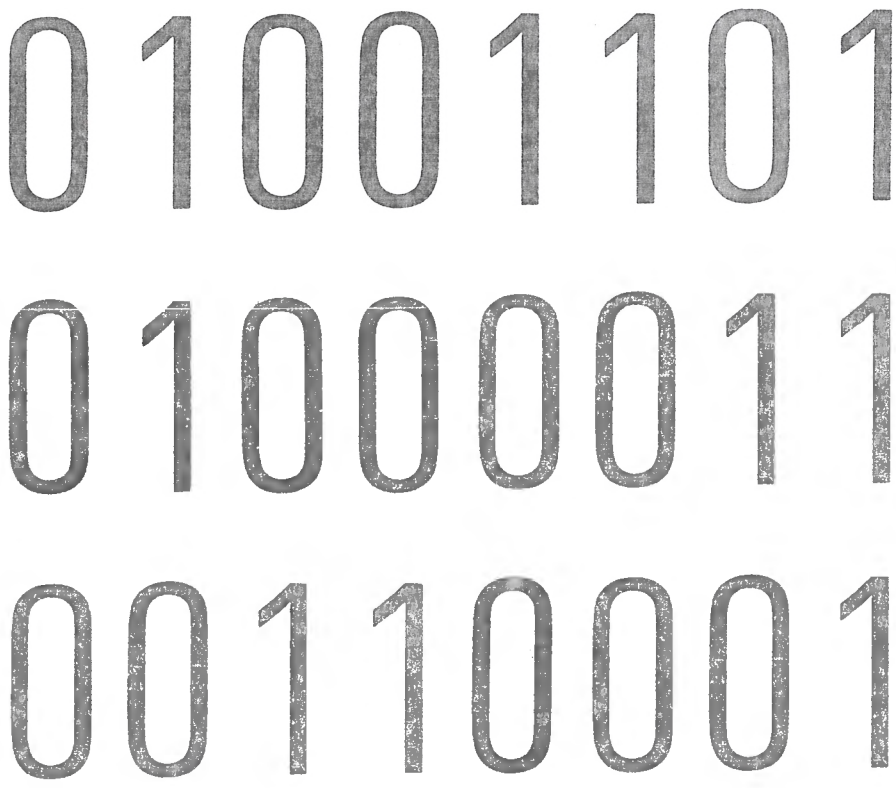
senai-sp

Informática Industrial

Circuitos Digitais e Microcomputadores para Controle de Processos

Módulo IV – MC-1

Microcomputadores - 1



Circuitos Digitais e Microcomputadores para Controle de Processos

Trabalho elaborado pela Divisão de Material Didático do
Departamento Regional do SENAI-SP.

Equipe responsável

Coordenação do projeto	Waldemar de Oliveira Junior
Coord. de planej. e produção do material	Arlette Fernandes Portella
Assessoria em tecnologia educacional	Adilson Tabain Kole Jānis Ivars Rītiņš (SENAI-DN)
Elaboração	Ricardo Figueiredo Terra
Revisão técnica	Waldemar de Oliveira Junior
Edição de texto	Marinilzes Moradillo Mello
Definição gráfica dos mapas de informação	Niriá Rangel Ribeiro
Composição	Niriá Rangel Ribeiro
Produção gráfica	José Luciano de Souza Filho Maisa Dal Prete Regina Bouzan
Coordenação da impressão	Victor Atamanov

Ficha catalográfica
Elaborada pela Unidade de Editoração — SENAI-SP

S47m SENAI-SP. Divisão de Material Didático.
Microcomputadores — 1; MC-1.
Por Ricardo Figueiredo Terra.
São Paulo, 1986. 437 p.
(Circuitos Digitais e Microcomputadores
para Controle de Processos, 2)

1. HARDWARE. 2. UNIDADE ARITMÉTICA
LÓGICA. 3. MICROPROCESSADOR.
4. MEMÓRIA (PD). 5. INTERFACE.
6. INSTRUÇÃO DE MICROPROCESSADOR.
I. TERRA, Ricardo Figueiredo. II. t. III. s.

681.3
(CDU, IBICT, 1976)

SENAI — Serviço Nacional de Aprendizagem Industrial
Departamento Regional de São Paulo
Av. Paulista, 750 — Bela Vista — SP
CEP 01310 — Fone: (011) 289 8022

SENAI — Instituição mantida e administrada pela Indústria

CÓDIGO HEXADECIMAL

TE 10035
(2 páginas)

Introdução — O código hexadecimal, como o nome já diz, utiliza a base 16.

Descrição — O código hexadecimal é composto por 16 símbolos, representados pelos símbolos de 0 a 9 e pelas letras A, B, C, D, E, F.

Comentário — Cada dígito usa 4 bits, e como pocemos ter 16 combinações (2^4) com 4 bits cada uma, dispomos assim de um aproveitamento total devido à utilização de símbolos e letras.

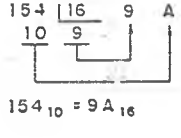
Exemplo

Hexadecimal	Decimal	Binário
00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
0A	10	1010
0B	11	1011
0C	12	1100
0D	13	1101
0E	14	1110
0F	15	1111

Exemplo:

$154_{10} = X_{16}$

portanto...



$154_{10} = 9A_{16}$

Palavras-chave

CONVERSÃO ENTRE NÚMEROS HEXADECIMAIS E BINÁRIOS TE 10036

(4 páginas)

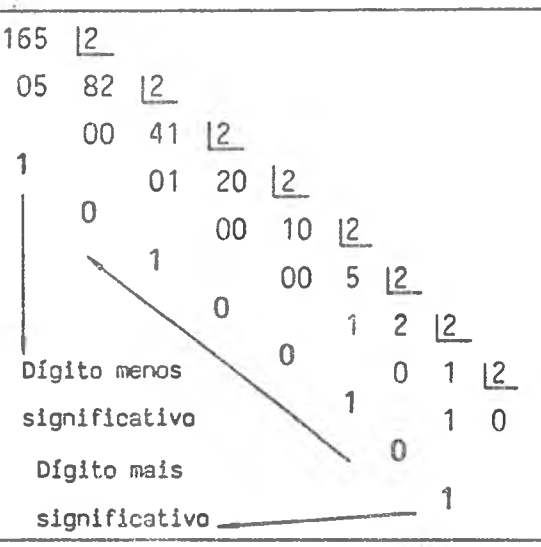
Descrição — Conversão entre números hexadecimais e binários é a passagem de um sistema de numeração para outro.

Procedimento

Se a conversão for...	... então utilize...
... hexadecimal → binário...	... procedimento 1 ou 1a.
... binário → hexadecimal...	... procedimento 2.

Procedimento 1

Passo	Exemplo
1º Escreva o número hexadecimal.	A5
2º Faça a conversão de hexadecimal para decimal.	$10 \times 16^1 + 5 \times 16^0 = 165$
3º Divida o número sucessivamente por dois até que o quociente seja 0, destacando os restos obtidos em cada divisão.	165 |2 05 82 |2 00 41 |2 01 20 |2 0 10 |2 0 5 |2 1 2 |2 0 1 |2 1 0 1 0 1

4º Leia os restos de baixo para cima como indica a seta.. O primeiro dígito lido é o mais significativo e o último, o menos significativo	
5º Escreva o número lido, sabendo que o dígito mais significativo deverá ficar à esquerda.	10100101 Dígito mais significativo

Procedimento 1a

Passo	Exemplo
1º Escreva o número em hexadecimal.	A5
2º Converta cada dígito hexadecimal em seu equivalente binário, através de uma tabela.	A 5 1010 0101
3º Reagrupe os dígitos binários resultantes em um só conjunto.	10100101

Procedimento 2

Passo	Exemplo
1º Escreva o número binário.	01011101
2º Agrupe os dígitos em grupos de 4 bits.	0101 1101
3º Procure o equivalente em uma tabela binário puro.	0101 = número 5 1101 = 13 em decimal ou D em hexadecimal
4º Agrupe os números hexadecimais já decodificados, através da tabela.	0101 1101 5D

Palavras-chave

CONVERSÃO ENTRE HEXADECIMAL E BINÁRIOS

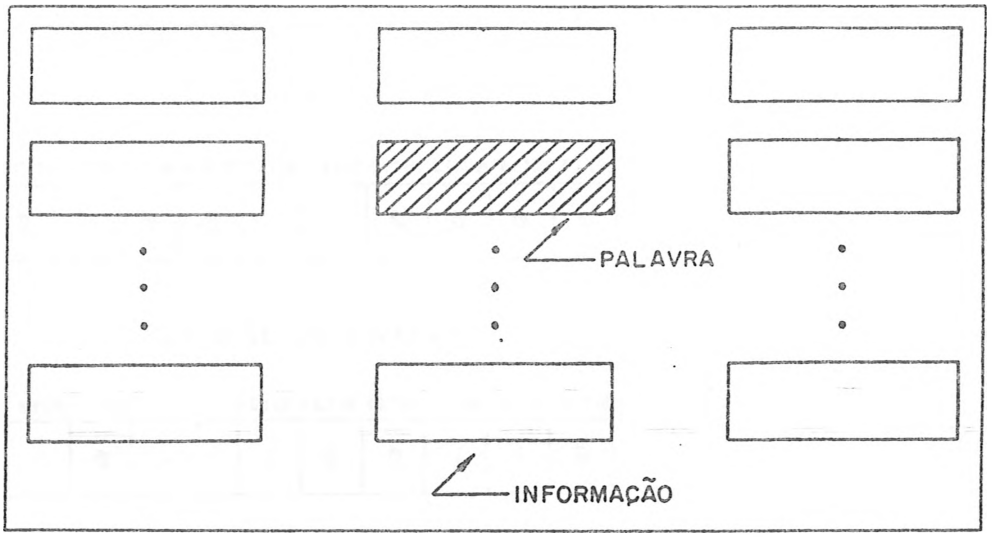
PALAVRA BINÁRIA
(Binary word, word)

TE 20120
(2 páginas)

Descrição

Palavra binária é a denominação dada a um agrupamento de bits que representa uma unidade padronizada de informação em um sistema de processamento digital.

Ilustração

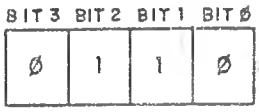


Classificação

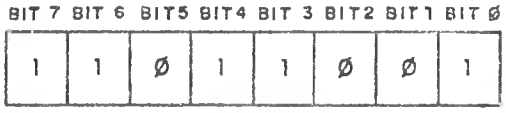
As palavras binárias podem ser classificadas quanto ao tamanho em:

Nº de bits	Tipos de processador
4	Intel 4004, atualmente em desuso
8	8080, 8085, Z-80, 6000, μ processadores de oito bits
16	8086, 68000, micro e minicomputadores de 16 bits
32	Micro e minicomputadores bit sliced
64/128	Computadores de grande porte (main frame)

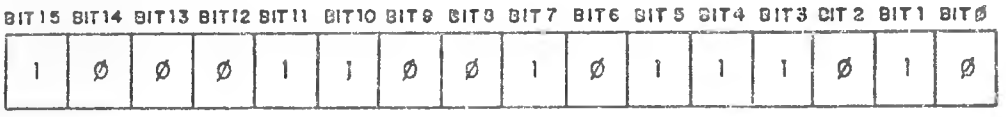
Palavra de 4 bits



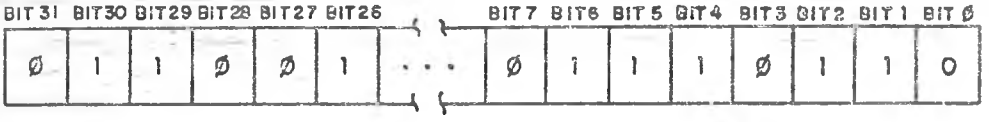
Palavra de 8 bits



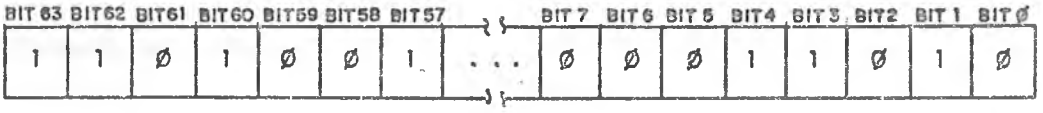
Palavra de 16 bits



Palavra de 32 bits



Palavra de 64 bits



Comentário

Em geral as palavras binárias têm o mesmo número de bits de uma locação de memória do processador em questão; entretanto, algumas vezes seu tamanho pode corresponder a mais de uma locação.

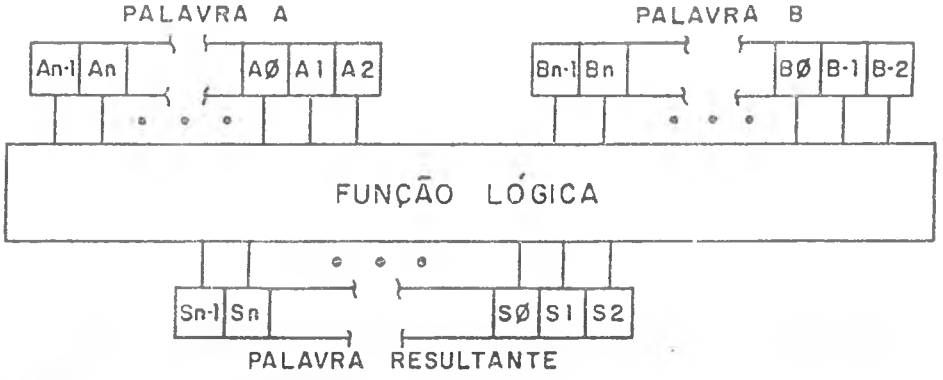
Palavras-chave

PALAVRA BINÁRIA

OPERAÇÕES LÓGICAS ENTRE PALAVRAS BINÁRIAS — TE 20121 (2 páginas)

Descrição — Uma operação lógica entre duas palavras b árias resulta em uma outra palavra com o mesmo número d bits das anteriores.

Ilustração

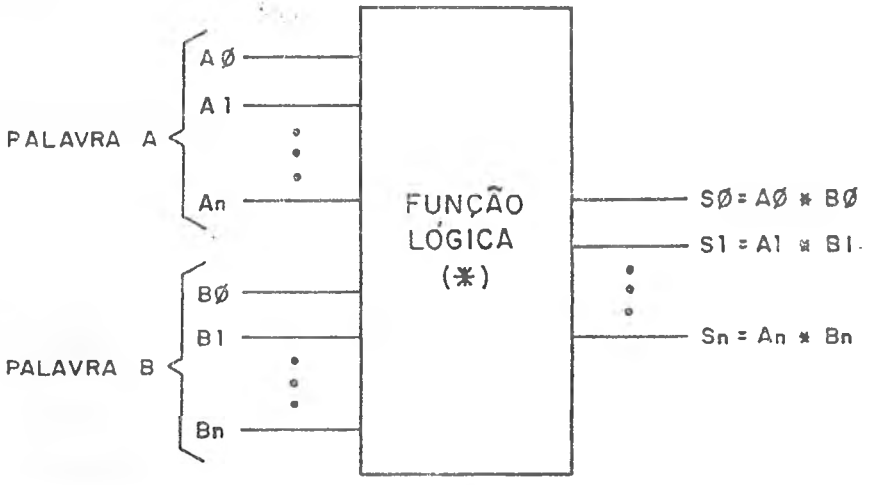


Classificação — As operações lógicas podem ser classificadas em:

Diretas	Inversas
E	NÃO E
OU	NÃO OU
OU EXCLUSIVO	COINCIDÊNCIA
SIM	NÃO

Processo — A realização de uma operação lógica entre duas palavras é feita através da combinação lógica entre os bits de mesma posição relativa (2^n, 2^{n-1}, ..., 2^2, 2^1, 2^0) das palavras operadas.

Ilustração



Comentário

As palavras binárias podem ser combinadas logicamente de forma paralela, ou seja, todos os bits simultaneamente, como ilustrado acima, ou de forma seqüencial onde cada par de bits é combinado em um instante de tempo diferente e os resultados parciais vão sendo armazenados e deslocados em registradores de deslocamento para, após um tempo determinado, apresentar o resultado da operação.

Aplicações

Em sistemas de processamento digital, as operações lógicas entre operandos, representados por palavras, em geral são realizadas para definir um possível desvio no fluxo de processamento, mascaramento de determinados bits de um dado, teste de bits, etc.

Palavras-chave

OCUPAÇÕES LÓGICAS ENTRE PALAVRAS BINÁRIAS

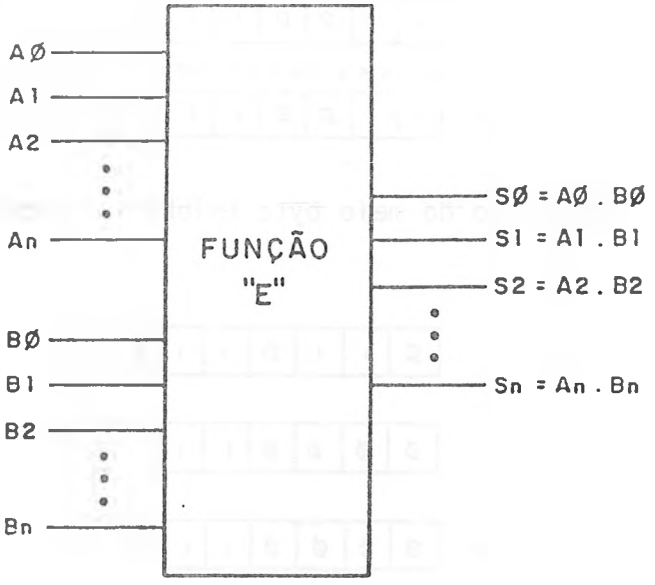
FUNÇÃO "E" ENTRE PALAVRAS BINÁRIAS — TE 20122

(AND entre palavras) — (2 páginas)

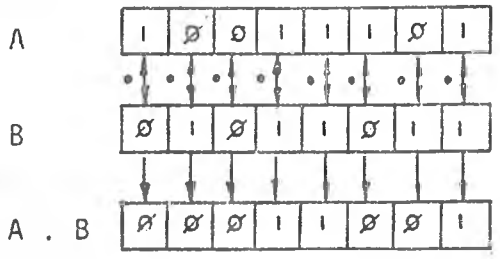
Descrição — A função "E" entre palavras binárias segue as mesmas regras utilizadas na realização do "E" entre dois bits.

Processo — A aplicação da função "E" entre duas palavras de n bits resulta em uma outra palavra onde cada bit é expresso pela combinação "E" entre os bits de mesma posição relativa das palavras operadas.

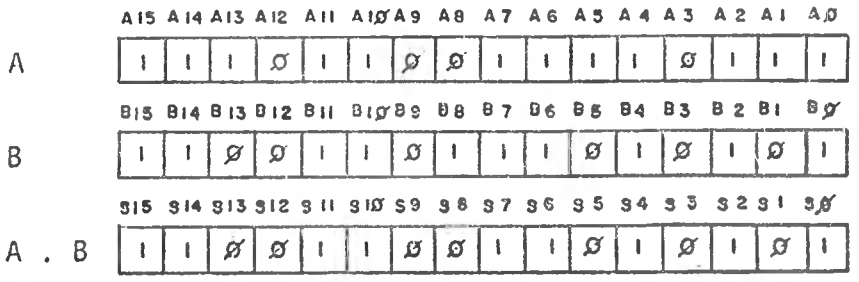
Ilustração



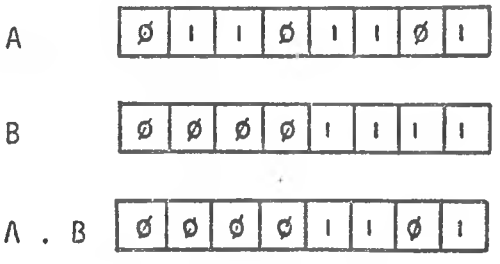
Aplicação "E" entre palavras de 8 bits



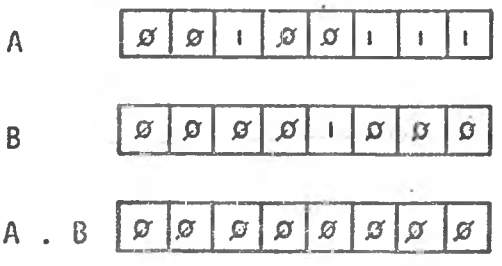
"E" entre palavras de 16 bits



Separação do meio byte (nibble) menos significativo da palavra



Teste de um bit



Palavras-chave

FUNÇÃO E ENTRE PALAVRAS

FUNÇÃO "OU" ENTRE PALAVRAS BINÁRIAS — TE 20123

(OR entre palavras) — (2 páginas)

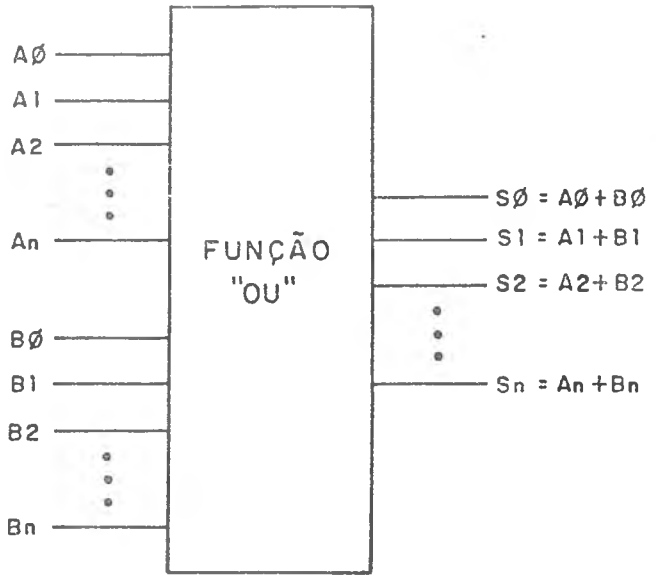
Descrição

A função "OU" entre palavras segue as mesmas regras utilizadas na realização do "OU" entre dois bits.

Processo

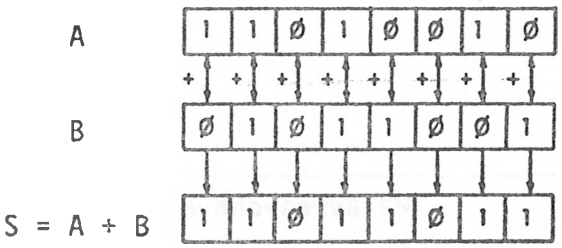
A realização da função "OU" entre duas palavras resulta em outra palavra onde cada bit é expresso pela combinação "OU" entre os bits de mesma posição relativa das palavras operadas.

Ilustração



Aplicação

"OU" entre palavras de 8 bits



TE 20123

"OU" entre palavras de 16 bits

A	1	Ø	Ø	1	Ø	1	1	Ø	Ø	Ø	1	1	Ø	1	1	Ø
B	1	Ø	1	Ø	1	Ø	Ø	1	Ø	1	1	Ø	Ø	Ø	1	Ø
S = A + B	1	Ø	1	1	1	1	1	1	Ø	1	1	1	Ø	1	1	Ø

Palavras-chave

FUNÇÃO OU ENTRE PALAVRAS

FUNÇÃO "OU EXCLUSIVO" ENTRE PALAVRAS BINÁRIAS (EXCLUSIVE OR entre palavras)

TE 20124
(2 páginas)

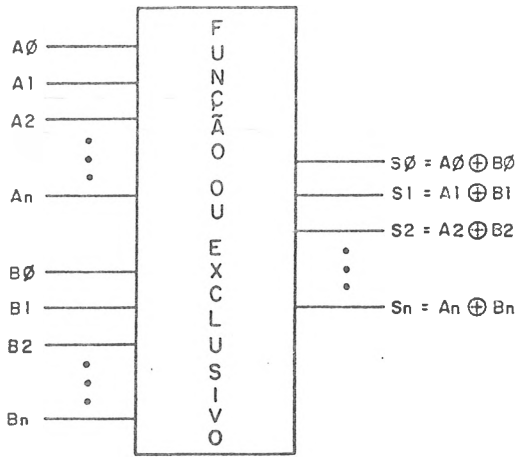
Descrição

A função "OU EXCLUSIVO" entre palavras obedece às mesmas regras utilizadas na realização do "OU EXCLUSIVO" entre dois bits.

Processo

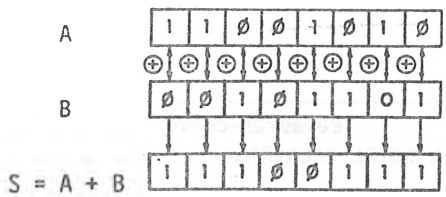
A realização da função "OU EXCLUSIVO" entre duas palavras resulta em outra palavra onde cada bit é expresso pela combinação "OU EXCLUSIVO" entre os bits de mesma posição relativa das palavras operadas.

Ilustração



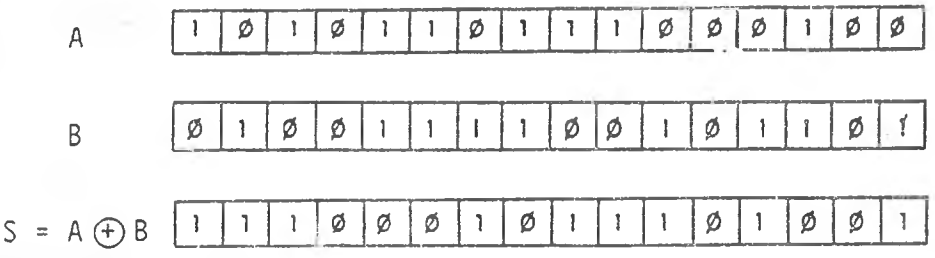
Aplicação

"OU EXCLUSIVO" entre palavras de 8 bits



TE 20124

"OU EXCLUSIVO" entre palavras de 16 bits



Palavras-chave

FUNÇÃO OU EXCLUSIVO ENTRE PALAVRAS

FUNÇÃO "NÃO" APLICADA A UMA PALAVRA BINÁRIA

TE 20125
(2 páginas)

Sinônimo

Negação
Complemento
Inverso

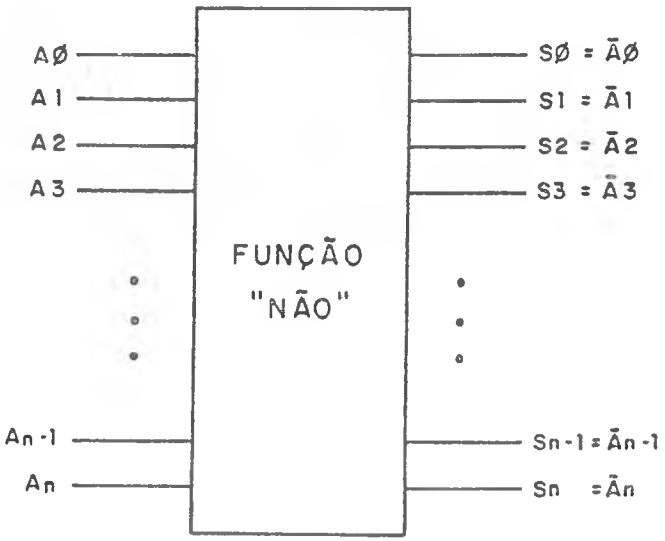
Descrição

A função "NÃO" aplicada a uma palavra segue as mesmas regras utilizadas na realização do "NÃO" de um bit.

Processo

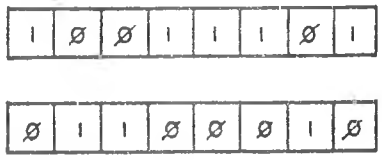
A função "NÃO" aplicada a uma palavra de n bits resulta em outra palavra onde cada bit é expresso pelo complemento do bit de mesma posição relativa na palavra original.

Ilustração

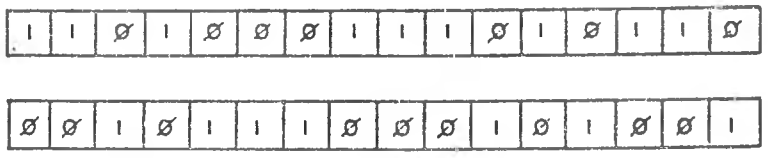


Aplicação

Complemento de uma palavra de 8 bits



Complemento de uma palavra de 16 bits



Comentário

As funções "NÃO E", "NÃO OU" e "COINCIDÊNCIA" são a negação respectivamente das funções "E", "OU" e "OU EXCLUSIVO", não sendo em geral disponíveis como operações diretas nos microprocessadores.
Quando há necessidade de realizar alguma delas, normalmente isto é feito em dois passos: primeiro realiza-se as funções diretas "E", "OU", "OU EXCLUSIVO" e em seguida faz-se o complemento deste resultado parcial, obtendo-se a função desejada.

Palavras-chave

FUNÇÃO NÃO ENTRE PALAVRAS

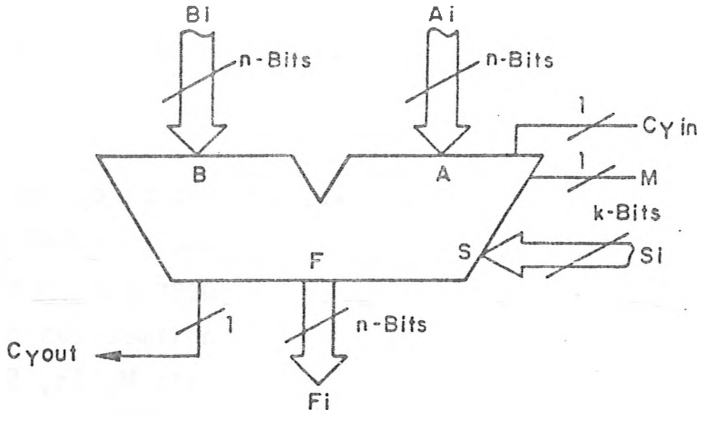
UNIDADE LÓGICA ARITMÉTICA - ULA (Arithmetic Logic Unit - ALU)

TE 30420 (5 páginas)

Descrição

Uma unidade lógica artimética (ULA) é um dispositivo combinacional que aceita duas palavras de n bits como entrada e efetua operações lógicas ou aritméticas entre elas, segundo a função determinada por bits seletores, gerando uma palavra de n bits de saída.

Ilustração



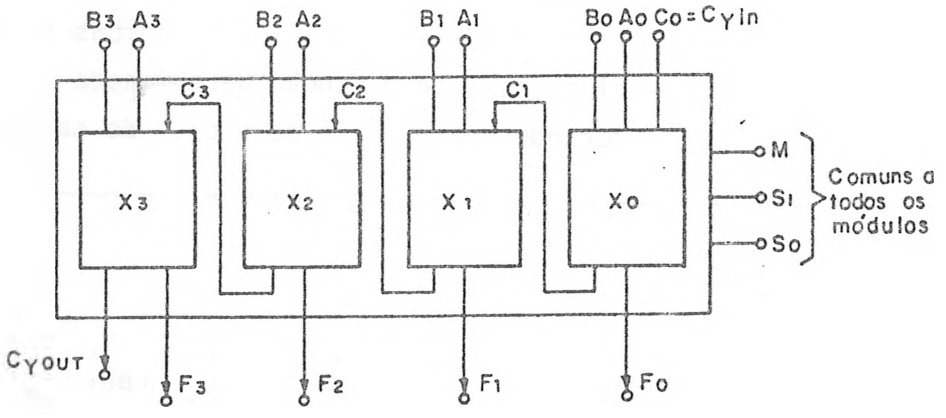
Ai: palavra binária de n bits de entrada.
Bi: palavra binária de n bits de entrada.
M: bit de seleção que define se a operação a ser realizada é lógica ou aritmética.
Si: palavra binária de k bits de entrada que define a função a ser realizada.
Cy_{in}: entrada de um bit vem um (carry in).
Cy_{out}: saída de um bit vai um (carry out).
Fi: palavra binária de n bits de saída.

Comentário

As palavras binárias de entrada e de saída podem possuir qualquer quantidade de bits, porém, na maior parte dos circuitos integrados, a ULA efetua operações lógicas ou aritméticas entre palavras de quatro bits.

Estrutura

A ULA pode ser construída a partir de uma cascata de módulos idênticos conforme figura abaixo.



xo, x1, x2, x3: são circuitos combinacionais que executam bit por bit as operações lógicas ou aritméticas de acordo com os bits M, S1, So (bits de controle).

Processo

Será descrito o processo de funcionamento genérico para as ULAs.

Bit M

O bit M determina se a ULA executará operações lógicas ou aritméticas, de acordo com os seus níveis lógicos.

Sinais de seleção de função

Os sinais de S_0 a S_1 determinam o tipo de função a ser implementada às duas palavras A e B de entrada, isto é, para cada combinação possível temos uma função lógica ou aritmética (dependendo se M = 0 ou M = 1) correspondente.

Comentário

Nos microprocessadores as instruções são quem determinaram que tipo de função a ALU interna ao mesmos implementará.

Sinal de carry in

O sinal C_{yin} é o sinal de vem um utilizado para ligação de várias ULAs em cascata, com significado apenas quando a ULA está selecionada para efetuar operações aritméticas.

Sinal de carry out

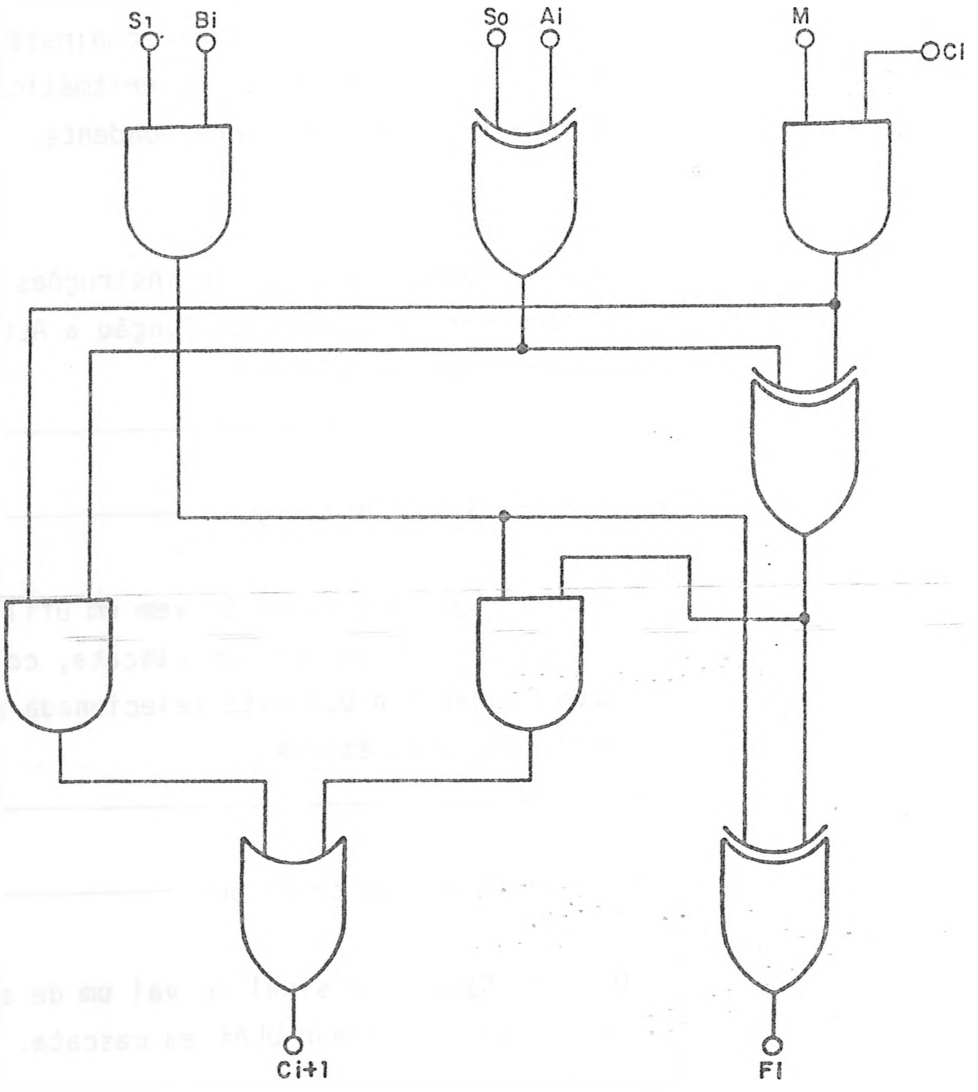
O sinal C_{yout} é o sinal de vai um de saída, utilizado para ligar ULAs em cascata.

Palavra de saída

Os bits de F_0 a F_i correspondem à palavra binária de saída, resultado da operação efetuada entre as entradas A e B.

Exemplo

Uma estrutura de portas possível para configurar um dos estágios de uma ULA é apresentada na figura abaixo.



As funções geradas pela estrutura lógica apresentada na figura anterior é a seguinte:

M = 0 → Função lógica

S_1	So	Fi	Comentário
0	0	Ai	Entrada A = saída F
0	1	$\bar{A}i$	Complemento de A = saída F
1	0	Ai ⊕ Bi	"OU EXCLUSIVO"
1	1	$\overline{Ai \oplus Bi}$	"EQUIVALÊNCIA"

M = 1 C_{in} = 0 → Função aritmética			
S_1	S_0	Fi	Comentário
0	0	A	Entrada A = saída A
0	1	A	Complemento de 1 de A
1	0	A mais B	Soma de A e B
1	1	A mais B	Soma de B com o complemento de 1 de A

M = 1 C_{in} = 1 → Função aritmética			
S_1	S_0	Fi	Comentário
0	0	A mais 1	Incrementa A
0	1	A mais 1	Complemento de 2 de A
1	0	A mais B mais 1	Incrementa soma de A e B
1	1	A mais B mais 1	B menos A

Aplicações

As ULAs encontram aplicações principais na configuração de arquiteturas de unidades centrais de processamento nos computadores.

Palavras-chave

UNIDADE LÓGICA ARITMÉTICA - ULA

ALU

CONCEITO DE PROGRAMA ARMAZENADO

TE 70010
(1 página)

Sinônimo — Conceito de Von Neumann

Introdução — Para superar as limitações existentes nos primeiros computadores surgidos, que exigiam demoradas e cansativas tarefas de programação por hardware, Von Neumann introduziu o conceito de programa armazenado.

Descrição — O conceito de programa armazenado introduz a idéia de organizar seqüencialmente as operações a serem executadas (instruções) e dados a serem manipulados (operandos) em uma memória. Essa memória é acessada, também de forma seqüencial, para fornecer à unidade processadora (executante) as instruções e dados necessários à execução da seqüência de operações (programa) desejada.

Palavras-chave

CONCEITO DE VON NEUMANN

ENDEREÇOS DE RAM E DE ROM EM MICROCOMPUTADORES TE 60118

(2 páginas)

Apresentação

Em sistemas de microcomputador, as memórias RAM e ROM são dispostas em módulos, com capacidade de armazenamento conforme os circuitos integrados, utilizados na implementação destas no sistema.

Ao todo, os módulos podem ser visualizados como uma única memória RAM e uma única memória ROM, com cada uma das posições acessável através de uma única palavra de endereço.

Comentário

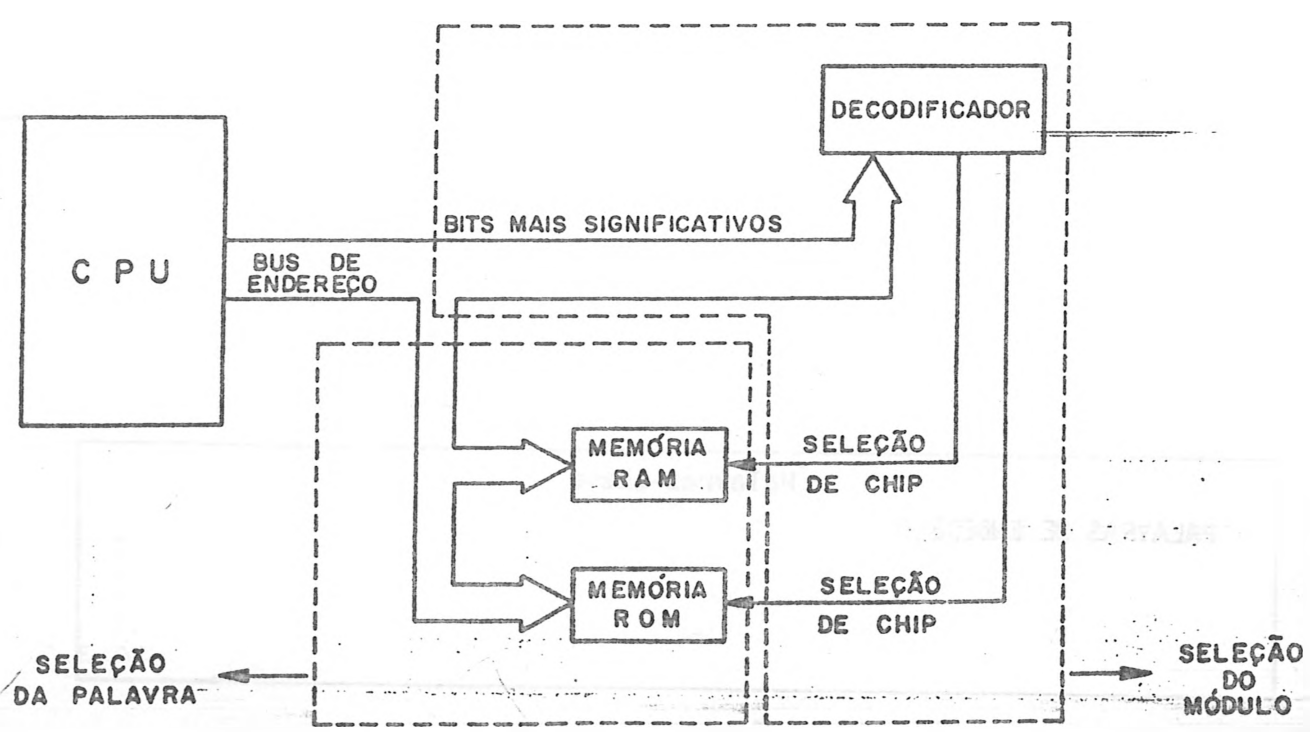
A palavra de endereços em um sistema de microcomputador terá tantos bits quantos forem necessários para endereçar a capacidade de memória implementada no sistema

---

Estrutura

A palavra de endereços em um sistema de microcomputador é constituída de bits de chip select ou seleção de módulos e bits de endereço de palavras.

Exemplo



Comentário

. O número de bits de seleção da palavra varia de acordo com a capacidade de endereçamento de cada módulo.

. O número de seleção de módulo varia de acordo com o número de módulos implementados no sistema.

Procedimento — Os bits de seleção de módulos são decodificados por circuitos combinacionais que geram o sinal de habilitação do módulo selecionado.

Dessa forma, acessa-se qualquer uma das posições de memória do módulo através dos bits de seleção de palavras.

Palavras-chave

PALAVRAS DE ENDEREÇO

MICROCOMPUTADOR TE 60010

3 páginas)

Introdução

O microcomputador faz parte da família de arquiteturas de processamento de dados, que utiliza o conceito de programa armazenado.

Descrição

O microcomputador é um sistema de processamento de dados de baixo custo, média capacidade de processamento, capacidade de memória de acesso direto mediana e grande versatilidade em termos de possibilidades de aplicação.

Classificação

Quanto ao número de bits da palavra de dados eles podem ser de:

. 8 bits
. 16 bits
. 32 bits

Quanto à capacidade de endereçamento direto, a memória pode ser de:

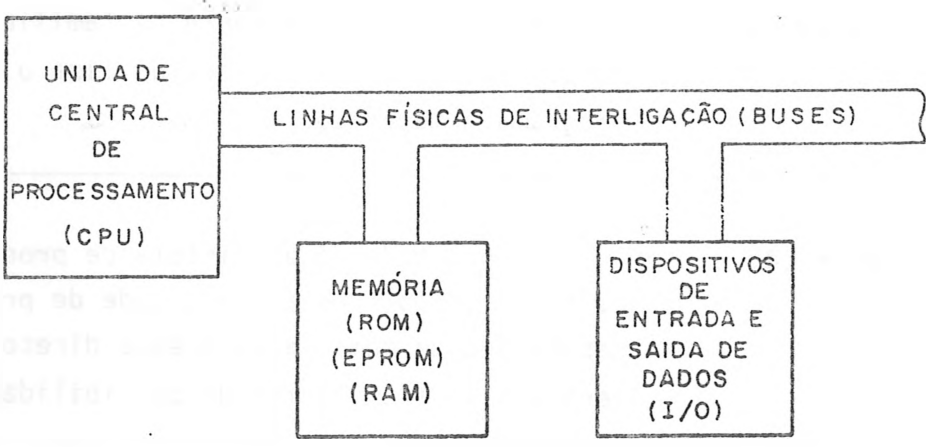
. até 64K bytes para 8 bits
. até 16M bytes para 16 bits

Comentário

Em face ao desenvolvimento acelerado da tecnologia nesta área, alguns dos ítens anteriores podem tornar-se desatualizados em curto espaço de tempo.

Estrutura

O microcomputador possui a seguinte estrutura geral:



Processo

O funcionamento do microcomputador em linhas gerais é o seguinte:

. O programa (seqüência de instruções) é disposto seqüencialmente na memória.
. Os dados a serem manipulados também podem estar armazenados na memória, ou são obtidos através de dispositivos de entrada.
. A cada passo, a unidade central de processamento lê uma instrução, na memória, interpreta-a e executa-a.
. O processo segue, instrução por instrução, varrendo todo o programa.
. O resultado do processamento, em geral, é enviado a um dispositivo de saída para algum tipo de atuação externa ou visualização.

Aplicação

Em plantas industriais:

. em sistemas de automação de pequeno e médio porte;
. em sistemas digitais de controle distribuído (SDCD);
. em sistemas de supervisão e aquisição de dados;
. em casamento de protocolo de comunicação entre computadores;
. em sistemas de distribuição de informação;
. em sistemas de apoio operacional e gerencial;
. na ampliação da capacidade de sistemas implantados com recursos próprios esgotados;
. etc.

Comentário

As aplicações não industriais dos microcomputadores não fazem parte do escopo do presente texto.

Palavras-chave

MICROCOMPUTADORES

UNIDADE CENTRAL DE PROCESSAMENTO EM MICROCOMPUTADORES — TE 60020
(Central processing unit) — (2 páginas)

Sinônimo CPU
UCP

Descrição A CPU é o conjunto de circuitos responsável pela busca, interpretação e execução das instruções. É ainda responsável pela geração dos sinais para o controle da memória e dos dispositivos de entrada e saída, necessários ao desempenho de suas próprias funções.

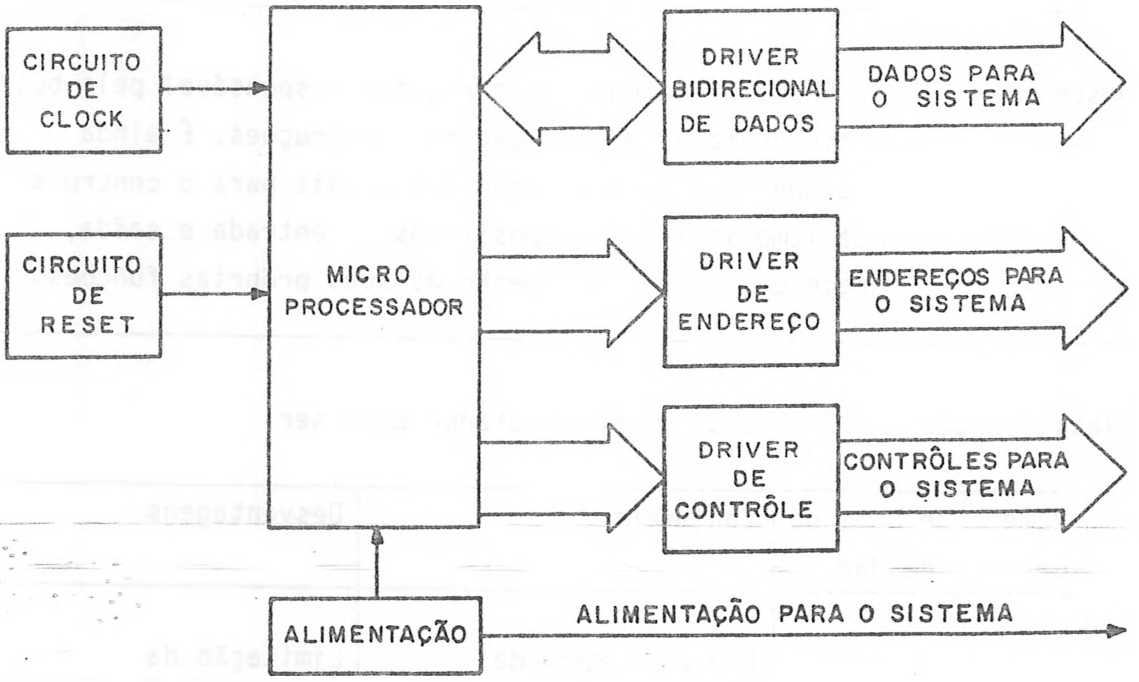
Classificação A CPU de um microcomputador pode ser:

Número de bits da palavra de dados	Vantagens	Desvantagens
8	Grande número de dispositivos periféricos disponíveis.	Limitação da capacidade de processamento.
8/16	Trabalha com dispositivos externos de 8 bits e palavras internas de 8 ou 16 bits.	Perde em processamento para os micros específicos de 16 bits.
16	Boa capacidade de representação numérica e processamento.	Exige circuitos periféricos externos especiais para palavra de 16 bits.

Comentário
Existem microcomputadores de 32 bits ainda não disponíveis comercialmente no mercado nacional.

Estrutura

A estrutura da CPU dos microcomputadores, em geral, é baseada em microprocessador e pode assumir diversas configurações, sendo que a básica deve ser:



Palavras-chave

CPU
UCP

MEMÓRIAS
(Memories)

TE 60040
(2 páginas)

Apresentação

Todo sistema que se utiliza de microprocessador deve possuir uma ou mais memórias pois é na memória que encontramos a seqüência de instruções que o microprocessador irá executar, além de variáveis que estão sendo tratadas pelo sistema de microcomputador.

Descrição

Memórias são componentes pertencentes à classe dos LSI e dos VLSI (very large scale integration), encapsulados em plástico ou cerâmica, que armazenam informações temporária ou definitivamente dependendo do tipo a que pertençam.

Classificação

Podemos classificar as memórias em dois grandes grupos.

. memórias de massa
. memórias semicondutoras

Memórias de massa

Descrição
São as que armazenam informações em meios físicos.

Exemplo
Discos magnéticos, fitas magnéticas etc.

Memórias semicondutoras

Descrição
São as que armazenam informações em circuitos integrados.

Exemplo
Memórias ROM, RAM, EPROM, etc.

Palavras-chave

MEMÓRIAS

MEMÓRIA ROM
(Read only memory)

TE 60100
(4 páginas)

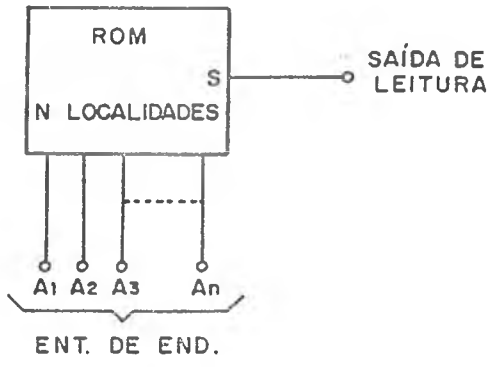
Sinônimo — Memória apenas de leitura
MAL

Descrição — A memória ROM é um circuito estritamente combinacional que apresenta como característica principal a de permitir somente a leitura dos dados nela gravados.

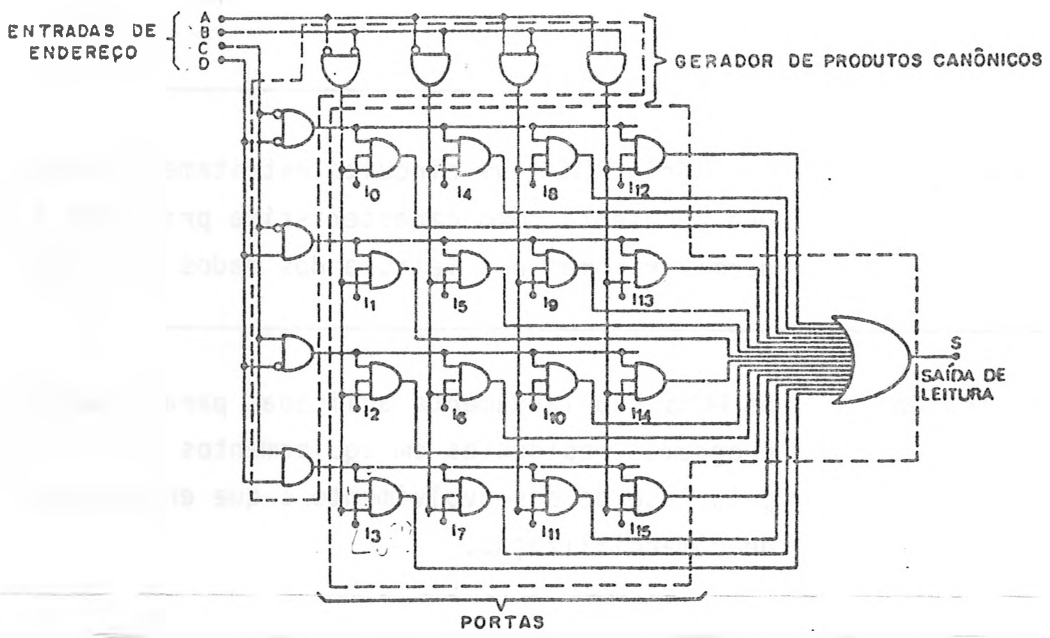
Aplicação — As ROMs são largamente aplicadas para armazenar programas residentes em equipamentos que já passaram pela fase de desenvolvimento e que entram para a fase de comercialização.

Comentário
É inviável que o usuário encomende ao fabricante uma pequena quantidade de ROMs, pois o desenvolvimento de máscaras para sua gravação apresenta um custo relativamente elevado.

Ilustração



Estrutura : A estrutura básica de uma ROM é ilustrada a seguir.



Parte	Função
Entradas de endereço (A, B, C e D)	Meio físico pelo qual a informação de endereço chega ao gerador de produtos canônicos.
Saída de leitura	Meio físico pelo qual a informação contida no endereço selecionado é retirada.
Gerador de produtos canônicos	Decodificador de endereço.
Portas	Circuito combinacional que quando liberado leva para a saída o dado contido no endereço acessado.

Comentário

Os dados contidos nas entradas I_0, I_1, I_2... I_{n-1} são gravados pelos fabricantes do circuito integrado e não podem ser alterados.

Processo

Para a descrição do funcionamento de leitura das memórias ROM será utilizada como exemplo uma ROM com 16 localidades (N = 16 ∴ n = 4)

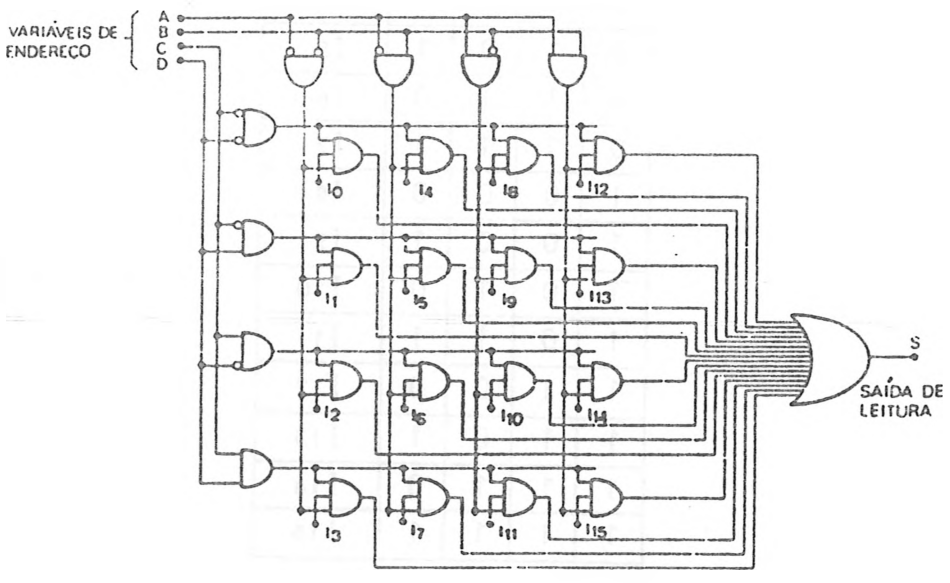


Tabela-verdade

Variáveis de endereço				S
A	B	C	D	
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3
0	1	0	0	I_4
0	1	0	1	I_5
0	1	1	0	I_6
0	1	1	1	I_7
1	0	0	0	I_8
1	0	0	1	I_9
1	0	1	0	I_{10}
1	0	1	1	I_{11}
1	1	0	0	I_{12}
1	1	0	1	I_{13}
1	1	1	0	I_{14}
1	1	1	1	I_{15}

Palavras-chave

MEMÓRIAS ROM

MEMÓRIAS PARA ARMAZENAMENTO DE DADOS BINÁRIOS TE 60101

(3 páginas)

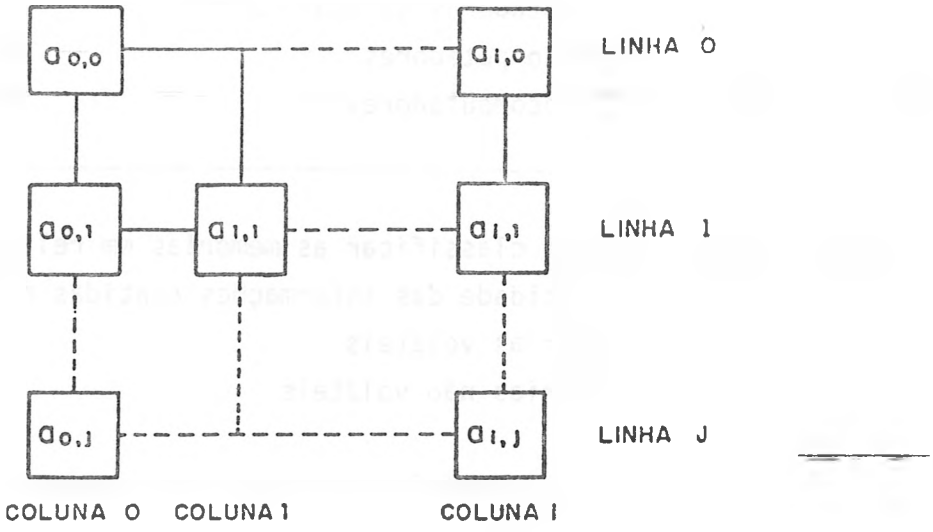
Descrição

As memórias para armazenamento de dados binários são dispositivos que permitem armazenar dados na forma binária (bit) de modo permanente ou temporário, para serem posteriormente manipulados.

Estrutura

As memórias para armazenamento de dados binários consistem normalmente em uma rede de elementos biestáveis, que armazenam bits, endereçáveis individualmente.

Ilustração



Comentário

. Na ilustração acima temos um arranjo de elementos biestáveis em uma rede de acesso matricial.

. Cada elemento biestável é responsável pelo armazenamento de um bit de informação ($a_{i,j}$).

. O índice i,j é o endereço da posição correspondente a este bit, sendo j o número da linha (horizontal) e i o número da coluna (vertical).

Comentário

Todo dispositivo de memória deve possuir duas propriedades necessárias:

. Toda posição que armazena um dígito binário deve ter um único endereço.

. Deve ser possível a leitura do estado de cada dígito armazenado.

Aplicação

As memórias são utilizadas principalmente em dispositivos que se utilizam do conceito de programa armazenado, tais como:

. Computadores de grande porte

. Computadores de médio porte

. Minicomputadores

. Microcomputadores

Classificação

Podemos classificar as memórias em relação à volaticidade das informações contidas nas mesmas em:

. Memórias voláteis

. Memórias não voláteis

Memórias voláteis

Descrição

Memórias voláteis são aquelas que perdem toda a informação quando se desliga a alimentação elétrica.

Memórias não voláteis

Descrição

Memórias não voláteis são aquelas que mesmo sem a alimentação elétrica não perdem a informação.

Podemos, também, classificar as memórias conforme suas características de acesso de escrita e/ou leitura de dados em:

. Memórias de acesso randônico ou aleatório (RAM)

. Memórias de apenas leitura (ROM)

Memórias de acesso randônico ou aleatório (RAM)

Descrição

São memórias semicondutoras que podem ser lidas ou escritas e são utilizadas para armazenar informações que devem ser modificadas no decorrer do tempo.

Memórias de apenas leitura (ROM)

Descrição

São memórias semicondutoras nas quais as informações podem ser apenas lidas.

Palavras-chave

MEMÓRIA

MEMÓRIAS SEMICONDUTORAS

TE 60102
(1 página)

Apresentação — Em microcomputadores, principalmente, os dispositivos de memória utilizados para armazenamento de dados são as memórias do tipo semicondutora.

Descrição — Memórias semicondutoras são componentes eletrônicos de integração de alta escala que guardam determinada quantidade de informação na forma binária, temporária ou permanente.

Classificação — Podemos classificar as memórias semicondutoras em dois grandes grupos:

. Memórias de apenas leitura (ROM)
. Memórias de leitura e escrita (RAM)

Aplicação

. Armazenamento de informações em sistemas digitais
. Sistemas de microcomputador

Palavras-chave

MEMÓRIAS SEMICONDUTORAS

MEMÓRIA ROM ORGANIZADA NA FORMA N x m

TE 60110
(2 páginas)

Sinônimo — Memórias apenas de leitura organizadas na forma N x m

---

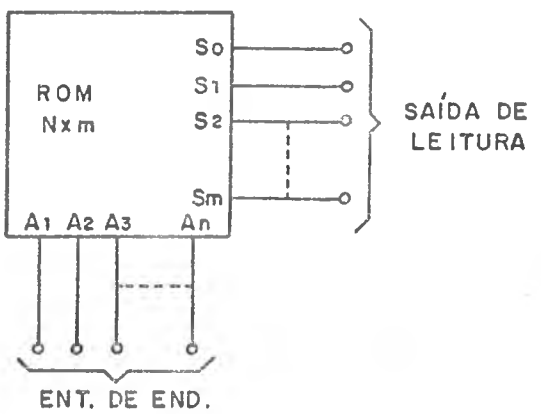
Descrição — As memórias apenas de leitura organizadas na forma N x m são um conjunto de memórias com endereços associados em paralelo de modo que, a partir de um mesmo endereço, pode-se ler uma palavra de mais de um bit.

Comentário
N é o número de endereços de memória e m o número de bits de informação gravados no mesmo endereço.

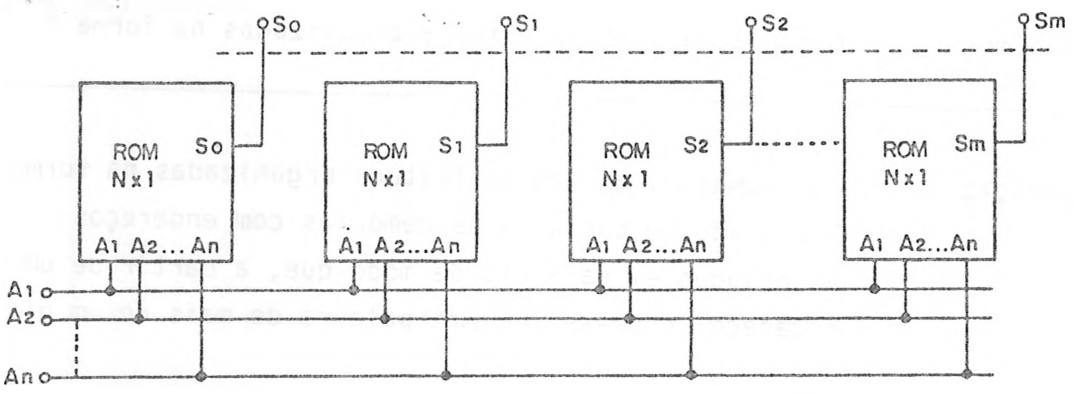
Aplicações — Essas memórias são aplicadas em:

. geradores de seqüência;
. tabelas de consultas;
. armazenamento de programas residentes de máquinas digitais;
. geradores de caracteres.

ILustração



Estrutura



Processo

O processo de funcionamento das memórias apenas de leitura organizadas na forma N x m é idêntico ao da memória apenas de leitura de um bit, variando apenas o número de saídas.

A1	A2	A3	...	Am	So	S1	S2 ...	Sm
0	0	0		0	I00	I01	I02 ...	I0m
				1	I10	I11	I12 ...	I1m
				0	I20	I21	I22 ...	I2m
				1	I30	I31	I32 ...	I3m
				⋮	⋮	⋮	⋮	⋮

Palavras-chave

MEMÓRIAS ROM

MEMÓRIA PROM (Programmable read-only memory)

TE 60120
(4 páginas)

Sinônimo

Memória apenas de leitura programável
MALP

Descrição

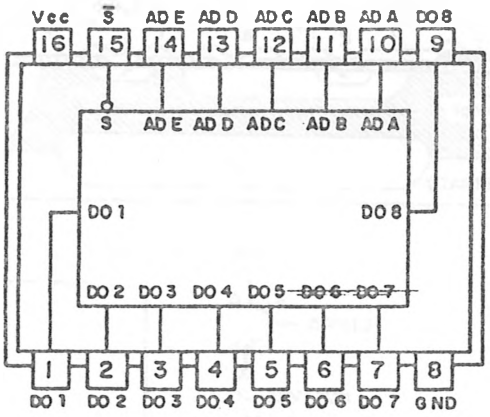
As PROMs são memórias programáveis pelo usuário que têm como característica o fato de o programa após ser armazenado não poder ser mais alterado.

Aplicação

As PROMs são aplicadas para armazenar programas residentes já perfeitamente definidos e para os quais o uso de ROMs é inviável.

Diagrama

Como exemplo é ilustrado abaixo o programa de uma ROM construída com tecnologia TTL (SN 74188).



Parte	Função
DO1 a DO8	Saídas de dados
ADA a ADE	Entradas de endereços
$\overline{S}$	Habilitador de memória

Classificação Quanto ao tipo de estrutura interna para programação as PROMs dividem-se em:

. rompimento de elo fusível

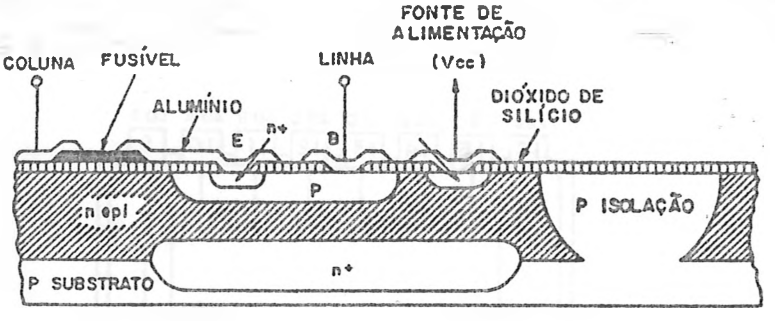
. destruição de diodo

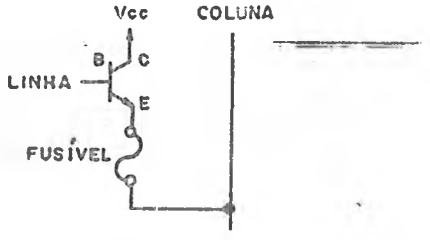
Rompimento de elo fusível

Descrição

Rompimento de elo fusível é o processo pelo qual a corrente de saturação do transistor bipolar abre o fusível conectado em seu emissor, resultando numa programação permanente na célula de memória com um estado binário aberto (nível 0).

Estrutura





Destruição de diodo

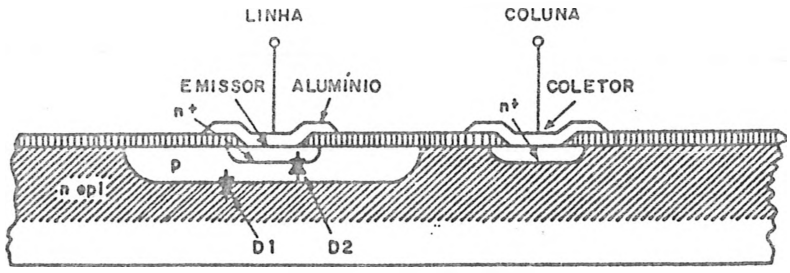
Descrição

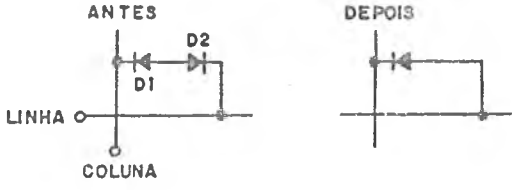
Destruição de diodo é o processo pelo qual um dos diodos montados em série oposta é destruído (colocado em curto), liberando o outro para a condução (nível 1).

Comentário

Na condição inicial, ou seja, com os diodos em perfeito estado não circulará corrente (nível 0).

Estrutura





Procedimento — Será descrito o procedimento para gravação de PROMs.

1º passo: Com a memória desabilitada, injetar nas linhas de endereço a posição da memória que se quer programar.

2º passo: Aplicar um pulso de grande amplitude às saídas correspondentes aos bits que se deseja levar ao estado contrário ao inicial.

Comentário

Inicialmente todos os bits da memória estão em nível 1 ou nível 0.

Palavras-chave

MEMÓRIAS PROM

MEMÓRIA EPROM — TE 60121

(Erasable programmable read only memory) — (4 páginas)

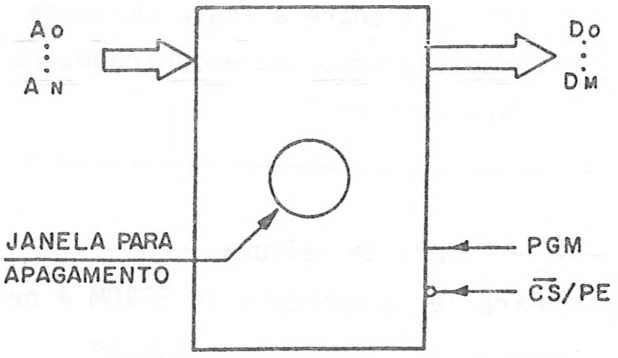
Descrição

As memórias EPROM são dispositivos de memórias semicondutoras não voláteis, de acesso aleatório, ou seja, nelas pode-se ler qualquer dado armazenado no mesmo tempo, independentemente da sua posição dentro da estrutura da memória.

São dispositivos da família de memórias ROM (memórias de apenas leitura), com a vantagem de poderem ser, através da exposição de raios ultravioletas, apagadas e depois reprogramadas pelo usuário.

Estrutura

Em termos gerais, uma EPROM tem configuração conforme a figura abaixo.



Parte	Função
A0... AN	Bits de endereçamento.
D0... DM	Bits de dados.
PGM	Comando de programação.
$\overline{CS}$/PE	Sinal de seleção da pastilha ou habilitação do modo de programação.
Janela de apagamento	Permite expor a pastilha do circuito integrado a uma fonte de raios ultravioletas para seu apagamento.

Comentário

O número de pinos do circuito integrado varia de acordo com a capacidade de armazenamento.

Processo

Uma EPROM tem três modos de operação:

. Modo de apagamento
. Modo de leitura
. Modo de programação

Modo de apagamento

Descrição

Para o apagamento de uma EPROM é necessário expor a janela transparente a uma fonte de raios ultravioletas.

Comentário

Os parâmetros tempo de exposição, comprimento de onda, intensidade da fonte emissora e distância entre a fonte emissora e o encapsulamento são encontrados no manual do fabricante de CI.

Modo de leitura

Para ler o conteúdo da EPROM é necessário executar os seguintes passos:

1º passo

Endereçar, através dos bits de A_0 a A_N, a posição de memória na qual se deseja efetuar a leitura.

2º passo

Habilitar a saída de dados em de D_0 a D_M colocando em nível lógico 0 no terminal $\overline{CS}$.

Comentário

Quando $\overline{CS}$ está em nível lógico 1 a saída de dados de D_0 a D_M fica em tri-state.

Modo de programação

Para programar uma EPROM é necessário executar os seguintes passos:

1º passo
Verificar se ela está apagada. Se não estiver, apagá-la.

2º passo
Colocar um nível de tensão especificada pelo fabricante no terminal PE, a fim de habilitar o modo de programação. Desta forma os bits de Do a D_M passam a ser entradas de dados.

3º passo
Colocar o dado a ser gravado nos bits de Do a D_M.

4º passo
Endereçar através dos bits de Ao a An a posição de memória em que se deseja armazenar o dado.

5º passo
Aplicar um pulso de tensão de largura e amplitude determinados pelo fabricante no pino PGM.

Comentário
Normalmente esta operação é feita seqüencialmente, da primeira até a última posição da memória. Existem equipamentos especialmente dedicados à gravação de EPROMs.

TE 60121

Aplicação — Em dispositivos digitais tais como:

- . Armazenamento de dados seqüenciais
- . Tabelas de conversão
- . Armazenamento de programas monitores

Palavras-chave

EPROM

MEMÓRIA DE APENAS LEITURA PROGRAMÁVEL E ALTERÁVEL

MEMÓRIA EAROM — TE 60122

(Electrically alterable read only memory) — (1 página)

Descrição

A memória tipo EAROM é um dispositivo semicondutor de armazenamento de dados binários da família de memórias ROM (memórias apenas de leitura). Além das características intrínsecas à família ROM, as memórias EAROM podem, também, ter seu conteúdo alterado eletricamente, quando observadas determinadas condições na sua operação. São memórias do tipo não volátil e de acesso aleatório.

Aplicação

A memória EAROM é usada em sistemas de microcomputadores de coleta de dados, onde não é possível utilizar memórias voláteis, e a informação a ser armazenada pode ser alterada.

Comentário

O problema das EAROMs é que eletronicamente elas são de difícil manipulação, devido à necessidade de circuitos especiais para seu apagamento. Também a informação armazenada vai se deteriorando ao longo do tempo, o que não ocorre com as demais memórias de sua família.

Palavras-chave

EAROM
EEPROM
E²PROM

GRAVADORES DE EPROM

TE 60123
(3 páginas)

Descrição — Gravadores de EPROM são dispositivos dedicados à programação de memórias EPROM.

Classificação — Podemos classificar os gravadores de EPROM segundo sua forma de implementação:

. passo-a-passo
. utilizando um sistema de microcomputador

Gravador de EPROM passo-a-passo

Descrição

É um processo de gravação de memória EPROM onde os dados são inseridos manualmente, sendo que os recursos utilizados são apenas de hardware.

Processo

1º passo

Inserir memória EPROM num soquete "TEXTOOL".

2º passo

Gerar todos os sinais de controle e endereçamento por hardware.

3º passo

Inserir manualmente os dados a serem gravados em um buffer de memória.

4º passo

Gerar um comando para que os dados armazenados no buffer sejam transferidos para a posição endereçada na EPROM.

TE 60123

Gravador de EPROM utilizando um sistema de microcomputador

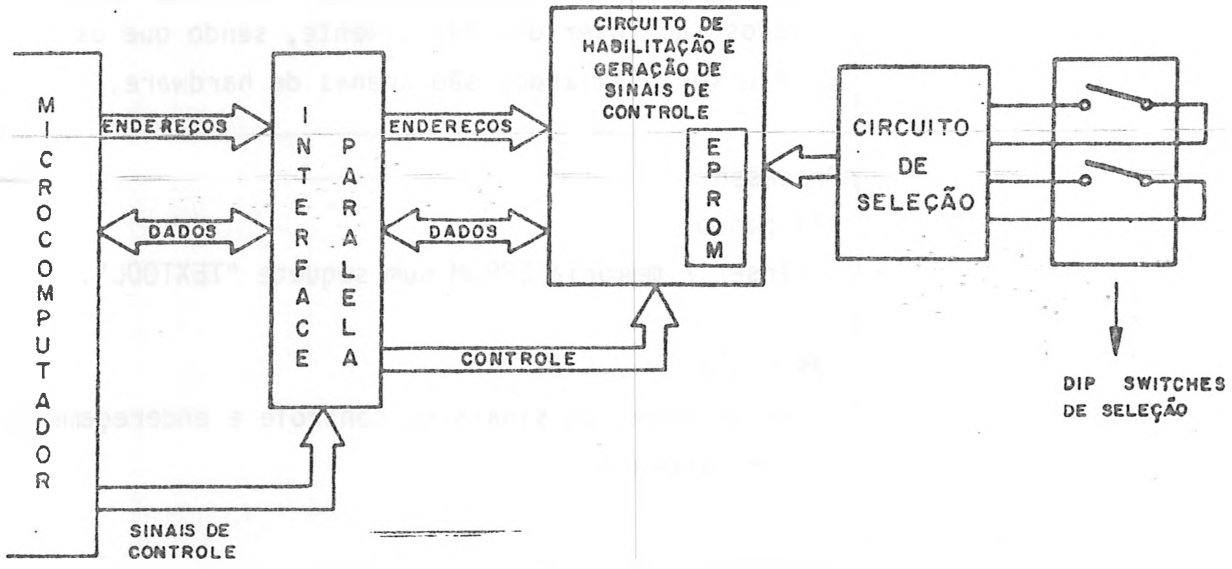
Descrição

É um processo de gravação de memória EPROM onde os dados são inseridos através de um sistema de microcomputador, sendo que os recursos utilizados são de software e hardware.

Comentário

Neste processo de gravação os sinais de controle e endereçamento são gerados pelo sistema, através de programas dedicados a esta função normalmente fornecidos com o gravador de EPROM.

Estrutura



Parte	Função
Interface paralela	Viabilizar a conexão do gravador aos barramentos de dados, endereços e controle do microcomputador.
Circuito de habilitação e geração dos sinais de controle	Gerar os sinais de controle de programação da EPROM de acordo com as especificações do fabricante da memória
Circuito de seleção	Selecionar o tipo da memória a ser gravado. Normalmente esta seleção é feita por dip-switches ou traps.

Características

O software dedicado de um gravador de EPROM deste tipo é normalmente fornecido em um disco flexível, possibilitando:

. verificar se a EPROM está apagada;
. ver o conteúdo de um EPROM;
. copiar EPROMs;
. programar EPROMs.

Comentário

Este pacote de programas dedicados devem ser compatíveis com o sistema de microcomputador a ser utilizado.

Palavras-chave

GRAVADOR DE EPROM

PALAVRAS DE MEMÓRIA

TE 60124
(1 página)

Descrição

Palavras de memória são as informações codificadas em binário contidas em cada uma das posições de memória.

Classificação

Estas palavras podem ser:

. números binários simples armazenados individualmente em cada posição de memória;

. números binários compostos de palavras múltiplas, ocupando várias posições de memória;

. quaisquer dados representados em codificação binária tais como:

  a) números decimais representados em binário (código BCD)

  b) números em 7 segmentos

  c) caracteres (ex. código ASCII)

. códigos de instrução (programa para microprocessador).

Comentário

Ao se examinar o conteúdo de qualquer palavra de memória, é impossível determinar se esta palavra contém um dado numérico, um código ou uma instrução, a não ser que se conheça a convenção adotada para a aplicação em que a memória foi utilizada.

Palavras-chave

PALAVRAS DE MEMÓRIA

MEMÓRIA DE ACESSO ALEATÓRIO - MEMÓRIA RAM (RAM - random access memory)

TE 60111
(3 páginas)

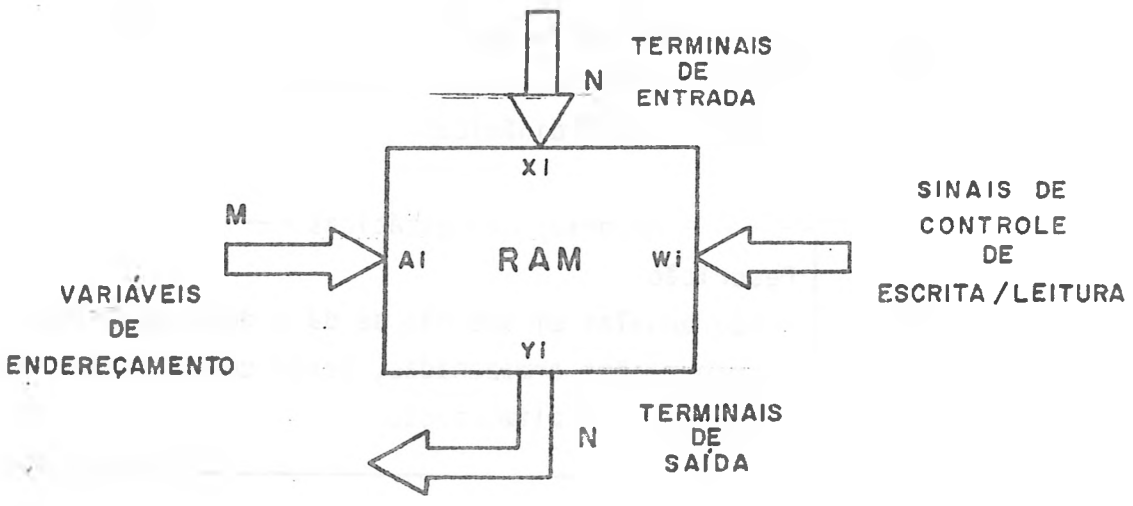
Descrição

Define-se como memória de acesso aleatório - RAM o dispositivo de memória, do tipo volátil, no qual é possível escrever ou ler dados binários.
A memória RAM, tipo estado sólido, é um circuito integrado LSI.

Estrutura

Uma memória de acesso aleatório é constituída por um certo número de células, capazes de armazenar uma informação binária.
Estas células são agrupadas em uma forma matricial do tipo N x m, a fim de permitir o acesso aleatório para escrita ou leitura de informações, em qualquer uma das m células, através do endereçamento da sua posição N.
Normalmente, dentro da estrutura matricial, as células têm endereço único. Entretanto são acessadas na configuração de palavras binárias, de m bits, e cada palavra possui um único endereço de acesso N.

Ilustração



Parte	Função
Xi	São terminais de entrada de dados em forma de palavra binária, habilitados quando a operação de escrita na memória é selecionada.
Yi	São terminais de saída de dados em forma de palavra binária, habilitados quando a operação de leitura na memória é selecionada.
Ai	São terminais que permitem o endereçamento da posição de memória a ser acessada pela operação escrita ou leitura na memória.
Wi	São sinais de controle que habilitam o acesso a memória e também determinam se a função a ser efetuada é de escrita ou leitura na memória.

Classificação As memórias RAM podem ser classificadas em:

. Memórias RAM estáticas

. Memórias RAM dinâmicas

Memórias RAM estáticas

Descrição

São aquelas em que não se dá a degradação das informações armazenadas, desde que se mantenha constante a alimentação.

Memórias RAM dinâmicas

Descrição

São aquelas nas quais se dá, com o tempo, a degradação das informações armazenadas, mesmo sendo mantida constante a alimentação. Tais memórias necessitam, então, ser refrescadas periodicamente para o restabelecimento das informações nelas gravadas.

Aplicação As memórias RAM são utilizadas principalmente em sistemas de microcomputador.

Palavras-chave

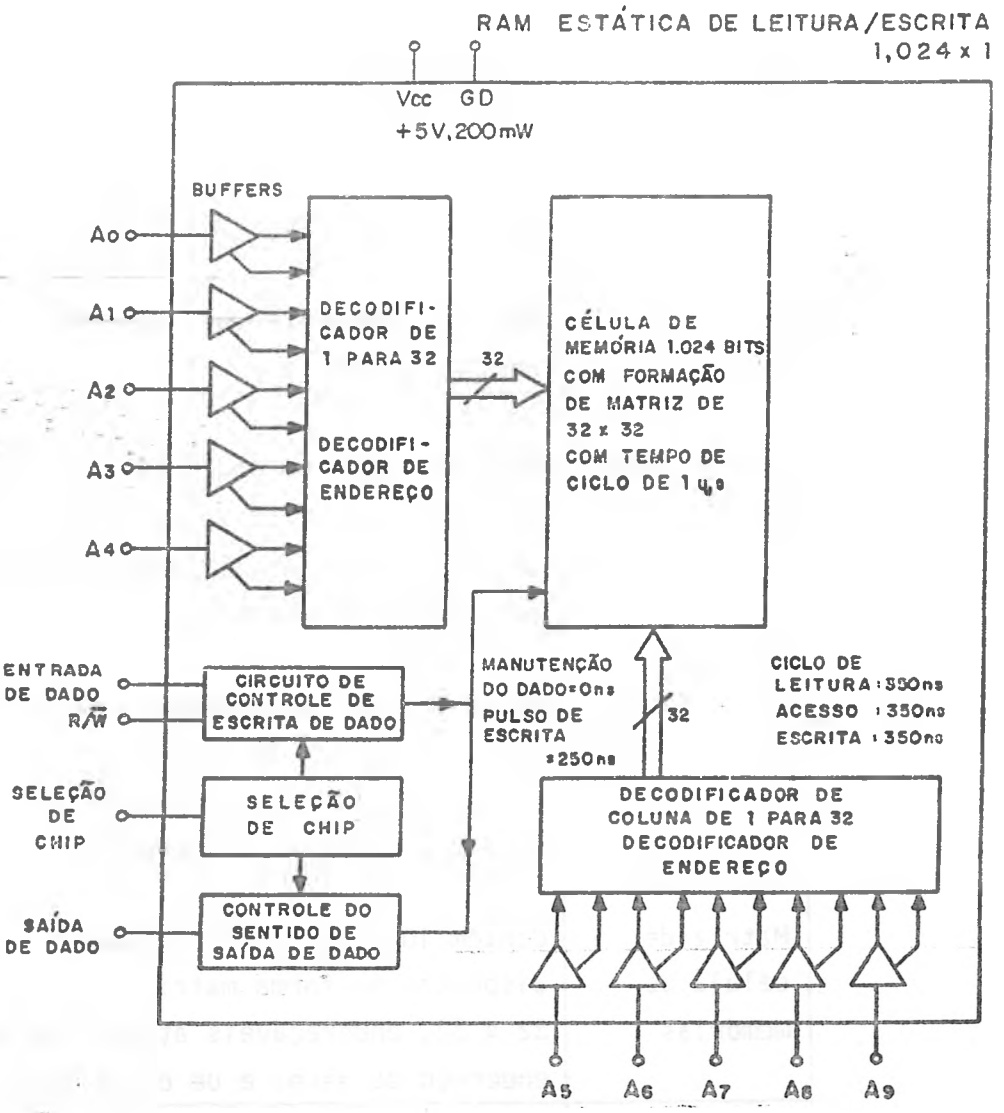
MEMÓRIAS RAM

MEMÓRIA RAM ESTÁTICA TE 60112

(3 páginas)

Descrição A memória RAM estática é um dispositivo de memória de escrita ou leitura, capaz de manter indefinidamente as informações armazenadas, enquanto a alimentação dessas for mantida constante.

Estrutura A estrutura típica de uma RAM estática de 1024 posições de um bit é apresentada na figura abaixo.



Parte	Função
De Ao a A4	Endereço das linhas da matriz 32 x 32 onde estão as células de armazenamento.
De A5 a A9	Endereço das colunas da matriz 32 x 32 onde estão as células de memória.
Data-in	Terminal de entrada de dados.
Data-out	Terminal de saída de dados.
R/W	Sinal de controle para habilitação da escrita ou leitura da memória.
CE	Sinal de controle que habilita o acesso à memória.
Decodificador de linha	Decodificador de endereços de linha de 5 entradas x 32 saídas que seleciona uma das 32 linhas da matriz de células de memória.
Decodificador de coluna	Decodificador de endereços de coluna de 5 entradas x 32 saídas que seleciona uma das 32 colunas da matriz de células de memória.
Matriz de célula de memórias	Contém 1024 células de memória, dispostas na forma matricial de 32 x 32, endereçáveis através de um endereço de linha e um de coluna.
Célula de memórias	Estas células de memória nada mais são que circuitos flip-flops tipo D.

Circuito de escrita e controle de dados	Circuito que recebe o sinal R/W que habilita a escrita através do terminal data-in, colocando data-out em tri-state.
Chip select	Circuito que habilita o acesso à escrita ou leitura na memória.
Circuito de saída de dados	Circuito que permite a leitura de dados da RAM.

Comentário

As memórias RAM estáticas podem armazenar palavras binárias de um a oito bits, endereçáveis individualmente. Normalmente, em RAMs de mesmo número de bits, as que armazenam palavras maiores de uma só vez apresentam um custo mais elevado do que aquelas que armazenam palavras menores.

Exemplo

Uma RAM de 1024 posições de quatro bits apresenta um custo menor que outra RAM de 512 posições de oito bits, sendo que as mesmas possuem a mesma capacidade em número de bits: 4096.

Aplicação

As memórias RAM estáticas são utilizadas principalmente em sistemas de microcomputador que necessitam de uma área de memória de rascunho pequena, e onde o consumo de energia não constitui um parâmetro crítico.

Palavras-chave

MEMÓRIA RAM ESTÁTICA

MEMÓRIA RAM DINÂMICA

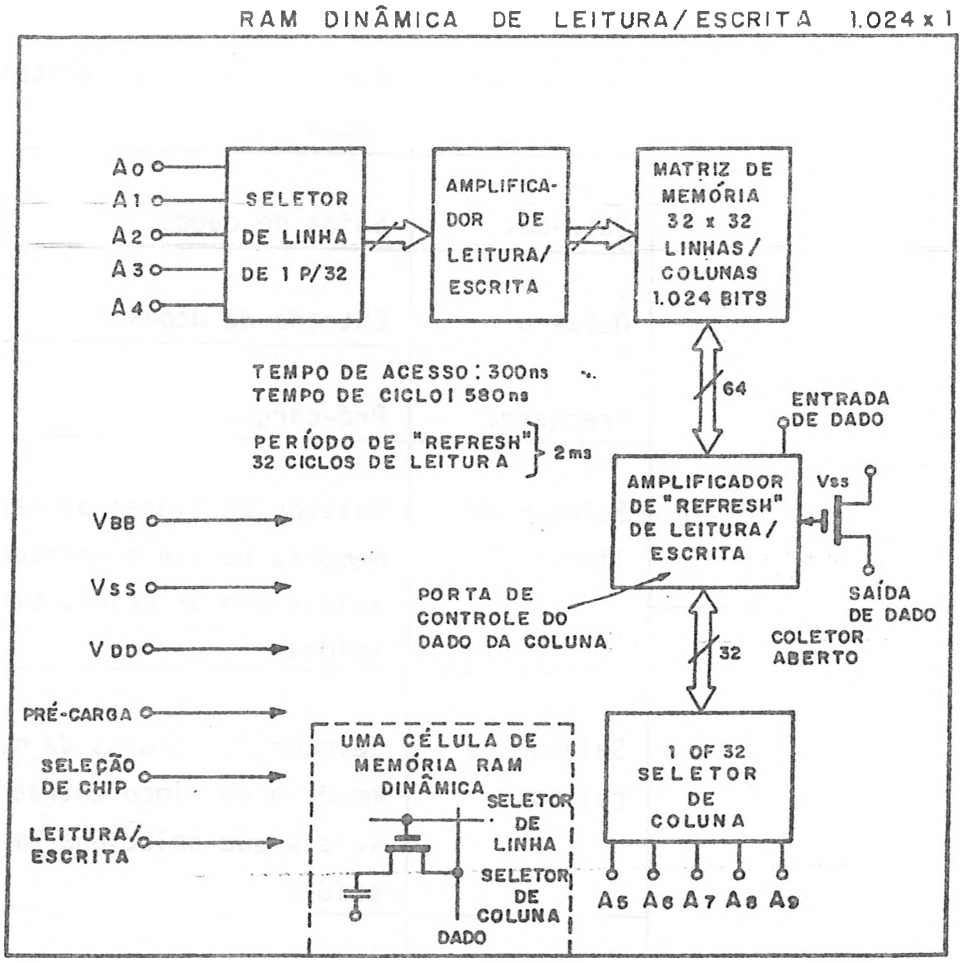
TE 60113
(5 páginas)

Descrição

A memória RAM dinâmica é um dispositivo de memória de escrita ou leitura, cujo conteúdo armazenado se deteriora com o tempo, necessitando pois de uma ação de refrescamento à fim de que as informações gravadas não se apaguem.

Estrutura

A estrutura típica de uma RAM dinâmica de 1024 posições de um bit é apresentada na figura abaixo.



Parte	Função
De A_0 a A_4	Endereços das linhas da matriz, células de memória.
De A_5 a A_9	Endereços das colunas da mesma matriz.
V_{BB}, V_{SS}, V_{DD}	Alimentações.
Chip enable	Habilita acesso à memória.
Read/write	Habilita escrita (**write**) ou leitura (read).
Data-out	Saída de dados.
Data-in	Entrada de dados.
Precharge	Pré-carga.
Seletor de linhas	Seletor de linhas da matriz de memória de cinco entradas para 32 saídas que seleciona uma das 32 saídas.
Seletor de colunas	Seletor de colunas da matriz de memória de cinco entradas para 32 saídas que seleciona uma das 32 saídas.

Circuito de refrescamento	. Encaminha os sinais de seleção das colunas e refresca conteúdo das células de memória da matriz periodicamente. . Faz controle de encaminhamento dos dados a serem lidos ou gravados.
Matriz de memória	Onde estão localizados os módulos de memória em forma de estrutura matricial de 32 colunas x 32 linhas.
Módulos de memória	Os módulos de memória são onde se armazenam os bits de informações. Cada módulo de memória armazena um único bit.

Processo

A RAM dinâmica tem como célula de memória basicamente um capacitor que, conforme o nível lógico que se quer armazenar, pode estar carregado ou descarregado.

Se...	... então...
... desejarmos manter uma célula carregada...	... deveremos, em intervalos especificados pelo fabricante, alimentar seus terminais de refrescamento, não deixando, desta maneira, que o nível de tensão armazenado caia abaixo do limite mínimo.
... desejarmos manter uma célula descarregada...	... deveremos provocar um curto-circuito.

Comentário

Um capacitor carregado, mesmo com os terminais abertos, acaba descarregando-se, perdendo assim a informação nele armazenada. Por esta razão se faz presente a operação de refrescamento (**refresh**).

Vantagem

Normalmente, as RAMs dinâmicas são circuitos integrados que armazenam bits endereçáveis individualmente, têm custo inferior ao das memórias RAM estáticas e consomem menos energia do que estas.

Desvantagem	As RAMs dinâmicas exigem circuitos especiais para efetuar a operação de refrescamento, o que as torna inconvenientes quando é necessário implementar apenas uma pequena área de RAM.
Aplicação	As RAMs dinâmicas são usadas em sistemas de microcomputador onde há necessidade de uma área de memória de rascunho relativamente extensa.

Palavras-chave

MEMÓRIA RAM DINÂMICA

FIRMWARE

TE 70020
(2 páginas)

Conceituação

Firmware é basicamente um programa ou programas gravados de maneira não volátil em uma memória que é acessada sempre que uma seqüência de instruções deva ser realizada e eventualmente repetida.

Aplicação

Como exemplo de aplicação, vejamos o caso de uma impressora que, ao ser ligada, sempre tem de cumprir uma série de procedimentos, antes de poder receber dados do sistema para imprimi-los.
Essa série de procedimentos de inicialização é uma parte do firmware da impressora pois, sempre que é ligada a impressora, o firmware assume o controle da máquina e verifica, por exemplo, se:

. há papel na máquina;
. o carro de impressão está na margem esquerda;
. não há problemas de hardware (eletrônica).

De certa forma podemos dizer que o firmware é a inteligência da máquina.

Processo

O firmware fica normalmente gravado numa memória do tipo EPROM ou PROM e é transparente para o usuário, ou seja, é através dele que uma série de operações é automaticamente realizada.

Exemplo

Um programa-monitor é uma espécie de firmware pois, ao ligarmos um microcomputador, uma das primeiras tarefas executadas é a varredura do teclado, a fim de identificar a tecla que será apertada pelo usuário.

Palavras-chave

FIRMWARE

DISPOSITIVOS DE ENTRADA E SAÍDA — TE 60030
(Input output devices) — (1 página)

Descrição

As portas de entrada e saída de dados de um microcomputador são responsáveis pela comunicação do sistema com o mundo exterior. Dessa maneira, dados que devem ser enviados para fora ou dados que devem ser lidos de dispositivos externos são processados via dispositivos de entrada e saída.

Classificação

Podemos classificar os dispositivos de entrada e saída quanto:

ao modo de operação

. fixo: São os dispositivos que funcionam numa configuração inflexível que não pode ser alterada facilmente.

. programável: São os dispositivos que podem ter sua configuração alterada, através de instruções de programação.

à forma de entrada e saída de dados

. serial: É a forma pela qual os dados entram e saem no formato serial.

. paralelo: É a forma pela qual os dados entram e saem no formato paralelo.

Palavras-chave

DISPOSITIVOS DE ENTRADA E SAÍDA

MICROPROCESSADOR
(Microprocessor)

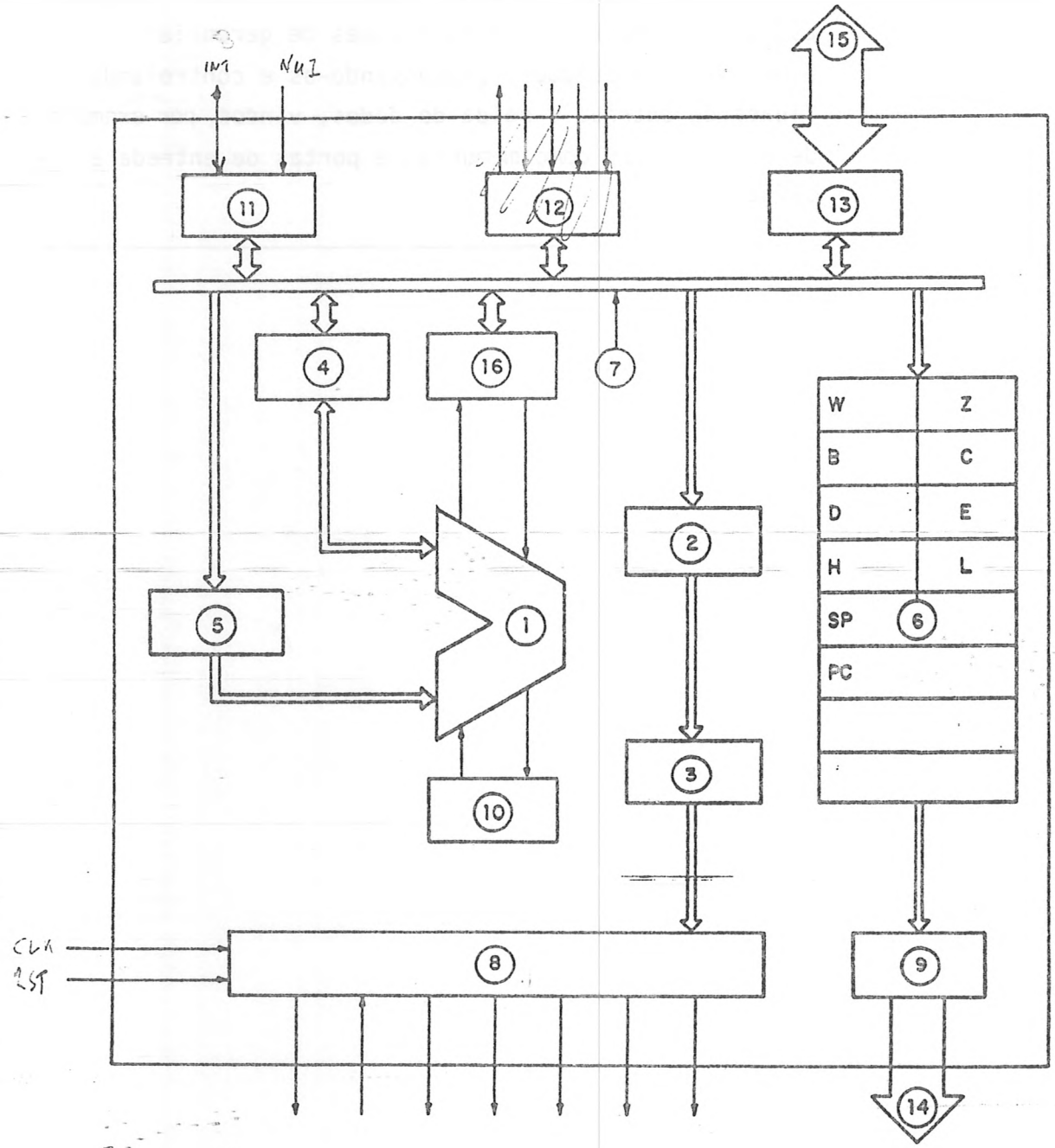
TE 60050
(6 páginas)

Descrição

O microprocessador é normalmente um circuito integrado LSI que congrega circuitos capazes de gerenciar programas armazenados, processando-os e controlando fluxos de entrada e saída de dados, vindos, por exemplo, de dispositivos como memórias e portas de entrada e saída.

Estrutura

A estrutura interna de um microprocessador típico pode ser vista abaixo.



Processo

A tabela de função das partes é a seguinte:
seguinte tabela de função:

Bloco	Nome	Função
1	Unidade lógica e aritmética (ULA)	Executar todas as operações lógicas e aritméticas do microprocessador. Ex.: Operações de soma, subtração, inversão, soma lógica (OR), produto lógico (AND), etc.
2	Registrador de instrução	Receber as instruções provenientes da memória do programa e enviar essas instruções ao decodificador.
3	Decodificador de instrução	Decodificar os códigos das instruções e acionar os sinais de controle através da unidade de controle. Esses sinais podem ser de natureza interna do microprocessador ou externa, para controle de outros circuitos integrados do sistema.
4	Registrador temporário	Armazenar temporariamente um dos operandos que será utilizado pela unidade lógica e aritmética.

5	Acumulador	Armazenar dados que são utilizados pela unidade lógica e aritmética, assim como servir de circuito intermediário para operações de leitura e escrita em dispositivos externos ao microprocessador, além de armazenar o resultado das operações realizadas pela ULA.
6	Conjunto de registradores	Armazenar informações que podem ser dados ou endereços, para serem processados internamente ou externamente ao microprocessador.
7	Bus de dados interno	Permitir a interligação entre os vários blocos de circuitos internos do microprocessador.
8	Unidade de controle	Gerar sinais de controle para os circuitos integrados periféricos a fim de mantê-los em sincronismo com o microprocessador e possibilitar o seu controle.
9	Buffer/latch de endereço	Reforçar e reter temporariamente os bits que formam o bus de endereço externo.

10	Registrador para ajuste decimal	Ajustar o resultado de determinada operação, realizada pela ULA, para obter um resultado compatível com o sistema de numeração decimal.
11	Controle de entrada e saída serial	Promover a comunicação externa de dados que devam entrar e sair do microprocessador no formato serial.
12	Controlador de interrupções	Decodificar sinais externos examinando prioridades e alterando a seqüência do programa segundo essas prioridades.
13	Buffer/latch de dados	Reforçar e reter temporariamente os bits que formam o bus de dados. Obs.: O bus de dados é bidirecional, podendo receber dados externos ou enviar dados para fora.
14	Bus de endereço	Fazer o transporte dos bits que formam o bus de endereço para fora do sistema. São normalmente em número de 16 linhas.

15	Bus de dados	Fazer o transporte dos bits que formam o bus de dados bidirecional. São normalmente em número de 8 linhas.
16	Registrador de flags	Indicar através de níveis lógicos os resultados das operações realizadas pela ULA. Ex.: Se a operação realizada teve resultado negativo, determinado sinal vai a nível lógico 1 e pode ser utilizado para mudar a seqüência normal do programa.

Exemplo

Alguns microprocessadores que utilizam a configuração típica descrita são:

. 8080
. 8085 do fabricante INTEL
. Z80 do fabricante ZILOG
. 6800 do fabricante MOTOROLA

Comentário

Todos os microprocessadores que utilizam palavra de 8 bits.

Palavras-chave

MICROPROCESSADOR

UNIDADE LÓGICA E ARITMÉTICA DE UM MICROPROCESSADOR - ULA INTERNA

TE 60230
(1 página)

Descrição

A ULA interna é a unidade que realiza a manipulação real de dados em um microprocessador, efetuando as operações lógicas e aritméticas determinadas pela unidade de controle deste microprocessador.

Estrutura

A ULA interna opera com dados binários em incrementos de palavras de memória, ou seja, a ULA de um microprocessador de oito bits irá operar com unidades de oito bits.

Aplicação

A unidade lógica aritmética interna de um microprocessador serve para efetuar as seguintes operações:

. Adição binária
. Operações booleanas
. Complementação de uma palavra de dados
. Deslocamento de uma palavra de dados de um bit para direita e/ou esquerda

Comentário

Quaisquer outras operações mais complexas de manipulação de dados requerida de um microprocessador devem ser construídas a partir destes poucos elementos lógicos da ULA.

Palavras-chave

ULA DE UM MICROPROCESSADOR

REGISTROS INTERNOS DE UM MICROCOMPUTADOR TE 60200

(3 páginas)

Descrição

Um microprocessador deve possuir um ou mais registros internos, que servem como elementos de estocagem temporária de dados, a fim de efetuar as operações necessárias requeridas ao processamento e transferências destes dados provenientes do meio externo (memórias e dispositivos de entradas e saídas).

Classificação

Estes registradores são classificados em tipos, de acordo com a aplicação que lhe é destinada. São estes os tipos:

. Acumuladores
. Registradores de uso geral
. Contador de programas
. Registro de instrução
. Ponteiro de pilha (stack-pointer)

Acumuladores

Descrição

São registradores nos quais os dados buscados na memória ou nos dispositivos de entrada são armazenados temporariamente para processamento das operações requeridas, principalmente as que envolvem a unidade lógica aritmética interna. Os acumuladores têm o mesmo número de bits da palavra de dados do microcomputador.

Comentário

Todo microprocessador deve possuir ao menos um acumulador.

Registradores de uso geral

Descrição

São registradores usados para armazenamento de informações temporárias com estrutura semelhante à dos acumuladores. Estes registradores geralmente têm o mesmo número de bits da palavra de dados do microcomputador.

Comentário

As informações neles armazenadas são normalmente dados auxiliares, usados nas operações efetuadas pelo microprocessador.

As operações lógicas e/ou aritméticas normalmente são efetuadas entre o conteúdo de um acumulador e o conteúdo de um dos registros auxiliares de uso geral.

Contador de programas

Descrição

É um registro que contém o endereço da palavra de memória, na qual será lido o código de instrução a ser executado pelo microprocessador. Ele é incrementado cada vez que terminar um ciclo de execução de instrução.

Este registro tem o número de bits da palavra de endereço do microcomputador.

Registro de instrução

Descrição

É o registro onde deve ser armazenado o código de operação acessado ao código de instrução, para ser interpretado pela unidade de controle (UC) do microprocessador.

O tamanho deste registro depende da configuração interna da UC, que determina o número de operações possíveis de efetuar.

Ponteiro de pilha (stack-pointer)

Descrição

É um registro cujo conteúdo é um endereço de memória RAM, que permite o acesso a esta memória, organizando-a na estrutura de pilha em uma forma imediata.

Comentário

É muito útil quando é necessário salvar conteúdos de registradores durante a execução dos programas. Tem o mesmo número de bits da palavra de endereço do microprocessador.

Palavras-chave

REGISTRADORES INTERNOS DA UCP

FLAGS DE ESTADO
(Flag de status)

TE 60250
(4 páginas)

Descrição

Flags de estado são o conjunto de portas lógicas dedicadas (flip-flops) que armazenam dígitos binários simples, que são automaticamente setados ou resetados, de acordo com o resultado das operações da ULA.

Classificação

Os flags de estado são classificados em:

. Flag carry ou vai um
. Flag auxiliar carry ou vai um intermediário
. Flag zero
. Flag de sinal
. Flag de overflow ou sobrecarga
. flag de paridade

Flag carry ou vai um

Processo

Este flag é setado quando efetuada uma operação na ULA interna do microprocessador que faça ocorrer um sinal de vai um.

Exemplo

```
     10101011
   + 11111011
  1  10100110
  |
carry
```

Flag auxiliar ou vai um intermediário

Descrição

É um flag existente nos microprocessadores de oito bits.

Processo

É setado quando se realizam operações com a ULA e ocorre sinal de vai um para operação com os quatro bits menos significativos das palavras.

Aplicação

É usado nas operações com números, no código BCD.

Exemplo

```
                4 bits menos significativos
    7 6 5 4 3 2 1 0   n. bits
    1 0 1 0 1 0 1 1
  + 0 0 0 1 1 0 1 0
  ---------------
    1 1 0 0 0 1 0 1
```

Flag auxiliar carry = [1]

Flag zero

Processo

Este flag é setado quando a operação de manipulação de dados pela ULA resulta igual a zero.

Exemplo

```
    10101111
  - 10101111
  ----------
    00000000 → Flag zero = [1]
```

Flag de sinal e flag de overflow

Descrição

A utilização do bit mais significativo de memória como bit de sinal quando se realizam operações binárias dá origem a dois flags de estado:

. Flag de sinal

. Flag de overflow

Processo

. O flag de sinal é setado quando o bit mais significativo da palavra binária resultante de alguma operação aritmética for 1, isto é, número negativo, e resetado quando o bit for 0, indicando que o número é positivo.

. O flag de overflow é setado quando a operação "OU EXCLUSIVO" dos carrys do bit de sinal e do penúltimo bit da palavra resultante de alguma operação for igual a 1, significando que o resultado representa um número muito grande para caber no espaço disponível de dados.

Exemplo

```
   01011010          5A
 + 01101011        + 6B
   11000101        - 3B ?
      |               |
   Cs = 1 Cp = 1      Flag de overflow = 1
   Cs + Cp = 1        Flag de sinal = 1
```

Flag de paridade

Processo

O flag de paridade é setado toda vez que em uma operação de transferência de dados for detectado um byte com paridade errada.

Comentário

Este flag será ignorado na maioria das vezes, uma vez que só tem algum significado real quando o conteúdo de uma palavra de memória está sendo interpretado como um código de caracteres.

Comentário — As instruções que setam ou resetam os flags de estado, e também as que não os alteram, são cuidadosamente escolhidas pelos projetistas de microcomputadores.

Palavras-chave

FLAGS DE ESTADO
FLAGS DE STATUS

GERADOR DE SINAL DE CLOCK

TE 60220
(3 páginas)

Descrição

O gerador de sinal de clock é um circuito lógico de suporte para geração dos sinais de temporização de um microprocessador.

Classificação

O gerador de sinal de clock pode ser:

. Interno ao microprocessador, fazendo parte de sua arquitetura.

. Externo ao microprocessador, na forma de circuito integrado ou discreto.

Comentário

Os microprocessadores mais recentes não contêm a lógica de clock internamente, necessitando de um dispositivo externo para geração deste clock.

A maioria dos microprocessadores possui sua própria lógica de clock internamente desde 1978.

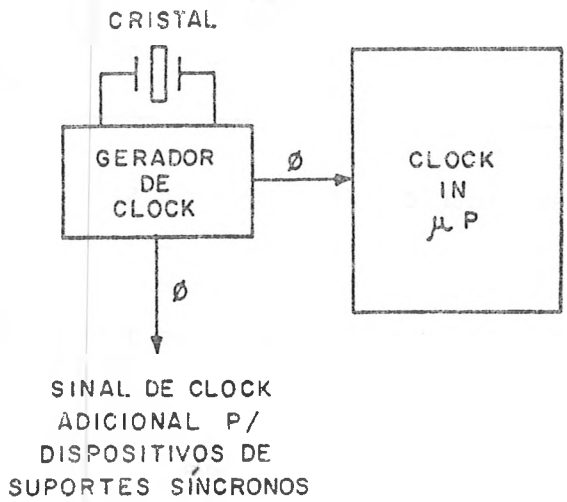
Processo

. A lógica responsável pela geração do sinal de clock normalmente baseia a freqüência deste sinal na freqüência ressonante de um cristal.

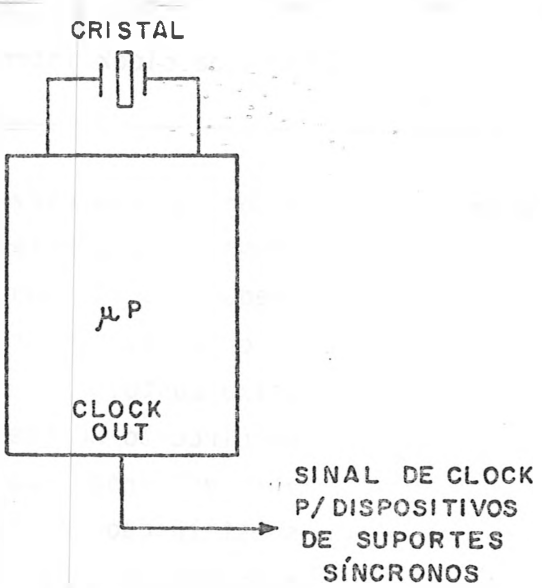
. Os cristais apropriados são produzidos em massa e a baixo custo.

. Um circuito RC (resistor-capacitor), dimensionado para valores apropriados, pode vir a ser usado, em substituição do cristal, fornecendo um controlador de freqüências mais barato, porém bem menos preciso.

Estrutura — Gerador de clock externo



Gerador de clock interno



Comentário

Um microprocessador que gera seu próprio sinal de clock normalmente oferece a operação de entrar com um sinal de clock gerado externamente.

Isto é necessário, uma vez que os dois ou mais microprocessadores podem coexistir em um mesmo sistema, sendo assim eles deverão ser alimentados pelo mesmo clock para evitar que trabalhem defasados, gerando graves problemas de temporização.

Palavras-chave

GERADOR DE CLOCK

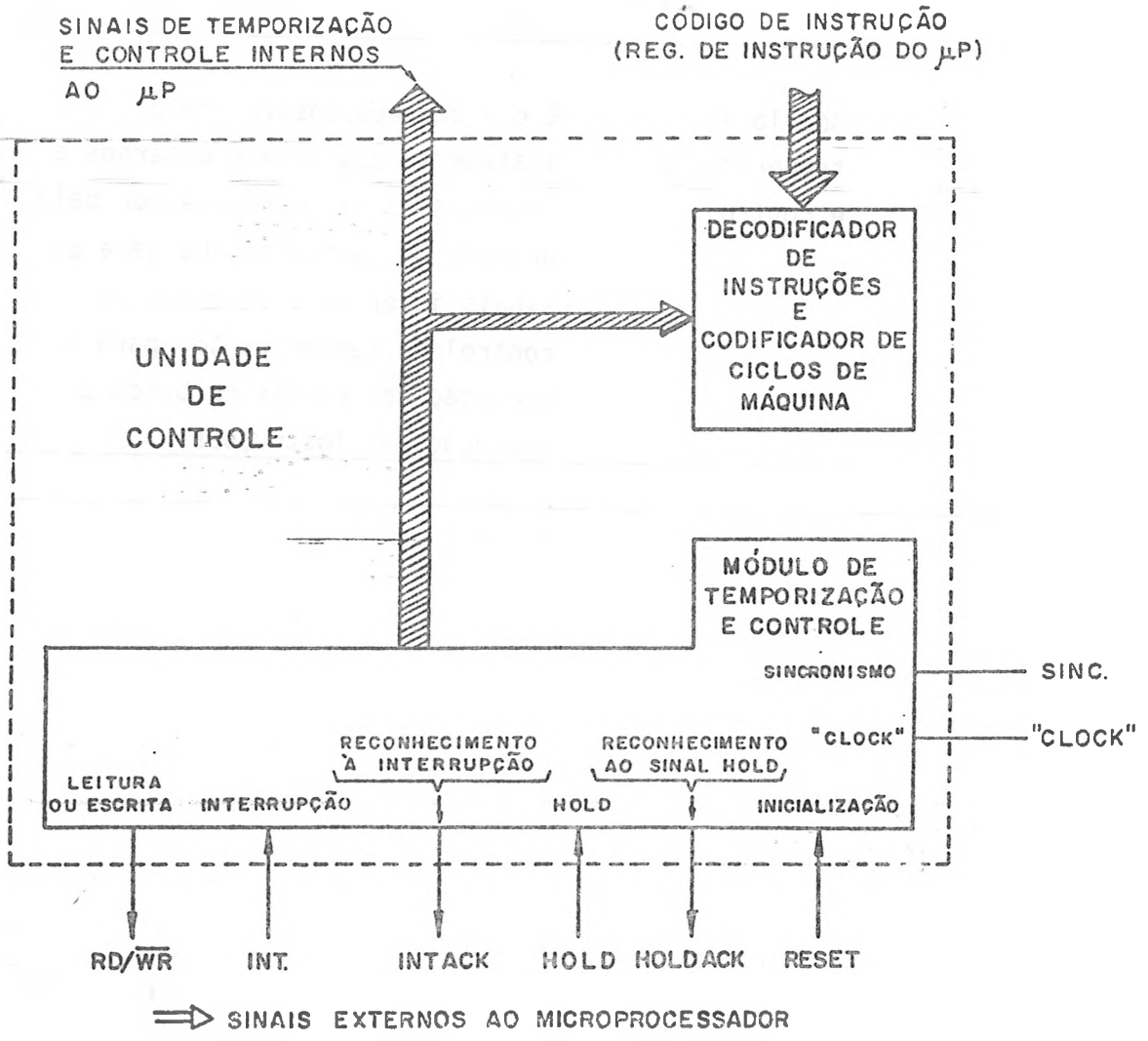
UNIDADE DE CONTROLE DE UM MICROPROCESSADOR TE 60210

(5 páginas)

Descrição

A unidade de controle é a parte interna de um microprocessador, responsável pela decodificação do código de instrução, contido no registro de instrução e geração dos sinais de controle e temporização, requeridos pela instrução decodificada ou provenientes da lógica externa, tanto para os módulos internos quanto para os externos deste microprocessador.

Estrutura



Parte	Função
Decodificador de instruções e codificador de ciclos de máquina	É o bloco que, através da execução de um microprograma nele implementado, decodifica os códigos de instrução, provenientes do registro de instrução interno do microprocessador e envia os sinais decoficados para o bloco de temporização e controle. Tais sinais são interpretados pela sua lógica e geram, por sua vez, sinais convenientes para a execução da instrução requerida.
Módulo de temporização e controle	É o bloco responsável pelo tratamento dos sinais externos e internos ao microprocessador pela unidade de controle, que gera os sinais internos e externos de controle e temporização, para a execução dos ciclos de busca e execução das instruções.

Características É característica da unidade de controle de uma CPU a manipulação dos seguintes sinais:

. Sinais de entrada

. Sinais de saída

Sinais de entrada

Os sinais de entrada dividem-se em sinais externos e internos.

Sinais externos ao microprocessador

. Sinais de clock, provenientes de um gerador de clock externo ou de um cristal.

. Sinais de interrupção não-mascarada e mascarada, gerados por dispositivos externos.

. Sinal de HOLD, gerado por dispositivos externos para congelar as operações do microprocessador.

. Sinal de RESET ou inicialização, o qual determina que o microprocessador deve assumir as condições de inicialização das suas operações, como zerar o conteúdo do contador de programa.

Comentário

Os sinais externos de entrada para controle em um microprocessador variam de acordo com a concepção de seu projeto.

Sinais internos ao microprocessador

O principal sinal de entrada interno ao microprocessador é o código de instrução, proveniente do registro de instrução.

Sinais de saída

Os sinais de saída dividem-se nos que vão para o meio externo do processador e naqueles que vão para o meio interno.

Sinais que vão para o meio externo do microprocessador

. Sinais para habilitação de escrita ou leitura em dispositivos externos (RD/$\overline{WR}$).
. Sinais de reconhecimento a interrupções (INTACK) ou ao sinal HOLD (HOLDACK).
. Sinal de sincronismo para dispositivos externos.

Comentário

Também, neste caso, os sinais externos enviados pelo microprocessador variam de acordo com a concepção de seu projeto.

Sinais de controle para o meio interno

. Sinais de temporização dos módulos internos.
. Sinais de habilitação dos buffers e latches que fazem o controle do fluxo de informações nos barramentos internos do microprocessador.
. Sinais de controle que determinam os ciclos de busca e de execução.
. Sinais de comando que determinam que função a ULA deverá executar.
. Demais sinais de controle e temporização para execução das instruções implementadas.

Processo	O funcionamento da unidade de controle baseia-se na execução de um microprograma implementado. Cada instrução proveniente do meio externo implica na execução de várias microinstruções deste microprograma. Estas microinstruções são executadas seqüencialmente na freqüência determinada pelo clock do microprocessador. O microprograma é que determina a potencialidade e abrangência do conjunto de instruções do microprocessador.

Palavras-chave

UNIDADE DE CONTROLE

INSTRUÇÕES DE UM MICROPROCESSADOR

TE 60270
(7 páginas)

Introdução

As informações manipuladas por um microprocessador são de dois tipos a saber:

. Instruções
. Dados

Descrição

Instruções são comandos que governam a transferência de informações dentro da máquina, especificam a operação lógica ou aritmética a ser realizada, onde buscar o dado operando e, também, a transferência de dados entre a máquina e o meio exterior.
Esses comandos são dispostos de forma que sejam logicamente relacionados a endereços de memória externa, que são acessados seqüencialmente pelo microprocessador.
A esta disposição de instruções denominamos programa.
Cada microprocessador em particular possui um conjunto de instruções inerentes a suas características de projeto.
Este conjunto de instruções identifica todas as operações específicas que o microprocessador pode executar.

Classificação

Os tipos de instruções de um microprocessador variam de acordo com a sua concepção e arquitetura interna.
Genericamente podemos destacar entre outros os seguintes tipos:

. Instruções de entrada e saída de dados
. Instruções de referência à memória
. Instruções de transferência de dados entre registros internos
. Instruções de desvio e desvio a sub-rotinas
. Carregamento direto
. Instruções de interrupções
. Instruções de operações lógicas
. Instruções de operações aritméticas

. Instruções de status ou estado ou ainda controle de máquina
. Instruções de parada (HALT)

Instruções de entrada e saída de dados

Descrição

São instruções que permitem o acesso à leitura ou escrita de dados, em dispositivos periféricos pelo microprocessador.

Instruções de referência à memória

Descrição

São instruções usadas pelo microprocessador para movimentar dados dos registradores internos para a memória ou vice-versa.

Classificação

Essas instruções classificam-se em:

. **Instruções de carregamento**
Movem conteúdos de posições de memória para os registradores internos da UCP.

. **Instruções de armazenamento**
Movem conteúdos dos registradores internos da UCP para posições de memória externa.

Instruções de transferência de dados entre registros internos

Descrição

São instruções que movem dados de um registro para outro internamente na UCP.

Instruções de status

Descrição

São instruções que permitem setar ou resetar os flags de status do microprocessador ou ainda acessar seus conteúdos.

Instruções de parada (HALT)

Descrição

Esta instrução quando executada bloqueia todas as operações do microprocessador, fazendo com que seus barramentos externos sejam colocados em alta impedância.

O microprocessador somente sai deste estado quando ocorre um sinal de interrupção externa ou de reset.

Estrutura

Cada instrução é formada por uma, ou mais palavras binárias, denominada código-objeto ou código de isntrução.
Cada código de instrução, por sua vez, é formado por duas partes:
. Código de operação
. Operando

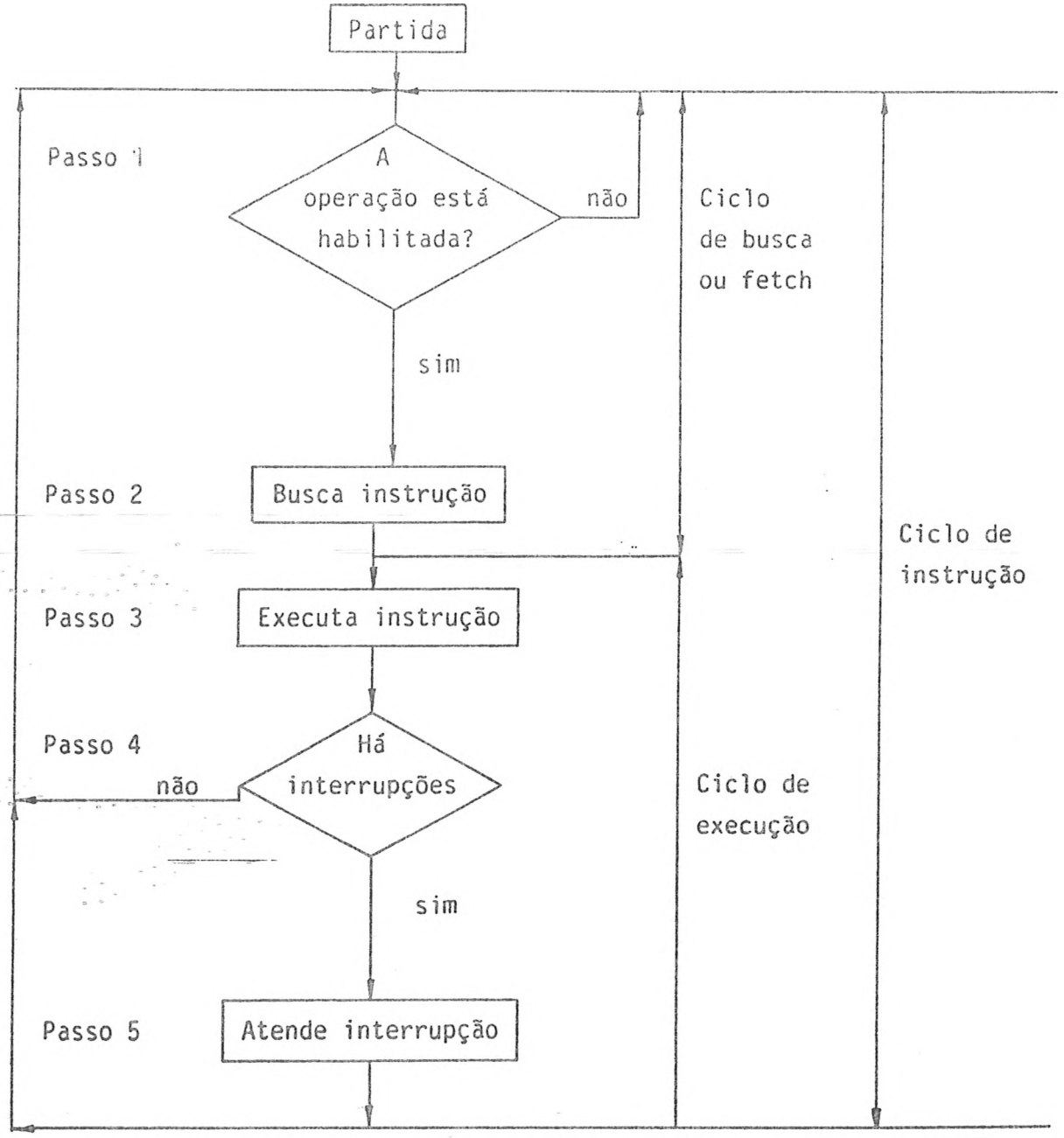
Código de operação
É uma ou mais palavras binárias que identificam a instrução a ser executada pelo microprocessador.

Operando
É uma ou mais palavras binárias que representam um endereço de dispositivo externo (posição de memória ou dispositivo de entrada e saída) ou ainda um dado binário a ser manipulado.
Denominamos programa-objeto ou programa em linguagem de máquina a um conjunto de códigos-objetos ordenados de uma maneira lógica.

Comentário
Cada código-objeto refere-se a um mnemônico. Isto facilita muito o trabalho de elaboração de programas. Os programas elaborados que utilizam estes artifícios são chamados programa fonte ou programa em linguagem Assembly ou de montagem.
Outro artifício utilizado é o de representar os códigos objetos em Hexadecimal.
De qualquer forma, para executar um programa, este deve estar codificado em binário e, para traduzir uma forma de escrevê-lo em outra, usa-se o recurso manual ou automático, através de dispositivos de Software ou Hardware da própria máquina.

Processo

O microprocessador segue o seguinte processo seqüencial para efetuar as operações requeridas por uma instrução:



Palavras-chave

CONJUNTO DE INSTRUÇÕES

DIAGRAMAS DE TEMPO

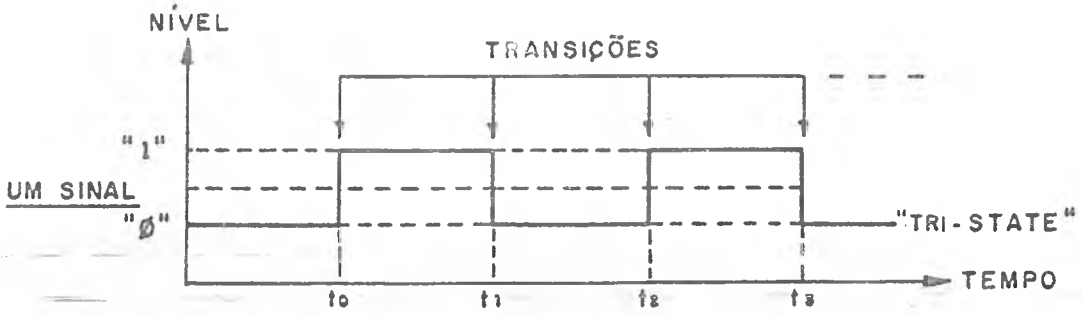
TE 60260
(5 páginas)

Descrição

Diagramas de tempo são os meios pelos quais se representam as seqüências de eventos lógicos, disparados por mudanças de níveis dos sinais de um microprocessador.

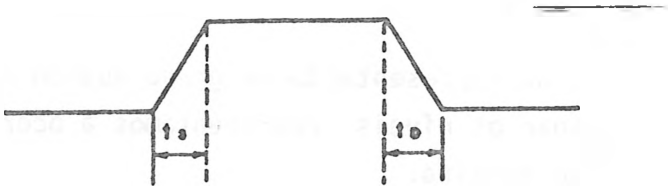
Estrutura

Cada sinal é representado em um gráfico nível x tempo, mostrando os períodos de ocorrência das transições.



Comentário

Normalmente as transições de um nível para outro são ilustradas como instantâneas, embora estas transições ocorram dentro de um limite de tempo finito e mensurável.



Ts: tempo de subida

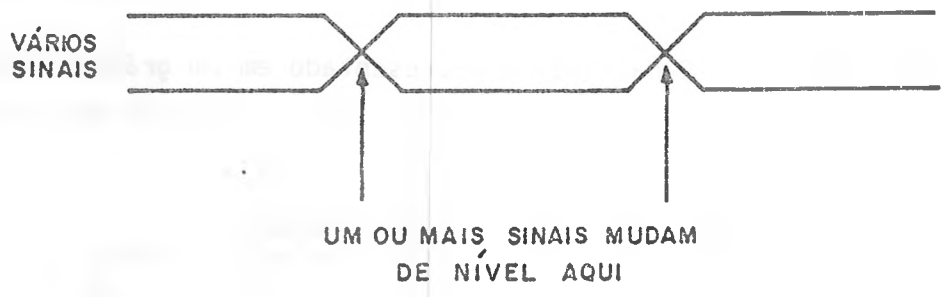
TD: tempo de descida

As transições são chamadas bordas do sinal e são de dois tipos:

. Borda de subida

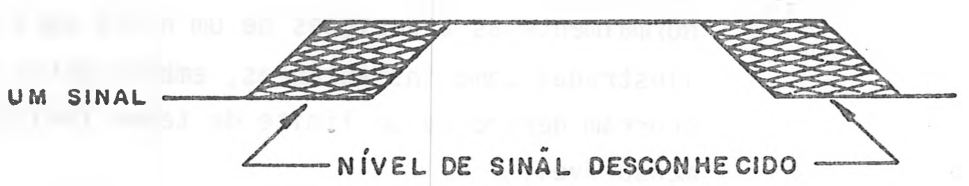
. Borda de descida

Muitas vezes representamos sinais em grupos, conforme figura abaixo.

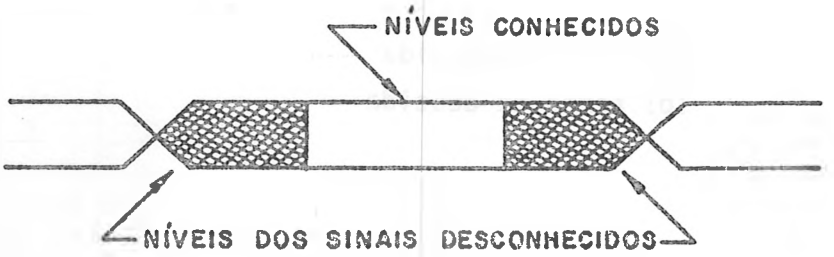


Existem ocasiões em que um sinal pode estar tanto em nível lógico baixo como em nível lógico alto, mas não existe maneira de determinar isso.

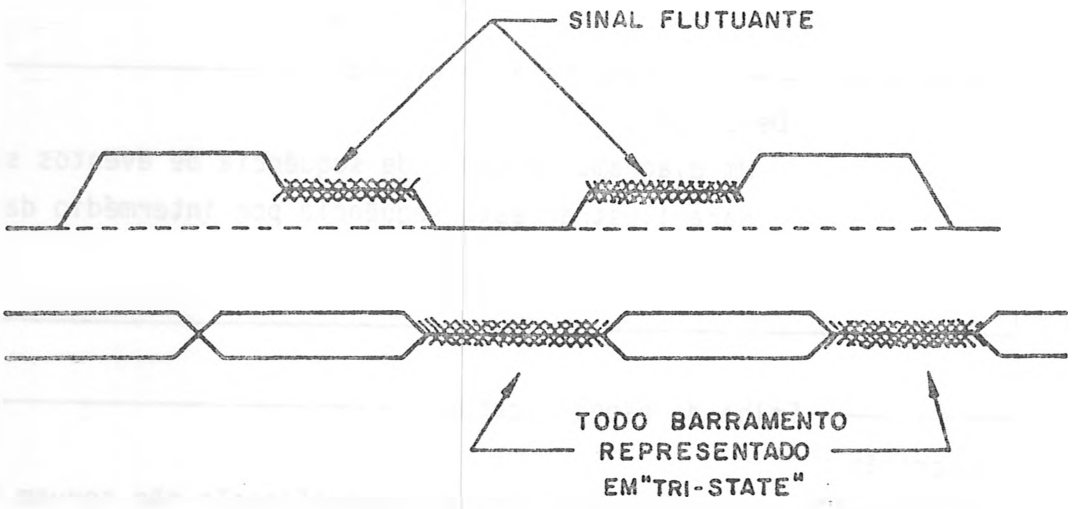
Representamos essa ocorrência da seguinte forma:



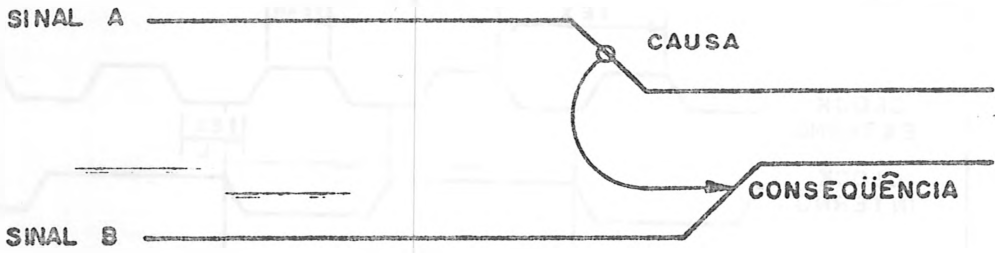
No caso da representação em grupo quando não é possível determinar os níveis, representamos a ocorrência da seguinte maneira:



Também é possível que um sinal esteja desligado ou em alta impedância. Neste caso é chamado sinal flutuante ou em tri-state e é representado conforme figura a seguir.



Representamos a relação entre causa e conseqüência de um sinal da seguinte forma:



Um círculo representa a causa da transição que ocorre em outro sinal.
Uma seta indica uma conseqüência relativa a uma transição ou alteração do nível de outro sinal.
Uma conseqüência é sempre uma transição e pode ter várias causas. A causa pode ser um estado lógico ou uma transição.

Classificação Existem dois tipos de diagramas de tempo:

. Seqüência de eventos

. Folha de especificação

Seqüência de eventos

Descrição

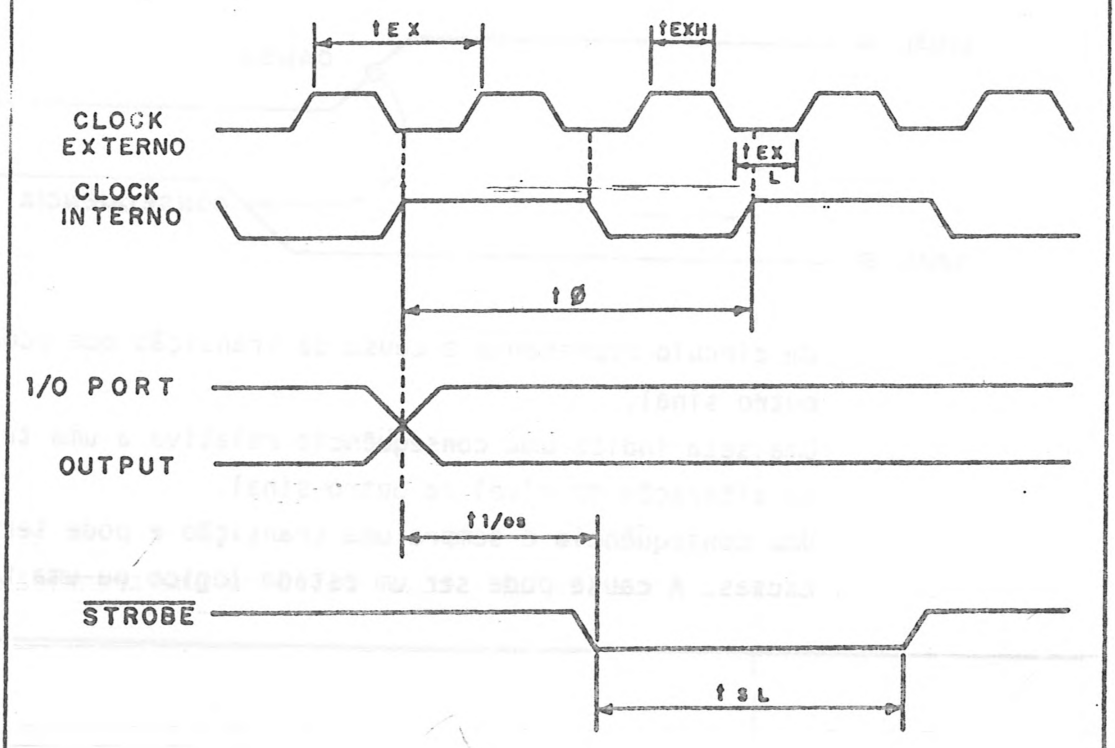
Um diagrama de tempo de seqüência de eventos serve para ilustrar esta seqüência por intermédio das transições e níveis de sinais.

Folha de especificação

Descrição

Os diagramas de tempo de folha de especificação não servem para especificar seqüências de eventos, mas eles representam com bastante precisão os intervalos de tempo e atrasos.

Exemplo



Carta de tempo de folha de especificação

Tabela-guia

Sinal	Símbolo	Parâmetro	Min.	Máx.	Unid.
Clock externo	t_{EX}	Período total	250	1000	ns
	t_{EXH}	Tempo de nível alto	90	700	ns
	t_{EXL}	Tempo de nível baixo	100	700	ns
Clock interno	t_O	Período	$2.t_{EX}$		
STROBE	$t_{1/OS}$	Atraso para validar a saída	3t0 -1000	3t0 +250	ns
	T_{SL}	Tempo de nível válido	8t0 -250	12t0 +250	

O exemplo citado mostra quatro sinais:

. Dois sinais de clock

. Um sinal de controle chamado strobe

. Várias linhas paralelas chamadas I/O INPUT OUTPUT

Como ilustrado, estes sinais não explicitam seqüências de eventos. Em vez disso, são mostrados intervalos de tempo de atrasos, determinados conforme tabela-guia.

Palavras-chave

DIAGRAMAS DE TEMPO

SINAIS DE TEMPORIZAÇÃO DE UM MICROPROCESSADOR — TE 60240

(2 páginas)

Sinônimo — Clock de um microprocessador

Descrição — As operações dentro de um microprocessador são controladas por um sinal de relógio cujo período pode variar de um mínimo de 100 nanosegundos até um máximo de um microsegundo.

O período de clock é uma característica inerente a cada microprocessador, dependendo da tecnologia empregada na concepção do projeto.

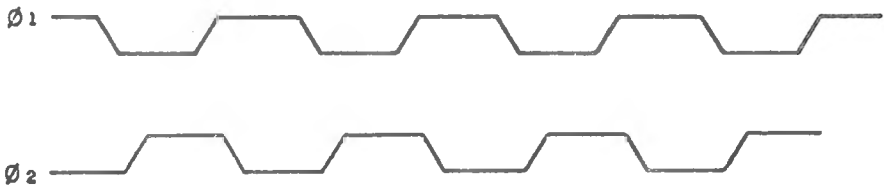
Classificação — Dependendo de como o microprocessador foi projetado, o sinal de clock poderá ser:

. um sinal simples e quadrado;

. um sinal constituído por interações mais complexas de sinais quadrados.

Exemplo

Interação de sinais



Estrutura

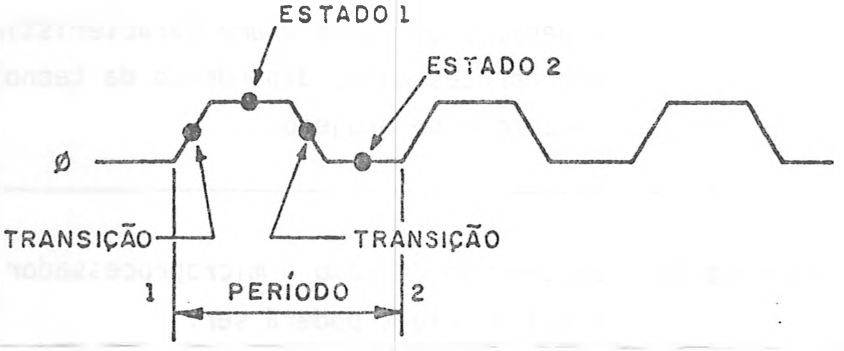
A estrutura dos sinais de temporização apresenta-se sob as formas de:

. Sinal simples

. Sinal complexo

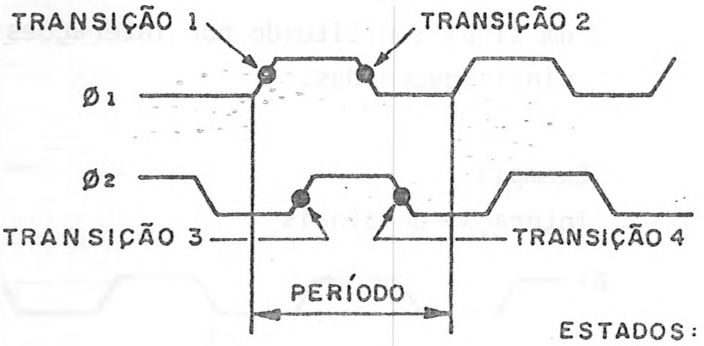
Sinal simples

Possui duas transições e dois estados por período.



Sinal complexo

Pode possuir quatro transições e quatro estados por período.



ESTADOS: 1: Ø1 = 1, Ø2 = 0
2: Ø1 = 1, Ø2 = 1
3: Ø1 = 0, Ø2 = 1
4: Ø1 = 0, Ø2 = 0

Palavras-chave

TEMPORIZAÇÃO
CLOCK DE UM MICROPROCESSADOR

TEMPORIZAÇÃO DE INSTRUÇÕES (Timing)

TE 60280 (5 páginas)

Descrição

Cada instrução de um microprocessador é executada por meio da combinação de algumas operações básicas do tipo:

. Leitura ou escrita na memória

. Leitura ou escrita em dispositivos de entrada/saída

. Atendimento a interrupções

Cada uma destas operações demora um determinado número de períodos de clock para ser completada. Os conjuntos destes períodos de clock referentes à execução das operações são denominados ciclos de máquina.

Estes ciclos de máquinas podem ser estendidos para poderem sincronizar as operações da UCP com os dispositivos externos mais lentos.

Sendo assim, cada instrução é processada pelo microprocessador em um ciclo denominado ciclo de instrução, o qual pode ser constituído de um número de ciclos de máquina, dependendo do tipo de instrução e da concepção básica do seu projeto.

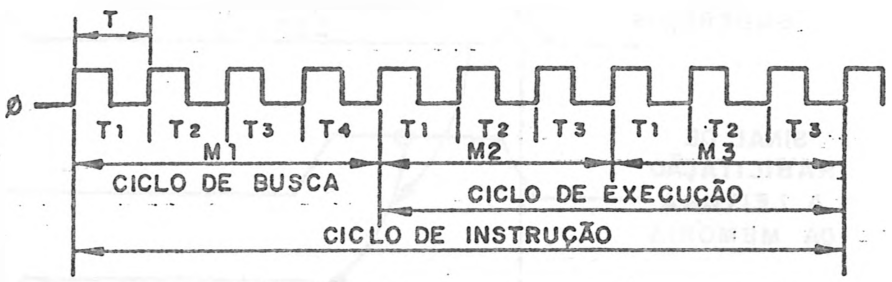
Estrutura

Cada ciclo de instrução possui duas fases distintas:

. Primeira fase ou **ciclo de busca** do código de operação da instrução

. Segunda fase ou **ciclo de execução** da instrução

A figura a seguir representa melhor esta idéia.



T = um período de clock

M1, M2, M3 = ciclos de máquina

Ø = clock do microprocessador

Processo

Ciclo de busca

Durante um ciclo de busca de instrução, a lógica da UCP envia o conteúdo do registro contador de programa juntamente com sinais de controle apropriado, o que faz a lógica externa retornar o conteúdo da palavra de memória endereçada pelo contador de programas.

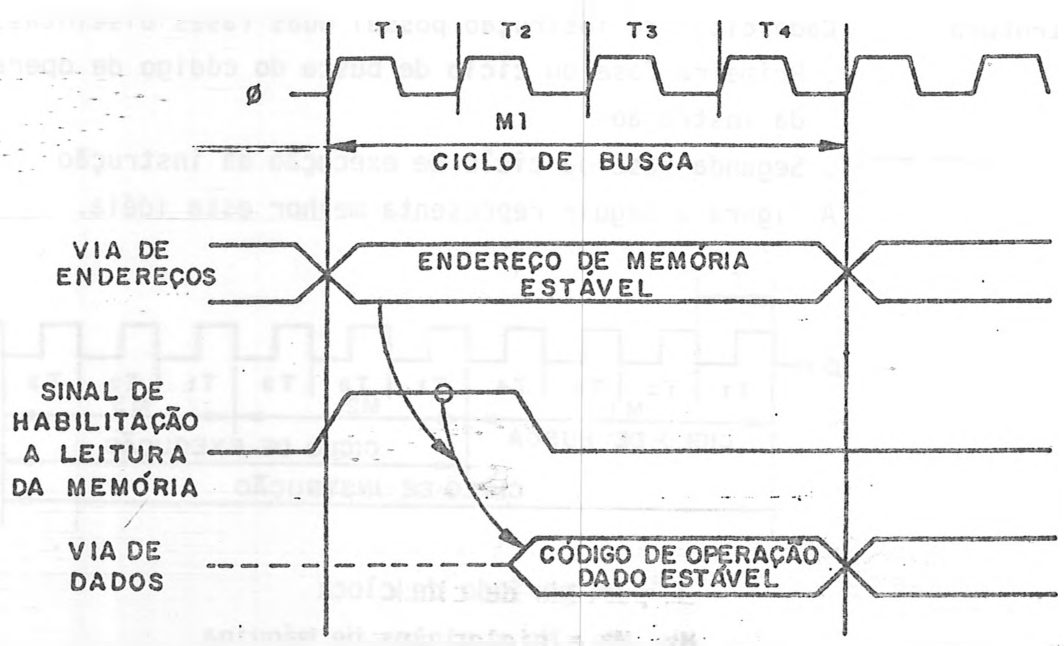
Para a lógica externa, esta operação é encarada como uma simples operação de leitura de memória.

O conteúdo desta palavra de memória é então armazenado no registro de instruções e logo é interpretado como código de operação da instrução.

Enquanto a lógica externa está respondendo à busca de instrução, a UCP utiliza-se de sua lógica interna para incrementar de 1 o conteúdo de seu contador de programa, fazendo com que este aponte para a posição seguinte de memória de programa.

Ilustração

Diagrama de tempos dos sinais nos barramentos da lógica externa



Ciclo de execução

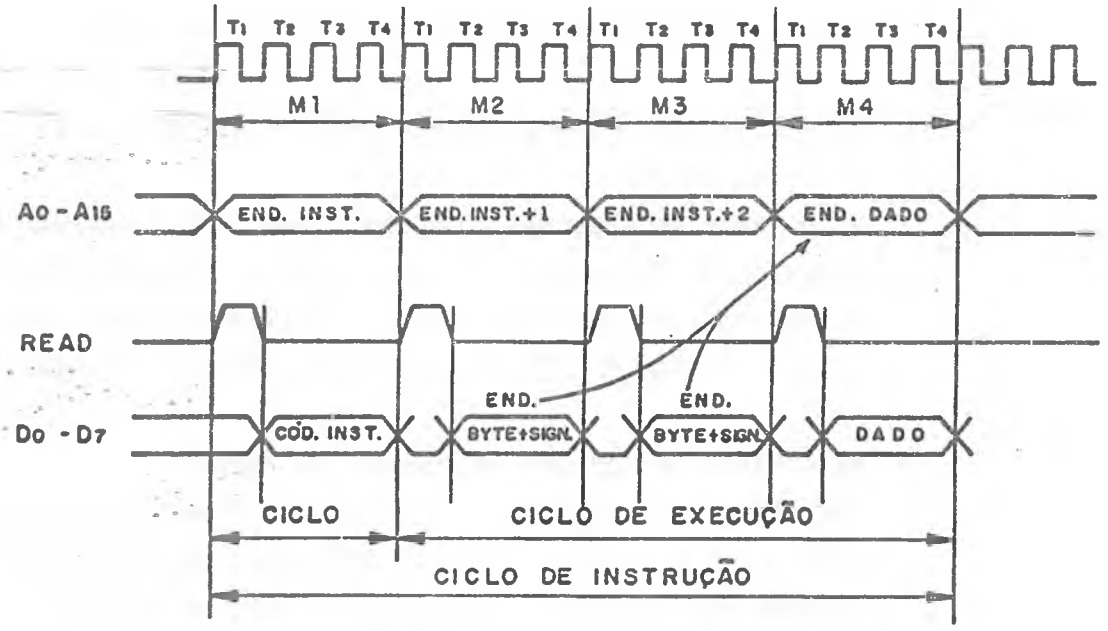
Uma vez que o código da instrução está no registro de instruções, ele dispara uma seqüência de eventos controlados pela unidade de controle. Esta seqüência de eventos constitui a execução da instrução.

Exemplo

Ciclo de uma instrução de leitura de memória através de endereçamento direto

Suposições

. Microprocessador de oito bits de palavra de dados
. Capacidade de endereçamento direto 65.535 x 8 bits, palavra de endereço de 16 bits



M1 - ciclo de busca do código de operação
O conteúdo do PC é colocado no barramento de endereços. Então será lida a posição de memória endereçada pelo mesmo, sendo o conteúdo desta locação decodificado pelo registro de instrução. Após a decodificação, verifica-se que se trata de uma instrução de leitura à memória por endereçamento direto e que esta apresenta mais dois bytes, os quais formam, juntos, o endereço da posição de memória a ser acessada.

M2 - ciclo de leitura do byte menos significativo de endereços
O contador de programa é incrementado, colocando seu conteúdo no barramento de endereço. É efetuado outro ciclo de leitura cujo byte lido é armazenado no byte menos significativo do registro contador de dados.

M3 - ciclo de leitura do byte mais significativo de endereços
O contador de programa é incrementado e seu conteúdo é colocado no barramento de endereços. É então efetuado outro ciclo de leitura cujo byte acessado é armazenado no byte mais significativo do registro contador de dados.

M4 - ciclo de leitura de memória de dados
O contador de programa é novamente incrementado para se preparar para o ciclo seguinte de busca de instrução.
O endereço contido no registro contador de dados é colocado no barramento de endereços.
Então procede-se a um novo ciclo de leitura, cujo conteúdo da posição de memória acessada é colocado no acumulador, encerrando-se o ciclo de instrução.

O ciclo seguinte da memória corresponderá a um novo ciclo de busca de código de operação da instrução a ser executada.

Comentário Não existe, realmente, nenhuma forma padrão para que um microprocessador temporize ou execute instruções. A forma apresentada constitui a preferência da maior parte dos fabricantes.

Palavras-chave

TEMPORIZAÇÃO DE INSTRUÇÕES

MICROPROCESSADOR Z-80 - ARQUITETURA

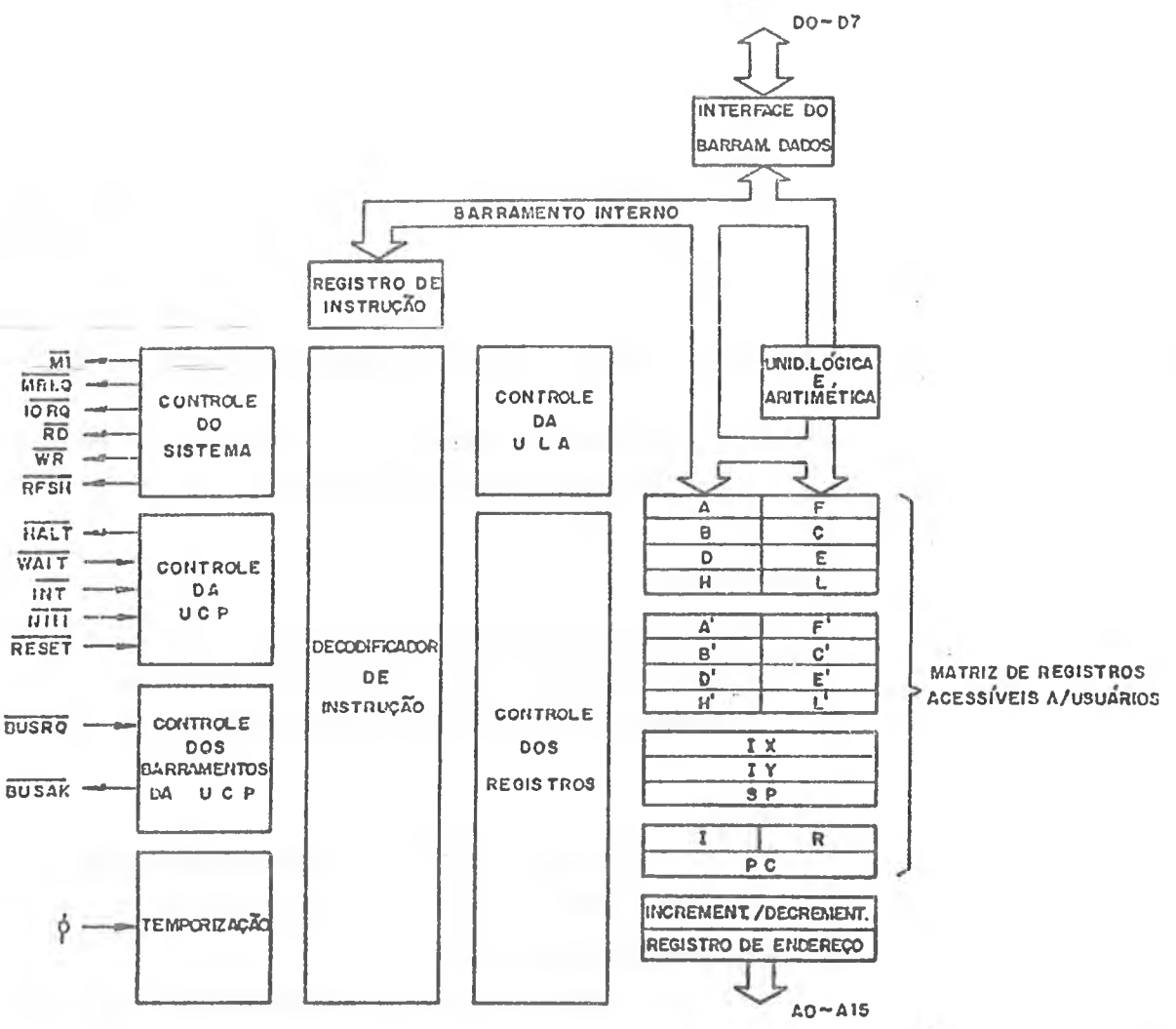
TA 30870
(9 páginas)

Descrição

O Z-80 é um microprocessador de oito bits, com capacidade de endereçamento direto de 65536 bytes de memória comportando, portanto, 16 linhas de endereçamento.

Estrutura

Arquitetura da UCP Z-80



Interface do barramento de dados

Descrição

É uma interface bidirecional que isola eletricamente o barramento interno de dados do externo.

Processo

Numa operação de saída de dados, o conteúdo do barramento interno é carregado num registro de oito bits (latch), cujas saídas vão para um buffer de saída. Quando não está ocorrendo esta operação, este buffer permanece com suas saídas em alta impedância.

Numa operação de entrada de dados, o conteúdo do barramento externo é transferido para o barramento interno.

Registro de endereço e incrementador/decrementador

Descrição

Trata-se de um registro de 16 bits, cujas saídas vão para circuitos buffers de saída, para acionamento do barramento de endereços (A_0 - A_{15}).

Processo

O registro recebe endereços provenientes dos registros concatenados de 16 bits ou do incrementador/decrementador.

O circuito incrementador/decrementador recebe os dados do registro de endereço, envia-os para os registros de 16 bits ou devolve-os ao registro de endereços.

Registro de instrução, decodificador de instrução, controles

Descrição

Trata-se de um registro de bits, onde são armazenados os códigos de instrução.

Processo

Durante uma operação de leitura de instrução, os dados lidos na memória do programa são transferidos para o registro de instrução. O conteúdo do registro de instrução é decodificado, de modo que as saídas do decodificador de instrução, combinadas com sinais de temporização, gerem os sinais de controle para a matriz de registros, para a ULA, para a interface do barramento de dados e para os circuitos do barramento de controle.

Os circuitos do barramento de controle combinam os sinais de suas entradas com os do decodificador de instrução e geram os sinais de controle, que vão para os buffers de saída para acionamento do barramento de controle.

TA 30870

Unidade lógica aritmética

Descrição

A ULA do Z-80 tem capacidade para manipular oito bits de cada vez. As operações com 16 bits são efetuadas em duas etapas. As operações lógicas e aritméticas são efetuadas na ULA, sendo que esta se comunica internamente com os registros da UCP e não é diretamente acessável por programa.

Processo

Em geral as operações lógicas e aritméticas são efetuadas entre o acumulador e outro registro, com o resultado sendo armazenado no acumulador. O registro de flags é afetado conforme o resultado da operação.

Registros

Descrição

O Z-80 é um microprocessador do tipo register-oriented, ou seja, que possui muitos registros e muitas instruções para manipulação de seus registros internos. Outras UCPs tipo memory-oriented possuem menos registros e maiores facilidades para endereçamento da memória.

Há 18 registros de oito bits e quatro de 16 bits, acessíveis por programa, que funcionam como uma memória tipo RAM.

Alguns dos registros de oito bits podem ser agrupados aos pares para serem usados como registros de 16 bits.

A configuração de registro é mostrada a seguir.

Conjunto principal

Acumulador A	Flags F
B	C
D	E
H	L

uso geral

Conjunto alternativo

Acumulador A'	Flags F'
B'	C'
D'	E'
H'	L'

Vetor de interrupção	I
Refresh de memória	R
Registro índice	Ix
Registro índice	Iy
Ponteiro de pilha	SP
Contador de programa	PC

Uso especial

Processo

Há um grupo principal de registros de oito bits e um outro alternativo, com o objetivo de oferecer ao programador mais registros.

Não é possível utilizar ambos os grupos simultaneamente, pois não há instruções para operações nos registros alternativos.

Para acessá-los, é necessário antes comandar uma troca de funções, de modo que o conjunto principal passe a ser o alternativo e vice-versa.

Posteriormente, nova instrução de troca poderá ser efetuada, de tal forma que retorne ao estado original.

Acumuladores e flags

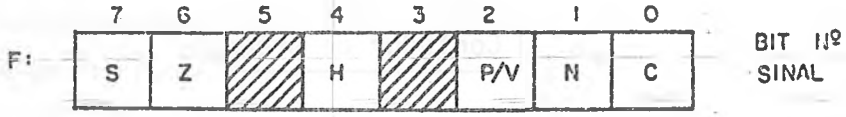
Descrição

Há dois pares de registros acumuladores e flags. O primeiro encontra-se no conjunto principal (A e F) e o outro no alternativo (A' e F').

Processo

O acumulador recebe os resultados de todas as operações lógicas e aritméticas de oito bits. O registro de flags indica a ocorrência de condições específicas decorrentes das operações lógicas e aritméticas.

Registro de flags



S: Flag de sinal
Z: Flag zero
H: Half carry ou vai um intermediário
P/V: Paridade/overflow
N: Soma/subtração
C: Carry ou vai um

Comentário

. O registro de flag também é chamado registro de status.
. O estado dos bits é afetado, dependendo da operação executada.

Registros de uso geral

Descrição

Há dois grupos de registros de uso geral, um no conjunto principal (A, B, C, D, E, H, L) e outro no alternativo (B', C', D', E', H', L').
Para operações de 16 bits, estes registros podem ser agrupados em pares (BC, DE, HL, BC', DE', HL').

Contador de programa

Descrição

O contador de programa faz parte do grupo de registradores especiais e possui 16 bits.

Processo

O CP contém um endereço de 16 bits que indica a posição de memória onde se encontra o código da instrução em execução.
Após a execução da instrução, o CP é incrementado (se o programa deve prosseguir no endereço seguinte) ou substituído por novo valor (se a instrução executada é de desvio de seqüência, tipo salto para outro endereço, chamada ou retorno à sub-rotina).

Ponteiro de pilha (stack-pointer SP)

Descrição

O SP é um registro de 16 bits que permite a implementação de pilha de dados na memória.

Processo

Uma pilha é uma área reservada na memória onde as inclusões e as retiradas de dados são do tipo os últimos a entrarem serão os primeiros a saírem (LIFO).

O programador não precisa preocupar-se com o endereço físico onde armazenar dados, pois basta retirá-los na ordem inversa da entrada para que sejam recuperados corretamente.

A pilha é utilizada para armazenamento de dados de 16 bits, provenientes de registros de oito bits; a retirada de dados também é feita em grupos de 16 bits armazenados nos pares de registros internos à UCP. Os registros A e F também participam das inserções e retiradas da pilha com a designação AF.

Registros índice Ix e Iy

Descrição

São registros que facilitam a manipulação de tabelas de dados pelo uso de instruções com endereçamento em modo indexado.

Processo

O conteúdo destes registros são normalmente endereços-base (início ou fim de uma tabela, por exemplo) que, ao serem somados a um deslocamento especificado na instrução, determinarão o endereço efetivo do operando na memória.

Interrupt-vector I

Descrição

É um registro de oito bits, utilizado para armazenar a metade mais significativa do endereço na memória, onde está localizada a tabela de vetores de interrupção.

A metade menos significativa do endereço deve ser fornecida pelo dispositivo que pede a interrupção.

R - memory refresh

Descrição

É um registro de oito bits, dos quais apenas sete são utilizados.

Em conjunto com o pino RFSH, este registro pode ser utilizado para controlar o refrescamento de memórias RAM dinâmicas (DRAM).

Palavras-chave

ARQUITETURA INTERNA DA UCP Z-80

MICROPROCESSADOR Z-80 - DESCRIÇÃO DA PINAGEM TA 30880

(9 páginas)

Introdução — No desenvolvimento de projeto ou manutenção de um microcomputador que utiliza como CPU o microprocessador Z-80, é necessário conhecer a função de cada terminal.

Classificação — Os terminais da CPU Z-80 podem ser classificados em:

- Controle do sistema
- Controle da CPU
- Controle do barramento da CPU
- Barramento de endereço
- Barramento de dados
- Clock
- Alimentação

Controle do sistema

Descrição

São seis pinos de saída que informam aos demais componentes do sistema qual o tipo de operação com memória ou dispositivos de entrada/saída que a CPU está efetuando.

Classificação

Os terminais de controle do sistema são:

- $\overline{M1}$
- $\overline{MREQ}$
- $\overline{IORQ}$
- $\overline{RD}$
- $\overline{WR}$
- $\overline{RFSH}$

$\overline{M1}$

Tipo

Saída, ativo baixo.

Descrição

Indica que a CPU está efetuando o ciclo de busca de um código de operação (op code fetch).

$\overline{MREQ}$

Tipo

Saída três estados, ativo baixo.

Descrição

Indica que no barramento de endereços há um endereço válido para uma operação de leitura ou escrita na memória.

Comentário

Como as instruções devem estar na memória, $\overline{MREQ}$ também é válido durante o ciclo de captura do código de operação.

$\overline{IORQ}$

Tipo

Saída três estados, ativo baixo.

Descrição

Indica que na metade inferior do barramento de endereços há um endereço válido para uma operação de leitura ou escrita num dispositivo de E/S.

Comentário

Durante o ciclo de atendimento de interrupção, $\overline{IORQ}$ é validado juntamente com $\overline{M1}$, para sinalizar que a CPU está efetuando a captura de um vetor de oito bits no barramento de dados, o qual deve ser fornecido pelo dispositivo que pediu a interrupção.

$\overline{RD}$

Tipo

Saída três estados, ativo baixo.

Descrição

Indica que a CPU deseja ler dados da memória ou dispositivo de E/S endereçado.

$\overline{WR}$

Tipo

Saída três estados, ativo baixo.

Descrição

Indica que a CPU colocou no barramento de dados um byte que deve ser gravado na memória ou dispositivo de E/S endereçado.

$\overline{RFSH}$

Tipo

Saída, ativo baixo.

Descrição

Indica que os sete bits menos significativos de barramento de endereços contêm um endereço para reciclagem de memória dinâmica e que o sinal $\overline{MREQ}$, validado juntamente com $\overline{RFSH}$, pode ser usado para uma leitura de reciclagem (refresh) de memória dinâmica.

Comentário

O bit A_7 de barramento de endereço indica zero e de A_8 a A_{15} o conteúdo do registro I.

Controle da CPU

Descrição

São quatro pinos de entrada que permitem aos dispositivos exteriores influenciar o funcionamento da CPU, e um pino de saída para sinalizar ao exterior um estado especial da CPU.

Classificação

Os terminais de controle da CPU são:

. $\overline{\text{HALT}}$
. $\overline{\text{WAIT}}$
. $\overline{\text{INT}}$
. $\overline{\text{RESET}}$

$\overline{\text{HALT}}$

Tipo

Saída, ativo baixo.

Descrição

Indica que a CPU executou uma instrução de parada (HALT) e permanecerá neste estado, enquanto não ocorrer uma interrupção ou um reset.

Comentário

Equanto em HALT,a CPU permanece executando instruções NOP, para manter a atividade de reciclagem de memória dinâmica.

$\overline{WAIT}$

Tipo

Entrada, ativo baixo.

Descrição

Indica para a CPU que a memória, ou dispositivo de E/S endereçado, não está pronta para a transferência de dados.

Comentário

Componentes de qualquer velocidade podem ser sincronizados com a CPU, bastando existir um circuito que gere $\overline{WAIT}$ enquanto não estiverem prontos para a operação.

$\overline{INT}$

Tipo

Entrada, ativo baixo.

Descrição

É um sinal gerado por dispositivo de E/S para interromper a seqüência do programa em execução, sendo que o pedido será atendido no fim da instrução em curso se a CPU estiver habilitada a aceitar interrupção e se o sinal $\overline{BUSRQ}$ não estiver ativo.

Comentário

Quando a CPU atende à interrupção, executa uma operação de reconhecimento de interrupção, identificada pela ativação de $\overline{IORQ}$ e M_1.

$\overline{\text{RESET}}$

Tipo

Entrada, ativo baixo.

Descrição

Força o zeramento do registro PC (contador de programa) e inicializa a CPU.

Comentário

Durante o $\overline{\text{RESET}}$ os barramentos de endereços e dados vão para o estado de alta impedância e os sinais de controle vão para o estado inativo, não ocorrendo $\overline{\text{RFSH}}$.

No momento do $\overline{\text{RESET}}$, a CPU é inicializada, obedecendo ao seguinte procedimento:

1º passo: Zera o PC.

2º passo: Inibe atendimento de interrupção.

3º passo: Registra I igual 00.

4º passo: Registra R igual 00.

5º passo: Estabelece o modo 0 de interrupção, ou seja, o dado fornecido pelo dispositivo é aceito como uma instrução que a CPU executa normalmente. O mais usual é que esta instrução seja uma chamada de sub-rotina.

Controle do barramento da CPU

Descrição

São dois pinos: um tipo entrada para pedido de liberação dos barramentos e outro tipo saída para indicar que a CPU atendeu ao pedido.

Classificação

Os terminais de controle do barramento da CPU são:

. $\overline{\text{BUSRQ}}$

. $\overline{\text{BUSAK}}$

$\overline{BUSRQ}$

Tipo

Entrada, ativo baixo.

Descrição

É usado para pedir que a CPU coloque em estado de alta impedância seus barramentos de endereços e de dados e as saídas dos sinais de controle, a fim de que outro dispositivo possa controlar estes barramentos.

Comentário

Quando $\overline{BUSRQ}$ é ativado, a CPU libera estes barramentos assim que conclui o ciclo de máquina em curso.

$\overline{BUSAK}$

Tipo

Saída, ativo baixo.

Descrição

É emitido pela CPU para sinalizar que os barramentos de endereços e dados, e os sinais de controle, foram colocados em alta impedância e que o dispositivo externo já pode ativar os barramentos.

Comentário

. Enquanto a CPU estiver em WAIT ou BUSAK, não há geração de $\overline{RFSH}$ para a memória.
. $\overline{BUSRQ}$ é atendido no final do ciclo de máquina em curso.
. $\overline{INT}$ e $\overline{NMI}$ são atendidos no final da instrução em curso.
. Quando em BUSAK, a CPU não atende a $\overline{INT}$ ou $\overline{NMI}$.
. A ordem de prioridade é
  1º - $\overline{BUSRQ}$
  2º - $\overline{NMI}$
  3º - $\overline{INT}$

Barramento de endereço

Tipo

Saída três estados, ativo alto.

Descrição

São os pinos de A_0 a A_{15}, que fornecem endereços para a troca de dados entre a CPU e a memória (até 64k bytes) ou dispositivos de entrada/saída (até 256 entradas e 256 saídas).

Comentário

. Durante operações de endereçamento de E/S somente os oito bits menos significativos de A_0 a A_7 contêm o endereço válido. Nos oito bits mais significativos aparece o conteúdo do acumulador.

. Durante o refresh nos sete bits menos significativos de A_0 a A_6 aparece o conteúdo do registro R, que é um endereço válido para operação de reciclagem de memórias dinâmicas.

Barramento de dados

Tipo

Entrada/saída três estados, ativo alto.

Descrição

Constituído de D_0 a D_7, é um barramento bidirecional de dados, utilizado para as trocas de dados entre a CPU e a memória ou periféricos de E/S.

Clock

Descrição

É o pino (ø) pelo qual é injetada a freqüência de clock da CPU, gerada num oscilador externo.

Procedimento

Na geração da freqüência de clock devem ser observados alguns procedimentos:

1. Tempos de subida e descida inferiores a um limite máximo.
2. Duração do sinal em níveis baixo e alto dentro de limites máximos e mínimos.
3. Em decorrência dos itens 1 e 2, a freqüência do sinal deve estar dentro de limites máximos e mínimos.
4. Nível de tensão do sinal em níveis baixo e alto dentro de limites máximos e mínimos.

Alimentação

Descrição

A alimentação do Z-80 é aplicada através dos pinos +5V e GND e deve ser mantida na faixa de 5V ± 5%.

Palavras-chave

MICROPROCESSADOR Z-80

MICROPROCESSADOR Z-80 - TIPOS DE INSTRUÇÃO TA 30890

(6 páginas)

Descrição — Instruções são ordens dadas ao microprocessador para execução de uma tarefa, que pode ser de ler ou escrever numa posição de memória, fazer operações aritméticas e lógicas, transferir blocos, etc.

Classificação — O conjunto de instruções do microprocessador Z-80 é de 158, sendo que estas podem ser divididas, de acordo com a função que executam, da seguinte maneira:

. Carga e troca de informações
. Aritmética e lógica
. Transferência de blocos
. Rotação e deslocamento
. Manipulação de bits
. Salto, chamadas e retornos
. Entrada e saída
. Controle interno

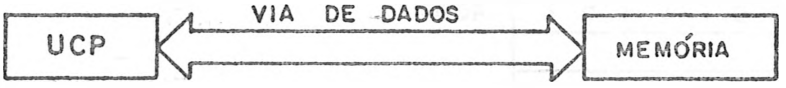
Carga e troca de informações

Descrição

A carga e troca de informações consiste em manusear dados entre a UCP e a memória e vice-versa ou entre os próprios registros internos.

Ilustração

Transferência de dados entre a UCP e a memória



Transferência de dados entre os registros internos

TA 30890

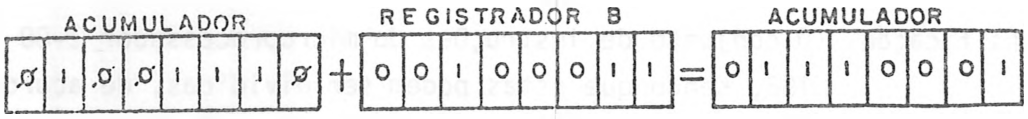
Operações aritméticas e lógicas

Descrição

As operações aritméticas e lógicas consistem na realização de soma, subtração e funções lógicas entre o registrador e a memória.

Ilustração

Operação aritmética



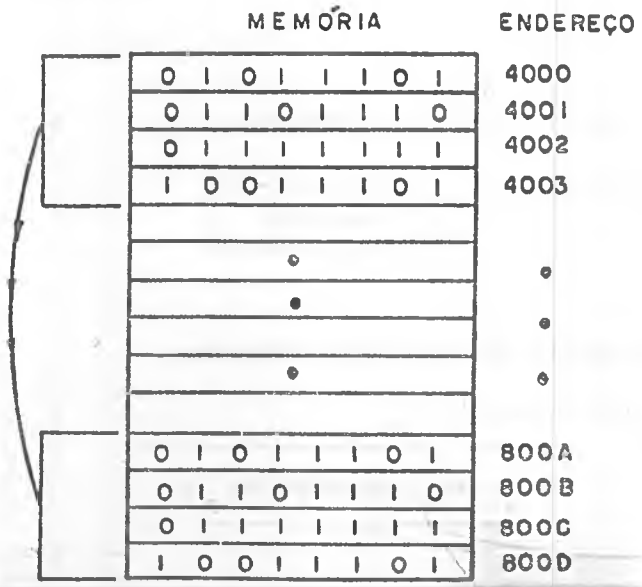
Operação lógica

Transferência de blocos

Descrição

A transferência de blocos é um grupo de instruções que o microprocessador Z-80 apresenta que possibilita copiar ou transferir um conjunto de bytes de uma determinada área da memória para uma outra dada posição da mesma.

Ilustração



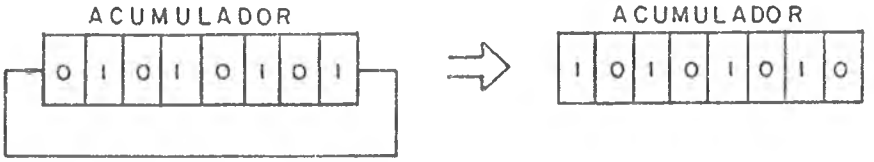
Rotação e deslocamento

Descrição

As instruções de rotação e deslocamento são utilizadas para deslocar uma informação (bits) para esquerda ou direita.

Ilustração

Rotação de um bit para a direita



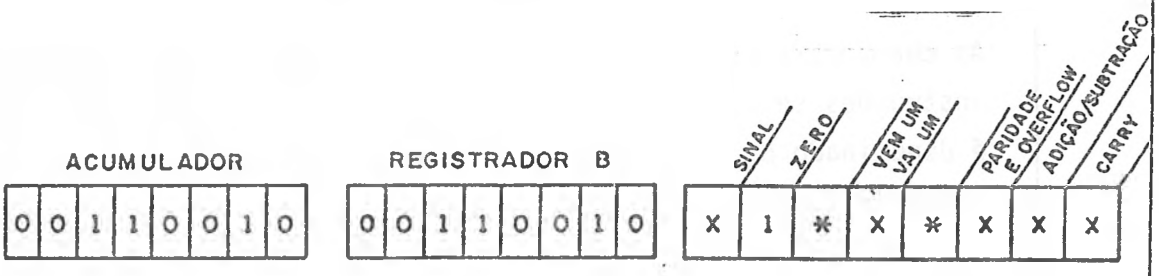
Manipulação de bits

Descrição

A manipulação de bits é conseguida através de um grupo de instruções do Z-80 que compara bytes e por sua vez alteram os flags.

Ilustração

A comparação entre o acumulador e o registro C leva a flag de zero a nível 1, se os conteúdos dos mesmos forem iguais.



TA 30890

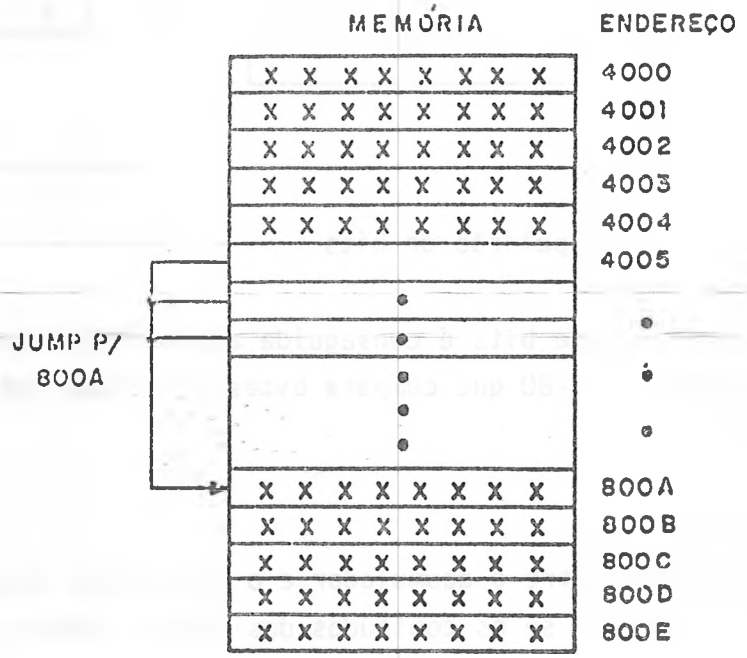
Saltos, chamadas e retornos

Descrição

As instruções de saltos, chamadas e retornos são usadas para mudar o contador de programa (PC) e seguir num outro endereço predeterminado.

Exemplo

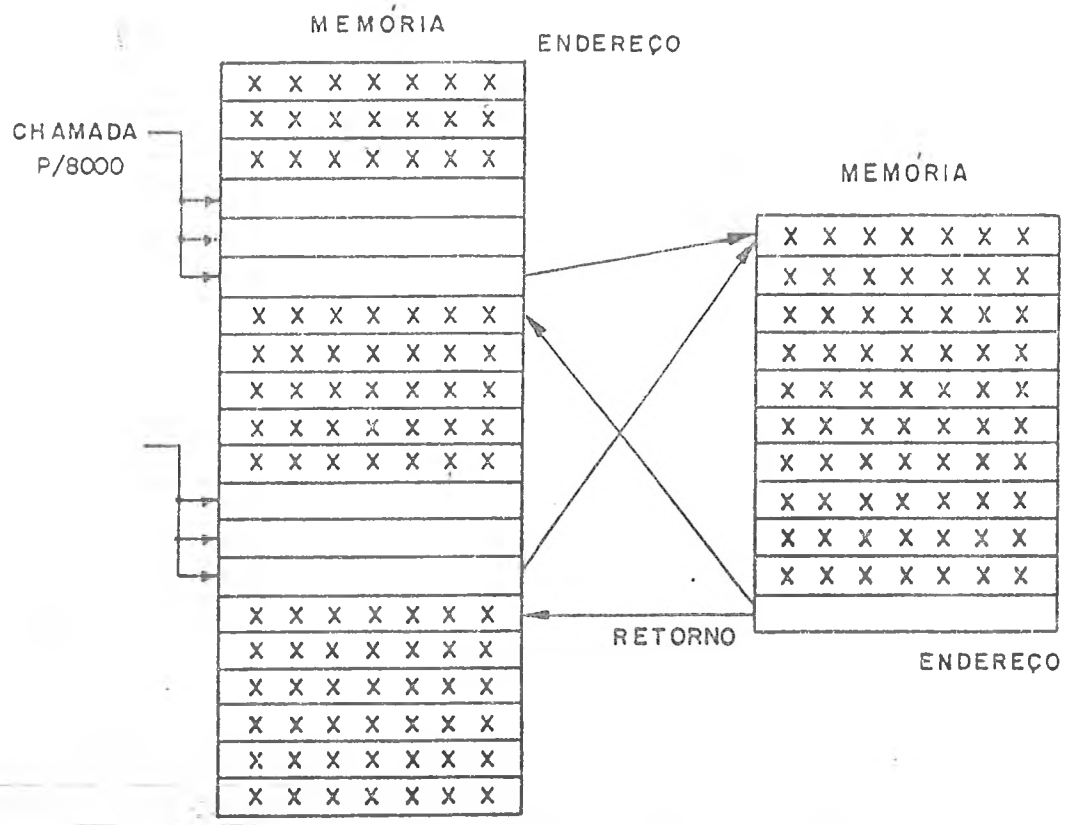
Os saltos (jumps) são utilizados para mudar a seqüência de um determinado programa.



As chamadas e retornos são utilizadas quando um certo grupo de instruções se repete várias vezes no mesmo programa. Este grupo é denominado de sub-rotina e em seu final deverá haver uma instrução de retorno.

Comentário

As instruções de salto ocupam três posições de memória.



Comentário

A instrução de chamada ocupa três posições de memória. A instrução de retorno não é acompanhada do endereço e quem aponta onde o programa deve retornar é o contador de programa.

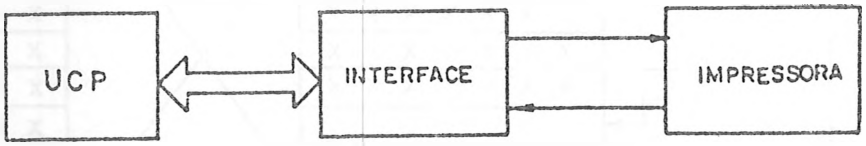
TA 30890

Entrada e saída

Descrição

As instruções de entrada e saída possibilitam que a UCP se comunique com dispositivos externos, como terminais de vídeo, impressora, drivers e outros.

Exemplo



Controle interno

Descrição

As instruções de controle interno são usadas para liberação de interrupção mascarada, para designar modo de interrupção mascarada e para parada geral da UCP.

Palavras-chave

TIPOS DE INSTRUÇÃO DO Z-80
GRUPO BÁSICO DE INSTRUÇÕES DO Z-80

Descrição

Sistema de interrupção é o processo pelo qual um dispositivo periférico gera um pedido de interrupção fazendo com que a CPU suspenda sua operação normal a fim de atender ao pedido através de uma rotina previamente programada. Ao término desta rotina, a CPU volta a executar as operações que executava antes da interrupção.

Classificação

Os modos de interrupção classificam-se em:
. Interrupção não mascarável
. Interrupção mascarável

Interrupção não mascarável (entrada $\overline{NMI}$)

Descrição

É uma entrada de interrupção que tem prioridade sobre as demais e é aceita a qualquer instante, não importando as condições do programa.

Processo

Uma vez ativada a entrada $\overline{NMI}$, a CPU executa uma instrução restart (RST 66), que corresponde a uma chamada de sub-rotina com início no endereço 0066H.

Comentário

. A entrada $\overline{NMI}$ é sensível à borda de descida do sinal aplicado, para evitar que a manutenção do nível baixo em $\overline{NMI}$ gere novas interrupções durante o atendimento do pedido inicial de $\overline{NMI}$.
. Devido à sua maior prioridade, quando o NMI for acionado a CPU não aceitará $\overline{INT}$.
. A instrução final da rotina de $\overline{NMI}$ deve ser um return from non-maskable interrupt (RETN) que restaura a condição de mascaramento de $\overline{INT}$.

Interrupção mascarável (entrada INT)

Descrição

É uma entrada de interrupção, que pode ou não aceitá-la, dependendo da programação de sua máscara interna.

Classificação

A interrupção mascarável divide-se em:

. Modo 0
. Modo 1
. Modo 2

Modo 0

Processo

A interrupção causa um salto no programa, cujo endereço de destino é dado conforme o código colocado no Data Bus durante o tempo em que $\overline{IORQ}$ e $\overline{M1}$ estão ativados.

Comentário

A correlação entre os códigos e endereços é dada a seguir.

Código	Endereços
C7H	0000 H
CFH	0008 H
D7H	0010 H
DFH	0018 H
E7H	0020 H
EFH	0028 H
F7H	0030 H
FFH	0038 H

Modo 1

Processo

Um sinal na linha $\overline{INT}$ faz com que o programa salte para a posição 38 H

Modo 2

Processo

O programa salta para um endereço da memória. Os oito bits mais significativos são dados pelo registro de interrupção I, e a parte menos significativa pelo vetor interrupção do dispositivo periférico, sendo que apenas sete bits são necessários, ficando o bit menos significativo sempre em zero.

Ilustração

Registro I	Periférico Ø

Comentário

O modo pelo qual a CPU irá atender uma interrupção é programado por Software.

Modo	Instrução
0	IM_0
1	IM_1
2	IM_2

Palavras-chave

MICROPROCESSADOR Z-80 - SISTEMA DE INTERRUPÇÃO

MICROPROCESSADOR Z-80 - CICLO DE TEMPO TA 30920
(12 páginas)

Descrição — Os ciclos de tempo do microprocessador Z-80 são os tempos gastos por ele para ler/escrever na memória, ler/escrever em dispositivos de entrada/saída, etc.

Classificação — Os ciclos de tempo do microprocessador Z-80 classificam-se em:

. Ciclo de busca
. Ciclo de escrita e leitura de memória
. Ciclo de escrita e leitura de periféricos
. Ciclo de requisição das vias/reconhecimentos
. Ciclo de interrupção
. Resposta de interrupção não mascarada
. HALT

Ciclo de busca (Instruction fetch)

Processo

Será descrito o processo do ciclo de busca (M_1) para pino de $\overline{WAIT}$ (espera) desativado.

Passo 1 - É colocado o conteúdo do contador de programa (PC) na via de endereços (address bus).

Passo 2 - O sinal $\overline{MREQ}$ é ativado no meio ciclo do primeiro período de clock.

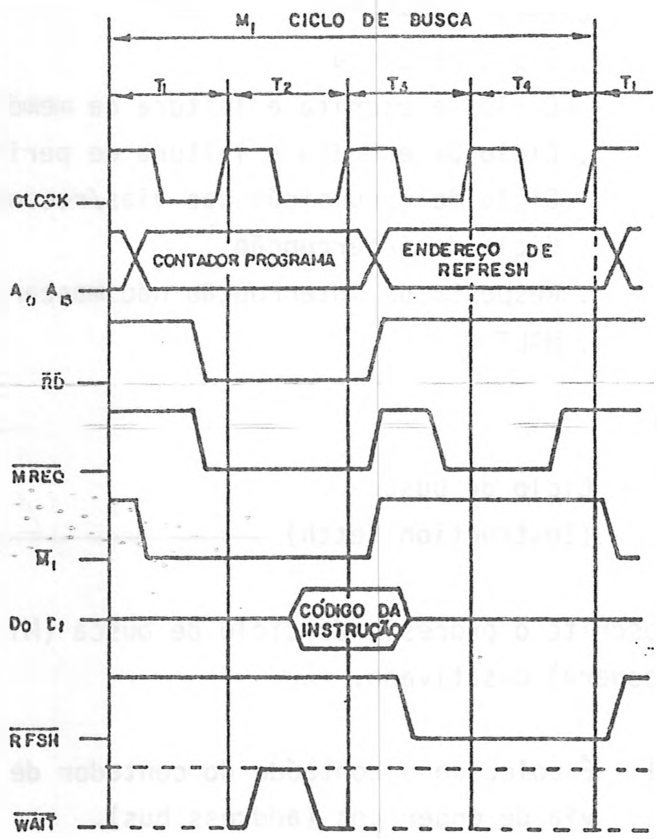
Passo 3 - O sinal $\overline{RD}$ é ativado também, significando uma operação de leitura da memória.

Passo 4 - Na subida do pulso T_3, o dado da memória é transferido para o registro de instrução.
Ainda na subida do pulso T_3 são desativados os sinais $\overline{MREQ}$ e $\overline{RD}$.

TA 30920

Passo 5 - Os períodos T3 e T4 são usados para o refresh da memória dinâmica, junto com o sinal RFSH (da subida de T3 ao fim de T4), sendo que neste período MREQ também está ativado.

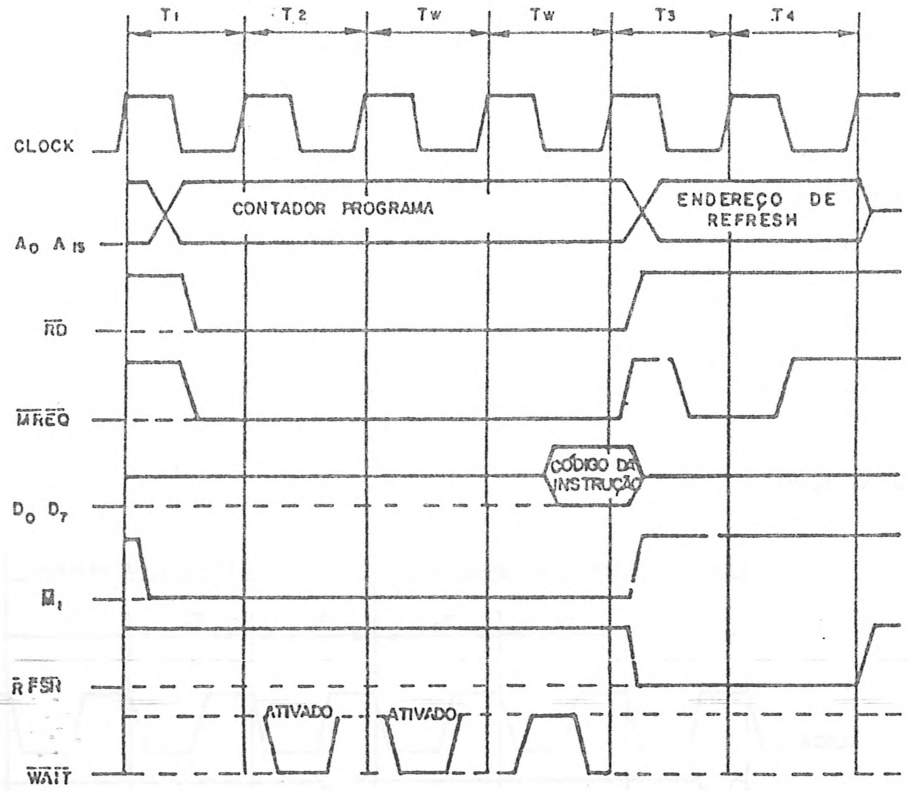
Ilustração



Comentário

Há casos em que o sinal de WAIT é ativado, pois o dispositivo com o qual a CPU se comunicará apresenta um tempo de acesso relativamente grande. Neste caso, o ciclo T em que for ativado o sinal de WAIT será repetido subseqüentemente até ser desativado, sendo que deste ponto em diante se repetirá o processo descrito.

Ilustração



Ciclo de escrita e leitura de memórias

Processo

Será descrito o processo do ciclo de escrita e leitura da memória para o pino de $\overline{WAIT}$ desativado.

Passo 1 - O sinal $\overline{MREQ}$ torna-se ativo quando o endereço está estável na via de endereços (address bus), sendo assim pode ser usado como sinal para habilitar as memórias dinâmicas.

Passo 2 - O sinal $\overline{WR}$ torna-se ativo quando o dado na via de dados atinge estabilidade e a operação é de escrita na memória. Este sinal é ativado na metade do ciclo T1 e desativado na metade do ciclo T3.

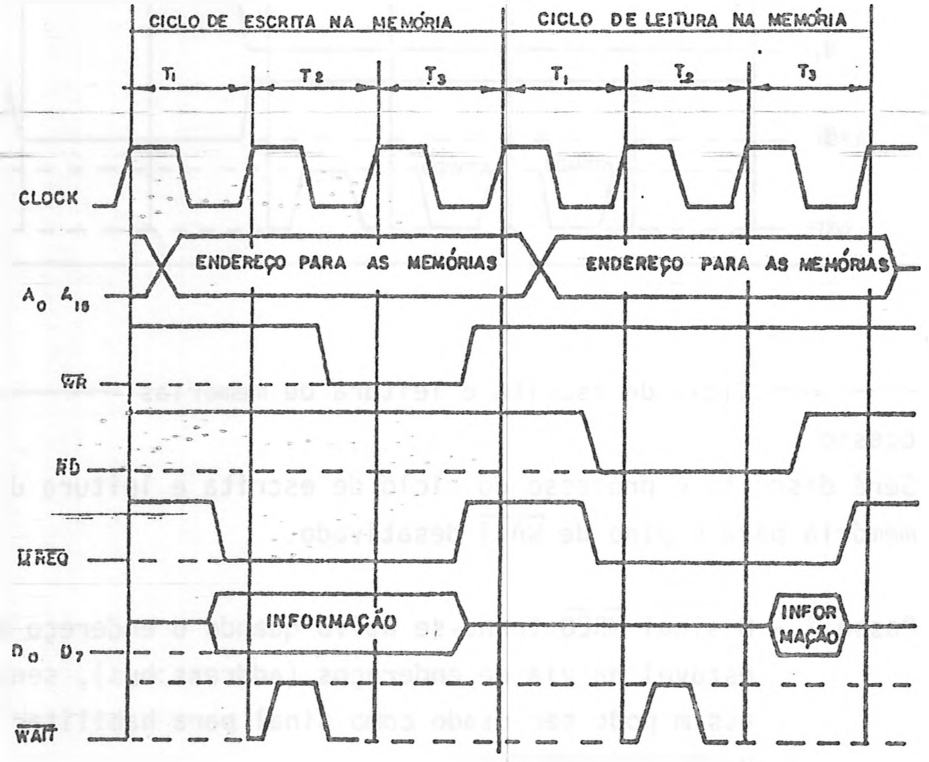
Passo 3 - O sinal $\overline{RD}$ torna-se ativo quando o dado na via de dados atinge estabilidade e a operação é de leitura na memória. Este sinal é ativado na metade do ciclo T_1 e desativado na metade do ciclo T_3.

Comentário

. Tanto o sinal de $\overline{WR}$ como o sinal de $\overline{RD}$ poderão ser usados como sinal de leitura/escrita (R/W) para qualquer memória de semicondutor.

. Nunca serão ativados simultaneamente os sinais de $\overline{RD}$ e $\overline{WR}$.

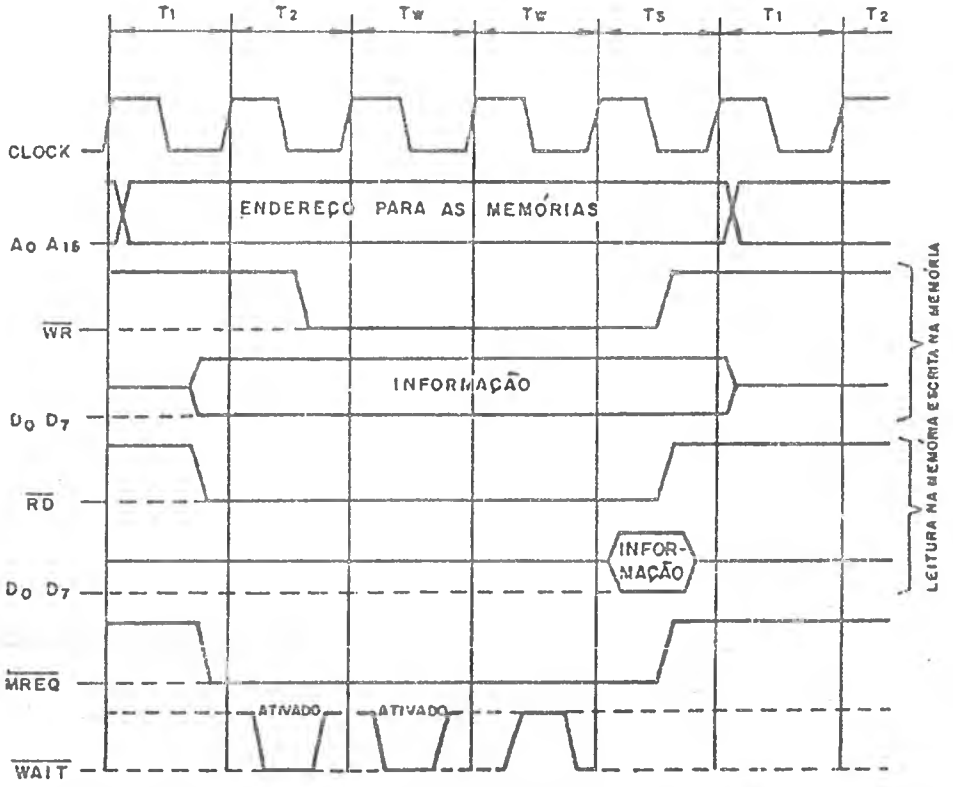
Ilustração



Comentário

Se durante o ciclo de escrita ou de leitura na memória for gerado sinal de $\overline{WAIT}$, serão gerados n ciclos Tw até que este desapareça.

Ilustração


T1
T2
Tw
Tw
T3
T1
T2
CLOCK
ENDEREÇO PARA AS MEMÓRIAS
A0 A15
WR
INFORMAÇÃO
D0 D7
RD
INFOR-
MAÇÃO
D0 D7
MREQ
ATIVADO
ATIVADO
WAIT
LEITURA NA MEMÓRIA ESCRITA NA MEMÓRIA

Ciclo de escrita e leitura de periféricos

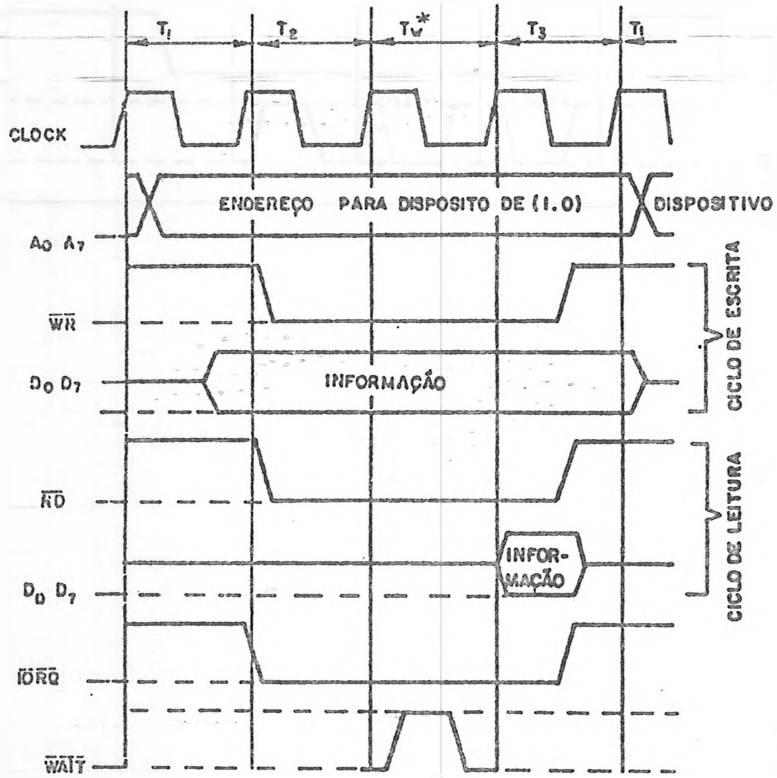
Processo

Será descrito o processo do ciclo de escrita e leitura de periféricos para o pino de $\overline{WAIT}$ (espera) positivado.

Comentário

Durante as operações de E/S, a CPU gera automaticamente um estado de espera, porque como $\overline{IORQ}$ só é ativado no início de T2, há um intervalo muito pequeno até a amostragem de $\overline{WAIT}$ na descida do clock no estado T2. Para dar mais tempo aos decodificadores e dispositivos de E/S, a CPU insere automaticamente um Tw(*) independentemente do pedido de um periférico durante T2.

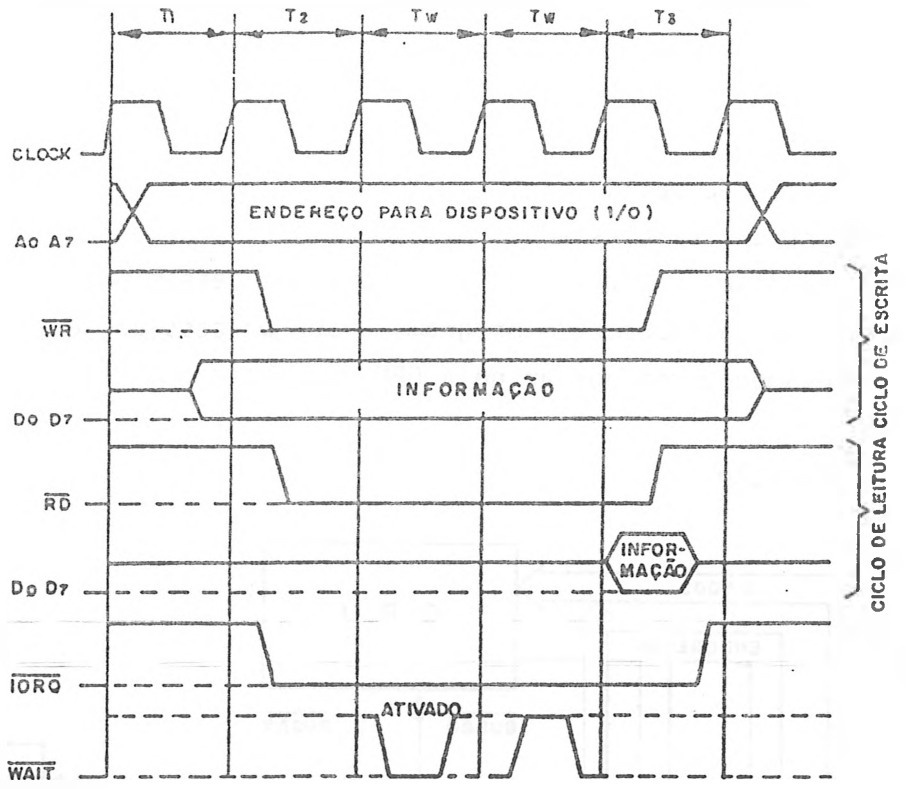
Ilustração



Comentário

Se durante o ciclo de escrita ou de leitura de periféricos for gerado sinal de $\overline{WAIT}$, serão gerados n ciclos Tw até que este desapareça.

Ilustração


T1
T2
Tw
Tw
T3
CLOCK
ENDEREÇO PARA DISPOSITIVO (I/O)
A0 A7
WR
INFORMAÇÃO
D0 D7
RD
INFOR-
MAÇÃO
D0 D7
IORQ
ATIVADO
WAIT
CICLO DE LEITURA CICLO DE ESCRITA

TA 30920

Ciclo de requisição das vias/reconhecimento

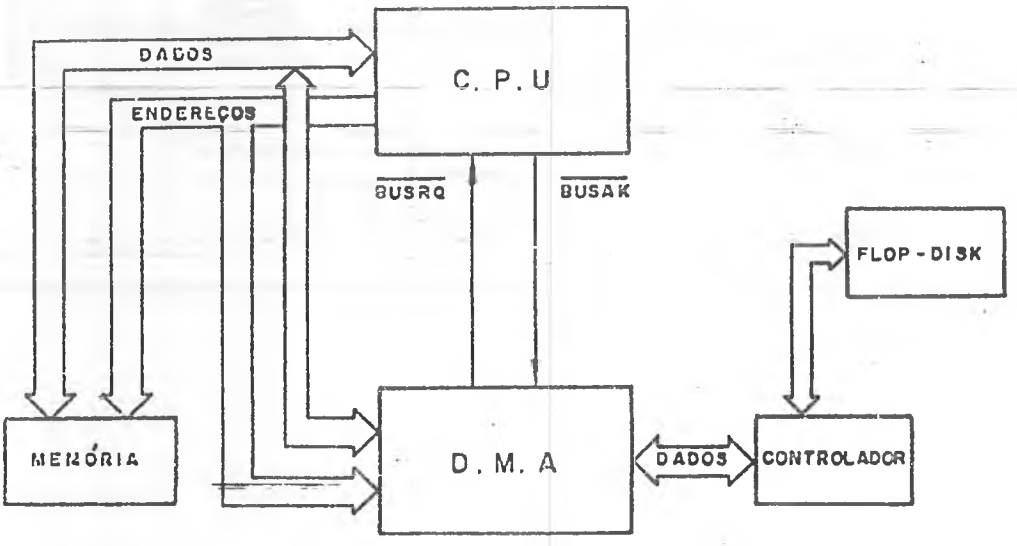
Introdução

Em casos específicos alguns dispositivos externos precisam das vias de endereço, dados e de controle para suas operações.

Exemplo

É o caso de pedido de acesso direto à memória (DMA). Isto ocorre quando no sistema existem memórias de grande capacidade (flop-disk) e a transferência de dados dessas memórias para a RAM, que armazenará o programa a ser processado, é feita diretamente sem passar pela CPU.

Ilustração



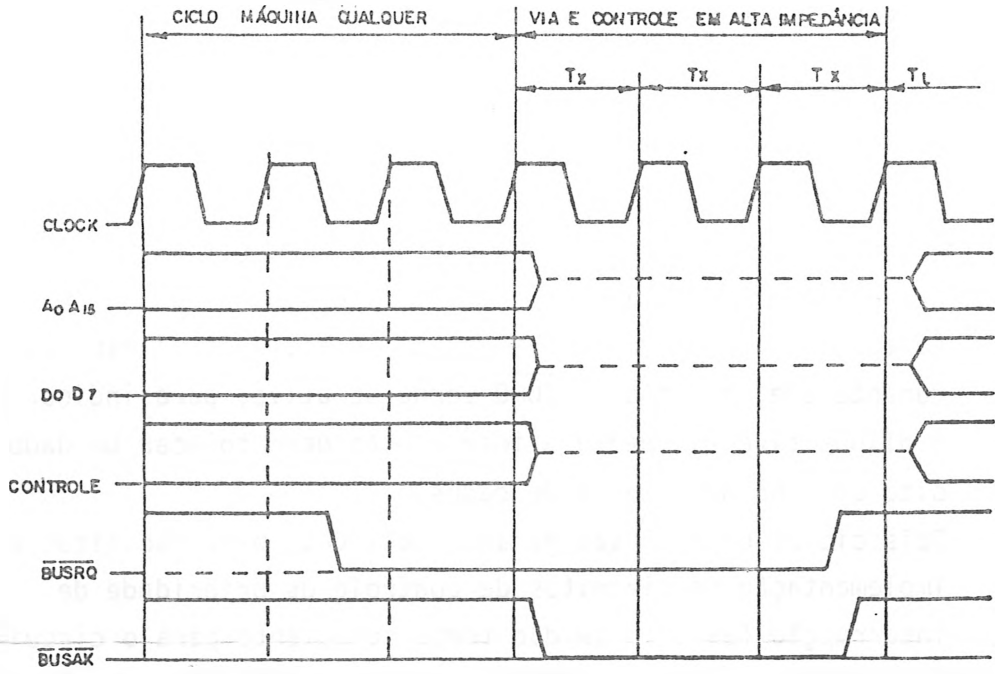
Processo

Passo 1 - O sinal $\overline{BUSRQ}$ é amostrado pela CPU na subida do último ciclo de clock de qualquer ciclo de máquina (não necessariamente no fim do ciclo de instrução).

Passo 2 - Quando o sinal $\overline{BUSRQ}$ for reconhecido pela CPU, está gerará o $\overline{BUSAK}$ na subida seguinte ao clock.

Passo 3 - A CPU coloca em estado de alta impedância os barramentos de endereços, dado e controle.

Ilustração



Comentário

. Se houver memória dinâmica, o controlador externo deverá gerar os sinais para reciclagem da memória durante operações de DMA.

. Durante o ciclo de requisição de vias/reconhecimento, a CPU não atende a $\overline{INT}$ ou $\overline{NMI}$.

Ciclo de interrupção

Processo

Será descrito a seguir o processo pelo qual a CPU atende a uma interrupção.

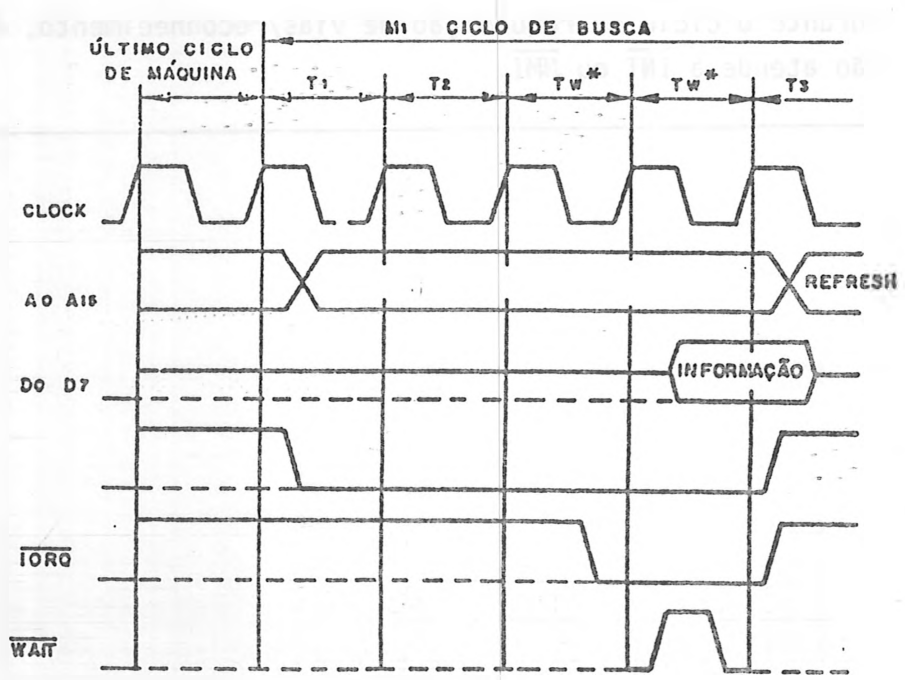
. Sinal $\overline{INT}$ é mostrado pela CPU na subida do sinal de clock, referente ao último ciclo de clock, do último ciclo de máquina de uma instrução.

. O pedido de interrupção é aceito se a CPU estiver habilitada e não houver $\overline{BUSRQ}$ ativo.

. Ao aceitar a interrupção, é gerado um ciclo de M1 especial e, durante ele, o ciclo de $\overline{IORQ}$ torna-se ativo, para indicar que o dispositivo que pediu a interrupção deve colocar um dado de oito bits no barramento de dados.

. Dois ciclos de WAIT são gerados pela CPU, para facilitar a implementação de circuitos de controle de prioridade de interrupção (os dois Tw dão tempo suficiente para o circuito determinar, dentro da cadeia de prioridades, qual dispositivo deve colocar os dados na via.

Ilustração



Resposta de interrupção não mascarada

Processo

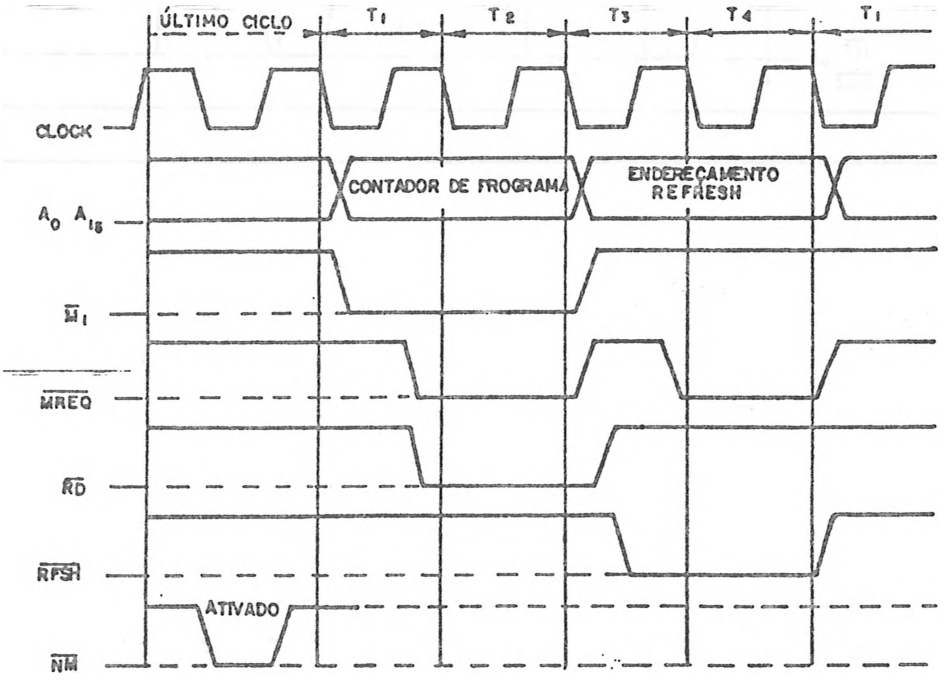
Será descrito a seguir o processo pelo qual a CPU atende a uma interrupção não mascarada.

. O sinal $\overline{NMI}$ é verificado pela CPU, ao mesmo tempo que o sinal $\overline{INT}$, porém esta linha tem prioridade sobre a interrupção normal.

. A resposta da CPU é similar a uma operação de leitura de memória, com a diferença que o conteúdo do barramento de dados é ignorado.

. O conteúdo do PC é automaticamente preservado na pilha e a execução é transferida para o endereço 0066H, onde deve estar situado o início da rotina de atendimento a interrupção não mascarável.

Ilustração



Comentário

O $\overline{NMI}$ geralmente é utilizado para que a CPU responda rapidamente a sinais de alarme de situações catastróficas como uma falha de alimentação iminente.

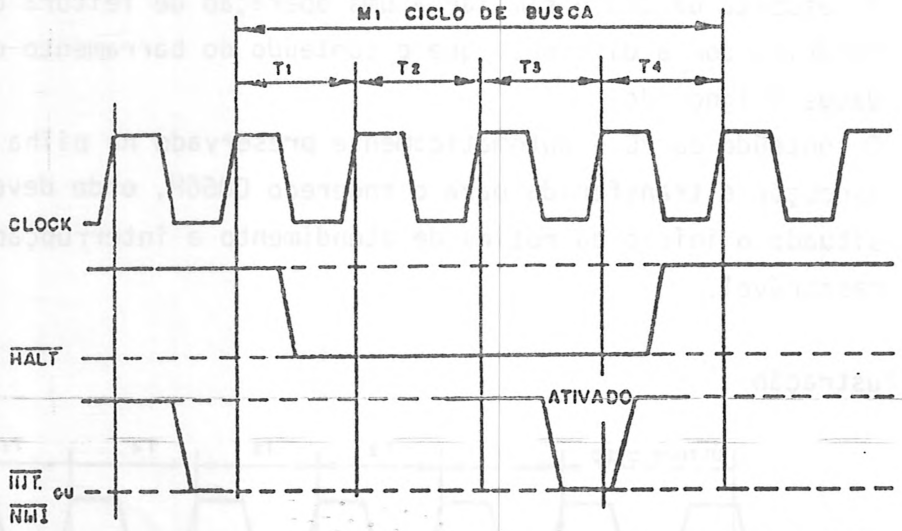
TA 30920

HALT

Processo

Sempre que através de Software for gerada a instrução HALT, a CPU começa a executar instruções NOP (no operations), até que uma interrupção seja requerida ou um reset seja dado.

Ilustração



Palavras-chave

MICROPROCESSADOR Z-80 - CIRCUITO DE TEMPO

INTRODUÇÃO ÀS INTERFACES PARALELAS PROGRAMÁVEIS — TE 60061
(Programmable peripheral interface - PPI) — (5 páginas)

Apresentação

Para que um sistema baseado em microprocessadores consiga processar devidamente as informações é necessário que haja uma adequação das informações que entram e saem do micro.
Essa adequação é feita pelos circuitos de interface.

Descrição

São circuitos integrados que podem ser programados para receber ou enviar dados na forma paralela, sendo normalmente conectados entre o bus de dados e os circuitos externos que levam e trazem informações para o sistema

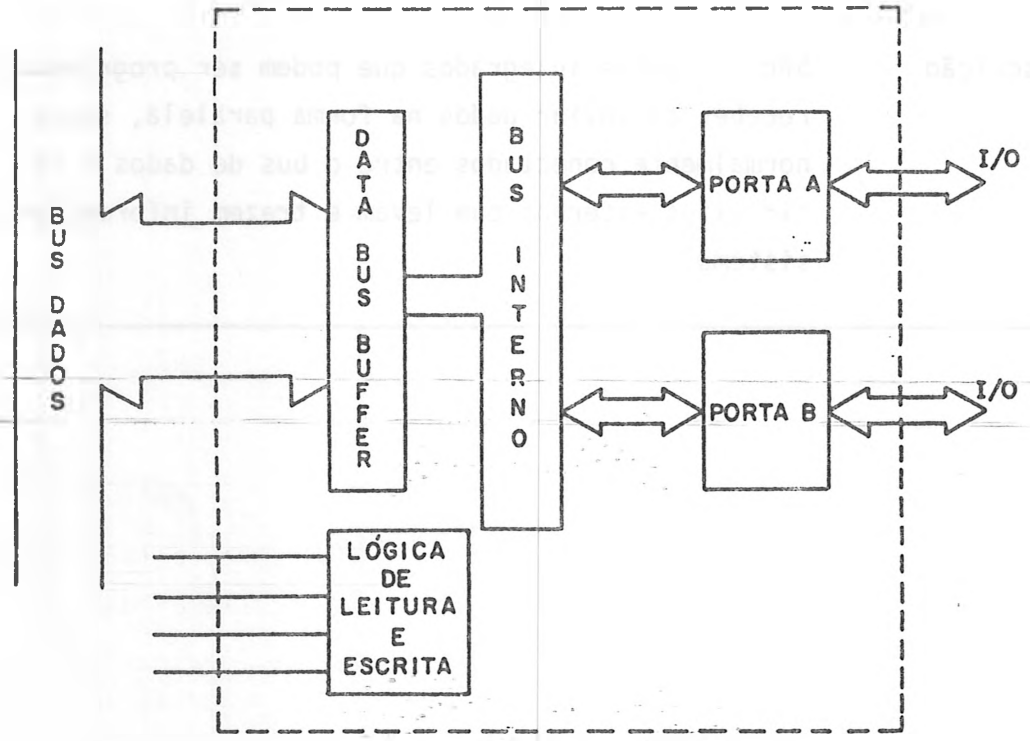
---

Estrutura

Dispositivos de entrada e saída paralela

Geralmente os componentes desta família têm a seguinte estrutura.

Ilustração



Processo

Os dados a serem enviados através da interface podem seguir dois fluxos distintos a saber:

. dados do sistema de microprocessador para fora do sistema;

. dados externos para dentro do sistema de microprocessador.

Se...	... então...
... os dados vão do sistema para fora...	... eles chegam pelo bus de dados do sistema, são armazenados no data bus buffer, caminham pelo bus interno e são colocados nas portas de saída, conforme a programação que o componente sofreu anteriormente.
... os dados vêm de fora para dentro do sistema...	... eles entram pela portas de I/O já programadas para funcionarem como entrada, chegam até o bus interno, são armazenados no data bus buffer e finalmente são entregues ao microprocessador pelo bus de dados.

Usos

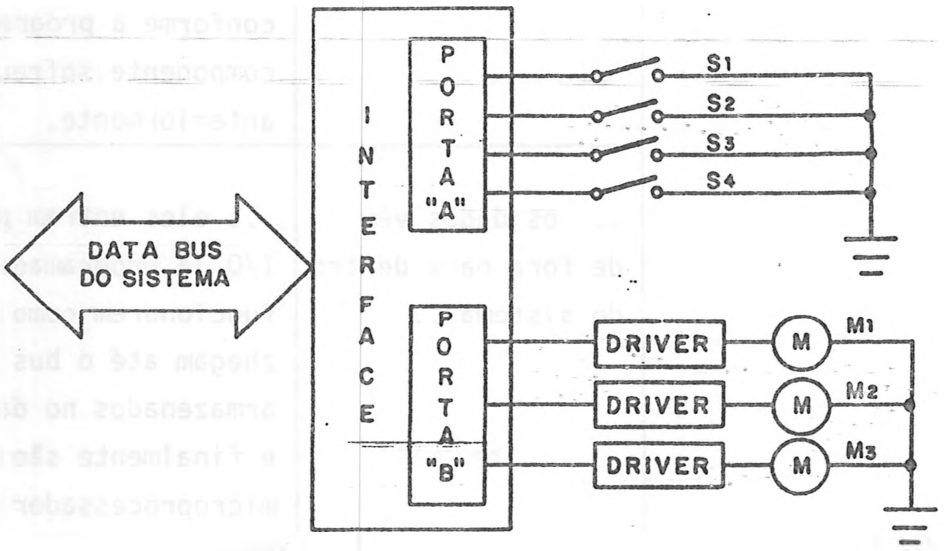
Algumas aplicações das interfaces paralelas programáveis são:

. leitura de circuitos sensores (chaves fim-de-curso, dispositivos optoeletrônicos, etc.);
. acionamento de cargas (drivers de motores, solenóides, etc.);
. comunicação de dados no formato paralelo.

Exemplo

Um exemplo de aplicação para um dispositivo de entrada e saída paralela é dado abaixo.

Ilustração



Comentário

As chaves S_1, S_2, S_3 e S_4 são ligadas à porta A e os motores M_1, M_2 e M_3 à porta B.

A seqüência das chaves é lida pelo microprocessador através da interface. Conforme a seqüência de chaves ligadas, os motores serão acionados em determinada ordem.

Existem circuitos integrados com várias configurações, variando o número de portas e os modos de programação de fabricante para fabricante.

Fabricante	Componente	Nº de portas programáveis
INTEL	8255	3
ZILOG	Z80 PIO	2
MOTOROLA	MC6821	2

Palavras-chave

INTERFACES PARALELAS PROGRAMÁVEIS

INTERFACE PARALELA PROGRAMÁVEL 8255 — TA 30530

(Programmable perifheral interface 8255) — (8 páginas)

Apresentação

O circuito integrado 8255 foi projetado para funcionar como conjunto de portas paralelas bidirecionais em aplicações que envolvam interfaces paralelas.
O 8255 é compatível com as famílias de microprocessadores 8080 e 8085.

Descrição

A interface paralela programável 8255 é apresentada em invólucro plástico de quarenta pinos tipo dual-in-line, utiliza tecnologia LSI e contém três conjuntos de portas bidirecionais que interligam a via de dados do sistema aos circuitos externos.

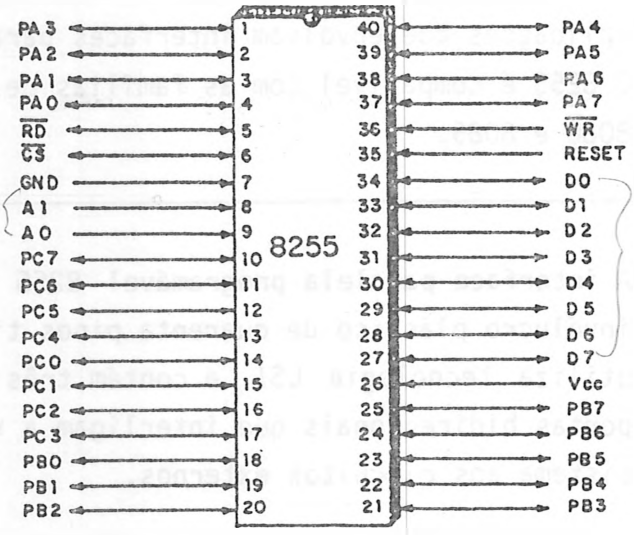
Características

As principais características do componente são as seguintes:

. alimentação: 5 V ± 5 %
. três portas de oito bits cada, sendo que a programação de cada porta é selecionada por software (programa);
. compatibilidade com microprocessadore 8080 e 8085;
. programação feita utilizando-se as instruções de entrada e saída (IN e OUT) do conjunto de instruções do microprocessador 8085.

Estrutura

A pinagem do componente pode ser vista abaixo, assim como os respectivos sinais de entrada e saída.



A tabela de funções abaixo mostra o significado de cada sinal.

Pino	Nome do sinal	Função
1	PA_3	Bit 3 da porta A
2	PA_2	Bit 2 da porta A
3	PA_1	Bit 1 da porta A
4	PA_0	Bit 0 da porta A
5	$\overline{RD}$	Sinal em nível baixo que habilita o envio de dados da interface para o microprocessador através do bus de dados.
6	$\overline{CS}$	Sinal em nível baixo que habilita o funcionamento da interface.
7	GND	Referência 0V (terra)

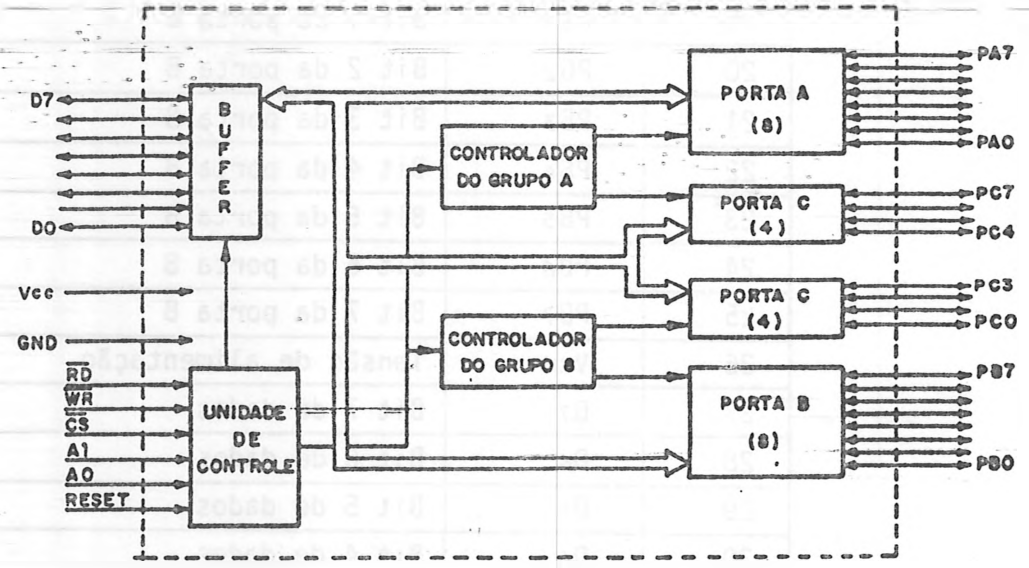
8	A_1	Pino conectado com a via de endereço do sistema, usado para selecionar uma das três portas ou o registrador da palavra de controle 8255.
9	A_0	Idem ao pino A_1
10	PC_7	Bit 7 da porta C
11	PC_6	Bit 6 da porta C
12	PC_5	Bit 5 da porta C
13	PC_4	Bit 4 da porta C
14	PC_0	Bit 0 da porta C
15	PC_1	Bit 1 da porta C
16	PC_2	Bit 2 da porta C
17	PC_3	Bit 3 da porta C
18	PB_0	Bit 0 da porta B
19	PB_1	Bit 1 da porta B
20	PB_2	Bit 2 da porta B
21	PB_3	Bit 3 da porta B
22	PB_4	Bit 4 da porta B
23	PB_5	Bit 5 da porta B
24	PB_6	Bit 6 da porta B
25	PB_7	Bit 7 da porta B
26	Vcc	Tensão de alimentação
27	D_7	Bit 7 de dados
28	D_6	Bit 6 de dados
29	D_5	Bit 5 de dados
30	D_4	Bit 4 de dados
31	D_3	Bit 3 de dados
32	D_2	Bit 2 de dados
33	D_1	Bit 1 de dados
34	D_0	Bit 0 de dados
35	RESET	Sinal vindo do sistema para limpar os registradores internos e resetar a interface.

36	$\overline{WR}$	Sinal em nível baixo que habilita a interface a receber dados ou palavras de controle do microprocessador.
37	PA_7	Bit 7 da porta A
38	PA_6	Bit 6 da porta A
39	PA_5	Bit 5 da porta A
40	PA_4	Bit 4 da porta A

Estrutura

O diagrama de blocos abaixo sintetiza os vários circuitos que compõem a interface 8255.

Ilustração



As três portas de comunicação da interface são divididas em dois grupos: o grupo A, formado pela porta A e os quatro bits mais significativos da porta C (PC_7 - PC_4) e o grupo B formado pela porta B e os quatro bits menos significativos da porta C (PC_3 - PC_0).

Tabela de função das portas

Bloco	Função
Buffer	Armazenar temporariamente dados que entram ou saem da interface.
Unidade de controle	Receber e processar os sinais de controle vindos do sistema e selecionar as portas de entrada e saída através dos controladores.
Controladores	São dois circuitos que controlam os dois grupos de portas.
Porta A	Porta bidirecional de oito bits.
Porta B	Porta bidirecional de oito bits.
Porta C	Duas portas bidirecionais de quatro bits.

Procedimento

A interface 8255 pode operar em três modos distintos:

Modo 0

As portas são programadas basicamente para serem entradas e saídas.

Modo 1

As portas A e B são programadas para entrada e saída, enquanto a porta C recebe sinais especiais de controle.

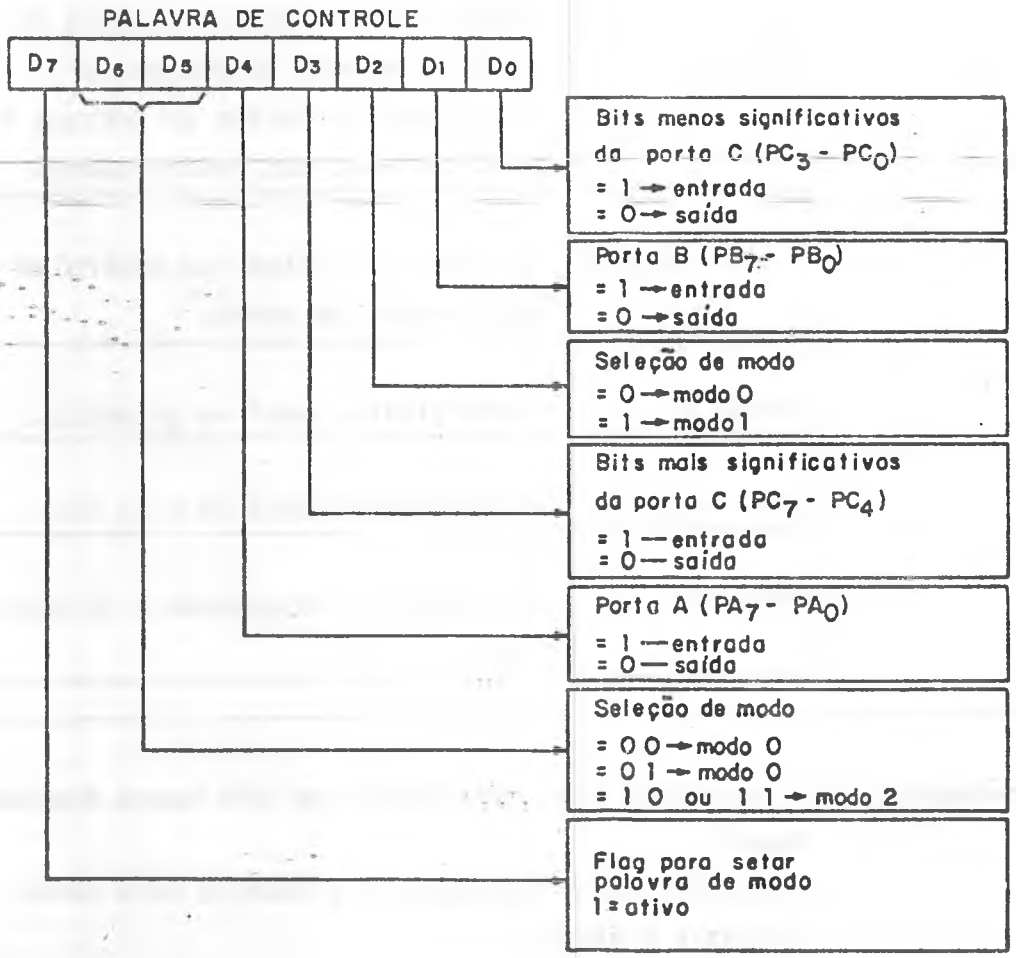
Modo 2

A porta B não é usada, a porta A é uma via bidirecional e a porta C recebe sinais especiais de controle.

Comentário

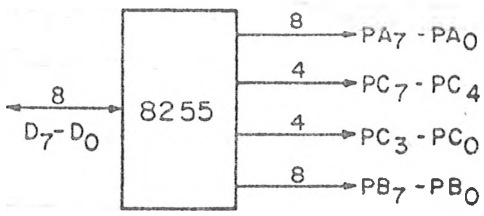
Será analisado a seguir o Modo 1 que é o mais utilizado. Os outros dois modos são configurações especiais encontradas no manual do fabricante.

A programação é feita através de uma palavra de controle, enviada ao registrador da unidade de controle. Essa palavra de controle é definida como segue:

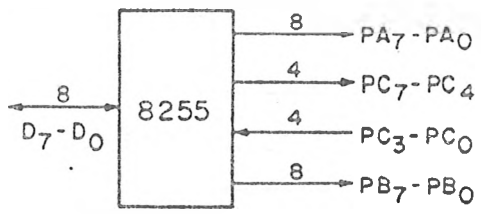


As variações da palavra de controle resultam em dezesseis possibilidades que são mostradas a seguir.

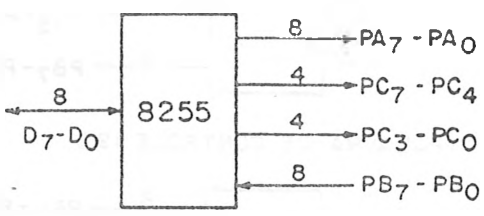
PALAVRA DE CONTROLE = 80



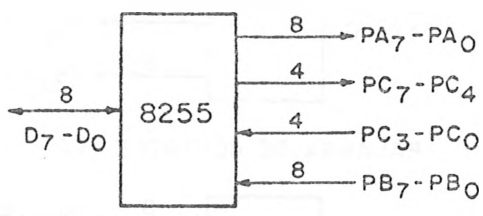
PALAVRA DE CONTROLE = 81



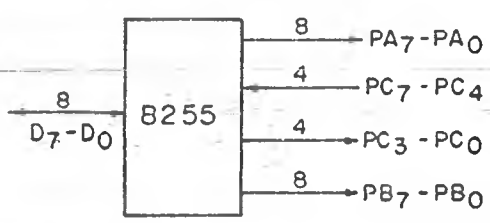
PALAVRA DE CONTROLE = 82



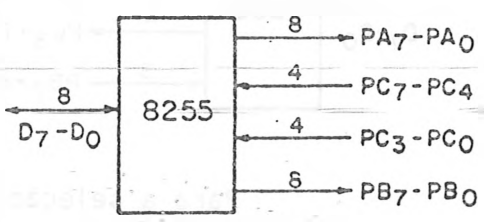
PALAVRA DE CONTROLE = 83



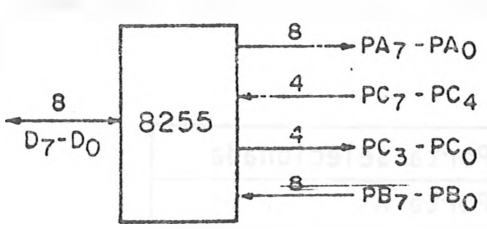
PALAVRA DE CONTROLE = 88



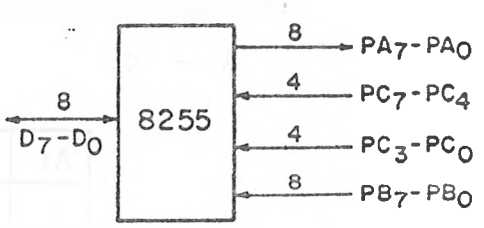
PALAVRA DE CONTROLE = 89



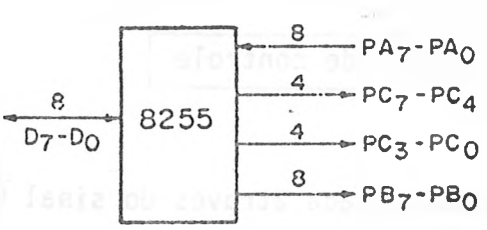
PALAVRA DE CONTROLE = 8A



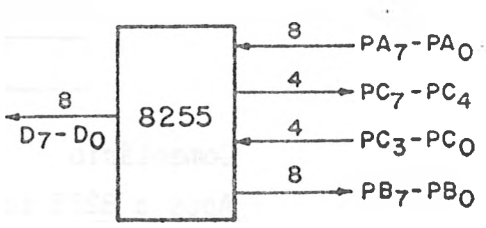
PALAVRA DE CONTROLE = 8B



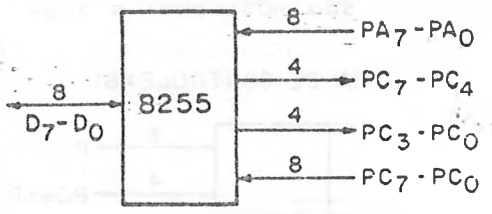
PALAVRA DE CONTROLE = 90



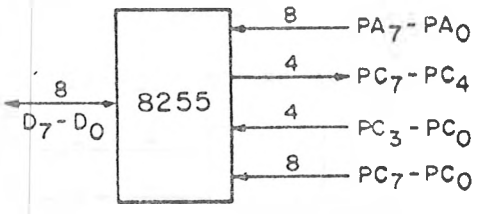
PALAVRA DE CONTROLE = 91



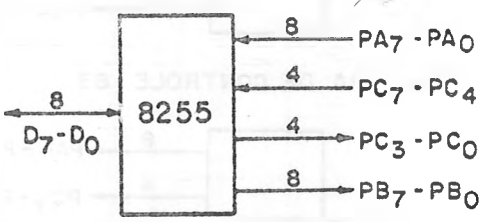
PALAVRA DE CONTROLE = 92



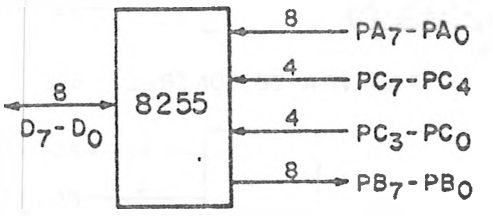
PALAVRA DE CONTROLE = 93



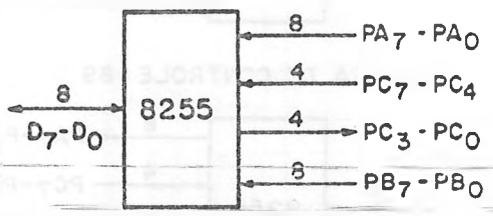
PALAVRA DE CONTROLE = 98



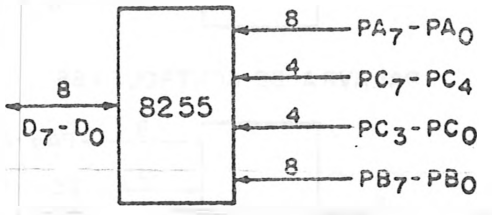
PALAVRA DE CONTROLE = 99



PALAVRA DE CONTROLE = 9A



PALAVRA DE CONTROLE = 9B



Para a seleção de um dos três conjuntos de portas (A, B, C) e da unidade de controle da 8255, utilizamos a seguinte tabela de endereçamento através dos bits A_1 e A_0.

A_1	A_0	Porta selecionada
0	0	Porta A
0	1	Porta B
1	0	Porta C
1	1	Unidade de controle

Comentário

Após a 8255 ter sido habilitada através do sinal $\overline{CS}$, a seleção da porta que será utilizada é feita conforme a combinação dos bits A_1 e A_0 da tabela acima.

Palavras-chave
PPI 8255

LINGUAGEM DE MÁQUINA

TE 70160
(1 página)

Descrição **Linguagem de máquina** são os códigos binários utilizados pelo microprocessador.

Estrutura A linguagem de máquina é representada em hexadecimal, e também é chamada de programa objeto.

Ilustração Linguagem de máquina em **hexadecimal**

Endereço	Dado
0160	0A
0161	66
0162	68
0163	59
0164	00
0165	69
0166	40
0167	3F
0168	56
0169	01
016A	0E
016B	F0
016C	A3

CONJUNTO DE INSTRUÇÕES

TA 31050
(1 página)

Descrição Conjunto de instruções é um grupo de comandos reconhecidos pelo microprocessador.

Comentário As instruções são intrínsecas à arquitetura do microprocessador, o que requer seu conhecimento pelo programador.

Características As características principais desse conjunto são os números e os tipos de instruções que variam para cada família de microprocessador.

Classificação O conjunto de instruções de um microprocessador apresenta, quanto à sua funcionalidade, a seguinte classificação:

. Transferência de oito bits
. Transferência de 16 bits
. Lógicas e aritméticas de oito bits
. Aritméticas de 16 bits
. Instruções aritméticas especiais
. JUMPs, ~~CALLs e~~ RETs
. Transferência de blocos e pesquisa
. Deslocamento e rotações
. Instruções que suspendem a operação da CPU
. Manipulação e teste de bits
. Instruções que modificam o flag de carry
. Controle da CPU
. Modos de interrupção
. Habilitação e desabilitação de interrupção
. Entradas e saídas

TRANSFERÊNCIA DE OITO BITS

TA 31060
(13 páginas)

Descrição

Transferência de oito bits consiste na cópia, em local predeterminado, de um byte a partir de uma origem definida.

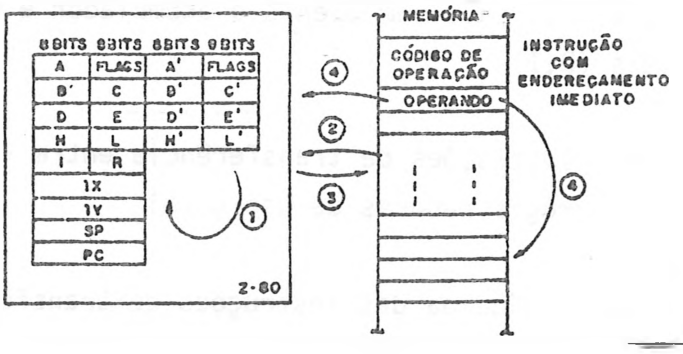
Classificação

O conjunto de instruções de transferência de oito bits classifica-se em quatro grupos.

① Registrador ← Registrador
② Registrador ← Memória
③ Memória ← Registrador
④ Registrador, memória ← Constante

Ilustração

Funções dos subgrupos de instruções de transferência de oito bits



Registrador ← Registrador

Descrição

O conjunto de instruções de transferência de oito bits Registrador ← Registrador são instruções de transferência entre registradores de uso geral, internas ao Z80.

Símbolo

LD r,r'

O mnemônico que identifica essas instruções de transferência no Z80 é **LD**, decorrente da palavra **LOAD**.

Os operandos **r** e **r'** podem ser qualquer registrador de uso geral: A, B, C, D, E, H e L.

Classificação

As instruções de transferência de oito bits de Registrador ← Registrador podem ser:

. Instruções de transferência entre registradores de uso geral
. Instruções que envolvem apenas o acumulador e registradores especiais (R,I)

Instruções de transferência entre registradores de uso geral

Estrutura

O código de máquina das instruções de transferência de oito bits é formado por um byte que tem a seguinte forma:

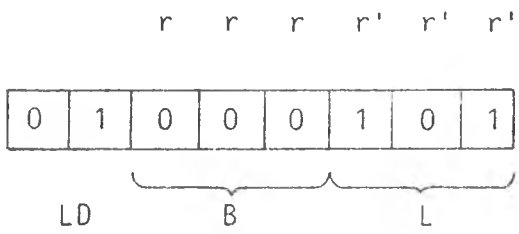
0	1	r	r	r	r'	r'	r'

onde:

Registrador	r r r	r' r' r'
B	0 0 0	0 0 0
C	0 0 1	0 0 1
D	0 1 0	0 1 0
E	0 1 1	0 1 1
H	1 0 0	1 0 0
L	1 0 1	1 0 1
(HL)	1 1 0	1 1 0
A	1 1 1	1 1 1

Exemplo

Determinar o código de máquina da instrução LD B, L.



Código = 45H

Processo

As instruções do tipo LD r, r' transferem o conteúdo do registrador r' para o registrador r. O operando da esquerda indica sempre para onde é destinada (destino) a informação e o da direita indica de onde ela se origina (origem).

Observação

Como estas instruções não envolvem a ULA, os bits dos flags não são afetados.

Exemplo

Dada a instrução LD B,E

Mnemônico: LD

Operando: r = B Destino

Operando: r' = E Origem

A instrução transfere o conteúdo do registrador E para o registrador B.

Observação

O registrador de origem, aquele do qual a informação é retirada, permanece inalterado após a transferência.

Instruções que envolvem apenas o acumulador e os registradores especiais (R,I)

Estrutura

As instruções que envolvem o acumulador e os registradores especiais R (refresh) e I (interrupt vector) são instruções de dois bytes e de pouco uso devido ao fato de sua aplicação ser específica.

Processo

As instruções especiais de transferência de oito bits se comportam como na tabela abaixo:

Destino	Origem	Código da instrução
A	I	ED 57
A	R	ED 5F
I	A	ED 47
R	A	ED 4F

Exemplo

Dada a instrução LD I,A

Mnemônico: LD

Operando: I Destino

Operando: A Origem

A instrução transfere o conteúdo do registrador A para o registrador I.

Registrador ← Memória

Descrição

O conjunto de instruções de transferência de oito bits Registrador ← Memória são instruções que transferem dados da memória para registradores internos do Z80.

Símbolo

LD r, (memória)

O mnemônico que identifica estas instruções de transferência no Z80 é LD decorrente da palavra LOAD.

O operando r pode ser qualquer registrador de uso geral: A, B, C, D, E, H e L.

Estrutura

As instruções do segundo grupo são do tipo LD r, (memória), onde:

r	(Memória)
A	
B	(IX + d)
C	
D	(IY + d)
E	
H	(HL)
L	

r	(Memória)
A	(BC)
	(DE)
	(nn)

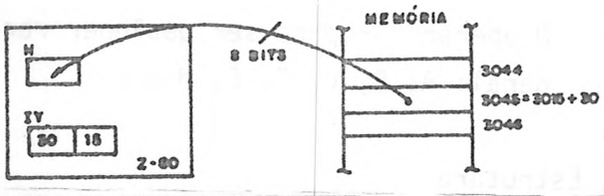
Processo

O operando da esquerda indica sempre um registrador de oito bits para onde é destinada a informação.

O da direita indica o endereço de memória onde se encontra o byte a ser transferido para o registrador.

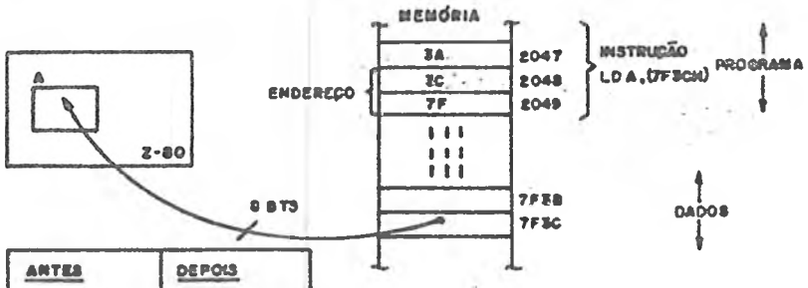
Exemplos

. Instrução LD H, (IY + 30H)



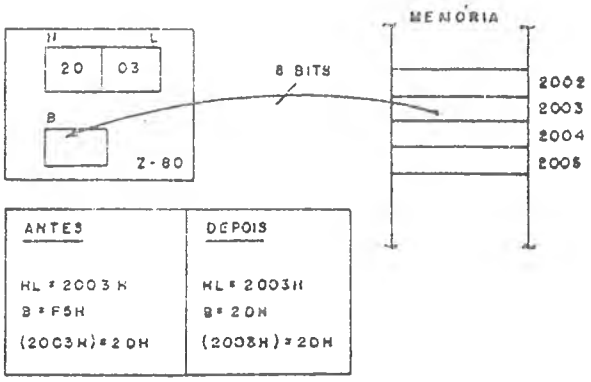
ANTES	DEPOIS
H=23H IY=3015H (3045H)=48H	H=48H IY=3015H (3045H)=48H

. Instrução LD A, (7F3CH)



ANTES	DEPOIS
A=95H (7F3CH)=06H	A=06H (7F3CH)=06H

. Instrução LD B,(HL)



A instrução transfere o conteúdo da posição de memória dada pelo par de registradores HL no registrador B.
A posição de memória de origem não é alterada após a execução da instrução.

Memória ← Registrador

Descrição

São instruções que transferem dados de registradores internos do Z80 para posições de memória.

Símbolo

LD (memória), r

O mnemônico que identifica estas instruções de transferência no Z80 é o LD, decorrente da palavra LOAD.

O operando r pode ser qualquer registrador de uso geral: A, B, C, D, E, H e L.

Estrutura

As instruções do terceiro grupo são do tipo LD (memória), r, onde:

(Memória)	r
(IX + d)	A
(IY + d)	B
(HL)	C
	D
	E
	H
	L

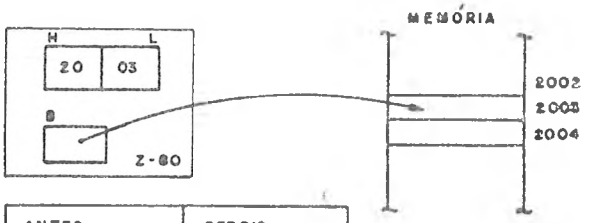
(Memória)	r
(BC)	A
(DE)	
(nn)	

Processo

O operando da esquerda indica o endereço de memória para onde é destinada a informação. O da direita indica o registrador de onde será transferido o byte para a posição de memória indicada.

Exemplos

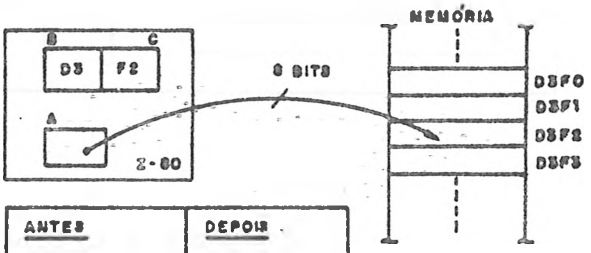
. Instrução LD (HL),B



ANTES	DEPOIS
HL = 2003 H	HL = 2003H
B = F5H	B = F5H
(2003) = 2DH	(2003) = F5H

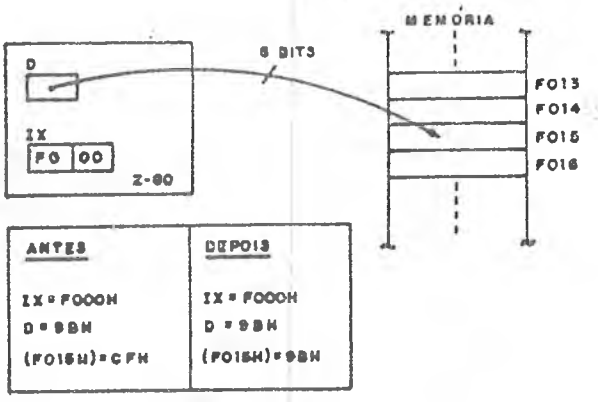
A instrução transfere o conteúdo do registrador B para o endereço de memória especificado pelo par de registradores (HL).
A posição de origem (B = F5H) não é alterada após a execução da instrução.

. Instrução LD (BC),A

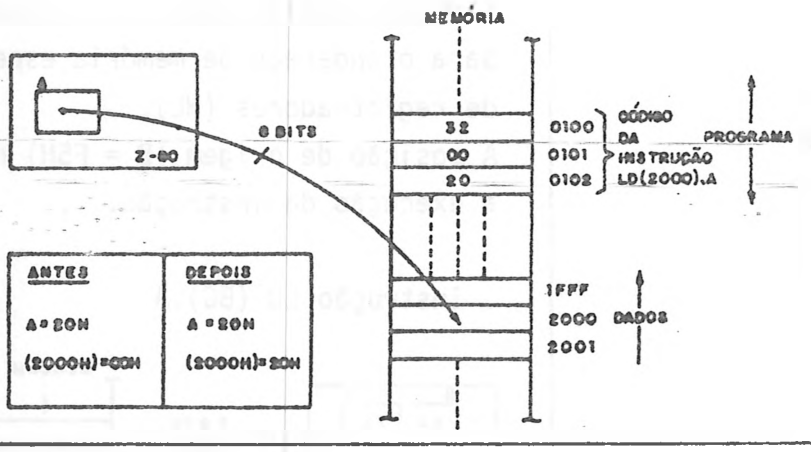


ANTES	DEPOIS
BC = D3F2H	BC = D3F2H
A = FFH	A = FFH
(D3F2H) = 3CH	(D3F2H) = FFH

. Instrução LD (IX + 15),D



. Instrução LD (2000),A



Registrador, memória ← Constante

Descrição

São instruções que transferem um operando imediato (constante) para um registrador interno do Z80 ou para uma posição de memória.

Símbolos

LD r,n

LD (memória), n

O mnemônico que identifica estas instruções de transferência no Z80 é o **LD**, decorrente da palavra **LOAD**.

O operando **r** pode ser qualquer registrador de uso geral: A, B, C, D, E, H e L.

O **n** é uma constante denominada **dado imediato**.

Estrutura

As instruções do quarto grupo são do tipo LD r,n ou LD (memória),n onde:

. LD r,n

r	n
A	DADO
B	DADO
C	DADO
D	DADO
E	DADO
H	DADO
L	DADO

. LD (memória),n

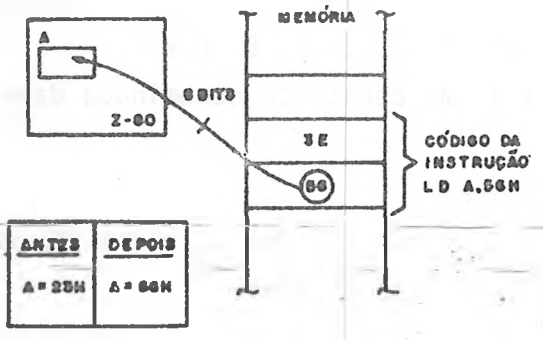
Memória	n
(HL)	DADO
(Ix + d)	DADO
(Iy + d)	DADO

Processo

O operando da esquerda representa sempre um registrador de oito bits ou uma posição de memória para onde é destinada a informação.
O da direita é sempre o dado imediato que será copiado no registrador ou na posição de memória indicada.

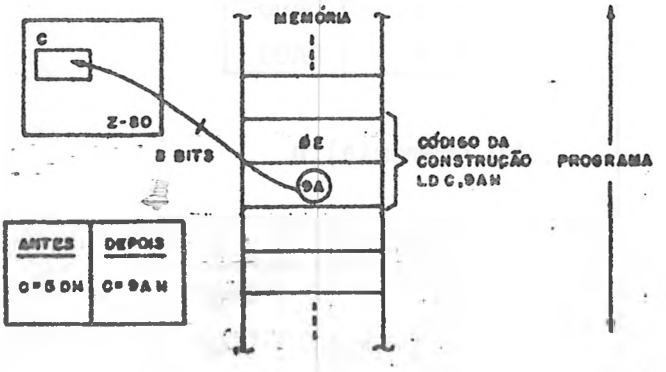
Exemplos

. Instrução LD A, 56H



A instrução transfere o dado imediato ao código de operação para o registrador A indicado pela instrução.
A posição de memória não é alterada após a execução da instrução.

. Instrução LD C, 9AH



INSTRUÇÕES DE TRANSFERÊNCIA DE 8 BITS

Mnemônico 8080	Mnemônico Z-80	Operação Simbólica	Flags C	Z	P/V	S	N	H	Código de Máquina Binário 76	543	210	Nº de Bytes	Nº de M ciclos	Nº de T ciclos	SUB GRUPO
OPERAÇÕES REGISTRADOR ← REGISTRADOR															
Mov r,r'	LD r,r'	r ← r'	•	•	•	•	•	•	01	r	r'	1	1	4	1
–	LD, A,I	A ← I	•	↕	IFF	↕	0	0	11 01	101 010	101 111	2	2	9	
–	LD A,R	A ← R	•	↕	IFF	↕	0	0	11 01	101 011	101 111	2	2	9	
–	LD, I,A	I ← A	•	•	•	•	•	•	11 01	101 000	101 111	2	2	9	
–	LD R,A	R ← A	•	•	•	•	•	•	11 01	101 001	101 111	2	2	9	
OPERAÇÕES REGISTRADOR ← MEMÓRIA															
Mov r,M	LD r,(HL)	r ← (HL)	•	•	•	•	•	•	01	r	110	1	2	7	2
–	LD r, (IX + d)	r ← (IX + d)	•	•	•	•	•	•	11 01 ←	011 r d	101 110 →	3	5	19	
–	LD r, (IY + d)	r ← (IY + d)	•	•	•	•	•	•	11 01 ←	111 r d	101 110 →	3	5	19	
LDAX B	LD A,(BC)	A ← (BC)	•	•	•	•	•	•	00	001	010	1	2	7	
LDAX D	LD A,(DE)	A ← (DE)	•	•	•	•	•	•	00	011	010	1	2	7	
LDA end	LD A,(nn)	A ← (nn)	•	•	•	•	•	•	00 ← ←	111 n n	010 → →	3	4	13	
OPERAÇÕES MEMÓRIA ← REGISTRADOR															
MOV M,r	LD (HL),r	(HL) ← r	•	•	•	•	•	•	01	110	r	1	2	7	3
–	LD (IX + d),r	(IX + d) ← r	•	•	•	•	•	•	11 01 ←	011 110 d	101 r →	3	5	19	
–	LD (IY + d),r	(IY + d) ← r	•	•	•	•	•	•	11 01 ←	111 110 d	101 r →	3	5	19	
STAX B	LD (BC),A	(BC) ← A	•	•	•	•	•	•	00	000	010	1	2	7	
STAX D	LD (DE),A	(DE) ← A	•	•	•	•	•	•	00	010	010	1	2	7	
STA end	LD (nn),A	(nn) ← A	•	•	•	•	•	•	00 ← ←	110 n n	010 → →	3	4	13	
OPERAÇÕES REGISTRADOR, MEMÓRIA ← CONSTANTE															
MVI r, Const.	LD r,n	r ← n	•	•	•	•	•	•	00 ←	r n	110 →	2	2	7	4
MVI M, Const.	LD (HL),n	(HL) ← n	•	•	•	•	•	•	00 ←	110 n	110 →	2	3	10	
–	LD (IX + d),n	(IX + d) ← n	•	•	•	•	•	•	11 00 ← ←	011 110 d n	101 110 → →	4	5	19	
–	LD (IY + d),n	(IY + d) ← n	•	•	•	•	•	•	11 00 ← ←	111 110 d n	101 110 → →	4	5	19	
1ª coluna		2ª coluna	3ª coluna						4ª coluna			5ª coluna	6ª coluna	7ª coluna	8ª coluna

NOTAS:

1 – "r,r' " simbolizam qualquer um dos registros A, B, C, D, E, H, L e na representação do código de máquina binário respeitam a seguinte convenção:

r,r'	Reg.
000	B
001	C
010	D
011	E
100	H
101	L
111	A

2 – "IFF" indica que o conteúdo do FLIP-FLOP IFF1 é transferido para o flag P/V
3 – "d" simboliza o deslocamento no endereçamento indexado.
4 – "n" simboliza o operando imediato.
5 – "nn" simboliza o endereço no modo de endereçamento estendido.
6 – Notação dos flags: • – não afetado, 0 – resetado, 1 – setado, X – indefinido.
↕ – o flag é afetado em função do resultado da operação.

TRANSFERÊNCIA DE 16 BITS

TA 31120
(15 páginas)

Descrição

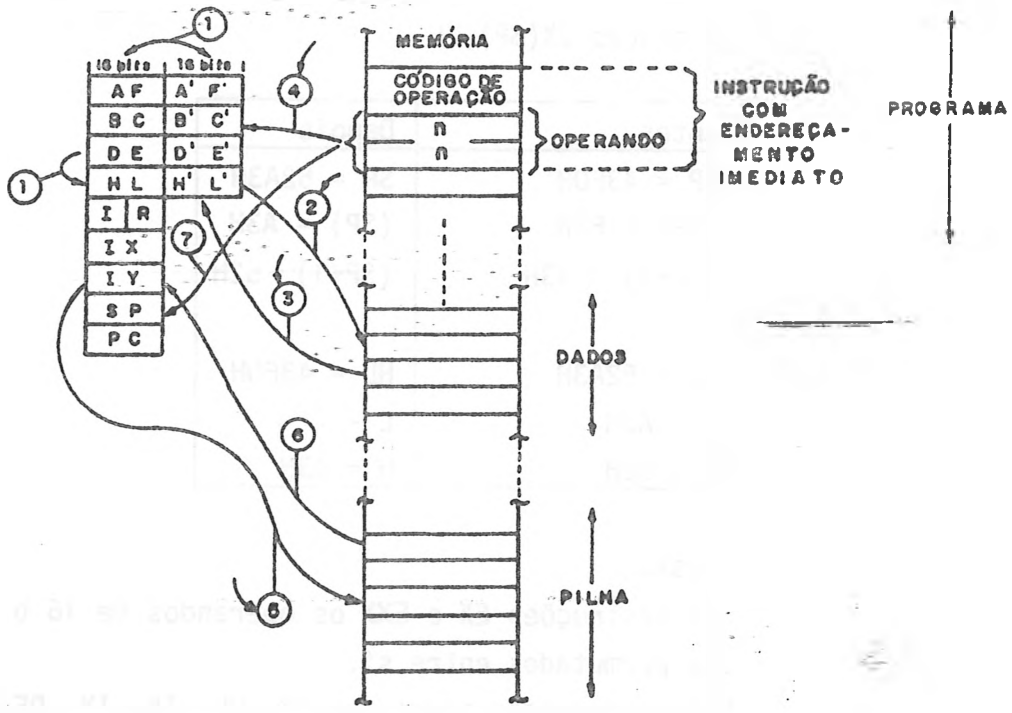
Transferência de 16 bits consiste na cópia, em local predeterminado, de dois bytes, a partir de uma origem definida.

Classificação

O grupo de instruções de transferência de 16 bits classifica-se da seguinte forma:

① Registrador ← Registrador
② Registrador ← Memória
③ Memória ← Registrador
④ Registrador ← Constante
⑤ Pilha ← Registrador
⑥ Registrador ← Pilha
⑦ Stack Pointer — Registrador

Ilustração



Registrador ← Registrador

Descrição

O conjunto de instruções de transferência de 16 bits Registrador ← Registrador consiste na permuta entre o conteúdo de registradores de uso geral, internos ao Z80

Símbolo

EX e EXX

Estrutura

São as seguintes as instruções:

EX(SP),HL	E3	EXDE,HL	EB	
EX(SP),IX	DDE3	EXAF,AF'	08	
EX(SP),IY	FDE3	EXX	D9	BCBC' DEDE' HLHL'

Exemplo

Instrução EX(SP),HL

Antes	Depois
SP = 43FOH	SP = 52A3H
(SP) = FFH	(SP) = A3H
(SP+1) = 43H	(SP+1) = 52H
HL = 52A3H	HL = 43FOH
L = A3H	L = FFH
H = 52H	H = 43H

Processo

Nas instruções EX e EXX os operandos de 16 bits são permutados entre si.

Estes operandos podem ser SP, HL, IX, IY, DE, AF, AF', BC', DE', DL'.

Registrador ← Memória

Descrição

O conjunto de instruções de transferência de 16 bits Registrador ← Memória são instruções que transferem os conteúdos de dois bytes em endereços consecutivos da memória para os pares de registradores BC, HL, DE, SP, IX e IY.

Símbolo

LD qq,(nn)

O mnemônico que identifica estas instruções de transferência no Z80 é o LD, decorrente da palavra LOAD.

qq pode ser um dos seguintes pares de registradores internos à CPU: BC, DE, HL, SP, IX, IY.

(nn) especifica o endereço da memória.

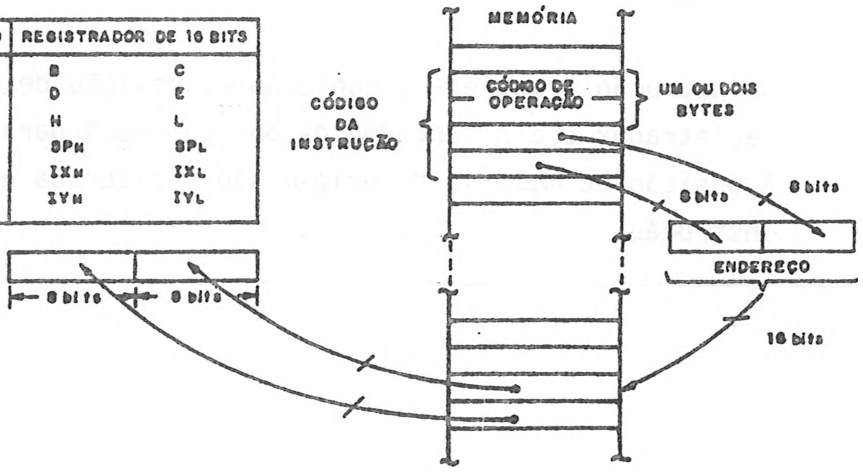
Estrutura

As instruções Registrador ← Memória são:

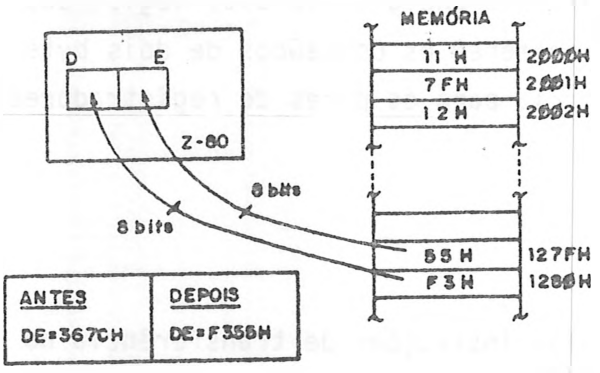
. LD BC,(nn)
. LD DE,(nn)
. LD HL,(nn)
. LD SP,(nn)
. LD IX,(nn)
. LD IY,(nn)

Procedimento das operações Registrador ← Memória

MNEMÔNICO DA INSTRUÇÃO	REGISTRADOR DE 16 BITS	
LD BC,(nn)	B	C
LD DE,(nn)	D	E
LD HL,(nn)	H	L
LD SP,(nn)	SPH	SPL
LD IX,(nn)	IXH	IXL
LD IY,(nn)	IYH	IYL



Instrução LD DE, (127FH)



ANTES	DEPOIS
DE=367CH	DE=F355H

Processo

O operando da esquerda representa sempre um par de registradores de 16 bits para onde é destinada a informação.

O da direita indica o endereço do qual o primeiro byte será transferido, sendo que o segundo byte a ser transferido está no endereço consecutivo.

Exemplo

. Dada a instrução LD BC,(4000H)

Mnemônico: LD

Operando: BC Destino

Operando: (4000H) Origem

C ← (4000H)

B ← (4001H)

A instrução transfere o conteúdo da posição de memória 4000H para o registrador C e o conteúdo da posição 4001 para o registrador B. A posição de memória de origem não é alterada após a execução da instrução.

Memória ← Registrador

Descrição

O conjunto de instruções de tránsferência de 16 bits Memória ← Registrador são instruções que transferem os conteúdos de pares de registradores BC, DE, HL, SP, IX, IY para a posição de memória especificada pelo operando (nn).

Símbolo

LD (nn),qq

O mnemônico que identifica estas instruções de transferência no Z80 é o LD, decorrente da palavra LOAD.

qq pode ser um dos seguintes pares de registradores internos à CPU: BC, DE, HL, SP, IX, IY.

(nn) especifica o endereço da memória.

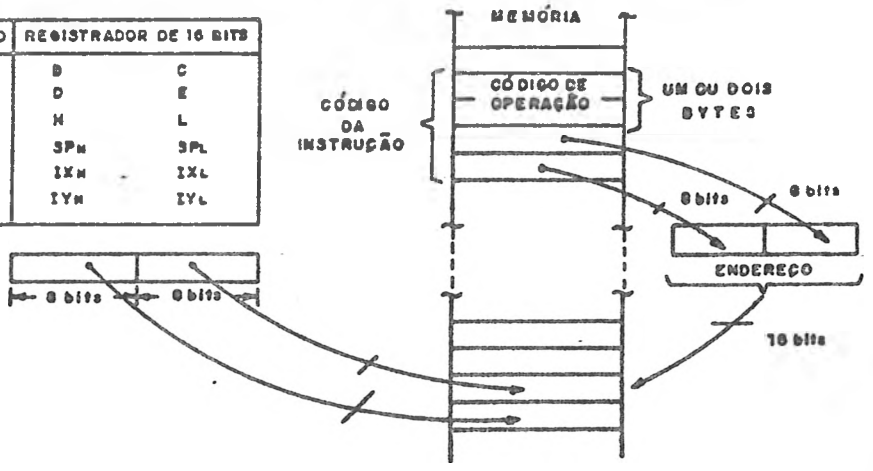
Estrutura

As instruções Memória ← Registrador são:

LD (nn),BC
LD (nn),DE
LD (nn),HL
LD (nn),SP
LD (nn),IX
LD (nn),IY

Procedimento das operações Memória ← Registrador

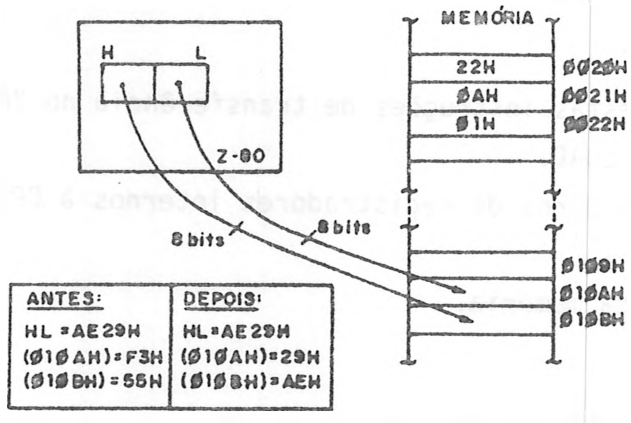
MNEMÔNICO DA INSTRUÇÃO	REGISTRADOR DE 16 BITS	
LD (nn),BC	B	C
LD (nn),DE	D	E
LD (nn),HL	H	L
LD (nn),SP	SPH	SPL
LD (nn),IX	IXH	IXL
LD (nn),IY	IYH	IYL



Processo

O operando da esquerda indica sempre uma posição de memória para onde é destinada a informação. O da direita indica o par de registradores do qual será transferido o conteúdo para a memória.

Instrução LD(010AH),HL



Exemplo

. Dada a instrução LD (4000),HL

Operando: (4000H) Destino

Operando: HL Origem

(4000H) ← L

(4001H) ← H

A instrução transfere o conteúdo do par de registrador HL para as posições de memória 4000H e 4001H.

O conteúdo dos registradores não se altera após a execução da instrução.

Registrador ← Constante

Descrição

O conjunto de instruções de transferência de 16 bits Registrador ← Constante são instruções que transferem uma constante de 16 bits para pares de registradores BC, DE ou HL e para registradores de 16 bits, SP, IX ou IY.

Símbolo

LD qq,(nn)

O mnemônico que identifica estas instruções de transferência no Z80 é o **LD**, decorrente da palavra **LOAD**.

qq pode ser um dos seguintes pares de registradores internos à CPU: BC, DE, HL, SP, IX, IY.

(nn) é uma palavra de 16 bits.

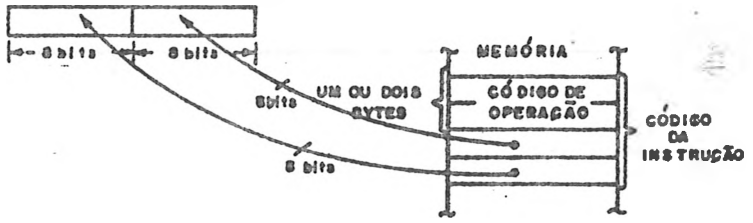
Estrutura

As instruções Registrador ← Constante são:

. LD BC,nn
. LD DE,nn
. LD HL,nn
. LD SP,nn
. LD IX,nn
. LD IY,nn

Procedimento das operações Registrador ← Constante

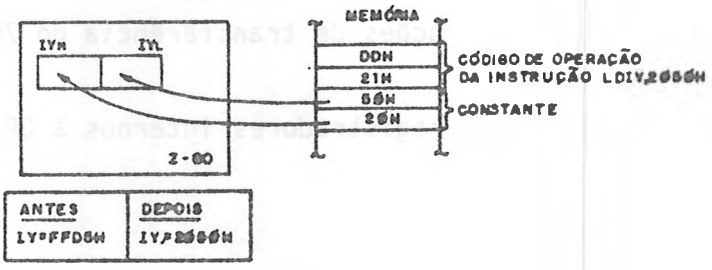
MNEMÔNICO DA INSTRUÇÃO	REGISTRADOR DE 16 BITS	
LD BC, nn	B	C
LD DE, nn	D	E
LD HL, nn	H	L
LD SP, nn	SPH	SPL
LD IX, nn	IXH	IXL
LD IY, nn	IYH	IYL



Processo

O operando da esquerda indica sempre um par de registradores ou um registrador de 16 bits para onde é destinada a informação.
O da direita é o dado que será transferido para o registrador especificado pelo operando.

Instrução LD IY,2050H



Exemplo

. Dada a instrução LD DE,2000H
Mnemônico: LD
Operando: DE Origem
Operando: 2000H Destino
E ← 00H
D ← 20H

Posição memória	Conteúdo
4000	11H
4001	00H
4002	20H

penúltimo byte

Observe-se que o **penúltimo byte** do código de construção é transferido para a parte menos significativa do registrador de 16 bits e o último byte para a parte mais significativa.

Pilha ← Registrador

Descrição

O conjunto de instruções de transferência de 16 bits Pilha ← Registrador são instruções que transferem o conteúdo dos registradores de 16 bits para o topo da pilha.

Símbolo

PUSH, qq

Nas instruções de PUSH, os operandos de 16 bits são transferidos para pilha.

Estes operandos podem ser os pares AF, BC, DE ou HL e os registros IX e IY.

Estrutura

As instruções do quinto grupo são:

PUSH BC
PUSH DE
PUSH HL
PUSH AF
PUSH IX
PUSH IY

Processo

Os endereços de memória acessados durante uma operação de PUSH são determinados da seguinte maneira:

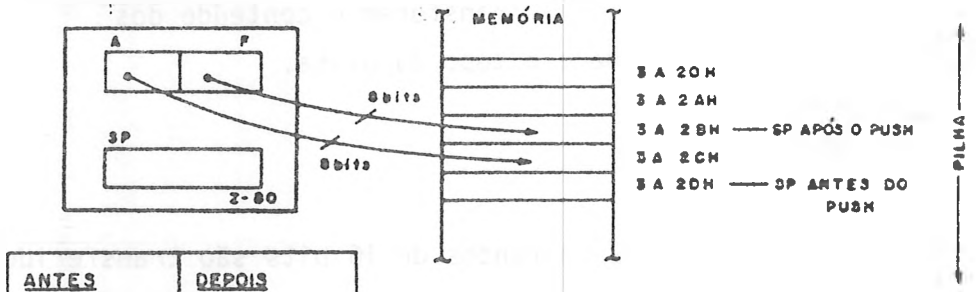
. Os oito bits mais significativos do operando de 16 bits (A, B, D, H, IXH e IYH) são armazenados no endereço de memória fornecido pelo conteúdo do SP decrementado de uma unidade.

. Os oito bits menos significativos do operando de 16 bits (F, C, E, L, IXL e IYL) são armazenados no endereço de memória fornecido pelo conteúdo do SP decrementado de duas unidades.

. O SP é automaticamente decrementado de duas unidades (SP ← SP-2).

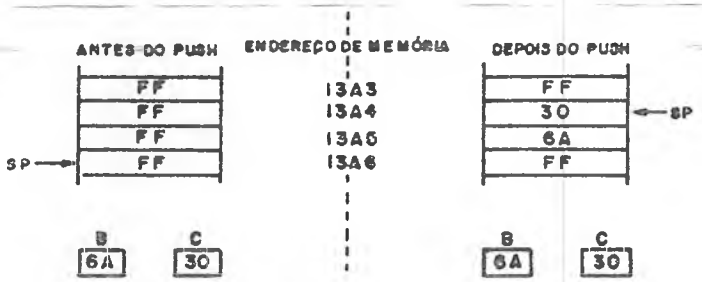
Exemplos

. Instrução PUSH AF



ANTES	DEPOIS
A=1FH	A=1FH
F=2AH	F=2AH
SP=3A24H	SP=3A22H
3A24H=FFH	3A24=2AH
3A23H=FFH	3A23=1FH

. Instrução PUSH BC



Registrador ← Pilha

Descrição

O conjunto de instruções de transferência de 16 bits Registrador ← Pilha são instruções que transferem o conteúdo do topo da pilha para um registrador de 16 bits.

Símbolo

POP qq

Nas instruções POP, os conteúdos do topo da pilha são transferidos para os operandos de 16 bits (qq).

Estes operandos podem ser os pares AF, BC, DE ou HL e os registros IX e IY.

Estrutura

As instruções do sexto grupo são:

POP BC
POP DE
POP HL
POP AF
POP IX
POP IY

Processo

Os endereços de memória acessados durante uma operação de POP são determinados da seguinte maneira.

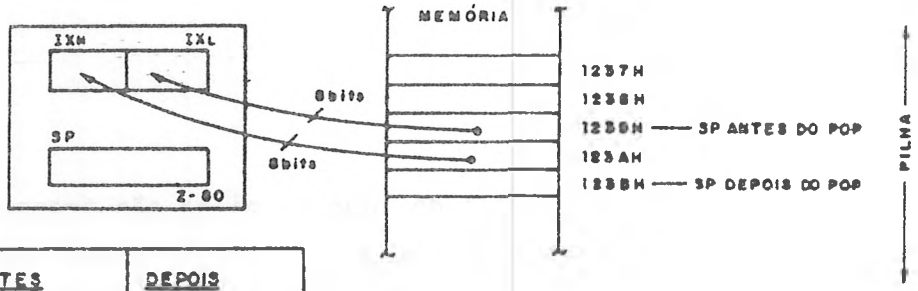
. O conteúdo da posição de memória endereçada pelo SP é transferido para a parte menos significativa do registro do Z80 (F, C, E, L, IXL, IYL).

. O conteúdo da posição de memória endereçada pelo SP incrementado de uma unidade é transferido para a parte mais significativa do registro do Z80 (A, B, D, H, IXH e IYH).

. O SP é automaticamente incrementado de duas unidades (SP ← SP+2).

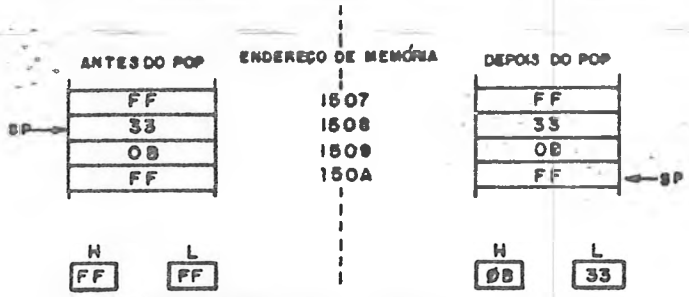
Exemplos

. Instrução POP IX



ANTES	DEPOIS
IX=ØFFØH	IX=Ø33ØH
SP=1235H	SP=1237H
(1235H)=30H	(1235H)=30H
(1236H)=ØBH	(1236H)=ØBH

. Instrução POP HL



Stack ← Registrador

Descrição

O conjunto de instruções de transferência de 16 bits Stack ← Registrador são instruções de transferência entre registradores de uso geral internos ao Z80.

Símbolo

LD SP,qq

O mnemônico que identifica estas instruções de transferência no Z80 é LD, decorrente da palavra LOAD.

SP (stack point) é o registrador de 16 bits cujo conteúdo indica sempre o endereço da pilha.

qq pode ser um dos seguintes pares de registradores internos à CPU: HL, IX, IY.

Estrutura

As instruções do primeiro grupo são:

LD SP, HL

LD SP, IX

LD SP, IY

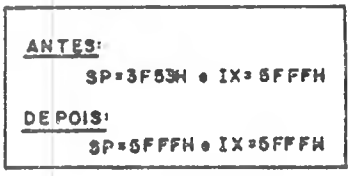
Essas instruções transferem para o SP, respectivamente, os conteúdos dos registradores HL, IX e IY.

Processo

O operando da esquerda indica sempre para onde é destinada (destino) a informação e o da direita indica de onde se origina (origem) a informação.

Exemplos

. Instrução LD SP, IX



. Instrução LD SP, HL

Mnemônico: LD

Operando: SP Destino

Operando: qq = HL Origem

A instrução transfere o conteúdo do par de registradores SP para o par HL.
Os registradores de origem, dos quais a informação é retirada, permanecem inalterados após a transferência.

INSTRUÇÕES DE TRANSFERÊNCIAS DE 16 BITS

MNEMÔNICO 8080	MNEMÔNICO Z-80	OPERAÇÃO SIMBÓLICA	Flags C	Z	P/V	S	N	H	Código de Máquina Binário 76	543	210	Nº de Bytes	Nº de M-ciclos	Nº de T-ciclos	SUB-GRUPO
OPERAÇÕES REGISTRADOR ← REGISTRADOR															
SPHL	LD SP,HL	SP ← HL	•	•	•	•	•	•	11	111	001	1	1	6	1
–	LD SP,IX	SP ← IX	•	•	•	•	•	•	11	011	101	2	2	10	
									11	111	001				
–	LD SP,IY	SP ← IY	•	•	•	•	•	•	11	111	101	2	2	10	
									11	111	001				
OPERAÇÕES REGISTRADOR ← MEMÓRIA															
LHLD end	LD HL,(nn)	H ← (nn + 1) L ← (nn)	•	•	•	•	•	•	00	101	010	3	5	16	2
									←	n	→				
									←	n	→				
–	LD dd,(nn)	dd_H ← (nn + 1) dd_L ← (nn)	•	•	•	•	•	•	11	101	101	4	6	20	
									01	dd1	011				
									←	n	→				
									←	n	→				
–	LD IX,(nn)	IX_H ← (nn + 1) IX_L ← (nn)	•	•	•	•	•	•	11	011	101	4	6	20	
									00	101	010				
									←	n	→				
									←	n	→				
–	LD IY,(nn)	IY_H ← (nn + 1) IY_L ← (nn)	•	•	•	•	•	•	11	111	101	4	6	20	
									00	101	010				
									←	n	→				
									←	n	→				
OPERAÇÕES MEMÓRIA ← REGISTRADOR															
SHLD end	LD (nn),HL	(nn + 1) ← H (nn) ← L	•	•	•	•	•	•	00	100	010	3	5	16	3
									←	n	→				
									←	n	→				
–	LD (nn),dd	(nn + 1) ← dd_H (nn) ← dd_L	•	•	•	•	•	•	11	101	101	4	6	20	
									01	dd0	011				
									←	n	→				
									←	n	→				
–	LD (nn),IX	(nn + 1) ← IX_H (nn) ← IX_L	•	•	•	•	•	•	11	011	101	4	6	20	
									00	100	010				
									←	n	→				
									←	n	→				
–	LD (nn),IY	(nn + 1) ← IY_H (nn) ← IY_L	•	•	•	•	•	•	11	111	101	4	6	20	
									00	100	010				
									←	n	→				
									←	n	→				
OPERAÇÕES REGISTRADOR ← CONSTANTE															
LXI dd, const.	LD dd,nn	dd ← nn	•	•	•	•	•	•	00	dd0	001	3	3	10	4
									←	n	→				
									←	n	→				
–	LD IX,nn	IX ← nn	•	•	•	•	•	•	11	011	101	4	4	14	
									00	100	001				
									←	n	→				
									←	n	→				
–	LD IY,nn	IY ← nn	•	•	•	•	•	•	11	111	101	4	4	14	
									00	100	001				
									←	n	→				
									←	n	→				
OPERAÇÕES PILHA ← REGISTRADOR															
PUSH p	PUSH qq	(SP – 2) ← qq_L (SP – 1) ← qq_H	•	•	•	•	•	•	11	qq0	101	1	3	11	5
–	PUSH IX	(SP – 2) ← IX_L (SP – 1) ← IX_H	•	•	•	•	•	•	11	011	101	2	4	15	
									11	100	101				
–	PUSH IY	(SP – 2) ← IY_L (SP – 1) ← IY_H	•	•	•	•	•	•	11	111	101	2	4	15	
									11	100	101				
OPERAÇÕES REGISTRADOR ← PILHA															
POP p	POP qq	qq_H ← (SP + 1) qq_L ← (SP)	•	•	•	•	•	•	11	qq0	001	1	3	10	6
–	POP IX	IX_H ← (SP + 1) IX_L ← (SP)	•	•	•	•	•	•	11	011	101	2	4	14	
									11	100	001				
–	POP IY	IY_H ← (SP + 1) IY_L ← (SP)	•	•	•	•	•	•	11	111	101	2	4	14	
									11	100	001				

NOTAS:

1 – "dd" simboliza qualquer um dos registradores de 16 bits: BC(00), DE(01), HL(10) ou SP(11).
2 – "qq" simboliza qualquer um dos registradores de 16 bits: BC(00), DE(01), HL(10) ou AF(11).
3 – Os 8 bits menos significativos de um registrador de 16 bits é representado por $(PAR)_L$ e os 8 bits mais significativos por $(PAR)_H$. Por exemplo, BC_L = C e AF_H = A.
4 – "p" na linguagem Assembly do 8080 é representado por B, D, H ou PSW.
5 – Notação dos flags: • = não afetado.

INSTRUÇÕES LÓGICAS E ARITMÉTICAS DE OITO BITS TA 31070

(3 páginas)

Descrição — O grupo de instruções lógicas e aritméticas de oito bits realiza operações de adição, subtração, "E" lógico, "OU lógico, "OU EXCLUSIVO", comparação, incrementação e decrementação e uma unidade.

Classificação — Essas isntuções classificam-se em dois subgrupos:

. Operações com dois operandos

. Operações com um operando

Operações com dois operandos

Descrição

Na operações com dois operandos, um operando é sempre o acumulador e o outro pode ser especificado por endereçamento imediato, registrador indireto através do par HL, indexado ou registrador.

Estrutura

Os mnemônicos das instruções que envolvem as operações com dois operandos são: ADD, ADC, SUB, SBC, AND, OR, XOR e CP. O segundo operando varia em função da técnica de endereçamento usada: r,n, (HL), (IX +d) e (IY +d).

Processo

As funções lógicas e aritméticas, processadas pelas operações dos dois operandos, operam o acumulador e outro operando especificado, e o resultado é sempre depositado no acumulador. Os bits de condição são atualizados em função do resultado a operação realizada.

Classificação

Podem-se classificar as operações com dois operandos em:

. Lógicas
  AND
  OR
  XOR
  CP

. Aritméticas
  adição
  subtração

Operações com um operando

Descrição

As operações com um operando envolvem as operações e incrementar, decrementar de uma unidade, o conteúdo e um registrador interno do Z80 ou uma posição e memória.

Estrutura

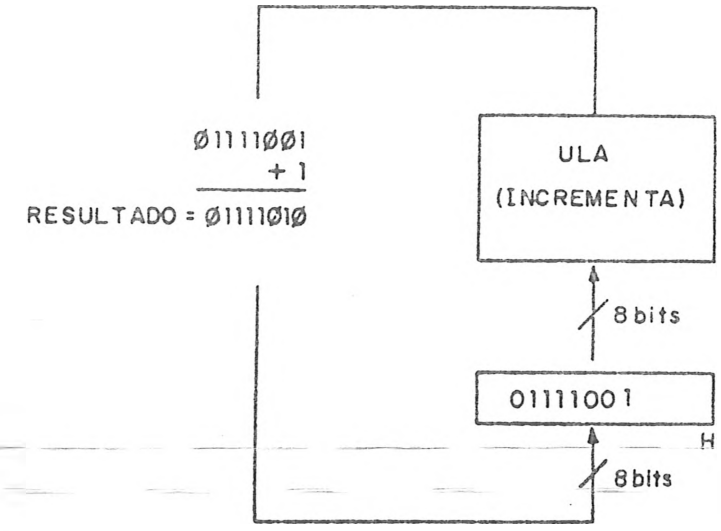
Os mnemônicos das instruções são INC e DEC e os operandos, r, (HL), (IX +d) e (IY +d).

Processo

O resultado destas operações é depositado no próprio operando. Os bits de condição flag Z, H e V são atualizados em função do resultado. No entanto, nas operações com um operando o carry-bit fica inalterado.

Exemplo

Se o registrador H contiver 79H, então após a execução da instrução INC H, ele passará a ter 7AH. A figura abaixo mostra o procedimento da instrução INC H.



OPERAÇÕES DE ADIÇÃO DE OITO BITS

TA 31080
(2 páginas)

Descrição

Operações de adição são instruções que executam a soma entre dois bytes especificados pela instrução.

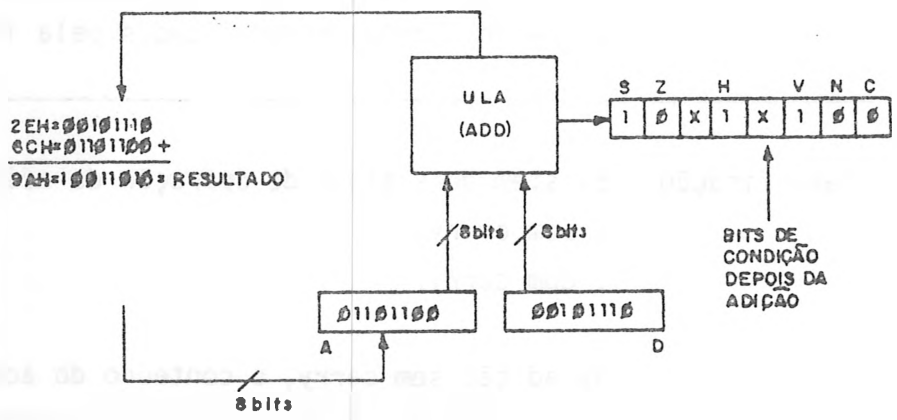
Classificação

Existem dois tipos de operação de adição:
. sem carry;
. com carry.

Na adição sem carry, o conteúdo do acumulador é adicionado ao segundo operando e o resultado é depositado no acumulador.

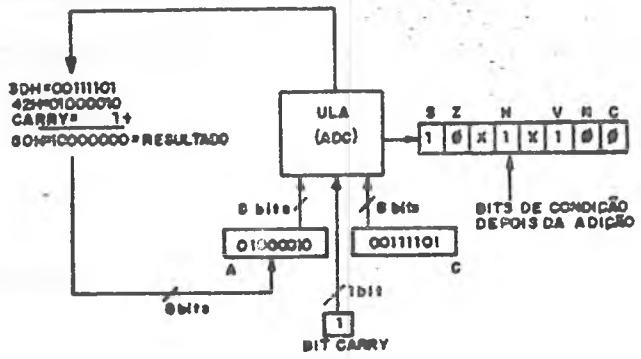
Na adição com carry, o conteúdo do acumulador é adicionado ao segundo operando e ao bit de carry, e este resultado é depositado no acumulador.
As operações de adição com carry permitem a realização de operações de adição de multiprecisão.

Exemplo — Adição sem carry ADD D



Se o registrador contiver 2EH e o acumulador contiver 6CH, então a instrução ADD D realizará a operação de adição conforme a figura acima.

Adição com carry ADC C



Se o registrador C contiver 3DH, o acumulador 42H e o carry-bit for igual a um, então a instrução ADC C realizará a operação de adição conforme mostra a figura acima.

Processo — **Nas operações de adição o registrador especificado é adicionado ao acumulador da aritmética complemento a dois.**

OPERAÇÕES DE SUBTRAÇÃO DE OITO BITS

TA 31090
(3 páginas)

Descrição

Operações de subtração são instruções que executam a subtração entre dois bytes especificados pela instrução.

Classificação

Existem dois tipos de subtração:
. sem borrow;
. com borrow.

Na subtração sem **borrow**, o segundo operando é subtraído do onteúdo do acumulador. Em contrapartida, na subtração **com o borrow**, do acumulador é subtraído o resultado da soma o segundo operando ao estado atual do bit-carry. As instruções de subtração com borrow são muito úteis na implementação e operações de subtração de multiprecisão.

Processo

Nas operações de subtração o conteúdo do registrador especificado é subtraído do conteúdo do acumulador na aritmética complemento a dois.

Procedimento

A ação do bit de carry nas operações de subtração difere da operação de adição. Na subtração, se não é gerado "vai um" do bit mais significativo, então o carry-bit é setado, indicando a ocorrência do "um emprestado" (borrow). Caso contrário, o carry-bit é ressetado.

Exemplo . Subtração sem o borrow

- Determinação do 2'S de 3EH

```
      3EH =  ØØ111111Ø
1'S de 3EH =  11000001
                    + 1
              ---------
2'S de 3EH =  11ØØØØ1Ø
```

- Operação de subtração

```
      3EH =  ØØ111111Ø
2'S de 3EH =  11000001Ø +
              ---------
              ØØØØØØØØ     = resultado Ø
```

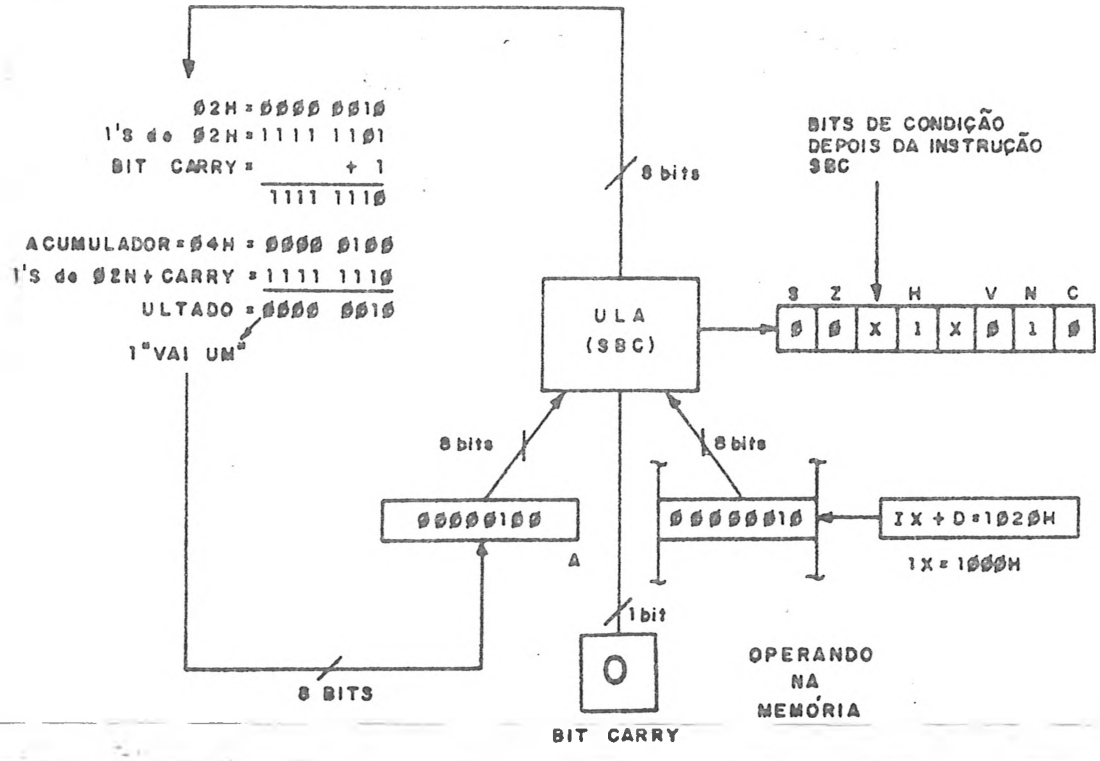
1 "vai um"

- Bits de condição após a operação

S	Z		H		V	N	C
Ø	1	x	1	x	Ø	1	Ø

Se o acumulador contiver 3EH, então a instrução SUB A subtrairá o acumulador dele mesmo, produzindo um resultado nulo.

- Subtração com o borrow



Se o conteúdo da posição de memória 1020H contiver 02H, o acumulador contiver 04H e o carry-bit for igual a um, então a instrução SBC A (IX + 20H) realizará a operação de subtração, conforme mostra a figura acima. Sabendo-se ainda que, no momento de sua execução, IX = 1000H.

Descrição

As três operações lógicas possíveis são "E", "OU" e "OU EXCLUSIVO". Ver tabelas abaixo.

Simbologia das operações lógicas

Função lógica	Mnemônico da instrução
"E"	AND
"OU"	OR
"OU EXCLUSIVO"	XOR

Tabelas-verdade

Operandos		Resultados
Operando 1	Operando 2	
0	0	0 0 0
0	1	0 1 1
1	0	0 1 1
1	1	1 1 0

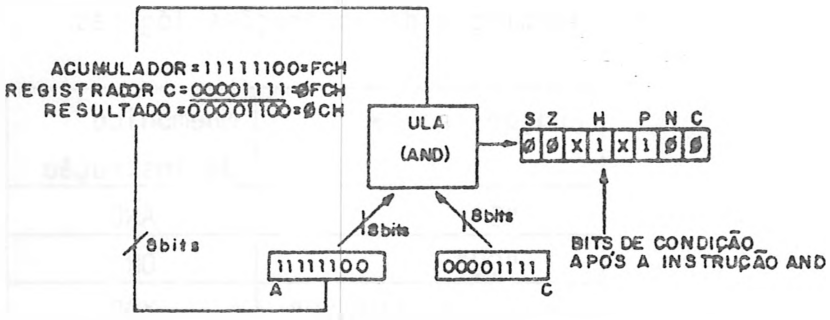
Comentário

As funções lógicas no Z80 operam os bits de mesma ordem em cada byte, realizando a operação bit a bit. Como um bit não afeta o seu adjacente, em conseqüência não há "vai um" (carry-bit = 0).

Exemplo

. Instrução and C

Se o acumulador contiver FCH e o registrador C contiver OFH, então a instrução AND C realizará as ações mostradas abaixo.



Observe-se que neste exemplo os quatro bits de mais alta ordem do acumulador ficam zerados e os quatro bits menos significativos ficam inalterados; na terminologia de processamento de dados, diz-se que os quatro bits mais significativos foram "mascarados".

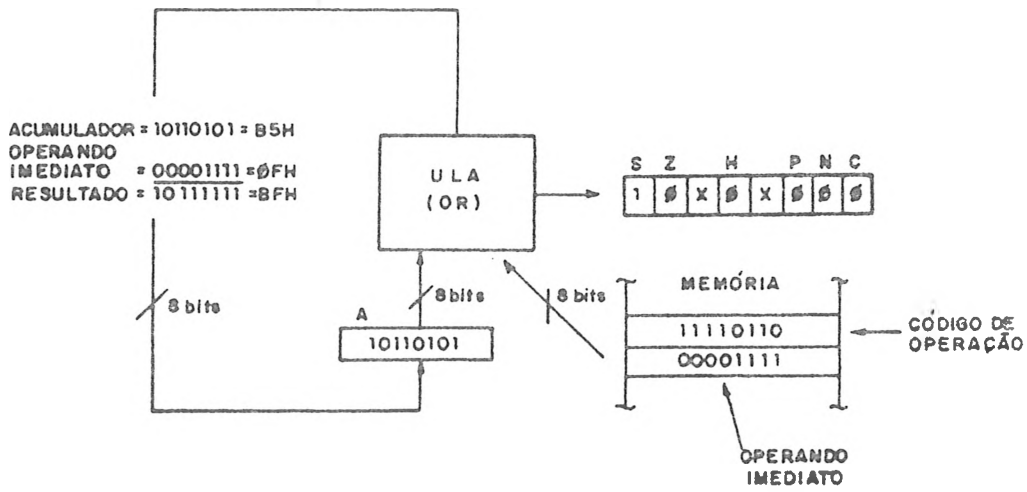
Comentário

De um modo geral, como o "E" lógico de um bit com "zero" dá "zero", o "E" lógico" de "um", com qualquer bit, mantém este bit inalterado. O "E" lógico é usado para gerar grupos de bits.

A operação de ressetar ou setar bits mantendo os demais inalterados é denominada mascarar (MASK).

. Instrução OR OFH

Se o registro A contiver 0B5H, então a instrução OR A,OFH se processará como na figura abaixo.



Observe-se neste exemplo que os quatro bits de mais baixa ordem do acumulador ficam setados e os quatro bits mais significativos ficam inalterados.
Pode-se dizer que os quatro bits mais significativos foram mascarados.

Comentário

De um modo geral, o "OU" lógico de um bit com "um" dá "um" e o "OU" lógico de "zero" com qualquer bit mantém este bit inalterado. O "OU" lógico é usado para setar grupos de bits.

. Instrução XOR A

Como o "OU EXCLUSIVO" de qualquer bit com ele mesmo produz um zero, então a instrução XRA A pode ser usada para zerar o conteúdo do acumulador.

. Instrução XOR B

Como o "OU EXCLUSIVO" de "um" lógico com qualquer bit é o seu complemento (0 XOR 1 = 1 e 1 XOR 1 = 0), então se o conteúdo do acumulador for FFH, a instrução XOR B produzirá o complemento "a um" do registrador B no acumulador.

NÚMERO DE ESTADOS

TA 31300
(2 páginas)

Descrição

Número de estados é o número de períodos de clock que a CPU leva para executar uma instrução.

Estrutura

Exemplos de instrução e seus números de estados.

Código de Instrução	Mnemônico	N. de bytes	N. de CM	N. de estados
78	LD A,B	1	1	4T
06 0A	LD B,0AM	2	2	7T
01 3 a 13	LD BC,123AH	3	3	10T
FD 22 3A 12	LD (123AH),IY	4	6	20T

Conhecendo esses parâmetros, podemos definir o tempo gasto na execução de um programa.

Exemplo

No programa acima temos quatro instruções, cada uma com o seu número de estados. Se somarmos essas instruções, teremos o tempo total gasto no programa.

Cálculo do tempo total

$T_{TOTAL} = T_1 + T_2 + T_3 + T_4$

$T_{TOTAL} = 4T + 7T + 10T + 20T$

$T_{TOTAL} = 41T$

Se o clock da CPU for igual a 1MHz teremos:

$$T = \frac{1}{f} = \frac{1}{1MHz} = 1\mu S$$

$\boxed{T = 1\mu S}$

$T_{TOTAL} = 41 \times 1\mu S$

$\boxed{T_{TOTAL} = 41\ \mu S}$

Comentário

Podemos observar no exemplo que quanto maior for a freqüência do clock da CPU, mais rápida será a execução do programa.

Para um clock de 2MHz temos:

$$T = \frac{1}{f} = \frac{1}{2MHz}$$

$\boxed{T = 0,5\ \mu S}$

Neste caso, teremos:

$T_{TOTAL} = 41 \times T = 41 \times 0,5\mu S$

$\boxed{T_{TOTAL} = 20,5\mu S}$

O tempo cai para a metade quando dobramos a freqüência de clock. Porém essa facilidade em diminuir o tempo de execução está limitada pela freqüência de clock máxima do Z80A.

CICLO DE MÁQUINA

TA 31290
(3 páginas)

Descrição

Ciclo de máquina é todo acesso que o microprocessador faz à memória ou a uma porta de saída/entrada. O número de ciclos de máquina possíveis varia de um a seis (CM).

Estrutura

Exemplo de instrução e seus ciclos de máquina.

Código da instrução	Mnemônicos	Número de bytes	Número de CM
78	LD A,B	1	1
06 0A	LD B,0AH	2	2
01 3A 13	LD BC, 123AH	3	3
FD 22 3A 12	LD (123AH), IY	4	6

Os ciclos de máquinas podem ser de:

. OPCODE FECTH (busca)

. Leitura

. Escrita

OPCODE FETCH

É um ciclo de leitura feito pelo microprocessador para buscar o código da operação. É sempre o primeiro ciclo de máquina.

Ciclo de leitura

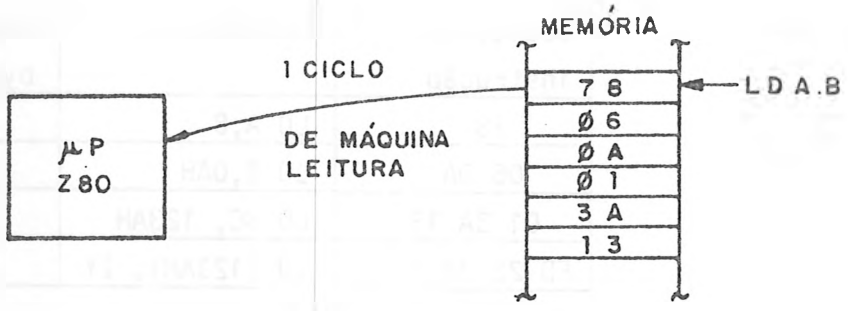
É feito para trazer da memória um dado, um operando imediato ou um dado de um I/O.

Ciclo de escrita

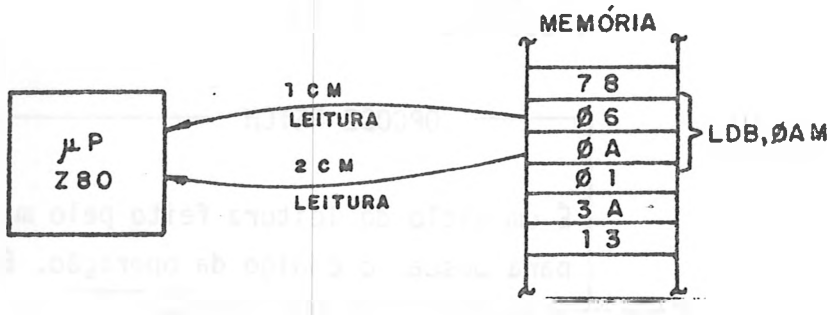
É feito pelo microprocessador para armazenar dados na memória ou em I/O.

Podemos observar melhor isto nos exemplos a seguir:

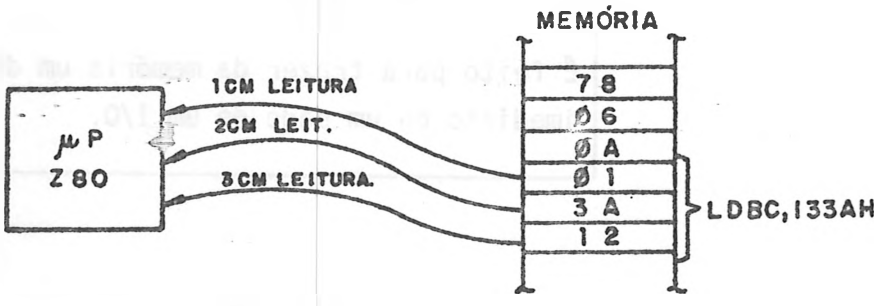
. Instrução LD A,B



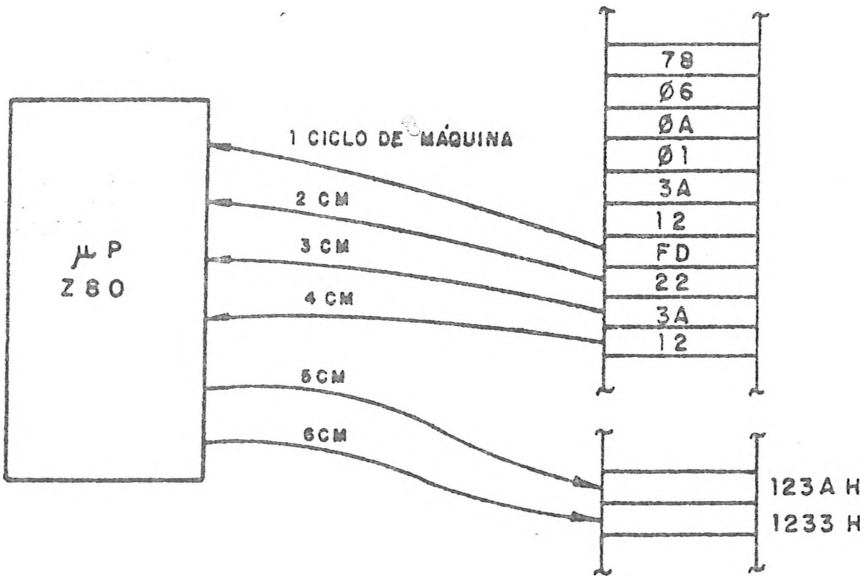
. Instrução LD B,0AH



. Instrução LD BC, 123AH



. Instrução LD (123AH), IY



Comentário

. Os ciclos de 1 a 4 são de leitura.

. Os ciclos 5 e 6 são de escrita.

INSTRUÇÕES ARITMÉTICAS DE 16 BITS

TA 31140
(2 páginas)

Descrição

O grupo de instruções aritméticas de 16 bits realiza operações de adição e subtração com os pares de registradores BC, DE ou HL, ou com registradores de 16 bits: IX, IY ou SP.

Classificação

As instruções deste grupo classificam-se em:
. Operações com dois operandos
. Operações com um operando

Comentário
Em todas as instruções dos dois subgrupos, os operandos só podem ser especificados pelo endereçamento por registrador.

Operações com dois operandos

Descrição

Este subgrupo opera dois números de 16 bits. Essas operações são realizadas entre pares de registradores e registradores de 16 bits.

Estrutura

Os mnemônicos deste subgrupo são ADD, ADC ou SBC. Os dois operandos são separados por uma vírgula. No operando da esquerda é armazenado o resultado da operação. O operando da direita é um registrador e 16 bits do Z80.

Classificação

As instruções aritméticas com dois operando classificam-se em adição e subtração.

Operações com um operando

Descrição

As operações com um operando envolvem as operações de incrementar e decrementar de uma unidade o conteúdo e um par de registradores ou registradores e 16 bits internos ao Z80.

Processo

As operações de incremento (INC) adicionam uma unidade o operando e as de decremento (DEC) subtraem dele uma unidade.
As operações são realizadas na aritmética complemento a dois.

Estrutura

Os mnemônicos das instruções são INC e DEC e os operandos são os pares de registradores BC, DE, HL e os registradores de 16 bits SP, IX e IY.

Comentário

As instruções INC e DEC não afetam os bits do registrador e flags.

OPERAÇÕES DE ADIÇÃO DE 16 BITS

TA 31150
(3 páginas)

Descrição

Operações de adição são instruções que executam a soma de dois registradores de 16 bits especificados pela instrução.

Classificação

As operações de adição podem ser **sem carry** (ADD) ou **com carry** (ADC) e são realizadas sempre na aritmética complemento dois.

Estrutura

Os resultados das operações de adição sem carry são depositados no par HL (ADD HL, ss),no registrador IX (ADD IX, pp) ou no registrador IY (ADD IY, rr).

. Na instrução ADD HL,ss o par HL pode ser adicionado aos pares BC e DE ao registrador SP e ao próprio HL.
. Na instrução ADD IX,pp, o registrador IX pode ser adicionado os pares BC e DE, ao registrador SP e ao próprio IX.
. Na instrução ADD IY,rr, o registrador IY pode ser adicionado os pares BC e DE, ao registrador SP e ao próprio IY.

Comentário

As operações de adição sem carry só permitem testar o carry-bit dos seus resultados.

Processo

Nas operações de adição, sem carry, o par de registradores especificado é adicionado ao par de registradores HL ou aos registradores de 16 bits IX ou IY, onde será armazenado o resultado da operação.
Já nas operações de adição com carry, o par de registradores especificado é adicionado ao par de registradores HL, ou aos registradcres de 16 bits IX ou IY, e ao carry. O resultado é armazenado no par de registradores HL.

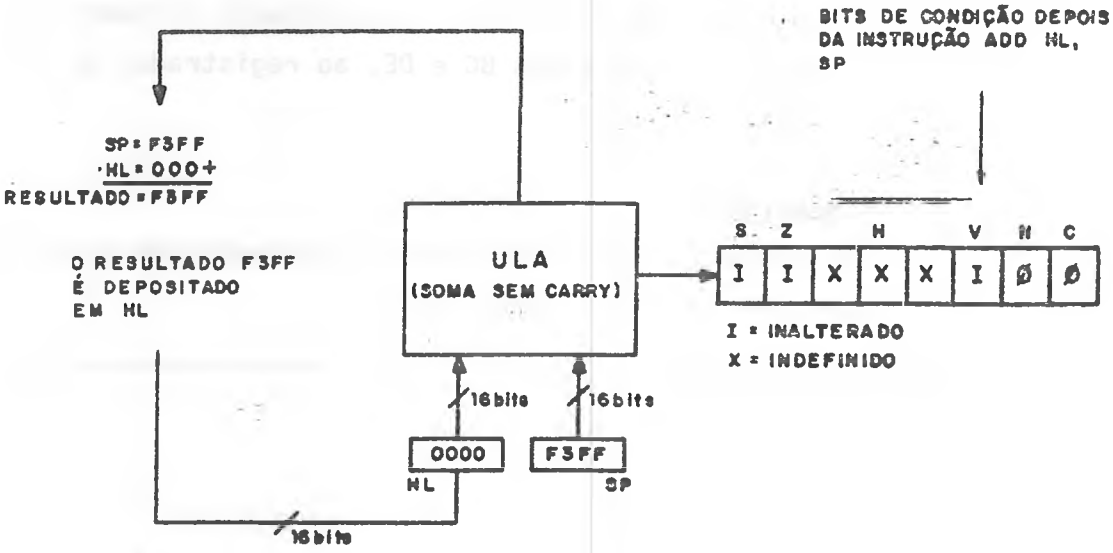
Observação

O carry que está sendo adicionado é resultado de uma operação anterior.

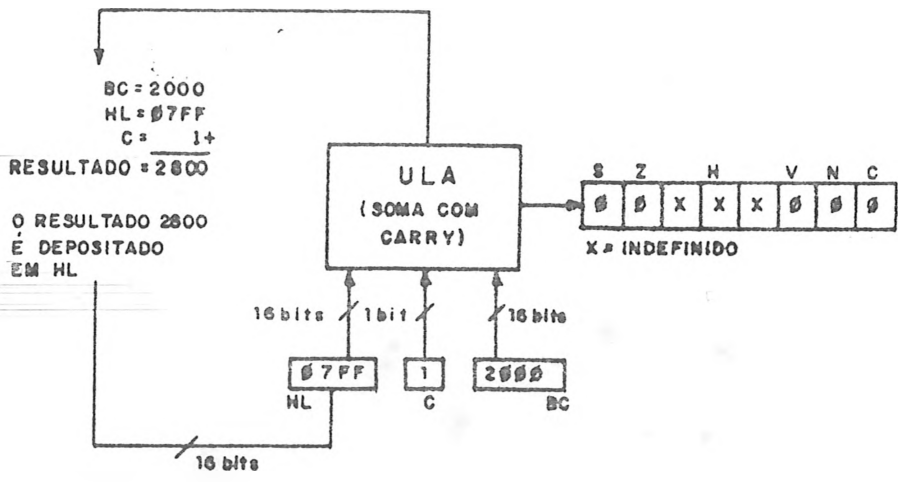
Exemplos

. Instrução ADD HL,SP

Se o registrador SP contiver F3FFH e o registrador HL HL = 0000H, então a instrução ADD HL,SP realizará a operação de adição, conforme a figura abaixo.



. Instrução ADC HL,BC



LINGUAGEM ASSEMBLY

TE 70190
(5 páginas)

Descrição

A linguagem ASSEMBLY foi criada para facilitar o manuseio das instruções e possibilitar o desenvolvimento de programas de todos os tipos: desde uma simples rotina para efetuar uma multiplicação, até um sofisticado programa para controle de dispositivos para a indústria. Na programação em ASSEMBLY utilizam-se abreviaturas, que são mais fáceis de serem memorizadas que as instruções da linguagem de máquinas, expressas em hexadecimal. Essas abreviaturas são denominadas mnemônicos.

Comentário

As siglas de empresas ou entidades são mnemônicos.

Exemplo

Mnemônico	Significado
DER	Departamento de Estradas de Rodagem
SENAI	Serviço Nacional de Aprendizagem Industrial
LD	"LOAD" = carregar

Estrutura A linguagem ASSEMBLY é dividida em quatro campos:

. Label
. Código de operação
. Operando/endereço
. Comentário

Label	Código de operação	Operando/endereço	Comentário da instrução
Início	LD	SP, FIM	;inicialização do stack pointer
	IM	2	;modo de interrupção 2

OPERAÇÕES DE COMPARAÇÃO (CP) TA 31110

(3 páginas)

Descrição

Operações de comparação são instruções que efetuam a comparação entre dois bytes especificados pela instrução e dizem se esses bytes são iguais ou se um é maior ou menor que o outro.

Processo

A instrução CP compara o segundo operando com o acumulador. Esta comparação se faz subtraindo internamente o operando do acumulador, deixando ambos inalterados, e atualizando os bits de condição em função do resultado.

Estrutura

O flag de zero (Z = 1) indica que as quantidades são iguais, enquanto "z = 0" indica que as quantidades são diferentes.

Como se trata de uma subtração, o bit carry fica setado se não houver "vai um" do bit mais significativo do resultado. Se os operandos forem números positivos, este resultado (C = 1) significa que o segundo operando é maior que o acumulador.

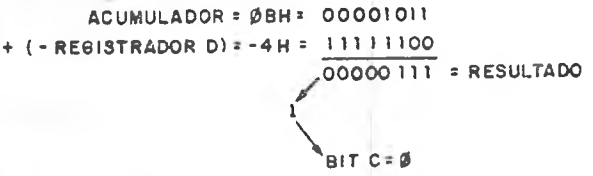
Análise dos bits do flag de zero e carry, assumindo-se a comparação de números positivos.

Z = 1 —— A = 2º OPERANDO
Z = Ø —— A ≠ 2º OPERANDO
C = Ø —— A ≥ 2º OPERANDO
C = 1 —— A < 2º OPERANDO

ANÁLISE DOS BITS "Z" e "C"

Exemplo

Se o conteúdo do acumulador for 0BH e o registrador D contiver 04H, então a instrução CP D realizará a operação de subtração interna mostrada abaixo.



Após a subtração A = 0BH, D = 04H e bit C = 0, indicando que o conteúdo de D é menor ou igual ao conteúdo do acumulador.

Para saber com certeza se os bytes são iguais, é necessário testar o flag de zero e verificar se ele é igual a um.

INSTRUÇÕES LÓGICAS E ARITMÉTICAS DE 8 BITS

MNEMÔNICOS 8080/85	MNEMÔNICOS Z-80	OPERAÇÃO SIMBÓLICA	FLAGS C	Z	P/V	S	N	H	Código de Máquina Binário 76	543	210	Nº de Bytes	Nº de M-Ciclos	Nº de T-Ciclos	SUB-GRUPO
OPERAÇÕES ARITMÉTICAS DE 8 BITS COM DOIS OPERANDOS															
ADDr	ADD r	A ← A + r	↕	↕	V	↕	0	↕	10	[000]	r	1	1	4	
ADI n	ADD n	A ← A + n	↕	↕	V	↕	0	↕	11	[000]	110	2	2	7	
									←	n	→				
ADD M	ADD (HL)	A ← A + (HL)	↕	↕	V	↕	0	↕	10	[000]	110	1	2	7	
–	ADD (IX + d)	A ← A + (IX + d)	↕	↕	V	↕	0	↕	11	011	101	3	5	19	
									10	[000]	110				
									←	d	→				
–	ADD (IY + d)	A ← A + (IY + d)	↕	↕	V	↕	0	↕	11	111	101	3	5	19	
									10	[000]	110				
									←	d	→				
ADC, ADI	ADC s	A ← A + s + CY	↕	↕	V	↕	0	↕		[001]					1
SUB, SUI	SUB s	A ← A – s	↕	↕	V	↕	1	↕		[010]					
SBB, SBI	SBC s	A ← A – s – CY	↕	↕	V	↕	1	↕		[011]					
OPERAÇÕES LÓGICAS DE 8 BITS COM DOIS OPERANDOS															
ANA, ANI	AND s	A ← A ∧ s	0	↕	P	1	0	1		[100]					
ORA, ORI	OR s	A ← A ∨ s	0	↕	P	1	0	1		[110]					
XRA, XRI	XOR s	A ← A ⊕ s	0	↕	P	1	0	1		[101]					
CMP, CPI	CP s	A – s	↕	↕	V	↕	1	↕		[111]					
OPERAÇÕES ARITMÉTICAS COM UM OPERANDO															
INR r	INC r	r ← r + 1	•	↕	V	↕	0	↕	00	r	[100]	1	1	4	
INR M	INC (HL)	(HL) ← (HL) + 1	•	↕	V	↕	0	↕	00	110	[100]	1	3	11	
–	INC (IX + d)	(IX + d) ← (IX + d) + 1	•	↕	V	↕	0	↕	11	011	101	3	6	23	
									00	110	[100]				
									←	d	→				2
–	INC (IY + d)	(IY + d) ← (IY + d) + 1	•	↕	V	↕	0	↕	11	111	101	3	6	23	
									00	110	[100]				
									←	d	→				
DCR	DEC d	d ← d – 1	•	↕	V	↕	1	↕			[101]				

NOTAS:

1 – "r" simboliza qualquer um dos registradores de 8 bits: B (000), C (001), D (010), E (011), H (100), L (101), A (111).

2 – "s" simboliza r, n, (HL), (IX + d) e (IY + d), conforme mostrado nas instruções ADD.

3 – Os bits no interior dos retângulos das instruções do subgrupo 1 substituem o $\phi\phi\phi$ nas instruções ADD.

4 – O "d" na instrução DEC simboliza r, (HL), (IX + d) e (IY + d), conforme mostrado nas instruções INC.

5 – Para se obter os códigos de máquina binários das instruções DEC, basta substituir os bits $1\phi\phi$ no interior dos retângulos nas instruções INC por $1\phi1$.

6 – A letra V indica que o flag P/V fornece o overflow do resultado, e a letra P a paridade.

7 – V = 1 indica a ocorrência de overflow
P = 1 indica que a paridade do resultado é par

8 – Notação dos flags: • = não afetado, 0 = resetado, 1 = setado.
X = indefinido.
↕ = o flag é afetado em função do resultado da operação.

NÚMERO DE BYTES

TA 31280
(1 página)

Descrição

O número de bytes refere-se ao código da instrução mais o operando que no Z80 pode assumir um, dois, três e quatro bytes por instrução.

Estrutura

Exemplos de instrução com um, dois, três e quatro bytes.

Código da instrução	Mnemônicos	Número de bytes
78	LD A,B	1
06 0A	LD B,0AH	2
01 3A 12	LD BC,123AH	3
FD 22 3A 12	LD (123AH),IY	4

Quanto maior o número de bytes mais demorada será a execução do programa.

PROGRAMAÇÃO EM LINGUAGEM DE MÁQUINA DO Z80 TA 31270

(2 páginas)

Descrição

Programação em linguagem da máquina do Z80 é feita utilizando-se as intruções do Z80 e os seus códigos de máquina correspondentes. A seqüência de códigos de máquinas é denominada programação em linguagem de máquina.

Estrutura

O código de cada instrução encontra-se em uma posição de memória.

Abaixo pode-se ver um exemplo da multiplicação do conteúdo do acumulador por 5.

Endereço	Código de máquina	Mnemônico
4000	47	LD B,A
4001	87	ADD A,A
4002	87	ADD A,A
4003	80	ADD A,B
4004	C9	RET

Procedimento

Os códigos de cada instrução devem ser carregados em seqüência na memória para que o ~~micro~~processador execute o programa.

A inversão de um desses códigos leva a erros na programação.

Como a programação é feita pela digitação direta dos códigos de máquina, a possibilidade de erro é maior, exigindo assim do programador atenção muito grande durante a programação.

Características A programação em linguagem de máquina tem como característica principal sua alta velocidade de execução.

Mas para que o programa esteja realmente otimizado, são necessários alguns cuidados quanto aos números de:

. instruções;
. bytes do programa;
. ciclo de máquina;
. estados.

Comentário

Todo bom programa deve ter esses itens muito bem dimensionados mas, para conseguir isto, é necessário que o programador se exercite elaborando inicialmente pequenos programas.

Descrição Entradas e saídas são instruções que possibilitam a comunicação do processador central com o mundo externo.

Estrutura

Instruções de entrada

Instrução	Função
IN A, (n)	A ← (n)
	O operando n é colocado na metade inferior da via de endereços para selecionar o dispositivo de entrada/saída (E/S) em um dos 256 endereços possíveis. O conteúdo do acumulador também aparece na metade superior da via de endereço neste tempo. Um byte da porta selecionada é, então, colocado na via de dados e escrito no acumulador do processador.
INr, (C)	r ← (C) O conteúdo do registrador (C) é colocado na metade inferior da via de endereços para selecionar uma das 256 portas de entrada/saída possíveis. O conteúdo do registrador (B) é colocado na metade superior da via de endereços neste tempo. Um byte da porta selecionada é, então, colocado na via de dados e escrito no registrador r do processador.

INI	(HL) ← (C), B ← B - 1, HL ← HL + 1 O conteúdo do registrador (C) é colocado na metade inferior da via de endereços para selecionar uma das 256 portas possíveis. O registrador B pode ser usado como contador e o seu conteúdo é colocado na metade superior da via de endereços. Um byte da porta selecionada é colocado na via de dados e escrito na localização de memória correspondente. Finalmente, o contador é decrementado e o par de registradores HL é incrementado.
INIR	(HL) ← (C), B ← B - 1, HL ← HL + 1 O conteúdo do registrador (C) é colocado na metade inferior da via de endereços para selecionar uma das 256 portas possíveis. O registrador B é usado como contador e seu conteúdo é colocado na metade superior da via de endereços. Um byte é selecionado, colocado na via de dados e escrito no processador central. O conteúdo do par de registradores (HL) é colocado no endereço e o byte de entrada é escrito na localização de memória correspondente. O contador é decrementado e o par de registradores HL é incrementado. Quando B chegar a 0, a instrução terminará. Se B não chegar a 0, o PC será decrementado de dois e a instrução será repetida. As interrupções são reconhecidas após cada transferência.

IND	(HL) ← (C), B ← B - 1, HL ← HL - 1 O conteúdo do registrador (C) é colocado na metade inferior da via de endereço para selecionar um componente de E/S. O registrador B pode ser usado como contador e o seu conteúdo é colocado na metade superior da via de endereços. Um byte da porta selecionada é colocado na via de dados e escrito no processador. O conteúdo do par de registradores (HL) é colocado na via de endereços e o byte de entrada é, então, escrito na localização de memória correspondente. Finalmente o contador e o par de registradores são decrementados.
INDR	(HL) ← (C), B ← B 1, HL ← HL - 1 O conteúdo do registrador (C) é colocado na metade inferior da via de endereços para selecionar o dispositivo de E/S. O registrador B é usado como contador e seu conteúdo é colocado na metade superior da via de endereços. Um byte da porta selecionada é colocado na via de dados e escrito no processador. O conteúdo do par de registradores (HL) é colocado na via de endereços e o byte de entrada é escrito na localização da memória correspondente. O par de registradores HL e o contador B são decrementados. Quando B chegar a 0, a instrução terminará. Se B não chegar a 0, o PC será decrementado de dois e a instrução será repetida. As interrupções são reconhecidas após cada transferência de dados.

TA 31250

Instruções de saída

OUT(n), A	(n) ← A O operando n é colocado na metade inferior da via de endereços para selecionar o dispositivo de E/S. O conteúdo do acumulador aparece na metade superior da via de endereços. Então o byte contido no acumulador é colocado na via de dados e escrito no dispositivo selecionado.
OUT(C), r	(C) ← r O conteúdo do registrador (C) é colocado na metade inferior da via de endereços para selecionar o dispositivo de E/S. O conteúdo do registrador B é colocado na metade superior da via de endereços. O byte contido no registrador r é colocado na via de dados e escrito no dispositivo de E/S.
OUTI	(C) ← (HL), B ← B - 1, HL ← HL + 1 O conteúdo do par de registradores (HL) é colocado na via de endereços para selecionar uma localização de memória. O byte contido nesta localização de memória é temporariamente guardado no processador. Depois de decrementado o registrador B, o conteúdo do registrador (C) é colocado na metade inferior da via de endereços para selecionar o dispositivo de E/S. O registrador B pode ser usado como contador, e seu valor decrementado é colocado na metade inferior da via de endereços. O byte a ser enviado é colocado na via de dados e escrito no dispositivo selecionado. Finalmente o par de registradores Hl é incrementado.

OUTIR	(C) ← (HL), B ← B - 1, HL ← HL + 1 O conteúdo do par de registradores (HL) é colocado na via de endereços para selecionar a localização de memória. O byte contido nesta localização de memória é temporariamente guardado no processador. Depois que o contador é decrementado (registrador B), o conteúdo do registrador (C) é colocado na metade inferior da via de endereços para selecionar o dispositivo de E/S. O registrador B pode ser usado como contador e o seu valor é colocado na metade superior da via de endereços. O byte a ser enviado é colocado na via de dados e escrito no dispositivo de E/S selecionado; logo, o par de registradores HL é incrementado. Se o registrador B não chegar a 0, o PC será decrementado por dois e a instrução será repetida. Se B chegar a 0, a instrução estará terminada. As interrupções são reconhecidas após cada transferência de dados.
OUTD	(C) ← (HL), B ← B 1, HL ← HL - 1 O conteúdo do par de registradores (HL) é colocado na via de endereços para selecionar uma localização de memória. O byte contido nesta localização de memória é temporariamente guardado no processador. Depois que o contador é decrementado, o conteúdo do registrador (C) é colocado na metade inferior da via de endereço para selecionar o dispositivo de E/S. O byte a ser enviado é colocado na via de dados e escrito no dispositivo E/S selecionado. Finalmente o par de registradores HL é decrementado.

OUTDR	(C) ← (HL), B ← B - 1, HL ← HL - 1 O conteúdo do par de registradores (HL) é colocado na via de endereços para selecionar uma localização de memória. O byte contido nesta localização de memória é temporariamente guardado no processador. Depois que o contador é decrementado, o conteúdo do registrador C é colocado na metade inferior da via de endereços para selecionar o dispositivo de E/S. O registrador B pode ser usado como contador e, depois de decrementado seu valor, colocado na metade superior da via de endereços. O byte a ser enviado é, então, colocado na via de dados e escrito no dispositivo selecionado. O par de registradores HL é, então, decrementado. Se o registrador B não chegar a 0, o PC será decrementado de dois e a instrução será repetida. Se o registrador B chegar a 0, então a instrução estará terminada. As interrupções são reconhecidas após cada transferência de dados.

MODOS DE INTERRRUPÇÃO

TA 31190
(7 páginas)

Descrição

O microprocessador Z80 possui três modos de interrupção distintos. Esses modos são habilitados através do uso das instruções de modo de interrupção.

Classificação

Os modos de interrupção são:

. IM0
. IM1
. IM2

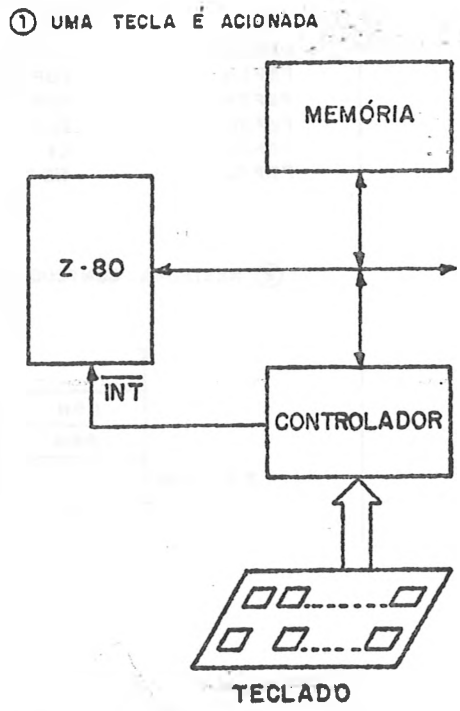
IM0

Descrição

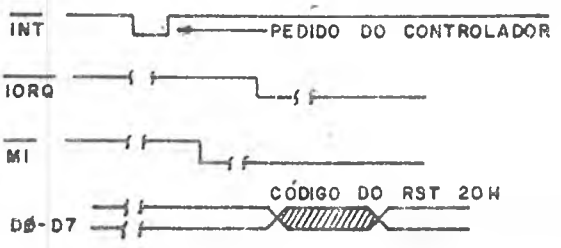
Quando o Z80 for ressetado, se nenhum modo for definido para operação da interrupção INT, o modo 0 será automaticamente assumido.
Este modo também pode ser assumido através da instrução IM0.

Exemplo

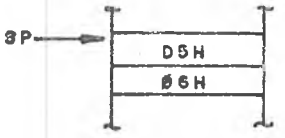
Modo 0 - Instrução IM0



② O CONTROLADOR FAZ O PEDIDO DE INTERRUPÇÃO E FORNECE O RST 20H



③ SALVA NA PILHA O CONTEÚDO DO PC (EXEMPLO: Ø6D5H)



④ O PROCESSAMENTO PROSSEGUE NO ENDEREÇO ØØ2ØH

ENDEREÇO	INSTRUÇÃO
ØØ2ØH	JP FEEØH

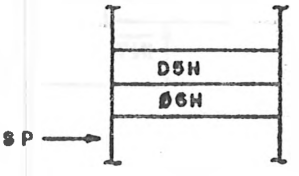
⑤ ROTINA DE TRATAMENTO DA INTERRUPÇÃO DE TECLADO

```
FEEØH    PUSH AF
FEE1H    PUSH BC
FEE2H    PUSH DE
FEE3H    PUSH HL

         OUTROS PROCESSAMENTOS

FEFØH    POP HL
FEF1H    POP DE
FEF2H    POP BC
FEF3H    POP AF
FEF4H    EI
FEF5H    RETI
```

⑥ RESTAURA CONTEÚDO DO PC (Ø6D5H)



Funcionamento do exemplo anterior

. Ocorrência da interrupção (INT = 0).

. No final da instrução, que se encontra em fase de execução, a interrupção é reconhecida.

. O Z80 sinaliza o reconhecimento da interrupção, ativando as linhas $\overline{IORQ}$ e $\overline{M1}$ ($\overline{IORQ}$ = 0 e $\overline{M1}$ = 0).

. O dispositivo externo, ao identificar $\overline{IORQ}$ e $\overline{M1}$ ativos, deposita o código de uma instrução na barra de dados, normalmente um restart (RST). Durante a busca desta instrução, o Z80 insere, automaticamente, dois estados de wait.

. A instrução (RST), que define a localização da rotina de tratamento de interrupção, é executada.

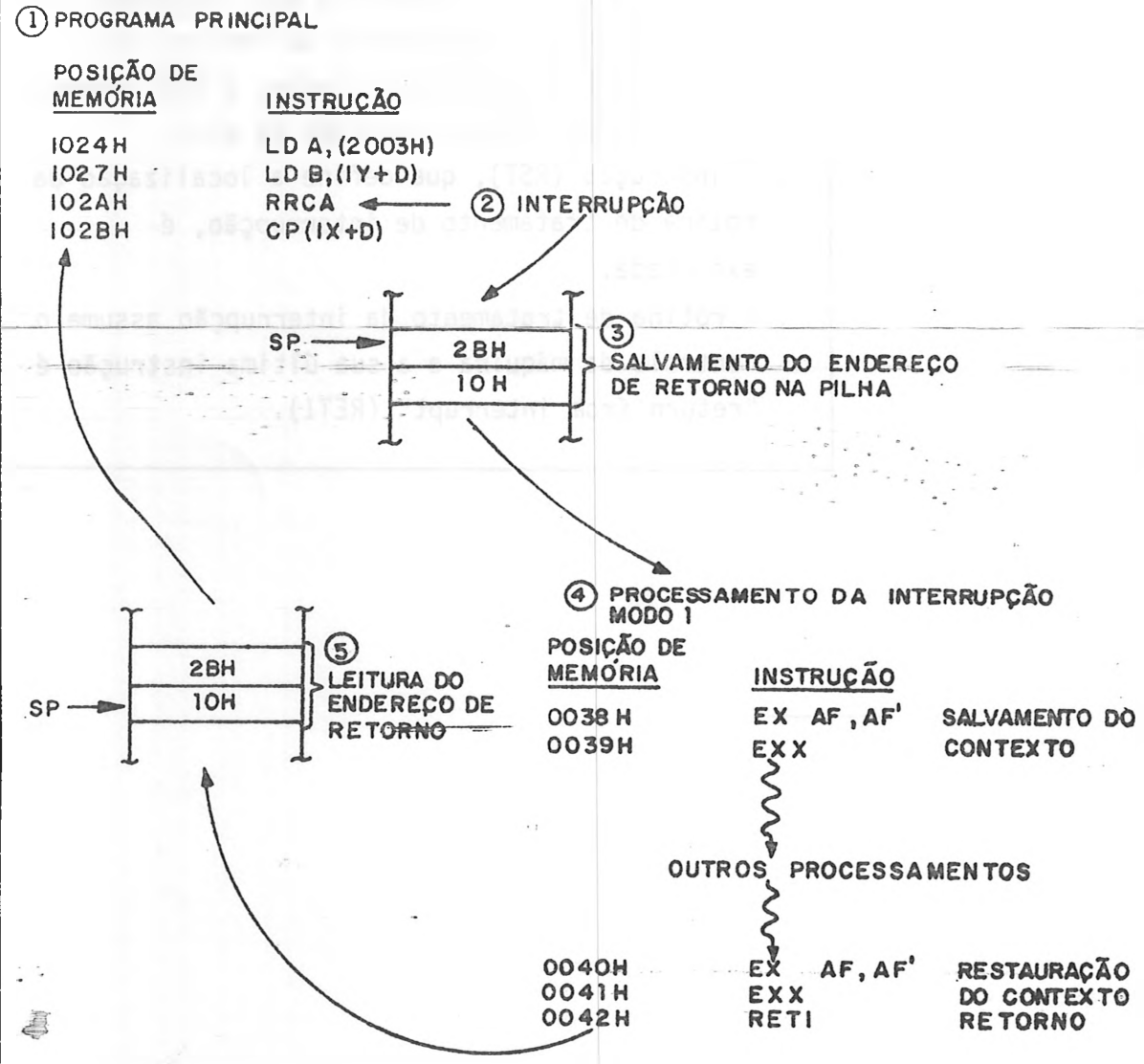
. A rotina de tratamento da interrupção assume o controle da máquina e a sua última instrução é "return from interrupt" (RETI).

IM1

Descrição

Quando a entrada $\overline{INT}$ fica ativa ($\overline{INT}$ = 0), a interrupção é reconhecida no final da instrução que está em processo. O Z80, efetivamente, executa uma instrução de restart (RST) para a posição de memória 0038H.

Exemplo



Funcionamento do exemplo anterior

. Ocorrência da interrupção ($\overline{INT}$ = 0)

. No final da instrução que se encontra em fase de execução, a interrupção é reconhecida.

. O Z80 sinaliza o reconhecimento da interrupção, ativando as linhas $\overline{IORQ}$ e $\overline{M1}$ ($\overline{IORQ}$ = 0 e $\overline{M1}$ = 0).

. O Z80 reconhece o sinal de ($\overline{INT}$ = 0) e deposita na barra de endereços o valor (0038H), gerando logo depois um $\overline{MRD}$ (leitura de memória).

. A instrução (RST7), que define a localização da rotina de tratamento e interrupção, é executada.

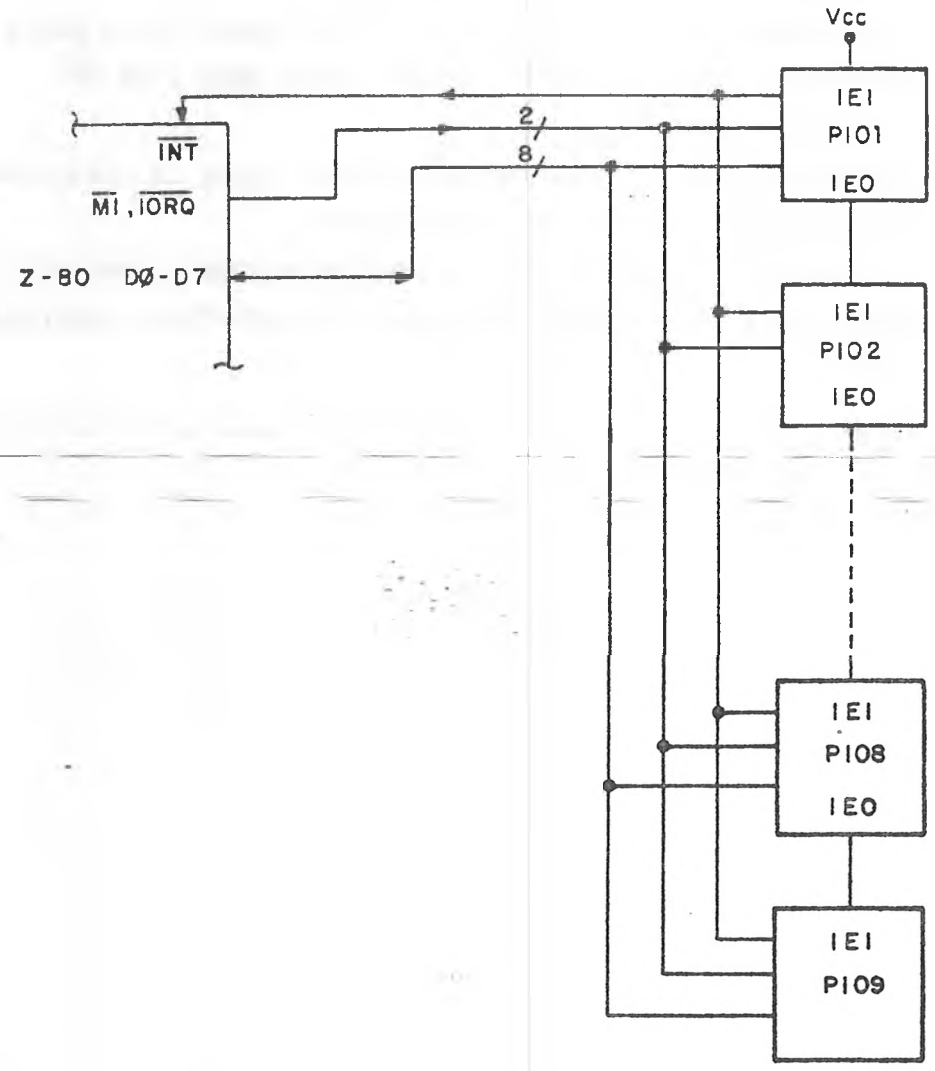
. A rotina de tratamento de interrupção assume o controle da memória e a sua última instrução é "return from interrupt" (RETI).

IM2

Descrição

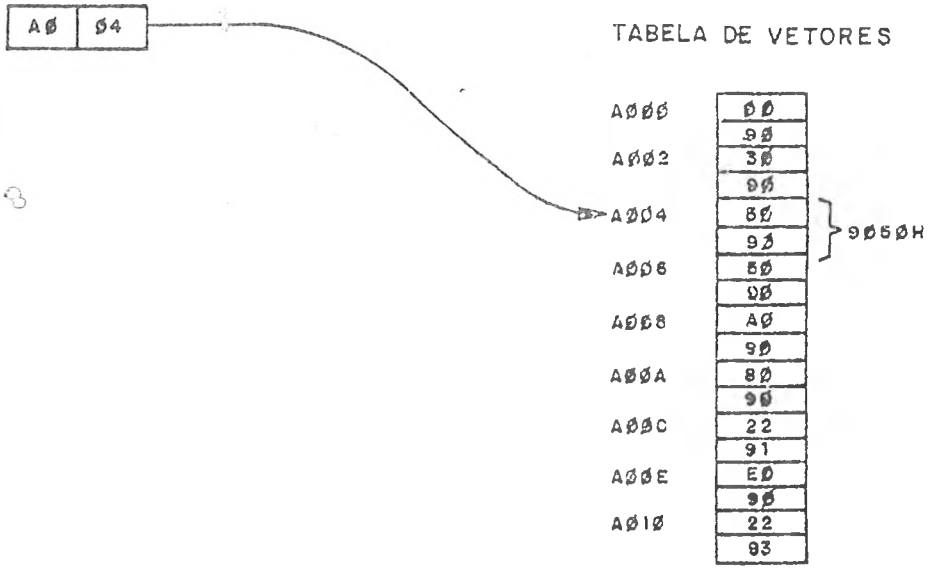
Este modo de operação permite até 128 interrupções vetoradas para posições de memórias pré-definidas pelo programador.

Exemplo



Funcionamento do exemplo anterior

- PIO8 solicita pedido ativando $\overline{INT}$ ($\overline{INT}$ = 0).
- O Z80 aceita pedido ativando $\overline{M1}$ e $\overline{IORQ}$ ($\overline{M1}$ e $\overline{IORQ}$ = 0).
- PIO8 fornece, via D0 - D7, oito bits do vetor (04H).
- O Z80 forma o vetor com o valor I (A0H) e o valor fornecido pela PIO8 (04H).



. O valor do PC é armazenado no topo da pilha.
. O Z80 busca na tabela endereço de início da rotina de tratamento do PIO8 e transfere para o PC (PC = 9050H).
. O Z80 executa rotina de tratamento até encontrar a instrução RETI.
. O Z80 carrega o PC com o topo da pilha e retorna ao programa interrompido.

HABILITAÇÃO E DESABILITAÇÃO DE INTERRUPÇÃO TA 31200

(1 página)

Descrição Instruções de habilitação e desabilitação de interrupção são instruções usadas para dizer ao Z80 se este pode ou não atender a uma chamada de interrupção (INT = 0).

Símbolo EI e DI

Processo A interrupção $\overline{INT}$ só é atendida quando as interrupções estão habilitadas (ENABLE).
Esta condição é indicada pelo flip-flop de interrupção (IFF1), quando este é setado pela instrução EI (ENABLE INTERRUPT).
Quando IFF1 é ressetado pela instrução DI (DISABLE INTERRUPT), as interrupções ficam bloqueadas.

Exemplo
Toda vez que o Z80 for executar uma rotina que não deve ser interrompida, deve-se colocar na entrada desta a instrução DI e na saída a instrução EI.

```
Rotina 1   DI
             .
             .
             .
           EI
           RETI
```

INSTRUÇÕES DE CONTROLE

TA 31180
(1 página)

Descrição

Instruções de controle são instruções que atuam diretamente no funcionamento do microprocessador, gerando os sinais de controle através dos respectivos pinos (Hardware).

Classificação

As instruções de controle classificam-se em quatro grupos:

. Modos de interrupção
. Habilitação e desabilitação de interrupção
. Instruções que modificam o flag de carry
. Instruções que suspendem a operação da CPU

INSTRUÇÕES QUE MODIFICAM O FLAG DE CARRY

TA 31210
(1 página)

Descrição

São instruções qué alteram o estado do flag de carry (CY).

Símbolo

CCF e SCF

Estrutura

CCF é uma instrução que complementa o flag de carry.
SCF é uma instrução que seta o flag de carry.

Exemplos

. Instrução CCF

Antes	Depois
Carry = 1	Carry = 0
Carry = 0	Carry = 1

. Instrução SCF

Antes	Depois
Carry = 0	Carry = 1
Carry = 0	Carry = 1

INSTRUÇÕES DE MANIPULAÇÃO E TESTE DE BITS TA 31230

(3 páginas)

Descrição **Instruções de manipulação e teste de bits** são instruções que atuam diretamente no bit especificado pelo operando, podendo este bit estar num registrador ou numa posição de memória especificada pelo par de registradores HL ou pelos registradores de 16 bits IX e IY.

Classificação Estas instruções classificam-se em:

. Manipulação de bit

. Teste de bit

Manipulação de bit

Descrição

As instruções de manipulação de bits são aquelas capazes de alterar diretamente o estado do bit especificado pelo operando.

Símbolo

As instruções de manipulação de bit são:

. RES

. SET

Características

As instruções são compostas por:

Instrução	Operando
RES	b, r
	b, (HL)
	b, (IX + d)
	b, (IY + d)

Instrução	Operando
SET	b, r
	b, (HL)
	b, (IX + d)
	b, (IY + d)

Comentário

. b pode ser qualquer um dos oito bits do operando da direita.

. A instrução RES faz com que o bit especificado pelo operando fique em nível lógico 0.

. A instrução SET faz com que o bit especificado pelo operando fique com nível lógico 1.

Teste de bit

Descrição

A instrução de teste de bit apenas verifica se o bit especificado pelo operando está em nível lógico 1 ou 0.

Símbolo

bit

Estrutura

A instrução de teste de bit é composta por:

Instrução	Operando
bit	b, r
	b, (HL)
	b, (IX + d)
	b, (IY + d)

Comentário

A instrução bit apenas verifica se o bit especificado pelo operando está em nível lógico 1 ou 0. Condição esta que pode ser utilizada para tomar decisões em programas.

Exemplo

Instrução bit 1,A

Programa bit 1,A
salte se for zero

INSTRUÇÕES QUE SUSPENDEM A OPERAÇÃO DA CPU

TA 31220
(1 página)

Descrição

Instruções que suspendem a operação da CPU são instruções que mantêm a CPU sem fazer tanto operações internas quanto operações de escrita e leitura, quer em memória quer em I/O.

Símbolo

NOP e HALT

Estrutura

NOP é uma instrução que faz com que o processador central não execute nenhuma operação durante este ciclo de máquina, mas mantém ativo o refresh (restauração) de memória.

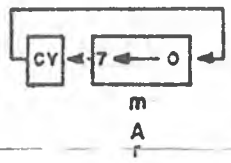
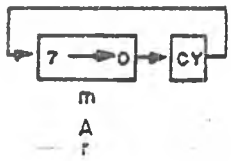
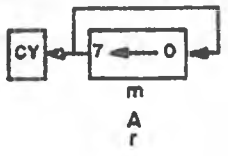
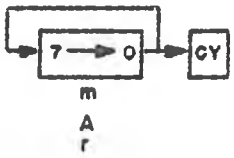
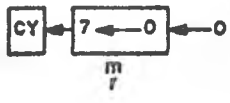
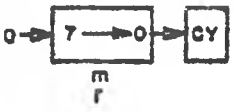
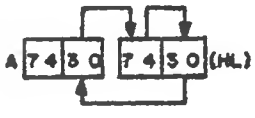
A instrução HALT suspende a operação do processador central até ocorrer uma interrupção ou um REARME do sistema. Enquanto estiver em HALT, o processador estará executando NOP para que seja mantida a restauração de memória.

DESLOCAMENTO E ROTAÇÕES

TA 31260
(2 páginas)

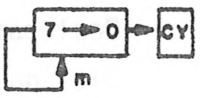
Descrição **Deslocamento e rotações** são instruções que movimentam os bits do acumulador ou de uma posição de memória para a direita ou para a esquerda, passando pelo flag de carry ou não.

Classificação As instruções de rotação classificam-se como mostra a tabela abaixo.

Rotação esquerda		Rotação direita	
RLr RLA RLm		RRr RRA RRm	
RLCr RLCA RLCm		RRCr RRCA RRCm	
SLAr SLAm		SRLr SRLm	
RLD		RRD	

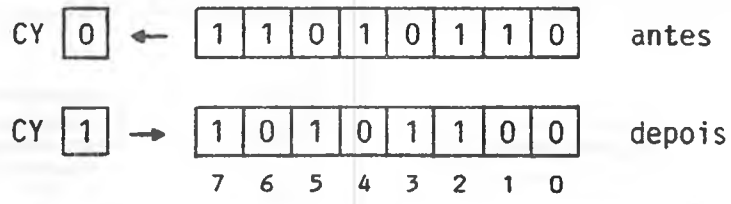
r	m
B	(HL)
C	
D	(IX+d)
E	
H	(IY+d)
L	

SRAr
SRAm



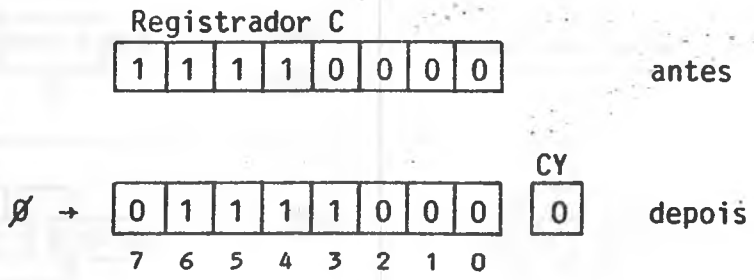
Exemplo

. Instrução RLr → RLB



O conteúdo do registrador B é deslocado à esquerda, o conteúdo do bit 7 é copiado no flag de carry e o conteúdo do flag de carry é copiado no bit 0 do registrador B.

. Instrução SRLr → SRLC



O conteúdo do registrador C é deslocado à direita, o bit 7 é ressetado e o conteúdo do bit 0 é copiado no flag de carry.
Pode-se observar que se essa instrução for executada oito vezes, o conteúdo do registrador C será igual a zero e o flag de carry igual ao conteúdo do bit 7.

INSTRUÇÕES DE TRANSFERÊNCIA DE BLOCOS E PESQUISA TA 31130

(9 páginas)

Descrição

Instruções de transferência de blocos e pesquisa são um grupo de instruções específicas do Z80 que transfere um conjunto de bytes de uma posição de memória para outra especificada pela instrução.

As instruções de pesquisa procuram na memória um byte especificado no acumulador.

Classificação

Essas instruções são divididas em dois grupos:

. Instruções de transferência de bloco

LDI

LDIR

LDD

LDDR

. Instruções de pesquisa

CPI

CPIR

CPD

CPDR

LDI

Descrição

Esta instrução carrega o conteúdo endereçada pelo par de registradores HL na posição de memória endereçada pelo par de registradores DE. Os pares de registrados HL e DE são incrementados de uma unidade nquanto o par de registradores BC é decrementado de uma unidade.

Estrutura

A instrução pode ser entendida como:

LD (DE), (HL)

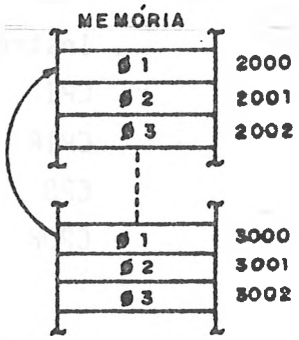
BC ← BC-1

DE ← DE+1

HL ← HL+1

Exemplo

ANTES	DEPOIS
HL = 3000H DE = 2000H BC = 03H	HL = 3001 DE = 2001 BC = 02



Comentário

Esta instrução transfere apenas um byte toda vez que é executada. Mas, para que o byte seja transferido, BC tem de ser diferente de zero.

LDIR

Descrição

Esta instrução carrega o conteúdo da posição de memória endereçada pelo par de registradores HL na posição de memória endereçada pelo par de registradores DE. Os pares de registradores HL e DE são incrementados de uma unidade. O par de registradores BC é decrementado de uma unidade até que BC seja igual a zero.

Estrutura

A instrução pode ser entendida como

LD (DE), (HL)

BC ← BC-1

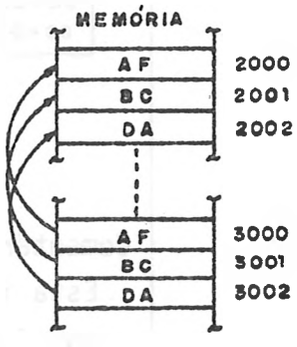
DE ← DE+1

HL ← HL+1

PC ← PC-2 se BC ≠ 0

Exemplo

ANTES	DEPOIS
HL = 3000H	HL = 3002
DE = 2000H	DE = 2002
BC = 03H	BC = Ø
PC = 4006H	PC = 4000H



Comentário

O par BC também é chamado contador, pois é nele que stá especificado o número de bytes a ser transferido.

LDD

Descrição

Esta instrução carrega o conteúdo de memória endereçada pelo par de registradores **HL** na posição de memória endereçada pelo par de registradores DE. Os pares de registradores BC, DE, HL são decrementados de uma posição.

Estrutura

A instrução pode ser entendida como:

LD (DE), (HL)

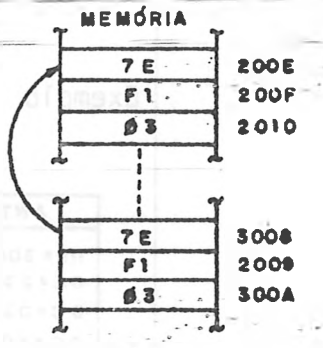
BC ← BC-1

DE ← DE-1

HL ← HL-1

Exemplo

ANTES	DEPOIS
HL = 300A DE = 2010 BC = Ø3	HL = 3009 DE = 200F BC = Ø2



Comentário

Esta instrução transfere apenas um byte toda vês que é executada, mas para que o byte seja transferido BC tem que ser diferente de zero.

LDDR

Descrição

Esta instrução carrega o conteúdo da posição de memória endereçada pelo par de registradores HL na posição de memória endereçada pelo par de registradores DE. Os pares de registradores BC, DE, HL são decrementados de uma posição. O programa COUNTER (PC) será decrementado de duas posições se o conteúdo do par BC for diferente de zero.

Estrutura

A instrução pode ser entendida como:

LD (DE), (HL)

BC ← BC-1

DE ← DE-1

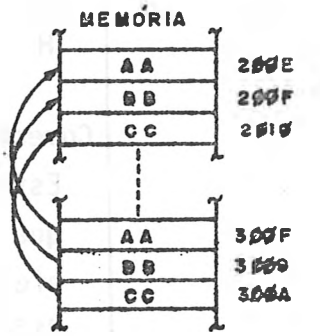
HL ← HL-1

PC ← PC-2 se BC ≠ Ø

Exemplo

LDDR

ANTES	DEPOIS
HL = 300AH DE = 2010H BC = 03H PC = 400AH	HL = 3008H DE = 200EH BC = 00 H PC = 4004H



Comentário

Esta transferência se repete até BC igual a Zero. O par BC também é chamado de contador, pois é nele que está especificado o número de bytes a ser transferido.

CPI

Descrição

Esta instrução subtrai oc onteúdo do acumulador da posição de memória, endereçada pelo par de registradores HL, afetando apenas os flags. Após esta operação, o par de registradores HL é incrementado, o que indicará a posição seguinte de memória a ser comparada. O par de registradores BC é decrementado de uma unidade.

Estrutura

A instrução pode ser entendida como:

A - (HL)

HL ← HL+1

BC ← BC-1

Condição dos flags:

C: não há alteração

Z = 1 se A = (HL)

P/V = 1 se BC - 1 ≠ ∅

S = 1 se o bit 7 = 1

N = 1 em qualquer condição

H = 1 se vai 1 no bit 4

Comentário

Esta instrução compara apenas um byte toda vêz que é executada, mas para que a comparação seja efetuada, o par de registradores BC tem que ser diferente de zero.

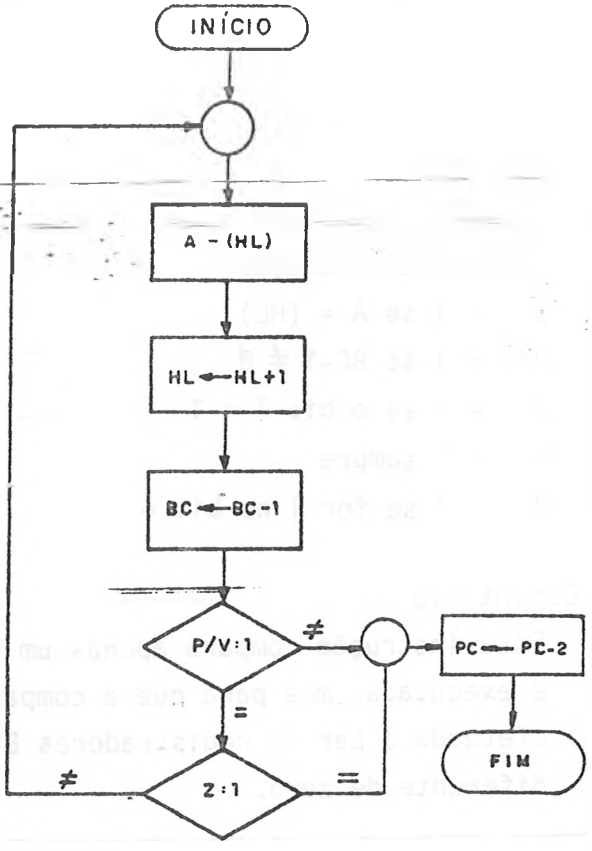
CPIR

Descrição

Esta instrução, uma vez executada, entra em repetição, saindo dela apenas quando o conteúdo do acumulador for igual à posição de memória, endereçada pelo par de registradores HL, ou quando o conteúdo do par de registradores BC for igual a zero.

Estrutura

A instrução pode ser entendida através do seguinte fluxograma:



Condição dos flags:

Z = 1 se A = (HL)

P/V = 1 se BC-1 ≠ Ø

CPD

Descrição

Esta instrução subtrai o conteúdo do acumulador da posição de memória, endereçada pelo par de registradores HL, afetando os flags. Após esta operação, o par de registradores HL é decrementado, o que indicará a posição seguinte a ser comparada. O par de registradores BC é decrementado de uma unidade.

Estrutura

A instrução pode ser entendida como:

A - (HL)
HL HL-1
BC BC-1

Condição dos flags:

C: não há alteração
Z = 1 se A = (HL)
P/V = 1 se BC-1 $\neq \emptyset$
S = 1 se o bit 7 = 1
N = 1 sempre
H = 1 se for 1 no bit 4

Comentário

Esta instrução compara apenas um byte toda vêz que é executada, mas para que a comparação seja efetuada o par de registradores BC tem que ser diferente de zero.

CPDR

Descrição

Esta instrução, uma vez executada, entra em repetição, saindo dela apenas quando o conteúdo do acumulador for igual à posição de memória, endereçada pelo par de registradores **HL**, ou quando o conteúdo do par de registradores **BC** for igual a zero.

Estrutura

A instrução pode ser entendida como:

A - (HL)
HL ← HL-1
BC ← BC-1
PC ← PC-2 se BC $\neq \emptyset$ ou A = (HL)

Condição dos flags:
C: não há alteração
Z = 1 se A = (HL)
P/V = 1 se BC-1 $\neq \emptyset$
S = 1 se o bit 7 = 1
N = 1 sempre
H = 1 se for 1 no bit 4

JUMPs, CALLs E RETs

TA 31240
(3 páginas)

Descrição

JUMPs, CALLs e RETs são instruções usadas para desviar a execução de um programa principal para sub-rotinas e retornar ao programa principal, quando possível, através de RETs.

Classificação

Essas instruções podem ser classificadas em:

. desvios condicionais;
. desvios incondicionais;
. chamadas condicionais;
. chamadas incondicionais;
. retornos condicionais;
. retornos incondicionais.

Características

. Instruções de desvios condicionais (JUMPs) são aquelas que só saem do programa principal após testar a condição do flag especificado pelo operando da instrução.
. Instruções de desvios incondicionais são aquelas que saem do programa principal toda vez que são executadas.
. Instruções de chamadas condicionais são aquelas que saem do programa principal após testar a condição do flag especificado pelo operando da instrução que está sendo utilizada.
Então, essas instruções passam a executar a sub-rotina chamada, até encontrar um retorno.
. Instruções de chamadas incondicionais são aquelas que saem do programa principal, passando a executar a sub-rotina chamada, até encontrar um retorno.
. As instruções de retornos condicionais são retornos de chamadas que voltam ao programa principal, após testar a condição do flag especificado pelo operando.
. As instruções de retornos incondicionais retornam ao programa principal toda vez que são executadas.

Estrutura . . . Tabela das instruções

JUMPs, CALLs e RETs

	Z	NZ	C	NC	PE	PO	M	P
JP	CAXXXX	C2XXXX	DAXXXX	D2XXXX	EAXXXX	E2XXXX	FAXXXX	F2XXXX
CALL	CCXXXX	C4XXXX	DCXXXX	D4XXXX	ECXXXX	E4XXXX	FCXXXX	F4XXXX
RET	C8	C0	D8	D0	E8	E0	F8	F0
JR	28XX	20XX	38XX	30XX				

	Incondicionais	(HL)	(IX)	(IY)
JP	C3XXXX	E9	DDE9	FDE9
CALL	CDXXXX			
RET	C9			
JR	18XX			

RST	00	08	10	18	20	18	30	38
	C7	CF	D7	DF	E7	EF	F7	FF

DJNZ	10XX	DEC B; JR NZ,d
RETI	ED4D	Retorno de interrupção
RETN	ED45	Retorno de interrupção não mascarada

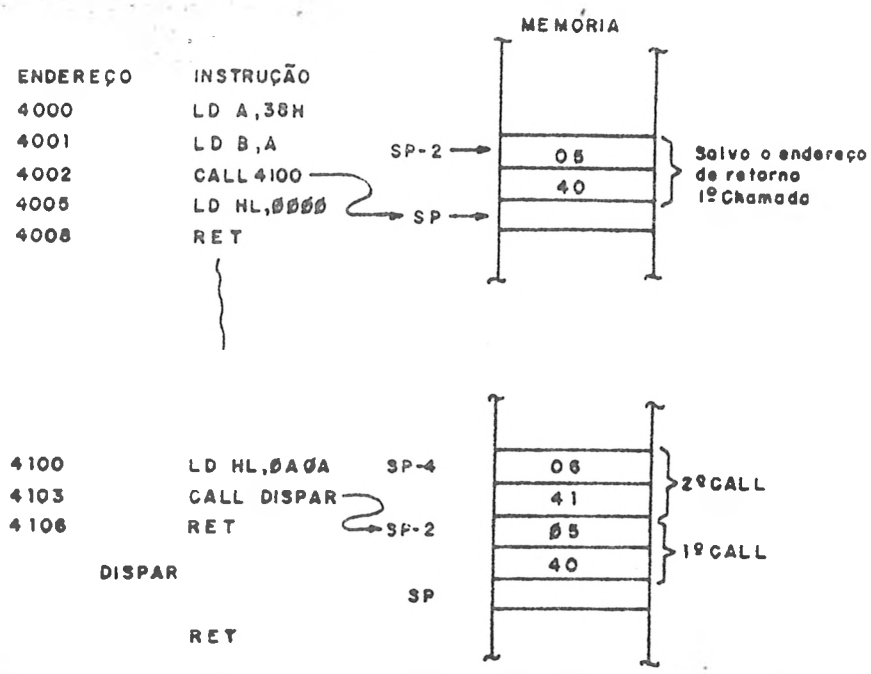
Comentário

Os JUMPs podem ser:

- . JP - JUMP incondicional
- . JR - JUMP relativo incondicional
- . JP FLAG - JUMP condicional
- . CALL - Desvio incondicional
- . CALL FLAG - Desvio condicional
- . RET - Retorno incondicional
- . RET FLAG - Retorno condicional
- . JR FLAG - JUMP relativo condicional
- . RETI - Retorno de interrupção
- . RETN - Retorno de interrupção não mascarada
- . RST - CALL com endereço definido pelo microprocessador

Exemplo

. Instrução CALL



Observação

No exemplo pode-se observar que para cada instrução CALL o microprocessador guarda na pilha o endereço da instrução seguinte, e em seguida salta para a rotina chamada.

O retorno é feito toda vez que se executa uma instrução RET; essa instrução lê na pilha o endereço de retorno e volta para o endereço seguinte da última instrução CALL chamada.

INSTRUÇÕES ARITMÉTICAS ESPECIAIS

TA 31170
(3 páginas)

Descrição O grupo de instruções aritméticas especiais realiza operações de complementação, complementação a dois e ajuste de números na representação BCD.

Classificação As instruções aritméticas especiais são compostas por três instruções:

. CPL (complemento a um)
. NEG (complemento a dois)
. DAA (ajuste decimal)

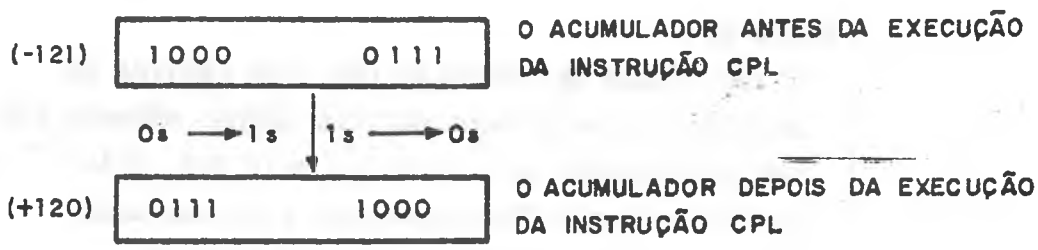
CPL

Descrição

A instrução CPL determina o complemento a um do conteúdo do acumulador pela troca de todos os bits zero por um e vice-versa.

Exemplo

Instrução CPL



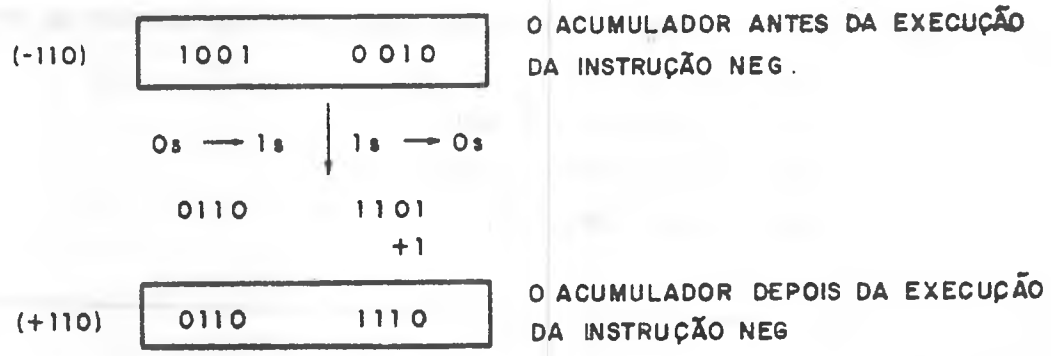
NEG

Descrição

A instrução NEG determina o complemento a dois do conteúdo do acumulador, pela adição de uma unidade ao seu complemento a um.

Exemplo

Instrução NEG



DAA

Descrição

A instrução DAA realiza o ajuste decimal do acumulador após operações com números BCD.

Processo

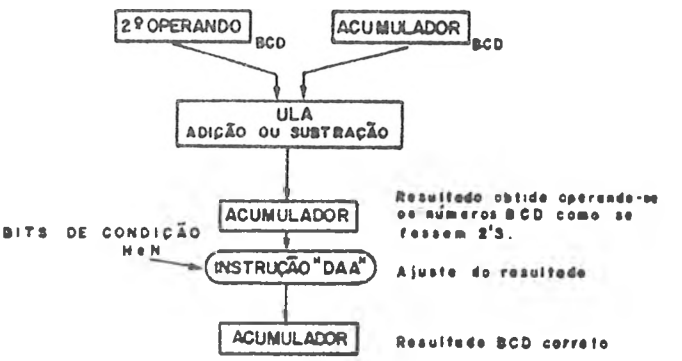
O Z80 dispõe da instrução DAA, que realiza os ajustes dos resultados obtidos quando números BCD são adicionados ou subtraídos pela ULA. Isto ocorre porque a ULA assume que está operando números representados na notação binária.

Comentário

A instrução DAA usada imediatamente após as instruções ADD, ADC, INC, SUB, SBC, DEC ou NEG ajusta o resultado obtido para BCD.

Exemplo

Instrução DAA



Para as operações de adição, ou seja, quando **N** é igual a 0, o resultado é ajustado pela DAA da seguinte maneira:

. Se H for igual a 1, subtrai-se seis do dígito BCD menos significativo.

. Se C for igual a 1, subtrai-se seis do dígito BCD mais significativo.

. Se C for igual a 1 e H for igual a 1, subtrai-se seis de ambos os dígitos BCD.

Para as operações de subtração, isto é, quando **N** é igual a 1, o resultado é ajustado pela instrução DAA, subtraindo-se seis ao dígito BCD nas seguintes condições:

. Se H for igual a 1, soma-se seis ao dígito BCD menos significativo

. Se C for igual a 1, soma-se seis ao dígito BCD mais significativo

. Soma-se seis ao dígito BCD que tiver valor superior a 1001B.

OPERAÇÕES DE SUBTRAÇÃO DE 16 BITS

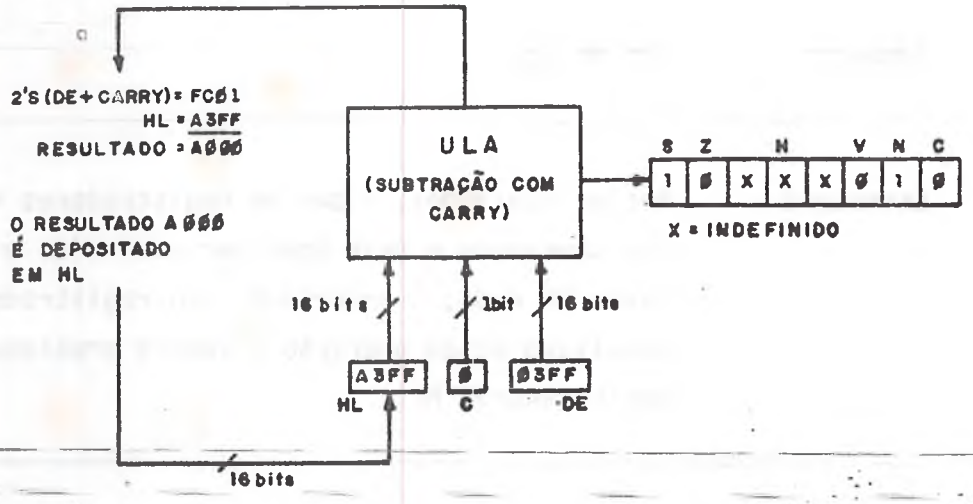
TA 31160
(2 páginas)

Descrição	**Operações de subtração** são instruções que executam a subtração entre dois registradores de 16 bits especificados pela instrução.
Símbolo	SBC HL,qq
Estrutura	Nestas operações, o par de registradores HL é sempre um dos operandos e dele pode ser subtraído o carry-bit, os pares BC e DE, o próprio HL e o registrador SP. O resultado dessa operação é sempre armazenado no par de registradores HL.
Comentário	As instruções de subtração com operandos de 16 bits possibilitam testar os bits S, Z, C e o overflow.
Processo	As operações de subtração com operandos de 16 bits são sempre com carry e são realizadas sempre na aritmética complemento a dois.

Exemplos

. Instrução SBC HL,DE

Se o par DE contiver 03FF H, o par HL for igual A3FF H e o bit C for igual a 0, então a instrução SBC HL, DE realizará a operação de subtração com carry, conforme mostra a figura abaixo.



. Instrução INC IX

Se o registrador IX contiver FFFFH, então a instrução INC IX realizará a operação de incremento, conforme a figura abaixo.