Arquitetura e Organização de Computadores 1

LISTA DE EXERCÍCIOS 2

junho/2025

Prof. Marcio Merino Fernandes

DC-UFSCAR – BCC/EnC

RISC-V: Microarquitetura **Monociclo**

1- Considere o **datapath monociclo** da figura abaixo e detalhamento da unidade de controle correspondente. Suponha que um dos seguintes sinais de controle na versão monociclo do processador RISC-V tem uma falha, estando **travado sempre com o sinal = 0.** Quais instruções funcionariam mal? Por quê ?

a) RegWrite

(b) ALUOp1

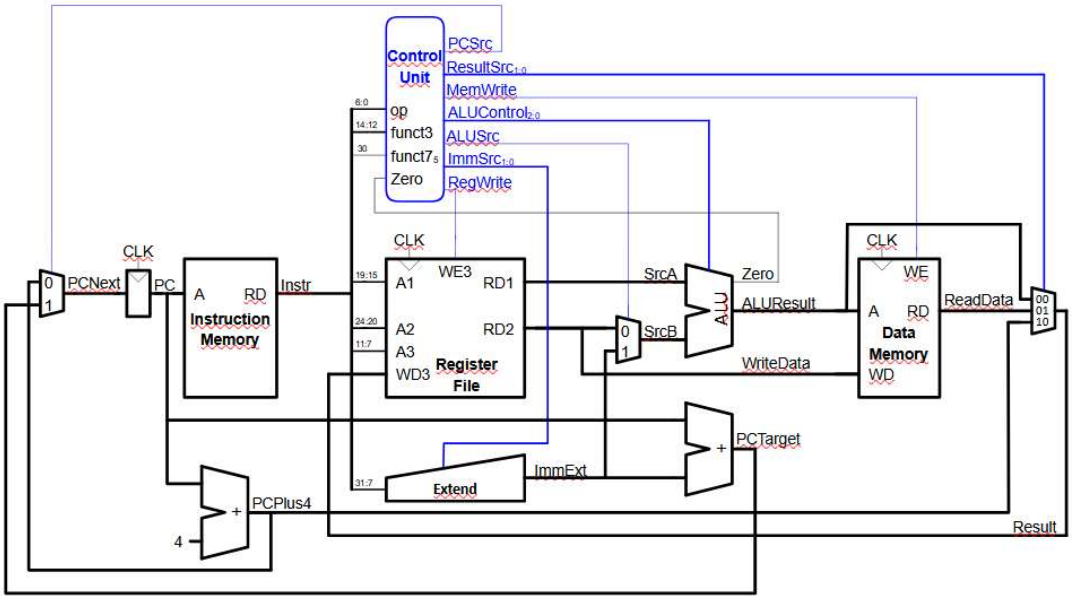
(d) MemWrite

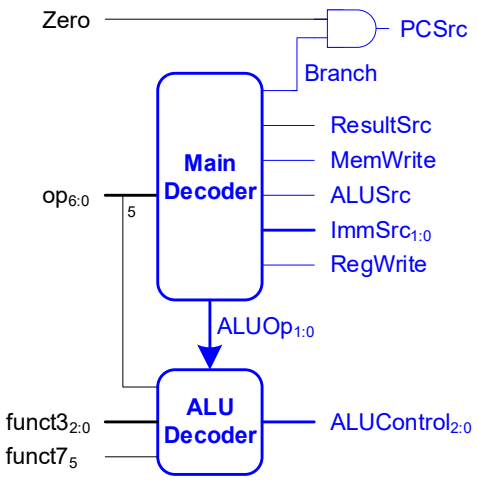
(f) ImmSrc0

(g) ResultSrc1

(i) PCSrc

(j) ALUSrc


Control Unit
PCSrc
ResultSrc1:0
MemWrite
ALUControl2:0
ALUSrc
ImmSrc1:0
RegWrite
op
funct3
funct7 5
Zero
CLK
PCNext
PC
Instruction Memory
Instr
Register File
A1
A2
A3
WD3
WE3
RD1
RD2
SrcA
SrcB
ALU
Zero
ALUResult
WriteData
Data Memory
WE
WD
ReadData
PCTarget
Extend
ImmExt
PCPlus4
Result


Zero
PCSrc
Branch
Main Decoder
ResultSrc
MemWrite
ALUSrc
ImmSrc1:0
RegWrite
op6:0
5
ALUOp1:0
ALU Decoder
funct32:0
funct75
ALUControl2:0

2- Modifique o processador RISC-V de ciclo único abaixo para implementar uma das seguintes instruções. Anote as modificações necessárias no diagrama da fogura, incluindo novos sinais de controle, caso necessário. Faça também as alterações de controle necessárias na tabelas abaixo, de modo a incluir as mudanças necessárias no decodificador principal e no decodificador da ALU.

(a) xor

(b) sll

(d) bne

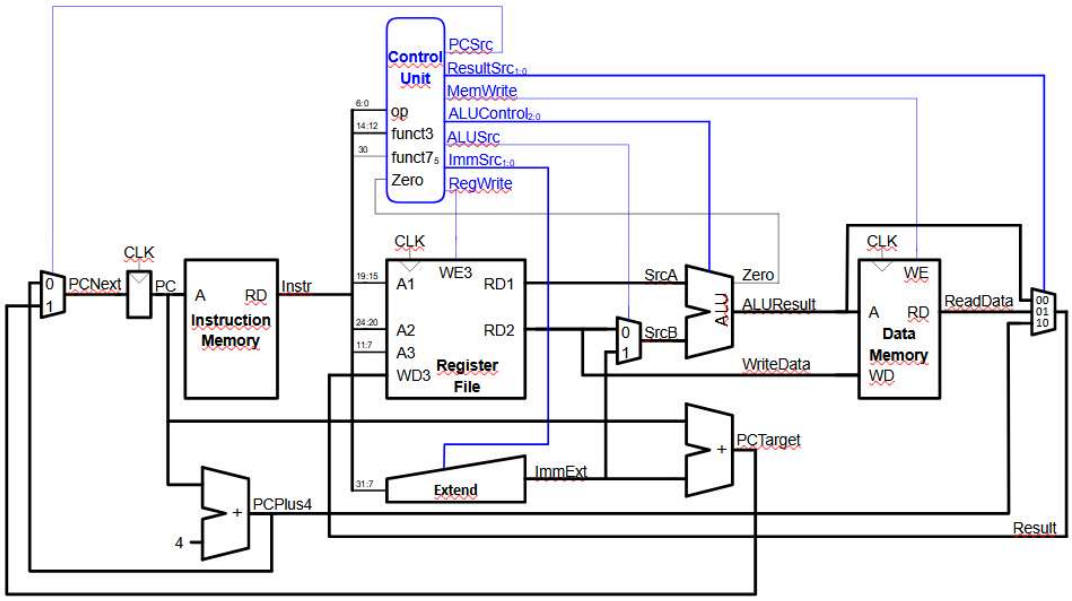


Table 7.6 Main Decoder truth table enhanced to support `jal`

Instruction	Opcode	RegWrite	ImmSrc	ALUSrc	MemWrite	ResultSrc	Branch	ALUOp	Jump
`lw`	0000011	1	00	1	0	01	0	00	0
`sw`	0100011	0	01	1	1	xx	0	00	0
R-type	0110011	1	xx	0	0	00	0	10	0
`beq`	1100011	0	10	0	0	xx	1	01	0
I-type ALU	0010011	1	00	1	0	00	0	10	0
`jal`	1101111	1	11	x	0	10	0	xx	1

Table 7.3 ALU Decoder truth table

ALUOp	funct3	$\{op_5, funct7_5\}$	ALUControl	Instruction
00	x	x	000 (add)	`lw, sw`
01	x	x	001 (subtract)	`beq`
10	000	00, 01, 10	000 (add)	`add`
	000	11	001 (subtract)	`sub`
	010	x	101 (set less than)	`slt`
	110	x	011 (or)	`or`
	111	x	010 (and)	`and`

RISC-V: Microarquitetura **Multiciclo**

3- Considere o **datapath multiciclo** da Figura abaixo e detalhamento da unidade de controle correspondente. Suponha que um dos seguintes sinais de controle na versão multiciclo do processador RISC-V tem uma falha**, estando travado sempre com o sinal = 0.** Quais instruções funcionariam mal? Por que?

(b) ResultSrc0

(c) ALUSrcB1

(f) ALUSrcA0

(h) ImmSrc0

(i) RegWrite

(j) PCUpdate

(k) Branch

(l) AdrSrc

(m) MemWrite

(n) IRWrite

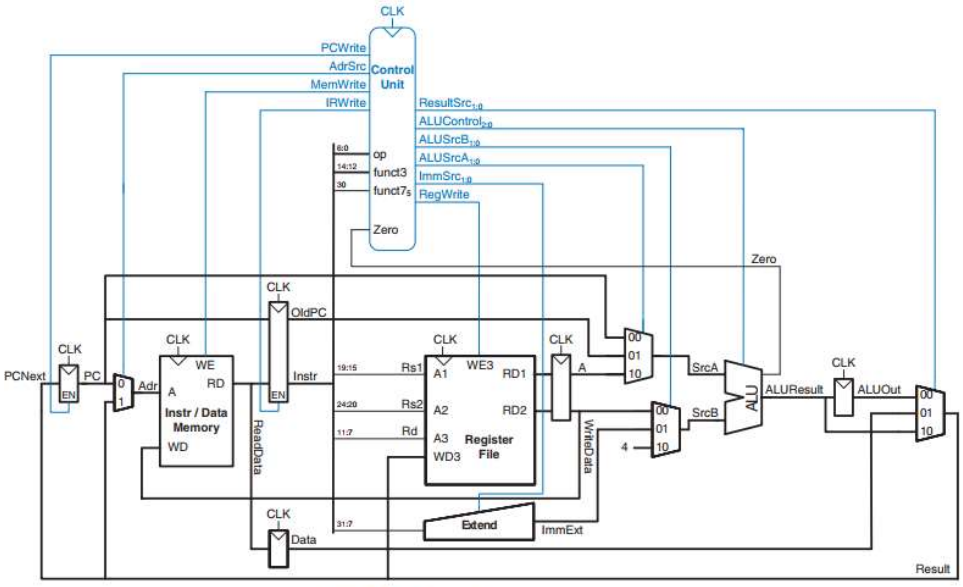
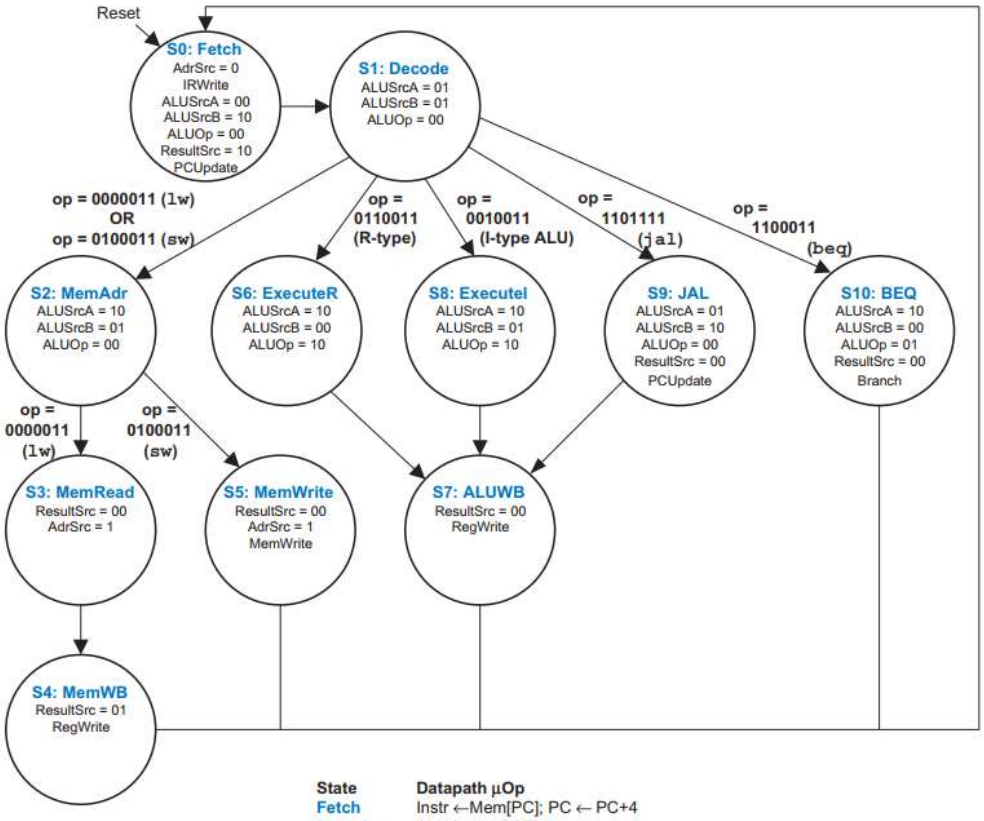


Figure 7.27 Complete multicycle processor



State	Datapath μOp
Fetch	Instr ← Mem[PC]; PC ← PC+4
Decode	ALUOut ← PCTarget
MemAdr	ALUOut ← rs1 + imm
MemRead	Data ← Mem[ALUOut]
MemWB	rd ← Data
MemWrite	Mem[ALUOut] ← rd
ExecuteR	ALUOut ← rs1oprs2
ExecuteI	ALUOut ← rs1opimm
ALUWB	rd ← ALUOut
BEQ	ALUResult = rs1-rs2; if Zero, PC = ALUOut
JAL	PC = ALUOut; ALUOut = PC+4

Figure 7.45 Complete multicycle control FSM

4- Considere as figuras do exercício anterior. Modifique o processador RISC-V monociclo para implementar uma das seguintes instruções. Anote as modificações necessárias no diagrama da nas figuras, incluindo novos sinais de controle, caso necessário. Faça também as alterações de controle necessárias na FSM principal, de modo a incluir as mudanças necessárias no decodificador principal e no decodificador da ALU.

(a) xor

(b) sll

(d) bne

5- Quantos ciclos são necessários para executar o seguinte programa em o processador multiciclo RISC-V? Qual é o valor de CPI (Ciclos por Instrução) deste programa ?. Qual é a influência deste parâmetro no desempenho do processador ?

```
addi s0, zero, 5 # result = 5

L1:
     bge zero, s0, done # se result <= 0, exit loop
     addi s0, s0, -1 # result = result - 1
j L1

done:
```

RISC-V: Microarquitetura c/ **Pipeline**

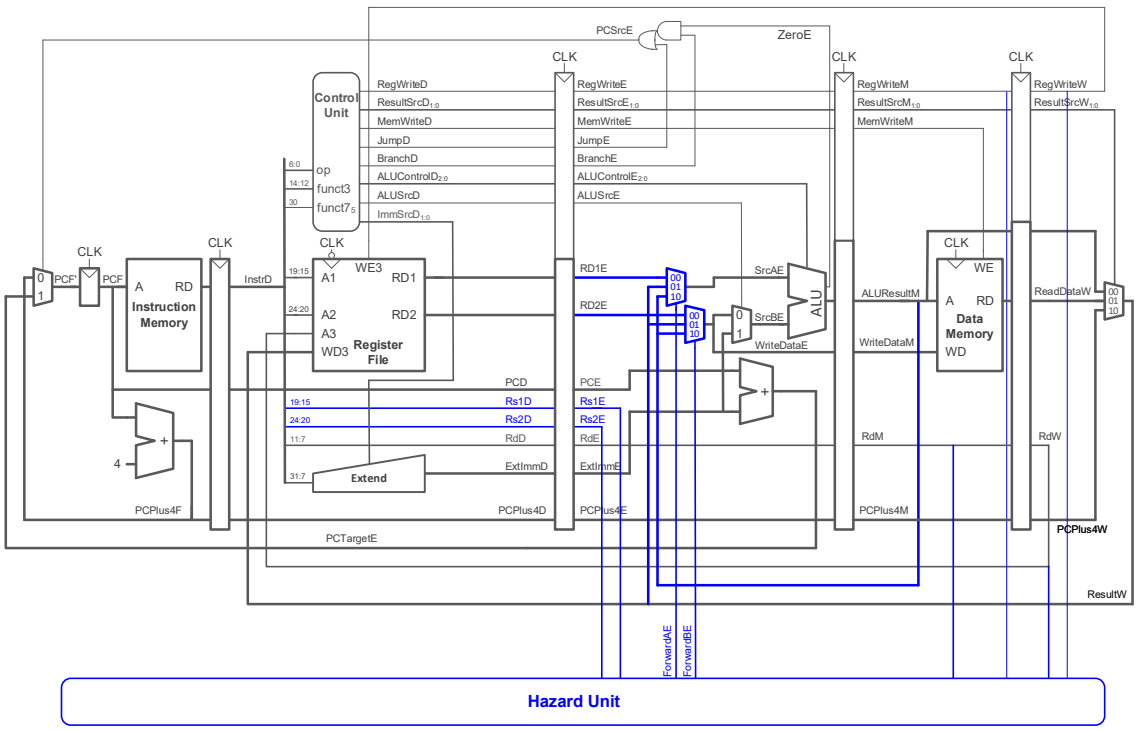
6- Explique as vantagens dos microprocessadores com pipeline.

7- Se estágios de pipeline adicionais permitem que um processador vá mais rápido, por que os processadores não têm 100 estágios de pipeline?

8- Descreva o que é hazard em um microprocessador com pipeline, e explique como pode ser resolvido. Quais são os prós e contras de cada forma ?

9- Considere o processador RISC-V com pipeline da figura abaixo, que está executando o seguinte trecho de código. trecho. Quais registradores estão sendo escritos e quais estão sendo lidos no quinto ciclo? Lembre-se de que o processador RISC-V com pipeline possui uma hazard unit. Você pode assumir um sistema de memória que retorna o resultado lido em um e um ciclo.

```
xor s1, s2, s3 # s1 = s2 ^ s3
addi s0, s3, -4 # s0 = s3 - 4
lw s3, 16(s7) # s3 = memory[s7+16]
sw s4, 20(s1) # memory[s1+20] = s4
or t2, s0, s1 # t2 = s0 | s1
```

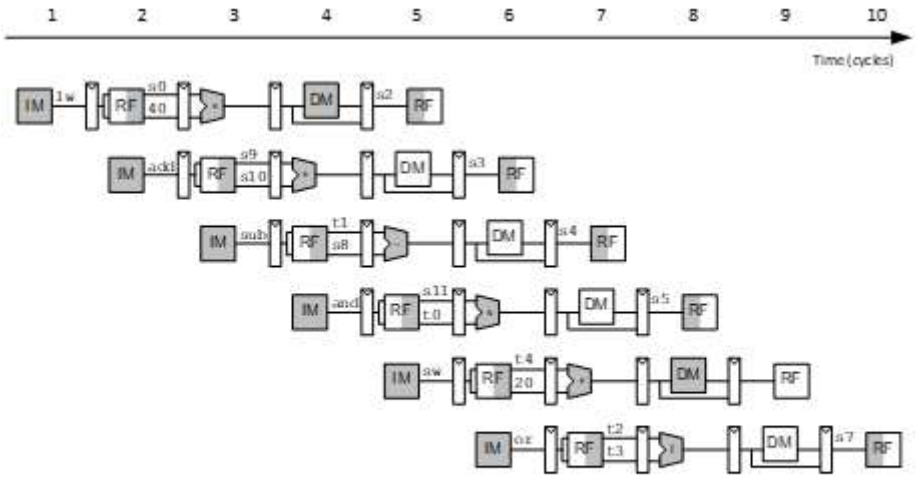


10- Repita o exercício anterior para o seguinte trecho de código:

```
addi s1, zero, 52 # s1 = 52
addi s0, s1, -4 # s0 = s1 - 4 = 48
lw s3, 16(s0) # s3 = memory[64]
sw s3, 20(s0) # memory[68] = s3
xor s2, s0, s3 # s2 = s0 ^ s3
or s2, s2, s3 # s2 = s2 | s3
```

11- Usando um diagrama semelhante à figura abaixo (*desconsidere os nros de registradores indicados*), mostre o encaminhamento (forwarding) e paradas (stalls) necessários para executar as instruções abaixo no pipeline do processador RISC-V.

```
addi s1, zero, 11 # s1 = 11
lw s2, 25(s1) # s2 = memory[36]
lw s5, 16(s2) # s5 = memory[s2+16]
add s3, s2, s5 # s3 = s2 + s5
or s4, s3, t4 # s4 = s3 | t4
and s2, s3, s4 # s2 = s3 & s4
```



12- Quantos ciclos são necessários para o processador RISC-V com pipeline emitir (buscar) todas as instruções para o programa abaixo ? Qual é o valor de CPI (Ciclos por Instrução) deste programa?

```
addi s1, zero, 52 # s1 = 52
addi s0, s1, -4 # s0 = s1 - 4 = 48
lw s3, 16(s0) # s3 = memory[64]
sw s3, 20(s0) # memory[68] = s3
xor s2, s0, s3 # s2 = s0 ^ s3
or s2, s2, s3 # s2 = s2 | s3
```