



Register Renaming

Fernanda Alcântara Andaló

RA 041462

fernanda.andalo@ic.unicamp.br

- 1** Introdução
 - 1.1** O que é Register Renaming?
 - 1.2** O Problema
- 2** Considerações iniciais
- 3** Esquemas de Register Renaming
 - 3.1** Banco de registradores físicos
 - 3.2** Reorder buffer – ROB
- 4** Principais implementações
 - 4.1** Algoritmo de Tomasulo
- 5** Histórico
- 6** Conclusão

- ◇ O que é Register Renaming?
 - Explorar paralelismo
 - Aumento do número de instruções que um processador pode executar em paralelo

- ◇ O problema:
 - Número limitado de registradores disponíveis no conjunto de instruções
 - Isso causa problemas de dependências falsas

```
(1) r1 := r2 / r3
(2) r2 := r1 + r3
(3) r1 := r5 + r8
(4) r4 := r1 - r7
```

- Dependências de fluxo
 - Dependências de saída
 - Anti-dependências
- } Dependências falsas

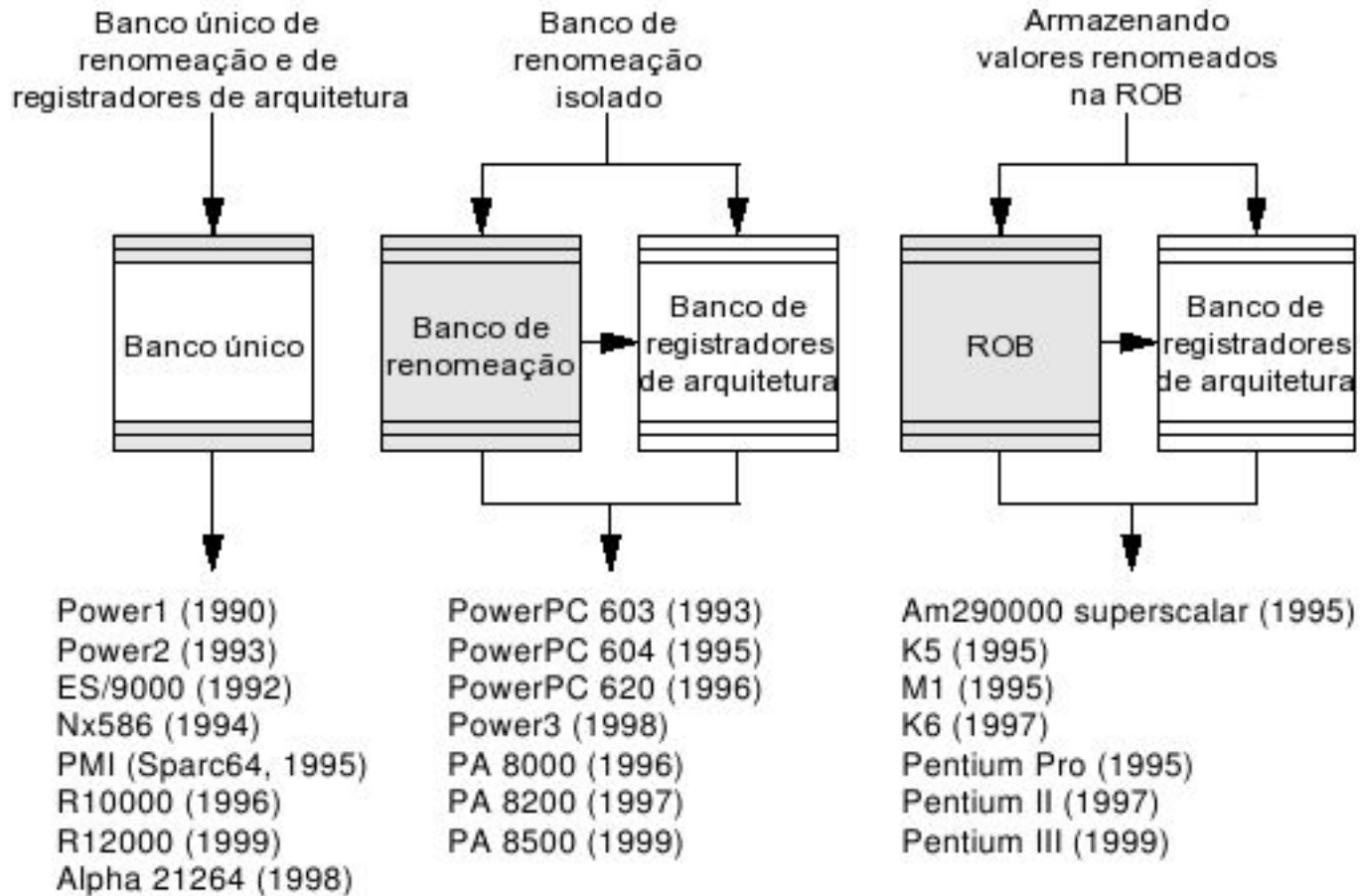
- ◇ Register Renaming realizado pelo compilador
- ◇ Aumentar número de registradores não é solução
 - Não é simples evitar duplicação de código
 - Número grande de registradores requer muitos bits para especificação
 - Muitos conjuntos de instruções já especificaram um número pequeno de registradores – Incompatibilidade!

Considerações Iniciais

- ◇ Registradores físicos vs. Registradores de arquitetura
- ◇ Renomeação parcial vs. Renomeação total
- ◇ Conflito de dados
 - Leitura após escrita (RAW)
 - Escrita após escrita (WAW)
 - Escrita após leitura (WAR)

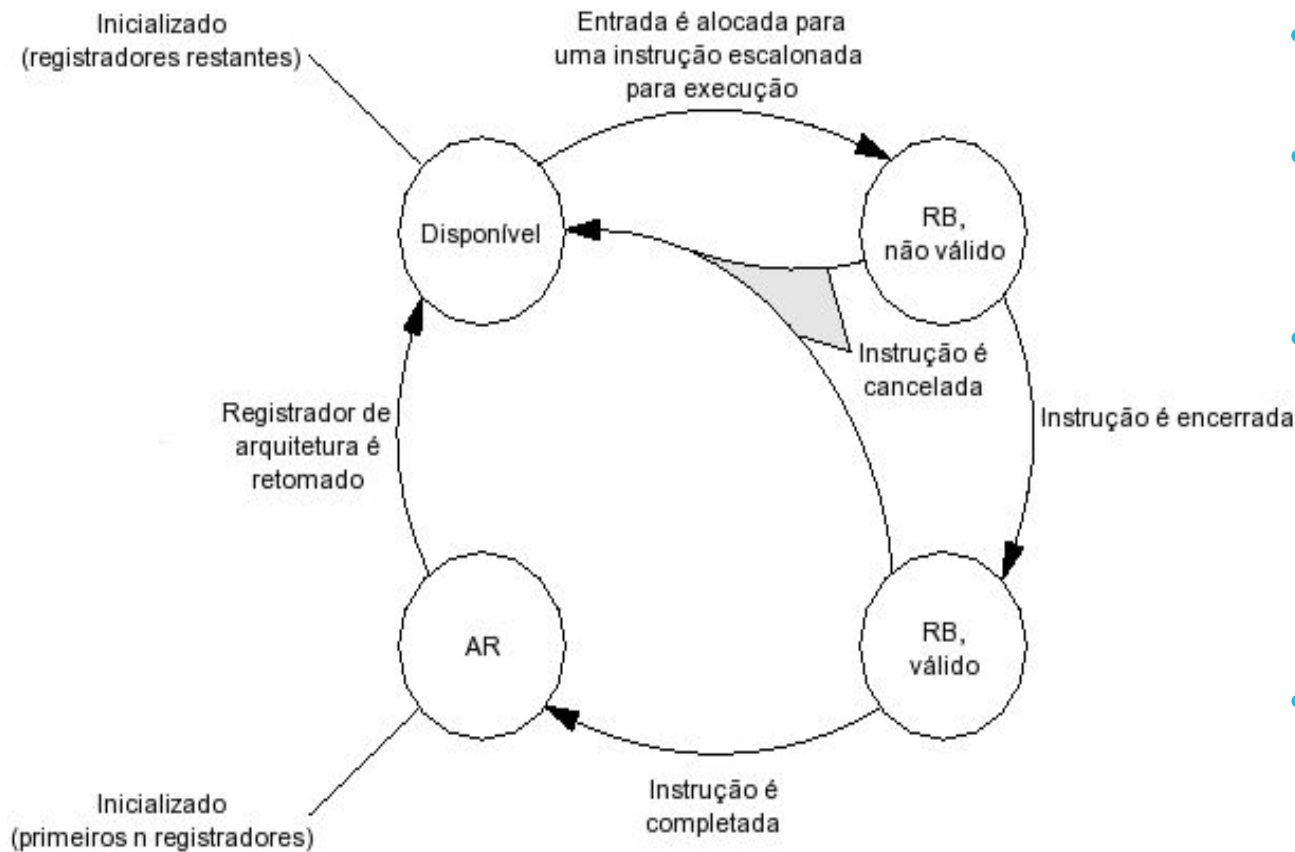
Esquemas de Register Renaming

◆ Buffer de Renomeação



Esquemas de Register Renaming

◇ Banco de registradores físicos

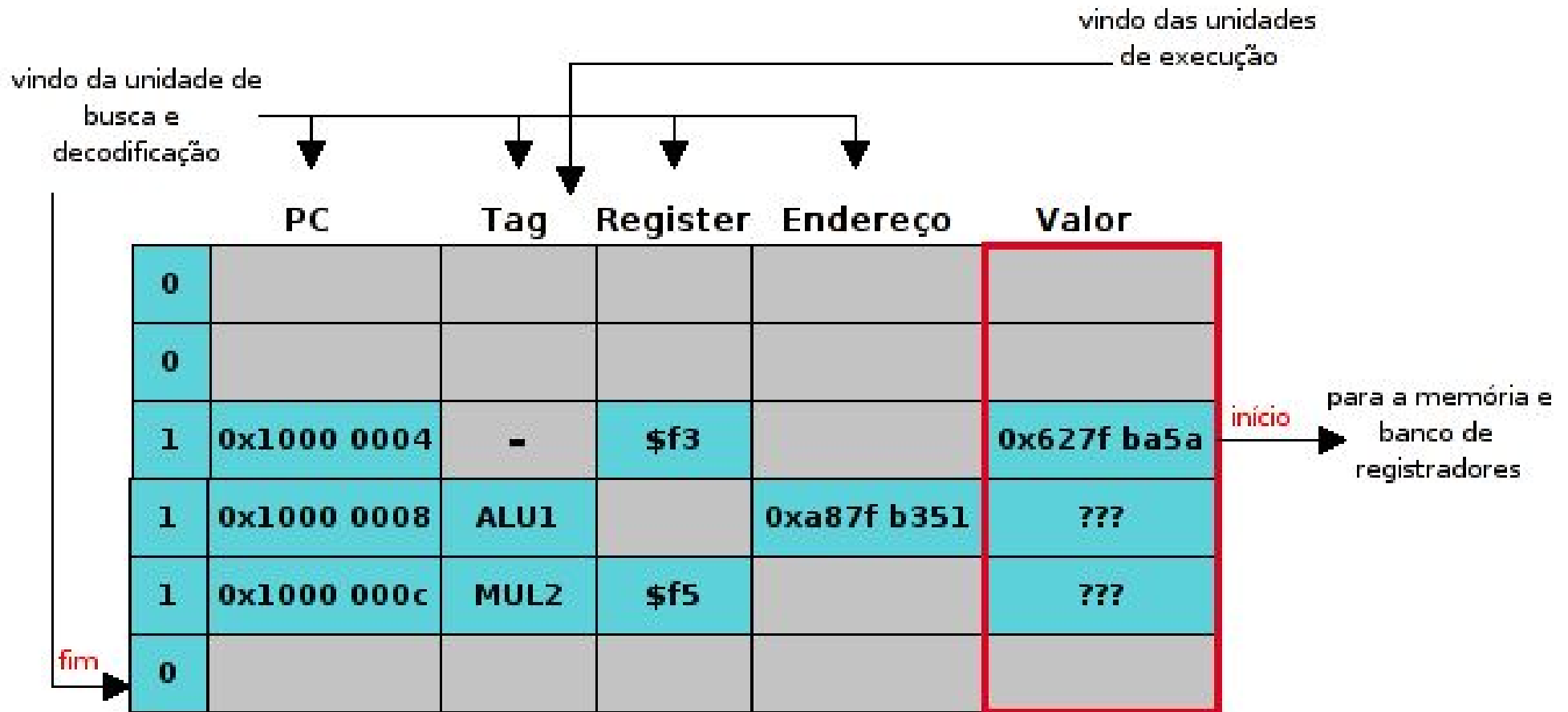


Estados possíveis:

- Disponível
- AR – Utilizado como registrador de arquitetura
- RB, não válido – Utilizado como buffer de renomeação, mas registrador ainda não contém o resultado da instrução associada
- RB, válido – Utilizado como buffer de renomeação e o registrador já contém o resultado da instrução associada

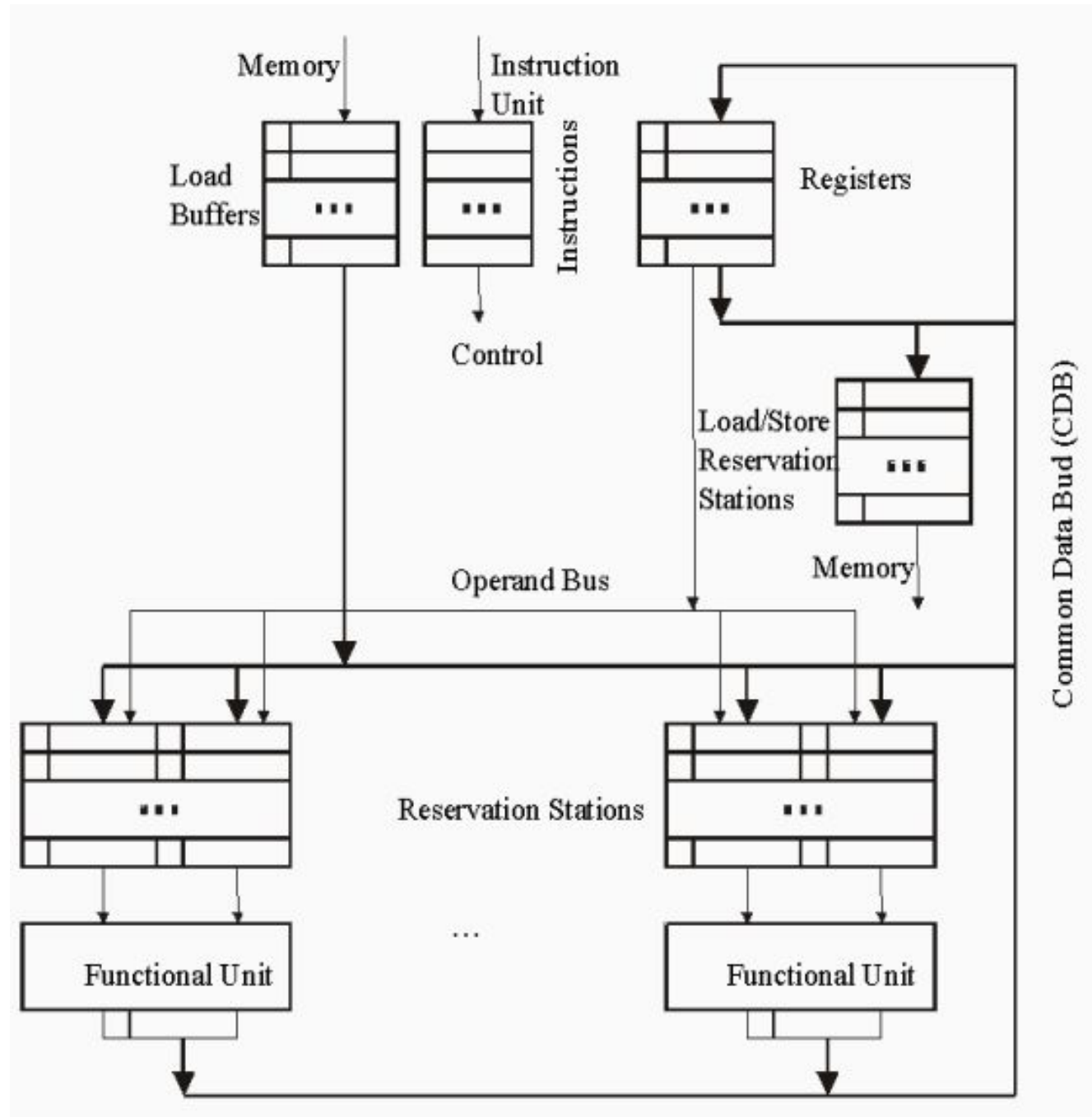
Esquemas de Register Renaming

◇ Reorder buffer (ROB)



Principais implementações

◆ Algoritmo de Tomasulo



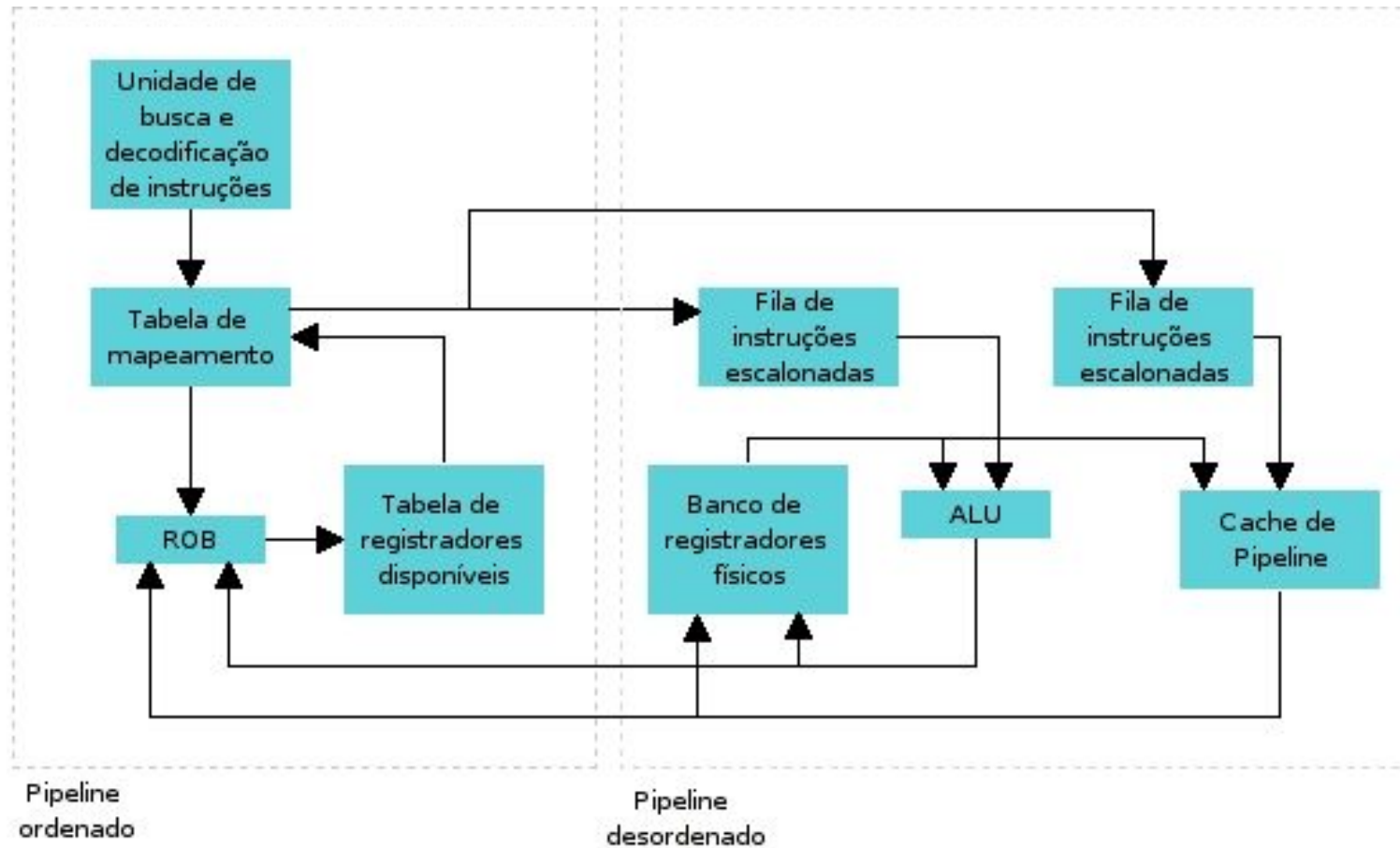
◆ Algoritmo de Tomasulo

O algoritmo de Tomasulo possui três estágios:

1. **Emissão** - retira instrução da Fila de Instruções
Se a *reservation station* está livre, o algoritmo de Tomasulo emite a instrução e procura os operandos dos registradores, se possível.
2. **Execução** - opera nos operandos
Quando ambos os operandos estão prontos, então envia a instrução para uma unidade funcional e a executa.
3. **Escrita do resultado** - finaliza a execução
Escreve no caminho de dados comum para todas as unidades que esperam;
Marca a *reservation station* como disponível.

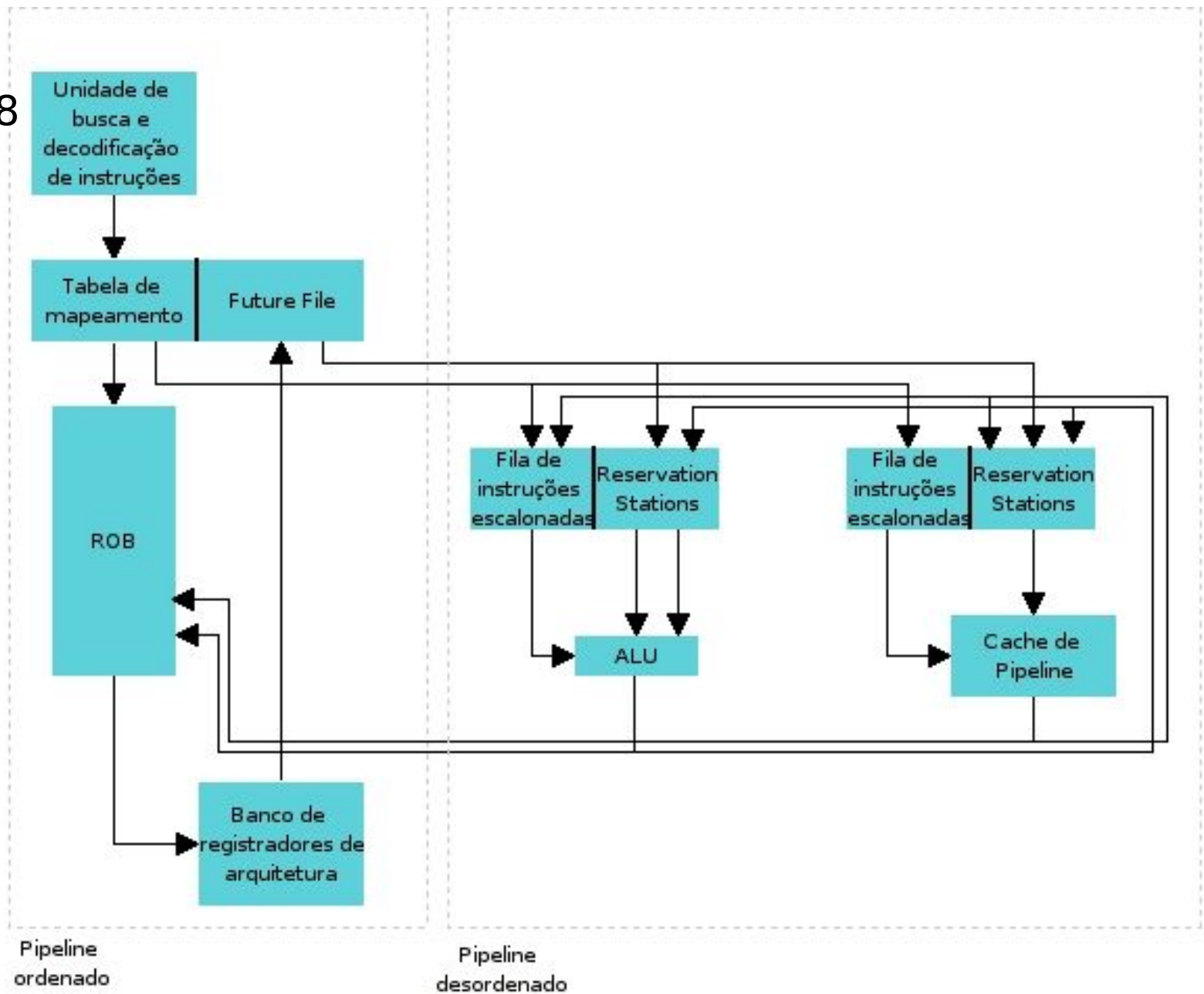
Principais implementações

- ◆ MIPS R10000 e Alpha21264



Principais implementações

◆ AMD K7 e K8



Histórico

Processadores RISC

Compaq	Alpha		Alpha 21064 (2)		Alpha 21064 (2)		Alpha 21064 (2)	
Motorola	MC88000		MC88110 (2)					
HP	PA		PA 7100 (2)		PA7200 (2)	PA8000 (4)	PA82000 (4)	PA8500 (4)
IBM	Power	Power1 (4) (RS/6000)	Power2 (6/4) ****			P25C (6/4) ****		
PowerPC Alliance	PowerPC		PPC 601 (3) *		PPC 604 (4) *	PPC 620 (4) *		Power3 (4)
			PPC 603 (3) *		PPC 602 (2) *			
MIPS	R			R8000 (4)	UltraSparc (4)	R10000 (4)		R12000 (4)
Sun/Hal	Sparc		SuperSparc (3)		PMI (4) (Sparc 64)		UltraSparc-2 (4)	UltraSparc-3 (4)

Processadores CISC

Intel	80x86		Pentium (2)		Pentium Pro (3)		Pentium MMX (2) Pentium II (3)		Pentium III (3)
IBM	ES		ES/9000 (2)						
TRON	Gmicro		Gmicro/500 (2)						
Cyrix	M					M1 (2)	MII (2)		
Motorola	MC68000		MC68060 (3)						
AMD	Nx/K			Nx586 (1/3) **	K5 (4)		K6 (3)		K7 (3)

	Renomeação Parcial
	Renomeação Total

1990 1991 1992 1993 1994 1995 1996 1997 1998 1999

Tipo de registrador (ano de maior produção)	Tipo de buffer de renomeação	Número de buffers de renomeação	
		FX	FP
RISC processors			
PowerPC 603 (1993)	Ren. reg. file	N/A	4
PowerPC 604 (1995)	Ren. reg. file	12	8
PowerPC 620 (1996)	Ren. reg. file	8	8
Power3 (1998)	Ren. reg. file	16	24
R10000 (1996)	Merged	32	32
R12000 (1998)	Merged	32	32
Alpha 21264 (1998)	Merged	48	41
PA 8000 (1986)	Ren. reg. file	56	56
PM1 (1996)	Merged	38	24
x86 (CISC) processors			
Pentium Pro (1995)	In the ROB	40	
Pentium II (1997)	In the ROB	40	
K5 (1995)	In the ROB	16	
K6 (1996)	In the ROB	24	
M3 (2000 expected)	Merged	32	N/A

- ◇ A maioria dos processadores atuais empregam alternativas básicas
 - Uso de banco único (R10000, R12000, M3)
 - Uso de banco isolado e mapeamento de registradores (linha PA8x00, Power3)
 - Renomeação no ROB e uso de tabelas de mapeamento (Pentium Pro, Pentium II, Pentium III)