

F ▷ E ▷ W

Core

Elves Mateus  
Gabriel Luz  
João Gabriel

# O Projeto



# Motivação

Aprender como implementar um núcleo de processador completo, usando o conteúdo aprendido na disciplina.

# Objetivos

- Construir um núcleo de processador simples, a partir da menor variante do RISC-V.
- Manter as especificações originais da variante da ISA escolhida.
- Fazer o projeto e a implementação de todos os módulos necessários.
- Implementar um pipeline.
- Implementar tratamento de exceção e interrupção.
- Executar um programa com sucesso no processador implementado.
- Executar o processador em uma FPGA.

# Descrição inicial do projeto

- **Embarcado:** será um processador inteiro, objetivando sistemas embarcados;
- **Pipeline Simples:** por se tratar de um sistema embarcado, um pipeline com muitos estágios acaba trazendo complexidade desnecessária para o projeto. Assim, pensamos em cerca de 3 ou 4 estágios;
- **From scratch:** nossa ideia é fazer o máximo possível do processador por nós mesmos, usando no máximo algumas unidades básicas pré-feitas, se precisarmos;

# Descrição inicial do projeto

- **Compatível com a especificação:** queremos seguir o guia da especificação RV32E do RISC-V e, se conseguirmos, implementar também a extensão M, que trata de multiplicações/divisões;
- **Baixo consumo energético:** tentaremos focar em fazer o processador ter um baixo consumo de energia, mesmo que a custo de performance.

# Implementação



# Descrição final do projeto

- O processador continua focado em **embarcados**, por ser simples e usar a variante mais básica do RISC-V.
- **Pipeline** de 3 estágios foi escolhido.
- **From scratch**: todo o processador foi feito a partir do zero , com exceção do decoder, que foi adaptado de outro processador, apenas para evitar o trabalho repetitivo de fazer um *if/case* para cada instrução.
- **Compatível com a especificação**: seguimos a risca os detalhes da especificação, e implementando quase todas as instruções, com exceção, principalmente, da *Store Byte* e *Store Half*, que julgamos dispensáveis para o propósito deste processador, já que exigia algumas operações a mais.
- **Não** foi implementada a extensão de **multiplicação**.

# Descrição final do projeto

- A questão de **economia de energia** não foi levada em consideração durante a implementação, apesar de que é provável que o processador não consuma tanta energia, já que seu clock médio calculado é baixo (cerca de 200MHz (período de clock de 5ps)).

## Lista de instrução não implementadas:

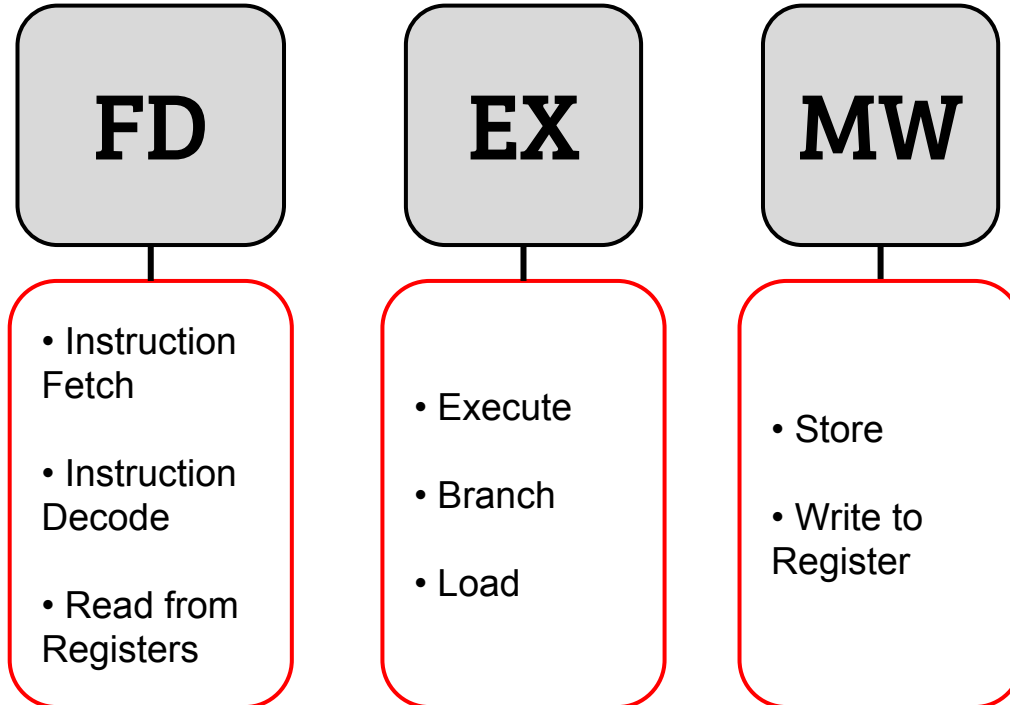
*(apenas as que originalmente eram necessárias para a nossa variante da ISA base)*

- SB
- SH
- FENCE
- FENCE.I
- ECALL
- ESBREAK

OBS.: SB e SH podem ser feitas por software usando combinação de L{B,H,W}, LUI, Shift e SW.



# Visão Geral da Pipeline



# Hazard de Dados?

LW T0, T1, T2;  
SUB T1, T0, T3;

F	E	W		
	F	-	E	W

# Hazard de Dados?

LW  $T_0$ ,  $T_1$ ,  $T_2$ ,  $E$   $W$   
SUB  $T_1$ ,  $T_0$ ,  $T_2$ ,  $E$  -  $E$   $W$

Não há mais stalls por causa de hazards de dados

# Hazard de Datos: Forwarding-Board



# Hazard de Dados: RAW

ADD <span style="color: red;">T0</span> , T1, T2;	F	E	<span style="color: red;">W</span>
SUB T1, T2, T3;	F	E	W
SW <span style="color: red;">T0</span> , 0(T2);		<span style="color: red;">F</span>	E W

- Escrita na subida do clock, e leitura na descida

# Hazard de Controle?

BEQ T0, T1, DEST;	F	E	W	
SUB T1, T2,T3;	F	E	W	
ADD T2, T0, T1;		F	E	W

- Desvio é previsto como não tomado
- Caso não seja, a instrução é morta no W.

# Hazard de Controle?

BEQ T0, T1, DEST;	F	E	W	
JAL T6, DEST2	F	E	W	
ADD T2, T0, T1;		F	E	W

- Desvio é previsto como não tomado
- Caso não seja, a instrução posterior não pode alterar o PC

# Hazard de Controle-BH2B

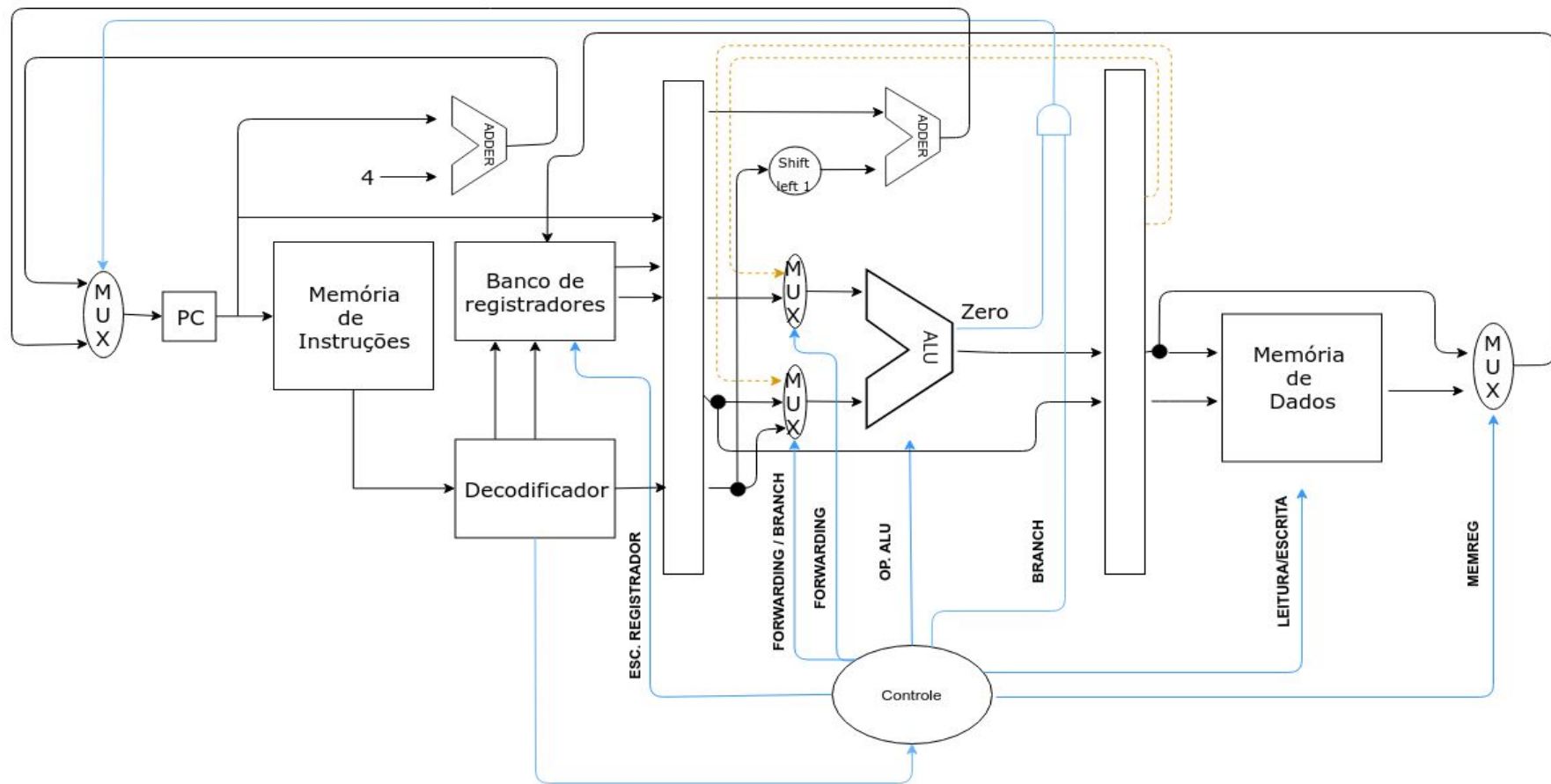
Último Branch tomado?

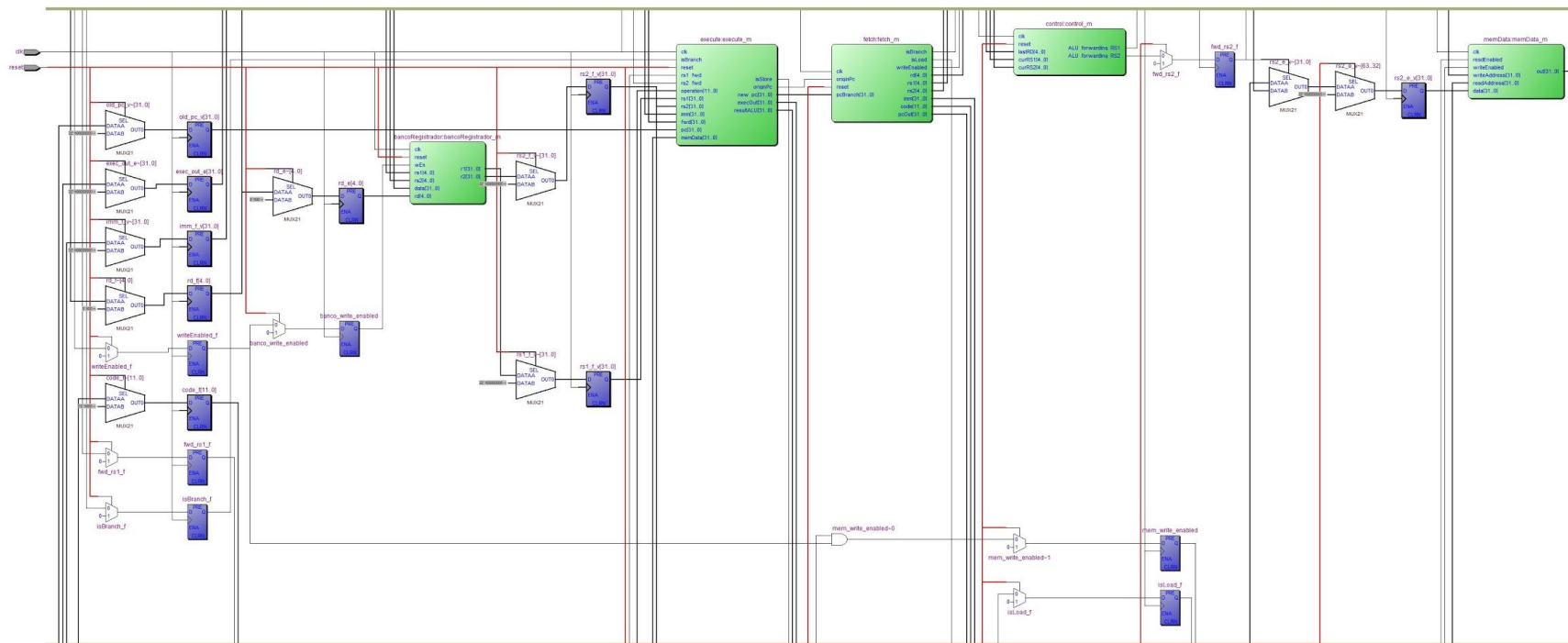
Branch tomado?



# Caminho de Dados







# Considerações de Projeto



# Considerações de Projeto

- **Memória:** Matriz de registradores e pequena
- **Memória de Dados e Instruções** separadas
- **Estágio Write:** implícito

# Objetivos cumpridos e dificuldades

- Construir um núcleo de processador simples, a partir da menor variante do RISC-V - ok!
- Manter as especificações originais da variante da ISA escolhida - ok, com ressalvas.
- Fazer o projeto e a implementação de todos os módulos necessários - ok!
- Implementar um pipeline. - ok!
- Implementar tratamento de exceção e interrupção. - não foi cumprido
- Executar um programa com sucesso no processador implementado - ok, c/ ressalvas.
- Executar o processador em uma FPGA

- ...

## Dificuldades

As principais dificuldades enfrentadas foram na separação dos estágios, na sincronização dos estágios e nos encaminhamentos, apesar de que todas as outras etapas também foram bem desafiadoras e difíceis.

Além disso, enfrentamos diversos problemas com a FPGA na parte final do trabalho, já que apesar de que alguns módulos isolados tenham funcionado, não conseguimos executar o core inteiro nela.

F ▷ E ▷ W

Core