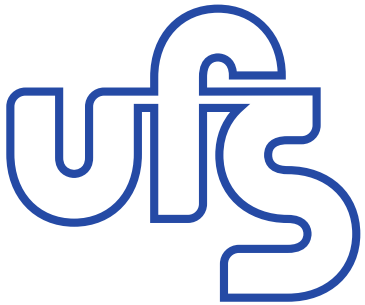
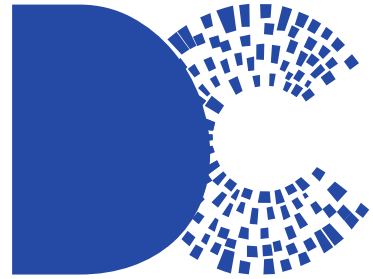




Multiciclo e *pipeline*

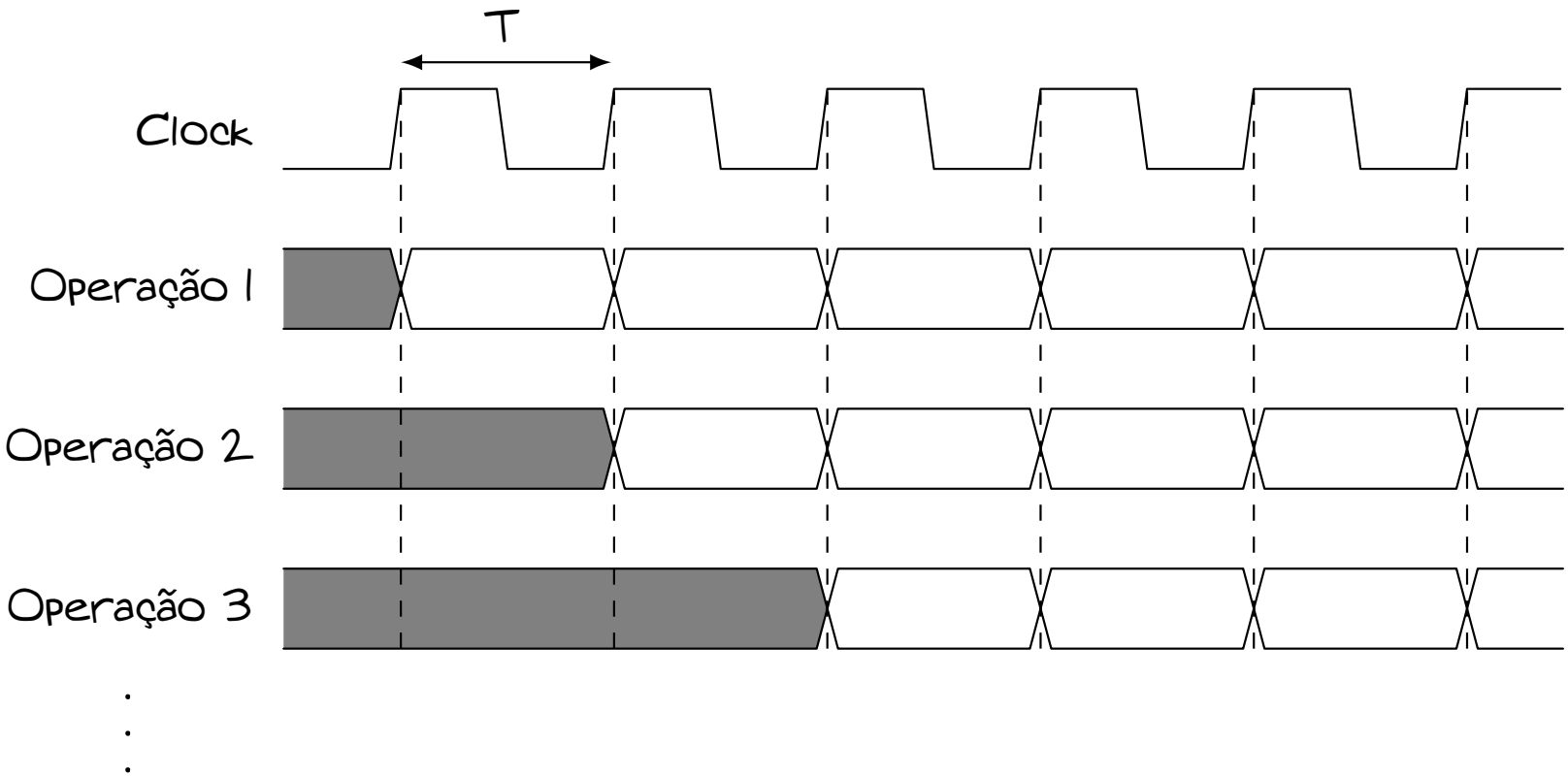
Arquitetura de Computadores

Bruno Prado

Departamento de Computação / UFS

Introdução

- A grande maioria dos computadores são síncronos
  - Um oscilador de cristal de quartzo gera uma frequência para operação de todo o circuito
  - Todas as operações ocorrem em ciclos de relógio



$$f = 1 / T$$

Introdução

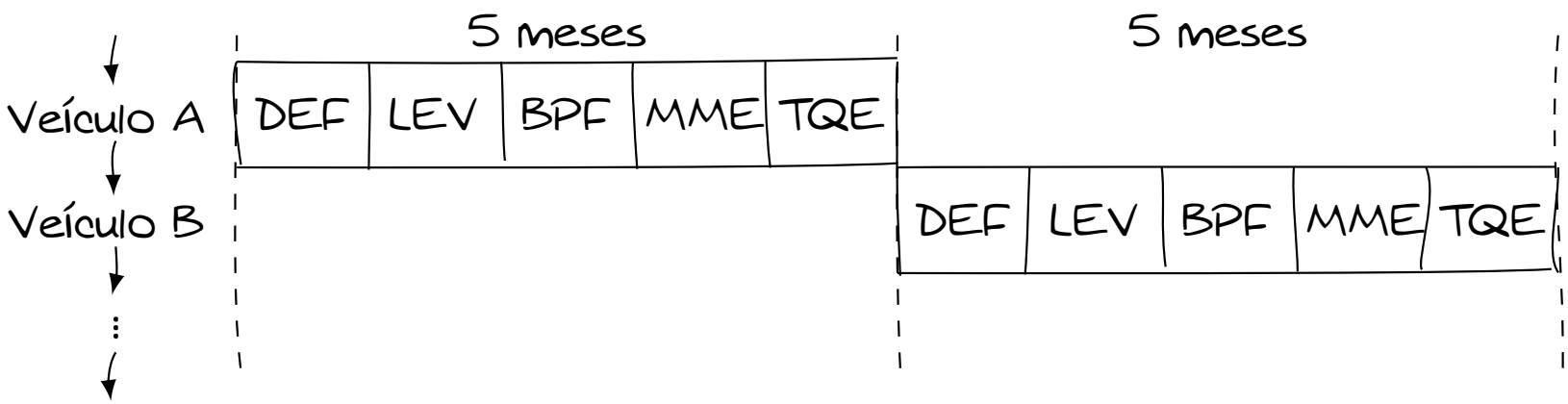
- Ciclo de execução de um processador
  1. Cálculo do PC para próxima instrução (BPI)
  2. Decodificação da instrução (DEC)
  3. Busca dos operandos (BOP)
  4. Execução da instrução (EXE)
  5. Acesso à memória (MEM)

Introdução

- Analogia com produção de veículos personalizados
  1. Definição do modelo que será fabricado (DEF)
  2. Leitura das especificações do pedido do veículo (LEV)
  3. Busca de peças e ferramentas necessárias (BPF)
  4. Montagem das partes mecânicas e elétricas (MME)
  5. Teste de qualidade e entrega do veículo (TQE)

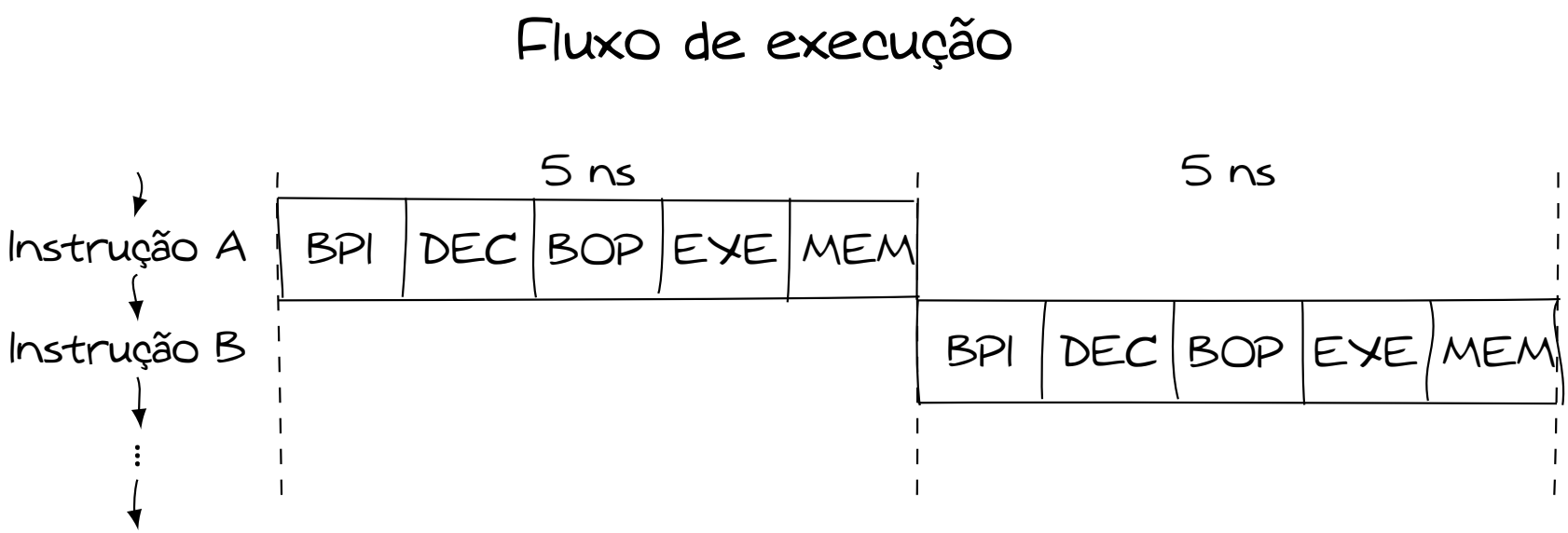
Introdução

- Analogia com produção de veículos personalizados
  - A produção em série de cada veículo na linha de montagem utiliza todos os recursos da fábrica



Introdução

- Execução em ciclo único
  - Todas as instruções do processador são executadas sequencialmente em um ciclo único de relógio



Introdução

- Execução em ciclo único
  - Devido ao comportamento sequencial, cada etapa do ciclo de execução depende da etapa anterior

Introdução

- Execução em ciclo único
  - Devido ao comportamento sequencial, cada etapa do ciclo de execução depende da etapa anterior
  - Como apenas uma etapa é executada por vez, em 80% do tempo os recursos de hardware ficam ociosos

Introdução

- Projetos multiciclo e *pipeline*
  - As instruções são implementadas em múltiplos ciclos de relógio para reaproveitamento dos componentes de hardware entre as etapas de execução

Introdução

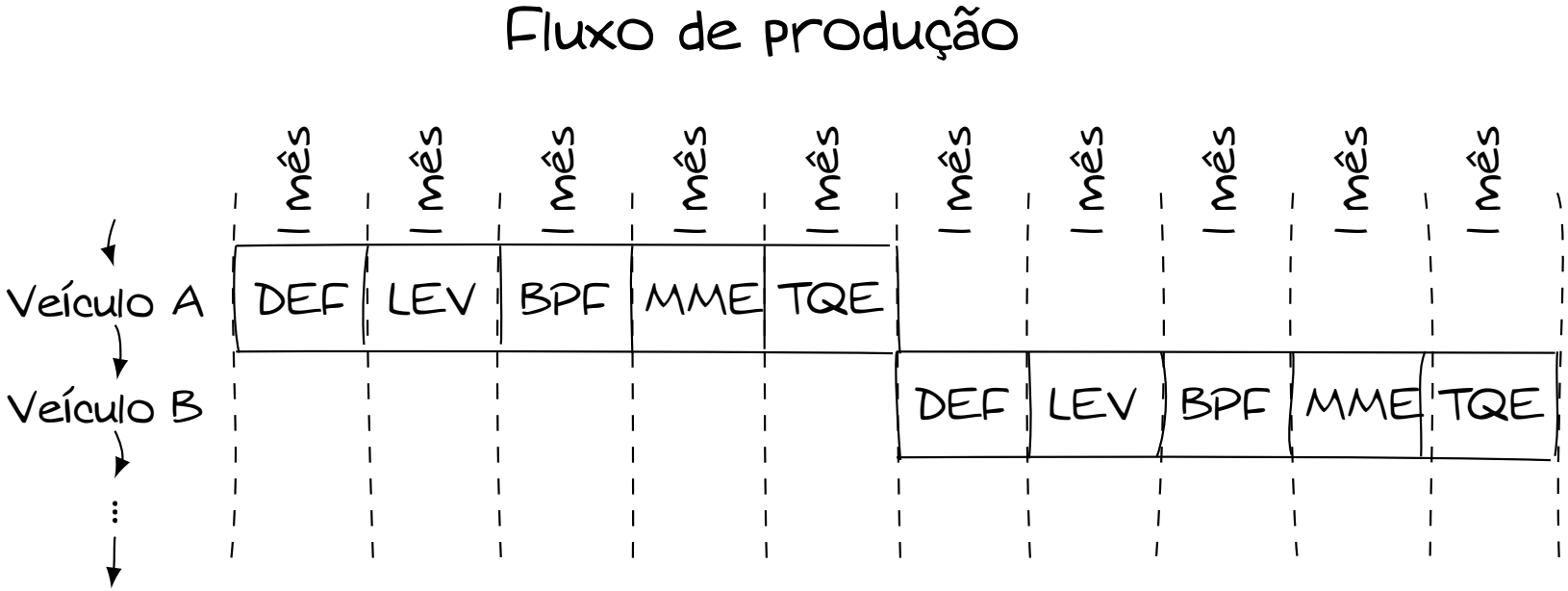
- Projetos multiciclo e *pipeline*
  - As instruções são implementadas em múltiplos ciclos de relógio para reaproveitamento dos componentes de hardware entre as etapas de execução
  - **Multiciclo**: compartilhamento dos recursos de hardware em cada ciclo (otimização do hardware)

Introdução

- Projetos multiciclo e *pipeline*
  - As instruções são implementadas em múltiplos ciclos de relógio para reaproveitamento dos componentes de hardware entre as etapas de execução
  - **Multiciclo**: compartilhamento dos recursos de hardware em cada ciclo (otimização do hardware)
  - ***Pipeline***: aumento do desempenho pela execução sobreposta de instruções (taxa de execução)

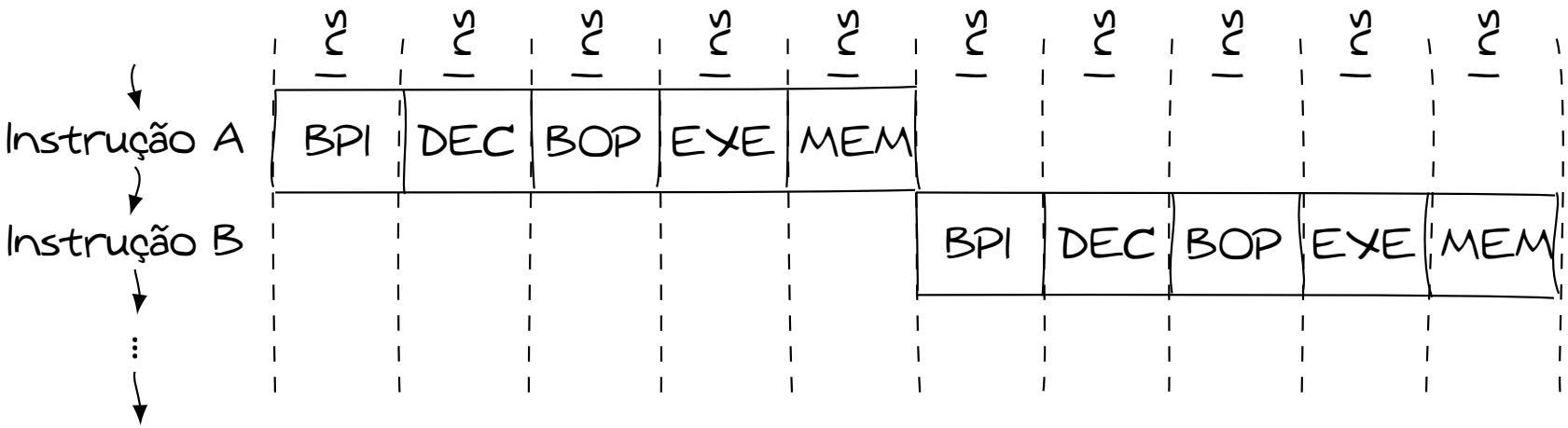
Multiciclo

- Analogia com produção de veículos personalizados
  - Cada etapa da fabricação possui um prazo determinado para ser realizada, compartilhando os recursos da fábrica, como ferramentas e funcionários



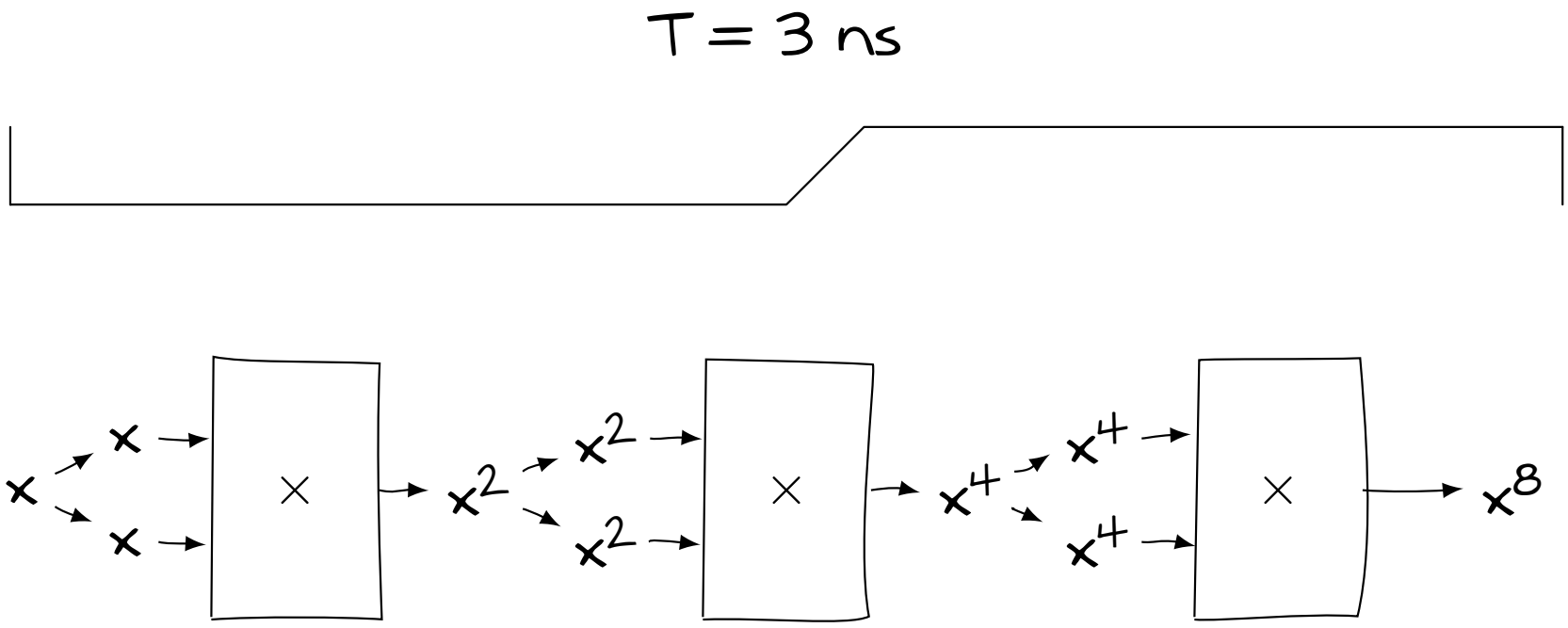
Multiciclo

- Execução em multiciclo
  - Cada etapa de execução é executada em ciclos distintos de relógio, possibilitando que os recursos de hardware sejam realocados em cada etapa



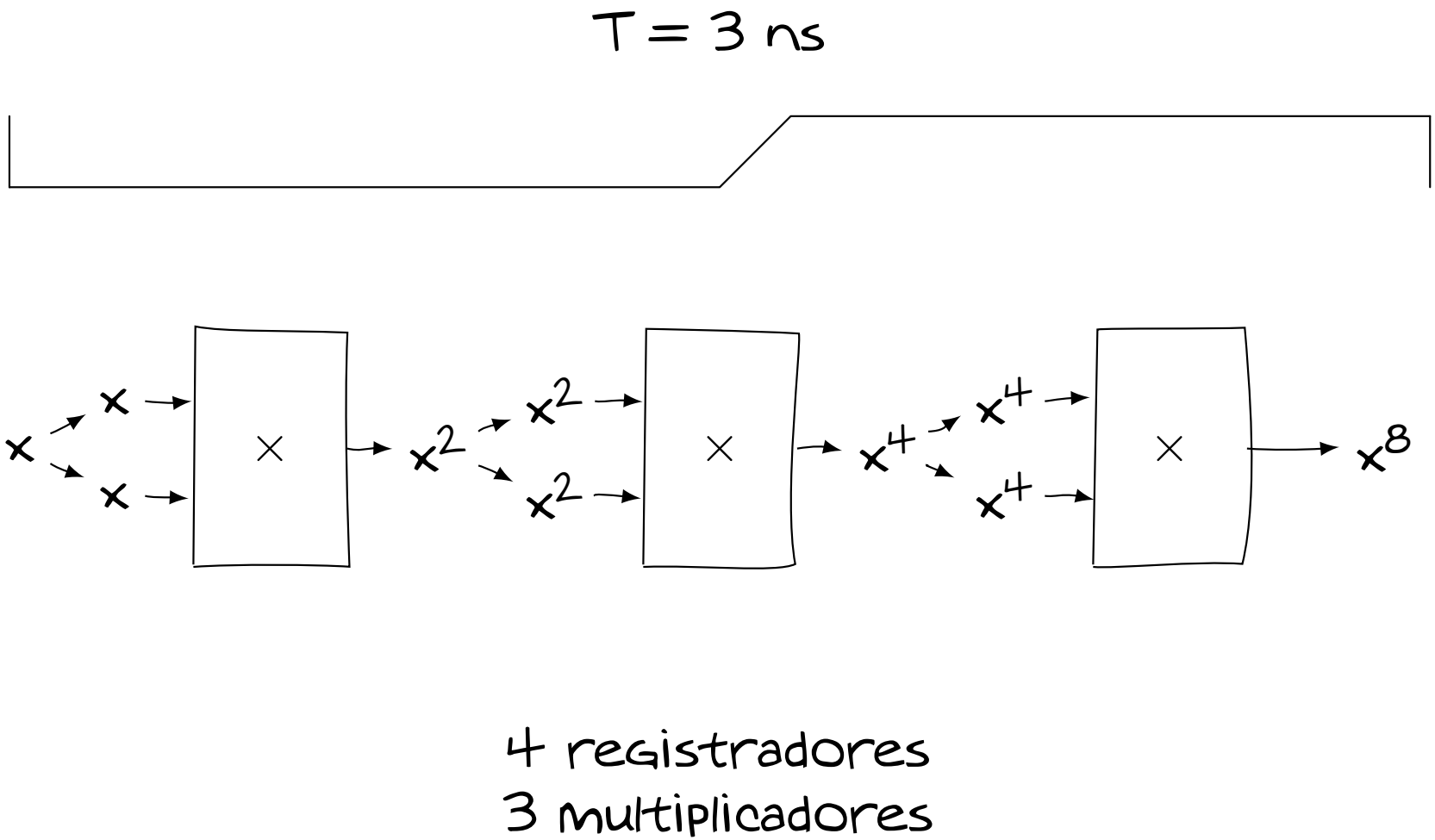
Multiciclo

- Comparativo entre ciclo único e multiciclo
  - Implementação em ciclo único de uma instrução hipotética que calcula $f(x) = x^8$



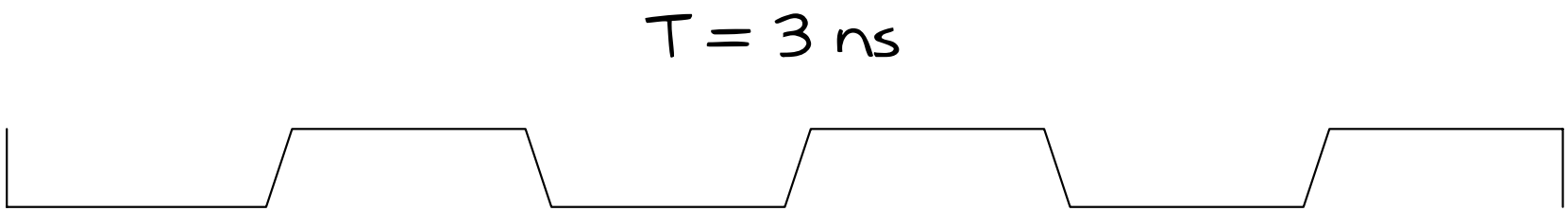
Multiciclo

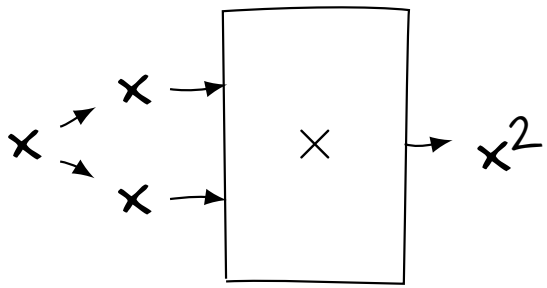
- Comparativo entre ciclo único e multiciclo
  - Implementação em ciclo único de uma instrução hipotética que calcula $f(x) = x^8$



Multiciclo

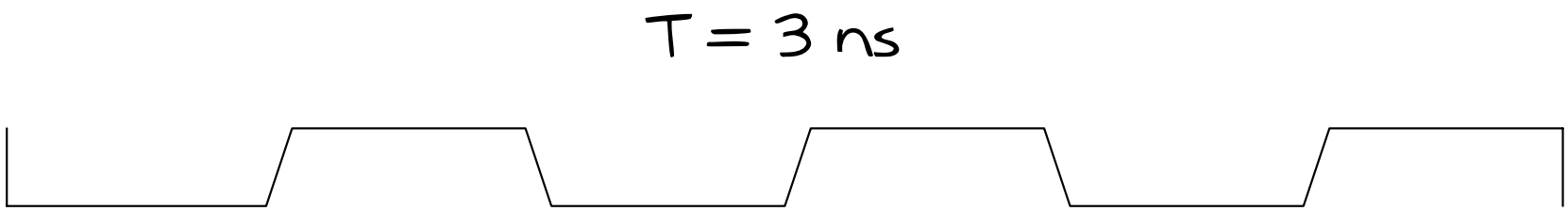
- Comparativo entre ciclo único e multiciclo
  - Implementação em multiciclo de uma instrução hipotética que calcula $f(x) = x^8$

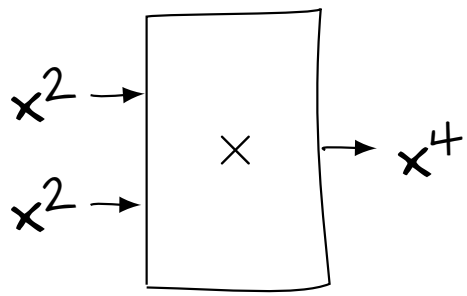




Multiciclo

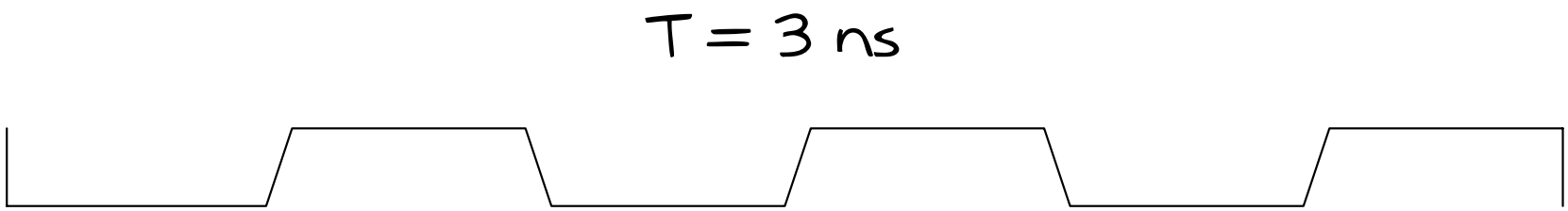
- Comparativo entre ciclo único e multiciclo
  - Implementação em multiciclo de uma instrução hipotética que calcula $f(x) = x^8$

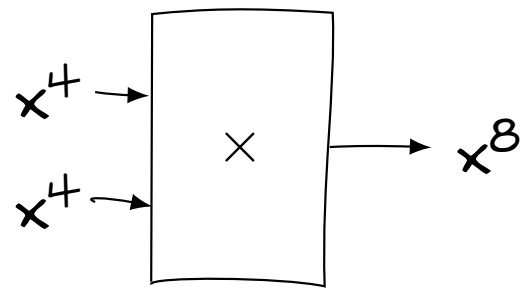




Multiciclo

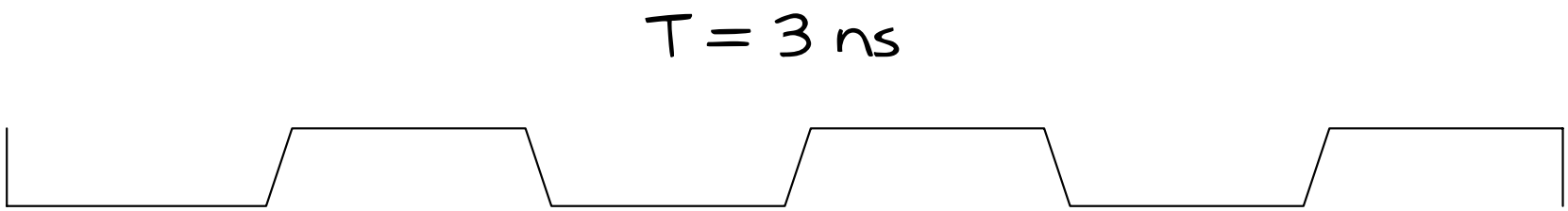
- Comparativo entre ciclo único e multiciclo
  - Implementação em multiciclo de uma instrução hipotética que calcula $f(x) = x^8$

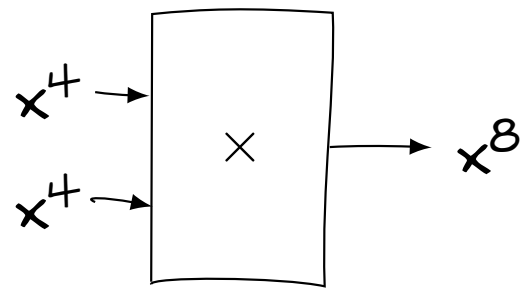




Multiciclo

- Comparativo entre ciclo único e multiciclo
  - Implementação em multiciclo de uma instrução hipotética que calcula $f(x) = x^8$





2 registradores
1 multiplicador

Multiciclo

- Execução em multiciclo
  - Vantagens
    - ✓ Melhor aproveitamento dos recursos de hardware
    - ✓ Quantidade reduzida de operações por ciclo
    - ✓ Maior frequência de relógio

Multiciclo

- Execução em multiciclo
  - Vantagens
    - ✓ Melhor aproveitamento dos recursos de hardware
    - ✓ Quantidade reduzida de operações por ciclo
    - ✓ Maior frequência de relógio
  - Desvantagens
    - ✗ Não reduz o tempo para execução das instruções
    - ✗ Mesma taxa de execução do ciclo único

Pipeline

- O que é uma execução em *pipeline*?
  - Sobreposição das etapas de execução das instruções

Pipeline

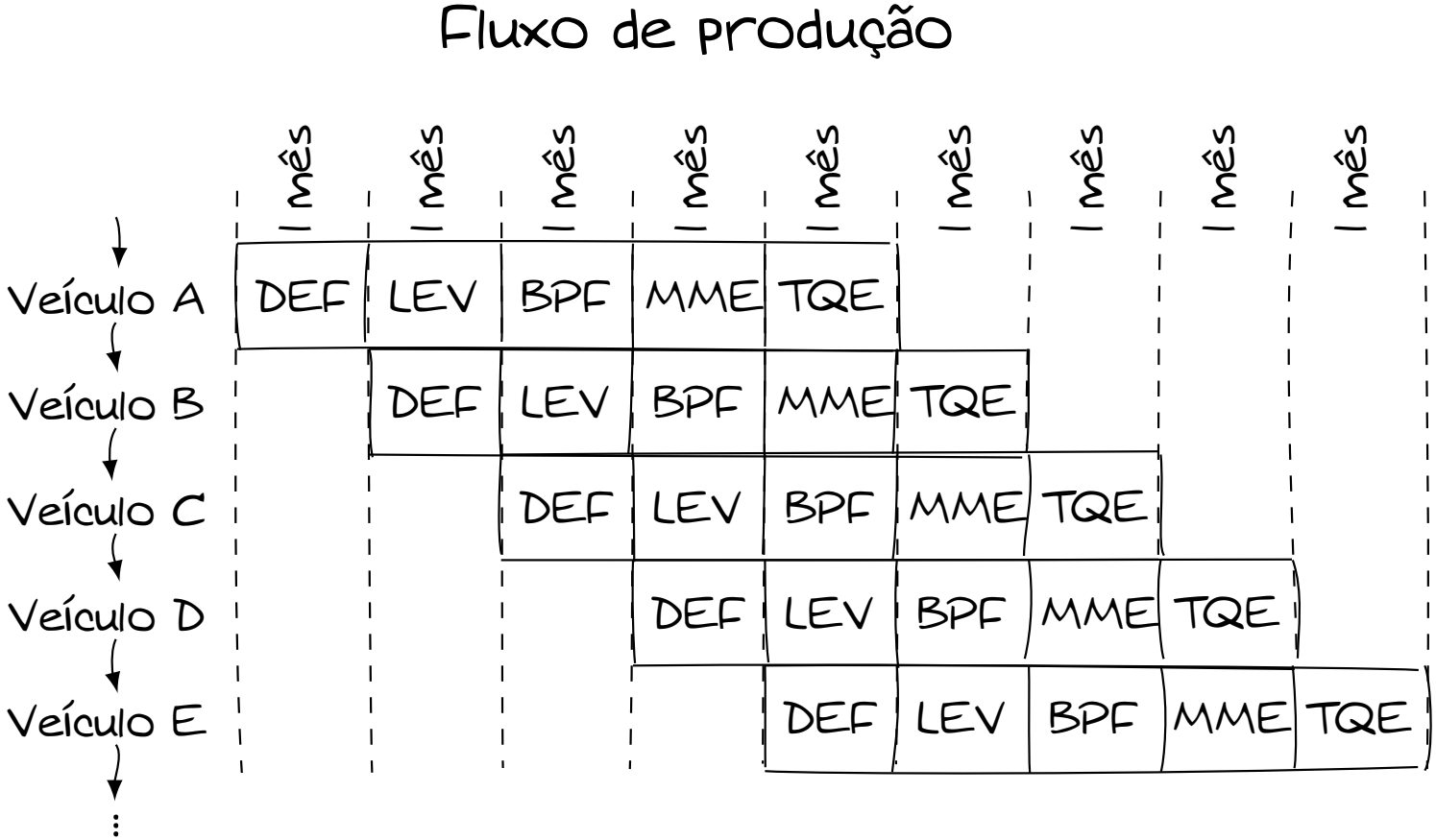
- O que é uma execução em *pipeline*?
  - Sobreposição das etapas de execução das instruções
  - Aumentar a taxa de execução das operações

Pipeline

- O que é uma execução em *pipeline*?
  - Sobreposição das etapas de execução das instruções
  - Aumentar a taxa de execução das operações
  - Possibilitar o aproveitamento completo dos recursos

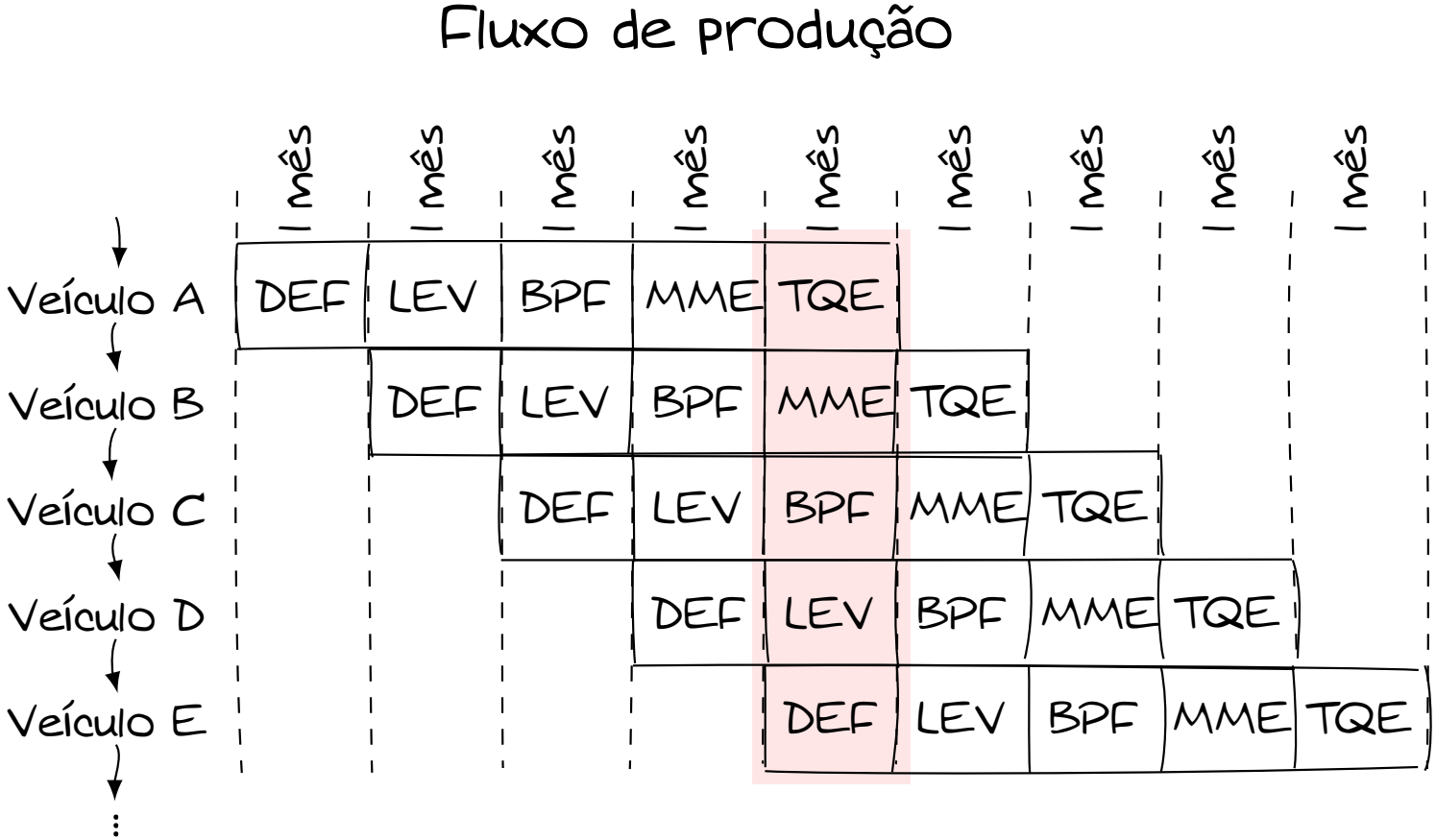
Pipeline

- Analogia com produção de veículos personalizados
  - Cada etapa do processo de fabricação possui um prazo determinado para ser completada
  - As etapas de produção são sobrepostas, executando concorrentemente em diferentes veículos



Pipeline

- Analogia com produção de veículos personalizados
  - Cada etapa do processo de fabricação possui um prazo determinado para ser completada
  - As etapas de produção são sobrepostas, executando concorrentemente em diferentes veículos



Pipeline

- Analogia com produção de veículos personalizados
  - Modelo de fabricação em série das linhas de montagem (segunda revolução industrial)

Pipeline

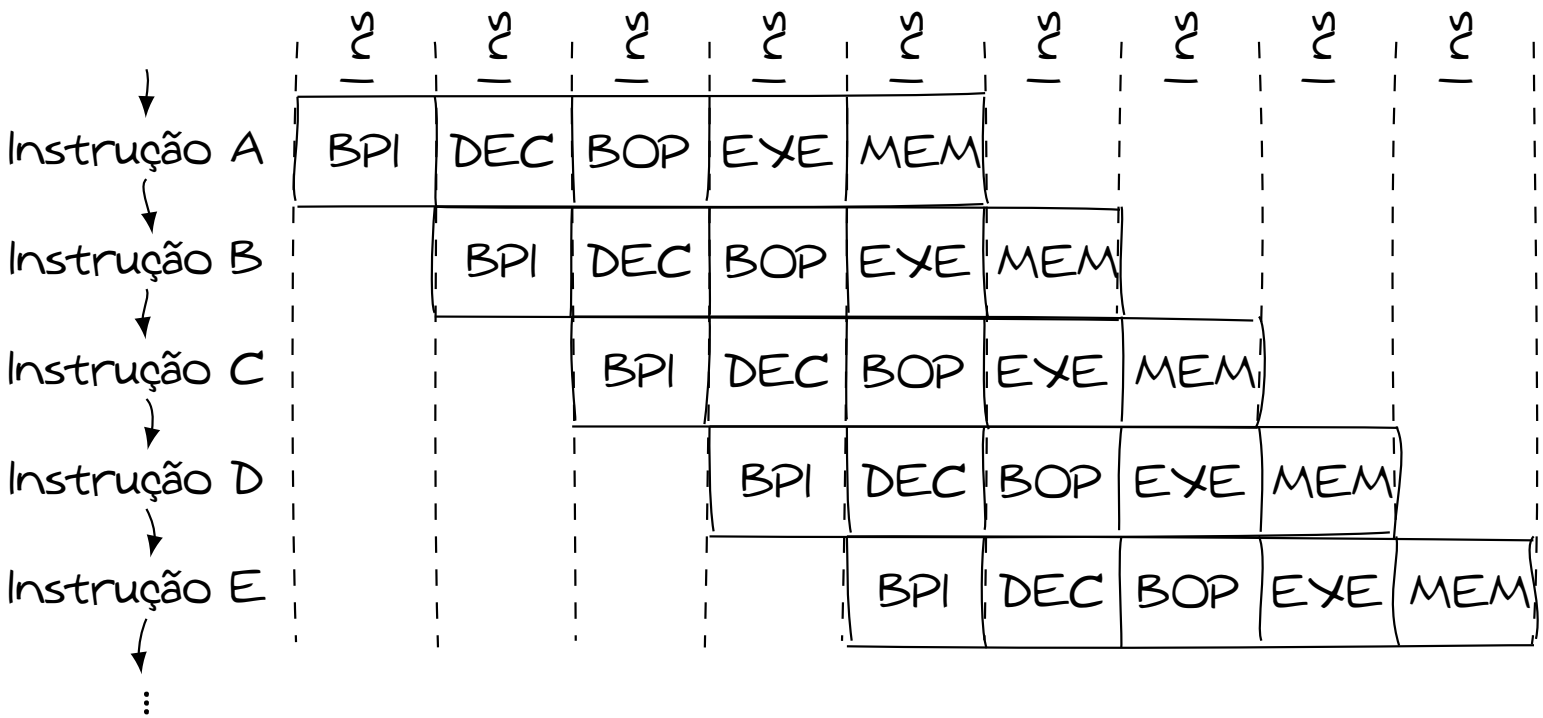
- Analogia com produção de veículos personalizados
  - Modelo de fabricação em série das linhas de montagem (segunda revolução industrial)
  - Cada veículo individualmente continua sendo produzido no mesmo tempo e recursos

Pipeline

- Analogia com produção de veículos personalizados
  - Modelo de fabricação em série das linhas de montagem (segunda revolução industrial)
  - Cada veículo individualmente continua sendo produzido no mesmo tempo e recursos
  - A taxa de produção aumenta de 1 veículo a cada 5 meses para 1 veículo por mês (duração dos estágios)

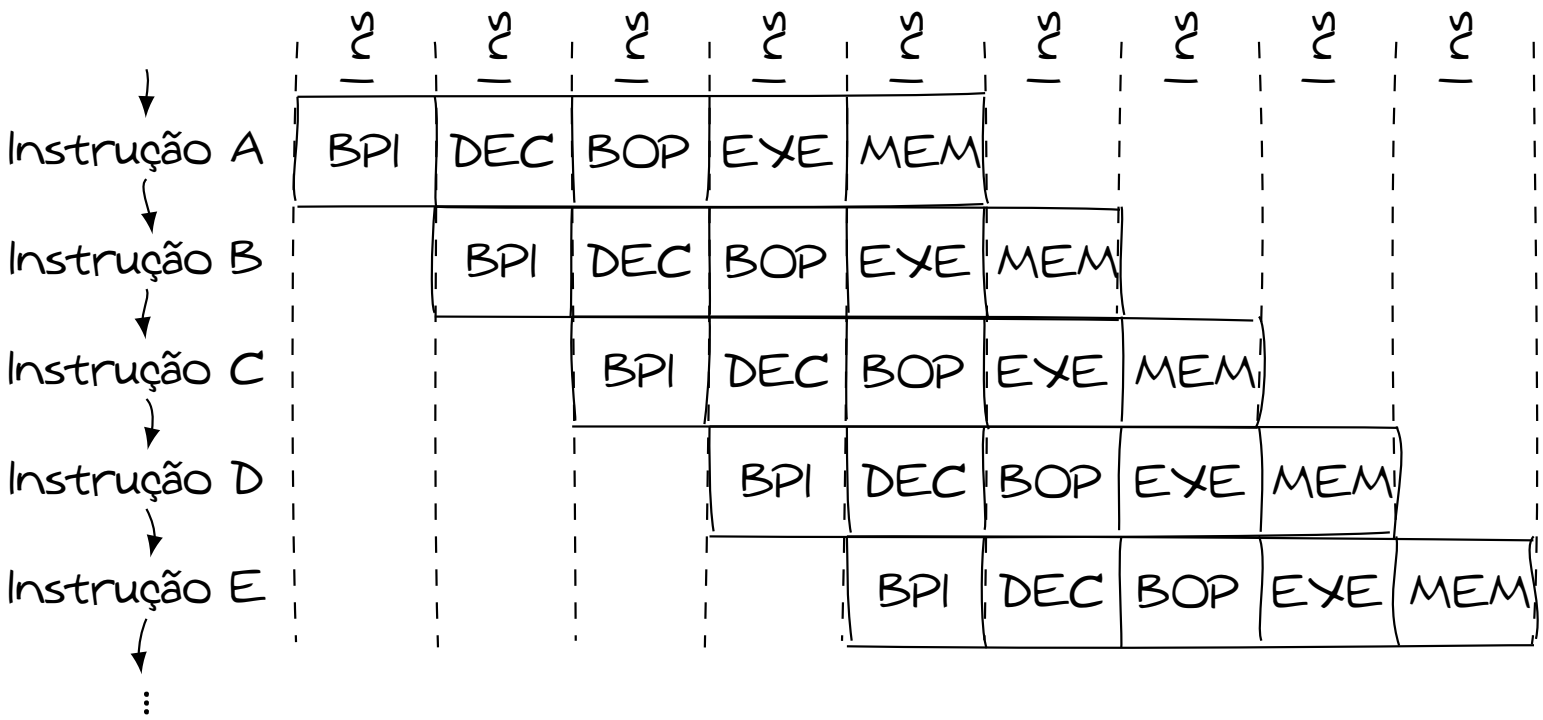
Pipeline

- Execução em *pipeline*
  - Cada etapa da instrução é executada em um ciclo relógio, de forma sobreposta e concorrente



Pipeline

- Execução em *pipeline*
  - Cada etapa da instrução é executada em um ciclo relógio, de forma sobreposta e concorrente



Pipeline não é paralelismo!

Pipeline

- Execução em *pipeline*
  - Cada instrução continua gastando 5 ns para ser executada individualmente, assim como nas implementações de ciclo único e multiciclo

Pipeline

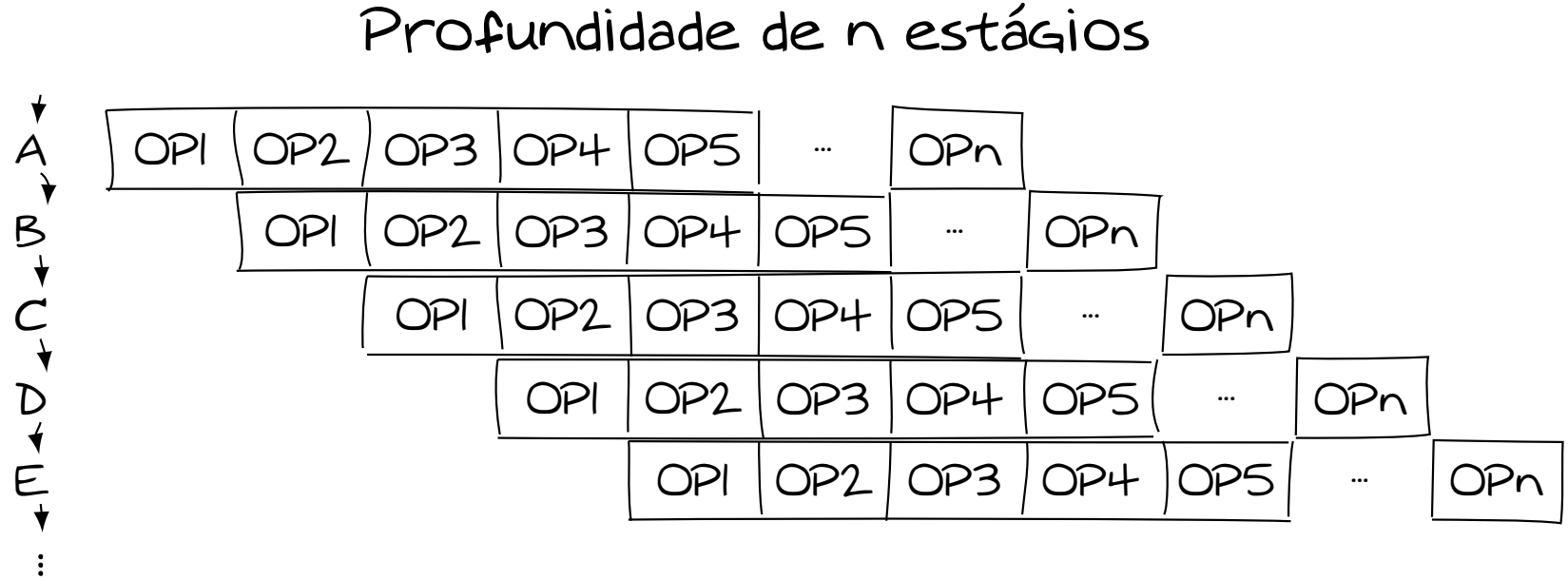
- Execução em *pipeline*
  - Cada instrução continua gastando 5 ns para ser executada individualmente, assim como nas implementações de ciclo único e multiciclo
  - A taxa de execução é aumentada em 5 vezes, permitindo que o processador execute 1 instrução a cada 1 ns ao invés de 1 instrução a cada 5 ns

Pipeline

- Execução em *pipeline*
  - Cada instrução continua gastando 5 ns para ser executada individualmente, assim como nas implementações de ciclo único e multiciclo
  - A taxa de execução é aumentada em 5 vezes, permitindo que o processador execute 1 instrução a cada 1 ns ao invés de 1 instrução a cada 5 ns
  - Este incremento de desempenho é resultante do melhor aproveitamento dos recursos já existentes, mantendo o mesmo comportamento sequencial

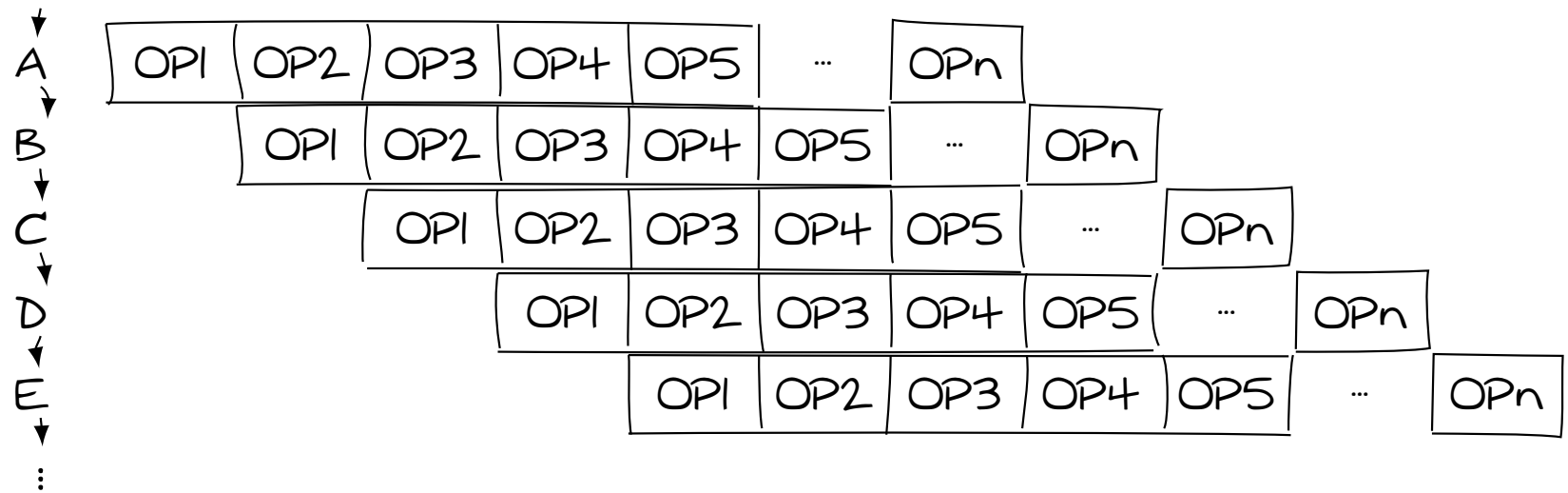
Pipeline

- Projeto de *pipeline*
  - Cada etapa da execução de um *pipeline* é chamada de estágio e, em condições ideais, a quantidade e o tempo de processamento dos estágios definem a taxa de execução



Pipeline

- Projeto de *pipeline*
  - Cada etapa da execução de um *pipeline* é chamada de estágio e, em condições ideais, a quantidade e o tempo de processamento dos estágios definem a taxa de execução



$$\textit{Tempo de instrução} = T = n \times T_S \textit{ ns}$$

$$\textit{Taxa de execução} = \frac{n}{T} \textit{ instruções/ns}$$

Pipeline

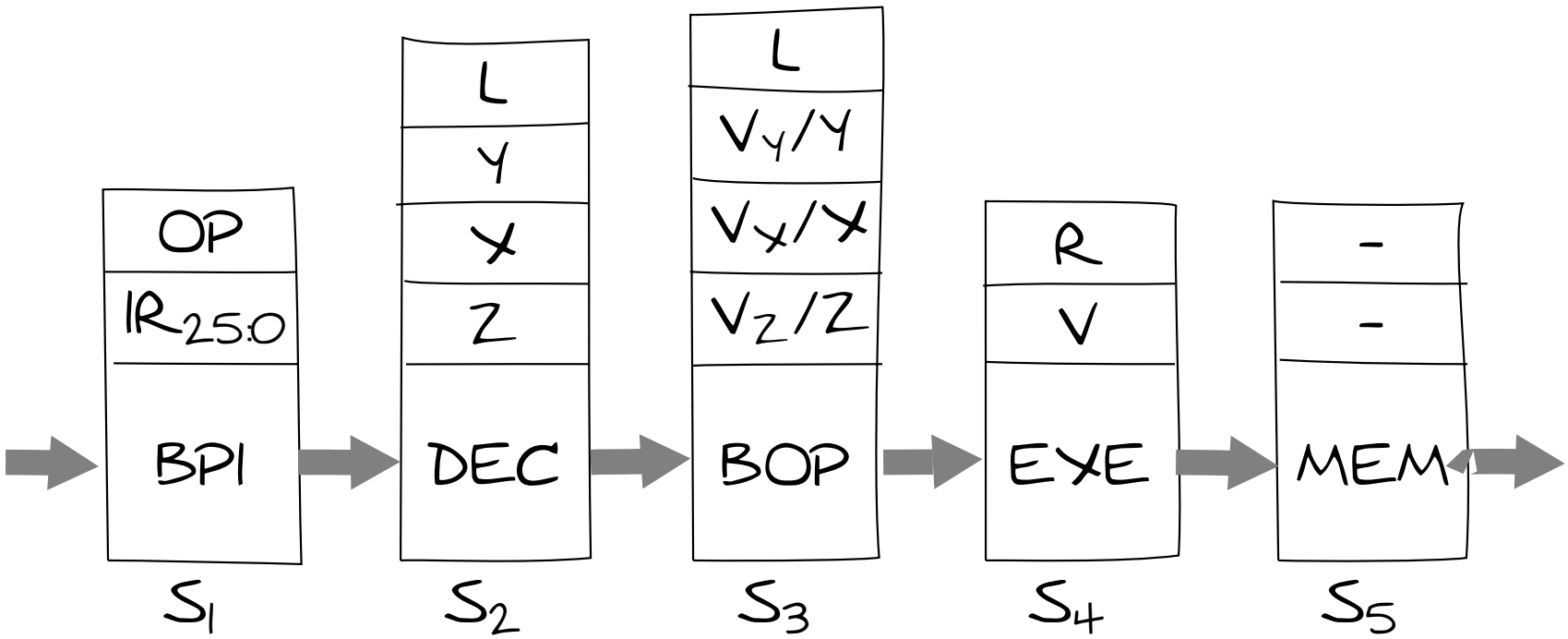
- Projeto de *pipeline*
  - Aumento de desempenho do processador
    - Decorrente do aumento da taxa de execução, com melhor aproveitamento dos recursos de hardware
    - Por executarem sequencialmente, não existe redução do tempo execução individual das instruções

Pipeline

- Projeto de *pipeline*
  - Aumento de desempenho do processador
    - Decorrente do aumento da taxa de execução, com melhor aproveitamento dos recursos de hardware
    - Por executarem sequencialmente, não existe redução do tempo execução individual das instruções
  - Simplificação do projeto
    - Poucas variações nos formatos das instruções
    - Todas as instruções devem ter o mesmo tamanho

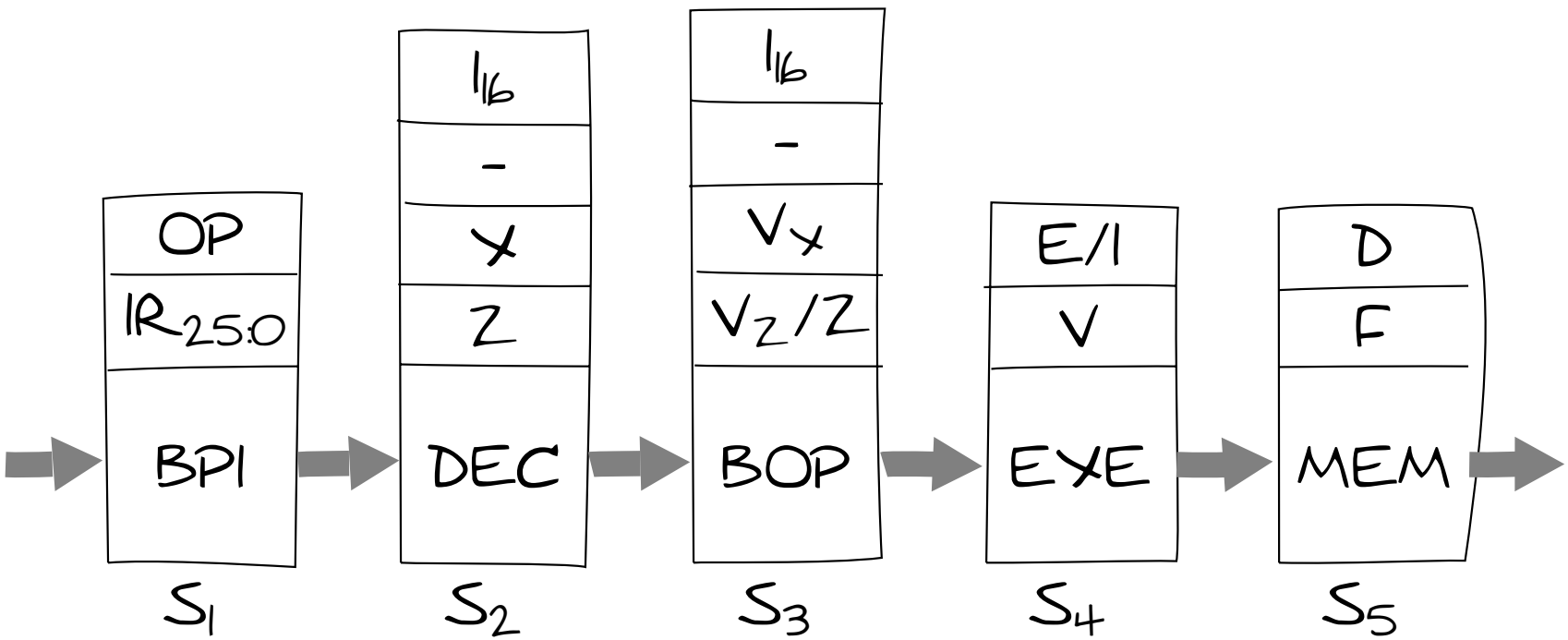
Pipeline

- Representação gráfica dos estágios de *pipeline*
  - Formato U



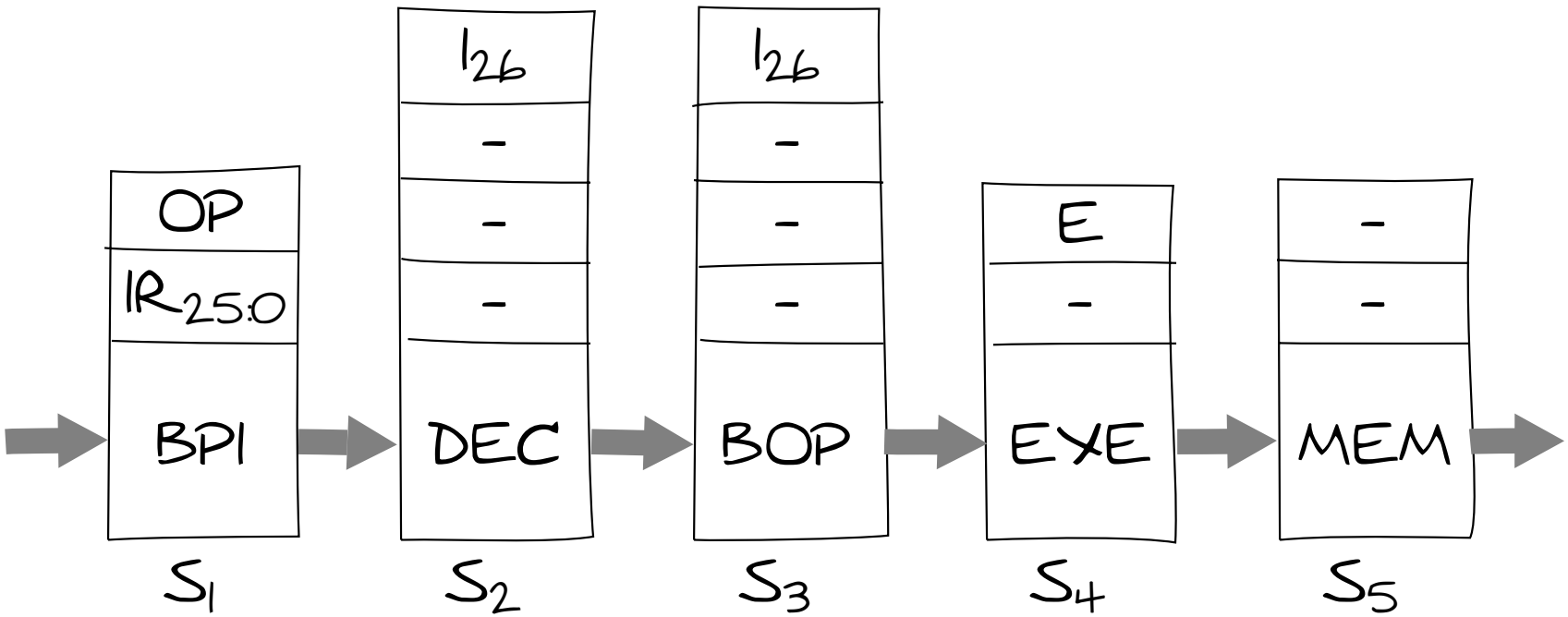
Pipeline

- Representação gráfica dos estágios de *pipeline*
  - Formato F



Pipeline

- Representação gráfica dos estágios de *pipeline*
  - Formato S



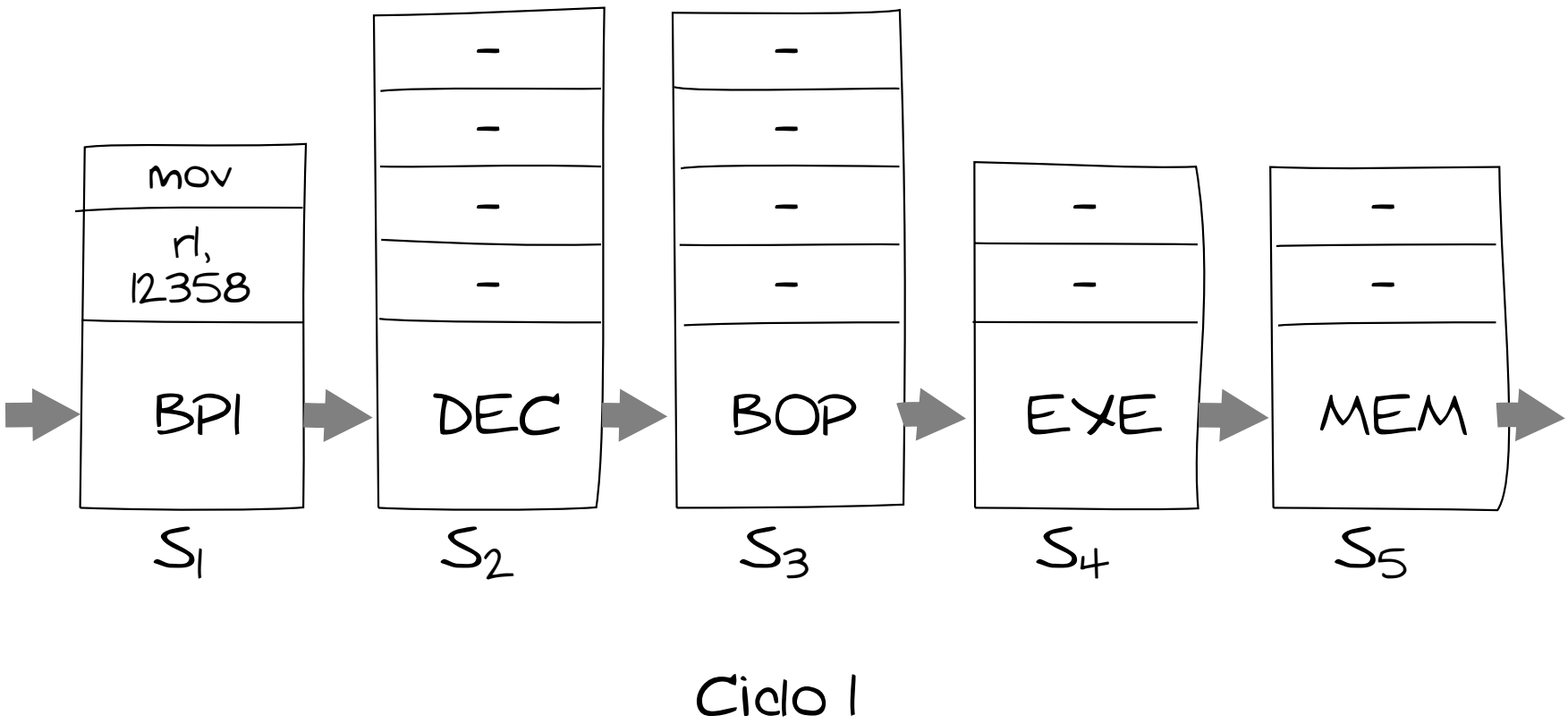
Pipeline

- Execução de programa em *pipeline* com 5 estágios

```
1  // R1 = 12358
2  mov r1, 12358
3  // R2 = 0x40
4  addi r2, r0, 0x40
5  // Desvio incondicional
6  bun 0
7  // MEM[0x100] = 12358
8  s32 [r2], r1
9  // Fim
10 int 0
```

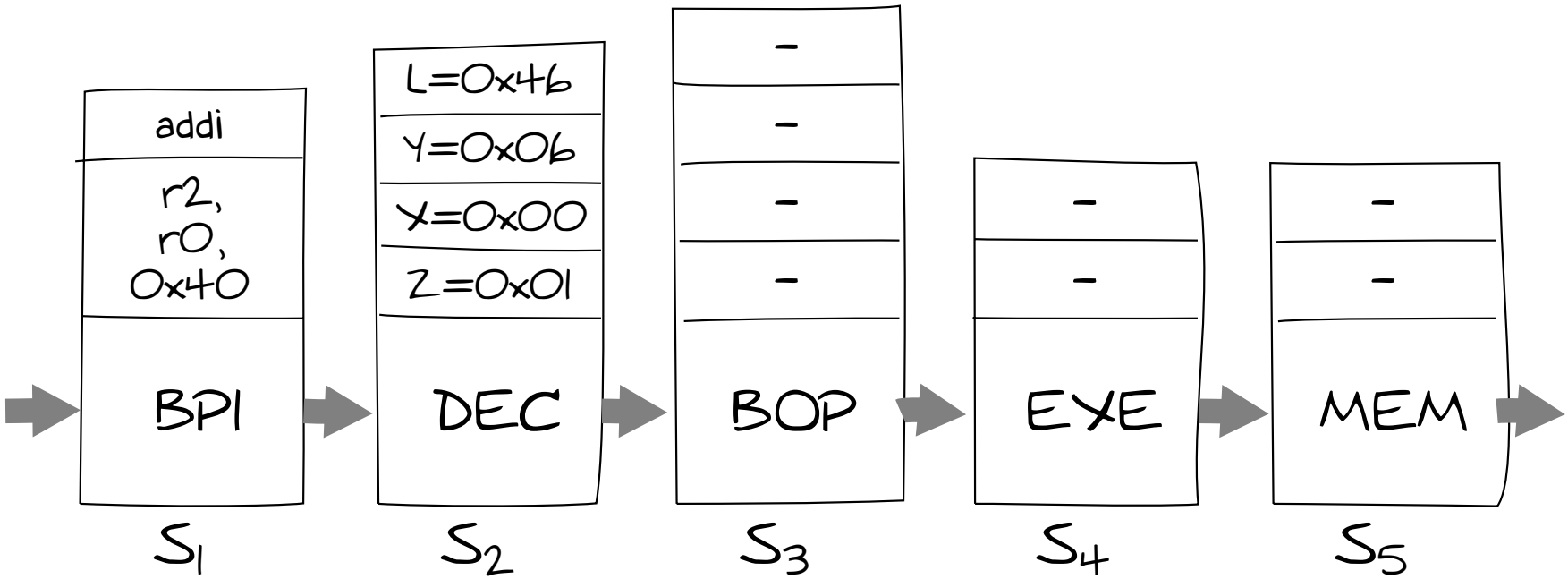
Pipeline

- Execução de programa em *pipeline* com 5 estágios



Pipeline

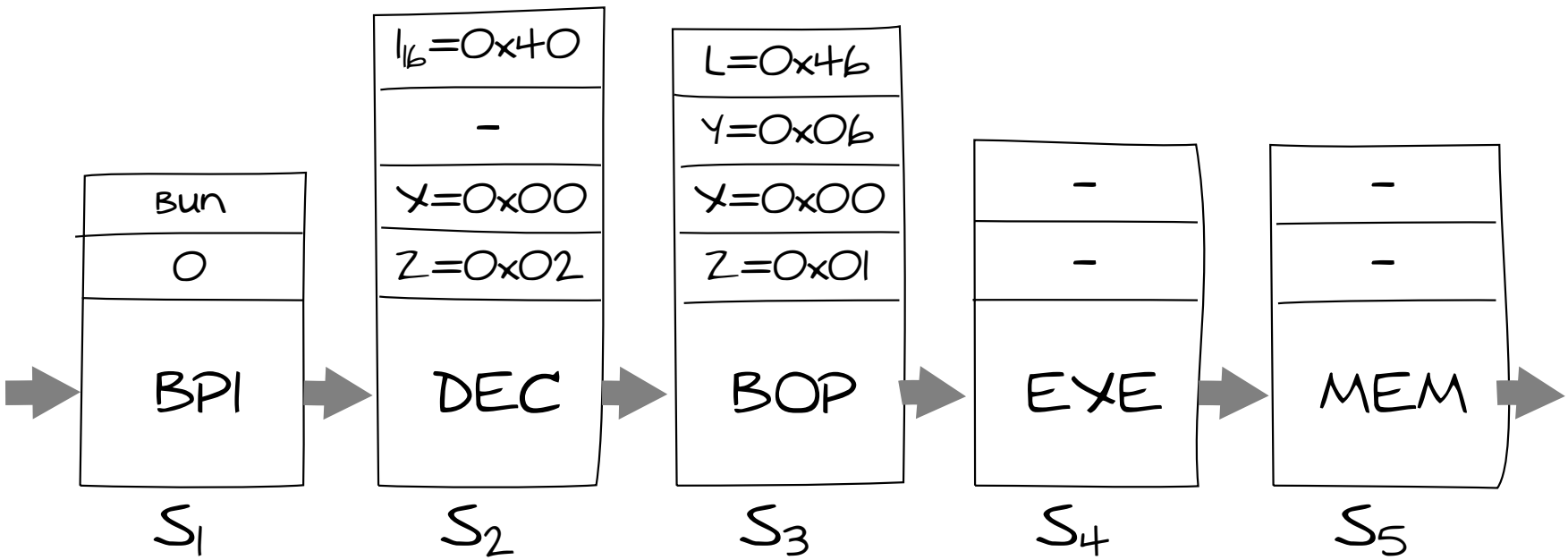
- Execução de programa em *pipeline* com 5 estágios



Ciclo 2

Pipeline

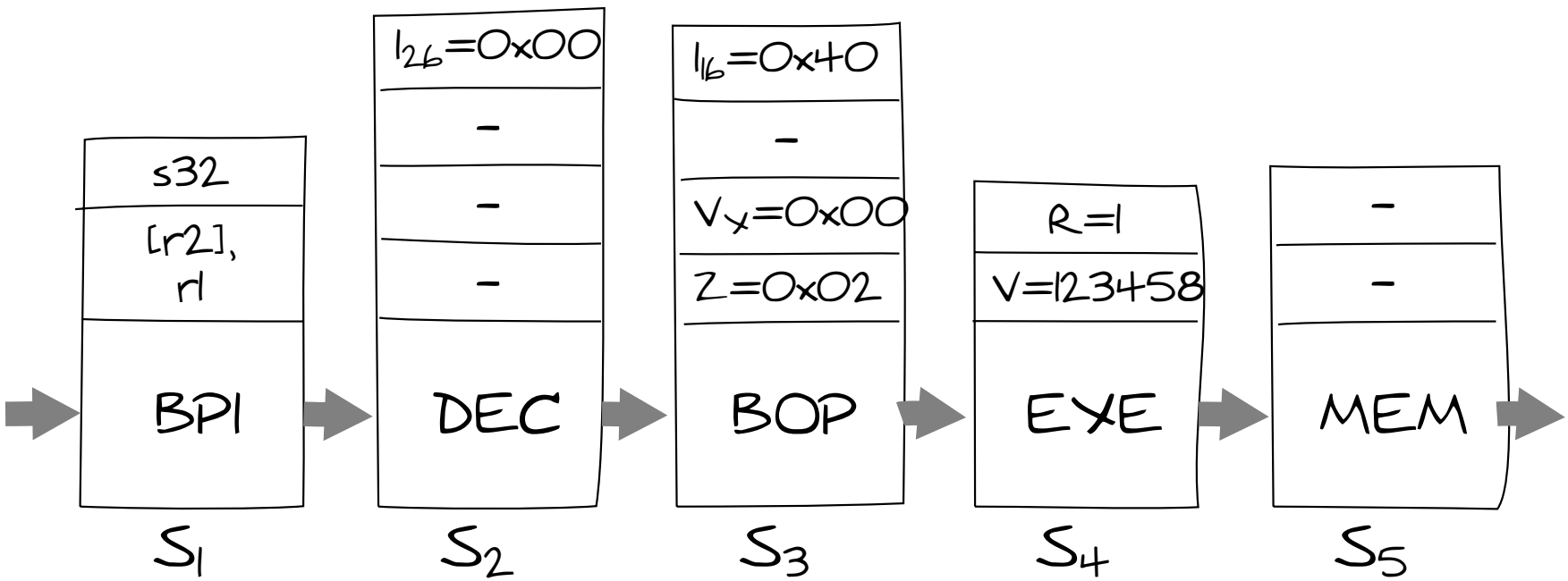
- Execução de programa em *pipeline* com 5 estágios



Ciclo 3

Pipeline

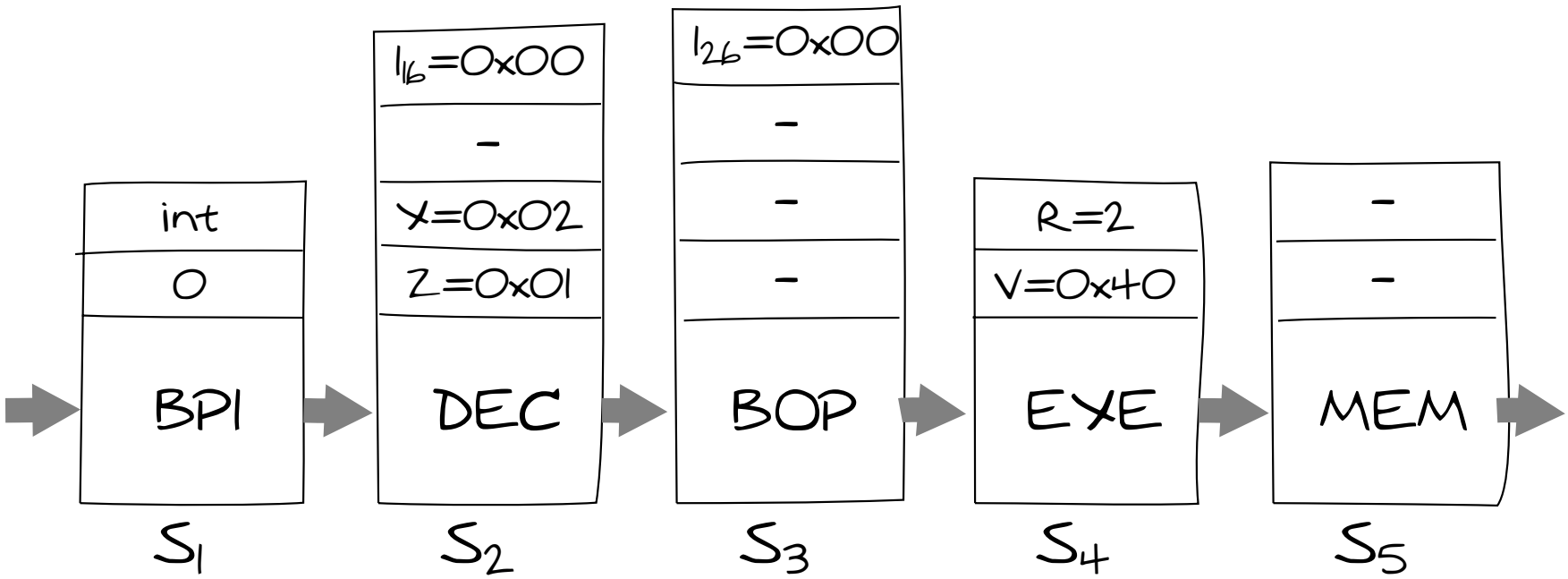
- Execução de programa em *pipeline* com 5 estágios



Ciclo 4

Pipeline

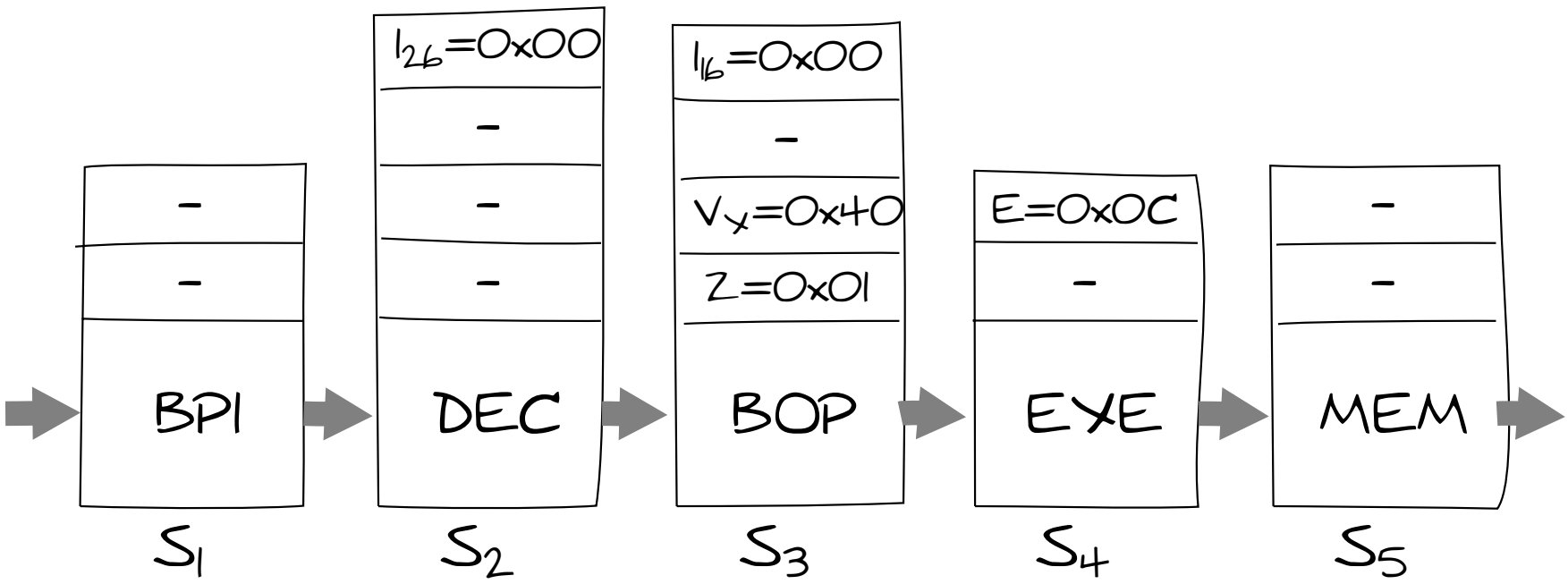
- Execução de programa em *pipeline* com 5 estágios



Ciclo 5

Pipeline

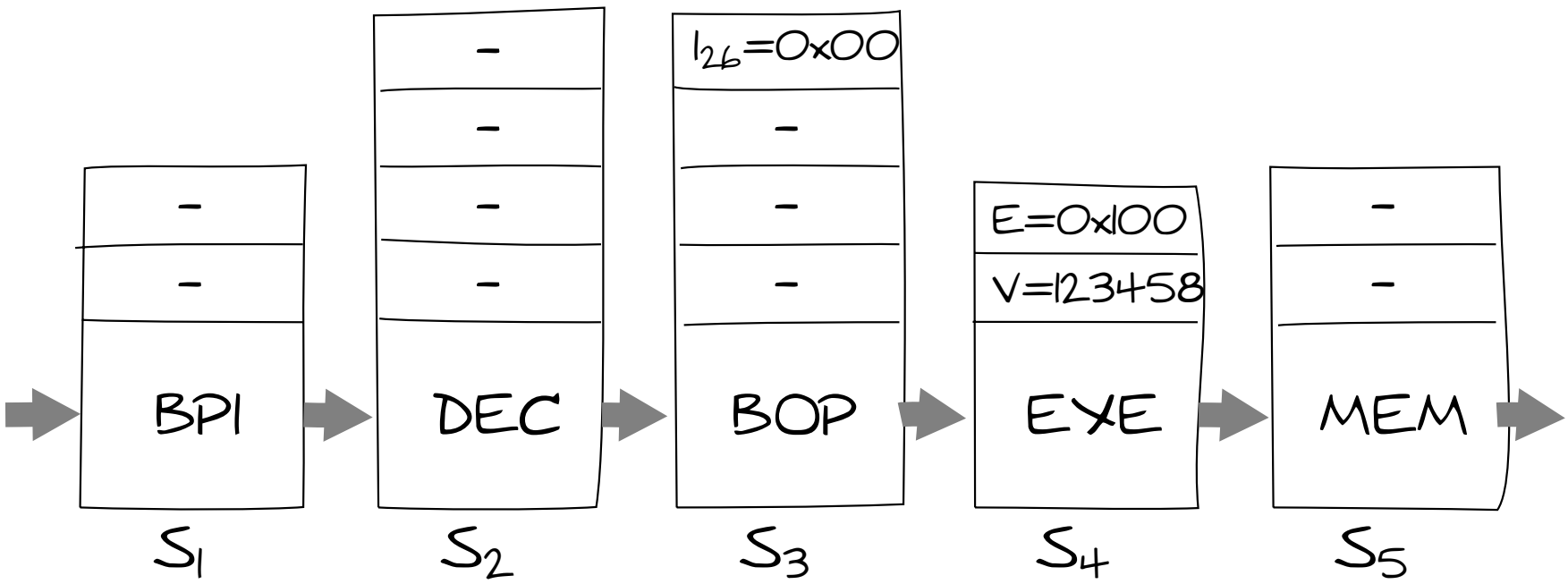
- Execução de programa em *pipeline* com 5 estágios



Ciclo 6

Pipeline

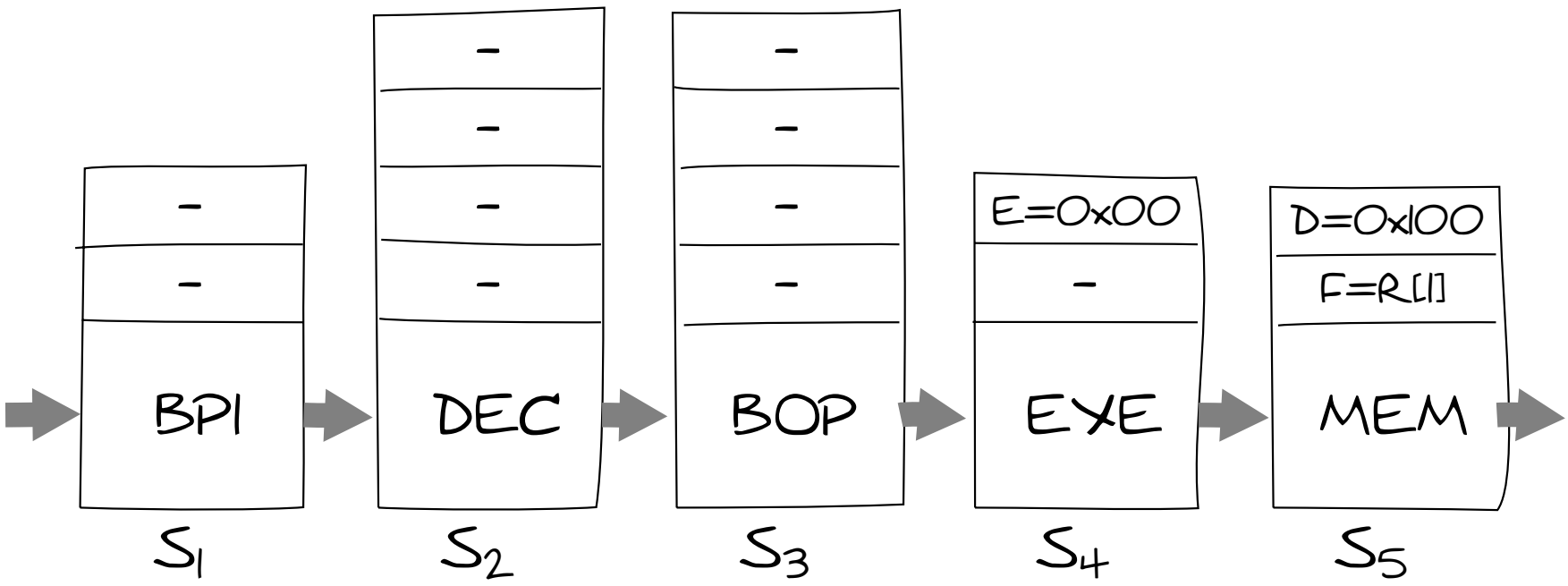
- Execução de programa em *pipeline* com 5 estágios



Ciclo 7

Pipeline

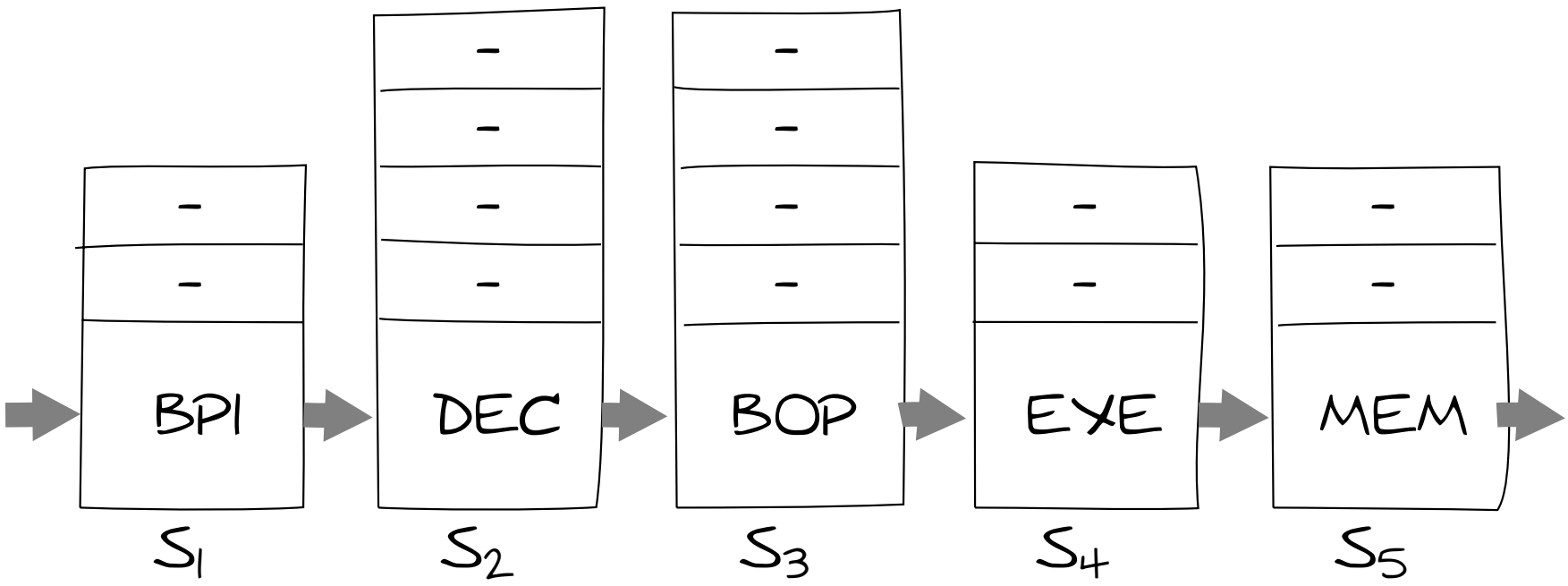
- Execução de programa em *pipeline* com 5 estágios



Ciclo 8

Pipeline

- Execução de programa em *pipeline* com 5 estágios



Pipeline

- Comparativo entre multiciclo e *pipeline*
  - Ciclo de relógio de 1 ns

	Multiciclo	*Pipeline*
# Instruções	5	5
Tempo (ns)	25	9

Pipeline

- Comparativo entre multiciclo e *pipeline*
  - Ciclo de relógio de 1 ns

	Multiciclo	*Pipeline*
# Instruções	5	5
Tempo (ns)	25	9

Cerca de 3 vezes mais rápido!

Exercício

- Execute o código abaixo no *pipeline* de 5 estágios
  - O comportamento deve ser igual ao multiciclo
  - Represente graficamente os estágios do *pipeline*

```
1  // R1 = 0x3C
2  mov r1, 0x3C
3  // R2 = R1
4  add r2, r0, r1
5  // R3 = R2 + 4
6  addi r3, r2, 4
7  // R4 = MEM[0x100]
8  l32 r4, [r3]
9  // Fim
10 int 0
```