Introdução ao Processamento Vetorial

Tópicos:

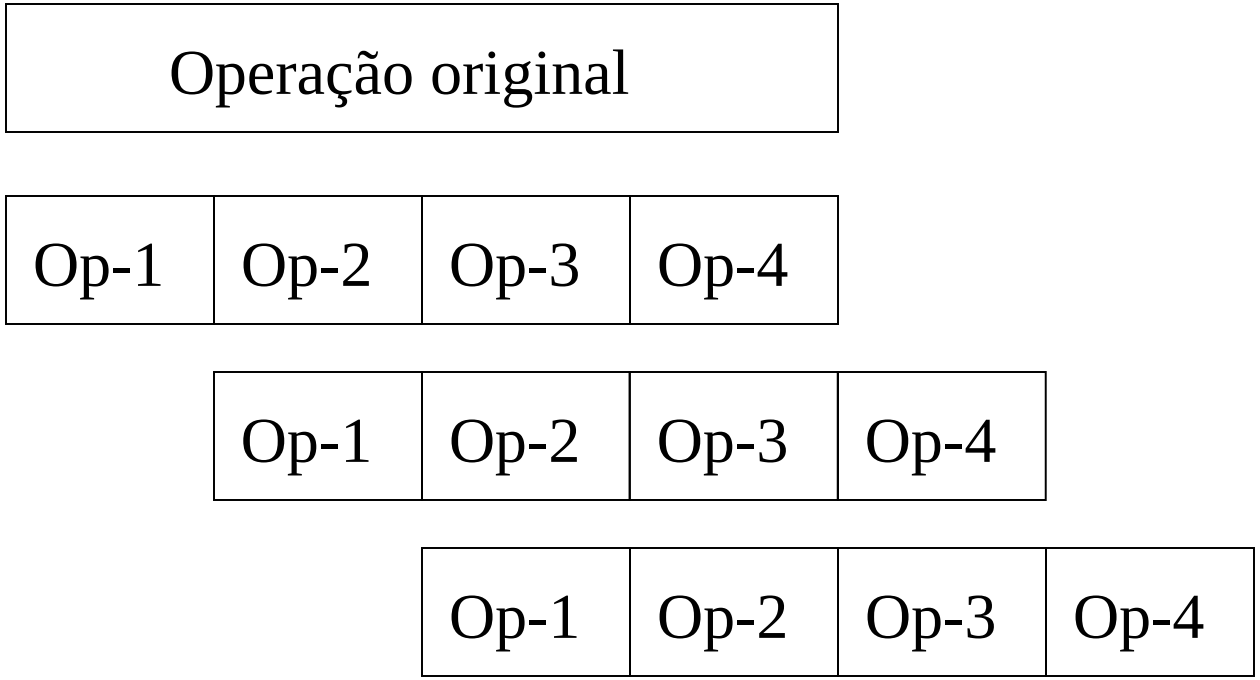
- Conceito de *pipelining*
- Exemplo 1 : *Pipeline* de instruções
- Exemplo 2 : *Pipeline* de operações aritméticas
- Introdução a Sistemas Vetoriais
- Operações Vetoriais
- Registros Vetoriais e Escalares
- Opções de Projeto

Conceito de *Pipelining*

- ***Pipelining***: Operação num modo semelhante a uma *linha de produção:*
  - Operação global pode ser dividida em *estágios*
  - Cada estágio realiza uma operação mais *simples*
  - Tempo de cada estágio pode ser curto
  - Cada estágio recebe resultados do estágio anterior
  - É possível realizar várias operações simultaneamente (cada uma em um estágio num dado momento)
  - Estágios operam sincronamente
  - Ciclo de operação é ditado pelo estágio mais lento

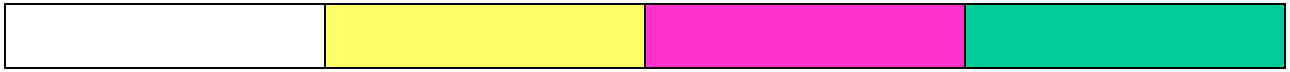
Conceito de *Pipelining* (cont.)

- Exemplo:

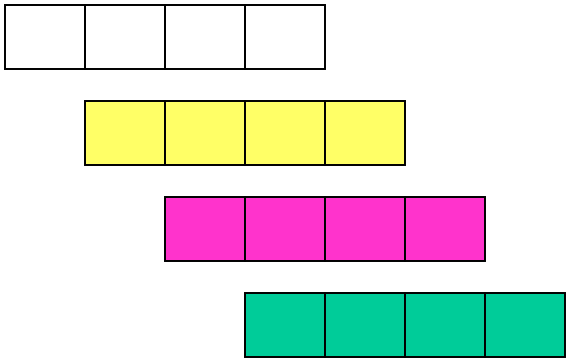


Conceito de *Pipelining* (cont.)

- Execução **sem** pipelining:



- Execução **com** pipelining:

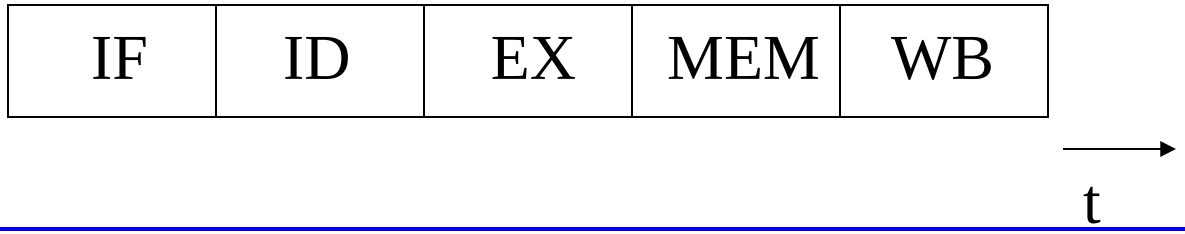


Conceito de *Pipelining* (cont.)

- **Observações**:
  - Tempo de *uma operação* é o mesmo, com ou sem pipelining
  - Tempo de término da *primeira operação* é o mesmo, com ou sem pipelining
  - Ganho começa a existir a partir da *segunda* operação
  - Hipótese: todos os estágios têm a mesma duração

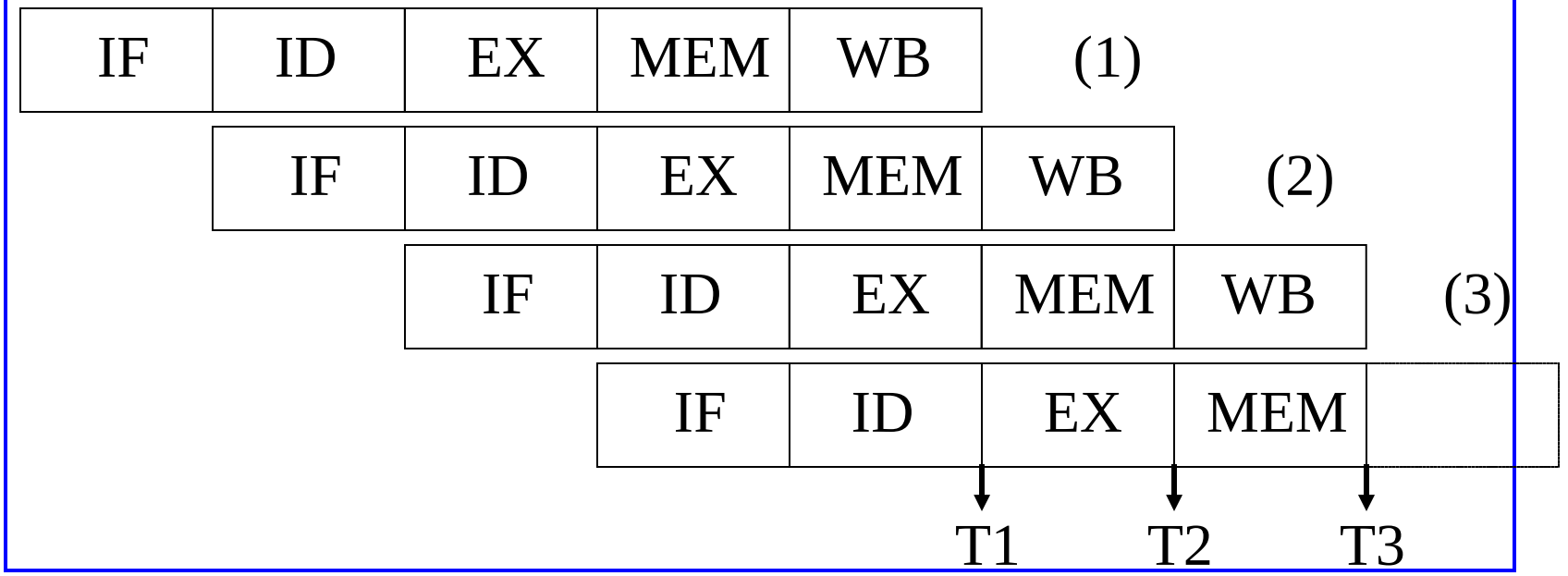
Pipeline de Instruções

- Componentes da execução de *uma* instrução:
  - IF: Instruction Fetch
  - ID: Insruction Decode, register fetch
  - EX: Execution
  - MEM: Memory access
  - WB: register Write Back



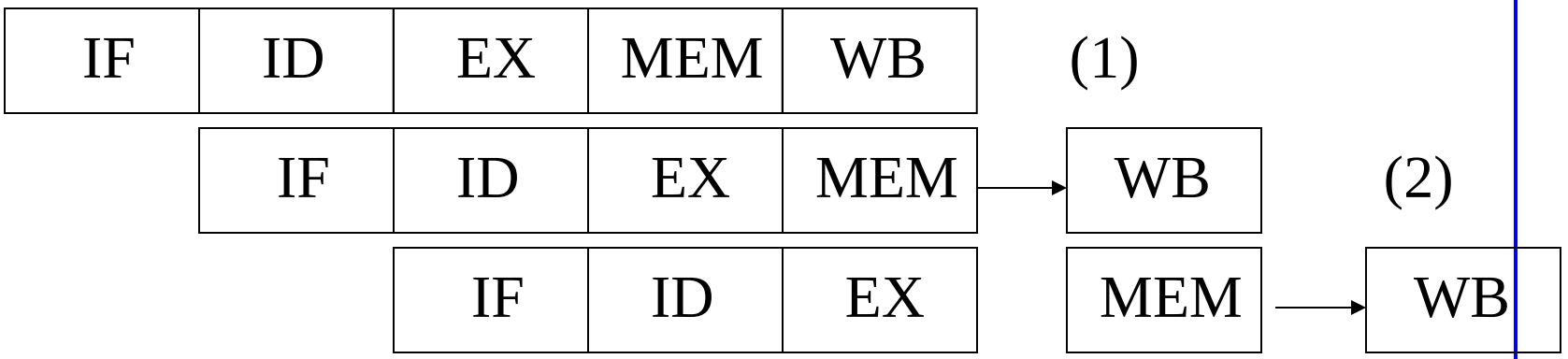
Pipeline de Instruções (cont.)

Objetivo: completar uma instrução por ciclo



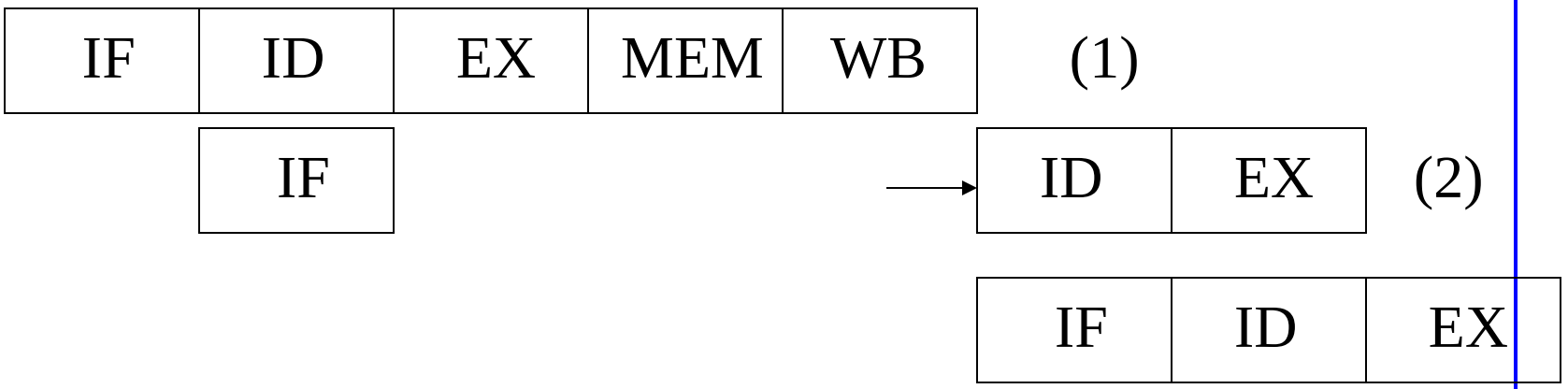
Pipeline de Instruções (cont.)

Problema 1: Atrasos no acesso à memória



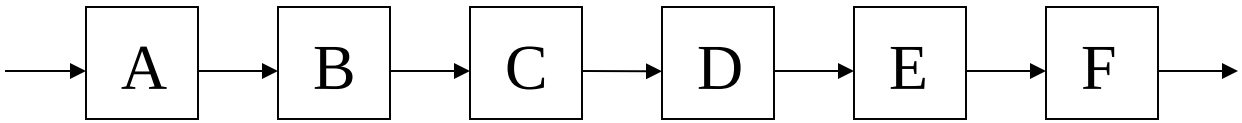
Pipeline de Instruções (cont.)

Problema 2: Dependências entre instruções (1 ⇒ 2)



Pipeline de Operações Aritméticas

- Componentes de *uma* operação de soma em ponto flutuante (mantissa&expoente):
  - **A**: Comparar expoentes
  - **B**: Deslocar mantissa relativa ao menor expoente
  - **C**: Somar mantissas
  - **D**: Normalizar resultado
  - **E**: Verificar excessões (overflow, etc.)
  - **F**: Arredondar resultado



Pipeline de Operações Aritméticas (cont.)

Ex: **X**+**Y** , **X**=$\{x_1,x_2,...x_N\}$, **Y**=$\{y_1,y_2,...,y_N\}$

Execução sem pipeline: Tempo = 6 T . N

Execução **com** pipeline:

	A	**B**	**C**	**D**	**E**	**F**	
T	x_1+y_1						
2T	x_2+y_2	x_1+y_1					
3T	x_3+y_3	x_2+y_2	x_1+y_1				
4T	x_4+y_4	x_3+y_3	x_2+y_2	x_1+y_1			
5T	x_5+y_5	x_4+y_4	x_3+y_3	x_2+y_2	x_1+y_1		
6T	x_6+y_6	x_5+y_5	x_4+y_4	x_3+y_3	x_2+y_2	x_1+y_1	→ 1º result
7T	x_7+y_7	x_6+y_6	x_5+y_5	x_4+y_4	x_3+y_3	x_2+y_2	→ 2º result
8T	x_8+y_8	x_7+y_7	x_6+y_6	x_5+y_5	x_4+y_4	x_3+y_3	→ 3º result
	...						

Introdução a Sistemas Vetoriais

- **Sistemas Vetoriais**: voltados para operações numéricas em *vetores*
- **Vetor**: coleção linear de valores

  Ex: $\mathbf{X} = \{ x_1 , x_2 , x_3 , \ldots , x_N \}$

  $N \rightarrow$ comprimento do vetor
- **Operações Vetoriais**:
  - Realizadas elemento-a-elemento
  - Mesma operação é repetida N vezes
  - Caso típico para aplicação de *pipelining* !

Introdução a Sistemas Vetoriais (cont.)

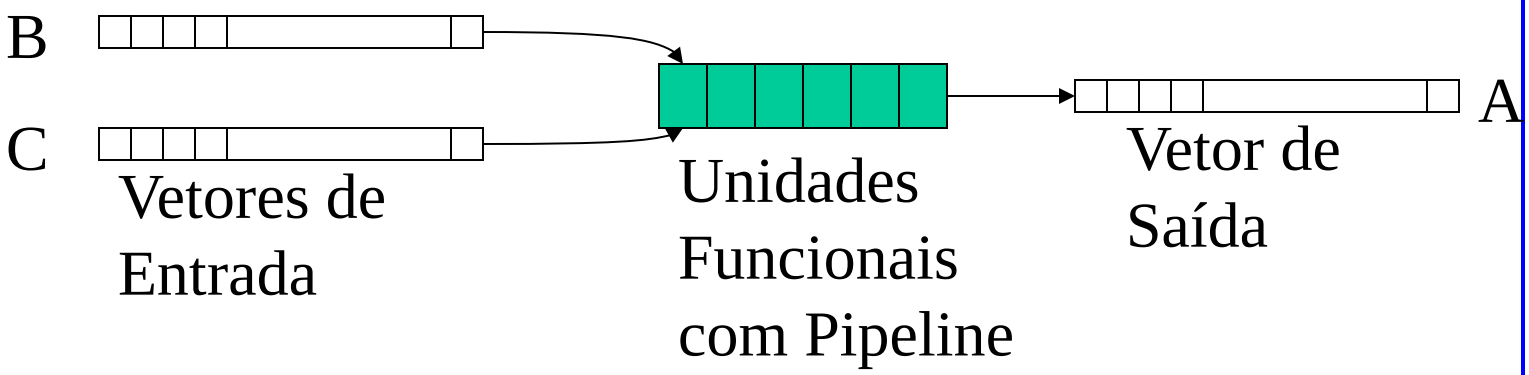
- **Características de Sistemas Vetoriais**:
  - Suporte de hardware para operações vetoriais
  - Instrução vetorial para execução de várias operações similares sobre elementos distintos dos vetores
  - Uso intensivo de *pipelining*
  - Unidades funcionais especiais para execução das operações vetoriais
  - Acesso à memória deve ser rápido e *uniforme* (mesmo tempo de acesso para todos os elementos de um vetor)

∴ Tipicamente, **não** há caches nem memória virtual!

Operações Vetoriais

- Exemplo: **A** = **B** + **C**

  A , **B** , **C** : vetores



Operações Vetoriais (cont.)

- Exemplo: **A** = s * **B** + **C**

  A , **B** , **C** : vetores s : escalar

Código de alto-nível:

```
      real  A(N), B(N), C(N), s
      do  10  i = 1 , N
            A(i) = s * B(i) + C(i)
10    continue
```

Operações Vetoriais (cont.)

- Pseudo-código num processador convencional:

```
        carregar s para registro
        carregar limite do loop para registro
        carregar índice (inicialmente zero)
Loop:   incrementar índice
        comparar índice & limite
        se igual, desviar para Fim
        carregar B(i) para registro
        carregar C(i) para registro
        multiplicar s * B(i)
        somar [s * B(i) ] + C(i)
        salvar resultado na memória
        desviar para Loop
Fim:    terminar a execução
```

Operações Vetoriais (cont.)

- Pseudo-código num processador **vetorial**:

  carregar s para registro escalar
  carregar limite do loop para registro de comprimento
  carregar **B** para registro vetorial
  carregar **C** para registro vetorial
  multiplicar s * **B** (multiplicação vetorial)
  somar s * **B** + **C** (adição vetorial)
  salvar resultado na memória (operação vetorial)

- **Constatações**:
  - Número de instruções executadas é bem menor
  - Número de operações aritméticas é o mesmo

Registros Vetoriais e Escalares

- **Registro Vetorial**:
  - Conjunto de *K* registros convencionais
  - Valores típicos de *K*: 32, 64, 128
    Ex: Cray-1 → 64 registros, 64 bits em cada registro
  - Pode conter valores inteiros ou em ponto-flutuante
  - Pode ser fonte ou destino de operações vetoriais

    Ex: $V_A \leftarrow V_B + V_C$ implica

    $V_A(1) \leftarrow V_B(1) + V_C(1)$
    $V_A(2) \leftarrow V_B(2) + V_C(2)$
    $V_A(3) \leftarrow V_B(3) + V_C(3)$
    ...
    $V_A(K) \leftarrow V_B(K) + V_C(K)$

Registros Vetoriais e Escalares

- **Registro Escalar**:
  - Contém um único valor
  - Pode haver registros separados para inteiros e para ponto-flutuante
  - Pode ser fonte de operações vetoriais

    Ex: $V_A \leftarrow S * V_B$ implica

    $V_A(1) \leftarrow S * V_B(1)$

    $V_A(2) \leftarrow S * V_B(2)$

    $V_A(3) \leftarrow S * V_B(3)$

    ...

    $V_A(K) \leftarrow S * V_B(K)$

Opções de Projeto

- Operandos de operações com unidades funcionais:
  - registros vetoriais
  - memória (não há registros vetoriais)
- Registros vetoriais:
  - Quantidade e tamanho fixos
  - Quantidade e tamanho reconfiguráveis
- Número de portas de leitura/escrita na memória
- Estrutura da memória (com ou sem *interleaving*)
- Operações especiais:
  - *Scatter*, *Gather*, Redução