

ARQUITETURA DE COMPUTADORES

Conjunto de Instruções 03
Propriedades de uma ISA

Tipos de instrução

- ☐ Processamento de dados
- ☐ Armazenamento de dados (memória principal)
- ☐ Movimentação de dados (I/O)
- ☐ Controle do fluxo do programa

Tipos de operação

- ☐ Transferência de dados
 - ☐ Aritméticas
 - ☐ Lógicas
 - ☐ Transferência de controle
 - ☐ Conversão
 - ☐ I/O
 - ☐ Controle do sistema
-

3

Transferência de dados

- ☐ Especificam
 - Origem
 - Destino
 - Quantidade de dados
-

4

Aritmética

- ☐ Soma, subtrai, multiplica, divide
- ☐ Inteiros com sinais
- ☐ Ponto flutuante
- ☐ Pode incluir
 - Incremento (a++)
 - Decremento (a--)
 - Negação (-a)

5

Lógica

- ☐ Operações com bits (set/reset)
- ☐ AND, OR, NOT

6

Transferência de Controle

- Salto/desvio (Jump/Branch)
 - Ex. desvia para x se resultado é zero
- Chamada Sub-rotina
 - Ex. call x

7

Conversão

- Ex. Binário para Decimal

8

Input/Output

- ☐ Podem ser instruções específicas
- ☐ Podem ser usadas instruções de movimento de dados (I/O mapeado em memória)

9

Controle do sistema

- ☐ Instruções privilegiadas
- ☐ CPU precisa estar em um estado específico
 - Kernel mode
- ☐ Para uso do sistema operacional

10

Arquiteturas de endereços

- ❑ Número de Endereços
- ❑ Modos de endereçamento

11

Número de Endereços (a)

- ❑ 4 endereços
 - Operando 1, Operando 2, Resultado, Próxima Instrução
 - ❑ INSTR E1 E2 E3 E4
 - Precisa de palavras de instrução longas
 - Registrador PC
 - Não existem instruções de desvio

12

Número de Endereços (b)

☐ 3 endereços

- Operando 1, Operando 2, Resultado
 - ☐ INSTR E1 E2 E3
- Ex.: $E3 = E1 + E2$;
- Precisa de palavras de instrução longas
- Registrador PC
- Instruções explícitas de desvio

13

Número de Endereços (c)

☐ 2 endereços

- Um endereço define um operando e o resultado
 - ☐ INSTR E1 E2
- Ex.: $E1 = E1 + E2$;
- Reduz o tamanho da instrução
- Requer algum trabalho extra
 - ☐ Armazenamento temporário de algum resultado
 - ☐ Instrução de movimentação de dados

14

Número de Endereços (d)

- ☐ 1 endereço
 - Segundo endereço é implícito
 - ☐ INSTR E1
 - Usualmente possui um registrador específico (acumulador)
 - Instrução de movimento de dados indica o sentido (Load/Store)

15

Número de Endereços (e)

- ☐ 0 (zero) endereços
 - Todos endereços implícitos
 - Uso de pilha (stack)
 - ☐ Ex.: $c = a + b$
 - ☐ push a
 - ☐ push b
 - ☐ add
 - ☐ pop c
 - Não apresenta vantagens significativas sobre arquiteturas de 1 ou 2 endereços

16

Modos de endereçamento

- ☐ Imediato
 - ☐ Direto
 - ☐ Indireto
 - ☐ Registrador
 - ☐ Registrador Indireto
 - ☐ Deslocamento (Indexado)
 - ☐ Pilha
-

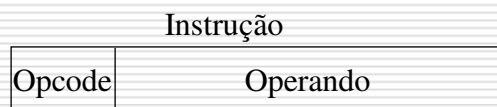
17

Endereçamento Imediato

- ☐ O operando é parte da instrução
 - ☐ Operando É o campo de endereço da instrução
 - ☐ Ex.: ADD 5
 - Soma 5 ao conteúdo do acumulador
 - 5 é o operando
 - ☐ Não há referência a memória para busca de dados
 - ☐ Rápido
 - ☐ Faixa (range) limitado
-

18

Diagrama para endereçamento imediato



Sem referência à memória
Magnitude do operando é limitada

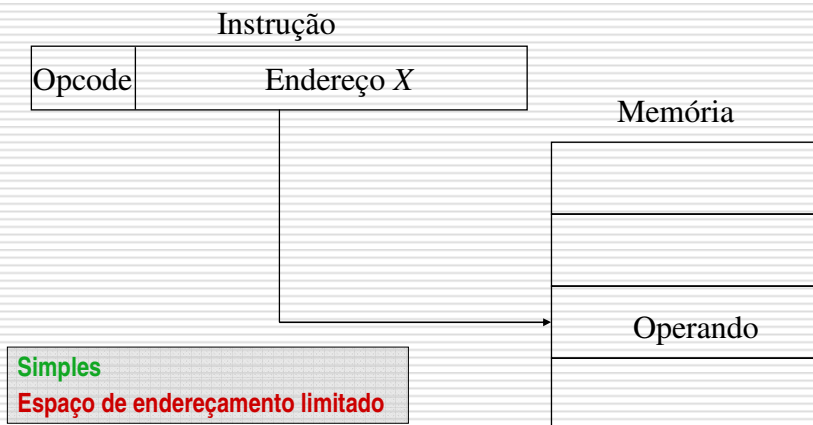
19

Endereçamento Direto

- ☐ Campo de endereço contém endereço do operando
- ☐ Endereço Efetivo (EA) = Campo de Endereço (X)
- ☐ Ex.: ADD X
 - Soma o conteúdo de X ao acumulador
 - Busca pelo operando na memória no endereço X
- ☐ Sem cálculos adicionais para obter o endereço efetivo
- ☐ Espaço de endereçamento limitado

20

Diagrama para endereçamento direto



21

Endereçamento Indireto (1)

- ❑ Posição de memória apontada pelo campo de endereço contém endereço do ponteiro do operando
- ❑ $EA = (X)$
 - Busca em X , encontra o endereço e busca lá o operando
- ❑ Ex.: ADD (X)
 - Soma o conteúdo da posição de memória apontada pelo conteúdo de X ao acumulador

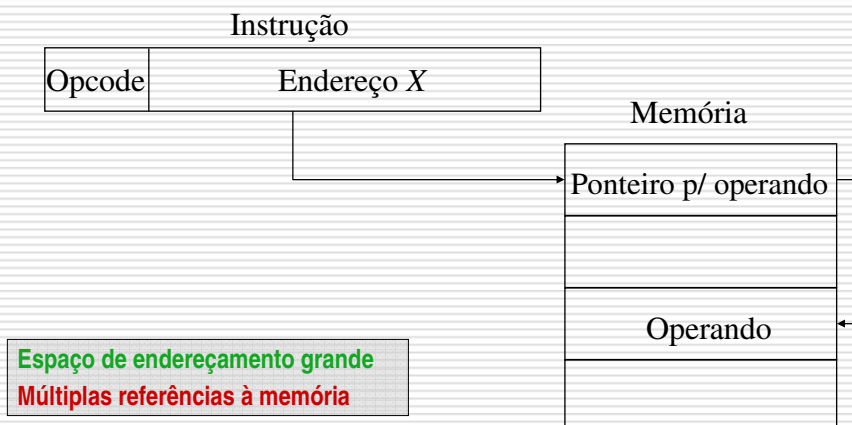
22

Endereçamento Indireto (2)

- ❑ Largo espaço de endereçamento
 - 2^n , onde n = tamanho da palavra
- ❑ Múltiplos acessos à memória para obter o operando
 - Logo, mais lento.

23

Diagrama para endereçamento indireto



24

Endereçamento por Registrador (1)

- ☐ Operando está em um registrador cujo nome vai no campo de endereço
- ☐ $EA = R$
- ☐ Numero de registradores é limitado
- ☐ Campo de endereço reduzido
 - Instruções curtas
 - Tempo de busca por instrução reduzido

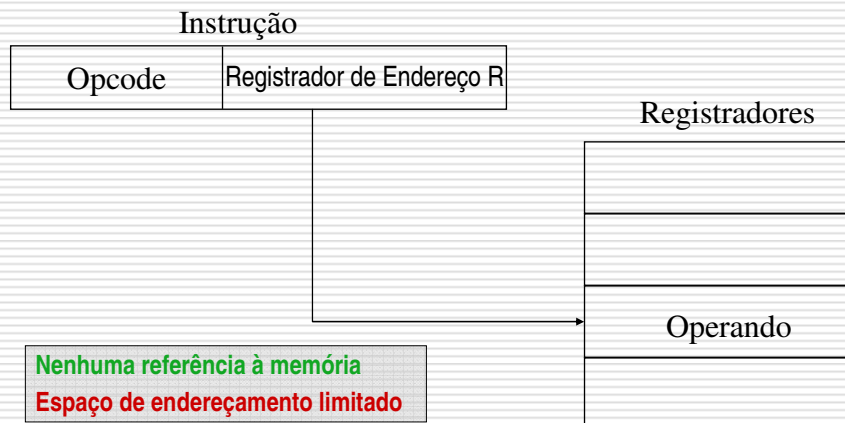
25

Endereçamento por Registrador (2)

- ☐ Sem acessos à memória
- ☐ Execução rápida
- ☐ Espaço de endereçamento limitado
- ☐ Múltiplos registradores melhora o desempenho
 - Requer um bom compilador (ou um bom programador Assembly)
 - Programação C: register int a;
- ☐ Semelhante ao endereçamento direto

26

Diagrama para endereçamento por registrador



27

Endereçamento Indireto por Registrador

- ☐ Similar ao endereçamento indireto
- ☐ $EA = (R)$
- ☐ Operando está na posição de memória apontada pelo registrador R
- ☐ Largo espaço de endereçamento (2^n)
- ☐ Um acesso de memória a menos que o endereçamento indireto

28

Diagrama para endereçamento indireto via registrador



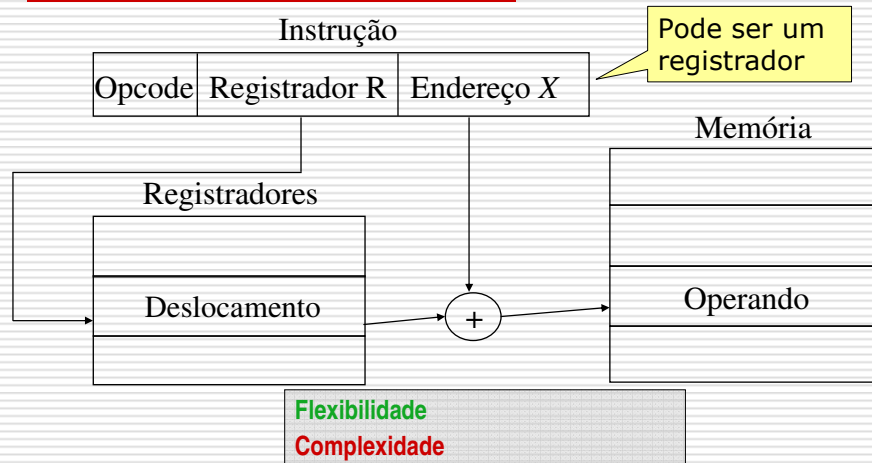
29

Endereçamento Indexado (por deslocamento)

- ☐ $EA = X + R$
- ☐ Campo de endereço contém dois valores
 - X = valor base
 - R = registrador q contém o deslocamento
- ☐ Bom para acessar arrays
 - $EA = X + R$
 - $R++$

30

Diagrama para endereçamento indexado (por deslocamento)



31

Endereçamento pilha

- ☐ Operando está (implicitamente) no topo da pilha
- ☐ Ex.: ADD
 - Recupera os dois valores no topo da pilha e soma

Nenhuma referência à memória
Aplicabilidade limitada

32

Resumo

- ❑ Tipos de instrução / operação
- ❑ Número de endereços
- ❑ Modos de endereçamento

33

Literatura

- ❑ Weber – cap 3.6
- ❑ *Stallings – cap 9 e 10*

34

Um erro aqui, outro lá... fazem
parte da descoberta de novos
caminhos.

Mas, para perder mesmo, você
deve ser perseverante nos
erros.

Anônimo

35

Questões adicionais...

- ☐ Identificar as propriedades apresentadas no conjunto de instruções dos processadores abaixo:
 - ☐ PowerPC (IBM)
 - ☐ ARM7TDMI
 - ☐ 8051 (INTEL)
 - ☐ 8086 - X86 (INTEL)
 - ☐ ColdFire (Motorola - Freescale)
 - ☐ SPARC (SUN)
 - ☐ TMS320C6000 (Texas Instruments)
 - ☐ Z-80 (Zilog)
 - ☐ NIOS (Altera)
 - ☐ PIC 18F (Microchip)
 - ☐ XScale (Intel)
 - ☐ OMAP (Texas Instruments)
 - ☐ MicroBlaze (Xilinx) - DLX

36