

Arquitetura e Organização de Computadores

Conjunto de Instruções

Givanaldo Rocha de Souza

<http://docente.ifrn.edu.br/givanaldorochoa>
givanaldo.rocha@ifrn.edu.br

Introdução

- Se um programador estiver usando uma linguagem de alto nível, muito pouco da arquitetura da máquina básica é visível
- Um limite onde o projetista de computador e o programador podem ver a mesma máquina é o conjunto de instruções de máquina

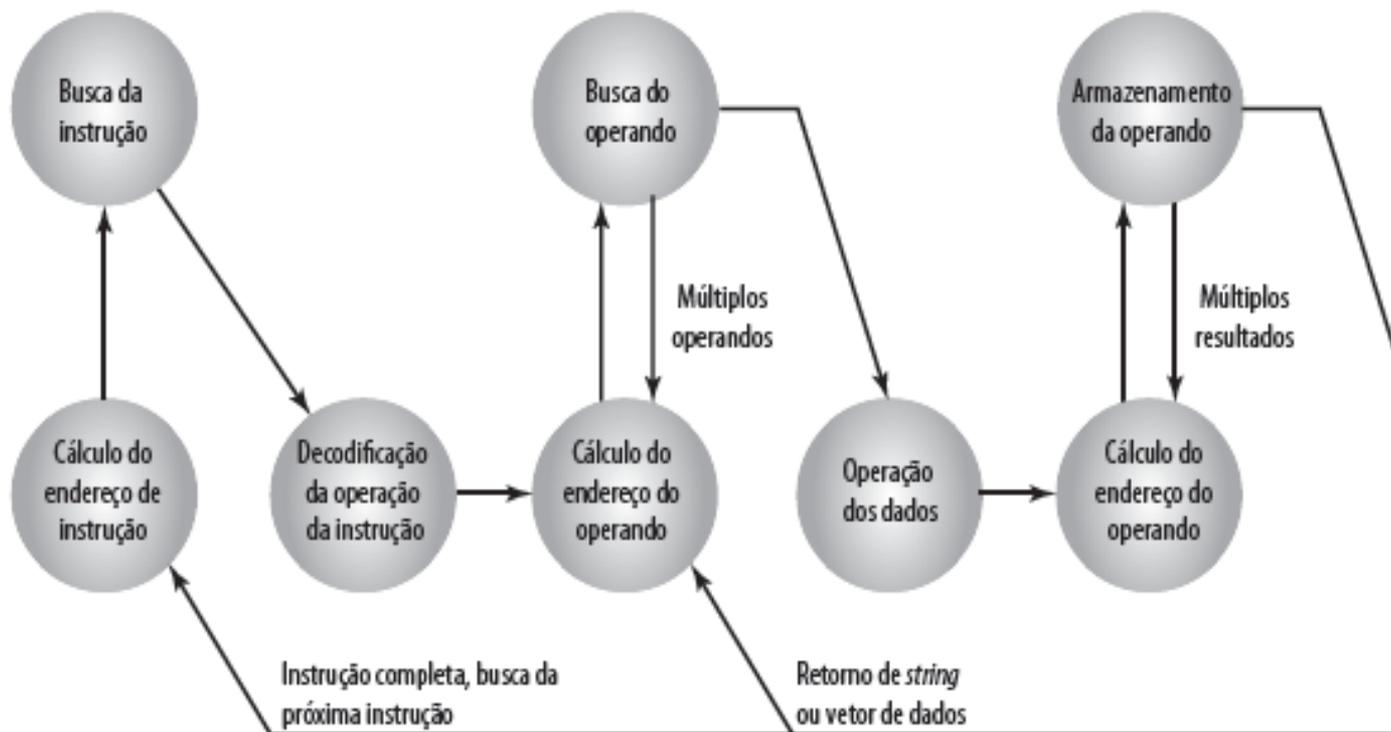
O que é um conjunto de instruções?

- A coleção completa de instruções que são entendidas por uma CPU.
- Código de máquina.
- Binário.
- Normalmente, representado por códigos em *assembly*.

Elementos de uma instrução

- Código de operação (Op code):
 - Faça isto.
- Referência a operando fonte:
 - Nisto.
- Referência a operando de destino:
 - Coloque a resposta aqui.
- Referência à próxima instrução:
 - Quando tiver feito isso, faça isto...

Diagrama de estado do ciclo de instrução



Operandos fonte e destino

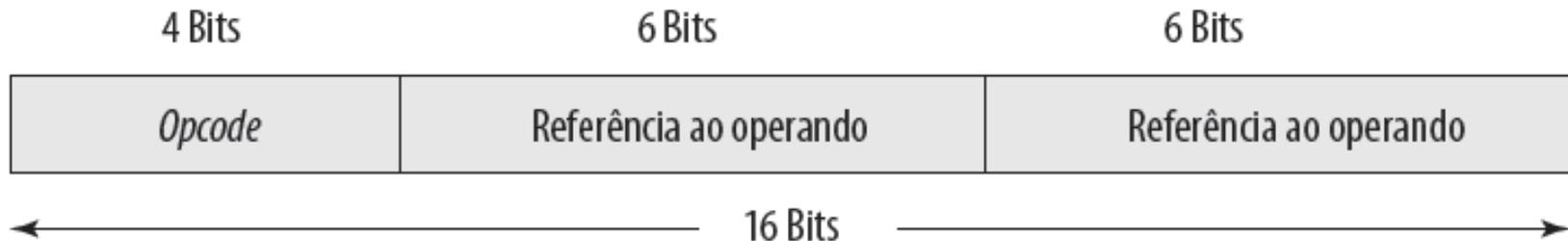
- Memória principal ou virtual
 - Como as referências à próxima instrução
- Registrador do processador
 - Com raras exceções, um processador contém um ou mais registradores que podem ser referenciados por instruções. Se houver apenas 1 registrador, a referência é implícita
- Imediato
 - O valor do operando está contido na instrução
- Dispositivo de E/S
 - A instrução precisa especificar o módulo de E/S

Representação da instrução

- Em código de máquina, cada instrução tem um padrão de bits exclusivo.
- Para consumo humano (bem, para programadores), uma representação simbólica é utilizada.
 - Ex: ADD, SUB, LOAD.
- Operandos também podem ser representados desta maneira:
 - ADD A,B.

Representação da instrução

- Formato de instrução simples
 - Registrador de instruções (IR) do processador



Tipos de instrução

- Processamento de dados.
 - Aritmética e lógica
- Armazenamento de dados
 - Movimentação de dados para dentro ou fora do registrador e/ou locais de memória
- Movimentação de dados (E/S).
 - Instruções e E/S
- Controle de fluxo do programa.
 - Instruções de teste e desvio

Tipos de instrução

- Considere uma instrução em linguagem de alto nível $X = X+Y$
- Como isso poderia ser feito com instruções de máquina? Supor X e Y como 513 e 514
 1. Carregue um registrador com o conteúdo do local de memória 513
 2. Some o conteúdo do local de memória 514 ao registrador
 3. Armazene o conteúdo do registrador no local de memória 513

Números de endereços

- 3 endereços:
 - Operando 1, Operando 2, Resultado.
 - $a = b + c$.
 - Pode ser uma instrução for-next (normalmente implícita).
 - Não é comum.
 - Precisa de palavras muito longas para manter tudo.
- 2 endereços:
 - Um endereço servindo como operando e resultado.
 - $a = a + b$.
 - Reduz tamanho da instrução.
 - Requer algum trabalho extra.
 - Armazenamento temporário para manter alguns resultados.

Números de endereços

- 1 endereço:
 - Segundo endereço implícito.
 - Normalmente, um registrador (acumulador).
 - Comum nas primeiras máquinas.
- 0 (zero) endereços:
 - Todos os endereços implícitos.
 - Usa uma pilha.
 - Ex: push a.
 - push b.
 - add.
 - pop c.

 - $c = a + b.$

Números de endereços

- Mais endereços:
 - Instruções mais complexas.
 - Mais registradores.
 - Operações entre registradores são mais rápidas.
 - Menos instruções por programa.
- Menos endereços:
 - Instruções menos complexas.
 - Mais instruções por programa.
 - Busca/execução de instruções mais rápida.

Números de endereços

Figura 10.3

Programas para executar $Y = \frac{A - B}{C + (D \times E)}$

Instrução	Comentário
SUB Y, A, B	$Y \leftarrow A - B$
MPY T, D, E	$T \leftarrow D \times E$
ADD T, T, C	$T \leftarrow T + C$
DIV Y, Y, T	$Y \leftarrow Y \div T$

(a) Instruções com três endereços

Instrução	Comentário
MOVE Y, A	$Y \leftarrow A$
SUB Y, B	$Y \leftarrow Y - B$
MOVE Y, D	$T \leftarrow D$
MPY T, E	$T \leftarrow T \times E$
ADD T, C	$T \leftarrow T + C$
DIV Y, T	$Y \leftarrow Y \div T$

(b) Instruções de dois endereços

Instrução	Comentário
LOAD D	$AC \leftarrow D$
MPY E	$AC \leftarrow AC \times E$
ADD C	$AC \leftarrow AC + C$
STOR Y	$Y \leftarrow AC$
LOAD A	$AC \leftarrow A$
SUB B	$AC \leftarrow AC - B$
DIV Y	$AC \leftarrow AC \div T$
STOR Y	$Y \leftarrow AC$

(c) Instruções de um endereço

Números de endereços

Tabela 10.1 Utilização de endereços de instrução (instruções sem desvio)

Número de endereços	Representação simbólica	Interpretação
3	OP A, B, C	$A \leftarrow B \text{ OP } C$
2	OP A, B	$A \leftarrow A \text{ OP } B$
1	OP A	$AC \leftarrow AC \text{ OP } A$
0	OP	$T \leftarrow (T - 1) \text{ OP } T$

AC = acumulador

T = topo da pilha

(T - 1) = segundo elemento da pilha

A, B, C = locais de memória ou registradores

Tipos de operandos

- Endereços.
- Números:
 - Inteiro/ponto flutuante.
- Caracteres:
 - ASCII etc.
- Dados lógicos:
 - Bits ou flags

Tipos de operações

- Transferência de dados.
- Aritmética.
- Lógica.
- Conversão.
- E/S.
- Controle do sistema.
- Transferência de controle.

Tipos de operações

Tabela 10.3 Operações comuns do conjunto de instruções

Tipo	Nome da operação	Descrição
Transferência de dados	Move (transferência)	Transfere palavra ou bloco da origem ao destino
	Store (armazenar)	Transfere palavra do processador para a memória
	Load (carregar)	Transfere palavra da memória para o processador
	Exchange	Troca o conteúdo da origem e do destino
	Clear (reset)	Transfere palavra de 0s para o destino
	Set	Transfere palavra de 1s para o destino
	Push	Transfere palavra da origem para o topo da pilha
	Pop	Transfere palavra do topo da pilha para o destino
Aritmética	Add	Calcula a soma de dois operandos
	Subtract	Calcula a diferença de dois operandos
	Multiply	Calcula o produto de dois operando
	Divide	Calcula o quociente de dois operandos
	Absolute	Substitui o operando pelo seu valor absoluto
	Negate	Troca o sinal do operando
	Increment	Soma 1 ao operando
	Decrement	Subtrai 1 do operando

Tipos de operações

Tipo	Nome da operação	Descrição
Lógica	AND	Realiza o AND lógico
	OR	Realiza o OR lógico
	NOT (complemento)	Realiza o NOT lógico
	Exclusive-OR	Realiza o XOR lógico
	Test	Testa condição especificada; define flag(s) com base no resultado
	Compare	Faz comparação lógica ou aritmética de dois ou mais operandos; define flag(s) com base no resultado
	Set control variables	Classe de instruções para definir controles para fins de proteção, tratamento de interrupção, controle de tempo etc.
	Shift	Desloca o operando para a esquerda (direita), introduzindo constantes na extremidade
	Rotate	Desloca ciclicamente o operando para a esquerda (direita), de uma extremidade à outra

Tipos de operações

- Transferência de dados
 - Especificam:
 - Origem.
 - Destino.
 - Quantidade de dados.
 - Podem ser instruções diferentes para diferentes movimentações.
 - Ou uma instrução e diferentes endereços.

Tipos de operações

- Aritmética
 - Adição, Subtração, Multiplicação, Divisão.
 - Inteiro com sinal.
 - Ponto flutuante.
 - Pode incluir:
 - Incremento ($a++$).
 - Decremento ($a--$).
 - Negação ($-a$).

Tipos de operações

- Lógicas
 - Operações lógicas básicas sobre dados booleanos ou binários
 - NOT: inverte um bit
 - AND, OR, Exclusive-OR (XOR): mais comuns
 - EQUAL: teste binário útil
 - Deslocamento lógicos:
 - Os bits de uma palavra são deslocados para direita ou esquerda
 - O bit deslocado para fora se perde
 - Na outra extremidade, um 0 é deslocado para dentro

Tipos de operações

- Conversão
 - Mudam o formato ou operam sobre o formato dos dados
 - EX: decimal para binário

Tipos de operações

- Entrada/saída
 - Podem ser instruções específicas.
 - Pode ser feita usando instruções de movimentação de dados (mapeadas na memória).
 - Pode ser feita por um controlador separado (DMA).

Tipos de operações

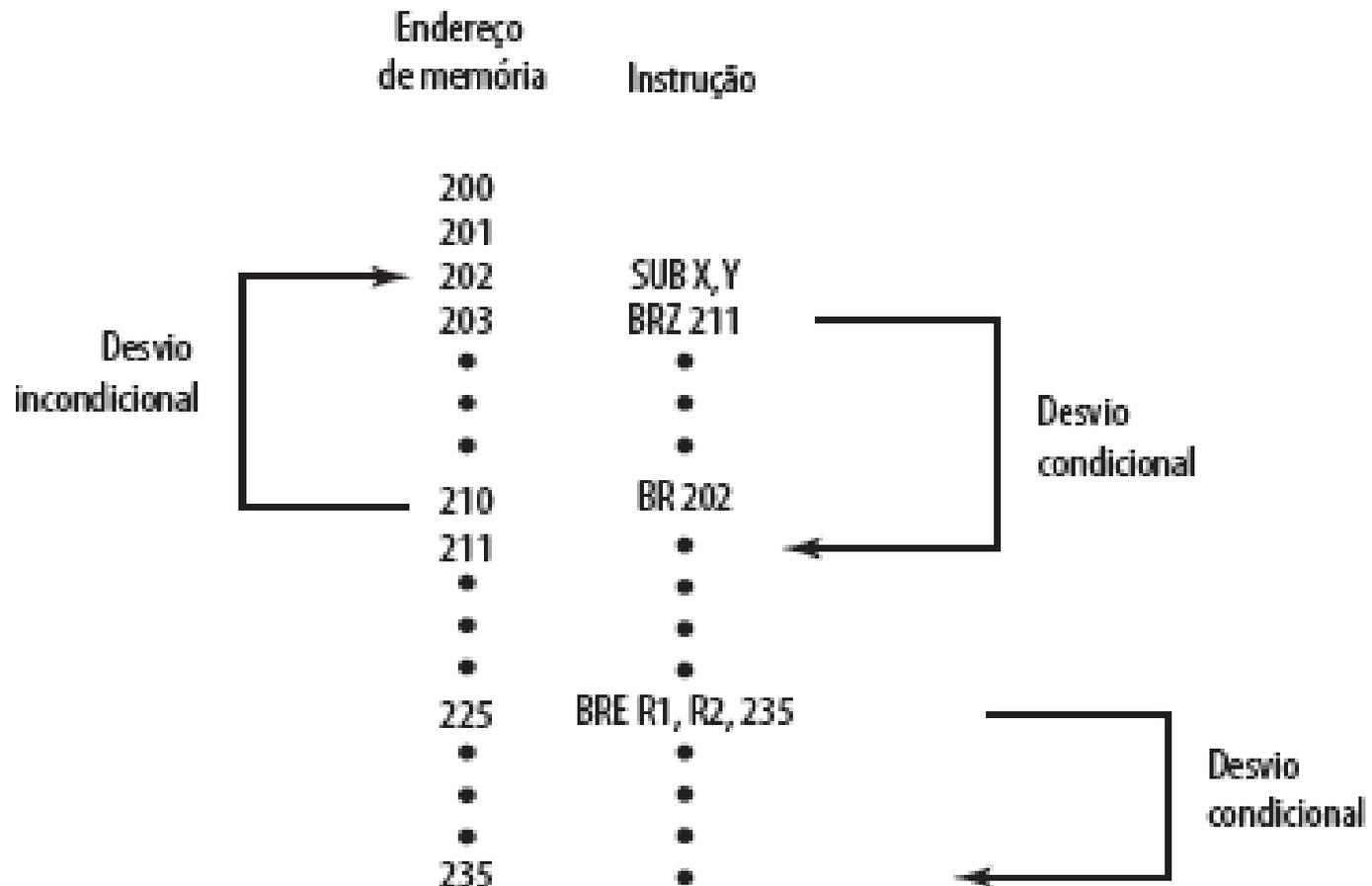
- Controle do sistema
 - Instruções privilegiadas.
 - CPU precisa estar em estado específico:
 - Anel 0 no 80386+.
 - Modo kernel.
 - Para uso dos sistemas operacionais.

Tipos de operações

- Transferência de controle
 - Desvio:
 - Ex: desvio para x se resultado for zero.
 - Salto:
 - Ex: incrementa e salta se for zero.
 - ISZ Registrador 1.
 - Desvia xxxx.
 - ADD A.
 - Chamada de sub-rotina:
 - C.f. chamada de interrupção.

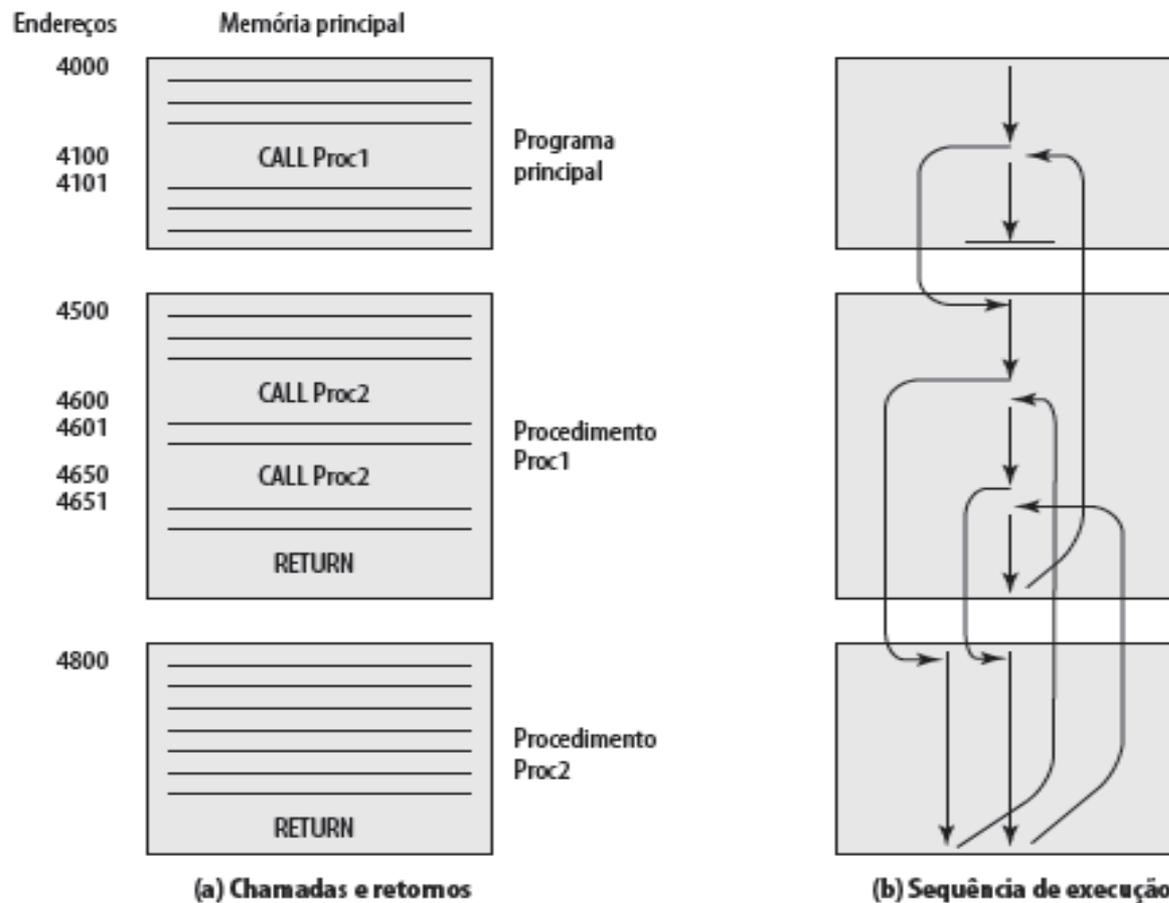
Tipos de operações

- Transferência de controle



Tipos de operações

- Transferência de controle



Modos de Endereçamento

- Os campos de endereços das instruções são relativamente pequenos
- Para possibilitar o acesso a uma grande quantidade de posições da memória, foram criados vários modos de endereçamento
- Quase todas as arquiteturas de computadores fornecem mais de um desses modos de endereçamento

Modos de Endereçamento mais Comuns

- Endereçamento Imediato
- Endereçamento Direto
- Endereçamento Indireto
- Endereçamento de Registrador
- Endereçamento Indireto via Registrador
- Endereçamento por Deslocamento
 - Endereçamento Relativo
 - Endereçamento via Registrador Base
 - Indexação
- Endereçamento à Pilha

Endereçamento Imediato

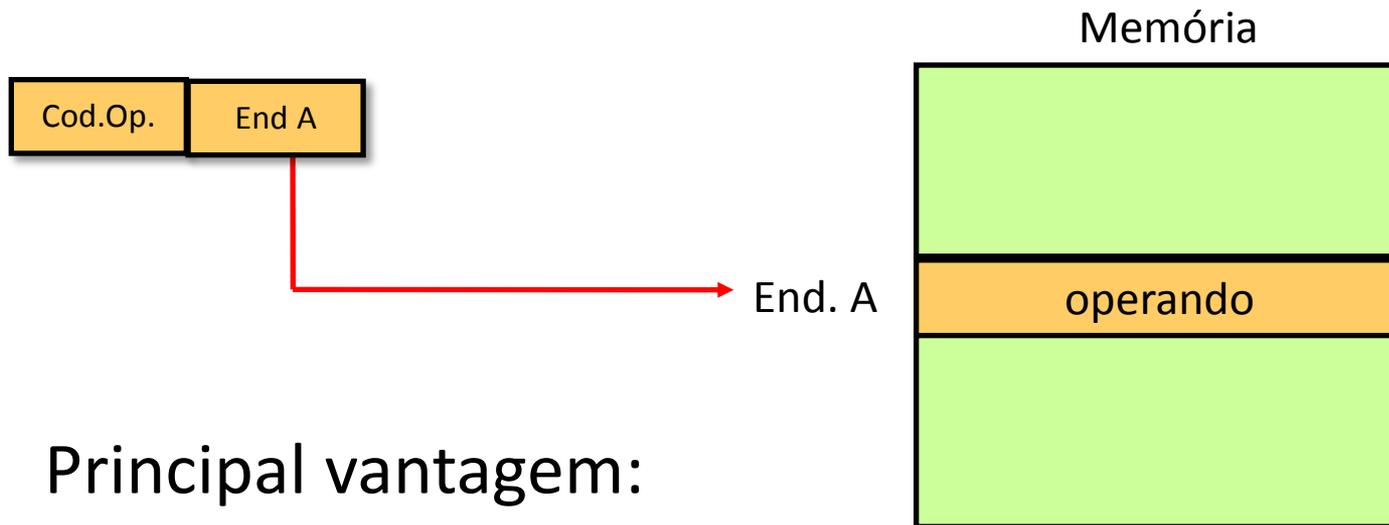
- O valor do operando é especificado diretamente na instrução



- Principal vantagem:
 - Nenhum acesso à memória (economia de tempo)
- Principal desvantagem:
 - O tamanho do operando é limitado pelo tamanho do campo de endereço da instrução

Endereçamento Direto

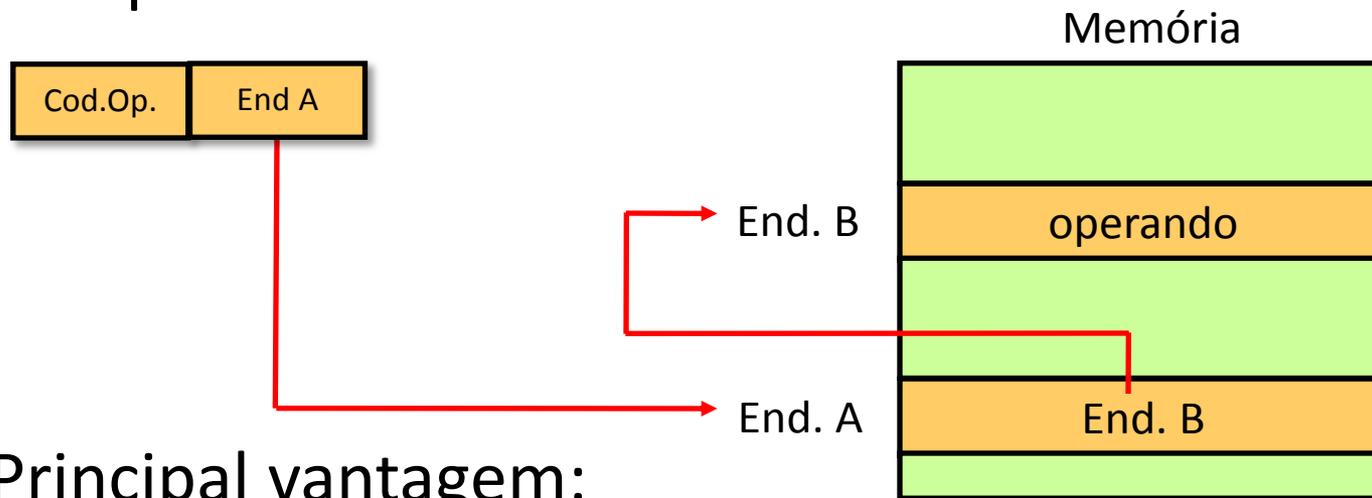
- O campo de endereço da instrução contém o endereço do operando na memória



- Principal vantagem:
 - Simplicidade
- Principal desvantagem:
 - Espaço de endereçamento limitado

Endereçamento Indireto

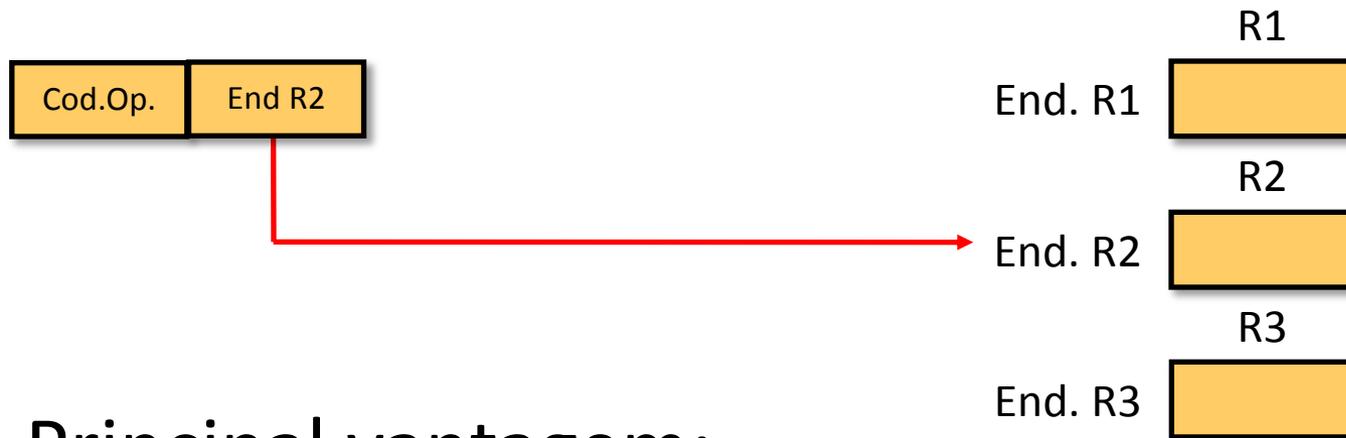
- O campo de endereço da instrução contém um endereço da memória cujo conteúdo é o endereço do operando na memória



- Principal vantagem:
 - Espaço de endereçamento grande
- Principal desvantagem:
 - Acessos múltiplos à memória

Endereçamento de Registrador

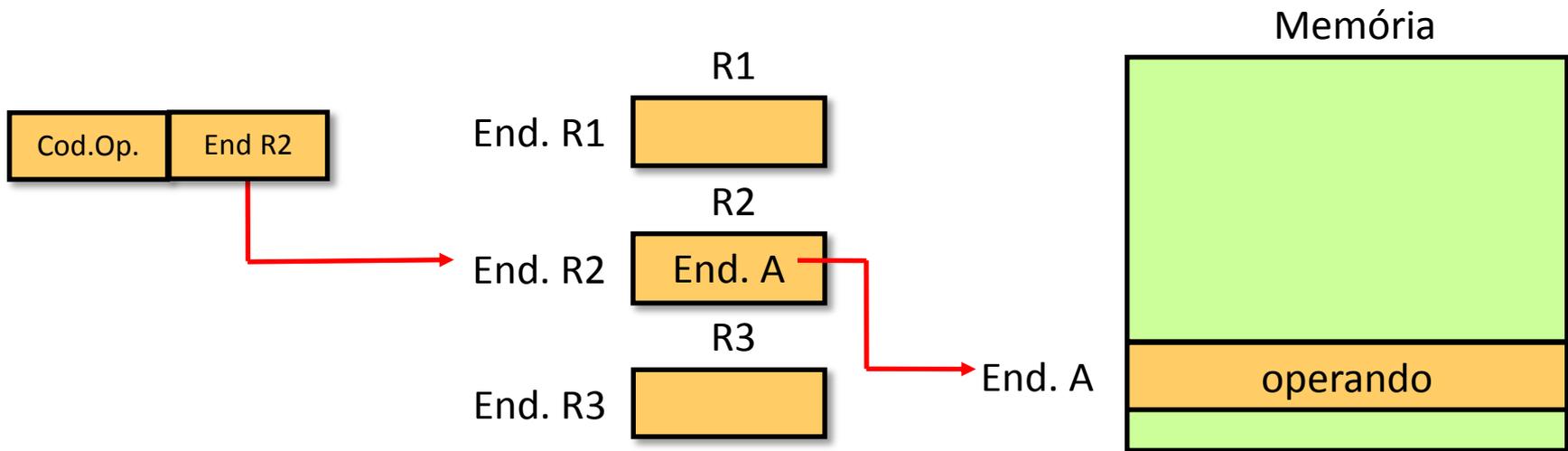
- Semelhante ao endereçamento direto



- Principal vantagem:
 - Nenhum acesso à memória
- Principal desvantagem:
 - Espaço de endereçamento limitado

Endereçamento Indireto via Registrador

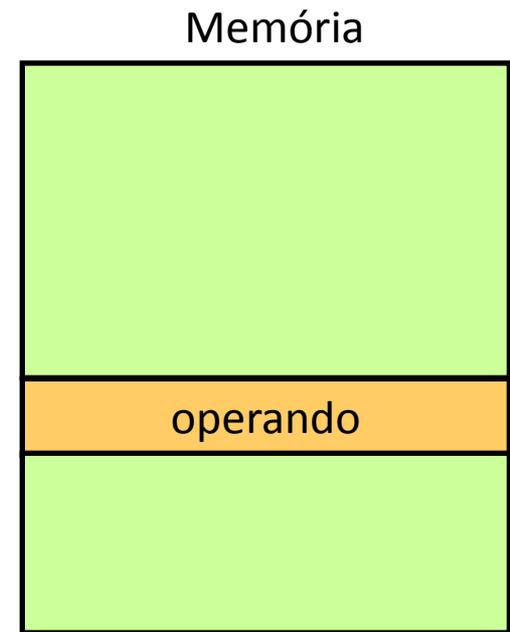
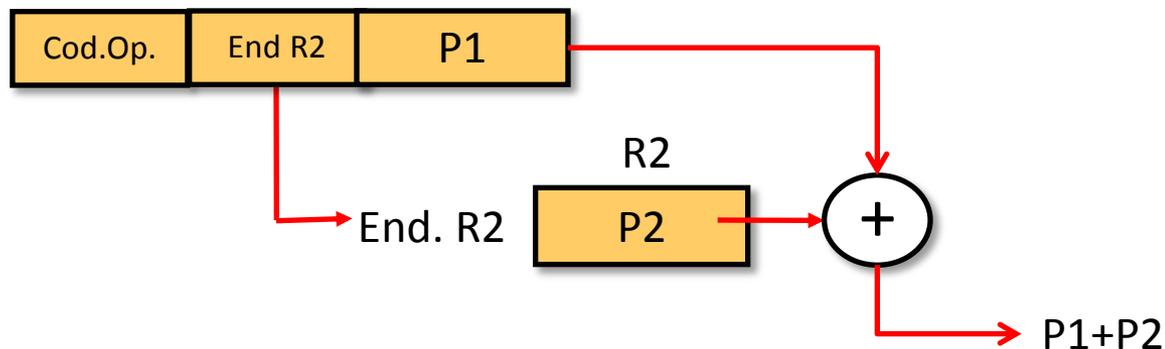
- Semelhante ao endereçamento indireto



- Principal vantagem:
 - Espaço de endereçamento grande
- Principal desvantagem:
 - Acesso extra à memória

Endereçamento por Deslocamento

- É uma combinação dos modos Direto e Indireto via Registrador



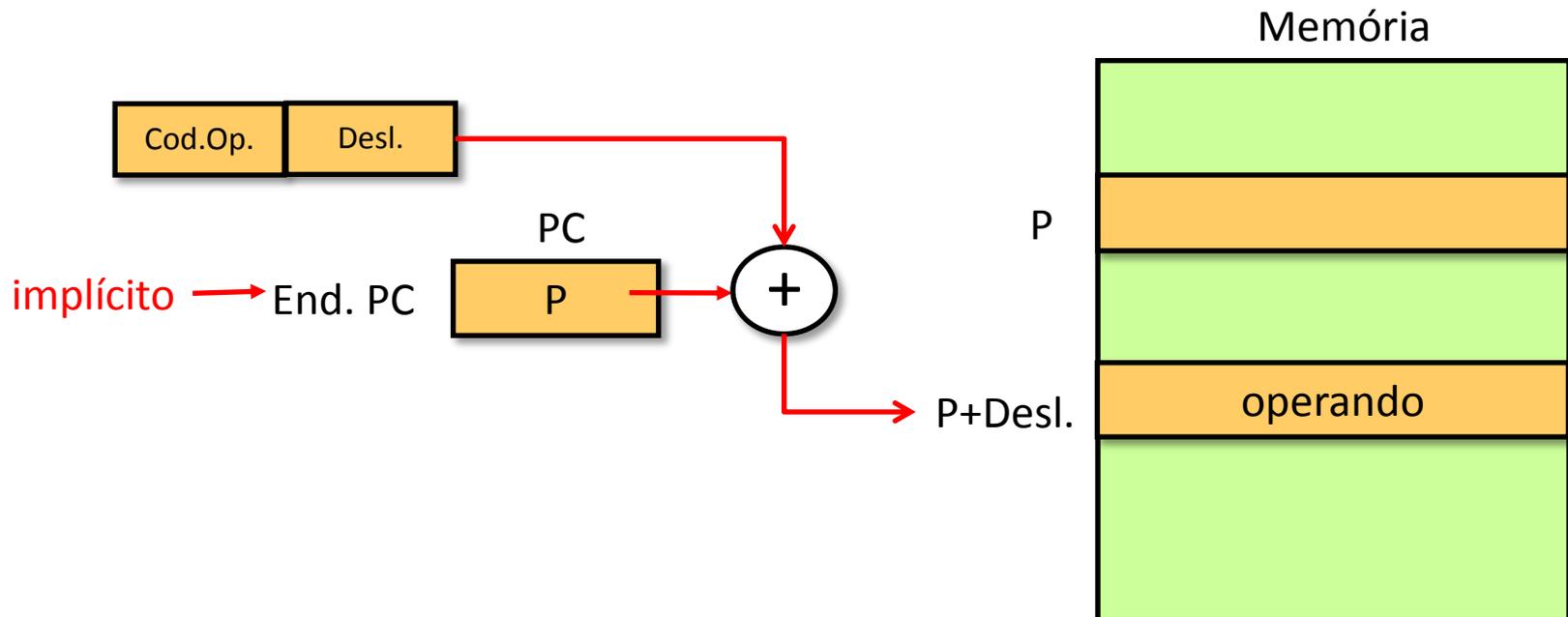
- Principal vantagem:
 - Flexibilidade
- Principal desvantagem:
 - Complexidade

Endereçamento por Deslocamento

- Tipos comuns:
 - Endereçamento Relativo
 - Endereçamento via Registrador Base
 - Indexação

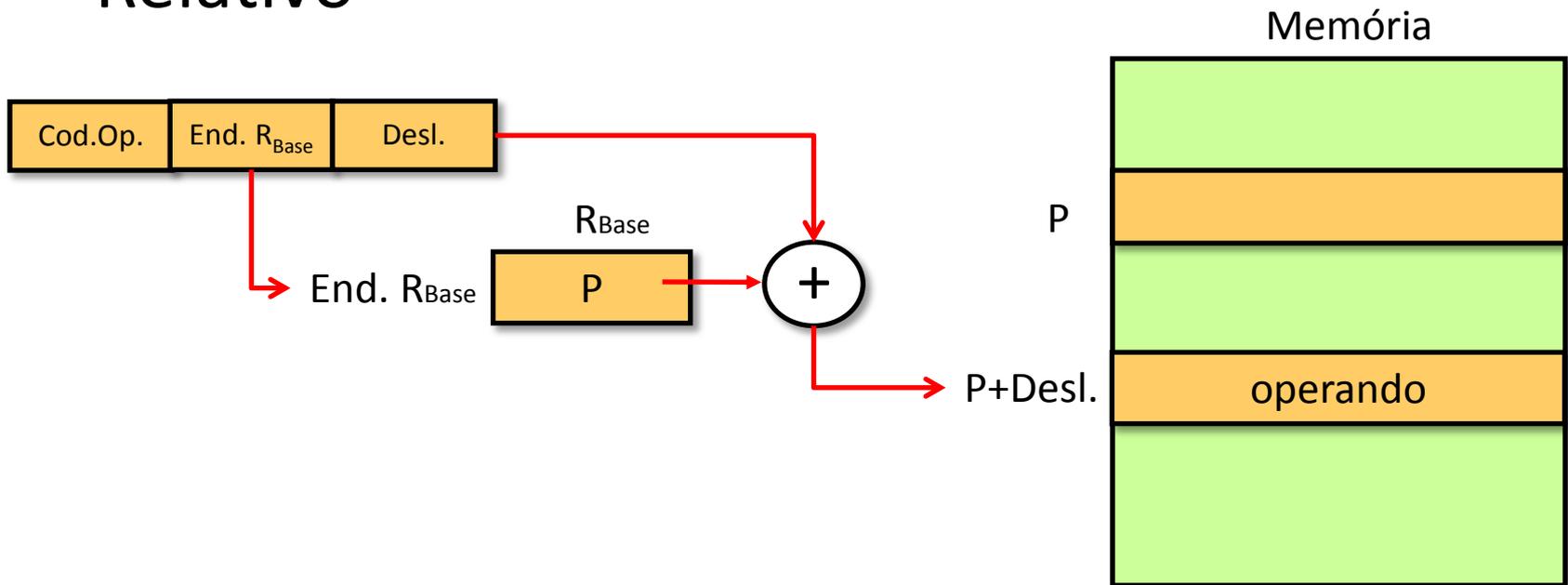
Endereçamento Relativo

- Explora o conceito de localidade, economizando bits no campo de endereço da instrução



Endereçamento via Registrador Base

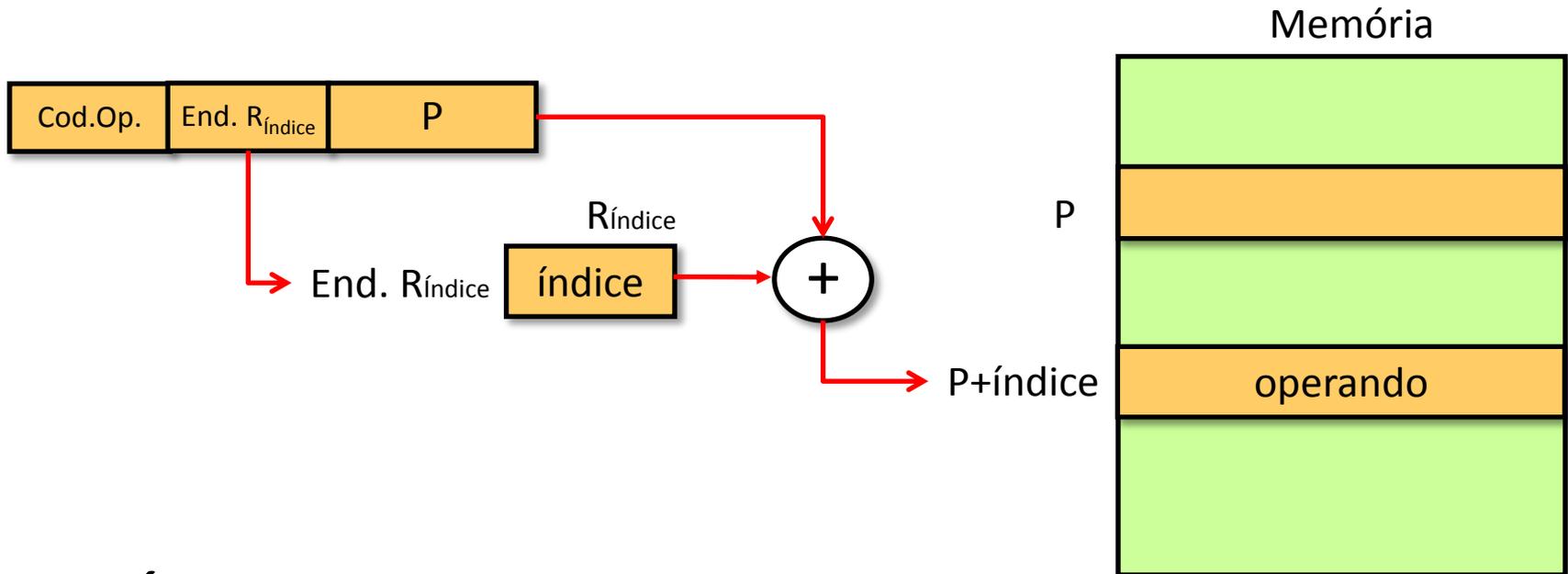
- É uma generalização do Endereçamento Relativo



- Útil na implementação de segmentação de memória

Indexação

- Interpretação oposta ao do Endereçamento via Registrador Base



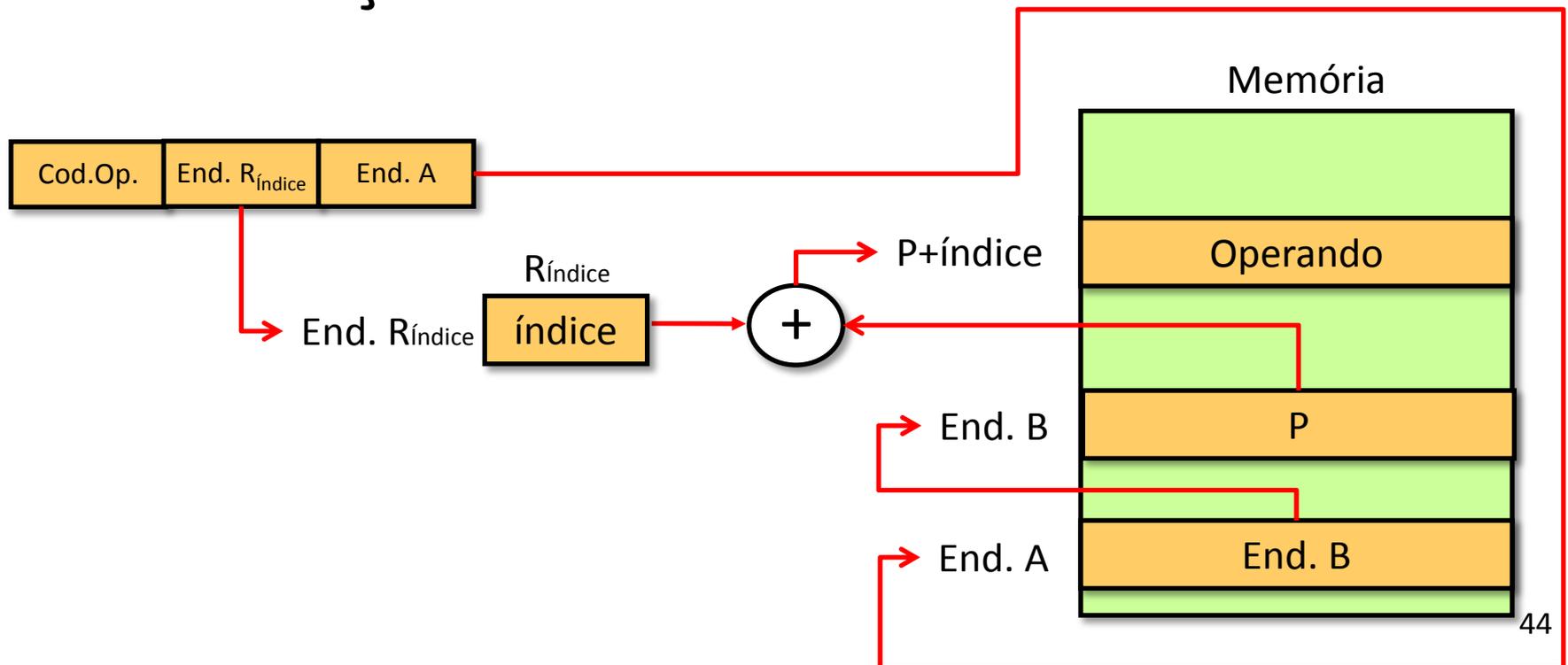
- Útil em operações iterativas

Indexação

- Auto-indexação: incremento do registrador índice no mesmo ciclo de instrução
 - Realizada automaticamente em registradores dedicados exclusivamente à indexação
 - Quando registradores de propósito geral são utilizados, é necessária indicação na instrução de que deva ser realizada
- Algumas máquinas permitem Endereçamento Indireto e Indexação na mesma instrução

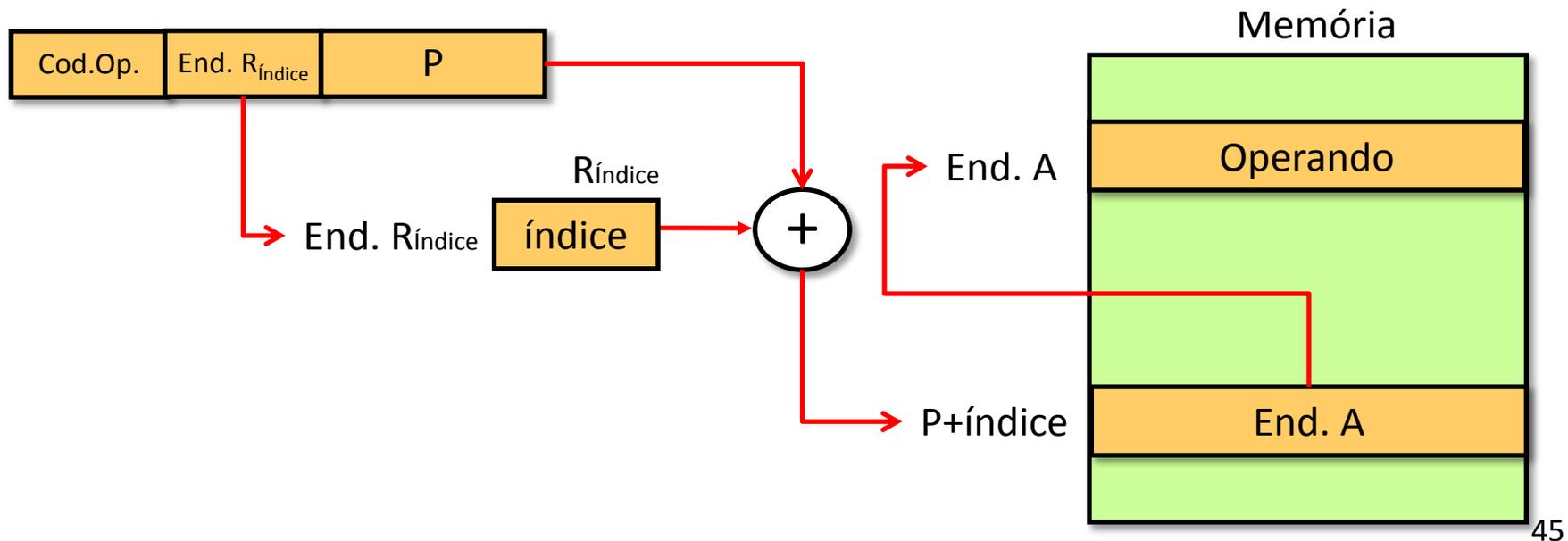
Indexação

- Quando a indexação é feita após o endereçamento indireto, ela é chamada PÓS INDEXAÇÃO



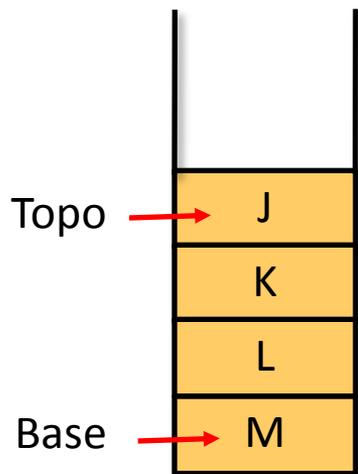
Indexação

- Na PRÉ INDEXAÇÃO, a indexação é feita antes do endereçamento indireto

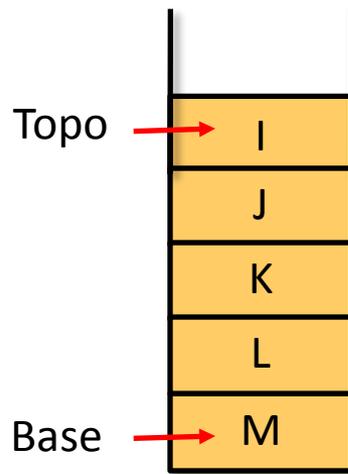


Endereçamento à Pilha

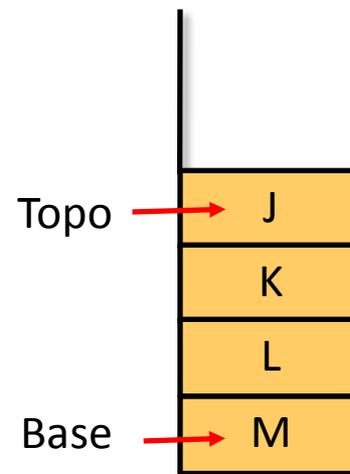
- Pilha: Fila LIFO (*Last In First Out*)
- Operações: PUSH, POP, operações unárias, operações binárias



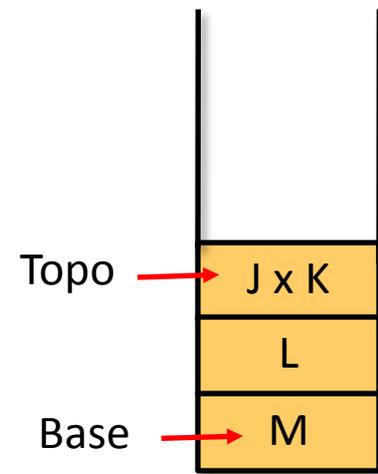
(a)
Início



(b)
Depois do PUSH



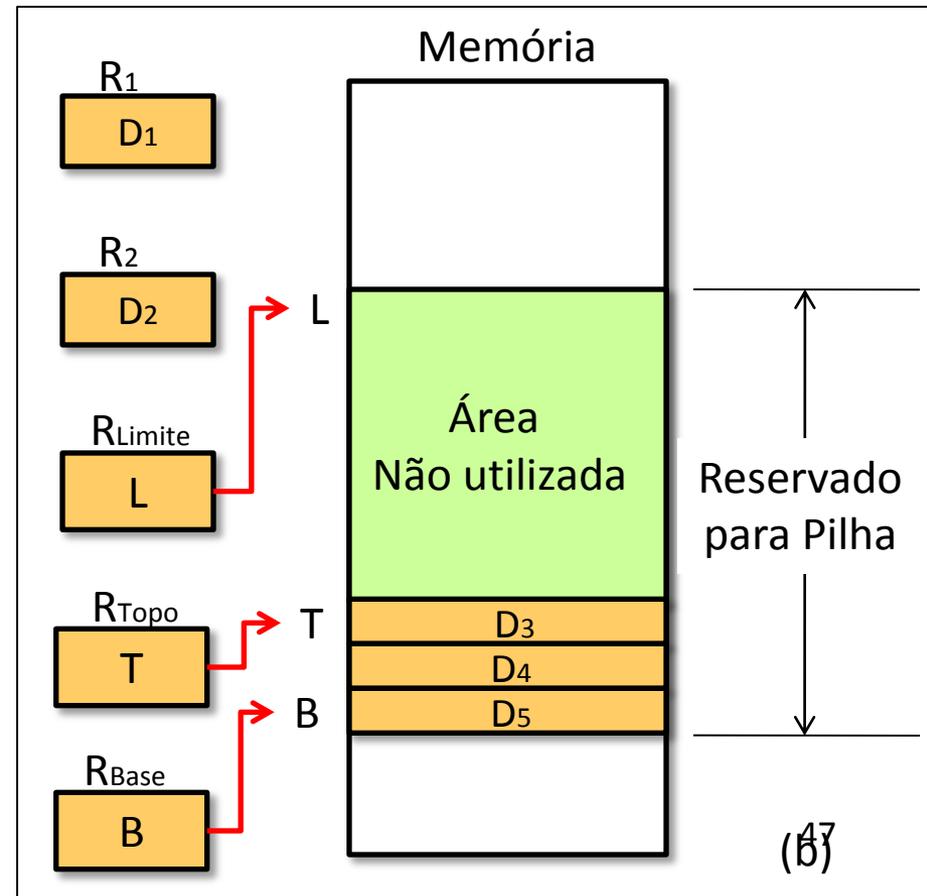
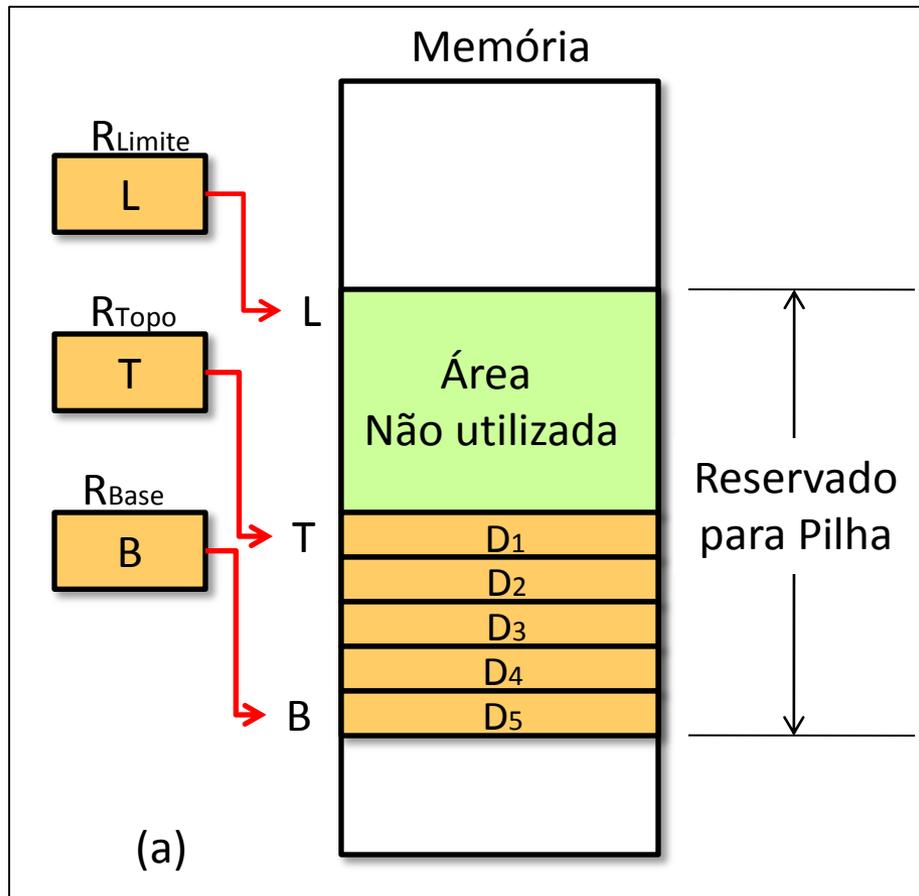
(c)
Depois do POP



(d)
Depois da \times
Multiplicação

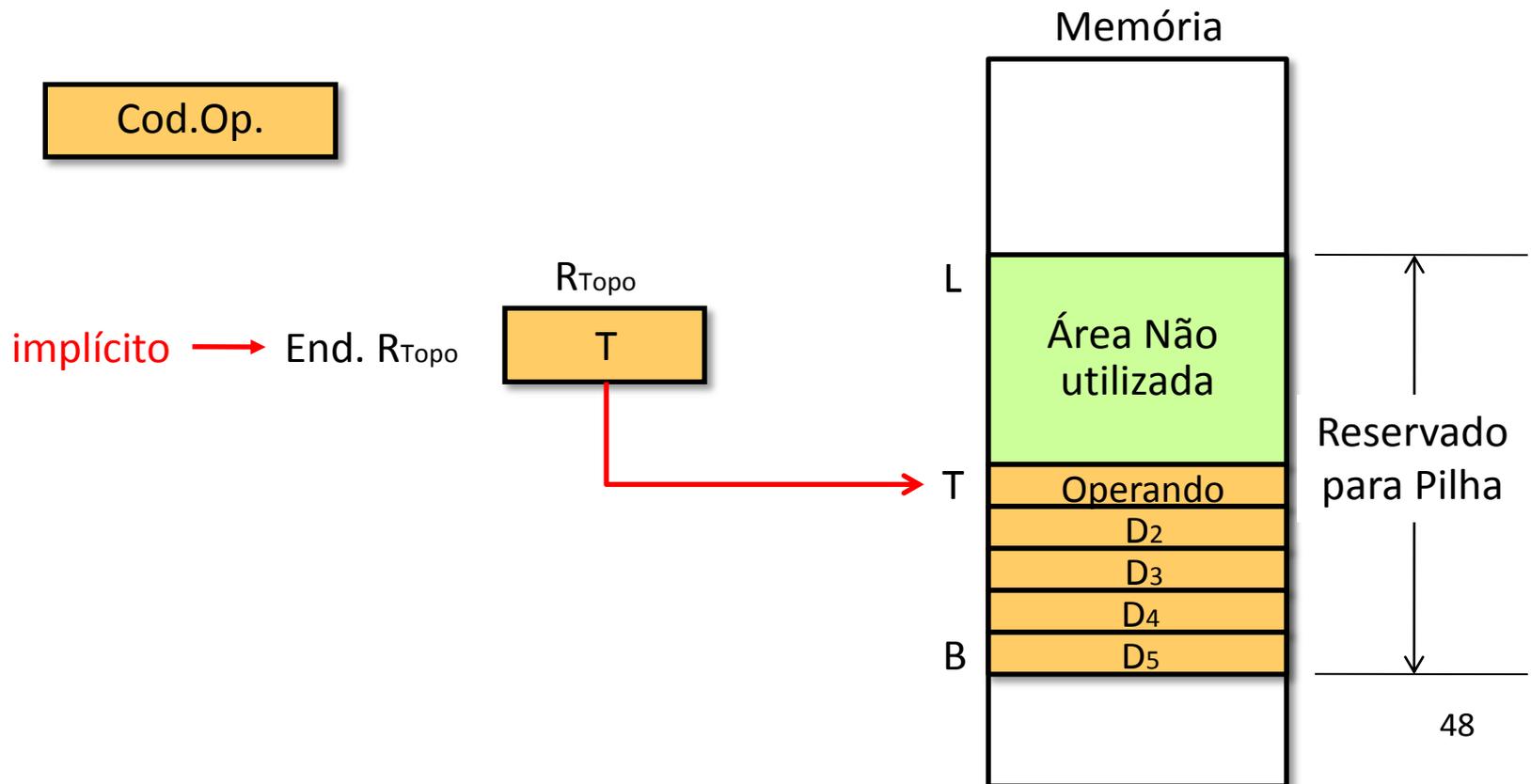
Endereçamento à Pilha

- Implementações típicas da Pilha:



Endereçamento à Pilha

- Trata-se de uma especialização do Endereçamento Indireto via Registrador



Formatos de instruções

- Layout de bits em uma instrução.
- Inclui opcode.
- Inclui operando(s) (implícitos ou explícitos).
- Normalmente, mais de um formato de instrução em um conjunto de instruções.

Formatos de instruções

- Tamanho da instrução
 - Afetado por e afeta:
 - Tamanho da memória.
 - Organização da memória.
 - Estrutura de barramento.
 - Complexidade da CPU.
 - Velocidade da CPU.
 - Escolha entre repertório de instrução poderoso e economia de espaço.

Formatos de instruções

- Alocação de bits
 - Número de modos de endereçamento.
 - Número de operandos.
 - Registrador versus memória.
 - Número de conjuntos de registradores.
 - Intervalo de endereços.
 - Granularidade do endereço.

Referências

- STALLINGS, W. Arquitetura e organização de computadores: projeto para o desempenho. 8. ed. Prentice Hall, 2009.
- DELGADO, J.; RIBEIRO, C. Arquitetura de Computadores. 2 ed. LTC, 2009.
- PATTERSON, D. A. ; HENNESSY, J.L. Organização e projeto de computadores – a interface hardware software. 3. ed. Editora Campus, 2005.