

Arquitetura de Computadores

Paralelismo em Nível de Instrução
Despacho Múltiplo e
Especulação Baseada em Hardware

Despacho Múltiplo

- O Algoritmo de Tomasulo permite a execução concomitante de instruções ao despachar instruções, no mesmo ciclo, para unidades funcionais diferentes.
- Isso porém torna a dependência de controle um problema pois a previsão de desvios com exatidão pode não ser suficiente ao explorar mais o paralelismo em nível de instrução. Pois pode haver situações em que o processador precise executar uma instrução de desvio por ciclo.

Especulação baseada em Hardware

- Para superar a Dependência de Controle
 - o resultado dos desvios são especulados; e
 - as instruções são executadas supondo serem as corretas,
 - porém para que seus resultados sejam efetivados é preciso esperar a confirmação das instruções após a definição do desvio.

Especulação de hardware

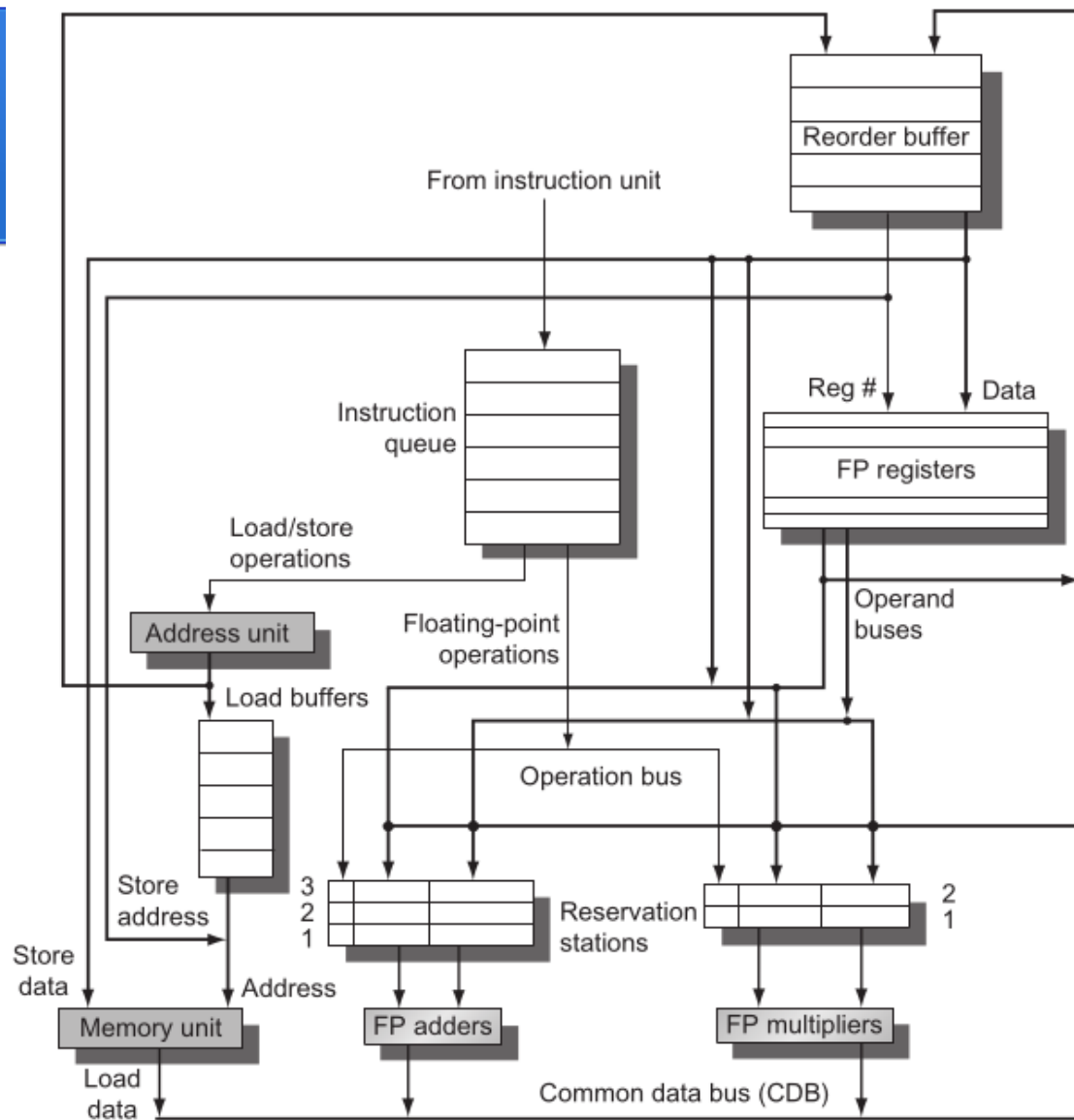
- A Especulação de hardware combina três ideias principais:
 - Previsão Dinâmica de Desvios para escolher as instruções a serem executadas;
 - Especulação para permitir a execução de instrução antes que as dependências de controle sejam resolvidas; e
 - Escalonamento Dinâmico para tratar com o escalonamento de diferentes combinações de blocos básicos.

Especulação de Hardware

- A ideia por trás da especulação de hardware é permitir a execução das instruções fora da ordem mas forçá-las a serem confirmadas em ordem.
- Para isto é necessário separar o processo de execução do processo de confirmação; ou seja, o hardware deve permitir a execução de uma instrução especulada porém seu resultado só poderá ser compartilhado após a confirmação de que a instrução realmente deveria ser executada.

Especulação de Hardware

- Logo é necessário incrementar um local que armazene os resultados das instruções especuladas até serem confirmadas, para isto é adicionado ao hardware de Tomasulo o Buffer de Reordenação (ROB).



Buffer de Reordenação

- O Buffer de Reordenação (ROB):
 - Armazena o resultado de instrução (conclusão – confirmação)
 - Fornece operandos; assim como o arquivo de registradores;
 - Cada entrada tem 4 campos:
 - Tipo da instrução (desvio, store ou ALU-load)
 - Destino (registrador ou endereço de memória de destino)
 - Valor (resultado da instrução)
 - Pronto (indica se a instrução foi concluída ou ainda não)

Especulação de Hardware

- Além de estender o hardware é necessário incluir uma fase de confirmação (commit) da instrução ao pipeline de execução das instruções.
- Assim o pipeline passa a ter 4 fases:
 - Despacho (Issue);
 - Execução (Execute);
 - Escrita de Resultados (Write Result); e
 - Confirmação (Commit);

Especulação de Hardware

- Despacho (Issue):
 - Lê instrução da memória de instruções;
 - Se houver
 - Uma estação de reserva vazia; e
 - Um slot vazio no ROB
 - Então
 - Despacha a instrução;
 - Coloca os operandos disponíveis na estação de reserva;
 - Atualiza as entradas para indicar os buffers em uso (inclusive do ROB)
 - Caso contrário o despacho é interrompido até liberar

Especulação de Hardware

- Execução (Execute):
 - Se
 - um ou mais operando não estão disponíveis
 - Então
 - Monitora o CDB até os valores serem liberados
 - Quando os operandos estiverem disponíveis executa a instrução.

Especulação de Hardware

- Escrita de Resultados (Write Result):
 - Com o resultado disponível
 - Escreve no CDB;
 - E do CDB no ROB e nas estações de reservas.
 - No caso de instruções store se o valor estiver disponível escreve no campo Valor do ROB, caso contrário monitora o CDB.

Especulação de Hardware

- Confirmação (Commit):
 - Se a instrução é load ou operação na ALU correta
 - Quando a confirmação chega no ROB e o resultado está lá,
 - Então o processador grava o valor no registrador de destino e remove a entrada do ROB .
 - Se a instrução é um store (armazenamento)
 - Quando a confirmação chega no ROB e o resultado está lá,
 - Então o processador grava o valor no endereço de memória calculado e remove a entrada do ROB .
 - Se a instrução é um desvio incorreto
 - Quando um desvio incorreto chega ao ROB então a entrada é liberada, nenhum resultado é escrito, e a execução é iniciada na instrução sucessora correta.
 - Se o desvio foi previsto corretamente então ele é apenas encerrado.

Especulação de Hardware

- Exemplo: Assuma as seguintes latências para as unidades funcionais de ponto flutuante:
 - o load corresponde a 1 ciclo de clock,
 - a adição/subtração corresponde a 2 ciclos de clock,
 - a multiplicação a 6 ciclos de clock; e
 - a divisão a 12 ciclos de clock.
- Usando o seguinte segmento de código a seguir, mostre como as tabelas de status ficam quando o `fmul.d` está pronto para ser enviado para confirmação.

Especulação de Hardware

fld f6, 32(x2)

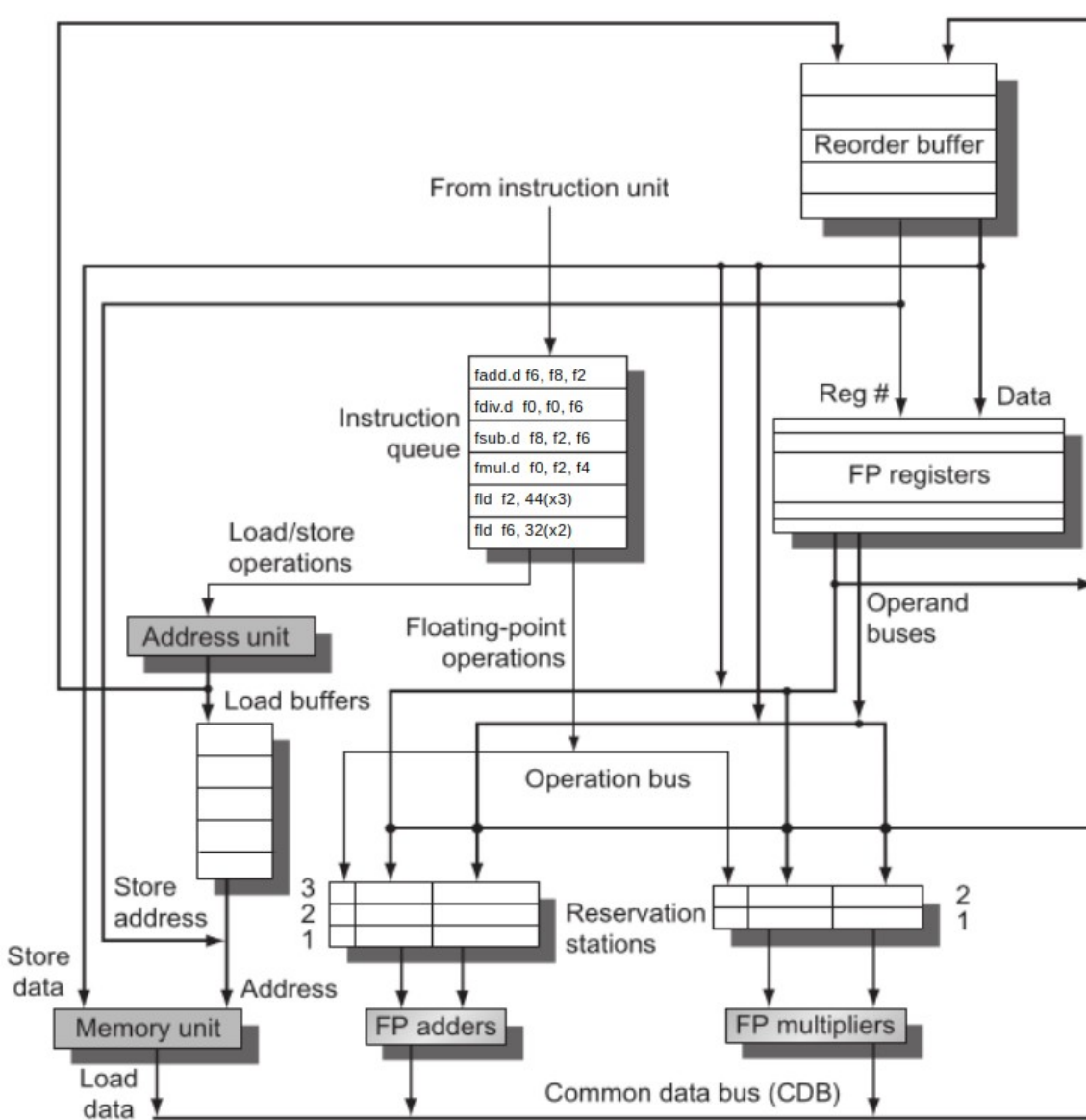
fld f2, 44(x3)

fmul.d f0, f2, f4

fsub.d f8, f2, f6

fdiv.d f0, f0, f6

fadd.d f6, f8, f2



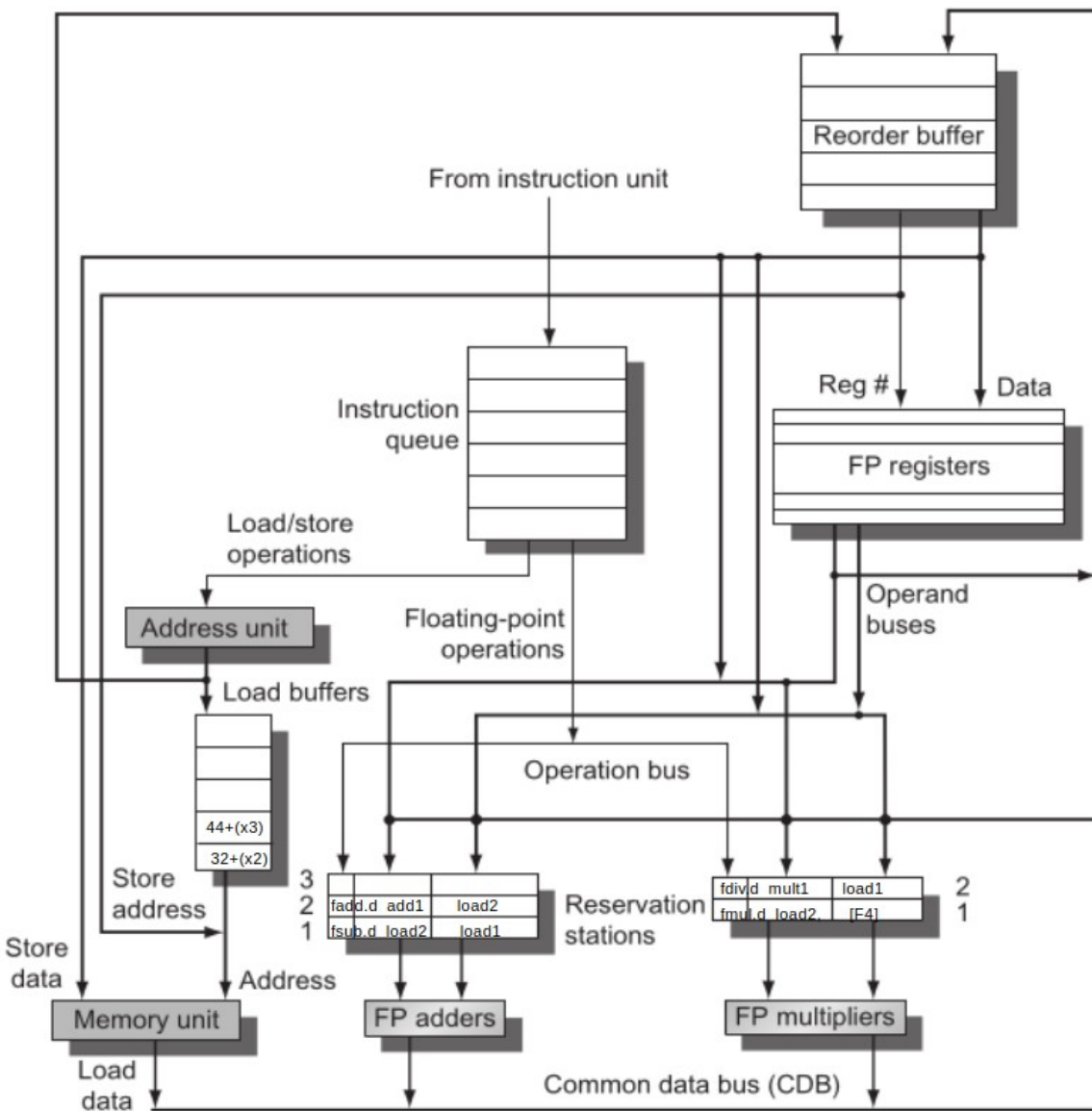
Tabelas do Reorder Buffer e das Estações de Reservas vazias, sem instruções no pipeline.

Tabela do Reorder Buffer

Entrada	Instrução	Destino	Valor	Pronto	Busy
1					N
2					N
3					N
4					N
5					N
6					N

Tabela das estações de Reservas

Nome	OP	Qj	Qk	Vj	Vk	A	Busy
load1							N
load2							N
add1							N
add2							N
mult1							N
mult2							N



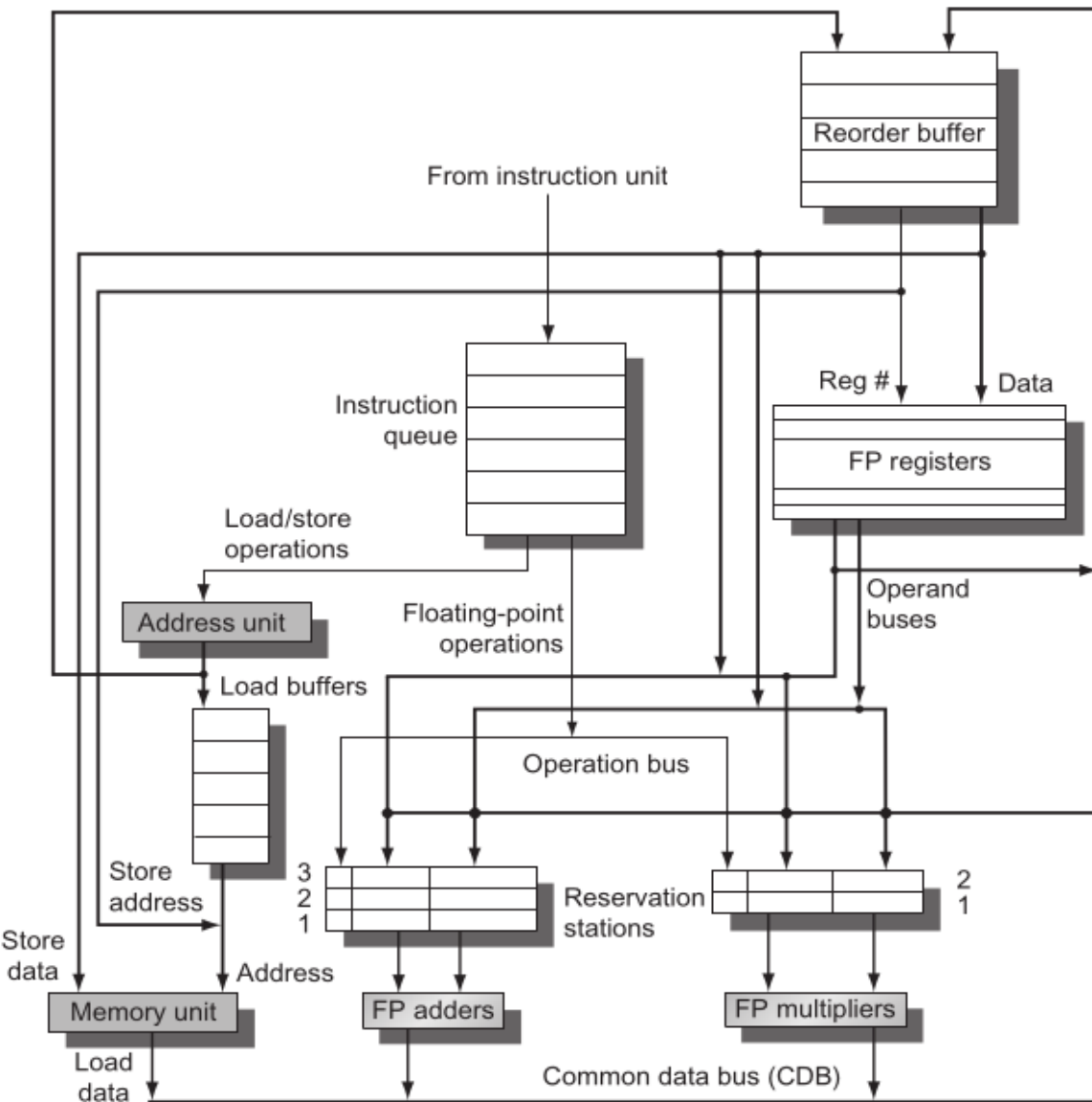
Tabelas no início, assim que as instruções foram despachadas

Tabela do Reorder Buffer

Entrada	Instrução	Destino	Valor	Pronto	Busy
1	fld.d	f6		N	S
2	fld.d	f2		N	S
3	fmul.d	f0		N	S
4	fsub.d	f8		N	S
5	fdiv.d	f0		N	S
6	fadd.d	f6		N	S

Tabela das estações de Reservas

Nome	OP	Qj	Qk	Vj	Vk	A	Busy
load1	fld.d					32+x2	S
load2	fld.d					44+x3	S
add1	fsub.d	load2	load1				S
add2	fadd.d	add1	load2				S
mult1	fmul.d	load2				[F4]	S
mult2	fdiv.d	mult1	load1				S



Tabelas no momento em que a instrução fmul.d está pronta para ser confirmada. As instruções fld.d já foram confirmadas. Apenas fdiv.d ainda não está pronta, todas as outras já foram calculadas, têm seus valores definidos mas não foram confirmadas.

Tabela do Reorder Buffer

Entrada	Instrução	Destino	Valor	Pronto	Busy
1					N
2					N
3	fmul.d	f0	[F0]	S	S
4	fsub.d	f8	[F8]	S	S
5	fdiv.d	f0		N	S
6	fadd.d	f6	[F6]	S	S

Tabela das estações de Reservas

Nome	OP	Qj	Qk	Vj	Vk	A	Busy
load1							N
load2							N
add1							N
add2							N
mult1							N
mult2							N