

Simulador Didático do Algoritmo de Tomasulo

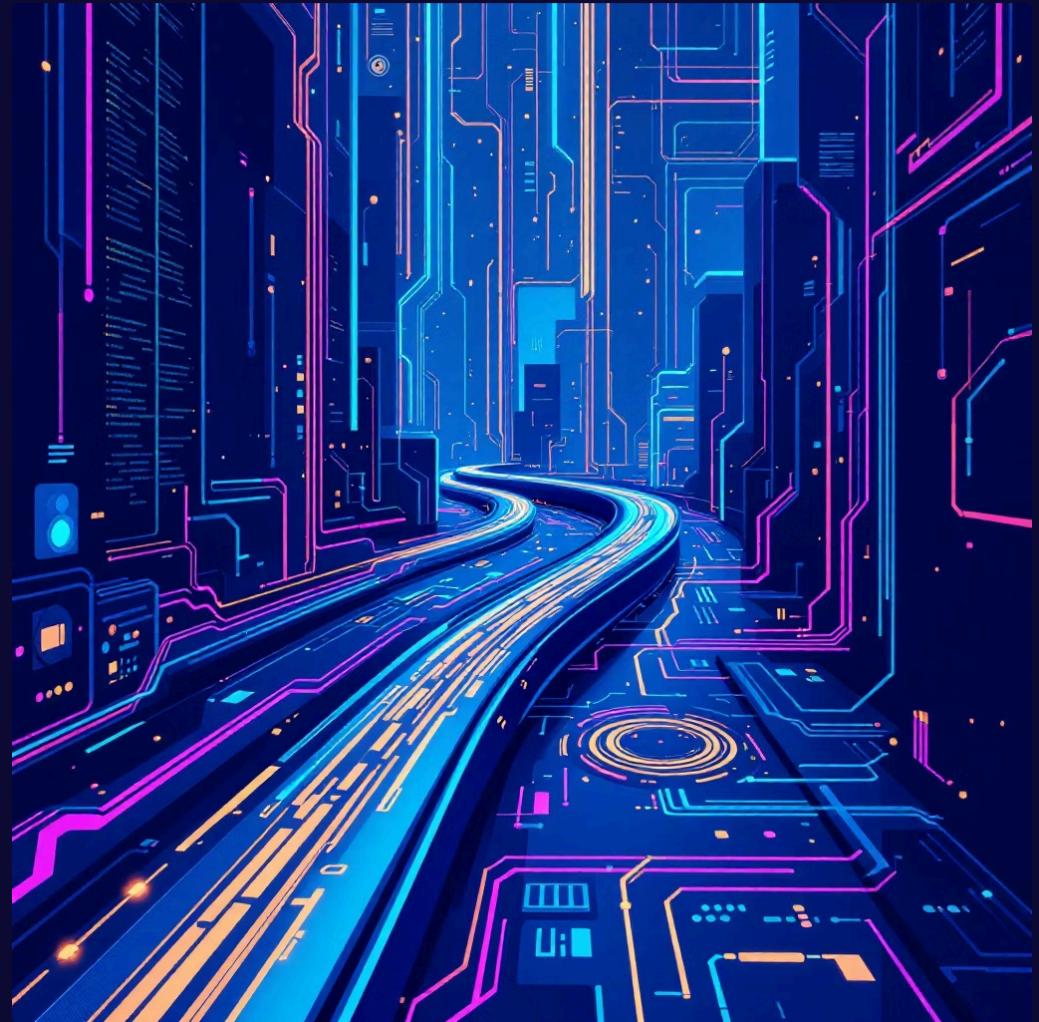
Implementação de um simulador com arquitetura superescalar, buffer de reordenação e especulação de desvio para instruções MIPS simplificadas

O Problema das Arquiteturas Simples

Limitações Tradicionais

Em arquiteturas simples, as instruções executam estritamente em ordem. Qualquer dependência de dados ou latência de memória trava o pipeline inteiro, desperdiçando ciclos preciosos de processamento.

O resultado? Subutilização dos recursos do processador e desempenho comprometido.



A Solução Elegante do Tomasulo



Estações de Reserva

Instruções aguardam em RS enquanto operandos ficam prontos, liberando o fluxo do pipeline



Renomeação de Registradores

RAT elimina dependências falsas (WAR/WAW), cada resultado novo vira entrada no ROB



Broadcast via CDB

Resultados são transmitidos para todas RS que esperavam aquele valor específico



Buffer de Reordenação

ROB garante commits em ordem, preservando semântica do programa

- ❑ **Princípio fundamental:** O Tomasulo permite execução fora de ordem, mas mantém a aparência de execução em ordem usando o ROB.

Arquitetura Superescalar de Largura 2

01

Issue Width = 2

Simulador tenta emitir até 2 instruções por ciclo para estações de reserva disponíveis

02

Commit Width = 2

Pode finalizar até 2 instruções por ciclo, maximizando throughput do pipeline

03

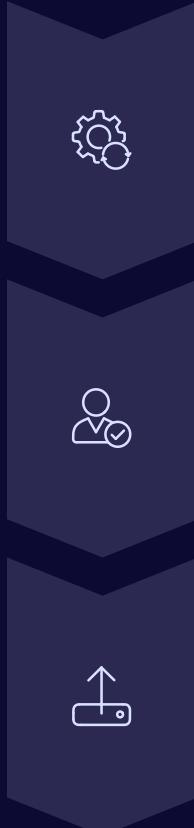
RS Especializadas

Tipos diferentes de estações: ADD/SUB/ADDI/BEQ/BNE, MUL/DIV, LD/ST para paralelismo real

Sem dependências e com recursos livres, múltiplas instruções executam simultaneamente, aumentando significativamente o desempenho.



Ciclo de Execução: Três Fases Essenciais



1. Execução e Broadcast

RS em execução avançam latência. Ao terminar: calcula resultado, marca ROB como pronto, faz broadcast para RS dependentes via CDB

2. Commit (ROB)

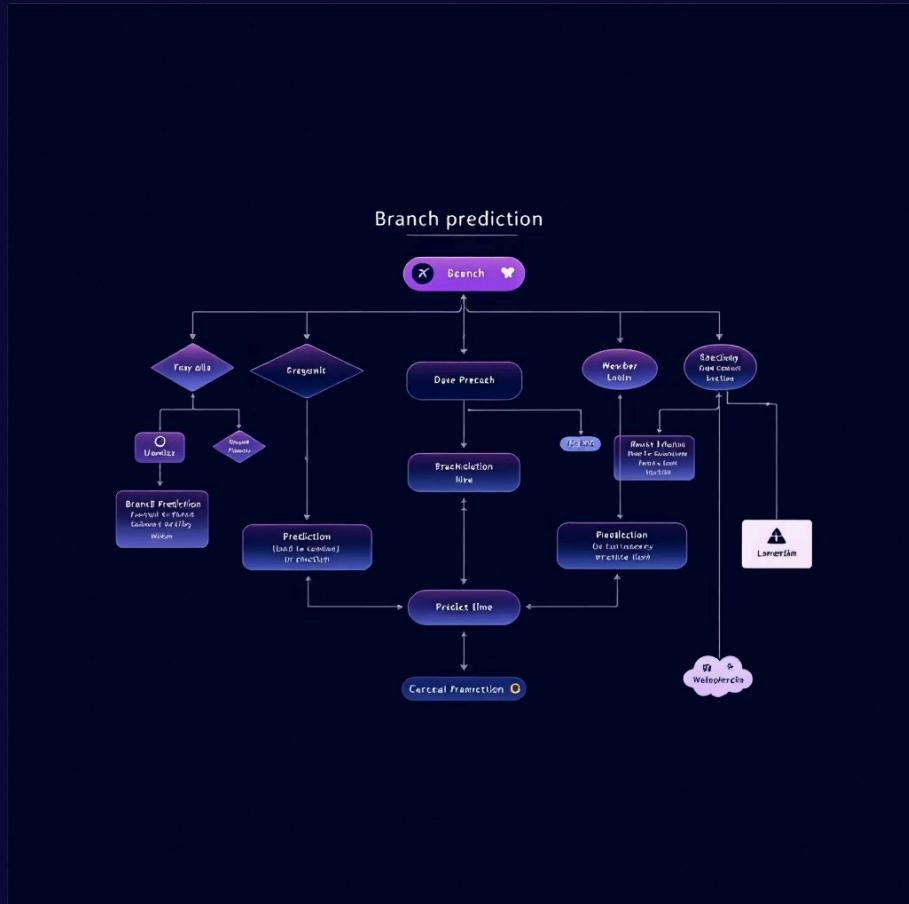
Analisa cabeça do ROB. Se pronto: escreve em registrador/memória, atualiza estatísticas, libera entrada. Até 2 commits por ciclo

3. Issue (Despacho)

Com espaço em ROB/RS: lê próxima instrução, aloca ROB, coloca em RS adequada, marca dependências via RAT usando tags Qj/Qk ou valores Vj/Vk

Ao final: atualiza contador de ciclos. Se não comitou mas há trabalho pendente, incrementa ciclos de bolha.

Especulação de Desvio: Previsão Inteligente



Estratégia de Previsão

Previsão inicial: Desvio NÃO será tomado (branch not taken)

Execução especulativa: Emite instruções após o branch antes de conhecer o resultado real

Validação do Branch

- Previsão correta:** Commit normal, sem penalidade
- Previsão errada:** Flush de instruções especulativas, restaura RAT do snapshot, ajusta PC para alvo correto

O snapshot da RAT guardado no ROB permite recuperação precisa do estado anterior ao branch mispredicted

Interface Gráfica do Simulador

Tela Principal

Programa MIPS simplificado com métricas em tempo real: ciclo, instruções commitadas, IPC e ciclos de bolha

Abas de Visualização

Instruções, RS/ROB, Registradores e Memória organizados para acompanhamento detalhado do estado

Controles de Execução

Modos Step (1 ciclo) e Run (automático) para análise granular ou execução completa do programa

Demonstração Prática: Exemplo Passo a Passo



Métricas de Desempenho Calculadas

100+

Ciclos Totais

Tempo completo de execução do programa medido em ciclos de clock

2

Largura

Instruções emitidas e commitadas por ciclo na arquitetura superescalar

1.5

IPC Médio

Instructions Per Cycle demonstrando eficiência da execução paralela

15

Ciclos de Bolha

Ciclos sem commit mas com instruções pendentes, indicador de contenção

O IPC maior que 1 é evidência direta do ganho proporcionado pela arquitetura superescalar com execução fora de ordem.

Conclusões e Contribuições

Implementação Completa

Simulador didático funcional do algoritmo de Tomasulo com:

- Arquitetura superescalar de largura 2
- Buffer de reordenação (ROB) para commits em ordem
- Especulação de desvio com recovery de misprediction
- Interface gráfica intuitiva para visualização
- Cálculo de métricas de desempenho (IPC, ciclos, bolhas)

Valor Educacional

Ferramenta poderosa para compreender conceitos avançados de arquitetura de computadores de forma prática e visual.

