Conjuntos de Instruções - Características e Funções

Arquitetura e Organização de Computadores

Prof. Lucas de Oliveira Teixeira

UEM

Introdução

Introdução

- A operação de uma CPU é determinada pelas instruções (comandos) que ela executa, denominadas instruções de máquina ou instruções do computador.
- A coleção dessas diferentes instruções é chamada conjunto de instruções.
- Exemplos: instruções MMX (processamento de vídeo), SSE (processamento de ponto flutuante), entre outros.

Características das Instruções de Máquina

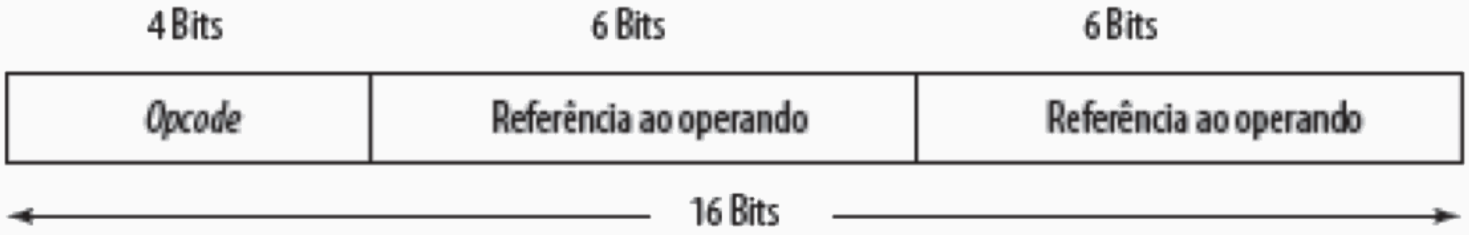
Características das Instruções de Máquina

Elementos:

- Código da operação (opcode).
- Referência ao operando fonte.
- Referência ao operando destino.
- Referência a próxima instrução (incomum).

Características das Instruções de Máquina

Elementos:



Características das Instruções de Máquina

Representação da instrução:

- Em código de máquina, cada instrução é representada por um padrão de bits exclusivo.
- Para melhor entendimento dos programadores utiliza-se uma representação simbólica (Assembly). Exemplo: ADD, SUB, LOAD.
- Operandos também possuem padrões de bits exclusivos e podem ser representados de maneira simbólica. Exemplo: ADD eax, ebx.

Características das Instruções de Máquina

Tipos de instrução:

- Processamento de dados: operações aritméticas e lógicas.
- Armazenamento de dados: movimentação de dados entre registradores e memória principal.
- Movimentação de dados para E/S: instruções para realização de operações de E/S.
- Controle de fluxo: instruções de teste e desvio.

Características das Instruções de Máquina

Número de endereços:

- A quantidade de endereços em uma instrução se refere à quantidade de operandos e destino que podemos usar.
- Existem processadores com três, dois, um e nenhum operando.
- Atualmente, as instruções com dois endereços são mais usadas.

Características das Instruções de Máquina

Instruções de três endereços:

- As instruções contém dois operandos e um destino.
- Por exemplo, a = b + c:

```
1 add a, b, c
```

- O problema é a necessidade de palavras muito longas para representar todos os operandos em uma instrução.

Características das Instruções de Máquina

Instruções de dois endereços:

- As instruções contém um operando e um destino.
- O destino é também um operando.
- Por exemplo, a = a + b:

```
1 add a, b
```

- Isso reduz o tamanho da palavra.
- No entanto, exige um trabalho extra do programador para manter o valor do operando original (no exemplo, a é sempre alterado).

Características das Instruções de Máquina

Instruções de um endereço:

- As instruções contém apenas um operando.
- Nesse caso, o segundo operando e o destino são implicítos e normalmente se referem ao acumulador (AC).
- Por exemplo, a = a + b:

```
1 load a
2 add b
3 store a
```

- Isso reduz ainda mais o tamanho da instrução.
- No entanto, exige muito mais trabalho do programador para a criação do programa.
- Bastante comum nos primeiros computadores com palavras pequenas.

Características das Instruções de Máquina

Instruções de zero endereços:

- As instruções não possuem nenhum operando, todos os endereços são implícitos.
- Nesse caso, os operandos e o destino são todos implícitos e normalmente se referem a pilha.
- Por exemplo, a = a + b:

```
1  push a
2  push b
3  add
4  pop a
```

- Bastante incomum.
- A instrução contém apenas o código da operação (com exceção do push e pop).
- Usado apenas em arquitetura experimentais.

Características das Instruções de Máquina

Mais endereços:

- Instruções mais complexas.
- Necessidade de mais registradores.
- Menos instruções por programa.

Menos endereços:

- Mais instruções por programa.
- Instruções menos complexas.
- Busca/execução de instruções mais rápida.

Características das Instruções de Máquina

Projeto do conjunto de instruções:

- Repertório de operações: quantas e quais operações?
- Tipos de dados: quais os tipos de dados aceitos?
- Formatos de instrução: qual o tamanho da instrução, quantidade de endereços, ...?
- Registradores: qual o número e o propósito dos registradores?
- Modos de endereçamento: quais as possíveis maneiras de referenciar um operando?
- Paradigma arquitetural: RISC ou CISC?

Tipos de Operandos

Tipos de Operandos

- Endereços.
- Números: inteiros ou de ponto flutuante.
- Caracteres: símbolos representados como números (ASCII).
- Dados lógicos: bits.

Tipos de Operações

Tipos de Operações

- Transferência de dados.
- Artimética.
- Lógica.
- Desvios.
- E/S.
- Controle do sistema.
- Transferência de controle.

Tipos de Operações

Transferência de dados:

- Necessário especificar origem, destino e quantidade de dados.
- Podem ser instruções diferentes para diferentes movimentações (entre registradores e memória): IAS, IBM 370.
- Pode ser uma instrução com diferentes endereços: VAX, Intel x86.

Tipos de Operações

Aritmética:

- Etapas:
  - Transferência de dados antes e/ou depois da operação.
  - Execução da operação na ULA.
  - Atualização dos códigos de condições (flags).
- Operações básicas: Soma, Subtração, Multiplicação, Divisão.
- Outras operações possíveis: incremento, decremento, negação e valor absoluto.

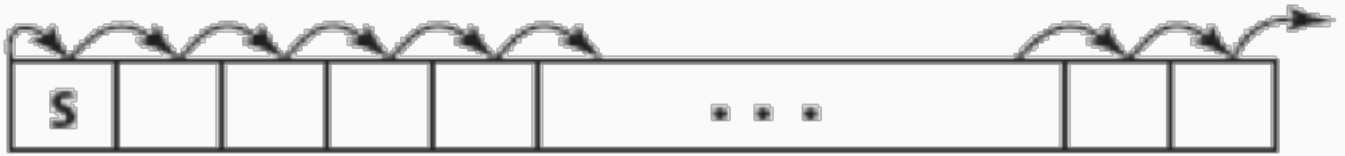
Tipos de Operações

Lógica:

- Operações lógicas básicas sobre dados booleanos ou binários: NOT, AND, OR, XOR.
- A comparação (CMP) também é uma operação lógica.

(a) Deslocamento lógico à direita

(b) Deslocamento lógico à esquerda



(c) Deslocamento aritmético à direita

Tipos de Operações

Deslocamentos:

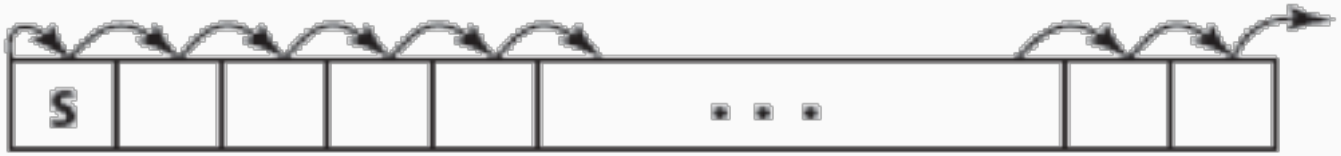
- As operações de deslocamento movem todos os bits de uma palavra são deslocados uma casa para direita ou esquerda, o bit da extremidade é perdido.
- Existem dois tipos: deslocamento lógico e deslocamento aritmético.
- O deslocamento lógico simplesmente desloca os bits para uma direção, o bit da extremidade é perdido e um 0 é colocado na outra extremidade.
- O deslocamento aritmético faz a mesma coisa, mas mantém o sinal no bit mais significativo.

Tipos de Operações

Deslocamentos:

(a) Deslocamento lógico à direita

(b) Deslocamento lógico à esquerda

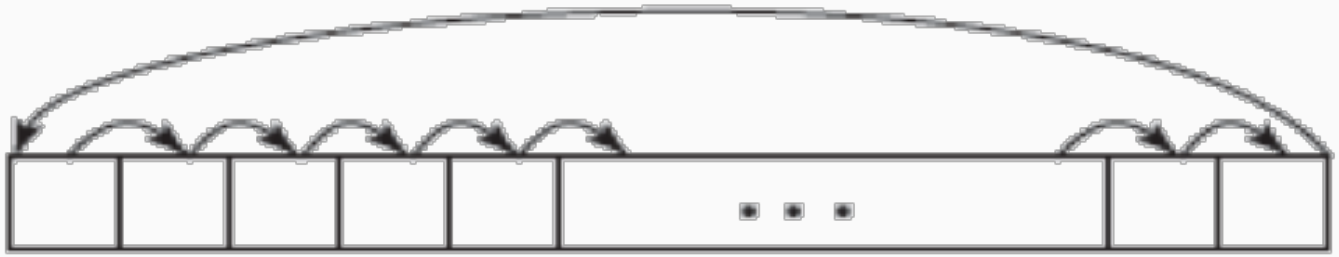


(c) Deslocamento aritmético à direita

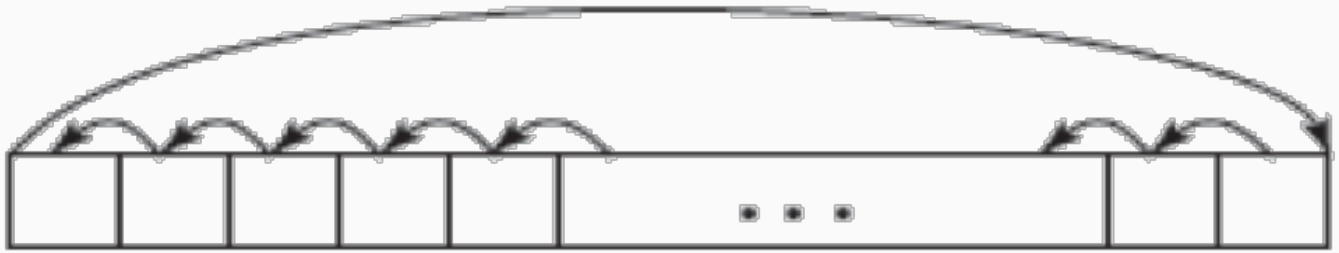
Tipos de Operações

Deslocamentos:

(d) Deslocamento aritmético à esquerda



(e) Rotação à direita



(f) Rotação à esquerda

Tipos de Operações

Entrada e saída:

- Podem ser feitas com instruções específicas ou usando instruções de movimentação de dados em conjunto com interrupções.
- Exemplo: Intel x86 – Operação de saída (impressão na tela).

```
1  MOV $4, %eax          ; Interrupção de escrita (sys_write)
2  MOV $1, %ebx          ; Indica modo de escrita (console)
3  MOV hello, %ecx       ; Indica o endereço da string
4  MOV helloLenght, %edx ; Indica o tamanho da string
5  INT 80h               ; Chamada da interrupção
```

Tipos de Operações

Controle de sistema:

- Instruções privilegiadas.
- CPU precisa estar em estado específico: Modo kernel.
- Em geral são apenas de uso dos sistemas operacionais.
- Exemplos:
  - Leitura ou alteração de um registrador de controle (PC, IR, MAR, MBR, etc.).
  - Leitura ou alteração de uma chave de proteção da memória.
  - Acesso a blocos de memória usados para controle de processos.

Tipos de Operações

Transferência de controle:

- Existem duas situações que ocorrem a transferência de controle: desvios e procedimentos.

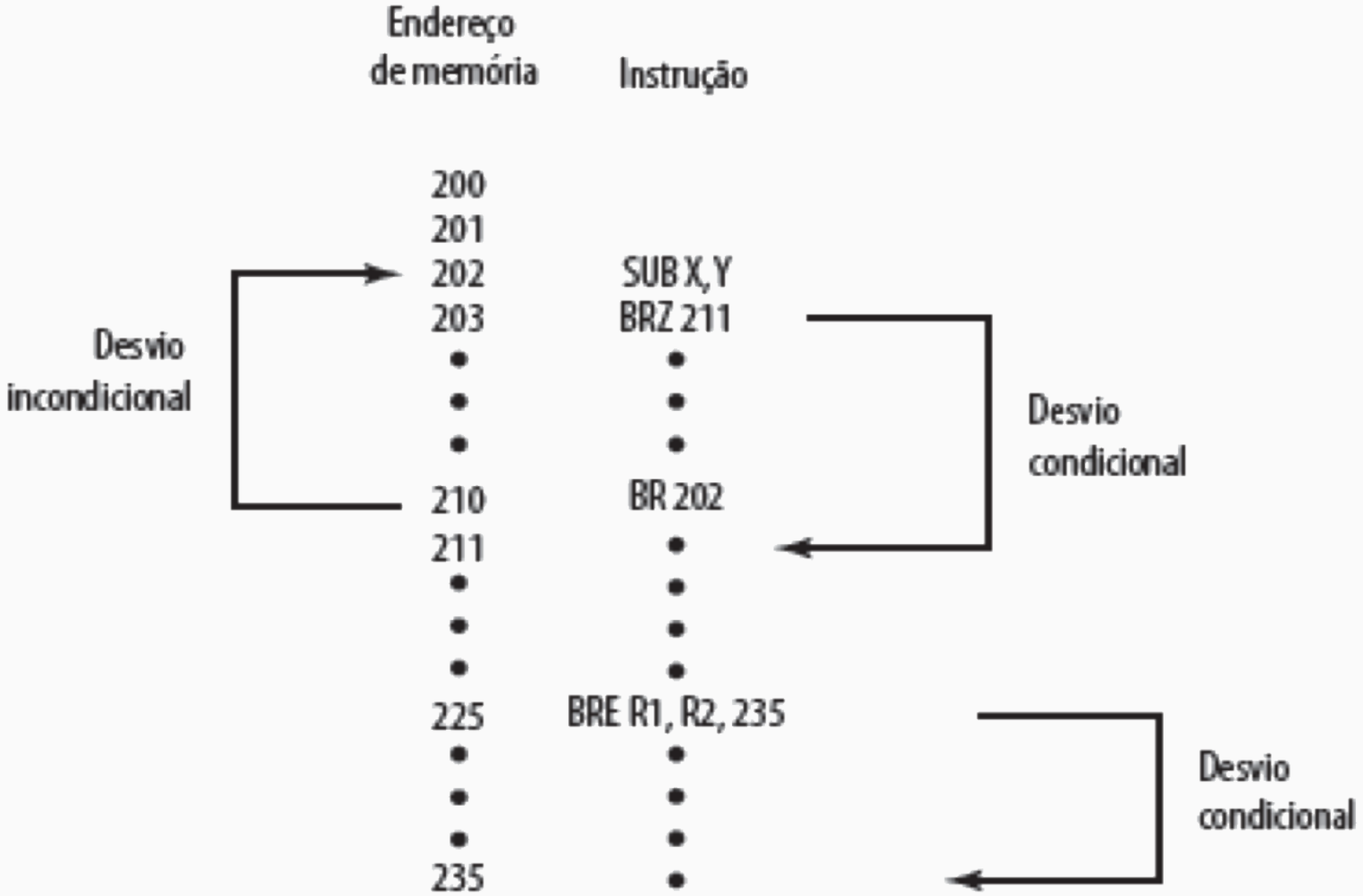
Tipos de Operações

Desvios:

- O desvio tem como um de seus operandos o endereço da próxima instrução a ser executada.
- Desvio Condicional: Só é tomado se uma condição for atendida.
- Desvio Incondicional: É sempre tomado.

Tipos de Operações

Desvios:



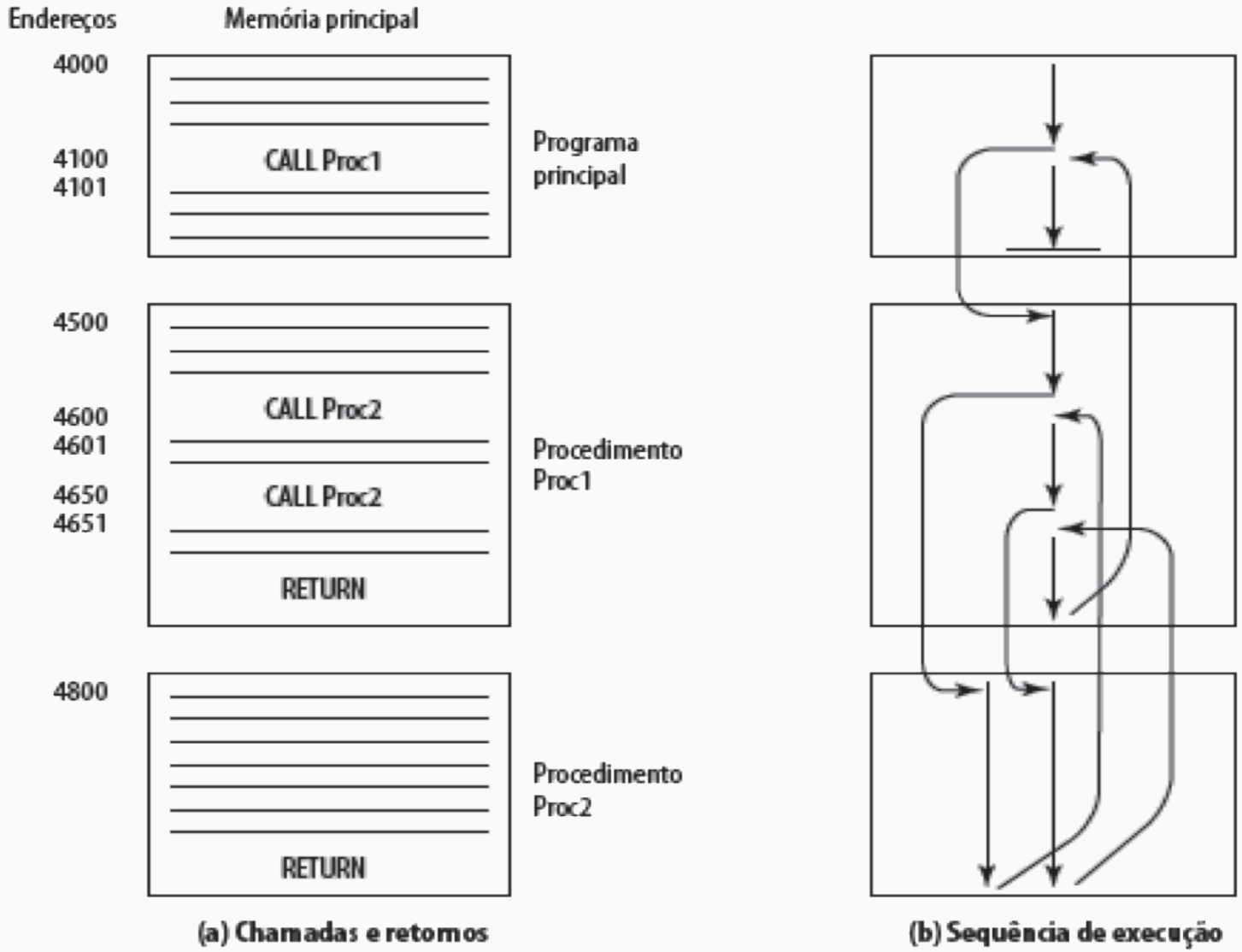
Tipos de Operações

Procedimento:

- Em uma chamada de procedimento, o processador é instruído a ir executar o procedimento inteiro e depois retornar ao ponto onde ocorreu a chamada.
- Inovação mais importante no desenvolvimento de linguagens de programação.
- Dois principais motivos: elimina repetição do código e faz divisão de grandes tarefas em unidades menores.
- Envolve duas instruções essenciais:
  - Instrução de chamada (CALL): Desvia a execução da instrução corrente para o início da subrotina.
  - Instrução de retorno (RET): Retorna a execução para o endereço que ocorreu a chamada.

Tipos de Operações

Procedimento:



Modos de Endereçamento

Modos de Endereçamento

- O modo de endereçamento diz respeito as diferentes maneiras que o programador pode se referir à um operando.

Modos de Endereçamento

Endereçamento imediato:

- O campo de endereço contém o próprio valor do operando, ele é parte da instrução.
- Exemplo: ADD 5
- O operando é o valor 5, com isso não existe nenhuma referência à memória para buscar dados.
- Vantagem: Rápido.
- Desvantagem: Tamanho do operando é limitado pelo tamanho do campo de endereço.

Modos de Endereçamento

Endereçamento imediato:

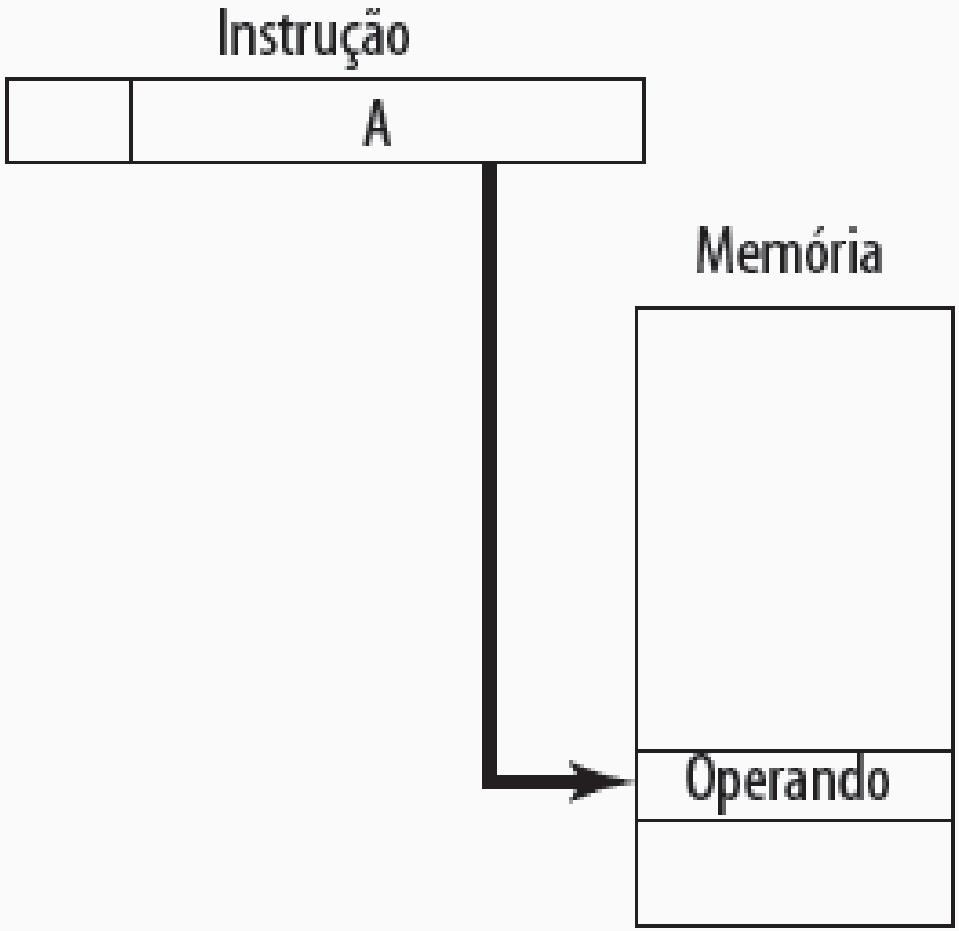
Modos de Endereçamento

Endereçamento direto:

- Campo de endereço contém endereço do operando.
- Exemplo: ADD A
- Normalmente, A é uma variável dentro da .section data.
- O operando é o conteúdo de A, com isso uma busca na memória é realizada no endereço A.
- Vantagem: Requer apenas um acesso à memória para acessar o dado.
- Desvantagem: Espaço de endereçamento limitado pelo tamanho do campo de endereço.

Modos de Endereçamento

Endereçamento direto:



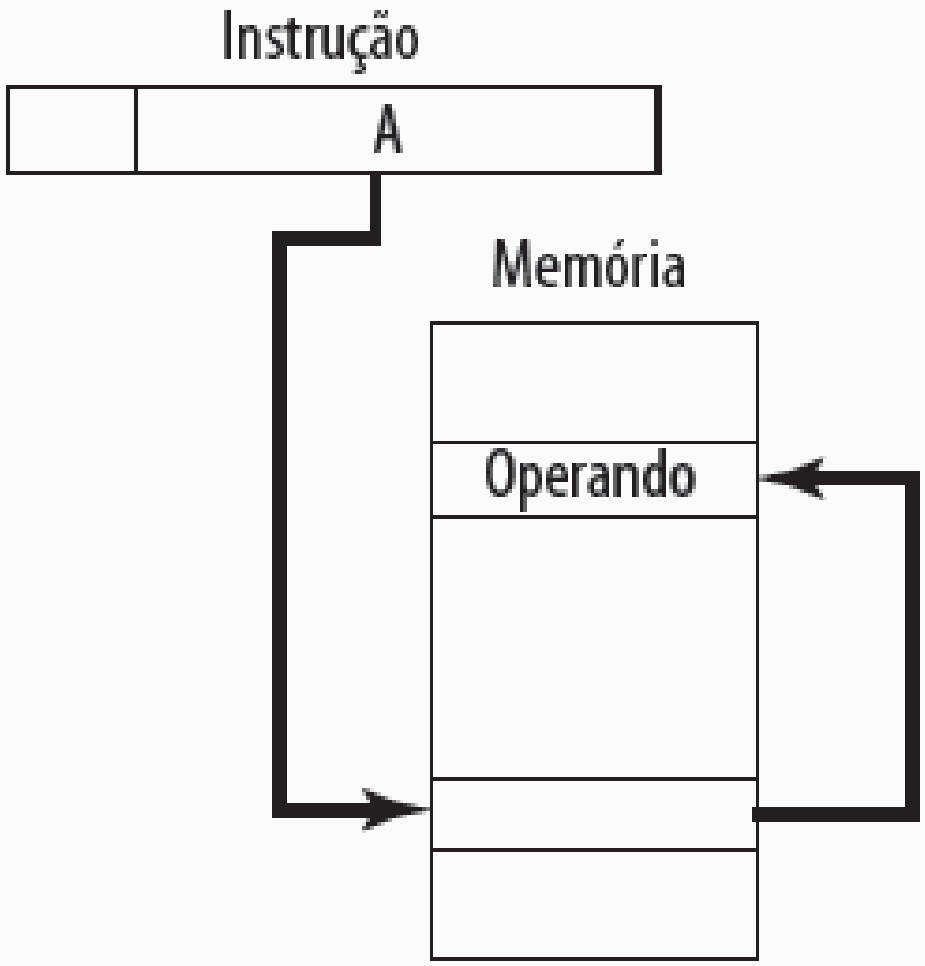
Modos de Endereçamento

Endereçamento indireto:

- Campo de endereço contém o endereço de memória que contém o endereço do operando, ou seja, contém um ponteiro para o operando.
- Exemplo: ADD [A]
- Vantagem: Fornece um grande espaço de endereços.
- Desvantagem: Pelo menos dois acessoa à memória são necessários.

Modos de Endereçamento

Endereçamento indireto:



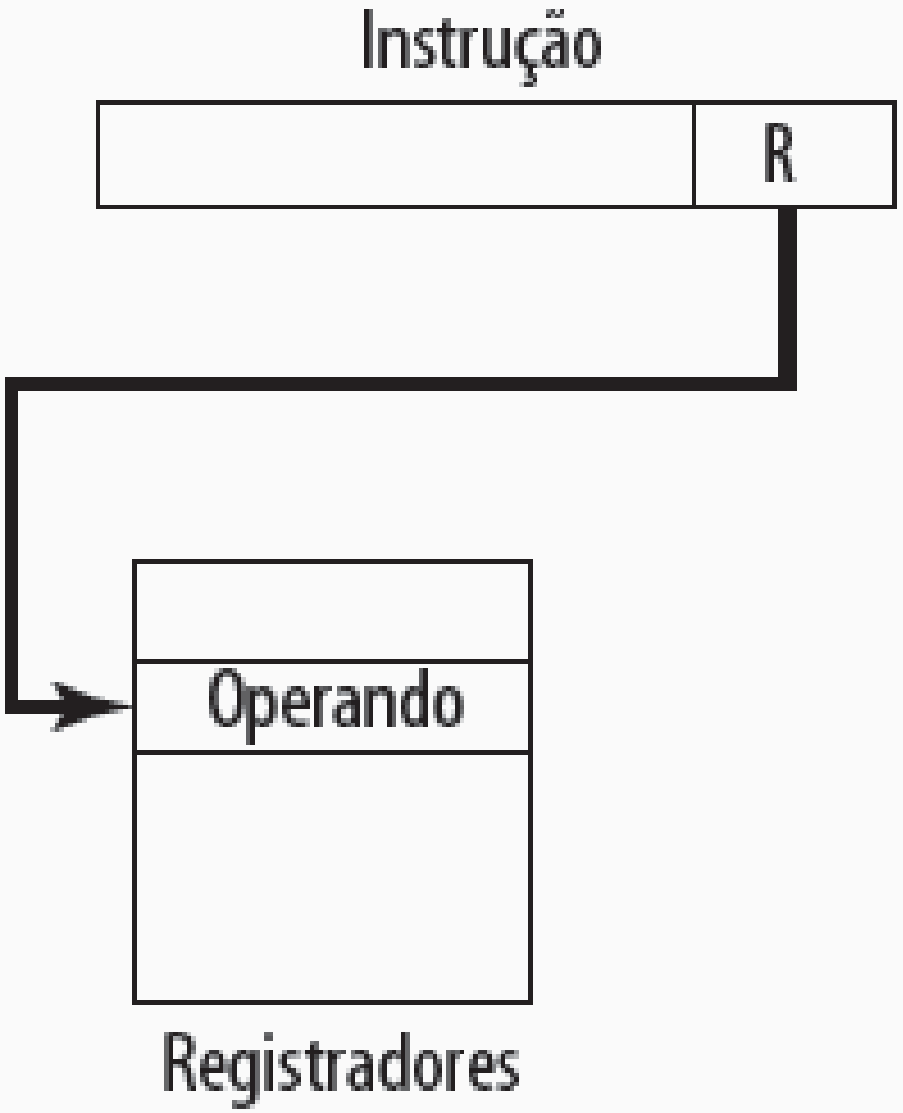
Modos de Endereçamento

Endereçamento direto por registradores:

- Campo de endereço contém a referência do registrador que contém o operando.
- Exemplo: ADD RAX
- Vantagem: Tamanho do campo de endereço necessário na instrução pequeno, assim as instruções são mais curtas; e, nenhum acesso à memória é necessário.
- Desvantagem: Número limitado de registradores.

Modos de Endereçamento

Endereçamento direto por registradores:



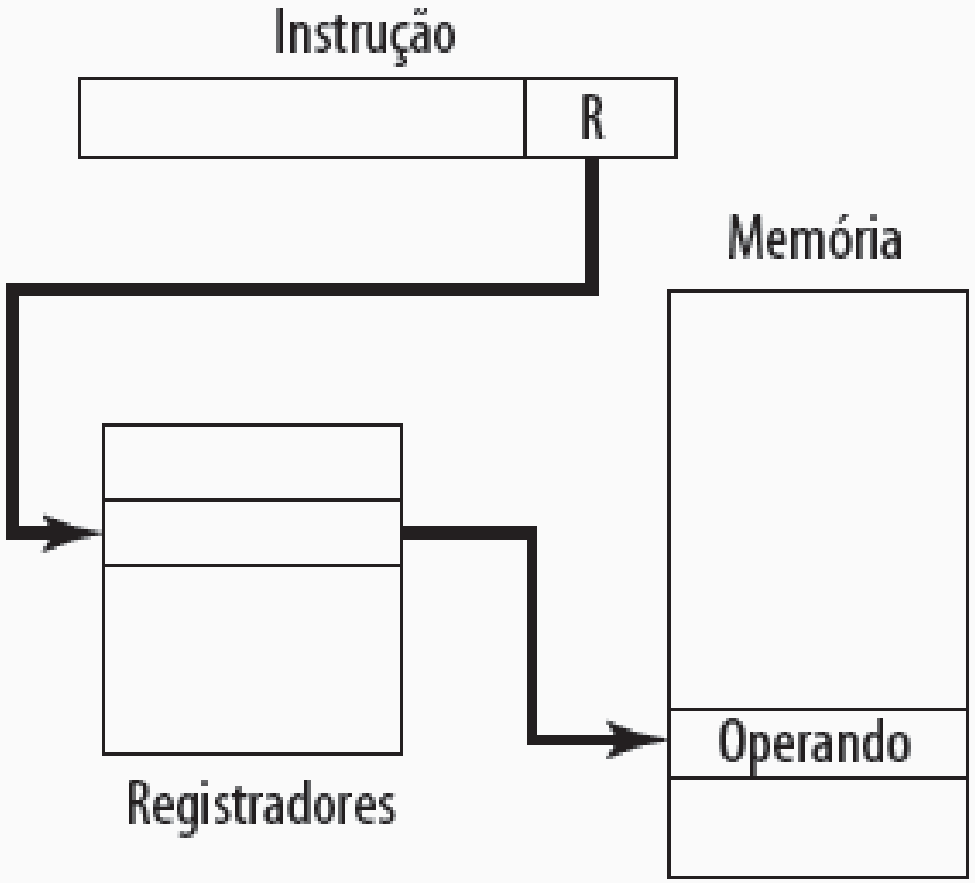
Modos de Endereçamento

Endereçamento indireto por registradores:

- Campo de endereço contém a referência do registrador que contém o endereço do operando.
- Exemplo: ADD [RAX]
- Vantagem: Fornece um grande espaço de endereços, e ao mesmo tempo necessita de apenas um acesso à memória.
- Desvantagem: Espaço de endereçamento limitado ao tamanho do registrador (o que não é uma grande desvantagem).

Modos de Endereçamento

Endereçamento indireto por registradores:



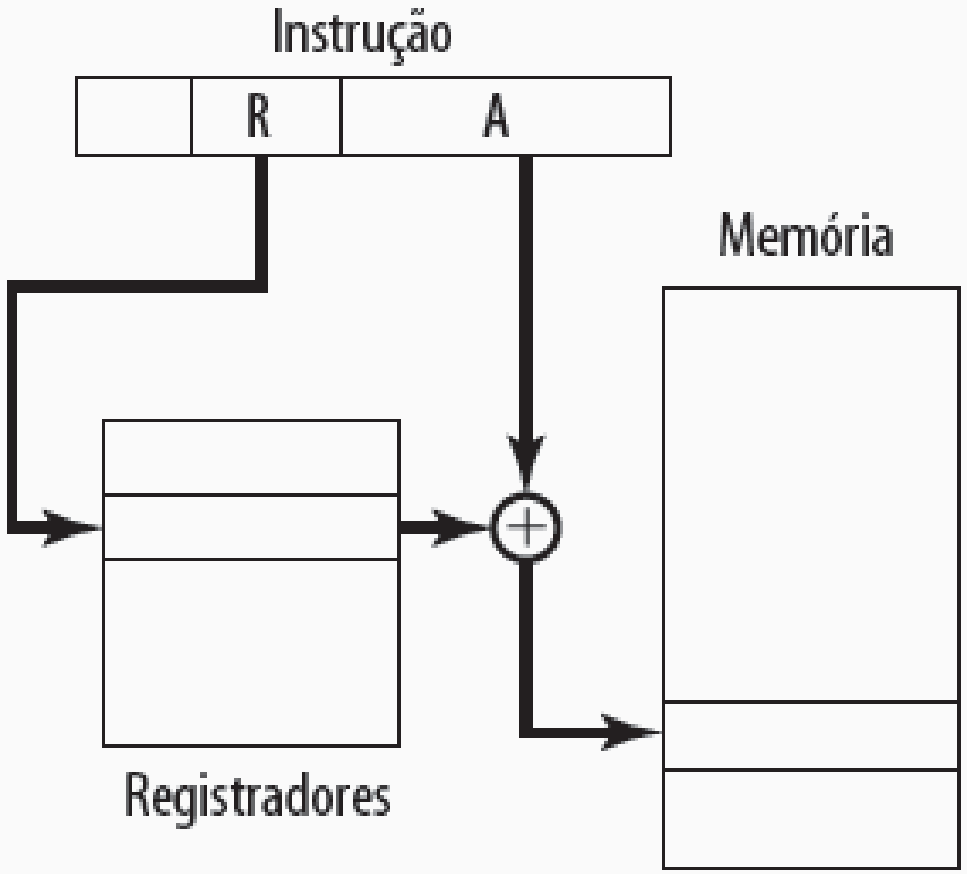
Modos de Endereçamento

Endereçamento por deslocamento:

- Campo de endereço contém a referência para um endereço base mais um valor de deslocamento.
- O valor de deslocamento é somado ou subtraído do endereço base e o endereço final do operando é obtido.
- Exemplo: ADD [RBP + 8]

Modos de Endereçamento

Endereçamento por deslocamento:



Modos de Endereçamento

Endereçamento por deslocamento relativo:

- O endereço base é implícito como o endereço da instrução atual.
- Assim o campo de endereço apenas o valor de deslocamento.
- Explora conceito de localidade espacial, se a maioria das referências à memória são próximas, é possível economizar bits no campo de endereço da instrução.

Modos de Endereçamento

Endereçamento na pilha:

- Os operandos ficam implicitamente no topo da pilha.
- As operações retiram os operandos do topo, realizam a operação e empilham o resultado.

```
1 push a
2 push b
3 add
4 pop a
```

Formatos de Instrução

Formatos de Instrução

- Vimos que uma instrução deve conter pelo menos o código da operação e os operandos (sejam explícitos ou implícitos).
- Normalmente existe mais de um formato de instrução em um conjunto de instruções.
- Assim, existem instruções em que o tamanho dos campos são diferentes e até mesmo instruções de tamanho diferentes.

Formatos de Instrução

O número de bits em uma instrução influencia em:

- Tamanho da memória.
- Organização da memória.
- Estrutura de barramento.
- Complexidade da CPU.
- Velocidade da CPU (instruções por segundo).

Formatos de Instrução

O número de bits em uma instrução influencia em:

- Quantidade de modos de endereçamento.
- Número de operandos.
- Operando em registrador versus operando na memória:
  - Quanto menos registradores, menos bits necessários para identificar cada um.
  - Exemplo: Se temos apenas um registrador de uso geral o operando é implícito, então não precisamos de nenhum bit para identifica-lo.

Formatos de Instrução

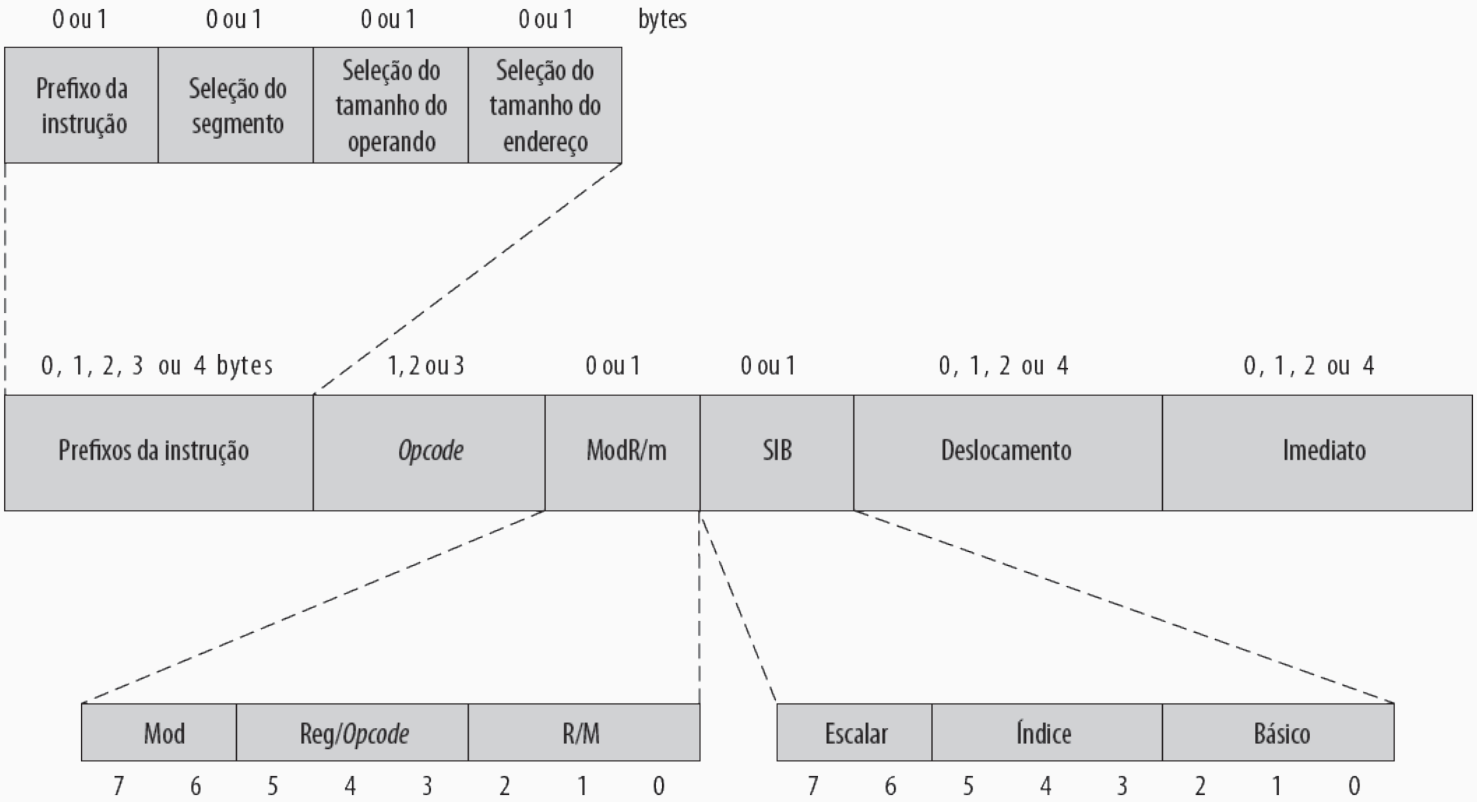
O número de bits em uma instrução influencia em:

- Número de conjuntos de registradores.
  - Dividir registradores em conjuntos especializados, no qual o opcode determina implicitamente o banco usado.

O objetivo é encontrar um equilíbrio entre um repertório de instruções poderoso e economia de espaço.

Formatos de Instrução

Formato de instrução x86:



Formatos de Instrução

Prefixos da instrução:

- Prefixo da instrução: LOCK (bloqueia memória compartilhada) ou REP, REPE, REPZ, REPNE e REPNZ (repete instrução).
- Seleção de segmento: Determina qual registrador de segmento uma instrução deve utilizar.
- Tamanho do operando: 16, 32 ou 64 bits.
- Tamanho do endereço: 16, 32 ou 64 bits.
- Opcode: Pode informar também a direção da operação (de/para memória).

Formatos de Instrução

ModR/m, SIB e Deslocamento:

- Especificam se um operando está em um registrador ou na memória.
- Se estiver na memória, então especificam também o modo de endereçamento a ser usado.

Formatos de Instrução

Imediato:

- Fornece o valor de um operando de 8, 16, 32 ou 64 bits.