

Linguagem de Montagem

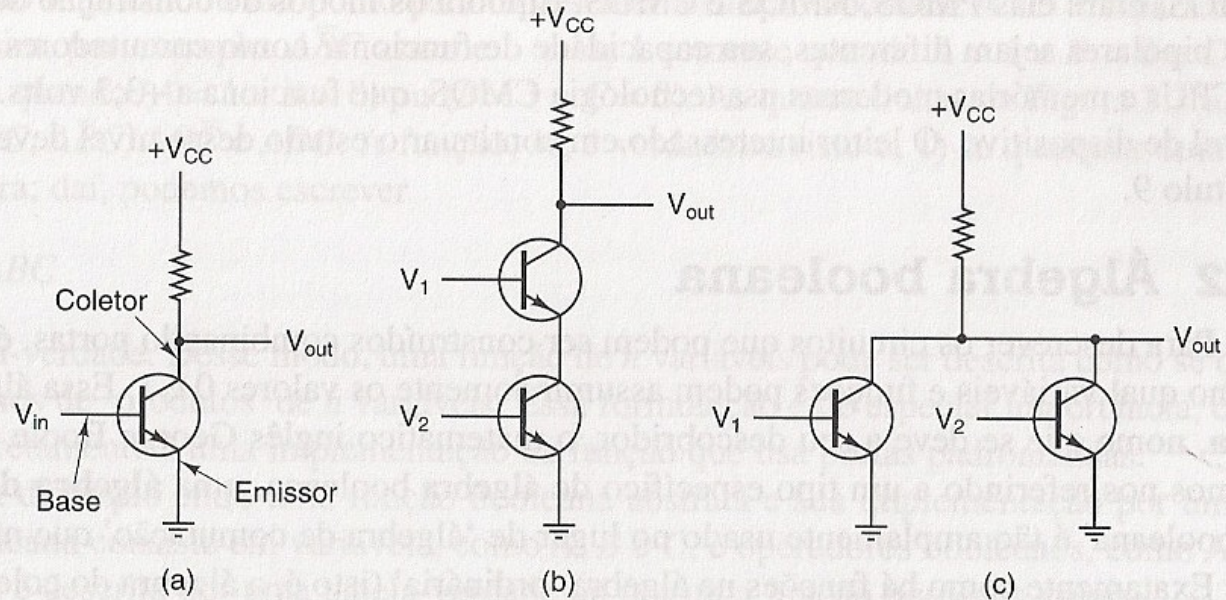
Lembretes

Prof Osvaldo V. Jaques

Como é um transistor

Figura 3.1

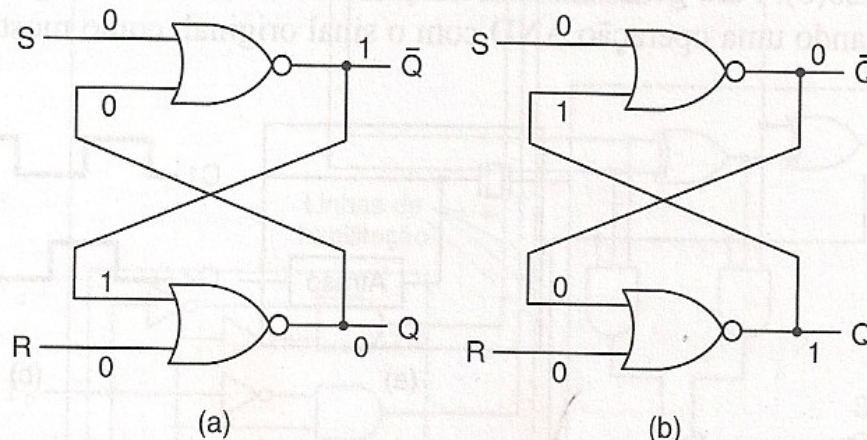
(a) Inversor a transistor.
(b) Porta NAND. (c) Porta NOR.



Um latch

Figura 3.21

a) Latch NOR no estado 0. (b)
Latch NOR no estado 1. (c)
Tabela-verdade para NOR.



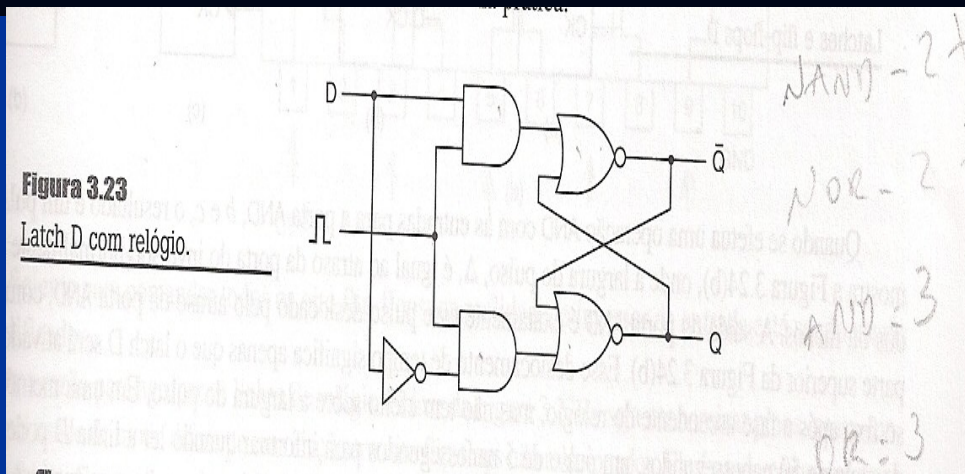
A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

$S=R=0$
manter em

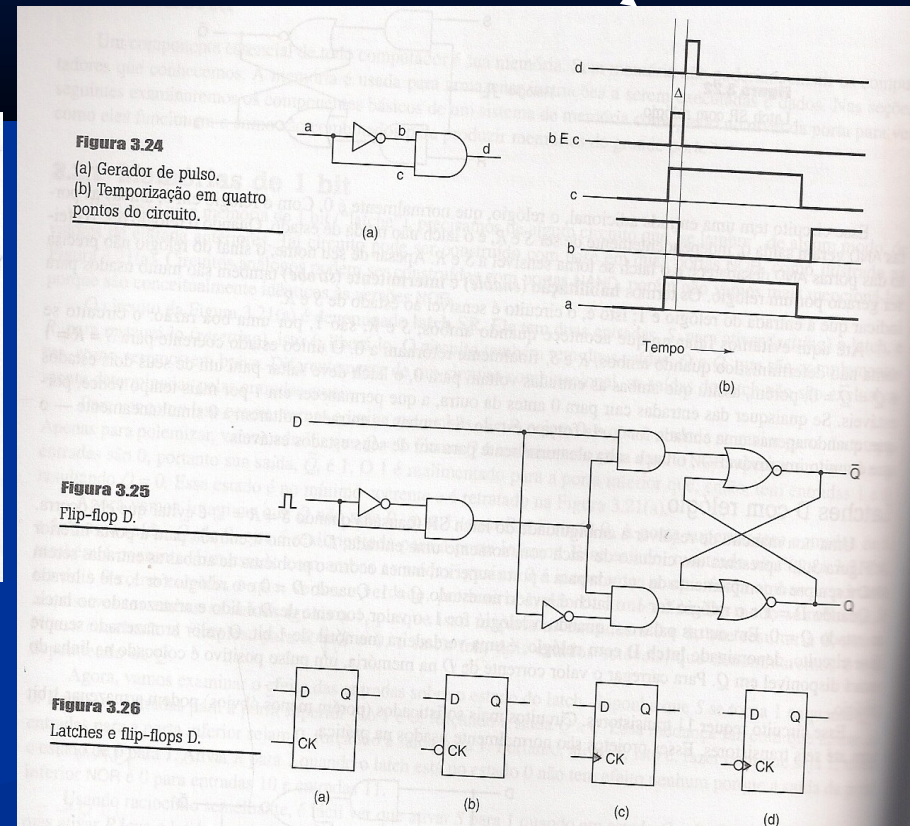
$S=1$ $Q=1$ $\bar{Q}=0$
 $R=1$ $Q=0$ $\bar{Q}=1$

Latches x Flip-flops

Muda na transição do relógio



Quantos transistors?

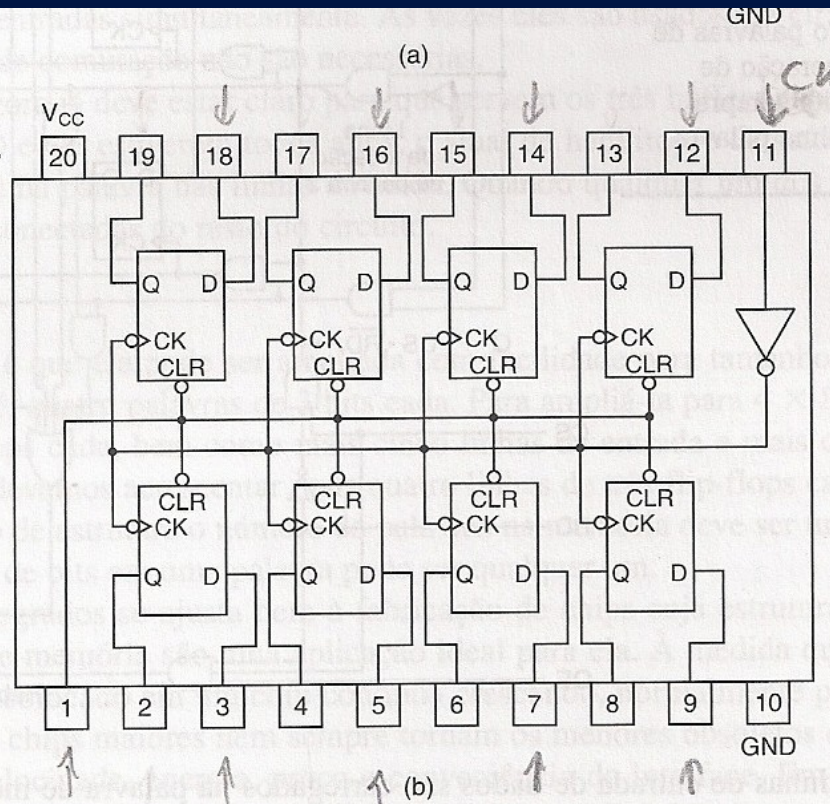


Memória

Figura 3.27

(a) Flip-flop D dual.

(b) Flip-flop octal.



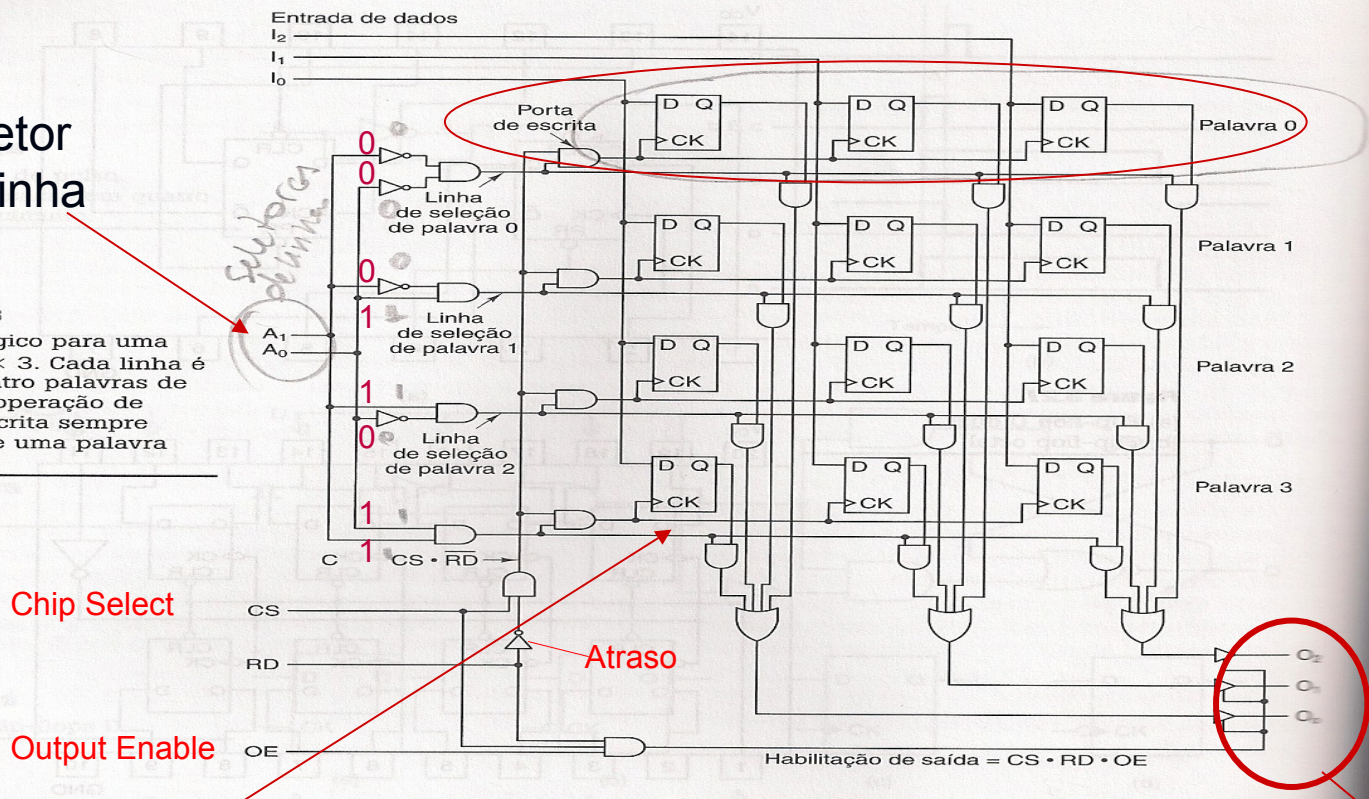
Memória 4x3

Palavra de 3 bits

Seletor de linha

Figura 3.26

Diagrama lógico para uma memória 4×3 . Cada linha é uma das quatro palavras de 3 bits. Uma operação de leitura ou escrita sempre lê ou escreve uma palavra completa.



CS.RD alta \rightarrow escrita

Clk=1 habilita entrada

CS.RD baixa \rightarrow leitura e sem modificação

saida

Registradores Gerais do 8086

Registradores gerais
de dados e endereços

AX

BX

CX

DX

SI

DI

BP

SP

Registradores
de segmento

CS

SS

DS

ES

Ponteiro de instruções
e registrador de status

IP

FLAGS

Um código no velho DOS

Endereço	Código	Assembly
1B8D:0100	01D8	ADD AX,BX
1B8D:0102	C3	RET
1B8D:0103	16	PUSH SS
1B8D:0104	B03A	MOV AL,3A
1B8D:0106	380685D5	CMP [D585],AL
1B8D:010A	750E	JNZ 011A
1B8D:010C	804E0402	OR BYTE PTR [BP+04],02
1B8D:0110	BF86D5	MOV DI,D586
1B8D:0113	C6460000	MOV BYTE PTR [BP+00],00
1B8D:0117	E85F0B	CALL 0C79
1B8D:011A	8B7E34	MOV DI,[BP+34]
1B8D:011D	007C1B	ADD [SI+1B],BH

segmento

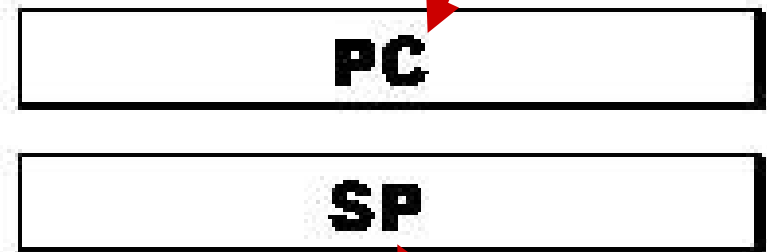
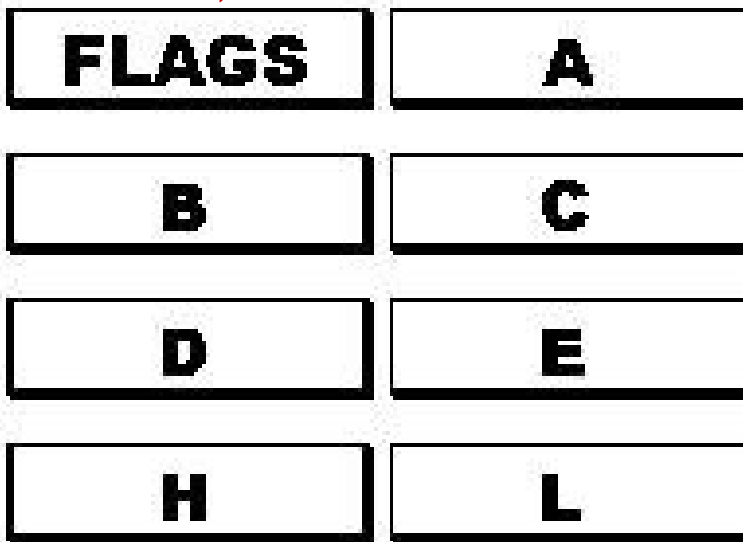
deslocamento

01D8=0000 0001 1101
1000 8

Registradores internos (8080)

Sinalizadores

Sempre o endereço da próxima
Instrução a ser executada



Uso geral
(operandos)

Após operação entre operandos o resultado sai no acumulador

Chamadas de funções

Flags

Símbolo	Nome	Descrição
Z	Zero	Indica se o resultado da operação foi zero
CY	Carry	Indica se uma operação aritmética teve "vai um" ou "pede emprestado"
P	Parity	Indica a paridade do resultado da operação.
S	Signal	Indica o sinal de uma operação, se foi positivo ou negativo
AC	Aux. Carry	Carry auxiliar, em algumas instruções especiais.

Instruções lógicas

- Programar em assembly do 8080 consiste em utilizar suas instruções, manipulando seus registradores para executar as funções que desejamos.

	LXI H, 1000h	; HL vai apontar para a origem
	LXI D, 2000h	; DE vai apontar para o destino
	MVI C, 30	; C será usado como contador:
TRAN:	MOV A,M	; Pega o byte da origem
	STAX	; Guarda no destino
	INX H	; Aponta para o próximo byte
	INX D	; Aponte para o próximo byte
	DCR C	; Decrementa o contador
	JNZ TRAN	; Vai para TRAN se não chegou a ZERO

Registradores do 8086 (de novo)



AX, BC, CX e DX são de 16 bits. São divididos em duas partes de 8 bits: AH e AL, BH e BL, CH e CL, DH e DL. H é high e L é low. Essas partes são chamadas de nibbles.

AX é o acumulador, portanto AL, corresponderia ao A no 8080.

O registrador BX do 8086 corresponde ao par HL do 8080 (assim como BH corresponde a H e BL corresponde a L). Da mesma forma, CX corresponde ao par BC e DX corresponde ao par DE. O contador de programa (PC) do 8080 é chamado no 8086 de IP (Instruction Pointer). O Stack Pointer (SP) é similar, e os flags (Cy, Z, AC, P e S) ficam em um registrador de flags, com 16 bits. Para os processadores de 32 bits em diante surgem os EAX, EBX, e assim por diante.

Algumas instruções do 8086

- · Multiplicação de números inteiros de 16 bits
- · Divisão de números inteiros de 32 bits
- · Rotações com qualquer número de bits
- · Movimentação e comparação de strings
- · Instruções para manipulação direta de bits
- · Instruções de repetição

Registradores BX, BP, SI e DI

- Esses registradores permitem várias operações comuns em outros registradores, e além delas, podem ser usados como índices. Por exemplo, podemos usá-los para apontar posições de memória usando expressões como:
 - [BX+valor]
 - [BX+SI+valor]
 - [BP+valor]
 - [BX+DI+valor]
 - [SI+valor]
 - [BP+SI+valor]
 - [DI+valor]
 - [BP+DI+valor]

Registradores BX, BP, SI e DI

Exemplos:

- `MOV BX,1000h` ; Aponta para o endereço 1000h
- `MOV AL,[BX+15h]` ; Lê para AL o valor que está em 1015h
- `MOV BX,2000h`
- `MOV SI,100h`
- `MOV AL,[BX+SI+20h]` ; Lê para AL o valor que está em 2120h

O uso de índices torna a programação extremamente mais simples quando temos que lidar com estruturas de dados na memória, como strings e matrizes.