Simulador Educacional Visual para Arquiteturas Superescalares

Algoritmo de Tomasulo e Reorder Buffer (ROB)

Felipe Vilhena Dias, Gabriel Cunha Schlegel,
Iago Fereguetti, Lucas Henrique Rocha Hauck

PUC Minas - Arquitetura de Computadores III

Contexto e Problema

O Abismo Teórico-Prático

O ensino de ILP (Instruction Level Parallelism) esbarra na dificuldade de visualizar processos dinâmicos.

Diagramas estáticos em livros não mostram:

- O preenchimento ciclo a ciclo das Estações de Reserva.
- A disputa pelo CDB (Common Data Bus).
- O efeito cascata de uma predição errada (Flush).

Nosso Objetivo

Criar uma ferramenta que "abra a caixa preta" do processador, permitindo que alunos injetem código MIPS e vejam a mágica acontecer.

Conceitos Fundamentais

Arquiteturas Superescalares

Definem processadores capazes de despachar e executar **múltiplas instruções por ciclo de clock**. Para isso, exploram o Paralelismo a Nível de Instrução (ILP) para manter várias unidades funcionais ocupadas simultaneamente.

Algoritmo de Tomasulo

Técnica de agendamento dinâmico que permite a **Execução Fora de Ordem**. Seus principais mecanismos são:

- **Despacho Dinâmico:** Verifica dependências em tempo de execução.
- **Renomeação de Registradores:** Elimina conflitos de escrita (WAW) e leitura (WAR) usando buffers.

Trabalhos Relacionados

Gem5

Ferramenta padrão para pesquisa acadêmica e industrial. Oferece precisão de ciclo ("cycle-accurate"), mas possui alta complexidade de configuração e carece de visualização gráfica nativa.

Ripes / WebMIPS

Ferramentas visuais excelentes para o ensino de Assembly e pipelines estáticos. Contudo, abstraem a complexidade da execução especulativa e despacho dinâmico.

Nossa Abordagem

Foco específico na visualização das estruturas do algoritmo de Tomasulo (RS e ROB), preenchendo a lacuna entre simuladores de ISA simples e ferramentas de pesquisa complexas.

Componentes Chave do Simulador

RS (Reservation Stations)

Buffers na entrada das unidades funcionais. Elas guardam:

- **Vj, Vk:** Valores dos operandos (se já conhecidos).
- **Qj, Qk:** Tags de quem vai produzir o valor (se desconhecido).

RAT (Register Alias Table)

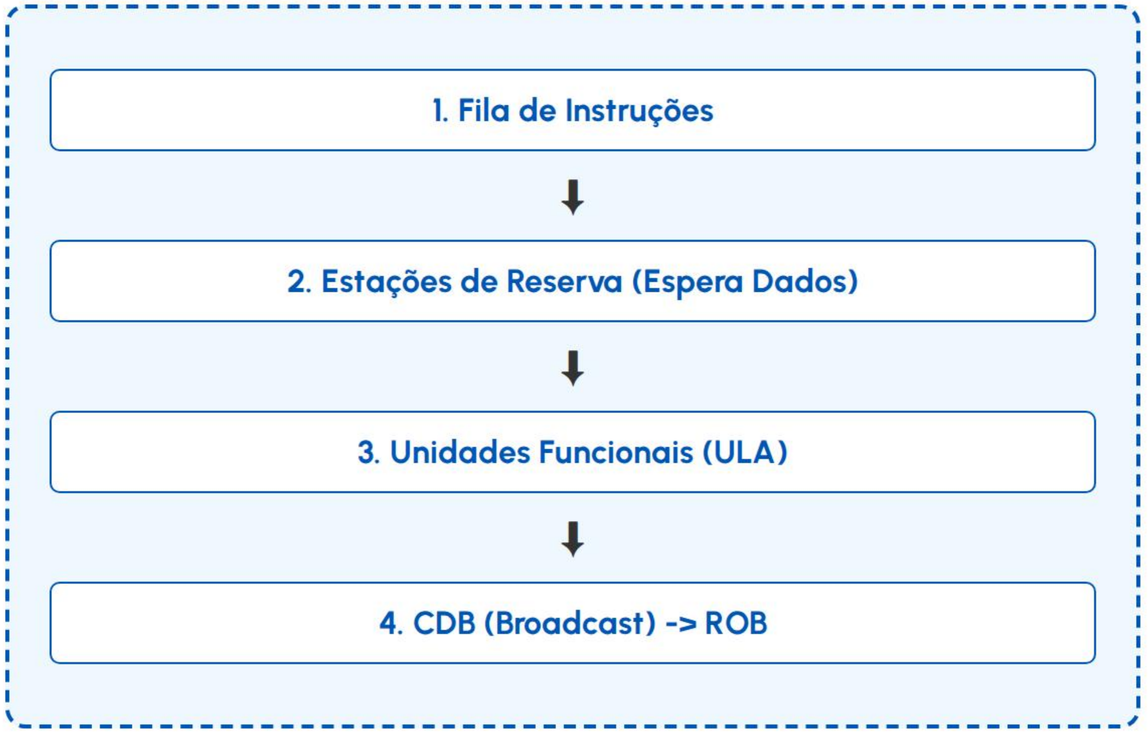
A "tabela de tradução". Ela diz se o valor de `R1` está no banco de registradores ou sendo produzido por uma instrução no ROB (ex: `ROB#3`).

CDB (Common Data Bus)

A via expressa. Quando uma instrução termina, ela coloca seu `ROB ID` e `Valor` aqui. Todas as RS "escutam" o barramento.

O Ciclo de Vida da Instrução

1. **ISSUE (Despacho):** Se há vaga na RS e no ROB, despacha. Renomeia registradores destino na RAT.
2. **EXECUTE:** Se os operandos (Vj, Vk) estão prontos, executa. Senão, monitora o CDB.
3. **WRITE RESULT:** Terminou? Coloca resultado no CDB e no ROB. Libera a RS.



Implementação do Simulador

Para simular hardware paralelo usando Python (sequencial), utilizamos a técnica de **Pipeline Reverso** dentro de cada ciclo de clock.

Isso garante que uma instrução não seja despachada e graduada "magicamente" no mesmo instante.

Tecnologias: Python 3 + Tkinter (GUI Nativa).

```
# Ordem de processamento no simulador: def step(self): self.clock += 1
# 1. Quem está no topo do ROB pode sair? self.commit_stage() # 2. Quem
terminou de calcular? Avise o CDB! self.write_result_stage() # 3. Quem
tem operandos? Comece a calcular! self.execute_stage() # 4. Há espaço?
Despache nova instrução! self.issue_stage()
```

Hora do Teste!

Vamos pausar a teoria e ver o simulador processando instruções MIPS, tratando hazards e fazendo especulação em **tempo real**.

Comparação com Outras Ferramentas

Por que nossa abordagem é ideal para o ensino de Arquitetura III?

Ferramenta	Foco Principal	Visualização Tomasulo	Edição Dinâmica
Gem5	Pesquisa/Precisão	Não (Texto/Stats)	Não (Script)
WebMIPS	Pipeline 5 Estágios	Não	Sim
Ripes	RISC-V Visual	Limitado	Sim
Nosso Simulador	**Ensino Tomasulo+ROB**	**Sim (RS/ROB/CDB)**	**Sim**

Conclusão

Resultados Obtidos

O simulador atingiu seu objetivo pedagógico ao:

- Tornar visível o invisível (buffers, tags, barramentos).
- Quantificar o ganho do ILP (IPC) e o custo dos hazards (Stalls).
- Desmistificar o mecanismo de recuperação de especulação (Flush).

Trabalhos Futuros

- Integração com Hierarquia de Memória (Caches L1/L2 visualizáveis).
- Implementação de preditor de desvios dinâmico (BPB).

Obrigado!

Perguntas?

github.com/felipevidias/tomasulo_sim