Pipeline

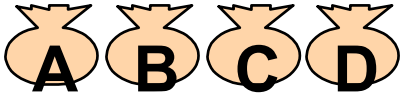
Profa. Débora Christina Muchaluat Saade

debora@midiacom.uff.br

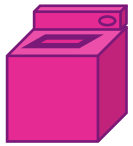
http://www.ic.uff.br/~debora/fac

Processo de Pipelining (exemplo)

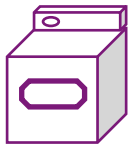
- ✓Ana, Bruno, Carla, Luiz têm roupas sujas a serem lavadas, secadas, dobradas e guardadas



- ✓Lavadora leva 30 minutos



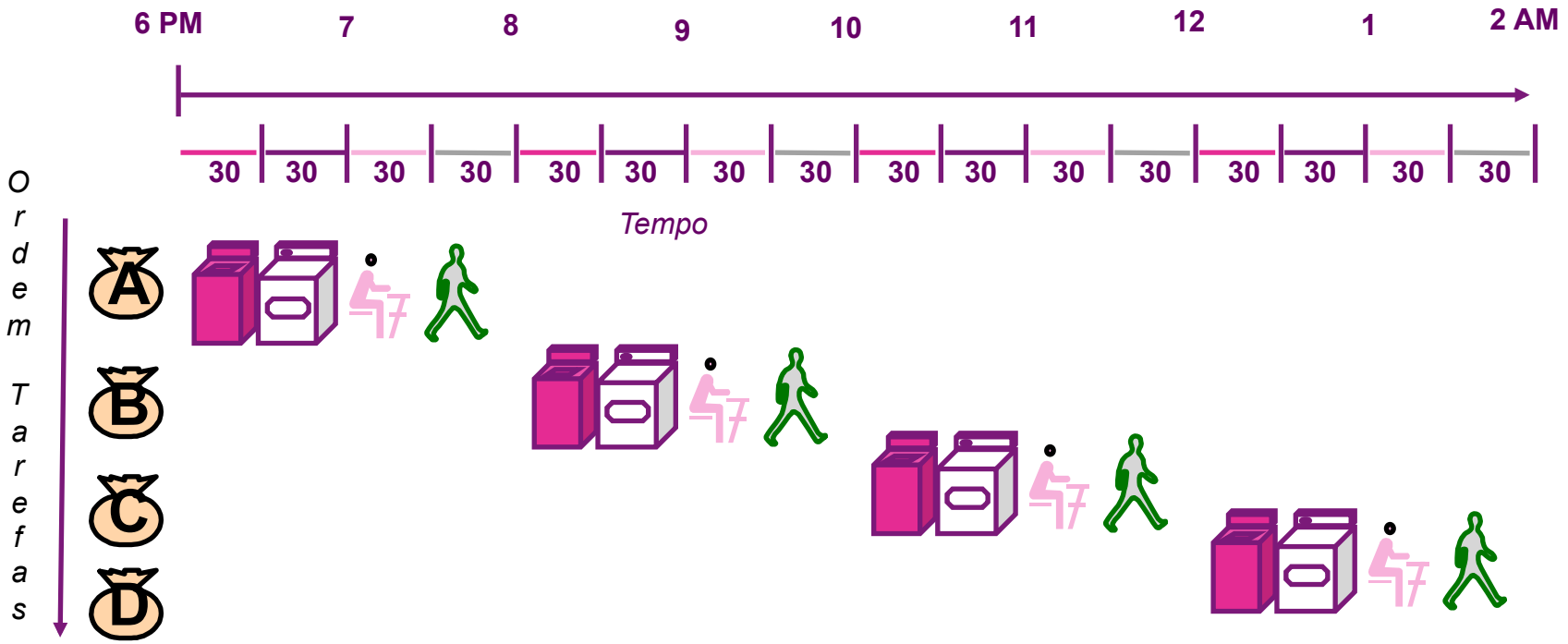
- ✓Secadora leva 30 minutos



- ✓"Dobrar" leva 30 minutos

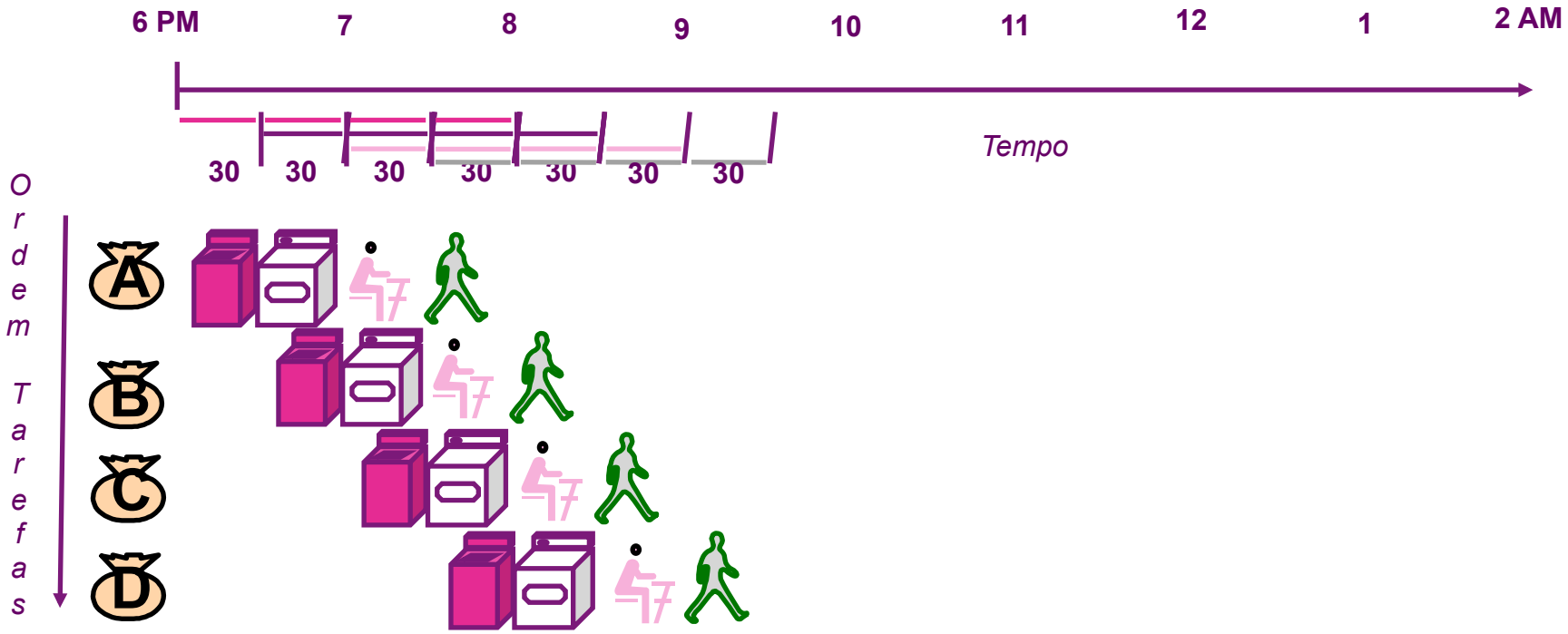
- ✓"Guardar" leva 30 minutos

Pipelining (exemplo da lavanderia)



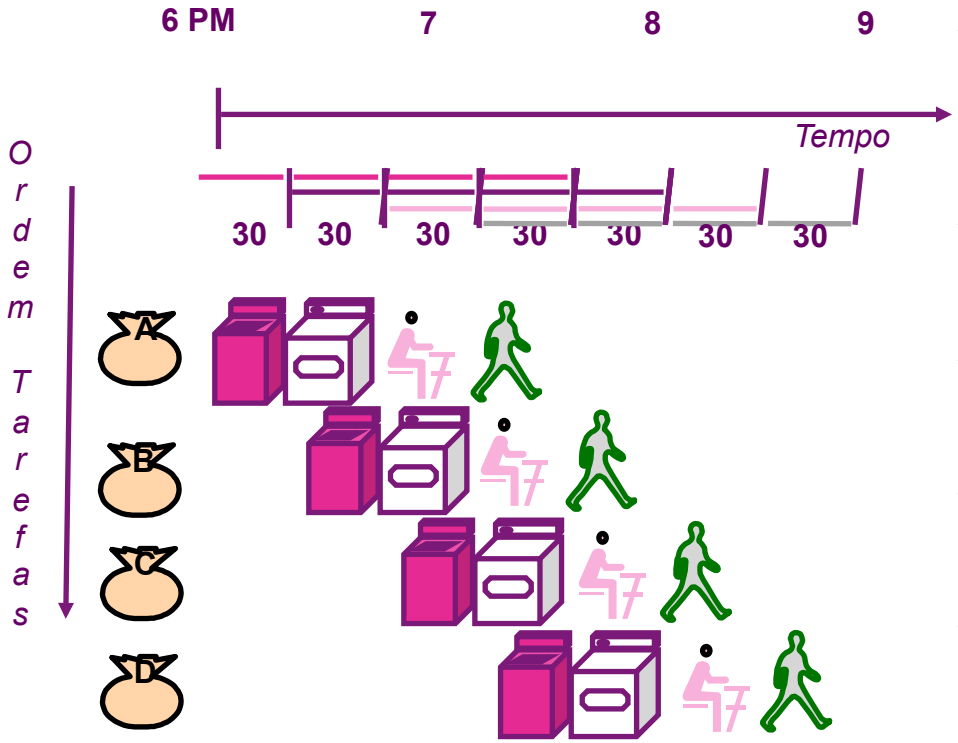
- ✓ **Processo sequencial de lavagem leva oito horas para os quatro**
- ✓ **Quanto tempo levaria, utilizando-se pipelining?**

Pipelining (exemplo da lavanderia)



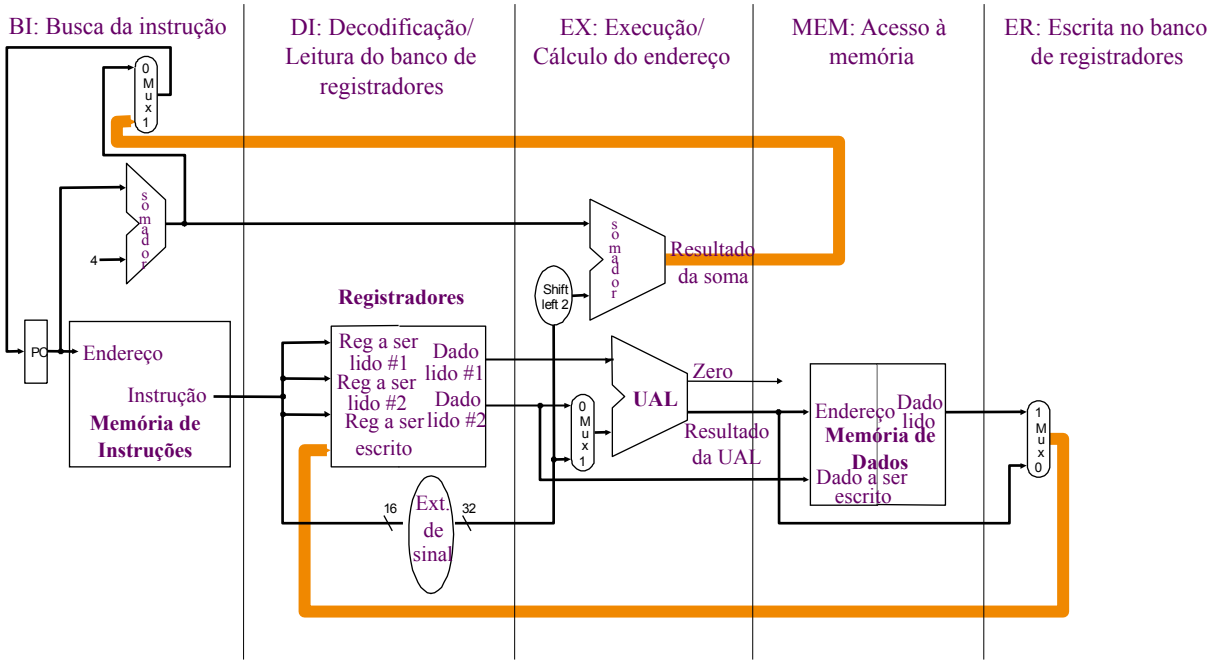
✓**Utilizando-se a técnica de pipeline consome-se 3,5 horas no processo de lavagem!**

Observações sobre Pipelining

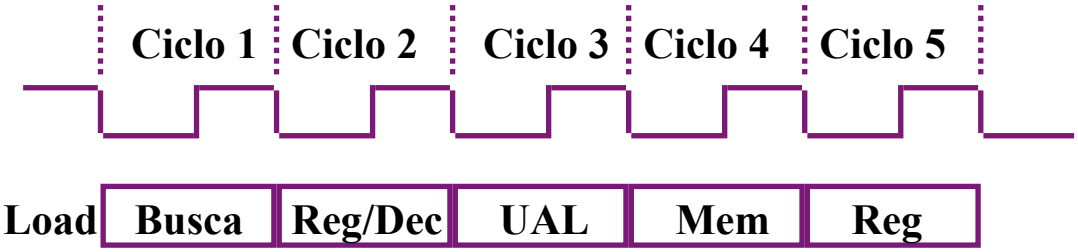


- ✓ **Pipelining não ajuda a melhorar a latência de uma atividade, mas aumenta o throughput**
- ✓ **Várias tarefas operando em paralelo utilizam recursos diversos**
- ✓ **Aceleração potencial = Número de estágios de pipe**
- ✓ **Taxa de pipeline limitada pelo estágio mais lento**
- ✓ **Desequilíbrios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

Idéia Básica

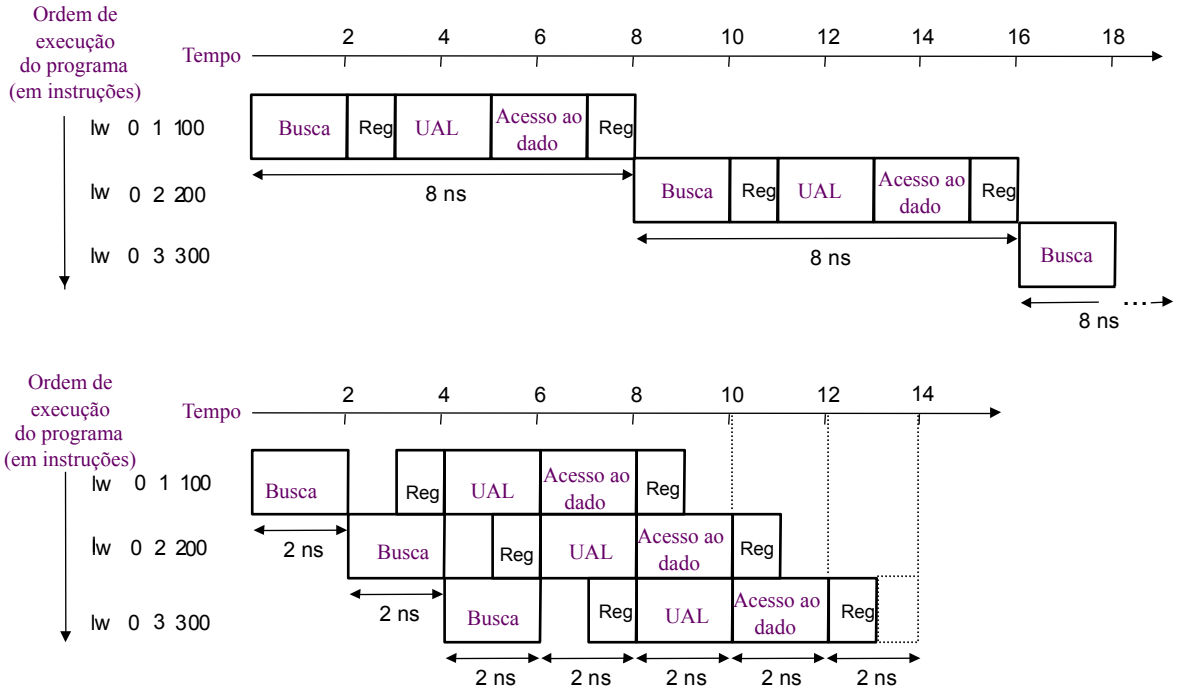


Os Cinco Estágios da Instrução de Carga

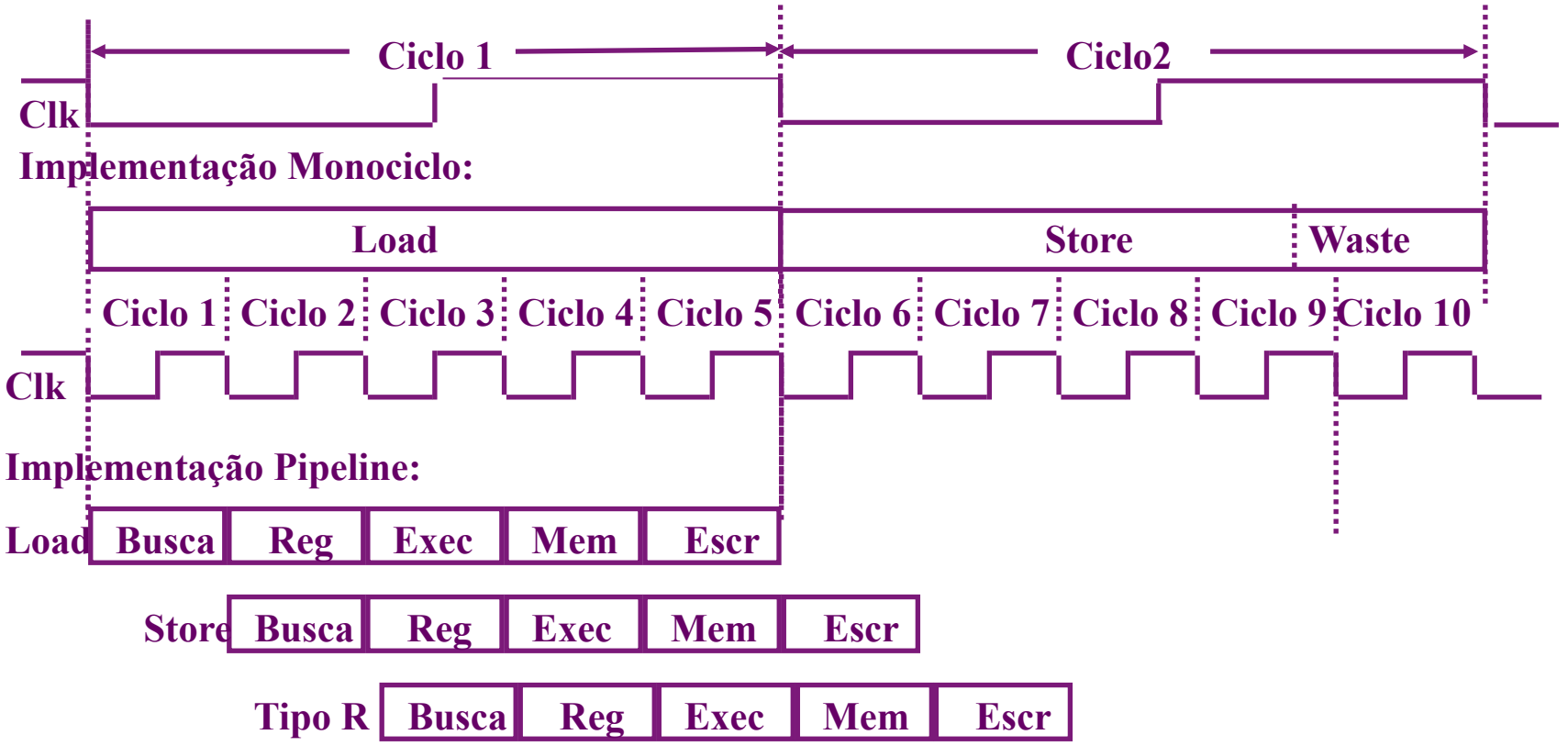


- ✓**Busca: Busca da instrução da memória de instruções**
- ✓**Reg/Dec: Leitura do(s) registrador(es) e decodificação da Instrução**
- ✓**UAL: Calcula o endereço da memória de dados**
- ✓**Mem: Lê dado da memória de dados**
- ✓**Reg: Escreve o dado no banco de registradores**

Pipelining


Ordem de execução do programa (em instruções)
Tempo
2 4 6 8 10 12 14 16 18
lw 0 1 100
lw 0 2 200
lw 0 3 300
Busca
Reg
UAL
Acesso ao dado
Reg
8 ns
Busca
Reg
UAL
Acesso ao dado
Reg
8 ns
Busca
8 ns
Ordem de execução do programa (em instruções)
Tempo
2 4 6 8 10 12 14
lw 0 1 100
lw 0 2 200
lw 0 3 300
Busca
Reg
UAL
Acesso ao dado
Reg
2 ns
Busca
Reg
UAL
Acesso ao dado
Reg
2 ns
Busca
Reg
UAL
Acesso ao dado
Reg
2 ns
2 ns
2 ns
2 ns
2 ns

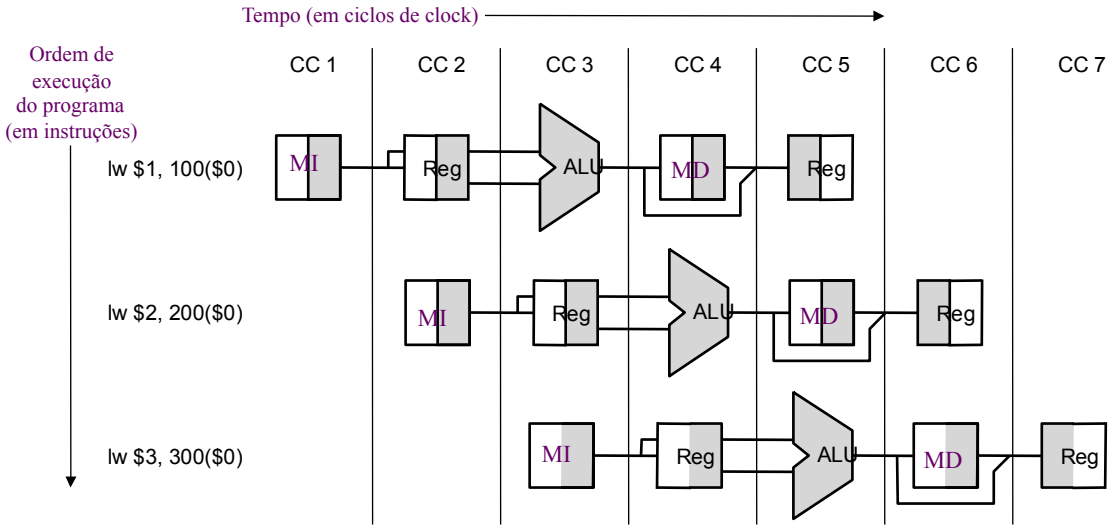
Monociclo vs Pipeline



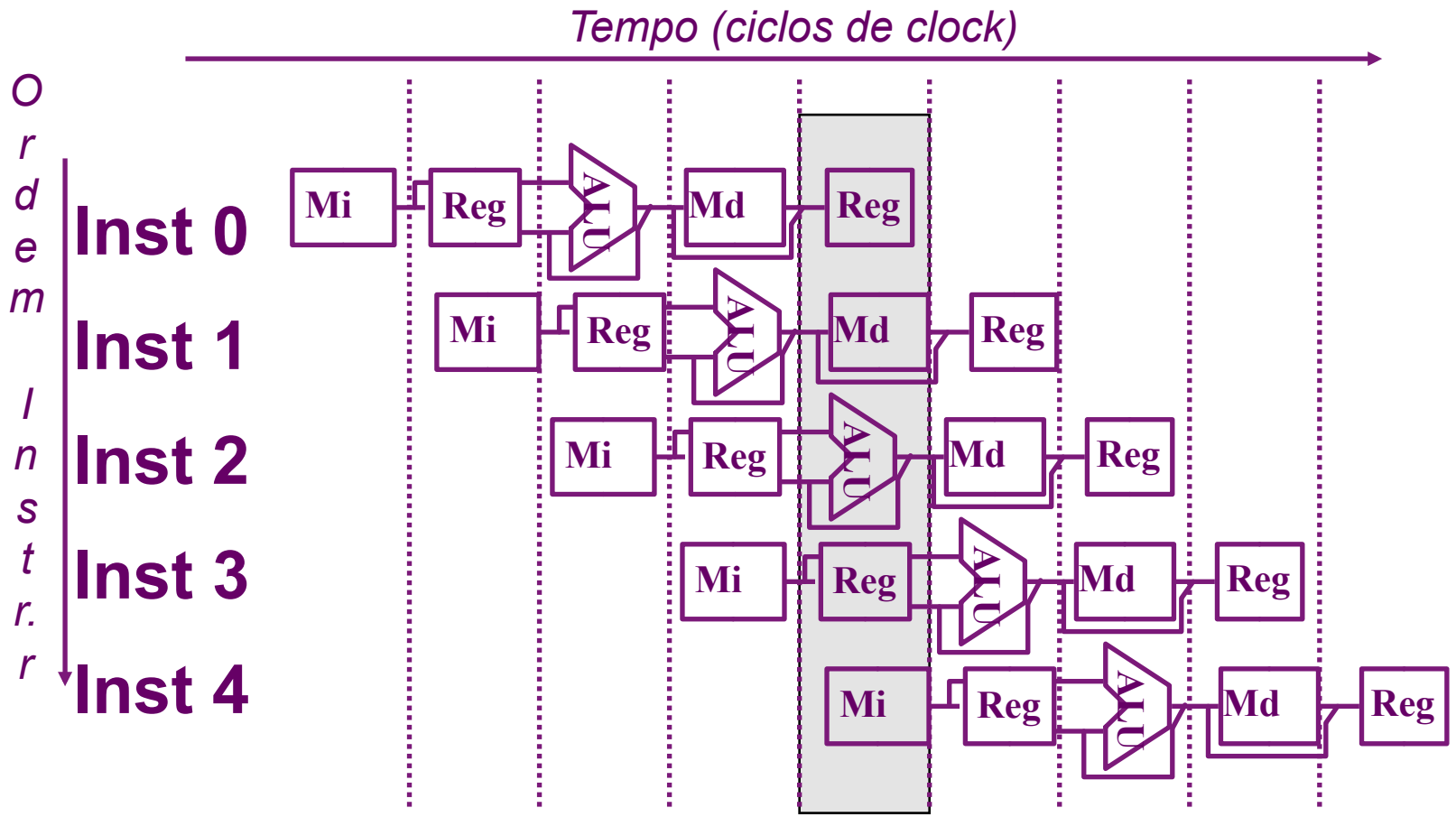
Por que usar Pipeline?

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

Representação Gráfica de Pipeline



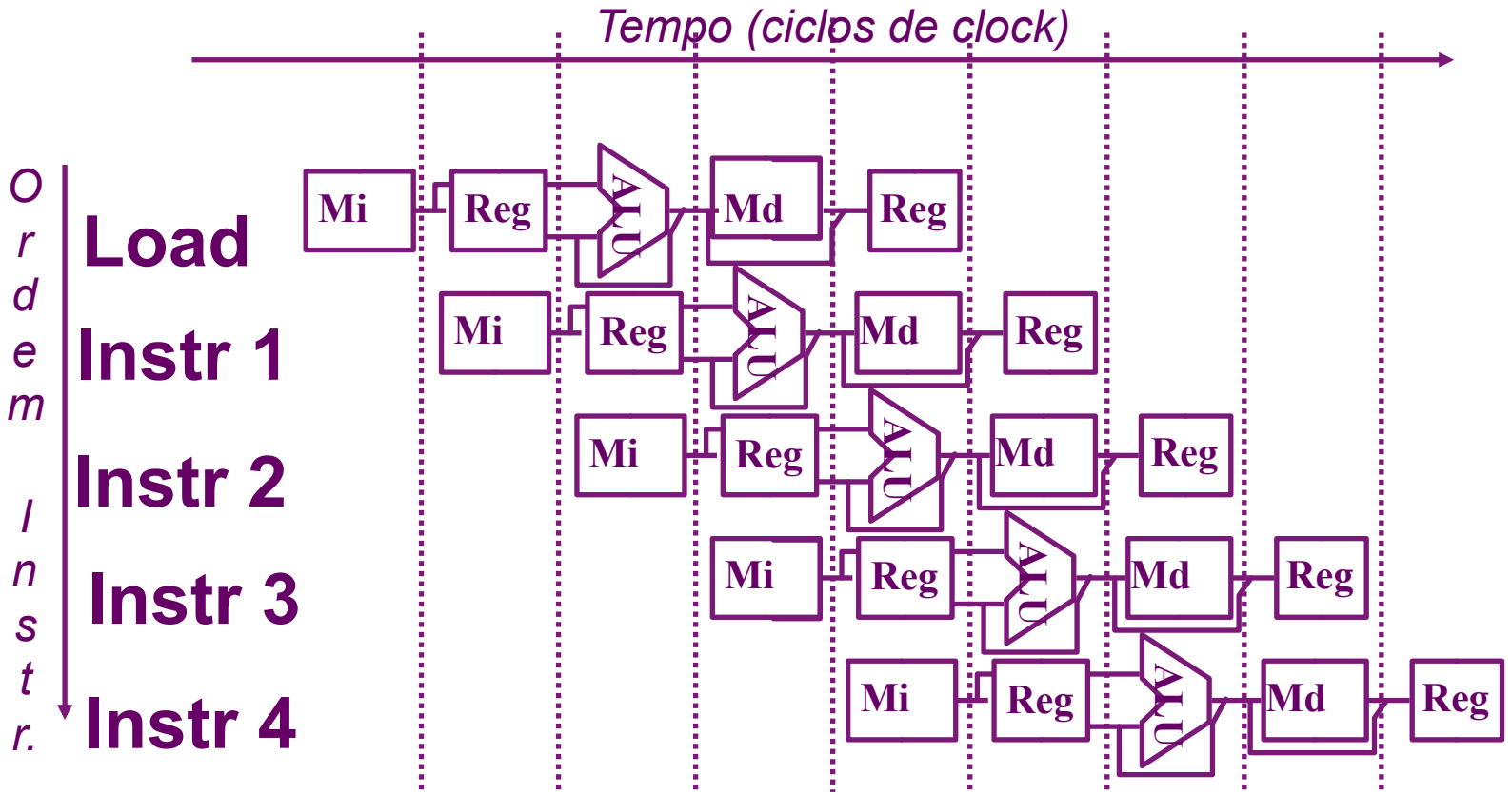
Pipeline - Recursos disponíveis



Conflitos (Hazards) do Pipeline

- ✓**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***

Conflito Estrutural - Memória Única



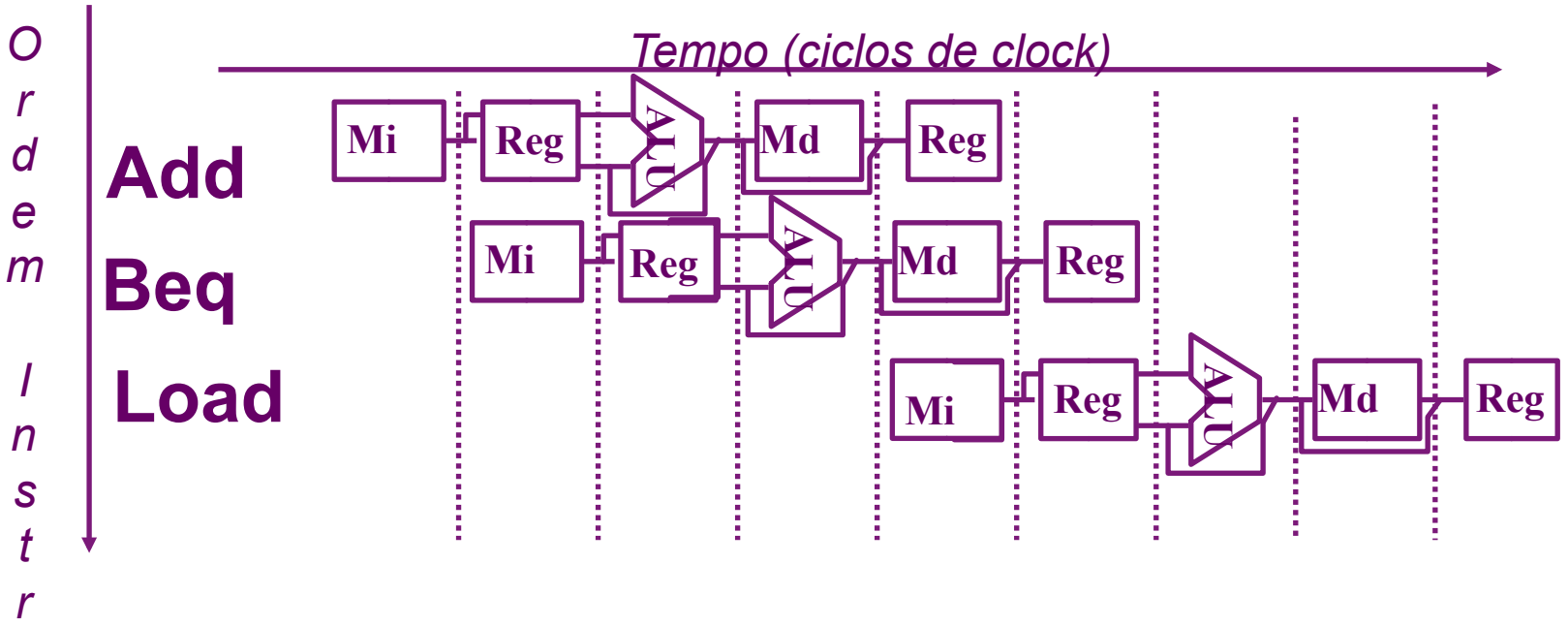
Detecção fácil nesse caso!

Conflitos do Pipeline

- **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*

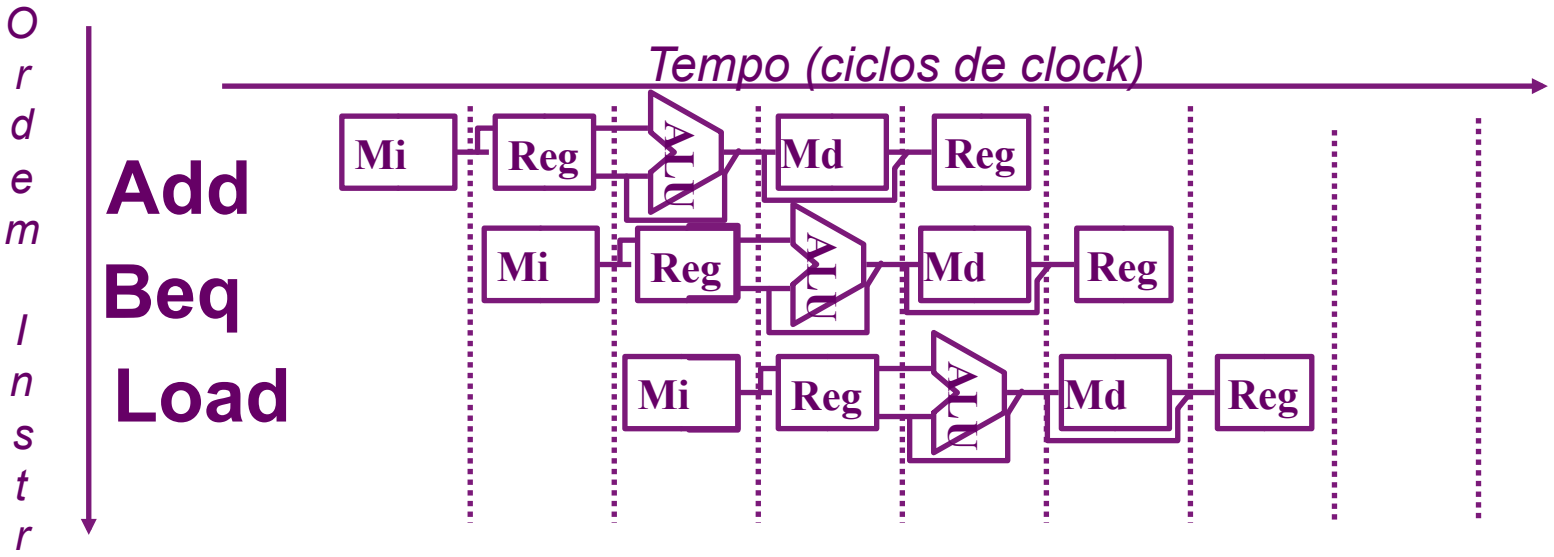
Soluções para Conflito de Controle

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



Soluções para Conflito de Controle

- ✓**Predição: escolhe uma direção e retorna se errada**
  - *predição não executada*



- ✓**Certo - 50% do tempo**

Conflitos do Pipeline

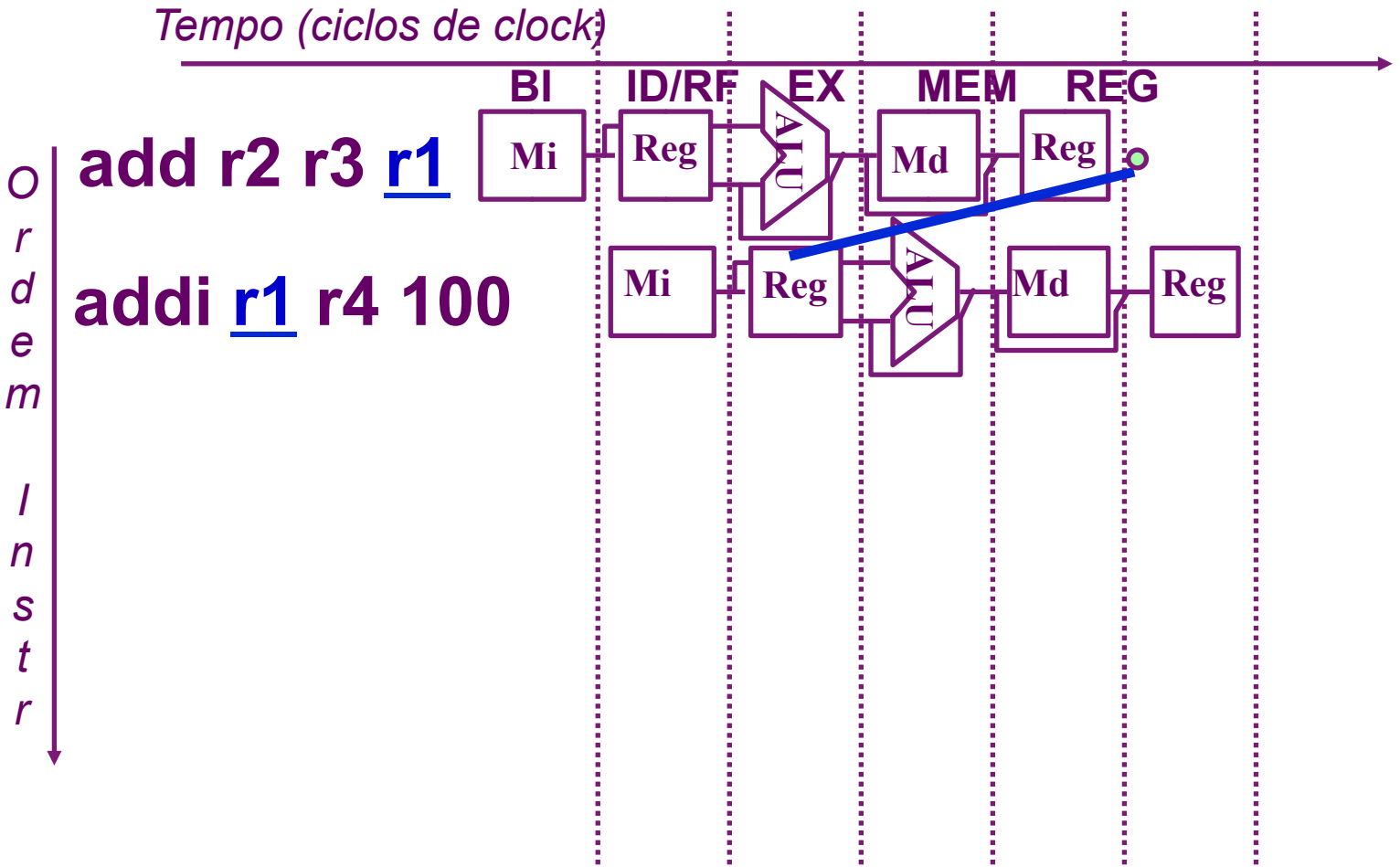
- ✓**Conflitos de dados**: **tentativa de utilizar um item antes de estar pronto**
  - *Ex.: uma meia na secadora e outra na lavadora; não pode dobrar*
  - *instrução depende de resultado da instrução anterior, ainda no pipeline*

Conflito de Dados com r1

add r2 r3 r1

addi r1 r4 100

Conflito de Dados com r1



Solução para Conflito de Dados

- **"Adiantamento" (forwarding) do resultado de um estágio para outro**