

ARQUITETURA DE COMPUTADORES

Superescalaridade

1

Introdução

- ❑ Princípios da superescalaridade
 - várias instruções completadas simultaneamente em cada ciclo de relógio
 - várias unidades de execução
- ❑ Hardware é responsável pela extração de paralelismo
- ❑ Término das instruções pode não seguir a seqüência estabelecida no programa
- ❑ 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)

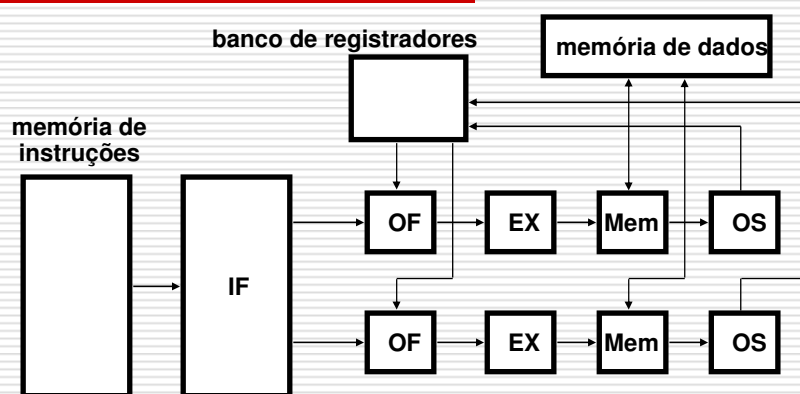
2

Introdução

- ❑ 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
- ❑ Na prática, obtém-se IPC (Instruções Por Ciclo) pouco maior do que 2
 - limitação do paralelismo intrínseco dos programas

3

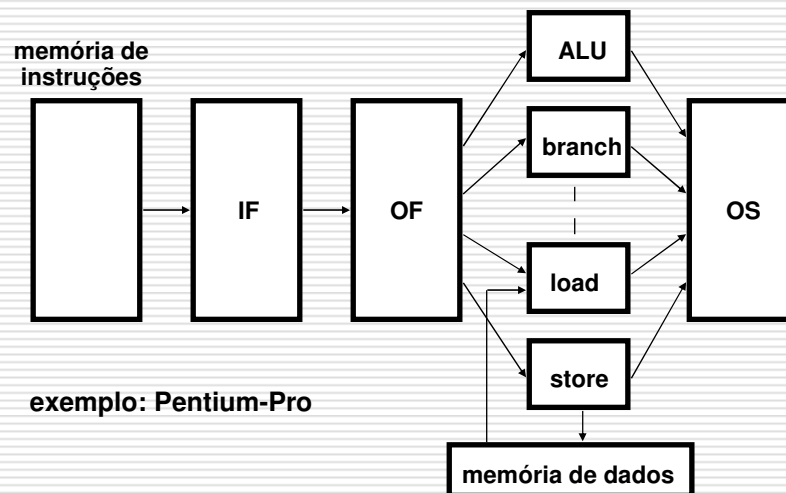
Processador com 2 pipelines



exemplo: Pentium I
Memória de instruções precisa fornecer dobro de instruções por ciclo

4

Unidades de execução especializadas



5

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, terminação em ordem
 - despacho em ordem, terminação fora de ordem
 - despacho fora de ordem, terminação fora de ordem

6

Despacho em ordem, terminação em ordem

- ❑ 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

7

Despacho em ordem, terminação em ordem

decodificação		execução			write-back		ciclo
I1	I2						1
I3	I4	I1	I2				2
I3	I4	I1					3
	I4			I3	I1	I2	4
I5	I6			I4	I3		5
	I6		I5			I4	6
			I6		I5		7
						I6	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 (exec.)
IPC = 1.0

IPC – Instruções Por Ciclo

8

Despacho em ordem, terminação fora de ordem

- ❑ 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

9

Despacho em ordem, terminação fora de ordem

decodificação		execução			write-back		ciclo
I1	I2						1
I3	I4	I1	I2				2
	I4	I1		I3	I2		3
I5	I6			I4	I1	I3	4
	I6		I5		I4		5
			I6		I5		6
					I6		7

6 instruções em
5 ciclos (exec.)
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

10

Despacho em ordem, terminação fora de ordem

- ❑ Supondo a seguinte situação
 - $R3 := R3 + R5$
 - $R4 := R3 + 1$
 - $R3 := R5 + 1$
- ❑ 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
- ❑ Terminação fora de ordem ...
 - exige controle mais complexo para testar dependências de dados
 - torna mais difícil o tratamento de interrupções

11

Despacho fora de ordem, terminação 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
 - Processador não tem capacidade de *look-ahead* além da instrução que causou o problema, mesmo que haja instruções posteriores independentes

12

Despacho fora de ordem, terminação fora de ordem

□ 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 da janela independentemente de sua ordem de chegada: **despacho fora de ordem**

13

Despacho fora de ordem, terminação fora de ordem

decodificação		janela	execução			write-back		ciclo
I1	I2							1
I3	I4	I1, I2	I1	I2				2
I5	I6	I3, I4	I1		I3	I2		3
		I4, I5, I6		I6	I4	I1	I3	4
		I5		I5		I4	I6	5
						I5		6

Não é um estágio adicional

6 instruções em
4 ciclos (exec.)
IPC = 1.5

notar:

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

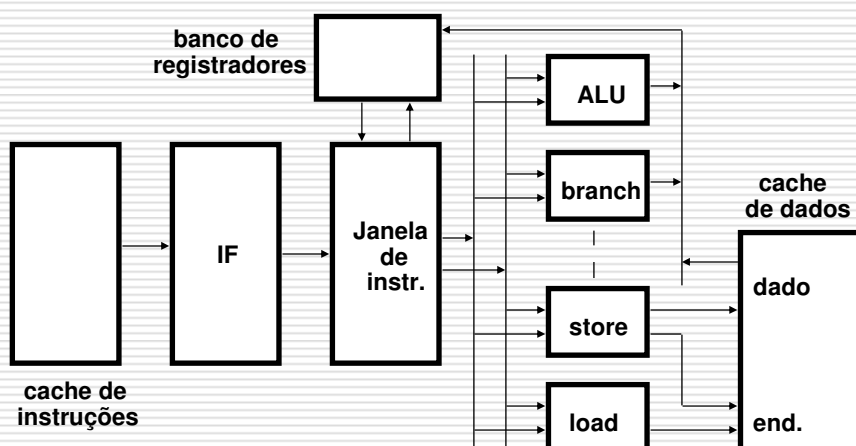
14

Despacho fora de ordem, terminação 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

15

Janela de instruções centralizada



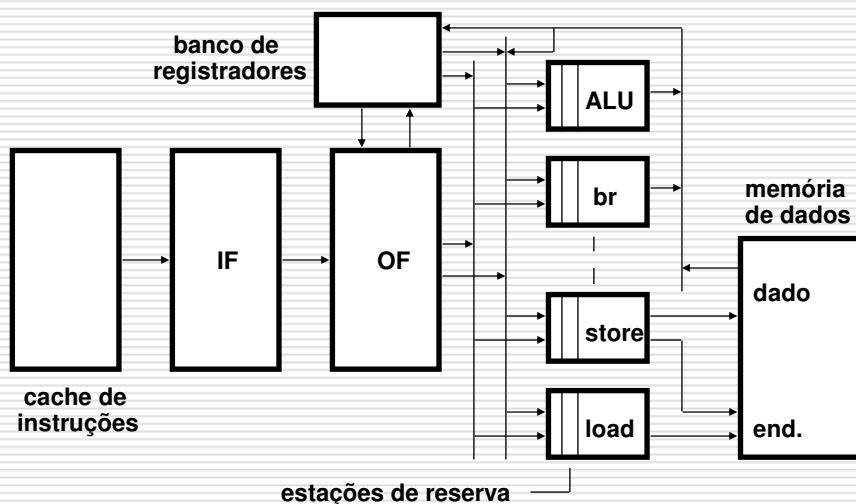
16

Janela de instruções centralizada

- ❑ “Janela de instruções” é um buffer que armazena todas as instruções pendentes para execução
- ❑ Instrução é enviada para unidade de execução correspondente quando operandos estão disponíveis
 - operandos são 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

17

Janela de instruções distribuída



18

Exemplo

- ❑ Supondo um processador superescalar com a seguinte configuração:
 - 4 unidades funcionais - 2 somadores, 1 multiplicador, 1 load/store
 - pode executar 4 instruções por ciclo em cada estágio do pipeline
 - latências
 - ❑ somador - 1 ciclo
 - ❑ multiplicador - 2 ciclos
 - ❑ load/store - 2 ciclos
- ❑ Deve ser executado o seguinte programa:

```

ADD  R1, R2, R3
LW   R10, 100 (R5)
ADD  R5, R1, R6
MUL  R7, R4, R8
ADD  R2, R7, R3
ADD  R9, R4, R10
ADD  R11, R4, R6
    
```

19

Exemplo

ciclo	somador 1	somador 2	multiplicador	load/store
1	R1 = R2 + R3	R11 = R4 + R6	R7 = R4 * R8	R10 = mem (R5+100)
2	R5 = R1 + R6		R7 = R4 * R8	R10 = mem (R5+100)
3	R9 = R4 + R10	R2 = R7 + R3		

- ▶ ADD R1, R2, R3 ✓
 - ▶ LW R10, 100 (R5) ✓
 - ▶ ADD R5, R1, R6 ✓
 - ▶ MUL R7, R4, R8 ✓
 - ▶ ADD R2, R7, R3
 - ▶ ADD R9, R4, R10
 - ▶ ADD R11, R4, R6 ✓
- dependências verdadeiras
→ dependências falsas

20

Renomeação de registradores

- ❑ Antidependências e dependências de saída são causadas pela reutilização de registradores
- ❑ Efeito destas dependências pode ser reduzido pelo aumento do número de registradores ou pela utilização de outros registradores disponíveis

21

Renomeação de registradores

- ❑ Exemplo
 - $\text{ADD } R1, R2, R3$; $R1 = R2 + R3$
 - $\text{ADD } R2, R1, 1$; $R2 = R1 + 1$
antidependência em R2
 - $\text{ADD } R1, R4, R5$; $R1 = R4 + R5$
dependência de saída em R1
- ❑ Utilizando 2 outros registradores R6 e R7 pode-se eliminar as dependências falsas
 - $\text{ADD } R1, R2, R3$; $R1 = R2 + R3$
 - $\text{ADD } R6, R1, 1$; $R6 = R1 + 1$
 - $\text{ADD } R7, R4, R5$; $R7 = R4 + R5$

22

Renomeação de registradores

Limitações

- ❑ Não é possível criar número ilimitado de registradores
- ❑ Arquitetura deve manter compatibilidade quanto aos registradores visíveis para o programador

23

Renomeação de registradores

- ❑ Solução
 - Utilizar banco de registradores interno, bem maior do que o banco visível
 - Renomear registradores temporariamente
 - Cada registrador visível que é escrito numa instrução é renomeado para um registrador interno escolhido dinamicamente
- ❑ No exemplo anterior, supondo registradores internos Ra, Rb, Rc, Rd, Re, Rf, Rg
 - ADD Ra, Rb, Rc
 - ADD Rd, Ra, 1
 - ADD Re, Rf, Rg
- ❑ Antidependência e dependência de saída foram eliminadas

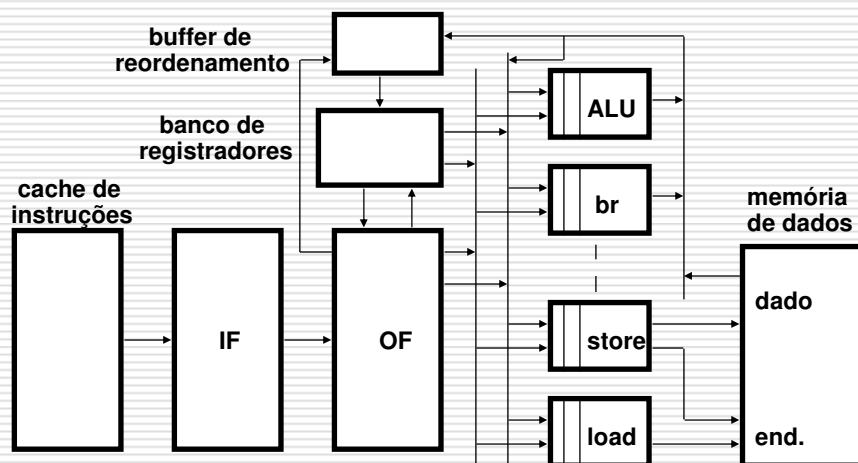
24

Buffer de reordenamento

- ❑ Buffer é organizado como FIFO
- ❑ Quando decodifica-se instrução que escreve em registrador, uma posição do buffer é alocada para o resultado
- ❑ Cada posição do buffer contém
 - Número do registrador original
 - Campo para armazenamento do resultado
 - Tag de renomeação
- ❑ Quando resultado está disponível, valor é escrito no buffer
 - Valor é simultaneamente enviado para estações de reserva e substitui tag de renomeação correspondente, se encontrado

25

Buffer de reordenamento



26

Literatura

- *Stallings – cap 13*
- *Patterson – cap 6.9, 6.10*

27

Resumo

- Superescalaridade
 - Várias instruções completadas simultaneamente
 - Várias unidades de execução
- Despacho e finalização de instruções fora de ordem
- Renomeação de registradores

28

O que mais preocupa não é o grito dos violentos, nem dos corruptos, nem dos desonestos, nem dos sem ética. O que mais preocupa é o silêncio dos bons.

Martin Luther King

29

Questões adicionais...

- ❑ O que significa *Loop Unrolling*?
- ❑ Como esta técnica pode influenciar o desempenho de um processador superescalar?

30

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

31

Janela de instruções distribuída

- ❑ Cada unidade de execução tem uma "estação de reserva"
 - Estação tem capacidade para armazenar 2 a 6 instruções
- ❑ Instruções são decodificadas e enviadas para a estação de reserva apropriada
- ❑ Instruções são enviadas para unidade de execução quando operandos estão disponíveis
- ❑ Mesmo mecanismo de identificação de registradores nas instruções
- ❑ Quando registradores são atualizados, valores são passados diretamente para as estações de reserva
 - Busca associativa para substituição de identificadores por valores

32

Grandes realizações são
possíveis quando se dá
importância aos pequenos
começos.

Lao Tzu