

arRISCado

MC851 - Projeto em
Computação 1

FPGA 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

Ampliação para RV32IMAC com
suporte para 2 periféricos

(E fazer a primeira e segunda parte funcionar)

Metas

1. Implementação do conjunto RV32IMAC
2. Adição de periférico para o processador
3. Utilizar o ambiente de execução para o periférico
4. Demonstrar código que use o periférico em FPGA

Pendências Anteriores

1. Corrigir problemas da pipeline na FPGA
2. Corrigir bugs da integração do periférico
3. Criação de um script para execução do periférico
4. Adicionar as instruções do conjunto RV32MA
5. Implementação da Cache L1

Pipeline

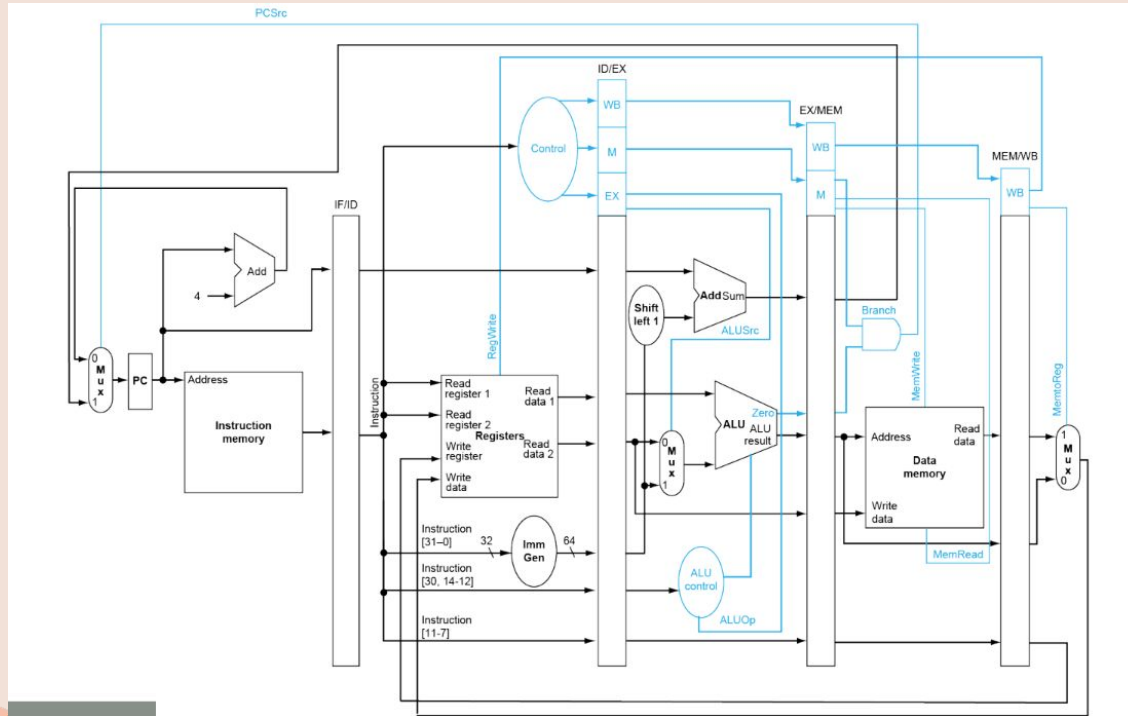
Fetch

Decode

Execute

Mem

Wb



Pipeline

- Funcionamento dos Branches
 - Hazard de controle não corrigido
- Elaboração e refinamento de testes
- Ambiente de execução funcional

Hazards

- Adição de Data forwarding para resolver Hazards do tipo RAW
- Implementação do Stall para resolução de saltos, adicionando bolhas de execução
- Hazards estruturais ainda não são um problema devido as memórias separadas e inexistência de paralelismo

UART

- Otimização do uso de SLICES da FPGA
- Redução do limite máximo de instruções
 - Com fácil alteração para modificações futuras
- Solidificação da interface idêntica à ROM para testes

Testbenches

- Reformulação dos testbenches
- Documentação e correção de erros
- Aperfeiçoamento dos testes
- Desenvolvemos testes para validar as instruções do tipo RV32IM

Testes com as instruções

Step 0	Step 1	Step 2	Step 3
IF 0 instr: 20000537 pc: 4	IF 1 instr: 00000013 pc: 8	IF 2 instr: 00000013 pc: 12	IF 3 instr: 00000013 pc: 16
Decode -1 DECODE IN----- DECODE INTERNAL----- instruction: 00100013 DECODE OUT----- imm: 1 rs1: 0 rs2: 1 shamt: 1 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 0 DECODE IN----- DECODE INTERNAL----- instruction: 20000537 DECODE OUT----- imm: 536870912 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 16 opcode: 55 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 10 AluSrc: 1 AluOp: 4 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 1 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 2 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0

Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Step 10	Step 11
IF 4 instr: 00000013 pc: 20	IF 5 instr: 00000013 pc: 24	IF 6 instr: 00000013 pc: 28	IF 7 instr: 00000013 pc: 32	IF 8 instr: 00000013 pc: 36	IF 9 instr: 00000013 pc: 40	IF 10 instr: 00000013 pc: 44	IF 11 instr: 00000013 pc: 48
Decode 3 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 4 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 5 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 6 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 7 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 8 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 9 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0	Decode 10 DECODE IN----- DECODE INTERNAL----- instruction: 00000013 DECODE OUT----- imm: 0 rs1: 0 rs2: 0 shamt: 0 func3: 0 func7: 0 opcode: 19 MemWrite: 0 MemRead: 0 RegWrite: 1 RegDest: 0 AluSrc: 1 AluOp: 2 AluControl: 2 Branch: 0 MemToReg: 0 RegDataSrc: 0

Análise passo-a-passo procurando o problema

Suporte a RV32IM

- Adição das instruções nos módulos de decode, execute e na ALU
- As alterações de legibilidade do código foram finalmente adicionadas a main.
- No execute, os sinais de controle AluOp e AluSrc foram utilizados para determinar os operandos a serem enviados para a ALU
- O sinal AluControl é empregado para determinar a operação aritmética ou lógica a ser executada entre os dois operandos
- RV32M ainda não está funcionando 100%, apenas *mul* e *div*.

Implementação Periférico

- Periféricos mapeados na memória
 - Prefixo do endereço
 - 000: memória principal
 - 001-111: periféricos
 - Até 7 dispositivos
-
- Implementado e mergeado com a pipeline principal!

Aí deu errado...

parte 3

O que será que deu errado dessa vez ??



Pendências

- Garantir funcionamento do conjunto RV32M
- Adicionar as instruções do conjunto RV32AC
- Finalizar a Cache
- Finalizar script para utilizar a toolchain da Gowin
- Utilizar os testes para validar alterações no pipeline
- Transformar a UART em um periférico?

Aprendizado

- WaveTrace é mais fácil para debugar o projeto;
- Usar um único modelo de simulação
- Yosys ainda não está pronto
- Planejar todo o comportamento dos sinais antes de desenhar os módulos
- Verilog complica projetos com muitos arquivos