

Modo Real

Endereçamento no x86

- O processador usa *byte addressing*
 - A memória é organizada e acessada como uma sequência de bytes
 - A faixa de memória que pode ser acessada é chamada de espaço de endereçamento

Segmentos

- Regiões independentes de memória
- Em modo protegido (32 bits) os segmentos são protegidos quanto à invasão por códigos em outras regiões da memória
- Em modo real, um segmento é um bloco de memória de 64KB, que pode iniciar em qualquer endereço múltiplo de 16 bytes
 - Exemplos: 00000, 00010, 00020, 20000, 8CE90...

Segmentos – Modo real

- Com a memória segmentada, é possível endereçar 1MB de memória em um processador 16bits
 - Como?
 - O endereço em um registrador de segmento é combinado com o endereço em outro registrador de 16-bits para gerar o endereço de 20bits.

Modo Real

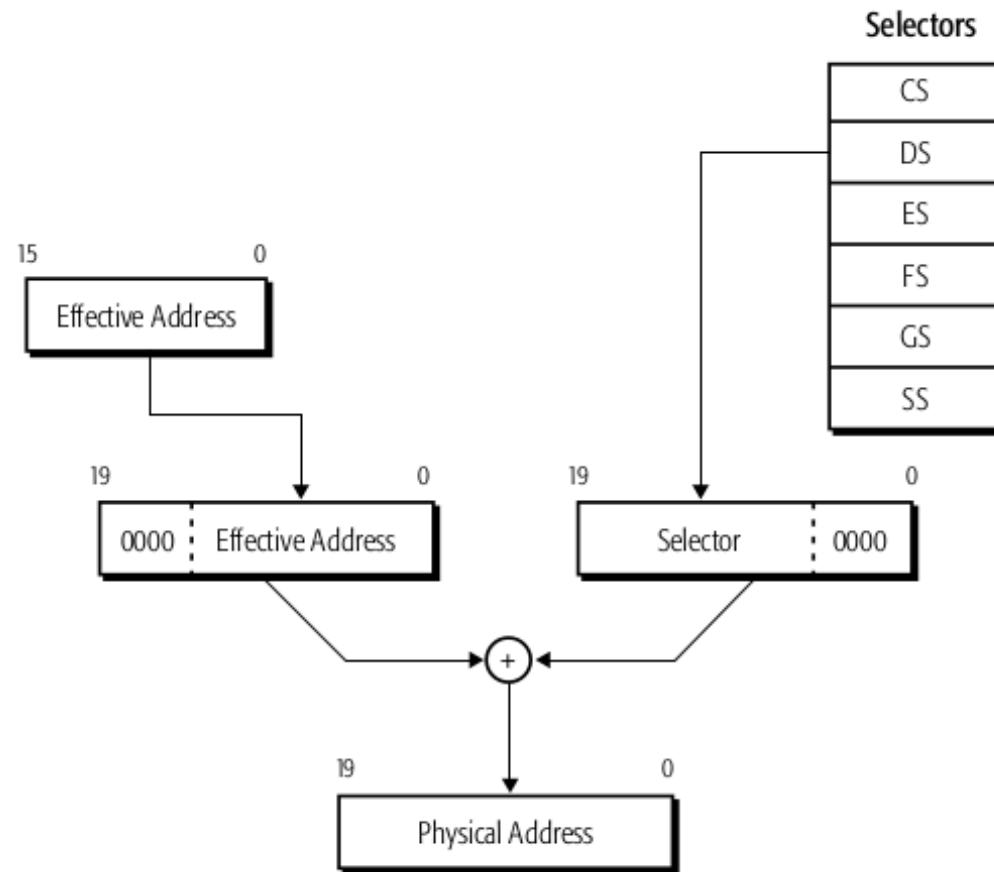
Endereços de 20 bit

CS	<table border="1"><tr><td>A</td><td>0</td><td>0</td><td>0</td></tr></table>	A	0	0	0	A0000
A	0	0	0			
		+ 5F00				

IP	<table border="1"><tr><td>5</td><td>F</td><td>0</td><td>0</td></tr></table>	5	F	0	0	A5F00
5	F	0	0			

- Cada registrador de segmento tem um “uso default”.
- O processador sabe qual registrador de segmento usar para formar o endereço para cada função específica (busca de instrução, operação na pilha etc).
- O processador também permite que o programador diga especificamente qual registrador de segmento usar na geração de alguns endereços.

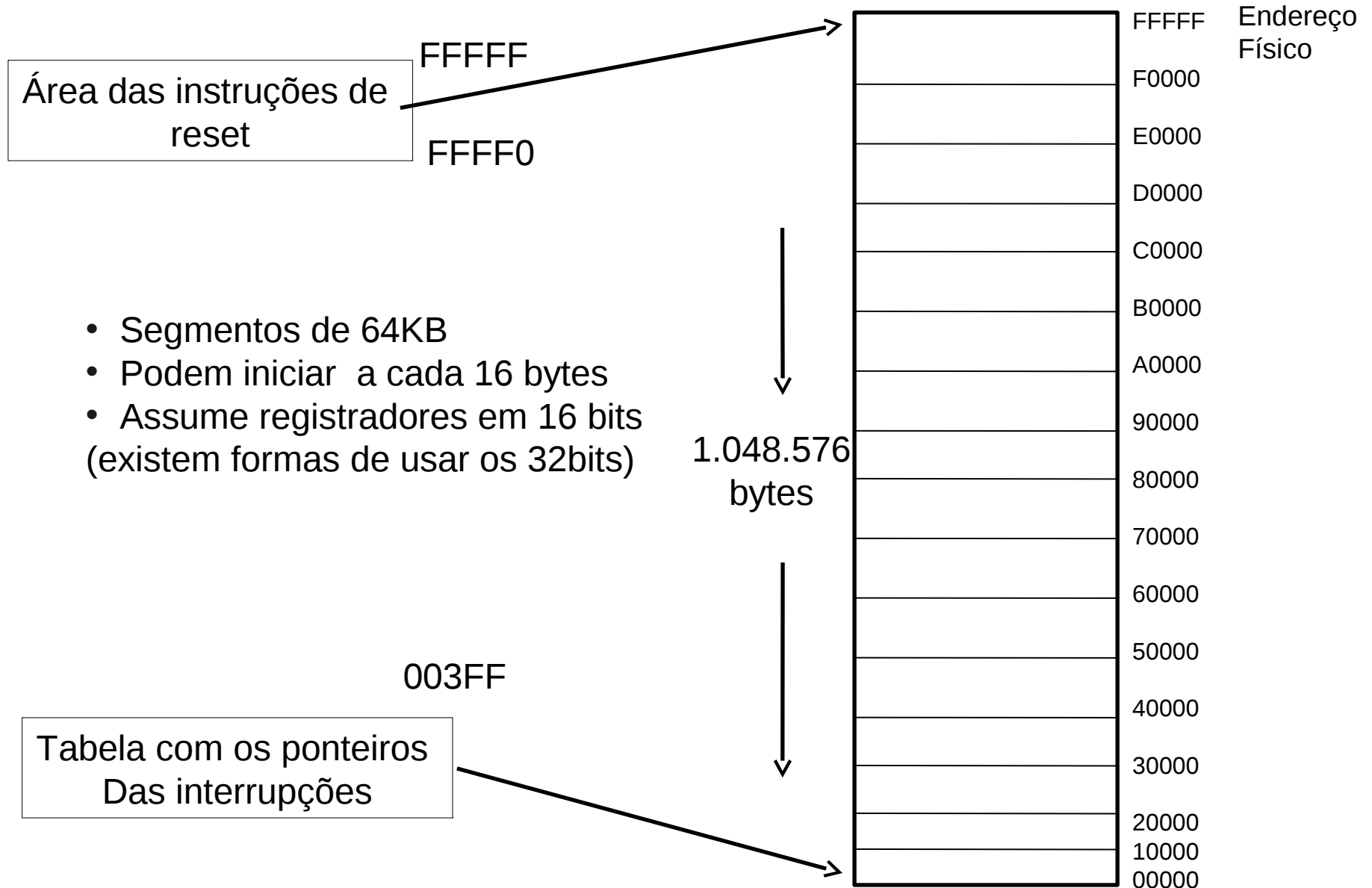
Modelo de memória, modo real



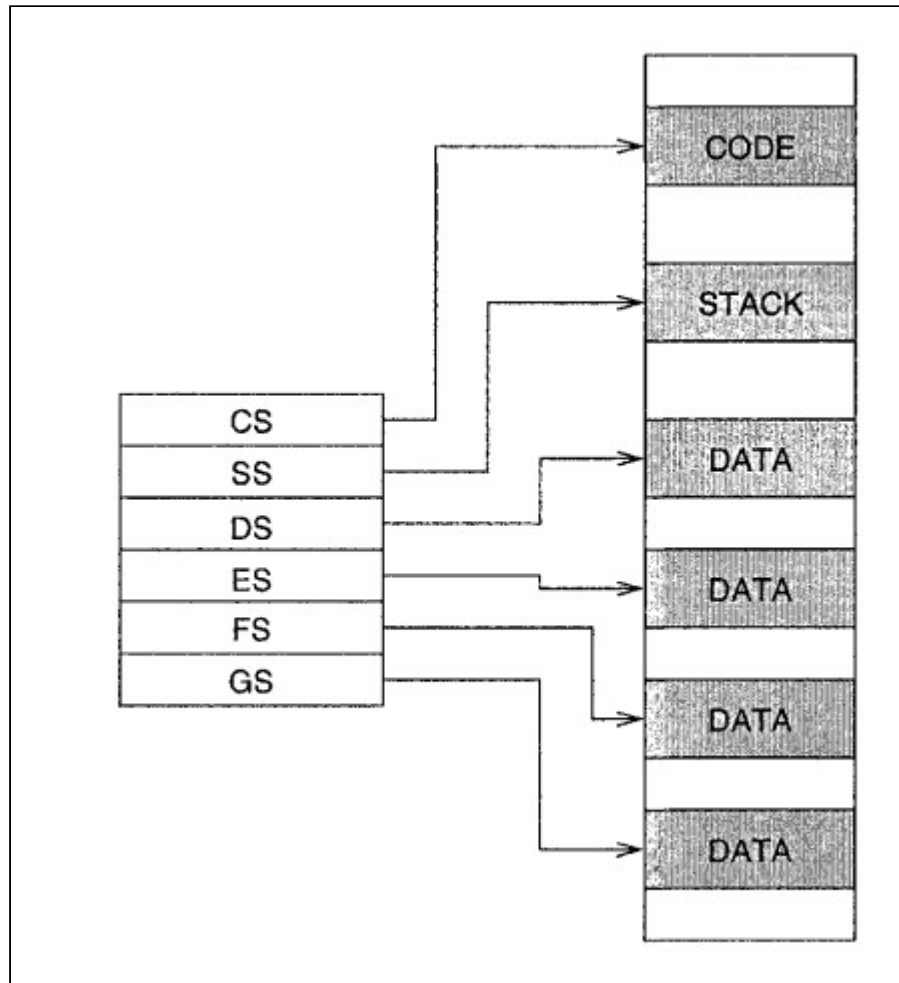
Obs: Registradores de segmento também são chamados na literatura de seletores de segmento

Figure 1-5. Real-Address Memory Model

Espaço de Endereçamento, Modo real



Modo Real



Dados: nomenclatura

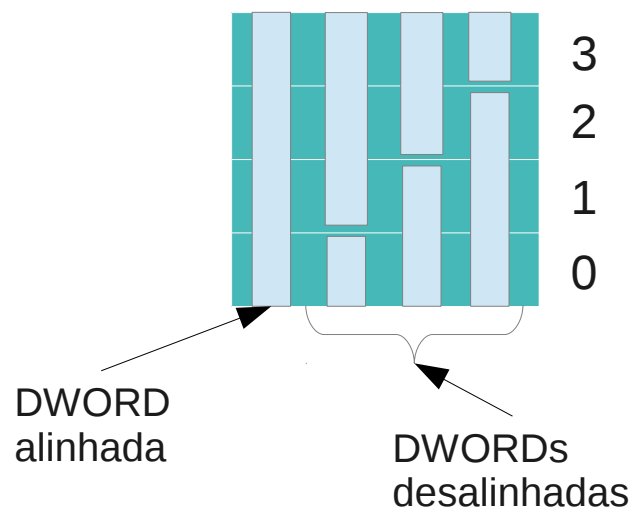
- Byte (8 bits)
- Word (16 bits)
- Double-Word (32 bits)
- Quadword (64 bits)

Questões de revisão

- No modo real, se o registrador CS contém 03e0h e IP contém 1f20h, de qual endereço de memória será obtida a próxima instrução?
- Se o SS contém 0400h e SP contém 3ffeh, onde está localizado o topo da pilha?
- Se um segmento de dados começa no endereço 24000h, qual é o endereço da última posição do segmento?
- Duas posições de memória, começando no endereço 3000h, contém os bytes 34H e 12H. Qual é a word armazenada no endereço 3000h?

Alinhamento de memória

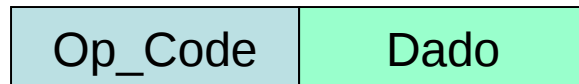
- Como lemos dados que ocupam múltiplos bytes como um load a partir da posição de memória inicial, é possível começar uma DWORD (WORD/QUADWORD) a partir de qualquer posição de memória?



- Manter dados desalinhados na memória normalmente implica em uma queda substancial de performance!
- Exemplo: a pilha deve ser mantida sempre alinhada!

Modos de endereçamento de dados

Imediato



– Ex.: `mov cx,1024H`

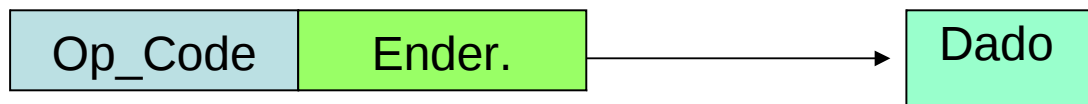
Registrador



– Ex.: `mov al,ah`

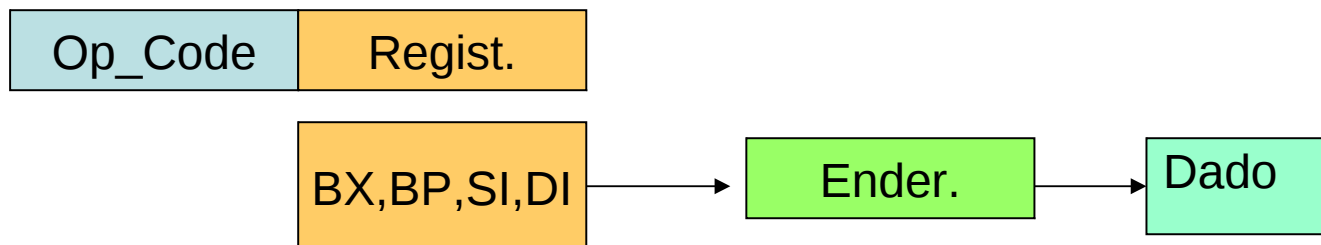
Modos de endereçamento de dados

Direto



- Ex.: `mov al,[0300h]`
`mov ax,counter`

Registrador Indireto

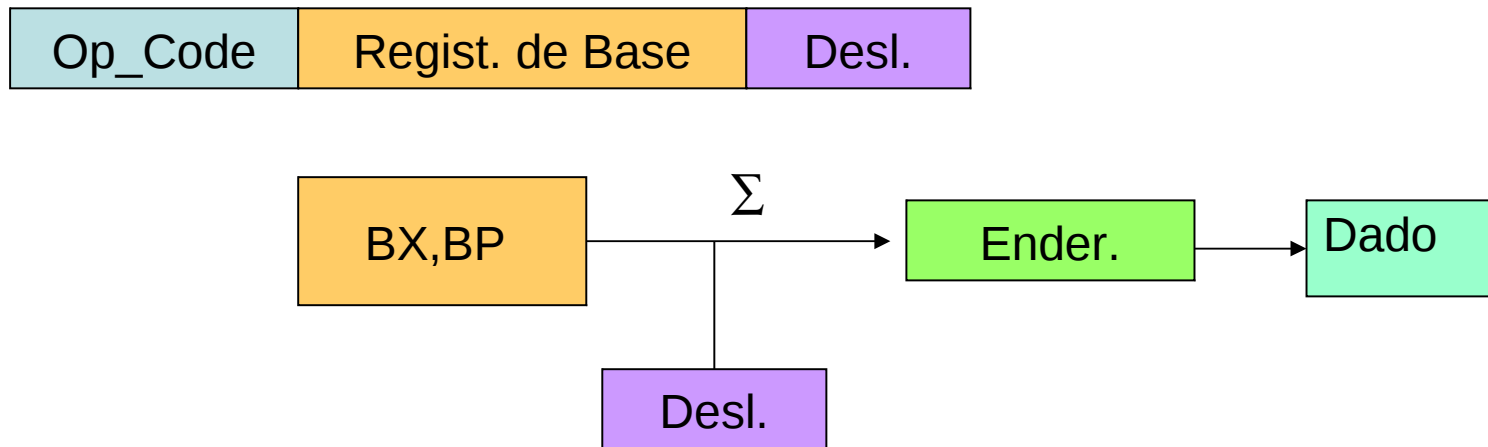


- Ex.: `mov ch,[bx]`

Por padrão, o valor é adicionado a DS
Se for usado BP, o padrão é SS

Modos de endereçamento de dados

Base

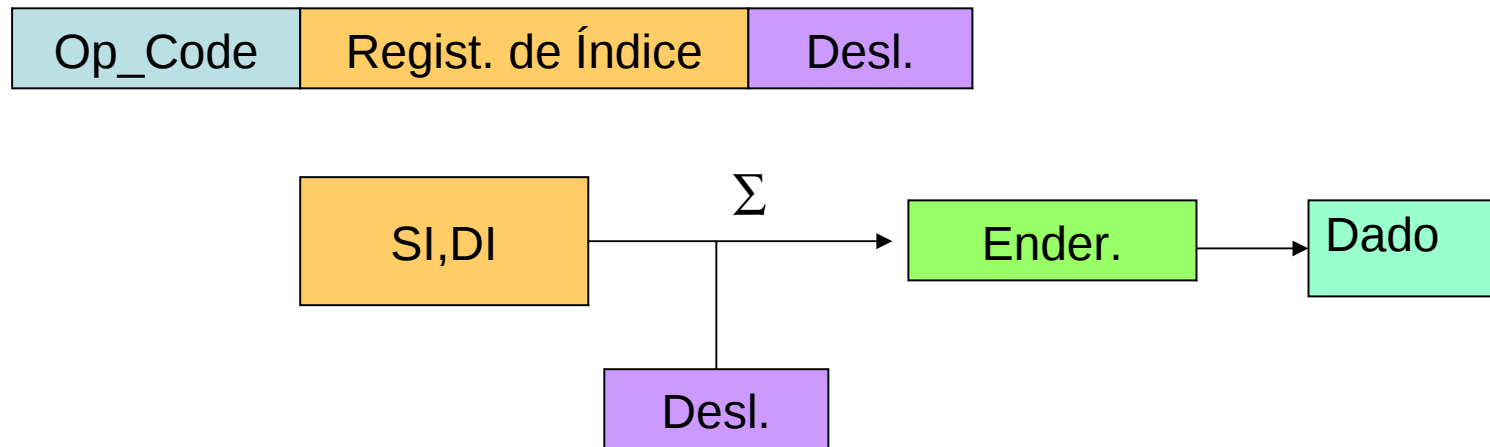


– Ex.: `mov ch,[bx+02]`

Deslocamento de 8 ou 16
Bits [com sinal]

Modos de endereçamento de dados

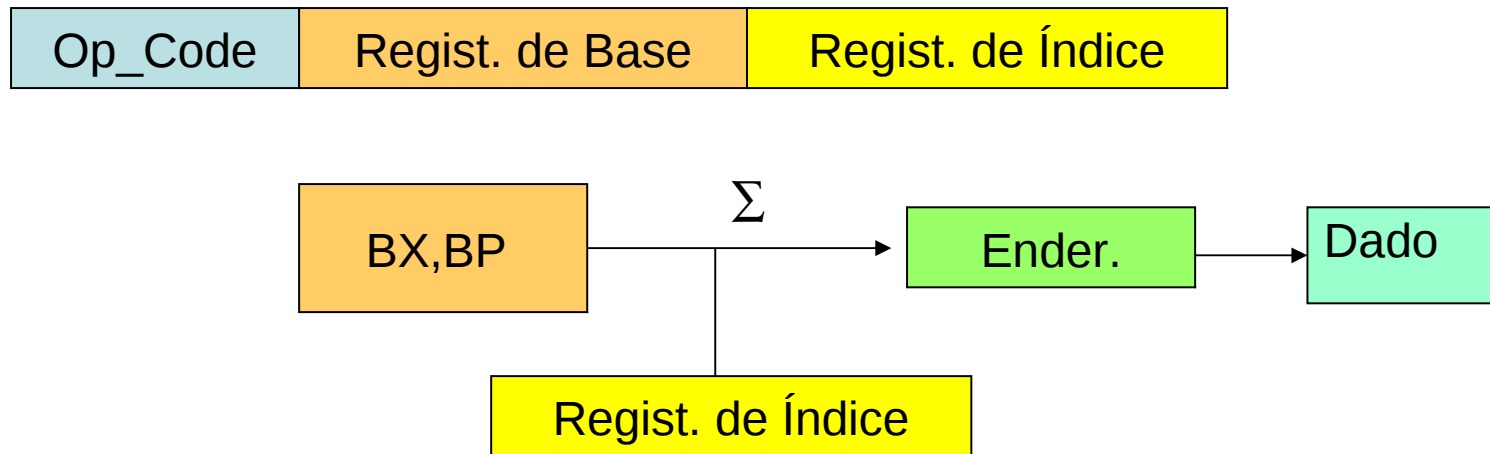
Indexado



– Ex.: `mov ch,[si+02]`

Modos de endereçamento de dados

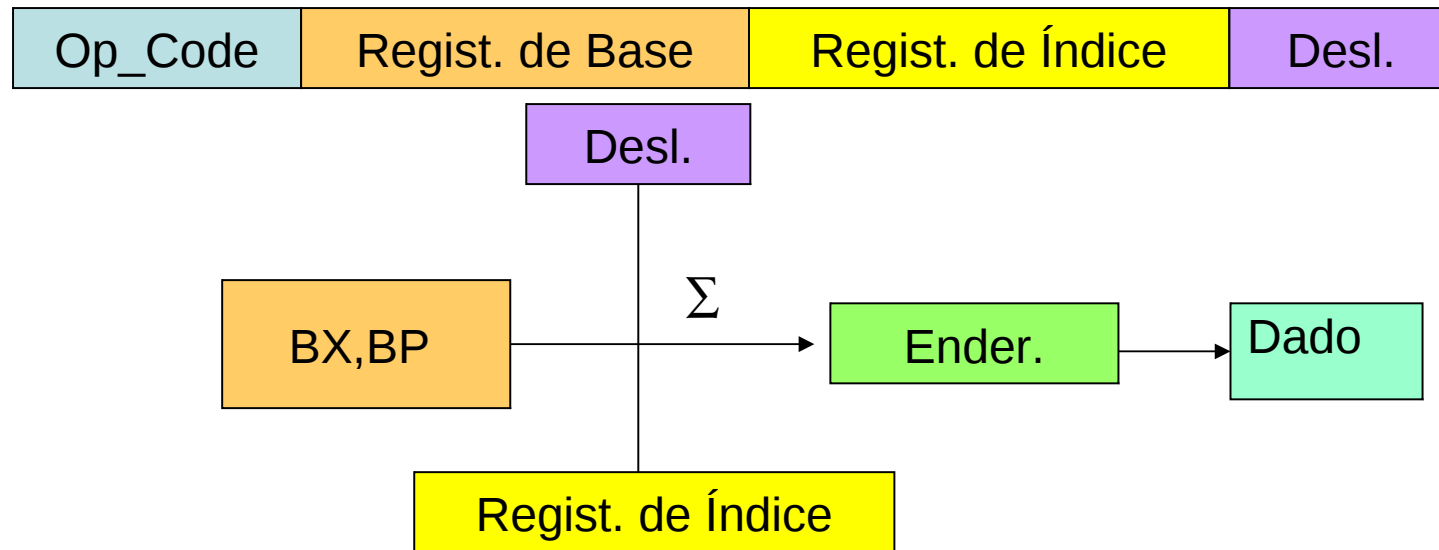
Base Indexado



– Ex.: `mov ch,[bx+si]`

Modos de endereçamento de dados

Base Indexado com Deslocamento



– Ex.: `mov ch,[bx+si+03]`

Modos de endereçamento de dados

Porta



- Ex.: in al,80h ; espaço de
- in al,dx ; endereçamento = 256
- out 90h,al ; espaço de
- out dx,al ; endereçamento = 65536

Neste caso o endereço é de uma porta de IO, e não de uma posição de memória!

Questões de revisão

- Qual a diferença entre
 - MOV AX, 1000H ; e
 - MOV AX,[1000H]
- Qual a diferença entre endereçamento de portas e endereçamento de memória?