

ARQUITETURA DE COMPUTADORES

Unidade Central de Processamento – U C P

Plano de Aula

2

□ Aula Passada...

□ Definição

■ Evolução dos Computadores

- Histórico

■ Modelo de Von-Neumann

- Básico

- CPU

- Memória

- E/S

- Barramentos

Plano de Aula

3

□ Unidade Central de Processamento

□ Composição

■ UC - ULA

■ Função

■ Principais registradores

□ Pipeline

■ Definição

■ Funcionamento

■ Problemas

■ Soluções

Arquitetura de Computadores

4

Unidade Central De Processamento

Arquitetura de Computadores

Unidade Central de Processamento

5

- A Unidade Central de Processamento - CPU, desempenha um papel vital no funcionamento de um sistema de computação.



Arquitetura de Computadores

Unidade Central de Processamento

6

- O processador é o componente vital do sistema de computação
 - ▣ Responsável pela realização das operações de processamento (os cálculos matemáticos etc.)
 - ▣ Responsável pelo controle, durante a execução de um programa.
 - Um programa, para ser efetivamente executado pelo processador, deve ser constituído de uma série de instruções de máquina.
 - Para que a execução tenha início, as instruções devem ser armazenadas em células sucessivas, na memória principal.

Arquitetura de Computadores

Unidade Central de Processamento

7

□ Composição da UCP

- Formado basicamente por um conjunto de registradores
 - Armazenam dados e instrução correntes.
- Os registradores são utilizados como locais de armazenamento temporário de dados provenientes da memória destinados à UAL, ou vice-versa.
- Existem ainda registradores cuja função é armazenar informações relevantes ao controle do processamento.
- Os principais registradores utilizados na arquitetura de Von Neumann e suas respectivas funções são os seguintes

Arquitetura de Computadores

Unidade Central de Processamento

8

□ Composição da UCP

■ Os principais registradores utilizados na arquitetura de Von Neumann e suas respectivas funções são os seguintes:

- Contador de Programa (**PC**, do inglês “Program Counter”) :
 - Contém o endereço da memória cujo conteúdo deve ser interpretado como a próxima instrução.
- Registrador de Instrução (**IR**, do inglês “Instruction Register”):
 - Contém a próxima instrução a ser executada.
- Registrador de Endereço da Memória (**MAR**, do inglês “Memory Address Register”):
 - Contém o endereço da posição da memória a ser lida ou escrita.

Arquitetura de Computadores

Unidade Central de Processamento

9

□ Composição da UCP

■ Os principais registradores utilizados na arquitetura de Von Neumann e suas respectivas funções são os seguintes:

- Registrador de Dados da Memória (**MDR**, do inglês “Memory Data Register”):
 - Contém o dado a ser lido ou escrito na memória.
- Registradores de Propósito Geral (**R**):
 - São utilizados para armazenar resultados intermediários

Arquitetura de Computadores

Unidade Central de Processamento

10

□ Função da CPU

□ Buscar uma instrução na memória

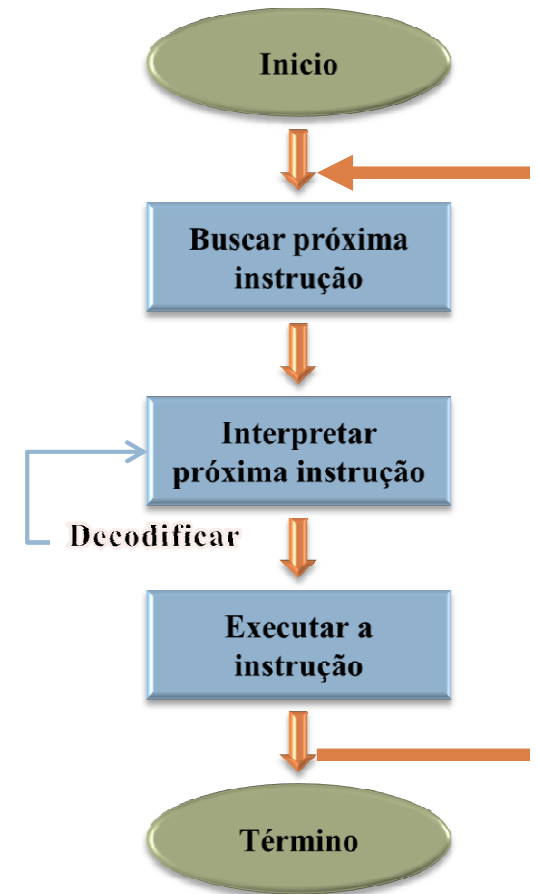
■ Operação de leitura

- Uma de cada vez

□ Interpretar que operação a instrução está explicitando

■ Exemplos:

- Soma de dois números,
- Uma multiplicação,
- Uma operação de E/S de dados,
- Uma operação de movimentação de um dado de uma célula para outra.



Fluxo básico de um ciclo de instrução

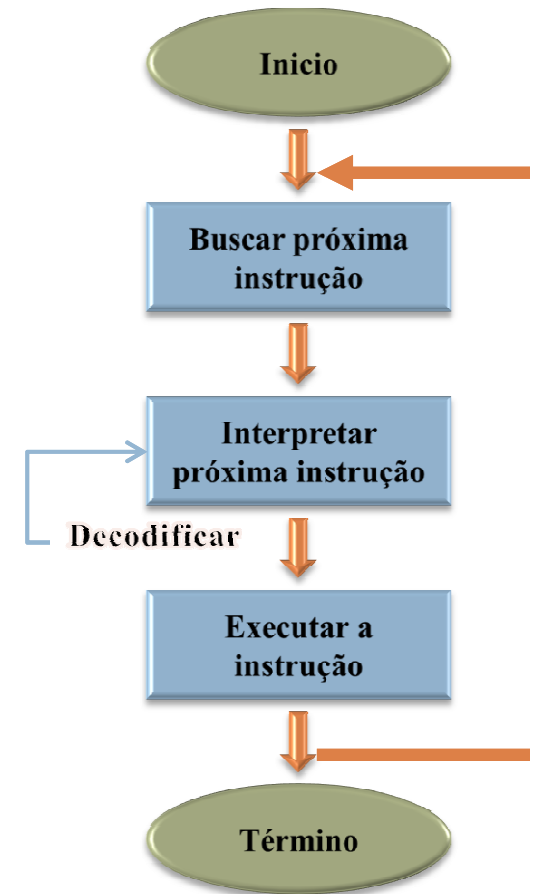
Arquitetura de Computadores

Unidade Central de Processamento

11

□ Função da CPU

- Buscar os dados onde estiverem armazenados
 - Trazê-los até a UCP;
- Executar efetivamente a operação com o(s) dado(s)
- Guardar o resultado (se houver algum) no local definido na instrução;
- Reiniciar o processo apanhando nova instrução.



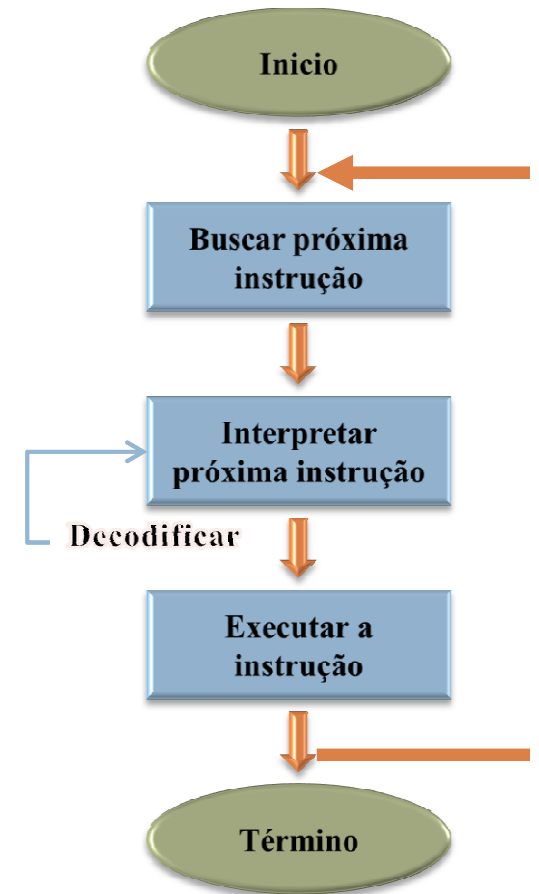
Fluxo básico de um ciclo de instrução

Arquitetura de Computadores

Unidade Central de Processamento

12

- Função da CPU (Resumindo...)
 - Buscar uma instrução na memória
 - Decodificar a operação
 - Buscar os dados
 - Executar a operação com o(s) dado(s)
 - Guardar o resultado
 - Reiniciar o processo



Fluxo básico de um ciclo de instrução

Arquitetura de Computadores

Unidade Central de Processamento

13

□ Função da CPU

- As etapas mencionadas, compõem o que se denomina um ***ciclo de instrução***.
- Esse ciclo se ***repete indefinidamente*** até que o sistema seja **desligado**, ou ocorra algum tipo de **erro**, ou seja encontrada uma instrução de **parada**.
- Em outras palavras, a UCP é projetada e fabricada com o **propósito único de executar** sucessivamente pequenas **operações matemáticas** (ou outras manipulações simples com dados), **na ordem** e na **seqüência definidas** pela **organização do programa**.

Arquitetura de Computadores

Unidade Central de Processamento

14

- As atividades realizadas pela UCP podem ser divididas em duas grandes categorias funcionais:
 - ▣ Função processamento
 - Se encarrega de realizar as atividades relacionadas com a efetiva execução de uma operação, ou seja, *processar*.
 - ▣ Função controle
 - É exercida pelos componentes da UCP que se encarregam das atividades de **busca**, **interpretação** e **controle** da execução das instruções
 - e do controle da ação dos demais componentes do sistema de computação
 - Memória - E/S).

Arquitetura de Computadores

Unidade Central de Processamento

15

- Conceitualmente, podemos imaginar que uma UCP simples pode ser dividida nestas duas áreas:
 - ▣ Função processamento
 - ▣ Função controle.
- Esta divisão é bem apropriada para caracterizarmos e entendermos melhor o processo de execução serial de uma instrução.

Arquitetura de Computadores

Unidade Central de Processamento

16

- **Processo serial** é aquele no qual cada pequena atividade do ciclo de instrução é realizada em seqüência à anterior
 - Algumas poucas atividades podem ser realizadas simultaneamente
- É um processo lento e pouco eficiente, se desejarmos maior velocidade de processamento.

Arquitetura de Computadores

Unidade Central de Processamento

17

- Unidades lógicas e aritméticas (ALU)
 - ▣ A unidade lógica e aritmética (ULA), como o nome indica, é responsável pela realização de operações:
 - Lógicas
 - Testes e comparações
 - Aritméticas
 - Somas e subtrações

Arquitetura de Computadores

Unidade Central de Processamento

18

- Unidade de controle e microprogramação
- A unidade de controle (UC) é responsável pelo controle das atividades de todos os componentes do computador, através da emissão de pulsos elétricos (sinais de controle).
 - ▣ Este controle pode ser a gravação de um dado no disco ou a busca de uma instrução da memória.

Arquitetura de Computadores

Unidade Central de Processamento

19

- Unidade de controle e microprogramação
- Unidade de controle é um circuito lógico responsável pelo funcionamento da máquina.
 - ▣ Este funcionamento realiza-se em ciclos, onde as seguintes operações são executadas :
 - Transferência do conteúdo do PC para o MAR, busca da próxima instrução e sua transferência do MDR para o IR (em inglês, “instruction fetch”).
 - Decodificação da instrução pelo circuito lógico da unidade de controle.

Arquitetura de Computadores

Unidade Central de Processamento

20

- Unidade de controle e microprogramação
- Unidade de controle é um circuito lógico responsável pelo funcionamento da máquina.
 - ▣ Este funcionamento realiza-se em ciclos, onde as seguintes operações são executadas :
 - Transferência do endereço do operando (contido no IR) para o MAR e leitura ou escrita do operando. No caso de leitura, o valor lido deve ser posteriormente transferido do MDR para um registrador de propósito geral (em inglês, “operand fetch”). No caso de escrita, o conteúdo de um registrador deve ser previamente transferido para o MDR. Algumas instruções não têm operandos.

Arquitetura de Computadores

Unidade Central de Processamento

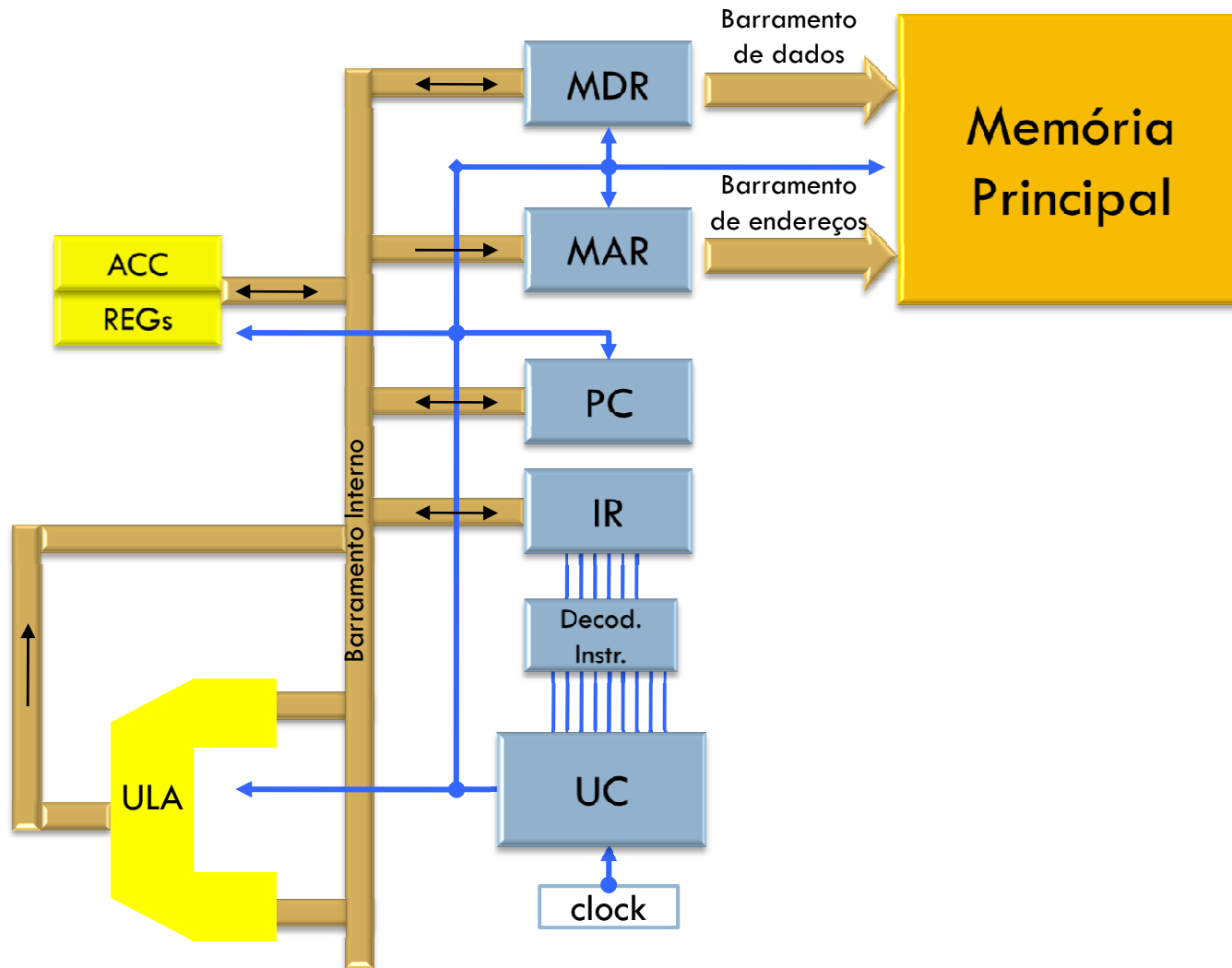
21

- Unidade de controle e microprogramação
- Unidade de controle é um circuito lógico responsável pelo funcionamento da máquina.
 - ▣ Este funcionamento realiza-se em ciclos, onde as seguintes operações são executadas :
 - Execução da instrução, tipicamente utilizando a UAL para realizar alguma operação sobre os valores contidos nos registradores de propósito geral.
 - Volta a 1.

Arquitetura de Computadores

Unidade Central de Processamento

22



Arquitetura de Computadores

Unidade Central de Processamento

23

- Unidade de controle e microprogramação
- A maneira como a unidade de controle gerencia este funcionamento é baseada em um relógio
 - Marca o início de cada nova operação do ciclo, e em circuitos lógicos capazes de, a cada ciclo, ativar ou desativar as conexões adequadas entre os diversos componentes de modo que a instrução correspondente seja executada.
 - A ativação e desativação é realizada através de Portas Lógicas, por exemplo do tipo E (em inglês, "gate and").



Arquitetura de Computadores

24

Pipeline

Arquitetura de Computadores

Unidade Central de Processamento

25

- Atualmente, a grande maioria dos sistemas de computação é fabricado em torno de processadores que buscam maior velocidade na realização de suas atividades.
- Um dos processos mais usados é o que conhecemos na indústria como linha de montagem ("*pipeline*"), no qual a UCP se **divide em várias partes funcionais** distintas (estágios), cada uma correspondendo a uma determinada atividade.

Arquitetura de Computadores

Unidade Central de Processamento

26

□ Paralelismo a Nível de Instrução

- É possível obter maior desempenho computacional com:
 - Tecnologias mais avançadas, tais como circuitos mais rápidos;
 - Melhor organização da CPU, tais como o uso de múltiplos registradores e memória cache
 - Pipeline de instruções.
- A idéia básica num **pipeline de instruções** é a de **novas entradas serem aceitas, antes que as entradas aceitas previamente tenham terminado.**
 - Este conceito assume que uma instrução tem vários estágios.

Arquitetura de Computadores

Unidade Central de Processamento

27

- Pipeline:
 - Várias instruções são realizadas de forma simultânea, embora em estágios diferentes.
- Exemplo
 - Lavanderia
 - Colocar uma carga de roupa na lavadora. (L) (30)
 - Quando terminar de lavar, colocar na secadora. (S)(40)
 - Quando terminar de secar, dobrá-las. (D)(20)
 - 90 minutos depois, começar de novo a lavar ...

Arquitetura de Computadores

Unidade Central de Processamento

28

□ Pipeline:

- Várias instruções são realizadas de forma simultânea, embora em estágios diferentes.

□ Exemplo

□ Lavanderia

Pessoa	L	S	D	L	S	D	L	S	D	L	S	D
a	x	x	x									
b				x	x	x						
c							x	x	x			
d										x	x	x

- Tempo Total = 4 x 90minutos = 6horas

Arquitetura de Computadores

Unidade Central de Processamento

29

□ Pipeline:

- Várias instruções são realizadas de forma simultânea, embora em estágios diferentes.

□ Exemplo

□ Lavanderia

Pessoa	L	S	D	L	S	D	L	S	D	L	S	D
a	x	x	x									
b		x	x	x								
c			x	x	x							
d				x	x	x						

- Tempo Total = 30 + 4 x 40m + 20m = 3 horas e meia

Arquitetura de Computadores

Unidade Central de Processamento

30

□ Pipeline:

- Várias instruções são realizadas de forma simultânea, embora em estágios diferentes.

□ Exemplo

□ Lavanderia

	30	40	40	40	40	20						
Pessoa	L	S	D	L	S	D	L	S	D	L	S	D
a	x	x	x									
b		x	x	x								
c			x	x	x							
d				x	x	x						

- Tempo Total = $30 + 4 \times 40m + 20m = 3 \text{ horas e meia}$

Arquitetura de Computadores

Unidade Central de Processamento

31

- Pipeline:
 - ▣ O tempo para cada instrução é o mesmo que sem pipeline.
 - ▣ O tempo total, de várias instruções, é menor.
 - ▣ Potencialmente 3 vezes mais rápido que o processo não pipeline.
 - ▣ Para exemplo anterior:
 - Pipeline: 4 tarefas em 6 ciclos.
 - Sem Pipeline: 4 tarefas em 12 ciclos.
 - Pipeline: 100 tarefas em 102 ciclos.
 - Sem Pipeline: 100 tarefas em 300 ciclos.

Arquitetura de Computadores

Unidade Central de Processamento

32

□ Pipeline:

□ Considerações

- O tempo gasto para se limpar uma sacola de roupa não diminuiu, na realidade até aumentou, pois como pode se ver, sem o pipelining eram necessários 30 minutos para lavar, 40 minutos para secar e 20 minutos para dobrar a roupa, ou seja, 90 minutos para realizar todas estas etapas (pipe stage ou pipe segment), enquanto que com o pipeline os tempos variaram de 90 a 120 minutos.

Arquitetura de Computadores

Unidade Central de Processamento

33

□ Pipeline:

▣ Considerações

- A taxa de inserção de tarefas no pipeline é limitada pela tarefa mais lenta;
- Meta é aumentar a produtividade (throughput).
 - Enquanto que numa lavanderia sem pipelining temos somente uma sacola de roupas sendo limpa num determinado instante, numa outra lavanderia com pipelining temos até três sacolas de roupas sendo limpas simultaneamente, já que o pipeline de nossa lavanderia é composto de 3 estágios (profundidade 3).
 - Conseqüentemente a produtividade da lavanderia é aumentada em aproximadamente 3 vezes, já que a cada ciclo uma nova sacola de roupa é completamente limpa (lavada, secada e dobrada) e uma nova sacola de roupa começa a ser lavada;

Arquitetura de Computadores

Unidade Central de Processamento

34

□ Pipeline:

□ Considerações

- A produtividade da lavanderia é aumentada devido à sua capacidade de executar múltiplas tarefas ao mesmo tempo.
 - Exemplo sem o pipelining:
 - A cada 90 minutos uma sacola é completamente limpa,
 - Exemplo com o pipelining:
 - A cada 30 minutos, aproximadamente, uma nova sacola de roupas é completamente limpa.

$$\text{Tempo entre execuções}_{\text{pipeline}} = \frac{\text{Tempo entre execuções}_{\text{não pipeline}}}{\text{Número de estágios do pipeline}}$$

Arquitetura de Computadores

Unidade Central de Processamento

35

- Pipeline:
 - ▣ Considerações
 - O pipeline melhora a performance por meio do *throughput* das instruções.

Arquitetura de Computadores

Unidade Central de Processamento

36

□ Pipeline:

□ Exemplo 2: Ciclo de instrução:

- Considere um pipeline com 6 estágios de mesma duração:
 - busca de instrução (BI);
 - decodificação de instrução (DI);
 - cálculo de operandos (CO);
 - busca de operandos (BO);
 - execução de instrução (EI);
 - escrita de operando (EO).

Arquitetura de Computadores

Unidade Central de Processamento

37

□ Pipeline:

▣ Exemplo 2: Ciclo de instrução:

busca de instrução (BI) - decodificação de instrução (DI) - cálculo de operandos (CO)

busca de operandos (BO)- execução de instrução (EI)- escrita de operando (EO).

tempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
instrução														
1	BI	DI	CO	BO	EI	EO								
2		BI	DI	CO	BO	EI	EO							
3			BI	DI	CO	BO	EI	EO						
4				BI	DI	CO	BO	EI	EO					
5					BI	DI	CO	BO	EI	EO				
6						BI	DI	CO	BO	EI	EO			
7							BI	DI	CO	BO	EI	EO		
8								BI	DI	CO	BO	EI	EO	
9									BI	DI	CO	BO	EI	EO

Arquitetura de Computadores

Unidade Central de Processamento

38

□ Pipeline:

□ Exemplo 2: Ciclo de instrução:

- Consideramos que cada instrução passa por todos os estágios do pipeline.
- Consideramos que todos os estágios podem ser executados em paralelo
 - Não há conflito, por exemplo, nos acesso à memória (o dado pode estar no cache ou alguns estágios, que requerem acesso à memória, não estão sendo usados).

Arquitetura de Computadores

Unidade Central de Processamento

39

□ Pipeline:

□ Exemplo 2: Ciclo de instrução:

- Se os seis estágios **não têm** duração igual, existe certa **espera envolvida** em vários estágios.
- Uma instrução de **desvio condicional** pode **invalidar** diversas buscas de instrução. Da mesma forma, a ocorrência de interrupção.

Arquitetura de Computadores

Unidade Central de Processamento

40

□ Pipeline:

□ Exemplo 2: Ciclo de instrução:

- Instrução 3 = um desvio condicional para a instrução 15.

tempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
instrução														
1	BI	DI	CO	BO	EI	EO								
2		BI	DI	CO	BO	EI	EO							
3			BI	DI	CO	BO	EI	EO						
4				BI	DI	CO	BO							
5					BI	DI	CO							
6						BI	DI							
7							BI							
15								BI	DI	CO	BO	EI	EO	
16									BI	DI	CO	BO	EI	EO

Arquitetura de Computadores

Unidade Central de Processamento

41

□ Pipeline:

□ Conflitos do Pipeline

- Eventos em que a instrução seguinte não pode ser executada.
- 3 tipos:
 - Estrutural
 - Controle
 - Dados

Arquitetura de Computadores

Unidade Central de Processamento

42

- Pipeline:
 - Conflito Estrutural
 - O hardware não pode suportar a combinação de instruções que o pipeline deseja executar no mesmo ciclo de clock.
 - Exemplos: Lavagem de roupa.
 - A lavadora e a secadora estão combinadas em um mesmo dispositivo, e não separados.
 - A empregada está ocupada em alguma outra coisa, e não pode guardar as roupas após dobrá-las.

Arquitetura de Computadores

Unidade Central de Processamento

43

- Pipeline:
 - ▣ Conflito de Controle
 - Ocorre quando há necessidade de se tomar uma decisão com base nos resultados de uma instrução enquanto outras estão sendo executadas.

Arquitetura de Computadores

Unidade Central de Processamento

44

□ Pipeline:

□ Conflito de Controle

■ Instrução de desvio condicional:

- Se o computador tiver que desviar, deverá interromper as instruções do pipeline.
- Solução:
 - Se houvesse um hardware extra pode testar os registradores e calcular o endereço de desvio condicional e atualizar o PC durante o segundo estágio.
 - Mesmo assim, o pipeline necessitaria um atraso.
 - Este atraso, é conhecido como **parada do pipeline**.
 - Também conhecido como **bolha**.

Arquitetura de Computadores

Unidade Central de Processamento

45

□ Pipeline:

▣ Conflito de Controle

■ Instrução de desvio condicional:

tempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
instrução														
1	BI	DI	CO	BO	EI	EO								
2		BI	DI	CO	BO	EI	EO							
3			BI	DI	CO	BO	EI	EO						
4				BI	DI	CO	BO							
5					BI	DI	CO							
6						BI	DI							
7							BI							
15								BI	DI	CO	BO	EI	EO	
16									BI	DI	CO	BO	EI	EO

Arquitetura de Computadores

Unidade Central de Processamento

46

□ Pipeline:

□ Conflito de Controle

- Instrução de desvio condicional, Soluções...

- Múltiplos fluxos

- Consiste em duplicar os estágios iniciais do pipeline para permitir a busca de ambas as instruções, usando dois fluxos de instruções.

- Problemas:

- Uso de múltiplos pipelines introduz atrasos devidos à contenção de acesso a registradores e à memória.

- Pode ocorrer a entrada de instruções de desvio adicionais no pipeline, antes que seja tomada a decisão sobre o desvio original.

Arquitetura de Computadores

Unidade Central de Processamento

47

□ Pipeline:

□ Conflito de Controle

- Instrução de desvio condicional, Soluções...
- Busca antecipada da instrução-alvo do desvio
 - Consiste em buscar antecipadamente
 - a instrução-alvo do desvio
 - a instrução consecutiva ao desvio
 - no instante em que a instrução de desvio condicional é reconhecida.
 - A instrução-alvo é armazenada em um registrador, até que a instrução de desvio seja executada.

Arquitetura de Computadores

Unidade Central de Processamento

48

□ Pipeline:

□ Conflito de Controle

- Instrução de desvio condicional, Soluções...
- Memória de laço
 - Consiste em usar uma **pequena memória de alta velocidade**, mantida pelo estágio de busca de instrução, para guardar as n instruções buscadas mais recentemente, em seqüência.
 - Vantagens:
 - A memória conterà certo número de instruções que estão à frente da instrução corrente.
 - Se ocorrer um desvio para alguma posição adiante do endereço da instrução de desvio, essa posição já estará na memória de laço (útil em instruções do tipo IF-THEN-ELSE);

Arquitetura de Computadores

Unidade Central de Processamento

49

□ Pipeline:

□ Conflito de Controle

- Instrução de desvio condicional, Soluções...
- Memória de laço
 - Consiste em usar uma **pequena memória de alta velocidade**, mantida pelo estágio de busca de instrução, para guardar as n instruções buscadas mais recentemente, em seqüência.
 - Vantagens:
 - Particularmente adequada para lidar com laços de repetição ou iterações (se a memória for grande o suficiente para conter as instruções de uma iteração, estas terão que ser buscadas da memória apenas uma vez, para a primeira iteração).

Arquitetura de Computadores

Unidade Central de Processamento

50

□ Pipeline:

▣ Conflito de Controle

- Instrução de desvio condicional, Soluções...
- Previsão de desvio
- Feita de várias formas:
 - Prever que o desvio **nunca** será tomado:
 - Abordagem simples e estática, isto é, não depende do histórico das instruções até o momento em que ocorre a instrução de desvio condicional; continua buscando instruções na seqüência em que ocorrem no programa.

Arquitetura de Computadores

Unidade Central de Processamento

51

□ Pipeline:

▣ Conflito de Controle

- Instrução de desvio condicional, Soluções...
- Previsão de desvio
- Feita de várias formas:
 - Prever que o desvio **sempre** será tomado:
 - Abordagem simples e estática, isto é, não depende do histórico das instruções até o momento em que ocorre a instrução de desvio condicional; busca sempre as próximas instruções a partir do endereço-alvo do desvio.

Arquitetura de Computadores

Unidade Central de Processamento

52

□ Pipeline:

□ Conflito de Controle

- Instrução de desvio condicional, Soluções...
- Previsão de desvio
- Feita de várias formas:
 - Prever se o desvio será tomado ou não conforme o código de operação:
 - Abordagem simples e estática.
 - Prever o desvio com base em chaves de desvio tomado e de desvio não tomado:
 - Abordagem dinâmica, isto é, depende do histórico de execução.
 - Prever o desvio com base em uma tabela de histórico de desvios:
 - Abordagem dinâmica.

Arquitetura de Computadores

Unidade Central de Processamento

53

□ Pipeline:

▣ Conflito de Controle

- Instrução de desvio condicional, Soluções...
- Previsão de desvio

- Se a busca da instrução consecutiva à instrução de desvio causar uma **falta de página** ou uma **violação de proteção**, o processador **interromperá** a busca antecipada da instrução, até que tenha **certeza** de que essa instrução deve ser mesmo buscada.

Arquitetura de Computadores

Unidade Central de Processamento

54

□ Pipeline:

▣ Conflito de Controle

■ Instrução de desvio condicional (Considerações)

- Análises de comportamento de programas mostram que desvios condicionais são tomados em mais de 50% das vezes.
- Se o custo da busca antecipada de instruções for o mesmo em qualquer caminho, o resultado obtido deverá ser melhor se a busca antecipada de instruções for sempre efetuada a partir do endereço-alvo do desvio.

Arquitetura de Computadores

Unidade Central de Processamento

55

□ Pipeline:

▣ Conflito de Controle

■ Instrução de desvio condicional (Considerações)

- Em uma máquina que usa paginação, a busca antecipada de instruções, a partir do endereço de desvio, tem maior probabilidade de causar uma falta de página do que a busca de instruções consecutivas à instrução de desvio.
- A previsão de desvio com base no **código de operação da instrução** de desvio pressupõe que para determinados códigos o desvio é sempre tomado e para outros não, havendo um aproveitamento de **75%**.

Arquitetura de Computadores

Unidade Central de Processamento

56

□ Pipeline:

▣ Conflito de Controle

■ Instrução de desvio condicional (Considerações)

- **Estratégias dinâmicas** de previsão de desvio mantêm um histórico sobre as instruções de desvio condicional.
- Utilizando-se **somente um** bit de histórico, pode-se registrar se a última execução da instrução resultou em desvio ou não.
- Uma desvantagem neste caso ocorre quando o desvio é quase sempre tomado, tal como em instruções de desvio usadas para implementar laços de repetição. Sempre ocorrerão **dois erros** de previsão de desvio, cada vez que o laço de repetição for executado: uma vez na **entrada** e outra na **saída**.
- Atraso de desvio consiste em reordenar as instruções, de modo que as instruções de desvio ocorram mais tarde.

Arquitetura de Computadores

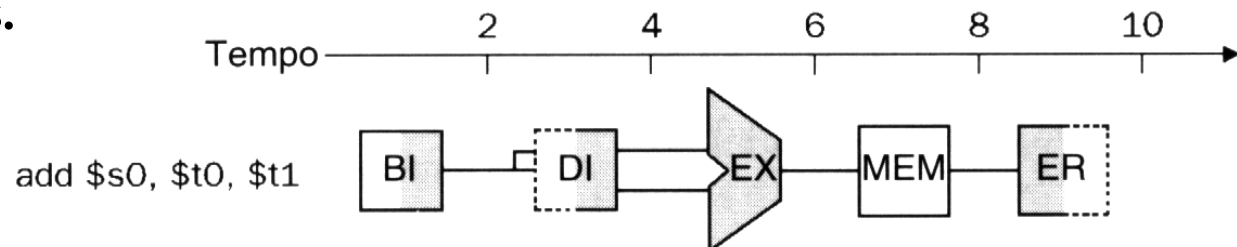
Unidade Central de Processamento

57

□ Pipeline:

▣ Conflito por Dados

- Ocorre quando o efeito de uma instrução depende do resultado de uma instrução anterior.
 - add \$s0, \$t0, \$t1
 - sub \$t2, \$s0, \$t3
- Se não houver intervenção, o pipeline deverá ser interrompido.
- O dado só será escrito no registrador no quinto ciclo:
- 3 bolhas.



Arquitetura de Computadores

Unidade Central de Processamento

58

□ Pipeline:

□ Conflito por Dados

■ Solução:

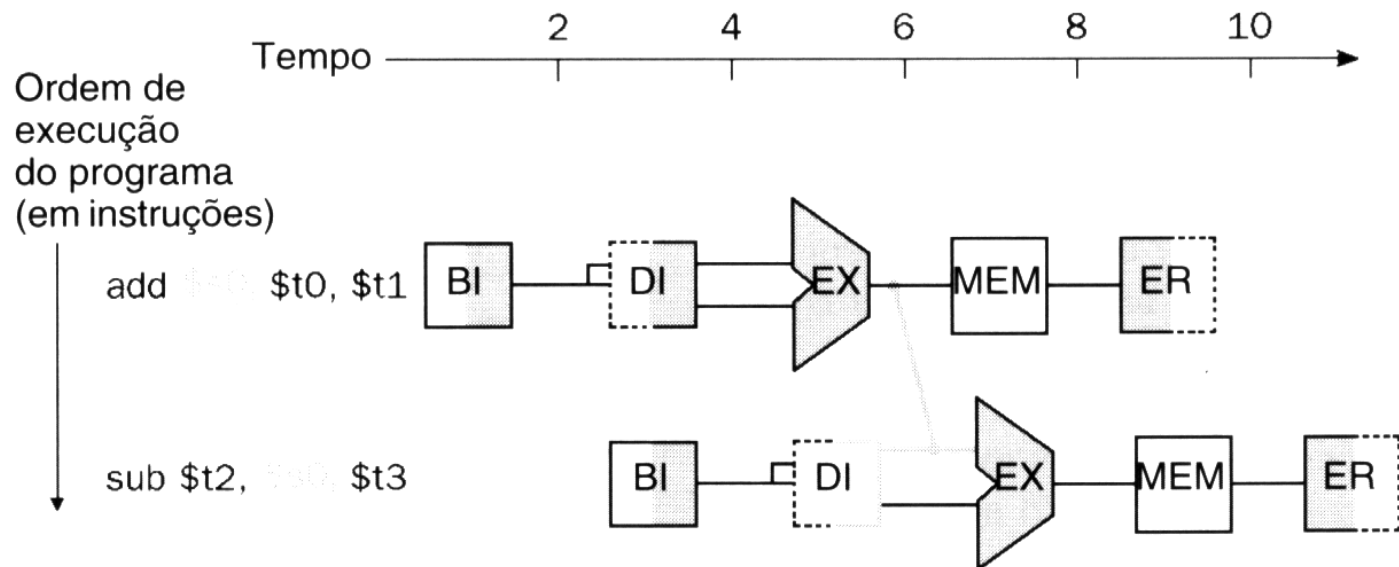
- Na verdade, o cálculo é realizado no terceiro ciclo.
- Uma via de bypass poderá enviar os dados à outra instrução, já no terceiro ciclo.
- Obtenção antecipada de determinado item faltante em uma operação, a partir de recursos internos da máquina: adiantamento ou *bypass*.

Arquitetura de Computadores

Unidade Central de Processamento

59

- Pipeline:
 - Conflito por Dados
 - Solução:



Arquitetura de Computadores

60

Exercícios

Arquitetura de Computadores

Unidade Central de Processamento

61

□ Exercícios:

- Como a CPU é dividida?
- Qual o Sistema Numérico que o computador adota?
 - Qual sua menor unidade?
- Qual o elemento do hardware que distribui as tarefas a todos os componentes do sistema?
- Suponha um processador que necessite executar 500 tarefas. Cada tarefa utiliza 5 ciclos de CPU. Quantos ciclos serão necessários em um processador com e sem pipeline?

Arquitetura de Computadores

Unidade Central de Processamento

62

□ Exercícios:

- Suponha um processador que necessite executar 500 tarefas. Cada tarefa utiliza 5 ciclos de CPU. Quantos ciclos serão necessários em um processador com e sem pipeline?

Ciclos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Tarefas	X	X	X	X	X												
						X	X	X	X	X							
											X	X	X	X	X		

Arquitetura de Computadores

Unidade Central de Processamento

64

□ Exercícios:

- Suponha um processador que necessite executar 500 tarefas. Cada tarefa utiliza 5 ciclos de CPU. Quantos ciclos serão necessários em um processador com e sem pipeline?

Sem Pipeline = $5 \times 500 = 2.500$ ciclos

Com Pipeline = $5 + 500 = 505$ ciclos

Tempo entre execuções = 5 ciclos

$$\text{Tempo entre execuções}_{\text{pipeline}} = \frac{\text{Tempo entre execuções}_{\text{não pipeline}}}{\text{Número de estágios do pipeline}}$$