

# Pipeline

Profa. Débora Christina Muchaluat Saade  
debora@midia.com.uff.br

<http://www.ic.uff.br/~debora/fac>

1

## Processo de Pipelining (exemplo)

✓ Ana, Bruno, Carla, Luiz têm roupas sujas a serem lavadas, secadas, dobradas e guardadas



✓ Lavadora leva 30 minutos



✓ Secadora leva 30 minutos



✓ “Dobrar” leva 30 minutos

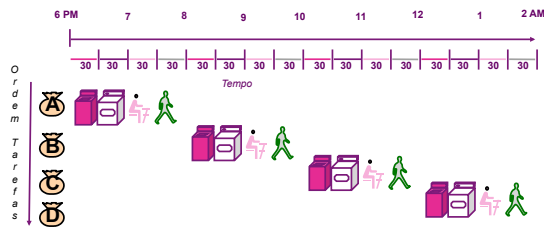


✓ “Guardar” leva 30 minutos



2

## Pipelining (exemplo da lavanderia)

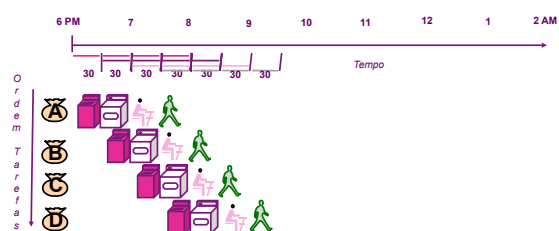


✓ Processo sequencial de lavagem leva oito horas para os quatro

✓ Quanto tempo levaria, utilizando-se pipelining?

5

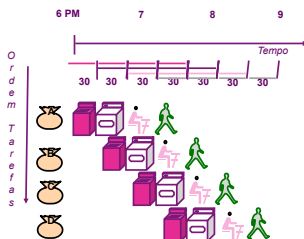
## Pipelining (exemplo da lavanderia)



✓ Utilizando-se a técnica de pipeline consome-se 3,5 horas no processo de lavagem!

4

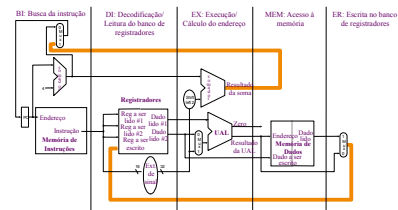
## Observações sobre Pipelining



- ✓ Pipelining não ajuda a melhorar a latência de uma atividade, mas aumenta o throughput
- ✓ Várias tarefas operando em paralelo utilizam recursos diversos
- ✓ Aceleração potencial = Número de estágios de pipe
- ✓ Taxa de pipeline limitada pelo estágio mais lento
- ✓ Desequilíbrios na duração dos estágios reduz a aceleração
- ✓ Tempo para “encher” o pipeline e para “esvaziá-lo” reduz a aceleração
- ✓ Pode parar por dependências

5

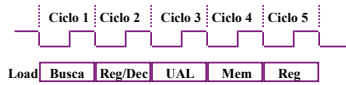
## Idéia Básica



6

## Os Cinco Estágios da Instrução de Carga

Fundamentos de Arquiteturas de Computadores

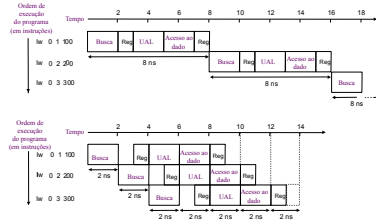


- ✓ **Busca:** Busca da instrução da memória de instruções
- ✓ **Reg/Dec:** Leitura do(s) registrador(es) e decodificação da Instrução
- ✓ **UAL:** Calcula o endereço da memória de dados
- ✓ **Mem:** Lê dado da memória de dados
- ✓ **Reg:** Escreve o dado no banco de registradores

7

## Pipelining

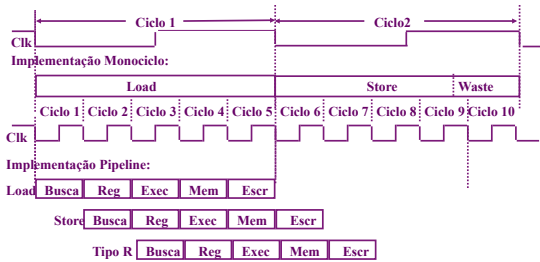
Fundamentos de Arquiteturas de Computadores



8

## Monociclo vs Pipeline

Fundamentos de Arquiteturas de Computadores



9

## Por que usar Pipeline?

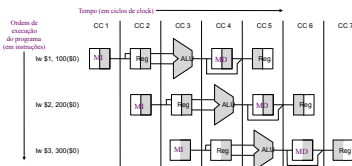
Fundamentos de Arquiteturas de Computadores

- ✓ Suponha que vão ser executadas 100 instruções
- ✓ **Máquina monociclo**
  - 1 ciclo de relógio tem duração de 45 ns
  - $45 \text{ ns/ciclo} \times 1 \text{ CPI} \times 100 \text{ inst} = 4500 \text{ ns}$
- ✓ **Máquina ideal pipelined**
  - 1 ciclo de relógio tem duração de 10 ns
  - cada estágio de pipeline utiliza um ciclo de relógio
  - $5 \times 1 \text{ ciclo} + (1 \text{ ciclo} \times (100 \text{ inst} - 1)) = 50 \text{ ns} + 99 \times 10 \text{ ns} = 1040 \text{ ns}$

10

## Representação Gráfica de Pipeline

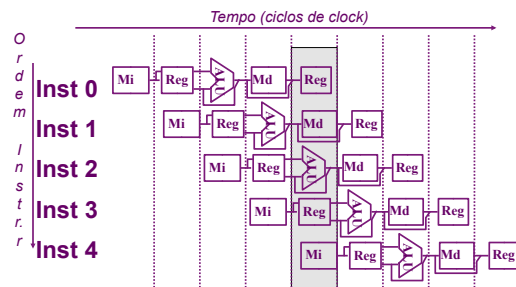
Fundamentos de Arquiteturas de Computadores



11

## Pipeline - Recursos disponíveis

Fundamentos de Arquiteturas de Computadores



12

## Conflitos (Hazards) do Pipeline

Fundamentos de Arquiteturas de Computadores

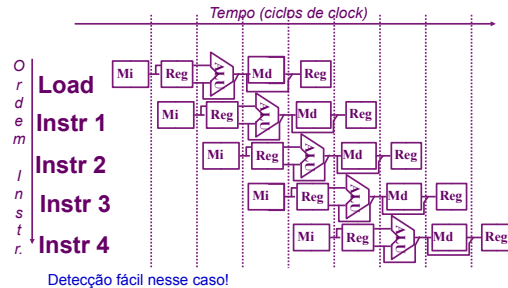
✓ **Conflitos estruturais:** tentativa de utilizar o mesmo recurso de modos diferentes ao mesmo tempo

- Ex.: lavadora/secadora combinadas, ou pessoa que dobra ocupada com outra atividade (lavando panela)
- sem duas memórias, não poderia ter acesso à instrução simultâneo com dados

13

## Conflito Estrutural - Memória Única

Fundamentos de Arquiteturas de Computadores



14

## Conflitos do Pipeline

Fundamentos de Arquiteturas de Computadores

✓ **Conflitos de controle:** tentativa de tomar uma decisão antes que a condição seja avaliada

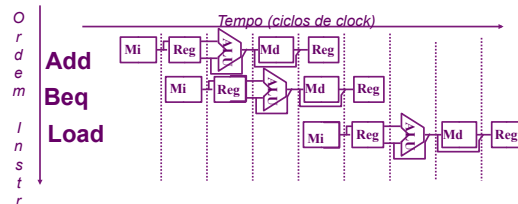
- Ex.: lavar uniformes de time de futebol e precisa saber quantidade de sabão; precisa esperar a secadora para colocar próximo uniforme
- instruções de desvio

15

## Soluções para Conflito de Controle

Fundamentos de Arquiteturas de Computadores

✓ **Parada:** espera até decisão estar clara



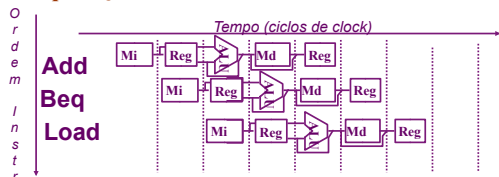
16

## Soluções para Conflito de Controle

Fundamentos de Arquiteturas de Computadores

✓ **Predição:** escolhe uma direção e retorna se errada

- predição não executada



✓ Certo - 50% do tempo

17

## Conflitos do Pipeline

Fundamentos de Arquiteturas de Computadores

✓ **Conflitos de dados:** tentativa de utilizar um item antes de estar pronto

- Ex.: uma meia na secadora e outra na lavadora; não pode dobrar
- instrução depende de resultado da instrução anterior, ainda no pipeline

18

## Conflito de Dados com r1

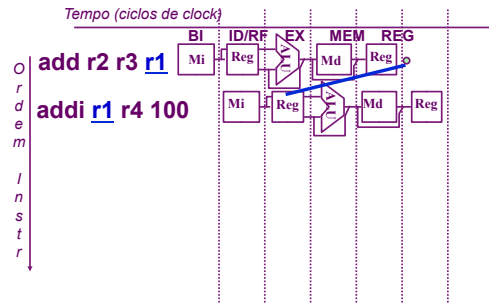
Fundamentos de Arquiteturas de Computadores

add r2 r3 **r1**  
 addi **r1** r4 100

19

## Conflito de Dados com r1

Fundamentos de Arquiteturas de Computadores



## Solução para Conflito de Dados

Fundamentos de Arquiteturas de Computadores

- “Adiantamento” (forwarding) do resultado de um estágio para outro

