UNIVERSIDADE FEDERAL DE PERNAMBUCO - UFPE
CENTRO DE INFORMÁTICA - CIn

RELATÓRIO DE PROJETO

JOSÉ ALEX DE CARVALHO - JAC
JOSÉ IZAIAS DA SILVA JÚNIOR - JISJ
JULIANA SILVA - JS3
LEANDRO VYNICIUS RAMOS DA SILVA - LVRS

RECIFE, 2023

ÍNDICE

1. Introdução

Este relatório tem como objetivo descrever nossa experiência ao realizar o projeto de implementação de um processador de arquitetura RISC-V no modelo Pipeline para disciplina de Infraestrutura de Hardware ministrada pela professora Edna Natividade, para assim avaliar nosso aprendizado na disciplina em questão.

Neste relatório apresentaremos nossas escolhas, resultados obtidos e dificuldades que encontramos durante a implementação das instruções na linguagem de descrição de hardware SystemVerilog.

2. Materiais e Métodos

Para a realização deste projeto utilizamos o projeto base disponibilizado pelos monitores da disciplina e implementamos as instruções demandadas a partir do mesmo. Para a análise e simulação das implementações e do projeto base foi utilizado o programa de licença gratuita ModelSim, que se trata de um ambiente para simulação para linguagens de descrição de hardware como a utilizada neste projeto, o SystemVerilog. Foi utilizada a plataforma de hospedagem GitHub para a integração do projeto com toda a equipe.

3. Implementações

Apresentaremos as implementações realizadas em cada instrução a seguir, aqui está o link para o repositório de nosso fork do projeto original no GitHub, onde armazenamos nossos resultados e códigos.

3.1 Instruções Registrador - Registrador

Para implementar as funcionalidades dessa categoria, foi preciso definir no controlador da ALU um caso que levasse em conta as partes do OPcode que são específicas de cada instrução (funct3 e funct7) para permitir o tratamento da mesma na ALU. Nesse componente, por sua vez, realiza-se a execução da operação específica da instrução, com base no resultado provindo do controlador.

3.1.1 SUB

Utilizamos o caso da operação ser (0110) e difnimos a saída ALUResult = SrcA - SrcB.

3.1.2 SLT

Utilizamos o caso da operação ser (1100) e separamos a saída em três casos:

- Quando o primeiro registrador tem um valor negativo e o segundo um valor positivo, o primeiro sempre será menor, logo ALUResult = 1;
- Quando o primeiro registrador tem um valor positivo e o segundo tem valor negativo, o primeiro nunca será menor que o segundo, logo ALUResult = 0;
- Para outra configuração diferente dos dois casos citados acima, usamos um operador ternário que verifica se (SrcA < SrcB) e atribui o resultado a ALUResult.

3.1.3 XOR
Utilizamos o caso da operação ser (0101) e definimos a saída ALUResult = SrcA ^ SrcB;

3.1.4 OR
Utilizamos o caso da operação ser (0001) e definimos a saída ALUResult = SrcA | SrcB;

3.2 Instruções Registrador - Imediato

Para implementar as instruções que requerem um imediato, nós modificamos o Gerador de Imediato para reconhecer as instruções I-Type e o controlador para para permitir o uso da ALU e da escrita do Registrador. Adicionamos na Alu novos casos para as instruções. No ALUController, usamos o código da operação, a Funct3 e Funct7 de cada instrução para encaminhar para o caso da Alu correspondente a instrução. O registrador da instrução fica no fio “SrcA” e o imediato fica no fio “SrcB”.

3.2.1 ADDI
Utilizamos o caso da operação ser (0010) e definimos a saída ALUResult = SrcA + SrcB;

3.2.2 SLTI
Utilizamos o caso da operação ser (1100) e separamos a saída em três casos:

- Quando o registrador tem um valor negativo e o Imediato um valor positivo, o registrador sempre será menor que o Imediato, logo ALUResult = 1;
- Quando o registrador tem um valor positivo e o imediato tem valor negativo, o registrador nunca será menor que o imediato, logo ALUResult = 0;
- Para outra configuração diferente dos dois casos citados acima, usamos um operador ternário que verifica se (SrcA < SrcB) e atribui o resultado a ALUResult.

3.2.3 SLLI
Utilizamos o caso da operação ser (0111) e o operador shift lógico à esquerda do SystemVerilog e atribuímos o resultado a ALUResult na ALU.

3.2.4 SRLI
Utilizamos o caso da operação ser (1111) e o operador shift lógico à direita do SystemVerilog e atribuímos o resultado a ALUResult na ALU.

3.2.5 SRAI
Utilizamos o caso da operação ser (1110) e o operador de shift lógico aritmético à direita do SystemVerilog e atribuímos o resultado a ALUResult na ALU.

3.2.6 LUI
Utilizamos o caso da operação ser (1011), retorna o valor de ScrB e põe 0 nos últimos 12 bits e atribuímos o resultado a ALUResult na ALU.

3.3 Instruções de Transferência de Controle

Desvio indondicional: Para implementar as funções de desvio incondicional, o controller foi modificado para reconhecer as intruções e enviar o sinal (um para indicar que é um desvio incondicionl e outro para indicar que o resultado da busca nos registradores deve ser usado)para o data path o sinal “é computado” na segunda parte do pipeline e é guardado no banco B de registradores, na terceira parte do pipeline foi adicionado um mux o qual seleciona o valor do sourceA de acordo com o sinal, esse valor é o que será guardado no registrador, e será o valor vindo do registrador ou o próprio Pc+4. O Branch unit também foi adaptado, ele agora recebe o valor obtido do registrador e os sinais do tipo-J, caso o jalr seja identificado o valor do próximo pc é atualizado para o valor guardado no registrador + Imm.

3.3.1 JAL

O sinal jal é ativo, o pc+4 é escrito nos registradores e o próximo pc atualizado para Pc+Imm.

3.3.2 JALR

Os dois sinais jal e jalr são ativos, o pc+4 é escrito nos registradores e o próximo pc é atualizado para FAmux_Result+Imm.

Desvio Condicional: Os desvios condicionais foram implementados modificando o “Controller” para reconhecer as instruções do tipo Branch, a “ALU” e “ALUController” para enviar o resultado das operações que decidem se ocorrerá desvio e a “Branch Unit” para mandar o sinal vindo da ALU e, se ocorrer desvio, atribuir PC + Imediato ao endereço de desvio.

3.3.3 BNE

Utilizamos o caso da operação ser (1001) e o operador lógico “ != ” do SystemVerilog para verificar se os valores nos registradores são diferentes.

3.3.4 BLT

Utilizamos o mesmo caso de operação do SLT e SLTI já que suas operações na “ALU” são equivalentes.

3.3.5 BGE

Utilizamos o caso da operação ser (1010) e separamos a saída em três casos, similar ao que foi feito para verificar se o valor de SrcA é menor que SrcB:

- Quando o primeiro registrador tem um valor negativo e o segundo registrador tem um valor positivo, o primeiro registrador nunca será maior que o segundo, logo ALUResult = 0;
- Quando o primeiro registrador tem um valor positivo e o segundo registrador tem valor negativo, o primeiro registrador sempre será maior que o segundo, logo ALUResult = 1;
- Para outra configuração diferente dos dois casos citados acima, usamos um operador ternário que verifica se (SrcA >= SrcB) e atribui o resultado a ALUResult.

3.4 Instruções de Load e Store

Load: As funções de load envolvem capturar diferentes números de bits, a escolha de implementação foi capturar a palavra completa (32 bits) da memória e resetar os bits que não eram necessários estendendo o bit de sinal (fora a lbu que despreza o sinal), assim só é escrito o que é necessário.

3.4.1 LB

Considera os byte menos significativos e o sinal.

3.4.2 LH

Considera os 2 bytes menos significativos e o sinal.

3.4.3 LBU

Considera o byte menos significativo e desconsidera o sinal.

Store: Para implementar os stores nós modificamos o sinal “Wr” que é enviado para saber qual Byte vai ser escrito na memória, no caso de uma half, desativamos o sinal de escrita para os dois Bytes mais significativos, assim só os 2 bytes menos significativos serão escritos.

3.4.4 SB

Escreve o byte menos significativo da palavra passada na memória.

3.4.5 SH

Escreve os 2 bytes menos significativos da palavra passada na memória.

3.5 Pseudo-Instrução

3.5.1 HALT

O halt foi implementado usando um novo módulo chamado HaltUnit, esse módulo pode controlar o opcode que vai para o controller, o opcode do halt foi definido para 0000001, caso um halt seja detectado a HaltUnit passa a transformar todos os opcodes que envia para o controller para 0000000, que não tem função nenhuma. Como o assembler não estava reconhecendo o halt, testamos mudando diretamente o intruction.mif.

4. Simulações de Referência

Nesta seção, apresentaremos as simulações realizadas em cada uma das instruções implementadas neste projeto, as simulações foram previamente informadas no repositório de referência.

4.1 Simulações ALU

Estas simulações foram fornecidas no repositório de referência para o teste das instruções que utilizam a ALU para realizar as suas operações.

4.1.1 ADDI, OR, ADD, SLL, SRL, SRA, SLT, SLTU, SLTI, SLTIU, SLLI, SRLI, SRAI, XORI, ORI, ANDI, XOR

```
addi x0,x0,0
addi x1,x0,8
addi x2,x0,4
or x3,x1,x2
or x4,x2,x0
add x6,x4,x2
addi x4,x0,2
addi x5,x0,-2
sll x18,x1,x4
srl x19,x5,x4
sra x20,x5,x4
slt x21,x1,x2
slt x22,x2,x1
sltu x23,x5,x1
sltu x24,x1,x5
slti x25,x1,8
slti x26,x1,16
addi x5,x0,-4
sltiu x27,x1,-2
sltiu x28,x5,-2
slli x29,x5,1
srli x30,x5,1
srai x31,x5,1
xori x6,x1,10
ori x7,x1,2
andi x8,x1,10
xor x9,x1,x2
```

Código Fornecido

```
45: Register [ 0] written with value: [00000000] | [         0]
55: Register [ 1] written with value: [00000008] | [         8]
65: Register [ 2] written with value: [00000004] | [         4]
75: Register [ 3] written with value: [0000000c] | [        12]
85: Register [ 4] written with value: [00000004] | [         4]
95: Register [ 6] written with value: [00000008] | [         8]
105: Register [ 4] written with value: [00000002] | [         2]
115: Register [ 5] written with value: [fffffffe] | [4294967294]
125: Register [18] written with value: [00000020] | [        32]
135: Register [19] written with value: [3fffffff] | [1073741823]
145: Register [20] written with value: [ffffffff] | [4294967295]
155: Register [21] written with value: [00000000] | [         0]
165: Register [22] written with value: [00000001] | [         1]
175: Register [23] written with value: [00000000] | [         0]
185: Register [24] written with value: [00000001] | [         1]
195: Register [25] written with value: [00000000] | [         0]
205: Register [26] written with value: [00000001] | [         1]
215: Register [ 5] written with value: [fffffffc] | [4294967292]
225: Register [27] written with value: [00000001] | [         1]
235: Register [28] written with value: [00000001] | [         1]
245: Register [29] written with value: [fffffff8] | [4294967288]
255: Register [30] written with value: [7ffffffe] | [2147483646]
265: Register [31] written with value: [fffffffe] | [4294967294]
275: Register [ 6] written with value: [00000002] | [         2]
285: Register [ 7] written with value: [0000000a] | [        10]
295: Register [ 8] written with value: [00000008] | [         8]
305: Register [ 9] written with value: [0000000c] | [        12]
```

Resultado esperado

instructions.txt

```
1  addi x0,x0,0
2  addi x1,x0,8
3  addi x2,x0,4
4  or x3,x1,x2
5  or x4,x2,x0
6  add x6,x4,x2
7  addi x4,x0,2
8  addi x5,x0,-2
9  slt x21,x1,x2
10 slt x22,x2,x1
11 slti x25,x1,8
12 slti x26,x1,16
13 addi x5,x0,-4
14 slli x29,x5,1
15 srli x30,x5,1
16 srai x31,x5,1
17 xor x9,x1,x2
18
```

Código após tirar instruções não descritas no projeto

```
Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_defaul
                45: Register [ 0] written with value: [00000000] | [         0]
                55: Register [ 1] written with value: [00000008] | [         8]
                65: Register [ 2] written with value: [00000004] | [         4]
                75: Register [ 3] written with value: [0000000c] | [        12]
                85: Register [ 4] written with value: [00000004] | [         4]
                95: Register [ 6] written with value: [00000008] | [         8]
               105: Register [ 4] written with value: [00000002] | [         2]
               115: Register [ 5] written with value: [fffffffe] | [4294967294]
               125: Register [21] written with value: [00000000] | [         0]
               135: Register [22] written with value: [00000001] | [         1]
               145: Register [25] written with value: [00000000] | [         0]
               155: Register [26] written with value: [00000001] | [         1]
               165: Register [ 5] written with value: [fffffffc] | [4294967292]
               175: Register [29] written with value: [fffffff8] | [4294967288]
               185: Register [30] written with value: [7ffffffe] | [2147483646]
               195: Register [31] written with value: [7ffffffe] | [2147483646]
               205: Register [ 9] written with value: [0000000c] | [        12]
```

Resultado obtido

4.1.2 SUB, AND, LUI

```
addi x1,x0,8
sub x6,x6,x1
and x7,x6,x1
lui x6,3
```

Código Fornecido

```
45: Register [ 1] written with value: [00000008] | [        8]
55: Register [ 6] written with value: [fffffff8] | [4294967288]
65: Register [ 7] written with value: [00000008] | [        8]
75: Register [ 6] written with value: [00003000] | [    12288]
```

Resultado Esperado

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                45: Register [ 1] written with value: [00000008] | [        8]
#
#                55: Register [ 6] written with value: [fffffff8] | [4294967288]
#
#                65: Register [ 7] written with value: [00000008] | [        8]
#
#                75: Register [ 6] written with value: [00003000] | [    12288]
#
```

Resultado obtido

4.2 Simulações LOAD

Estas simulações foram fornecidas no repositório de referência para o teste das instruções que utilizam o acesso à memória e carregam valores para realizar operações.

4.2.1 LB, LH, LW

```
addi x7,x0,1
addi x2,x0,4
or x4,x2,x0
lb x6,0(x7)
add x6,x4,x0
lb x7,0(x6)
lh x8,0(x6)
lw x9,0(x6)
```

Código Fornecido

```
45: Register [ 7] written with value: [00000001] | [        1]
55: Register [ 2] written with value: [00000004] | [        4]
65: Memory [  1] read with value: [xxxxxxxx] | [        x]
65: Register [ 4] written with value: [00000004] | [        4]
65: Memory [  1] read with value: [ffffffff] | [4294967295]
70: Memory [  1] read with value: [ffffff8f] | [4294967183]
75: Register [ 6] written with value: [ffffff8f] | [4294967183]
85: Register [ 6] written with value: [00000008] | [        8]
85: Memory [  8] read with value: [ffffff8f] | [4294967183]
85: Memory [  8] read with value: [fffffffb] | [4294967291]
95: Register [ 7] written with value: [fffffffb] | [4294967291]
95: Memory [  8] read with value: [ffffaafb] | [4294945531]
105: Register [ 8] written with value: [ffffaafb] | [4294945531]
105: Memory [  8] read with value: [0001aafb] | [    109307]
115: Register [ 9] written with value: [0001aafb] | [    109307]
```

Resultado esperado

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                    45: Register [ 7] written with value: [00000001] | [         1]
#
#                    55: Register [ 2] written with value: [00000004] | [         4]
#
#                    65: Memory [  1] read with value: [xxxxxxxx] | [         x]
#
#                    65: Register [ 4] written with value: [00000004] | [         4]
#
#                    65: Memory [  1] read with value: [00000003] | [         3]
#
#                    70: Memory [  1] read with value: [00000000] | [         0]
#
#                    75: Register [ 6] written with value: [00000000] | [         0]
#
#                    85: Register [ 6] written with value: [00000004] | [         4]
#
#                    85: Memory [  4] read with value: [00000000] | [         0]
#
#                    85: Memory [  4] read with value: [00000003] | [         3]
#
#                    95: Register [ 7] written with value: [00000003] | [         3]
#
#                   105: Register [ 8] written with value: [00000003] | [         3]
#
#                   115: Register [ 9] written with value: [00000003] | [         3]
```

Resultado obtido

4.2.2 LBU, LHU

```
addi x7,x0,1
addi x2,x0,4
or x4,x2,x0
lb x6,0(x7)
add x6,x4,x0
lbu x7,0(x6)
lhu x8,0(x6)
```

Código Fornecido

```
45: Register [ 7] written with value: [00000001] | [         1]
55: Register [ 2] written with value: [00000004] | [         4]
65: Memory [  1] read with value: [xxxxxxxx] | [         x]
65: Register [ 4] written with value: [00000004] | [         4]
65: Memory [  1] read with value: [ffffffff] | [4294967295]
70: Memory [  1] read with value: [ffffff8f] | [4294967183]
75: Register [ 6] written with value: [ffffff8f] | [4294967183]
85: Register [ 6] written with value: [00000004] | [         4]
85: Memory [  8] read with value: [ffffff8f] | [4294967183]
85: Memory [  8] read with value: [000000fb] | [       251]
95: Register [ 7] written with value: [000000fb] | [       251]
95: Memory [  8] read with value: [0000aafb] | [     43771]
105: Register [ 8] written with value: [0000aafb] | [     43771]
```

Resultado Esperado

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                  45: Register [ 7] written with value: [00000001] | [         1]
#
#                  55: Register [ 2] written with value: [00000004] | [         4]
#
#                  65: Memory [  1] read with value: [xxxxxxxx] | [         x]
#
#                  65: Register [ 4] written with value: [00000004] | [         4]
#
#                  65: Memory [  1] read with value: [00000003] | [         3]
#
#                  70: Memory [  1] read with value: [00000000] | [         0]
#
#                  75: Register [ 6] written with value: [00000000] | [         0]
#
#                  85: Register [ 6] written with value: [00000004] | [         4]
#
#                  85: Memory [  4] read with value: [00000000] | [         0]
#
#                  85: Memory [  4] read with value: [00000003] | [         3]
#
#                  95: Register [ 7] written with value: [00000003] | [         3]
#
#                 105: Register [ 8] written with value: [00000003] | [         3]
#
```

Resultado obtido

4.3 Simulações de Desvio

Estas simulações foram fornecidas no repositório de referência para o teste das instruções que realizam desvios no código para as suas operações.

4.3.1 JAL, BEQ (taken)

```
addi x7,x0,1
addi x2,x0,4
jal x10,8
or x4,x2,x0
add x6,x4,x2
addi x7,x0,1
addi x8,x0,2
beq x7,x7,-8
or x4,x2,x0
```

Código Fornecido

```
45: Register [ 7] written with value: [00000001] | [         1]
55: Register [ 2] written with value: [00000004] | [         4]
65: Register [10] written with value: [0000000c] | [        12]
95: Register [ 6] written with value: [00000004] | [         4]
105: Register [ 7] written with value: [00000001] | [         1]
115: Register [ 8] written with value: [00000002] | [         2]
155: Register [ 7] written with value: [00000001] | [         1]
165: Register [ 8] written with value: [00000002] | [         2]
205: Register [ 7] written with value: [00000001] | [         1]
215: Register [ 8] written with value: [00000002] | [         2]
255: Register [ 7] written with value: [00000001] | [         1]
265: Register [ 8] written with value: [00000002] | [         2]
305: Register [ 7] written with value: [00000001] | [         1]
315: Register [ 8] written with value: [00000002] | [         2]
355: Register [ 7] written with value: [00000001] | [         1]
365: Register [ 8] written with value: [00000002] | [         2]
405: Register [ 7] written with value: [00000001] | [         1]
415: Register [ 8] written with value: [00000002] | [         2]
455: Register [ 7] written with value: [00000001] | [         1]
465: Register [ 8] written with value: [00000002] | [         2]
505: Register [ 7] written with value: [00000001] | [         1]
```

Resultado Esperado

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                   45: Register [ 7] written with value: [00000001] | [          1]
#
#                   55: Register [ 2] written with value: [00000004] | [          4]
#
#                   65: Register [10] written with value: [0000000c] | [         12]
#
#                   95: Register [ 6] written with value: [00000004] | [          4]
#
#                  105: Register [ 7] written with value: [00000001] | [          1]
#
#                  115: Register [ 8] written with value: [00000002] | [          2]
#
#                  155: Register [ 7] written with value: [00000001] | [          1]
#
#                  165: Register [ 8] written with value: [00000002] | [          2]
#
#                  205: Register [ 7] written with value: [00000001] | [          1]
#
#                  215: Register [ 8] written with value: [00000002] | [          2]
#
#                  255: Register [ 7] written with value: [00000001] | [          1]
#
#                  265: Register [ 8] written with value: [00000002] | [          2]
#
#                  305: Register [ 7] written with value: [00000001] | [          1]
#
#                  315: Register [ 8] written with value: [00000002] | [          2]
#
#                  355: Register [ 7] written with value: [00000001] | [          1]
#
#                  365: Register [ 8] written with value: [00000002] | [          2]
#
#                  405: Register [ 7] written with value: [00000001] | [          1]
#
#                  415: Register [ 8] written with value: [00000002] | [          2]
#
#                  455: Register [ 7] written with value: [00000001] | [          1]
#
#                  465: Register [ 8] written with value: [00000002] | [          2]
#
#                  505: Register [ 7] written with value: [00000001] | [          1]
#
```

Resultado obtido

4.3.2 BEQ (not taken)

*Já implementado

```
addi x7,x0,1
addi x2,x0,4
jal x10,8
or x4,x2,x0
add x6,x4,x2
addi x7,x0,1
addi x8,x0,2
beq x8,x7,-8
or x4,x2,x0
```

Código Fornecido

```
45: Register [ 7] written with value: [00000001] | [          1]
55: Register [ 2] written with value: [00000004] | [          4]
65: Register [10] written with value: [0000000c] | [         12]
95: Register [ 6] written with value: [00000004] | [          4]
105: Register [ 7] written with value: [00000001] | [          1]
115: Register [ 8] written with value: [00000002] | [          2]
135: Register [ 4] written with value: [00000004] | [          4]
```

Resultado Esperado

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                   45: Register [ 7] written with value: [00000001] | [         1]
#
#                   55: Register [ 2] written with value: [00000004] | [         4]
#
#                   65: Register [10] written with value: [0000000c] | [        12]
#
#                   95: Register [ 6] written with value: [00000004] | [         4]
#
#                  105: Register [ 7] written with value: [00000001] | [         1]
#
#                  115: Register [ 8] written with value: [00000002] | [         2]
#
#                  135: Register [ 4] written with value: [00000004] | [         4]
#
```

Resultado obtido

4.3.2 BNE (taken)

```
addi x7,x0,1
addi x2,x0,4
jal x10,8
or x4,x2,x0
add x6,x4,x2
addi x7,x0,1
addi x8,x0,2
bne x8,x7,-8
or x4,x2,x0
```

Código Fornecido

```
45: Register [ 7] written with value: [00000001] | [         1]
55: Register [ 2] written with value: [00000004] | [         4]
65: Register [10] written with value: [0000000c] | [        12]
95: Register [ 6] written with value: [00000004] | [         4]
105: Register [ 7] written with value: [00000001] | [         1]
115: Register [ 8] written with value: [00000002] | [         2]
155: Register [ 7] written with value: [00000001] | [         1]
165: Register [ 8] written with value: [00000002] | [         2]
205: Register [ 7] written with value: [00000001] | [         1]
215: Register [ 8] written with value: [00000002] | [         2]
255: Register [ 7] written with value: [00000001] | [         1]
265: Register [ 8] written with value: [00000002] | [         2]
305: Register [ 7] written with value: [00000001] | [         1]
315: Register [ 8] written with value: [00000002] | [         2]
355: Register [ 7] written with value: [00000001] | [         1]
365: Register [ 8] written with value: [00000002] | [         2]
405: Register [ 7] written with value: [00000001] | [         1]
415: Register [ 8] written with value: [00000002] | [         2]
455: Register [ 7] written with value: [00000001] | [         1]
465: Register [ 8] written with value: [00000002] | [         2]
505: Register [ 7] written with value: [00000001] | [         1]
```

Resultado Esperado

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                   45: Register [ 7] written with value: [00000001] | [         1]
#
#                   55: Register [ 2] written with value: [00000004] | [         4]
#
#                   65: Register [10] written with value: [0000000c] | [        12]
#
#                   95: Register [ 6] written with value: [00000004] | [         4]
#
#                  105: Register [ 7] written with value: [00000001] | [         1]
#
#                  115: Register [ 8] written with value: [00000002] | [         2]
#
#                  155: Register [ 7] written with value: [00000001] | [         1]
#
#                  165: Register [ 8] written with value: [00000002] | [         2]
#
#                  205: Register [ 7] written with value: [00000001] | [         1]
#
#                  215: Register [ 8] written with value: [00000002] | [         2]
#
#                  255: Register [ 7] written with value: [00000001] | [         1]
#
#                  265: Register [ 8] written with value: [00000002] | [         2]
#
#                  305: Register [ 7] written with value: [00000001] | [         1]
#
#                  315: Register [ 8] written with value: [00000002] | [         2]
#
#                  355: Register [ 7] written with value: [00000001] | [         1]
#
#                  365: Register [ 8] written with value: [00000002] | [         2]
#
#                  405: Register [ 7] written with value: [00000001] | [         1]
#
#                  415: Register [ 8] written with value: [00000002] | [         2]
#
#                  455: Register [ 7] written with value: [00000001] | [         1]
#
#                  465: Register [ 8] written with value: [00000002] | [         2]
#
#                  505: Register [ 7] written with value: [00000001] | [         1]
#
```

Resultado obtido

4.3.3 BLT (taken)

```
addi x7,x0,1
addi x2,x0,4
jal x10,8
or x4,x2,x0
add x6,x4,x2
addi x7,x0,2
addi x8,x0,1
blt x8,x7,-8
or x4,x2,x0
```

Código Fornecido

```
45: Register [ 7] written with value: [00000001] | [         1]
55: Register [ 2] written with value: [00000004] | [         4]
65: Register [10] written with value: [0000000c] | [        12]
95: Register [ 6] written with value: [00000004] | [         4]
105: Register [ 7] written with value: [00000002] | [         2]
115: Register [ 8] written with value: [00000001] | [         1]
155: Register [ 7] written with value: [00000002] | [         2]
165: Register [ 8] written with value: [00000001] | [         1]
205: Register [ 7] written with value: [00000002] | [         2]
215: Register [ 8] written with value: [00000001] | [         1]
255: Register [ 7] written with value: [00000002] | [         2]
265: Register [ 8] written with value: [00000001] | [         1]
305: Register [ 7] written with value: [00000002] | [         2]
315: Register [ 8] written with value: [00000001] | [         1]
355: Register [ 7] written with value: [00000002] | [         2]
365: Register [ 8] written with value: [00000001] | [         1]
405: Register [ 7] written with value: [00000002] | [         2]
415: Register [ 8] written with value: [00000001] | [         1]
455: Register [ 7] written with value: [00000002] | [         2]
465: Register [ 8] written with value: [00000001] | [         1]
505: Register [ 7] written with value: [00000002] | [         2]
```

Resultado Esperado

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                    45: Register [ 7] written with value: [00000001] | [         1]
#
#                    55: Register [ 2] written with value: [00000004] | [         4]
#
#                    65: Register [10] written with value: [0000000c] | [        12]
#
#                    95: Register [ 6] written with value: [00000004] | [         4]
#
#                   105: Register [ 7] written with value: [00000002] | [         2]
#
#                   115: Register [ 8] written with value: [00000001] | [         1]
#
#                   155: Register [ 7] written with value: [00000002] | [         2]
#
#                   165: Register [ 8] written with value: [00000001] | [         1]
#
#                   205: Register [ 7] written with value: [00000002] | [         2]
#
#                   215: Register [ 8] written with value: [00000001] | [         1]
#
#                   255: Register [ 7] written with value: [00000002] | [         2]
#
#                   265: Register [ 8] written with value: [00000001] | [         1]
#
#                   305: Register [ 7] written with value: [00000002] | [         2]
#
#                   315: Register [ 8] written with value: [00000001] | [         1]
#
#                   355: Register [ 7] written with value: [00000002] | [         2]
#
#                   365: Register [ 8] written with value: [00000001] | [         1]
#
#                   405: Register [ 7] written with value: [00000002] | [         2]
#
#                   415: Register [ 8] written with value: [00000001] | [         1]
#
#                   455: Register [ 7] written with value: [00000002] | [         2]
#
#                   465: Register [ 8] written with value: [00000001] | [         1]
#
#                   505: Register [ 7] written with value: [00000002] | [         2]
```

Resultado obtido

4.3.4 BEG (taken)

```
addi x7,x0,1
addi x2,x0,4
jal x10,8
or x4,x2,x0
add x6,x4,x2
addi x7,x0,2
addi x8,x0,1
bge x7,x8,-8
or x4,x2,x0
```

Código Fornecido

```
45: Register [ 7] written with value: [00000001] | [         1]
55: Register [ 2] written with value: [00000004] | [         4]
65: Register [10] written with value: [0000000c] | [        12]
95: Register [ 6] written with value: [00000004] | [         4]
105: Register [ 7] written with value: [00000002] | [         2]
115: Register [ 8] written with value: [00000001] | [         1]
155: Register [ 7] written with value: [00000002] | [         2]
165: Register [ 8] written with value: [00000001] | [         1]
205: Register [ 7] written with value: [00000002] | [         2]
215: Register [ 8] written with value: [00000001] | [         1]
255: Register [ 7] written with value: [00000002] | [         2]
265: Register [ 8] written with value: [00000001] | [         1]
305: Register [ 7] written with value: [00000002] | [         2]
315: Register [ 8] written with value: [00000001] | [         1]
355: Register [ 7] written with value: [00000002] | [         2]
365: Register [ 8] written with value: [00000001] | [         1]
405: Register [ 7] written with value: [00000002] | [         2]
415: Register [ 8] written with value: [00000001] | [         1]
455: Register [ 7] written with value: [00000002] | [         2]
465: Register [ 8] written with value: [00000001] | [         1]
505: Register [ 7] written with value: [00000002] | [         2]
```

Resultado Esperado

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                    45: Register [ 7] written with value: [00000001] | [         1]
#
#                    55: Register [ 2] written with value: [00000004] | [         4]
#
#                    65: Register [10] written with value: [0000000c] | [        12]
#
#                    95: Register [ 6] written with value: [00000004] | [         4]
#
#                   105: Register [ 7] written with value: [00000002] | [         2]
#
#                   115: Register [ 8] written with value: [00000001] | [         1]
#
#                   155: Register [ 7] written with value: [00000002] | [         2]
#
#                   165: Register [ 8] written with value: [00000001] | [         1]
#
#                   205: Register [ 7] written with value: [00000002] | [         2]
#
#                   215: Register [ 8] written with value: [00000001] | [         1]
#
#                   255: Register [ 7] written with value: [00000002] | [         2]
#
#                   265: Register [ 8] written with value: [00000001] | [         1]
#
#                   305: Register [ 7] written with value: [00000002] | [         2]
#
#                   315: Register [ 8] written with value: [00000001] | [         1]
#
#                   355: Register [ 7] written with value: [00000002] | [         2]
#
#                   365: Register [ 8] written with value: [00000001] | [         1]
#
#                   405: Register [ 7] written with value: [00000002] | [         2]
#
#                   415: Register [ 8] written with value: [00000001] | [         1]
#
#                   455: Register [ 7] written with value: [00000002] | [         2]
#
#                   465: Register [ 8] written with value: [00000001] | [         1]
#
#                   505: Register [ 7] written with value: [00000002] | [         2]
```

Resultado obtido

4.3.5 JALR

```
addi x7,x0,-1
sw x7,0(x0)
lw x9,0(x0)
or x4,x2,x0
add x6,x4,x2
jalr x12,x0,12
```

Código Fornecido

```
45: Memory [  0] written with value: [ffffffff] | [4294967295]
45: Register [ 7] written with value: [ffffffff] | [4294967295]
55: Memory [  0] read with value: [xxxxxxxx] | [         x]
55: Memory [  0] read with value: [00000000] | [         0]
60: Memory [  0] read with value: [ffffffff] | [4294967295]
65: Register [ 9] written with value: [ffffffff] | [4294967295]
75: Register [ 4] written with value: [00000000] | [         0]
85: Register [ 6] written with value: [00000000] | [         0]
95: Register [12] written with value: [00000018] | [        24]
125: Register [ 4] written with value: [00000000] | [         0]
135: Register [ 6] written with value: [00000000] | [         0]
145: Register [12] written with value: [00000018] | [        24]
175: Register [ 4] written with value: [00000000] | [         0]
185: Register [ 6] written with value: [00000000] | [         0]
195: Register [12] written with value: [00000018] | [        24]
225: Register [ 4] written with value: [00000000] | [         0]
235: Register [ 6] written with value: [00000000] | [         0]
245: Register [12] written with value: [00000018] | [        24]
275: Register [ 4] written with value: [00000000] | [         0]
285: Register [ 6] written with value: [00000000] | [         0]
295: Register [12] written with value: [00000018] | [        24]
325: Register [ 4] written with value: [00000000] | [         0]
335: Register [ 6] written with value: [00000000] | [         0]
345: Register [12] written with value: [00000018] | [        24]
375: Register [ 4] written with value: [00000000] | [         0]
385: Register [ 6] written with value: [00000000] | [         0]
395: Register [12] written with value: [00000018] | [        24]
425: Register [ 4] written with value: [00000000] | [         0]
435: Register [ 6] written with value: [00000000] | [         0]
445: Register [12] written with value: [00000018] | [        24]
475: Register [ 4] written with value: [00000000] | [         0]
485: Register [ 6] written with value: [00000000] | [         0]
495: Register [12] written with value: [00000018] | [        24]
```

Resultado Esperado

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                   45: Memory [  0] written with value: [ffffffff] | [4294967295]
#
#                   45: Register [ 7] written with value: [ffffffff] | [4294967295]
#
#                   55: Memory [  0] read with value: [xxxxxxxx] | [         x]
#
#                   55: Memory [  0] read with value: [00000001] | [         1]
#
#                   60: Memory [  0] read with value: [ffffffff] | [4294967295]
#
#                   65: Register [ 9] written with value: [ffffffff] | [4294967295]
#
#                   75: Register [ 4] written with value: [00000000] | [         0]
#
#                   85: Register [ 6] written with value: [00000000] | [         0]
#
#                   95: Register [12] written with value: [00000018] | [        24]
#
#                  125: Register [ 4] written with value: [00000000] | [         0]
#
#                  135: Register [ 6] written with value: [00000000] | [         0]
#
#                  145: Register [12] written with value: [00000018] | [        24]
#
#                  175: Register [ 4] written with value: [00000000] | [         0]
#
#                  185: Register [ 6] written with value: [00000000] | [         0]
#
#                  195: Register [12] written with value: [00000018] | [        24]
#
#                  225: Register [ 4] written with value: [00000000] | [         0]
#
#                  235: Register [ 6] written with value: [00000000] | [         0]
#
#                  245: Register [12] written with value: [00000018] | [        24]
#
#                  275: Register [ 4] written with value: [00000000] | [         0]
#
#                  285: Register [ 6] written with value: [00000000] | [         0]
#
#                  295: Register [12] written with value: [00000018] | [        24]
#
#                  325: Register [ 4] written with value: [00000000] | [         0]
#
#                  335: Register [ 6] written with value: [00000000] | [         0]
#
#                  345: Register [12] written with value: [00000018] | [        24]
#
#                  375: Register [ 4] written with value: [00000000] | [         0]
#
#                  385: Register [ 6] written with value: [00000000] | [         0]
#
#                  395: Register [12] written with value: [00000018] | [        24]
#
#                  425: Register [ 4] written with value: [00000000] | [         0]
#
#                  435: Register [ 6] written with value: [00000000] | [         0]
#
#                  445: Register [12] written with value: [00000018] | [        24]
#
#                  475: Register [ 4] written with value: [00000000] | [         0]
#
#                  485: Register [ 6] written with value: [00000000] | [         0]
#
#                  495: Register [12] written with value: [00000018] | [        24]
#
# ** Note: $stop    : verif/tb_top.sv(43)
#    Time: 510 ns  Iteration: 0  Instance: /tb_top
```

Resultado obtido

4.4 Simulações de STORE

Estas simulações foram fornecidas no repositório de referência para o teste das instruções que realizam armazenamentos na memória para as suas operações.

4.4.1 SW

```
addi x7,x0,-1
sw x7,0(x0)
lw x9,0(x0)
```

```
45: Memory [  0] written with value: [ffffffff] | [4294967295]
45: Register [ 7] written with value: [ffffffff] | [4294967295]
55: Memory [  0] read with value: [xxxxxxxx] | [         x]
55: Memory [  0] read with value: [ffffaa80] | [4294945408]
60: Memory [  0] read with value: [ffffffff] | [4294967295]
65: Register [ 9] written with value: [ffffffff] | [4294967295]
```

Código Fornecido *Resultado Esperado*

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                  45: Memory [  0] written with value: [ffffffff] | [4294967295]
#
#                  45: Register [ 7] written with value: [ffffffff] | [4294967295]
#
#                  55: Memory [  0] read with value: [xxxxxxxx] | [         x]
#
#                  55: Memory [  0] read with value: [00000001] | [         1]
#
#                  60: Memory [  0] read with value: [ffffffff] | [4294967295]
#
#                  65: Register [ 9] written with value: [ffffffff] | [4294967295]
#
# ** Note: $stop    : verif/tb_top.sv(43)
#    Time: 510 ns  Iteration: 0  Instance: /tb_top
```

Resultado obtido

4.4.2 SB, SH

```
addi x7,x0,0
sb x7,2(x0)
lw x9,0(x0)
sh x7,2(x0)
lw x8,0(x0)
```

```
45: Memory [  2] written with value: [00000000] | [         0]
45: Register [ 7] written with value: [00000000] | [         0]
55: Memory [  0] read with value: [xxxxxxxx] | [         x]
55: Memory [  0] read with value: [0000ffff] | [     65535]
60: Memory [  0] read with value: [ff00aa80] | [4278233728]
65: Memory [  2] written with value: [00000000] | [         0]
65: Register [ 9] written with value: [ff00aa80] | [4278233728]
75: Memory [  0] read with value: [ff00aa80] | [4278233728]
75: Memory [  0] read with value: [0000ffff] | [     65535]
80: Memory [  0] read with value: [0000aa80] | [     43648]
85: Register [ 8] written with value: [0000aa80] | [     43648]
```

Código Fornecido *Resultado Esperado*

```
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                  45: Memory [  2] written with value: [00000000] | [         0]
#
#                  45: Register [ 7] written with value: [00000000] | [         0]
#
#                  55: Memory [  0] read with value: [xxxxxxxx] | [         x]
#
#                  55: Memory [  0] read with value: [00030000] | [    196608]
#
#                  60: Memory [  0] read with value: [00000000] | [         0]
#
#                  65: Memory [  2] written with value: [00000000] | [         0]
#
#                  65: Register [ 9] written with value: [00000000] | [         0]
#
#                  75: Memory [  0] read with value: [00000000] | [         0]
#
#                  75: Memory [  0] read with value: [00030000] | [    196608]
#
#                  80: Memory [  0] read with value: [00000000] | [         0]
#
#                  85: Register [ 8] written with value: [00000000] | [         0]
#
# ** Note: $stop    : verif/tb_top.sv(43)
```

Resultado obtido

4.5 Halt

A pseudo-instrução Halt foi implementada para realizar a sua funcionalidade de parar o programa quando executada.

```
1   lw x1,0(x0)
2   add x2,x1,x1
3   halt
4   lw x3,4(x0)
5   add x4,x2,x3
6   sub x5,x4,x2
```

Código Utilizado

```
DEPTH = 65536;          -- The size of memory in words
WIDTH = 8;              -- The size of data in bits
ADDRESS_RADIX = DEC;    -- The radix for address values
DATA_RADIX = BIN;       -- The radix for data values
CONTENT                 -- Start of (address: data pairs)
BEGIN

000: 10000011;      -- lw x1,0(x0)
001: 00100000;
002: 00000000;
003: 00000000;

004: 00110011;      -- add x2,x1,x1
005: 10000001;
006: 00010000;
007: 00000000;

008: 00000001;      -- halt
009: 00000000;
010: 00000000;
011: 00000000;

012: 10000011;      -- lw x3,4(x0)
013: 00100001;
014: 01000000;
015: 00000000;

016: 00110011;      -- add x4,x2,x3
017: 00000010;
018: 00110001;
019: 00000000;

020: 10110011;      -- sub x5,x4,x2
021: 00000010;
022: 00100010;
023: 01000000;

END;
```

instruction.mif

```
# hexadecimal
# hexadecimal
# Warning: read_during_write_mode_mixed_ports is assumed as              OLD_DATA
# Time: 0  Instance: tb_top.riscV.dp.instr_mem.meminst.memBlock0.altsyncram_component.m_default.altsyncram_inst
# Warning: read_during_write_mode_mixed_ports is assumed as              OLD_DATA
# Time: 0  Instance: tb_top.riscV.dp.instr_mem.meminst.memBlock1.altsyncram_component.m_default.altsyncram_inst
# Warning: read_during_write_mode_mixed_ports is assumed as              OLD_DATA
# Time: 0  Instance: tb_top.riscV.dp.instr_mem.meminst.memBlock2.altsyncram_component.m_default.altsyncram_inst
# Warning: read_during_write_mode_mixed_ports is assumed as              OLD_DATA
# Time: 0  Instance: tb_top.riscV.dp.instr_mem.meminst.memBlock3.altsyncram_component.m_default.altsyncram_inst
# Warning: read_during_write_mode_mixed_ports is assumed as              OLD_DATA
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock0.altsyncram_component.m_default.altsyncram_inst
# Warning: read_during_write_mode_mixed_ports is assumed as              OLD_DATA
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock1.altsyncram_component.m_default.altsyncram_inst
# Warning: read_during_write_mode_mixed_ports is assumed as              OLD_DATA
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock2.altsyncram_component.m_default.altsyncram_inst
# Warning: read_during_write_mode_mixed_ports is assumed as              OLD_DATA
# Time: 0  Instance: tb_top.riscV.dp.data_mem.mem32.memBlock3.altsyncram_component.m_default.altsyncram_inst
#                   35: Memory [  0] read with value: [xxxxxxxx] | [         x]
#
#                   35: Memory [  0] read with value: [00000001] | [         1]
#
#                   45: Register [ 1] written with value: [00000001] | [         1]
#
#                   65: Register [ 2] written with value: [00000002] | [         2]
#
# ** Note: $stop    : verif/tb_top.sv(43)
#    Time: 510 ns  Iteration: 0  Instance: /tb_top
# Break in Module tb_top at verif/tb_top.sv line 43
```

Resultado obtido

5. Conclusão

Apesar de diversas dificuldades no caminhar do projeto, chegamos a conclusão que o mesmo foi muito proveitoso do ponto de vista de aprendizado. Com este projeto conseguimos concluir que, na prática, a implementação de um processador não é nada trivial. Os desafios que essa construção requer necessitam de mentes criativas e bastante tentativa e erro, pois a cada etapa o projeto se tornava mais desafiador. Esta implementação de pipeline é apenas uma versão simplificada do que ela pode se tornar na realidade, saber disso só nos faz pensar do qual complexa e intrigante essa área é.