

**UNISINOS – Universidade do Vale do Rio dos Sinos**

**Organização e Arquitetura de Computadores**

# **Arquiteturas de Ciclo**

**Professor:** Rodrigo Marques de Figueiredo

# Tipos de Ciclos

Sequencial



Paralelo

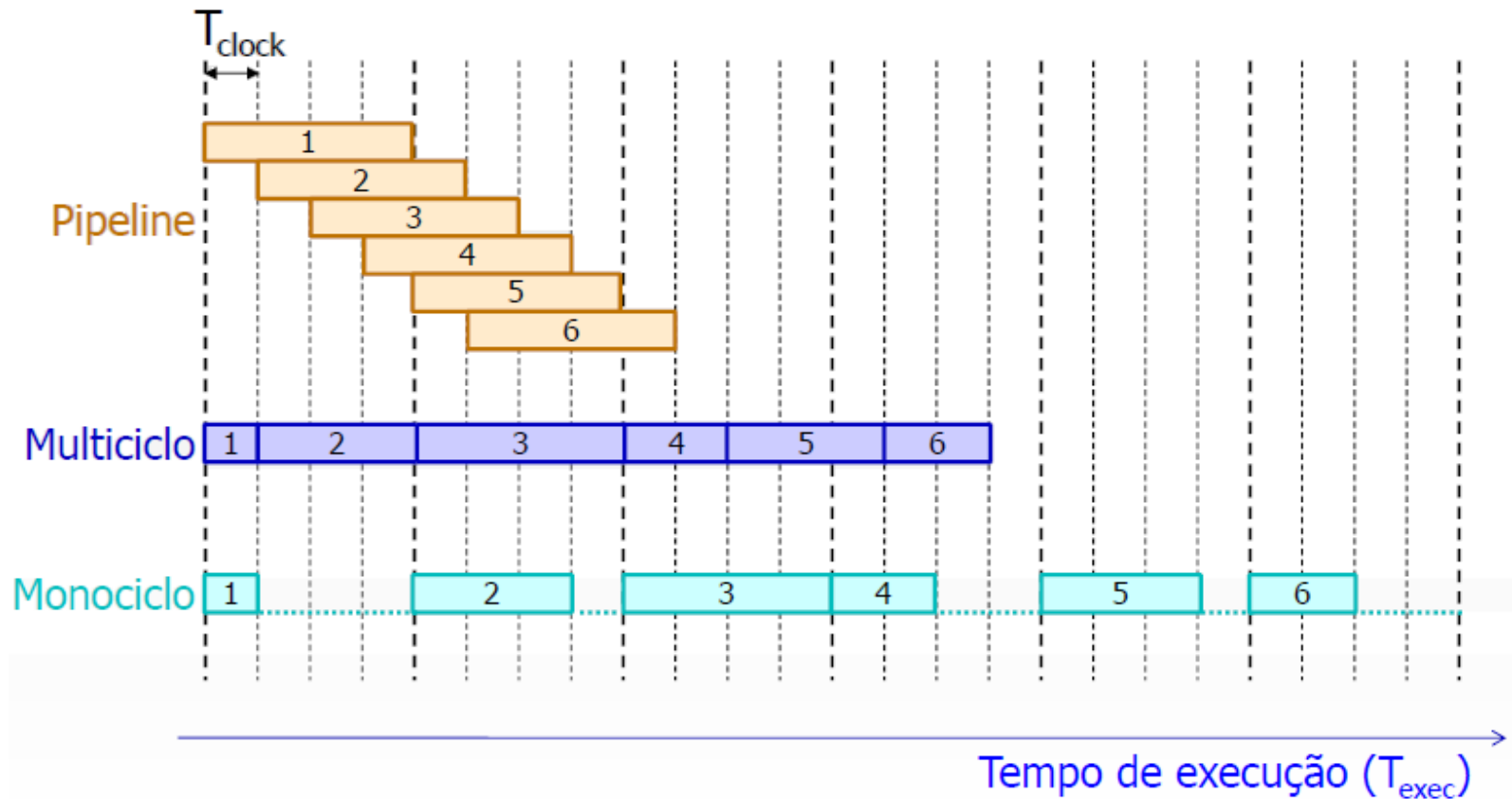


*Pipeline*



# Tipos de Ciclo

- Arquiteturas *Pipeline*, Multiciclo e Monociclo:

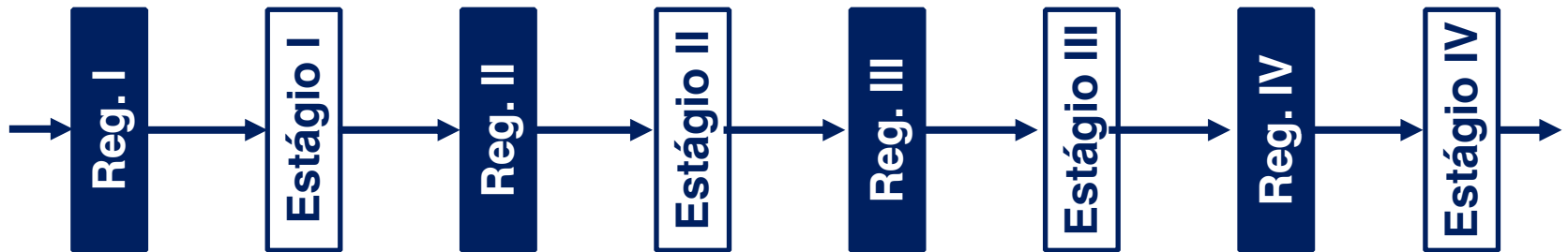


# Pipelining

- É uma técnica computacional que busca minimizar o tempo de processamento de um dado;
- Se vale do uso da sobreposição de execução de processos de diferentes instruções;
- Tira proveito do uso do paralelismo entre as ações necessárias para executar um processamento;
- Normalmente aplicamos esta técnica de maneira intuitiva no nosso dia a dia.
- **A premissa do *pipeline* é a vazão de dados processados.**
- Como em uma linha de montagem industrial, o *pipelining*, baseia-se em pulmões (registradores) e estações de processamento (estágios);

# Pipelining

- Em um *pipeline* ideal temos:



- Todos elementos (dados) são processados pelos mesmos estágios;
- Não há recursos entre dois processos;
- O atraso de propagação é igual para todos os estágios;
- O agendamento (*scheduling*) na entrada do *pipeline* não é afetado por transições (processamento) nos demais estágios;

Normalmente em uma linha de produção industrial esta última condição não é verdadeira.

# *Pipelining*

Um *pipeline* computacional tem como objetivo diminuir o tempo de resposta entre a entrada do dado no *processador* e sua entrega na saída. Ou seja, um *pipeline* computacional permite a maior vazão para o sistema.

# Pipelining

- Exemplo (TORRES, 2006):

Quatro usuários estão em uma lavanderia para lavar, secar e passar suas roupas:



Lavar leva **30 minutos**



Secar leva **40 minutos**



Passar leva **20 minutos**

# Pipelining

## Abordagem Sequencial





# Pipelining

## Abordagem Pipeline



# *Pipelining*

- O *pipeline* **não melhora a latência** de tarefa única;
- **Melhora o *throughput*** de todo trabalho;
- O tempo de execução de uma tarefa é o mesmo, com ou sem o *pipelining*;
- O ganho dado pelo *pipelining* começa a ser dado a partir da segunda tarefa entrar no sistema;
- A taxa de *input* de tarefas é dada pela tarefa mais lenta (Teoria das Restrições);
- É uma técnica *invisível* ao programador (usuário) ao contrário das técnicas de computação paralela.

*Pipelining* dá-se no nível da Engenharia e *Clustering* nos níveis de Engenharia e Programação.