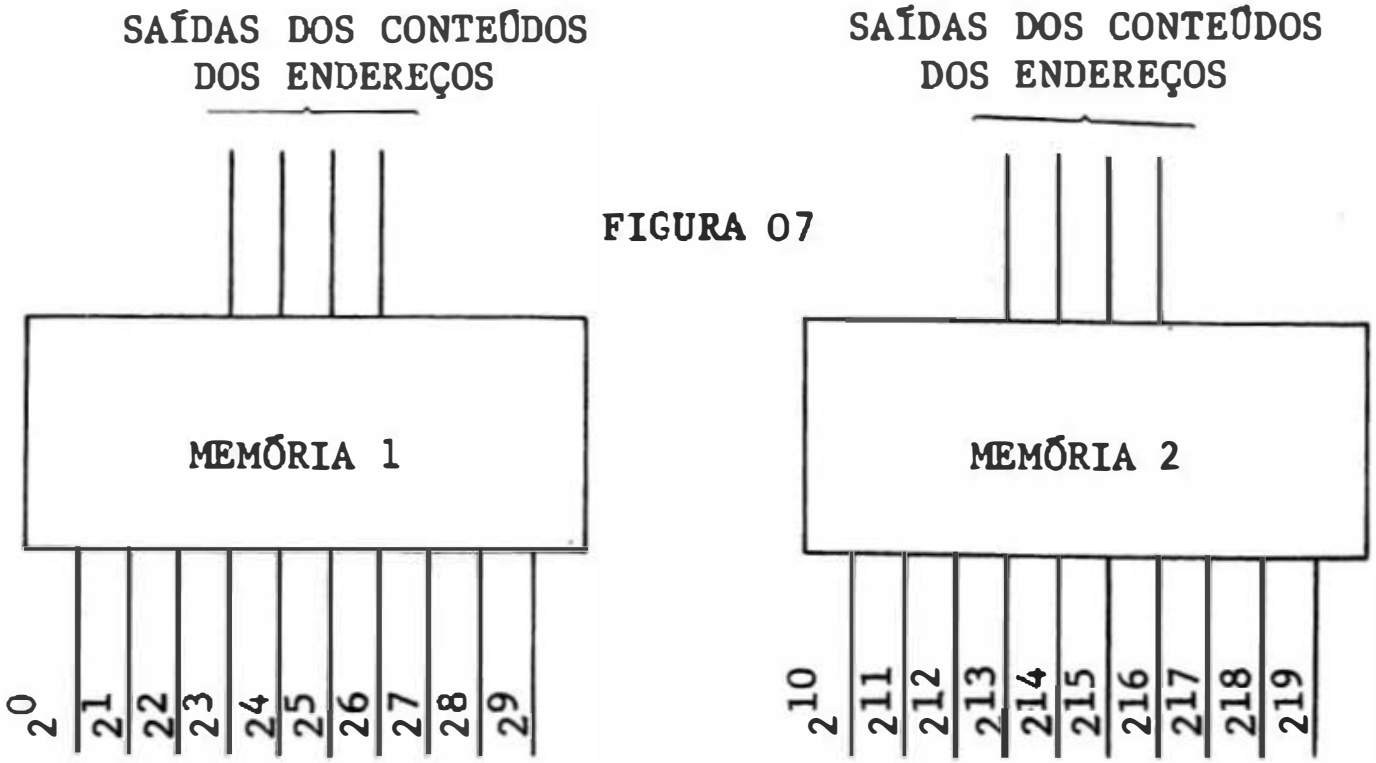


FIGURA 07

Na figura 8, detalhamos o endereçamento das memórias, de maneira mais simples, principalmente no que diz respeito à fiação reduzida.

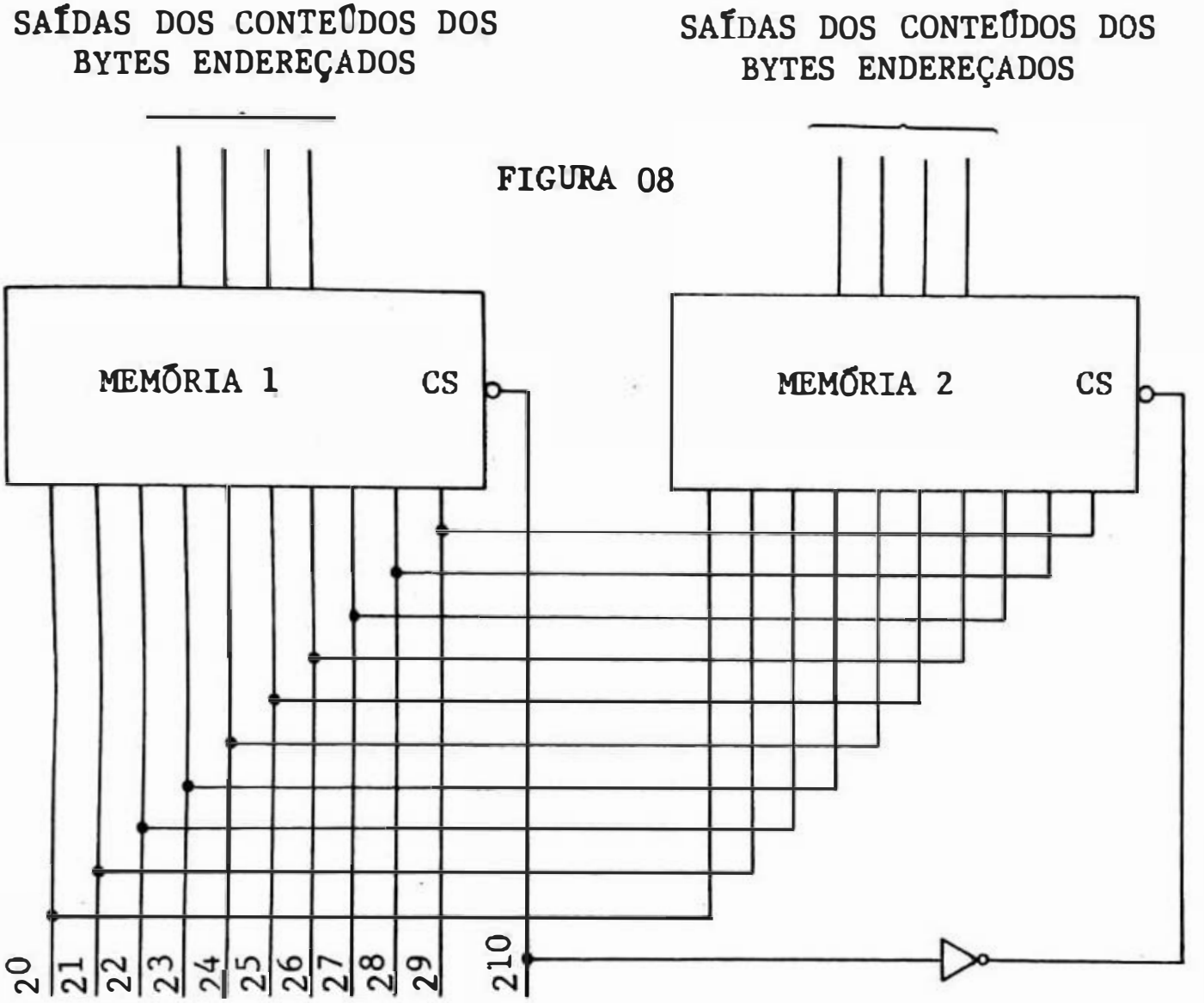


FIGURA 08

Este método de trabalho em paralelo facilita em muito o desenvolvimento de projetos, tanto com relação à economia de material quanto em termos de economia de tempo.

Neste tipo de memória, faz-se necessária a introdução - dos chamados "circuitos de habilitação " - chip select - (CS). As memórias que possuem os circuitos de habilitação (CS) são caracterizadas pela possibilidade de serem totalmente desligadas (desativadas) ou ligadas (ativadas).

Com a opção CS, nós poderemos endereçar, por exemplo, somente uma memória num grupo de várias. No caso apresentado na figura 8, podemos observar que o fio $2^{10} = 1024$, quando em estado alto (nível lógico = 1), desabilita a "memória 1" e habi

lita a"memória 2", através do pino CS.

O pino CS é muitas vezes chamado de pino "enable".

Cabe ressaltar o fato de que a memória está ativa quando a entrada CS estiver válida (nível lógico "1" ou "0", dependendo da lógica adotada), e que a memória está desativada - quando a entrada CS estiver inválida (nível lógico "1" ou "0", dependendo da lógica adotada).

Com base na figura 8, se quisermos endereçar a posição de memória 1500 (byte 1500), não devemos nos preocupar com o número de memórias existentes, pois o fio $2^{10} = 1024$ será automaticamente endereçado através de um circuito decodificador externo que irá habilitar a posição 476 da memória "2", correspondendo à diferença "1500-1024". Os bits correspondentes a 476 são: 4, 8, 16, 64, 128 e 256.

Resumindo, podemos dizer que toda vez que endereçamos a posição 1500, estaremos, na realidade, selecionando a posição 476 da memória "2".

NÍVEIS HIERÁRQUICOS EM UM SISTEMA INFORMÁTICO:

Em qualquer computador, o ideal seria dispormos de uma memória central (ou principal) muito rápida e de capacidade muito grande. Essa solução seria demasiadamente custosa em termos financeiros e tecnicamente irrealizável, em virtude da complexibilidade resultante dos circuitos de seleção. Em decorrência desse fato, estabeleceu-se uma hierarquia das memórias, em todos os níveis: uma memória principal relativamente rápida, porém de capacidade bastante limitada, uma memória auxiliar de capacidade muito maior, porém com tempo de acesso considerável.

Dessa forma, distinguem-se cinco níveis de memórias, levando-se em conta seu uso e localização, bem como a arquitetura da máquina.

A) Memórias-Tampão ou Memórias de Anotações:

Essas memórias, normalmente, são de acesso aleatório (ou direto), caracterizadas por sua elevada eficácia e baixa capacidade. São utilizadas em máquinas (computadores) de terceira geração, desempenhando a função de memória tampão ou lo-

cal na unidade central de processamento (UCP).

Geralmente, são fabricadas com semicondutores, excepcionalmente com películas magnéticas finas. Hoje em dia, essas memórias possuem um tempo de acesso entre 50 e 200 nanossegundos.

B) Memória Principal:

Apenas, complementando o que já foi dito a respeito desse tipo de memória, podemos dizer que geralmente, na prática, são do tipo de acesso aleatório, posto que ela "procura", na unidade central de processamento (UCP), precisamente a informação que necessita. Contudo, na primeira geração, utilizaram-se tambores magnéticos e linhas de magnetoestrição para desempenhar essas funções.

Atualmente, são empregadas as memórias magnéticas estáticas: memórias de núcleos e, menos freqüentemente, memórias de película magnética fina. Também já se começa a utilizar memórias de semicondutores. Para a memória principal, os tempos de acesso oscilam entre 250 nanossegundos e 2 microssegundos para memórias com capacidade de até 256K bytes por módulo.

C) Extensões da memória principal:

De concepção relativamente recente, nessa primeira classe de memória auxiliar excluem-se memórias de capacidade muito grande (vários milhões de bits) com tempo de acesso muito parecido com os das memórias principais.

D) Memórias de Massa:

São memórias de acesso seletivo, caracterizadas por capacidades muito grandes, tempo de acesso considerável e alta velocidade de transferência. Até hoje, esse campo é quase que exclusivo das memórias magnéticas dinâmicas, cujo baixo custo por bit viabiliza capacidades importantes e cujo caudal pode alcançar vários megabits por segundo. Nesta classe de memórias se distinguem os tambores magnéticos e os discos magnéticos com cabeças fixas (tempo de acesso de 5 a 20 milissegundos), os discos magnéticos com cabeças móveis, freqüentemente intercambiáveis, com tempo de acesso de 50 a 100 milissegundos, e por fim, as memórias com lâminas magnéticas, cujo tem-

po de acesso está entre 200 e 600 milissegundos.

E) Memórias Fichários:

São as fitas magnéticas, caracterizadas pelo seu acesso seqüencial (que implica em tempos de acesso de até vários minutos) e por sua intercambialidade, levando a uma capacidade quase infinita de armazenamento. Também podem ser usadas para permutar informações entre computadores. O caudal das fitas magnéticas pode oscilar entre 15 e 300K bits por segundo.

ZONAS DE UTILIZAÇÃO DAS DIFERENTES TECNOLOGIAS DE MEMÓRIA:

O gráfico apresentado na figura 9 nos mostra as zonas (regiões) atuais de utilização das diferentes classes de memórias, levando em consideração as limitações tecnológicas e a questão de preços (custo).

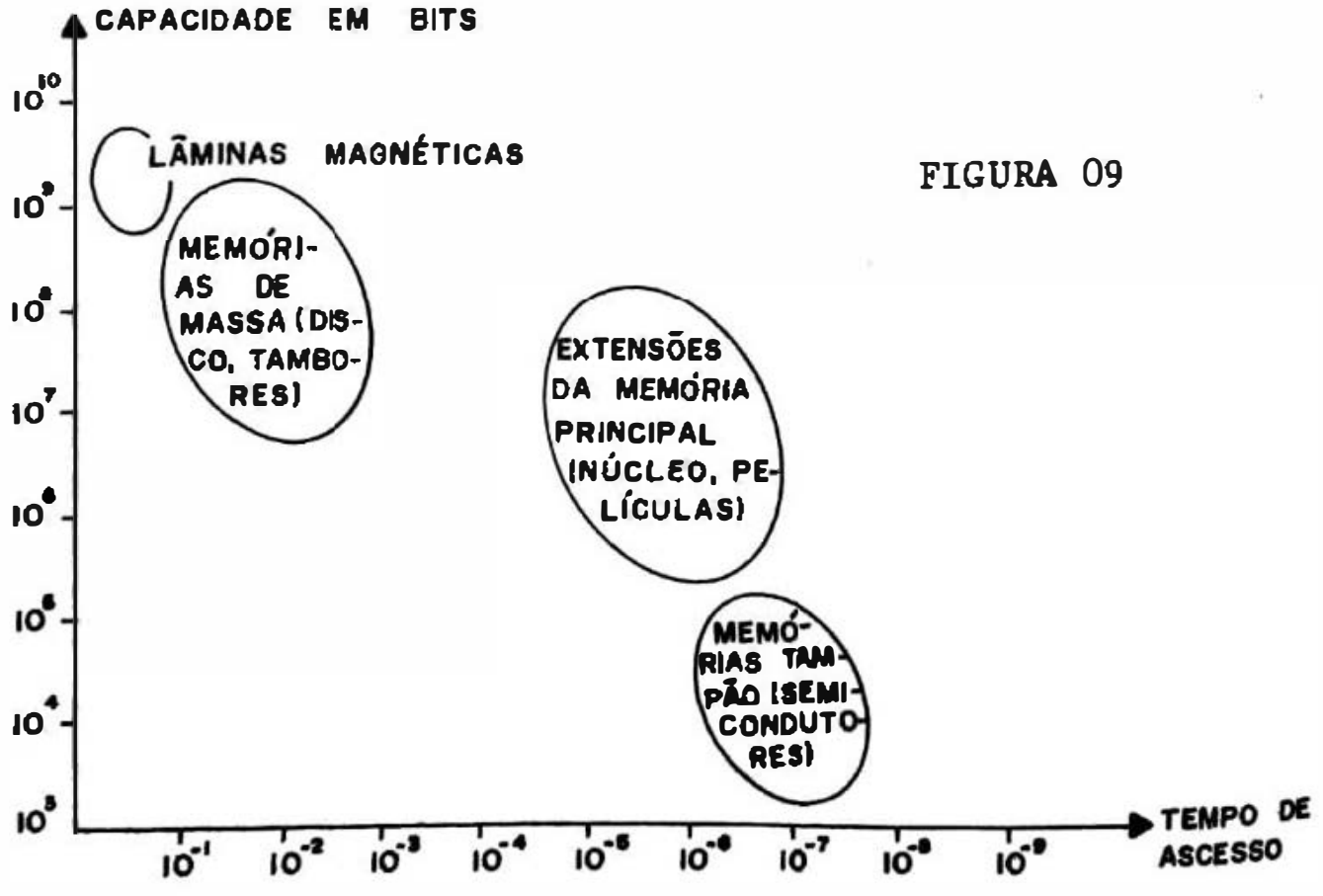


FIGURA 09

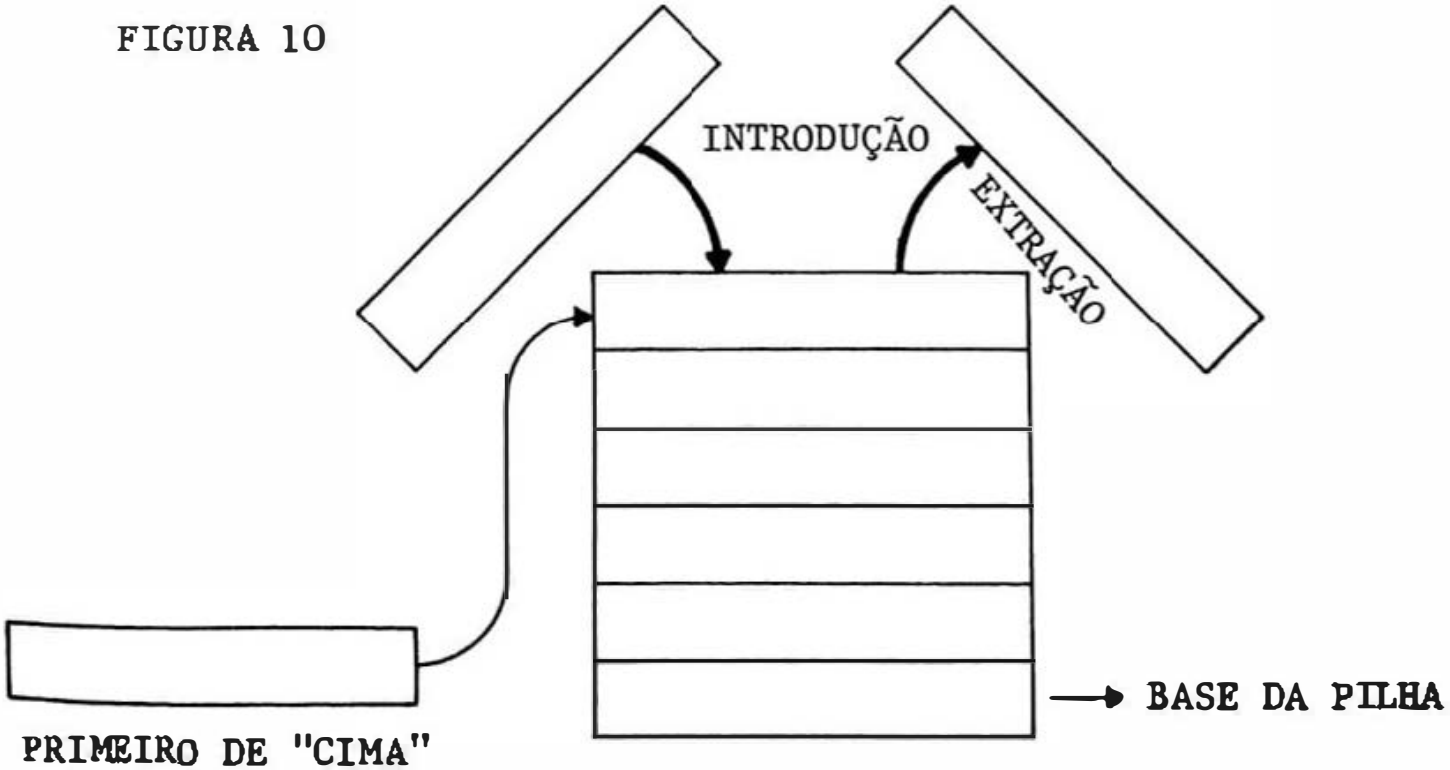
MEMÓRIAS ESPECIAIS:

Dentro desta categoria, incluímos aquelas memórias que, pela peculiaridade de seu funcionamento, estão destinadas a aplicações especiais. Nesta categoria de memória, distinguimos três tipos: as pilhas e as filas de espera, as memórias associativas e as memórias mortas.

A) Pilhas e filas de espera:

As pilhas se caracterizam pelo fato de que certas operações implicam em um deslocamento global de toda a informação contida na memória. Assim, por exemplo, a inserção de uma informação em uma pilha "empurra" o conteúdo da pilha em um "degrau", enquanto que a extração de uma informação "atrai" o conteúdo da pilha em um "degrau". De uma forma bem simplificada, podemos dizer que o mecanismo de funcionamento da pilha nos lembra o de uma pilha de caixas em um depósito, ou seja, tem-se acesso à última informação armazenada, com o esquema "último a entrar, primeiro a sair" (veja figura 10).

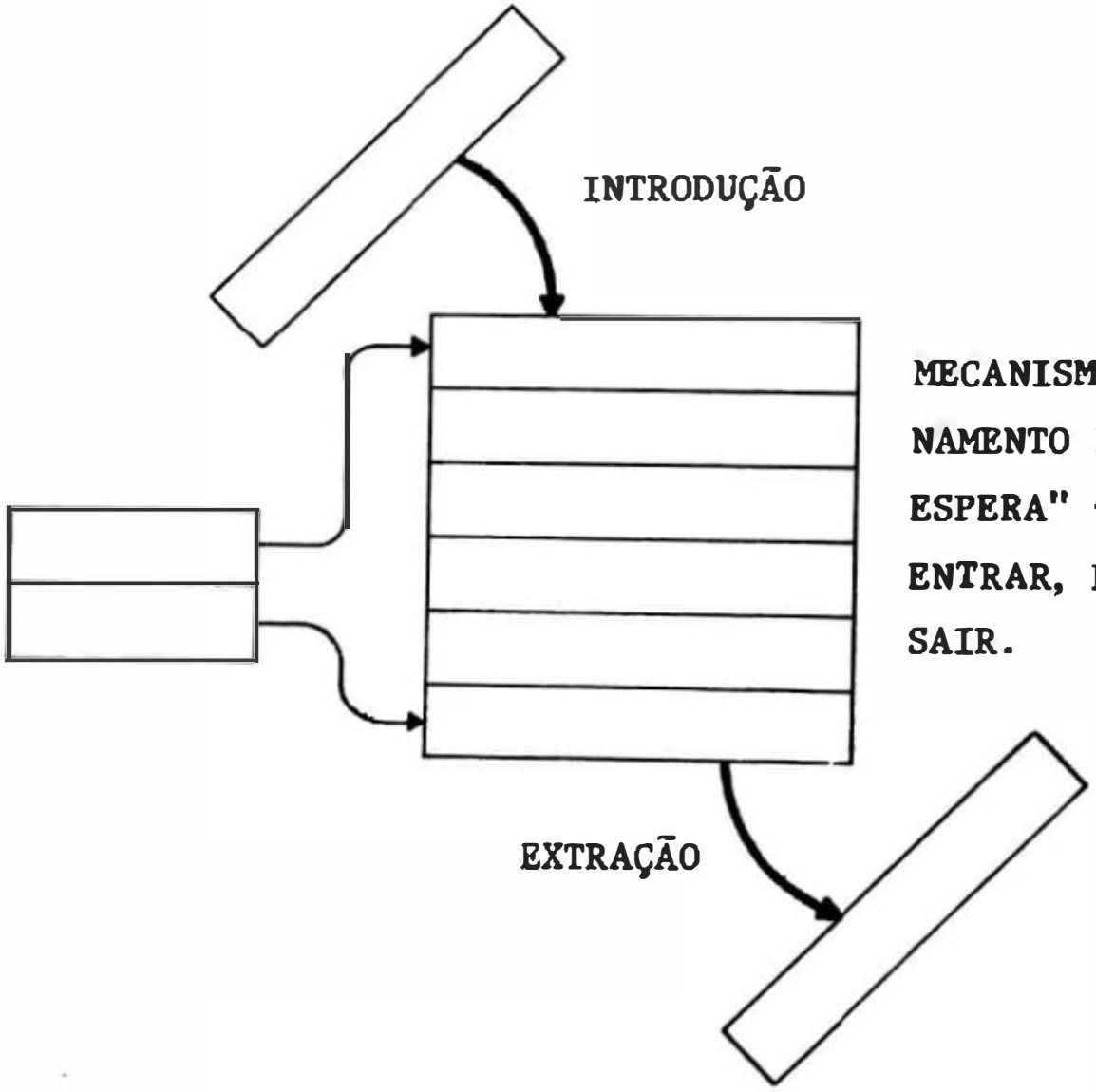
FIGURA 10



MECANISMO DA PILHA "ÚLTIMO A ENTRAR, PRIMEIRO A SAIR".

Tais memórias (pilhas) podem ser concebidas, com pequena capacidade, por meio de lógica celular à base de semicondutores.

Quanto à "fila de espera", esta funciona segundo um princípio muito conhecido popularmente: "primeiro a chegar, primeiro a ser atendido" (veja figura 11). As filas de espera são muito utilizadas para compensar a diferença de caudal de suas "máquinas" entre as quais se estabelece uma comunicação. Em tais casos, estas memórias também são chamadas de memórias-tampão.



MECANISMO DE FUNCIONAMENTO DA "FILA DE ESPERA" - PRIMEIRO A ENTRAR, PRIMEIRO A SAIR.

FIGURA 11

As memórias "Fifo" (First in, First out) são um exemplo de aplicação das memórias tipo fila de espera.

B) Memórias associativas:

Nas memórias de acesso aleatório, ao se proporcionar uma direção, recupera-se a informação armazenada. As memórias associativas ou "memórias direcionáveis pelo conteúdo" funcionam de maneira semelhante. Estas memórias (associativas) respondem à seguinte pergunta: Está contida na memória uma determinada informação, chamada DESCRITOR? E, se estiver contida, em que posição?

Na figura 12 temos um esquema no qual é feita uma comparação entre a memória convencional (no caso de acesso aleatório) e a memória associativa.

QUADRO COMPARATIVO ENTRE MEMÓRIA CONVENCIONAL E MEMÓRIA ASSOCIATIVA

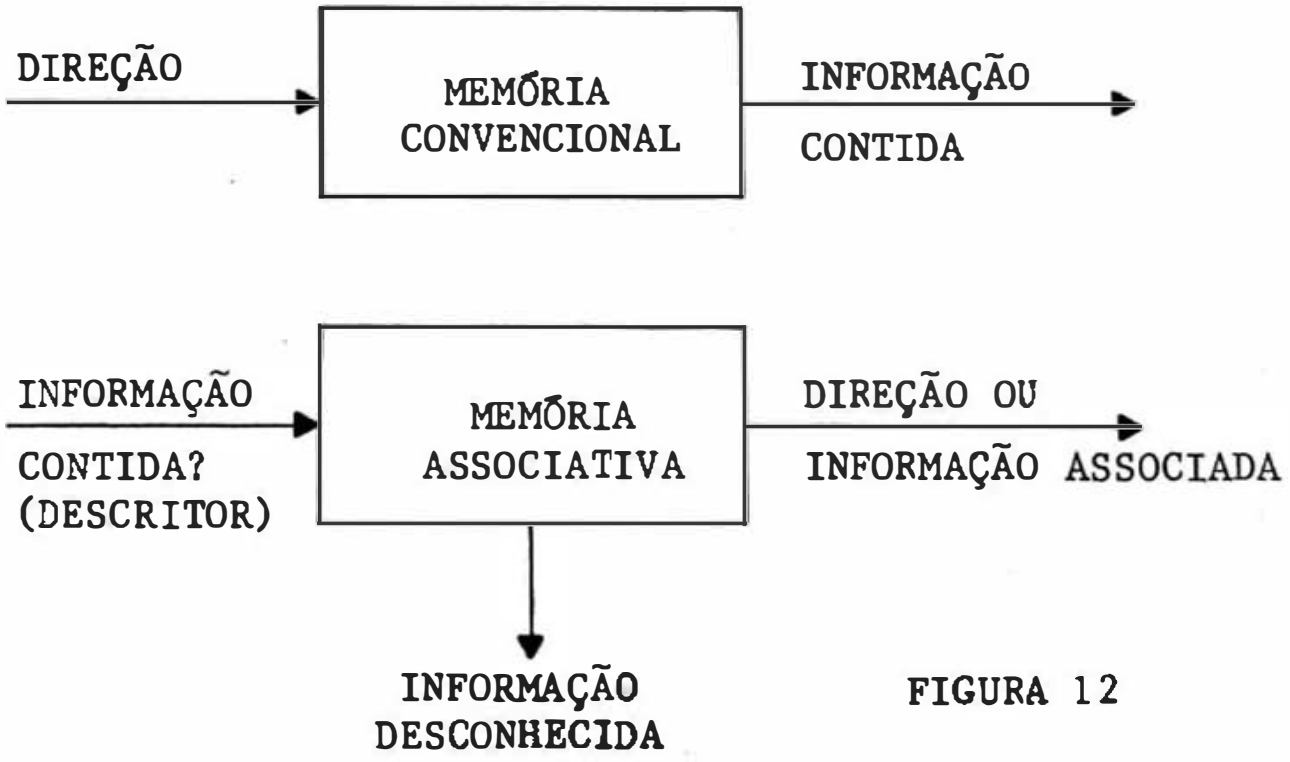


FIGURA 12

C) Memórias mortas:

As memórias mortas ou memórias de somente leitura são memórias que operam unicamente em leitura. As memórias reinscritíveis permitem uma leitura rápida, porém, uma escrita lenta.

Esse tipo de memória é muito utilizada no campo da microcomputação.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-3

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) A unidade básica de informação é o:

   a) bit;

   b) byte;

   c) CS;

   d) N.R.A.

2) A principal característica das memórias estáticas reside no fato:

   a) de necessitarem de pulsos de clock constante para manterem a informação;

b) de não necessitarem de pulsos de clock constantes para manterem a informação;

c) de haver propagação das descontinuidades físicas, de maneira cíclica;

d) N.R.A.

3) Nas memórias dinâmicas:

a) a informação necessita ser recirculada para que não se perca;

b) a informação não necessita ser recirculada;

c) a forma de trabalho é idêntica à forma de trabalho das memórias estáticas;

d) os ítens b e c estão corretos.

4) Como exemplo de memórias dinâmicas podemos citar:

a) fitas magnéticas;

b) discos magnéticos;

c) tambores magnéticos;

d) os ítens a, b e c estão corretos.

5) Uma memória é dita volátil quando:

a) a informação armazenada em seu interior não é perdida, devido às falhas no sistema de alimentação;

b) a informação armazenada em seu interior não é perdida, devido às falhas no sistema de leitura;

c) a informação armazenada em seu interior é perdida, devido às falhas no sistema de alimentação;

d) N.R.A.

6) Quanto ao modo de acesso, as memórias se dividem em:

a) memórias de acesso direto;

b) memórias de acesso periódico;

c) memórias de acesso seqüencial;

d) os ítens a, b e c estão corretos.

7) Como exemplo de dispositivo de memória que fornece acesso periódico podemos citar:

a) as memórias de núcleos magnéticos;

b) as memórias de fitas magnéticas;

c) as memórias do tipo tambor magnético;

d) N.R.A.

8) Qual é a capacidade total de uma memória de 8 bits e 32 bytes?

a) 1024;

b) 2048;

c) 256;

d) 512.

32
x 8
256

9) Memórias fichários são:

a) memórias de acesso seletivo;

b) memórias caracterizadas pelo seu acesso seqüencial;

c) memórias caracterizadas pelo seu acesso direto;

d) NRA.

10) As memórias tipo "pilhas" trabalham segundo o princípio:

a) último a entrar, primeiro a sair;

b) primeiro a entrar primeiro a sair;

c) da arrumação;

d) NRA.

Memoria "fila de espera" = (FIFO)

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-4

MEMÓRIAS II

INTRODUÇÃO:

Procuraremos nesta lição e nas próximas (Memórias III e IV) descrever os diversos tipos de memórias, visando levar até o aluno informações sobre os aspectos construtivos e aplicativos das memórias.

Levando em consideração o fato de que os materiais magnéticos suprem, desta ou daquela forma, os princípais meios de armazenamento nos computadores, começaremos o nosso estudo, descrevendo mais amplamente as memórias magnéticas.

MEMÓRIAS DE NÚCLEOS MAGNÉTICOS:

A tecnologia para a memória principal que prevaleceu na segunda e terceira gerações de computadores foi a memória de núcleos magnéticos ou simplesmente memória de núcleos.

As memórias de núcleos utilizam núcleos toroidais de ferrite (que é uma cerâmica ferro-magnética). Métodos altamente automatizados foram desenvolvidos para formar, impulsionar, testar e montar os núcleos em séries de memória. Nas primeiras memórias de núcleo-magnético, os núcleos tinham um diâmetro externo aproximadamente igual a um "duodécimo" de polegada (1/12 de polegada) e custavam em torno de US$ 1,00 (1 dólar) por bit de capacidade de armazenamento. O tempo de ciclo dessas memórias (tempo mínimo desde o início de um ciclo de acesso até o começo do próximo) estava entre 10 e 20 microssegundos. Com o desenvolvimento da tecnologia, as dimensões dos núcleos diminuíram, o tempo de ciclo foi reduzido e a capacidade máxima aumentou. Na maioria dos computadores contemporâneos, os núcleos têm um diâmetro de um vigésimo de polegada; o tempo do ciclo está entre 0,75 microssegundos e 2 microssegundos. Já, as memórias de núcleos mais rápidas, possuem núcleos de menos de um qüinquagésimo de polegada de diâmetro e o período inferior a 500 nanossegundos.

A figura 1 nos mostra alguns aspectos de um núcleo toroidal de ferrite.

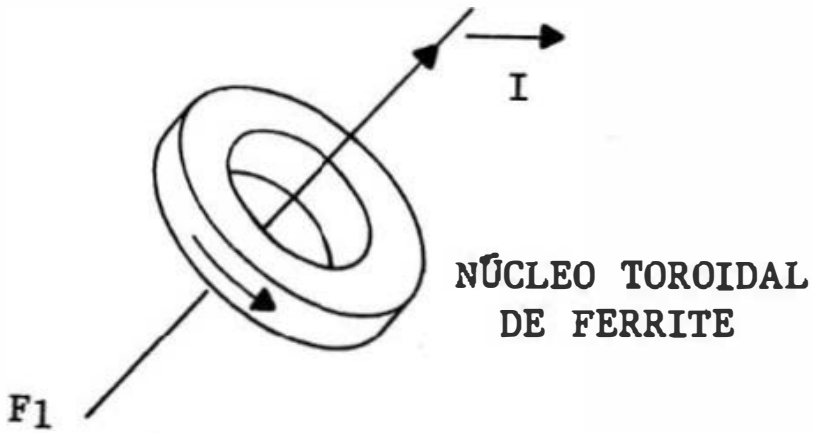
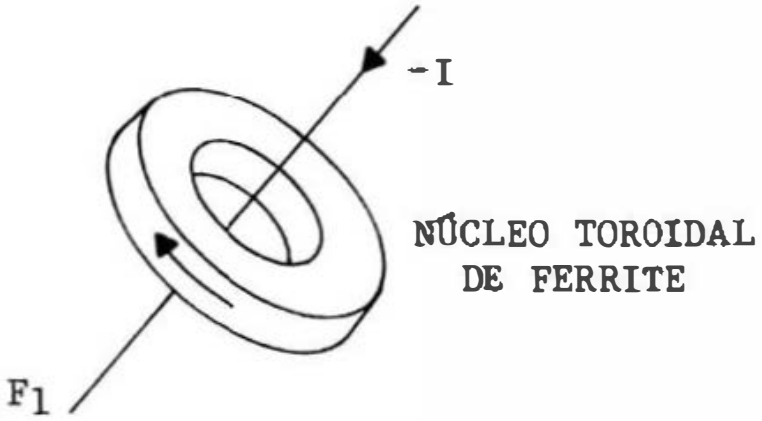


FIGURA 01

Quando uma corrente elétrica de intensidade I atravessa o núcleo (veja figura 1), esse se imanta (se magnetiza) em um dado sentido, o qual pode ser determinado pela regra do "saca-rolhas" (regra da mão direita) de Maxwell (ponto A da figura 3) e permanece imantado, mesmo que a corrente I deixe de circular (trecho Al da figura 3). Na figura 2, temos uma cor

rente elétrica -I circulando através do fio F_1. Nesse caso o núcleo se magnetizará em outro sentido, o que corresponde ao ponto B da figura 3. Quando a corrente -I deixar de circular, o núcleo permanecerá imantado, como no caso anterior, o que corresponde ao "segmento" B1 da figura 3.

Dessa forma,podemos concluir que o núcleo pode ser magnetizado em duas direções: no sentido horário e no sentido anti-horário.



ESCRITA FIGURA 02

Por conseguinte, o núcleo de ferrite apresenta dois estados estáveis, representados pelos segmentos A1 e B1, mostrados na curva de "magnetização-corrente" (veja figura 3), os quais podem servir como elementos de memória. Por convenção, diremos que o núcleo está no estado lógico "1" quando ele es

CURVA DE "MAGNETIZAÇÃO-CORRENTE"

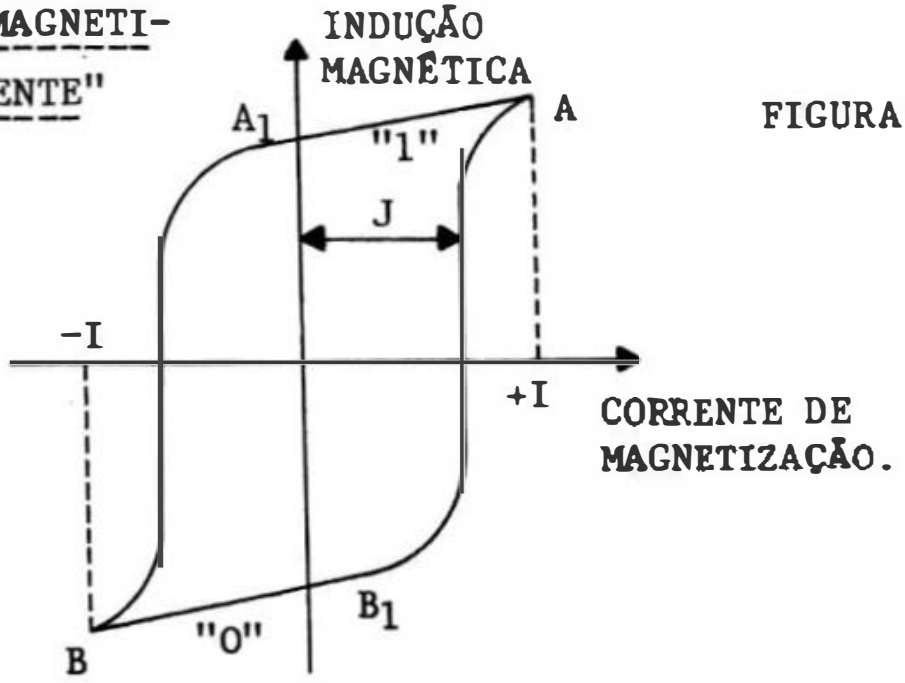


FIGURA 03

tiver magnetizado de acordo com o trecho A1 da curva de "magnetização - corrente" (sentido anti-horário). Caso contrário (sentido horário de magnetização), diremos que o núcleo está no estado lógico "0" o que corresponde ao trecho B1 da curva de "magnetização-corrente".

O núcleo permanecerá em seu estado enquanto não for atravessado (ou não circular através do fio F1) por correntes de intensidade superior a J, em valor absoluto. Uma corrente igual a I/2, em valor absoluto, é incapaz de polarizar o núcleo.

A operação de escrita de um dígito sobre o núcleo toroidal, de acordo com a convenção adotada, é feita da seguinte forma:

a) para a escrita do dígito "1" basta enviarmos, através do fio F1, um impulso de corrente igual a +I;

b) para a escrita do dígito "0" basta enviarmos, através do fio F1, um impulso de corrente igual a -I.

A leitura da informação contida em um núcleo de ferrite pode ser feita utilizando-se as leis da indução magnética. Assim, ao polarizar-se o núcleo, a variação produzida pelo campo magnético induz uma corrente em um fio de leitura que o atravessa. Se no núcleo estiver escrito o dígito "1", ao fazermos circular uma corrente -I através do fio F1, essa irá inverter a polarização do núcleo, provocando uma variação do campo magnético. Em decorrência dessa variação, obteremos uma corrente induzida no fio F2. (veja figura 4).

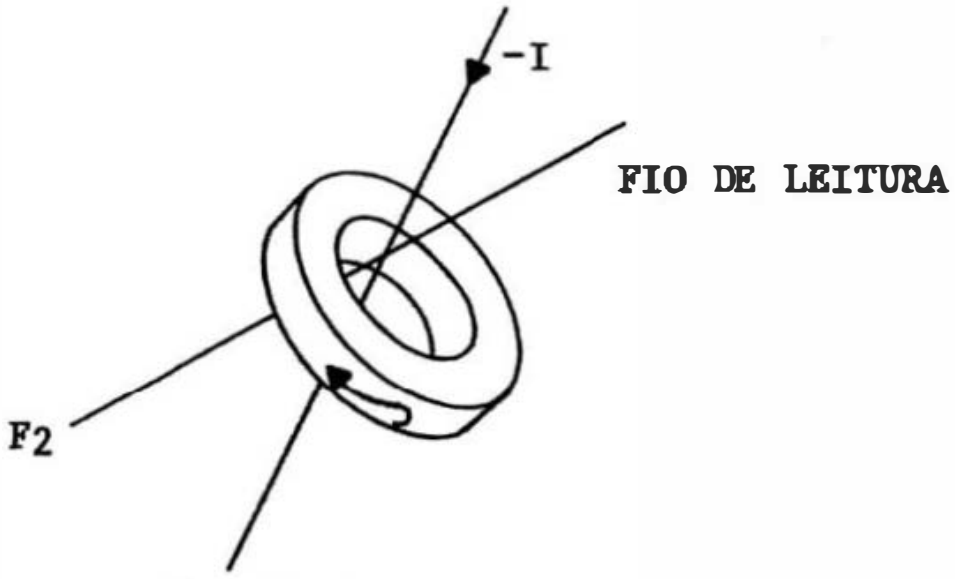


FIGURA 04

Entretanto, se o núcleo estiver, inicialmente, no estado "0", nada ocorrerá ao enviarmos uma corrente -I através do fio F1. Portanto, a presença de um impulso de corrente no fio de leitura F2 indica o conteúdo do núcleo. Esse tipo de leitura é destrutivo, uma vez que, ao final da operação, o núcleo sempre irá para o estado "0".

Resumindo, podemos dizer que o processo de escrever "0" é a técnica utilizada para ler o conteúdo do núcleo.

ORGANIZAÇÃO DAS MEMÓRIAS DE NÚCLEOS:

As memórias de núcleos magnéticos são organizadas em três grandes grupos, os quais correspondem a três tipos de seleção: seleção linear ou seleção em duas dimensões (2D); seleção por coincidência de correntes, que compreende a seleção em três dimensões (3D) e, por último, a seleção em duas dimensões e meia (2 1/2 D). Vejamos, então, cada um desses grupos.

A) Seleção em duas dimensões (2D):

A memória mais simples consiste em uma matriz de núcleos, sendo que cada núcleo é atravessado por dois fios perpendiculares, um fio de palavra e um fio de bit. Os núcleos de uma mesma linha (fila) memorizam uma palavra. Daí o fato das memórias assim concebidas serem chamadas de "organizadas por palavras" e que sua "seleção é linear ou em duas dimensões (2D)".

Para se escrever um "1" em um núcleo magnético, basta enviarmos uma corrente de intensidade igual a +I/2 através do fio de palavras e uma corrente de mesma intensidade (+I/2) através do fio de bit (veja figura 5).

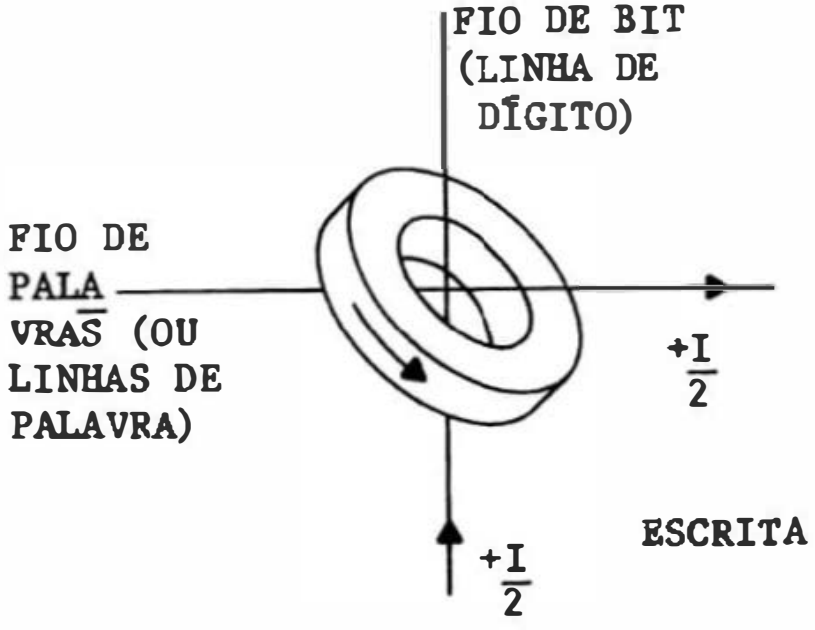
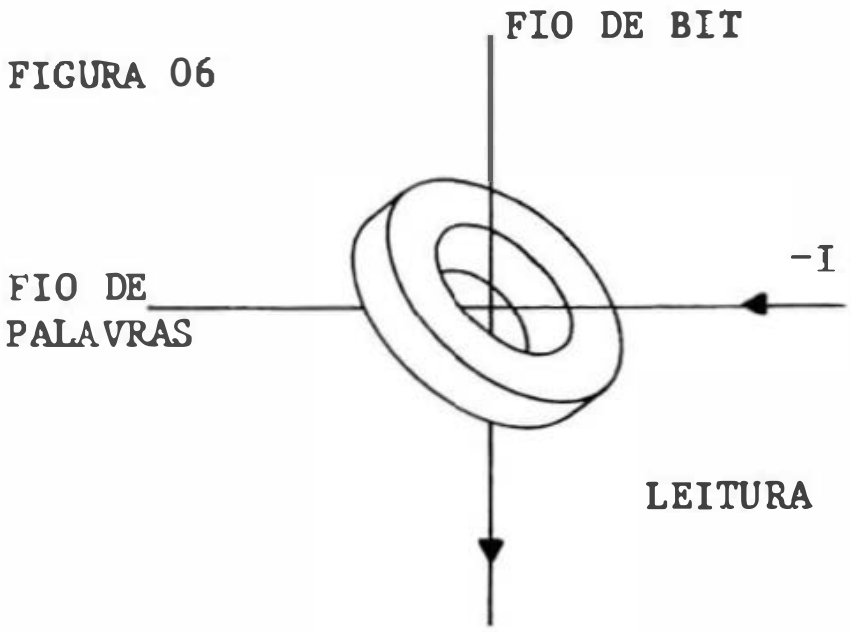


FIGURA 05

Para a operação de leitura, basta enviarmos uma corrente de intensidade -I através do fio de palavra (veja figura 6).

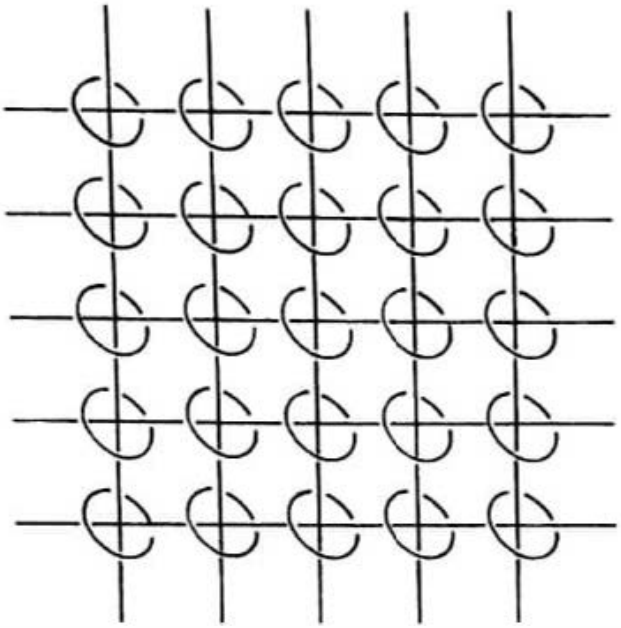
FIGURA 06



Para se escrever uma determinada configuração binária em uma palavra da memória (suposta, inicialmente, zerada), devemos enviar uma corrente +I/2 através do fio de palavra correspondente e uma corrente +I/2 através de todos os fios de bit. Dessa forma, somente os núcleos que são atravessados simultaneamente pelas duas correntes serão polarizados (ou magnetizados) visto que a corrente +I/2 por si só é incapaz de provocar a magnetização de um núcleo. A operação de leitura de uma palavra consiste em enviarmos uma corrente de intensidade -I, através do fio de palavras. Desse modo, os núcleos da palavra que estiverem na posição "1" têm sua polarização invertida, o que provoca o aparecimento de uma corrente induzida nos fios de bits correspondentes.

Na figura 7 temos uma matriz de núcleos, construída de acordo com a técnica 2D.

Observa-se que a leitura é destrutiva, uma vez que leva a "0" todos os bits da palavra. O processo de escrita, por sua vez, exige que todos os bits da palavra sejam levados a "0" previamente. Com base nesses fatos, é comum dividir toda a operação da memória em duas fases: uma fase de leitura e



MATRIZ DE NÚCLEOS

FIGURA 07

uma fase de escrita. No caso de uma busca na memória,a segunda fase tem por objetivo reescrever a palavra lida, o que nos leva a concluir que, ao nível do ciclo da memória, ela não é destrutiva. Já, no caso de um registro na memória, a primeira fase leva a "0" (zeramento) a palavra na qual se pretende realizar o registro. A figura 8 nos mostra um resumo dessas duas fases.

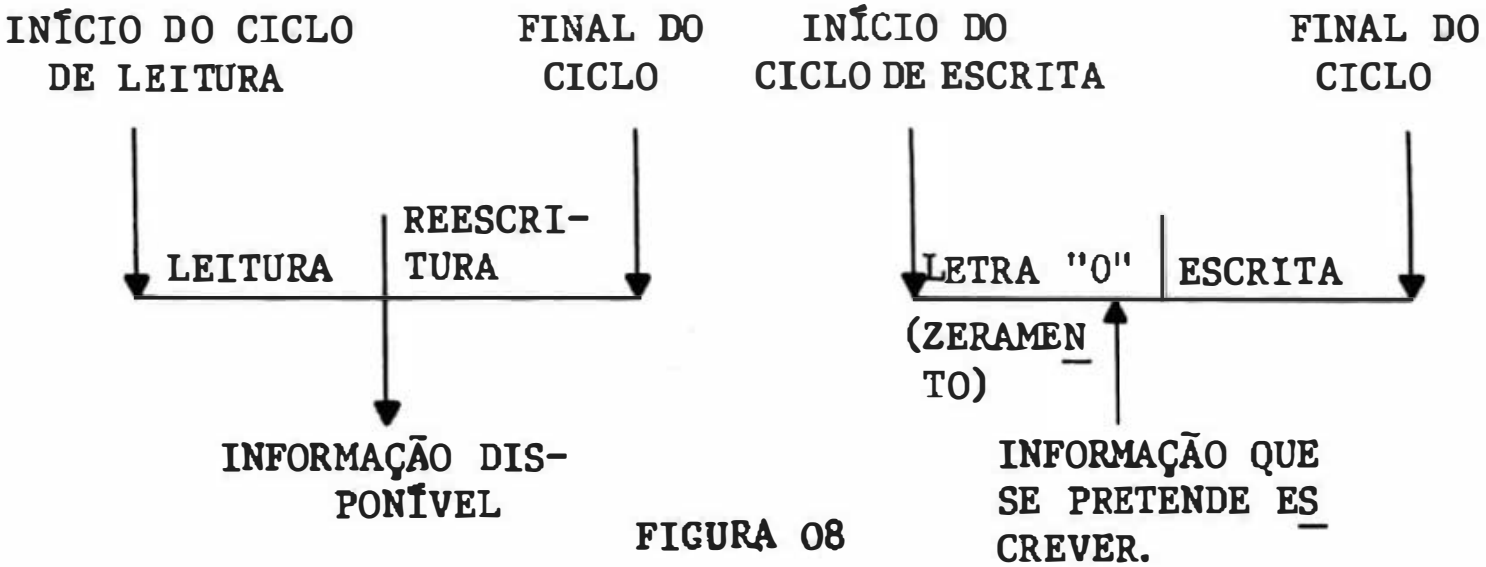
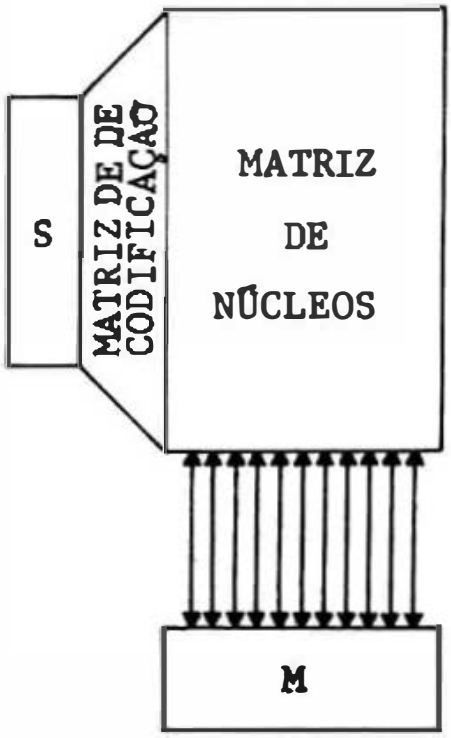


FIGURA 08

Na figura 9, temos um diagrama esquemático da organização geral de uma memória, organizada por palavras, sem incluir os amplificadores para gerar as correntes I e I/2 e pa-

ra amplificar os impulsos de leitura. O registro S contém a direção da memória e, por intermédio da matriz de decodificação, pode sensibilizar os "fios de palavra". (linhas de palavras)

FIGURA 09



ORGANIZAÇÃO DE UMA MEMÓRIA QUE EMPREGA A TÉCNICA 2D.

REGISTRO DUPLO (UMA PARTE PARA LEITURA, OUTRA PARA ESCRITA).

B) Seleção por correntes coincidentes (3D):

Nesse tipo de técnica de organização da memória, parte da decodificação se realiza na própria memória. A memória divide-se em "m" matrizes de 2^n núcleos, onde m é o número de bits da palavra da memória, enquanto n é o número de bits de seleção. Cada núcleo "i" de qualquer uma das matrizes corresponde ao bit de "peso" (valor)"i"de uma das 2^n palavras da memória.

Cada núcleo está atravessado por quatro fios, sendo dois para seleção, X e Y, ligados em série com as "m" matrizes de núcleos e dois fios funcionais, utilizados para leitura e inibição, sendo um par para cada matriz.

Na figura 10, temos dois núcleos, os quais nos mostram os quatro fios que os atravessam.

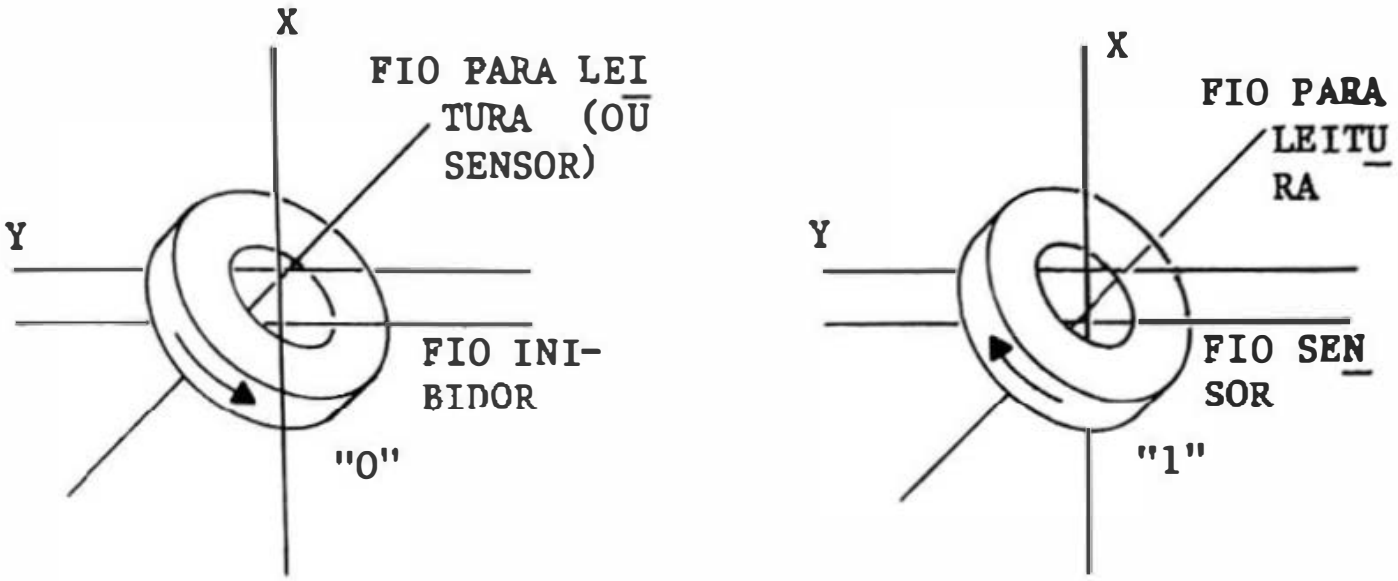


FIGURA 10

Os fios X e Y transportam somente metade da corrente de magnetização (1/2H) necessária para mudar a direção de polarização do núcleo.

Os núcleos são montados na forma de uma matriz, com os fios de endereçamento(X e Y) formando um reticulado com um núcleo em cada interseção (veja figura 11).

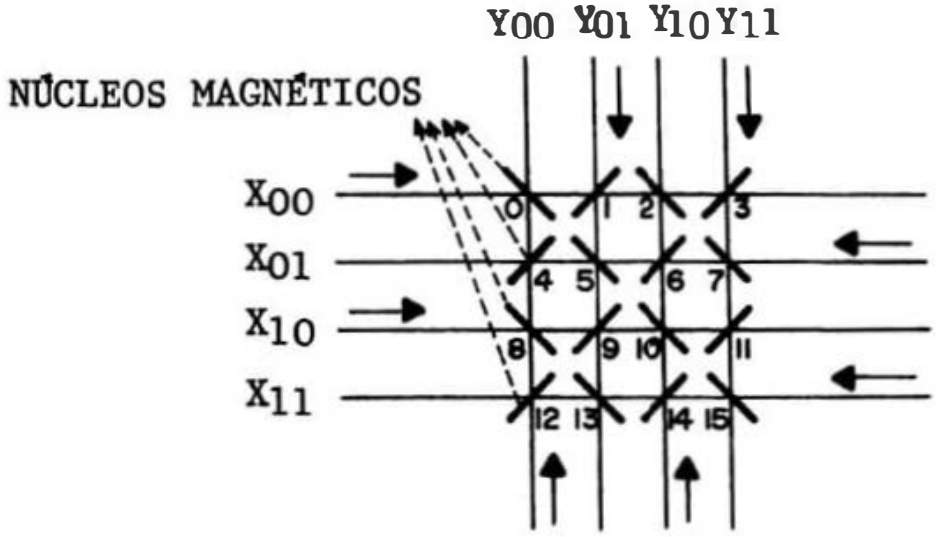


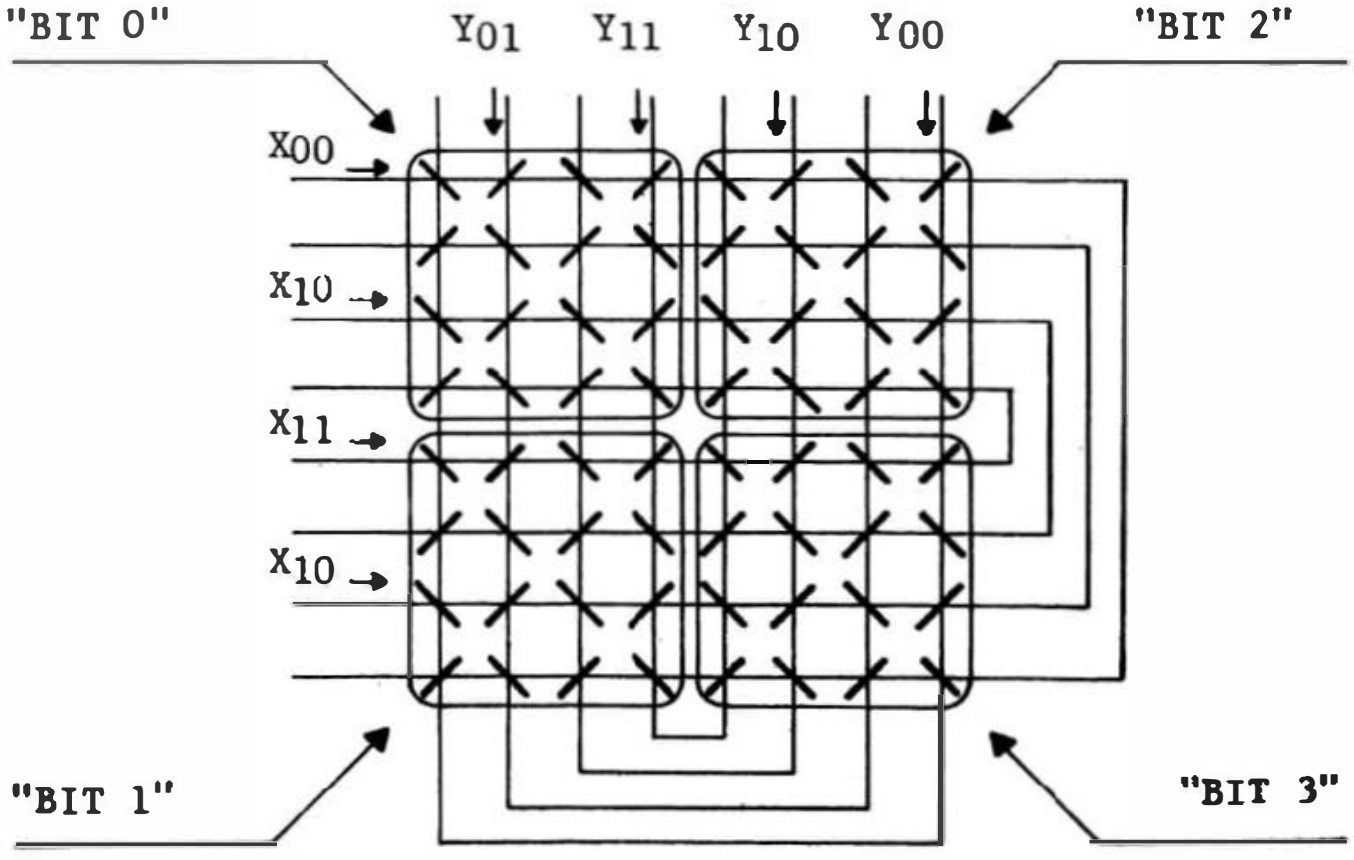
FIGURA 11

FIAÇÃO DE ENDEREÇAMENTO DE CORRENTE COINCIDENTE

Se aplicarmos uma corrente +I/2 no fio X_{10} e +I/2 no fio Y_{01}, somente o núcleo nove (9) receberá uma corrente total igual a +I. Dessa forma, diremos que somente o núcleo nove (9) foi selecionado. Os núcleos 1, 5 e 13 recebem +I/2 proveniente do fio Y_{01} e os núcleos 8, 10 e 11 recebem +I/2 do fio X10. Esses núcleos (1,5,13,8,10 e 11) são perturbados um pouco, devido à passagem de uma corrente igual a +I/2. Porém, essa corrente não é suficiente para fazer com que esses núcleos mudem de estado. Para escrevermos "1" binário no núcleo 9, basta aplicarmos uma corrente de intensidade -I/2 no fio X_{10} e -I/2 no fio Y_{01}.

Uma memória cuja palavra seja de 4 bits, normalmente possui um reticulado (X-Y) para cada bit, ou seja, possui um total de quatro reticulados. Por uma questão de economia nos drivers, o driver X10, por exemplo, passa através de todos os quatro retículos, conforme nos mostra a figura 12.

FIGURA 12



MEMÓRIA DE NÚCLEOS DE 16 PALAVRAS
DE 4 BITS

O processo de reescrever a palavra lida é um pouco mais complicado. Aplicando-se +I/2 e -I/2 respectivamente aos fios de endereçamento (X e Y) só podemos escrever palavras "0000" ou "1111", respectivamente. A solução para esse problema foi obtida a partir da colocação de um quarto fio no núcleo, chamado "inhibit", cuja finalidade é anular uma das correntes - (fio inibidor). Cada reticulado possui seu próprio fio inibidor (inhibit). Na figura 13 mostramos a fiação do inibidor (inhibit) em um retículo.

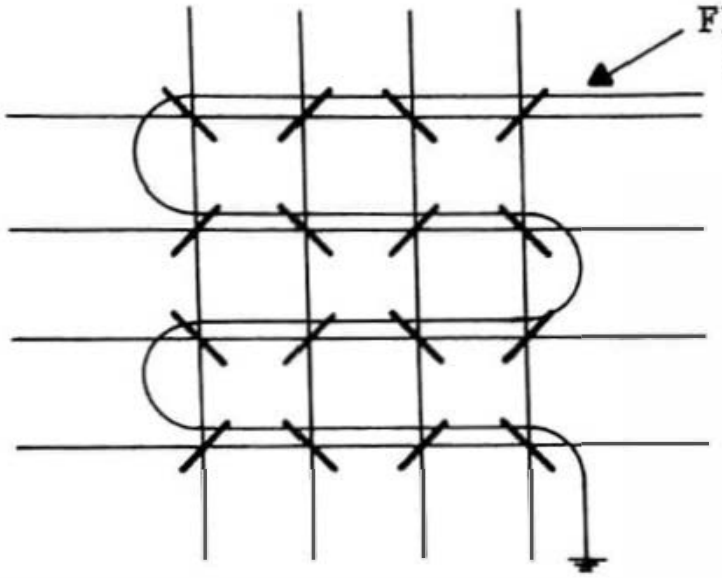


FIGURA 13

FIAÇÃO DO INHIBIT
EM UM RETÍCULO

Na figura 14 apresentamos uma fotografia de uma parte de um plano típico de núcleo de memória, ampliada cerca de 10 vezes.

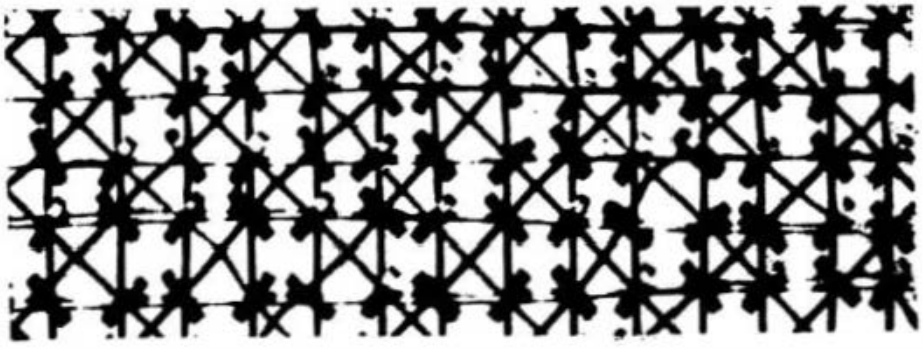


FIGURA 14

A estrutura de fiação dos núcleos magnéticos, com os fios de endereçamento (X e Y), o fio de leitura e o fio inibidor (inhibit) é chamada 3D de quatro fios. A notação 3D vem das três dimensões da memória: os fios X e Y formam um reticulado de duas dimensões. A terceira dimensão é constituída pelo número de bits por palavra que é igual ao número de reticulos. Normalmente, o fio de leitura e o fio inibidor são combinados em um único fio, facilitando assim a passagem dos fios através do núcleo, pois, ao invés de passarmos 4 fios, agora iremos passar apenas 3 fios.

Esse tipo de memória foi, por muitos anos, nos computadores, a memória padrão de alta velocidade.

C) Seleção em duas dimensões e meia (2 1/2 D):

Trata-se, também, de uma técnica de seleção por correntes coincidentes, diferenciando da seleção 3D pelo fato de que a inibição na escrita não se situa ao nível dos núcleos, mas sim ao nível das correntes de seleção. Nessa técnica, temos apenas dois fios atravessando os núcleos. Os fios, na direção dos bits horizontais, são usados como fio de leitura, no meio-ciclo de "leitura", ou para conduzirem correntes +I/2

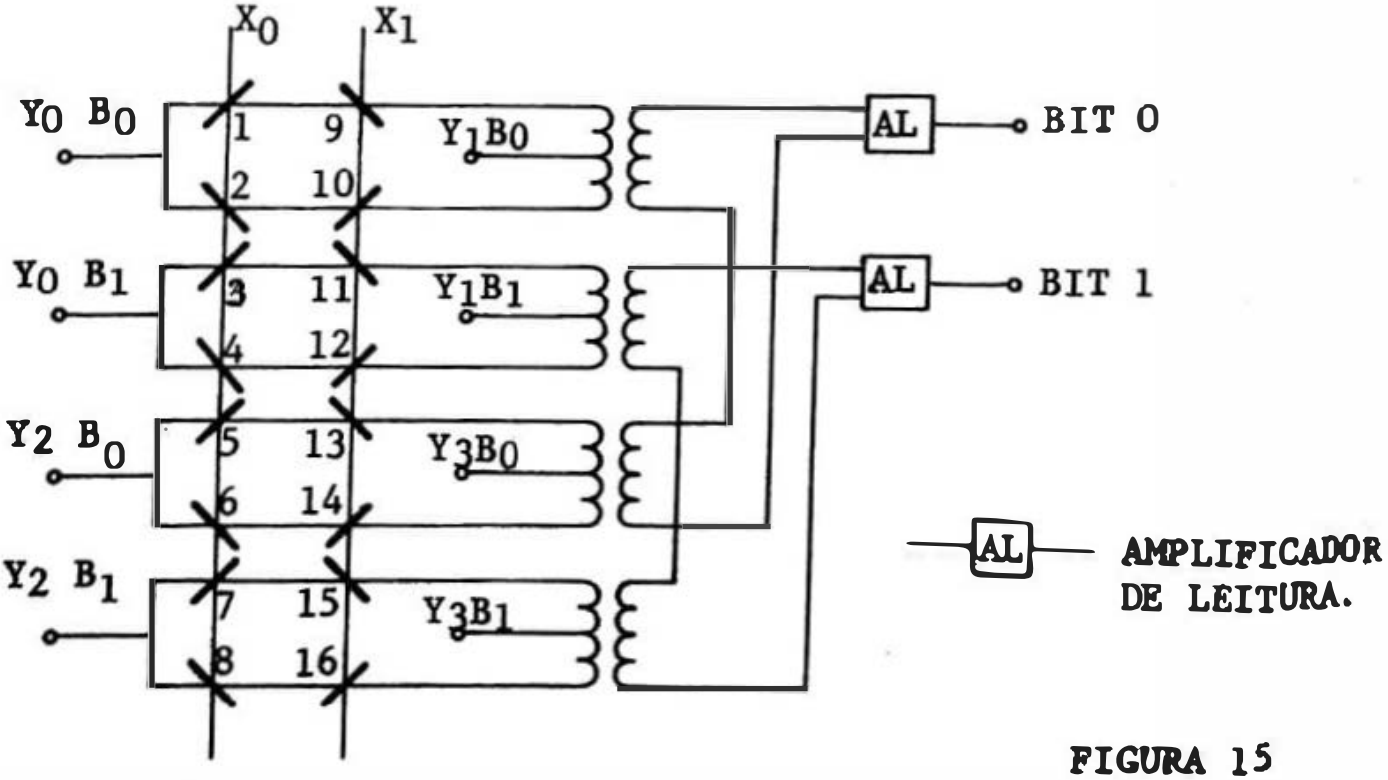


FIGURA 15

MEMÓRIA DE DOIS FIOS TIPO 2 1/2 D.

ou -I/2 no meio-ciclo de "escrita". Trata-se, portanto, de um esquema relativamente lento.

A figura 15 nos mostra uma memória de dois fios, tipo 2 1/2 D, de oito palavras de 2 bits.

O processo de seleção dos bits a serem lidos depende da polaridade da corrente na direção dos bits. Por exemplo, se queremos ler o conteúdo dos dois bits da palavra XOY1, é preciso que uma corrente de intensidade +I/2 passe pelo fio X_0 e que correntes, também de intensidade +I/2, circulem nos pares de fios, cujos terminais sejam identificados por Y0B0 e $Y1B_1$. No instante da leitura, teremos correntes +I/2 circulando da esquerda para a direita, nos quatro fios superiores da figura 15, e teremos também uma corrente +I/2 circulando, de cima - para baixo, no fio X_0. Dessa forma, os únicos núcleos que serão atravessados por uma corrente total de intensidade +I, são os núcleos marcados com os números 2 e 4. Observe que o processo básico é praticamente o mesmo utilizado nos outros dois tipos de seleção. O único inconveniente reside na lentidão apresentada por esse processo.

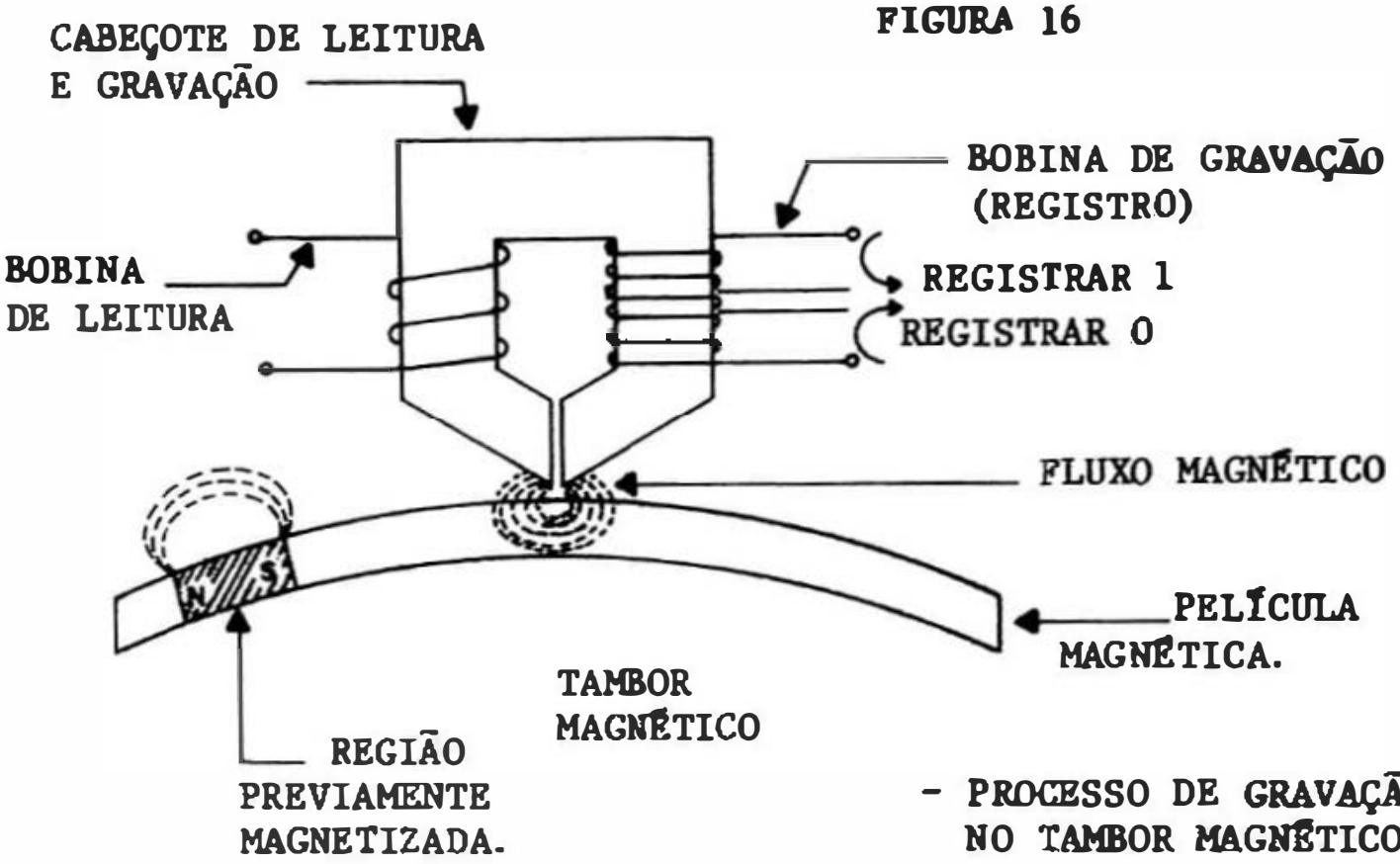
MEMÓRIA DE TAMBOR MAGNÉTICO:

Trata-se de um cilindro, cuja superfície externa está revestida de uma película magnética que se constitui no meio de registro propriamente dito.

Ao se colocar uma certa região dessa película magnética sob a ação de um campo magnético, ela se torna magnetizada. Mesmo que, após a magnetização da região, a força magnetizante for suprimida, a região continuará magnetizada indefinidamente. As informações gravadas na superfície do tambor podem ser lidas várias vezes. Porém, cada vez que um novo dado é gravado, o dado antigo é automaticamente apagado. A informação é gravada ou recuperada (lida) através de cabeçotes de leitura e gravação, os quais ficam separados da superfície do cilindro por uma distância muito pequena (veja figura 16).

A superfície do tambor magnético é magnetizada através do envio de pulsos de correntes para as bobinas de registro dos cabeçotes de leitura e gravação. Todo tambor possui um certo número específico de localizações de memória, sendo que cada uma delas é endereçável pelo computador. A capacidade de cada localização de memória depende do projeto do tambor

FIGURA 16



- PROCESSO DE GRAVAÇÃO NO TAMBOR MAGNÉTICO

e do código de representação de dados utilizados.

Como exemplo de memória de tambor, podemos citar o "Tambor 2303" da IBM, o qual gira com uma velocidade constante igual a 3500 rpm. Sua superfície é dividida em canais de endereço que se estendem ao longo da periferia do tambor e que são usados para a armazenagem de dados. Esse tambor (2303)possui 800 canais de dados standard e 80 canais de dados alternados. Os canais alternados são fornecidos para garantir que cada bit registrado possa ser armazenado em um meio magneticamente seguro. Se houver, por exemplo, algum defeito em um canal standard, esse será substituído por um canal alternado, o qual passa a receber o endereço do canal defeituoso.

Cada canal de dados possui seu próprio cabeçote de leitura e gravação, o qual é usado para gravação e recuperação de informações (dados). Os cabeçotes estão fixados em 20 grades verticais que circundam o tambor, sendo que, em cada uma dessas grades, temos 40 cabeçotes. Os cabeçotes de leitura e

gravação contêm pequenos núcleos magnéticos envolvidos por bobinas. Durante as operações de gravação, esses núcleos convertem sinais elétricos, recebidos do computador, em um fluxo magnético capaz de magnetizar pontos determinados da superfície do tambor. Porém, durante o processo de leitura, a ação é invertida, ou seja, os pontos magnetizados da superfície do tambor geram um fluxo magnético, o qual é convertido em sinal elétrico, pelo cabeçote e transmitido ao computador.

Na figura 17, temos uma memória de tambor IBM 2303.

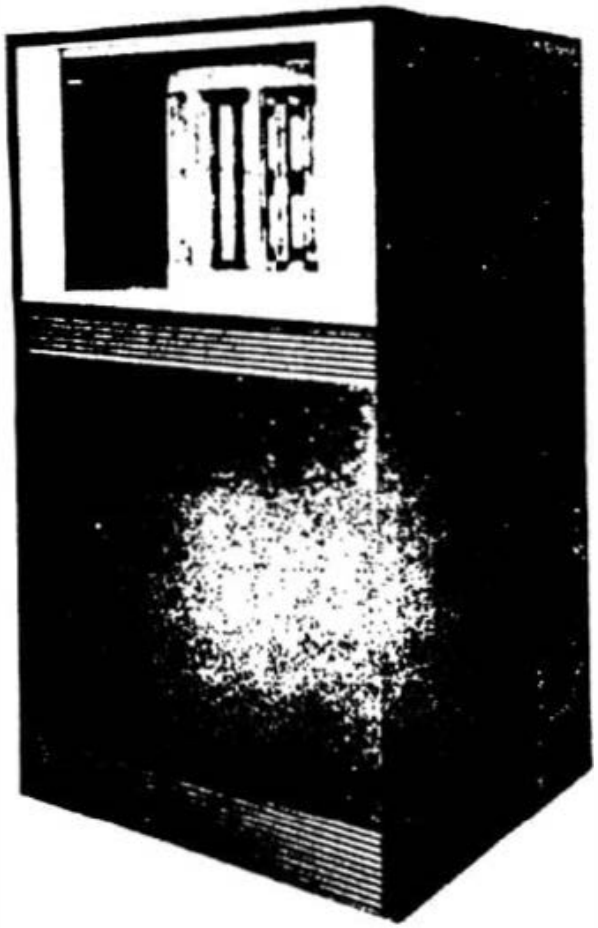


FIGURA 17

MEMÓRIA DE DISCOS MAGNÉTICOS:

Esse tipo de memória possibilita a gravação ou recuperação de dados de modo seqüêncial ou arbitrário. Os sistemas de discos magnéticos permitem o acesso imediato a áreas específicas de informações, sem necessidade de se examinar seqüencialmente todas as informações (dados) armazenadas.

A superfície de um disco é revestida de um material

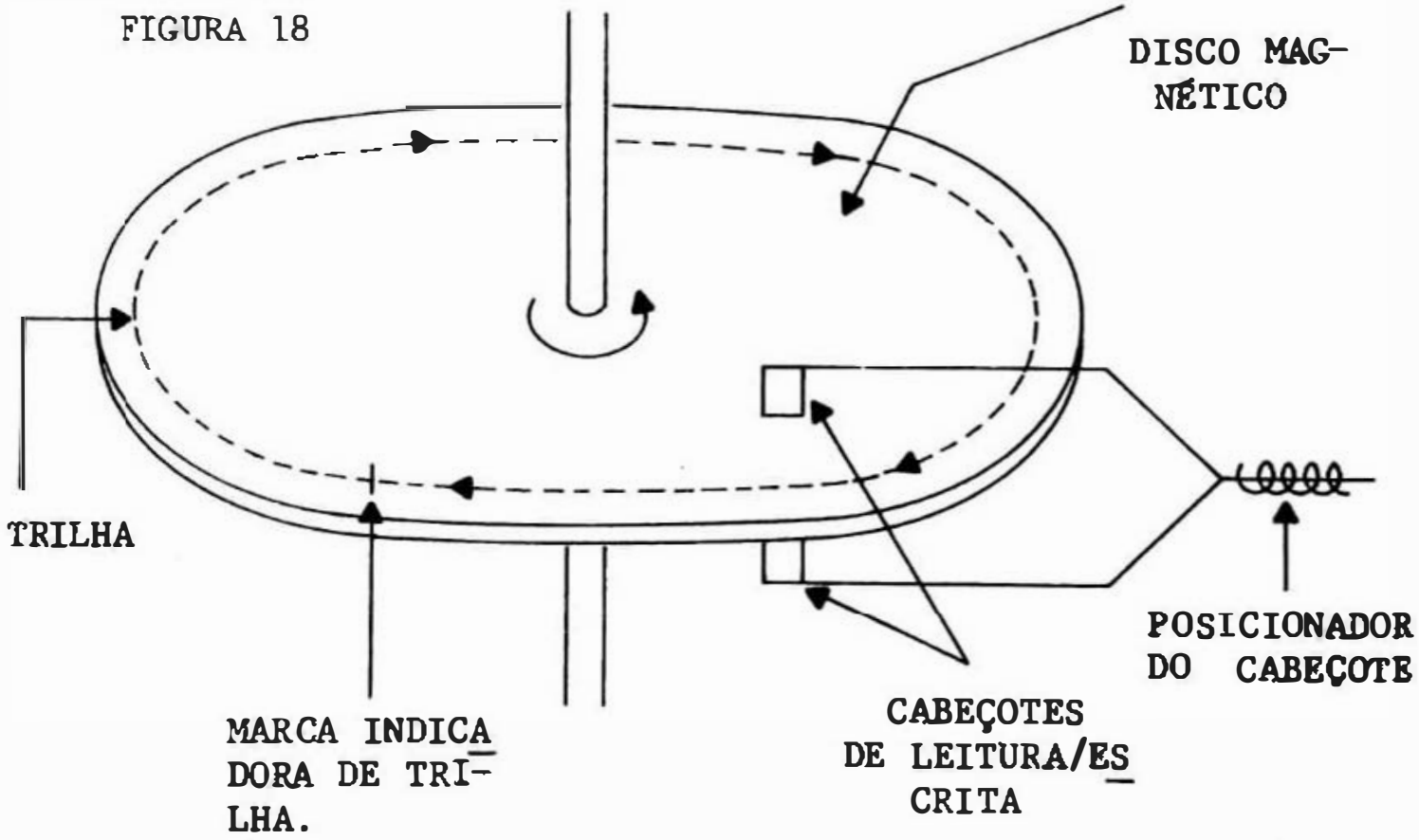
magnético apropriado para a gravação e a reprodução de sinais magnéticos.

Nos discos magnéticos, os sinais gravados são digitais, onde os dois níveis lógicos, alto (1) e baixo (0) correspondem aos dois sentidos em que se pode magnetizar um determinado ponto do disco, o qual está passando sob a cabeça.

A cabeça de leitura e escrita (ou cabeça de gravar e reproduzir) pode se movimentar, por passos, radialmente ao disco e, a cada posição, ela pode gravar, de forma circular, uma seqüência de informações a qual denominamos "trilha".

A figura 18 nos mostra um esquema básico de uma memória de disco magnético.

FIGURA 18



Cada trilha é fechada sobre si mesma, conforme podemos ver através da figura 18. Tanto o início como o final da trilha é assinalada por intermédio de uma marca indicadora de trilha, a qual é utilizada para a sincronização do volume. Essa marca não é sentida pelos cabeçotes de leitura/escrita, po

rém esta é detectada por um sensor especial do mecanismo do disco.

A informação registrada na trilha, em séries consecutivas de bytes, é denominada "bloco", o qual é limitado por um intervalo que não possui registro de informações e é utilizado pela unidade de controle para sua orientação (veja figura 19).

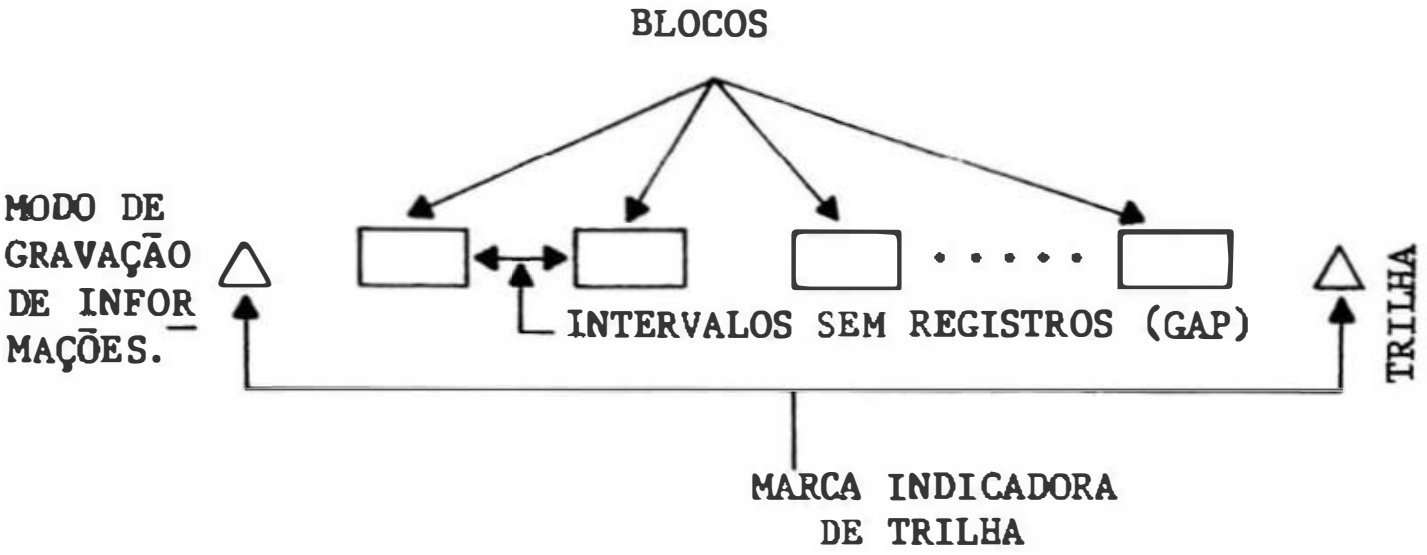


FIGURA 19

A capacidade de um disco magnético é calculada, multiplicando-se o número de bits que se consegue ler ou escrever em uma trilha pelo número de trilhas que se pode escrever na face do disco.

Atualmente a resolução do disco magnético já atinge a casa dos 12 (doze) milhões de bits por polegada quadrada, correspondente a aproximadamente 800 trilhas por polegada, ou 30 trilhas por milímetro, a uma razão aproximada de 15K bits (15 mil bits) por polegada (ou 600 bits por milímetro), ao longo da trilha.

O material magnético mais empregado, atualmente, na fabricação de discos magnéticos contém óxido de ferro e uma pequena quantidade de alumina, que aumenta a resistência da superfície do material aos impactos do cabeçote.

Um dos grandes problemas enfrentados na tecnologia das memórias de discos magnéticos era a distância entre a cabeça de leitura/escrita em relação à superfície do disco. Hoje, as

cabeças do tipo Winchester, utilizadas na maioria dos discos rígidos, permitem um funcionamento seguro a uma distância cabeça-disco da ordem de 0,3 micron. Esse tipo de cabeça trouxe grandes simplificações de projeto que culminaram com um grande salto em termos de índice custo/desempenho.

Hoje, dispomos também dos discos flexíveis (diskettes), os quais ganharam um lugar definitivo na família das memórias a disco, graças ao seu baixo custo e crescente durabilidade, como também em relação ao baixo custo dos drivers.

Os diskettes são confeccionados em um substrato (base) de "Mylar", revestido de óxido de ferro e envolvidos em um envelope de plástico com janelas, através das quais a unidade de acionamento prende o disco para que esse possa girar e a cabeça ler ou escrever sobre sua superfície, tocando-a.

A tendência atual em relação às memórias de massa está voltada para os discos ópticos. Esses discos, baseados no mesmo princípio dos video discos, utiliza o raio "laser" como elemento de leitura/escrita. Como sabemos, o raio "laser" nada mais é do que um foco de luz altamente concentrado que pode facilmente ser focalizado para formar um foco de dimensões micrométricas. Esse foco é então utilizado para alterar localmente as propriedades físicas da superfície do disco, adequadamente tratado.

De um modo geral, os discos magnéticos tendem a se tornar, muito em breve, um ítem cada vez mais importante nos sistemas de computação, à medida que aumenta a capacidade de processamento dos microcomputadores. Além do avanço obtido nesse setor, principalmente, com o desenvolvimento dos discos Winchester (de alta capacidade), abre-se fora do domínio magnético, uma nova opção, os discos ópticos digitais, os quais utilizam a tecnologia do raio laser.

MEMÓRIA DE FITA MAGNÉTICA:

Esse tipo de memória de massa foi muito usado durante vários anos, sendo ainda a de menor custo por bit armazenado.

A fita magnética é semelhante à fita utilizada em gravadores de som, ou seja, trata-se de uma fita plástica, revestida em uma das faces com uma película de óxido metálico. A informação é gravada na parte (face) revestida, sob a forma de

pontos magnetizados, denominados bits. A gravação pode permanecer na fita por tempo indeterminado ou ser automaticamente apagada, quando for gravada uma nova informação. Em decorrência desse fato, a fita pode ser utilizada inúmeras vezes.

A unidade de fita magnética grava e lê informações, à medida que a fita se move, passando através do cabeçote de leitura/gravação. Como já dissemos, a gravação na fita magnética é destrutível (à medida que novas informações são gravadas, as informações velhas são apagadas), porém a leitura não é destrutível, o que permite ler as informações repetidas vezes.

O processo de gravação das informações na fita é feito pela magnetização de áreas em trilhas paralelas ao longo do comprimento da fita. Na figura 20, mostramos um cabeçote de leitura/gravação de 1 entreferro.

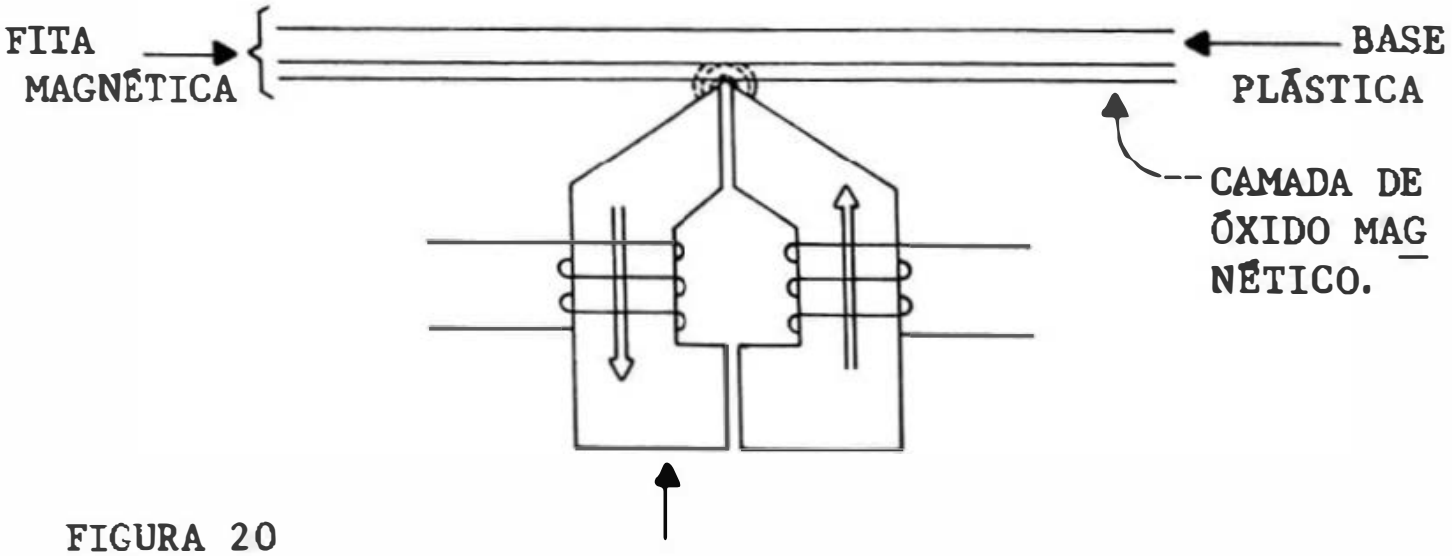


FIGURA 20

A figura 21 nos mostra o esquema de alimentação de fita magnética. Durante a gravação ou leitura, a fita é transferida do carretel arquivo, passando pelo cabeçote de leitura/gravação para o carretel da máquina. Durante o retrocesso ou reenrolamento da fita, o movimento da fita ocorre do carretel - da máquina para o carretel arquivo.

Existe um mecanismo especial que permite o deslizamento da fita através de cabeçotes de leitura/gravação, em velocidades constantes e um mecanismo especial para o controle da

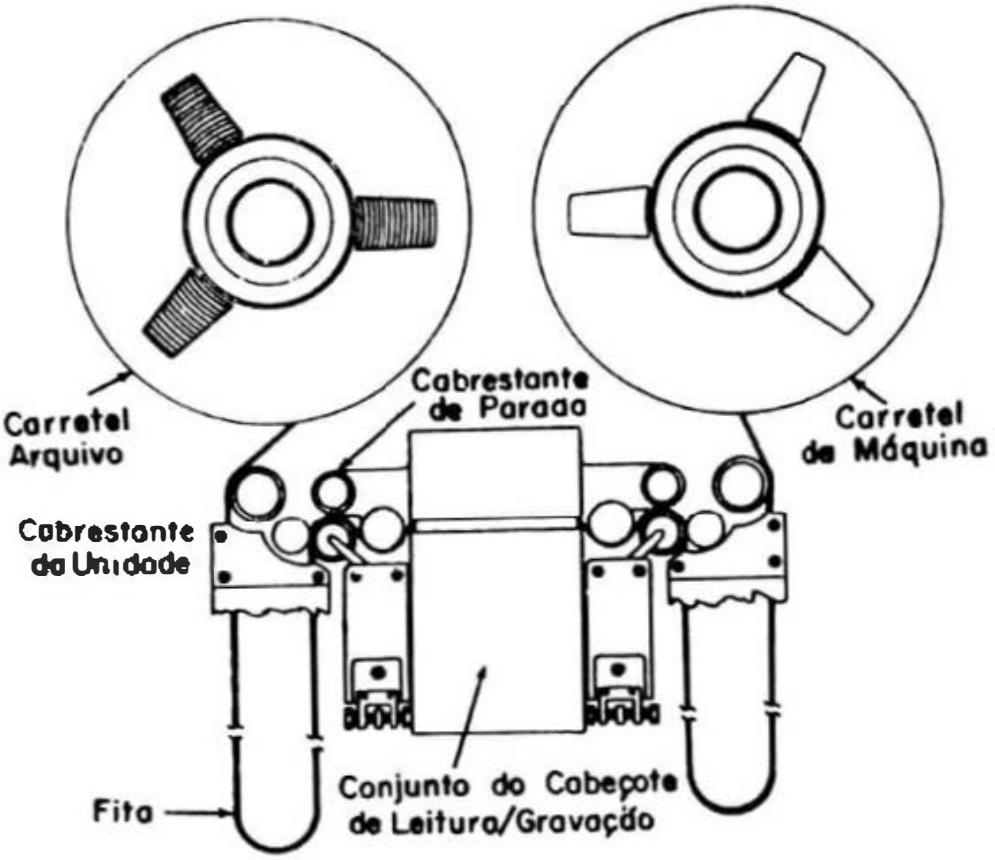


FIGURA 21

folga existente entre os carretéis e o cabrestante.

Hoje em dia, esse tipo de memória está sendo muito utilizado para a formação de "arquivos mortos".

COMENTÁRIOS FINAIS:

Nesta apostila, apresentamos apenas os "modelos" mais comuns de memória magnética. Atualmente, dispomos de outros tipos de memórias magnéticas como, por exemplo, as memórias de filmes finos de permalói, cujo princípio de funcionamento nos levou a duas técnicas possíveis de memórias integradas: memórias de lâminas planas (figura 22-A) e memórias de fios folheados (figura 22-B). Nessas memórias, na condição de memorização, o elemento permalói (ligas de ferro-níquel) é magnetizado em uma ou outra das direções fáceis. Já, na fase de escrita, a corrente de palavra leva a direção de magnetização próxima à direção difícil. A corrente de dígito (ou de bit) comuta essa direção para um ou outro lado e assim estabelece a direção fácil no final dos pulsos.

FIGURA 22A

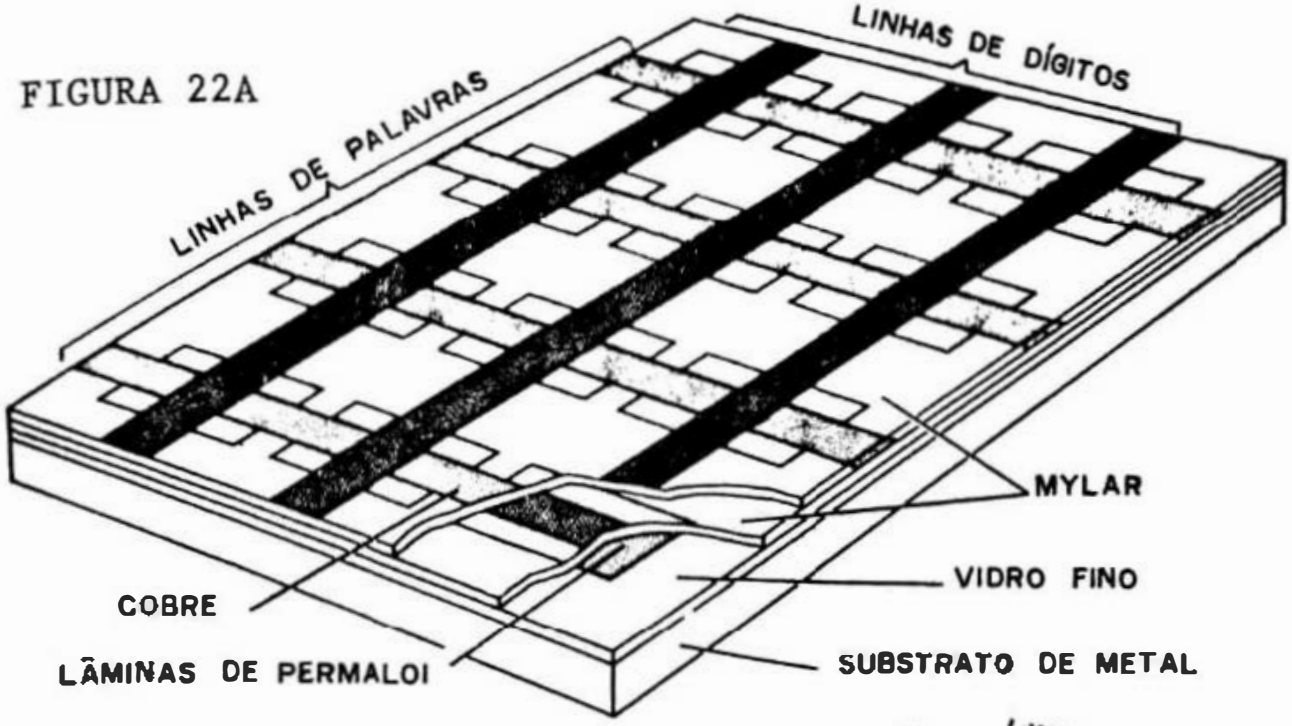
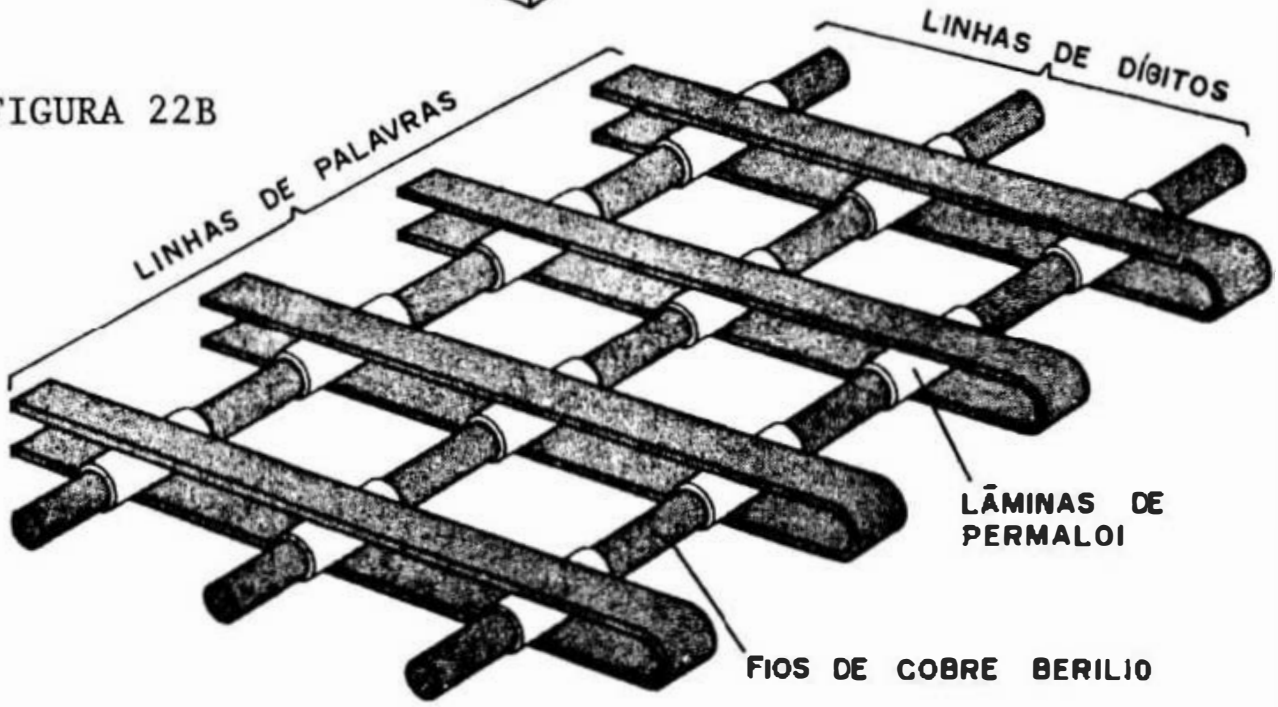


FIGURA 22B



No ciclo de leitura, um pulso de corrente é aplicada à linha da palavra selecionada. Os pontos correspondentes são forçosamente magnetizados na direção difícil, induzindo assim voltagens sonoras nas linhas de dígito, numa direção correspondente ao seu estado.

De uma forma generalizada, podemos dizer que todas as memórias magnéticas possuem características comuns, principalmente quanto aos processos de leitura e gravação. Ainda hoje, essas memórias são muito utilizadas, apesar de que com o advento dos circuitos integrados, surgiu a tendência de se eliminarem completamente essas memórias.

Na próxima apostila, iniciaremos o estudo sobre as memórias semicondutoras, muito aplicadas atualmente.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 · CEP 86100 · LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-4

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Os núcleos de ferrite são constituídos por:

a) cerâmica ferro-magnética;
b) cerâmica comum;
c) bolhas magnéticas;
d) película de óxido metálico.

2) Nas primeiras memórias de núcleo magnético, os núcleos tinham um diâmetro externo aproximadamente igual a:

a) 1 1/12 de polegada;
b) 1 polegada;
c) 12/19 de polegada;
d) 1/12 de polegada.

3) Na maioria dos computadores contemporâneos, o tempo do ciclo está entre:

a) 15 microssegundos e 20 microssegundos;
b) 0,75 microssegundos e 2 microssegundos;
c) 5 microssegundos e 10 microssegundos;
d) N.R.A.

4) Um núcleo de ferrite pode ser magnetizado em:

a) 3 direções;
b) 1 direção;
c) 2 direções;
d) 4 direções.

5) Na organização das memórias de núcleo, de acordo com a seleção em duas dimensões (2D), os núcleos são atravessados por:

a) 2 fios;
b) 4 fios;
c) apenas um fio;
d) N.R.A.

6) Na seleção tipo 2D, para se escrever "1" em um núcleo magnético, basta enviar:

a) uma corrente +I/2 através do fio de palavras;
b) uma corrente +I/2 através do fio de bit;
c) os itens a e b se completam;
d) uma corrente -I/2 através do fio de bit e uma corrente +I/2 através do fio de palavras.

7) Na seleção tipo 3D, cada núcleo está atravessado por:

a) 2 fios;
b) 4 fios;
c) apenas um fio;
d) N.R.A.

8) O tambor 2303 da IBM possui:

a) 800 canais standard;
b) 80 canais de dados;
c) 100 canais standard;
d) os ítens a e b estão corretos.

9) Com relação às memórias de discos magnéticos, sabe-se que:

a) as trilhas formam uma espiral;
b) cada trilha é fechada sobre si mesma;
c) não é possível a determinação do início e nem do final da trilha pelo cabeçote;
d) N.R.A.

10) O elemento permalói, normalmente, é feito de:

a) uma massa de óxido de ferro;
b) cerâmica ferro-magnética;
c) ligas de ferro-níquel;
d) N.R.A.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-5

MEMÓRIAS III

INTRODUÇÃO:

Nesta lição, iniciaremos o estudo sobre as memórias semicondutoras. Toda vez que se fala em dispositivos semicondutores, é comum, hoje em dia, associá-los a circuitos integrados. Pois bem, geralmente ou quase na totalidade, as memórias semicondutoras são disponíveis sob a forma de circuitos integrados. Todavia, além das memórias com circuitos integrados MOS e bipolar, temos outras memórias que também são encontradas sob a forma de CI, algumas das quais passaremos a enumerar, de acordo com a tecnologia aplicada em sua construção. São elas:

- Memórias CCD;
- Memórias de bolhas magnéticas;
- Memórias semicondutoras.

Hoje, dispomos de um grande número de memórias de circuito integrado. Dentre elas podemos distinguir:

A) Memórias RAM:

Do inglês "random access memories", memória de acesso aleatório. Trata-se, na realidade, de um tipo de memória na qual se pode escrever, ler, apagar e voltar a escrever informações. As memórias RAM se equivalem às memórias de núcleo de ferrite.

B) Memórias ROM:

Do inglês "read-only memories", são memórias de apenas leitura. Também de acesso aleatório, porém uma vez gravadas as informações, pelo fabricante, estas permanecem na memória para sempre. Não podem ser apagadas e regravadas.

C) Memórias PROM:

Trata-se de uma memória "ROM programável", a qual pode ser gravada pelo próprio usuário, em função de seu programa. Uma vez gravada pelo usuário, comporta-se como uma memória tipo ROM comum.

OBS.: Existem outras variações de memória ROM, as quais serão vistas oportunamente, quando do estudo das memórias ROM.

D) Registradores de Deslocamento:

Trata-se de memórias de acesso seqüêncial,nas quais as informações se deslocam de uma célula para a outra.

E) Registros:

São pequenas memórias lineares de acesso paralelo.

F) Pilhas:

Podem ser definidas como um empilhamento de registros, cujo princípio de funcionamento já foi discutido anteriormente.

G) FIFO:

Trata-se de uma memória tipo pilha, cuja forma de funcionamento já foi vista anteriormente.

Dando continuidade ao assunto iniciado na lição passada, vamos agora apresentar um breve estudo sobre as memórias CCD e as memórias de bolhas magnéticas.

MEMÓRIAS CCD (DISPOSITIVOS ACOPLADOS POR CARGA):

O funcionamento básico deste tipo de memória é idêntico ao dos registradores de deslocamento, porém a tecnologia empregada é diferente. Na figura 1, temos uma estrutura MOS, ou seja, uma camada de material dopado, recoberta por uma camada de óxido, com metalização seletiva sobre o óxido.

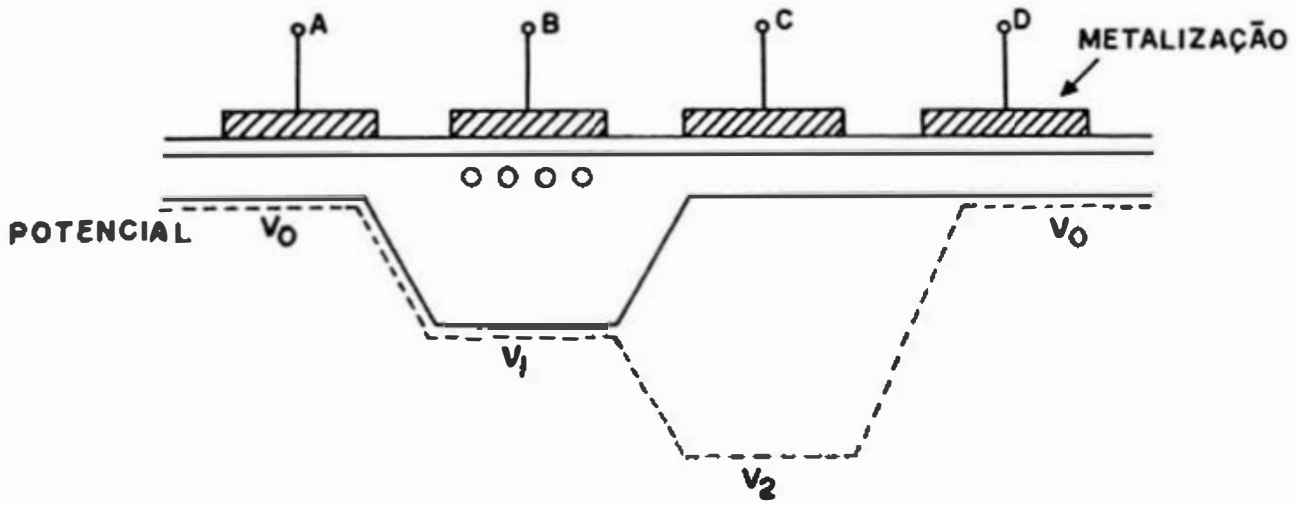


FIGURA 01

Seu funcionamento é bem simples. Vejamos: Com a aplicação de um potencial positivo (V1) em B, cria-se, em relação aos outros terminais, uma região na superfície do semicondutor, de frente ao terminal B, capaz de manter fixa uma quarta carga. Se, neste mesmo instante, aplicarmos um pulso positivo de tensão (V2), no terminal C, maior que o que foi aplicado em B, o potencial aparece agora, distribuído sobre a superfície do semicondutor, de acordo com a linha tracejada (veja figura 1). Isto significa que as cargas armazenadas em baixo

(na região) do terminal B se deslocaram para uma outra região, situada abaixo do terminal C. Desta forma se origina um deslocamento de carga análogo ao deslocamento de informações em um registrador de deslocamento comum. A carga é normalmente introduzida no dispositivo,através de uma junção PN de um transistor MOS convencional. Por outro lado, a carga retirada ou recolhida,através de um amplificador cujo funcionamento é semelhante à dos amplificadores usados nas memórias MOS dinâmicas.

Uma das grandes vantagens apresentadas por esta tecnologia é que não se torna necessário nenhum tipo de difusão seletiva, evitando-se problemas com o armazenamento de máscaras,o que permite uma maior concentração de informações por unidade de área de superfície. No quadro abaixo, figura 2, comparamos a quantidade de informações armazenadas em alguns tipos de memórias.

memória MOS estática	1024 bits	tipo 2102
memória MOS dinâmica	4096 bits	tipo 2107
memória CCD	16.384 bits	tipo 2416

FIGURA 02

MEMÓRIAS DE BOLHAS MAGNÉTICAS (MBM):

A tecnologia de fabricação das memórias de bolhas magnéticas é similar à tecnologia empregada na fabricação de dispositivos semicondutores em grande escala de integração (LSI).

Sobre um substrato de material magnético se faz um crescimento epitaxial de um material magnético cristalino. No interior desta camada de material magnético existem, ou podem existir, pequenos domínios (bolhas) estáveis, com uma polaridade inversa à de um campo magnético moderado e constante, o qual é aplicado externamente em todo o material, no sentido perpendicular. Essas bolhas podem chegar a ter um diâmetro em torno de 0,5 micro ($0,5 \times 10^{-6}$ metros), o que nos permite chegar a uma densidade de armazenamento de informação da ordem de 10^8 bits por polegada quadrada.

As bolhas magnéticas existentes no interior da camada

epitaxial se movimentam quando da aplicação de campos magnéticos transversais, permitindo assim a gravação ou a leitura de informações.

A camada epitaxial constitui o elemento base de uma memória de bolhas. Sobre a superfície desta camada se depositam algumas porções de "permaloi" (liga de ferro e níquel com elevada permeabilidade magnética) sendo que cada porção depositada corresponde a uma dada região da superfície. Da mesma forma também são formadas algumas regiões de AlCu. As regiões formadas pelo permaloi e pelo AlCu são responsáveis pelas funções de circulação, geração, réplica e aniquilação, e transferência.

Os movimentos das bolhas no interior da camada epitaxial são conseguidos através da aplicação de um campo magnético giratório, em sentido transversal. Na figura 3, podemos observar os movimentos das bolhas em um giro completo do campo aplicado. Podemos observar, também, como a trajetória da bolha é determinada pelo formato das regiões de permaloi, depositadas na superfície.

O formato das regiões de permaloi, apresentado na figura 3-b, é denominado de "estrutura de galões", devido à grande semelhança com as divisas utilizadas nos uniformes militares.

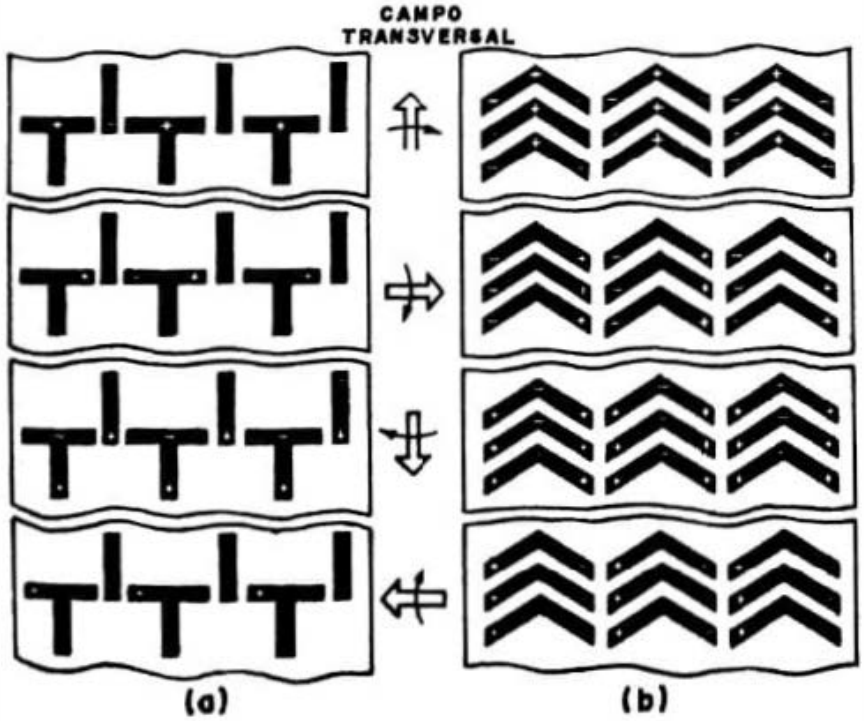


FIGURA 03

a - estrutura T - barra

b - estrutura de galões

Cabe esclarecer que a estrutura de galões não é o único método utilizado para a formação das regiões de permaloi. Existem outros. Dentre eles, destacamos o sistema chamado "T - barra" (veja figura 3-a).

A geração de uma bolha se consegue, aplicando-se um pulso de corrente bastante elevado sobre uma espira condutora de AlCu. Este pulso gera um campo magnético localizado, o qual deve ser suficientemente elevado para se opor ao campo magnético exterior aplicado e gerar a bolha. Esta bolha se desloca, conforme já foi exposto (veja figura 4).

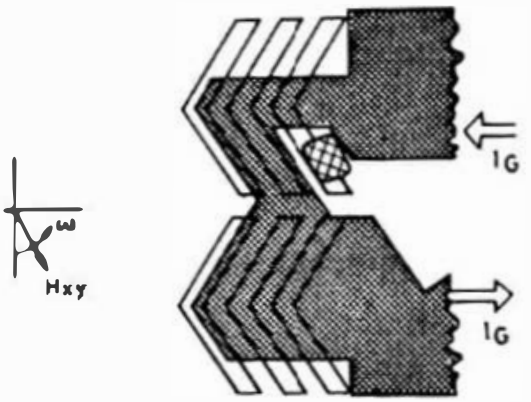
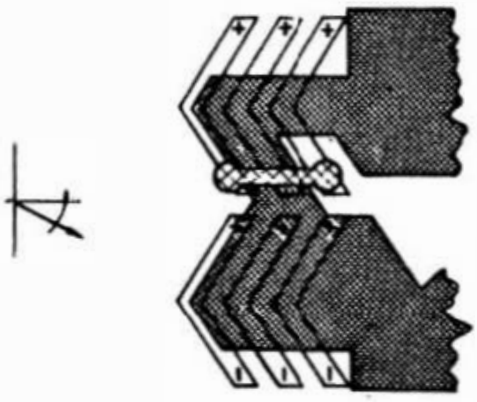


FIGURA 04

geração de uma bolha magnética



A réplica ou aniquilação de uma bolha se consegue através da dilatação do tamanho da bolha e, em seguida, dividindo-a em duas metades, através da aplicação de um pulso de corrente. Caso não seja aplicado esse pulso de corrente, a bolha segue sua trajetória normal. Ao contrário, se aplicarmos o pulso de corrente, a bolha se divide em duas metades, sendo que uma delas segue a trajetória normal, enquanto que a outra se dirige para um circuito adjacente ao replicador (ou circuito duplicador). Veja a figura 5.

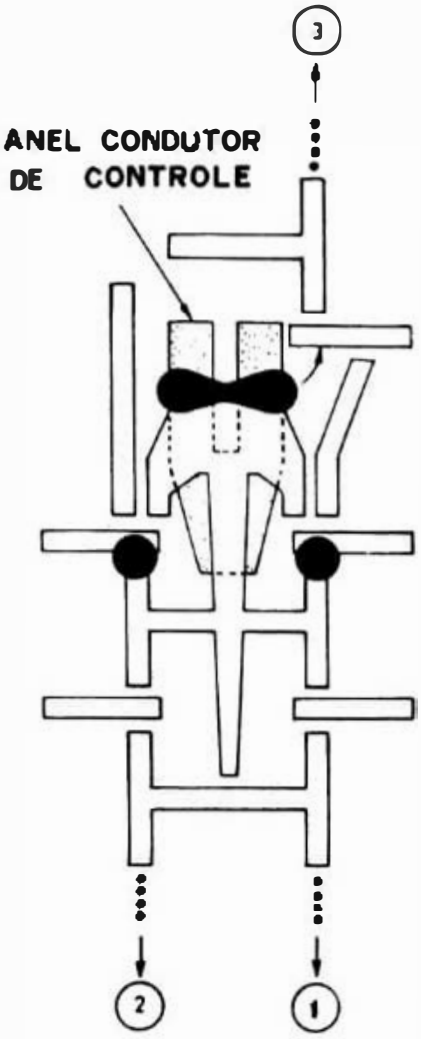


FIGURA 05

Esquema de um replicador (duplicador) de semidisco. A bolha entra por 1, se parte em duas metades, uma das quais segue o caminho normal 2 enquanto, que a outra vai para a saída 3 .

A detecção de uma bolha se faz com detectores magneto-resistivos, com uma prévia expansão da bolha, incrementando-se o número de galões que definem a trajetória da bolha até o detector. (Veja figura 6).

A porta de transmissão permite a transferência de uma bolha de um circuito para outro adjacente. Esta transferência pode ser controlada, de modo que seja realizada em um sentido ou no outro.

As principais etapas tecnológicas, necessárias para a fabricação de uma memória de bolha, são:

a) Crescimento epitaxial do material magnético;
b) Deposição das regiões de permaloi;
c) Deposição de uma camada de óxido de silício, a qual servirá como isolante;
d) Deposição dos circuitos de cobre-alumínio, os quais formam as pequenas espiras do gerador, do replicador, do detector e das portas de transmissão.

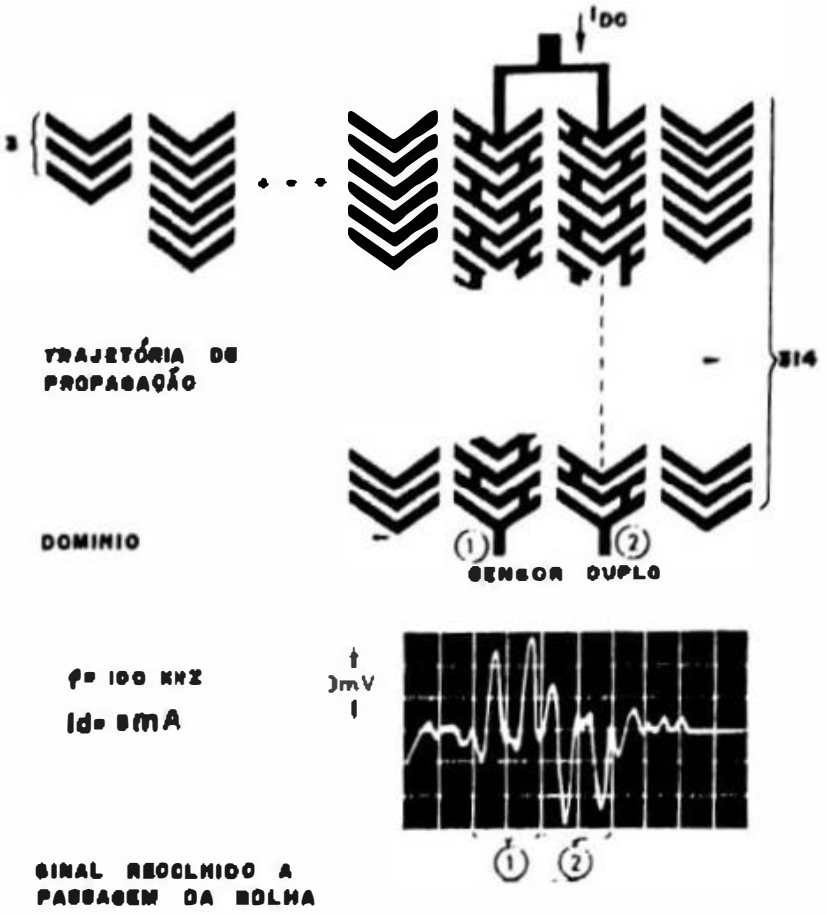


FIGURA 06

ORGANIZAÇÃO DE UMA MEMÓRIA DE BOLHA MAGNÉTICA:

A organização de uma memória de bolha magnética de aproximadamente 100 K bits pode ser vista na figura 7 (trata-se de uma TBM 0101).

As trajetórias, normalmente executadas pelas bolhas magnéticas, são trajetórias fechadas, denominadas "laços menores". Na memória em questão (TBM 0101), existem 157 laços deste tipo, sendo que cada laço contém 641 prováveis domínios (bolhas), o que significa que após 641 ciclos completos do campo giratório, as bolhas magnéticas estarão exatamente na mesma posição inicial.

Em um dos extremos de cada um dos laços menores, existe uma porta de transferência, ou de transmissão, adjacente a um único laço comum, denominado laço maior. Todas as portas de transmissão são controladas pelo mesmo pulso de corrente e permitem a transferência das bolhas contidas no laço maior a cada um dos laços menores e vice-versa. O comprimento do laço

maior é de aproximadamente 640 posições. É sobre o laço maior que são construídos o gerador ou gravador, o replicador e o detector.

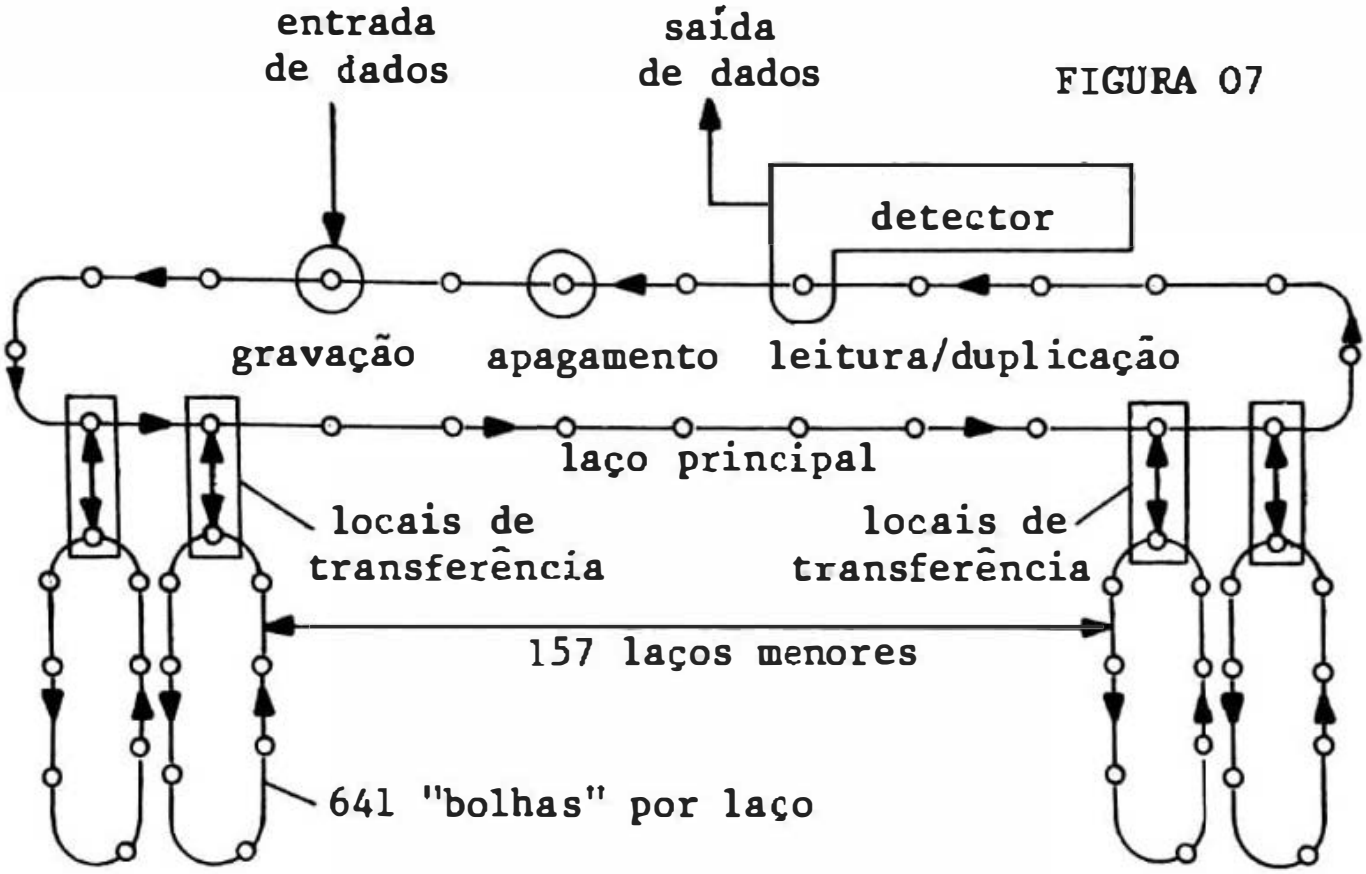


FIGURA 07

Cada laço menor corresponde a 1 bit. Desta forma, a memória em questão contém 157 bits de dados. Durante o ciclo de escrita, os 157 bits penetram, primeiramente, no laço maior, a partir do qual serão transferidos, a um certo sinal (pulso de corrente), para os 157 laços menores, sendo que cada um desses laços receberá um bit.

Por outro lado, durante a operação de leitura, os 157 bits são transferidos simultaneamente, a um certo sinal, dos laços menores para o laço maior, o qual os leva até a cabeça de leitura.

A capacidade total deste tipo de memória é definida como sendo o número de laços menores (157), multiplicado pelo número de bolhas que contém cada um desses laços (641). Desta forma, o dispositivo em questão apresenta uma capacidade total igual a 157 x 641, ou seja, 100.637 bits.

As memórias de bolhas possuem um custo inicial mais baixo que o custo dos sistemas de discos e fitas magnéticas, porém possuem um tempo de acesso muito inferior, da ordem de 4ms (4×10^{-3} seg.). Mesmo assim, a principal aplicação das memórias de bolhas magnéticas reside na substituição das memórias à base de fitas e discos magnéticos. Hoje em dia sua principal utilidade é como memória auxiliar de médio volume, exibindo uma capacidade de armazenagem entre 1 e 10 milhões de bits. O fato de serem memórias não voláteis, permite utilizá-las para a armazenagem do estado de um sistema, no caso de interrupção no fornecimento de energia.

Na verdade, as memórias de bolhas magnéticas são mais confiáveis, uma vez que se trata de um elemento puramente estático, ou seja, não possui peças móveis, apresentando, portanto, uma "alta" resistência às vibrações mecânicas. Além disso, apresentam um tamanho bem reduzido, assim como um baixo consumo de energia e podem operar em uma larga faixa de temperatura. Tudo isso nos leva a crer que as memórias de bolhas possuem um grande futuro.

MEMÓRIAS SEMICONDUTORAS:

Existem duas tecnologias amplamente difundidas no campo da microeletrônica: a MOS e a bipolar, as quais são utilizadas na fabricação de memórias.

Os dispositivos MOS são predominantemente dispositivos de superfície. Com esses dispositivos podemos construir dois tipos de memórias: voláteis (estáticas ou dinâmicas) e não-voláteis. Ainda, com relação à tecnologia MOS, podemos ter memórias de canal tipo "P", de canal tipo "N", de carga - depleção, C-MOS, V-MOS e H-MOS. As primeiras memórias semicondutoras construídas com tecnologia MOS foram as memórias MOS canal "P", as quais deixaram de ser fabricadas principalmente devido à sua lentidão. Já, as memória MOS canal "N" podem ser dinâmicas ou estáticas, as quais vêm sendo continuadamente aperfeiçoadas. Os últimos modelos de memórias MOS possuem transistores MOS de carga, operando em modo-depleção, o que resulta em maiores velocidades de operação. Outros aperfeiçoamentos na tecnologia permitiram reduzir o comprimento do canal (4μm) e a espessura da camada do óxido da região da porta

do dispositivo MOS canal "N", resultando em dispositivos de altíssima performance (H-MOS) e de dimensões reduzidas.

As tecnologias V-MOS e D-MOS são processos alternativos que permitem obter dispositivos MOS de pequeno comprimento de canal e de alta performance. As memórias do tipo C-MOS são construídas com dispositivos MOS canal P e MOS canal N, de modo que um atua como carga do outro.

Em relação à tecnologia bipolar,são fabricadas memórias RAM estáticas, de altíssima velocidade, pela tecnologia TTL schottky,e de maior capacidade, pela tecnologia I^2L. Já as memórias RAM dinâmicas têm sido construídas a partir da tecnologia I^3L.

As memórias semicondutoras que iremos estudar são do tipo memórias de acesso direto. Dentre elas podemos destacar: RAM, ROM, PROM, etc. Os dois tipos de memórias mais utilizados em microcomputadores são as memórias semicondutoras RAM e ROM, disponíveis sob a forma de circuitos integrados com pinos em linha dupla (dual-in-line), as quais contêm centenas ou milhares de elementos armazenadores de bits, denominados - células.

MEMÓRIAS RAM:

A memória RAM ("Randon access memory"), ou memória de acesso aleatório, também conhecida por memória de escrita e leitura, é constituída por uma matriz de centenas de flip-flops difundidos em uma única pastilha de silício. Como já foi dito anteriormente, um flip-flop se resume em um multivibrador que possui dois estados estáveis e, portanto, pode ser utilizado como um elemento armazenador de informações, ou células, de um bit. Assim, por exemplo, um grupo de cem (100) flip-flops pode armazenar até cem (100) bits (níveis lógicos "0" e "1").

Assim como em um flip-flop podemos mudar o seu estado estável, em uma memória RAM podemos alterar as informações ali armazenadas, ou apagá-las, bastando, para tal levar todas as células a um nível lógico "0" ou "1". Da mesma forma, se interrompermos a alimentação de uma RAM, toda a informação nela armazenada é apagada (perdida). Isto se deve ao fato de

que, se logo em seguida reestabelecermos a alimentação, os flip-flops assumirão qualquer estado lógico, ao acaso, descaracterizando, assim, a última informação armazenada antes da interrupção da alimentação.

O termo acesso aleatório significa que podemos ter acesso a qualquer célula da memória, bastando, para isso, selecionarmos, através de um "endereço", a célula que desejamos verificar ou alterar sua informação.

Quanto às memórias RAM de tecnologia bipolar, temos na figura 8 o circuito esquemático de uma célula de memória RAM multiemissor, a qual tem sido muito utilizada.

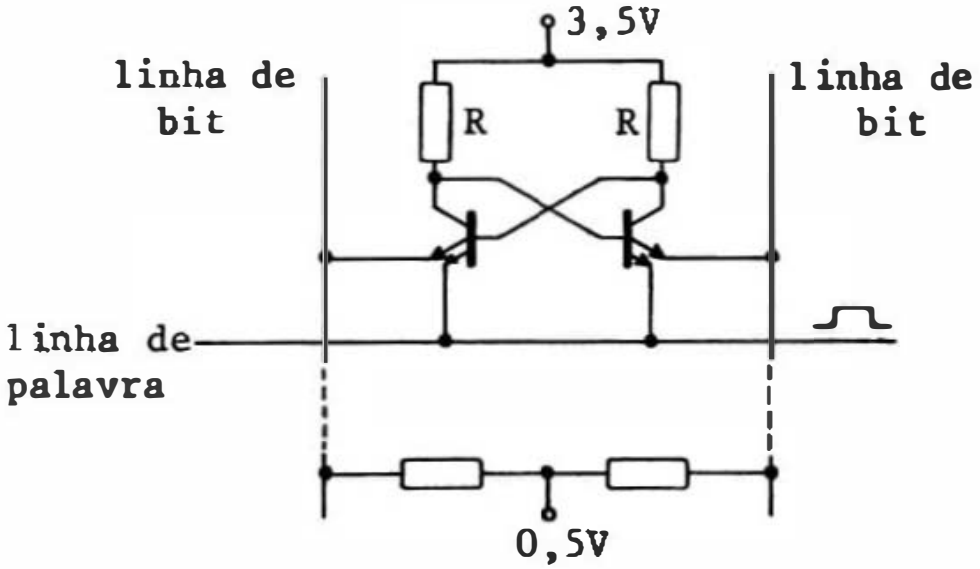


FIGURA 08

Com relação ao circuito da figura 8, se não desejamos ler nem escrever, basta mantermos a linha de palavra a uma baixa tensão (0,3V). Nestas condições, a célula se comporta como um elemento biestável, com um transistor conduzindo e o outro em corte. Ao contrário, para se ler o estado em que se encontra a célula, basta elevarmos a tensão da linha de palavras (3,0V) a qual irá fazer com que uma corrente circule pela linha de bit, corrente essa correspondente ao transistor que está conduzindo. Esta corrente, normalmente, é detectada por um amplificador com impedância relativamente baixa.

Na figura 9, mostramos uma memória de 16 células.

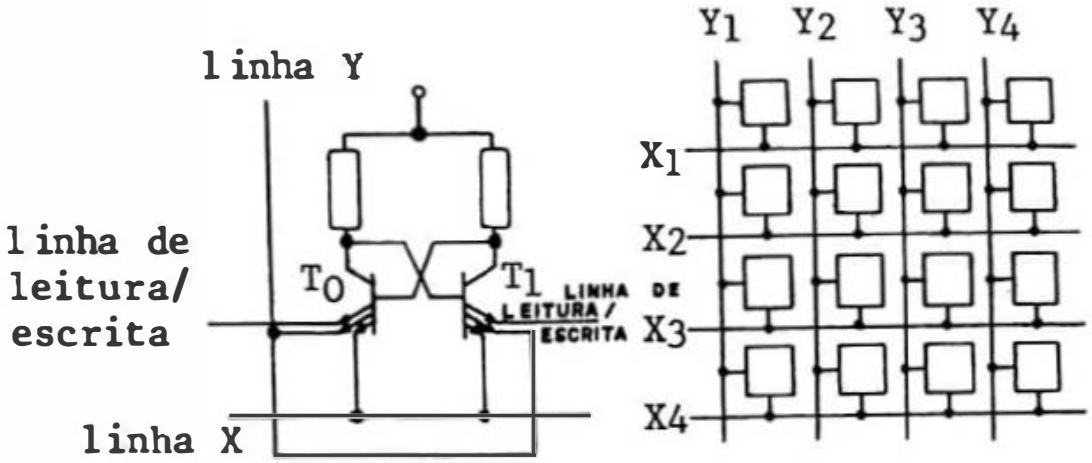


FIGURA 09

Para descrevermos o funcionamento de cada célula, temos que levar em consideração o fato de que em uma porta TTL, construída com transistor multiemissor, somente o emissor conectado ao menor potencial é que conduz; os demais emissores ficam polarizados no sentido inverso.

Na condição de repouso, as linhas de direção (linhas X e Y) se encontram a um nível lógico "0". Já as linhas de leitura/escrita são mantidas a um potencial superior ao potencial correspondente ao nível lógico "0". Com isto, os emissores correspondentes ficam inversamente polarizados e, conseqüentemente, não circulará corrente alguma por essas linhas. Nestas condições podemos dizer que a célula funciona como um biestável.

Suponhamos agora que o transistor T_0 esteja no "corte" e que T_1 esteja "conduzindo". Nesta condição, se enviarmos simultaneamente dois pulsos positivos, um por uma linha X e outro por uma linha Y, verificaremos que dois dos emissores de T_1 correspondentes a tais linhas (X e Y) ficaram inversamente polarizados, enquanto que o outro emissor de T_1, correspondente à linha de leitura/escrita (ou linha de bit) ficará diretamente polarizado e, conseqüentemente, enviará uma corrente através desta linha (linha leitura/escrita). Quanto ao transistor T_0, como já dissemos, se encontra em corte, logo não envia corrente alguma através da linha de leitura/escrita.

Com relação à tecnologia MOS, apresentamos na figura 10 uma unidade básica (uma célula) de uma memória RAM-MOS, constituída por 4 transistores MOS. Esta célula se baseia em um

elemento biestável, onde os transistores T3 e T4 atuam como resistências e os transistores T1 e T2 equivalem aos transistores bipolares do elemento biestável.

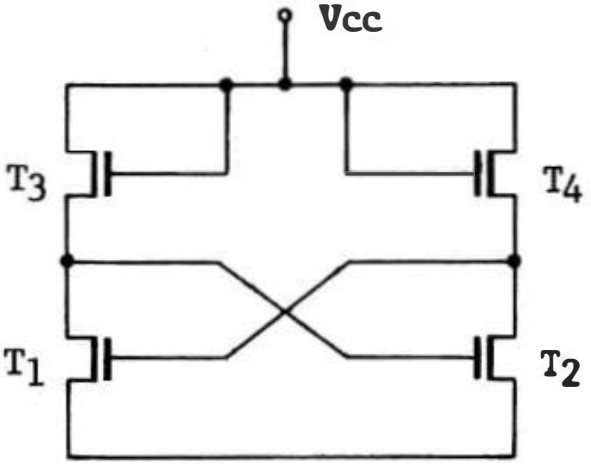


FIGURA 10

Quando o transistor T1 conduz, sua tensão de dreno, de baixo valor, é aplicada na porta ("gate") do transistor T3. Como esta tensão (tensão aplicada na porta de T3) é de valor inferior ao valor necessário de tensão para que T3 conduza, este permanece bloqueando, com uma tensão de dreno praticamente igual à tensão de alimentação Vcc. A tensão de alimentação Vcc, aplicada na porta do transistor T1, faz com que este conduza por um tempo indefinido. Podemos então dizer que o nível lógico "0" equivale ao estado de saída do dreno de T1 e que o nível lógico "1" equivale ao estado de saída do dreno de T3. Logo, podemos concluir que, para esta situação, T1 conduzindo e T3 bloqueado, temos os estados "0" e "1" memorizados nos transistores T1 e T3, respectivamente. Se invertermos a situação, T1 bloqueado e T3 conduzindo, teremos memorizado a informação oposta. Este princípio é válido para memórias RAM-MOS estáticas.

Na figura 11 apresentamos uma célula RAM-MOS estática, -

constituída por oito transistores. Neste caso, os transistores T1, T2, T3 e T4 formam o elemento biestável enquanto que os transistores T5, T6, T7 e T8 são utilizados para a seleção. O funcionamento desta célula é análogo ao das células - RAM bipolares, onde X e Y são as linhas de seleção (ou linhas de endereçamento).

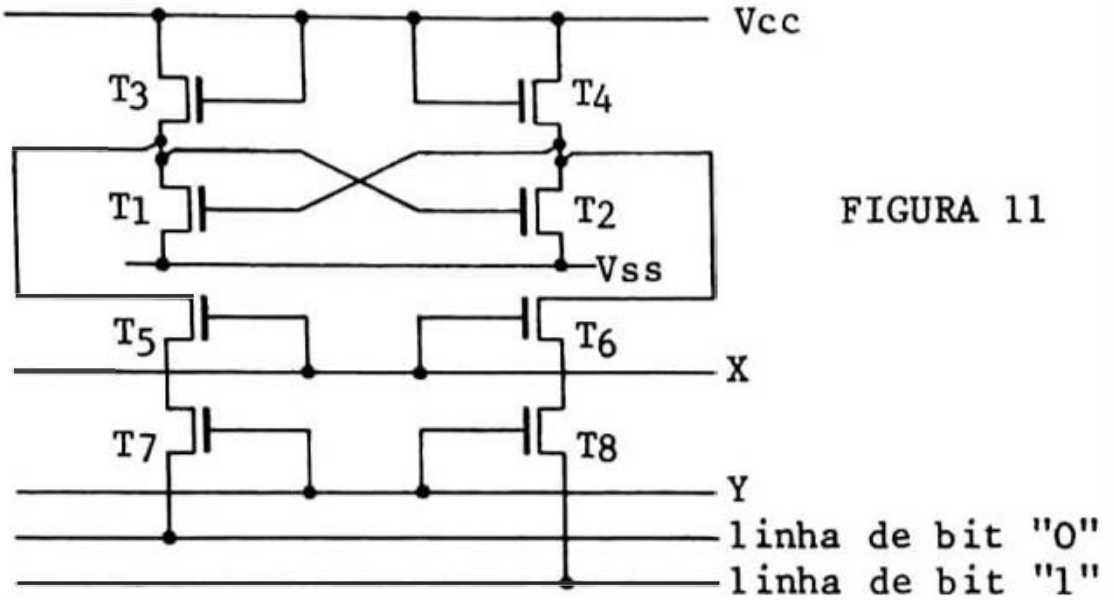
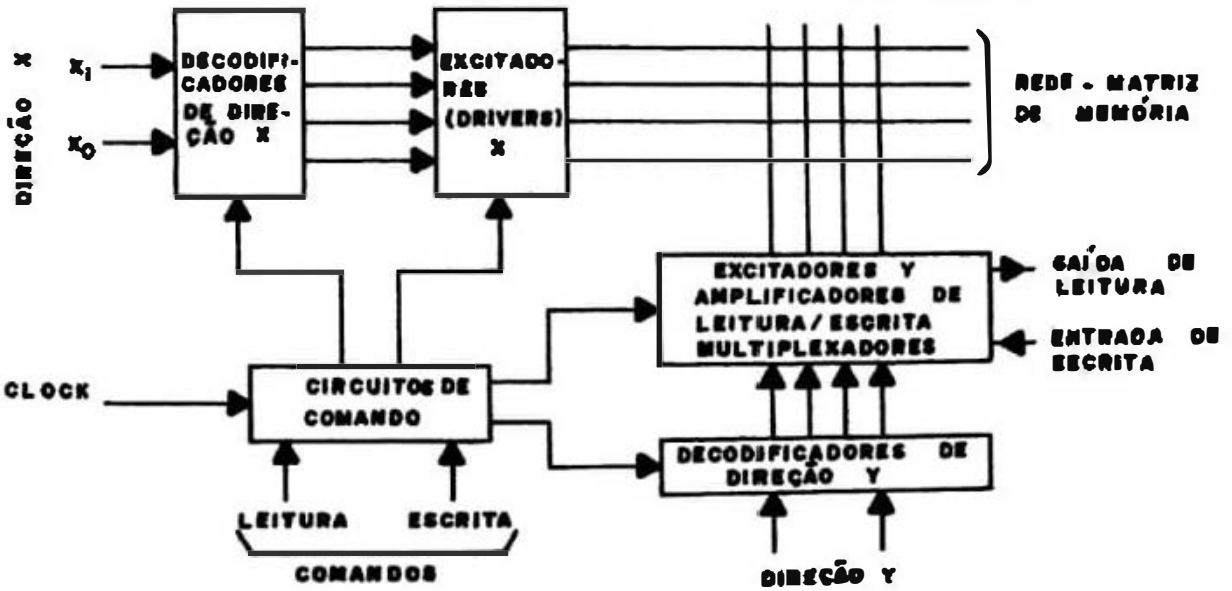


FIGURA 11

Estas células podem ser agrupadas em redes para constituírem uma matriz-memória completa. Na figura 12 apresentamos o diagrama de blocos de uma estrutura de memória completa, do tipo RAM-MOS estática.

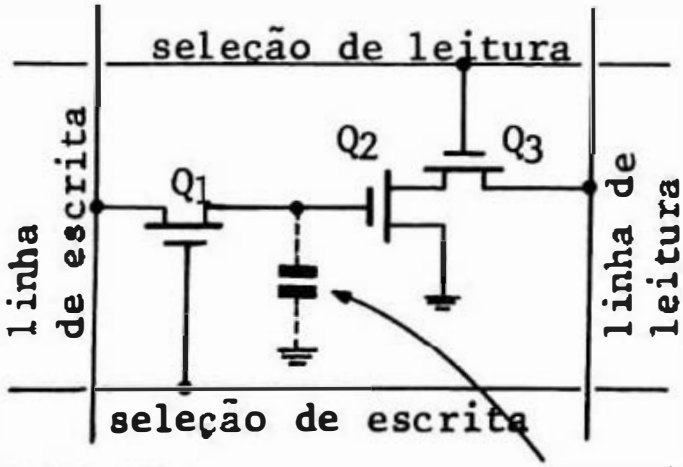
FIGURA 12



As memórias RAM-MOS dinâmicas dispensam os biestáveis, porém se utilizam de uma característica própria dos circuitos MOS, ou seja, de capacitância existente entre a porta (G) e o substrato. Esta pode ser carregada como um capacitor comum, o qual permanecerá carregado por muito tempo, devido à alta impedância de entrada dos transistores MOS. Normalmente esta carga permanece praticamente inalterada durante vários milessegundos, o que permite a utilização dessa capacitância como memória.

A descarga do capacitor ocorre entre 5 e 10 milessegun - dos, porém por motivos de segurança, a carga é completada periodicamente a cada 2 milessegundos como maneira de se compensar as perdas, devido a descargas havidas sob a forma de corrente de fuga.

Na figura 13, apresentamos o circuito de uma célula de memória MOS dinâmica, constituída por uma simples rede de três (3) transistores MOS.



célula RAM dinâmica 3 MOS

capacidade porta-substrato de Q_2

FIGURA 13

Neste caso, o primeiro transistor (T1) é responsável pelo direcionamento (seleção) da célula. Logo, para realizarmos uma gravação, é necessário fazermos com que o transistor T1 - conduza, bastando para tanto aplicarmos uma tensão em sua

porta (G), através do ponto A. O "capacitor" existente entre a porta e o substrato de T2 se carregará segundo a informação presente na linha de escrita.

Para realizarmos a leitura da informação gravada, devemos aplicar uma tensão na porta (G) de T3, através do ponto B, fazendo com que ele conduza. Logo em seguida, aparecerá uma corrente na linha de leitura, devido à condução de T2, quando o capacitor C está carregado (nível lógico 1). Se o capacitor C estiver descarregado (nível lógico 0), T2 não conduzirá e, conseqüentemente, não teremos corrente circulando através da linha de leitura.

Através da integração de capacidades (capacitâncias) suplementares ou simplesmente aumentando as já existentes, podemos construir uma célula de memória MOS-dinâmica com apenas transistor MOS (veja figura 14).

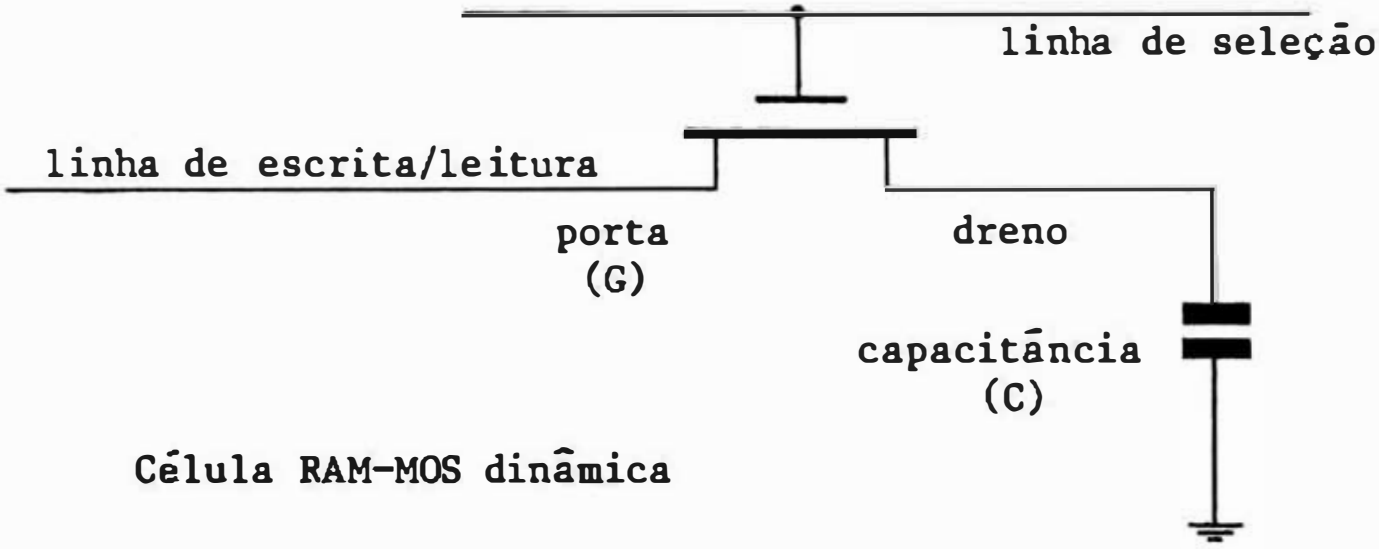


FIGURA 14

Esta estrutura RAM-MOS dinâmica, com um só transistor MOS, é muito comum em vários tipos de memórias de 4 K bits (4096 bits), com grande possibilidade de uso nas memórias de 16 K bits.

Na figura 15, mostramos uma célula de memória RAM, construída com base na tecnologia COS-MOS.

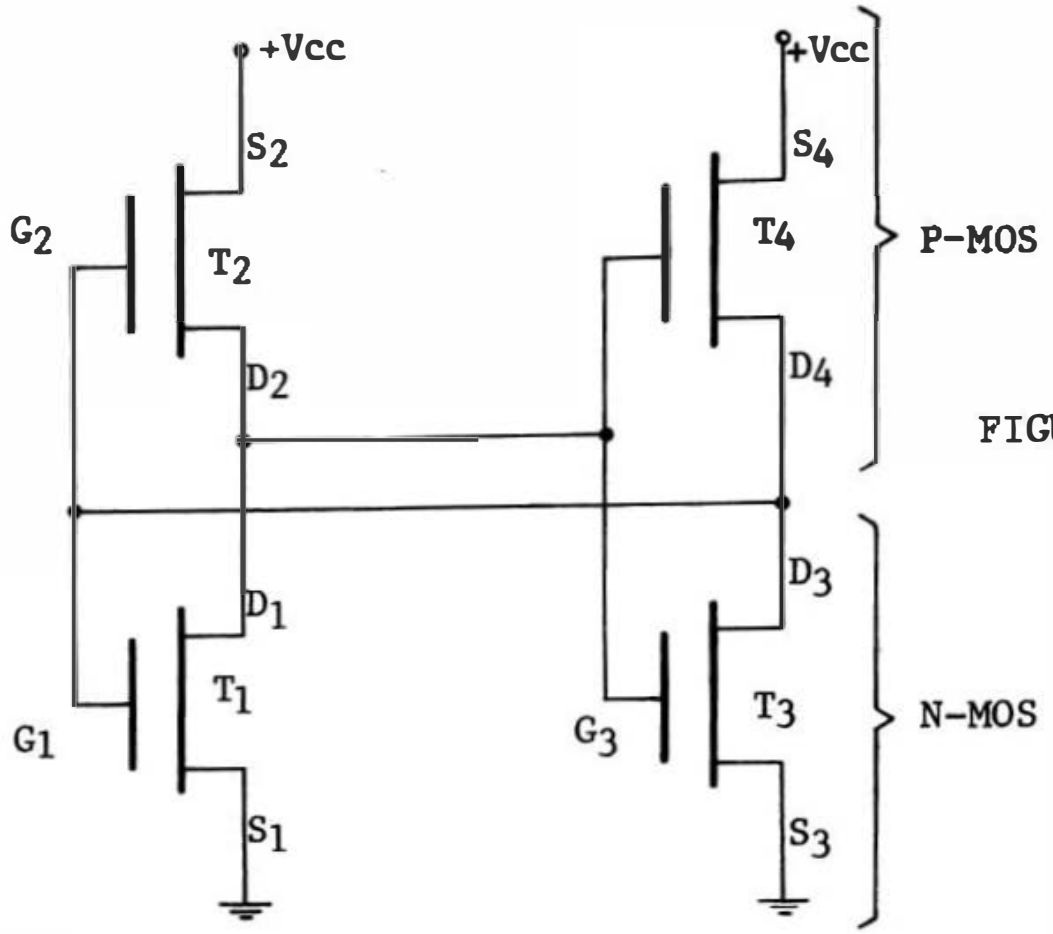


FIGURA 15

APLICAÇÃO DAS MEMÓRIAS RAM:

Normalmente, utilizam-se memórias RAM estáticas quando a capacidade de memória exigida, por exemplo, for menor que 32 K bits (4K octetos, onde 1 octeto é igual a 8 bits). Quando a capacidade de memória exigida for superior a 4K octetos (32 K bits), torna-se mais econômico utilizar memórias RAM do tipo dinâmica.

Por outro lado, as memórias RAM, de grande capacidade de armazenamento, geralmente são utilizadas em conjunto com microprocessadores, com a finalidade de armazenarem informações temporárias, dados de memórias ROM, dados de periféricos, tais como impressoras, leitoras de cartões e fitas, etc. Os microprocessadores também podem "guardar" nas memórias RAM as informações que serão transferidas, posteriormente, para os periféricos.

Concluindo, podemos dizer que as memórias RAM são utilizadas para armazenarem dados e programas que requerem constantes mudanças.

Na tabela 1, procuraremos, de maneira resumida, estabelecer uma pequena comparação entre as memórias RAM de núcleo magnético e as memórias RAM de circuitos integrados.

TABELA 01

CARACTERÍSTICAS	NÚCLEOS	CI	
		BIPOLAR	MOS
VOLATILIDADE	não	sim	sim
SENSIBILIDADE A RADIAÇÕES IONIZANTES	não	sim	sim
LEITURA DESTRUTIVA	sim	não	não
CONSUMO POR BIT (mW)	1	10 a 1	0,1
TEMPO DE CICLO (ns)	500 a 1.000	10 a 200	100 a 1.000

A tabela 2 a seguir nos mostra uma série de memórias RAM de circuitos integrados do tipo bipolar.

TIPO	DESCRIÇÃO	CONFIGURAÇÃO	SAÍDA	MÁXIMO Taa(ns)	FAIXA DE TEMPERATURA
82S21	64-bit-RAM	32x2	OC	50	C
82S25	64-bit-RAM	16x4	OC	50	Mc
54/74S89	64-bit-RAM	16x4	OC	50	Mc
54/74S189	64-bit-RAM	16x4	TS	35	Mc
3101A	64-bit-RAM	16x4	OC	35	Mc
82S16	256-bit-RAM	256x1	TS	50	Mc
82S17	256-bit-RAM	256x1	OC	50	Mc
82S116	256-bit-RAM	256x1	TS	40	C
82S117	256-bit-RAM	256x1	OC	40	C

TABELA 02

TIPO	DESCRIÇÃO	CONFIGURAÇÃO	SAÍDA	MÁXIMO Taa(ns)	FAIXA DE TEMPERATURA
54/74S200	256-bit-RAM	256x1	TS	50	Mc
54/74S201	256-bit-RAM	256x1	TS	50	Mc
54/74S301	256-bit-RAM	256x1	OC	50	Mc
82S09	576-bit-RAM	64x9	OC	45	Mc
10144	256-bit-RAM	256x1	OE	30	Mc
82S10	1024-bit-RAM	1024x1	OC	45	Mc
82S11	1024-bit-RAM	1024x1	TS	45	Mc
93415A	1024-bit-RAM	1024x1	OC	45	Mc
93425A	1024-bit-RAM	1024x1	TS	45	Mc

Na primeira coluna da tabela 2 temos o tipo de memória e o código do fabricante. Na segunda coluna temos a capacidade da memória em bits, por exemplo: 64 bits RAM.

Na terceira coluna aparece o número de bytes em que é formado o comprimento da memória, por exemplo: 32, 16, 64, etc. E, em seguida, ainda na mesma coluna, temos o número de bits de possuir um byte da respectiva memória, por exemplo: 2, 4, 1, 9, etc...

A quarta coluna nos mostra a especificação do tipo de saída utilizada em cada uma das memórias relacionadas. Temos três tipos básicos de saída:

a) saída em tri state - TS

b) saída em coletor aberto (open collector) - OC

c) saída em emissor aberto (open emitter) - OE

A quinta coluna mos mostra o fator de velocidade máxima para a temperatura ambiente (Taa), fornecida em nanosegundos (tempo).

E, para finalizar, a sexta coluna nos mostra a faixa de temperatura em que a memória pode trabalhar.

C - comercial "0ºC até 75ºC";
M - militar "-55ºC até 125ºC".

SAÍDA EM TRI STATE:

Utilizados nas saídas das memórias, esses circuitos (tri state) permitem colocar um número infinito de memórias em paralelo, ou seja, é o mesmo que várias chaves conectadas ao mesmo fio sendo que somente uma delas é acionada de cada vez, transportando a informação correspondente à entrada conectada. Esse método de comutação é conhecido pelo nome "time sharing". Na figura 16 temos um esquema deste tipo de ligação; note que somente a informação 3 chega ao circuito receptor.

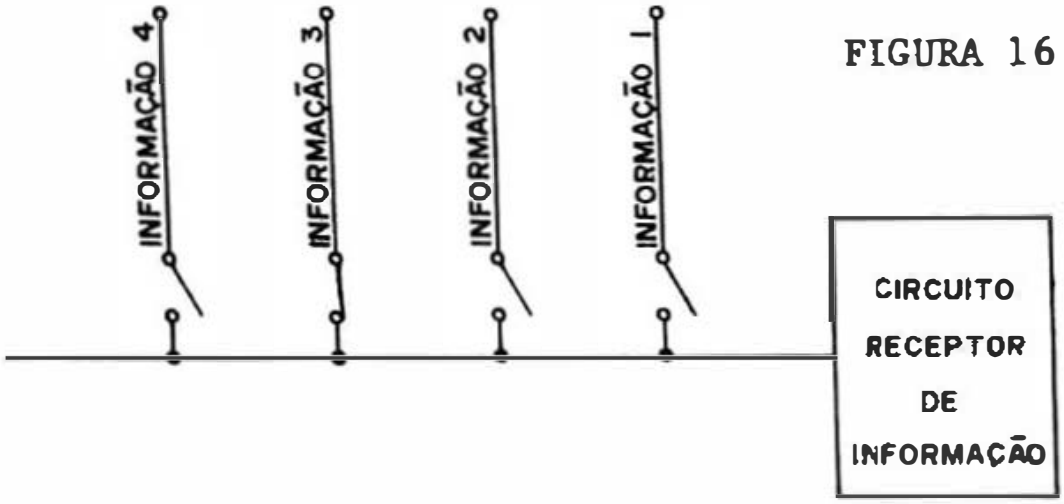
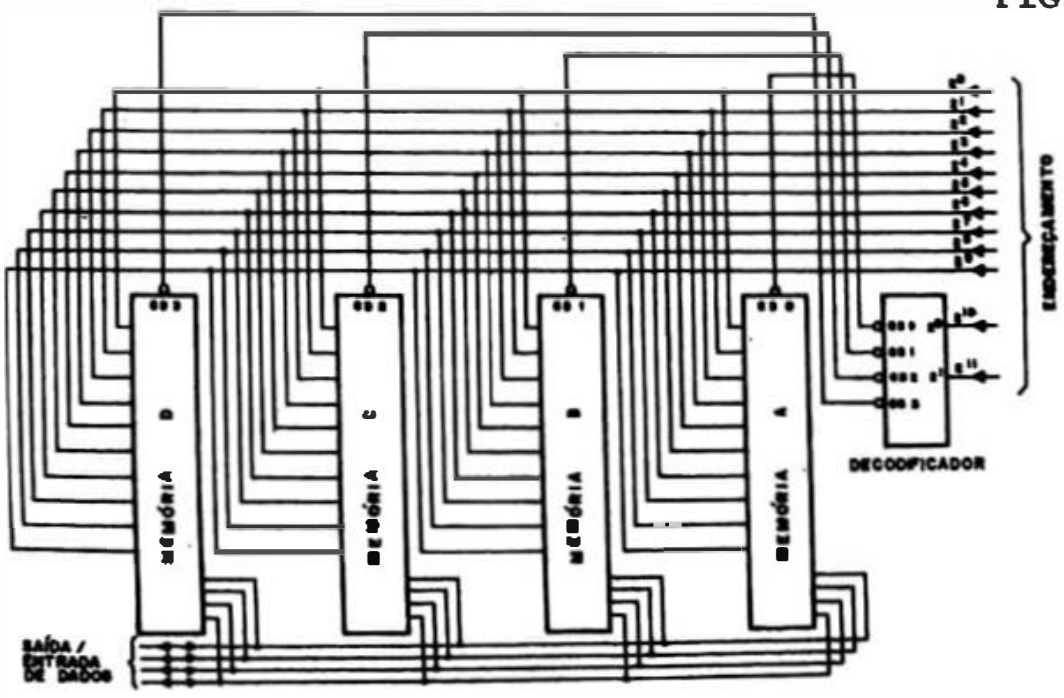


FIGURA 16

Com o auxílio das memórias de saída tri state é possível montarmos um banco de memória de capacidade quase que infinita, limitado apenas pelo circuito endereçador, pois quanto maior for o número de memórias, maior é a quantidade de fios endereçadores. Na figura 17 mostramos um exemplo de um banco de memória de 4 K bits (4096 bits x 4 bits).

FIGURA 17



CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-5

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) O funcionamento básico de uma memória CCD é idêntico ao funcionamento:

a) dos registradores de deslocamento;
b) das memórias de núcleo magnético;
c) das memórias ROM;
d) das memórias PROM.

2) Uma das principais vantagens da tecnologia CCD é:

a) a necessidade de vários tipos de difusão seletiva;
b) a não necessidade de nenhum tipo de difusão seletiva;
c) a utilização de amplificadores para o processo de leitura;
d) N.R.A.

3) A tecnologia de fabricação das memórias de bolhas magnéticas (MBM) é similar à tecnologia:

a) SOS;

b) LMI;

c) LSI;

d) N.R.A.

4) O diâmetro das bolhas magnéticas pode chegar a um valor em torno de:

a) $0,5 \times 10^{-3}$ m;

b) 500×10^{-6} m;

c) 500×10^{-3} m;

d) $0,5 \times 10^{-6}$ m ou 0,5 micro.

5) Sobre a superfície da camada epitaxial (de uma memória de bolha) depositam-se algumas porções de:

a) permaloi;

b) magnetita;

c) cobre;

d) cerâmica magnética.

6) Os movimentos das bolhas no interior da camada epitaxial se conseguem através da aplicação de:

a) uma tensão superior a 50 volts;

b) um campo elétrico constante;

c) uma corrente de polarização;

d) um campo magnético giratório em sentido transversal.

7) As trajetórias normalmente executadas pelas bolhas magnéticas são trajetórias:

a) fechadas, denominadas "laços menores"; (loop)

b) abertas;

c) semi-abertas;

d) N.R.A.

8) Cada laço menor de uma MBM corresponde a:

a) domínio;

b) 1 bit;

c) ciclo completo;

d) replicador.

9) Os elementos armazenadores de bit recebem o nome de:

a) dual-in-line;

b) pastilha;

c) células;

d) N.R.A.

10) Quando a capacidade de memória exigida for superior a 32 K bits, torna-se mais econômico utilizar memórias RAM do tipo:

a) estática;

b) dinâmica;

c) acesso seqüêncial;

d) N.R.A.

CEAM
CURSOS DE APERFEIÇOAMENTO

CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-6

MEMÓRIAS IV

INTRODUÇÃO

Dando continuidade ao estudo das memórias semicondutoras, apresentaremos nesta lição as memórias ROM, PROM, EPROM, EEPROM, EAROM, etc. Vamos então dar início ao nosso estudo apresentado as memórias ROM.

MEMÓRIAS ROM

A memória ROM (read only memory), ou memória de apenas leitura, é um circuito eletrônico utilizado para armazenar, permanentemente, informações binárias. A principal característica deste tipo de memória é o fato de que a informação binária nela contida está permanentemente gravada.

A organização básica das memórias ROM é praticamente idêntica à das memórias RAM, uma vez que ambas possuem um certo número de localizações de memória onde as informações podem ser armazenadas. Cada localização corresponde a um ponto

de memória (ou célula elementar). As informações gravadas nas memórias ROM podem ser lidas em qualquer ordem. Na figura 1, temos um diagrama de blocos, geral, de uma memória ROM, o qual é formado por três seções principais: o decodificador de endereços, os elementos de armazenamento da memória (conjunto de pontos de memória ou células) e os circuitos de saída.

O decodificador de endereços opera como qualquer outro decodificador binário, ou seja, aceita uma palavra binária de múltiplos bits de entrada e a decodifica de modo a "cobrir" todos os possíveis estados de entrada. Neste processo, apenas uma das saídas decodificadas será ativada. Na figura 1, o decodificador de endereços possui 5 bits de entrada o que corresponde a 32 (2^5) diferentes estados que podem ser decodificados.

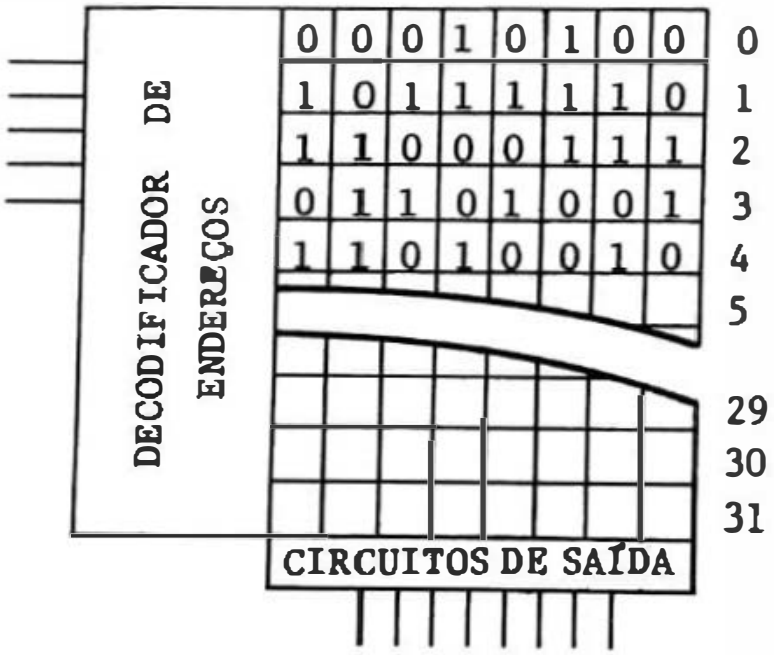


FIGURA 01

A palavra de entrada, geralmente, é conhecida por "endereço".

O corpo principal da memória é formado por pontos de memória (ou células), utilizados para acumular os dados binários. Estes elementos de memória são arranjados de modo que um número específico de palavras de múltiplos bits possa ser

armazenado. Por exemplo, o diagrama de blocos apresentado na figura 1 permite que 32 palavras de oito bits sejam armazenadas. As localizações da memória são designadas de 0 a 31. Ao aplicarmos um código de endereço de cinco bits à entrada, este deverá fazer com que o conteúdo do local endereçado apareça na saída.

Por último, temos os circuitos de saída, cuja função é acoplar o conteúdo da memória para que os dados possam ser usados em outros circuitos lógicos.

As memórias ROM, de aplicações específicas, normalmente são programadas,na fábrica,para determinadas funções. Trazem consigo programas-padrões, ou instruções fixas para microprocessadores. Essas memórias não empregam flip-flops (caso contrário, também seriam voláteis), porém utilizam conjuntos de diodos e/ou transistores, os quais formam uma matriz semelhante à das memórias RAM.

Existem muitas maneiras diferentes de se implementar memórias do tipo ROM, com componentes eletrônicos. Qualquer componente ou circuito capaz de armazenar um binário na condição "0" ou "1" pode ser utilizado. Os núcleos magnéticos e os capacitores são exemplos de elementos que vêm sendo utilizados para armazenar dados binários em uma memória ROM. No entanto, as memórias ROM mais modernas são construídas de acordo com as técnicas dos dispositivos semicondutores. Como sabemos, duas técnicas são amplamente utilizadas: Bipolar e MOS. Devido ao fato das memórias ROM poderem armazenar uma quantidade significativa de dados, elas são, geralmente, classificadas como circuitos integrados em larga escala (LSI). A maioria das memórias ROM é embalada em cápsulas, com pinos em linha dupla (Dip ou dual-in-line).

Vejamos então os tipos mais populares de circuitos integrados ROM, em uso atualmente.

MEMÓRIAS ROM A DIODOS

Na figura 2, podemos observar parte de uma matriz de memória ROM a diodos, com as informações binárias nela contidas.

Por exemplo, se selecionarmos a linha A e nela aplicarmos uma tensão positiva, verificaremos que nas colunas 2 e 4 não teremos nenhum sinal, porém nas colunas 1 e 3 detectare-

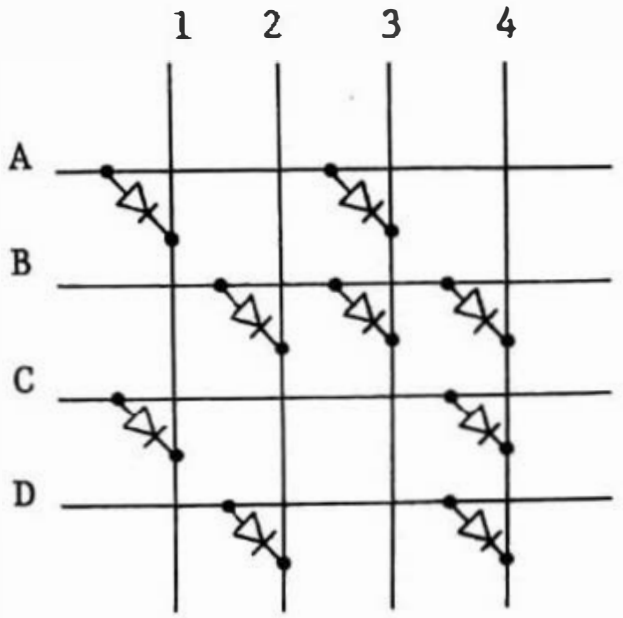


FIGURA 02

PARTE DE UMA MATRIZ DE MEMÓRIA ROM A DIODOS.

mos a presença da tensão.

Na figura 3, também apresentamos uma parte de uma matriz de memória ROM a diodos. Neste caso, o direcionamento é feito através dos interruptores (I_1, I2, I3 e I4). Observando a figura 3 podemos notar que o interruptor 3 (I3) está fechado, - unindo a sua coluna à massa. Nestas condições, toda vez que tivermos um diodo no ponto de cruzamento da linha com a coluna, a tensão de saída desta linha, quando o interruptor da coluna que está cruzando a linha em questão estiver fechado, será nula. No caso, como o interruptor I3 está fechado, a tensão de saída referente às linhas A e D serão nulas, pois os diodos que se encontram nos respectivos cruzamentos estarão conduzindo. Entretanto, a tensão de saída será positiva e igual a +Vcc se não houver diodo entre o cruzamento da linha com a coluna em questão. Assim, a palavra de saída lida será: 0110. Os diodos conduzem através de seus resistores limitadores (Ra; Rb; Rc e Rd).

Podemos então concluir que cada ponto de memória é um elemento unidirecional, por exemplo, o diodo, suprimido para a informação inversa.

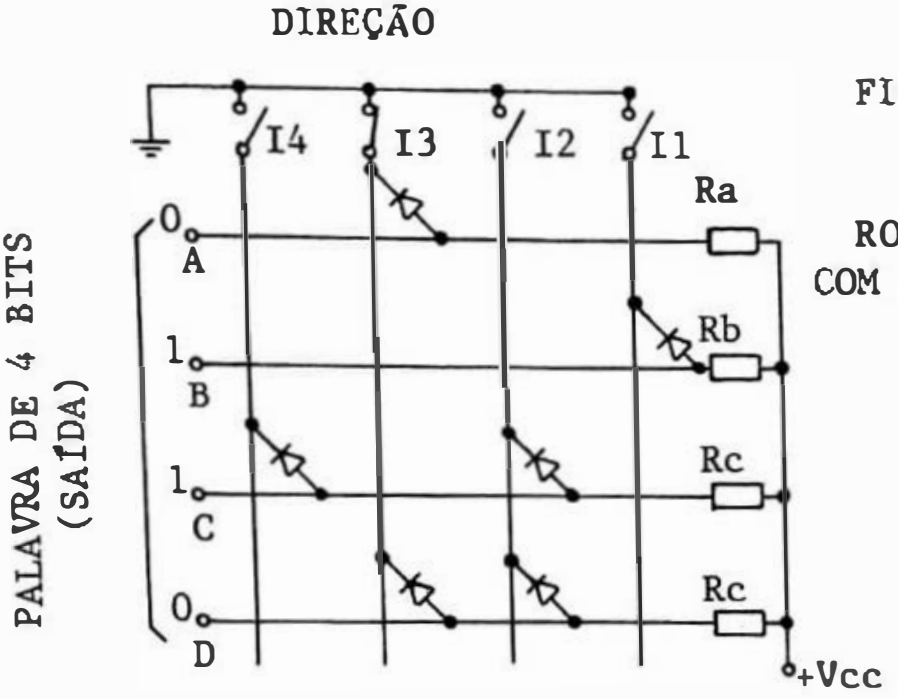


FIGURA 03

ROM A DIODOS
COM 4 PALAVRAS DE
4 BITS

Na figura 4 apresentamos um esquema de uma memória ROM capaz de armazenar 4 palavras de 4 bits, cujo direcionamento é feito mediante dois bits. É ativada uma só coluna de cada vez, em função da direção dada pelo decodificador de direção.

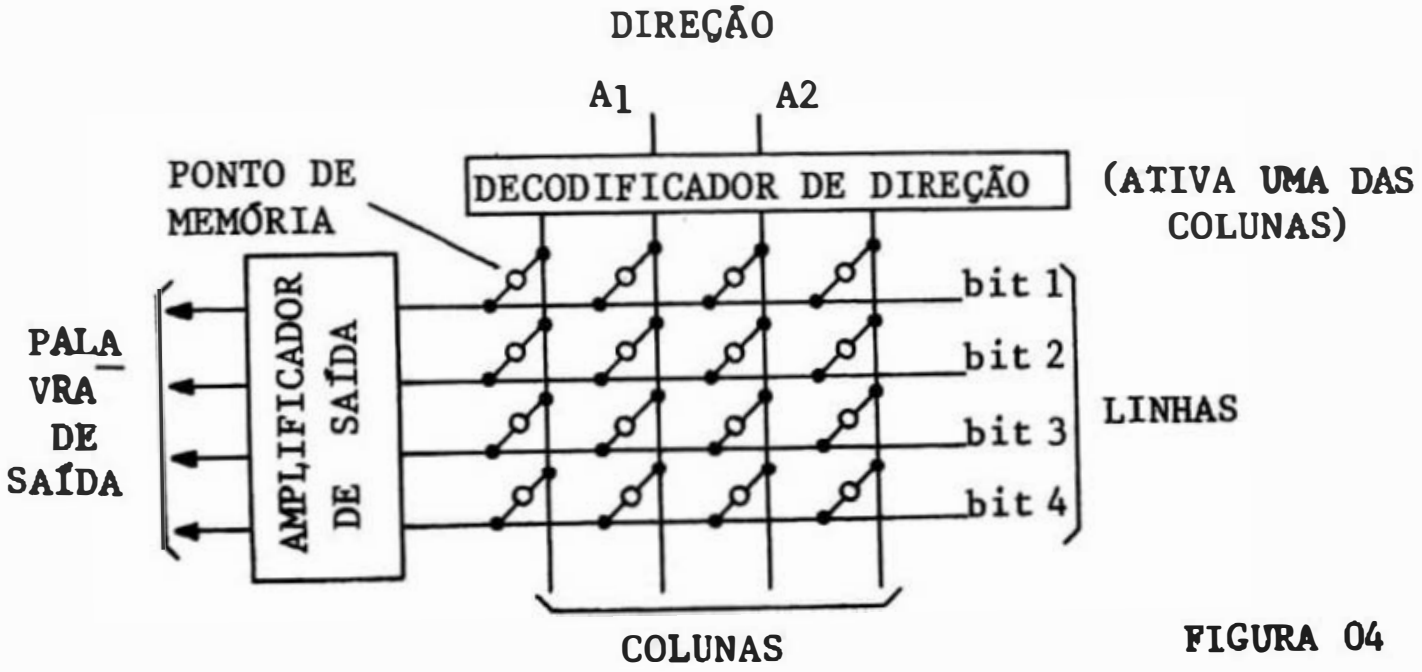


FIGURA 04

A figura 5 nos mostra uma memória ROM de dezesseis palavras de quatro bits cada uma. Como podemos notar, apenas uma linha da matriz será selecionada para um dado endereço.

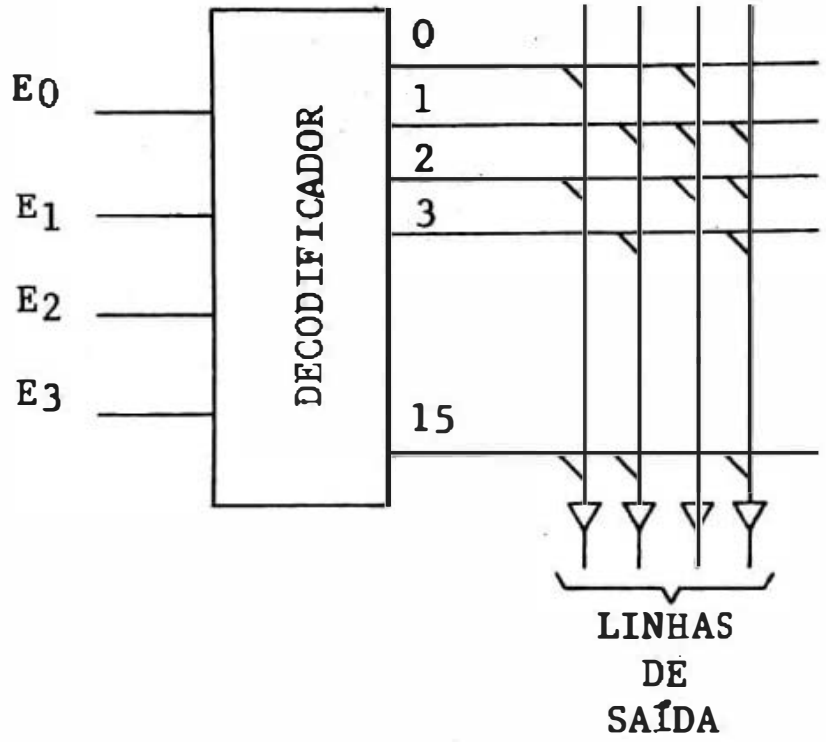


FIGURA 05

ROM DE 16 PALAVRAS DE 4 BITS

ROM BIPOLAR

Na figura 6 temos representada uma típica memória ROM comercial de 256 bits, a qual utiliza circuitos TTL. Esta memória está organizada em 32 palavras de oito bits. As entradas de endereços são cinco (indicadas pelas letras A, B, C, D e E), utilizadas para a seleção de uma das 32 palavras armazenadas na memória.

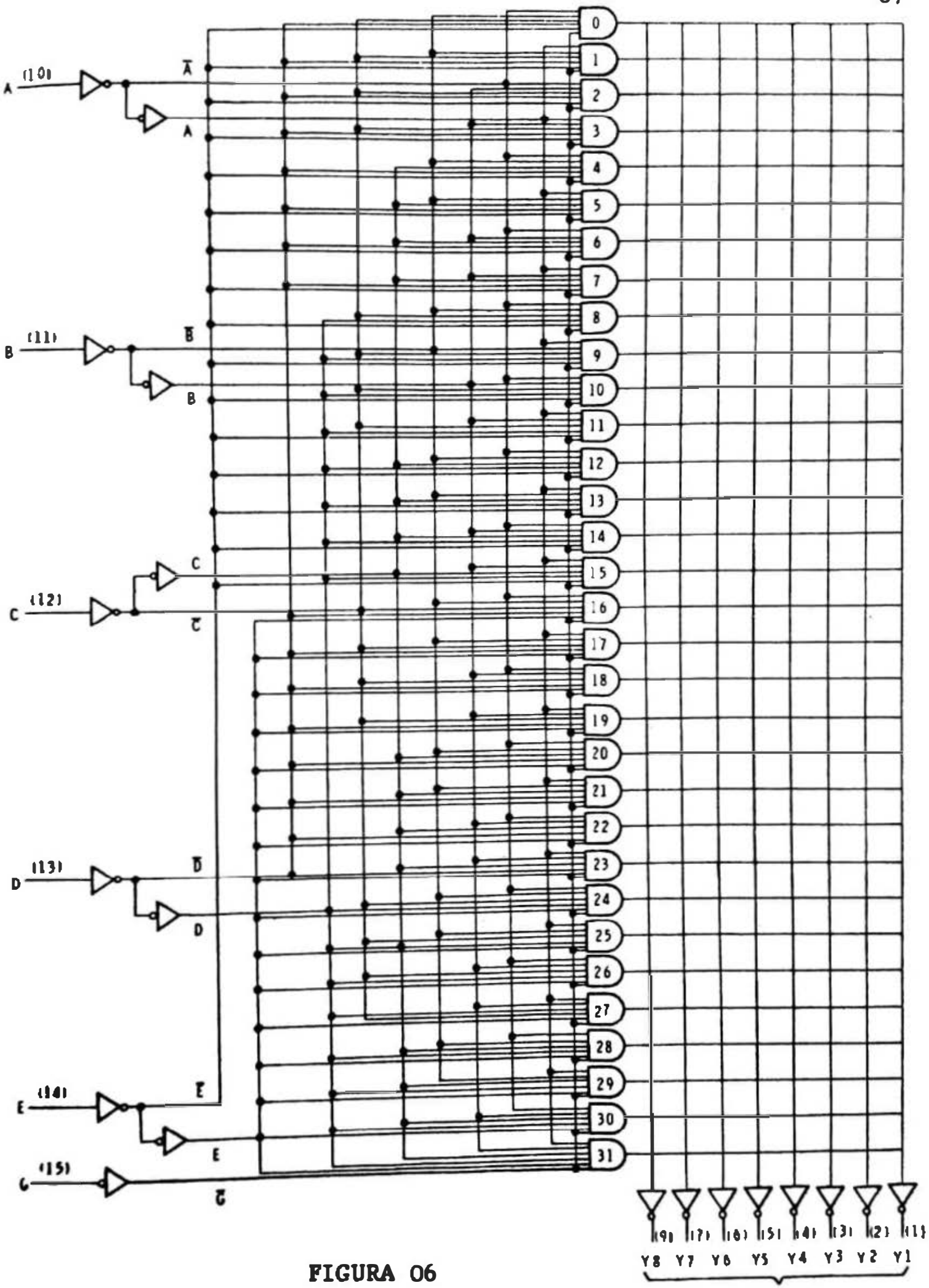
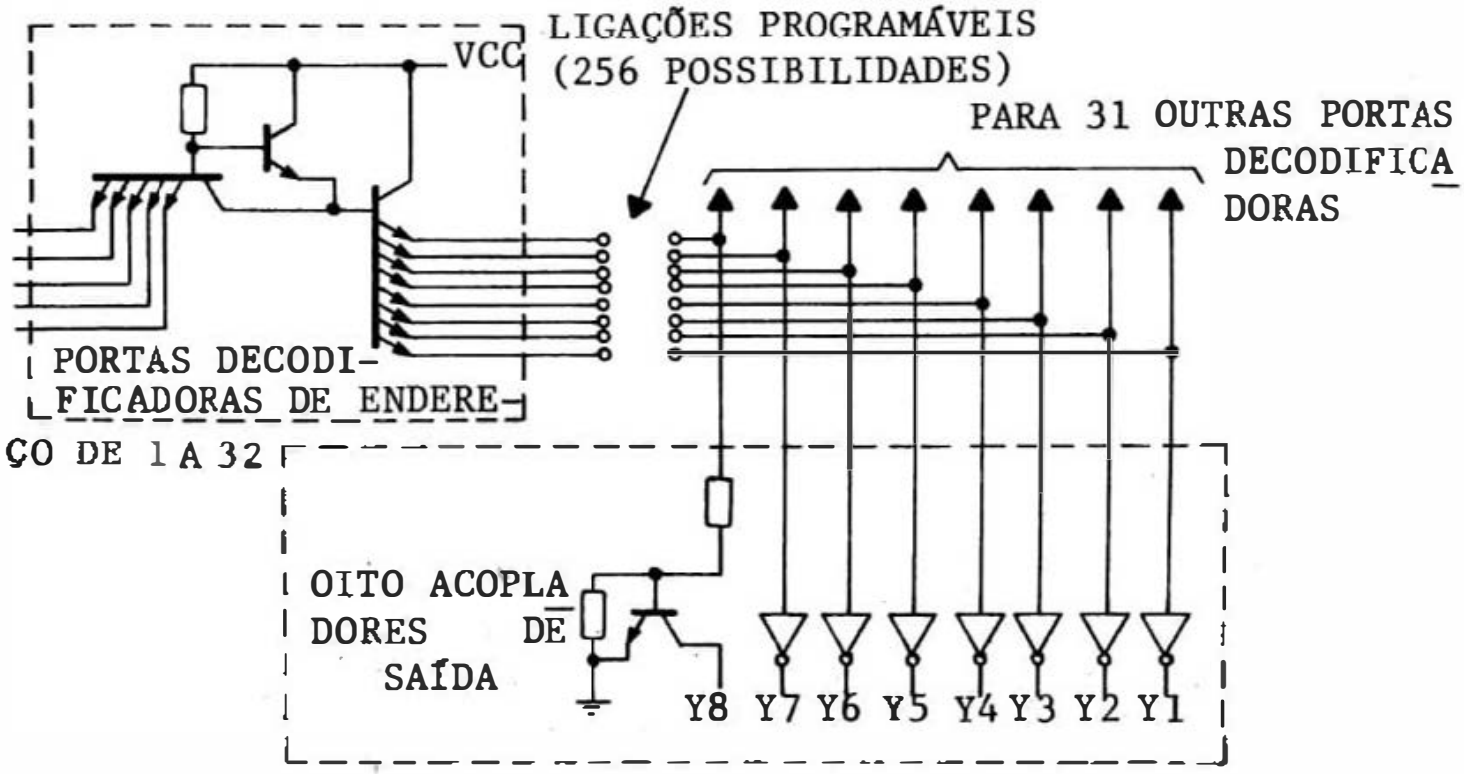

A
(10)
$\overline{A}$
A
B
(11)
$\overline{B}$
B
C
(12)
C
$\overline{C}$
D
(13)
$\overline{D}$
D
E
(14)
$\overline{E}$
E
G
(15)
$\overline{G}$
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
(9)
(7)
(6)
(5)
(4)
(3)
(2)
(1)
Y8 Y7 Y6 Y5 Y4 Y3 Y2 Y1
SAÍDA

FIGURA 06

Na figura 7, mostramos uma das 32 portas decodificadoras de endereço e os oito circuitos acopladores de saída. A saída de cada porta decodificadora é um transistor multiemissor,com precisamente oito (8) emissores, os quais são interligados aos oito (8) acopladores de saída. Neste caso, a programação da memória é feita conectando ou não conectando os emissores aos acopladores de saída. Por exemplo, se um emissor for co nectado a um acoplador de saída, a tensão de saída irá para

FIGURA 07

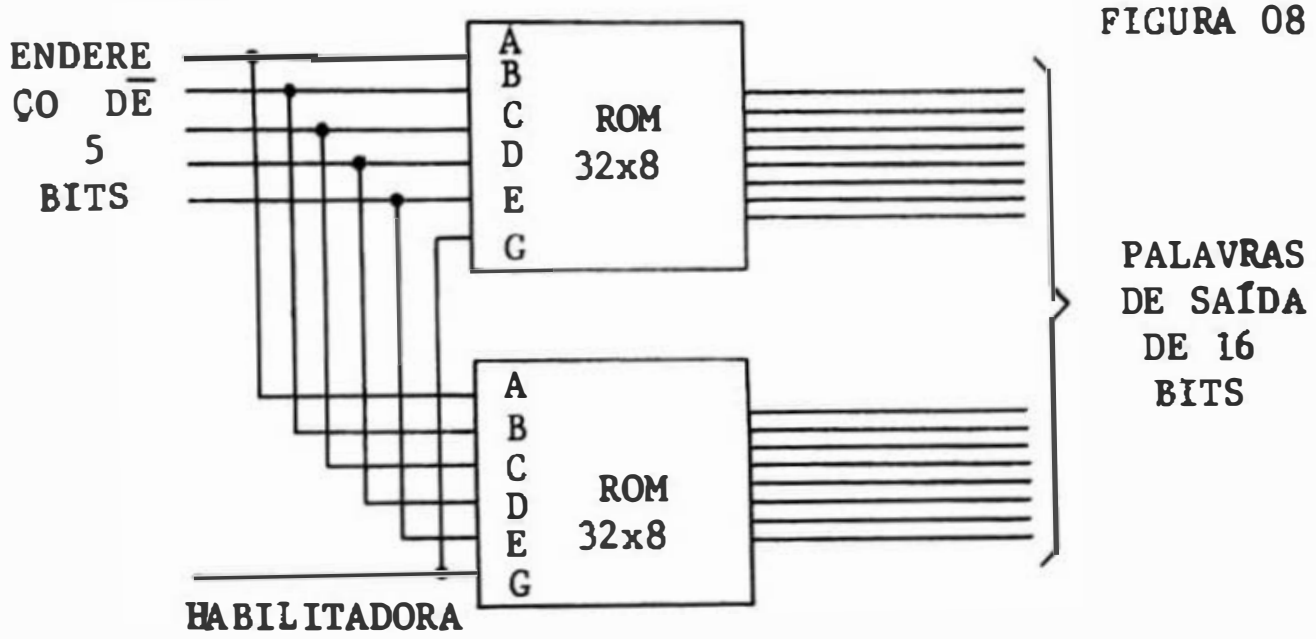


zero ("0") quando aquela porta decoficiadora for endereçada. Caso contrário, se o emissor não for conectado, quando essa mesma porta decodificadora for endereçada, o acoplador irá ler um nível lógico alto de tensão. A linha G, indicada na figura 6, é utilizada para liberar ou inibir o circuito, de maneira que este dispositivo possa ser combinado com outros similares na formação de um "banco de memórias". Essa linha (G) geralmente é referida como linha de seleção do integrado e é utilizada como uma entrada extra de bit de endereçamento

em memórias expandidas.

Na figura 8 temos duas memórias ROM de 32 palavras de 8 bits cada. A forma com que elas foram interligadas nos permite utilizar o circuito (o conjunto) como uma memória única, de 32 palavras, porém de 16 bits. Note que as cinco linhas de endereçamento estão ligadas em paralelo. Assim cada ROM é liberada ao mesmo tempo. Desta forma, metade da palavra de 16 bits é armazenada na memória ROM superior, enquanto que a outra metade (os outros 8 bits restantes) é armazenada na memória ROM inferior. Uma vez que ambas as ROMs são endereçadas ao mesmo tempo, as duas deverão ser lidas, também, ao mesmo tempo. A linha G (entrada) irá habilitar a memória quando estiver em "0".

FIGURA 08



Já, na figura 9, mostramos como duas ROMs de 32 palavras de 8 bits cada uma,podem ser interligadas, para formar uma única memória ROM, capaz de armazenar 64 palavras de 8 bits cada uma.

De acordo com o que podemos observar na figura 9, 32 das palavras estão na ROM superior, enquanto que as outras 32 palavras estão na ROM inferior. Note que as cinco linhas de endereçamento, de A até E, estão ligadas em paralelo. Neste caso as linhas habilitadoras (linha G) dos integrados estão sendo usadas como uma sexta linha de endereçamento, pois são necessários seis bits para endereçar 64 palavras ($2^6 = 64$). Esta sexta linha corresponde ao dígito menos significativo(MSB) do endereço. Os cinco bits (de menor ordem, 1º, 2º, 3º, 4º e

5º) endereçam, simultaneamente, as duas memórias ROM, porém apenas uma das duas ROMs será liberada pela entrada G (MSB).

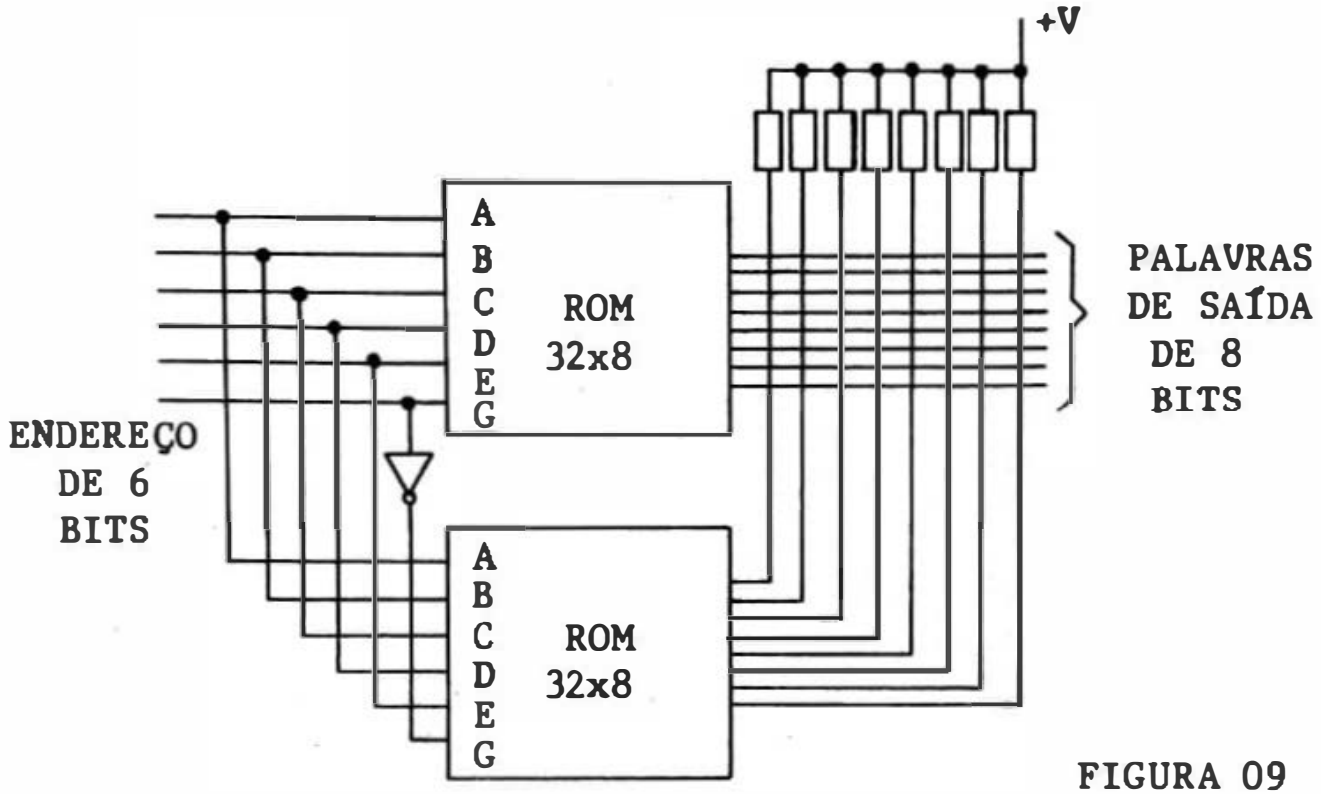


FIGURA 09

A porta inversora interligada entre as duas entradas G das memórias ROM mantém essas linhas complementares.

Suponhamos que seja aplicada, na entrada de endereçamento, a palavra (GEDCBA) (101101). Nestas condições, a localização 13 (01101), em cada memória, será endereçada. Porém, a entrada G é 1 binário. Isto faz com que a ROM superior seja desabilitada, levando todas as suas linhas de saída para um nível lógico alto. Em contrapartida, devido à porta inversora, a ROM inferior será liberada e a palavra de localização 01101 será lida.

ROM MOS

Muitas memórias ROM são implementadas com circuitos integrados de metal-óxido-semicondutor (MOS). O pequeno tamanho da maioria dos circuitos MOSFET tornou possível a construção de muitos elementos de memória e circuitos lógicos, em um pequeno espaço. Esta alta densidade de circuitos permite que memórias ROM com conteúdo de bits muito elevado sejam facilmente fabricadas. Aliada a essa alta densidade de integração, temos um baixo consumo de potência e um baixo custo de

fabricação.

Nas memórias ROM MOS, o elemento básico de armazenamento é um MOSFET de modo de crescimento. Assim, a presença de um MOSFET programa, na memória, um "1" binário. Já, a ausência deste dispositivo significa que um "0" binário foi programa - do.

A organização básica, bem como a estrutura de uma ROM MOS, é essencialmente a mesma de qualquer outra memória ROM. Na figura 10 temos uma ROM MOS de 4 palavras de 4 bits cada uma.

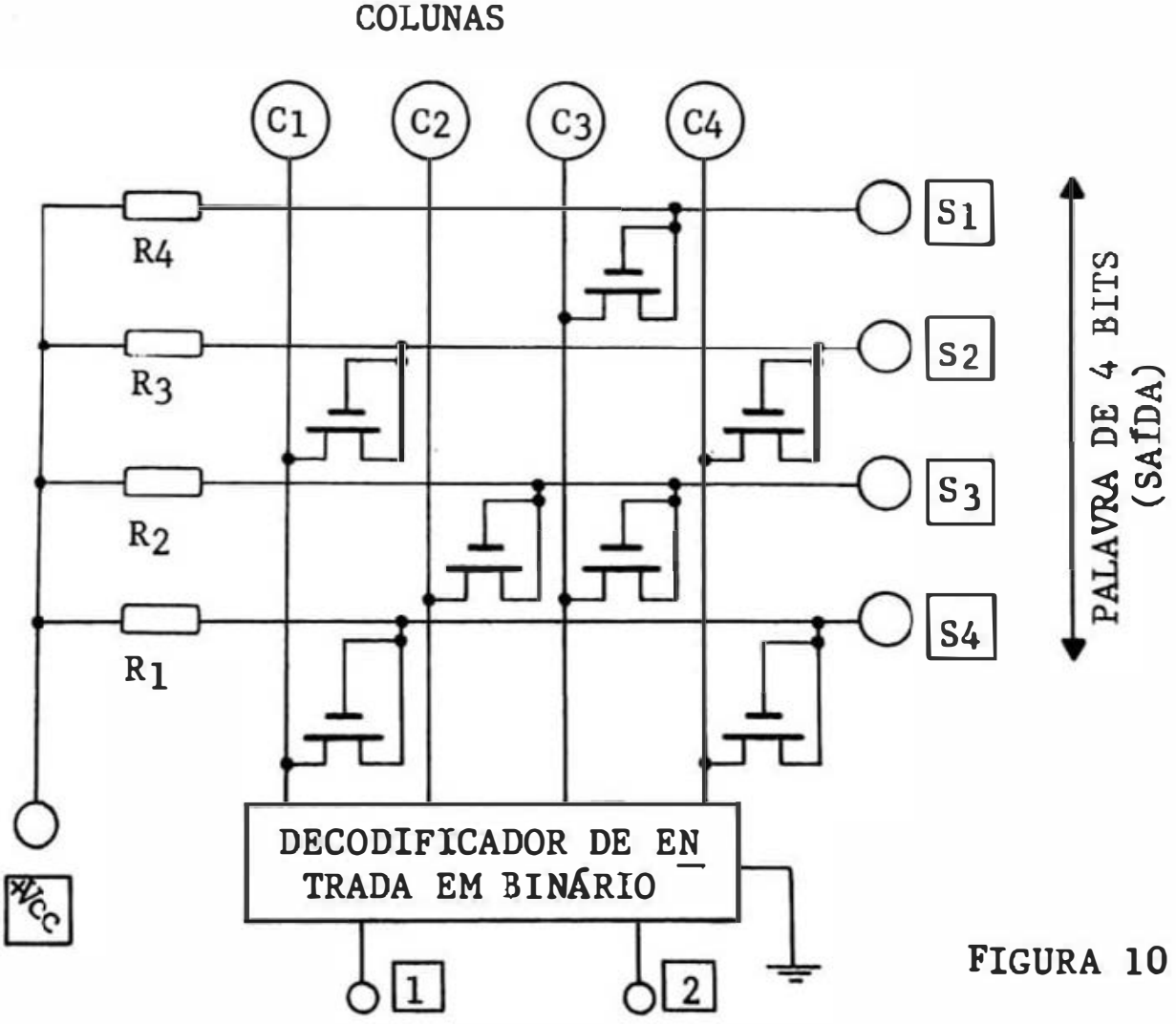


FIGURA 10

Por exemplo, suponhamos que, em uma determinada combinação das entradas 1 e 2, o decodificador leve a coluna C3 ao potencial de massa (interligue a coluna C3 à massa). Nessas - condições, as saídas S1, S2, S3 e S4 serão:

Saída S_1 = "0" binário, por estar conectada à coluna C_3 (através de um MOSFET) ligada à massa;

Saída S_2 = "1" binário, por estar conectada ao barramento +Vcc, através da resistência R_3;

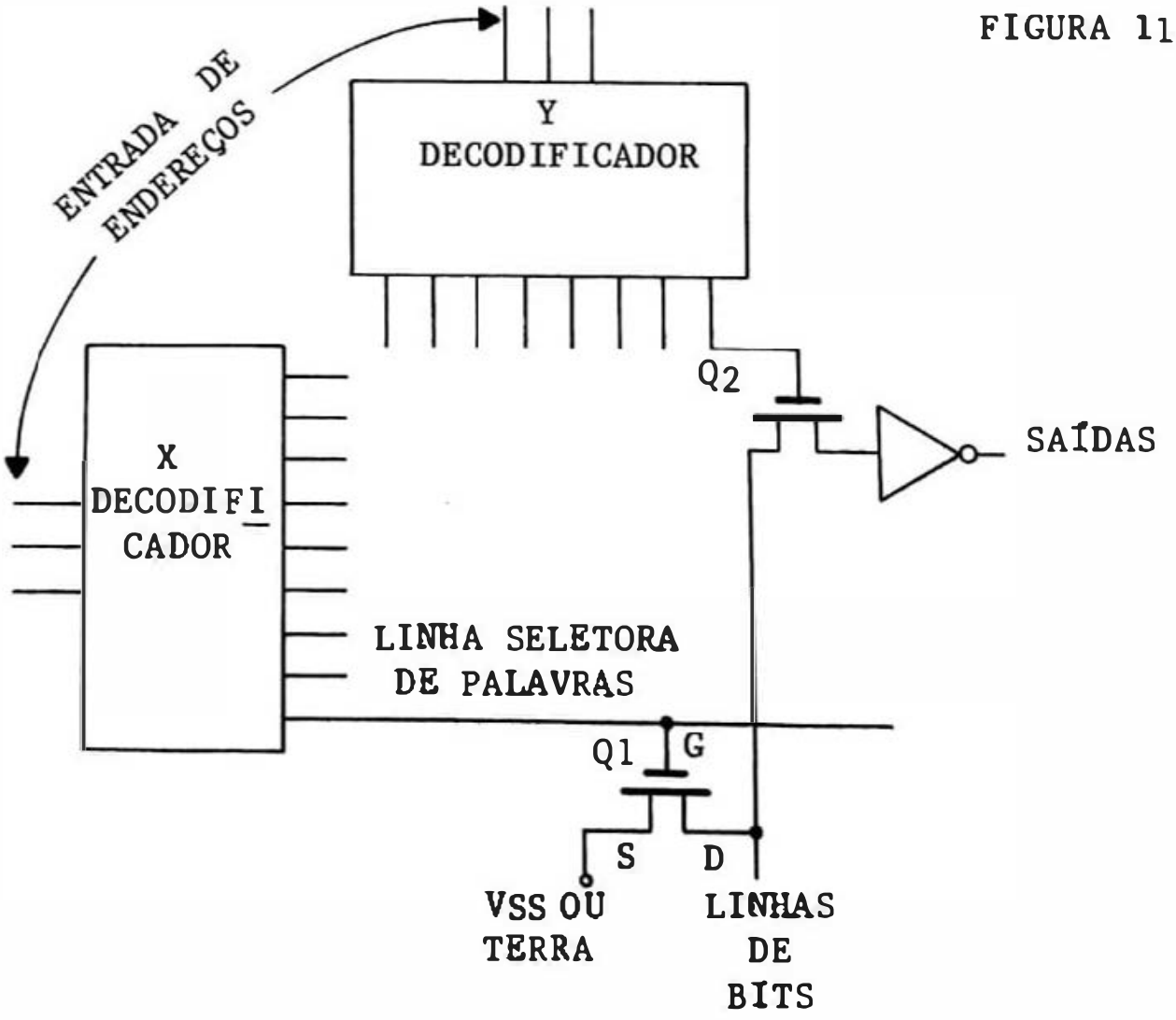
Saída S_3 = "0" binário, por estar conectada à coluna C_3 (através de um MOSFET) ligada à massa;

Saída S_4 = "1" binário, por estar conectada ao barramento +Vcc através da resistência R_1.

Em resumo, para a combinação suposta nas entradas 1 e 2, a palavra obtida na saída (S_1, S_2, S_3 e S_4) será 0101. Logicamente, para outras combinações binárias aplicadas nas entradas 1 e 2, teremos outras palavras de saída.

A figura 11 nos mostra um circuito ROM MOSFET, onde Q_1 e Q_2 são dois MOSFETs. A presença ou ausência desses transistores é função do processo de programação por mascaramento, levado a efeito durante a fabricação da ROM. A maioria das memórias ROM MOS emprega o método de decodificação por matriz XY. (Veja figura 11).

FIGURA 11



Como já dissemos várias vezes, as memórias ROM são programadas durante o processo de fabricação, mediante a utilização de máscaras.

A programação consiste em controlar, no caso das memórias ROM MOS, a espessura da camada de óxido da porta do MOSFET. É pois justamente essa espessura da camada de óxido da porta que irá definir a tensão de porta (THRESHOLD) do transistor. Por exemplo, para uma fina camada de óxido, podemos ter uma tensão de -5V e para uma camada mais espessa podemos ter uma tensão de -50V. Desta forma, ao aplicarmos, por exemplo, uma tensão de -10V na porta do transistor, este conduzirá ou não, dependendo da espessura da camada de óxido (fina ou grossa).

No caso das memórias ROM a diodos e ROM bipolares, este processo de programação também é válido, pois a ausência ou presença de um dispositivo semicondutor em uma matriz especifica um "1" ou um "0" binário, armazenado no local endereçado.

Um outro aspecto a ser considerado é o tempo de acesso. Como qualquer circuito lógico, a memória ROM apresenta um atraso de propagação. Isto quer dizer que há um tempo finito entre a aplicação de um endereço na entrada e o aparecimento de uma informação na saída. A este atraso de propagação dá-se o nome de tempo de acesso, o qual pode ser definido como sendo o tempo "tomado" para encontrar uma palavra na memória, no caso uma ROM, e lê-la.

Para memórias ROM bipolares, o tempo de acesso, geralmente, é menor que 100 nanossegundos (100×10^{-9}s). Já, para as memórias ROM MOS, o tempo de acesso típico é um pouco maior, girando em torno de algumas centenas de nanossegundos.

MEMÓRIAS ROM E SUAS APLICAÇÕES

As memórias ROM são extremamente versáteis na implementação de funções lógicas. Quando adequadamente programadas, podem ser utilizadas para substituir diversos tipos de circuitos combinacionais. Essas memórias são muito úteis na substituição de funções lógicas complexas, com múltiplas entradas e saídas. Aliada a todas essas possíveis aplicações, a ROM oferece as vantagens de maior rapidez e facilidade de projeto, baixo custo, tamanho menor e consumo reduzido de potência.

Como sabemos, os circuitos lógicos combinacionais geram sinais de saída que são funções dos estados da entrada, do tipo de portas usadas e do modo particular como elas estão interligadas.

A ROM desempenha o que é conhecido como uma função semelhante a uma tabela. Todos os lugares da memória podem ser considerados para fazerem parte de uma grande tabela de números. Ao aplicarmos um certo endereço à ROM, estaremos efetivamente selecionando um único número na tabela. Na verdade, a ROM não desempenha uma operação lógica. O que acontece é que os estados de saída, desejados para um dado conjunto de condições de entrada, estão simplesmente armazenados na memória.

Os exemplos enumerados a seguir ilustram algumas das muitas aplicações da ROM.

a) Lógica Aleatória:

Uma memória ROM pode ser usada para implementar, de maneira rápida, qualquer função lógica randônica (aleatória) que envolva múltiplas entradas e saídas. Ao invés de se projetar um circuito lógico combinacional com portas lógicas comuns, desenvolver a tabela verdade, definir a operação (através da tabela verdade) a ser desempenhada, aplicar a álgebra booleana para minimizar as equações e, por fim, partindo das equações finais, executar o circuito lógico, quando uma memória ROM é utilizada, o único passo do projeto é a implementação da tabela verdade. Como se sabe, a tabela verdade define as entradas e saídas e essa é toda a informação necessária para desenvolver uma memória ROM, capaz de desempenhar a função lógica requerida.

Note que, usando uma memória ROM, passamos, diretamente, da tabela verdade para o circuito lógico, o que representa uma economia substancial de tempo em termos de projeto. Lógico que, em se tratando de circuitos combinacionais elementares, envolvendo apenas algumas entradas e saídas, tal circuito é melhor implementado por intermédio dos meios convencionais, usando portas lógicas comuns. Isto é válido apenas com relação ao fator custo. Normalmente, o ponto de decisão entre utilizar uma memória ROM ou os meios convencionais ocorre quando o número de entradas e saídas do circuito é maior ou igual a quatro.

b) Conversão de código:

Para se utilizar uma memória ROM como conversor de código, o código de entrada deve ser igualado ao código de endereçamento binário da ROM e, no lugar da memória, especificado pelo código de endereçamento, colocaremos o código de saída desejado. Isto significa que os códigos de saída desejados são simplesmente acumulados nos lugares da memória e lidos quando o código equivalente de entrada é aplicado.

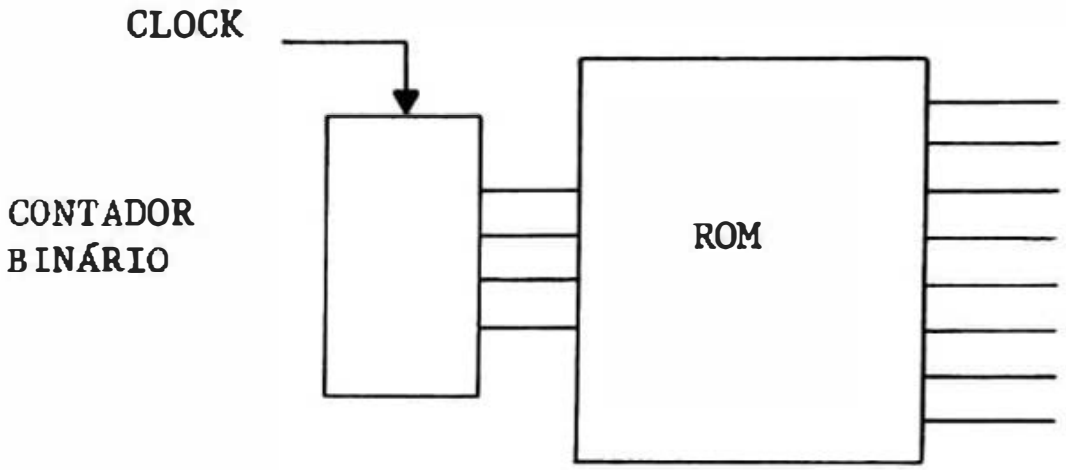
c) Operações aritméticas:

Trata-se pois de uma das funções mais difíceis de serem implementadas com circuitos digitais. Circuitos lógicos combinacionais simples têm sido desenvolvidos para realizar adições e subtrações. Vários algoritmos foram desenvolvidos para usar adições e subtrações, juntamente com outras operações digitais, como por exemplo "deslocamento", para se obter multiplicação e divisão. As funções trigonométricas e logarítmicas são mais difíceis ainda de serem implementadas. Através da utilização de memórias ROM, consegue-se um método mais simples de se implementar as operações aritméticas mais complicadas. Desta forma, gravam-se na memória ROM, por exemplo, as funções trigonométricas correspondentes aos ângulos. Ou ainda podemos utilizá-la para armazenar os logaritmos de números de entrada específicos. Desta forma, podemos dizer que a ROM foi transformada em uma tabela de logs ou uma tabela de funções trigonométricas. Para que isto ocorra, basta injetarmos na ROM o ângulo do número desejado, sob a forma de endereço binário e, no endereço que representa o ângulo ou o número requerido, armazena-se a função trigonométrica ou o logaritmo correto.

d) Microprogramação:

Trata-se de uma técnica desenvolvida para sistematizar a lógica de controle automático em um computador digital. A memória ROM nada mais é do que o"coração" de uma unidade de controle microprogramada. Esta memória, combinada com outros elementos lógicos, passa então a realizar operações lógicas seqüenciais. A esta combinação dá-se o nome de controlador microprogramado.

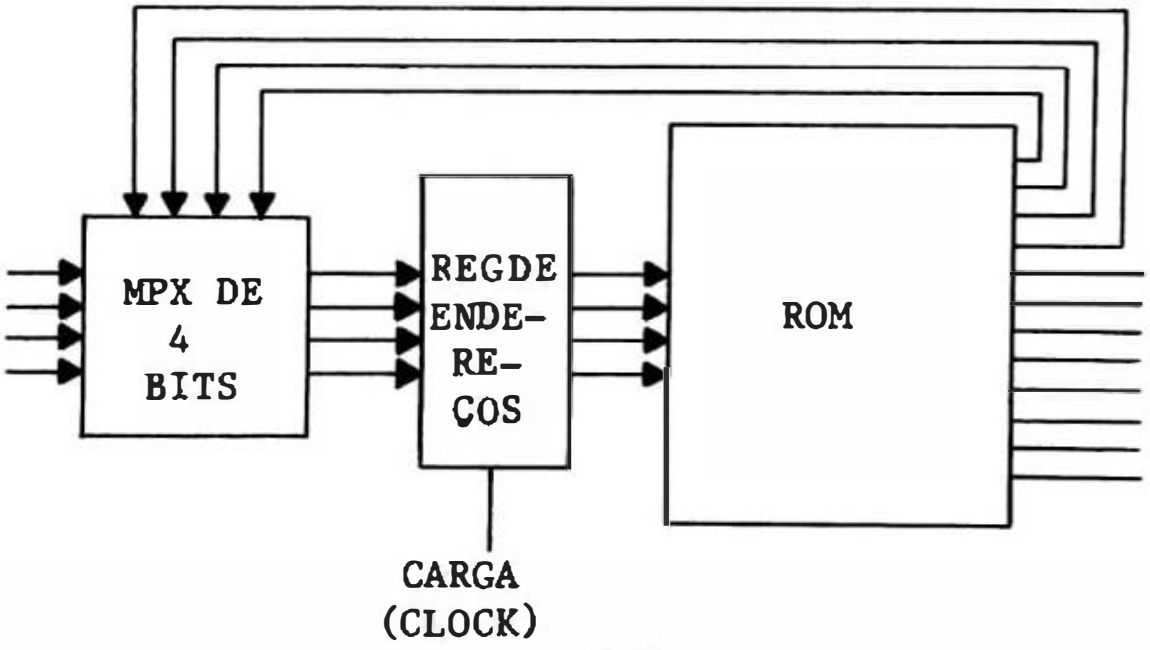
Pelo uso do controlador microprogramado, toda a rede de circuitos lógicos seqüenciais e combinacionais pode ser substituída por um circuito muito simples, contendo uma memória ROM. Na figura 12, apresentamos dois exemplos (a e b) de métodos de implementação de funções lógicas seqüenciais com memórias ROM.



(A)

FIGURA 12

IMPLEMENTAÇÃO DE FUNÇÕES LÓGICAS UTILIZANDO MEMÓRIAS ROM.



(B)

Como podemos observar na figura 12-a, a ROM é comandada por um contador binário, o qual é incrementado periodicamente por um sinal de clock. A saída do contador é utilizada como entrada de endereçamento da ROM. O decodificador de endereços é considerado como parte integrante da própria ROM. As palavras armazenadas na ROM são programadas de tal modo que os estados binários que aparecem nas saídas da ROM causam o acontecimento das operações lógicas desejadas na seqüência correta. O ritmo de variação das saídas da ROM é função direta da

freqüência dos pulsos de clock aplicados no contador binário.

A figura 12-B nos mostra uma versão mais sofisticada do circuito apresentado na figura 12-A.

O termo microprogramação é aqui utilizado no sentido de que as palavras armazenadas na ROM constituem um programa específico para o desempenho de uma função também específica. Desta forma, cada palavra gravada na ROM é referida como uma micro instrução. De uma forma bem resumida podemos dizer que um microprograma define a operação completa a ser iniciada(ou realizada).

MEMÓRIAS PROM

Do inglês "Programmable Read only Memory", que significa memória de apenas leituras programáveis, são memórias ROM que podem ser programadas pelo próprio usuário. Normalmente - são adquiridas no estado "virgem" (não programadas). Muitas - memórias PROM são fabricadas sob a forma de matriz de diodos, porém com diodos presentes em todos os cruzamentos das linhas com as colunas (veja figura 13). Nessas condições, uma memória PROM, virgem, apresenta sempre o nível lógico "1" (ou"0", se as saídas forem trocadas) para qualquer que seja o endereço selecionado. A programação consiste simplesmente em eliminar os diodos colocados nos cruzamentos onde desejamos ter nível lógico "0".

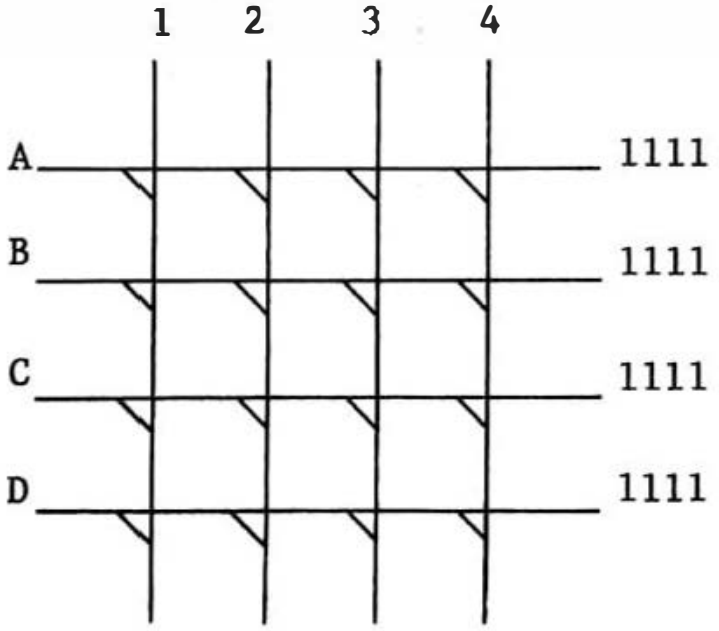


FIGURA 13

MEMÓRIA PROM VIRGEM

Existem dois procedimentos para se construir uma memória PROM, a saber:

a) Memórias PROM com "fusíveis em série":

Nas memórias PROM com fusíveis é conectado um fusível em série a cada ponto de memória. Estes fusíveis são feitos de pequenos fragmentos de nicromo ou alumínio, ou ainda de silício, conectado ao componente utilizado (diodo, transistor). Na figura 14-A, temos um esquema de ligação dos fusíveis e na figura 14-B, temos um esquema que nos mostra alguns aspectos construtivos dos fusíveis.

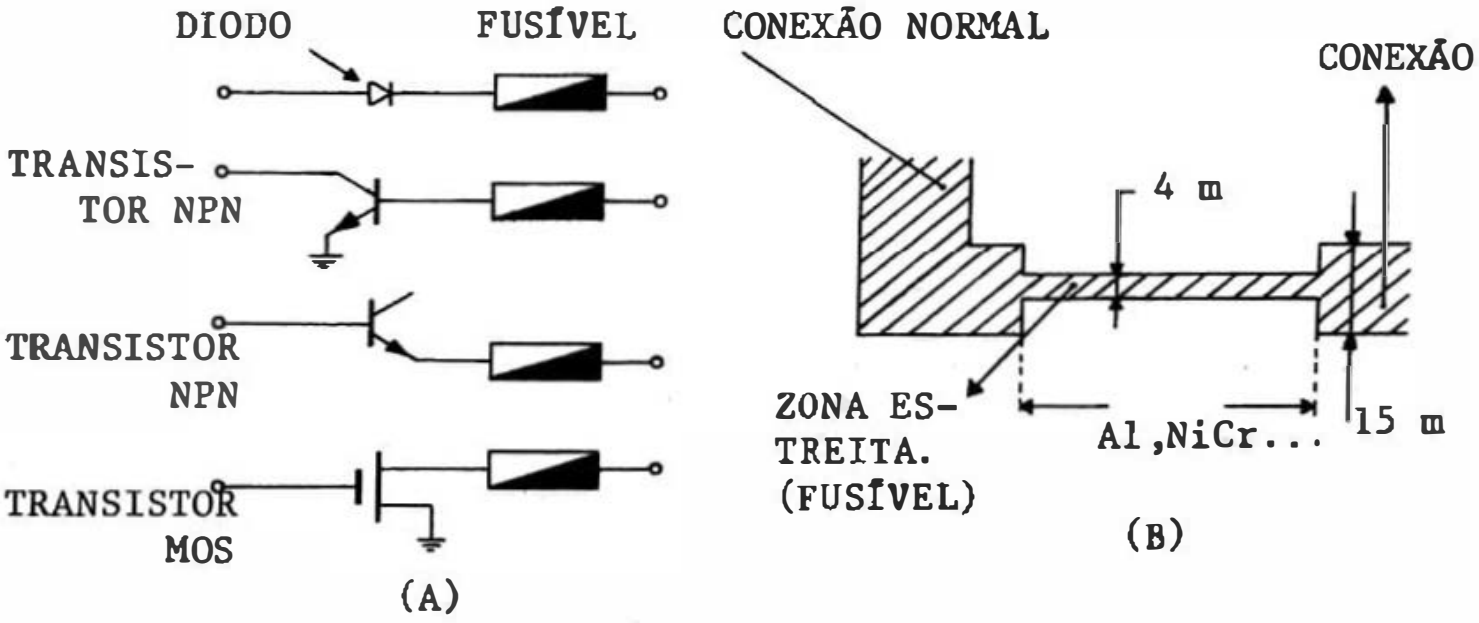


FIGURA 14

Consideremos a matriz de diodos da figura 15-A. Todos os pontos de memória são constituídos por diodos, sendo que, em cada um desses diodos, temos um fusível conectado (em série).

Para gravarmos uma certa informação, basta fazermos circular, através dos fusíveis correspondentes aos pontos de memórias desejados, uma corrente suficiente para romper esses fu-

síveis. (Veja figura 15-B). Deste modo, interrompem-se as interconexões adequadas entre linhas e colunas (cruzamentos). A figura 15-C nos mostra alguns aspectos de uma estrutura real, de uma memória PROM com fusíveis.

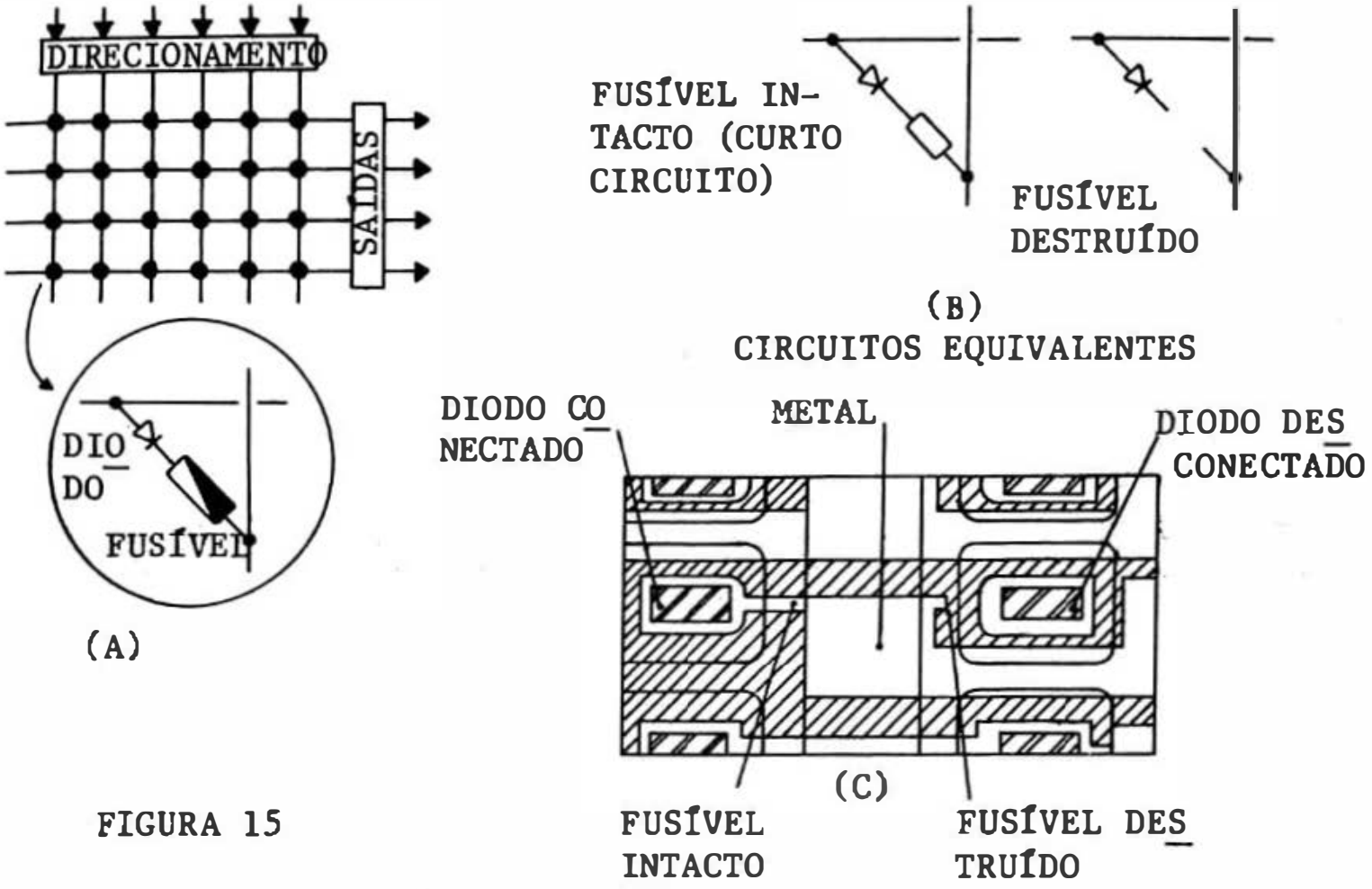


FIGURA 15

b) Memórias PROM com "destruição do isolamento":

O caso anterior consiste em destruirmos um fusível, o

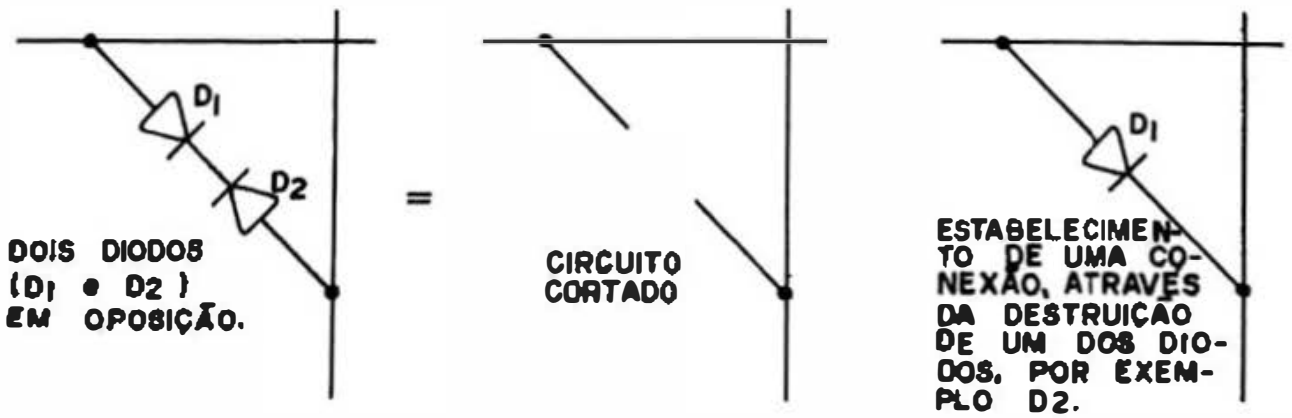


FIGURA 16

que corresponde a eliminarmos uma conexão. Já este método é justamente o contrário, pois ao invés de destruir uma conexão, iremos estabelecer uma conexão,destruindo, para tanto,um elemento de isolamento, por exemplo um diodo (veja figura 16).

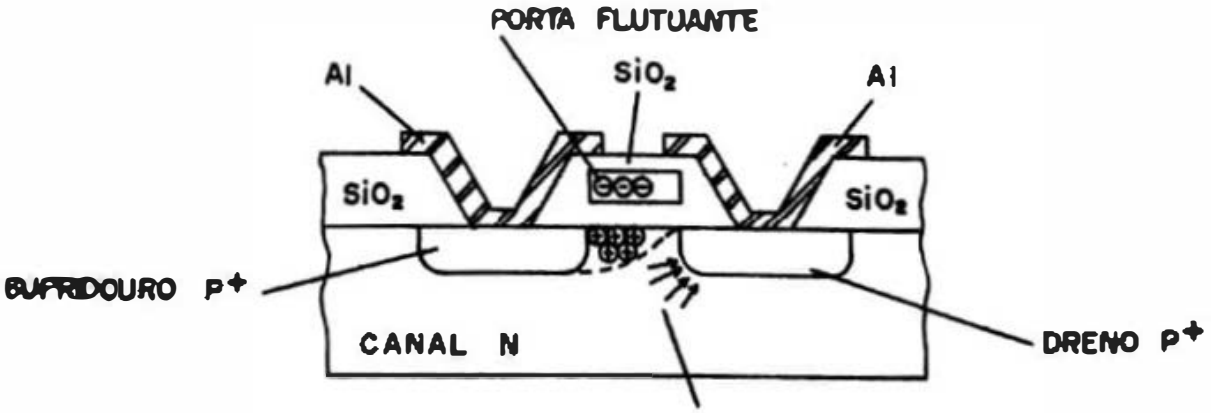
Para se destruir um diodo, basta colocá-lo em curto circuito, mediante um forte pulso de corrente e é nisto que se resume este processo.

MEMÓRIAS EPROM:

Do inglês "erasable PROM" que significa PROM apagável.Em outras palavras podemos dizer que as memórias EPROM são memórias que podem ser apagadas e reprogramadas.

A tecnologia empregada na fabricação das memórias EPROM é a FAMOS ("Floating-gate Avalanche-injection MOS"). Cada elemento de memória de uma memória EPROM FAMOS consiste de uma pequena barra de silício, localizada entre o dreno e o supridouro de um transistor de efeito de campo MOS (veja figura - 17). Essa pequena barra de silício não tem qualquer ligação com o dreno ou com o supridouro do transístor, nem mesmo externamente. Daí o nome de porta flutuante ("floating-gate"). Durante o processo de programação, aplica-se um pulso de tensão fortemente negativo entre o dreno e o supridouro, o qual irá provocar uma injeção de elétrons na porta flutuante, tornando a porta flutuante eletricamente carregada. No processo de apagamento da memória, os raios ultravioletas criam um efeito de ionização. Este efeito faz com que o excesso de elétrons presentes na porta flutuante volte ao substrato.

FIGURA 17



Como já dissemos, o apagamento de uma memória EPROM é feito através de raios ultravioletas. É óbvio que há necessidade desses raios penetrarem na memória para que possa então ocorrer o processo de apagamento. Daí a razão da existência de uma janela de quartzo no encapsulamento das memórias EPROMs, uma vez que o quartzo é transparente à luz ultravioleta. Na figura 18 temos alguns aspectos físicos com relação ao encapsulamento de uma memória EPROM.

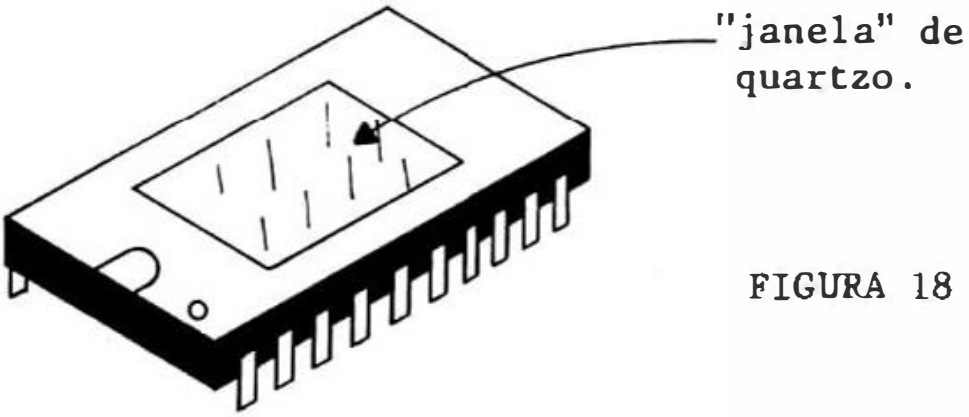
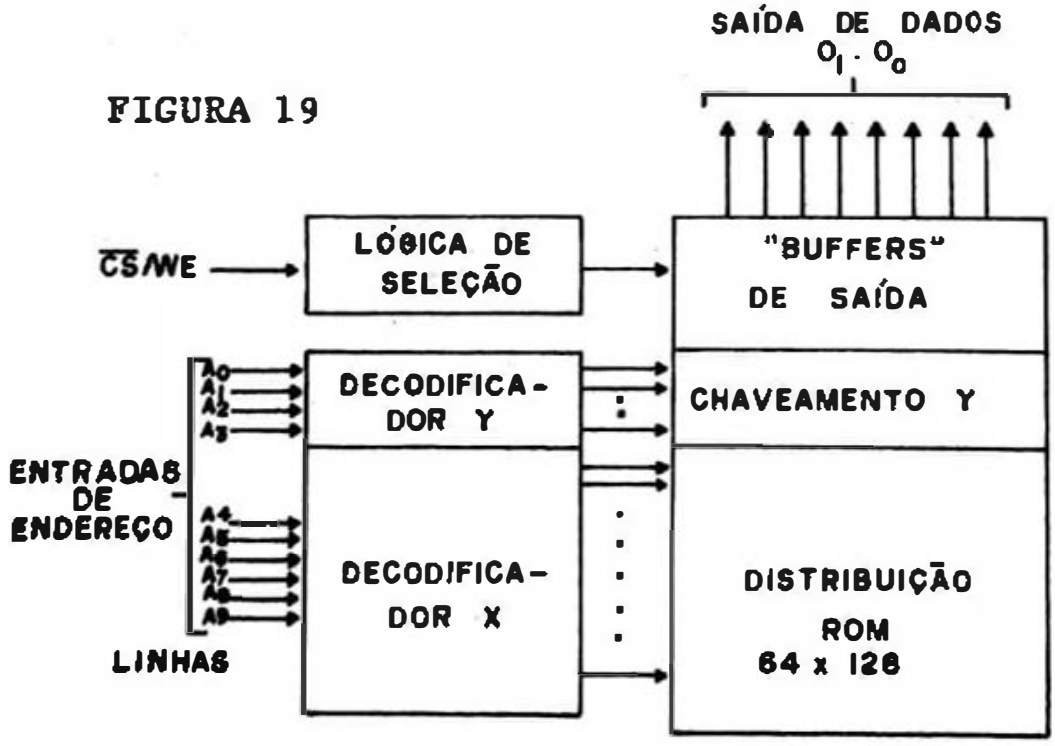


FIGURA 18

Quanto à organização, as memórias EPROMs podem ser encontradas com as mesmas características organizacionais de uma memória ROM ou de uma memória PROM.

Na figura 19 apresentamos a distribuição interna de circuitos de uma memória EPROM, sob a forma de diagrama de blocos.

FIGURA 19



Para se efetuar a leitura de uma memória EPROM, deve-se, primeiramente, selecionar o endereço, do mesmo modo que é feito para a programação. Em seguida, basta levar o terminal CS/WE ao nível "0" da lógica TTL e, então, ler os dados correspondentes àquele endereço.

Devido ao fato de utilizar raios ultravioletas no processo de apagamento, essas memórias também recebem a sigla de UV EPROM.

MEMÓRIAS EEPROM:

As memórias EEPROM (Electrically Erasable PROM) são memórias de funcionamento análogo ao das EPROMs, porém apagáveis eletricamente. A tecnologia empregada para sua fabricação é a MNOS (Semicondutores do tipo metal-nitreto-óxido-semicondutor). Na figura 20, temos uma estrutura, mostrada em corte, deste tipo de memória.

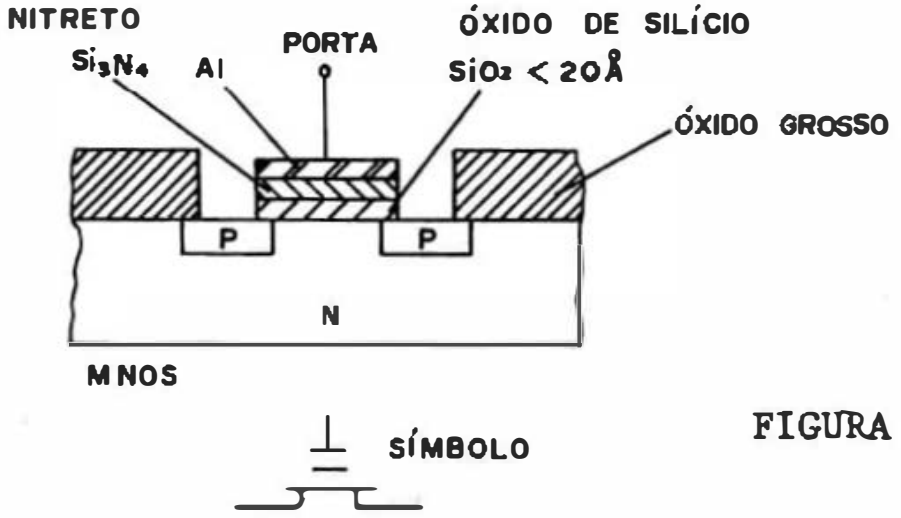


FIGURA 20

Seu funcionamento é o seguinte: se polarizarmos a porta positivamente e, estando os terminais de dreno e supridouro alimentados corretamente, os elétrons que atravessam a camada de óxido de silício (SiO2), por efeito túnel, dirigindo-se para a camada de nitreto, ao se encontrarem entre esses dois materiais serão "barrados", permanecendo aí por um tempo indefinido, mesmo que desapareça a polarização. Nestas condições, o transistor MOS é mantido no estado condutor.

Se aplicarmos uma tensão negativa, relativamente alta, na porta, os elétrons que foram "barrados" serão eliminados por

efeito túnel, atravessando o silício em sentido inverso. Desta forma, o transistor MOS estará bloqueado.

Esses dois fatos nos levam a um transistor MOS programável, o qual pode ser apagado eletricamente.

Apesar das memórias EEPROM poderem reter tais cargas(elétrons que se encontram entre o óxido e o nitreto) por um tempo indefinido, as memórias de nitreto exibem o problema de deterioração da informação armazenada, após um certo número de operações de leitura. Além disso, são relativamente lentas em operação, comparando-se os tempos de outras memórias ROM. Um outro fator a ser considerado é que alguns fabricantes de semicondutores dizem que o mecanismo de armazenamento dessas memórias não é totalmente confiável.

MEMÓRIAS EAROM

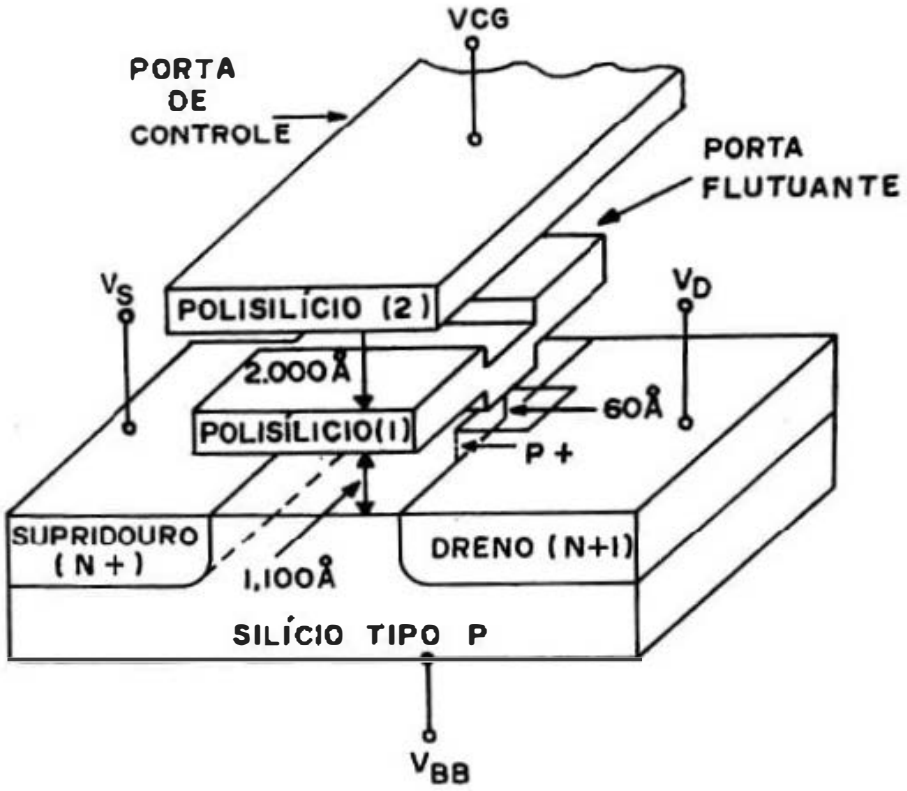
As memórias EAROM (Electrically Alterable Read only Memories), utilizando o princípio FAMOS modificado, reúne as vantagens da retenção de cargas e alta velocidade das UV EPROM, as vantagens de apagamento por sinais elétricos, além de utilizarem um mecanismo de armazenagem mais confiável e mais familiar aos fabricantes de semicondutores.

O ponto de memória ou a célula básica de memória de uma EAROM FAMOS é composta de dois transistores FET, fabricados com a tecnologia MOS de canal N. O transistor A é responsável pelo endereçamento externo e o transistor B é, realmente, o elemento de memória. A figura 21 nos mostra alguns detalhes construtivos do transistor B.

Observando a figura 20, podemos notar que o transistor nela representado contém duas portas, sendo uma de controle e outra, flutuante, responsável pelo armazenamento das cargas injetadas.

A porta flutuante, quando carregada, reproduz o nível lógico "1" e, quando descarregada, reproduz o nível lógico "0". Essas duas condições determinam a leitura e escrita de informações na célula de memória. Após cada operação de apagamento, o transistor "B" (veja figura 20) permanece no estado conduti vo, o que corresponde a uma carga positiva na porta flutuante. A operação de escrita comuta o transistor de memória para o corte, correspondendo a uma carga negativa na porta flutuante.

FIGURA 21



Na figura 22, apresentamos o diagrama de blocos de uma memória EAROM com capacidade de 8192 bits, com comprimento de palavra de 8 bits.

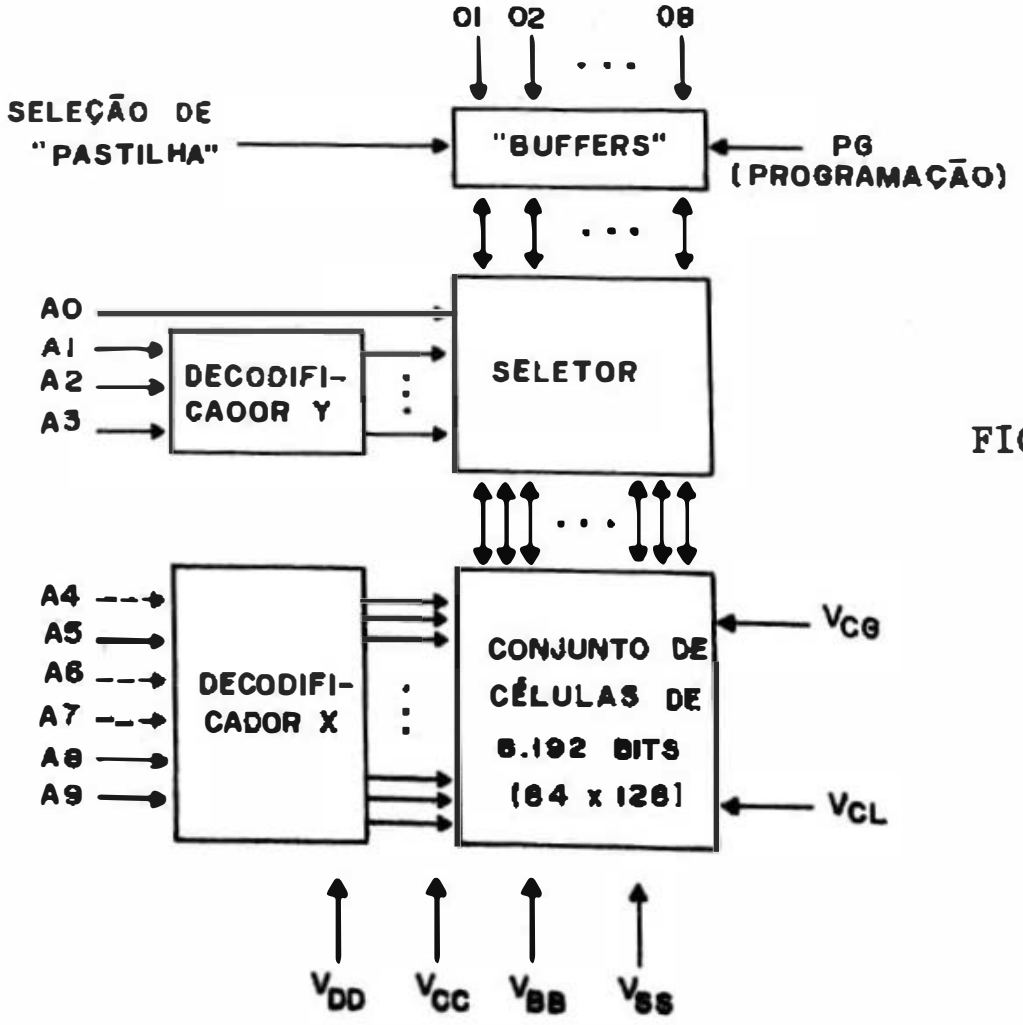


FIGURA 22

Devido as suas características excepcionais de retenção de informações e apagamento/reprogramação, as memórias EAROM de nitreto e FAMOS possuem inúmeras aplicações especiais, como por exemplo, em odômetros de automóveis, onde armazenam a leitura em quilômetros, atualizando-a a cada décimo de quilômetro. As memórias EAROM também são utilizadas em calculadoras programáveis, em aparelhos digitais de TV, em sintonizadores de FM, em seções de sintonia de rádios da faixa do cidadão,em máquinas operatrizes, etc...

COMENTÁRIOS FINAIS:

Com esta lição, finalizamos o estudo de memórias, porém no decorrer do curso, quando necessário, daremos outras informações sobre o assunto.

CEAM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-6

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) As informações gravadas em uma memória ROM:

a) podem ser alteradas sem qualquer prejuízo;
b) ficam permanentemente gravadas;
c) são apagadas quando houver corte no fornecimento de energia;
d) N.R.A.

2) As informações são escritas na memória ROM quando ela está sendo:

a) fabricada;
b) endereçada;
c) operada;
d) N.R.A.

3) A palavra binária de entrada para uma memória ROM é, geralmente, conhecida por:

a) byte;
b) bit;
c) dados;
d) endereço.

4) Com relação à memória ROM com diodos, quando não houver diodo entre o cruzamento da linha com a coluna, a tensão de saída será igual a:

a) -Vcc;
b) +Vcc/2;
c) +Vcc
d) -Vcc/2.

5) Com relação à figura 6, a linha G é utilizada para:

a) liberar o circuito;
b) inibir o circuito;
c) os itens a e b estão corretos;
d) N.R.A.

6) Em uma memória ROM MOS, a presença de um MOSFET indica que:

a) foi programado na memória um "1" binário;
b) foi programado na memória um "0" binário;
c) a memória está danificada;
d) N.R.A.

7) Ainda com relação às memórias ROM MOS, a programação consiste em controlar:

a) a espessura da camada de óxido da porta do MOSFET;
b) a espessura da camada de óxido do supridouro do MOSFET;
c) a espessura da camada de óxido do dreno do MOSFET;
d) N.R.A.

8) Para as memórias ROM bipolares, o tempo de acesso fica, geralmente, em torno de:

a) 1.000×10^{-2} segundos;
b) 1.000×10^{-9} segundos;
c) 100×10^{-6} segundos;
d) 100×10^{-9} segundos.

9) Normalmente as memórias PROM são adquiridas:

a) gravadas;
b) virgens;
c) programadas;
d) N.R.A.

10) Nas memórias EPROM, quando do processo de apagamento, os raios ultravioletas criam:

a) uma oxidação;
b) uma desintegração molecular;
c) um efeito de ionização;
d) N.R.A.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-7

DISPOSITIVOS PERIFÉRICOS I

INTRODUÇÃO:

Procuraremos, nesta lição e nas seguintes, levar até o aluno algumas informações sobre os periféricos de um computador, seu funcionamento, sua mecânica e sua utilidade. Iniciaremos, portanto, esta lição com um pequeno histórico dos computadores e seus periféricos.

HISTÓRICO

Os computadores começaram a ser utilizados comercialmente há cerca de 25 anos. O desenvolvimento tecnológico obtido, neste setor, a partir do final da década de 60, permitiu uma dramática queda no custo dos computadores e dispositivos periféricos, conduzindo diretamente para o uso popular de minicomputadores e microcomputadores, bem como de seus respectivos periféricos, resultando, desta forma, em um vasto mercado para os pequenos computadores e uma grande demanda para os dispositivos periféricos.

A explosão do uso de mini e microcomputadores está dire-

tamente ligada ao desenvolvimento da "unidade central de processamento" UCP, em um único chip eletrônico, o que só se tornou possível através do surgimento da tecnologia de integração em larga escala (L.S.I.)

Antes de 1.970 era muito difícil interligar os periféricos de um fabricante com o computador principal de um outro fabricante. Até a pouco tempo, os computadores eram, então, comercializados como sistemas intactos. Porém, com o advento dos mini e microcomputadores de baixo custo, começou a ocorrer uma inversão dos fatos e os computadores começaram a ser comercializados não mais como sistemas intactos. Hoje, um sistema de computador pode ser formado por várias "caixas" (periféricos) separadas, interligadas ao computador, bastando para tanto, conectar essas "caixas" aos plugs padrão e às interfaces que se fizerem necessárias.

Desde que o barramento de fiação do computador e as marcações das pinagens das placas de um circuito interface periférico forem compatíveis, pode-se ligar essas placas de circuitos individuais em computadores fabricados por firmas diferentes. Os pesquisadores, e mesmos alguns hobistas costumam projetar as suas próprias interfaces periféricas para os seus computadores. Outros já preferem modificar a configuração dos chips em uma placa de circuito existente, utilizando, para tanto, outros chips lógicos de circuitos integrados (outros componentes).

Também é muito útil examinar a hierarquia do HARDWARE (circuitos lógicos) das máquinas, ou seja, chips, placas de circuitos, dispositivos periféricos, etc., pois tudo isto leva a um melhor conhecimento da máquina ou do sistema com que se está trabalhando.

Procuraremos então, nestas lições, relacionar as características funcionais dos equipamentos e também o uso dos periféricos em computadores simples.

PERIFÉRICOS

Um periférico pode ser entendido como sendo um dispositivo ligado ao computador, a fim de obter informações úteis fora da UCP (Unidade Central de Processamento) do computador.

O nome periférico surgiu por volta de 1.960, quando a UCP do computador era tida como sendo um dos itens mais signi

ficantes de um sistema e também o de maior custo. Assim,qualquer equipamento (unidades de disco magnético, unidades de fita magnética, teclado, impressora, terminal de vídeo, etc.)conectado ao computador central, era na verdade, periférico ao computador, pois estava interligado ao redor do computador.Essa idéia nos leva a ver os periféricos como sendo caixas separadas do computador.

Hoje, com o advento dos mini e microcomputadores, essa idéia já não é necessáriamente verdadeira, pois muitos computadores, mini e microcomputadores já possuem dispositivos periféricos colocados dentro do mesmo gabinete (caixa) que contém a UCP. Assim, o "abrigo" físico da UCP e dos periféricos não é mais um indicador seguro do que é um computador e do que é um periférico. Também, o elemento custo deixou de ser um bom indicador para diferenciarmos um computador de um periférico, pois hoje em dia é até comum o custo de alguns periféricos ultrapassarem o custo do computador em si.

Generalizando, podemos dizer que a maioria dos periféricos possui três (3) funções principais, a saber:

a) controlar o veículo de transporte da informação (fita, papel, disco, etc.);

b) ler a informação e montar as palavras;

c) desmontar as palavras e escrever a informação.

Essas funções foram colocadas em uma seqüência lógica de execução. Com base nesses fatos, podemos então considerar os periféricos como sendo dispositivos utilizados para realizar a entrada/saída de informações em um sistema de computador.

Classificar os periféricos em uma ordem sensível, em que se enquadrem às atuais tendências deste setor, é tarefa árdua, e difícil. Porém, com base neste último conceito sobre o que vem a ser um periférico, podemos dividí-los em dois grandes grupos:

a) Periféricos que executam a entrada/saída (E/S) verdadeira: Trata-se dos periféricos utilizados para a comunicação do homem com a máquina e vice-versa.

b) Periféricos que executam a entrada/saída (E/S) de armazenamento: Neste caso, trata-se dos periféricos utilizados para o armazenamento secundário das informações(memórias de massa, arquivos a longo prazo,etc.).

Uma outra classificação possível dos periféricos pode ser feita com base no destino e na origem da informação intercambiada pelas unidades de entrada/saída (E/S). Assim, podemos classificá-los em dois grupos.

a) Periférico de entrada: Trata-se do periférico que é fonte de informação, por exemplo, o teclado de um computador ou de um microcomputador;

b) Periférico de saída: Trata-se do periférico destino da informação, por exemplo, o terminal de vídeo ou a impressora de um computador ou de um microcomputador.

Devido ao incremento dos tipos de aplicações e da evolução tecnológica, o universo dos periféricos tem crescido com grande rapidez, em parte motivado, precisamente, pelas necessidades crescentes de se dispor de periféricos aptos a serem utilizados em sistemas baseados em microcomputadores.

Ao se discorrer sobre o tema periféricos, devemos levar em consideração algumas características, como:

a) Custo: em muitos casos, o custo de um periférico e dos circuitos associados para seu controle podem representar uma parte muito significativa do custo total do sistema. Porém, o ideal é poder dispor de periféricos baratos e sistemas de controle sensíveis. O custo varia muito, dependendo do sistema, da aplicação e da tecnologia empregada na fabricação do mesmo.

b) Tamanho e peso: muitos sistemas baseados em microcomputadores são portáteis, pelo que o volume e peso do equipamento passam a ser um fator determinante.

c) Consumo: esta característica está muito ligada ao tamanho e peso do equipamento e ao consumo associado às partes eletromecânicas do mesmo. Ao se projetar as fontes de alimentação, deve-se tomar o cuidado de fazê-las compactas e baratas.

d) Fiabilidade: pela forma de operação de muitos sistemas microcomputadores, em comunicação constante com o mundo externo (real), é muito importante dispor de equipamentos periféricos com índices de fiabilidade o mais próximo possível do índice do próprio microcomputador.

Como já dissemos, há pouco,os tipos de periféricos utilizados em sistemas de microcomputadores são muito variados em função da aplicação a que são destinados, o que torna exaustiva a enumeração de todos eles. Porém, de uma maneira bem resumida, apenas a título de introdução ao assunto, apresentaremos alguns desses periféricos. Nas próximas lições, será feita uma análise mais detalhada dos dispositivos periféricos mais comuns. Então vejamos:

1) Periféricos internos: são sub-sistemas do microcomputador que não são, na realidade, periféricos, entretanto podem ser considerados como tais. Entre estes, podemos destacar os circuitos utilizados para aumentar a potência aritmética do sistema, como os multiplicadores monolíticos, os geradores de funções e os temporizadores programáveis, os quais permitem a realização por hardware de "contadores de sucessos externos", de geração de um relógio diário ou de um relógio de tempo real, etc...

2) Interruptores, pulsadores e comutadores: são dispositivos que permitem a entrada de dados no sistema, a seleção de diversas opções ou modo de funcionamento,o envio de ordens de atuação, etc. A forma de se estabelecer o contato dependerá do tipo de contato considerado: contatos metálicos, de lâmina, de acoplamento óptico, de efeito capacitivo, magnéticos, por efeito Hall, etc.

3) Teclados: é uma forma de se dispor de uma série de pulsadores, de forma compacta, para se fazer a entrada de informações alfanuméricas e de controle ao sistema. Apresenta vantagens quanto à flexibilidade de operação, sensibilidade de manejo, custo, estética e outras.

4) Visores: podendo ser construídos por diversos métodos ópticos (diodos emissores de luz, cristal líquido, incandescência, fluorescência) e eletromecânicos. Fornecem ao operador, de uma forma visível, informações sobre o sistema. As informações, segundo os visores utilizados, podem ser de três tipos:

- binárias: indicadores individuais;
- numéricos: permitem visualizar os dígitos de 0 a 9 e algumas letras;
- alfanuméricos: permitem visualizar os dígitos de 0 a 9, todo o alfabeto e alguns caracteres especiais.

Devido à forma de controle, os visores podem ser estáticos e multiplexados.

5) Visor de "Tubo de raios catódicos" (TRC): consiste na incorporação de um sistema de controle em um tubo de raios catódicos (TRC), de modo a dispormos de um visualizador alfanumérico. Um terminal TRC é formado por uma tela, um teclado e um sistema de hardware/software adequado, para que sua utilização seja flexível e precisa. Existem sistemas mais arrojados que permitem a visualização de gráficos, enquanto outros permitem a entrada de dados, por meio de uma ponta luminosa sobre uma tela.

6) Impressoras: as impressoras mais utilizadas em sistemas de microcomputadores são as de média e baixa velocidade, numéricas ou alfanuméricas e, em geral, de escrita carácter por carácter. Para aplicações que exigem uma escrita mais rápida, pode-se utilizar uma impressora de linha. Quanto ao modo de escrita, as impressoras podem ser classificadas em: "eletromecânicas" (de tambor, de matriz de agulhas, de bola, de margarida, etc), "térmicas", etc. Algumas impressoras também permitem a geração de gráficos por pontos (registradores e ploters).

7) Memórias Auxiliares: permitem dispor de um meio de memória de grande capacidade e baixo custo. Os sistemas mais usados em minicomputadores e computadores (discos rígidos e fitas magnéticas) não são utilizados nos sistemas de microcomputadores. Nos sistemas de microcomputadores empregam-se muito os discos flexíveis (floppy disk) e os cassets digitais (fitas).
O disco flexível é um sistema cujo custo é relativamente baixo e bastante confiável. Os tamanhos mais utilizados são os de 5 1/4" (polegadas) e 8". A capa-

cidade de um disco flexível simples,"face estander" e simples densidade oscila entre 2 e 3 M bits (milhões de bits),enquanto que a velocidade de transferência, uma vez estando o disco posicionado, é da ordem de 250 K bits/segundo. As fitas cassets (ou cassets digitais) são mais baratas que o disco flexível, porém apresentam desvantagens como: tempo de acesso muito grande, velocidade de transferência e índices de fiabilidade bem baixos.

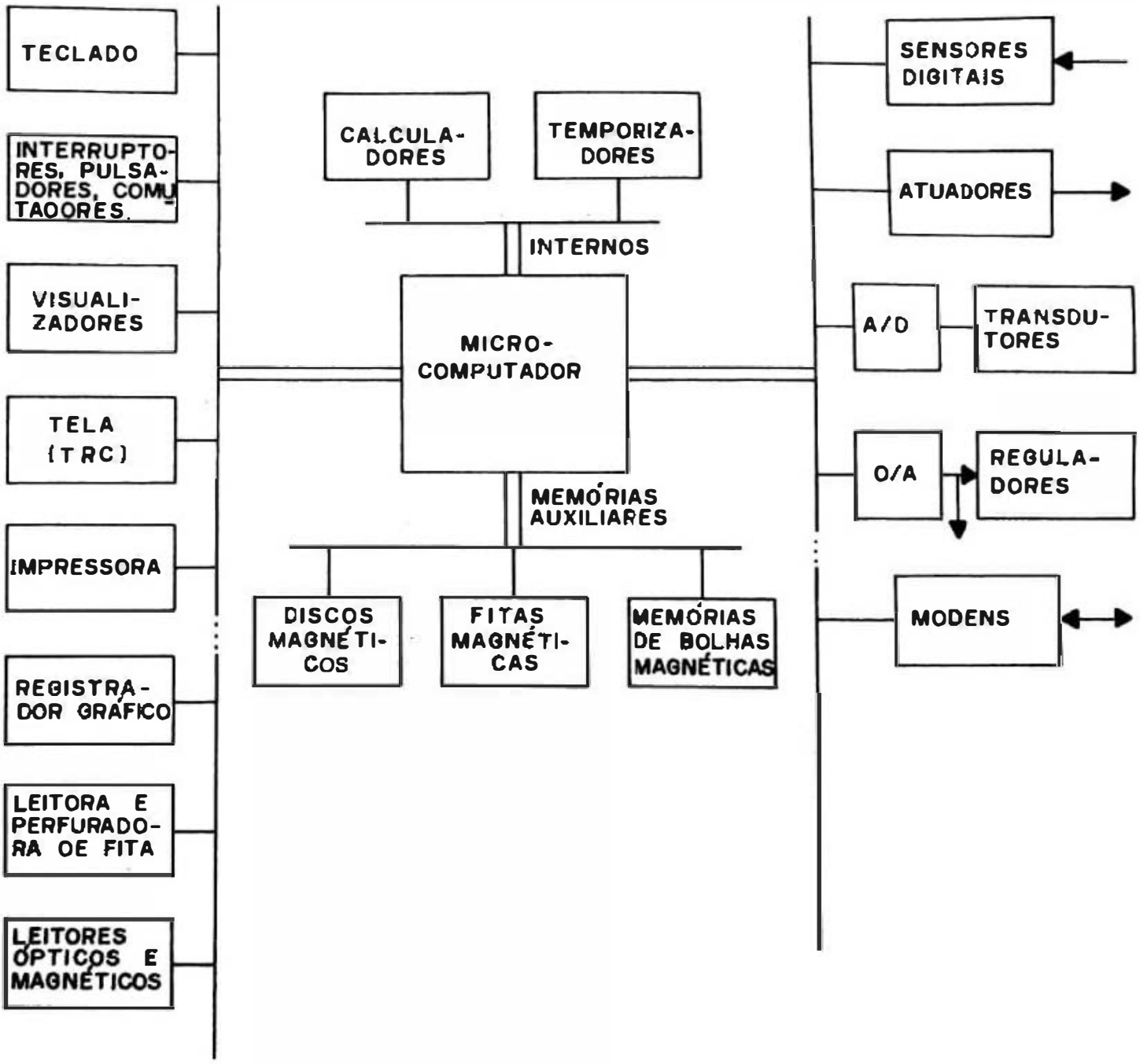
8) Modems: são sistemas utilizados na conversão e reconversão de sinais digitais em analógicos, de forma que se pode enviar a informação, em série, através de linhas de transmissão. Será, portanto, necessário se dispor de um sistema modulador/demodulador (modem) em cada terminal da linha. As técnicas mais utilizadas - são as de modulação de amplitude, freqüência e fase com referência fixa ou diferencial. A transmissão pode ser efetuada pelo método full-duplex (transmissão e recepção simultânea) ou half-duplex.

9) Conversores A/D e D/A: muitas das variáveis precisas em um sistema de controle (temperatura, pressão, posição, velocidade, caudal, isolação, etc.) são captadas por dispositivos transdutores que transladam a magnetude física, considerada em variáveis elétricas contínuas (tensão, corrente), sendo portanto, necessário um "meio" de conversão A/D da informação de forma que esta possa ser processada pelo computador. Igualmente, a informação de saída também pode ser requisitada na forma analógica, tornando-se, portanto, necessário, o uso de um conversor D/A.

10) Leitoras e perfuradoras de fita de papel: são periféricos que permitem a leitura e escrita da informação, utilizando como suporte físico uma fita perfurada, de papel ou de plástico. Esses periféricos são relativamente lentos, com velocidades de perfuração de até 300 caracteres/segundo e leituras da ordem de 1.000 caracteres/segundo para leitoras ópticas e de acoplamento capacitivo.

A figura 1 nos mostra um diagrama de blocos com alguns dos principais dispositivos periféricos, interligados a um microcomputador.

FIGURA 01



Vamos então dar início a um estudo mais pormenorizado sobre alguns dos principais dispositivos periféricos, onde procuraremos mostrar suas principais características, suas vantagens e suas desvantagens. Começaremos por estudar o teclado.

TECLADO

Denomina-se "teclado" o periférico de entrada, constituído por um conjunto de botões pulsadores, sendo que cada botão corresponde a um determinado carácter, função, instrução ou idéia.

Os tipos de teclas, assim como seu número e distribuição, são determinados pela aplicação concreta que se deseja realizar. Daí o fato de não existir um modelo genérico de teclado, mas sim, projetos específicos.

O número de teclados conectados a um determinado sistema é muito variável, oscilando entre zero (tal como em sistemas muitos simples de automatização) e várias dezenas (tal como em sistemas multiterminais de recolhimento de informações e consulta de bancos de dados). Entretanto, nos sistemas baseados em microcomputadores, é comum o uso de apenas um teclado.

É comum o teclado estar associado a um outro periférico de saída, tal como uma impressora, uma tela ou ainda um visualizador, com o que o operador obtém uma comunicação bidirecional com o sistema.

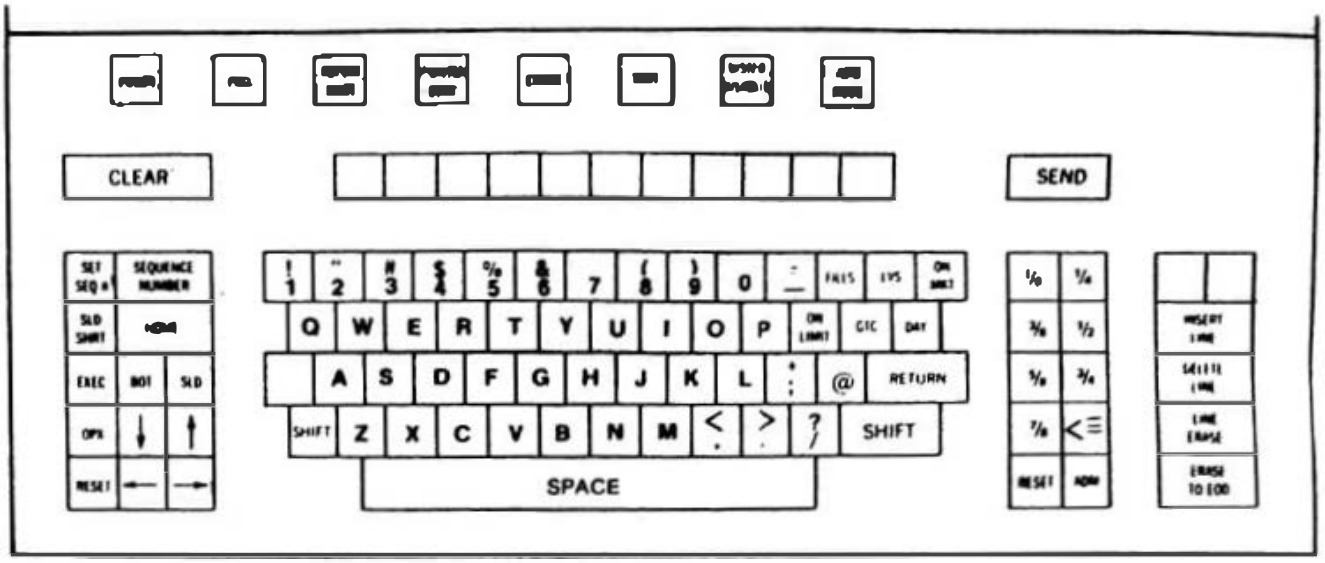
Quanto à função, pode-se classificar os teclados em três grupos:

a) Teclados alfanuméricos: são teclados utilizados para introduzir letras, números, símbolos e comandos, por exemplo, em um microcomputador. A figura 2 nos mostra um teclado alfanumérico. Nesses teclados, as teclas são dispostas com um certo espaçamento, para maior conveniência do operador.

FIGURA 02

Já a figura 3 nos mostra um outro teclado alfanumérico, projetado especialmente para um terminal de um departamento de estoque, combinando várias funções de programa especialmente marcadas e projetadas pelo usuário.

FIGURA 03



b) Teclados numéricos: utilizados para introduzir dados com uma colocação numérica de teclas. Este tipo é, usualmente, mais conveniente do que as teclas numéricas em um teclado alfanumérico. Os teclados numéricos são também utilizados em conjunto com um teclado alfanumérico (observe o teclado numérico disposto no lado direito de um alfanumérico, na figura 2). Os teclados decimais são utilizados para entrada de dados auxiliares, particularmente em conjunto com outros periféricos de entrada. Os teclados numéricos projetados para serem utilizados com um toque por vez (um dedo por vez) não são espaçados como os teclados alfanuméricos, porém possuem todas as teclas colocadas em um mesmo plano. A figura 4 nos mostra um teclado hexadecimal contendo, portanto, 16 teclas, frequentemente utilizado em microcomputadores de função limitada, para entrada de programas diretos.

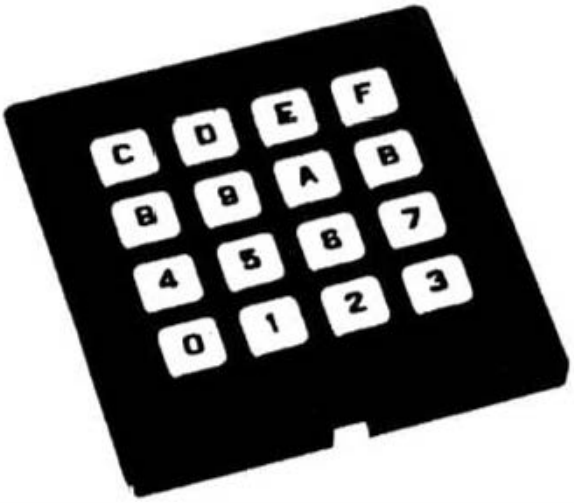
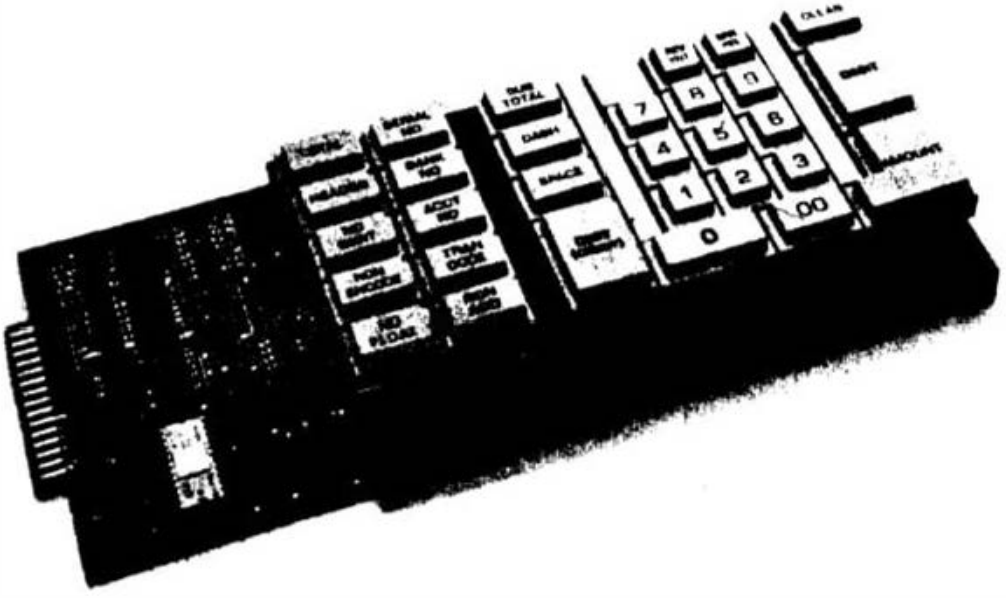


FIGURA 04

c) Teclados para fins especiais: são teclados especialmente desenvolvidos para terminais de fins especiais. Como exemplo, podemos citar os teclados utilizados nos terminais de transações de caixas de bancos (Veja figura 5).

FIGURA 05



Eventualmente, poderemos nos deparar com algum teclado específico que engloba os elementos destas três categorias. Como exemplo, podemos citar um teclado alfanumérico, usado em conjunto com uma impressora, no qual pode-se incluir, também, um teclado numérico, em separado, com 10 teclas, para facilitar a entrada de dados puramente numéricos e teclas de funções especiais, para controlar o espaçamento do cabeçote de

impressão, o movimento do comprensor e o espaçamento vertical do papel (veja figura 3).

COMUTAÇÃO:

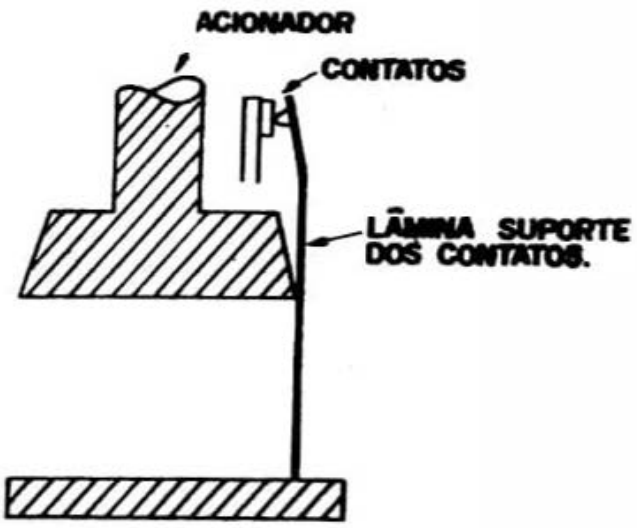
Quando uma tecla de um certo teclado é pressionada, esta opera momentaneamente alguns tipos de comutação elétrica. Esta pode ser uma chave mecânica convencional, semelhante à usada na comutação de uma lâmpada, ou algum outro método de afetar uma mudança no estado elétrico.

Normalmente a comutação de um estado elétrico para outro, em um teclado convencional, é realizada por intermédio de uma chave elétrica. Portanto, podemos dizer que o componente básico de um teclado é uma chave ou pulsador individual. Preferimos nos referir mais ao termo pulsador pelo fato deste nos dar a idéia de, ao pressionarmos a tecla, gerar um pulso, ou uma certa quantidade de pulsos.

Geralmente, cada tecla controla um simples interruptor,o qual permanece aberto. Nestas condições, dizemos que o pulsador está descansando. Vejamos então alguns tipos de pulsadores mais comuns, uma vez que existe uma grande variedade de pulsadores.

a) Pulsadores mecânicos: os interruptores mecânicos,hoje em dia, são ainda utilizados devido a sua economia. Na figura 6, temos representado o esquema básico de um pulsador mecânico.

FIGURA 06



Note que o movimento mecânico atua diretamente sobre os contatos. Esse tipo permite, sem grandes complicações, configurações de contactos múltiplos.

b) Pulsador mecânico de lâmina flexível: este tipo de interruptor é formado por uma série de lâminas sobrepostas. Seu funcionamento se baseia na deflexão de um diafragma flexível, com o seu lado inferior metalizado, o que permite estabelecer o contato com o circuito impresso, através de aberturas praticadas em um separador dielétrico.
Uma cobertura de silicone protege os contatos contra possíveis contaminações ou elementos que possam danificar os contatos. Algumas versões mais econômicas empregam lâminas flexíveis de silicone condutor, substituindo, assim, a cobertura protetora e o diafragma metalizado. Em alguns casos, emprega-se uma base serigrafada de tinta condutora, para substituir a face metalizada do diafragma flexível. Este tipo de chave, ou pulsador, é muito empregado na construção dos teclados de membrana. A figura 7 nos mostra alguns detalhes construtivos de um pulsador de lâmina flexível.

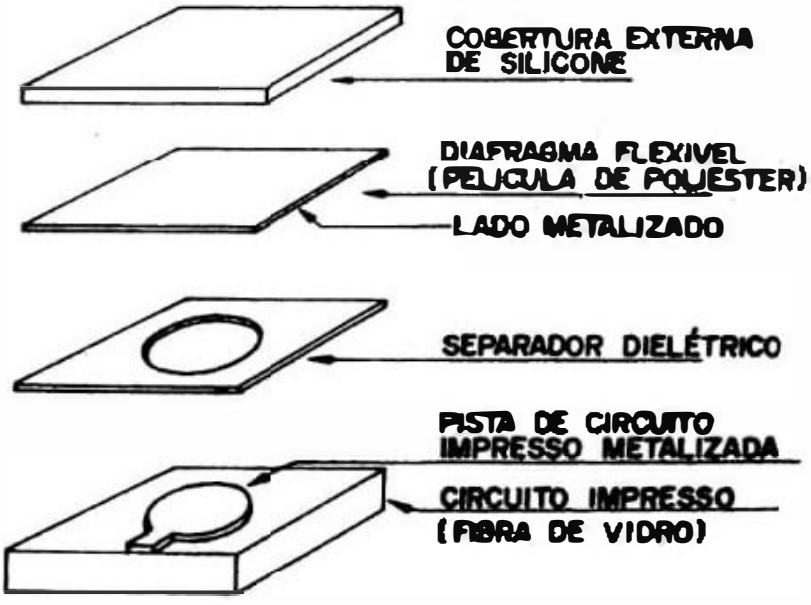
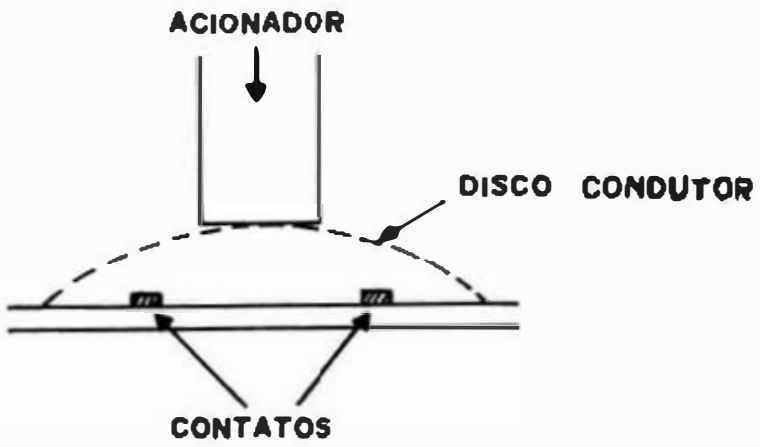


FIGURA 07

c) Pulsadores mecânicos de abôbadas: estes pulsadores(veja figura 8) utilizam uns discos metálicos embutidos

em forma de uma "casca" esférica que, ao serem comprimidos, passam a uma condição invertida, estabelecendo o contato requerido.

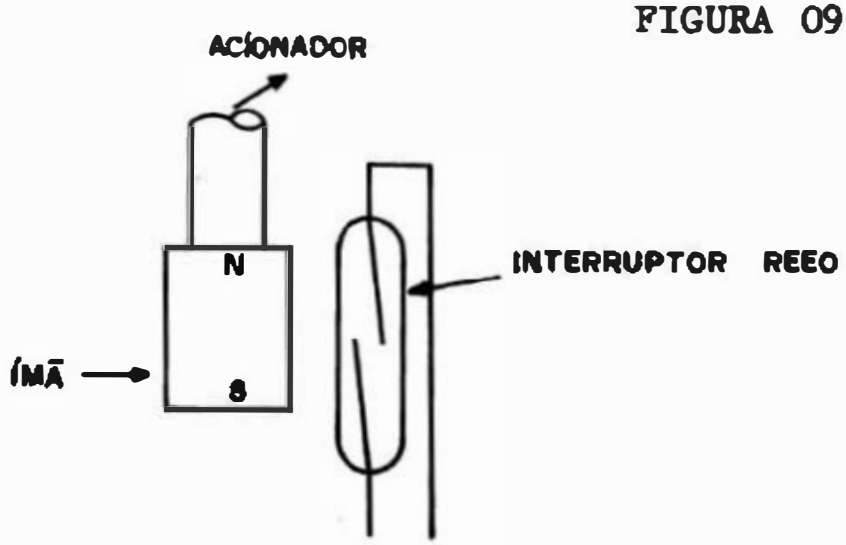
FIGURA 08



d) Pulsador reed: estes pulsadores são constituídos por contactos imersos em uma atmosfera inerte, selados por uma cápsula de vidro hermeticamente fechada (veja figura 9).

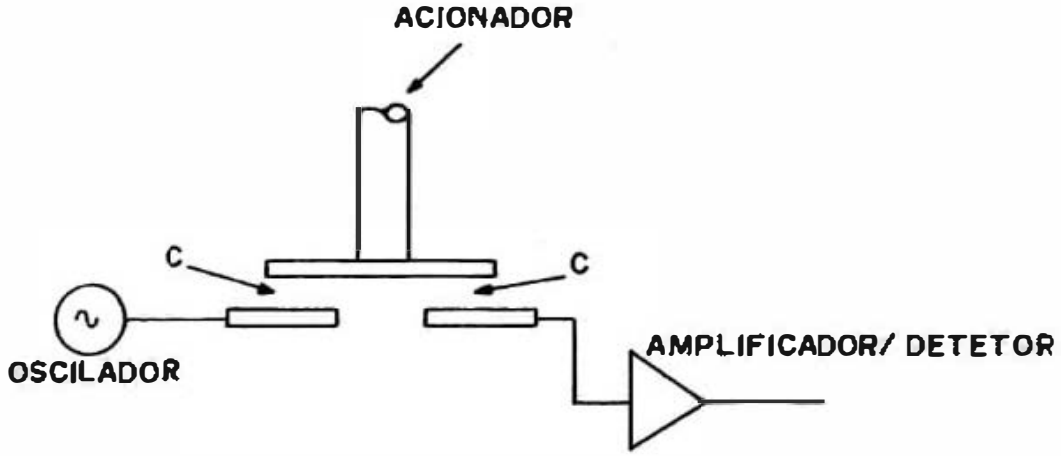
O movimento da tecla desloca um pequeno ímã permanente, provocando, assim, o fechamento dos contatos. Devido à ação indireta de acionamento dos contatos, não se transmite para estes sobrecargas mecânica, capazes de provocar fadiga e desgastes prematuros. Devido a este fato e devido à hermeticidade do encapsulamento, impedindo a contaminação dos contatos, esse tipo de pulsador oferece uma vida útil em torno de 5 vezes maior, se comparada com a vida útil de um pulsador mecânico comum.

FIGURA 09



e) Pulsadores capacitivos: este tipo de pulsador se baseia na mudança da capacitância de um capacitor. Na figura 10 apresentamos um esquema básico deste tipo de pulsador. Este pulsador emprega duas superfícies vizinhas sobre um mesmo circuito impresso. Estando uma delas excitada por um sinal alternado, produzido por um oscilador, ao fazermos aproximar paralelamente uma placa condutora sobre ambas as superfícies, provoca-se um acoplamento entre elas, o que faz com que apareça, na saída, uma fração do sinal alternado, aplicado na entrada.
O sinal de saída do pulsador deve ser convenientemente amplificado e convertido em níveis lógicos.

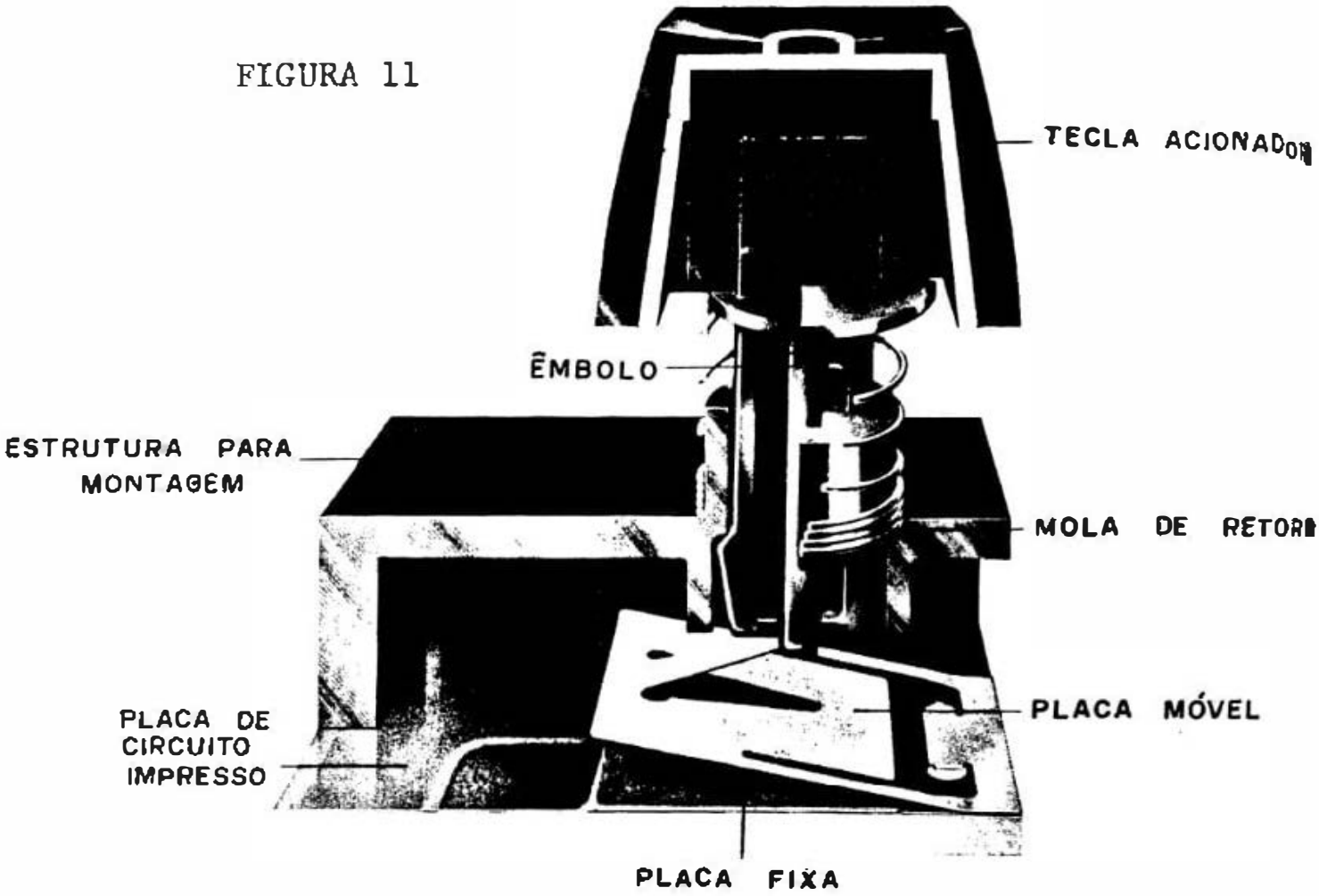
FIGURA 10



Os pulsadores capacitivos oferecem uma elevada fiabilidade, decorrente dos interruptores sem contatos móveis. A durabilidade destas chaves (ou pulsadores) é da ordem de 10^8 operações. A figura 11 nos mostra, com maiores detalhes, um pulsador de capacitância em um teclado.

Existem outros tipos de pulsadores empregados na construção de teclados. Entre eles,podemos citar os pulsadores de núcleos magnéticos e os pulsadores de efeito Hall. Porém, os pulsadores mais comuns são os que foram apresentados nesta lição. Dando continuidade ao nosso assunto, vejamos agora o conceito de tecla e os vários perfis de teclados.

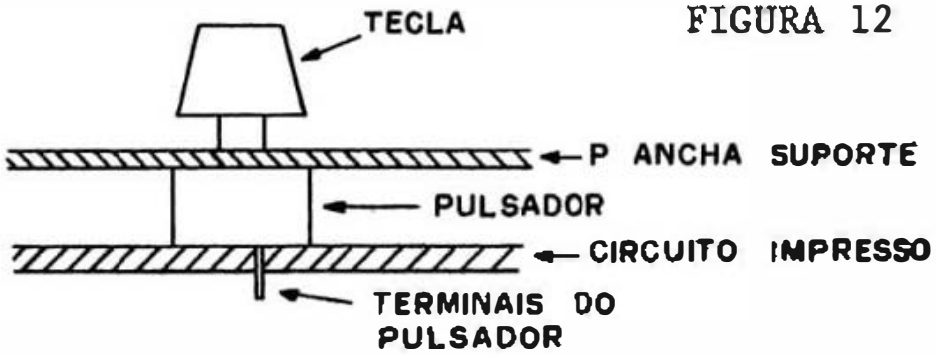
FIGURA 11



TECLAS:

O elemento unitário de um teclado é denominado tecla. Desta forma, podemos dizer que o teclado é formado pelo conjunto pulsador mais tecla, que geralmente é chamado de tecla. A principal função da tecla é acionar o pulsador, porém, fora esta função, a tecla é imprescindível para se identificar adequadamente cada pulsador,bem como para oferecer uma correta superfície de atuação, apropriada às características fisiológicas das mãos e dedos do operador. A figura 12 nos mostra as partes de uma tecla comum.

FIGURA 12



PERFIL:

O conjunto formado pela superfície superior de cada tecla pode assumir configurações distintas, baseadas nas preferências e aplicações desejadas. Normalmente, os teclados são montados de maneira tal a formarem um ângulo entre 10º e 13º com a horizontal.

A interação entre orientação de teclas e inclinação geral do teclado dá lugar às várias configurações, as quais podem resumir-se em três tipos fundamentais:

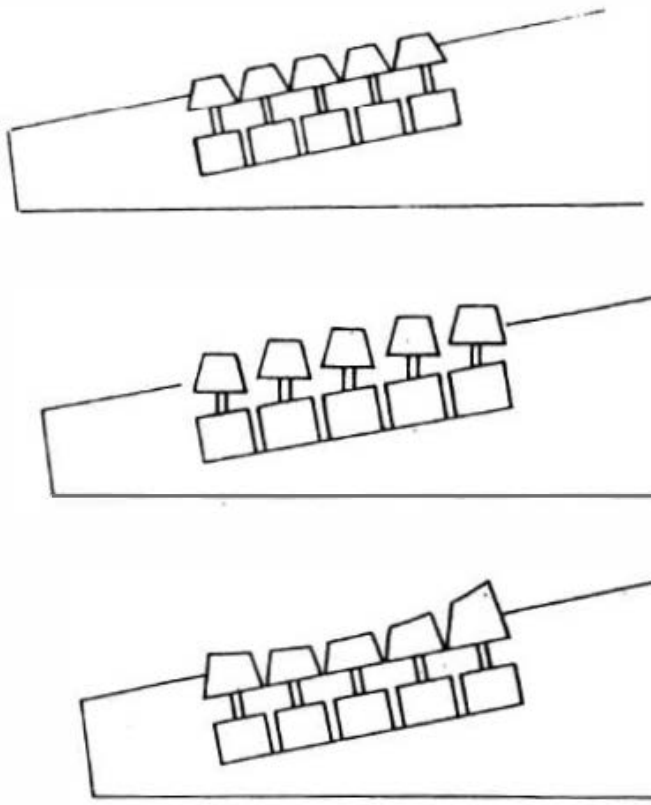
- Inclinado (Sloped)
- Escalonado (Stepped)
- Esculpido (Stulptured)

O teclado inclinado (figura 13-a) é formado por um conjunto de pulsadores e teclas totalmente verticais que, posteriormente, sofrerão um giro, até atingirem a inclinação desejada. Note que as superfícies superiores das teclas são coplanares.

Já, o teclado escalonado (figura 13-b) é formado por um conjunto de pulsadores e teclas não alinhados originalmente, mas sim formando um ângulo entre 10º e 12º. Ao girarmos o conjunto para que este assuma a inclinação geral desejada, as superfícies superiores das teclas ficam na horizontal, formando assim, filas separadas uma da outra, ou escalões.

O teclado esculpido (figura 13-c) apresenta uma certa semelhança com o teclado escalonado, com a diferença de que os ângulos iniciais dos pulsadores e teclas não são comuns à totalidade dos elementos. O conjunto adota, como um todo, um perfil curvado.

Os teclados inclinados e escalonados são mais comuns e, segundo diversos estudos, esses perfis não apresentam substanciais benefícios com relação, à velocidade, segurança ou comodidade do operador. Já o teclado esculpido está justificado, por razões ergonômicas, de modo que este realmente apresenta uma maior segurança, melhor conforto para o operador e maior velocidade, resultando, portanto, em um maior rendimento em aplicações de trabalho contínuo.



a)

b)

c)

FIGURA 13

Modos de operação:

Quanto aos modos de operação, um teclado pode ser:

- monomodo;
- múltiplomodo.

Nos teclados monomodos, cada tecla gera somente um único código. Já nos teclados de múltiplos modos, cada tecla pode gerar dois ou mais códigos. Esta geração de dois ou mais códigos vai depender, logicamente, de como se estão utilizando as teclas seletoras.

O número de códigos gerados, ao se acionar uma tecla, é definido como sendo o número de modos do teclado.

Os teclados monomodos são preferidos, quando o número de códigos é reduzido. Os de dois modos podem ser utilizados para se dispor de maiúsculo e minúsculo alfabéticos.

Normalmente,os teclados de múltiplos modos reduzem a produtividade do operador, posto que este deve utilizar uma das mãos para ativar as teclas seletivas. Porém, a vantagem apresentada por este tipo de teclado consiste na redução do número de teclas, sendo este fator primordial,quando se exige ele

vada velocidade de entrada de dados.

Os códigos "estander", como: ASCII ou EBCDIC são particularmente adequados para os teclados de múltiplos modos, tendo em vista que a própria estrutura desses códigos nos proporciona obter os códigos gerados pelas teclas, com a modificação, tão somente, de um ou dois bits.

CODIFICAÇÃO:

A função do teclado é entregar para o microprocessador (ou processador) caracteres apropriadamente codificados para o processamento. Assim, a codificação consiste em numerar, de forma binária, cada um dos distintos códigos emitidos pelo teclado, de tal modo que o número total de bits necessários - para expressar qualquer código não supere o limiar de operação.

A codificação mais usada para os teclados numéricos pequenos é a hexadecimal. No caso de teclados alfanuméricos, a codificação é ampliada, sendo o código ASCII o mais utilizado.

CONEXÃO SIMPLES:

Em se tratando de teclado de poucas teclas, como,por exemplo,um teclado numérico e sendo cada tecla constituída - por uma chave de um pólo, com dois terminais ligados a um CI, o número total de ligações elétricas seria igual a duas vezes o número de teclas do teclado em questão. Partindo-se deste raciocínio, um teclado hexadecimal de 16 teclas necessitaria de 32 ligações elétricas. Desta forma, torna-se necessário explorar eletronicamente 16 pares de terminais das teclas para se determinar qual tecla foi pressionada. Uma vez identificada qual tecla foi acionada, o sinal é codificado e então é enviado para o processamento. A figura 14 nos mostra um exemplo de aplicação deste método, onde temos a codificação de um teclado de 10 teclas, o qual consta de 10 pulsadores, um codificador BCD do tipo 74147 e um grupo de portas para gerar o sinal de validade (VAL).

Muito embora este método seja aplicado na prática, o método mais prático de se codificar os "fechamentos das chaves" (acionamento das teclas) é o de conexão matricial.

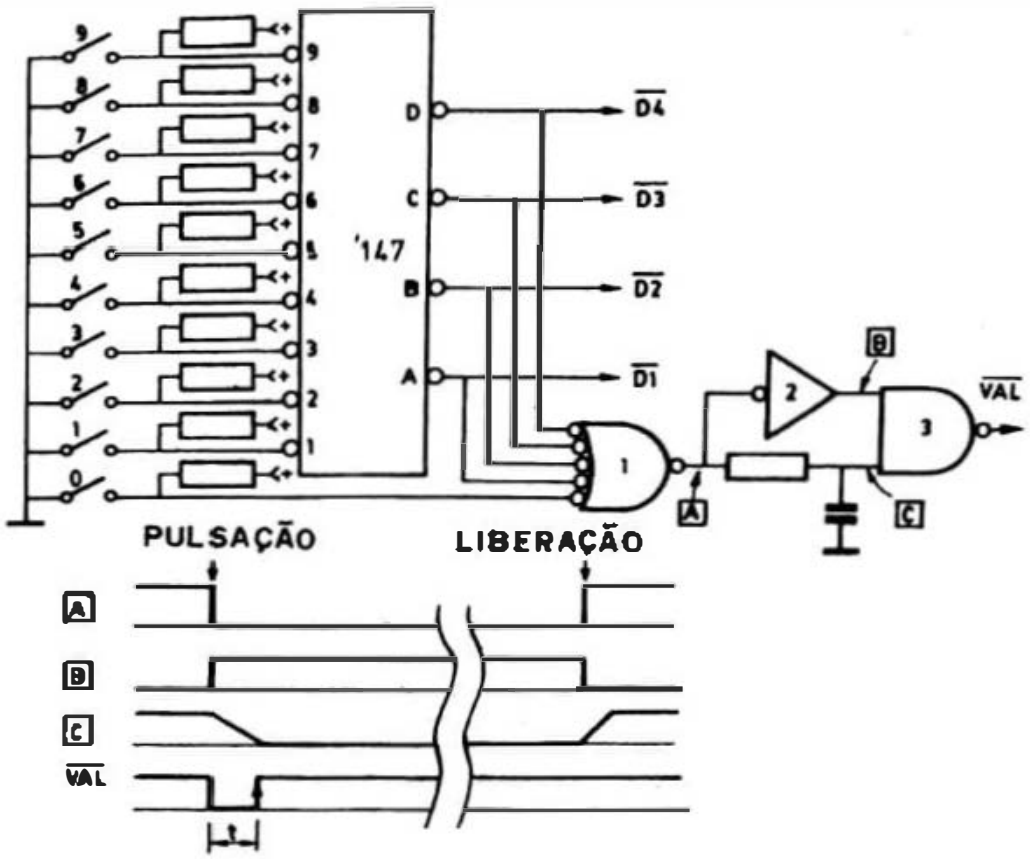


FIGURA 14

CONEXÃO MATRICIAL:

Quando o número de teclas é muito elevado (é o caso dos teclados alfanuméricos), o método de conexão simples não é muito aconselhável. Nestes casos, a técnica mais usada consiste em conectar as teclas em forma matricial, de tal modo que o número total de teclas conectadas seja igual ao número de interconexões. A figura 15 nos mostra o circuito utilizado na codificação de um teclado, conectado sob a forma matricial. Este teclado permite a conexão de até 128 teclas, distribuí - das em uma matriz de 8 linhas por 16 colunas.

Mediante 3 codificadores do tipo 74148, obtemos um código de 7 bits. Com o objetivo de adaptar a codificação para um outro código desejado, assim como para se obter um teclado de 4 modos, se injetam os 7 bits de saída dos codificadores, juntamente com as saídas das teclas estáticas "shift" e "Control" nas entradas de uma memória PROM de 512 x 8 bits. Desta forma, obteremos uma saída transcodificada de 8 bits.

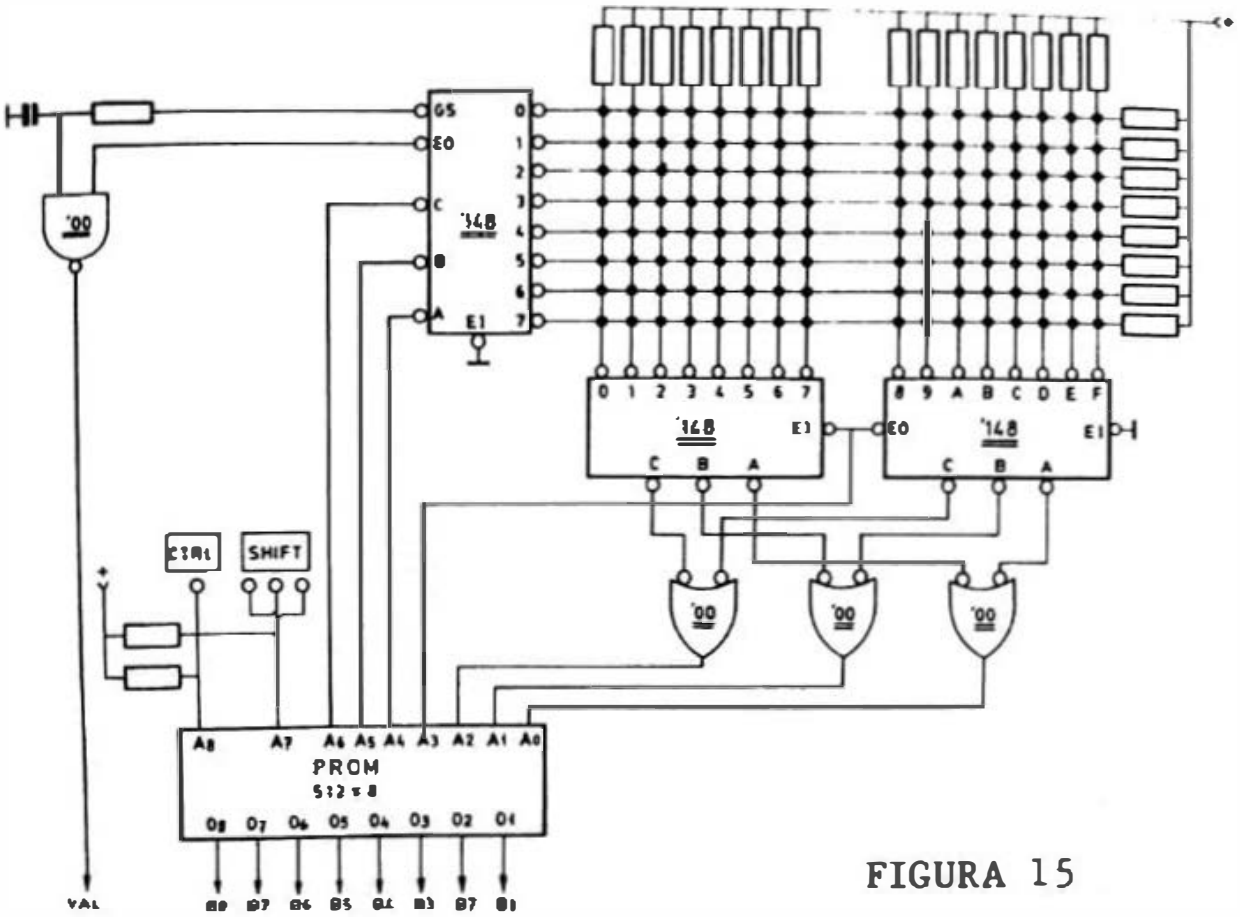


FIGURA 15

CODIFICAÇÃO POR INTERMÉDIO DE MICROPROCESSADOR

Esta técnica consiste na utilização de um microcomputador, como elemento de lógica ativa, na codificação de teclados. Tal técnica permite, com um mínimo de componentes, realizar funções que até a pouco eram raramente levadas a efeito, por ação direta de um teclado. Entre essas funções, podemos citar: exploração seqüencial; proteção contra pulsos simultâneos; transcodificação; multiplomodos; seleção de modo complexo; saídas em paralelo ou em série; auto-repetição em teclas selecionadas; repetição por tecla "REPETE", independente de certas teclas ou funções; autorização ou inibição total ou parcial do teclado; geração de sinal audível para realimentação acústica; paridade; detecção de erros de operação; etc.

A título de exemplo, a figura 16 nos mostra o esquema de uma codificação por microprocessador, utilizando exploração seqüencial de teclas de efeito Hall, cujo esquema interno é mostrado na figura 17.

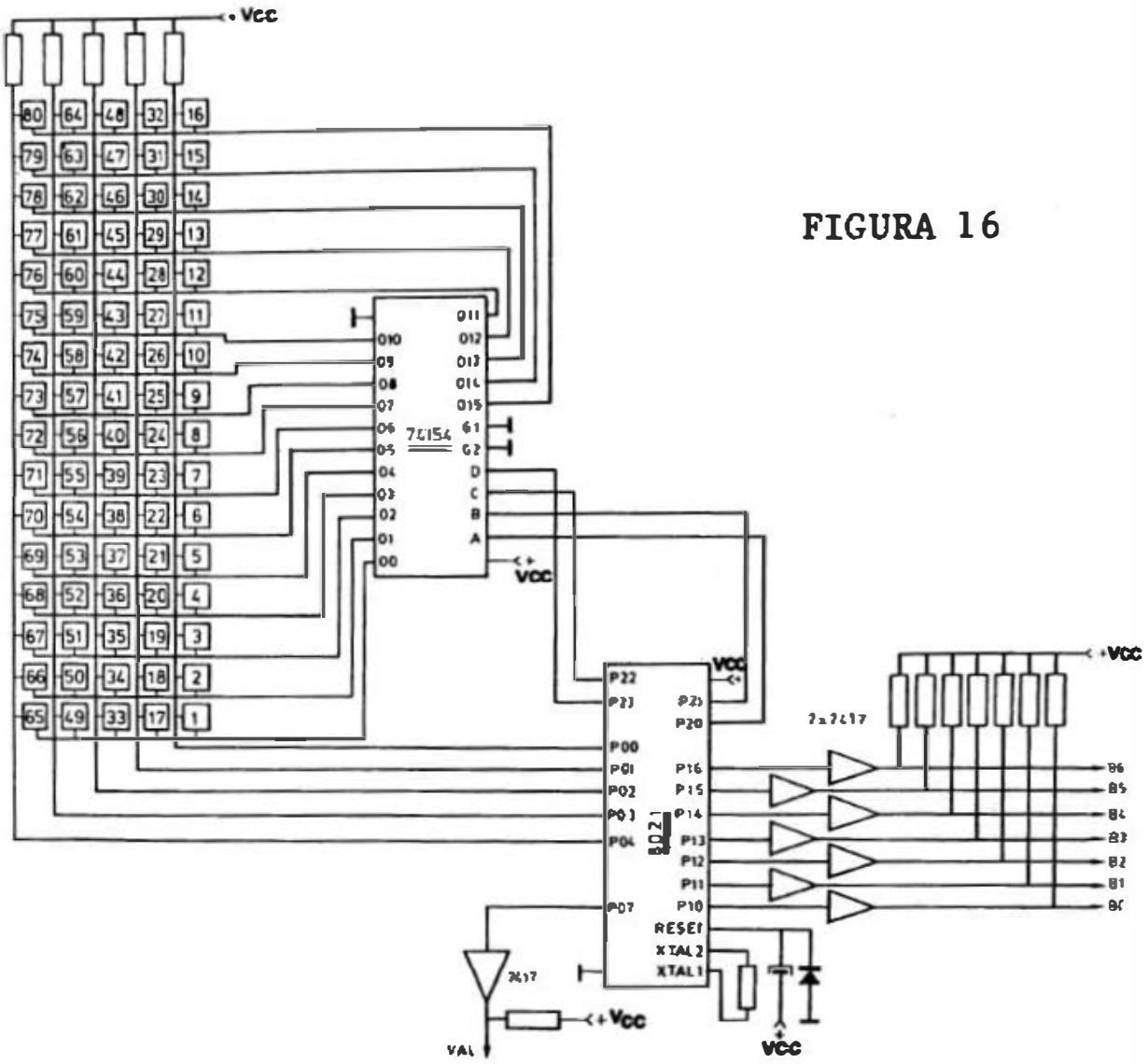


FIGURA 16

O microprocessador utilizado é um INTEL 8021. Este circuito opera a uma tensão única de +5 volts, processa palavras de 8 bits, possui uma memória ROM de 1K palavras, uma memória RAM de 64 x 8 bits, 21 portas de entrada/saída e um temporizador/contador. Tudo isto em uma única pastilha, colocada em uma cápsula com 28 terminais.

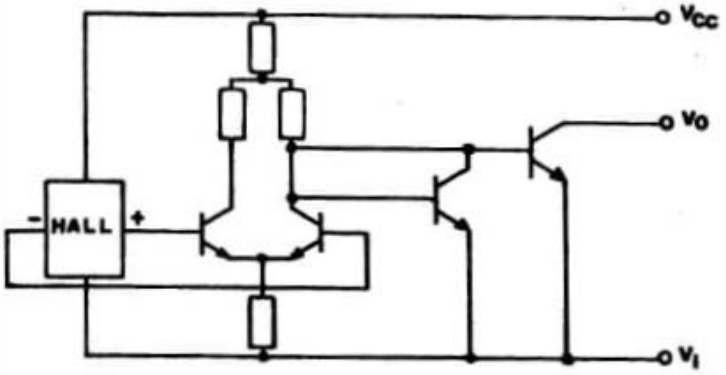


FIGURA 17

COMENTÁRIOS:

Com isto, encerraremos o assunto teclado, sendo que qualquer informação adicional que julgarmos necessária, será fornecida oportunamente.

Na próxima lição estudaremos mais alguns dispositivos periféricos.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-7

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Todo periférico, que é tido como fonte de informação, pode ser classificado como:

a) periférico de saída;
b) periférico de entrada;
c) periférico de varredura;
d) N.R.A.

2) Se considerarmos os periféricos como sendo dispositivos utilizados para realizar a entrada/saída de informações, poderemos classificá-los como:

a) periféricos que executam a E/S verdadeira;
b) periféricos que executam a E/S de armazenamento;
c) os ítens a e b estão corretos;
d) N.R.A.

3) Um exemplo de periférico que executa a E/S de armazenamento é:

a) as memórias de massa;
b) as impressoras;
c) os terminais de vídeo;
d) o teclado.

4) Quanto à função, os teclados podem ser do tipo:

a) teclados alfanuméricos;
b) teclados numéricos;
c) teclados para fins especiais;
d) os ítens a, b e c estão corretos.

5) Um pulsador pode ser entendido como:

a) uma chave ou interruptor;
b) um amplificador;
c) um oscilador;
d) N.R.A.

6) Os pulsadores constituídos por contatos imersos em uma atmosfera inerte, selados por uma cápsula de vidro hermeticamente fechada, recebem o nome de:

a) pulsador de lâmina flexível;
b) pulsador de núcleos magnéticos;
c) pulsador mecânico de abóbadas;
d) pulsador reed.

7) A durabilidade de um pulsador capacitivo é da ordem de:

a) 10^{28} operações;
b) 10^{18} operações;
c) 10^{8} operações;
d) N.R.A.

8) Quanto ao perfil, os teclados podem ser:

a) alfanuméricos, numéricos e inclinados;
b) inclinado, escalonado e esculpido;
c) escalonado, esculpido e numérico;
d) N.R.A.

9) Quanto ao modo de operação, os teclados podem ser:

a) monomodo e múltiplomodo;
b) monomodo e flexível;
c) de lâmina rígida e múltiplomodo;
d) N.R.A.

10) Na conexão matricial, o número total de teclas conectadas deve ser:

a) maior que o número de interconexões;
b) menor que o número de interconexões;
c) igual ao número de interconexões;
d) N.R.A.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANA

LIÇÃO MC-8

DISPOSITIVOS PERIFÉRICOS II

INTRODUÇÃO

Dando continuidade ao assunto: periféricos, apresentaremos nesta lição alguns periféricos de entrada, como, por exemplo, os digitadores de posição, joystick, os leitores ópticos, reconhecimento pela fala, e outros.

DIGITADORES DE POSIÇÃO

Trata-se de um dispositivo periférico de entrada, utilizado para introduzir dados gráficos. Alguns tipos de digitadores de posição, operados manualmente, são apresentados a seguir:

- digitadores de imagem;
- joysticks;
- bolas de curso;
- canetas luminosas ou ótica, usadas em conjunto com os displays de vídeo;
- etc.

De um modo genérico, os digitadores de posição manuais possuem, basicamente, três (3) partes:

1 - Um objeto que deve ser movido pela ação da mão do homem, por exemplo, a alavanca de um joystick, ou o

"corpo" de uma caneta luminosa ou ótica, etc;

2 - Um transdutor para sentir o movimento que está sendo feito ou a posição do objeto;

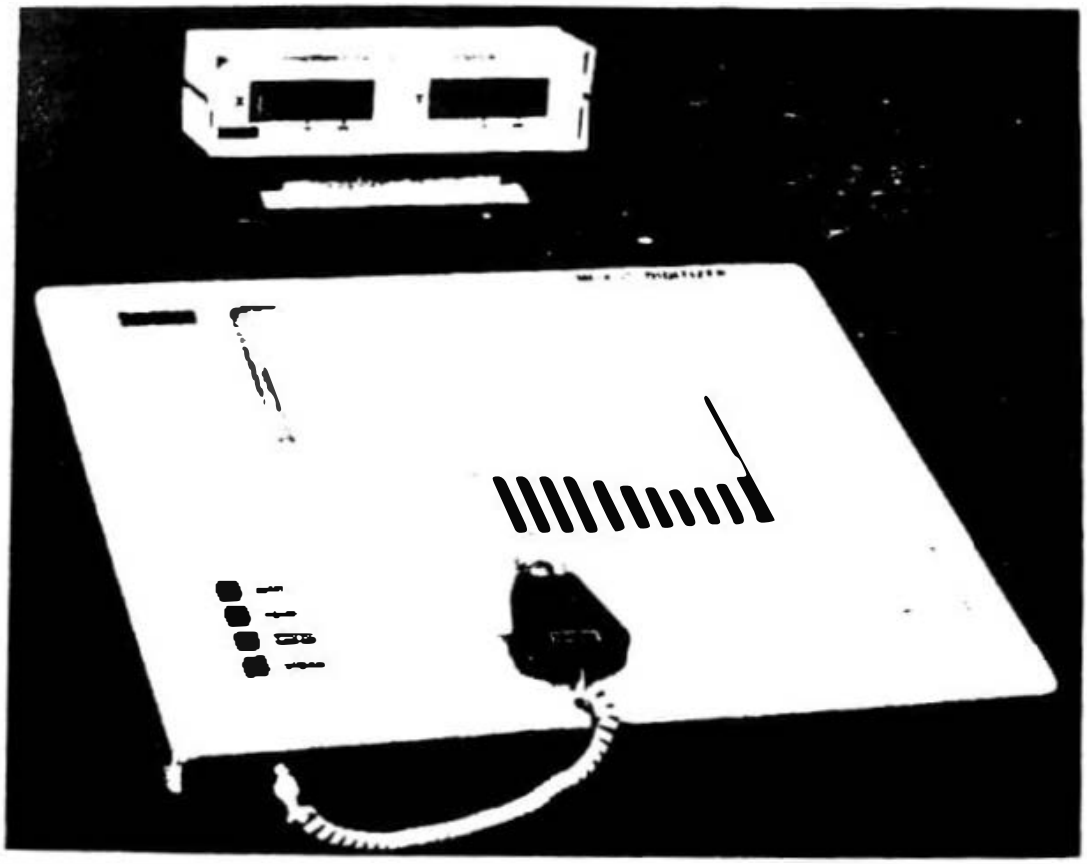
3 - Um dispositivo capaz de converter o movimento ou a posição sentida para a forma digital.

Vejamos então alguns desses digitadores de posição.

DIGITADORES DE IMAGEM

O processo de funcionamento de um digitador de imagem é o inverso do processo de funcionamento de um traçador de gráfico (ploters). Nos traçadores de gráficos, uma vez fornecidas as coordenadas XY e os comandos do traçado, passam a desenhar linhas específicas, criando uma imagem. Por outro lado, os digitadores de imagem operam sobre uma imagem, ou um desenho, ou uma fotografia, que já está pronta e transfere as informações dessas imagens para o computador. Na figura 1, apresentamos um digitador de imagem.

FIGURA 01



Digitador de imagem

Observando a figura 1, podemos notar que o operador utiliza-se de um cursor manual para indicar a posição do dado a ser digitalizado.

Usualmente o digitador de imagem executa a digitalização do ponto com maior precisão do que a combinação do monitor de vídeo com a caneta sensível à luz. O digitador de superfície plana é, também, muito utilizado para a digitação de gráficos contínuos. Conseqüentemente, selecionar qual método de digitalização de imagem é o mais adequado para uma certa aplicação, dependerá, basicamente da precisão do ponto de resolução e dos fatores humanos que afetam a produtividade da digitação.

Os digitadores de imagem podem ser distribuídos em três categorias:

1 - Grandes digitadores de mesa, tipicamente usados para digitar dados de grandes desenhos ou de grandes imagens;

2 - Pequenos digitadores de mesa, também conhecidos por "mesas de dados";

3 - Digitadores para fins especiais, por exemplo, o digitador utilizado para digitar gravações de cartões de barra, onde o cartão de barras move-se na direção X, com uma velocidade constante, sob um cursor montado em uma barra que se move somente na direção Y.

A seguir, apresentamos os componentes básicos de um digitador de imagem:

a) Um cursor, que deverá ser movido pela ação da mão do operador e os controles de entrada de dados, normalmente chaves do tipo push-button;

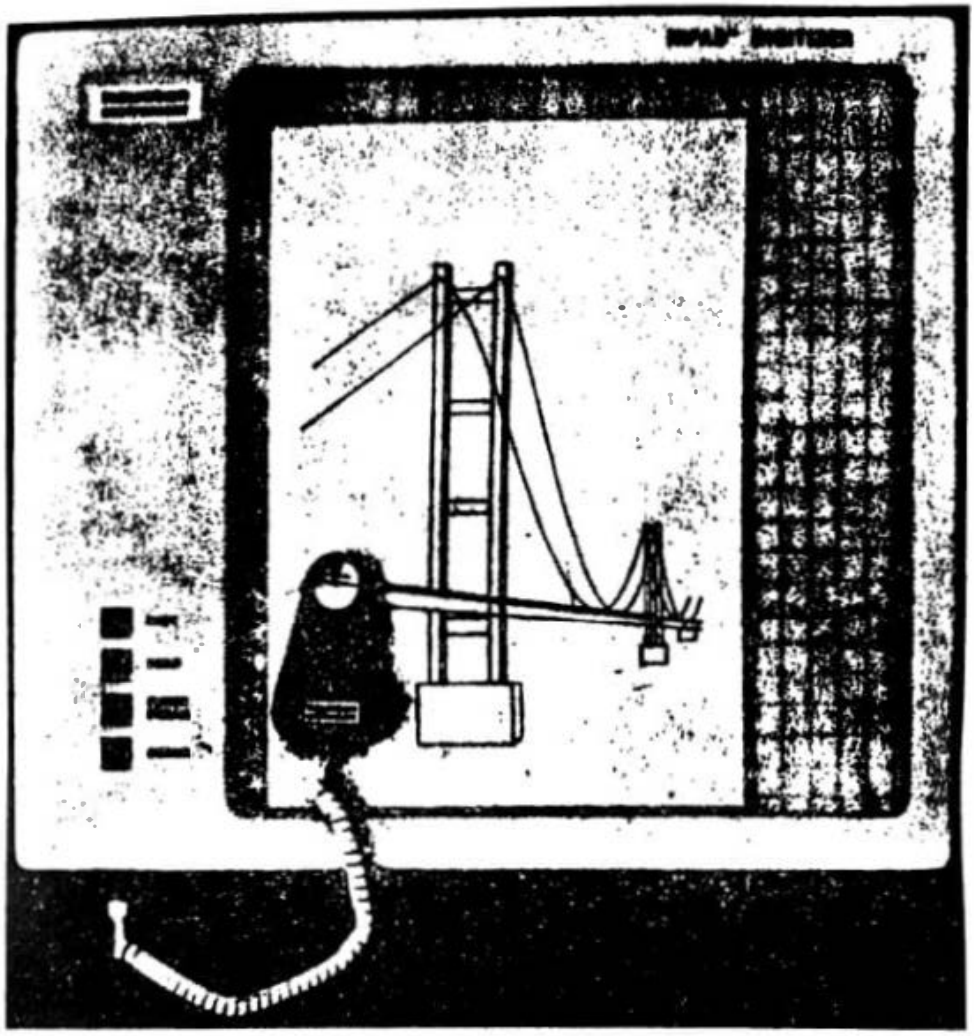
b) Uma superfície de traçamento (prancha) para sentir a posição do cursor (ambas as superfícies, do cursor e de traçamento, trabalham juntas, como um transdutor de posição);

c) Um circuito controlador e um circuito de interface;

d) Sistema de exibição visual de coordenada opcional;

e) Um gravador, fora da linha opcional.

Um ponto será especificado, toda vez que se mover o cursor de uma posição para outra desejada. O cursor tipo "olho de boi" é o mais comum e é constituído por um círculo plástico transparente, com fibras cruzadas, utilizadas para localizar um ponto com segurança. Na figura 2, mostramos um cursor do tipo "olho de boi". Um outro cursor muito utilizado é o do tipo caneta.



Cursor tipo "olho de boi"

FIGURA 02

Quando o cursor estiver em posição (ou quando está se movendo sobre uma linha ou uma curva), o operador ativa uma chave, sinalizando a entrada de dados. Normalmente, a chave é construída dentro do próprio cursor, de modo que, ao se pressionar o cursor para baixo, a chave de entrada de dados é ativada.

Um outro transdutor de posição, menos usado, é o chamado cursor "RATO", que recebe este nome devido a sua semelhança com um rato. Este cursor é montado sobre rodas, para sentir o movimento direcional X e Y. Conforme o cursor se movimenta

sobre a superfície da imagem, a rotação das rodas do cursor são transformadas em pulsos X e Y, sinalizando, desta forma, a nova posição XY do cursor RATO, com relação a um referencial previamente adotado.

A resolução (grau de precisão) de um digitalizador de mesa varia de 3,9 até 7,9 aproximadamente.

JOYSTICK:

Um joystick é um "manobrador" vertical, semelhante ao controle STICK, usado em aeronaves. Um operador move o joystick para frente e para trás, para a esquerda ou para a direita, ou ainda para alguma combinação destas duas direções, dentro da escala completa de 360º. A figura 3 nos mostra um joystick típico de duas dimensões.

FIGURA 03

Joystick

Um resistor variável é usado como transdutor de posição. Dois potenciômetros representam as direções X e Y do joystick manual. Um potenciômetro representa a posição posterior e anterior, enquanto que o outro potenciômetro representa a posição esquerda e direita do joystick. Desta forma, uma posição

XY qualquer do joystick é identificada pelas resistências XY dos potenciômetros. Assim, através de um sistema de interface, conseguimos fazer com que o joystick e os seus componentes digitem o movimento original, de tal modo que o computador passe a processar esses movimentos.

Os joystick são tipicamente utilizados em sistemas de computadores, para controlar a posição X e Y dos objetos em um display de vídeo, para o controle manual de um movimento de canetas em um traçador XY, e em jogos recreativos, controlados pelo computador.

BOLA DE CURSO:

Trata-se de uma esfera (bola) com aproximadamente 51mm ou 76mm de diâmetro, sustentada por um suporte, tipo sino, no interior de uma caixa (veja figura 4). A esfera é movimentada através da ação dos dedos ou da palma da mão do operador.A rotação imprimida à esfera gera pulsos X e Y, através de um transdutor proporcional ao movimento da mesma. Através de um sistema de interface, esses pulsos são levados para um formato apropriado.

FIGURA 04

Bola de curso

Em muitos casos, utilizam-se os decodificadores ópticos como transdutores para bolas de curso.

As bolas de curso são usadas, tipicamente, em "displays

de radares", para introduzir dados de curso sobre um objeto rastreado. Fora essa aplicação específica, existe uma série de outras aplicações, uma vez que as bolas de curso executam as mesmas funções dos joysticks, podendo, portanto, substituir os mesmos.

RECONHECIMENTO DA FALA:

São dispositivos periféricos, utilizados para introduzir dados em um sistema de computador. A entrada de dados pelo reconhecimento automático da fala (ASR), também chamado de sistema de entrada de dados por voz (VDE), é válida tanto para aplicações voltadas para o Hobby, como para aplicações comerciais. O equipamento ASR, para a entrada de dados, é usado em uma grande variedade de aplicações comerciais e industriais, incluindo, por exemplo, o roteiro de empacotamento - controlado pela "voz" em um depósito automatizado, ou ainda a entrada de dados em um centro de controle de qualidade, conforme nos mostra a figura 5, na qual se pode observar o operador enviando informações para o computador, através do sistema de reconhecimento da fala.

Sistema de reconhecimento de fala

FIGURA 05

Em um sistema de reconhecimento de fala comercial, podemos distinguir três componentes principais:

1 - Entrada, via microfone e console do operador, com realimentação acústica ou visual de entrada de dados;

2 - Um microprocessador-controlador, incluindo um detector de característica acústica, com o programa de examinação e o processador de correlação da memória do tipo de referência;

3 - Interface de saída.

Uma palavra é uma articulação de som que dura de 0,15 a 2,0 segundos aproximadamente, separada de outras articulações por uma pausa de 100ms. As palavras faladas em um microfone produzem formas de ondas que representam a palavra, as quais serão analisadas pelas suas características acústicas e pelo espectro fonético. Os componentes fonéticos são os sons da fala elementar que formam as palavras. Já os componentes do espectro são as freqüências produzidas ao se pronunciar uma palavra específica e a energia contida em sub-faixas do espectro da freqüência da voz.

Estas características acústicas são convertidas de tensões analógicas para sinais digitais. O tipo do sinal digital que representa uma palavra falada é então comparado aos sinais digitais correspondentes das palavras de referência, contidas no vocabulário do sistema de reconhecimento automático da fala. A figura 6 nos mostra o diagrama de blocos funcional do

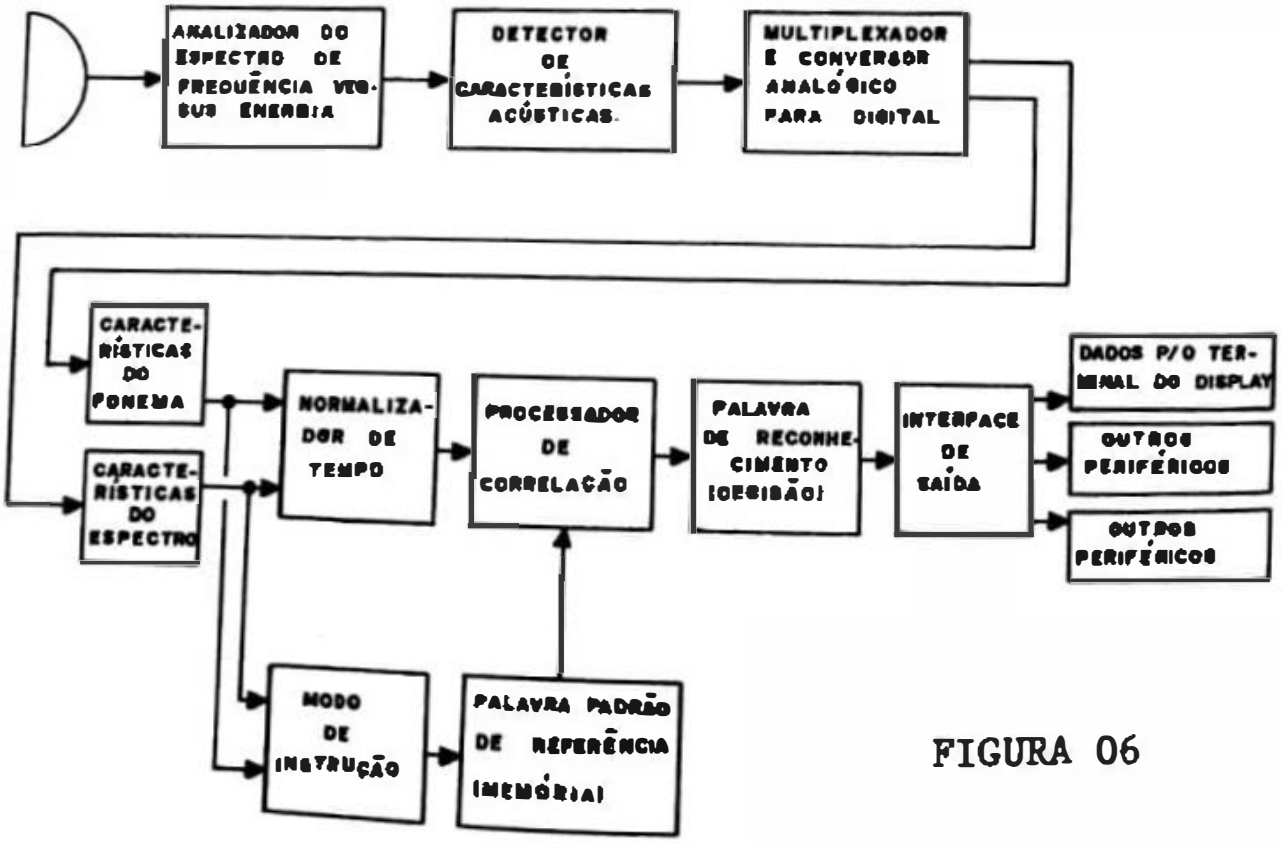


FIGURA 06

processo de reconhecimento pela fala.

No sistema apresentado na figura 6, o analisador de espectro trabalha como um filtro, dividindo a tensão gerada pelo sinal acústico para várias faixas de freqüências, onde as características do fonema e espectro do sinal são separadamente detectadas. Em seguida, cada característica separada é comutada para um conversor A/D, onde a palavra inteira é convertida em uma matriz de bits, ou em um bit de amostra.

Normalmente, as palavras do vocabulário do sistema de reconhecimento pela fala são estabelecidas, colocando-se o periférico de reconhecimento pela fala em um tipo de "treinamento", repetindo-se a mesma palavra cerca de 10 vezes. O bit de amostra médio para estas palavras de reconhecimento formam a palavra de amostra de referência para o uso subseqüente. Alguns sistemas já são projetados ativamente para modificar o bit de amostra de referência, baseado nas variações do bit de amostra durante o uso. Porém, a grande maioria das unidades de reconhecimento da fala é "treinada" para a voz do operador que irá utilizá-la.

LEITORES ÓPTICOS:

Este tipo de dispositivo periférico de entrada permite a entrada de dados, para o computador, de uma forma direta, a partir de uma fonte codificada.

Os sistemas de leitura óptica podem ser classificados como sistemas de "código de barra" ou de "execução do caracter óptico". Vejamos então algumas características a respeito dos mesmos.

LEITORES ÓPTICOS DE CÓDIGO DE BARRA:

Um sistema de leitura de código de barra é, basicamente, formado por:

1 - Um código de barra imprimido, o qual identifica o elemento;

2 - Um "varredor" ou leitor de código de barra, podendo ser um bastão ou uma caneta sensível à luz;

3 - Um decodificador eletrônico, capaz de converter os sinais elétricos do varredor para os caracteres do

código ASC II, ou seja, em níveis lógicos de tensão;

4 - Um circuito de interface, capaz de apresentar os sinais para uma UCP.

CÓDIGO DE BARRA:

Trata-se de um conjunto de barras paralelas espaçadas entre si,as quais servem para identificar os caracteres numéricos e alfanuméricos. Às vezes, a cor com que são feitas as barras também serve como parte da identificação em alguns códigos. Os códigos de barras mais freqüentemente utilizados são os códigos de produto universal (CPU), utilizados para codificar números e os códigos de elementos "N fora de M" utilizados na indústria e no comércio.

Apenas o código de barra CPU é fixado (imposto), pois trata-se de um padrão normalmente adotado. Já, os demais códigos de barra são flexíveis, adaptando-se às necessidades do usuário.

VARREDORES DE CÓDIGO DE BARRA:

Os varredores (bastão ou caneta sensíveis à luz) emitem luz. Quando a luz incidente sobre o código de barra é refletida de volta, esta incide sobre um dispositivo fotossensor e uma tensão proporcional ao tipo do código de barra é então produzida. Este fato ocorre quando temos o varredor se movendo sobre o código de barras ou quando temos o varredor fixo e os objetos marcados com os códigos de barra se movendo,como em uma linha de produção de movimento contínuo.

A figura 7 nos mostra o esquema de um bastão manual de cerca de 13mm de diâmetro. A luz de um led é focalizada sobre a superfície de leitura, através de uma esfera de safira. Por outro lado, a luz refletida pelo código de barras é, também, focalizada pela esfera de safira e dirigida para um fototransistor. A saída elétrica deste fototransistor é então amplificada. Desta forma,teremos o bastão gerando um sinal elétrico proporcional à luz refletida pelas barras e pelos espaços vazios entre as barras do código.

O circuito de interface e o de decodificação, normalmente, trazem um alto-falante o qual serve para indicar se o bastão está ou não realizando a leitura.

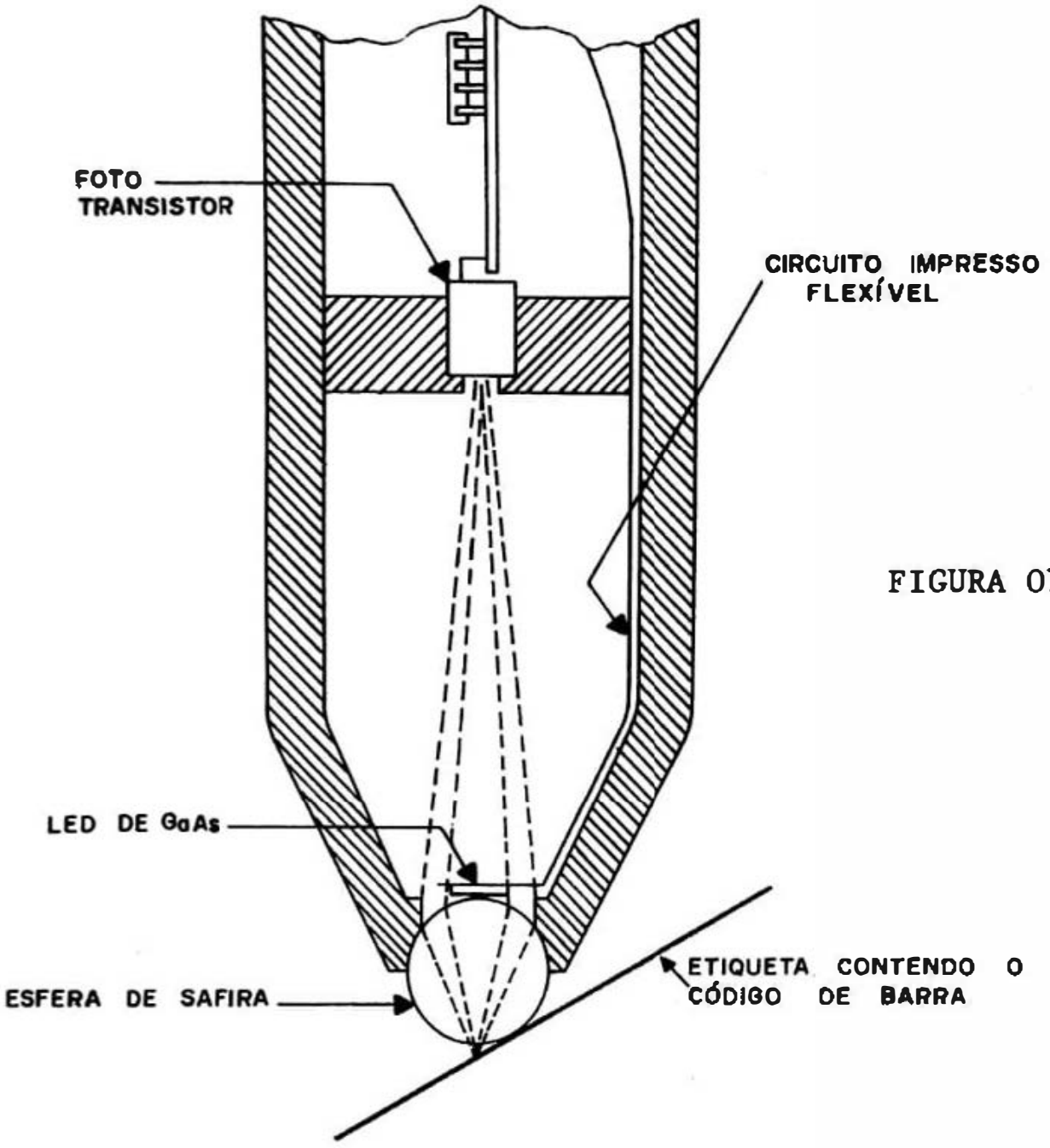


FIGURA 07

Esquema de um bastão manual

As aplicações deste tipo de dispositivo de leitura são muito variadas. Uma das vantagens deste sistema de entrada de dados está no fato de poderem ser acoplados a um modem para transmitirem dados de um ponto distante para a UCP (unidade central de processamento).

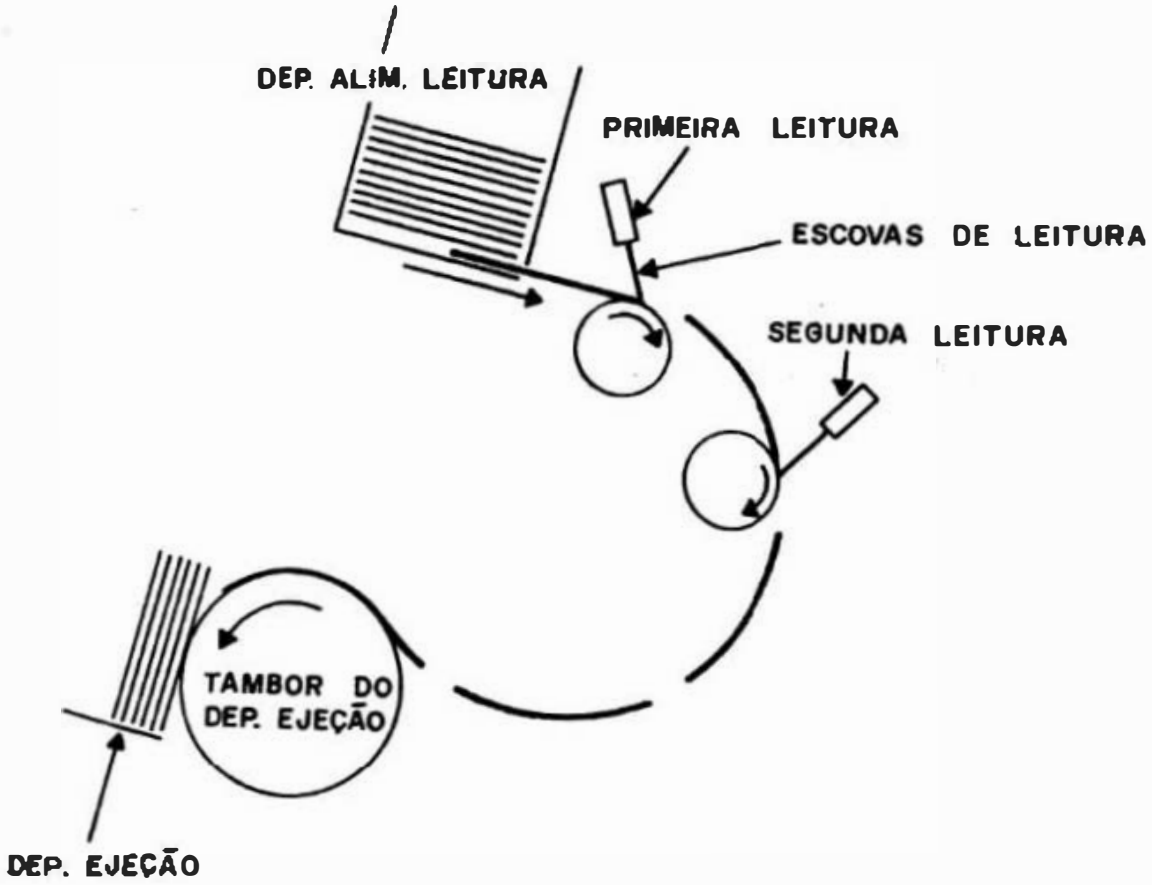
OUTROS DISPOSITIVOS PERIFÉRICOS DE ENTRADA

Apresentaremos agora, de uma forma bem resumida, a função e a aplicação de mais alguns dispositivos periféricos de entrada. Então vejamos:

a) Leitora de cartões

Trata-se de um periférico de entrada que permite a introdução, na memória central, de programas e de dados devidamente registrados em cartões perfurados. Este periférico foi muito utilizado até há pouco tempo, hoje não é muito utilizado devido à tendência atual de utilização progressiva de periféricos magnéticos.

A figura 8 nos mostra um esquema de uma leitora do tipo escova, na qual os cartões são movidos, mecanicamente, em um depósito de alimentação (dep. alim.)de cartões, através da unidade de alimentação de cartões e sob as escovas de leitura.



Esquema de uma leitora tipo escova

FIGURA 08

Nas leitoras ,tipo escova, as escovas percebem eletricamente a presença ou ausência de perfurações em cada coluna do cartão.

b) Leitora de fita

Trata-se de um periférico que foi muito utilizado para a introdução, na memória central, de programas e de dados registrados em fita (de papel) perfurada. Hoje, também, está sendo trocada por periféricos magnéticos, exceto quando isto não for possível.

c) Instrumentos analógicos de medida

São instrumentos feitos para medir uma determinada grandeza elétrica, apresentando o resultado na forma de uma tensão de valor proporcional ao da grandeza aferida. Normalmente este tipo de instrumento tem sua saída acoplada à entrada de um conversor A/D, para converter a tensão em um número binário que a represente.

Geralmente,este tipo de periférico é utilizado como fonte de telemedida no controle automático de processos.

d) Sensor analógico

Trata-se de um periférico de entrada, destinado a converter uma grandeza física numa tensão elétrica com valor proporcional ao da grandeza. Esse dispositivo é muito empregado no controle automático de processos como conversor de temperatura, de intensidades luminosas, de pressão, de dilatação, etc...

e) Sensor digital

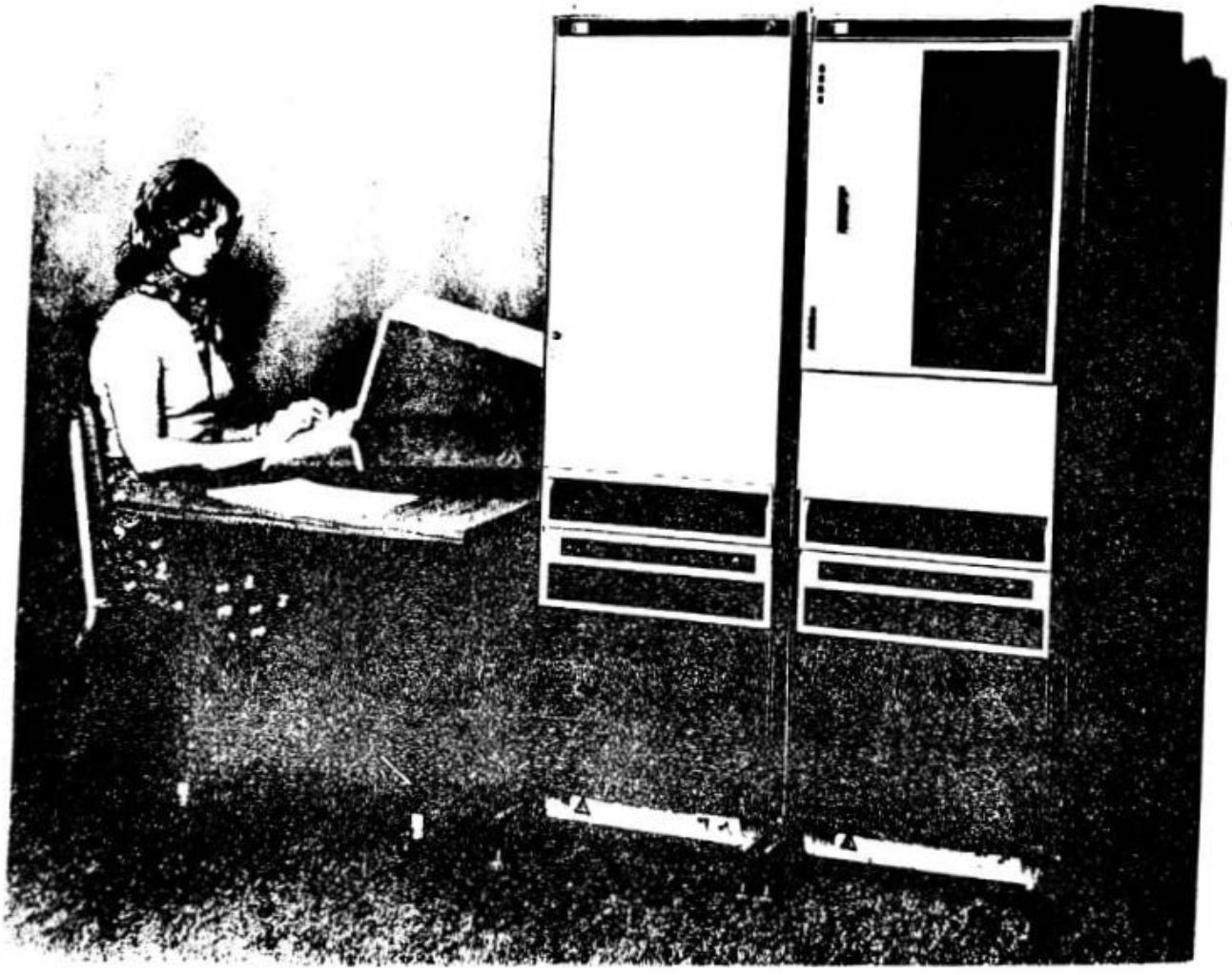
Este periférico de entrada destina-se a informar o microcomputador da presença ou da ausência de determinadas condições do mundo exterior.

O sensor digital também é muito utilizado no controle automático de processos, para sinalizar estados como, por exemplo, motor ligado ou desligado, disjuntor aberto ou fechado, porta aberta ou fechada, etc...

MEMÓRIAS AUXILIARES

As memórias auxiliares, tais como fitas magnéticas, fitas cassete (ou fitas de áudio), discos magnéticos e tambores magnéticos, são tidas como periféricos do tipo entrada/saída (E/S), uma vez que são dispositivos que podem ser fonte ou destino de informação.

Os sistemas de minicomputadores e microcomputadores, normalmente, não utilizam memórias auxiliares do tipo disco magnético. O princípio de funcionamento destas memórias auxiliares (memórias de massa) já foram estudadas em lições anteriores. A figura 9 nos mostra uma unidade de fita magnética.



Unidade de fita magnética

FIGURA 09

A figura 10 nos mostra uma unidade de memória auxiliar, do tipo disco magnético. O funcionamento do disco magnético é semelhante ao funcionamento de um toca-disco e já foi visto

em lições anteriores.

FIGURA 10

Unidade de disco magnético

Na figura 11, temos uma fotografia de uma unidade de disco fixo.

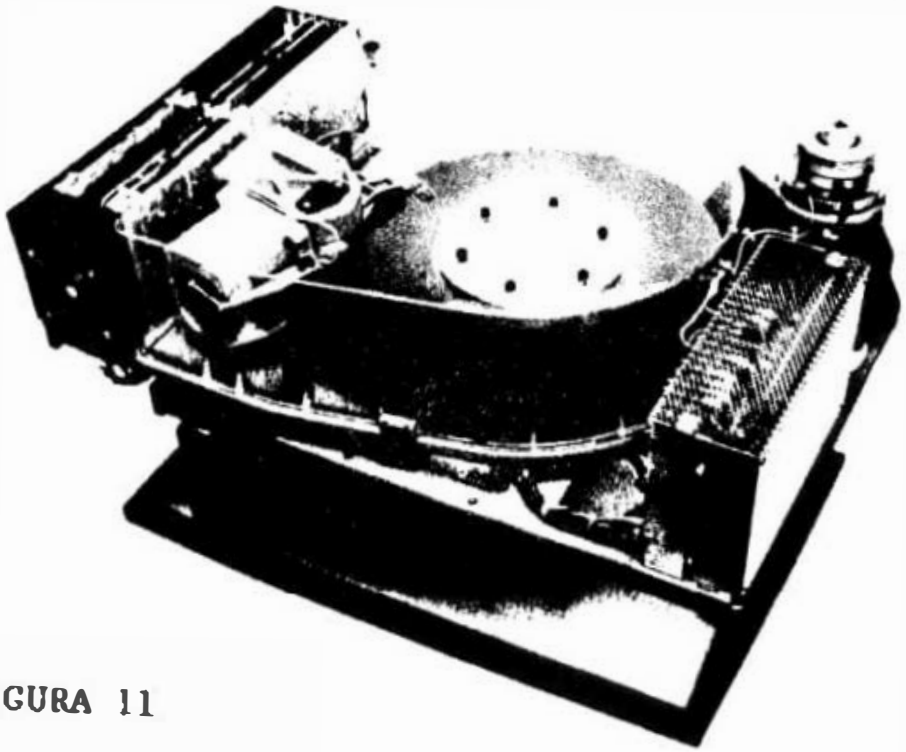


FIGURA 11

Unidade de disco magnético fixo

As memórias de fita cassete utilizam gravadores e fitas cassetes comuns. Este tipo de memória é muito utilizado em microcomputadores, devido, principalmente, ao seu baixo custo.

Ao utilizarmos fitas cassetes como dispositivo periférico de memória auxiliar, é óbvio, que devemos utilizar, também o gravador cassete, devidamente adaptado ao sistema.

Apesar de um tanto lentos, os gravadores cassetes, usados como memória de massa para pequenos sistemas, apresentam melhorias significantes sobre a fita de papel perfurada, anteriormente usada para estes fins. A figura 12 nos mostra um sistema de microcomputador que utiliza um gravador cassete como dispositivo de memória auxiliar.

FIGURA 12

Utilização de fita cassete como memória auxiliar.

Uma fita cassete comum de 60 minutos possui cerca de 171 metros de fita e pode armazenar aproximadamente 500000 bytes (500 K bytes) de dados em um só lado.

Existe uma série de sistemas de codificação para unidades de fitas cassetes. Entre eles podemos citar:

a - TK 85
b - CP 300
c - CP 500
d - TK 2000
e - MICROENGENHO 2
f - UNITRON

Cada um desses sistemas de codificação é um método projetado para usar a pista de uma gravação para gravar informações sob a forma de bits serial, utilizando gravadores cassetes comuns.

Um outro dispositivo periférico, utilizado como memória auxiliar, são os discos flexíveis (floppy disk).

Os discos flexíveis são, normalmente, protegidos por uma capa de papelão com uma abertura para permitir que um cabeçote magnético faça a escrita ou a leitura de informações. O disco gira dentro desta capa de papelão e um dispositivo de material plástico, devidamente alinhado e de baixo coeficiente de atrito, se encarrega de limpar o disco, enquanto este estiver girando.

Geralmente, os discos flexíveis são encontrados em dois tamanhos padrões: 8 polegadas (ou 203mm) e 5 1/4 polegadas (ou 133mm). A figura 13 nos mostra alguns detalhes de ambos os discos.

Os discos flexíveis foram introduzidos pela IBM, como um meio para a entrada de dados teclada, com vistas a substituir os cartões perfurados. Hoje, os sistemas de discos flexíveis são largamente usados como memória de massa, em sistemas de mini e microcomputadores e em terminais que fazem parte de grandes sistemas de processamento de dados.

Sendo os discos flexíveis, planos e finos, tornam-se convenientes para o transporte, o envio e o armazenamento.

Nos sistemas de discos flexíveis, durante uma operação de leitura/escrita, o cabeçote toca a superfície do disco. Em seguida, o cabeçote é retirado da superfície para evitar desgastes nos discos.

Os discos de 8 polegadas contêm 77 pistas divididas em 26 setores. Antes de cada gravação, os identificadores são escritos em um cabeçote, designando a pista e o setor. Este

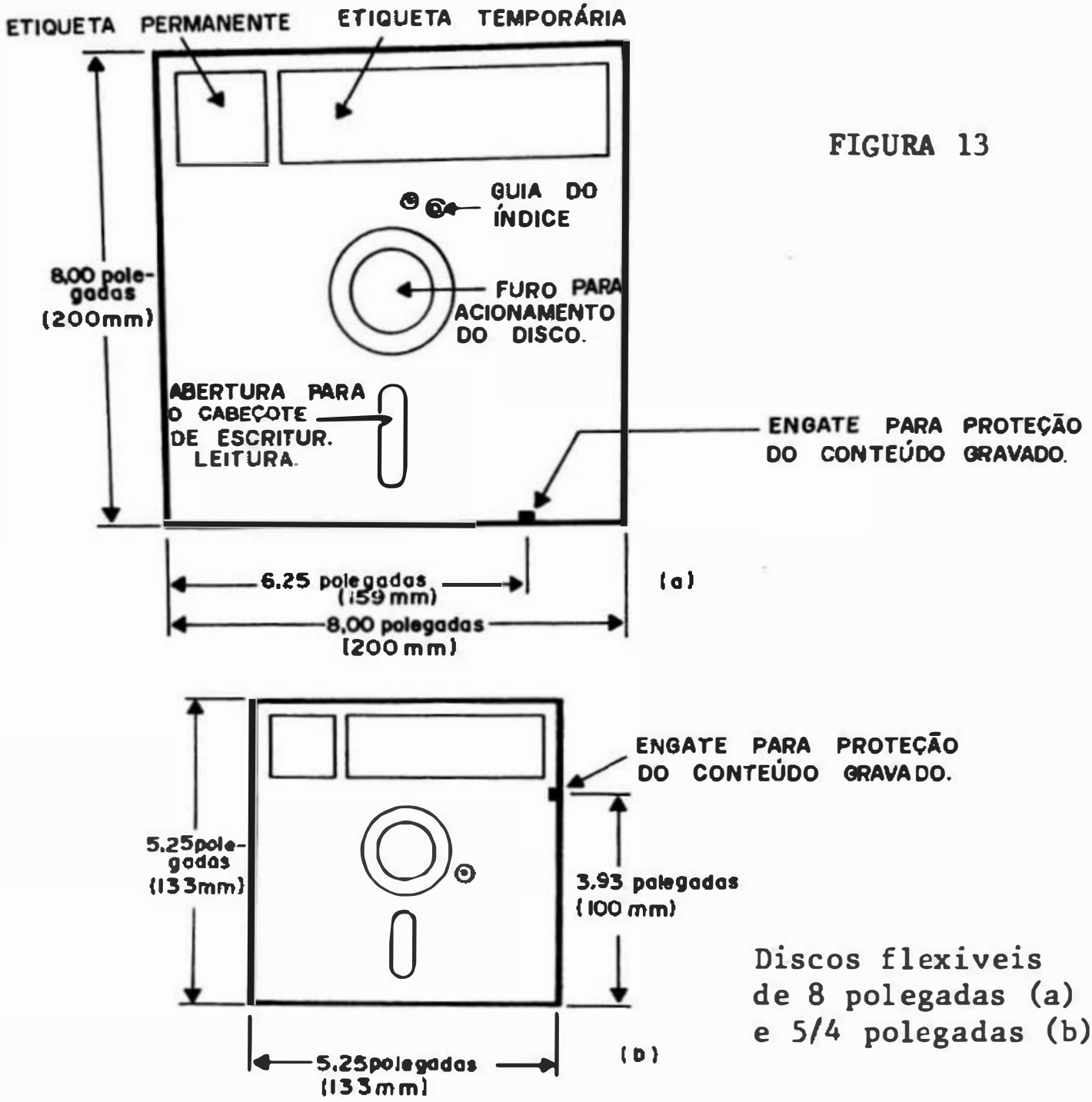


FIGURA 13

Discos flexiveis de 8 polegadas (a) e 5/4 polegadas (b)

formato básico utiliza um único lado do disco de densidade simples e é conhecido por SOFT-SECTORED ou 3740 da IBM. Densidade simples significa que a decodificação usada é do tipo FM. Na verdade, ao dizermos que o formato é SOFT-SECTORED, significa que a identificação do primeiro setor do disco flexível é feita por uma lacuna de um único índice perfurado no disco e lida por um sensor fotoelétrico. Cada um dos 26 setores subseqüentes será então identificado por um cabeçote de identificação de setor, precedido dos 128 bytes de dados que estão contidos em cada setor.

O formato SOFT-SECTORED padrão usa 74 das 77 pistas válidas para gravação de dados, o que implica em uma capacidade -

de 246.272 bytes.

Vários refinamentos do formato do disco de 8 polegadas - foram sendo introduzidos, aumentando a capacidade de cada disco. A introdução da gravação de dupla densidade vence o fator de limitação das mudanças do fluxo magnético,fixo por superfície de gravação, a partir da utilização de um método de codificação mais eficiente para representar o dado binário. Entre esses métodos de codificação, podemos citar o MFM e o M^2FM com os quais se elimina a maioria dos pulsos clock, usados na identificação do bit. Isto tornou possível dobrar a densidade de decodificação FM. Com a gravação de dupla densidade,256 bytes são gravados em cada um dos 26 setores, elevando a capacidade total de gravação de 246.272 bytes para nada menos que 492.544 bytes.

Um disco flexível do tipo HARD-SECTORED possui um "buraco", perfurado no disco, para cada setor do mesmo. Conforme esses buracos passam por uma luz, um detector foto sensível emite um pulso, identificando, assim, o início dos setores.Um disco HARD-SECTORED típico possui 32 setores, cada um identificado por uma lacuna.

Os discos flexíveis de 5 1/4 polegadas ou mini discos são, simplesmente, uma versão menor do disco flexível de 8 polegadas. A figura 14 nos mostra um drive para discos flexíveis de 5 1/4 polegadas.

FIGURA 14

Unidade de drive para discos flexiveis

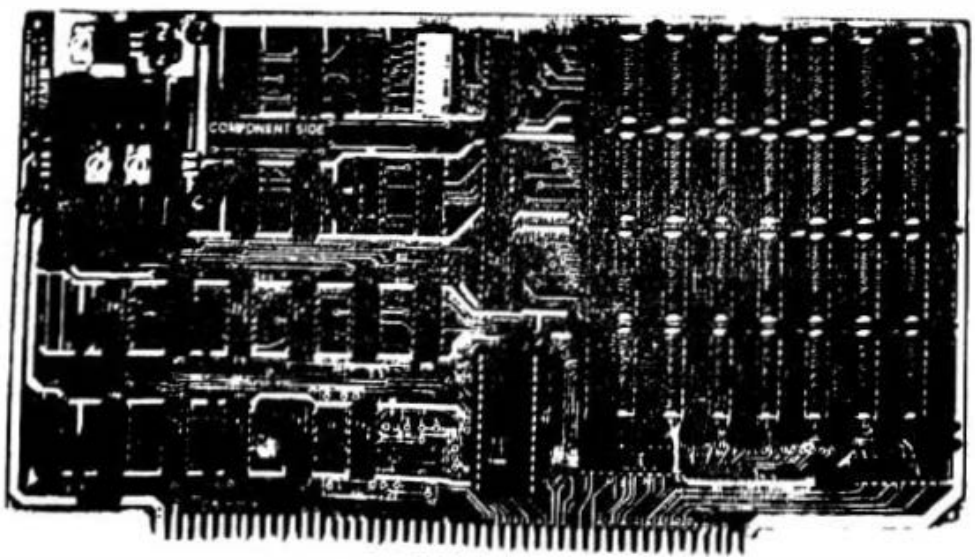
Um disco flexível de 5 1/4 de polegadas, de uma única densidade e um único lado, armazena cerca de 72K bytes de dados. Estes discos possuem de 10 a 16 setores de 40 pistas e giram com uma rotação média de aproximadamente 300 rpm. O tempo de acesso médio varia de 174 a 296 nanossegundos (ns). O principal fator que levou à aplicação em larga escala desses discos flexíveis foi o custo relativamente baixo da unidade de operação (cerca de 60% mais barato que uma unidade de operação de discos flexíveis de 8 polegadas).

Os dispositivos de memória de estado sólido (memórias de semicondutores), já estudadas em lições anteriores, são usadas para armazenarem dados e programas, de modo que o computador possa ter acesso relativamente rápido a essas informações armazenadas. Devido a esta consideração, as memórias de estado sólido são tidas como memórias de trabalho interno, distinguindo-se assim, das memórias de massa ou secundárias. As memórias de estado sólido também são consideradas como sendo memórias de circulação.

Em muitos casos, uma memória de estado sólido pode ser adicionada em um computador ou ainda, se estiver em uma placa de circuito impresso, esta pode ser conectada no sistema de barramento principal do computador, como sendo memória adicional. Nestas condições, podemos considerar a memória de estado sólido adicional, como sendo um dispositivo periférico.

A figura 15 nos mostra uma expansão de memória RAM, para microcomputador, de 64K byte. Essa expansão de memória ou memória adicional pode então ser considerada como um dispositi-

FIGURA 15



Expansão de memória RAM

vo periférico, conectado ao microcomputador.

Na próxima lição, dando continuidade ao assunto, estudaremos outros dispositivos periféricos.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-8

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Em um digitador de imagem, a introdução de dados é feita através:

a) do cursor;
b) dos raios luminosos;
c) do tubo de raios catódicos;
d) N.R.A.

2) Os digitadores de mesa também são conhecidos por:

a) Gran digitadores;
b) Mesas de dados;
c) Digitadores para fins especiais;
d) N.R.A.

3) Em um joystick, normalmente, se utiliza como transdutor de posição:

a) um capacitor de placas paralelas;
b) uma mesa de dados;
c) um resistor variável;
d) um cursor tipo RATO.

4) As bolas de curso são usadas, tipicamente, em:

a) displays de radares;
b) mesas de dados;
c) sistemas de reconhecimento da fala;
d) N.R.A.

5) Nos sitemas de reconhecimento da fala, as características acústicas são convertidas, de tensões analógicas, para:

a) sinais acústicos;
b) sinais visuais;
c) sinais digitais;
d) N.R.A.

6) Nos sistemas que utilizam leitores ópticos, um bastão gera um sinal elétrico:

a) proporcional à luz emitida;
b) inversamente proporcional à tensão aplicada;
c) diretamente proporcional à corrente de alimentação do bastão;
d) N.R.A.

7) Nas leitoras de cartões tipo escova, as escovas:

a) percebem se há "sujeira" sobre os cartões;
b) percebem eletricamente a presença ou ausência de perfurações em cada coluna do cartão;
c) determinam o movimento dos cartões;
d) N.R.A.

8) Como exemplo de periférico utilizado como fonte de telemedida no controle automático de processos, podemos citar:

a) as leitoras de cartão;
b) as leitoras de fita de papel;
c) os instrumentos analógicos de medida;
d) N.R.A.

9) Uma fita cassete comum de 60 minutos, utilizada como memória auxiliar, pode armazenar aproximadamente, em um único lado:

a) 5000K bytes;
b) 5000 bytes;
c) 50.000 bytes;
d) 500.000 bytes.

10) Um disco flexível de 8 polegadas contém:

a) 71 pistas divididas em 26 setores;
b) 71 pistas divididas em 36 setores;
c) 77 pistas divididas em 36 setores;
d) 77 pistas divididas em 26 setores.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-9

DISPOSITIVOS PERIFÉRICOS III

INTRODUÇÃO:

Estudaremos, nesta lição, alguns dos principais dispositivos periféricos de saída. Começaremos pelos displays que são utilizados nos painéis planos.

PAINÉIS PLANOS COM BASE EM DISPLAYS:

Os painéis feitos de displays são muito utilizados, quando as variáveis espaço e aspereza são altamente relevantes, substituindo os convencionais tubos de raios catódicos (TRC).

Três tecnologias são correntemente usadas para a obtenção de displays de painéis planos:

1 - Display de cristal líquido (LCD);
2 - Display emissor de luz (LED);
3 - Display de "plasma" (ou de descarga a gás).

Existem outras tecnologias, como por exemplo, a tecnologia do "laser pulsado programável" (utilizada para escrever grandes imagens projetáveis) que, por não serem freqüentemente utilizadas em dispositivos periféricos, não iremos estudá-las.

Muito embora algumas das tecnologias enumeradas anteriormente já tenham sido objeto de nosso estudo em lições ante-

riores (veja lição ED-21), iremos, ainda, tecer alguns comentários sobre as mesmas.

Como sabemos, os LCDs não emitem luz, muito pelo contrário, necessitam de uma fonte de luz externa para que a informação possa se tornar visível. O material de cristal líquido é compactado entre duas placas de vidro; o lado interno dessas placas é revestido por um material condutivo sob a forma de barras de um número segmentado (ou algum outro símbolo). Veja a figura 1.

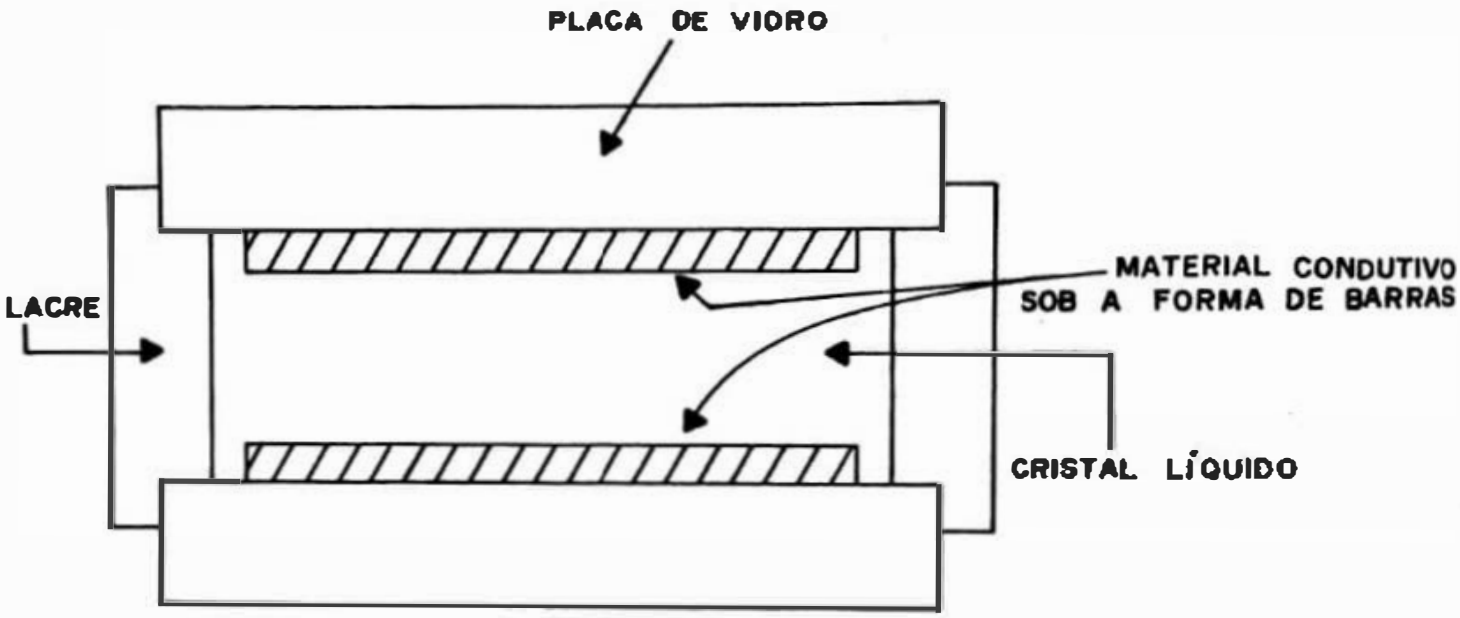


FIGURA 01

Ao se aplicar uma tensão sobre os segmentos condutores (barras) selecionados, estes se tornam escuros, devido às propriedades eletro-ópticas do material de cristal líquido, fazendo, assim, com que o número ou símbolo se torne visível.

Esses displays são muito utilizados como displays numéricos de sete segmentos (veja figura 2), para calculadoras de bolso e relógios digitais.

Os LEDs, ao contrário dos LCDs, emitem luz quando uma tensão lhe é aplicada, o que os torna altamente visíveis no escuro. Este tipo de display, na configuração de sete segmentos (veja figura 2), também é muito utilizado para a exibição numérica em calculadoras eletrônicas e relógios digitais.

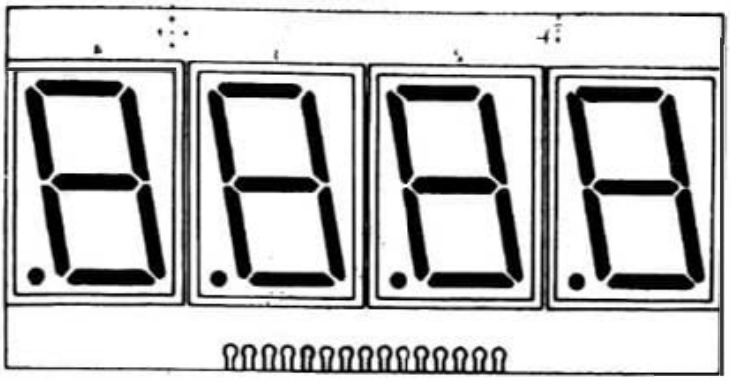
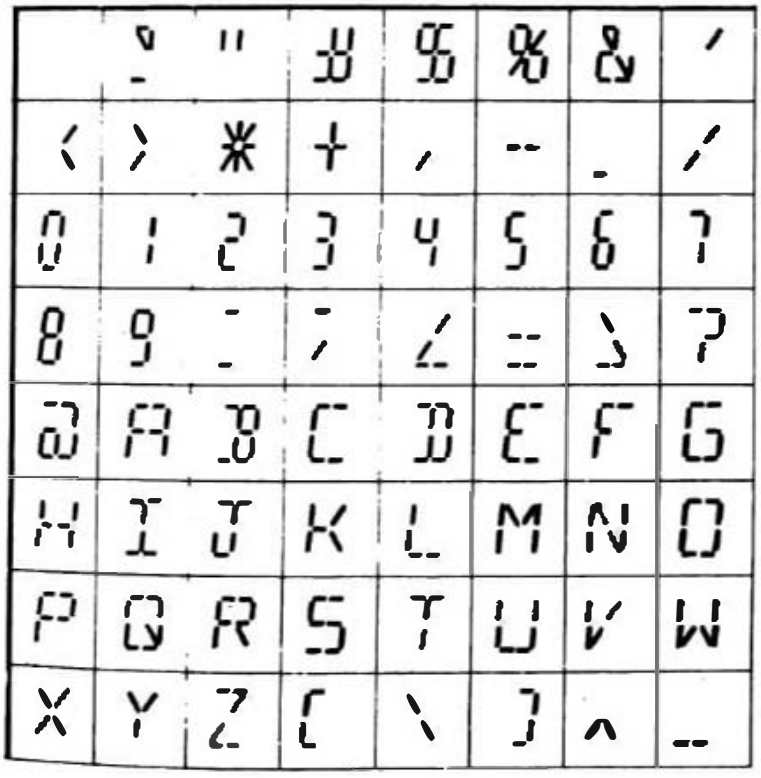
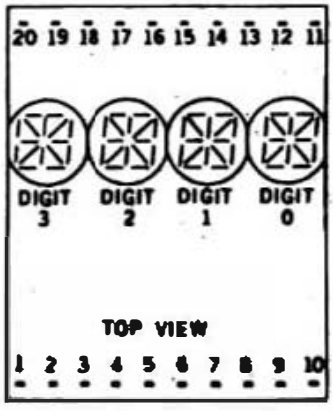


FIGURA 02

Os displays de LEDs também estão sendo muito utilizados em microcomputadores, sob a forma de displays alfanuméricos, em uma configuração de 16 segmentos (veja figura 3).

FIGURA 03



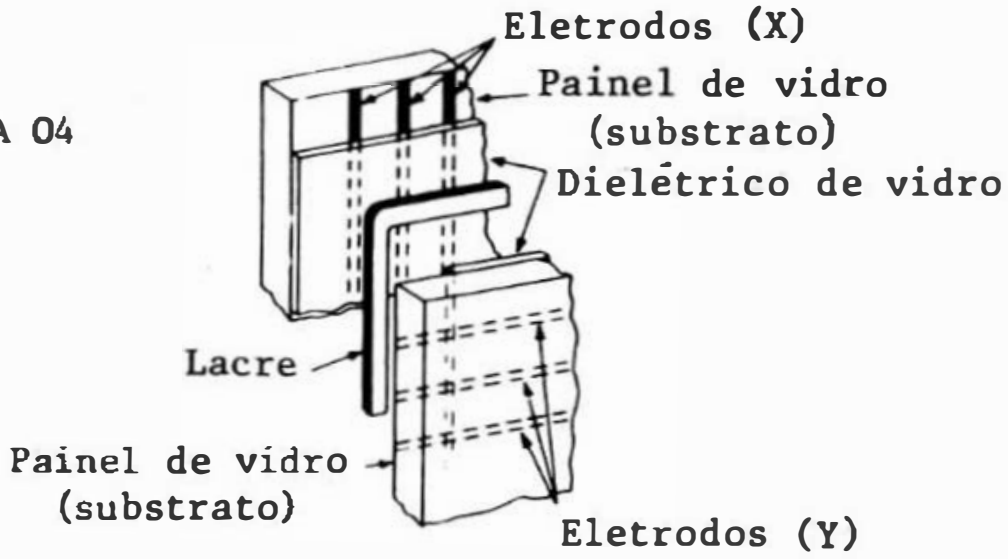


Dispostos como uma matriz, os LEDs também são capazes de exibir gráficos. Isto se tornou possível através da miniaturização dos LEDs, permitindo assim uma densidade da ordem de 50 LEDs por polegada (ou 2 LEDs por milímetro). Os LEDs são montados sobre um substrato e endereçados por uma matriz XY de circuito impresso.

Uma outra tecnologia que tem sido utilizada ultimamente na confecção dos painéis planos é a dos displays de plasma.

Esses displays são constituídos de um gás (normalmente uma mistura de neon e argônio), fechado entre dois painéis de vidro (substratos). Embutidos nesses painéis de vidro ou substratos, tem-se uma colocação paralela de eletrodos. Esses eletrodos formam um ângulo reto (90º) com relação aos eletrodos do painel de vidro oposto, constituindo assim uma matriz de fios XY (veja figura 4).

FIGURA 04



Construção do painel de plasma.

A figura 5 nos mostra um painel plano, constituído de displays de plasma, composto por uma matriz de fios XY de 512 linhas (X) e 512 colunas (Y).

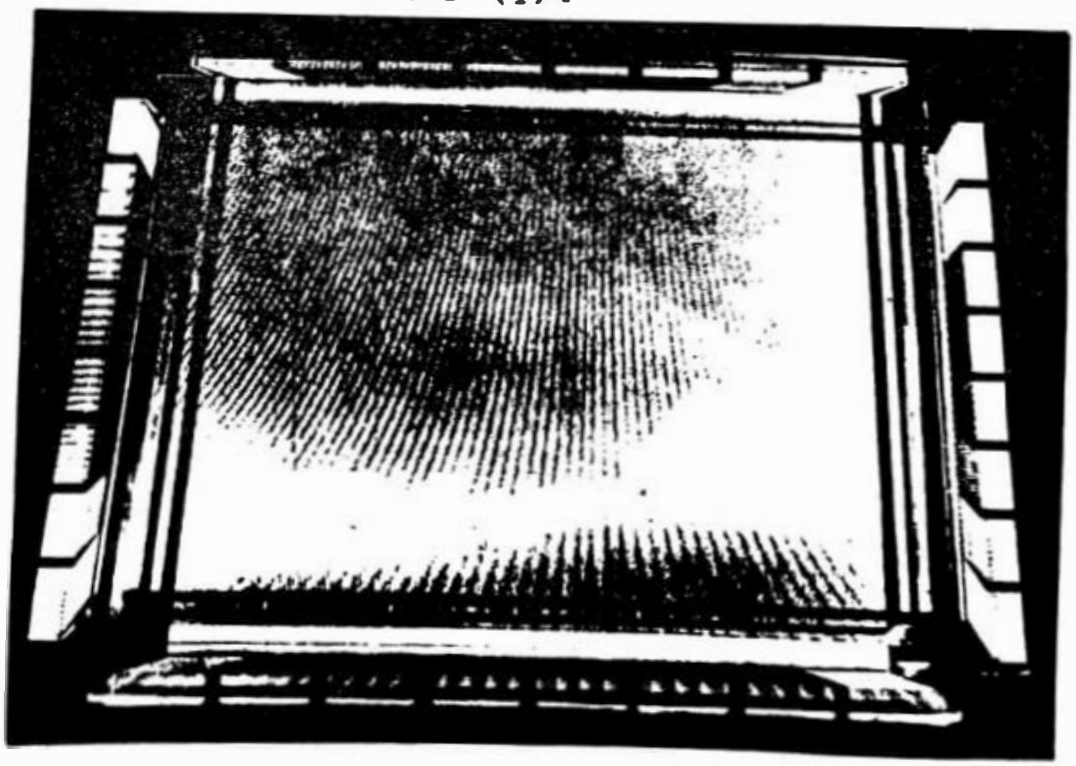


FIGURA 05

Quando, em algumas das interseções, os fios X e Y são energizados, a tensão desenvolvida entre os dois eletrodos excede o valor da tensão de ionização do gás que está entre o par de fios X e Y. Desta forma, o gás passa para um estado de plasma do tipo ion-elétron e, conseqüentemente, passa a emitir um pulso de luz de alta freqüência, fazendo parecer, ao olho humano, um ponto continuadamente iluminado.

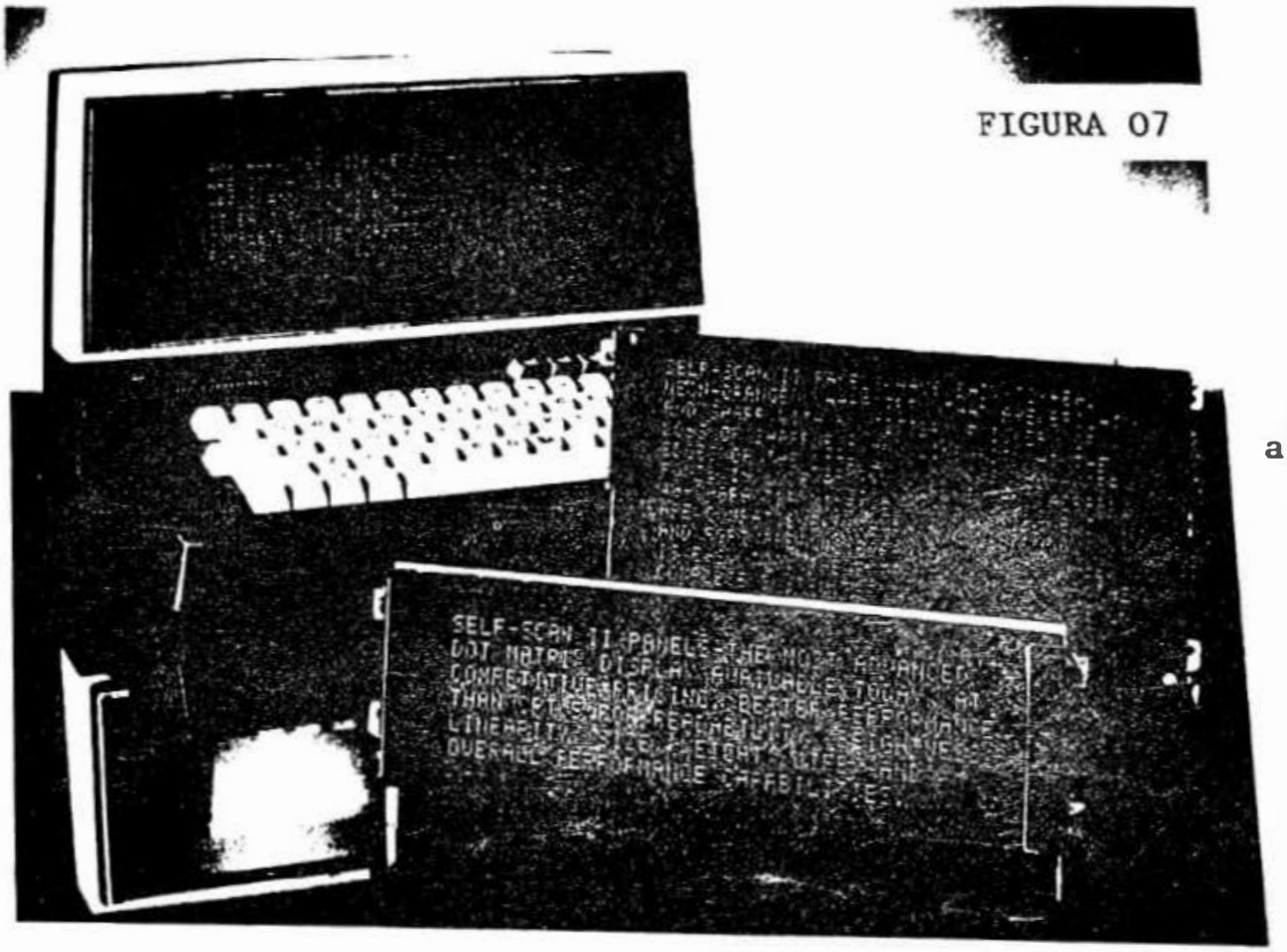
O espaçamento entre os fios que formam a matriz XY permite uma resolução da ordem de 60 pontos iluminados por polegadas (ou 2,36 pontos por milímetro). Com isto, podem-se representar, por exemplo, gráficos com uma boa precisão. A mistura dos gases neon e argônio imprime a esse display uma cor laranja-avermelhada. Como exemplo de aplicação, a figura 6 nos mostra um típico painel de display de plasma (projetado para competir com os tubos de raios catódicos), cuja área de visão do painel é um quadrado de cerca de 8,5 polegadas (216mm) de lado, formado por uma matriz de 512 x 512 eletrodos e com uma espessura de aproximadamente 1/2 polegada.

FIGURA 06

Estes tipos de displays são muito utilizados em condições rústicas onde os tubos de raios catódicos fatalmente seriam destruídos, geralmente em aplicações militares. Comercialmente já se produz displays de plasma para painéis de texto (figura 7-a) e displays de plasma de linha simples, com caracteres de vários tamanhos (figura 7-b). Estes tipos de dis-

plays são freqüentemente usados em mini e microcomputadores, tanto para exibir textos como para "advertir" o operador.

FIGURA 07



a)

b)

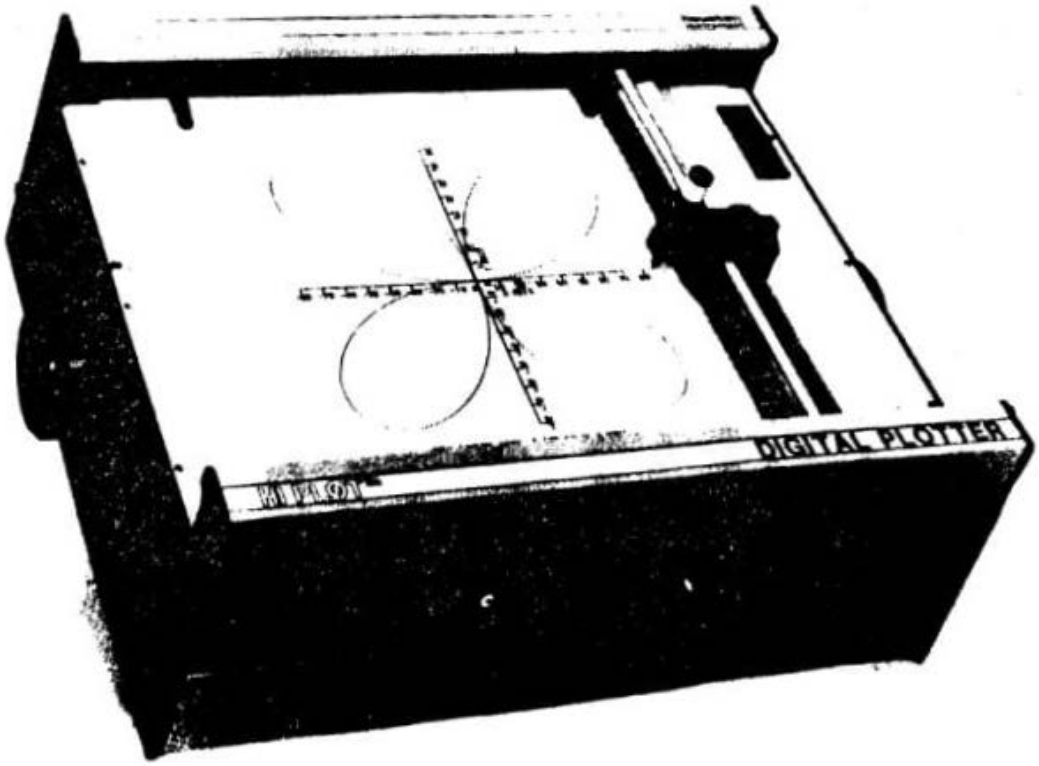
A tecnologia do display de plasma também permite a representação de valores analógicos, sob a forma de gráficos de barras lineares e/ou circulares.

TRAÇADORES DE GRÁFICOS

Mais conhecido como "plotter" (derivado do inglês), é um periférico de saída, que imprime linhas e curvas as quais, - quando devidamente combinadas, formam desenhos, gráficos, diagramas e todas as formas de "retratos" gráficos.

Os dados utilizados para traçar um gráfico são armazenados na memória do computador. Estes dados são então requisitados e juntamente com um programa de "traçamento", irão traçar o gráfico. Os comandos do programa de traçamento são transformados deste programa para um sinal diferente, através de um controlador de traçamento, o qual transforma esses sinais em movimentos de um conjunto de motores operadores de canetas. A figura 8 nos mostra um traçador de grafico do tipo mesa com caneta XY, muito utilizado como periférico de mini e microcomputadores.

FIGURA 08



Existe, no mercado, uma variedade relativamente grande de traçadores gráficos, os quais podem ser classificados de quatro maneiras:

1 - Pelos métodos de marcação e varredura;
2 - Com ou sem movimento do papel;
3 - Pelo tamanho do traçador;
4 - Quanto à aplicação, ou seja, se o sistema gráfico é dedicado a um tipo particular de uso.

Com relação aos métodos de marcação, o mais utilizado é o da caneta, que pode ser tanto do tipo de nylon ou do tipo de feltro. Os traçadores que utilizam esse método são conhecidos como "traçadores de caneta".

Os traçadores de caneta incorporam um ou mais motores de "escalonamento" (ou STEPPING), os quais operam as canetas em pequenos passos de linha reta, (sob o comando do programa traçador) tipicamente, de 0,25 a 0,032 milímetros (0,01 a 0,00125 polegadas). A combinação dos pequenos passos X e Y permite ao traçador desenhar curvas com uma resolução tão boa quanto a de uma linha reta (traçada em alguma direção). Devido ao movimento dos traçadores de canetas em pequenos passos, estes, também passaram a ser conhecidos como "traçadores incrementais". Já, devido ao fato deste tipo de traçador poder traçar uma linha ou curva, continuadamente, em duas direções (X e Y) eles também passaram a receber o nome de traçadores XY ou traçadores de vetor.

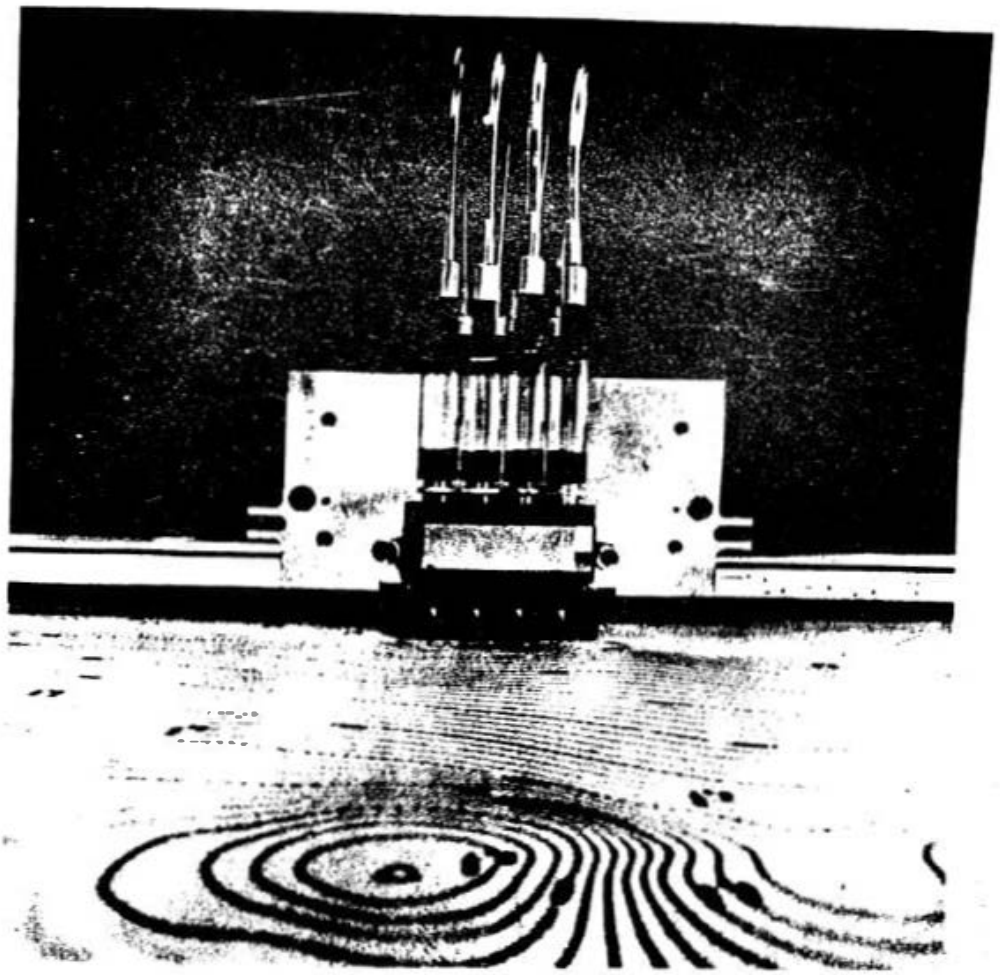
Um traçador de vetor de uma só caneta traça uma cor por vez. Se for necessário alguma outra cor adicional, isto deve ser previsto no programa, de tal forma a dividi-lo em seções, por cor. Ao final de cada seção o sistema pára e, sob a ação de um operador, troca-se a caneta por uma com a outra cor desejada. Em seguida, recomeça-se o traçamento.

Já, nos traçadores de múltiplas canetas, tal fato não ocorre, pois estes traçam múltiplas cores sem a necessidade de se interromper o programa. A figura 9 nos mostra um traçador gráfico de múltiplas canetas.

Além da caneta e do papel que são "ferramentas" costumeiras de desenho, outras podem ser utilizadas. Por exemplo, como superfície de desenho podem-se usar: plástico (MILAR), vidro, transparências com proteção por cima, filme fotográfico, placa de metal, etc. Já, como instrumentos de desenho po

demos citar: cabeçotes fototraçadores de ferramentas para gravações termocrônicas.

FIGURA 09



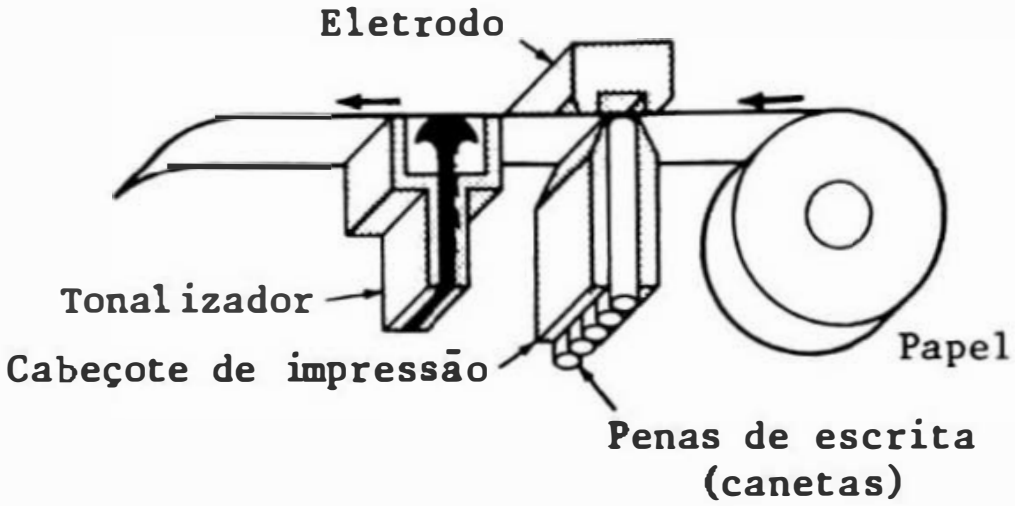
Os traçadores eletrostáticos (ou Raster-Scan) processam primeiramente a imagem, dividindo-a em seguida em linhas. Neste periférico, um cabeçote de impressão eletrostático traça uma linha de pontos por vez. Esses traçadores geralmente marcam os pontos individuais de uma linha, com uma densidade de 100 a 200 pontos por polegada, o que os habilita a escrever (traçar) textos com a mesma facilidade com que podem traçar um gráfico. Devido a este fato, os traçadores eletrostáticos são freqüentemente referidos como "traçadores - impressores". A figura 10 nos mostra o mecanismo básico de um traçador eletrostático.

Como já vimos, o fato do papel se mover ou não se mover também é um critério freqüentemente adotado para a classificação dos traçadores gráficos.

Os traçadores gráficos, sem movimento de papel, são traçadores do tipo FLAT-BED. Esses traçadores são como uma mesa de desenho comum (prancheta plana). Sobre sua superfície pla

na coloca-se o papel ou outro material de desenho, enquanto que a ferramenta de desenho se move sobre a superfície em duas direções X e Y. Ambas as direções de desenho são fixadas pela escala de movimentos da caneta e pelo tamanho da superfície plana.

FIGURA 10



Os traçadores com movimento de papel são traçadores de tambor (às vezes referidos como traçadores de correia) ou traçadores RASTER. Nesses traçadores gráficos, a caneta move-se somente em uma direção (direção X) sobre a direção do movimento do papel (direção Y). A combinação dos movimentos da caneta e do papel produzem as linhas e curvas desejadas, formando uma imagem completa. Tipicamente o papel move-se para frente e para trás conforme seja necessário para desenhar uma curva ou uma linha. A figura 11 nos mostra um traçador gráfico de movimento de papel através do empilhamento de papel dobrado.

Os traçadores eletrostáticos RASTER-SCAN são também traçadores gráficos de movimento de papel. Diferem dos traçadores de tambor, nos quais o movimento de papel é contínuo em uma direção. Nos traçadores RASTER-SCAN nenhum movimento do papel é utilizado para criar uma linha ou curva, como é o caso dos traçadores de tambor e caneta.

Para todos os traçadores gráficos com movimento de papel, a largura do "traço" (desenho) é limitada pela largura do tambor ou do cabeçote de impressão. Esta largura, normalmente, está entre 11 e 36 polegadas (279 e 914 mm). Por outro lado, o cumprimento da superfície de traçamento é virtualmente ilimi-

tado.

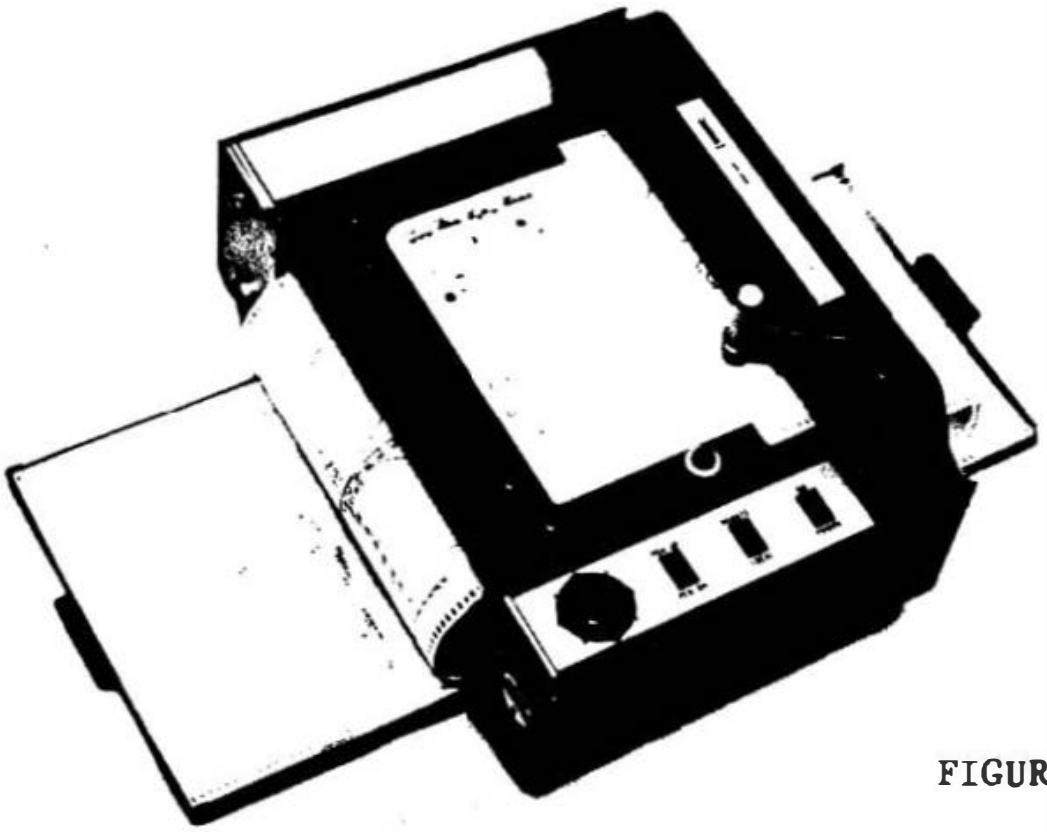


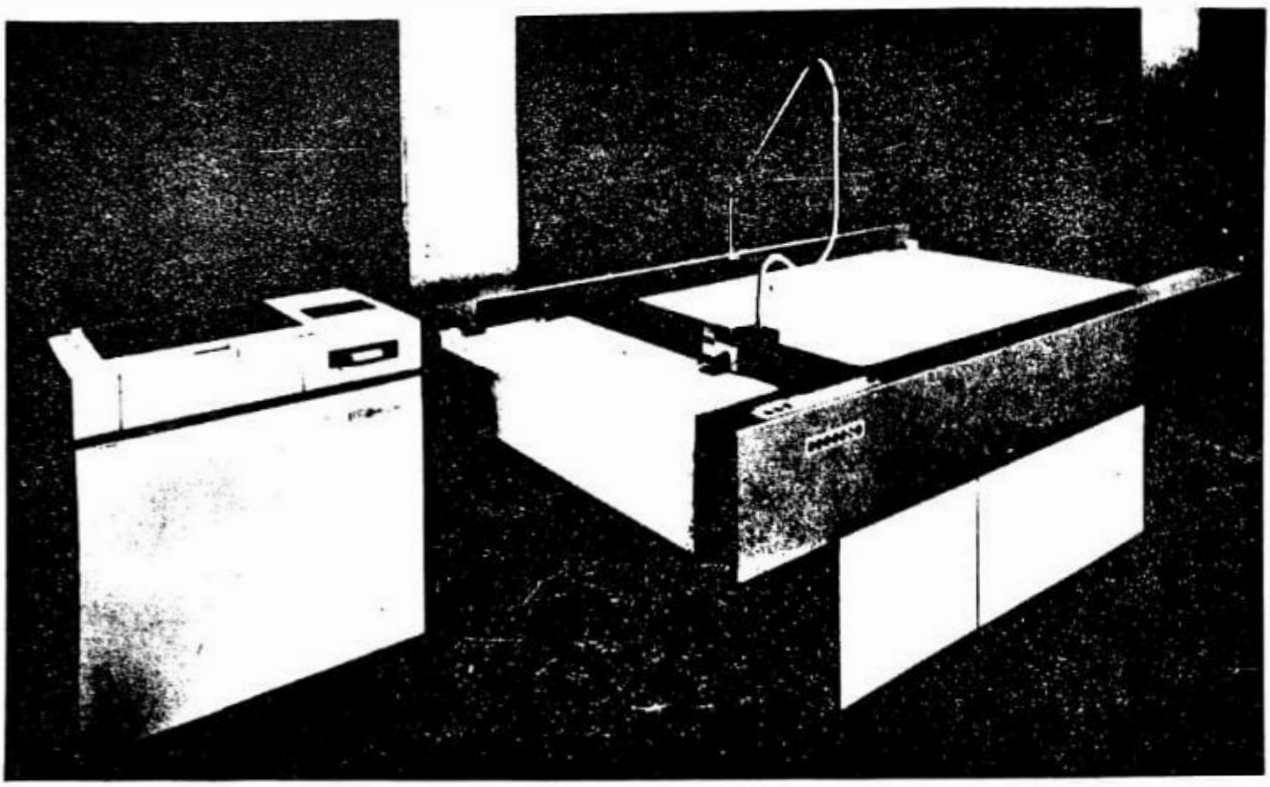
FIGURA 11

Um outro tipo de traçadores gráficos são os traçadores - de mesa, utilizados tanto para a reprodução de documentos do tamanho de uma página e pequenos desenhos como para grandes desenhos. Já os traçadores de fixação livre são empregados na reprodução de desenhos científicos e de engenharia, como mapas, por exemplo. Tanto os traçadores FLAT-BED como os de tambor podem ser classificados como traçadores de mesa. A figura 12 nos mostra um traçador de mesa (tipo FLAT-BED), com uma superfície de desenho de 50 x 82 polegadas (ou 1270 x 2083mm).

Os traçadores de superfície são largamente usados em sistemas de desenhos automáticos. Estes sistemas fazem desenhos de engenharia, "mapas" de terrenos, diagramas de circuitos e arte das placas de circuito impresso, entre outros. Esses traçadores são, geralmente, usados em conjunção com os digitadores de imagem. Como exemplo, podemos citar o caso de alguns digitadores de estereoscópia que, juntamente com um sistema de desenho automático, traçam contornos de matas diretamente de fotografias aéreas estereográficas. Os traçadores gráficos, tidos como sistemas "dedicados", diferenciam-se dos traçadores gráficos, tidos como periféricos, pelo fato de não serem Periféricos a uma UCP (Unidade Central de Processamento), mui

to pelo contrário, o sistema de desenho automático (sistemas dedicados) é o ponto focal de um sistema com fins especiais.

FIGURA 12



Como exemplo de sistema dedicado, podemos citar um sistema especial, formado por um traçador gráfico, um monitor de vídeo e um microcomputador, destinado a realizar o lay-out de placas de circuito impresso.

SAÍDA DE ÁUDIO:

Os periféricos de saída do computador possuem a função comum de fornecer informação em uma forma que seja útil para as pessoas. A maioria das saídas é feita através de textos legíveis e gráficos visuais, porém, atualmente a forma de saída audível está se desenvolvendo muito. Dentre os dispositivos de saída de áudio o que mais nos chama a atenção é o processo de sintetização da voz humana.

SINTETIZADORES DE FALA:

Os sintetizadores de fala de um computador compõem a fala através de fonemas (que são elementos da fala), tornando-a reconhecível. Um sistema típico utiliza 63 comandos de fonemas e 4 comandos de ênfase. Com as combinações possíveis dos

comandos de fonemas com os de ênfase, um programador, ao reunir os fonemas em um programa de fala, consegue formar a palavra desejada.

Os terminais de fala e as calculadoras são correntemente usados para assistir as pessoas deficientes fisicamente, bem como para auxiliar na educação de crianças com deficiência oral e visual.

De uma forma bem simplificada, podemos relacionar os componentes de um sintetizador de fala, como sendo:

1 - Uma analogia elétrica do trajeto vocal humano;
2 - Um programa para verificar o som desejado nos parâmetros de controle analógicos do trajeto vocal;
3 - Interface de controle do trajeto vocal.

Os métodos de sintetização da fala humana pelo computador diferem levemente de um fabricante para outro. Apenas a título de informação, descreveremos um dos métodos de sintetização da fala humana. Vejamos então, como um circuito eletrônico é capaz de gerar vogais, consoantes, etc.:

Vogais: Um sintetizador de fala eletrônico é uma analogia direta do trajeto vocal humano. O trajeto vocal humano forma um tubo de ressonagem acústica, normalmente de 173 mm de comprimento. Quando o ar passa a vibrar em um ressonador acústico com estas dimensões, a pressão do ar forma ondas com freqüências de 500, 1.500 e 2.500 hertz. Estas três freqüências ressonantes são chamadas de "freqüências formantes" (simbolicamente representadas por F1, F2 e F3, respectivamente) as quais são, atualmente, as freqüências do centro das faixas de freqüências ressonantes que determinam a qualidade fonética de uma vogal. As freqüências ao redor de cada uma destas três freqüências centrais são controladas no trajeto vocal humano pelo controle muscular da garganta e da boca. Como primeira aproximação de um sintetizador de fala eletrônico, estas freqüências formantes são produzidas em quatro amplificadores, com filtro de faixa de passagem digitalmente controlado. Cada amplificador é alimentado por uma freqüência de voz original, de amplitude controlada que corresponde à caixa de voz na garganta. A figura 13 nos mostra um diagrama de blocos de um circuito que produz ondas com freqüências iguais às freqüências formantes.

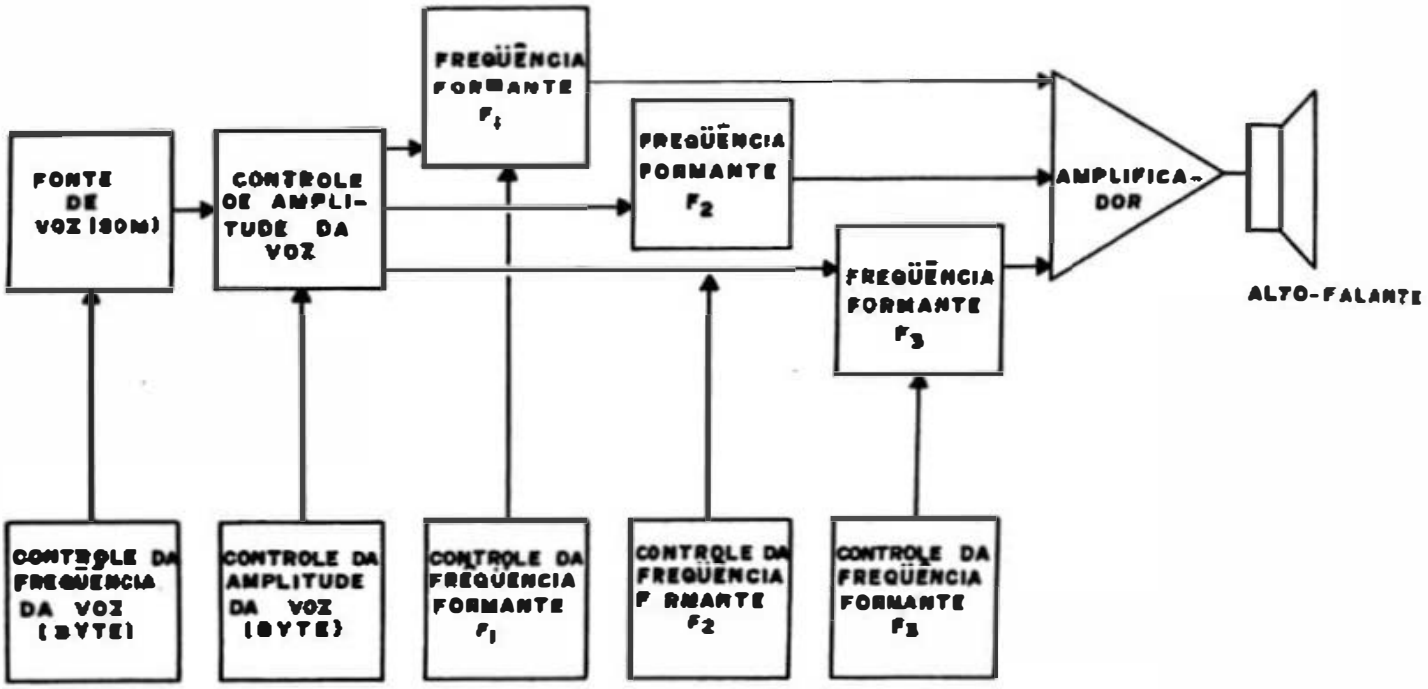


FIGURA 13

Esta parte de um sintetizador de freqüência produz as vogais A, E, I, O, U, dependendo das freqüências especificadas por F1, F2 e F3. Cada um desses amplificadores de passagem é controlado por um byte de oito bits, para permitir a freqüência formante dentro da faixa a ser ajustada, exatamente como os músculos controlam a forma do trajeto vocal.

O sintetizador de fala lê os bytes de controle F1, F2 e F3 do barramento de dados a cada 10 ou 20 milissegundos, dependendo do modelo do sintetizador. Isto constitui a velocidade da estrutura da fala, ou seja, o limite da fala para o qual o sintetizador é levado a mudar os parâmetros da mesma. Quando se tratar de vogais de estado fixo, não há necessidade de se mudarem os parâmetros de controle. Porém, em se tratando de palavras que contêm mais de uma vogal, em algumas delas existe uma transição suave de uma vogal para a outra, quando a palavra é pronunciada (a esse tipo de palavra dá-se o nome de ditongo), havendo aqui a necessidade de se controlarem os parâmetros da fala. A figura 14 nos mostra os pontos iniciais e finais da freqüência de várias palavras, em inglês, com ditongos.

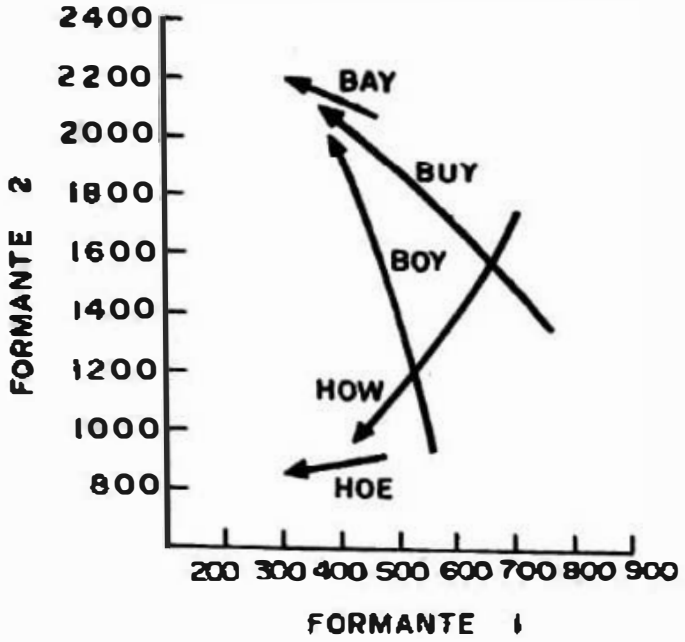


FIGURA 14

O tempo gasto, em média, para se pronunciar um ditongo está entre 150 e 250 milissegundos. Em uma velocidade de estrutura de 10 milissegundos, podemos concluir que as freqüências formantes irão mudar de 15 a 25 vezes sobre o tempo de duração da palavra pronunciada. Então, neste caso, há necessidade de se mudarem os bytes de controle na memória, para que possa ser pronunciada a palavra. Concluindo, podemos dizer que para se produzir um ditongo com um sintetizador de fala, é necessário especificar não somente as freqüências formantes da vogal, mas também o sincronismo destas mudanças sobre o tempo de duração da transição da vogal.

Consoantes: Para que se possa, através de um circuito eletrônico, pronunciar as consoantes foi necessário adicionar três novos elementos ao circuito eletrônico, utilizado anteriormente para pronunciar as vogais. Foram adicionados os seguintes elementos: uma fonte de ruído de aspiração, um ressonador nasal e um ressonador fricativo. Vejamos então em que consistem esses elementos.

A fonte de ruído de aspiração produz o pulso de ruído iniciante, de ar, para as consoantes que se iniciam com uma exclamação. Por exemplo, a consoante K inicia-se por abrir rapidamente o fundo da cavidade oral e então, logo em seguida, soltar uma explosão de ar (KA), seguindo-se uma transição para a vogal "a" (KAY). Este ruído de aspiração iniciante (KA), em um sintetizador de fala típico, é formado pelo ruído amplificado do rompimento de um diodo zener. A amplitude do ruído da aspiração é controlada por um byte, que controla também oito (8) resistores (um conversor D/A). O ruído da aspiração da própria amplitude é combinado com a freqüência da voz

em um amplificador MIXER, na entrada dos três ressonadores formantes. A figura 15 nos mostra o diagrama de blocos de um sintetizador de fala humana eletrônico, de pronúncia perfeitamente análoga à humana.

A fonte de ruído de aspiração é utilizada adicionalmente para produzir as consoantes fricativas, SH, ZH, S, Z, F, V, TH - todas elas caracterizadas por uma freqüência entre 2.500 e 8.000 hertz (Hz).

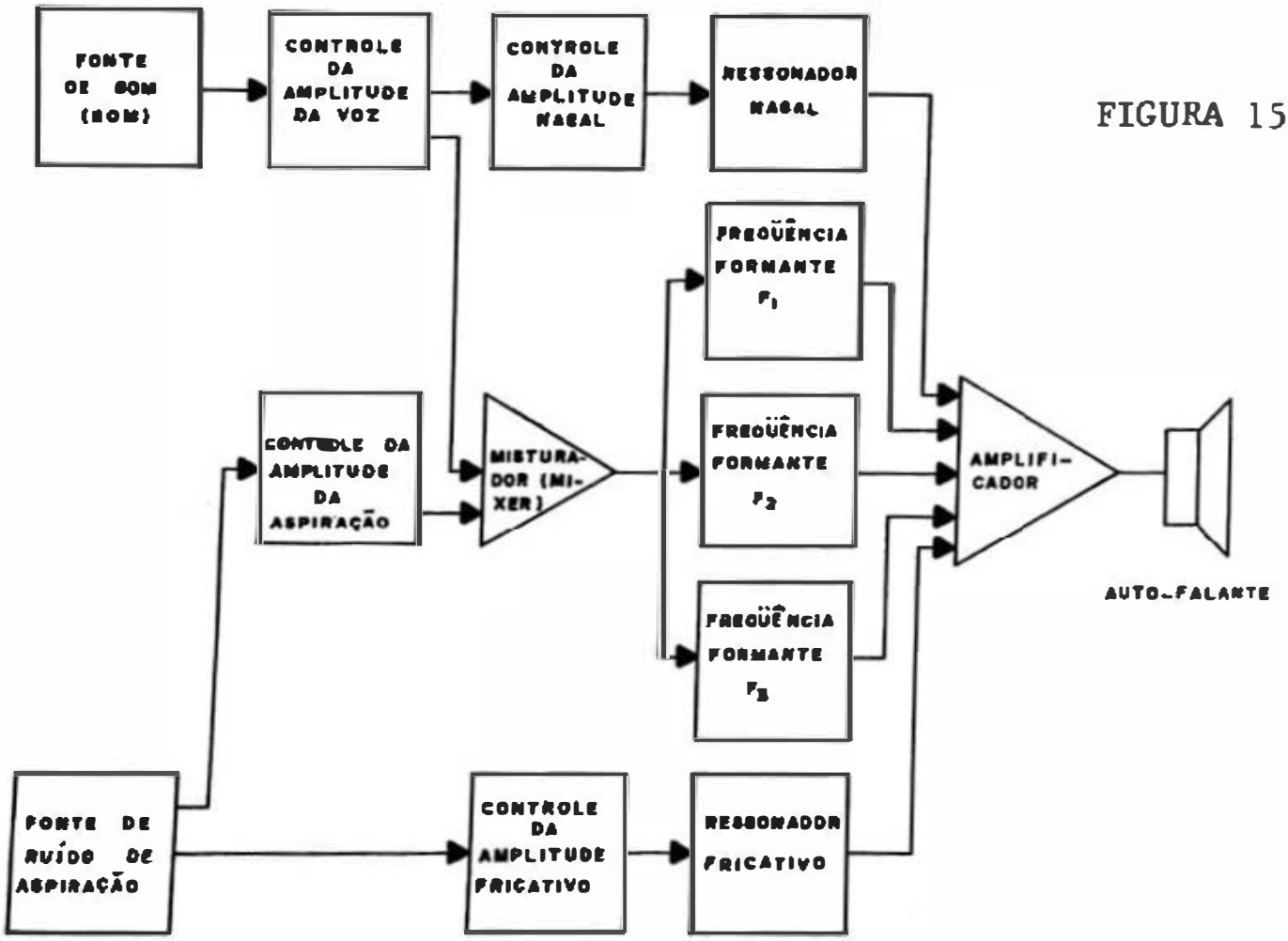


FIGURA 15

As consoantes fricativas são produzidas adicionando-se uma certa freqüência ao som original, vindo da fonte de ruído de aspiração. A operação do ressonador fricativo de amplitude controlada é justamente essa, ou seja, adicionar uma certa freqüência dentro do espectro de freqüência que vai de 2.500 Hz a 8.000 Hz. Dependendo de qual consoante é falada, o ressonador fricativo adiciona a componente de freqüência fricativa adequada.

As consoantes nasais M, N e NG são formadas com a ajuda de um ressonador de freqüência nasal, de cerca de 1.400 Hz, controlado por bytes e pela regulação dos resistores para o controle de amplitude nasal. Os componentes eletrônicos, capa-

zes de executar a combinação destas funções, são montados em placas de circuito impresso, compatível com o sistema de barramento de vários computadores. A adição de um alto-falante completa a analogia eletrônica do trajeto vocal humano.

Vejamos agora alguns parâmetros de controle do circuito.

Parâmetros de controle do circuito: Corresponde ao controle muscular do trajeto vocal; move parâmetros de controles diferentes com a finalidade de produzir os sons das consoantes e das vogais do sintetizador de voz.

Existem dois métodos de se estabilizar a linha de parâmetros para gerar palavras: "especificação do parâmetro pela análise do espectro de freqüência e potência das palavras faladas" e "especificação do parâmetro pela regra".

O primeiro método permite uma fala natural quase que perfeita, porém o esforço requerido é muito grande. Os parâmetros de controle do sintetizador de fala são desenvolvidos para analisar as palavras faladas, analisando os componentes de freqüência e amplitudes da palavra como uma função do tempo (ft). Este método é capaz de reproduzir a fala, independentemente da linguagem ou do dialeto.

Já o segundo método desenvolve os parâmetros de controle da fala de letras, representando os fonemas da língua em questão. As palavras são, então, decodificadas em fonemas e um programa ASSEMBLER aplica uma colocação de regras em função dos fonemas especificados e a sua relação com os outros para, em seguida, gerar os bytes de controle, utilizados para o controle dos parâmetros do sintetizador, tendo como resultado final a palavra pronunciada através do sintetizador de fala.

Com as informações prestadas sobre o assunto "sintetizador", esperamos que o aluno tenha condições de formar, a partir de agora, uma idéia mais clara a respeito deste dispositivo periférico de saída, o qual acreditamos que será muito utilizado em um futuro bem próximo.

OUTROS DISPOSITIVOS PERIFÉRICOS DE SAÍDA:

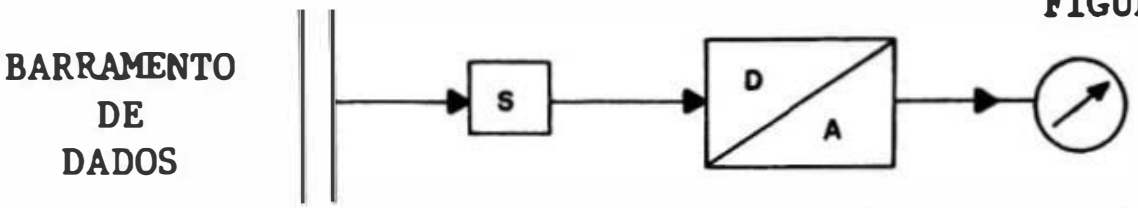
Procuraremos apresentar, de uma forma bem resumida, outros tipos de dispositivos periféricos de saída, mostrando suas principais características e aplicações. Então vejamos:

a) Quadro sinóptico: Trata-se de um periférico de saída, o qual fornece, por meio de um esquema sinóptico, a informa-

ção relativa a um controle automático de processos. Sua maior aplicação se dá no controle de fábricas automatizadas, de redes automatizadas de abastecimento de água, de redes elétricas automatizadas, de redes ferroviárias automatizadas, de rodovias automatizadas, etc.

b) Instrumento indicador: Este periférico de saída caracteriza-se por nos permitir visualizar, de forma analógica, resultados de processamento de medidas. Junto ao instrumento indicador é necessário acoplarmos um conversor D/A (veja figura 16). Este tipo de periférico de saída é muito utilizado no controle automático de processos.

FIGURA 16



c) Sinalizador luminoso: Trata-se de um dos dispositivos de saída mais simples, podendo ser construído a partir de um LED, de uma lâmpada, etc. Destina-se a fornecer certas informações acerca do estado interno do microcomputador ou do estado de processos automatizados. É um dispositivo de aplicação generalizada.

d) Executor digital: Este tipo de dispositivo periférico de saída destina-se a fornecer os resultados de processamento sob a forma da presença ou da ausência de uma tensão elétrica de comando. É muito utilizado no controle automático de processos para ligar e desligar motores e disjuntores, para ligar e desligar aparelhagem e instrumentação diversa, etc.

e) Alarme sonoro: Este tipo de periférico de saída, muito simples por sinal, destina-se a fornecer a indicação de uma ocorrência grave que normalmente não pode ser solucionada pelo microcomputador. Sua aplicação se dá principalmente - no controle automático de processos para, por exemplo, sinalizar ocorrências graves, as quais exigem a intervenção humana imediata, tal como a inoperância de uma parte essencial do sistema controlado.

Comentários finais: Existem outros tipos de dispositivos periféricos de saídas, uns mais complexos outros menos complexos, porém nos preocupamos em estudar os mais utilizados no momento. Na verdade, suprimimos desta lição, dois importantes dispositivos periféricos de saída, o "tubo de raios catódicos" (TRC) e a impressora. Estes periféricos são os mais utilizados em sistemas de microcomputadores, por isso, desenvolveremos uma lição especial sobre cada um deles.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-9

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) As tecnologias mais utilizadas para a obtenção de displays de painéis planos são:

a) display de cristal líquido (LCD);
b) display emissor de luz (LED);
c) display de plasma;
d) os itens a, b e c estão corretos.

2) A tecnologia do laser pulsado programável é muito utilizada:

a) para construção de alarmes sonoros;
b) para escrever grandes imagens projetáveis;
c) para escrever pequenas (minúsculas) imagens projetáveis;
d) N.R.A.

3) Uma das principais características dos LEDs é o fato de:

a) emitirem luz;
b) não emitirem luz;
c) serem construídos à base de cristal líquido;
d) N.R.A.

4) Os LEDs também são capazes de exibir gráficos. Isto se tornou possível através:

a) da aplicação de altas tensões em seus terminais;
b) da criação dos displays de plasma;

c) da miniaturização dos LEDs;
d) N.R.A.

5) Os displays de plasma são constituídos de:

a) uma mistura de gás neon e argônio;
b) uma mistura de gás carbônico e argônio;
c) uma mistura de gás hélio e neon;
d) N.R.A.

6) Com relação ao método de marcação, o traçador de gráfico mais utilizado é:

a) o de movimento de papel;
b) o de caneta (de nylon ou de feltro);
c) o traçador vertical;
d) o traçador escalonado.

7) A densidade de pontos por polegadas, impresso pelo cabeçote eletrostático de um traçador Raster-scan está entre:

a) 10 e 20;
b) 200 e 500;
c) 2000 e 2500;
d) 100 e 200.

8) Os valores das três freqüências ressonantes F1, F2 e F3, em hertz (Hz) são, respectivamente:

a) 1500, 500, 2500;
b) 2500, 1500, 500;
c) 500, 1500, 2500;
d) 2500, 500, 1500.

9) Para que um circuito eletrônico pudesse pronunciar as consoantes, foram adicionados ao circuito, utilizado para a pronúncia de vogais, os seguintes elementos:

a) um ressonador fricativo;
b) um ressonador nasal;
c) uma fonte de ruído de aspiração;
d) os itens a, b e c estão corretos.

10) O dispositivo periférico chamado de quadro sinóptico é utilizado:

a) no controle automático de processos;
b) nos sintetizadores de fala;
c) no controle manual de processos;
d) N.R.A.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 · CEP 86100 · LONDRINA - PARANÁ

LIÇÃO MC-10

MICROPROCESSADOR 8008

INTRODUÇÃO:

Para que possamos compreender melhor os princípios de funcionamento do microprocessador 8080 e seus similares, vamos, inicialmente fazer, um breve estudo sobre o seu antecessor, o 8008, de estrutura mais simples e cujas características (algumas delas) foram reaproveitadas no próprio 8080.

Antes de darmos início ao nosso estudo, queremos deixar bem claro que não será possível estudarmos todos os microprocessadores, pois, atualmente, dispomos de um número incalculável de microprocessadores, desde os de aplicação geral até os de aplicação específica. O quadro da figura 1 apresenta um resumo dos principais microprocessadores de uso geral, e seus fornecedores mais populares, encontrados hoje no mercado, dos quais estudaremos apenas alguns.

Fabricante	Microprocessador (número do modelo)
INTEL	8080 8080A 8085 8086
ZILOG	Z80 Z8000
MOTOROLA	6800 6802 6809
SIGNETICS	2650A

FIGURA 01

BREVE HISTÓRICO DOS MICROPROCESSADORES:

O surgimento dos microprocessadores não se deve a nenhuma manobra planificada de antemão, sob o ponto de vista comercial, e nem a algum tipo de projeto engenhoso e ambicioso. O surgimento destes foi puramente acidental, podendo-se dizer que os primeiros microprocessadores que apareceram no mercado, foram verdadeiros acidentes tecnológicos. Com isso, os microprocessadores passaram a crescer de forma desordenada e incontrolável, transmitindo, conseqüentemente, os erros iniciais de projeto e seus inconvenientes até os dias de hoje; erros esses escondidos atrás do nome "compatibilidade". Daí a importância de se conhecerem algumas facetas da história desses componentes.

A primeira aparição autêntica de transistores capazes de funcionar corretamente, se deu no final da década de 40, após a 2ª Guerra Mundial. Somente 10 anos mais tarde é que St. Clair Kilby obtinha o primeiro modelo de circuito integrado na Texas Instruments. Nessa mesma época, Hoerni desenvolve, na Fairchild, o processo planar, utilizado para a fabricação de grandes quantidades de circuitos integrados.

A integração de circuitos foi progredindo rapidamente. Em 1964 surgiu a técnica SSI (integração em pequena escala). Era o princípio das portas lógicas completas, integradas em uma só pastilha de silício.

Os primeiros frutos produzidos a partir da técnica LSI (integração em larga escala) começam a surgir por volta de 1971: a primeira memória de 1 K bit, a primeira UART (Receptor-Transmissor assíncrono universal) e o primeiro microprocessador de uso geral (o 4004, fabricado pela Intel).

A busca de níveis de integração mais elevados tiveram origem, a princípio, nos contratos governamentais, em especial os da NASA (Centro Aeroespacial dos Estados Unidos) e de outros programas militares. Devido à diminuição dos contratos de pesquisa, realizados pela NASA, os fabricantes foram forçados a buscar outras alternativas de mercado. Foi nesta época que os fabricantes de circuitos integrados encontraram a primeira opção de mercado para os seus produtos, junto ao grande público: surgiram então as primeiras calculadoras eletrônicas de mesa, em seguida as de bolso. Pela primeira vez na história da eletrônica se vendiam dispositivos LSI ao público. A produção e a venda de centenas de milhares de unidades converteu-se, desta forma, em uma realidade.

Até o início de 1971 haviam sido desenvolvidos somente dois produtos LSI stander: a memória dinâmica ativa de 1 K-bit e a primeira UART (conversor série-paralelo). Até então, não se tinha a menor idéia de qual seria o próximo componente stander a ser comercializado. Foi então que, por um acidente técnico, surgiram os microprocessadores.

O projeto e desenvolvimento do microprocessador 4004 da Intel, introduzido no mercado em 1971, foi resultado de um contrato firmado entre a Intel e um fabricante japonês de calculadoras eletrônicas de mesa. Os primeiros compradores do 4004 (microprocessador de 4 bits) tiveram que firmar um contrato no qual se comprometiam a não utilizarem o circuito em calculadoras de mesa, por um prazo de um ano. Esse primeiro microprocessador de uso geral foi desenvolvido, claramente, como uma calculadora. Portanto, não era muito potente e, de um certo modo, inadequado para um computador de uso geral.

Em 1972, a Intel lançou o microprocessador 8008, sendo o primeiro microprocessador de uso geral, de 8 bits. Passados alguns anos, a "Display Terminals Corporation", atualmente conhecida como "Datapoint", abriu uma concorrência pública para a fabricação de um processador monolítico, capaz de

controlar um tubo de raios catódicos (TRC), por intermédio de um único circuito integrado. Duas companhias se apresentaram e obtiveram o contrato para o desenvolvimento do projeto: a Texas Instruments e a Intel. Após alguns meses, a Texas Instruments desistia do projeto. A Intel prosseguiu até obter o componente que satisfazia praticamente todas as exigências funcionais, impostas pela Datapoint, exceto uma: era muito lento. Era o 8008 que estava nascendo. A Datapoint, por uma série de fatores, resolveu construir o seu controlador TRC com elementos bipolares. A Intel, por sua vez, sem saber ao certo o que fazer com este novo componente que acabara de desenvolver, resolveu então lançá-lo no mercado com a esperança de aumentar a venda dos circuitos de memória. Ao contrário do que se esperava, as pesquisas de mercado mostraram que o 8008 era aplicado para desempenhar uma série de tarefas diferentes.

As vendas, ao contrário do que se esperava, cresceram rapidamente, consolidando o novo produto. Em face a todo esse sucesso, a Intel formou uma nova equipe de pesquisa e, um ano mais tarde, lançou o sucessor do 8008: o 8080. Simultaneamente, todos os competidores importantes se puseram a desenvolver sua própria versão do microprocessador 8080. Nos anos seguintes se introduziram no mercado todos os microprocessadores stander que conhecemos atualmente.

Estamos, agora, presenciando a terceira geração dos microprocessadores na qual estão sendo lançados no mercado os sucessores do 8080. O Z-80 da Zilog e o 8085 da Intel são os mais recentes microprocessadores monopastilha.

A seguir, apresentaremos um pequeno estudo sobre os primeiros microprocessadores, o 4004 e o 8008, para, em seguida, passarmos ao estudo do 8080 e do Z-80, os quais nos interessam mais de perto.

MICROPROCESSADOR 4004:

Fabricado pela Intel, o 4004 foi o primeiro microprocessador (de 4 bits) que surgiu no mercado. A sua estrutura de direcionamento é tal que existe diferença entre programas e dados, tanto sob o ponto de vista do meio de armazenamento

utilizado, como do sistema de direcionamento. Eis algumas características do 4004:

a) Encapsulamento com 16 terminais;
b) 4 K de memória ROM;
c) 16 registros;
d) Stack de direção de retorno 4 x 12.

A figura 2 nos mostra o diagrama de blocos do microprocessador MCS-4 que utiliza como UCP um 4004.

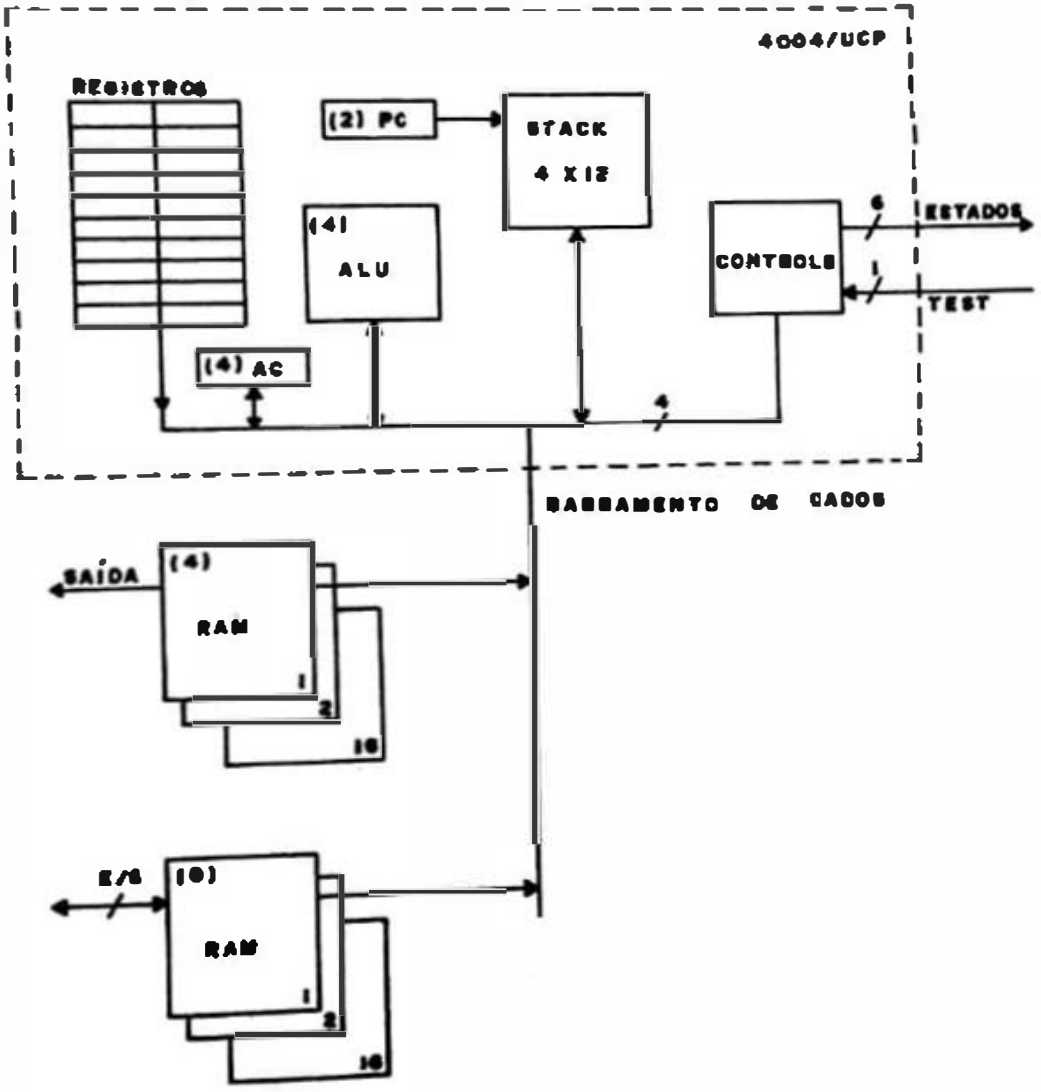


FIGURA 02

O MCS-4 é constituído por chips que realizam as funções de UCP, RAM e ROM, podendo ser interconectados com vários meios de armazenamento. A comunicação entre os chips se realiza através de um único barramento de 4 bits, permitindo assim, que todos os chips do MCS-4 sejam do tipo dual-in-line, com 16 terminais. O microprocessador tem um conjunto de registros para uso geral, constituído de registros de 4 bits, além de um acumulador, também de 4 bits.

As direções de retorno se armazenam em um stack de 4 posições, no qual o dígito selecionado é o contador do atual programa.

Tanto o stack como o contador de programa são de 12 bits o que nos leva a uma capacidade de armazenamento de 4 K bits de programa. O único modo de direcionamento de que se dispõe para este caso é o direcionamento indireto, através de registros. Para se selecionar um dado armazenado é necessária uma seqüência inteira de instruções. No entanto, uma cuidadosa colocação dos dados na memória RAM pode facilitar em muito o direcionamento.

MICROPROCESSADOR 8008:

O microprocessador 8008, fabricado pela Intel e sucessor do 4004, foi o primeiro microprocessador construído em um único circuito integrado. A figura 3 nos mostra o diagrama de blocos do 8008.

Este microprocessador de 8 bits, construído com a tecnologia P-MOS, possui uma velocidade de operação da ordem de 20μs por instrução. Existe também uma versão melhorada do mesmo, o 8008-1, o qual realiza as instruções em 12,5 μs. O encapsulamento com 18 terminais obriga a que todos os dados e direção sejam transportados por um único barramento bidirecional de 8 bits.

Com o objetivo de facilitar a compreensão da forma com que opera a maioria dos microprocessadores, iremos descrever, resumidamente, a estrutura do microprocessador 8008. Para tanto, vamos passar a descrever alguns elementos do diagrama de blocos, apresentado na figura 3.

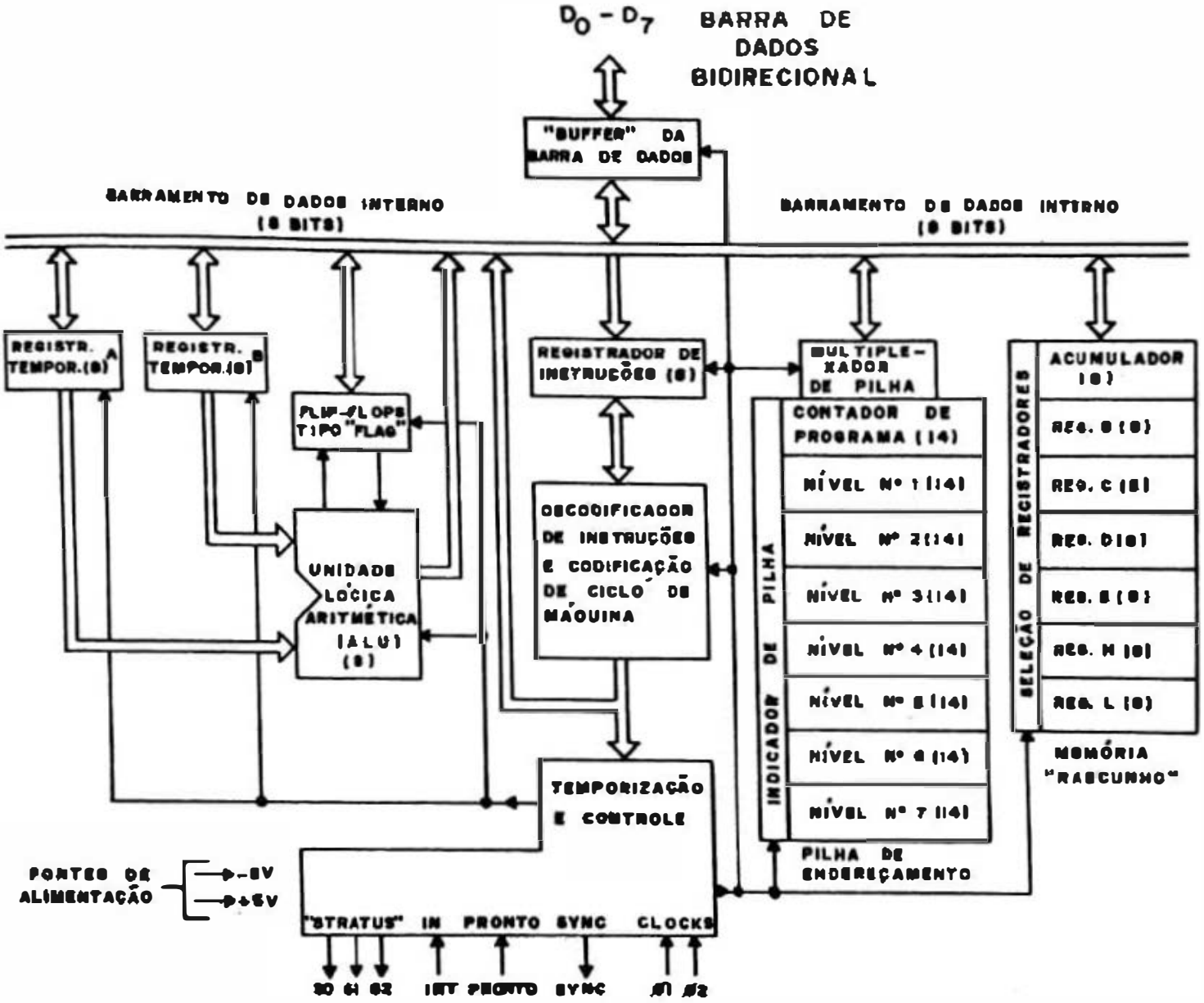


FIGURA 03

a) Estrutura de barras:

O microprocessador 8008 manipula dados em 8 bytes de 8 bits. Devido a este fato, a grande parte dos elementos internos da UCP são interligados por meio de barramentos de dados de 8 bits. O "buffer" do barramento de dados (um excitador bidirecional desse barramento) serve de contato entre a UCP e o exterior, tanto na recepção quanto no envio de dados.

b) Decodificador de instrução:

O registrador de instruções nada mais é do que um registrador de armazenagem temporária, de 8 bits, que recebe instruções via memórias externas. O decodificador de instruções por sua vez, irá "decifrar" as operações que devem ser executadas pelo microprocessador.

c) Contador de programa ou PC (Program Counter):

Trata-se de um circuito, composto de uma pilha de oito registradores, em oito níveis diferentes, que lhe permite manter-se informado das sub-rotinas do programa. Uma instrução de chamada (CALL), por exemplo, faz com que a UCP (ou CPU) deixe de processar o programa principal do microcomputador e passe para uma sub-rotina, cujas instruções estão armazenadas em uma outra parte da memória. Desta forma, o endereço anterior de memória permanece no nível nº 0 (programa principal que a UCP estava executando), enquanto que o atual fica no nível 1 da pilha (sub-rotina que está sendo executada).

Assim sendo, o 8008 permite acomodar sete subrotinas na pilha do contador de programa. Depois de executada a subrotina, um sinal de "retorno" faz com que a UCP volte para o nível nº 0 da pilha. Ao "indicador de pilha" (stack pointer) é dada a função de informar, na pilha, o nível que está sendo processado.

No 8008, cada nível da pilha do contador de programa possui uma capacidade de 14 bits. Esta capacidade permite que o microprocessador 8008 enderece 2^{14} bytes ou aproximadamente 16 K bytes de memória. Os endereços da memória são divididos em oito bits de baixa ordem e seis bits de alta ordem.

d) Unidade lógica aritmética (ULA):

A unidade lógica aritmética (ULA) ou ALU - Arithmetic Logic Unit - do 8008 pode efetuar adições, subtrações e várias outras operações lógicas, como E, OU, OU Exclusivo, Comparação, etc.

e) Registradores temporários:

O microprocessador 8008 possui dois registradores temporários, cuja função é a de providenciar a armazenagem de dados intermediários. Denominados registradores *a* e *b* (não confundir com os indexadores A e B), são utilizados pela UCP para realizar a transferência interna de dados, não sendo, portanto, diretamente acessíveis, através de instruções de programa (que é o caso dos indexadores A e B).

f) Temporização e Controle:

O microprocessador 8008 necessita de um sinal de clock (CK) de duas fases, normalmente gerado por circuitos externos. São, na verdade, duas freqüências (F_1 e F_2), utilizadas para sincronizar os vários estágios, durante as operações internas do processador.

g) Flip-flops tipo "Flag":

Estes flip-flops são utilizados para indicarem o momento exato em que certas operações foram completadas. Os "flags" são acionados pela operação da unidade lógica aritmética (ULA) e são em número de quatro: o de transporte (C), o de paridade (P), o de sinal (S) e o de zero (Z). Cada um deles pode ser testado por meio de instruções secundárias condicionadas, fazendo com que o fluxo subseqüente do programa fique dependendo do estado de cada "flag" testado.

h) Indexadores:

Durante o processamento de dados, é comum utilizar-se uma pequena memória de fácil acesso, denominada área para "rascunho" (em inglês "scratch pad"). O microprocessador 8008 possui sete memórias de "rascunho", as quais são chamadas de indexadores. Esses indexadores são codificados como A, B, C, D, E, H e L. Através de certas instruções do 8008, consegue-se realizar a transferência de dados entre esses elementos, de forma direta, o que facilita em muito o processamento de dados.

O indexador A, também denominado acumulador, foi designado como sendo um registrador especialmente utilizado para a comunicação entre os vários elementos do microcomputador. Assim sendo, os dados de entrada do 8008, vindos de um dispositivo periférico de entrada, externo ao microprocessador, - vão acabar no indexador A, no término de uma instrução de entrada. Analogamente, os dados de saída, destinados a um dispositivo periférico de saída, devem, primeiramente, ser carregados no indexador A, antes da execução de uma instrução - de saída. A unidade de lógica aritmética (ULA) realiza operações aritméticas com dados no indexador A e guarda nele o resultado dessas operações.

Já os indexadores H e L foram especialmente projetados para auxiliar o relacionamento das memórias externas com a UCP, tanto para a leitura de dados como para a escrita dos mesmos nessas memórias (no caso de escrita dos dados, somente é válido em se tratando de memórias RAM). Deve-se tomar muito cuidados para não confundir esta função com a função dos registradores, ligados ao contador de programa, e cuja função é, justamente, a de buscar instruções na memória.

Vimos assim, muito embora de uma forma bem resumida, os principais elementos que compõem o microprocessador 8008 conforme o diagrama de blocos apresentado na figura 3. Dando continuidade ao estudo do microprocessador 8008,vejamos mais algumas particularidades a respeito do mesmo.

O 8008 requer, como já mencionamos, um sinal de clock externo de duas fases que deve ser fornecido às duas entradas de clock, F_1 e F_2. Com o auxílio de um flip-flop interno, disparado pelo sinal F_2, produz-se um terceiro sinal, chamado SYNC (ou abreviadamente SY). Com esses três sinais obtêm-se as quatro fases, conforme nos mostra o quadro apresentado na figura 4.

$F_{11} = F_1 \cdot SYNC$	$F_{12} = F_1 \cdot \overline{SYNC}$
$F_{21} = F_2 \cdot SYNC$	$F_{22} = F_2 \cdot \overline{SYNC}$

FIGURA 4

Na verdade, a utilização desse flip-flop para produzir o terceiro sinal é um artifício que permite gerar quatro fases, com apenas três terminais do circuito integrado. Essas quatro fases são utilizadas na sincronização de transferência de dados, na estrutura interna do barramento do 8008 e também na sincronização da UCP com o "hardware" externo.

Durante as operações do processador, o 8008 passa por oito estados diferentes. A mudança de um estado para outro é, geralmente, realizada numa borda ascendente do sinal SYNC. O quadro da figura 5 nos mostra os oito estados possíveis para o 8008.

FIGURA 05

Estado	S_2	S_1	S_0	Função
T_1	0	1	0	endereço de memória de baixa ordem
T_{1_L}	1	1	0	endereço de memória de baixa ordem, porém com interrupção reconhecida
T_2	1	0	0	endereço de memória de alta ordem
ESPERA (WAIT)	0	0	0	esperar a memória mais lenta
T_3	0	0	1	receber dados de uma CPU externa para escrever na RAM
PARADA	0	1	1	espera para interrupção
T_4	1	1	1	transferência interna de dados
T_5	1	0	1	transferência interna de dados

S_0, S_1 e S_2 são sinais de estados, gerados pelo 8008, que podem ser obtidos em três terminais do seu encapsulamento. Esses sinais de estados podem ser utilizados na sincronização com os circuitos externos.

A execução de uma instrução do 8008, dependendo da instrução, requer um, dois ou três ciclos de máquina para ser completada. Temos quatro tipos de ciclos à disposição, os quais estão representados no quadro da figura 6.

CC1	CC2	Símbolo do ciclo	Nome	Descrição
0	0	PCL	Ciclo de instrução	o 1º byte da instrução é lido na memória
1	0	PCR	ciclo de leitura	dados ou bytes adicionais da instrução são lidos na memória
0	1	PCC	ciclo de comando	dados para entrar ou sair
1	1	PCW	ciclo de escrita	dados a serem escritos na memória RAM.

FIGURA 6

Os bits de controle CC1 e CC2 estão presentes no barramento de dados no instante T_2, sendo eles os dois bits de mais alta ordem (D_7 e D_6).

Na execução de cada instrução, o 8008 não é obrigado a passar por todos os estados ou por todos os ciclos de máquina. Normalmente, o que ocorre são transições de estado, conforme a instrução que está sendo executada.

O microprocessador 8008 possui um conjunto de 256 instruções de 8 bits, as quais são mostradas na figura 7, em ordem numérica.

Octal	Binário			Mnemônico	Descrição
000	00	000	000	HLT	Halt until interrupted
001	00	000	001	HLT	Halt until interrupted
002	00	000	010	RLC	Rotate A register left
003	00	000	011	RFC	Return false carry
004	00	000	100	ADI*	Add immediate to the A register
005	00	000	101	RST 000	Restart at location 000000
006	00	000	110	LAI*	Load A immediate
007	00	000	111	RET	Return one level down in stack
010	00	001	000	INB	Increment the B register
011	00	001	001	DCB	Decrement the B register
012	00	001	010	RRC	Rotate A register right
013	00	001	011	RFZ	Return false zero
014	00	001	100	ACI*	Add immediate with carry to A reg
015	00	001	101	RST 010	Restart at location 000010
016	00	001	110	LBI*	Load B immediate
017	00	001	111	RET	*(Same as 007)*
020	00	010	000	INC	Increment the C register
021	00	010	001	DCC	Decrement the C register
022	00	010	010	RAL	Rotate A reg. left thru carry
023	00	010	011	RFS	Return false sign
024	00	010	100	SUI*	Subtract immediate from A register
025	00	010	101	RST 020	Restart at location 000020
026	00	010	110	LCI*	Load C immediate
027	00	010	111	RET	*(Same as 007)*
030	00	011	000	IND	Increment the D register
031	00	011	001	DCD	Decrement the D register
032	00	011	010	RAR	Rotate A reg. right thru carry
033	00	011	011	RFP	Return false parity
034	00	011	100	SBI*	Subtract immediate with borrow A
035	00	011	101	RST 030	Restart at location 000030
036	00	011	110	LDI*	Load D immediate
037	00	011	111	RET	*(Same as 007)*
040	00	100	000	INE	Increment the E register
041	00	100	001	DCE	Decrement the E register
042	00	100	010	——	*(Undefined)*
043	00	100	011	RTC	Return true carry

FIGURA 07

Octal	Binário			Mnemônic	Descrição
044	00	100	100	NDI*	AND immediate with A register
045	00	100	101	RST	Restart at location 000040
046	00	100	110	LEI*	Load E immediate
047	00	100	111	RET	*(Same as 007)*
050	00	101	000	INH	Increment the H register
051	00	101	001	DCH	Decrement the H register
052	00	101	010	——	*(Undefined)*
053	00	101	011	RTZ	Return true zero
054	00	101	100	XRI*	Exclusive or immediate with A reg
055	00	101	101	RST 050	Restart at location 000050
056	00	101	110	LHI*	Load H immediate
057	00	101	111	RET	*(Same as 007)*
060	00	110	000	INL	Increment the L register
061	00	110	001	DCL	Decrement the L register
062	00	110	010	——	*(Undefined)*
063	00	110	011	RTS	Return true sign
064	00	110	100	ORI*	OR immediate with A reg.
065	00	110	101	RST 060	Restart at location 000060
066	00	110	110	LLI*	Load L immediate
067	00	110	111	RET	*(Same as 007)*
070	00	111	000	——	*(Undefined)*
071	00	111	001	——	*(Undefined)*
072	00	111	010	——	*(Undefined)*
073	00	111	011	RTP	Return true parity
074	00	111	100	CPI*	Compare immediate with A reg.
075	00	111	101	RST 070	Restart at location 000070
076	00	111	110	LMI*	Load memory immediate
077	00	111	111	RET	*(Same as 007)*
100	01	000	000	JFC*	Jump false carry
101	01	000	001	INP 0	Input 0 to A; out A reg.
102	01	000	010	CFC*	Call false carry
103	01	000	011	INP 1	Input 1 to A; out A reg.
104	01	000	100	JMP*	Jump to indicated memory address
105	01	000	101	INP 2	Input 2 to A; out A reg.
106	01	000	110	CAL*	Call the indicated subroutine
107	01	000	111	INP 3	Input 3 to A; out A reg.
110	01	001	000	JFZ*	Jump false zero
111	01	001	001	INP 4	Input 4 to A; out A reg.
112	01	001	010	CFZ*	Call false zero
113	01	001	011	INP 5	Input 5 to A; out A reg.

Octal	Binário			Mnemônico	Descrição
114	01	001	100	JMP*	(Same as 104)
115	01	001	101	INP 6	Input 6 to A; out A reg.
116	01	001	110	CAL*	(Same as 106)
117	01	001	111	INP 7	Input 7 to A; out A reg.
120	01	010	000	JFS*	Jump false sign
121	01	010	001	OUT 10	Output A reg. to output 10
122	01	010	010	CFS*	Call false sign
123	01	010	011	OUT 11	Output A reg. to output 11
124	01	010	100	JMP*	(Same as 104)
125	01	010	101	OUT 12	Output A reg. to output 12
126	01	010	110	CAL*	(Same as 106)
127	01	010	111	OUT 13	Output A reg. to output 13
130	01	011	000	JFP*	Jump false parity
131	01	011	001	OUT 14	Output A reg. to output 14
132	01	011	010	CFP*	Call false parity
133	01	011	011	OUT 15	Output A reg. to output 15
134	01	011	100	JMP*	(Same as 104)
135	01	011	101	OUT 16	Output A reg. to output 16
136	01	011	110	CAL*	(Same as 106)
137	01	011	111	OUT 17	Output A reg. to output 17
140	01	100	000	JTC*	Jump true carry
141	01	100	001	OUT 20	Output A reg. to output 20
142	01	100	010	CTC*	Call true carry
143	01	100	011	OUT 21	Output A reg. to output 21
144	01	100	100	JMP*	(Same as 104)
145	01	100	101	OUT 22	Output A reg. to output 22
146	01	100	110	CAL*	(Same as 106)
147	01	100	111	OUT 23	Output A reg. to output 23
150	01	101	000	JTZ*	Jump true zero
151	01	101	001	OUT 24	Output A reg. to output 24
152	01	101	010	CTZ*	Call true zero
153	01	101	011	OUT 25	Output A reg. to output 25
154	01	101	100	JMP*	(Same as 104)
155	01	101	101	OUT 26	Output A reg. to output 26
156	01	101	110	CAL*	(Same as 106)
157	01	101	111	OUT 27	Output A reg. to output 27
160	01	110	000	JTS*	Jump true sign
161	01	110	001	OUT 30	Output A reg. to output 30
162	01	110	010	CTS*	Call true sign
163	01	110	011	OUT 31	Output A reg. to output 31

Octal	Binário			Mnemônico	Descrição
164	01	110	100	JMP*	(*Same as* 104)
165	01	110	101	OUT 32	Output A reg. to output 32
166	01	110	110	CAL*	(*Same as* 106)
167	01	110	111	OUT 33	Output A reg. to output 33
170	01	111	000	JTP*	Jump true parity
171	01	111	001	OUT 34	Output A reg. to output 34
172	01	111	010	CTP*	Call true parity
173	01	111	011	OUT 35	Output A reg. to output 35
174	01	111	100	JMP*	(*Same as* 104)
175	01	111	101	OUT 36	Output A reg. to output 36
176	01	111	110	CAL*	(*Same as* 106)
177	01	111	111	OUT 37	Output A reg. to output 37
200	10	000	000	ADA	Add A register to A register
201	10	000	001	ADB	Add B register to A register
202	10	000	010	ADC	Add C register to A register
203	10	000	011	ADD	Add D register to A register
204	10	000	100	ADE	Add E register to A register
205	10	000	101	ADH	Add H register to A register
206	10	000	110	ADL	Add L register to A register
207	10	000	111	ADM	Add memory to A register
210	10	001	000	ACA	Add A reg. to A reg. with carry
211	10	001	001	ACB	Add B reg. to A reg. with carry
212	10	001	010	ACC	Add C reg. to A reg. with carry
213	10	001	011	ACD	Add D reg. to A reg. with carry
214	10	001	100	ACE	Add E reg. to A reg. with carry
215	10	001	101	ACH	Add H reg. to A reg. with carry
216	10	001	110	ACL	Add L reg. to A reg. with carry
217	10	001	111	ACM	Add memory to A reg. with carry
220	10	010	000	SUA	Subtract A reg. from A reg.
221	10	010	001	SUB	Subtract B reg. from A reg.
222	10	010	010	SUC	Subtract C reg. from A reg.
223	10	010	011	SUD	Subtract D reg. from A reg.
224	10	010	100	SUE	Subtract E reg. from A reg.
225	10	010	101	SUH	Subtract H reg. from A reg.
226	10	010	110	SUL	Subtract L reg. from A reg.
227	10	010	111	SUM	Subtract memory from A with borrow
230	10	011	000	SBA	Subtract A reg. from A with borrow
231	10	011	001	SBB	Subtract B reg. from A with borrow
232	10	011	010	SBC	Subtract C reg. from A with borrow
233	10	011	011	SBD	Subtract D reg. from A with borrow
234	10	011	100	SBE	Subtract E reg. from A with borrow
235	10	011	101	SBH	Subtract H reg. from A with borrow

Octal	Binário			Mnemôn.	Descrição
236	10	011	110	SBL	Subtract L reg.from A with borrow
237	10	011	111	SBM	Subtract memory from A with borrow
240	10	100	000	NDA	AND the A reg. with the A reg.
241	10	100	001	NDB	AND the B reg. with the A reg.
242	10	100	010	NDC	AND the C reg. with the A reg.
243	10	100	011	NDD	AND the D reg. with the A reg.
244	10	100	100	NDE	AND the E reg. with the A reg.
245	10	100	101	NDH	AND the H reg. with the A reg.
246	10	100	110	NDL	AND the L reg. with the A reg.
247	10	100	111	NDM	AND memory with the A reg.
250	10	101	000	XRA	EXCL. OR the A reg. with A
251	10	101	001	XRB	EXCL. OR the B reg. with A
252	10	101	010	XRC	EXCL. OR the C reg. with A
253	10	101	011	XRD	EXCL. OR the D reg. with A
254	10	101	100	XRE	EXCL. OR the E reg. with A
255	10	101	101	XRH	EXCL. OR the H reg. with A
256	10	101	110	XRL	EXCL. OR the L reg. with A
257	10	101	111	XRM	EXCL. OR memory with A
260	10	110	000	ORA	OR the A reg. with the A reg.
261	10	110	001	ORB	OR the B reg. with the A reg.
262	10	110	010	ORC	OR the C reg. with the A reg.
263	10	110	011	ORD	OR the D reg. with the A reg.
264	10	110	100	ORE	OR the E reg. with the A reg.
265	10	110	101	ORH	OR the H reg. with the A reg.
266	10	110	110	ORL	OR the L reg. with the A reg.
267	10	110	111	ORM	OR memory with the A reg.
270	10	111	000	CPA	Compare A reg. with A reg.
271	10	111	001	CPB	Compare B reg. with A reg.
272	10	111	010	CPC	Compare C reg. with A reg.
273	10	111	011	CPD	Compare D reg. with A reg.
274	10	111	100	CPE	Compare E reg. with A reg.
275	10	111	101	CPH	Compare H reg. with A reg.
276	10	111	110	CPL	Compare L reg. with A reg.
277	10	111	111	CPM	Compare memory with A reg.
300	11	000	000	LAA	Load A register from A register
301	11	000	001	LAB	Load A register from B register
302	11	000	010	LAC	Load A register from C register
303	11	000	011	LAD	Load A register from D register
304	11	000	100	LAE	Load A register from E register
305	11	000	101	LAH	Load A register from H register
306	11	000	110	LAL	Load A register from L register
307	11	000	111	LAM	Load A register from memory

Octal	Binário			Mnemôn.	Descrição
310	11	001	000	LBA	Load B register from A register
311	11	001	001	LBB	Load B register from B register
312	11	001	010	LBC	Load B register from C register
313	11	001	011	LBD	Load B register from D register
314	11	001	100	LBE	Load B register from E register
315	11	001	101	LBH	Load B register from H register
316	11	001	110	LBL	Load B register from L register
317	11	001	111	LBM	Load B register from memory
320	11	010	000	LCA	Load C register from A register
321	11	010	001	LCB	Load C register from B register
322	11	010	010	LCC	Load C register from C register
323	11	010	011	LCD	Load C register from D register
324	11	010	100	LCE	Load C register from E register
325	11	010	101	LCH	Load C register from H register
326	11	010	110	LCL	Load C register from L register
327	11	010	111	LCM	Load C register from memory
330	11	011	000	LDA	Load D register from A register
331	11	011	001	LDB	Load D register from B register
332	11	011	010	LDC	Load D register from C register
333	11	011	011	LDD	Load D register from D register
334	11	011	100	LDE	Load D register from E register
335	11	011	101	LDH	Load D register from H register
336	11	011	110	LDL	Load D register from L register
337	11	011	111	LDM	Load D register from memory
340	11	100	000	LEA	Load E register from A register
341	11	100	001	LEB	Load E register from B register
342	11	100	010	LEC	Load E register from C register
343	11	100	011	LED	Load E register from D register
344	11	100	100	LEE	Load E register from E register
345	11	100	101	LEH	Load E register from H register
346	11	100	110	LEL	Load E register from L register
347	11	100	111	LEM	Load E register from memory
350	11	101	000	LHA	Load H register from A register
351	11	101	001	LHB	Load H register from B register
352	11	101	010	LHC	Load H register from C register
353	11	101	011	LHD	Load H register from D register
354	11	101	100	LHE	Load H register from E register
355	11	101	101	LHH	Load H register from H register
356	11	101	110	LHL	Load H register from L register
357	11	101	111	LHM	Load H register from A register
360	11	110	000	LLA	Load L register from A register

Octal	Binário			Mnemôn.	Descrição
361	11	110	001	LLB	Load L register from B register
362	11	110	010	LLC	Load L register from C register
363	11	110	011	LLD	Load L register from D register
364	11	110	100	LLE	Load L register from E register
365	11	110	101	LLH	Load L register from H register
366	11	110	110	LLL	Load L register from L register
367	11	110	111	LLM	Load L register from memory
370	11	111	000	LMA	Load memory from A register
371	11	111	001	LMB	Load memory from B register
372	11	111	010	LMC	Load memory from C register
373	11	111	011	LMD	Load memory from D register
374	11	111	100	LME	Load memory from E register
375	11	111	101	LMH	Load memory from H register
376	11	111	110	LML	Load memory from L register
377	11	111	111	HLT	Halt until interrupted

Observe que as instruções 042, 052, 062, 070, 071 e 072 foram designadas como indefinidas. Isto se deve ao fato dos fabricantes não especificarem o que essas instruções podem fazer. Entretanto, existe uma forma do microprocessador 8008 considerar essas instruções não-operativas (NOP), que é a de utilizá-las para preencher períodos de tempo, durante um programa.

O sistema de interrupção do 8008 funciona da seguinte forma: quando chega um pedido de interrupção, o qual é detectado por um nível lógico alto (1), o processador termina a instrução em curso e inibe a incrementação do contador de programa, através da emissão de um sinal que indica o reconhecimento da interrupção e, em seguida, passa para uma fase normal de extração da próxima instrução.

O microprocessador 8008 apresenta um sério problema de manejo de interrupções. Quando chega uma interrupção, é necessário se protegerem os registros que estão na memória. Para tanto, torna-se necessário o uso dos registradores de índice, os quais levam somente a dois tipos de solução: a) reservar dois registros para o uso exclusivo das interrupções; ou b) utilizar algum outro tipo de hardware. Este problema de manejo de interrupções é um exemplo dos vários efeitos se

cundários que podem ocorrer quando um sistema de direcionamento não é muito bem elaborado.

A maioria dos problemas apresentados pelo 8008 foi superado com a aparição do microprocessador 8080.

A figura 8 nos mostra a pinagem do 8008.

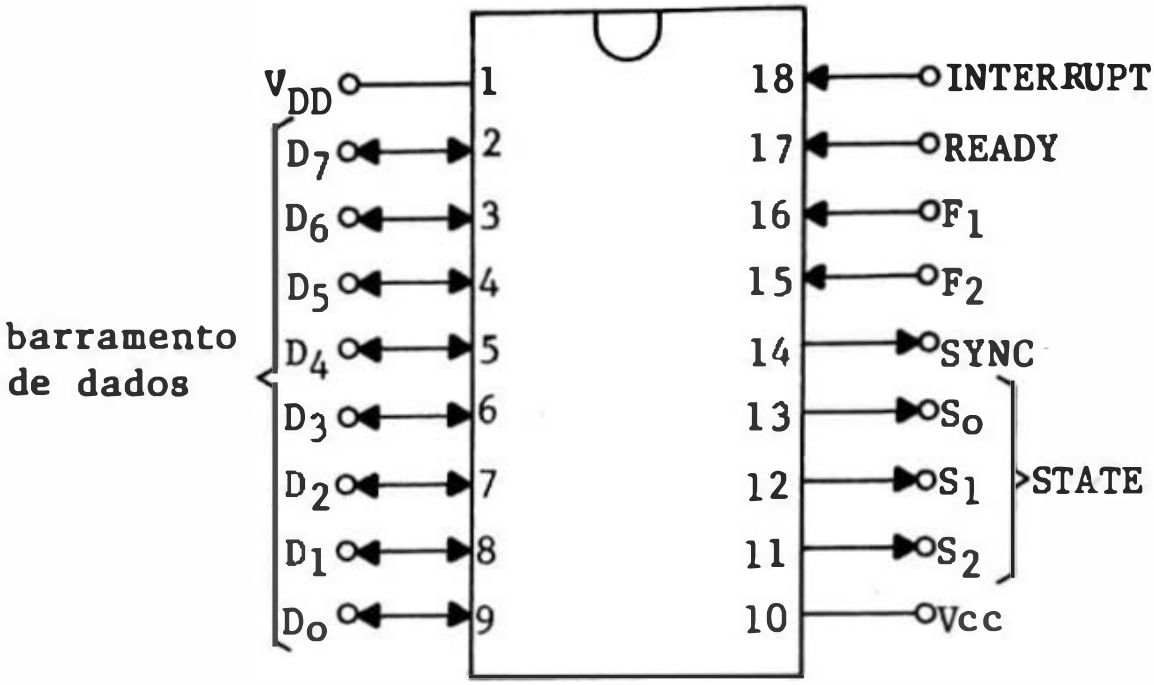


FIGURA 08

- Pinagem do microprocessador 8008, fabricado pela INTEL.

Com relação à pinagem do 8008, apresentada na figura 8, cabe destacar os pinos 1 e 10 (através dos quais se efetua a alimentação do microprocessador), respectivamente, V_{DD} e V_{cc} (-9 v e +5 v). Também chamamos a atenção para o pino 18 "Interrupt", pois como já foi dito anteriormente, quando existir um nível lógico alto (1) em sua entrada, o processador reconhecerá que há de fazer uma interrupção. Por fim, o pino "Ready" (pino 17) serve para sincronizar o 8008 com a memória.

Software do 8008:

A linguagem de máquina do 8008 é composta por 256 instruções (veja figura 7), codificadas em mnemônico. Por outro

lado, a Intel tem estudado uma variante de linguagem, através do aperfeiçoamento da linguagem PL/I da IBM, chamada PL/M, a qual deverá reduzir, de dez(10) vezes, o tempo de realização de um programa de aplicação.

Sistema Intellec 8 / Modelo 8

Trata-se de um microcomputador, formado basicamente por quatro partes distintas. Os diagramas de blocos de cada uma dessas partes serão mostrados a seguir.

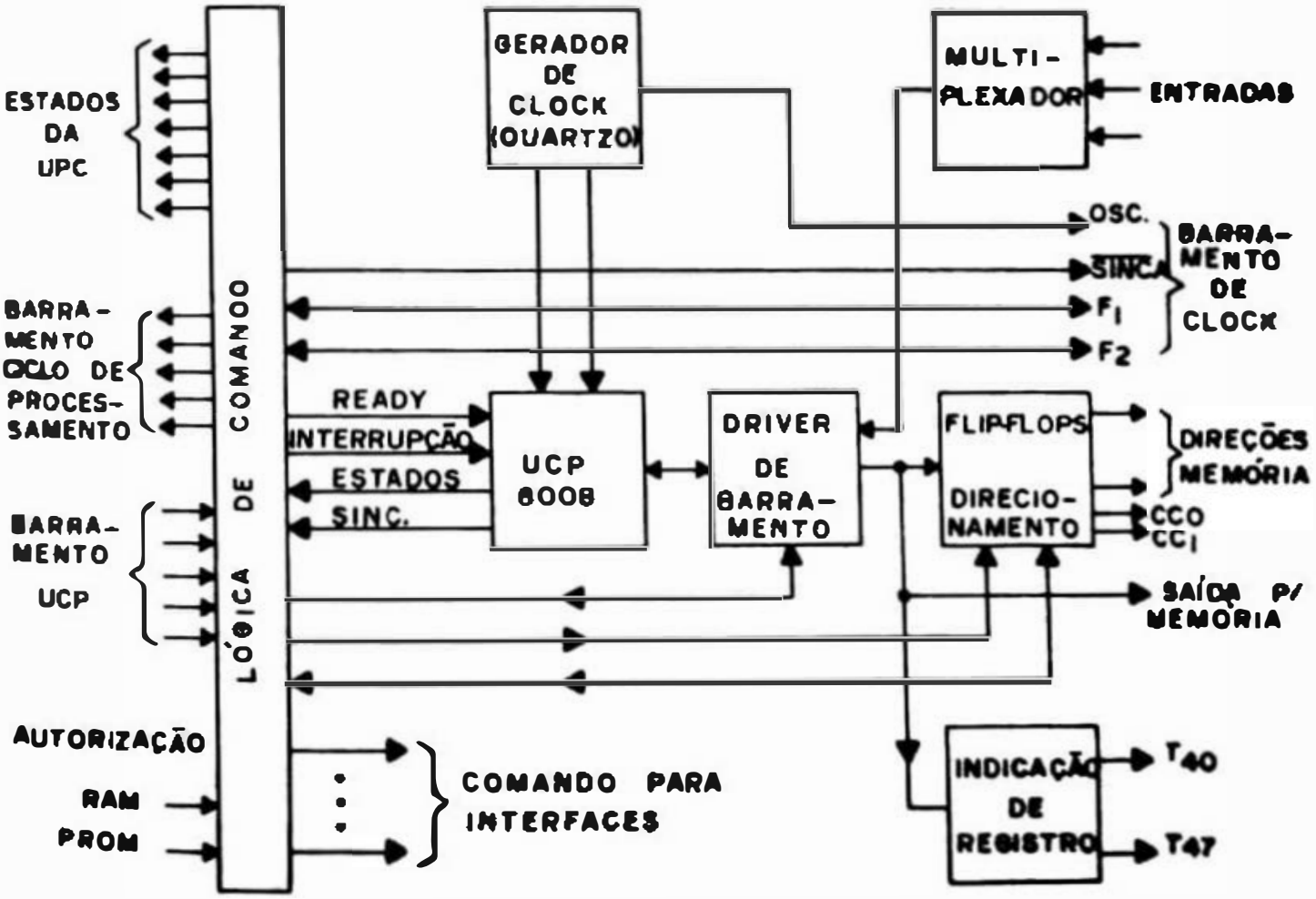


FIGURA 09 - Módulo completo, contendo a unidade central de processamento (UCP) 8008.

A figura 9 nos mostra o diagrama de blocos da unidade central de processamento, tendo como elemento principal o 8008.

Já o diagrama de blocos da figura 10 nos mostra o módulo de memória RAM.

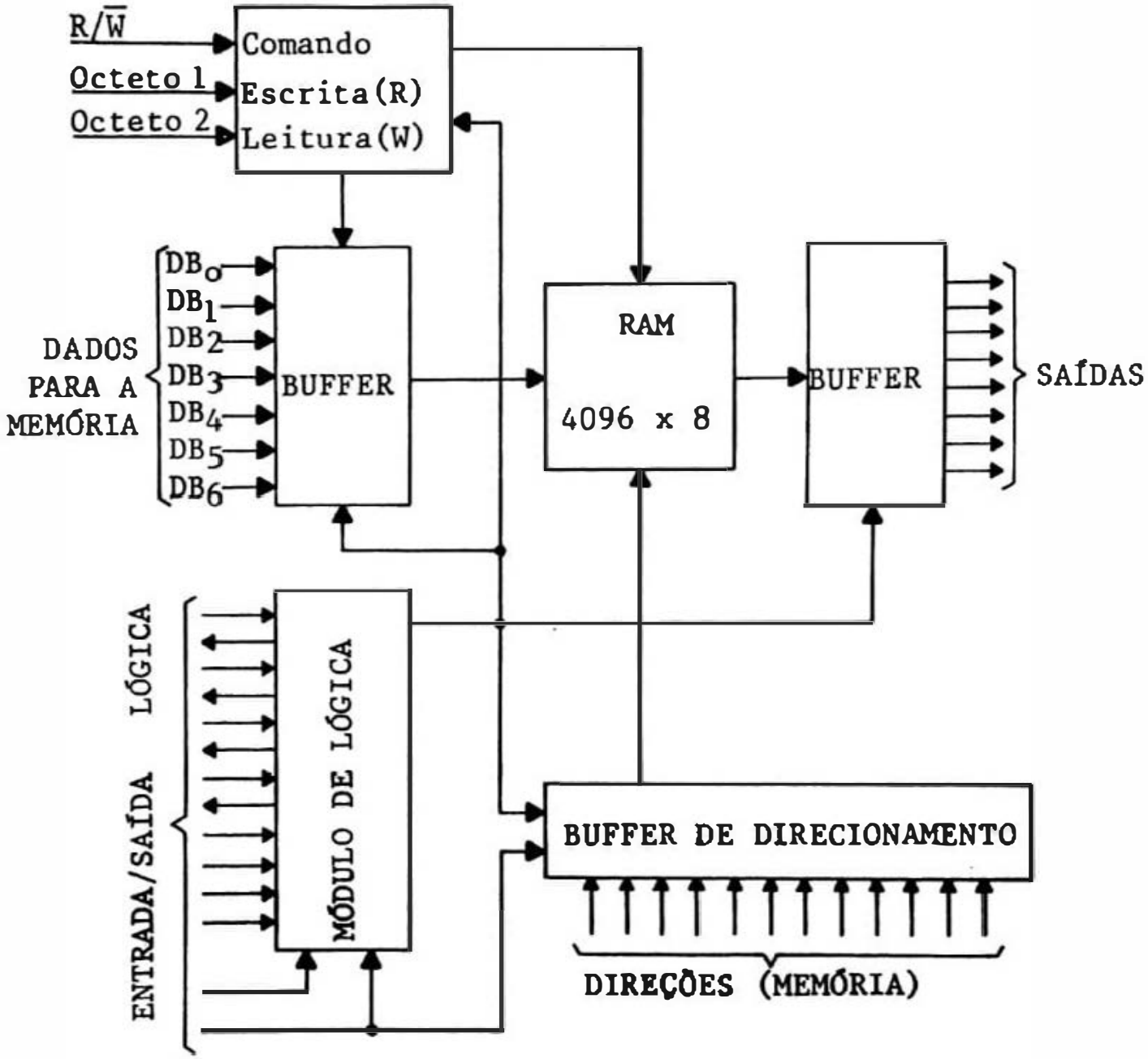


FIGURA 10 - Módulo completo da memória RAM.

A figura 11, por sua vez, nos mostra o módulo completo da memória PROM.

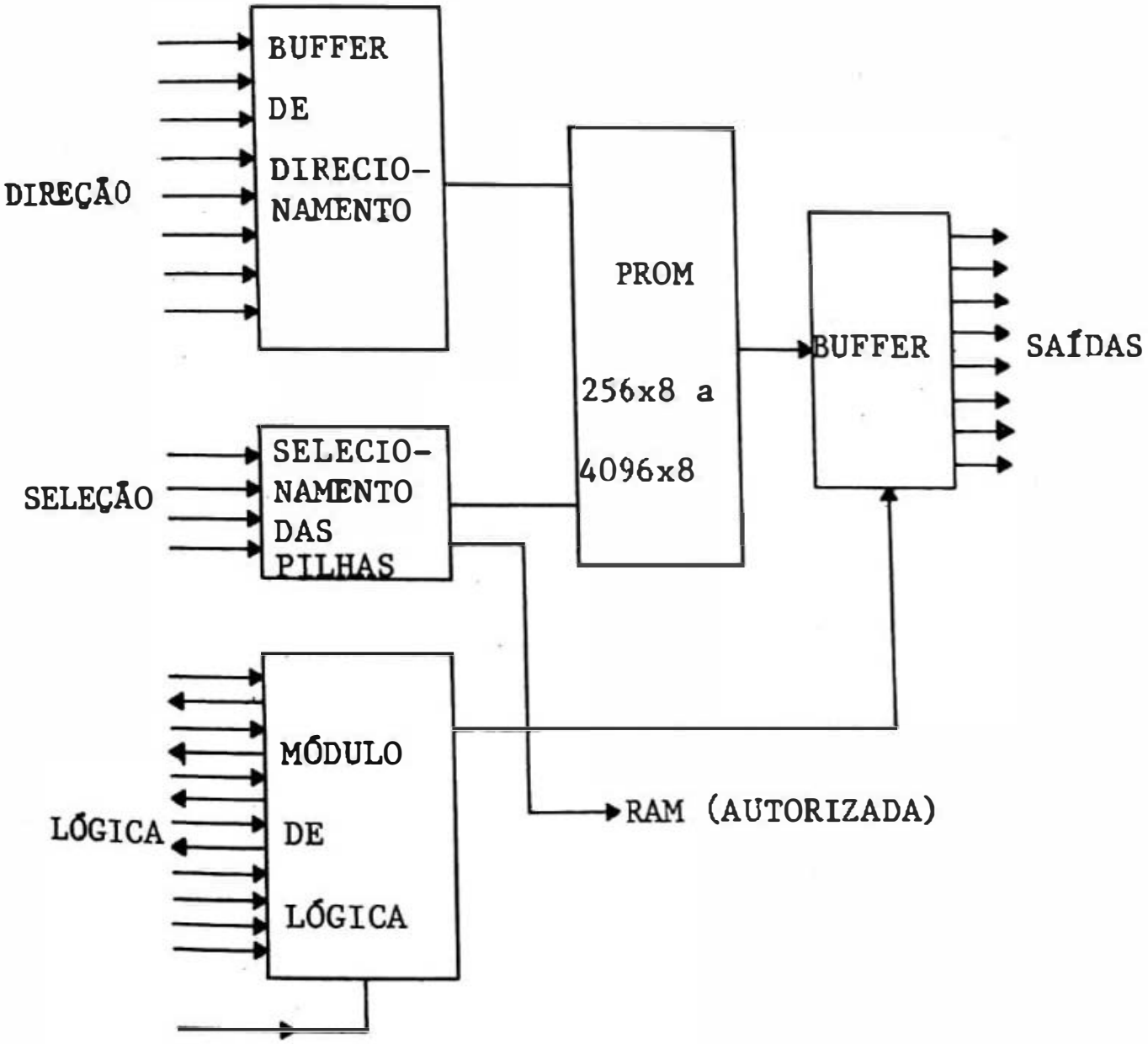


FIGURA 11 - Módulo de memória PROM programada para o 8008.

Para finalizar, a figura 12 nos mostra o módulo de entrada e saída (circuitos E/S), especialmente desenvolvido para o sistema Intellec/Modelo 8.

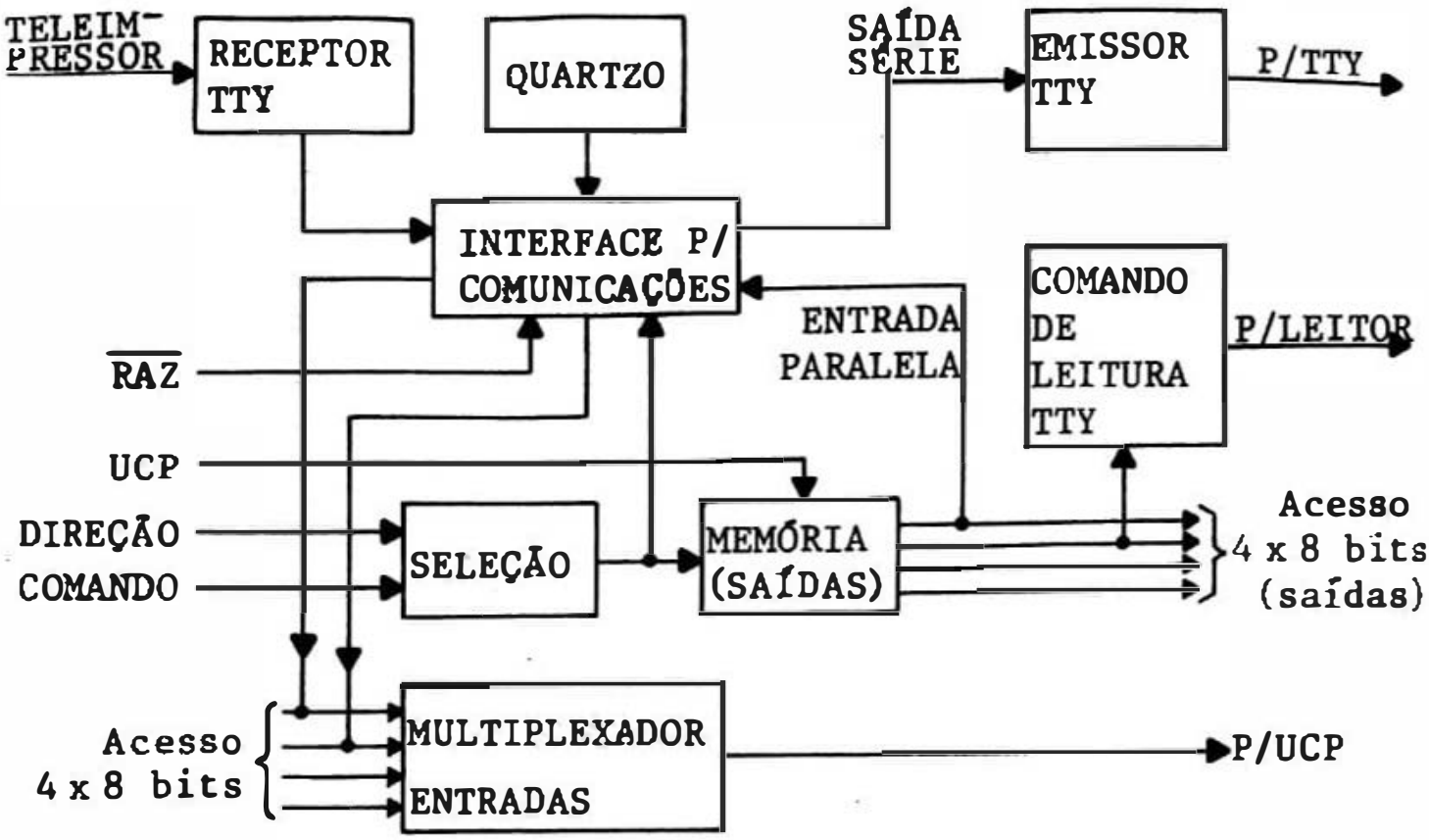


FIGURA 12 - Módulo de comando do sistema de entrada/saída (E/S) para o 8008.

O sistema Intellec 8/Modelo 8 foi desenvolvido pela própria Intel, fabricante do microprocessador 8008. Este sistema foi por nós apresentado apenas como exemplo de aplicação prática do 8008.

Na próxima lição, iniciaremos o estudo do microprocessador 8080, também fabricado pela Intel, o qual representa a "segunda geração" dos microprocessadores 8008.

Antes de passarmos ao exame desta lição, apresentaremos a seguir algumas tabelas comparativas dos microprocessadores de 4 bits e de 8 bits.

A tabela da figura 13 nos mostra os diversos tipos de microprocessadores de 4 bits.

FIGURA 13

MICROPROCESSADORES DE 4 BITS

	Intel 4004	Intel 4040	Rockwell PPS4	Rockwell PPS4/1	Rockwell PPS4/2	TMS 1000
Tipo	stander	stander	stander	µC	µC(2 c.)	µC
Tecnologia	PMOS	PMOS	PMOS	PMOS	PMOS	PMOS
Repertório de Instruções	46	60	50	50	50	curto
Tempo ciclo(µs)	10,8	10,8	5	12,5	5	15
Direcionamento direto (bits)	12	12	12	11	11	10 ou 11
Registros	16	24	4	3+RAM	2+RAM	4
Pilha	Hard(3)	Hard(7)	Hard(1)	Hard(2)	-	Hard(1)
Interrupções	0	1	1	2	0	-
Incorpora no circuito: Clock					sim	sim
Incorpora no circuito: ROM(bytes)				1K		2K
Incorpora no circuito: RAM(bytes)				96		128
Incorpora no circuito: Temporizador						
Incorpora no circuito: Arranq. inicial						
Incorpora no circuito: Outros						
Linhas (E/S)			12	26	12	19 a 24
Cápsula	16	16(24)	42	42	42	40 a 28
Alimentação	15	15	17	15	15	15
Observações		Instr. compatíveis c/ 4004	ADD em BCD em 30 µs		Repertório do PPS4	

A figura 14, por sua vez, nos mostra uma tabela comparativa entre os diversos microprocessadores de 8 bits.

Na figura 15 temos uma tabela comparativa entre os diversos microprocessadores de 8 bits, que na verdade, é uma continuação da tabela apresentada na figura anterior (14).

MICROPROCESSADORES DE 8 BITS

	INTEL				MOSTEK	MOTORO-	NAT. SEM.
	8008	8080	8085	8048	5065	LA 6800	SC/MP
Tipo	stander	stander	stander	µC	stander	stander	stander
Tecnologia	PMOS	NMOS	NMOS	NMOS	PMOS	NMOS	PMOS
Repertório de instruções	48	69	71	50	51	71	50
Tempo ciclo(µs)	12,5/20	1,3/2	1,3	2,5	10	2	5 a 25
Direcionamento direto	14	16	8 e 8	-	15	16	12
Registros	7	7	7	1	3	4	6
Pilha	Hard(7)	Soft	Soft	Hard(8)	-	Soft	Soft
Interrupções	1	1	4	1	2	1	1
Incorpora o circuito: Clock			sim	sim			sim
Incorpora o circuito: ROM(bytes)				1K			
Incorpora o circuito: RAM(bytes)				64			
Incorpora o circuito: Temporizador				sim			
Incorpora o circuito: Arranq. inicial							
Incorpora o circuito: Outros			8228				
Linh s E/S			2	27			2
Cápsula	18	40	40	40	40	40	40
Alimentação	-9; +5	+5; -5 +12	5	5	-12; +5; -5	5	+5; -7
Outras fontes		AMD/NEC TI/NS SIEMENS	NEC SIEMENS	SIEMENS		AMI/ SESCOSEM	

FIGURA 14

MICROPROCESSADORES DE 8 BITS (Cont.)

	RCA COSMAC	SIGNETICS 2650	ROCKWELL PPS8	ROCKWELL PPS8/2	WESTERN DIGITAL MCP 1600	ZILOG Z-80	INTERSIL IN6100
Tipo	stander	stander	stander	µC	microprogrdo.	stander	estad. 12
Tecnologia	CMOS	NMOS	PMOS	PMOS	NMOS	NMOS	CMOS
Repertório de instruções	59	75	90	109	-	8080 + muitas	PDP8
Tempo ciclo(µs)	3	4,8 e 9,6	4	5	16	1,6	2,5 a 11
Direcionamento direto	16	15	15	14	16	(16)	12
Registros	16(RAM)	7	3	3		17	3
Pilha	não	Hard(8)	Soft (S/SP)*	Soft (S/SP)*		Soft	
Interrupções	1	1	3	3		3	1
Incorpora o circuito: Clock	sim			sim		sim	sim
Incorpora o circuito: RAM(bytes)	sim						
Incorpora o circuito: Arranq. inicial			sim	sim			
Incorpora o circuito: Outros						8228 + refresc. RAM	
Linhas E/S							
Cápsula	40	4	42	42		40	40,28
Alimentação	3 a 10	5	17	17		5	5
Outr s fontes						Mostek/ NEC	Harris
Observações	consumo baixo (50 mW)	somador estático direc.	soma BCD em 12 µs		repertório inst ções do PDP 11	repertório do 8080	repertório do PDP 8 10mW estático

*S.P. de 5 Bits.

FIGURA 15

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-10

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) O surgimento dos microprocessadores se deve:

a) a uma manobra planificada de antemão, sob o ponto de vista comercial;

b) a um projeto engenhoso e ambicioso;

c) a verdadeiros acidentes tecnológicos.

d) N.R.A.

2) O microprocessador 4004, fabricado pela Intel, foi introduzido no mercado em:

a) 1971
b) 1981
c) 1961
d) 1969

3) O microprocessador 8008, lançado no mercado em 1972, foi o primeiro microprocessador de uso geral de:

a) 4 bits;
b) 6 bits;
c) 16 bits;
d) 8 bits.

4) Os microprocessadores Z-80 e 8085 são alguns exemplos de microprocessadores de:

a) 1ª geração;
b) 3ª geração;
c) 2ª geração;
d) N.R.A.

5) O microprocessador 4004 possui:

a) 4 K de memória ROM;
b) 16 registros;
c) 16 terminais;
d) os itens a, b e c estão corretos.

6) A velocidade de operação do 8008 é da ordem de:

a) 2 µs;
b) 200 µs;
c) 20 µs;
d) 12,5 µs.

7) Praticamente todos os elementos internos da UCP do 8008 são interligados por meio de:

a) barramentos de dados de 16 bits;
b) barramentos de dados de 8 bits;
c) barramentos de "buffer";
d) N.R.A.

8) Providenciar a armazenagem de dados temporários, no 8008, é função dos dois:

a) dos dois registradores temporizadores:
b) do buffer de saída;
c) do buffer de entrada;
d) N.R.A.

9) Os indexadores do 8008 também recebem o nome de:

a) memórias tampão;
b) memórias de acesso aleatório;
c) memórias de auxiliares;
d) memórias de rascunho.

10) O estado T_3, que ocorre durante o ciclo de operação do 8008, pode ser identificado pelo binário:

a) 1 1 1;
b) 1 0 0;
c) 0 0 1;
d) 1 0 1.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-11

MICROPROCESSADORES 8080 e 8085

INTRODUÇÃO

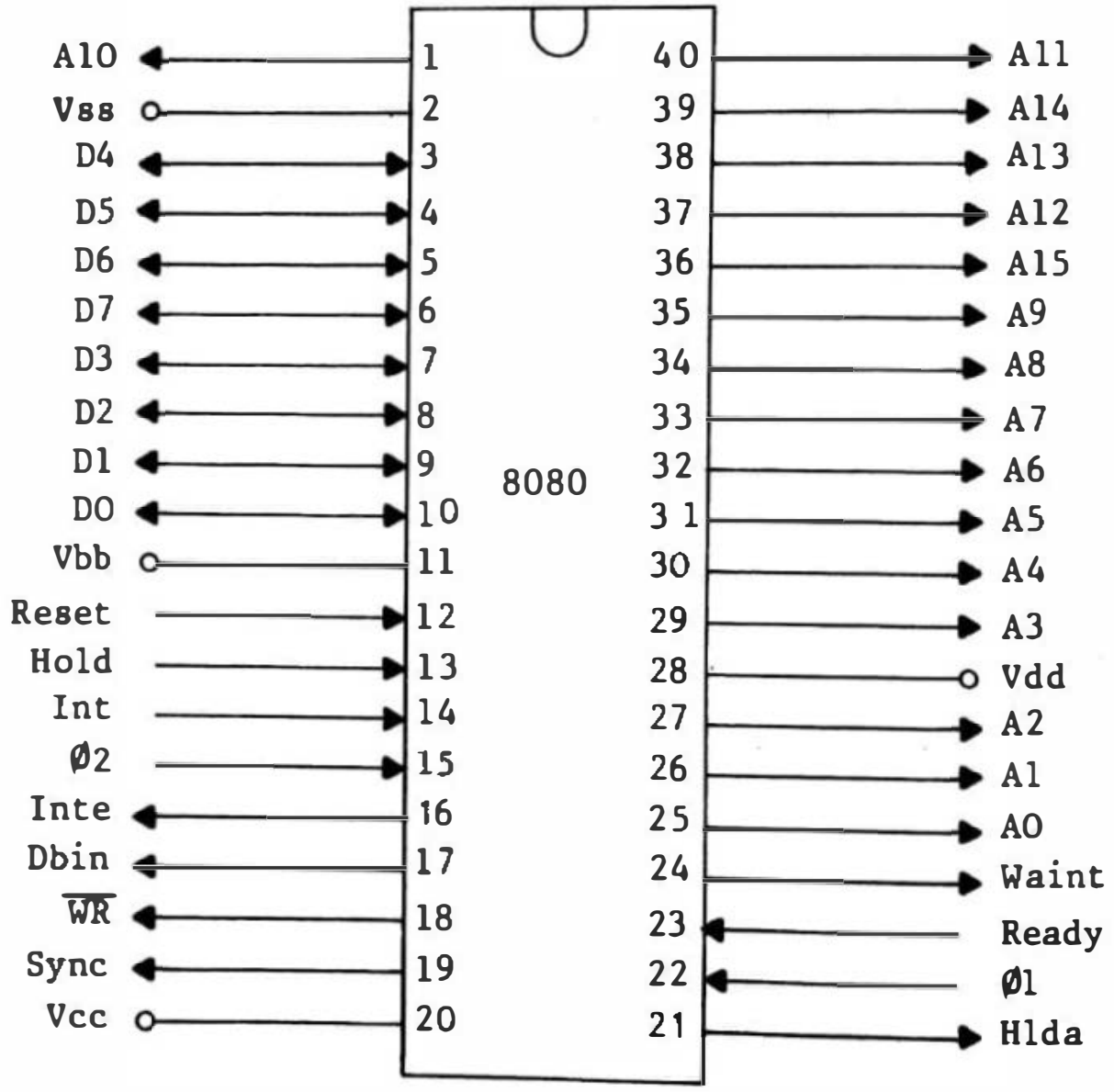
Dando continuidade ao estudo dos microprocessadores, estudaremos nesta lição os microprocessadores 8080 e 8085.

Pertencente à segunda geração dos microprocessadores, o 8080 nasceu da experiência adquirida do seu antecessor, o 8008. O 8080 foi o primeiro microprocessador potente de 8 bits que surgiu no mercado e, até hoje, devido ao seu grande sucesso, ocupa um lugar de destaque entre vários outros microprocessadores de 8 bits que surgiram após ele. Podemos citar duas razões principais às quais se deve o grande êxito do 8080: a primeira razão é por ser compatível com o microprocessador - 8008, ou seja, o 8080 possui os mesmos registros e o mesmo repertório de instruções que o seu antecessor, além, é claro, de outros registros e outras instruções adicionais. A segunda razão se deve ao fato da Intel (fabricante do 8080) ter se dedicado muito no desenvolvimento de um suporte adequado, tanto no que se refere a circuitos integrados complementares para o microprocessador como em relação ao software, aplicado ao mesmo. Esses dois fatores, aliados a uma série de outros fatores, fizeram do 8080 o microprocessador mais popular da atualidade.

Com o objetivo de facilitar o estudo do 8080, iremos, inicialmente, fazer uma descrição da pinagem do mesmo.

DESCRIÇÃO DA PINAGEM DO MICROPROCESSADOR 8080

Construído a partir da tecnologia N-MOS a qual lhe confere maior velocidade, (2 microssegundos por ciclo de instrução) o microprocessador 8080 é uma unidade central de processamento contida em um único circuito integrado de 40 pinos. A figura 1 nos mostra a configuração de seus pinos.



Pinagem do microprocessador 8080.

FIGURA 01

Vejamos então uma descrição da pinagem 8080:

V_{SS}, V_{BB}, V_{CC} e V_{DD}:

Correspondem, respectivamente, aos pinos 2, 11, 20 e 28. Esses são os pinos de alimentação do 8080, o qual requer três níveis de tensão de alimentação e ainda o sinal de terra.

V_{SS} = referência 0 volts (terra)

V_{BB} = -5 ± 5% volts

V_{CC} = +5 ± 5% volts

V_{DD} = +12 ± 5% volts

$\emptyset_1$ e $\emptyset_2$:

Correspondem aos pinos 21 e 15 respectivamente. Trata-se de duas entradas para o sinal de "clock" de duas fases ($\emptyset_1$ e $\emptyset_2$) geradas externamente. A tensão de +12 V (V_{DD}) é necessária para a alimentação desse "clock" externo. Muito embora essa tensão não possa ser gerada por componentes TTL normais, sua obtenção não é problema, devido à existência do gerador de "clock" 8224.

A_0 até A_{15}:

São dezesseis pinos, os quais constituem o barramento de endereço (ou barramento de direções). Em outras palavras, podemos dizer que se trata de dezesseis saídas three-states (três estados), as quais podem assumir, portanto, os níveis lógicos "0" ou "1" ou ainda "alta impedância". Estas saídas são utilizadas para o endereçamento de memórias ou de dispositivos de entrada/saída (E/S).

Devido ao fato do 8080 ser um microprocessador de 8 bits, o máximo de memória que podemos endereçar, através dessas dezesseis saídas é de 64 K (ou seja, $2^{16} = 65536 \cong 64$ K) e o máximo de dispositivos de entrada/saída (E/S) endereçáveis é de 512 ($2^{n+1} = 2^{8+1} = 2^9 = 512$)

A_0 é o bit de endereço menos significativo e A_{15} é o endereço mais significativo.

D_0 até D_7:

Esses oito pinos (D_0 a D_7) formam o barramento de dados

Trata-se pois de pinos de entrada/saída three-states. O barramento de dados forma uma via de comunicação bidirecional entre o microprocessador e as memórias e entre os dispositivos de entrada/saída, para a transferência de dados ou instruções. Esse barramento de dados também é utilizado durante o primeiro ciclo de clock de cada ciclo de máquina, para gerar uma palavra de controle a qual deverá ser interpretada pelo controlador do sistema que deverá então produzir os sinais de controle.

SYNC:

Trata-se da saída do sinal de sincronização, o qual serve para indicar aos circuitos externos o início de cada ciclo de máquina.

WAIT:

O sinal emitido por este pino de saída indica que o microprocessador está em estado de espera, no aguardo de um comando externo. O sinal WAIT tem como finalidade levar ao conhecimento dos circuitos externos quando a unidade central de processamento terminou uma determinada "tarefa" e está disponível para uma nova "tarefa".

DBIN:

O sinal emitido por este pino (pino 17) serve para indicar aos circuitos externos que o barramento de dados está no modo entrada.

HOLD:

Correspondente ao pino 13, este sinal de entrada faz com que a unidade central de processamento fique no estado de retenção, isto é, tão logo o 8080 terminar seu atual ciclo de máquina, o barramento de dados e barramento de endereço serão levados para o estado de alta impedância. Com isto, a saída HLDA (pino 21) enviará um sinal o qual significa que a partir deste sinal, o 8080 admite o estado de retenção e permite, desta forma, que um determinado dispositivo externo assuma o controle do barramento de dados e do barramento de endereço.

READY:

A entrada READY corresponde ao pino 23 do 8080. Quando

existir um sinal no pino READY, o 8080 irá reconhecê-lo e "entenderá" que um dado proveniente de uma memória ou de um dispositivo de entrada/saída (E/S) se encontra à disposição no barramento de dados, para ser lido pelo microprocessador 8080.

Quando este microprocessador enviar um certo endereço pelo barramento de endereço e não receber um sinal READY, o microprocessador (no caso o 8080) entrará em estado de WAIT (espera) e permanecerá neste estado enquanto a entrada READY estiver em um nível lógico baixo. Tão logo a entrada READY receba um sinal, esta irá para um nível lógico alto e o 8080 poderá, então, ler este dado. Normalmente este sinal é utilizado para sincronizar a unidade central de processamento (UCP) com memórias ou com dispositivos de entrada/saída (dispositivos E/S) lentos.

$\overline{WR}$:

Correspondente ao pino 18, este sinal de saída é utilizado para o controle da escrita, tanto em memórias como em dispositivos de entrada/saída. Enquanto o sinal $\overline{WR}$ estiver no nível lógico "0", a informação contida no barramento de dados permanecerá estável.

HLDA:

O sinal emitido por esta saída (pino 21) indica, para os dispositivos externos, que o 8080 se encontra no estado de retenção. Este sinal nada mais é do que uma resposta ao sinal recebido pelo pino HOLD e indica que o barramento de dados e o barramento de endereço encontram-se em estado de alta impedância.

INT:

Correspondente ao pino 14 (entrada) o nome INT significa interrupção. Deste modo, um sinal no pino INT faz com que ocorra uma interrupção na unidade central de processamento, tão logo a instrução atual, que está sendo processada, termine. Caso a UCP se encontre no estado HOLD ou no estado INTE, ou seja, em nível lógico baixo, a interrupção não será aceita.

INTE:

Corresponde ao pino 16 (saída). Este sinal indica que a UCP está habilitada a receber uma interrupção. Através do software, podemos habilitar ou desabilitar a interrupção através de instruções apropriadas, porém a interrupção será desabilitada sempre que a unidade central de processamento aceitar uma interrupção ou receber um sinal de RESET.

RESET:

Um sinal no pino RESET (pino 12) faz com que um determinado programa recomece a partir da posição zero, de memória. Por outro lado, este sinal faz com que a unidade central de processamento saia do estado HOLD desabilitando a possibilidade de interrupção. As saídas INTE e HLDA irão para um nível lógico baixo. Observe que o sinal de RESET não altera os registradores, o acumulador, os flags e o stack pointer.

Uma vez feita a descrição da pinagem do microprocessador 8080, cabe ressaltar o fato de que somente os sinais de RESET e INT serão manipulados pelo operador do sistema. As demais variáveis de controle do 8080 serão utilizadas para controlar outros circuitos que fazem parte do sistema do 8080. Vejamos agora a arquitetura do 8080.

ARQUITETURA DO MICROPROCESSADOR 8080

De uma forma bem simples, podemos dividir o microprocessador 8080 em três blocos básicos, ou seja:

1) Seção de lógica e aritmética;

2) Rede de registradores;

3) Seção de controle.

A figura 2 nos mostra o diagrama de blocos do 8080, destacando-se os três blocos básicos.

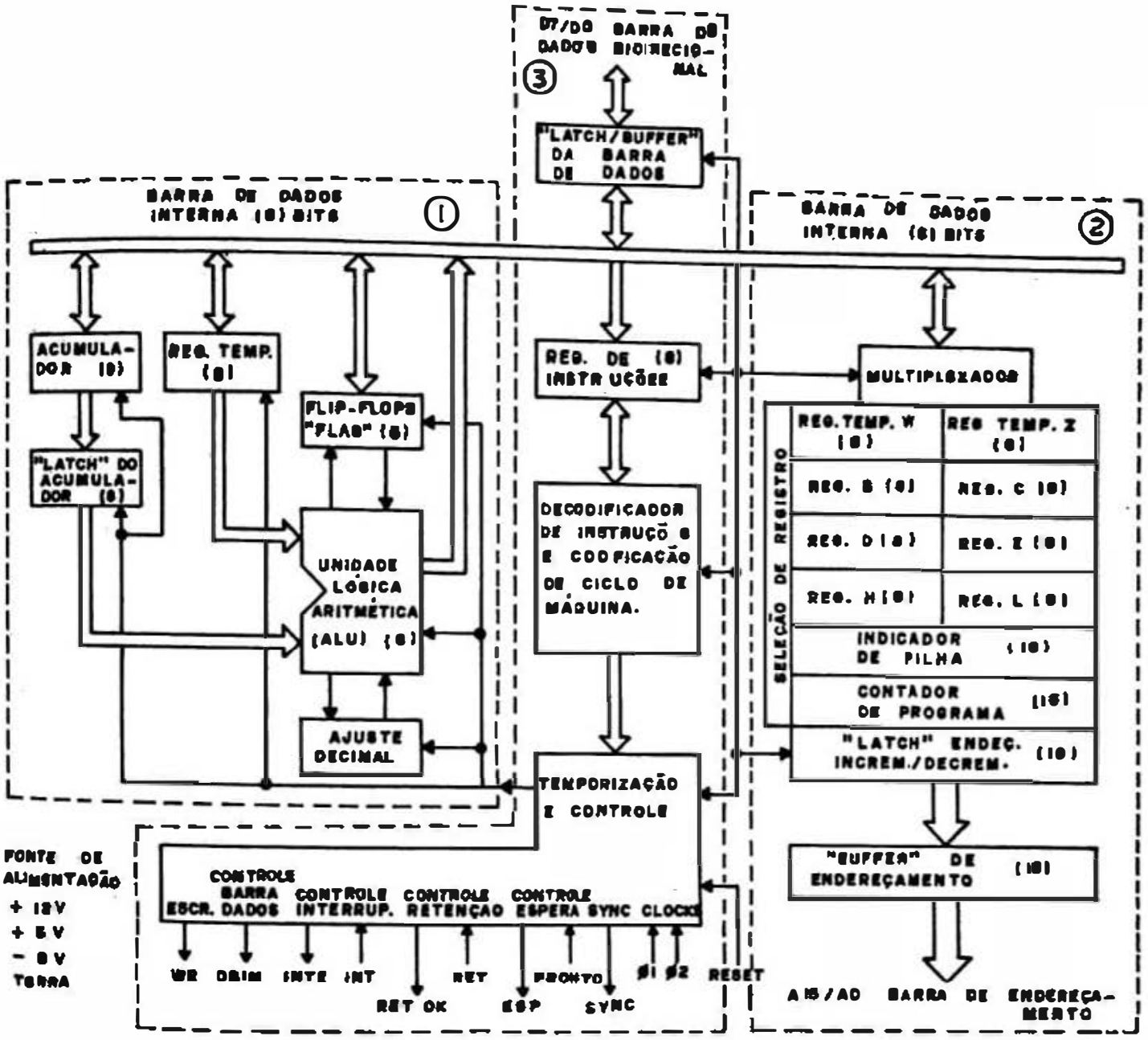


FIGURA 02

Estes três blocos básicos constituem a unidade central de processamento, a qual, juntamente com as memórias e os dispositivos de entrada/saída, formam o sistema básico de um "microcomputador".

Ao estudo da arquitetura e dos componentes físicos de um sistema dá-se o nome de HARDWARE. Podemos, de uma maneira mais simples, dizer que o hardware do microprocessador 8080 é o conjunto das portas lógicas responsáveis por toda operação do 8080.

Em seguida, apresentaremos algumas características dos três blocos básicos.

1) Seção de lógica e aritmética

A seção de lógica e aritmética é constituída pelos seguintes blocos:

a) Unidade lógica e aritmética (ULA);
b) Flags;
c) Registrador temporário;
d) Latch;
e) Acumulador;
f) Registrador de ajuste decimal.

A figura 3 nos mostra, com maiores detalhes, a seção de lógica e aritmética.

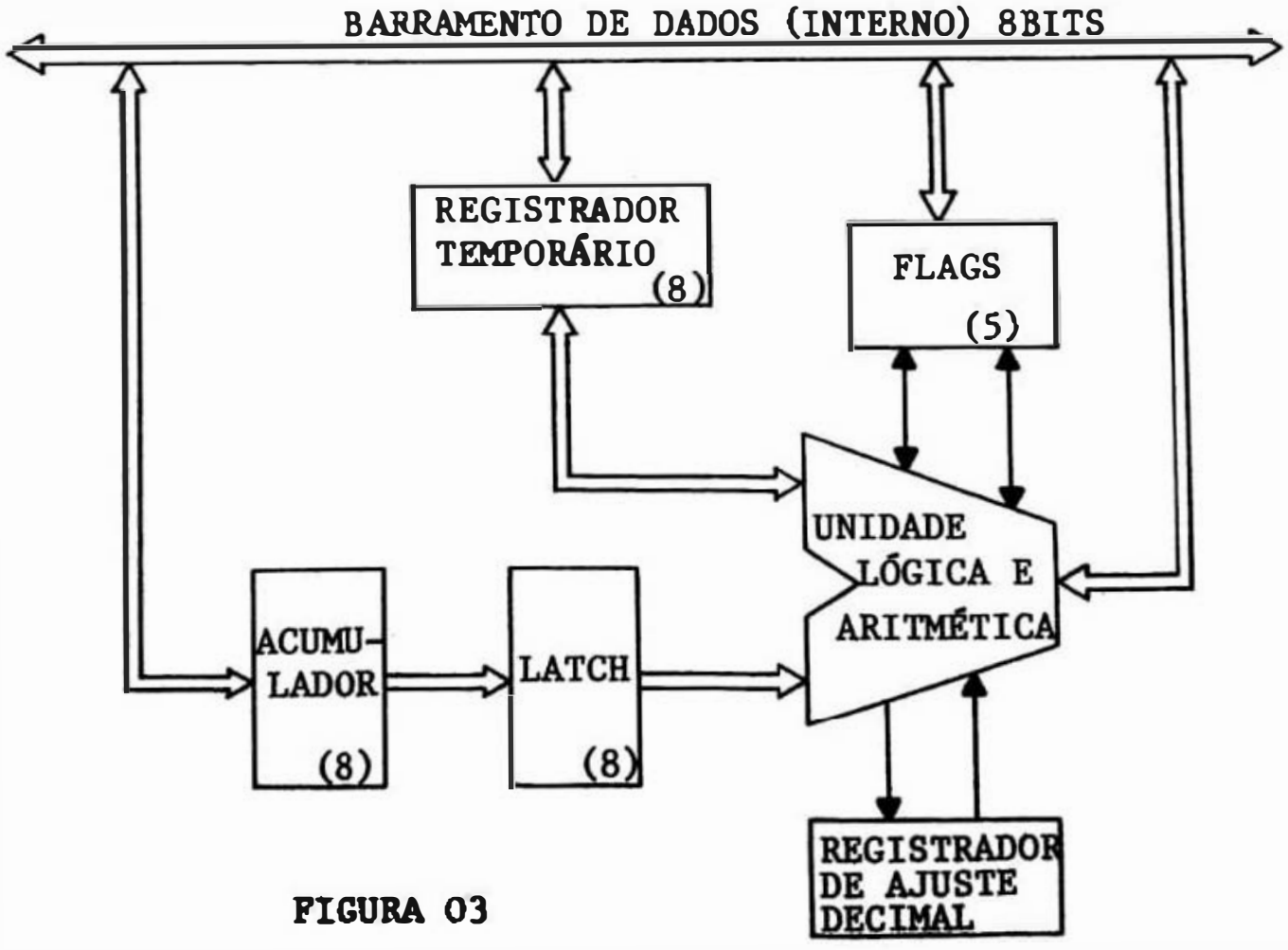


FIGURA 03

Podemos observar, através da figura 2, que a unidade central de processamento possui um barramento de dados interno, através do qual passam os dados para serem utilizados pela seção lógica e aritmética. Os dados que circulam neste barramento irão se alojar no registrador temporário ou no acumulador (que também é um registrador).

Nestes dois registradores se encontram os dois operandos da operação lógica ou aritmética. O resultado da operação lógica ou aritmética será sempre armazenado no acumulador. Deste modo, ao se somarem dois números por uma instrução de soma, primeiramente, um desses números deve ser armazenado no acumulador enquanto o outro deve ser armazenado num dos registradores da rede de registrador. Ao receber a instrução de soma, a unidade central de processamento transferirá o número armazenado no acumulador para o latch do acumulador. Em seguida irá buscar o outro número, que está armazenado num dos registradores da rede de registradores, e irá transferi-lo para o registrador temporário. A operação soma desses dois números será então realizada na unidade lógica aritmética (ULA), sendo o resultado final transferido para o acumulador.

O flags é um bloco formado por flip-flops, os quais serão posicionados conforme o resultado da operação. O microprocessador 8080 possui os seguintes flags:

Z - Zero: Este flag será levado para um nível lógico alto(1), ou seja, será posicionado toda vez que o resultado de uma operação for zero. Caso o resultado da operação for diferente de zero, será levado para um nível lógico baixo (0), ou seja, será posicionado em "0".

S - Sinal: Este flag será posicionado com o nível lógico alto "1" toda vez que o resultado de uma operação for negativo e com nível lógico baixo "0" toda vez que o resultado de uma operação for positivo.

P - Paridade: Toda vez que o resultado de uma operação apresentar um número par de bits com o nível lógico "1", este flag será então posicionado com o nível lógico alto "1" e quando tivermos um número ímpar este flag será posicionado com o nível lógico baixo "0".

C - Carry: Este flag só será posicionado com o nível lógico alto "1", quando o resultado de uma operação apresentar um "vai um" no bit mais significativo. Caso contrário,será posicionado com o nível lógico baixo "0".

AC - Auxiliar Carry: Quando o resultado de uma operação apresentar um "vai um" no seu terceiro bit, este flag será então posicionado com o nível lógico alto "1". Caso contrário, será posicionado com o nível lógico baixo "0".

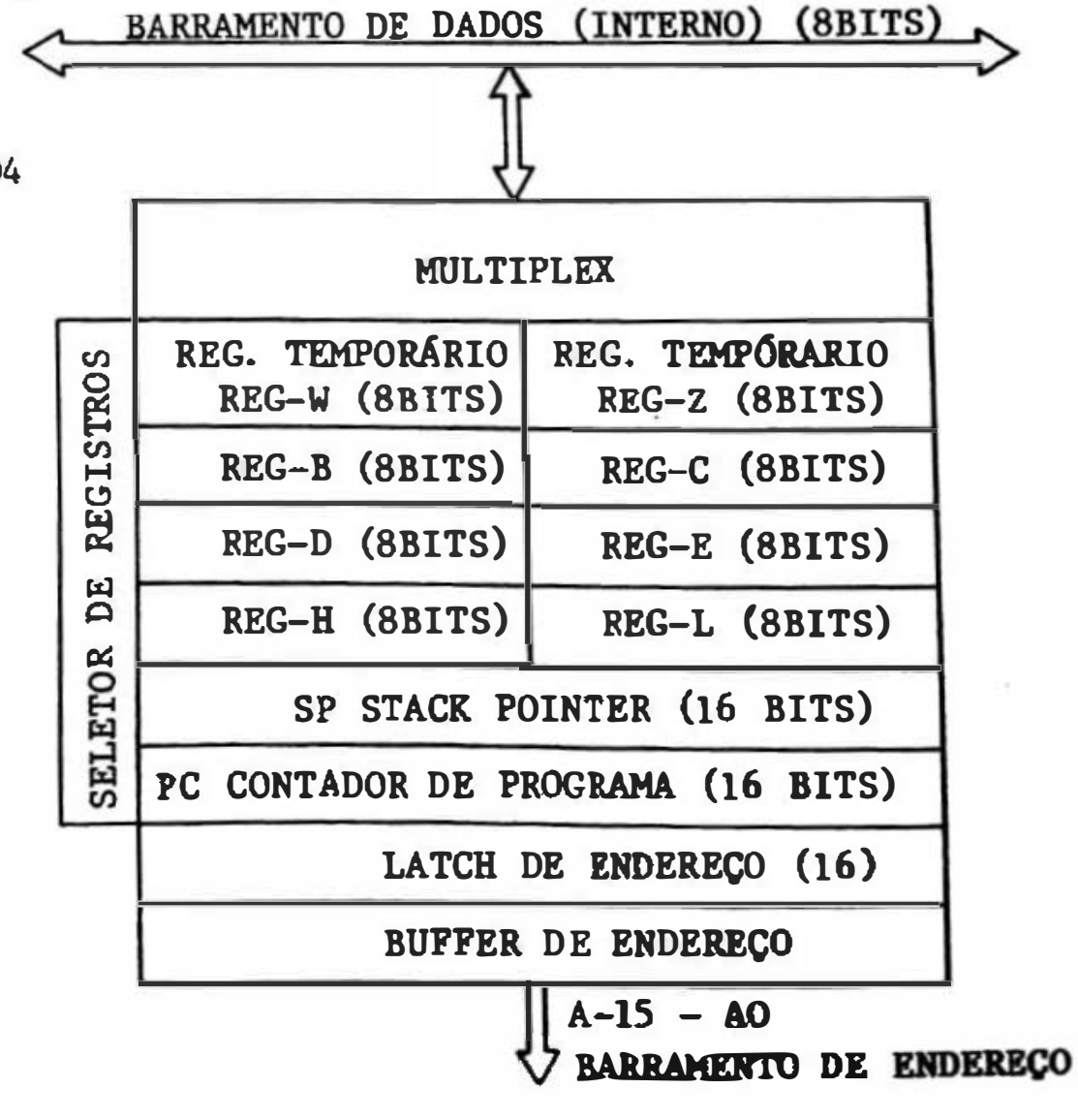
É possível, através de instruções especiais em um programa, testar esses flags para depois analisarmos o resultado da operação lógica ou aritmética. Apenas o flag auxiliar carry (AC) não pode ser analisado através deste método, pois ele é usado somente pelo registrador de ajuste decimal.

A função do registrador de ajuste decimal é auxiliar a seção de lógica e aritmética na transformação de um byte hexadecimal em dois dígitos decimais.

2) Rede de Registradores:

Essa rede de registradores internos do microprocessador 8080 são registradores utilizados para armazenar informações para posterior utilização. Para o 8080 esses registradores são em número de seis: B, C, D, E, H, L, todos de 8 bits para uso geral, com acesso direto pelo programa. A figura 4 nos mostra os registradores do 8080.

FIGURA 04



Estes seis registradores podem ser endereçados separadamente ou em pares, formando, desta forma, registradores de 16 bits. Normalmente, esses registradores possuem acesso direto à memória e também à seção de lógica e aritmética, permitindo assim, a transferência direta de seus conteúdos para uma destas áreas ou vice-versa. É nestes registradores que se armazena um dos operandos de uma operação lógica ou aritmética.

Os outros registradores que compõem a rede de registradores são os seguintes:

W, Z - Trata-se de dois registradores de 8 bits, para uso geral, porém sem acesso pelo programa.

SP - Trata-se do STACK POINTER ("indicador de pilha" em português), um registrador de 16 bits. O microprocessador tem a possibilidade de utilizar uma parte de uma memória externa como uma pilha, onde podemos armazenar o conteúdo dos demais registradores. Este registrador (o SP) possui o endereço da posição de memória onde está armazenado o último dado enviado, ou transferido, para esta pilha.

PC - Conhecido por "Contador de Programa", este registrador de 16 bits possui o endereço de memória que contém a próxima instrução a ser executada.

3) Seção de Controle:

A figura 5 nos mostra um diagrama de blocos da seção de controle.

É esta seção que fornece as variáveis de controle para o funcionamento do sistema. Podemos distinguir dois grupos de variáveis de controle:

a) Variáveis de controle internas - pertencem a este grupo de variáveis os sinais que acionam os latchs, buffers, registradores, multiplexadores e demais circuitos internos da unidade central de processamento (UCP).

b) Variáveis de controle externas - são os sinais WR, DBIN, INTE, INT, HLDA, HOLD, WAIT, READY, $\emptyset_1$ e $\emptyset_2$, RESET, já comentados anteriormente.

Para a geração destas variáveis de controle, é preciso que uma determinada instrução entre, através do barramento de dados externo, na unidade central de processamento. Esta ins-

trução é então armazenada no registrador de instrução e, por intermédio de um decodificador, é então decodificada para a unidade de controle a qual irá gerar as variáveis de controle. Concluindo, podemos dizer que cada instrução irá dar origem a uma configuração diferente de variáveis de controle, variáveis estas que irão permitir a execução de uma determinada instrução.

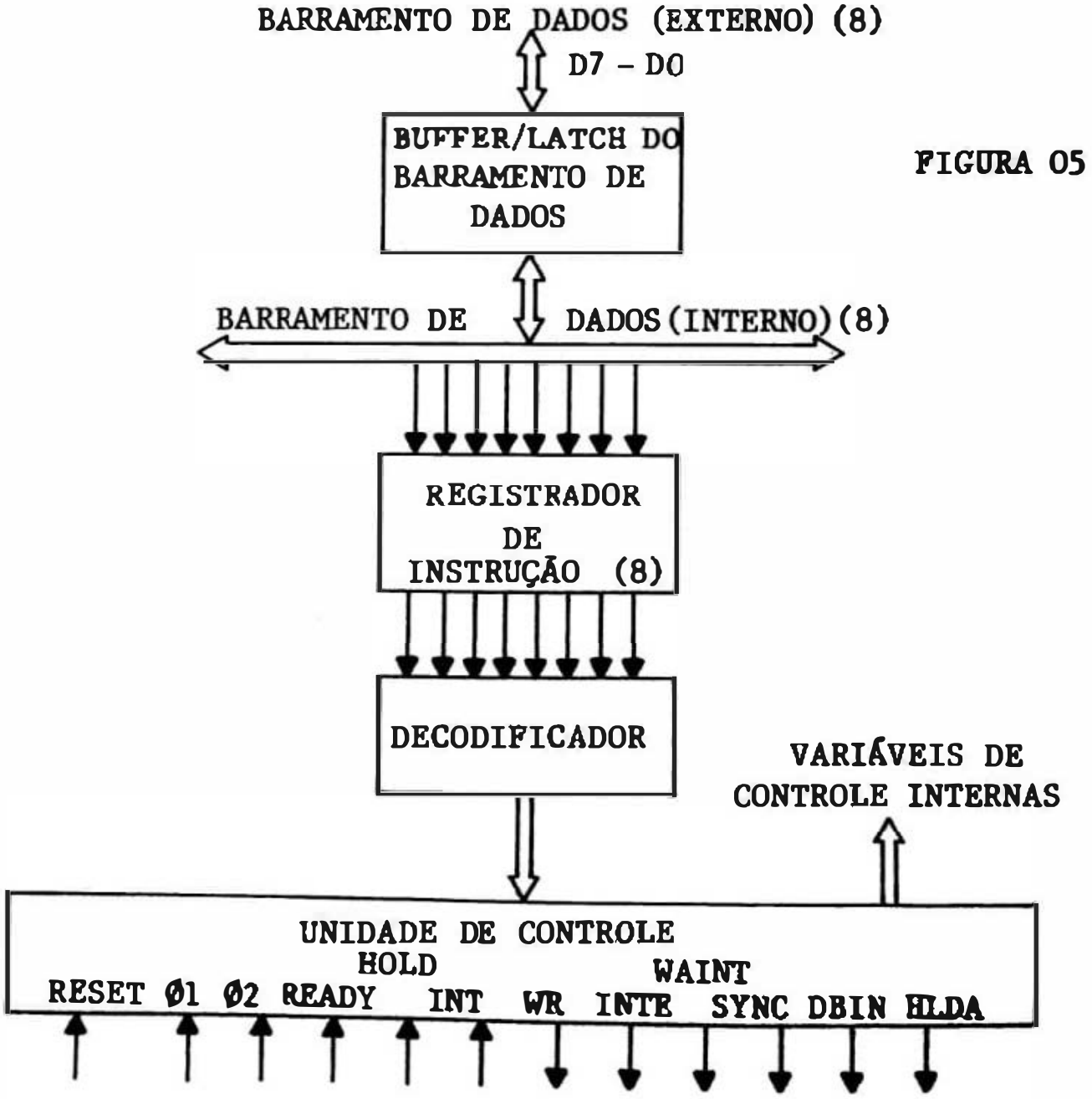


FIGURA 05

A figura 6 nos mostra uma microfotografia do microprocessador 8080.

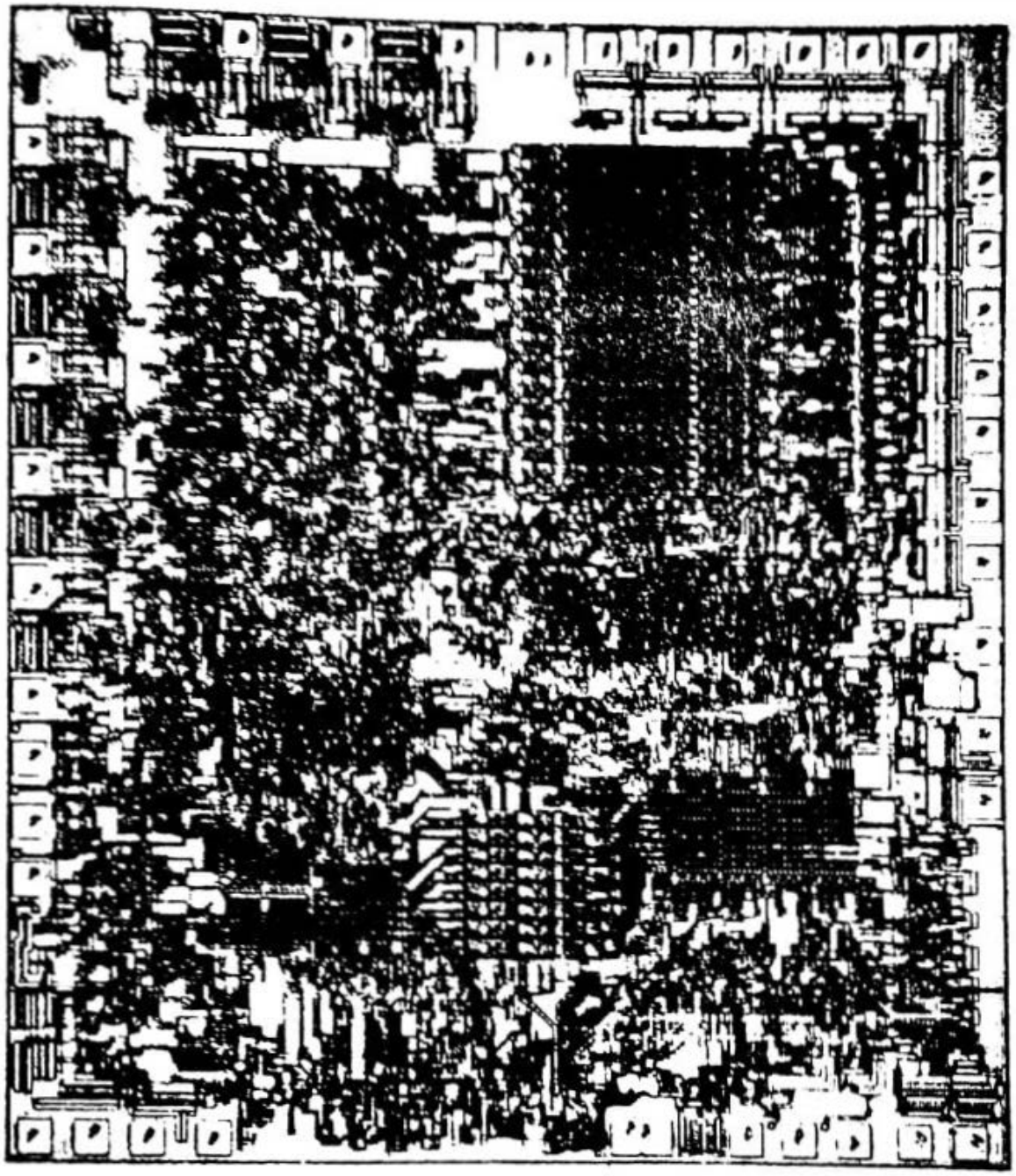


FIGURA 06

Base de tempo:

As instruções do 8080 contêm de um a três bytes. Cada instrução requer de 1 a 5 ciclos de máquina ou memória para sua localização e execução. Os ciclos de máquina são denominados M_1, M_2, M_3, M_4 e M_5. Sabe-se também que cada ciclo de máquina precisa de 3 a 5 estados (T_1, T_2, T_3, T_4 e T_5) para que possa realizar-se. Cada estado tem uma duração de um período de clock de 0,5 microssegundos. Fora estes cinco estados, existem outros três: Wait, Hold e Halt, todos com um número indefinido de períodos, pois são controlados por sinais externos. A figura 7 nos mostra o ciclo básico de uma instrução do microprocessador 8080.

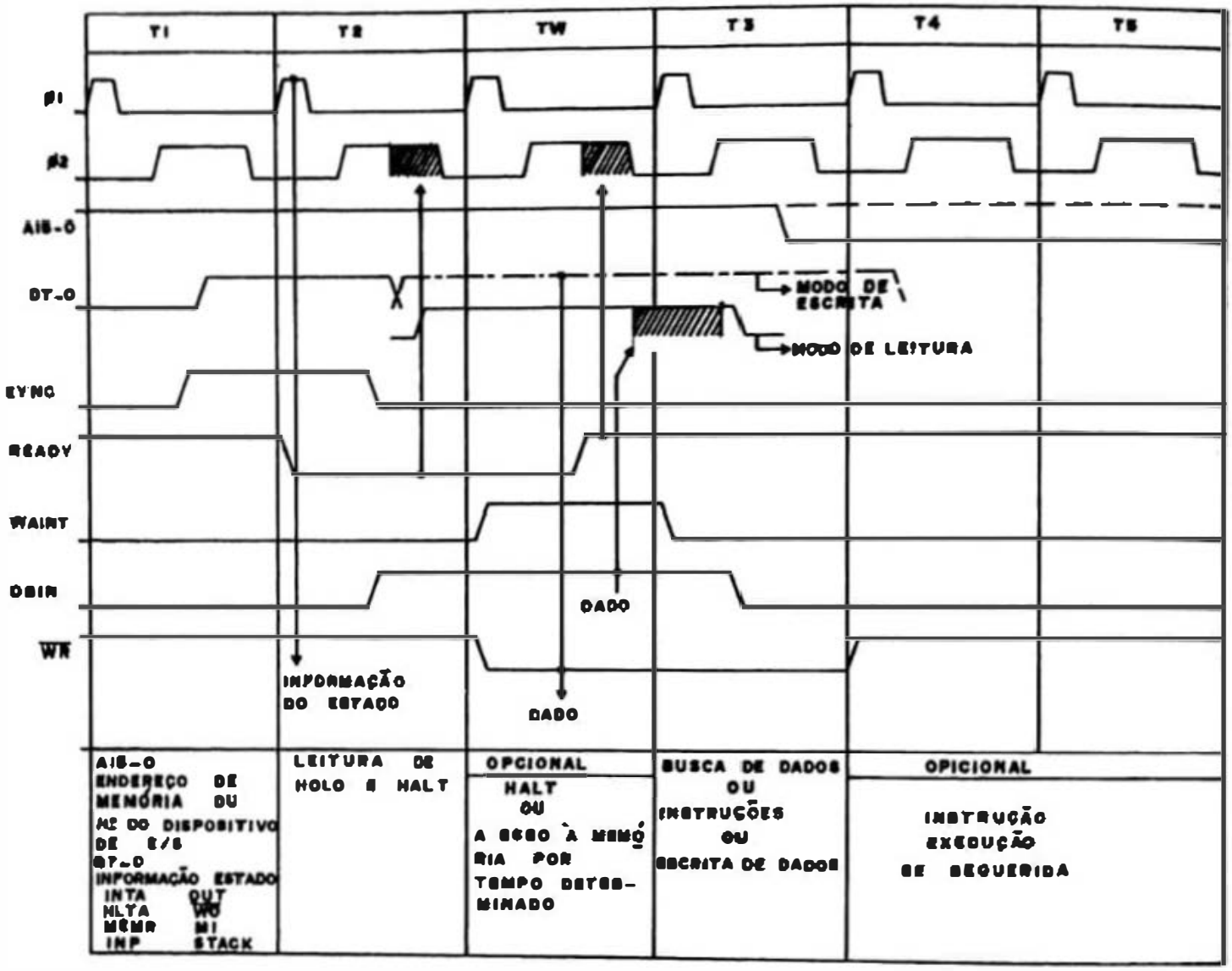


FIGURA 07

O ciclo de máquina M_1 localiza a instrução e é determinado por um código de operação com uma duração entre 2 e 2,5 microssegundos.

Em cada ciclo de instrução existem cinco (5) fases elementares (T_1 a T_5) as quais, geralmente, se desenvolvem da seguinte forma: Na fase T_1, o conteúdo do contador de programa é depositado no barramento de direções, sendo que os bits de "status" (D_0 a D_7) é que indicam o ciclo que se realiza. Em T_2 é feita uma comparação dos sinais de "status", ready, hold e halt. Se houver um sinal ready então passa-se para a fase seguinte (T_3) e não se passa a um estado denominado T_w, com o que se sincroniza a UCP a qualquer memória de qualquer tempo de acesso. Na fase T_3, os dados que estavam no barramento de direções em T_1 são transladados para o barramento de dados e

em seguida ao registrador de instruções. Se a operação for de escrita ao invés de leitura sobre a memória, haverá alguma unidade que introduzirá na memória os dados que devem ser gravados.

Na fase T_4 realiza-se a decodificação e se completa a instrução, então, passa-se para T_1 (outro ciclo de máquina). Caso contrário inicia-se o ciclo de máquina M_2.

Para finalizar, no último ciclo da instrução se confirma o estado da entrada de Interrupção (INT.) a qual, se estiver ativada, passa para um ciclo especial M_1, sem incrementação do contador de programa.

Com relação ao software do 8080, sabe-se que este microprocessador dispõe de 78 instruções, sendo que cada uma delas pode ter 1, 2 ou até 3 bytes de 8 bits. Estas instruções se dividem em cinco grandes grupos:

1º) Grupo de transferência de dados: instruções utilizadas para a movimentação de dados entre registros ou entre memórias e registros.

2º) Grupo de instruções aritméticas: adição, subtração, incremento e decremento de dados na memória ou em registros.

3º) Grupo de instruções lógicas: E, OU, OU-Exclusivo, Comparação ou complementação de dados em registros ou em memória.

4º) Grupo de saltos: instruções condicionais ou não, de salto, de sub-rotinas e de retorno.

5º) Grupo de controle de stack, E/S e máquina: Instruções de Entrada/Saída (E/S), controle de stacks e flags.

Um programa escrito para o 8008 pode, sem muita dificuldade, ser traduzido para o 8080. A microcodificação interna do 8080 exigiu um re-arranjo dos códigos de operação, de modo que os códigos binários de máquina de várias instruções do 8008 tiveram que ser mudados.

O conjunto de instruções do microprocessador 8080 será fornecido mais adiante, quando do estudo do 8085, uma vez que se trata, praticamente, das mesmas instruções (com exceção de apenas duas delas).

A figura 8 nos mostra o diagrama de blocos, contendo os CIs 8080, 8224 e 8228. Esses três circuitos juntos formam o mínimo indispensável para o funcionamento de uma unidade central de processamento com o microprocessador 8080.

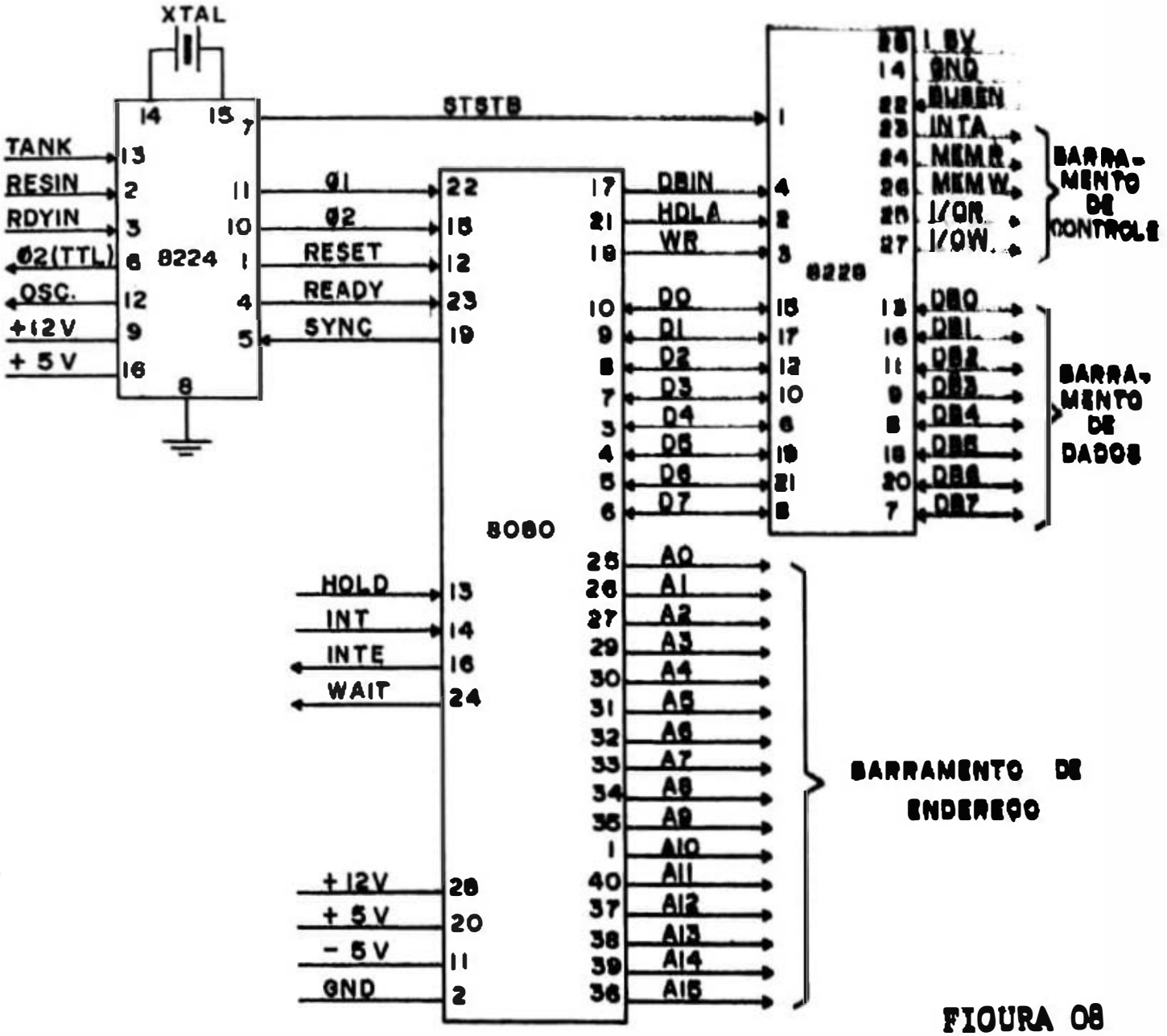


FIGURA 08

Como sabemos, todo microprocessador é um dispositivo dinâmico e, devido a isto, necessita de um sinal de clock como referência, o qual irá sincronizar o seu funcionamento. A cada variação deste sinal, obteremos mudanças das variáveis de estado do microprocessador, mudanças essas que irão causar a execução de suas funções.

Para gerar o sinal de clock, geralmente, usa-se um oscilador a cristal de quartz que, quando devidamente polarisado oscila numa freqüência estável e bem determinada. Para a polarização do cristal e também sua adequação de sinal com o microprocessador 8080, foi desenvolvido um CI especial, o 8224, cuja pinagem é mostrada na figura 9.

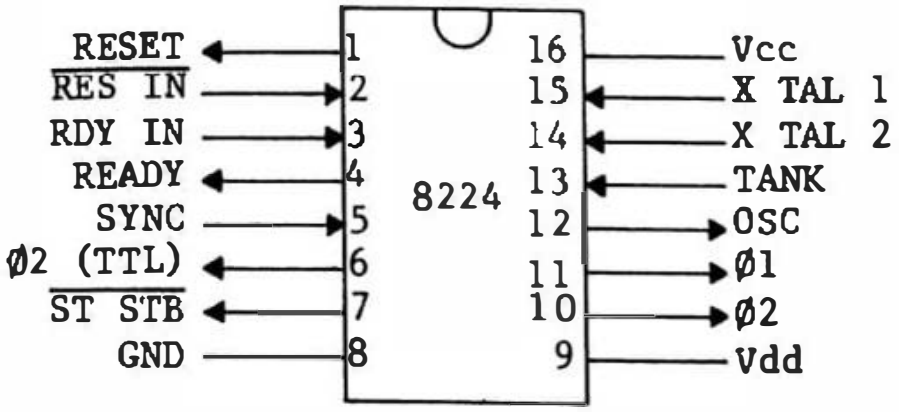
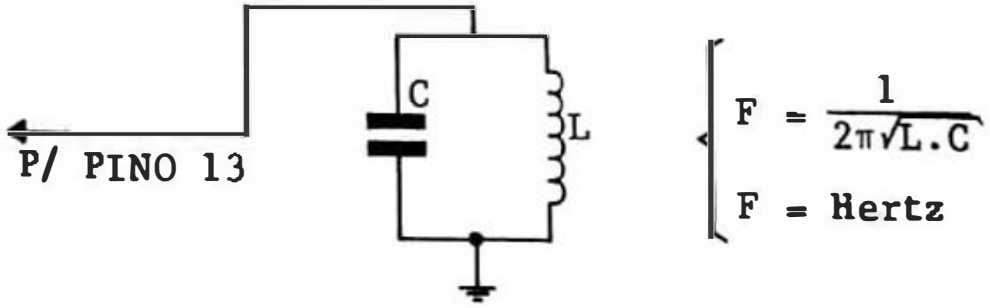


FIGURA 09

Internamente, o 8224 é formado por um circuito oscilador, por flip-flops tipo D e por um circuito divisor de freqüência.

No pino 13 (TANK) vai conectado um circuito "tank" igual ao apresentado na figura 10.

FIGURA 10



Com relação à pinagem do 8224 (veja figura 9), mais precisamente com relação ao pino 7 ($\overline{\text{ST STB}}$), cabe dizer que se trata de uma saída cuja função é diferenciar a informação contida no barramento de dados do 8080, pois, neste barramento, podemos ter dados ou informações de controle. Esse diferenciamento é feito da seguinte forma: como sabemos, no início de cada ciclo de máquina, as informações contidas neste barramento são informações de controle. Com isto, nesse mesmo instante, o sinal SYNC é gerado pelo 8080 e recebido pelo 8224, o qual irá então gerar um sinal $\overline{\text{ST STB}}$, permitindo ao controlador do sistema, o CI 8228, identificar que, neste instante, as informações contidas no barramento de dados são informações de controle.

A figura 11 nos mostra o diagrama de blocos do CI 8224.

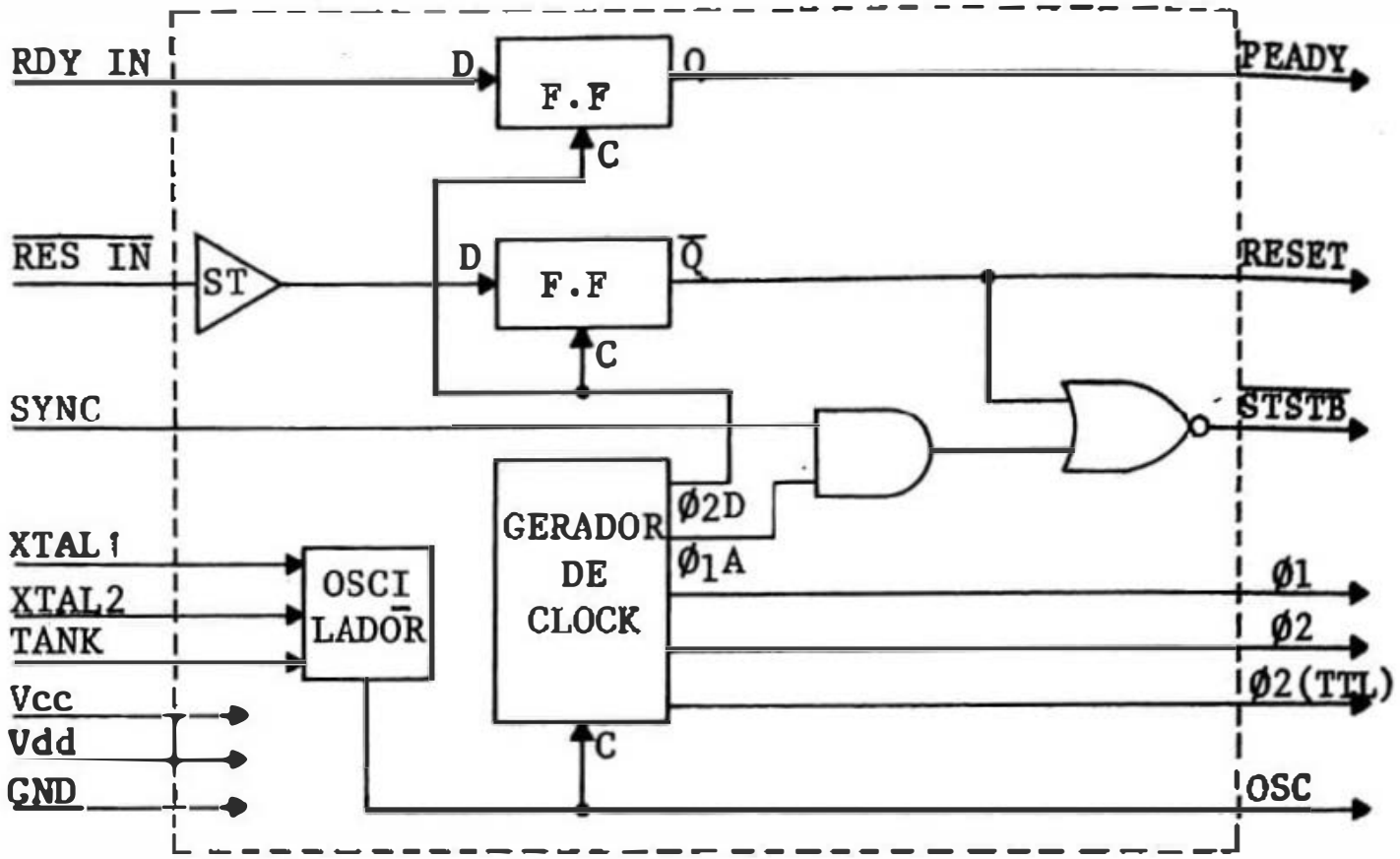


FIGURA 11

Um outro CI muito importante e que se faz necessário para o funcionamento de uma unidade central de processamento que utiliza o microprocessador 8080, é o CI 8228 (veja figura 12). O 8228 é um circuito integrado, cuja função é controlar o sistema de microprocessador de multiplexando, o barramento de dados do 8080, de modo a dar origem ao barramento de dados e ao barramento de controle do sistema. A figura 12 nos mostra a pinagem do 8228.

O 8228 faz também a ligação da unidade central de processamento com as memórias e os dispositivos de entrada e saída.

Com relação à arquitetura interna, o 8228 é constituído de um latch para as variáveis de estados, controlado pelo sinal $\overline{ST\ STB}$; um driver bidirecional para o barramento de dados e um decodificador para a geração das variáveis de controle. A figura 13 nos mostra um diagrama de blocos da arquitetura interna do 8228.

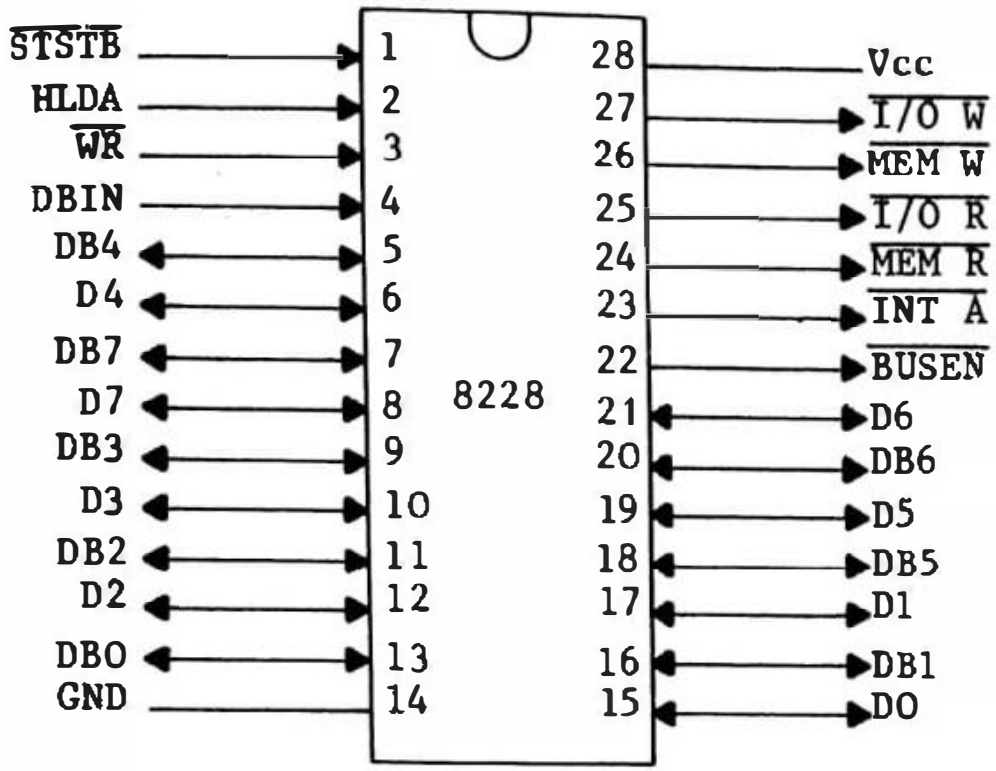


FIGURA 12

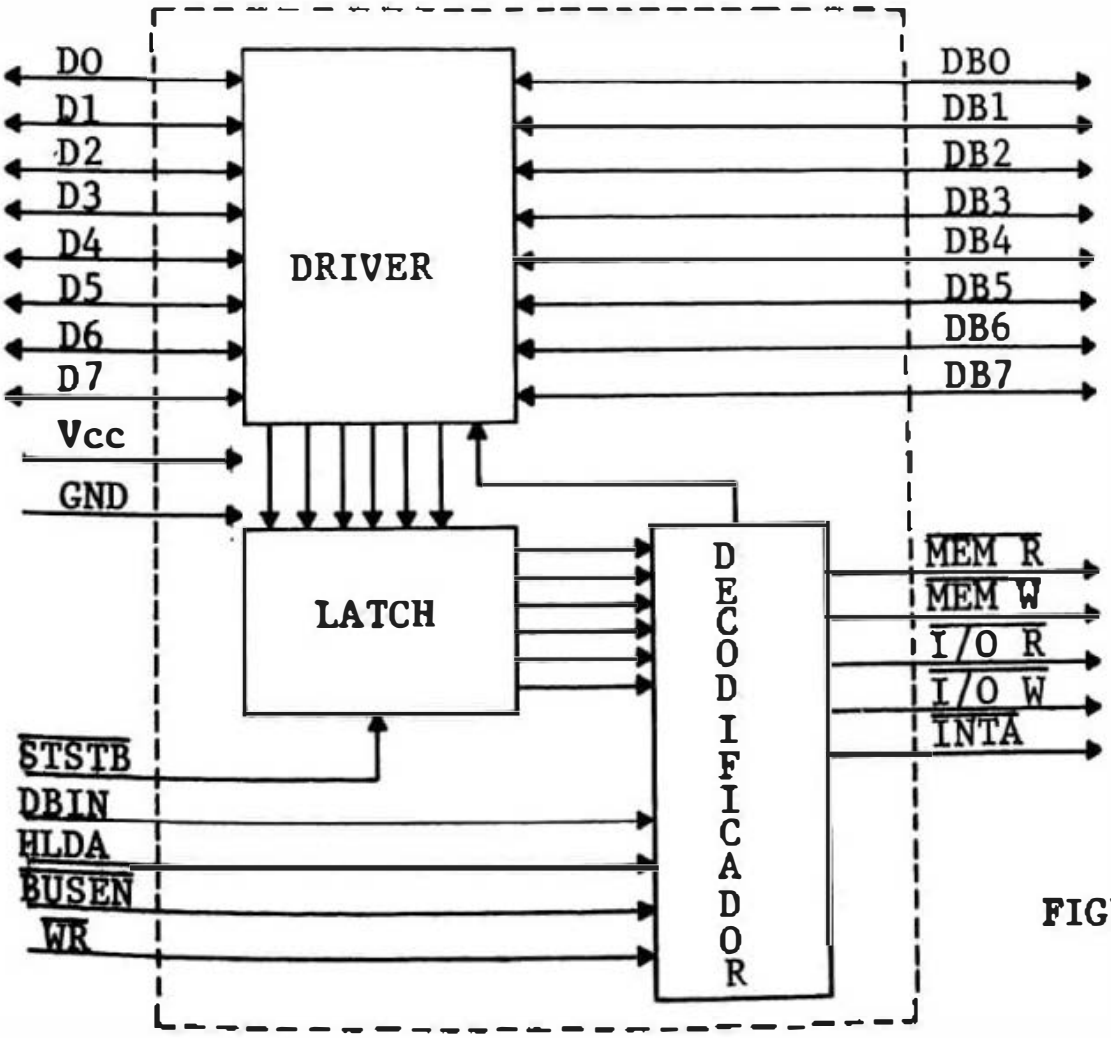


FIGURA 13

Com relação à pinagem do 8228, cabe ressaltar que as saídas $\overline{I/OW}$, $\overline{I/OR}$, $\overline{MEM\ W}$, $\overline{MEM\ R}$, $\overline{INTA}$ formam o barramento de controle, sendo que as quatro primeiras saídas são utilizadas para o controle de leitura e escrita em memórias ou em dispositivos de entrada/saída (E/S). A quinta saída ($\overline{INTA}$), por sua vez, é um sinal de controle utilizado para indicar, para os outros dispositivos, que uma interrupção foi aceita pela unidade central de processamento.

O 8080 pode reconhecer até oito níveis diferentes de interrupção, (instrução RST0 à RST7), controlados por CIs especiais (8214 ou 8259). Se desejarmos apenas um nível de interrupção, basta ligarmos a saída $\overline{INTA}$ (pino 23) à fonte de + 12 volts, através de uma resistência de 1 KΩ. Nesta condição, assim que houver uma solicitação de interrupção, automaticamente o controlador do sistema colocará no barramento de dados a instrução RST7.

Devido à grande semelhança que há entre os microprocessadores 8080 e 8085, apresentaremos, a seguir, o microprocessador 8085.

MICROPROCESSADOR 8085

O microprocessador 8085, fabricado pela Intel, possui toda a sua estrutura, bem como o seu conjunto de instruções, - 100% compatível com o seu antecessor, o 8080. O fato do 8085 ser totalmente compatível com o 8080 significa que se um determinado programa é processado por um sistema constituído pelo 8080, este mesmo programa também pode ser processado por um sistema como o 8085.

Podemos dizer, a grosso modo, que o 8085 é uma versão melhorada do microprocessador 8080, uma vez que o 8085, também de 8 bits, possui praticamente todas as características do 8080. As principais vantagens oferecidas pelo 8085 em relação ao 8080 são:

a) maior velocidade de processamento (o ciclo básico de instrução passou de 2 μs, que era o caso do 8080, para 1,3 μs);
b) maior simplicidade de uso;
c) mais níveis de interrupção;
d) uma só tensão de alimentação (5,0 volts), enquanto que o 8080 requeria três tensões de alimentação +5, -5 e +12 volts);

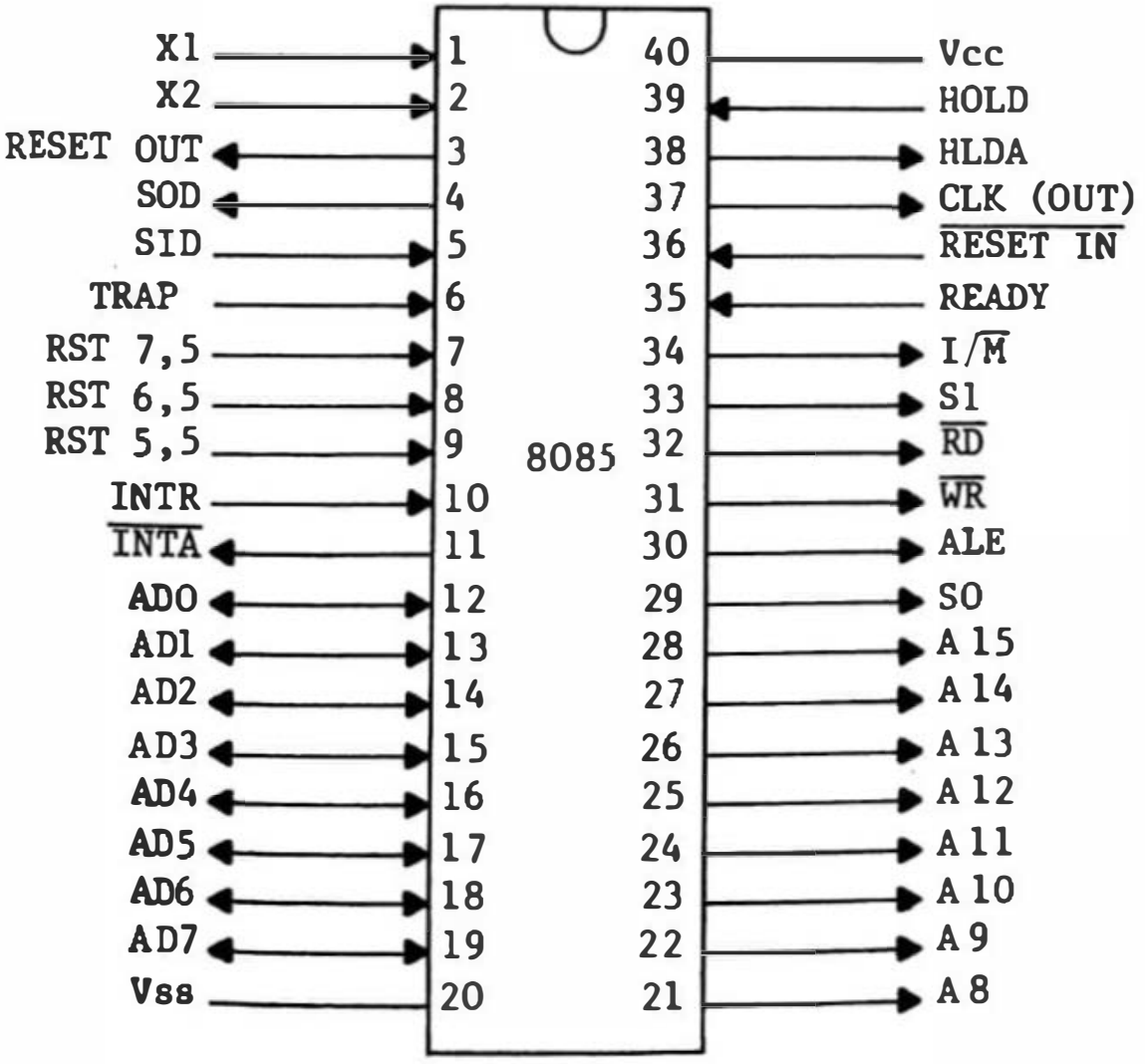
e) decodificador de estados incluído no chip, ou seja, no 8085 a função do controlador do sistema (8228) já está incluída no chip e o sinal de controle sai diretamente por determinados pinos da "pastilha".

Essas vantagens apresentadas pelo 8085 advêm do fato de o mesmo dispensar o uso de gerador de clock externo (8224) e também, como já dissemos, do controlador do sistema (8228), pois já estão agregados ao mesmo.

O microprocessador 8085 está encapsulado em uma "pastilha" de 40 pinos, porém, para tanto, foi necessário multiplexar o barramento de dados, ou seja, os pinos do barramento de dados foram também utilizados como parte do barramento de endereço. Deste modo, em um dado instante, nestes pinos, são fornecidas informações de endereço que podem ser agrupadas em um latch e, em um outro instante, são fornecidas informações de dados.

A figura 14 nos mostra a pinagem do microprocessador 8085.

FIGURA 14



Apresentamos, a seguir, uma breve descrição da pinagem do 8085:

V_{CC} , V_{SS}:

Pinos de alimentação do 8085.
V_{SS} = referência 0 volts.
V_{CC} = + 5 ± 5% volts.

X_1 , X_2:

Entradas de clock, podendo ser este sinal gerado diretamente entre estes dois pinos, através de um cristal de quartz ou de um circuito ressonante LC.

CLK:

Saída do sinal de clock, para uso geral, cuja freqüência é a metade da freqüência em X_1 e X_2.

ALE:

Sinal de sincronização, gerado no primeiro ciclo da instrução, utilizado para habilitar os latchs de endereço, de modo a reter a atual informação, contida nos pinos AD_7 a AD_0.

A_{15} - A_8:

Barramento de endereço: são os 8 bits mais significativos que formam as saídas three-states do barramento de endereço.

SID:

Entrada de dados serial.

SOD:

Saída de dados serial.

INTR:

Possui a mesma função que o pino INT do microprocessador 8080 (requisição de interrupção).

TRAP:

Este pino (requisição de interrupção) reconhece a in-

terrupção da mesma forma que os pinos INTR, RST 5,5, RST 6,5, RST 7,5, porém não pode ser desabilitado através do programa. Em ordem decrescente de nível de prioridade teremos: TRAP ; RST 7,5; RST 6,5; RST 5,5 e por fim o INTR (menor nível de prioridade).

RST 5,5; RST 6,5; RST 7,5:

Possuem a mesma característica do pino INTR, porém, quando algumas dessas interrupções ocorrerem, o contador de programa (PC) será então carregado com um endereço fixo, o qual dependerá apenas de quais destes pinos receberam o sinal de interrupção. Essas interrupções podem ser desabilitadas ou habilitadas através do programa.

$\overline{\text{INTA}}$:

Indicador de interrupção.

HOLD, HLDA, READY:

Possuem as mesmas funções dos pinos de mesma nomenclatura do microprocessador 8080.

RESET IN:

Entrada de reset.

RESET OUT:

Saída de reset (indica que o 8085 recebeu um sinal de reset).

$\overline{\text{WR}}$:

Controle de escrita.

$\overline{\text{RD}}$:

Controle de leitura. Este sinal indica que o dispositivo de E/S ou a memória selecionada será lida e que o barramento de dados está posicionado como entrada.

I/$\overline{\text{M}}$:

Dispositivo de entrada e saída de memória.

S_0 , S_1:

Status: esses pinos indicam o estado atual em que se encontra o microprocessador.

S_1	S_0	Estado
0	0	retenção
0	1	escrita
1	0	leitura
1	1	interrompido

ARQUITETURA DO MICROPROCESSADOR 8085:

Um diagrama de blocos, mostrando a arquitetura interna do microprocessador 8085, pode ser visto na figura 15. Como podemos observar, a arquitetura interna do 8085 é basicamente a mesma do 8080, constituída pela rede de registradores, seção de controle e seção de lógica e aritmética.

A rede de registradores é idêntica à do microprocessador 8080, constituída por seis registradores de 8 bits, para uso geral (B, C, D, E, H, L), os quais podem ser utilizados pelo programa aos pares, formando assim, três registradores de 16 bits para uso geral (BC, DE, HL), um registrador de 16 bits para o stack pointer (indicador de pilha) e outro indicador de 16 bits para armazenamento dos endereços das instruções do programa. Além disto, a rede de registradores possui ainda dois registradores temporário de 8 bits (W,Z), sem acesso por programa.

Quanto à seção lógica aritmética, esta é formada pelo registrador acumulador, um registrador para armazenamento temporário de operandos e a unidade que processará as operações lógicas aritméticas (ULA) dos dados contidos no registrador temporário e no acumulador. Esta seção também possui um registrador para ajuste decimal e um registrador com 5 flags, idênticos aos flags do microprocessador 8080.

Finalmente, temos a seção de controle a qual é constituída pelo registrador de instruções, por um controlador de interrupções que decodifica os sinais externos, examinando as prioridades e alterando o conteúdo do contador de programa e por um outro controlador cuja função é transmitir ou receber dados seriais.

O fabricante do microprocessador 8085 oferece uma série de produtos, tais como memórias e dispositivos E/S, os quais podem ser conectados diretamente ao 8085. Entre esses produtos podemos citar a memória de programa 8355 (ROM), a memória de dados 8155 (RAM) e memória EPROM 8755. Além destes produtos, o 8085 suporta toda a família de produtos do sistema 8080, o que o torna um microprocessador de grande versatilidade.

Para finalizar o estudo sobre os microprocessadores 8080 e 8085, apresentamos o conjunto de instruções destes microprocessadores. Com vistas a facilitar qualquer trabalho de consulta técnica, as instruções serão apresentadas em ordem alfabética:

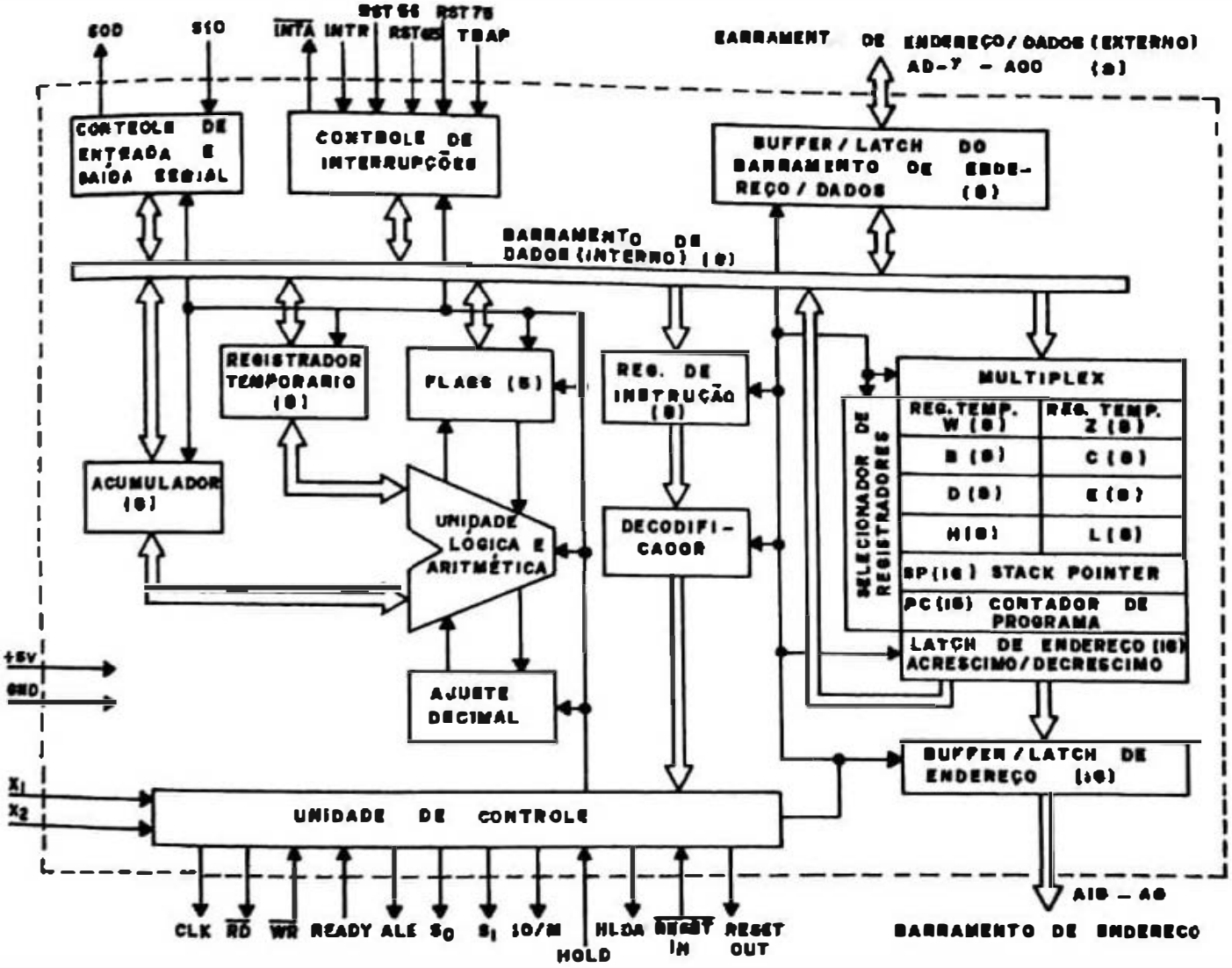


FIGURA 15

CONJUNTO DE INSTRUÇÕES

Nº			
01)	ACI	- ADD IMMEDIATE WITH CARRY	- Soma imediata com carry
02)	ADC	- ADD WITH CARRY	- Soma com carry
03)	ADD	- ADD	- Soma
04)	ADI	- ADD IMMEDIATE	- Soma imediata
05)	ANA	- LOGICAL AND WITH ACCUMULATOR	- "And" lógico com o acumulador

06) ANI - AND IMMEDIATE WITH ACCUMULATOR - "And" imediato com o acumulador
07) CALL - CALL - Chama sub-rotina
08) CC - CALL IF CARRY - Chama sub-rotina, se houver carry
09) CM - CALL IF MINUS - Chama sub-rotina, se negativo
10) CMA - COMPLEMENT ACCUMULATOR - Complementa o acumulador
11) CMC - COMPLEMENT CARRY - Complementa o flag carry
12) CMP - COMPARE WITH ACCUMULATOR - Compara com o acumulador
13) CNC - CALL IF NO CARRY - Chama sub-rotina, se não houver carry
14) CNZ - CALL IF NOT ZERO - Chama sub-rotina, se não houver zero
15) CP - CALL IF POSITIVE - Chama sub-rotina, se positivo
16) CPE - CALL IF PARITY EVEN - Chama sub-rotina, se paridade for par
17) CPI - COMPARE IMMEDIATE - Compara um dado imediato
18) CPO - CALL IF PARITY ODD - Chama sub-rotina, se paridade for ímpar
19) CZ - CALL IF ZERO - Chama sub-rotina, se zero
20) DAA - DECIMAL ADJUST ACCUMULATOR - Ajuste decimal do acumulador
21) DAD - DOUBLE REGISTER ADD - Soma o registrador duplo
22) DCR - DECREMENT - Decremento
23) DCX - DECREMENT REGISTER PAIR - Decremento par de registradores
24) DI - DISABLE INTERRUPTS - Desabilita interrupção
25) EI - ENABLE INTERRUPTS - Habilita interrupções
26) HLT - HALT - Halt (retenção)
27) IN - INPUT FROM PORT - Entrada através de porta
28) INR - INCREMENT - Incrementa
29) INX - INCREMENT REGISTER PAIR - Incrementar par de registradores
30) JC - JUMP IF CARRY - Salta, se houver carry

31) JM - JUMP IF MINUS - Salta, se negativo
32) JMP - JUMP - Salta
33) JNC - JUMP IF NO CARRY - Salta, se não houver carry
34) JNZ - JUMP IF NOT ZERO - Salta, se não houver zero
35) JP - JUMP IF POSITIVE - Salta, se positivo
36) JPE - JUMP IF PARITY EVEN - Salta, se a paridade for par
37) JPO - JUMP IF PARITY ODD - Salta, se a paridade for ímpar
38) JZ - JUMP ZERO - Salta, se for zero
39) LDA - LOAD ACCUMULATOR DIRECT - Carrega acumulador diretamente
40) LDAX - LOAD ACCUMULATOR INDIRECT - Carrega acumulador indiretamente
41) LHLD - LOAD H AND L DIRECT - Carrega HL diretamente
42) LXI - LOAD REGISTER PAIR IMMEDIATE - Carrega par de registradores com dado imediato
43) MOV - MOVE - Move
44) MVI - MOVE IMMEDIATE - Move imediato
45) NOP - NO OPERATION - Não operação
46) ORA - INCLUSIVE OR WITH ACCUMULATOR - Inclusive OR com acumulador
47) ORI - INCLUSIVE OR IMMEDIATE - Inclusive OR imediato
48) OUT - OUT PUT TO PORT - Saída por porta
49) PCHL - MOVE HL TO PROGRAM COUNTER - Move HL para o contador de programa
50) POP - POP - Retira do topo de stack
51) PUSH - PUSH - Coloca no topo do stack
52) RAL - ROTATE LEFT THROUGH CARRY - Girar para a esquerda, através do carry
53) RAR - ROTATE RIGHT THROUGH CARRY - Girar para a direita, através do carry
54) RC - RETURN IF CARRY - Retorno, se houver carry
55) RET - RETURN FROM SUBROUTINE - Retorno de sub-rotina
56) RIM - READ INTERRUPT MASK - Ler a máscara de interrupção
57) RLC - ROTATE ACCUMULATOR LEFT - Girar o acumulador para a esquerda
58) RM - RETURN IF MINUS - Retorno, se negativo
59) RNC - RETURN IF NO CARRY - Retorno, se não houver carry
60) RNZ - RETURN IF NOT ZERO - Retorno, se não zero

61) RP	- RETURN IF POSITIVE	- Retorno, se positivo
62) RPE	- RETURN IF PARITY EVEN	- Retorno, se a paridade for par
63) RPO	- RETURN IF PARITY ODD	- Retorno, se a paridade for ímpar
64) RRC	- ROTATE ACCUMULATOR RIGHT	- Girar o acumulador para a direita
65) RST	- RESTART	- Recomeça
66) RZ	- RETURN IF ZERO	- Retorno, se zero
67) SBB	- SUBTRACT WITH BORROW	- Subtração com carry
68) SBI	- SUBTRACT IMMEDIATE WITH BORROW	- Subtração imediata com carry
69) SHLD	- STORE HL DIRECT	- Armazenar par de registradores HL diretamente
70) SIM	- SET INTERRUPT MASK	- Posiciona a máscara de interrupção
71) SPHL	- MOVE HL TO SP	- Mover par de registradores HL para o stack pointer
72) STA	- STORE ACCUMULATOR DIRECT	- Armazena acumulador diretamente
73) STAX	- STORE ACCUMULATOR INDIRECT	- Armazena acumulador indiretamente
74) STC	- SET CARRY	- Posiciona o carry-bit
75) SUB	- SUBTRACT	- Subtrai
76) SUI	- SUBTRACT IMMEDIATE	- Subtração imediata (subtrai do acumulador uma constante)
77) XCHG	- EXCHANGE H L WITCH D E	- Trocar H e L com D e E
78) XRA	- EXCLUSIVE OR WITCH ACCUMULATOR	- OU exclusivo com acumulador
79) XRI	- EXCLUSIVE OR IMMEDIATE WITCH ACCUMULATOR	- OU exclusivo imediato com acumulador
80) XTHL	- EXCHANGE H e L WITCH TOP OF STACK	- Trocar H e L com o início do "STACK"

Com relação ao conjunto de instruções apresentado, cabe destacar o fato de que tanto para o 8080 como para o 8085 esse conjunto de instruções é válido, pois para ambos os microprocessadores cada instrução possui o mesmo formato e a mesma função. Destacamos ainda o fato de que as instruções RIM e SIM são instruções exclusivas do microprocessador 8085, não fazendo parte, portanto, do conjunto de instruções do microprocessador 8080.

TIPOS DE INSTRUÇÕES:

O conjunto de instruções pode ser organizado em diferentes subconjuntos de acordo com certos critérios. Um dos critérios mais utilizados diz para agruparmos as instruções dos microprocessadores 8080 e 8085 "por grupo de instruções de função semelhante". Este critério nos permite ter uma idéia das principais operações imediatas que podem ser executadas pelas instruções dos microprocessadores.

Com base nesses fatos, teremos os seguintes tipos de instruções:

a) instruções aritméticas;
b) instruções lógicas;
c) instruções de transferência de dados;
d) instruções de mudança de seqüência do programa;
e) instruções de controle.

Vejamos agora a definição de cada um desses tipos:

a) Instruções aritméticas:

São as instruções utilizadas na realização das operações aritméticas: soma, subtração ou ajuste decimal.

b) Instruções lógicas:

São as instruções que permitem a realização de operações lógicas tais como: AND, OR, OR EXCLUSIVA, complemento ou rotação do conteúdo dos registradores.

c) Instruções de transferência de dados:

São as instruções utilizadas na transferência de dados entre registradores, memórias ou periféricas.

d) Instruções de mudança de seqüência do programa:

Este tipo de instruções são as que alteram a seqüência de execução do programa por meio de salto para um determinado endereço.

e) Instruções de controle:

São as instruções de controle da unidade central de pro-

cessamento tais como: habilita interrupção, desabilita interrupção, entre em estado de retenção, etc.

Para finalizar esta lição, mostramos na figura 16 uma fotografia interna do microprocessador 8085 da Intel.

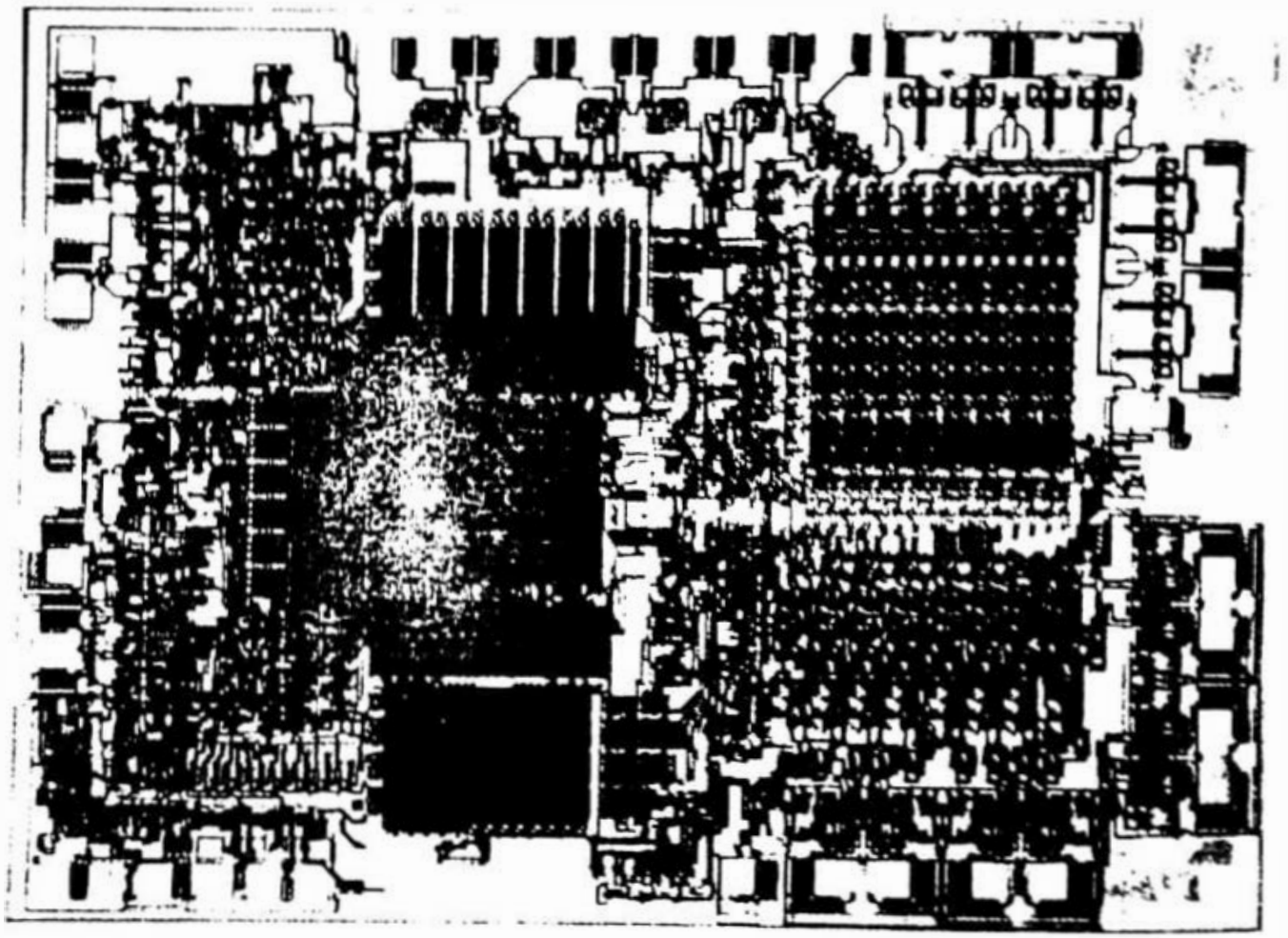


FIGURA 16

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 CEP 86100 · LONDRINA · PARANÁ

EXAME DA LIÇÃO MC-11

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) O microprocessador 8080 pertence à:

a) quarta geração dos microprocessadores;
b) terceira geração dos microprocessadores;
c) segunda geração dos microprocessadores;
d) primeira geração dos microprocessadores.

2) Além do sinal terra, o microprocessador 8080 requer, para o seu funcionamento:

a) três níveis de tensão de alimentação;
b) dois níveis de tensão de alimentação;
c) apenas um nível de tensão de alimentação;
d) N.R.A.

3) Os pinos que vão de D_0 até D_7 formam o:

a) barramento de endereço;
b) barramento de dados;
c) barramento de direções;
d) barramento de sincronização.

4) Para que o 8080 reconheça e entenda que um dado proveniente de uma memória ou de um dispositivo de entrada/saída se encontra à disposição no barramento de dados para ser lido por ele, é necessário que exista um sinal no pino:

a) SYNC;
b) RESET;
c) INT;
d) READY.

5) Qual é o sinal que faz com que um programa recomece a partir da posição zero?

a) sinal de SYNC;
b) sinal de RESET;
c) SINAL de INTE;
d) Sinal de READY;

6) No microprocessador 8080, os sinais que poderão ser manipulados pelo operador são:

a) INT e RESET;
b) INT e INTE;
c) RESET e $\overline{WR}$;
d) RESET e INTE;

7) A seção de lógica e aritmética do 8080 possui:

a) unidade de lógica e aritmética;
b) flags;
c) latch e acumulador;
d) os itens a, b e c estão corretos.

8) O circuito integrado responsável pelo controle do sistema do microprocessador 8080, demultiplexando o barramento de dados de tal modo a dar origem aos barramentos de dados e controle, é:

a) 8085;
b) 8224;
c) 8228;
d) N.R.A.

9) Entre as diversas vantagens, oferecidas pelo microprocessadore 8085, com relação ao 8080, temos:

a) maior velocidade de processamento;
b) maior faixa de temperatura de trabalho;
c) apenas dois níveis de tensão de alimentação;
d) N.R.A.

10) Quais das instruções abaixo são exclusivas do microprocessador 8085:

a) SIM;
b) RIM;
c) SUB;
d) os itens a e b estão corretos.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-12

MICROPROCESSADOR Z-80

INTRODUÇÃO

Continuando o nosso estudo sobre os principais microprocessadores, estudaremos nesta lição o microprocessador Z-80.

O Z-80 é produzido pela ZILOG. Trata-se, pois, de um microprocessador de 8 bits e cuja popularidade tem crescido muito atualmente. Este microprocessador não requer muitos circuitos de sustento, o que vem facilitar a sua utilização em unidades centrais de processamento.

As razões do grande sucesso do microprocessador Z-80 devem-se às vantagens do seu hardware e do seu conjunto de instruções quando comparado com outros microprocessadores de 8 bits.

Muito embora a tendência atual é passar a incrementar o uso dos microprocessadores de 16 bits, os microprocessadores de 8 bits devem, por um bom tempo, continuar a dominar o mercado. Um exemplo disto é que hoje existem versões do microprocessador Z-80 de 2,5 MHz, de 4,0 MHz, de 6,0 MHz e, mais recentemente, a ZILOG lançou no mercado uma outra versão do Z-80 que é o de 8,0 MHz, designado Z-80 H. Com isto, consegue-se uma maior velocidade de processamento, sem que

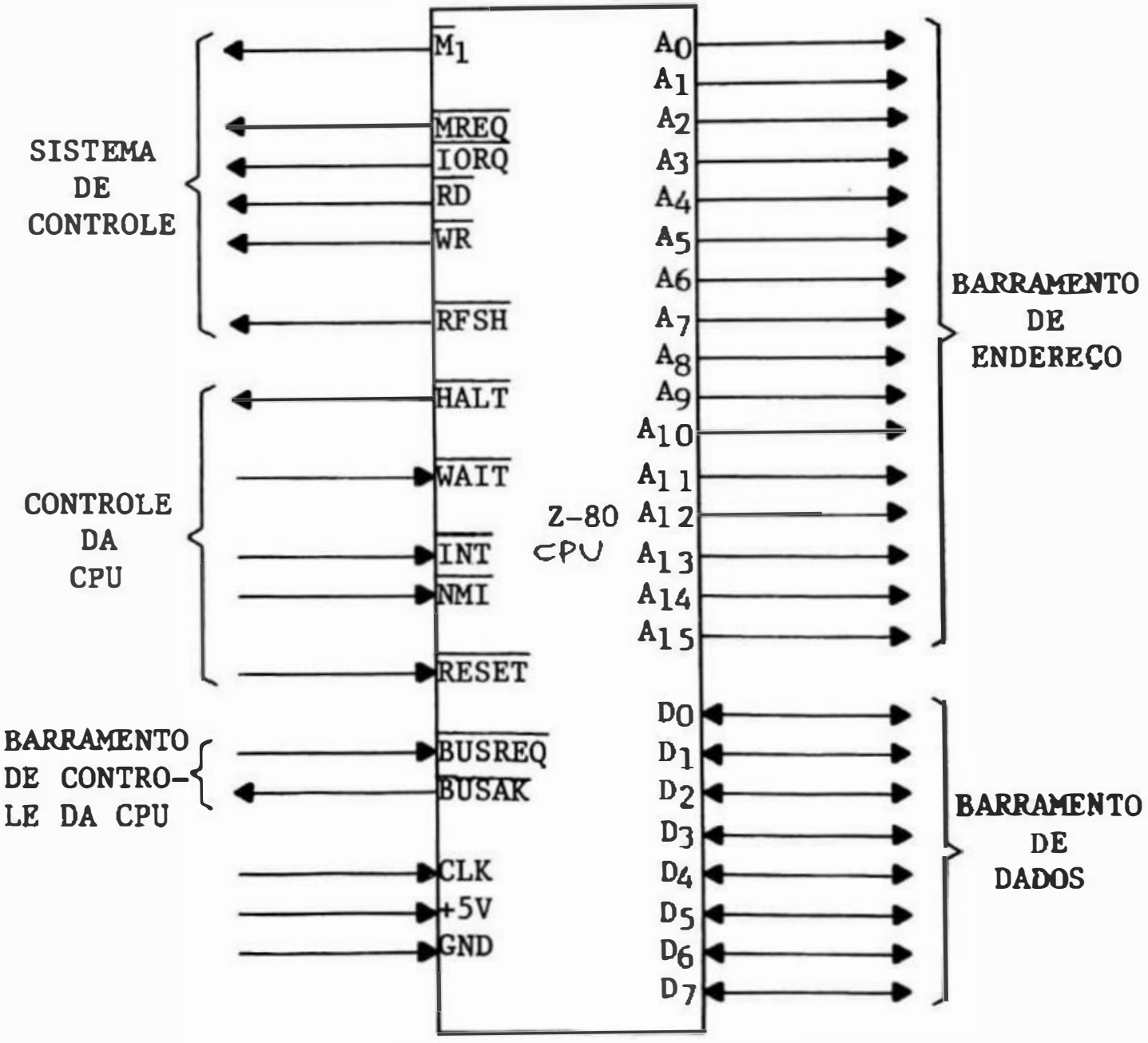
seja necessário investir em infra-estrutura para o desenvolvimento de microprocessadores bem mais complexos.

Com o intuito de facilitar o estudo do microprocessador Z-80, vamos apresentar em seguida a pinagem do mesmo.

DESCRIÇÃO DA PINAGEM DO MICROPROCESSADOR Z-80

O Z-80 é um microprocessador de terceira geração, desenvolvido em um único circuito integrado com 40 pinos. A figura 1 nos mostra a configuração de seus pinos e as respectivas funções.

FIGURA 01



A figura 2, por sua vez, nos mostra a configuração da pinagem do Z-80 com suas referidas atribuições.

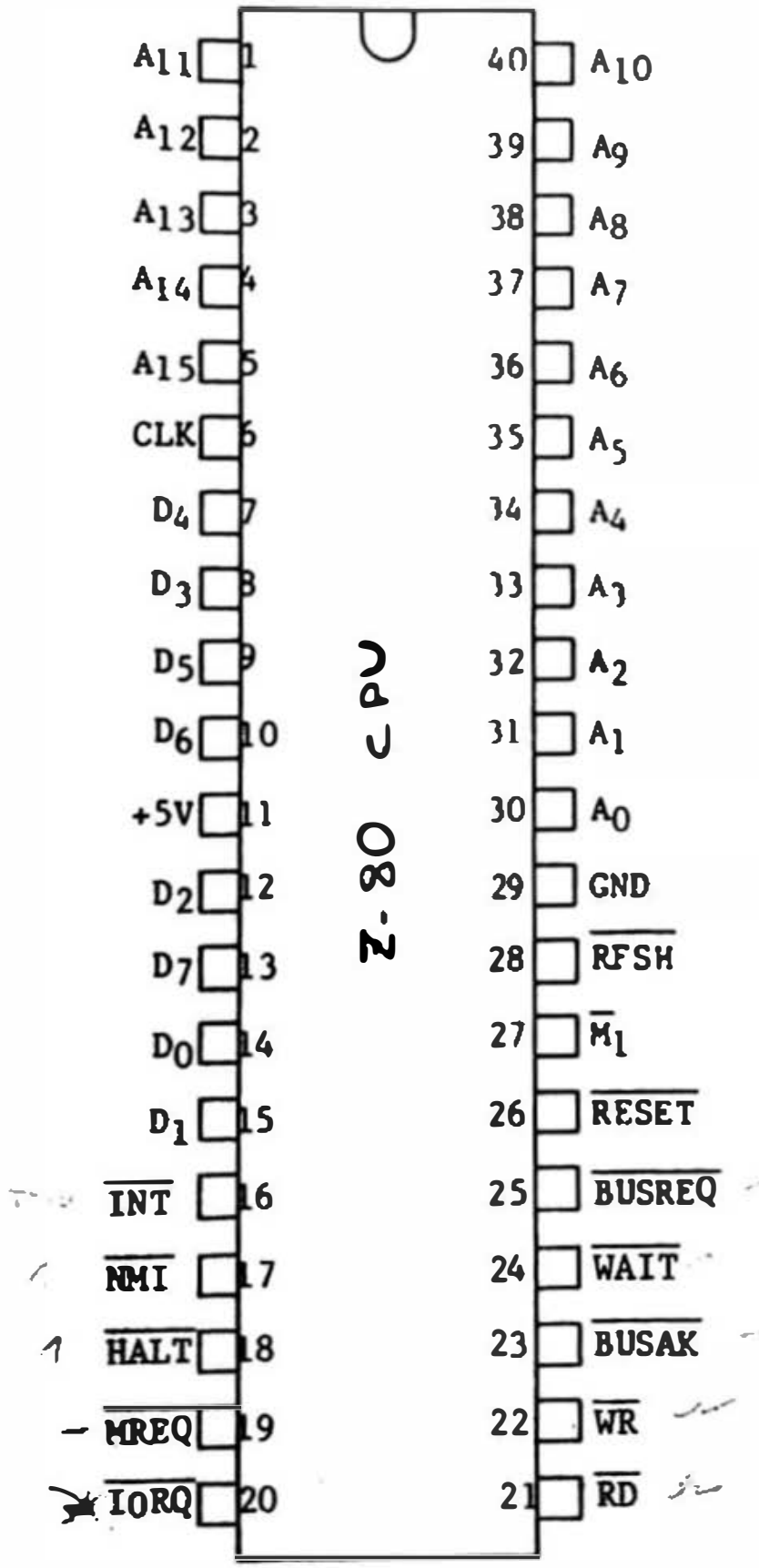


FIGURA 02

Em seguida, apresentamos uma descrição da pinagem do microprocessador Z-80.

A_0 até A_{15}:

São dezesseis pinos, os quais constituem o barramento de endereço (ou barramento de direções). Esses pinos, de saída, são do tipo "três-estados", ativados quando colocados em nível lógico alto; podendo, também, assumir o nível lógico baixo (desativado) e o estado de alta impedância. Estas saídas, normalmente, são utilizadas para o endereçamento de memórias ou de dispositivos de entrada/saída (E/S).

Devido ao fato do Z-80 ser um microprocessador, cujo barramento de endereço é composto de 16 bits, o máximo de memória que podemos endereçar, através deste barramento, é de 64K, ou seja, $2^{16} = 65.536 \cong 64K$. Além disto, o Z-80 utiliza um barramento de dados composto de 8 bits, o que possibilita endereçar até 256 ($2^8 = 256$) dispositivos de entrada/saída (E/S).

A_0 é o bit de endereço menos significativo e A_{15} é o endereço mais significativo.

D_0 até D_7:

Esses oito pinos (D_0 a D_7) formam o barramento de dados. Trata-se, pois, de pinos de entrada/saída (E/S) do tipo "três-estados" (three-states). Esse barramento de dados forma uma via de comunicação bidirecional entre o microprocessador e as memórias e entre os dispositivos de entrada/saída (E/S), para a transferência de dados ou instruções.

$\overline{M_1}$ (Machine cycle one):

Trata-se de uma saída, a qual é ativada, quando se encontra em um nível lógico baixo. Este pino indica que está sendo executado um ciclo de busca de instrução. Toda instrução, ao ser executada, exige que o Z-80, primeiramente, realize a busca de seu código de operação (OP) que está armazenado na memória, depositando-o, em seguida, no registro de instrução para então interpretá-lo.

$\overline{\text{MREQ}}$ (MEMORY REQUEST):

Trata-se de uma saída, também em "três-estados", ativada, quando colocada em um nível lógico baixo. Este sinal indica a requisição da memória e, em seguida, é levado ao barramento de endereços. Se o endereço for válido, a memória poderá então ler ou escrever.

$\overline{\text{IORQ}}$ (INPUT/OUTPUT REQUEST):

É uma saída em "três-estados", ativada, quando levada para um nível lógico baixo. O sinal $\overline{\text{IORQ}}$ indica que, entre os 8 bits menos significativos (A_0 a A_7) do barramento de endereço, existe um deles apto para a realização de uma operação de entrada ou saída (E/S). Um sinal $\overline{M_1}$ também será capaz de gerar um sinal $\overline{\text{IORQ}}$, se uma interrupção for aceita.

$\overline{\text{RD}}$ (MEMORY READ) e $\overline{\text{WR}}$ (MEMORY WRITE):

Saídas em "três-estados", ativadas, quando em nível lógico baixo (0). Esses sinais ($\overline{\text{RD}}$ e $\overline{\text{WR}}$) quando ativados indicam respectivamente, entrada ou saída de dados da memória.

O dispositivo endereçado por qualquer um desses sinais ($\overline{\text{RD}}$ e $\overline{\text{WR}}$) coloca os dados no barramento de dados (isto quando $\overline{\text{RD}}$ ou $\overline{\text{WR}}$ estiverem ativados) deixando-os disponíveis para o periférico ou para a memória. Assim, podemos dizer que esses sinais, $\overline{\text{RD}}$ e $\overline{\text{WR}}$, indicam, respectivamente, que o Z-80 está realizando operações de leitura ou escrita.

A figura 3 nos mostra uma relação das linhas $\overline{\text{MREQ}}$, $\overline{\text{RD}}$ e $\overline{\text{WR}}$.

$\overline{\text{RFSH}}$ (REFRESH):

Trata-se de uma saída que também é ativada, quando em nível lógico baixo (0). Este sinal somente é usado quando trabalhamos com memórias RAM'S, dinâmicas, pois este tipo de memória necessita de um refresh periódico para que não se perca a informação nela armazenada. Na maioria das memórias RAM'S, dinâmicas, o refresh de cada célula deve ocorrer em espaços de tempo inferiores a 2 ms.

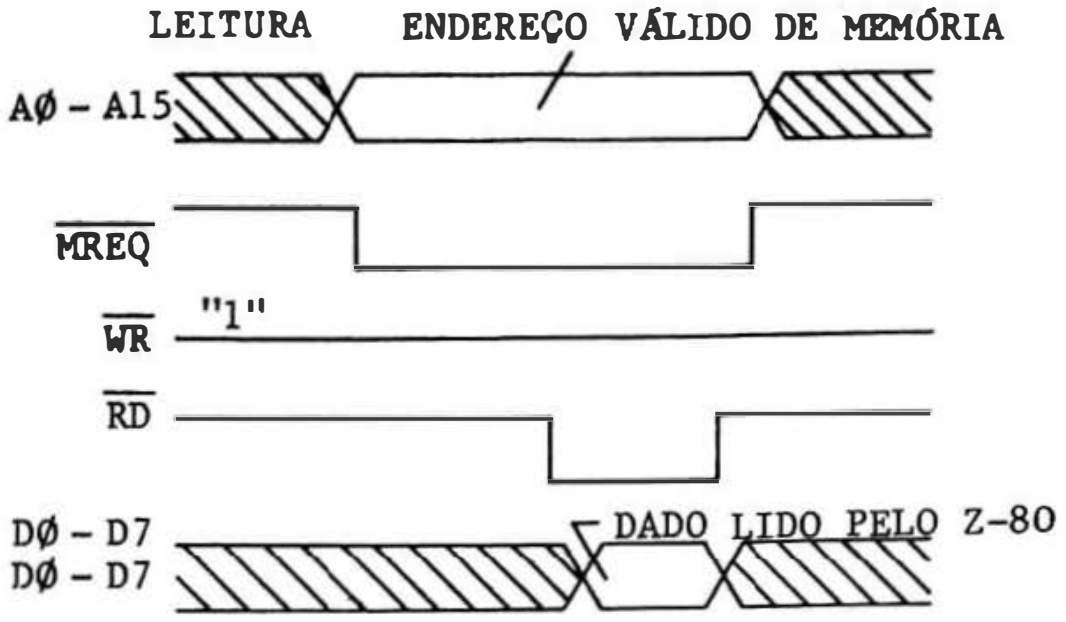
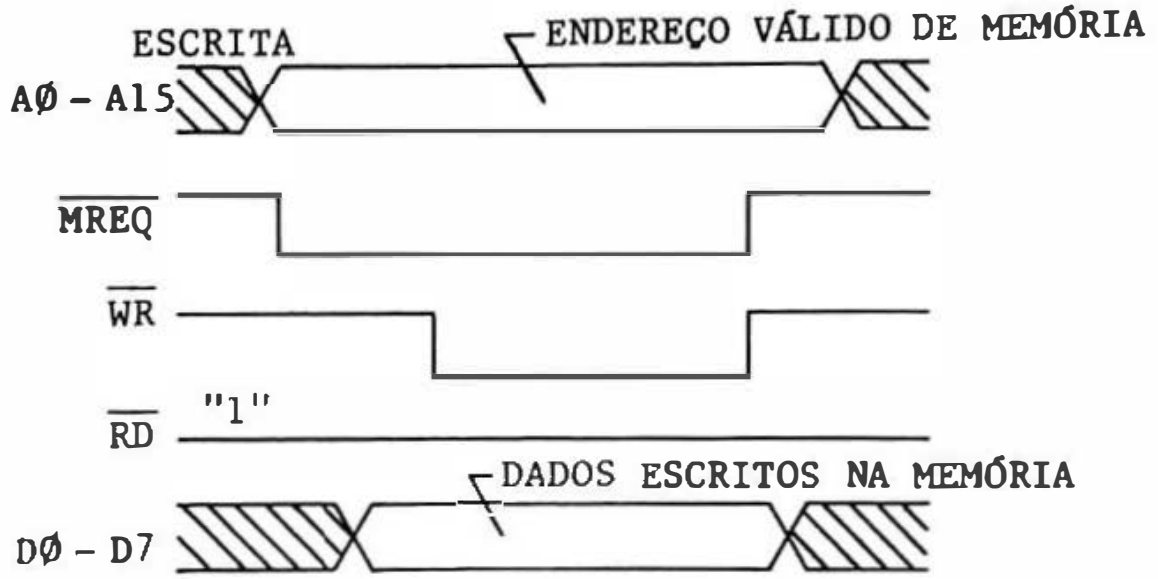


FIGURA 03



Podemos dizer que o refresh é, basicamente, uma operação de leitura em determinados endereços, sem que haja, de uma forma efetiva, a transferência de informação.

Com o objetivo de se implementar o refresh costuma-se utilizar um controlador adicional, o qual é constituído de contadores, portas e outros circuitos lógicos. Porém, no caso do microprocessador Z-80, as funções de refresh são implementadas a partir do barramento de endereço, do registrador R e do sinal $\overline{RFSH}$, dispensando, deste modo, os controladores externos.

Quando o $\overline{RFSH}$ for igual a "0" binário e o $\overline{MREQ}$ também for "0", o conteúdo do registrador R será colocado nos sete bits menos significativos (A_0 a A_6) do barramento de endereço. E, a cada busca de instrução, o conteúdo do registrador R é incrementado de uma unidade.

$\overline{\text{HALT}}$ (HALT STATE):

Trata-se de uma saída ativada, quando colocada em nível lógico baixo (0), o que ocorre após a execução da instrução "halt". Essa instrução (halt) coloca o microprocessador Z-80 no estado de espera e, basicamente, é utilizado em dois casos:

a) no final de um programa, após todas as instruções terem sido executadas;

b) quando for necessário que o Z-80 permaneça parado, aguardando um pedido de interrupção.

O estado de halt caracteriza-se pela paralisação do processamento e, para o Z-80 sair deste estado, é preciso uma interrupção ou a ativação da linha $\overline{\text{RESET}}$.

O sinal de halt é um sinal indicador de parada.

$\overline{\text{WAIT}}$ (WAIT-STATE):

O sinal $\overline{\text{WAIT}}$ é um sinal de pedido de espera, cujo objetivo é de sincronizar memórias e dispositivos de E/S lentos com o microprocessador Z-80. A entrada $\overline{\text{WAIT}}$ é ativada quando estiver em nível lógico (0), o microprocessador Z-80 ficará "parado" aguardando que a memória ou o dispositivo de E/S responda a sua solicitação de leitura ou escrita.

$\overline{\text{INT}}$ (INTERRUPT REQUEST):

A entrada $\overline{\text{INT}}$ (pino 16) só reconhece o pedido de interrupção, quando esta estiver em nível lógico baixo (0). Este sinal é gerado pelos dispositivos externos e é aceito no término da instrução que está sendo executada, isto se o flip-flop de "interrupt enable" estiver habilitado e se a entrada $\overline{\text{BUSREQ}}$ estiver desativada ($\overline{\text{BUSREQ}} = 1$).

Este tipo de interrupção tem três modos de operação: modos 0, 1 e 2 ou, também chamados de IN_0, IN_1 e IN_2. Para os modos IN_0 e IN_2 é preciso que o circuito externo forneça um código que define o endereço inicial da rotina de tratamento. Para que isto seja possível, o microprocessador Z-80 ativa a linha $\overline{\text{IORQ}}$ ($\overline{\text{IORQ}} = 0$) durante o ciclo de busca de instru-

ção ($M_1 = 0$) seguinte à aceitação do pedido. Desta forma, a combinação destes dois sinais, conforme nos mostra a figura 4, fornece o sinal de reconhecimento de interrupção $\overline{INTA}$.

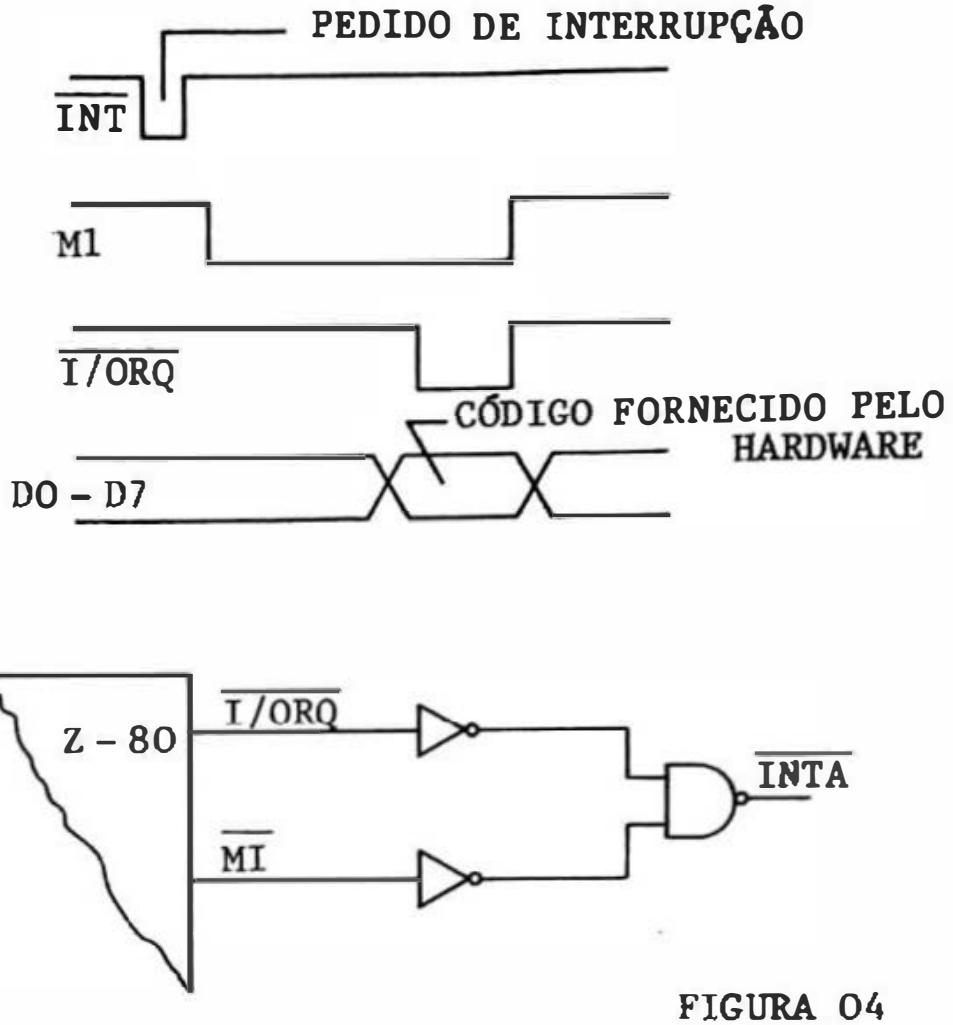


FIGURA 04

NMI (NON MASKABLE INTERRUPT):

O $\overline{NMI}$ (pino 17) reconhece o pedido de interrupção quando ocorre uma transição negativa, ou seja, transição do nível lógico 1 para o nível lógico 0. Quando ocorre esta transição, o microprocessador Z-80 reconhece, compulsoriamente, no final da instrução em exercício, o pedido para interromper.

Esta linha tem total prioridade sobre a linha $\overline{INT}$. A importância desta entrada é tão grande que com exceção das entradas $\overline{BUSREQ}$ e $\overline{WAIT}$, não existe outro mecanismo de hardware nem de software capaz de impedir o atendimento à interrupção

proveniente dela ($\overline{NMI}$).

Ao reconhecer um pedido de interrupção procedente da entrada $\overline{NMI}$, ocorrem as seguintes ações no Z-80:

a) O conteúdo atual do contador de programa (PC) é salvo na pilha, a fim de num futuro próximo ser utilizado como endereço de retorno de sub-rotina.

b) O Z-80 carrega 0066H (base hexadecimal) no contador de programa, transferindo o controle para este endereço, onde, efetivamente, inicia a rotina de tratamento da interrupção $\overline{NMI}$.

$\overline{RESET}$:

Trata-se de uma entrada, a qual é ativada quando se encontra um nível lógico "0". Este sinal (sinal de RESET) é utilizado para inicializar o microprocessador Z-80. Ele deve ser acionado imediatamente após o "power-up" ou a qualquer instante em que se deseja reinicializar o Z-80. Assim que a entrada $\overline{RESET}$ vai para o nível lógico "0", ocorrem os seguintes fatos:

1º) Os barramentos de dados e de endereços vão para o terceiro estado;

2º) Todas as saídas de controle ficam inativas (nível lógico 1);

3º) Os registros R e I também são resetados, ou seja, ficam com valor "0";

4º) O modo de interrupção é levado para um nível lógico baixo (0);

5º) As interrupções provenientes da entrada $\overline{INT}$ são inibidas;

6º) O contador de programa (PC) é zerado, ou seja, fica com o valor 0000.

$\overline{BUSREQ}$ (BUS REQUEST) e $\overline{BUSAK}$ (BUS ACKNOWLEDGE):

A entrada $\overline{BUSREQ}$ é ativada quando em nível lógico baixo (0). Este sinal solicita à UCP suas linhas de endereço, levando-as para o estado de alta impedância, de modo que outros

dispositivos externos passam a controlar as linhas do barramento.

Já, a saída $\overline{\text{BUSAK}}$, que também é ativada quando colocada em nível lógico baixo (0), indica a requisição de um dispositivo externo ao barramento de endereços. Neste momento as linhas de endereços e de dados são levadas para o estado de alta impedância, possibilitando o controle das mesmas através de outros dispositivos externos.

Quando, por exemplo, um dispositivo externo necessita controlar os barramentos para realizar transferências entre memória ou dispositivos de entrada e saída (E/S), ele fornece um nível lógico "0" para a entrada $\overline{\text{BUSREQ}}$. Logo em seguida o Z-80 coloca os barramentos (de dados e de endereços) no estado de alta impedância e sinaliza, através da saída $\overline{\text{BUSAK}}$, colocando-a no nível lógico baixo (0). No momento em que isto ocorre, o microprocessador Z-80 paralisa o seu processamento, permitindo que algum dispositivo externo assuma o controle dos barramentos.

Para ilustrar o nosso exemplo, a figura 5 nos mostra um diagrama de blocos de um microcomputador equipado com um controlador de disco. Através desta figura, podemos observar as linhas $\overline{\text{BUSREQ}}$ e $\overline{\text{BUSAK}}$, as quais controlam os barramentos de dados e endereços.

No caso apresentado na figura 5, uma unidade controladora de disco pode realizar operações de acesso direto à memória (DMA), ou ainda, um outro microprocessador pode dispor dos recursos dos barramentos.

Na maioria dos controladores, a taxa de transferência de dados é superior à velocidade de processamento do Z-80. Isto leva o Z-80 a não conseguir realizar transferências controladas por software, sendo, portanto, necessário o uso do DMA com controlador de disco. Assim, o controlador bloqueia o Z-80, que coloca seus barramentos no estado de alta impedância, e realiza a troca de informações com a memória sem o auxílio do microprocessador.

CLK DO Z-80 (CLOCK):

Trata-se de uma entrada para o pulso de sincronização (pulso de clock) do Z-80. Esta entrada corresponde ao pino 6.

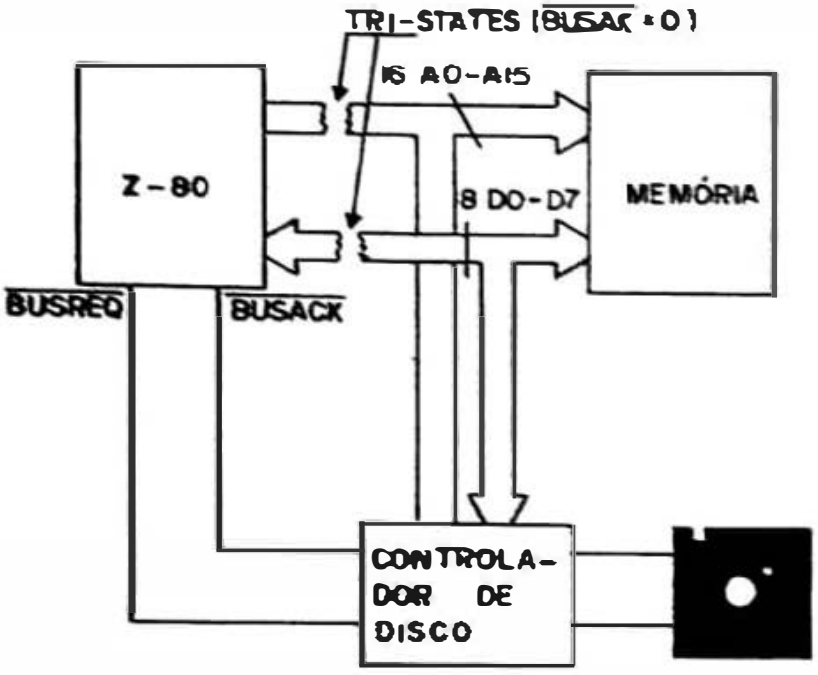
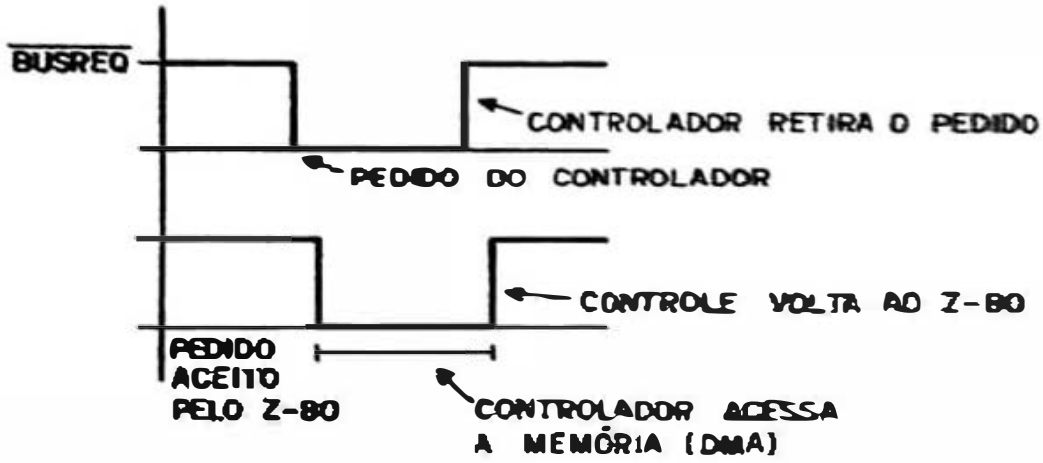


FIGURA 05



Este pulso de clock, na maioria das vezes, é implementado com um oscilador a cristal. A figura 6 nos mostra um diagrama de blocos da entrada de clock do Z-80.

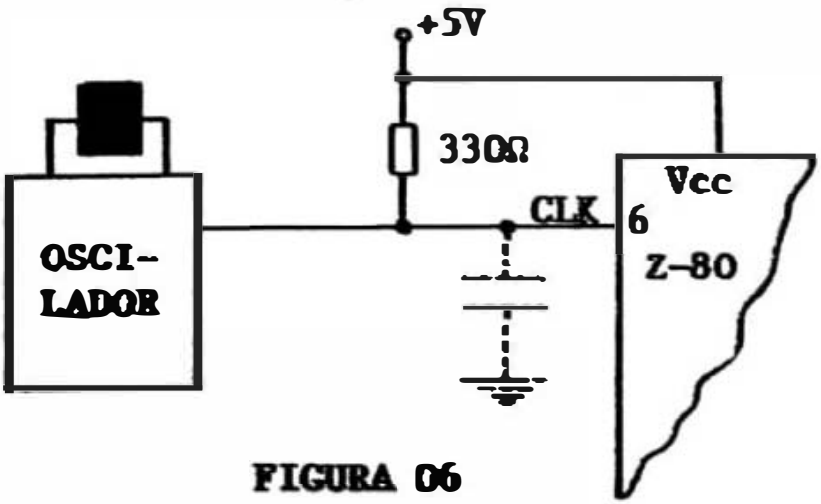


FIGURA 06

O capacitor da entrada de clock possui uma capacitância de 35 pF. Portanto, é recomendável a colocação de um resistor de 300Ω, chamado resistor de "pull up", para a obtenção de valores menores nos tempos de subida e descida do sinal pulsado.

A ZILOG, fabricante do Z-80, oferece quatro versões do microprocessador Z-80, cada uma delas caracterizada pela freqüência máxima do pulso de clock. A tabela apresentada na figura 7 nos mostra estas quatro versões do microprocessador Z-80.

VERSÕES DO Z-80	
MODELO	FREQÜÊNCIA MÁXIMA DO CLOCK
Z-80	2,5 MHz
Z-80A	4 MHz
Z-80B	6 MHz
Z-80H	8 MHz

FIGURA 07

ARQUITETURA DO MICROPROCESSADOR Z-80

Para facilitar o estudo da arquitetura do microprocessador Z-80, iremos fazê-lo por partes:

1) Barramentos de endereços e de dados;

2) Mapa de registros;

3) Unidade de controle;

4) Unidade Lógica e Aritmética (ULA).

A figura 8 nos mostra o diagrama de bloco da arquitetura interna do microprocessador Z-80.

BARRAMENTOS DE ENDEREÇOS E DE DADOS

O barramento de endereços é composto de 16 bits, o que possibilita endereçar até 65536 ($= 2^{16}$) bytes ou 64 K bytes de memória. Além disso, o Z-80 utiliza 8 bits deste barramento para endereçar até 256 ($= 2^{8}$) dispositivos de entrada e saída da (E/S).

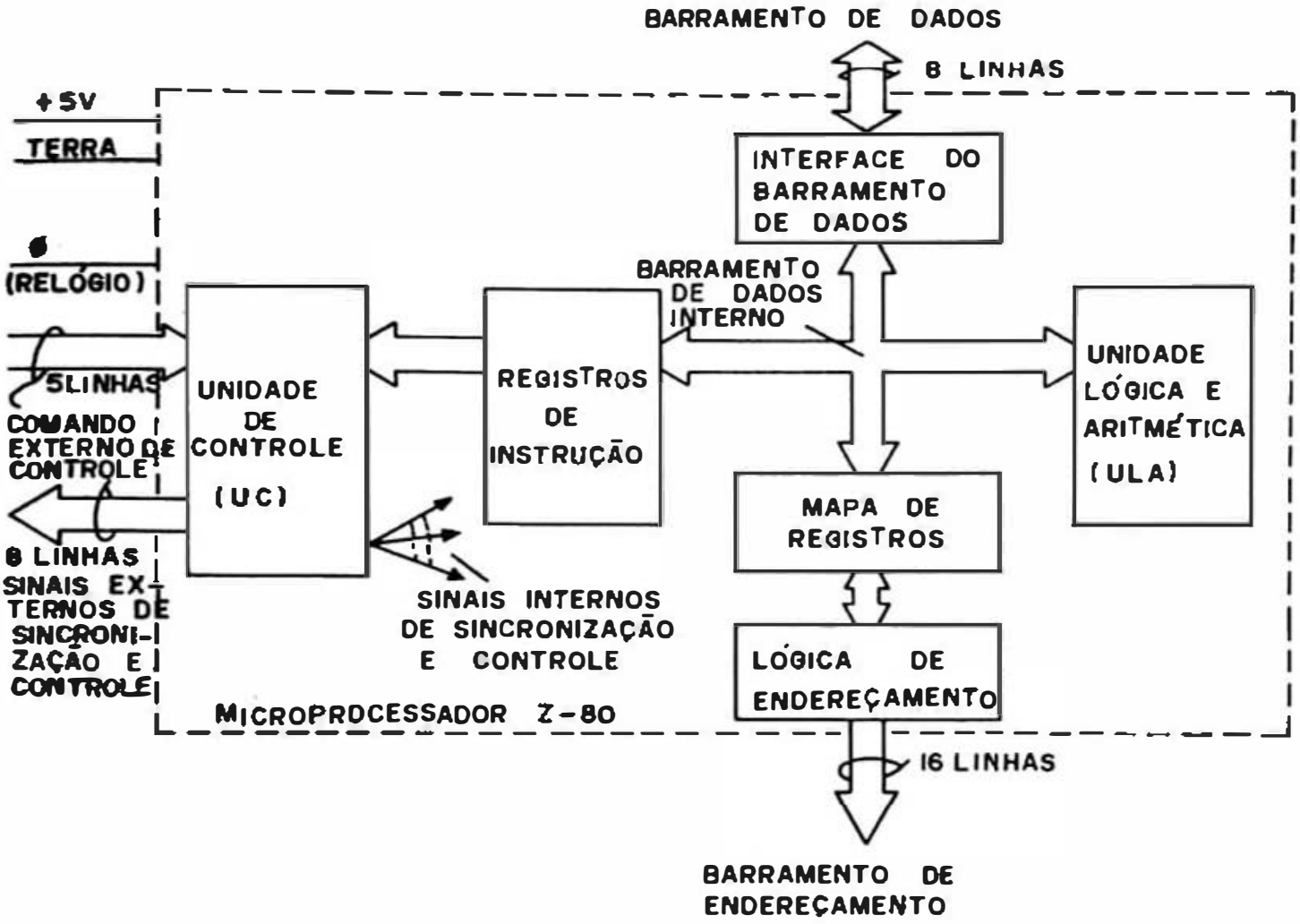


FIGURA 08

O barramento de dados bidirecional é composto por 8 bits. Este barramento se ramifica no interior do Z-80, através do barramento de dados interno, o qual estabelece a comunicação entre o mapa de registros, o registro de instruções e a Unida de Lógica e Aritmética (ULA). A interface do barramento de dados acopla os setores internos e externos deste barramento e possui duas funções básicas:

a) isolamento elétrico;

b) captura e armazenamento.

O isolamento elétrico (buffer) entre circuitos elétricos dos barramentos internos e externos, faz com que as memórias e dispositivos de entrada e saída não sobrecarreguem os circuitos internos do microprocessador Z-80 permitindo, assim,

que um número maior destes dispositivos externos sejam conectados ao barramento.

Já, a captura e armazenamento (latch) permite que estados lógicos de curta permanência no barramento de dados sejam capturados e armazenados temporariamente.

MAPA DE REGISTROS

O mapa de registros é um conjunto formado por todos os registros de processamento a que o programador tem acesso. Trata-se, pois, de um dos aspectos mais importantes a ser considerado no estudo do hardware do Z-80, uma vez que o programador tem acesso aos mesmos.

A figura 9 nos mostra um quadro com o mapa de registros do microprocessador Z-80.

Em seguida, apresentaremos um breve estudo sobre a organização lógica, estrutural e funcional desses elementos.

FIGURA 09

REGISTROS DE USO GERAL

Também conhecidos por registradores de função geral, são registros de 8 bits cada um e, no caso do Z-80, são em número de quatorze: A, B, C, D, E, H, L e, A', B', C', D', E', H', L'. Para facilitar o nosso estudo, vamos classificá-los em dois grupos: primários e secundários (veja figura 10).

REGISTRADORES DE USO GERAL	
PRIMÁRIOS	SECUNDÁRIOS
A	A'
B	B'
C	C'
D	D'
E	E'
H	H'
L	L'

FIGURA 10

Os registros de uso geral e os de condição (F e F') também são classificados em ativos e passivos. Isto, durante a execução de um programa. Dentre os registros, em qualquer instante, teremos sempre oito registradores passivos e oito registradores ativos.

Um registro é dito ativo quando responde pelos operandos das instruções. Por outro lado, um registro é tido como passivo quando mantém seu conteúdo inalterado até o momento de se tornar ativo e, conseqüentemente, modificar o seu conteúdo. Uma instrução especial do Z-80 seleciona A e F ou A' e F', enquanto que uma segunda instrução seleciona B, C, D, E, H, L, ou b', C', D', E', H', L'. A figura 11 nos mostra as combinações possíveis de A e F e os seis registros de uso geral restantes.

A	F
B	C
D	E
H	L

A	F
B'	C'
D'	E'
H'	L'

FIGURA 11

A'	F'
B	C
D	E
H	L

A'	F'
B'	C'
D'	E'
H'	L'

Uma das vantagens de se ter dois blocos de registros de uso geral está no fato de que o programador pode comutar rapidamente de um bloco para outro. Ele permite também um acesso mais rápido às informações do que se as mesmas estivessem armazenadas na memória, pois a capacidade de armazenagem dos registros é maior.

Os registros que o microprocessador 8080 possui são os mesmos registros de uso geral do Z-80, porém sem duplicação. Tanto no 8080 como no Z-80 as principais funções destes registros são:

a) Transferência de um grupo de 8 bits contidos num determinado registro ativo para uma certa posição de memória e vice-versa (veja a figura 12-a);

b) Transferência de um grupo de 8 bits contidos num determinado registro ativo para um outro registro (veja figura 12-b).

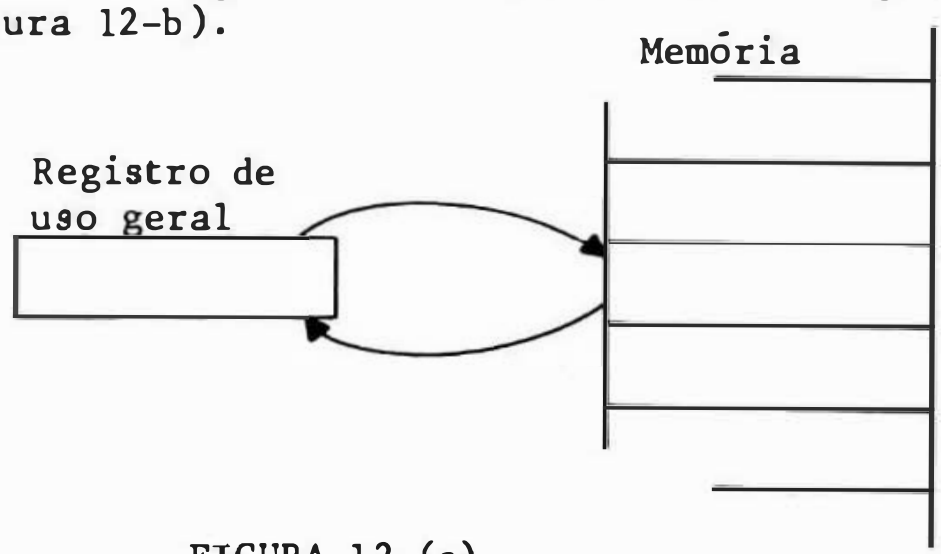


FIGURA 12 (a)

FIGURA 12 (b)

Em operações lógicas e aritméticas, tais como a adição de dois operandos ou a função OU-Exclusiva sobre dois operandos, somente podem ser feitas, usando o registro A (ou A') de um outro registro ou localização de memória. O resultado da operação sempre vai para o registro A.

Geralmente o registro A (ou A') é selecionado como registro principal para executar as operações lógicas e aritméticas. Já os seis registros restantes são agrupados aos pares: B,C; D,E; H,L. Em muitas das operações do Z-80, o dado contido dentro dos três pares de registros representam endereços de memória.

A ocorrência do estouro de capacidade (OVERFLOW) indica que o resultado da operação aritmética não coube no registro.

Vejamos agora alguns bits de condição:

a) Bit P/V - Este bit pode representar a paridade do resultado ou indicar o overflow. No caso de P/V estar representando a paridade, quando P/V = 1, significa que a paridade é par e, quando P/V = 0, significa que a paridade é ímpar. Porém, quando o bit P/V tiver representado o overflow, isto é, toda vez que ele for igual a "1", após uma operação aritmética, significa a ocorrência do overflow (estouro da capacidade).

b) Bit Z (Zero bit) - Toda vez que este bit for igual a "1" (Z = 1), significa que o resultado da operação (lógica ou aritmética) é nulo. Por outro lado, quando o valor de Z for igual a zero (Z = 0), significa que o resultado da operação não é nulo.

c) Bit H (Half-carry bit) - Sempre que o valor deste bit for igual a "1" (H = 1), significa a ocorrência de "vai um" ou "empresta um", quando da realização de uma soma ou de uma subtração.

d) Bit S (Signal bit) - Este bit representa o sinal do resultado, ou seja, se S = 0, significa que o resultado é positivo e, se S = 1, significa que o resultado é negativo.

REGISTROS DE USO ESPECÍFICO

Esses registros possuem uma função muito bem definida durante o processamento, sendo também acessíveis ao programa-

dor. São eles:

a) Contador de programa (PC - Program Counter);
b) Ponteiro de pilha (SP - Stack Pointer);
c) Registro de índice (IX - Index Register IX e IY);
d) Registro de interrupção (I - Interrupt Register);
e) Registro de refresh (R - Memory refresh).

Os quatro primeiros registros (PC, SP, IX e IY) são registros de 16 bits. Já os dois últimos (I e R) são registros de 8 bits. Vejamos agora uma descrição de cada um deles.

a) Contador de Programa (PC):

Trata-se de um registrador de 16 bits que mantém o endereço da instrução corrente, ou seja, da instrução que está sendo executada, durante o tempo de busca da mesma na memória. Em outras palavras, podemos dizer que o Contador de Programa guarda os 16 bits de endereço da instrução que está sendo procurada na memória.

O PC é, automaticamente, incrementado após o seu conteúdo ter sido transferido para as linhas de endereço e é atualizado ao término de cada instrução, de modo a conter o endereço da próxima instrução a ser executada.

No microprocessador Z-80, as instruções podem ocupar um, dois, três ou até quatro bytes consecutivos da memória. O PC sempre indica o início da próxima instrução a ser executada, isto é, o seu primeiro byte. Desta forma, a UCP irá incrementar automaticamente o PC de um, dois, três ou quatro, dependendo do tamanho (número de bytes) da instrução que está sendo executada.

Devido à própria estrutura do PC, as instruções de um programa qualquer são organizadas na memória em posições consecutivas e em ordem crescente de endereços. A figura 13 nos mostra uma ilustração do funcionamento do PC.

Com relação ao programador, o PC somente é acessível no sentido em que este possa ser carregado ou armazenado. Nenhuma operação lógica ou aritmética é permitida no PC.

O PC também é tido como um indicador para a memória externa, o qual especifica o endereço da próxima instrução a ser executada.

Quando o microprocessador Z-80 executa instruções, tais como JP, CALL, RET e RESET e as quais quebram a atualização seqüencial do PC, pois estas instruções alteram o curso normal do programa, o contador de programa é então "carregado" com um novo valor.

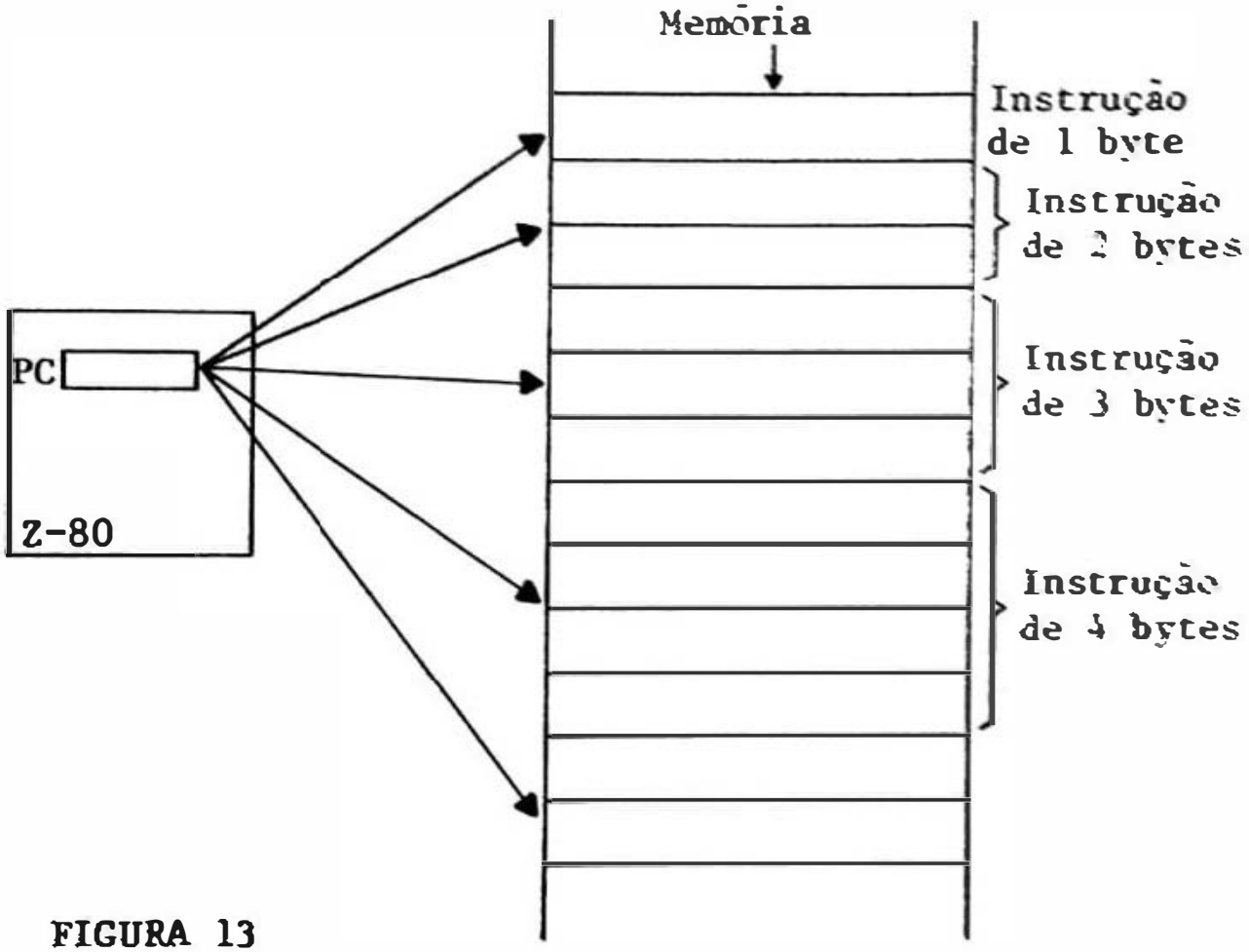


FIGURA 13

b) Ponteiro de Pilha (SP):

O registro SP, de 16 bits, que possui a função de ponteiro de pilha, é um indicador de empilhamento da memória externa. O conceito de empilhamento da memória externa não é exclusivo do Z-80, na verdade, a pilha é uma estrutura de informação implementada em RAM, muito utilizada em computadores de modo geral. Esse empilhamento da memória externa é simplesmente uma área da memória, posicionada à parte, para armazenagem temporária dos registradores da UCP, do registro Flag e do contador de programa.

A memória externa do ponteiro de pilha (SP) é organizada como uma LIFO (Last in first out), ou seja, último que entra, primeiro que sai. Conforme o dado é introduzido para

a área do empilhamento, o indicador de empilhamento é decrementado por uma contagem. Ao contrário, conforme o dado é retirado da pilha, o indicador de empilhamento é incrementado por uma contagem.

A pilha nos permite realizar interrupções de multiníveis, bem como manipular inúmeras sub-rotinas uma dentro da outra. Nos microcomputadores, a pilha se presta para o armazenamento temporário de registros, de forma sistemática e organizada.

c) Registro de Índice (IX e IY):

São dois registros de 16 bits cada um, os quais permitem o endereçamento indexado nos programas do microprocessador Z-80. Quando uma instrução é executada em um modo de endereçamento indexado, um dos registros de índices (IX ou IY) é utilizado para calcular endereços de operandos na memória.

O endereço efetivo do operando na memória é calculado, através da soma do campo de endereço da instrução (ou campo de deslocamento da instrução) com o conteúdo de um dos registros de índice especificado no campo do código da instrução.

Para finalizar, podemos dizer que nesta modalidade de endereçamento, um registrador índex é usado como um apontamento base para uma certa região da memória, onde o dado será gravado ou lido. Este tipo de endereçamento simplifica vários tipos de programa, especialmente, onde tabelas de dados são usadas. A figura 14 nos mostra um diagrama de blocos do funcionamento do registro de índice.

d) Registro I:

Registro I ou registro "vetor de interrupção" possui 8 bits, utilizado no atendimento de interrupções. O registro I pode, então, ser carregado com 8 bits de dados, especificando um endereço de memória. Este endereço, quando combinado com um endereço de 8 bits de ordem mais baixa, pelo dispositivo de interrupção, representará o endereço de memória, cujo conteúdo identifica o endereço de memória da "rotina de manejar interrupção".

Esses 8 bits de ordem mais baixa, ou seja, menos significativos do endereço, são gerados pelo dispositivo de entra-

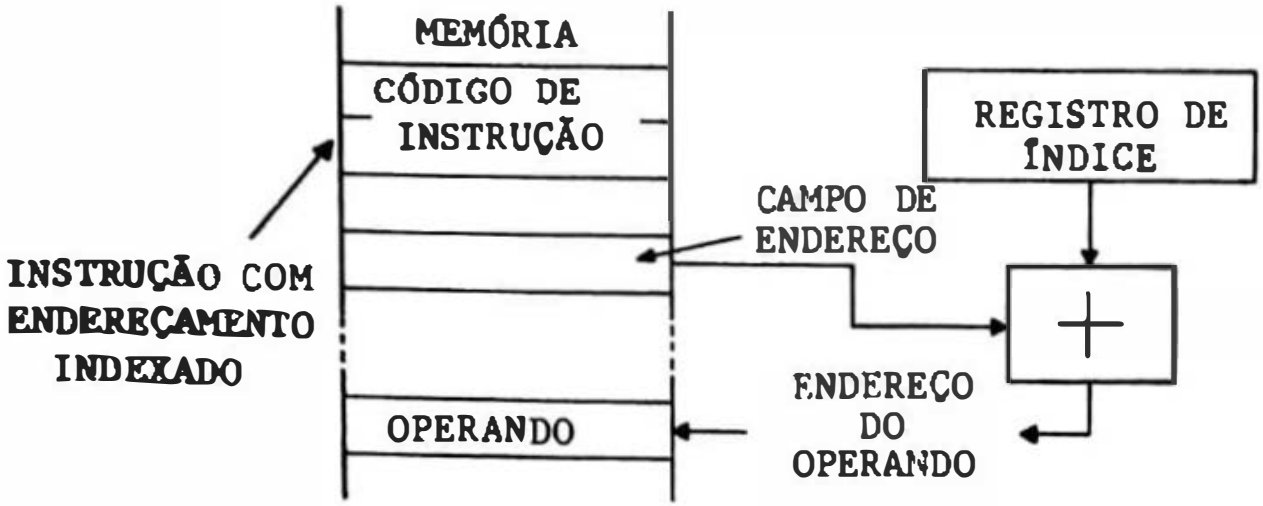


FIGURA 14

da e saída que deu origem ao pedido de interrupção. Os 8 bits menos significativos são fornecidos através do barramento de dados, em intervalos de tempo sincronizados pelo sinal de clock do microprocessador Z-80, durante o processo de atendimento da interrupção.

O registro I é utilizado em um dos três modos de interrupção que o Z-80 pode utilizar sob o controle de um programa. Dois destes modos são idênticos à ação de interrupção do 8080, permitindo até oito interrupções do tipo "VECTORED". Já, o terceiro modo permite uma interrupção especial, chamada "NINTH".

A figura 15 nos mostra, de uma forma bem simplificada, a formação do endereço VETOR DE INTERRUPÇÃO (I).

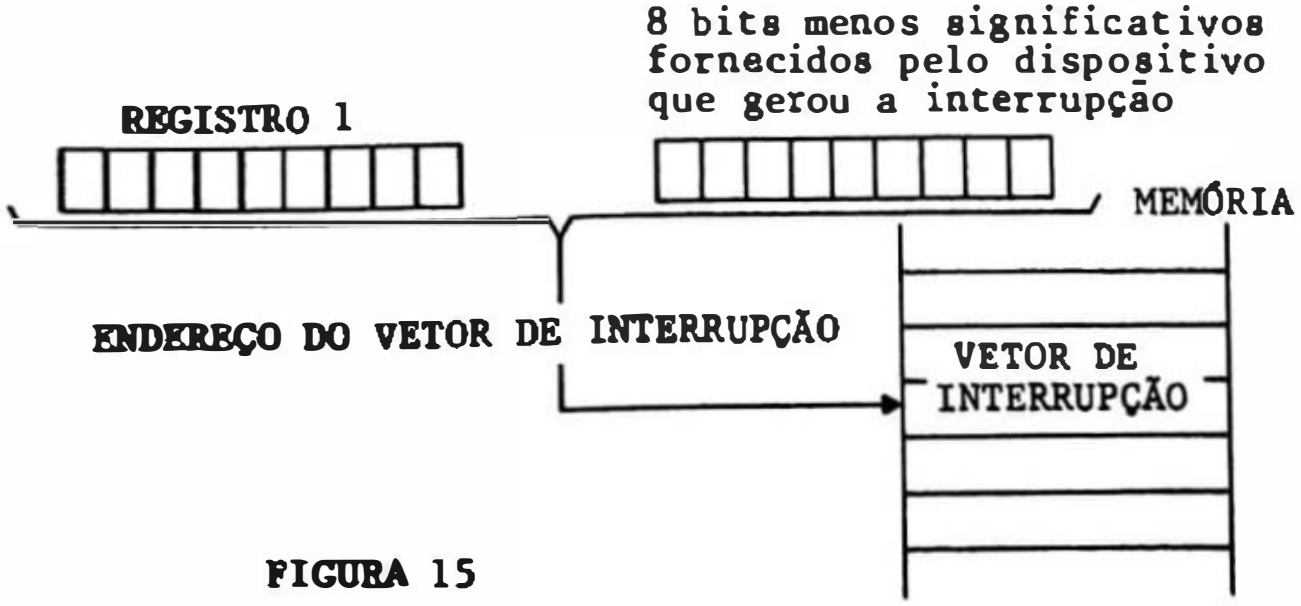


FIGURA 15

e) Registro R:

Trata-se de um registro de 8 bits, mas apenas 7 destes são utilizados. Esse registro também é conhecido como registro de refresh da memória ou registro de refresco para as memórias do tipo RAM dinâmica. Sua principal função é controlar o "refresh" automático de memórias semicondutoras dinâmicas.

Sabe-se que, normalmente, cada célula de memória dinâmica sofre um "refresh" periódico, num espaço de tempo inferior a 2 milissegundos, pois este tipo de memória exige que todas as suas células, sejam lidas ou "refrescadas", para que possam reter os seus conteúdos.

O registro R é incrementado, automaticamente, a cada ciclo de busca de instruções. O conteúdo do contador de refresco é, então, enviado para a memória, através dos 7 bits menos significativos do barramento de endereços (A_0 a A_6), durante o sinal de controle de refresh. Enquanto isso, a UCP do Z-80 está decodificando ou mesmo executando uma outra instrução.

Este modo de refresh é totalmente transparente ao programa, pois ocorre em paralelo com o processamento interno da instrução, justamente nos intervalos de tempo em que a execução da instrução não requer o uso dos barramentos, dispensando assim, o uso de circuitos externos.

REGISTRO DE CONDIÇÃO (FLAG REGISTER)

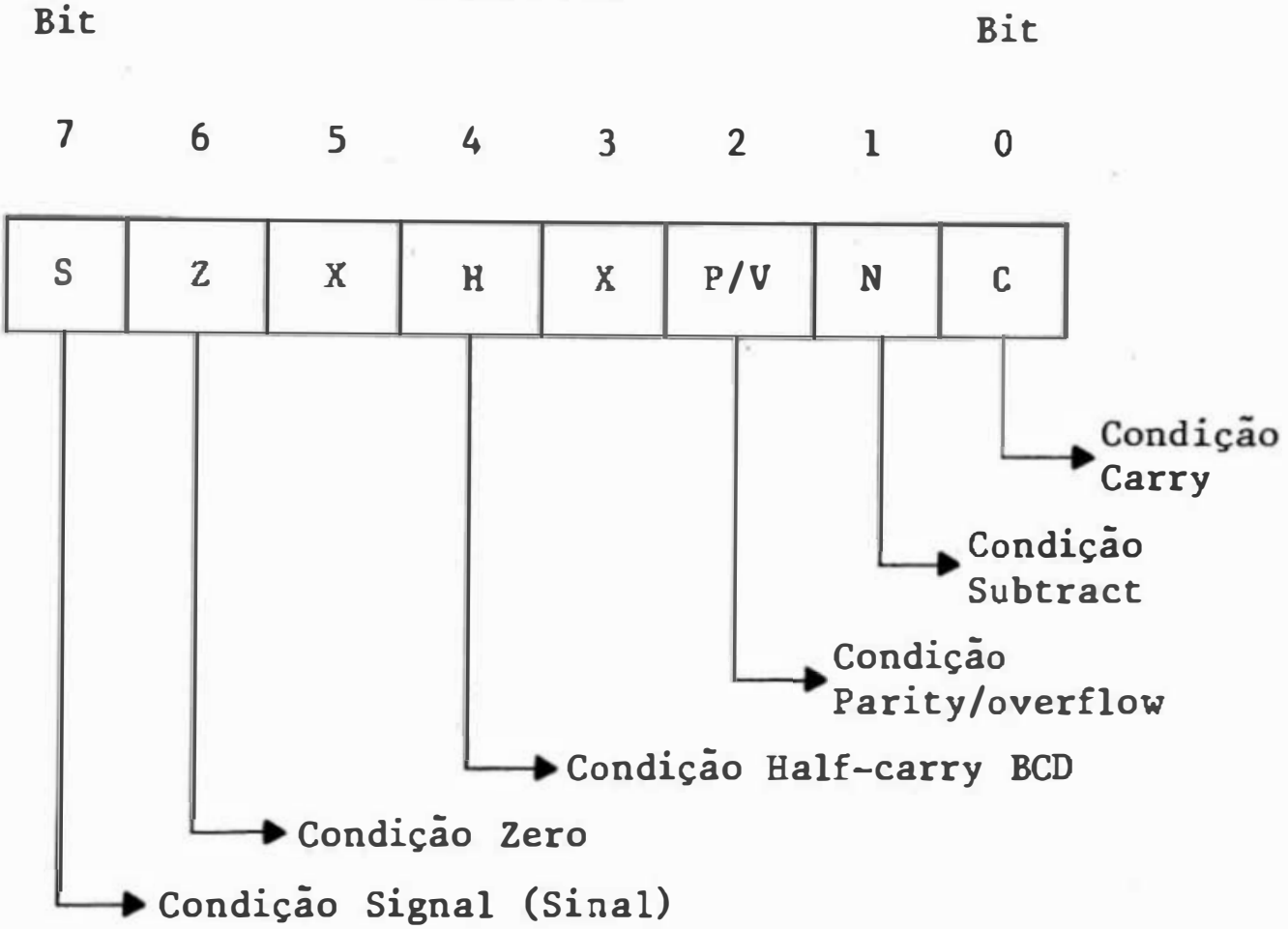
Como nos mostra a figura 9, o Z-80 possui dois registros de condição F e F', sendo que um deles é ativo e o outro, passivo.

Cada um desses registros possui 8 bits, agrupados convenientemente. Cada um desses bits especificam uma situação definida, ou seja, especificam várias condições da UCP que ocorrem após uma operação de lógica e aritmética ou após uma outra operação da UCP. São os seguintes os bits de condição dos registros F e F':

- Bit C (Carry bit)
- Bit N (Subtract bit)
- Bit P/V (Parity/overflow bit).

A figura 16 nos mostra como é o agrupamento dos 8 bits do registro "Flag", ou seja, como é o formato deste registro.

FIGURA 16



Obs.: X = não utilizado (condição indeterminada).

Resta dizer ainda que o registro F (Flag) é selecionado com o registro A (Accumulator).

UNIDADE DE CONTROLE

A cada vez que uma instrução é retirada da memória, ela é colocada no registrador de instruções, decodificada e executada.

A unidade de controle gera e alimenta todos os sinais necessários para a leitura ou gravação, fornecidos para os registros. Além disso, controla, também, a Unidade de Lógica e Aritmética (ULA) bem como fornece comandos para a lógica externa, inclusive memórias.

UNIDADE LÓGICA E ARITMÉTICA (ULA)

As instruções aritméticas e lógicas de 8 bits na UCP são executadas dentro da Unidade de Lógica e Aritmética (ULA). Nesta unidade são executados os seguintes tipos de funções: soma, subtração, lógica E, lógica OU, lógica OU Exclusivo, comparação, deslocamento para a direita ou para a esquerda, incrementação, decrementação, ligação de bits, desligamento de bits, teste de bits, etc.

As operações lógicas e aritméticas de 8 bits são realizadas, usando o acumulador ativo, outro registro ativo de uso geral ou o conteúdo de uma célula de memória. O resultado é sempre depositado no acumulador.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 · CEP 86100 · LONDRINA · PARANÁ

EXAME DA LIÇÃO MC-12

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) A versão do microprocessador Z-80 H opera com uma freqüência de:

   a) 2,5 MHz;

   b) 4,0 MHz;

   c) 6,0 MHz;

   (d) 8,0 MHz.

2) O Z-80 é tido como sendo um microprocessador de:

   a) primeira geração;

   b) segunda geração;

   (c) terceira geração;

   d) N.R.A.

3) O barramento de endereços ou barramento de direções do Z-80 é composto por:

a) 16 pinos (A_0 até A_{15});

b) 8 pinos (A_0 até D7);

c) 8 pinos (D_0 até D7);

d) N.R.A.

4) O $\overline{\text{MREQ}}$ (Memory Request) é uma saída em "três-estados" que indica:

a) a requisição da memória e em seguida leva este sinal ao barramento de endereços;

b) a colocação dos dados na memória auxiliar;

c) os itens a e b estão corretos;

d) N.R.A.

5) Os sinais $\overline{\text{RD}}$ e $\overline{\text{WR}}$, quando ativados, indicam respectivamente:

a) saída ou entrada de dados da memória;

b) entrada ou saída de dados da memória;

c) apagamento e a colocação de novos dados na memória;

d) N.R.A.

6) O sinal $\overline{\text{RFSH}}$ somente é utilizado quando trabalhamos com:

a) memórias EPROM;

b) memórias ROM;

c) memórias EAROM;

d) memórias semicondutoras do tipo RAM dinâmica.

7) O $\overline{\text{NMI}}$ somente reconhece o pedido de interrupção quando ocorrer:

a) uma transição positiva;

b) a habitação do pino interrupt enable;

c) uma transição negativa;

d) N.R.A.

8) No Z-80, os registros de uso geral são em número de:

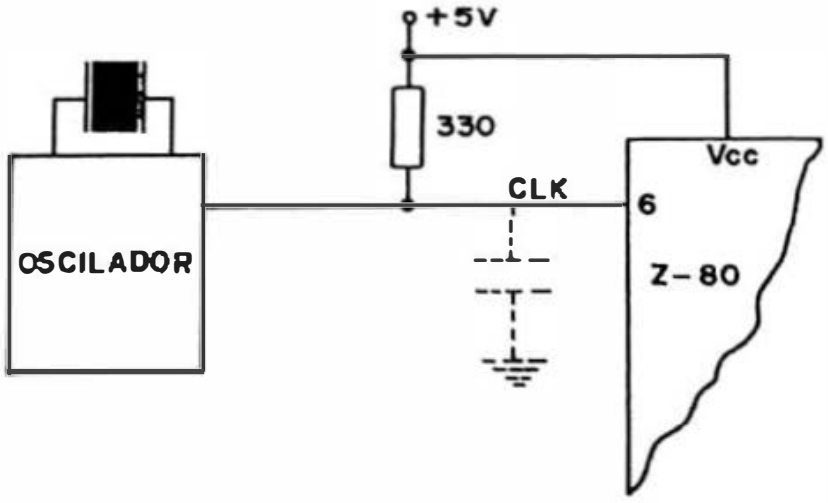
a) dez;

b) sete;

c) dezesseis;

d) quatorze. A BC D E H L
A' B' C' D' E' H' L'

9) A figura abaixo representa:



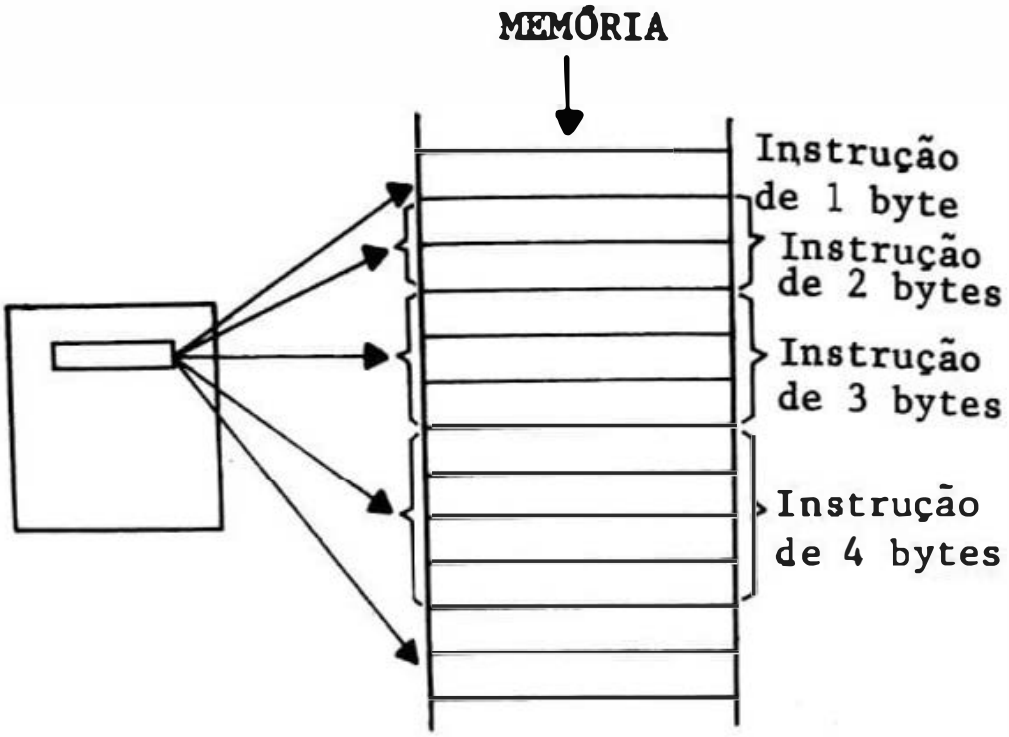
a) a saída de clock do Z-80;

b) a entrada de clock do Z-80;

c) a unidade de controle de disco;

d) N.R.A.

10) A figura abaixo representa:



a) uma ilustração do funcionamento do PC;

b) uma ilustração do funcionamento do SP;

c) uma ilustração do funcionamento de Ix e Iy;

d) N.R.A.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-13

MICROPROCESSADOR Z-80

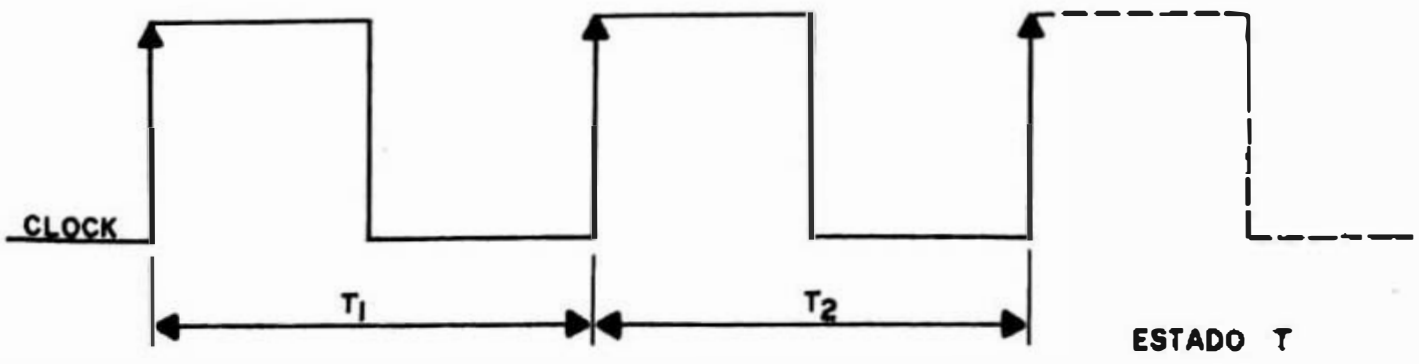
Dando continuidade à lição MC-12, na qual iniciamos o estudo do microprocessador Z-80, apresentaremos, nesta lição, alguns detalhes do seu mecanismo de sincronismo, bem como exemplos de interface para módulos de memória.

CONCEITOS BÁSICOS SOBRE O SINCRONISMO (TIMING) DO Z-80

Para podermos entender como funciona o mecanismo de sincronização do Z-80, precisaremos definir alguns conceitos importantes.

O primeiro deles é o estado T que corresponde à unidade básica de tempo, sendo esta igual a um período do sinal de clock. A figura 1 nos mostra dois estados T (T_1 e T_2).

FIGURA 01



O estado T também é conhecido por ciclo de clock ou simplesmente ciclo T.

Como sabemos, o período é igual ao inverso da freqüência, ou seja:

$$T = \frac{1}{f}$$

Portanto, para um clock de 2,5 MHz que, no caso, corresponde ao clock do Z-80, teremos um período (ou ciclo T) igual a:

$$T = \frac{1}{2,5 \times 10^6} \qquad T = 400 \text{ ns (ou } 400 \times 10^{-9} \text{ segundos)}$$

Analogamente, para a versão Z-80A, o período (ou ciclo T) será igual a 250 ns.

Os ciclos T são utilizados para controlar operações dentro de um ciclo maior chamado "ciclo de máquina" ou simplesmente "ciclo M". Toda instrução executada pelo Z-80 possui de um a seis (1 à 6) ciclos de máquina (com exceção das instruções relacionadas com o bloco especial) e cada um dos ciclos de máquina, por sua vez, possui de três a seis (3 à 6) ciclos de clock (ciclo T). A figura 2 nos mostra os ciclos M e os ciclos T de uma instrução do microprocessador Z-80A.

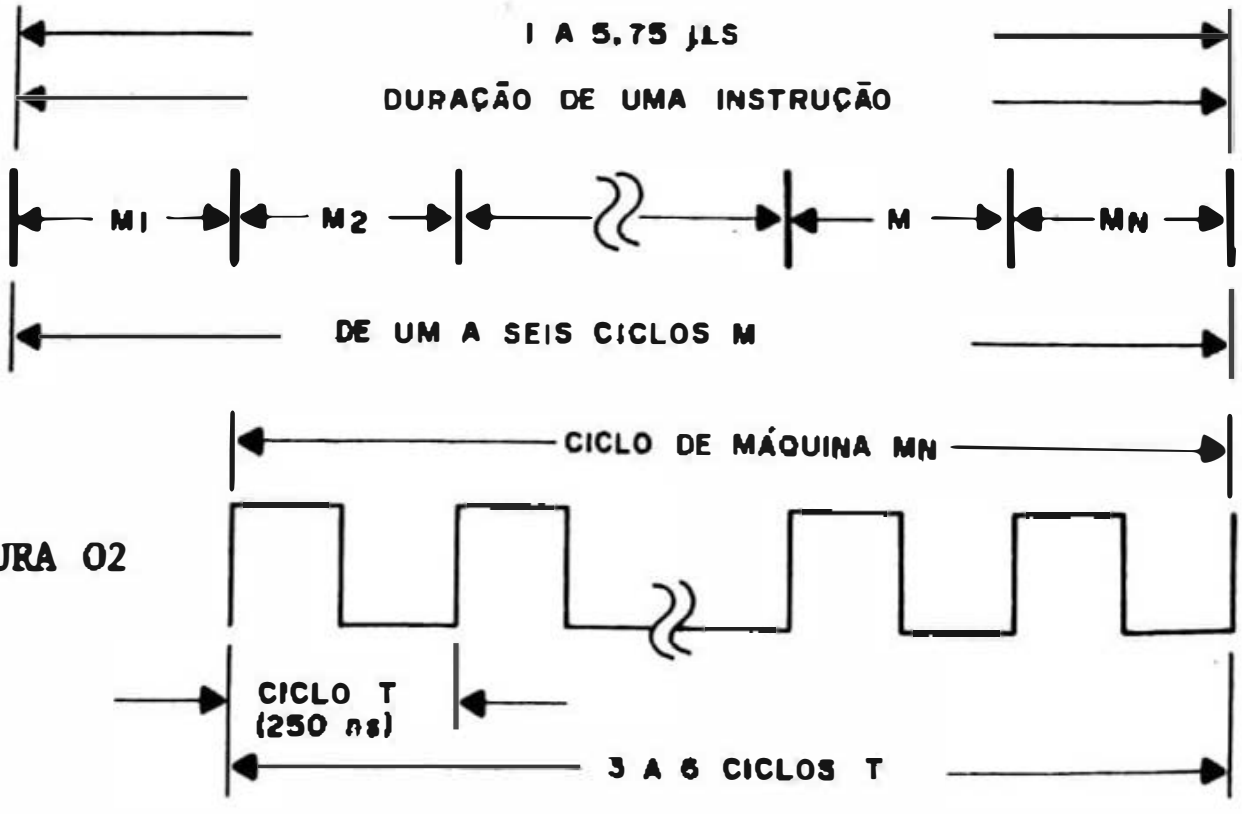


FIGURA 02

Obs.: os tempos indicados nesta figura são para o Z-80A.

Toda instrução executada pelo Z-80 segue um conjunto básico e preciso de operações:

1 - Escrever ou ler a memória;
2 - Escrever ou ler através do dispositivo de E/S;
3 - Interrupção "reconhecida".

Podemos então dizer que todas as instruções nada mais são do que uma série destas operações básicas, sendo que cada uma delas pode ter de 3 a 6 ciclos T (ou períodos de clock) a ser completada ou ainda aumentada com o intuito de sincronizar a UCP, com relação à velocidade dos dispositivos externos.

Durante a operação do microprocessador Z-80 pode ocorrer sete (7) tipos de ciclos de máquina básicos:

1 - Ciclo de busca do código de operação ou ciclo de busca de instrução (M1);
2 - Ciclo de leitura e escrita de dados na memória:
3 - Ciclo de leitura e escrita em dispositivos de E/S;
4 - Ciclo de requisição/admissão de controle dos barramentos ($\overline{BUSREQ}$);
5 - Ciclo de Interrupção $\overline{INT}$;
6 - Ciclo de Interrupção NMI;
7 - Ciclo de escape da instrução HALT.

Antes de iniciarmos o estudo de cada um dos sete (7) tipos de ciclos de máquina básicos, cabe observar que o número de ciclos de máquina não é fixo para as instruções de pesquisa e movimentação de blocos.

Vejamos agora cada um dos sete ciclos de máquina básicos

1 - CICLO DE BUSCA DO CÓDIGO DE OPERAÇÃO (M_1)

Na execução de qualquer instrução, existe um ciclo de busca do código de operação (M_1) que se encontra armazenado na memória. Existem casos de instrução que possuem 2 bytes para o código de operação, portanto, possuem 2 ciclos M_1.

O ciclo M1 permite à UCP realizar três tarefas básicas:

a) ler o byte do código de operação da memória externa, colocando-o no registro de instrução do Z-80;
b) decodificar o código de operação, identificando a instrução a ser executada;
c) implementar uma parte ou possivelmente (no caso de pe

quenas instruções de somente um ciclo de máquina) toda a operação.

A figura 3 nos mostra o diagrama de sincronização para uma instrução INC R, que é uma instrução que possui somente o ciclo M_1, composto por quatro ciclos T (T_1, T_2, T_3 e T_4).

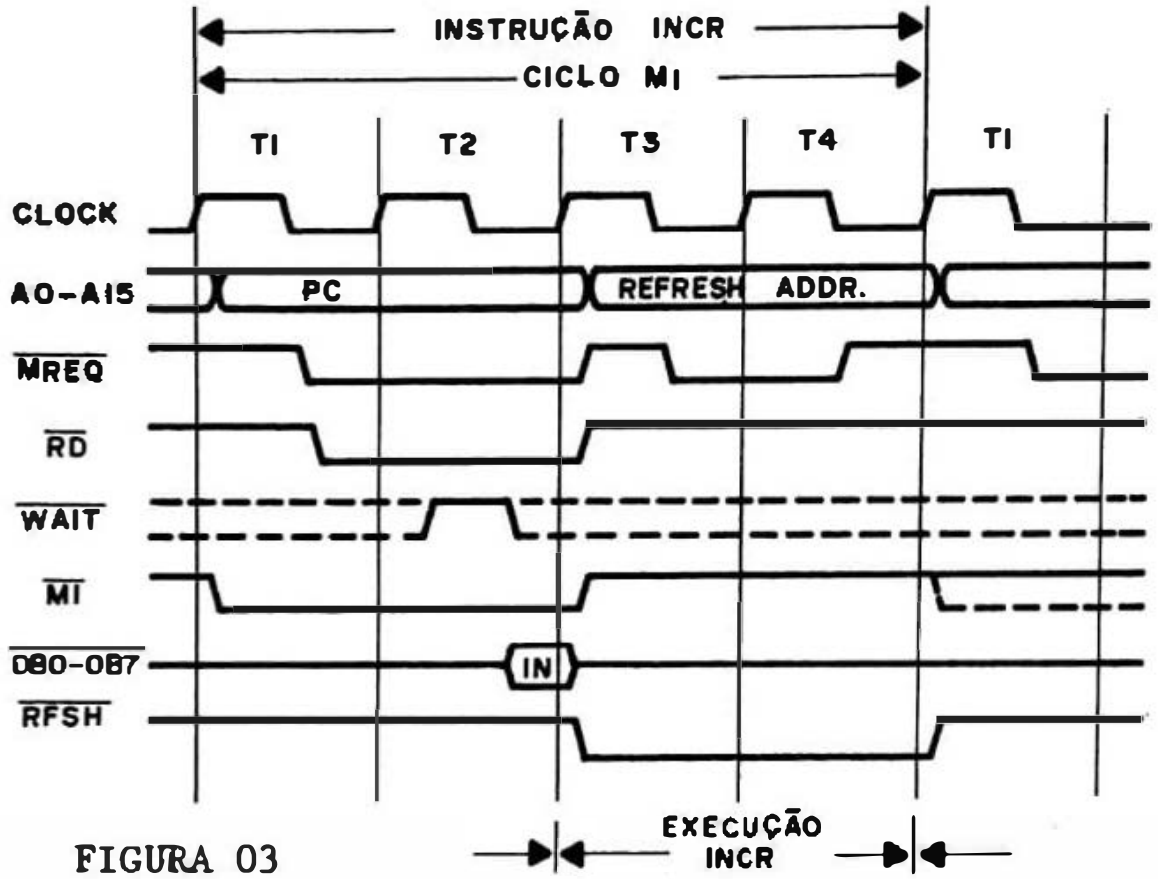


FIGURA 03

Observando-se a figura 3, podemos notar que à medida que a UCP introduz o ciclo M_1, o sinal M_1 vai para o nível lógico "0", indicando ao meio externo que este ciclo (ciclo M_1) está ativado. O conteúdo do contador de programa (PC) é colocado no barramento de endereço, de modo a preparar para a busca do código de operação (OP) da próxima instrução. No meio de T_1, isto é, na descida do sinal de clock (⌐↓), os sinais $\overline{MREQ}$ e $\overline{RD}$ são levados para o nível lógico "0", indicando que o Z-80 já colocou o endereço no barramento e vai realizar uma leitura da memória. A memória externa irá então colocar no barramento de dados o conteúdo da posição endereçada pelo Z-80, isso, no máximo, até o início da subida do ciclo T_3 que é caracterizado pela subida do sinal de clock (CLK). Neste instante (início de T_3) o byte do código de operação, que estava no barramento de dados, é colocado no registro de instrução do Z-80, ou seja, (de um modo mais amplo) na UCP do Z-80. Logo, em seguida, os sinais $\overline{M_1}$, $\overline{RD}$ e $\overline{MREQ}$ vão para o nível lógico "1", que é o estado inativo destes sinais.

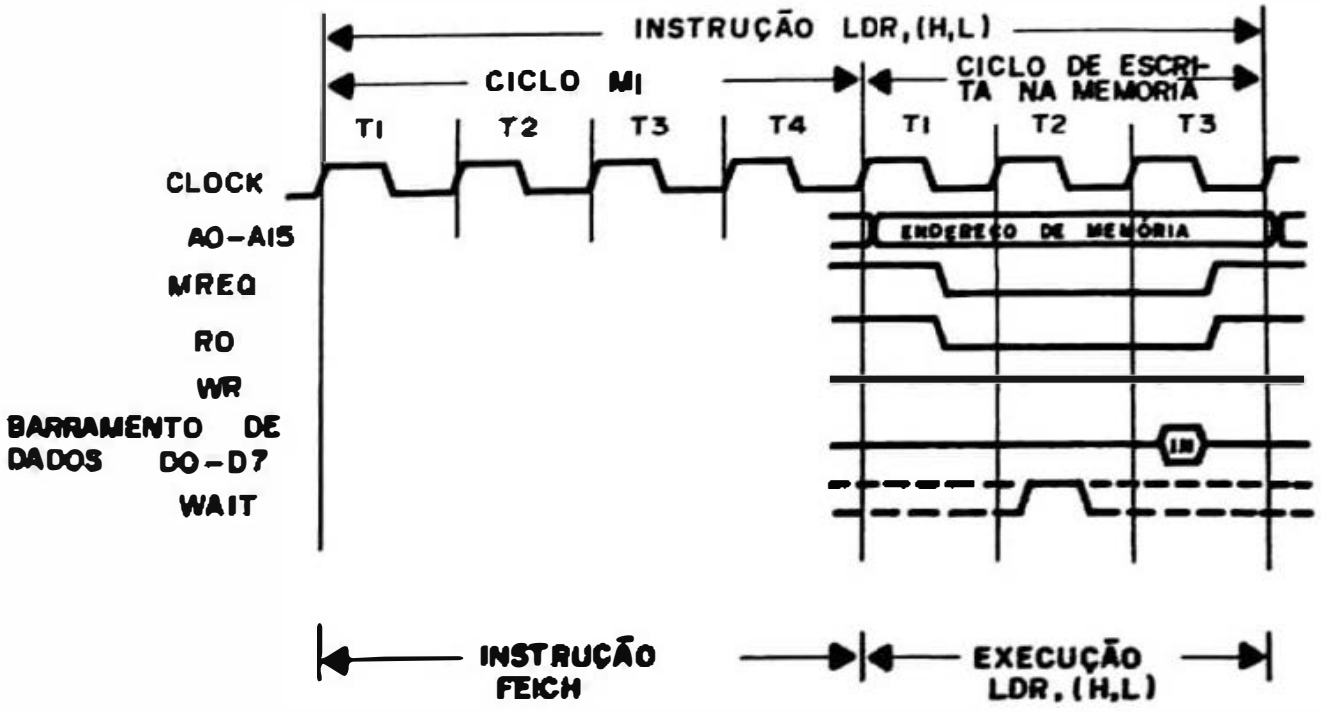
Os dois ciclos restantes T_3 e T_4 são utilizados pelo Z-80 para fornecer um tempo de refrescamento para as memórias dinâmicas externas. O sinal RFSH é, então, levado para o nível lógico "0" (ativado) e, juntamente, com o sinal $\overline{MREQ}$ que também é ativado neste instante, indicam para a memória dinâmica externa que um refrescamento ou, simplesmente, refresh, irá acontecer. Nesse mesmo instante, o Z-80 informa que o conteúdo do registro R encontra-se no barramento de endereço. Enquanto o microprocessador Z-80 encontra-se nesses dois ciclos de clock, T_3 e T_4, ele executará simultâneamente operações internas de decodificação e execução da instrução INC R, utilizando o barramento de endereços para realizar o refresh das memórias dinâmicas.

2 - CICLO DE LEITURA E ESCRITA DE DADOS NA MEMÓRIA:

Para facilitar a compreensão deste ciclo de máquina, lançaremos mão de dois exemplos de instruções do Z-80.

A figura 4 nos mostra a execução de uma instrução LDR, (H,L), a qual carrega os conteúdos da posição de memória indicada pelo par de registro H,L para o registro R da UCP.

FIGURA 04



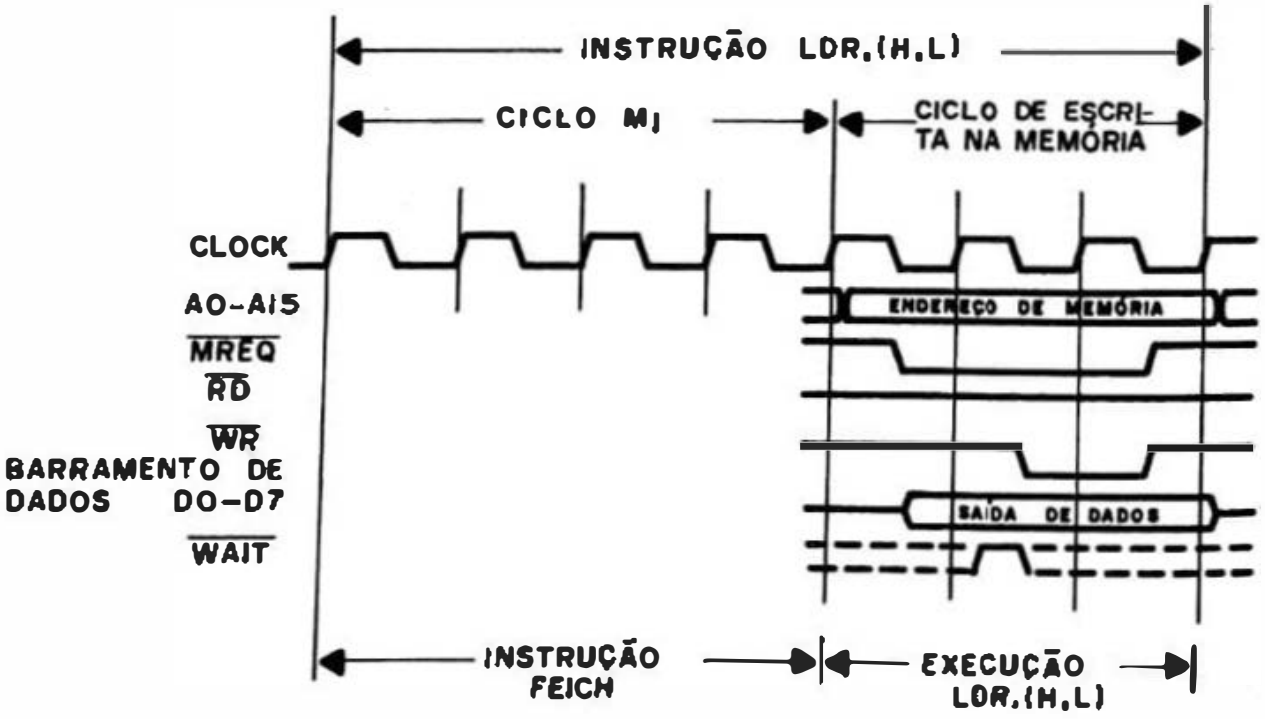
O ciclo de máquina M_1 é idêntico ao que acabamos de descrever. No final do ciclo M_1, a UCP decodifica o código de operação e inicia um ciclo de leitura do operando de 8 bits que se encontra na memória. Assim, como no caso do ciclo M_1, os sinais $\overline{MREQ}$, $\overline{RD}$ e o barramento de endereço são ativados, ou seja, no ciclo T_1.

Desta forma, o barramento de endereço retém o conteúdo do par de registros H,L e durante este tempo a memória externa fornece, via barramento de dados, o operando que deverá ser carregado no registro R da UCP. Isto é feito no meio do estado T_3, ou seja, na descida do (⌐↓¬) do sinal de clock. Note bem que, neste caso, o registro R somente é carregado na "borda" de descida de T_3.

Estes ciclos são, geralmente, compostos de 7 ciclos de clock, sendo que quatro (4) deles no ciclo M1 e três (3) no ciclo de leitura da memória.

A figura 5 nos mostra um ciclo de escrita na memória. Neste caso, é a instrução LD(HL)R que carrega o conteúdo do registro R na posição de memória, cujo endereço é igual ao conteúdo do par de registros H,L.

FIGURA 05



As saídas do barramento de endereço e o sinal $\overline{MREQ}$ também são ativados como no exemplo que acabamos de expor. Nenhum sinal $\overline{RD}$ é fornecido, mas o conteúdo do registro da UCP especificado é conectado ao barramento de dados depois da descida do ciclo T_1 (↓). Este dado permanece no barramento de dados e na borda de descida do ciclo T_2, o sinal $\overline{WR}$ é então ativado. Com $\overline{MREQ}$ e $\overline{WR}$ ativados, a memória externa escreve o dado do barramento de dados na posição de memória especificada, utilizando para tanto, as saídas do barramento de endereço (A_0 -A_{15}).

Observando as figuras 4 e 5, podemos verificar que nos ciclos estudados, o sinal de $\overline{WAIT}$ foi considerado no nível lógico "1", durante a descida de clock do ciclo T_2. Neste instante, o Z-80 realiza um teste com a entrada $\overline{WAIT}$. Se ela estiver em um nível lógico baixo ("0"), entende-se que há estados de espera TW, isto até que o sinal de $\overline{WAIT}$ volte para o nível lógico "1". A figura 6 nos mostra como uma requisição $\overline{WAIT}$ pode alterar o tempo de uma leitura ou gravação.

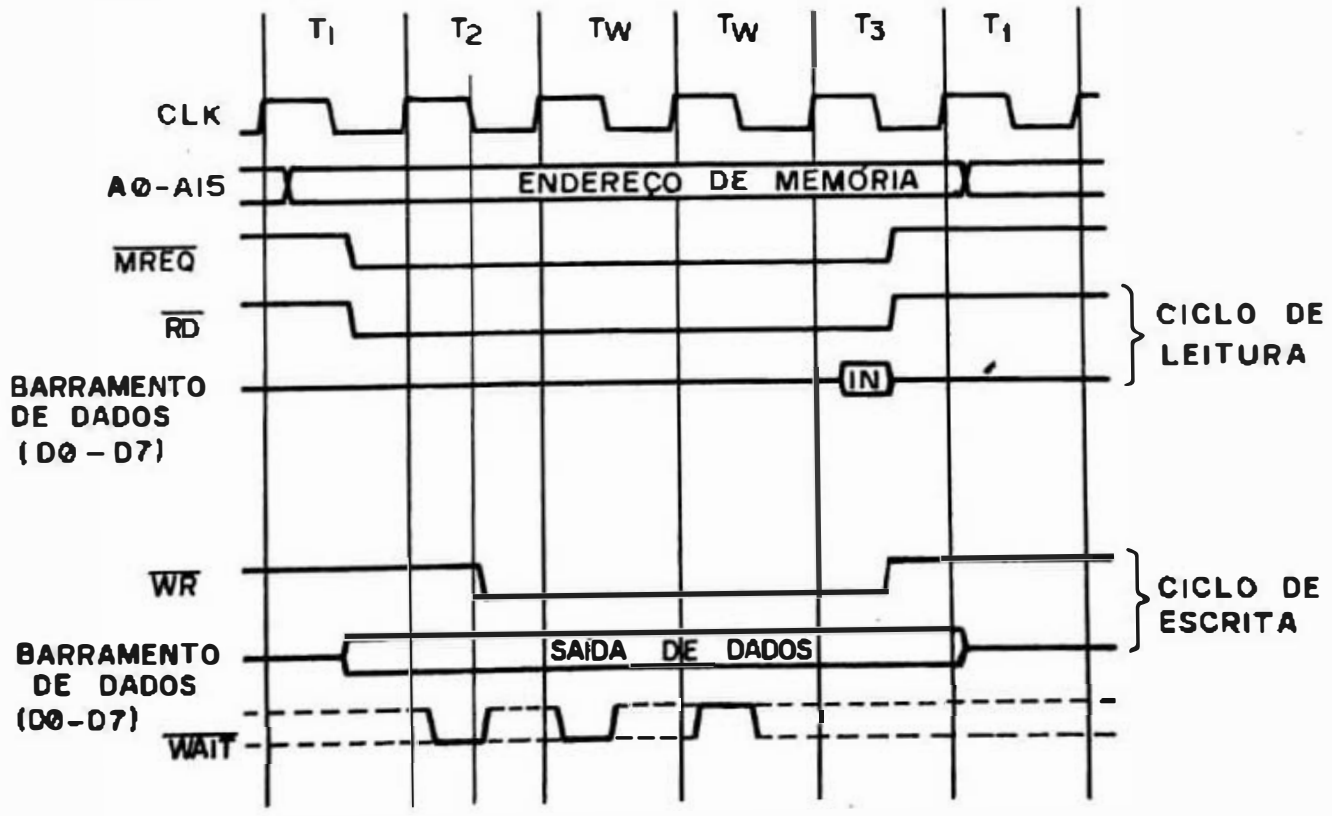


FIGURA 06

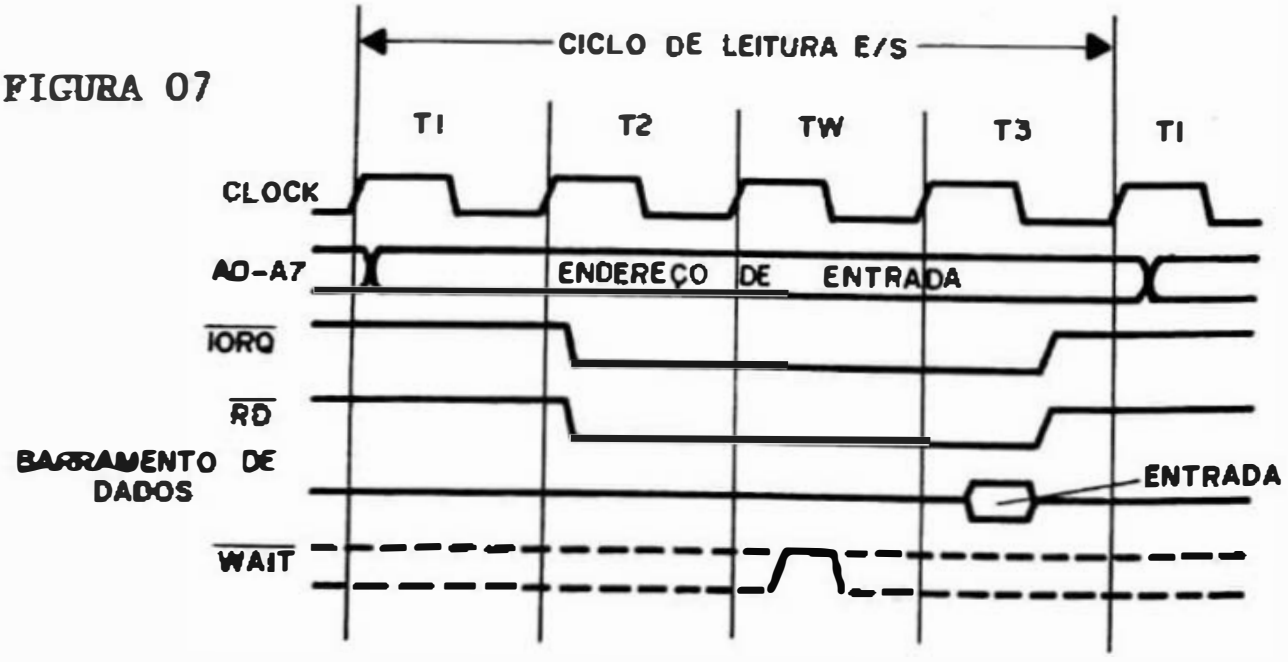
Note que nunca pode ocorrer um ciclo de leitura e escrita simultaneamente.

3 - CICLO DE LEITURA E ESCRITA EM DISPOSITIVOS DE E/S:

Um ciclo de entrada de dados ocorre quando o microprocessador Z-80 executa uma instrução de leitura de dados de um dispositivo de entrada (pode ser um periférico de entrada). Já um ciclo de saída de dados para o Z-80, ocorre quando este executa uma instrução de escrita de dados em um dispositivo de saída. As instruções de entrada e saída de dados, geralmente, possuem três ou quatro ciclos de máquina, com um total de 10 a 20 ciclos T (2,5 a 5 microssegundos para um clock de 4 MHz). As instruções de entrada e saída (E/S) de blocos de dados (INIR, INDR, OTIR, OTDR), que transferem até 256 bytes, executam repetidas vezes os ciclos de máquina, até que todos os bytes sejam transferidos. Desta forma, o tempo total de execução destas instruções irá depender do número de bytes a ser transferido e da velocidade dos dispositivos de entrada e saída. Na figura 7, temos a representação de um ciclo de entrada e, na figura 8, a representação de um ciclo de saída.

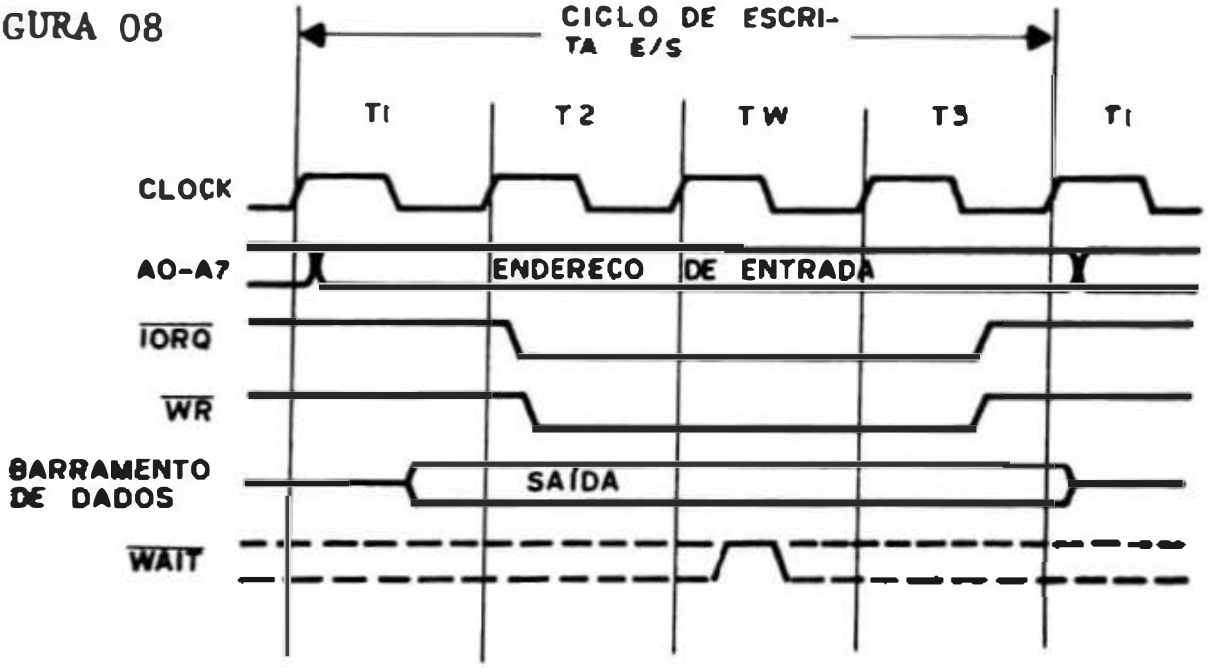
O endereço do dispositivo de entrada e saída (E/S) é colocado nas linhas A_0-A_7 do barramento de endereços no início do ciclo de máquina, ou seja, no ciclo T_1. No início do ciclo T_2, mais precisamente, logo em seguida à borda de subida de T_2, o sinal $\overline{IORQ}$ é ativado ($\overline{IORQ} = 0$), indicando que teremos uma leitura ou escrita em um dispositivo de entrada e saída.

FIGURA 07



No caso de termos uma leitura de um dispositivo de entrada, o sinal $\overline{RD}$ é ativado ($\overline{RD}$ = 0) no mesmo instante em que é ativado o sinal $\overline{IORQ}$. O controlador do dispositivo externo reconhece o pedido de leitura através dos sinais $\overline{IORQ}$ e $\overline{RD}$, quando ambos estiverem simultaneamente no nível lógico "0", colocando a informação no barramento de dados. Na borda de descida do ciclo T_3, esta informação é lida pelo Z-80, ou seja, é introduzida na UCP do microprocessador Z-80.

FIGURA 08



Se uma operação de escrita está ocorrendo (veja figura 8) o sinal $\overline{WR}$ é ativado no lugar do sinal $\overline{RD}$, ou seja, o sinal $\overline{WR}$ é ativado ($\overline{WR}$ = 0) no mesmo instante em que é ativado o sinal $\overline{IORQ}$. Note que, já no ciclo T_1, o Z-80 coloca no barramento de dados o conteúdo a ser escrito no dispositivo de saída. Este dado (ou informação) está disponível durante todo do ciclo de escrita, ou seja, até o final do ciclo T_3, devendo ser introduzido, durante este período, em algum lugar do dispositivo de saída.

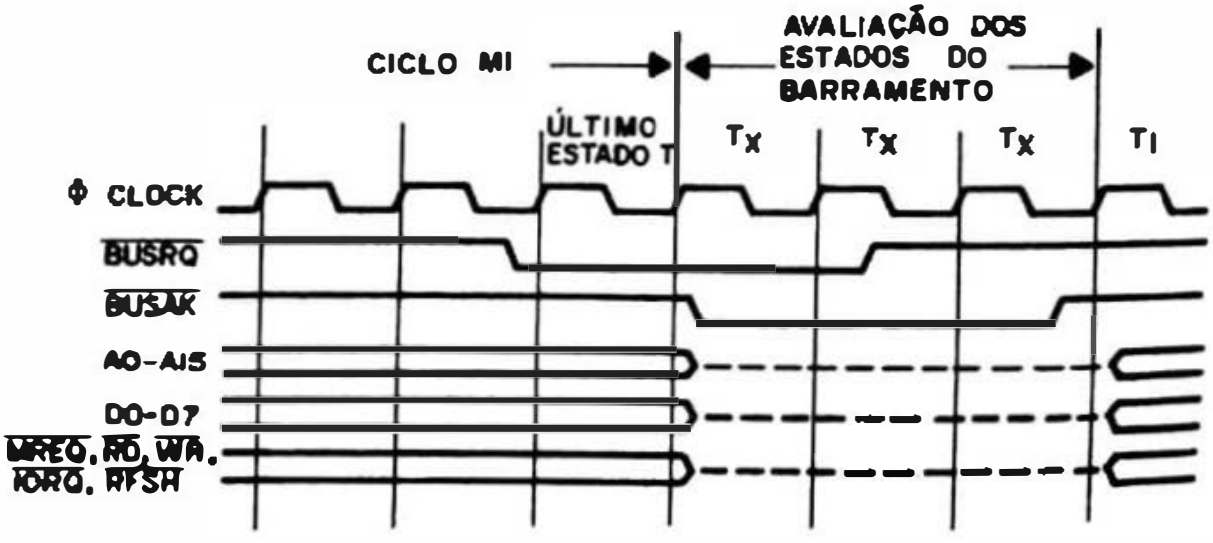
Note que, para ambos os ciclos de entrada e saída, o sinal $\overline{WAIT}$ é ativado internamente, após o ciclo T_2. Portanto, teremos um ciclo TW adicional, se a entrada $\overline{WAIT}$ estiver no nível lógico "1". Isto leva a UCP a interromper o processamento até que a entrada $\overline{WAIT}$ seja desativada novamente. Esta condição é implementada para dar à UCP um certo tempo adicional para "atender" os dispositivos de entrada e saída mais vaga-

rosos. Por exemplo, no caso de termos um dispositivo E/S lento, deve-se manter a entrada $\overline{WAIT}$ no nível lógico "0", por um certo tempo, permitindo assim a resposta deste dispositivo de entrada e saída. Isto é possível devido ao fato de que enquanto a entrada $\overline{WAIT}$ estiver no nível lógico "0", o Z-80 permanecerá no estado de espera.

4 - CICLO DE REQUISIÇÃO/ADMISSÃO DE CONTROLE DOS BARRAMENTOS (BUSRQ):

Em qualquer instante, um dispositivo externo pode requerer o controle do barramento de endereços (ou de endereçamento) A_0-A_{15}, do barramento de dados D_0-D_7 e das linhas $\overline{MREQ}$, $\overline{RD}$, $\overline{WR}$, $\overline{IORQ}$ e $\overline{RFSH}$, bastando, para tanto, ativar o sinal de entrada $\overline{BUSRQ}$ ($\overline{BUSRQ}$ = 0). A explicação da necessidade deste ciclo de máquina está no fato desta entrada, quando ativada, permitir que um controlador de dispositivo externo comunique-se diretamente com a memória externa para a transferência de dados entre dispositivos de entrada e saída de alta-velocidade e a memória, sem a interferência da UCP, ou seja, do microprocessador (acesso direto à memória - DMA). A figura 9 nos mostra a representação de um ciclo de requisição/admissão de controle dos barramentos.

FIGURA 09



O microprocessador Z-80 realiza uma leitura da linha $\overline{BUSRQ}$ no último ciclo T de cada ciclo de máquina. A UCP detecta o sinal $\overline{BUSRQ}$, quando este estiver ativado, na borda de subida do último ciclo T. O ciclo T é então completado. Se $\overline{BUSRQ}$ for igual a zero, ele ativa no próximo ciclo T a saída BUSAK (BUSAK = 0), indicando, desta forma, ao meio externo, que reconhece o pedido de controle dos barramentos. Neste mesmo instante, o Z-80 leva o barramento de endereços, o barramento de dados e as linhas $\overline{MREQ}$, $\overline{RD}$, $\overline{WR}$, $\overline{IORQ}$ e $\overline{RFSH}$, para a posição de alta impedância em três estados (three-states). Assim, o dispositivo externo pode atuar nestes barramentos e nestas linhas de controle, para realizar a transferência de dados (ou informações) com a memória sem interferência da UCP. A linha $\overline{BUSRQ}$ é testada, continuamente, nas subidas do sinal de clock.

Quando o controlador do dispositivo de entrada e saída completar a transferência DMA, ele irá desativar a linha $\overline{BUSRQ}$ ($\overline{BUSRQ}$ = 1). Esta condição será detectada pelo Z-80 na próxima borda de subida de um ciclo T. Então, com isto, a linha $\overline{BUSAK}$ será desativada ($\overline{BUSAK}$ = 1) na borda de subida do ciclo T seguinte. Teremos, assim, o reinício do processamento e os barramentos e as linhas de controle voltam a ser comandadas pela UCP. O processamento é retomado a partir do ponto, no qual ocorreu a requisição do barramento, através da entrada (ou linha) $\overline{BUSRQ}$.

5 - CICLO DE INTERRUPÇÃO ($\overline{INT}$):

A UCP somente irá permitir interrupções externas, se a linha $\overline{BUSRQ}$ estiver desativada e se o Z-80 estiver no modo ENABLE INTERRUPT, que é ativado por software, através da instrução EI.

Um dispositivo externo faz a requisição de interrupção para ativar o sinal $\overline{INT}$ ($\overline{INT}$ = 0). Durante a borda de subida do último ciclo T do último ciclo de máquina de uma dada instrução, a UCP realiza a leitura da entrada $\overline{INT}$, e, se esta estiver em um nível lógico "0", inicia-se, neste mesmo instante, um ciclo de interrupção. A figura 10 nos mostra a representação de um ciclo de interrupção $\overline{INT}$.

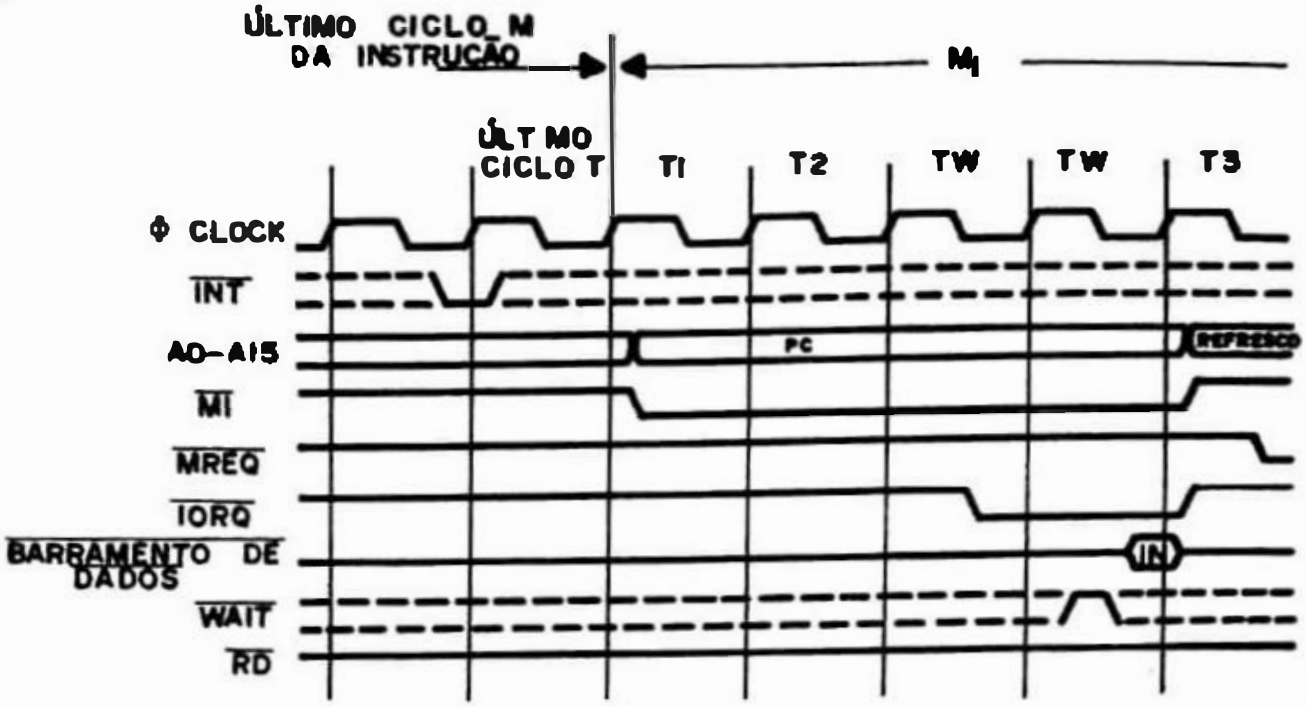


FIGURA 10

No ciclo T_1 do ciclo de interrupção, a saída $\overline{M_1}$ é ativada ($\overline{M_1} = 0$). Temos, além do ciclo T_2, dois ciclos TW (ou dois estados WAIT) fornecidos para dar tempo suficiente para o circuito de interrupção (DAISY-CHAINED) responder à admissão da interrupção e colocar uma resposta ao pedido de interrupção no barramento de dados.

A lógica externa de interrupção identifica o reconhecimento da interrupção, quando temos as entradas $\overline{M_1}$ e $\overline{IORQ}$ ativadas, ou seja, $\overline{M_1} = \overline{IORQ} = 0$. Após o reconhecimento da interrupção, mais precisamente, no início do ciclo T_3, o circuito externo coloca o dado apropriado no barramento de dados, o qual será, então, lido pelo Z-80 no início deste ciclo (ciclo T_3). Ainda, durante esse mesmo ciclo, as entradas $\overline{M_1}$ e $\overline{IORQ}$ são desativadas e o Z-80 inicia o procedimento de refresco.

6 - CICLO DE INTERRUPÇÃO $\overline{NMI}$:

A figura 11 nos mostra a ação da UCP durante este ciclo de máquina.

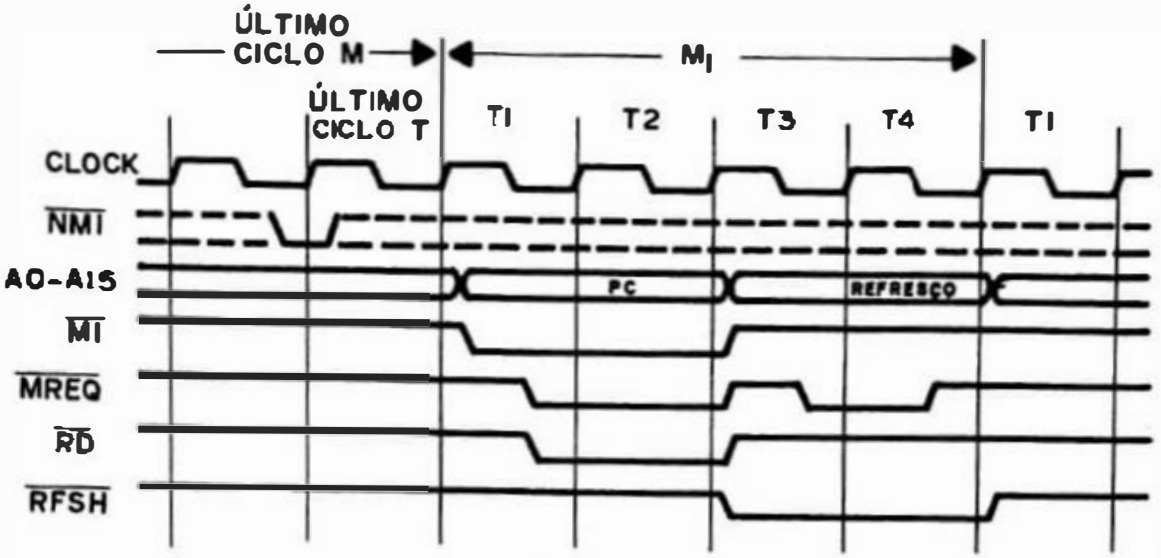


FIGURA 11

Um pedido de interrupção, através da entrada $\overline{NMI}$, não pode ser inibido por meio de instruções. Além disso, o pedido de interrupção, através desta entrada, tem prioridade sobre a interrupção externa, ou seja, tem prioridade sobre pedidos de interrupção oriundos da entrada $\overline{INT}$.

Este sinal é reconhecido durante o último ciclo T do último ciclo de máquina da instrução corrente, de forma análoga ao caso da interrupção externa ($\overline{INT}$), apresentado anteriormente. A figura 11 nos mostra apenas uma parte desta ação de interrupção. Neste caso, o sinal $\overline{IORQ}$ não é ativado, pois torna-se desnecessário informar ao dispositivo externo se a interrupção foi aceita. O primeiro ciclo de máquina é semelhante a uma operação de leitura de memória, com a exceção de que neste primeiro ciclo de máquina, não se realiza nenhuma leitura de dado, ou seja, nenhum dado é lido pelo Z-80.

As operações de refrescamento, ou seja, de "refresh" são transportadas para os ciclos T_3 e T_4, quando as linhas $\overline{RFSH}$ e $\overline{MREQ}$ são ativadas e os conteúdos do registro R são colocados no barramento de endereços A_0-A_7.

7 - CICLO DE ESCAPE DA INSTRUÇÃO HALT:

Quando uma instrução HALT é executada, o sinal $\overline{HALT}$ é ativado automaticamente pela UCP. O microprocessador Z-80 gera, continuamente, ciclos M_1 para este ciclo HALT e não avança o contador de programa (PC). Isto significa que, mesmo parado, temos ciclos M_1 que, por sua vez, realizam o refresca-

mento (refresh) das memórias dinâmicas.

A figura 12 nos mostra um ciclo de escape da instrução HALT.

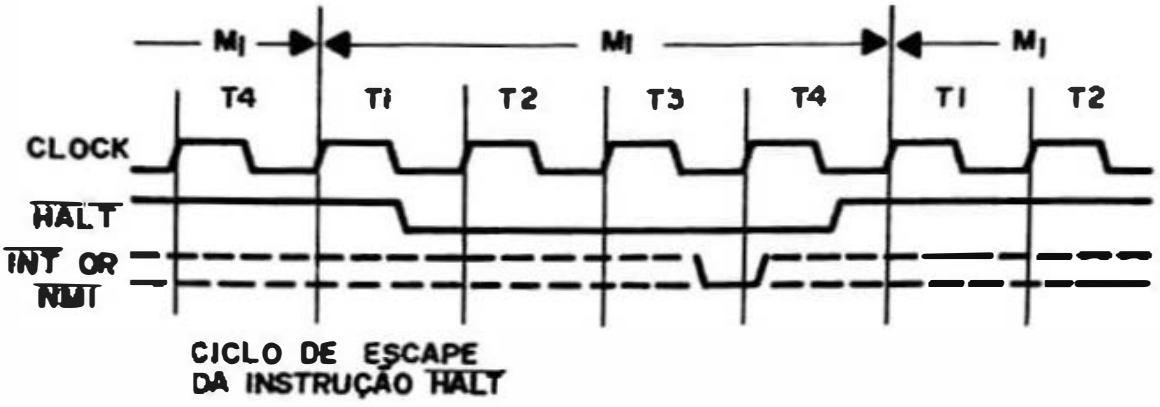


FIGURA 12

A lógica de refrescamento das memórias dinâmicas é ativada durante os últimos ciclos T de um ciclo M_1.

O ciclo $\overline{\text{HALT}}$ somente pode ser interrompido por um $\overline{\text{RESET}}$ ou por sinais externos de interrupção ou ainda por um sinal $\overline{\text{NMI}}$. Esses sinais provocam a interrupção do processamento normal, e, como já foi dito anteriormente, fazem com que a UCP avance o contador de programa (PC) para a próxima instrução, antes que o conteúdo do contador de programa seja armazenado na pilha (STACK) da memória. De uma forma mais resumida, podemos dizer que o ciclo HALT somente pode ser interrompido pelo RESET ou por pedidos de interrupção (externo e NMI). Quando isto ocorre, a saída HALT é desativada e o Z-80 inicia um novo ciclo de máquina.

Depois de estudados os principais ciclos de maquina para o microprocessador Z-80, vamos, agora, estudar as técnicas de entrada e saída (E/S).

TÉCNICAS DE E/S:

Podemos definir um microcomputador como sendo: "todo complexo controlado por um microprocessador", tais como: memórias externas, circuitos de controle de entrada e saída, im-

pressoras, flopy disks, teclados, displays etc. Note que há necessidade de se ter um bom controle sobre este complexo e foi para auxiliar este controle que se desenvolveram as técnicas de E/S. As técnicas de E/S permitem ao microprocessador um controle mais eficiente, de modo a não prejudicar o desempenho dos principais objetivos do sistema. Por outro lado, se as técnicas de E/S não forem adequadamente escolhidas, essas vem contribuir como um fator de degradação para todo o sistema.

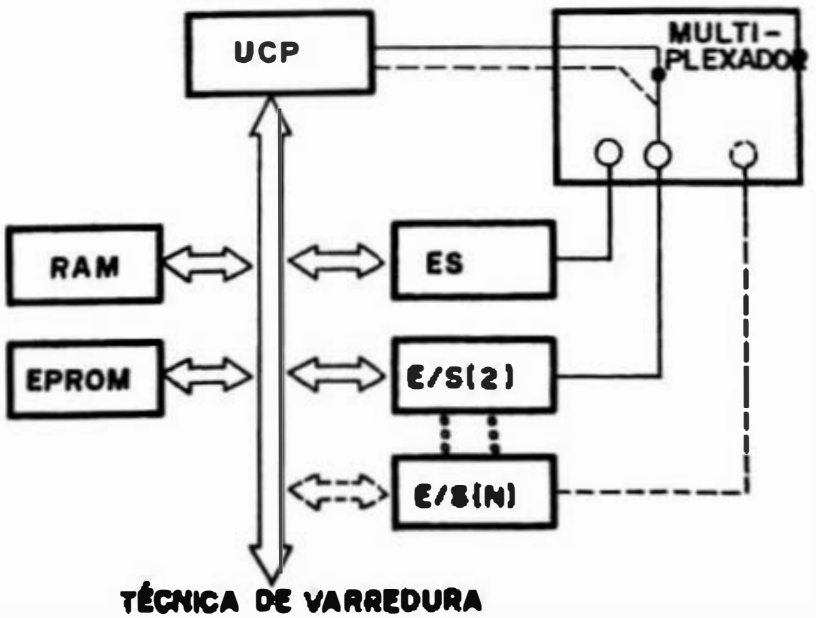
Hoje em dia, são utilizadas duas técnicas de E/S, que são:

- Técnica de varredura.
- Técnica de interrupção.

Vejamos então, cada uma delas.

TÉCNICA DE VARREDURA

Também conhecida por "POLLING" ou "SCANNING", nesta técnica, o microprocessador Z-80 testa por software, em seqüência, cada dispositivo de entrada e saída (E/S). Basicamente, este teste consiste em verificar se o dispositivo de E/S necessita de alguma tarefa do microprocessador. Uma desvantagem apresentada por esta técnica está no fato de que uma grande parte do tempo é perdida nesta observação contínua dos dispositivos de E/S. Note que no caso de um elevado número destes dispositivos de E/S, teremos uma degradação do desempenho de todo o sistema, degradação esta que se tornará mais crítica ainda em sistemas de tempo real. A figura 13 nos mostra uma ilustração a respeito desta técnica.



TÉCNICA DE VARREDURA

FIGURA 13

Esta técnica é muito utilizada em pequenos sistemas.

TÉCNICA DE INTERRUPÇÃO

Nesta técnica o microprocessador executa o programa principal, normalmente, e, quando existir uma tarefa a ser realizada por um dispositivo de E/S, o microprocessador recebe um pedido de interrupção, o qual interromperá a execução do programa principal, momentaneamente, para que então seja realizada a tarefa. Para que isto ocorra, existe no microprocessador uma entrada assíncrona que permite ao dispositivo de E/S interromper o processamento. Ao ocorrer uma interrupção o microprocessador termina a execução da instrução que está em andamento e, em seguida, inicia uma nova rotina, a qual irá realizar as tarefas solicitadas pelo dispositivo de E/S. Uma vez realizada a tarefa e encerrada esta rotina, o microprocessador prossegue a execução do programa principal, retomando-o a partir do ponto, onde ocorreu a interrupção do mesmo.

Caso o número de dispositivos de E/S seja superior ao número de entradas do microprocessador, torna-se necessário o desenvolvimento de um hardware ou de um software para auxiliar no processo de identificação da origem da interrupção. Neste caso, uma saída não muito complicada seria a aplicação da técnica de varredura, ou seja. a pesquisa por software dos dispositivos de E/S. Uma segunda solução seria a aplicação de um hardware que forneça ao microprocessador uma informação denominada vetor, através da qual seja possível ao Z-80 identificar o dispositivo que gerou o pedido de interrupção. Esse hardware também terá que prever uma maneira de diferenciar as interrupções geradas simultaneamente. Para que isto seja viável, esse hardware deverá, então, dispor de uma lógica capaz de escalonar prioridades. Esta técnica é denominada de técnica vetorada.

Por outro lado, dispomos ainda de uma terceira solução para este caso, que seria a aplicação das duas primeiras (técnica de varredura e técnica vetorada) técnicas misturadas. O vetor pode identificar um grupo de entradas de interrupções que, por sua vez. são identificadas uma a uma através da técnica de varredura.

Não resta a menor dúvida que a técnica de interrupção melhora em muito o desempenho dos sistemas, pois permite ao microcomputador assumir mais tarefas, porém esta técnica neces

sita de um hardware e de um software bem mais apurados do que os exigidos pela técnica de varredura.

A grande maioria dos sistemas de interrupções permitem inibir (mascarar) ou ativar (habilitar) interrupções através de software, dando, assim, liberdade ao programador de evitar interrupções durante certo tipo de processamento, tais como: loops de tempo e trecho de sistemas operacionais. As entradas de interrupções que não podem ser inibidas ou ativadas via software são denominadas de "non maskable interrupt" e servem para previnir quedas de alimentação.

Como exemplo de utilização da técnica de interrupções, mostramos na figura 14 o diagrama de blocos de um calendário controlado pelo microprocessador Z-80.

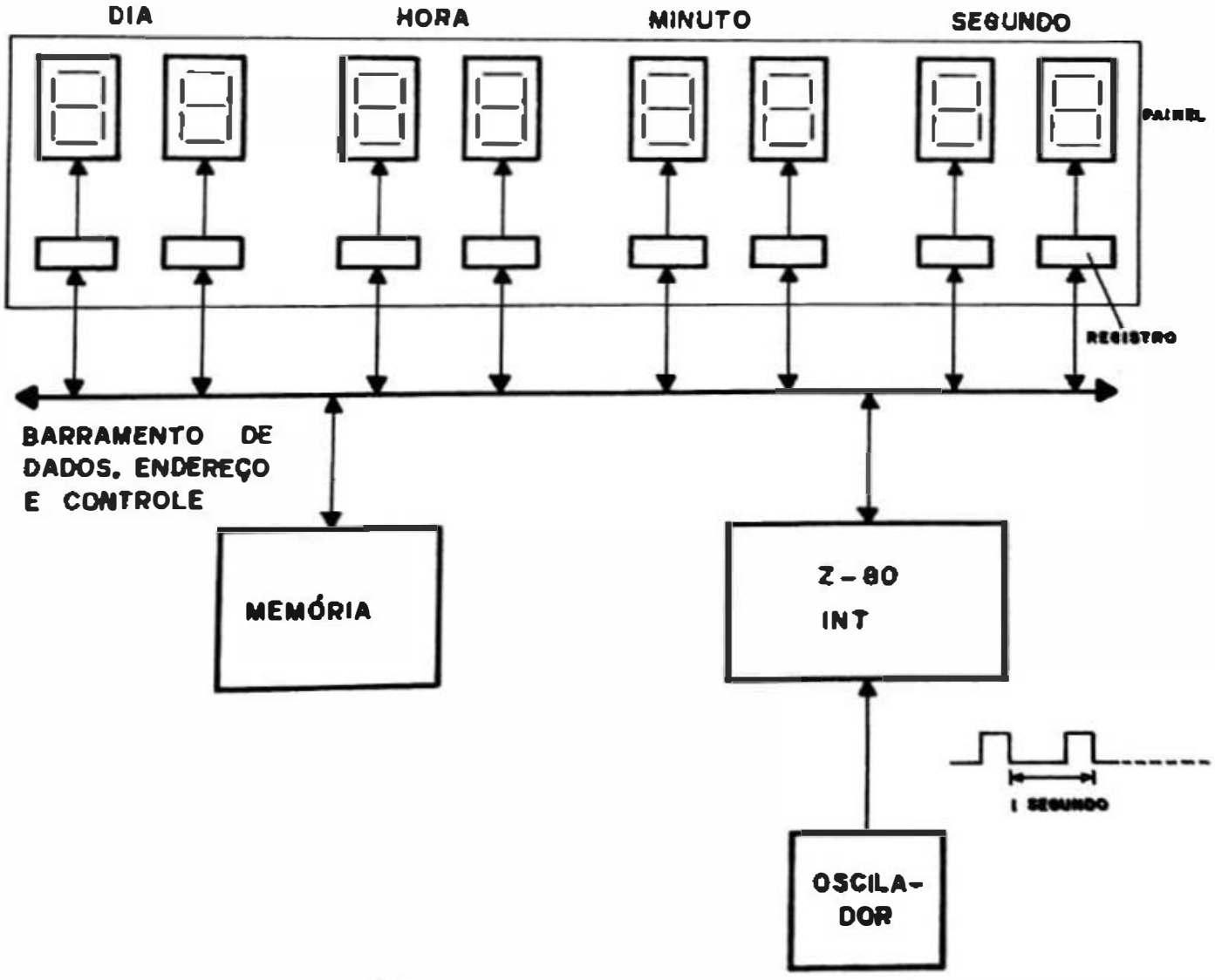


FIGURA 14

Neste caso, o relógio gera uma interrupção a cada 1 segundo, forçando o processamento de uma rotina a qual faz a atualização da data/hora. Esta rotina não leva mais do que 5 ms para realizar a atualização, o que deixa nada menos que 995 ms para a realização de outras tarefas.

Neste exemplo, a técnica de interrupções permite a realização de operações de E/S concomitantemente com o processamento.

MÓDULOS DE MEMÓRIA:

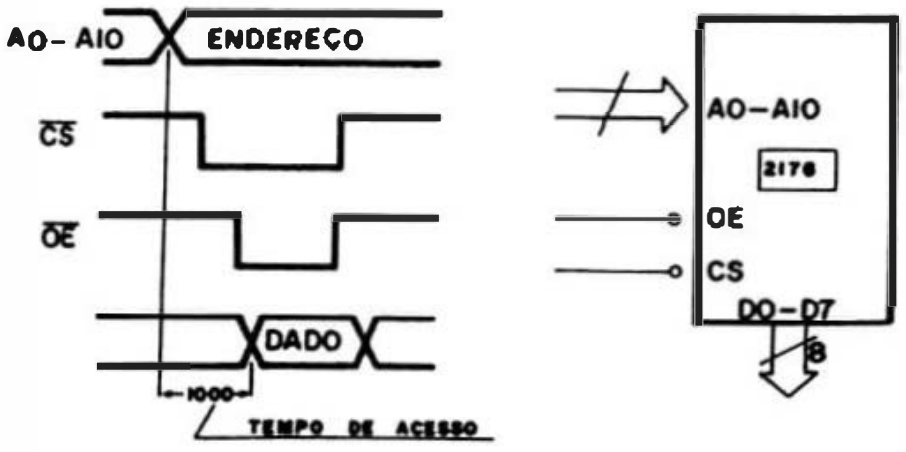
Um outro aspecto do hardware do microprocessador Z-80 que tem que ser considerado é a forma de como ele está conectado às memórias EPROM, RAM estáticas e RAM dinâmicas.

A seguir, apresentaremos alguns projetos de módulos de memórias, todos eles de caráter puramente didático, nos quais, o enfoque maior será com relação à lógica de controle necessária ao bom funcionamento do Z-80. A lógica de controle que faz a ligação entre o Z-80 e os módulos de memória é denominada INTERFACE ou CIRCUITO DE INTERFACE.

MÓDULO DE MEMÓRIA EPROM:

A figura 15 nos mostra a pinagem de uma memória EPROM, C.I. 2716 de 2 Kbytes, bem como o procedimento de leitura da mesma.

FIGURA 15



Por ser uma memória de 2 Kbytes, ela necessita de onze linhas de endereçamento, pois 2^{11}=2048 bytes. A entrada CS (chip select) é a responsável pela ativação do chip, quando este é colocado em um nível lógico "0". A entrada $\overline{OE}$ (output enable) libera os circuitos internos da memória 2716 que são buffers e "three-states" (três-estados).

A figura 16 nos mostra o projeto de um módulo de 6 Kbytes de memória EPROM.

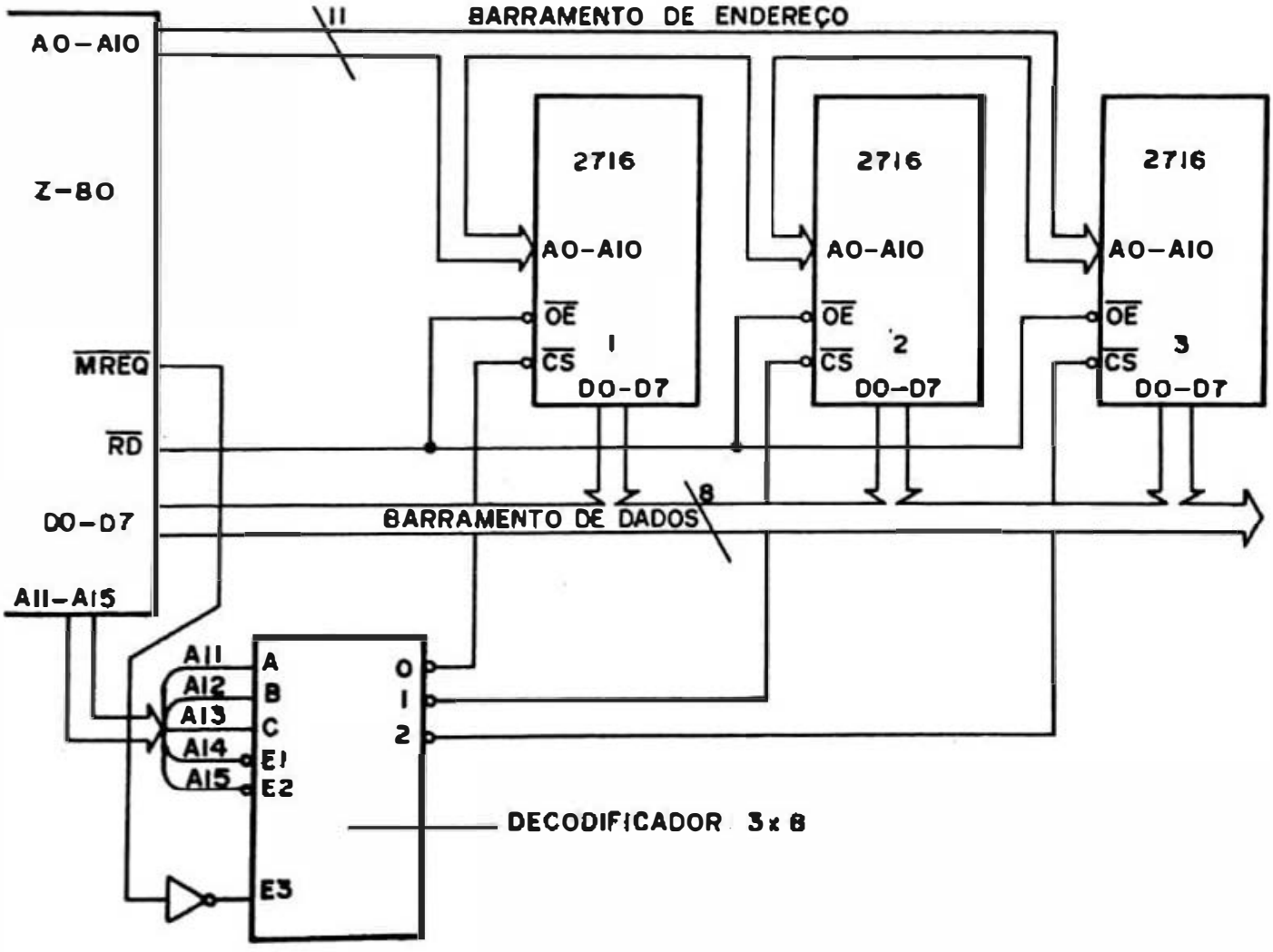


FIGURA 16

Através da figura 16, podemos observar que as onze linhas menos significativas (A_0-A_{10}) do barramento de endereçamentos chegam às três memórias EPROM, ou melhor, aos três chips de memória. Note que a linha $\overline{RD}$ é que controla a saída de dados da EPROM, pois, quando ela está no nível lógico zero, o barramento de dados do Z-80 (D_0-D_7) estará no modo de entrada.

Para selecionar cada chip de memória (2716), utilizam-se as demais linhas do barramento de endereços (A_{11}-A_{15}), devidamente conectadas a um decodificador. Esse decodificador possui três entradas enable (E_1, E_2 e E_3). Dessas entradas, as duas primeiras (E_1 e E_2) são ativadas no nível lógico "0" e nelas estão conectadas as linhas A_{14} e A_{15}. Na entrada E_3, temos uma porta inversora que recebe como entrada a linha $\overline{MREQ}$. O quadro apresentado na figura 17 nos mostra a faixa de endereços das EPROM'S.

FIGURA 17

A15	A14	A13	A12	A11	A1Ø-----AØ	
Ø	Ø				X------X	
		↓	↓	↓		
		Ø	Ø	Ø	- EPROM Nº 1	ØØØØ - Ø7FF
		Ø	Ø	1	- EPROM Nº 2	Ø8ØØ - ØFFF
		Ø	1	Ø	- EPROM Nº 3	1ØØØ - 17FF

ENDEREÇOS DA EPROM's

MÓDULO DE MEMÓRIA RAM (Estática)

A figura 18 nos mostra a pinagem de uma memória RAM, C.I. da 2114, de 1K x 4, bem como a lógica de leitura e escrita mesma.

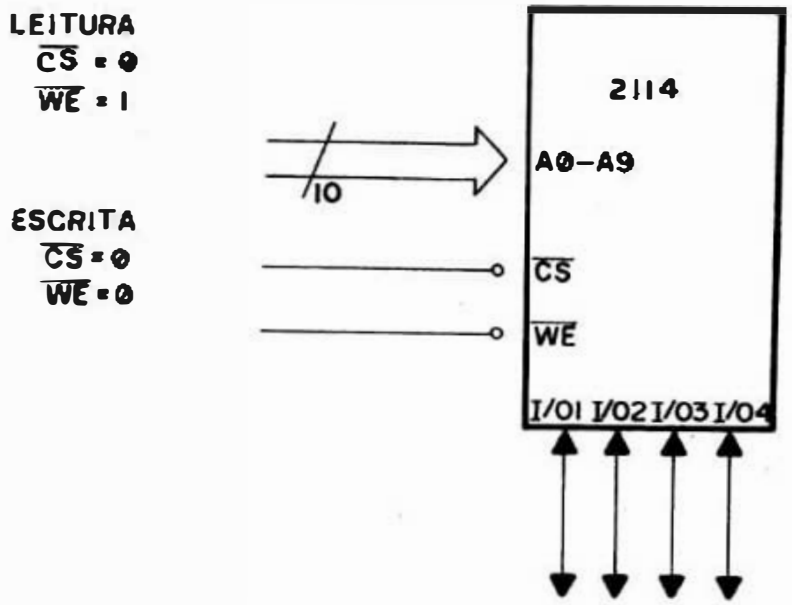


FIGURA 18

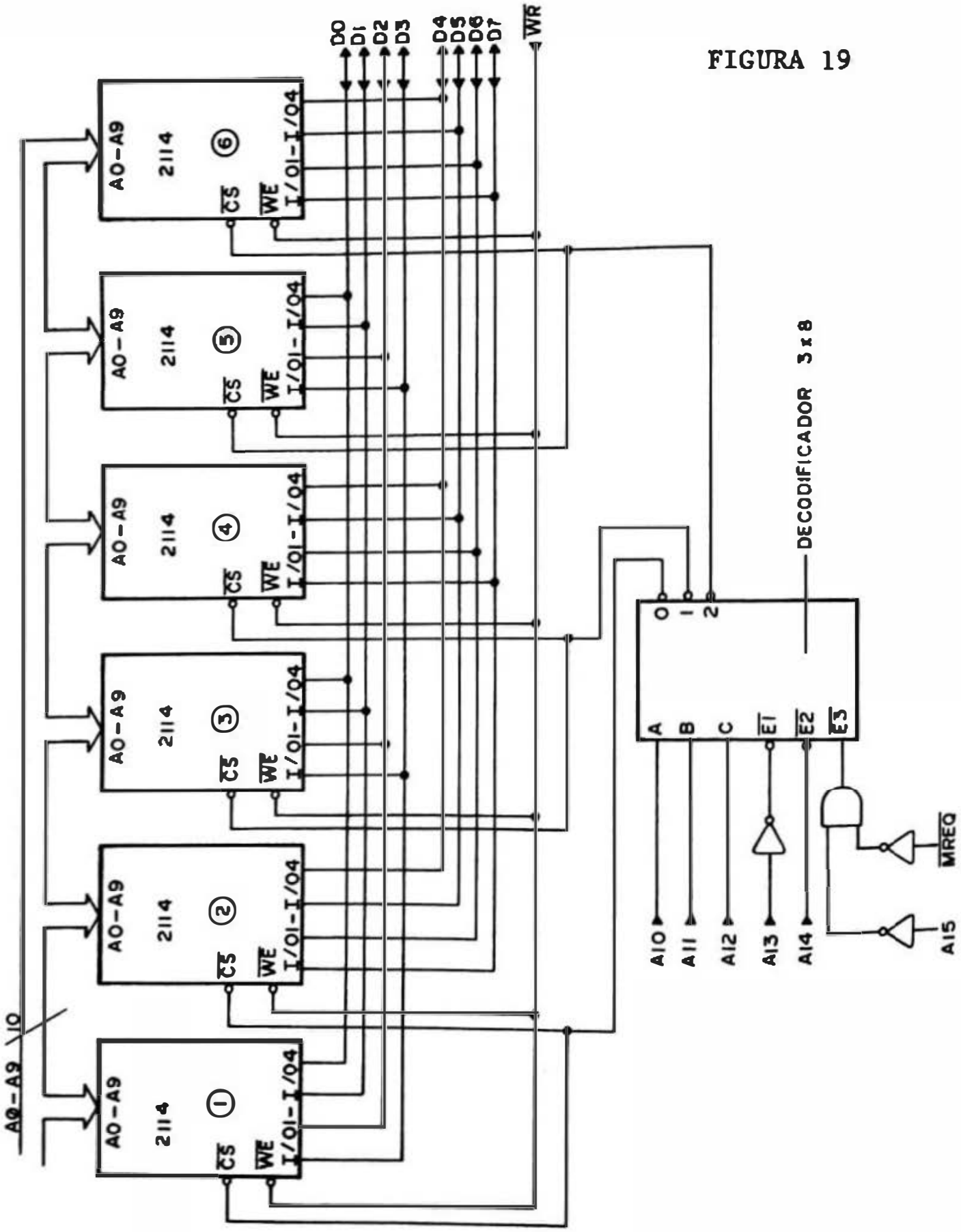
A figura 19 nos mostra o projeto de um módulo de memória RAM de 3 Kbytes. Da mesma forma que no projeto anterior, não entraremos em todos os detalhes do sincronismo (TIMING) da 2114, mas procuraremos dar maior ênfase à lógica necessária para a ligação deste módulo de memória com o microprocessador Z-80.

Observando a figura 19, podemos verificar que a entrada CS (chip select) é a responsável pela seleção da 2114 e seu estado ativo é o nível lógico "0". A comunicação com o barramento de dados é feita por intermédio de quatro (4) linhas bidirecionais, todas elas controladas pela entrada WE (WRITE NABLE). Quando CS e WE forem iguais a zero ("0"), o conteúdo do barramento de dados é escrito no endereço especificado pelos bits A_0-A_9. Porém, quando CS = "0" e WE = "1", teremos uma leitura de dados no endereço especificado.

Cada C.I., ou melhor, cada chip de memória 2114 é organizado em 1024 x 4 bits. Daí decorre o fato de precisarmos de dois chips para termos a configuração 1024 x 8 bits, a qual é compatível com o Z-80, pois este é um microprocessador de 8 bits. Desta forma, temos os chips 1, 3 e 5 ligados às li-

nhas D_0, D_1, D_2 e D_3 do barramento de dados. E os chips 2, 4 e 6 conectados às outras quatro linhas do barramento de dados (D_4, D_5, D_6 e D_7).

FIGURA 19



Observando-se a figura 19, também podemos notar que as dez linhas menos significativas do barramento de endereços (A_0-A_9) estão conectadas aos seis (6) chips 2114 que constituem o módulo de memória em questão.

Uma vez que a memória RAM é de escrita e leitura, teremos a linha de controle WR, do Z-80, conectada à entrada WE (WRITE ENABLE) de todos os chips. Assim sendo, toda a vez que o Z-80 realizar uma operação de escrita ($\overline{WR} = 0$), a entrada $\overline{WE}$ é ativada.

A seleção de cada par de chip (1024 x 8 bits) é feita por um decodificador 3x8, cujas entradas (A, B, C) correspondem às linhas A_{10}, A_{11} e A_{12}. Porém, a seleção é realmente fornecida pelas 3 linhas de bits mais significativos do barramento de endereços, ou seja, A_{13}, A_{14} e A_{15}, e pelo sinal $\overline{MREQ}$ do Z-80. Quando tivermos $A_{13} = 1$, $A_{14} = 0$, $A_{15} = 0$ e $\overline{MREQ} = 0$, o decodificador é selecionado. Note que cada uma das saídas do decodificador, também conhecidas por "linhas de seleção", estão ligadas a dois chips 2114. Para finalizar, a figura 20 nos mostra um quadro contendo a faixa de endereçamento deste módulo de memória RAM.

FIGURA 20

A15	A14	A13	A12	A11	A1Ø	A9------AØ		
Ø	Ø	1				X-------X		
			Ø	Ø	Ø →	1ºK	2ØØØ	23FF
			Ø	Ø	1 →	2ºK	24ØØ	27FF
			Ø	1	Ø →	3ºK	28ØØ	2BFF

ENDEREÇOS DAS RAM's

MÓDULO DE MEMÓRIA RAM (Dinâmica):

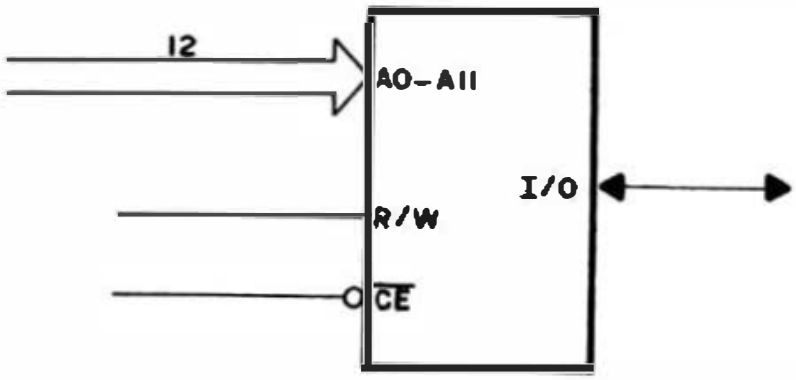
A maneira de se ligar uma memória RAM dinâmica ao microprocessador Z-80, é semelhante à ligação de uma memória RAM estática, pois, em ambos os casos, os objetivos são os mesmos, ou seja, ler e gravar dados na memória. Porém, a principal diferença está no fato de que as memórias dinâmicas necessitam de um "refresh" periódico em cada célula, para poderem reter a informação. Isto significa que devemos realizar um ciclo de

leitura em cada célula, tipicamente, de a cada 2 ms. Para que este ciclo de leitura ou "refresh" seja realizado, é necessário um hardware adicional, o qual permita, também, a consecução de acessos normais de leitura.

O microprocessador Z-80 apresenta um refresh automático, o qual é realizado por intermédio da linha $\overline{RFSH}$ e do registro R. Nos ciclos de busca da instrução $\overline{M1}$, quando o Z-80 estiver realizando operações internas, o conteúdo do registro R é colocado nas linhas A_0-A_6, as quais correspondem aos sete (7) bits menos significativos do barramento de endereços. Este fato é levado ao conhecimento do meio externo, através da ativação dos sinais $\overline{RFSH}$ e $\overline{MREQ}$. O registro R, como sabemos, é incrementado de uma unidade a cada ciclo $\overline{M1}$. Sabe-se, também, que o registro R possui sete (7) bits, o que nos leva a concluir que temos uma repetição do conteúdo do registro R, no máximo a cada 128 ciclos $\overline{M1}$.

Apresentamos, em seguida, como exemplo, um módulo de 4 K bytes x 8 bits de memória RAM dinâmica. Neste projeto iremos utilizar um chip de memória RAM dinâmica de 4 Kbytes x 1 bit. Logo, para obtermos o módulo de 4 Kbytes x 8 bits, necessitamos de 8 destes circuitos integrados (ou chips). Na figura 21 temos a apresentação e a identificação dos pinos do chip de 4 Kbytes x 1 bit.

FIGURA 21



Os 12 bits de endereçamento estão conectados internamente às colunas e às linhas da matraiz da memória. Tanto as linhas como as colunas são endereçadas por 6 bits, o que perfaz um total de 64 linhas e 64 colunas. Para que a informação permaneça na memória, precisamos acessar as 64 linhas a cada 2 ms, o que não pode ser garantido por acessos randômicos realizados durante a execução de um programa. Esta leitura periódica ou este acesso às 64 linhas a cada 2 ms constitui o chamado "refresh". Para consegui-lo há então a necessidade de implementarmos esta leitura. A figura 22 nos mostra um módulo de 4K x 8 bits de memória RAM dinâmica.

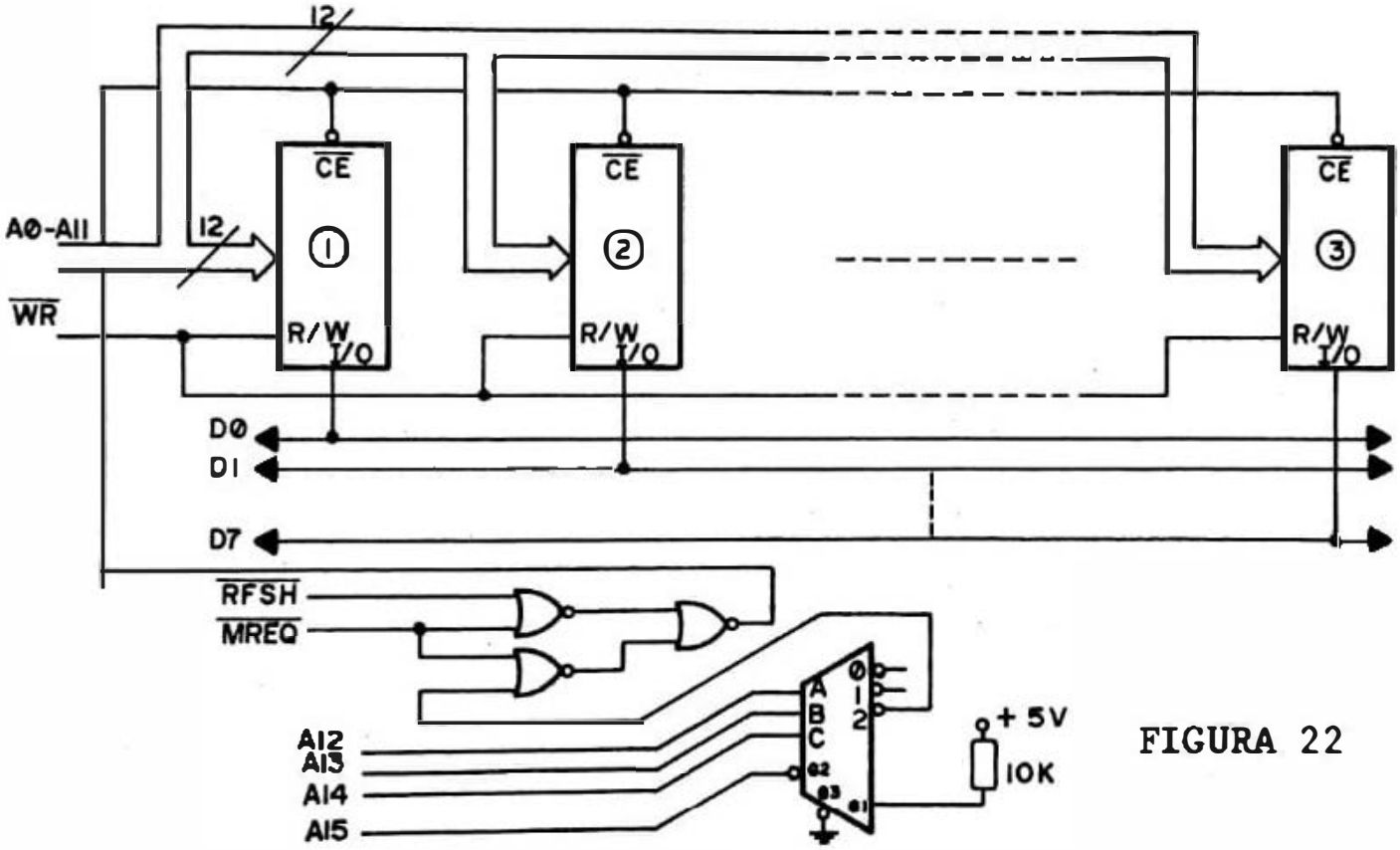


FIGURA 22

Teremos a seleção dos chips para a leitura ($\overline{WR} = 1$) ou escrita ($\overline{WR} = 0$), quando o sinal $\overline{MREQ} = 0$, $A_{15} = A_{14} = A_{12}$ e $A_{13} = 1$.

O refresh é obtido, quando os sinais $\overline{RFSH} = 0$ e $\overline{MREQ} = 0$, pois, quando isto ocorre, o conteúdo do registro R é transferido para o barramento de endereços (A_0-A_6), o qual irá, então, selecionar as linhas da matriz interna do chip (A_0-A_5), ou seja, irá acessar as 64 linhas da memória.

Atualmente, empregam-se muito chips de memórias RAM dinâmicas, as quais já possuem a entrada de endereços multiplexadas no tempo. Esta multiplexação, apesar de complicar o hardware de interface entre o Z-80 e a memória (ou módulo de memória), permite uma certa economia de pinos do chip.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-13

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) O estado T é também conhecido por:

a) ciclo T;

b) ciclo de clock;

c) ciclo de refresh;

d) os itens a e b estão corretos.

2) O ciclo T para o microprocessador Z-80 é de:

a) 400 ns;

b) 250 ns;

c) 500 ns;

d) 125 ns.

3) Toda instrução executada pelo microprocessador Z-80, com exceção das instruções do bloco especial, possuem:

a) de 1 a 3 ciclos de máquina;
b) de 1 a 6 ciclos de máquina;
c) de 3 a 6 ciclos de máquina;
d) apenas 1 ciclo de máquina.

4) Cada ciclo de máquina, ou ciclo M, do Z-80 possui:
a) de 3 a 5 ciclos T;
b) de 1 a 6 ciclos T;
c) de 3 a 6 ciclos T;
d) de 1 a 3 ciclos T.

5) Durante a operação do microprocessador Z-80, quantos ciclos de máquina básicos podem ocorrer?
a) 6
b) 5
c) 3
d) 7

6) O ciclo M_1 permite à UCP:
a) ler o byte do código de operação da memória externa, colocando-o no registro de instrução do Z-80;
b) decodificar o código de operação, identificando a instrução a ser executada;
c) implementar uma parte ou possivelmente toda a operação;
d) Os itens a, b e c estão corretos.

7) Com relação ao ciclo de leitura e escrita em dispositivos de E/S, o endereço do dispositivo de E/S é colocado:
a) nas linhas D_0-D_7 do barramento de dados;
b) nas linhas A_0-A_7 do barramento de endereços;
c) nas linhas A_0-A_5 do barramento de endereços;
d) N.R.A.

8) A principal desvantagem da técnica de varredura diz respeito:

a) ao tempo perdido na observação contínua dos dispositivos E/S;

b) à execução do programa principal;

c) ao tempo ganho na observação contínua dos dispositivos E/S;

d) N.R.A.

9) O C.I. 2716 necessita de:

a) 10 linhas de endereçamento;

b) 5 linhas de endereçamento;

c) 11 linhas de endereçamento;

d) N.R.A.

10) Com relação ao módulo de memória RAM dinâmica, o conteúdo do registro R, se repete no máximo a cada:

a) 228 ciclos $\overline{M_1}$;

b) 128 ciclos $\overline{M_1}$;

c) 628 ciclos $\overline{M_1}$;

d) 314 ciclos $\overline{M_1}$.

CURSOS DE APERFEIÇOAMENTO

CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP86100 - LONDRINA - PARANÁ

LIÇÃO MC-14

MICROPROCESSADOR Z-80 III

INTRODUÇÃO

Até o presente momento, nas lições referentes ao microprocessador Z-80, analisamos apenas os aspectos ligados ao hardware deste microprocessador. Agora, começaremos a estudar o software do Z-80. Na verdade, iremos dar maior ênfase ao estudo dos modos de endereçamento do Z-80, os quais são fundamentais para o perfeito entendimento das instruções deste microprocessador. A parte que se refere à programação em si será vista mais adiante, nas lições de programação e nos manuais de prática.

MODOS DE ENDEREÇAMENTO DO Z-80:

O código de uma instrução de memória do Z-80 possui dois campos: um que designa o código de operação e outro que designa o operando. Este fato se encontra ilustrado na figura 1.

Dentro do conjunto de instruções do microprocessador Z-80, as quais estudaremos logo mais, encontraremos algumas ins

truções que não necessitam do campo do operando, as quais podem ser executadas durante os últimos períodos do ciclo de busca da instrução (ciclo M_1, estudado na lição anterior). Porém, a grande maioria das instruções do Z-80 necessita do campo do operando para que possa executar as suas funções. Esses operandos podem ser, por exemplo, o conteúdo de registros internos da UCP ou de posições de memórias externas ou ainda os dados contidos nos canais de entrada e saída.

FIGURA 1

Podemos então definir os modos de endereçamento do Z-80 como sendo as maneiras pelas quais as suas instruções podem designar os seus operandos. Esses modos de endereçamento são em número de onze (11), a saber:

a) Modo Imediato;

b) Modo Imediato Extendido;

c) Modo Implícito;

d) Modo Página Zero Modificada;

e) Modo Relativo;

f) Modo Extendido;

g) Modo Indexado;

h) Modo por Registro;

i) Modo por Registro Indireto;

j) Modo por Bit.

l) Modo por Ponteiro de Pilha.

Vejamos agora em que consiste cada um desses modos de endereçamento.

a) Endereçamento Imediato:

Nesta modalidade de endereçamento, o byte que segue o código de operação na memória contém o atual operando, ou seja, o segundo ou o terceiro byte do código da instrução é o próprio operando. Esta modalidade de endereçamento é muito útil, quando desejamos operar constantes de 8 bits.

Um exemplo de uma instrução imediata de 8 bits é mostrada na figura 2.

FIGURA 2

ADD A,N (ADD IMMEDIATE)

1	1	0	0	0	1	1	0	Byte 0	(CGH)
			N					Byte 1	(8-bit operando imediato)

código da instrução

ADD A,35

1	1	0	0	0	1	1	0	Byte 0	(CGH)
0	0	1	0	0	0	1	1	Byte 1	(23H = 35)

decimal

hexadecimal

ADD = código de operação
A,N = operando

Note que, neste exemplo, a instrução ADD A,N soma o conteúdo do acumulador ao conteúdo do segundo byte e deposita o resultado no acumulador. O primeiro byte 11000110 (CGH) é fixo, pois este é o "código op" (código de operação) para ADD A,N. O segundo byte "N", por sua vez, permanece constante durante uma determinada operação.

Podemos então concluir que o código desta instrução varia em função da "constante" que o programador escolher para ser adicionada ao acumulador.

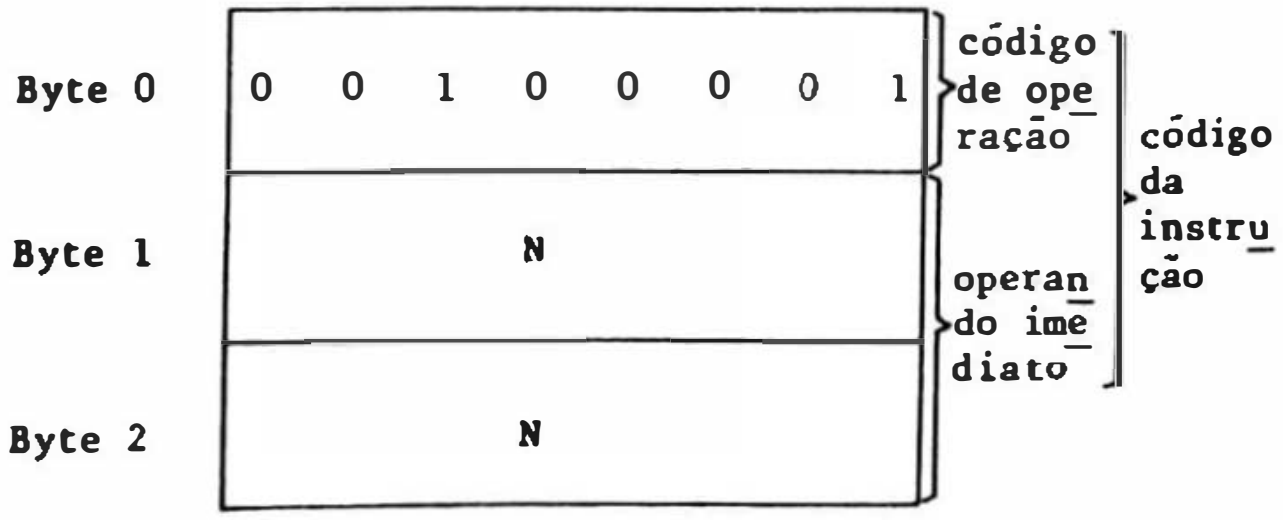
Os oito (8) bits de instrução imediata são facilmente encontrados para instruções LOADs (LD B,38), instruções aritméticas (OR 59) e muitas outras. Cabe dizer ainda que na forma de instrução imediata o dado nunca é colocado entre parênteses.

b) Endereçamento Imediato Extendido:

Normalmente, as instruções com modo de endereçamento imediato extendido possuem operandos de 16 bits que fazem parte do código da instrução. A figura 3 nos mostra um exemplo de instrução imediata de 16 bits. No caso, a instrução apresentada é a LD HL,NN.

FIGURA 3

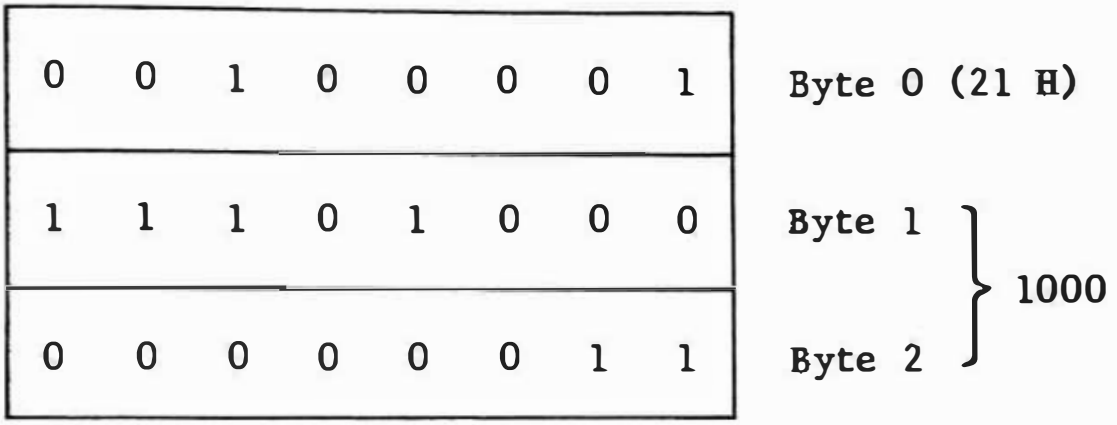
LD HL,NN (LOAD HL IMMEDIATE)



O byte "0" representa o código de operação da instrução (código op) e é fixo. Já os bytes 1 e 2 representam o operando imediato, sendo que o byte 1 é a parte menos significativa e o byte 2, a parte mais significativa. Para exemplificar melhor tais fatos, vejamos a instrução LD HL, 1000, a qual irá carregar o valor decimal 1000 para o par de registro HL. A figura 4 ilustra este exemplo.

FIGURA 4

LD HL,1000



O byte 0 é o código de operação "00100001" em binário ou "21 H" em hexadecimal e está fixo. Os bytes 1 e 2 representam o decimal 1000 que equivale ao hexadecimal "3E8" ou ao binário "0000001111101000", o qual será carregado para o par de registros HL.

Este modo de endereçamento,é muito utilizado para carregar constantes em registros de 16 bits, tais como "IX, IY e SP" e em pares de registros como "BC, HL e DE".

c) Endereçamento Implícito:

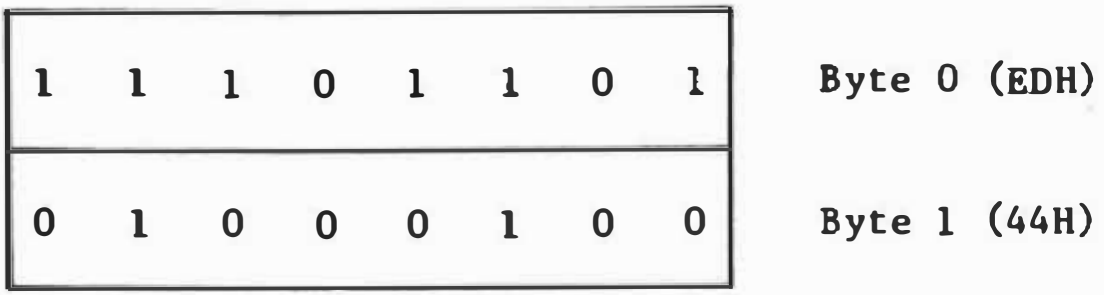
Neste modo de endereçamento as instruções não possuem campo de operando, pois, neste caso, é o próprio código de operação que determina qual registro da UCP irá conter o operando. Devido a este fato, os códigos de operação destas instruções são fixos.

Como exemplo, podemos citar as instruções aritméticas que sempre determinam se o acumulador será o registro do destino do resultado das operações. Outras instruções, tais como NEG (Acumulador negado), SCP (Colocar o Flag de Carry) e NOP

(Nenhuma operação), bem como as que executam operações simples, ou são limitadas para operações no registro A, ou podem ser somadas ao rol de exemplos deste modo de endereçamento. A figura 5 ilustra o formato e a operação das instruções NEG, SCF e NOP.

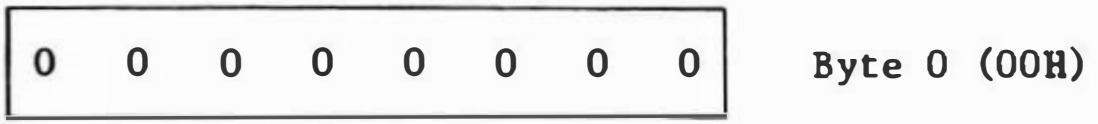
FIGURA 5

NEG (NEGATIVE)



SCF (SET CARRY FLAG)

NOP (NO OPERATION)



Obs.: Os códigos indicados entre parênteses representam o respectivo byte em hexadecimal.

d) Endereçamento Página Zero Modificada (Pag. 0):

Este modo de endereçamento é utilizado somente por oito (8) instruções que são as instruções RESTART (RST T). Estas instruções (RST T) são instruções de 1 byte que equivalem ao CALL.

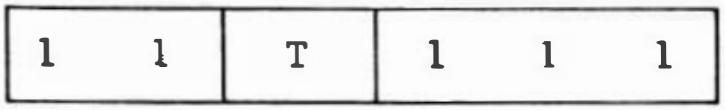
A ação desta instrução (RST T) é armazenar o conteúdo do PC (contador de programa) no alto (topo) da pilha e, em segui

da, desviar o programa para uma das oito posições fixas de memória na "página zero".

Por outro lado, a "página zero" representa uma certa região da memória que pode ser endereçada com os oito (8) bits menos significativos do barramento de endereços, mantendo os seus oito (8) bits mais significativos zerados.

A figura 6 nos mostra o formato da instrução RST T.

RST (RESTART) FIGURA 6



Byte 0

ENDEREÇO DE DESVIO NA PÁGINA ZERO	VALOR DE T	ASSEMBLY
0 0 0 0 H	0 0 0 B	RST 0 0 H
0 0 0 8 H	0 0 1 B	RST 0 8 H
0 0 1 0 H	0 1 0 B	RST 1 0 H
0 0 1 8 H	0 1 1 B	RST 1 8 H
0 0 2 0 H	1 0 0 B	RST 2 0 H
0 0 2 8 H	1 0 1 B	RST 2 8 H
0 0 3 0 H	1 1 0 B	RST 3 0 H
0 0 3 8 H	1 1 1 B	RST 3 8 H

Note que o campo "T" ocupa três (3) bits e, em função destes, o desvio ocorre para uma das oito posições fixas de memória (0H, 8H, 10H, 18H, 20H, 28H, 30H ou 38H), codificadas na figura 6 em hexadecimal. A figura 7 nos mostra o formato da instrução RST 38H.

RST 38H FIGURA 7

Byte 0

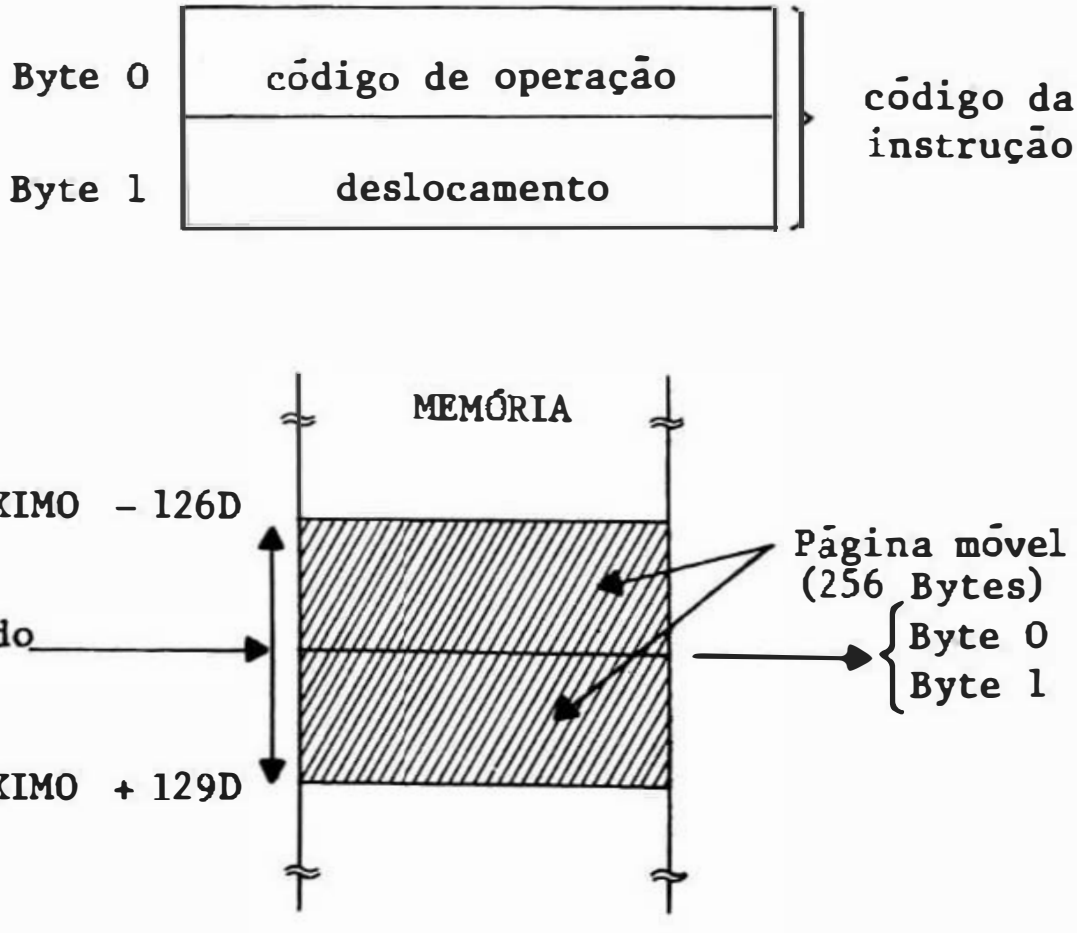
As instruções RESTART são comumente utilizadas em programas como "sub-rotinas".

e) Endereçamento Relativo:

Este modo de endereçamento utiliza um byte de dado seguido do código de operação, a fim de especificar o deslocamento referente ao programa, ou seja, especificar a mudança de curso do programa existente o que permite a execução de JUMP's condicionais e incondicionais.

Neste tipo de instrução, o segundo byte é chamado de deslocamento (DISPLACEMENT) e representa um número algébrico em notação de complemento de 2 na faixa de -128D a +127D (1000000B a 01111111B). Este valor, somado ao conteúdo do contador de programa, nos dá um certo resultado o qual representa o endereço efetivo da instrução que deve dar continuidade ao programa. A figura 8 ilustra o formato e a operação desta instrução.

FIGURA 08

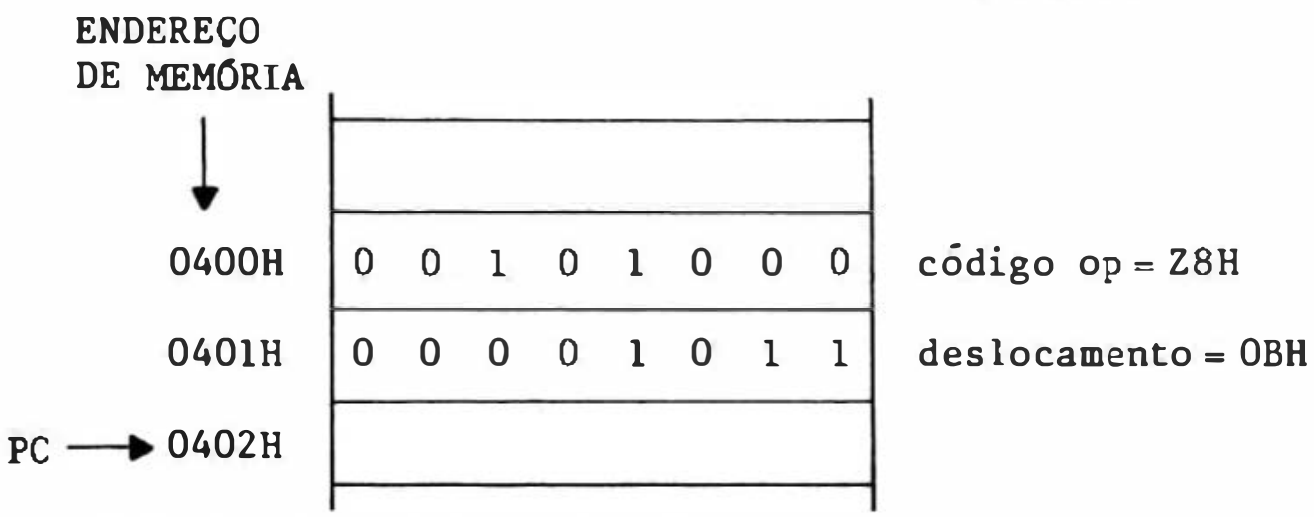


Devido ao fato das instruções com endereçamento relativo ocuparem dois (2) bytes (byte 0 e 1), o endereço efetivo fica

na faixa de -126D a +129D, isto, em relação ao primeiro byte (byte 0) da instrução, permitindo, assim, o acesso a 256 bytes, ou posições de memória localizadas em torno do endereço da instrução corrente. Este conjunto de 256 bytes, que pode ser acessado pelo endereçamento relativo, é designado: "página móvel".

A título de exemplo, citamos a instrução JR Z,E, a qual provoca um desvio no curso normal do programa, se o bit de condição Z for igual a "1". A figura 9 nos mostra esta instrução com deslocamento igual a 11D.

FIGURA 9



Cálculo do endereço efetivo:

0402 H + 0BH = 040DH

Para o exemplo ilustrado na figura 9, se o bit Z for igual a "1", o programa sofre um desvio e prossegue na posição de memória 040DH. Caso contrário, isto é, se o bit Z for igual a "0" (zero), o programa prossegue de acordo com sua orientação natural, de acordo com o contador de programa (PC), ou seja, a partir da posição de memória 0402H.

f) Endereçamento Extendido ou Direto:

Neste modo de endereçamento, a instrução contém no seu próprio código o endereço final do operando com dois (2) bytes que seguem a partir do código de operação (op code). Esta técnica permite o acesso à posição aleatória de memória.

Como exemplo de uma instrução com modo de endereçamento extendido, citamos a LD A,(NN), cujo formato e operação se encontram ilustrados na figura 10.

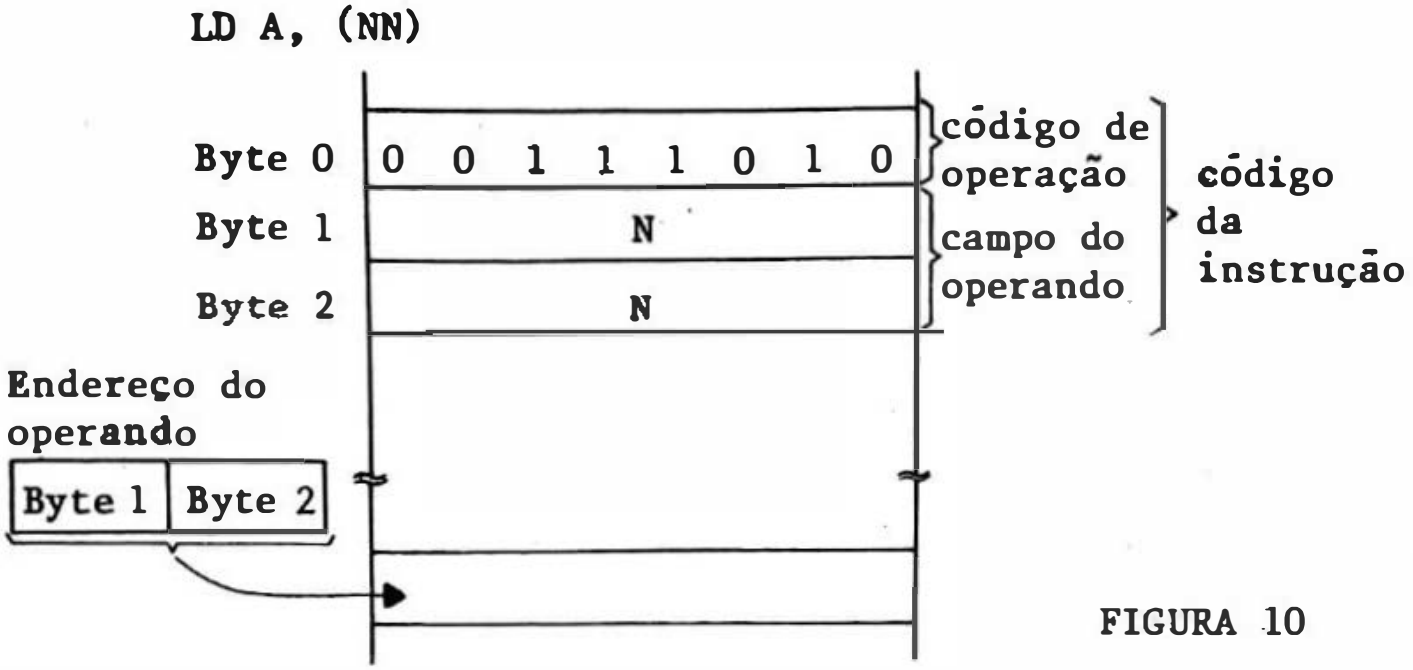


FIGURA 10

O byte "0" (zero) representa o código de operação da instrução, enquanto que os bytes 1 e 2 representam o campo do operando, sendo que o byte 1 corresponde à parte menos significativa e o byte 2, à parte mais significativa. O campo do operando ou mais precisamente os bytes 1 e 2 da instrução LD A,(NN) especificam uma posição de memória, cujo conteúdo é transferido para o acumulador. Um exemplo mais palpável é mostrado na figura 11, onde a operação da memória é diretamente referenciada como LD A,(803H).

LD A,(803 H)

FIGURA 11

0	0	1	1	1	0	1	0	Byte 0
0	0	0	0	0	0	1	1	Byte 1
0	0	0	0	1	0	0	0	Byte 2

A instrução LD A,(803 H) irá carregar o registro A com os conteúdos do endereço de memória 803 H.

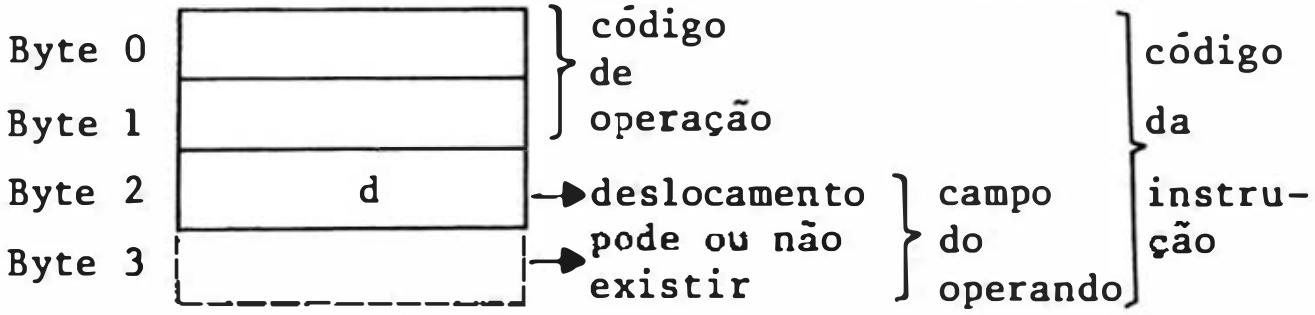
Este modo de endereçamento também é utilizado para o caso de programas que requerem um determinado salto para uma posição de memória ou ainda para armazenar e ler dados em qualquer lugar da memória.

g) Endereçamento Indexado:

Trata-se do modo de endereçamento mais completo do micro processador Z-80. Neste modo de endereçamento, o endereço efetivo do operando na memória é obtido somando-se o conteúdo de um dos dois registros índices IX ou IY ao byte de deslocamento, no campo do operando da instrução. O resultado desta soma é um endereço que deve estar na faixa de -128D a +127D em relação ao conteúdo ou valor armazenado no registro índice.

No modo de endereçamento indexado, o código de operação é composto de dois bytes (byte 0 e byte 1), formando a execução mais longa e gerando dois ciclos M_1. O byte 2 (3º byte) contém o valor do deslocamento "d"; já, o byte 3 (4º byte) pode ou não existir, dependendo da natureza da instrução. A figura 12 nos mostra o formato das instruções com endereçamento indexado.

FIGURA 12



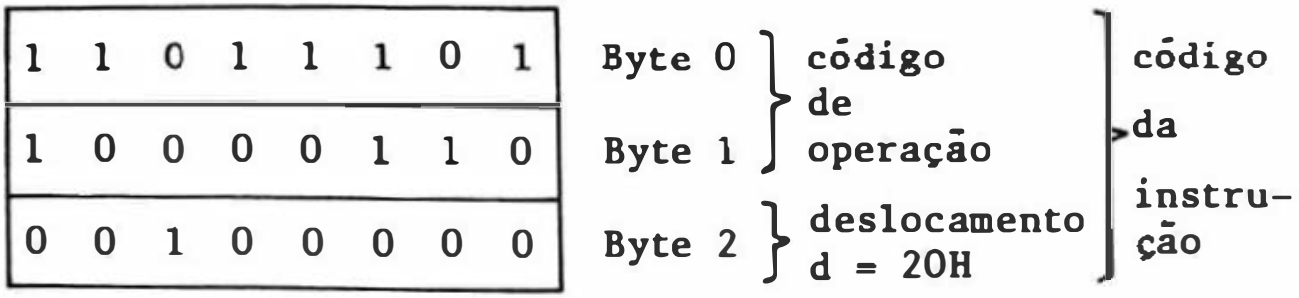
Cálculo do endereço efetivo:

(IX) + d ou (IY) + d

Para maior esclarecimento, vejamos um outro exemplo. Consideremos a instrução ADD A, (Ix + 20H). Suponhamos agora que o registro IX contém 840H. Isto nos leva a um endereço efetivo igual a 860H (840H + 20H = 860H). Esta instrução irá somar o conteúdo do endereço da memória, 860H, ao conteúdo do registro A e colocará o resultado final no registro A. Este exemplo se encontra ilustrado na figura 13.

ADD A,(IX + 20H)

FIGURA 13



Cálculo do endereço efetivo:

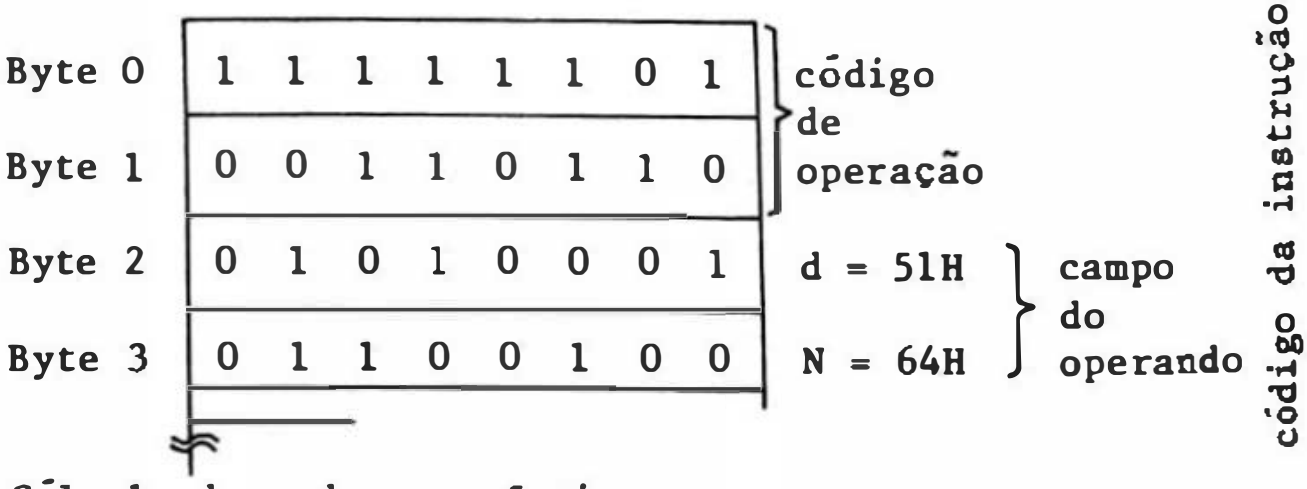
(IX) = 840H
d = 20H

860H (endereço efetivo)

Vejamos agora um outro exemplo: Consideremos a instrução LD (IY + d),N. Esta instrução transfere o operando imediato "N", que corresponde ao byte 3 (ou 4º byte), para a posição de memória determinada pela soma do conteúdo do registro IY com o byte de deslocamento "d". A figura 14 nos mostra o formato e a operação desta instrução, atribuindo-lhe os seguintes valores numéricos: (IY) = 1001H; d = 51H e N = 64H. Executando-se esta instrução com estes valores, teremos o valor 64H armazenado na posição de memória ou endereço 1052H.

LD (IY + 51H), 64H

FIGURA 14



Cálculo do endereço efetivo:

(IY) = 1001H
d = 51H
1052H

O endereço indexado pode ser utilizado com grande vantagens no caso de acesso à tabela de dados. O registro IX ou IY é colocado no início da tabela e o dado pode ser lido ao se mudar o valor do deslocamento, na instrução.

h) Endereçamento por Registro:

Neste modo de endereçamento, um ou mais registros do microprocessador Z-80 são utilizados como operandos da instrução. Contêm campos que especificam os registros do Z-80, os quais serão utilizados na execução das referidas instruções.

Por exemplo, a instrução ADD A,R faz a vez da "porta lógica OU" do acumulador com um registro R de uso geral, depositando o resultado no próprio acumulador. O formato e a operação desta instrução se encontram ilustrados na figura 15.

ADD A,R (ADD REGISTER R TO A)

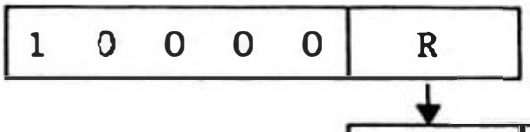


FIGURA 15

R	REGISTRO
0 0 0	B
0 0 1	C
0 1 0	D
0 1 1	E
1 0 0	H
1 0 1	L
1 1 1	A

Para executarmos a instrução ADD A,H (ADD H TO A), basta acrescentarmos no lugar de R o "valor" de H, conforme nos mostra a figura 16.

Byte 0 1 0 0 0 0 1 0 0 H = 100 FIGURA 16

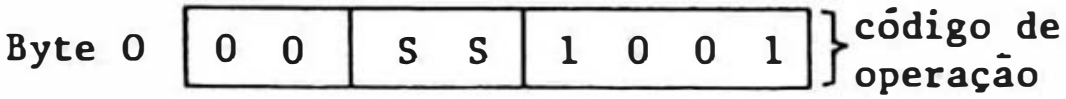
ADD A,100 R

Observe que, neste caso, os cinco (5) bits mais significativos do código de operação são fixos e os três bits menos significativos selecionam o registro de uso geral (2º operando), conforme a tabela mostrada na figura 15.

Cabe ressaltar que um registro de uso geral da UCP é sempre especificado para um campo de três bits.

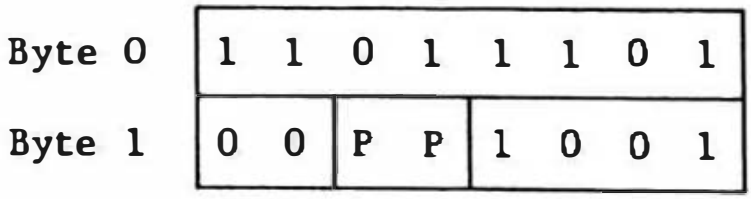
Este mesmo tipo de endereçamento geral pode ser aplicado em outras instruções. Por exemplo, a instrução ADD HL,SS soma um registro específico ao par HL. Neste caso, o "SS" representa dois campos de bit, conforme nos mostra a figura 17.

ADD HL,SS (ADD REGISTER PAIR SS TO HL) FIGURA 17



0 0 = BC
0 1 = DE
1 0 = HL
1 1 = SP

ADD IX, PP (ADD REGISTER PAIR PP TO IX)



0 0 = BC
0 1 = DE
1 0 = IX
1 1 = SP

Nos dois tipos de instruções apresentadas na figura 17, o par de registros varia em função do tipo da instrução.

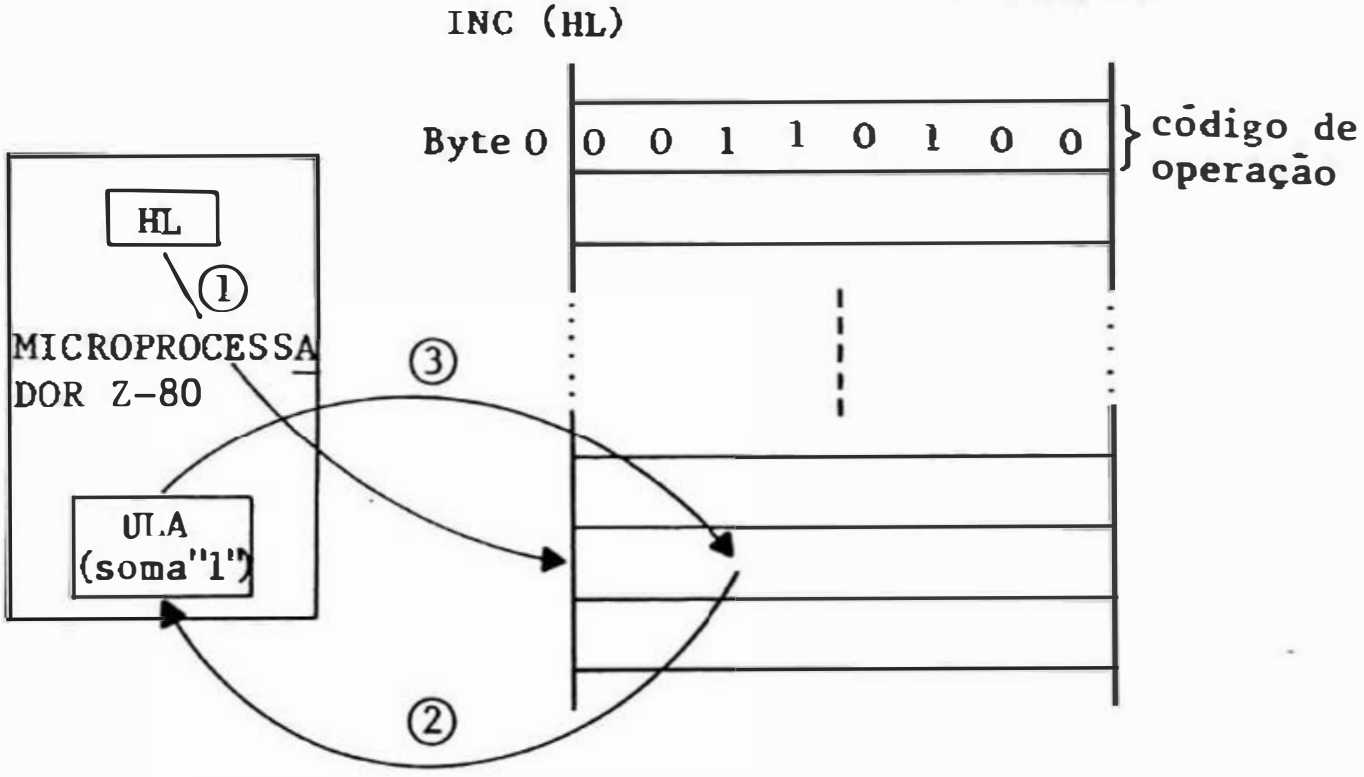
i) Endereçamento por Registro Indireto:

Neste modo de endereçamento, o operando fica em uma posição de memória cujo endereço está contido em um par de registros (BC), (DE) ou (HL).

Esta técnica mostra-se muito poderosa por permitir o acesso de posições de memória de endereço aleatório.

Como exemplo, podemos citar a instrução INC (HL), a qual incrementa, de uma unidade, o conteúdo da posição de memória que foi endereçada pelo par HL. Assim, ao conteúdo da posição de memória é lida e somada uma unidade. Em seguida, o resultado é depositado na própria origem. A figura 18 nos mostra o formato e a operação desta instrução.

FIGURA 18



Em Assembly, o conteúdo de uma posição de memória é sempre representado entre parênteses.

Geralmente este tipo de endereçamento é utilizado pelas instruções: aritméticas, lógicas, incrementais, decrementais, etc. Todas as instruções deste tipo são instruções de um byte.

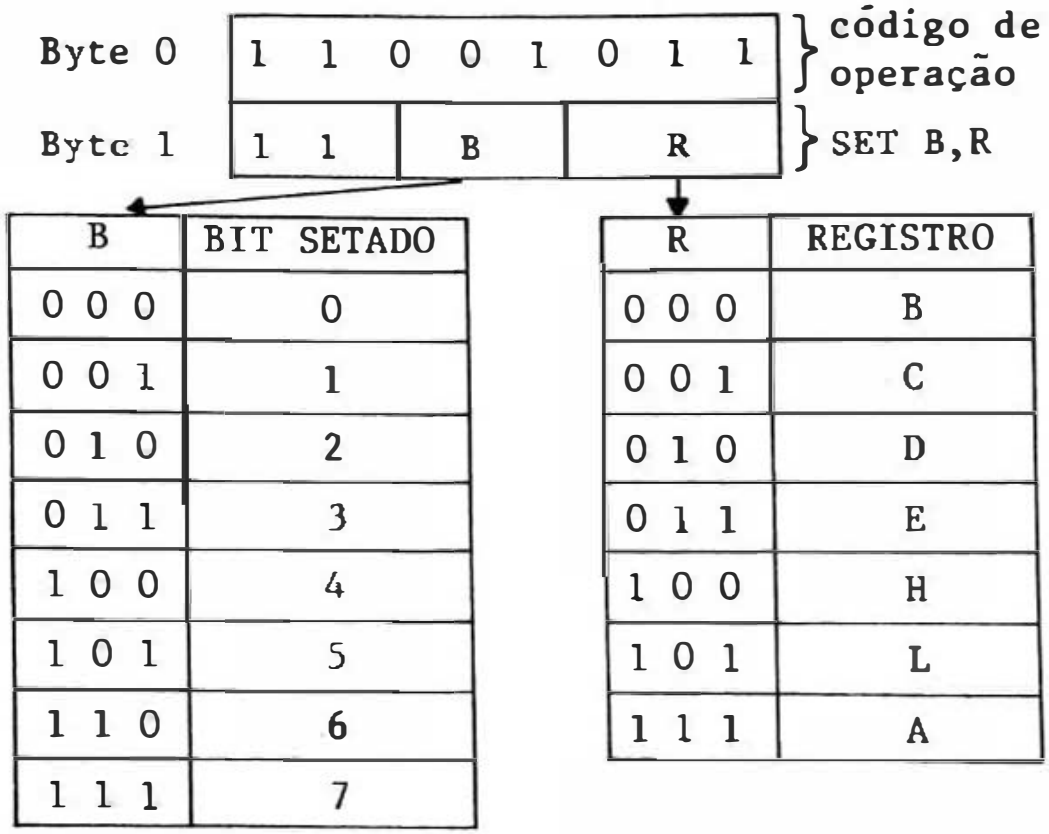
j) Endereçamento por Bit:

Este modo de endereçamento permite acessar individualmente qualquer um dos oito bits que compõem um byte. Por isso, este modo de endereçamento é aplicado para testar, resetar e colocar o bit.

Esta técnica de endereçamento pode ser utilizada em con-

junto com as demais técnicas apresentadas. Por exemplo, esta instrução pode utilizar "registro indireto", "registro", "endereçamento por bit" que refere o campo em que todos estes tipos de instruções especificam o "Bit" a ser usado na instrução. O formato geral deste tipo de instrução é mostrado na figura 19.

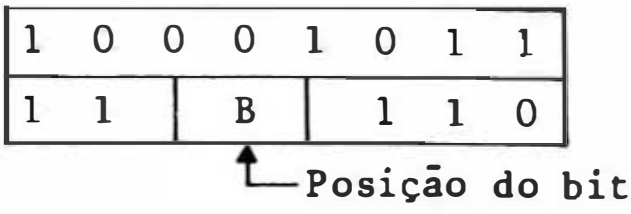
SET B,R (SET BIT, ENDEREÇAMENTO POR REGISTRO) FIGURA 19



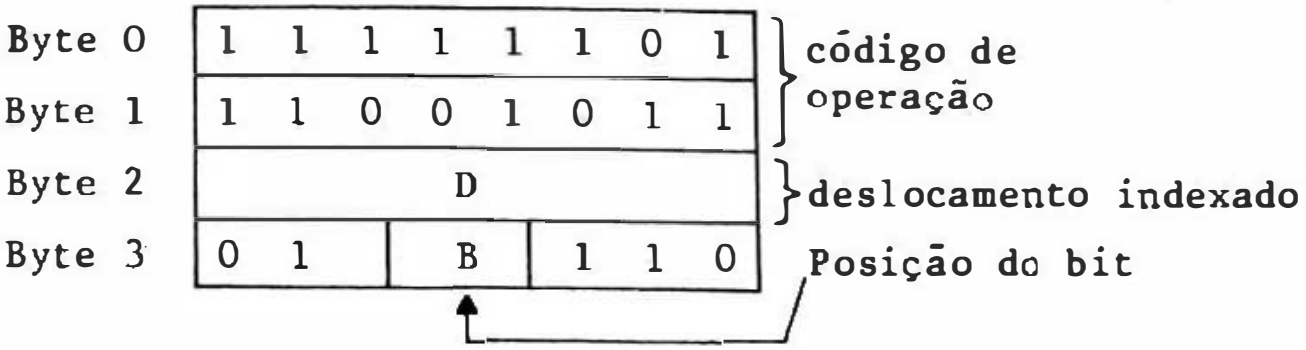
B	BIT SETADO
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

R	REGISTRO
0 0 0	B
0 0 1	C
0 1 0	D
0 1 1	E
1 0 0	H
1 0 1	L
1 1 1	A

RES B, (HL) (RESET BIT, ENDEREÇAMENTO POR REGISTRO INDIRETO)



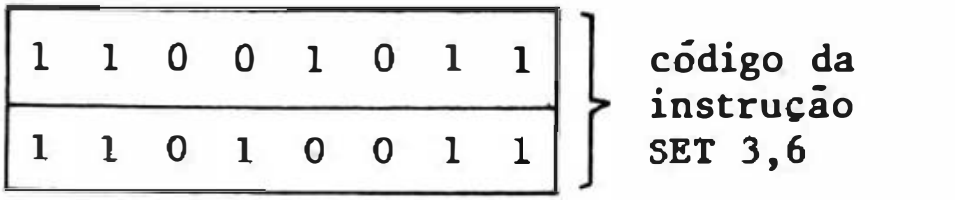
BIT B,(IY + d) (TEST BIT, ENDEREÇAMENTO INDEXADO)



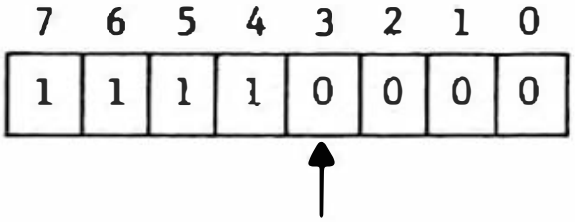
Tomemos agora como exemplo a instrução SET 3,E, cujo formato e operação se encontram ilustrados na figura 20. Note que esta instrução "seta" o bit 3 do registro E. Para facilitar o nosso estudo, vamos supor que o conteúdo inicial do registro E seja F1H. Então vejamos:

SET 3,E

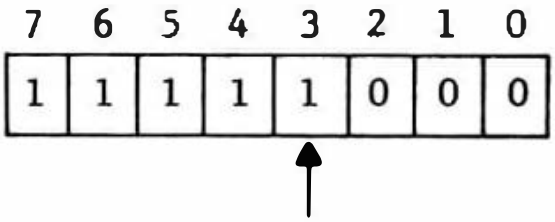
FIGURA 20



Registro E, antes da execução da instrução SET 3,E



Registro E, após a execução da instrução SET 3,E



1) Endereçamento por Ponteiro de Pilha:

Neste modo de endereçamento, o operando fica na posição de memória, endereçada pelo ponteiro de pilha "stack pointer" (SP). Poderíamos ter englobado este modo de endereçamento quando do estudo do endereçamento por registro indireto, porém, para facilitar o nosso estudo, classificamos à parte as instruções que utilizam o ponteiro de pilha (SP). Veremos então as instruções PUSH e POP, as quais são exemplos típicos de endereçamento por ponteiro de pilha.

Instrução PUSH: Neste tipo de instrução, os operandos de 16 bits (os pares AF, BC, DE ou HL e os registros IX e IY) são transferidos para a pilha. Os endereços acessados durante uma operação de PUSH podem ser determinados da seguinte forma:

a) Os 8 bits mais significativos do operando de 16 bits (A,B,D,H,IX_H e IY_H) são armazenados no endereço de memória, fornecido pelo próprio conteúdo do ponteiro de pilha, decrementado de uma unidade.

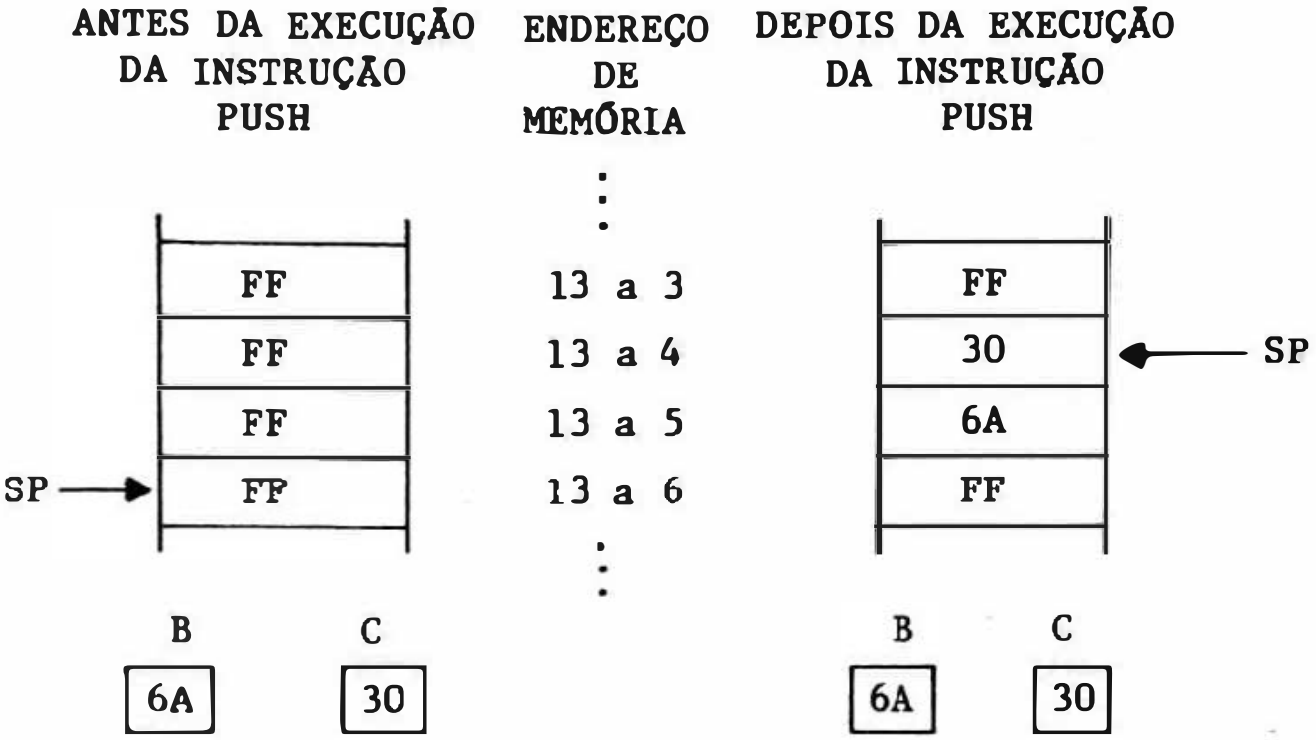
b) Os 8 bits menos significativos do operando de 16 bits (F,C,E,L,IX_L e IY_L) também são armazenados no endereço de me-

mória fornecido pelo próprio conteúdo do ponteiro de pilha, porém decrementado de duas unidades.

c) O ponteiro de pilha é automaticamente decrementado de duas unidades (SP ← SP-2).

Para ilustrar como "opera" a instrução PUSH, apresentamos a figura 21, na qual a instrução PUSH BC transfere o par BC para a pilha.

FIGURA 21



Instrução POP: Neste modo de endereçamento, os operandos de 16 bits são transferidos da pilha para alguns dos seguintes registros do Z-80: AF, BC, DE, HL, IX ou IY. Para determinarmos os endereços de memória acessados durante uma operação de POP, devemos proceder da seguinte forma:

a) O conteúdo da posição de memória endereçada pelo SP é transferido para a parte menos significativa do registro do Z-80 (F,C,E,L,IX_L,IY_L).

b) O conteúdo da posição de memória endereçada pelo SP, quando incrementado de uma unidade, é transferido para a parte mais significativa do registro do Z-80 (A,B,D,H,IX_H e IY_H).

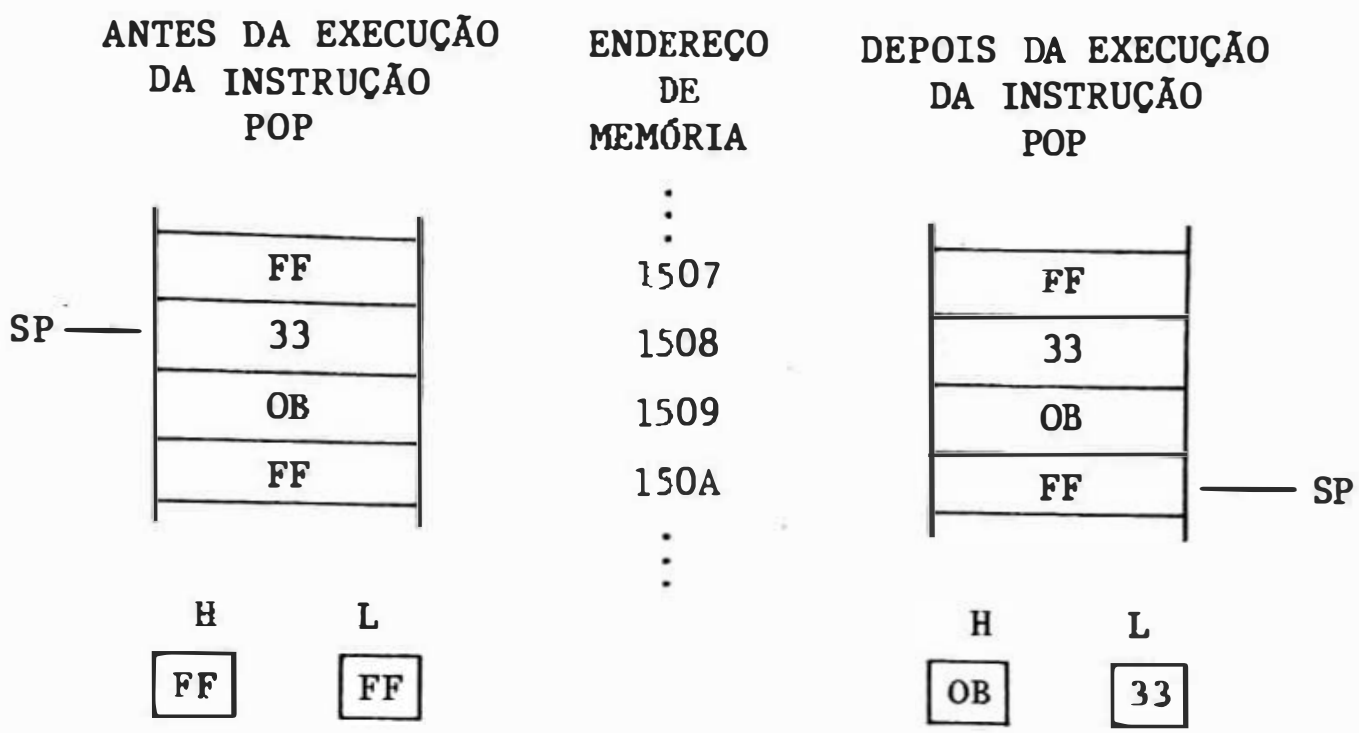
c) O ponteiro de pilha (SP) é automaticamente incrementado de duas unidades (SP ← SP+2).

Apenas a título de ilustração deste modo de endereçamen-

to, apresentamos a figura 22. Neste exemplo, o par HL é carregado com o conteúdo das posições localizadas no topo da pilha.

INSTRUÇÃO: POP HL

FIGURA 22



O microprocessador Z-80 possui, na verdade, dez (10) modos distintos de endereçamento e não 11, como apresentamos. Apenas, por uma questão didática, apresentamos este modo de endereçamento a mais (veja item 1), pois o modo de endereçamento por "ponteiro de pilha" faz parte do modo de endereçamento por registro indireto.

Cabe ressaltar que o termo "modo de endereçamento" se refere ao método pelo qual a instrução tem acesso ao dado (ou à informação). Vejamos agora o conjunto de instruções do microprocessador Z-80.

CONJUNTO DE INSTRUÇÕES DO MICROPROCESSADOR Z-80

Como sabemos, o microprocessador Z-80 é uma versão avançada do microprocessador 8080. Este opera com uma velocidade muito maior, possui o dobro de registros de função geral e possui, ainda, uma colocação de instrução muito maior. As instruções do Z-80 são compatíveis com as instruções do 8080 a nível de linguagem Assembly. Mnemônicos padronizados e formatos de linguagem de máquina, para os dois microprocessadores,

diferem-se consideravelmente. Porém, na forma hexadecimal, quase todo programa escrito para o 8080 produzirá resultados idênticos quando executados pelo Z-80. Os LOOPs de sincronismo do software, devido à velocidade mais alta do Z-80, são exceções a esta característica de compatibilidade. Existe também uma pequena diferença quanto à operação do flag de PARIDADE.

O Z-80 é um microprocessador de 8 bits e, com esses oito (8) bits de informação, é possível obtermos 256 ($2^8 = 256$) códigos de operação distintos, apenas doze (12) a mais que no 8080.

Quanto a sua funcionalidade, podemos classificar o conjunto de instruções do microprocessador Z-80 em:

1 - Instruções do grupo - carga de 8 bits (veja figura 23);
2 - Instruções do grupo - carga de 16 bits (veja figura 24);
3 - Instruções Aritméticas e Lógicas de 8 bits (veja figura 25);
4 - Instruções de Rotação e Deslocamento (veja figura 26);
5 - Instruções de entrada e saída (veja figura 27);
6 - Instruções do grupo de Troca, Pesquisa e Transferência de Bloco (veja figura 28);
7 - Instruções Aritméticas de 16 bits (veja figura 29);
8 - Instruções de Salto (veja figura 30);
9 - Instruções de chamada e retorno (veja figura 31);
10 - Instruções Aritméticas e de controle de uso geral (veja figura 32);
11 - Instruções de Teste, Set e Reset de bits (veja figura 33).

Nas tabelas que se seguem, encontramos o termo "Mnemônico", o qual descreve algo que se utiliza para ajudar a memória humana, ou seja, trata-se de uma abreviação. As instruções escritas sob a forma de "Mnemônico" facilitam em muito o trabalho do programador em recordar a instrução e convertê-la em linguagem de máquina.

Vejamos então as tabelas, contendo o conjunto de instruções do Z-80:

1) INSTRUÇÕES DO GRUPO - CARGA DE 8 BITS:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
LDr,r'	01 rrr r'r'r'	1	4	Carrega o registrador r com o conteúdo de r' r e r' = A,B,C,D,E,H,L
LDr,n	00 rrr 110 n	2	7	Carrega o registro r com a constante n r = A,B,C,D,E,H,L
LDR, (HL)	01 rrr 110	1	7	Carrega o registro r com o conteúdo da posição de memória dada por HL r = A,B,C,D,E,H,L
LDr, (IX+d)	11 011 101 01 rrr 110 d	3	19	Carrega o registro r com o conteúdo da posição de memória dada por (IX+d) r = A,B,C,D,E,H,L
LDr, (IY+d)	11 111 101 01 rrr 110 d	3	19	Carrega o registro r com o conteúdo da posição de memória dada por (IY+d) r = A,B,C,D,E,H,L
LD (HL),r	01 110 rrr	1	7	Carrega a posição de memória dada por HL com o conteúdo do registro r r = A,B,C,D,E,H,L
LD(IX+d),r	11 011 101 01 110 rrr d	3	19	Carrega a posição de memória dada por (IX+d) com o conteúdo do registro r r = A,B,C,D,E,H,L
L (IY+d),r	11 111 101 01 110 rrr d	3	19	Carrega a posição de memória dada por (IY+d) com o conteúdo do registro r r = A,B,C,D,E,H,L
LD(HL),n	00 110 110 n	2	10	Carrega a posição de memória dada por HL com a constante n
D(IX+d),n	11 011 101 00 110 110 d n	4	19	Carrega a posição de memória dada por (IX+d) com a constante n
LD(IY+d),n	11111101 00110110 d n	4	19	arrega a posição de memória dada por (IY+d) com a constante n
L D A, (BC)	00001010	1	7	Carrega o acumulador com o conteúdo da memória dada por BC
L D A, (DE)	00011010	1	7	Carrega o acumulador com o conteúdo da memória dada por DE
L D A, (nn)	00111010 n n	3	13	Carrega o acumulador com o conteúdo da memória dada por nn
L D (BC), A	00000010	1	7	Carrega a posição de memória dada por BC com o conteúdo do acumulador
L D (DE), A	00010010	1	7	Carrega a posição de memória dada por DE com o conteúdo do acumulador
L D (nn), A	00110010 n n	3	13	Carrega a posição de memória dada por nn com o conteúdo do acumulador
L D A, I	11101101 01010111	2	9	Carrega o acumulador com o conteúdo do registro I I - Interrupt register
L D A, R	11101101 01011111	2	9	Carrega o acumulador com o conteúdo do registro R R - Refresh register
L D I, A	11101101 01000111	2	9	Carrega o registro I com o conteúdo do acumulador I - Interrupt register
L D R, A	11101101 01001111	2	9	Carrega o registro R com o conteúdo do acumulador R - Refresh register

FIGURA 23

2) INSTRUÇÕES DO GRUPO - CARGA DE 16 BITS:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
L D dd, nn	00dd0001 n n	3	10	Carrega o par de regiatradorea dd com o valor nn dd = BC, DE, HL, SP
L D IX, nn	11011101 00100001 n n	4	14	Carrega o registro IX com o valor nn
L D IY, nn	11111101 00100001 n n	4	14	Carrega o registro IY com o valor nn
L D H L, (nn)	00101010 n n	3	16	Carrega o par de regiatradores HL com o conteúdo da memória dada por nn e nn + 1
L D dd, (nn)	11101101 01dd1011 n n	4	20	Carrega o par de regiatradores dd com o conteúdo da memória dada por nn a n + 1 dd = BC, DE, HL, SP
L D IX, (nn)	11011101 00101010 n n	4	20	Carrega o regiatro IX com o conteúdo da memória dada por nn e nn + 1
L D IY, (nn)	11111101 00101010 n n	4	20	Carrega o registro IY com o conteúdo da memória dada por n a nn + 1
L D (nn), HL	00100010 n n	3	16	Carrega a memória dada por nn e nn + 1 com o conteúdo do par de registradores HL
L D (nn), dd	11101101 01dd0011 n n	4	20	Carrega a memória dada por n e n + 1 com o conteúdo do par de registradores dd
L D (nn), IX	11011101 00100010 n n	4	20	Carrega a memória dada por n e nn + 1 com o conteúdo do registro IX
L D (nn), IY	11111101 00100010 n n	4	20	Carrega a memória dada por nn e n + 1 com o conteúdo do registro IY

FIGURA 24

Continuação da tabela apresentada na FIGURA 24.

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
L D S P; HL	11111001	1	6	Carrega o SP com o conteúdo de HL
L D S P, IX	11011101 11111001	2	10	Carrega o SP com o conteúdo do registro IX
L D S P, IY	11111101 11111001	2	10	Carrega o SP com o conteúdo do registro IY
PUSH qq	11qq0101	1	11	Carrega a memória endereçada pelo SP com o conteúdo par de registradores qq qq = BC, DE, HL, AF
PUSH IX	11011101 11100101	2	15	Carrega a memória endereçada pelo SP com o conteúdo do registro IX.
PUSH IY	11111101 11100101	2	15	Carrega a memória endereçada pelo SP com o conteúdo do registro IY
P O P qq	11qq0001	1	10	Carrega o par de registradores qq com o conteúdo da memória endereçada pelo SP qq = BC, DE, HL, AF
P O P IX	11011101 11100001	2	14	Carrega o registro IX com o conteúdo da memória endereçada pelo SP
P O P IY	11111101 11100001	2	14	Carrega o regiatro IY com o conteúdo da memória endereçada pelo SP
CPI	11101101 10100001	2	16	Compara o conteúdo do acumulador com o endereço inicial de memória dado por HL. O par de registradores BC contém o número de bytes a serem pesquisados. Obs.: Esta instrução é realizada apenas 1 vez a cada passada do programa A - (HL), HL HL + 1, BC BC - 1
C P I R	11101101 10110001	2	16/21	Compara o conteúdo do acumulador com o endereço inicial de memória dado por HL. O par de registradores BC contém o número de bytes a serem pesquisados. Obs.: Esta instrução é realizada diversas vezes até o dado ser encontrado na memória A - (HL), HL HL + 1, BC BC - 1
C P D	11101101 10101001	2	16	Compara o conteúdo do acumulador com o endereço final da memória dado por HL. O par de registradores BC contém o número de bytes a serem pesquisados. Esta instrução é executada apenas 1 vez a cada passada do programa A - (HL), HL HL - 1, BC BC - 1
CPDR	11101101 10111001	2	16/21	Compara o conteúdo do acumulador com o endereço final de memória dada por HL. O par de registradores BC contém o número de bytes a serem pesquisados. Esta instrução é executada diversas vezes até o dado ser encontrado na memória A - (HL), HL HL - 1, BC BC - 1

3) INSTRUÇÕES ARITMÉTICAS E LÓGICAS DE 8 BITS:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
ADD A, r	10000 rrr	1	4	Soma o conteúdo do registro r ao conteúdo do acumulador e o resultado fica no acumulador. AC ← AC+r r = A,B,C,D,E,H,L
ADD A, n	11000110 n	2	7	Soma a constante "n" ao conteúdo do acumulador e o resultado fica no acumulador. AC ← AC+n
ADD A, (HL)	10000110	1	7	Soma o conteúdo da posição de memória dada por HL ao conteúdo do acumulador. O resultado fica no acumulador. AC ← AC + (HL)
ADD A, (IX+d)	11011101 10000110 d	3	19	Soma o conteúdo da posição de memória dada por (IX+d) ao conteúdo do acumulador. O resultado fica no acumulador. AC ← AC + (IX+d)
ADD A, (IY+d)	11111101 10000110 d	3	19	Soma o conteúdo da posição de memória dada por (IY+d) ao conteúdo do acumulador. O resultado fica no acumulador. AC ← AC + (IY+d)
ADC A, r	10001rrr	1	4	Soma o conteúdo do carry bit + o conteúdo do registro r com o conteúdo do acumulador. O resultado fica no acumulador. AC ← AC + (r+carry) r = A,B,C,D,E,H,L.
ADC A, n	11001110 n	2	7	Soma o conteúdo do carry bit + a constante n com o conteúdo do acumulador. O resultado fica no acumulador. AC ← AC + (n+carry)
ADC A, (HL)	10001110	1	7	Soma o conteúdo do carry bit + o conteúdo da memória endereçada por HL com o conteúdo do acumulador. O resultado fica no acumulador. AC ← AC + (HL)
ADC A, (IX+d)	11011101 10001110 d	3	19	Soma o conteúdo do carry bit + o conteúdo da memória endereçada por (IX+d) com o conteúdo do acumulador. O resultado fica no acumulador AC ← AC + ((IX + d) + carry)
ADC A, (IY+d)	11111101 10001110 d	3	19	Soma o conteúdo do carry bit + o conteúdo da memória endereçada por (IY + d) com o conteúdo do acumulador. O resultado fica no acumulador. AC ← AC + ((IY + d) + carry)
S U B r	10010rrr	1	4	O conteúdo do registro r é subtraído do acumulador e o resultado fica no acumulador. r = A,B,C,D,E,H,L
SUB n	11010110 n	2	7	A constante n é subtraída do acumulador e o resultado fica no acumulador.
SUB (HL)	10010110	1	7	O conteúdo da posição de memória dada por HL é subtraída do acumulador e o resultado fica no acumulador.
SUB (IX+d)	11011101 10010110 d	3	19	O conteúdo da posição de memória dada por (IX+d) é subtraída do acumulador e o resultado fica no acumulador.
SUB (IY+d)	11111101 10010110 d	3	19	O conteúdo da posição de memória dada por (IY+d) é subtraída do acumulador e o resultado fica no acumulador.
SBC A, r	10011rrr	1	4	O conteúdo do carry bit + o registro r são subtraídos do acumulador e o resultado fica no acumulador.
SBC A, n	11011110 n	2	7	A constante n + o carry bit são subtraídos do acumulador, e o resultado fica no acumulador
SBC A, (HL)	10011110	1	7	O conteúdo do carry bit + o conteúdo da posição de memória dada por HL são subtraídos do acumulador e o resultado fica no acumulador.

FIGURA 25

Continuação da tabela apresentada na FIGURA 25.

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	EST DOS	FUNÇÃO
SBC A, (IX+d)	11011101 10011110 d	3	19	O conteúdo do carry bit + o conteúdo da posição de memória dada por (IX+d) são subtraídos do acumulador, e o resultado fica no acumulador.
SBC A, (IY+d)	11111101 10011110 d	3	19	O conteúdo do carry bit + o conteúdo da posição de memória dada por (IY+d) são subtraídos do acumulador, e o resultado fica no acumulador.
XOR r	10101rrr	1	4	Executa uma lógica OR exclusiva entre o registro r e o acumulador. O resultado fica no acumulador. AC ← AC ● r r = A,B,C,D,E,H,L
XOR n	11101110 n	2	7	Executa uma lógica OR exclusiva entre a constante n e o acumulador. O resultado fica no acumulador. AC ← AC ● n
XOR (HL)	10101110	1	7	Executa uma lógica OR exclusiva entre a posição da memória dada por HL e o acumulador. O resultado fica no acumulador. AC ← AC ● (HL)
XOR (IX+d)	11011101 10101110 d	3	19	Executa uma lógica OR exclusiva entre a posição de memória dada por (IX+d) e o acumulador. O resultado fica no acumulador AC ← AC ● (IX + d)
XOR (IY+d)	11111101 10101110 d	3	19	Executa uma lógica OR exclusiva entre a posição de memória dada por (IY+d) e o acumulador. O resultado fica no acumulador. AC ← AC ● (IY + d)
CP r	10111rrr	1	4	Compara o conteúdo do registro r com o acumulador. Se houver igualdade o Flag "zero" é setado. Caso contrário, será resetado A - r r = A,B,C,D,E,H,L
CP n	11111110 n	2	7	Compara o valor da constante n com o acumulador, Se houver igualdade o Flag "zero" é setado. Caso contrário, será resetado A - n
CP (HL)	10111110	1	7	Compara o conteúdo da memória dada por HL com o conteúdo do acumulador, se houver igualdade o Flag "zero" é setado. Caso contrário, será resetado. A - (HL)
CP (IX+d)	11011101 10111110 d	3	19	Compara o conteúdo da memória dada por (Ix+d) com o acumulador, se houver igualdade o Flag "zero" é setado. Caso contrário, será resetado A - (IX + d)
CP (IY+d)	11111101 10111110 d	3	19	Compara o conteúdo da memória dada por (IY+d) com o acumulador, se houver igualdade o Flag "zero" é setado. Caso contrário, será resetado A - (IY+d)
INC r	00rrr100	1	4	Incrementa o conteúdo do registro r r r + 1 r = A,B,C,D,E,H,L
INC (HL)	00110100	1	11	Incrementa o conteúdo da memória dada por HL HL (HL) + 1
INC (IX+d)	11011101 00110100 d	3	23	Incrementa o conteúdo da memória dada por (IX+d) (IX + d) (IX + d) + 1
INC (IY+d)	11111101 00110100 d	3	23	Incrementa o conteúdo da memória dada por (IY+d) (IY + d) (IY + d) + 1
DEC r	00rrr101	1	4	Decrementa o conteúdo do registro r r r - 1 r = A,B,C,D,E,H,L
DEC (HL)	00110101	1	11	Decrementa o conteúdo da memória dada por HL HL (HL) - 1
DEC (IX+d)	11011101 00110101 d	3	23	Decrementa o conteúdo da memória dada por (IX+d) (IX + d) (IX + d) - 1
DEC (IY + d)	11111101 00110101 d	3	23	Decrementa o conteúdo da memória dada por (IY+d) (IY + d) (IY + d) - 1

4) INSTRUÇÕES DE ROTAÇÃO E DESLOCAMENTO:

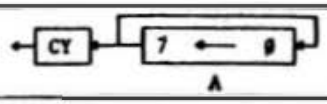
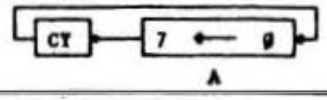
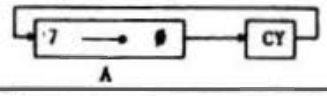
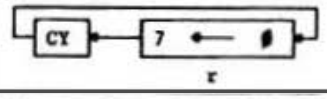
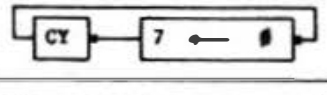
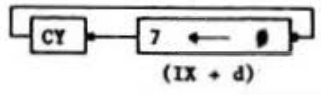
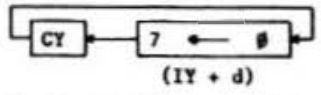
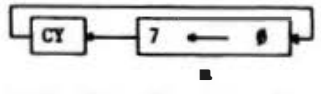
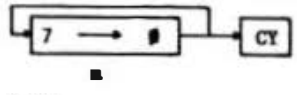
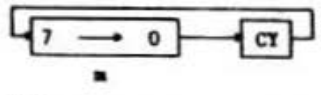
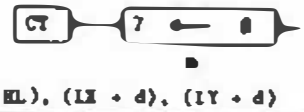
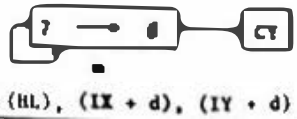
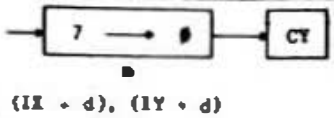
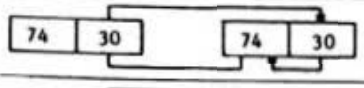
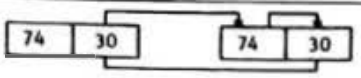
MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
RLCA	00000111	1	4	
RLA	00010111	1	4	
RRCA	00001111	1	4	
RRA	00011111	1	4	
RLC r	11001011 00000rrr	2	8	
RLC (HL)	11001011 00000110	2	15	
RLC (IX + d)	11011101 11001011 d 00000110	4	23	
RLC (IY + d)	11111101 11001011 d 00000110	4	23	
RLM	-	-	-	 m = r, (HL), (IX + d), (IY + d)
RRC m	-	-	-	 m = r, (HL), (IX + d), (IY + d)
RRm	-	-	-	 m = r, (HL), (IX + d), (IY + d)
SLA m	-	-	-	 m = r, (HL), (IX + d), (IY + d)
SRA m	-	-	-	 m = r, (HL), (IX + d), (IY + d)
SRL m	-	-	-	 m = r, (HL), (IX + d), (IY + d)
RLD	11101101 01101111	2	18	
RRD	11101101 01100111	2	18	

FIGURA 26

5) INSTRUÇÕES DE ENTRADA E SAÍDA:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	EST DOS	FUNÇÃO
INA (n)	11011011 n	2	11	Carrega o acumulador com o conteúdo do canal de entrada endereçado por N. N = 0 - 255
INr (C)	11101101 01rrr000	2	12	Carrega o registro r com o conteúdo do canal de entrada endereçado pelo registrador C. r = A,B,C,D,E,H,L
INI	11101101 10100010	2	15	Carrega a posição de memória dada por HL com o conteúdo do canal da entrada endereçado pelo registrador C. Obs.: O registro B é decrementado e o par HL é incrementado.
INIR	11101101 10110010	2	20	Carrega a posição de memória dada por HL com o conteúdo do canal de entrada endereçado pelo registrador C. Esta operação continua até o registro B chegar em zero. Obs.: o registro B é decrementado e o par HL é incrementado.
IND	11101101 10101010	2	15	Carrega a posição de memória dada por HL com o conteúdo do canal de entrada endereçado pelo registrador C. Obs.: O registro B é decrementado bem como o par HL.
INDR	11101101 10111010	2	20	Carrega a posição de memória dada por HL com o conteúdo do canal de entrada endereçado pelo registrador C. Esta operação continua até o registro B chegar em zero. Obs.: O registro B é decrementado, bem como o par HL.
OUT, (n), A	11010011 n	2	11	Carrega o canal de saída endereçado por "n" com o conteúdo do acumulador. N = 0 - 255
OUT (C), r	11101101 01rrr001	2	12	Carrega o canal de saída endereçado pelo registrador C com o conteúdo do registro r. Obs.: r = A,B,C,D,E,H,L.
OUT I	11101101 10100011	2	15	Carrega o canal de saída endereçado pelo registro C com o conteúdo da memória endereçada por HL. Obs.: O par HL é incrementado e o registro B é decrementado.
OTIR	11101101 10110011	2	20	Carrega o canal de saída endereçada pelo registro C com o conteúdo da memória endereçada por HL. Esta operação se repete até B = 0. Obs: O par HL é incrementado e B é decrementado.
OUT D	11101101 10101011	2	15	Carrega o canal de saída endereçado pelo registro C com o conteúdo da memória endereçada por HL. Obs.: O par HL é decrementado e o registro B é incrementado.
OTDR	11101101 10111011	2	20	Carrega o canal de saída endereçado pelo registro C com o conteúdo da memória endereçada por HL. Esta operação se repete até B = 0 Obs.: O par HL é decrementado e B é decrementado.

FIGURA 27

6) INSTRUÇÕES DO GRUPO DE TROCA, PESQUISA E TRANSFERÊNCIA DE BLOCO:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	EXECUÇÃO
EX DE,HL	11101011	1	4	Troca o conteúdo do par de registradores DE com HL
EX AF,AF'	00001000	1	4	Troca o conteúdo do par de registradores AF com AF'
EXX	11011001	1	4	Troca o conteúdo dos pares de registradores BC, DE, HL com BC', DE', HL'
EX(SP),HL	11100011	1	19	Troca o conteúdo da memória endereçada pelo SP com o conteúdo do par de registradores HL
EX(SP),IX	11011101 11100011	2	23	Troca o conteúdo da memória endereçada pelo SP com o conteúdo do registro IX
LDI	11101101 10100000	2	16	Carrega o conteúdo da memória dada por HL na memória dada por DE, incrementa HL, incrementa DE e decrementa BC. Obs.: HL = endereço de partida, DE= endereço de parada e BC = nº de bytes a serem transferidos.
LDIR	11101101 10110000	2	16/21	Carrega o conteúdo da memória dada por HL na memória dada por DE, incrementa HL, incrementa DE, decrementa BC. Circula até que BC = 0. Obs.: HL = endereço de partida, DE = endereço de parada e BC = nº de bytes a serem transferidos.
LDD	11101101 10101000	2	16	O mesmo que LDI, somente que transfere os blocos do fim para o começo. Obs.: HL é decrementado, DE é decrementado e BC é decrementado.
LDDR	11101101 10111000	2	16/21	O mesmo que LDIR, somente que transfere os blocos do fim para o começo. OBS: HL é decrementado, DE é decrementado e BC é decrementado.

FIGURA 28

7) INSTRUÇÕES ARITMÉTICAS DE 16 BITS:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
ADD HL, ss	00SS1001	1	11	Adiciona o conteúdo do par de registradores com o conteúdo do par de registradores HL. O resultado fica em HL. HL < HL + SS SS = BC,DE,HL,SP
ADC HL, ss	11101101 01SS1010	2	15	Adiciona o conteúdo do carry bit e o conteúdo do par de registradores SS ao conteúdo do par de registradores HL. O resultado fica em HL. HL < HL + SS + carry SS = BC,DE,HL,SP
SBC HL, ss	11101101 01SS0010	2	15	Subtrai o conteúdo do par de registradores SS mais o carry bit do conteúdo do par de registradores HL. O resultado fica em HL. HL < HL - (SS + carry) SS = BC,DE,HL,SP
ADD IX, PP	11011101 00pp1001	2	15	Soma o conteúdo do par de registradores PP ao par de registradores IX. O resultado fica em IX. IX < IX + pp pp = BC,DE,IX,SP
ADD IY, RR	11111101 00RR1001	2	15	Soma o conteúdo do par de registradores RR ao par de registradores IY. O resultado fica em IY. IY < IY + RR RR = BC,DE,IY,SP
INC SS	00SS0011	1	6	Incrementa o conteúdo do par de registradores SS. SS + 1 SS = BC,DE,HL,SP
INC IX	11011101 00100011	2	10	Incrementa o conteúdo do par de registradores IX. IX + 1
INC IY	11111101 00100011	2	10	Incrementa o conteúdo do par de registradores IY. IY + 1
DEC SS	00SS1011	1	6	Decrementa o conteúdo do par de registradores SS. SS - 1 SS = BC,DE,HL,SP
DEC IX	11011101 00101011	2	10	Decrementa o conteúdo do par de registradores IX. IX - 1
DEC IY	11111101 00101011	2	10	Decrementa o conteúdo do par de registradores IY. IY - 1

FIGURA 29

8) INSTRUÇÕES DE SALTO:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
JP nn	11000011 n n	3	10	O programa prossegue no endereço nn.
JP CC, nn	11ccc010 n n	3	10	O programa prossegue no endereço nn, somente se os flags estiverem de acordo com a especificação CC.
JR E	00011000 E - 2	2	12	O programa prossegue no endereço relativo cujo deslocador é dado no byte "E".
JRC, E	00111000 E - 2	2	7/12	O programa prossegue no endereço relativo cujo deslocador é dado no byte E, somente se o "carry" flag estiver ligado.
JRNC, E	00110000 E - 2	2	7/12	O programa prossegue no endereço relativo cujo deslocador é dado no byte E, somente se o "carry" flag estiver desligado.
JRZ, E	00101000 E - 2	2	7/12	O programa prossegue no endereço relativo cujo deslocador é dado no byte E, somente se o "ZERO" flag estiver ligado.
JRNZ, E	00100000 E - 2	2	7/12	O programa prossegue no endereço relativo cujo deslocador é dado no byte E, somente se o "ZERO" flag estiver desligado.
JP (HL)	11101001	1	4	O programa prossegue no endereço dado por HL.
JP (IX)	11011101 11101001	2	8	O programa prossegue no endereço dado por IX.
JP (IY)	11111101 11101001	2	8	O programa prossegue no endereço dado por IY.
DJNZ, E	00010000 E - 2	2	8/13	O programa prossegue no endereço relativo cujo deslocador é dado no byte "E", somente se o registro B não for igual a zero. Esta instrução se repete até B = Ø. Obs.: O registro B é usado como contador e é decrementado automaticamente.

FIGURA 30

9) INSTRUÇÕES DE CHAMADA E RETORNO:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
CALL nn	11001101 n n	3	17	O programa prossegue no endereço "nn" porém antes guarda o endereço de retorno.
CALL CC, nn	11ccc100 n n	3	10/17	O programa prossegue no endereço "nn" apenas se os flags de condição estiverem verdadeiros de acordo com a especificação CC. Obs: Guarda o endereço de retorno quando executada esta instrução.
RET	11001001	1	10	Retorna ao programa principal diretamente.
RET CC	11ccc000	1	5/11	Retorna ao programa principal apenas se os flags de condição estiverem verdadeiros de acordo com a especificação CC.
RET I	11101101 01001101	2	14	Retorna ao programa principal após uma sub-rotina de interrupção.
RET n	11101101 01000101	2	14	Retorna ao programa principal após uma sub-rotina de interrupção não mascarada
RST p	11TTT111	1	13	Realiza uma instrução de call com apenas 1 BYTE Obs.: É dirigido para apenas 8 áreas distintas da memória.

FIGURA 31

10) INSTRUÇÕES ARITMÉTICAS E DE CONTROLE DE USO GERAL:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
DAA	00100111	1	4	Ajusta o conteúdo do acumulador em 2 dígitos BCD
CPL	00101111	1	4	Complementa o acumulador (complemento de 1).
NEG	11101101 01000100	2	8	Realiza o complemento de 2 no conteúdo do acumulador.
CCF	00111111	1	4	Complementa o carry flag.
SCF	00110111	1	4	Seta o carry flag.
NOP	00000000	1	4	Nenhuma operação é executada, apenas são gastos 4 estados.
HALT	01110110	1	4	Pára o programa.
DI	11110011	1	4	Desabilita o interrupt.
EI	11111011	1	4	Habilita o interrupt.
IM Ø	11101101 01000110	2	8	Seta a máscara de interrupção no modo Ø.
IM 1	11101101 01010110	2	8	Seta a máscara de interrupção no modo 1.
IM 2	01000110 01011110	2	8	Seta a máscara de interrupção no modo 2.

FIGURA 32

11) INSTRUÇÕES DE TESTE, SET E RESET DE BITS:

MNEMÔNICO	CÓDIGO DE OPERAÇÃO (OP. CODE)	BYTES	ESTADOS	FUNÇÃO
BIT b, r	11001011 01BBBrrr	2	8	Complementa o valor do bit b do registro r, e grava no Flag "Z". B = 0 a 7
BIT b, (HL)	11001011 01BBB110	2	12	Complementa o valor do bit b do conteúdo da memória dada por HL e grava no Flag "Z" B = 0 a 7
BIT b, (IX+d)	11011101 11001011 d 01BBB110	4	20	Complementa o valor do bit b do conteúdo da memória dada por (IX + d) e grava no Flag "Z". B = 0 a 7
BIT b, (IY+d)	11111101 11001011 d 01BBB110	4	20	Complementa o valor do bit b do conteúdo da memória dada por (IY + d) e grava no Flag "Z". B = 0 a 7
SET b, r	11001011 11BBBRRR	2	8	Seta o bit b do registro r B = 0 a 7
SET b, (HL)	11001011 11BBB110	2	15	Seta o bit b da memória dada por HL. B = 0 a 7
SET b, (IX+d)	11011101 11001111 d 11bbb110	4	23	Seta o bit b da memória dada por (IX + d) B = 0 a 7
SET b, (IY+d)	11111101 11001011 d 11BBB110	4	23	Seta o bit b da memória dada por (IY + d) B = 0 a 7
RES b, S	--	-	--	Reseta o bit b do operando S S = r, (HL), (IX + d), (IY + d)

FIGURA 33

O aluno não precisará memorizar todo este conjunto de instruções. Nosso propósito aqui é simplesmente mostrar o conjunto de instruções completo, de forma que o aluno, toda vez que necessitar, deverá consultar essas tabelas.

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES
CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-14

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) A grande maioria das instruções do Z-80, para que possam executar as suas funções, necessitam:

a) do campo do operando;

b) do ciclo M_1;

c) os itens a e b estão corretos;

d) N.R.A.

2) Os operandos exigidos por uma instrução para que ela execute as suas funções, podem ser:

a) o conteúdo de registros internos da UCP;

b) dados contidos nos canais de entrada e saída;

c) posições de memórias externas;

d) os itens a, b e c estão corretos.

3) No modo de endereçamento imediato, o segundo ou terceiro byte do código da instrução é:

a) o código de operação;
b) o próprio operando;
c) muito útil, quando desejamos operar constantes de 16 bits;
d) N.R.A.

4) Normalmente as instruções com modo de endereçamento imediato, extendido possui operandos:

a) de 8 bits;
b) de 32 bits;
c) de 16 bits;
d) de 10 bits.

5) Com relação ao modo de endereçamento implícito, as instruções:

a) não possuem campo de operando;
b) possuem campo de operando;
c) possuem 16 bits;
d) N.R.A.

6) As instruções RESTART (RST T), que são em número de oito, utilizam o modo de endereçamento:

a) por registro;
b) indexado;
c) relativo;
d) página zero modificada.

7) Através do endereçamento relativo, podem-se executar:

a) Jump's condicionais;
b) Jump's incondicionais;
c) Os itens a e b estão corretos;
d) N.R.A.

8) A técnica de endereçamento extendido ou direto permite

a) o acesso a uma posição pré-determinada da memória;
b) o acesso a posições aleatórias de memória;
c) o acesso a um determinado registro;
d) N.R.A.

9) O modo de endereçamento mais completo do microprocessador Z-80 é o:

a) relativo;

b) página zero modificada;

c) extendido;

d) indexado.

10) O conjunto de instruções do microprocessador Z-80 é composto por:

a) 128 códigos de operação;

b) 256 códigos de operação; $2^8 = 256$

c) 244 códigos de operação;

d) N.R.A.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 · CEP 86100 · LONDRINA - PARANÁ

LIÇÃO MC-15

MICROPROCESSADORES Z-8000/8086

INTRODUÇÃO:

Nesta lição desenvolveremos um breve estudo sobre os microprocessadores de 16 bits. Dentre os microprocessadores de 16 bits os mais comuns, hoje em dia, são: Z-8000 da Zilog e 8086 da Intel. Vejamos então algumas características desses microprocessadores e uma rápida abordagem sobre os demais.

MICROPROCESSADOR Z-8000:

Trata-se de um microprocessador com palavra de 16 bits e de avançada arquitetura interna.

A UCP Z-8000 é oferecida em duas versões: a UCP segmentada de 48 pinos, designada Z-8001 e a UCP não segmentada de 40 pinos, designada Z-8002. A diferença principal entre os dois tipos está na variação de endereçamento. O Z-8001 pode endereçar diretamente 8 megabytes de memória; já, o Z-8002 endereça diretamente 64 Kilobytes de memória. Os dois modos de operação, "sistema" e "normal", respectivamente, e a distinção en-

tre o código, dados e espaços de empilhamento dentro de cada um dos modos permitem a expansão da capacidade de endereçamento de memória até 48 megabytes para o Z-8001 e até 384 Kilobytes para o Z-8002. Porém, em ambas as versões, a regularidade do conjunto de instruções torna a programação bastante simples.

O Z-8000 foi concebido pela tecnologia MOS de canal N (NMOS) e portas de silício, condensando 17.500 transistores em apenas 39 mm² de área. Sua densidade de 148 portas/mm² ultrapassa em muito a dos microprocessadores anteriores. Este microprocessador é alimentado por uma só fonte de 5V e requer, para sua sincronização, um pulso de clock de fase única de 4 MHZ (ou 250 ns). Devido ao fato de que para se formar um ciclo de memória é preciso pelo menos 3 ciclos da UCP, o Z-8000 trabalha com memórias que apresentam um único ciclo de 750 ns e um tempo de acesso em torno de 430 ns (nanossegundos).

A figura 1 nos mostra uma representação simplificada da pinagem do microprocessador Z-8000.

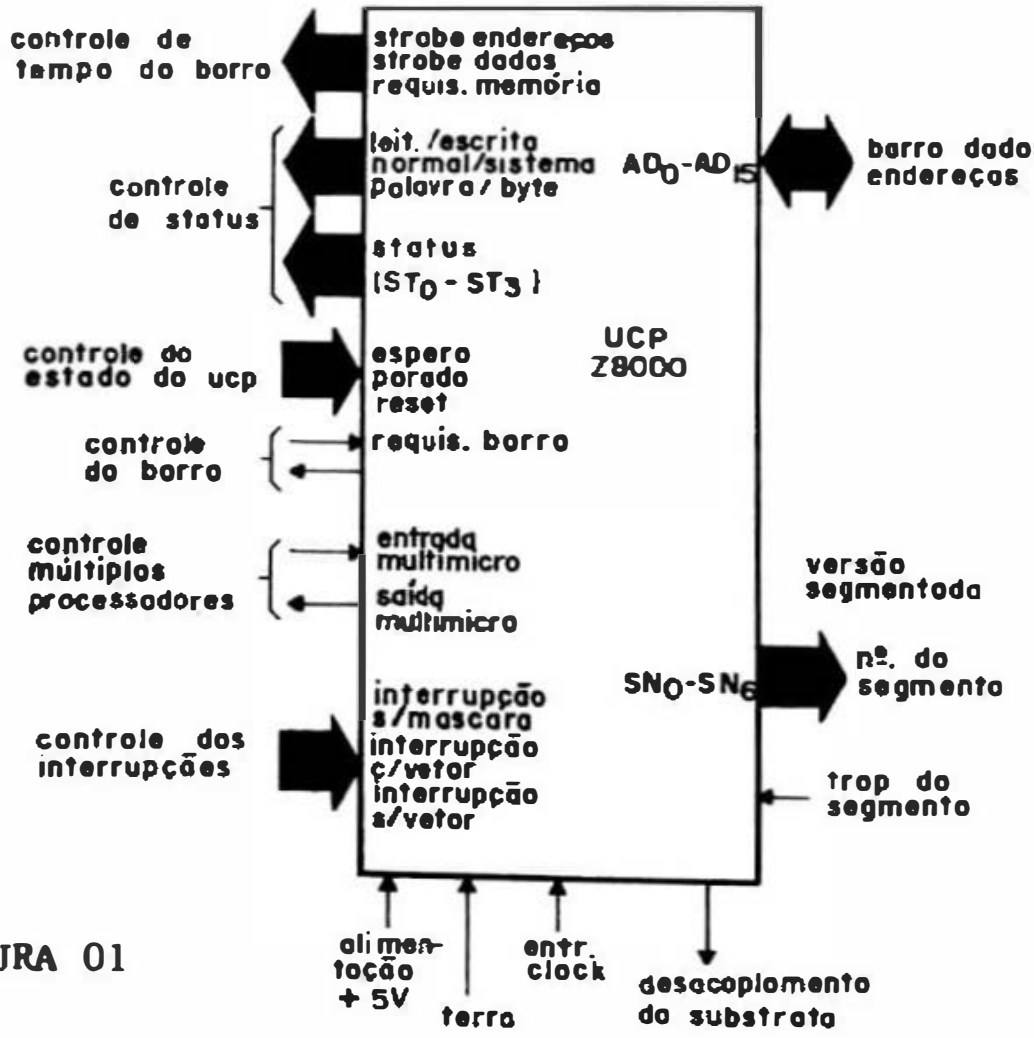


FIGURA 01

Através da figura 1 podemos observar que além do barramento de dados e endereços e das entradas de clock e alimentação, o microprocessador Z-8000 possui seis (6) tipos de barramentos de controle: de temporização de barramento, de status, de estado da UCP, de interrupção, de barramento e de múltiplos microprocessadores. As três saídas de controle de temporização do barramento são responsáveis pela coordenação do fluxo de dados nas linhas de dados e endereços do microprocessador. Um strobe de endereços sinaliza a validade dos mesmos, enquanto que um strobe de dados fornece os tempos de entrada e saída dos mesmos, em relação À UCP. A linha de requisição de memória conduz um sinal de tempo que facilita a comunicação com a memória dinâmica.

A figura 2 nos mostra o símbolo lógico do Z-8001 e Z-8002.

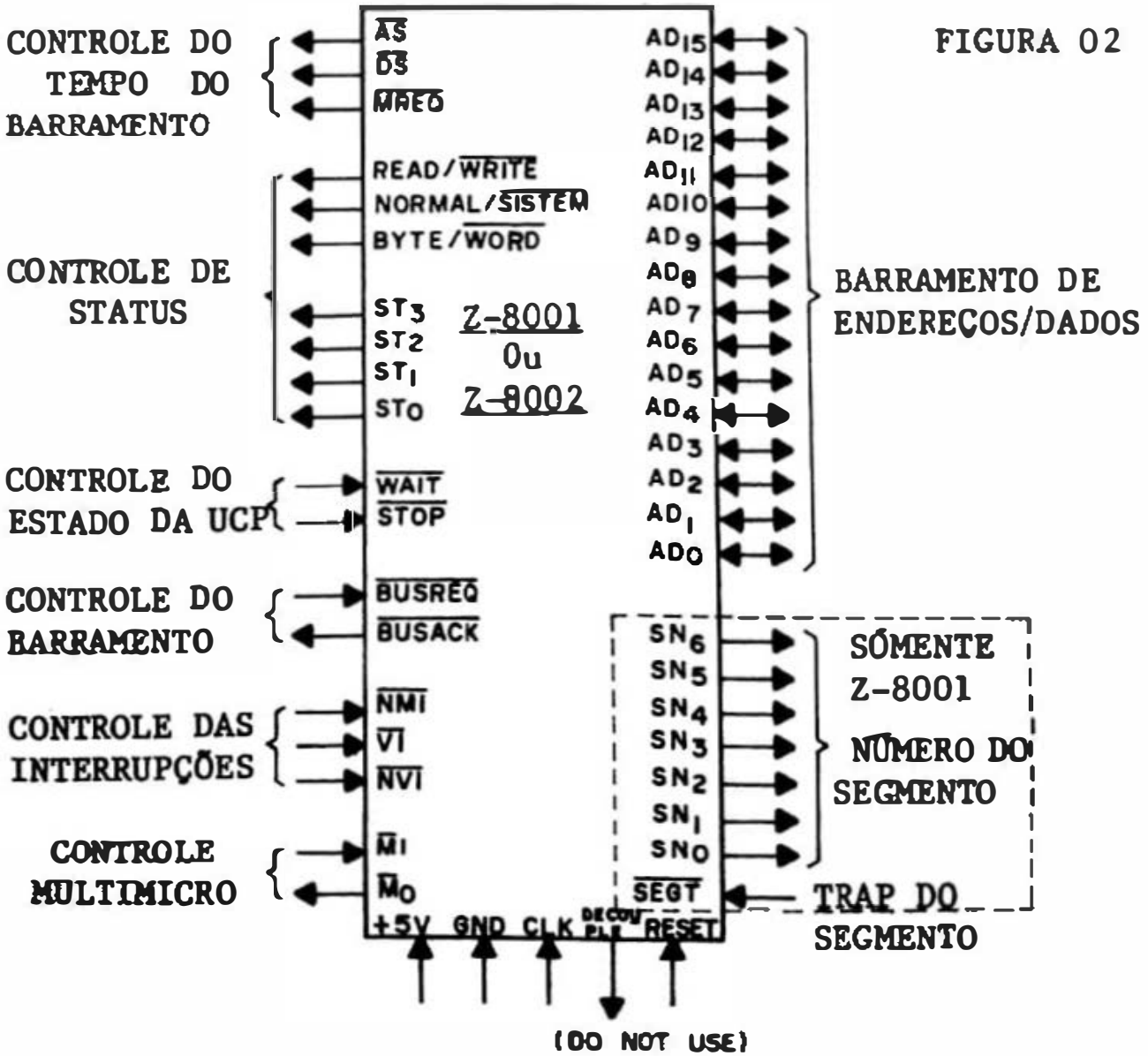


FIGURA 02

A pinagem do microprocessador Z-8001 e Z-8002 é mostrada na figura 3.

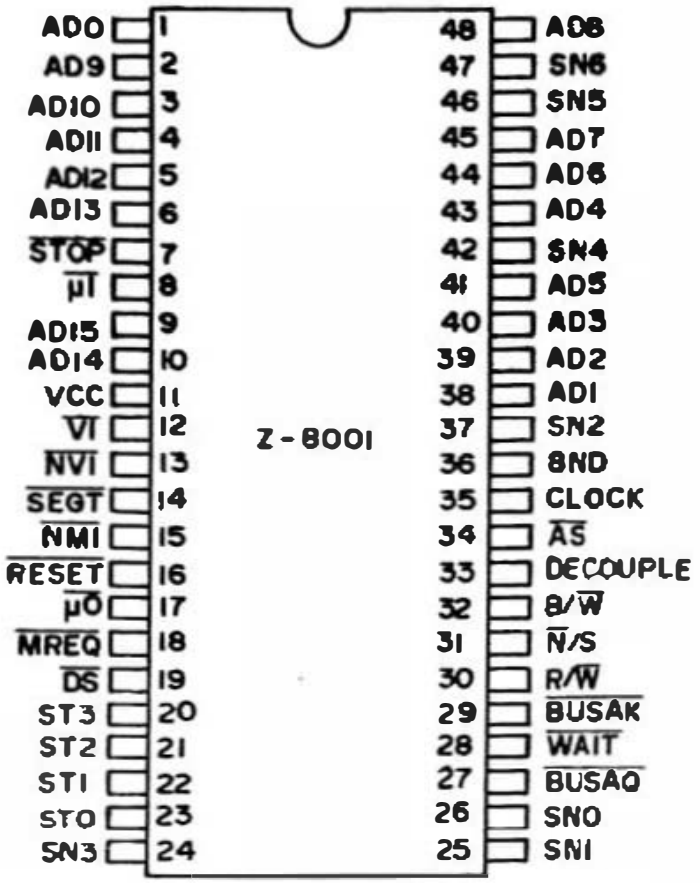
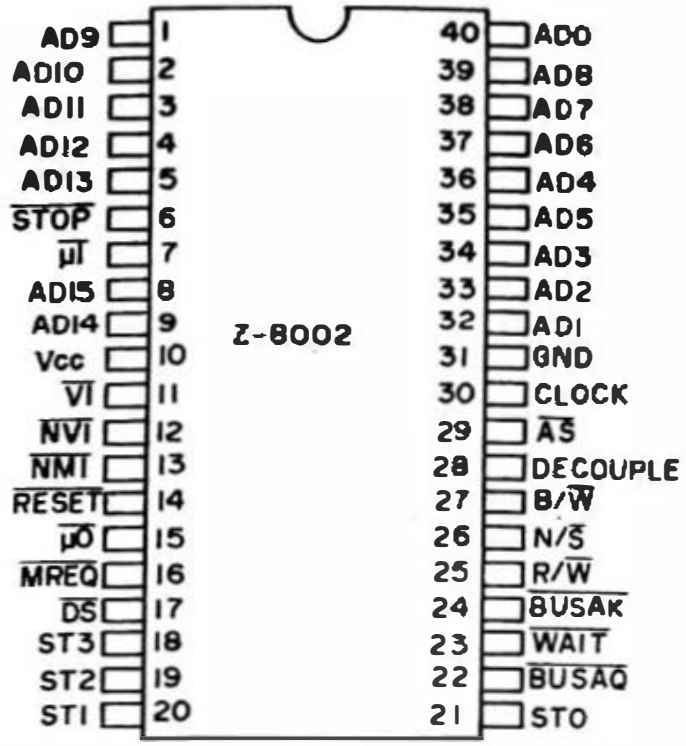


FIGURA 03



ORGANIZAÇÃO DOS REGISTROS:

A UCP Z-8000 é uma máquina de registros ou registradores orientados que oferece 16 registradores de função geral de 16 bits e uma colocação de registradores de sistemas especiais, organizados em bytes altos e baixos (R_H e R_L, respectivamente), palavras longas de 32 bits (RR) e palavras quádruplas de 64 bits (RQ).

A grande flexibilidade dos registradores foi obtida por meio de um arranjo inédito de sobreposições e pares. Os 16 registradores de 8 bits (RH_0 até RH_7 e RL_0 até RL_7), os quais podem ser utilizados como acumuladores, foram sobrepostos aos primeiros 16 registradores de 16 bits (R_0 até R_7). Já, os oito registros ou registradores de palavras longas, de 32 bits (RR_0 até RR_{14}), são formados por pares, enquanto os quatro registradores de palavras quádruplas de 64 bits (RQ_0 até RQ_{12}), os quais são utilizados por algumas instruções, como a de multiplicação, divisão e extensão de sinal, não passam de quartetos de registradores.

A figura 4 nos mostra a organização dos registradores na UCP do Z-8000.

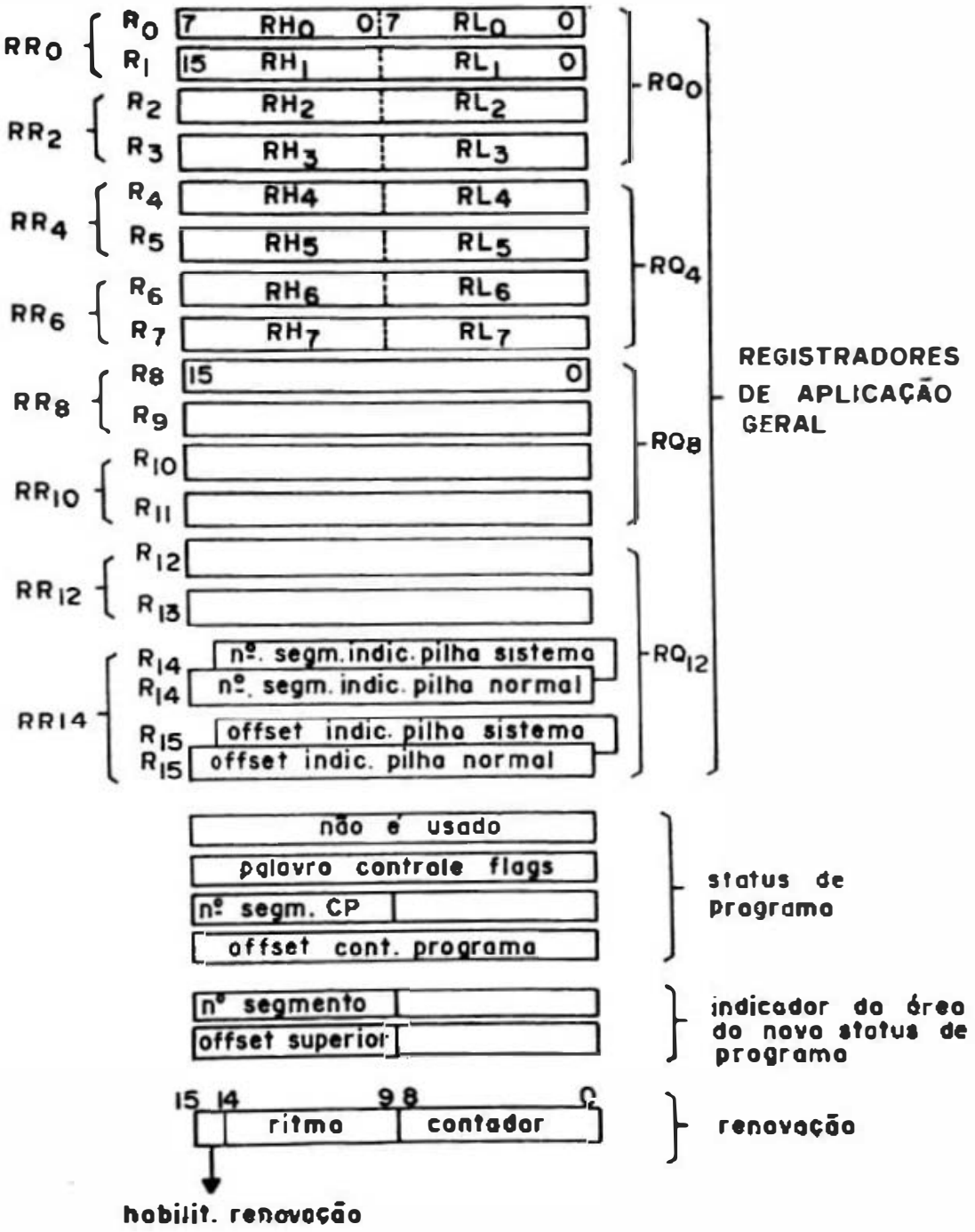


FIGURA 04

Na versão de memória não-segmentada, ou seja, o Z-8002, o último registrador de aplicação geral (R_{15}) faz as vezes do indicador de pilha. Já, na versão de memória segmentada, ou seja, no Z-8001, essa função (indicador de pilha) é exercida pelos registradores R_{14} e R_{15}, sendo o R_{14} o responsável pela armazenagem do número do segmento, e R_{15}, pelo offset. As ins

truções de chamada, chamada relativa, retorno e retorno da interrupção, são as únicas que utilizam exclusivamente o indicador de pilha. Por outro lado, as instruções POP e PUSH possuem a liberdade de utilizar qualquer registrador como indicador de pilha. Entretanto, todas as instruções possuem a capacidade de manipular o indicador de pilha, já que ele faz parte do grupo de aplicação geral.

A existência de dois conjuntos de indicadores de pilha facilita a mudança de tarefa, quando ocorre uma interrupção ou trap. A pilha normal é sempre mantida isenta de informações do sistema, já que a informação preservada na ocorrência de interrupções ou "traps" é sempre transferida para a pilha, antes que o novo status de programa seja introduzido.

Fora os registradores de uso geral, o Z-8000 apresenta outros registradores, como os registradores de status de programa o qual contém os "flags", os bits de controle e o contador de programa. Na versão Z-8002, o status do programa é mantido em dois registradores de 16 bits: o primeiro constitui o flag e a palavra de controle e o segundo, o contador de programa. Já, na versão Z-8001, o status do programa é representado por quatro palavras: o flag e a palavra de controle, um contador de programa de duas palavras e uma última palavra, reservada para uso futuro.

O Z-8000 executa suas instruções ao passar por um conjunto de ciclos básicos de máquina: leitura ou escrita na memória, escrita ou leitura de um dispositivo de entrada/saída e execução interna de dados. Como o ciclo da memória toma três ciclos de clock para buscar instruções e dados ou para escrever dados na mesma, cada ciclo de máquina dura, também, um mínimo de três ciclos de clock; porém, em se tratando de operações mais complexas, o ciclo de máquina pode durar até um máximo de oito ciclos.

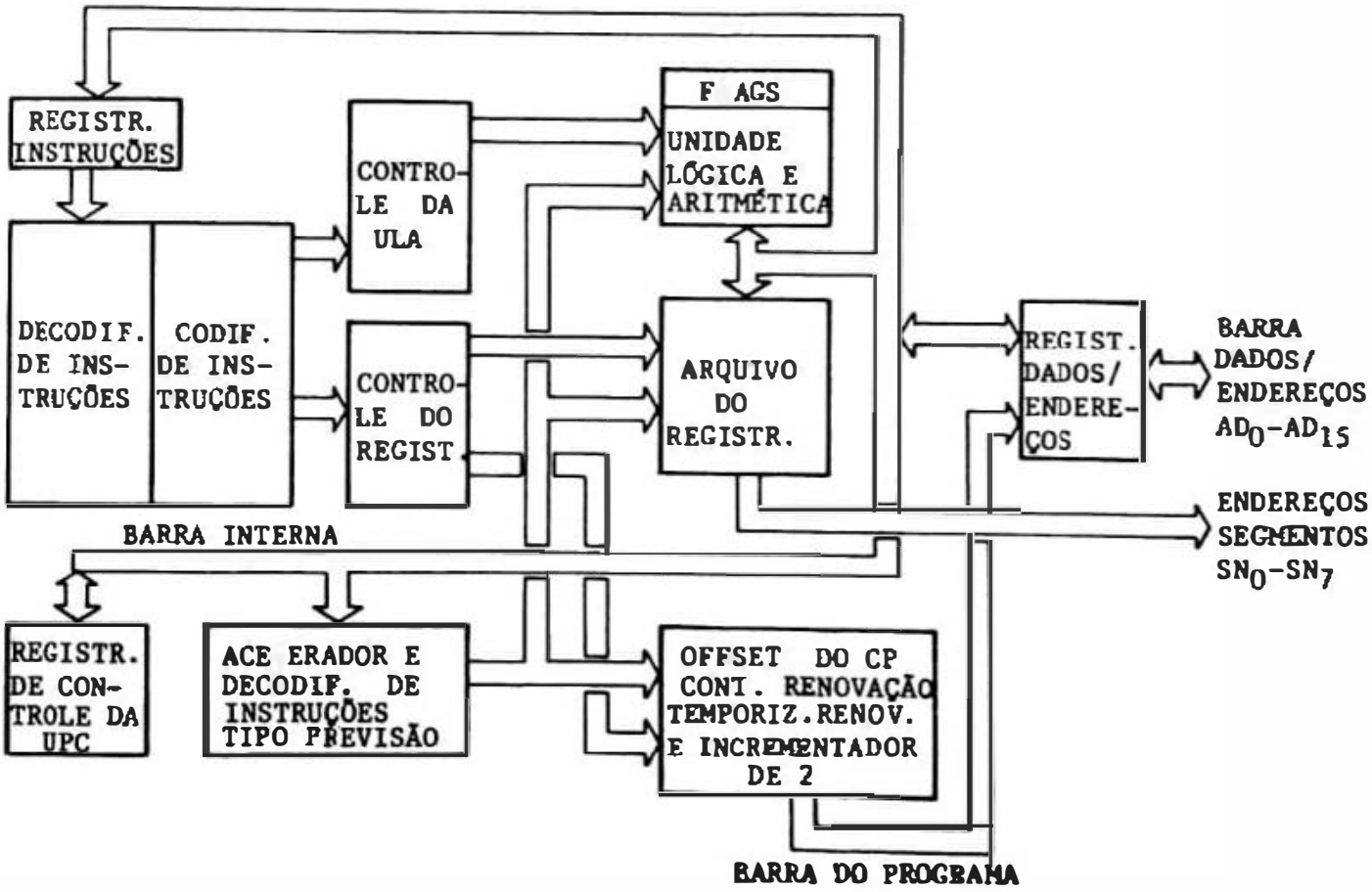
Vejamos agora a arquitetura do microprocessador de 16 bits Z-8000.

ARQUITETURA DO Z-8000:

A figura 5 nos mostra um diagrama de blocos da arquitetura do Z-8000. Uma das razões da elevada eficiência do Z-8000 está em seu decodificador e acelerador de instruções tipo "pre

visão" (look-ahead). Note que o sistema de "previsão" está diretamente relacionado com o barramento interno. Devido a este fato e graças ao conjunto de instruções, que é muito regular, uma instrução pode começar a ser executada enquanto ainda está sendo armazenada no registrador de instruções.

FIGURA 05



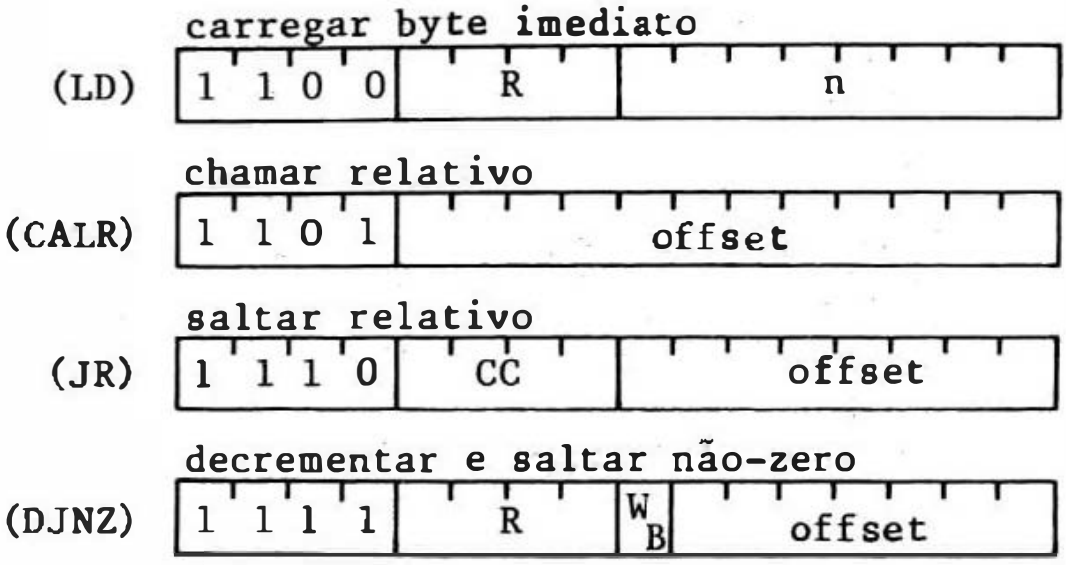
Este sistema de "previsão" constitui em um grande avanço na eficiência, pois consegue-se, assim, otimizar a instrução, reduzindo-se o número de ciclos de clock necessários ao acesso à memória, por exemplo. Já, em outros tipos de instruções, ocorre a sobreposição do tempo de manipulação de dados com a busca da palavra da instrução seguinte.

Para elevar ainda mais a velocidade de operação, além de reduzir a utilização da memória, todas as instruções mais utilizadas do Z-8000 foram codificadas numa só palavra. Por outro lado, os sofisticados blocos pré-programados e as instruções em cadeia são capazes de executarem manipulações de dados memória a memória, a uma velocidade de até 88000 bytes por segundo.

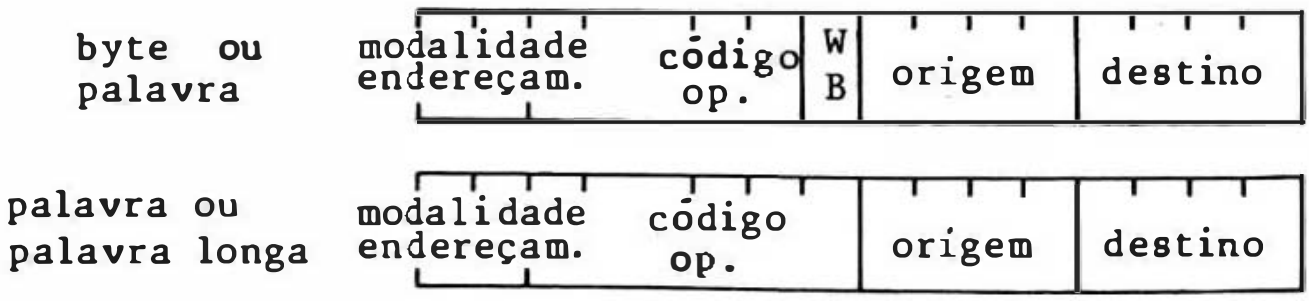
FORMATO DAS INSTRUÇÕES:

A figura 6 nos mostra os formatos adotados para as instruções do Z-8000.

FORMATO DE INSTRUÇÕES COMPACTAS



FORMATO DAS INSTRUÇÕES GERAIS



modalidade endereçam.

FIGURA 06

registrador	1 0
imediato	0 0 } origem = 0000
direto	0 1 }
registr.indireto	0 0 } origem ≠ 0000
indexado	0 1 }

Os dois bits mais significativos da palavra da instrução é que determinam o formato utilizado, compacto (a) ou geral (b). A adoção das instruções de codificação compacta (uma única palavra) além de reduzir a utilização da memória, executa com mais rapidez a tarefa em questão.

Enquanto ambos os bits mais significativos não estiverem ao nível "1", simultaneamente, permanece válido o formato geral.

O Z-8000 não possui instruções aritméticas memória a memória. Porém, ele executa transferências memória a memória através de um sofisticado conjunto de blocos pré-programados e instruções por manipulação em cadeia, além de oferecer as instruções de armazenagem, "push" e comparação imediatas.

TIPOS DE DADOS:

As instruções do Z-8000 podem operar em bits, dígitos BCD (4 bits), bytes (8 bits), palavras (16 bits), longas palavras (32 bits), sentenças de bytes e sentenças de palavras até 64 Kilobytes. Os bits podem ser posicionados, zerados e testados; os dígitos BCD são mais utilizados em operações aritméticas BCD; os bytes, por sua vez, são mais usados para caracteres ou pequenos valores integrados, instruções e endereços não segmentados; longas palavras são usadas para valores inteiros, longos e para endereços segmentados. Todos os elementos de dados com exceção das sentenças podem residir em registradores ou memórias. As sentenças são armazenadas somente na memória.

ENDEREÇAMENTO SEGMENTADO:

Um espaço de endereçamento segmentado - comparado com endereçamento linear - está próximo à maneira que um programador utiliza a memória, pois cada procedimento e espaço de dados reside no seu próprio segmento.

Os 8 megabytes do espaço de endereçamento do Z-8001 são divididos em 128 segmentos relocalizáveis até 64 Kilobytes cada um. Um endereço segmentado de 23 bits usa um endereço de segmento de 7 bits para indicar o segmento, e um offset de 16 bits para endereçar alguma localização relativa ao início do

segmento. As duas partes do endereço segmentado podem ser manipuladas separadamente. O Z-8001 segmentado pode operar algum código escrito para o Z-8002 não-segmentado em algum dos seus 128 segmentos, para tanto, este deve ser posicionado no modo não-segmentado.

DIREÇÃO DA MEMÓRIA:

O endereço manipulado pelo programador, usado pelas instruções e fornecidos pelo Z-8001 são chamados de endereços lógicos. A "unidade de direção da memória" toma os endereços lógicos e os transforma em endereços físicos os quais são utilizados para acessar a memória. A este processo de transformação de endereço dá-se o nome de relocalização. A relocalização do segmento faz os endereços de software do usuário independente da memória física, liberando desta forma, o usuário de especificar onde a informação está localizada verdadeiramente na memória física.

O processo de relocalização é "transparente" para o software do usuário. Uma tabela de translação na "unidade de direção da memória" associa o número de segmento de 7 bits com o endereço de base do segmento da memória física. O offset de 6 bits é adicionado ao endereço de base física para obtermos o endereço físico verdadeiro. O sistema pode recarregar dinamicamente as tabelas de translação conforme as tarefas sejam criadas, suspendidas ou mudadas.

A figura 7 nos mostra o esquema em forma de diagrama de blocos de uma unidade de direção da memória.

Os pares de unidades de direção suportam os 128 números de segmentos válidos para cada um dos seis espaços de endereço da UCP. Dentro de um espaço de endereço, várias unidades de direção podem ser utilizadas para criar múltiplas tabelas de translação.

PROCESSAMENTO EXTENDIDO:

A arquitetura de processamento extendido, criada pela Zilog (EPA), fornece uma aproximação extremamente popular e flexível para expandir ambas as capacidades do "software" e do "hardware" da UCP Z-8000. As características da EPA incluem:

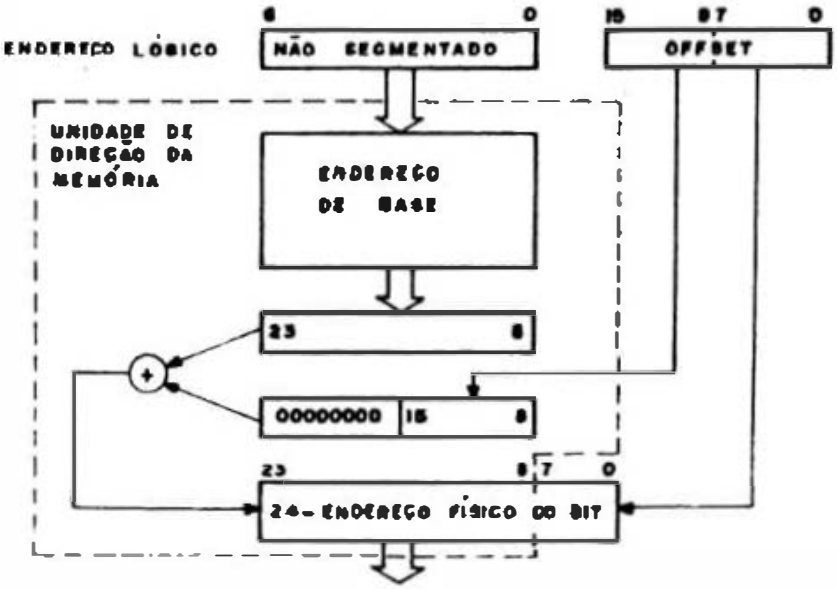


FIGURA 07

a) - As instruções especializadas para os processadores externos e traps de software podem ser adicionadas a colocação de instrução da UCP;

b) - Aumento do desempenho do sistema por utilizar até quatro processadores externos especializados, em paralelo com a UCP;

c) - Permite projeto modular dos sistemas baseados no Z-8000;

d) - Fornece fácil direção de múltiplas configurações do microprocessador via comunicação de "corrente de instrução única";

e) - Suporta correção (DEBUGGING) do hardware suspeito contra o software provado;

f) - Os EPU's podem ser adicionados conforme o sistema é expandido ou conforme as EPU's com funções especializadas são desenvolvidas;

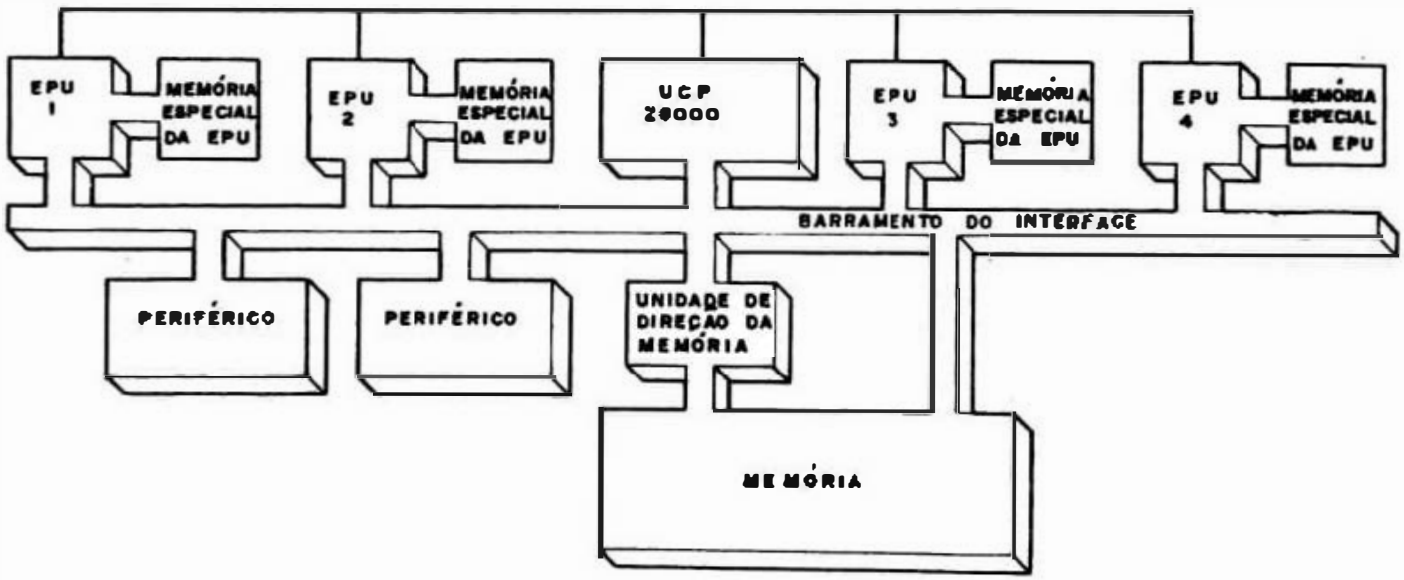
g) - O controle das EPU's é feito via "corrente de instrução única" na UCP do Z-8000, eliminando muito software do sistema significativo e obstáculos de direção do barramento que ocorrem em outros esquemas de organização de multiprocessadores.

Podemos, portanto, aumentar a potência de processamento do microprocessador de 16 bits Z-8000, para além da sua capacidade intrínseca através do uso da arquitetura de processamento extendido. O Z-8000 permite acomodar até quatro unidades de processamento extendido (EPU's), para executarem funções especializadas em paralelo com a corrente de execução da instrução principal da UCP.

O uso dos processadores extendidos, para aumentar a capacidade de execução da UCP, tem sido aprovada com grande sucesso em computadores e minicomputadores. Nestes sistemas, as funções especializadas, tais como o processamento da disposição, o processamento de entrada/saída especial e comunicação de dados são marcados para o HARDWARE do processador extendido.

A figura 8 nos mostra a configuração de um processador extendido utilizando, como UCP, o Z-8000 a qual permite um processamento paralelo muito eficaz.

FIGURA 08



Neste caso (figura 8), a UCP é responsável por instruir a EPU e entregar os operandos e dados a este.

A arquitetura de processamento extendido oferece, ainda, proteção contra instrução extendida sobreposta. Cada EPU conecta-se à UCP do Z-8000 através do barramento STOP, de modo que se uma EPU é requisitada para executar uma segunda instrução extendida, antes ela terá que executar a anterior, levan-

do a UCP ao estado STOP/REFRESH até que seja executada a primeira instrução extendida. Depois que esta instrução extendida for executada, a EPU desativa a linha STOP e as transações da UCP continuam normalmente.

O microprocessador Z-8000 vem acompanhado de uma série de circuitos integrados que o complementa: Z-SIO (controlador série), Z-MBU e Z-FIFO (buffer do barramento), Z-UPC (controlador inteligente de periféricos), Memórias, etc.

Os controladores de periféricos são baseados no Z-8 que, além da unidade central de processamento, contém 2 K ROM, 128 octetos de RAM, 32 linhas de entrada e saída, uma UART programável e dois contadores ou temporizadores.

Por outro lado, o Z-8000 dispõe de sistemas de desenvolvimento especiais ou derivados do Z-80, com compiladores (PL-Z, Pascal, Cobol,...). Também se encontra disponível programas de conversão de assembler do Z-80 para o código objeto do Z-8000.

O desenvolvimento do Z-8000 partiu do microprocessador tradicional, orientado para bytes e adotou a arquitetura bem mais regular dos minicomputadores.

Incluindo, assim, muitas das características estruturais dos minicomputadores, o Z-8000 é indicado para as aplicações da área, tanto de minicomputadores, como de microcomputadores. Seus 81 códigos distintos de operação, combinados aos vários tipos de dados e modalidades de operação, formam um conjunto de 414 instruções, mais poderoso do que o conjunto de instruções de muitos minicomputadores.

O Z-8000 procurou satisfazer, como já vimos, tanto as necessidades de minicomputadores, como as exigências já existentes com relação aos microcomputadores. Trata-se, portanto, de um microprocessador muito versátil e de grande futuro.

MICROPROCESSADOR 8086:

Este é o iniciador de uma nova geração de microprocessadores de altos préstimos juntamente com o Z-8000 e o MC 68000. Trata-se, portanto, de um microprocessador de 16 bits, fabricado pela Intel com base na tecnologia H-MOS. Sua capacidade de direcionamento é de 1 megabytes. Contém, internamente, um

sistema de segmentação da memória, podendo operar em três (3) espaços de direcionamento, para código, dados e stack. Não dispõe de uma diferenciação efetiva entre o modo sistema e o modo programa, apresentando algumas limitações para sua utilização em sistemas multiusuário (ou multiprocessadores).

Como todos os microprocessadores, o 8086 produz ou recebe os seguintes tipos de sinais:

- linhas de endereço;
- linhas de dados;
- linhas de estado e de controle;
- linhas de sincronização e de alimentação.

A figura 9 nos mostra o símbolo lógico do 8086 com a respectiva designação dos pinos e dos sinais.

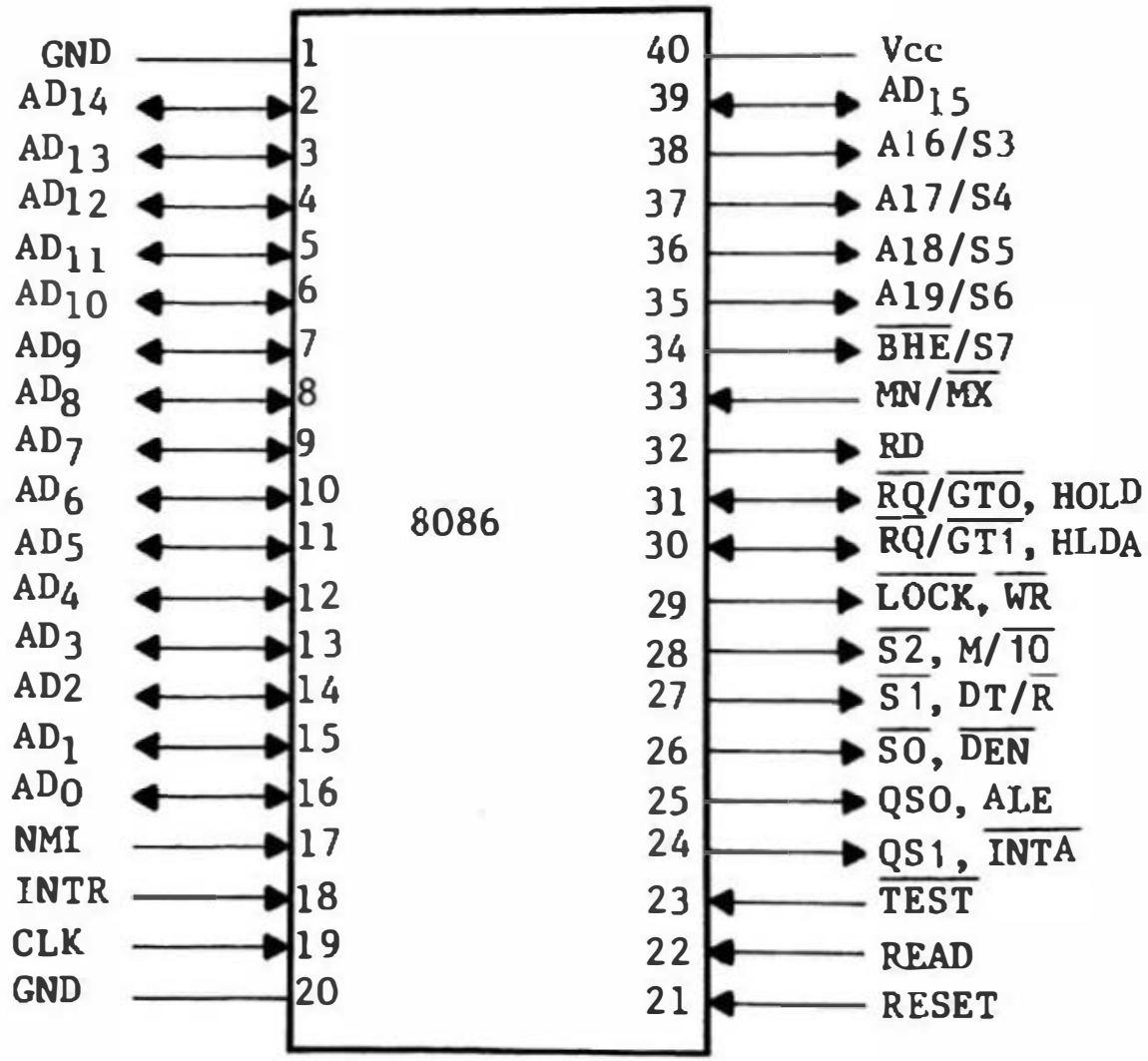


FIGURA 09

Vejamos, em seguida, uma rápida e resumida descrição dos pinos do microprocessador 8086.

DESCRIÇÃO DA PINAGEM DO 8086:

A descrição que segue mostra a função de cada pino quando MN/$\overline{MX}$ estiver alto ou baixo.

NOME DO PINO		DESCRIÇÃO	TIPO
AD_0 - AD_{15}		Barramento de endereços e de dados	Três-estados, bidirecional
A_{16}/S_3, A_{17}/S_4		Identificador do endereço/ segmento	Saída três-estados
A_{18}/S_5		Estado de habilitar Interrupções/endereços	Saída três-estados
A_{19}/S_6		Estados/endereços	Saída três-estados
$\overline{BHE}/S_7$		Estado/byte de alta ordem	Saída três-estados
$\overline{RD}$		Controle de leitura	Saída três-estados
READY		Requisição do estado de espera	Entrada
$\overline{TEST}$		Esperar pelo controle de teste	Entrada
INTR		Requisição de interrupção	Entrada
NMI		Requisição de interrupção NON-MASKABLE	Entrada
RESET		Zerar o sistema	Entrada
CLK		Clock do sistema	Entrada
MN/$\overline{MX}$	modo de operação máximo	=GND para um sistema máximo	
$\overline{S_0}$, $\overline{S_1}$, $\overline{S_2}$		Estados do ciclo de máquina	Saída três-estados
$\overline{RQ}/\overline{GT_0}$; $\overline{RQ}/\overline{GT_1}$		Controle de prioridade do barramento total	Bidirecional
QS_0, QS_1		Estados da instrução	Saída
$\overline{LOCK}$		Controle de manter o barramento	Saída três-estados

MN/$\overline{MX}$	modo de operação mínimo	=Vcc para um sistema mínimo	
M/$\overline{IO}$		Acesso da memória ou I/O (E/S)	Saída três-estados
$\overline{WR}$		Controle de escrita	Saída três-estados
ALE		Habilitação da trava de endereço	Saída
DT/$\overline{R}$		Recepção/transmissão de dados	Saída três-estados
$\overline{DEN}$		Habilitação de dados	Saída três-estados
$\overline{INTA}$		Admissão de interrupção	Saída três-estados
HOLD		Requisição HOLD (manter)	Entrada
HLDA		Admissão HOLD (manter)	Saída
Vcc		Alimentação (+ 5 V ± 10%)	
GND		Terra (existe dois pinos terra)	

A UCP 8086 pode endereçar, como já foi dito, até um milhão de bytes (1 megabytes) de memória. Isto significa que 20 bits de informação de endereço são necessários.

A UCP 8086 acessa dados em unidades de 16 bits, tratados como um byte de baixa ordem e um byte de alta ordem.

Para permitir um barramento de endereço de 20 bits e um barramento de dados de 16 bits em uma encapsulação de apenas 40 pinos, o barramento de dados teve que ser multiplexado com os 16 bits menos significativos do barramento de endereço. As quatro linhas de endereço adicionais são multiplexadas com a informação de estado. Os sinais de estado/endereço e dados/endereço são:

- AD_0 - AD_{15}
- A_{16}/S_3
- A_{17}/S_4
- A_{18}/S_5
- A_{19}/S_6
- $\overline{BHE}/S_7$

A freqüência do sinal de clock do 8086 pode ser de 8/5/4 megahertz.

A arquitetura do 8086 consiste em dois processadores assíncronos. Um se encarrega da interface com o barramento e contém um suplemento, no qual se pode guardar até seis (6) octetos extraídos da memória antecipadamente. O segundo processador se encarrega da execução das instruções. Esta arquitetura apresenta a vantagem de permitir uma velocidade de execução das instruções extremamente alta sem exigir que a memória tenha um tempo de acesso muito pequeno.

Sua arquitetura, juntamente com a tecnologia H-MOS de grande velocidade, permite-nos conseguir um ciclo de instrução de 400 ns (400 nanossegundos).

O 8086 é capaz de suportar configurações de multiprocessador, dispondo para tanto, dos sinais LOCK e TEST para proteção e sincronização. O repertório de instruções é muito completo e potente, incluindo multiplicação e divisão que permitem o manejo de diversos tipos de dados (BCD, bit, octeto, palavra, sentenças) e diversos modos de direcionamento (24 no total).

O 8086 dispõe também de uma ampla gama de circuitos adicionais que complementam a família. Entre eles podemos destacar o 8089 que é um controlador de canal com acesso direto à memória e que contém uma unidade central, baseada no próprio microprocessador 8086, com instruções especializadas para a execução de programas de canal.

ORGANIZAÇÃO DOS REGISTROS:

O microprocessador 8086 possui um total de 14 registradores de 16 bits. A figura 10 nos mostra um esquema orientativo da organização dos mesmos.

De um modo geral, os registradores do 8086 podem ser considerados como uma extensão dos registros (ou registradores) do 8085.

Os registradores de dados podem ser utilizados sem constrangimento na maioria das operações lógicas e aritméticas. Os registradores POINTER e INDEX também podem participar na maioria das operações lógicas e aritméticas. De fato, em todos os oito (8) registradores gerais encaixa a definição de acumula-

dor quando utilizados em microprocessadores de primeira e segunda geração.

Um fato interessante a mencionar é o uso dos registros para comprimentos de dados maiores que 16 bits. Para tanto, os primeiros quatro registradores podem ser endereçados como duas seções individuais de 8 bits cada. Efetivamente, isto significa que eles podem ser utilizados como quatro registradores de 16 bits ou oito registradores de 8 bits, ou qualquer outra combinação.

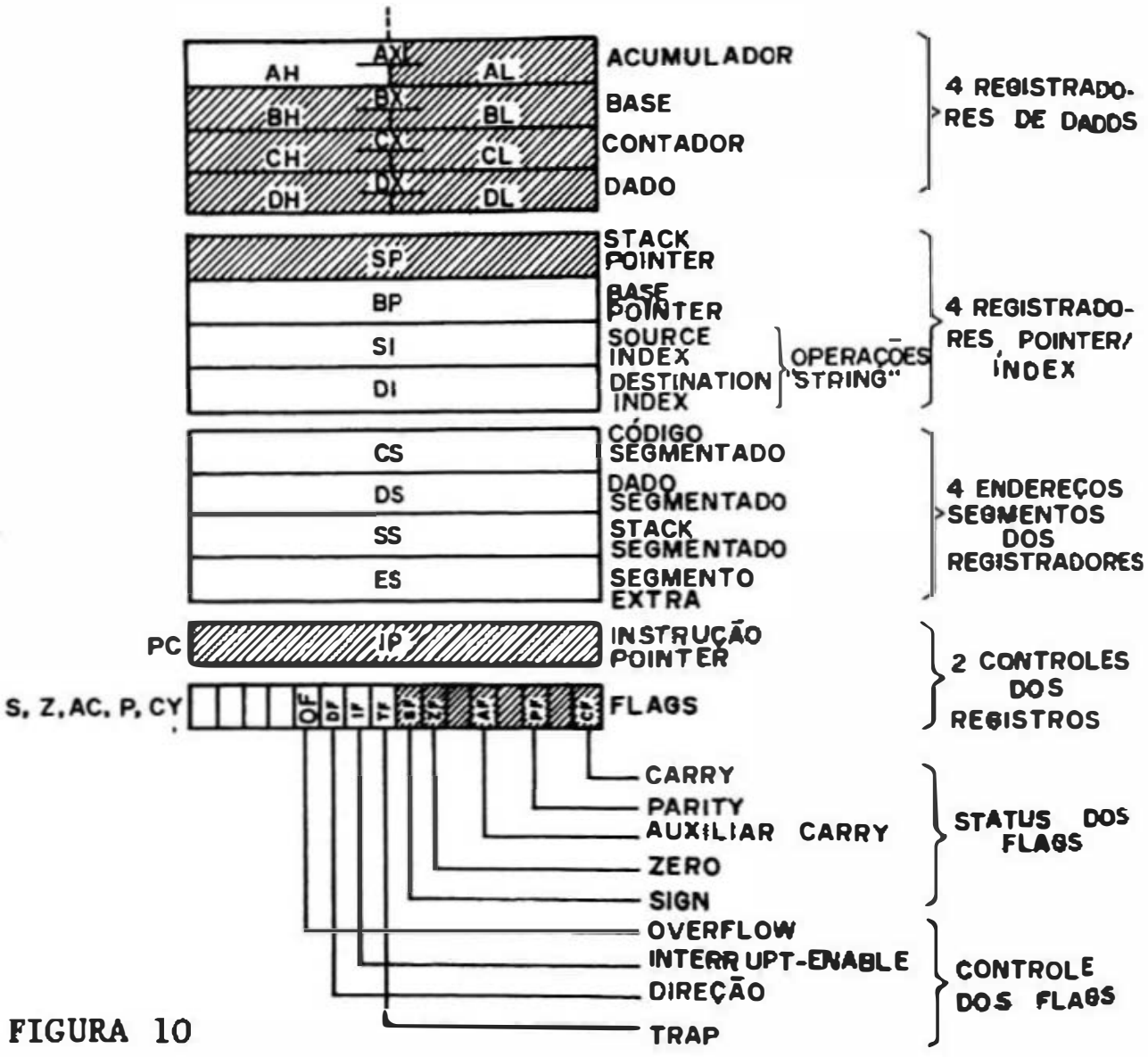


FIGURA 10

EXEMPLO DE CONFIGURAÇÃO UTILIZANDO A UCP 8086:

A figura 11 nos mostra o diagrama de blocos de um sistema que utiliza uma unidade central de processamento 8086.

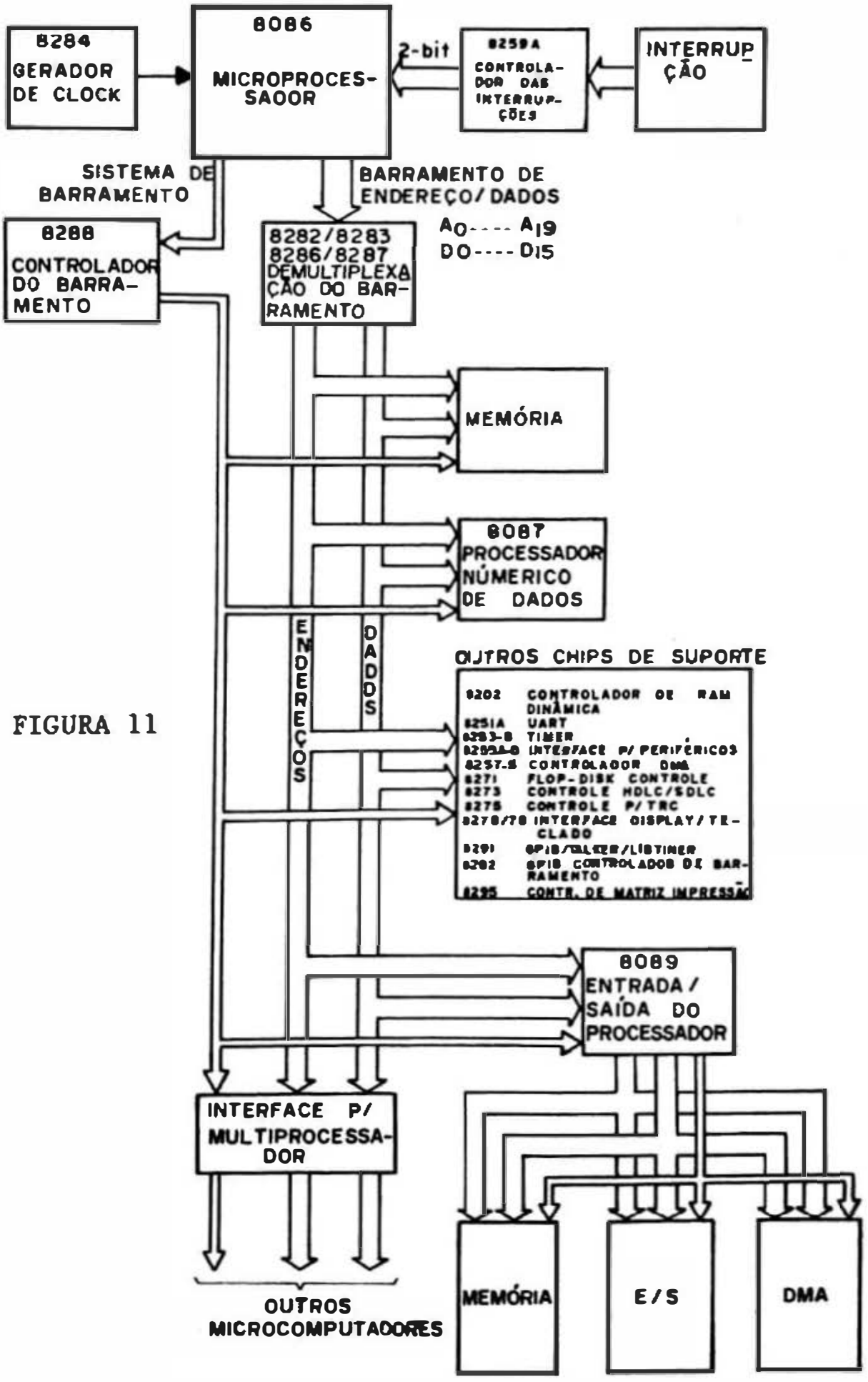

8284
GERADOR DE CLOCK
8086
MICROPROCESSAOOR
2-bit
8259A
CONTROLADOR DAS INTERRUPÇÕES
INTERRUPÇÃO
SISTEMA DE BARRAMENTO
BARRAMENTO DE ENDEREÇO/DADOS
A0---- A19
D0---- D15
8288
CONTROLADOR DO BARRAMENTO
8282/8283 8286/8287 DEMULTIPLEXAÇÃO DO BARRAMENTO
MEMÓRIA
8087
PROCESSADOR NÚMERICO DE DADOS
ENDEREÇOS
DADOS
OUTROS CHIPS DE SUPORTE
8202 CONTROLADOR DE RAM DINÂMICA
8251A UART
8253-5 TIMER
8255A-5 INTERFACE P/ PERIFÉRICOS
8257-5 CONTROLADOR DMA
8271 FLOP-DISK CONTROLE
8273 CONTROLE HDLC/SDLC
8275 CONTROLE P/ TRC
8278/79 INTERFACE DISPLAY/TECLADO
8291 GPIB/TALKER/LISTINER
8292 GPIB CONTROLADOR DE BARRAMENTO
8295 CONTR. DE MATRIZ IMPRESSÃO
8089
ENTRADA/SAÍDA DO PROCESSADOR
INTERFACE P/ MULTIPROCESSADOR
OUTROS MICROCOMPUTADORES
MEMÓRIA
E/S
DMA

FIGURA 11

O microprocessador 8086 faz parte da família IAPX-86 da Intel, a qual inclui vários processadores auxiliares.

OUTROS MICROPROCESSADORES DE 16 BITS:

Existe no mercado uma série de microprocessadores de 16 bits. A tabela apresentada na figura 12 nos mostra os principais tipos de microprocessadores.

FIGURA 12

TIPO	ORIGEM (FABRICANTE)	PROCESSO DE FABRICAÇÃO	APLICAÇÃO
MN601	Data General	NMOS	Minicomputadores OEMs
9440	Fairchild	I^2L	Minicomputadores OEMs
F100L	Ferranti	Bipolar	Militar
CP1600	General Instruments	NMOS	Jogos eletrônicos
8086	Intel	HMOS	Microprocessador de propósito geral
MC68000	Motorola	NMOS	Microprocessador de propósito geral
NS16032	National Semicondutor	XMOS	Microprocessador de propósito geral
MN1610	Panafacom	NMOS	
TMS9900	Texas Instruments	NMOS	Microprocessador de propósito geral
WD16	Western digital	NMOS	Minicomputadores OEMs
Z-8001	Zilog	NMOS	Microprocessador de propósito geral

Todos eles possuem seus pontos fortes e seus pontos fracos, porém cada um deles é capaz de executar praticamente todas as tarefas impostas a um determinado microcomputador. Portanto, torna-se difícil eleger qual deles é o mais eficiente, pois para cada tipo de aplicação um desses microprocessadores, com certeza, reunirá melhores condições para realizá-la, o que

nos induz a dizer que,neste caso específico,o microprocessador X é o melhor (ou o mais indicado), isto é claro, sob o ponto de vista puramente técnico.

Porém, para realizarmos uma escolha mais acertada, é necessário levarmos em consideração dois outros fatores importantíssimos, a saber:

a) Colocação de instrução de linguagem de máquina

A colocação de instrução sob a forma de linguagem de máquina deve ser a mais simples, clara e direta possível. Deve-se observar o que os fabricantes apresentam e o que os microprocessadores são,realmente, capazes de fazer.

b) Preço e validade

Como não poderia deixar de ser, o fator econômico também influencia na decisão de escolha do microprocessador. O preço varia muito, portanto, é aconselhável fazer contato com vários fornecedores.

Concluindo, o aluno pode achar que todos estes microprocessadores são semelhantes, mas o que podemos garantir é que todos eles são muito mais poderosos do que os microprocessadores de 8 bits. Qualquer escolha entre eles, quando do desenvolvimento de um projeto qualquer, deve ser baseada em uma grande extensão de testes.

Os microprocessadores 68000, 160000 e Z-8000 estão todos muito próximos um do outro, isso, em termos de estrutura e de capacidade. Já, o microprocessador 8086, apesar de ser um microprocessador de 16 bits, está mais próximo, sob alguns aspectos, da prática dos microprocessadores de 8 bits, o que pode ser considerado como uma vantagem ou uma desvantagem,dependendo do ponto de vista com que se está estudando o mesmo.

A tendência sobre o futuro dos microprocessadores, é cada vez melhorar mais,tornando-os mais potentes, mais velozes e de baixo custo.

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES
CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANA

EXAME DA LIÇÃO MC-15

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) A versão Z-8001 com 48 pinos, é dita:

a) segmentada;

b) não segmentada;

c) de endereçamento indireto;

d) N.R.A.

2) O modo de operação da versão Z-8001, de 40 pinos, é dita:

a) sistema;

b) normal;

c) paranormal;

d) N.R.A.

3) A tecnologia utilizada para a fabricação do microprocessador Z-8000 é:

a) HMOS;

b) XMOS;

c) I^2L;

d) NMOS.

4) A UCP Z-8000 possui:

a) 8 registradores de função geral de 8 bits cada;

b) 32 registradores de função geral de 16 bits cada;

c) 16 registradores de função geral de 16 bits cada;

d) N.R.A.

5) No Z-8002, o último registrador de aplicação geral (R_{15}) faz as vezes:

a) do contador de programa;

b) do indicador de pilha;

c) dos flags;

d) N.R.A.

6) A grande eficiência do Z-8000 se deve, em parte, ao:

a) decodificador e acelerador de instruções, tipo previsão;

b) enorme volume de memória, tipo RAM;

c) os itens a e b estão corretos;

d) N.R.A.

7) As instruções do Z-8000 podem operar em:

a) bits;

b) dígitos BCD;

c) bytes;

d) os itens a, b e c estão corretos.

8) A EPA (arquitetura de processamento extendido), criada pela Zilog, permite:

a) aumento do desempenho do sistema por utilizar até quatro processadores externos especializados, em paralelo com a UCP;

b) o projeto modular dos sistemas baseados no Z-8000;

c) Os itens a e b estão corretos;

d) N.R.A.

9) A tecnologia utilizada na fabricação do microprocessador 8086 é a:

a) NMOS;

b) I^2L;

c) XMOS;

d) HMOS.

10) A capacidade de endereçamento da UCP 8086 é da ordem de:

a) 10 Megabytes;

b) 1 Megabyte;

c) 1 Kilobyte;

d) N.R.A.

CEBRAM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP86100 - LONDRINA - PARANA

LIÇÃO MC-16

TELEPROCESSAMENTO

INTRODUÇÃO

Basicamente, o "Teleprocessamento" vem a ser uma modalidade de tratamento da informação, na qual um sistema de processamento de dados utiliza-se dos meios empregados nas telecomunicações. Este tipo de tratamento implica na transmissão dos dados do ponto em que são gerados até o computador central, para seu processamento e a retransmissão subseqüente dos resultados ao ponto de partida. Com base na definição apresentada e, de um modo muito mais amplo, pode-se dizer que a "informática" e a "comunicação" (ou telecomunicação) são tecnologias que se convergem, tanto do ponto-de-vista de sua utilização, quanto do desenvolvimento e fabricação dos equipamentos. E esta "semelhança" é tão expressiva que se tende a confundir, conseqüentemente, no plano dos negócios e, além disso, começam a criar o que os especialistas chamam de "telemática": pedra na qual se fundamenta a sociedade pós-industrial ou sociedade da informação.

Este novo ambiente que se cria, baseado na telemática, envolve e integra centrais telefônicas de processamento armazenado, satélites, microondas, fibras óticas, redes públicas, banco de dados e, finalmente, toda uma tecnologia de ponta,

com linguagem orientada ao usuário final. Toda essa estrutura poderá levar a humanidade a uma nova revolução industrial e a um novo estilo de vida. Por outro lado, se colocarmos toda essa infra-estrutura num meio ambiente "predatório", as conseqüências serão desastrosas, de modo que, para viabilizar-se a chamada "sociedade pós-industrial", um novo pacto social deverá ser negociado, pois, sem ética, seremos todos vítimas de nossas próprias criações.

Nesta lição, procuraremos dar um pequeno enfoque às categorias de sistemas de transmissão de dados e, logo em seguida, colocaremos em estudo um dispositivo eletrônico de grande utilização, atualmente, na transmissão e recepção de informações (dados), conhecido pelo nome de MODEM. Não entraremos em maiores detalhes sobre teleprocessamento, pois o assunto é muito amplo, o qual também pode ser abordado sobre vários pontos-de-vista, de modo que para desenvolvermos um estudo um pouco mais aprofundado, necessitaríamos de várias lições, além de estarmos nos distanciando um pouco do real objetivo do nosso curso. Passemos então ao nosso estudo.

CATEGORIAS DE "SISTEMAS DE TRANSMISSÃO DE DADOS"

Os sistemas de transmissão de dados são construídos para uma larga variedade de propósitos, diferenciando-se um do outro de acordo com o meio onde operam.

Provavelmente, no futuro, a forma mais comum de sistema será aquele que pessoas operando terminais, comunicam-se com computadores distantes, os quais responderão rapidamente as consultas. Assim, podemos verificar que ocorrerá um diálogo entre o usuário do terminal e o computador remoto. Este tipo de aplicação não existia para os primeiros cem anos de transmissão de dados, portanto muitas das idéias tradicionais estão sendo repensadas, mudando, assim, alguns conceitos, principalmente, no que diz respeito às categorias de sistemas de transmissão de dados. Vejamos, agora, o que venham a ser sistemas on-line e off-line.

1) SISTEMAS ON-LINE E OFF-LINE

Muitos dos sistemas existentes hoje em dia não são, quanto à natureza, do tipo real-time (tempo certo), mas são requ

ridos para moverem uma certa quantidade de dados de um ponto a outro. Nestes casos, as ligações de comunicação para um computador podem ser "on-line" ou "off-line". On-line significa que os dados transmitidos vão diretamente para dentro do computador, e este controla a transmissão. Já, off-line significa que os dados transmitidos não vão diretamente para dentro do computador, mas são escritos em fita magnética ou em um disket ou, ainda, esses dados são perfurados em fitas de papel ou cartões para processamento posterior.

Um sistema on-line pode ser definido como sendo um sistema, no qual os dados de entrada (input) partem de sua origem e entram diretamente no computador e/ou os dados de saída (output) são transmitidos diretamente para onde eles serão utilizados. Neste tipo de sistema, os estágios intermediários de perfuração de dados nos cartões ou fita de papel ou ainda a gravação em fita magnética são evitados, pois são característicos nos sistemas off-line.

2) SISTEMAS INTERATIVOS E NÃO INTERATIVOS

Normalmente os sistemas off-line não são interativos, - pois nenhum computador é diretamente conectado ao local de origem dos dados. Nenhuma resposta (de dados) será recebida pelo ponto de origem dos dados, muito embora sinais simples de controle possam ser recebidos para controlar o funcionamento mecânico dos dispositivos (ou terminais) e para indicar se a transmissão realizada está livre de erros.

Alguns sistemas on-line também são do tipo não interativo. Neste caso, o computador pode, meramente, receber uma transmissão de dados por lotes, não tendo a necessidade de responder a ela. Algumas vezes pode ocorrer resposta interativa, porém são respostas apenas para confirmação de recebimento correto da transmissão.

A maioria das transmissões para operadores humanos, em terminais, são do tipo interativas, pois, deve, sempre que possível, haver algum tipo de comunicação entre o operador e o computador, de modo que o primeiro seja informado se o seu input alcançou ou não o computador.

Para os sistemas não interativos ou sistemas que dêem uma resposta muito rudimentar, os dados flutuarão em uma única di

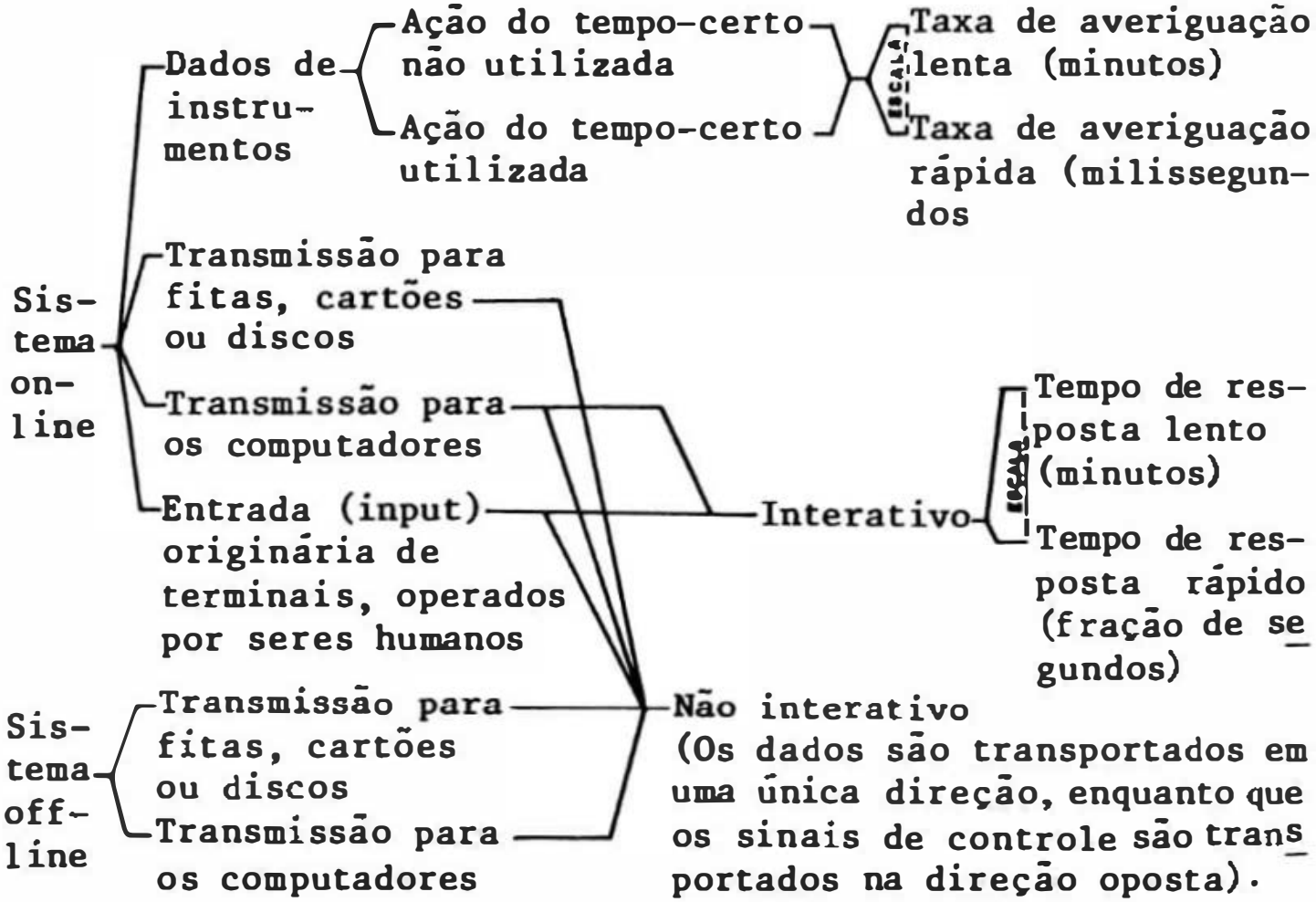
reção, muito embora tenhamos uma pequena "parcela" de sinais de controle indo em direção oposta. A figura 1 nos mostra as categorias mais comuns de sistemas de transmissão de dados.

QUANTIDADE DE DADOS TRANSMITIDOS

A quantidade de dados requeridos para serem transmitidos variam enormemente de um sistema para outro. Em um extremo, encontramos arquivos inteiros sendo transmitidos; no extremo oposto, temos apenas um bit, condição necessária para indicar somente SIM ou NÃO.

Quando ocorre um diálogo entre um usuário de um terminal e um computador, o tamanho das mensagens dependerá única e exclusivamente da estrutura projetada para o diálogo. O input para o computador é caracterizado por uma declaração feita pelo usuário, enquanto que o output é a resposta do computador. Tanto para o input, como para o output, a quantidade de dados requerida deve-se encontrar entre o máximo e o mínimo já citados.

FIGURA 01



TEMPO PARA TRANSMISSÃO

A velocidade requerida para o transporte de dados depende do sistema. Por exemplo, quando lotes de dados são enviados para o processamento, num computador distante, um tempo de entrega maior do que uma hora é, às vezes, aceitável. Entretanto, onde um diálogo homem-computador está ocorrendo, as respostas devem ser enviadas para o homem, de maneira suficientemente rápida, de modo a não impedir o seu raciocínio. Nestes casos, tempos de respostas entre 1 e 5 segundos são típicos.

Em sistemas do tipo tempo-certo (real-time), no qual uma máquina ou processo está sendo controlado, os tempos de respostas podem variar de uns poucos milésimos de segundos a vários minutos.

O tempo de resposta para um operador de terminal pode ser definido como aquele intervalo de tempo do aperto da última tecla do "input" (entrada) até à exibição do primeiro caracter da resposta.

A rapidez do tempo de resposta pode ter um importante efeito sobre o projeto e o custo das redes de transmissão de dados.

A figura 2 nos mostra um gráfico do tempo de resposta, em função da quantidade típica de dados utilizados nas transmissões de dados mais comuns.

O bloco rotulado "sistemas de terminal de diálogo" (veja figura 2), por exemplo, indica uma exigência de tempo de resposta de 1 a 10 segundos e um tamanho de mensagem, oscilando entre 1 (um) caracter (normalmente 7 ou 8 bits) até por volta de 4000 caracteres (± 30000 bits).

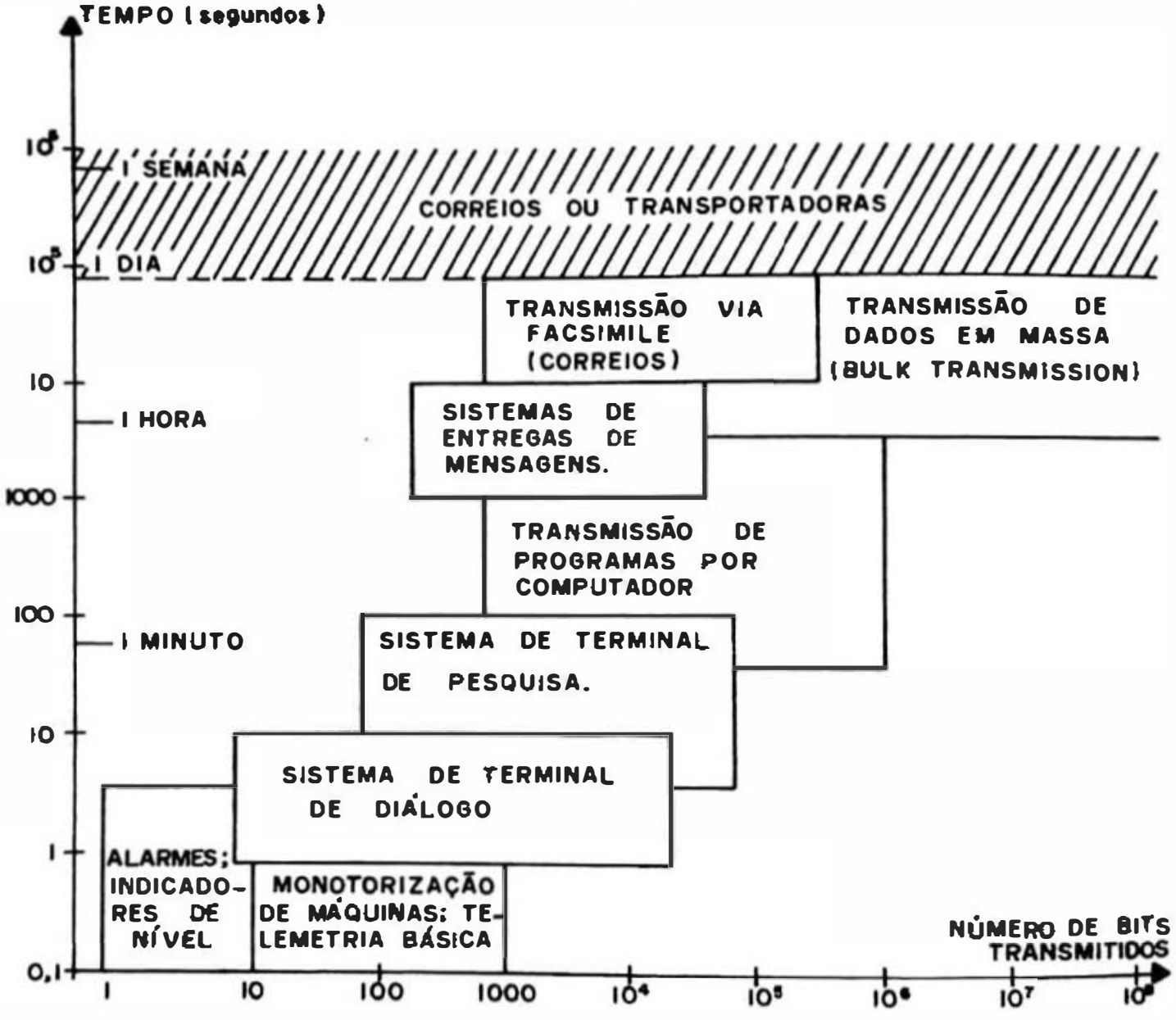
Em todos os casos apresentados na figura 2, o custo de prover diferentes tempos de resposta deve ser relacionado às exigências do usuário.

Nos casos em que a transmissão de dados, através de linhas, seja tão lenta que leve mais de 24 horas, é aconselhável utilizar-se dos serviços dos correios ou transportadoras aéreas, pois são usualmente mais baratos.

Possivelmente, no futuro, a transmissão de dados poderá utilizar áreas do lado de fora dos blocos que foram mostrados

na figura 2, a qual foi construída com base nas velocidades das linhas de transmissão disponíveis. A velocidade de transmissão é definida como sendo o número de bits transmitidos (eixo horizontal), dividido pelo tempo de entrega da mensagem, em um sentido (eixo vertical).

FIGURA 02



SISTEMA DE TEMPO COMPARTILHADO

Muitos sistemas com terminais de operação manual são de tempo compartilhado. Isto significa que mais de um usuário está utilizando o sistema ao mesmo tempo. Quando a máquina termina o processamento de itens de um dos usuários, imediatamen

te comuta sua "atenção" para o processamento de itens de um outro usuário.

Neste tipo de sistema, cada usuário sente-se como se fosse ele a única pessoa a utilizar-se do sistema. Atualmente estão em uso quatro categorias de sistemas on-line de tempo compartilhado, cuja diferença de um para o outro se encontra no grau de independência dos usuários. São eles:

1 - Sistemas que executam uma função cuidadosamente especificada e limitada. Por exemplo, sistemas bancários, nos quais todos os usuários de terminais podem atualizar os mesmos fichários (arquivos).

2 - Sistemas para uma função limitada específica. Neste caso, o usuário possui arquivos pessoais independentes ou então possui arquivos compartilhados, os quais podem ser lidos, mas não atualizados pelo usuário do terminal.

3 - Sistemas nos quais programadores podem programar qualquer coisa que eles desejarem nos terminais, sendo que todos utilizam a mesma linguagem.

4 - Sistemas nos quais programadores podem programar qualquer coisa, utilizando-se de uma variedade de linguagens de programação. Neste caso, normalmente, permite-se a utilização de uma variedade de terminais diferentes.

Nos dois últimos tipos de sistemas, divide-se o recurso de computação "time wise" e delega-se as partes a diferentes programadores para fazerem o que eles quiserem. Os arquivos (files) também são divididos de acordo com o uso.

LINHAS DE COMUNICAÇÃO DE COMPARTILHAMENTO DE TEMPO

A principal razão pela qual o compartilhamento de tempo é tão importante em computadores está no fato de que a taxa de "teclagem" (digitação) humana, bem como a taxa de leitura são consideravelmente mais lentas do que as respectivas taxas de um computador.

Como sabemos, o ser humano requer longas pausas para pensar sobre as transações. Então, a chave para se utilizar eficientemente o computador, em operações de diálogo do tipo real-time (tempo certo), é dividir o tempo do computador en-

tre muitos usuários. Fato semelhante ocorre com as linhas de telecomunicações e a maneira mais eficiente de utilizá-las seria através do compartilhamento do tempo entre diversos usuários. A necessidade de dividir as linhas de comunicação tornar-se-á maior no futuro, devido aos novos recursos que serão implantados, com o objetivo de transmitir os "bits" mais rapidamente.

O grau de independência dos usuários, em linhas de comunicação compartilhada, varia muito. Em algumas linhas compartilhadas, todos os usuários devem empregar o mesmo tipo de terminal, com o mesmo procedimento de controle-de-linha. Já, em outros casos, os terminais podem ser diferentes, porém devem usar o mesmo código de caracter. Existem casos de alguns sistemas, onde o código de caracter pode ser totalmente independente, ou seja, podem ser diferentes. Os usuários podem ser todos de um único tipo, por exemplo, os agentes de viagem ligados a um sistema de reservas de passagens de uma aerolinha. Por outro lado, os usuários podem ser de diferentes tipos, mas compartilham dos mesmos recursos de comunicação, instalados em uma organização. Finalmente, os usuários podem ser independentes, compartilhando os recursos de transmissão de dados públicos.

ESTRUTURA DAS REDES

Com o crescente número de terminais sendo utilizados em sistemas de computador, a necessidade de se desenvolver meios de abaixar o custo total das redes continua a crescer.

Muitos terminais transmitem e recebem dados a uma velocidade relativamente baixa. Em uma grande rede, os terminais podem ser conectados a linhas de baixa velocidade, as quais são derivadas de linhas de maior capacidade. Em alguns casos, estas linhas de velocidades maiores podem ser derivadas de uma outra linha de "band witch" ainda maior. A figura 3 nos mostra o diagrama esquemático de um trecho de uma rede. Podemos verificar, com base na figura 3, que muitos fluxos menores podem ser juntados em um ponto para formar um fluxo maior. Isto é conseguido através do uso de um multiplexador, o qual facilita o compartilhamento da linha.

Ainda com relação ao diagrama da figura 3, a linha de ve

locidade maior é, freqüentemente uma "linha de voz" alugada full-duplex, operando a 4800 bits por segundo. As linhas de velocidades mais baixas podem ser half-duplex, operando a 75 ou 150 bits por segundo. Eles podem enlaçar (link) ou englobar máquinas de teletipo ou terminais de máquina de escrever ao multiplexador. O multiplexador examina minuciosamente as linhas de baixa velocidade e retransmite os resultados de cada exame, através de uma linha de alta velocidade, a um computador "remoto". Normalmente, o multipexador envia um bloco de dados para a linha de alta velocidade. Este bloco é formado por um bit de cada linha de baixa velocidade ou também um caracter de cada uma das linhas de baixa velocidade. Ao mesmo tempo que envia blocos de dados, o multiplexador também recebe dados oriundos do computador e os distribui aos respectivos terminais, através de um processo de conversação. Neste modo de operação, uma máquina típica pode suportar 24 linhas de 150 bits por segundo.

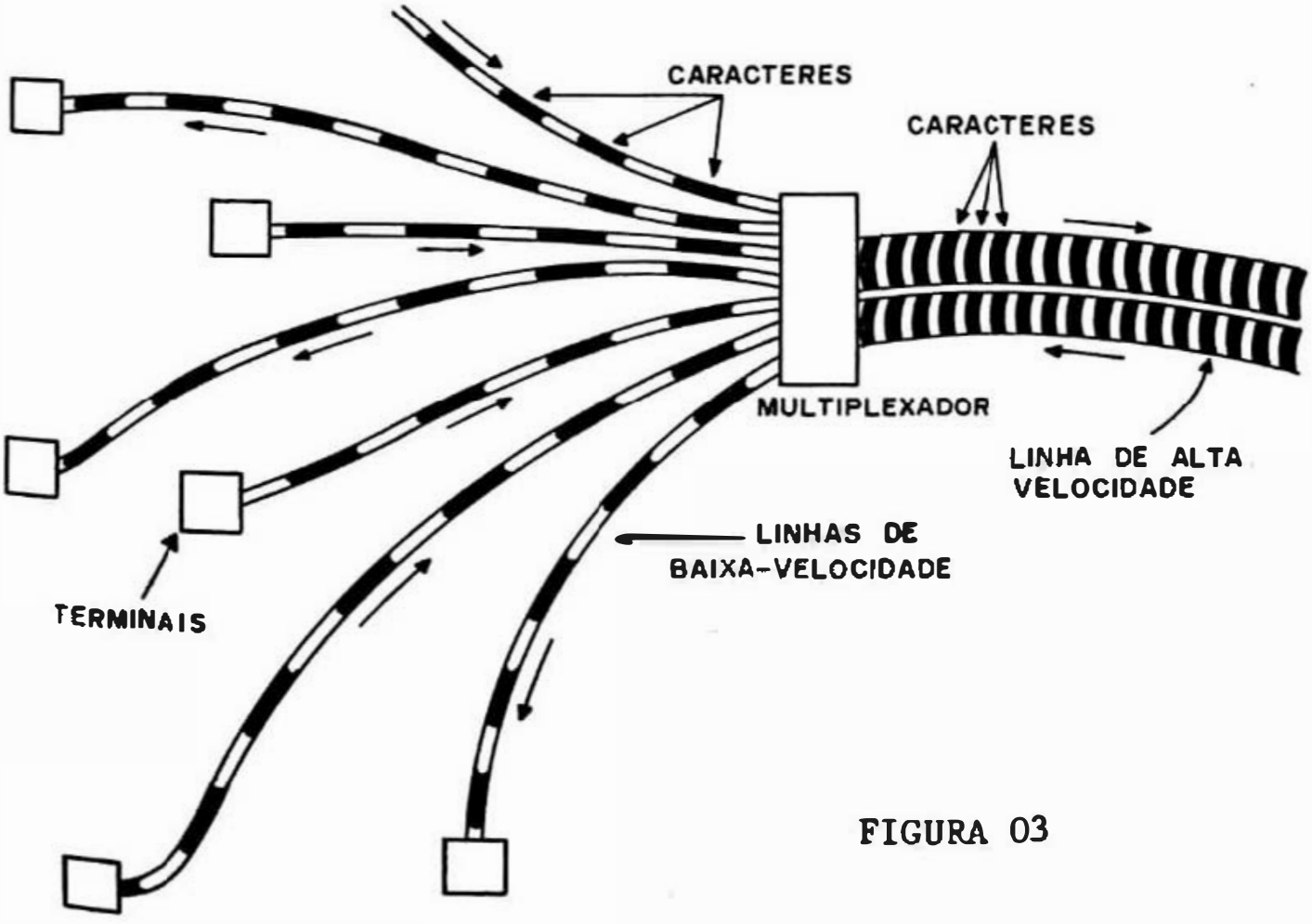


FIGURA 03

Os terminais da figura 3 podem ser manejados sem a necessidade de uma linha de alta velocidade, o que é possível, quando a velocidade de transmissão de caracter for relativamente baixa (± 20 bits por segundo). Neste caso, utiliza-se um circuito chamado "concentrador" (veja figura 4).

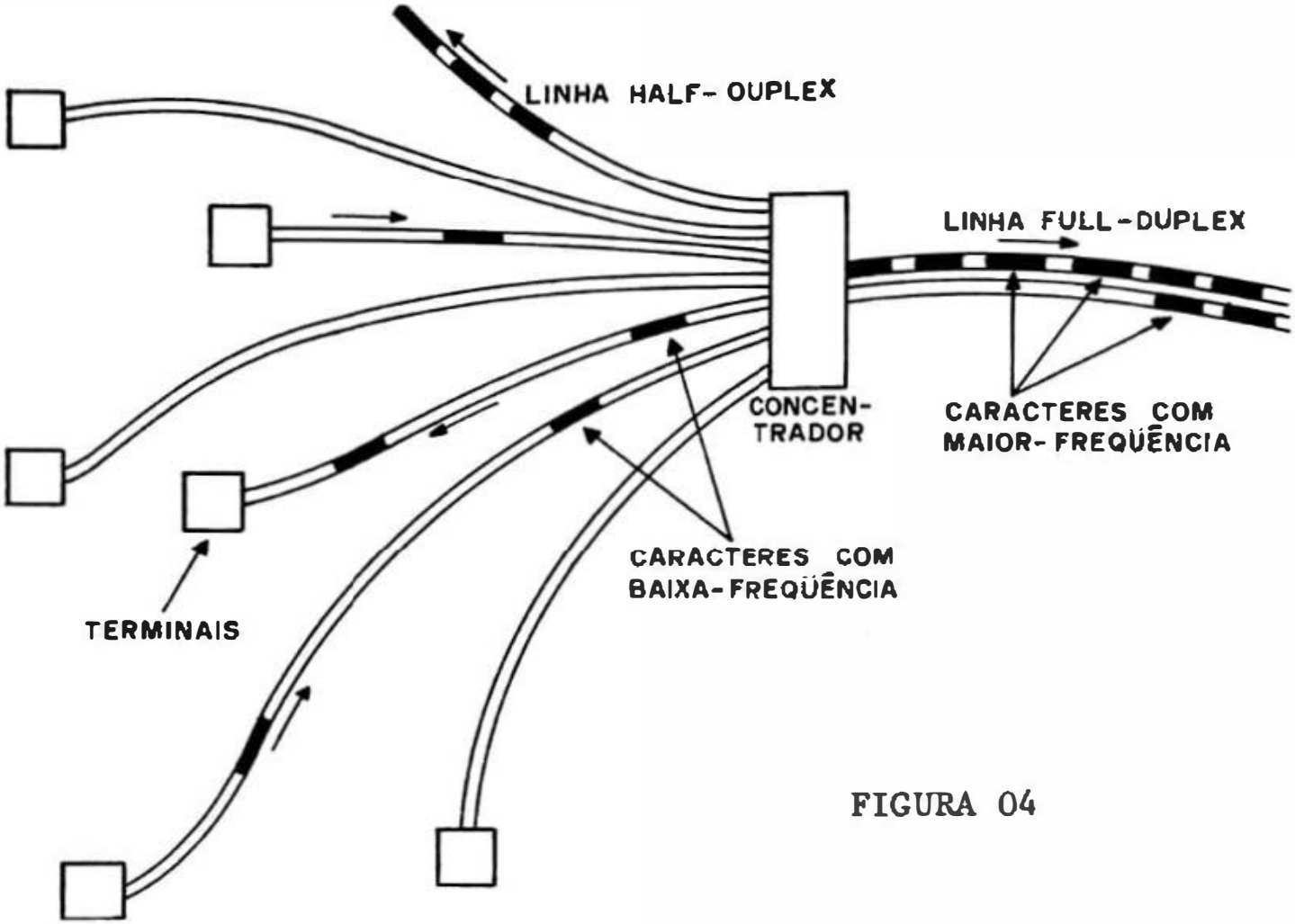


FIGURA 04

O concentrador da figura 4 coleta os caracteres que vão chegando dos terminais e os envia, progressivamente, sob a forma de blocos ou de mensagem. Cada um desses blocos ou mensagens deve indicar de qual terminal seus caracteres vieram. Da mesma forma, blocos ou mensagens que são enviados ao concentrador devem indicar o terminal de destino. As linhas entre o concentrador e os terminais são do tipo half-duplex e entre o concentrador e o computador temos uma linha do tipo full-duplex.

CATEGORIAS DE LINHAS DE COMUNICAÇÃO E TARIFAS

As categorias de linhas de comunicação e tarifas são mostradas, de forma resumida, na tabela a seguir.

TIPOS DE LINK	COMENTÁRIOS
Link digital	Projetados para transmissão digital. Nenhum modem requerido. São sensíveis a códigos, em alguns casos.
Link análogo	Transmitem uma média contínua de freqüência como uma linha de voz. Modem é requerido.
Público comutado	Mais barato se o uso é baixo. Linhas telefônicas comutadas são universalmente disponíveis.
Alugado (às vezes, chamado de "privado")	Mais barato do que linhas públicas, seu uso é alto. Pode ter menor taxa de erros. Velocidades maiores possíveis em linhas telefônicas usadas do que em linhas comutadas.
Alugado com comutação privada	Pode dar o mais baixo custo. Combina as vantagens de linhas alugadas com a flexibilidade de comutação. Linhas públicas de wide band comutada podem não estar disponíveis.
Privado (non-common carrier)	Usualmente, permitido somente dentro de certas premissas por parte do assinante. (Veja o próximo item).
Privado (non-common carrier) links	
Inplant	Velocidades de bits muito altas. Realizáveis.
Rádio Microonda	Formidável em casos especiais para links ponto-a-ponto.
Shortwave ou Rádio VHF	Usado para transmissão, para veículos em movimento ou pessoas.
Óptico ou Infravermelho	Usados para links curtos -e.g, intercidade - a velocidades altas de bits (250.000 bps, típico). Nenhuma licença requerida. Tirado fora de ação por neblina ou chuva muito intensa.

Velocidades Subvoice grade Voice grade Wide band	Usualmente, refere-se a velocidades abaixo de 600 bits por segundo. Usualmente as linhas de voz análoga usam modems de velocidades entre 600 e 10.500 bits por segundo. Velocidades acima daquelas de linhas de voz, mais comumentemente 19.200, 40.800, 50.000 e 240.000 bps.
Modo de operação Simplex Half-duplex Full-duplex	Não é normalmente usado em transmissão de dados. O mais comum nos E.U.A. No Bell System custa 10% mais do que half-duplex e pode dar produção desproporcionalmente maior, se o terminal é projetado para levar vantagem dele.

NOTA: Estes termos, às vezes, descrevem a limitação de uma máquina mais do que a limitação da linha em que eles estão fixados.

Condicionamento Tipos → C_1, C_2, C_4	Aplicado a linhas de voz análogas, alugadas, realiza mais atenuações uniformes e atraso de maneira que modems de velocidade maior possam ser empregadas.
Tarifas Tarifa inclui jogo de dados Tarifa simples para bandwidth	Empresas de comunicação provêm o jogo de dados (modem) e melhora a velocidade de transmissão. Usuário usa o modem. Velocidade de transmissão depende do projeto do modem.
Data-phone Service	Um serviço AT & T que fornece um jogo de dados com a linha.

Redes públicas de telégrafo Telex TWX TWX → TTY-TWX TWX → CPT-TWX TWX → CE-TWX	Rede mundial de telégrafo, código Baudot, 6.7 caracteres por segundo. Network de telégrafo norte-americano. Código Baudot a 6 caracteres por segundo ou código DIC a 10 caracteres por segundo. Com máquina teletipo de empresa de comunicação. Com terminais providos de cliente, aderindo a TTY-TWX, códigos e controle de linha. Com equipamento do cliente. Não precisa aderir aos códigos TTY-TWX. Velocidade até 150 bits por segundo. Não listado em diretórios TWX.
WATS (Wide Area Telephone Service) WATS → INWATS WATS → OUTWATS	Qualquer assinante na zona específica pode chamar a linha WATS. O criador da chamada é conectado à linha WATS e pode chamar qualquer assinante na zona específica. INWATS e OUTWATS devem estar em linhas separadas.
TELPAK Bundling → TELPAK C, Tipo 5700 Bundling → TELPAK D, Tipo 5800	Uma forma de desconto de quantidade na qual um bundle de linhas pode ser comprada. Equivalente a 60 canais de voz. Equivalente a 240 canais de voz. O bandwidth TELPAK pode ser subdividido em muitos modos diferentes.

Visto estas pequenas particularidades sobre o assunto telecomunicações, passamos agora ao estudo dos Modems.

MODEMS

Trata-se de um dispositivo eletrônico, utilizado para modular e demodular os sinais transmitidos por equipamentos e redes de transmissão de comunicação, convertendo os dados de

uma forma compatível com o equipamento de processamento de dados para outra forma compatível com o equipamento de transmissão e vice-versa. A figura 5 ilustra uma aplicação típica de um modem.

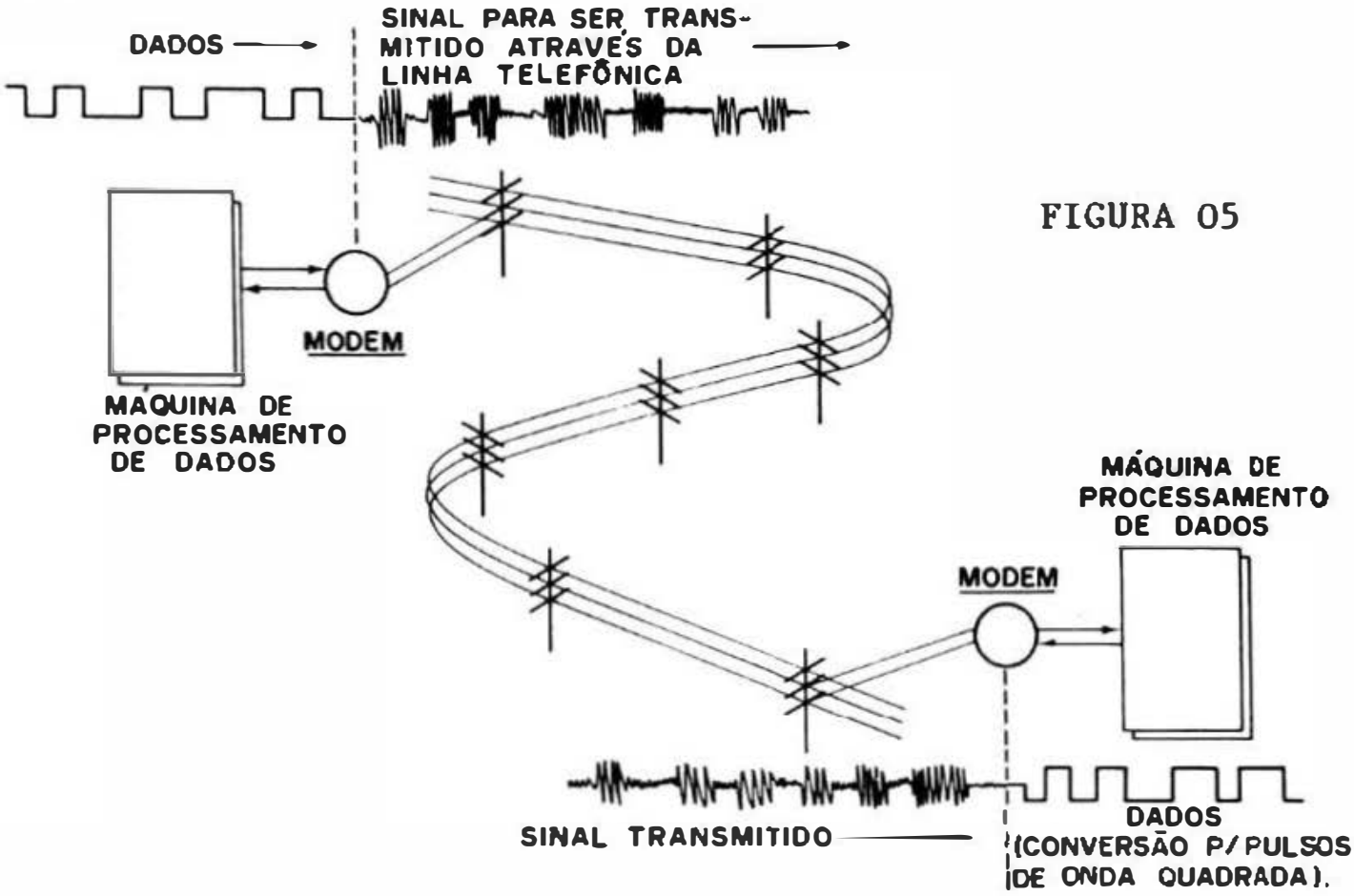


FIGURA 05

A figura 6, por sua vez, nos mostra um modem típico, utilizado comercialmente.

FIGURA 06

Além de converter o sinal para uma forma na qual uma taxa alta de bits possa ser transmitida, o modem serve, também, para proteger as linhas de sinais indesejáveis que podem causar interferência em outros usuários ou no sistema de sinalização da rede.

O desenvolvimento e aperfeiçoamento dos modens são responsáveis pela crescente velocidade, na qual os dados podem ser enviados, através de linhas telefônicas. Existe uma variedade de modens, projetados para diferentes sistemas, cada qual com características peculiares.

Terminais operando até 300 bits por segundo podem operar com sucesso, sem a utilização de modens para a transmissão, através de linhas sem amplificadores e a uma distância de aproximadamente 5 Km. Este tipo de transmissão, sem a utilização de modens, onde os impulsos (sinais de onda quadrada) emergem do terminal e seguem diretamente para dentro do par de fios que são utilizados para a transmissão recebe o nome de "baseband".

Quando bits são enviados muito longe ou muito rápido para a sinalização tipo "baseband", eles devem ser convertidos para freqüências apropriadas para a linha de transmissão em questão. Um método barato e que pode ser classificado como intermediário entre o "baseband" e o modem é o acoplamento acústico. Para tanto, são utilizados os acopladores acústicos, os quais convertem os caracteres para serem transmitidos para tons audíveis, tais como aqueles gerados por um telefone. Uma máquina que utiliza este processo não necessita, fisicamente, estar conectada à linha através de fios, pois ela pode gerar sons que são captados diretamente pelo "hanset" (fone) do telefone. Daí o nome de acoplamento acústico. Os sons utilizados (gerados) devem ter freqüências dentro do bandwidth do telefone. Os sons são reproduzidos na extremidade remota da conecção pelo "earpiece" (parte do telefone que vai ao ouvido) do telefone, e são, então, convertidos para a faixa de sinais de dados. Este tipo de transmissão é um pouco lento e é muito utilizado em terminais de máquinas de escrever. A figura 7 nos mostra um dispositivo de acoplagem acústica, no qual o braço do telefone é encaixado dentro de uma caixa especial.

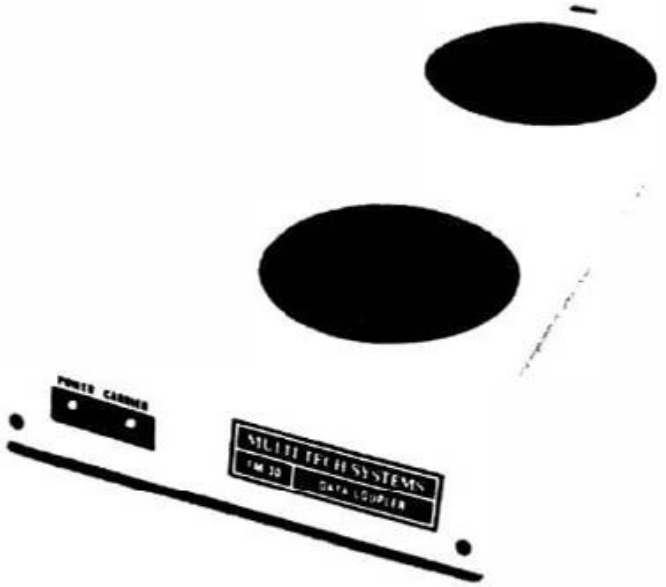


FIGURA 07

Geralmente, os acopladores acústicos apresentam um custo mais baixo que o modens. Outra vantagem do acoplamento acústico é que o terminal pode ser facilmente transformado em terminal portátil.

O acoplador acústico também é utilizado em conjunto com o modem. Neste caso, ele passa a receber o nome de acoplador de dados. Este aparelho permite, então, a ligação de fios entre os modens e as linhas telefônicas públicas (veja figura 8).

FIGURA 08

A isolação elétrica, para estes casos, é obtida através de transformadores de isolação, sistemas de filtragem e técnicas de aterramento.

MODENS DE ACOPLAMENTO PARA PERIFÉRICOS

Os modens são tipicamente acoplados aos periféricos, de acordo com o padrão RS 232 C. Alguns modens fornecem também um loop (laço) de corrente de 20 mA ou 60 mA, o que permite o seu acoplamento em máquinas de "teletipo".

CLASSIFICAÇÃO

Os modens podem ser classificados de acordo com a capacidade de operação, conforme nos mostra a tabela da figura 9.

A capacidade de transferência de dados do dispositivo periférico limita a velocidade de transmissão de dados. Os modens com múltiplas velocidades de transmissão de dados possuem um método para mudar a velocidade de transferência de dados, de modo a se equiparar com a velocidade do periférico.

TIPO DE MODEM	VELOCIDADE EM BITS/S
Baixa velocidade	110; 300 e 600
Média velocidade	1200 e 2400
Alta velocidade	4800 e 9600
Larga Faixa	9600 e acima

FIGURA 09

Quanto aos métodos de modulação, os modens podem ser classificados em três categorias, conforme a figura 10.

MÉTODOS DE MODULAÇÃO

1 - Modulação de Amplitude (AM);

2 - Modulação de Freqüência (FM);

3 - Modulação de Fase (PSK).

FIGURA 10

INTERFACE ENTRE MODENS E MÁQUINAS

É muito importante que a interface entre o modem e as várias máquinas de transmissão de dados seja padronizada, pois possibilita que um determinado modem possa ser substituído por outro, ou ainda, possibilita que terminais e unidades de controle de transmissão possam ser intercambiadas. A interface EIA RS-232-C standard, fabricada pela "Eletronics Industries Association", é um exemplo de interface padrão. A figura 11 nos mostra as interconexões padrões entre o modem e o terminal.

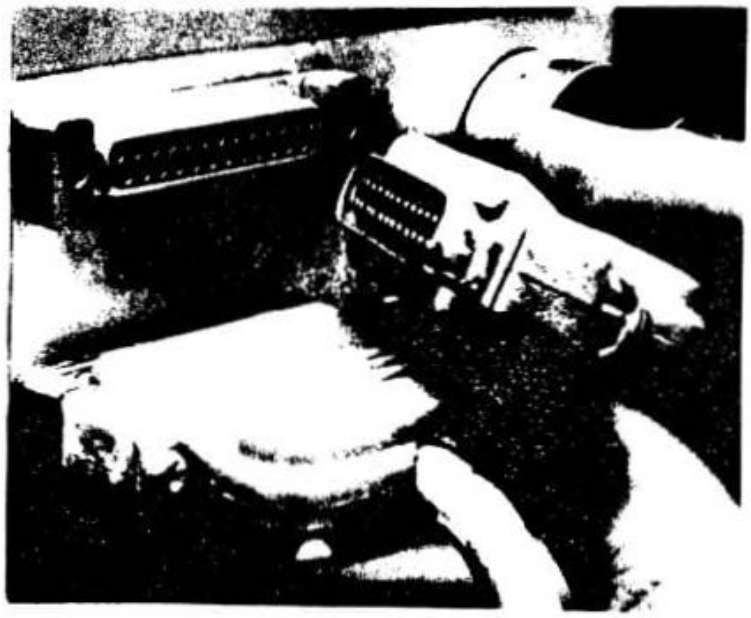


FIGURA 11

- Este é um plug de 25 pinos, entre o modem e o equipamento de processamento de dados.

A figura 12 apresenta um diagrama de blocos, no qual podemos verificar a utilização de interfaces do tipo RS-232-C standard.

Note que, no caso apresentado na figura 12, podemos verificar que não há ligação elétrica direta entre o modem e a linha telefônica, porém existem, somente, as freqüências da voz produzida e recebida acusticamente.

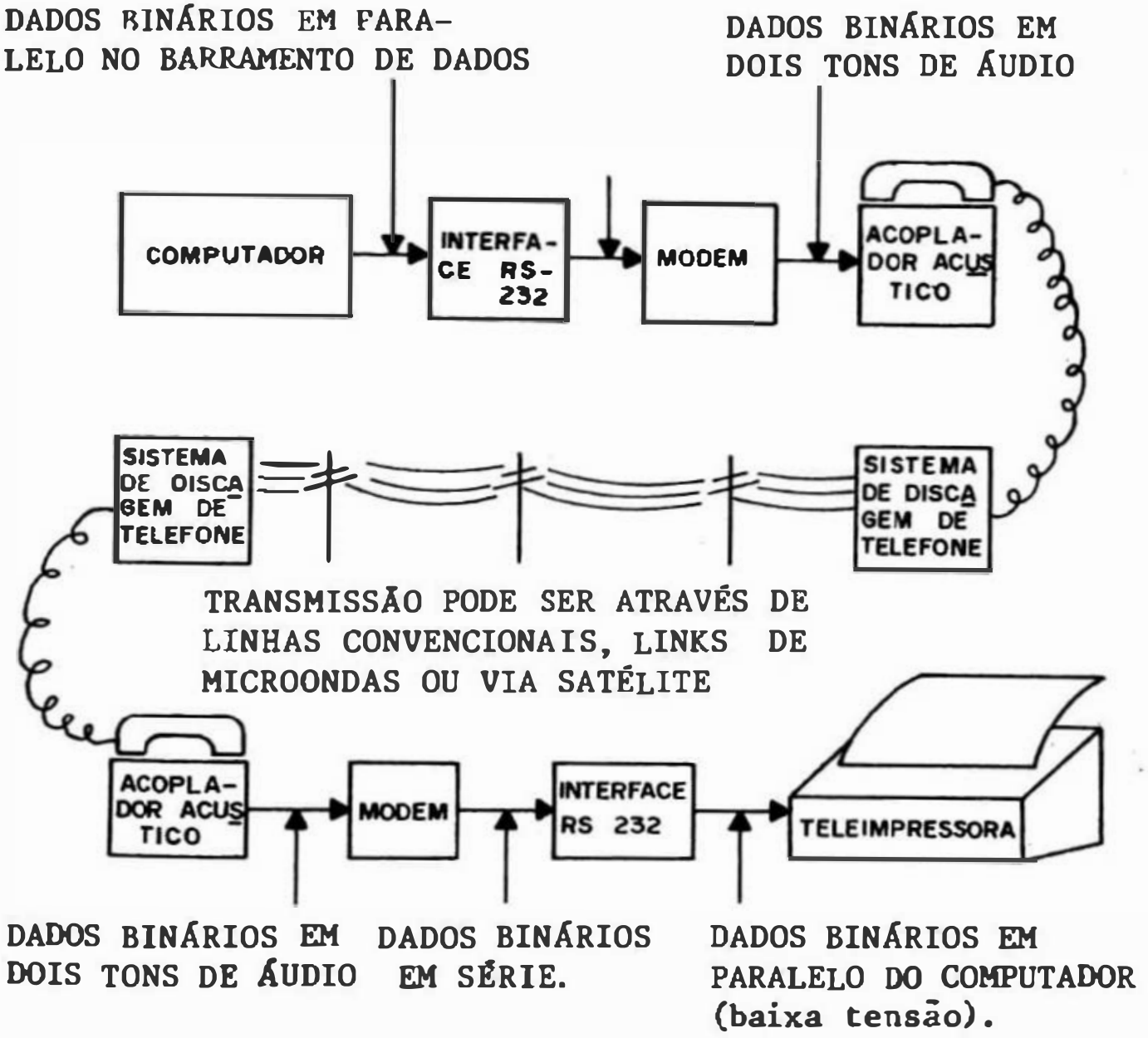


FIGURA 12

RESPOSTA AUTOMÁTICA E SISTEMAS DE DISCAGEM AUTOMÁTICA

Um dispositivo de resposta automática pode ser adicionado ao sistema, no terminal principal ou terminal do computador. Neste caso, o modem que está acoplado diretamente ao terminal do computador, possui um circuito adicional, capaz de identificar os sinais de "chamada" da linha telefônica, ligando o modem de resposta à linha de transmissão e ao modem de origem. Este método é muito utilizado quando muitos terminais remotos acessam um computador central.

Os dispositivos correspondentes à discagem automática, com equipamento eletrônico adicional apropriado, podem ser controlados por uma UCP, sob o controle de um programa.

Nas instalações mais sofisticadas, hoje em dia, é comum a instalação de dispositivos de respostas automáticas e de discagem automática entre dois computadores.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-16

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Num sistema on-line, os dados transmitidos vão diretamente para:

a) dentro do computador;

b) as fitas de papel;

c) os disketes nos quais serão armazenados;

d) N.R.A.

2) Num sistema off-line, os dados transmitidos:

a) vão diretamente para dentro do computador;

b) não vão diretamente para dentro do computador;

c) vão diretamente para onde eles serão utilizados;

d) N.R.A.

3) Normalmente os sistemas off-line são:

a) interativos;
b) não interativos;
c) indiferentes;
d) N.R.A.

4) Nos sistemas não interativos:

a) os dados flutuarão em ambas as direções;
b) os dados são transmitidos por lotes;
c) os dados flutuarão em uma única direção;
d) N.R.A.

5) Nos sistemas de tempo compartilhado:

a) temos apenas um usuário, utilizando o sistema;
b) temos mais de um usuário, utilizando o sistema;
c) a transmissão de dados é feita através de transporte aéreo e rodoviário;
d) N.R.A.

6) O compartilhamento de tempo em linhas de telecomunicação tornou-se viável, devido:

a) ao seu baixo custo de operação;
b) à baixa taxa de digitação humana;
c) à baixa taxa de leitura do digitador;
d) os itens b e c estão corretos.

7) Quanto ao modo de operação, as linhas de transmissão podem ser:

a) simplex;
b) half-duplex;
c) full-duplex;
d) todos os itens estão corretos.

8) Os modens são utilizados para:

a) modular os sinais a serem transmitidos;
b) demodular os sinais recebidos;
c) os itens a e b estão corretos;
d) N.R.A.

9) A transmissão de dados, sem a utilização de modens, recebe o nome de:

a) baseband;
b) bandwidth;
c) wideband;
d) N.R.A.

10) Um modem do tipo média velocidade possui uma velocidade de transmissão entre:

a) 300 e 600 bits/seg.;
b) 1200 e 2400 bits/seg.;
c) 4800 e 9600 bits/seg.;
d) acima de 9600 bits/seg.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-17

APLICAÇÕES DE MICROPROCESSADORES I

INTRODUÇÃO

Muito embora os microprocessadores não estejam tão difundidos entre nós, como nos países mais desenvolvidos, podemos dizer, com toda certeza, que este "componente" conquistou definitivamente seu lugar na eletrônica. Partindo de um começo sem muito entusiasmo, por volta de 1971, o microprocessador é tido, hoje em dia, como uma resposta mágica para a indústria, a qual procura utilizá-lo, onde quer que haja necessidade de peças eletrônicas discretas e até mesmo mecanismos inteiros. Atualmente os microprocessadores são aplicados em máquinas tão variadas como: computadores, fornos de microondas, relógios, flipperamas, calculadoras, brinquedos, jogos eletrônicos, veículos, aparelhos de TV, alarmes, equipamentos industriais, etc.

Iniciaremos nesta lição um breve estudo sobre as aplicações dos microprocessadores. Abordaremos, inicialmente, alguns aspectos mais genéricos e, numa segunda parte, apresentaremos algumas aplicações práticas, a título de exemplo.

GENERALIDADES

Atualmente, os sistemas com microprocessadores podem

ser aplicados na maioria dos produtos acabados, desde que estes tenham um certo grau de complexidade e um preço que viabilize tal procedimento. Por exemplo, se um determinado produto "x" possui um mecanismo muito simples e é comercializado por um preço relativamente baixo, a utilização de microprocessadores pode ser desnecessária, ou desaconselhável, tanto sob o ponto de vista técnico, quanto sob o ponto de vista econômico.

Atualmente, o preço de alguns chips utilizados nos sistemas com microprocessadores são comercializados a um custo relativamente baixo, sendo que a tendência do mercado é trabalhar com preços mais baixos ainda. Isto nos leva a crer que, em breve, a parcela do custo da implantação de um sistema com microprocessador, em um produto acabado, será, relativamente, de pouca importância. Tal fato se deve, em grande parte, à constante pesquisa desenvolvida na área da microeletrônica, trazendo sempre inovações, de modo que os fabricantes passam a oferecer componentes cada vez mais sofisticados e potentes a um preço relativamente menor. Com base nesses fatos, podemos tirar duas conclusões importantes:

1ª) O emprego em massa dos sistemas com microprocessador na indústria, nos negócios, na pesquisa, na escola e no lar. Isto nos leva a supor que deverá ocorrer uma aproximação progressiva do homem com os sistemas de computadores, desde a infância, o que motivará o ensino das linguagens de programação e o uso dos computadores, desde os primeiros anos de escola. Por outro lado, será necessário a criação de escolas de nível superior, especializadas tanto em software como em hardware, formando engenheiros e especialistas em arquitetura, projeto e programação de computadores.

2ª) Os avanços obtidos no sentido de se aumentar a intensidade de integração e a redução dos custos dos componentes tornarão possível a aplicação dos sistemas com microprocessadores em todos os campos. Isto nos levará a um grande desenvolvimento da pesquisa e da fabricação dos equipamentos de processamento de dados, exigindo, conseqüentemente, a modernização e a capacitação dos atuais fabricantes para que possam avaliar e aplicar toda esta tecnologia de ponta. Os produtos, os processos de fabricação, a administração, os escritórios de projetos, os laboratórios, os técnicos, os engenheiros, as empresas, de um modo geral, e os países que não se prepararem

devidamente para assimilar a tecnologia da microeletrônica correrão um sério risco de "degradação", ante a impossibilidade de competir, sob qualquer aspecto, com quem domina esta tecnologia.

Esta segunda conclusão serve mais como alerta para que seja estabelecida uma planificação a nível de Estado, a nível empresarial e a nível de consumidor (pessoal). Para tanto, deve-se partir para uma preparação cultural e técnica em todos os níveis, iniciada já nos primeiros anos de escola, com a utilização de microcomputadores na aprendizagem de algumas matérias, com a utilização de calculadoras, bem como a iniciação ao aprendizado de alguma linguagem de programação, como o "Basic", por exemplo. Um outro aspecto desta planificação seria a conscientização, a nível de Estado (Governo), para impulsionar o conhecimento, o desenvolvimento e a aplicação industrial dos sistemas de processamento de dados.

O IMPACTO DA MICROELETRÔNICA

A aplicação dos sistemas de tratamento de dados com base em componentes que procedem da microeletrônica está baseada em dois grandes focos:

1º) As grandes indústrias e as grandes corporações, as quais destinam parte de seus lucros à pesquisa, desenvolvimento e aplicação da tecnologia. Também merece destaque o esforço e o nível adquirido pelos fabricantes de eletrodomésticos, de máquinas-ferramenteiras, eletromedicina, instrumentação, controle remoto, terminais inteligentes, etc...

2º) A nível pessoal, representados pelos aficcionados, apaixonados, em muitos casos, por sistemas microinformáticos, que, a partir de um módulo básico, assimilam a tecnologia, na maioria das vezes, apenas por satisfação pessoal, sem uma orientação ou um motivo de aplicação real e útil.

Entre esses dois focos existe um vazio muito grande com relação ao emprego dos microcomputadores. Este vazio se deve ao pouco uso destes aparelhos por parte das pequenas e médias empresas as quais, na maioria dos casos, desconhecem esta tecnologia e as grandes possibilidades que estes sistemas podem

proporcionar-lhes.

Uma das áreas de maior aplicação dos sistemas com microprocessadores é a substituição dos circuitos eletrônicos de chaveamento, em geral, construídos à base de elementos de tecnologia TTL. A partir de um certo grau de complexidade, de um modo geral, quando estivermos trabalhando com cifras da ordem de 30 ou 40 chips SSI e/ou MSI, torna-se interessante implementar o circuito à base de microprocessadores, cujo programa de instruções ou de trabalho está gravado em uma memória não volátil. Eis algumas vantagens desta substituição:

a) Menor tempo requerido para o projeto e testes do protótipo.

b) Possibilita o emprego de placa standard que se adapta à maior parte do sistema, reduzindo-se, desta forma, os custos de montagem e verificação.

c) Para quantidades relativamente grandes, fazendo-se o pedido diretamente para o fabricante, pode-se estudar a possibilidade de se integrar a maior parte do sistema em um só chip. Esta alternativa reduz em muito o custo de montagem, de teste e de material, melhorando os aspectos relacionados com a redução do volume, aumento de fiabilidade e maior competividade. Nestes casos, há de se levar em conta se os fabricantes podem produzir ou não esses tipos de chips de baixa demanda (ou chips específicos).

d) Possibilita, no futuro, a troca do programa de trabalho do sistema (basta trocar o programa da memória), assim como ampliá-lo, se necessário.

e) Redução dos custos de mão-de-obra e verificação.

f) Importante aumento da fiabilidade, devido à redução de componentes nos sistemas com microprocessador.

g) Redução do espaço ocupado e do gasto de energia consumida.

h) Facilita a introdução de programas de autocorreção, que facilitam a reparação e manutenção do equipamento.

i) Capacitação do pessoal e, em geral, da empresa, possibilitando o desenvolvimento futuro de produtos novos e mais avançados.

j) Melhoria da imagem do produto e da empresa que o fabrica.

l) A utilização de sistemas com microprocessadores, representa um avanço, provocando um distanciamento dos possíveis competidores.

APLICAÇÕES DOS SISTEMAS COM MICROPROCESSADORES

Para podermos mostrar, de uma forma mais prática, as aplicações dos sistemas com microprocessadores, estabelecemos uma classificação por setores. Trata-se de uma classificação geral de caráter básico que delimita os campos industriais de maior interesse e desenvolvimento dos sistemas baseados em microprocessadores. A classificação é a seguinte:

1) Indústria em geral;
2) Indústria automobilística;
3) Controle do tráfico de elevadores;
4) Indústria elétrica e de eletrodomésticos;
5) Eletromedicina;
6) Instrumentação e medida;
7) Terminais inteligentes;
8) Jogos e derivados.

Esta classificação pode ser reorganizada em (5) campos de aplicações gerais, conforme nos mostra o quadro da figura 1.

FIGURA 01

Vejamos, então, de forma bem simples e objetiva, alguns comentários sobre as possíveis aplicações em cada um desses

cinco campos.

1 - APLICAÇÕES NO CAMPO DA INDÚSTRIA EM GERAL

Neste campo, podemos encontrar os sistemas com microprocessadores aplicados nas condições mais variadas, como, por exemplo, na laminação, na aquisição de dados, nos transdutores, no controle numérico de máquinas ferramenteiras, no controle parcial e total de processos, na dosificação de portas lógicas, nos automatas programáveis, na substituição da eletrônica convencional, na classificação de chapas, na pesagem eletrônica, na comprovação dos elementos ativos e passivos, etc...

A maioria das aplicações industriais caracteriza-se pelos seguintes detalhes técnicos:

a) Grande parte das aplicações requer entradas e saídas analógicas. Nestes casos, o sistema processador resultante é equivalente a um controlador analógico com um certo número de "bucles" de controle. Um "bucle" de controle é simplesmente a realização de um algoritmo que regulará uma saída em função de uma ou mais entradas.

b) Muitas das aplicações industriais de sistemas com microprocessadores caracterizam-se pelo emprego de sensores e mecanismos de controle de altos custos. O custo dos sensores de entrada e dos dispositivos de saída é tal que o preço de um sistema com microprocessador típico pode ser considerado desprezível quando comparados. O baixo custo do microprocessador em si não é uma vantagem significativa, tendo em vista o alto custo do sistema completo. Nestes casos, a principal vantagem de se utilizar um sistema com microprocessadores está na disponibilidade de um elemento de lógica programada emsubstituição aos de lógica de chaveamento fixa. Todo o mérito do microprocessador, neste caso, provém do software.

A programação permite realizar funções de qualquer grau de complexibilidade, as quais não poderiam, anteriormente, serem resolvidas mediante a lógica de chaveamento. Os microprocessadores encontram aplicação no controle de processos, de fluxos contínuos ou não e na regulação de praticamente qualquer processo de controle.

Vejamos, agora, alguns exemplos de aplicações no campo industrial.

a) SISTEMAS COM MICROPROCESSADORES APLICADOS AOS TRANSDUTORES

A maioria dos transdutores, ou seja, detectores de medidores de magnetudes (pressão, posição, velocidade, temperatura, força, umidade, corrente elétrica, etc.) são do tipo analógico e fornecem um nível elétrico proporcional à amplitude da magnetude que medem. A aplicação destes sinais analógicos a sistemas digitais, tais como os que rodeiam os microprocessadores, requer a conversão analógica/digital (A/D), que geralmente é feita antes de realizar-se a interface com o sistema.

Uma notável aplicação dos sistemas com microprocessadores consiste em conseguir uma conversão precisa dos sinais analógicos em digitais, que de forma simplificada está representada pelo diagrama de blocos da figura 2.

De um modo geral, os transdutores não proporcionam uma resposta "linear". Esta deficiência é superada por meio dos sistemas com microprocessadores.

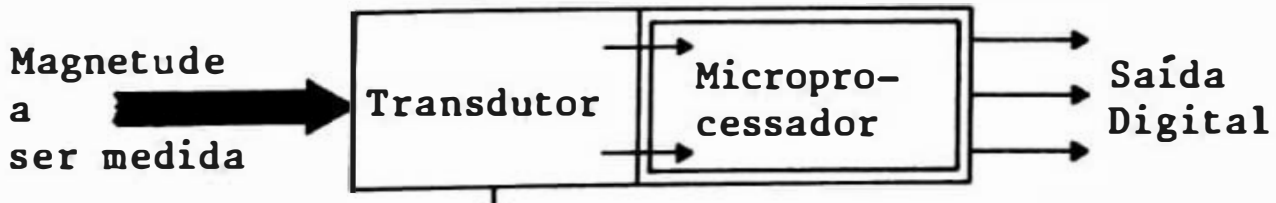


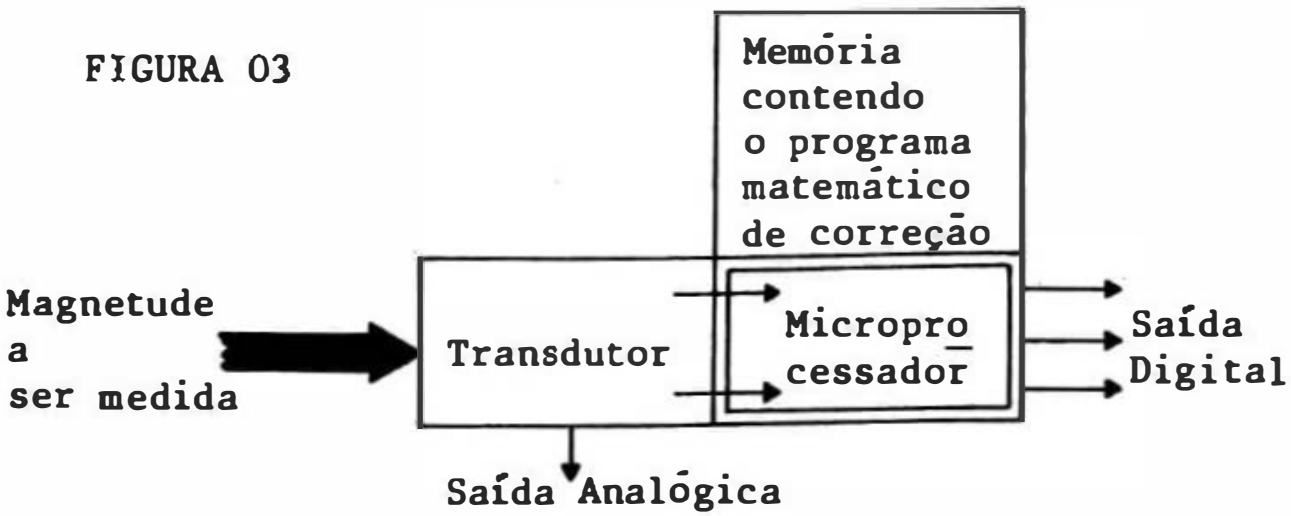
FIGURA 02

O sistema com microprocessador irá tentar reduzir ao mínimo o erro do transdutor, utilizando-se, para tanto, de dois caminhos:

- Armazenar na memória um programa matemático que corrija os desvios da saída do transdutor.
- Processar tempos muito curtos.

A figura 3 nos mostra a adição da memória, a qual contém o programa matemático de correção, para o diagrama de blocos apresentado na figura anterior.

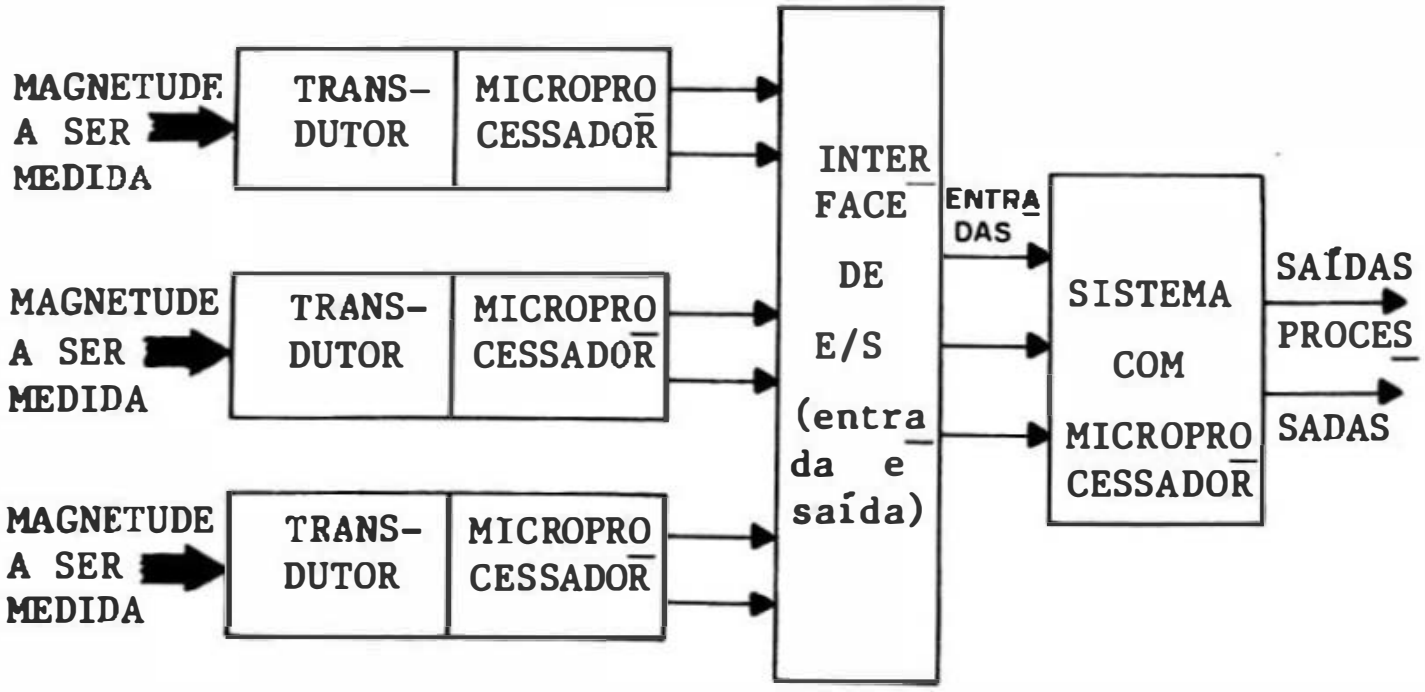
FIGURA 03



O sistema microprocessador realiza filtragens, operações matemáticas, série de Fourier, aproximações, etc., que garantem ao máximo as saídas digitais com relação à magnetude que está sendo medida, eliminando-se as imperfeições do transdutor.

A figura 4 nos mostra o diagrama de blocos de um sistema de microprocessadores integrados no mesmo transdutor, os quais alimentam, mediante um módulo de E/S, um outro sistema com microprocessador que irá processar os dados digitais.

FIGURA 04



Um exemplo real de aplicação dos transdutores descritos é a medição de potência, na qual, se mede, em curtos intervalos de tempo, a tensão em função do tempo "V(t)" e a corrente em função do tempo "i(t)", chegando-se ao valor da potência mediante o desenvolvimento de uma "série de Fourier". O sistema com microprocessador que recebe os valores instantâneos de i(t) e V(t), oriundos dos transdutores integrados, dispensa a esses sinais um tratamento matemático para proporcionar a potência procurada.

b) SISTEMAS COM MICROPROCESSADORES APLICADOS NA INDÚSTRIA AUTOMOBILÍSTICA

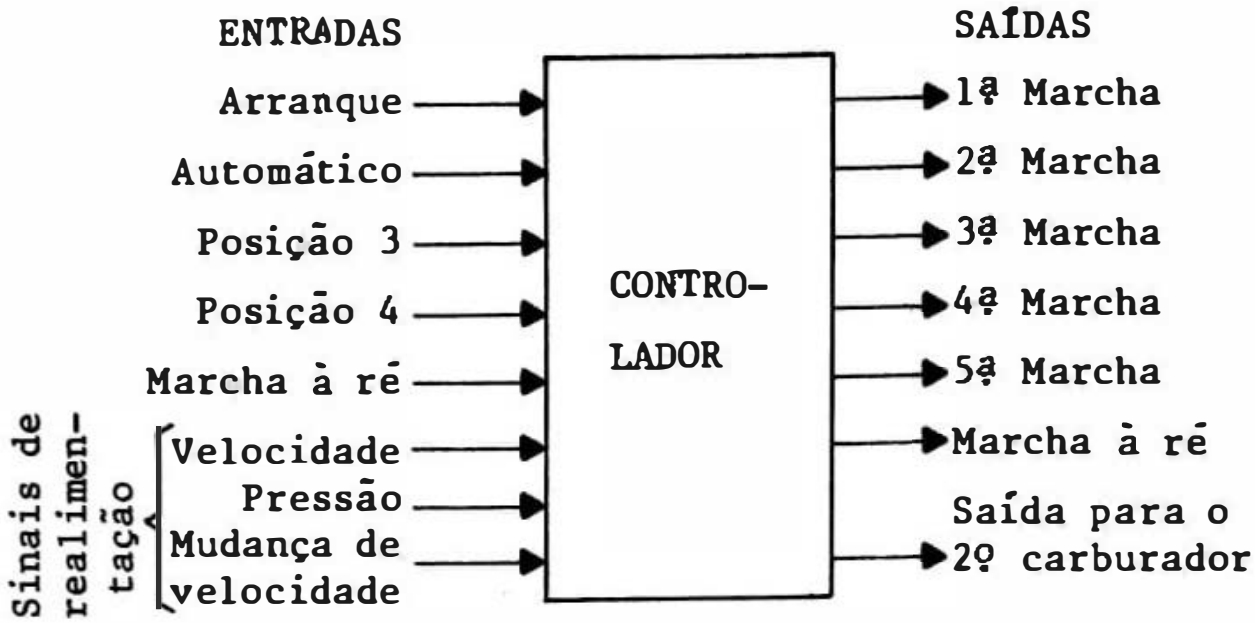
As aplicações mais importantes na indústria automobilística estão relacionadas a seguir:

- Regulação da injeção de combustível e do tempo de ignição para favorecer uma redução no consumo;
- Controle da saída dos gases procedentes da combustão;
- Instrumentação e painel de controle;
- Distribuição do peso para regular a carga sobre os eixos;
- Sistema de alarme para detecção de avarias ou falhas previsíveis;
- Multiplexação dos cabos elétricos do veículo;
- Operações de manutenção, regulação e localização de avarias;
- Controle de tráfego;
- "Mudança automática de velocidades".

A mudança automática de velocidades se baseia em um circuito composto de relés e circuitos integrados SSI, constituindo o "controlador", o qual ativa um dispositivo eletromecânico que irá acionar diretamente a caixa de câmbio.

Com o uso de microprocessadores, reduzem-se os custos destes controladores, além de aumentar a fiabilidade e diminuir os componentes e o consumo. A figura 5 nos mostra o diagrama de blocos geral de um controlador.

FIGURA 05



Mediante um interruptor múltiplo, podem-se determinar as entradas do controlador correspondente: ao arranque, à posição 3, à posição 4 e à marcha à ré.

A regulação do controlador efetua-se mediante um sistema com microprocessador, o qual recebe a informação das entradas pelo interruptor múltiplo (já mencionado) e pelos detectores que enviam dados do estado do motor (velocidade, pressão e mudança de velocidades). As saídas do sistema são adequadamente amplificadas e aplicadas a transistores de potência que, por sua vez, regulam as bobinas do mecanismo eletromagnético.

A figura 6 nos mostra um gráfico com as diferenças de velocidade na mudança de marcha, conforme esta esteja aumentando ou diminuindo. Neste caso, cabe à UCP realizar o ajuste dos saltos apresentados na mudança de marcha (conforme podemos ver na figura 6), regulando as zonas mortas ou "ciclos de histereses".

Deve-se prever se o carro vai trafegar em terreno plano ou montanhoso (posição 3 e posição 4), ou, ainda, se o mesmo vai trafegar em condições especiais, tais como: pista escorregadia ou pista com neve, devendo-se evitar a troca de marchas. Nestas condições, a UCP deverá calcular a aceleração ou desaceleração do veículo e proibir a mudança de marcha, quando seus valores forem anormais e estiverem incluídos dentro de uma condição especial. A figura 7 nos mostra um diagrama es-

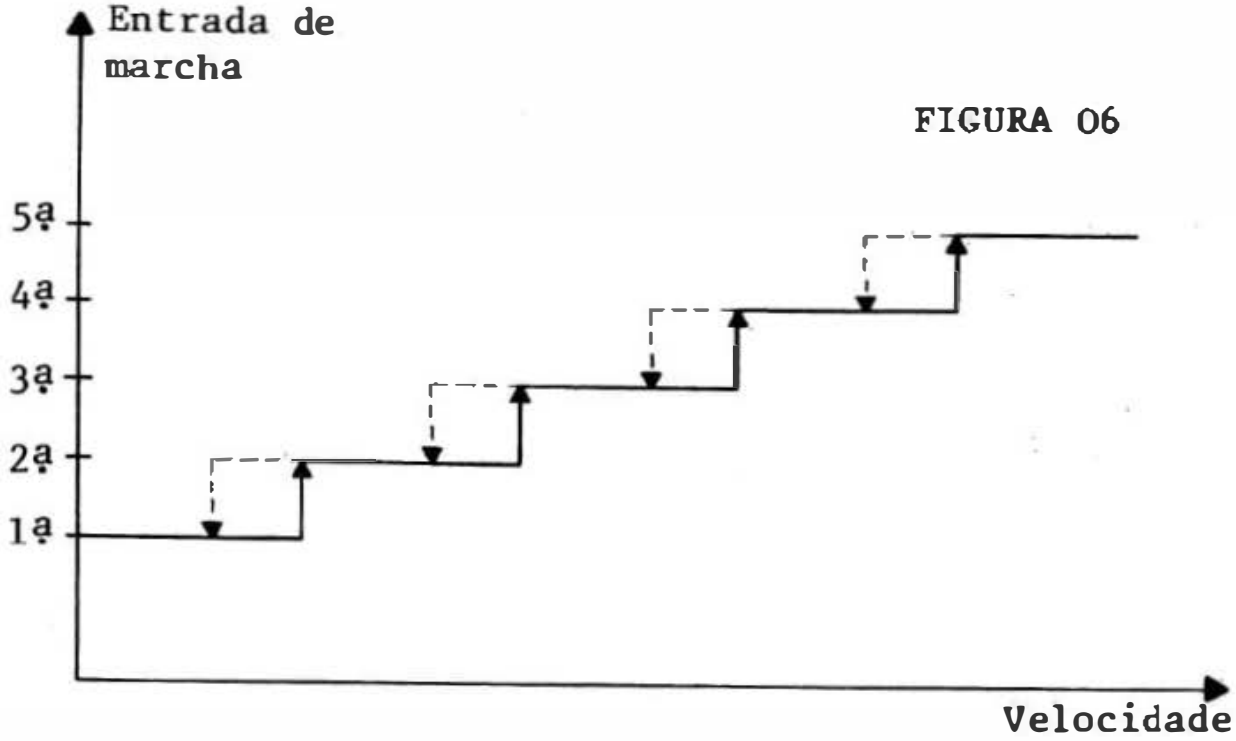
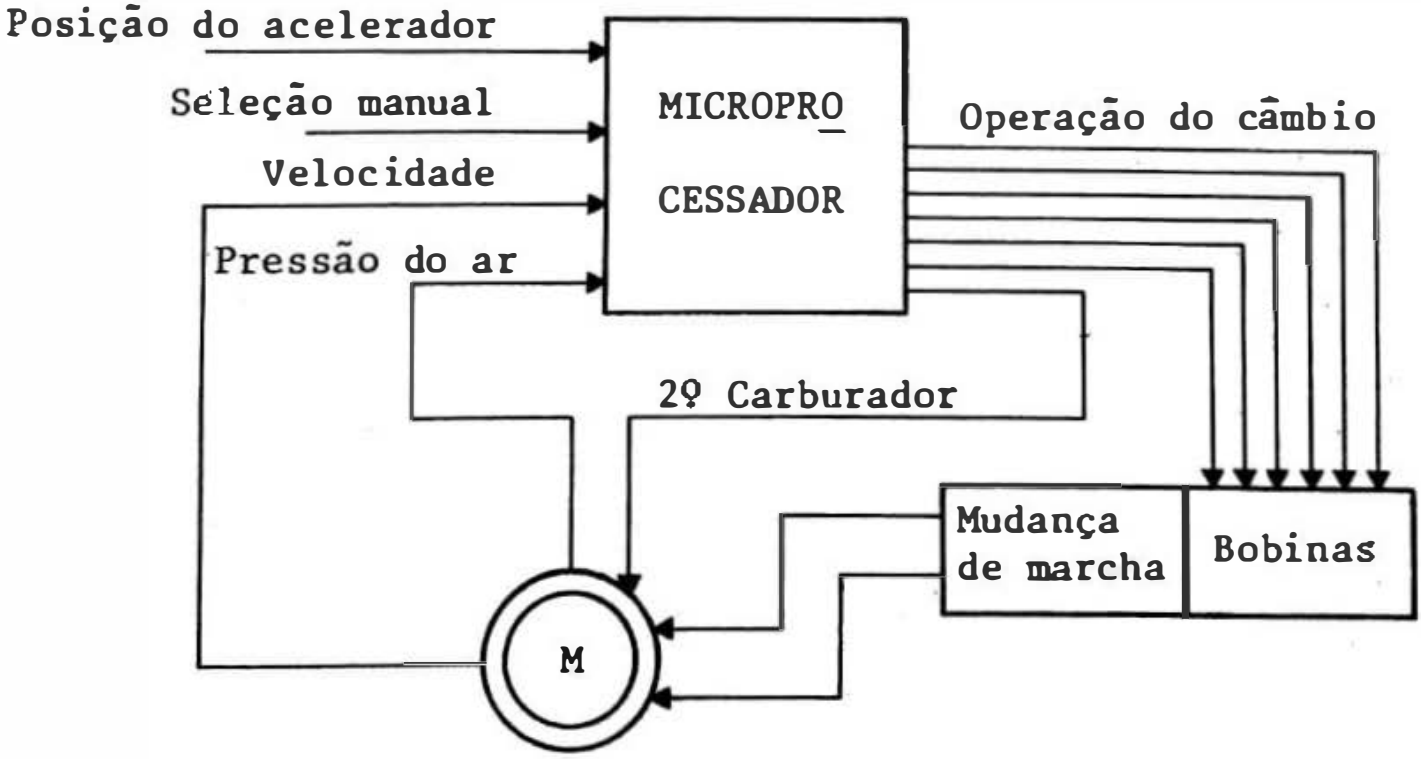


FIGURA 06

quemático do sistema com microprocessador utilizado no controle de mudança de marchas de um veículo.

FIGURA 07



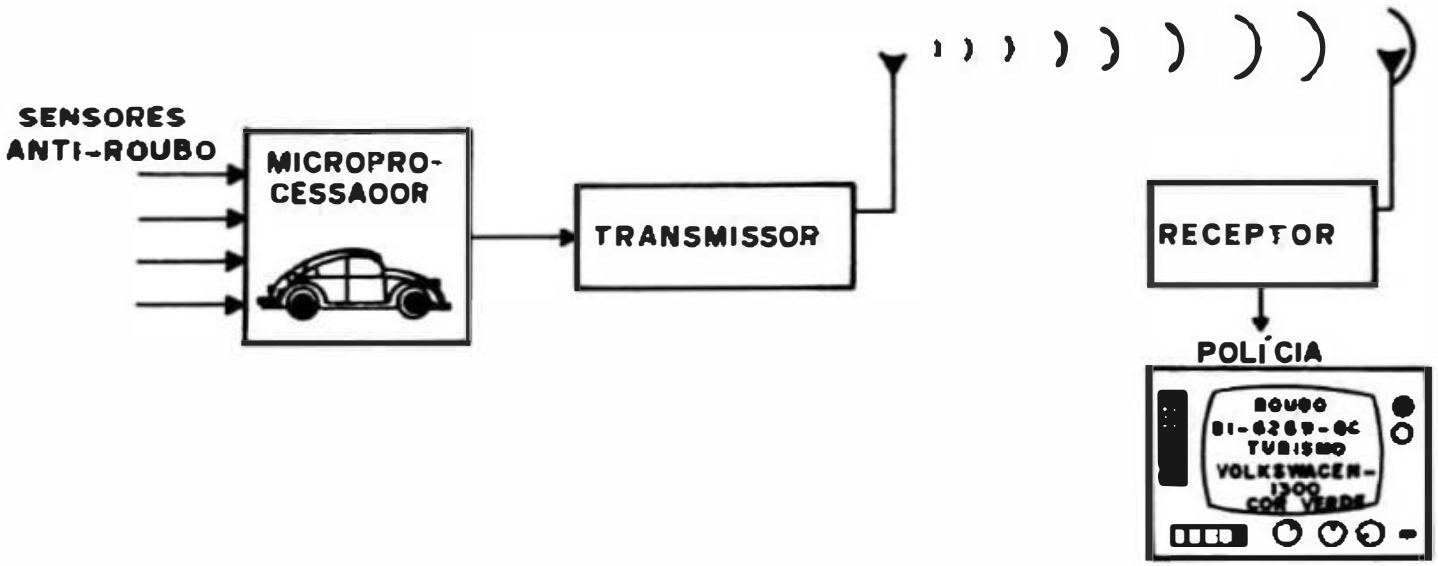
Um outro exemplo de aplicação dentro deste campo é a otimização do consumo de combustível para o máximo rendimento do motor, cujo cálculo envolve numerosas variáveis, tais como:

- temperatura do motor;
- velocidade;
- pressão do ar;
- posição do acelerador.

A memória ROM irá "manejar" uma fórmula que relacionará todos os parâmetros de entrada, de modo a proporcionar uma óti ma entrada de combustível no carburador.

Uma aplicação que está sendo muito utilizada atualmente são os sistemas anti-roubo. Estes sistemas desenvolvidos em torno de um microprocessador são altamente sofisticados. A fi gura 8 nos mostra o diagrama geral de blocos de um sistema anti-roubo, baseado em um microprocessador.

FIGURA 08



Este sistema permite enviar para a delegacia mais próxi- ma, via rádio, uma mensagem de socorro, indicando os dados e as características do veículo furtado, além de fornecer tam- bém um sinal que permite a localização do mesmo.

Neste caso, um completo conjunto de sensores anti-roubo é estrategicamente instalado nas portas, nos dispositivos de partida e mudança de marcha, nas ventarolas, na bateria, etc. os quais formam o conjunto de entradas do sistema com micro-

processador. No instante em que ocorrer a ativação de uma dessas entradas, imediatamente um receptor de rádio passa a transmitir a mensagem de socorro na mesma freqüência em que o sistema de rádio da polícia local opera.

Por outro lado, temos uma aplicação bem mais sofisticada que a anterior e que proporciona uma série de informações a respeito do veículo. Trata-se, pois, do "computador de bordo". Nesta área, existem vários projetos em andamento. Um destes projetos procura, através da utilização de um microcomputador, dar ao automóvel um quadro (painel) de comando semelhante ao de uma aeronave.

Este equipamento consta de um teclado formado por 12 teclas e um terminal de vídeo para visualização das informações. O display oferece uma certa "magnetude" de acordo com a tecla acionada, as quais podem controlar ou prestar informações sobre as seguintes variáveis:

- temperatura exterior;
- alarme, quando a temperatura exterior for menor que -3 °C, pois há perigo de ocorrência de neve;
- distância que falta para percorrer sobre um valor determinado;
- hora provável de chegada ao destino;
- velocidade média durante o percurso;
- alarme, quando se ultrapassa a máxima velocidade permitida;
- consumo médio de combustível;
- autonomia do veículo com o combustível existente no depósito (tanque) do mesmo;
- aviso da aproximação da chegada a uma distância pré-determinada;
- hora em que se efetuará a conexão da calefação ou do sistema de ar refrigerado em um estacionamento;
- código para a chave do carro, evitando-se o uso indevido do mesmo. Esta função está intimamente ligada com o sistema de alarme.

Resta dizer que existem, em desenvolvimento, outros projetos bem mais arrojados que o anterior e que muito em breve deverão fazer parte dos equipamentos dos veículos.

c) SISTEMAS COM MICROPROCESSADORES APLICADOS NA INDÚSTRIA ELÉTRICA E DE ELETRODOMÉSTICOS

Nesta área, uma das aplicações dos sistemas com microprocessadores em desenvolvimento é o controle de acionamentos de velocidade variável em motores de corrente alternada (C.A.).

Outras aplicações que se enquadram neste campo são:

- Gerador senoidal trifásico de tensão e freqüência variável, utilizado no controle de motores de indução trifásico de baixa ou de alta rotação.

- Automatização de central elétrica. Este projeto propõe o controle do arranque (partida) e da parada de dois grupos de motores, além da regularização proporcional/integral da potência em função do nível na câmara de carga.

- Freio para parada automática de um alternador.

- Sistemas de controle dos processos de lavagem, seus programas e sinais de alarmes em máquinas de lavar roupas automáticas.

- Máquinas de costura e de bordar.

- Temporizadores para eletrodomésticos: regula funções de tempo em frigoríficos, fornos de microondas, máquinas de lavar roupas e de lavar pratos (louça), fogões, etc...

- Regulação e controle de sintonia, em aparelhos de televisão.

- Comando à distância de televisores, à base de raios infravermelhos.

- Controle de cozinhas pré-programáveis.

- Controle de consumos e serviços, sistema muito utilizado em hotéis para contabilização de chamadas telefônicas, comidas e bebidas, serviços prestados, etc...

- Calibrador de motores. Trata-se de equipamento dotado de um sistema com microprocessador que, em um intervalo de tempo relativamente pequeno, subministra (ou fornece) a informação necessária para o equilíbrio de qualquer tipo de peça rotativa.

d) APLICAÇÕES DE MICROPROCESSADORES NO CAMPO DAS INDÚSTRIAS DIVERSAS

Nesta área, as possibilidades de aplicação dos microprocessadores são imensas. A seguir, destacamos algumas aplicações de relevante importância e que já se encontram disponíveis no mercado:

- sistemas de segurança e alarmes complexos e sofisticados;
- tratamento de águas;
- instrumentação para processos que se desenvolvem nos laboratórios de fotografia;
- controle automático para soldagem por microprocessador.

2) APLICAÇÕES NO CAMPO DA ELETROMEDICINA

A eletromedicina é um dos campos mais idôneos na implementação de sistemas através da utilização de microprocessadores. Atualmente, nos grandes hospitais, a maioria dos aparelhos modernos possue algum tipo de controle eletrônico. A aplicação da eletrônica na medicina é muito grande e a tendência é aumentar mais, pois a cada dia que passa, novos dispositivos e novos aparelhos são lançados no mercado.

Dentro do campo da eletromedicina, podemos citar as seguintes aplicações como exemplos de utilização de sistemas com microprocessadores:

- Sistemas de monotorização;
- Mesas de radiologia;
- Sistemas de imagem generalizada;
- Visualizador gammagráfico;
- Sistema com microprocessador para o controle de um gerador de raios X;
- Regulação do motor que guia o anodo do tubo de raios X;
- Marca-passos com microprocessador;
- Sistemas de diagnósticos;
- Sistemas de apoio a pacientes com defeitos físicos;
- Sistemas de auxílio à visão, audição, fala, locomoção, etc...

Esses são apenas alguns exemplos. Existem muitas outras aplicações dentro do campo da eletromedicina e todas elas trazem em seu bojo a preocupação de alguém em salvar vidas humanas.

Na próxima lição, continuaremos a discorrer sobre a aplicação de sistemas com microprocessadores nos campos da instrumentação, dos terminais inteligentes e dos jogos e derivados.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-17

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Os microprocessadores surgiram por volta do ano de:

a) 1951;

b) 1961;

c) 1971;

d) 1981.

2) A aplicação dos sistemas com microprocessadores em todos os campos será possível devido:

a) ao aumento dos custos dos componentes;

b) à redução dos custos dos componentes;

c) ao aumento da intensidade de integração;

d) os itens b e c estão corretos.

3) O grande desenvolvimento do setor da microeletrônica se deve:

a) às grandes indústrias e corporações;

b) exclusivamente aos governos e empresas estatais;

c) exclusivamente aos aficcionados na eletrônica;

d) N.R.A.

4) A substituição dos circuitos eletrônicos de chaveamento, geralmente de tecnologia TTL, por sistemas com microprocessadores, apresenta as seguintes vantagens:

a) possibilita o emprego de placa standard;

b) redução dos custos de mão-de-obra e verificação;

c) redução do volume ocupado e do gasto de energia consumida;

d) todas as respostas estão corretas.

5) As aplicações industriais de sistemas com microprocessadores caracterizam-se por:

a) baixo custo;

b) utilizarem sensores e mecanismos de controle;

c) serem altamente complexas,não podendo ser utilizadas na prática;

d) N.R.A.

6) Os transdutores são:

a) medidores de magnetude;

b) detectores de magnetude;

c) circuitos de interface;

d) os itens a e b estão corretos.

7) Na indústria automobilística os sistemas com microprocessadores podem ser utilizados para:

a) controle de tráfego;

b) regular a injeção de combustível e o tempo de ignição e, também, para favorecer uma redução no consumo;

c) sistemas de alarme, utilizados na detecção de avarias ou falhas previsíveis;

d) todas as respostas estão corretas.

8) A aplicação que visa à otimização do consumo de combustível para o máximo rendimento do motor envolve as seguintes variáveis:

a) temperatura do motor e pressão do ar;

b) posição do acelerador e velocidade;

c) os itens a e b estão corretos;

d) N.R.A.

9) Na indústria elétrica e de eletrodomésticos, uma das principais aplicações diz respeito:

a) ao controle de acionamento de velocidade variável em motores de corrente alternada;

b) ao controle de acionamento de velocidade constante em motores de corrente contínua;

c) os itens a e b estão corretos;

d) N.R.A.

10) No campo da eletromedicina, temos, como exemplo de aplicação de sistemas com microprocessadores:

a) tratamento de águas;

b) sistema com microprocessador para o controle de um gerador de raios X;

c) freio para parada automática de um alternador;

d) N.R.A.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-18

APLICAÇÕES DE MICROPROCESSADORES II

INTRODUÇÃO

Dando continuidade ao assunto iniciado na lição MC-17 (Aplicações de Microprocessadores I), veremos agora alguns exemplos de aplicações de microprocessadores nos campos da instrumentação, dos terminais inteligentes e dos jogos e derivados.

3 - APLICAÇÕES NO CAMPO DA INSTRUMENTAÇÃO

Trata-se de um campo de aplicações relativamente grande, onde as magnetudes a serem medidas são feitas com o auxílio de um microprocessador. Atualmente, os principais fabricantes mundiais de aparelhos de medida e instrumentação estão optando pelo uso do microprocessador em seus produtos acabados. Lo

go em seguida, apresentamos, a título de exemplo, algumas aplicações realizadas neste campo:

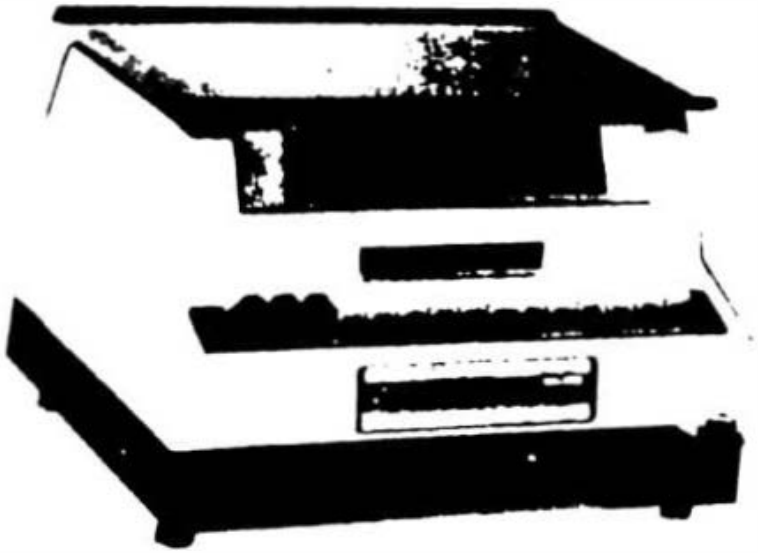
a) Caixas Registradoras

Utilizadas nas casas comerciais, como controle de vendas, são dotadas de impressora, visualizador e teclado, o que permite múltiplas possibilidades de uso. As caixas registradoras mais modernas, além de possuírem visualizador, impressora e teclado, possuem memória e podem ser conectadas a um computador central.

b) Balança para pesagem

Este tipo de balança (veja a figura 1) é muito empregado no comércio varejista, principalmente nas grandes redes de supermercados. Essas balanças medem o peso exato, calculam as importâncias parciais e totais correspondentes e memorizam o conjunto de operações realizadas. Normalmente, estas balanças possuem teclado e visualizador.

FIGURA 01



c) Painéis visualizadores

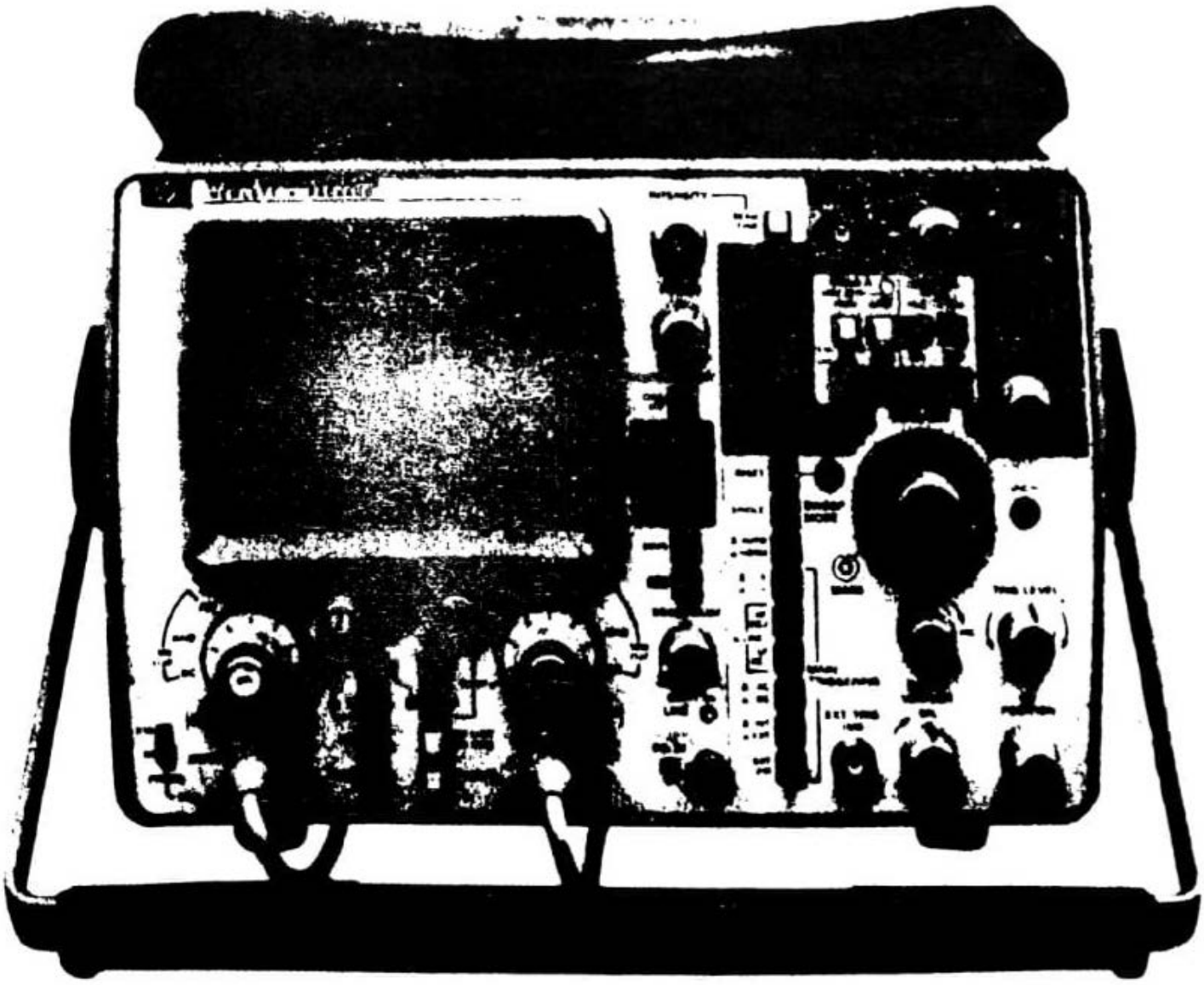
Trata-se de grandes painéis, controlados por microprocessador, destinados a informar o público a respeito de resulta-

dos ou acontecimentos de competições esportivas, de espetáculos e shows, de operações nas bolsas de valores, de operações bancárias, etc...

d) Osciloscópios especiais de duplo traço

A introdução dos microprocessadores nos osciloscópios nos proporcionou uma série de vantagens, entre as quais, podemos citar: funções de visualização para a medida exata de tempos, proporcionando o cálculo direto da voltagem de corrente contínua (Vcc), da freqüência, da voltagem instantânea e porcentagens de amplitude, além de permitirem medidas exatas de tempos sobre um determinado sinal ou sobre sinais distintos. A figura 2 nos mostra uma fotografia de um osciloscópio, equipado com microprocessador.

FIGURA 02

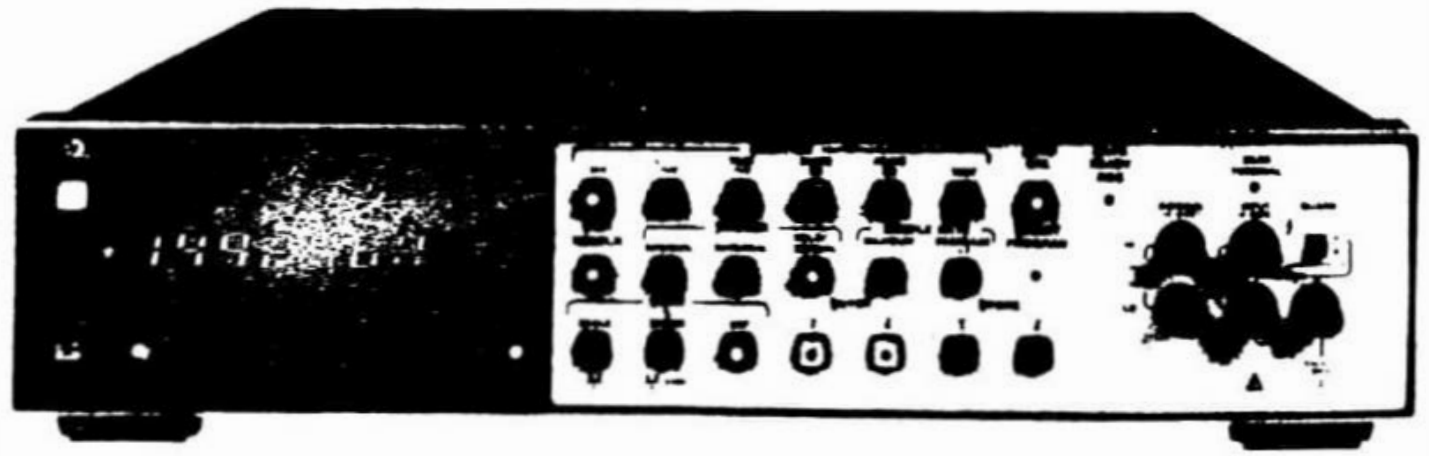


Outros modelos mais sofisticados, por utilizarem um microprocessador, realizam diretamente o cálculo da freqüência além de detectarem qualquer tipo de erro, com relação ao manejo do aparelho, assim como no comportamento dos seus circuitos internos.

e) Voltímetros e Multímetros digitais

Atualmente se podem encontrar no mercado diversas versões de multímetros digitais, dotados de um sistema com microprocessador, os quais apresentam uma série de melhorias sobre os multímetros e voltímetros digitais que não possuem microprocessadores. Entre essas melhorias podemos citar: conversão analógico/digital, comutação automática de escalas, autocalibração, além de permitir a medição de capacitâncias, freqüências e temperaturas. A figura 3 nos mostra um multímetro digital, dotado de um sistema com microprocessador.

FIGURA 03

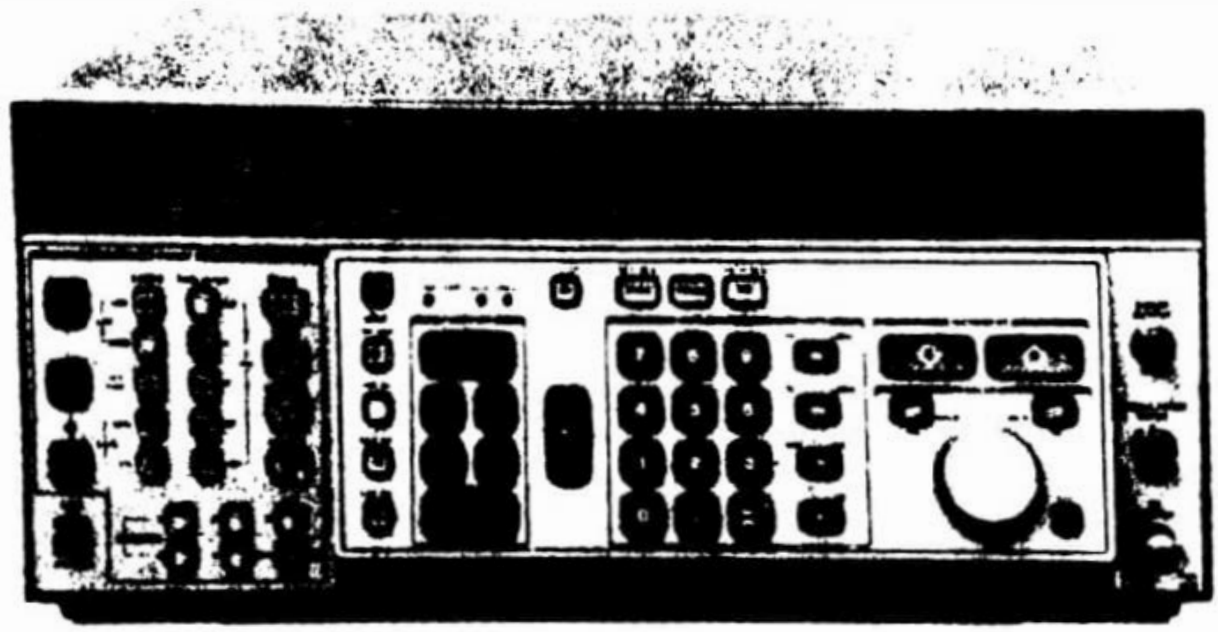


f) Geradores de funções

Este tipo de instrumento está destinado a gerar, fundamentalmente, todo tipo de ondas (senoidais, dentes de serra, quadradas, etc.) em diferentes freqüências. O microprocessa-

dor pode ser utilizado para formar o circuito sintetizador, o qual proporciona os diferentes sinais, tratando aritmeticamente a freqüência fundamental originada (ou gerada) por um oscilador, controlado por um cristal de quartzo. A figura 4 nos mostra um gerador de funções, dotado de um sistema com microprocessador.

FIGURA 04



g) Provador de Circuitos Integrados

Mediante um sistema à base de 3 microprocessadores, um conceituado fabricante de equipamentos projetou e desenvolveu um aparelho capaz de verificar qualquer tipo de circuito integrado, desde um microprocessador até um circuito integrado SSI, de tecnologia bipolar, PMOS ou NMOS. Este instrumento é dotado de uma tela (tubo de raios catódicos) para visualização de formas de ondas, um teclado e diferentes "soquetes" como base de provas.

Para se verificar o funcionamento dos circuitos integrados foi desenvolvida uma série de programas específicos para os diferentes tipos de C.Is. Esses programas se encontram armazenados em fitas cassetes e trabalham com velocidade relativamente alta.

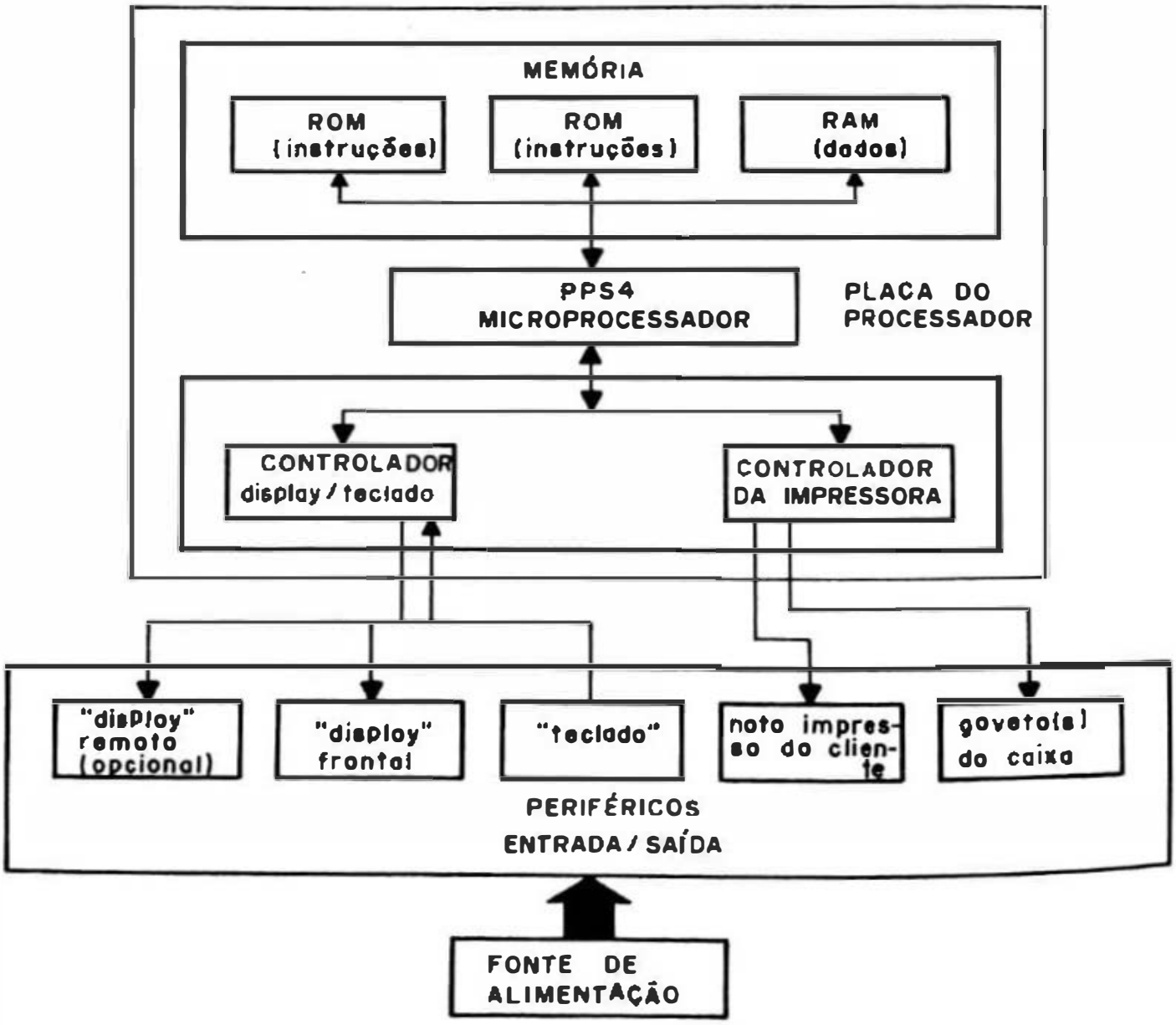
4 - APLICAÇÕES NO CAMPO DOS TERMINAIS INTELIGENTES

Trata-se de máquinas de "gestão" que, ao serem associadas a um sistema com microprocessador, se transformam em máquinas operativas capazes de desenvolverem trabalhos muito interessantes. Eis alguns exemplos de aplicações dentro deste campo:

a) "Terminais de venda"

Trata-se de uma caixa registradora "inteligente", que proporciona uma série de informações. Adaptáveis a vários tipos de negócios, essas caixas podem resolver alguns tipos de problemas de gestão a ela relacionados. A figura 5 nos mostra o diagrama de blocos deste tipo de caixa registradora.

FIGURA 05



Esses terminais de vendas, normalmente, são dotados de teclados completos, o que contribui para o aumento da capacidade de operações. Além disso, possuem um amplo display, memória integrada e possibilidade de conexão com uma unidade "mestra" (central) de processamento.

Uma atividade comercial que utiliza por demais este tipo de terminal de vendas são os supermercados. Neste ramo de atividade, os requisitos de equipamento podem variar, pois um pequeno estabelecimento pode necessitar apenas do equivalente eletrônico de uma caixa registradora, enquanto que uma cadeia de lojas requer unidades com interligação múltipla, cada qual com escalas eletrônicas, notas impressas minuciosamente e leitoras remotas.

Os requisitos computacionais e de informação, também podem variar, já que cada uma das cadeias de supermercados lida com métodos diferentes, ou seja, possui suas peculiariedades, quando da manipulação de cheques, dinheiro, cupons de desconto e até mesmo depósitos de garrafas. Além disto, as taxas referentes aos impostos podem variar, dependendo do estado ou da cidade. E, finalmente esses terminais de vendas precisam fornecer dados de controle e administração que satisfaçam às necessidades de cada cadeia, em particular.

A figura 6 nos mostra o diagrama de blocos de um terminal de venda de supermercado, controlado a microprocessador.

Através do diagrama de blocos, podemos verificar como a impressora, o teclado, a escala de peso e os vários periféricos estão conectados ao microprocessador (ou processador central) IMP-16, por meio de barramentos. O decodificador de endereço e ordem manipula códigos de ordem de 3 bits e códigos de endereço de 5 bits, numa única linha de endereço e uma só linha de ordem, que percorrem todos os periféricos.

b) Terminais para Bancos

Este tipo de terminal, utilizado nas agências bancárias, pode imprimir e memorizar as operações de contas correntes. Quando interligado a um computador central, passa a ter a capacidade de controlar o crédito e todo o tipo de transações, de forma rápida e segura.

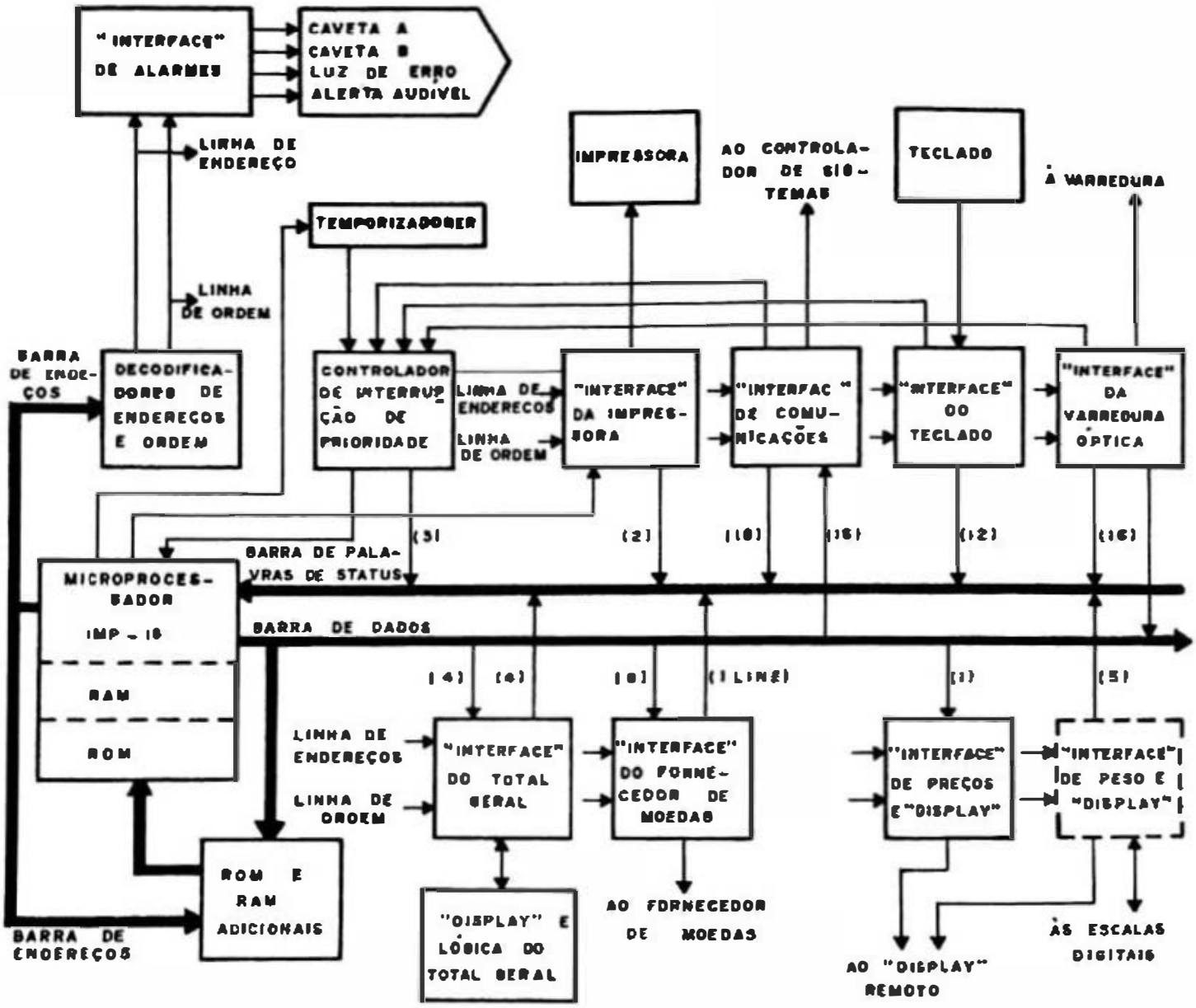


FIGURA 06

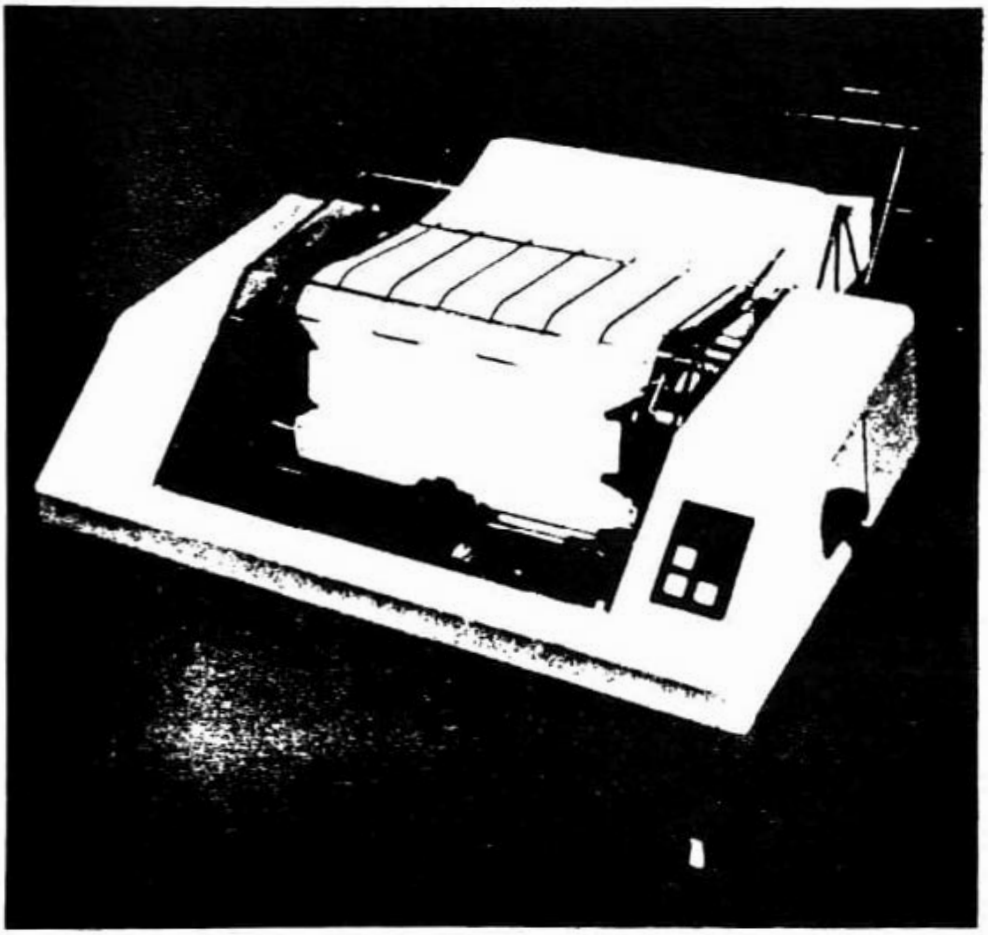
c) Sistemas de controle, impressão e visualização de cronometragem de caráter esportivo

Estes sistemas são largamente utilizados em competições esportivas das mais diversas modalidades. Normalmente, encontram-se instalados em quadras esportivas, estádios de futebol, piscinas, hipódromos, autódromos, etc... Este tipo de aplicação, na prática, pode confundir-se com um outro tipo de aplicação, já citado, que são os "painéis visualizadores".

d) Impressoras de caracteres alfanuméricos

O assunto sobre impressoras será abordado em lições posteriores. Apenas, a título de exemplo, mostramos na figura 7, uma impressora do tipo alfanumérico.

FIGURA 07



e) Periféricos interpretadores de escrita manual

Trata-se de um terminal de mesa, capaz de traduzir a escrita feita pelo homem e transferir essas informações a um computador central, para o devido processamento.

f) Todo o tipo de terminais inteligentes, com tela de vídeo e teclado

Entende-se como terminal inteligente, todo terminal de computador que possua, no mínimo, um teclado e uma tela de vídeo. A figura 8 nos mostra um terminal típico.

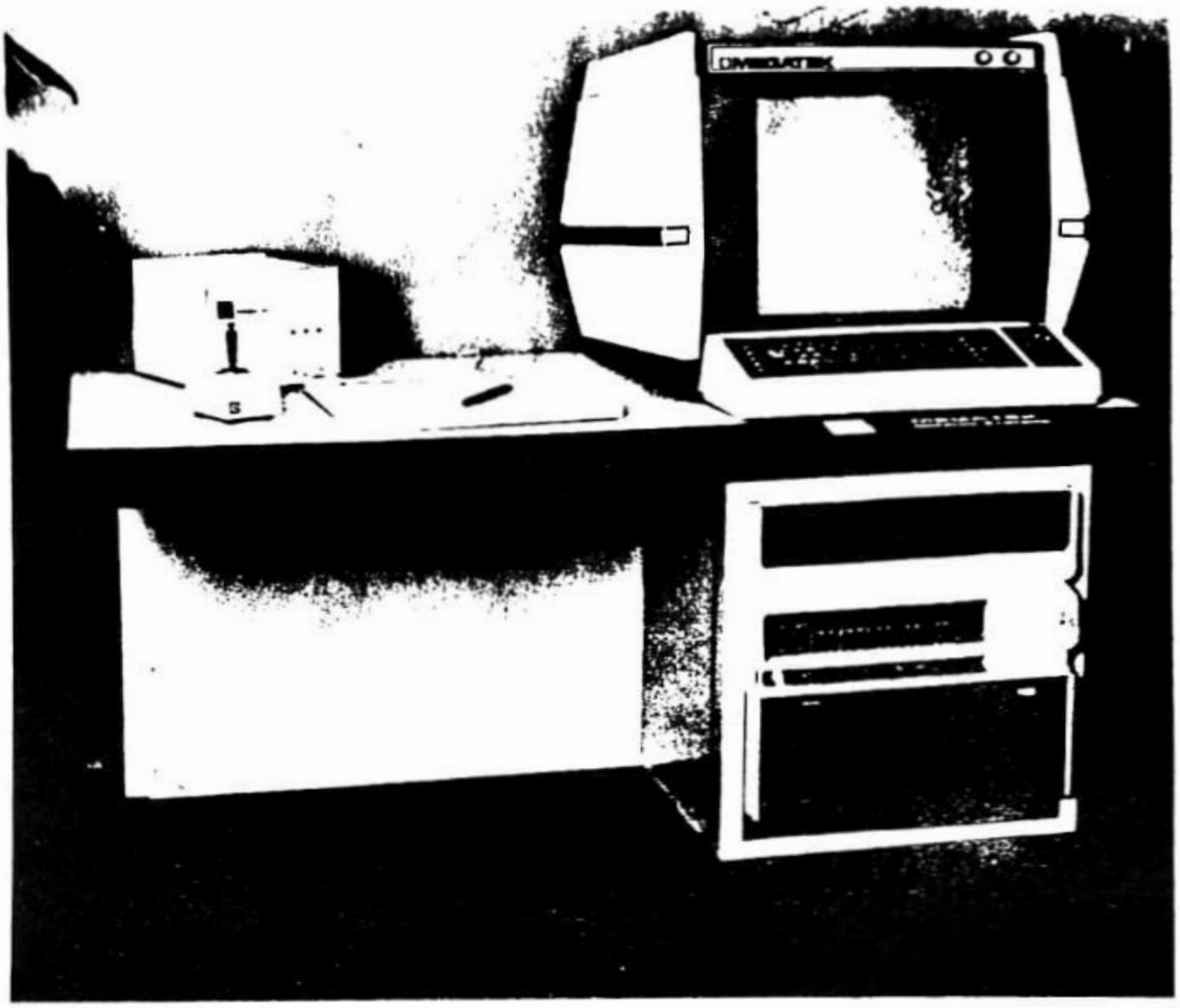


FIGURA 08

g) Controladores de periféricos

Os controladores de periféricos são aparelhos destinados a "reduzir" a utilização da UCP ou do próprio computador, com base no tempo em que cada um desses periféricos requisita a UCP.

4 - APLICAÇÕES NO CAMPO DOS JOGOS E DERIVADOS

Atualmente, já existe no mercado uma gama relativamente grande de aparelhos controlados por microprocessadores, destinados à área de lazer, ou mais especificamente, ao campo dos jogos e derivados.

Os primeiros jogos de vídeo, desenvolvidos com base na lógica TTL, eram tão simples que em pouco tempo de uso a maioria dos jogadores perdia o entusiasmo pelos mesmos. Os fabricantes, por sua vez, pouco podiam fazer, pois não tinham condições de projetar sua própria lógica TTL, de modo que lhes restava adquirir os projetos e, às vezes, o próprio conjunto lógico. Porém, com o advento dos microprocessadores, tornou-se possível aumentar a complexidade dos jogos, tornando-os mais emocionantes e mais competitivos. Essa complexidade pode ser multiplicada de duas para cinco vezes e, ao fabricante, foi permitido realizar as programações em seus próprios estabelecimentos. Além disso, a utilização de sistemas com microprocessadores reduziu o tempo de desenvolvimento desses projetos em torno de 75%.

É evidente que a transição da lógica TTL para os microprocessadores não se efetuou sem dificuldades. Uma primeira dificuldade com que os técnicos esbarraram foi em relação à freqüência de dados necessária à rápida varredura que mantém as imagens na tela, a qual está bem além da capacidade de um microprocessador. Depois de alguns estudos, este problema foi sanado através do uso de uma memória RAM.

O treinamento rapidamente elevou a eficiência dos técnicos, nessa área. As memórias do tipo ROM passaram a ser utilizadas para resolver eventuais problemas ocorridos com o emprego das memórias RAM. Com o auxílio das memórias ROM e a sua crescente experiência, os técnicos puderam reparar ou melhorar o dobro de circuitos defeituosos.

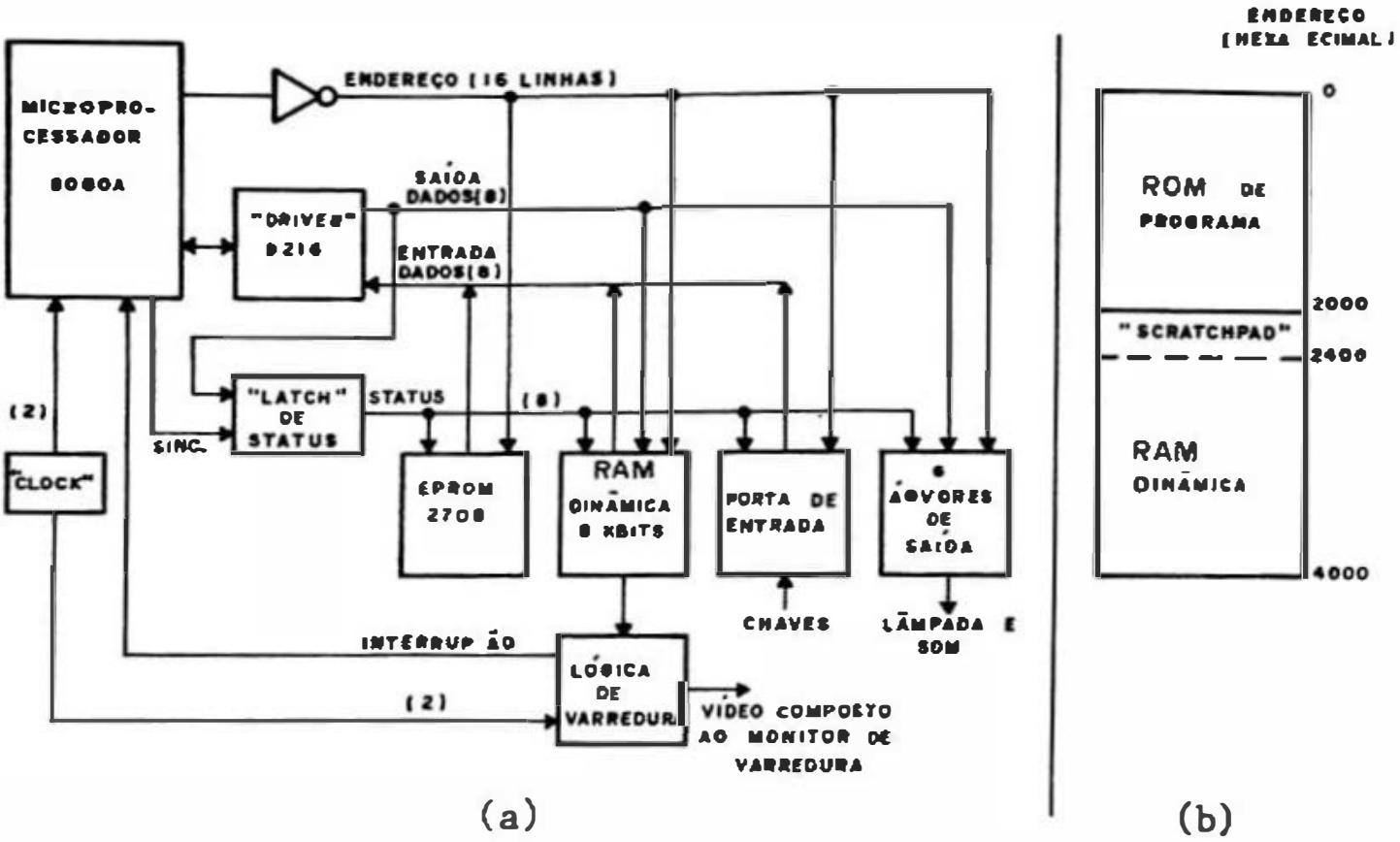
Em seguida, foi realizada a adaptação do microprocessador ao vídeo, com as memórias RAM armazenando grandes quantidades de informações de imagem, as quais eram lidas e enviadas à tela, ao ritmo de um bit por vez, contrariando assim os circuitos convencionais de deflexão horizontal e vertical que trabalhavam como qualquer outro existente nos aparelhos domésticos de TV.

O traçado na tela é formado por 244 linhas horizontais, sendo que cada linha é formada por 256 pontos. A memória RAM, então, deve ser capaz de receber os 244 x 256 bits de informação. Neste tipo de arranjo, a função do microprocessador é a de mudar o conteúdo da memória RAM, a um ritmo lento (uma vez a cada nova imagem, ou uma vez a cada conjunto de imagens).

A única interação entre o microprocessador e a lógica de varredura da tela é um sinal de interrupção, na porção inferior da tela, que o microprocessador utiliza para fins de sincronização e temporização.

A figura 9 nos mostra o diagrama de blocos de um circuito de jogo típico.

FIGURA 09



No exemplo apresentado, o microprocessador responde a comandos vindos de chaves controladas pelos jogadores e forma novas imagens, de acordo com o programa armazenado numa memória PROM (figura 9-a). O mapa da memória (figura 9-b) mostra as ROMs e RAMs para um jogo típico.

A UTILIZAÇÃO DE UMA MEMÓRIA RAM PARA A FORMAÇÃO DA IMAGEM

Em uma tela de 244 linhas e 256 pontos por linha, necessitamos de uma memória RAM de 57 K bits (57.344 bits), de modo a corresponder ao desenho de 57.344 pontos (244x256) da te

la. O circuito de rastreiamento lê o conteúdo da memória em 1/30 de segundo. O microprocessador emprega uma área de 1K byte por 8 bits da memória RAM como "scratchpad", sem perda significativa da resolução vertical. As portas entrada/saída convencionais são manipuladas pelo microprocessador como chaves liga/desliga.

Um jogo típico de pequena complexidade requer 8K bits de memória ROM para o programa. O microprocessador "escreve" a imagem na tela, movimentando as variações bidimensionais dos desenhos, da memória ROM para a memória RAM, um byte por vez.

Cada deslocamento de um byte move o desenho, de modo a simular movimento. A interrupção limita o número de operações dentro da rotina de interrupção. Como a interrupção produz os sinais síncronos, ele providencia informação de temporização ao programa principal. Seus precisos tempos de "clock" controlam uma chave de saída, que deve ser mantida aberta durante um certo período de tempo. E, por fim, a interrupção "verifica" a condição das portas de entrada, além de alertar a memória sobre certas mudanças.

O PROGRAMA DE JOGO

Um programa de jogo é, normalmente, dividido em duas partes: a lógica que produz os desenhos e a interrupção. O programa principal coloca uma ação em movimento e verifica a sua execução, enquanto que a interrupção move os desenhos e informa o programa principal, se as condições esperadas foram satisfeitas. O desenho, por exemplo, poderia ser um carro no ponto A, com velocidade B, deslocando-se numa certa direção, até que a condição C fosse satisfeita. A função da interrupção será a de movimentar esse desenho do ponto A, a uma velocidade B e informar o programa, quando a condição C for satisfeita. O programa principal então, inicia a próxima operação, parte da qual poderia ser uma sinalização do tipo visual ou audível. A interrupção atua a partir do pulso de apagamento, quando a varredura atinge a parte inferior da tela de vídeo.

Vejamos agora alguns tipos de aplicações de sistemas com microprocessadores neste campo.

a) Fliperamas

Trata-se de aparelhos dotados de teclas e/ou alavancas

de comando e uma tela para visualização (veja figura 10).

FIGURA 10

Dependendo do tipo de jogo, esses aparelhos podem aceitar até dois jogadores. Geralmente esses aparelhos apresentam o número de jogadas e o total de pontos que cada jogador obtém. Os fliperamas já são comuns hoje em dia e encontram-se instaladas em casas especializadas neste ramo.

b) Jogos de Xadrez

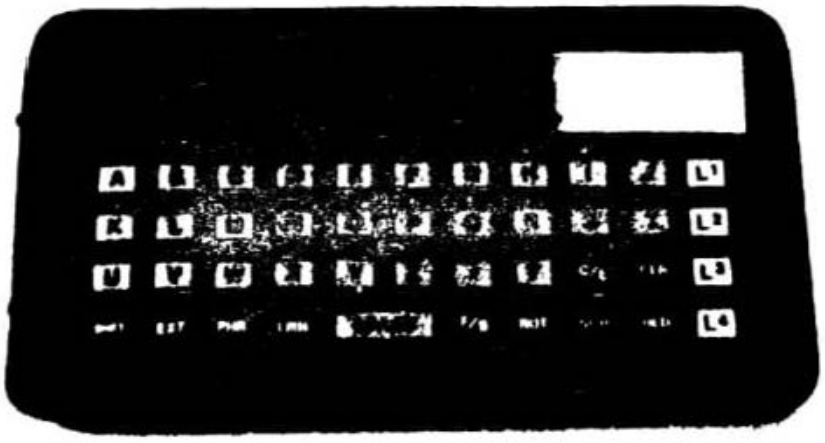
Trata-se de um aparelho constituído de um tabuleiro de "xadrez". As jogadas são realizadas e os dados correspondentes às mesmas são enviados para um sistema com microprocessador, através de um teclado. Muitos desses aparelhos possuem um visor alfanumérico (constituído de um ou mais displays), cuja finalidade é mostrar a jogada que foi feita.

Alguns tabuleiros de jogos de xadrez mais modernos possuem vários níveis, adaptando-se assim à capacidade do jogador, além de emitirem mensagens sonoras através de um sintetizador de voz, podendo estas mensagens ser ditas em inglês, francês ou espanhol. Outros apresentam também um sistema gráfico para que o jogador possa orientar-se com relação às jogadas seguintes.

c) Tradutor de idiomas

Desenvolvido com base na microeletrônica, esse sistema pode ser aplicado às pequenas calculadoras de bolso e permitem a tradução de vários idiomas distintos. Estas pequenas máquinas possuem um sistema de memória recambiável, de modo a atuar em áreas específicas e de interesse do usuário, como por exemplo, a tradução técnica, comercial, etc... A figura 11 nos mostra a fotografia de um tradutor de idiomas típico.

FIGURA 11



d) Controles de jogos

Especialmente indicados para o controle de brinquedos que possuem um certo grau de movimento, tais como: trens elétricos, bonecas, carros elétricos, autoramas, etc...

e) Videogames

O videogame é um aparelho dotado de avançado circuito

eletrônico à base de um microcomputador, o qual é programado para ler e decodificar sinais digitais transmitidos pelos cartuchos, os quais darão origem às imagens coloridas, sons complexos, etc... A figura 12 nos mostra o console de um videogame.

O console (aparelho) é conectado a um aparelho de T.V., através dos terminais da antena. Geralmente, esses aparelhos apresentam um orifício retangular na parte superior, destinado à colocação dos cartuchos (tipo cassete) que contêm o programa do jogo. Os comandos que fazem parte dos aparelhos de videogames são dois "joysticks" (alavancas), utilizados pelos jogadores para controlar os jogos. Em muitos casos, o comando também pode ser feito através de um teclado alfanumérico, ou através de dois botões (paddles).

FIGURA 12

Os cartuchos que contêm os programas (softwares) dos jogos possuem internamente um circuito integrado de memória não volátil do tipo ROM, na qual é armazenado o programa traduzido numa linguagem de microcomputador (linguagem de máquina). Geralmente, a capacidade dessas memórias está entre 2 K bytes e 8 K bytes. Ao se encaixar um determinado cartucho no orifício retangular do console, forma-se o contato elétrico, o qual permitirá que se processe a leitura dos dados armazenados no circuito da memória ROM, gerando, em seguida, imagens e sons correspondentes.

Quando o usuário adquire um videogame, ele está na realidade comprando um microcomputador completo, com exceção do teclado e da memória, muito embora alguns modelos já tragam um pequeno teclado. Muitos aparelhos de videogames permitem a conexão de teclado, interface para impressora, interface para disquetes, expansão de memória, etc., de modo a torná-lo muito mais flexível em termos de jogos, podendo, dependendo da aplicação, operar como um microcomputador. A figura 13 nos mostra uma série de equipamentos que podem ser acoplados ao aparelho de videogame.

FIGURA 13

O mercado de videogames é um mercado em expansão e promete muito para o futuro, pois o desenvolvimento de tecnologias com raios laser, vídeo-disco e projeção de alta definição farão dos videogames do futuro algo deslumbrante.

Atualmente já estamos vivendo a primeira grande revolu-

ção nos videogames mais complexos, com a associação de microcomputador e sistemas de vídeo-disco interativo. O vídeo-disco armazena imagens de TV e desenhos muito mais perfeitos do que os atuais sistemas de cartuchos que utilizam chips de memória ROM. Nesse caso, caberá ao próprio usuário desenvolver e programar seus jogos, podendo, desta forma, colocar toda a sua criatividade à prova. Esse novo tipo de videogame é chamado de "interativo", constituindo-se em uma forma de lazer e em um sistema de informação bidirecional.

COMENTÁRIOS FINAIS

Com relação à aplicação dos microprocessadores, procuramos abordar diversos campos, todos eles acompanhados de exemplos práticos. Porém, pouco comentamos a respeito dos microcomputadores, os quais, sem dúvida alguma, podem ser citados como sendo o exemplo número um de aplicação dos microprocessadores. Não o citamos de maneira mais específica, pois todo o nosso curso está voltado para esta aplicação.

A primeira grande façanha dos microprocessadores foi tornar possível a confecção de máquinas de calcular de bolso, tanto populares como científicas e comerciais.

Torna-se repetitivo tentar definir o microprocessador e citar suas possibilidades. As aplicações tornam-se mais diversificadas a cada dia, principalmente devido à imaginação dos usuários e da versatilidade desses dispositivos. A evolução neste setor da eletrônica tem sido tão grande que só nos resta dizer que onde houver possibilidade da utilização de controle digital, a baixo custo, lá estarão os microprocessadores dominando.

CURSOS DE APERFEIÇOAMENTO

CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANA

EXAME DA LIÇÃO MC-18

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) As caixas registradoras mais modernas possuem:

   a) visualizador e impressora;
   b) teclado;
   c) memória;
   d) todas as respostas estão corretas.

2) As balanças do tipo apresentado na figura 1, além de medirem o peso exato

   a) calculam as importâncias parciais e totais;
   b) memorizam o conjunto de operações realizadas;
   c) os itens a e b estão corretos;
   d) N.R.A.

3) Os osciloscópios, equipados com microprocessadores, nos proporcionam:

   a) cálculo direto da voltagem e da corrente contínua(Vcc);
   b) cálculo direto da voltagem e da corrente alternada(V_{CA});
   c) os itens a e b estão corretos;
   d) N.R.A.

4) Os multímetros digitais, dotados de sistema com microprocessadores, podem realizar:

a) medidas de pressão;
b) conversão analógico/digital;
c) os itens a e b estão corretos;
d) N.R.A.

5) Num gerador de funções, o microprocessador pode ser utilizado para:

a) formar o circuito conversor;
b) formar o circuito analisador dos sinais;
c) formar o circuito sintetizador, o qual proporciona os diferentes sinais;
d) N.R.A.

6) Os primeiros jogos de vídeo foram desenvolvidos com base na:

a) lógica RTL;
b) lógica DTL;
c) os itens a e b estão corretos;
d) lógica TTL.

7) O problema com relação à freqüência de dados, necessária à rápida varredura que mantém as imagens na tela, foi sanado através do uso de uma:

a) Memória RAM;
b) Memória ROM;
c) Memória EAROM;
d) N.R.A.

8) O microprocessador "escreve" a imagem na tela, movimentando as variações bidimensionais dos desenhos, da memória ROM para a memória RAM:

a) três bytes por vez;
b) um byte por vez;
c) dez bytes por vez;
d) N.R.A.

9) Os comandos utilizados pelos jogadores para controlar os jogos, geralmente, são:

a) dois joysticks;
b) dois paddles;
c) os itens a e b estão corretos;
d) N.R.A.

10) Os cartuchos que contêm os programas dos jogos para videogames possuem internamente:

a) uma memória ROM;
b) uma memória RAM;
c) uma memória EPROM;
d) N.R.A.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-19

IMPRESSORAS

INTRODUÇÃO:

Retomando o estudo sobre periféricos, estudaremos, nesta lição, um dos periféricos de saída mais utilizados em sistemas de computação: a impressora.

Na verdade, as impressoras são dispositivos cuja função é imprimir letras, números e caracteres, semelhantes (na forma legível) aos do texto. Porém, a maioria das impressoras, além da função básica de imprimir, também desempenham uma segunda função que é traçar gráficos, utilizando-se para tanto de alguns símbolos como: X, *, - e outros. Esta função de traçar gráficos é característica dos traçadores de gráficos, porém, como acabamos de ver, não é uma função exclusiva deste dispositivo.

Vamos então reiniciar o estudo das impressoras, através da classificação das mesmas.

CLASSIFICAÇÃO DAS IMPRESSORAS:

Normalmente, as impressoras utilizadas em minicomputadores e microcomputadores podem ser classificadas em três categorias:

a) Impressoras de impacto versus sem impacto;
b) Impressoras de caracteres por matriz de pontos versus caracteres completamente formados;
c) Impressoras de um caracter por vez versus uma linha por vez.

Vejamos então algumas particularidades de cada uma dessas categorias.

a) Impressoras de Impacto versus sem Impacto:

As impressoras de impacto formam caracteres, no papel, pela forma tradicional, ou seja, "batendo" no papel com um cabeçote de impressão, o qual pressiona uma fita, com tinta, contra o papel. Ao contrário, as impressoras sem impacto formam os caracteres sem envolver o mecanismo de impressão com a superfície de impressão. Neste caso, a impressão se dá por aquecimento, utilizando-se, para tanto, um papel sensível ao calor ou por pulverização de tinta de um jato.

b) Impressoras de caracteres por matriz de pontos versus caracteres completamente formados:

Os caracteres completamente formados são iguais àqueles feitos por uma máquina de datilografia convencional, ou seja, todas as partes dos caracteres são gravados no inverso da barra de símbolos da máquina e, quando imprimidos, aparecem ligados ou completamente formados.

Já os caracteres por matriz de pontos são formados por uma combinação de pontos que formam um grupo o qual, quando visualizado, representa uma letra, um número ou um símbolo qualquer.

c) Impressoras de um caracter por vez versus uma linha por vez:

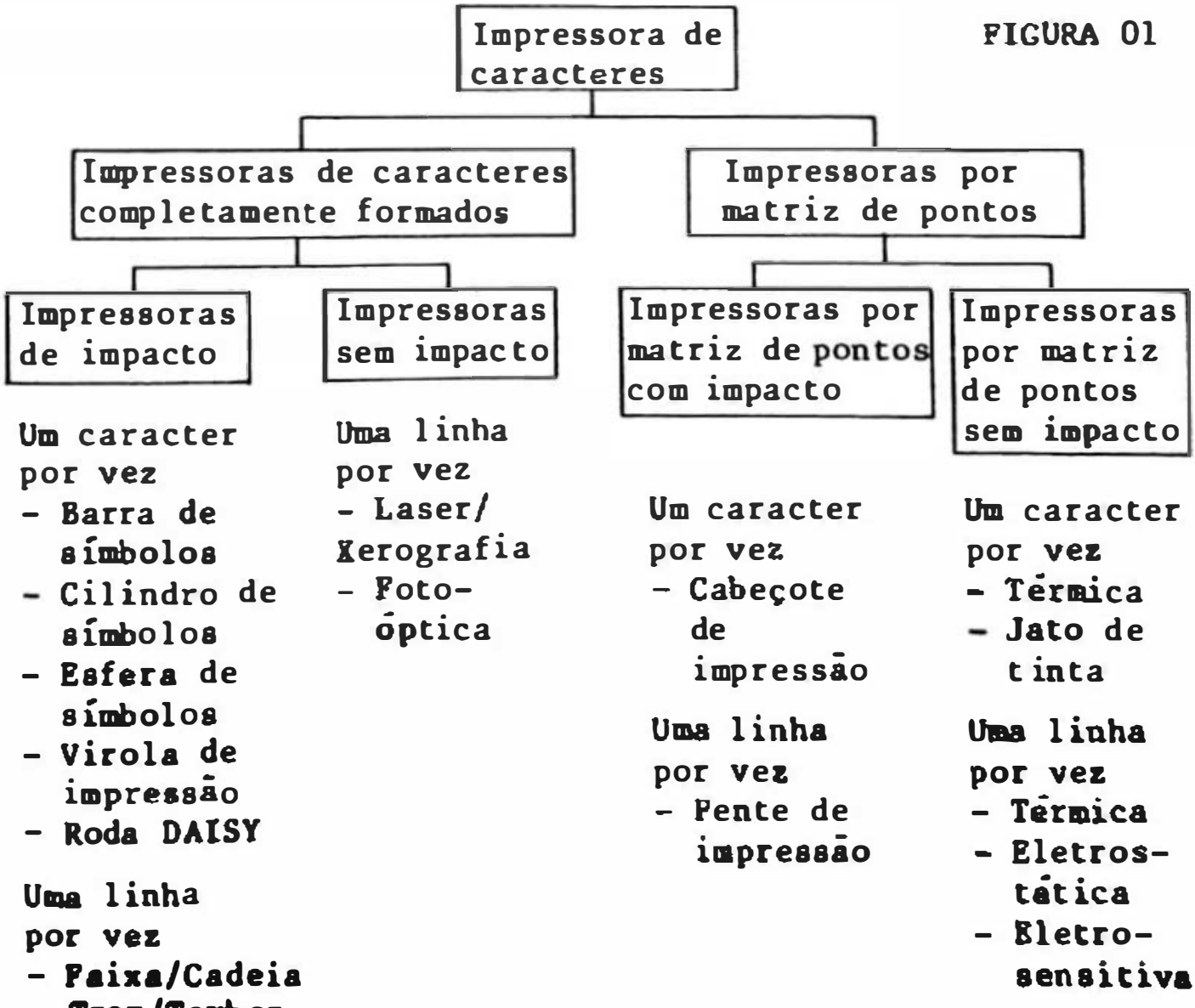
As impressoras de um caracter por vez (também conhecidas

por impressoras de caracteres ou impressoras-série) imprimem cada caracter serial e instantaneamente. As impressoras de uma linha por vez (ou simplesmente impressoras de linha) imprimem cada linha instantaneamente. Algumas impressoras, de alta tecnologia, existentes no mercado, utilizam LASER, enquanto outras utilizam métodos xerográficos. Essas impressoras imprimem as linhas tão rapidamente que, virtualmente, imprimem uma página por vez, daí o fato de muitas vezes serem referenciadas como "impressoras de página". As impressoras de página dificilmente são utilizadas em sistemas de mini e microcomputadores.

A figura 1 nos mostra a classificação das impressoras, dentro destas categorias, com classificação adicional pela tecnologia da impressora.

Note que a classificação mostrada na figura 1 inicia-se com base nas impressoras de caracteres, subdividindo-se em dois outros grupos que são as impressoras de caracteres completamente formados e as de matriz de pontos.

FIGURA 01



Na figura 2, apresentamos uma tabela na qual se encontram resumidas as características das impressoras. Esta tabela, montada com base na classificação que apresentamos, destaca, como característica fundamental, a velocidade de impressão, a qual é fornecida em c.p.s. (caracteres por segundo), em l/min (linhas por minuto) ou em l/s (linhas por segundo).

CARACTERÍSTICAS DAS IMPRESSORAS				
CLASSIFICAÇÃO	CAPACIDADE	TECNOLOGIA DE IMPRESSÃO	VELOCIDADE APROXIMADA	ANOTAÇÕES
IMPACTO	Um caracter por vez	Carro cilíndrico	10 c.p.s	Tele-impressora modelo 33
		Carro esférico	15 c.p.s.	Máquina de datilografar IBM
		Roda dentada, tambor	30 a 55 c.p.s.	Impressora em série de caracteres, completamente formados de alta velocidade
		Arame de matriz de pontos	30 a 330 c.p.s.	Impressora de impacto de caracteres, com velocidade mais alta, para uso do computador.
	Uma linha por vez	Faixa	Até 3000 l/min	Colocação de caracteres mutáveis internamente
		Cadeia ou trem	Até 2000 l/min	Colocação de caracteres mutáveis internamente

FIGURA 02

CLASSIFICAÇÃO	CAPACIDADE	TECNOLOGIA DE IMPRESSÃO	VELOCIDADE APROXIMADA	ANOTAÇÕES
		Tambor	300 a 2000 l/min	Sujeita a leve perda de registração vertical, em velocidades mais altas
		Matriz de pontos	Até 600 l/s	
SEM IMPACTO	Um caracter por vez	Matriz de pontos térmicos	30 a 120 c.p.s.	Baixo custo, baixo ruído. Requer papel especial. Não admite cópia múltipla.
		Matriz de pontos eletrostáticos	160 a 2000 c.p.s.	Requer papel especial; baixa qualidade de impressão
		Jato de tinta	30-1000 c.p.s	Não requer papel especial Não admite cópia múltipla
	Uma linha por vez	Matriz de pontos eletrostáticos	300 a 18000 l/min.	Requer papel especial a processos de timbre
		Laser ou Xerográfico	4000 a 21000 l/min	Alta velocidade, alto custo, alta manutenção, impressão de alto volume
		Foto-óptica	150-1000 l/min	Utilizada em composição com controladores, em minicomputadores

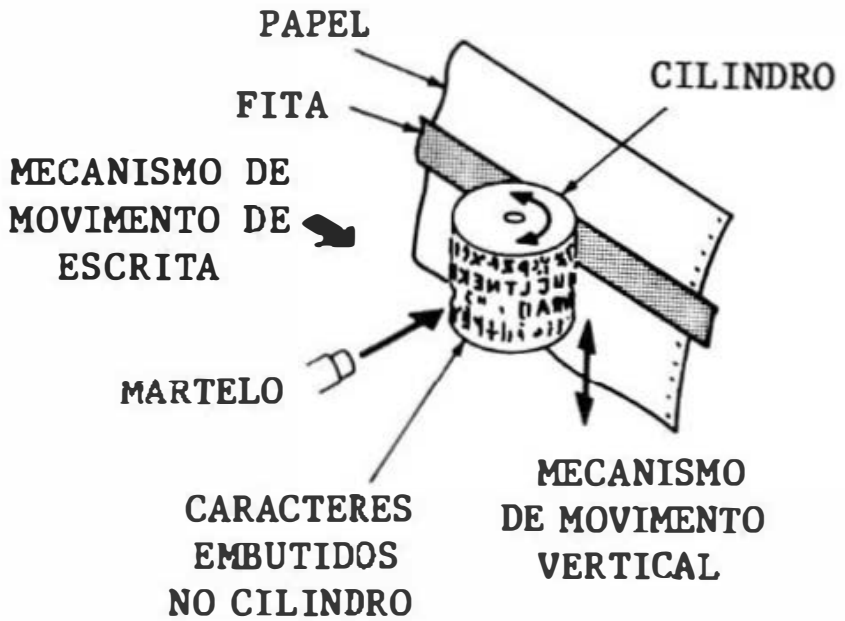
Vejamos, agora, as características dos principais tipos de impressoras:

a) Impressoras de impacto de um caracter por vez, para caracteres completamente formados:

A máquina de escrever clássica é um exemplo deste tipo de impressora. Os caracteres são ditos completamente formados porque todos os componentes do caracter são "embutidos" em cada barra de símbolo e o caracter inteiro é impresso em uma única vez. Além de serem relativamente vagarosas, as máquinas de escrever de "barra de símbolos" travam a interface de codificação do computador, dificultando assim a utilização das mesmas em sistemas de computação.

A impressora clássica, utilizada com os mini e microcomputadores no passado, era a "Teleimpressora modelo 33", projetada para comunicação impressa, através de linhas telefônicas. O baixo custo, aliado ao fácil interfeiceamento, tornou o uso desta impressora muito comum entre os pequenos computadores. O modelo 33 imprime a uma velocidade de aproximadamente 10 caracteres por segundo ou 10 c.p.s. O seu mecanismo de impressão é um cilindro vertical, ao redor do qual se encontram distribuídos (em linhas e colunas) os caracteres (veja figura 3).

FIGURA 03



O mecanismo de funcionamento é simples: o código ASCII, enviado para a impressora, é transformado em um sinal, o qual irá produzir um movimento capaz de girar o cilindro, colocando a coluna, que contém o caracter desejado, faceando o papel. Um mecanismo de martelo empurra o cilindro na direção do papel, onde somente o caracter posicionado "bate" na fita, imprimindo-se assim o caracter no papel.

Muitos computadores domésticos e mesmo comerciais, possuem uma entrada e uma saída, construídas dentro do computador, capazes de converter uma porta RS 232C para o uso de impressora modelo 33. A despeito de ser um projeto velho e também relativamente vagarosa, ainda encontramos teleimpressoras modelo 33 em uso.

Semelhante à teleimpressora modelo 33, no tocante ao cabeçote de impressão, temos a máquina de escrever IBM, cujo cabeçote de impressão está montado em uma esfera (comumente referida como bola de golfe), com os caracteres embutidos ao redor da superfície. Para prensar a fita e em seguida o papel, o código do caracter transmitido faz com que a esfera rode ao redor de um eixo vertical e se incline sobre um eixo horizontal até que o caracter desejado faceie a fita. A figura 4 nos mostra uma esfera com caracteres embutidos.

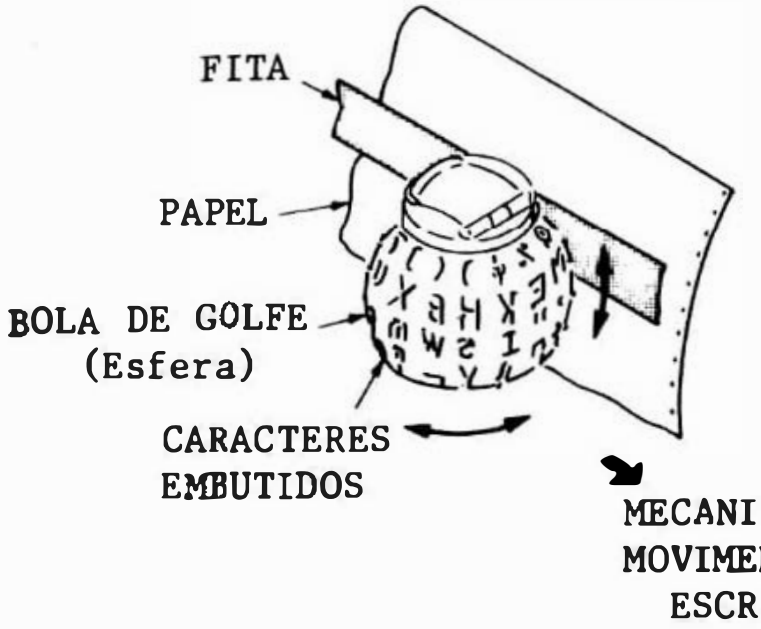


FIGURA 04

Um outro tipo de impressora de caracteres completamente formados, projetada para o uso do computador, possui os caracteres embutidos na periferia de um cabeçote de impressão "es-

pinhoso". Existem duas versões deste método de impressão: as impressoras de "roda dentada" ou "margarida" (veja figura 5) e as impressoras que utilizam "dedais". Ambas operam de forma semelhante.

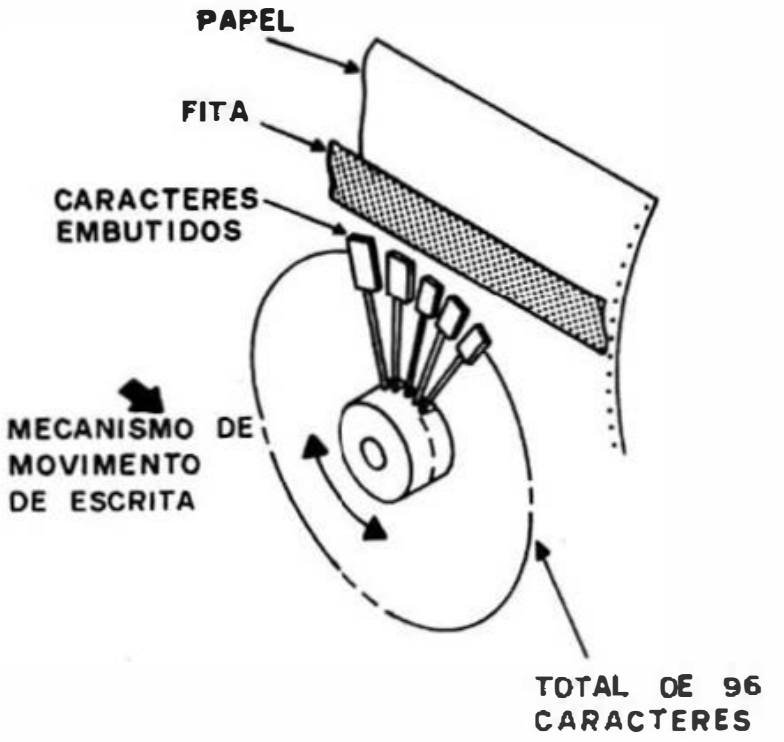


FIGURA 05

Um cabeçote de impressão do tipo "roda dentada" é montado em um disco rotativo, como pétalas de flores, semelhante a uma margarida. Cada pétala contém o caracter "embutido" em inverso. Um martelo bate na pétala flexível que contém o caracter desejado, pressionando a fita com tinta contra o papel, imprimindo-se assim o caracter no mesmo.

Já um cabeçote de impressão do tipo "dedal" possui as mesmas características de um cabeçote do tipo margarida, porém as pétalas flexíveis giram para cima e possuem um formato semelhante ao de um dedal. Neste caso, os caracteres são embutidos em braços flexíveis ao redor do "dedal". Conforme o "dedal" se move horizontalmente sobre o papel, um martelo bate no braço flexível do caracter apropriado (desejado), que vai girando até atingir a posição correta.

Ambos os tipos de impressoras de "roda dentada" possuem uma única característica que as torna úteis no emprego em sistema de computadores: a alta velocidade de impressão (cerca

de 50 caracteres por segundo). Esta característica, aliada ao fato de os caracteres serem completamente formados, torna este tipo de impressora muito útil em aplicações de processamento de palavras.

Além disto, a cabeça de impressão pode ser trocada, permitindo assim, fontes de símbolos proporcionais, fontes de leitura de caracteres ópticos, fontes de multilínguas e fontes de aplicação especial. Por outro lado, as versões deste tipo de impressora, que utilizam grandes carros, podem operar com papéis de até 16 polegadas de largura. Alguns modelos mais sofisticados possuem duas rodas de impressão que imprimem simultaneamente na mesma linha.

Estes tipos de impressoras são capazes de imprimir bidirecionalmente, ou seja, a segunda linha da impressão é armazenada em uma memória BUFFER, dentro do circuito de controle da impressora e pode ser impressa em ambas as direções, optando pela que tomar menos tempo de impressão. Os circuitos de controle da impressora avaliam cada linha impressa e selecionam qualquer direção de impressão que leve o menor tempo. Com relação ao espaçamento entre linhas, o papel é espaçado verticalmente (alimentar a linha) para imprimir seis (6) linhas por polegada. Sob o controle do programa, o espaçamento de alimentar linhas é controlável até incrementos de 0,5 mm. Este fato habilita estas impressoras para o traçamento de alguns gráficos.

b) Impressoras de Impacto de uma linha por vez para caracteres completamente formados:

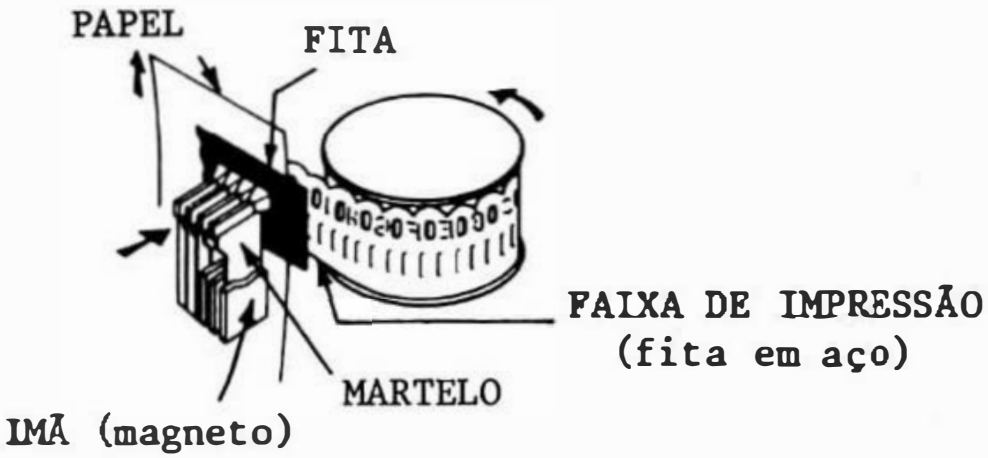
Estas impressoras são chamadas de "impressoras de linha". Os caracteres ou espaços que constituem uma linha (tipicamente 132 posições de caracteres) são impressos simultaneamente sobre uma linha inteira. O papel é espaçado para cima e uma nova linha é então impressa. A velocidade de impressão para estas impressoras varia de algumas centenas a milhares de linhas por minuto. Devido ao alto custo do equipamento e também à alta velocidade de impressão, estas impressoras são utilizadas para grandes volumes de impressão.

Nestas impressoras, uma fonte de símbolos embutidos é posicionada sobre uma linha para imprimir, utilizando os símbo-

los embutidos sobre um carro, consistindo de uma corrente, de um trem, ou ainda de uma faixa, que se move horizontalmente sobre o papel. Também pode ser utilizado um tambor rotativo, colocado de frente para o papel e com os caracteres embutidos tipicamente em 132 colunas do tambor. Conforme o tambor gira, as colunas de caracteres passam verticalmente sobre o papel e a linha de impressão. Em ambos os métodos, os martelos (um para cada uma das 132 posições de impressão) batem quando o caracter correto está posicionado, imprimindo o caracter no papel, através de uma fita com tinta.

A figura 6 nos mostra a faixa de impressão para uma impressora de impacto de uma linha por vez.

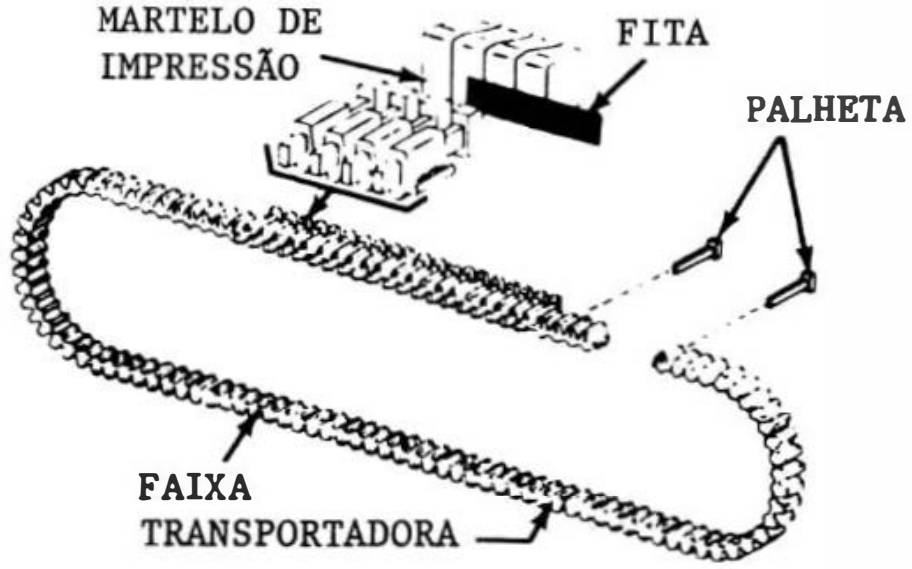
FIGURA 06



Note que os caracteres de impressão são embutidos na faixa. A faixa envolve dois cabrestantes, passando de frente para o papel. Uma fita com tinta é posicionada entre a faixa em movimento e o papel. Conforme os caracteres vão sendo impressos, a faixa vai se movendo pelas 132 posições de impressão horizontal. Os 132 martelos de impressão, atrás do papel, batem na faixa, no tempo apropriado, causando a linha de caracteres impressos. Os circuitos de sincronismo, no controlador da impressora, controlam o tempo de posicionamento da letra ou do número desejado, com relação ao tempo de disparar o martelo.

Uma impressora de cadeia opera no mesmo princípio. A figura 7 nos mostra um esquema de uma impressora de cadeia. Nesta técnica, uma cadeia gira na frente da fita e do papel. Cada argola da corrente é projetada para segurar uma palheta na qual os símbolos dos caracteres estão embutidos. Os martelos estão localizados atrás do papel e cada um dos 132 martelos bate na palheta de símbolos em movimento, quando o caracter desejado passa na posição na qual o martelo está sincronizado para bater.

FIGURA 07



Mais que uma colocação de caracteres é gravada em uma corrente ou faixa e quanto menor forem os caracteres, mais linhas por minuto são impressas. Por esta razão, muitas impressoras de linhas usam somente a colocação de 64 caracteres ASCII. Algumas impressoras utilizam menos, fornecendo somente os 26 caracteres do alfabeto, 10 numerais e alguns outros símbolos. A maioria das impressoras apresenta a remoção fácil do carro, sendo possível, portanto, mudar as colocações de caracteres e conseqüentemente ajustar a eficiência da impressão para melhor adequar as necessidades imediatas da aplicação da impressora.

Na figura 8, mostramos uma impressora de tambor. Cada um dos 64 ou 96 caracteres utilizados é gravado em 132 colunas ao redor do tambor, correspondendo a 132 posições de impressão. O

tambor gira na frente do papel e da fita; os martelos de impressão batem no papel, imprimindo o caracter do tambor, através da fita e formando assim a impressão do caractere.

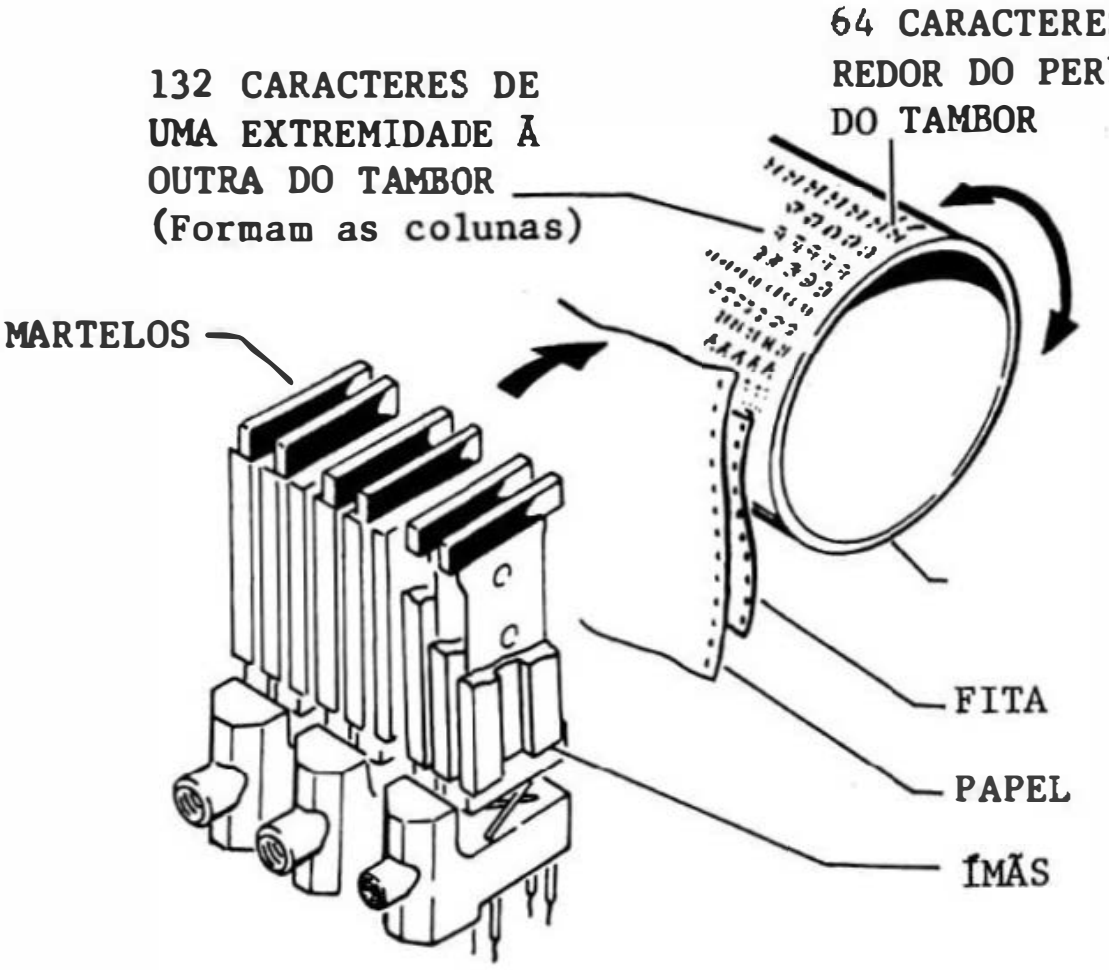


FIGURA 08

c) Impressoras sem Impacto de uma linha por vez para caracteres completamente formados

São impressoras para fins especiais, por exemplo, para "compor" textos. Também pertencem a este grupo, as impressoras da linha de velocidade extremamente alta. Este tipo de impressora utiliza tecnologias fotográficas, xerográficas e laser para formar os caracteres completos sem "tocar" a superfície de impressão. Devido a sua função especial e ao seu alto custo, este tipo de impressora é pouco utilizado como periférico de saída para mini e microcomputadores, porém, muitas vezes, estas impressoras utilizam mini e microcomputadores como dispositivo de controle.

As impressoras "LASER" combinam as tecnologias de impressão LASER e XEROGRÁFICA para imprimir em velocidades extremamente altas, por exemplo velocidades na faixa de 18000 a 21000 linhas por minuto, o que corresponde a uma média de 300 a 350 linhas por segundo.

A corrente de LASER, controlada pelo computador, coloca o dispositivo de impressão sobre um tambor rotativo carregado eletricamente. Partículas de energia aderem-se às secções carregadas e então são transferidas para o papel, para formarem a página impressa.

Os mini e microcomputadores são também usados para controlarem a "fotocomposição". Nestes sistemas, as imagens do caracter, com um tamanho específico e a fonte são selecionados pelo computador e opticamente projetados sobre um papel fotossensível (ou filme), onde as linhas da cópia são gravadas por processo fotográfico. O papel (ou filme) exposto é revelado e a cópia fotocomposta é, muitas vezes, usada para fazer placas de impressão, para imprimir livros, revistas, ou jornais. A figura 9 nos mostra o esquema de um fotocompositor de tubo de raios catódicos (TRC), controlado por meio de um mini computador e, atualmente, opera como uma impressora de linha sem impacto de fim especial.

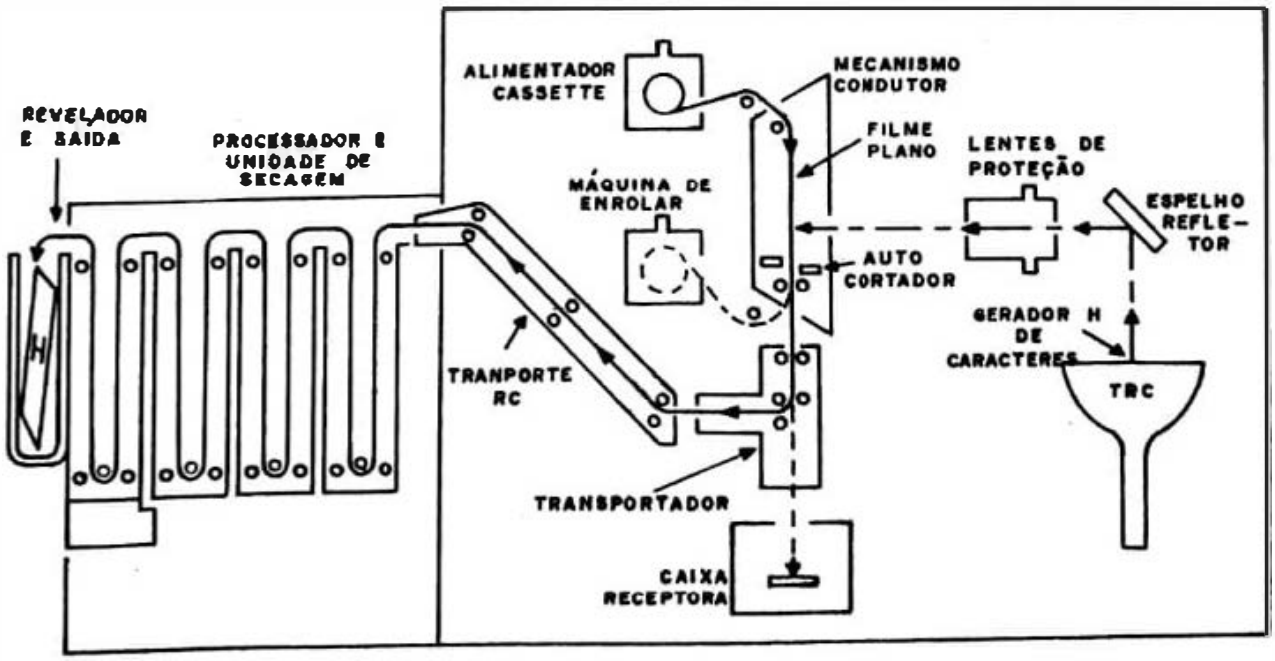


FIGURA 09

Para a impressora mostrada na figura 9, a projeção da fonte de caracter é executada por um tubo de raios catódicos (TRC), com os geradores de caracteres armazenados na memória, ou por dissimulação do caracter, processo este utilizado para projetar opticamente os caracteres selecionados.

d) Impressoras por Matriz de Pontos:

Os caracteres por matriz de pontos são formados pela impressão de um grupo de pontos, os quais formam uma letra, um número, ou um outro símbolo. Este tipo de impressora é largamente utilizada em sistemas de mini e microcomputadores.

Os pontos podem ser formados pelos métodos de impressão de impacto e sem impacto. Em ambos os casos, as impressoras podem ser de um caracter por vez ou de uma linha por vez. Para imprimir os pontos, os métodos mais utilizados são: de impacto, térmico, eletro-sensitivo e jatos de tinta.

A figura 10 nos mostra a letra "A", formada por uma matriz de cinco pontos de largura e sete pontos de altura (5x7) e, também, em uma matriz 9x7 onde os pontos são espaçados horizontalmente em 9 incrementos de meio passo.

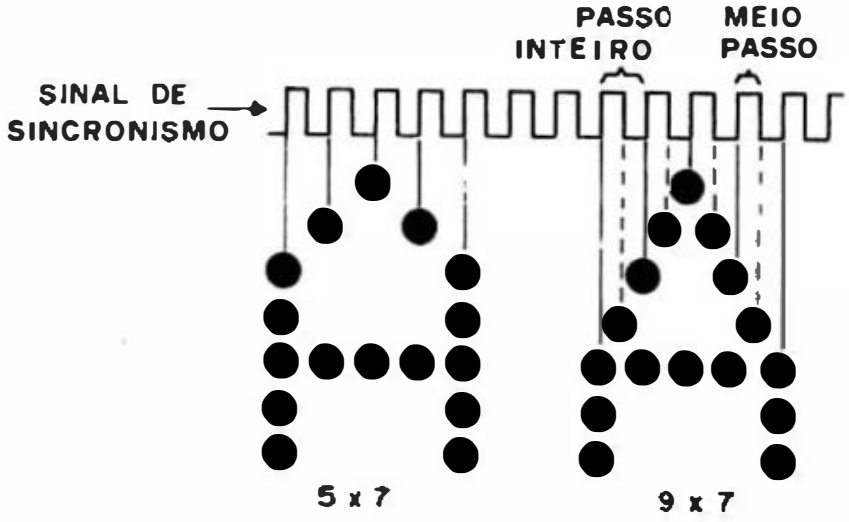


FIGURA 10

Outras configurações da matriz de representação são 7x7, 8x7, 7x9, 9x9 e 9x12 (essas são as mais comuns). A figura 11 mostra uma colocação de 63 caracteres completos, formados por uma matriz 7x7.

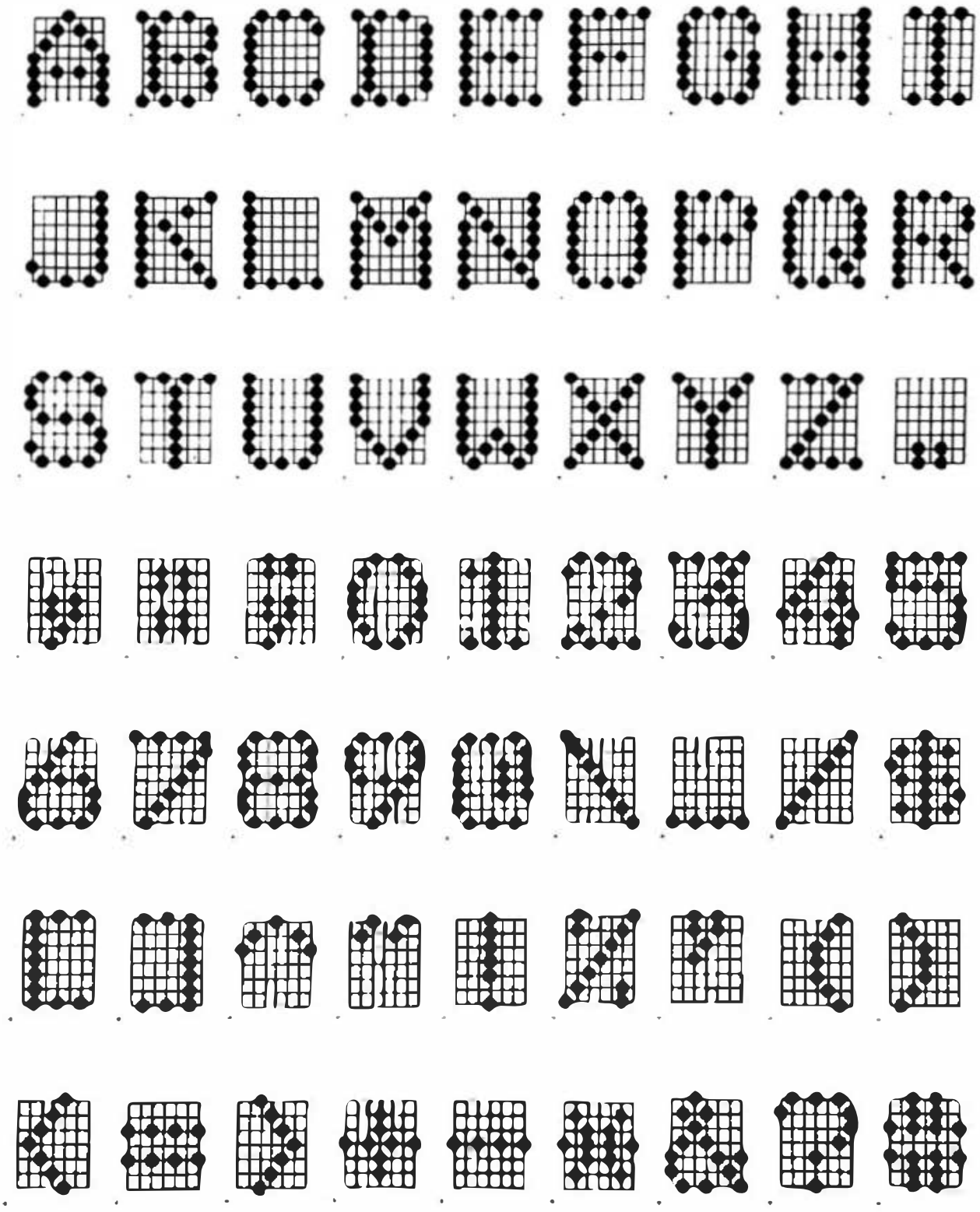


FIGURA 11

A figura 12, por sua vez, nos mostra várias formas de caracteres, expandidos horizontalmente, comprimidos e normais, todos demonstrando a versatilidade das impressoras por matriz de pontos.

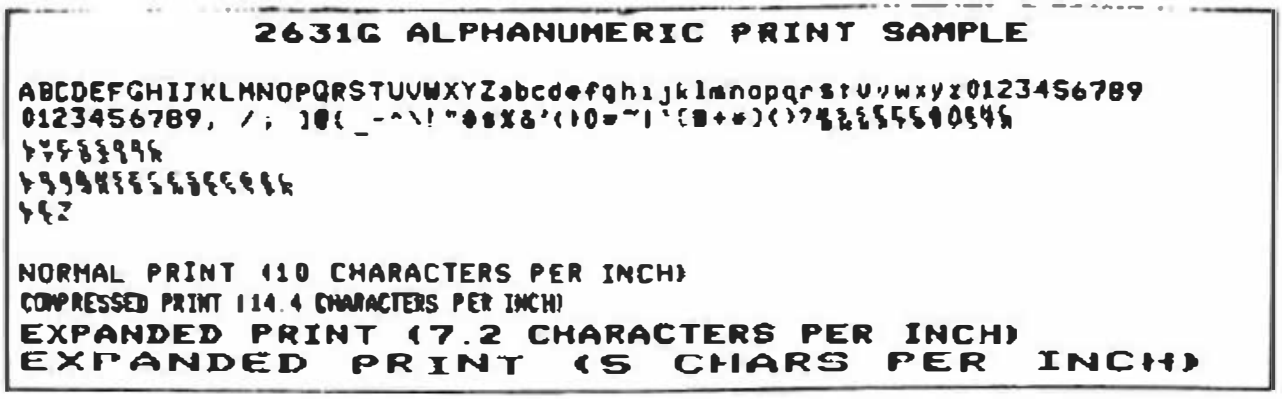


FIGURA 12

Uma matriz de pontos 5x7 é normalmente utilizada quando as letras maiúsculas são aceitáveis (esta não pode manejar letras minúsculas regulares, pois não há espaço suficiente, ficando parte das letras abaixo da linha). Já uma matriz 7x9 é utilizada tanto para se imprimir caracteres maiúsculos como minúsculos, podendo manejar "restos" (partes) de letras, abaixo da linha.

As impressoras de matriz de pontos podem imprimir qualquer combinação de pontos com todas as posições de impressão, válidas para a matriz. O caracter impresso, quando umdos 128 códigos ASCII é sinalizado, é controlado por meio de uma memória ROM, a qual controla a espécie de pontos. Para se mudar a linguagem ou para a colocação de caracteres puramente gráficos, basta trocar a memória ROM.

Nas impressoras onde os pontos horizontais e verticais se sobrepõem, os blocos de quatro ou mais pontos podem ser impressos para formar blocos de caracteres e formas maiores. A figura 13 ilustra vários tipos de colocações de caracteres de língua estrangeira, produzidos por uma impressora por matriz de pontos.

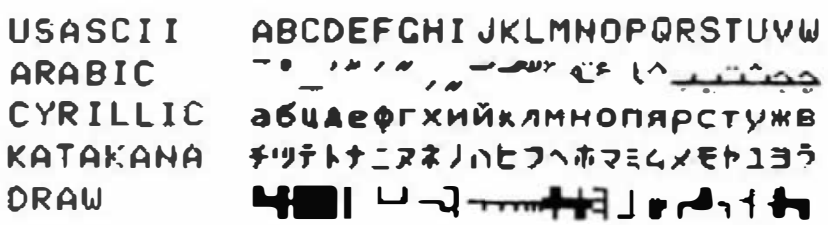


FIGURA 13

e) Impressoras por Matriz de Pontos de Impacto de um caracter por vez:

O cabeçote de impressão para este tipo de impressora, normalmente, é composto por um conjunto de fios rígidos que batem o caracter através de uma fita com tinta (veja figura 14). Por esta razão, estas impressoras são muitas vezes referidas por "impressoras de matriz-de-fios".

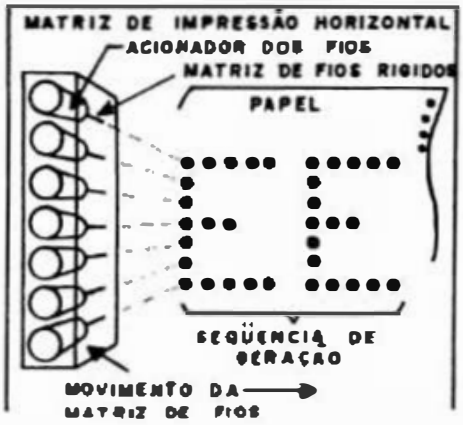


FIGURA 14

O cabeçote de impressão, geralmente, contém uma única coluna de sete (7) agulhas, muito embora possa ter duas ou mais colunas de agulhas. A figura 15 ilustra um cabeçote de impressão de duas (2) colunas, contendo sete (7) agulhas cada uma.

FIGURA 15

Para fins de ilustração e de didática, vamos supor que o cabeçote de impressão contenha uma única coluna de agulhas. As sete agulhas são empurradas eletromagneticamente em qualquer combinação que o controlador de impressão determinar. Para um caracter de matriz de pontos 5x7, o cabeçote de impressão, após imprimir a primeira coluna, dá um passo e, em seguida, imprime a segunda coluna, repetindo este processo até que todas as cinco colunas sejam impressas.

Virtualmente, todas as impressoras por matriz de pontos utilizam uma ou duas colunas verticais de agulhas. A complexidade mecânica do cabeçote de impressão e do circuito de controle aumenta com o número de colunas de múltiplas agulhas.

A impressora de caracteres por matriz de pontos não imprime um caracter por vez, mas sim uma coluna de um caracter por vez. Apesar deste fato, a velocidade de impressão destas máquinas às vezes é tão alta que elas chegam a se equiparar com as impressoras de um caracter por vez, com caracteres completamente formados.

Além da alta velocidade de impressão, a maioria das impressoras por matriz de pontos pode imprimir em duas direções.

A principal desvantagem deste tipo de impressora reside no fato de que os caracteres impressos não possuem a aparência de caracteres completamente formados e são, portanto, menos úteis para aplicações de processamento da palavra. Porém, para outros usos em computadores, elas se mostram altamente eficientes.

Hoje em dia, já é comum encontrarmos impressoras deste tipo, imprimindo bidirecionalmente, com o cabeçote de impressão movendo-se horizontalmente. A figura 16 nos mostra um carro de impressão com duplo cabeçote de impressão.

f) Impressoras com Matriz de Pontos de Impacto de uma linha por vez:

Este tipo de impressora utiliza um pente de impressão horizontal (veja figura 17), o qual se estende sobre a largura do papel. Este pente possui 132 martelos e um ponto elevado (saliente) está sobre cada um desses martelos.

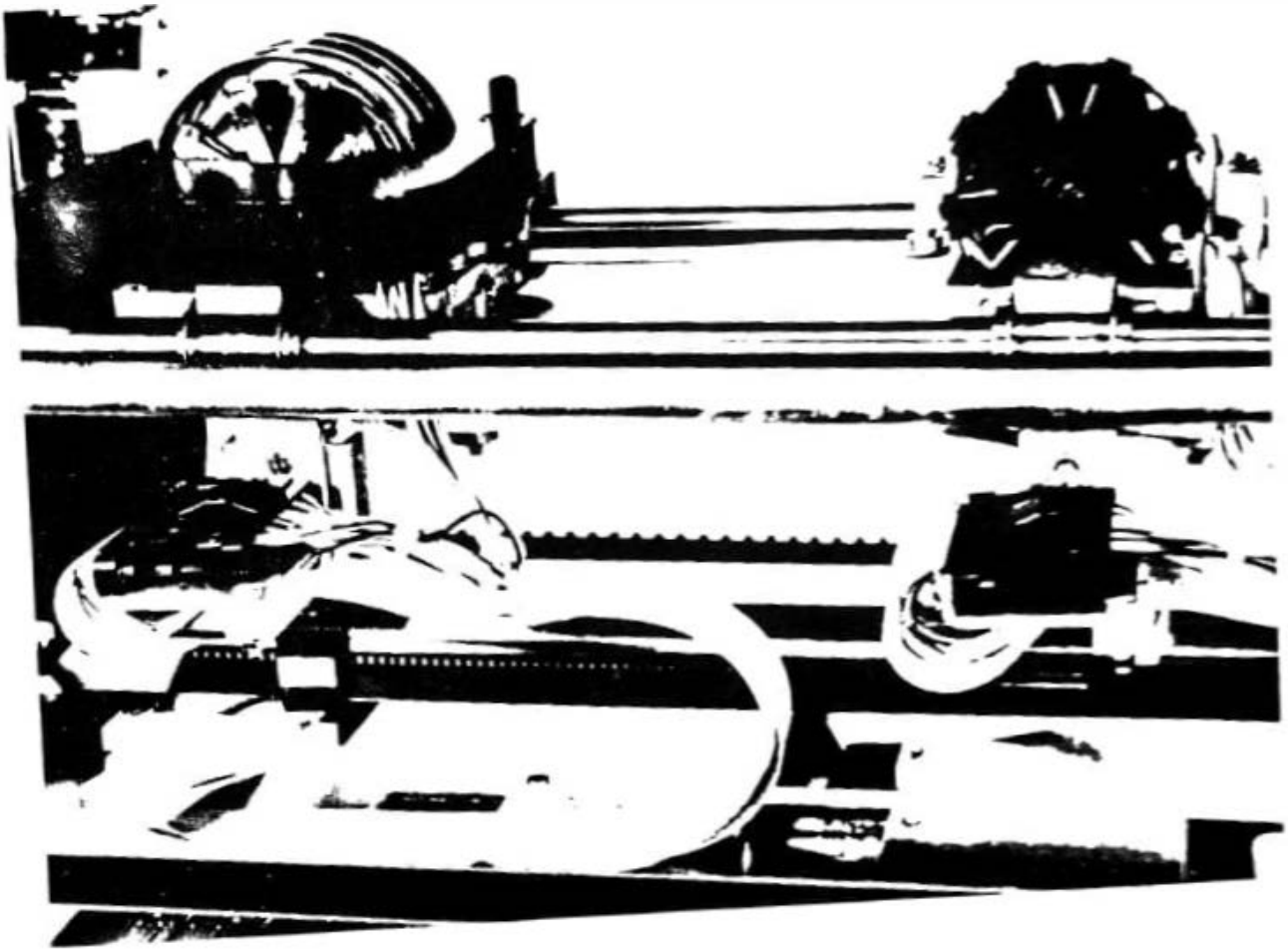


FIGURA 16

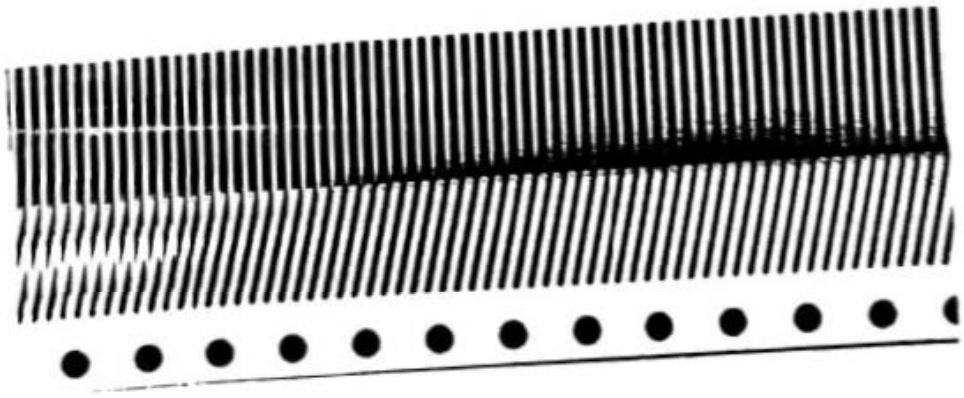
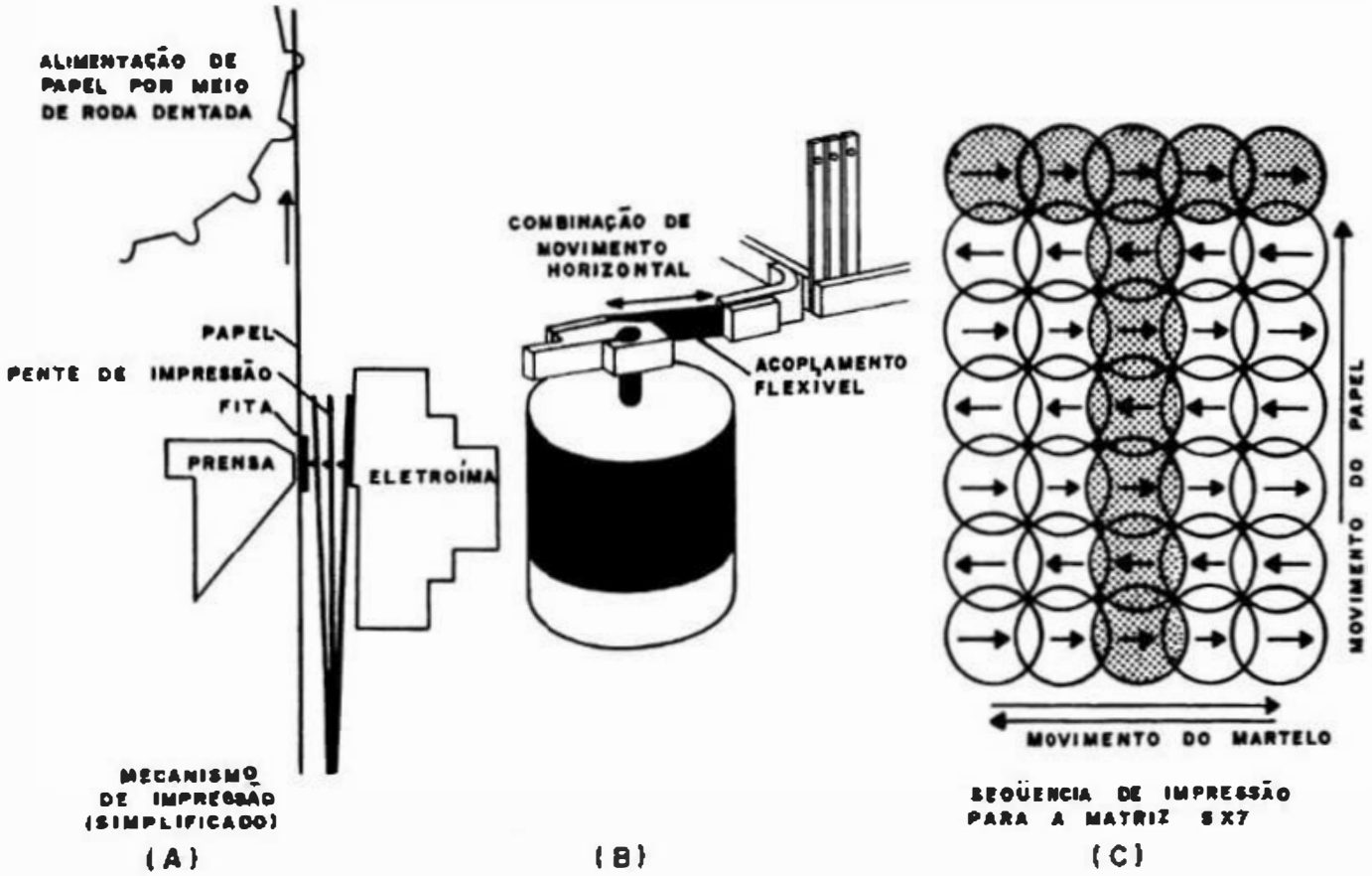


FIGURA 17

Os martelos são atirados de volta por um processo eletromagnético o qual está sob o controle dos geradores de caracteres e, em seguida, são liberados (soltos) para baterem (em direção reta, na fita e conseqüentemente no papel). Na figura 18, mostramos um esquema deste processo de impressão.

FIGURA 18



Cada martelo corresponde a uma posição de impressão do caracter e o pente de 132 martelos é, portanto, do tamanho dos 132 espaços de impressão. O pente inteiro oscila horizontalmente para imprimir. Esta impressora imprime uma linha de pontos para todos os 132 caracteres e, em seguida, imprime a linha subseqüente e assim por diante, até que os sete (7) passos tenham sido completados, de modo a obtermos uma matriz de pontos sobrepostos 5x7, com 132 caracteres.

A velocidade típica de impressão da linha é algo em torno de 125 linhas por minuto. Cabe ressaltar, ainda, que o movimento do papel se dá na vertical.

g) Impressoras por Matriz de Ponto sem Impacto:

Essas impressoras fazem uma marca no papel, sem tocar o mesmo de forma direta.

Este tipo de impressora não pode fazer cópias com carbono, pois não há força para pressionar o caracter. Mesmo assim, este tipo de impressora se mostra útil, na prática, para imprimir simples cópias da saída do computador, para gravar a saída de calculadoras de impressão e displays, etc.

Para a construção deste tipo de impressora, dispõe-se de pelo menos cinco tecnologias:

1 - Térmica;
2 - Eletro-sensitiva;
3 - Eletrostática;
4 - Jato de tinta;
5 - Laser, xerografia e fotografia (já vistas anteriormente).

Vejamos então algumas informações sobre as quatro primeiras tecnologias:

1) Impressoras Térmicas:

A impressora térmica aquece, eletricamente, uma matriz de pontos, na espécie de caracter desejada. O papel para estas impressoras é recoberto por uma fina camada de tinta sensível ao calor. A matriz de ponto, quando aquecida, provoca o aparecimento de pontos pretos ou azuis (a cor irá depender do tipo de papel) no papel, os quais irão formar os caracteres desejados.

Os cabeçotes térmicos de impressão movem-se horizontalmente sobre o papel, imprimindo cada coluna de caracteres em série. Alguns cabeçotes térmicos de impressão são estacionários, imprimindo uma linha horizontal de pontos em um certo tempo. A figura 19 nos mostra um cabeçote de impressão em série para imprimir uma matriz de sete pontos de altura. Os elementos resistivos que provocam o aquecimento se encontram no canto direito superior do cabeçote de impressão. A potência é fornecida ao circuito, para cada ponto, resultando no aquecimento necessário para uma determinada posição causar uma mudança de cor no papel.

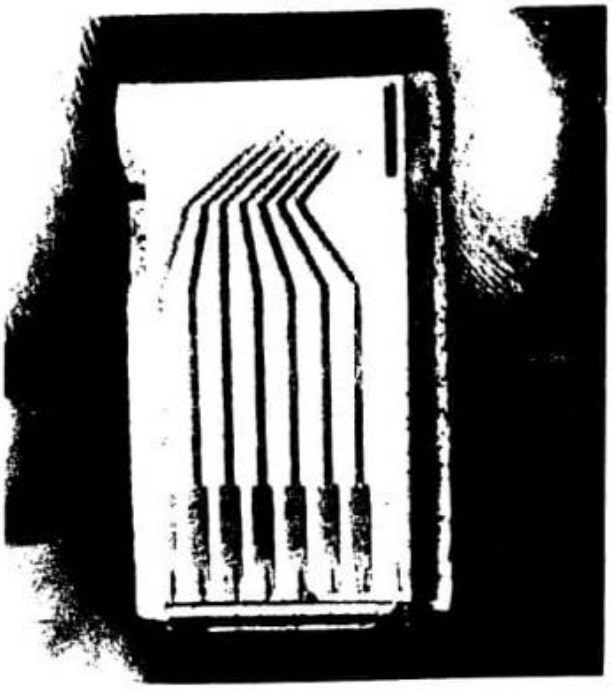


FIGURA 19

Algumas impressoras térmicas de uma linha por vez utilizam almofadas de impressão fixas, contendo uma linha de elementos de aquecimento resistivo sobre um substrato de cerâmica.

Uma das principais aplicações deste tipo de impressora está nas calculadoras eletrônicas.

2) Impressoras Eletro-sensitivas:

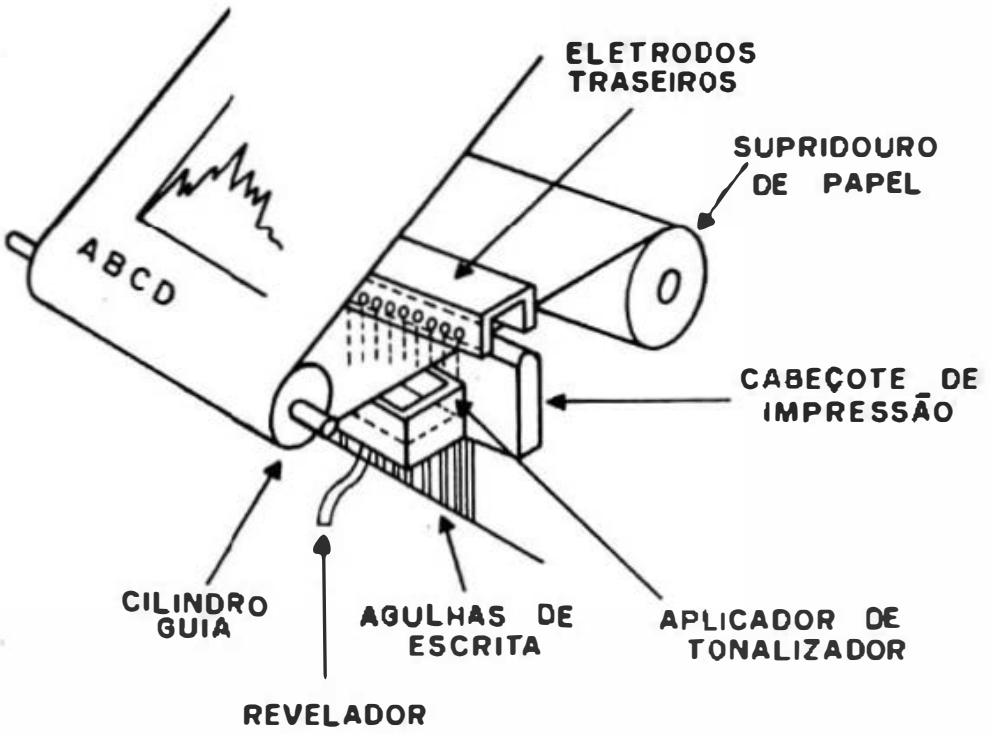
A corrente elétrica passa através da ponta de uma agulha ou da extremidade de um cabeçote de impressão para um papel especial, sensível à mudança de tensão. Em muitos casos, o revestimento do papel é de alumínio, o qual vaporiza quando uma tensão é aplicada ao cabeçote de impressão, deixando uma marca preta no papel. Este tipo de papel é diferente do papel térmico, pois não é afetado pelo calor nem pela luz ou água, mantendo, desta forma, uma impressão relativamente constante. Este tipo de impressora é muito utilizado para gravar a exibição de um monitor de vídeo.

3) Impressoras Eletrostáticas:

Nestas impressoras, uma tensão é aplicada a uma determinada disposição de agulhas (do cabeçote de impressão), para

formar os caracteres desejados, com base em uma matriz de pontos. A figura 20 nos mostra o esquema de uma impressora eletrostática.

FIGURA 20



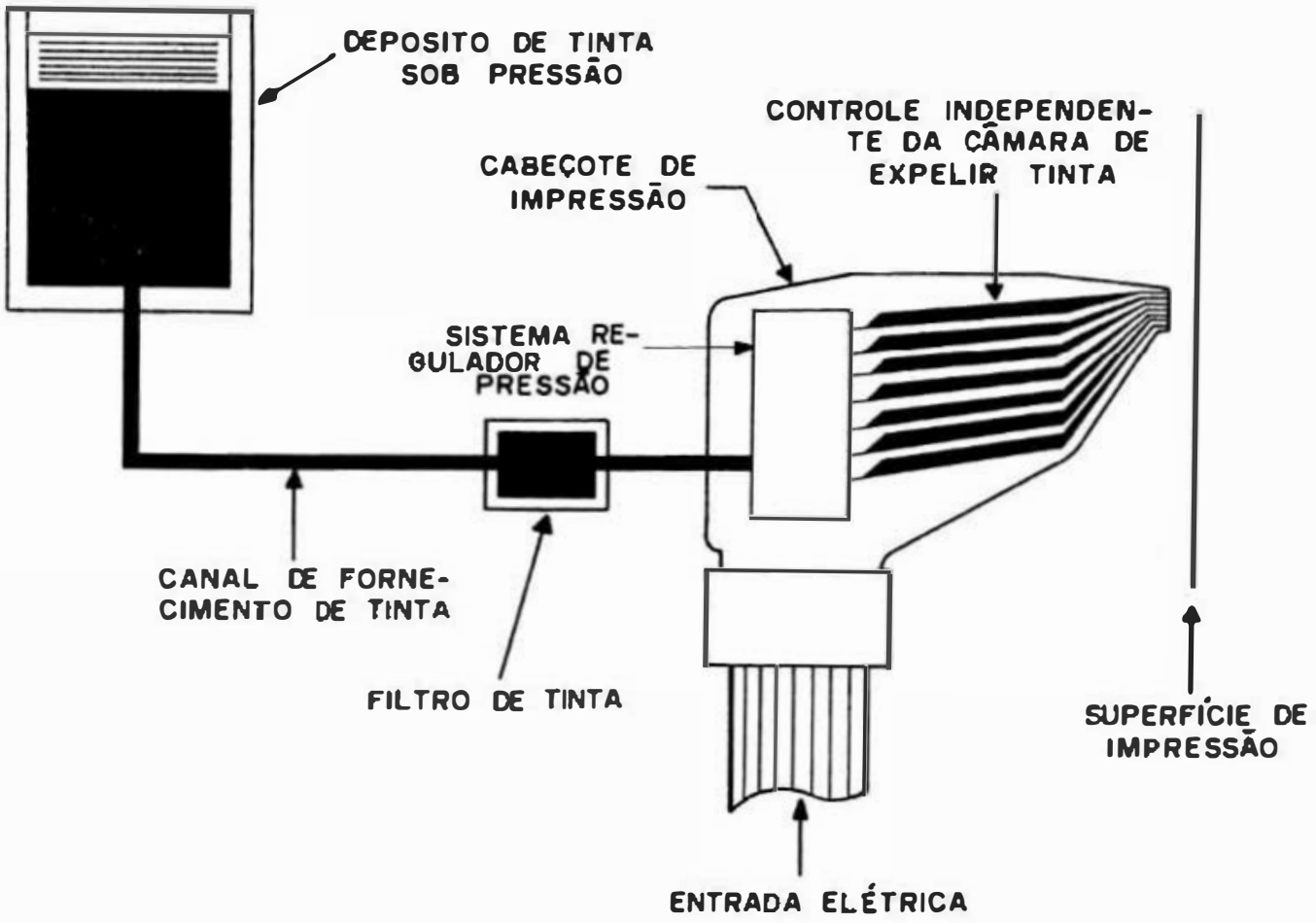
As agulhas energizadas formam pontos "carregados eletricamente" em um papel dielétrico, à medida que o papel se desenrola e passa pelas agulhas. Em seguida, faz-se com que o papel passe através de um energizador e, neste instante, os pontos "carregados eletricamente" passam a atrair certas partículas especiais, as quais irão dar origem aos caracteres da matriz de pontos.

4) Impressoras de Jato de Tinta:

Neste tipo de impressora, gotinhas de tinta são expelidas diretamente sobre o papel, para a formação dos caracteres por matriz de pontos. Existem duas tecnologias principais, empregadas na fabricação destas impressoras: a tecnologia "DROP ON DEMAND" e a de "FLUXO CONTÍNUO".

No sistema DROP ON DEMAND, gotas de tintas sob pressão são pulverizadas no papel, através de bicos dirigidos. Cada bico é aberto por um pulso elétrico. Uma disposição vertical de bicos de uma matriz expele gotas de tinta, formando, assim, uma coluna da matriz de pontos. O bico é então levado, horizontalmente, para a segunda coluna da matriz e, em seguida, a tinta dos jatos selecionados é expelida. E assim por diante, até que um caracter da matriz de pontos seja formado. Um esquema, contendo alguns detalhes deste sistema, é mostrado na figura 21.

FIGURA 21



Já as impressoras de jato de tinta de "fluxo contínuo" bombeiam a tinta continuamente através de um bico. O fluxo de tinta passa através de um tubo, onde as gotas de tintas são eletricamente carregadas. A condução das gotas de tinta para o papel é eletronicamente direcionada pelas placas de deflexão horizontais e verticais. A carga de tinta que é dirigida pelas placas de deflexão é controlada por um gerador de carac

ter eletrônico. Este método de direcionar o fluxo de tinta é semelhante ao de um monitor de vídeo. Quando um caracter não é impresso, o fluxo de tinta continua e é capturado e levado para um reservatório, a partir do qual a tinta é reciclada.

A figura 22 nos mostra um esquema elementar do sistema de impressão, com fluxo contínuo de tinta.

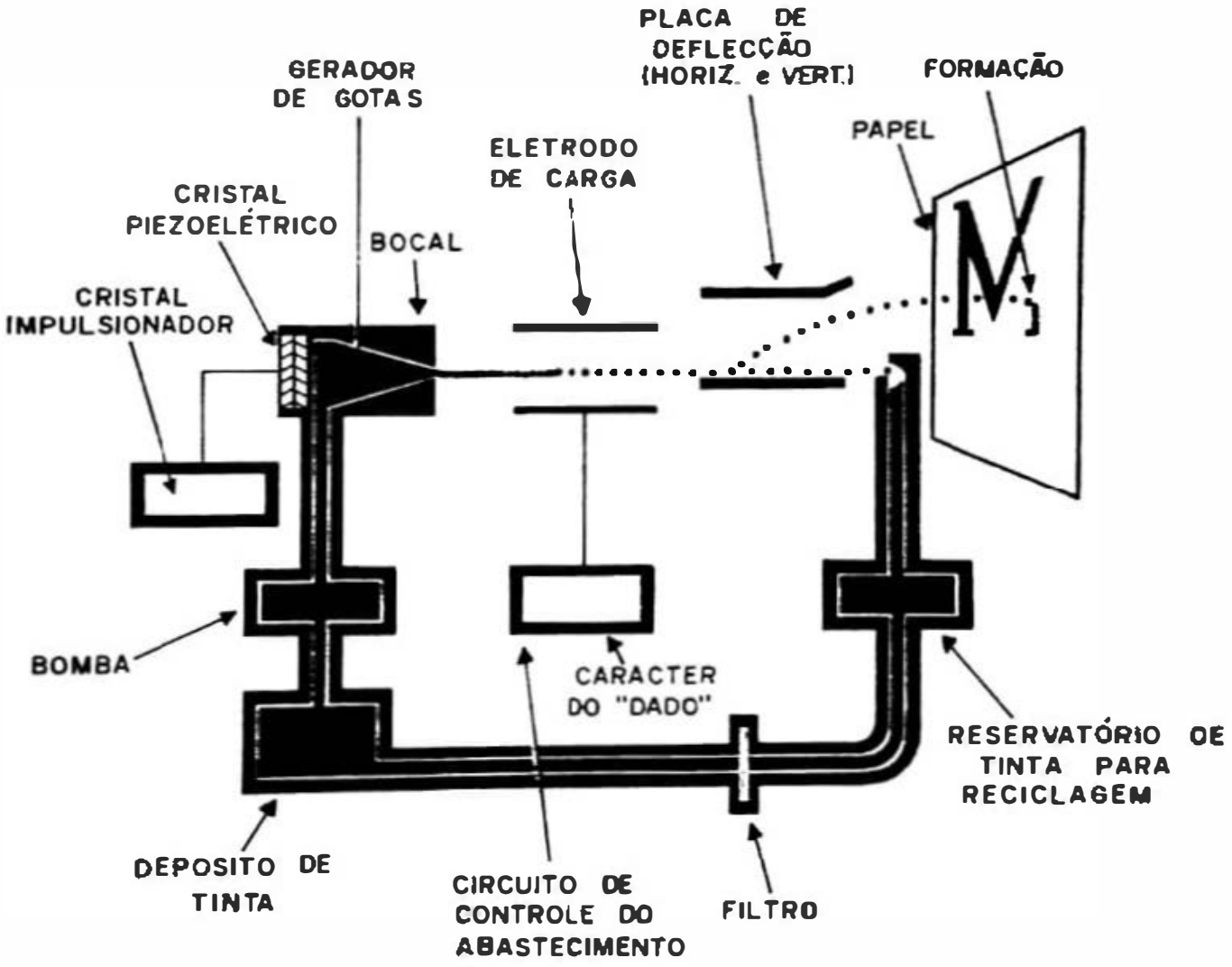


FIGURA 22

A impressão obtida através deste método (jato de tinta de fluxo contínuo) melhora a qualidade do caracter impresso de modo que a impressão, em termos de qualidade, fica próxima à das impressoras de caracter completamente formado.

Uma das vantagens apresentadas pelos sistemas de impressão por jato de tinta é que este não requer papel especial pa

ra a impressão.

Para finalizar este breve estudo sobre as impressoras, iremos analisar um detalhe comum a todas elas, que é a colocação ou manejo do papel.

Colocação ou Manejo do papel:

Todas as impressoras possuem a necessidade de mover o papel para que possam imprimir. Para tanto, existem três métodos: manter a fricção, alimentar o rolo com pinos e alimentar a tração do pino. Vejamos então em que consiste cada um desses métodos.

- Manter a fricção:

Este é o método utilizado pela máquina de escrever comum. O papel é mantido pela fricção contra o rolo de impressão e se move através da impressora, conforme o rolo vai girando. Este método é utilizado em impressoras de velocidades relativamente baixas.

- Alimentar o rolo com pinos:

Neste caso, o papel a ser utilizado pela impressora deverá ter buracos perfurados nas bordas, de uma forma contínua, ou seja, em toda a extensão das bordas do papel. Os pinos montados no rolo engatam nos buracos e movem o papel através da impressora. Este método é muito empregado em impressoras de velocidade média, pois facilita o alinhamento do papel na impressora.

Ainda com relação a este método cabe ressaltar que devido ao fato dos pinos serem montados no rolo e não serem ajustáveis horizontalmente, só é possível a utilização de um determinado papel de largura padrão.

- Alimentação por tração dos pinos:

Este método utiliza dois circuitos de tração, operados

por motor (veja figura 23), montados em uma única barra, sendo possível de ajustamento horizontal. Cada circuito de tração possui uma colocação de pinos. O movimento do circuito de tração (trator) está sob o controle do programa, para pulsos de alta velocidade.

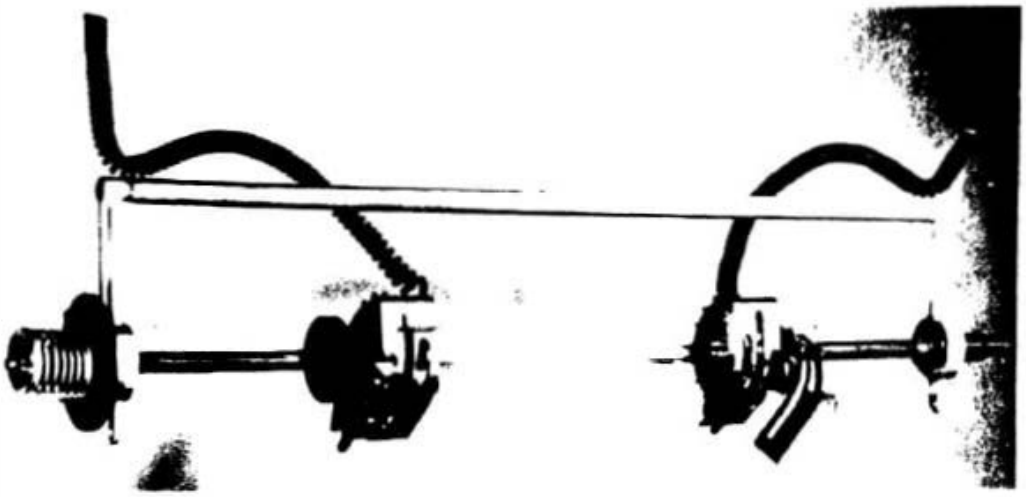


FIGURA 23

As impressoras de alta velocidade de impressão, normalmente, incorporam a alimentação por tração de pinos.

É evidente que as "formas" de folhas simples podem ser alimentadas manualmente. Como uma alternativa, a opção de introdução das folhas simples é válida para muitas impressoras, utilizadas no processamento de palavras, principalmente para as que preparam a letra individualmente para em seguida bater (imprimir).

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES
CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-19

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) As impressoras que formam o caracter desejado "batendo" no papel são conhecidas por impressoras:

   a) de não impacto;

   b) de impacto;

   c) xerográficas;

   d) N.R.A.

2) Quando o caracter é formado por um conjunto de pontos, podemos afirmar que este foi produzido por uma impressora:

   a) de impacto;

   b) de matriz de linhas;

   c) de matriz de pontos;

   d) N.R.A.

3) As impressoras de uma linha por vez também são conhecidas como:

a) impressoras de linha;

b) impressoras de folha;

c) impressoras de pontos;

d) N.R.A.

4) Uma impressora de impacto, com tecnologia matriz de pontos ou arame, com capacidade de um caracter por vez, possui uma velocidade aproximada:

a) entre 30 e 50 l/min;

b) entre 100 e 200 l/min;

c) entre 10 e 15 c.p.s.;

d) entre 30 e 330 c.p.s.

5) Uma impressora sem impacto, com tecnologia matriz de pontos eletrostáticos, com capacidade de uma linha por vez, possui uma velocidade aproximada:

a) entre 50 e 150 c.p.s.;

b) entre 300 a 1800 l/min;

c) de 3000 l/min;

d) N.R.A.

6) Um exemplo clássico de impressora de impacto é:

a) a máquina de escrever;

b) o tubo de raios catódicos;

c) as impressoras à laser;

d) N.R.A.

7) As impressoras sem impacto, de uma linha por vez, são impressoras destinadas a operar:

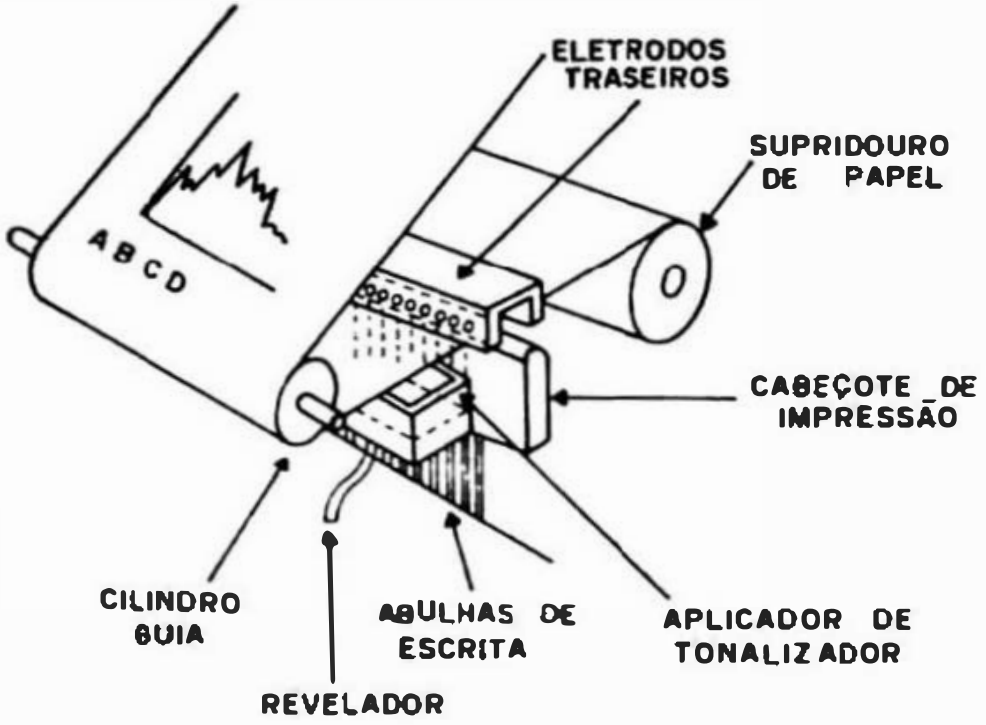
a) em velocidades extremamente altas;

b) em velocidades extremamente baixas;

c) sem tocar a superfície de impressão;

d) os itens a e c estão corretos.

8) A figura a seguir nos mostra o esquema de uma impressora:



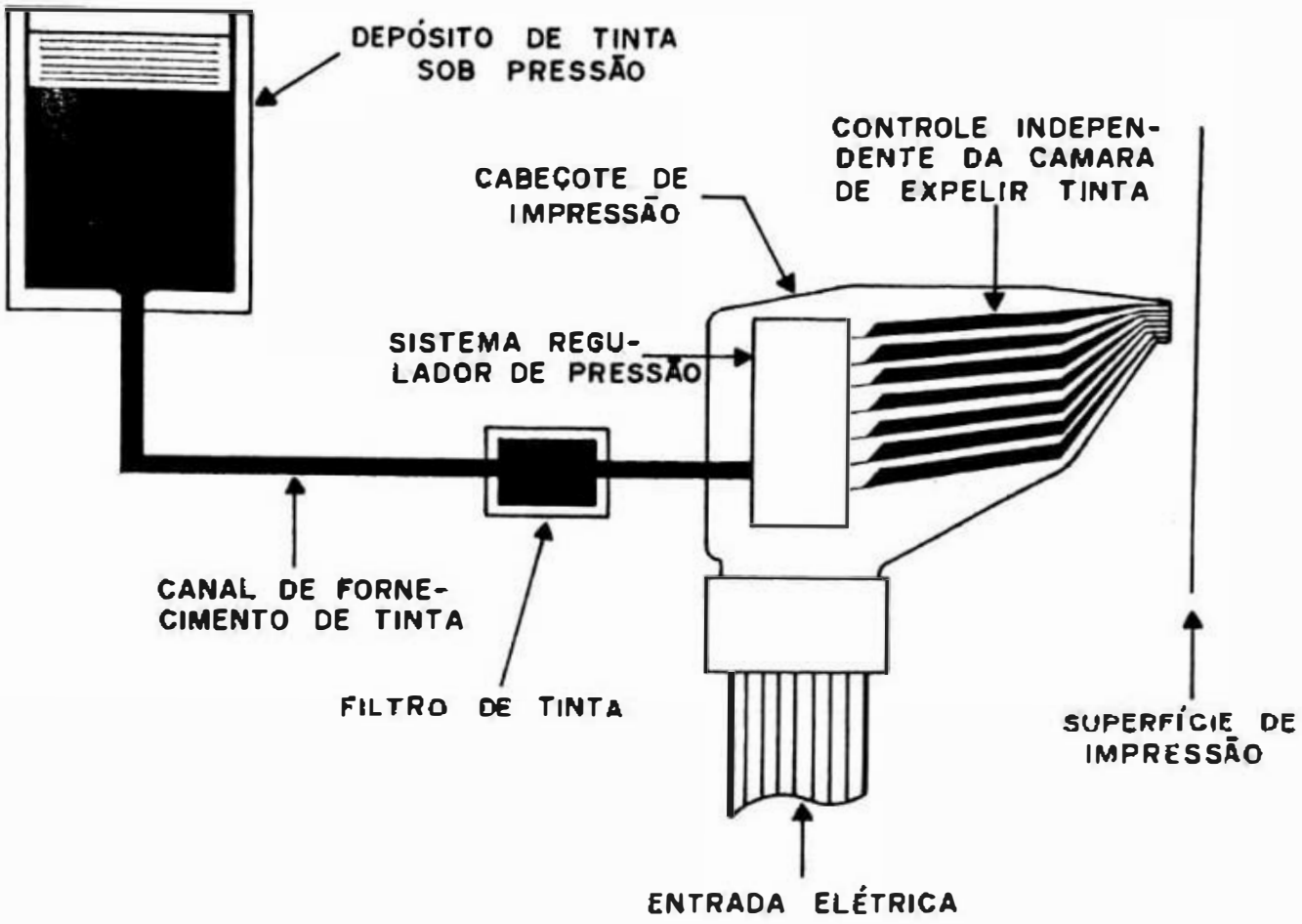
a) de jato de tinta;

b) eletrostática;

c) tipo margarida;

d) térmica.

9) O esquema mostrado a seguir é de uma impressora do tipo:



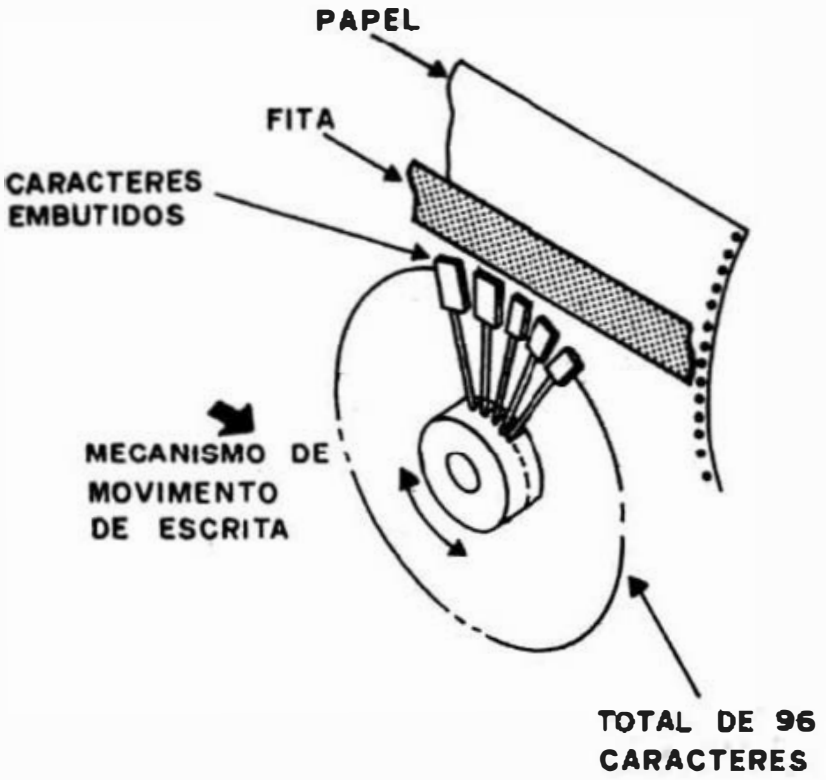
a) margarida;

b) eletrostática;

c) de jato de tinta que utiliza o sistema de FLUXO CONTÍNUO;

d) de jato de tinta que utiliza o sistema DROP ON DEMAND.

10) O cabeçote de impressão da figura a seguir é do tipo:



a) jato de tinta;

b) laser;

c) margarida;

d) xerográfico.

CEBM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP86100 - LONDRINA - PARANÁ

LIÇÃO MC-20

TERMINAL DE VIDEO I

INTRODUÇÃO

Os terminais de vídeo são tidos, hoje em dia, como um dos mais importantes dispositivos de entrada e saída de dados dos microcomputadores,para o usuário. Esses terminais permitem, por exemplo, que os diversos setores de uma empresa agilizem os seus respectivos processos decisórios, uma vez que as informações de interesse comum,armazenadas no microcomputador (ou no computador),podem ser acessadas e/ou alteradas, através destes equipamentos.

Com o aparecimento dos circuitos integrados em larga escala (LSI), em combinação com os microprocessadores, foi possível a implementação dos circuitos do terminal de vídeo com um reduzido número de componentes. O hardware básico para saída gráfica é o sistema de exibição com tubo de raios catódicos (TRC), o qual é constituído de um tubo de raios catódicos e alguns dispositivos eletrônicos que o computador pode controlar.

Nas lições referentes a "Terminais de Vídeo I e II", pretendemos levar ao aluno algumas informações com relação aos principais componentes dos terminais de vídeo, bem como sua classificação e utilização. Assim, iniciaremos esta lição, apresentando o conceito de terminal de vídeo.

CONCEITO DE TERMINAL DE VÍDEO

Basicamente, o terminal de vídeo é um equipamento que possibilita ao operador ter acesso às informações contidas em um computador, para que estas possam ser alteradas e/ou simplesmente acessadas. A figura 1 nos mostra um diagrama de blocos básico de um terminal de vídeo.

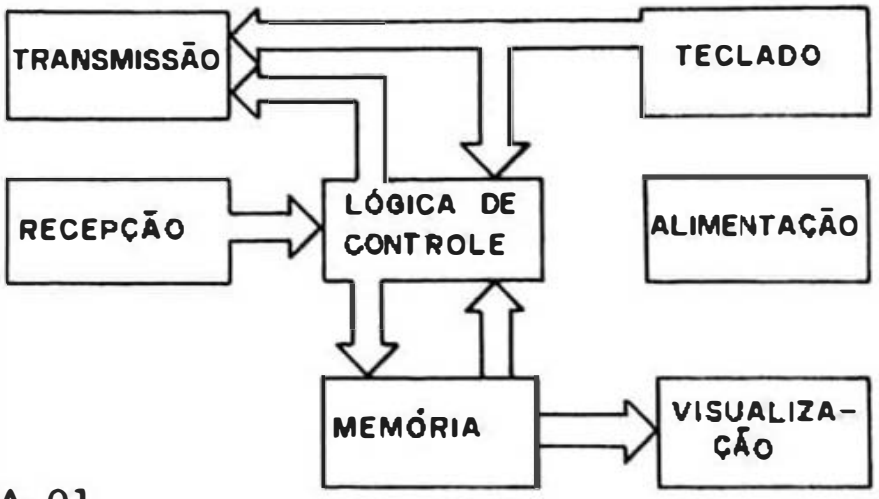


FIGURA 01

O circuito de recepção transforma as variações de tensão ou corrente da linha em níveis compatíveis com a lógica TTL, os quais serão, em seguida, entregues ao processador (ou microprocessador se for o caso) e, caso a informação tiver representação gráfica, o processador irá, então, colocá-la na memória de vídeo.

O bloco de alimentação, como o próprio nome diz, alimenta todos os demais blocos com tensões reguladas e confiáveis.

Através de um terminal de vídeo podem-se normalmente visualizar caracteres do tipo alfanuméricos (números e letras) e gráficos.

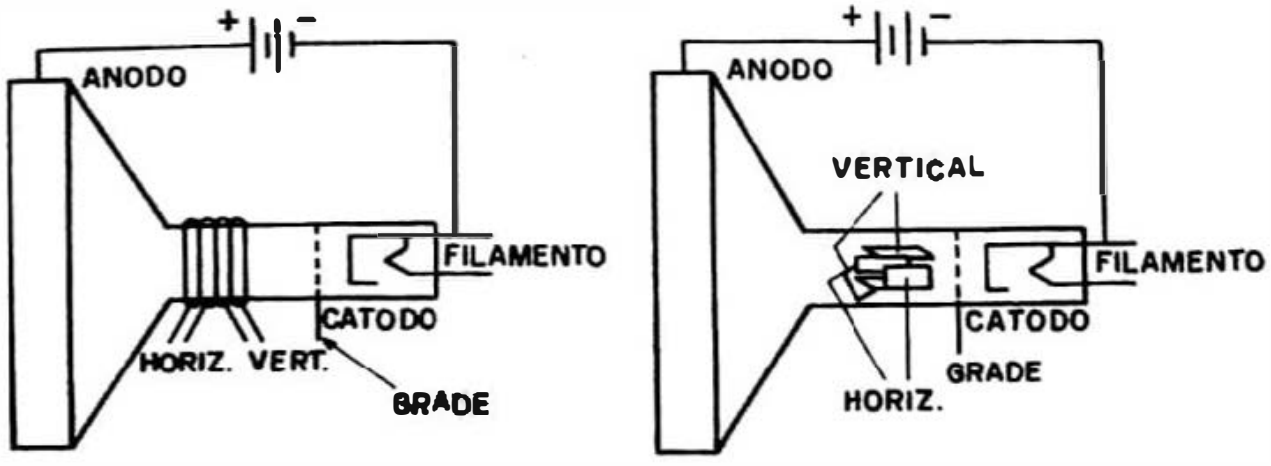
Os terminais de vídeo mais comuns utilizam como tela (ou elemento de exibição) um tubo de raios catódicos (TRC) iguais aos utilizados em aparelhos de T.V.

TUBO DE RAIOS CATÓDICOS (TRC)

Também conhecido por vídeo ou cinescópio, consiste basicamente de um tubo a vácuo dotado de uma fonte de elétrons, chamada catodo, a qual se encontra instalada em uma das extremidades do tubo (veja figura 2). Ao aplicarmos um potencial elevado ao tubo, provocaremos a aceleração do feixe de elétrons produzido pelo catodo, fazendo-o incidir na tela existente na extremidade oposta. A tela é dotada de um revestimento especial, chamado genericamente de fósforo, capaz de transformar o feixe de elétrons num ponto luminoso. A intensidade luminosa do ponto é controlada por um terceiro elemento chamado grade. Ao aplicarmos um determinado potencial à grade, este irá colaborar favoravelmente ou desfavoravelmente com a aceleração dos elétrons.

O movimento do ponto luminoso sobre a tela pode ser controlado de duas maneiras: por deflexão eletromagnética (figura 2-a), ou por deflexão eletrostática (figura 2-b). Na primeira maneira, fazemos variar o campo produzido por um conjunto de bobinas e na segunda maneira, variamos os potenciais aplicados a dois pares de placas, dispostos de modo estratégico, na trajetória do feixe de elétrons.

FIGURA 02



a) Deflexão eletromagnética b) Deflexão eletrostática

O resultado é um movimento como é mostrado, muito embora de forma exagerada, na figura 3.

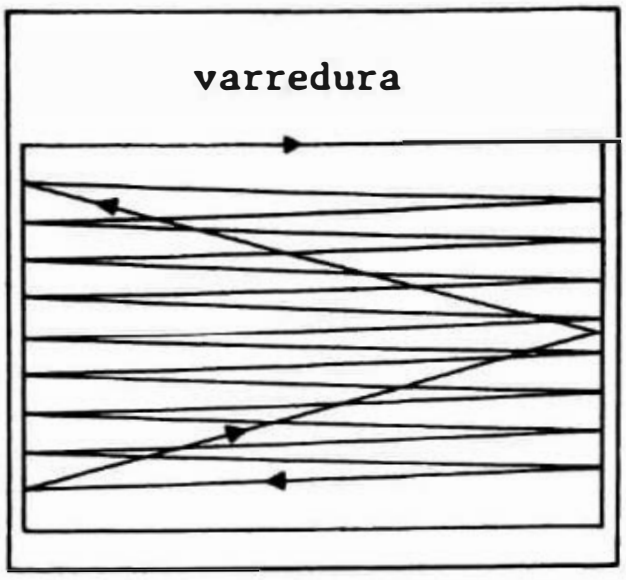


FIGURA 03

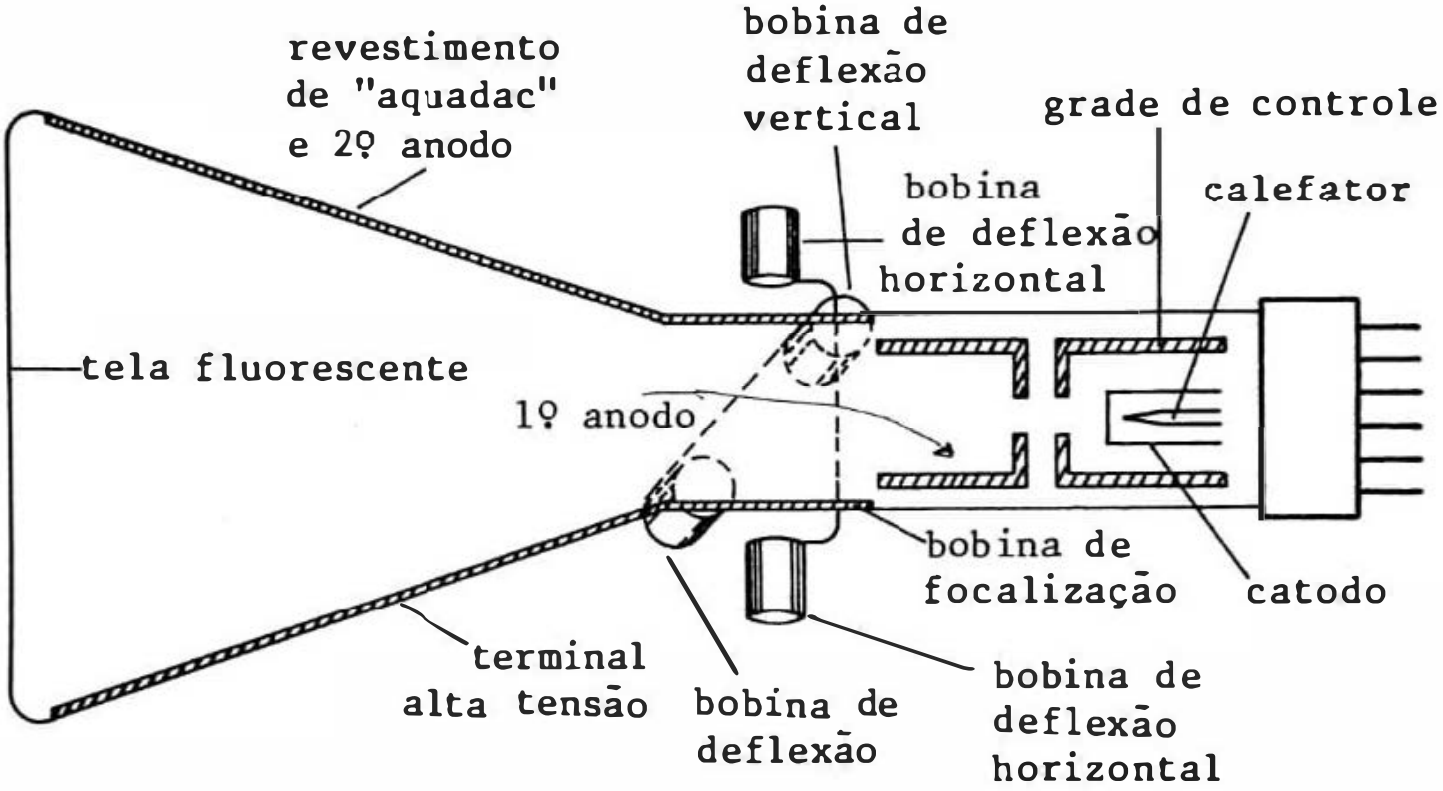
Esse movimento recebe o nome de varredura e as variações da intensidade do ponto luminoso, durante a varredura, é que vão formar as imagens na tela. No Brasil, a varredura horizontal é feita a 15.600 hertz enquanto que a varredura vertical é feita num período muito mais longo: 60 hertz. O nosso sistema de televisão resultante desses números apresenta 30 telas por segundo e cada tela é formada por 525 linhas.

A figura 4 nos mostra um tubo de raios catódicos (TRC) do tipo eletromagnético, onde o feixe eletrônico é focalizado e desviado por um campo magnético.

Note que o tubo apresentado na figura 4 é composto de um filamento calefat, um catodo, uma grade de controle e dois anodos. A função do primeiro anodo é atrair os elétrons do catodo, já a função do segundo anodo, que é o anodo de alta tensão, é acelerar o feixe de elétrons em direção á tela.A bobina de focalização é colocada em volta do pescoço do tubo.

Na deflexão eletromagnética são utilizadas bobinas em vez de placas, as quais são colocadas externamente em volta do pescoço da válvula de imagem (ou canhão eletrônico). A intensidade do campo magnético, produzido por uma dessas bobinas é proporcional à corrente que a atravessa. E o desvio instantâneo do feixe eletrônico é proporcional, portanto, à corrente

instantânea através desta bobina. Para que possamos deslocar o feixe de elétrons com uma velocidade uniforme, é necessário que as correntes, através das bobinas de deflexão, também variem de maneira uniforme. Para realizar isto, faz-se passar, através de cada uma das bobinas, uma corrente com forma de onda dente de serra.



CRT com focalização e deflexão magnéticas

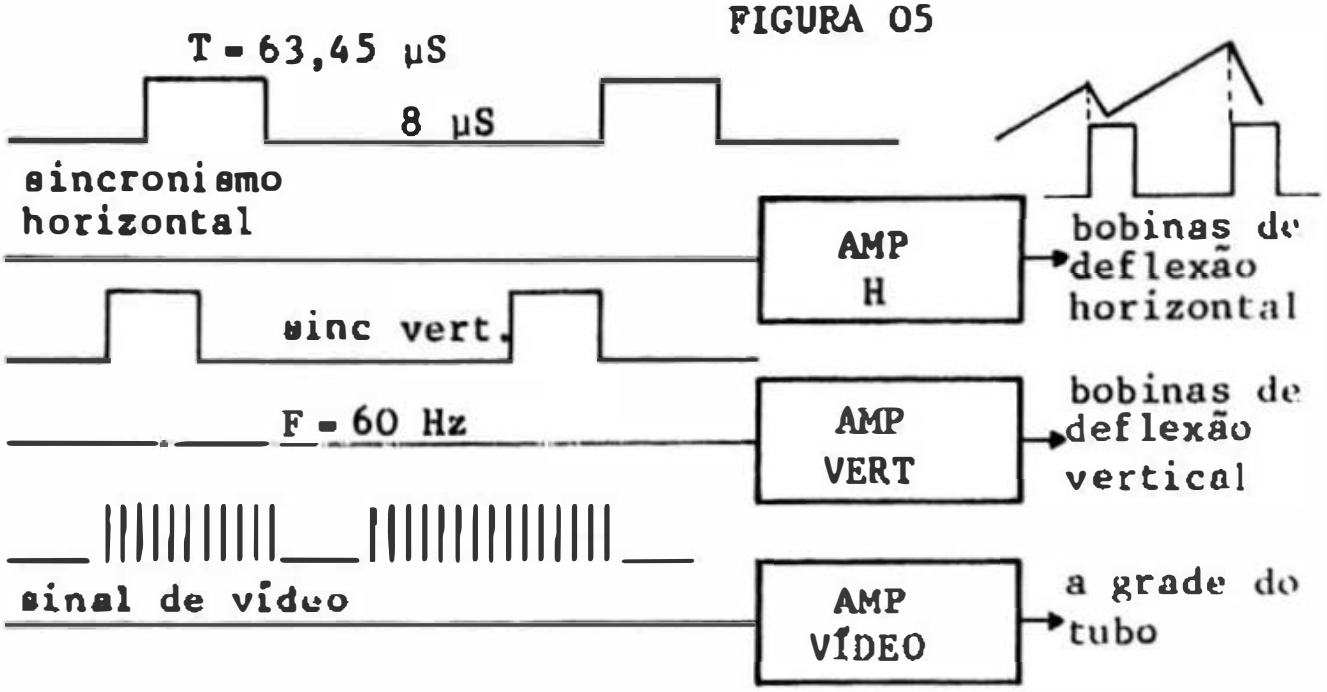
FIGURA 04

A diferença básica entre deflexão eletromagnética e deflexão eletrostática é que, na primeira, como acabamos de ver, utilizamos uma corrente dente de serra, enquanto que na segunda utilizamos uma tensão dente de serra.

Para produzir uma trama (forma) retangular de tela, é necessário que ambas as forças de deflexão horizontal e vertical operem simultaneamente e perpendicularmente entre si, para tanto, utilizam-se dois jogos de bobinas colocadas em volta do pescoço do TRC perpendicularmente um ao outro.

Para que possamos utilizar os circuitos de deflexão dos aparelhos comuns de televisão, os circuitos eletrônicos dos terminais de vídeo precisam utilizar os mesmos sinais de sincronismo horizontal (SINCH) e vertical (SINCV), os quais irão comandar as bobinas de deflexão horizontal e vertical, permitindo uma correta varredura do TRC. A figura 5 nos mostra esses sinais.

FIGURA 05



SINCH, SINCV, SINAL DE VÍDEO

Muito embora possamos empregar o mesmo tipo de tubo de raios catódicos de televisor num microcomputador, dá-se preferência aos TRC, dotados de fósforo de alta persistência, geralmente, verde, pois ele é capaz de transformar em luz um feixe de elétrons de intensidade muito menor. Na realidade, o que acontece é que, nos televisores comuns, o feixe de elétrons é acelerado por potenciais entre 15 e 25 KV (Kilovolts) e, da colisão dos elétrons com o material do tubo, resulta uma pequena emissão de raios X, muito embora, num aparelho de televisão comum, essa emissão não seja prejudicial à saúde, pois o indivíduo se encontra relativamente distante do aparelho. No caso, porém, de terminais de vídeo de microcomputadores, onde o indivíduo, que opera o microcomputador e terminais de vídeo, está a uma pequena distância do vídeo, uma exposição prolongada aos raios X pode ser prejudicial à saúde. Daí a razão de se utilizarem tubos com um potencial acelerador muito menor, de 10 KV e, de preferência, com fósforo de alta persistência.

APLICAÇÕES DO VÍDEO:

Em aplicações digitais, o vídeo pode desempenhar três papéis: dispositivo de exibição alfanumérica, gráfica e semigráfica. No primeiro caso, exibição alfanumérica, a varredura é utilizada para formar letras, números e caracteres especiais através da combinação de barras e pontos. No segundo caso, o vídeo gráfico pode ser digital ou analógico. No caso de vídeos gráficos digitais, o processo de varredura é o mesmo utilizado para a formação da imagem. Porém são dispositivos de elevada resolução (em comparação com as 525 linhas, normalmente, utilizadas nos aparelhos de televisão comuns), podendo atingir até quatro mil linhas na tela. Por outro lado, os vídeos gráficos, analógicos não utilizam varredura contínua, porém traçam na tela em qualquer direção, como nos osciloscópios, através da conversão digital/analógica. Por fim, os vídeos semigráficos, como o próprio nome diz, é um sistena intermediário entre os outros dois já apresentados. Esses vídeos atendem às necessidades gráficas por dispor de figuras no gerador de caracteres, além de permitir também o controle de meios-tons e, eventualmente, de cor. Devido às suas características, este sistema de vídeo é o mais utilizado nos sistemas de microcomputadores.

TIPOS DE TERMINAIS DE EXIBIÇÃO

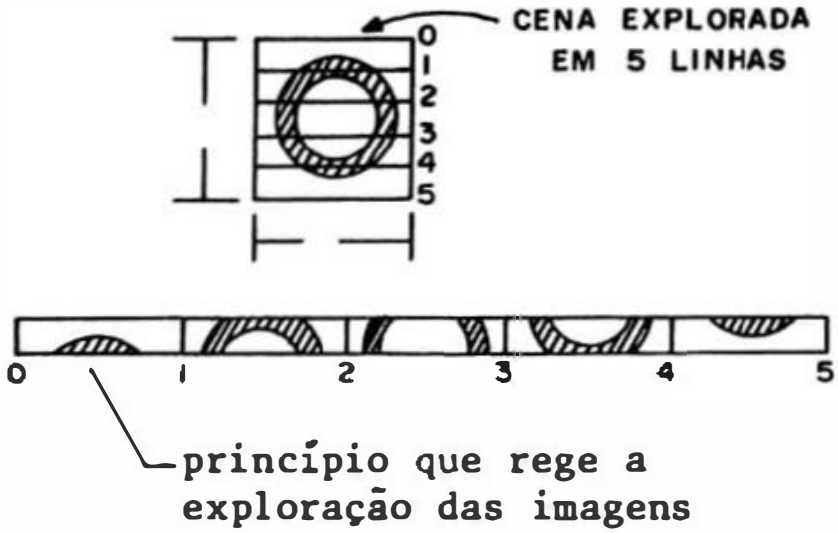
Atualmente, podemos distinguir duas classes amplas de sistemas de exibição para os computadores: terminais de exibição caligráficos e terminais de exibição de rastreamento (mais utilizados em microcomputadores). Os terminais de exibição caligráfica "pintam" as partes de uma figura num tubo de raios catódicos em qualquer seqüência dada pelo computador. Neste caso, o feixe de elétrons se move de um lugar para outro num padrão que traça as linhas individuais e os caracteres que formam a figura. Por outro lado, os terminais de exibição de rastreamento fazem as figuras do mesmo modo que os aparelhos de televisão: a imagem é pintada numa seqüência fixa, geralmente, da esquerda para a direita e de cima para baixo. A vantagem do sistema de exibição caligráfico é que a informação a ser exibida pode ser armazenada numa memória em qualquer ordem, enquanto que a informação para um sistema de exibição de rastreamento deve primeiro ser separada de cima para

baixo e da esquerda para a direita, para poder ser colocada na tela na seqüência correta.

FORMAÇÃO DA IMAGEM DA TELEVISÃO E PRINCÍPIO DE OPERAÇÃO

Em um aparelho de televisão, a imagem ótica é transformada numa imagem eletro-sensível, a qual é, então, decomposta em seus elementos individuais de imagem, ou seja, em impulsos elétricos, por um processo conhecido por "Exploração" e somente um elemento da imagem é transmitido em cada instante. A figura 6 nos mostra uma ilustração do princípio que rege a exploração das imagens.

FIGURA 06



O processo de exploração (ou rastreamento) é semelhante à maneira pela qual um leitor percorre uma página de um livro, ou seja, lendo-a da esquerda para a direita e de cima para baixo.

Como já dissemos, no sistema de televisão adotado no Brasil, a imagem é explorada em 525 linhas, as quais dão um detalhe de, aproximadamente, 200.000 elementos de imagem. Geralmente, entende-se como "Quadro" uma imagem completa de 525 linhas.

No aparelho de televisão, a variação da intensidade luminosa corresponde à variação de amplitude do sinal aplicado ao

amplificador de vídeo, fazendo com que as imagens possuam gradações de luz como cinza claro, cinza escuro, preto, etc.

Já, em vídeos utilizados em terminais, a imagem já não possui normalmente tais gradações de luz, sendo composta de pontos claros (ou pontos luminosos) ou ausência de pontos. Na prática, esse método de transmissão se baseia no fenômeno da persistência da visão, ou seja, o processo de exploração deve ser muito rápido, para que o olho receba a ilusão de uma imagem completa.

Sabendo-se que o olho humano pode reter uma imagem por, aproximadamente, 1/20 de segundo, todos os elementos da imagem devem ser transmitidos dentro desse período de tempo, de modo que o olho passa a ter a impressão da figura inteira em conjunto. Portanto, o processo de exploração deve ser bastante rápido para que a última linha explorada da imagem apareça na tela, enquanto a impressão da primeira linha ainda persiste na visão.

No atual sistema de televisão, é necessário um período de 1/30 de segundos para transmitir todos os elementos contidos em uma imagem completa ou quadro.

CINTILAÇÃO

Quando a freqüência com a qual a luz é projetada não é suficiente para dar a impressão de luz constante, o olho humano irá perceber uma variação rápida da intensidade de iluminação da tela. Essa variação é conhecida como cintilação.

EXPLORAÇÃO PROGRESSIVA E EXPLORAÇÃO INTERCALADA

Este método de exploração consiste no fato de que se a imagem que aparece na câmara de televisão é explorada de tal maneira que as linhas exploradas sucessivamente sejam sempre adjacentes, então a tela inteira do tubo de raios catódicos será iluminada uma vez em cada 1/30 de segundo. Cabe ressaltar o fato de que este método não é usado no atual sistema de televisão, muito embora no que concerne à definição da imagem e à ilusão de continuidade de movimento, a exploração progressiva seria satisfatória.

A não utilização da exploração progressiva é justificada com base no fato de que a freqüência de iluminação de 30 vezes por segundo é muito lenta para dar a impressão de luz constante o que acarretará uma rápida flutuação na iluminação da tela, resultando, assim, em uma cintilação.

Uma solução bem simples, no caso de terminais de computação, seria a de aumentar a freqüência com a qual a tela é iluminada para 60 quadros por segundo, por exemplo. Porém, em termos de televisão, esta solução não pode ser implementada. Então, ao invés de aumentar a freqüência de quadros, passou-se a usar o método conhecido como exploração intercalada, o qual tem o efeito de dobrar a freqüência com a qual a tela é iluminada, sem as exigências que o método de exploração progressiva requereria.

No método de exploração intercalada, em vez das linhas exploradas sucessivamente serem adjacentes, elas são separadas pelo espaço de uma linha, de modo que uma linha sim, uma linha não seja explorada de alto a baixo da tela de imagem, em um período de 1/60 de segundo.

Deste modo, diz-se que cada quadro de imagem é composto de dois "campos": o campo par e o campo ímpar. Sendo que cada um desses campos é composto de 262,5 linhas o que perfaz um total de 525 linhas (as quais formam a tela).

A figura 7 nos mostra alguns detalhes do método de exploração intercalada.

As linhas cheias 1, 3, 5, etc. representam o primeiro campo, ou seja, 262,5 linhas, a partir do ponto "A" até a extremidade inferior da tela e de volta à extremidade superior no ponto "B" em um período de tempo de 1/60 de segundo. Por outro lado, as linhas 2, 4, 6, etc., representam o segundo campo, ou seja, as restantes 262,5 linhas, a partir do ponto "B" até a extremidade inferior da tela e de volta à extremidade superior no ponto "A" em um período de tempo de 1/60 de segundo. As linhas cheias e as interrompidas combinadas fazem um quadro de 525 linhas em 1/60 de segundo.

Deste modo, ao se usar a exploração intercalada, a tela será iluminada 2 vezes por quadro, ou seja, uma vez durante cada campo, aumentando-se a freqüência de iluminação para 60 ciclos por segundo ou 60 hertz.

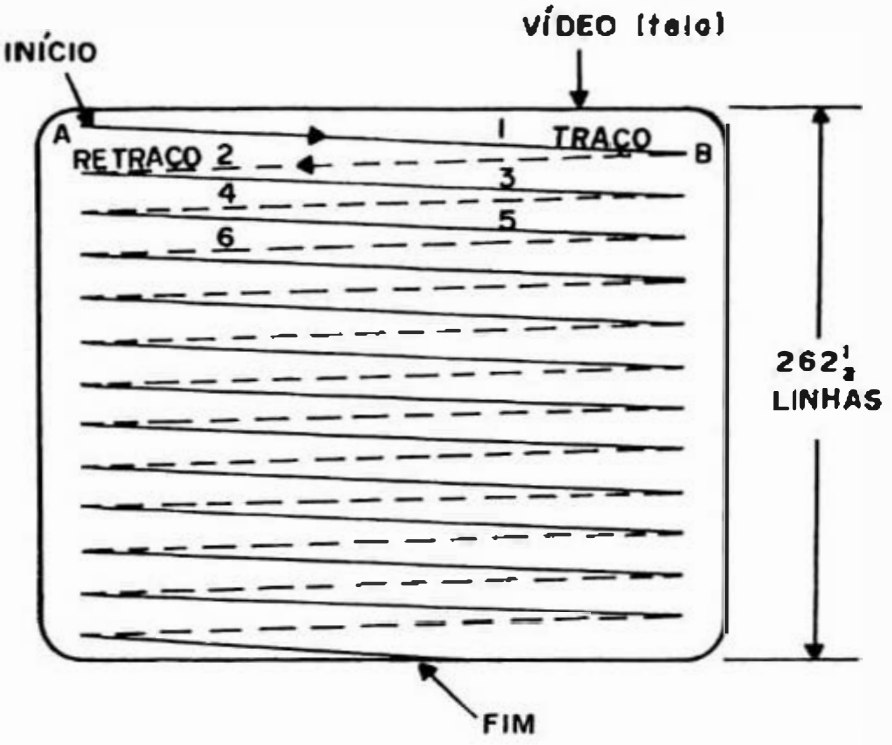


FIGURA 07

Muito embora a freqüência real de quadros seja somente de 30 vezes por segundo, o uso do método de exploração intercalada, no que diz respeito à cintilação, produz o mesmo efeito que se a freqüência de quadros tivesse sido dobrada, uma vez que a vista (olho humano) recebe realmente duas impressões luminosas de conjunto durante cada quadro.

Para finalizar, cabe ressaltar que o método de exploração intercalada não é para melhorar a definição da imagem, mas sim para resolver o problema da cintilação.

IMPLEMENTAÇÃO DE TERMINAIS DE VÍDEO

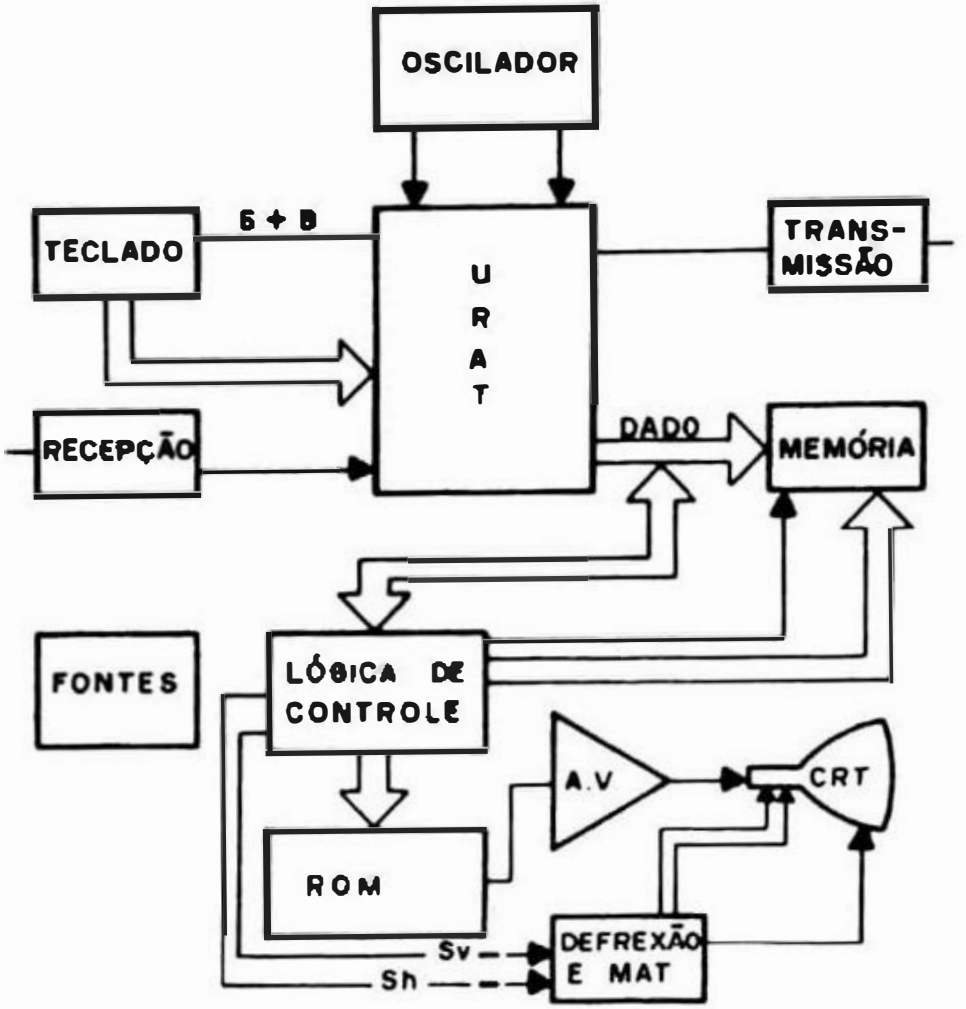
Os terminais de vídeo podem ser implementados a partir da lógica convencional ou com microprocessadores.

A figura 8 nos mostra um terminal de vídeo implementado com lógica convencional, com pouco recursos. Este tipo de terminal é aplicado em substituição aos terminais eletromecânicos.

Num terminal de vídeo implementado com lógica convencional, o teclado, ao gerar um pulso de "strobe", faz com que a UART (receptor-transmissor universal assíncrono) pegue o con-

teúdo do barramento (ou simplesmente barra) de dados e apresente em sua saída-série a informação já serializada, com velocidade de comunicação definida pelo oscilador. Esta informação serializada é então aplicada na interface de transmissão, a qual irá lhe dar o tratamento adequado.

FIGURA 08



A interface de recepção transforma as variações de níveis de tensão ou corrente em níveis de tensão compatíveis com a lógica TTL. Esta informação serializada é então conectada à entrada série da UART, a qual irá realizar a conversão de dados série para paralelo. Uma vez convertidos os dados série para paralelo, eles estarão presentes no barramento de dados e a lógica de controle irá então se incumbir de verificar se o dado em questão é um dado com representação gráfica. Se afirmativo, este dado será então escrito na memória RAM, no endereço correspondente ao cursor.

Durante os ciclos de refrescamento de vídeo, o conteúdo da memória de vídeo é transmitido para o tubo de raios catódicos, por meio de uma lógica de controle, a qual armazena o conteúdo de uma linha de caracteres e o transmite linha de vídeo por linha de vídeo.

O circuito gerador de caracteres é formado por uma memória ROM, que transforma o código ASCII do caracter,em códigos correspondentes a sinais visuais. Em seguida, esses sinais passam pelo amplificador de vídeo (A.V.), o qual consiste de um circuito que irá transformar os níveis TTL recebidos, em níveis de tensão suficientemente elevados para ativar o tubo de raios catódicos.

Os circuitos de deflexão são os responsáveis pelo deslocamento do feixe eletrônico na tela do tubo de raios catódicos (veja figura 9).

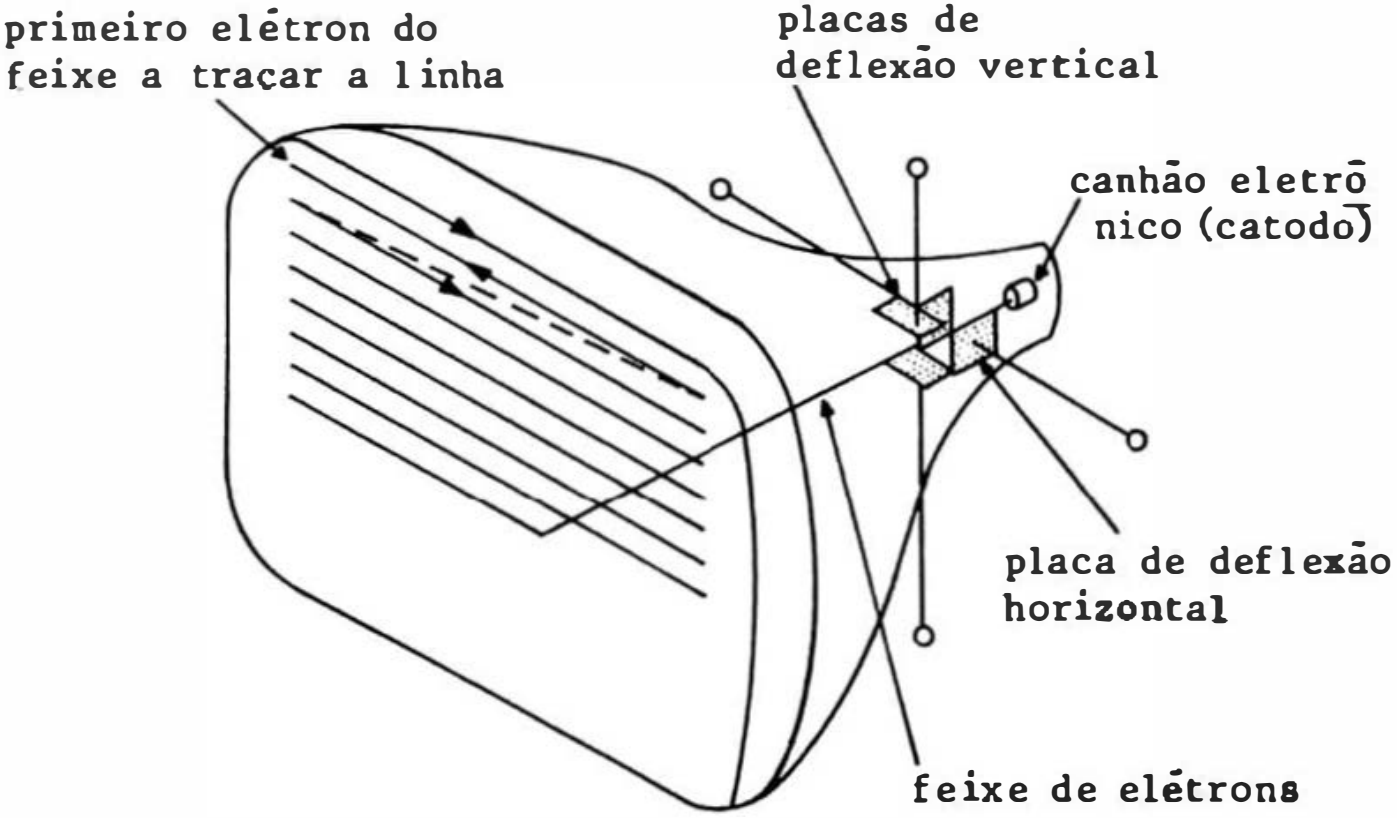


FIGURA 09

A tensão gerada a partir de um circuito tipo "flyback", com base na tensão gerada na deflexão horizontal, dá-se o nome de MAT (muito alta tensão), a qual é utilizada para alimen

tar o anodo do tubo de raios catódicos.

Vejamos agora um terminal de vídeo implementado com microprocessador, também conhecido como "terminal inteligente", uma vez que possibilita uma maior flexibilidade do equipamento, além de emitir notas (ou avisos) de erro.

A figura 10 nos mostra o diagrama de blocos de um terminal de vídeo implementado com microprocessadores.

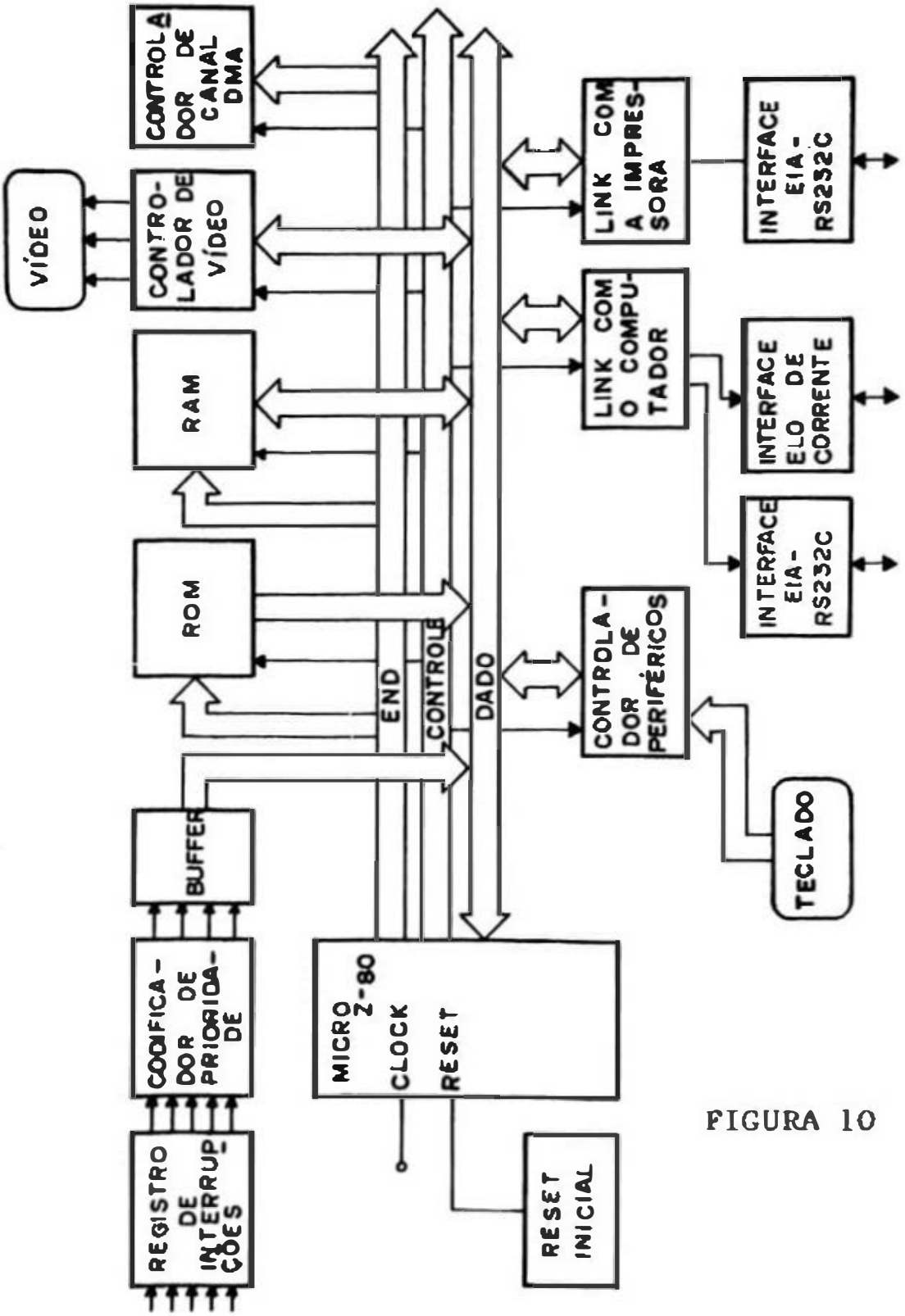


FIGURA 10

Na memória ROM estão contidas todas as características do terminal, determinadas pelo projetista, as quais irão orientar exatamente como será o funcionamento do terminal. Ao ligarmos o terminal, o circuito de "reset inicial" atuará sobre o microprocessador e sobre outros componentes. Após o reset, o microprocessador irá executar o programa contido na memória ROM; inicializando todas as variáveis dos circuitos periféricos e, em seguida, passará para um programa de espera de interrupção do microprocessador, podendo esta interrupção vir do teclado ou do vídeo ou ainda de qualquer outro bloco, para lhe dar o tratamento necessário.

Os circuitos de tratamento de interrupção irão fazer um julgamento das mesmas e, se duas ou mais interrupções ocorrerem ao mesmo tempo, estes circuitos farão com que a interrupção de maior prioridade tenha tratamento preferencial.

A memória RAM é utilizada como sendo área de vídeo, memória de trabalho do microprocessador e outras atividades direcionadas para a comunicação.

O controlador do canal do dispositivo de acesso direto á memória (DMA) atua juntamente com o controlador de vídeo, fazendo a transferência do conteúdo de cada linha de caracteres completa, porém, uma por vez para não sobrecarregar o microprocessador. O circuito de vídeo, semelhante ao do diagrama de blocos anterior, gera sinais de vídeo, sinais de sincronismo horizontal e vertical, sinais de controle como, por exemplo, o sinal "highlight", o qual proporciona maior intensidade de iluminação a algumas áreas do vídeo (ou da tela), podendo este ser controlado por intermédio de "software". Um outro exemplo de sinal de controle seria o de supressão do sinal de vídeo, o qual serve para tornar certos campos invisíveis ao operador.

Toda vez que ocorrer uma transferência de linha de caracteres, o controlador DMA assume o controle do barramento de dados e endereços, além, é claro, das linhas de controle. O controle passa para o DMA por intermédio do sinal HOLD aplicado na UCP, fazendo a transferência sem a intervenção da mesma.

O controlador de periféricos, por sua vez, controla o teclado e coloca o conteúdo do dado digitado no barramento de

dados, quando for solicitado e o programa de tratamento de interrupções de teclado lhe dará o devido tratamento.

A figura 11 nos mostra o diagrama de blocos para uma possível configuração de um terminal inteligente, controlado por intermédio de um microprocessador.

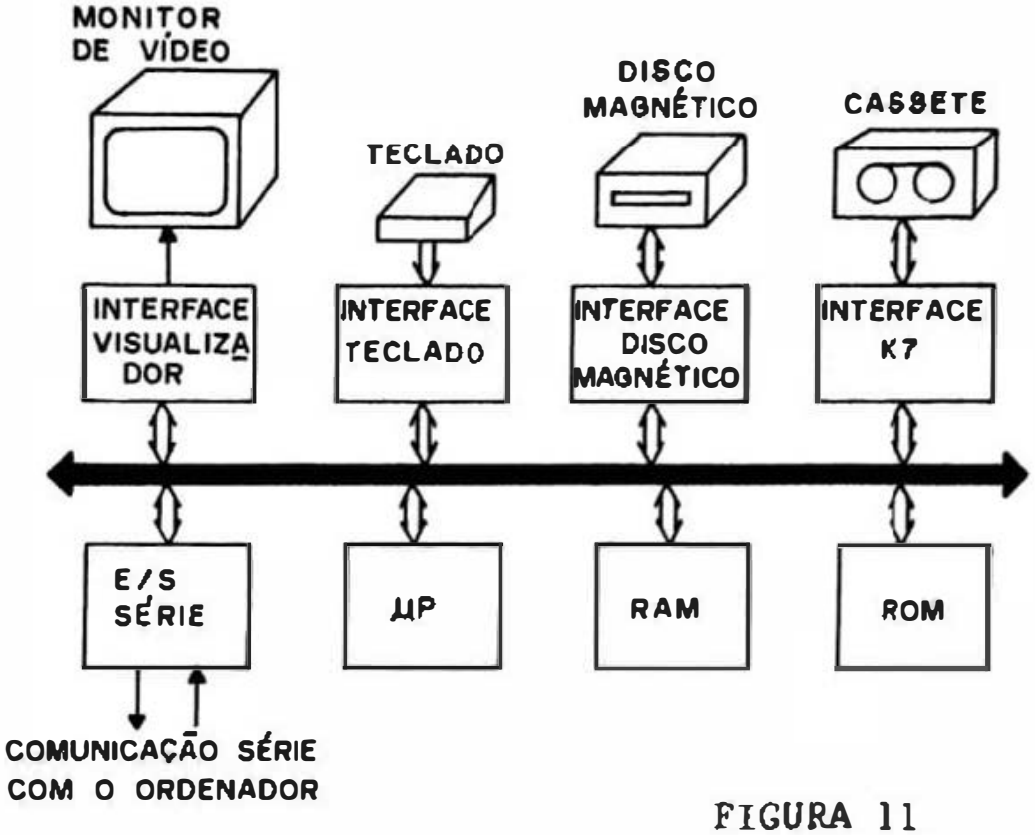


FIGURA 11

Uma das vantagens da implementação com microprocessadores é que o circuito de controle próprio do visualizador se torna mais sensível a um pré-tratamento da informação a ser projetada na tela de um tubo de raios catódicos.

O circuito de controle do vídeo pode ser implementado a partir de componentes discretos ou através da utilização de um circuito integrado especializado. A utilização de um CI especializado tem a vantagem de que este pode admitir várias formas de funcionamento. O circuito de controle deve realizar todas as tarefas que sejam função do tempo, por exemplo, a geração de um sinal de refresco para a tela e o controle da transferência de informação entre memória e gerador de caracteres.

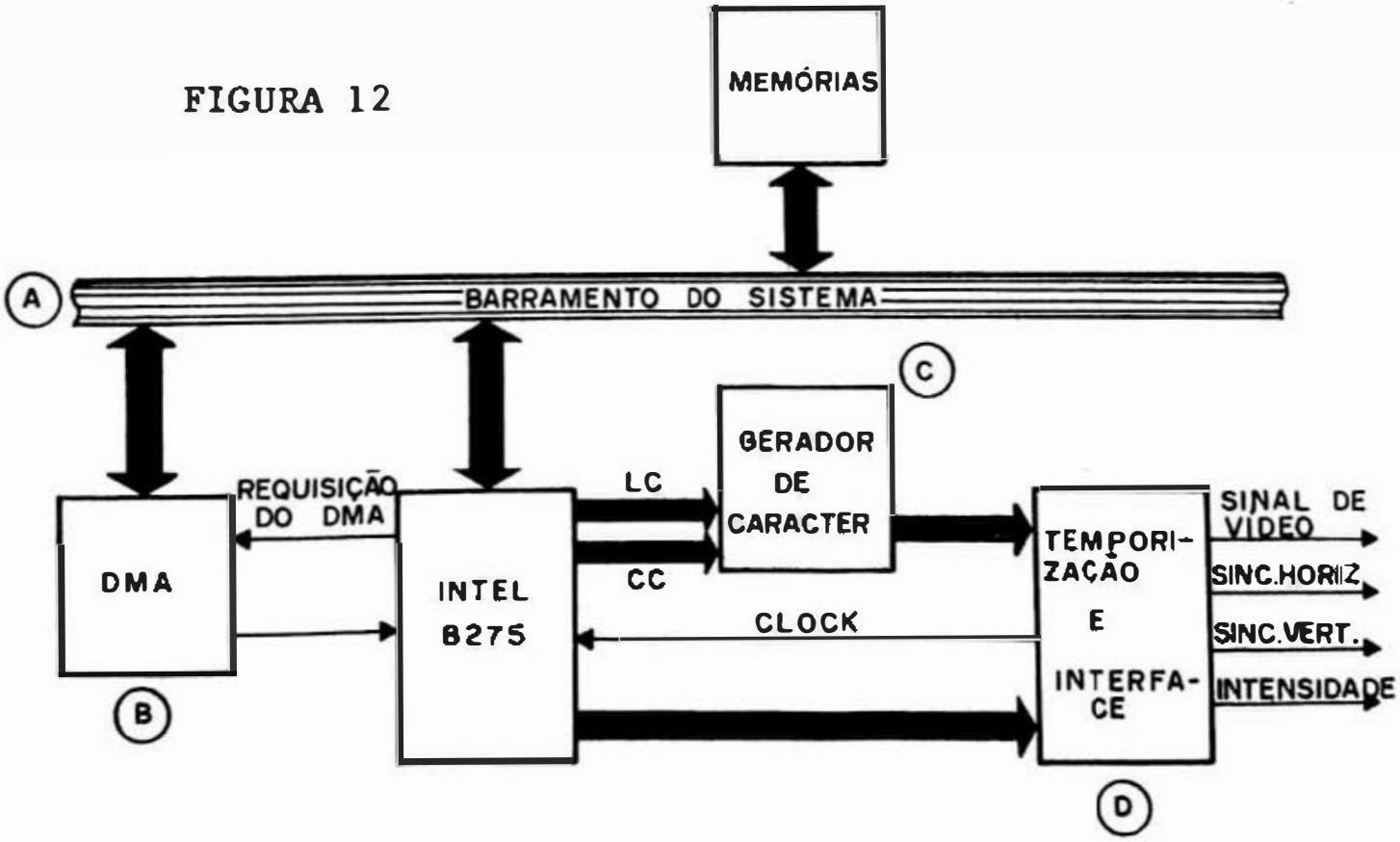
Uma outra vantagem que apresenta a implementação com microprocessadores é que, como já dissemos anteriormente, pode-se dotar o sistema de um certo grau de inteligência e realizar tarefas como edição de textos e controle de unidades de armazenamento de informação.

Como exemplo de circuito integrado especializado no controle de vídeo, iremos apresentar o CI 8275, fabricado pela Intel.

CONTROLADOR PROGRAMÁVEL DE VÍDEO INTEL 8275:

O integrado 8275 da Intel fornece os controles necessários às aplicações alfanuméricas e semigráficas simples. A figura 12 nos mostra a posição do 8275 dentro de um sistema, aqui representado sob a forma de um diagrama de blocos.

FIGURA 12

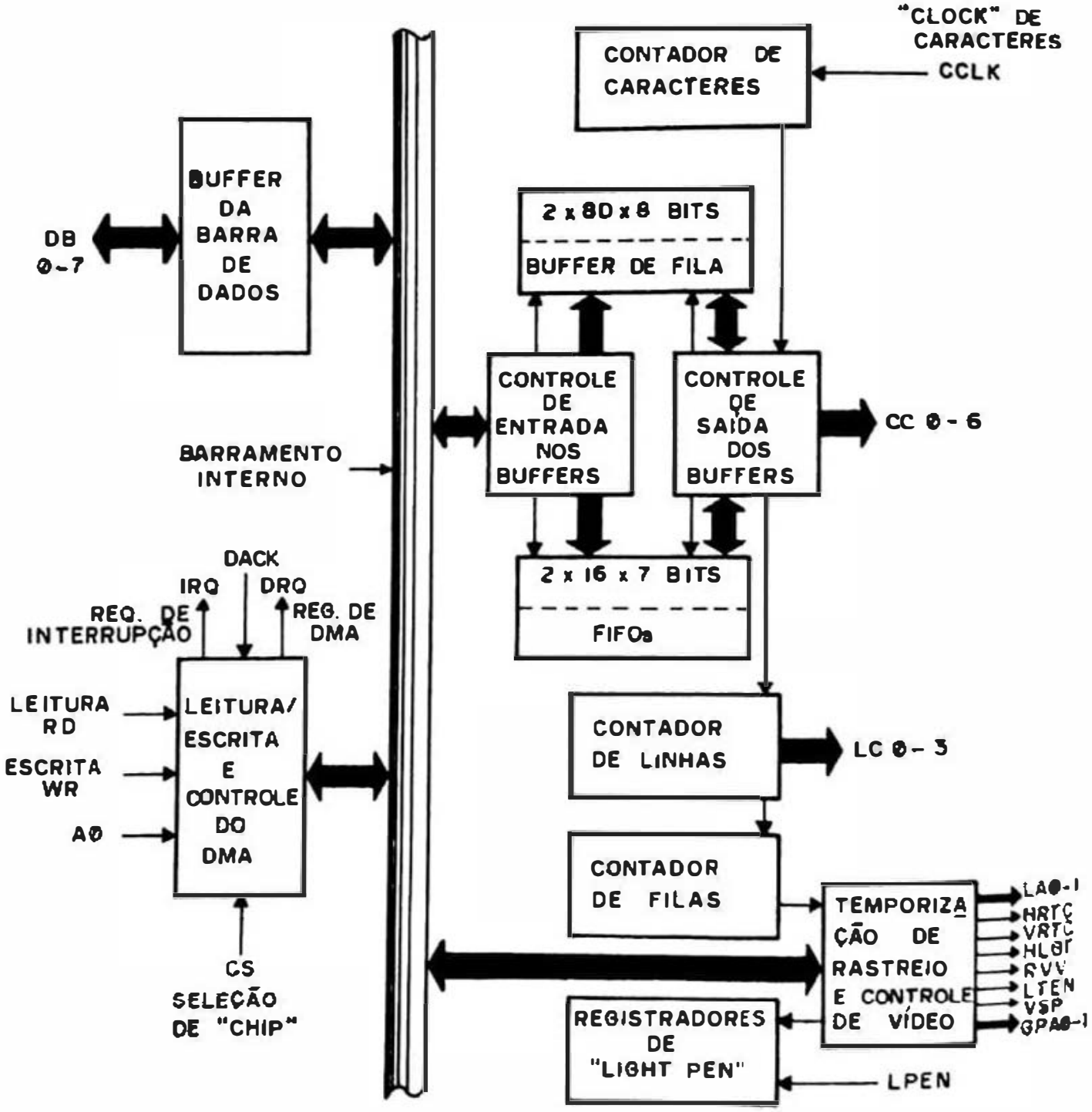


Diretamente do barramento do sistema, o 8275 recebe comandos de programação ou leitura de seus registros internos e pode ainda interferir no sistema, fazendo uma solicitação de in

terrupção. Através de um DMA, o 8275 recebe uma linha de teste completa, a ser exibida na tela. Este CI possui ainda uma matriz geradora de caracteres (normalmente uma memória do tipo EPROM), cuja finalidade é traduzir o caracter em bits correspondentes aos pontos e barras que formam o caracter.

A figura 13 nos mostra um diagrama de blocos do controlador programável de vídeo 8275.

FIGURA 13



Para finalizar,a figura 14 nos mostra a composição do código de atributo de campo no CI 8275. As funções correspondentes aos bits selecionados passam a ter efeito no próximo caracter mostrado na tela e o efeito permanece para todo o campo, até o instante em que ocorra outro código de atributo de campo, ou o fim da tela.

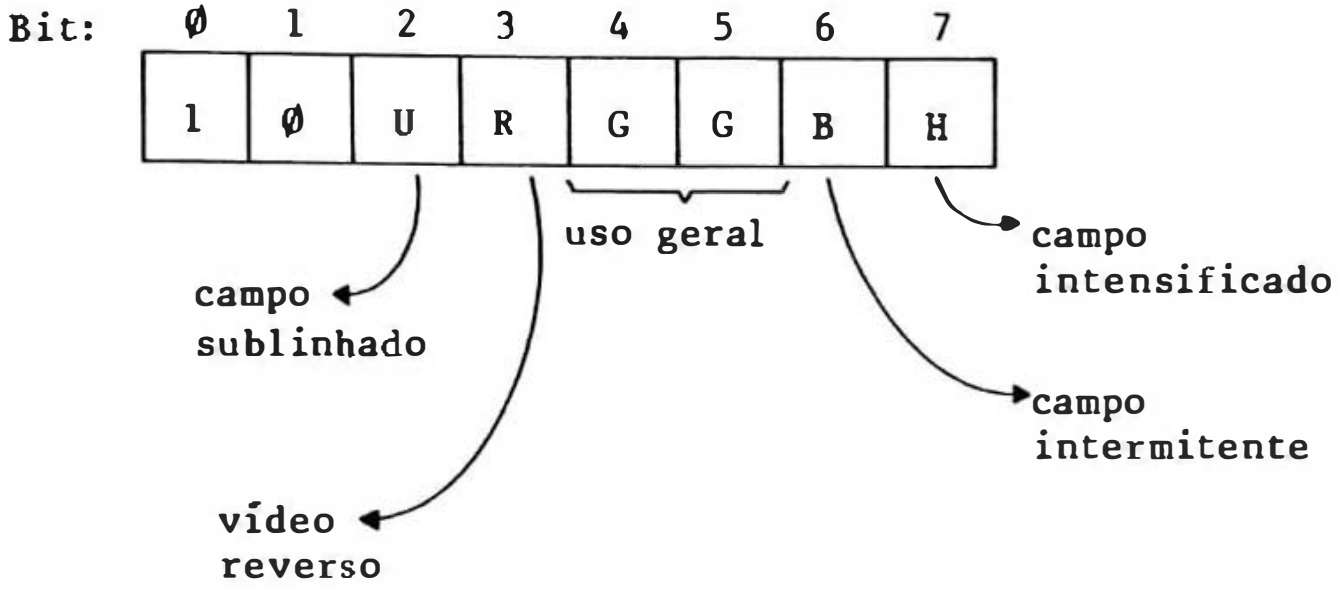


FIGURA 14

O campo sublinhado é obtido, ativando-se o bit 2. O vídeo reverso é controlado pelo bit 3, produzindo um campo claro com letras escuras. Os bits 4 e 5 são os atributos de uso geral, os quais permitem aos projetistas produzir qualquer efeito no vídeo. Já, o campo intermitente, os piscantes (blink), que é controlado pelo bit 6, ativa a linha VSP, suprimindo a exibição do campo em períodos de 32 vezes o traço vertical. Por outro lado, o campo intensificado (bit 7) ativa a linha HLGT, a qual comandará o aumento de brilho nesse campo.

Além desses códigos de atributo de campo, há também os códigos de atributo de caracter, cuja finalidade básica é prover o sistema de um conjunto de códigos para aplicações semigráficas, os quais não necessitam ser decodificados pelo gera

dor de caracteres. A figura 15 nos mostra a composição dos códigos de atrito de caracter.

atributo

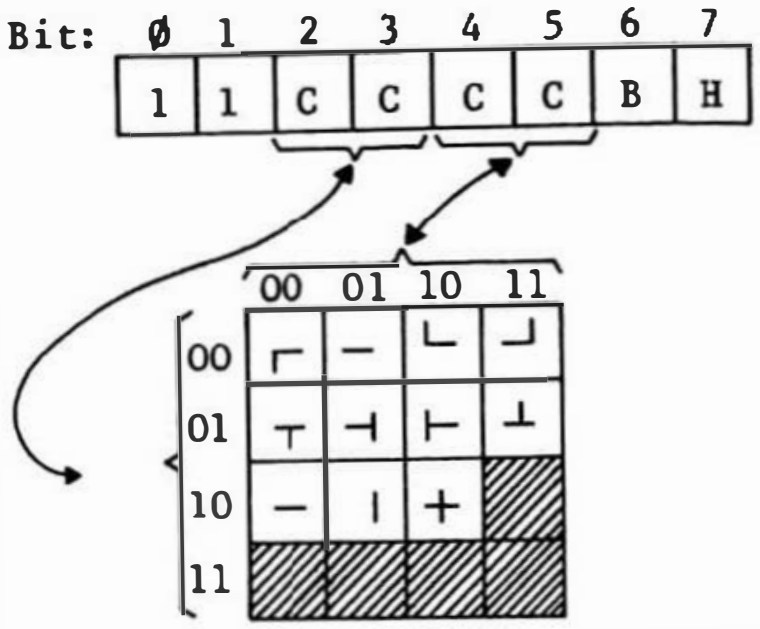
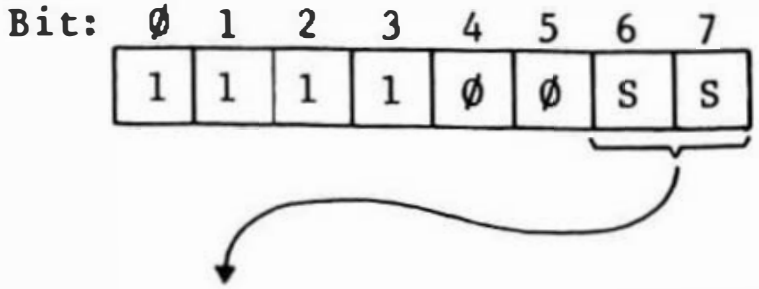


FIGURA 15

Por último, temos, na figura 16, os códigos especiais de controle os quais se destinam a economizar memória, simplificar a programação do vídeo e minimizar o uso do DMA e, conseqüentemente, o acesso à memória.

FIGURA 16



00 = fim de fila
01 = fim de fila, parar o DMA
10 = fim de tela
11 = fim de tela, parar o DMA

Na lição "Terminal de Vídeo II", estudaremos a formação dos caracteres e veremos algumas das principais características dos terminais de vídeo.

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES
CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANA

EXAME DA LIÇÃO MC-20

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Através de um terminal de vídeo, um operador pode:

a) ter acesso às informações contidas em um computador;

b) alterar as informações contidas em um computador;

c) os itens a e b estão corretos;

d) N.R.A.

2) Os terminais de vídeo mais comuns utilizam como tela:

a) um display;

b) um tubo de raios catódicos (TRC);

c) um tubo de raios caóticos;

d) N.R.A.

3) Num tubo de raios catódicos o feixe de elétrons é produzido pelo:

a) catodo;

b) anodo;

c) tela;

d) N.R.A.

4) A tela de um TRC é revestida por um elemento capaz de transformar o feixe de elétrons:

a) em um conjunto de pontos escuros;

b) em um ponto luminoso;

c) em um feixe de prótons;

d) N.R.A.

5) O movimento do ponto luminoso sobre a tela pode ser controlado por:

a) deflexão eletromagnética;

b) deflexão eletromecânica;

c) deflexão eletrostática;

d) os itens a e c estão corretos.

6) Normalmente um TRC, como o apresentado na figura 4, possui:

a) 1 anodo;

b) 2 catodos;

c) 2 anodos;

d) N.R.A.

7) Na deflexão eletromagnética, ao invés de placas, são utilizadas:

a) bobinas;

b) telas fosforescentes;

c) grades;

d) planos defletores.

8) Em aplicações digitais, o vídeo pode ser utilizado como dispositivo de exibição:

a) alfanumérica;

b) gráfica;

c) semigráfica;

d) os itens a, b e c estão corretos.

9) No método de exploração intercalada, teremos uma linha sim, uma linha não, explorada de alto a baixo da tela em período de:

a) 1/30 de segundo;

b) 1/60 de segundo;

c) 1/90 de segundo;

d) 1/525 de segundo.

10) Os terminais de vídeo implementados com microprocessadores também recebem o nome de:

a) terminais lógicos;

b) terminais inteligentes;

c) terminais mecânicos;

d) N.R.A.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-21

TERMINAL DE VÍDEO II

INTRODUÇÃO

Nesta lição, verificaremos a formação dos caracteres e as principais características dos terminais de vídeo.

FORMAÇÃO DO CARACTER

Os caracteres alfanuméricos se representam sobre a tela do tubo de raios catódicos, segundo um formato baseado em uma matriz de pontos. Dois são os formatos normalizados mais comuns, utilizados na representação de caracteres alfanuméricos:

a) Formato 5x7:

Neste formato, cada caracter é representado mediante uma série de pontos distribuídos dentro de uma matriz de 5 colunas por 7 linhas. (Veja figura 1)

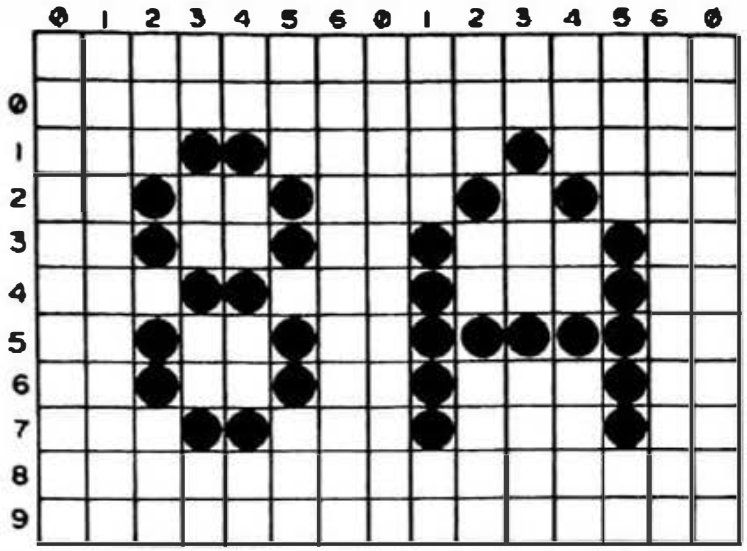


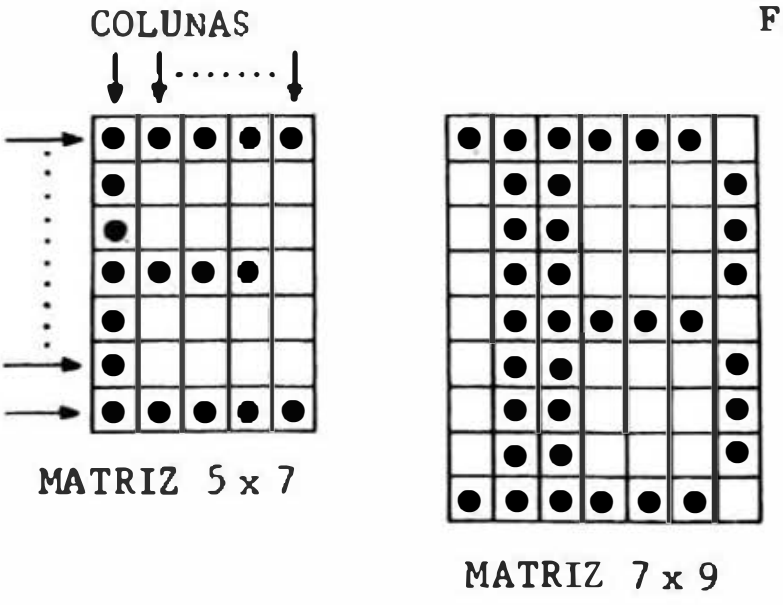
FIGURA 01

Um ponto na matriz representa um ponto luminoso na tela do tubo de raios catódicos e sua ausência representa o efeito contrário.

b) Formato 7x9:

Neste caso, a matriz, que representa o caracter, possui 7 colunas e 9 linhas. A figura 2 nos mostra um caracter, cujo formato é baseado numa matriz 7x9.

FIGURA 02



Ambos os formatos são similares e apresentam suas vantagens e desvantagens. O formato 7x9 possui uma maior resolução para representar os caracteres. Por outro lado, a maior quantidade de pontos da matriz requer um incremento da banda do amplificador de vídeo para a representação da mesma quantidade de caracteres em uma linha. No caso de 64 caracteres por linha, a largura da banda requerida supera em vários megahertz a largura de banda do amplificador de vídeo de um aparelho receptor de televisão comum. Devido a este fato, o formato 7x9 somente é utilizado nos monitores digitais alfanuméricos.

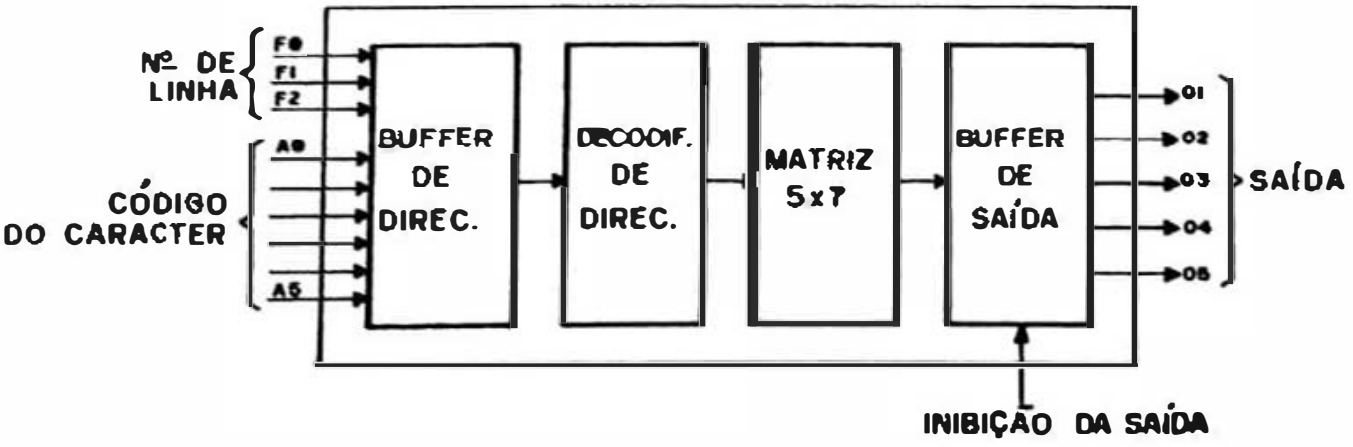
Já, o formato 5x7 apresenta uma resolução aceitável para monitores que utilizam um receptor de televisão comum.

Em um terminal alfanumérico, podemos representar pelo menos 64 tipos diferentes de caracteres. A informação correspondente às matrizes de pontos de todos esses caracteres pode ser armazenada em uma memória de semicondutores tipo ROM.

GERADORES DE CARACTERES

A figura 3 nos mostra o diagrama de blocos de um gerador de caracteres.

FIGURA 03



Este circuito utiliza, como parâmetro de entrada, o código ASCII e o número da linha correspondente ao caracter que se está visualizando. Na saída do gerador de caracteres, teremos a informação em paralelo da linha da matriz direcionada (5 linhas para uma matriz 5x7).

Os geradores de caracteres são fabricados pela maioria dos fabricantes de dispositivos semicondutores e trata-se de um componente de baixo custo.

GERAÇÃO DE LINHAS DE TEXTO

Para representarmos uma linha de texto de "n" caracteres, formados por uma matriz 5x7, precisaremos, no mínimo, de 8 varreduras horizontais da tela do monitor, ou seja, uma varredura para cada linha da matriz de pontos e mais uma, pelo menos, para separar entre si duas linhas de caracteres. (Veja figura 4).

A primeira varredura representa os "n" grupos de 5 pontos dos "n" caracteres da linha, correspondentes à primeira linha da matriz de pontos. O mesmo ocorre com a segunda varredura em relação à segunda linha e, assim, sucessivamente até totalizar as 7 linhas da matriz de pontos.

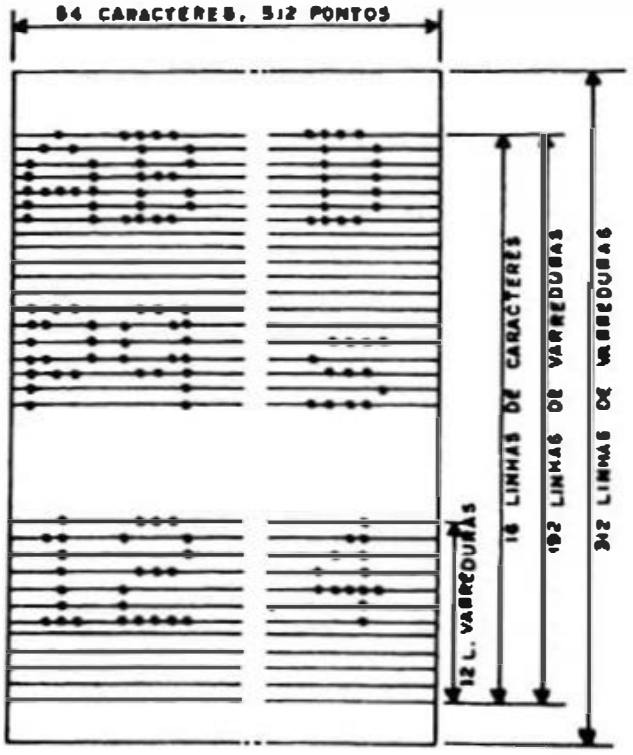


FIGURA 04

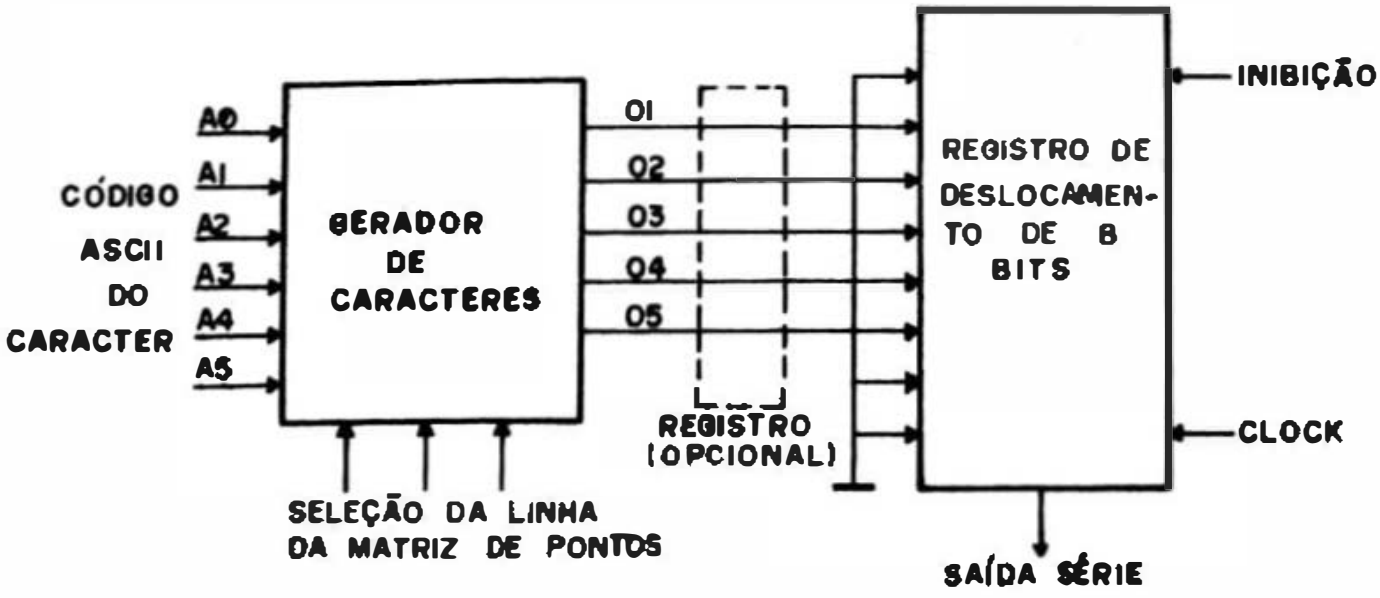
Normalmente, entre uma linha de caracteres e outra, intercala-se mais de uma varredura em branco para se obter uma representação visual do texto. Assim sendo, o número de varreduras, para cada linha de caracteres, pode chegar a 10, 12 ou mais.

Para representar completamente uma linha de caracteres alfanuméricos, extraem-se os códigos ASCII destes, da memória em forma seqüencial, tantas vezes quantas linhas tiverem a matriz de pontos utilizada, ou seja, devem ler-se os códigos de todos os caracteres de forma seqüencial em cada varreção do feixe eletrônico. Precisa-se, para tanto, de um circuito contador, que nos indique qual é o número do caracter da linha, que estamos visualizando naquele momento e de outro circuito contador, que nos indique qual é o número da linha da matriz de pontos, que devemos visualizar.

Como haverá também um contador de linhas de texto, em todo momento, poderemos saber a direção absoluta da posição da memória, que devemos ler em cada instante para obter o código ASCII do caracter a ser visualizado.

Na saída do gerador de caracter, temos, em paralelo, a informação referente à linha da matriz de pontos direcionada (cinco linhas). Esta informação é então carregada (ou ligada) em paralelo a um registrador de deslocamento governado por um sinal de clock de freqüência igual à freqüência de visualização dos pontos sobre a tela. A saída série deste registrador de deslocamento é o sinal de vídeo que controla o feixe eletrônico. Este circuito é conhecido como "serealizador de informação". A figura 5 nos mostra o diagrama de blocos deste circuito.

FIGURA 05



Note que, para podermos visualizar totalmente uma linha de caracteres, os códigos dos caracteres são lidos tantas vezes quantas linhas tiver a matriz de pontos utilizada.

Pode-se, ainda, empregar uma memória do tipo FIFO para operar como registrador de deslocamento.

GERAÇÃO DE UMA PÁGINA DE TEXTO

A geração de uma página de texto, composta por "n" linhas de caracteres, é feita repetindo-se n vezes o processo utilizado para a geração de linhas de textos. Porém, neste caso, necessita-se de um circuito contador que indique em qualquer momento o número da linha que está sendo visualizada.

As leituras das linhas de texto na memória podem ser efetuadas de duas formas:

a) Leitura seqüencial

Neste caso, o texto representado na tela do monitor está disposto exatamente da mesma forma que na memória do sistema. A direção inicial de cada linha do texto é calculada a partir da direção inicial do texto e do estado do contador de linhas visualizadas. Este método possui a vantagem de ser muito sensível e de ser facilmente controlado. Por outro lado, apresenta o inconveniente de que a modificação de uma linha de caracteres implica, muitas vezes, na modificação parcial ou total do conteúdo da memória de texto.

b) Leitura aleatória

O texto representado na tela do monitor não está disposto da mesma forma que na memória, como no caso anterior. Então, neste caso, utiliza-se uma memória auxiliar devidamente associada ao contador de linhas visualizadas para se determinar ou para se realizar o cálculo da direção de cada linha. Para modificar a ordem das linhas visualizadas, basta modificar o conteúdo da memória. Este método permite maior flexibilidade na utilização do monitor de vídeo, porém aumenta a complexibilidade do projeto e do controle do mesmo, tornando-se imprescindível a utilização de um microprocessador para tal.

CURSOR

O cursor, normalmente, representado na tela de um monitor de tubo de raios catódicos, mediante um pequeno "guia", indica a posição em que se visualizará o próximo caracter que o sistema visualizador enviará e serve, muitas vezes, para sincronizar a página de texto.

O cursor está situado normalmente dentro da última linha do texto, representado na tela e vai se deslocando da esquerda para a direita, mediante à chegada dos novos caracteres. Deve existir somente um cursor em uma página de texto, utilizando, para sua representação na memória do texto, um bit adicional para cada caracter. Este bit adicional assume o valor lógico 1, somente para o caracter a que estiver associado o cursor.

O controle do posicionamento do cursor dentro da memória do texto deve ser efetuado mediante o uso de um microprocessador ou mediante um circuito controlador especial. Parte da gestão do cursor deve estar integrada aos CIs controladores de monitores de tubo de raios catódicos.

Nos sistemas em que não se utiliza o cursor como forma de sincronismo de página de texto, a direção do primeiro caracter a ser visualizado na tela é sempre a mesma, no caso de se utilizar memória local ou memória comum de página fixa. No caso de se dispor de várias páginas de texto o primeiro caracter pode ser introduzido por intermédio de um microprocessador conectado a um registrador de direção inicial de página.

REFRESCAMENTO DA TELA DO MONITOR DE VÍDEO

Para que a imagem visualizada seja estável perante os olhos do operador é necessário refrescar a tela do monitor periodicamente.

O refrescamento da tela consiste em reescrever nela, a cada 20 ms, aproximadamente, todo o texto a visualizar, de modo a transferir, periodicamente, todos os códigos dos caracteres visualizavéis desde a memória de textos até o circuito de controle. Esse intercâmbio de dados entre a memória e o sistema de controle do monitor pode ser feito de quatro

maneiras diferentes, mediante:

- amostras;
- interrupções;
- acesso direto à memória (DMA);
- acesso transparente à memória de textos.

No sistema do monitor de vídeo, deve existir algum elemento encarregado de dar ordens oportunas para realizar a tarefa de transferência da informação, desde a memória de textos até o circuito de controle do monitor no instante em que se deve refrescar a tela.

Esta tarefa pode ser realizada por um microcomputador ou pelo circuito de controle ou circuitos especializados. A utilização de CIs especializados no controle dos monitores simplifica em muito o problema, já que a gestão do refresco é uma função que todos eles imcorporam (vela "Implementação de terminais de vídeo", na lição "Terminais de Vídeo I"). A figura 6 nos mostra o diagrama de blocos de um controlador de vídeo.

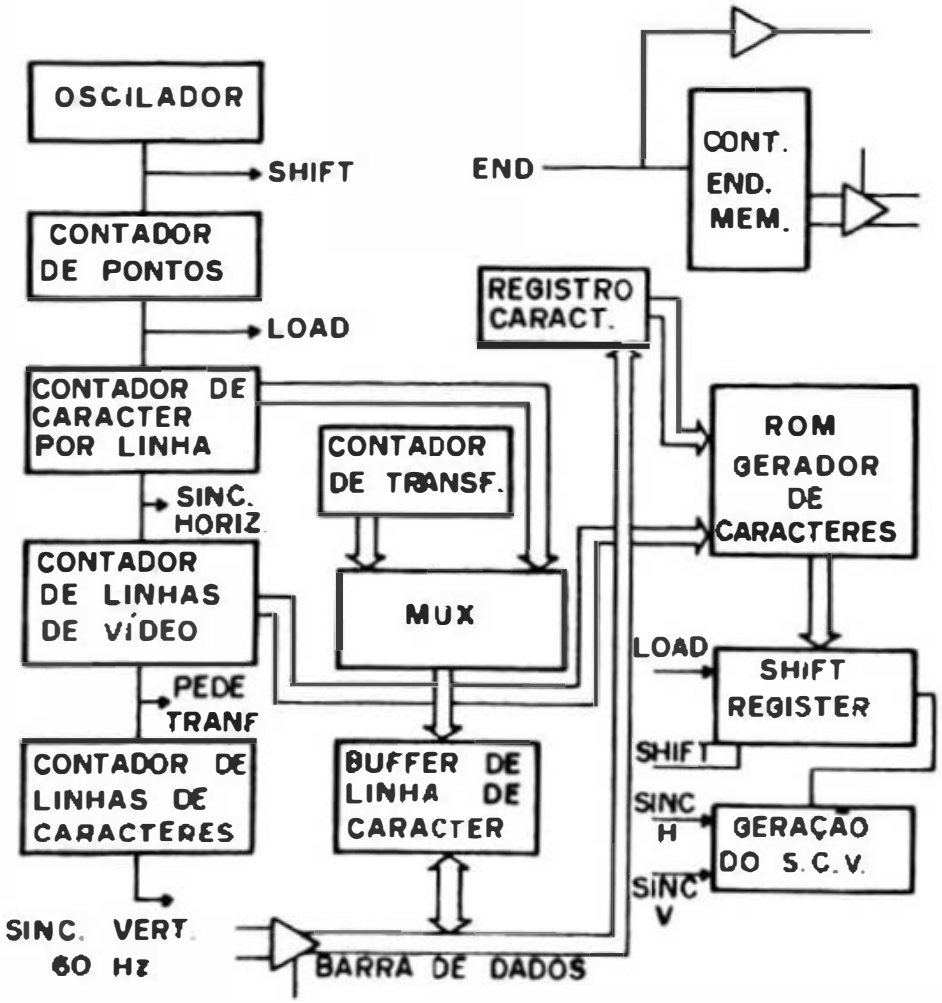


FIGURA 06

Vejamos agora algumas características dos terminais de vídeo mais comuns.

CARACTERÍSTICAS DOS TERMINAIS DE USO CORRENTE

A seguir, apresentaremos uma listagem contendo as características mais comuns, encontradas nos terminais de vídeo comercializados no Brasil. Vejamos:

- Tela de 12" (polegadas) medida na diagonal;
- Comunicação síncrona até 19.200 bps (bits por segundo);
- Comunicação assíncrona até 9.600 bps;
- Saída opcional para impressora;
- Número de caracteres por linha: de 16 a 80;
- Número de caracteres por coluna: de 4 a 25;
- Interface loop de corrente;
- Interface EIA RS 232-C ou CCITT-V24;
- Teclado, tipo máquina de escrever;
- Teclado, tipo "data entry";
- Apresentação de dados no vídeo em 6 modos diferentes:
  a) Normal;
  b) Invisível;
  c) Intensificado;
  d) Reverso;
  e) Sublinhado;
  f) Piscante.

APLICAÇÕES

Os terminais de vídeo são tipicamente utilizados para comunicação remota interativa. Os seguintes modos servem de exemplo:

a) Consulta Simples: trata-se de pequenas mensagens de entrada e saída;

b) Consulta Complexa: neste caso, teremos pequenas mensagens de entrada e grandes mensagens de saída;

c) Consulta com Atualização: teremos, neste caso, pequenas mensagens de entrada e grandes mensagens de saída, sendo que parte dos dados mostrados pode ser modificada pelo operador;

d) Consulta Conversacional: uma progressão de mensagens de entrada e saída completa uma conversação;

e) Consulta Conversacional com Atualização: trata-se de uma combinação das aplicações descritas nos itens c e d;

f) Entrada de Dados: grandes mensagens de entrada e pequenas mensagens de saída.

De modo geral, podemos dizer que a interação do terminal de vídeo com o operador é feita de duas formas: uma por intermédio de um teclado, no qual o operador faz a entrada de dados e programas ou, então, mantém um processo conversacional. E outra, feita por intermédio de uma unidade de vídeo (monitor), na qual o operador obtém a resposta de suas consultas e faz a verificação dos dados e/ou programas digitados.

Visando dar mais conforto ao operador, de modo que ele sinta maior satisfação pelo serviço, que realiza e, conseqüentemente, aumente a sua produtividade, faz-se necessário uma adequação do projeto do terminal de vídeo, bem como das instalações onde este será instalado.

Nunca o ambiente, no qual será instalado o terminal de vídeo, poderá ser negligenciado, em termos de iluminação, mas também não deverá ficar exposto à iluminação excessiva. Podemos observar que, no primeiro caso, o operador não conseguiria ler seus manuscritos e originais, enquanto que, no segundo, o operador seria seriamente prejudicado pelos reflexos causados pelo excesso de iluminação e, nos dois casos, iríamos constatar uma baixa produtividade no trabalho.

Um outro ponto muito importante a ser levado em consideração é o posicionamento do operador. A estética e a beleza dos móveis não é o fator mais importante quando da escolha do mobiliário da sala de operação do terminal de vídeo. Primeiramente, deve-se pensar em termos de conforto para o operador de modo a evitar ao máximo a fadiga e o mal estar durante a jornada de trabalho. Por exemplo, a cadeira deve ser adequada ao trabalho executado, pois se utilizarmos um banco qualquer sem recosto, tornaremos o trabalho extremamente cansativo (veja figura 7).

FIGURA 07

Por outro lado, uma cadeira bem projetada proporcionará ao operador um conforto maior e, conseqüentemente, obteremos, via de regra, uma maior produtividade (veja figura 8).

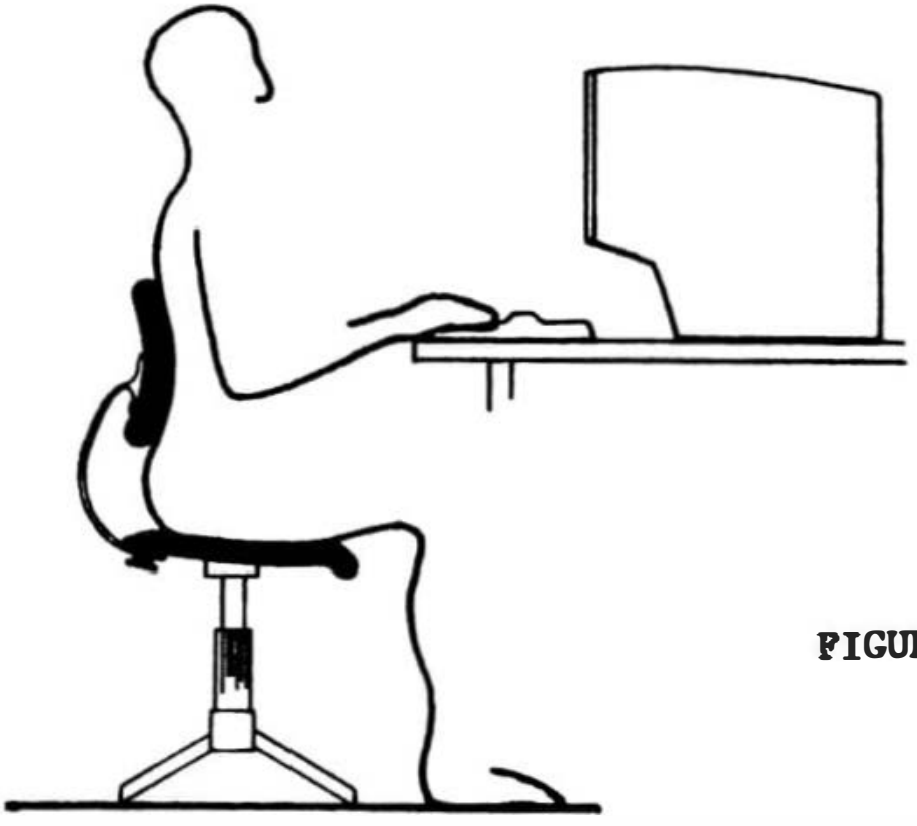


FIGURA 08

Com relação à postura do operador na cadeira, podemos considerar as seguintes observações:

- A cabeça não deve estar excessivamente dobrada, mas sim, formando uma curva côncava com relação ao pescoço;

- A espinha dorsal deve formar uma curva convexa na região do tórax;

- O braço e o antebraço devem formar um ângulo de aproximadamente 90^{o}, com o braço quase que na vertical;

- Na região lombar, a espinha deve formar uma curva côncava;

- Na parte inferior da coxa, não deve atuar nenhum tipo de pressão;

- Os pés devem estar sobre o chão ou no descanso próprio para os pés.

O teclado do terminal de vídeo também deve ser adequado ao operador. Por exemplo, para que o trabalho de digitação se faça com o menor esforço possível, por parte do digitador, o teclado deve receber teclas com diversos dispositivos já adaptados, de modo que a um leve toque elas proporcionem a emissão do seu código para o sistema lógico do terminal. Um outro aspecto que deve ser observado com relação ao teclado, é o apoio para as mãos, pois sem este cuidado o trabalho se tornaria muito cansativo em curto espaço de tempo (veja figura 9).

Um outro ponto interessante é a velocidade crescente com que o operador digita as informações, fazendo com que nos teclados convencionais (nos que o operador deve apertar uma tecla de cada vez) o operador possa cometer erros por excesso de eficiência. Para solucionar este tipo de problema, foi desenvolvido um novo tipo de teclado, o qual recebe o nome de "N KEY ROLL-OVER"; este teclado faz com que o código de cada tecla seja emitido assim que esta seja pressionada, independentemente, do fato de outras teclas estarem pressionadas no mesmo instante. Além desses fatores, no teclado as legendas devem ser claras e objetivas.

Ainda, com relação à questão conforto, deve-se evitar que o operador fique exposto às correntes de ar, utilizadas inter

namente para refrigeração dos equipamentos. Devemos ainda alocar a unidade de teclado independentemente da unidade de vídeo, possibilitando, desta forma, uma melhor organização do trabalho por parte do operador (ou digitador).

O terminal de vídeo irá interagir com o meio externo, na forma de emissão ou recepção de informações. Estas informações, como sabemos, são de forma digital e podem ir ou vir diretamente do computador ou ainda através de um "modem", fazendo a ligação por teleprocessamento.

FIGURA 09

A figura 10 nos mostra alguns aspectos de um terminal de vídeo, válidos tanto para minicomputadores como para microcomputadores.

FIGURA 10

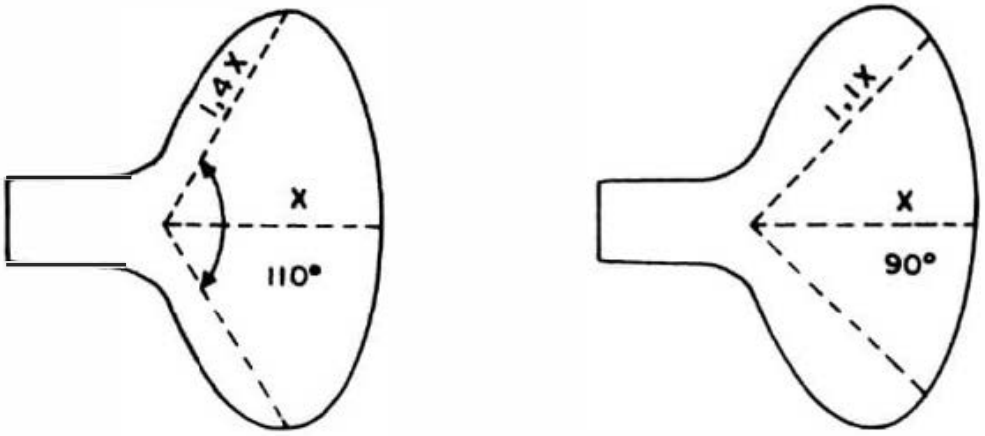
CARACTERÍSTICAS DO VÍDEO (TUBO DE RAIOS CATÓDICOS)

Quando da escolha do tubo, deve-se levar em consideração dois fatores muito importantes: o tamanho e o ângulo do tubo. O tubo deve ser suficientemente grande para podermos mostrar todos os caracteres que desejamos, de forma clara e legível. O segundo fator, o ângulo do tubo, este deve ser levado em consideração devido às distorções que são introduzidas. Por exemplo, a distância percorrida pelo feixe eletrônico varia de um tubo de 90^{o}, para um tubo de 110^{o}, o que ocasiona distorção de linearidade (veja figura 11).

Uma outra característica, também muito importante e que deve ser observada na hora da escolha, é o fósforo utilizado no tubo. Nos receptores de televisão comuns, a persistência não pode ser grande, pois isto resultaria em imagens manchadas. Porém, nos terminais de vídeo, isto não ocorre, pois a imagem é praticamente constante, portanto, para este caso, podemos ter um grau de persistência maior. Outro fator que tam-

bém deve ser considerado é a cor da luz emitida.

FIGURA 11



Tubos de ráios catódicos de 90^{o} e 110^{o}.

A figura 12 nos mostra uma tabela comparativa entre diversos tipos de fósforo.

TIPOS DE FÓSFORO

Tipo de Fósforo	Cor	Tempo de "Decay" (em milissegundos)	
		Para 10% de brilho	Para 1% de brilho
P_1	Verde	24	50
P_4	Branca	0,15	0,90
P_{20}	Amarela	0,5	16
P_{22G}	Verde	6	16
P_{22B}	Azul	4,8	17
P_{22R}	Vermelha	1,5	23
P_{31}	Verde	7	19
P_{38}	Laranja	1100	6000
P_{39}	Verde	400	1300
P_{45}	Branca	1,5	5,2

O fósforo tipo P_4 é comumente utilizado nos televisores em preto e branco. Já os fósforos tipo P22G, P22B e P22R são comumente utilizados nos tubos de raios catódicos de televisores a cores.

Um aspecto ergonômico, com relação ao vídeo, que também deve ser considerado é o problema da ambientação da iluminação da tela. Normalmente, os terminais de vídeo possuem um controle de brilho sobre o qual o operador tem acesso, possibilitando, desta forma, que cada pessoa (operador) trabalhe com um nível de iluminação de tela que lhe seja agradável. A figura 13 nos mostra um texto exibido em uma tela com alto brilho e com o vídeo invertido.

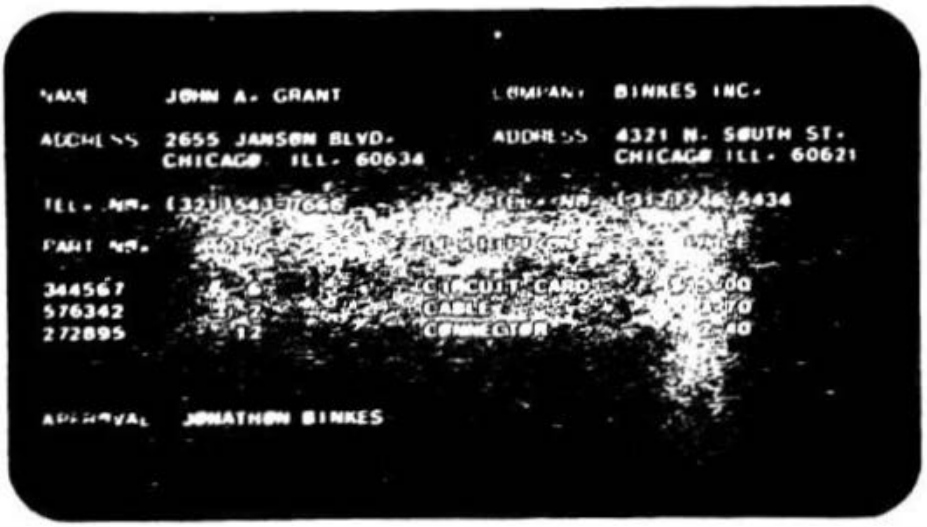


FIGURA 13

EXIBIÇÃO COM "BRILHO" E "PULSAÇÃO"

Estas características de exibição são utilizadas para chamar a atenção dos segmentos (ou das partes) de uma exibição de significado particular. Para que um segmento brilhe em uma exibição em preto e branco normal (ON durante 1/2 seg. e OFF durante 1/2 seg.) o tubo de raios catódicos deverá receber e interpretar um comando, vindo diretamente do programa. Para brilhar, o comando deverá reduzir o sinal de vídeo para 1 volt (preto) por 1/2 segundo e, em seguida, elevar o sinal de vídeo para 3 volts para o próximo período de 1/2 segundo.

No caso da pulsação, o tubo de raios catódicos interpreta um comando do programa pulsante como sendo um comando para iluminar, mas ao invés de cortar o sinal de vídeo do caracter, escurecendo-o, o que equivaleria a levar o sinal de vídeo para o nível de 1 volt, este é cortado somente para um nível cinza intermediário, equivalente a 2 volts.

TAMANHO DO CARACTER

A escolha do tamanho do caracter é feita com base na legibilidade dos caracteres na tela. As pequenas telas requerem caracteres maiores. Os aparelhos de televisão, convertidos, usados para exibição de vídeo, também requerem caracteres maiores, o que é justificado com base em duas razões: primeiro, sendo a tela pequena, os caracteres tendem a unir-se horizontalmente, desde que o sinal de vídeo ("preto e branco") seja cortado rigidamente, provocando o aparecimento de uma cor cinza intermediária ao redor dos caracteres, o que é facilmente superado com o uso de caracteres maiores, tornando a cor cinza menos notada. Em segundo lugar, pode ocorrer também o preenchimento de alguns pontos nas linhas horizontais. Por exemplo, as partes horizontais da letra E pode não aparecer como cinco pontos separados (em uma matriz 5x7), mas sim, como uma linha contínua. Porém, se utilizarmos um caracter maior (uma matriz 7x9) fica mais fácil distinguir os pontos que constituem as partes horizontais da letra em questão. A figura 14 nos mostra alguns formatos de caracteres.

MONITORES DE VÍDEO COLORIDOS

As exibições no vídeo a cor utilizam três canhões eletrônicos (um para a cor vermelha, um para a cor verde e um para a cor azul) ao invés de um único canhão eletrônico, como é o caso dos monitores em preto e branco. Cada um desses canhões eletrônicos iluminam somente os pontos vermelhos, azuis e verdes respectivamente. As combinações dos pontos vermelhos, azuis e verdes, adjacentes é que darão origem às cores intermediárias. Basicamente, três sinais "OFF e ON" são enviados para o monitor de vídeo, um para o canhão vermelho, outro para o canhão azul e

um terceiro conjunto de sinais para o canhão verde. Por outro lado, o sistema gráfico, empregado em um monitor de vídeo colorido é o mesmo daquele utilizado para um monitor preto e branco.

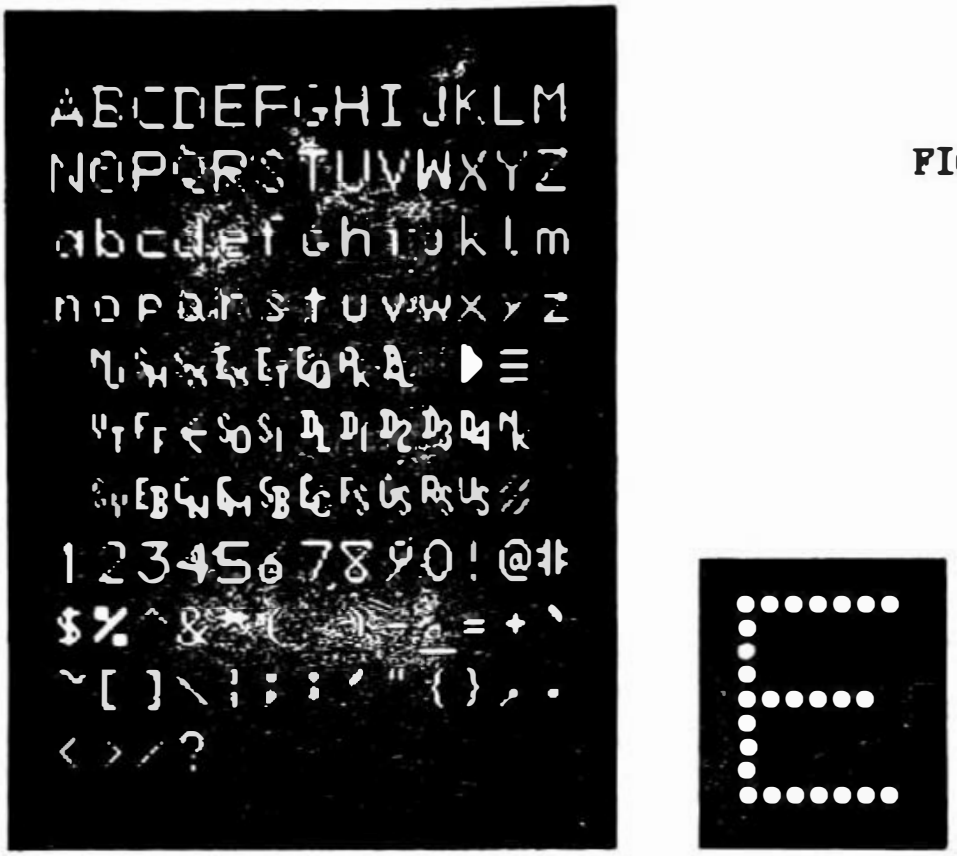


FIGURA 14

A utilidade de um monitor de vídeo colorido é maior quando houver alguma informação adicional que o fator cor pode fornecer, além da capacidade de uma exibição em preto e branco. Em se tratando de exibição de textos, o monitor de vídeo colorido não apresenta nenhuma vantagem especial, uma vez que o mesmo texto pode ser exibido em um terminal em preto e branco. Porém, em se tratando de figuras complexas, como, por exemplo, partes de um controle de processo com múltiplos operadores de canais, o monitor colorido pode separar os canais sobrepostos, usando uma determinada cor para cada um deles. Nestes tipos de aplicações, os monitores de vídeo coloridos apresentam-se muito mais vantajosos que os monitores preto e branco.

ADAPTANDO OS RECEPTORES DE TV

O monitor de vídeo é semelhante a um receptor de TV, exceto que este não possui o estágio detector de rádio-freqüência e nem os circuitos demoduladores necessários aos aparelhos de TV para receberem e decodificarem os sinais de radiodifusão da TV. Porém, os monitores de vídeo, comumente, possuem um controlador eletrônico para a geração de caracter e armazenagem de dados, além de outras funções relacionadas com os computadores.

Os sinais do computador (ou mini e microcomputador) utilizados para a geração de texto e gráficos para serem exibidos nos monitores de vídeo (do terminal), podem também serem enviados para um receptor de TV comum. Para tanto, é necessário ligarmos um gerador de rádio-freqüência, comercial, para criar um sinal de radiodifusão para um transmissor de TV. A saída deste circuito é então conectada diretamente aos terminais da antena do receptor de televisão. Muito embora este método seja aceitável, a disposição não é tão clara como em um monitor de vídeo, devido a problemas da "banda de passagem" e a necessidade de se converter o sinal para radio-freqüência e, em seguida, convertê-lo, novamente, em um sinal de vídeo regular.

Uma outra maneira de utilizarmos o receptor de TV como monitor de vídeo seria ligarmos o sinal de saída de vídeo do computador ao receptor de TV, diretamente atrás do terminal frontal de rádio-freqüência do receptor de TV. Este método também é viável, porém requer o emprego de um transformador de isolação para evitar a possibilidade de um choque elétrico.

COMENTÁRIOS FINAIS

A utilização dos terminais de vídeo, com o grande avanço tecnológico do setor, tornaram-se amplamente difundidos, tanto é que, praticamente, todos os sistemas de computação possuem um terminal de vídeo acoplado à UCP.

Mais inovações estão por vir, principalmente no que se refere aos tubos de raios catódicos, os quais, dependendo da aplicação, poderão ser substituídos por painéis de plasma.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-21

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Os formatos normalizados, mais comuns, utilizados na representação de caracteres alfanuméricos são:

   a) Formato 5x7;

   b) Formato 7x9;

   c) Formato 9x12;

   d) Os itens a e b estão corretos.

2) O formato 7x9 apresenta:

   a) maior resolução;

   b) menor resolução;

   c) maior cintilação;

   d) N.R.A.

3) O formato 5x7 apresenta uma resolução aceitável para monitores de vídeo que utilizam:

a) aparelhos de rádio comum;

b) aparelho transmissor de radio-freqüência;

c) aparelho receptor de televisão comum;

d) N.R.A.

4) Em um terminal alfanumérico, podemos representar pelo menos:

a) 32 tipos diferentes de caracteres;

b) 64 tipos diferentes de caracteres;

c) 16 tipos diferentes de caracteres;

d) N.R.A.

5) Como parâmetro de entrada, os geradores de caracteres utilizam:

a) o código ASCII;

b) o código BCD 8421;

c) o código Gray;

d) N.R.A.

6) O número mínimo de varreduras necessárias para podermos representar uma linha de texto de n caracteres, formados por uma matriz 5x7, é:

a) 16;

b) 8;

c) 5;

d) 1.

7) Normalmente o cursor se encontra situado:

a) dentro da primeira linha do texto representado;

b) dentro do primeiro caracter do texto representado;

c) dentro da última linha do texto representado;

d) N.R.A.

8) O refrescamento da tela do monitor de vídeo consiste em re escrever nela, todo o texto a ser visualizado, aproximadamente, a cada período de:

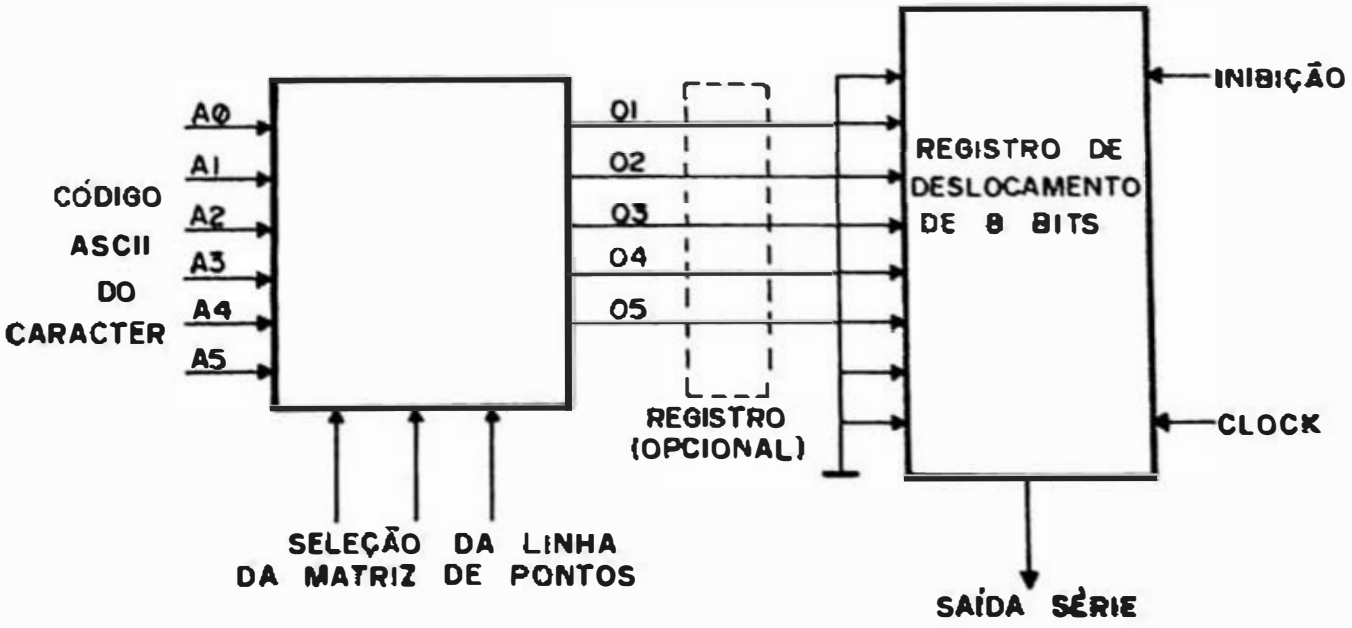
a) 5 ms;

b) 10 ms;

c) 15 ms;

d) 20 ms.

9) A figura a seguir representa um circuito:



a) serealizador de informação;

b) gerador de caracteres;

c) controlador de vídeo;

d) N.R.A.

10) Em um terminal de vídeo comum, o número de caracteres por linha está entre:

a) 16 e 100;

b) 32 e 50;

c) 16 e 80;

d) 4 e 25;

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANA

LIÇÃO MC-22

MICROCOMPUTADOR CEDM-80

INTRODUÇÃO

Iniciaremos, a partir desta lição, a apresentação do Kit do microcomputador CEDM-80, o qual, na sua configuração básica, foi projetado de modo a proporcionar ao aluno uma noção bem clara e objetiva das partes que compõem um microcomputador, bem como o seu funcionamento.

Como acabamos de dizer, o microcomputador CEDM-80 é puramente didático e, com vistas a uma implementação de sua configuração básica, foi projetada uma série de opcionais para o Kit.

Tanto nesta lição como nas lições seguintes desta série, procuraremos tratar do assunto de forma objetiva, de modo a propiciar ao aluno condições de visualizar o funcionamento do microcomputador CEDM-80. Nestas lições, trataremos mais dos assuntos relativos ao hardware, sendo que a última lição, desta série de três, será dedicada aos aspectos de montagem do Kit.

Para iniciar o estudo do microcomputador CEDM-80, apresentaremos, a seguir, as principais características do Kit CEDM-80 com relação ao hardware e ao software.

CARACTERÍSTICAS DO MICRO CEDM-80 (HARDWARE)

Vejamos então algumas especificações referentes ao hardware do nosso Kit:

1 - CPU (Unidade Central de Processamento): em nosso projeto, optamos pela utilização do microprocessador Z-80 com 158 instruções e freqüência máxima de clock igual a 2,5 MHz. Pode-se utilizar também o Z-80 A, pois a freqüência do clock do sistema é de 1,79 MHz.

2 - RAM (Memória de acesso aleatório): optamos pela memória RAM estática "6116" de 2 K x 8 bits. Endereço da memória RAM: 1800 - 1FFF.

3 - ROM (Memória de apenas leitura): optamos pela utilização da memória EPROM "2716" de 2 K x 8 bits. Endereço da memória EPROM do monitor: 0000 - 07FF.

4 - Expansão da capacidade de memória: a área reservada para expansão no cartão da CPU é de 2 K bytes. Nesta expansão, poderá ser utilizada um único chip de memória RAM (6116) ou EPROM (2716).

5 - Porta programável de entrada e saída (PPI): optamos pela utilização do PPI 8255, o qual possui um total de 24 linhas paralelas de entrada e saída. Endereço de entrada e saída: 00-03. Das 24 linhas paralelas de E/S (entrada/saída), utilizaremos apenas 16 linhas, cujos endereços de E/S vão desde 80H até 83H (Hexadecimal).

6 - Conector para opcionais: está previsto um "circuito contador de tempo" (CTC), programável e um "controlador paralelo de entrada/saída" (PIO). Este conjunto de chips é opcional.

7 - Display: em nosso projeto está previsto um display composto de 6 (seis) dígitos, sendo que cada dígito é formado por 7 segmentos.

8 - Teclado: o teclado projetado possui um total de 36 teclas, sendo que 16 teclas são destinadas ao sistema de numeração hexadecimal, 19 teclas destinadas às funções e 1 tecla será definida pelo usuário através de software.

9 - Interface para gravador cassete: este circuito permite a gravação e reprodução de programas em qualquer gravador cassete, sendo que a velocidade de transferência de dados deste sistema é igual a 165 bps (Opcionais).

10 - Alimentação: o microcomputador CEDM-80 é alimentado por uma tensão de +5V/±600 mA. Para tanto poderá ser utilizado como fonte de alimentação o Kit CEDM-78.

11 - Sistema monitor e "passo a passo": este sistema permitirá ao aluno, ao pressionar a tecla PASSO, executar programas instrução por instrução, bem como verificar o funcionamento do programa, modificando memória e/ou os registros antes de executar o mesmo.

CARACTERÍSTICAS DO MICRO CEDM-80 (SOFTWARE)

O microcomputador CEDM-80 possui um programa monitor de alta performance de 2 Kbytes. Assim que o microcomputador é ligado à tomada de força, o programa monitor começa a funcionar. Além de todas as funções constantes do teclado, o programa monitor possui também uma rotina de verificação de memória. Na lição seguinte, falaremos mais sobre o teclado e apresentaremos, posteriormente, uma descrição das funções das teclas.

DIAGRAMAS DE BLOCOS DO MICROCOMPUTADOR CEDM-80

O diagrama de blocos do microcomputador CEDM-80 é mostrado na figura 1. Com base nesta figura, podemos observar que o diagrama de blocos se encontra dividido em duas partes distintas:

1ª - bloco da CPU;
2ª - bloco de periféricos.

Esta divisão do diagrama de blocos, feita por nós, visa facilitar o seu estudo.

Nesta lição, estudaremos apenas o bloco referente à CPU, o qual é constituído das seguintes partes:

a) Gerador de clock;
b) Divisor por dois (÷ 2);
c) Circuito de reset;
d) Sistema "passo a passo" e monitor;
e) Unidade central de processamento;
f) Bloco de memória.

ESTRUTURA SIMPLIFICADA DO MICROCOMPUTADOR CEDM-80

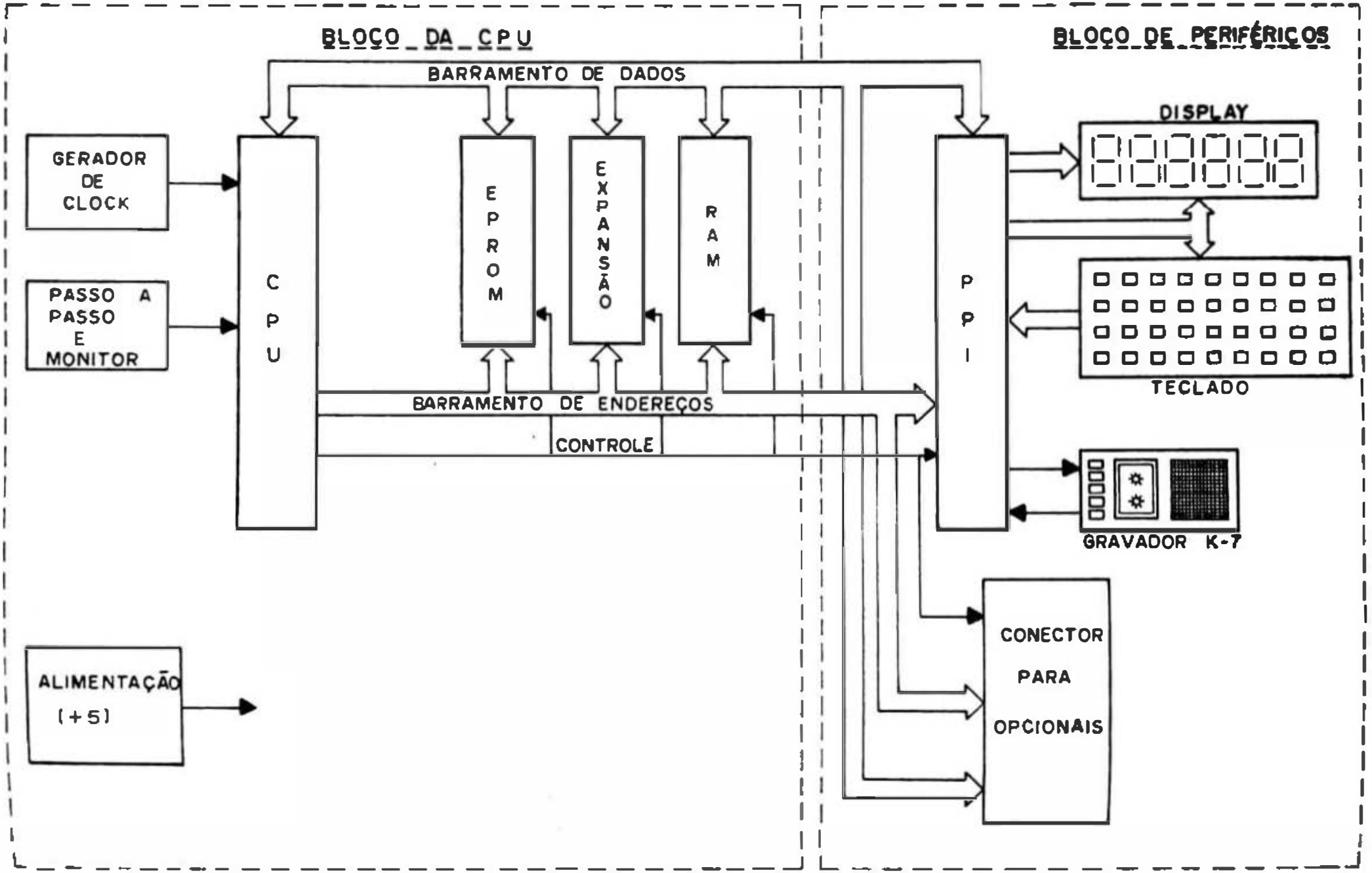


FIGURA 01

Todas essas partes serão apresentadas de maneira prática e objetiva, tanto do ponto de vista descritivo, como a nível de componentes.

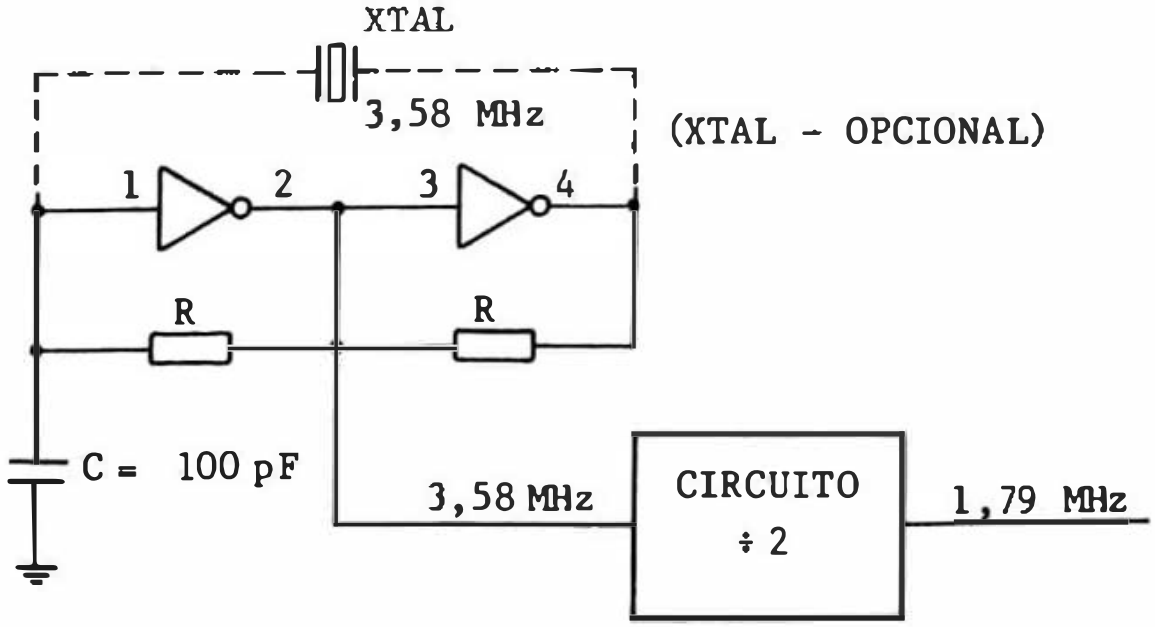
Antes, porém, de iniciarmos o nosso estudo, cabe observar que doravante designaremos o microcomputador CEDM-80 por micro CEDM-80 ou, simplesmente, por CEDM-80.

Passemos então ao estudo de cada uma das partes que compõem o bloco da CPU.

a) Gerador de clock:

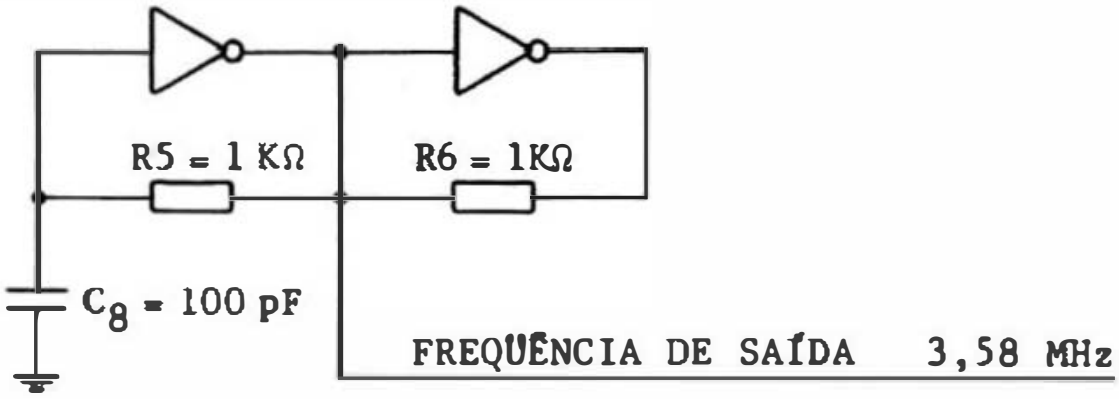
Como sabemos, o microprocessador Z-80 necessita de um clock de uma só fase para poder operar, diferindo, assim, do 8080A. A freqüência de clock máxima permitida para o Z-80 é de 2,5 MHz (o Z-80A funciona com 4 MHz), o que nos induziu a adotar uma freqüência de clock um pouco menor para a CPU. O circuito gerador de clock foi desenvolvido de modo a fornecer uma freqüência máxima igual a 3,58 MHz. Note que esta freqüência é superior à máxima freqüência de clock admissível pelo Z-80. Para resolver este problema, decidimos adotar uma freqüência de clock para o sistema igual a 1,79 MHz, bastando, para tanto, dividirmos a freqüência de saída do gerador de clock por dois (2). A figura 2 nos mostra uma possível configuração do circuito gerador de clock.

FIGURA 02



O circuito oscilador controlado a cristal, mostrado na figura 2, torna-se útil a programas, nos quais o tempo de execução é preciso. No nosso caso, especificamente, poderemos eliminar o cristal (XTAL) do circuito, uma vez que o mesmo é utilizado apenas para se estabilizar a freqüência. O que nos garante a "geração" de uma freqüência em torno de 3,58 MHz é o oscilador formado pelo resistor R_5 de 1 KΩ e pelo capacitor C_8 de 100 pF (malha R.C). A figura 3 nos mostra o circuito gerador de clock, empregado no micro CEDM-80.

FIGURA 03



b) Divisor por 2 (÷ 2):

O bloco divisor por 2 é constituído exclusivamente por um flip-flop tipo D modelo 74 LS 74. A figura 4 nos mostra a configuração básica deste circuito.

A divisão por 2, como já dissemos, faz-se necessária, devido ao fato de que a máxima freqüência de clock admissível pelo microprocessador Z-80 é de 2,5 MHz.

Na saída Q do flip-flop tipo D (pino 5), obteremos a freqüência de saída do circuito gerador de clock, dividida por 2, ou seja, obteremos uma freqüência de 1,79 MHz.

c) Circuito de reset (rearme):

Este circuito utiliza o segundo flip-flop tipo D do chip 74 LS 74. A figura 5 nos mostra um esquema do circuito de reset (RST).

FIGURA 04

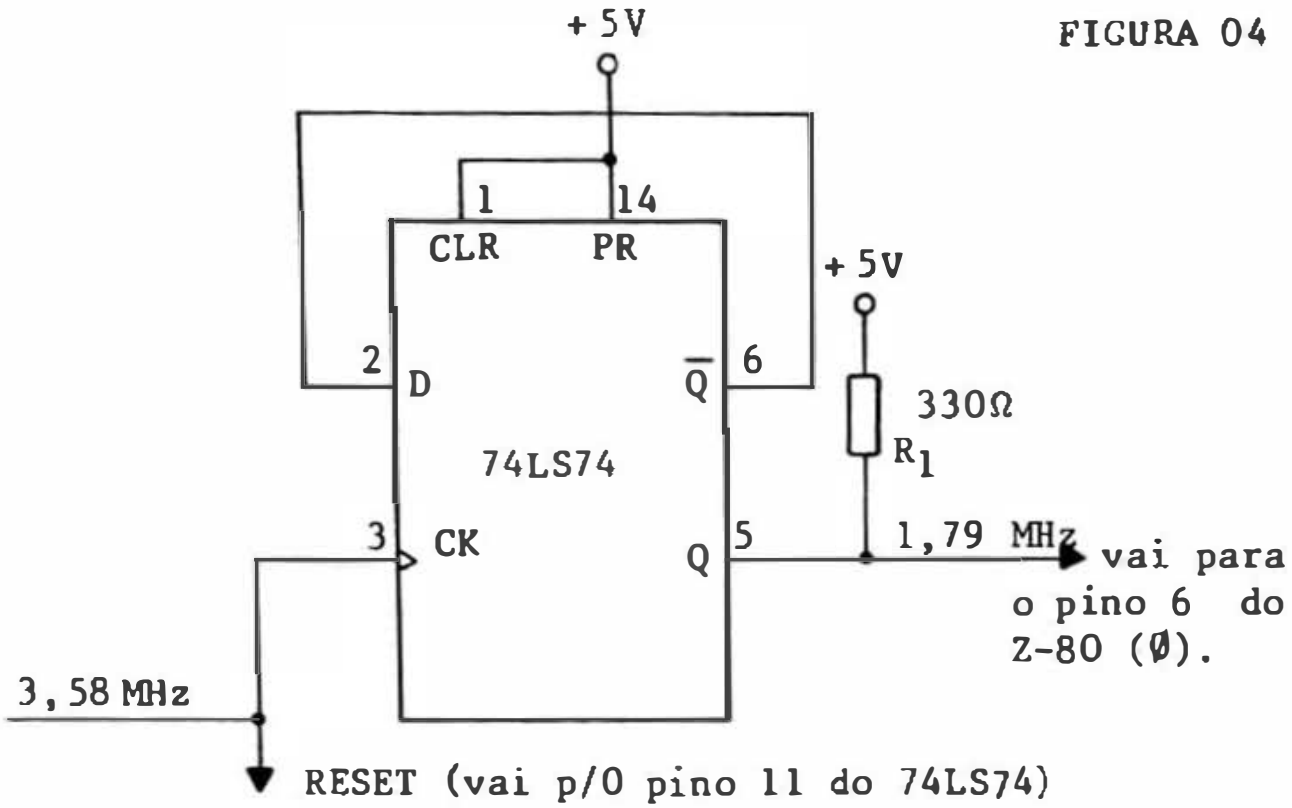
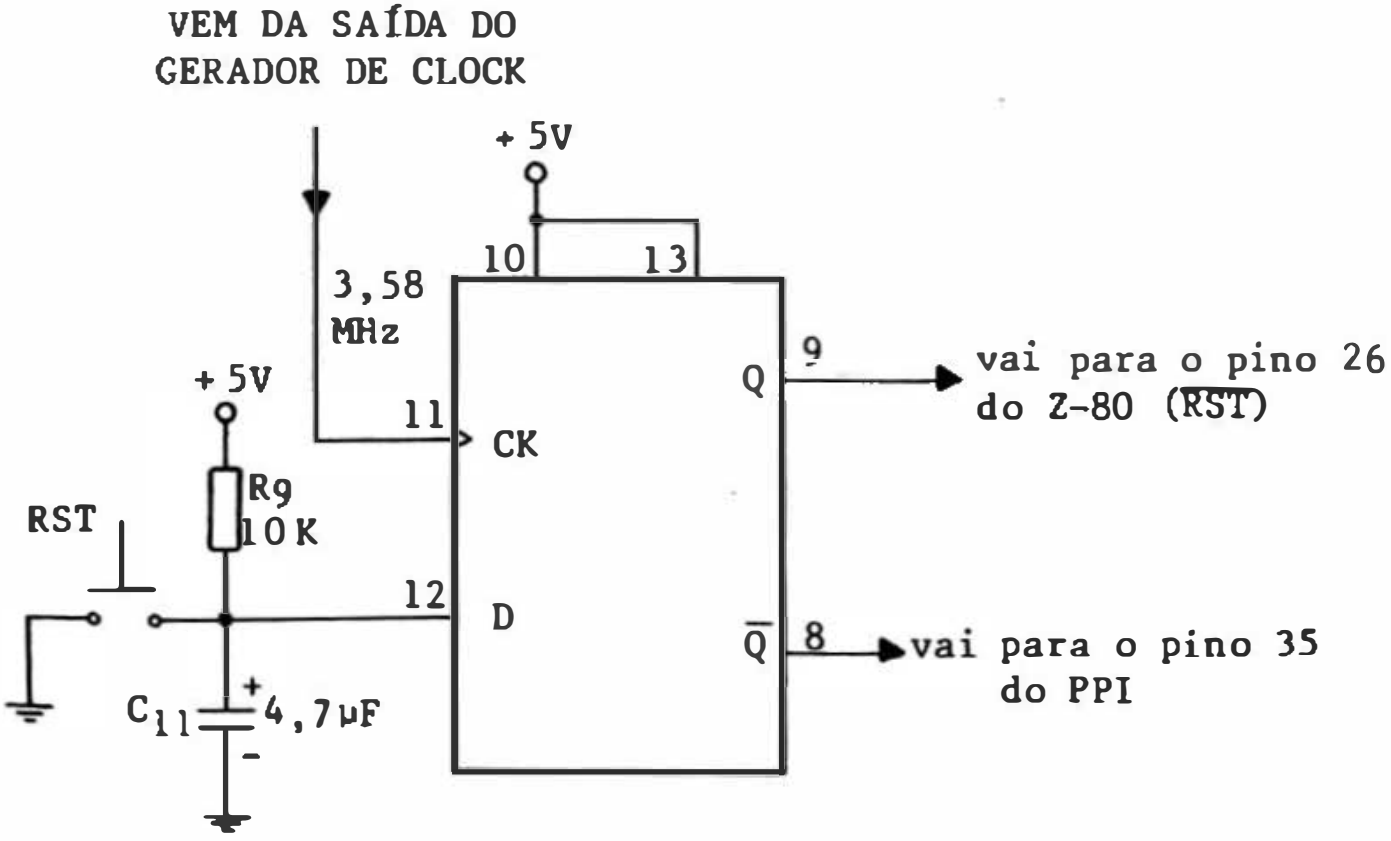


FIGURA 05



A freqüência utilizada pelo circuito de "reset" é de 3,58 MHz.

A função reset é um dos controles mais requisitados num computador, pois permite parar a execução de um determinado programa que se está rodando e zerar o contador de programa, levando-o ao mais baixo endereço de memória $(00)_H$.

O circuito de reset é sincronizado com o dobro do clock da unidade central de processamento (CPU), de modo que este só venha operar no bordo frontal alto de um ciclo de máquina.

Ao pressionarmos a chave RST correspondente à entrada de dados (D) do 74LS74 (pino 12) (o que equivale a levarmos a entrada D a um nível lógico baixo "0"), inicializaremos, simultaneamente, a CPU e o PPI, pois a saída Q sincronizada com o sinal de clock irá para um nível lógico baixo "0", enquanto que a saída $\overline{Q}$ irá para um nível lógico alto "1".

A malha "RC" formada pelo resistor R_9 e pelo capacitor C_{11} (veja figura 5) constitui o que chamamos de sistema de reset (rearme) automático. Este sistema de reset automático é utilizado para se evitar que os espúrios de corrente elétrica, gerados ao se ligar o micro, leve o contador de programa (PC) para uma posição indesejada, por exemplo, XXXX. Sabemos, pois, que ao ligarmos a máquina, o contador de programa (PC) deverá, normalmente, estar na posição 0000 e, para tanto, tem que haver o pulso de reset.

O circuito da figura 5 nos mostra o sistema de reset (rearme) automático e manual, cuja combinação permite ao microcomputador iniciar a execução de um programa qualquer imediatamente após ser ligado.

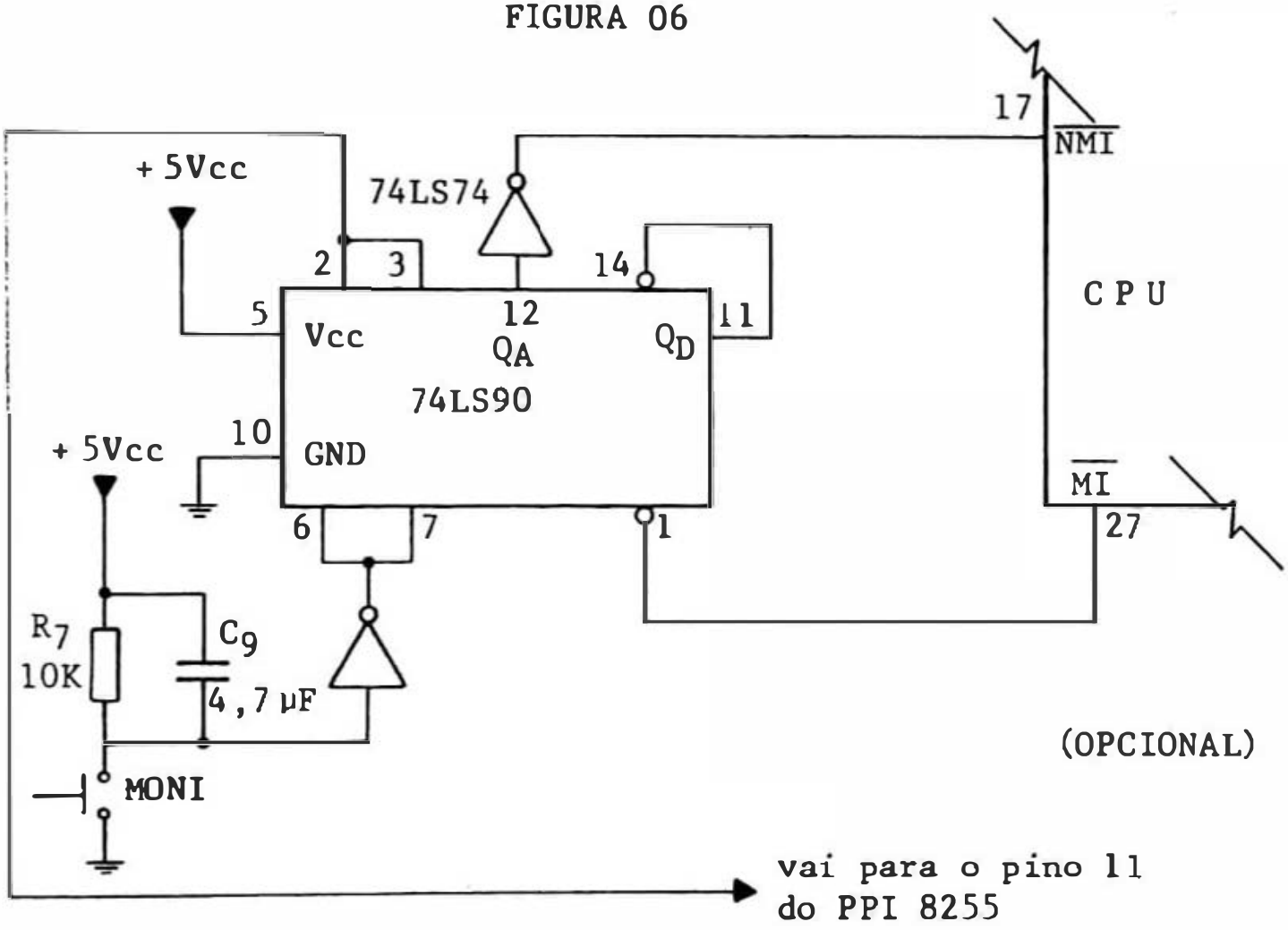
d) Sistema Passo a Passo e Monitor (Opcional):

A figura 6 nos mostra o esquema do circuito "Passo a Passo" e "Monitor". Este sistema é de grande importância pelo fato de facilitar em muito a análise de programas em linguagem de máquina.

O sistema "passo a passo" é comandado através do software interno e acionado por intermédio do PPI-8255 (pino 11), toda vez que a tecla PASSO for pressionada. Quando tal fato

ocorrer, serão visualizados no display os dados e endereços de uma instrução do programa, que está sendo executado naquele dado instante. Se, em seguida, pressionarmos novamente a tecla PASSO a CPU irá então executar a próxima instrução, dando continuidade ao programa. O sistema "passo a passo" nos permite examinar e modificar conteúdos de registro e memória de cada passo.

FIGURA 06



Por outro lado, o sistema monitor, que se encontra junto ao sistema "passo a passo", nos permite interromper o programa em um certo endereço, monitorando, através do display, o endereço da instrução e o dado correspondente no instante em que a tecla MONI foi pressionada. Se pressionarmos novamente a tecla MONI, o programa não seguirá o seu curso normal, a partir do ponto em que houve a interrupção, acusando, assim, um erro no display.

Para se evitar os espúrios decorrentes da abertura ou fechamento da tecla MONI, utilizamos o capacitor C9, ligado em

paralelo com o resistor R7 (veja figura 6). Note que o resistor irá manter a linha do monitor em nível lógico alto "1", garantindo assim o "reset" do sistema "passo a passo" e monitor, pois as entradas 6 e 7 do 74LS90 estarão em um nível lógico baixo "0". O capacitor, por sua vez, irá evitar que os espúrios decorrentes da tecla MONI interfiram no acionamento do sistema, atuando, para tanto, como um filtro.

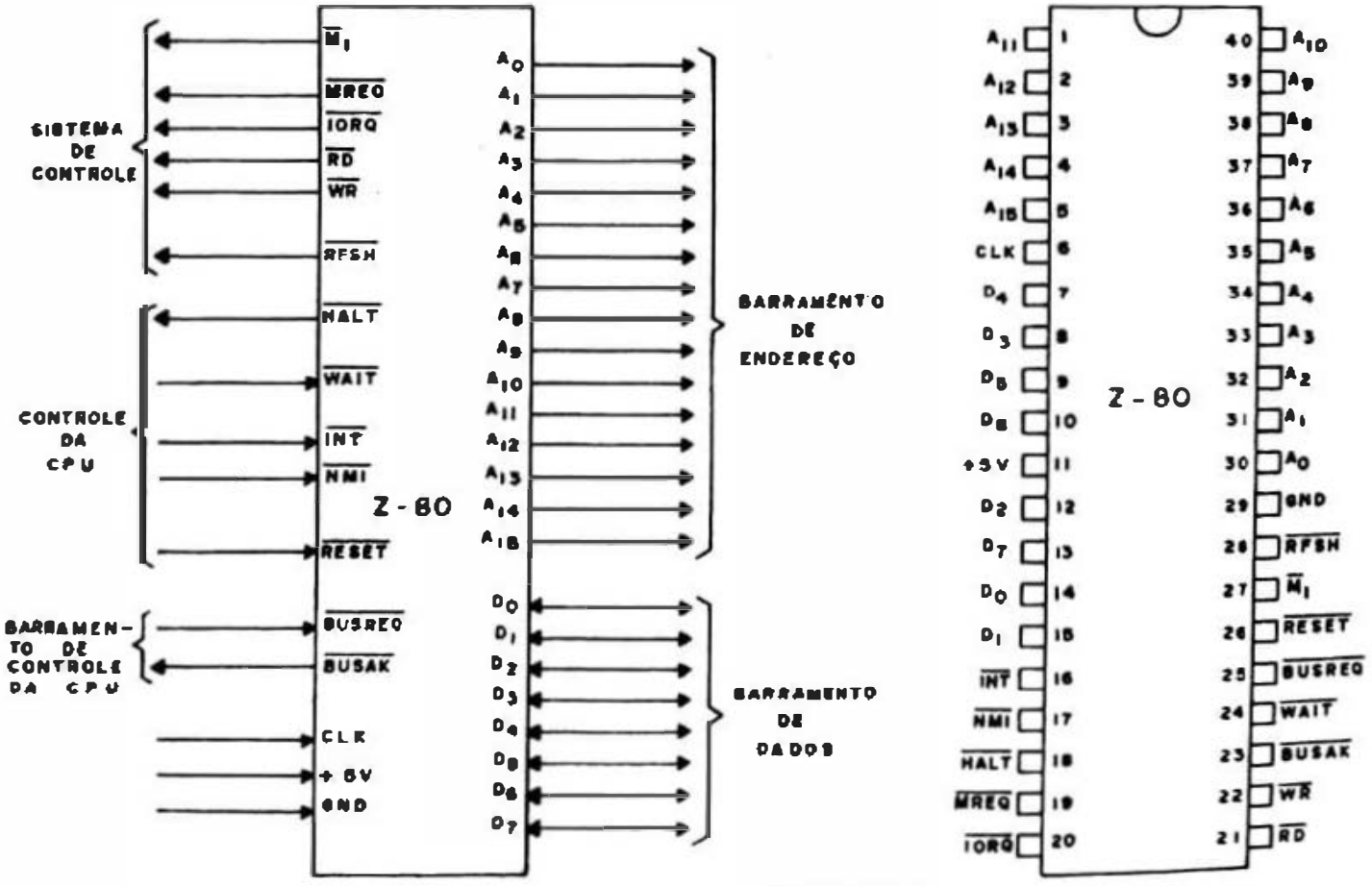
Note que apesar do sistema "passo a passo" e monitor atuarem no modo de interrupção $\overline{NMI}$ da CPU, ambos possuem funções diferentes.

e) Unidade Central de Processamento (CPU):

A unidade central de processamento (CPU) é o principal componente de controle do sistema, cujas funções são:

- Procurar e executar as instruções da memória;
- Armazenar e retirar dados (informações) da memória;
- Armazenar e retirar dados da seção de entrada/saída;
- Comandar todas as funcões do sistema.

FIGURA 07



PINAGEM

A CPU utilizada no micro CEDM-80 é o microprocessador Z-80, fabricado originalmente pela Zilog. As lições MC-12, MC-13 e MC-14 são dedicadas exclusivamente ao estudo do microprocessador Z-80, portanto, aconselhamos ao aluno que reveja essas lições. A figura 7 nos mostra a pinagem do microprocessador Z-80.

A figura 8, por sua vez, nos mostra o diagrama de blocos da arquitetura interna do microprocessador Z-80.

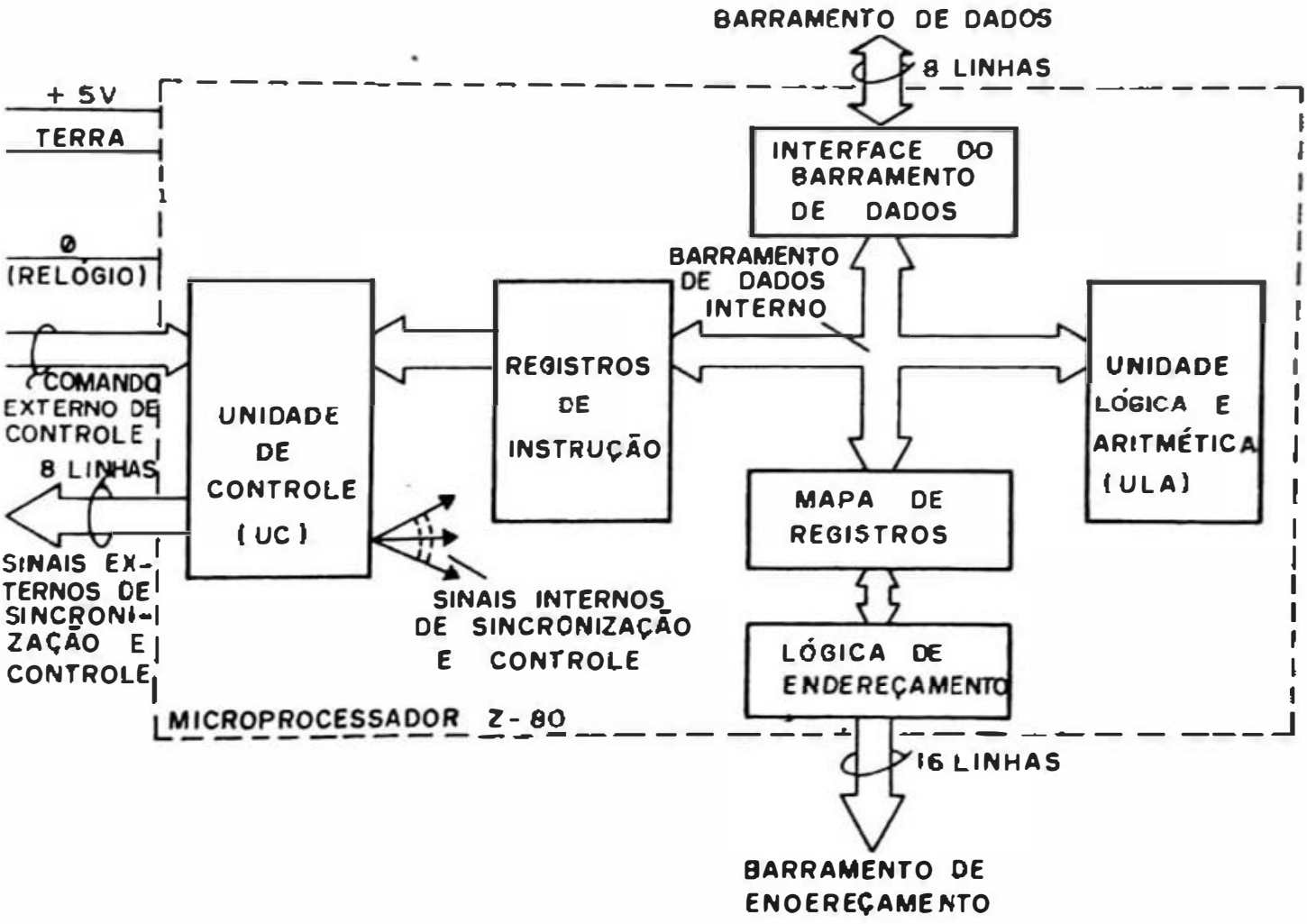


FIGURA 08

Um outro aspecto de vital importância para o nosso estudo relaciona-se ao mapa de registros do microprocessador Z-80, uma vez que o programador tem acesso ao mesmo. A figura 9 nos mostra a organização lógica, estrutural e funcional dos registros do Z-80.

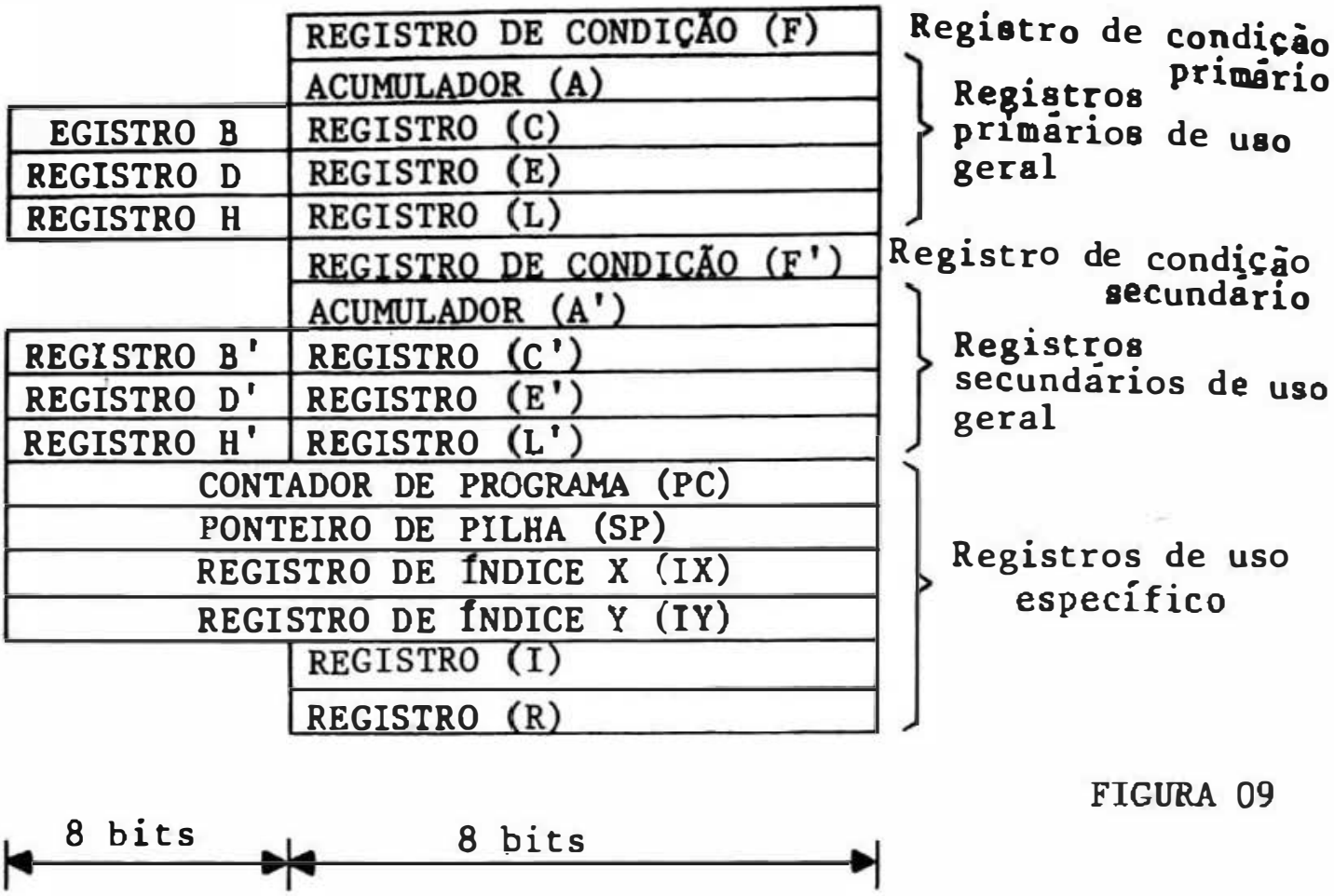


FIGURA 09

Ainda com relação à CPU, é conveniente rever os modos de endereçamento e o conjunto de instruções(256) do microprocessador Z-80 (veja a lição MC-14). Finalizando este tópico, a figura 10 nos mostra o esquema do circuito indicador de instrução "Halt" (76 H), que é opcional.

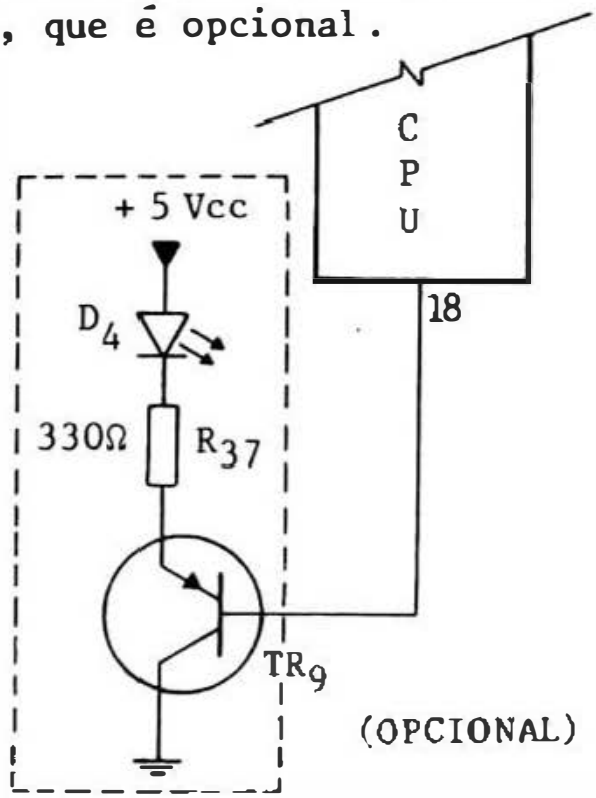


FIGURA 10

O transistor TR_9 funciona como um driver, ou seja, fornece corrente suficiente para o acionamento do led D_4, visto que a corrente enviada pelo pino 18 da CPU é relativamente baixa para acionar a CPU.

f) Bloco de memória:

O bloco referente à memória do microcomputador CEDM-80 é mostrado na figura 11.

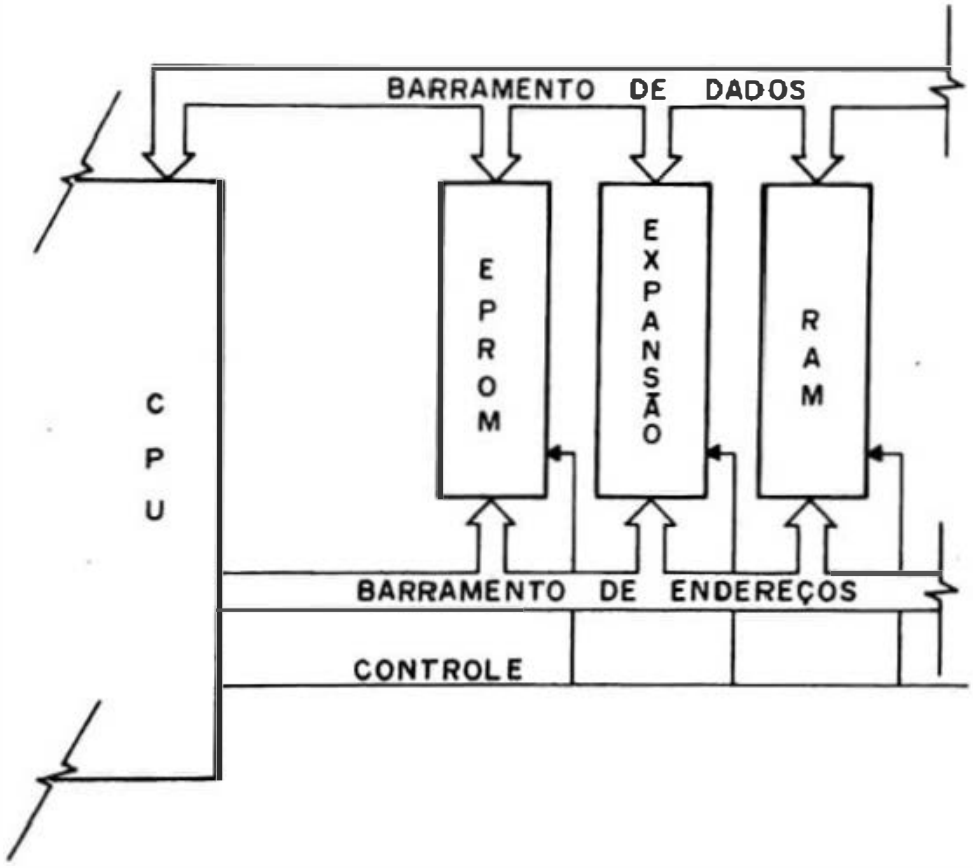


FIGURA 11

No bloco de memória, também chamado de memória externa, temos basicamente dois tipos de memórias: RAM (memória de escrita e leitura) e ROM (memória de apenas leitura). Cabe ressaltar que, dentro da categoria de memórias ROM, existem três subcategorias: ROM, PROM e EPROM - as quais diferem entre si, como já sabemos, pela aplicação e pela utilização. No micro CEDM-80, nós optamos pela utilização da memória EPROM 2716, cujo diagrama de blocos e pinagem são mostrados na figura 12.

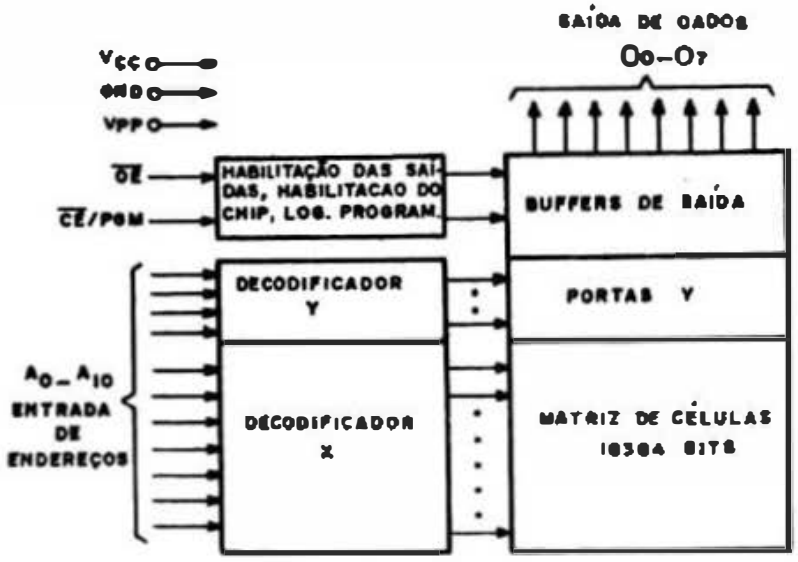
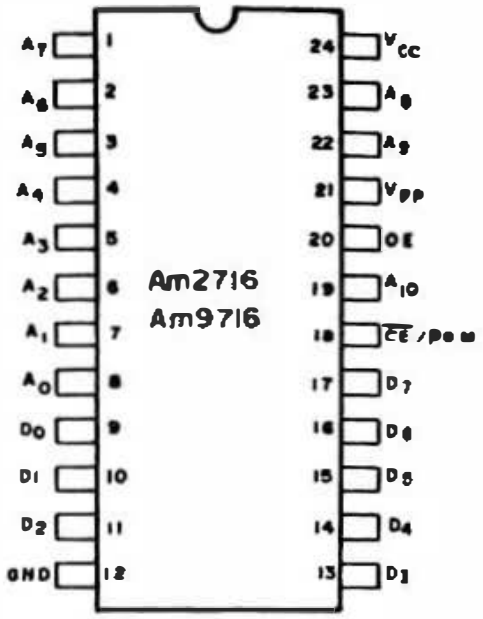


DIAGRAMA DE BLOCOS



PINAGEM

FIGURA 12

A grande vantagem da memória EPROM é que pode ser programada pelo usuário e utilizada como uma memória ROM. Caso o seu conteúdo tiver de ser mudado, ela poderá ser apagada (através de uma fonte de luz ultravioleta) e reprogramada. Cabe ressaltar que, uma vez gravadas as informações na EPROM, estas não poderão ser alteradas pela CPU. Outro fato importante com relação a este tipo de memória é que esta não é volátil, ou seja, mesmo com a fonte desligada, o conteúdo do programa não é perdido.

A memória EPROM 2716 possui uma capacidade de armazenamento de 16 Kbits (2 K x 8) e alimentação única de +5 V.

Com relação à memória RAM, optamos pela memória RAM estática 6116, cuja capacidade de armazenamento é de 16 K bits (2 K x 8) e alimentação única de +5 V. Este tipo de memória RAM não necessita de refrescamento. A figura 13 nos mostra o diagrama de blocos e a pinagem da memória 6116.

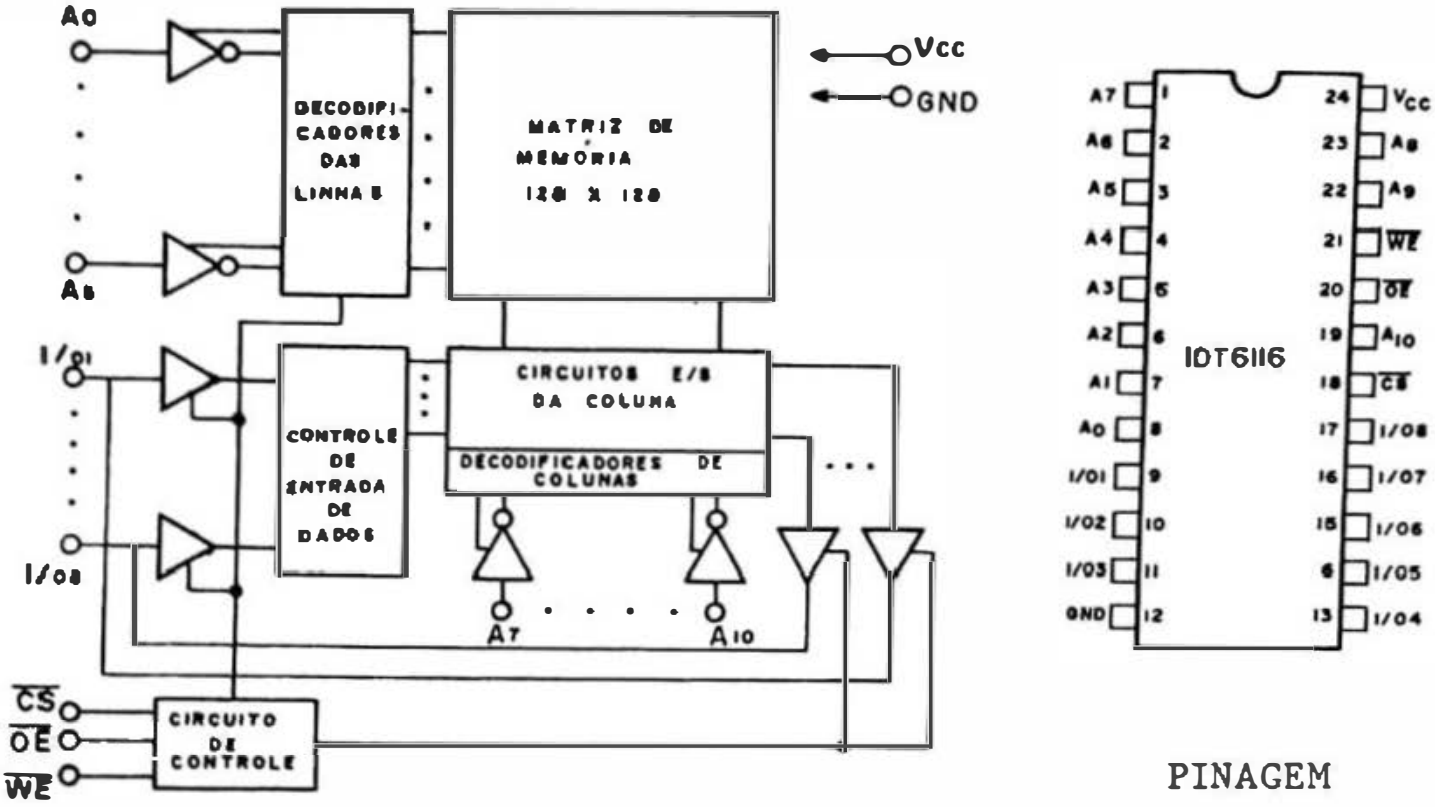


DIAGRAMA DE BLOCOS

PINAGEM

FIGURA 13

A CPU considera as memórias RAM e EPROM como sendo um único bloco, cuja diferenciação será feita pelo hardware, tornando-se de vital importância, tanto sob o aspecto de hardware como de software, o conhecimento do mapa de memória.

O mapa de memória nada mais é do que a divisão racional do bloco de memória em vários setores aos quais a CPU terá acesso. Para o micro CEDM-80, o mapeamento da memória foi elaborado de modo a conter três (3) setores distintos. Com vistas a facilitar o nosso estudo, apresentamos na figura 14 um diagrama esquemático do mapa de memória do micro CEDM-80.

Cabe ressaltar que, para haver leitura ou escrita na memória, é necessária a emissão, pela CPU, de um pulso $\overline{\text{MREQ}}$, ou seja, um pulso de requisição de memória. Caso contrário, tanto a leitura como a escrita não serão possíveis.

Através do mapa de memória, podemos verificar que os endereços que vão do ØØØØ (hexadecimal) até ØFFF estão reserva-

dos para a memória EPROM. Nesta memória estará gravado todo o sistema operacional do micro CEDM-80. O acesso a esta memória (ou à memória RAM) se faz através do decodificador de memória, formado pelos circuitos integrados 74LS139.

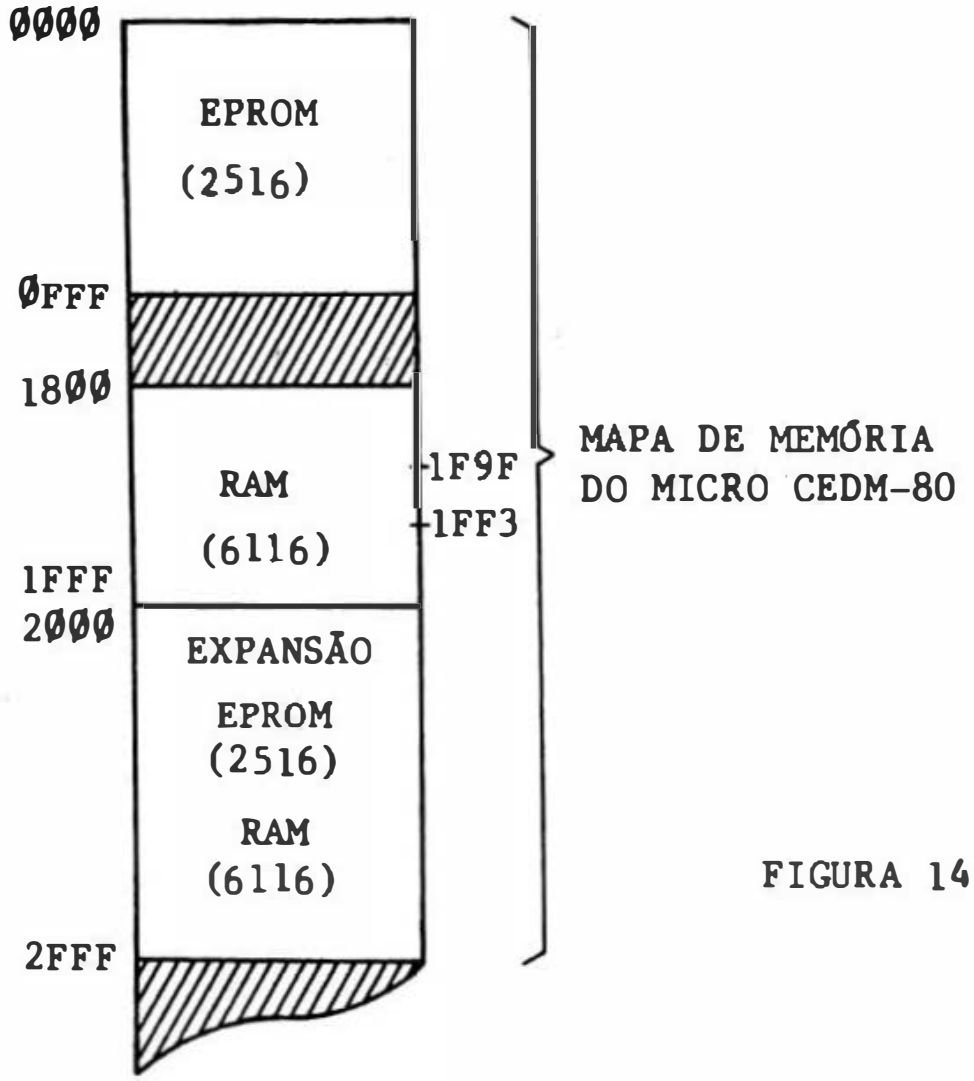


FIGURA 14

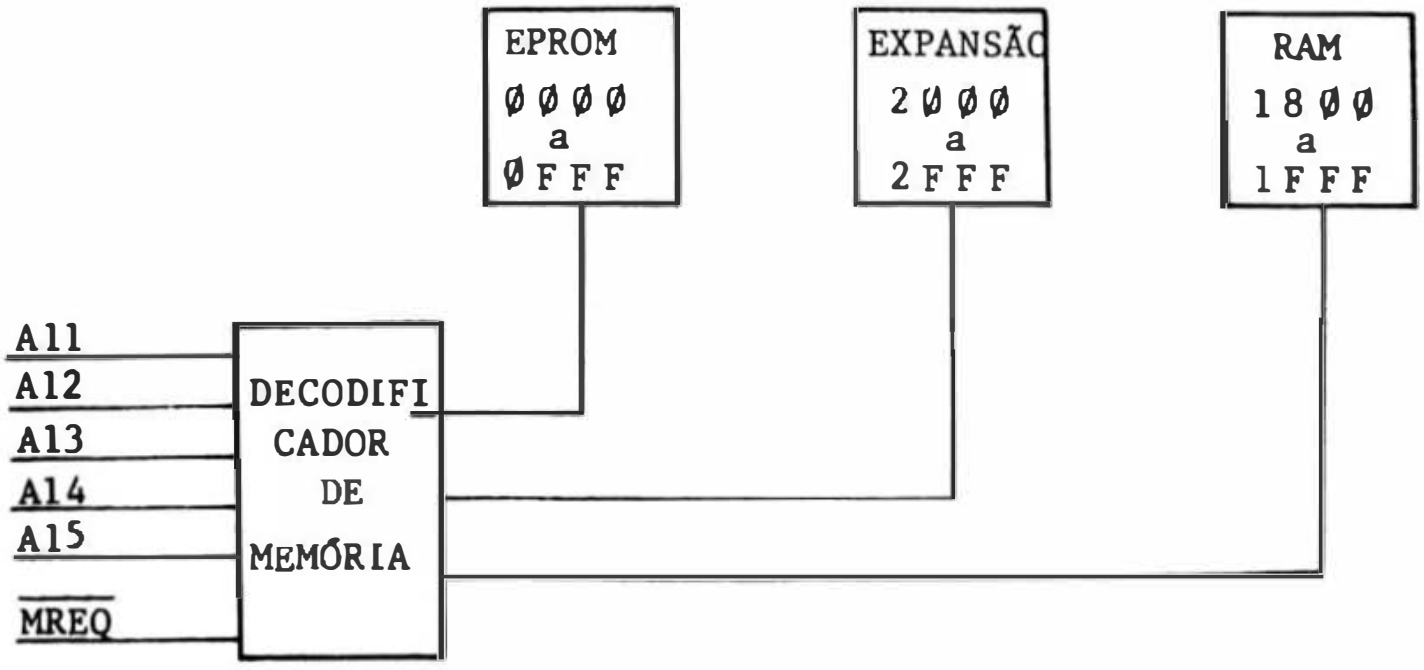
O sistema operacional, gravado na EPROM, também conhecido como programa residente, quando acessado, permite que a CPU realize a busca das instruções necessárias para a operação do sistema. Por exemplo, ao acionarmos uma determinada tecla, um sistema de leitura do teclado identificará a tecla acionada. Em seguida, a CPU irá emitir um pulso $\overline{MREQ}$ (supondo que seja este o caso), o qual, mediante o endereço que se encontra no barramento de endereços, atuará sobre o sistema decodificador de memória, o qual identificará a memória que está sendo endereçada. Caso seja a memória EPROM, a CPU irá então realizar a leitura da instrução contida no programa residente e, em seguida, irá executá-la.

Conforme podemos observar, na figura 13, temos também uma área reservada para as memórias do tipo RAM. Esta área de memória é responsável pelo armazenamento temporário de dados, os quais serão utilizados posteriormente. Em uma operação normal, os dados são transferidos continuamente entre a memória RAM e a CPU. O endereço de memória RAM que vai de "1F9F - 1FF3" (indicação hexadecimal) corresponde à área de RAM, reservada para "ponteiro de pilha" (stack pointer). Nesta área, a CPU irá arquivar as suas variáveis, ou os endereços, para chamar subrotinas, se ainda puder arquivar também, os dados provenientes dos registradores. Em síntese, esta região da memória RAM é o que vulgarmente poderíamos chamar de rascunho da CPU.

A terceira região do mapa de memória corresponde à área destinada para "expansão". Esta área foi reservada com vistas a uma possível expansão do banco de memórias RAM ou ainda uma provável expansão de memória EPROM. O bloco da CPU foi projetado de modo a poder receber nesta região (destinado à expansão) uma memória RAM 6116 ou uma EPROM 2516. Poderemos optar pelo uso de qualquer uma dessas memórias, uma vez que são equivalentes pino a pino. Essa expansão é opcional.

A figura 15 nos mostra um diagrama de blocos simplificado da memória do microcomputador CEDM-80.

FIGURA 15



A seguir, apresentamos as condições para a habilitação das memórias (tabela 1).

CONDIÇÕES PARA HABILITAÇÃO DAS MEMÓRIAS			
PULSO	EPROM	EXPANSÃO	RAM
$\overline{MREQ}$	Ø	Ø	Ø
A_{15}	Ø	Ø	Ø
A_{14}	Ø	Ø	Ø
A_{13}	Ø	1	Ø
A_{12}	Ø	Ø	1
A_{11}	X	X	1

TABELA 01

Obs.: X = condição irrelevante

Em seguida (figura 15), apresentamos as interconexões existentes entre o decodificador 74LS139, o bloco de memória e a CPU.

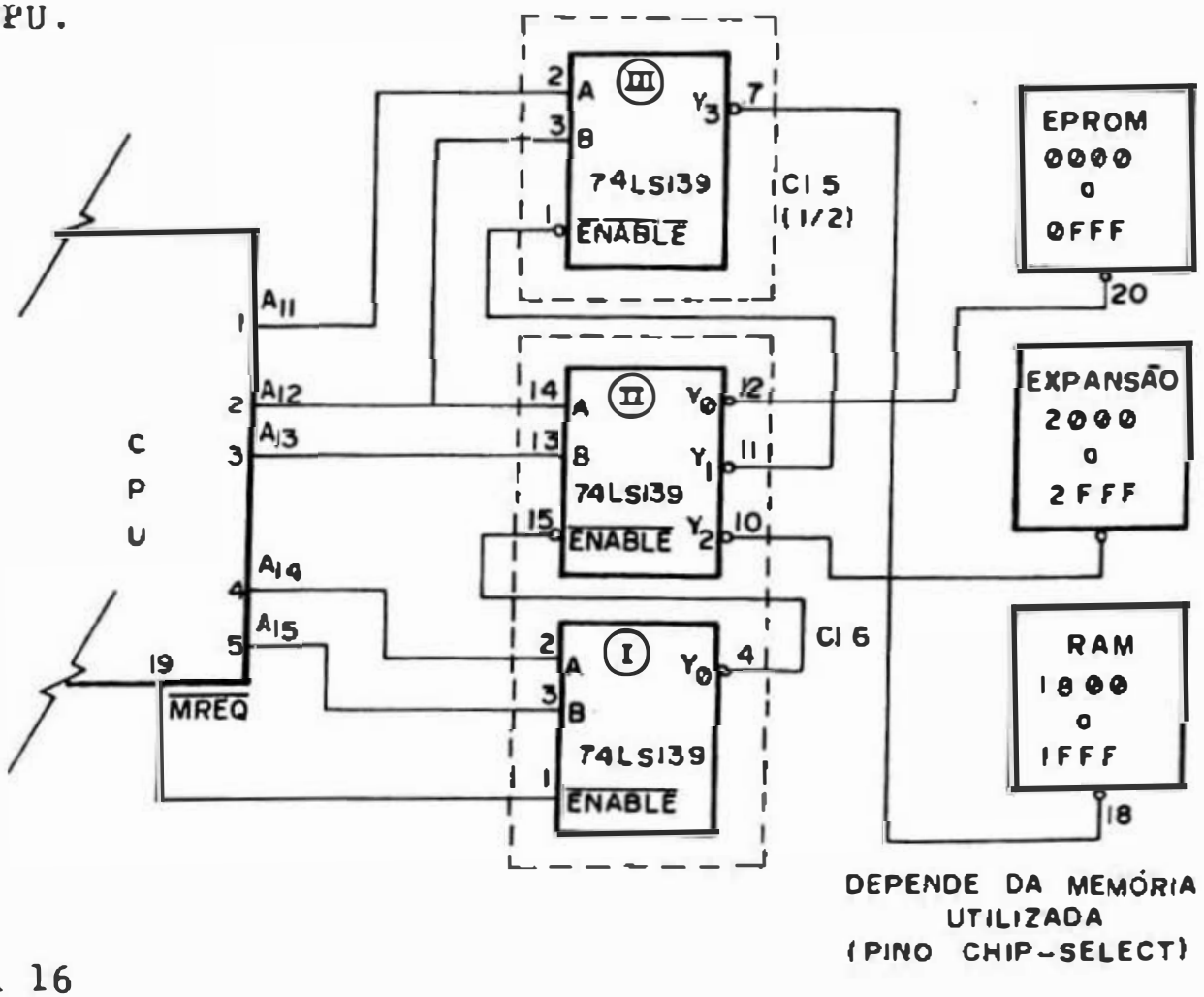


FIGURA 16

Cada circuito integrado 74LS139 possui dois decodificadores. No caso específico do micro CEDM-80, o bloco decodificador utiliza três (3) decodificadores, interconectados, conforme nos mostra a figura 15.

Dando continuidade ao nosso assunto, apresentamos a seguir, três tabelas, as quais tratam das condições necessárias para que um decodificador (74LS139) possa identificar, ou endereçar, uma determinada memória. As tabelas a seguir referem-se às condições necessárias para que os decodificadores de memória possam operar.

A_{15}	A_{14}	$\overline{MREQ}$	$Y_{\emptyset}$
X	X	1	1
Ø	Ø	Ø	Ø
Ø	1	Ø	1
1	Ø	Ø	1
1	1	Ø	1

TABELA 02 - Referente ao decodificador I

A_{13}	A_{12}	$\overline{EN}$	$Y_{\emptyset}$	Y_1	Y_2
X	X	1	1	1	1
Ø	Ø	Ø	Ø	1	1
Ø	1	Ø	1	Ø	1
1	Ø	Ø	1	1	Ø
1	1	Ø	X	X	X

TABELA 03 - Referente ao decodificador II

A_{12}	A_{11}	$\overline{EN}$	Y_3
X	X	1	1
1	1	Ø	Ø

TABELA 04 - Referente ao decodificador III

NOTAS: X = condição irrelevante;
$\overline{EN}$ = entrada "enable";
1 = nível lógico alto;
Ø = nível lógico baixo.

De acordo com as condições para habilitação das memórias (tabela 1) e, mediante um pulso $\overline{MREQ}$ emitido pela CPU, pode-se então realizar leitura ou escrita na memória.

Ainda, com relação à habilitação de memória, temos os jumpers de 1 a 7 que irão selecionar o chip utilizado na expansão. A tabela 5 nos mostra as ligações dos jumpers para cada tipo de memória que eventualmente poderão ser conectados na área dedicada à expansão.

		2716	6116	2732
CORTAR	Pinos	3 - 4	3 - 4	1 - 2
	Pinos	5 - 6	—	3 - 4
	Pinos	—	—	5 - 6
INTERLIGAR	Pinos	6 - 7	4 - 5	2 - 3
	Pinos	4 - 5	—	4 - 5
	Pinos	—	—	6 - 7

TABELA 05 - Referente à habilitação do chip de expansão

Na lição seguinte, estudaremos as partes referentes ao bloco de periféricos.

No encarte, segue o esquema geral do Kit CEDM-80, o qual será utilizado tanto para o estudo desta lição como nas duas lições seguintes, referentes ao Kit.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-22

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) O microprocessador utilizado no micro CEDM-80 é:

a) Z-80;

b) 4040;

c) 8085;

d) N.R.A.

2) A freqüência gerada pelo circuito mostrado na figura 3 é:

a) 4 MHz;

b) 3,58 Hz;

c) 3,58 MHz;

d) 1,79 MHz.

3) A freqüência de 1,79 MHz, utilizada pelo microprocessador Z-80, é obtida através de um circuito:

a) somador;

b) divisor por 2;

c) decodificador;

d) N.R.A.

4) A função "reset" (rearme) é utilizada para:

a) parar a execução de um determinado programa;

b) zerar o contador de programa;

c) endereçar a memória RAM;

d) as respostas a e b estão corretas.

5) O sistema de "reset" automático é utilizado para:

a) não zerar o contador de programa;

b) evitar espúrios de corrente elétrica gerados ao se ligar o micro;

c) evitar espúrios de corrente elétrica gerados, devido a possíveis choques mecânicos (quedas);

d) N.R.A.

6) O sistema "passo a passo" é:

a) comandado através de software interno;

b) acionado por intermédio do PPI-8255;

c) controlado pela malha RC formada por R7 e C9;

d) as respostas a e b então corretas.

7) Com relação à figura 6, podemos dizer que:

a) o resistor R7 garante o "reset" do sistema "passo a passo";

b) o capacitor C9 é utilizado para evitar espúrios de corrente, oriundos do acionamento da tecla MON1;

c) as respostas a e b estão corretas;

d) N.R.A.

8) De acordo com a figura 14, os endereços que vão desde 0000 até 0FFF determinam a área de memória destinada à memória:

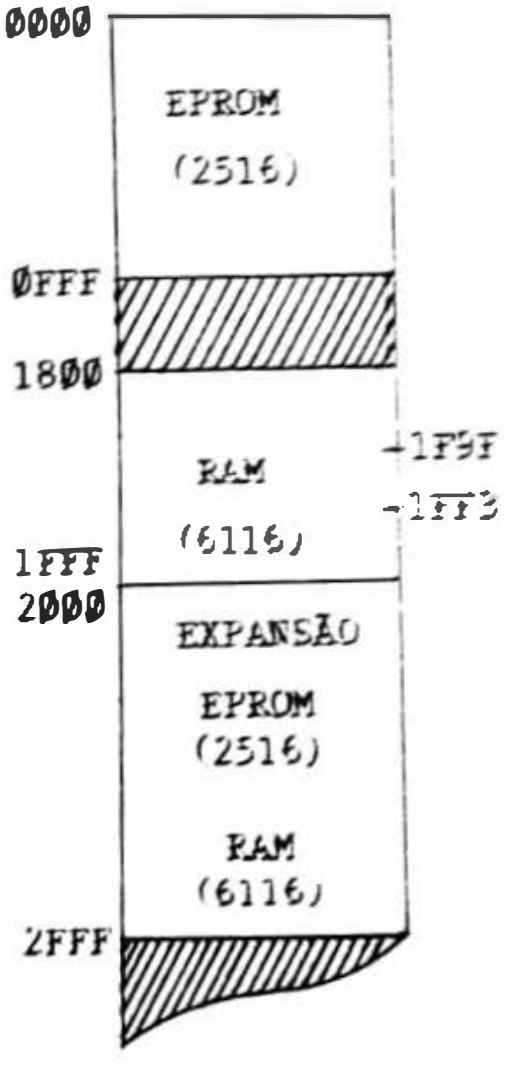
a) EPROM;

b) RAM;

c) EXPANSÃO;

d) N.R.A.

9) A figura a seguir se refere:



a) ao diagrama de blocos da CPU;

b) ao diagrama de blocos da memória EPROM;

c) ao mapa de memória do micro CEDM-80;

d) N.R.A.

10) Para que haja leitura ou escrita na memória, é necessário que a CPU emita um pulso:

a) $\overline{MNI}$;

b) $\overline{MI}$;

c) $\overline{MREQ}$;

d) N.R.A.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-23

MICROCOMPUTADOR C E D M - 8 0

INTRODUÇÃO

Dando continuidade ao assunto iniciado na lição anterior, passaremos agora ao estudo do bloco de periféricos. O nosso estudo será direcionado da mesma forma que a lição anterior, ou seja, inicialmente faremos uma apresentação descritiva das partes que compõem este bloco e, em seguida, apresentaremos essas partes a nível de componentes.

BLOCO DE PERIFÉRICOS

O bloco de periféricos é formado basicamente pelas seguintes partes:

a) PPI.
b) Teclado.
c) Display.
d) Gravador Cassete.
e) Conector para opcionais.

A figura 1 nos mostra o diagrama esquemático do bloco de periféricos. Nele não está incluído os diversos circuitos decodificadores necessários ao funcionamento de cada uma das

partes do bloco de periféricos. Esses circuitos decodificadores serão apresentados juntamente com cada uma das partes que serão estudadas logo em seguida.

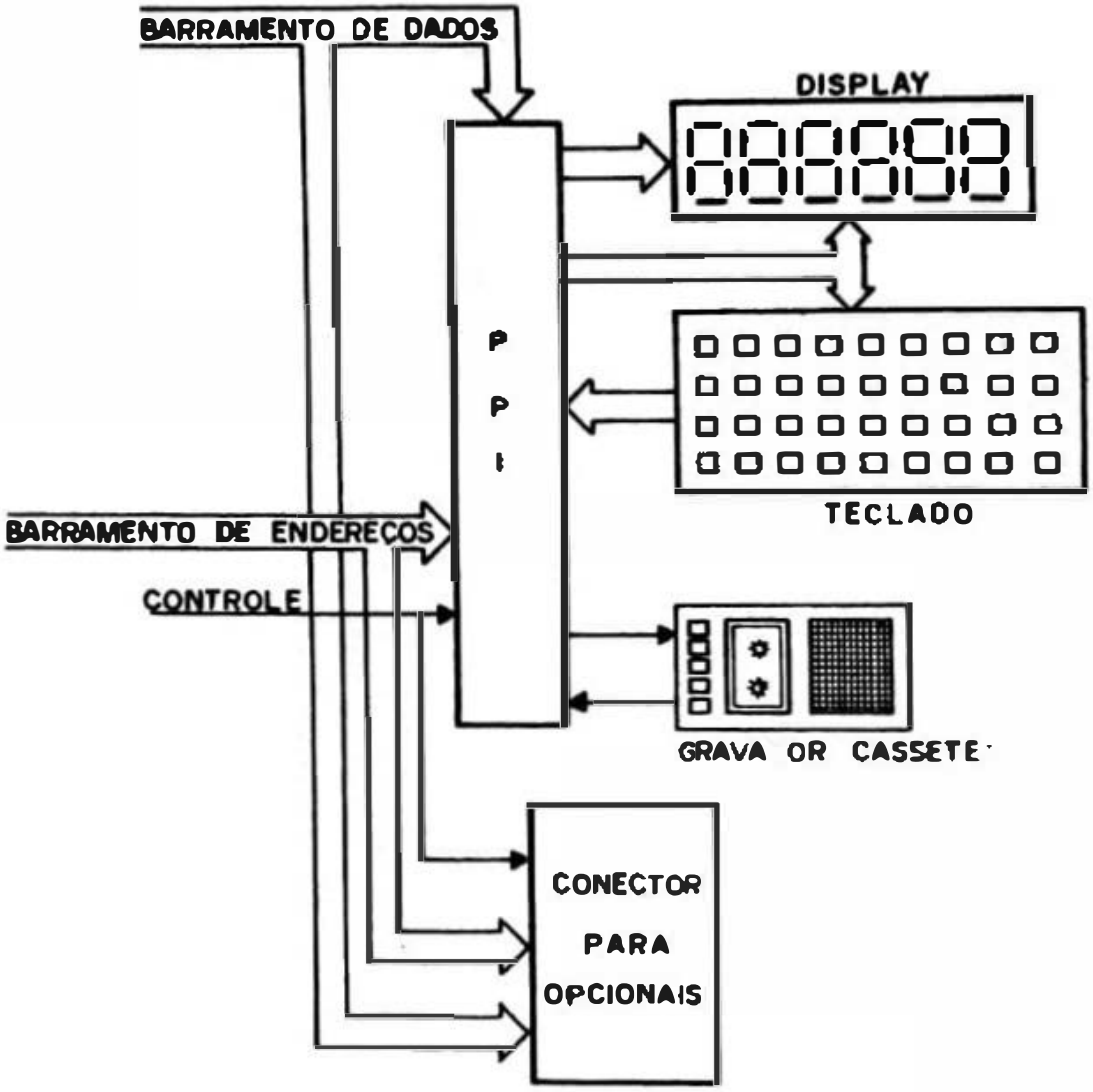


FIGURA 01

Passemos, então, ao estudo de cada uma das partes, iniciando pela "interface de periféricos programável" (PPI).

a) Interface de Periféricos Programável (PPI):

A interface de periféricos programável ou, simplesmente, PPI é um chip LSI (grande escala de integração), utilizado em microcomputadores, servindo de interface entre a máquina (microcomputador) e os dispositivos externos de entrada/saída (E/S). Atualmente, além da PPI, existem outros chips que também

fazem o "interfaceamento" com os dispositivos externos de entrada/saída, como, por exemplo, a interface programável de comunicações ou simplesmente USART. A principal diferença entre os chips PPI e USART reside no fato de que o PPI foi projetado para realizar transferência paralela de dados, enquanto que a USART realiza a transferência de dados na forma serial (ou série).

No caso específico do microcomputador CEDM-80, devido a uma série de fatores, optamos por utilizar um chip PPI para interfacear com os dispositivos externos de E/S. Entre os fatores que nos levaram a esta escolha podemos citar dois como sendo de grande importância: o primeiro, talvez o mais importante, diz respeito à alta flexibilidade que o PPI proporciona ao interfaceamento do microcomputador com os dispositivos externos de E/S. Esta flexibilidade é obtida, tornando o software do PPI configurável, de forma que as suas linhas de E/S estejam mais sob o controle do software do que do hardware, facilitando, assim, a alocação de linhas de E/S e quaisquer outras alterações. O segundo fator está relacionado ao fato de que o PPI concentra as operações paralelas de entrada e saída em um único circuito integrado (a menos que as suas linhas de E/S não sejam suficientes) o que nos proporciona uma redução do número de circuitos integrados, além de diminuir o grau de complexibilidade do sistema de interfaceamento.

O circuito integrado PPI escolhido foi o 8255, cujo diagrama de blocos e pinagem são mostrados na figura 2.

A interface de periféricos programável 8255 possui 24 linhas paralelas de entrada e saída as quais se encontram divididas em três portas A, B e C. A porta A é uma porta de entrada enquanto que as portas B e C são portas de saída (modo "0"). A figura 3 nos mostra um esquema referente à pinagem dessas portas.

As portas A e B possuem 8 bits. A porta C também é constituída por 8 bits, porém, separados em dois grupos de 4 bits, os quais poderão ser usados tanto para entrada como para saída; particularmente, no caso do micro CEDM-80, os dois grupos de 4 bits, que formam a porta C, serão utilizados como portas de saída.

Com relação ao modo de operação, o circuito integrado PPI 8255 pode operar de três (3) modos diferentes:

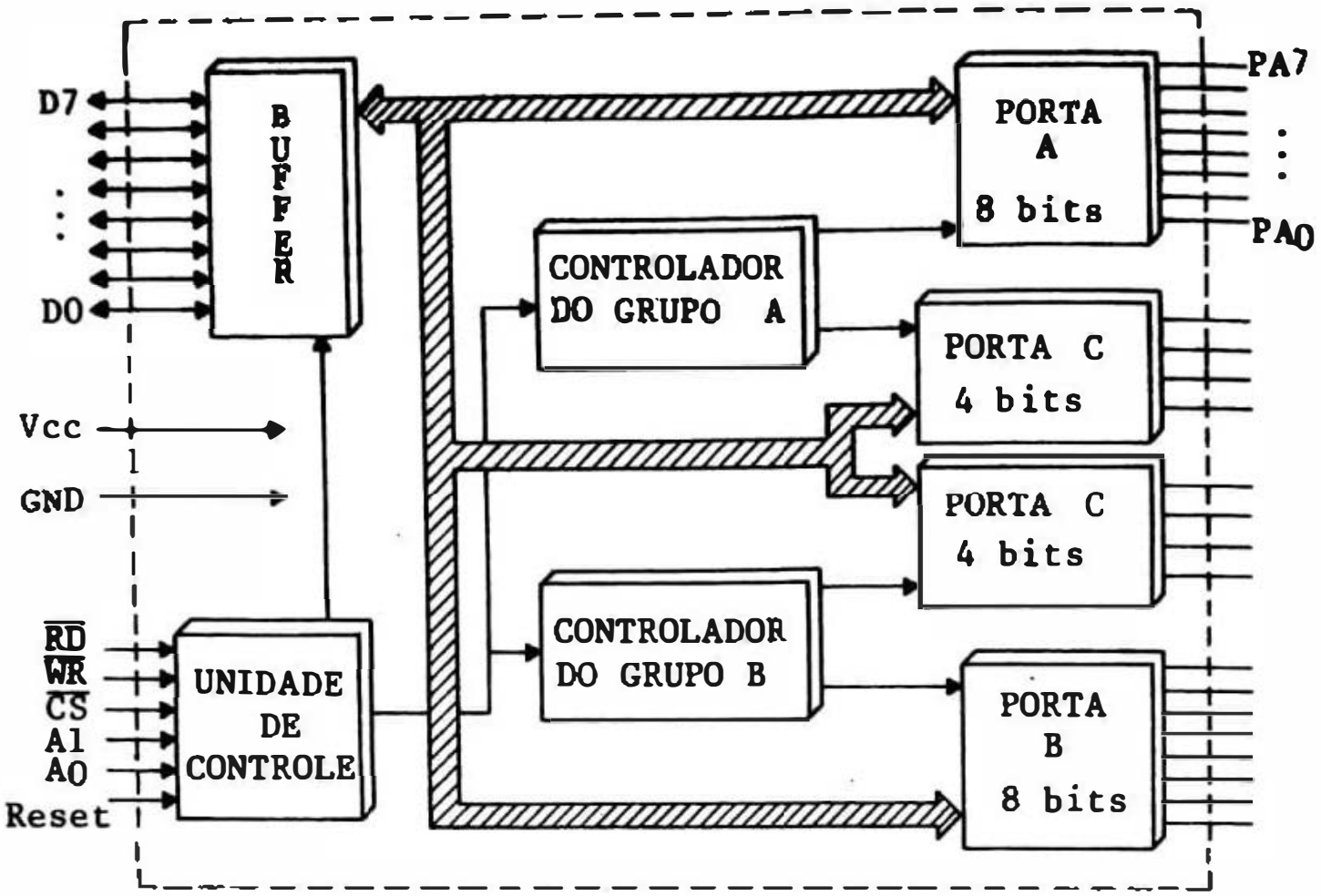
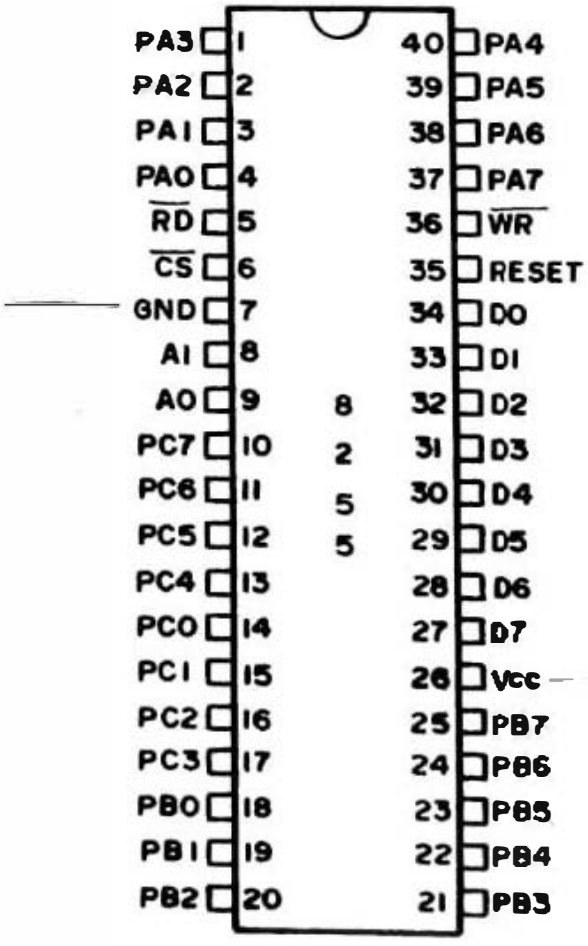


DIAGRAMA DE BLOCOS



PINAGEM

FIGURA 02

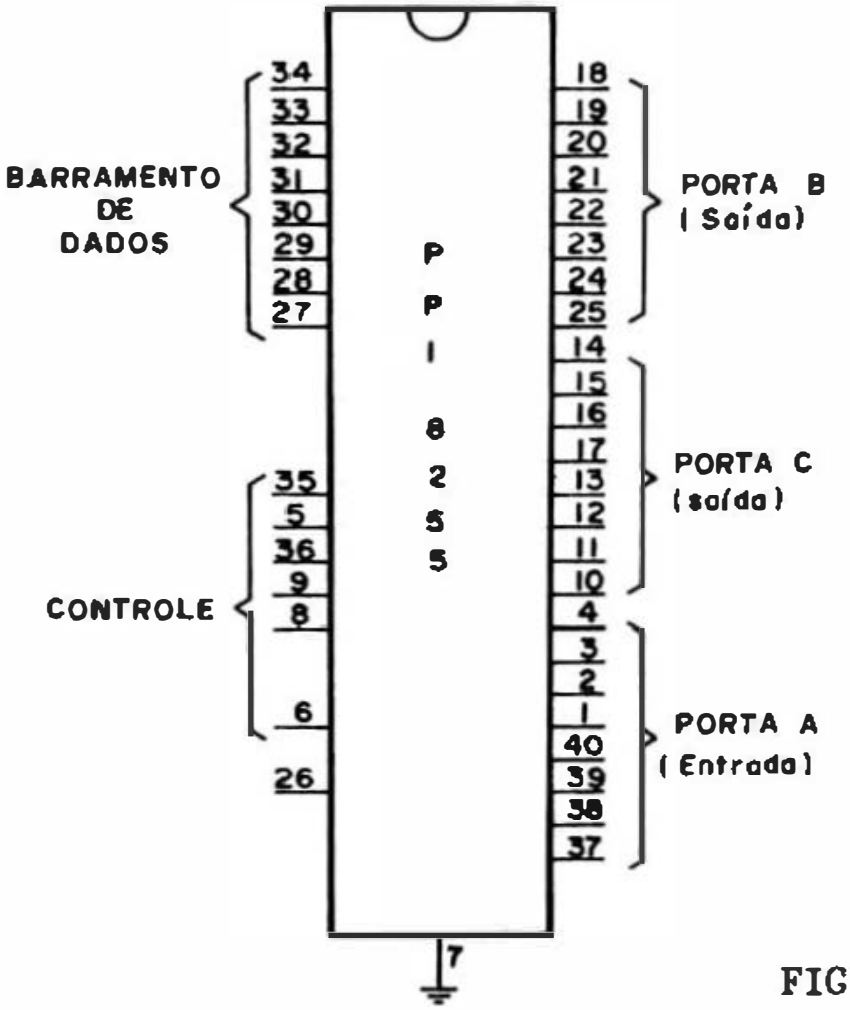


FIGURA 03

a) Modo 0:

Neste modo de operação, as portas A, B e C se encontram programadas para servirem, basicamente, de entradas ou saídas de dados.

b) Modo 1:

Neste modo de operação, as portas A e B se encontram programadas para servirem de entradas ou saídas de dados, porém deverão estar sincronizadas por meio de sinais especiais de controle, ligados à porta C.

c) Modo 2:

Neste modo de operação, a porta B não é utilizada, pois a porta A irá constituir-se em um barramento de dados bidirecional de entrada e saída. Os bits da porta C serão utilizados para sinais de controle, de forma semelhante ao que ocorre no modo 1.

O circuito integrado PPI 8255 poderá trabalhar em um des

ses três (3) modos em separado ou ainda poderá trabalhar por meio de combinações destes modos, ou seja, uma porta poderá operar em um dado modo, enquanto que a(s) outra(s) poderão operar em outro modo, simultaneamente.

No caso específico do micro CEDM-80, o PPI 8255 irá operar no modo zero, cuja palavra de controle é 90_H que equivale a 1001000_2. A figura 4 nos mostra um esquema simplificado de como as portas do PPI 8255 irão trabalhar.

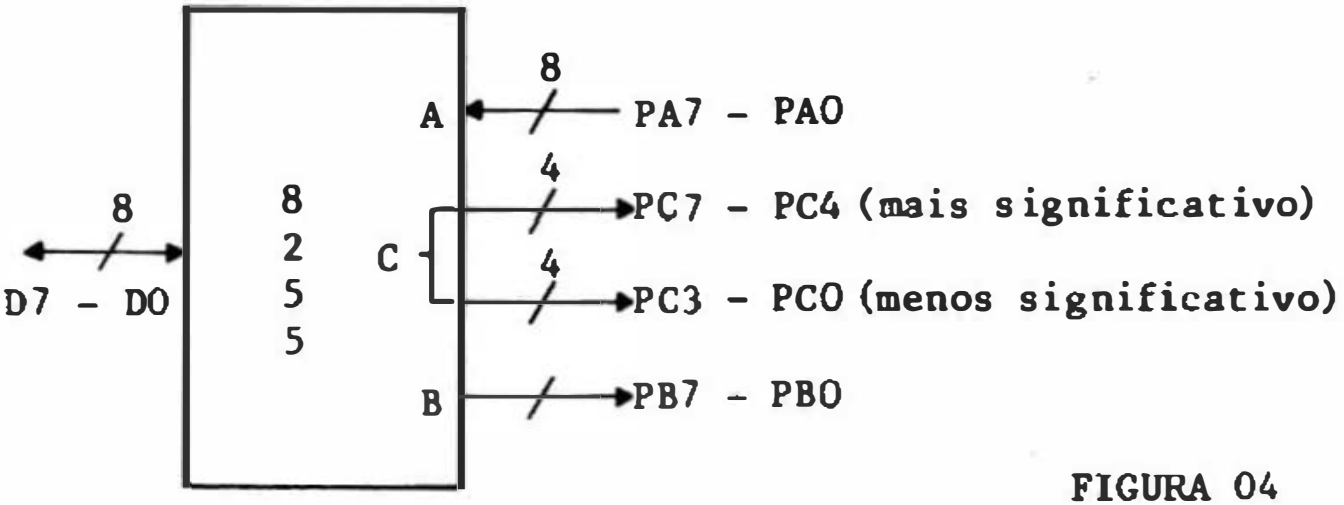


FIGURA 04

A tabela verdade para operação do PPI 8255 é mostrada na tabela 1, a seguir.

TABELA 1

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Operação de entrada (leitura)
0	0	0	1	0	Porta A - barramento de dados
0	1	0	1	0	Porta B - barramento de dados
1	0	0	1	0	Porta C - barramento de dados
					Operação de saída (escrita)
0	0	1	0	0	Barramento de dados - Porta A
0	1	1	0	0	Barramento de dados - Porta B
1	0	1	0	0	Barramento de dados - Porta C
1	1	1	0	0	Barramento de dados - Controle
					Função de desabilitação
X	X	X	X	1	Barramento de dados - 3-state — O PPI não é selecionado
1	1	0	1	0	Condição ilegal

De modo semelhante ao que ocorre com as memórias, o PPI 8255 também possui um mapa de endereços, o qual é mostrado na figura 5.

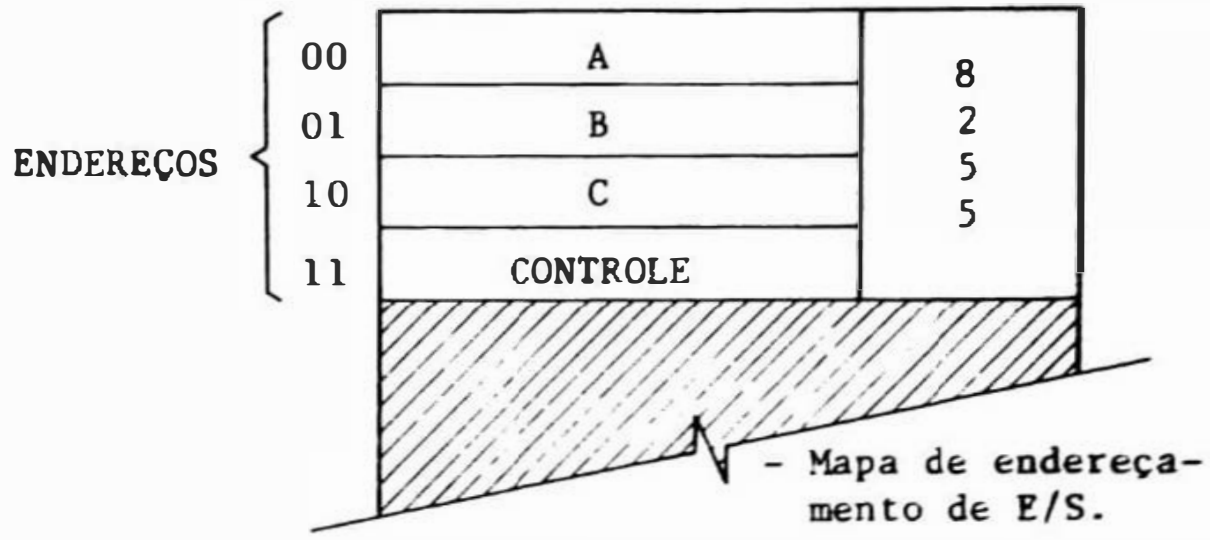


FIGURA 05

A figura 6 nos mostra de forma resumida, como determinar a palavra de controle do PPI 8255.

No caso do micro CEDM-80, a palavra de controle do PPI 8255 é 90H ou 10010000 em binário, a qual é colocada no barramento de dados. Vejamos, então, como foi determinada a palavra de controle:

- O bit D_0 referente à porta C será igual a "0" (zero), pois a porta C irá operar como saída. O bit D_1 também será igual a "0", uma vez que a porta B irá operar como saída. Como o modo de seleção escolhido é o modo "0", isto leva o bit D_2 ser igual a "0". O bit mais significativo da porta C, representado pelo bit D_3, também será igual a "0", uma vez que a porta C é uma porta de saída. O bit D_4 será igual a "1", pois a porta A é uma porta de entrada. Por outro lado, os bits D_5 e D_6, que representam o modo de seleção (grupo A), nos levam a considerá-los como sendo "00", uma vez que o modo de seleção é o modo "0". Finalmente, temos o bit D_7 que se refere ao Flag de interrupção que, neste caso, deverá estar ativado. Assim sendo, podemos compor a palavra de controle como nos mostra a figura 7.

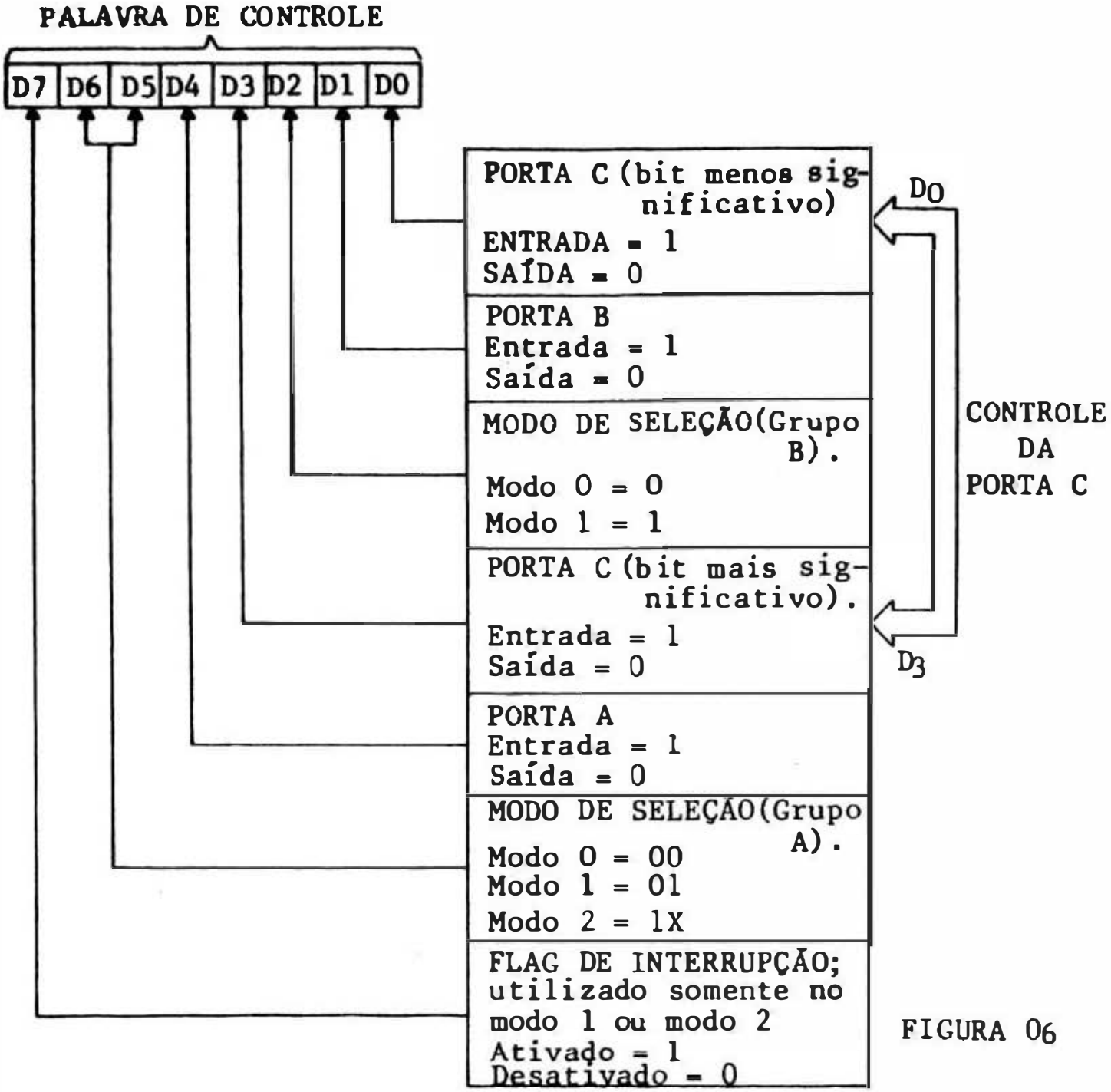


FIGURA 06

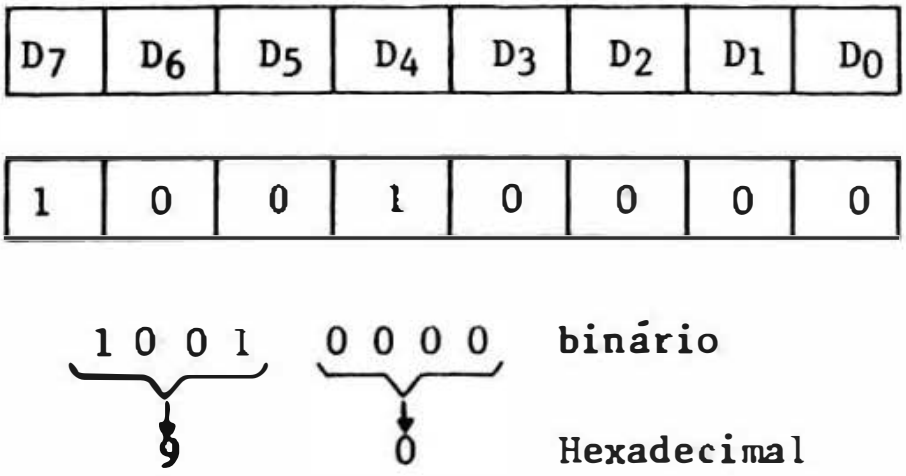


FIGURA 07

Por outro lado, os endereços de dispositivos para as portas A, B e C e para o registro de controle do PPI 8255 são determinados, em geral, pelas linhas de endereço $\overline{A0}$, $\overline{A1}$ e $\overline{CS}$. As linhas A_0 e A_1 do PPI 8255 estão conectadas às linhas de endereço A_0 e A_1 do microprocessador. A figura 8 nos mostra o formato do código de endereço para seleção de dispositivo.

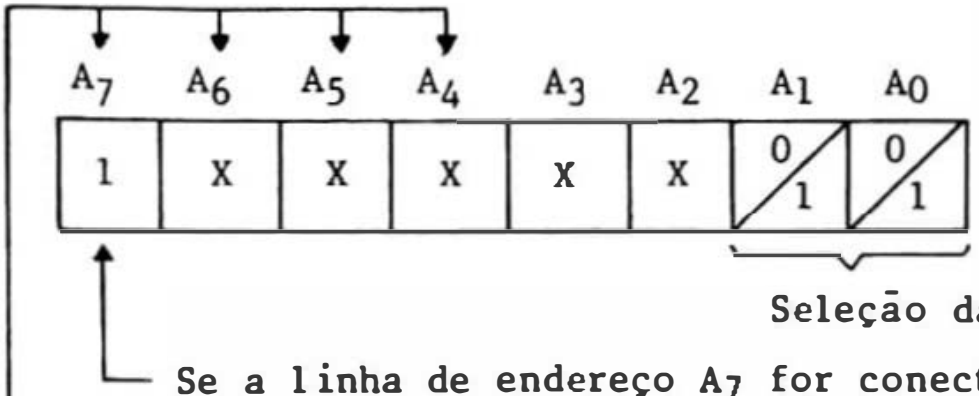


FIGURA 08

Nota: X = não utilizado - fixar em zero.

A tabela 2, a seguir, nos mostra de forma resumida a função dos pinos A_0 e A_1 do PPI 8255:

TABELA 2

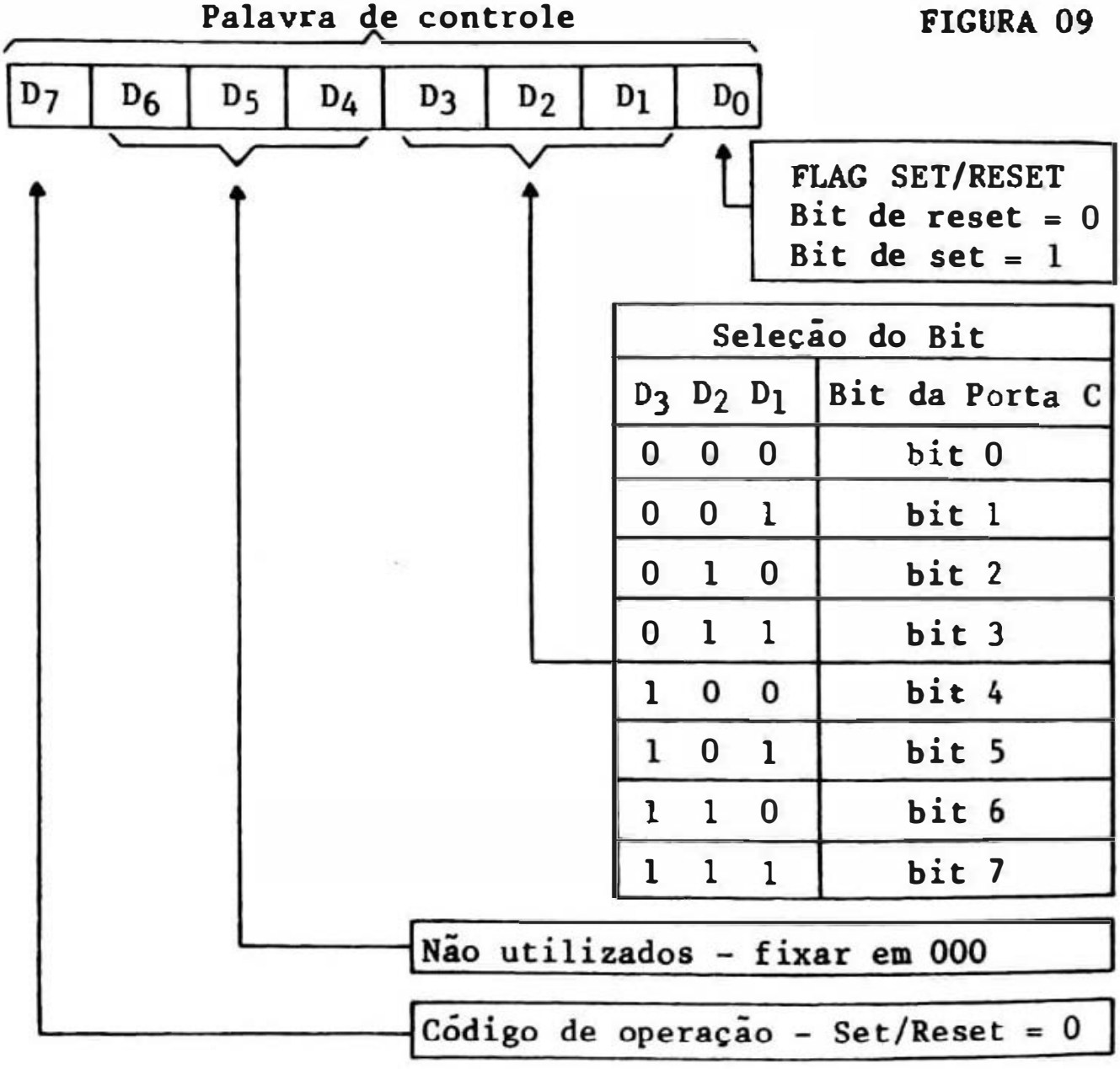
A_1	A_0	Operação de E/S
0	0	Barramento de dados ↔ porta A
0	1	Barramento de dados ↔ porta B
1	0	Barramento de dados ↔ porta C
1	1	Barramento de dados ↔Registro de controle

Por exemplo, para imprimir no registro de controle, o bit A_7 deve ser "1" e os bits A_0 e A_1 também devem ser "1". Assim, o código de endereço será 1XXXXX11. Ajustando-se X à lógica "0", o código será: 10000011 em binário ou, simplesmente, 83 em hexadecimal (também indicado como 83 H).

Um outro aspecto de relevante importância é a operação

de Set/Reset do PPI 8255. Esta função (Set/Reset) é uma característica peculiar da porta C. Para que esta propriedade da porta C possa ser utilizada, uma palavra de controle de set/reset deve ser enviada para o registro de controle do PPI. O formato desta palavra de controle é mostrado na figura 9. Note que o bit D_7 desta palavra de controle é que irá determinar se a palavra de controle (ou byte de controle) será interpretada pelo PPI 8255 como sendo uma palavra de controle de modo ($D_7 = 1$) ou uma palavra de controle de bit set/reset ($D_7 = 0$).

FIGURA 09



A característica de set/reset é muito utilizada quando o PPI estiver operando nos modos 1 e 2, pois, nestes casos, são utilizados E/S uni e bidirecional, respectivamente.

Uma outra utilização para a característica de set/reset da porta C é que os seus bits podem ser usados para as funções "gating", "strobing" e "reset", as quais, normalmente, estão associadas à seleção de dispositivos ou pulsos de seleção de endereços. Tudo isto é feito através do software.

Por outro lado, o PPI 8255 possui uma entrada de reset (pino 35) que permite o "zeramento" dos seus registradores internos, incluindo a palavra de controle. Após ocorrer o reset do PPI 8255, a primeira palavra, que surgir no barramento de dados, o PPI irá interpretá-la como sendo a palavra de controle.

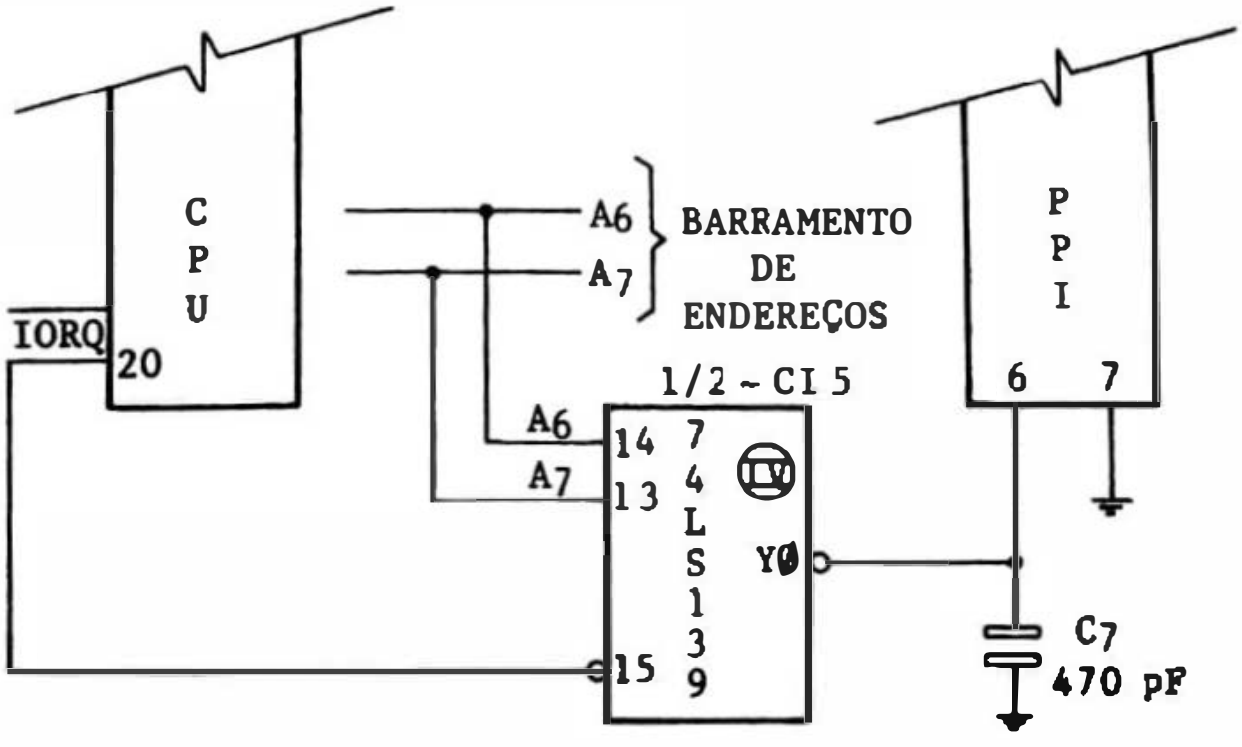
No micro CEDM-80, o PPI 8255 é acionado quando tivermos presente, na entrada do decodificador 74LS139 (C.I.5), um pulso de nível lógico baixo, enviado pela CPU, através de sua linha de controle $\overline{IORQ}$. A tabela 3, a seguir, nos mostra as condições necessárias para que o decodificador do PPI possa operar.

A_7	A_6	$\overline{IORQ}$	Y_0	
0	0	0	0	Seleciona o PPI
X	X	1	1	Não seleciona o PPI

TABELA 3 - Referente ao decodificador IV (2ª metade do CI5)

Em seguida, a figura 10 nos mostra as interconexões entre o decodificador 74LS139 e o PPI 8255.

FIGURA 10

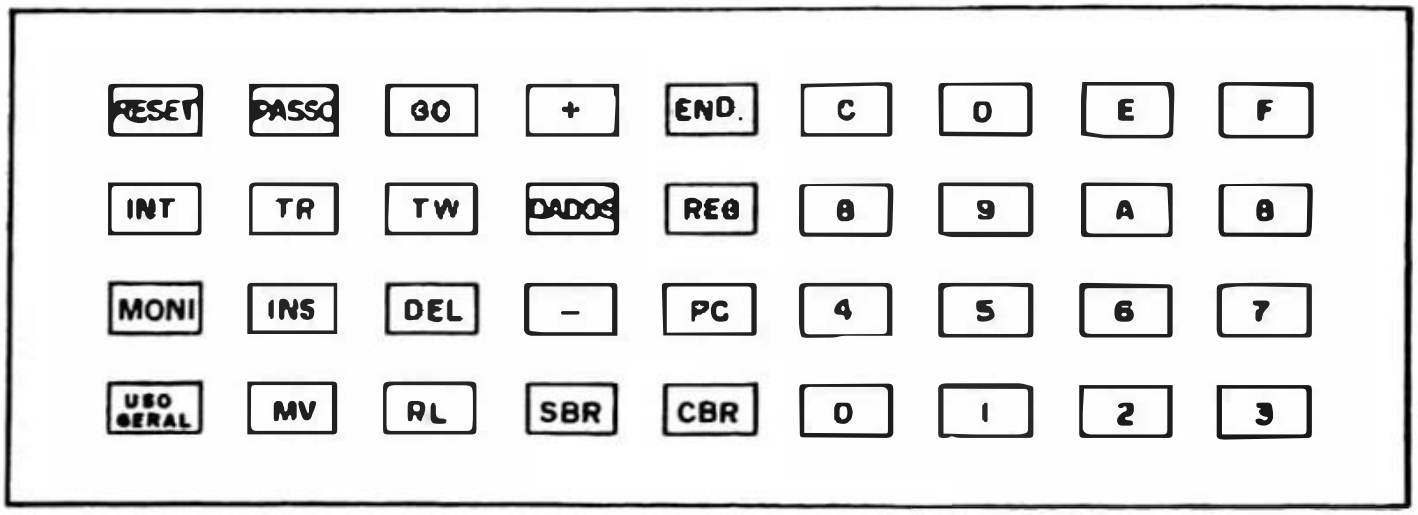


O capacitor C_7 funciona como um ceifador de sinais.

Finalizando este tópico referente ao PPI, reafirmamos que este só poderá operar mediante a emissão de um pulso de nível lógico baixo, enviado pela CPU, através de sua linha de controle $\overline{IORQ}$ (INPUT/OUTPUT REQUEST).

b) Teclado:

O teclado do micro CEDM-80 é composto por 36 teclas, sendo 19 de funções, 16 para numeração hexadecimal e 1 tecla definida pelo usuário. A figura 11 nos mostra um lay-out do teclado do micro CEDM-80.



TECLADO

FIGURA 11

A figura 12, por sua vez, nos mostra o esquema elétrico do teclado. Note que, inicialmente, todas as linhas do teclado se encontram a um nível lógico alto ("1"); ao pressionarmos uma tecla qualquer, a linha correspondente será levada a um nível lógico baixo.

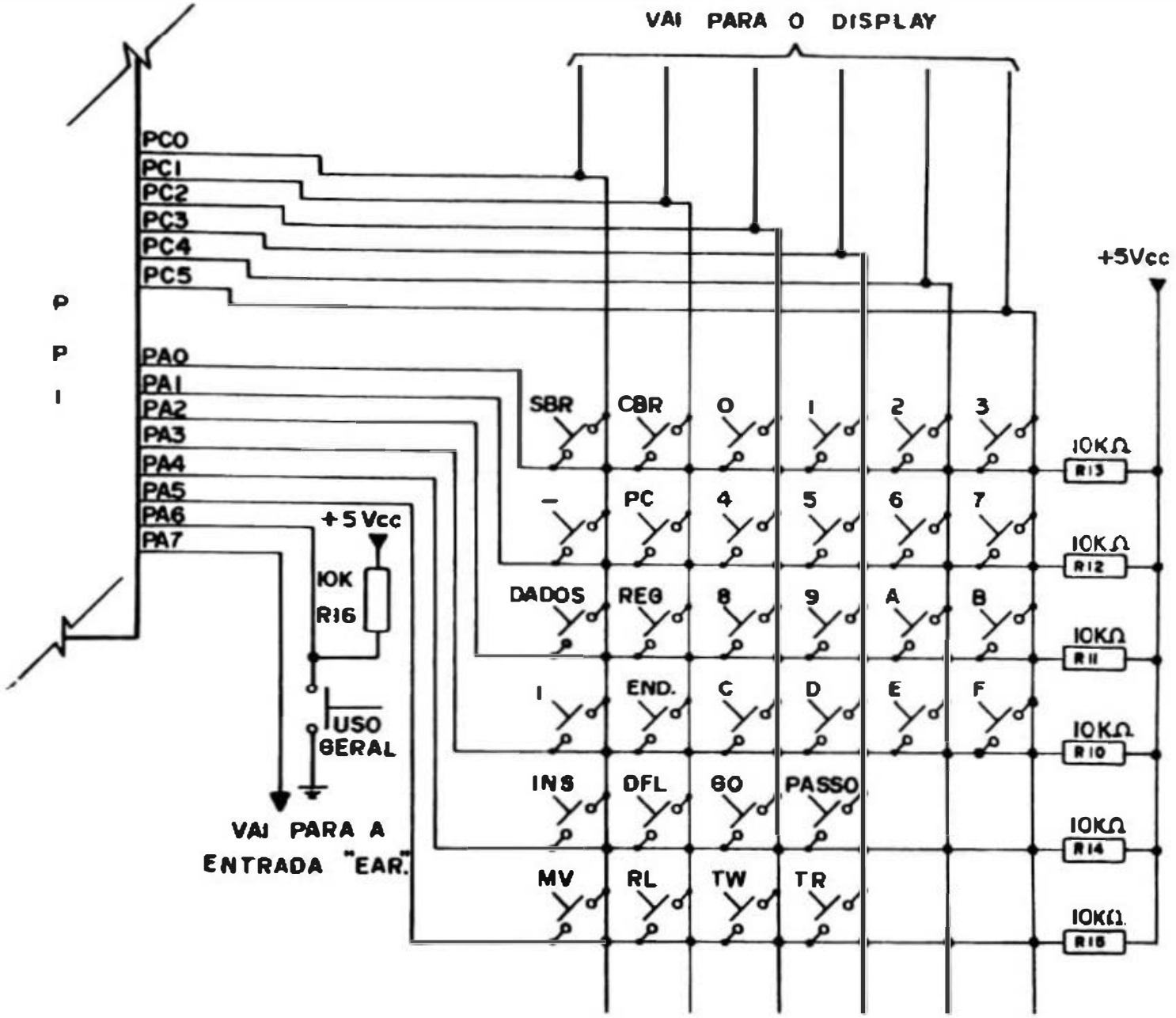


FIGURA 12

No processo de varredura do teclado, a CPU envia para o PPI um pulso de habilitação da porta C. Imediatamente, após enviar o pulso de habilitação da porta C, a CPU coloca no barramento de dados um determinado dado, o qual tem, por função, executar a varredura do teclado. Por exemplo, se o dado transferido pela CPU para a porta C do PPI for 01111111, este proporcionará a varredura das teclas que se encontram na primeira coluna do teclado, ou seja, as teclas SBR, -, Dados, +, INS e MV, conforme podemos observar na figura 12. Se alguma dessas teclas estiver pressionada, a linha correspondente

a esta tecla estará a um nível lógico baixo ("0"), gerando, assim, um "dado" que é colocado na porta A do PPI, o qual permitirá a identificação da tecla acionada, através de um processo de comparação, realizado pela CPU, entre o dado (01111111), colocado na porta C e o dado colocado na porta A. Vamos supor, por exemplo, que a tecla MV esteja sendo pressionada; neste caso, a linha PA5 será levada a um nível lógico baixo ("0") e a palavra gerada para a porta A do PPI será 11111011. Assim, os dados 01111111 (colocados na porta C) e 11111011 (que se encontram na porta A) permitirão que a CPU interprete qual é a tecla que está sendo pressionada. Essa interpretação por parte da CPU ocorre no instante em que a tecla deixar de ser pressionada, ou seja, quando a linha correspondente à tecla volta para o nível lógico alto ("1"). A interpretação deste fato, em termos de nível lógico, é mostrada na figura 13.

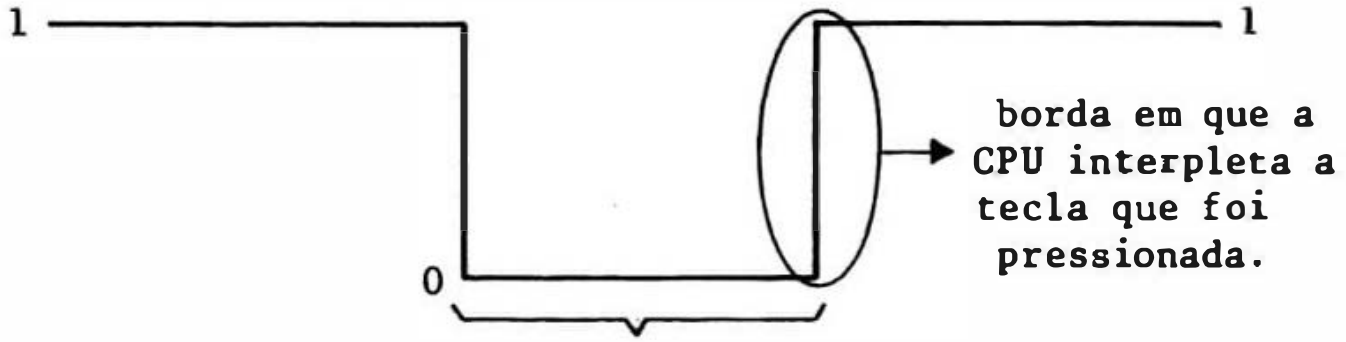


FIGURA 13

O princípio de varredura do teclado, onde o PPI opera no "modo 0", consiste no fato da CPU colocar um nível lógico baixo em uma das colunas, imediatamente, após "recolher" este nível lógico baixo, através de uma das linhas. Quando a CPU não recebe de volta, através das linhas, o pulso baixo que foi colocado em uma das colunas, ela interpretará que nenhuma tecla foi pressionada.

A tabela 4, apresentada, a seguir, nos mostra as condições impostas pela CPU para cada uma das colunas, durante o processo de varredura do teclado.

TABELA 4

Colunas / Varredura	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
1ª Varredura	0	1	1	1	1	1	1	1
2ª Varredura	1	0	1	1	1	1	1	1
3ª Varredura	1	1	0	1	1	1	1	1
4ª Varredura	1	1	1	0	1	1	1	1
5ª Varredura	1	1	1	1	0	1	1	1
6ª Varredura	1	1	1	1	1	0	1	1
7ª Varredura	1	1	1	1	1	1	0	1
8ª Varredura	1	1	1	1	1	1	1	0

Cabe observar que a 7ª varredura somente irá ocorrer quando a tecla "Passo" for pressionada. Por outro lado, a 8ª varredura ocorrerá quando a tecla "TW" for pressionada para a gravação de informações na fita de áudio (gravador cassete) ou quando qualquer uma das teclas for pressionada, acionando o circuito de sinalização (visual ou sonoro, ambos opcionais), que indica se a tecla foi ou não pressionada.

A tabela 5 nos mostra as divisões referentes à porta A do PPI.

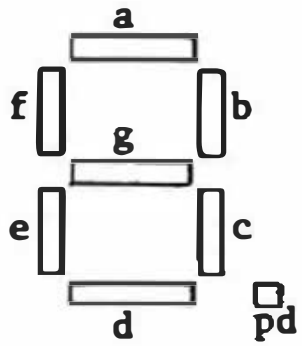
TABELA 5

Porta A	Divisões
PA0	Estas 6 entradas da porta A estão conectadas às 6 colunas do teclado.
PA1	
PA2	
PA3	
PA4	
PA5	
PA6	Acionada através do software, corresponde à tecla de "uso geral", a qual é ativada quando em nível lógico baixo ("0").
PA7	Esta entrada é utilizada para leitura de dados da fita cassete, através da conexão EAR. A CPU inicia o recebimento destes dados somente após pressionarmos a tecla TR.

O endereço da porta A (tabela 2) é 00. Os resistores R_{10} a R_{15} forçam as linhas do teclado a ficarem em um nível lógico alto ("1"). Um outro detalhe importante é que o micro CEDM 80 n o irá interpretar duas ou mais teclas pressionadas, simultaneamente.

c) Display:

As portas PB0 a PB7 (pinos 18 a 25 do 8255) excitam, por intermédio dos transistores TR_1 a TR_8 (todos do tipo BC 548) os segmentos e o ponto decimal do display). A figura 14 nos mostra os bits da porta B e os correspondentes segmentos do display



7	6	5	4	3	2	1	0
d	pd	c	b	a	f	g	e

nota: pd = ponto decimal

FIGURA 14

Os transistores TR_1 a TR_8 funcionam como driver de corrente para excitar os segmentos do display. Para uma correta polarização da base desses transistores, foram empregadas as resistências R_{20}, R_{22}, R_{24}, R_{26}, R_{28}, R_{30}, R_{32} e R_{34}, todas de 3,3 KΩ (ou 3K3Ω). As resistências R_{21}, R_{23}, R_{25}, R_{27}, R_{29}, R_{31}, R_{33} e R_{35}, todas de 10Ω, operam como resistências limitadoras da corrente de excitação dos segmentos do display. A figura 15 nos mostra o circuito eletrônico de excitação e multiplexação do display.

A multiplexação do display é feita através do CI_{10} (portas lógicas inversoras do tipo 74LS04).

d) Gravador Cassete (Sistema de leitura "EAR" e de gravação "MIC" opcionais):

A entrada EAR é utilizada para se fazer a leitura de dados da fita cassete, podendo esta ser conectada à saída de fone de qualquer gravador do tipo cassete. A figura 16 nos mostra o circuito eletrônico referente ao sistema de leitura EAR.

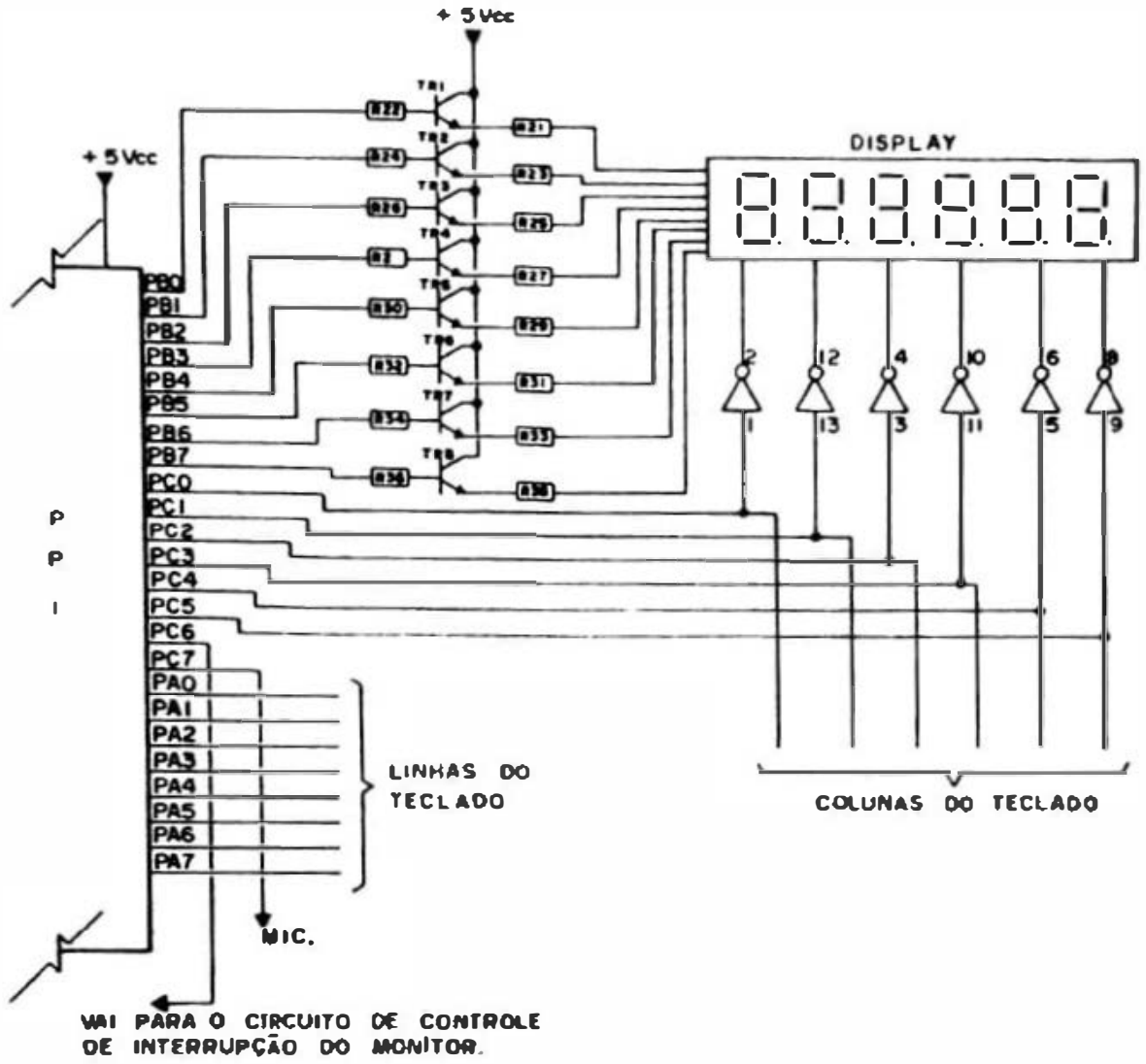


FIGURA 15

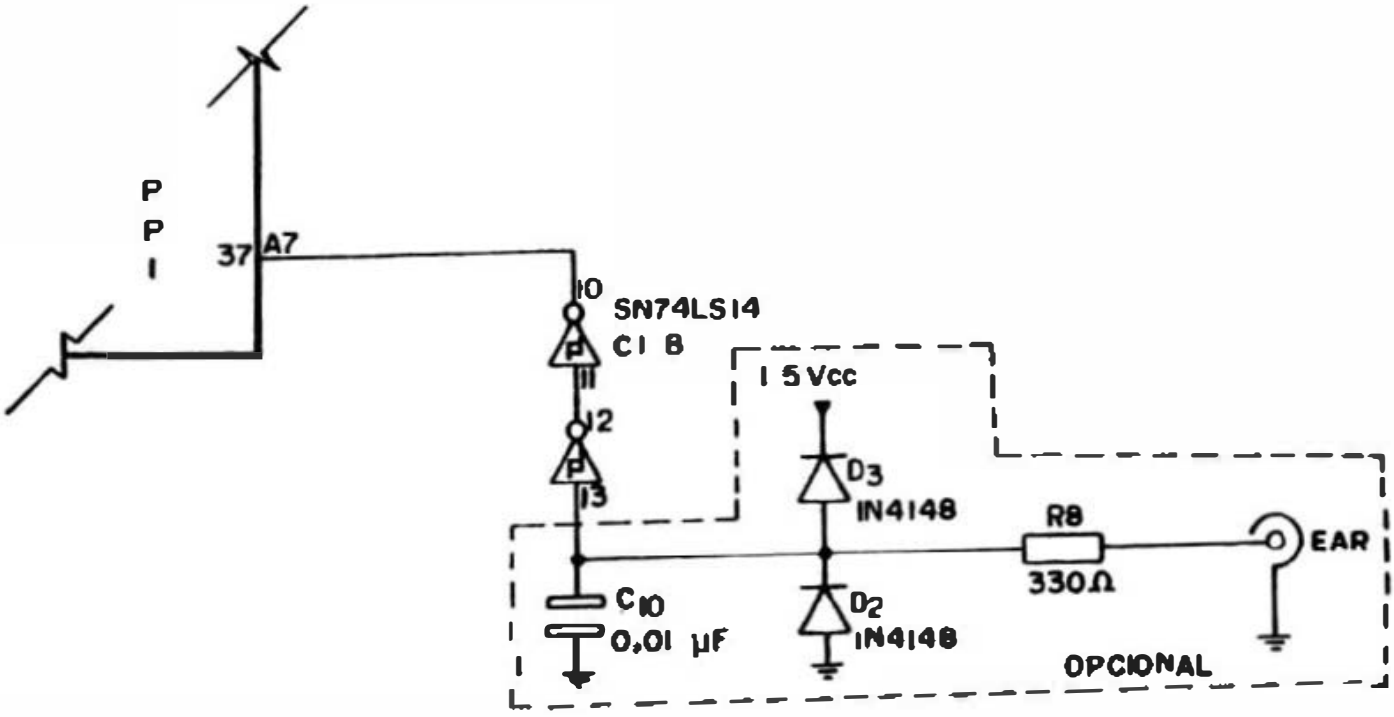


FIGURA 16

O resistor R_8 tem por finalidade isolar o nível de tensão do sinal de áudio enviado pelo gravador cassete. Os diodos D_2 e D_3 (ambos 1N4148) juntamente com o capacitor C_{10} (0,01 µF) formam um circuito ceifador de pico, o qual tem por função ceifar os picos do sinal de áudio. As formas de onda, mostradas na figura 17, exemplificam o ceifamento do sinal de áudio.

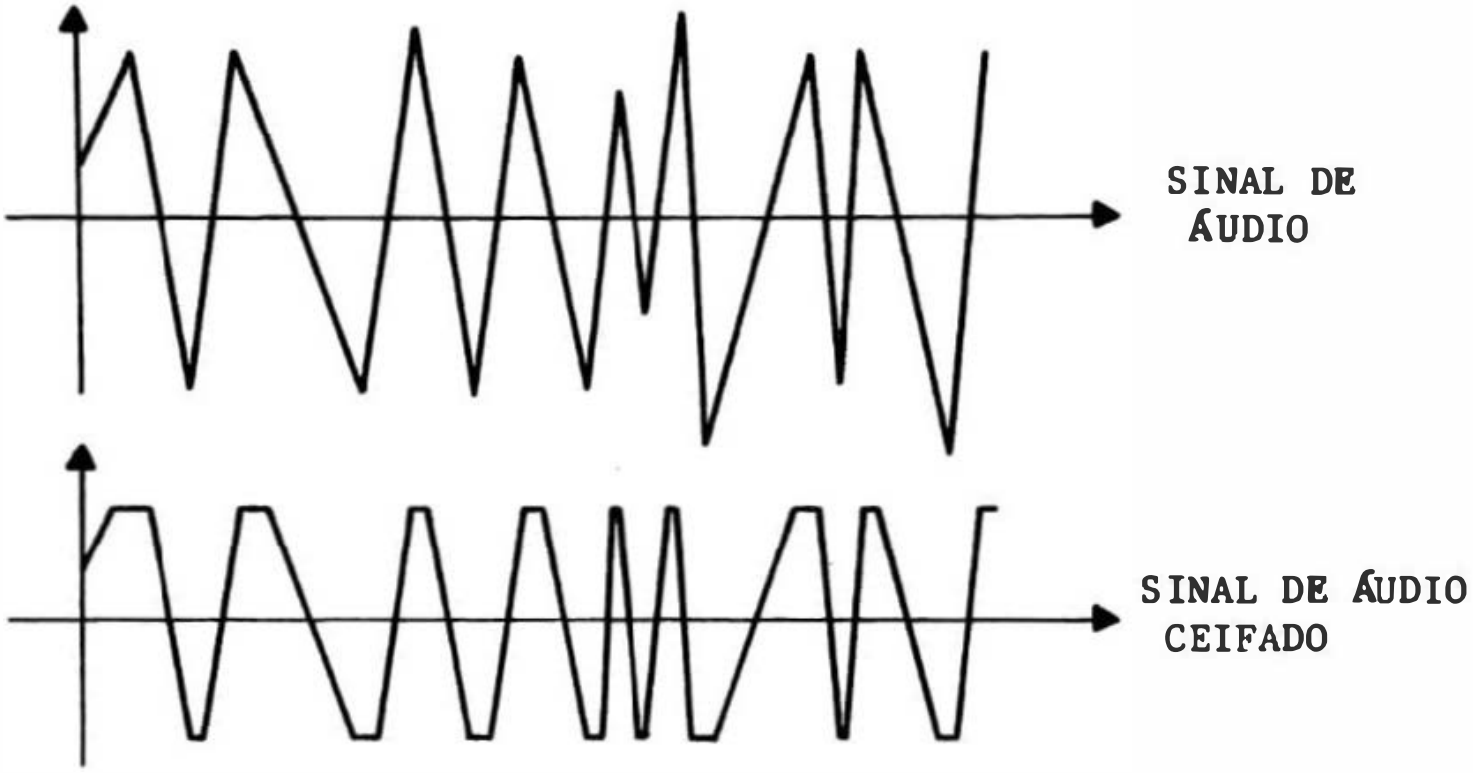
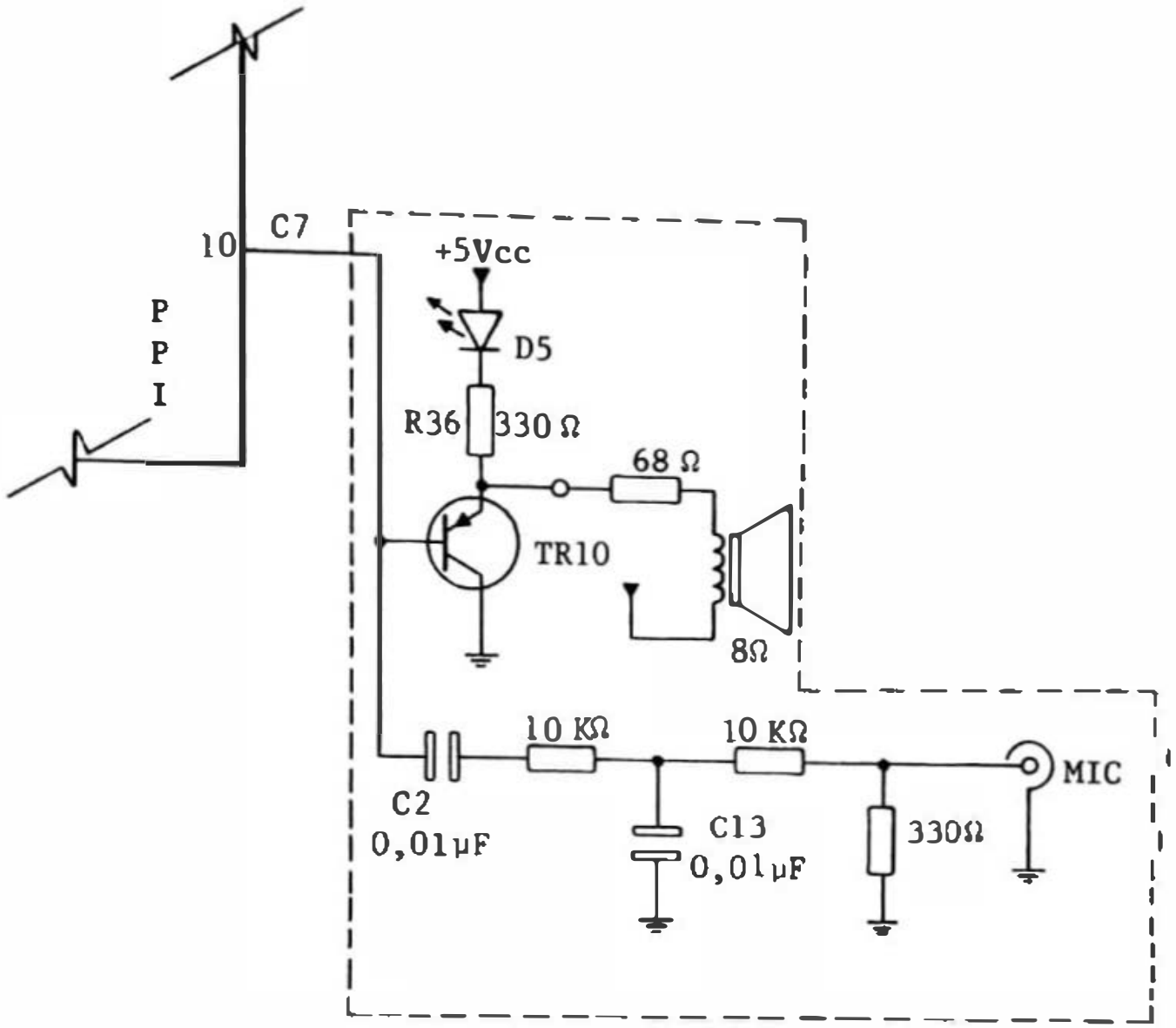


FIGURA 17

As duas portas inversoras do tipo schmitt-trigger do CI_8 (SN 74LS14), utilizadas neste circuito, têm por função "quadrar" o sinal de áudio e ao mesmo tempo evitar que a fase do sinal de áudio não seja invertida.

A saída para gravação de programas e dados em fita cassete (MIC) é compatível com a entrada MIC da maioria dos gravadores cassete. Essa compatibilidade é conseguida através do circuito formado pelo capacitor C_{12} e C_{13} e pelas resistências R_{17}, R_{18} e R_{19}. Esses dispositivos, da forma que estão interligados, operam como filtro e atenuador, tornando o sinal da saída MIC compatível tanto em termos de impedância como em nível de sinal com a entrada MIC da maioria dos gravadores cassete comercializados atualmente. A figura 18 nos mostra o circuito eletrônico, referente ao sistema de gravação em fita cassete MIC.



OPCIONAL

FIGURA 18

O trem de pulso proveniente da porta PC_7 (pino 10 do PPI) poderá ser utilizado para polarizar o transistor TR_{10}, o qual funciona como driver tanto para o sinalizador visual como para o sinalizador sonoro (veja figura 18). Quando os dois sinalizadores operam como indicador de tecla pressionada, teremos um breve acendimento do led D_5 ou um "bip" no alto-falante. Se os circuitos sinalizadores estiverem funcionando como indicador de gravação, durante o período em que o microprocessador estiver transferindo dados para o gravador cassete, o led D_5 permanecerá aceso e o alto-falante emitirá uma série de

bips, cuja freqüência de repetição está diretamente ligada à velocidade de transmissão dos dados.

e) Conector para Opcionais:

Trata-se de conectores, diretamente, ligados aos pinos da CPU e que permitem acionar e controlar os periféricos que possivelmente o aluno poderá adquirir e adicionar ao sistema do microcomputador CEDM-80.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 · CEP 86100 · LONDRINA · PARANÁ

EXAME DA LIÇÃO MC-23

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) O PPI realiza a transferência de dados na forma:

a) paralela;
b) série;
c) paralela e série;
d) N.R.A.

2) Um outro chip muito utilizado para fazer o interfaceamento com os dispositivos externos de E/S é:

a) ULA;
b) USART;
c) CPU;
d) N.R.A.

3) O PPI 8255 possui três portas A, B e C, as quais nos fornecem um total de:

a) 12 linhas de entrada/saída;

b) 8 linhas de entrada/saída;

c) 24 linhas de entrada/saída;

d) N.R.A.

4) A palavra de controle para operação do PPI 8255 no modo zero é:

a) 89_H;

b) 92_H;

c) 80_H;

d) 90_H.

5) De acordo com a tabela verdade do PPI 8255 (tabela 1), a condição ilegal corresponde a:

a) 00010;

b) 11010;

c) 10010;

d) N.R.A.

6) Os endereços das portas A e B são, respectivamente:

a) 00 e 10;

b) 00 e 11;

c) 00 e 01;

d) 11 e 01.

7) O reset do PPI 8255 permite o "zeramento":

a) somente dos registradores internos;

b) somente da palavra de controle;

c) apenas do modo de operação;

d) dos registradores internos e da palavra de controle.

8) O PPI somente será selecionado, caso a CPU envie para a entrada do decodificador, 74LS139 (C.I.5), um pulso baixo, através da linha de controle:

a) $\overline{\text{IORQ}}$;

b) $\overline{\text{NMI}}$;

c) $\overline{\text{MREQ}}$;

d) N.R.A.

9) As linhas do teclado conectadas à porta A (PA_0 a PA_5) permanecem sempre a um nível lógico:

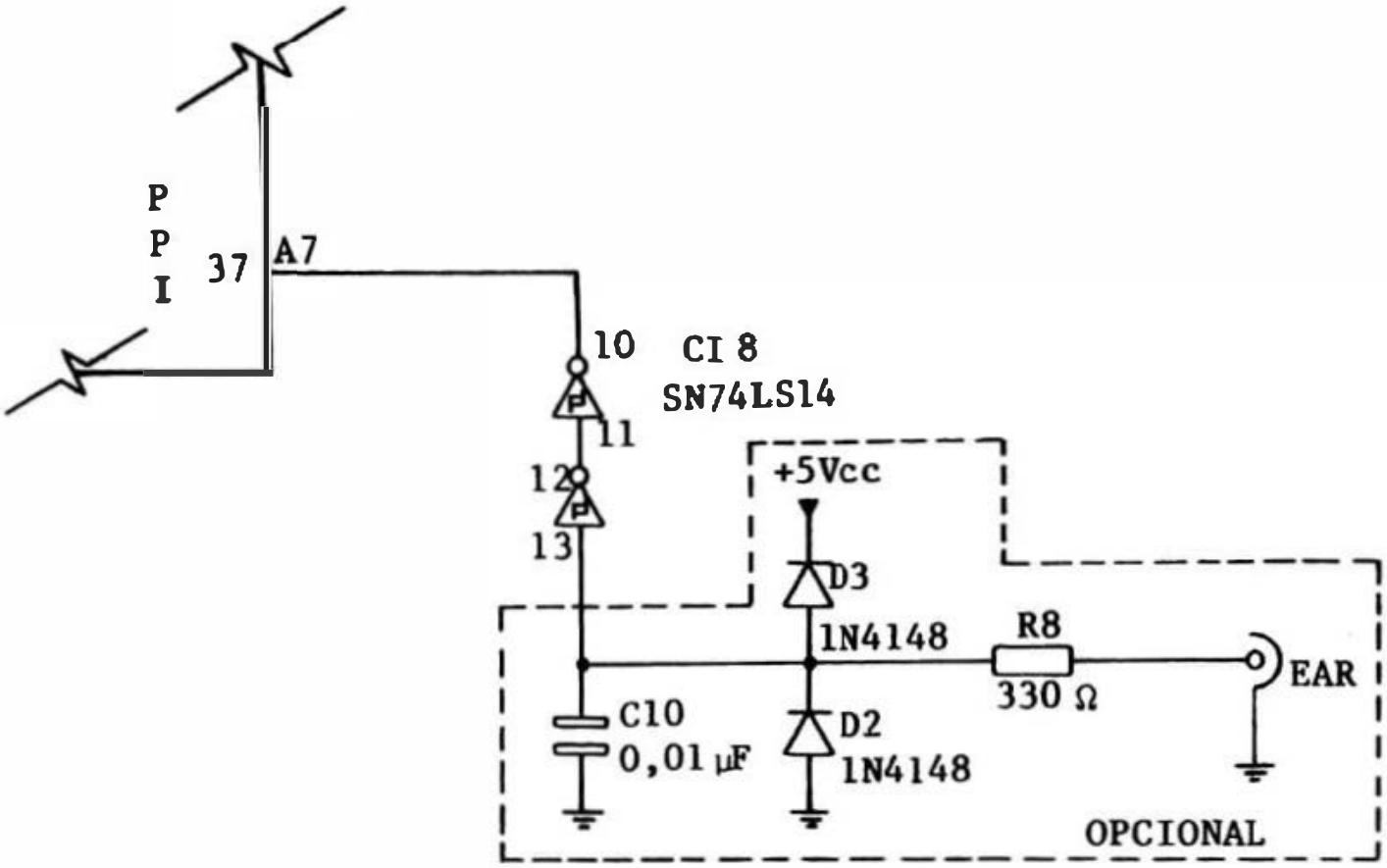
a) alto;

b) baixo;

c) o nível lógico dessas linhas não interfere no processo de varredura do teclado;

d) N.R.A.

10) A figura, a seguir, nos mostra o circuito eletrônico referente ao sistema de:



a) gravação (ou escrita) MIC;

b) leitura EAR;

c) sinalização visual e sonora;

d) N.R.A.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-24

CIRCUITOS INTERFACE

INTRODUÇÃO

Nesta lição, pretendemos desenvolver um breve estudo sobre os circuitos que realizam a interconexão entre um computador e os dispositivos periféricos; esses circuitos são chamados de interface. Antes de passarmos ao estudo, propriamente dito, desses circuitos, vamos conceituá-los.

INTERFACE

De um modo bem genérico, podemos dizer que interface é todo dispositivo capaz de intermediar (ou conciliar) sinais de duas diferentes origens. Em outras palavras, podemos dizer que a interface é a união física e lógica entre dois sistemas que não podem ser diretamente conectados.

A interface de um sistema pode vir acompanhada de um circuito controlador de dispositivos, cuja aplicação se faz necessária para os dispositivos de mecânica complicada. Geralmente, os controladores possuem uma espécie de processador que recebe instruções e as executa, realizando a seqüência de controle, requerida pelo dispositivo em questão. Por exemplo, o controlador encarregar-se-á de fazer avançar um elemento me

cânico, mediante um motor passo a passo, a um certo número de passos. A complexidade destes controladores pode variar notavelmente, indo desde os circuitos que possuem um único circuito integrado, até os que requerem várias placas de circuito impresso.

Atualmente, algumas interfaces utilizam circuitos integrados específicos (LSI) ou microprocessadores, que, na realidade, nada mais são do que pequenos microcomputadores com hardware e software internos, já orientados para uma aplicação específica. Mais adiante, estudaremos alguns destes circuitos integrados, específicos, aplicados em sistemas de microcomputadores.

TÉCNICAS DE ENTRADA/SAÍDA

Existem três técnicas principais de entrada/saída, as quais também são conhecidas como modos de comunicação com o computador, utilizadas para controlar os dispositivos de entrada/saída. São elas:

a) polling (de consultas sucessivas) ou entrada/saída programada;
b) interrupção;
c) DMA (Direct Memory Acess) ou acesso direto à memória.

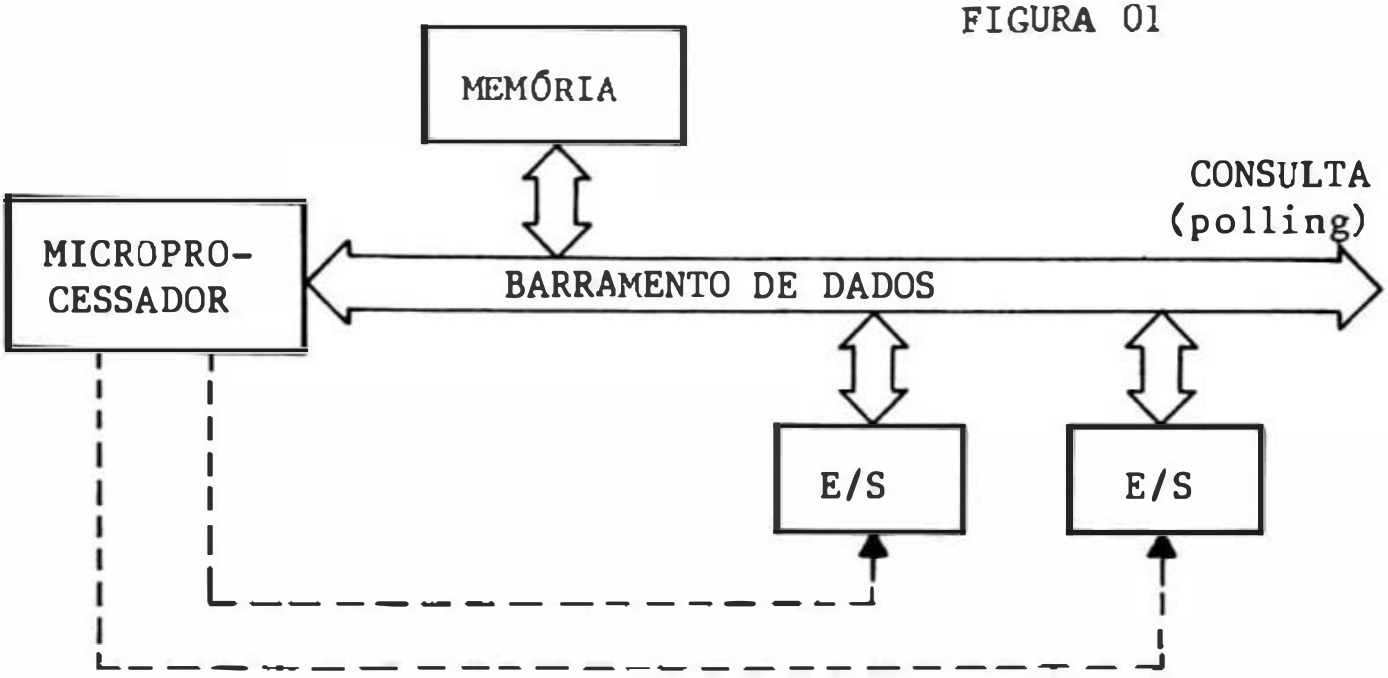
Vejamos, então, em que consiste cada uma dessas técnicas.

a) Polling (ou consultas sucessivas):

Dentre as três técnicas, esta é a mais sensível. Os dispositivos de E/S são conectados, normalmente, nos barramentos de endereços e de dados do sistema. Dependendo do tipo do sistema do microprocessador, faz-se necessário conectar também, um certo número de linhas do barramento de controle. A figura 1 nos mostra um diagrama de blocos, esquemático, referente à esta técnica.

Esta técnica é considerada como do tipo síncrono, pois o computador é o responsável pela sincronização da recepção e transmissão dos dados. Dentre as várias vantagens apresentadas pela técnica de consultas sucessivas, destacamos as seguintes:

FIGURA 01



- a interface, em si, necessita de um mínimo de hardware e não há necessidade de se utilizarem linhas especiais;

- trata-se de uma operação totalmente sincronizada com a execução do programa, o que nos permite saber exatamente quando o dispositivo está sendo solicitado e quanto se demora em dar-lhe serviço.

A principal desvantagem desta técnica é a sobrecarga que ela representa para o programa, pois as consultas são feitas repetitivamente, dispositivo por dispositivo, determinando, de cada vez, se se necessita ou não de alguma atuação.

Esta técnica é recomendada sempre que os programadores não estiverem familiarizados com o projeto de sistemas interruptivos.

b) Interrupção:

Entende-se por interrupção um sinal de hardware externo ao microprocessador, que altera a seqüência normal de execução do programa. Esta técnica deve ser considerada toda vez que a técnica de consultas sucessivas não permitir alcançar um tempo de resposta suficiente, ou quando representar uma grande perda de tempo para o microprocessador. A figura 2 nos mostra um diagrama de blocos esquemático, referente a esta técnica.

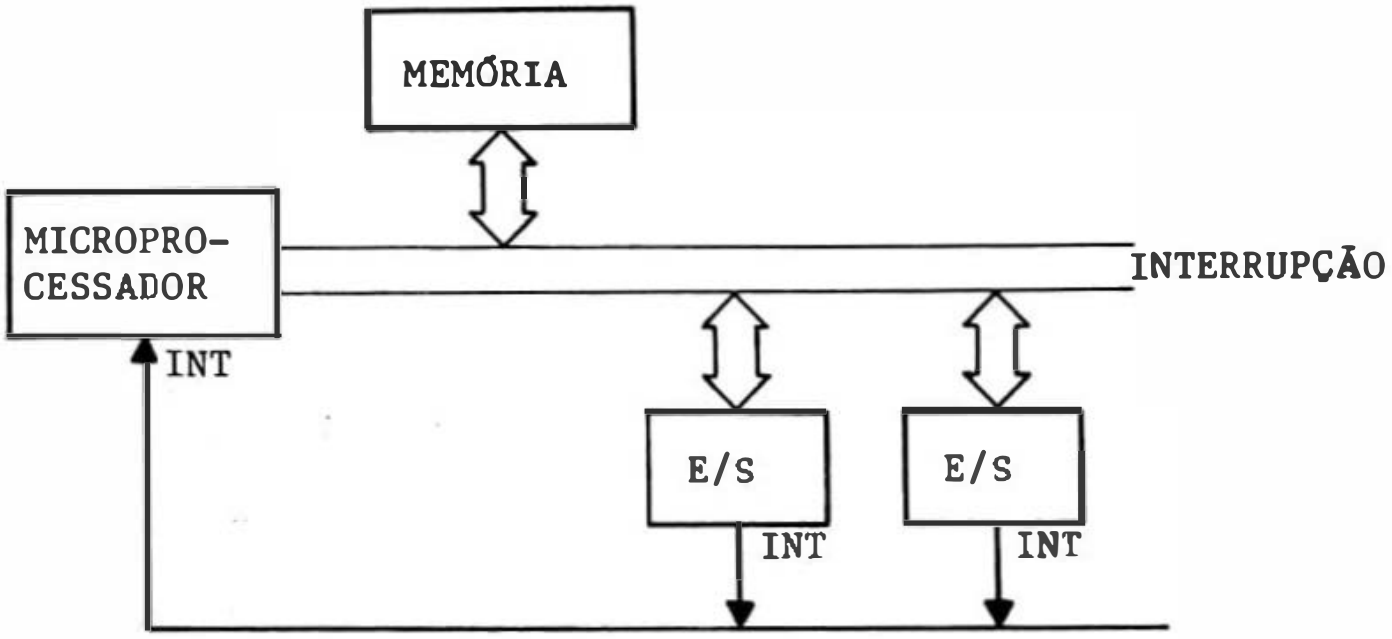


FIGURA 02

Em sistemas interruptivos, os dispositivos de E/S podem solicitar serviços ao microprocessador. Este tipo de sistema utiliza uma linha adicional (veja figura 2), que é a linha de interrupção "INT", conectada ao microprocessador. Todos os dispositivos de E/S são, então, conectados à linha INT. Assim, qualquer dispositivo periférico, que necessite de serviço, pode gerar um nível ou pulso de interrupção através da linha INT. Cabe, portanto, ao microprocessador aceitar a interrupção, identificá-la e dar-lhe o oportuno serviço.

O principal objetivo da interrupção associada à E/S é liberar o processador do loop contínuo, pesquisando o status dos controladores para, em seguida, poder realizar processamento durante o tempo que o controlador leva entre a captura do dado e o final de sua transmissão, no caso de escrita; ou durante o acesso a um dado, no caso de leitura.

Uma vez aceita a interrupção, o microprocessador irá identificar qual o dispositivo que solicitou a interrupção. Ademais, poderá ocorrer que a interrupção tenha sido solicitada simultaneamente por vários dispositivos diferentes. Assim sendo, toda vez que se conectarem vários dispositivos a uma mesma linha de interrupção, deve-se estabelecer uma ordem de prioridades.

Supondo, agora, que a interrupção tenha sido aceita, e o dispositivo identificado, pode-se então executar o serviço solicitado pelo dispositivo em questão.

A execução da rotina de tratamento da interrupção é análoga à utilizada na técnica de consultas sucessivas, ou seja, uma vez concluída a rotina, o microprocessador reinicia a execução do programa que foi interrompido, a partir do ponto em que ocorreu a interrupção.

Para que o programa interrompido possa ser retomado posteriormente, são "salvos" os parâmetros mais importantes da máquina, imediatamente antes de ser iniciada a rotina de interrupção. O método normalmente empregado é o empilhamento dos registros da máquina (PC, status, acumuladores, indexadores). Isso é feito para se evitar que uma segunda interrupção de prioridade maior apague os parâmetros "salvos" pela primeira. O método de organização em pilha permite também o retorno progressivo dos parâmetros "salvos" na ordem inversa em que foram "salvos", ou seja, na mesma ordem em que as rotinas voltam a ser ativadas.

A principal vantagem desta técnica está na rapidez das respostas. Supondo-se que não se apresentem conflitos pela coincidência de solicitações de interrupção, os dispositivos serão servidos imediatamente após efetuada a solicitação.

Dentre as desvantagens que esta técnica apresenta, destacamos:

- cada vez que um dispositivo externo interrompe um programa em curso é necessário uma série de manobras para preservar as operações que deverão ser executadas posteriormente.

- a execução da rotina de tratamento da interrupção é assíncrona com relação ao programa principal.

- requer, normalmente, um circuito (hardware) adicional, principalmente, no caso de necessidade de decidir sobre as prioridades externas.

A figura 3 nos mostra, através de diagrama de blocos, como se processa esta técnica.

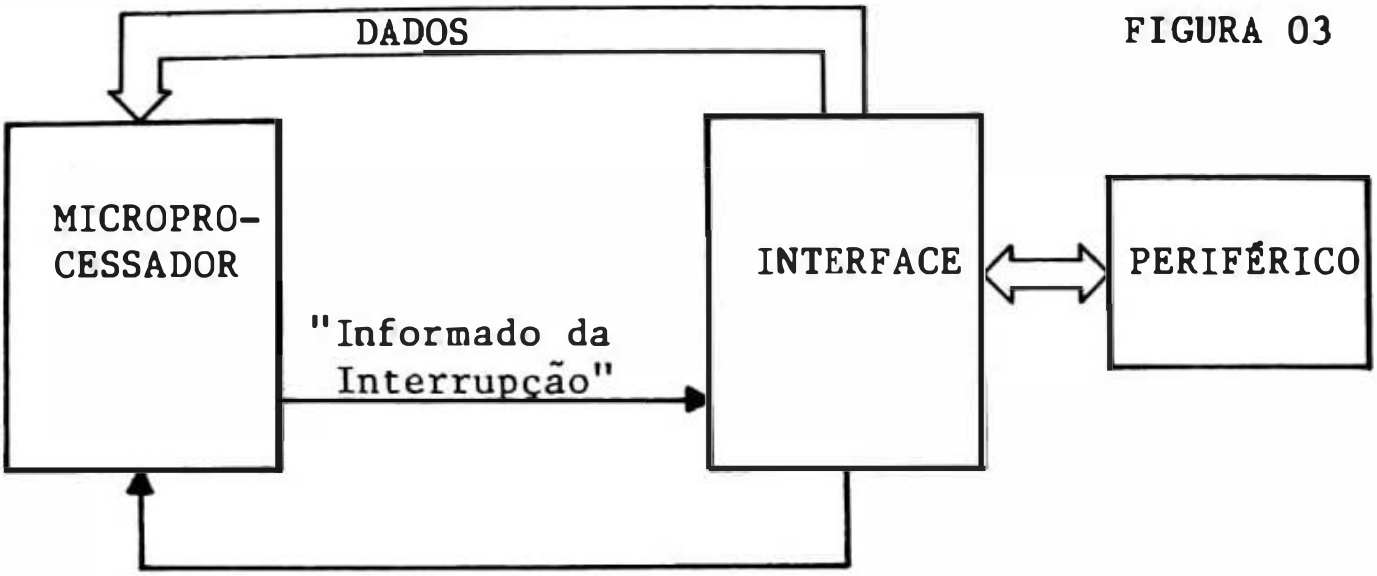


FIGURA 03

OBS.: O microprocessador interrompe o programa que está executando, processa a rotina de interrupção e volta a executar o programa.

Esta técnica é muito utilizada em sistemas que operam em tempo real, nos quais é fundamental a garantia de um tempo mínimo de resposta, perante às solicitações externas. O uso de interrupções para E/S leva a um aumento da capacidade de processamento da máquina, porém a taxa de transferência de dados é prejudicada, uma vez que esta fica limitada pelo tempo que o computador leva para reconhecer a interrupção, salvar seus parâmetros e executar a rotina de serviço de interrupção. Com vista a superar essa limitação, aplica-se a técnica chamada acesso direto à memória ou DMA, a qual passaremos a estudar em seguida.

c) DMA - acesso direto à memória:

Trata-se de um recurso apenas de hardware, no qual a interface assume o controle dos barramentos e transfere dados diretamente para a memória, sem que o microprocessador venha a participar ou tomar conhecimento dessa operação. Para que esta operação possa ser realizada, torna-se necessário o uso de um dispositivo, que sincronize os acessos à memória pela CPU e pelos controladores (ou interfaces), que se utilizam do DMA. Esse dispositivo é o chamado controlador de memória. Atualmente alguns microprocessadores já dispõem, internamente, de

toda lógica necessária para realizar o sincronismo com o DMA, através do sinal "hold", o qual faz com que o microprocessador entre num estado de espera, logo após termine de executar a instrução corrente, com seus barramentos de dados e endereço em três-estados (tri-state).

Após completar toda essa operação, o microprocessador envia um sinal "holda" (hold acknowledge), indicando para que a interface (controlador) prossiga. Terminado o acesso direto à memória, o controlador desativa o sinal "hold" do microprocessador. Toda essa seqüência de sinais "hold" e "holda" recebe o nome de protocolo, cujo objetivo principal é justamente a obtenção de sincronismo entre acessos diretos à memória pelos diversos elementos do sistema.

A figura 4 ilustra o princípio de funcionamento do controlador DMA.

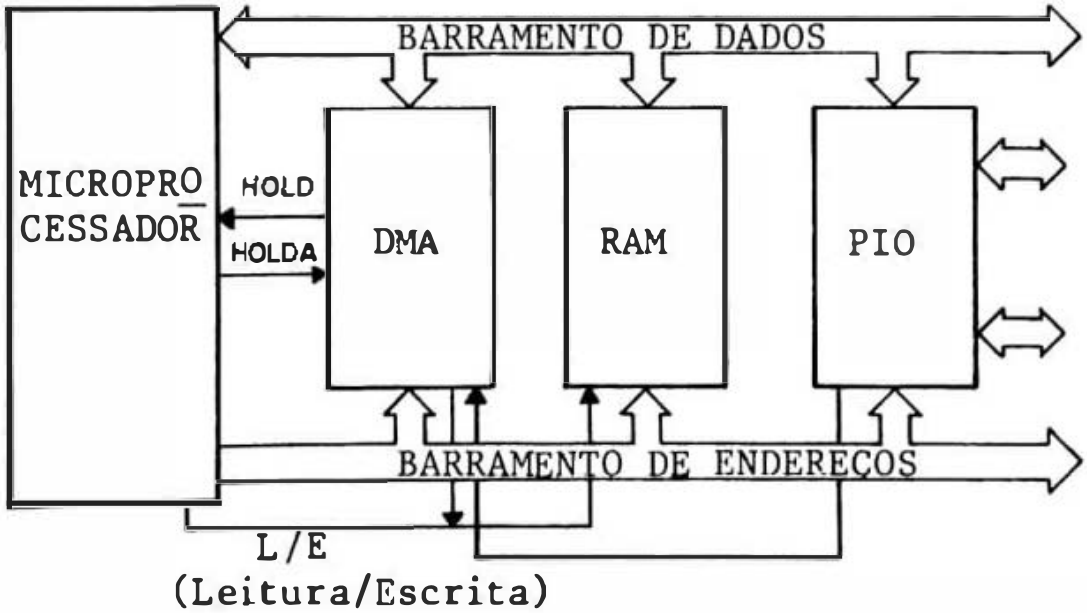


FIGURA 04

Normalmente, os controladores DMA possuem um grau de complexidade próximo ao dos microprocessadores, o que resulta em um custo relativamente maior.

A principal vantagem desta técnica sobre a técnica de interrupções está no fato de não haver mudança de contexto para o microprocessador, dispensando-se o tempo de "fetch" e decodificação de instruções da rotina de interrupção, o que torna este processo (DMA) bem mais rápido.

O acesso direto à memória é usado, sempre que a velocidade de transferência de blocos do microprocessador não é suficiente, nem se quer no sistema de interrupções; nestes casos, torna-se viável a utilização de um controlador DMA, o qual se encarregará então das transferências rápidas de palavras ou blocos. A utilização de um controlador também é justificada no caso em que se utilizam dispositivos de E/S de alta velocidade, como, por exemplo, as unidades de discos ou os terminais de vídeo com TRC (tubo de raios catódicos).

Uma vez estudado os métodos ou técnicas de entrada/saída, vamos agora passar ao estudo dos dois tipos de interface de entrada/saída (E/S) programáveis, os quais são largamente aplicados nos sistemas de microcomputadores.

INTERFACES DE E/S PROGRAMÁVEIS

Dois são os tipos fundamentais de circuitos integrados, utilizados em uma interface digital: as interfaces para transferências em série e em paralelo. Tais interfaces são também conhecidas como interfaces universais e, geralmente, são fornecidas em um único chip. Elas oferecem ao usuário um alto grau de flexibilidade. Eis algumas das tarefas básicas, desempenhadas pelas interfaces programáveis:

a) Controle do fluxo bidirecional de dados paralelos entre a CPU e o dispositivo periférico;

b) Controle da transmissão bidirecional de dados em série, entre a CPU e o modem;

c) Tratamento das interrupções dos periféricos para a CPU;

d) Execução das transferências do DMA (acesso direto à memória) entre a CPU e as memórias auxiliares.

Passemos então ao estudo desses dois tipos de interfaces programáveis.

INTERFACES PROGRAMÁVEIS PARA TRANSFERÊNCIAS EM SÉRIE

Representando este tipo de interface, temos dois dispositivos: UART e USART. Ambos são amplamente utilizados para interfacear um ou mais dispositivos periféricos com a CPU. Vejamos, então, em que consiste cada uma delas.

UART:

O "transmissor/receptor assíncrono universal", ou simplesmente UART, foi um dos primeiros circuitos integrados LSI estander. Daí o fato de que a quase totalidade das UART atuais serem praticamente idênticas, ou seja, suas características, exceto sua velocidade, são similares.

Uma UART estander possui três secções: a de recepção, a de transmissão e a de controle. A figura 5 nos mostra um diagrama esquemático de uma UART.

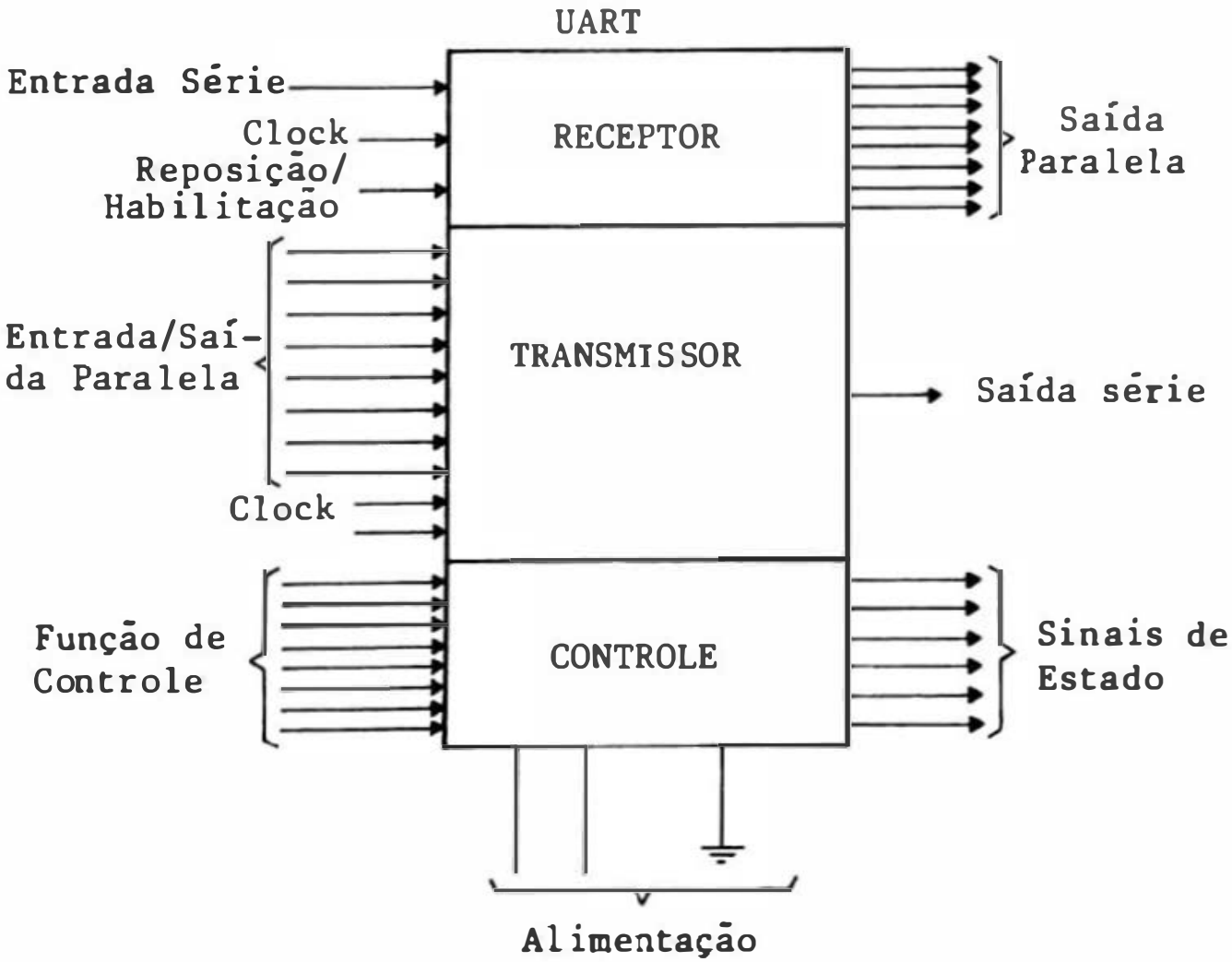


FIGURA 05

Atualmente, existe no mercado uma variedade de UART. Muito embora, cada circuito integrado UART possua seus próprios dispositivos únicos, genericamente desempenham as seguintes funções básicas:

a) Converte os dados, que chegam em forma paralela, para a forma em série, a fim de que possam ser aceitos pelo sistema de microcomputador;

b) Converte os dados em paralelo, que saem do sistema de microcomputador, em formatos série para que possam ser aceitos pelos periféricos que operam com corrente de bits em série, por exemplo, modem;

c) Durante as conversões de paralelo para série (função de transmissão), a UART gera os bits de paridade apropriados;

d) Durante uma operação de transmissão, a UART estrutura as palavras de dados (incluindo o bit de paridade, se for o caso) com os bits adequados de partida e parada;

e) Durante as conversões de dados em série para paralelo (função de recepção), os bits de partida e parada são reconhecidos e eliminados do processamento posterior;

f) Durante as operações de recepção, é verificada a paridade da palavra e, se acaso houver algum erro de paridade, este será sinalizado;

g) A UART recebe e atua sobre os sinais de estado, enviados pelos periféricos e enviam sinais para os dispositivos periféricos e para a CPU, indicando o seu atual estado. A UART também envia comandos específicos para os periféricos;

h) Durante uma operação de recepção, a UART cronometra e sincroniza os dados em série, em uma média compatível com o sistema de clock do microcomputador.

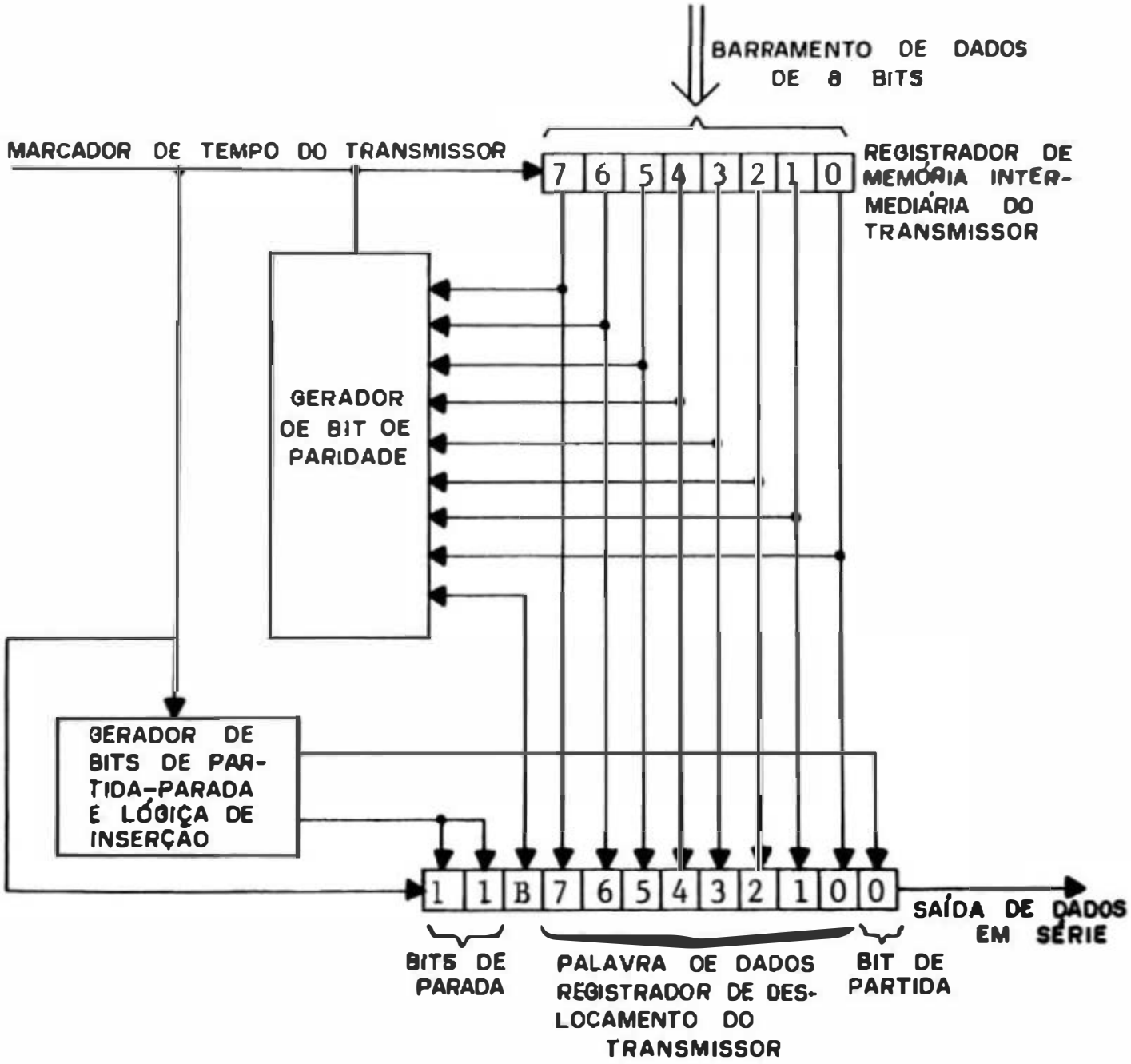
Resumidamente, podemos dizer que a função básica da UART é a conversão série/paralela. Todas as funções citadas há pouco são realizadas pela UART, durante o processo de serialização/desserialização.

O tamanho (ou comprimento) das palavras utilizadas pela UART pode ser de 5, 6, 7 ou 8 bits. A elas serão acrescentados os bits de partida e parada (que podem ser 1, 1 ½ ou 2 em código de 5 bits) e o bit de paridade, par ou ímpar. Este último (bit de paridade) pode, ou não, ser utilizado.

A figura 6 nos mostra como é a formação da palavra de dados no registrador de deslocamento do transmissor da UART. O

bit de paridade é gerado adequadamente (no gerador de bit de paridade) para a palavra de dados em particular e inserido no circuito biestável, marcado com um "B", no registrador de deslocamento do transmissor.

FIGURA 6



Por outro lado, a figura 7 nos mostra como a memória intermediária dupla é utilizada na seção receptora da UART. Primeiramente, é feita uma verificação de um bit de partida legítimo nos sinais seriais que chegam. Assim que um bit de partida, adequado, é reconhecido, os 11 bits seguintes (8 bits de dados, 1 bit de paridade e 2 bits de parada) são deslocados para o interior do registrador de deslocamento do receptor.

Em seguida, os 2 bits de parada são enviados para a lógica de verificação de bits de parada, a fim de se determinar se o caracter de chegada foi corretamente estruturado. Os 8 bits de dados são transferidos para o registrador de memória intermediária do receptor e, simultaneamente, a lógica de verificação do erro de paridade é alimentada, através do envio do bit de paridade. Caso a paridade não esteja correta, um sinal é enviado para a CPU. Durante este período, os 8 bits de palavra de dados são mantidos no registrador de memória intermediária, receptora, podendo ser transferidos para o microcomputador a qualquer instante, no decorrer do período em questão.

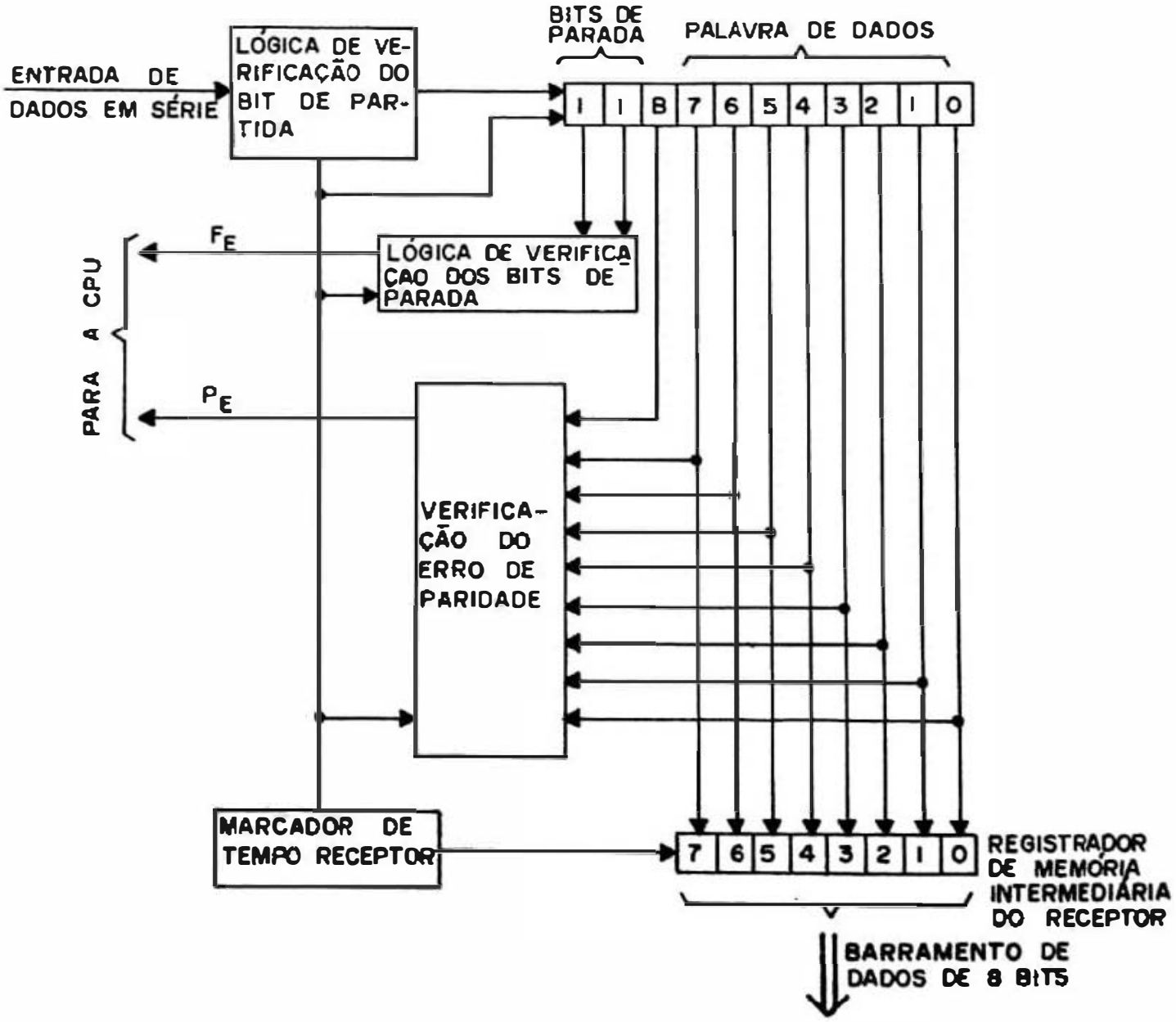
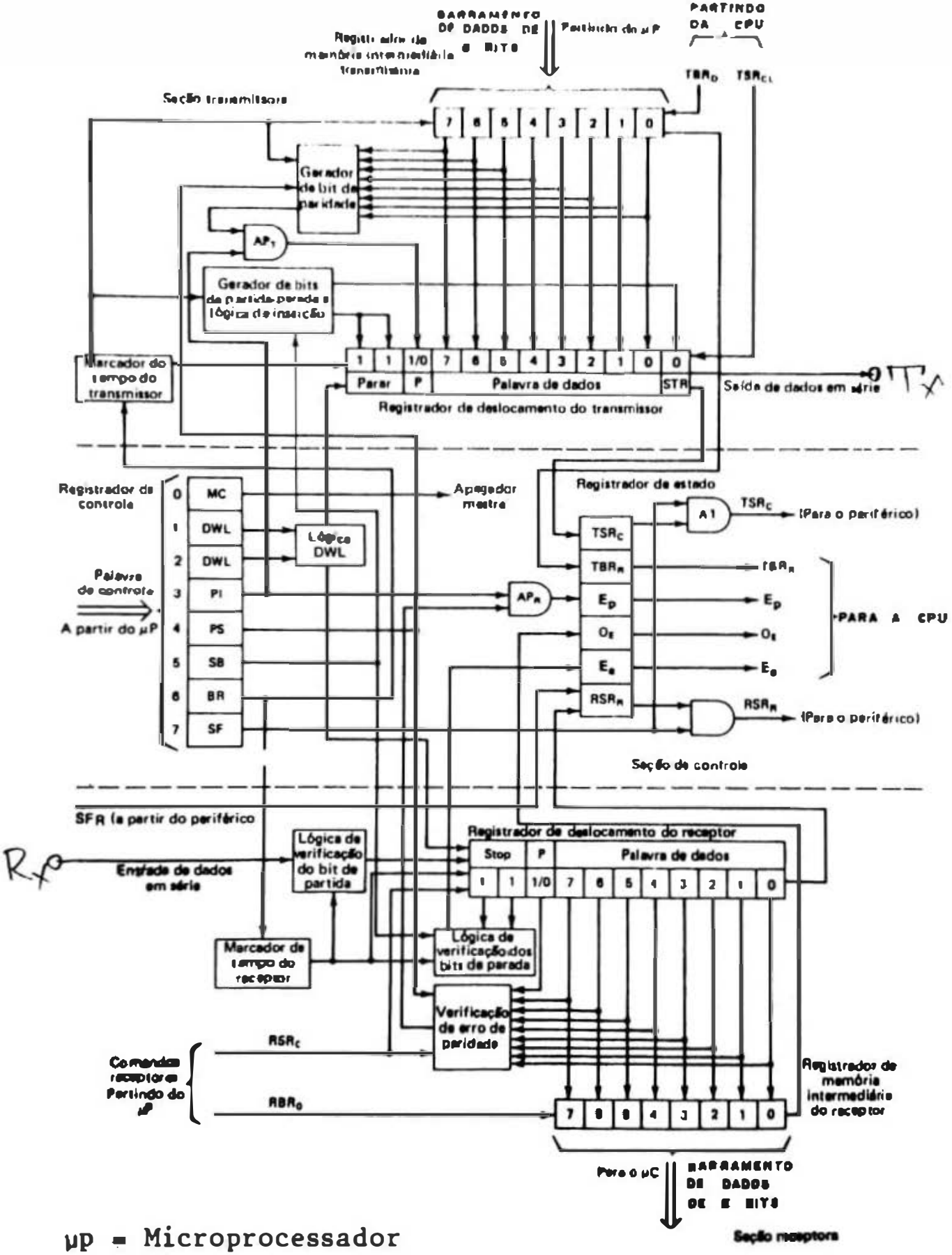


FIGURA 7

A figura 8, por sua vez, nos mostra o diagrama de blocos conceitual (teórico) de uma UART típica.

FIGURA 8



μp = Microprocessador
μc = Microcomputador

A UART, apresentada na figura 8, pode operar em uma modalidade bidirecional, simultaneamente, ou seja, a transmissão e recepção de dados em série podem ocorrer simultaneamente, tendo em vista o fato das seções de recepção e transmissão serem totalmente independentes, muito embora sejam controladas pelo mesmo registrador de controle.

Finalizando, podemos dizer, com relação à aplicação da UART, que esta pode ser aplicada, em geral, na comunicação com os dispositivos série de entrada/saída (E/S), tais como teletipos, impressoras ou modens (conectados a uma linha telefônica, por exemplo).

As memórias intermediárias aqui citadas são pequenas memórias de características especiais. Como exemplo, podemos citar as pilhas (LIFO) e as filas (FIFO).

USRT e USART:

A USRT é um transmissor/receptor universal síncrono, utilizado em sistemas de microcomputador, que opera na modalidade síncrona.

A transmissão de dados na modalidade síncrona é caracterizada por sua alta velocidade e por operar com blocos de dados. Se os dispositivos transmissores não possuírem dados legítimos para enviar, eles inserem caracteres simulados, chamados SYNC, nos furos em branco.

Ainda com relação ao modo síncrono de transmissão, cabe ressaltar que cada fluxo de dados é sempre precedido por uma palavra de identificação predeterminada, que sempre tem um padrão estabelecido de "0s" e "1s". No padrão SYNC ou padrão de identificação do código ASCII, a palavra de identificação é 01101001. Normalmente, são enviados dois desses padrões, consecutivamente, adiante da palavra de dados. A lógica do equipamento receptor detecta e decodifica estes modelos e estabelece a lógica para capturar os próximos 8 bits que entram serialmente. Em seguida, a palavra série é convertida para a forma paralela e enviada para o sistema do microcomputador.

Já, nas transmissões assíncronas, quando o dispositivo transmissor não tem nenhum dado legítimo para enviar, ele simplesmente para de transmitir. Assim sendo, nenhum caracter SYNC será inserido, decorrendo daí o fato do fluxo de bits em

série possuir espaços em branco ou ter quebras no fluxo de dados.

A USRT é utilizada apenas nas transmissões síncronas em série.

Quando as transmissões síncrona e assíncrona em série estiverem combinadas em um único circuito integrado, nós teremos a USART (transmissor/receptor síncrono/assíncrono universal). Para a USART, um bit de modalidade adicional é incluído na palavra de controle. É justamente este bit que, durante o processo de inicialização, irá colocar o chip em operação em uma das modalidades síncrona ou assíncrona.

A figura 9 nos mostra um exemplo típico de aplicação de uma USART em um sistema de microcomputador.

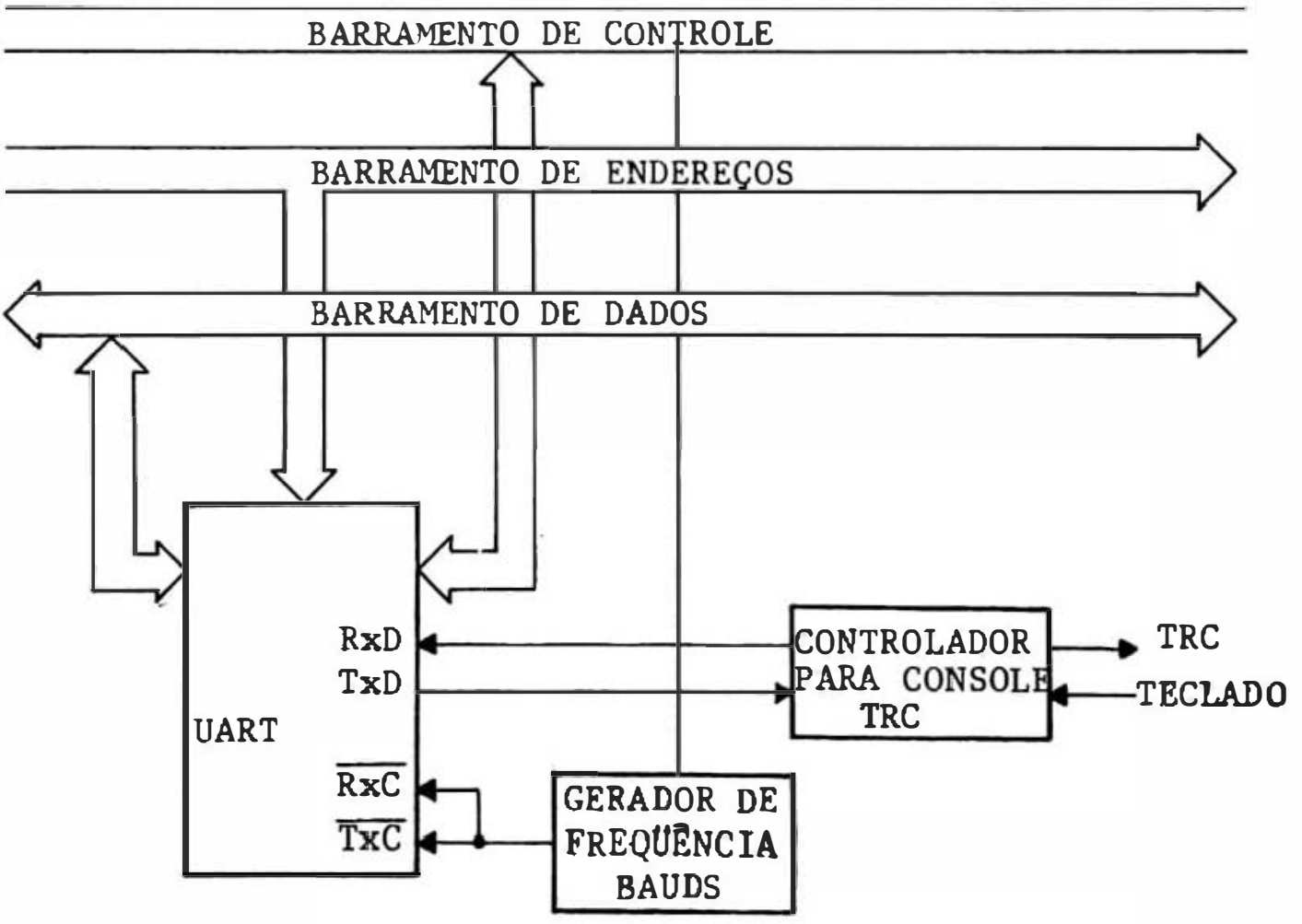


FIGURA 09

INTERFACES PROGRAMÁVEIS PARA TRANSFERÊNCIAS EM PARALELO

Infelizmente, não existe um formato universal de dados, que seja aplicado no caso de transferências paralelas. Essas interfaces possuem comprimento de palavra, tempo e intercâmbio de sinais variados, o que torna difícil o projeto de uma interface, que possa ser aplicada ou utilizada universalmente. Com isto, não se conseguiu ainda estabelecer normas para estes dispositivos, como no caso das UARTS.

Uma vez que inexiste normalização específica sobre o assunto, os fabricantes de interfaces programáveis para transferências em paralelo, colocam nomes diferentes em seus produtos. Assim, encontraremos a MOTOROLA com suas "PIAs", a INTEL com seus "PPIs", a Rockwell com seus "PDCs" e a ZILOG com seus "PIOs". De um modo geral, pode-se dizer que as características destes circuitos integrados, com relação às suas funções, são muito parecidas.

Apenas, a título de exemplo, apresentaremos, a seguir, alguns comentários sobre os elementos PIO e PPI.

PIO (entrada/saída paralela):

Trata-se de um dispositivo programável de interface, cuja missão é permitir uma interface básica de entrada e saída para dados de 8 bits em paralelo.

Normalmente, toda interface paralela de entrada/saída (E/S) deve dispor, pelo menos, de um registro de entrada e outro de saída, uns bits de estados e algum circuito lógico de interrupção. Porém, só isto não basta. Oito linhas não são suficientes para a maioria das aplicações de entrada/saída. As aplicações típicas requerem, pelo menos, de 16 a 24 linhas de entrada/saída. Será muito complicado e antieconômico conectar um circuito de interface de 8 bits para cada grupo de 8 linhas. O circuito de interface de uso geral há de possuir vários canais (grupos de 8 linhas). O PIO deverá, geralmente, multiplexar uma única conexão com o barramento de dados da CPU com dois ou três canais externos. Cada um desses canais dispõe de seus próprios registros com a informação de estado adequada. Seria bom poder contar com o máximo possível de canais ou portas de entrada/saída, porém isto é limitado, na prática, pelo número máximo de pinos no circuito integrado (geral

mente 40 pinos). Desta dorma, dispõe-se no máximo de três portas de entrada/saída e a maioria dos circuitos existentes incluem duas ou três portas paralelas.

Duas são as formas de se programar uma PIO:

1 - As funções da lógica de controle são programáveis para cada porta em particular. O usuário pode especificar, por exemplo, a linha que deseja utilizar para o diálogo (troca de informações) com o dispositivo, a direção (ou endereço) que estas linhas usarão e, especificamente, sua função. Pode-se decidir, também, se um sinal tem que gerar interrupções ou não, se há de gerar pulsos ou níveis, operar com lógica positiva ou negativa, etc.

2 - A PIO possui linhas de dados programáveis. Cada linha (ou grupo de linhas) de dados de uma porta ou canal possui seu endereço programável. Cada linha pode ser definida, individualmente, como entrada ou saída. Com uma PIO se pode programar todas as linhas como saídas e, mais tarde, fazê-las operar como entradas. Isto faz com que a PIO seja considerada como uma interface de uso geral, podendo ser utilizada em qualquer situação.

A figura 10 nos mostra o diagrama de blocos esquemático de uma interface PIO.

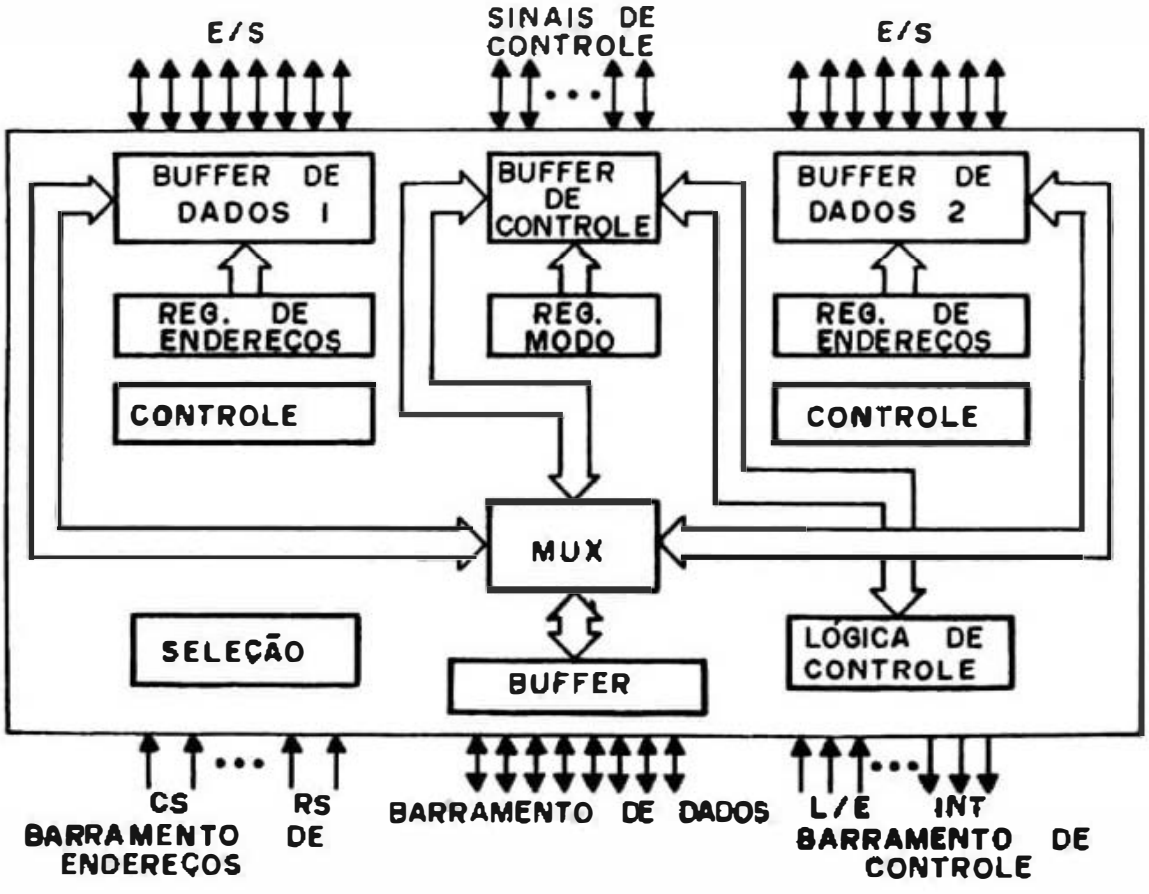


FIGURA 10

Note que, na estrutura da PIO, mostrada na figura 10, ela dispõe de duas portas (de 8 bits cada uma) para entrada/saída mais as linhas de controle, barramento de dados e barramento de endereços. Note também que cada uma das portas possui três registros (ou registradores): buffer de dados, registro de endereço e registro de controle.

A PIO é um circuito de concepção muito sensível. Uma de suas vantagens principais é a de ser programável. Sua capacidade de programação é de ação muito rudimentar. Prevê-se que, no futuro, a potência das PIOs aumente consideravelmente com a inclusão de mais funções programáveis e, com a inclusão de pequenos processadores elementares que permitirão efetuar diversas operações à nível local.

PPI (Interface de Periféricos Programável)

Trata-se de um chip (circuito integrado) LSI, utilizado em microcomputadores, servindo de interface entre a máquina (microcomputador) e os dispositivos externos de entrada/saída (E/S).

O PPI 8255 fabricado pela Intel é atualmente o representante mais divulgado e conhecido da família das interfaces de periféricos programáveis. Esta interface foi projetada para trabalhar com os microprocessadores 8080 ou 8085, podendo, entretanto, ser utilizada em sistemas com outros microprocessadores, como, por exemplo, o microprocessador Z-80.

A interface de comunicação paralela 8255 é um dispositivo programável pelo microprocessador e pode operar em três modalidades:

- Modo 0:

Neste modo, as portas A, B e C se encontram programadas para servirem, basicamente, de entradas ou saída de dados.

- Modo 1:

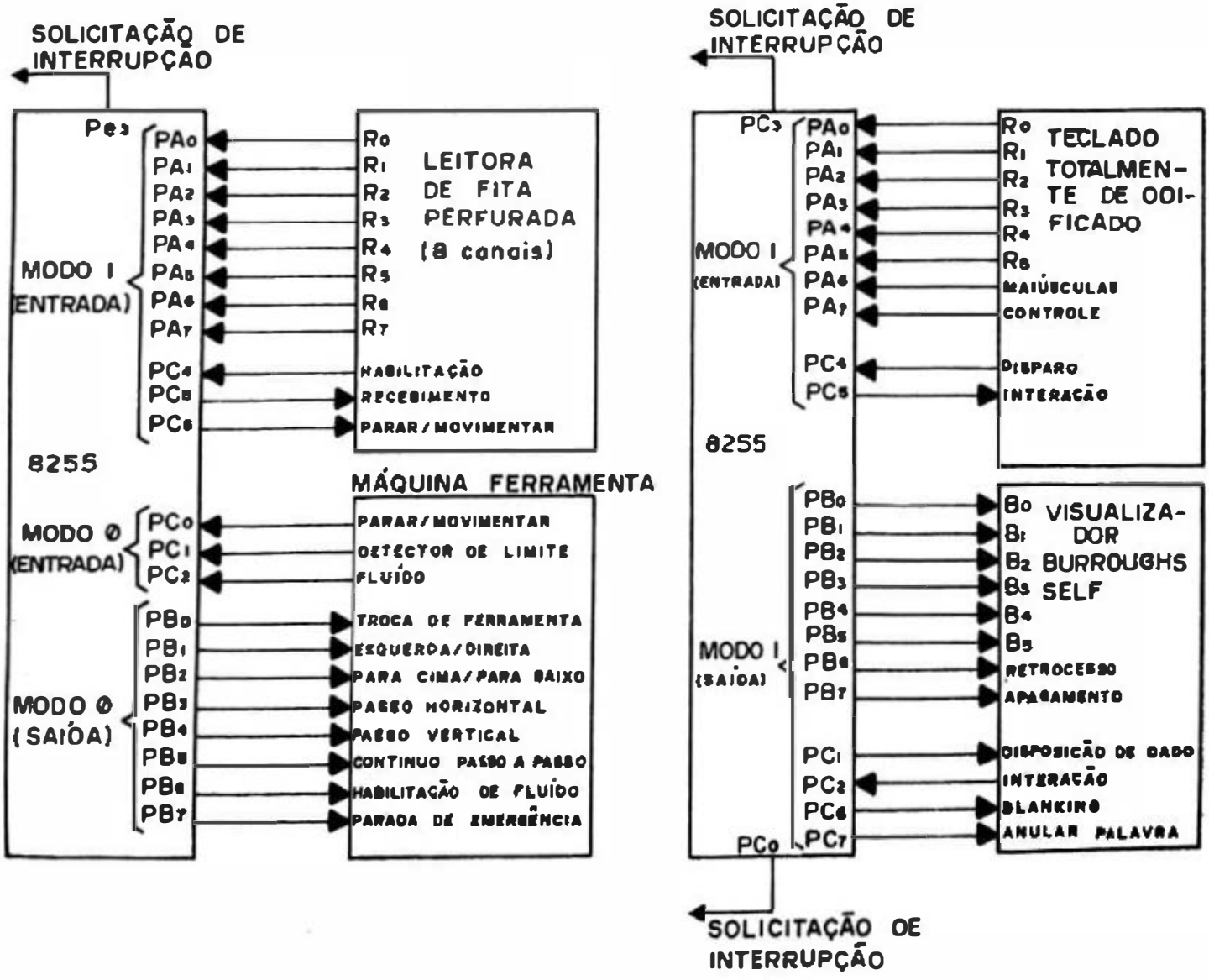
As portas A e B encontram-se programadas para servirem de entradas ou saídas de dados, porém deverão estar sincronizadas por meio de sinais especiais de controle, ligados à porta C.

- Modo 2:

Neste modo de operação, a porta B não é utilizada, uma vez que a porta A irá constituir-se em um barramento de dados bidirecional de entrada e saída. Os bits da porta C serão utilizados para sinais de controle, de forma semelhante ao que ocorre no modo 1.

O PPI 8255 pode operar em um desses três (3) modos, em separado, ou ainda, poderá trabalhar por meio de combinações destes modos. A figura 11 nos mostra algumas aplicações típicas do PPI 8255.

FIGURA 11



(APLICAÇÃO 1)

(APLICAÇÃO 2)

Mais informações sobre o PPI 8255 o aluno encontrará na lição MC-23 onde um estudo mais detalhado sobre o mesmo foi realizado.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-24

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Interface é:

a) todo dispositivo capaz de consiliar sinais de uma única origem;

b) todo dispositivo capaz de consiliar sinais de duas diferentes origens;

c) a união física e lógica entre dois sistemas que não podem ser diretamente conectados;

d) os itens b e c estão corretos.

2) As técnicas de entrada/saída também são conhecidas como:

a) modos de interpolação com o periférico;

b) modos de comunicação com o computador;

c) polling;

d) interrupção.

3) A técnica polling é considerada do tipo:

a) síncrono;

b) assíncrono;

c) interruptiva;

d) N.R.A.

4) Os sistemas interruptivos utilizam uma linha adicional que é a linha de:

a) controle síncrono;

b) reset;

c) interrupção INT;

d) N.R.A.

5) A técnica DMA é, na essência, apenas um recurso de:

a) software;

b) hardware;

c) interrupção;

d) N.R.A.

6) O sincronismo entre o microprocessador e o DMA, em muitos microprocessadores, é conseguido através do sinal:

a) INT;

b) RESET;

c) de clock;

d) HOLD.

7) Uma UART estander é constituída pela:

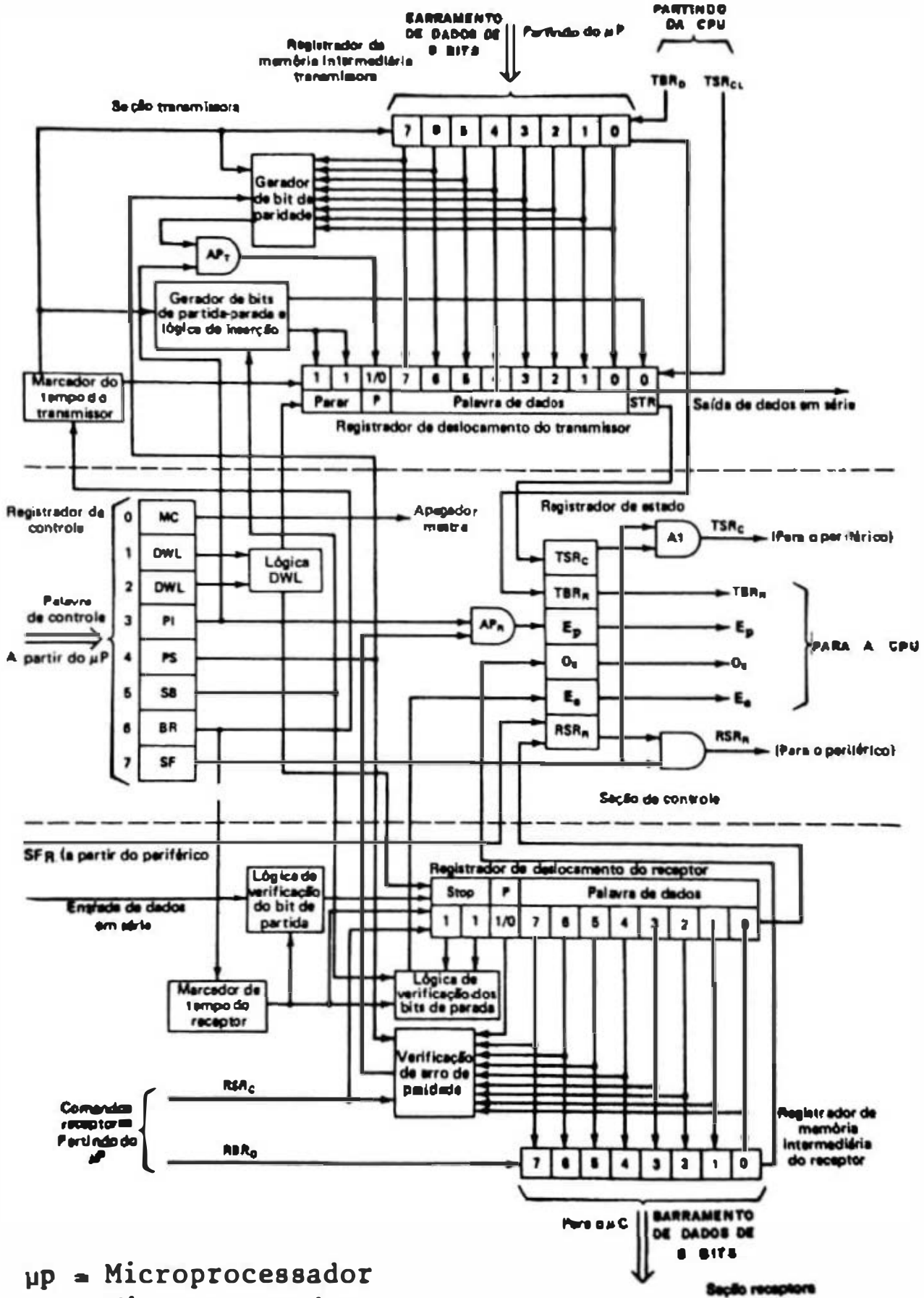
a) seção de recepção;

b) seção de transmissão;

c) seção de controle;

d) todas as respostas estão corretas.

8) A figura, a seguir, nos mostra:



μp = Microprocessador
μc = Microcomputador

a) o diagrama de blocos de uma UART;

b) o diagrama de blocos de um PPI;

c) o diagrama de blocos de uma USART;

d) N.R.A.

9) A USART é um transmissor/receptor universal no qual temos:

a) somente o modo síncrono de transmissão;

b) o modo síncrono e assíncrono, combinados em um único circuito integrado;

c) somente o modo assíncrono de transmissão;

d) N.R.A.

10) A PIO é um dispositivo programável de entrada/saída para dados de:

a) 18 bits em série;

b) 18 bits em paralelo;

c) 8 bits em paralelo;

d) N.R.A.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-25

GRAVADORES DE PROM E EPROM

INTRODUÇÃO:

Inicialmente, apresentaremos uma rápida revisão sobre as memórias semicondutoras e suas características. E, em seguida, passaremos ao estudo dos gravadores de PROM e EPROM.

MEMÓRIAS SEMICONDUTORAS (Revisão):

Na eletrônica e, mais especificamente, no que tange o tratamento da informação, nós dispomos basicamente de dois tipos principais de memórias: RAM (Randon Access Memory - memórias que se podem ler e escrever) e ROM (Read Only Memory - memórias de apenas leitura), esta última destinada a armazenar permanentemente uma determinada informação.

Como sabemos, as memórias RAM são aplicadas, em geral, para o armazenamento de dados intermediários. Em uma memória RAM, podemos escrever uma determinada informação e, mais tarde, podemos ler esta informação, sem que esta se perca. As memórias RAM são memórias voláteis, ou seja, se desconectarmos a sua fonte de alimentação, todas as informações nelas contidas se perderão.

Por outro lado, nas memórias ROM, que são normalmente utilizadas para armazenarem programas operativos e programas fixos, podemos ler as informações nelas contidas em qualquer momento, sem que se perca a informação, mesmo que a fonte de alimentação seja desconectada. Daí o fato da memória ROM também ser conhecida como memória não volátil.

A memória ROM foi assim denominada por volta de 1950. Conhecida como a ROM de máscara, ou seja, para programá-la, era necessário, primeiramente, fabricar uma máscara e depois a memória ROM. Para se substituir um determinado programa, era preciso trocar a máscara, o que implicava em um período de espera de várias semanas por parte dos consumidores finais, pois o fabricante tinha que fazer uma nova máscara (veja processo de fabricação de circuitos integrados) para, em seguida, obter-se a memória ROM.

Com o avanço tecnológico, nasceram as memórias PROM (ROM programável) ou memórias programáveis, não voláteis, que, se introduzidas em uns aparelhos, chamados programadores ou gravadores de memórias, o próprio usuário pode gravar o seu programa e testá-lo diretamente.

Após o aparecimento das memórias PROM, surgiram as memórias EPROM (Erasable Programmable Read Only Memory). Estas memórias podem ser gravadas pelo próprio usuário e as informações nelas contidas podem ser apagadas, bastando, para isso, expor a EPROM a uma radiação ultravioleta por algum tempo, permitindo, com isso, que novas informações ou programa possam ser introduzidas. É impossível, na prática, apagar somente parte do conteúdo de uma memória EPROM. Se as especificações de gravação, enviadas pelo fabricante, forem mantidas corretamente, em média, pode-se fazer cerca de 100 "apagamentos" em uma memória EPROM.

A figura 1 nos mostra a construção de uma célula de memória, segundo o procedimento EPROM. Esta célula de memória é constituída de apenas um transistor FAMOS com dois eletrodos de porta (gate). A porta superior é a de controle e encontra-se conectada ao condutor fila, enquanto que a porta inferior é uma porta aberta e serve para armazenar a informação.

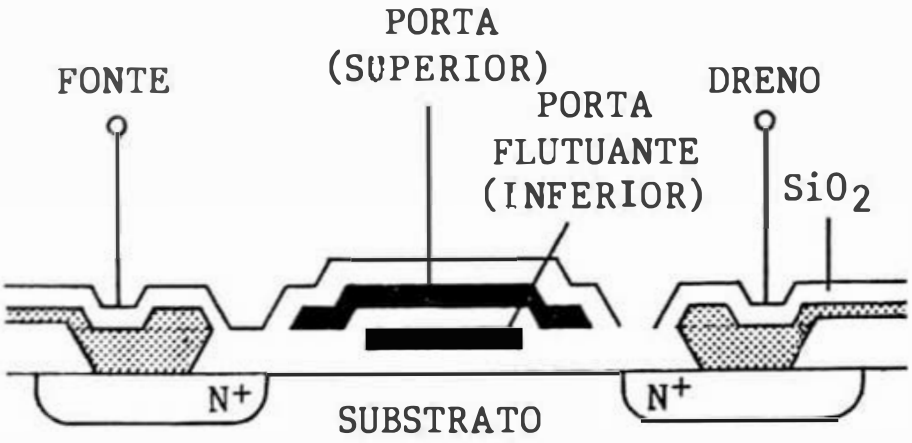


FIGURA 1

Na verdade, cada célula de memória é um transistor de efeito de campo com uma porta flutuante. Ao ser aplicado um nível especial de tensão ao dispositivo, elétrons de alta energia serão injetados para o interior da porta, fazendo com que o transistor entre em operação. Ao remover-se a tensão especial de programação, não haverá nenhuma via de descarga para os elétrons acumulados na porta flutuante, desde que esta, é claro, esteja envolta por um material de baixíssima condutividade (isolante). Uma vez gravada a informação numa determinada célula, aí ela permanece indefinidamente a não ser, que a célula de memória seja exposta à luz ultravioleta, pois este tipo de luz irá drenar a carga e o resultado será o apagamento da informação gravada na célula de memória.

Como regra geral, as EPROMs, quando apagadas (ou virgens) possuem em todas as suas células um nível lógico alto "1".

As memórias PROMs são programadas de forma semelhante às EPROMs. O que muda são os processos internos de gravação da informação. Por outro lado, as memórias EPROMs apresentam uma série de vantagens sobre as PROMs, principalmente o fato das EPROMs serem reprogramáveis. Com base nestes fatos, iremos dar maior ênfase ao estudo dos gravadores de EPROM.

GRAVADORES DE EPROM:

Tendo em vista o fato de que uma memória EPROM, inicialmente, ou mesmo depois de passar pelo processo de apagamento, conter "1s" (ums) em todas as posições de bit, o ato de programar a este tipo de memória irá então constituir-se em introduzir "0s" (zeros) dentro das posições de bits requeridas. Para tanto, foram desenvolvidos alguns circuitos eletrônicos, especiais, os quais recebem o nome de gravadores de EPROM.

O circuito de um gravador de EPROM deve, em primeira instância, ser capacitado para fornecer os níveis de tensão e os parâmetros de tempo requeridos pela memória EPROM. Assim sendo, o projeto de um gravador de EPROM irá depender muito das características da memória EPROM que irá ser gravada por ele. Estas características são fornecidas pelo fabricante.

Atualmente, existem várias memórias EPROMs no mercado, porém as mais consagradas são as 2708, 2716, 2732 e 2764; as quais, além de apresentarem maior facilidade de programação em comparação com as EPROMs 1702, possuem também uma maior densidade de armazenamento.

Com base nestes fatos, estudaremos somente os gravadores de memórias EPROMs 2708 e 2716, mais especificamente.

GRAVADOR DE EPROM 2708:

A EPROM 2708 é uma memória de 8192 bits organizados internamente sob a forma de 1K x 8 bits. Normalmente, esta memória é utilizada em aplicações que requerem rápidas alterações de programas e experimentos de diferentes padrões de gravação. Todas as entradas e saídas de dados são compatíveis com níveis TTL, durante os modos de programação e leitura. As saídas são do tipo três-estados (Tri-state), permanecendo no estado de alta impedância quando não ativadas, o que permite um "interfaceamento" direto com os barramentos do sistema. A figura 2 nos mostra o diagrama de blocos e a pinagem da EPROM 2708.

FIGURA 2

PINAGEM

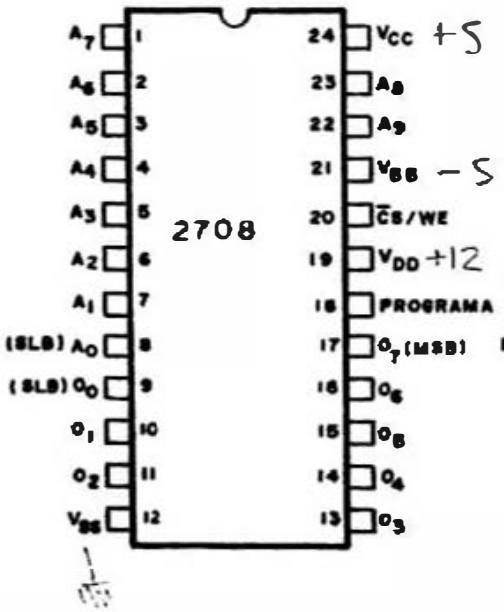
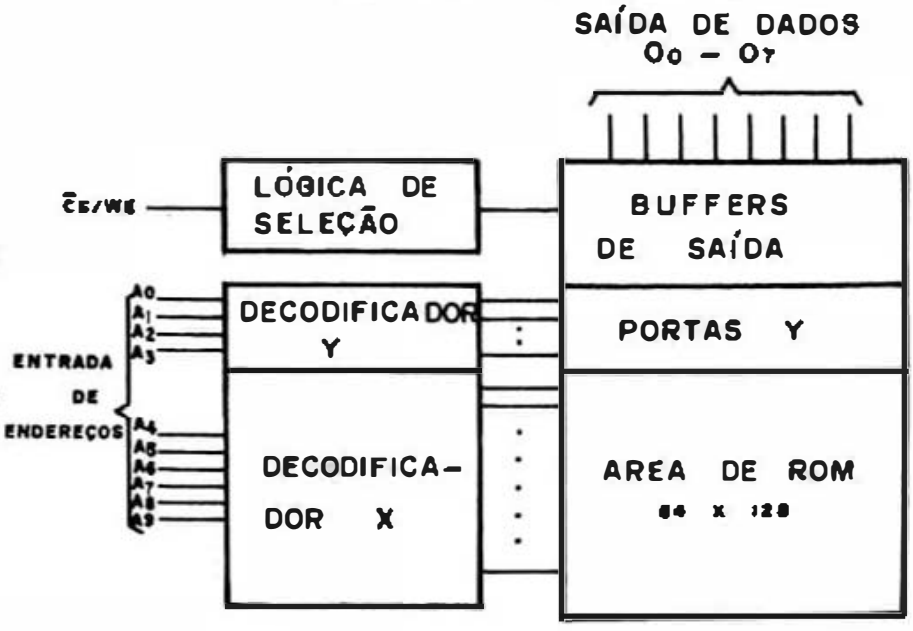


DIAGRAMA DE BLOCO



Fabricada com base na tecnologia FAMOS canal-N, a 2708 tem um tempo de acesso de 450 ns e dissipa uma potência máxima igual a 1,5 W. Para o modo de programação, esta memória requer uma tensão de 25 volts. As tabelas 1 e 2 nos mostram as características de programação DC e AC, respectivamente, enquanto que a figura 3 nos mostra as formas de onda da programação.

CARACTERÍSTICAS D.C. DE PROGRAMAÇÃO

Símbolo	Parâmetros	Min.	Típico	Máx.	Unidade	Condições de teste
I_{LI}	Corrente sink da entrada de endereços e CS/WE			10	µA	Vin=5,25V
I_{IPL}	Corrente da fonte para o pulso de programação			3	mA	
I_{IPH}	Corrente sink para o pulso de programação			20	mA	
I_{DD}	Corrente de alimentação V_{DD} +12 V		50	65	mA	Todas as entradas altas. CS/WE = 5V $T_A = 0^oC$
I_{CC}	Corrente de alimentação V_{CC} +5V		6	10	mA	
I_{BB}	Corrente de alimentação V_{BB} −5V		30	45	mA	
V_{IL}	Nível baixo de entrada (exceto durante a programação)	V_{SS}		0,65	V	
V_{IH}	Nível alto de entrada para endereços e dados	3,0		Vcc+1	V	
V_{IHW}	Nível alto na entrada CS/WE	11,4		12,6	V	Com referência à V_{SS}
V_{IHP}	Nível alto do pulso de programação	25		27	V	Com referência à V_{SS}
V_{ILP}	Nível baixo do pulso de programação	V_{SS}		1	V	$V_{IHP}-V_{ILP}$ = 25Vmin

TABELA 1

CARACTERÍSTICAS A.C. DE PROGRAMAÇÃO

Símbolo	Parâmetros	Min.	Típico	Máx.	Unidade
t_{AS}	Tempo de ajuste (setup) do endereço	10			µs
t_{CSS}	Tempo de ajuste (setup) do $\overline{CS}$/WE	10			µs
t_{DS}	Tempo de ajuste dos dados	10			µs
t_{AH}	Tempo de retenção do endereço	1			µs
t_{CH}	Tempo de retenção de $\overline{CS}$/WE	0,5			µs
t_{DH}	Tempo de retenção de dados	1			µs
t_{DF}	Tempo de atraso entre o "chip de select" e a saída instável	0		120	µs
t_{DPR}	Tempo de atraso entre o programa e a leitura			10	µs
t_{PW}	Largura do pulso de programação	0,1		1,0	ms
t_{PR}	Tempo de subida do pulso de programação	0,5		2,0	ns
t_{PF}	Tempo de descida do pulso de programação	0,5		2,0	µs

TABELA 2

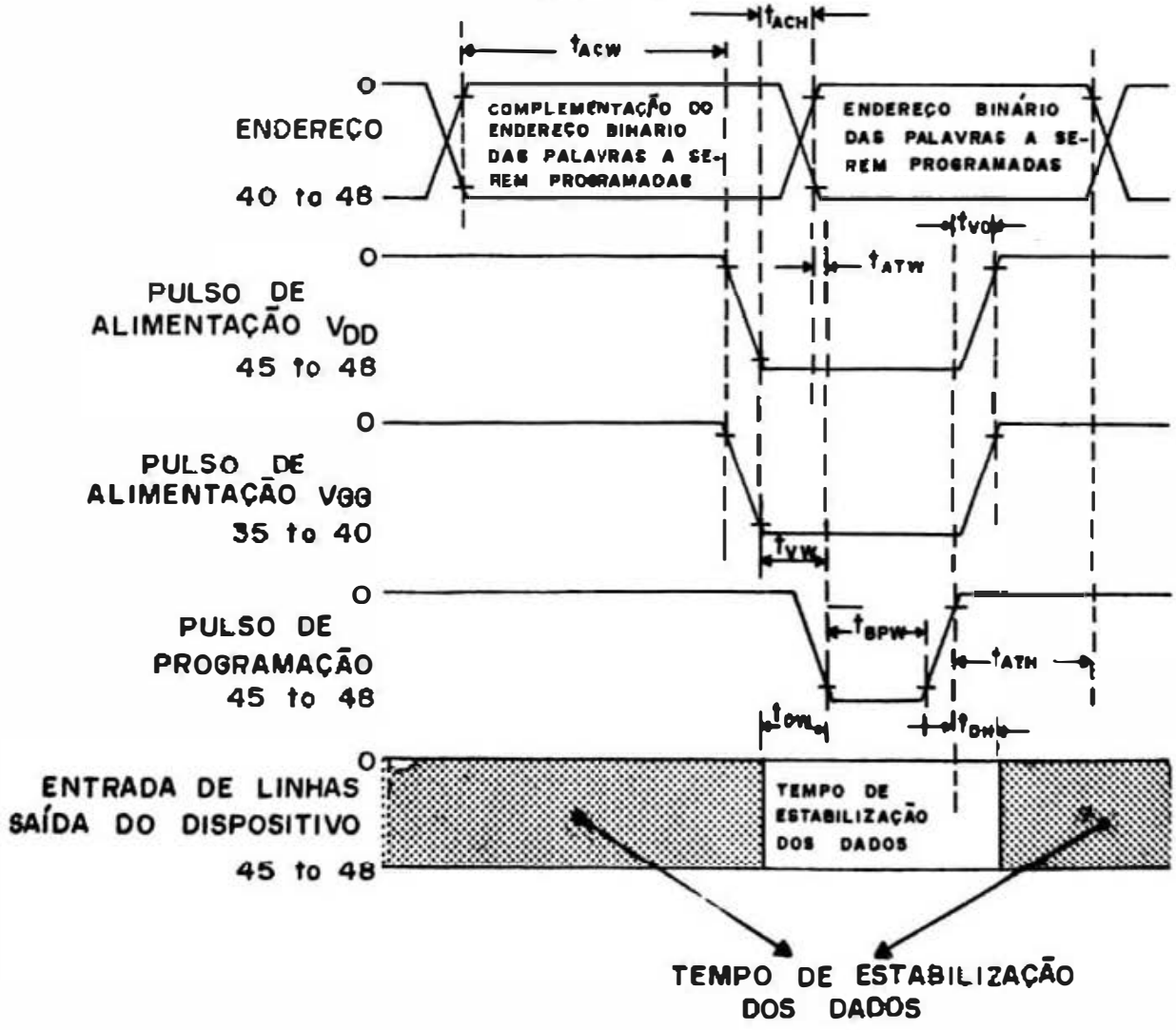


FIGURA 3

Existem duas maneiras de se programar uma memória EPROM: manualmente e automaticamente. Naturalmente, o programador manual é mais lento e requer certa habilidade por parte do usuário. Apenas, a título de exemplo, a figura 4 nos mostra o esquema elétrico de um gravador de memória EPROM2708, Manual.

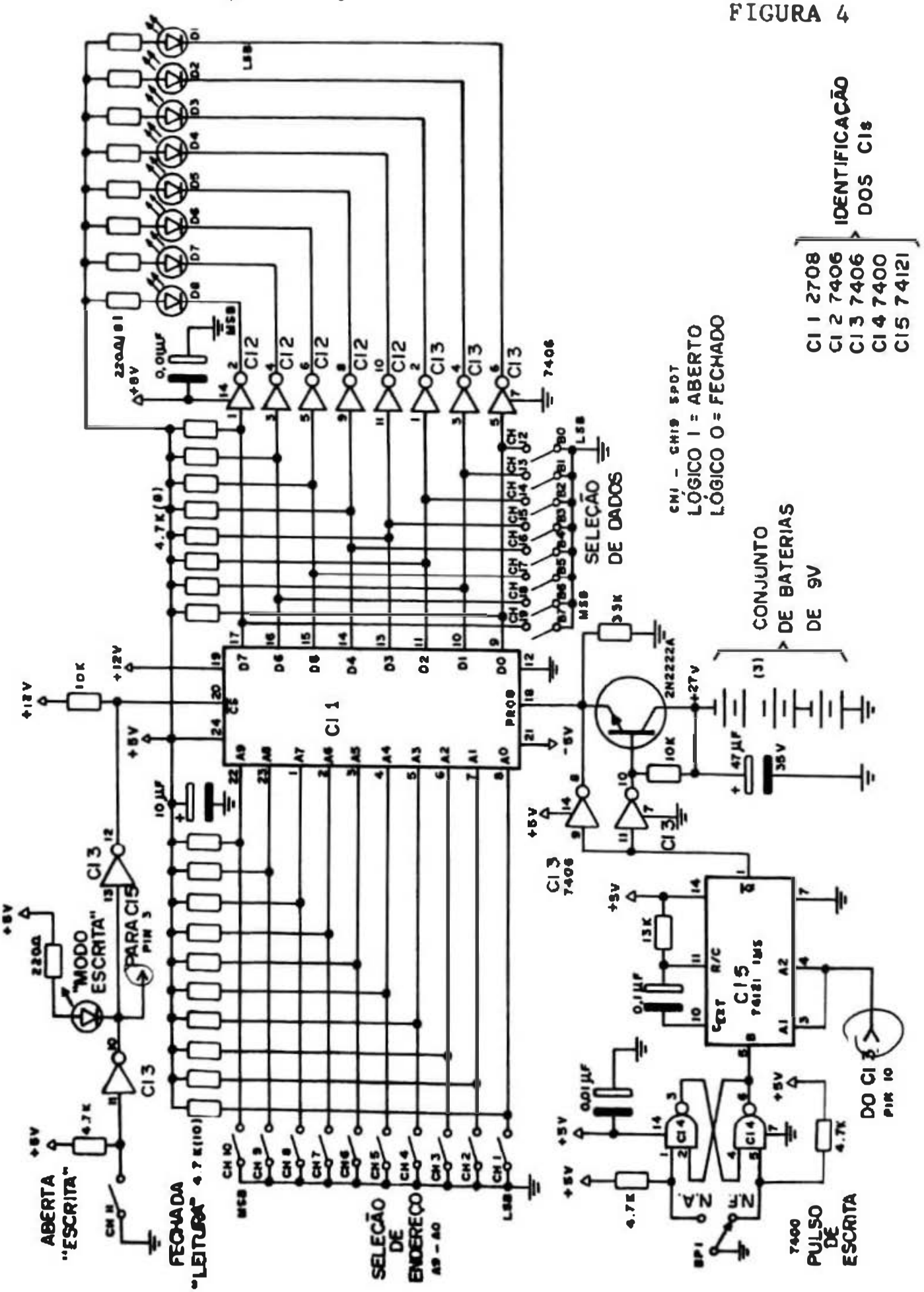
Com relação ao circuito da figura 4, os circuitos integrados de nº 5 e parte do de nº 3 são os responsáveis pelo fornecimento do pulso de +25V para a EPROM. O circuito integrado nº 5 (ou simplesmente 5) SN74121 está armado para uma duração de 1 ms e é disparado na transição de 0 para 1 (flanco positivo) na sua entrada. Através do pino 18 (programa), a EPROM 2708 tanto fornece quanto requer corrente. No "modo escrita", quando CS/WE (pino 20) está em 12V e entre pulsos de programação, o pino 18 fornece uma corrente baixa. A corrente do pulso de programação é de cerca de 30 mA e deve estar entre 25V e 27V.

Para se determinar qual dos 1024 bytes receberá o dado (byte) a ser gravado, um endereço de 10 bits deverá estar presente nas chaves CH_1 a CH_{10}. Para se iniciar o processo de programação, a partir da posição 0 (zero), todas as chaves deverão estar fechadas. Em seguida, os 8 bits (do dado) que serão armazenados, serão colocados nas chaves CH_{12} a CH_{19} e poderão ser mostrados no display D_1 a D_8 (LEDs). Finalmente, para o programador estar no modo de escrita, a chave CH_{11} deve estar na posição aberta. A gravação do byte de dado ocorre no instante em que o botão de pulso de escrita BP1 for pressionado. Este fornece um pulso de 25V para o pino de programação da EPROM, durante um tempo de 1 ms. Para a máxima retenção dos dados, 100 destes pulsos de programação são recomendados, os quais devem ser aplicados seqüencialmente num total de 100 aplicações de 1 ms, o que nos leva a um tempo de 100 ms, no mínimo. Assim sendo, podemos concluir que para uma retenção completa, cada endereço deverá ser reescrito em um programador automático, pois, no caso do gravador manual, esta tarefa torna-se exaustiva por demais.

O processo de leitura do conteúdo ou das informações armazenadas na EPROM 2708 é fácil de ser realizado, bastando, para tal, levar todas as chaves de entrada de dado CH_{12} a CH19 para a posição aberta e a chave CH_{11} (lê/escreve) para a posição fechada (modo de leitura) e teremos, no display de saída, o conteúdo do byte apontado (ou endereçado) pelas chaves de entrada de endereço CH_1 a CH_{10}, o qual permanecerá no display

até que seja colocado um outro endereço nas chaves de entrada de endereço (CH_1 a CH_{10}).

FIGURA 4



A figura 5 nos mostra o esquema elétrico de um gravador de memória EPROM automático.

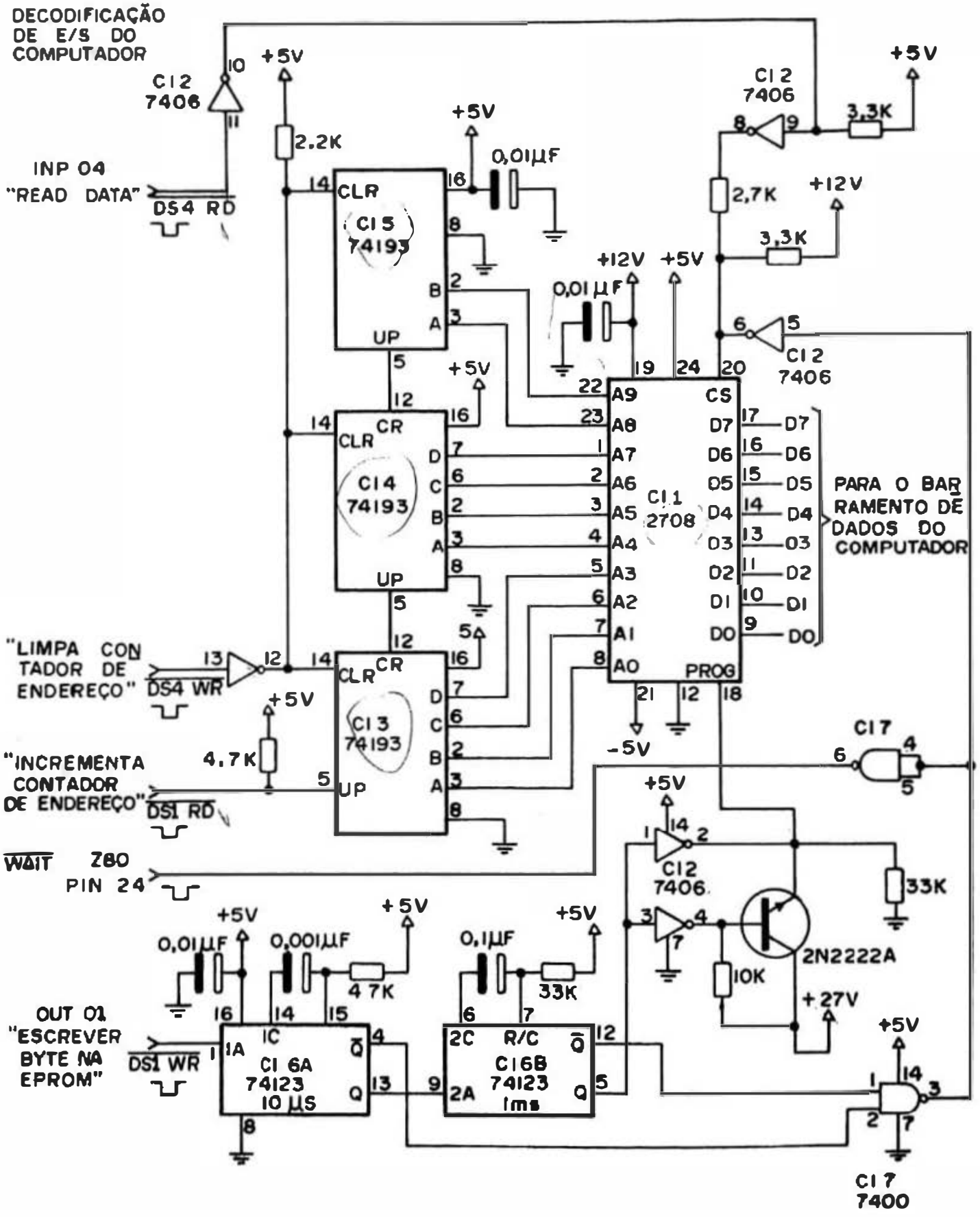


FIGURA 5

O modo de como ocorre o processo de gravação da memória EPROM 2708 é semelhante ao processo de gravação, através do gravador manual. O fluxograma mostrado na figura 6 resume o funcionamento do gravador automático, quando este estiver realizando um ciclo de escrita.

O out 1 (escrita) libera o pulso de programação de 25 V por 1 ms, enquanto o dado a ser gravado encontra-se no barramento de dados, após o que, um INP 01 (strobe decodificado da instrução INP 01) é executado, levando o contador de endereços para a próxima posição de endereço.

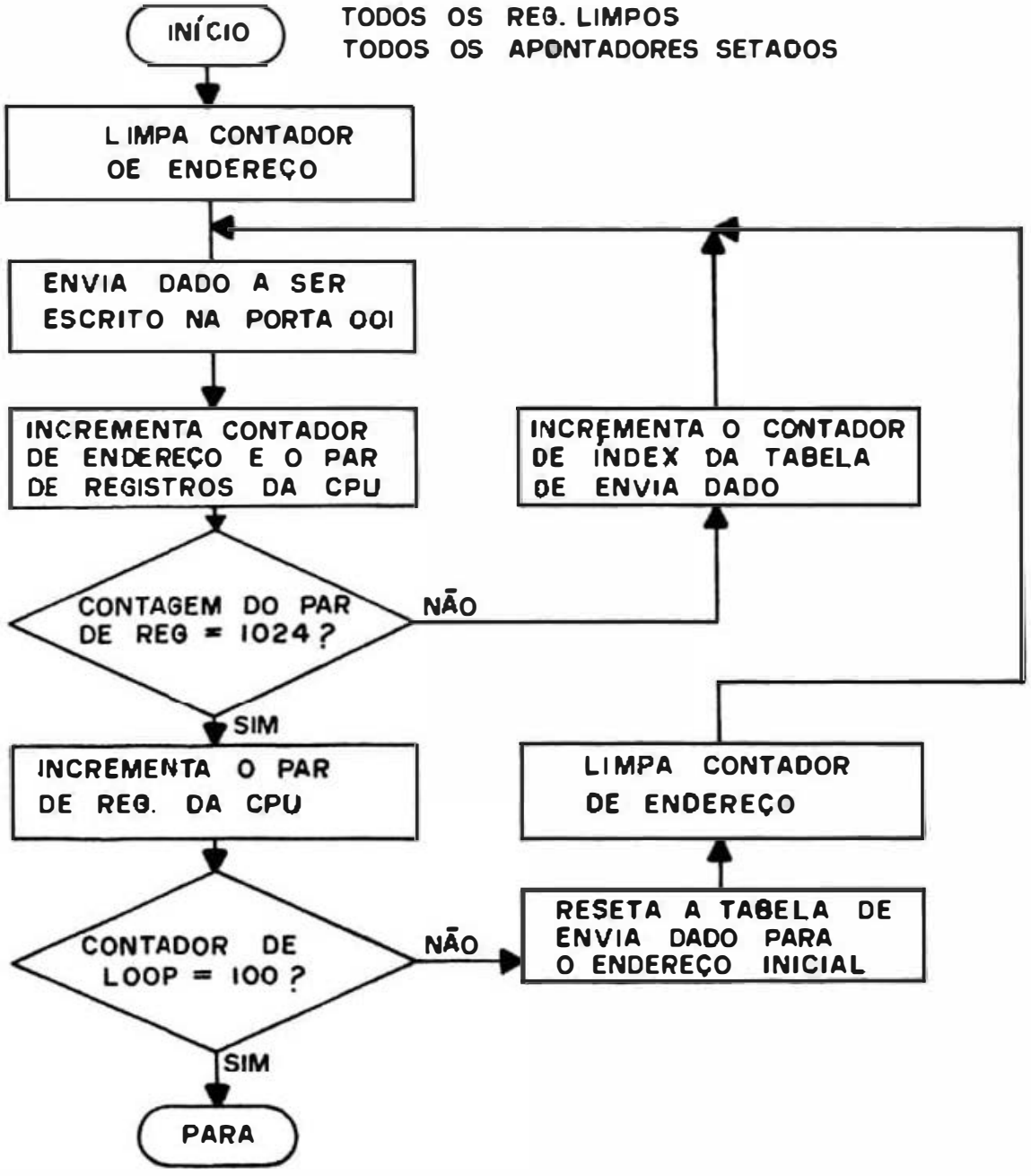


FIGURA 6

O hardware mantém o curso de endereço automaticamente, porém o software deve implementar seus próprios contadores para manter o curso das posições 0 a 1023, tanto quanto o número de vezes que as 1024 posições de memória (bytes) têm de ser programadas.

O processo de leitura também é muito simples. O contador de endereços é novamente limpo, através da execução de uma instrução INP 04. Este dado pode ser armazenado e analisado. Finalmente, o contador de endereços é incrementado, novamente, com uma instrução INP 01 e o processo é repetido para ler o próximo byte. O fluxograma de um ciclo de leitura é mostrado na figura 7. As instruções referentes ao software do circuito apresentado na figura 5 não foram apresentadas, pois os esquemas aqui colocados são apenas a nível de exemplos.

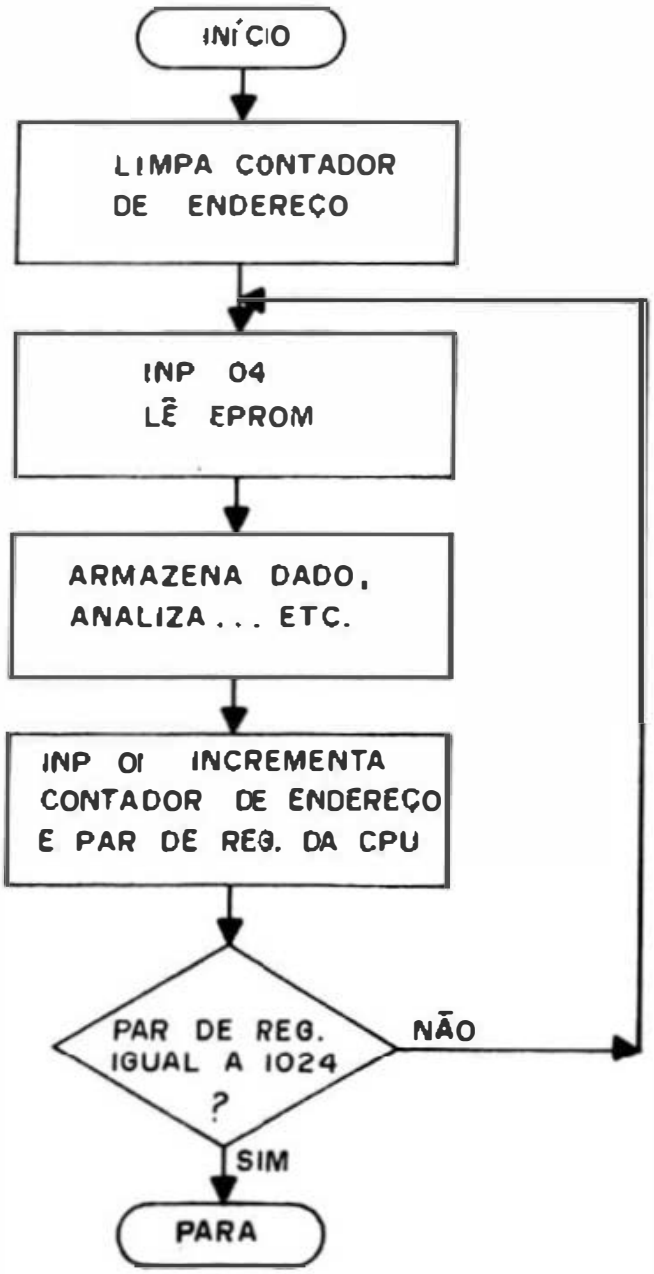


FIGURA 7

Os circuitos para gravação de EPROMs, apresentados até o momento, são para a 2708, porém podem ser facilmente modificados para outras memórias. Mudando-se, por exemplo, a temporização de 1 ms para 50 ms e modificando-se as ligações entre alguns pinos, conseguiremos um gravador de EPROM 2716 ou 2758, ambas de fabricação Intel.

Com relação ao processo de apagamento da EPROM 2708, podemos dizer, resumidamente, que esta pode ser apagada através da exposição à luz ultravioleta de onda curta de alta intensidade, com um comprimento de onda de 2537 Å. Normalmente, recomenda-se uma dosagem de 12,5 W.s/cm² (watt-segundos por centímetro quadrado).

Por outro lado, o tempo de exposição (Te) de qualquer EPROM é determinado pela relação entre a densidade requerida de apagamento pelo componente (J) e a densidade de potência incidente do apagador (I), ou seja:

$$Te = \frac{J}{I}$$

Por exemplo, para apagar o conteúdo de uma EPROM 2708 (J = 12,5 W.s/cm²) em um apagador, cuja densidade de potência incidente é de 10.000 μW/cm² (I), o tempo de exposição necessário será:

$$Te = \frac{12,5}{10.000 \times 10^{-6}} = 1250 \text{ segundos}$$

ou Te = 20 min. e 50 seg.

A densidade de potência incidente do apagador (I) é uma característica intrínseca do apagador em si, ou seja, cada tipo de apagador poderá ter um valor de I diferente.

O processo de determinação do tempo de exposição, necessário para o apagamento de uma memória EPROM, que acabamos de expor, é válido para todos os tipos de EPROMs existentes no mercado.

Na figura 8, apresentamos um típico apagador de EPROM, o qual pode ser utilizado para apagar memórias EPROMs do tipo 2708, 2716, 2758 e outras.

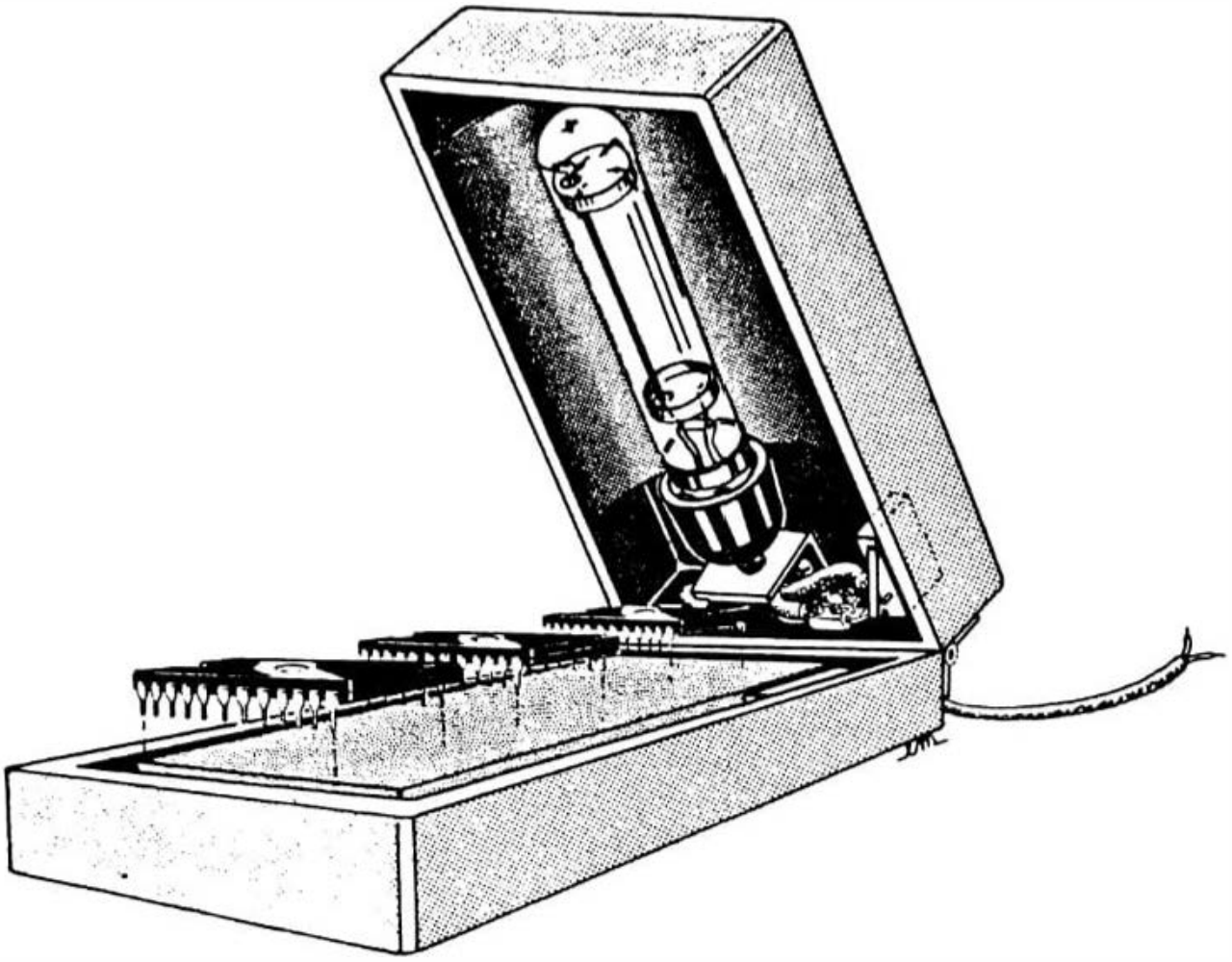


FIGURA 8

GRAVADOR DE EPROM 2716:

A memória EPROM 2716 possui uma capacidade de armazenamento de 16 K bits organizados internamente sob a forma de 2 K x 8 bits. Todas as entradas e saídas de dados são compatíveis com níveis TTL, durante a operação de programação e leitura. As saídas são do tipo três-estados (Tri-state), permanecendo no estado de alta impedância, quando não ativadas. O tempo de acesso máximo é de 450 ns, enquanto que o tipo de operação é completamente estático.

A EPROM 2716 é uma das memórias mais usadas atualmente, devido a rapidez e a facilidade de programação, proporcionando, assim, maior economia no desenvolvimento de projetos. A figura 9 nos mostra o diagrama de blocos e a pinagem da EPROM 2716.

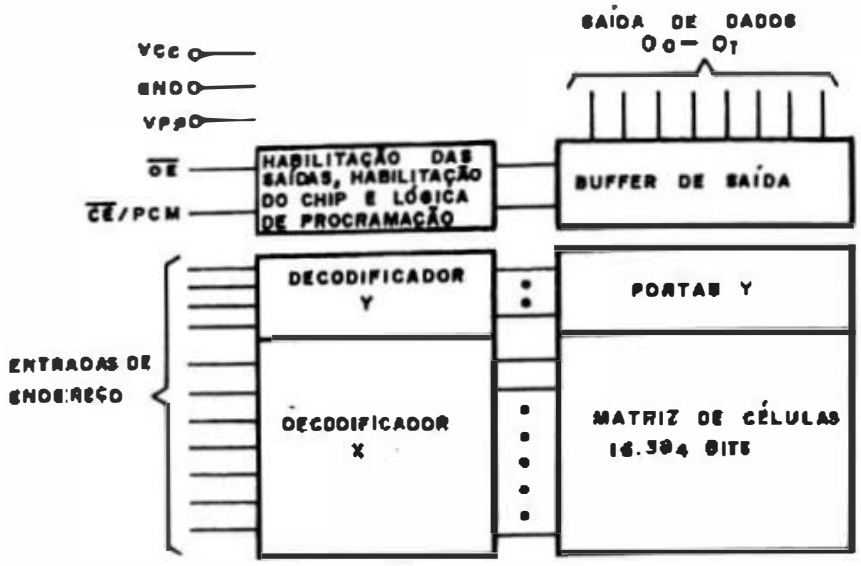
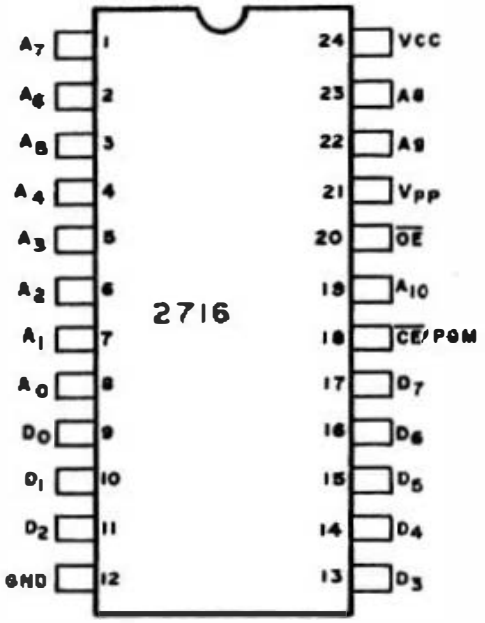


DIAGRAMA DE BLOCOS

FIGURA 9



PINAGEM

A memória EPROM 2716 é a primeira memória desta família que possui um modo de espera estática (standby), reduzindo, assim, a dissipação de potência sem incorrer num provável aumento do tempo de acesso. Além disto, esta memória EPROM possui o mais simples e rápido método de programação de EPROMs, uma vez que dispensa pulsos de alta voltagem, pois todos os controles de programação são executados por sinais TTL. Qualquer posição de memória pode ser programada em qualquer instante, seja seqüencialmente, aleatoriamente ou isoladamente. O tempo total de programação para os 16.384 bits é de 1 minuto e 40 segundos, ou seja, 100 segundos.

As tabelas 3 e 4, a seguir, nos mostram as características de programação DC e AC, respectivamente. Cabe ressaltar o fato de que a 2716 necessita de uma única fonte de tensão de +5 V.

CARACTERÍSTICAS D.C. DE PROGRAMAÇÃO

Símbolo	Parâmetros	Condições de teste	Min.	Máx.	Unidade
I_{LI}	Corrente de entrada	Vin = 5,25/0,45V		10	µA
I_{PP1}	Corrente de alimentação V_{PP}	$\overline{CE}$/PGM=V_{IL}		5	mA
I_{PP2}	Corrente de alimentação V_{PP} durante pulso de programação	$\overline{CE}$/PGM=V_{IH}		30	mA
I_{CC}	Corrente de alimentação V_{CC}			100	mA
V_{IL}	Nível baixo de entrada		-0,1	0,8	volts
V_{IH}	Nível alto de entrada				

TABELA 3

CARACTERÍSTICAS A.C. DE PROGRAMAÇÃO

Símbolo	Parâmetros	Condições de teste	Min.	Máx.	Unidade
t_{AS}	Tempo de ajuste do endereço	Entrada T_R e T_F (10% para 90%)= 20 ns Níveis dos sinais de entrada = 0,8 para 2,2 V Nível de referência da entrada de sincronismo = 1V e 2 V Nível de referência da saída de sincronismo=0,8V e 2 V	2		µs
t_{OES}	Tempo de ajuste da habilitação da saída		2		µs
t_{DS}	Tempo de ajuste dos dados		2		µs
t_{AH}	Tempo de retenção do endereço		2		µs
t_{OEH}	Tempo de retenção da habilitação da saída		2		µs
t_{DH}	Tempo de retenção dos dados		2		µs
t_{DF}	Tempo de atraso entre a inabilitação da saída e a saída ($\overline{CE}$/PGM = V_{IL})		0	120	ns
t_{OE}	Tempo de atraso entre a habilitação da saída e a saída ($\overline{CE}$/PGM = V_{IL})		-	120	ns
t_{PW}	Largura do pulso de programação		45	55	ms
t_{PRT}	Tempo de subida do pulso de programação		5	-	ns
t_{PFT}	Tempo de descida do pulso de programação		5	-	ns

TABELA 4

NOTAS:

- V_{CC} deve ser aplicado simultaneamente ou antes de V_{PP} e removido simultaneamente ou depois de V_{PP}.
- V_{PP} não deve ser maior do que 26 V. O dispositivo pode ser danificado se este for retirado ou colocado no soquete, quando V_{PP} = 25 volts estiver sendo aplicado. Também, durante $\overline{OE}$ = $\overline{CE}$/PGM = V_{IH}, V_{PP} não deve ser elevado de 5 volts para 25 volts e vice-versa.

Por sua vez, a figura 10 nos mostra as formas de onda da programação.

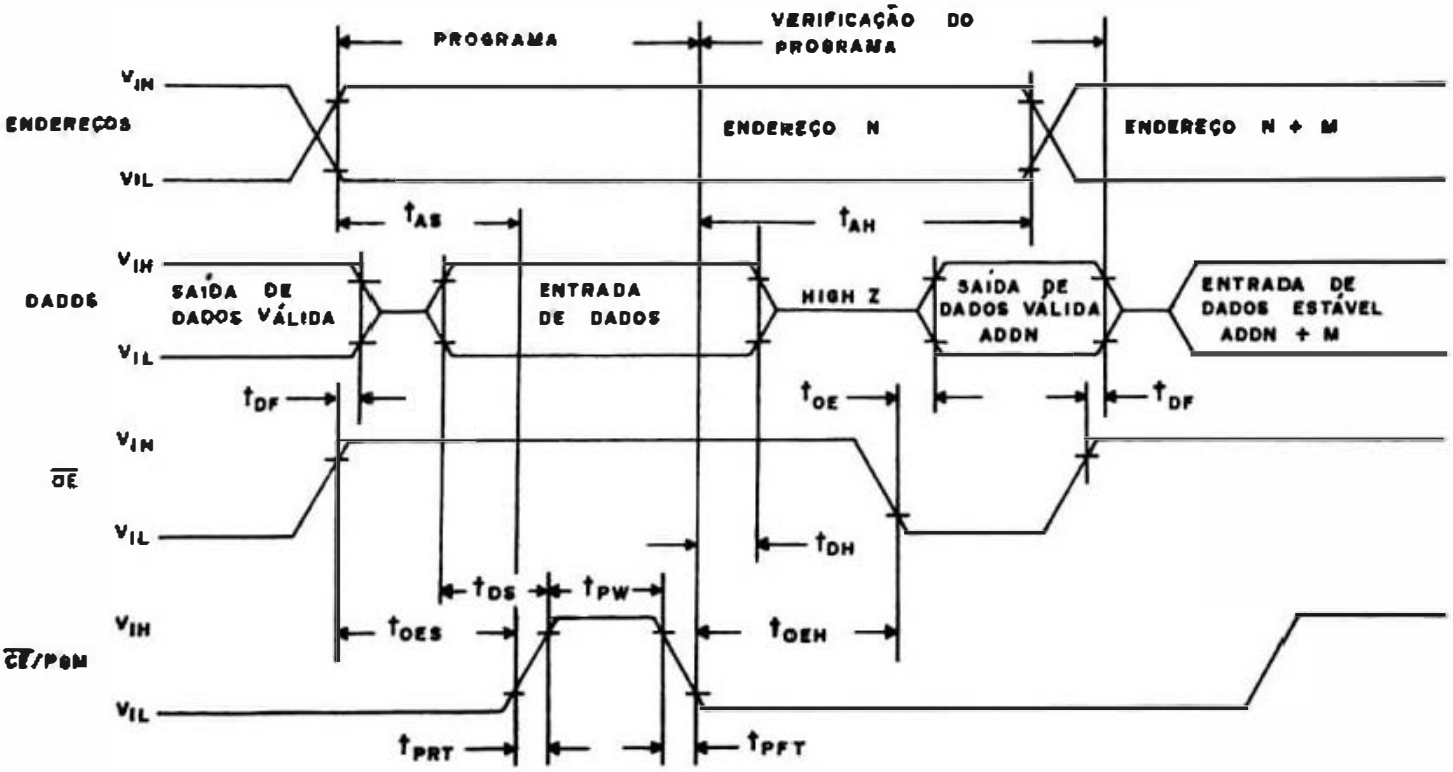


FIGURA 10

A operação de programação da EPROM 2716 inicia-se por elevar a entrada V_{PP} (pino 21) de +5V para +25V. O dispositivo é então desselecionado, levando a saída $\overline{OE}$ para um nível lógico alto "1". Um único pulso TTL de 50 ms é aplicado na entrada $\overline{CE}$/PGM, durante a programação de uma célula ou uma posição de memória.

Os passos requeridos para se programar uma simples célula ou posição de memória da EPROM 2716 são mostrados, a seguir:

a) Inicialmente, coloque o endereço da localização a ser programado sobre as linhas de endereço (A_0 até A_{10});

b) Coloque os 8 bits (ou byte), a serem escritos na EPROM, no barramento de dados (O_0 até O_7);

c) Uma vez elevada a entrada Vpp de +5 V para +25 V, faça com que a entrada $\overline{CE}$ saia do nível lógico baixo e vá para um nível lógico alto, retornando, em seguida, para o estado inicial (nível lógico baixo), gastando para tanto não mais do que 50 ms.

Estes passos ou etapas são repetidos tantas vezes quantas localizações forem requeridas. No caso da EPROM 2716, temos, no máximo, 2048 localizações (ou posições de memória). Depois que todas as localizações forem programadas normalmente, procede-se uma rotina de verificação, cujos passos são mostrados em seguida:

a) Inicialmente, coloque o endereço da localização a ser verificada nas linhas de endereço (A_{10} até A_0);

b) Leve a saída $\overline{OE}$ para o nível lógico baixo;

c) Em seguida, leve a entrada $\overline{CE}$ também para o nível lógico baixo;

d) Leia os dados da localização endereçada no barramento de dados (O_7 - O_0).

Um gravador de EPROM compatível com as características da memória 2716 deve implementar todas as etapas de escrever (programação) e ler dados na referida memória. A figura 11 nos mostra o circuito lógico de um programador de EPROM para a EPROM 2716.

As chaves CH_{1A} até CH_{10} são utilizadas para selecionarmos o endereço a ser programado. As chaves CH_{11} até CH_{18} são utilizadas para a entrada de dados para dentro da EPROM, durante a programação. Todas as chaves de CH_{1A} até CH_{18} são do tipo spdt (pólo simples, arremesso duplo), pois podem ligar os pinos a que estão conectadas tanto no barramento de terra (0V) como no barramento +Vcc (+5V).

A chave CH_{19} é uma chave de três pólos de duplo arremesso que desliga o potencial de terra e +Vcc das chaves que estão conectadas no barramento de dados, durante a programação e muda a entrada $\overline{OE}$ de +Vcc (programa) para terra (verificação).

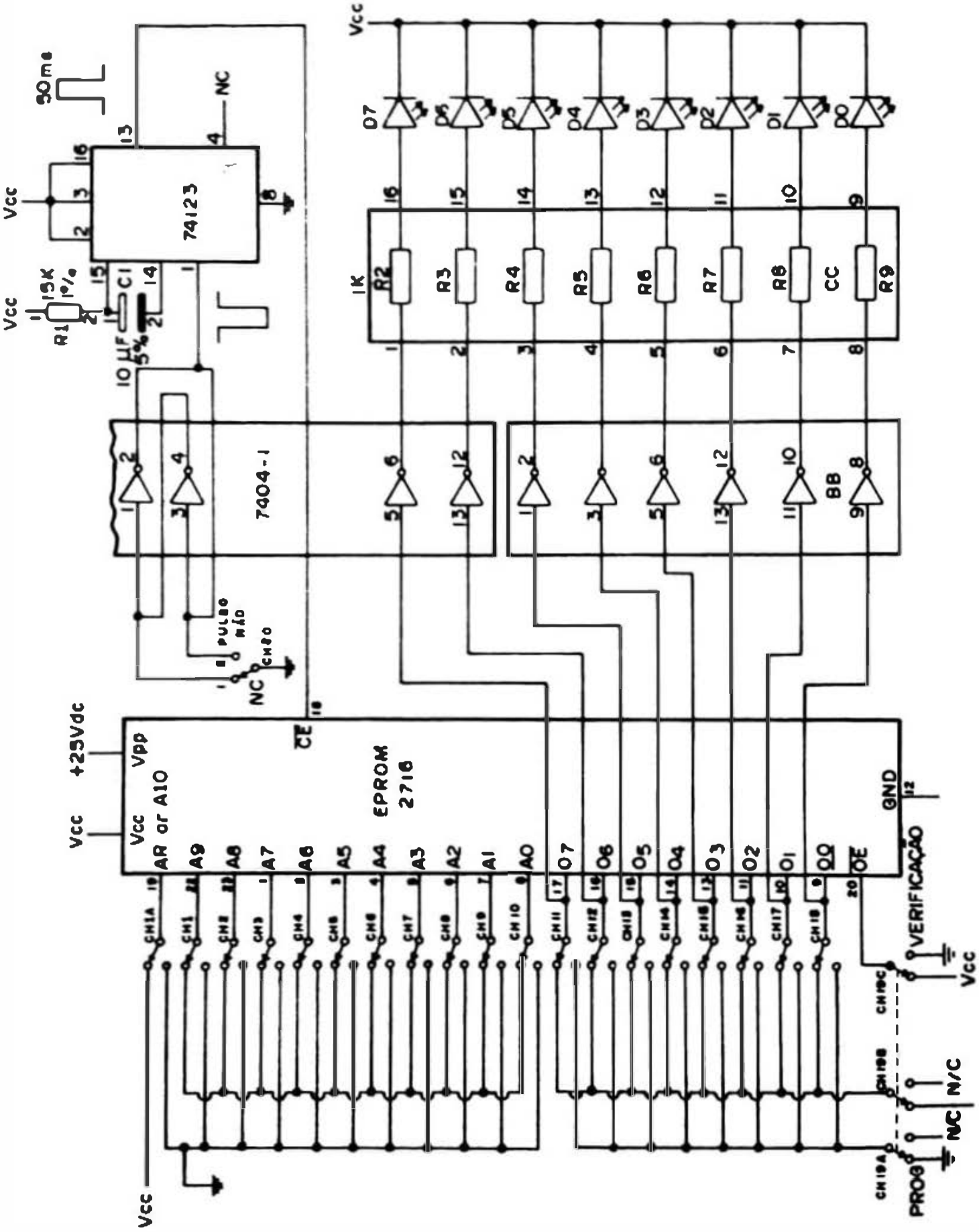

50ms
NC
74123
Vcc
15K 1%
R1
C1
10 µF 5%
7404-1
PULSO MIO
CHIO
NC
CE
+25Vdc
Vpp
Vcc
AR or A10
EPROM 2716
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
O7
O6
O5
O4
O3
O2
O1
O0
OE
GND
VERIFICAÇÃO
PROG
N/C
D7
D6
D5
D4
D3
D2
D1
D0
1K
R2
R3
R4
R5
R6
R7
R8
CC
R9
BB

FIGURA 11

Os dados que devem ser escritos (gravados) ou lidos na EPROM 2716 são mostrados em oito (8) LEDs (D_0 a D_7). Os resistores R_2 a R9 operam como limitadores de corrente para os LEDs. As portas inversoras, por sua vez, invertem a polaridade dos sinais vindos das saídas da EPROM ou das chaves de dados da entrada, de modo a torná-los compatíveis com a lógica aplicada no projeto do circuito; além disto, esses inversores operam também como drivers, amplificando o nível de força dos sinais, de modo que estes possam excitar os respectivos LEDs.

Os dois inversores conectados à chave CH_{20} fornecem, na saída, um pulso simples, durante o período de salto do interruptor, toda vez que este for acionado. Este pulso simples ( ) ativa o multivibrador monoestável SN74123, o qual produz, em sua saída, um pulso, cuja duração é de aproximadamente 50 ms, o qual é aplicado na entrada CE da EPROM 2716, completando, assim, o ciclo de programação.

A figura 12 nos mostra uma possível configuração física do programador aqui descrito. Muito embora o processo de gravação deste gravador de EPROM seja relativamente moroso, este gravador é simples de ser operado, fácil de ser construído e de baixo custo.

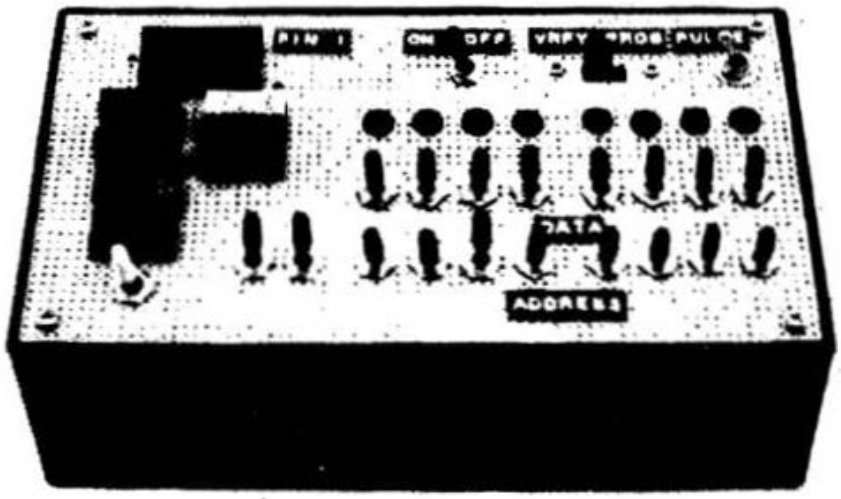


FIGURA 12

APAGANDO AS EPROMs:

Genericamente, o processo de se apagar o conteúdo das memórias EPROMs resume-se, pura e simplesmente, no ato de expor

a EPROM a uma fonte de luz ultravioleta de onda curta de alta intensidade, com um determinado comprimento de onda. É claro que se deve respeitar as especificações impostas pelo fabricante com relação à radiação ultravioleta, tais como: comprimento de onda, densidade de apagamento, tempo de exposição, distância da fonte de luz, etc. A não obediência dessas especificações irá influir sobre a qualidade do apagamento.

Muito embora o processo de apagamento seja simples, alguns cuidados devem ser tomados por parte do operador que está procedendo o apagamento da EPROM, pois a luz ultravioleta pode causar danos irreparáveis ao olho humano. Como prevenção, o usuário nunca deve olhar diretamente para uma fonte de luz ultravioleta, pois esta poderá causar queimaduras no olho. Recomenda-se, portanto, muita prudência.

Após o apagamento do conteúdo da EPROM, é conveniente verificar alguns endereços de memória, antes de reprogramá-la, para certificar-se se todos os valores de dados, agora contidos na EPROM, são uns (1s).

Ao proceder o apagamento, coloque o chip o mais próximo possível da fonte de luz ultravioleta e, é óbvio, com a janela translúcida descoberta.

Depois de programada (ou gravada) a EPROM, a janela não precisa ser necessariamente recoberta, pois mesmo que a EPROM fique exposta à luz do sol, seriam necessárias várias semanas para apagar o seu conteúdo, porém o uso de uma fita ou mesmo uma etiqueta não é uma má idéia.

COMENTÁRIOS FINAIS:

Toda vez que se pretender operar um gravador de EPROM ou um apagador de EPROM, devemos, inicialmente, verificar as instruções contidas no manual do fabricante, referentes a estes aparelhos.

O curso CEDM projetou e desenvolveu um gravador de EPROM e um apagador de EPROM, sendo que a comercialização dos mesmos deverá ser iniciada em breve. Caso o aluno se interesse por alguns destes aparelhos, poderá nos escrever solicitando maiores informações a respeito dos mesmos.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-25

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) A memória EPROM, quando apagada, contém em todas as posições de bit o binário:

a) 10;

b) 1;

c) 0;

d) N.R.A.

2) O ato de programar a EPROM consiste, pura e simplesmente, no fato de se introduzir:

a) zeros (0s) nas posições de bits requeridas;

b) uns (1s) nas posições de bits requeridas;

c) os itens a e b se complementam;

d) N.R.A.

3) Uma EPROM pode ser programada pelo processo:

a) manual;

b) automático;

c) os itens a e b estão corretos;

d) N.R.A.

4) Com relação ao circuito mostrado na figura 4, para se iniciar o processo de programação, a partir da posição zero (0):

a) todas as chaves de S_1 a S_{10} devem estar fechadas;

b) todas as chaves de S_1 a S_{10} devem estar abertas;

c) apenas a chave S_{10} deve estar aberta;

d) N.R.A.

5) Ainda com relação ao circuito apresentado na figura 4, para se ler o conteúdo de uma EPROM 2708:

a) as chaves CH_{12} a CH_{19} devem estar abertas;

b) as chaves CH_{12} a CH_{19} devem estar fechadas;

c) a chave CH_{11} deve estar fechada;

d) os itens a e c se completam e estão corretos.

6) Sendo a potência incidente igual a 10.000 $\mu W/cm^2$ e a densidade de apagamento requerida por uma determinada memória EPROM igual a 15 $W.s/cm^2$, podemos afirmar que o tempo de exposição necessário para apagar esta EPROM é igual a:

a) 3000 segundos;

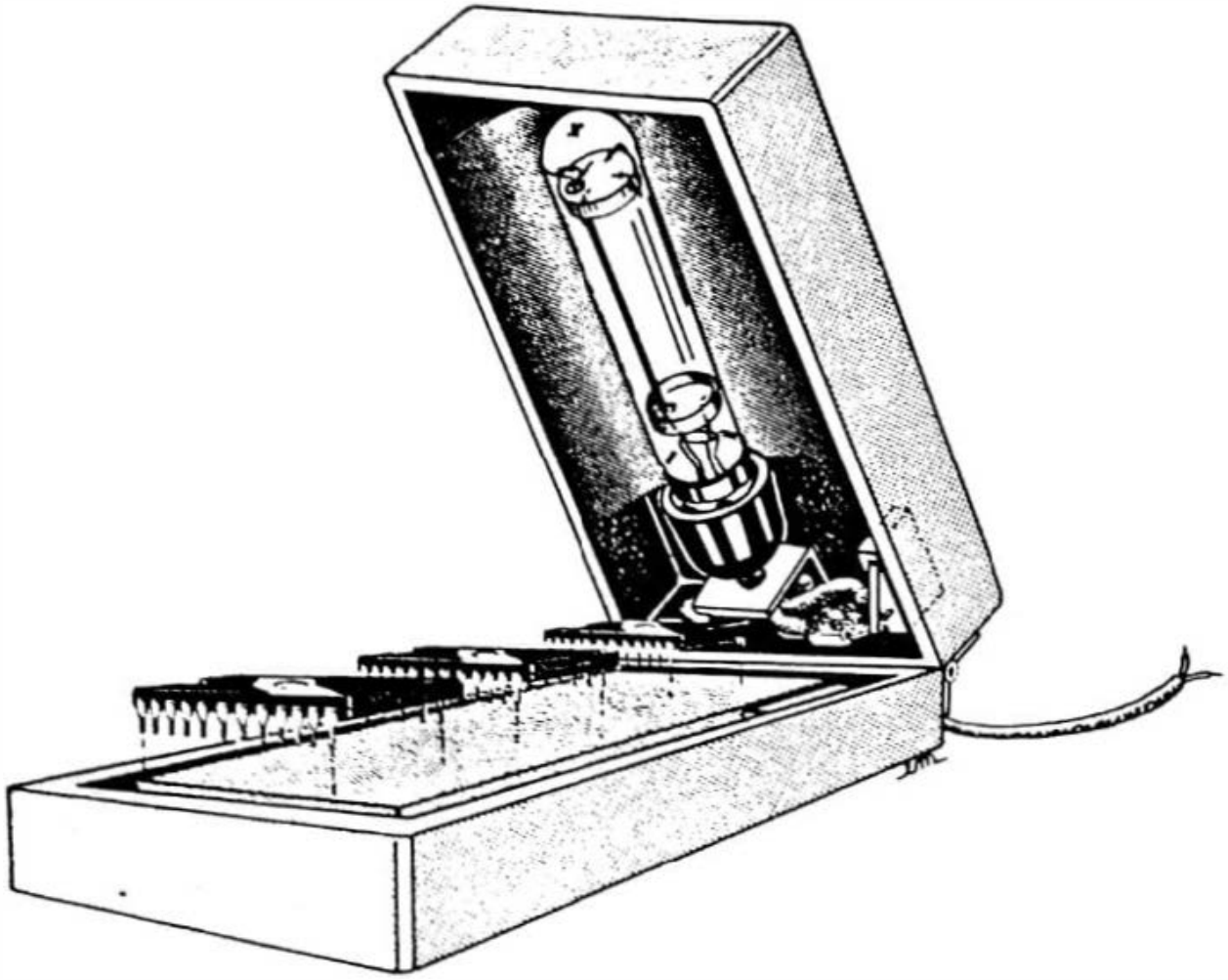
b) 1500 segundos;

c) 1250 segundos;

d) N.R.A.

$$\frac{15}{1000 \times 10^{-6}} = 1500\,s$$

7) Podemos afirmar que a figura apresentada, a seguir, é um:



a) gravador de EPROM automático;

b) gravador de EPROM manual;

c) apagador de EPROM;

d) N.R.A.

8) A operação de programação da EPROM 2716 inicia-se ao elevar a entrada V_{PP} (pino 21) de + 5 V para:

a) + 25 V;

b) + 12 V;

c) + 10 V;

d) N.R.A.

9) Com relação ao circuito apresentado na figura 11, as chaves CH_{1A} até CH_{10} são utilizadas para:

a) a entrada de dados;

b) selecionarmos o endereço a ser programado;

c) mudar o potencial aplicado na entrada $\overline{OE}$;

d) N.R.A.

10) Ainda com relação ao circuito apresentado na figura 11, podemos afirmar que os LEDs (D_0 a D7) são utilizados para:

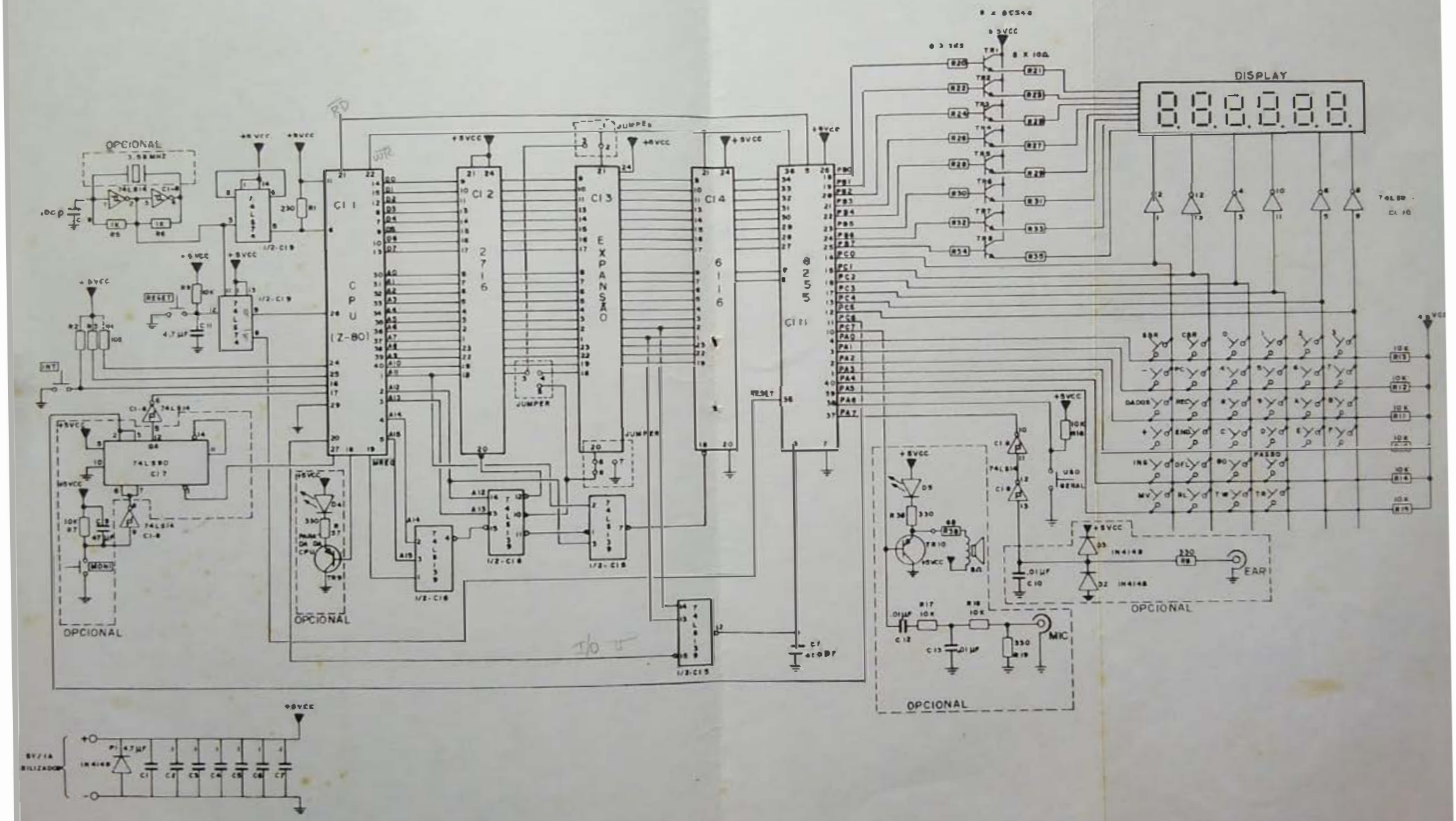
a) tão somente mostrar os dados que devem ser escritos;

b) tão somente mostrar os dados que devem ser lidos;

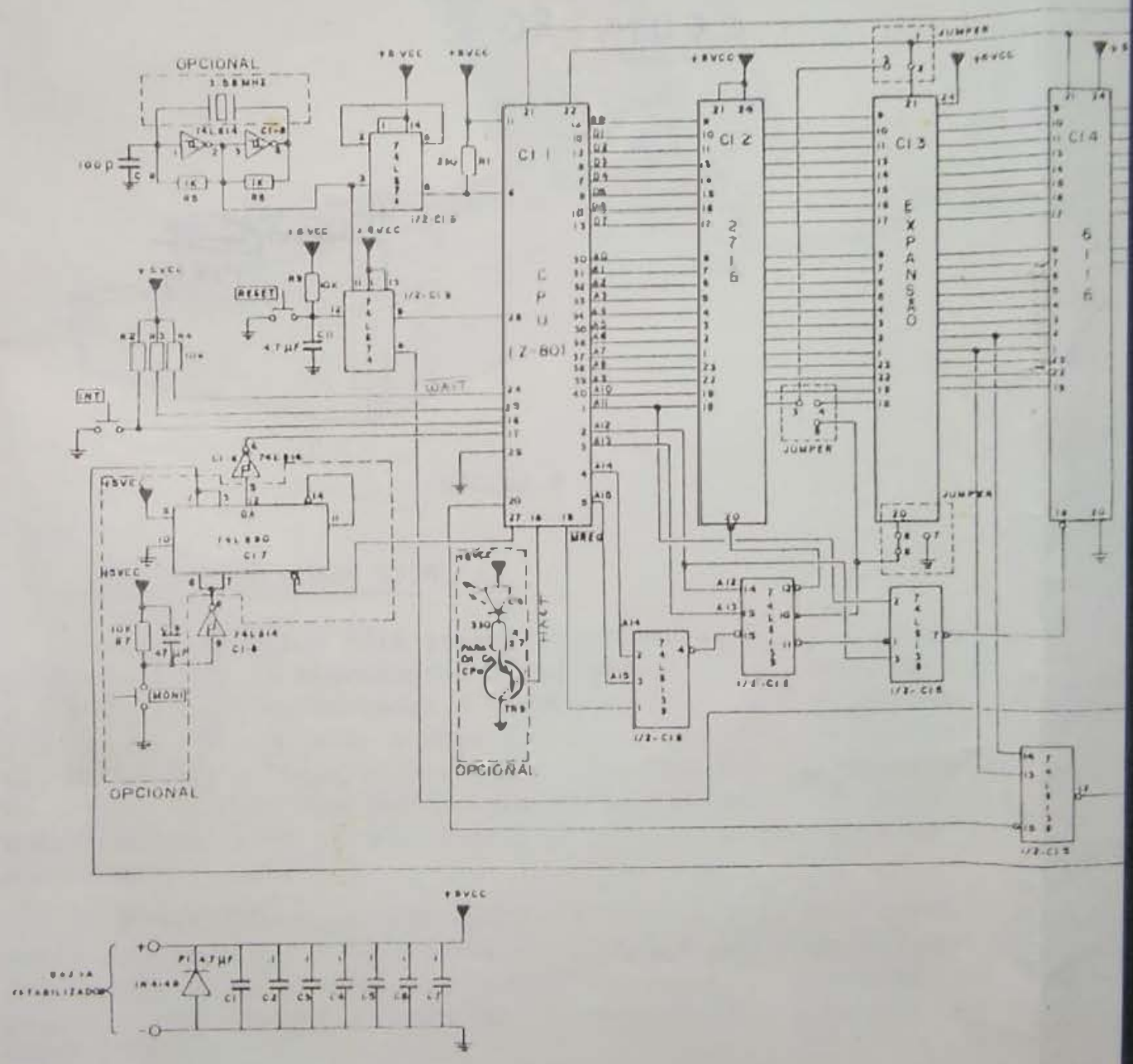
c) apenas para indicar os endereços de memória, no qual se pretende gravar um determinado dado;

d) mostrar os dados que devem ser escritos ou lidos na EPROM 2716.

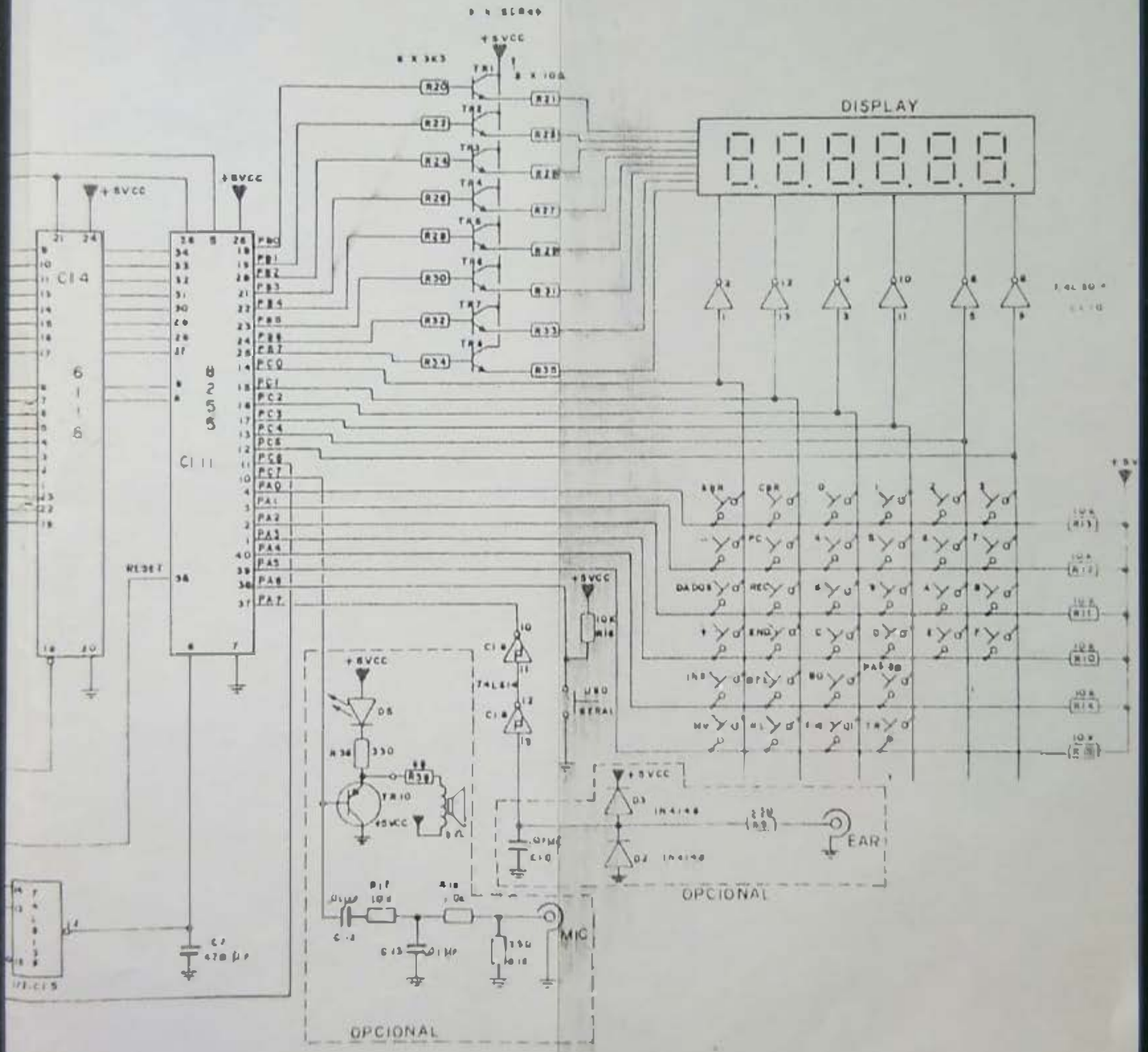
MICROCOMPUTADOR CEDM-80



MICROCOMPUTADO



TADOR CEDM-80



CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA-PARANÁ

LIÇÃO MC-26

PROGRAMAÇÃO ASSEMBLER I

INTRODUÇÃO

Nesta lição, pretendemos mostrar alguns conceitos e definições básicas, utilizados em programação assembler e, em seguida, passaremos ao estudo das instruções do microprocessador Z-80, para que, no final do curso, mais precisamente nas últimas lições da série MP, o aluno tenha condições de montar programas em assembler e visualizar o que acontece no interior do micro CEDM-80, durante a execução do referido programa.

INSTRUÇÕES

Todo microprocessador possui um conjunto de instruções, o qual é constituído de um conjunto de números binários, que produzem ações definidas, durante um ciclo de execução de uma instrução. Basicamente, uma instrução consiste de um número binário, que deve estar disponível no microprocessador, na ordem correta e no tempo apropriado, de modo a executar uma determinada ação desejada. Por exemplo, quando o microcomputador Z-80 recebe o número binário 1000 0000, de 8 bits, na entrada, durante um ciclo de busca de instrução na memória, este número será interpretado pelo microprocessador que, em se-

guida, executará a seguinte ação: "Adiciona (some) o conteúdo do registrador B (ou registro B) ao conteúdo do acumulador".

Toda vez que o microprocessador receber a instrução 1000 0000 na entrada, durante um ciclo de busca de instrução na memória, ele executará a ação acima.

PROGRAMA

Um programa pode ser definido como sendo um conjunto de instruções capaz de fazer com que um computador execute uma determinada tarefa.

Na realidade, um programa de computador é mais do que um simples conjunto de instruções, uma vez que ele deve conter os dados e os endereços de memória, que o microprocessador necessita para poder realizar a tarefa determinada pelas instruções, que compõem o programa. Além disto, um programa deve determinar as fontes ou origens dos dados e o destino do resultado, bem como especificar a operação a ser executada.

Todos os microprocessadores executam instruções em seqüência, ou seja, uma após a outra; na seqüência em que se encontram na memória.

Para o computador, todo programa é "visto" como sendo um conjunto de números binários. Por exemplo, um programa para o microprocessador Z-80, que adiciona os conteúdos das localizações de memória 60H e 61H, colocando o resultado na localização de memória 62H, seria visto pelo computador, conforme nos mostra a figura 1.

```
00111010
01100000
00000000
01000111
00111010
01100001
00000000
10000000
00110010
01100010
00000000
```

FIGURA 1

Este é um programa escrito em linguagem de máquina, ou seja, na "língua" que o microprocessador "entende". Um programa escrito em linguagem de máquina também é conhecido como programa objeto.

Para o ser humano, não é muito fácil "conversar" ou "comunicar-se" com o computador em termos de Os e 1s, para tanto, imagine uma página cheia de códigos Os e 1s. Há, portanto, muitas dificuldades associadas à elaboração de programas objetos ou à linguagem de máquina. Estes são alguns dos problemas:

a) Os programas em linguagem de máquina são difíceis de serem entendidos e corrigidos;
b) Os programas requerem um tempo relativamente grande para serem digitados, pois cada bit deve ser digitado individualmente;
c) Normalmente, é difícil, ao observarmos o programa, extrair do mesmo a tarefa que se deseja que o computador execute, dificultando, assim, qualquer tipo de alteração futura;
d) Os programas são extensos e tornam-se cansativos ao serem escritos;
e) Quando o programador comete algum erro, este é muito difícil de ser descoberto.

Devido à grande dificuldade de se trabalhar diretamente com linguagem de máquina que consiste, basicamente, na manipulação de códigos binários, adotaram-se os sistemas hexadecimal e octal, como sistemas intermediários entre o sistema binário do computador e o decimal do homem. A utilização do sistema hexadecimal para os microprocessadores de 8 bits proporcionou uma maior facilidade de visualização do programa. Por exemplo, o programa mostrado na figura 1 ao ser escrito com base no sistema hexadecimal ficaria mais simples, conforme nos mostra a figura 2.

No mínimo, a versão hexadecimal do programa (veja figura 2) é mais reduzida com relação à forma binária (veja figura 1). Os erros são mais fáceis de serem detectados e o trabalho de examinar o programa torna-se menos cansativo. É evidente que ao trabalharmos com programas em hexadecimal, é necessário dispormos de um outro programa intermediário, capaz de converter os códigos escritos em hexadecimal para binário. Isto se deve

ao fato do microprocessador trabalhar somente com códigos binários. Este programa intermediário é um programa padrão (desenvolvido junto com o microprocessador) também conhecido, no jarguão técnico, como programa Monitor.

```
3A
00
30
47
3A
00
40
80
32
00
50
```

FIGURA 2

Um programa Monitor não resolve, certamente, todos os problemas inerentes à programação em si, muito pelo contrário, pois a versão hexadecimal do programa ainda continua difícil de ser lida e entendido. Ele não faz, por exemplo, distinção entre instruções, dados e endereços. Veja os casos apresentados na figura 3.

Caso a:

$$1000\ 0000_{(2)} = 80_H$$

FIGURA 3

Caso b:

$$00111110\ 10000000 = 3\ E\ 80_H$$

No caso a, o número 80_H corresponde a uma instrução (para o microprocessador Z-80), enquanto que, no caso b, o mesmo número hexadecimal 80_H, acrescido de $3E_H$, corresponde a um dado da instrução precedente.

Para sanar essas dificuldades, adotou-se uma linguagem auxiliar, capaz de identificar, através de mnemônicos, quais são as instruções e os tipos de operações correspondentes. Antes de passarmos ao estudo dessa linguagem auxiliar, vejamos

o que vêm a ser os mnemônicos.

MNEMÔNICOS DE CÓDIGOS DE INSTRUÇÕES

Os mnemônicos surgiram (no caso de microprocessadores) como um aperfeiçoamento da programação e consiste em atribuir nomes a cada código de instrução. Basicamente, o mnemônico, além de identificar a instrução, ele deve descrever, de algum modo, o que a instrução faz. Normalmente, os fabricantes de microprocessadores determinam o conjunto de mnemônicos para o conjunto de instruções de seus microprocessadores.

Não há necessidade, por parte do usuário, em decorar o conjunto de mnemônicos de microprocessadores, pois, geralmente, o conjunto de mnemônicos são descritos em manuais de fabricantes, livros, revistas especializadas, etc.

Muito embora a utilização dos mnemônicos facilite bastante a compreensão dos programas, existe o problema de que nem todas as instruções possuem nomes óbvios. Algumas instruções possuem nomes óbvios, como, por exemplo, ADD, AND, OR, enquanto que outras apresentam contrações óbvias, como, por exemplo, SUB para subtração ou XOR para OR exclusiva. Mas existem instruções que não se enquadram nesses dois casos, como, por exemplo, WMP, SOB e outros apresentando-se como nomes completamente sem vínculo com a operação que executam.

Em nosso estudo, adotaremos o conjunto de mnemônicos, sugerido pela Zilog, fabricante do microprocessador Z-80. Com base neste conjunto de mnemônicos, o nosso programa de adição assume a forma mostrada na figura 4.

```
LD   A , (60H)
LD   B , A
LD   A , (61H)
ADD  A , B
LD   (62H) , A
```

FIGURA 4

Como podemos observar, através da figura 4, o programa continua longe do óbvio, mas no mínimo algumas partes são compreensíveis.

PROGRAMA ASSEMBLER

Assim como dispomos de um programa designado Monitor, o qual converte códigos hexadecimais para binário, entre outras funções, temos o programa chamado Assembler, também designado Montador (ou Tradutor), que converte códigos mnemônicos para seus respectivos códigos de máquina em binário.

O processo de conversão de mnemônicos para códigos de máquina, instrução por instrução, é chamado de assembly manual ou simplesmente assembly. Este tipo de tarefa é rotineiro e cansativo, normalmente, é atribuído a computadores. Ainda com relação ao nosso problema de adição, a figura 5 nos mostra seus códigos de instruções.

CÓDIGO DE INSTRUÇÕES (MNEMÔNICOS)	EQUIVALENTE HEXADECIMAL
LD A , (60H)	3A
LD B , A	47
LD A , (61H)	3A
ADD A , B	80
LD (62H) , A	32

FIGURA 5

Alguns microprocessadores complicam ainda mais a tarefa de tradução de códigos, pois possuem instruções com tamanhos de palavra variáveis, isto é, com 1, 2, 3 ou 4 bytes. Geralmente, o primeiro byte define a operação a ser executada, enquanto que os demais bytes definem dados, registradores (ou registros), endereços de memória e outros operandos necessários para que a instrução possa ser executada.

A tarefa de traduzir esses códigos, como já dissemos, é rotineira e cansativa, podendo ser transferida para o computador, utilizando-se, para tanto, um programa Assembler. Ele traduz o programa do usuário ou programa fonte, escrito com mnemônicos para um programa objeto ou programa em linguagem de máquina, o qual o computador pode executar.

A entrada de um programa Assembler é um programa fonte e a sua saída é um programa objeto (veja figura 6).

FIGURA 6

Muitos consideram a linguagem Assembler como sendo uma linguagem de baixo nível, uma vez que esta é basicamente uma linguagem de máquina com mnemônicos, em substituição aos códigos das instruções, estando, portanto, mais voltada para a linguagem de máquina do que para a linguagem humana.

As vantagens apresentadas pela utilização da linguagem Assembler, são:

a) utilização de menor espaço possível de memória;
b) grande velocidade de processamento;
c) fácil acesso aos bits de uma palavra;
d) maior flexibilidade de programação.

Estes são fatores positivos na utilização da linguagem Assembler em um computador. Entretanto existem os inconvenientes que são, da mesma forma, importantes:

a) exige grande quantidade de instruções;
b) difícil análise;
c) difícil detecção de erros e correção;
d) programação trabalhosa;
e) exige grande interação com o funcionamento interno do computador;
f) as instruções e os endereços de memória variam muito de computador para computador.

Vejamos, agora, as vantagens adicionais que um Montador/Assembler normalmente oferece:

a) possibilita ao usuário atribuir nomes para localizações de memórias, dispositivos de entrada e saída (E/S) e mesmo para conjunto de instruções;
b) permite ao usuário converter dados ou endereços de vários sistemas numéricos, por exemplo, decimal ou hexadecimal, para binário, e converte caracteres ASCII ou EBCDIC para o seu correspondente código binário;
c) executa algumas operações aritméticas como parte do processo assembly;

d) permite ao usuário determinar o local de memória, onde o programa será carregado e reservar áreas de memória para armazenamento temporário de dados e definir constantes em um programa;
e) permite incluir rotinas padrões, a partir da biblioteca de programas ou incluir outros programas no programa corrente;
f) permite que o usuário controle o formato da listagem e os dispositivos de entrada e saída utilizados;
g) permite, com certa facilidade, a inclusão, alteração ou exclusão de instruções em qualquer tempo.

Todas estas vantagens adicionais, naturalmente, envolvem custos adicionais e memória. Os microcomputadores, normalmente, são acompanhados de Assemblers muito mais simples do que os Assemblers que acompanham os computadores grandes, porém a tendência dos Assemblers é de expansão.

Um outro fator importante que não pode ser ignorado ao trabalhar com linguagem Assembler é a necessidade de se conhecer detalhadamente o computador ou microcomputador com o qual se está trabalhando.

Cabe ressaltar também o fato de que os programas feitos em linguagem assembly não são portáveis, ou seja, um programa desenvolvido para o Z-80 não "roda" no Motorola 6800. Isto decorre do fato de que cada microcomputador possui sua própria linguagem assembly, que reflete a sua arquitetura.

A solução para muitas das dificuldades associadas aos programas de linguagem Assembler é usar as linguagens de "alto nível", conhecidas, também, como linguagens de "procedimento orientado". Cada declaração em linguagem de alto nível executa uma função reconhecível e, geralmente, corresponde a muitas instruções de linguagem Assembler. Um programa chamado "Compilador" traduz o programa fonte, escrito em linguagem de alto nível, em código objeto ou instruções de linguagem de máquina.

Existem muitas linguagens diferentes de alto-nível, cada uma é dirigida para um determinado tipo de tarefa. Como exemplo de linguagens de alto nível podemos citar: COBOL (para aplicações comerciais), FORTRAN (para aplicações científicas), APL e BASIC (linguagens que são populares, uma vez que adotam os sistemas de compartilhamento de tempo), PL/1 (uma combina-

ção de FORTRAN, ALGOL e COBOL), etc...

De uma forma geral, as linguagens de alto nível tornam os programas mais fáceis e mais rápidos para serem escritos pelos programadores, além de não exigir destes, maiores conhecimentos sobre a arquitetura do computador que está sendo utilizado.

As vantagens decorrentes das linguagens de alto nível, geralmente, são:

a) maior facilidade na aprendizagem da programação;
b) simples conversão de programas de um sistema para outro;
c) fácil escrita e análise de programas;
d) permite melhor documentação de programas;
e) permite uma melhor exatidão nos testes dos programas;

Entretanto, as desvantagens são:

a) gasto de tempo utilizado para a compilação ou interpretação;
b) um programa objeto produzido pode ser menos eficiente que um programa escrito diretamente em linguagem de máquina;
c) a linguagem disponível pode não ser adequada ao tipo de problema a resolver.

Apesar dessas desvantagens, a programação em linguagem de alto nível se impôs, como sendo nitidamente mais eficiente, em termos de aplicação prática, devido à relativa rapidez e facilidade no desenvolvimento de programas.

FORMATO ASSEMBLER

O formato padrão de linguagem Assembler é mostrado na figura 7 e, como podemos notar, este é constituído de um conjunto de quatro (4) campos, os quais contêm as informações básicas, necessárias para que o programa em linguagem de máquina possa ser criado.

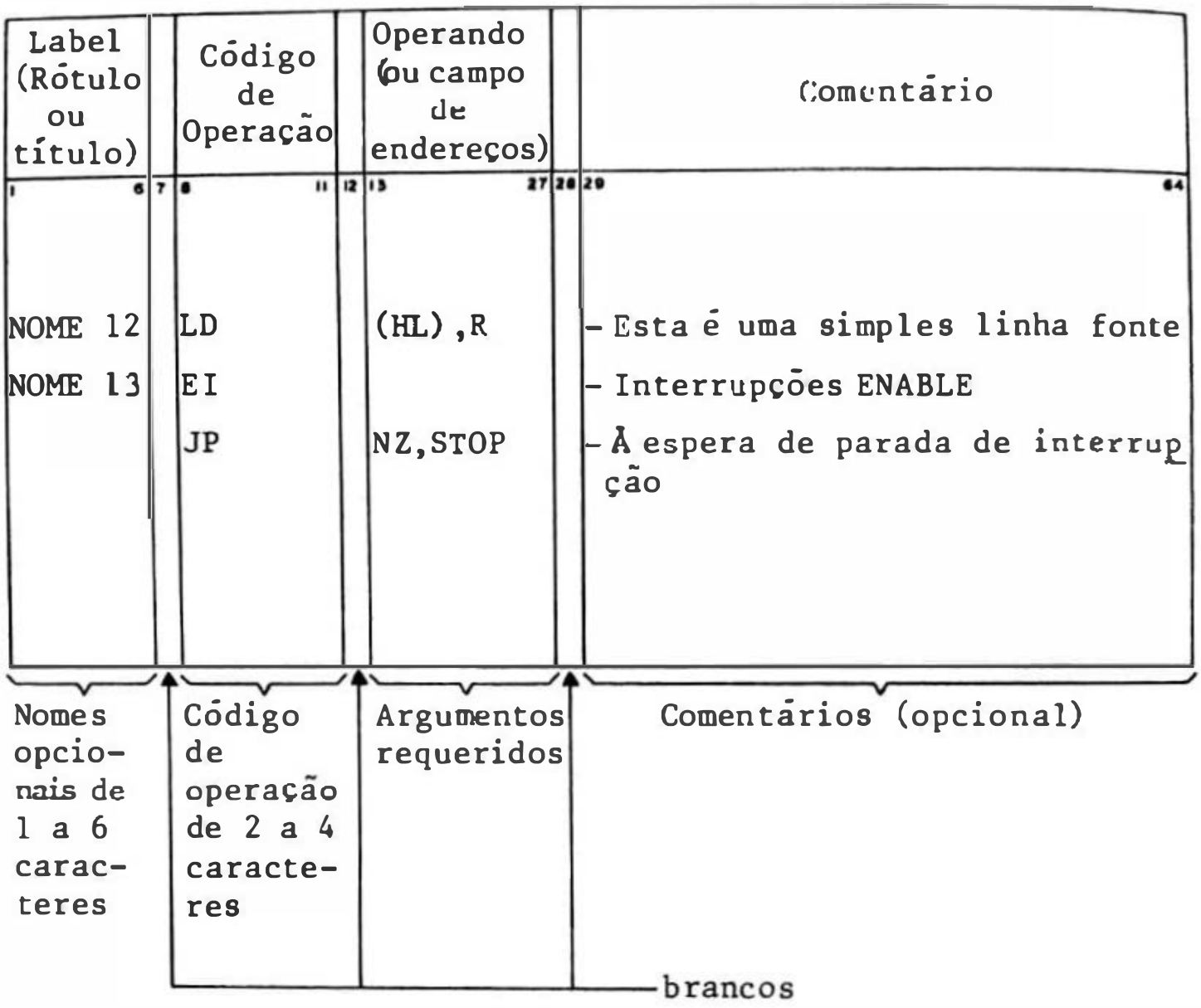


FIGURA 7

O campo "código de operação" (op code) é o único campo que nunca deve estar vazio, pois ele contém o mnemônico da instrução ou de uma pseudo-instrução. O campo referente ao título ou rótulo (Label) é opcional. Quando o rótulo é apresentado, ele pode possuir de 1 a 6 caracteres alfanuméricos, porém o primeiro caracter sempre deverá ser uma letra do alfabeto. Já, o campo do operando pode conter um endereço ou um dado e, eventualmente, pode apresentar-se em branco (ou vazio). O campo de comentários, por sua vez, também é opcional, uma vez que este é tido como uma conveniência pessoal, pois torna o programa mais fácil de ser codificado e lido. A figura 8 nos mostra o mesmo programa de soma da figura 5, só que, agora, escrito no formato do Montador/Assembler.

Label (Rótulo ou título)	Código de operação	Operando	Comentário
	ORG	40H	Define o endereço de carga do programa
Soma 1	EQU	60H	Define o endereço do primeiro operando
Soma 2	EQU	61H	Define o endereço do segundo operando
Desti	EQU	62H	Define o endereço de destino do resultado
Start	LD	A,(soma 1)	Carrega o primeiro operando
	LD	B,A	Salva o conteúdo de A
	LD	A,(soma 2)	Carrega o segundo operando
	ADD	A,B	Soma A com B e o resultado fica em A
	LD	(Desti),A	Move a soma para o seu destino
	END	Start	Define fim de programa e o local onde o programa se inicia.

FIGURA 8

Lendo na horizontal, as quatro colunas formam uma linha de programa em assembler. Em geral, o comprimento das linhas é determinado pelo comprimento de linhas nos dispositivos de entrada, tais como as leitoras de cartão perfurado.

Geralmente, cada uma dessas linhas de linguagem Assembler representa o conjunto completo de informação sobre uma determinada instrução do microprocessador, por exemplo, do Z-80. Cada linha irá então gerar de um a quatro bytes para representar uma instrução qualquer do Z-80. Uma das exceções desta regra diz respeito à uma linha de comentário, a qual é iniciada com um ponto e vírgula (;) e não é gerativa, ou seja, ela não gera nenhum código de máquina, servindo apenas para referência. No caso, o ponto e vírgula (;) é o que chamamos de caracter delimitador. Normalmente, este tipo de caracter pode ser usado no começo ou no fim de cada campo. A tabela 1 nos mostra alguns caracteres delimitadores típicos.

Caracter	Significado
:	depois de um título
'space'	entre o código de operação e o endereço
,	entre operandos no campo de endereço
;	antes de um comentário

TABELA 1

Ao se utilizar dos caracteres delimitadores, devem-se tomar alguns cuidados, tais como:

a) Não usar espaços extras, particularmente, depois de vírgulas que separam operandos;
b) Não usar caracteres delimitadores em nomes ou títulos de declarações (label);
c) Incluir os delimitadores padrões, mesmo se o assembler que você estiver usando não o exigir. Desta forma, os seus programas poderão ser montados por qualquer assembler.

Dissemos, a pouco, que há algumas linhas de programas, em Assembler, que não são gerativas e, nestes casos, os mnemônicos não geram instruções em linguagem de máquina, mas, sim, informam o programa Assembler de ações especiais a serem tomadas. Estas instruções recebem, normalmente, o nome de "pseudo-operações", as quais passamos a estudar em seguida.

PSEUDO-OPERAÇÕES

Trata-se de instruções para o próprio programa Assembler, as quais determinam certas áreas de memória, definem símbolos, designam áreas para armazenamento temporário de dados, fixam tabelas ou outros dados na memória e permitem referências a outros programas, etc.

Para usar estes diretivos de Assembler, ou pseudo-operações, um programador coloca o mnemônico da pseudo-operação no campo "código de operação" (op-code) e especifica os parâmetros requeridos no campo do operando.

As pseudo-instruções mais comuns são:

a) ORIGIN

b) END
c) EQUATE
d) DEFB
e) DEFW
f) DEFS
g) DEFM
h) DEFL

Vejamos, então, em que consiste cada uma destas pseudo-instruções:

a) ORIGIN — Esta pseudo-instrução permite ao programador localizar programas, subrotinas ou dados em qualquer lugar na memória, ou seja, permite ao programador determinar ao Assembler o local de memória, onde os programas devem ser carregados. O código mnemônico desta pseudo-instrução é ORG. Normalmente, os programas de microprocessadores, em linguagem Assembler, contêm várias declarações ORIGIN, podendo estas se enquadrarem em um dos seguintes objetivos:

- Reset (start up) endereço;
- Endereços de serviço de interrupção;
- Endereços de desvio (trap);
- Memória RAM;
- Pilha de memória;
- Programa principal;
- Subrotinas;
- Endereços de memória para dispositivos de E/S ou funções especiais.

Outras declarações ORIGIN podem liberar espaço para inserções futuras, colocar tabelas ou dados na memória ou fixar espaço de RAM para buffers de dados. A seguir, apresentamos algumas declarações ORIGIN típicas:

```
ORG     Reset
ORG     1200 H
```

Na primeira linha, o programa é montado, tendo, por endereço base, o valor atribuído ao label "reset", enquanto que, na segunda linha, o Assembler monta o programa, a partir do endereço 1200 H.

b) END — A pseudo-instrução END é a última declaração num programa e permite ao programador determinar o fim de um pro-

grama. Seu código mnemônico é o próprio END. A seguir, apresentamos alguns exemplos de aplicação desta pseudo-instrução:

```
END

END START
```

Na primeira linha, determina-se apenas o fim do programa e, na segunda, fica determinado que o programa inicia a partir do label "start".

c) EQUATE — A pseudo-instrução EQUATE permite ao programador relacionar dados ou endereços com labels ou nomes, isto é, atribui o valor numérico em seu campo de operando ao label (nome ou título) no seu respectivo campo de label. Seu código mnemônico é EQU. Eis alguns exemplos aplicativos:

```
VIDEO       EQU 4C 01 H
VIDEO 2     EQU VIDEO + 1
LAST        EQU 356
```

Note que, na primeira linha, é atribuído o endereço 4C01H ao label VIDEO. Na segunda linha, o label VIDEO 2 tem atribuído o endereço 4C02H, enquanto que, na terceira linha, o valor 356 é atribuído ao label LAST.

Cabe ressaltar que uma pseudo-instrução EQU não faz com que o Assembler coloque algo na memória, ela, simplesmente, entra com um título (label) adicional numa tabela de símbolos, mantida pelo próprio Assembler. As pseudo-instruções EQU devem ser colocadas, preferencialmente, no começo do programa, sob cabeçalhos apropriados de comentários, tais como: endereços de E/S, constante de tempo, localizações de programas, etc. Isto torna as definições mais simples de serem encontradas, em caso de haver interesse em mudá-las. Por outro lado, é aconselhável que as definições usadas somente numa sub-rotina específica devam aparecer no início da referida sub-rotina.

d) DEFB — Esta pseudo-instrução permite ao programador determinar o conteúdo de um byte no endereço corrente de memória, por exemplo:

```
DEFB  75
DEFB  HEX
```

Na primeira linha, temos a definição do caracter A no endereço corrente de memória (75 corresponde ao código ASCII de

W) e, na segunda linha, define-se o conteúdo do label HEX para o referido endereço corrente de memória.

e) DEFW — Esta instrução permite ao programador determinar o conteúdo de 2 bytes, a partir do endereço corrente de memória, por exemplo:

```
DEFW  42EAH
DEFW  HEX
```

Note que, na primeira linha, 42EAH é definido, a partir do endereço corrente de memória, enquanto que, na segunda linha, é definido o conteúdo do label HEX, também, a partir do endereço corrente de memória.

OBS.: As pseudo-instruções DEFB e DEFW definem as constantes e as variáveis num dado programa. É com base nessas duas pseudo-instruções, que o Assembler passa a gerar tabelas de dados, constantes, ou mesmo localizações para variáveis.

f) DEFS — Esta pseudo-instrução permite ao programador reservar uma área de memória, iniciando no endereço corrente de memória, por exemplo:

```
DEFS  60
DEFS  VALOR + 15
```

Na primeira linha, são reservados 60 bytes de memória, a partir do endereço corrente de memória. Na segunda linha, reserva-se o número de bytes pelo conteúdo do label VALOR, somando com 15, a partir do endereço corrente de memória. Em outras palavras, podemos dizer que o efeito desta pseudo-instrução é incrementar o contador de localização Assembler pelo operando, o qual representa o número de bytes a serem reservados.

g) DEFM — Esta pseudo-instrução permite ao programador definir um texto para ser utilizado pelo programa Assembler. Na montagem do programa objeto, o Montador/Assembler define os códigos ASCII de todos, os caracteres alfanuméricos e espe-

ciais, a partir do endereço de memória corrente. Vejamos um exemplo:

```
DEFM  'ENTRADA DE DADOS'
```

No exemplo apresentado, o texto 'ENTRADA DE DADOS' será gravado, a partir da posição corrente de memória, sendo que cada caracter será convertido para o seu correspondente código ASCII. Normalmente, o tamanho do texto está limitado ao número de caracteres permitidos por linha, na impressora ou no terminal de vídeo.

h) DEFL – Esta pseudo-instrução permite a atribuição de um determinado endereço a um label, por exemplo:

```
INICIO  DEFL  3235H
```

Neste caso, o endereço 3235H é atribuído ao label INICIO, sendo incluído na tabela de labels pelo programa montador.

As pseudo-operações, que acabamos de apresentar, são algumas das mais comumente utilizadas. Maiores informações sobre este assunto serão fornecidas no decorrer do curso.

Antes de finalizarmos esta lição, recomendamos ao aluno para que reveja as lições referentes ao microprocessador Z-80 (MC12, MC13 e MC14), principalmente, os assuntos referentes aos registradores (ou registros), modos de endereçamento e conjunto de instruções. Na próxima lição, iniciaremos um estudo sobre as instruções pertencentes ao conjunto de instruções do Z-80 e, em seguida, apresentaremos as regras de sintaxe do Assembler da Zilog.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-26

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Basicamente, uma instrução consiste de um número binário, que deve estar disponível no microprocessador, na ordem correta e no tempo apropriado, de modo a:

a) executar uma ação qualquer a esmo;

b) executar uma determinada ação desejada;

c) não executar nenhuma ação;

d) N.R.A.

2) Um programa nada mais é do que:

a) um conjunto de instruções capaz de fazer com que o computador execute uma determinada tarefa;

b) um conjunto de dados e endereços de memória;

c) os itens a e c se completam;

d) N.R.A.

3) Para um computador qualquer, todo programa é "visto", pelo mesmo, como sendo um conjunto de números:

a) binários;
b) hexadecimais;
c) decimais;
d) N.R.A.

4) A figura a seguir nos mostra um programa escrito de acordo com o sistema de numeração:

```
3A
00
30
47
3A
00
40
80
32
00
50
```

a) decimal;
b) binário;
c) octal;
d) hexadecimal.

5) A figura a seguir nos mostra o formato padrão da linguagem:

Label (Rótulo ou título)		Código de Operação		Operando (ou campo de endereços)		Comentário
1 6	7	8 11	12	13 27	28	29 64
NOME 12		LD		(HL),R		- Esta é uma simples linha fonte
NOME 13		EI				- Interrupções ENABLE
		JP		NZ,STOP		- À espera de parada de interrupção

Nomes opcionais de 1 a 6 caracteres

Código de operação de 2 a 4 caracteres

Argumentos requeridos

Comentários (opcional)

brancos

a) Assembler;
b) Basic;
c) Fortran;
d) N.R.A.

6) Basicamente, o mnemônico serve para:
a) identificar as instruções;
b) descrever de algum modo o que a instrução faz;
c) os itens a e b estão corretos;
d) N.R.A.

7) O programa Assembler também é conhecido por:
a) programa Montador;
b) programa Tradutor;
c) os itens a e b estão corretos;
d) N.R.A.

8) A linguagem Assembler apresenta como vantagem:
a) utilização de menor espaço possível de memória;
b) fácil acesso aos bits de uma palavra;
c) grande velocidade de processamento;
d) todos os itens estão corretos.

9) O formato Assembler é constituído por um conjunto de:
a) 4 campos;
b) 3 campos;
c) 8 campos;
d) 6 campos.

10) Os caracteres delimitadores, normalmente, são usados:

a) no fim de cada campo;

b) no começo de cada campo;

c) no meio de cada campo;

d) os itens a e b estão corretos.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-27

PROGRAMAÇÃO ASSEMBLER II

INTRODUÇÃO

Estamos praticamente prontos para começarmos a escrever programas em Assembler. Antes porém, necessitamos definir as instruções individuais do microprocessador Z-80, bem como as formas simbólicas utilizadas no Assembler da Zilog.

A descrição detalhada das instruções individuais será precedida por um apanhado geral do conjunto de instruções do microprocessador Z-80 (veja lição MC-14). Vamos então, iniciar o nosso estudo, revendo alguns fatos importantes sobre os registros ou registradores, flags de posição e modos de endereçamento.

REGISTROS DA CPU E FLAGS DE POSIÇÃO

O microprocessador Z-80 possui dois conjuntos de oito registradores de 8 bits, ambos de uso geral; dois registradores especiais de 8 bits e quatro registradores de 16 bits.

Como sabemos, um registrador é uma posição (ou localização) de memória interna que é utilizada durante o processamento de uma instrução. Um dos conjuntos de registradores de uso geral, de 8 bits, é denominado conjunto de registradores prin

cipais, enquanto o outro é denominado conjunto de registradores alternativos. O conjunto de registradores principais é utilizado durante a execução de uma instrução e o conjunto alternativo somente é utilizado por duas instruções, que trocam o conteúdo do conjunto principal com o conjunto alternativo.

Os registradores de uso geral, também conhecidos por registradores de função geral, são em número de quatorze, classificados em dois grupos (veja figura 1).

REGISTRADORES DE USO GERAL	
PRIMÁRIOS (OU PRINCIPAIS)	SECUNDÁRIOS (OU ALTERNATIVOS)
A	A'
B	B'
C	C'
D	D'
E	E'
H	H'
L	L'

FIGURA 1

O registrador A, também conhecido por acumulador, é o mais importante dos registradores, pois ele se constitui na primeira fonte e destino para instruções de um e dois operandos. Por exemplo, as transferências mais curtas e rápidas de dados entre a CPU e os dispositivos de E/S são realizadas através do acumulador. Uma instrução deve portanto carregar o acumulador antes que o microprocessador Z-80 possa realizar quaisquer operações aritméticas de 8 bits.

Os registradores B, C, D, E, H e L podem ser utilizados como registradores de 8 bits ou, em pares, como registradores de 16 bits. Neste caso, B e C, D e E, H e L são usados aos pares e são chamados BC, DE e HL, simplesmente.

O registrador F, também chamado de Flag ou registrador de condição, é assim denominado por seus bits indicarem várias condições internas e, também, por nunca ser usado diretamente

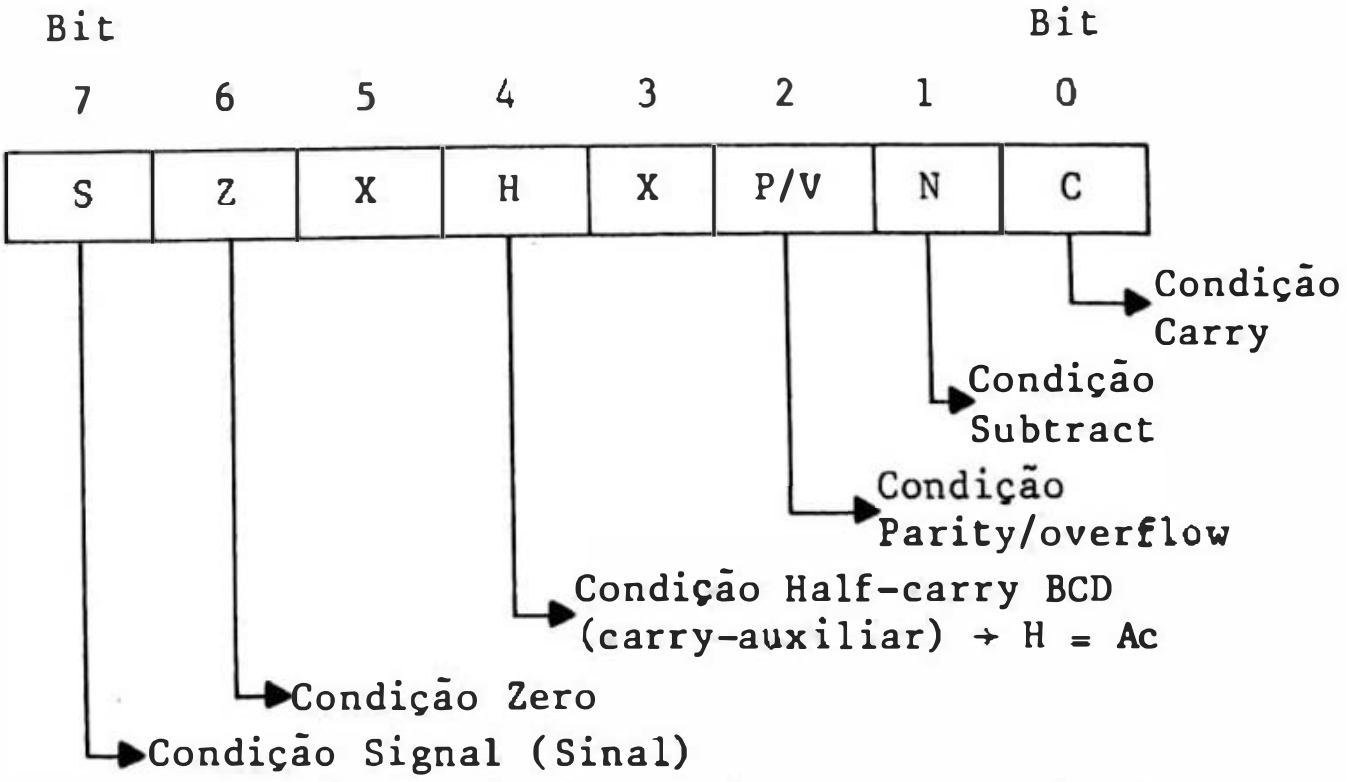
por uma instrução, pois os seus bits são automaticamente posicionados de acordo com o resultado da execução de uma determinada instrução. Tanto F como F' são registradores de 8 bits, agrupados convenientemente, de modo que cada um desses bits especificam várias condições da CPU que ocorrem após uma operação de lógica e aritmética ou após uma outra operação da CPU. A figura 2 nos mostra como é o agrupamento dos 8 bits do registrador Flag.

Os registradores F e F' possuem os seguintes bits de condição:

- Bit C (carry bit);
- Bit N (subtract bit);
- Bit P/V (Parity/overflow bit);
- Bit H (Half-carry BCD);
- Bit Z (Zero);
- Bit S (Sinal);

Com relação aos registradores F e F', resta observar ainda que estes são selecionados da mesma forma que o registrador A e nunca é utilizado para manipular dados. Todos os flags conservam os seus valores atuais até que uma instrução que os modifica seja executada. A simples mudança do valor do acumulador não mudará, necessariamente, o valor dos flags de posição.

FIGURA 2



Obs.: X = não utilizado (condição indeterminada).

Os registradores Ix e Iy, de 16 bits cada um, são conhecidos como registradores indexadores, permitindo o endereçamento indexado nos programas do microprocessador Z-80. Esses registradores são utilizados como ponteiros de posições de memória e, para que possamos determinar o local de memória que desejamos, é necessário adicionarmos um certo valor ao seu conteúdo.

Temos ainda mais dois registradores de 16 bits: Stack-Pointer (SP) e Contador de Programa (PC). O Stack-Pointer possui a função de ponteiro de pilha e é um indicador de empilhamento da memória externa. O endereço contido no Stack-Pointer aponta para a área livre de memória, a qual é usada para armazenamento temporário de dados durante a execução de um programa.

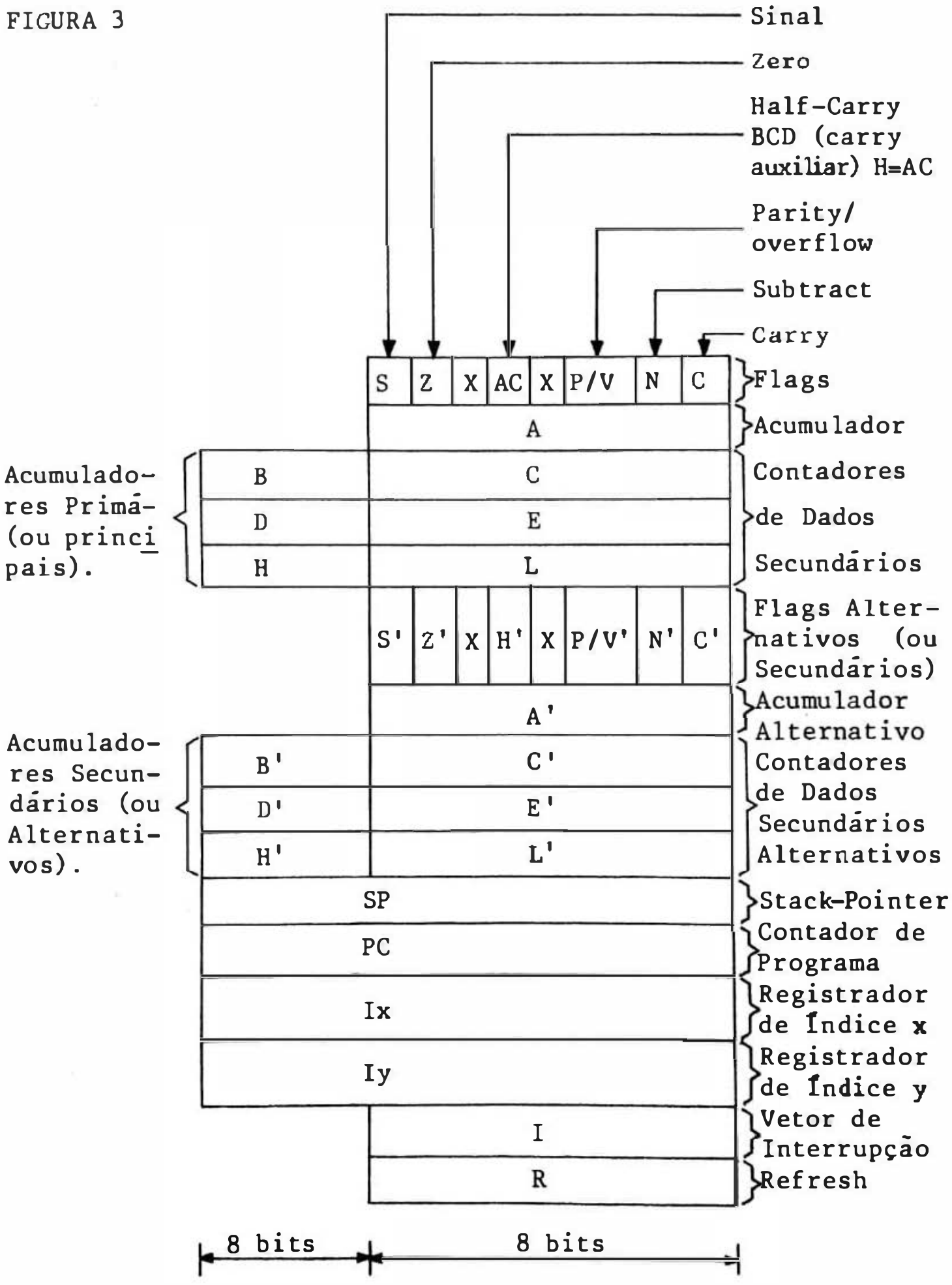
O PC (Contador de Programa ou "Program Counter") tem por função determinar a ordem de execução das instruções. Este registrador mantém o endereço da instrução que está sendo executada, durante o tempo de busca da mesma na memória. Assim que o conteúdo do PC seja transferido para as linhas de endereço, este é automaticamente incrementado e é atualizado ao término de cada instrução, de modo a conter o endereço da próxima instrução a ser executada. O conteúdo do PC pode ser modificado por uma instrução de desvio pois estas alteram o curso normal do programa.

Restam-nos ainda mais dois registradores de 8 bits: Interrupção (I - Interrupt Register) e Refresh (R - Memory Refresh). O registrador Refresh (R) tem apenas 7 dos seus 8 bits utilizados. Sua principal função é controlar o "refresh" (refrescamento) automático de memórias semicondutoras do tipo RAM dinâmicas. Este registrador está voltado para auxiliar o hardware existente no microcomputador, sendo totalmente transparente para o programador.

O registrador de Interrupção (I), também conhecido como registro "vetor de interrupção", possibilita uma maior flexibilidade no sistema de interrupções do microprocessador Z-80. As interrupções são utilizadas apenas nos sistemas de tempo real (real time), envolvendo uma programação avançada.

A figura 3 nos mostra o mapa de registros do microprocessador Z-80, mostrando também, o agrupamento dos 8 bits do registrador Flag.

FIGURA 3


Sinal
Zero
Half-Carry
BCD (carry auxiliar) H=AC
Parity/ overflow
Subtract
Carry
S Z X AC X P/V N C
Flags
A
Acumulador
Acumuladores Primários (ou principais).
B C
D E
H L
Contadores de Dados Secundários
S' Z' X H' X P/V' N' C'
Flags Alternativos (ou Secundários)
A'
Acumulador Alternativo
Acumuladores Secundários (ou Alternativos).
B' C'
D' E'
H' L'
Contadores de Dados Secundários Alternativos
SP
Stack-Pointer
PC
Contador de Programa
Ix
Registrador de Índice x
Iy
Registrador de Índice y
I
Vetor de Interrupção
R
Refresh
8 bits
8 bits

Vamos rever agora alguns fatos importantes sobre os modos de endereçamento da memória, referentes ao microprocessador Z-80.

MODOS DE ENDEREÇAMENTO DE MEMÓRIA

Não pretendemos estudar aqui, detalhadamente, todos os modos de endereçamento de memória do Z-80, pois este tipo de estudo foi feito na lição MC-14 (Microprocessador Z-80 III). Iremos fazer apenas alguns comentários sobre os modos de endereçamento, de forma bem resumida e objetiva.

Para executar alguma operação envolvendo memória, o microcomputador necessita conhecer os endereços das localizações de memórias envolvidas. O modo de endereçamento vem a ser as maneiras pelas quais as instruções do microprocessador Z-80 podem designar os seus operandos. Os modos de endereçamento do Z-80 são em número de onze (11), a saber:

a) *Modo Imediato*: neste modo de endereçamento um byte contido na instrução é movido para o registrador. É muito comum para instruções LOADs, Aritméticas, etc.

b) *Modo Imediato Extendido*: este modo de endereçamento é muito utilizado para carregar constantes em registros (ou registradores) de 16 bits, tais como Ix, Iy e SP e em pares de registros como por exemplo BC, HL e DE. Assim, podemos dizer que os dois bytes contidos nestas instruções são carregados em um par de registradores.

c) *Modo Implícito*: neste modo de endereçamento, as instruções não possuem campo de operando, pois o próprio código de operação determina qual o registro da CPU irá conter o operando. Muito utilizado nas instruções aritméticas, NEG, SCP e NOP.

d) *Modo Página Zero Modificada*: este modo de endereçamento é aplicado somente a instruções de RESTART (RST), as quais são comumente utilizadas em programas como "sub-rotinas".

e) *Modo Relativo*: este modo de endereçamento é aplicado somente a instruções de desvios relativos (JR), permitindo a execução de JUMP's condicionais e incondicionais. O valor do segundo byte (chamado de deslocamento), somado ao conteúdo do contador de programa, nos dá um certo resultado, o qual repre-

senta o endereço efetivo da instrução que deve dar continuidade ao programa. Normalmente o endereço indicado deve estar no intervalo -128 a +128 bytes. (00-7F / 80-FF)
(+) (-)

f) *Modo Extendido ou Direto*: neste caso, a instrução contém no seu próprio código o endereço final do operando com dois (2) bytes que seguem a partir do código de operação (op code) da instrução. Esta técnica permite o acesso a uma posição aleatória de memória.

g) *Modo Indexado*: este é o modo de endereçamento mais completo do microprocessador Z-80. Neste caso, o endereço efetivo do operando na memória é obtido somando-se o conteúdo de um dos dois registros índices Ix ou Iy ao byte de deslocamento, no campo do operando da instrução. O resultado desta soma é um endereço que deve estar na faixa de -128D a +127D, isto em relação ao conteúdo ou valor armazenado no registro índice.

h) *Modo por Registro*: neste modo de endereçamento, um ou mais registros do microprocessador Z-80 são utilizados como operandos da instrução e o conteúdo de um registrador é carregado para outro.

i) *Modo por Registro Indireto*: neste modo de endereçamento, o operando fica em uma posição de memória, cujo endereço está contido em um par de registros BC, DE ou HL. Esta técnica permite o acesso de posições de memória de endereço aleatório. Geralmente este tipo de endereçamento é utilizado pelas instruções: aritméticas, lógicas, incrementais, decrementais, etc. Cabe observar que todas essas instruções citadas são instruções de um byte.

j) *Modo por Bit*: este modo de endereçamento permite acessar individualmente qualquer um dos oito bits que compõem um byte. Por isso, este modo de endereçamento é aplicado para testar, resetar e colocar um bit de um registrador. Esta técnica pode ser utilizada juntamente com os demais modos de endereçamento.

l) *Modo por Ponteiro de Pilha (Stack-Pointer)*: neste modo de endereçamento, o operando fica na posição de memória endereçada pelo ponteiro de pilha (SP). Como exemplos típicos de endereçamento por ponteiro de pilha, podemos citar as instruções PUSH e POP.

Finalizando esta breve revisão, resta-nos dizer que boa

parte das vantagens oferecidas pelo Z-80 sobre os outros microprocessadores de 8 bits, se deve justamente a essa variedade de modos de endereçamento. Vamos agora passar ao estudo das instruções do microprocessador Z-80.

OPERAÇÕES COM O Z-80

Quando o microprocessador Z-80 executa uma determinada operação, esta pode afetar, ou mesmo ser afetada, por três diferentes tipos de operandos (ou argumentos).

O primeiro tipo de operando diz respeito ao uso de um ou mais registradores, ou seja: os registradores principais (A, B, C, D, E, H e L), os registradores Indexadores (Ix e Iy), o Stack-Pointer (SP) e o Contador de Programa (PC). O microprocessador Z-80 trata o operando HL para se referir ao endereço de memória, indicado pelo par de registradores HL. O conjunto alternativo (ou secundário) de registradores somente é utilizado pelas instruções EXX e EX AF, AF', as quais trocam o conteúdo dos registradores principais (ou primários) com o conjunto alternativo de registradores.

Já, o segundo tipo de operando afeta um ou mais endereços (ou localizações) de memória do microcomputador. Poucas instruções podem afetar blocos de dados na memória, pois a maior parte delas possui condições de afetar apenas um ou dois bytes. A título de exemplo deste tipo de operando podemos destacar:

LDI/LDIR
LDD/LDDR
CPI/CPIR
CPD/CPDR

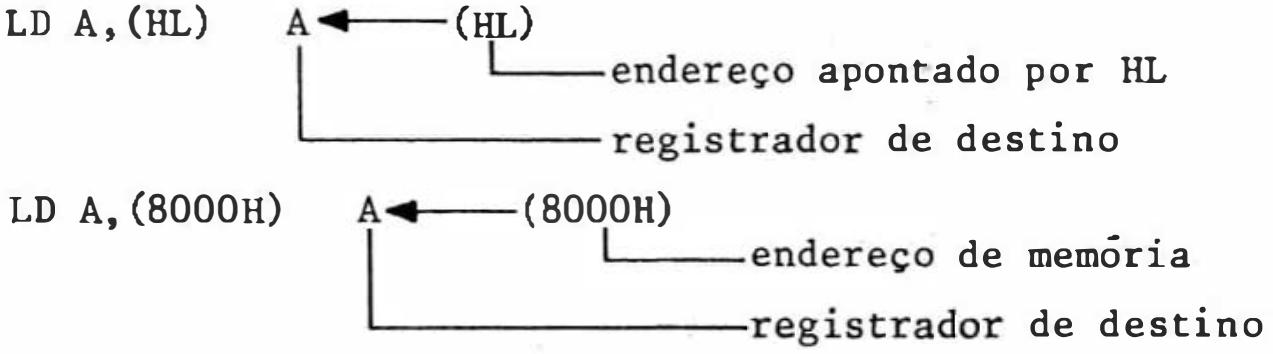
Por fim, o terceiro tipo de operando inclui os códigos de condição. Algumas vezes um código de condição é indicado na própria instrução, como no caso de um desvio quando o flag Z é "não zero", ou seja, "um" binário. Outras vezes, um ou mais códigos de condição são ajustados (set) de acordo com os resultados de uma operação. Neste caso, é preciso indicar quais flags são afetados por cada instrução.

Também são de grande valia as informações sobre a execução de uma instrução do Z-80, as quais referem-se: ao número de bytes que a instrução ocupa; ao tempo de execução da instru

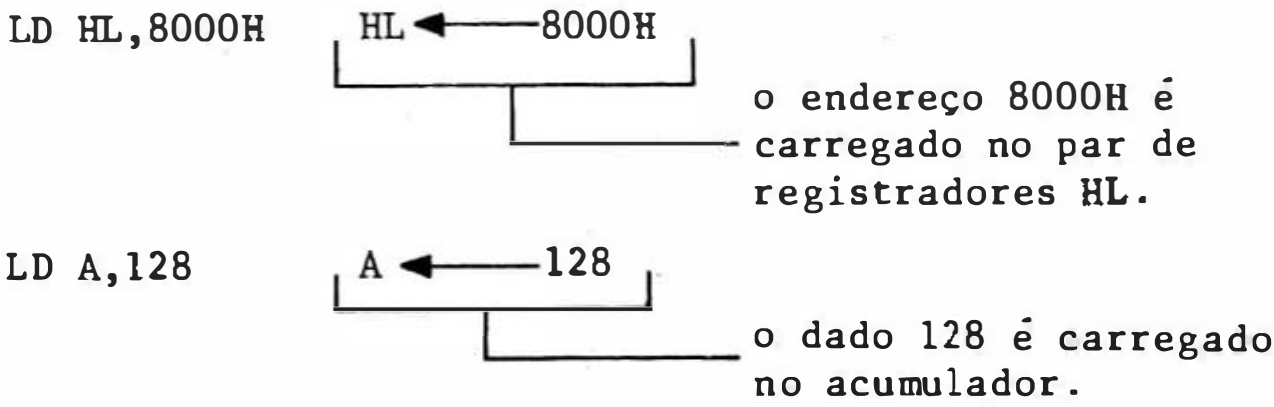
ção e ao seu código de máquina.

De acordo com a terminologia da Zilog, a ordem dos operandos indica a função dos itens envolvidos nas instruções de transferência de dados. O primeiro operando se refere ao destino, enquanto o segundo indica a origem dos dados. Por exemplo, a instrução LD A,B indica que o conteúdo do registrador B será copiado no acumulador A. Por outro lado, a instrução LD B,A indica que o conteúdo do acumulador será copiado no registrador B.

Se um operando está entre parênteses, significa que o operando está se referindo ao conteúdo de memória que está sendo apontado pelo registrador ou pelo endereço de memória que se encontra entre parênteses. Por exemplo:



Por outro lado, os operandos que não se encontram entre parênteses, referem-se a dados imediatos ou endereços de memória, por exemplo:



Vejamos agora mais informações sobre o conjunto de instruções do microprocessador Z-80.

CONJUNTO DE INSTRUÇÕES DO Z-80

Na lição MC-14, apresentamos o conjunto de instrução do

Z-80, dividido em onze (11) grupos de instruções, conforme sugere o fabricante (Zilog). Na descrição das instruções, que pretendemos iniciar agora, são utilizadas algumas abreviações, as quais passamos a mostrar em seguida.

ABREVIAÇÕES	SIGNIFICADO
A,F,B,C,D,E,H,L	Registradores de 8 bits. Sendo A = acumulador e F = flag.
AF',BC',DE',HL'	Pares de registradores alternativos (ou secundários).
addr	Endereço de memória de 16 bits.
x(b)	O bit b de um registrador de 8 bits ou de uma locação de memória x.
cond	Condições para desvio do programa. As condições são as seguintes: NZ = Não-Zero (Z=0) Z = Zero (Z = 1) NC = Não-Carry (C=0) C = Carry (C=1) PO = Paridade Ímpar (P=0) PE = Paridade Par (P=1) P = Sinal Positivo (S=0) M = Sinal Negativo (S=1)
data	Uma unidade de dados binários de 8 bits.
data 16	Uma unidade de dados binários de 16 bits.
disp	Um endereço de deslocamento de 8 bits com sinal.
xx(HI)	Trata-se dos 8 bits mais significativos de uma palavra de 16 bits.
I	Registrador de vetor de interrupção (8 bits).
Ix,Iy	Registradores indexadores de 16 bits cada um.
label	Um rótulo de memória.
xx(LO)	Trata-se dos 8 bits menos significativos de uma palavra de 16 bits.
LSB	Bit menos significativo (bit 0).

ABREVIAÇÕES	SIGNIFICADO
MSB	Bit mais significativo (bit 7).
PC	Contador de Programa.
port	Uma porta de entrada/saída (E/S) de 8 bits.
pr	Qualquer um dos seguintes pares de registradores: BC DE HL AF
R	Registrador de "refresh" das memórias dinâmicas. Possui 8 bits.
reg	Qualquer um dos seguintes registradores: A B C D E H L
rp	Qualquer um dos seguintes pares de registradores: BC DE HL SP
SP	Stack-Pointer (16 bits).
xy	Qualquer um dos registradores indexadores (Ix, Iy).
Object Code	Código objeto: bbb - número binário: 000(LSB) ou 111(MSB) ccc - código de condição 000 = não-zero 001 = zero 010 = carry 100 = paridade ímpar 101 = paridade par 110 = sinal positivo 111 = sinal negativo

ABREVIAÇÕES SIGNIFICADO

ddd - registrador de destino: o mesmo código de rrr.
ppqq - um endereço de memória de 16 bits.
rrrr - registradores: 111 = A
000 = B
001 = C
010 = D
011 = E
100 = H
101 = L
sss - registrador fonte: mesmo código de rrr
x - registrador índice: 0 = Ix
1 = Iy
xx - par de registradores: 00 = BC
01 = DE
10 = HL
11 = SP (rp) ou AF(pr)
xxx - código de restart (000 até 111)
yy - uma unidade de dados binários de 8 bits.
yyyy - uma unidade de dados binários de 16 bits.

Status - O Z-80 tem os seguintes flags de posição:
C - posição de Carry
Z - posição de Zero
S - posição de Sinal
P/O - posição de Paridade/Overflow
AC - posição de Carry Auxiliar
N - posição de Subtração

Os seguintes símbolos são usados nas colunas de posição:
X - Flag é afetada pela operação.
(space) - Flag não é afetada pela operação.
1 - Flag é ativada (ou setada) pela operação.
0 - Flag é desativada (ou ressetada) pela operação.
U - Flag é desconhecida após a operação.
P - Flag mostra a posição (status) ou estado, de paridade.
0 - Flag mostra a posição (status) ou estado, de overflow.
I - Flag mostra a posição (status) de habilitação/desabilitação de interrupção.

ABREVIAÇÕES	SIGNIFICADO
[[ ]]	Endereçamento de memória: 1) O conteúdo de uma locação de memória cujo endereço está contido no registrador; 2) Uma porta de E/S cujo endereço está contido no registrador designado.
[ ]	Os conteúdos de um registrador ou localização de memória. Por exemplo: [[HL]] ← [[HL]] + 1 indica que o conteúdo da localização de memória endereçada pelo conteúdo de HL é incrementado. Outro exemplo: [HL] ← [HL] + 1 indica que o par de registradores HL tem seu conteúdo incrementado.
$\wedge$	Operação lógica AND (E).
$\vee$	Operação lógica OR (OU).
$\not\vee$	Operação lógica OR-Exclusive (OU-Exclusiva).
←	Dados são transferidos na direção da seta.
← →	Dados são transferidos entre as localizações designadas, ou seja, são trocados.

Acrescentamos ainda que a parte fixa de uma instrução de linguagem assembly é mostrada sob a forma de letras maiúscula, enquanto que a parte variável (dados imediatos, número de dispositivo de entrada/saída (E/S), nome de registro, título ou endereço) é mostrada sob a forma de letra minúscula.

Uma outra informação importante diz respeito ao tempo de execução da instrução. O tempo real pode ser obtido dividindo o número de períodos (ciclos) de clock pela freqüência de clock. Por exemplo, para uma instrução que requer 7 períodos de clock, sendo a freqüência de clock igual a 4 MHz, teremos um tempo de execução igual a 1,75 microssegundos. Quando dois tempos de execução forem indicados, por exemplo, 4/12, significa que o número de períodos de clock depende dos flags de posição, sendo que o primeiro tempo (4) refere-se à "condição não encontrada" e o segundo (12) refere-se à "condição encontrada".

Na descrição individual das instruções do microproces-

sador Z-80, o efeito da execução de uma instrução sobre o "status" (positivo) será ilustrado na figura 4.

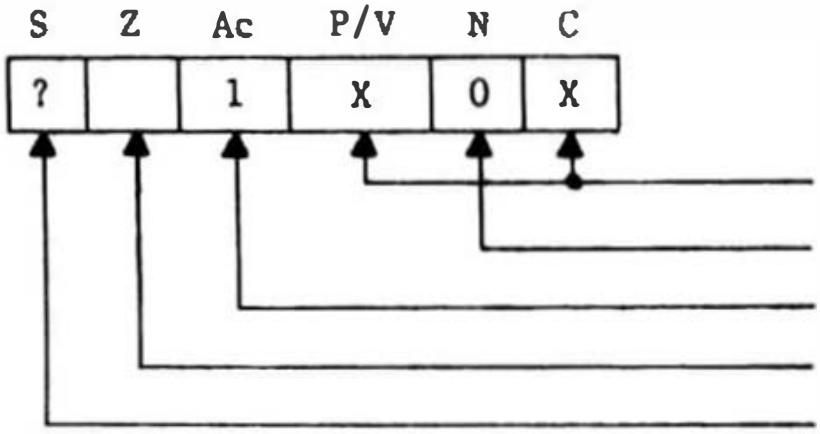


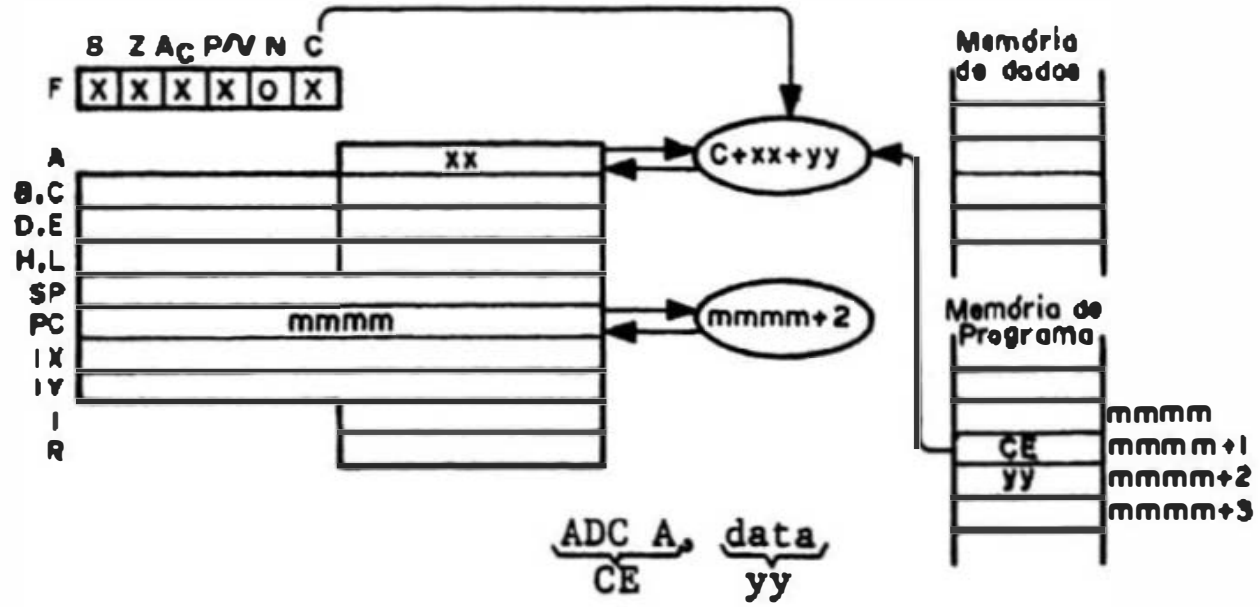
FIGURA 4

Obs.: Ac = H = carry auxiliar

O X identifica um "status" que é set ou reset. O 0 (zero), por sua vez, identifica um "status" que é sempre apagado (clear). Um 1 identifica um "status" que é sempre set (ajustado). Um "espaço em branco" significa que o "status" não muda, enquanto que um ponto de interrogação (?) significa que o "status" não é conhecido.

Visto esses pormenores referentes ao conjunto de instruções do microprocessador Z-80, vamos então dar início à descrição individual das instruções desse microprocessador, iniciando pela instrução "ADC A,dado".

1) ADC A,data - SOMA IMEDIATA COM CARRY NO ACUMULADOR.



Adicione o conteúdo do próximo byte da memória de programa ao carry bit no acumulador.

Suponhamos: xx = $3A_{16}$

yy = $7C_{16}$

Carry = 0

Após a execução da instrução ADC A,7CH, o acumulador irá conter $B6_{16}$, veja:

```
   3A = 0011 1010
   7C = 0111 1100  +
Carry =         0
       ----------
       1011 0110
```

1, ajusta S para 1.

Não carry, C é ajustado para 0.

Resultado não-zero, Z é ajustado (set) para 0.

Carry, AC ajustado para 1.

Instrução de adição, N é ajustado para 0.

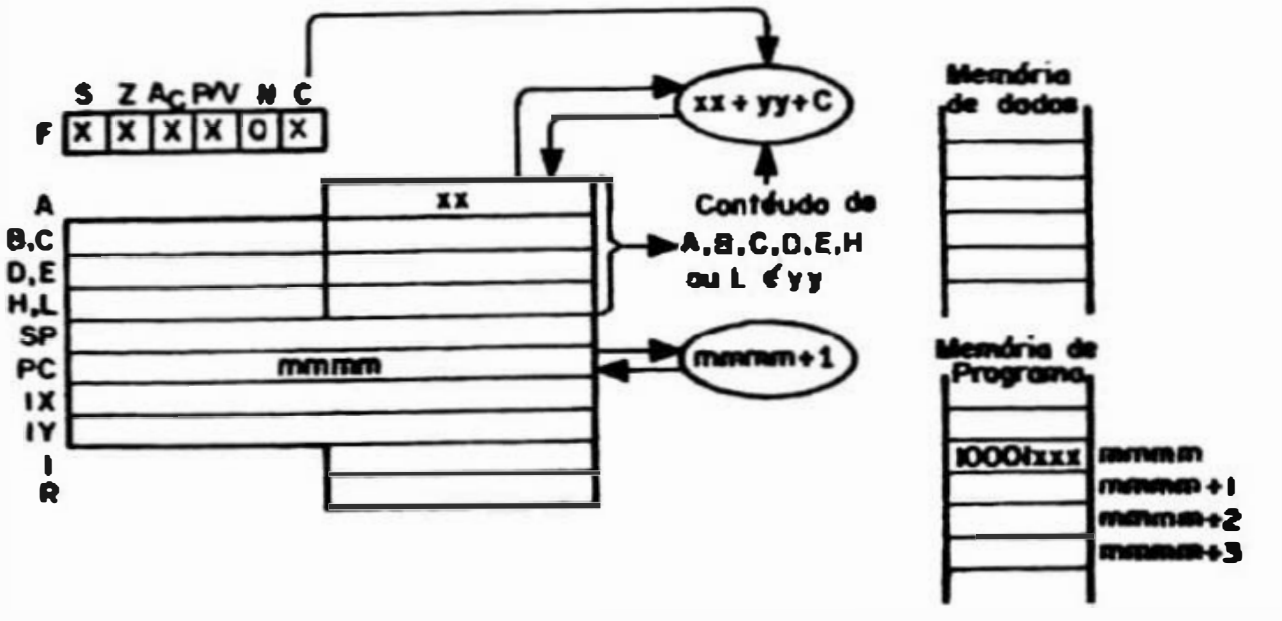
0 ⊕ 1 = 1, P/V é ajustado para 1.

Essa instrução (ADC) é freqüentemente usada na adição de multibyte para o segundo byte e o byte subseqüente.

Obs.: $3A_{16}$ = 3AH

16 e H indicam que se trata de um número hexadecimal.

2) ADC A, reg - SOMA O CONTEÚDO DO CARRY BIT E O CONTEÚDO DO REGISTRADOR r COM O CONTEÚDO DO ACUMULADOR.



ADC A, reg

10001 xxx

000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Adicione os conteúdos dos registradores A, B, C, D, E, H ou L e o carry bit ao acumulador.

Suponhamos: xx = $E3_{16}$
registrador E contém $A0_{16}$
carry = 1

Após a execução da instrução ADC A,E, o acumulador irá conter 84_{16}, veja:

```
    E3 = 1110 0011
    A0 = 1010 0000  +
 Carry =         1
        ----------
        1000 0100
```

1, ajusta S para 1

Carry, C é ajustado para 0

1 ⊻ 1 = 0, P/V é ajustado para 0

Resultado não-zero, Z é ajustado (set) para 0.

Não carry, Ac é ajustado para 0.

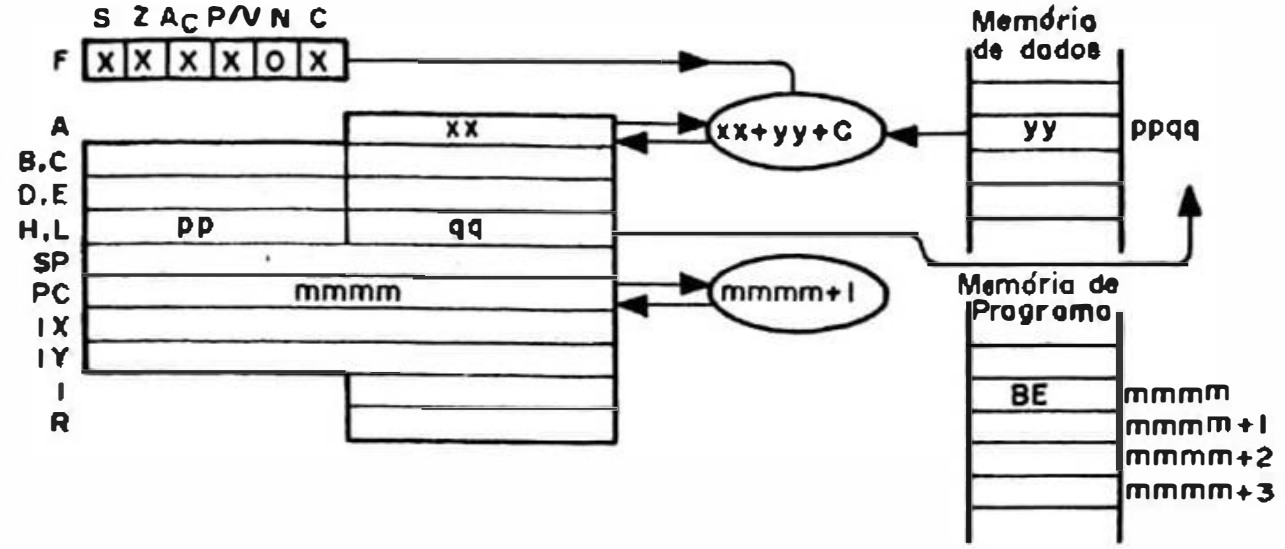
Instrução de adição, N é ajustado para 0.

A instrução ADC é mais freqüentemente usada na adição de multibyte para o segundo byte e o byte subseqüente.

3) ADC A,(HL) - SOMA O CONTEÚDO DO CARRY BIT + O CONTEÚDO DA MEMÓRIA COM O CONTEÚDO DO ACUMULADOR.

ADC A,(Ix + disp)
ADC A,(Iy + disp)

Neste caso, (HL), (Ix + disp) e (Iy + disp) indicam localizações ou endereços de memória.



A ilustração nos mostra a execução da instrução ADC A, (HL):

$$\underbrace{\text{ADC A,(HL)}}_{8E}$$

Adiciona o conteúdo da localização de memória, endereçada pelo conteúdo do par de registrador HL, e o carry bit ao acumulador.

Suponhamos: $xx = E3_{16}$

$yy = A0_{16}$

carry = 1

Após a execução da instrução ADC A,(HL), o acumulador irá conter 84_{16}. Veja:

```
   E3 = 1110 0011
   A0 = 1010 0000  +
Carry =         1
       ----------
       1000 0100
```

1, ajusta S para 1

Resultado não-zero, Z é ajustado (set) para 0.

Não carry, Ac é ajustado para 0.

Carry, C é ajustado para 1

Instrução de adição, N é ajustado para 0.

$1 \oplus 1 = 0$, P/V é ajustado para 0

$$\underbrace{\text{ADC A,(Ix}}_{\text{DD 8E}} + \underbrace{\text{disp)}}_{\text{d}}$$

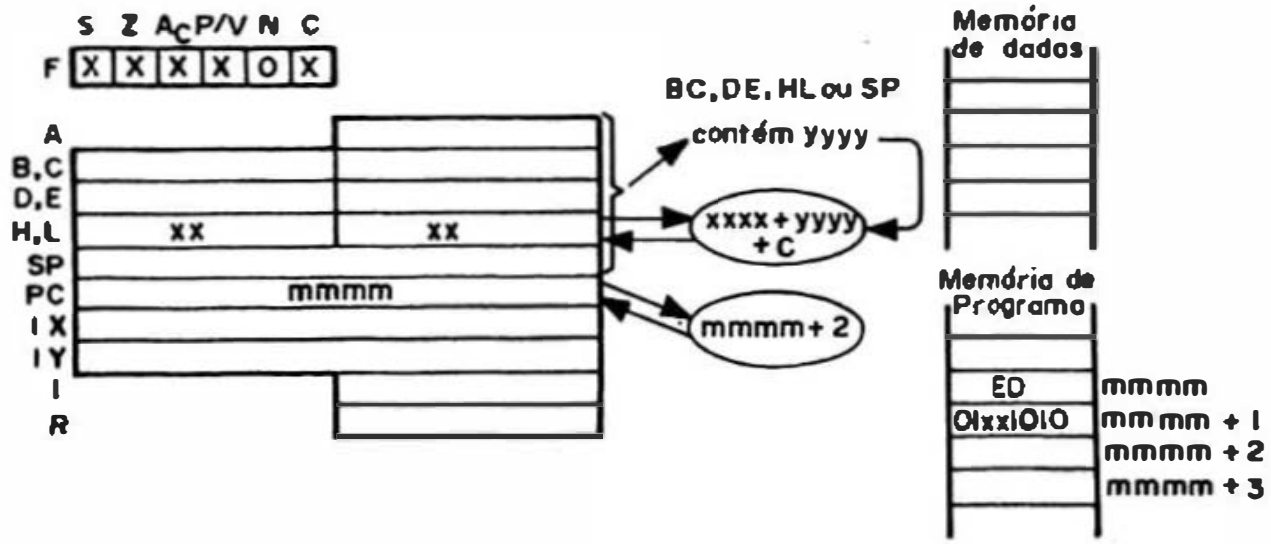
Soma o conteúdo da localização de memória (endereçada pela soma do conteúdo do registrador Ix e o dígito de deslocamento d) e o carry bit ao acumulador.

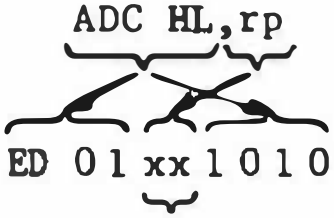
$$\underbrace{\text{ADC A,(Iy}}_{\text{FD 8E}} + \underbrace{\text{disp)}}_{\text{d}}$$

Esta instrução é idêntica à ADC A,(Ix + disp), exceto que ela utiliza o registrador Iy em vez do registrador Ix.

A instrução ADC é mais freqüentemente usada na adição de multibyte, para o segundo byte e o byte subseqüente.

4) ADC HL,rp - SOME O PAR DE REGISTROS COM CARRY A H e L.





00 para rp é o par de registradores BC
01 para rp é o par de registradores DE
10 para rp é o par de registradores HL
11 para rp é o Stack-Pointer

Some o valor de 16 bit do par de registradores BC, DE, HL ou o Stack-Pointer, juntamente com o carry bit, ao par de registros HL.

Suponhamos que HL contenha $A536_{16}$, BC contenha 1044_{16} e, carry seja igual a 1. Após a execução da instrução ADC HL,BC, o par de registros HL irá conter:

```
A536 = 1010 0101 0011 0110
1044 = 0001 0000 0100 0100 +
carry =                   1
        1011 0101 0111 1011
```

1, ajusta S para 1

Não carry, C é ajustado para 0

Resultado não-zero, Z é ajustado para 0.

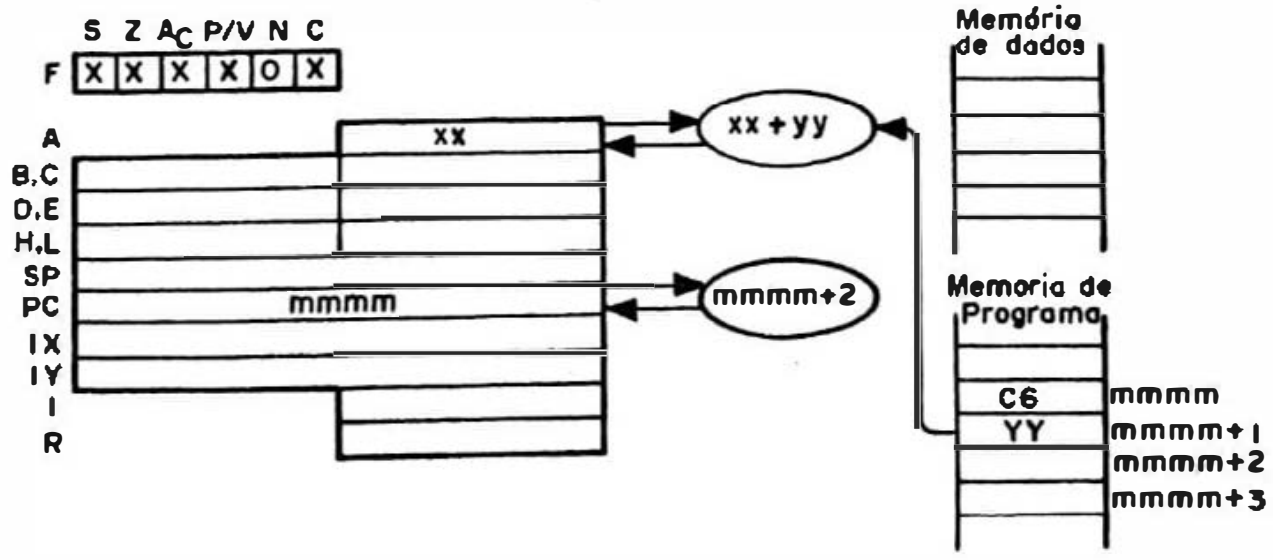
Não carry, AC é ajustado para 0.

Instrução de adição, N é ajustado para 0.

0 ⊽ 0 = 0, P/V é ajustado para 0.

A instrução ADC é mais freqüentemente usada na adição de multibyte, para o segundo byte e o byte subseqüente.

5) ADD A,data - SOME IMEDIATAMENTE NO ACUMULADOR.



ADD A, data

C6 yy

Some o conteúdo do próximo byte da memória de programa ao acumulador.

Suponhamos: $xx = 3A_{16}$

$yy = 7C_{16}$

carry = 0

Após a execução da instrução ADD A,7CH, o conteúdo do acumulador será $B6_{16}$. Veja:

```
    3A = 0011 1010
    7C = 0111 1100  +
carry =         0
       ------------
       1011 0110
```

1, ajusta S para 1

Não carry, C é ajustado para 0

Resultado não-zero, Z é ajustado para 0.

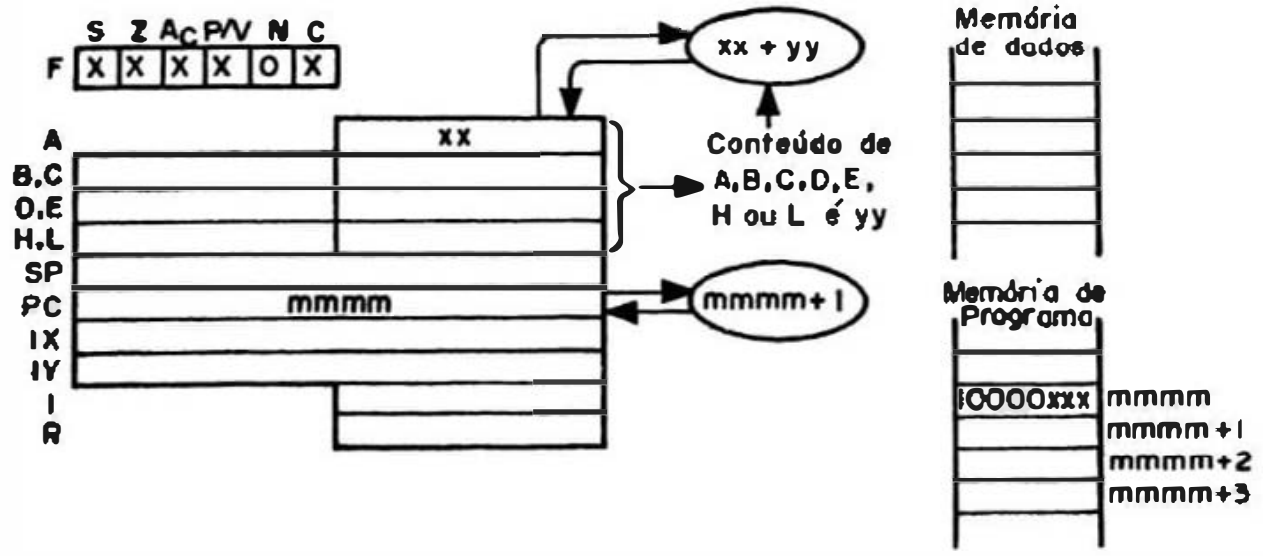
Carry, AC é ajustado para 1.

Instrução de adição, N é ajustado para 0.

0 ⊕ 1 = 1, P/V é ajustado para 0.

Esta é uma instrução de manipulação de dados de rotina.

6) ADD A,reg - SOME O CONTEÚDO DOS REGISTRADO ES AO ACUMULADOR.



ADD reg
10000 xxx

000 para o reg = B
001 para o reg = C
010 para o reg = D
011 para o reg = E
100 para o reg = H
101 para o reg = L
111 para o reg = A

Some o conteúdo dos registradores A, B, C, D, E, H ou L ao acumulador.

Suponhamos: $xx = E3_{16}$
$reg.\ E = A0_{16}$

Após a execução da instrução ADD A,E, o conteúdo do acumulador será 83_{16}. Veja:

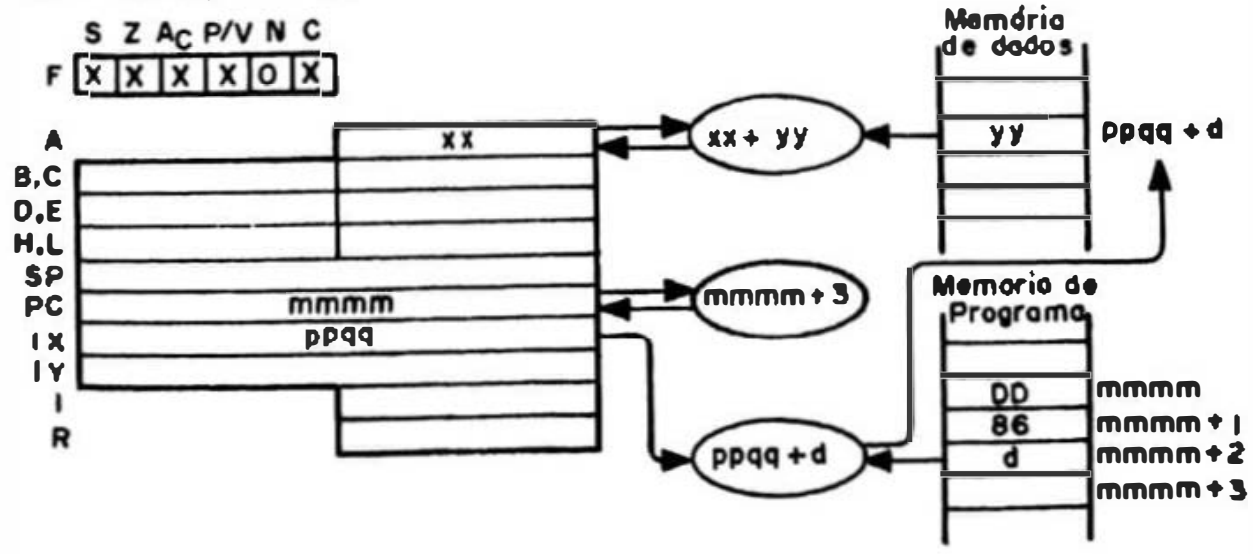
E3 = 1110 0011
A0 = 1010 0000 +
1000 0011

Resultado não-zero, Z é ajustado para 0.
Não carry, AC é ajustado para 0.
Instrução de adição, N é ajustado para 0.
1 ⊻ 1 = 0, P/V é ajustado para 0.

1, ajusta S para 1
Carry, C é ajustado para 0

Esta é uma instrução de manipulação de dados de rotina.

7) ADD A,(HL) - SOME O CONTEÚDO DA MEMÓRIA DO ACUMULADOR.
ADD A,(Ix+disp)
ADD A,(Iy+disp)



Esta instrução irá efetuar a soma entre o acumulador e a posição de memória, endereçada pelo par de registrador HL, ou pelo registrador Ix+disp ou Iy+disp, ficando o resultado retido no acumulador.

A ilustração a seguir nos mostra a execução da instrução ADD A,(Ix + disp):

$$\underbrace{\text{ADD A, (Ix}}_{\text{DD}} + \underbrace{\text{disp)}}_{\text{86}} \qquad \underbrace{\quad}_{\text{d}}$$

Suponhamos: ppqq = 4000_{16}
xx = $1A_{16}$
e a localização de memória $400F_{16}$ contém 50_{16}.

Após a instrução ADD A,(Ix + 0FH) ter sido executada, o conteúdo do acumulador será $6A_{16}$. Veja:

```
1A = 0001 1010
50 = 0101 0000  +
     0110 1010
```

0, ajusta S para 0.
Não carry, C é ajustado para 0.
Resultado não-zero, Z é ajustado para 0.
Não carry, AC é ajustado para 0.
Instrução de adição, N é ajustado para 0.
0 ⊻ 0 = 0, P/V é ajustado para 0.

$$\underbrace{\text{ADD A, (Iy}}_{\text{FD}} + \underbrace{\text{disp)}}_{\text{86}} \qquad \underbrace{\quad}_{\text{d}}$$

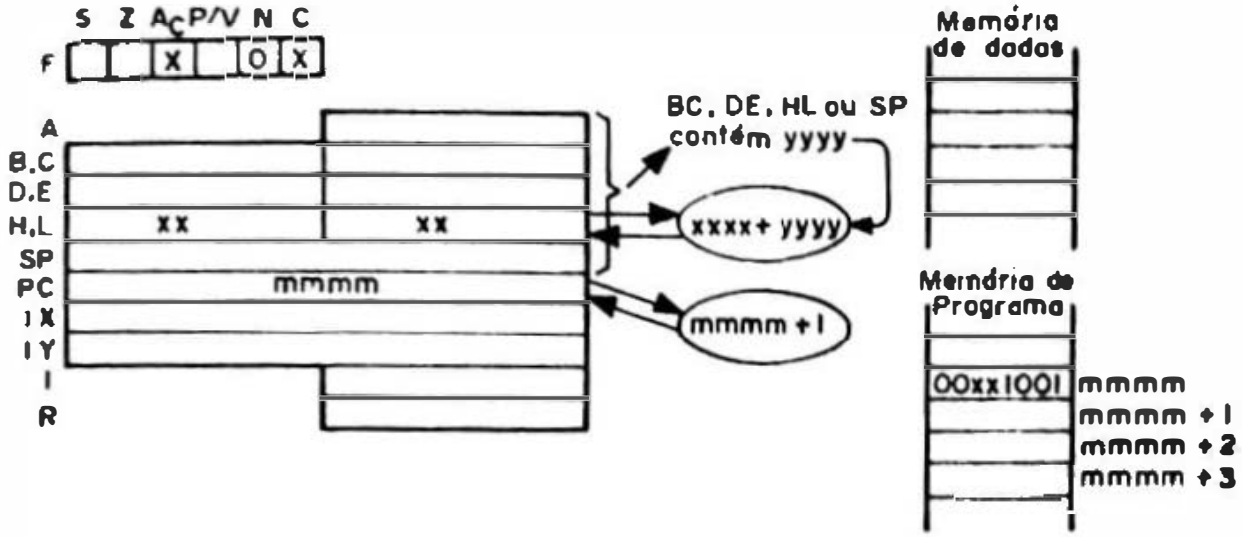
Esta instrução é idêntica à ADD A,(Ix + disp), exceto pelo fato dela usar o registro Iy em vez do registro Ix.

$$\underbrace{\text{ADD A,(HL)}}_{\text{86}}$$

Esta outra versão da instrução soma o conteúdo da localização de memória endereçada pelo par de registradores HL, com o conteúdo do acumulador, sendo que o resultado ficará retido no acumulador.

A instrução ADD é uma instrução de manipulação de dados da rotina.

8) ADD HL,rp - SOME O CONTEÚDO DO PAR DE REGISTRADORES "rp" COM O CONTEÚDO DO PAR DE REGISTRADORES HL.



ADD HL,rp

00 xx 1001

00 para rp = BC
01 para rp = DE
10 para rp = HL
11 para rp = Stack-Pointer

Esta instrução é constituída por dois bytes, representando ambos o código de operação. Ao se efetuar esta instrução, o conteúdo do par de registradores "HL" é somado com o conteúdo do par de registradores especificados (rp), mais o carry bit e o resultado deverá ser carregado no par de registradores HL.

Suponhamos que HL contenha $034A_{16}$ e DE contenha $214C_{16}$. Após a execução da instrução ADD HL,DE, o par de registradores HL terá seu conteúdo igual a 2496_{16}. Veja:

```
034A = 0000 0011 0100 1010
214C = 0010 0001 0100 1100   +
       0010 0100 1001 0110
```

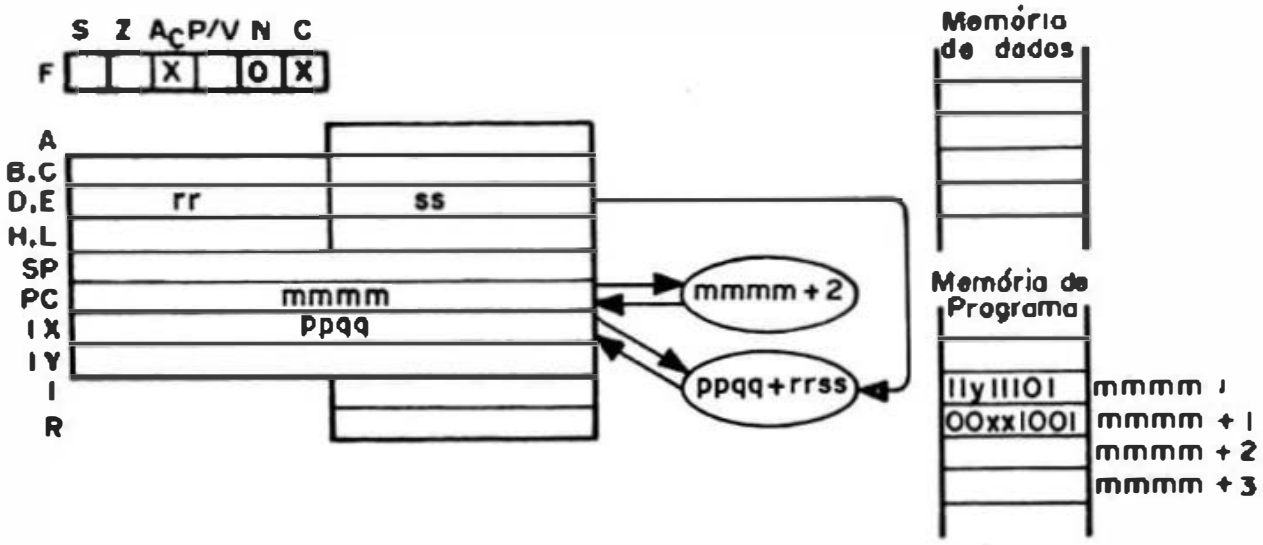
Não carry, C é ajustado para 0

Não-carry, AC é ajustado para 0.

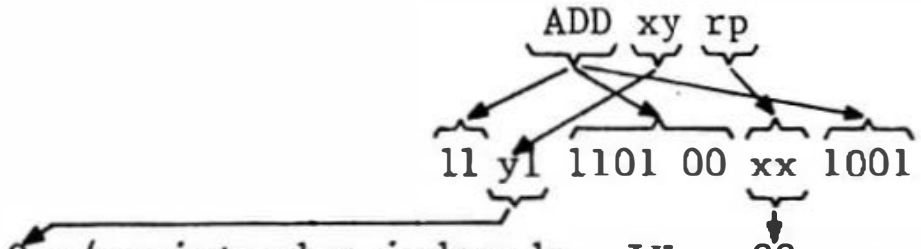
Instrução de adição, N é ajustado para 0

A instrução ADD HL,HL é equivalente a um deslocamento (shift) de 16 bits para a esquerda.

9) ADD xy,rp - SOME O CONTEÚDO DO PAR DE REGISTRADORES "rp", COM O CONTEÚDO DOS REGISTRADORES ÍNDICE "xy" (OU INDEXADOS "xy").



A ilustração a seguir nos mostra a execução da instrução ADD Ix,DE.



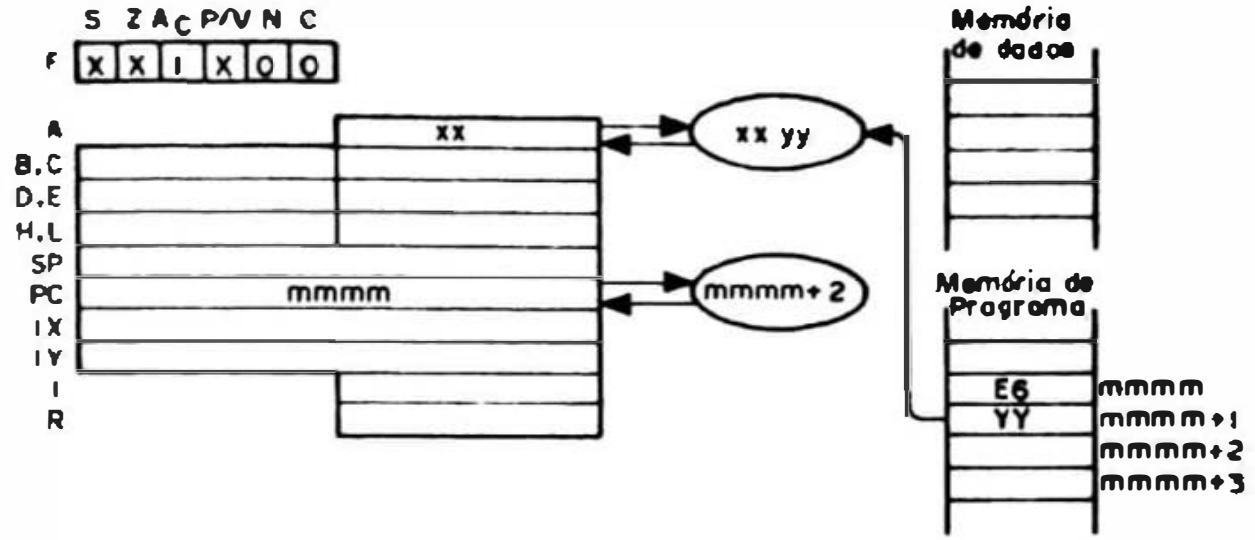
0 p/registrador indexado = IX
1 p/registrador indexado = IY

00 para rp = BC
01 para rp = DE
10 para rp = registrador índice especificado
11 para rp = Stack-Pointer

Some o conteúdo do par de registradores especificado "rp", ao conteúdo do registrador indexado (ou índice) especificado, carregando o resultado no registrador indexado.

Suponhamos que Iy contenha $4FF0_{16}$ e BC contenha $000F_{16}$. Após a execução da instrução ADD Iy,BC, o conteúdo do registrador indexado passará a ser igual a $4FFF_{16}$.

10) AND data - EXECUTE A LÓGICA E (AND) ENTRE O ACUMULADOR E O BYTE DE DADOS.



AND data

E6 yy

Esta instrução executa a lógica AND entre o conteúdo do acumulador e do próximo byte de memória do computador, carregando o resultado no acumulador.

Suponhamos que xx = $3A_{16}$. Após a execução da função AND 7CH, o conteúdo do acumulador passará a ser igual a 38_{16}. Veja:

```
3A = 0011 1010
7C = 0111 1100   +
     0011 1000
```

0, ajusta S para 0 — 3 bits 1 — Três bits 1, P/V é ajustado para 0. Resultado não-zero, Z é ajustado para 0.

Esta é uma instrução lógica de rotina, freqüentemente utilizada para "desligar" (ou zerar) os bits. Por exemplo, a instrução AND 7FH, irá, incondicionalmente, ajustar (set), o bit de alta ordem do acumulador, para 0.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 · CEP 86100 · LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-27

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Os registradores F e F':

a) são selecionados da mesma forma que o registrador A;

b) nunca são usados para manipular dados;

c) os itens a e b estão corretos;

d) N.R.A.

2) Os registradores indexados (ou índice) Ix e Iy possuem:

a) 16 bits cada um;

b) 8 bits cada um;

c) 12 bits cada um;

d) N.R.A.

3) O contador de programa (PC) tem por função determinar:

a) o número de instruções do programa;

(b) a ordem de execução das instruções;

c) os itens a e b estão corretos;

d) N.R.A.

4) Para carregar constantes em registros de 16 bits, é muito utilizado o modo de endereçamento:

a) Imediato;

b) Página zero;

c) Relativo;

d) Imediato extendido.

5) O modo de endereçamento mais completo do microprocessador Z-80 é o modo:

a) Página Zero;

b) Extendido Direto;

c) Indexado;

d) Relativo.

6) O segundo tipo de operando do microprocessador Z-80, normalmente:

a) afeta um ou mais endereços de memória do microcomputador;

b) afeta somente um endereço de memória do microcomputador;

c) afeta os registradores principais, os indexadores, o Stack-Pointer e o Contador de Programa;

d) N.R.A.

7) A abreviação indicada pelo sinal ⊻ indica uma operação lógica:

a) OU;

b) OU-Exclusiva;

c) E;

d) N.R.A.

8) A instrução ADD A,data soma o conteúdo do próximo byte da memória de programa com o conteúdo do:

a) registrador Ix;

b) registrador Iy;

c) par de registradores BC;

d) acumulador.

9) A instrução "ADD A,reg" soma o conteúdo do registrador especificado (reg) com o conteúdo:

a) do acumulador;

b) da localização de memória endereçada por HL;

c) do Stack-Pointer;

d) N.R.A.

10) A instrução "AND data" executa a lógica E (AND) entre o byte de dados e o conteúdo:

a) do registrador D;

b) do registrador de índice Ix;

c) do acumulador;

d) N.R.A.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

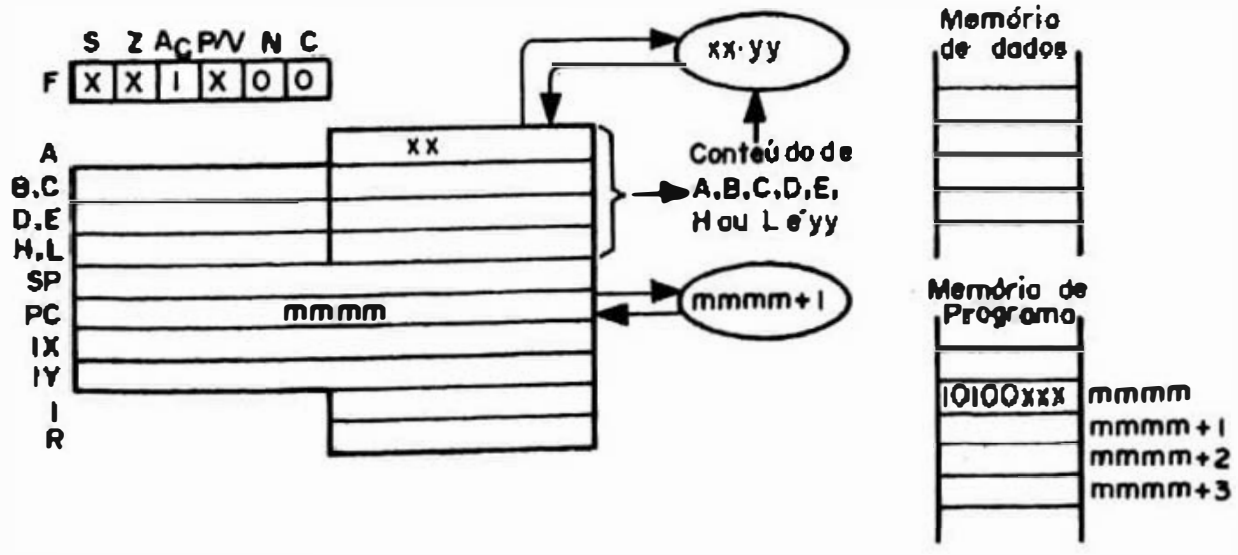
CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

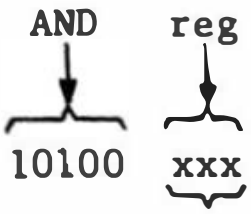
LIÇÃO MC-28

PROGRAMAÇÃO ASSEMBLER III

Dando continuidade à lição anterior, estudaremos agora mais algumas instruções do microprocessador Z-80.

11) AND reg - EXECUTE A LÓGICA E (AND) ENTRE O ACUMULADOR E OS REGISTRADORES.

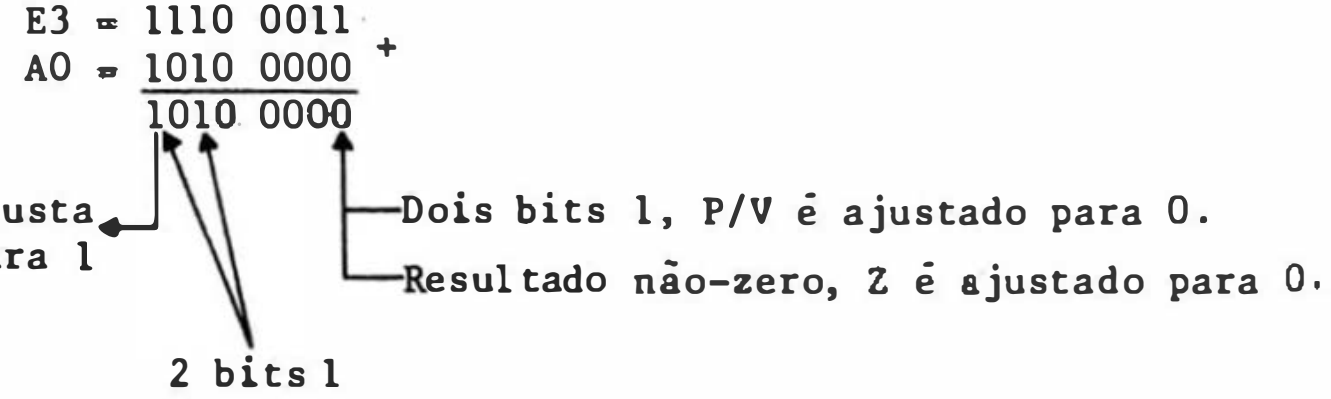




000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Esta instrução é constituída por um único byte, sendo este o código de operação. A presente instrução executa a lógica E (AND), entre o conteúdo do acumulador e o conteúdo de um dos registradores descritos, sendo o resultado carregado no acumulador.

Suponha xx = $E3_{16}$ e o conteúdo do registrador E = $A0_{16}$. Após a execução da instrução AND E, o conteúdo do acumulador será $A0_{16}$. Veja:



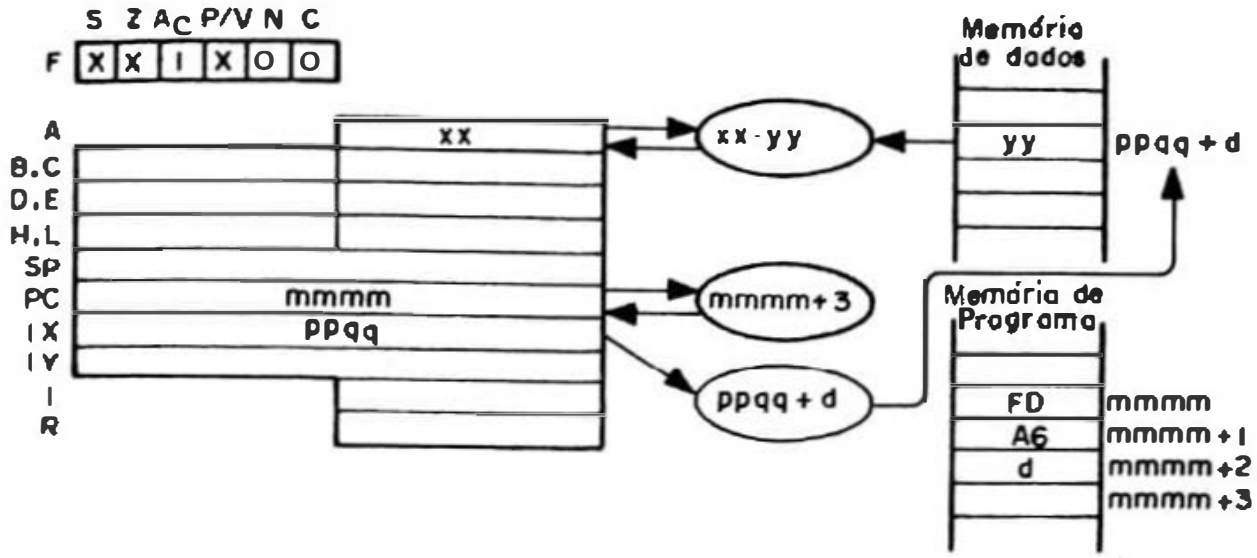
A instrução AND é uma instrução lógica freqüentemente usada.

12) AND (HL) - EXECUTE A LÓGICA E (AND) ENTRE O ACU-
AND (Ix+disp) MULADOR E O CONTEÚDO DA POSIÇÃO DE ME
AND (Iy+disp) MÓRIA ENDEREÇADA.

A ilustração a seguir nos mostra a execução da instrução AND (Iy + disp), onde:

AND (Iy + disp)

FD A6 d



Esta instrução é formada por três (3) bytes, sendo os dois primeiros o código de operação e o terceiro um byte de dados, o qual deverá ser somado ou subtraído do registrador de índice (Ix ou Iy), apontando assim a posição (ou endereço) do dado a ser manuseado. Esta instrução executa a lógica AND entre o conteúdo do acumulador e o conteúdo da localização de memória indicada pelo registrador de índice Iy, somado com o respectivo dígito de deslocamento. O resultado da operação deverá ser carregado no acumulador.

Suponhamos que: xx = 43, ppqq = 4000_{16} e que a localização de memória $400F_{16}$ contenha o valor $A0_{16}$. Após a execução da instrução AND (Iy + 0FH), o conteúdo do acumulador passa a ser igual a $A0_{16}$. Veja:

E3 = 1110 0011
A0 = 1010 0000 +
1010 0000

Dois bits 1, P/V é ajustado para 1 (paridade par).

Resultado não-zero, Z é ajustado para 0.

1, S é ajustado para 1.

$$\underbrace{\text{AND (Ix}}_{\text{DD A6}} + \underbrace{\text{disp)}}_{\text{d}}$$

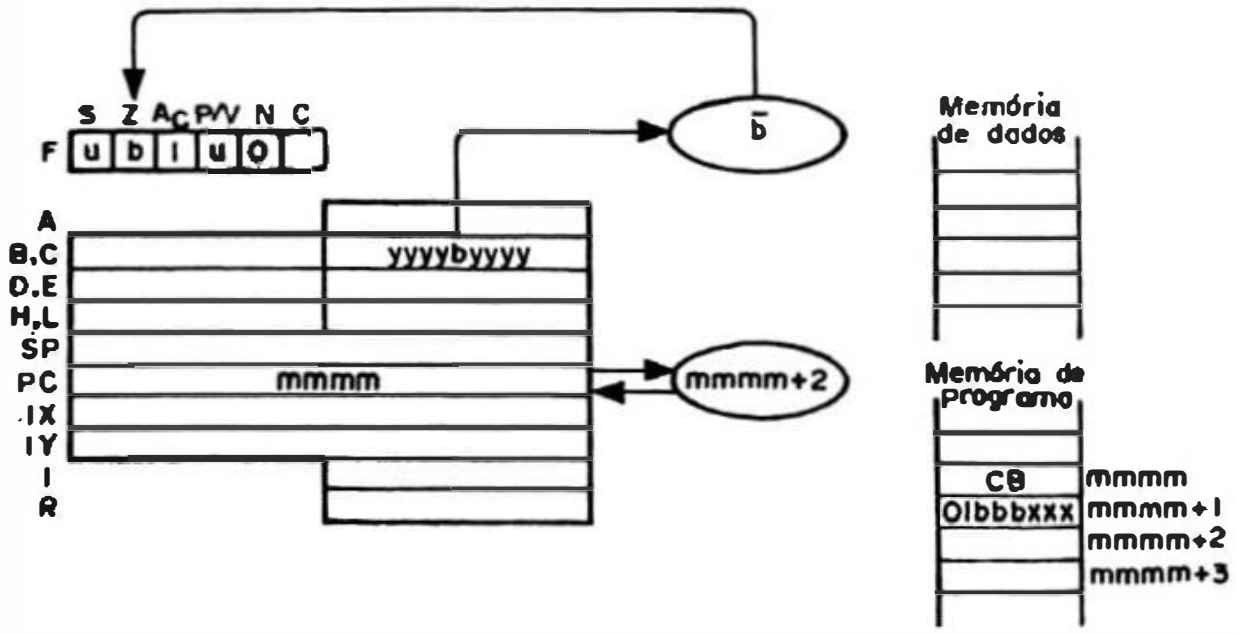
A instrução apresentada acima é idêntica à instrução AND (Iy + disp), exceto pelo fato desta utilizar o registrador Ix ao invés do registrador Iy.

$$\underbrace{\text{AND (HL)}}_{\text{A6}}$$

Por outro lado, a instrução AND (HL) é uma instrução de um só byte, sendo este o código de operação. Esta instrução executa a lógica AND entre os conteúdos do acumulador e da localização de memória (especificado pelo conteúdo do par de registradores HL).

A instrução AND é uma instrução lógica freqüentemente utilizada.

13) BIT b,reg - TESTE O BIT b NO REGISTRADOR reg.

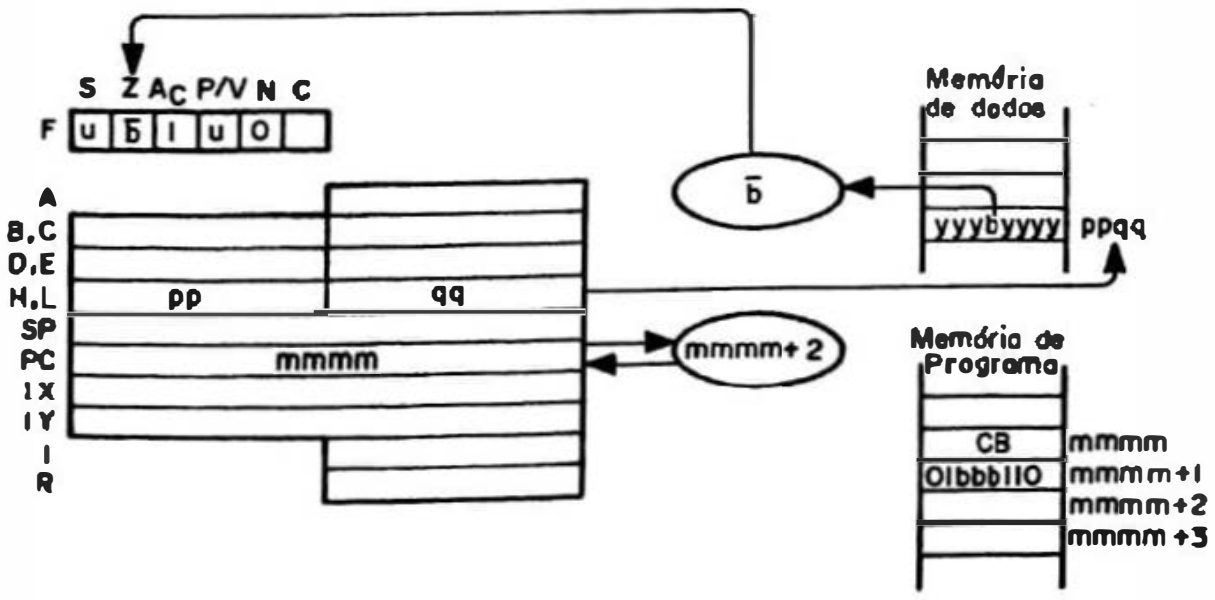


BIT	b,	reg	
CB 01	bbb	xxx	
Bit Testado			Registrador
0	000	000	B
1	001	001	C
2	010	010	D
3	011	011	E
4	100	100	H
5	101	101	L
6	110	111	A
7	111		

Esta instrução é formada por dois bytes sendo estes o código de operação. Basicamente, esta instrução testa um único bit de um registrador, sendo o flag "Z" o responsável por este teste. O teste consiste em colocar o complemento do bit especificado do registrador indicado no flag "Z" do registrador F.

Suponhamos que o registro C contenha o valor 11101111. A instrução BIT 4,C irá então ajustar (set) o flag "Z" para 1 enquanto o bit 4 do registrador C permanecer 0 (zero). O bit 0 (zero) é o bit menos significativo.

14) BIT b,(HL) - TESTE O BIT B DA POSIÇÃO DE MEMÓRIA INDICADA.
BIT b,(Ix+disp)
BIT b,(Iy+disp)



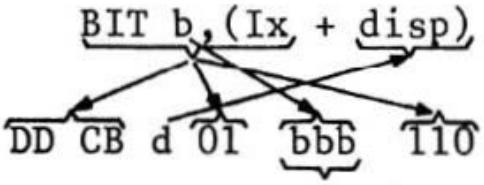
A ilustração nos mostra a execução da instrução BIT 4, (HL). O bit 0 (zero) é o bit menos significativo.

BIT	b,	(HL)
CB 01	bbb	110

Bit Testado	bbb
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

A instrução BIT b,(HL) é formada por dois bytes, sendo os dois o código de operação. Esta instrução testa o bit indicador de memória endereçada pelo conteúdo do par de registradores HL e coloca o complemento do bit no Flag Z do registrador F.

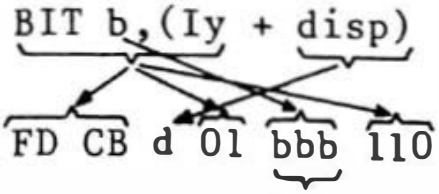
Suponhamos que o par de registradores HL contenha 4000H e, que o bit 3 da localização de memória 4000H seja igual a 1. Assim, sendo, a instrução BIT 3,(HL) irá ajustar o Flag Z para 0 (zero), enquanto que o bit 3 na localização (ou endereço) de memória 4000H permanece 1 (um).



bbb é o mesmo que na instrução BIT b,(HL).

A instrução BIT b, (Ix + disp) é constituída por quatro bytes, sendo que os dois primeiros e o quarto formam o código de operação. O terceiro byte é somado ou subtraído do registrador índice (ou indexado). Esta instrução examina o bit especificado dentro da localização de memória indicada pela soma de (Ix + disp) e coloca o complemento do bit no Flag Z do registro F.

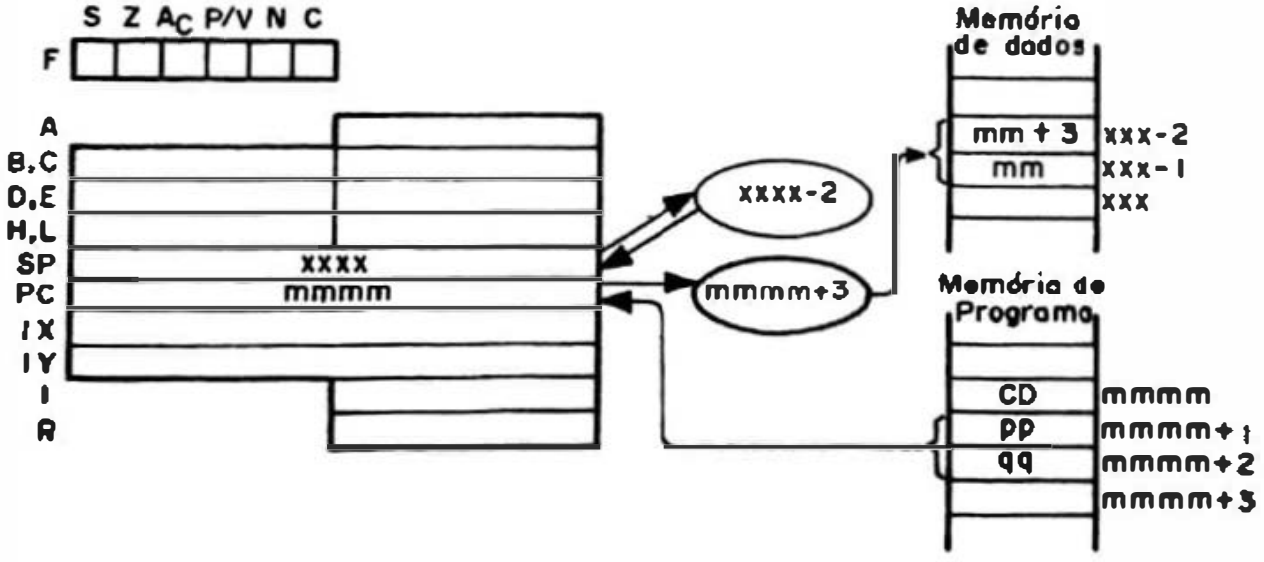
Suponhamos, para o caso acima descrito, que o conteúdo de Ix seja igual a 4000H e o bit 4 da localização de memória 4004H seja igual a 0 (zero). A instrução BIT 4,(Ix + 4H) irá então, ajustar o Flag Z a 1, enquanto o bit 4 da localização de memória 4004H permanece 0.



bbb é o mesmo que na instrução BIT b,(HL).

A instrução acima, BIT b,(Iy + disp), é idêntica à instrução BIT b,(Ix + disp), exceto pelo fato desta utilizar o registrador Iy em vez do registrador Ix.

15) CALL label - CHAME A SUB-ROTINA IDENTIFICADA NO OPERANDO.



Esta instrução é formada por três bytes, sendo o primeiro o código de operação, e os dois últimos o endereço que irá modificar o contador de programa (PC). Esta instrução armazena o endereço da instrução que segue o CALL no topo do Stack-Pointer (SP). Este endereço armazenado no Stack-Pointer vem a ser o endereço de retorno ao qual o programa voltará após a execução de uma certa rotina.

O segundo byte da instrução CALL é a metade de baixa ordem do endereço, e o terceiro byte é o byte de alta ordem.

Como exemplo, considere a seqüência de instrução:

```
     CALL     SUBR
     AND       7CH
      =
      =
      =
SUBR
```

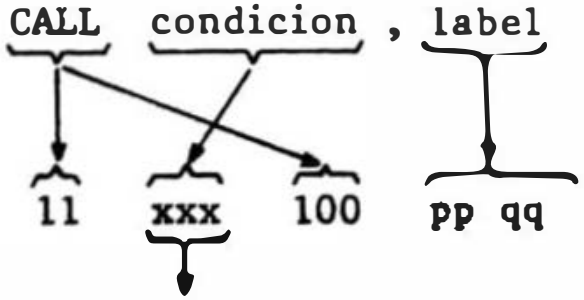
Após a execução da instrução, o endereço da instrução AND é armazenado no topo da pilha (SP). O Stack-Pointer é decrementado de 2 (dois), de modo a endereçar o novo conteúdo do topo da pilha. Em seguida a instrução intitulada SUBR será executada.

Este tipo de instrução é conhecido como instrução incondicional, pois não depende de nenhuma condição especial para ser executada.

16) CALL condition, label - CHAME A SUB-ROTINA IDENTIFICADA NO OPERANDO SE A CONDIÇÃO FOR SATISFEITA.

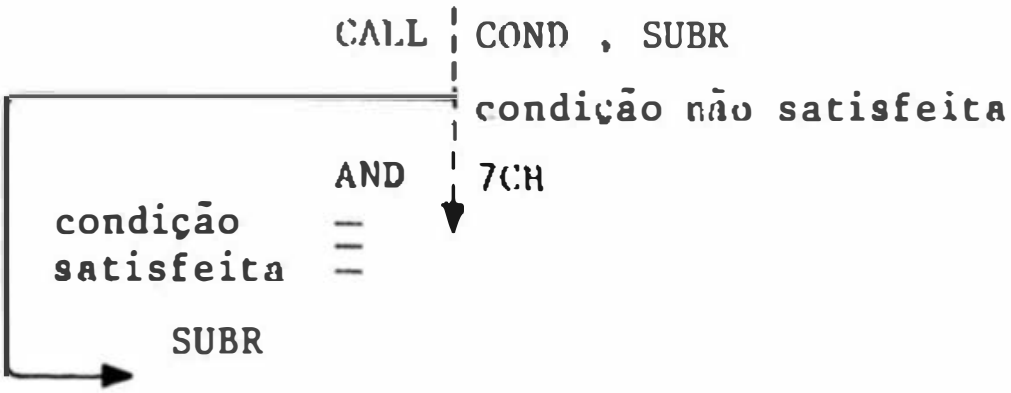
Esta instrução é idêntica à instrução CALL, exceto pelo fato de que a sub-rotina identificada somente será chamada se a condição for satisfeita. Caso contrário, a instrução que segue seqüencialmente à instrução CALL será executada.

Este tipo de instrução é chamada de instrução condicional, pois depende de alguma condição para ser executada. Se as condições dos Flags forem favoráveis, esta instrução deverá carregar o conteúdo do contador de programa (PC) na posição de memória endereçada pelo Stack-Pointer (SP). Esta operação irá indicar o endereço de retorno ao qual o programa voltará após a execução da respectiva sub-rotina. Em seguida, o PC será carregado com o endereço da sub-rotina, se as condições dos Flags forem favoráveis. Caso contrário, a instrução será ignorada.



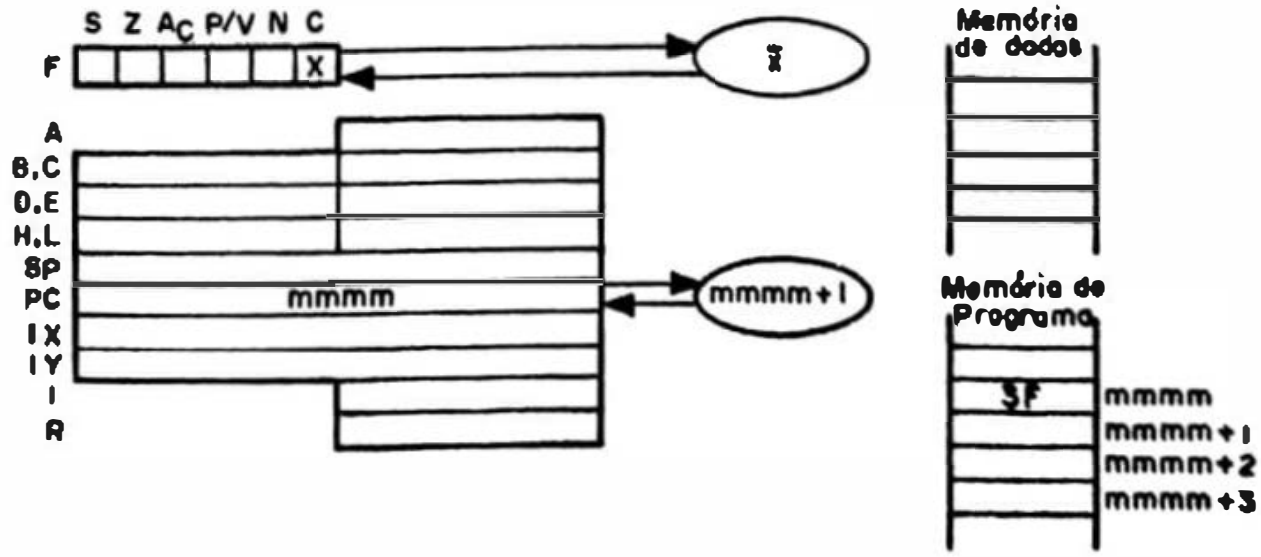
	Condição		Flag
000	NZ	Não-zero	Z
001	Z	Zero	Z
010	NC	Não-carry	C
011	C	Carry	C
100	PO	Paridade ímpar	P/V
101	PE	Paridade par	P/V
110	P	Sinal positivo	S
111	M	Sinal negativo	S

Como exemplo, considere a seqüência de instruções:



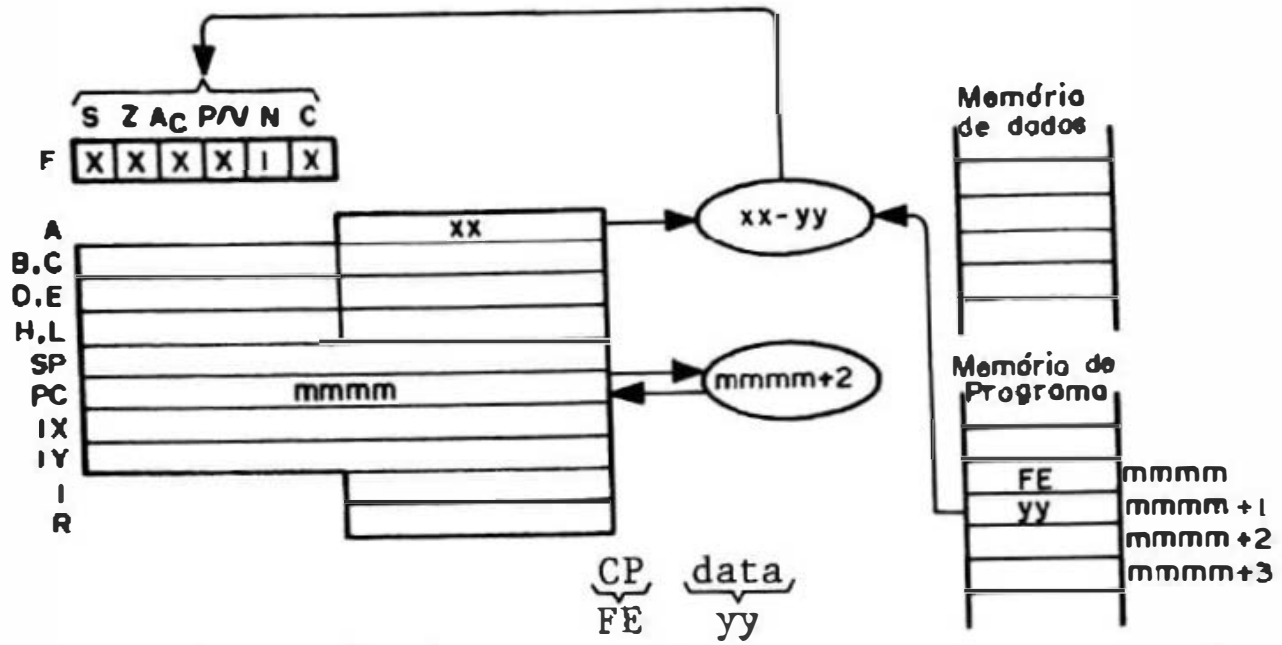
Se a instrução não for satisfeita, a instrução AND será executada depois da instrução CALL condition, SUBR. Caso a condição seja satisfeita, o endereço da instrução AND será carregado no topo da pilha, e o SP será então decrementado por 2. A instrução intitulada SUBR será executada logo após.

17) CCF - COMPLEMENTO DO FLAG CARRY



Formada por um único byte e, sendo este o código operação, esta instrução complementa o carry bit, sem afetar nenhum outro registrador ou conteúdo de posição.

18) CP data - COMPARE DADOS DIRETAMENTE COM O CONTEÚDO DO ACUMULADOR.



Esta instrução é constituída por dois bytes, sendo o primeiro o código de operação e o segundo o byte de dados. Basicamente, esta instrução efetua a subtração entre o conteúdo do acumulador e o conteúdo do segundo byte do código objeto, tratando ambos os números como simples dados binários, afetando apenas os Flags. O conteúdo do acumulador e dos outros registradores não é alterado, afetando-se apenas o registrador F.

Suponhamos que xx = $E3_{16}$ e que o segundo byte do código objeto da instrução CP contenha $A0_{16}$. Após a execução da instrução CP 0AOH, o conteúdo do acumulador continuará sendo $E3_{16}$, porém as posições do registrador F serão modificadas, como segue:

```
E3 = 1110 0011
A0 = 1010 0000  +
     0100 0011
```

0, ajusta S para 0

Não-borrow, C é ajustado para 0 (zero).

Resultado não-zero, Z é ajustado para zero (0).

Não-borrow, AC é ajustado para 0.

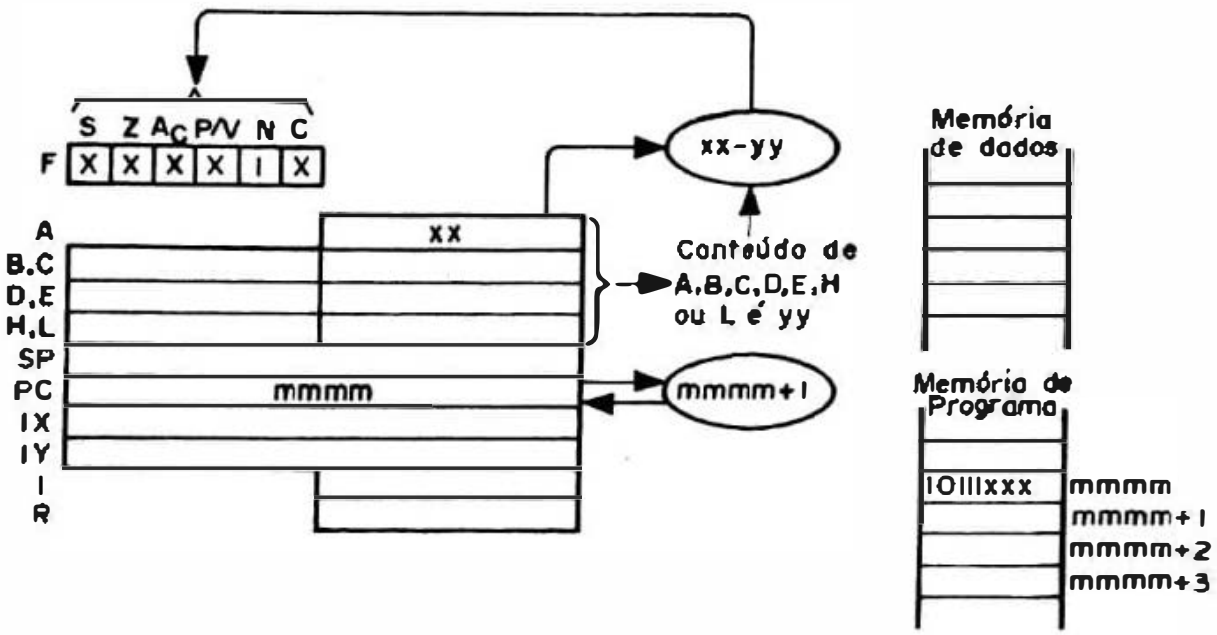
Instrução de subtração, N é ajustado para 1.

1 + 1 = 0, P/V é ajustado para 0

Observe que o carry bit resultante é complementado.

Esta instrução tem por finalidade comparar dois bytes.

19) CP reg - COMPARE O CONTEÚDO DO REGISTRADOR COM O ACUMULADOR.



CP reg

10111 xxx

000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Constituída por apenas um byte e, sendo este o próprio código de operação, esta instrução efetua a subtração entre o conteúdo do acumulador e o conteúdo do registrador (reg) especificado, afetando apenas os Flags. O acumulador permanece como está.

Suponhamos xx = $E3_{16}$ e o conteúdo do registrador B igual a $A0_{16}$. Após a execução da instrução CP B, o conteúdo do acumulador continuará a ser igual a $E3_{16}$, porém as posições do Flag serão modificadas como segue:

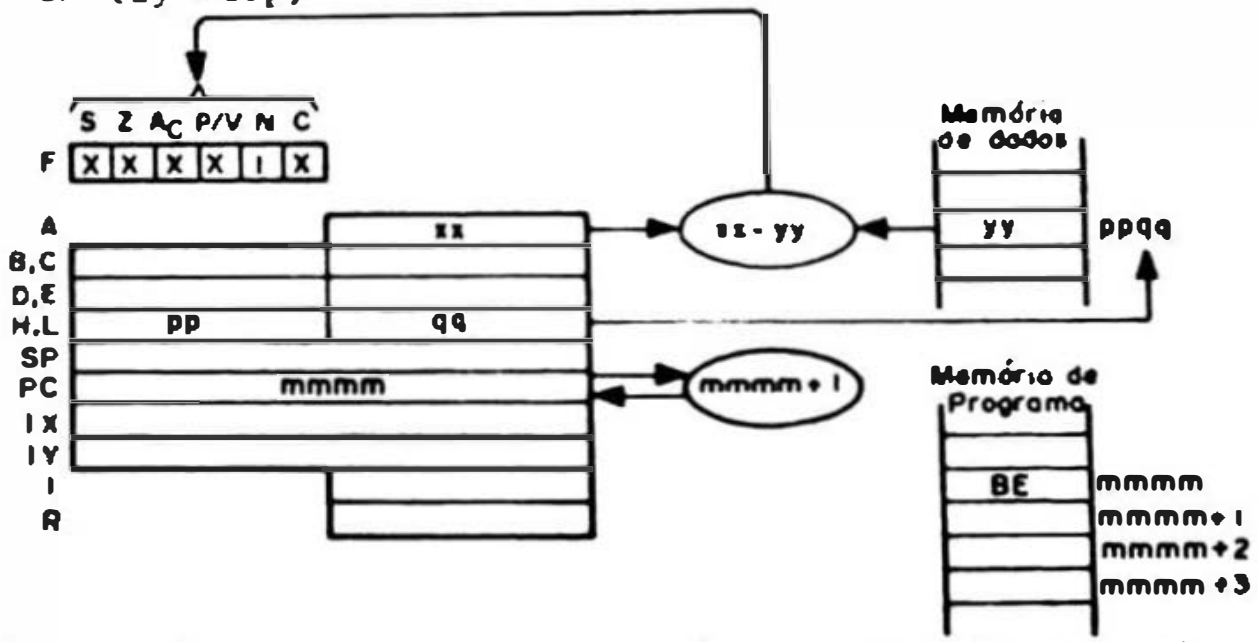
E3 = 1110 0011
A0 = 1010 0000
0100 0011

Resultado não-zero, Z é ajustado para 0.
Não-borrow, AC é ajustado para 0.
Instrução de subtração, N é ajustado para 1.
1 ⊻ 1 = 0, P/V é ajustado para 0.

0, S é ajustado para 0
Não-borrow, C é ajustado para 0.

Observe que o carry bit resultante é complementado.

20) CP (HL) - COMPARE O CONTEÚDO DA MEMÓRIA COM O
CP (Ix+disp) DO ACUMULADOR.
CP (Iy+disp)

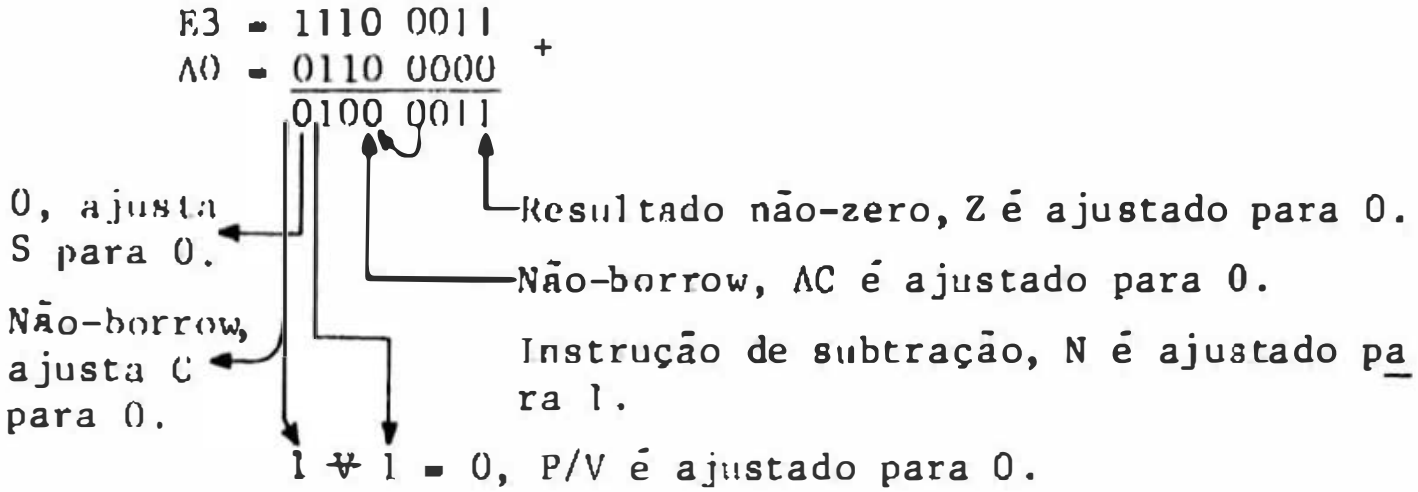


A ilustração nos mostra a execução da instrução CP (HL).

$$\underbrace{\text{CP (HL)}}_{\text{BE}}$$

A instrução CP (HL) é formada por um único byte, sendo este o código de operação. Esta instrução efetua a subtração entre o conteúdo da localização de memória (especificado pelo conteúdo do par de registradores HL) e o conteúdo do acumulador, tratando ambos os números (conteúdos) como simples dados binários, afetando apenas os Flags. O conteúdo do acumulador e do par de registradores HL não é alterado.

Suponhamos xx = $E3_{16}$ e yy = $A0_{16}$. Após a execução da instrução CP (HL), o conteúdo do acumulador continuará sendo $E3_{16}$, porém as posições do Flag serão modificadas como segue:



Observe que o carry bit resultante é complementado.

Vejamos agora a instrução CP (Ix + disp):

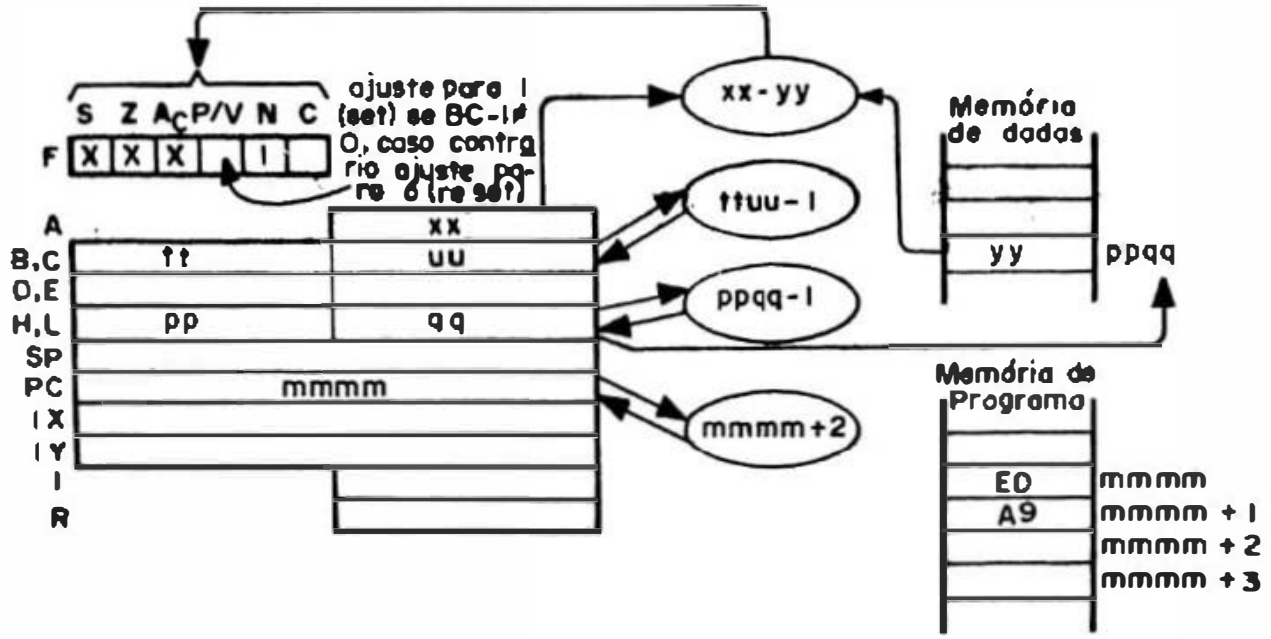
$$\underbrace{CP\ (Ix}_{DD\ BE} + \underbrace{disp)}_{d}$$

Esta instrução é constituída por três bytes, sendo os dois primeiros o código de operação e o terceiro um byte de dados. Analogamente à instrução anterior, esta instrução efetua a subtração entre o conteúdo da localização de memória (especificada pela soma do conteúdo do registro Ix e o valor do deslocamento d) e o conteúdo do acumulador, tratando ambos os números como simples dados binários, afetando apenas os Flags. O conteúdo do acumulador e do registrador Ix, em conjunto com o byte d, não e alterado.

$$\underbrace{CP\ (Iy}_{FD\ BE} + \underbrace{disp)}_{d}$$

A instrução CP (Iy + disp) é idêntica à instrução CP (Ix + disp), exceto pelo fato da primeira usar o registrador Iy em vez do registrador Ix.

21) CPD - COMPARE O CONTEÚDO DO ACUMULADOR COM A MEMÓRIA. DECREMENTE O CONTADOR DE ENDEREÇO E DE BYTES.



CPD

ED A9

Esta instrução é constituída de dois bytes, sendo estes o código de operação. Esta instrução compara o conteúdo do acumulador com o conteúdo da localização de memória endereçada pelo par de registradores HL, afetando apenas os Flags. Se o conteúdo do acumulador for igual ao da memória, ajuste o flag Z para 1 (set). Após a operação, o par de registradores HL e BC deve ser decrementado, de modo que HL passe a indicar a próxima localização de memória a ser comparada e BC passe a indicar a quantidade de bytes a ser comparada, uma vez que este último é usado como contador de byte.

Suponhamos que xx = $E3_{16}$, ppqq = 4000_{16}, BC contenha 0032_{16} e que yy = $E3_{16}$. Após a execução da instruçao CPD, o acumulador continuará contendo $E3_{16}$, porém as posições do Flag serão modificadas como segue:

```
E3 = 1110 0011
A0 = 1010 0000 +
     0100 0011
```

0, ajusta S para 0

Resultado não-zero, Z é ajustado para 0.

Não-borrow, AC é ajustado para 0.

O flag P/V será "zerado" (reset) porque BC-1 = 0
Instrução de subtração, N é ajustado para 1.
Carry bit não é afetado.

o par de registradores HL conterá $3FFF_{16}$, e BC = 0.

22) CPDR - COMPARE O ACUMULADOR COM A MEMÓRIA. DECREMENTE O CONTADOR DE ENDEREÇO E DE BYTE. PROSSIGA ATÉ QUE A COMPARAÇÃO SEJA ENCONTRADA OU O CONTADOR DE BYTE SEJA ZERO.

CPDR
ED B9

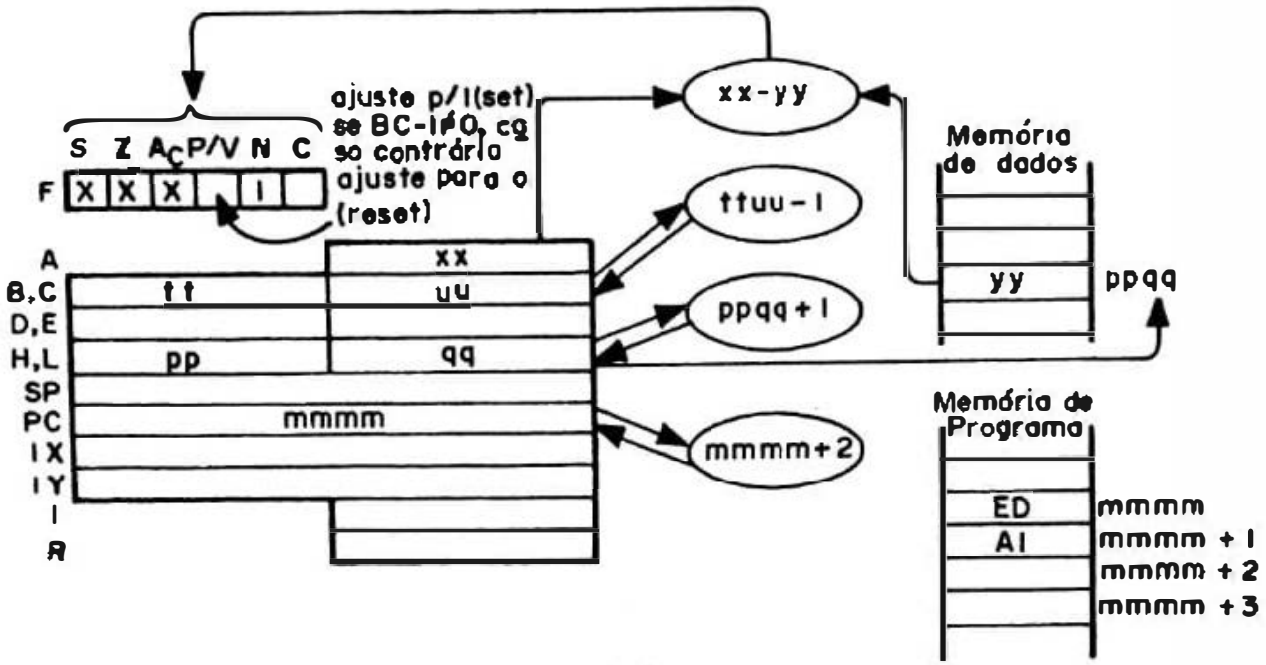
Esta instrução é idêntica à anterior (CPD), exceto que ela é repetitiva, saindo do ciclo de repetições somente quando o conteúdo do acumulador for igual à posição de memória, endereçada pelo par de registradores HL, ou quando o contador de byte BC tiver seu conteúdo igual a zero (0). Depois de cada transferência de dados, as interrupções serão reconhecidas e os dois ciclos de refrescamento serão executados.

Suponhamos que o par de registradores HL contenha 5000_{16}, o par de registradores BC contenha $00FF_{16}$, o acumulador contenha $F9_{16}$ e o conteúdo da memória com sua respectiva localização seja:

Localização	Conteúdo
5000_{16}	AA_{16}
$4FFF_{16}$	BC_{16}
$4FFE_{16}$	19_{16}
$4FFD_{16}$	$7A_{16}$
$4FFC_{16}$	$F9_{16}$
$4FFB_{16}$	DD_{16}

Após a execução da instrução CPDR, o flag P/V será igual a 1, o flag Z também será igual a 1, o par de registradores HL irá conter $4FFB_{16}$, e o par de registradores BC irá conter $00FA_{16}$.

23) CPI - COMPARE O ACUMULADOR COM A MEMÓRIA. DECREMENTE O CONTADOR DE BYTE (BC). INCREMENTE O ENDEREÇO (HL).



CPI

ED A1

Formada por dois bytes, sendo estes o código de operação, esta instrução é semelhante à instrução CPD, sendo que a única diferença reside no conteúdo do par de registradores HL, o qual é incrementado ao invés de decrementado.

Assim como a instrução CPD, a instrução CPI compara o conteúdo do acumulador ao conteúdo da localização de memória, endereçada pelo par de registradores HL. O par de registradores HL será incrementado enquanto o par de registradores BC (usado como contador de byte) será decrementado.

Suponhamos xx = $E3_{16}$, ppqq = 4000_{16}, BC contenha 0032_{16} e yy = $E3_{16}$. Após a execução da instrução CPI, o acumulador continuará a conter $E3_{16}$, porém as posições do Flag serão modificadas como segue:

E3 = 1111 0011
E3 = 0000 1101 +
0000 0000

0, S é ajustado para 0

Resultado é 0, Z é ajustado para 1.

Não-borrow, AC é ajustado para 0.

O flag P/V será ajustado para 1 porque BC-1 ≠ 0.

Instrução de subtração, N é ajustado para 1.

Carry bit não é afetado.

O par de registradores HL irá conter 4001_{16}, e BC irá conter 0031_{16}.

24) CPIR - COMPARE ACUMULADOR COM A MEMÓRIA. DECREMENTE O CONTADOR DE BYTE. INCREMENTE O ENDEREÇO. PROSSIGA ATÉ QUE A COMPARAÇÃO SEJA ENCONTRADA OU O CONTADOR DE BYTE SEJA ZERO.

CPIR
ED B1

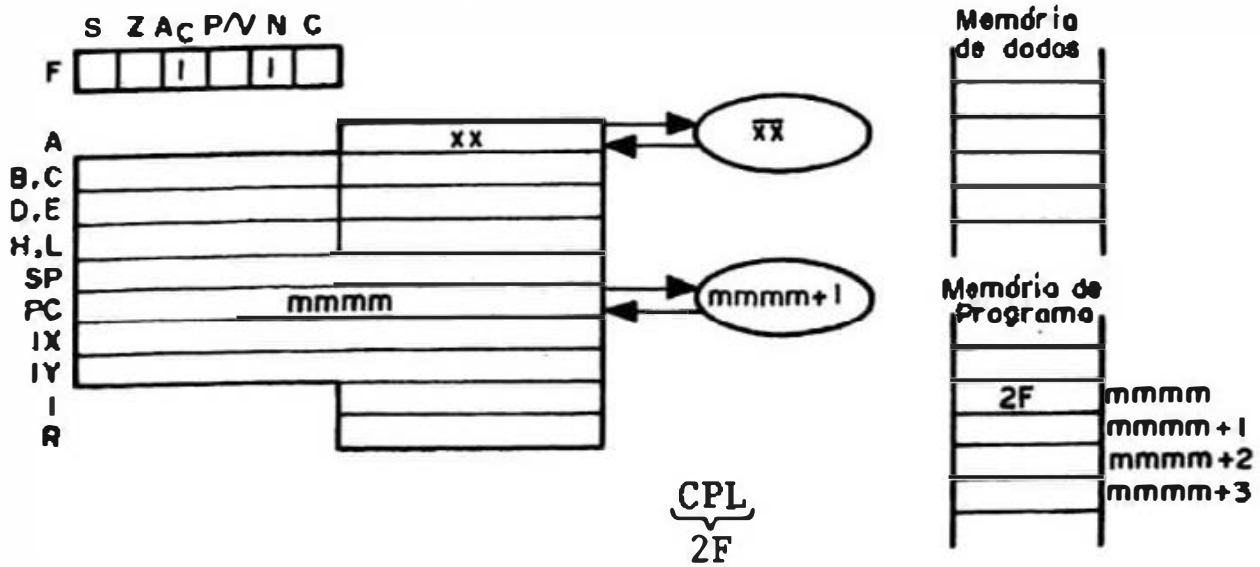
Constituída por 2 bytes, sendo estes o código de operação, esta instrução é praticamente idêntica à instrução CPI, exceto o fato dela ser repetitiva até que uma comparação seja encontrada ou o contador de byte (BC) seja zerado. Após cada transferência de dados, as interrupções serão reconhecidas e os dois ciclos de refrescamento serão executados. Ao contrário da instrução CPI, a instrução CPIR tem o conteúdo do par de registradores HL incrementado.

Suponhamos que o par de registradores HL contenha 4500_{16}, o par de registradores BC contenha $00FF_{16}$, o acumulador contenha $F9_{16}$ e, a memória tenha como conteúdo o seguinte:

Localização	Conteúdo
4500_{16}	AA_{16}
4501_{16}	15_{16}
4502_{16}	$F9_{16}$

Após a execução da instrução CPIR, o flag P/V será 1 e o flag Z também será igual a 1. O par de registradores HL, por sua vez, passará a conter 4503_{16} enquanto que o par BC passa a conter $00FC_{16}$.

25) CPL - COMPLEMENTE O ACUMULADOR.



CPL
2F

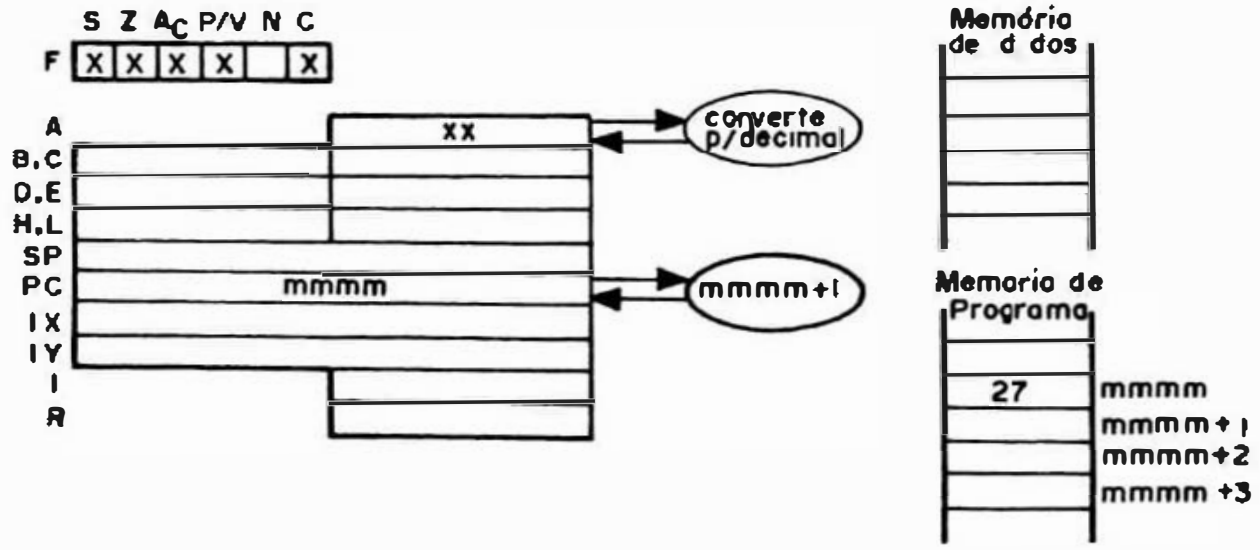
Constituída por apenas um byte e sendo este o código de operação, esta instrução complementa o conteúdo do acumulador, ou seja, inverte o conteúdo do acumulador "bit a bit". Nenhum outro registrador tem o seu conteúdo afetado.

Suponhamos que o acumulador contenha $3A_{16}$. Após a execução da instrução CPL, o acumulador passa a conter $C5_{16}$, veja:

3A = 0011 1010
complemento = 1100 0101

Esta instrução é uma instrução lógica de rotina. Não é preciso utilizá-la no caso de operações de subtração binária, pois existem instruções especiais para tal fim (SUB, SBC).

26) DAA - AJUSTE DECIMAL DO ACUMULADOR.



DAA
27

Constituída por apenas um byte e sendo este o código de operação, esta instrução converte o conteúdo do acumulador para a forma decimal, codificado em binário (BCD). Esta instrução deve ser utilizada somente depois de somar ou subtrair dois números BCD, isto é, ADD DAA, ADC DAA, INC DAA, SUB DAA, SBC DAA, DEC DAA ou NEG DAA, combinando instruções aritméticas decimais, as quais operam em fontes BCD para gerar respostas em BCD.

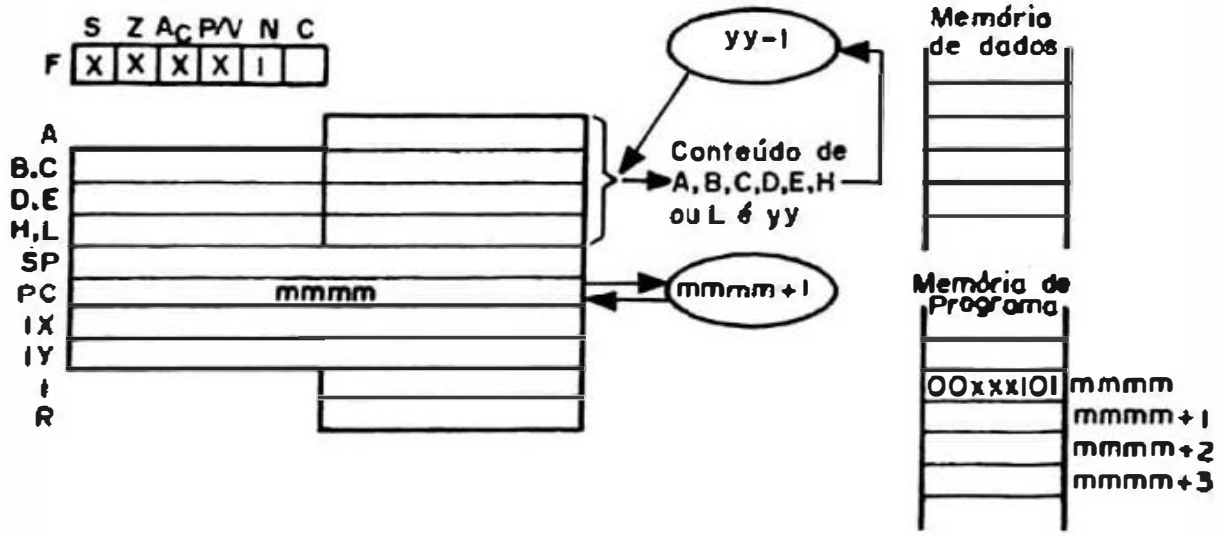
Suponhamos que o acumulador contenha 39_{16} e o registrador B contenha 47_{16}. Após a execução das instruções:

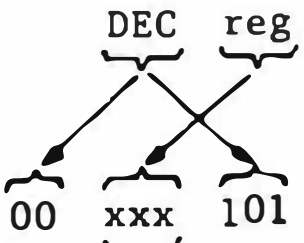
```
ADD B
DAA
```

o conteúdo do acumulador passa a ser 86_{16} e não 80_{16}.

A lógica da CPU do microprocessador Z-80 utiliza os valores não-carry e carry-auxiliar (AC), igualmente aos conteúdos do acumulador, na operação de "ajuste decimal".

27) DEC reg - DECREMENTE O CONTEÚDO DO REGISTRADOR.



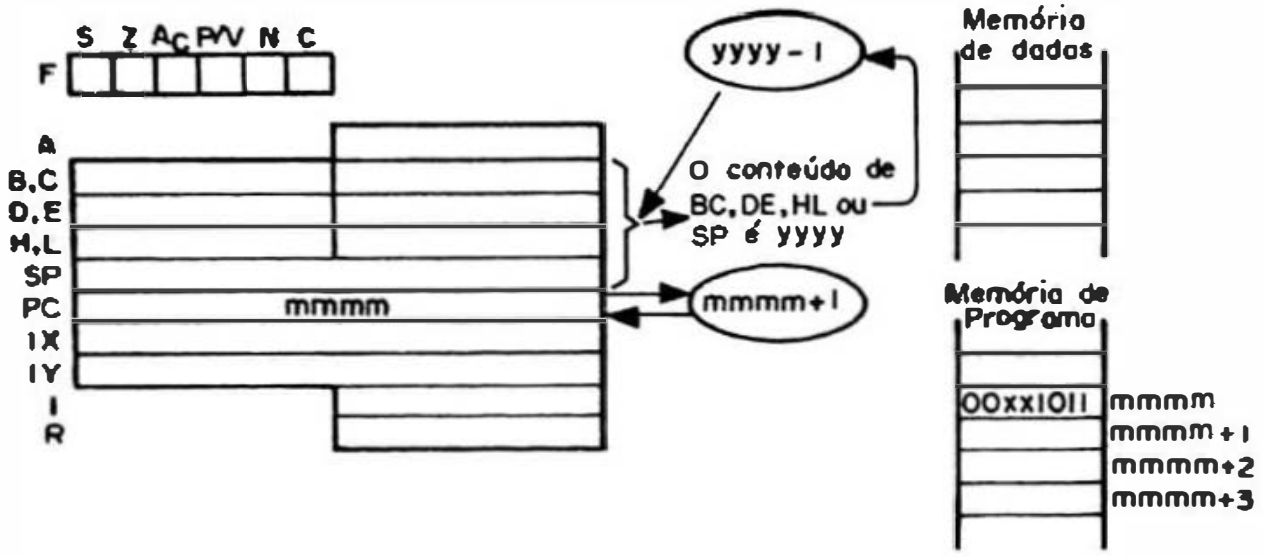


000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

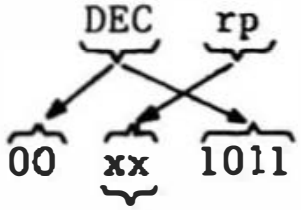
Constituída por apenas um byte e sendo este o código de operação, esta instrução decrementa uma unidade (1) do conteúdo do registrador especificado.

Suponhamos que o registrador A contenha 50_{16}. Após a execução da instrução DEC A, o conteúdo do registrador A passa a ser igual a $4F_{16}$.

28) DEC rp - DECREMENTE O CONTEÚDO DO PAR DE REGISTRADO-
DEC Ix RES ESPECIFICADOS.
DEC Iy



A ilustração nos mostra a execução da instrução DEC rp:



00 quando rp for o par de registrador BC
01 quando rp for o par de registrador DE
10 quando rp for o par de registrador HL
11 quando rp for o Stack-Pointer

A instrução DEC rp subtrai uma unidade (1) do valor de 16 bits contidos no par de registrador especificado. Nenhum flag de posição é afetado.

Suponhamos que o registrador H e L contenha $2F00_{16}$. Após a

execução da instrução DEC HL, o conteúdo dos registradores H e L passa a ser $2EFF_{16}$.

DEC Ix

DD 2B

A instrução DEC Ix é formada por dois bytes, sendo eles o código de operação. Esta instrução subtrai ou decrementa o conteúdo do registrador de índice Ix de uma unidade (1).

DEC Iy

FD 2B

A instrução DEC Iy também é constituída por dois bytes, sendo eles o próprio código de operação. Esta instrução decrementa de uma unidade (1) o conteúdo de 16 bits do registrador de índice Iy.

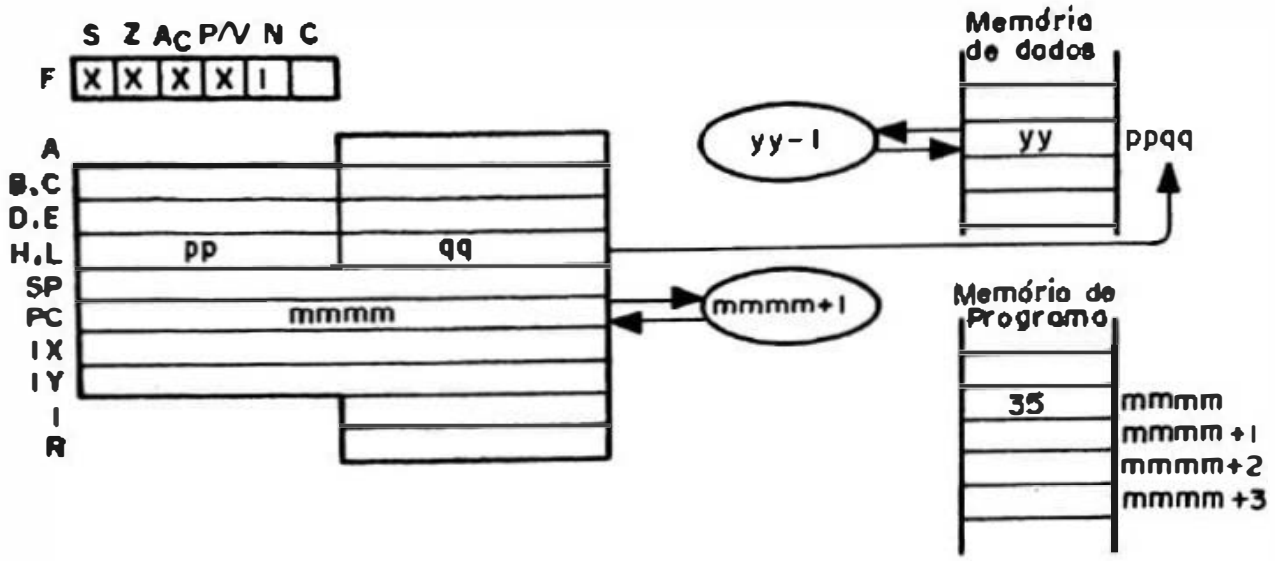
Nenhuma dessas instruções (DEC rp, DEC Ix e DEC Iy) afeta os flags de posição.

Enquanto a instrução DEC reg é utilizada nos loops interativos de instruções que usam um contador com um valor menor ou igual a 256, as instruções DEC rp, DEC Ix e DEC Iy devem ser utilizadas nos loops interativos de instruções queusamum contador com um valor superior a 256.

Devido ao fato da instrução DEC rp não ajustar nenhum flag de posição, outras instruções devem ser adicionadas para simplesmente testar um resultado zero. A seguir mostramos uma forma típica de loop.

```
        LD    DE,DATA  :  Carregue o valor inicial de 16 bits do
                          contador.
LOOP    —                 Primeira instrução do loop.
        —
        —
        DEC   DE       :  Decremente o contador.
        LD    A,D      :  Para testar para zero, mova D para A.
        OR    E        :  Então faça a lógica OU de A com E.
        JP    NZ,LOOP  :  Se não for zero, retorne.
                          (inicie o loop novamente).
```

29) DEC (HL) - DECREMENTE O CONTEÚDO DA MEMÓRIA.
DEC (Ix+disp)
DEC (Iy+disp)



A ilustração nos mostra a execução da instrução DEC (HL).

$$\underbrace{\text{DEC (HL)}}_{35}$$

Constituída por um único byte e sendo este o código de operação, a instrução DEC (HL) decrementa de uma unidade (1) o conteúdo da posição de memória endereçada pelo par de registradores (HL).

Suponhamos que ppqq = 4500_{16} e yy = $5F_{16}$. Após a execução da instrução DEC (HL), a localização de memória 4500_{16} passa a conter $5E_{16}$ (esse é o novo valor de yy). Veja:

```
 5F = 0101 1111
-01 = 1111 1111
      0101 1110
```

0, ajusta S para 0

Resultado não-zero, Z é ajustado para 0.

Não-borrow, AC é ajustado para 0.

1 ⊻ 1 = 0, P/V é ajustado para 0.

Instrução de subtração, N é ajustado para 1.

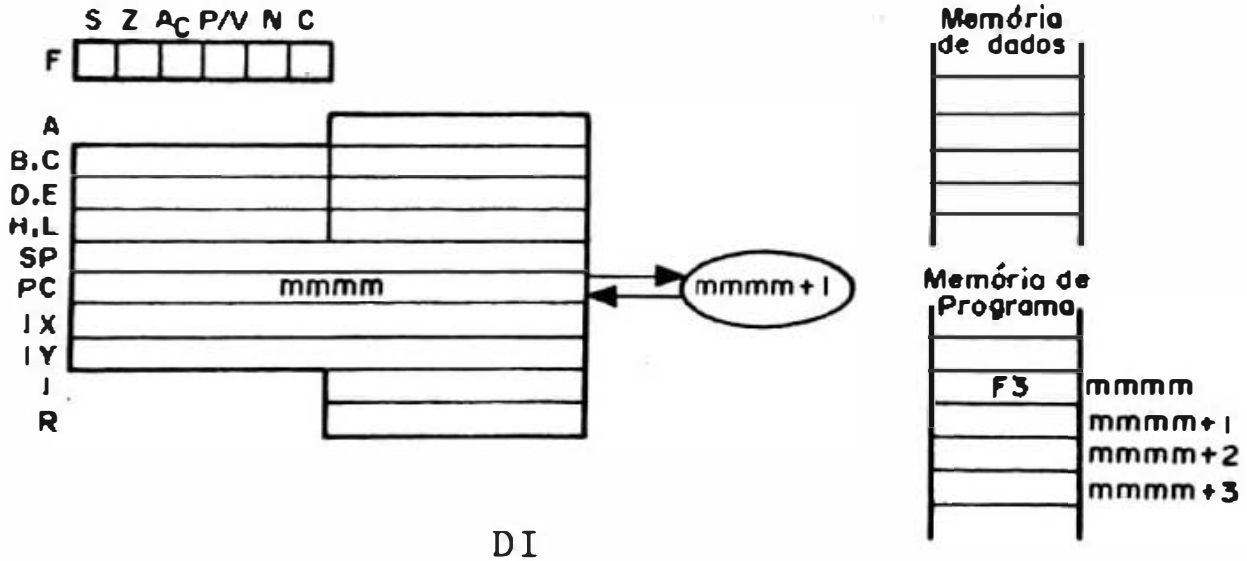
$$\underbrace{\text{DEC (Ix}}_{\text{DD 35}} + \underbrace{\text{disp)}}_{\text{d}}$$

Constituída por três bytes, sendo os dois primeiros o código de operação e o terceiro um byte de dados, a instrução DEC (Ix + disp) decrementa de uma unidade (1) a posição de memória endereçada pelo registrador de índice Ix, juntamente com o valor do deslocamento d.

$$\underbrace{\text{DEC (Iy}}_{\text{FD 35}} + \underbrace{\text{disp)}}_{\text{d}}$$

A instrução DEC (Iy + disp) é idêntica à instrução DEC (Ix + disp), exceto pelo fato da primeira usar o registrador Iy em vez do registrador Ix.

30) DI - DESATIVE INTERRUPÇÕES.



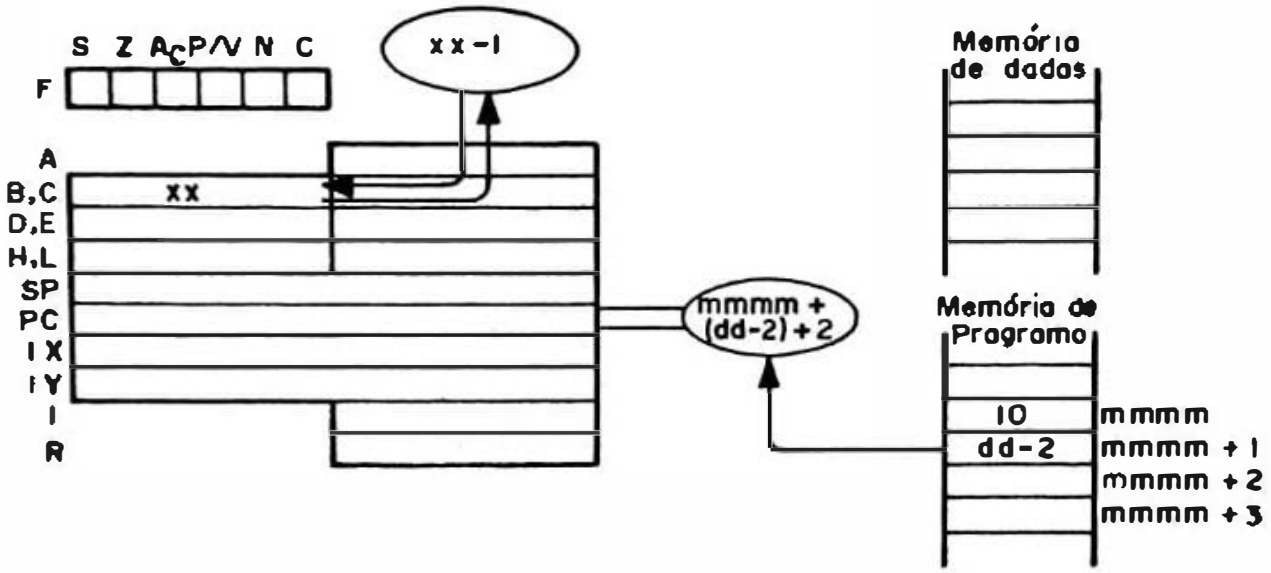
$$\underbrace{\text{DI}}_{\text{F3}}$$

Ao ser executada esta instrução, o pedido de interrupção mascarável (maskable) é desativado e a entrada $\overline{\text{INT}}$ da CPU passa a ser ignorada. Lembre-se de que, quando uma interrupção é reconhecida, a interrupção mascarável é automaticamente desativada.

O pedido de interrupção mascarável permanece desativado até que ele seja subseqüentemente desativado por uma instrução EI.

Nenhum registrador ou flag de posição é afetado por esta instrução.

31) DJNZ disp - SALTE RELATIVAMENTE AOS ATUAIS CONTEÚDOS DO CONTADOR DE PROGRAMA SE O REG B NÃO FOR ZERO.



DJNZ disp

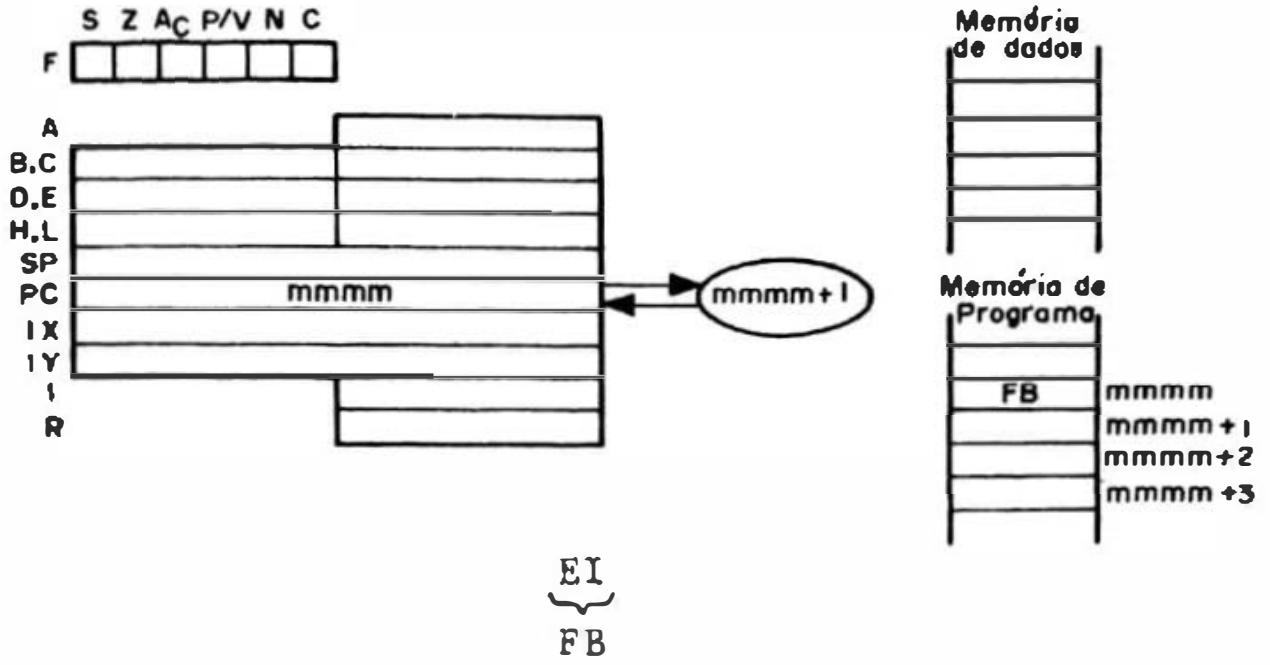
10 dd-2

Esta instrução é formada por dois bytes, sendo o primeiro o código de operação, e o segundo byte, um byte de dados. Esta instrução decrementa o registrador B, se o seu conteúdo for diferente de zero e adiciona 2 ao Contador de Programa. O salto é medido do endereço do código de operação da instrução e tem uma escala que vai de -126 a +129 bytes. O assembler ajusta automaticamente o PC duplamente incrementado.

Se após ter sido feita a decrementação, o conteúdo de B for zero (0), será então executada a próxima instrução seqüencial.

A instrução DJNZ é extremamente útil para qualquer operação de loop de programa, desde que a instrução 1 substitua a seqüência típica de "decrementação", desviando a condição.

32) EI - HABILITE INTERRUPÇÕES.



EI
FB

Constituída por um único byte, sendo este o código de operação, esta instrução habilita a interrupção mascarada.

Muitas rotinas de serviço de interrupção terminam com duas instruções:

EI : habilita interrupções
RET : retorna ao programa que foi interrompido.

Se as interrupções são processadas em série, para a duração inteira da rotina de serviço de interrupção todas as interrupções mascaradas serão desativadas. Isto significa que, numa aplicação de múltipla-interrupção, há grande possibilidade de que uma ou mais interrupções estejam pendentes, quando qualquer rotina de serviço de interrupção completa execução.

Se as interrupções fossem reconhecidas, tão logo as instruções EI fossem executadas, a instrução RETURN não seria executada. Sob estas circunstâncias, os retornos se acumulariam um após o outro, consumindo, desnecessariamente, espaço da pilha de memória. Veja ilustração a seguir:

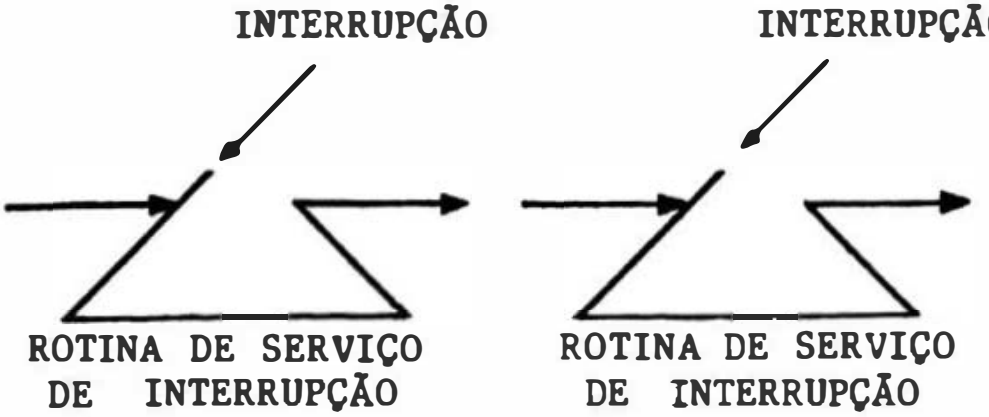
Inibindo as interrupções por mais uma instrução EI, a CPU do microprocessador Z-80 assegura que a instrução RET é executada na seqüência:

```
=
=
=
```

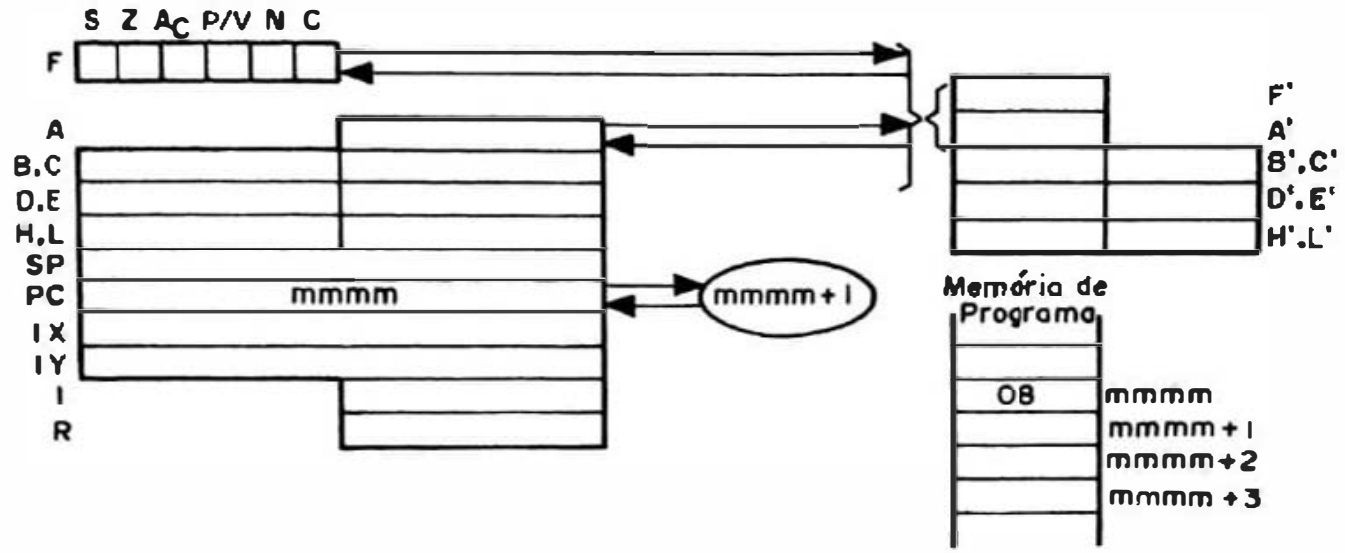
EI : habilita interrupções.

RET : retorna ao programa interrompido.

Não são raros os casos em que uma interrupção deve ser conservada enquanto uma rotina de serviço de interrupção estiver sendo executada. As interrupções são processadas em série:



33) EX AF,AF' - TROQUE A POSIÇÃO DO PROGRAMA E ALTERNE A POSIÇÃO DO PROGRAMA.

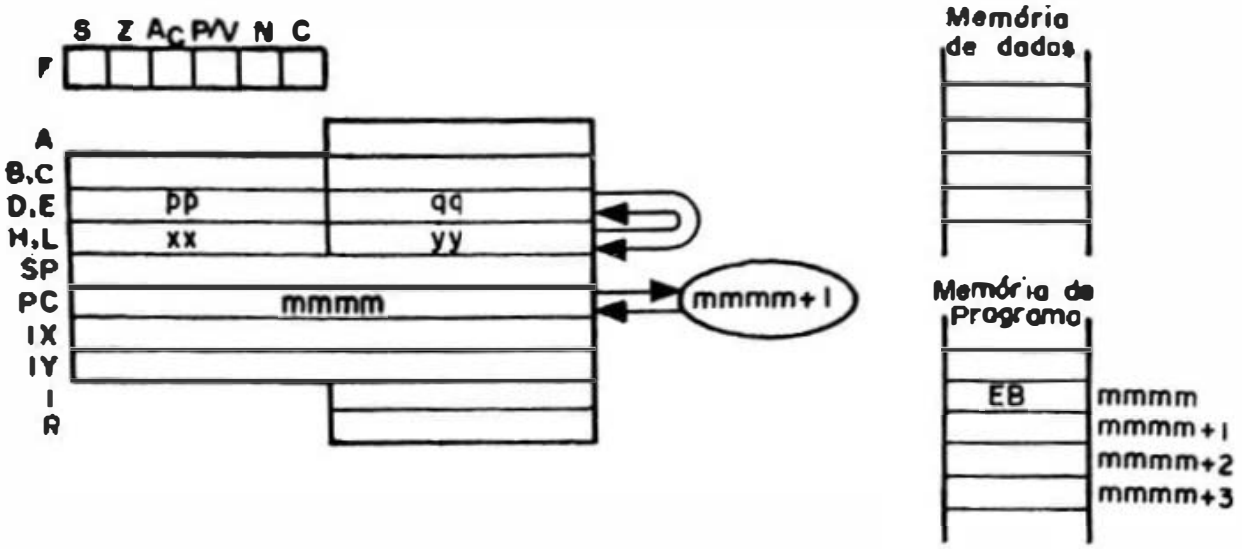


EX AF, AF'
08

Esta instrução é constituída por um único byte, sendo este o código de operação. Esta instrução troca o conteúdo do par de registradores AF pelo conteúdo do par AF'. Ao se executar esta instrução, todos os Flags serão afetados.

Suponhamos que o conteúdo do par de registradores AF seja $4F99_{16}$ e A'F' seja $10AA_{16}$. Após a execução da instrução EX AF,AF' o conteúdo de AF passa a ser $10AA_{16}$ e o conteúdo de AF' passa a ser $4F99_{16}$.

34) EX DE,HL - TROQUE O CONTEÚDO DOS PARES DE REGISTRADORES DE E HL.



$$\underbrace{\text{EX DE,HL}}_{\text{EB}}$$

Constituída por um único byte e sendo este o código de operação, esta instrução troca o conteúdo do par de registradores DE com o conteúdo do par de registradores HL.

Suponhamos que: $pp = 03_{16}$

$qq = 2A_{16}$

$xx = 41_{16}$

$yy = FC_{16}$

Após a execução da instrução EX DE,HL, o conteúdo dos registradores serão trocados. O novo conteúdo de cada um deles será:

$H = 03_{16}$

$L = 2A_{16}$

$D = 41_{16}$

$E = FC_{16}$

As duas instruções,

```
EX DE,HL
LD A,(HL)
```

são equivalentes a:

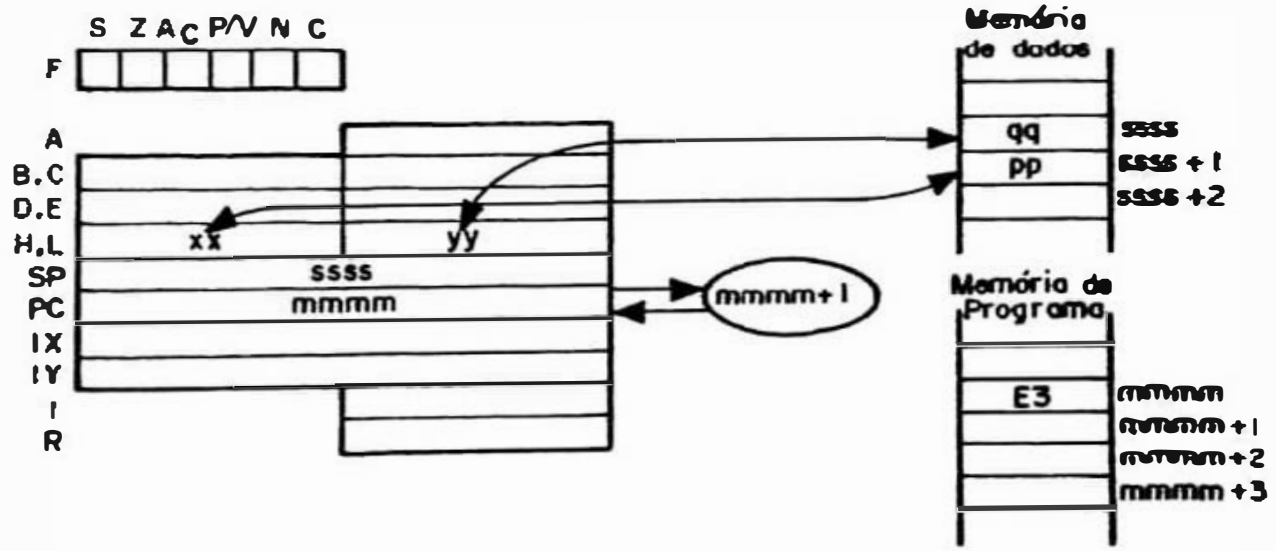
```
LD A,(DE)
```

Se desejarmos carregar os dados endereçados pelos registradores D e E no registrador B, podemos fazer:

```
EX DE,HL
LD B,(HL)
```

Para este último caso não há nenhuma instrução simples equivalente.

35) EX (SP),HL - TROQUE OS CONTEÚDOS DO REGISTRADOR E DO
EX (SP),Ix TOPO DA PILHA.
EX (SP),Iy



A ilustração nos mostra a execução de EX (SP),HL.

EX (SP),HL
E3

Esta instrução é constituída por dois bytes, sendo este o código de operação. Esta instrução troca o conteúdo do registrador L com o byte do topo da pilha e troca o conteúdo do registrador H com o byte logo abaixo do topo da pilha.

Suponhamos que: $xx = 21_{16}$

$yy = FA_{16}$

$pp = 3A_{16}$

$qq = E2_{16}$

Após a execução da instrução EX (SP),HL, o conteúdo do registrador H passa a ser $3A_{16}$ e o conteúdo do registrador L passa a ser $E2_{16}$. Por outro lado, os dois bytes do topo da pilha irão conter FA_{16} e 21_{16}, respectivamente.

A instrução EX (SP),HL é usada para acessar e manipular dados no topo da pilha.

EX (SP),Ix

DD E3

A instrução EX (SP),Ix é constituída por dois bytes, sendo estes o código de operação. Esta instrução troca o conteúdo do registrador de índice Ix pela posição de memória endereçada de Stack-Pointer (SP). A parte menos significativa de Ix é trocada com o "SP", enquanto a parte mais significativa de Ix é trocada com o "SP + 1".

EX (SP),Iy

FD E3

A instrução EX (SP),Iy também é constituída por dois bytes, sendo estes o código de operação. Esta instrução é idêntica à instrução EX (SP),Ix, porém utiliza o registrador IY em vez do registrador Ix.

CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-28

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) A instrução "AND reg" é constituída por:

a) três bytes;
b) quatro bytes;
c) dois bytes;
d) um único byte.

2) A instrução "BIT b,reg":

a) testa o bit b do Contador de Programa (PC);
b) testa o bit b do registrador reg.;
c) testa o conteúdo do Flag de posição P/V;
d) N.R.A.

3) A instrução "CALL label" é uma instrução do tipo:

a) incondicional;
b) puramente condicional;
c) tríplice, pois é formada por três bytes;
d) N.R.A.

4) A instrução "CALL condition,label" é do tipo:

a) incondicional;
b) tríplice, pois é formada por três bytes;

c) condicional;
d) N.R.A.

5) A instrução, de um só byte, que complementa o Carry bit é:

a) CCF;
b) BIT b,(HL);
c) DEC rp;
d) N.R.A.

6) A instrução CP (Ix + disp) é constituída por:

a) um byte;
b) dois bytes;
c) três bytes;
d) quatro bytes.

7) A instrução CPDR é idêntica à instrução:

a) CCF;
b) CPD, exceto pelo fato dela ser repetitiva;
c) CPI;
d) N.R.A.

8) A diferença entre a instrução "CPI" e a instrução "CPD" reside no fato de que na primeira, o conteúdo do par de registradores HL é:

a) incrementado ao invés de decrementado;
b) decrementado ao invés de incrementado;
c) zerado;
d) N.R.A.

9) A instrução DEC (HL) decrementa a posição de memória endereçada pelo par de registradores "HL", de:

a) cinco unidades;
b) três unidades;
c) duas unidades;
d) uma unidade.

10) A instrução EI, de um único byte:

a) desativa o pedido de interrupção mascarável (ou mascarada);
b) habilita o pedido de interrupção mascarada;
c) troca o conteúdo dos pares de registradores DE e HL;
d) N.R.A.

CURSOS DE APERFEIÇOAMENTO

CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

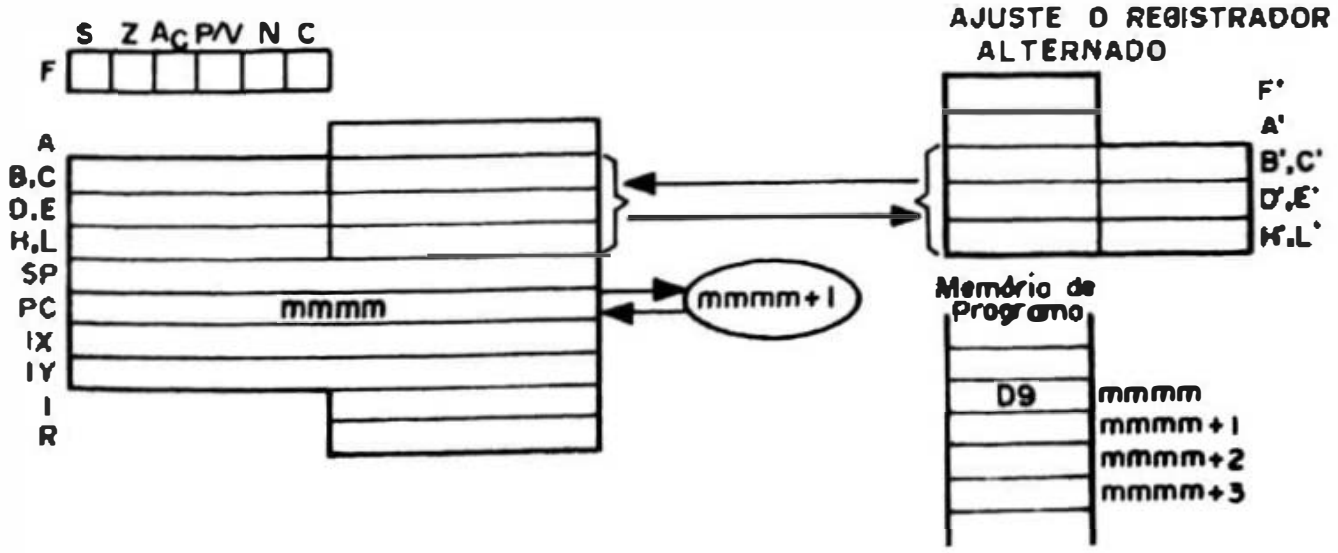
CAIXA POSTAL 1642 · CEP 86100 · LONDRINA - PARANÁ

LIÇÃO MC-29

PROGRAMAÇÃO ASSEMBLER IV

Dando continuidade à lição anterior, estudaremos agora mais algumas instruções do microprocessador Z-80.

36) EXX - TROQUE OS PARES DOS REGISTRADORES E ALTERNE OS PARES DE REGISTROS.



EXX
D9

Formada por um único byte e sendo este o código de operação, esta instrução troca o conteúdo dos pares de registradores BC, DE e HL pelo conteúdo dos pares de registradores BC', DE' e HL'.

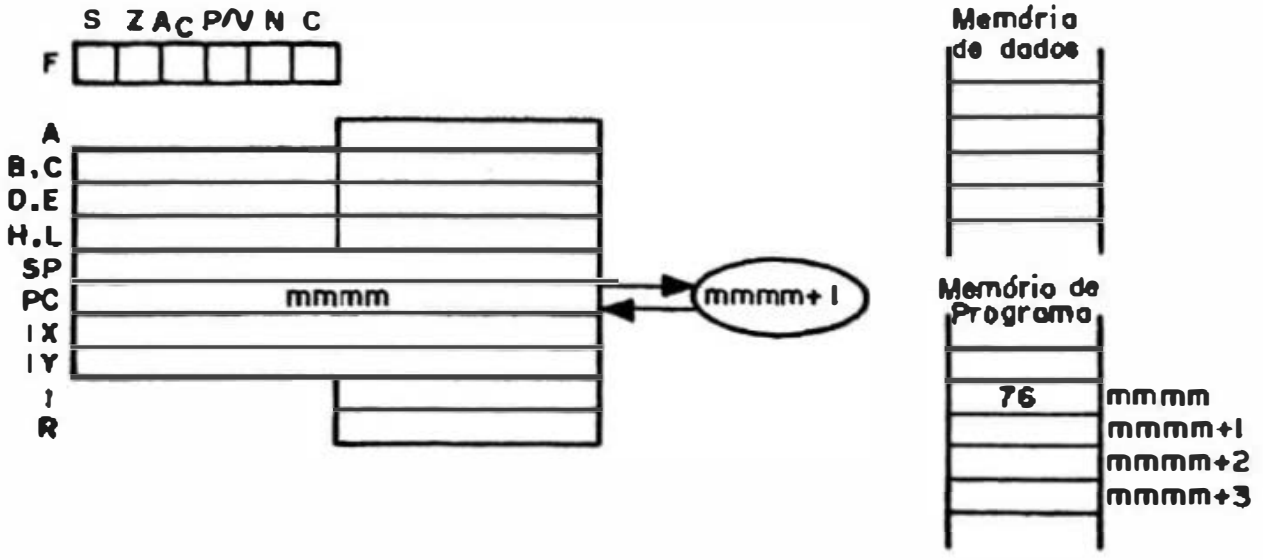
Suponhamos que os pares de registradores BC, DE e HL contenham 4901_{16}, $5F00_{16}$ e 7251_{16}, respectivamente, e que os pares de registradores BC', DE' e HL' contenham 000_{16}, $10FF_{16}$ e 3333_{16}, respectivamente. Após a execução da instrução EXX, o novo conteúdo desses pares de registradores passa a ser:

$BC = 0000_{16}$; $DE = 10FF_{16}$; $HL = 3333_{16}$

$BC' = 4901_{16}$; $DE' = 5F00_{16}$; $HL' = 7251_{16}$

Esta instrução (EXX) é utilizada para resguardar o conteúdo dos registradores dentro do próprio microprocessador. Além disto, ela pode também ser usada para trocar os bancos de registradores, de modo a fornecer tempos de interrupção muito mais rápidos.

37) HALT



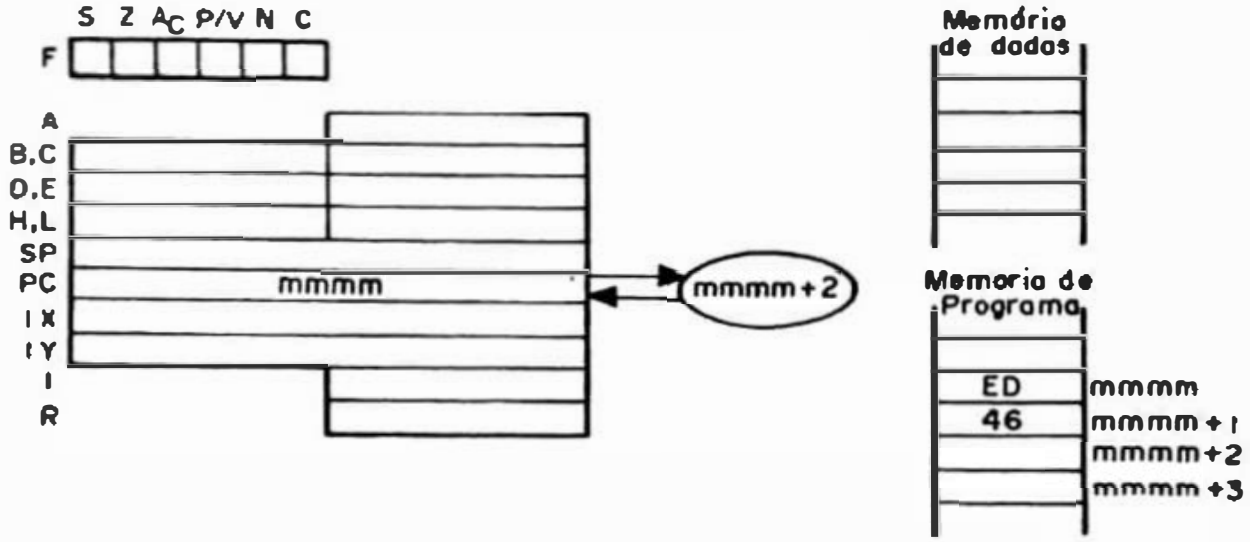
HALT
76

Constituída por um único byte, e sendo este o próprio código de operação, esta instrução ao ser executada, faz com que a CPU entre num estado de retenção, fazendo com que a CPU pare de executar qualquer outra instrução após esta, ou seja, esta instrução pára a execução do programa. Para sair deste es-

tado e recomeçar o programa, a CPU exige uma "interrupção" ou um "reset".

Nenhum registrador ou posição de Flag será afetado e a lógica de refrescamento (refresh) das memórias dinâmicas continuará operando normalmente.

38) IM 0 - MODO 0 (ZERO) DE INTERRUPÇÃO



IM 0
ED 46

Esta instrução é formada por dois bytes, os quais constituem o código de operação. Esta instrução coloca a CPU no modo 0 (zero) de interrupção, isto é, quando a linha $\overline{INT}$ for colocada no nível lógico "zero", ocorrerá um salto no programa para uma posição, dependendo do código colocado no barramento de dados (DATA BUS) e a CPU irá então executar aquela instrução. Nenhum registrador ou posição será afetado.

39) IM 1 - MODO 1 DE INTERRUPÇÃO.

IM 1
ED 56

Também formada por dois bytes, os quais constituem o código de operação, esta instrução coloca a CPU no modo 1 de interrupção. Neste modo, a CPU responde a uma interrupção, executando um "restart" (RST) para o endereço 0038$_{16}$, não sendo neces

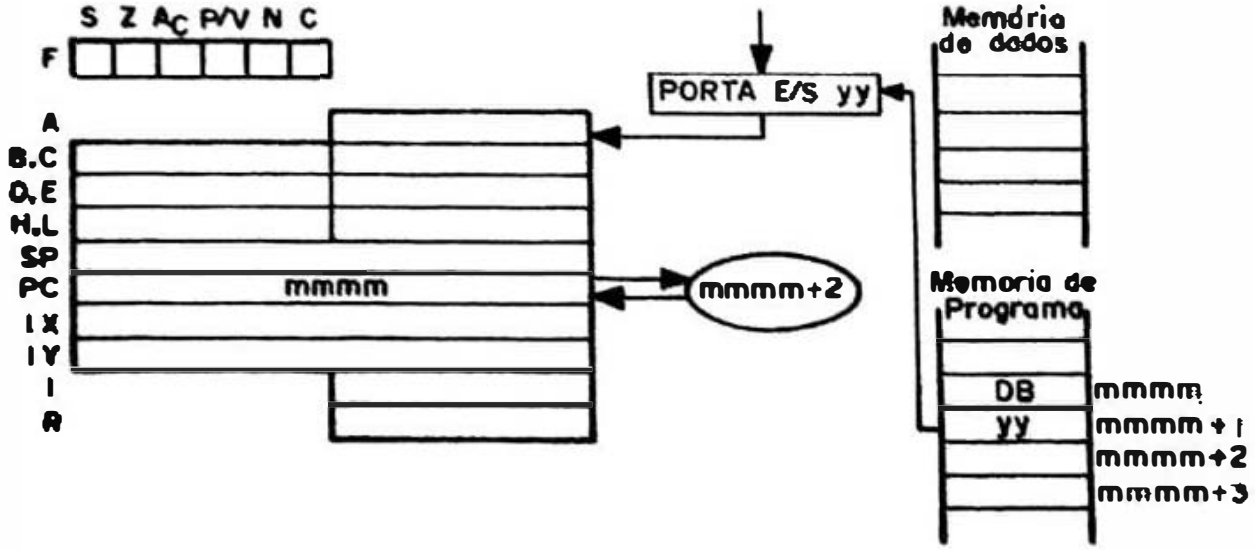
sário colocar nenhuma informação no barramento de dados (DATA BUS).

40) IM 2 - MODO 2 DE INTERRUPÇÃO.

IM 2
ED 5E

Constituída por dois bytes, sendo estes o código de operação, esta instrução coloca a CPU no modo 2 de interrupção. Neste modo, a CPU realiza uma chamada indireta a qualquer localização especificada na memória. Um endereço de 16 bits é formado, utilizando os conteúdos do registrador do vetor de interrupção (8 bits mais significativos) e do dispositivo de interrupção (8 bits menos significativos). Nenhum registro ou posição é afetado por esta instrução.

41) IN A,(port) - ENTRE NO ACUMULADOR.



IN A, (port)
DB yy

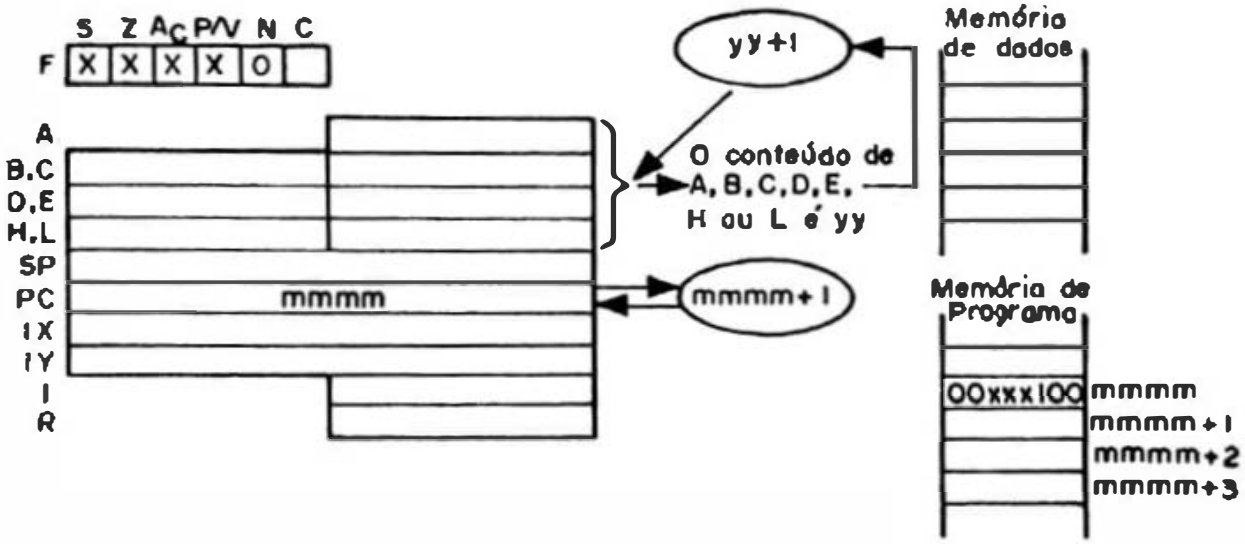
Esta instrução é constituída por dois bytes, sendo o primeiro o código de máquina e o segundo, um byte de dados, o qual irá indicar o endereço (ou localização) do dispositivo de entrada/saída (E/S). Esta instrução coloca no acumulador o conteúdo (um byte de dados) do dispositivo de entrada/saída, endereçado pelo segundo byte desta instrução (yy).

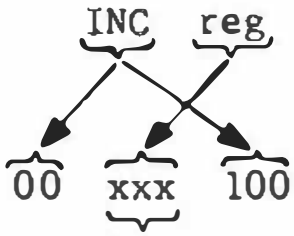
Suponhamos que 36_{16} seja o conteúdo do buffer da porta de

entrada $1A_{16}$. Após a execução da instrução IN A,(1AH), o conteúdo do acumulador será 36_{16}.

A instrução IN não afeta qualquer posição. Por outro lado, o uso desta instrução depende em muito do hardware, uma vez que os endereços válidos da porta de E/S são determinados pelo modo, no qual a lógica de E/S tenha sido implementada. Também é possível projetar um sistema de microcomputador que acesse a lógica externa, usando instruções de referência de memória com endereços específicos de memória.

42) INC reg - INCREMENTE O CONTEÚDO DOS REGISTRADORES.





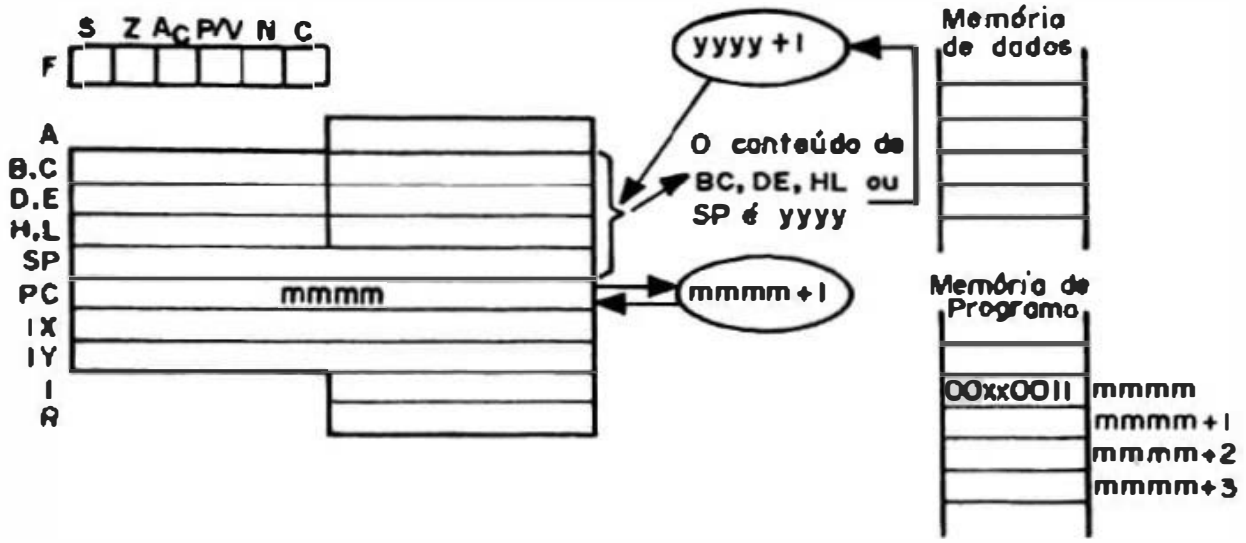
000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por um único byte, o qual constitui o próprio código de operação, esta instrução incrementa de uma unidade o conteúdo do registrador especificado.

Suponhamos que o conteúdo do registrador E seja $A8_{16}$.

Após a execução da instrução INC E, o registrador E passa a conter $A9_{16}$.

43) INC rp - INCREMENTE O CONTEÚDO DO PAR DE REGISTRADOR
INC Ix ESPECIFICADO.
INC Iy



A ilustração nos mostra a execução da instrução INC rp:

INC rp

00 xx 0011

00 para rp = BC
01 para rp = DE
10 para rp = HL
11 para rp = Stack-Pointer

Formada por um único byte, o qual constitui o código de operação, esta instrução incrementa de uma unidade o conteúdo do par de registradores "rp". Nenhum flag de posição é afetado.

Suponhamos que o conteúdo do par de registradores DE seja $2F7A_{16}$. Após a execução da instrução INC DE, o conteúdo do registrador DE passa a ser $2F7B_{16}$.

INC Ix

DD 23

A instrução INC Ix é formada por dois bytes, os quais

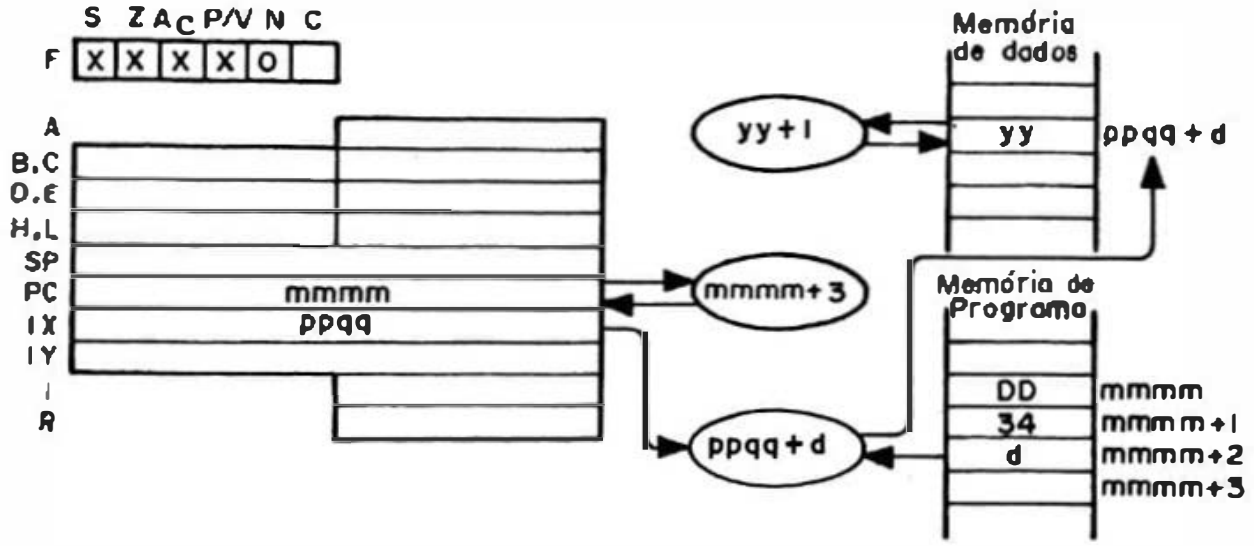
constituem o código de operação. Esta instrução incrementa de uma unidade o conteúdo do registrador Ix de 16 bits.

$$\underbrace{\text{INC Iy}}_{\text{FD 23}}$$

A instrução INC Iy também é formada por dois bytes, os quais constituem o código de operação. Esta instrução è idêntica à instrução INC Ix, exceto quanto ao uso do registrador de índice, pois enquanto aquela usa o Iy, esta utiliza o Ix.

Nenhuma dessas instruções afeta os flags de posição.

44) INC (HL) - INCREMENTE OS CONTEÚDOS DA MEMÓRIA.
INC (Ix+disp)
INC (Iy+disp)



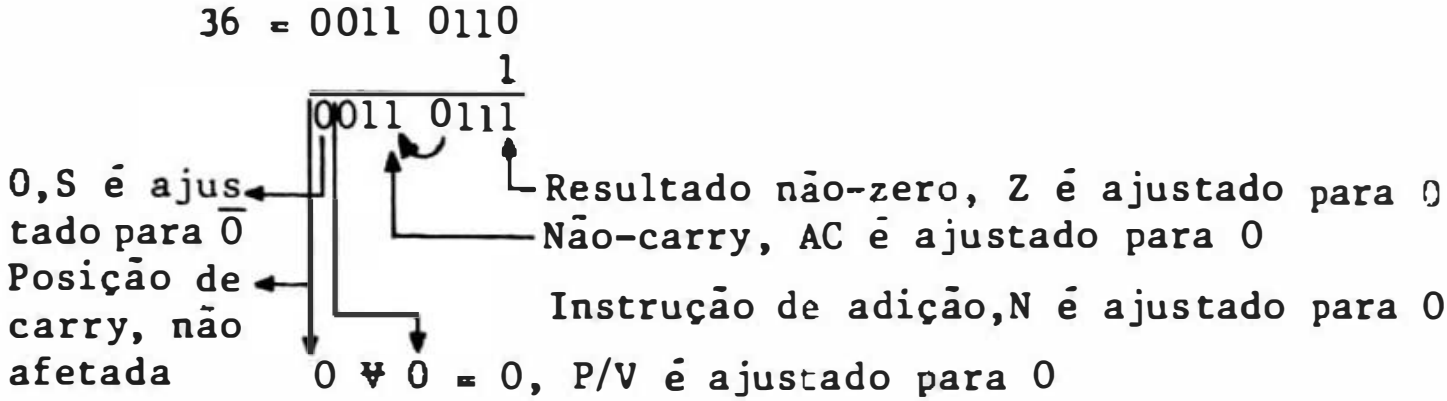
A ilustração nos mostra a execução da instrução INC (Ix + disp):

$$\underbrace{\text{INC (Ix}}_{\text{DD 34}} + \underbrace{\text{disp)}}_{\text{d}}$$

Constituída por três bytes, sendo os dois primeiros o código de operação e o terceiro um byte de dados. A instrução INC (Ix + disp) incrementa de uma unidade o conteúdo da localização de memória endereçada pelo registrador de índice Ix , somado com o valor do deslocamento d (disp).

Suponhamos que ppqq = 4000_{16} e que o conteúdo da localização de memória $400F_{16}$ seja igual a 36_{16}. Após a execução da

instrução INC (Ix + 0FH), onde 0FH é o deslocamento, a localização de memória $400F_{16}$ irá conter 37_{16}, veja:



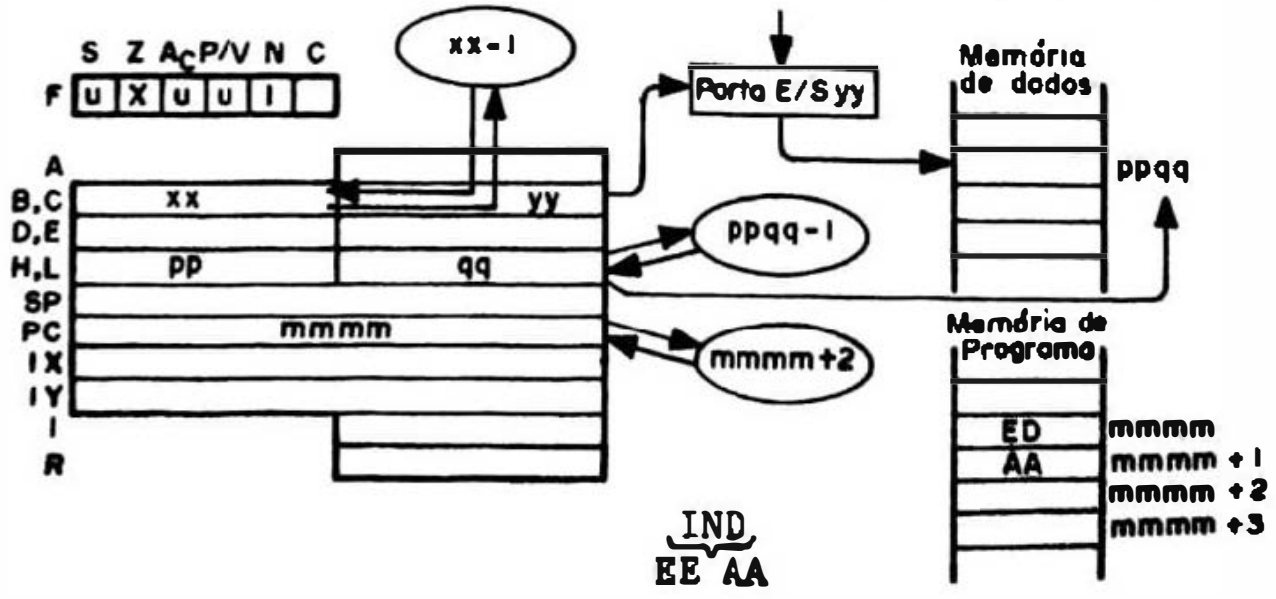
$$\underbrace{\text{INC (Iy}}_{\text{FD 34}} + \underbrace{\text{disp)}}_{\text{d}}$$

A instrução INC (Iy + disp), formada por três bytes sendo os dois primeiros o código de operação e o terceiro um byte de dados, é idêntica à instrução INC (Ix + disp) exceto pelo fato desta utilizar o registrador de índice Iy em vez do registrador Ix.

$$\underbrace{\text{INC (HL)}}_{\text{34}}$$

A instrução INC (HL) é formada por um único byte, o qual constitui o código de operação. Esta instrução incrementa de uma unidade o conteúdo da localização de memória endereçada pelo par de registradores HL.

45) IND - ENTRE NA MEMÓRIA E DECREMENTE O APONTADOR.



Formada por dois bytes, os quais constituem o código de operação, esta instrução carrega o conteúdo de um dispositivo de E/S, endereçado pelo registrador C, para uma localização de memória endereçada pelo par de registradores HL. O registrador B (utilizado como contador) é decrementado cada vez que esta instrução for executada. O par de registradores HL, por sua vez, é decrementado para carregamento de outros dados.

Suponhamos que xx = 05_{16}, yy = 15_{16}, ppqq = 2400_{16} e que o valor 19_{16} é seguro no buffer da porta de E/S 15_{16}. Após a execução da instrução IND, a localização de memória 2400_{16} irá conter o valor que estava seguro no buffer, ou seja, 19_{16}. O registrador B irá conter 04_{16} e o par de registradores HL irá conter $23FF_{16}$ (teve o seu conteúdo decrementado de uma unidade).

46) INDR - ENTRE NA MEMÓRIA E DECREMENTE O APONTADOR ATÉ QUE O CONTADOR DE BYTE SEJA ZERO.

INDR

ED BA

Esta instrução é formada por dois bytes, os quais constituem o código de operação. Esta instrução é idêntica à instrução IND, ou seja, segue os mesmos caminhos, porém entra em loop, fazendo as mesmas operações inúmeras vezes, saindo do loop somente quando o registrador B, que é utilizado como contador, for igual a zero.

Suponhamos que o registrador B contenha 03_{16}, o registrador C, 15_{16} e o par de registradores HL, 2400_{16}. A seqüência de bytes, a seguir, encontra-se disponível na porta de E/S 15_{16}:

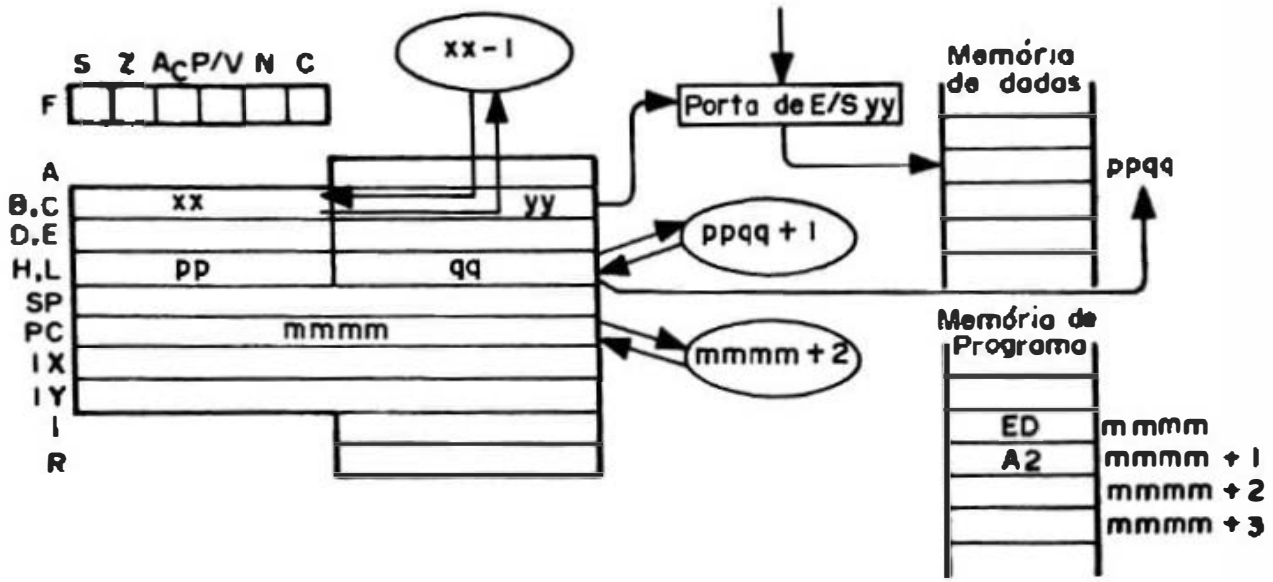
17_{16}, 59_{16} e AE_{16}

Depois de executada a instrução INDR, o par de registradores HL passa a conter $23FD_{16}$ e o registrador B contém zero, enquanto que as localizações de memória passam a conter a seqüência:

Localização	Conteúdo
2400	17_{16}
23FF	59_{16}
23FE	AE_{16}

Esta instrução é extremamente útil para carregar blocos de dados de um dispositivo de entrada para dentro da memória.

47) INI - ENTRE NA MEMÓRIA E INCREMENTE O APONTADOR.



INI

ED A2

Formada por dois bytes, os quais constituem o código de operação, esta instrução é semelhante à instrução IND, exceto pelo fato de incrementar o conteúdo do par de registradores "HL" em vez de decrementar.

Suponhamos que xx = 05_{16}, yy = 15_{16}, ppqq = 2400_{16} e o valor 19_{16} é seguro no buffer da porta de E/S 15_{16}.

Após a execução da instrução INI, a localização de memória 2400_{16} irá conter o valor que estava seguro no buffer, ou seja, 19_{16}. O registrador B passa a conter 04_{16} enquanto que o par de registradores HL passa a conter 2401_{16}.

48) INIR - ENTRE NA MEMÓRIA E INCREMENTE O APONTADOR ATÉ QUE O CONTADOR DE BYTE SEJA ZERO.

INIR

ED B2

Esta instrução é constituída por dois bytes, sendo estes o código de operação. Esta instrução é idêntica à instrução IN1, ou seja, segue os mesmos caminhos, porém entra em loop, fazendo as mesmas operações inúmeras vezes, saindo do loop somente quando o registrador B, que é utilizado como contador, for igual a zero.

Suponhamos que o registrador B contenha 03_{16}, o registrador C contenha 15_{16} e o par de registradores HL contenha 2400_{16}. A seqüência de bytes, a seguir, encontra-se disponível na porta de E/S 15_{16}:

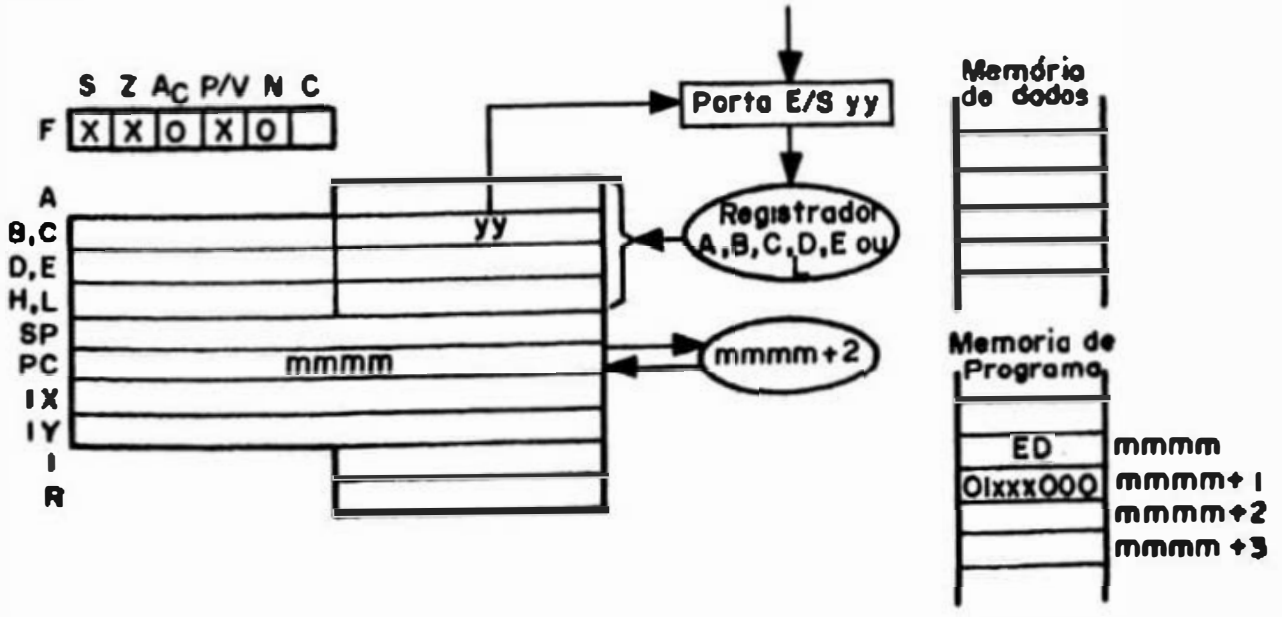
$$17_{16};\ 59_{16}\ \text{e}\ AE_{16}$$

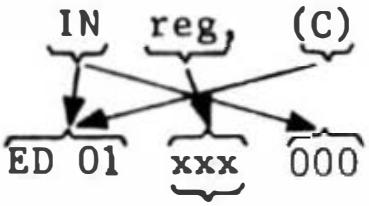
Após a execução da instrução INIR, o par de registradores HL passa a conter 2403_{16} e o registrador B contém zero, enquanto que as localizações de memória passam a conter a seqüência:

Localização	Conteúdo
2400	17_{16}
2401	59_{16}
2402	AE_{16}

Esta instrução é extremamente útil para carregar blocos de dados de um dispositivo de entrada para dentro da memória.

49) IN reg,(C) - ENTRE NOS REGISTRADORES.





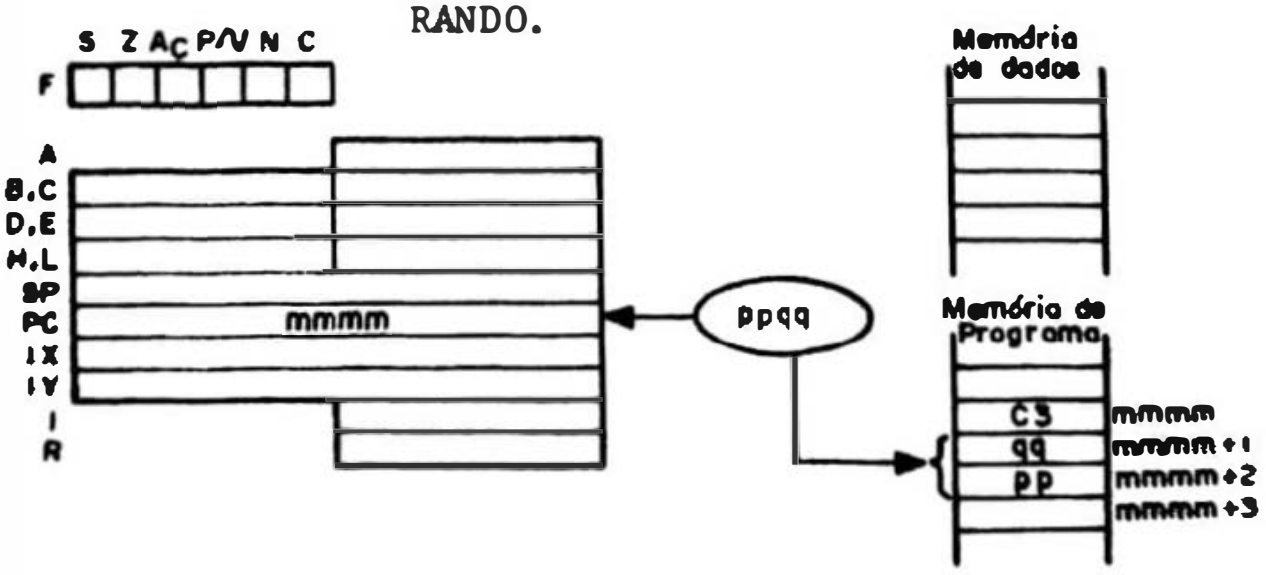
000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A
110 para ajustamento dos flags de posição.

Formada por dois bytes, os quais constituem o código de operação, esta instrução carrega um byte de dados no registrador especificado (reg), sendo que este byte de dados se encontra no dispositivo de E/S, endereçado pelo conteúdo do registrador C.

Suponhamos que o valor 42_{16} se encontre seguro no buffer da porta de E/S 36_{16} e que o conteúdo do registrador seja 36_{16}. Após a execução da instrução IN D,(C), o registrador D passa a conter o valor 42_{16}.

Durante a execução da referida instrução, o conteúdo do registrador B é colocado na metade superior do barramento de endereço, possibilitando, desta forma, estender o número de portas de E/S endereçáveis.

50) JP label - SALTE PARA A INSTRUÇÃO IDENTIFICADA NO OPERANDO.



```
JP  label
C3  ppqq
```

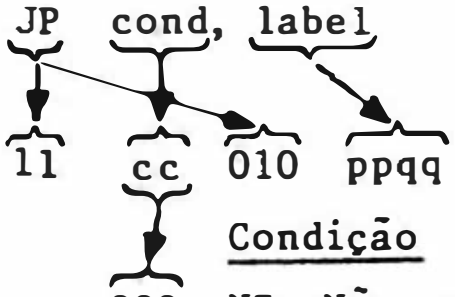
Esta instrução é formada por três bytes. O primeiro deles constitui o código de operação e os dois últimos, o endereço que irá modificar o CONTADOR DE PROGRAMA (PC). Esta instrução é usada para dar saltos (JUMP) e a modificação do CONTADOR DE PROGRAMA nos indica o endereço de memória para a próxima instrução a ser executada. Os conteúdos prévios (anteriores) do CONTADOR DE PROGRAMA (PC) estão perdidos.

A título de exemplo, consideremos a seguinte seqüência:

```
      JP    NEXT
      AND   7FH
      OR    B1
      =
      =
      =
NEXT  CPL
```

A instrução CPL será executada logo após a instrução JP. A instrução AND e a instrução OR nunca serão executadas, a menos que seja colocada uma outra instrução JP em qualquer outro lugar da seqüência, de modo a provocar um salto para estas instruções.

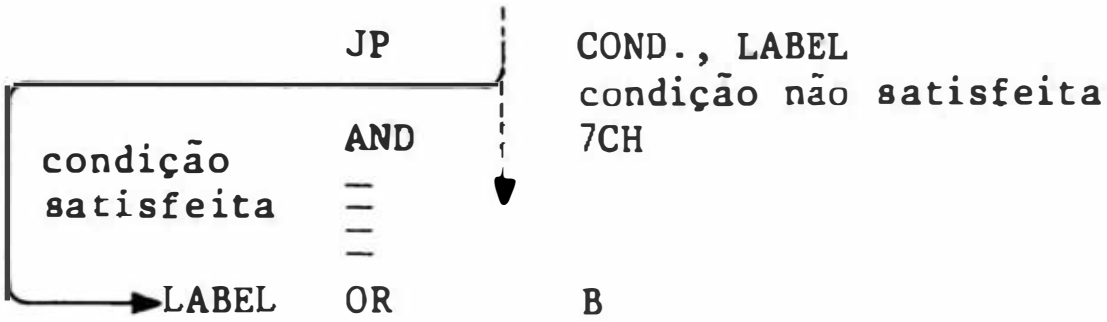
51) JP condicional, label - SALTE PARA O ENDEREÇO IDENTIFICADO NO OPERANDO SE A CONDIÇÃO FOR SATISFEITA.



	Condição		Flag relevante
000	NZ	Não-zero	Z
001	Z	Zero	Z
010	NC	Não-carry	C
011	C	Carry	C
100	PO	Paridade ímpar	P/V
101	PE	Paridade par	P/V
110	P	Sinal positivo	S
111	M	Sinal negativo	S

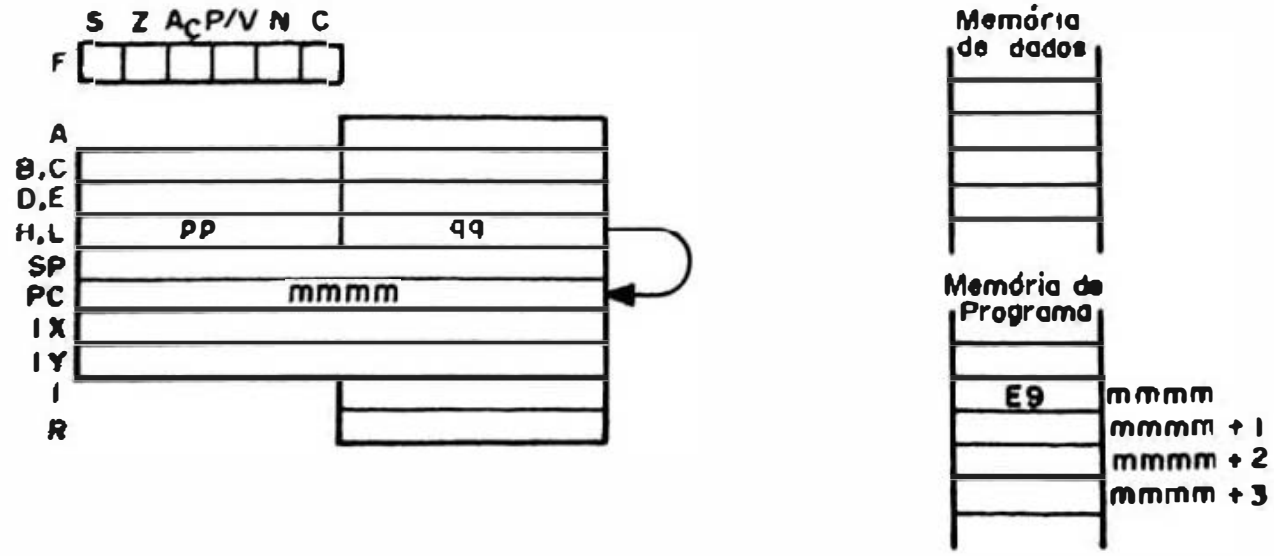
Esta instrução é idêntica à instrução anterior "JP label" exceto pelo fato de que o salto somente será realizado se a condição (cc) for satisfeita, caso contrário, a instrução seguinte à instrução "JP cond, label", em uma dada seqüência, será executada.

Consideremos a seguinte seqüência de instruções:



Após a execução da instrução JP cond., label, se a condição for satisfeita, a instrução OR será executada, caso contrário, a instrução AND, que é a próxima instrução seqüencial, será executada.

52) JP (HL) - SALTE PARA O ENDEREÇO ESPECIFICADO PELOS
JP (Ix) CONTEÚDOS DO REGISTRADOR DE 16 BITS.
JP (Iy)



A ilustração nos mostra a execução da instrução JP (HL).

JP (HL)
E9

A instrução JP (HL) é formada por dois bytes, os quais constituem o código de operação. Os conteúdos do par de registradores HL são movidos para o CONTADOR DE PROGRAMA, modificando-o. Esta instrução realiza então um salto incondicional, especificado pelo conteúdo do par de registradores HL.

Consideremos a seqüência:

```
LD      H, ADDR
JP      (HL)
```

Esta seqüência de instruções tem o mesmo efeito que a instrução JP ADDR, pois ambos os casos especificam que a instrução com título ADDR deve ser executada a seguir.

A instrução JP (HL) é muito útil, quando se deseja incrementar um endereço de retorno para uma sub-rotina que possui múltiplos retornos.

```
JP   (Ix)
DD   E9
```

A instrução JP (Ix) é formada por dois bytes, os quais constituem o código de operação. Esta instrução é idêntica à instrução JP (HL), exceto pelo fato desta utilizar o registrador Ix em vez do par de registradores HL.

```
JP   (Iy)
FD   E9
```

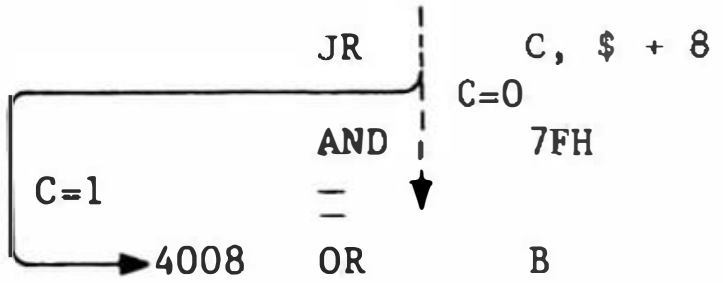
Esta instrução é idêntica à instrução JP (Ix), exceto pelo fato desta utilizar o registrador Iy em vez do registrador Ix.

53) JR C, disp - DE UM SALTO RELATIVO SOBRE O CONTEÚDO DO CONTADOR DE PROGRAMA SE O CARRY É AJUSTADO.

```
JR C , disp
38     dd-2
```

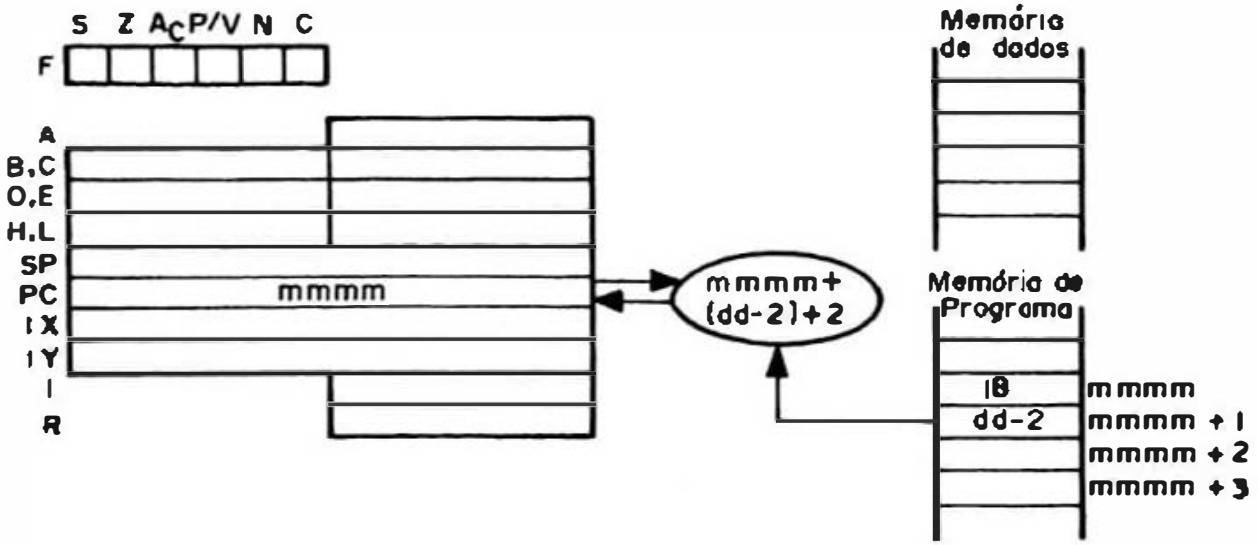
Esta instrução é idêntica à instrução JR disp, exceto pelo fato de que o salto somente é executado se a posição carry for igual a 1, caso contrário, a próxima instrução será executada.

Consideremos a seguinte seqüência de instruções:



Após a instrução JR C, $+8, a instrução OR será executada se somente a posição carry for igual a 1, caso contrário, a instrução AND será executada, obedecendo a seqüência normal apresentada.

54) JR disp - DÊ UM SALTO RELATIVO SOBRE O ATUAL CONTEÚDO DO CONTADOR DE PROGRAMA (PC).



JR disp

18 dd-2

Constituída por dois bytes, sendo o primeiro o código de operação e o segundo, um byte de dados. Esta instrução soma o conteúdo do segundo byte (byte de dados) do código objeto, com o conteúdo do CONTADOR DE PROGRAMA mais dois (2). O salto é medido, a partir do código de operação da instrução e tem uma escala que pode variar de -126 a +129 bytes. O Assembler ajusta automaticamente o CONTADOR DE PROGRAMA (PC) duplamente incrementado.

A seguinte declaração de linguagem assembly e usada para

saltar quatro passos à frente do endereço 4000_{16}:

```
JR $+4
```

O resultado desta instrução (JR $+4) é mostrado a seguir:

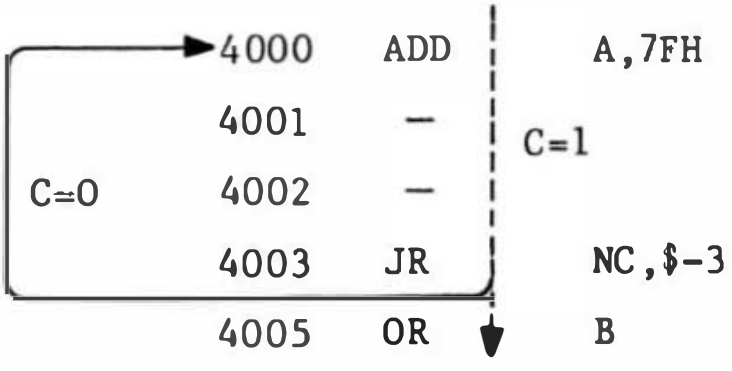
Localização	Instrução	
4000	18	
4001	02	
4002	—	
4003	—	
4004	—	← novo valor do PC.

55) JR NC, disp - DÊ UM SALTO RELATIVO SOBRE O ATUAL CONTEÚDO DO CONTADOR DE PROGRAMA SE O FLAG CARRY FOR RESETADO.

```
JR NC, disp
 30    dd-2
```

Esta instrução é idêntica à instrução JR disp, exceto pelo fato de que o salto somente será executado se a posição carry for igual a zero (resetada), caso contrário, a próxima instrução será executada.

Consideremos a seguinte seqüência de instruções:



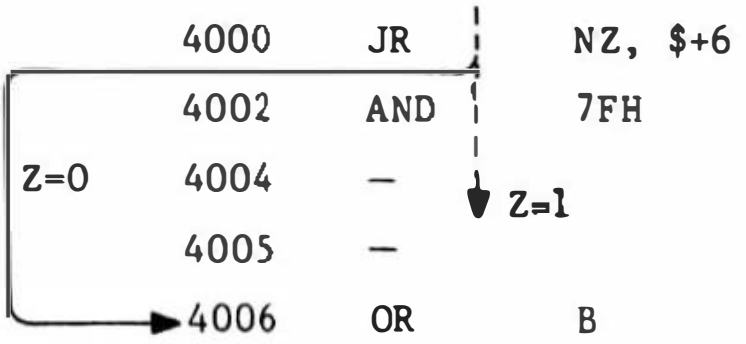
Após a instrução JR NC,$-3, a instrução OR será executada somente se a posição C for igual a 1 (C=1), caso contrário, (C=0) a instrução ADD será a próxima instrução a ser executada, seguindo a seqüência normal das instruções.

56) JR NZ, disp - DÊ UM SALTO RELATIVO SOBRE O ATUAL CONTEÚDO DO CONTADOR DE PROGRAMA SE O FLAG ZERO FOR RESETADO (RESET).

JR Z, disp
28 dd-2

Esta instrução é idêntica à instrução JR disp, exceto pelo fato de que o salto somente será executado se a posição ZERO for igual a zero (resetado). Caso contrário, a próxima instrução será executada.

Consideremos a seguinte seqüência de instruções:



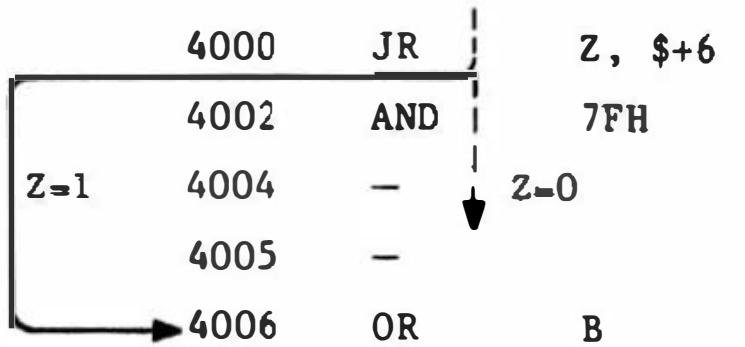
Após a instrução JR NZ,$+6, a instrução OR somente será executada se a posição Z for igual a zero (Z=0). Caso contrário, a próxima instrução, a ser executada, será a instrução AND.

57) JR Z, disp - DE UM SALTO RELATIVO SOBRE O ATUAL CONTEÚDO DO CONTADOR DE PROGRAMA SE O FLAG ZERO FOR SET (Z=1)

JR Z, disp
28 dd-2

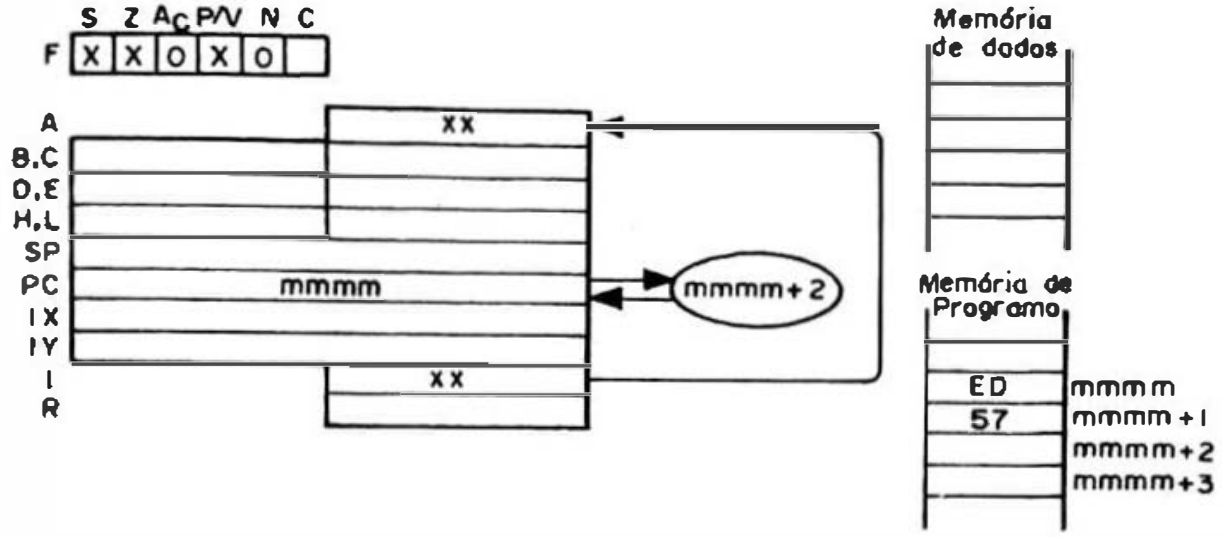
Esta instrução é idêntica à instrução JR disp, exceto pelo fato de que o salto somente será executado se a posição ZERO for igual a 1 (Z=1). Caso contrário, a próxima instrução será executada.

Consideremos a seguinte seqüência de instruções:



Após a instrução JR Z,$+6, a instrução OR será executada somente se a posição Z for igual a 1 (Z=1). E a instrução AND será executada se a posição Z for igual a zero (Z=0).

58) LD A,I - MOVA O CONTEÚDO DO REGISTRADOR I (INTERRUP-
LD A,R ÇÃO) OU R (REFRESCAMENTO) PARA O ACUMULADOR.



A ilustração nos mostra a execução da instrução LD A,I.

$$\underbrace{\text{LD A,I}}_{\text{ED 57}}$$

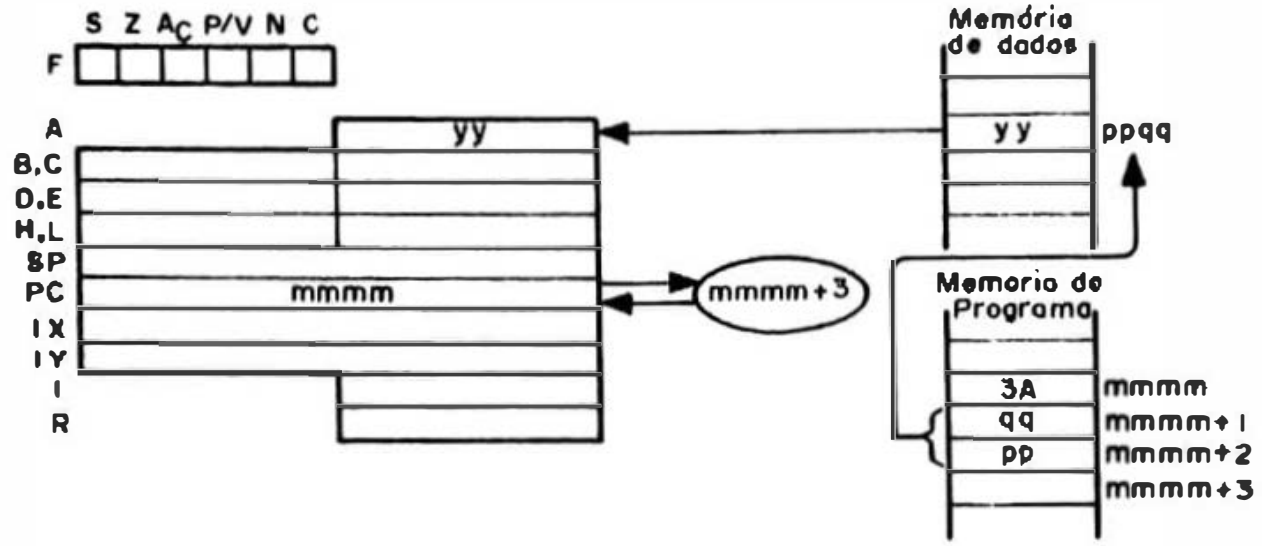
Formada por dois bytes, os quais constituem o código de operação, esta instrução move (ou carrega) o conteúdo do registrador vetor de Interrupção (I) para dentro do acumulador. Ao realizar esta operação, o Flag P/V é afetado.

Suponhamos que o registrador I contenha $7F_{16}$ e que as interrupções estejam desativadas. Após a execução da instrução LD A,I, o acumulador passa a conter o valor $7F_{16}$ e o flag P/V passa a ser igual a zero (0).

$$\underbrace{\text{LD A,R}}_{\text{ED 5F}}$$

A instrução LD A,R também é formada por dois bytes, os quais constituem o código de operação. Esta instrução move o conteúdo do registrador Refresh (R) para dentro do acumulador. Ao efetuar esta operação, o valor do flip-flop de interrupção irá aparecer no flag P/V. Este tipo de instrução é muito utilizada quando se trabalha com memórias dinâmicas, as quais necessitam de Refresh.

59) LD A, (addr) - CARREGUE O ACUMULADOR DA MEMÓRIA USA DO ENDEREÇAMENTO DIRETO.



LD A, (addr)

3A ppqq

Esta instrução é constituída por três bytes, sendo o primeiro o código de operação e os dois últimos o endereço. Esta instrução carrega (LOAD) os conteúdos do byte de memória, endereçados diretamente pelo segundo e terceiro byte do código objeto de LD A,(addr), para dentro do acumulador.

Suponhamos que o byte de memória $084A_{16}$ contenha 20_{16}. Após a execução da instrução,

```
Label   EQU 084A16
=
=
LD      A,(label)
```

o conteúdo do acumulador passa a ser igual a 20_{16}.

Note que EQU é um assembler diretivo e é mais do que uma instrução, pois ele diz ao assembler para usar o valor $084A_{16}$ de 16 bits, onde quer que o título ou rótulo (label) apareça.

A instrução

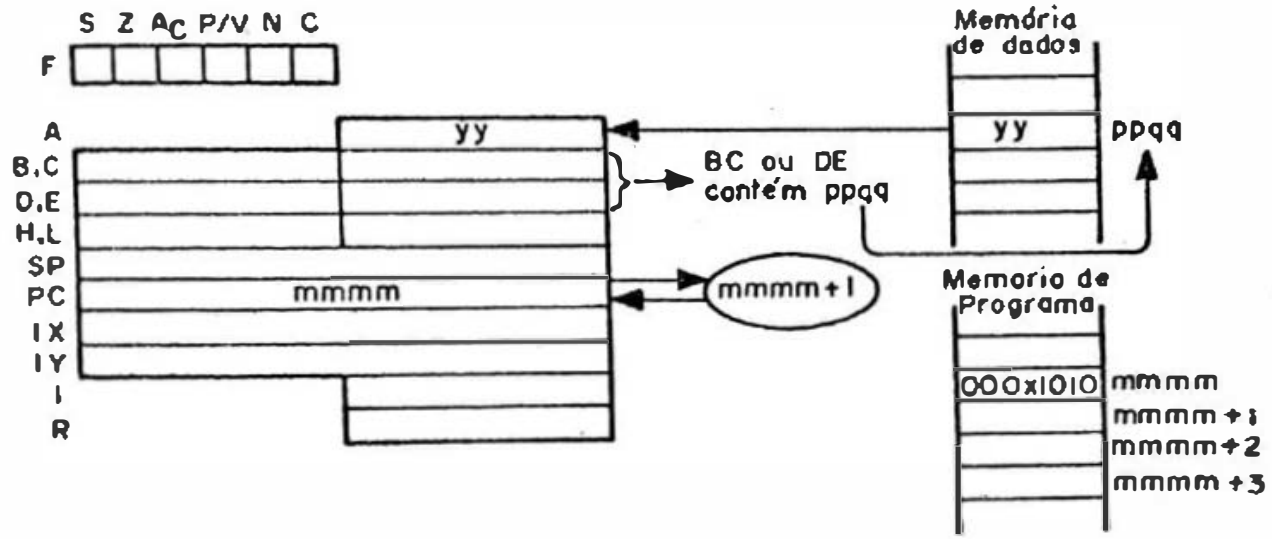
```
LD A,(label)
```

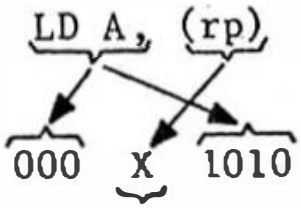
é equivalente às duas instruções:

```
LD HL,label
LD A,(HL)
```

A instrução LD A,(label) é muito utilizada para se carregar um valor simples da memória, pois ela usa uma instrução de três bytes de programa objeto para fazer o que a combinação LD HL, label ou LD A,(HL) faz em duas instruções e quatro bytes de programa objeto. Cabe observar também que a combinação LD HL, label e LD A,(HL) utiliza os registradores H e L, enquanto que a instrução LD A,(label) não os utiliza.

60) LD A,(rp) - CARREGUE O ACUMULADOR COM O CONTEÚDO DA MEMÓRIA ENDEREÇADA PELO PAR DE REGISTRADORES.





0 se o par de registradores for= BC
1 se o par de registradores for= DE

Formada por um único byte, o qual constitui o código de operação, esta instrução carrega o conteúdo do byte de memória (endereçados pelo par de registradores BC ou DE) para dentro do acumulador.

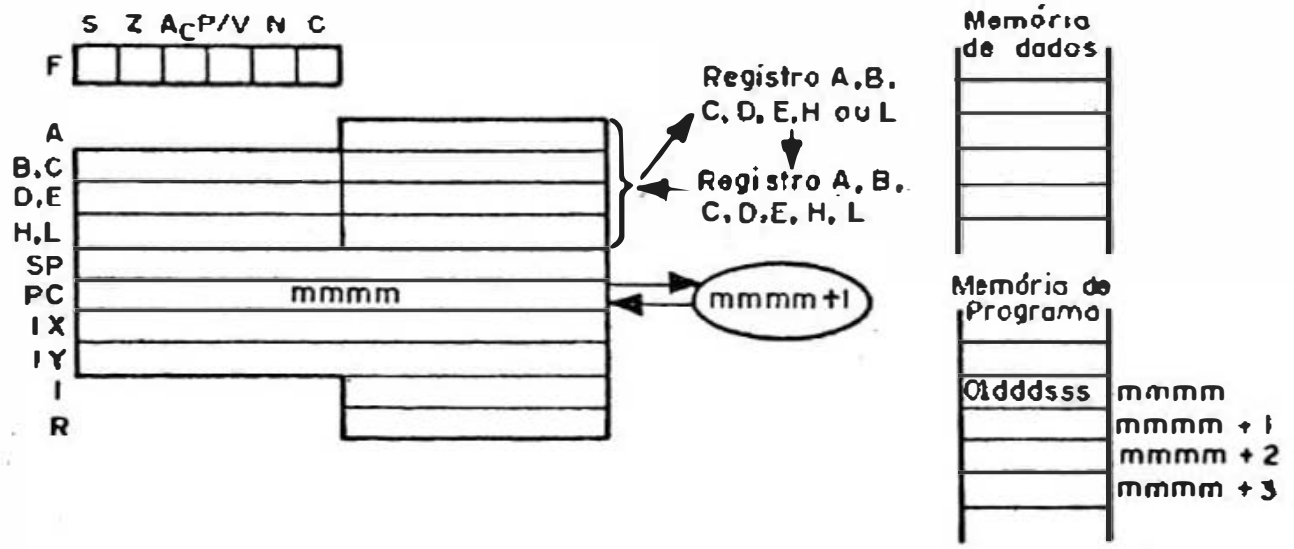
Suponhamos que o registrador B contenha 08_{16}, o registrador C contenha $4A_{16}$ e o byte de memória $084A_{16}$ contenha $3A_{16}$. Após a execução da instrução LD A,(BC), o acumulador passa a conter $3A_{16}$.

Normalmente,as instruções "LD A,(rp)" e "LD rp,data" pode

rão ser usadas juntas, desde que a instrução "LD rp,data" carregue um endereço de 16 bits para dentro dos registradores BC ou DE, como segue:

```
LD BC, 084AH
LD A, (BC)
```

61) LD dst, src - MOVA O CONTEÚDO DO REGISTRADOR FONTE PARA O REGISTRADOR DE DESTINO.



LD dst , src

01 ddd , sss

000 para dst ou src = B
001 para dst ou src = C
010 para dst ou src = D
011 para dst ou src = E
100 para dst ou src = H
101 para dst ou src = L
111 para dst ou src = A

Esta instrução carrega o conteúdo de qualquer registrador designado em qualquer outro registrador. Por exemplo:

```
LD A,B
```

A instrução LD A,B carrega o conteúdo do registrador B no registrador A.

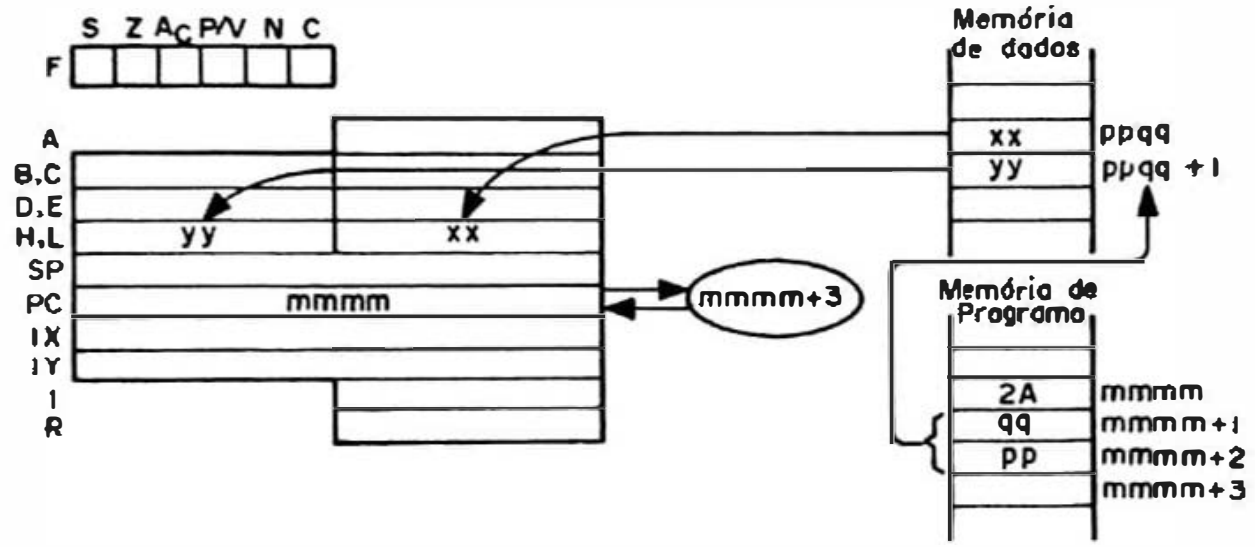
```
LD L,D
```

A instrução LD L,D, por sua vez, carrega o conteúdo do registrador D para o registrador L.

LD C,C

Por fim, a instrução LD C,C, nada faz, uma vez que o registrador C é especificado como destino.

62) LD HL,(addr) - CARREGUE O PAR DE REGISTRADORES OU O
LD rp,(addr) REGISTRADOR ÍNDICE DA MEMÓRIA USANDO
LD Ix,(addr) O ENDEREÇAMENTO DIRETO.
LD Iy,(addr)

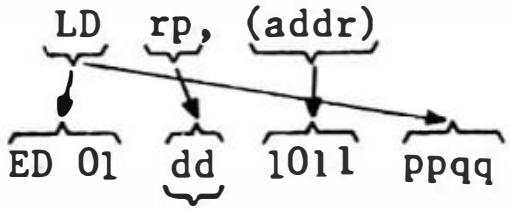


A ilustração nos mostra a execução da instrução LD HL (ppqq):

$$\underbrace{\text{LD HL}}_{\text{2A}} \text{ , } \underbrace{\text{addr}}_{\text{ppqq}}$$

A instrução LD HL,addr carrega o par de registradores HL com o conteúdo da localização de memória diretamente endereçada (addr). Esta instrução é formada por três bytes, sendo o primeiro o código de operação e os dois últimos indicam um endereço de memória.

Suponhamos que o conteúdo da localização de memória 4004_{16} contenha o valor AD_{16} e a localização de memória 4005_{16} contenha 12_{16}. Após a execução da instrução LD HL (4004H), o par de registradores passa a conter o valor $12AD_{16}$.



00 para rp igual ao par de registradores BC
01 para rp igual ao par de registradores DE
10 para rp igual ao par de registradores HL
11 para rp igual ao Stack-Pointer.

Carregue o par de registradores com o conteúdo da localização de memória diretamente endereçada.

Suponhamos, agora, que a localização de memória $49FF_{16}$ contenha o valor BE_{16} e a localização de memória $4A00_{16}$ contenha o valor 33_{16}. Após a execução da instrução LD DE,(49FFH), o par de registradores DE passa a conter o valor $33BE_{16}$.

LD Ix, (addr)
DD 2A ppqq

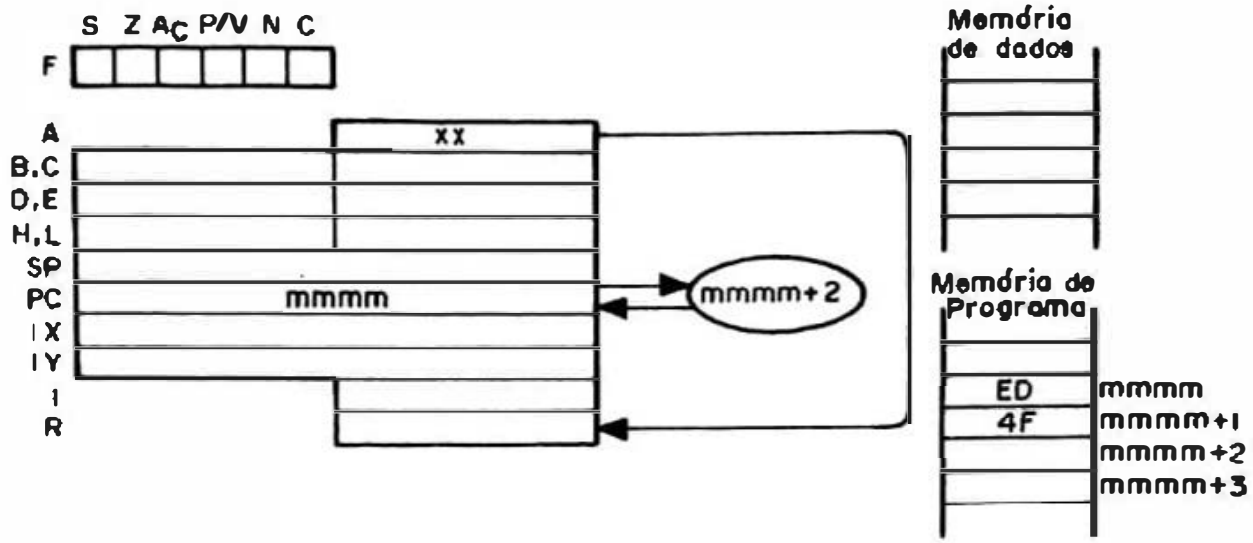
Carregue o registrador de índice Ix com o conteúdo da localização de memória diretamente endereçada.

Suponhamos, agora, que a localização de memória $D111_{16}$ contenha o valor FF_{16}, enquanto que a localização de memória $D112_{16}$ contém 56_{16}. Após a execução da instrução LD Ix,(D111H), o registrador Ix passa a conter o valor $56FF_{16}$.

LD Iy, (addr)
FD 2A ppqq

Semelhante à instrução anterior, esta instrução carrega o registrador Iy com o conteúdo da localização de memória diretamente endereçada. Difere da instrução anterior pelo fato de afetar o registrador Iy em vez do registrador Ix.

63) LD I,A - CARREGUE O REGISTRADOR VETOR DE INTERRUPÇÃO
LD R,A OU REFRESH (I OU R) COM O CONTEÚDO DO ACUMULADOR.



A ilustração nos mostra a execução da instrução LD R,A:

$$\underbrace{\text{LD R,A}}$$
$$\text{ED 4F}$$

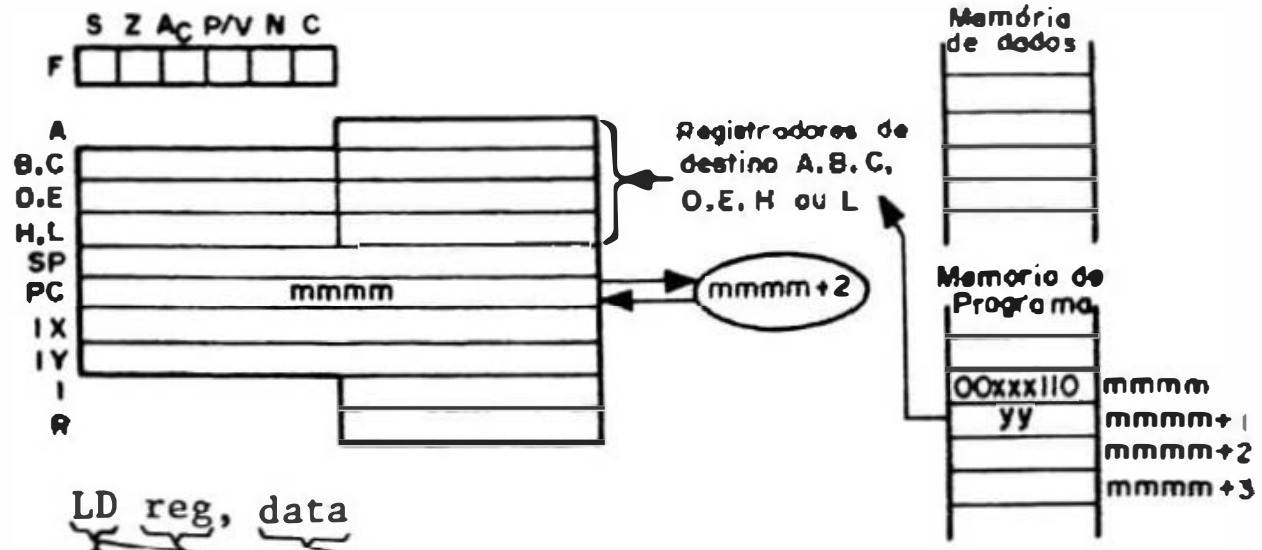
Formada por dois bytes, os quais constituem o código de operação, esta instrução carrega no registrador Refresh (R) o conteúdo do acumulador.

Suponhamos que o acumulador contenha o valor $7F_{16}$. Após a execução da instrução LD R,A, o conteúdo do registrador Refresh passa a ser igual a $7F_{16}$.

$$\underbrace{\text{LD I,A}}$$
$$\text{ED 47}$$

Semelhante à instrução anterior, com a exceção de que ela afeta o registrador vetor de Interrupção ao invés do registrador Refresh. Esta instrução carrega o registrador vetor de Interrupção com o conteúdo do acumulador.

64) LD reg, data - CARREGUE IMEDIATAMENTE PARA DENTRO DO REGISTRADOR.

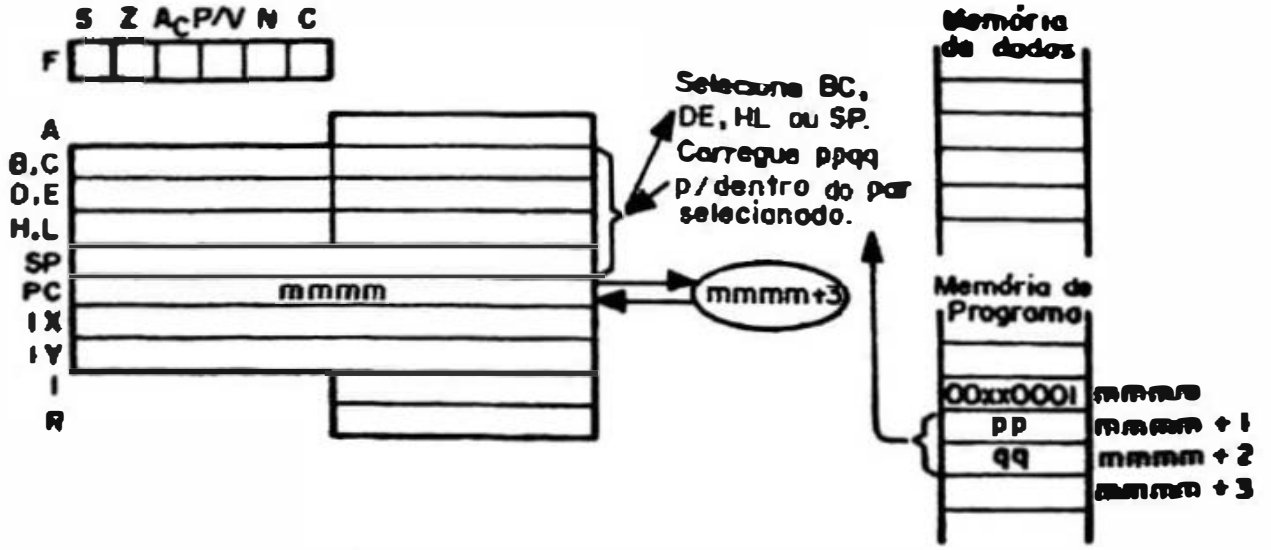


000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

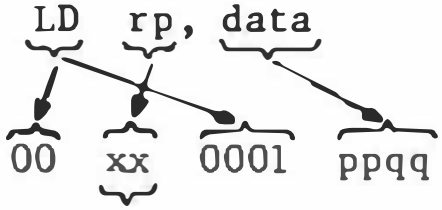
Constituída por dois bytes, sendo o primeiro byte o código de operação e o segundo byte, um byte de dados, esta instrução carrega o conteúdo do segundo byte do código objeto no registrador indicado.

Por exemplo, quando a instrução LD A,2AH é executada, o valor 2AH é carregado no acumulador (A).

65) LD rp,data - CARREGUE IMEDIATAMENTE PARA DENTRO DO REGISTRADOR OS 16 BITS DE DADOS.
LD Ix,data
LD Iy,data



A ilustração nos mostra a execução da instrução LD rp, data:



00 para rp igual ao par de registradores BC
01 para rp igual ao par de registradores DE
10 para rp igual ao par de registradores HL
11 para rp igual ao Stack-Pointer.

Constituída por três bytes, sendo o primeiro o código de operação e os dois últimos os bytes de dados. Esta instrução carrega o conteúdo do segundo e terceiro byte para dentro do par de registradores selecionados.

Por exemplo, depois de executada a instrução LD SP,217AH, o Stack-Pointer (SP) passa a conter o valor $217A_{16}$.

LD Ix , data
DD 21 ppqq

A instrução LD Ix,data, carrega o conteúdo do segundo e terceiro byte do código objeto no registrador de índice Ix.

LD Iy , data
FD 21 ppqq

A instrução LD Iy,data, carrega o conteúdo do segundo e terceiro byte do código objeto no registrador de índice Iy.

Cabe observar que a instrução LD rp,data é equivalente à duas instruções LD reg,data. Por exemplo:

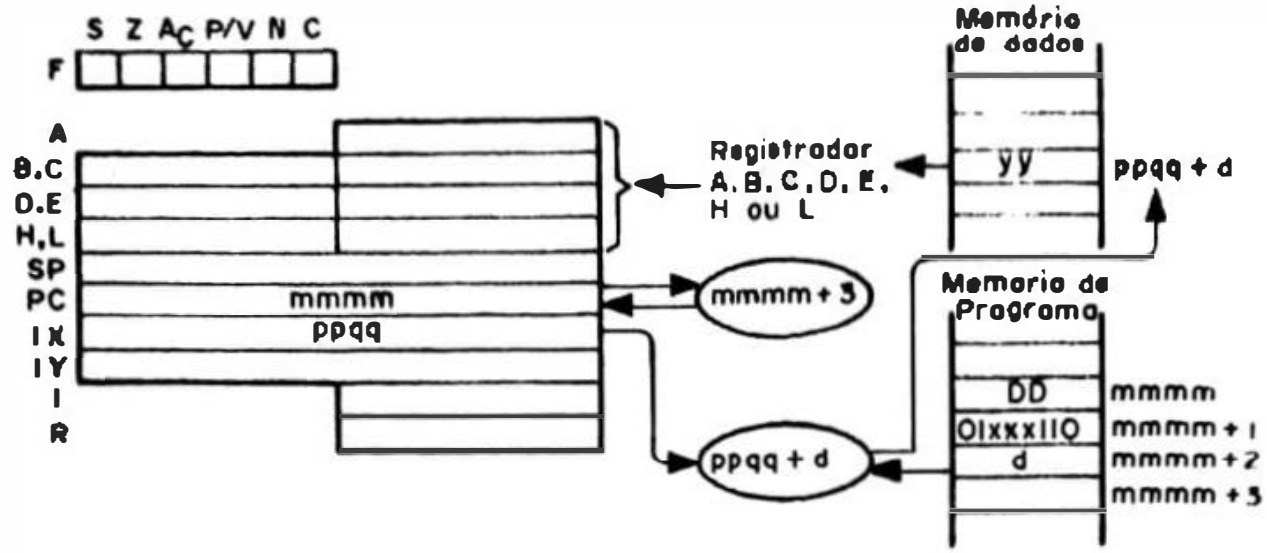
```
LD HL,032AH
```

é equivalente a:

```
LD H,03H
LD L,2AH
```

66) LD reg,(HL)
LD reg,(Ix+disp)
LD reg,(Iy+disp)

- CARREGUE O REGISTRADOR COM O CONTEÚDO DA MEMÓRIA ENDEREÇADA.



A ilustração nos mostra a execução da instrução LD reg, (Ix + disp):

LD reg, (Ix + disp)

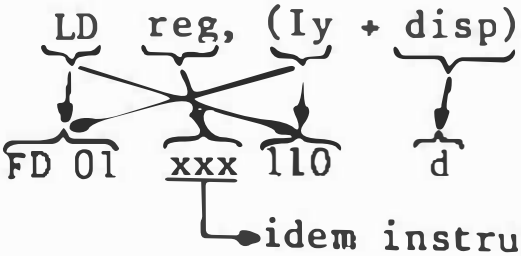
DD 01 xxx 110 d

000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

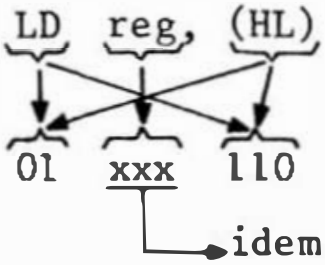
Esta instrução é constituída por três bytes, sendo os dois primeiros o código de operação e o terceiro byte indica uma posição de memória.

A instrução LD reg,(Ix+disp) carrega o conteúdo da posição de memória, endereçada pelo registrador de índice Ix, mais o deslocamento "d", que será somado ou subtraído do registrador de índice, no registrador especificado (reg.).

Suponhamos que ppqq = 4004_{16} e que a localização de memória 4010_{16} contenha o valor FF_{16}. Após a execução da instrução LD B, (Ix + 0CH), o registrador B passa a conter o valor FF_{16}.

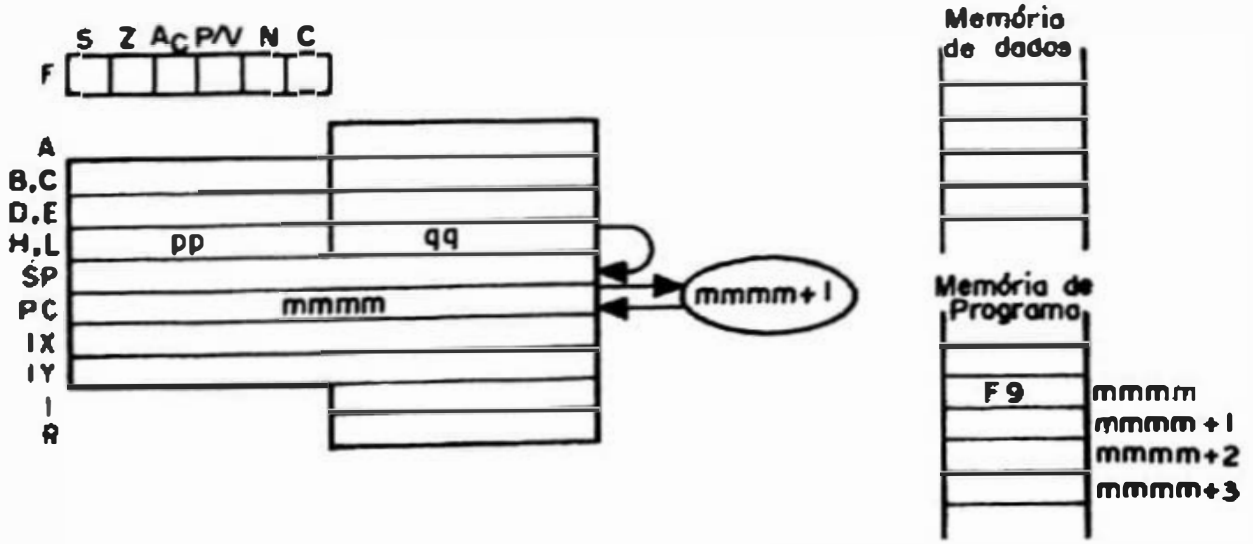


A instrução LD reg,(Iy + disp) é idêntica à instrução LD reg,(Ix + disp), exceto pelo fato dela usar o registrador Iy em vez do registrador Ix.



A instrução LD reg,(HL) carrega o registrador especificado (reg.) com o valor contido na posição de memória endereçada pelo par de registradores "HL".

67) LD SP,HL - MOVA O CONTEÚDO DO REGISTRADOR HL OU REGISTRADOR ÍNDICE PARA O STACK-POINTER (SP).
LD SP,Ix
LD SP,Iy



A ilustração nos mostra a instrução LD SP,HL:

LD SP,HL
F9

Constituída por um único byte, sendo este o código de operação. Esta instrução carrega o conteúdo do par de registradores "HL" no registrador Stack-Pointer (SP).

Suponhamos, por exemplo, que pp = 08_{16} e qq = $3F_{16}$. Após a execução da instrução LD SP,HL, o conteúdo do Stack-Pointer (SP) passa a ser $083F_{16}$.

LD SP,Ix

DD F9

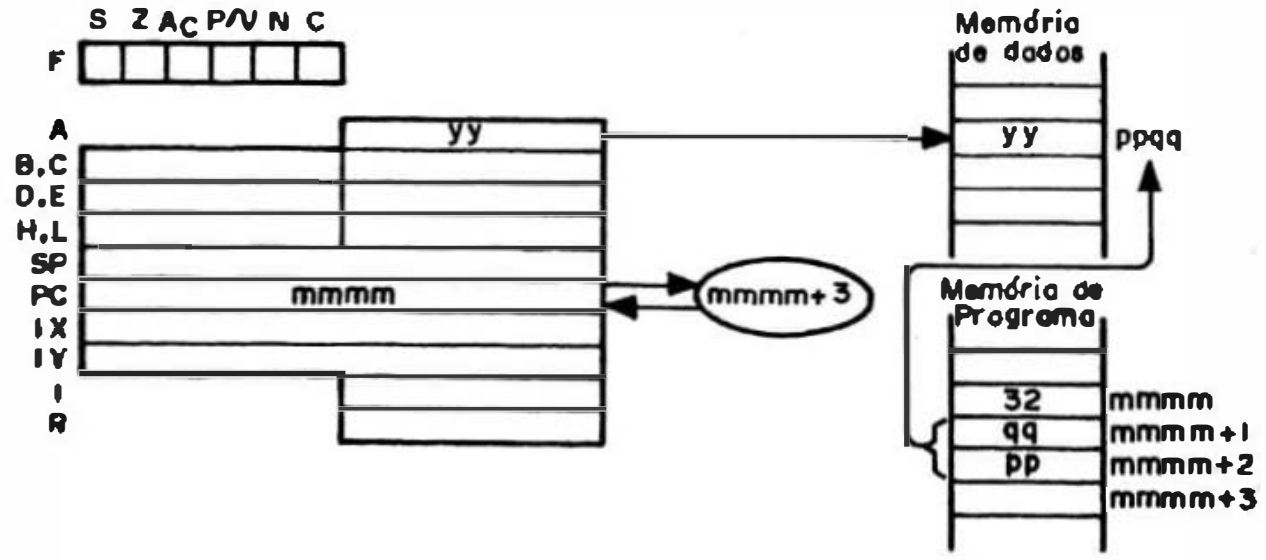
A instrução LD SP,Ix é constituída por dois bytes, sendo estes o código de operação. Esta instrução carrega o conteúdo do registrador de índice Ix no Stack-Pointer.

LD SP,Iy

FD F9

A instrução LD SP,Iy é idêntica à instrução LD SP,Ix, exceto pelo fato da primeira utilizar o registrador Iy em vez do registrador Ix.

68) LD (addr),A - ARMAZENE O CONTEÚDO DO ACUMULADOR NA POSIÇÃO DE MEMÓRIA USANDO ENDEREÇAMENTO DIRETO.



LD (addr),A

32 ppqq

Esta instrução é constituída por três bytes, o primeiro é o código de operação e os dois últimos indicam uma posição de memória. Deste modo, esta instrução armazena o conteúdo do acumulador em uma posição de memória, endereçada pelo segundo e terceiro byte do código objeto da instrução LD (addr),A.

Suponhamos que o acumulador contenha $3A_{16}$. Após a execução da instrução,

```
label  EQU  084AH
       =
       =
       LD   (label),A
```

o byte de memória 084AH irá conter $3A_{16}$.

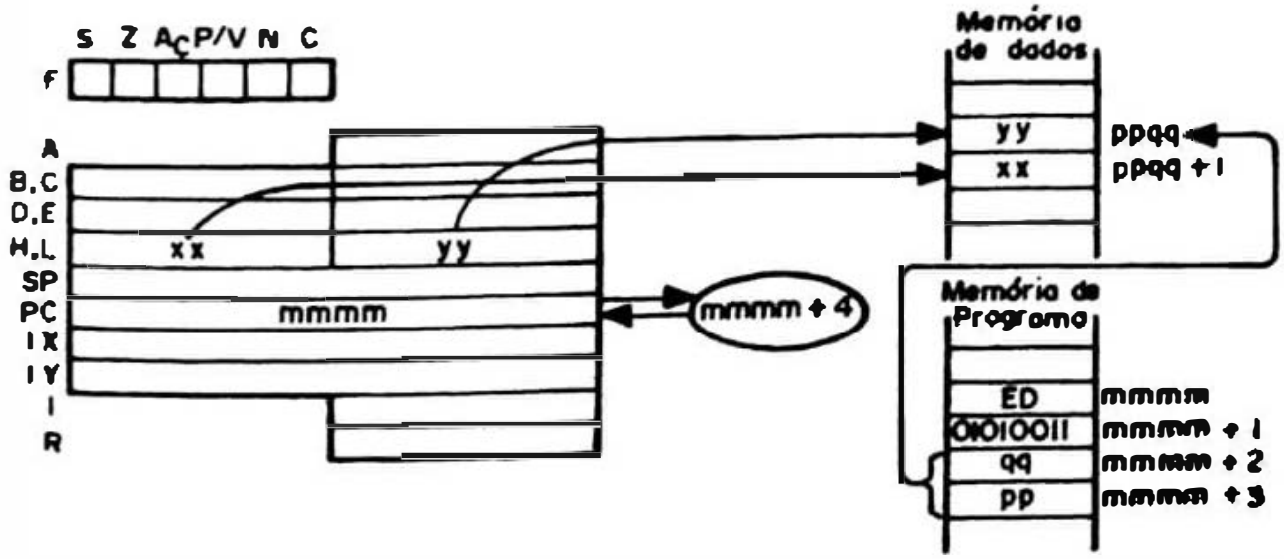
Lembre-se que EQU é um diretivo de Assembler.

A instrução LD (addr),A é equivalente a duas instruções:

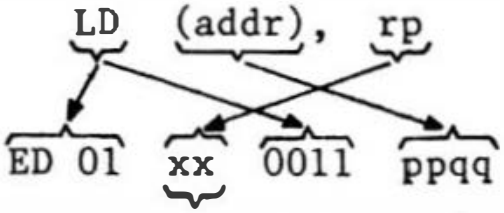
```
LD  H,label
LD  (HL),A
```

Para armazenar um simples valor de dados, a instrução LD (label),A é preferida porque ela usa três bytes do programa objeto para fazer o que a combinação LD H,(label), LD (HL),A faz, utilizando 4 bytes do programa objeto. Além disto, esta combinação utiliza os registradores H e L, enquanto que a instrução LD (label),A não os utiliza.

69) LD (addr),HL - ARMAZENE O CONTEÚDO DO PAR DE REGISTRADORES HL OU DOS REGISTRADORES INDEXADOS NA POSIÇÃO DE MEMÓRIA USANDO ENDEREÇAMENTO DIRETO.
LD (addr),rp
LD (addr),xy



A ilustração nos mostra a execução da instrução LD (ppqq), DE:



00 para rp igual ao par de registradores BC
01 para rp igual ao par de registradores DE
10 para rp igual ao par de registradores HL
11 para rp igual ao Stack-Pointer.

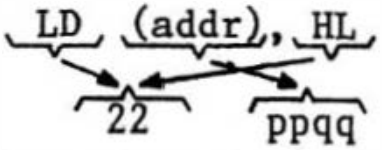
A instrução LD (ppqq),rp representada pela instrução LD (ppqq),DE é constituída por quatro bytes dos quais os dois primeiros representam o código de operação, enquanto que os dois últimos indicam uma posição de memória. Esta instrução armazena o conteúdo do par de registradores, especificado na localização de memória. O terceiro e quarto bytes do código objeto dão o endereço da localização de memória, onde o byte de baixa ordem deve ser escrito. O byte de alta ordem, por sua vez, deve ser escrito na localização de memória seqüencial.

Suponhamos que o par de registradores BC contenha $3C2A_{16}$. Após a execução da instrução:

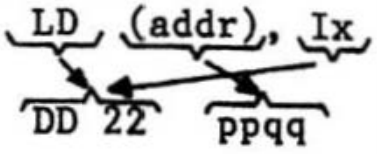
```
label  EQU  084AH
=
=
LD (label) , BC
```

O byte $084A_{16}$ da memória passa a conter $2A_{16}$ e o byte de memória $084B_{16}$ a conter $3C_{16}$.

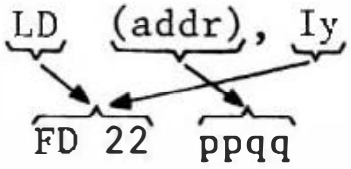
Lembre-se de que EQU é um diretivo de Assembler e é mais do que uma instrução, pois ele diz ao Assembler para usar o valor de 16 bits $084A_{16}$, onde a palavra "label" aparecer.



A instrução LD (addr),HL é uma versão de três bytes da instrução LD (addr),rp, onde o par de registradores HL é o par de registrador fonte.



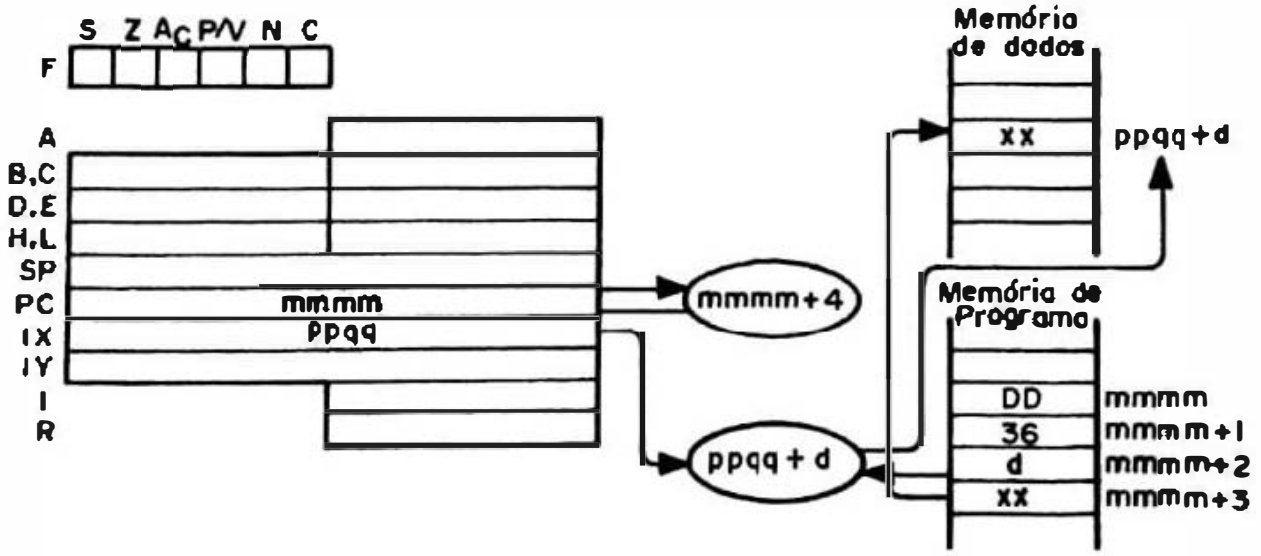
A instrução LD (addr),Ix também é formada por quatro bytes, dos quais os dois primeiros representam o código de operação e os dois últimos indicam uma posição de memória. Esta instrução armazena o conteúdo do registrador índice Ix na posição de memória endereçada. O terceiro e quarto bytes do código objeto nos fornecem o endereço da localização de memória, onde o byte de baixa ordem deve ser escrito. Já, o byte de alta ordem é escrito na próxima localização de memória seqüencial.



A instrução LD (addr),Iy é idêntica à instrução LD(addr), Ix, com a única diferença de que ela utiliza o registrador Iy em vez do registrador Ix.

70) LD (HL),data
LD (Ix+disp),data
LD (Iy+disp),data

- CARREGUE IMEDIATAMENTE PARA DENTRO DA MEMÓRIA.



A ilustração nos mostra a execução da instrução LD (Ix + disp),data:

LD (Ix + disp), data
DD 36 d xx

Esta instrução é formada por quatro bytes: os dois primeiros constituem o código de operação, o terceiro indica uma posição de memória junto com o registrador de índice Ix e o quarto indica justamente o valor que será carregado na memória. Assim, esta instrução está capacitada para carregar imediatamente um dado (xx) na localização de memória, projetada por endereçamento relativo de base.

Suponhamos que ppqq = 5400_{16}. Após a execução da instrução LD (Ix + 9),FAH, a localização de memória 5409_{16} conterá o valor FA_{16}.

$$\underbrace{\text{LD (Iy}}_{\text{FD 36}} + \underbrace{\text{disp)}}_{\text{d}}, \underbrace{\text{data}}_{\text{xx}}$$

Esta instrução, LD (Iy+disp),data é idêntica à instrução LD (Ix+disp),data, com exceção de que ela usa o registrador Iy em vez do registrador Ix.

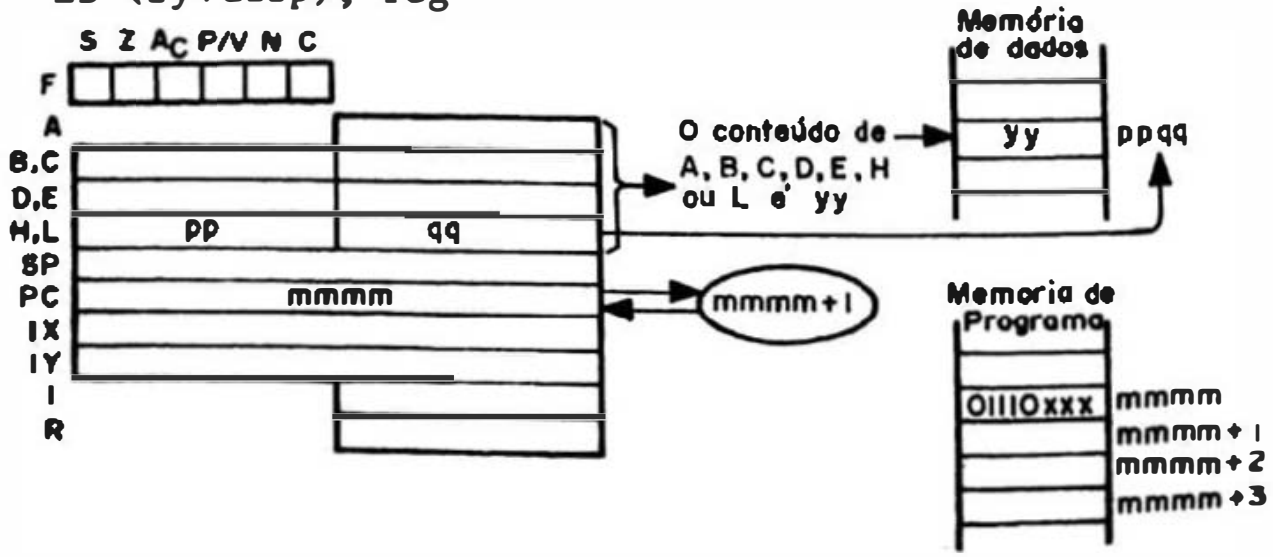
$$\underbrace{\text{LD (HL)}}_{\text{36}}, \underbrace{\text{data}}_{\text{xx}}$$

A instrução LD (HL),data, é formada por dois bytes, sendo o primeiro o código de operação e o segundo um byte de dados. Esta instrução carrega imediatamente um dado xx na localização de memória especificada pelo conteúdo do par de registradores HL.

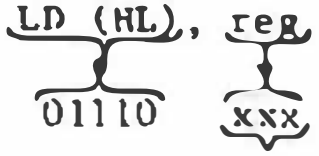
As instruções: CARREGUE IMEDIATAMENTE NA MEMÓRIA são muito menos utilizadas do que as instruções: CARREGUE IMEDIATAMENTE NO REGISTRADOR.

71) LD (HL), reg
LD (Ix+disp), reg
LD (Iy+disp), reg

- CARREGUE A MEMÓRIA ESPECIFICADA PELO REGISTRADOR.



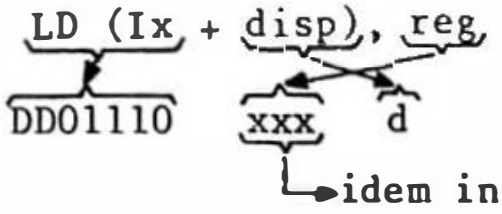
A ilustração nos mostra a execução da instrução LD (HL), reg:



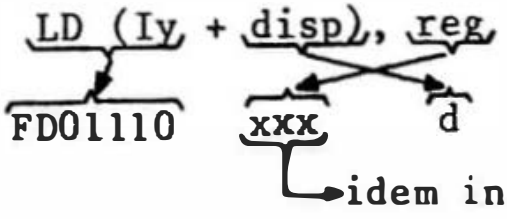
000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

A instrução LD (HL),reg é formada por dois bytes, sendo estes o código de operação. Esta instrução carrega o conteúdo do registrador "reg" na posição de memória endereçada pelo par de registradores HL.

Suponhamos que ppqq = 4500_{16} e que o conteúdo do registrador C seja $F9_{16}$. Após a execução da instrução LD (HL),C, a localização de memória 4500_{16} passa a conter o valor $F9_{16}$.

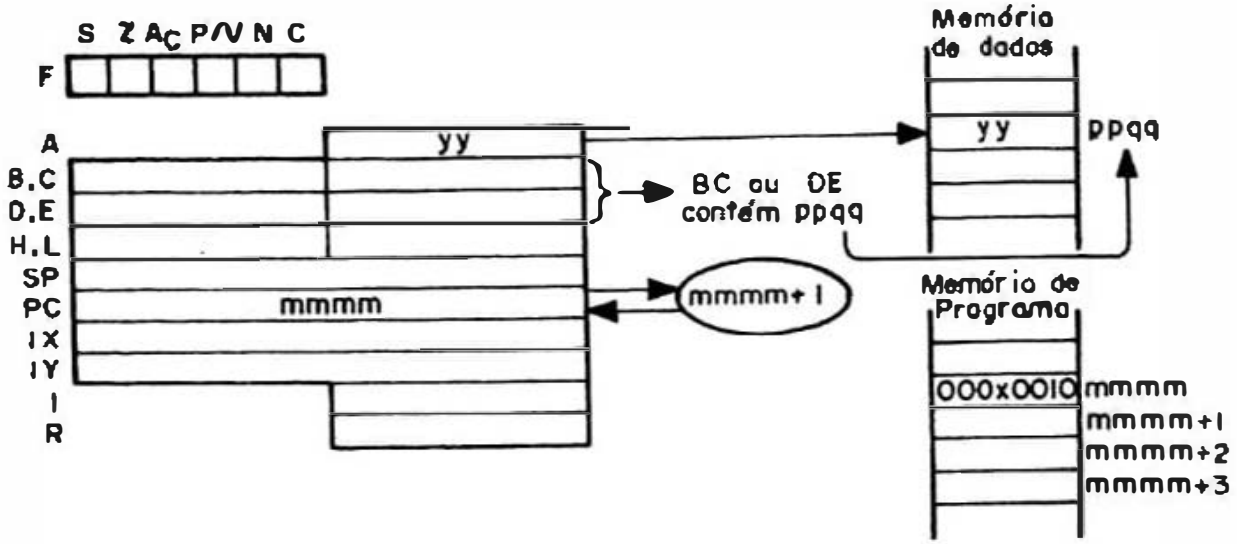


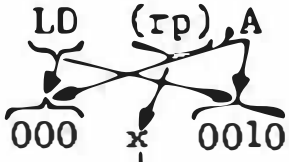
A instrução LD (Ix + disp), reg é formada por três bytes: os dois primeiros formam o código de operação, enquanto que o último indica uma posição de memória junto com o registrador de índice IX. Esta instrução carrega, na localização de memória especificada pela soma dos conteúdos do registrador Ix com o deslocamento d, o conteúdo do registrador "reg". especificado.



Esta instrução é idêntica à instrução LD (Ix + disp), reg, com a exceção de que ela utiliza o registrador Iy em vez do registrador Ix.

72) LD (rp),A - CARREGUE O CONTEÚDO DO ACUMULADOR NA LOCALIZAÇÃO DE MEMÓRIA ENDEREÇADA PELO PAR DE REGISTRADORES.





0 se o par de registradores = BC
1 se o par de registradores = DE

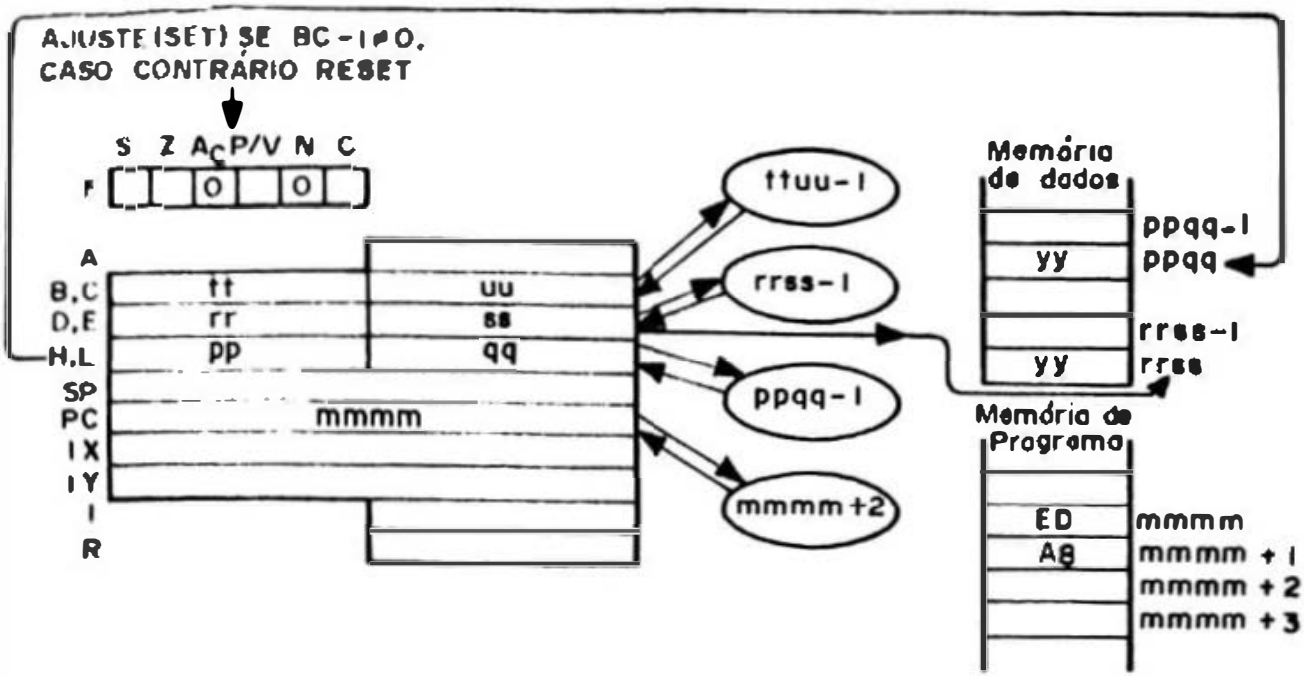
Formado por um único byte, o qual constitui o código de operação, esta instrução carrega (armazena) o conteúdo do acumulador no byte de memória, endereçado pelo par de registradores BC ou DE.

Suponhamos que o par de registradores BC contenha $084A_{16}$ e que o acumulador contenha $3A_{16}$. Após a execução da instrução LD (BC),A, o byte de memória $084A_{16}$ conterá $3A_{16}$.

As instruções LD (rp),A e LD rp,data serão, normalmente, usadas juntas, desde que a instrução LD rp, data carregue um endereço de 16 bits nos registradores BC ou DE, como segue:

```
LD BC, 084AH
LD (BC), A
```

73) LDD - TRANSFIRA DADOS ENTRE AS LOCALIZAÇÕES DE MEMÓRIA, DECREMENTE OS ENDEREÇOS DE DESTINO E FONTE.



LDD

ED A8

Formada por dois bytes, os quais constituem o código de operação, esta instrução transfere o byte de dados, contido na localização de memória endereçada pelo par de registradores-HL, para a localização de memória endereçada pelo par de registradores DE. Os pares de registradores BC, DE, HL, são decrementados de uma posição.

Suponhamos que o par de registradores BC contenha $004F_{16}$, DE contenha 4545_{16}, HL contenha 2012_{16} e a localização de memória 2012 contenha 18_{16}. Após a execução da instrução LDD, a localização de memória 4545_{16} passa a conter 18_{16} e os pares de registradores BC, DE e HL passam a conter os valores $004E_{16}$, 4544_{16} e 2011_{16}, respectivamente.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-29

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) A instrução EXX é constituída por:

a) um único byte;
b) dois bytes;
c) três bytes;
d) quatro bytes.

2) A instrução HALT, quando executada, faz com que a CPU entre:

a) num estado de contagem;
b) num estado de retenção;
c) num estado de memorização;
d) N.R.A.

3) A instrução INC reg:

a) incrementa o conteúdo dos flags;
b) incrementa o conteúdo da ALU;
c) incrementa o conteúdo dos registradores;
d) N.R.A.

4) A instrução INC (Ix + disp) incrementa o conteúdo da memó ria endereçada pelo registrador:

a) de índice Ix somado com o valor do deslocamento d;
b) de índice Iy somado com o valor do deslocamento d;
c) HL;
d) DE.

5) A instrução INDR é muito útil para:

a) efetuar operações de subtração;
b) efetuar operações de multiplicação;
c) carregar blocos de dados de um dispositivo de entrada para dentro da memória.
d) N.R.A.

6) Basicamente, na instrução IN reg,(C), quando reg for igual a 011, isto significa que o registrador especificado é o registrador:

a) A;
b) C;
c) H;
d) E.

7) A instrução "JP label" ordena que seja feito um:

a) retrocesso para a instrução imediatamente anterior;
b) salto para a instrução identificada no operando;
c) salto aleatório;
d) N.R.A.

8) Os dois bytes (escritos em hexadecimal) da instrução LD A, R, são respectivamente:

a) ED 57;
b) ED 5F;
c) ED 9E;
d) N.R.A.

9) A instrução LD C,C:

a) carrega o conteúdo do registrador B no registrador A;
b) carrega o conteúdo do registrador I no registrador R;
c) nada faz;
d) N.R.A.

10) A menos do registrador de índice, a instrução LD (addr),Iy é idêntica à instrução:

a) LD (HL),data;
b) LD (Iy + disp),data;
c) LD (addr),HL;
d) LD (addr),Ix.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MC-30

PROGRAMAÇÃO ASSEMBLER V

Nesta lição, daremos continuidade ao nosso estudo das instruções do microprocessador Z-80.

74) LDDR - TRANSFIRA DADOS ENTRE AS LOCALIZAÇÕES DE MEMÓRIA, ATÉ QUE O CONTADOR DE BYTES SEJA ZERO. DECREMENTE OS ENDEREÇOS DE DESTINO E FONTE.

LDDR
ED B8

Formada por dois bytes, os quais constituem o código de operação. Esta instrução carrega o conteúdo da posição de memória endereçada pelo par de registradores "HL" para a localização de memória endereçada pelo par de registradores "DE". Em síntese, podemos dizer que esta instrução é idêntica à instrução anterior, ou seja, LDD, com a exceção de que a instrução LDDR é repetida até que o par de registradores BC (contador de bytes) seja zero. Para isso, o conteúdo do par de registradores BC é decrementado de duas posições após cada transferencia.

Suponhamos que dispomos dos seguintes conteúdos na memória e pares de registradores:

REGISTRADORES/CONTEÚDOS		LOCALIZAÇÃO/CONTEÚDOS	
HL	2012_{16}	2012_{16}	18_{16}
DE	4545_{16}	2011_{16}	AA_{16}
BC	0003_{16}	2010_{16}	25_{16}

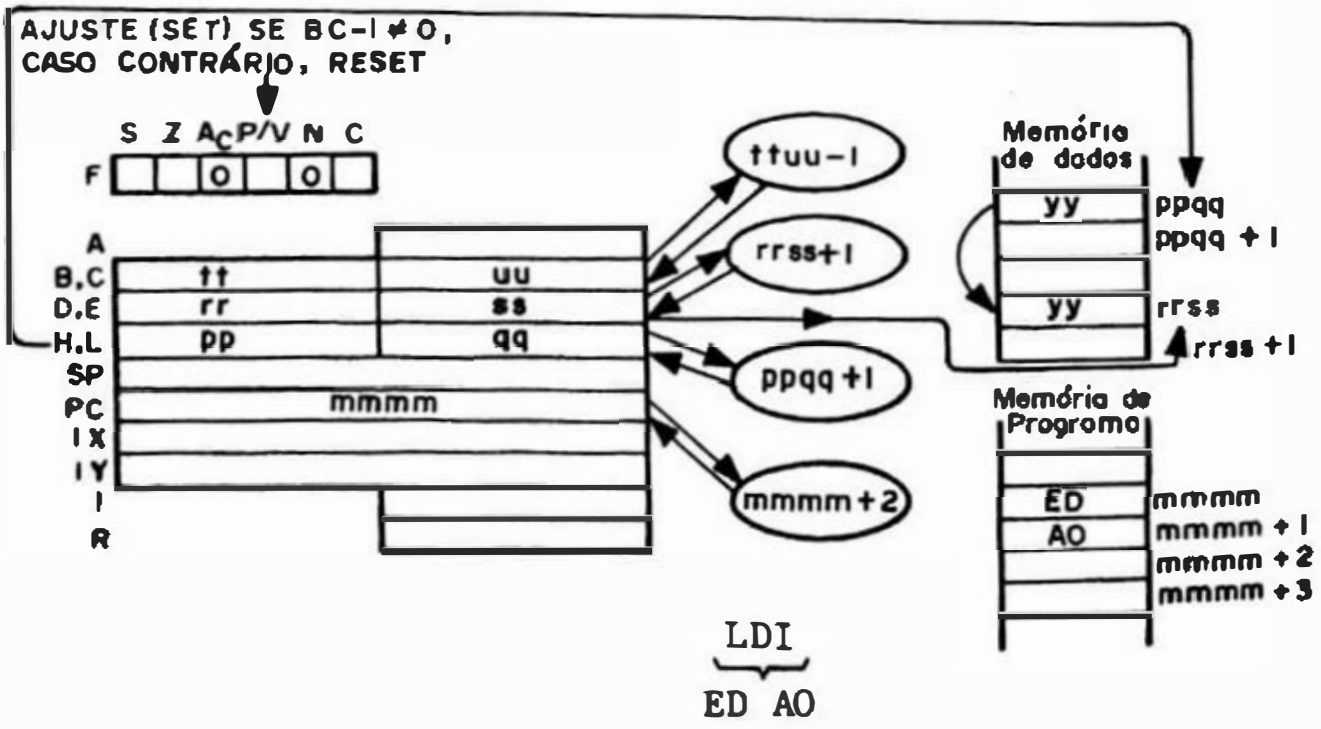
Após a execução da instrução LDDR, os pares de registradores e localizações de memória terão os seguintes conteúdos:

REGISTRADORES/CONTEÚDOS		LOCALIZAÇÃO/CONTEÚDOS	
HL	2009_{16}	2012_{16}	18_{16}
DE	4542_{16}	2011_{16}	AA_{16}
BC	0000_{16}	2010_{16}	25_{16}

LOCALIZAÇÃO/CONTEÚDOS	
4545_{16}	18_{16}
4544_{16}	AA_{16}
4543_{16}	25_{16}

Esta instrução é extremamente útil para transferir blocos de dados de uma área de memória para outra.

75) LDI - TRANSFIRA DADOS ENTRE LOCALIZAÇÕES DE MEMÓRIA. INCREMENTE OS ENDEREÇOS DE FONTE E DESTINO.



Formada por dois bytes, os quais constituem o código de operação. Esta instrução é semelhante à instrução LDD, diferindo desta última, devido ao fato dos registradores DE e HL serem incrementados, ao invés de serem decrementados. O par de registradores BC deve ser decrementado de uma posição.

Suponhamos que o par de registradores BC contenha o valor $004F_{16}$, o par DE contenha 4545_{16}, o par HL contenha 2012_{16} e que a localização de memória 2016_{16} contenha o valor 18_{16}. Após a execução da instrução LDI, a localização de memória 4545_{16} conterá 18_{16}, o par de registradores BC conterá $004E_{16}$ e o par DE conterá 2013_{16}.

76) LDIR - TRANSFIRA DADOS ENTRE AS LOCALIZAÇÕES DE MEMÓRIA, ATÉ QUE O CONTADOR DE BYTES SEJA ZERO. INCREMENTE OS ENDEREÇOS DE DESTINO E FONTE.

LDIR

ED B0

Formada por dois bytes, os quais constituem o código de operação. Esta instrução é idêntica à instrução LDI, com a única diferença de que ela é repetida várias vezes, até que o par de registradores BC (contador de bytes) seja zero. Para isso, o conteúdo do par de registradores BC é decrementado de duas posições após cada transferência.

Suponhamos que dispomos dos seguintes conteúdos na memória e pares de registradores:

REGISTRADORES/CONTEÚDOS		LOCALIZAÇÃO/CONTEÚDOS	
HL	2012_{16}	2012_{16}	18_{16}
DE	4545_{16}	2013_{16}	CD_{16}
BC	0003_{16}	2014_{16}	FO_{16}

Após a execução da instrução LDIR, os pares de registradores e localizações de memória terão os seguintes conteúdos:

REGISTRADORES/CONTEÚDOS		LOCALIZAÇÃO/CONTEÚDOS	
HL	2015_{16}	2012_{16}	18_{16}
DE	4548_{16}	2013_{16}	CD_{16}
BC	0000_{16}	2014_{16}	FO_{16}

LOCALIZAÇÃO/CONTEÚDOS

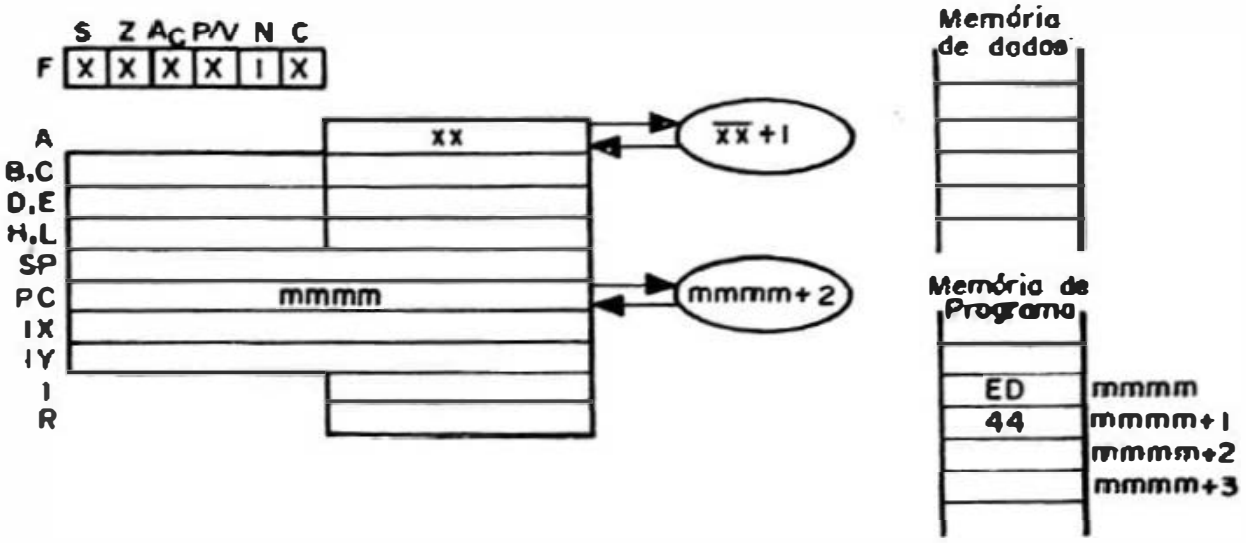
4545_{16} 18_{16}

4546_{16} CD_{16}

4547_{16} $F0_{16}$

Esta instrução é extremamente útil para transferir blocos de dados de uma área de memória para outra.

77) NEG - NEGUE O CONTEÚDO DO ACUMULADOR



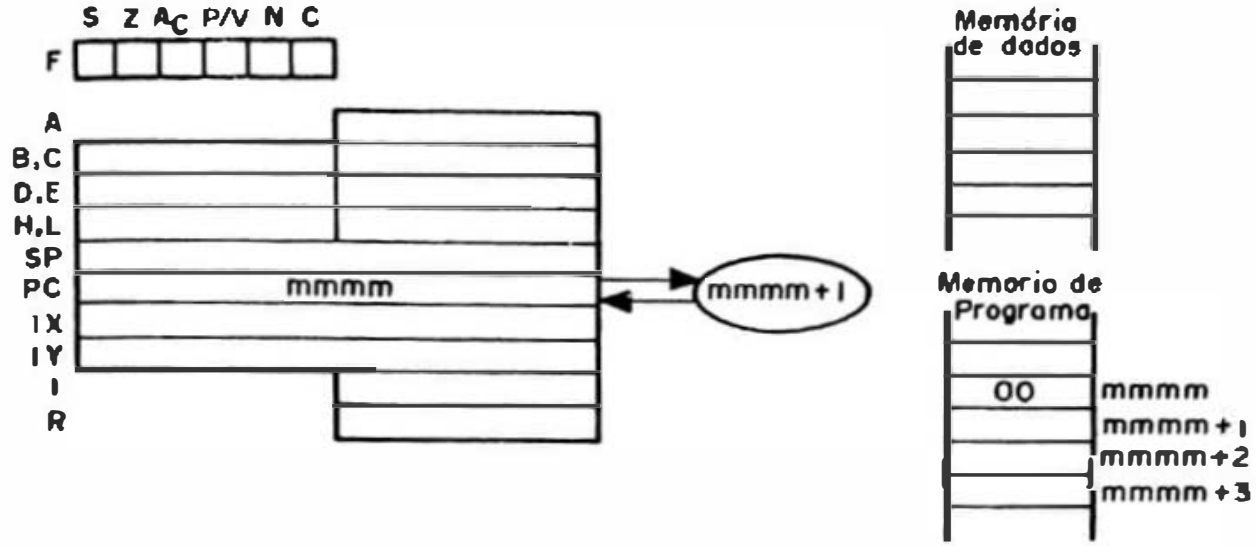
Formada por dois bytes, os quais constituem o código de operação. Esta instrução nega o conteúdo do acumulador, ou seja, subtrai o conteúdo do acumulador de zero, o que equivale a efetuar o complemento para dois (ou complemento de dois) do acumulador.

Suponhamos que xx = $5A_{16}$. Após a execução da instrução NEG, o conteúdo do acumulador passa a ser $A6_{16}$, veja:

5A = 01010 1010

complemento de dois = 1010 0110 = A6

78) NOP - SEM OPERAÇÃO.



NOP
00

Formada por apenas um byte, o qual constitui o código de operação. Esta instrução não realiza nenhuma operação, isto é, não altera os Flags, os registradores e posições de memória, com exceção do "contador de programa" que é incrementado e do refresh (refrescamento) de memória que continua operando. Esta instrução está presente por várias razões. Vejamos algumas delas:

a) Um erro de programa, que busca dados num código objeto de uma memória inexistente, buscará "00". É uma boa idéia assegurar que o erro de programa mais comum não fará nada.

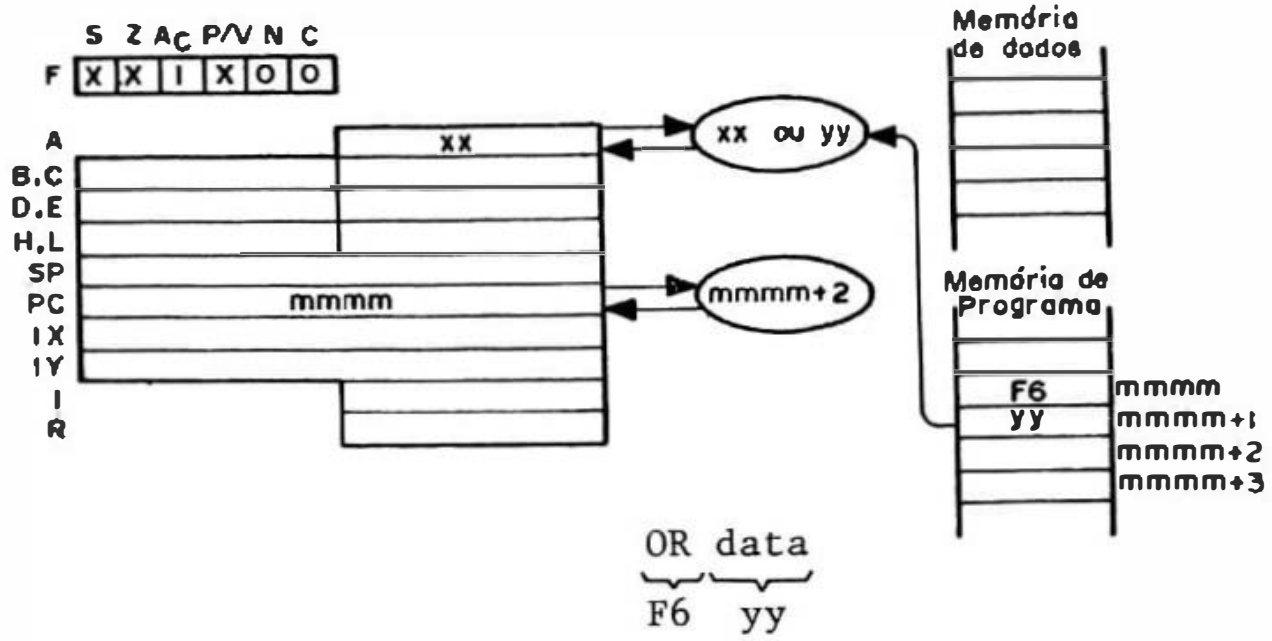
b) A instrução NOP permite que se dê um título a um byte de programa objeto:

HERE NOP

c) Para ajustar os tempos de atraso, cada instrução NOP soma quatro ciclos de clock a um atraso.

A instrução NOP não é muito utilizada.

79) OR DATA - OR IMEDIATO COM ACUMULADOR.



Constituída por dois bytes, sendo o primeiro o código de operação e o segundo um byte de dados. Esta instrução executa a operação lógica "OR" entre o conteúdo do acumulador e o conteúdo do segundo byte do código objeto da instrução. O resultado é armazenado no acumulador.

Suponhamos que xx = $3A_{16}$. Após a execução da instrução OR 7C H, o acumulador passa a conter $7E_{16}$. Veja:

```
3A = 0011 1010
7C = 0111 1100
     0111 1110
```

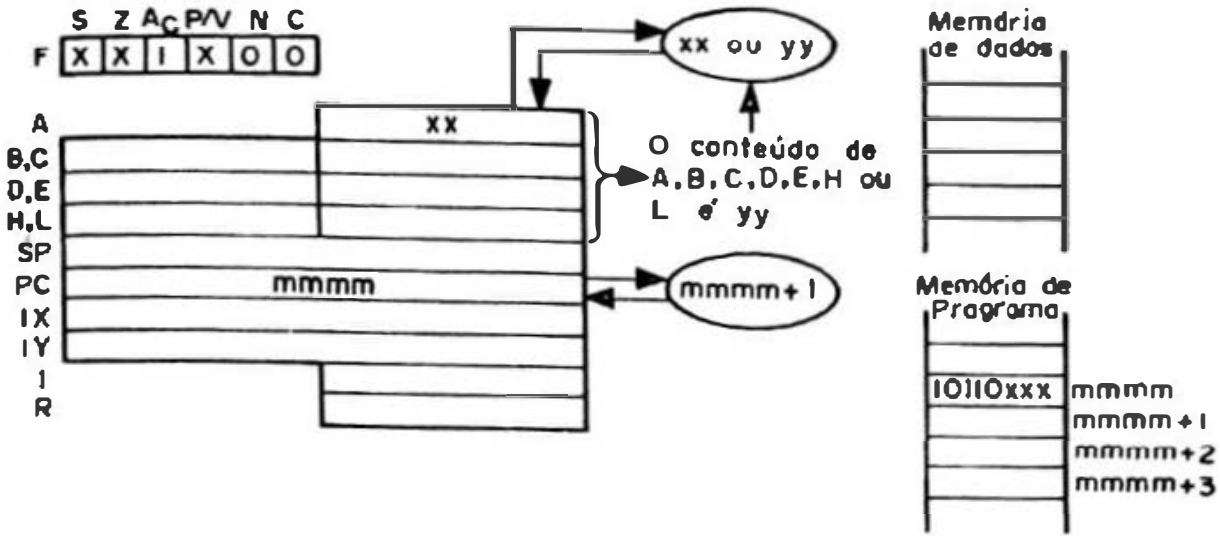
0, S é ajustado(set) para 0

Seis bits 1, P/V é ajustado (set) para 1.

Resultado Não-zero, Z é ajustado (set) para 0.

Esta é uma instrução lógica de rotina, freqüentemente usada para "ligar" os bits. Por exemplo, a instrução OR 80H irá, incondicionalmente, ajustar (set) o bit de alta ordem do acumulador para 1.

80) OR reg - EXECUTE A FUNÇÃO OR ENTRE O CONTEÚDO DO REGISTRADOR COM O CONTEÚDO DO ACUMULADOR.



OR reg

10110 xxx

000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por um único byte, o qual constitui o código de operação. Esta instrução executa a operação lógica OR entre o conteúdo do acumulador e o conteúdo do registrador reg. O resultado é armazenado no acumulador.

Suponhamos que xx = $E3_{16}$ e que o conteúdo do registrador E seja $A8_{16}$. Após a execução da instrução OR E, o acumulador passará a conter o valor EB_{16}. Veja:

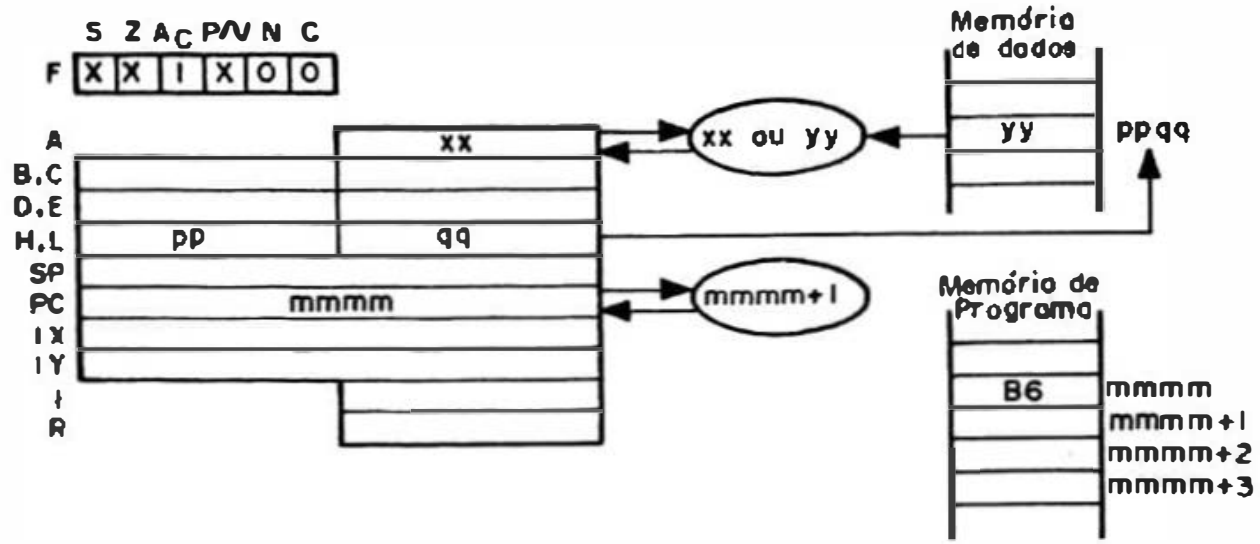
```
E3 = 1110 0011
A8 = 1010 1000
     1110 1011
```

1, S é ajustado (set) para 1

Seis bits, P/V é ajustado(set) para 1.

Resultado Não-zero, Z é ajustado para 0.

81) OR (HL) - EXECUTE A FUNÇÃO OR ENTRE O CONTEÚDO DA
OR (Ix+disp) LOCALIZAÇÃO DA MEMÓRIA E O CONTEÚDO DO
OR (Iy+disp) ACUMULADOR.



A ilustração nos mostra a execução da instrução OR (HL):

$$\underbrace{\text{OR (HL)}}_{\text{B6}}$$

A instrução OR (HL) é formada por um único byte, o qual constitui o código de operação. Esta instrução executa a operação lógica OR entre o conteúdo da localização de memória endereçada pelo par de registradores HL e o conteúdo do acumulador. O resultado é armazenado no acumulador.

Suponhamos que xx = $E3_{16}$, ppqq = 4000_{16} e que a localização de memória 4000_{16} contenha o valor $A8_{16}$. Após a execução da instrução OR (HL), o acumulador passara a conter o valor EB_{16}. Veja:

E3 = 1110 0011
A8 = 1010 1000
1110 1011

1, S é ajustado para 1.

Seis bits 1, P/V é ajustado para 1.

Resultado Não-zero, Z é ajustado para 0.

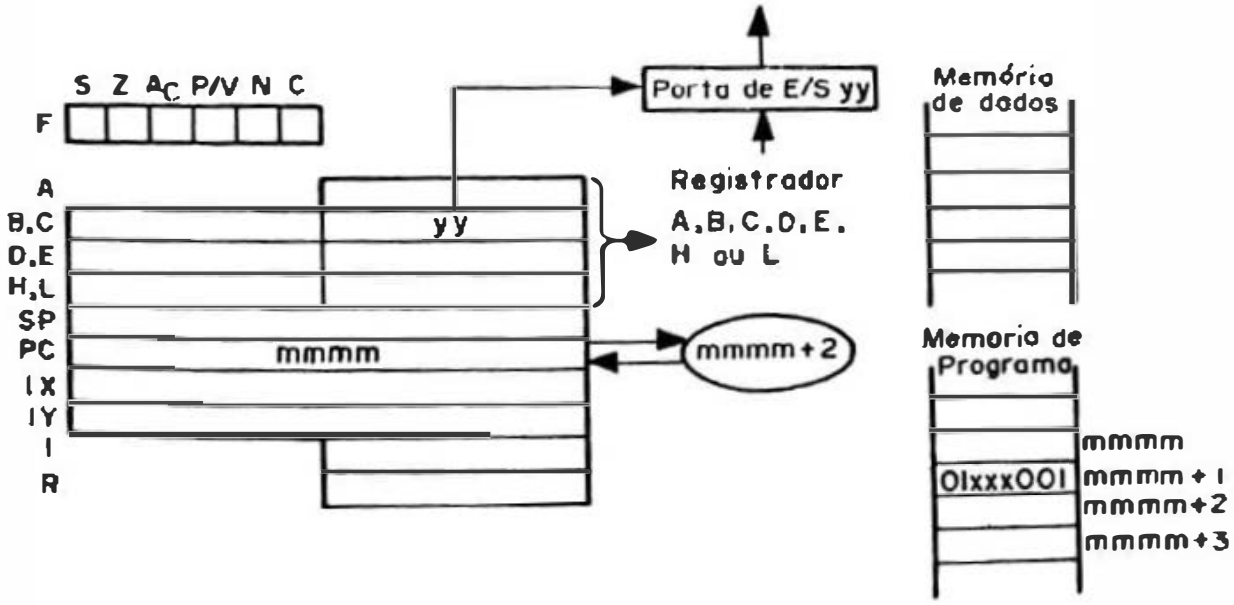
$$\underbrace{\text{OR (Ix}}_{\text{DD B6}} + \underbrace{\text{disp)}}_{\text{d}}$$

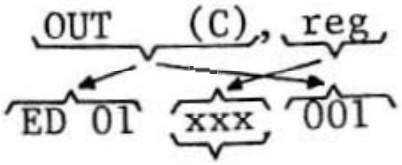
A instrução OR (Ix + disp) é formada por três bytes. Os dois primeiros constituem o código de operação e o terceiro, um byte de dados. Esta instrução executa a operação lógica OR entre o conteúdo do acumulador e o conteúdo da localização de memória, endereçada pelo registrador Ix e o valor de deslocamento disp (ou d).

$$\underbrace{\text{OR (Iy}}_{\text{FD B6}} + \underbrace{\text{disp)}}_{\text{d}}$$

A instrução OR (Iy + disp) é idêntica à instrução OR (Ix + disp), com a única diferença de que esta utiliza o registrador Iy, ao invés do registrador Ix.

82) OUT (C),reg - SAIA DO REGISTRADOR.



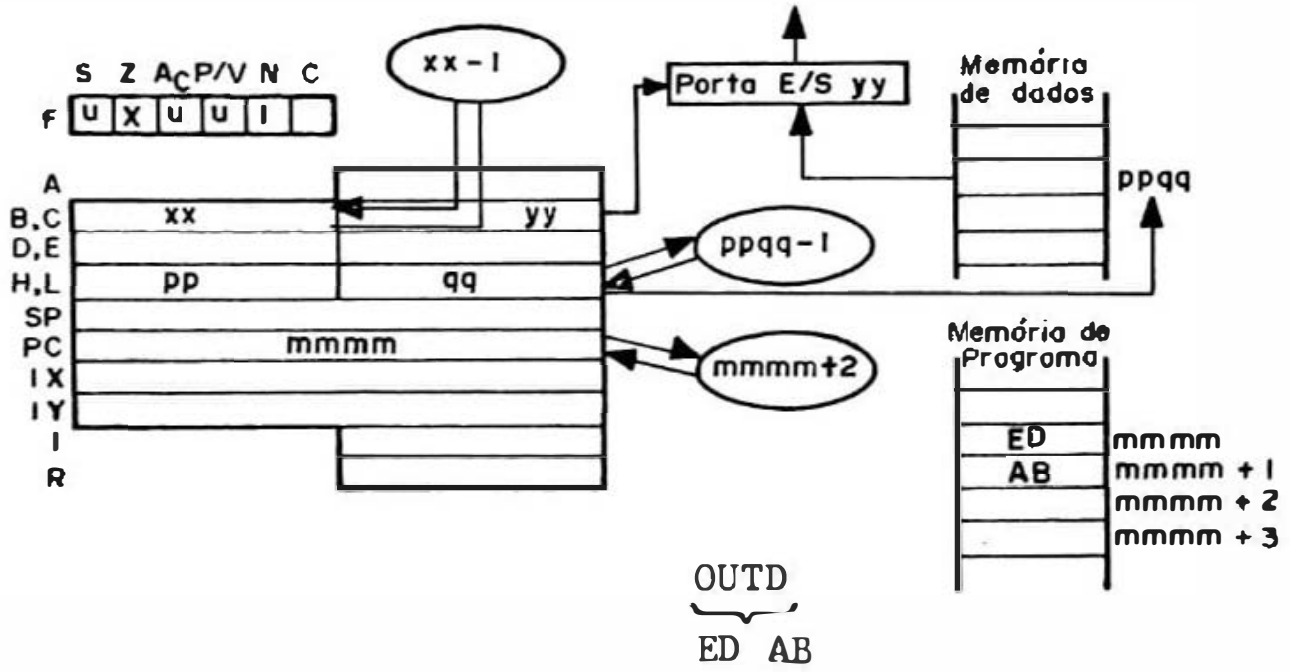


000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por dois bytes, os quais constituem o código de operação. Esta instrução coloca o conteúdo do registrador "reg" no dispositivo de entrada e saída (E/S), endereçado pelo registrador C.

Suponhamos que yy = $1F_{16}$ e que o conteúdo do registrador H seja AA_{16}. Após a execução da instrução OUT (C),H, o valor AA_{16} estará no buffer da porta de E/S $1F_{16}$.

83) OUTD - SAIA DA MEMÓRIA. DECREMENTE O ENDEREÇO.



A instrução OUTD é formada por dois bytes, os quais constituem o código de operação. Esta instrução carrega o conteúdo da localização de memória especificada (ou endereçada) pelo par de registradores HL, num dispositivo ou numa porta de E/S, endereçada pelo registrador C. O registrador B (utilizado como contador) é decrementado a cada execução desta instrução. Já, o par de registradores HL é decrementado de modo a permitir o carregamento de outros dados.

Suponhamos que xx = $0A_{16}$, yy = FF_{16}, ppqq = 5000_{16} e que a localização de memória 5000_{16} contenha o valor 77_{16}. Após a execução da instrução OUTD, o valor 77_{16} será carregado no buffer da porta de E/S FF_{16}. O registrador B irá conter 09_{16} e o par de registradores HL conterá $4FFF_{16}$.

84) OTDR - SAIA DA MEMÓRIA. DECREMENTE ENDEREÇO. CONTINUE ATÉ QUE O REGISTRADOR B SEJA ZERO.

OTDR

ED BB

Formada por dois bytes, os quais constituem o código de operação. Esta instrução é idêntica à instrução OUTD, porém é repetida até que o registrador B seja igual a zero, saindo assim do loop.

Suponhamos que o registrador B contenha o valor 03_{16}, o registrador C contenha FF_{16} e o par de registradores HL contenha 5000_{16}. E as localizações de memória $4FFE_{16}$ até 5000_{16} contenham:

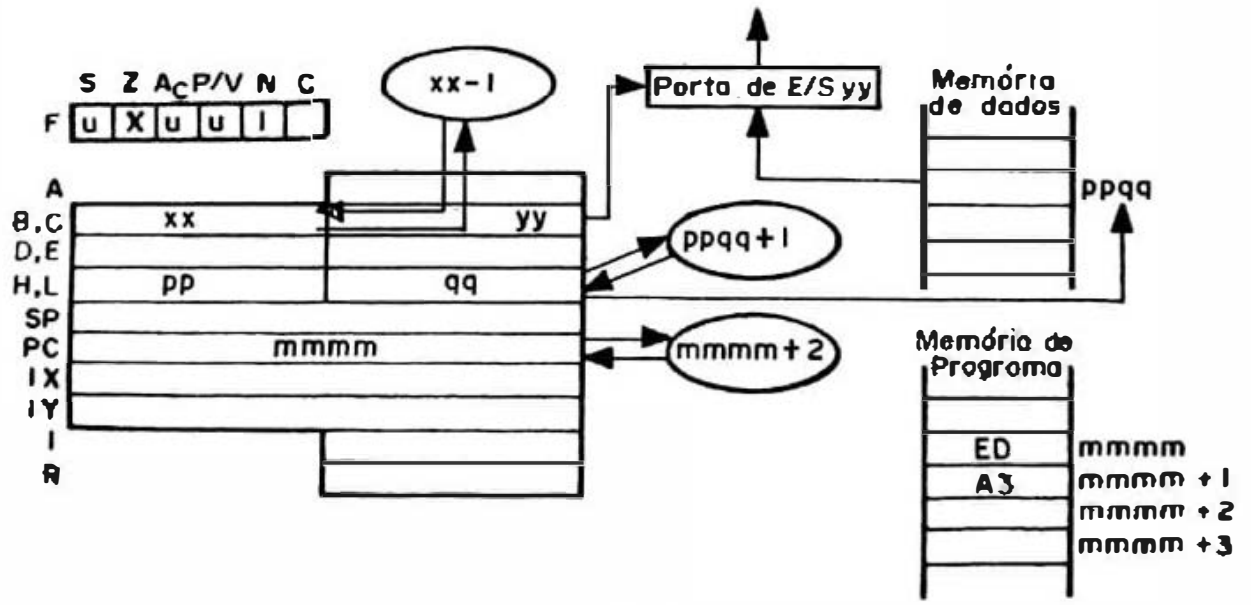
LOCALIZAÇÃO/CONTEÚDO

$4FFE_{16}$	CA_{16}
$4FFF_{16}$	$1B_{16}$
5000_{16}	$F1_{16}$

Após a execução da instrução OTDR, o par de registradores HL passará a conter $4FFD_{16}$, o registrador B conterá zero e a seqüência $F1_{16}$, $1B_{16}$ e CA_{16} deverá ter sido escrita na porta de E/S FF_{16}.

Esta instrução é muito útil para transferência de blocos de dados da memória para os dispositivos de saída.

85) OUTI - SAIA DA MEMÓRIA. INCREMENTE O ENDEREÇO.



OUTI

ED A3

Formada por dois bytes, os quais constituem o código de operação. Esta instrução carrega o conteúdo da localização de memória, endereçada pelo par de registradores HL, para a porta de E/S, endereçada pelo registrador C. O registrador B (utilizado como contador) é decrementado a cada execução desta instrução, ao passo que o par de registradores HL é incrementado. E é justamente este fato que difere a instrução OUTI da instrução OUTD.

Suponhamos que xx = $0A_{16}$, yy = FF_{16}, ppqq = 5000_{16} e que a localização de memória 5000_{16} contenha o valor 77_{16}. Após a execução da instrução OUTI, o valor 77_{16} deverá estar seguro (ou deverá ser escrito) no buffer da porta de E/S FF_{16}. O registrador B passará a conter 09_{16}, enquanto que o par de registradores HL conterá 5001_{16}.

86) OTIR - SAIA DA MEMÓRIA. INCREMENTE O ENDEREÇO, ATÉ QUE O CONTEÚDO DO REGISTRADOR B SEJA ZERO (B = 0).

OTIR
ED B3

Formada por dois bytes, os quais constituem o código de operação. A instrução OTIR é idêntica à instrução OUTI, com a única diferença de que esta entra em loop e só sairá do loop, quando o contador tiver chegado ao fim, ou seja, quando o registrador B for igual a zero.

Suponhamos que o registrador B contenha 04_{16}, o registrador C contenha FF_{16}, o par de registradores HL contenha 5000_{16} e que as localizações de memória de 5000_{16} até 5003_{16} contenham:

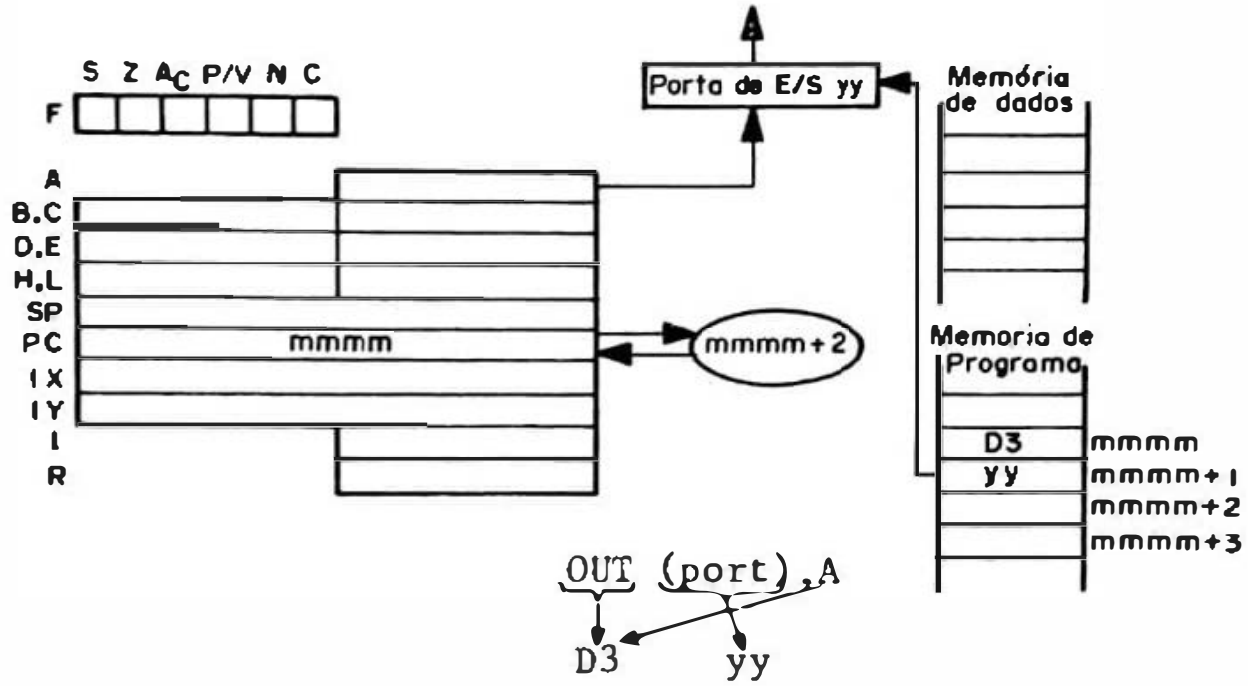
LOCALIZAÇÃO	CONTEÚDO
5000_{16}	CA_{16}
5001_{16}	$1B_{16}$
5002_{16}	$B1_{16}$
5003_{16}	AD_{16}

Após a execução da instrução OTIR, o par de registradores HL passará a conter o valor 5004_{16}, o registrador B conte

rá zero e a seqüência CA_{16}, $1B_{16}$, $B1_{16}$ e AD_{16} deverá ter sido escrita na porta de E/S FF_{16}.

Esta instrução é muito útil para a transferência de blocos de dados da memória para os dispositivos de saída.

87) OUT (port),A - SAIA DO ACUMULADOR.



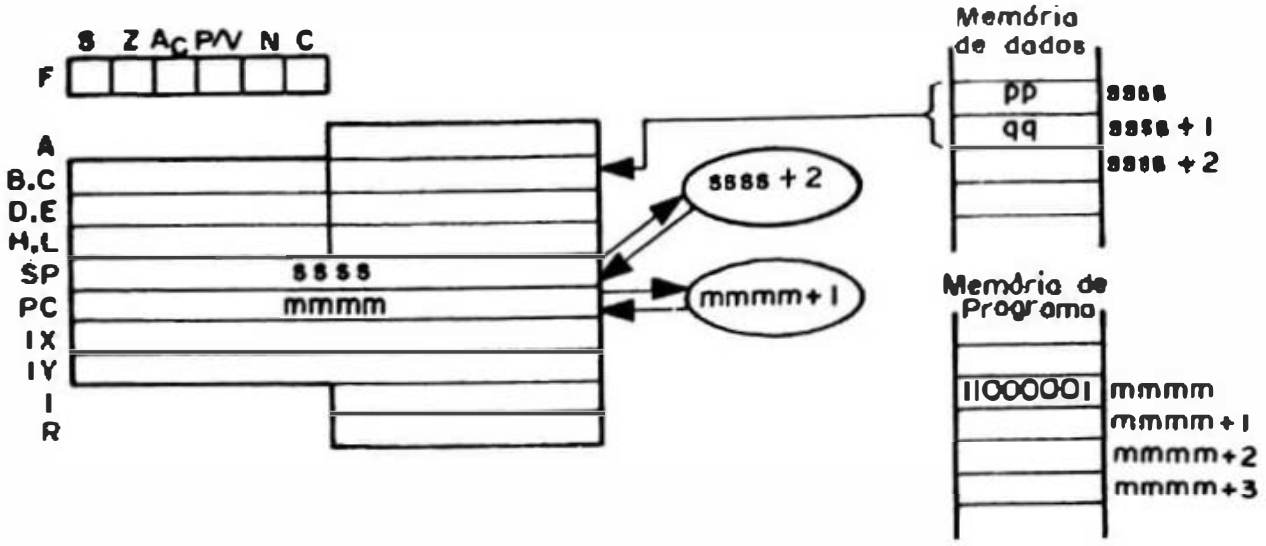
Formada por dois bytes, os quais constituem o código de operação. Esta instrução carrega o conteúdo do acumulador para a porta de E/S endereçada (ou especificada) pelo segundo byte do código objeto da instrução.

Suponhamos que o conteúdo do acumulador seja 36_{16}. Após a execução da instrução OUT (1AH),A, o valor 36_{16} estará no buffer da porta de E/S $1A_{16}$.

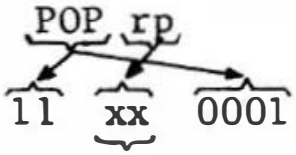
A instrução OUT não afeta nenhuma posição. O uso da instrução OUT depende, em muito, do hardware, pois os endereços válidos das portas de entrada/saída (E/S) são determinados pelo modo com que a lógica de E/S foi implementada.

As instruções OUT são freqüentemente usadas, em modos especiais, para controle da lógica de microcomputadores externos à "Unidade Central de Processamento" (CPU).

88) POP rp - FAÇA A LEITURA DO TOPO DA PILHA.
POP Ix
POP Iy



A ilustração nos mostra a execução da instrução POP BC



00 para rp igual ao par de registradores BC
01 para rp igual ao par de registradores DE
10 para rp igual ao par de registradores HL
11 para rp igual ao par de registradores A e F

Formada por dois bytes, os quais constituem o código de operação. Esta instrução carrega o par de registradores rp (no caso BC) com o conteúdo da memória endereçada pelo Stack-Pointer (ponteiro de pilha).

Suponhamos que qq = 01_{16} e pp = $2A_{16}$. Após a execução da instrução POP (HL), o valor 01_{16} é carregado no registrador L e o valor $2A_{16}$ é carregado no registrador H.

A execução da instrução POP AF carrega o valor 01 nos Flags de posição e o valor $2A_{16}$ no acumulador. Então, a posição de carry será ajustada (set) para "1", enquanto que as demais posições serão apagadas.

POP Ix

DD E1

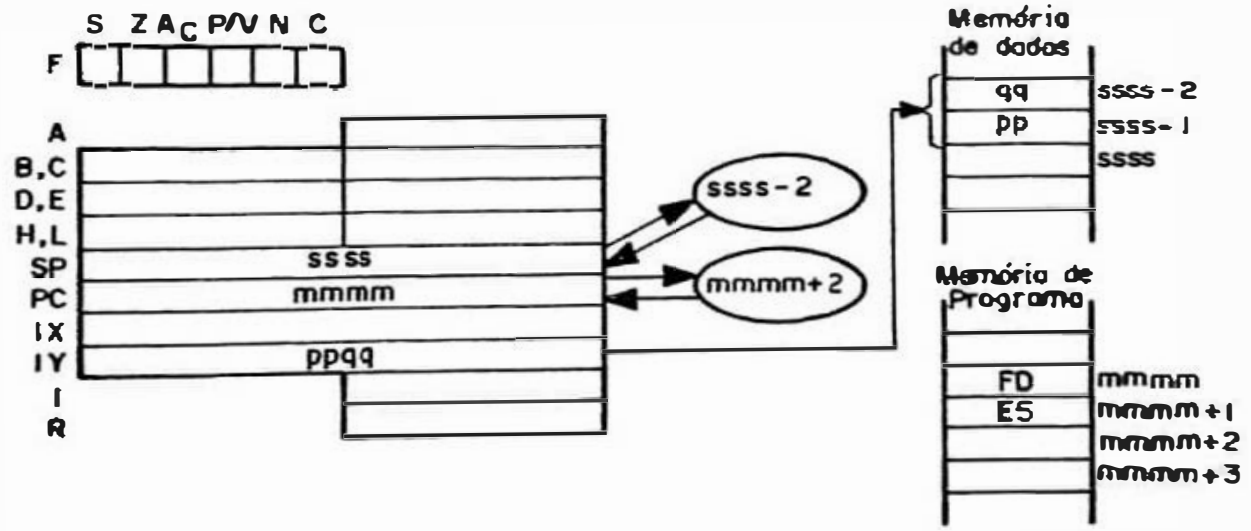
A instrução POP Ix é formada por dois bytes, os quais constituem o código de operação. Esta instrução carrega o conteúdo da memória endereçada pelo Stack-Pointer (SP) no registrador de índice Ix.

POP Iy
DD E1

A instrução POP Iy é idêntica à instrução POP Ix com a única diferença de que esta utiliza o registrador Iy em vez do registrador Ix.

A instrução POP é freqüentemente utilizada para restaurar (recuperar) o registrador e o conteúdo dos Flags de posição, gravados (ou guardados) na pilha.

89) PUSH rp - IMPRIMA (OU CARREGUE) NO TOPO DA PILHA.
PUSH Ix
PUSH Iy



A ilustração nos mostra a execução da instrução PUSH Iy.

PUSH Iy
FD E5

A instrução PUSH Iy é formada por dois bytes, os quais constituem o código de operação. Esta instrução carrega numa posição de memória, endereçada pelo registrador Stack-Pointer (ponteiro de pilha), o conteúdo do registrador Iy. O conteúdo do registrador SP é decrementado de duas posições. Esta instrução é usada para recuperação de pares de registradores.

Suponhamos que o registrador Iy contenha o valor $45FF_{16}$. Após a execução da instrução PUSH Iy, o valor $45FF_{16}$ será carregado para a posição de memória, indicada pelo registrador SP.

PUSH Ix

DD E5

A instrução PUSH Ix é idêntica à instrução PUSH Iy, exceto pelo fato da primeira utilizar o registrador Ix em vez do registrador Iy.

PUSH rp

11 xx 0101

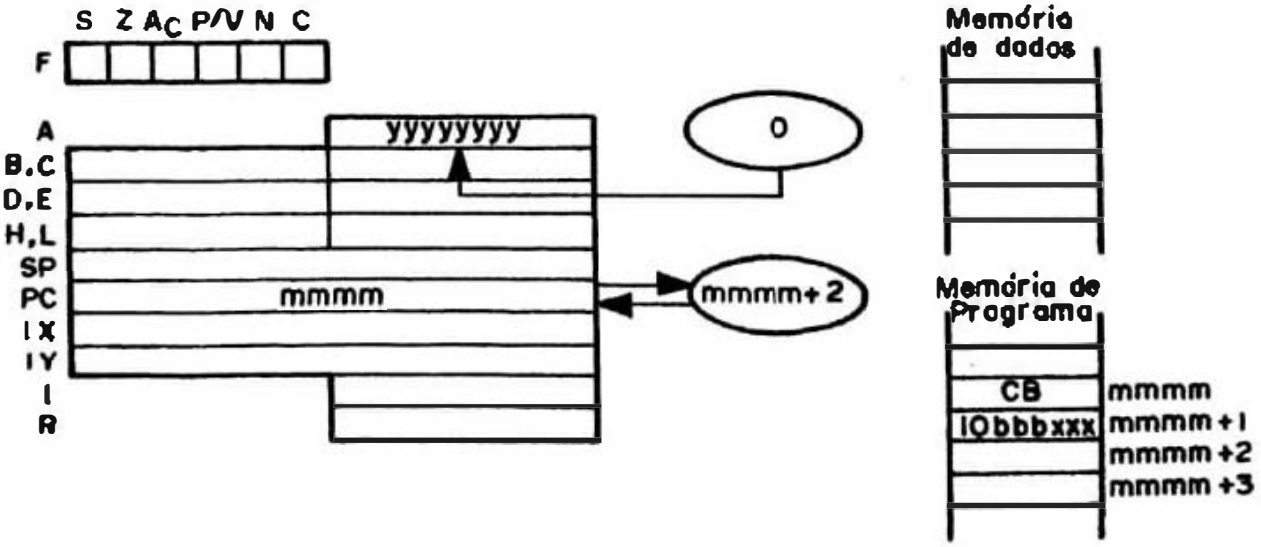
00 para rp igual ao par de registradores BC
01 para rp igual ao par de registradores DE
10 para rp igual ao par de registradores HL
11 para rp igual ao par de registradores A e F

Carregue o conteúdo do par de registradores designados para a posição de memória, indicada pelo Stack-Pointer (SP).

A execução da instrução PUSH AF carrega o conteúdo do acumulador e dos Flags de posição para a localização de memória, indicada pelo Stack-Pointer (SP).

A instrução PUSH é mais freqüentemente utilizada para guardar os conteúdos dos registradores e dos Flags de posição, como, por exemplo, antes de servir uma interrupção.

90) RES b,reg - RESET O BIT INDICADO DO REGISTRADOR.



RES b,reg

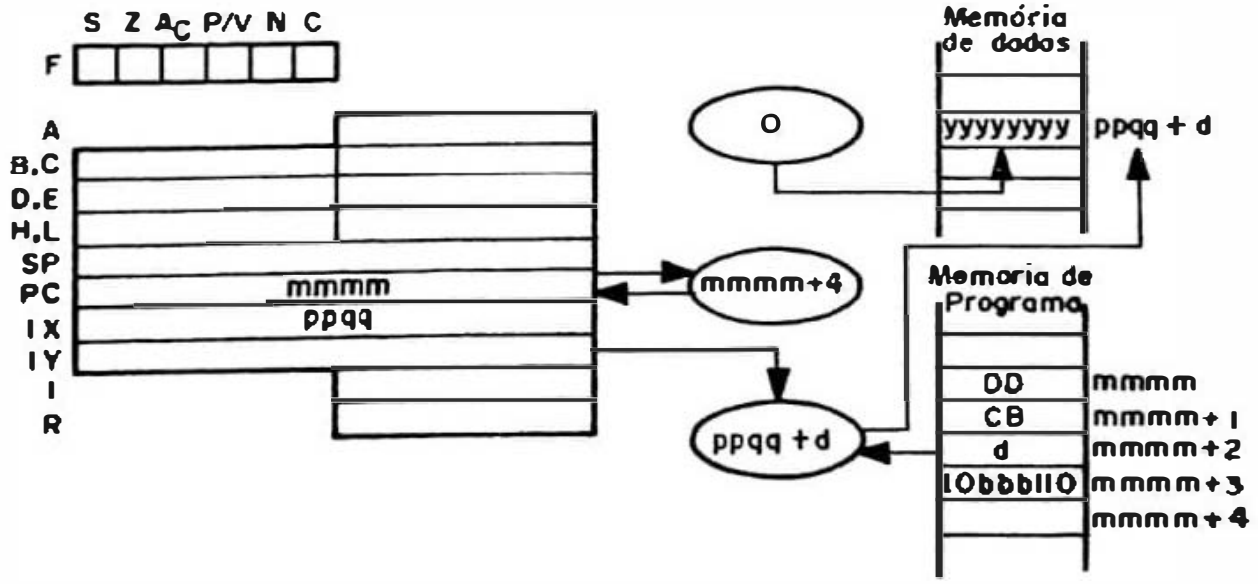
CB 10 bbb xxx

Bit	bbb	xxx	Registrador
0	000	000	B
1	001	001	C
2	010	010	D
3	011	011	E
4	100	100	H
5	101	101	L
6	110	111	A
7	111		

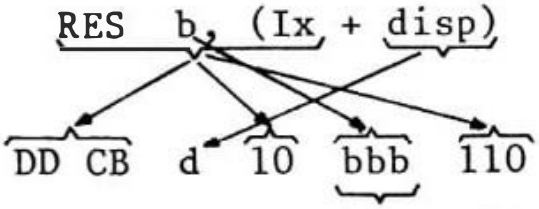
Formada por dois bytes, os quais constituem o código de operação. Esta instrução faz o reset, isto é, coloca o nível lógico zero em um único bit "b" do registrador reg especificado.

Após a execução da instrução RES 6,H, o bit 6 do registrador H será igual a 0 (zero).

91) RES b,(HL) - RESET O BIT b DA POSIÇÃO DE MEMÓRIA
RES b,(Ix+disp) INDICADA.
RES b,(Iy+disp)



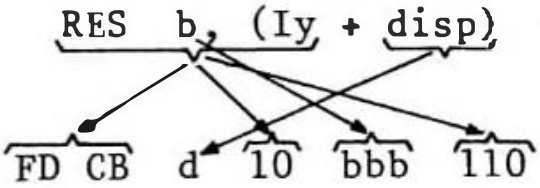
A ilustração nos mostra a execução do ajuste ou SET b, (Ix + disp). O bit 0 é o bit menos significativo.



bbb	Bit Reset
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

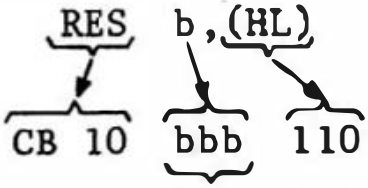
A instrução RES b, (Ix + disp) é formada por quatro bytes. Os dois primeiros e o quarto bytes constituem o código de operação e o terceiro byte é somado ou subtraído do registrador de índice. Esta instrução coloca o nível lógico zero em um único bit "b" da posição de memória, endereçada pelo registrador de índice Ix com o seu respectivo byte de deslocamento.

Suponhamos que Ix contenha o valor 4110_{16}. Após a execução da instrução RES 0,(Ix + 7), o bit zero "0" da localização de memória 4117_{16} será igual a zero "0".



bbb→idem instrução RES b,(Ix + d)

A instrução RES b,(Iy + disp) é idêntica à instrução RES b,(Ix + disp), exceto pelo fato da primeira utilizar o registrador Iy em vez do registrador de índice Ix.



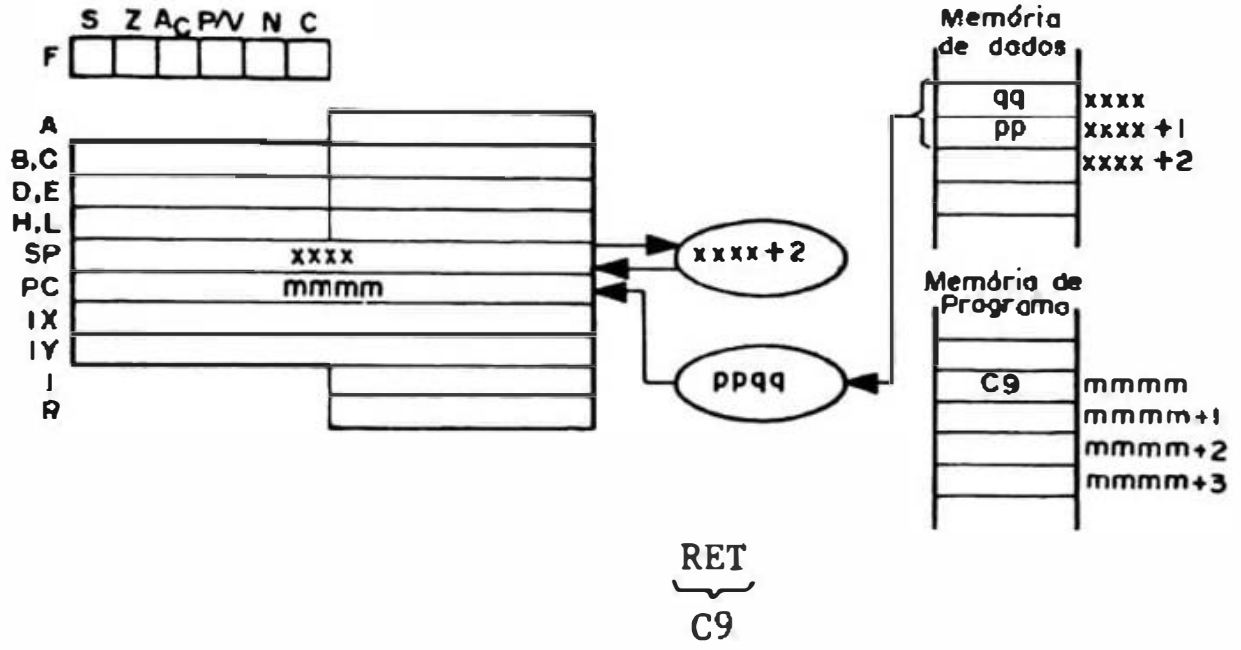
bbb→idem instrução RES b,(Ix + d)

A instrução RES b,(HL) é formada por dois bytes, os quais constituem o código de operação. Esta instrução coloca o ní-

vel lógico zero (reset) em um único bit "b" da posição de memória, endereçada pelo par de registradores HL.

Suponhamos que o conteúdo de HL seja igual a 4444_{16}. Após a execução da instrução RES 7,(HL), o bit de número 7 da localização de memória 4444_{16} passará a conter o valor "0" (zero).

92) RET - RETORNE DA SUB-ROTINA.

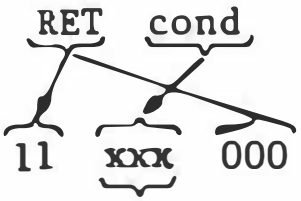


Formada por um único byte, o qual constitui o código de operação. Esta instrução carregará, incondicionalmente, isto é, sem verificar as condições dos Flags, o conteúdo dos dois bytes do topo da pilha (SP) para dentro do contador de programa PC. Estes dois bytes fornecem o endereço da próxima instrução a ser executada, indicando, desta forma, o fim de uma sub-rotina.

Os conteúdos prévios do "contador de programa" estão perdidos. O Stack-Pointer é incrementado de duas (2) posições para endereçar o novo topo de pilha.

Toda sub-rotina deve conter, no mínimo, uma instrução de retorno, a qual deverá ser a última instrução a ser executada dentro da sub-rotina. Esta instrução faz com que a execução retorne ao programa principal.

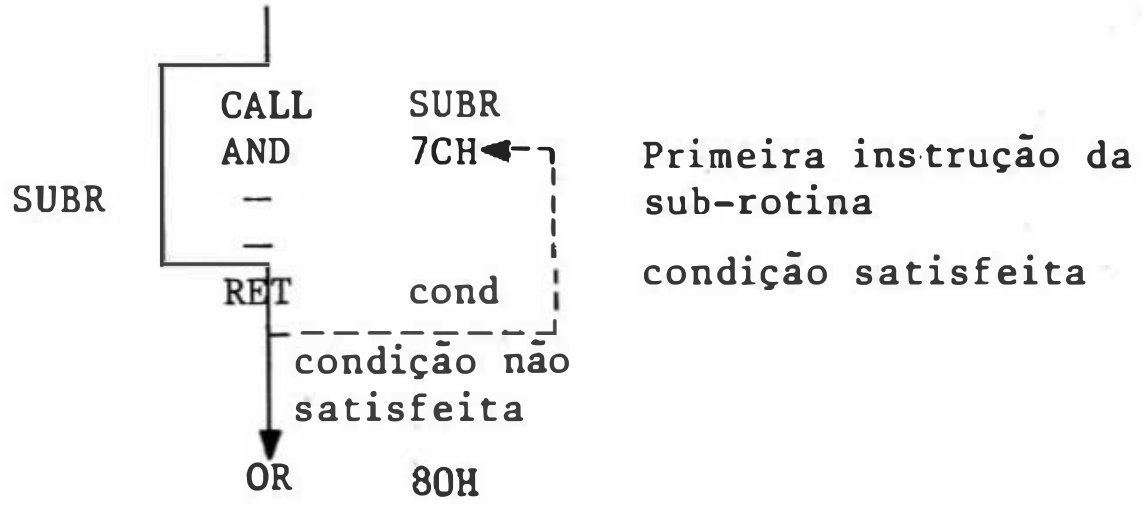
93) RET cond - RETORNE DA SUB-ROTINA SE A CONDIÇÃO É SATISFEITA.



	Condição		Flag Relevante
000	NZ	Não-zero	Z
001	Z	Zero	Z
010	NC	Não-carry	C
011	C	Carry	C
100	PO	Paridade ímpar	P/V
101	PE	Paridade par	P/V
110	P	Sinal positivo	S
111	M	Sinal negativo	S

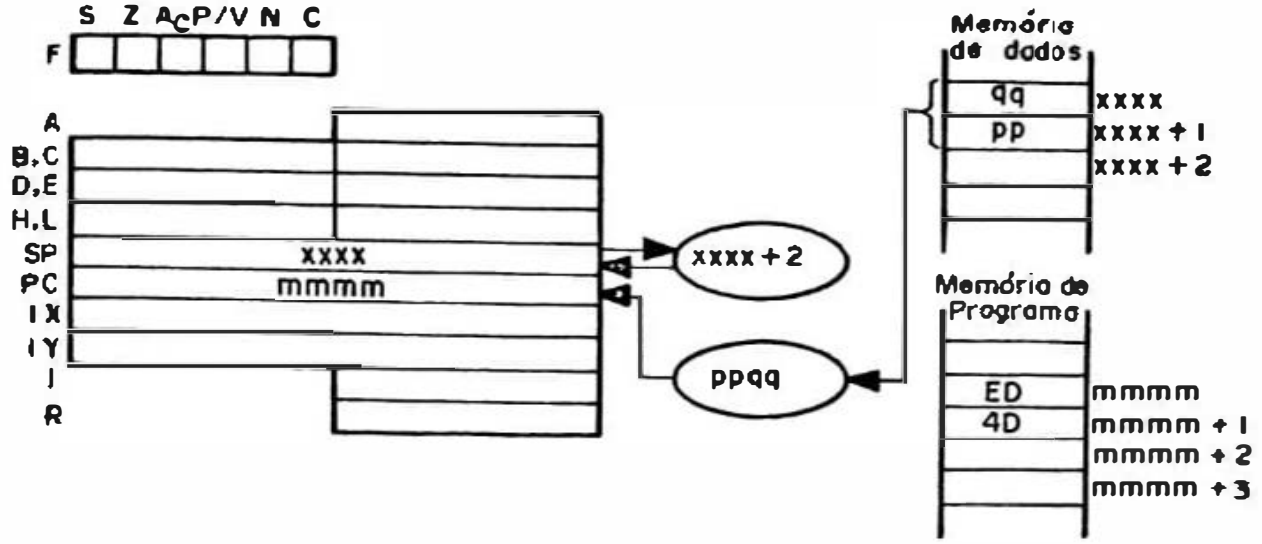
Esta instrução é formada por um único byte, o qual constitui o código de operação. Esta instrução é semelhante à instrução RET, com a exceção de que o retorno (ou RETURN) não é executado,caso a condição imposta não seja satisfeita. Assim, a instrução seguinte à instrução RET cond será executada.

Considere a seqüência de instruções:



Após a execução da instrução RET cond, se a condição for satisfeita, então, a execução retornará à instrução AND, a qual seguirá a instrução CALL. Se a condição não for satisfeita, a instrução OR, sendo a próxima instrução seqüencial, deverá ser executada.

94) RETI - RETORNE DA INTERRUPÇÃO.



RETI

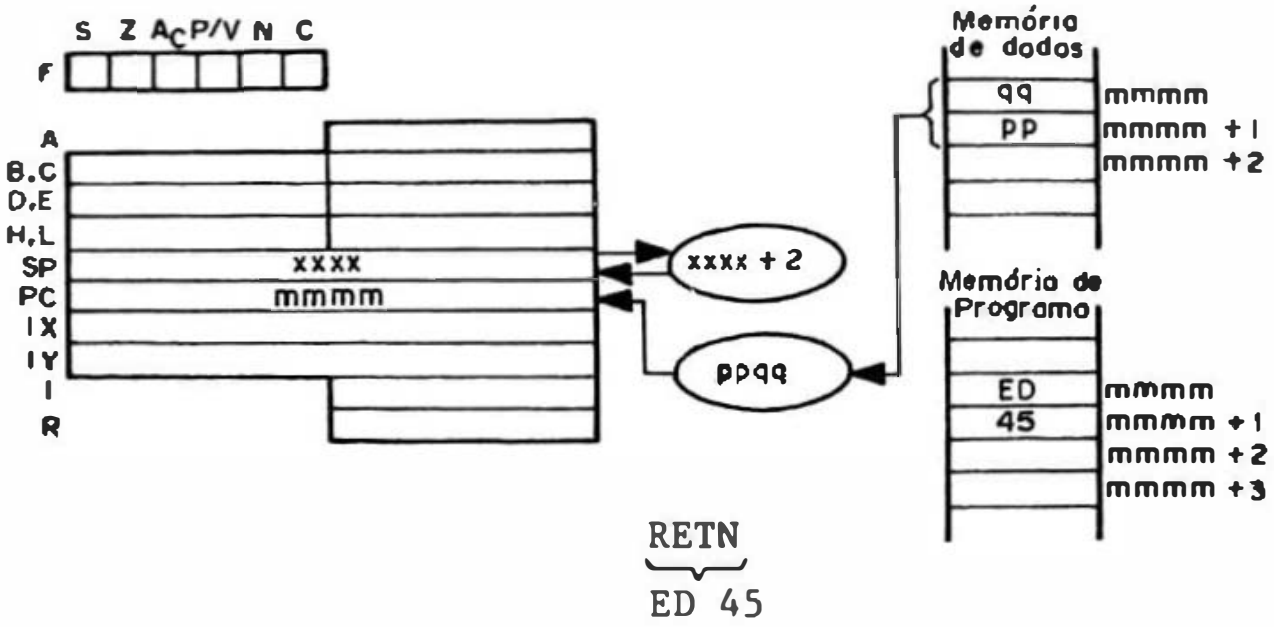
ED 4D

Formada por dois bytes, os quais constituem o código de operação. Esta instrução carrega, incondicionalmente, o conteúdo dos dois bytes do topo da pilha (Stack-Pointer) para dentro do contador de programa PC. Esses dois bytes fornecem o endereço da próxima instrução a ser executada.

Os conteúdos prévios (anteriores) do "contador de programa" estão perdidos. O Stack-Pointer é incrementado de duas (2) posições para endereçar o novo topo de pilha.

Esta instrução é, normalmente, utilizada no final de uma sub-rotina de interrupção em adição ao controle de retorno(RETURN) ao programa interrompido, avisando ou assinalando ao dispositivo de E/S que a rotina de interrupção foi completada. O dispositivo de E/S deverá fornecer a lógica necessária para dar sentido ao código de operação da instrução.

95) RETN - RETORNE DA INTERRUPÇÃO NON-MASKABLE (NÃO-MASCARADA).



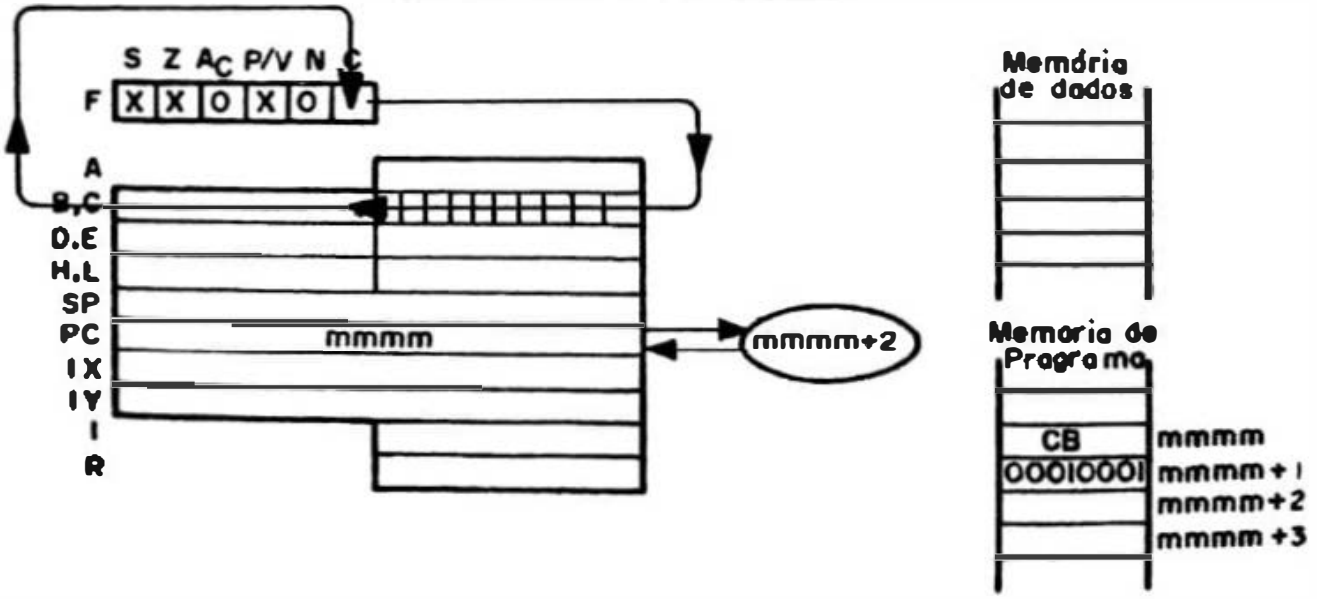
RETN

ED 45

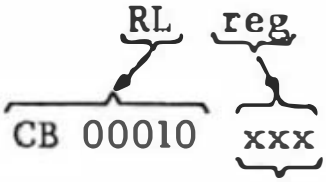
Formada por dois bytes, os quais constituem o código de operação. Esta instrução, semelhante à instrução RET, carregará o conteúdo dos dois bytes do topo da pilha (Stack-Pointer) para dentro do contador de programa PC. Estes dois bytes fornecem o endereço da próxima instrução a ser executada. Os conteúdos prévios (ou anteriores) do contador de programa (PC) são perdidos. O Stack-Pointer é incrementado de duas (2) posições para endereçar o novo topo de pilha. A lógica de ativação de interrupção é então restaurada para o estado em que ela se encontrava antes da ocorrência da interrupção não-mascarada.

Esta instrução é muito usada no final de uma rotina de serviço para uma interrupção não-mascarada, fazendo com que a execução retorne ao programa principal.

96) RL reg - GIRE O CONTEÚDO DO REGISTRADOR PARA A ESQUERDA ATRAVÉS DO CARRY.



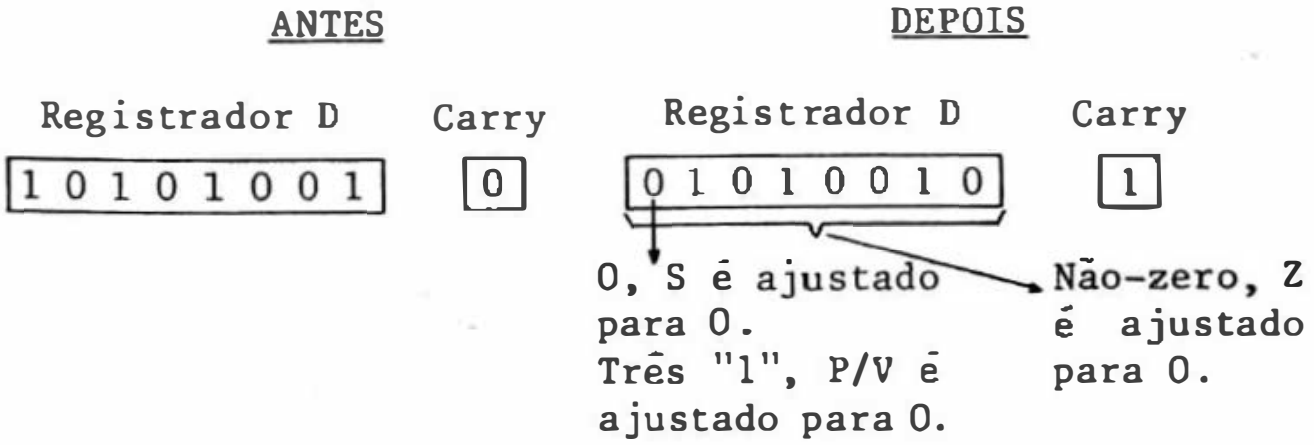
A ilustração nos mostra a execução da instrução RL C:



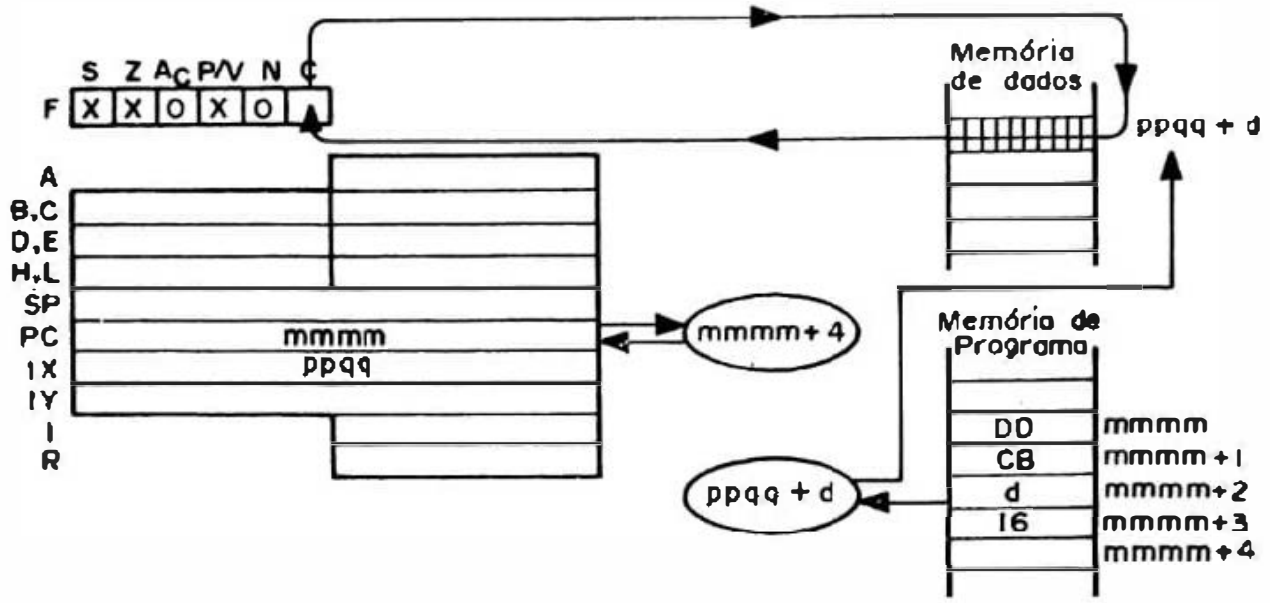
000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca (gira) de uma posição para a esquerda o conteúdo do registrador "reg", sendo o bit 7 carregado no Carry e o conteúdo do Carry carregado no bit 0 (zero).

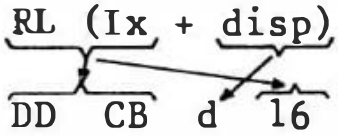
Suponhamos que o conteúdo do registrador D seja $A9_{16}$ e o conteúdo do Carry seja 0 (zero). Após a execução da instrução RLD, o conteúdo do registrador D passará a ser 52_{16} e o conteúdo do Carry será 1. Veja:



97) RL (HL) - GIRE (DESLOQUE) O CONTEÚDO DA LOCALIZA
RL (Ix+disp) ÇÃO DE MEMÓRIA PARA A ESQUERDA ATRAVÉS
RL (Iy+disp) DO CARRY.

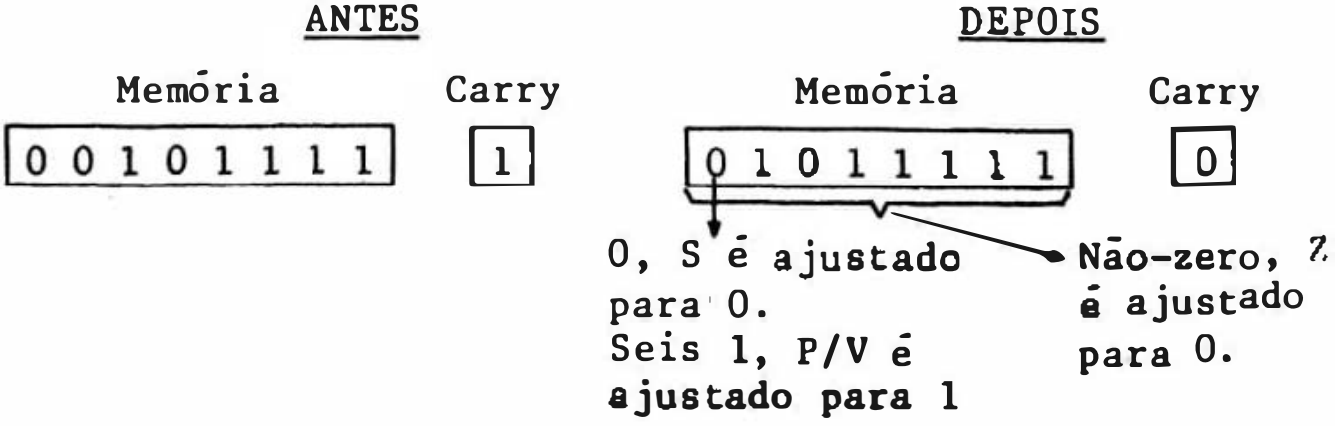


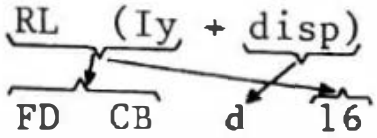
A ilustração nos mostra a execução da instrução RL (Ix + disp):



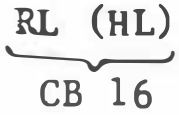
Formada por quatro bytes. Os dois primeiros, juntamente, com o quarto representam o código de operação, enquanto que o terceiro byte (disp) é somado ao registrador de índice Ix. Esta instrução gira ou desloca de uma posição para a esquerda o conteúdo da localização de memória, especificada por "Ix + disp".

Suponhamos que Ix contenha o valor 4000_{16}, a localização de memória 4007_{16} contenha $2F_{16}$ e que o Carry seja ajustado para 1. Após a execução da instrução RL (Ix + 7), a localização de memória 4007_{16} passará a conter o valor $5F_{16}$ e o Carry será igual a 0 (zero). Veja:



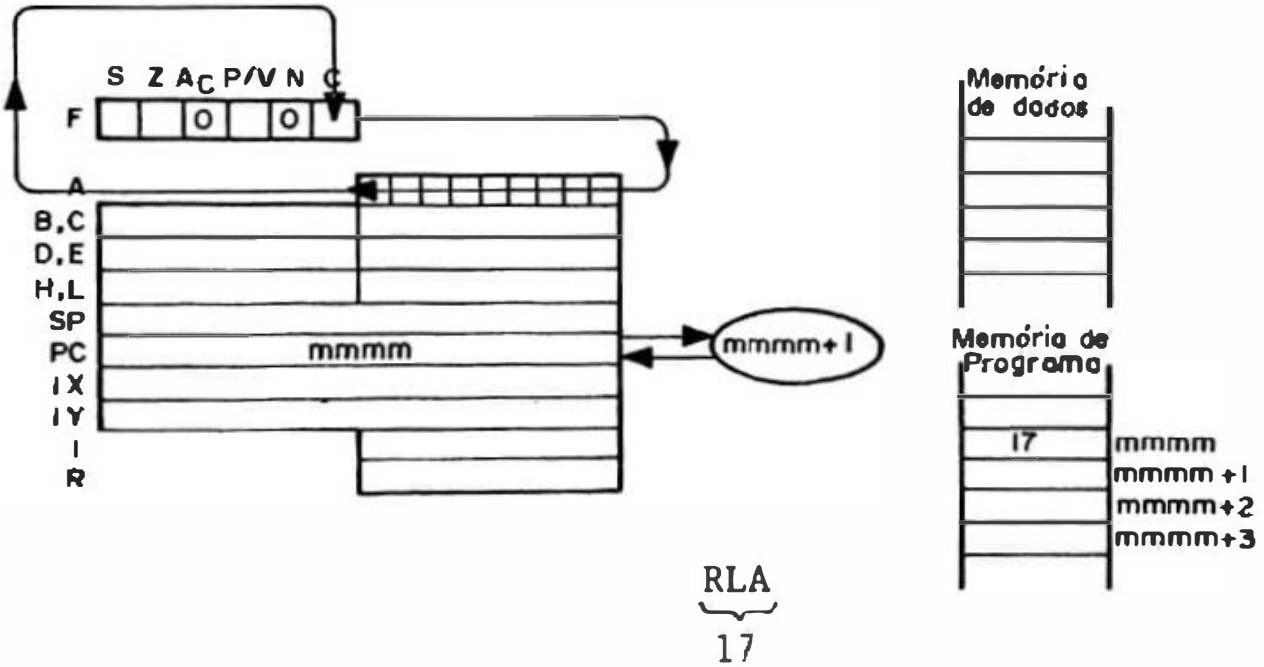


A instrução RL (Iy + disp) é idêntica à instrução RL(Ix + disp) com exceção do registrador Ix.



A instrução RL (HL) é formada por dois bytes, os quais constituem o código de operação. Esta instrução é semelhante à instrução "RL reg", com a única diferença de que o conteúdo da localização de memória, especificada (ou endereçada) pelo par de registradores HL, é que é deslocada de uma posição (ou um bit) para a esquerda através do carry.

98) RLA - GIRE (DESLOQUE) O ACUMULADOR PARA A ESQUERDA ATRAVÉS DO CARRY.

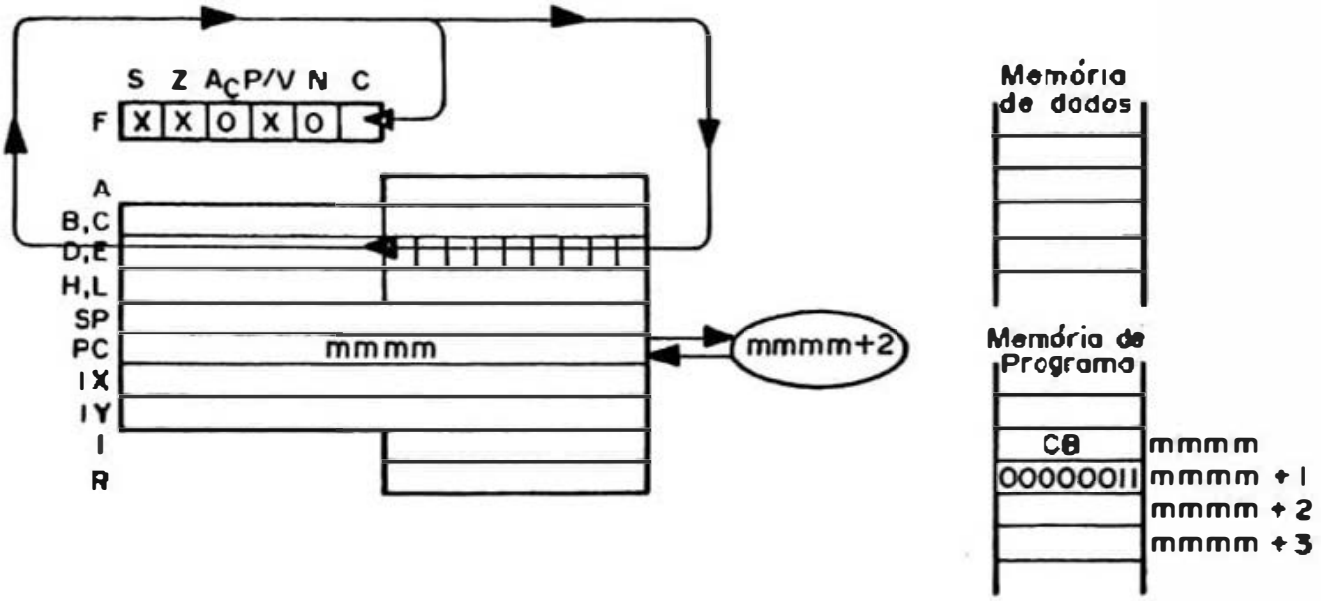


Constituída por um único byte, o qual constitui o código de operação. Esta instrução é semelhante à instrução "RL reg". Ela gira (ou desloca) o conteúdo do acumulador um bit para a esquerda, através da posição de carry.

Suponhamos que o acumulador contenha $2A_{16}$ e que o carry bit é ajustado (set) para 1. Após a execução RLA, o acumulador passará a conter $F5_{16}$ e a posição será 0 (zero).

ANTES		DEPOIS	
Acumulador	Carry	Acumulador	Carry
0 1 1 1 1 0 1 0	1	1 1 1 1 0 1 0 1	0

99) RLC reg - GIRE (DESLOQUE) O CONTEÚDO DO REGISTRADOR PARA A ESQUERDA EM FORMA CIRCULAR.



A ilustração nos mostra a execução da instrução RLC E:

RLC reg

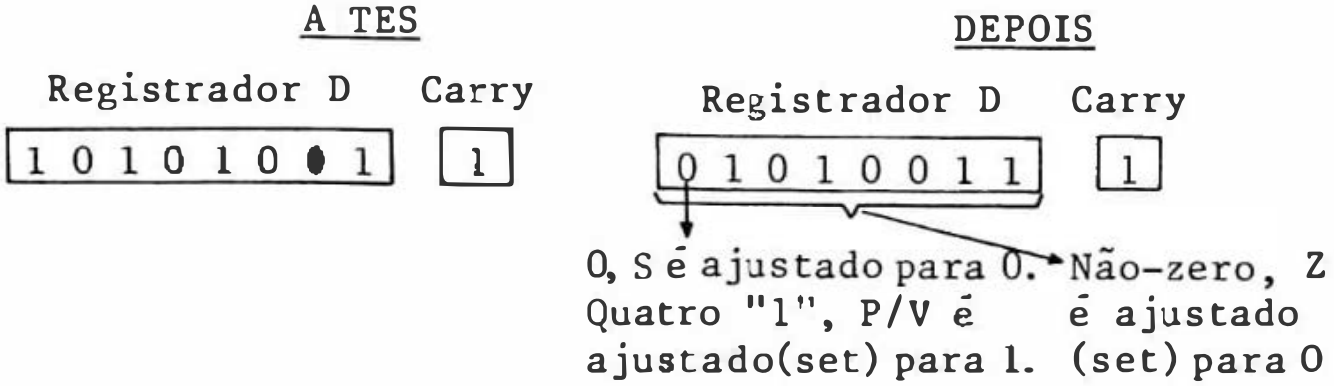
CB 00000 xxx

000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

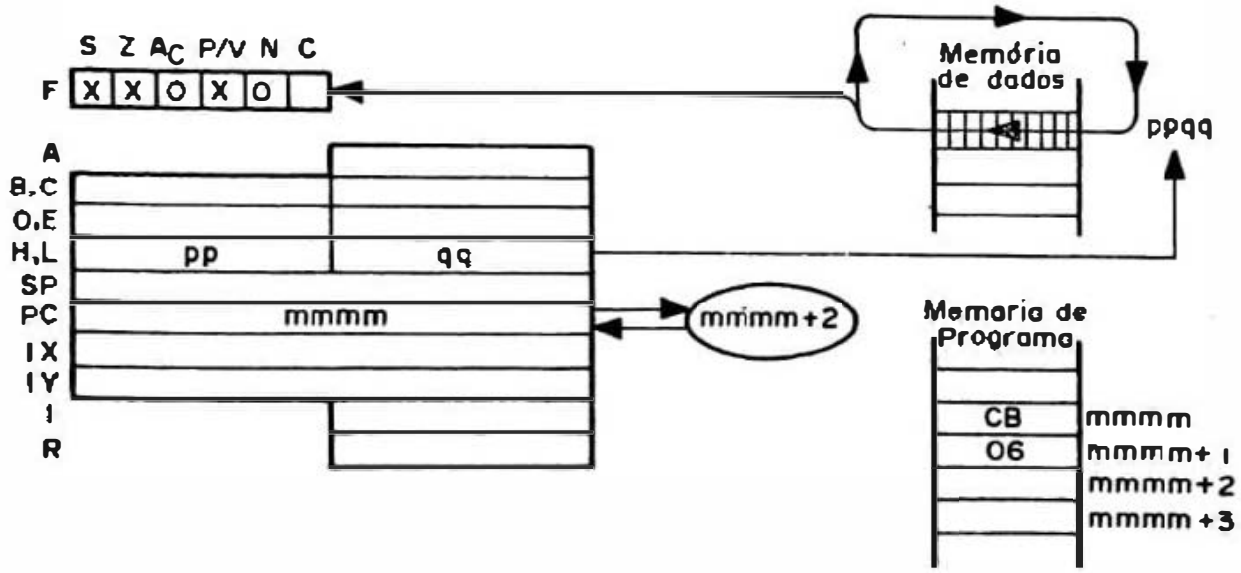
Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca (gira) o conteúdo do registrador "reg" de uma posição para a esquerda, sendo o bit 7 carregado no Carry e os demais bits, deslocados de uma posição. O conteúdo do Carry é afetado após a execução desta instrução.

Suponhamos que o conteúdo do registrador D contenha $A9_{16}$ e que o Carry contenha 1. Após a execução da instrução RLC D,

o registrador D passará a conter 53_{16} e o Carry será igual a 1. Veja:



100) RLC (HL) - GIRE (DESLOQUE) O CONTEÚDO DA LOCALI
RLC (Ix+disp) ZAÇÃO DE MEMÓRIA PARA A ESQUERDA EM
RLC (Iy+disp) FORMA CIRCULAR



A ilustração nos mostra a execução da instrução RLC (HL):

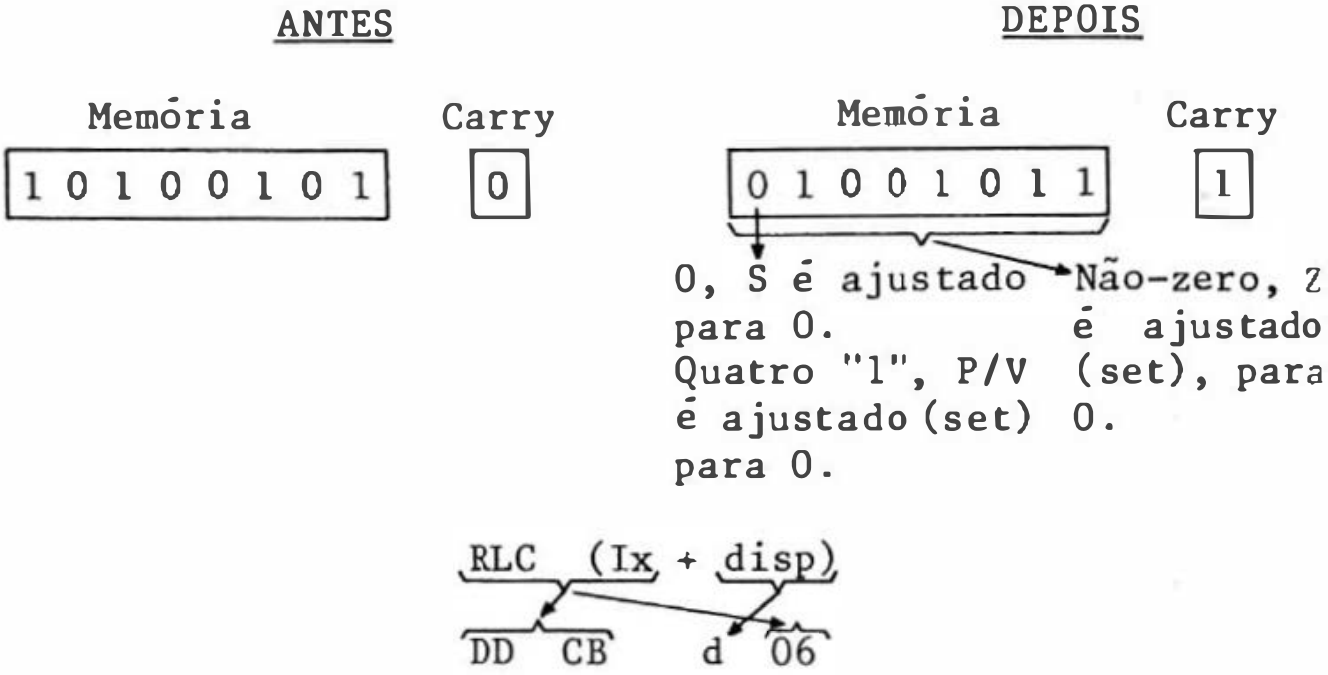
RLC (HL)

CB 06

Formada por dois bytes, os quais constituem o código de operação. A instrução RLC (HL) desloca (gira) o conteúdo da localização de memória, endereçada pelo conteúdo do par de registradores HL, um bit para a esquerda, transferindo o bit $\overline{7}$ Para o Carry.

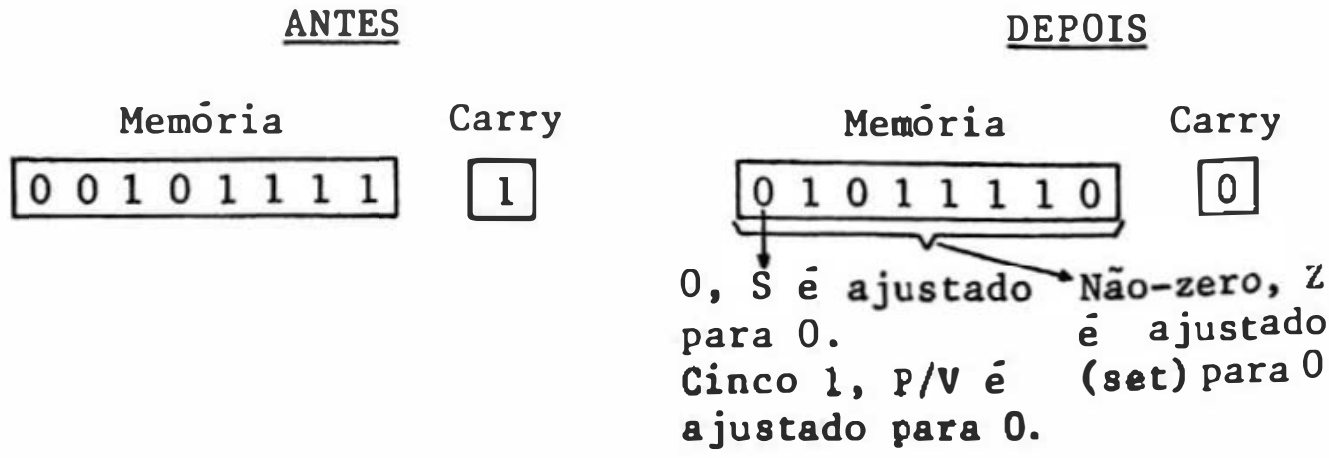
Suponhamos que o conteúdo do par de registradores HL seja $54FF_{16}$ e a localização de memória $54FF_{16}$ contenha $A5_{16}$ e

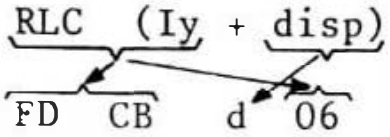
que o Carry seja igual a 0 (zero). Após a execução da instrução RLC (HL), a localização de memória $54FF_{16}$ passará a conter $4B_{16}$ e o Carry será igual a 1. Veja:



A instrução RLC (Ix + disp) é formada por quatro bytes. Os dois primeiros e o quarto constituem o código de operação e o terceiro é um byte de deslocamento. Esta instrução desloca (gira) o conteúdo da localização de memória, endereçada pelo conteúdo do registrador Ix, somando com o byte de deslocamento (disp), de um bit para a esquerda, transferindo (ou carregando) o bit 7 para o Carry.

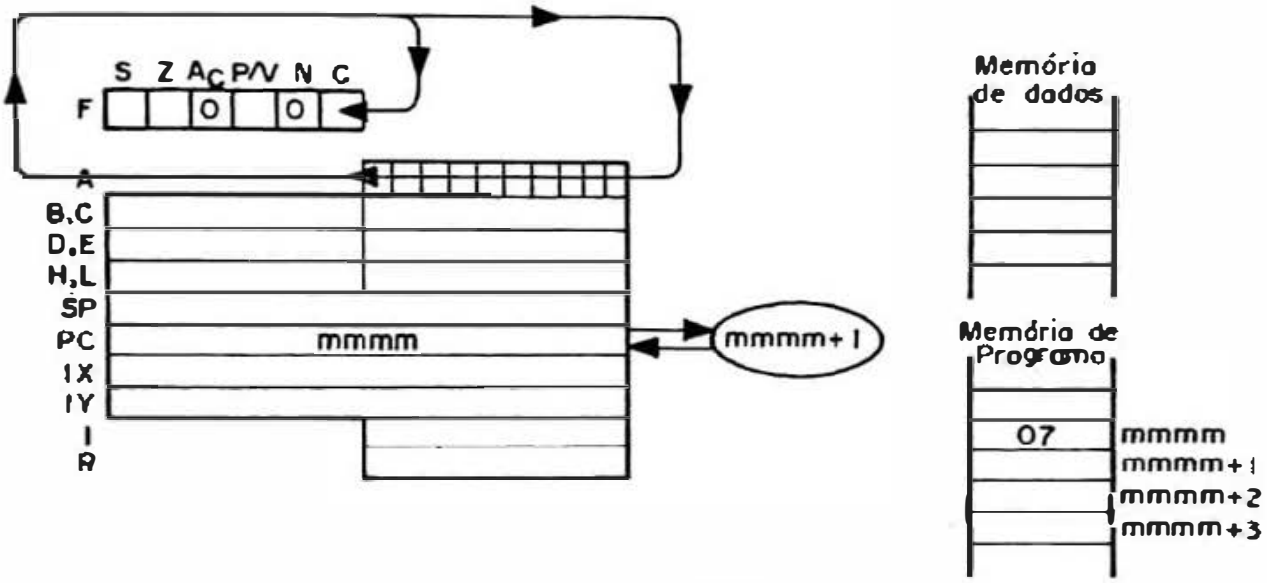
Suponhamos que o registrador Ix contenha 4000_{16}, o Carry seja igual a 1 e a localização de memória 4007_{16} contenha o valor $2F_{16}$. Após a execução da instrução RLC (Ix + 7), a localização de memória 4007_{16} passará a conter o valor $5E_{16}$ e o Carry será igual a 0 (zero).





A instrução RLC (Iy + disp) é idêntica á instrução RLC (Ix + disp), com exceção do registrador Ix.

101) RLCA - GIRE (DESLOQUE) O CONTEÚDO DO ACUMULADOR PARA A ESQUERDA EM FORMA CIRCULAR.



RLCA

07

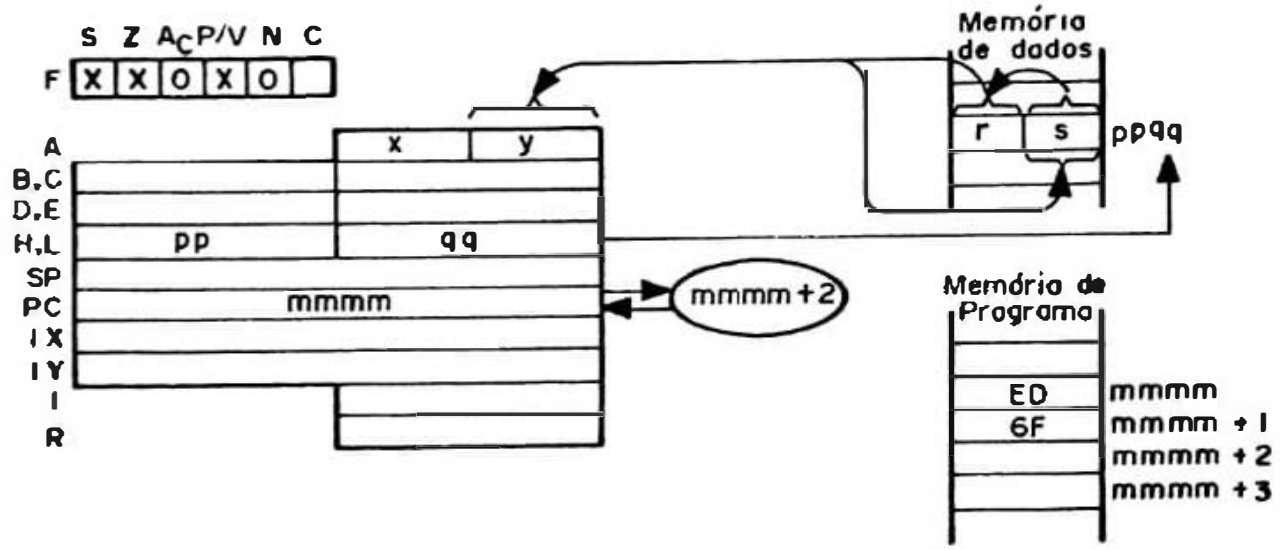
Formada por um único byte, o qual constitui o código de operação. Esta instrução desloca (ou gira) o conteúdo do acumulador de um bit (ou uma posição) para a esquerda e transfere (carrega) o bit 7 para o Carry.

Suponhamos que o acumulador contenha o valor $7A_{16}$ e a posição de Carry seja ajustada (set) para 1. Após a execução da instrução RLCA, o acumulador passará a conter $F4_{16}$, enquanto que a posição de Carry será zerada (reset), ou seja, o Carry será igual a 0 (zero). Vejamos:

ANTES		DEPOIS	
Acumulador	Carry	Acumulador	Carry
0 1 1 1 1 0 1 0	1	1 1 1 1 0 1 0 0	0

A instrução RLCA poderia ser utilizada como uma instrução lógica.

102) RLD - GIRE (DESLOQUE) UM DÍGITO BCD PARA A ESQUERDA ENTRE O ACUMULADOR E A LOCALIZAÇÃO DE MEMÓRIA ENDEREÇADA (OU ESPECIFICADA).

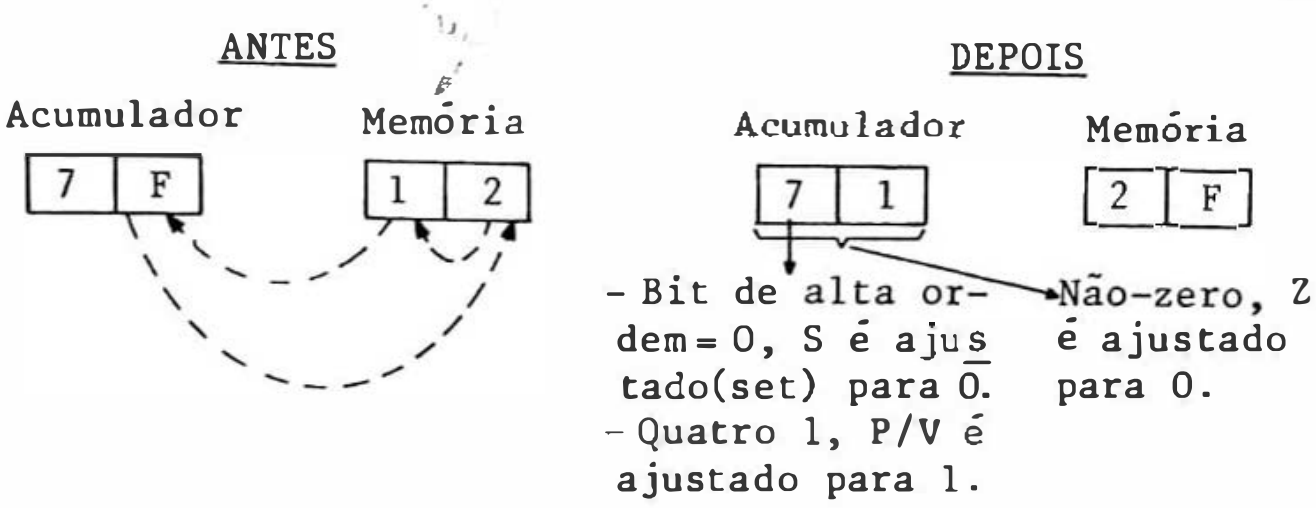


RLD

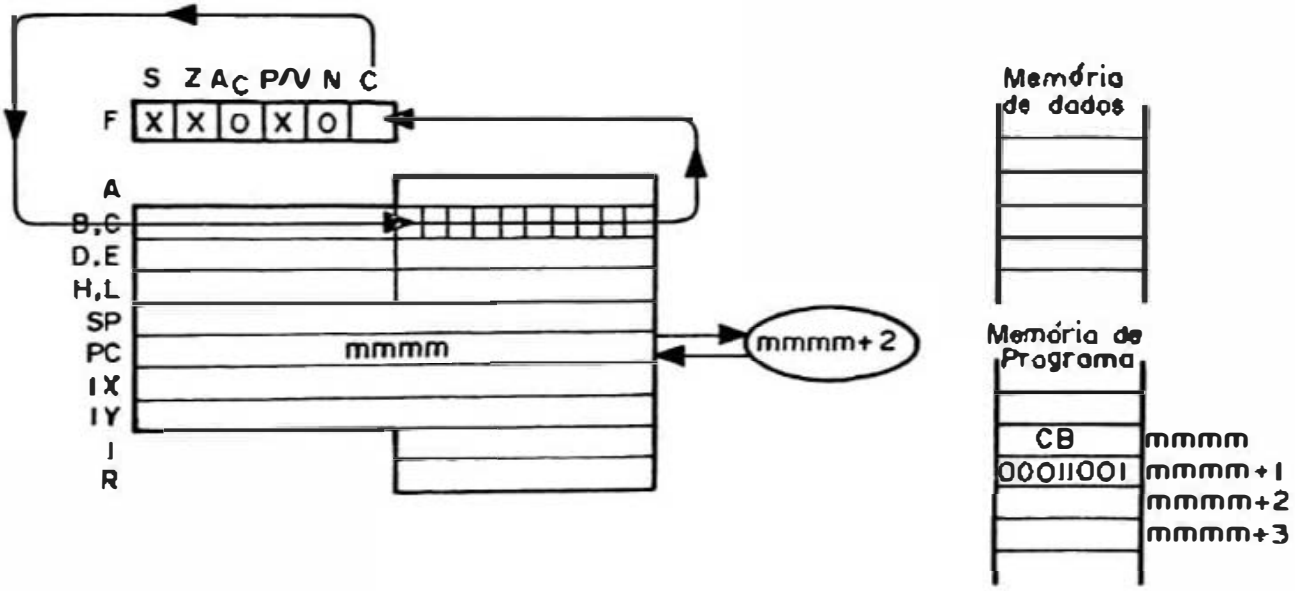
ED 6F

Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca (gira) os quatro bits de baixa ordem (ou menos significativos) da localização de memória (endereçada pelo conteúdo do par de registradores HL) para a posição dos quatro bits de alta ordem (ou mais significativos) da mesma localização de memória. A parte mais significativa desta localização de memória é então deslocada para a parte menos significativa do acumulador. Por último, a parte menos significativa do acumulador é deslocada (carregada) na parte menos significativa da localização de memória, especificada pelo par de registradores HL.

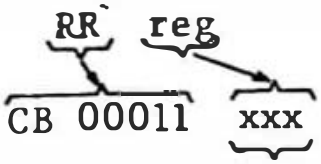
Suponhamos que o acumulador contenha $7F_{16}$, o par de registradores HL contenha 4000_{16} e a localização de memória 4000_{16} contenha 12_{16}. Após a execução da instrução RLD, o acumulador passará a conter 71_{16} e a localização de memória 4000_{16} conterá $2F_{16}$. Veja:



103) RR reg - DESLOQUE (GIRE) O CONTEÚDO DO REGISTRADOR PARA A DIREITA ATRAVÉS DO CARRY.



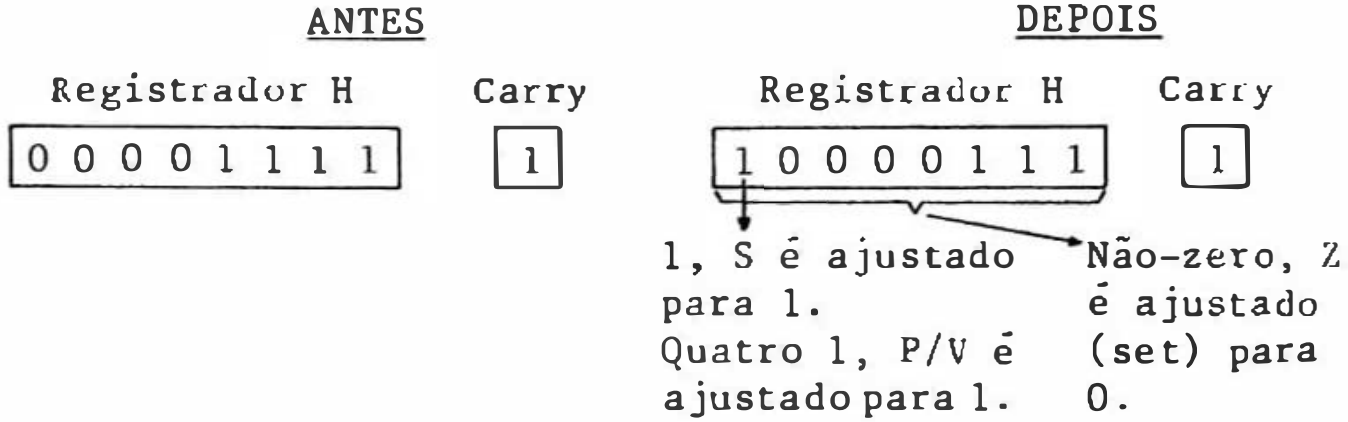
A ilustração nos mostra a execução da instrução RRC:



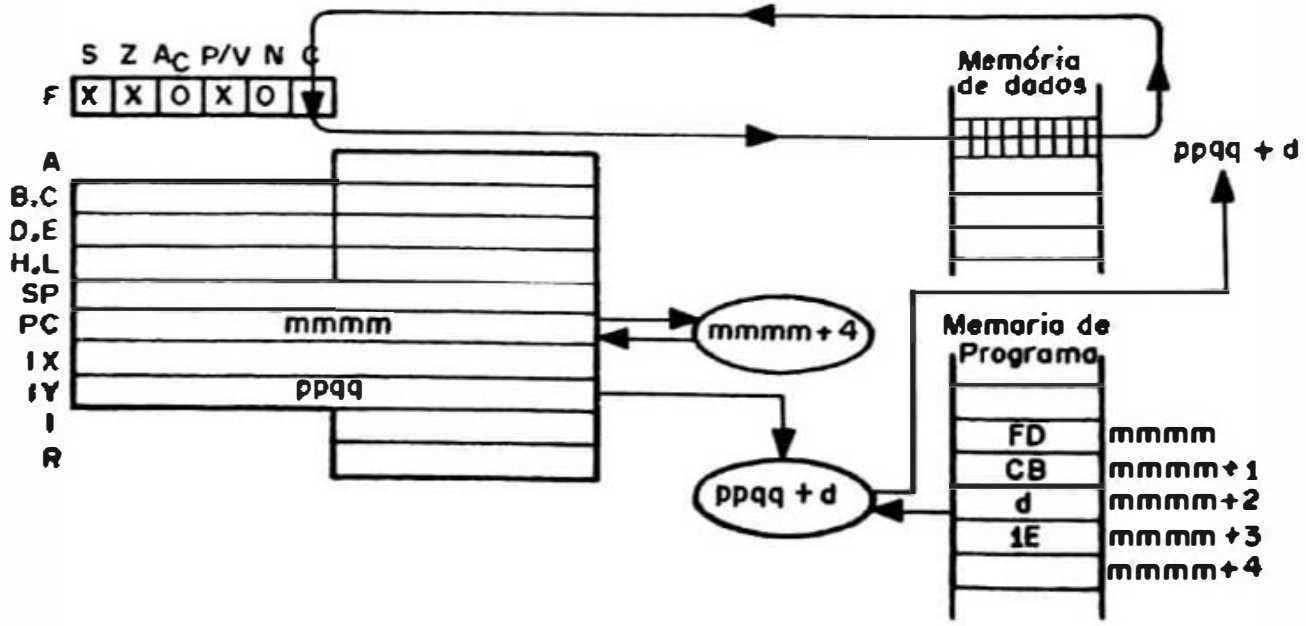
000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca (gira) de uma posição para a direita o conteúdo do registrador "reg", sendo o bit 7 carregado no Carry e o conteúdo do Carry, carregado no bit 0 (zero).

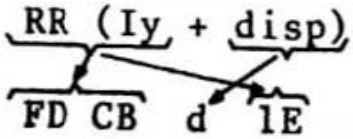
Suponhamos que o registrador H contenha $0F_{16}$ e que o Carry seja ajustado (set) para 1. Após a execução da instrução RR H, o registrador H passará a conter 87_{16} e o Carry será igual a 1. Veja:



104) RR (HL) - GIRE (DESLOQUE) O CONTEÚDO DA LOCALI-
RR (Ix+disp) ZAÇÃO DE MEMÓRIA PARA A DIREITA ATRA-
RR (Iy+disp) VÉS DO CARRY.

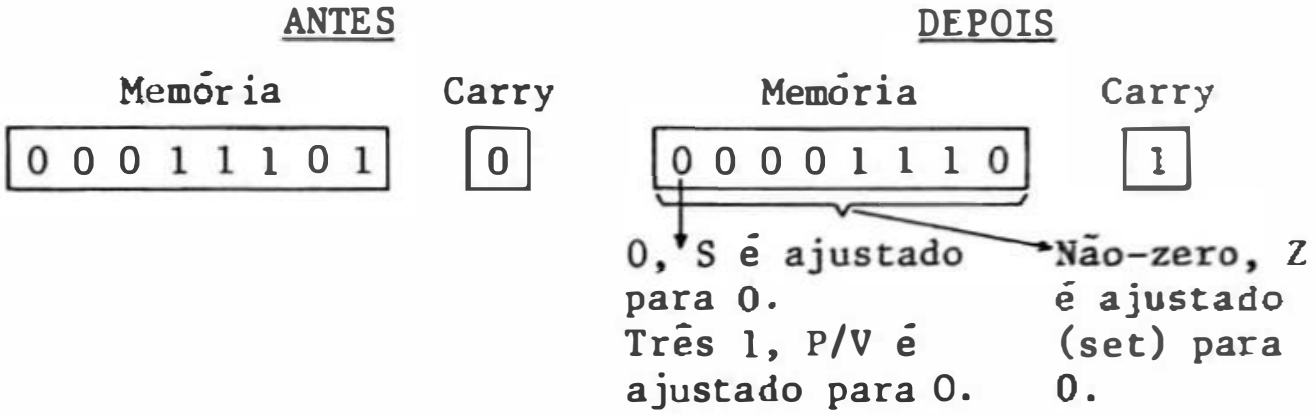


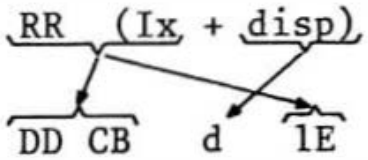
A ilustração nos mostra a execução da instrução RR (Iy + disp):



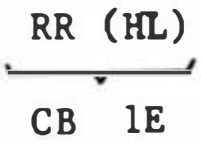
A instrução RR (Iy + disp) é formada por quatro bytes. Os dois primeiros e o quarto constituem o código de operação, enquanto que o terceiro byte (disp) é somado ao registrador de índice Iy. Esta instrução gira ou desloca de uma posição para a direita, através do Carry, o conteúdo da localização de memória, especificado (ou endereçado) pelo conteúdo de Iy, mais o valor do deslocamento d (ou Iy + disp).

Suponhamos que o registrador Iy contenha 4500_{16}, a localização de memória $450F_{16}$ contenha $1D_{16}$ e o Carry seja igual a 0 (zero). Após a execução da instruçao RR (Iy + 0FH), a localização de memória $450F_{16}$ passará a conter $0E_{16}$ e o Carry será igual a 1 (um). Veja:



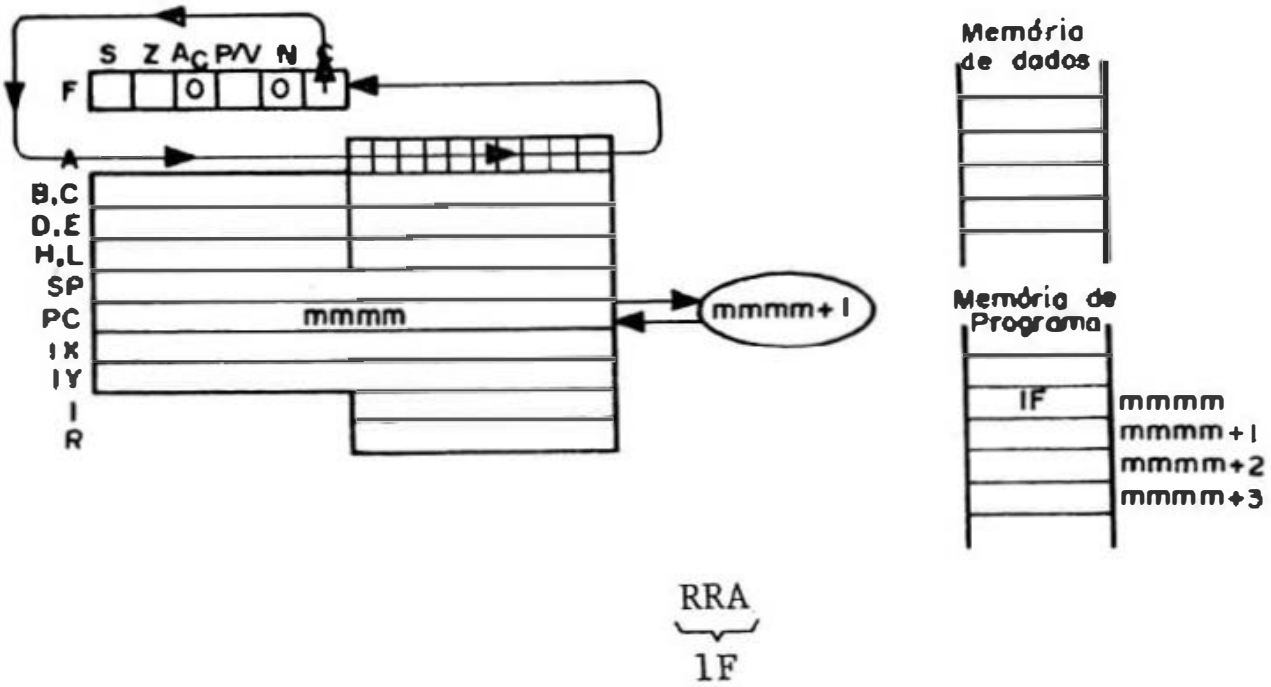


A instrução RR (Ix + disp) é idêntica à instrução RR (Iy + disp), com exceção do registrador Iy.



A instrução RR (HL) é formada por dois bytes, os quais constituem o código de operação. Esta instrução gira ou desloca de uma posição para a direita, através do Carry, o conteúdo da localização de memória, especificada ou endereçada pelo Par de registradores HL.

105) RRA - GIRE (DESLOQUE) O CONTEÚDO DO ACUMULADOR PARA A DIREITA ATRAVÉS DO CARRY.

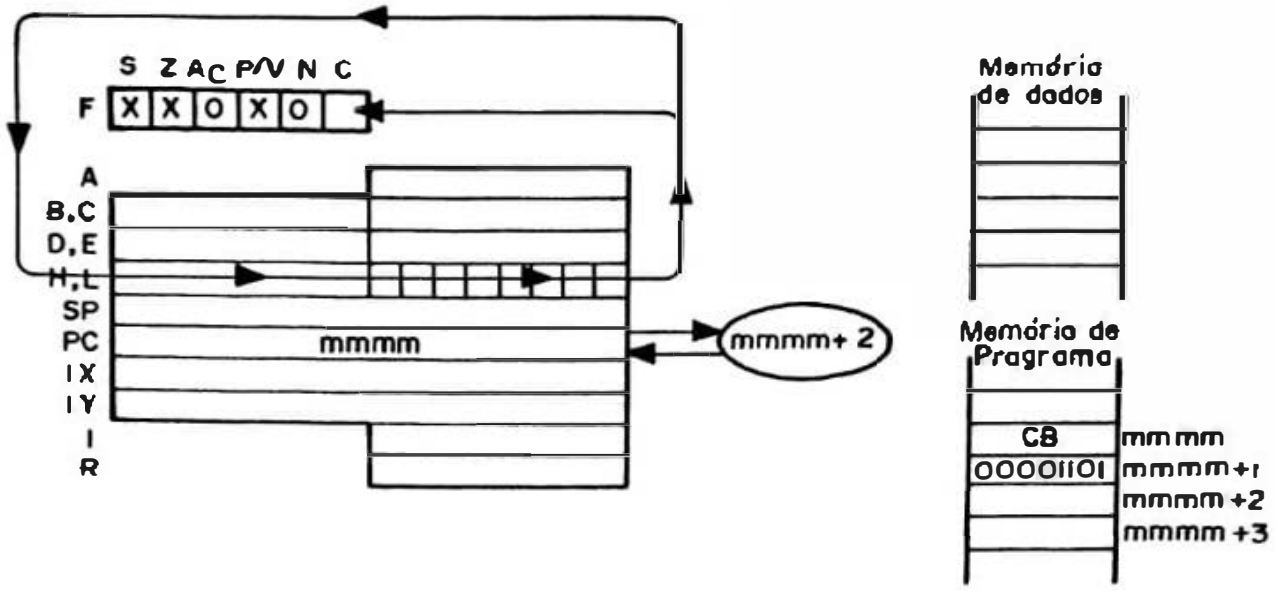


Formada por um único byte, o qual constitui o código de operação. Esta instrução desloca (gira) o conteúdo do acumulador de uma posição ou um bit para a direita através do Carry.

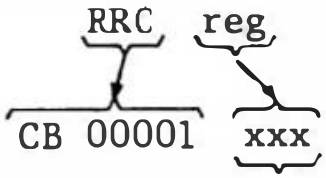
Suponhamos que o acumulador contenha $7A_{16}$ e que a posição de Carry seja igual a 1. Após a execução da instrução RRA, o acumulador passará a conter BD_{16} e a posição de Carry será zerada (reset), ou seja, será igual a 0 (zero). Veja:

ANTES		DEPOIS	
Acumulador	Carry	Acumulador	Carry
0 1 1 1 1 0 1 0	1	1 0 1 1 1 1 0 1	0

106) RRC reg - GIRE (DESLOQUE) O CONTEÚDO DO REGISTRADOR PARA A DIREITA EM FORMA CIRCULAR.



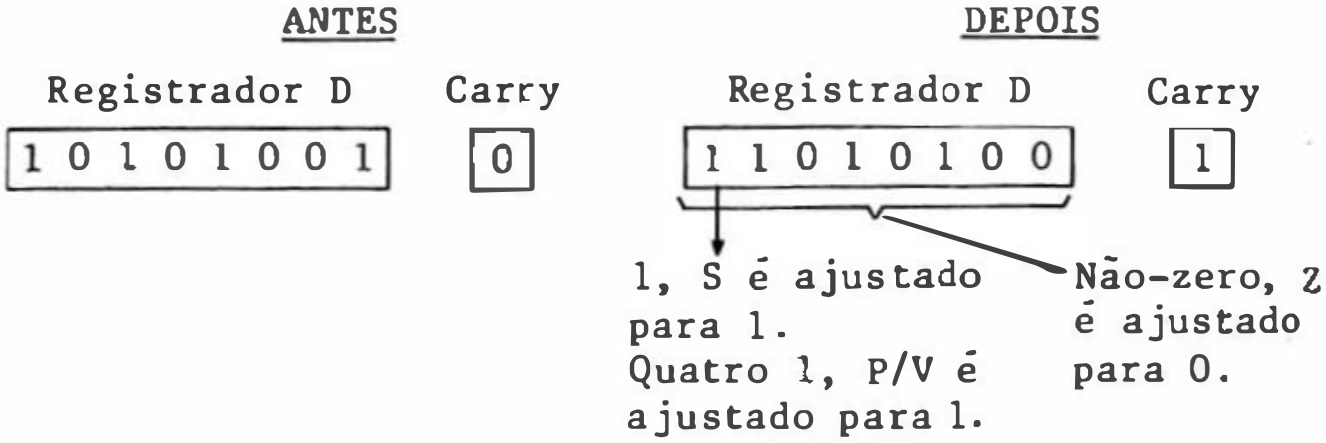
A ilustração nos mostra a execução da instrução RRC L:



000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca (gira) o conteúdo do registrador "reg" de uma posição para a direita, sendo o bit 7 carregado no Carry e os demais bits deslocados de uma posição. O conteúdo do "Carry bits" é perdido após a execução desta instrução.

Suponhamos que o conteúdo do registrador D contenha $A9_{16}$ e que o Carry seja igual a 0 (zero). Após a execução da instrução RRC D, o registrador D passará a conter $D4_{16}$ e o Carry será igual a 1.


ANTES
Registrador D
1 0 1 0 1 0 0 1
Carry
0
DEPOIS
Registrador D
1 1 0 1 0 1 0 0
Carry
1
1, S é ajustado para 1.
Quatro 1, P/V é ajustado para 1.
Não-zero, Z é ajustado para 0.

CEDM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-30

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) A instrução LDDR é constituída por:

   a) um byte;

   b) três bytes;

   c) nenhum byte;

   d) dois bytes.

2) Na instrução LDI, os registradores DE e HL são:

   a) decrementados;

   b) incrementados;

   c) mantidos como estavam, ou seja, permanecem indiferentes após a execução da instrução LDI;

   d) N.R.A.

3) A instrução NOP:

a) permite que se dê título a um byte do programa objeto;

b) permite ajustar os tempos de atraso;

c) os ítens a e b estão corretos;

d) N.R.A.

4) A instrução OR (HL) executa a função lógica:

a) OR entre o conteúdo da localização de memória, endereçada pelo par de registradores HL e o conteúdo do acumulador;

b) AND entre o conteúdo da localização de memória, endereçada pelo par de registradores HL e o conteúdo do acumulador;

c) OR EXCLUSIVA entre o conteúdo da localização de memória, endereçada pelo par de registradores HL e o conteúdo do acumulador;

d) N.R.A.

5) A instrução que coloca o conteúdo do registrador "reg" no dispositivo de E/S, endereçado pelo registrador C, é:

a) OUTD;

b) OR (Iy + disp);

c) OUTI;

d) OUT (C),reg

6) A instrução POP rp é constituída por:

a) um byte;

b) dois bytes;

c) três bytes;

d) quatro bytes.

7) As instruções POP são freqüentemente utilizadas para:

a) restaurar o registrador e o conteúdo dos flags de posição, os quais foram gravados na pilha;

b) imprimir dados na pilha;

c) em modos especiais, para controlar a lógica de microcomputadores externos à "Unidade Central de Processamento" (CPU);

d) N.R.A.

8) A instrução RES b,reg faz o reset no bit b do:

a) Flag de posição;

b) registrador reg. especificado;

c) da posição de memória, indicada pelo par de registradores HL;

d) N.R.A.

9) De uma forma genérica, a última instrução que, normalmente, é executada, dentro de uma sub-rotina, é a instrução:

a) POP rp;

b) RETN;

c) RET;

d) N.R.A.

10) A instrução RLC reg desloca o conteúdo do registrador "reg." de:

a) quatro posições para a esquerda;

b) três posições para a esquerda;

c) duas posições para a direita;

d) uma posição para a esquerda.

CEEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

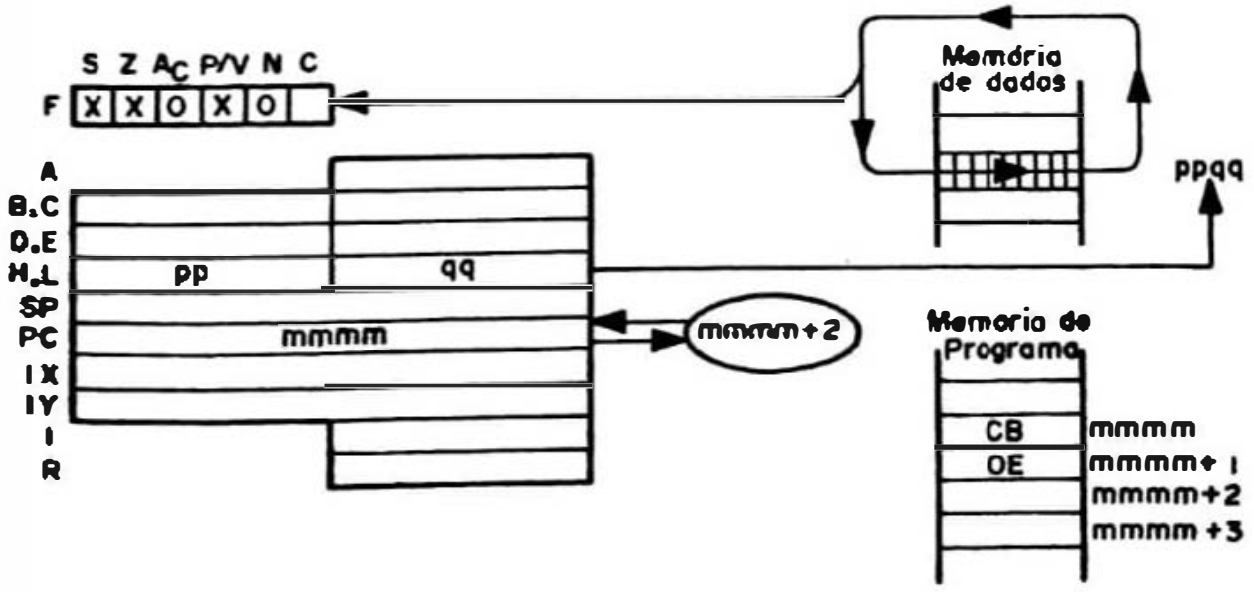
LIÇÃO MC-31

PROGRAMAÇÃO ASSEMBLER VI

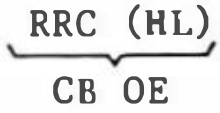
Nesta lição, finalizaremos o nosso estudo sobre as instruções do microprocessador Z-80.

107) RRC (HL)
RRC (Ix+disp)
RRC (Iy+disp)

- GIRE (DESLOQUE) O CONTEÚDO DA LOCALIZAÇÃO DE MEMÓRIA PARA A DIREITA EM FORMA CIRCULAR.

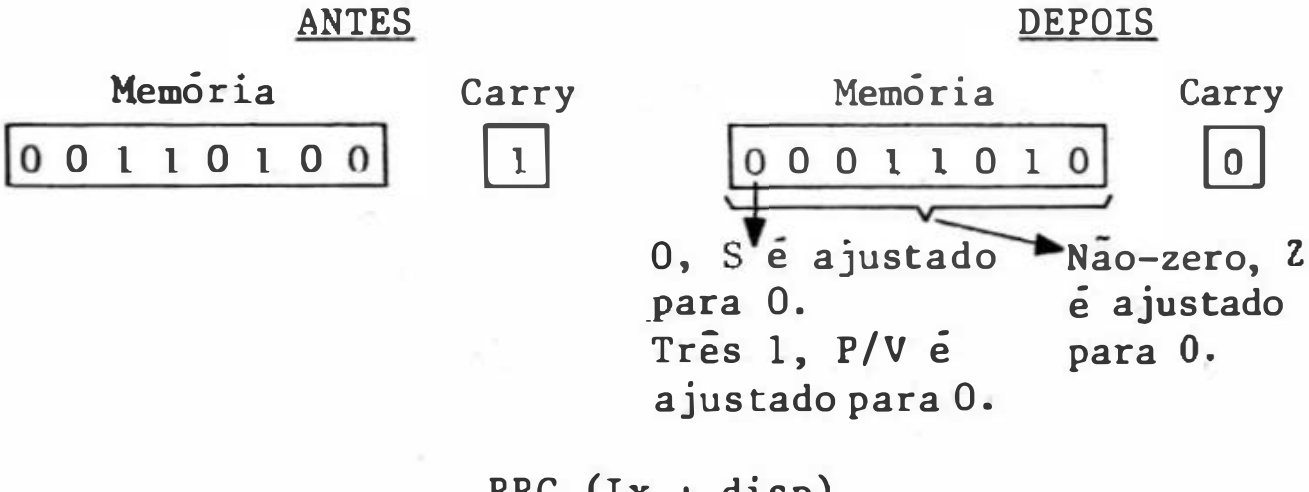


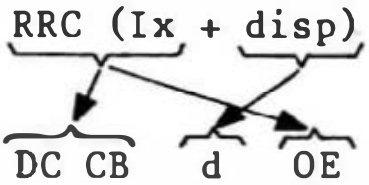
A ilustração nos mostra a execução da instrução RRC(HL):



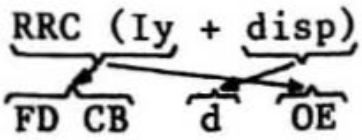
A instrução RRC (HL) é formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca (gira) o conteúdo da localização de memória, especificada (endereçada) pelo par de registradores HL, um bit ou uma posição para a direita, de forma circular, carregando o bit 0 na posição Carry.

Suponhamos que o par de registradores HL contenha o valor 4500_{16}, a localização de memória 4500_{16} contenha 34_{16} e que o Carry seja igual a 1. Após a execução da instrução RRC(HL), a localização de memória 4500_{16} passará a conter o valor $1A_{16}$ e o Carry será igual a 0 (zero). Veja:



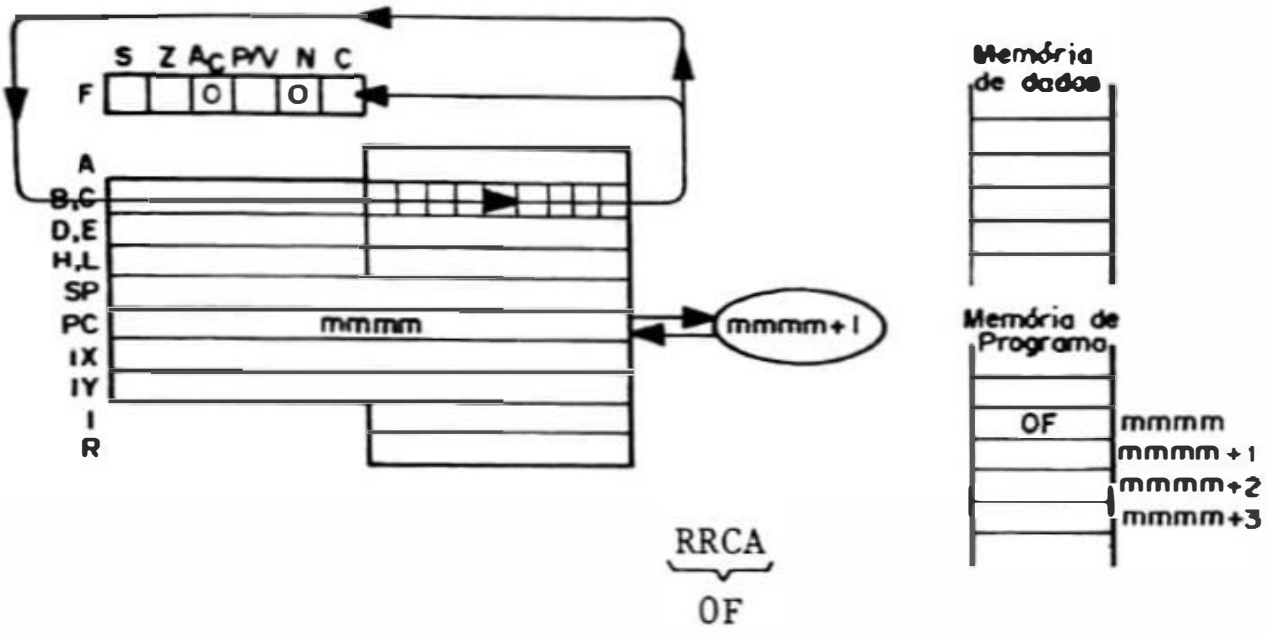


A instrução RRC (Ix+disp) é formada por quatro bytes. Os dois primeiros e o quarto constituem o código de operação e o terceiro byte (disp) é somado ao registrador de índice Iy. Esta instrução gira ou desloca o conteúdo da localização de memória, endereçada pela soma do conteúdo do registrador Ix com o valor do deslocamento (disp), de uma posição ou de um bit para a direita, de forma circular, carregando o bit 0 para a posição de Carry.



A instrução RRC (Iy + disp) é idêntica à instrução RRC (Ix + disp) com exceção do registrador Ix.

108) RRCA - GIRE O CONTEÚDO DO ACUMULADOR PARA A DIREITA EM FORMA CIRCULAR.



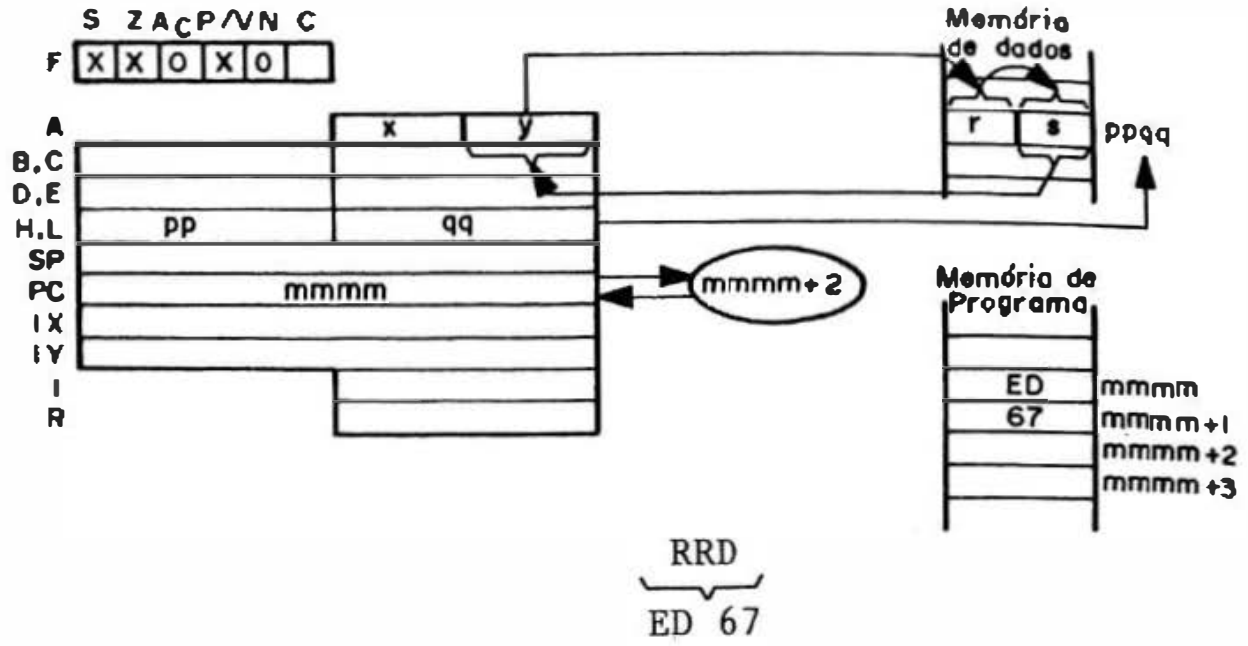
Formada por um único byte, o qual constitui o código de operação. Esta instrução desloca (gira) o conteúdo do acumulador uma posição de um bit para a direita, de forma circular, através do Carry.

Suponhamos que o conteúdo do acumulador seja $7A_{16}$ e que a posição de Carry seja igual a 1. Após a execução da instrução RRCA, o acumulador passará a conter o valor $3D_{16}$ e a posição de Carry será zerada (reset), ou seja, será igual a 0 (zero). Veja:

ANTES		DEPOIS	
Acumulador	Carry	Acumulador	Carry
0 1 1 1 1 0 1 0	1	0 0 1 1 1 1 0 1	0

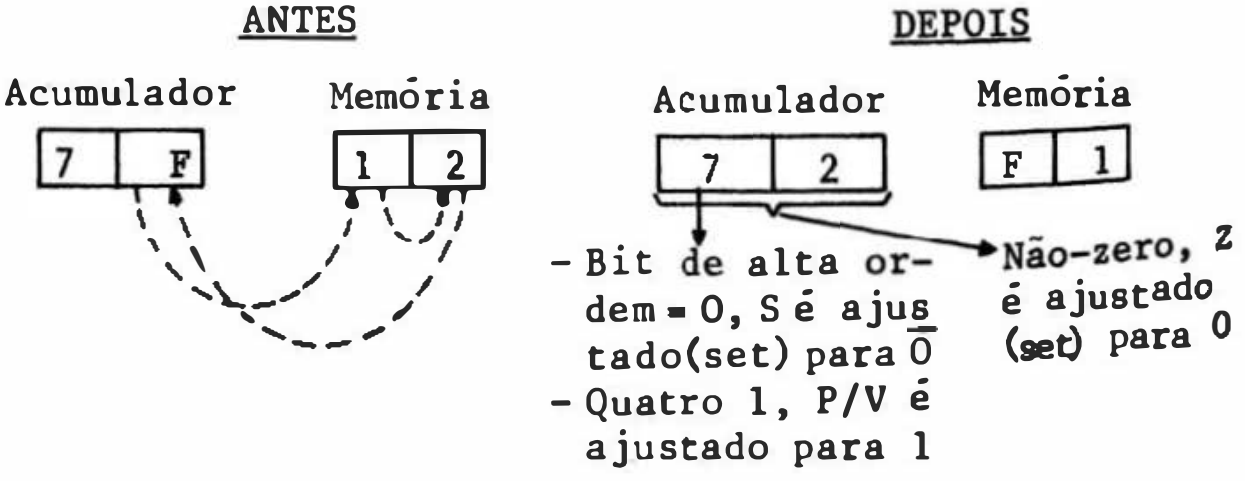
A instrução RRCA poderia ser utilizada como uma instrução lógica.

109) RRD - GIRE (DESLOQUE) UM DÍGITO BCD PARA A DIREITA ENTRE O ACUMULADOR E A LOCALIZAÇÃO DE MEMÓRIA ENDEREÇADA (OU ESPECIFICADA).

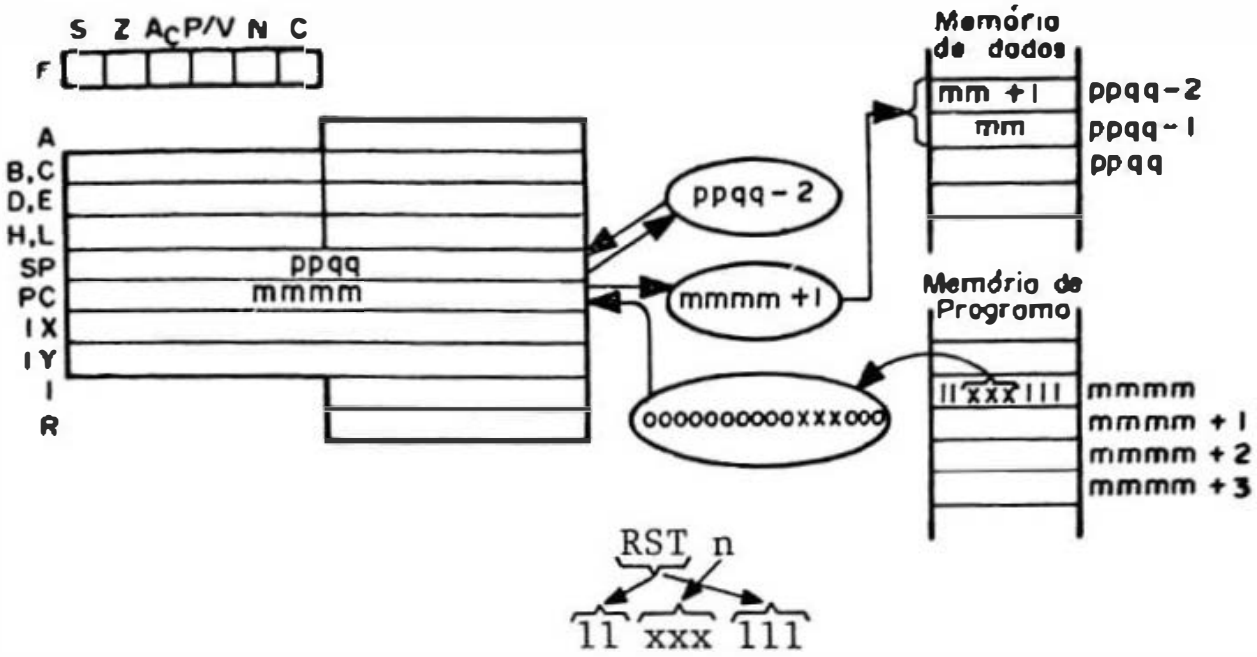


Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca (gira) os quatro bits de baixa ordem (ou menos significativos) do acumulador para a parte mais significativa, de uma posição de memória, endereçada pelo par de registradores HL. Os quatro bits de alta ordem (mais significativos), desta localização de memória, são deslocados para a posição dos quatro bits de baixa ordem, desta mesma localização de memória. E, por último, os quatro bits menos significativos (ou de baixa ordem), da localização de memória, endereçada pelo par de registradores HL, são deslocados para a posição dos quatro bits menos significativos do acumulador.

Suponhamos que o acumulador contenha o valor $7F_{16}$, o par de registradores HL contenha o valor 4000_{16} e que o conteúdo da localização de memória 4000_{16} seja 12_{16}. Após a execução da instrução RRD, o acumulador passará a conter 72_{16} e a localização de memória 4000_{16} conterá o valor $F1_{16}$. Veja:



110) RST n - RESTART (RECOMECE).



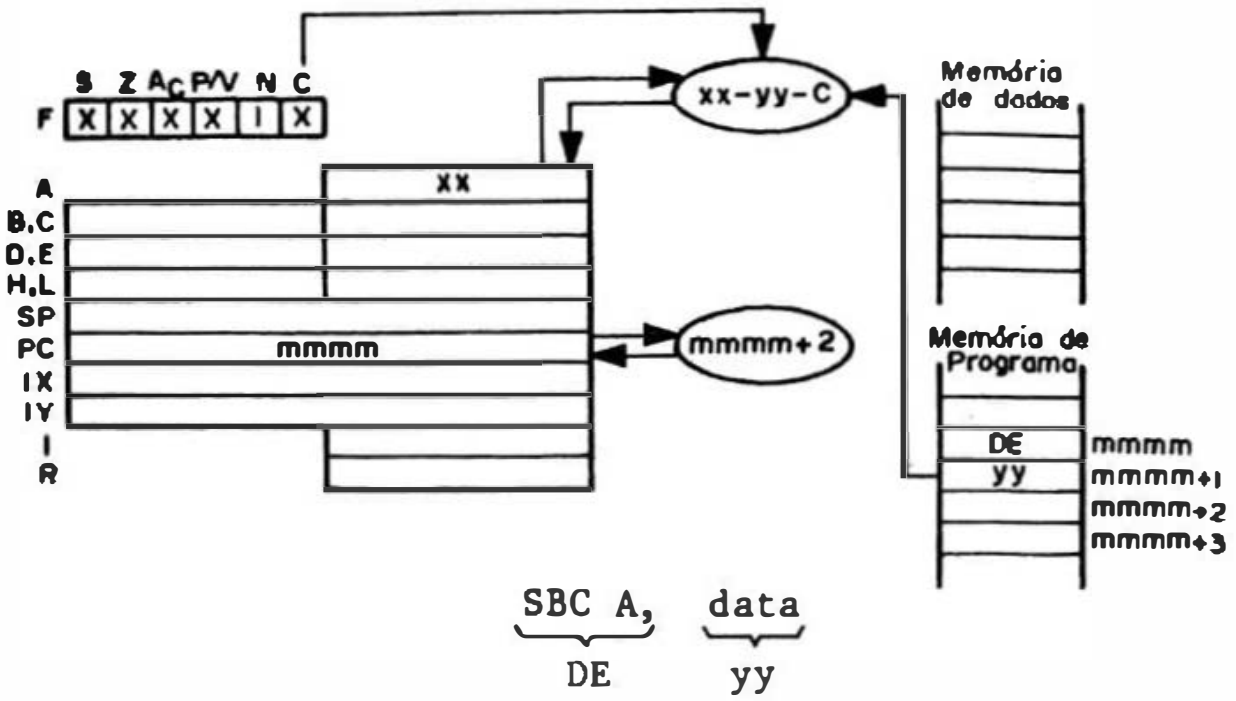
Formada por um único byte, o qual constitui o código de operação. Esta instrução "chama" a sub-rotina originada no endereço de memória especificada por "n".

Por exemplo, quando a instrução RST 18H for executada, a sub-rotina originada na localização de memória 0018_{16} é "chamada". Os conteúdos anteriores do Contador de Programa são empurrados (são levados) para o topo da pilha, de modo que o Contador de Programa (PC) assume um endereço pré-determinado, tendo uma variedade de 8 (oito) endereços.

A instrução RST é freqüentemente utilizada em conjunção com o processamento de interrupções.

Esta instrução pode ser utilizada, quando, em um programa qualquer, uma determinada sub-rotina é usada várias vezes. Essa sub-rotina pode ser chamada através da instrução RST de um único byte, ao invés de utilizarmos a instrução CALL de três bytes.

111) SBC A,data - SUBTRAIA IMEDIATAMENTE OS DADOS DO ACUMULADOR COM BORROW (EMPRÉSTIMO).



Constituída por dois bytes, sendo o primeiro o código de operação e o segundo um byte de dados. Esta instrução subtrai o conteúdo do acumulador com o byte de dados (data), menos o Carry bit, sendo o resultado carregado no acumulador.

Suponhamos que xx = $3A_{16}$ e Carry = 1. Após a execução da instrução SBC A,7CH, o acumulador passará a conter o valor: BD_{16}. Veja:

```
                              3A = 0011 1010
Complemento de dois de 7C =       1000 0100
Complemento de dois do Carry=     1111 1111
                                  1011 1101
```

1, S é ajustado para 1

Borrow, C é ajustado para 1

1 + 1 = 0, P/V é ajustado para 0.

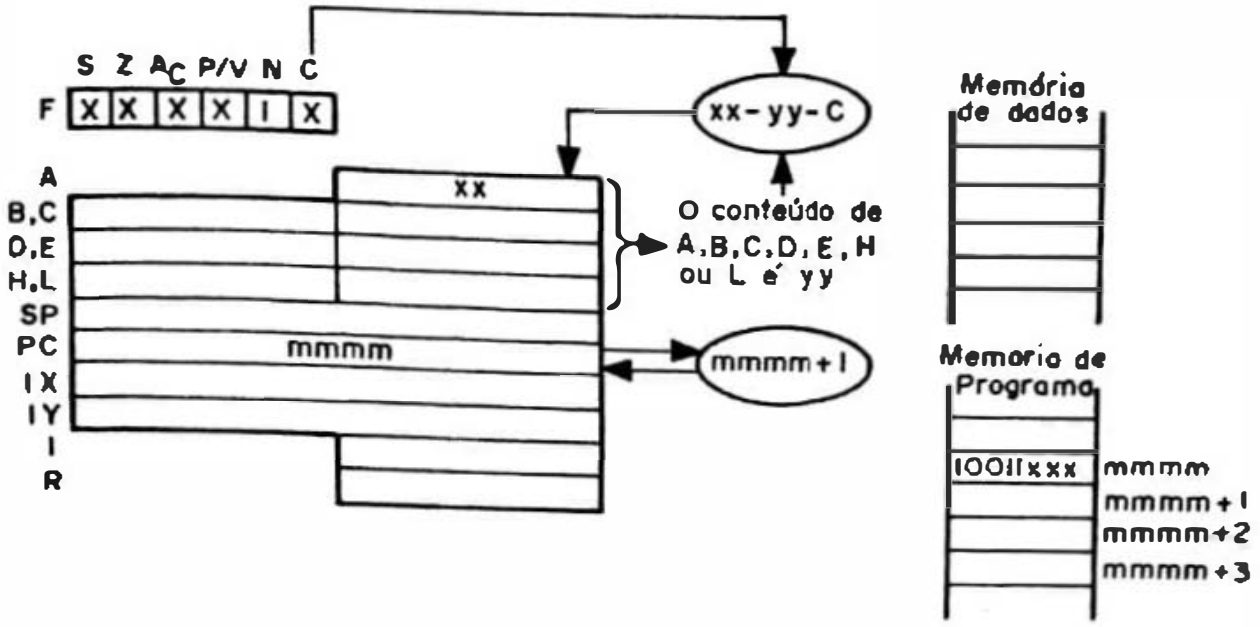
Resultado Não-zero, Z é ajustado para 0.

Borrow, AC é ajustado para 1.

Instrução de subtração, N é ajustado para 1.

Note que o Carry resultante é complementado. O flag de posição Carry é ajustado para 1 se houver borrow e para 0 (zero) se não houver borrow (empréstimo).

112) SBC A,reg - SUBTRAIA O REGISTRADOR DO ACUMULADOR COM BORROW (EMPRÉSTIMO).



SBC A, reg

10011 xxx

000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por um único byte, o qual constitui o código de operação. Esta instrução efetua a subtração entre o conteúdo do acumulador, o conteúdo do registrador especificado "reg" e o Carry, sendo o resultado carregado no próprio acumulador.

Suponhamos que xx = $E3_{16}$, o registrador E contenha $A0_{16}$ e o Carry igual a 1. Após a execução da instrução SBC A,E, o acumulador passará a conter 42_{16}. Veja:

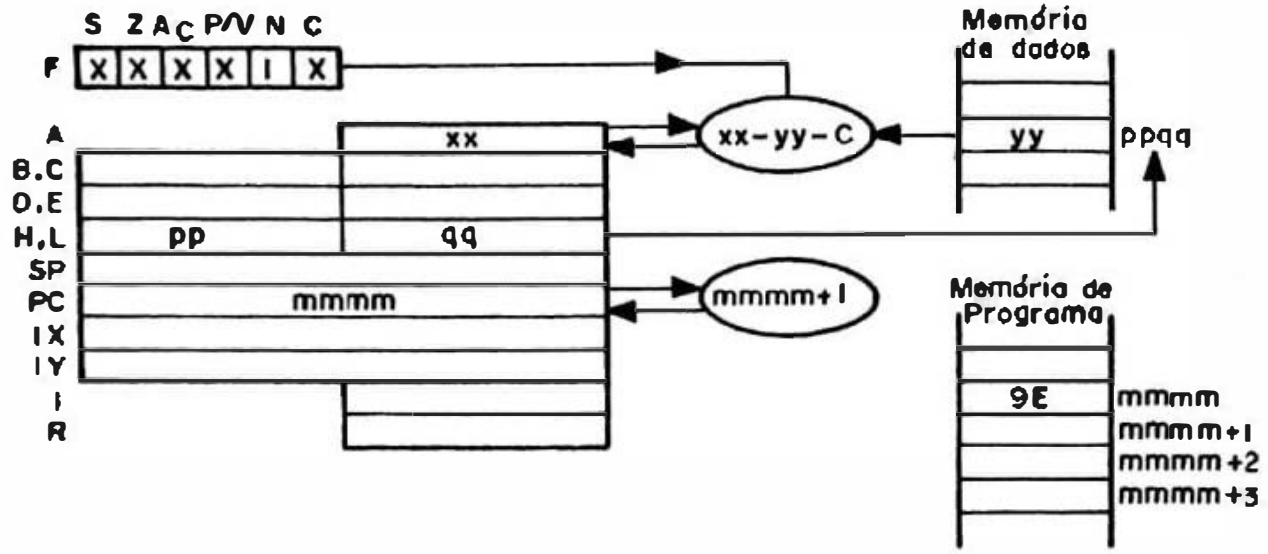
E3 = 1110 0011
Complemento de dois de A0 = 0110 0000
Complemento de dois de 1 = 1111 1111
0100 0010

0, S é ajustado para 0.
Não-borrow, C é ajustado
Para 0.
1 ⊻ 1 = 0, P/V é ajustado para 0.
Resultado Não-zero, Z é ajustado para 0.
Não-borrow, AC é ajustado para 0.

Instrução de Subtração, N é ajustado para 1.

Note que o Carry resultante é complementado. O flag de posição Carry é ajustado (set) para 1 se houver borrow (empréstimo) e para 0 (zero) se não houver borrow.

113) SBC A,(HL) - SUBTRAIA A MEMÓRIA E O CARRY DO ACUMULADOR.
SBC A,(IX+disp)
SBC A,(Iy+disp)



A ilustração nos mostra a execução da instrução SBC A, (HL):

$$\underbrace{\text{SBC A,(HL)}}_{\text{9E}}$$

Formada por um único byte, o qual constitui o código de operação. A instrução SBC A,(HL) subtrai do conteúdo do acumulador o conteúdo da localização de memória, endereçada pelo conteúdo do par de registradores HL e o Carry.

Suponhamos que o Carry seja igual a 0 (zero) e que ppqq= 4000_{16}, xx = $3A_{16}$ e a localização de memória 4000_{16} contenha $7C_{16}$. Após a execução da instrução SBC A,(HL), o acumulador passará a conter o valor BE_{16}. Veja:

```
                          3A = 0011 1010
Complemento de dois de    7C = 1000 0100
Complemento de dois do Carry =        0
                          ---------------
                          1011 1110
```

1, S é ajustado para 1 ← Resultado Não-zero, Z é ajustado (set) para 0

Borrow, C é ajustado para 1 ← Borrow, AC é ajustado para 1.

0 ⊻ 0 = 0, P/V é ajustado para 0.

Instrução de subtração, N é ajustado para 1.

O flag de posição Carry é ajustado (set) para 1 se houver borrow (empréstimo) e para 0 (zero) se não houver borrow.

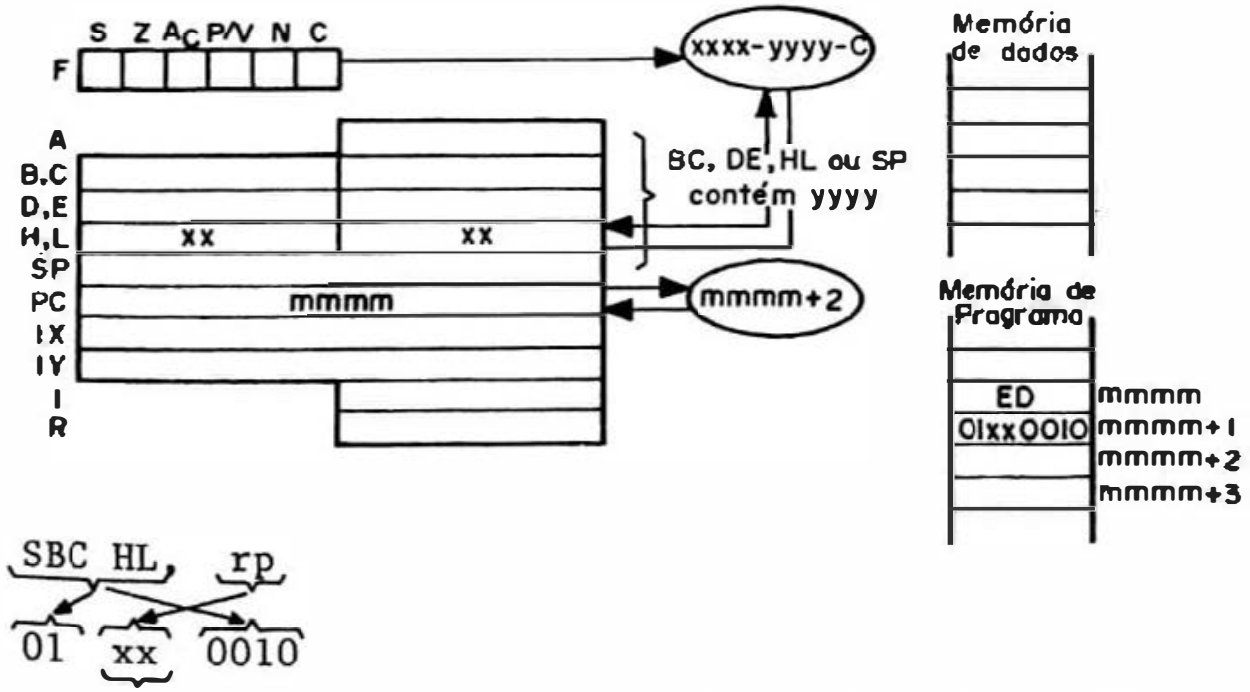
SBC A,(Ix + disp)
DD 9E d

A instrução SBC A,(Ix+disp) é formada por três bytes. Dois deles constituem o código de operação e o terceiro byte (disp) é somado ao registrador de índice Ix. Esta instrução, ao ser efetuada, subtrai do conteúdo do acumulador o conteúdo da posição de memória, endereçada pela soma do conteúdo do registrador Ix com o byte de deslocamento (disp), menos o Carry bit. O resultado é carregado no acumulador.

SBC A,(Iy + disp)
FD 9E d

A instrução SBC A,(Iy+disp) é idêntica à instrução SBC A, (Ix+disp), com exceção do registrador de índice Ix.

114) SBC HL,rp - SUBTRAIA O PAR DE REGISTRADORES COM O CARRY DE "H" E "L".



00 para rp igual ao par de registradores BC
01 para rp igual ao par de registradores DE
10 para rp igual ao par de registradores HL
11 para rp igual ao Stack-Pointer.

Formada por dois bytes, os quais constituem o código de operação. Esta instrução subtrai o conteúdo do par de registradores especificado, mais o Carry bit do conteúdo do par de registradores HL. O resultado é carregado no par de registradores HL.

Suponhamos que o par de registradores HL contenha $F4A2_{16}$, o par de registradores BC contenha $A034_{16}$ e o Carry seja igual a 0 (zero). Após a execução da instrução SBC HL,BC, o par de registradores HL passará a conter $546E_{16}$.

F4A2 = 1111 0100 1010 0010
Complemento de dois de A034 = 0101 1111 1100 1100
Complemento de dois do Carry= 0
0101 0100 0110 1110

0, S é ajustado para 0.

Não-borrow, C é ajustado para 0.

1 ⊻ 1 = 0, P/V é ajustado para 0.

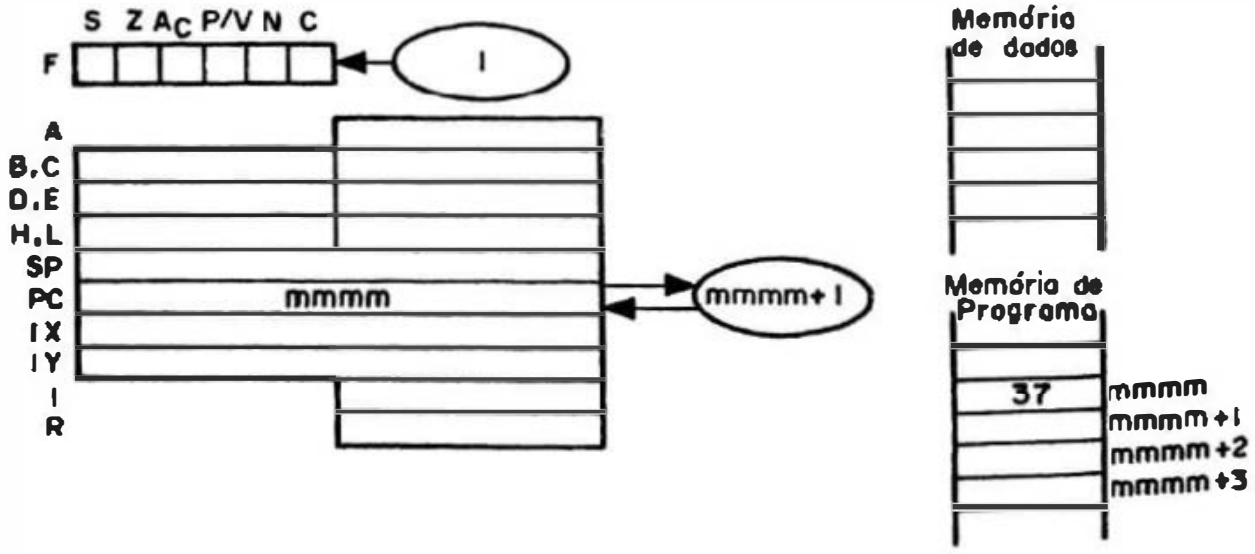
Resultado Não-zero, Z é ajustado(set) para 0.

Não-borrow.

Instrução de subtração, N é ajustado para 1.

O flag de posição Carry é ajustado (set) para 1 se houver borrow (empréstimo) e para 0 (zero) se não houver borrow.

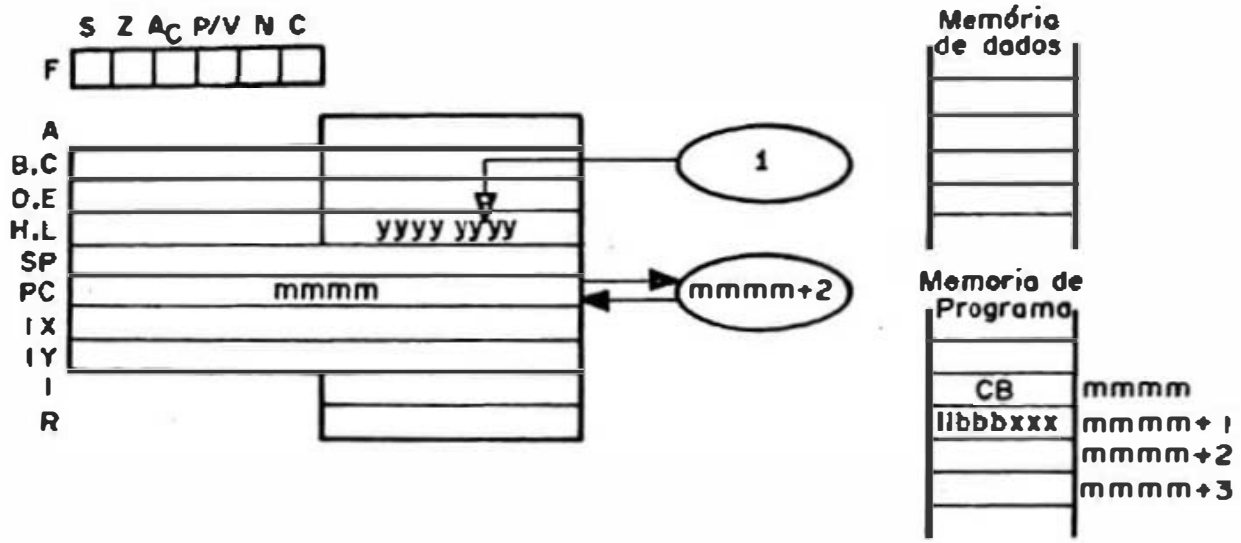
115) SCF - AJUSTE (SET) O FLAG CARRY.



SCF

Formada por um único byte, o qual constitui o código de operação. Esta instrução ajusta (set) a posição do Carry bit para 1 (um), sem considerar seu valor anterior. Nenhuma outra posição ou conteúdo do registrado é afetado.

116) SET b,reg - AJUSTE O BIT (b) INDICADO DO REGISTRADOR ESPECIFICADO.



SET b, reg

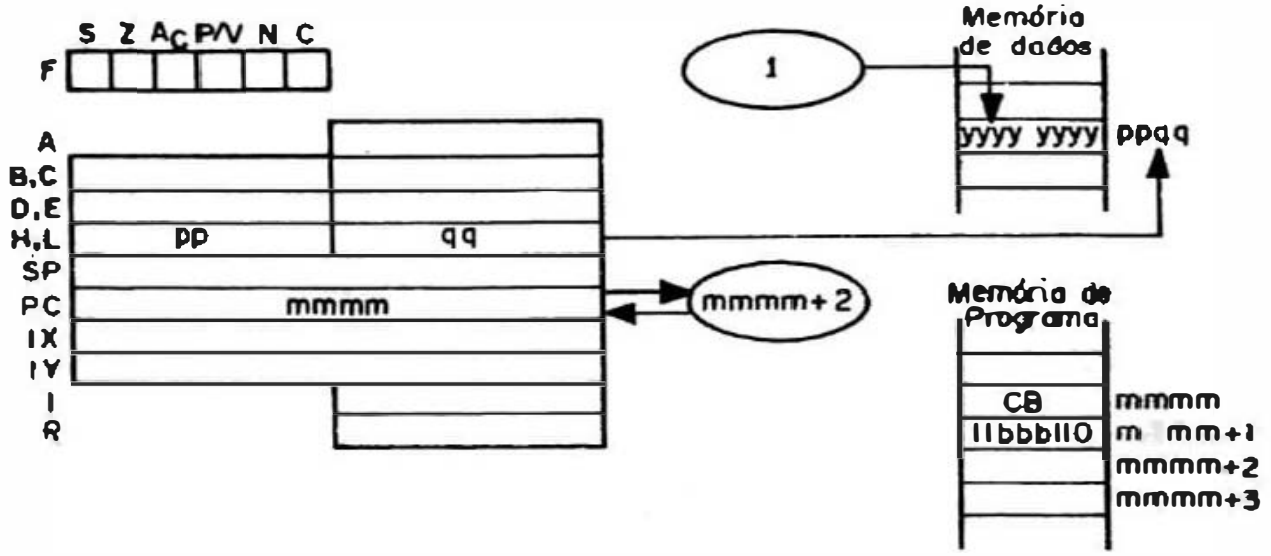
CB 11bbb xxx

Bit	bbb	xxx	Registrador
0	000	000	B
1	001	001	C
2	010	010	D
3	011	011	E
4	100	100	H
5	101	101	L
6	110	111	A
7	111		

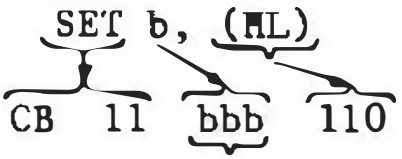
Formada por dois bytes, os quais constituem o código de operação. Esta instrução ajusta (set) para 1 o bit indicado (b), pertencente ao registrador especificado (reg). Em outras palavras, podemos dizer que esta instrução coloca um nível lógico alto (1) em um bit "b" do registrador "reg" especificado.

Por exemplo, a instrução SET 2,L, ao ser executada, ajusta (set) para 1 (um) o bit 2 do registrador L. Cabe ressaltar que o bit 0 (zero) é o bit menos significativo.

117) SET b,(HL) - AJUSTE O BIT (b) INDICADO DA POSIÇÃO DE MEMÓRIA ENDEREÇADA.
SET b,(Ix+disp)
SET b,(Iy+disp)



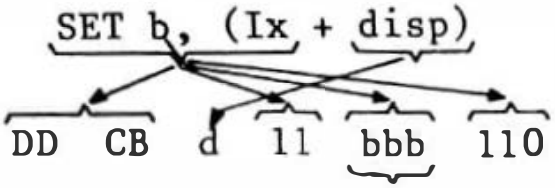
A ilustração nos mostra a execução da instrução SET b, (HL). O bit 0 (zero) é o bit menos significativo.



Bit ajustado(set)	bbb
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

A instrução SET b,(HL) é formada por dois bytes, os quais constituem o código de operação. Esta instrução ajusta (set) para 1 (um) o bit indicado (b) da posição de memória, endereçada pelo par de registradores HL.

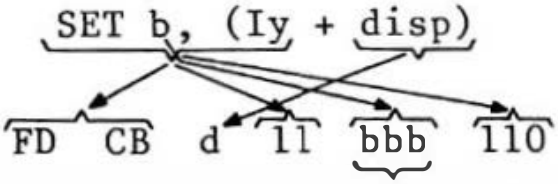
Suponhamos que o conteúdo do par de registradores HL seja 4000_{16}. Após a execução da instrução SET 5,(HL), o bit 5 da posição de memória 4000_{16} será ajustado (set) para 1 (um).



bbb → idem instrução SET b,(HL)

A instrução SET b,(Ix + disp) é formada por quatro bytes. Os dois primeiros e o quarto constituem o código de operação e o terceiro byte (disp) é somado ao conteúdo do registrador de índice Ix. Esta instrução ajusta (set) para 1(um) o bit indicado (b) da posição de memória, endereçada pela soma do conteúdo do registrador Ix, com o byte de deslocamento - (disp).

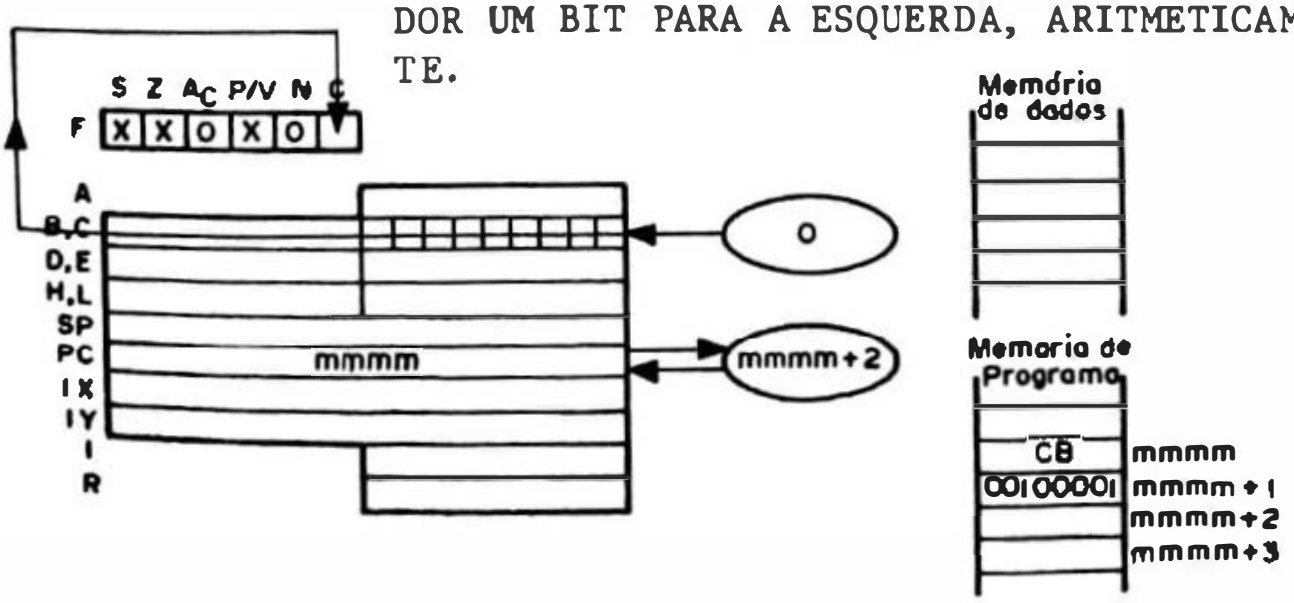
Suponhamos que o conteúdo do registrador Ix seja 4000_{16}. Após a execução da instrução SET 6,(Ix + 5H), o bit 6 da posição (ou localização) de memória 4005_{16} será ajustado (set) para 1 (um).



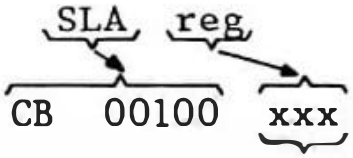
bbb → idem instrução SET b,(HL)

A instrução SET b,(Iy+disp) é idêntica à instrução SET b, (Ix+disp) com exceção do registrador de índice Ix.

118) SLA reg - DESLOQUE (SHIFT) O CONTEÚDO DO REGISTRADOR UM BIT PARA A ESQUERDA, ARITMETICAMENTE.



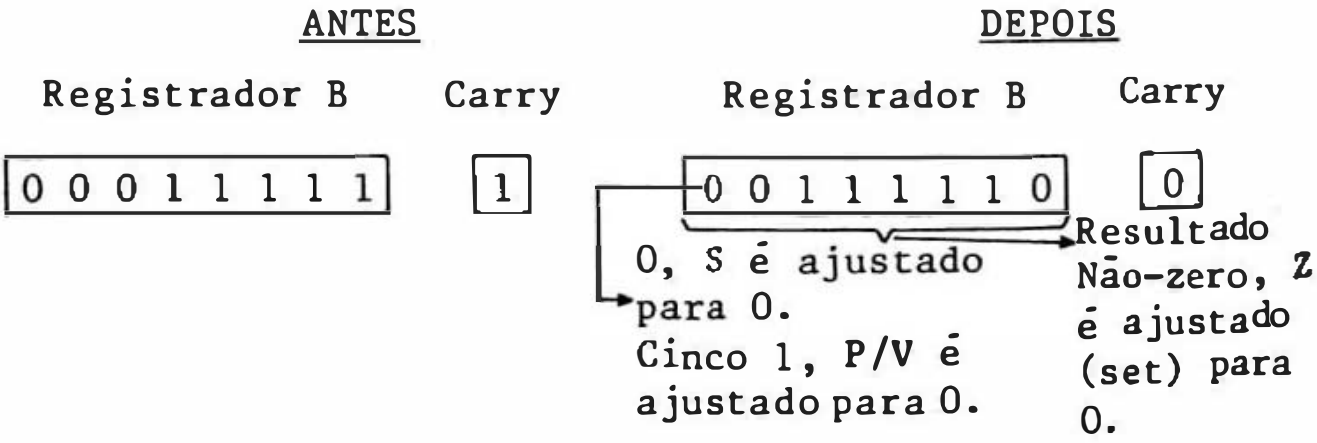
A ilustração nos mostra a execução da instrução SLA C:



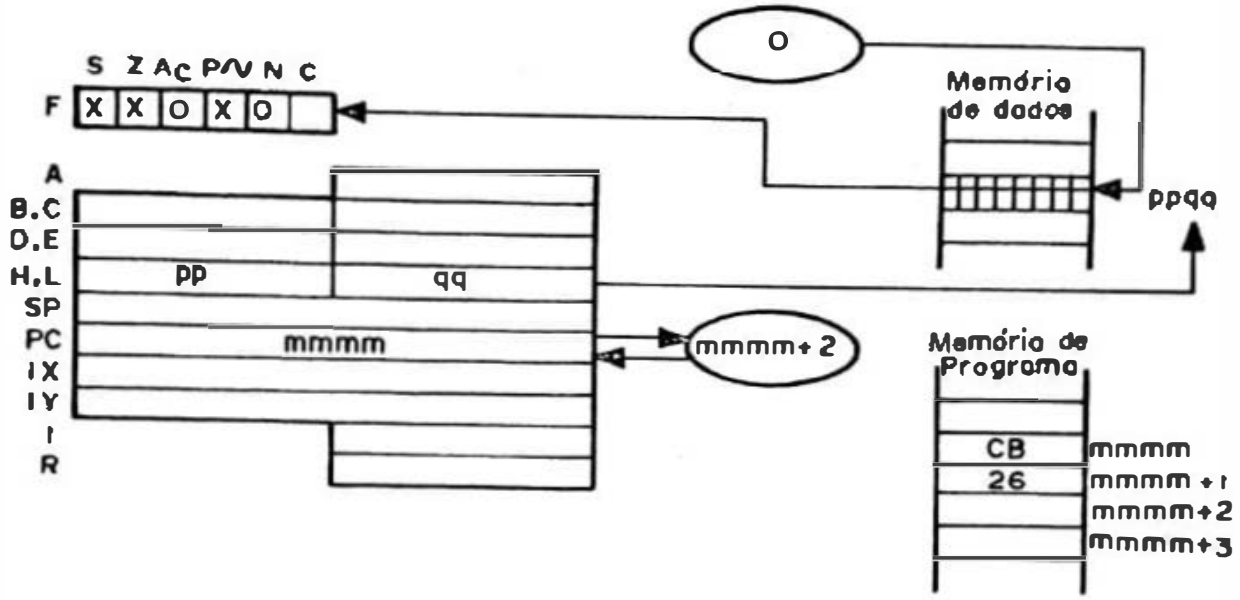
000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca o conteúdo do registrador, especificado de um bit (ou uma posição), para a esquerda, aritmeticamente, carregando o bit 7 no Carry bit e zerando (reset) o bit menos significativo (bit 0).

Suponhamos que o conteúdo do registrador B seja $1F_{16}$ e que o Carry seja igual a 1. Após a execução da instrução SLA B, o registrador B passará a conter $3E_{16}$ e o Carry será igual a 0 (zero). Veja:



119) SLA (HL) - DESLOQUE (SHIFT) O CONTEÚDO DA LOCA-
SLA (Ix+disp) LIZAÇÃO DE MEMÓRIA UM BIT PARA A ES-
SLA (Iy+disp) QUERDA, ARITMETICAMENTE.

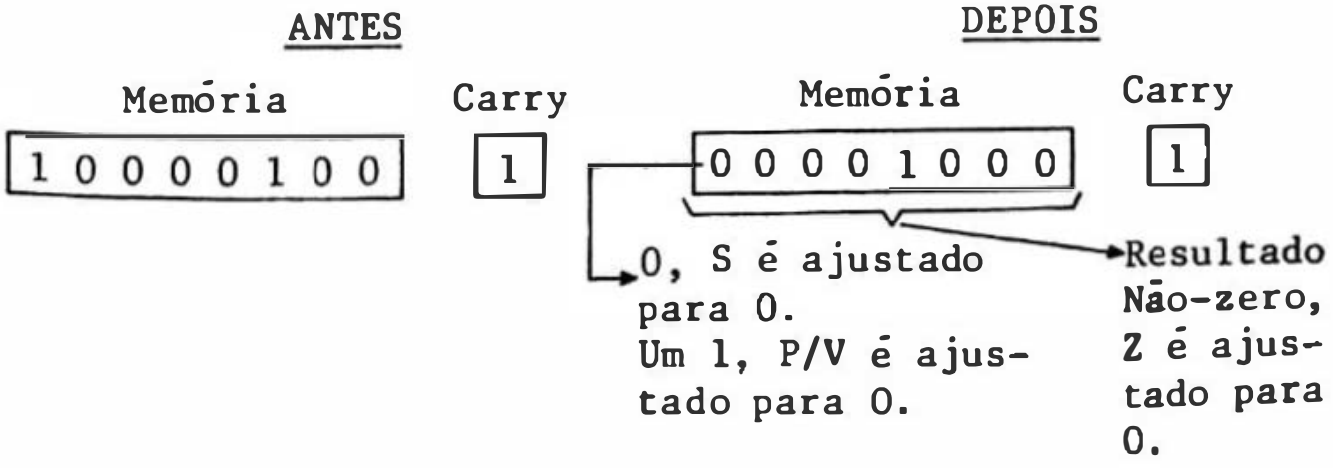


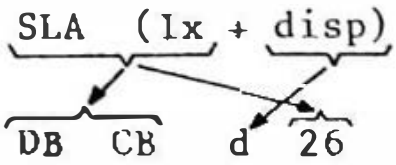
A ilustração nos mostra a execução da instrução SLA (HL):

$$\underbrace{\text{SLA (HL)}}_{\text{CB}\quad 26}$$

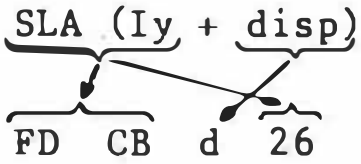
A instrução SLA (HL) é formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca o conteúdo da localização de memória (endereçada pelo conteúdo do par de registradores HL) um bit (ou uma posição) para a esquerda, aritmeticamente, zerando (reset) o bit 0 (zero) e carregando o bit 7 no Carry bit.

Suponhamos que o par de registradores HL contenha 4500_{16}, a localização de memória 4500_{16} contenha 84_{16} e o Carry contenha 0 (zero). Após a execução da instrução SLA (HL), a localização de memória 4500_{16} passará a conter 08_{16} e o Carry bit será igual a 1 (um).



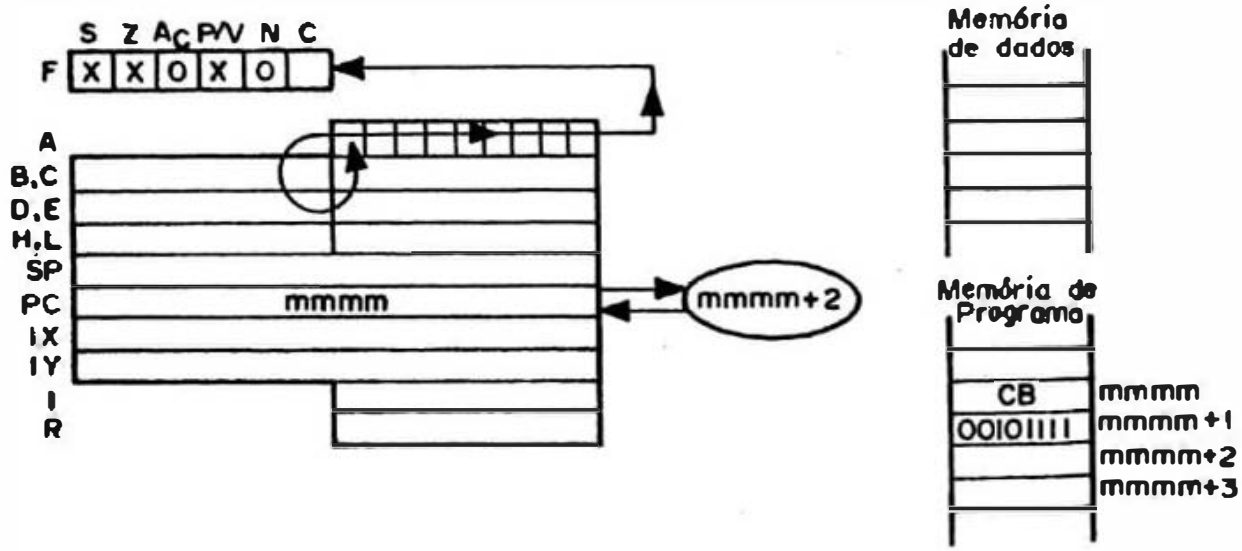


A instrução SLA (Ix + disp) é formada por quatro bytes. Os dois primeiros e o quarto constituem o código de operação e o terceiro byte (disp) é somado ao conteúdo do registrador de índice Ix. Esta instrução desloca o conteúdo da localização de memória (endereçada pela soma do conteúdo do registrador Ix com o byte de deslocamento) um bit (ou uma posição) para a esquerda, aritmeticamente, zerando (reset) o bit 0 (zero) e carregando o bit 7 no Carry bit.

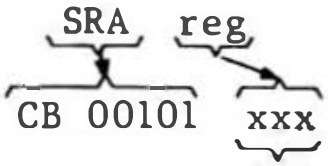


A instrução SLA (Iy + disp) é idêntica à instrução SLA (Ix + disp), com exceção do registrador de índice Ix.

120) SRA reg - DESLOQUE (SHIFT) O CONTEÚDO DO REGISTRADOR UM BIT PARA A DIREITA, ARITMETICAMENTE.



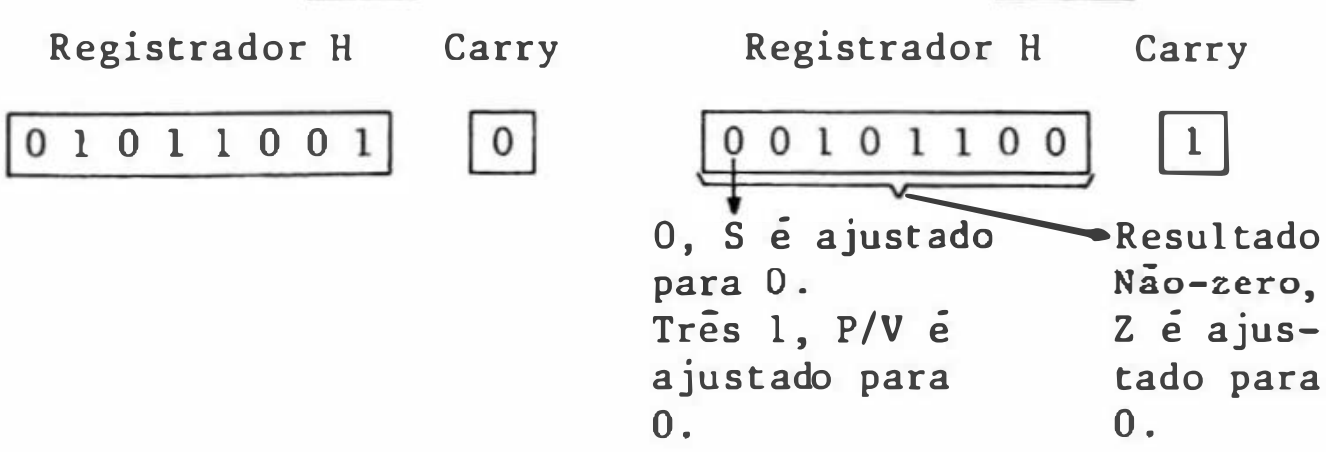
A ilustração nos mostra a execução da instrução SRA A:



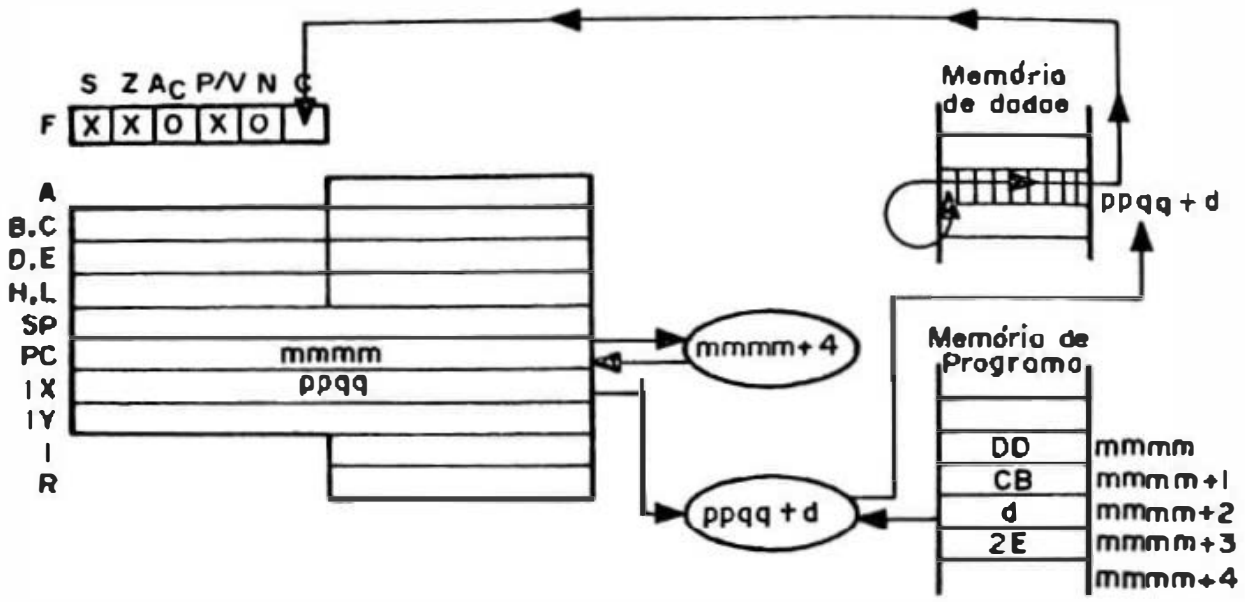
000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca o conteúdo do registrador, especificado de um bit (ou uma posição) para a direita, aritmeticamente, carregando o bit 7 com o seu próprio valor (ou seja, o bit 7 não é alterado) e o bit 0 (zero) será carregado no carry bit.

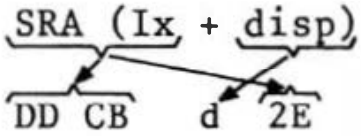
Suponhamos que o conteúdo do registrador H seja 59_{16} e que o Carry seja igual a 0 (zero). Após a execução da instrução SRA H, o registrador H passará a conter $2C_{16}$ e o Carry será igual a 1. Veja:



121) SRA (HL) - DESLOQUE (SHIFT) O CONTEÚDO DA LOCA-
SRA (IX+disp) LIZAÇÃO DE MEMÓRIA UM BIT PARA A DI-
SRA (Iy+disp) REITA, ARITMETICAMENTE.

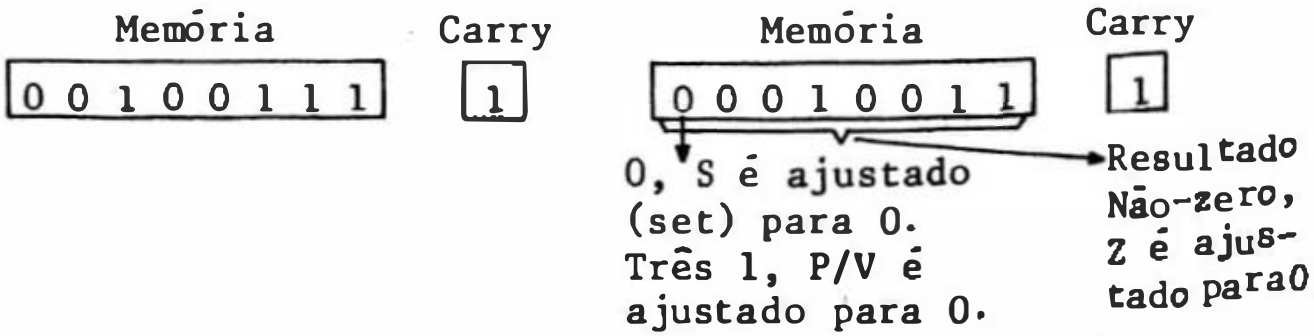


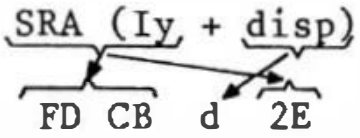
A ilustração nos mostra a execução da instrução SRA (Ix+ disp):



A instrução SRA (Ix+disp) é formada por quatro bytes. Os dois primeiros e o quarto constituem o código de operação e o terceiro byte (disp) é somado ao conteúdo do registrador de índice Ix. Esta instrução desloca o conteúdo da localização de memória (especificada ou endereçada pelo conteúdo do registrador Ix somado ao byte de deslocamento), de uma posição ou de um bit para a direita. O bit mais significativo (bit 7) será carregado com seu próprio valor.

Suponhamos que o registrador Ix contenha 3400_{16} e que a localização de memória $34AA_{16}$ contenha o valor 27_{16} e o Carry seja igual a 1 (um). Após a execução da instrução SRA (Ix + 0AAH), a localização de memória $34AA_{16}$ passará a conter 13_{16} e o Carry será igual a 1 (um). Veja:



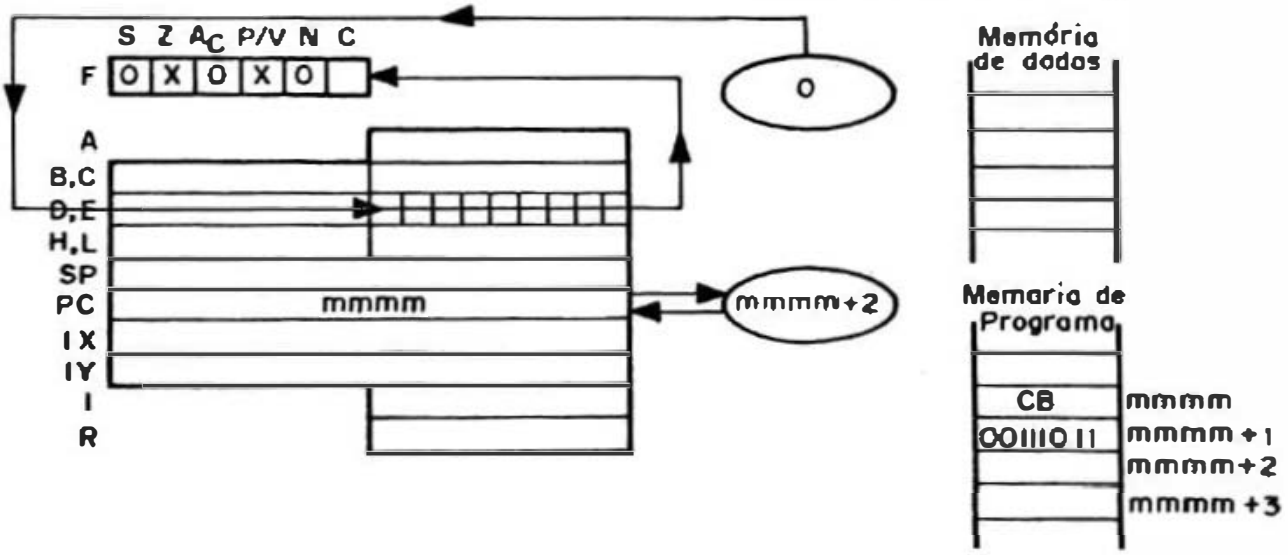


A instrução SRA (Iy + disp) é idêntica à instrução SRA (Ix + disp), com exceção do registrador de índice Ix.

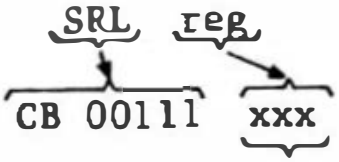
SRA (HL)
CB 2E

A instrução SRA (HL) é formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca o conteúdo da localização de memória, endereçada (ou especificada) pelo conteúdo do par de registradores HL, um bit ou uma posição para a direita. O bit mais significativo (bit 7) será carregado com o seu próprio valor.

122) SRL reg - DESLOQUE, LOGICAMENTE, PARA A DIREITA O CONTEÚDO DO REGISTRADOR ESPECIFICADO.



A ilustração nos mostra a execução da instrução SRL E:



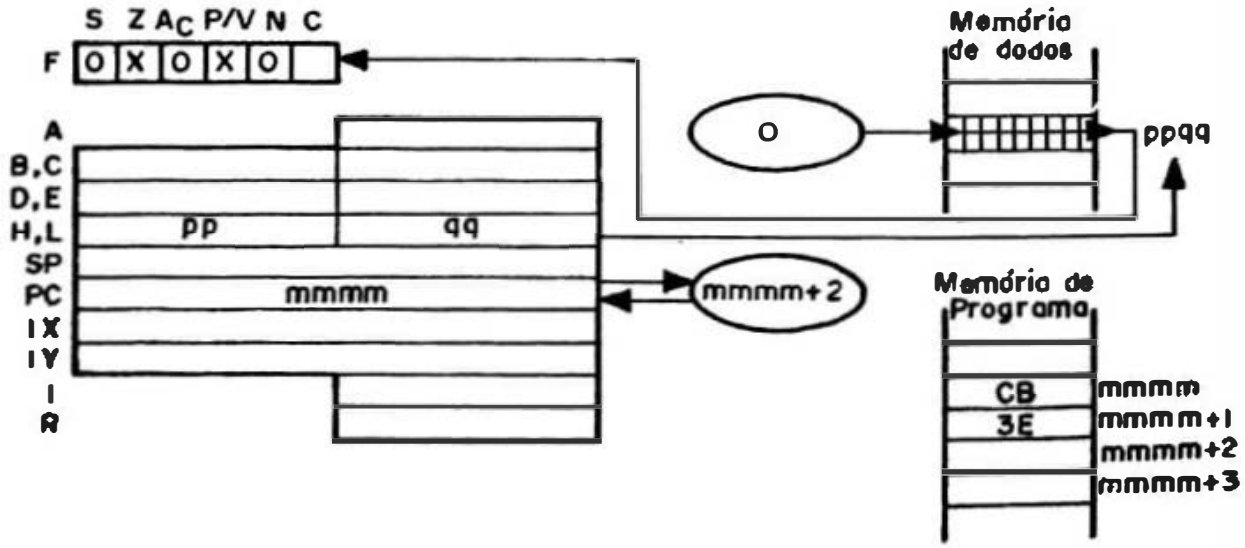
000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca o conteúdo do registrador especificado, um bit ou uma posição para a direita. O bit mais significativo (bit 7) é carregado com zero, ou seja, é zerado (reset), enquanto que o bit menos significativo (bit 0) é carregado no Carry bit.

Suponhamos que o conteúdo do registrador D contenha $1F_{16}$ e que o Carry seja igual a 0 (zero). Após a execução da instrução SRL D, o registrador D passará a conter $0F_{16}$ e o Carry será igual a 1 (um). Veja:

ANTES		DEPOIS	
Registrador D	Carry	Registrador D	Carry
0 0 0 1 1 1 1 1	0	0 0 0 0 1 1 1 1	1
		Quatro 1, P/V é ajustado para 1.	Resultado Não-zero, Z é ajustado para 0.

123) SRL (HL) - DESLOQUE, LOGICAMENTE, PARA A DIREITA O CONTEÚDO DA LOCALIZAÇÃO DE MEMÓRIA ENDEREÇADA.
SRL (Ix+disp)
SRL (Iy+disp)



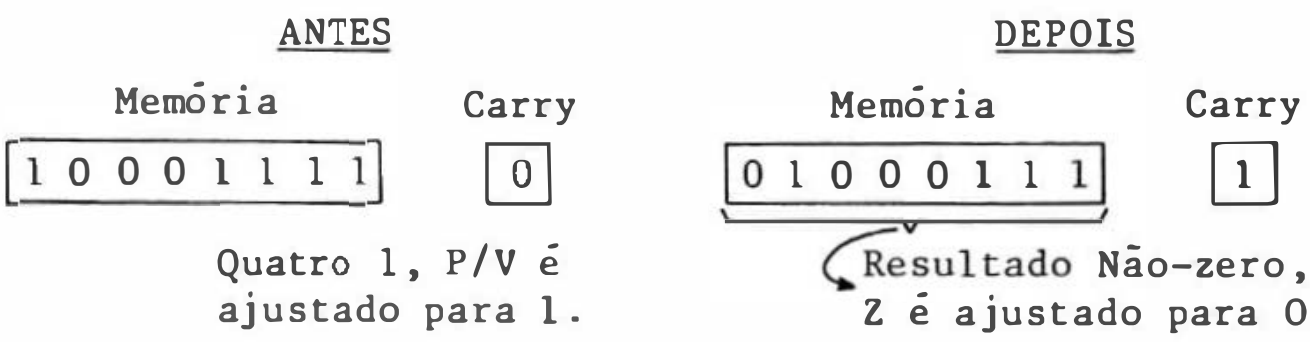
A ilustração nos mostra a execução da instrução SRL (HL):

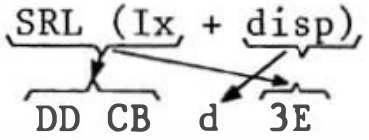
SRL (HL)

CB 3E

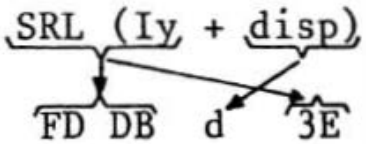
Formada por dois bytes, os quais constituem o código de operação. Esta instrução desloca o conteúdo da localização de memória, endereçada pelo par de registradores HL, de uma posição (ou de um bit) para a direita. O bit mais significativo - (bit 7) é zerado (reset).

Suponhamos que o par de registradores HL contenha 2000_{16} e o conteúdo da localização de memória 2000_{16} seja $8F_{16}$ e que o Carry seja igual a 0 (zero). Após a execução da instrução SRL (HL), a localização de memória 2000_{16} passará a conter 47_{16} e o Carry será igual a 1 (um). Veja o demonstrativo, referente ao exemplo, que acabamos de citar:



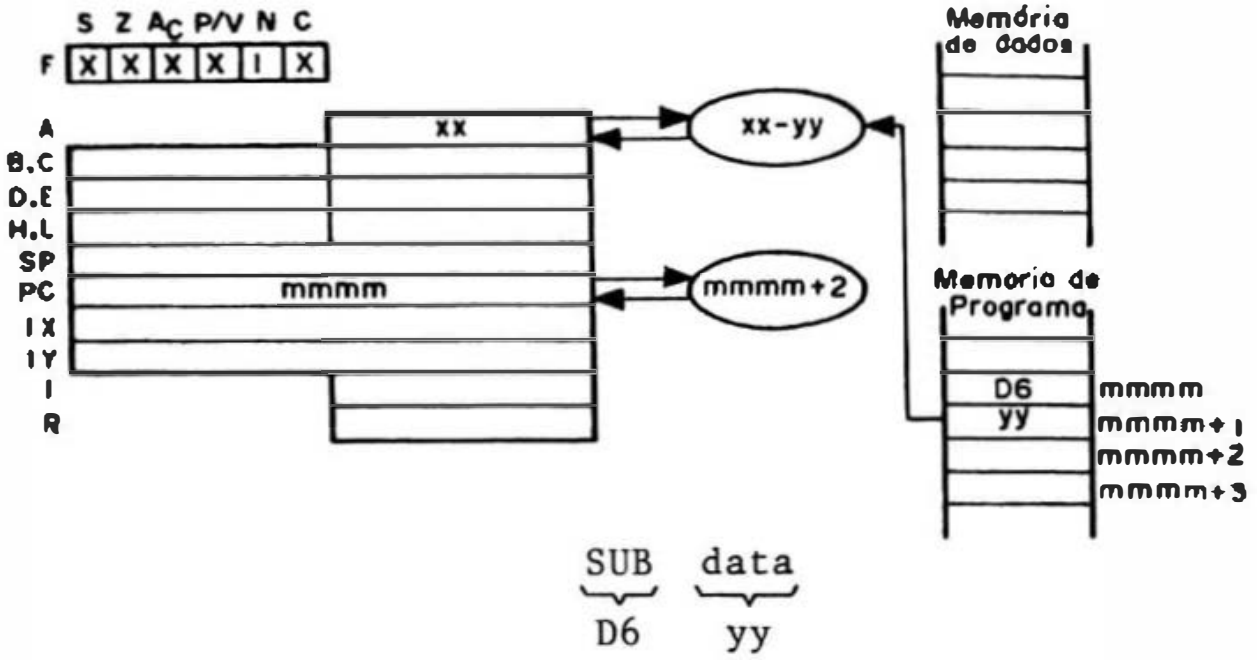


A instrução SRL (Ix+disp) é formada por quatro bytes. Os dois primeiros e o quarto constituem o código de operação e o terceiro byte (disp) é somado ao conteúdo do registrador Ix. Esta instrução desloca o conteúdo da localização de memória (endereçada pela soma do conteúdo de Ix com o conteúdo do terceiro byte) um bit ou uma posição para a direita. O bit mais significativo (bit 7) é zerado (reset).



A instrução SRL (Iy + disp) é idêntica à instrução SRL (Ix + disp), com exceção do registrador de índice Ix.

124) SUB data - SUBTRAIA IMEDIATAMENTE DO ACUMULADOR.



Formada por dois bytes. O primeiro constitui o código de operação e o segundo, o byte de dados. Esta instrução subtrai do conteúdo do acumulador o conteúdo do segundo byte do código objeto da instrução, ou seja, do byte de dados.

Suponhamos que xx = $3A_{16}$. Após a execução da instrução SUB 7CH, o acumulador passará a conter BE_{16}. Vejamos:

```
                    3A = 0011 1010
Complemento de dois de 7C = 1000 0100
                         1011 1110
```

1, S é ajustado para 1

Borrow, C é ajustado para 1

Resultado Não-zero, Z é ajustado para 0.

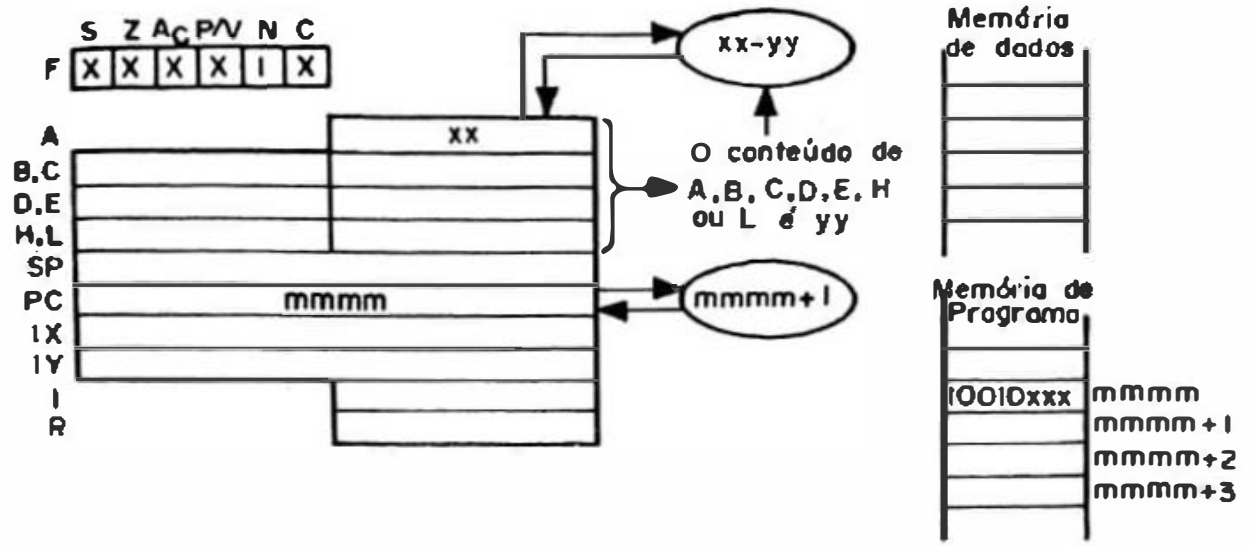
Borrow, AC é ajustado para 1.

0 ⊻ 0 = 0, P/V é ajustado para 0.

Instrução de subtração, N é ajustado para 1.

Note que o Carry resultante é complementado.

125) SUB reg - SUBTRAIA DO ACUMULADOR O CONTEÚDO DO REGISTRADOR.



SUB reg
10010 xxx

000 para reg = B
001 para reg = C
010 para reg = D
011 para reg = E
100 para reg = H
101 para reg = L
111 para reg = A

Formada por um único byte, o qual constitui o código de operação. Esta instrução subtrai do conteúdo do acumulador o conteúdo do registrador "reg" especificado e o resultado é guardado (ou carregado) no acumulador.

Suponhamos que xx = E3 e que o registrador H contenha $A0_{16}$. Após a execução da instrução SUB H, o acumulador passará a conter 43_{16}. Vejamos:

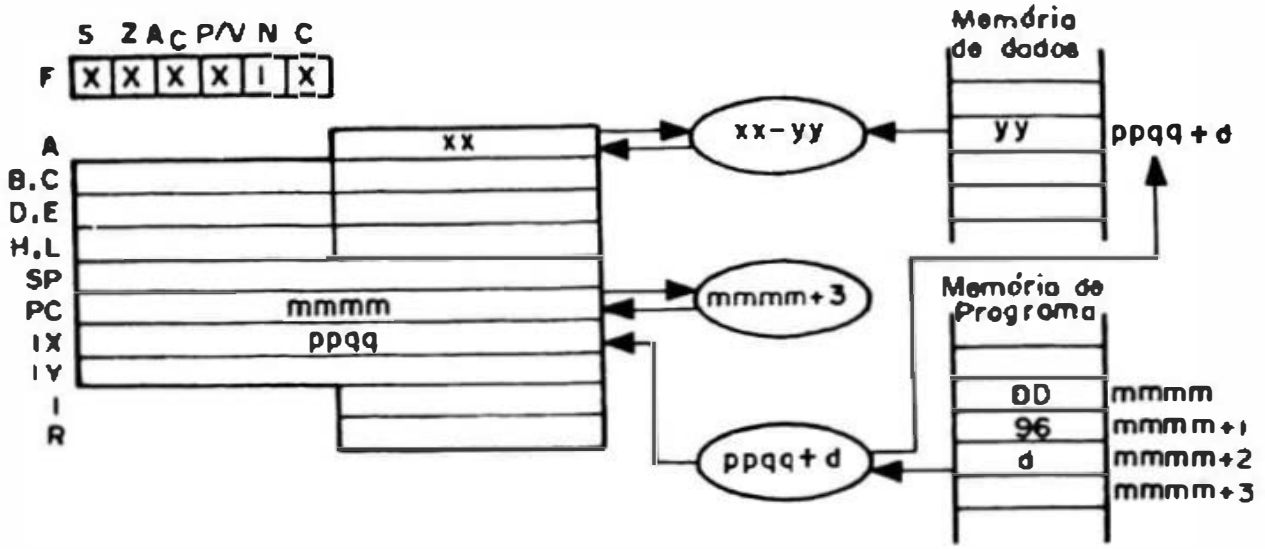
E3 = 1110 0011
Complemento de dois de A0 = 0110 0000
0100 0011

0, S é ajustado para 0.
Não-borrow, C é ajustado para 0.
Resultado Não-zero, Z é ajustado para 0.
Não-borrow, AC é ajustado para 0.
1 ⊻ 1 = 0, P/V é ajustado para 0.

Instrução de subtração, N é ajustado para 1.

Note que o Carry resultante é complementado.

126) SUB (HL) - SUBTRAIA DO ACUMULADOR O CONTEÚDO DA
SUB (Ix+disp) MEMÓRIA.
SUB (Iy+disp)



A ilustração nos mostra a execução da instrução SUB (Ix+d):

$$\underbrace{\text{SUB (Ix}}_{\text{DD 96}} + \underbrace{\text{disp)}}_{\text{d}}$$

A instrução SUB (Ix + disp) é formada por três bytes. Os dois primeiros constituem o código de operação e o terceiro (disp) é o byte de deslocamento, que será somado ao conteúdo do registrador de índice Ix. Esta instrução subtrai o conteúdo do acumulador do conteúdo da localização de memória, endereçada pelo registrador de índice Ix, somado ao byte de deslocamento (disp).

Suponhamos que ppqq = 4000_{16}, xx = FF_{16} e que a localização de memória $40FF_{16}$ contenha 50_{16}. Após a execução da instrução SUB (Ix + OFFH), o acumulador passará a conter o valor AF_{16}. Vejamos:

FF = 1111 1111
Complemento de dois de 50 = 1011 0000
1010 1111

1, S é ajustado para 1
Não-borrow, C é ajustado para 0.
Resultado Não-zero, Z é ajustado para 0.
Não-borrow, AC é ajustado para 0.
1 ⊕ 1 = 0, P/V é ajustado para 0.
Instrução de subtração, N é ajustado para 1.
Note que o Carry resultante é complementado.

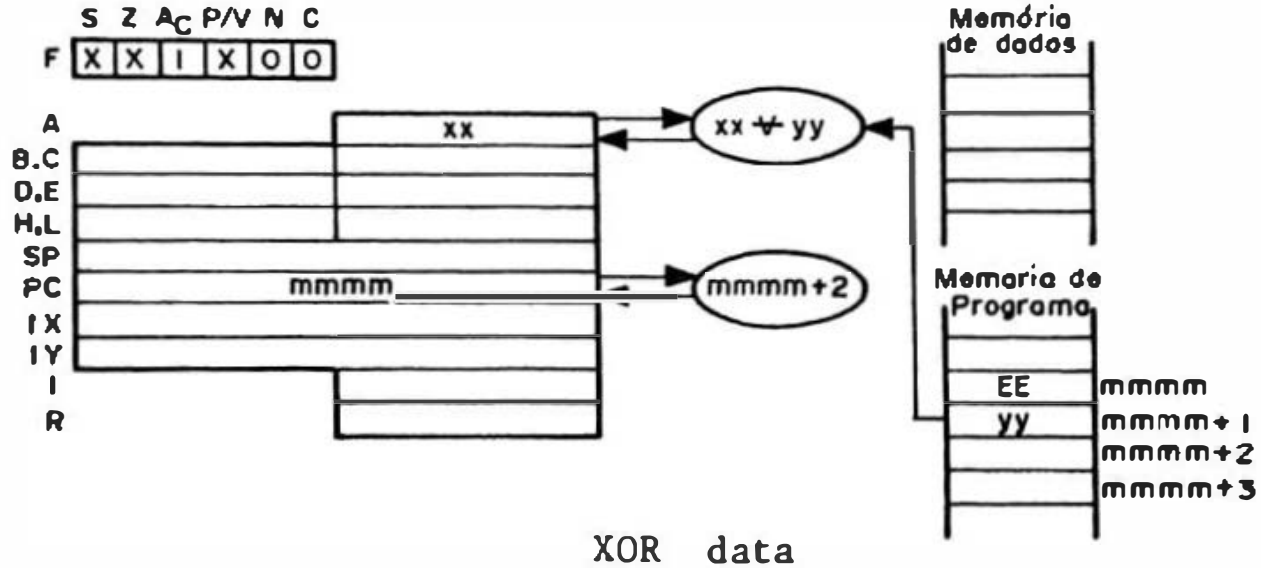
SUB (Iy + disp)
FD 96 d

A instrução SUB (Iy + disp) é idêntica à instrução SUB (Ix + disp), com exceção do registrador de índice Ix.

SUB (HL)
96

Formada por um único byte, o qual constitui o código de operação. A instrução SUB (HL) subtrai do conteúdo do acumulador o conteúdo da localização de memória, endereçada pelo par de registradores HL.

127) XOR data - "OU EXCLUSIVO" IMEDIATAMENTE COM O CONTEÚDO DO ACUMULADOR.



Formada por um único byte, o qual constitui o código de operação. Esta instrução executa a lógica "OU-Exclusiva" entre o conteúdo do acumulador e o byte de dados (data) e o resultado é carregado (ou guardado) no acumulador.

Suponhamos que xx = $3A_{16}$. Após a execução da instrução XOR 7CH, o acumulador passará a conter 46_{16}. Vejamos:

```
3A = 0011 1010
7C = 0111 1100
     0100 0110
```

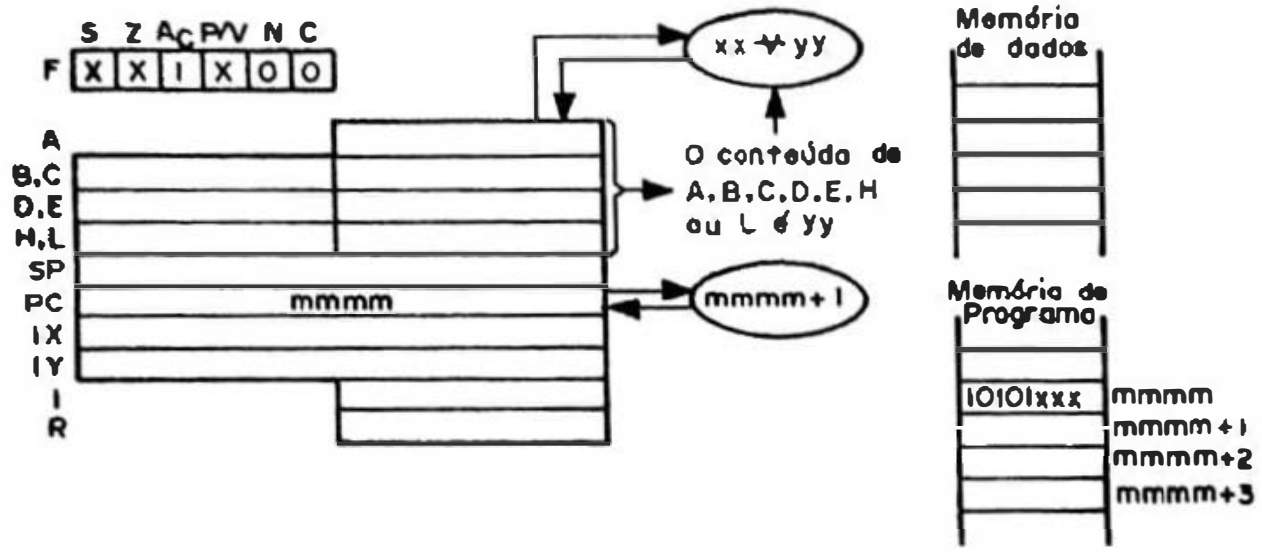
0, S é ajustado para 0.

Resultado Não-zero, Z é ajustado para 0.

Três bits 1, P/V é ajustado para 0.

A instrução "OU-Exclusiva" é utilizada para testar a posição de bit, por mudanças.

128) XOR reg - "OU EXCLUSIVO" ENTRE O ACUMULADOR E O CONTEÚDO DO REGISTRADOR.




```
XOR  reg
10101 xxx
      000  para reg = B
      001  para reg = C
      010  para reg = D
      011  para reg = E
      100  para reg = H
      101  para reg = L
      111  para reg = A
```

Formada por um único byte, o qual constitui o código de operação. Esta instrução executa a lógica "OU EXCLUSIVO" entre o conteúdo do acumulador e o conteúdo do registrador "reg" especificado. O resultado é carregado no acumulador.

Suponhamos que xx = $E3_{16}$ e que o conteúdo do registrador E seja $A0_{16}$. Após a execução da instrução XOR E, o acumulador passará a conter o valor 43_{16}. Vejamos:

```
E3 = 1110 0011
A0 = 1010 0000
     0100 0011
```

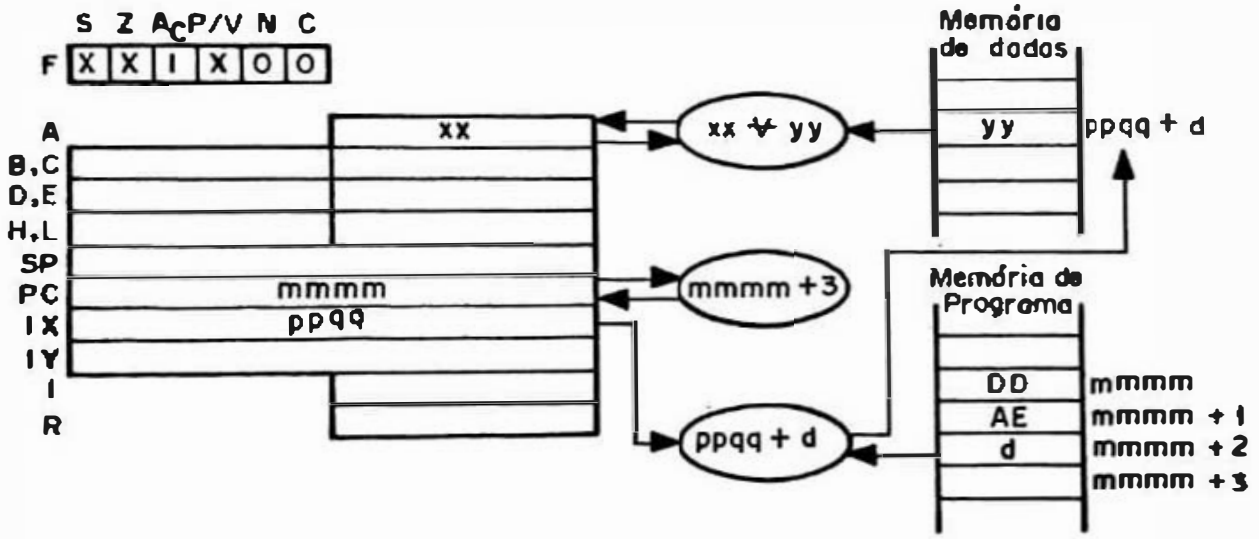
0, S é ajustado para 0.

Resultado Não-zero, Z é ajustado para 0.

Três bits 1, P/V é ajustado para 0.

A instrução "OU-Exclusivo" é utilizada para testar a posição de bit, por mudanças.

129) XOR (HL) - "OU-EXCLUSIVO" ENTRE O ACUMULADOR E
XOR (Ix+disp) A MEMÓRIA ENDEREÇADA.
XOR (Iy+disp)



A ilustração nos mostra a execução da instrução XOR (Ix+ disp):

$$\underbrace{\text{XOR (Ix}}_{\text{DD AE}} + \underbrace{\text{disp)}}_{\text{d}}$$

A instrução XOR (Ix + disp) é formada por três bytes. Os dois primeiros constituem o código de operação e o terceiro byte (disp) é somado ao conteúdo do registrador de índice Ix. Esta instrução executa a lógica "OU-Exclusiva" entre o conteúdo da localização de memória, endereçada por "Ix + disp" e o conteúdo do acumulador.

Suponhamos que xx = $E3_{16}$, ppqq = 4500_{16} e a localização de memória $45FF_{16}$ contenha $A0_{16}$. Após a execução da instrução XOR (Ix + OFFH), o acumulador passará a conter 43_{16}. Vejamos:

```
E3 = 1110 0011
A0 = 1010 0000
     0100 0011
```

0, S é ajustado para 0.

Resultado Não-zero, Z é ajustado para 0.

Três bits 1, P/V é ajustado para 0.

$$\underbrace{\text{XOR (Iy}}_{\text{FD AE}} + \underbrace{\text{disp)}}_{\text{d}}$$

A instrução XOR (Iy + disp) é idêntica à instrução XOR (Ix + disp), com exceção do registrador de índice Ix.

$$\underbrace{\text{XOR (HL)}}_{\text{AE}}$$

A instrução XOR (HL) é formada por um único byte, o qual constitui o código de operação. Esta instrução executa a lógica "OU-Exclusiva" entre o conteúdo do acumulador e o conteúdo da localização de memória, endereçada pelo par de registradores HL.

O item 129 foi o último sub-conjunto de instruções do microprocessador Z-80. Finalizamos, portanto, o estudo do conjunto de instruções do Z-80.

Antes, porém, pretendemos fazer um pequeno adendo com relação ao "Complemento a um" e "Complemento a dois", este, inclusive, foi citado no estudo do conjunto de instruções do Z-80. Vejamos, portanto, em que consiste cada um deles:

COMPLEMENTO A UM:

Neste caso, todo número inteiro positivo é representado na forma usual. Por exemplo: o número $+15_{10}$ é representado por 1111_2. Contudo, o seu complemento -15_{10} é obtido pela complementação bit a bit da representação original, representado por

0000_2. Não esqueça que o complemento do número 1 é o número 0 e que o complemento do número 0 é o número 1.

COMPLEMENTO A DOIS:

Neste caso, os números positivos são representados na forma usual. Por exemplo: o número $+11_{10}$ em complemento a dois é representado por 1011_2. Já, o complemento a dois de um número negativo é obtido pela aplicação da regra de "complemento a um", seguido da adição de um 1. Vamos então determinar o "complemento a dois" do número -11_{10}.

- representação usual (+11) = 1011
- complemento a um = 0100
- complemento a dois = 0101

OBS.: As denominações "complemento de um" e "complemento de dois" também são muito utilizadas.

CEM
CURSOS DE APERFEIÇOAMENTO

CURSO DE
ELETRÔNICA DIGITAL E
MICROPROCESSADORES

CAIXA POSTAL 1642 CEP 86100 LONDRINA - PARANÁ

EXAME DA LIÇÃO MC-31

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) A instrução RRC (Ix + disp) é constituída por:

a) apenas um byte;
b) dois bytes;
c) três bytes;
d) quatro bytes.

2) A instrução capaz de deslocar o conteúdo do acumulador um bit para a direita, de forma circular, é:

a) RRD;
b) RRCA;
c) SBC A,reg;
d) N.R.A.

3) Quando a instrução RST 30H é executada, a sub-rotina originada na localização de memória:

a) 0030_{16} é chamada;

b) 0018_{16} é chamada;

c) 0030_{16} não é chamada;

d) os itens b e c estão corretos.

4) A instrução RST é freqüentemente utilizada em conjunção com o processamento de:

a) dados;
b) habilitações;
c) interrupções;
d) N.R.A.

5) A instrução "SBC A,(HL)" subtrai do conteúdo do acumulador o conteúdo da localização de memória endereçada:

a) pelo registrador de índice Ix;
b) pelo registrador D;
c) pelo par de registradores DE;
d) pelo par de registradores HL.

6) O resultado oriundo da execução da instrução SBC A, (Ix + disp) é:

a) carregado no acumulador;
b) carregado na localização de memória "Ix + disp";
c) carregado na localização de memória "Iy + disp";
d) N.R.A.

7) A instrução SBC HL,rp é constituída por:

a) apenas um byte;
b) dois bytes;
c) três bytes;
d) quatro bytes.

8) A instrução capaz de deslocar o conteúdo da localização de memória, endereçada pelo par de registradores HL, uma posição para a direita, sendo o bit mais significativo (bit 7)

carregado com o seu próprio valor, é:

a) SRL reg;
b) SLA (HL);
c) SRA (HL);
d) N.R.A.

9) A instrução SUB data:

a) soma, ao conteúdo do acumulador, o conteúdo do byte de dados (data);
b) subtrai do conteúdo do acumulador o conteúdo do byte de dados (data);
c) desloca o conteúdo do acumulador um bit para a direita;
d) N.R.A.

10) O "complemento de dois" do número $7A_{16}$ é:

a) 36_{16};
b) $9B_{16}$;
c) FA_{16};
d) 86_{16}.