N_HARD COMPA4 - Texto de apoio

Site: EAD Mackenzie
Tema: HARDWARE PARA COMPUTAÇÃO {TURMA 01B} 2021/2
Livro: N_HARD COMPA4 - Texto de apoio

Impresso por: ANDRE SOUZA OCLECIANO .
Data: segunda, 27 set 2021, 18:42

Descrição

Índice

1. UCP – INSTRUÇÃO DE MÁQUINA, MODOS DE ENDEREÇAMENTO, PARALELISMO

COMPONENTE: Hardware para Computação

AULA: 4

PROFESSORA: Daniela Vieira Cunha

UCP – INSTRUÇÃO DE MÁQUINA, MODOS DE ENDEREÇAMENTO, PARALELISMO

Instrução de máquina

Uma instrução de máquina é a formalização de uma operação básica e simples que o hardware, especificamente a ULA, é capaz de realizar diretamente.

O projeto de um processador é centrado no conjunto de instruções de máquina que se deseja que ele execute, ou seja, no conjunto de operações primitivas que ele poderá entender e executar.

As instruções de máquina são aproximadamente 250, dependendo do projeto de processador, e devem incluir:

- operações matemáticas (aritméticas, lógicas, complemento);
- armazenamento de dados (entre memória e processador);
- movimentação de dados (leitura/escrita em dispositivo de E/S);
- controle de fluxo de dados (instruções de teste e desvio).

As instruções de máquina são compostas por (Figura 1):

- código de operação: especifica a operação a ser efetuada
- operando(s): indica(m) a localização do dado (ou dados) que será(ão) manipulado(s) durante a realização da operação. A instrução de máquina pode ter de 1 a, no máximo, 3 operandos.

Figura 1 – Formatos de instrução de máquina.

Fonte: elaborada pela autora.

Exemplos de instruções de máquina (Figura 1):

Formato com 3 campos de operandos: operando3 ß operando1 + operando2

ADD A, B, C -> A = B + C

Formato com 2 campos de operandos: operando1 ß operando1 + operando2

ADD A, B -> A = A + B

Formato com 1 campo de operando: operando1 ß operando1 + 1

INC A -> A = A + 1

Importante!

Um programa escrito em linguagem de alto nível, ao ser executado, foi primeiramente compilado ou interpretado. Quando o programa é compilado ou interpretado, cada linha de comando é traduzida em uma ou mais instruções de máquina que são interpretadas e executadas pelo processador (Figura 2).

Figura 2 – Tradução de um comando em linguagem de alto nível para uma linguagem de baixo nível utilizando tamanhos de instruções diferentes

Y = (A – B) / (C + D * E)

Instrução	Comentário
SUB Y, A, B	Y ← A – B
MPY T, D, E	T ← D * E
ADD T, T, C	T ← T + C
DIV Y, Y, T	Y ← Y / T

Instruções com 3 operandos

Instrução	Comentário
MOVE Y, A	Y ← A
SUB Y, B	Y ← Y – B
MOVE T, D	T ← D
MPY T, E	T ← T * E
ADD T, C	T ← T + C
DIV Y, T	Y ← Y / T

Instruções com 2 operandos

Instrução	Comentário
LOAD D	AC ← D
MPY E	AC ← AC * E
ADD C	AC ← AC + C
STOR Y	Y ← AC
LOAD A	AC ← A
SUB B	AC ← AC – B
DIV Y	AC ← AC / Y
STOR Y	Y ← AC

Instruções com 1 operando

Adaptado de: <http://www.univasf.edu.br/~fabio.nelson/arq/aoc1/aula_10.pdf>.

Enigma.

Por que o formato de instrução tem o máximo de três campos de operandos?

Porque a ULA tem, no máximo, duas entradas para operandos. E o terceiro campo de operando, quando utilizado, armazenará o endereço de memória onde o resultado da operação deve ser armazenado.

MODOS DE ENDEREÇAMENTO

Endereçamento é a forma como os operandos são referenciados no campo de operando, na instrução de máquina.

Existem diferentes formas de endereçamento para atender a necessidades de diferentes tipos de instruções.

(a) Endereçamento imediato:

- Campo do operando contém o próprio valor do operando (Figura 3).
- Vantagem: não requer acesso à memória.
- Desvantagem: limitação do tamanho do operando, o que reduz o valor máximo do dado a ser manipulado.

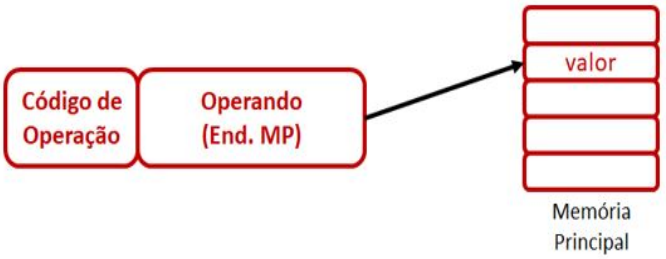
Figura 3 – Modo de endereçamento imediato

Fonte: elaborada pela autora.

(b) Endereçamento direto:

- Campo do operando contém o endereço do operando na memória principal (Figura 4).
- Acesso ao valor a ser manipulado por meio de uma única referência à memória.
- Vantagem: simplicidade do endereçamento.
- Desvantagem: espaço de endereçamento limitado.

Figura 4 – Modo de endereçamento direto



Fonte: elaborada pela autora.

(c) Endereçamento indireto:

- Campo do operando contém um endereço na memória principal. Este endereço armazena outro endereço de memória principal onde está o operando (Figura 5).
- Vantagem: grande espaço de endereçamento.
- Desvantagem: necessita de dois acessos à memória principal para obter o valor que será manipulado.

Figura 5 – Modo de endereçamento indireto

Fonte: elaborada pela autora.

(d) Endereçamento via registrador:

- Campo do operando contém o endereço do operando em um registrador (Figura 6).
- Utilizado em programação com linguagem de baixo nível (por exemplo: Assembly).
- Vantagens: campo de endereço pequeno que não requer acesso à memória.
- Desvantagem: espeço de endereço muito limitado.

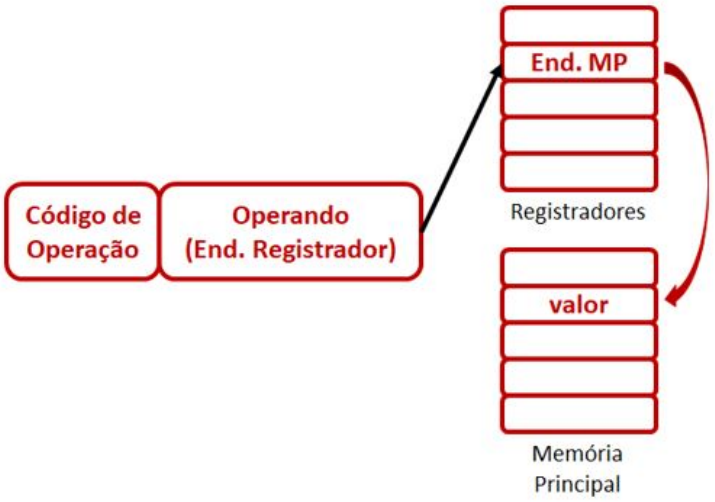
Figura 6 – Modo de endereçamento via registrador

Fonte: elaborada pela autora.

(e) Endereçamento indireto via registrador:

- Campo do operando contém o endereço de um registrador. O registrador armazena o endereço de memória principal onde o valor, a ser manipulado, está (Figura 7).
- Vantagem: grande espaço de endereçamento.
- Desvantagem: requer dois acessos de memória para obter o valor a ser manipulado. O primeiro acesso é realizado nos registradores públicos, e o segundo acesso é feito na memória principal.

Figura 7 – Modo de endereçamento indireto via registrador



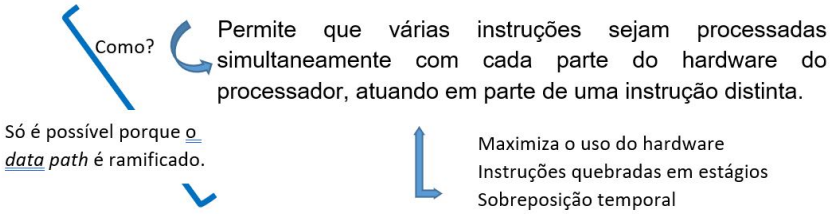
Fonte: elaborada pela autora.

PARALELISMO DE INSTRUÇÃO

Como conseguir um melhor desempenho de sistemas computacionais?

A estratégia da pipeline se aproxima à linha de montagem de uma indústria:

- Um produto passa por vários estágios de produção.
- Produtos em vários estágios do processo de produção podem ser trabalhados simultaneamente.
- Novas entradas são aceitas em uma extremidade antes que entradas aceitas previamente apareçam na saída.



Importante!

Tempo de execução de uma instrução é o mesmo ou maior, MAS o tempo de execução de todo o programa é muito menor à Pipeline melhora o *throughtput*

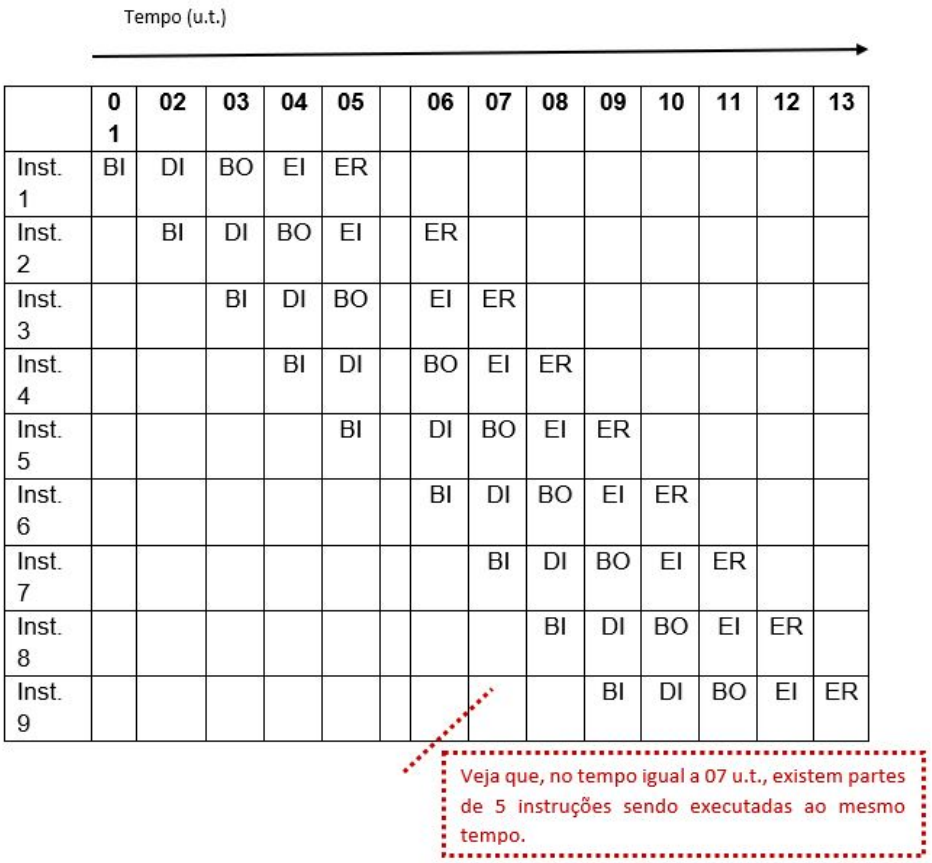
Requisitos para se utilizar uma pipeline:

- Instruções possuem vários estágios para execução
- Estágios com o mesmo tempo de duração
- Estágios independentes entre as instruções, garantindo a execução em paralelo
- Supõe-se que não há conflito de acesso à memória

As instruções são divididas normalmente em cinco estágios. Considera-se a seguinte decomposição do processamento de instrução:

- Busca de instrução (BI)
- Decodificação da instrução (DI)
- Busca dos operandos (BO)
- Execução da instrução (EI)
- Escrita do resultado (ER)

Por exemplo: execução de uma pipeline de 9 instruções e cada instrução com 5 estágios.



	01	02	03	04	05		06	07	08	09	10	11	12	13
Inst. 1	BI	DI	BO	EI	ER									
Inst. 2		BI	DI	BO	EI		ER							
Inst. 3			BI	DI	BO		EI	ER						
Inst. 4				BI	DI		BO	EI	ER					
Inst. 5					BI		DI	BO	EI	ER				
Inst. 6							BI	DI	BO	EI	ER			
Inst. 7								BI	DI	BO	EI	ER		
Inst. 8									BI	DI	BO	EI	ER	
Inst. 9										BI	DI	BO	EI	ER

9 instruções finalizadas à com pipeline – 13 u.t. (unidades de tempo)

à sem pipeline – 5 estágios * 9 instruções = 45 u.t.

A execução sequencial nem sempre acontecerá durante a execução de um processo (programa). Assim, a pipeline deve lidar com os possíveis desvios condicionais que podem acontecer. Esses desvios podem ser desvios de instruções, como em um if-then-else, mas também pode acontecer para atender a interrupções (Figura 8).

Figura 8 – Exemplo de desvio condicional

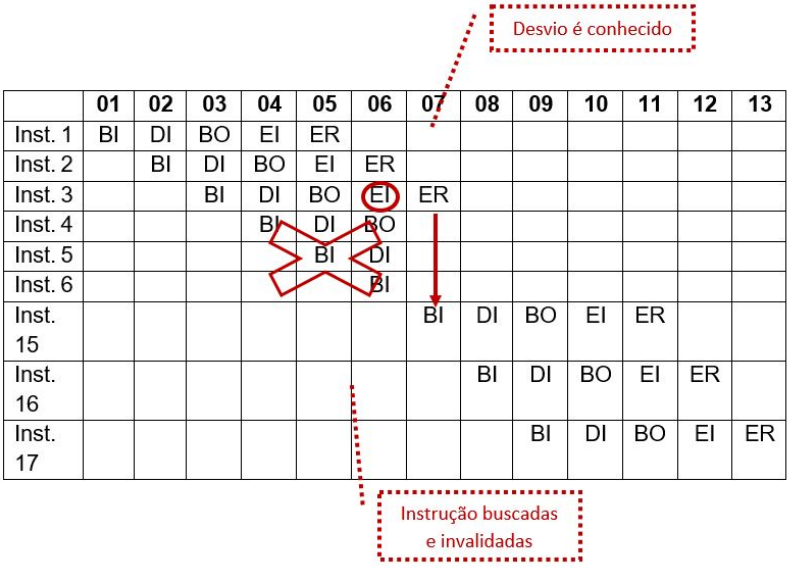
Figura 8 – Exemplo de desvio condicional

```
Inst. 3     if (a > b) ..... DESVIO PARA INST. 15 se condição for F
Inst. 4        then comando1
Inst. 5             comando2
Inst. 6             comando3
...                    ...
Inst. 14            comando11
Inst. 15      else  comando1
Inst. 16            comando2
Inst. 17    a = b * 4
...              ...
```

Fonte: elaborada pela autora.

O desvio condicional limita o aumento de desempenho porque, até que a instrução de desvio seja executada, não há como saber qual instrução virá a seguir. Quando o desvio acontece, diversas buscas de instruções são invalidadas.

Por exemplo: execução de uma pipeline de 9 instruções e cada instrução com 5 estágios. Há um desvio da instrução 3 para a 15.



	01	02	03	04	05	06	07	08	09	10	11	12	13
Inst. 1	BI	DI	BO	EI	ER								
Inst. 2		BI	DI	BO	EI	ER							
Inst. 3			BI	DI	BO	EI	ER						
Inst. 4				BI	DI	BO							
Inst. 5					BI	DI							
Inst. 6						BI							
Inst. 15							BI	DI	BO	EI	ER		
Inst. 16								BI	DI	BO	EI	ER	
Inst. 17									BI	DI	BO	EI	ER

6 instruções finalizadas à com pipeline – 13 u.t. (unidades de tempo)

à sem pipeline – 5 estágios * 6 instruções = 30 u.t.