

Organização de Computadores
Aulas 13
Superescalaridade 1

Superescalaridade

1. Introdução

1. Arquiteturas Superescalares

1. Despacho em ordem, Finalização em ordem

1. Despacho em ordem, Finalização fora-de-ordem

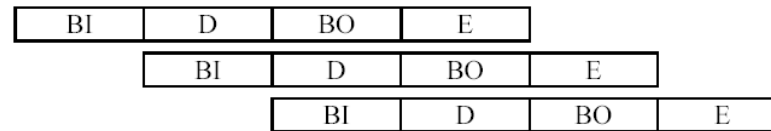
1. Despacho fora-de-ordem, Finalização fora-de-ordem

1. Janela de Instruções Centralizada

INTRODUÇÃO

1. Introdução

- A técnica Pipeline acelera a execução de instruções empregando a sobreposição na execução de etapas da instrução.



- A necessidade de aumento de desempenho nas Arquiteturas Pipeline:
 - Melhor execução das instruções;
 - Mais de uma instrução executada em paralelo;
 - Minimizando perdas devido a dependências.

Solução: Arquiteturas Superescalares!!

Introdução: Princípio do Superescalar

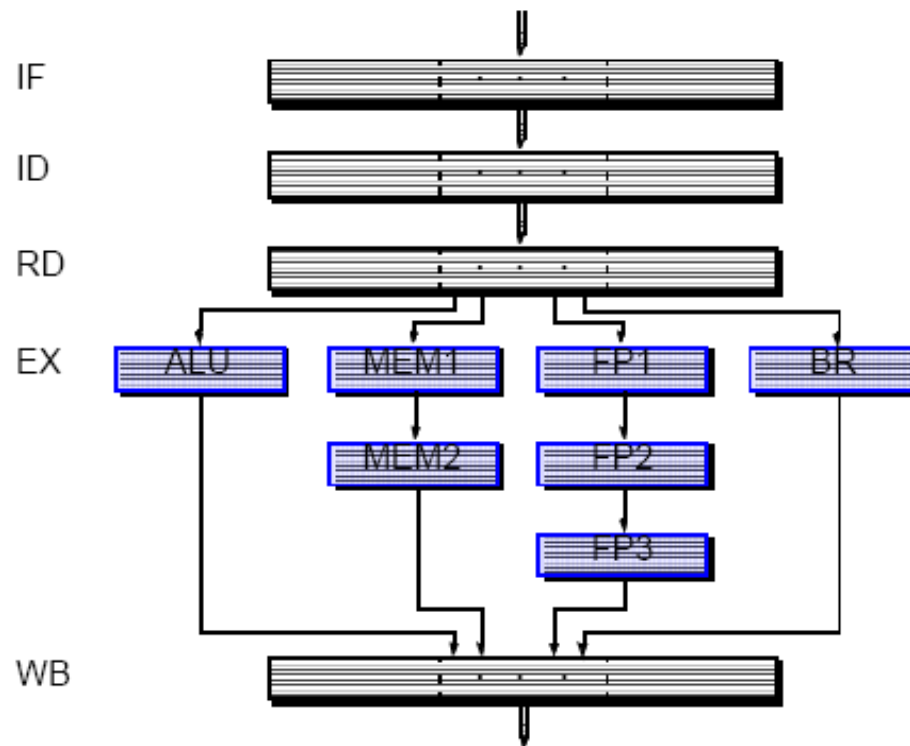
- **Processador Superescalar busca e decodifica várias instruções por ciclo**
- **Instruções executadas em Paralelo, desde que independentes**
- **ILP – Instruction Level Parallelism**
- **Necessário verificar a não existência de dependências entre estas**
- **Compromisso com a semântica seqüencial => Manter o ordenamento inicial do programa**

BI	D	BO	E
BI	D	BO	E

BI	D	BO	E
BI	D	BO	E

BI	D	BO	E
BI	D	BO	E

Arquiteturas com mais de um Pipeline



Introdução

- **Princípios da super-escalaridade**
 - várias unidades de execução
 - várias instruções completadas simultaneamente em cada ciclo de relógio**ILP –Instruction Level Paralelim**
- **Hardware é responsável pela extração de paralelismo**
- **Na prática, obtém-se IPC pouco maior do que 2**
 - limitação do paralelismo intrínseco dos programas
- **PROBLEMAS com a execução simultânea de instruções**
 - conflitos de acesso a recursos comuns
 - memória
 - dependências de dados
 - verdadeiras
 - falsas - anti-dependências, dependências de saída
 - dependências de controle (desvios)

Introdução - Características

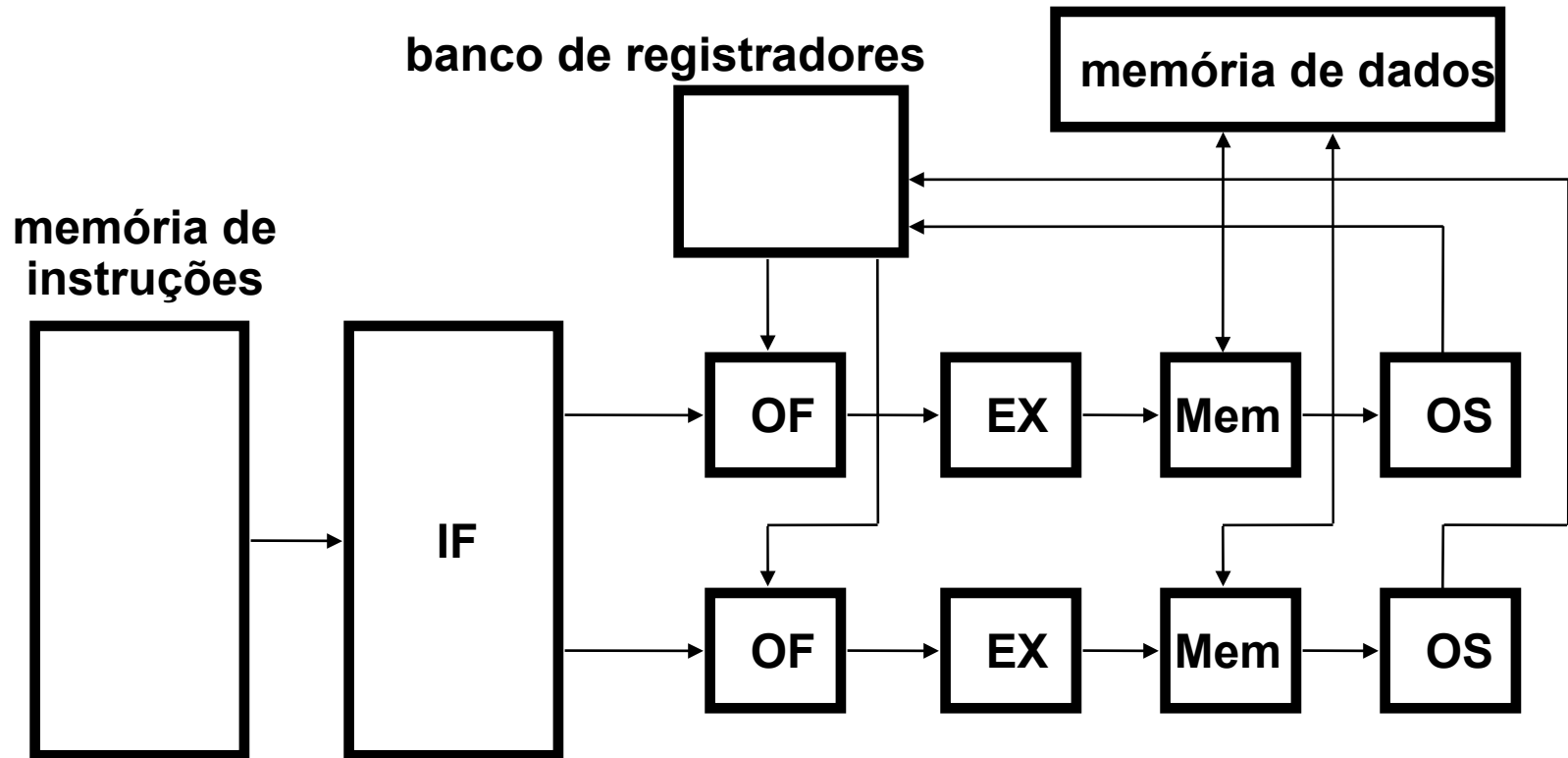
- **Pipelines ou unidades funcionais podem operar com velocidades variáveis – latências**
- **Término das instruções pode não seguir a seqüência estabelecida no programa**
- **Processador com capacidade de “look-ahead”**
 - **se há conflito que impede execução da instrução atual, processador**
 - **examina instruções além do ponto atual do programa**
 - **procura instruções que sejam independentes**
 - **executa estas instruções**
- **Possibilidade de execução fora de ordem**
 - **cuidado para manter a correção dos resultados do programa**

Características do Superescalar

- **Principais Características:**
 - técnicas para determinação de dependências entre os dados dos registradores
 - busca de múltiplas instruções por ciclo
 - antever/prever os desvios condicionais
 - despacho de múltiplas instruções
 - recuperação do ordenamento das instruções
 - comunicação de dados
 - recursos para execução paralela de múltiplas instruções

Arquiteturas Superescalares

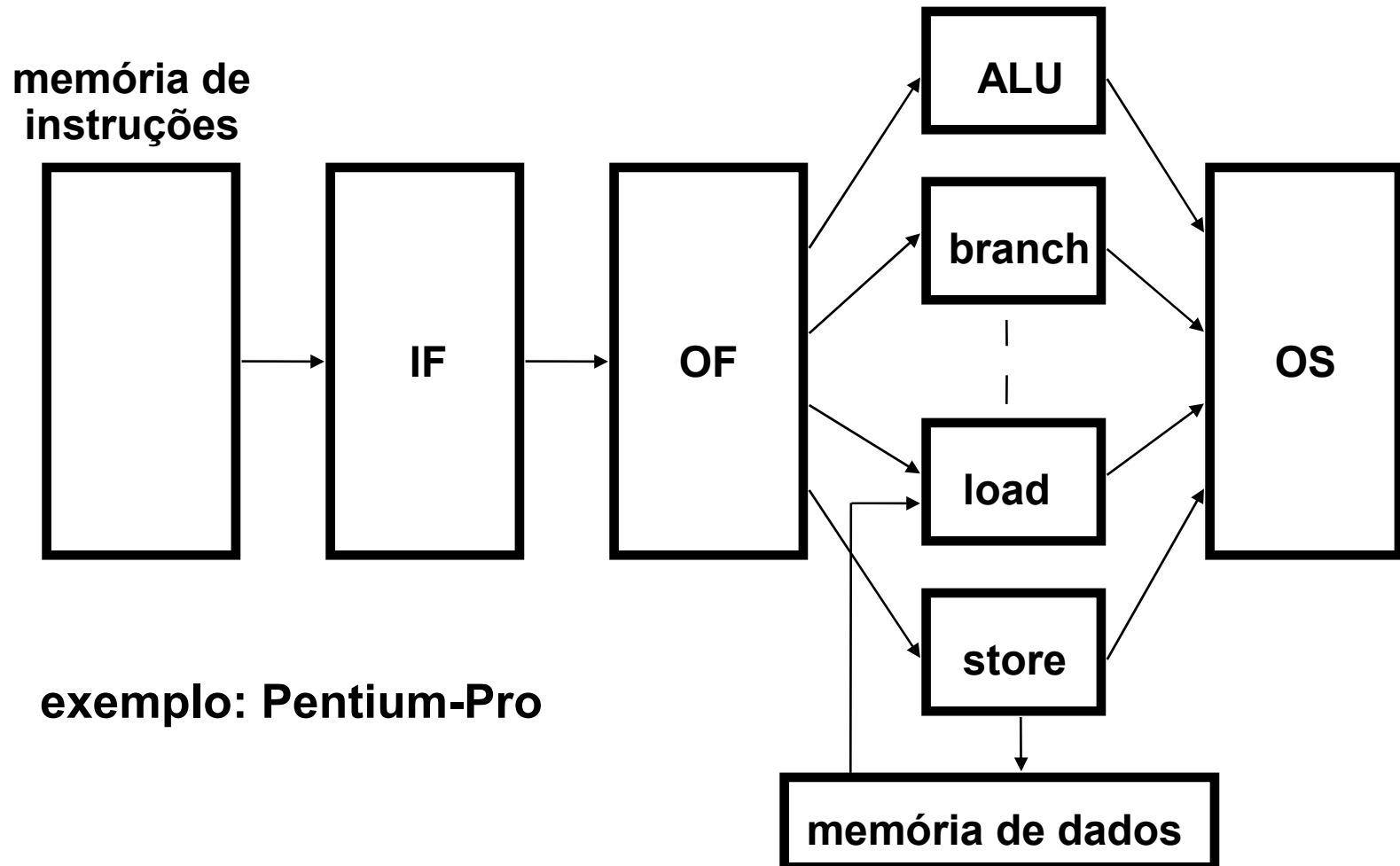
Arquitetura Superescalar: Processador com 2 pipelines



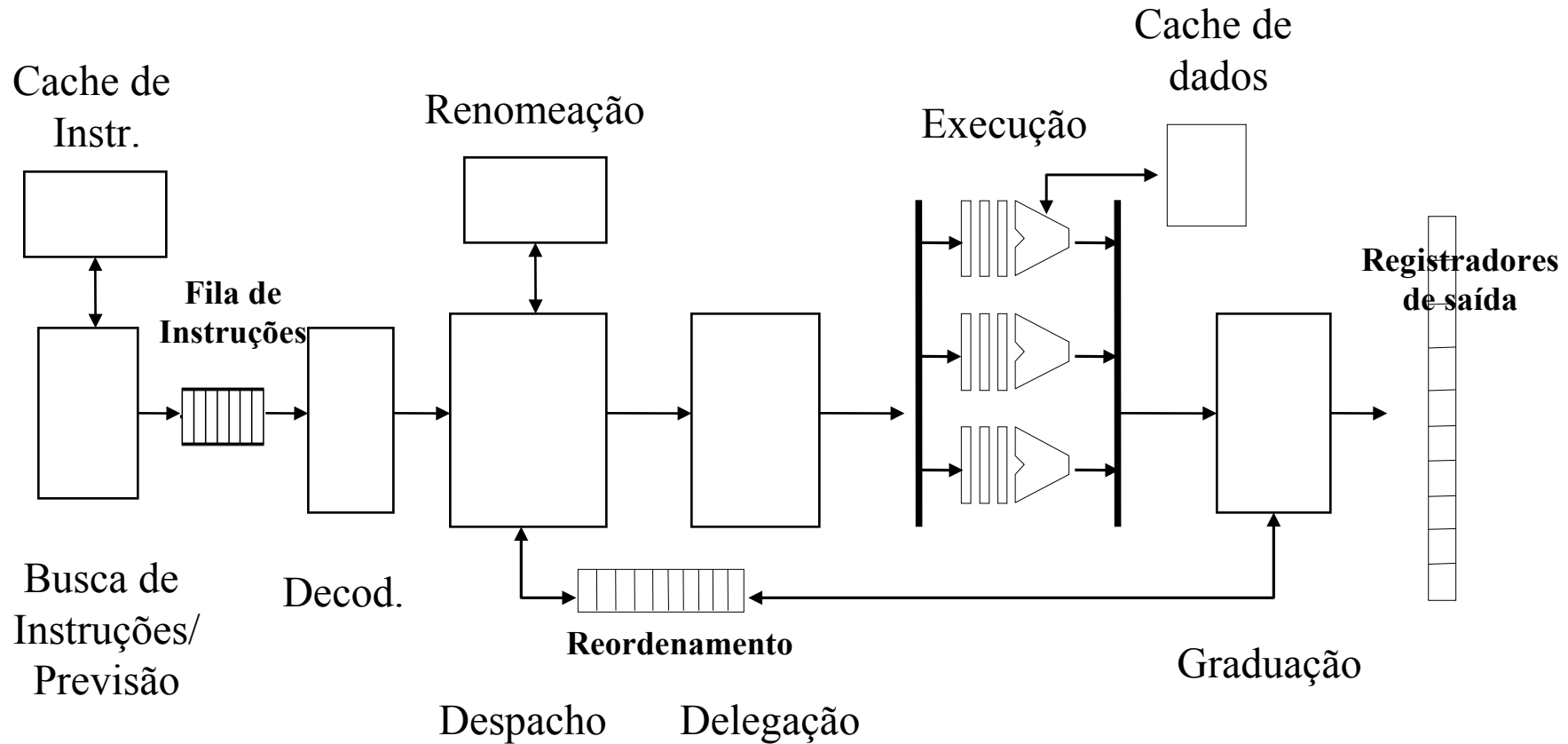
exemplo: Pentium I

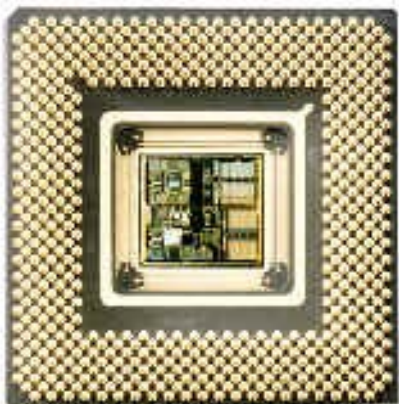
cache de instruções precisa fornecer o dobro de instruções por ciclo

Arquitetura Superescalar: Unidades de Execução Especializadas



Arquitetura Superescalar



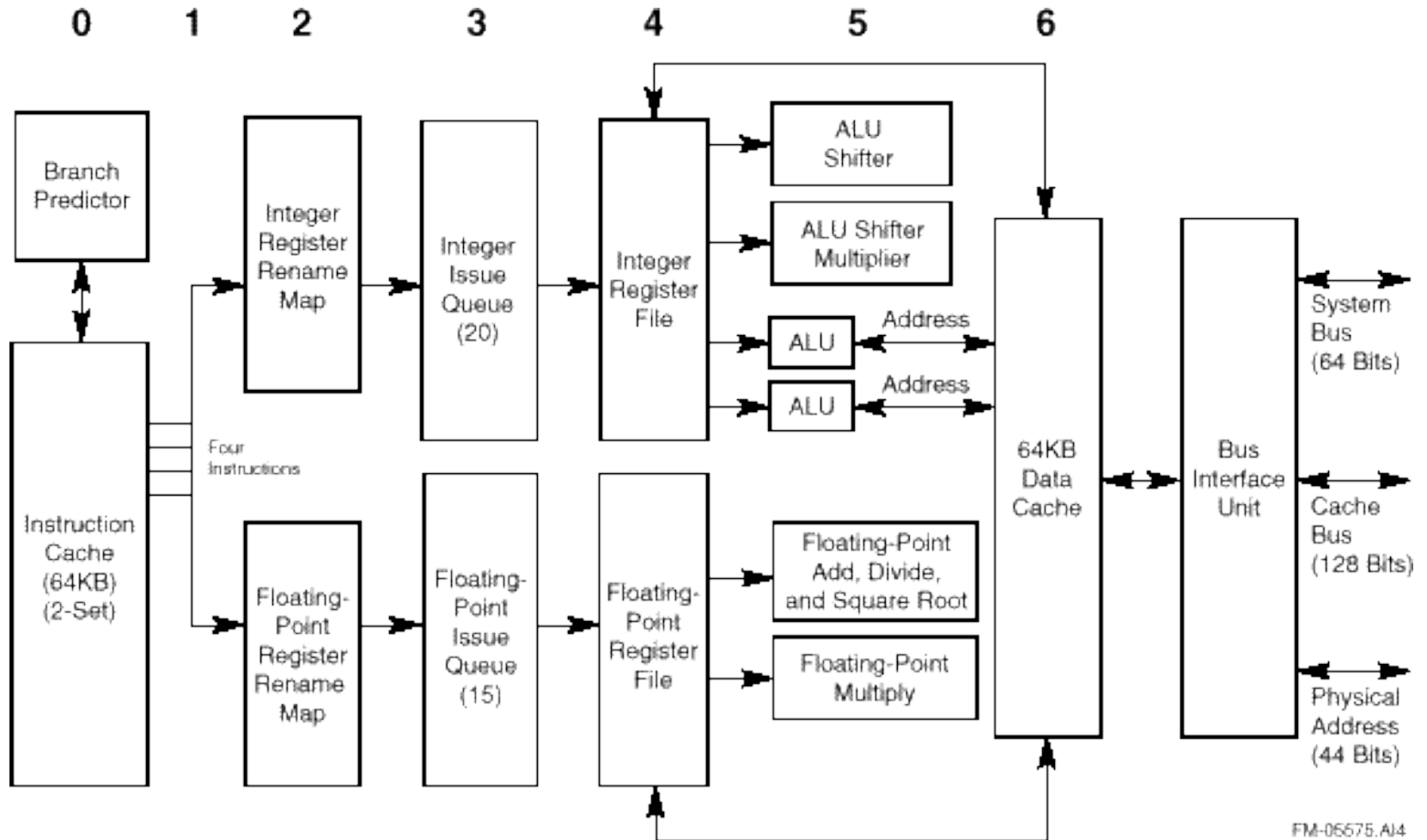


Exemplo: Compaq Alpha 21264

Compaq Alpha 21264

- **Very similar instruction set to MIPS**
- **1 64KB Instruction cache,
1 64 KB Data cache on chip;
16MB L2 cache off chip**
- **Clock cycle = 1.5 nanoseconds, or 667 MHz clock rate**
- **Superscalar: fetch up to 6 instructions /clock cycle,
retires up to 4 instruction/clock cycle**
- **Execution out-of-order**
- **15 million transistors, 90 watts!**

Compaq Alpha 21264



Despacho de Instruções

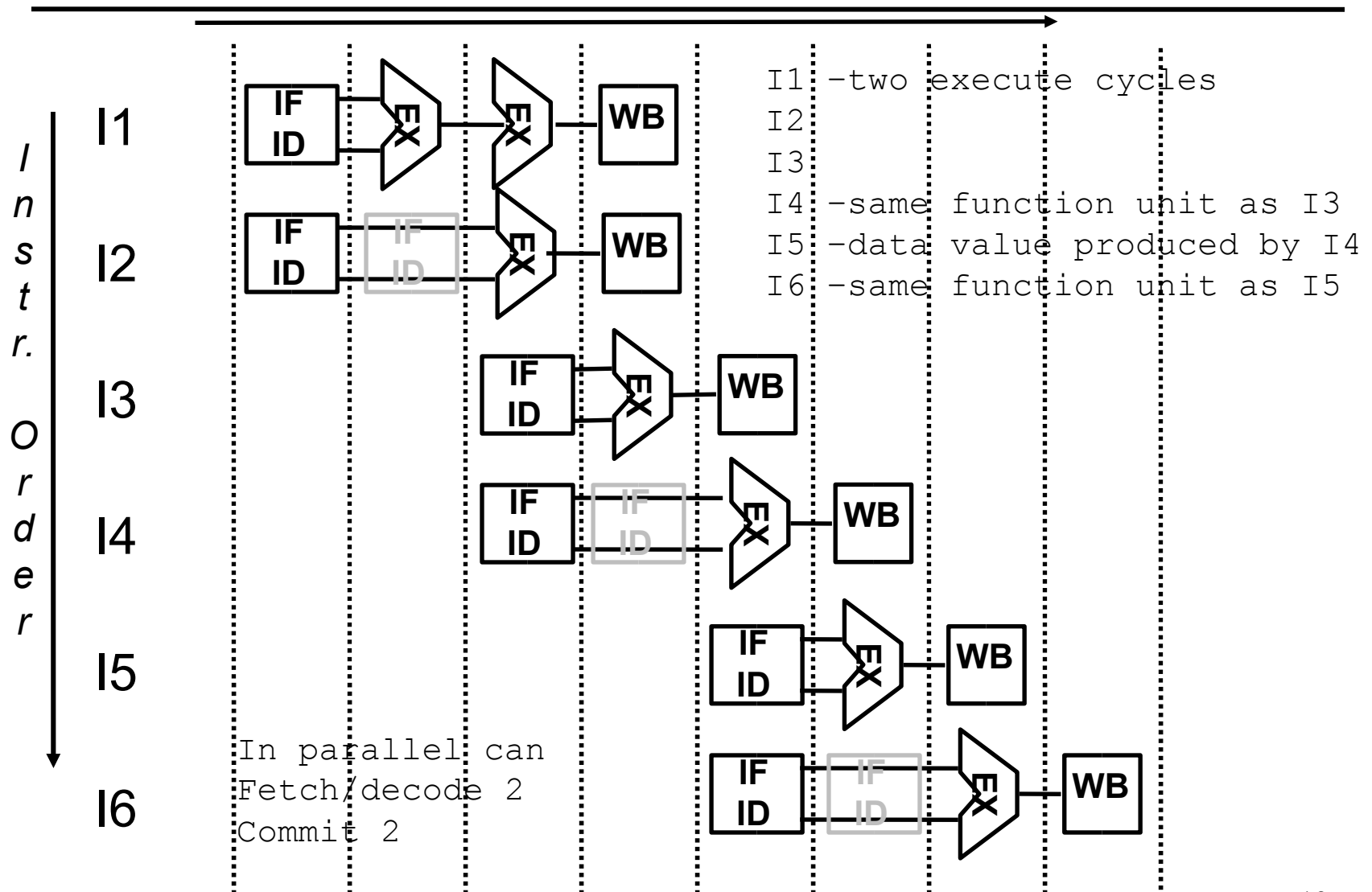
Despacho e Finalização de Instruções

- **Despacho de instruções**
 - refere-se ao fornecimento de instruções para as unidades funcionais
- **Finalização de instruções**
 - refere-se à escrita de resultados (em registradores, no caso de processadores RISC)
- **Alternativas**
 - despacho em ordem, finalização em ordem
 - despacho em ordem, finalização fora de ordem
 - despacho fora de ordem, finalização fora de ordem

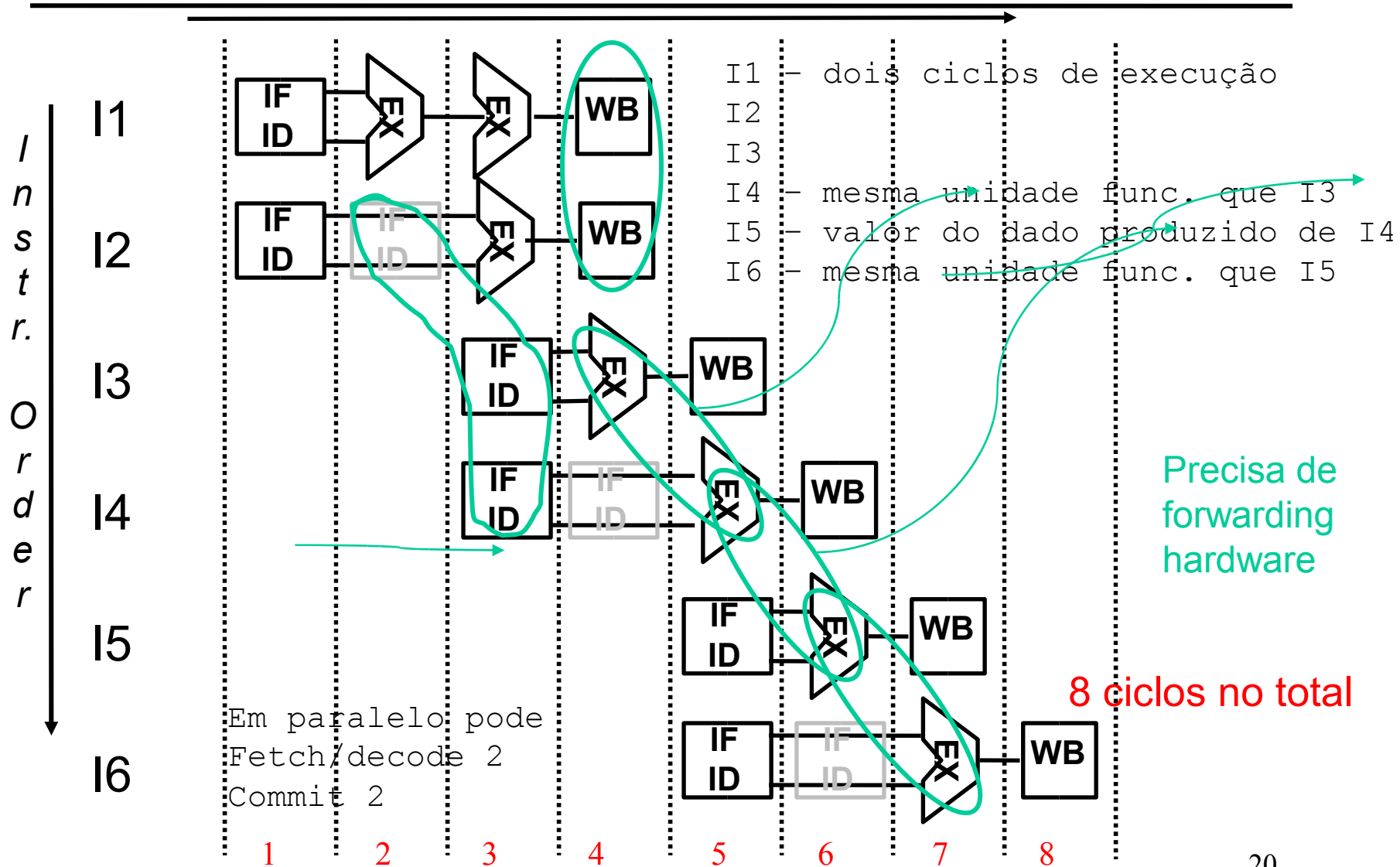
2. Despacho em ordem, Finalização em ordem

- **Despacho de novas instruções só é feito quando instruções anteriormente despachadas já foram executadas**
- **Despacho é congelado ...**
 - quando existe conflito por unidade funcional
 - quando unidade funcional exige mais de um ciclo para gerar resultado
- **Exemplo, supondo processador que pode a cada ciclo ...**
 - decodificar 2 instruções
 - executar até 3 instruções em 3 unidades funcionais distintas
 - escrever resultados de 2 instruções
- **In-Order Issue, In-Order Completion - IOI-IOC**

Despacho em ordem, Finalização em ordem



Despacho em ordem, Finalização em ordem



Despacho em ordem, Finalização em ordem

decodificação

I1	I2
I3	I4
I3	I4
	I4
I5	I6
	I6

execução

I1	I2	
I1		
		I3
		I4
	I5	
	I6	

write-back

I1	I2
I3	
	I4
I5	
	I6

ciclo

1
2
3
4
5
6
7
8

restrições:

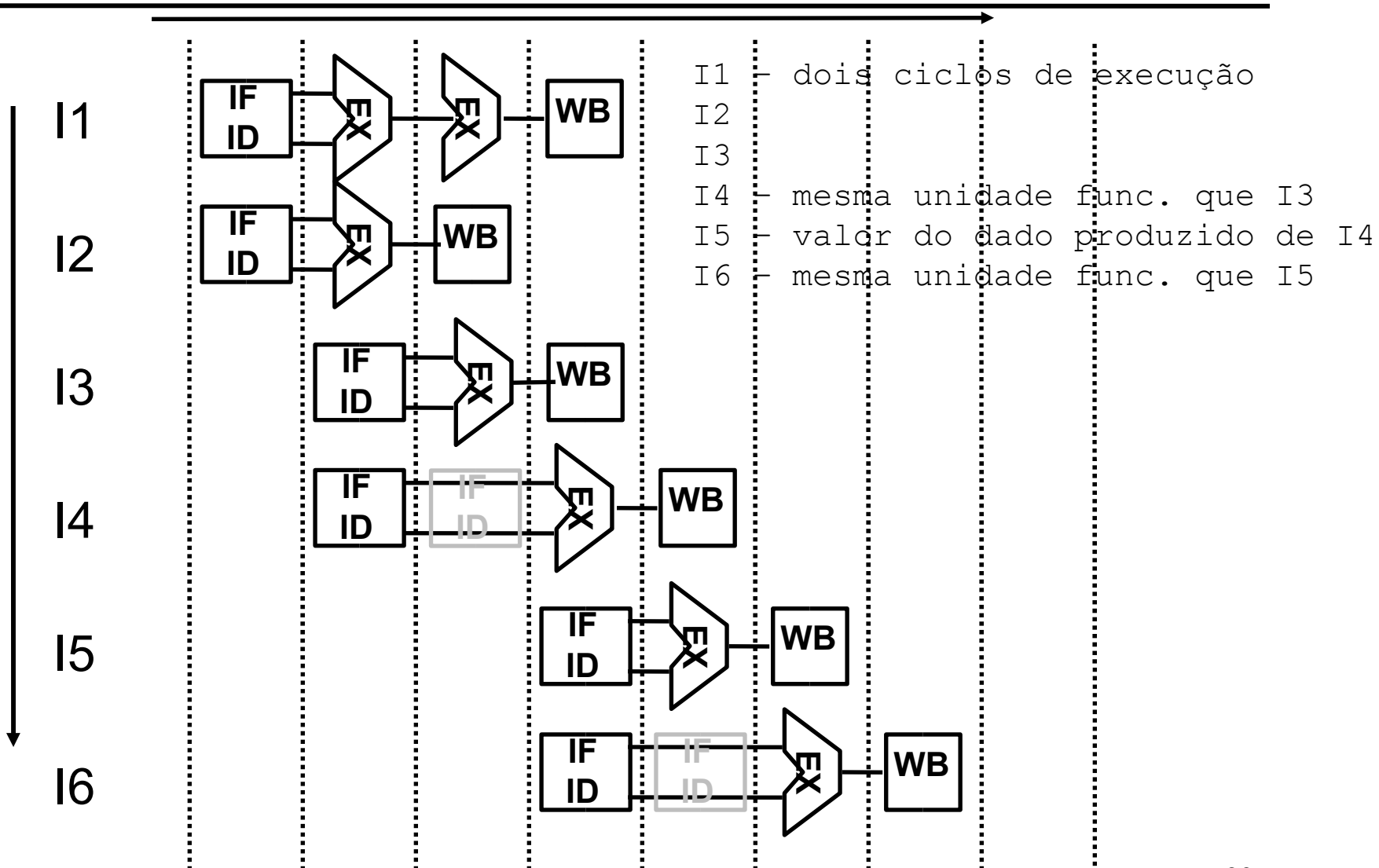
- fase de execução de I1 exige 2 ciclos
- I3 e I4 precisam da mesma unidade funcional
- I5 e I6 precisam da mesma unidade funcional
- I5 depende do valor produzido por I4

6 instruções em
6 ciclos
IPC = 1.0

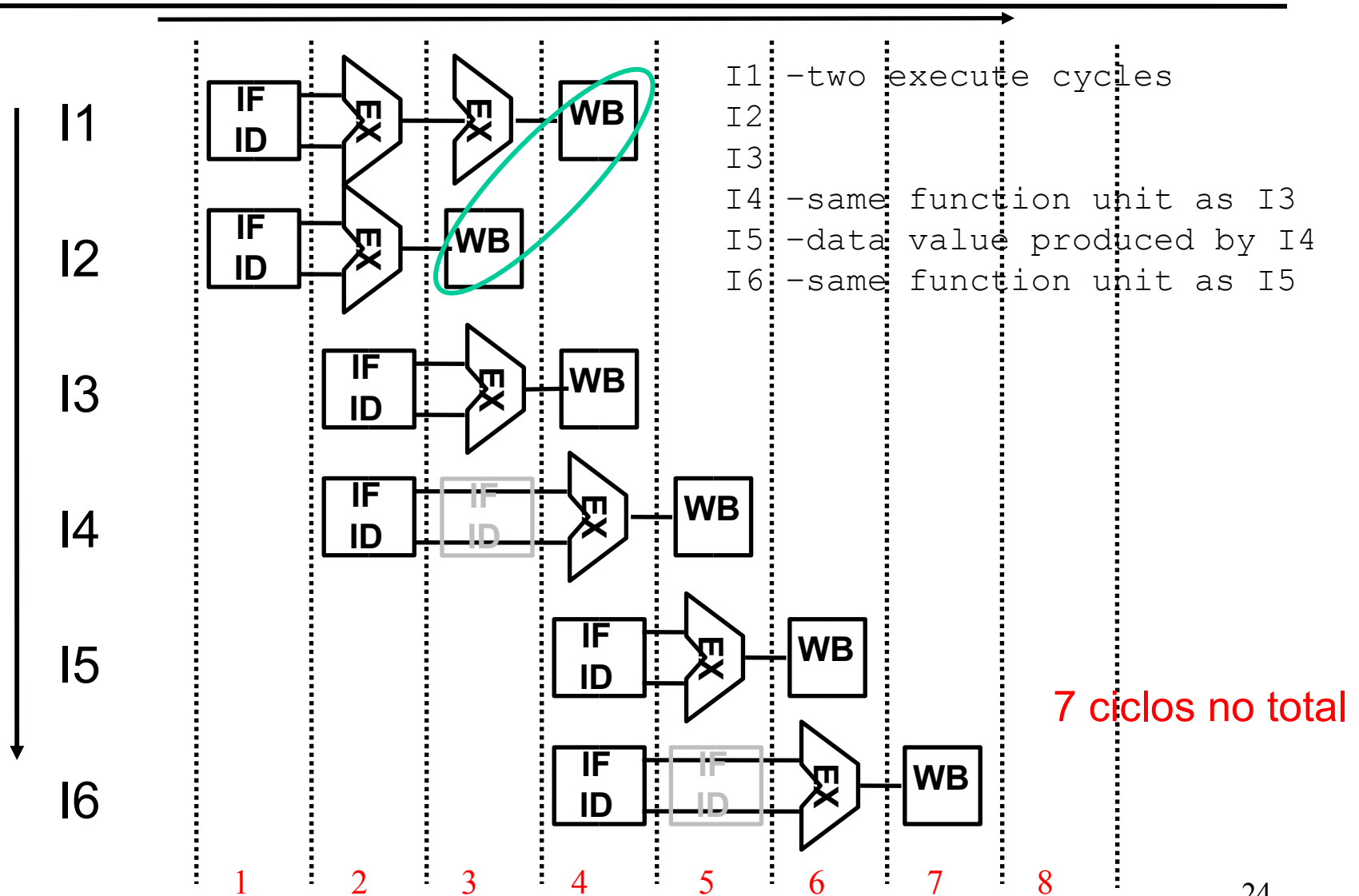
3. Despacho em ordem, Finalização fora de ordem

- **Despacho não espera que instruções anteriores já tenham sido executadas**
 - ou seja: despacho não é congelado quando unidades funcionais levam mais de um ciclo para executar instrução
- **Conseqüência: uma unidade funcional pode completar uma instrução após instruções subseqüentes já terem sido completadas**
- **Despacho ainda precisa ser congelado quando ...**
 - há conflito por uma unidade funcional
 - há uma dependência de dados verdadeira
- **In-Order Issue with Out-of-Order Completion – IOI-OOC**

Despacho em ordem, Finalização fora de ordem



Despacho em ordem, Finalização fora de ordem



Despacho em ordem, Finalização fora de ordem

decodificação

I1	I2
I3	I4
	I4
I5	I6
	I6

execução

I1	I2	
I1		I3
		I4
	I5	
	I6	

write-back

I2	
I1	I3
I4	
I5	
I6	

ciclo

1
2
3
4
5
6
7

6 instruções em
5 ciclos
IPC = 1.2

notar:

- I1 termina fora de ordem em relação a I2
- I3 é executada concorrentemente com último ciclo de execução de I1
- tempo total reduzido para 7 ciclos

Despacho em ordem, Finalização fora de ordem

- Supondo a seguinte situação

R3 := R3 op R5

R4 := R3 + 1

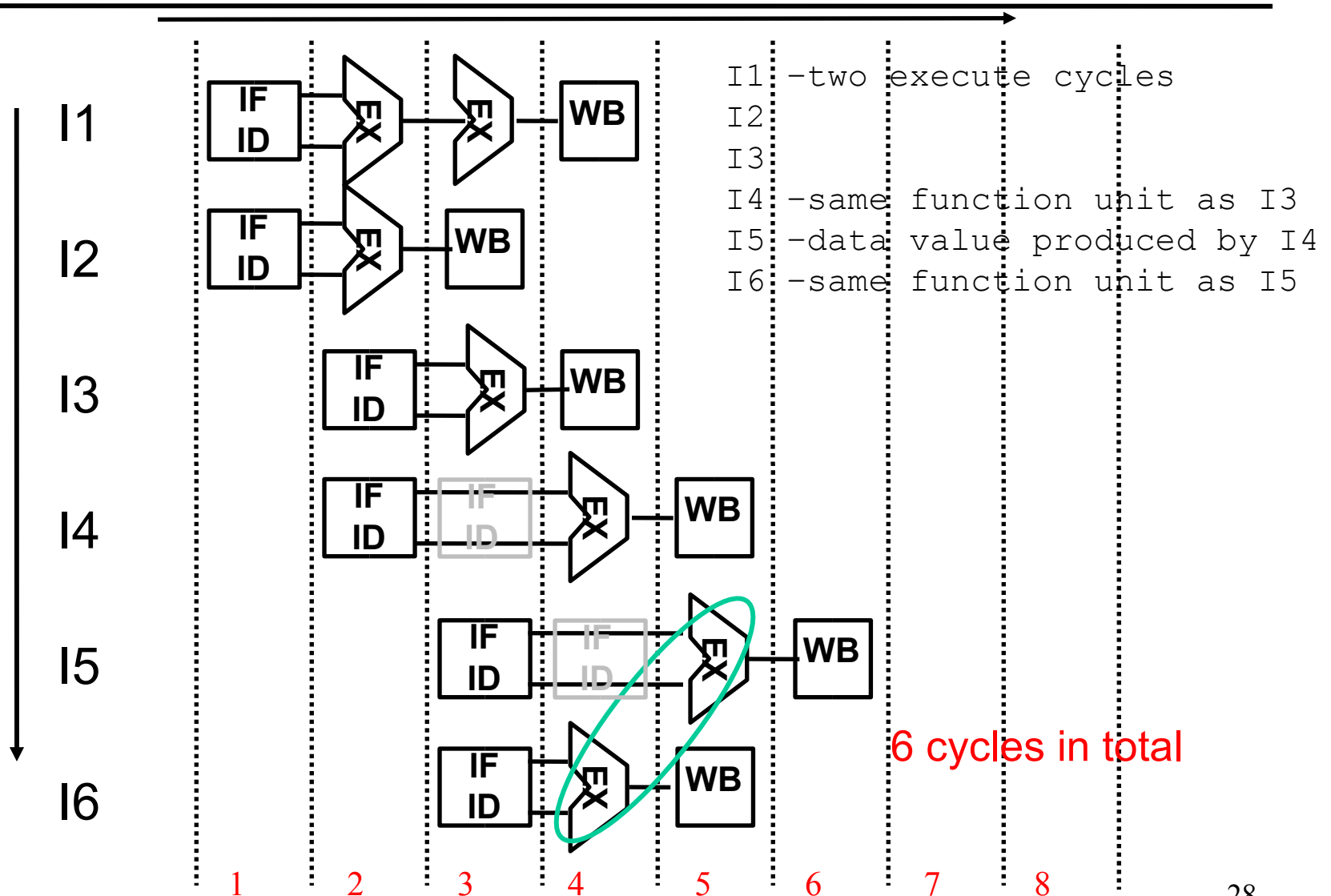
R3 := R5 + 1

- Dependência de saída
 - 1ª e 3ª instrução escrevem em R3
 - valor final de R3 deve ser o escrito pela 3ª instrução
 - atribuição da 1ª instrução não pode ser feita após atribuição da 3ª instrução
 - despacho da 3ª instrução precisa ser congelado
- Finalização fora de ordem ...
 - exige controle mais complexo para testar dependências de dados
 - torna mais difícil o tratamento de interrupções

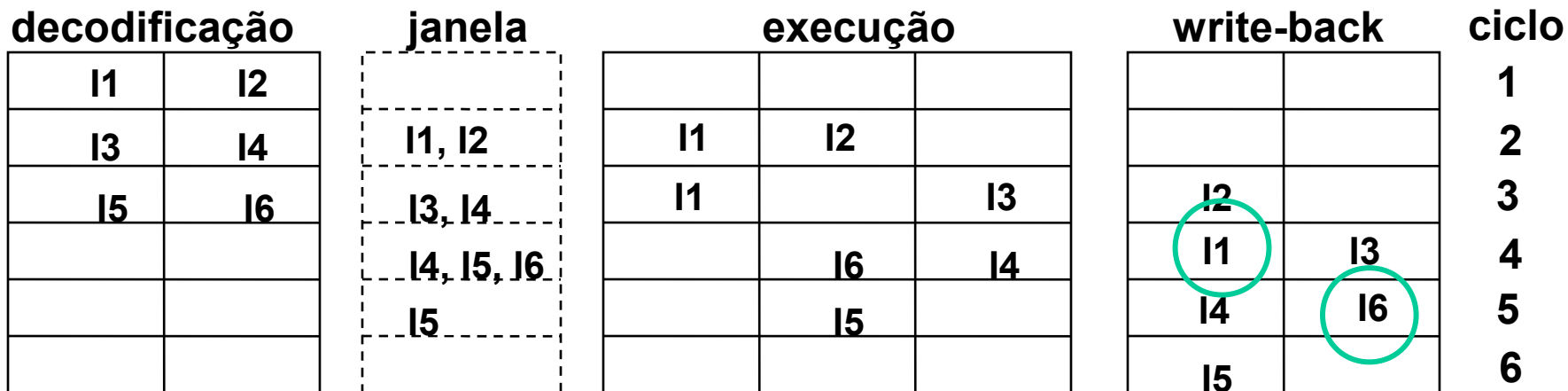
4. Despacho fora de ordem, Finalização fora de ordem

- **Problemas do despacho em ordem**
 - decodificação de instruções é congelada quando instrução cria ...
 - conflito de recurso
 - dependência verdadeira ou dependência de saída
 - consequência: processador não tem capacidade de *look-ahead* além da instrução que causou o problema, mesmo que haja instruções posteriores independentes
- **Solução**
 - isolar estágio de decodificação do estágio de execução
 - continuar buscando e decodificando instruções, mesmo que elas não possam ser executadas imediatamente
 - inclusão de um buffer entre os estágios de decodificação e execução:
janela de instruções
 - instruções são buscadas de janela independentemente de sua ordem de chegada: despacho fora de ordem
- **Out-of-Order Issue with Out-of-Order Completion - OOO-OOC**

Despacho fora de ordem, Finalização fora de ordem



Despacho fora de ordem, Finalização fora de ordem



6 instruções em
4 ciclos
IPC = 1.5

notar:

- estágio de decodificação opera a velocidade máxima, pois independe do estágio de execução
- I6 é independente e pode ser executada fora de ordem, concorrentemente com I4
- tempo total reduzido para 6 ciclos

Despacho fora de ordem, Finalização fora de ordem

- Supondo a seguinte situação

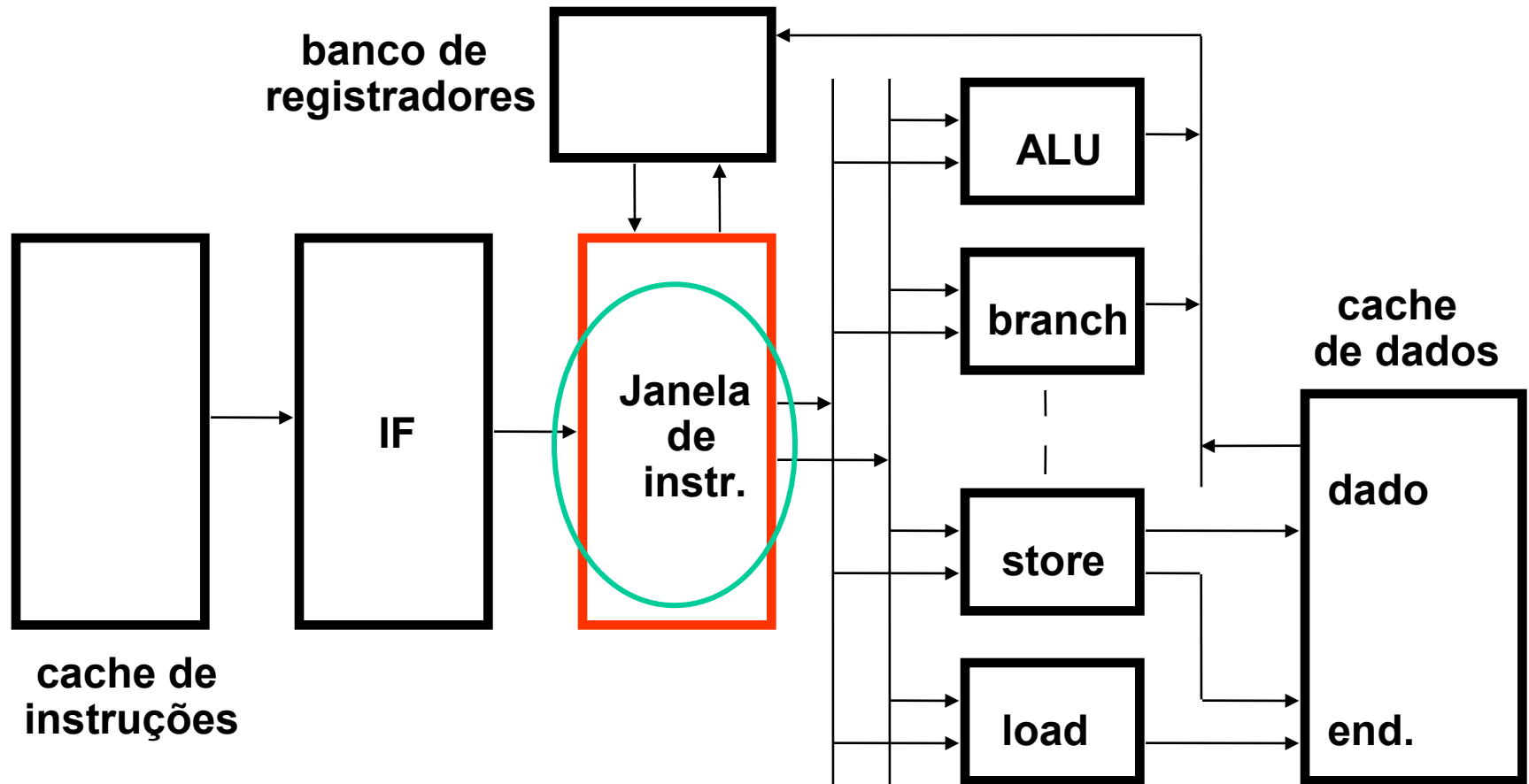
$R4 := R3 + 1$

$R3 := R5 + 1$

- Anti-dependência
 - 2ª instrução escreve em R3
 - 1ª instrução precisa ler valor de R3 antes que 2ª instrução escreva novo valor
 - despacho da 2ª instrução precisa ser congelado até que 1ª instrução tenha lido valor de R3

Janela de Instruções Centralizada

5. Janela de Instruções Centralizada



Janela de Instruções Centralizada

- **“Janela de instruções” é um buffer que armazena todas as instruções pendentes para execução**
- **Instrução enviada para unidade de execução correspondente quando operandos estão disponíveis**
 - **operandos buscados no banco de registradores**
- **Se operando não está disponível, identificador de registrador é colocado na instrução**
 - **quando instrução atualiza este registrador, janela de instruções é pesquisada associativamente e identificador do registrador é substituído pelo valor do operando**

Janela de Instruções Centralizada

instr.	código	registr. destino	oper. 1	reg.1	oper. 2	reg.2
1	operação	ID	valor			ID
2	operação	ID	valor			ID
3	operação	ID		ID	valor	
4	operação	ID	valor		valor	
5	operação	ID		ID		ID

FIM