

Microprocessadores

Família “x86” - Programação do i8086

António M. Gonçalves Pinheiro

Departamento de Física
Universidade da Beira Interior
Covilhã - Portugal

pinheiro@ubi.pt

“Microprocessadores i8086”

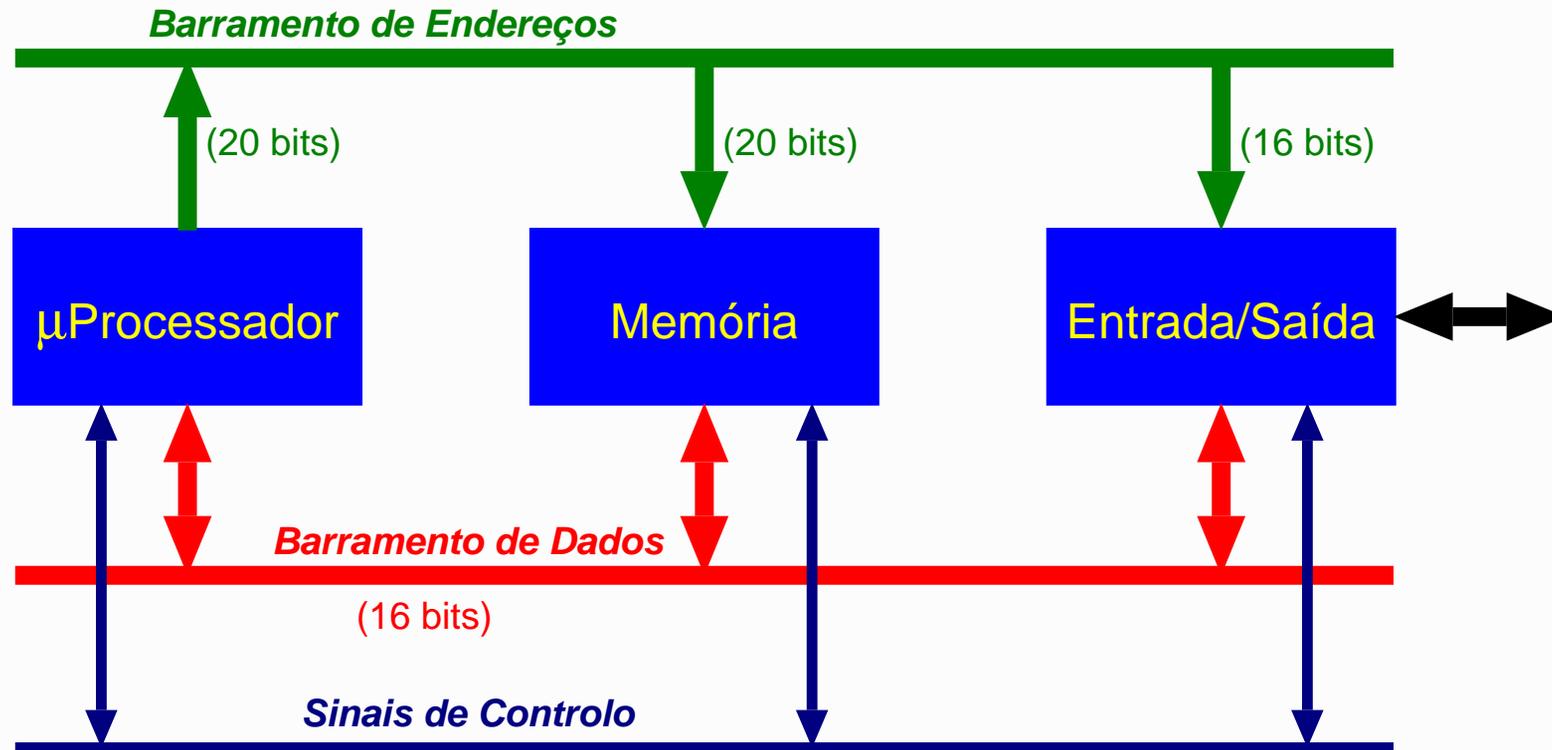
Estudo Genérico de Microprocessador

1. Modelo de Programação;
2. Espaço de Endereçamento de Memória e Organização de Dados
3. Registos
 - (a) Registos de Uso geral e especializado
 - (b) Contador de Programa
 - (c) Ponteiro da “Stack”
 - (d) Registo de Estado (“Flags”)
4. Endereçamento de I/O
5. Modos de Endereçamento
6. Instruções
7. Sinais de Controlo
8. Interrupções



“Microprocessadores i8086”

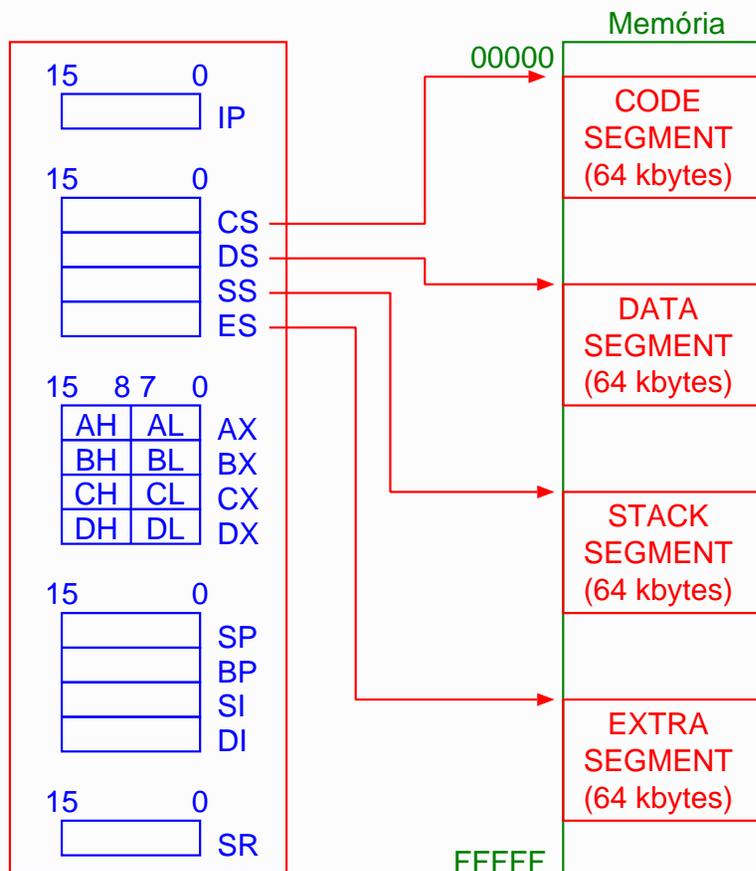
Modelo de Von Neumann aplicado ao INTEL 8086



Desenvolvido com base no Intel 8080
Microprocessador de 16 bits

“Microprocessadores i8086”

Modelo de Programação



IP - Instruction Pointer

Segment Registers

- CS - Code Segm.
- DS - Data Segm.
- SS - Stack Segm.
- ES - Extra Segm.

Data Registers

- AX - Acumulador
- BX - Base Reg.
- CX - Contador
- DX - Data Reg.

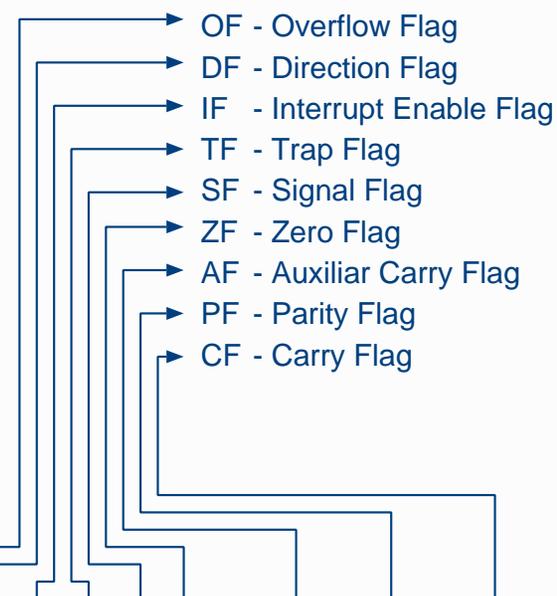
Pointer Registers

- BP - Base Pointer
- SP - Stack Pointer

Index Registers

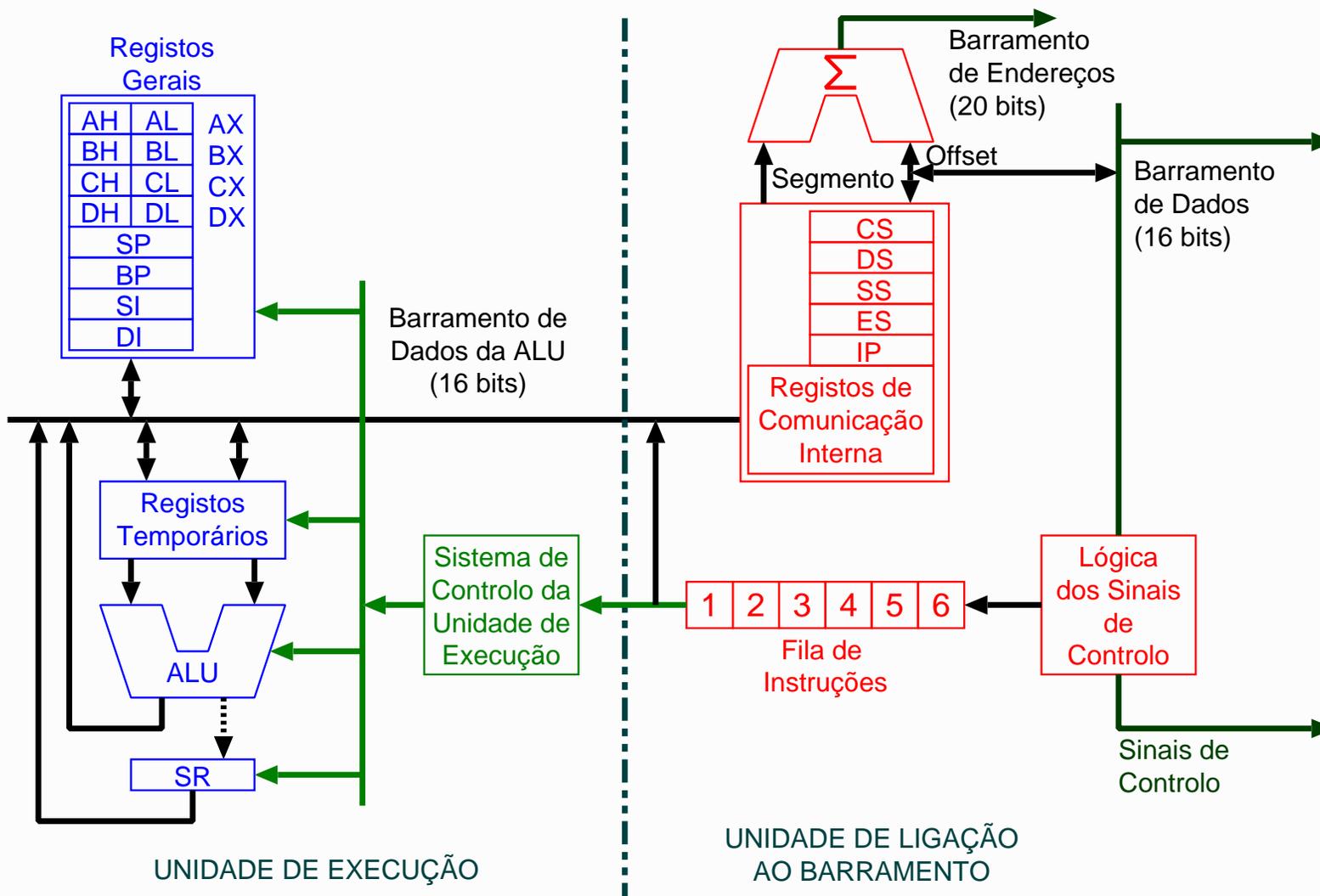
- SI - Source Index
- DI - Destination Index

Status Register



“Microprocessadores i8086”

Diagrama Funcional



“Microprocessadores i8086”

Espaço de Endereçamento de Memória e Organização de Dados

Registo de Segmento - Ponteiro para “Segmento” (Área) de Memória

Quatro segmentos (CODE, STACK, DATA, EXTRA)

Cálculo do Endereço: Endereço=

$$\begin{array}{r} \text{CS} \times 10\text{H} + \text{IP} \\ \text{SS} \times 10\text{H} + \text{SP} \\ \text{DS} \times 10\text{H} + \text{BX} \\ \text{ES} \qquad \qquad \text{BP} \\ \qquad \qquad \qquad \text{SI} \\ \qquad \qquad \qquad \text{DI} \end{array}$$

Nota: Desta forma constroi-se um endereço de 20 bits a partir de registos de 16 bits.

Exemplo:

Segmento= 8080H

Endereço= 80800H + 047BH = 80C7BH Offset= 047BH



“Microprocessadores i8086”

Espaço de Memória e I/O Reservado

1. Localizações de Memória entre 00000H e 003FFH (“Memory Low”) usadas para a tabela do **Vector Interrupt**.
2. Localizações de Memória entre FFFF0H e FFFFFH (“Memory High”) usadas para código de RESET do sistema já que o Processador é inicializado em FFFF0H.
3. Localizações de I/O entre 00F8H e 00FFH reservadas para comunicação de periféricos INTEL como é o caso do copressador matemático 8087.



Programação do i8086

Instruções de Transferência de dados

mov D,S	S→D	<ol style="list-style-type: none"> 1. mov reg,{reg/mem/imediato} 2. mov mem,{reg/imediato} 3. mov {reg16/mem16},{CS/DS/SS/ES} 4. mov {CS/DS/SS/ES},{reg16/mem16}
----------------	-----	--

Modos de Endereçamento

Endereçamento Imediato

```
mov AL, DATABYTE
mov CX, DATAWORD
```

Endereçamento Directo

```
mov CX, [ADDRESS]
```

Endereçamento Indirecto

```
mov DX, [BP]
```

Endereçamento de Base

```
mov [BX]+ADDRESS,CX
```

Endereçamento Indexado

```
mov [DI]+ARRAY,AX
```

Endereçamento de Base Indexado

```
mov AX,[BX][SI]+MATRIX
```



Programação do i8086

Instruções de Transferência de dados

push S	SP-2→SP S→[SP-1:SP]	push {reg16/mem16/CS/DS/SS/ES}
pop D	[SP+1:SP]→D SP+2→SP	pop {reg16/mem16/CS/DS/SS/ES}
xchg F,G	F⇌G	1. xchg {reg/mem},reg 2. xchg reg,{reg/mem}
xlat	[AL+BX+DS×10H]→AL	
lea reg16,mem16	mem16→reg16	
lds reg16,mem32	[mem32]→reg16; [mem32+2]→DS	
les reg16,mem32	[mem32]→reg16; [mem32+2]→ES	



Programação do i8086

Instruções de Transferência de dados

pushf

SP-2→SP

SR→[SP-1:SP]

popf

[SP+1:SP]→SR

SP+2→SP

lahf

SR_{low}→AH

sahf

AH→SR_{low}

Instruções de I/O (Entrada/Saída)

in D,S

S→D

S=Porto[{End8/DX}],D={AL/AX}

out D,S

S→D

S={AL/AX},D=Porto[{End8/DX}]



Programação do i8086

Instruções Aritméticas

				D	S
<i>Adição</i>	add D,S	$D+S \rightarrow D$	OF SF ZF AF PF CF	Registo	Registo
	adc D,S	$D+S+CF \rightarrow D$	OF SF ZF AF PF CF	Registo	Memória
	inc D	$D+1 \rightarrow D$	OF SF ZF AF PF	Memória	Registo
				Registo	Imediato
				Memória	Imediato
		aaa	→ Ajustamento ASCII para adição	AF CF (OF SF ZF PF)?	
	daa	→ Ajustamento decimal para adição	SF ZF AF PF CF (OF)?		
<hr/>					
<i>Subtracção</i>	sub D,S	$D-S \rightarrow D$	OF SF ZF AF PF CF	D	S
	sbc D,S	$D-S-CF \rightarrow D$	OF SF ZF AF PF CF	Registo	Registo
	dec D	$D-1 \rightarrow D$	OF SF ZF AF PF	Registo	Memória
	neg D	$0-D \rightarrow D$	OF SF ZF AF PF CF	Memória	Registo
	cmp D,S		OF SF ZF AF PF CF	Registo	Imediato
				Memória	Imediato
	aas	→ Ajustamento ASCII para subtracção	AF CF (OF SF ZF PF)?		
	das	→ Ajustamento decimal para subtracção	SF ZF AF PF CF (OF)?		



Programação do i8086

Instruções Aritméticas

Multiplicação e Divisão

mul S	<i>Sem sinal</i>	$AL \times S_8 \rightarrow AX$ $AX \times S_{16} \rightarrow (DX, AX)$	OF CF (SF ZF AF PF)?	S Reg ₈ Reg ₁₆ Mem ₈ Mem ₁₆
div S	<i>Sem sinal</i>	1 Quociente(AL/S_8) \rightarrow AL Resto(AL/S_8) \rightarrow AH	(OF CF SF ZF AF PF)?	
		2 Quociente($(DX, AX)/S_{16}$) \rightarrow AX Resto($(DX, AX)/S_{16}$) \rightarrow DX		
imul S	<i>Com sinal</i>		OF CF (SF ZF AF PF)?	
idiv S	<i>Com sinal</i>		(OF CF SF ZF AF PF)?	
aam		Quociente($AL/10$) \rightarrow AH Resto($AL/10$) \rightarrow AL	Ajustamento decimal de AL para multiplicação (unpacked BCD)	SF ZF PF (OF AF CF)?
aad		$AH \times 10 + AL \rightarrow AL$ (0 \rightarrow AH)	Ajustamento decimal de AX para divisão (unpacked BCD)	SF ZF PF (OF AF CF)?
cbw	<i>byte \rightarrow word16</i>		MSB de AL para todos os bits de AH	
cwd	<i>word16 \rightarrow word32</i>		MSB de AX para todos os bits de DX	



Programação do i8086

Instruções Lógicas

			D	S
and D,S	$D \cdot S \rightarrow D$	OF SF ZF PF CF (AF)?	Registo	Registo
or D,S	$D + S \rightarrow D$	OF SF ZF PF CF (AF)?	Registo	Memória
xor D,S	$D \oplus S \rightarrow D$	OF SF ZF PF CF (AF)?	Memória	Registo
not D	$\bar{D} \rightarrow D$		Registo	Imediato
			Memória	Imediato

Instruções de Controlo de Flags

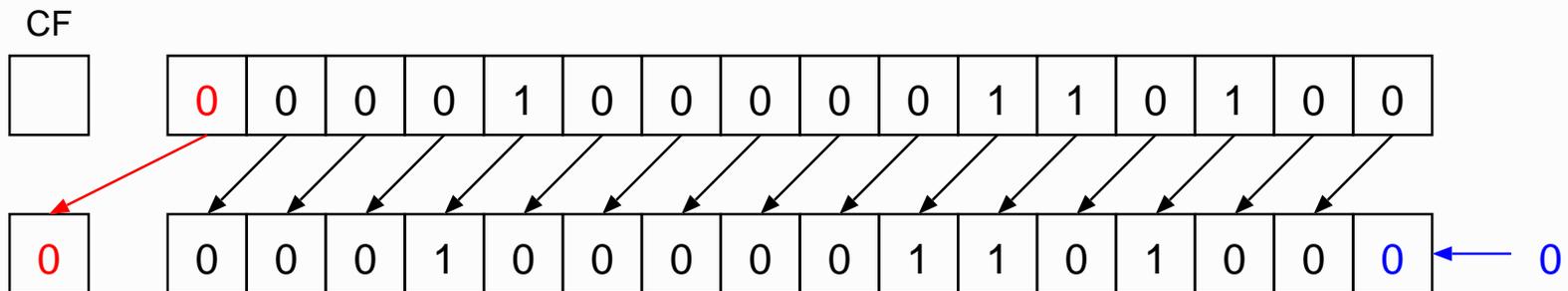
lahf	SRlow \rightarrow AH	
sahf	AH \rightarrow SRlow	
clc	0 \rightarrow CF	
stc	1 \rightarrow CF	
cli	0 \rightarrow IF	Clear Interrupt Flag
sti	1 \rightarrow IF	Set Interrupt Flag
cld	0 \rightarrow DF	Clear Direction Flag
std	1 \rightarrow DF	Set Direction Flag



Programação do i8086

Instruções de Deslocamento

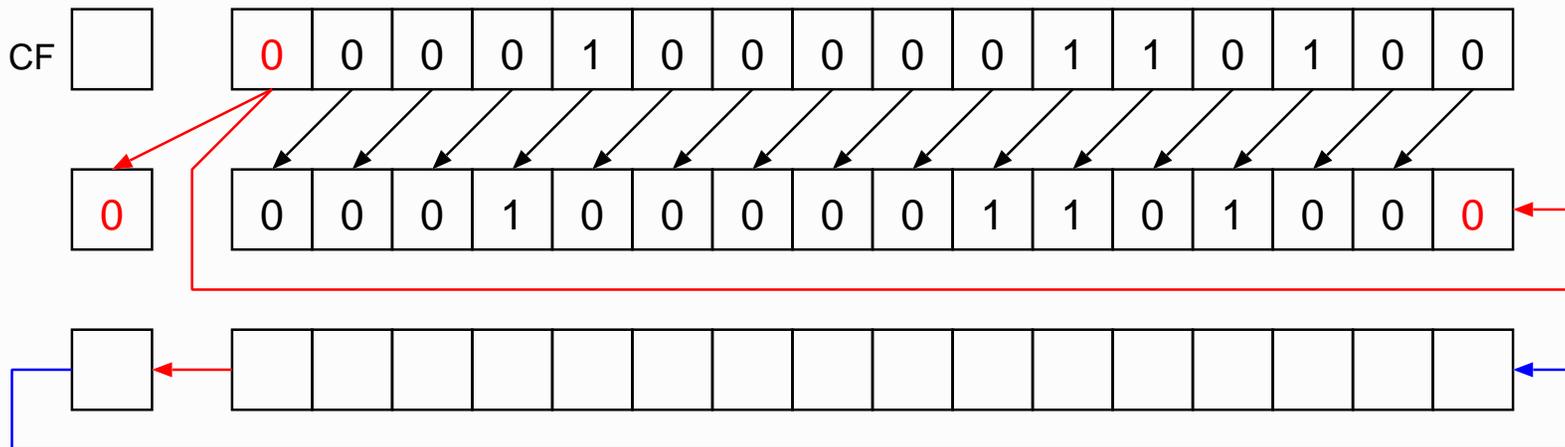
sal/shl D,Count	faz Count vezes $0 \rightarrow D_{LSB}$ e $D_i \rightarrow D_{i+1}$	SF ZF PF CF OF (AF)? (OF)? se count $\neq 1$	D	Count
shr D,Count	faz Count vezes $0 \rightarrow D_{MSB}$ e $D_{i+1} \rightarrow D_i$	SF ZF PF CF OF (AF)? (OF)? se count $\neq 1$	Registo	1
sar D,Count	faz Count vezes $D_{MSB} \rightarrow D_{MSB}$ e $D_{i+1} \rightarrow D_i$	SF ZF PF CF OF (AF)?	Memória	1
			Memória	CL



Programação do i8086

Instruções de Rotação

rol D,Count	faz Count vezes $D_{MSB} \rightarrow CF$, $D_{MSB} \rightarrow D_{LSB}$ e $D_i \rightarrow D_{i+1}$	CF (OF)? se count $\neq 1$	<table border="1"> <thead> <tr> <th>D</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>Registo</td> <td>1</td> </tr> <tr> <td>Registo</td> <td>CL</td> </tr> <tr> <td>Memória</td> <td>1</td> </tr> <tr> <td>Memória</td> <td>CL</td> </tr> </tbody> </table>	D	Count	Registo	1	Registo	CL	Memória	1	Memória	CL
D	Count												
Registo	1												
Registo	CL												
Memória	1												
Memória	CL												
rcl D,Count	faz Count vezes $CF \rightarrow D_{LSB}$, $D_{MSB} \rightarrow CF$ e $D_i \rightarrow D_{i+1}$	CF (OF)? se count $\neq 1$											
ror D,Count	faz Count vezes $D_{LSB} \rightarrow CF$, $D_{LSB} \rightarrow D_{MSB}$ e $D_{i+1} \rightarrow D_i$	CF (OF)? se count $\neq 1$											
rcr D,Count	faz Count vezes $D_{LSB} \rightarrow CF$, $CF \rightarrow D_{MSB}$ e $D_{i+1} \rightarrow D_i$	CF (OF)? se count $\neq 1$											



Programação do i8086

Instruções de Salto

jmp Op Salto Incondicional
jcc Op Salto Condicional

Ex:

jmp LABEL (short, near e far atribuída automaticamente)

jmp BX (BX tem o novo IP)

jmp WORD PTR [BX] (BX tem endereço do novo IP)

jmp DWORD PTR [BX] (BX tem end. de novo IP e BX+2 de novo CS)

Op

Short label

Near label

Far label

Memptr16

Regptr16

Memptr32

Salto condicionais

ja	above	jnb	not below or equal	CF=0 & ZF=0	jcxz	Reg. CX=0	(CF + ZF)=0
jae	above or equal	jnb	not below	CF=0	jnc	not carry	CF=0
jb	below	jnae	not above or equal	CF=1	jc	carry	CF=1
jbe	bellow or equal	jna	not above	CF=1 + ZF=1			
je	equal	jz	zero	ZF=1			
jne	not equal	jnz	not zero	ZF=0			
jg	greater	jnle	not less or equal	ZF=0 & SF=OF			
jge	greater or equal	jnl	not less	SF=OF			
jl	less	jnge	not greater nor equal	(SF⊕OF)=1	jno	not overflow	OF=0
jle	less or equal	jng	not greater	((SF⊕OF)+ZF)=1	jo	overflow	OF=1
jnp	not parity	jpo	parity odd	PF=0	jns	not sign	SF=0
jp	parity	jpe	parity even	PF=1	js	sign	SF=1



Programação do i8086

Instruções de manipulação de Subrotinas

call Op Chama Subrotina

Ex:

call LABEL (near e far atribuída automaticamente)

call BX (BX contém o novo IP)

call WORD PTR [BX] (BX tem endereço do novo IP)

call DWORD PTR [BX] (BX tem end. de novo IP e BX+2 de novo CS)

Op

Near label

Far label

Memptr16

Regptr16

Memptr32

ret Return

ret Op Return (soma-se op ao SP)

Op

DISP16



Programação do i8086

Instruções de Ciclo (LOOP)

loop SHORT-LABEL

$CX \leftarrow CX-1$

salta para SHORT-LABEL caso $CX \neq 0$
caso contrário, não salta.

loope/loopz SHORT-LABEL

$CX \leftarrow CX-1$

salta para SHORT-LABEL caso $CX \neq 0$ e $ZF=1$
caso contrário, não salta.

loopne/loopnz SHORT-LABEL

$CX \leftarrow CX-1$

salta para SHORT-LABEL caso $CX \neq 0$ e $ZF=0$
caso contrário, não salta.



Programação do i8086

Instruções de Manipulação de Cadeias (STRINGS)

movsb	move string byte	$[ES \times 10H + DI] \leftarrow [DS \times 10H + SI]$ $SI \leftarrow SI \pm 1$, dependendo se $DF=0$ ou 1 $DI \leftarrow DI \pm 1$, dependendo se $DF=0$ ou 1	
movsw	move string word	$[ES \times 10H + DI] \leftarrow [DS \times 10H + SI]$ $SI \leftarrow SI \pm 2$, dependendo se $DF=0$ ou 1 $DI \leftarrow DI \pm 2$, dependendo se $DF=0$ ou 1	
cmpsb cmpsw	compare string	$[ES \times 10H + DI] - [DS \times 10H + SI]$ $SI \leftarrow SI \pm 1$ ou 2 , dependendo se $DF=0$ ou 1 $DI \leftarrow DI \pm 1$ ou 2 , dependendo se $DF=0$ ou 1	CF,PF,AF,ZF,SF,OF
scasb scasw	scan string	AL ou AX $- [ES \times 10H + DI]$ $DI \leftarrow DI \pm 1$ ou 2 , dependendo se $DF=0$ ou 1	CF,PF,AF,ZF,SF,OF
lodsb lodsw	load string	AL ou AX $\leftarrow [DS \times 10H + SI]$ $SI \leftarrow SI \pm 1$ ou 2 , dependendo se $DF=0$ ou 1	
stosb stosw	store string	$[ES \times 10H + DI] \leftarrow AL$ ou AX $DI \leftarrow DI \pm 1$ ou 2 , dependendo se $DF=0$ ou 1	



Programação do i8086

Prefixos de Instruções para Repetição

rep	repmovs repstos	repete: enquanto $CX \neq 0$ $CX \leftarrow CX - 1$
repe/repz	repecmps repesaca	repete: enquanto $CX \neq 0$ e $ZF = 1$ $CX \leftarrow CX - 1$
repne/repnz	repnecmps repnesaca	repete: enquanto $CX \neq 0$ e $ZF = 0$ $CX \leftarrow CX - 1$



Programação do i8086

Instruções de Gestão de Interrupções

int N	Chamada de interrupção tipo N	Flags \rightarrow [SP-2] 0 \rightarrow TF; 0 \rightarrow IF CS \rightarrow [SP-4] $2 + 4 \times N \rightarrow$ CS IP \rightarrow [SP-6] $4 \times N \rightarrow$ IP SP \rightarrow SP-6
iret	Retorno de subrotina de interrupção	[SP] \rightarrow IP [SP+2] \rightarrow CS [SP+4] \rightarrow Flags SP+6 \rightarrow SP
into	Chamada de interrupção caso haja Overflow	Igual a int 4
hlt	Halt	Espera Interrupção ou Reset
wait	Estado de Espera	Espera que a entrada $\overline{\text{TEST}}$ fique activa

