

Organização e Arquitetura de Computadores

Capítulo 13

Paralelismo no nível de instruções

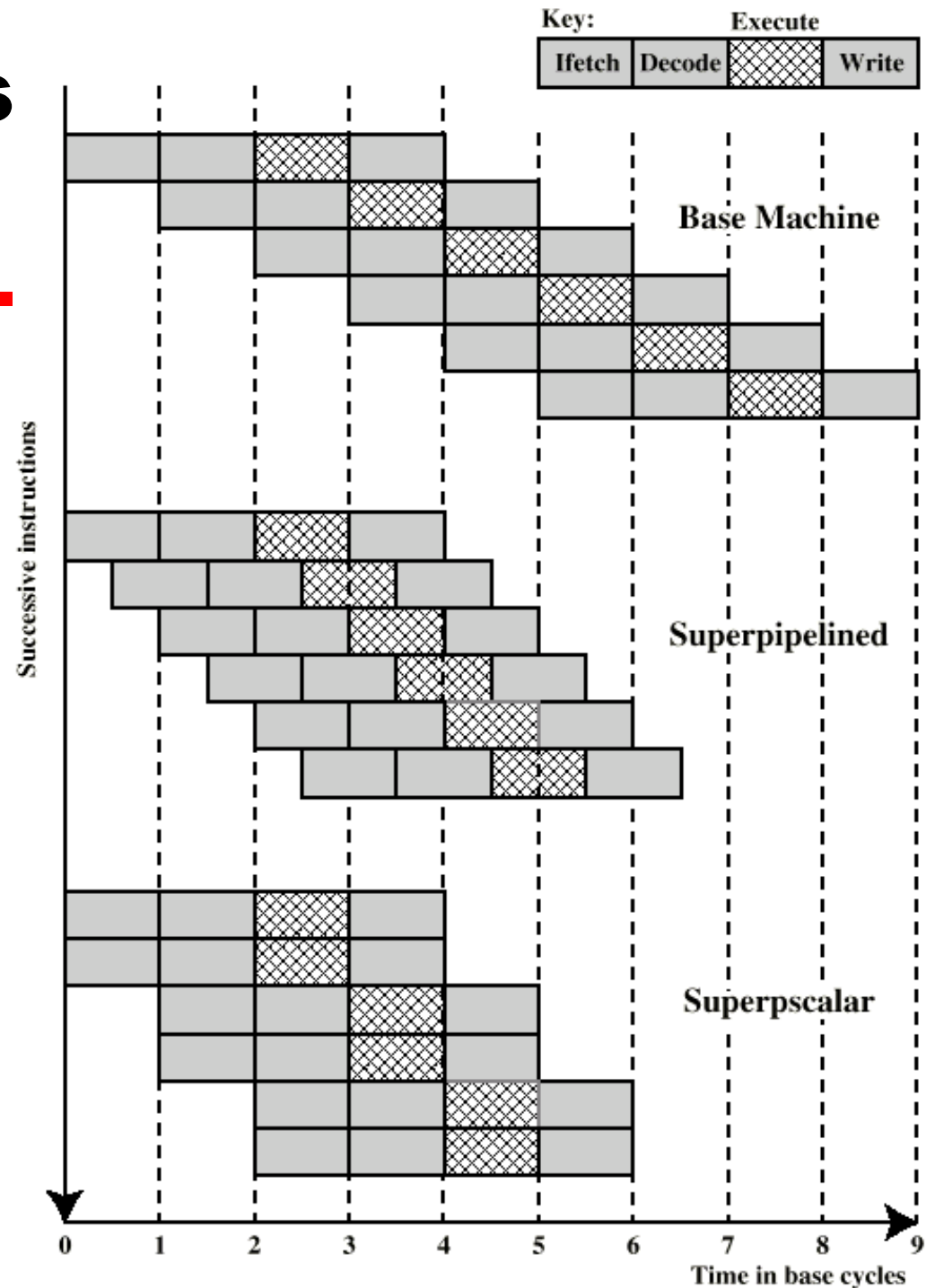
O que é Superescalar?

- ❑ As arquiteturas superescalares são aquelas capazes de buscar, decodificar, executar e terminar mais de uma instrução por ciclo de máquina
- ❑ São organizadas internamente como múltiplos pipelines, mas que devem trabalhar cooperativamente para que as dependências de dados entre as instruções sejam respeitadas

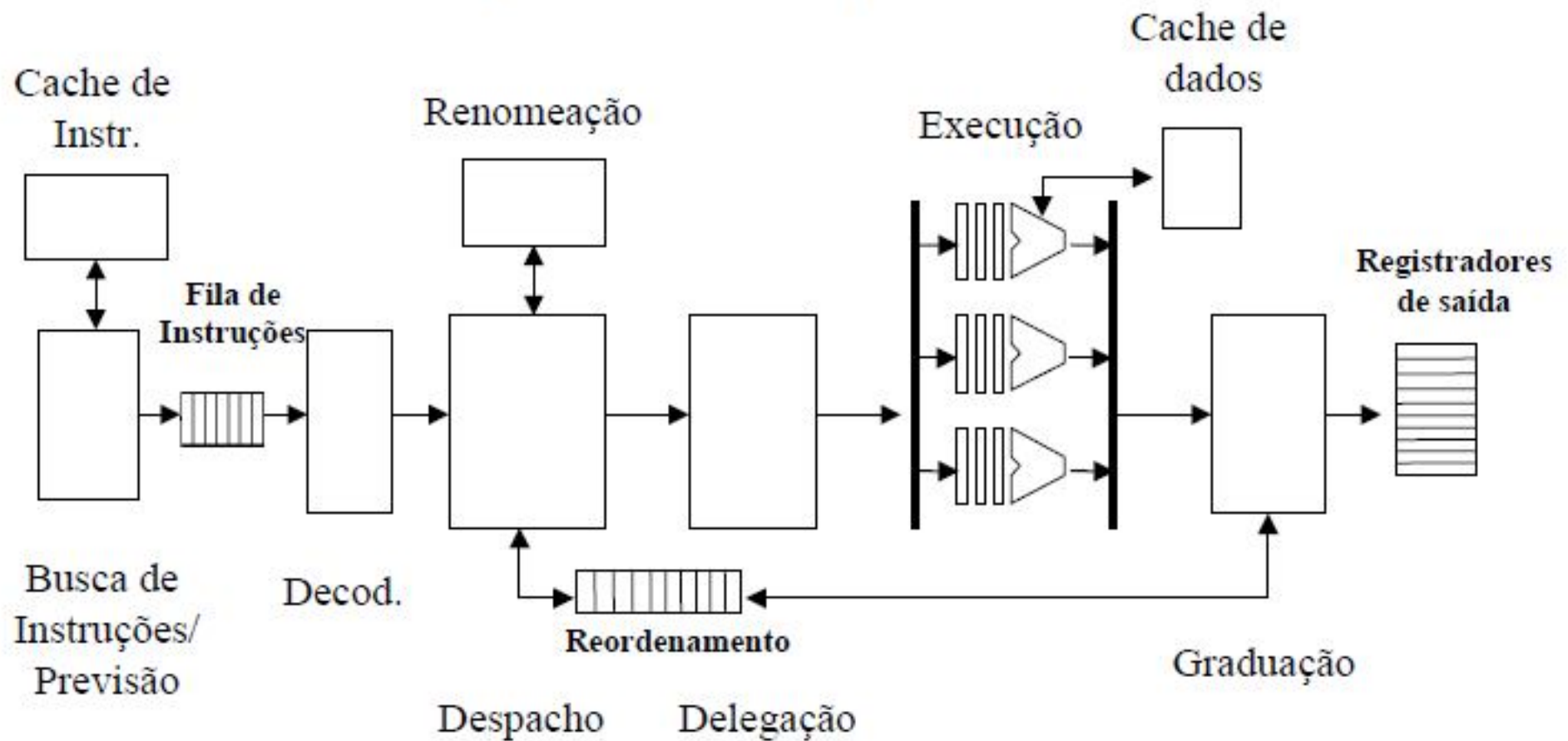
Superpipelined

- ❑ Muitos estágios do pipeline necessitam menos do que a metade de um clock
- ❑ Clocks internos operando em clock duplicado permite que duas tarefas sejam feitas para um ciclo de clock externo
- ❑ Superscalar permite os estágios de busca e execução serem paralelos.

Superscalar vs Superpipeline



Arquitetura Superescalar



Limitações

- ❑ Paralelismo a nível de instrução
 - Otimização a nível de compilação
 - Técnicas de hardware
- ❑ Limitado pela
 1. Dependência verdadeira de dados
 2. Dependência de desvio
 3. Conflito de recursos
 4. Dependência de saída
 5. Antidependência

Dependência Verdadeira de Dados

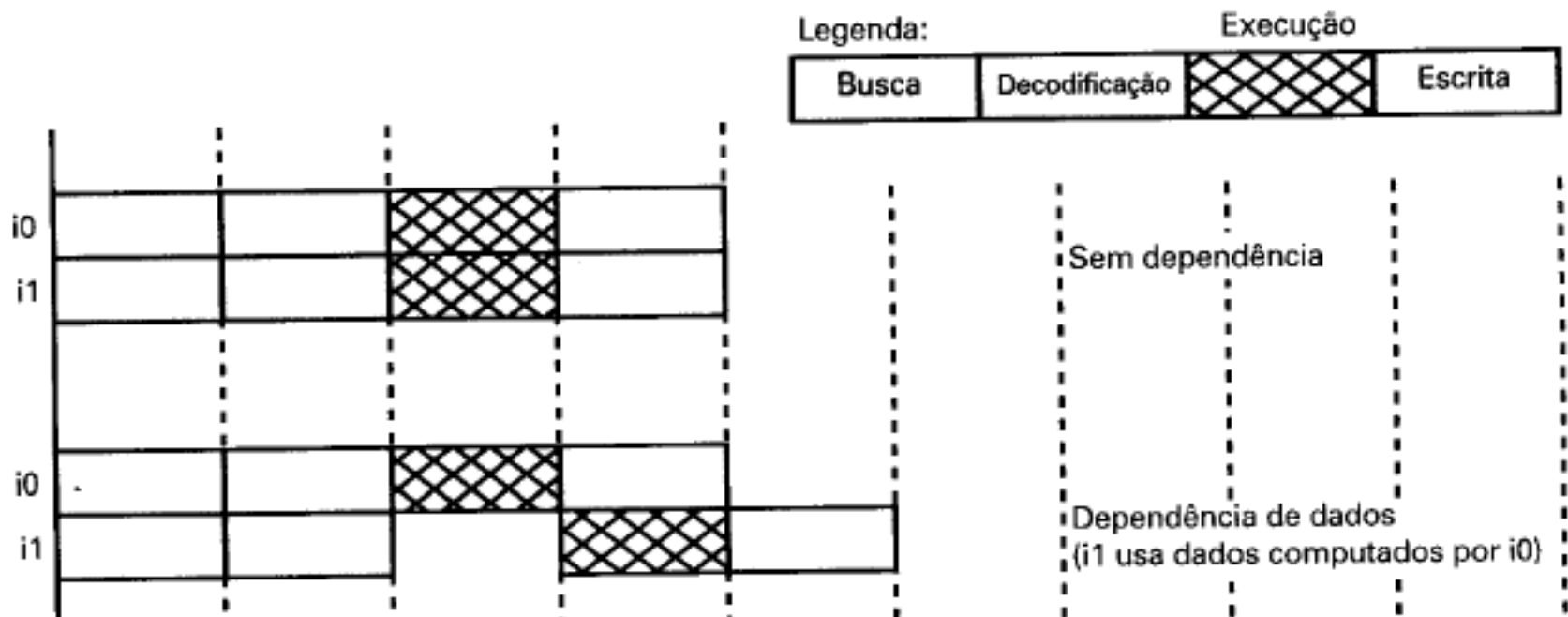
- ❑ Uma instrução utiliza um operando que é produzido p/ uma instrução anterior

ADD **r1**, r2 (**r1** := $r1 + r2$;))

MOVE r3, **r1** ($r3 := \mathbf{r1}$;))

- ❑ Pode-se buscar e decodificar a segunda instrução em paralelo com a primeira
- ❑ Não pode executar a segunda instrução até que a primeira tenha terminado

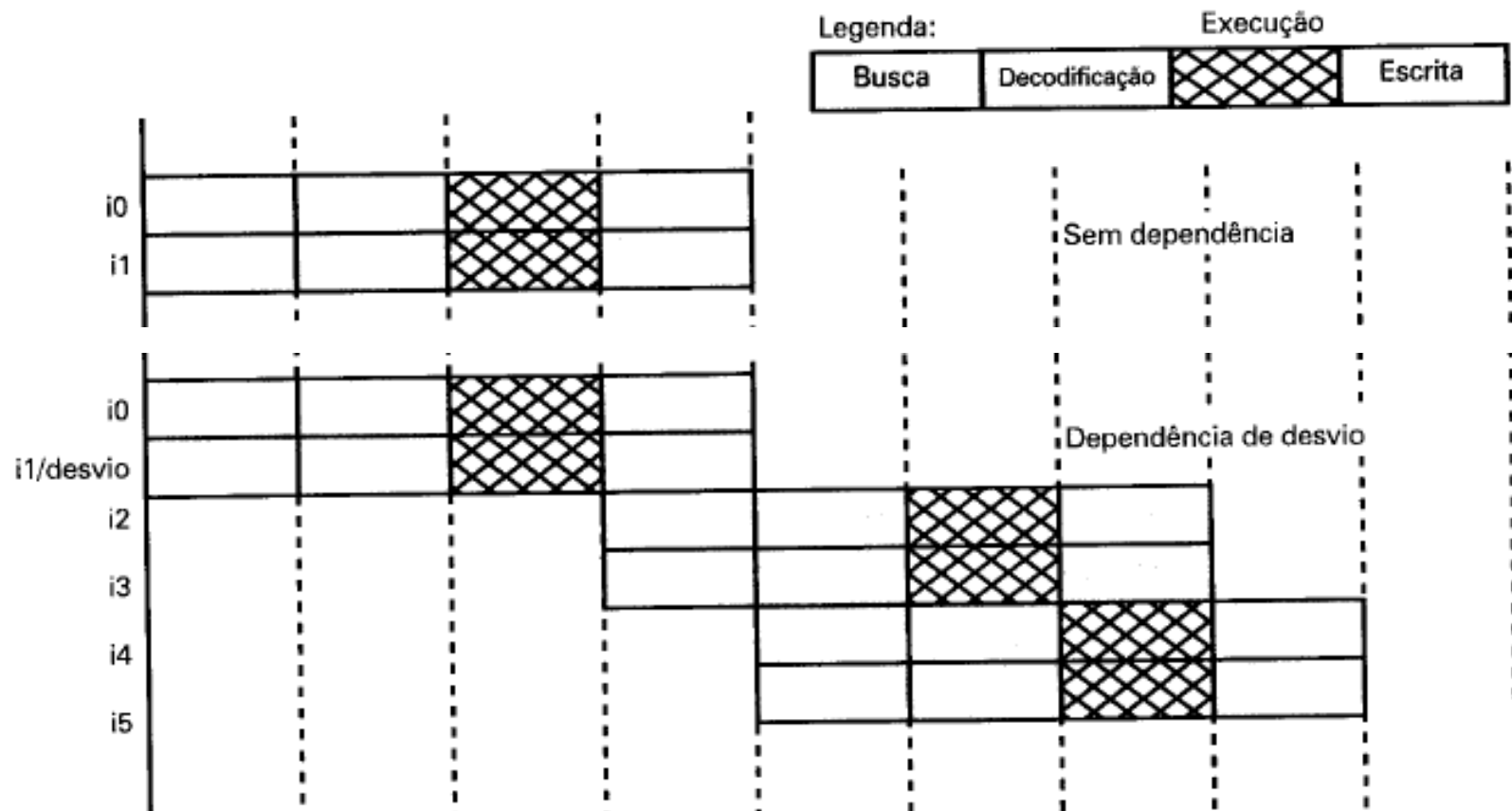
Dependência Verdadeira de Dados



Uma instrução deve ser atrasada até que todos os dados sejam produzidos

Dependência de desvio

- ❑ Não se pode executar instruções depois de um desvio em paralelo com as instruções antes do desvio

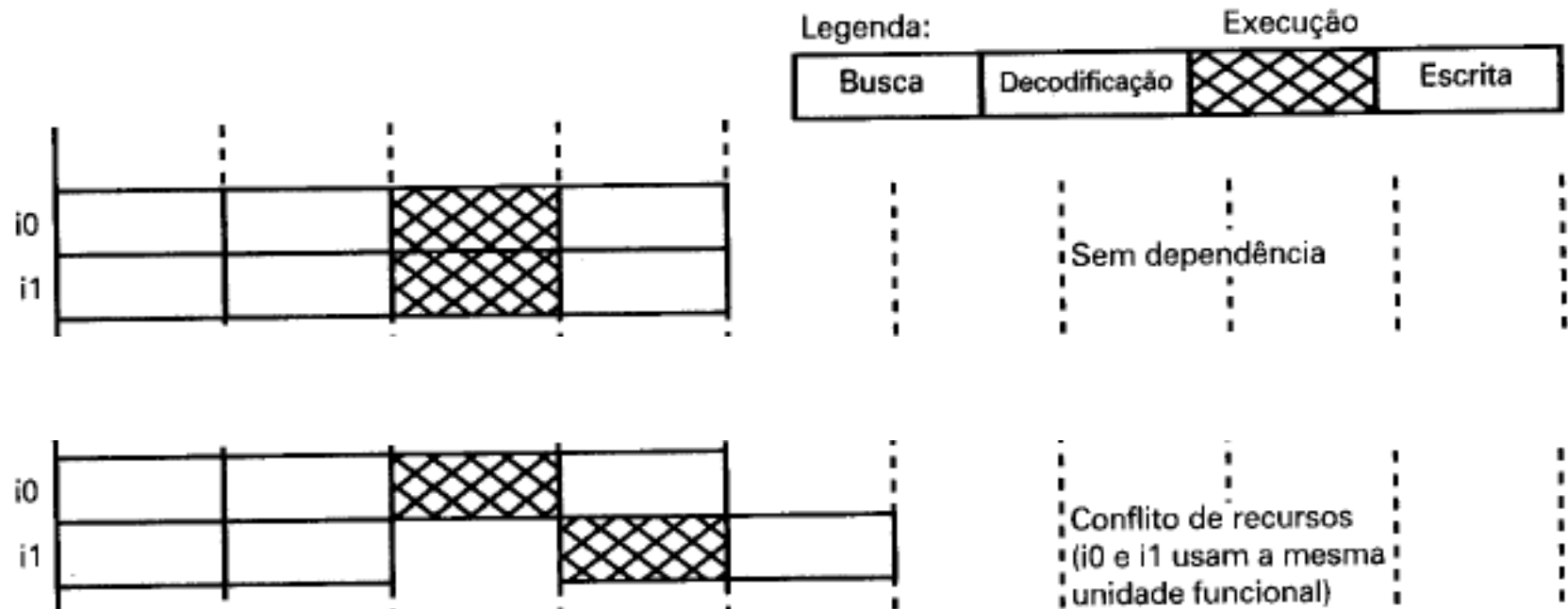


Conflito de recursos

- ❑ Duas ou mais instruções que necessitam usar o mesmo recurso ao mesmo tempo
 - Acesso concorrente à memória
 - Acesso concorrente a um banco de registradores
 - Uso simultâneo de unidades funcionais

- ❑ Recursos podem ser duplicados

Conflito de recursos



Dependência de Saída

- ❑ Ocorre quando duas ou mais instruções estão escrevendo em um mesmo local

R3 := R3 + R5; (I1)

R3 := R5 + 1; (I2)

R4 := R3 + 1; (I3)

- ❑ Existe uma dependência de saída entre I1 e I2
- ❑ Não podem executar em conjunto pois pode haver inconsistência no resultado

Antidependência

- ❑ Uma instrução usa um operando enquanto a seguinte está escrevendo no mesmo local
- ❑ Dependência de leitura-escrita
 - $R3 := R3 + R5;$ (I1)
 - $R4 := \textcolor{red}{R3} + 1;$ (I2)
 - $\textcolor{red}{R3} := R5 + 1;$ (I3)
 - $R7 := R3 + R4;$ (I4)
- ❑ I3 não pode ser completada antes que I2 comece como I2 precisa de um valor em R3 e I3 muda o R3

Necessidades de projeto

- ❑ Paralelismo a nível de instrução
 - Instruções numa seqüência são independentes
 - A execução pode ser sobreposta
 - Dirigido pela dependência de dados e de procedimentos
- ❑ Paralelismo de máquina
 - Habilidade de tirar vantagem do paralelismo a nível de instrução.
 - Dirigido por um conjunto de pipelines paralelas

Política de inicialização de instruções

- ❑ Processador deve identificar paralelismo e coordenar a execução de instruções em paralelo

- ❑ Ordenação de instruções:
 - Ordem na qual as instruções são buscadas
 - Ordem na qual as instruções são executadas
 - Ordem na qual as instruções atualizam registradores e memória

Política de inicialização de instruções

- ❑ Processador deve ajustar dependências e conflitos

- ❑ Políticas de inicialização:
 - Iniciação em ordem, com finalização em ordem
 - Iniciação em ordem, com finalização fora de ordem
 - Iniciação fora de ordem, com finalização fora de ordem

Inicialização em ordem

Finalização em ordem

- ❑ A emissão de instrução na ordem que elas ocorrem (sequencial)
- ❑ Não é muito eficiente
- ❑ Pode buscar mais de uma instrução

Inicialização em ordem

Finalização em ordem

Decodificação	Execução	Escrita	Ciclo
I1			1
I3	I1		2
I3			3
		I1	4
I5			5
	I5	I3	6
	I6		7
		I5	8

(a) Iniciação em ordem e terminação em ordem

- I1 requer dois ciclos para ser executada
- I3 e I4 competem pela mesma unidade funcional
- I5 depende do valor produzido por I4
- I5 e I6 competem por uma unidade funcional

Inicialização em ordem

Finalização fora de ordem

Decodificação	Execução	Escrita	Ciclo
I1			1
I3	I1		2
	I2	I2	3
I5		I1	4
		I3	5
	I5	I4	6
	I6	I5	7
		I6	

(b) Iniciação em ordem e terminação fora de ordem

- I1 requer dois ciclos para ser executada
- I3 e I4 competem pela mesma unidade funcional
- I5 depende do valor produzido por I4
- I5 e I6 competem por uma unidade funcional

Inicialização fora de ordem

Finalização fora de ordem

- ❑ Desacopla o estágio de decodificação do pipeline do estágio de execução
- ❑ Pode continuar a buscar e decodifica até que o pipeline fique cheio
 - **Janela de instruções**
- ❑ Quando uma unidade funcional se torna disponível, uma instrução pode ser executada
- ❑ Como as instruções foram decodificadas, o processador pode examinar instruções à frente

Inicialização fora de ordem

Finalização fora de ordem

Decodificação

I1	I2
I3	I4
I5	I6

Janela

<i>I1, I2</i>
<i>I3, I4</i>
<i>I4, I5, I6</i>
<i>I5</i>

Execução

I1	I2	
I1		I3
	I6	I4
	I5	

Escrita

I2	
I1	I3
I4	I6
I5	

Ciclo

1
2
3
4
5
6

Renomeação de registradores

- ❑ Registradores alocados dinamicamente pelo HW do processador
- ❑ Quando um novo valor é produzido, um novo registrador é criado para este valor
- ❑ Referências a um mesmo registrador pode ser convertido para registradores reais distintos

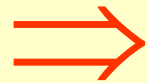
Renomeação de registradores

$r3 := r3 + r5$

$r4 := r3 + 1$

$r3 := r5 + 1$

$r7 := r3 - r4$

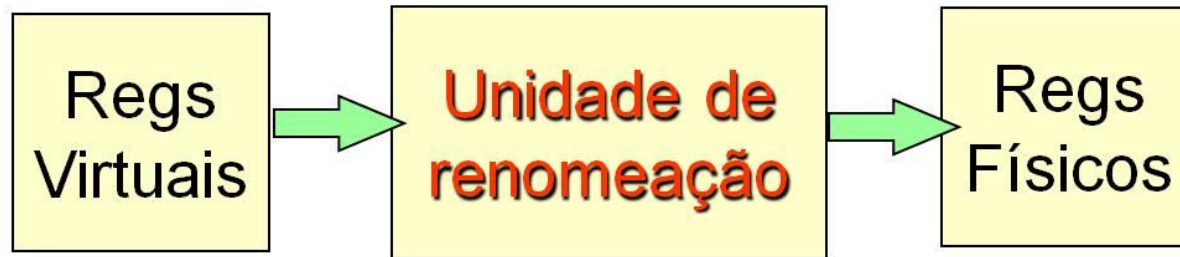


$r3b := r3a + r5a$

$r4b := r3b + 1$

$r3c := r5a + 1$

$r7a := r3c - r4b$



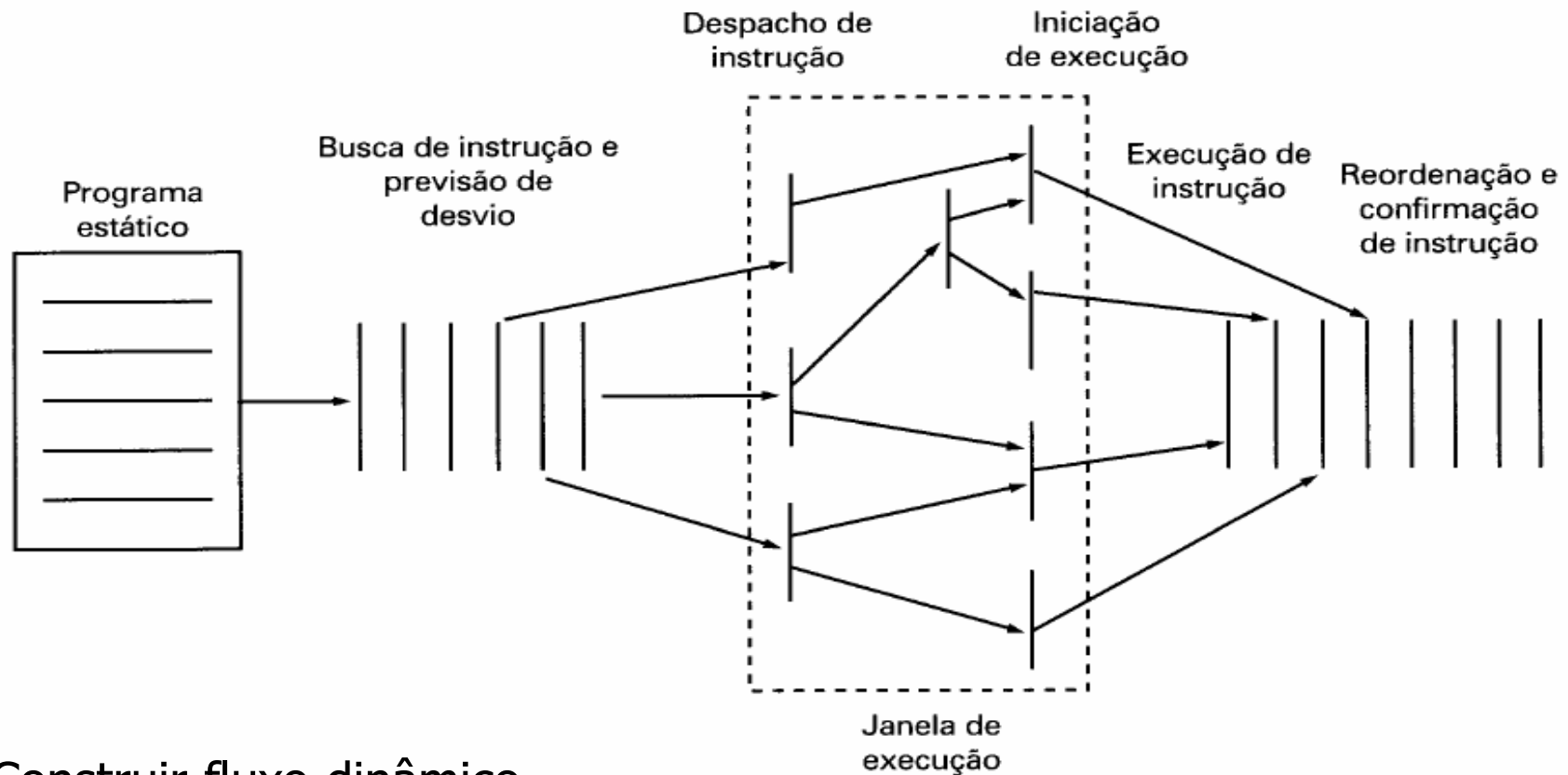
Renomeação de registradores

$r3 := r3 + r5$	$r3b := r3a + r5a$
$r4 := r3 + 1$	$r4b := r3b + 1$
$r3 := r5 + 1$	$r3c := r5a + 1$
$r7 := r3 - r4$	$r7a := r3c - r4b$

Referências originais a um mesmo registrador são convertidas em referências reais a registradores distintos

- Antidependência: I2, I3 (resolvido)
- Dependência de saída: I1, I3 (resolvido)
- I3 pode ser iniciada imediatamente após renomeação

Execução Superscalar



- Construir fluxo dinâmico
- Despachar instruções para a janela de instruções
- Ordenar por dependências de dados verdadeiras
- Execução paralela das instruções
- Realocação das instruções na ordem sequencial e registrar resultados

Implementação Superscalar

- ❑ Busca simultaneamente múltiplas instruções
- ❑ Lógica para determinar dependências verdadeiras envolvendo valores de registradores
- ❑ Mecanismos para comunicar esses valores
- ❑ Mecanismos para iniciar múltiplas instruções em paralelo
- ❑ Recursos para execução paralela de múltiplas instruções
- ❑ Mecanismos para garantir o estado de término de processos na ordem correta

Very Long Instruction Word

- ❑ O compilador descobre as instruções que podem ser executadas em paralelo e agrupa-as formando uma longa instrução que será despachada para a máquina

Exercícios

❑ 13.5

❑ 13.6

1. Explique o que é um processador superescalar.
2. Quais são os principais problemas com este tipo de arquitetura? Explique cada um deles.
3. Explique as políticas de inicialização de instruções.
4. No que consiste a renomeação de registradores?

Exercícios

- ❑ Identificar as situações de dependência na seguinte sequência de código, do MIPS64:

DIVD F1, F3, F5

ADDD F4, F1, F9

SD F4, 0(R1)

SUBD F1, F10, F14

MULD F9, F10, F8