Name (Last, First): SOLUTION KEY

CSCE 230, Fall 2015
Exam 2

Notes:

- This is a closed notes/book exam. You may use a scientific or graphing calculator.
- **This is an individual exam. Please ensure all work is your own.**
- Enter your name, last name first, on the line above. A bonus point will be rewarded to those who do so correctly. Do not open until told to do so.
- This test will be graded out of 100 points. You are allowed to attempt all problems and the problems that will give you the highest score will be considered. Only bonus will be given for any problems marked as bonus.

Material Covered:

Chapter 5

Chapter 6

Chapter 7

Chapter 9

Appendix A

Rubric:

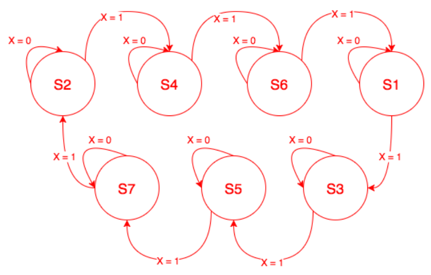
Problem Number	Possible Points	Points Earned
1	30	
2	10	
3	20	
4	20	
5	20	
6	20	
7	20	
8	10	
	Total:	

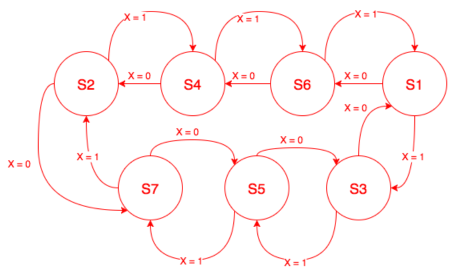
Problem 1 [30]

Topic: *Sequential Circuits*

For some reason, your crazy Computer Organization professor asked you to make a strange counter that counts in the order 2, 4, 6, 1, 3, 5, 7 and then output a signal any time the number is 1. After the counter reaches 7, it restarts back to 2.

a: [7] Using the space below, create a state diagram with the counter states





b: [7] Using the tables below create the state table. Then using the second table assign bit values to the specified states. Be sure to assign a variable to the inputs and output.

Current State	Next State $x = 0$	Next State $x = 1$	Output
S2	S2	S4	0
S4	S4	S6	0
S6	S6	S1	0
S1	S1	S3	1
S3	S3	S5	0
S5	S5	S7	0
S7	S7	S2	0

Current State	Next State $x = 0$	Next State $x = 1$	Output
000	000	001	0
001	001	010	0
010	010	011	0
011	011	100	1
100	100	101	0
101	101	110	0
110	110	000	0

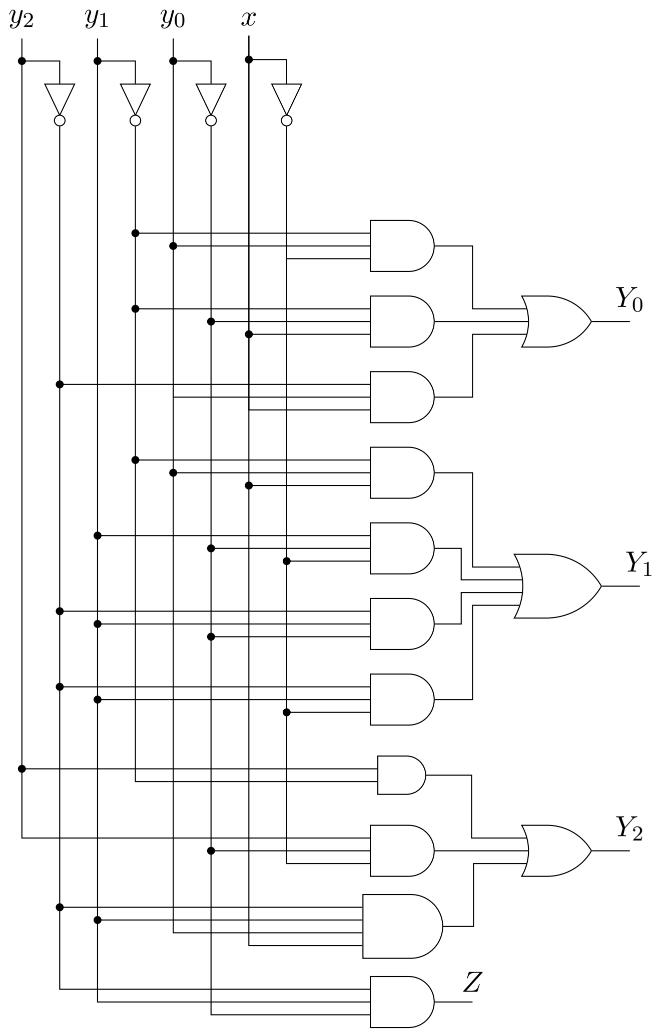
Alternatively you can use this input/output table to determine state values.

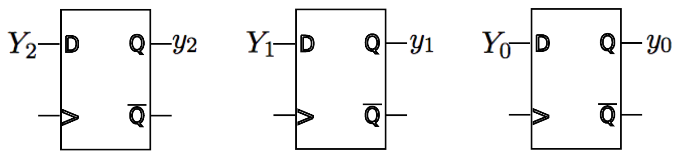
INPUT	**OUTPUT**

c: [8] Using the state table you created, find the minimized logic representation of next state and output variables as a function of the current state and input variables.

$$Y_0 = \bar{y_2}\bar{y_1}\bar{y_0}x + \bar{y_2}\bar{y_1}y_0\bar{x} + \bar{y_2}y_1\bar{y_0}x + \bar{y_2}y_1y_0\bar{x} + y_2\bar{y_1}\bar{y_0}x + y_2\bar{y_1}y_0\bar{x}$$
$$Y_0 = \bar{y_1}y_0\bar{x} + \bar{y_1}\bar{y_0}x + \bar{y_2}y_0x$$
$$Y_1 = \bar{y_2}\bar{y_1}y_0x + \bar{y_2}y_1\bar{y_0}\bar{x} + \bar{y_2}y_1\bar{y_0}x + \bar{y_2}y_1y_0\bar{x} + y_2\bar{y_1}y_0x + y_2y_1\bar{y_0}\bar{x}$$
$$Y_1 = \bar{y_1}y_0x + y_1\bar{y_0}\bar{x} + \bar{y_2}y_1\bar{y_0} + \bar{y_2}y_1\bar{x}$$
$$Y_2 = \bar{y_2}y_1y_0x + y_2\bar{y_1}\bar{y_0}\bar{x} + y_2\bar{y_1}\bar{y_0}x + y_2\bar{y_1}y_0\bar{x} + y_2\bar{y_1}y_0x + y_2y_1\bar{y_0}\bar{x}$$
$$Y_2 = y_2\bar{y_1} + y_2\bar{y_0}\bar{x} + \bar{y_2}y_1y_0x$$
$$Z = \bar{y_2}y_1y_0$$

d: [8] Using the provided D Flip-Flops draw the circuit for the counter.





Problem 2 [10]

Topic: *Arithmetic*

a:[5] Using any manual binary multiplication technique perform the following operation. Please show all work.

$$00011000 \times 0100$$

$$\begin{array}{ccccccccccc}
 & & & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 & & & \times & & & & 0 & 1 & 0 & 0 \\
\hline
 & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & \\
\hline
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}$$

b:[5] Using the manual division technique perform the following operation. Did you use a restoring or non-restoring algorithm? Please show all work.

$$00100100 \div 0100$$

$$\begin{array}{cccc|cccccccc}
 & & & & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
\cline{5-12}
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
 & & & & - & 0 & 1 & 0 & 0 & & & \\
\cline{5-9}
 & & & & & & & & 0 & 1 & 0 & 0 \\
 & & & & & & & - & 0 & 1 & 0 & 0 \\
\cline{8-12}
 & & & & & & & & & & & 0
\end{array}$$

Non-restoring

Problem 3 [20]

Topic: *Basic Processing Unit*

For the following please use the figure on page 10 of this exam.

a: [10] Using the following instruction fill out the table below. Assume R3 contains the value 68.

```
ADD R4, R3, #32
```

Stages:	**1**	**2**	**3**	**4**	**5**
RA:			68	68	68
RB:					
RM:					
RZ:				100	100
RY:					100

b: [10] Using the following instruction fill out the table below. Assume R3 contains the value 800 and the value at memory location 800 contains 100.

```
LOAD R4, 0(R3)
```

Stages:	**1**	**2**	**3**	**4**	**5**
RA:			800	800	800
RB:					
RM:					
RZ:				800	800
RY:					100

Problem 4 [20]

Topic: *Multicycle Datapath*

Using the diagrams on the following pages explain how to add the command "BNQ" or branch if not equal. Feel free to draw on any of the diagrams and reference the drawings for a complete answer. *HINT:* You **will** need to add items on all three figures.

To include a "BNQ" instruction in the multicycle datapath, another OP Code must be included. This will take the signal "ZERO" and AND the inverse of the signal with a second "PCWriteCond" signal. This signal would be produced by the control unit during a new state 10. All other signals will be identical to state 8. This would share the OR gate with "BEQ".

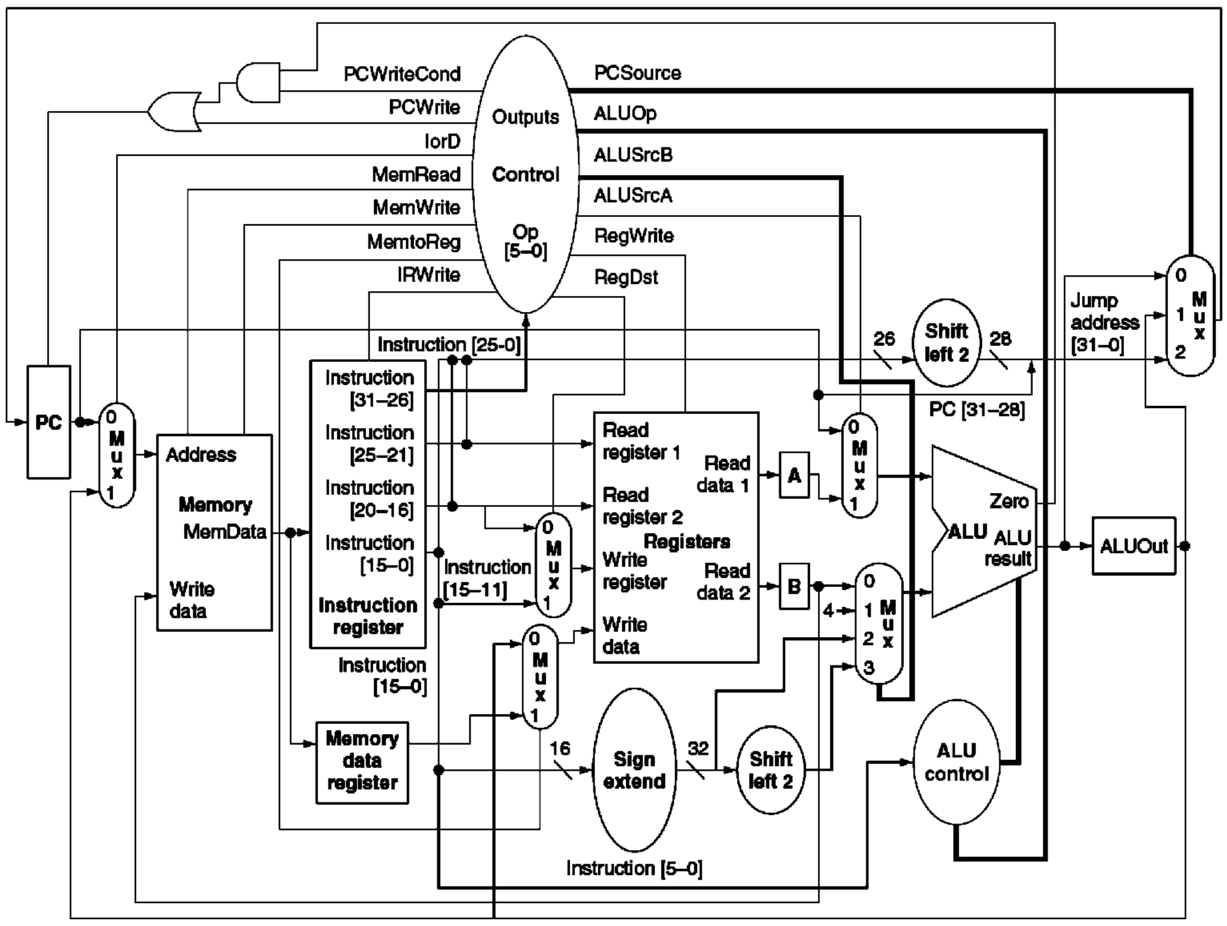


Figure 1: Complete datapath for the multicycle implementation.

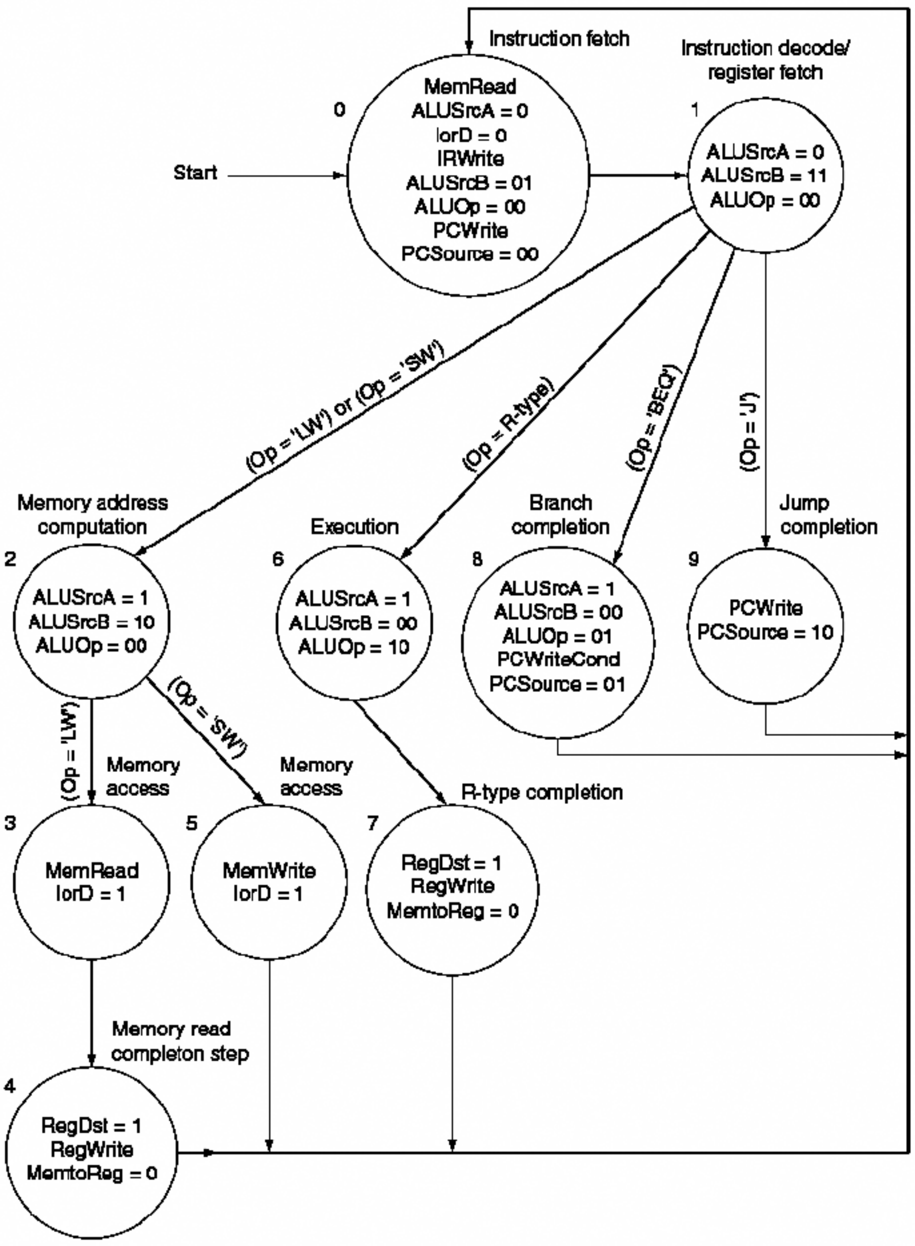


Figure 2: Finite State Machine for the multicycle implementation.

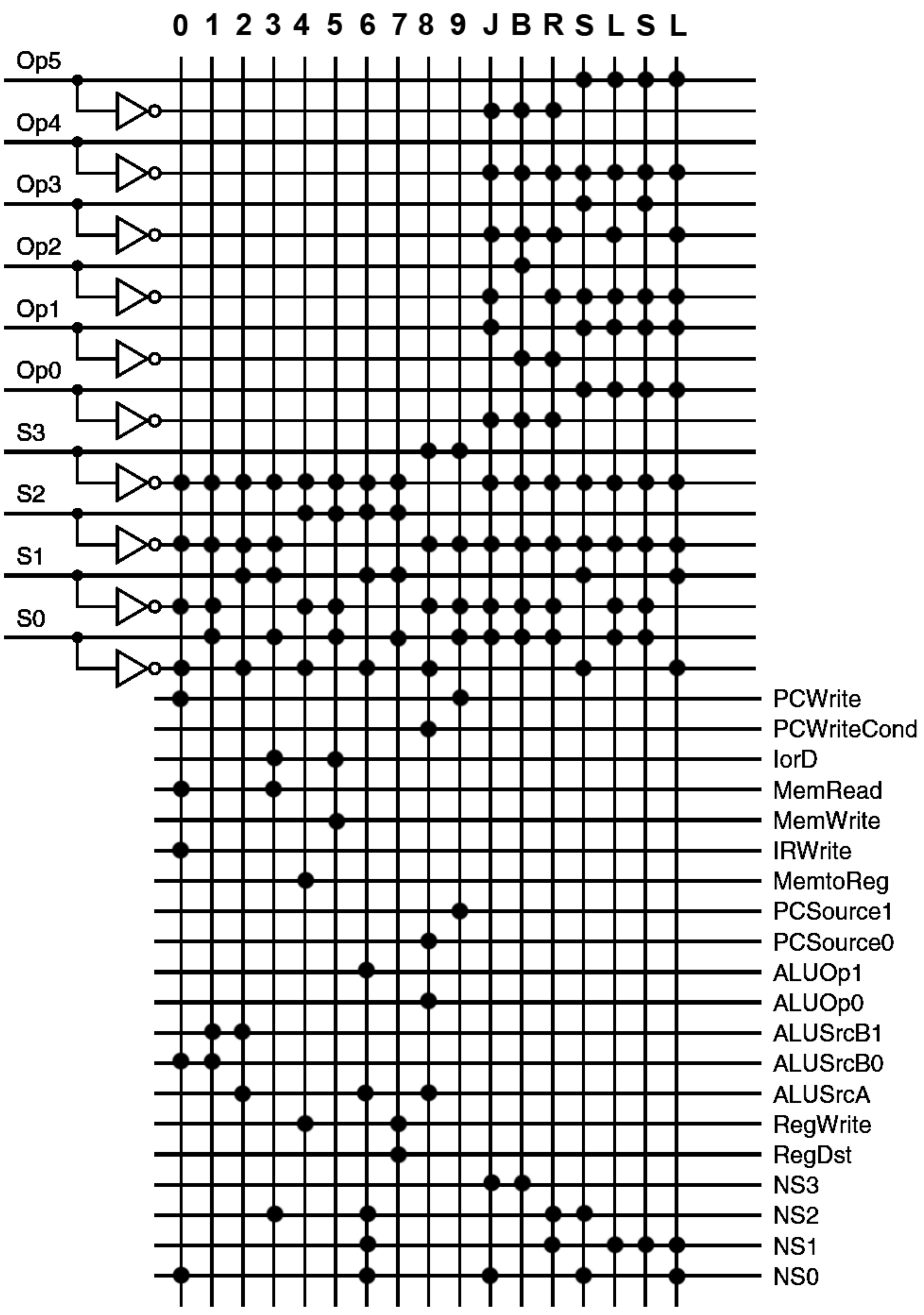


Figure 3: PLA implementing the control function logic for the multicycle implementation.

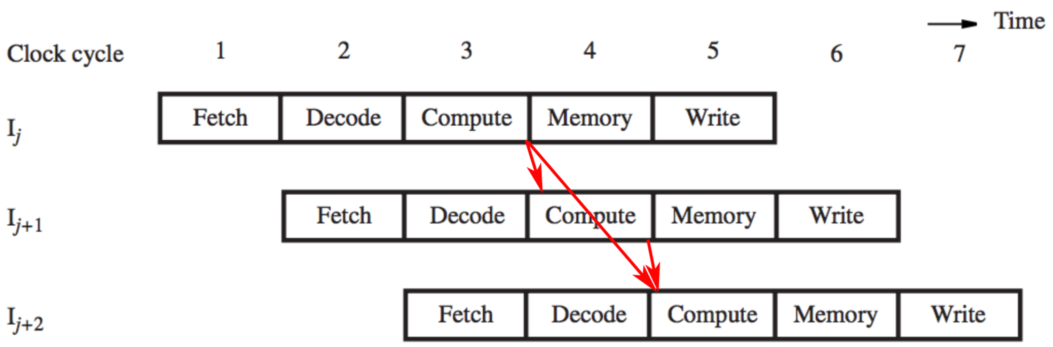
Problem 5 [20]

Topic: *Pipeline*

Consider the following instructions in a pipeline processor with operand forwarding implemented.

```
Add         R2, R3, #100
Subtract    R9, R2, #30
Or          R6, R9, R2
```

a:[10] Using the figure below specify any data dependencies that will require operand forwarding.
I_{j+1} instruction need forwarding from I_j instruction. I_{j+2} instruction needs forwarding from I_j and I_{j+1}



b:[10] Assuming R3 contains 85, complete the table below for the specified interstage buffer. Specify any operand forwarding from the previous problem.

Cycles:	1	2	3	4	5	6	7
RA:			85	R2?	R9?		
RB:					R2?		
RM:							
RZ:				185	155	187	
RY:					185	155	187

R2? and R9? means that the value is unknown but it's in that register

Problem 6 [20]

Topic: *Short Answer*

a:[7] In the basic processing unit an Add instruction is executed in five steps, but no processing actions take place in step 4. If it is desired to eliminate that step, what changes have to be made in the datapath to make this possible? Please be specific.

Step 4 can be skipped by sending the output of the ALU directly to register RY. This can be accomplished by adding one more input to multiplexer MuxY and connecting that input to the output of the ALU. Thus, the result of a computation at the output of the ALU is loaded into both registers RZ and RY at the end of step 3. For an Add instruction, or any other computational instruction, the register file control signal RF_write can be enabled in step 4 to load the contents of RY into the register file.

b:[7] Any condition that causes the pipeline to stall is called a hazard. Describe one of these hazards and give an example of a hardware or software solution to limit the stall on the processor.

Data Dependencies: Operand forwarding can be used to alleviate pipeline stalls.
Memory Delays: The compiler can eliminate stalls by reordering instructions to insert a useful instruction between the Load instruction and the instruction that depends on the data read from the memory.
Branch Delays: Branch prediction can be used reduce the stalls occurred by concurrent branches.

c:[6] An asynchronous bus utilizes a *handshake* protocol. Briefly describe the handshake control of data transfer during an input operation.

The master places the address and command information on the bus. Then it indicates to all devices that it has done so by activating the Master-ready line. This causes all devices to decode the address. The selected slave performs the required operation and informs the processor it has done so by activating the Slave-ready line. The master waits for Slave-ready to become asserted before it removes its signals from the bus.

Problem 7 [20]

Topic: *I/O Organization*

Take a deep breath. Assume you have three devices that can write data to a bus line. However, these devices have a priority scheme. Device 2 needs direct access to the bus line before the other two devices. Device 1 has a lower priority than Device 2 and Device 3 has the lowest priority.

a: [10] What can be implemented to determine these priorities? Briefly describe how this works.
Arbitrator or Priority Encoder. It works by controlling which device can use the bus line.

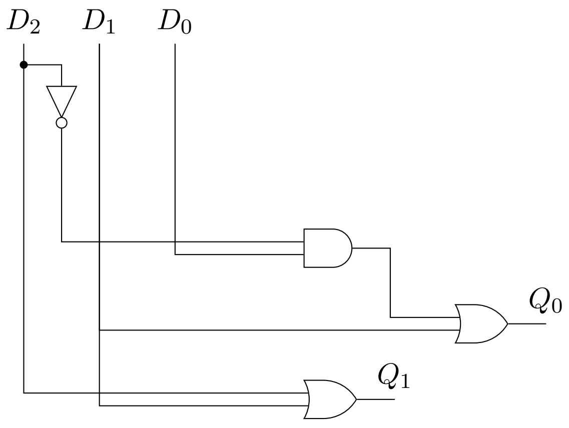
b: [10] Implement the device you mentioned in **Part A** to create a priority. Assume the larger the number, the higher the priority.

Truth table of arbitrator:

D_2	D_1	D_0	Q_1	Q_0
X	1	X	1	1
1	0	X	1	0
0	0	1	0	1

$$Q_0 = D_1 + \bar{D_2}D_0$$

$$Q_1 = D_2 + D_1$$



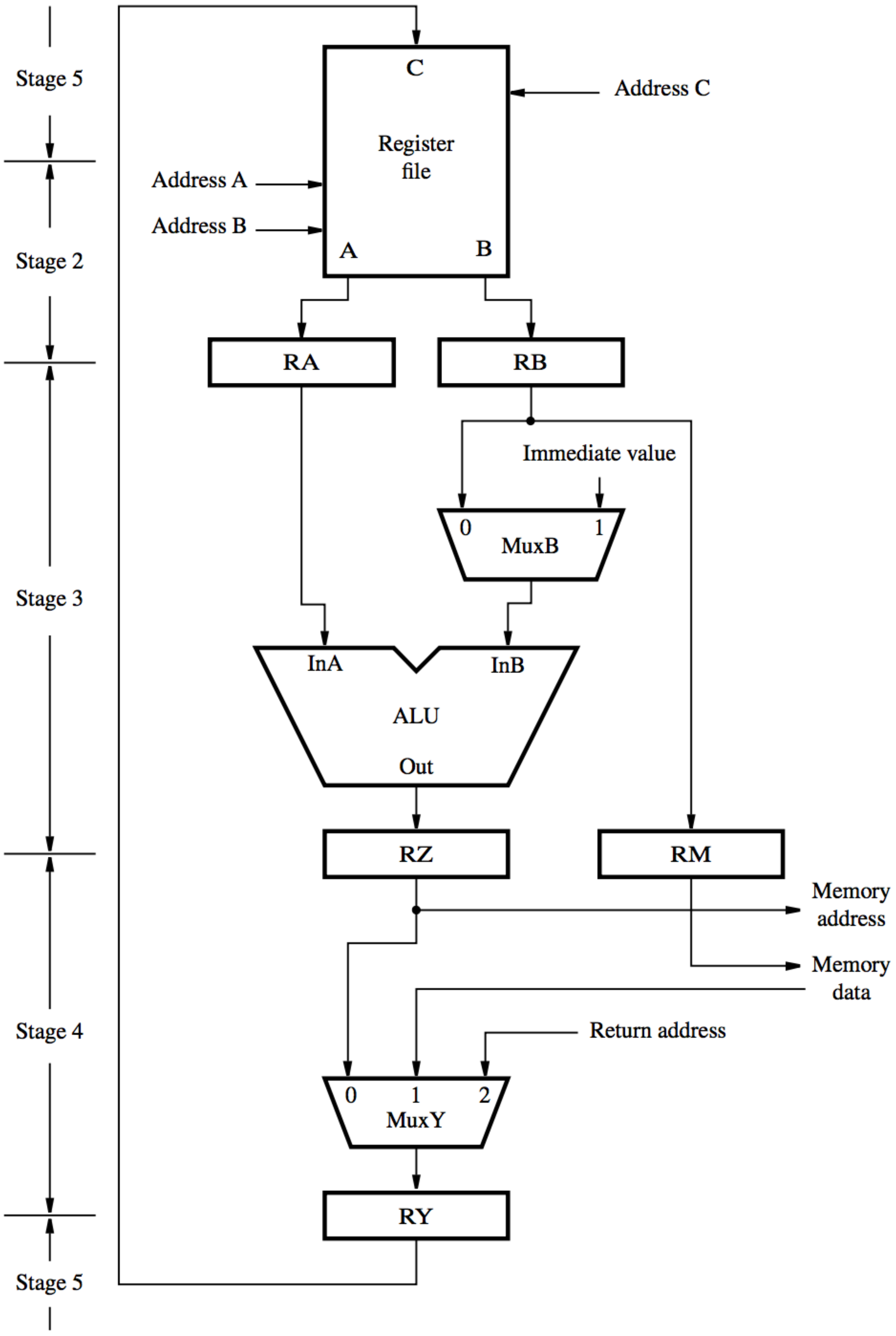
Problem 8 [10]

Topic: *Bonus*

NOTE: *This problem may be difficult so make sure you have completed most of the previous problems before attempting this problem.*

Using the 5-stage basic processing unit, what hardware would you need to add to circuitry to make an operation that increments the value in a specified address in memory *without using any general purpose registers.* Feel free to use the diagram on the following page to draw, or simply explain how you would implement this.

To implement this, another adder could be added into stage 4 that gets its inputs from the output of the memory data output and a constant value. The output of this adder can be directed to the memory data input and a write enable flag can be set to write it back into memory.



Stage 5
Stage 2
Stage 3
Stage 4
Stage 5
C
Register file
Address C
Address A
Address B
A
B
RA
RB
Immediate value
0
1
MuxB
InA
InB
ALU
Out
RZ
RM
Memory address
Memory data
Return address
0
1
2
MuxY
RY