M1233-CSE-3666-011.021 Exam 2 Priyanshu Agrawal

TOTAL POINTS

99.5 / 100

QUESTION 1

1 Q1(a) 7 / 7

✓ **- 0 pts** *Correct*

- 1 pts 1st incorrect, correct: 0

- 1 pts 2rd incorrect, correct: 1

- 1 pts 3rd incorrect, correct: 0

- 1 pts 4rd incorrect, correct: 0

- 1 pts 5rd incorrect, correct: 0

- 1 pts 6rd incorrect, correct: 0

- 1 pts 7rd incorrect, correct: 1

- 1 pts 8rd incorrect, correct: 0

QUESTION 2

2 Q1(b) 4 / 4

✓ **- 0 pts** *Correct*

- 2 pts Incorrect Row(0,0,0)

- 2 pts Incorrect Row(0,1,1)

- 3 pts Some Effort

- 4 pts No Effort

QUESTION 3

3 Q1(c) 4 / 4

✓ **- 0 pts** *Correct*

- 0.5 pts Correct Approach but Incorrect Expression

- 2 pts Partially Correct

- 3 pts Some Effort

- 4 pts No Effort

QUESTION 4

4 Q2(a) 6 / 6

✓ **- 0 pts** *Correct*

- 2 pts Incorrect time that register output changes

- 2 pts Incorrect multiplicand

- 2 pts Incorrect product

- 1 pts Output of registers changes after propagation delay

- 1 pts Multiplicand shifted incorrectly **- 0.5 pts** Using incorrect multiplicand in calculating product

- 1 pts Product calculated for incorrect time step **- 0.5 pts** Addition mistake when calculating product

- 6 pts No answer

QUESTION 5

5 Q2(b) 4 / 4

✓ **- 0 pts** *Correct*

- 2 pts First blank incorrect, correct: 10 **- 2 pts** Second blank incorrect, correct: 0101 1011

+ 1 pts Steps are correcct

QUESTION 6

6 Q3(a) 9.5 / 10

- 0 pts Correct
- 0.5 pts Partially incorrect for RegWrite **- 1 pts** Mostly incorrect for RegWrite **- 2 pts** Incorrect for RegWrite

- 0.5 pts Partially incorrect for PCSrc **- 1 pts** Mostly incorrect for PCSrc

- 2 pts Incorrect for PCSrc

- 0.5 pts Partially incorrect for MemtoReg **- 1 pts** Mostly incorrect for MemtoReg **- 2 pts** Incorrect for MemtoReg

- 0.5 pts Partially incorrect for ALUSrc **- 1 pts** Mostly incorrect for ALUSrc

- 2 pts Incorrect for ALUSrc

✓ **- 0.5 pts** *Partially incorrect for ImmGen* **- 1 pts** Mostly incorrect for ImmGen **- 2 pts** Incorrect for ImmGen

QUESTION 7

7 Q3(b) 4 / 4

✓ **- 0 pts** *Correct*

- 0.5 pts Very close, minor mistake or missing tiny detail

- 1 pts Close, mistake or missing some detail **- 2 pts** Right idea, mistakes or lacking detail **- 4 pts** Incorrect or no attempt

QUESTION 8

8 Q3(c)(1) 7 / 7

✓ **- 0 pts** *Correct*

- 1 pts LW/SW Incorrect; Should be 00 **- 1 pts** BEQ Incorrect; Should be 01

- 1 pts ADD incorrect; Should be 00

- 1 pts SUB incorrect; Should be 01

- 1 pts AND/ANDI incorrect; Should be 10 **- 1 pts** OR/ORI incorrect; Should be 11

- 1 pts ADDI incorrect; Should be 00

- 4 pts Meaningful attempt or partially incorrect. (If applies: ALU Operation should be one of the 2- bit numbers from given table)

- 7 pts No meaningful attempt

QUESTION 9

9 Q3(c)(2) 3 / 3

✓ **- 0 pts** *Correct*

- 2 pts Meaningful attempt; but incorrect **- 3 pts** No meaningful attempt

- 1.5 pts Correct conclusion, but premises/reasons incorrect

- 1 pts Incorrect conclusion, but

premises/reasons correct

QUESTION 10

10 Q3(d)(1) **3 / 3**

✓ **- 0 pts** *Correct*

- 0.75 pts The first blank incorrect, correct: ALU Result

- 0.75 pts The second blank incorrect, correct: Read Data

- 0.75 pts The third blank incorrect, correct: PC4 **- 0.75 pts** The forth blank incorrect, correct: Read Data

- 3 pts All wrong or never try

QUESTION 11

11 Q3(d)(2) **3 / 3**

✓ **- 0 pts** *Correct*

- 0.5 pts One input incorrect, others correct **- 1 pts** One correct input, others incorrect **- 3 pts** All inputs incorrect or no attempt

QUESTION 12

12 Q4(a) **10 / 10**

✓ **- 0 pts** *Correct*

- 2 pts ALUSrc = 0

- 2 pts Branch Target = (PC + 4 = 0x0440AA28) **- 1 pts** Branch Target added wrong pc **- 2 pts** Read Register 1 = 2

- 2 pts Write Register 1 = 21

- 2 pts Immd = 64

- 0.5 pts used x21 or x2

- 10 pts No Attempt

QUESTION 13

13 Q4(b)(1) **4 / 4**

✓ **- 0 pts** *Correct*

- 1 pts Shifted Up 1, Instruction Decode phase should shift over after the stalls

- 1 pts Added Extra Stall phases (Not needed) **- 2 pts** Multiple Incorrect Stalls

- 1 pts Stalls misplaced

- 1.5 pts Missing Stalls

- 3.5 pts No Stalls Shown.

QUESTION 14

14 Q4(b)(2) **4 / 4**

✓ **- 0 pts** *Correct*

- 2 pts Part A Fully Wrong (Number of Stall Cycles)

- 1 pts Part A Partially Wrong (within a couple (Answer is 2))

- 0.5 pts Part A misrepresented (Got one right but said 4 stall cycles (Answer is 2))

- 2 pts Part B Fully Wrong (Listed) (Both Wrong) **- 1 pts** Part B partially wrong (only 1 right or have multiple extras)

- 0.5 pts Part B misrepresented (1 extra answer or something)

QUESTION 15

15 Q4(c) **7 / 7**

✓ **- 0 pts** *Correct*

- 1 pts 4C1. = 4 or MEM stage of I1

- 2 pts 4C2. I2 Uses MEM/WB->MEM

- 1 pts 4C2. I2 half wrong

- 2 pts 4C2. I3 Uses MEM/WB->EX

- 1 pts 4C2. I3 half wrong

- 1 pts 4C3. = 0

- 1 pts 4C4. = 1

- 7 pts No attempt

QUESTION 16

16 Q5(a)(1) **4 / 4**

✓ **- 0 pts** *Correct*

- 1 pts Control & RF read parallel

- 1 pts wrong clock cycle time

- 1 pts LW Instruction decides the cycle time **- 2 pts** not showing adding step and wrong clock cycle time

- 0.5 pts there is no sw instruction

- 0.5 pts no RF Write

- 0.5 pts only control & RF Read work in parallel **- 0.5 pts** give something related but not showing that lw instruction decides the cycle time

QUESTION 17

17 Q5(a)(2) **2 / 2**

✓ **- 0 pts** *Correct*

- 1 pts 60ns
- 1 pts D-MEM

QUESTION 18

18 Q5(a)(3) **4 / 4**

✓ **- 0 pts** *Correct*

- 1 pts single cycle execution time wrong **- 1 pts** pipelined execution time wrong **- 2 pts** not showing the step to get execution time

- 1 pts right step but wrong answer **- 1 pts** wrong answer

QUESTION 19

19 Q5(b)(1) **5 / 5**

✓ **- 0 pts** *Correct*

- 2 pts Wrong total execution time

- **1 pts** Wrong execution time of MUL - **3 pts** Effort
- **1 pts** Wrong final answer
- **5 pts** No effort

QUESTION 20

20 Q5(b)(2) **5 / 5**

✓ **- 0 pts** *Correct*

- **2 pts** Wrong new execution time
- **1 pts** Wrong total execution time
- **1 pts** Wrong final answer
- **3 pts** Effort
- **5 pts** No effort

101

CSE 3666. Exam 2

April 11, 2023

This is a 90 minutes exam. The total number of points is 100.

- Read questions carefully.
- Budget your time wisely.
- Write your answers neatly and legibly on the exam paper.

Question	Points	Score
1	15	
2	10	
3	30	
4	25	
5	20	
Total:	100	

I pledge my honor that I have not violated and will not violate the exam policy of this course and the Student Conduct Code during this examination.

Printed Name: Priyanshu Agrawal

Signature: Priyanshu Agrawal Date: 4/11/2023

NetID: pra20003 Section: 021

1. (15 points) We design a module that takes three bits, `a` and `b`, and `s`, as input and generates a single-bit output `D`. The behavior of the module is as follows.

   `D` is 1 if and only if: (i) `a` and `b` are the same; and (ii) `s` is different from `a` and `b`.

   (a) (7 points) Complete the following truth table for `D`. No point is awarded for guessing, e.g., write all 0's or all 1's.

a	b	s	D
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

   (b) (4 points) Explain the values you placed in the following rows in the truth table. For example, if D is 0, describe which condition following "if and only if" is not true.

   Row where `a`, `b`, and `s` is 0, 0, 0 respectively.

   s is not different from a and b,
   Therefore D = 0

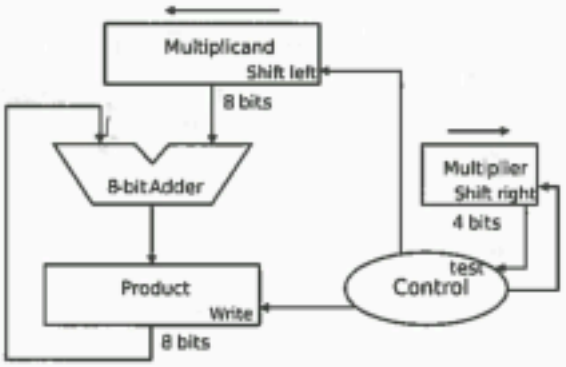
   Row where `a`, `b`, and `s` is 0, 1, 1 respectively.

   a and b are not the same,
   therefore D = 0

   (c) (4 points) Write the logic expression for ~~overflow~~ D from the truth table in part a). The expression is a sum of product like $\overline{X} \cdot Y + \overline{Z}$.

   D = $\overline{A} \cdot \overline{B} \cdot S + A \cdot B \cdot \overline{S}$

2. (10 points) We study the operation of a 4-bit multiplier shown in the following figure.



Fill in the blanks, assuming the following and excluding other delays.

- The setup time, the hold time, and the propagation delay of the registers is 1 ns, 0.5 ns, and 2 ns, respectively.
- The delay of the adder is 8 ns.
- The output of the control is stable 3 ns after the beginning of a cycle.
- The clock rate is 10 MHz.
- At time $t - 1$ ns, the bits in the Multiplicand register are 00011010, the bits in the Multiplier register is 0011, and the bits in the Product register is 00001101.
- A cycle starts at time t. **Only consider this cycle in the questions.**

(a) (6 points) At time $t+$ 2 ns, the output of the registers changes. After the changes, the bits in the Multiplicand register are 00110100, and the bits in the Product register are 00100111.

multiplicand 00011010
product 00001101
00100111

(b) (4 points) At time $t+$ 10 ns, the output of the adder changes to 0101 1011.

new product 00100111
new multiplicand 00110100
01011011

3. (30 points) Single-cycle processor.

(a) (10 points). The single-cycle RISC-V processor, as in the provided diagram, supports ADD, SUB, AND, OR, LW, SW, and BEQ instructions. Suppose RT denotes all of ADD, SUB, AND, and OR instructions. Select from **RT, LW, SW, and BEQ** to fill out the blanks. Include all instructions that make statements true.

RegWrite is 0 when the processor executes SW, BEQ instructions.

PCSrc is always 0 when the processor executes RT, LW, SW instructions.

MemtoReg is always 1 when the processor executes LW instructions.

ALUSrc is 1 when the processor executes LW, SW instructions.

An error in ImmGen affects the execution of LW, SW instructions.

(b) (4 points) Assume register x2 is `0x7FFFEF00` before the execution of the following instruction. A fault causes PCSrc to be 1 when the single-cycle processor executes the instruction. Describe how the fault affects PC in the next cycle and why.

```
0x00404800: 0x00812083          # lw   x1, 8(x2)
```

The fault will cause PC to be set to current PC + immediate. In this case, current PC is 0x00404800 and immediate is 8.
Therefore, next PC will be 0x00404808.
This because PCSrc is selector bit on mux3 that choses Next PC to be either PC+4 or PC+imm (normally for branch instructions).

(c) (10 points). Suppose we would like to support ADDI, ANDI, and ORI instructions, and also use a different ALU. The function of the new ALU is specified by ALU Operation, a 2-bit signal, as shown in the following table.

ALU Operation	Function
0 0	add
0 1	subtract
1 0	and
1 1	or

To build a new ALU control module, we construct the following table, which specifies how ALU Operation (to the ALU) should be set for each type of instructions.

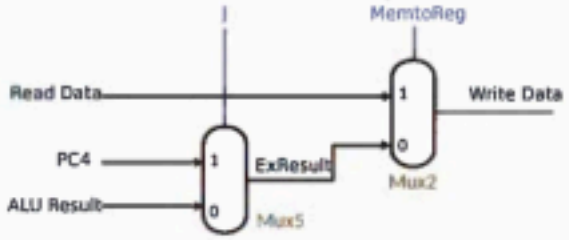
① (7 points) Fill out the table with proper ALU Operation values.

Instruction	**ALUOp**	**Instr[30]**	**funct3**	ALU Operation
LW/SW	0 0	x	x	0 0
BEQ	0 1	x	x	0 1
ADD	1 0	0	000	0 0
SUB	1 0	1	000	0 1
AND/ANDI	1 1	x	111	1 0
OR/ORI	1 1	x	110	1 1
ADDI	1 1	x	000	0 0

② (3 points) Explain with one or two sentences why you select the ALU Operation value you have put in the table for BEQ.

BEQ checks whether 2 registers have equal value. To do so, it subtracts the values. If the result is zero, the zero signal output of ALU is 1, which means both values are the same. (Subtracting a number from itself gives 0). The zero signal and branch signal can then be used to set PCSrc.

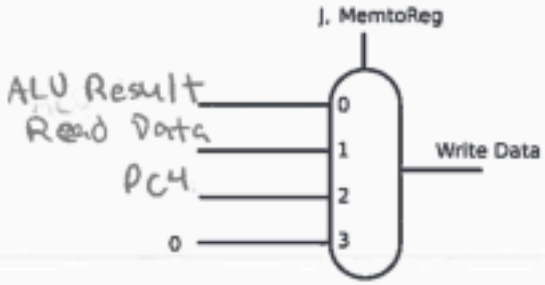
(d) (6 points) In a homework question, we improve the single-cycle processor to support JAL and JALR instructions. The following diagram sets correct Write Data for JAL and JALR instructions, using two 2-1 MUXes. Recall that signal J is set to 1 for both JAL and JALR.



① (3 points) Fill out the following table that shows how the above circuit sets Write Data (to be one of Read Data, PC4, or ALU Result) for different J and MemtoReg values.

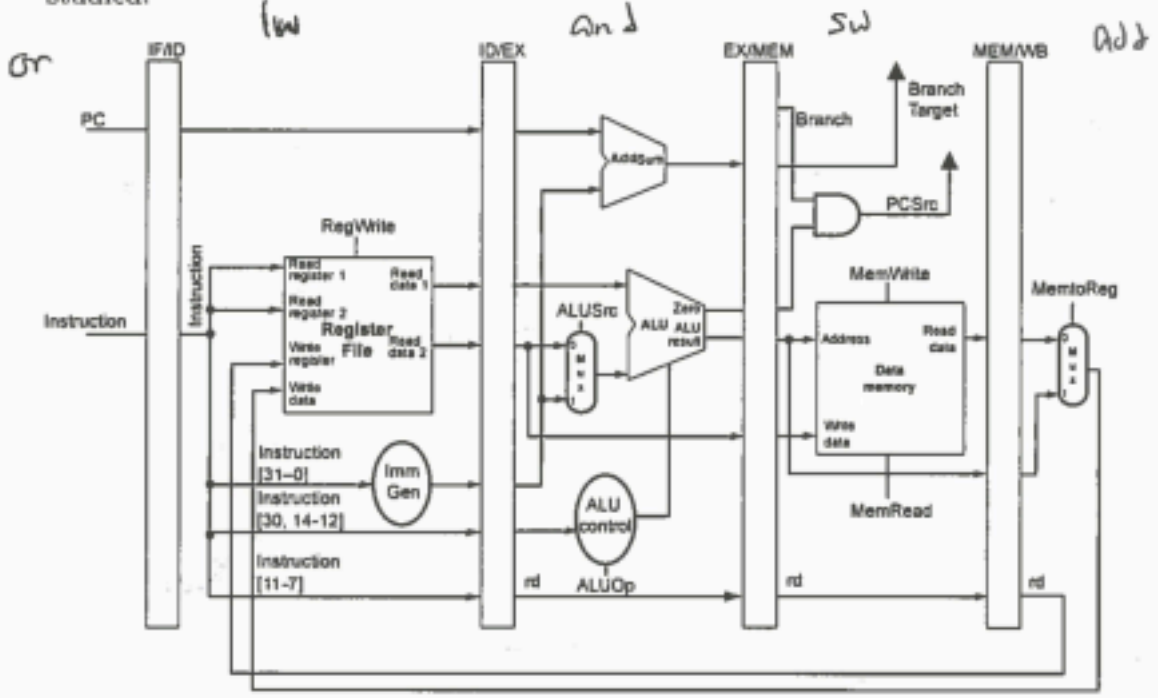
J	MemtoReg	Write Data
0	0	ALU Result
0	1	Read Data
1	0	PC4
1	1	Read Data

② (3 points) If a 4-1 MUX is faster than two 2-1 MUXes in series, how would we use it to replace the two MUXes in the above figure? In the following figure, label inputs 0, 1, and 2 of the 4-1 MUX with proper signal names. The select signal of the 4-1 MUX is J and MemtoReg, and J is the left (higher) bit. We assume the control module does not assert both J and MemtoReg at the same time. So input 3 is set to 0.



4. (25 points) Processor pipeline.

(a) (10 points) The following figure is part of the 5-stage RISC-V pipeline we have studied.



In a cycle, the following instruction sequence is in the pipeline. The ADD instruction is in the WB stage. PC in IF is 0x0440AA30.

```
        add     x21, x1, x2     # WB
        sw      x22, 4(x2)
        and     x23, x5, x6
        lw      x24, 64(x2)
        or      x25, x7, x8
```

Fill in the following blanks with the signal values that have become stable in the cycle. Write U if the value cannot be determined.

ID/EX.ALUSrc is 0.

EX/MEM.BranchTarget is 0x0440AA28 in hexadecimal.

Read Register 1 at the register file is 2 in decimal.

WriteRegister at the register file is 21 in decimal.

The output of ImmGen is 64 in decimal.

(b) (8 points) In this question, we study the following three instructions running on a processor that **does not** support forwarding, but detects data hazards and stalls the pipeline if necessary.

```
I1:     lw      x1, 0(x2)
I2:     sw      x1, 16(x2)
I3:     add     x2, x1, x4
```

① (4 points) Complete the following pipeline diagram to show how the three instructions are executed on the processor. The numbers in the first row are cycle numbers. There is no need to add cycle 12 even if any instruction does not reach WB by cycle 11.

	1	2	3	4	5	6	7	8	9	10	11
I1	IF	ID	EX	MEM	WB						
I2		IF	ID	–	–	EX	MEM	WB			
I3			IF	–	–	ID	EX	MEM	WB		

② (4 points) Fill in the blanks.

There are 2 stall cycles during the execution of the 3 instructions.

The processor stalls in cycles 4, 5 (List all the cycle numbers when the processor stalls).

(c) (7 points) In this question, we study the following instruction sequence on a processor that supports all necessary forwarding and stalls the pipeline if needed.

```
I1:     lw      x1, 0(x2)
I2:     sw      x1, 16(x2)
I3:     add     x2, x1, x4
```

You can use the following diagram to help you answer the questions. The diagram will not be graded.

	1	2	3	4	5	6	7	8	9	10	11
I1	IF	ID	EX	MEM	WB						
I2		IF	ID	EX	MEM	WB					
I3			IF	ID	EX	MEM	WB				

① (1 points) If I1 is fetched in cycle 1, the earliest time the new value of `x1` is available in the pipeline is near the end of cycle 4.

② (4 points) List the forwarding paths required in the execution of `I2` and `I3`. Write a forwarding path like `EX/MEM->EX`. Write **None** if no forwarding path is required.

`I2` uses MEM/WB → MEM.

`I3` uses MEM/WB → EX.

③ (1 point) There are 0 stall cycles during the execution of the 3 instructions.

④ (1 point) If the order of `I2` and `I3` is reversed, i.e., the add instruction is written before the sw instruction, the number of stall cycles is 1.

5. (20 points) Performance.

(a) (10 points) Consider a single-cycle processor and a 5-stage pipelined processor we have designed in this course. Assume the major components have the delays listed in the following table and we only consider those delays in this question. In the table, **I-Mem** stands for instruction memory. **Control** is the main control that generates control signals. **RF** stands for Register File. **D-Mem** stands for data memory.

I-Mem	Control	RF Read	ALU	D-Mem	RF Write
50 ns	20 ns	30 ns	40 ns	60 ns	10 ns

Answer the following questions.

① (4 points) What is the minimum clock cycle time of the single-cycle processor? Which instruction decides the cycle time? Which modules work in parallel?

For single cycle processor, minimum clock cycle time is 50 + 30 + 40 + 60 + 10 = 190 ns. Load word decides the cycle time, since it takes the longest, using I-Mem, RF-Read, ALU, D-Mem, and RF-Write. RF-Read and control work in parallel.

② (2 points) What is the minimum clock cycle time of the pipelined processor? Which stage decides the cycle time?

MEM takes the largest so it decides the cycle time. The minimum clock cycle time is 60 ns.

③ (4 points) If the CPI of an application on the pipelined processor is 2.5, what is the speedup of running the application on the pipelined processor versus on the single-cycle processor? Assume both processors run as fast as they can.

Pipelined = $2.5 \cdot 60$ ns

Single Cycle = 190 ns

$$\text{Speedup} = \frac{190}{2.5 \cdot 60} = 1.267$$

(b) (10 points) The table below shows the CPI of different types of instructions on a processor and their percentages of the executed instructions in a program. For example, 20% of the executed instructions are MUL instructions, each taking 15 cycles on average.

Instr. Type	R-Type	Load/Store	Branch	MUL
CPI	1	1.5	3	15
Percentages	30%	30%	20%	20%

Answer the following questions. No point is awarded to work not showing steps.

① (5 points) What percentage of the execution time is spent on the MUL instructions? Round your answer to the nearest tenth of a percent, like, 0.5%.

Total Cycles $= 0.3 \cdot 1 + 0.3 \cdot 1.5 + 0.2 \cdot 3 + 0.2 \cdot 15 = 4.35$

Mul Cycles $= 0.2 \cdot 15 = 3$

$$\frac{3}{4.35} = 68.96 = \boxed{69.0\%}$$

② (5 points) If a compiler replaces each MUL instruction with two R-Type instructions, what is the speedup the compiler achieves with this technique? Round your answer to the nearest hundredths, like, 3.14.

New Total Cycles $= 0.3 \cdot 1 + 0.3 \cdot 1.5 + 0.2 \cdot 3 + (0.2 \cdot 2 \cdot 1)$

$= 1.75$ CP

$$\text{Speedup} = \frac{4.35}{1.75} = 2.485714 = \boxed{2.49}$$