

Introduction to Computers II $MODULE \ 6$

Basic problems:

- 1. Provide the value of the control signals produced in a multicycle RISC-V when executing a **sw** instruction.
- 2. Provide the value of the control signals produced in a multicycle RISC-V when executing an add instruction.
- **3.** Provide the value of the control signals produced in a multicycle RISC-V when executing a jal instruction.
- 4. A given program, which is running on a multicycle processor, needs an average of 4.25 cycles per instruction. The program is formed by 10,000 instructions, with the following distribution: 30% are 1w, 15% are sw, 20% are arithmetic-logic and the rest are jal and beq. Determine how many jal instructions and how many beq instructions are in such program.
- **5.** A certain multicycle processor is running a program with 140 instructions. 70 of these instructions take 4 cycles to execute, 35 take 5 cycles, 20 take 3 cycles and the other 15 take 7 cycles. Calculate the CPI of this program. If the processor works with a 2 GHz frequency, calculate the execution time of the program.
- 6. Consider two multicycle processors with the following features:
 - PowerPC, which works with a 1.8 GHz frequency and 700 MIPS.
 - Pentium 4, which works with a 1.6 GHz frequency and 850 MIPS.

Calculate the CPI of each processor.

- 7. Consider the two processors of the previous exercise. When running a certain program, the processors obtain a CPI of 5.5 (PowerPC) and 7 (Pentium 4). The machine code generated by the compiler has 9 million instructions (PowerPC) and 7.2 million instructions (Pentium). Which computer will run the program faster?
- **8.** A given program runs on two multicycle processors (A and B), which work with 1 GHz and 1.5 GHz frequencies, respectively. The distribution of the program instructions running on A is as follows:

	arithmetic	load	store	branch
frequency	50%	25%	10%	15%
cycles	4	5	4	3

- a) Calculate the CPI of the program running on processor A.
- b) The number of instructions running on B is 60% of the executed in A, and its execution time is half the time in A. Calculate the CPI of the program running on processor B.

Additional problems:

9. Assume that the following program is running on a multicycle RISC-V, with the initial value of the memory and registers shown in the picture. Represent the execution diagrams for the registers and the memory, as well as for the control and status signals, so that their values are shown for each clock cycle.



10. Assume that the following program is running on a multicycle RISC-V, with the initial value of the memory and registers shown in the picture. Represent the execution diagrams for the registers and the memory, as well as for the control and status signals, so that their values are shown for each clock cycle.



- **11.** Indicate what instruction is being executed and the state of the multicycle processor controller, if the values of the control signals are the following:
 - a) MemWr = 1
 - b) ALUSrcB = 00 and ALUctr = 010
 - c) MDRwr = 1
 - d) ALUsrcA = 01 and ALUsrcB = 10
- e) ALUSrcB = 01 and ALUctr = 101
- f) ResSrc = 00 and PCupdate = 1
- g) ResSrc = 01
- h) Branch = 1

12. When silicon chips are fabricated, defects in materials and manufacturing errors can result in defective circuits. A very common defect is for one signal wire to get damaged and always register a logical 0 (stuck-at-0) or a logical 1 (stuck-at-1). Indicate what multicycle RISC-V instructions would fail if the following control signals suffered a stuck-at-0 defect:

a) ResSrc ₀	e) ALUsrcA ₀	i) Branch
b) ALUsrcB ₁	f) ImmSrc ₁	j) AddrSrc
c) ALUsrcB ₀	g) ImmSrc ₀	k) MemWr
d) ALUsrcA ₁	h) PCupdate	1) IRwr

- 13. Repeat the previous exercise, but now with a stuck-at-1 defect.
- **14.** Discuss the modifications required in the data path and the controller ASM diagram of the multicycle RISC-V, in order to extend its ISA with each of the following instructions:

a)	xor/xori	d)	slt/slti	g)	jarl
b)	sll/ssli	e)	lb/lh/lbu/lhu	h)	lui
c)	bne/blt/bge	f)	$\mathbf{sb} / \mathbf{sh}$	i)	auipc

15. Discuss the modifications required in the data path and the controller ASM diagram of the multicycle RISC-V, in order to extend its ISA with a new I-type instruction that reads a word from memory, pre-incrementing the base register, lwpreinc rd, imm(rs1):

{ rd \leftarrow Mem[rs1 + sExt(imm)], rs1 \leftarrow rs1 + sExt(imm) }

16. Discuss the modifications required in the data path and the controller ASM diagram of the multicycle RISC-V, in order to extend its ISA with a new I-type instruction that reads a word from memory, post-incrementing the base register, lwpostinc rd, imm(rs1):

{ rd \leftarrow Mem[rs1], rs1 \leftarrow rs1 + sExt(imm) }

17. Discuss the modifications required in the data path and the controller ASM diagram of the multicycle RISC-V, in order to extend its ISA with a new I-type instruction that stores the result of an immediate addition in memory, swaddi rs1, rs2, imm:

{ Mem[rs1] \leftarrow rs2 + sExt(imm) }

18. Discuss the modifications required in the data path and the controller ASM diagram of the multicycle RISC-V, in order to extend its ISA with a new I-type instruction that adds a register with a data read from memory, addm rd, rs1, imm:

$$\{ rd \leftarrow rd + Mem[rs1 + sExt(imm)] \}$$

- **19.** Discuss the modifications required in the data path and the controller ASM diagram of the multicycle RISC-V, in the case the register file had a single read port, instead of two. How many cycles would be required to execute each instruction? Would the processor cycle time be reduced? And in the case the register file had a single read/write port?
- **20.** Suppose that the register file of the multicycle processor could be redesigned so that it requires half the area, but at the expense of doubling its access delays. Would this change make sense?
- **21.** If you could reduce the delay of just one of the components of the multicycle RISC-V in the 90nm CMOS library by half, which one would you choose to get a smaller cycle

time? What would be the value of the new cycle time?

- **22.** If the ALU delay is reduced a 30%, the extension sign module delay a 10% and the register file read/write delays a 20% (referred to the multicycle RISC-V with the 90nm CMOS library), calculate:
 - a) The delay of the critical path for each state.
 - b) The speed-up obtained when running a program with 100 million instructions.
- **23.** Assuming that the instructions of a certain program running on a multicycle RISC-V follow this distribution:

	arithmetic	load/store	branch
frequency	60%	25%	15%
cycles	4	6	3

- a) Calculate the CPI.
- b) If we want to improve its performance a 25%, how many cycles (on average) would an optimized arithmetic instruction take to execute, if the rest of the instructions are not modified?
- c) Same if we want to improve its performance a 50%.
- **24.** Suppose that a new arithmetic instruction is added to the multicycle RISC-V ISA, which reduces the number of executed arithmetic instructions in 25%, but increases the cycle time a 10%. Does this change make sense?

	arithmetic	load/store	branch
frequency	60%	25%	15%
cycles	4	6	3

Exam problems:

25. (Adapted September 2016) The program fragment shown below is running on a multicycle RISC-V, which has a 1 GHz frequency, producing an execution time of 3,474 ns.

	mv	s4,	zero
	la	s5,	A
	mv	s6,	zero
	li	s4,	127
L1:	slli	s3,	s4, 2
	add	s3,	s3, s5
	lw	s0,	0(s3)
	add	s6,	s6, s0
	add	s4,	s4, -1
	bge	s4,	zero, Ll
L2:			
•	(res	st of	the program)

- a) Calculate the CPI of this program fragment.
- b) Calculate the value of the MIPS metric.

Solutions

1. to 10. See slides

11.	a) sw, S5 b) and, S6 c) sw, S5 d) jal, S9	e) slti, s8 f) jal, s9 g) lw, S4 h) beq, S10	
12.	a) 1w	e) jal, beq	i) beq
	b) all	f) sw, jal	j) lw, sw
	c) all except R-type	g) jal, beq	k) sw
	d) all	h) all	l) all
13.	a) all	e) all	i) all
	b) all	f) 1w, sw, I-type	j) all
	c) all	g) 1w, beq, I-type	k) all except sw
	d) all	h) all	l) all

14. a) Extend the functionality of the ALU.

- b) Extend the functionality of the ALU.
- c) Extend the functionality of the ALU.
- d) Extend the functionality of the ALU.
- e) Add width selection lines to the Data Memory.
- Add an extender to the output of the Data Memory.
- f) Add width selection lines to the Data Memory.
- g) Add a new state that performs ALUout \leftarrow A + sExt(imm).
- Add a transition from this state to S9.
- Add a transition from S1 to this state if op = jarl.
- h) Extend the functionality of the Sign Extender.
- Connect the output of the Sign Extender to a new input of the WB-MUX.
- Add a new state that performs $RF[rd] \leftarrow sExt(imm)$.
- Add a transition from this state to S0.
- Add a transition from S1 to this state if op = lui
- i) Extend the functionality of the Sign Extender.
- Add a transition from S1 to S7 if op = auipc
- 15. Connect a 2-input MUX to the WA input of the RF, with inputs connected to bits 11:7 and 19:11 of the IR.
 Expand the transition from S1 to S2 to also occur if op = lwpreinc
 Add a new state that performs RF[rs1] ← ALUout.
 Add a transition from this state to S0.
 Add a transition from S4 to this state if op = lwpreinc
- 16. Add a new input to the lower ALU MUX with a value of 0. Connect a 2-input MUX to the WA input of the RF, with inputs connected to bits 11:7 and 17:11 of the IR. Add a new state that performs ALUout ← A + 0. Add a transition from this state to S3. Add a transition from S1 to this state if op = lwpostinc

Expand S4 to also perform ALUout \leftarrow A + sExt(imm). Add another new state that performs RF[rs1] \leftarrow ALUout. Add a transition from this state to S0. Add a transition from S4 to this state if op = lwpostinc Add a transition from S4 to S0 if op = lw

- 17. Connect register A to a new input of the MEM-MUX. Connect register B to a new input of the upper ALU-MUX. Add a new state that performs ALUout ← B + sExt(imm). Add another new state that performs Mem[rs1] ← ALUout. Add a transition from the first to the second state. Add a transition from S1 to the first state if op = swaddi Add a transition from the second state to S0.
- **18.** Connect a 2-input MUX to the RA1 input of the RF, with inputs connected to bits 19:15 and 11:7 of the IR.

Connect the MDR register to a new input of the lower ALU-MUX. Expand the transition from S1 to S2 to also occur if op = addmAdd a new state that performs $A \leftarrow RF[rd]$. Add another new state that performs ALUout $\leftarrow A + MDR$. Add a transition from the first to the second state. Add a transition from S3 to the first state if op = addmAdd a transition from the second state to S7.

19. Connect a 2-input MUX to the single read address input of the RF, with inputs connected to bits 19:15 and 24:20 of the IR.

Connect the single read data output of the RF to registers A and B.

Split state S1 into two states: the first performs $A \leftarrow A + RF[rs1]$ and the second performs the remaining actions.

Each instruction would take one more cycle to execute, and the cycle time would remain the same.

If there is only one input/output port, the MUX should have 3 inputs connected to bits 19:15, 24:20, and 11:7 of the IR. The WB-MUX should also be connected to the single data port of the RF.

21.	ALU, 9312 ps.			
22.	a) S0: 9312 ps S4: 1321 ps S8: 6708 ps b) 1,05	S1: 7134 ps S5: 9140 ps S9: 6708 ps	S2: 6662 ps S6: 6708 ps S10: 7282 ps	S3: 9140 ps S7: 1321 ps t _{CLK} : 9312 ps
23.	a) 4,35	b) 2,55	c) 1,58	
24.	Yes			
25.	See slides.			

20.

Yes