



## Centro Federal de Educação Tecnológica de Minas Gerais

Departamento de Computação  
Curso de Graduação em Engenharia da Computação  
Laboratório de Arquitetura e Organização de Computadores II  
Profa. Poliana Aparecida Corrêa de Oliveira (poliana@cefetmg.br)

# Prática 3 – Algoritmo Tomasulo

## Data da Entrega: até 26/08/2024 – 25 pontos

O objetivo desse projeto é usar a linguagem Verilog para implementar o algoritmo Tomasulo de despacho simples sem especulação como descrito no livro texto da disciplina teórica (Seções 3.4 e 3.5).

HENNESSY, John L.; PATTERSON, David A. **Computer architecture: a quantitative approach**. 5 ed. Elsevier, 2011.

### Descrição

No algoritmo Tomasulo a execução das instruções é dividida em 3 estágios: despacho, execução e *write back*. Esses três estágios acessam componentes críticos de hardware: o CDB, as estações de reserva (nas quais ocorrem as renomeações) e as unidades funcionais. Você deverá implementar:

- (1) as estações de reserva;
- (2) os estágios do algoritmo;
- (3) as unidades funcionais: **tipo-R (soma/subtração)** e **tipo-I (load/store)**;
- (4) o banco de registradores;
- (5) as interconexões (barramentos e fios) e
- (6) a fila das instruções.

## Preparação

Inicie seu projeto criando todos os blocos básicos. Podem ser utilizados componentes da LPM, como memória de dados, mas não é obrigatório.

Caso seja possível, você também pode usar blocos criados para a Prática 2 ou de semestre anteriores, desde que sejam de sua autoria, portanto, não é permitido o compartilhamento de código com outros grupos.

Nesta versão simplificada do Tomasulo, somente serão realizadas operações de soma, subtração, leitura e escrita em memória de dados. Desta forma, inicialize previamente o banco de registradores e as primeiras posições da memória de dados e instruções.

Implemente uma arquitetura com 3 registradores. As instruções devem ter tamanho de 16 bits. A definição do formato da instrução faz parte do projeto. As unidades funcionais (tipo-R e tipo-I) demandam 3 ciclos para execução de cada instrução (busca, execução e escrita).

Descrição **INICIAL** dos blocos básicos:

- **CDB arbiter:** arbitra qual unidade funcional poderá colocar o dado no CDB e faz o *broadcast* do resultado para o resto das unidades funcionais e o banco de registradores.
- **CDB:** barramento comum de dados que fornece o valor e o rótulo da estação de reserva que produz o dado.
- **Estações de reserva:** armazenam as instruções. Você poderá criar um componente separado para armazenar os registradores renomeados (ex, como uma tabela de renomeações).
- **Unidade funcional:** realizam as operações do tipo-R e do do tipo-I.
- **Memória de dados:** armazenam os dados.
- **Fila de instruções:** Buffer que contém as instruções.
- **Banco de registradores:** Conjunto de registradores que armazenam os dados.

Esta é apenas uma descrição inicial dos componentes. Você deverá ser refinar, apresentar os detalhes e todas as decisões tomadas para criar o seu projeto.

## Por onde começar?

Pense na arquitetura do seu processador em alto nível (projeto *top-down*): quais componentes são necessários, como conectar esses componentes, ou seja, quais sinais são necessários. Em seguida, detalhe as funções de cada um dos componentes. Depois, implemente e simule individualmente cada um deles a fim de verificar o correto funcionamento. Por fim, faça a integração de todos os módulos e simule a versão simplificada do algoritmo de Tomasulo.

Você deverá simular a sua implementação utilizando o código de testes e explicar no relatório as formas de onda que demonstrem o funcionamento correto de cada instrução do código de testes.

## Código de Testes

O código abaixo é uma versão inicial para testes da versão simplificada do algoritmo de Tomasulo. Se necessário, você deve **adicionar** novas instruções para demonstrar o funcionamento da solução proposta para os seguintes casos de teste:

- a) Dependência de dados verdadeira ou RAW (*Read After Write*);
- b) Dependência de saída ou WAW (*Write After Write*);
- c) Antidependência ou WAR (*Write After Read*);
- d) Dependência/hazard estrutural (STALL - unidade funcional cheia);
- e) Conflito no CDB;
- f) Adiantamento na estação de reserva;
- g) Renomeação de registradores.

1	ADD R0, R1, R2
2	SUB R0, R0, R1
3	ADD R2, R0, R1
4	LD R0, 4(R4)
5	LD R1, 8(R4)
6	ADD R2, R0, R1
7	ADD R2, R2, R0
8	SUB R2, R2, R1
9	SW R2, 0(R3)
10	LD R3, 0(R3)

## Submissão

Crie um pacote contendo TODOS os códigos fontes, formas de onda, e o relatório do projeto. O nome do arquivo deve ser “**Pratica3\_nomedoaluno1\_nomedoaluno2.zip**”. Cada grupo deverá submeter apenas um pacote no Moodle.

O relatório deverá incluir os seguintes componentes:

1. Uma **introdução** em alto nível da sua solução para o algoritmo Tomasulo (por favor, não é para copiar a descrição do livro texto 😊).
2. O **projeto** do seu processador, incluindo detalhes necessários dos módulos criados.
3. Faça uma **figura** mostrando os blocos básicos e interconexões.
4. O código de **teste** utilizado e as formas de onda **com uma explicação que mostre o correto funcionamento**.
5. **Dificuldades** encontradas.
6. **Sugestões** de melhorias da prática.
7. **Comentários** adicionais.

## Apresentação em sala

Cada grupo deverá apresentar a estrutura do código e mostrar o código de teste por meio das simulações evidenciando o correto funcionamento da solução proposta para os seguintes casos de testes: **a)** Dependência de dados verdadeira ou RAW (Read After Write); **b)** Dependência de saída ou WAW (Write After Write); **c)** Antidependência ou WAR (Write After Read); **d)** Dependência/hazard estrutural (STALL - unidade funcional cheia); **e)** Conflito no CDB; **f)** Adiantamento na estação de reserva; **g)** Renomeação de registradores.

**Não é necessário a apresentação na placa FPGA.**

## Pontuação

- Qualidade, clareza e organização do código-fonte (30% - 7,5 pontos)
- Simulações com explicações no relatório (40% - 10 pontos)
- Apresentação do código + simulações (20% - 5 pontos)
- Relatório: Qualidade do texto, organização e descrição do projeto (10% - 2,5 pontos)

## **Pontos Extras**

- Inclusão da instrução de desvio e dos demais componentes necessários para o funcionamento – 2 pontos
- Apresentação na placa – 6 pontos
- Despacho duplo das instruções – 6 pontos
- Tomasulo completo com especulação – 12 pontos