

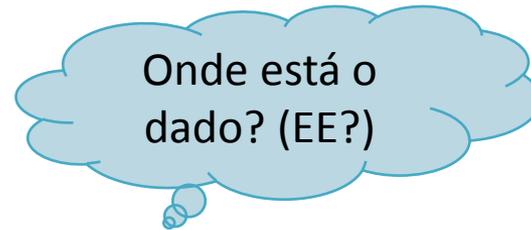
Modos de endereçamento

Considere uma linguagem *Assembly* fictícia na qual o primeiro operando será sempre o registrador destino. Vamos analisar o endereçamento do segundo operando.

Exemplo:

MOV R2 ,#1

R2 ← '1'

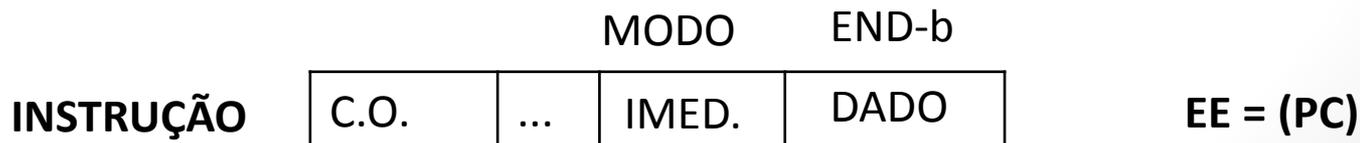


Na própria instrução

EE = (PC)

ENDEREÇAMENTO IMEDIATO

- Neste tipo de endereçamento, o operando, uma constante ou literal, é especificado diretamente no campo de endereço-base da instrução. Assim, o operando faz parte da instrução.
- O endereço efetivo é o conteúdo do próprio registrador PC.

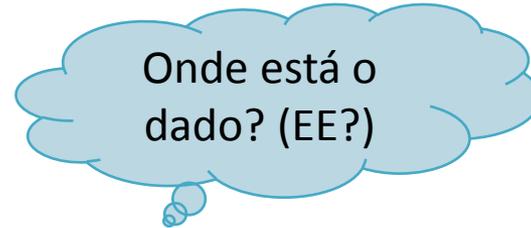


Modos de endereçamento

Exemplo:

MOV R2 , 1

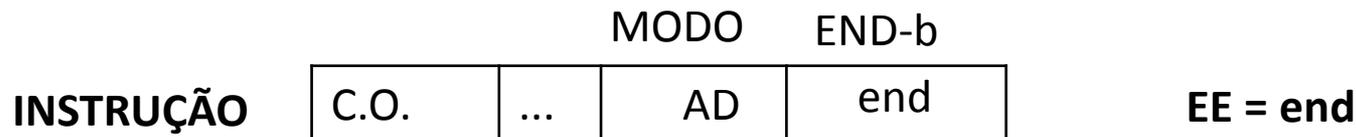
R2 ← (1)



EE = posição de memória 1

ENDEREÇAMENTO ABSOLUTO DIRETO

- O endereço efetivo é o endereço referenciado no campo de endereço do operando.

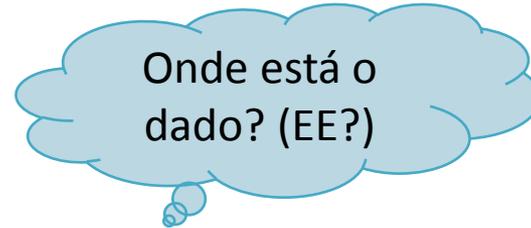


Modos de endereçamento

Exemplo:

MOV R2 ,(1)

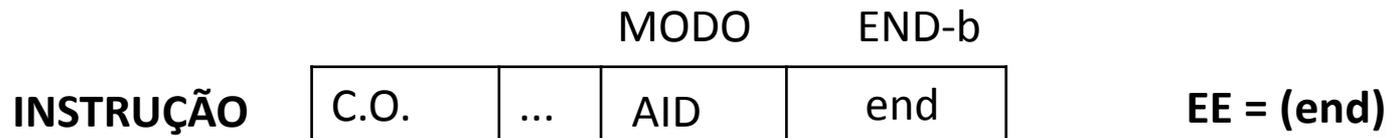
R2 ← ((1))



EE = (posição de memória 1)

ENDEREÇAMENTO ABSOLUTO INDIRETO

- O endereço efetivo está contido na posição de memória cujo endereço é referenciado no campo do operando.

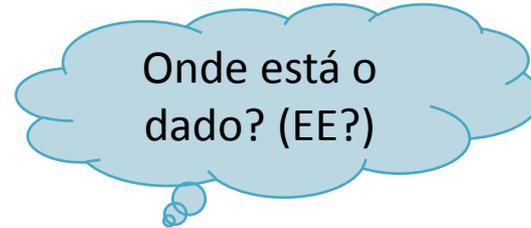


Modos de endereçamento

Exemplo:

MOV R2 ,R1

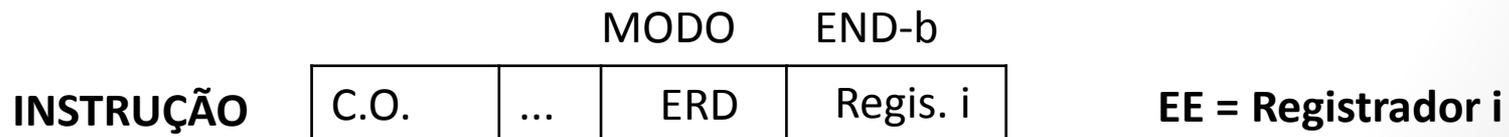
R2 ← (R1)



EE = Registrador R1

ENDEREÇAMENTO POR REGISTRADOR DIRETO

- O **endereço efetivo** é o próprio registrador referenciado no campo de endereço da instrução; o registrador contém o próprio operando.

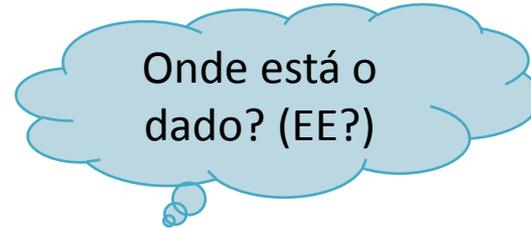


Modos de endereçamento

Exemplo:

MOV R2 ,(R1)

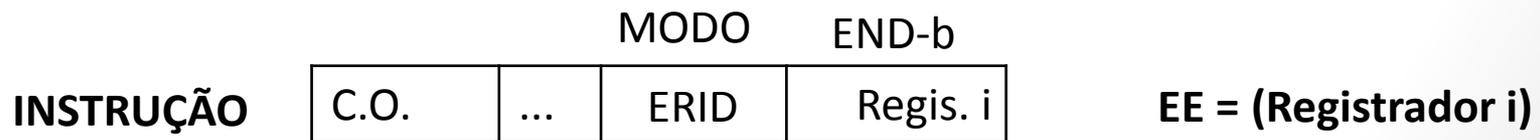
$R2 \leftarrow ((R1))$



EE = (Registrador R1)

ENDEREÇAMENTO POR REGISTRADOR INDIRETO

- O **endereço efetivo** está contido no registrador referenciado no campo de endereço do operando.

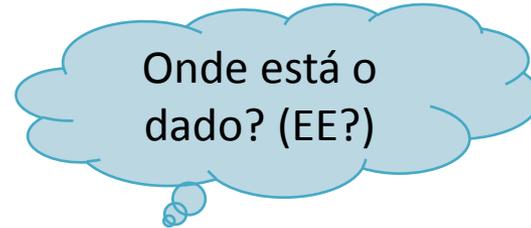


Modos de endereçamento

Exemplo:

MOV R2 , +8(PC)

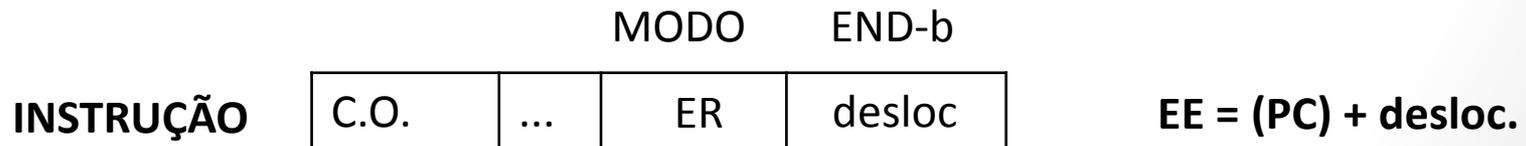
$R2 \leftarrow ((PC) + 8)$



$EE = (PC) + 8$

ENDEREÇAMENTO RELATIVO

- O endereço efetivo é calculado pela adição do conteúdo do **registrador PC** com um **deslocamento** contido no campo de endereço da instrução.



Modos de endereçamento

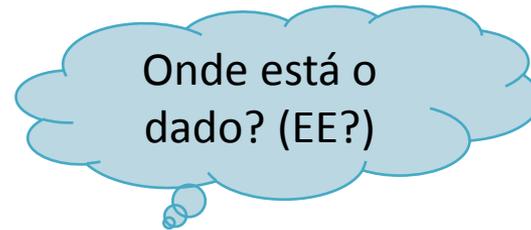
Note que estamos considerando o EE referente ao primeiro operando.

Exemplo:

MOV (R2)+ , R3

$(R2) \leftarrow (R3)$

$R2 \leftarrow (R2) + 1$



EE = (Registrador R2)

ENDEREÇAMENTO AUTO-INCREMENTO

- O **endereço efetivo** é incrementado durante o ciclo de acesso. No auto-incremento, após calculado o endereço efetivo, seu conteúdo é incrementado.

	MODO		END-b	
INSTRUÇÃO	C.O.	...	EAI	Regist. i

EE = (Regist. i)

Modos de endereçamento

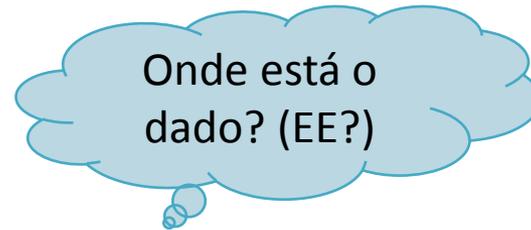
Note que estamos considerando o EE referente ao primeiro operando.

Exemplo:

MOV -(R2) , R3

$R2 \leftarrow (R2) - 1$

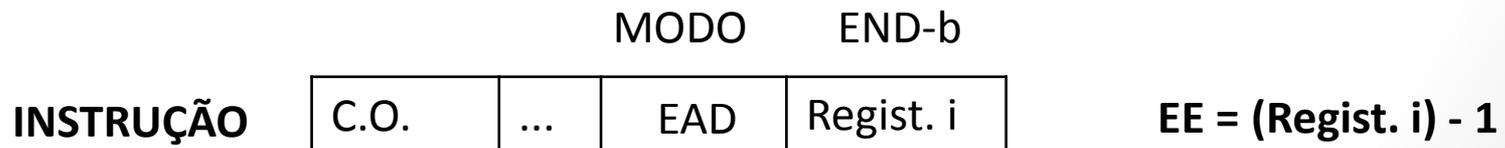
$(R2) \leftarrow (R3)$



$EE = (\text{Registrador } R2) - 1$

ENDEREÇAMENTO AUTO-DECREMENTO

- Antes de se determinar o endereço efetivo, o conteúdo do registrador é decrementado.



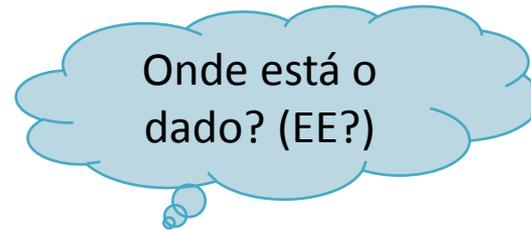
Modos de endereçamento

Considere que nosso processador possua um **Registrador Indexador (RIX)**, cuja função é definir um deslocamento para o endereço-base da instrução.

Exemplo:

MOV R2 , RIX(100)

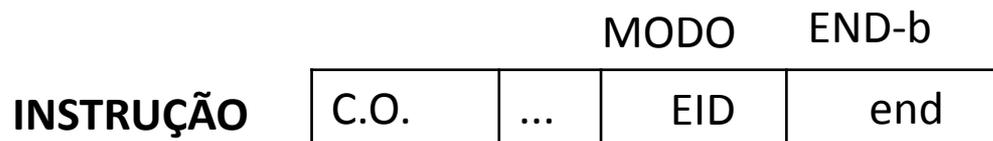
$R2 \leftarrow (100 + (RIX))$



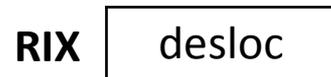
$EE = 100 + (RIX)$

ENDEREÇAMENTO INDEXADO DIRETO

- No modo indexado direto o conteúdo do registrador RIX é somado ao endereço-base para determinação do endereço efetivo.



$EE = \text{desloc} + \text{end}$



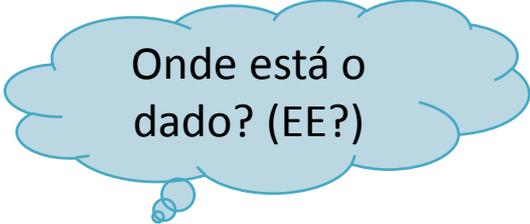
Modos de endereçamento

Considere que nosso processador possua um **Registrador Indexador (RIX)**, cuja função é definir um deslocamento para o endereço-base da instrução.

Exemplo:

MOV R2 ,(RIX(100))

$R2 \leftarrow ((100 + (RIX)))$



Onde está o dado? (EE?)

$EE = (100 + (RIX))$

ENDEREÇAMENTO INDEXADO INDIRETO

- PRÉ-INDEXADO: o deslocamento contido no registrador indexador é somado ao endereço base antes da busca do endereço indireto.

	MODO			END-b
INSTRUÇÃO	C.O.	...	EIIDPRE	End

$EE = (\text{desloc} + \text{end})$

RIX	desloc
-----	--------

Modos de endereçamento

Considere que nosso processador possua um **Registrador Indexador (RIX)**, cuja função é definir um deslocamento para o endereço-base da instrução.

Exemplo:

MOV R2 , RIX((100))

$R2 \leftarrow ((100) + (RIX))$

Onde está o dado? (EE?)

$EE = (100) + (RIX)$

ENDEREÇAMENTO INDEXADO INDIRETO

- PÓS-INDEXADO: o deslocamento contido no registrador indexador é somado ao conteúdo da posição de memória dada pelo endereço-base (indireto).

			MODO	END-b
INSTRUÇÃO	C.O.	...	EIIDPOS	end

RIX desloc

$EE = \text{desloc} + (\text{end})$

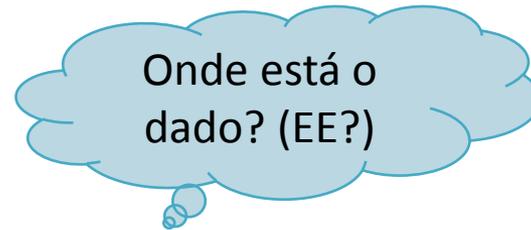
Modos de endereçamento

Agora, além do RIX, nosso processador possui um **Registrador de Base (RB)**, de forma que o endereço-base estará contido nele.

Exemplo:

MOV R2 , 8(RB)

$R2 \leftarrow ((RB) + 8)$



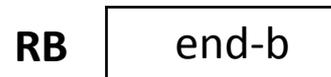
$EE = (RB) + 8$

ENDEREÇAMENTO BASEADO

- O endereço efetivo é calculado com a adição do conteúdo de um **Registrador de Base (RB)** a um **deslocamento** contido no campo de endereço da instrução. O deslocamento é interpretado como um número com sinal, permitindo um deslocamento, positivo ou negativo, em relação à posição dada pelo conteúdo do registrador RB.



$EE = \text{end-b} + \text{desloc}$



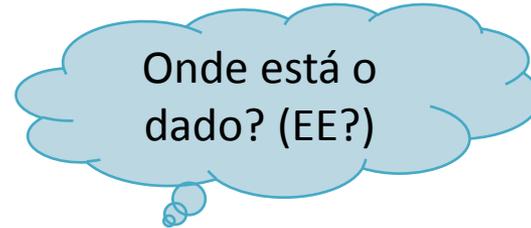
Modos de endereçamento

Agora, além do RIX, nosso processador possui um **Registrador de Base (RB)**, de forma que o endereço-base estará contido nele.

Exemplo:

MOV R2 , RIX(RB)

$R2 \leftarrow ((RIX) + (RB))$



$EE = (RIX) + (RB)$

ENDEREÇAMENTO BASEADO E INDEXADO

- O **endereço efetivo** é calculado com a **adição** do conteúdo de um Registrador de Base (**RB**) com o conteúdo do Registrador Indexador (**RIX**).
- Este é o modo de endereçamento mais versátil de todos. Qualquer um dos exemplos de utilização apresentados pode ser implementado com este modo de endereçamento.

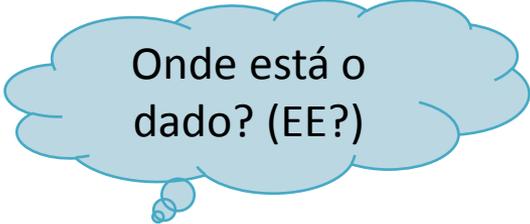
INSTRUÇÃO	MODO					END-b
	C.O.	...	EBI	RIX	RB	
						$EE = (RIX) + (RB)$

Modos de endereçamento

Exemplo:

NEG

$$\text{Acc} \leftarrow \overline{\text{Acc}}$$



Onde está o dado? (EE?)

$$\text{EE} = \text{Acc}$$

ENDEREÇAMENTO INERENTE OU IMPLÍCITO

- O próprio código de operação identifica o endereço dos dados a serem manipulados.
- O endereço do operando não é especificado na instrução, é implícito.

Modos de endereçamento

ENDEREÇAMENTO POR PILHA

- Neste caso, o apontador de pilha **SP** indica o endereço efetivo do dado.

- Retirando dados da pilha:**

POP R1 $SP \leftarrow (SP) - 1$
 $R1 \leftarrow ((SP))$

$EE = (SP) - 1$

- Inserindo dados na pilha:**

PUSH R1 $(SP) \leftarrow (R1)$
 $SP \leftarrow (SP) + 1$

$EE = (SP)$

Este tipo de endereçamento é muito utilizado para **salvar** o conteúdo de registradores e posições de memória que posteriormente serão recarregados em suas posições originais.

Uso típico: **subrotinas**.

Conclusões

- Ao escrever ou ler uma instrução *Assembly*, precisamos conhecer onde estão os dados que estamos manipulando.
- A instrução nos fornece o **endereço-base**, ou seja, um conteúdo que irá nos auxiliar a descobrir qual é o endereço efetivo do dado. Além disso, ela nos fornece os “**rótulos**” que determinarão onde estão os dados.
- Há diversos modos de combinar o endereço-base com os rótulos a fim de se acessar os dados. Cada forma define um **modo de endereçamento**.
- Processadores CISC possuem muitos modos de endereçamento já que não há restrição no acesso à memória.
- Por sua vez, os processadores RISC possuem poucos modos de endereçamento já que o acesso a memória se dá apenas por instruções especiais. Tal fato facilita a programação.
- Por fim, um último modo de endereçamento comumente utilizado nos processadores é o **endereçamento paginado**. Tal forma será detalhadamente estudada na disciplina de EA876, no contexto de sistemas operacionais. Aqui daremos uma visão inicial a seguir.