COEN 316 – Computer Architecture and Design

Department of Electrical and Computer Engineering

Assignment 3, Winter 2023

Due: April. 10, 2023
In this assignment, you will answer the following questions. Write your answer in the exact place:

Your Information:	
First Name:	**Samson**
Last Name:	**Kaller**
Student ID:	**40136815**
Grade:	

Question 1	**Grade**
1. Assume you examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:	**20**

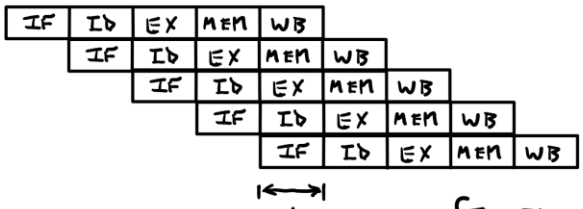
IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

Also, assume that instructions executed by the processor are broken down as follows:

ALU	BEQ	LW	SW
45%	20%	20%	15%

a- What is the clock cycle time in a pipelined and non-pipelined processor?

a) PIPELINED:



→ COMPLETES 1 INSTRUCTION PER CLOCK CYCLE, WITH 5 INSTRUCTION STAGES (PARALLEL).

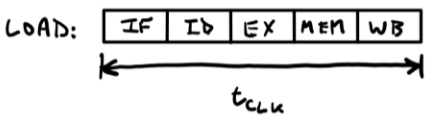
→ INSTRUCTION STAGES MUST BE SAME LENGTH (USE THE MAX).

$t_{CLK} = MAX\{IF, ID, EX, MEM, WB\}$

$= MAX\{250, 350, 150, 300, 200\} \Rightarrow t_{CLK} = 350\,ps$

PIPELINED

NON-PIPELINED:



→ NOT ALL INSTRUCTIONS USE ALL STAGES.

→ CLOCK CYCLE TIME MUST COVER WORST CASE: LOAD INSTRUCTION (ALL 5 STAGES)

→ CLOCK CYCLE TIME IS SUM OF STAGES FOR LOAD

$t_{CLK} = t_{IF} + t_{ID} + t_{EX} + t_{MEM} + t_{WB}$

$= 250 + 350 + 150 + 300 + 200 \Rightarrow t_{CLK} = 1250\,ps = 1.25\,ns$

NON-PIPELINED

b- What is the total latency of an LW instruction in a pipelined and non-pipelined processor?

b) PIPELINED:

→ ALL STAGES ARE SAME LENGTH: MAX LENGTH $t_{ID} = 350\,ps$

→ ALL INSTRUCTIONS GO THROUGH ALL 5 STAGES

LW LATENCY $t = 5 \cdot 350\,ps = 1750\,ps = 1.75\,ns$

PIPELINED LW LATENCY

NON-PIPELINED:

→ LW INSTRUCTION USES ALL 5 STAGES

LW LATENCY $t = 250 + 350 + 150 + 300 + 200 = 1250\,ps = 1.25\,ns$

NON-PIPELINED LW LATENCY

Question 2

Grade

50

The instruction "addi" can be executed using the single-cycle data path without modification. Answer the following questions:

a- What are the values of all control signals needed to execute this instruction on the single-cycle data path? (Note FIGURE 4.22 in the textbook)

Input/output	Signal name	addi	
Inputs	Op5	0	addi: Opcode = $8_{HEX} = 001000_2$
	Op4	0	
	Op3	1	
	Op2	0	
	Op1	0	
	Op0	0	
Output	RegDst	0	→ DEST IS Rt
	ALUSrc	1	→ SRC IS SIGN-EXT IMM.
	MemtoReg	0	→ ALU OUT IS INPUT TO REGFILE
	RegWrite	1	→ WRITE RESULT TO REGFILE
	MemRead	X	→ DON'T CARE ABOUT MEM. DATA
	MemWrite	0	→ DO NOT WRITE TO MEM.
	Branch	0	→ NO BRANCHING
	ALUOp1	0	→ ALUOp = 00 IS ADD
	ALUOp0	0	

The outputs were determined using the data on the following page…

MIPS Reference Data ①

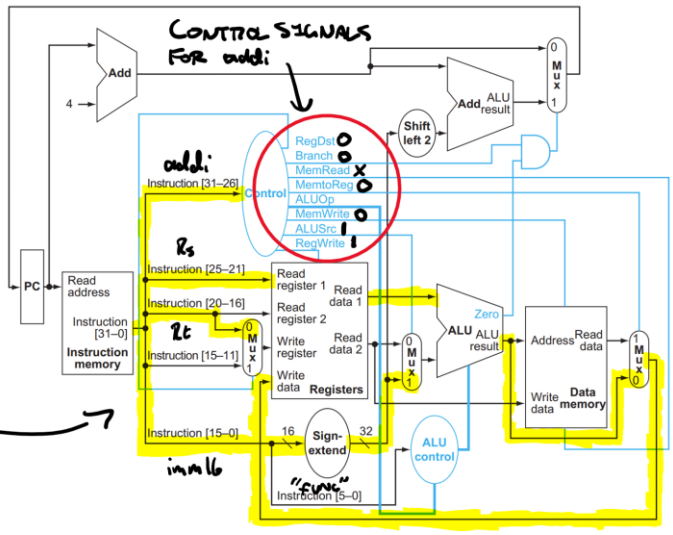
CORE INSTRUCTION SET

NAME, MNEMONIC		FOR-MAT	OPERATION (in Verilog)		OPCODE / FUNCT (Hex)
Add	add	R	R[rd] = R[rs] + R[rt]	(1)	$0 / 20_{hex}$
Add Immediate	addi	I	R[rt] = R[rs] + SignExtImm	(1,2)	8_{hex}
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + SignExtImm	(2)	9_{hex}

addi OpCode : 001000

addi Rt, Rs, imm16

NEEDS: Rs, Rt, imm16

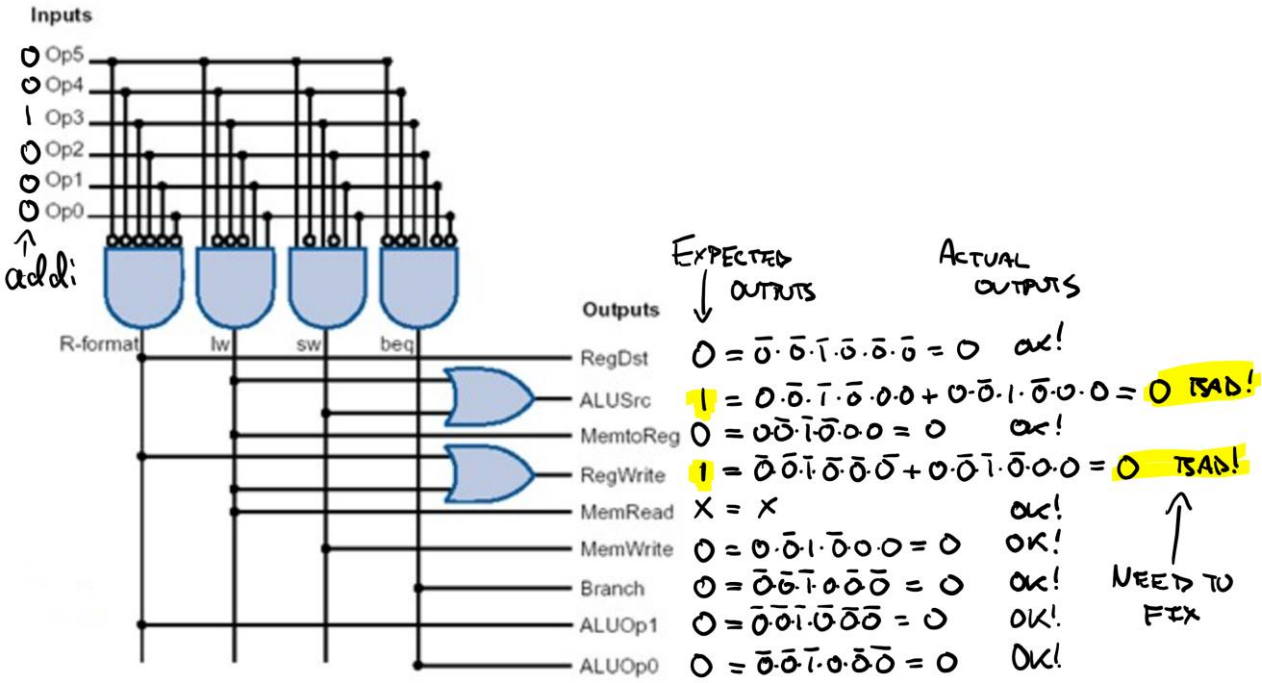


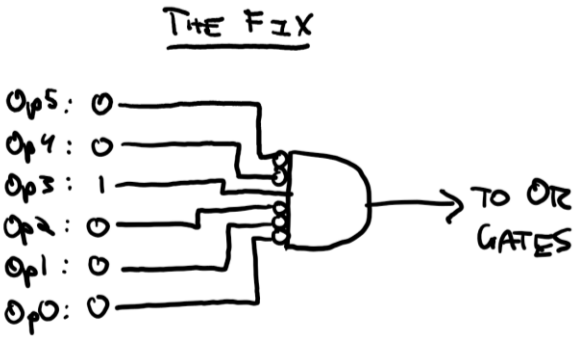
Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	set on less than	101010	set on less than	0111

LIKE LW/SW, WE WANT ALUOp TO BE 00.

USING ALUOp = 10 WILL CAUSE THE LOWER 6 BITS OF imm16 TO BE INTERPRET AS "FUNC" INPUT TO ALU CONTROL. WE DO NOT WANT THIS.

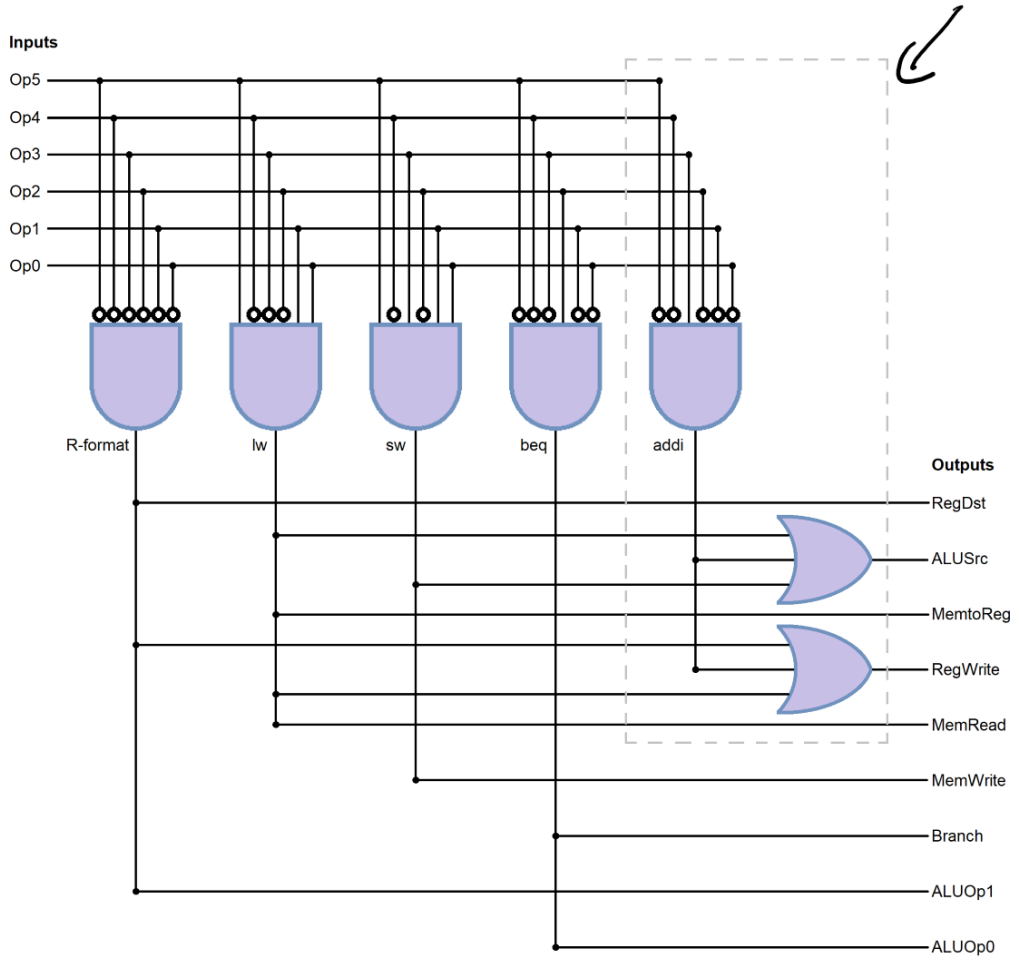
b- Modify the following main control unit to allow for the implementation of this instruction.





Modified circuit on next page…

NEW CONTROL UNIT INCLUDING addi (MODIFICATIONS IN RECTANGLE)



* NOTE THAT THIS COVERS ONLY THE ADDITION OF THE addi INSTRUCTION. FOR ADDING SUPPORT FOR MORE I-TYPE INSTRUCTIONS, ADDITIONAL MODIFICATIONS MUST BE MADE.

Question 3	Grade
Divide (1010)2 by (0011)2 using an optimized divider algorithm assuming all values are unsigned numbers. Use the table below to show the steps performed and give the values of the quotient and the remainder in the space provided in binary. You may add additional rows to the table as needed.	30

Step	comment	divisor	A
	initialization	00011	00000 1010
1			

STEP	COMMENT	DIVISOR	A
0	INITIALIZATION	0 00 11	0 00 00 10 10
1.1	SHIFT	0 00 11	0 00 01 01 00
1.2	SUB	0 00 11	1 11 10 01 00
1.3	RESTORE, $A_0 = 0$	0 00 11	0 00 01 01 00
2.1	SHIFT	0 00 11	0 00 10 10 00
2.2	SUB	0 00 11	1 11 11 10 00
2.3	RESTORE, $A_0 = 0$	0 00 11	0 00 10 10 00
3.1	SHIFT	0 00 11	0 01 01 00 00
3.2	SUB	0 00 11	0 00 10 00 00
3.3	$A_0 = 1$	0 00 11	0 00 10 00 01
4.1	SHIFT	0 00 11	0 01 00 00 10
4.2	SUB	0 00 11	0 00 01 00 10
4.3	$A_0 = 1$	0 00 11	0 00 01 00 11

REMAINDER QUOTIENT

QUOTIENT: 0011

REMAINDER: 0001

$10 \div 3 = 3\ 1/3$ ✓

Grading Policy:

The assignment score is out of 100 points.

Here are some aspects that may lead to points deduction:

- The answers are missing.
- Missing steps.
- Inappropriate data to answer your question.
- Do your best to include exhaustive details, the final answer alone is not enough to get points.
- Collaborate on the individual assignment.