



Arquitetura de Computadores

Paralelismo no nível de
Instruções e
Processadores Superescalares

Profa. Gisele S. Craveiro

0 0 0

O que é Superescalar?

- Instruções usuais (aritméticas, load/store, desvio condicional) podem ser iniciadas e executadas independentemente
- Aplicável igualmente tanto a RISC & CISC
- Na prática mais comum em RISC

0 0 0

Porque Superescalar?

- o Maioria das operações são sobre grandezas escalares
- o Melhoria dessas operações leva a uma melhoria geral

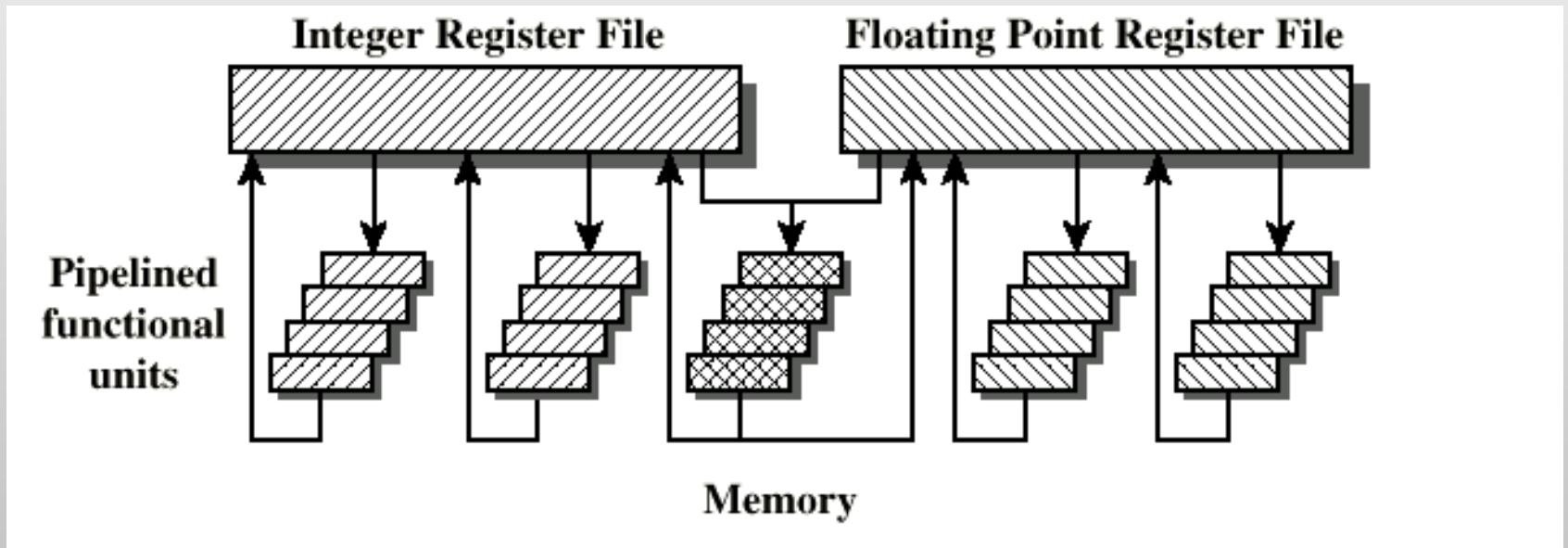


Mecanismo

- Executar instruções independentemente em diferentes *pipelines*.
- Permite busca e execução paralela
- Passo seguinte:
 - executar instruções em ordem diferente daquela que aparecem no programa original.

0 0 0

Organização Geral de Superescalar



0 0 0

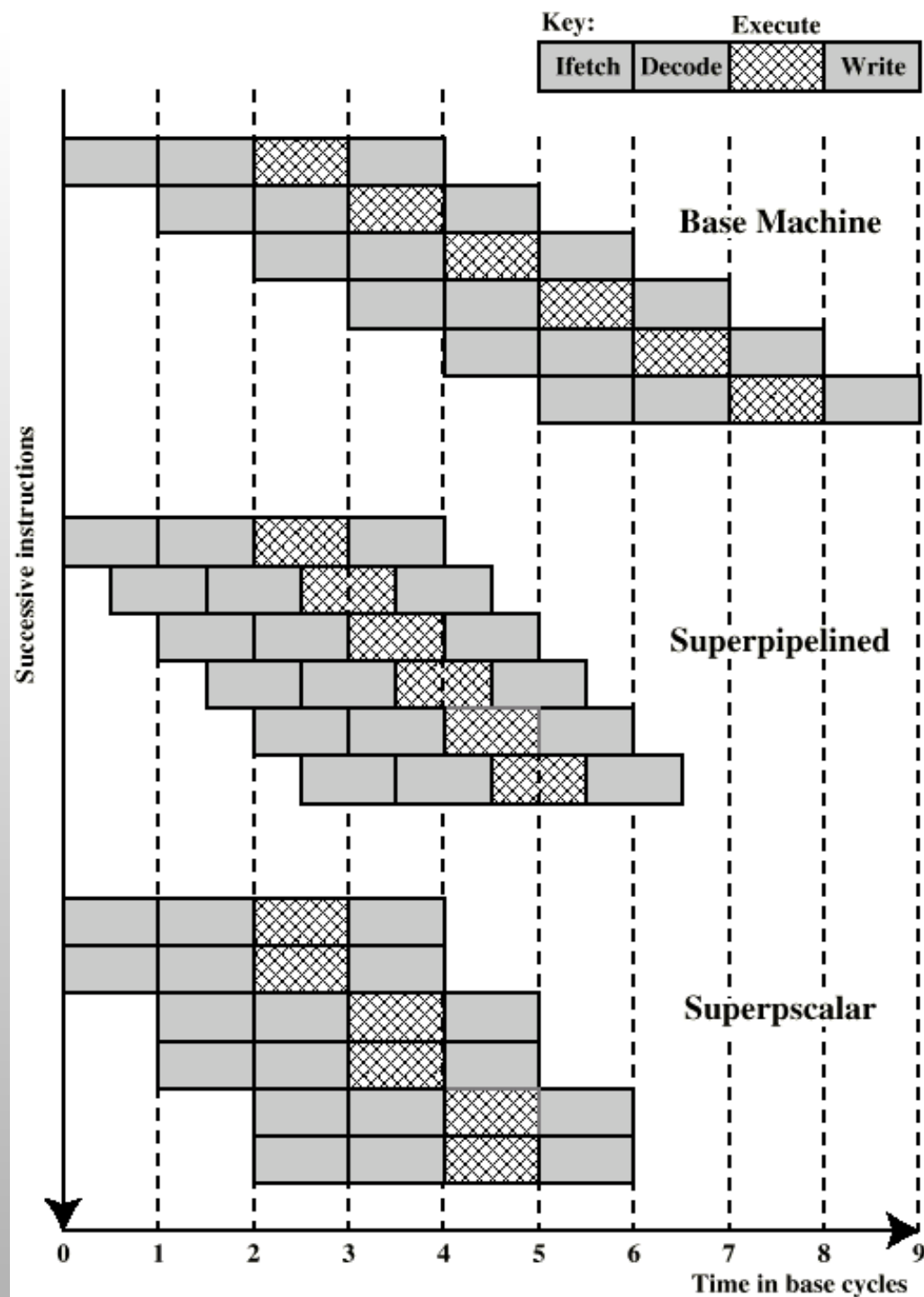
Superpipelined

- Muitos estágios do pipeline necessitam de menos da metade de um ciclo de relógio
- Dobrar a velocidade do relógio interno leva a duas tarefas por ciclo de relógio externo
- Superescalar propicia paralelismo da busca e execução

0 0 0

Superscalar X Superpipeline

- Pipeline com 4 estágios:
 - 1) busca de instrução;
 - 2) decodificação;
 - 3) execução;
 - 4) escrita de resultado.
- Superescalar:
 - 2 instâncias em paralelo
- Superpipeline: overhead maior no início e em caso de desvios.



0 0 0

Paralelismo no nível de instrução

- o Otimização baseada no compilador
- o Técnicas de Hardware
- o Limitado por
 - o Dependência de dados verdadeira
 - o Dependência de desvio
 - o Conflito de recurso
 - o Dependência de saída
 - o Antidependência

0 0 0

Dependência de dados verdadeira

- `ADD r1, r2 (r1 := r1+r2;)`
- `MOVE r3,r1 (r3 := r1;)`
- Pode buscar e decodificar a segunda instrução em paralelo com a primeira
- Pode NÃO executar a segunda instrução até que a primeira tenha terminado



Dependência de Desvio

- Não é possível executar as instruções depois de um desvio em paralelo com as instruções antes do desvio
- Além disso, se o tamanho da instrução não é fixo, as instruções devem ser decodificadas para saber quantas buscas são necessárias
- Isto impede buscas simultâneas

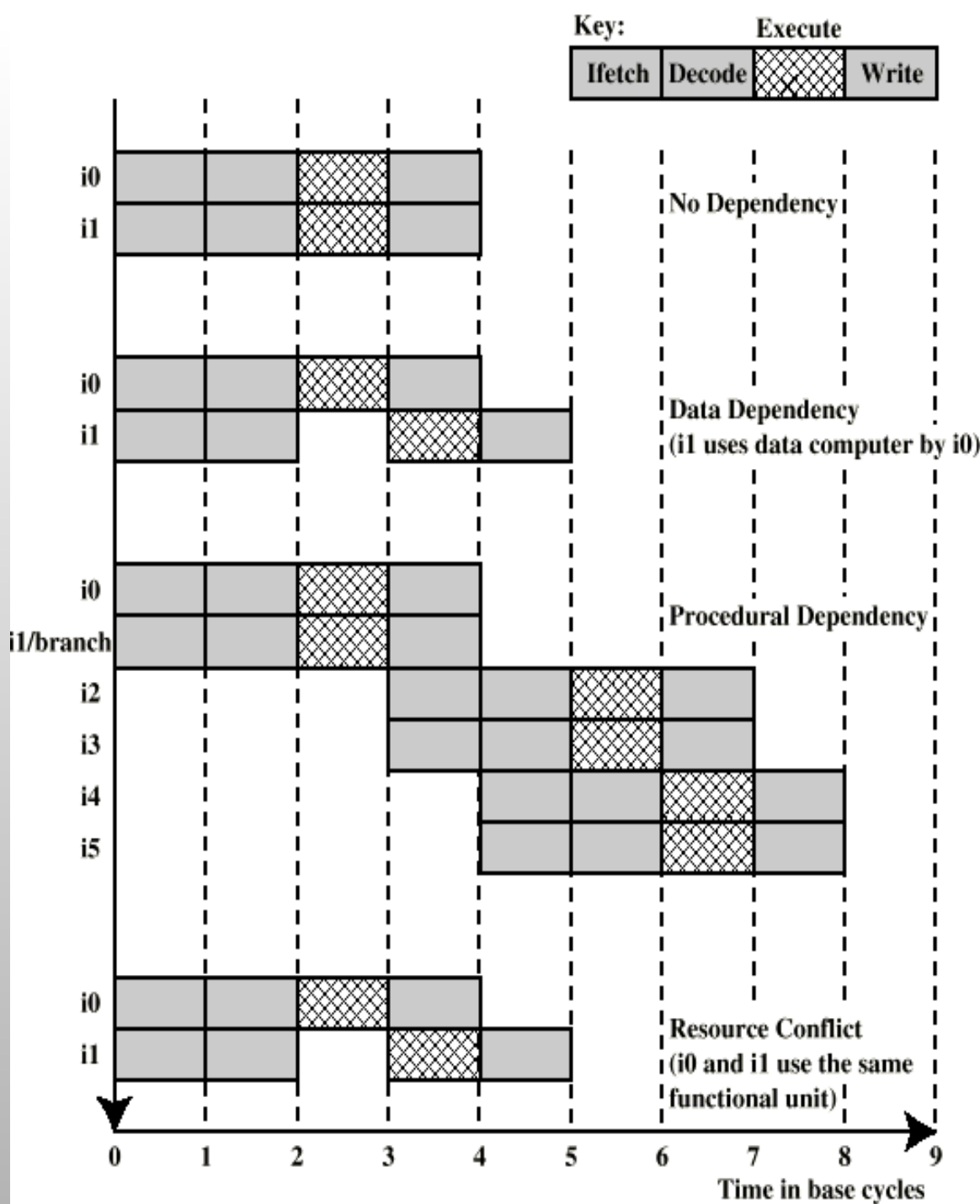
0 0 0

Conflito de Recurso

- o Duas ou mais instruções necessitando do mesmo recurso ao mesmo tempo
 - o ex. duas instruções aritméticas
- o Duplicação de recursos
 - o ex. ter duas unidades aritméticas

0 0 0

Dependências





Questões de Projeto

- Paralelismo no nível de instrução
 - Instruções em sequência são independentes
 - Execução pode ser sobreposta
 - Determinado pela dependência de dados e de desvio
- Paralelismo de Máquina
 - Habilidade de aproveitar o paralelismo no nível de instrução
 - Determinado pelo número de pipelines paralelos



Política de Iniciação de Instruções

- Processador:
 - precisa identificar o paralelismo no nível de instruções;
 - coordenar a busca, decodificação e execução de instruções em paralelo
- Iniciação de Instruções: processo de iniciar a execução de instruções nas unidades funcionais.
- Política: protocolo usado para iniciar a execução de instruções.

0 0 0

Tipos de Ordenação de Instruções

- Ordem na qual as instruções são buscadas
- Ordem na qual as instruções são executadas
- Ordem na qual as instruções atualizam o conteúdo de registradores e memória

0 0 0

Política de Iniciação de Instruções

- Iniciação em ordem com terminação em ordem.
- Iniciação em ordem com terminação fora de ordem.
- Iniciação fora de ordem com terminação fora de ordem.

o o o

Iniciação em Ordem com Terminação em Ordem

- o Inicia as instruções na ordem que elas ocorrem
- o Não muito eficiente
- o Pode buscar >1 instrução
- o Instruções devem esperar (*stall*) se for necessário

0 0 0

Iniciação em Ordem Terminação em Ordem (Diagrama)

Decode		Execute			Write		Cycle
I1	I2						1
I3	I4	I1	I2				2
I3	I4	I1					3
	I4			I3	I1	I2	4
I5	I6			I4			5
	I6		I5		I3	I4	6
			I6				7
					I5	I6	8

- I1 requer 2 ciclos para execução
- I3 e I4 - conflito de recursos
- I5 depende de dados de I4
- I5 e I6 - conflito de recursos

o o o

Iniciação em Ordem com Terminação Fora de Ordem

- o Melhora desempenho em processadores com execução de instruções que requerem vários ciclos.
- o Iniciação de instruções interrompida em caso de conflitos (dados, recursos ou desvio).
- o Dependência de Saída
- o Complexidade de interrupções

o o o

Iniciação em Ordem com Terminação Fora de Ordem

- o Dependência de Saída
 - o $R3 := R3 + R5; (I1)$
 - o $R4 := R3 + 1; (I2)$
 - o $R3 := R5 + 1; (I3)$
 - o I2 depende do resultado de I1 - dependência de dados
 - o Se I3 termina antes de I1, o resultado de I1 estará errado - dependência de saída (escrita-escrita)

0 0 0

Iniciação em Ordem Terminação Fora de Ordem (Diagrama)

Decode		Execute			Write		Cycle
11	12						1
13	14	11	12				2
	14	11		13	12		3
15	16			14	11	13	4
	16		15		14		5
			16		15		6
					16		7

o o o

Iniciação Fora de Ordem com Terminação Fora de Ordem

- o Desacopla pipeline de decodificação do pipeline de execução
- o Pode continuamente buscar e decodificar até que o pipeline esteja cheio
- o Quando uma unidade funcional se torna disponível pode executar uma instrução
- o Já que as instruções foram decodificadas, o processador pode examinar instruções à frente (*look ahead*)

0 0 0

Iniciação Fora de Ordem Terminação Fora de Ordem (Diagrama)

Decode		Window	Execute			Write		Cycle
11	12							1
13	14	11,12	11	12				2
15	16	13,14	11		13	12		3
		14,15,16		16	14	11	13	4
		15		15		14	16	5
						15		6

0 0 0

Antidependência

- Com a iniciação fora de ordem, outro tipo de dependência pode ocorrer:
- Dependência leitura-escrita
 - $R3 := R3 + R5;$ (I1)
 - $R4 := R3 + 1;$ (I2)
 - $R3 := R5 + 1;$ (I3)
 - $R7 := R3 + R4;$ (I4)
 - I3 não pode terminar antes que I2 inicie já que I2 necessita de um valor em R3 e I3 modifica R3



Renomeação de Registradores

- Dependência de saída e antidependência são conflitos de armazenamento
- O conteúdo dos registradores pode não refletir a correta ordenação do programa
- Pode levar à espera no pipeline (*pipeline stall*)
- Registradores alocados dinamicamente
 - registradores não são nomeados especificamente

0 0 0

Exemplo

$R3b := R3a + R5a$ (I1)

$R4b := R3b + 1$ (I2)

$R3c := R5a + 1$ (I3)

$R7b := R3c + R4b$ (I4)

- o Sem "letra": registrador lógico.
- o Com "letra": registrador em hardware alocado.
- o Observe R3a R3b R3c



Paralelismo de Máquina

- Técnicas de hardware:
 - duplicação de recursos;
 - iniciação fora de ordem;
 - renomeação de registradores.
- De acordo com estudo:
 - não vale a pena duplicar recursos sem renomeação de registradores;
 - Tamanho da janela de instruções: > 8 .

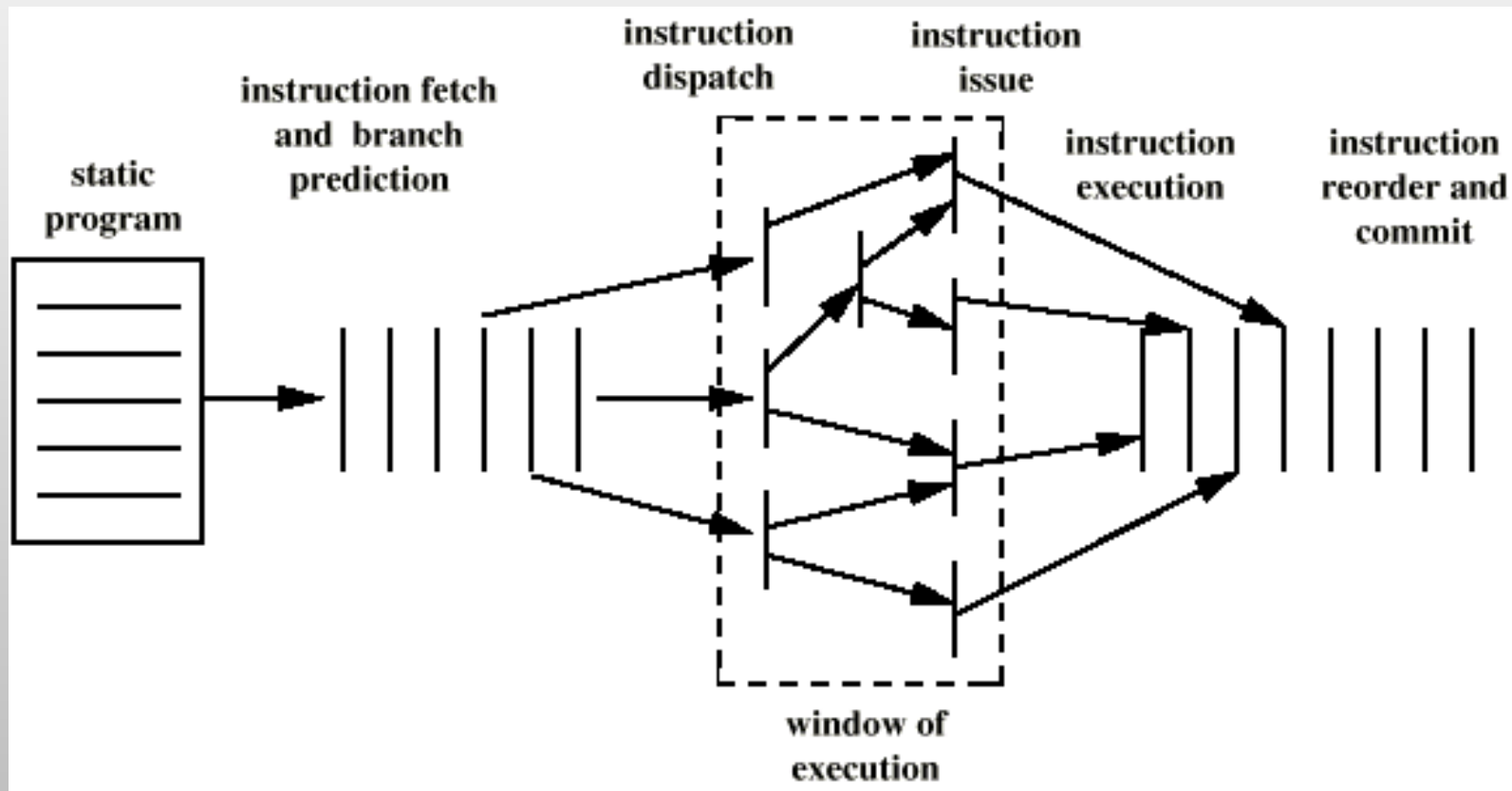


Predição de Desvio

- 80486 busca tanto a instrução sequencial seguinte como a instrução alvo.
 - atraso de 2 ciclos se o desvio é tomado.
- Atraso de desvio (RISC): não é eficiente em superescalares.
- Superescalares: técnicas estáticas ou dinâmicas de previsão de desvios.

0 0 0

Execução Superescalar





Implementação Superescalar

- Busca de múltiplas instruções simultaneamente.
- Lógica para determinar dependências de dados verdadeiras e mecanismos para transferir esses valores para onde necessários.
- Mecanismos para iniciar múltiplas instruções em paralelo.
- Recursos para execução de múltiplas instruções paralelas.
- Mecanismos para confirmar resultados do processamento na ordem correta.