

Intel 8086 (e 8088)

Prof. Sérgio L. Cechin

Histórico

- Início da família de processadores
 - 8086 (1978)
 - 8088 (1979)
- 8086
 - Processador de 16 bits
 - Com barramento externo de 16 bits
- 8088
 - Processador de 16 bits
 - Com barramento externo de 8 bits

Registradores

- A, B, C e D
 - AH, AL, BH, BL, CH, CL, DH e DL
 - AX, BX, CX e DX
- 8086
 - AX, BX, CX e DX
 - IP: Instruction Pointer
 - F: Flags

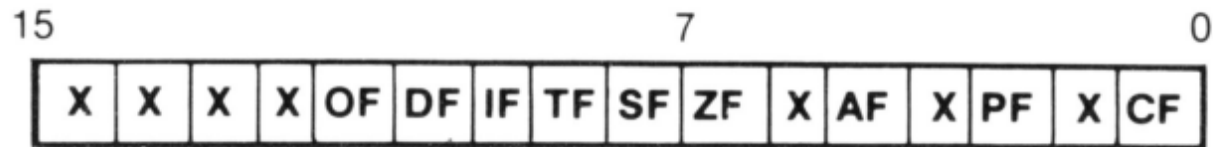
AH	AX	AL
BH	BX	BL
CH	CX	CL
DH	DX	DL

Flags

- **Códigos de condição**
 - CF: carry flag
 - ZF: zero flag
 - SF: sign flag
 - OF: overflow flag
 - etc...
- **Estado do hardware da UCP**
 - DF: direction flag (usado por algumas instruções)
 - IF: interrupt flag
 - IOPL: I/O Privilege level: nível de privilégio de execução das instruções (dois bits)
 - etc...

Flags (completo)

Instructions which reference the flag register file as a 16-bit object use the symbol FLAGS to represent the file:



X = Don't Care

AF: AUXILIARY CARRY — BCD
CF: CARRY FLAG
PF: PARITY FLAG
SF: SIGN FLAG
ZF: ZERO FLAG

8080 FLAGS

DF: DIRECTION FLAG (STRINGS)
IF: INTERRUPT ENABLE FLAG
OF: OVERFLOW FLAG ($CF \oplus SF$)
TF: TRAP — SINGLE STEP FLAG

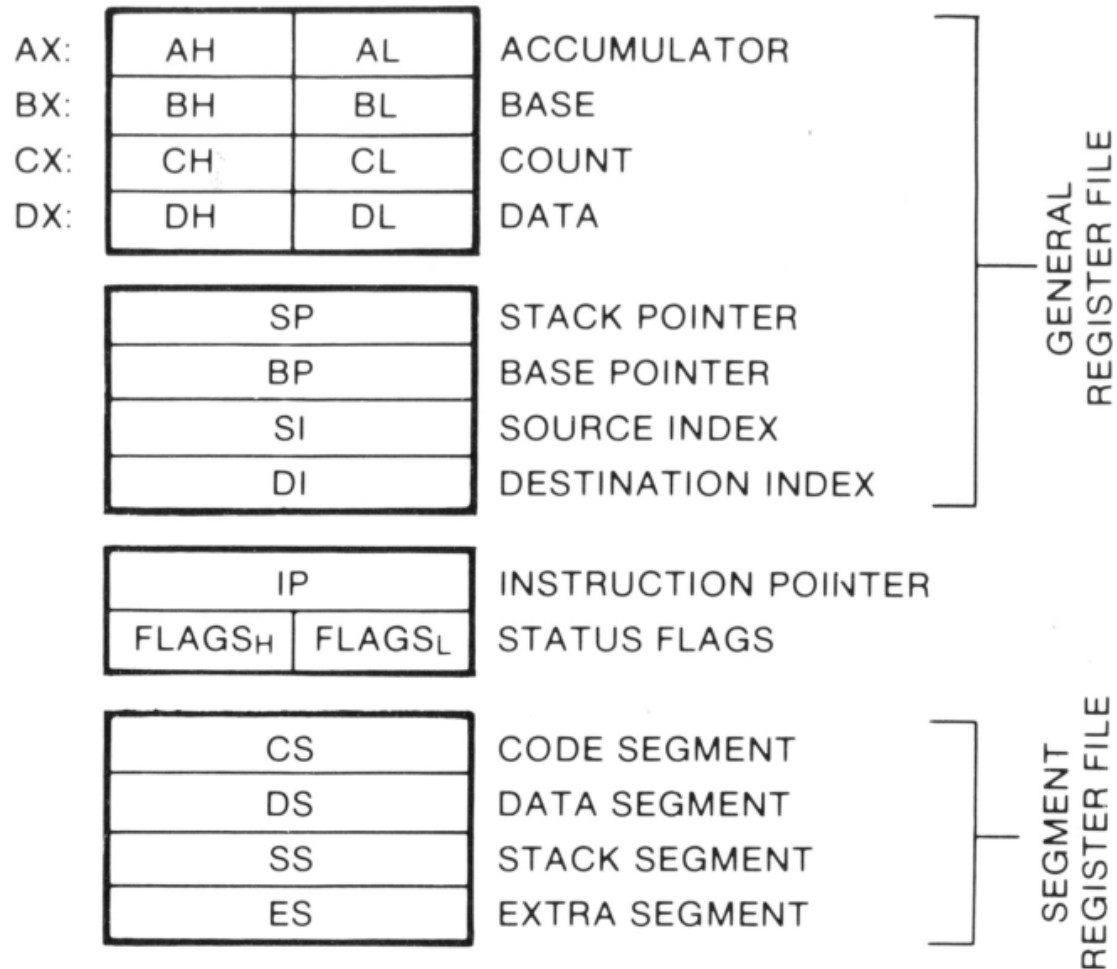
8086 FLAGS

Outros Registradores

- 8086
 - SP: stack pointer
 - BP: base pointer
 - SI: source index
 - DI: destination index

Modelo de Registradores

8086 REGISTER MODEL



Memória

- Memória física
 - Organizada em bytes
 - Até 1 Mbyte (2^{20})
- Memória lógica
 - Estruturada em segmentos

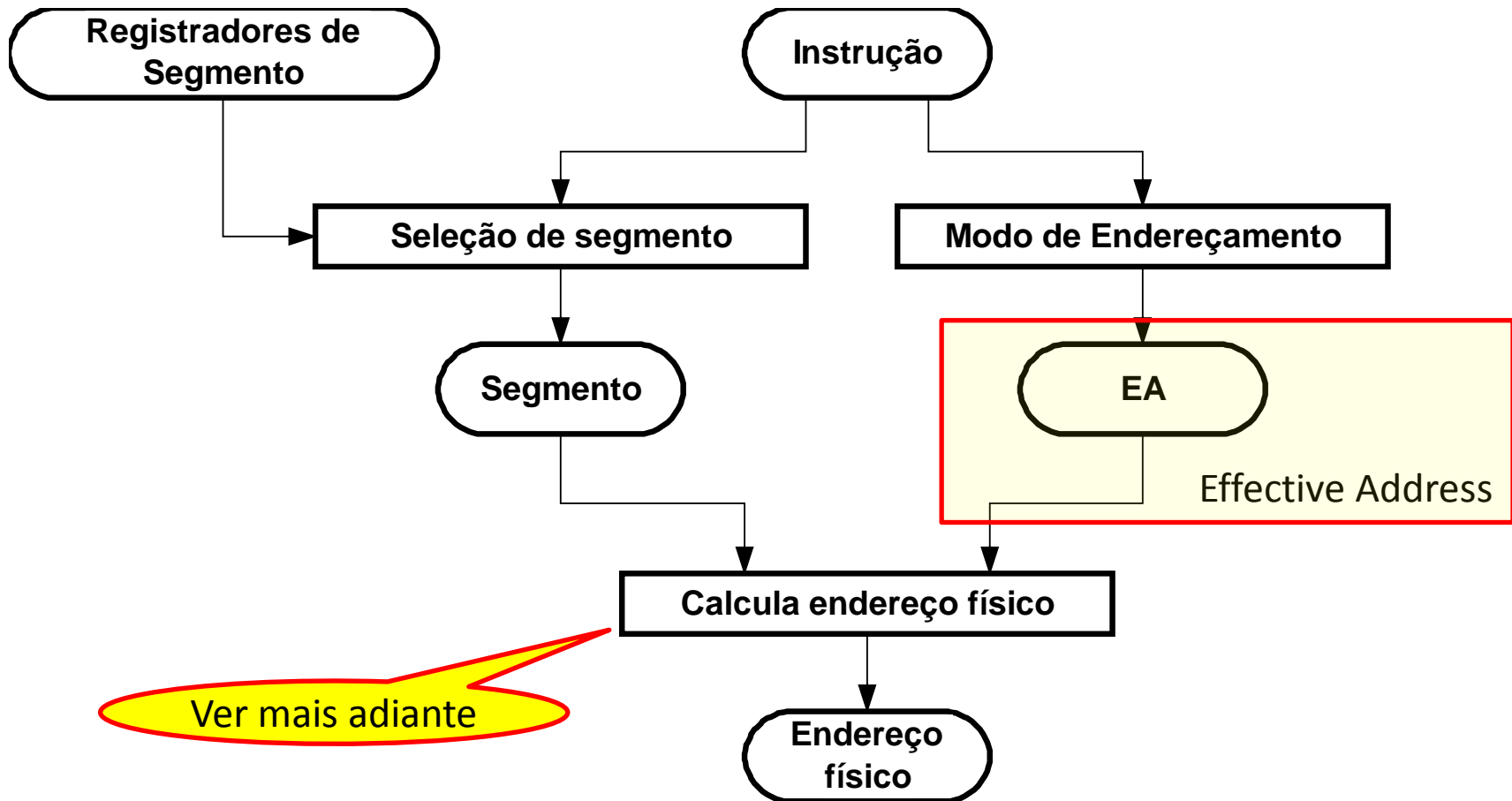
Acesso à Memória

- É feito através de:
 - Um SEGMENTO
 - Região da memória especificada por um início e tamanho
 - Um OFFSET dentro do segmento
 - Posição relativa dentro do segmento
- Um endereço é representado por
 - SEGMENTO:OFFSET
- Seleção do segmento:
 - Utiliza um Registrador de Segmento

Registradores de Segmento

- Registradores do 8086
 - CS: Code Segment
 - SS: Stack Segment
 - DS: Data Segment
 - ES: Extra Segment
- Uso dos registradores
 - Fetch: CS
 - Ex: CS:IP
 - Acessos à pilha
 - Ex: SS:SP
 - Acessos à dados: DS e ES
 - Depende da instrução

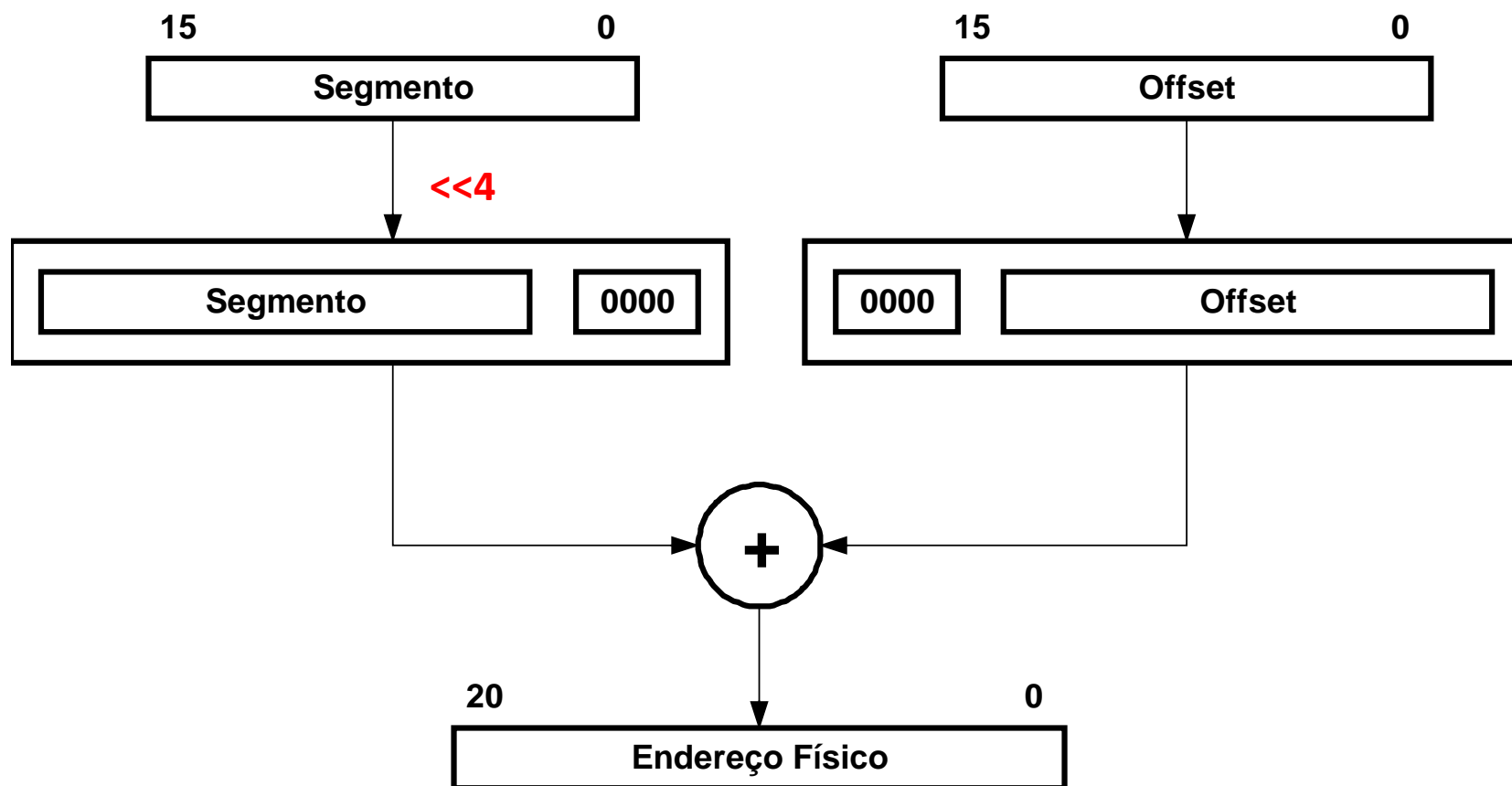
Geração de Endereços



Formação do Endereço

- Segmentos
 - Tamanho: 2^{16} bytes
 - Não existe separação entre os segmentos
 - Os segmentos podem ser **sobrepostos**
- Capacidade de acesso
 - 20 bits (1 Mbyte)
- Cada instrução faz uso, **implícito**, de um segmento
 - Pode-se alterar este segmento na própria instrução
- O **offset** a ser usado também depende do tipo de acesso
 - Busca de instrução
 - Segmento: CS; Offset: IP
 - Manipulação de pilha
 - Segmento: SS; Offset: SP
 - Manipulação de strings
 - Segmento: DS; offset SI
 - Segmento: ES; offset DI

Cálculo do Endereço Físico



Modos de Endereçamento

- Registrador
 - O operando é um dos registradores (8 ou 16 bits)
 - Reg8 = AL, CL, DL, BL, AH, CH, DH, BH
 - Reg16 = AX, CX, DX, BX, SP, BP, SI, DI
- Imediato
 - O operando é uma constante (8 ou 16 bits)
 - Im8 e Im16

Modos de Endereçamento

- Endereço da Memória (EA – Effective Address)
 - Absoluto (direto)
 - Valor constante na instrução
 - Indireto
 - Usando o Registrador como endereço
 - Indexado
 - Constante + Registrador = vetores constantes
 - Registrador + Registrador = vetores (ponteiros)
 - Constante + Registrador + Registrador = matrizes
 - Na realidade, existem 17 modos
 - São todas as combinações possíveis (ver a seguir)

Endereços da Memória (EA)

- Determina o EA (Effective Address)
 - $EA = DISP + BASE + (INDEX * Scale)$
- Descrição
 - *DISP*: um valor fixo de deslocamento
 - *BASE*: um valor em um registro de uso geral
 - BX e BP
 - *INDEX*: um valor em um registro de uso geral
 - SI e DI
- Formas de combinação

	[BX]	[SI]
DISP	[BP]	[DI]

Ver todas as combinações possíveis, a seguir

17 Combinações

- DISP
- [BX], [BP] , [SI] , [DI]
- [DISP+BX], [DISP+BP] , [DISP+SI] , [DISP+DI]
- [BX+SI], [BX+DI], [BP+SI], [BP+DI]
- [DISP+BX+SI], [DISP+BX+DI]
- [DISP+BP+SI], [DISP+BP+DI]

DISP	[BX]	[SI]
	[BP]	[DI]

Exemplo – instrução MOV

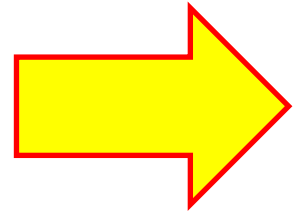
- Imediato
 - MOV AL, 0AFH
 - Constante 0AFH
- Registrador
 - MOV AL, DL
 - Registrador DL
- Absoluto (direto)
 - MOV AX, VARIAVEL
 - *DISP*: VARIAVEL

Mais Exemplos

- Indireto
 - MOV AL, [BX]
 - *BASE*: BX
- Indexado (1)
 - MOV AL, [SI+6]
 - *DISP*: 6
 - *INDEX*: SI
- Indexado (2)
 - MOV AL, [BX+DI+5]
 - *DISP*: 5
 - *BASE*: BX
 - *INDEX*: DI

Tipos de Dados

- Existem “tipos” de dados
 - Interpretados pelo montador
 - Interpretados pelo processador



Tipos de Dados do Montador

- Qual o modo de endereçamento de
 - MOV AX, VAR?
- Depende de como foi definida VAR
 - VAR: EQU 20
 - Gera endereçamento imediato
 - VAR: DW 20
 - Gera endereçamento absoluto

Tipos de Dados do Processador

- 8086
 - 8 bits (byte) e 16 bits (word)
- Organização na memória
 - *Little Endian*

Tipos de Dados Adicionais

- Algumas instruções suportam interpretações adicionais dos tipos básicos
 - Unsigned int: 8 e 16 bits
 - Signed int: 8 e 16 bits
 - Floating point: 32, 64 e 80 bits (usa a FPU)
- Outros tipos
 - Bits
 - String
 - Packed SIMD (Single Instruction Multiple Data)
 - 64-bit Packed SIMD
 - 128-bit Packed SIMD
 - BCD e Packed BCD
 - Floating point

Indicação de endereços (Ponteiros)

- NEAR
 - Representam apenas o OFFSET do endereço
 - Possui 16 bits
 - O SEGMENTO a ser usado dependerá do tipo de acesso
- FAR
 - Representam SEGMENTO e OFFSET do endereço
 - Possuem 16 bits no segmento + 16 bits do offset

Intel 8086 (e 8088)

Prof. Sérgio L. Cechin