

arRISCado

MC851 - Projeto em
Computação 1

Plaquinha 3

165429 - Ângelo Renato Pazin Malaguti

195727 - Claudio dos Santos Júnior

233840 - Elton Cardoso do Nascimento

234720 - Gabriel Costa Kinder

198921 - Iago Caran Aquino

218733 - João Pedro de Moraes Bonucci

Objetivo

Planejamento e protótipo inicial de
um processador RISC-V de 32 bits

Metas

- Conceber um processador RISC-V com o conjunto RV32I com pipeline em Verilog
- Compilar e programar uma FPGA para receber tal processador
- Criar e executar testes unitários e de integração
- Criar um ambiente de execução para a FPGA

Plano

- Elaboração de um módulo em Verilog para cada estágio do Pipeline
- Criação de módulos auxiliares para representar outras partes importantes do processador

Pipeline de referência

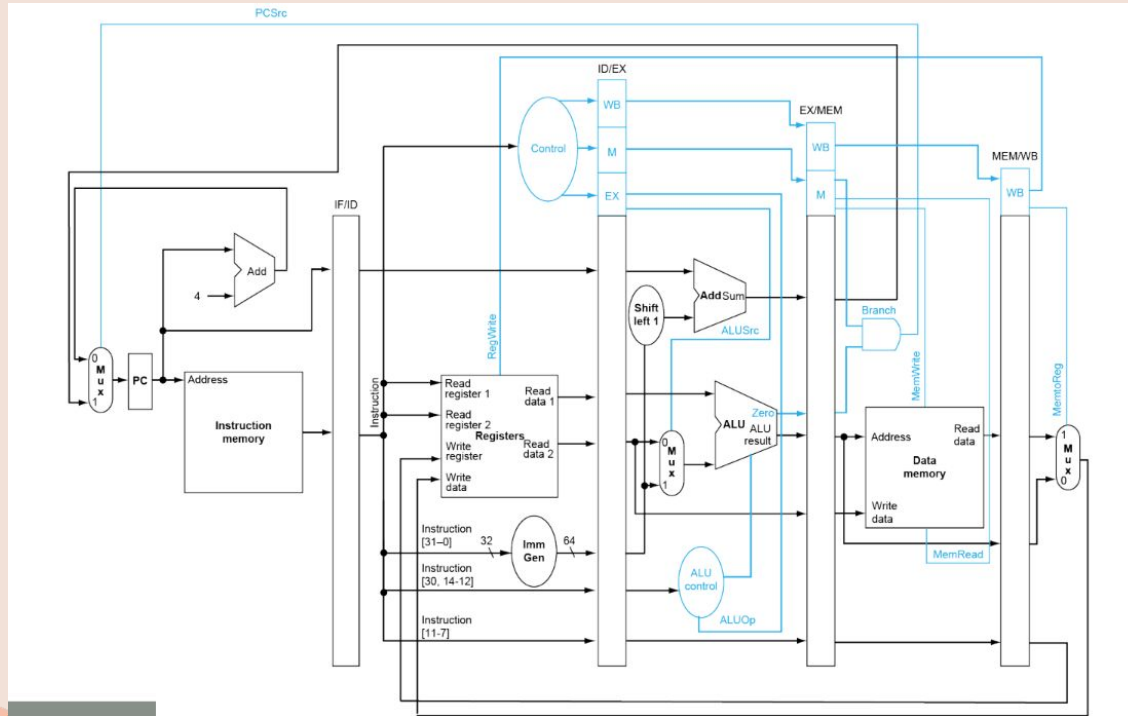
Fetch

Decode

Execute

Mem

Wb



Equipe

Divisão das tarefas em três duplas,
responsáveis por:

Testes

Claudio

Gabriel

Decode, Execute

Ângelo

Iago

Fetch, Mem, W/b

Elton

João

Aí deu errado...



Problemas e novo plano

- A abordagem de divisão anterior se demonstrou problemática, visto inexperiência da equipe com as tecnologias utilizadas
- Novo foco: pipeline com poucas instruções funcionais e testes por módulo implementado
- Por questão de simplicidade e tempo, o programa a ser executado está "hard-coded" na memória ROM, em um pipeline sem detecção de hazards

Outras premissas e simplificações

- Memórias totalmente separadas
- Leitura das memórias em 1 ciclo
- Todas instruções possuem 1 cpi

Resultados

- Etapas de "Fetch" e "Decode" funcionais e integradas
- Validação das etapas de Memória e Write-back
- Etapa de execução ainda não foi concluída
- Utilizados o "iverilog" para validar os módulos usando os testes
- Todos os testes unitários para RV32I foram elaborados
- Todos os códigos estão no GitHub

Dependências

- Integração entre as etapas de "Decode", "Execute" e Memórias
- Mecanismo de controle
- Detecção de hazards
- Combinar os códigos de barramento dentro dos estágios do pipeline
- Leitura em memória unificada

Referências

- PATTERSON, David A.; Hennessy, John L. Computer Organization and Design RISC-V Edition. Elsevier, 2020.
- PATTERSON, David; WATERMAN, Andrew. The RISC-V Reader: An Open Architecture Atlas. 1ª edição. Publicação livre, 2017.
- Component Byte. Verilog Tutorial for Beginners. [s.d.]. Disponível em: <https://www.youtube.com/playlist?list=PLAC_jmBddcjTPEh1UV_ojRJmsx2D9sQXH>. Acesso em: 20 ago. 2023.