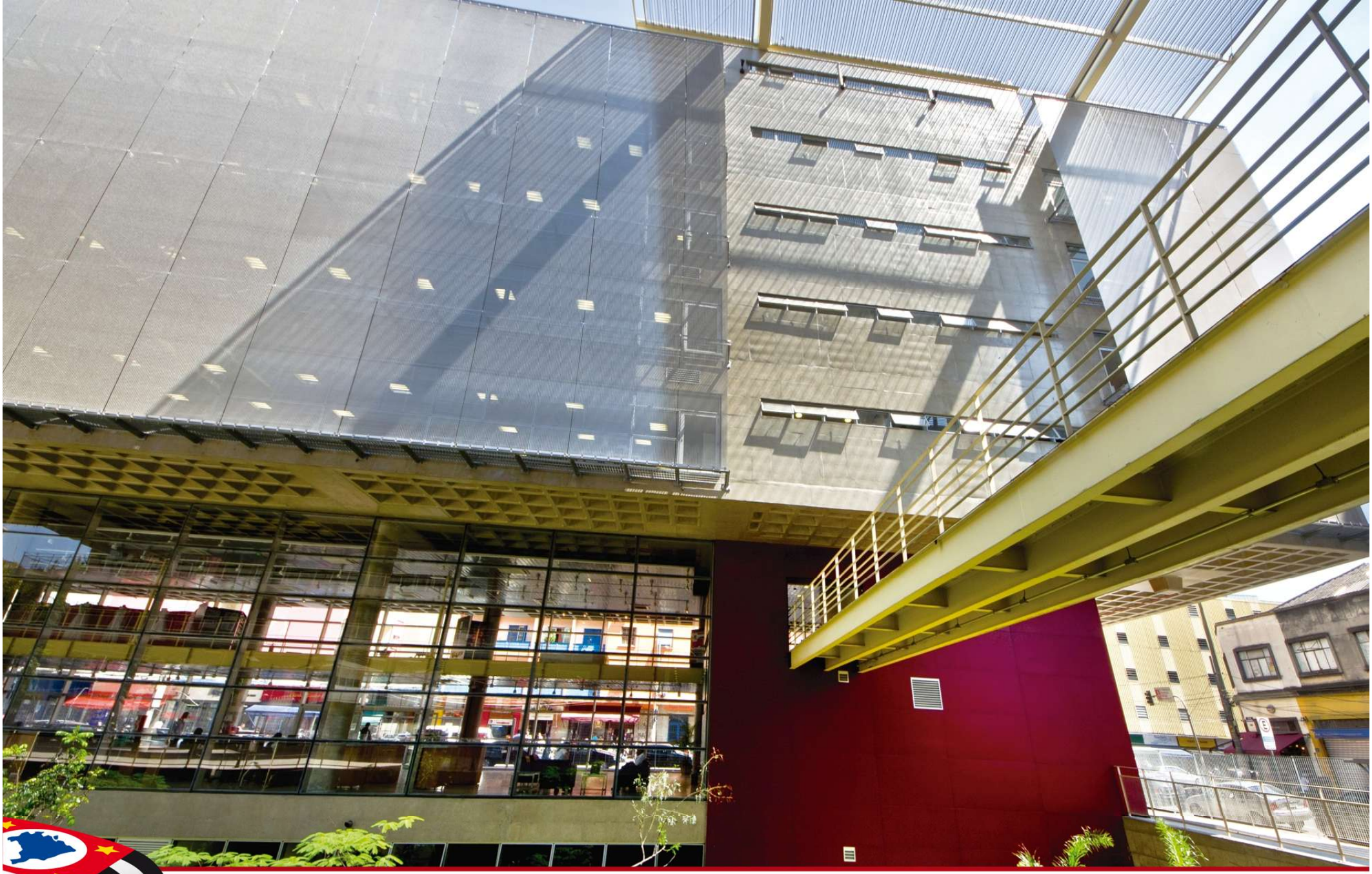


Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

Pipeline

- Pipeline:

  Técnica que permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada, que são colocadas em uma fila de memória dentro do processador onde aguardam o momento de serem executadas.

  Permite que uma instrução termine o primeiro estágio, partindo para o segundo. Nesse momento a próxima instrução já ocupa o primeiro estágio.

  Ou seja, uma instrução de processamento é subdividido em etapas, uma vez que cada uma destas etapas é executada. Isto traz um uso mais racional da capacidade computacional com ganho substancial de velocidade.

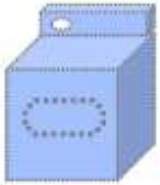
- Pipeline é:
  - Esta técnica é utilizada para acelerar a velocidade de operação da CPU, uma vez que a próxima instrução a ser executada está normalmente armazenada nos registradores da CPU e não precisa ser buscada da memória principal que é muito mais lenta.

  - OBS. A melhora de desempenho não é devido a frequência de clock, e sim pela racionalização do uso do processador

Pipeline

4 pessoas (A, B, C, D) possuem sacolas de roupa para lavar, secar e dobrar

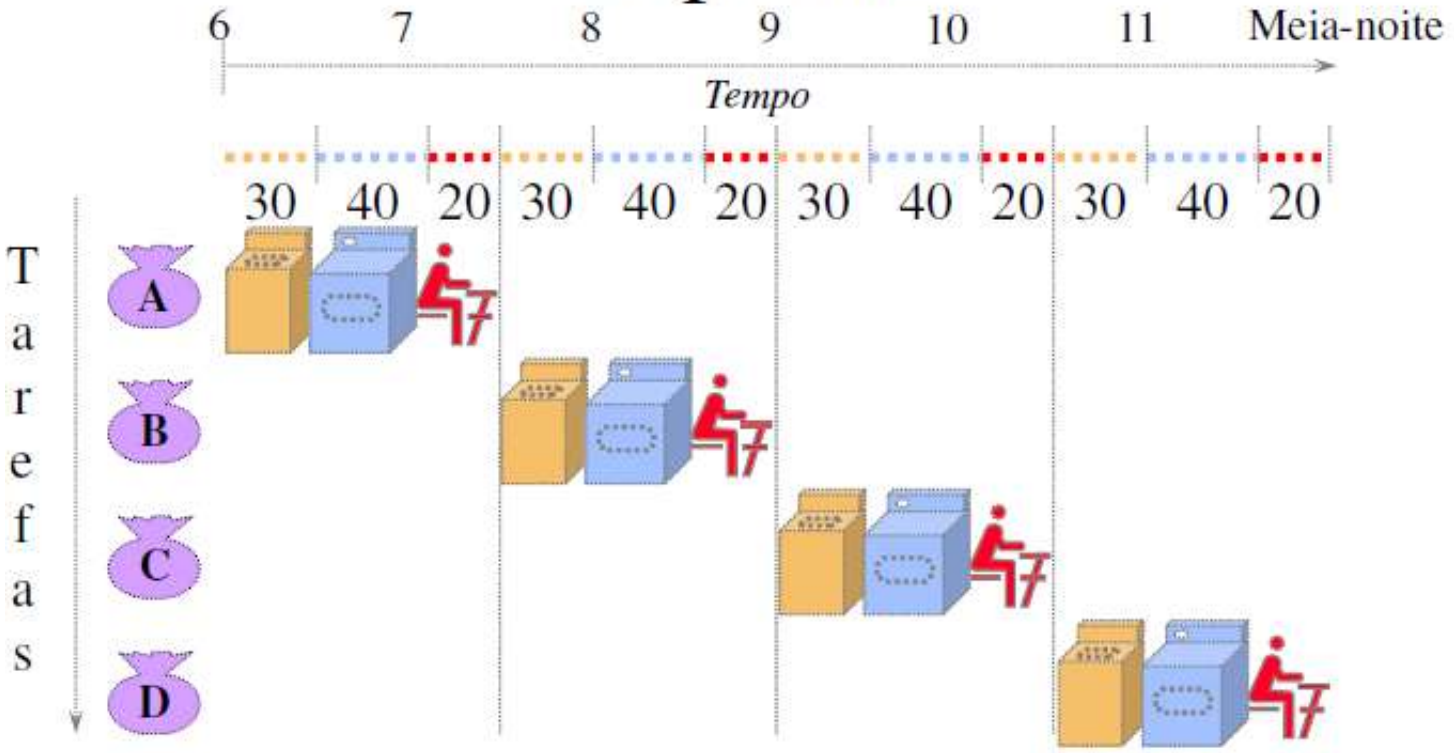
Lavar leva 30 minutos



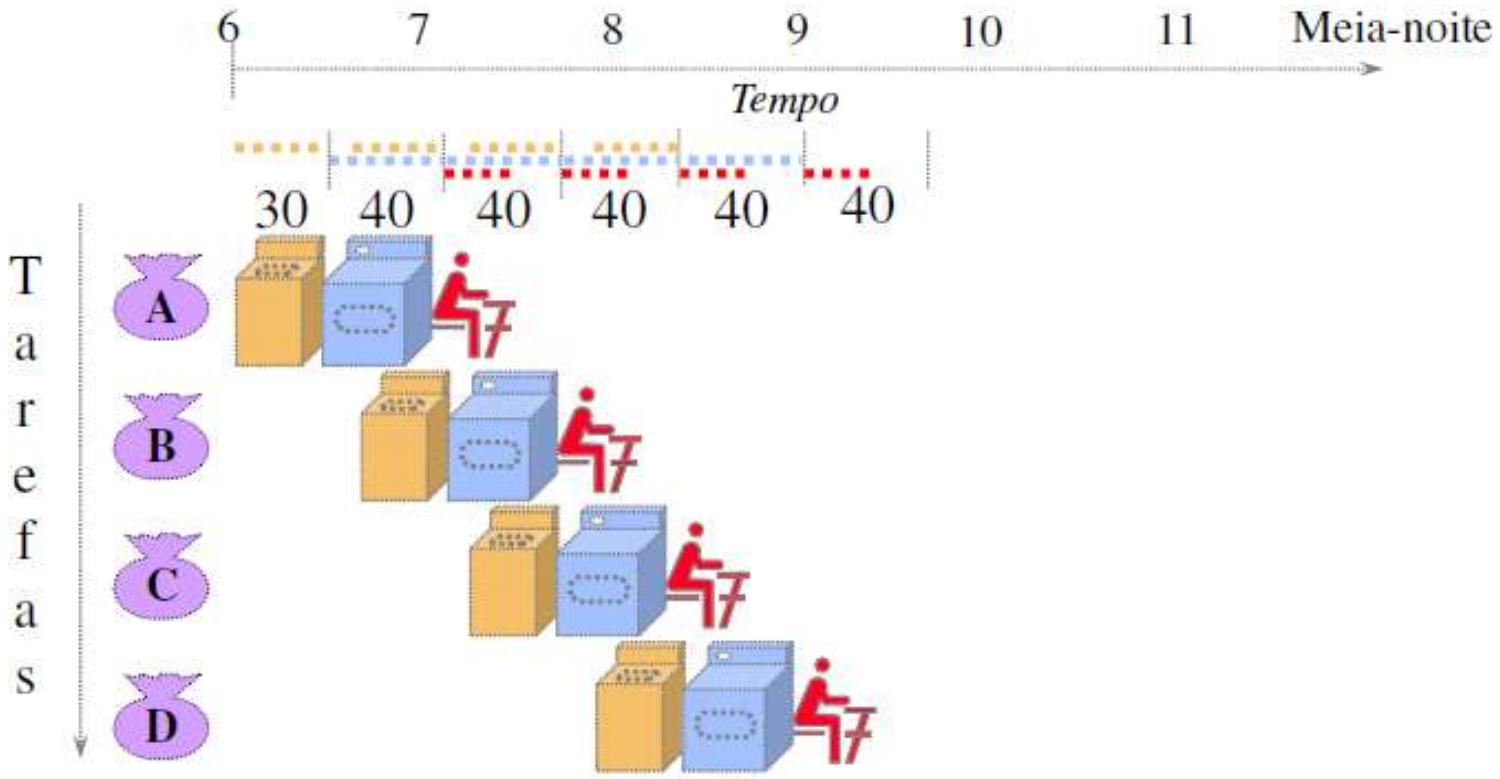
Secar leva 40 minutos

Dobrar leva 20 minutos

Pipeline



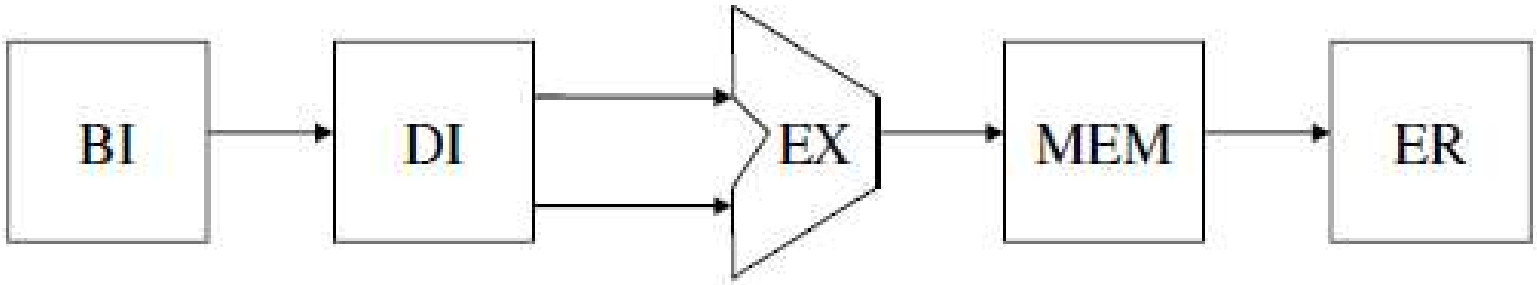
- Lavanderia seqüencial leva 6 horas para terminar
- Se eles conhecessem computação, quanto tempo levaria?



- Lavanderia com pipelining leva 3,5 horas

- Pipeline permite a sobreposição temporal de diversas fases de execução de instruções, ou seja, o hardware processa mais de uma instrução de cada vez, sem esperar que uma instrução termine antes de começar a outra.
- Pipeline não melhora a latência de uma única tarefa, mas melhora o throughput de todo trabalho.
- Tempo de execução de uma tarefa é o mesmo, com ou sem pipelining.
- Ganho começa a existir a partir da segunda tarefa.

- Tradicionalmente, as instruções do MIPS (Million Instructions per Second) são executadas em 5 etapas:
  - Busca da instrução na memória
  - Leitura dos registradores
  - Execução de uma operação / cálculo de um endereço
  - Acesso a um operando na memória
  - Escrita do resultado em um registrador



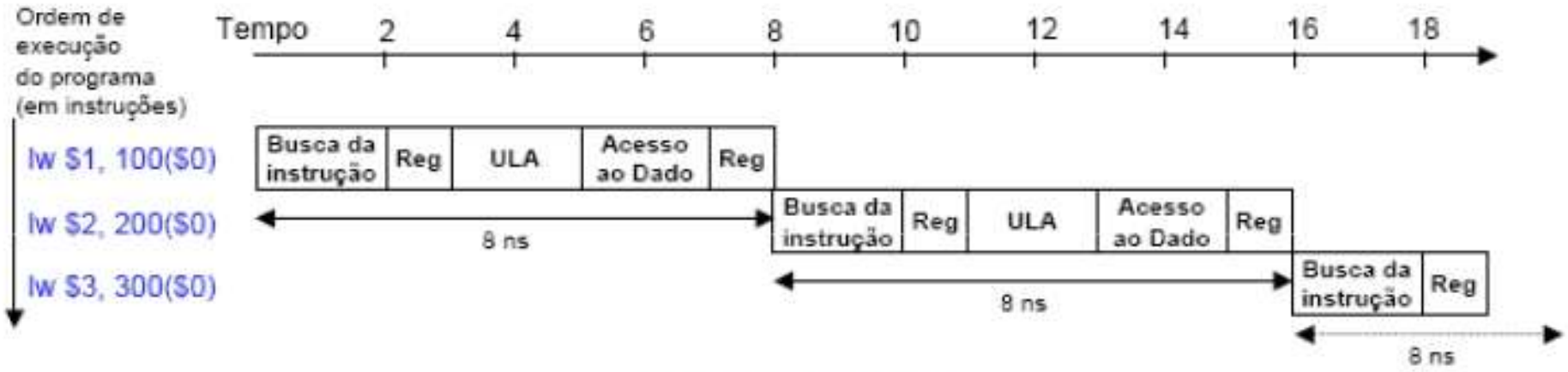
- BI: busca de instrução (memória de instruções)
- DI: decodificação da instrução e leitura do banco de registradores (banco de registradores sendo lido)
- EX: estágio de execução da instrução (ULA)
- MEM: acesso à memória (memória de dados)
- ER: escrita do resultado no banco de registradores (banco de registradores sendo escrito)

Uso a partir do 486

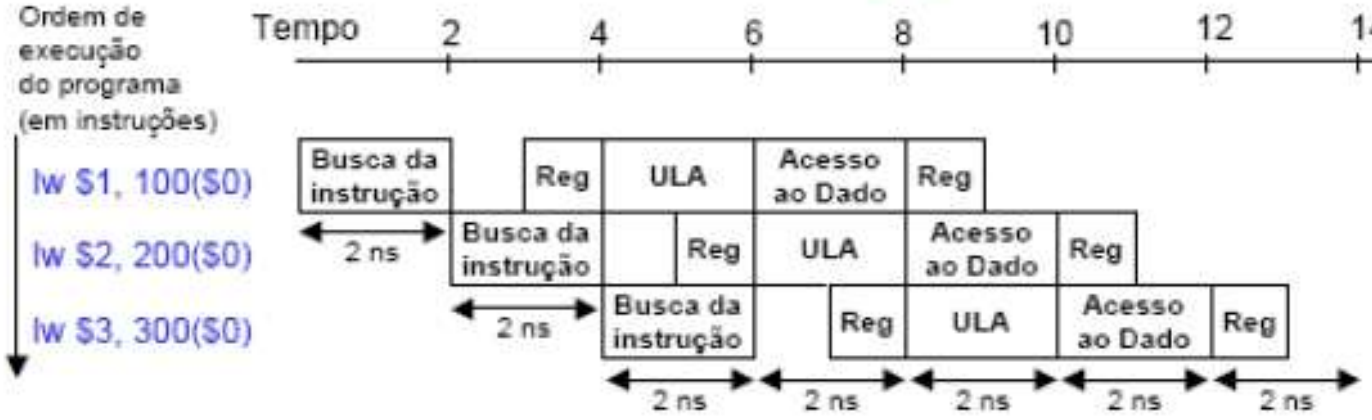
- Dois blocos de memória (instrução e dados) são necessários
- Nem todos os estágios executam com o mesmo tempo
- A execução precisa ser alinhada em tempo – os estágios mais rápidos esperam

Pipeline MIPS - Desempenho

sem pipeline



com pipeline

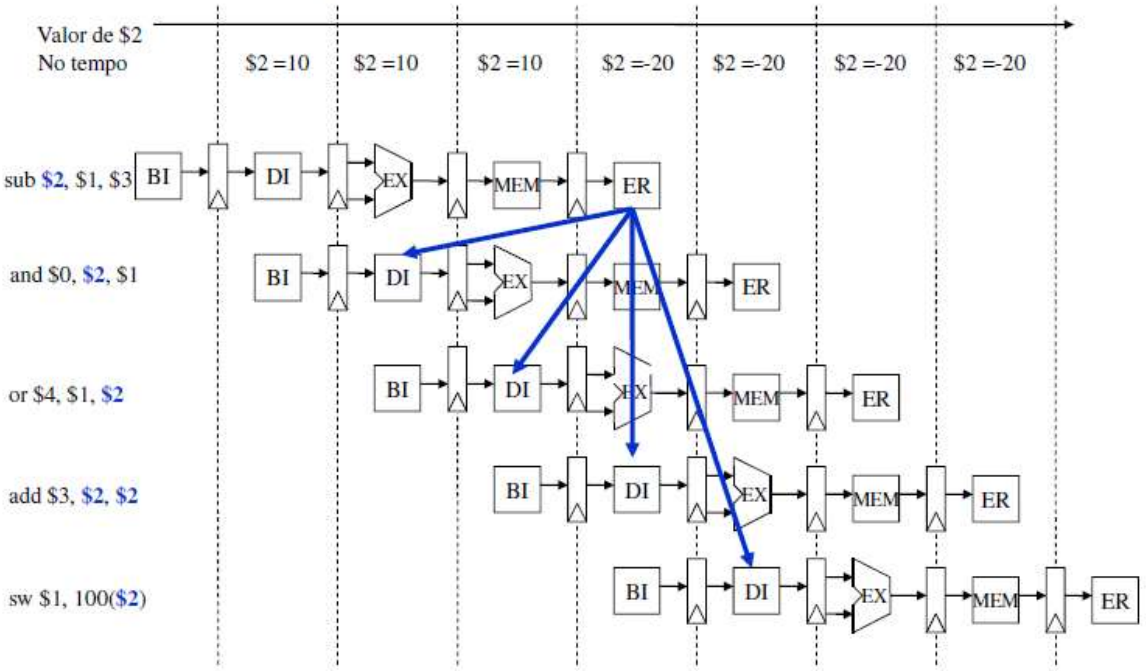


- Conflitos
  - “Situações de execução no pipeline em que a instrução seguinte não pode ser executada no próximo ciclo de relógio”

- Tipos de Conflitos:
  - Estruturais
    - O hardware não suporta uma combinação de instruções executadas no pileline
  - Dependência de instruções anteriores e desvios que dificultam o processo, bem como a diferença de complexidade de instruções que fazem com que as mesmas possam levar um tempo variável para execução.

- Conflitos de dados e de controle
  - A execução de uma instrução depende de um dado ainda não disponível
  - Originam-se na necessidade de se tomar uma decisão baseada nos resultados de uma instrução, a qual ainda não foi concluída.
  - Muito comuns em instruções de salto condicional (*beq*).
  - Uma possível solução é a parada (também chamada de “bolha”), ou seja, interromper a progressão das instruções pelo pipeline.

Dependências de dados



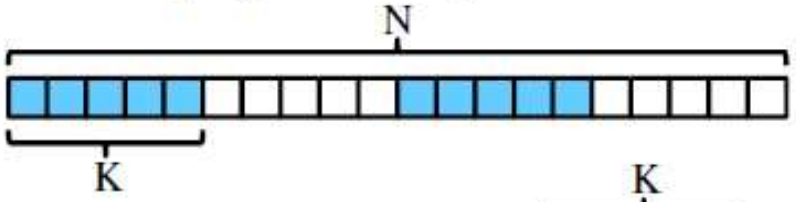
- Pipeline explora o paralelismo entre as instruções em um fluxo de instruções sequenciais.
- É uma técnica invisível ao programador

Supondo:

- N instruções
- K estágios de duração T

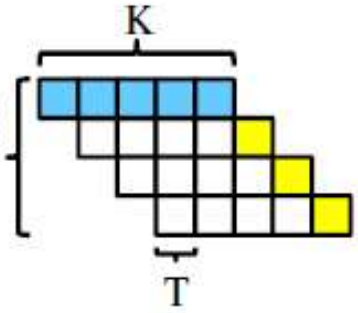
Tempo de execução sem pipelining:

$N \times K \times T$



Tempo de execução com pipelining:

$K \times T + (N-1) \times T$



$F = 1/T$

$T=1/F$

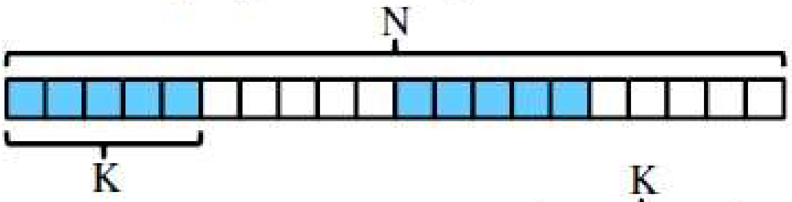
Calculando...

Supondo:

- N instruções
- K estágios de duração T

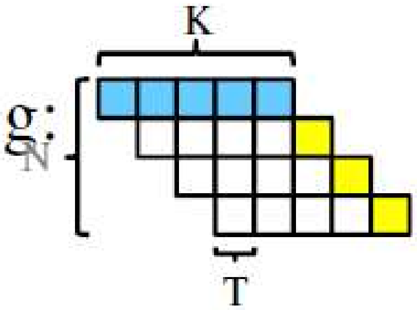
Tempo de execução sem pipelining:

$$N \times K \times T$$



Tempo de execução com pipelining:

$$K \times T + (N-1) \times T$$



Calculando...

Um programa tem 100 instruções. Em uma arquitetura sem pipeline, o tempo de execução de cada etapa da instrução é de 2ns. Qual o ganho na execução deste programa em um processador com pipeline de 5 estágios?

Sem Pipeline: 100*5*2=1000NS

Com Pipeline: 5*2+(99)*2=208NS

Ganho de: Sem Pipeline/Com Pipeline = 4,8076923076923076923076923 ou 4,8

O ganho de desempenho de uma CPU com uma pipeline decorre do aumento no fluxo de execução de instruções. O ganho teórico nunca é obtido na prática. Isto se deve à ocorrência de conflitos que provocarão a parada da pipeline

Prefix	Analoge waarde
p (pico)	10^{-12}
n (nano)	10^{-9}
μ (micro)	10^{-6}
m (milli)	10^{-3}
k (kilo)	10^{3} (1000)
M (mega)	10^{6} (1,000,000)
G (Giga)	10^{9} (1,000,000,000)
T (Tera)	10^{12} (1,000,000,000,000)

Fonte: http://www.telecomabc.nl/p/prefix.html

- **Dado um tempo de execução de ciclos: 11ns, qual a frequência de clock?**

$$F: \frac{1}{11ns} \Rightarrow \frac{1}{11.10^{-9}} \Rightarrow 0{,}09090. \frac{1}{10^{-9}} \Rightarrow 0{,}09090.10^{9} \Rightarrow 90{,}9.10^{6} \Rightarrow 90{,}9MHZ$$

- **Dado uma frequência de clock de 90Mhz, qual é o tempo de execução de ciclos?**

$$T: \frac{1}{90MHZ} \Rightarrow \frac{1}{90.10^{6}} \Rightarrow 0{,}01111. \frac{1}{10^{6}} \Rightarrow 0{,}01111.10^{-6} \Rightarrow 11{,}11.10^{-9} \Rightarrow 11{,}11ns$$

Exercícios

- Um programa tem 1.000.000 de instruções. Em uma arquitetura sem pipeline, o tempo médio de execução de cada instrução é 6,5ns. Qual o ganho na execução deste programa em um processador com pipeline de 5 estágios com ciclo de 2 ns?

- Um programa tem 2.500.000 de instruções. Em uma arquitetura sem pipeline, o tempo médio de execução de cada instrução é 6ns. Qual o ganho na execução deste programa em um processador com pipeline de 5 estágios com ciclo de 2 ns?

- Um programa tem 1.000.000 de instruções. Em uma arquitetura sem pipeline, o tempo médio de execução de cada instrução é 6,5ns. Qual o ganho na execução deste programa em um processador com pipeline de 5 estágios com ciclo de 2 ns?
  - ✓ 1000000x5x2 = 10.000.000
  - ✓ 10+999999x2 = 2.000.008
  - ✓ Ganho é igual a 10.000.000/2.000.008=>4,99998

- Um programa tem 2.500.000 de instruções. Em uma arquitetura sem pipeline, o tempo médio de execução de cada instrução é 6ns. Qual o ganho na execução deste programa em um processador com pipeline de 5 estágios com ciclo de 2 ns?
  - ✓ 2.500.000x5x2 = 25.000.000
  - ✓ 10+4.999.998 = 5.000.008
  - ✓ Ganho é igual a 25.000.000/ 5.000.008 => 4,999992

Exercícios

- **Dado um tempo de execução de ciclos de 8 ns, qual a frequência de clock?**

- **Dado um tempo de execução de ciclos de 13 ns, qual a frequência de clock?**

- **Dado uma frequência de clock de 120Mhz, qual é o tempo de execução de cada ciclo?**

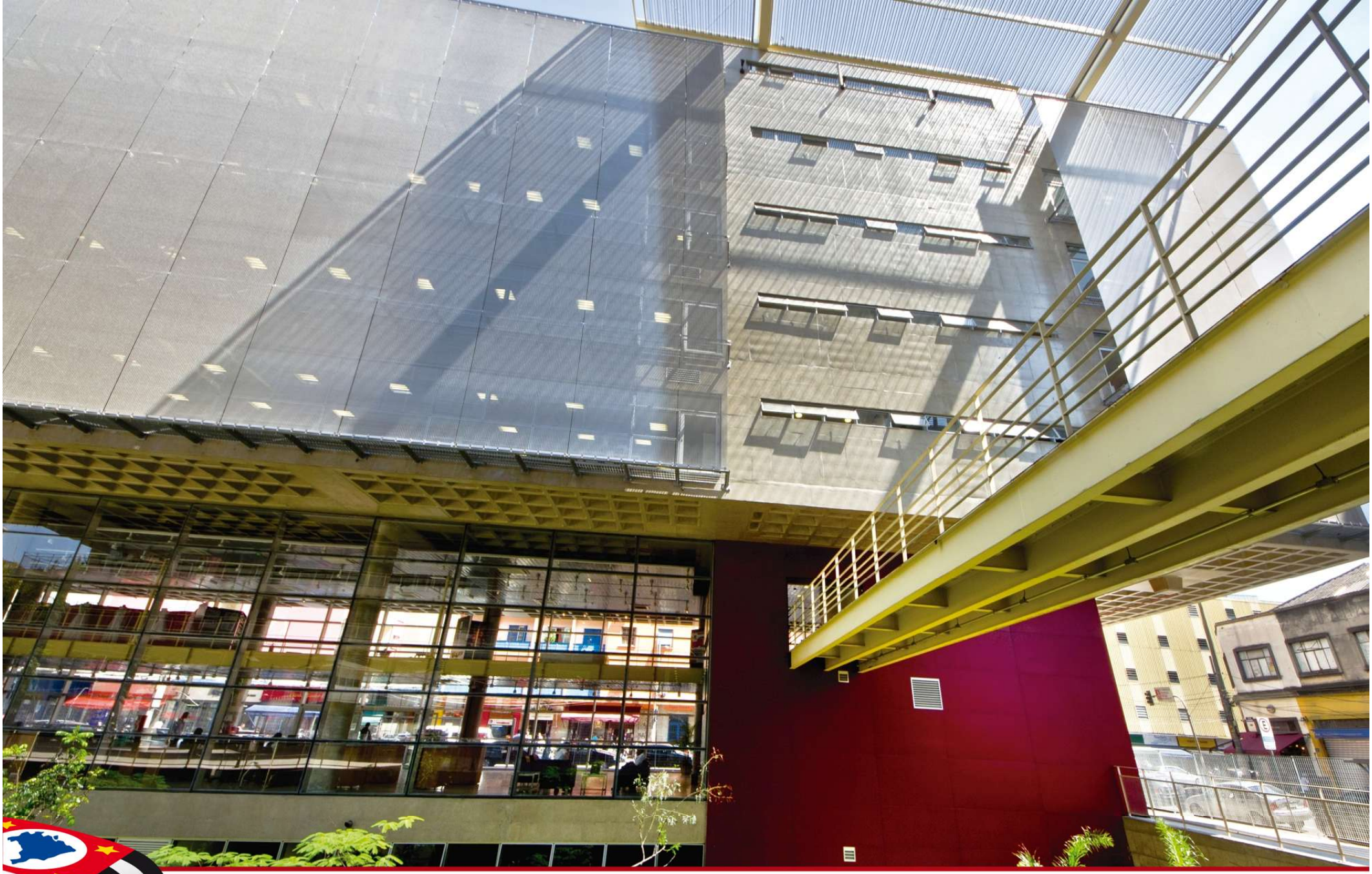
- **Dado uma frequência de clock de 70Mhz, qual é o tempo de execução de cada ciclo?**

- **Dado um tempo de execução de ciclos de 8 ns, qual a frequência de clock?**
  - **125Mhz**

- **Dado um tempo de execução de ciclos de 13 ns, qual a frequência de clock?**
  - **76,92Mhz – 77Mhz**

- **Dado uma frequência de clock de 120Mhz, qual é o tempo de execução de cada ciclo?**
  - **8,33ns – 8ns**

- **Dado uma frequência de clock de 70Mhz, qual é o tempo de execução de cada ciclo?**
  - **14,28ns – 14ns**



Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO