EECS 150 **Problem Set 8 Solutions**
Fall 1999 **prepared by Ilyas Nuri, revised by Ron Fearing**

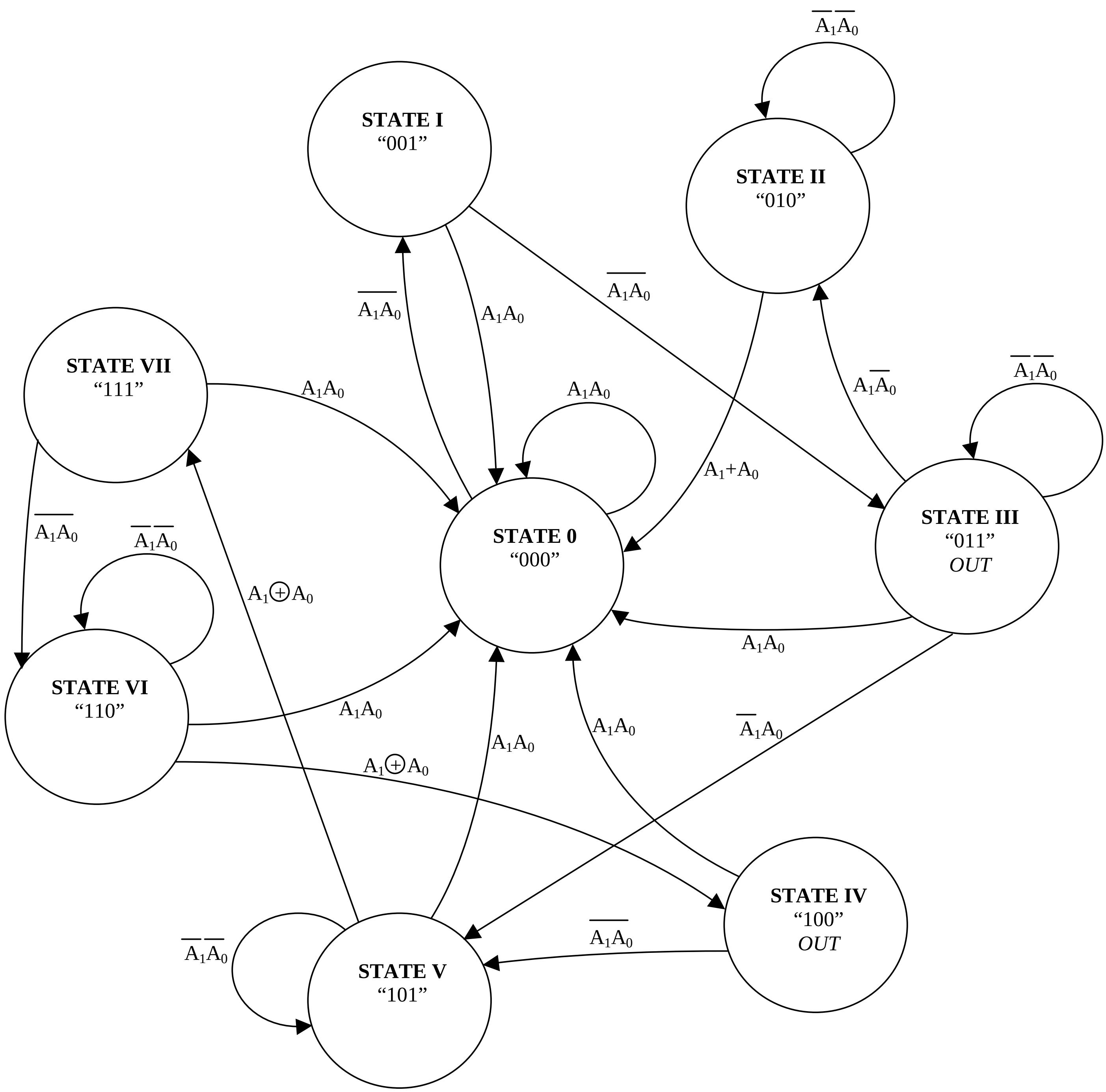
1. ROM Based FSM (20pts)

a) State Table

Address	A1.H	A0.H	Q2.H	Q1.H	Q0.H	Data	OUT	Q2	Q1	Q0
0	0	0	0	0	0	1	0	0	0	1
1	0	0	0	0	1	3	0	0	1	1
2	0	0	0	1	0	2	0	0	1	0
3	0	0	0	1	1	B	1	0	1	1
4	0	0	1	0	0	D	1	1	0	1
5	0	0	1	0	1	5	0	1	0	1
6	0	0	1	1	0	6	0	1	1	0
7	0	0	1	1	1	6	0	1	1	0
8	0	1	0	0	0	1	0	0	0	1
9	0	1	0	0	1	3	0	0	1	1
A	0	1	0	1	0	0	0	0	0	0
B	0	1	0	1	1	D	1	1	0	1
C	0	1	1	0	0	D	1	1	0	1
D	0	1	1	0	1	7	0	1	1	1
E	0	1	1	1	0	4	0	1	0	0
F	0	1	1	1	1	6	0	1	1	0
10	1	0	0	0	0	1	0	0	0	1
11	1	0	0	0	1	3	0	0	1	1
12	1	0	0	1	0	0	0	0	0	0
13	1	0	0	1	1	A	1	0	1	0
14	1	0	1	0	0	D	1	1	0	1
15	1	0	1	0	1	7	0	1	1	1
16	1	0	1	1	0	4	0	1	0	0
17	1	0	1	1	1	6	0	1	1	0
18	1	1	0	0	0	0	0	0	0	0
19	1	1	0	0	1	0	0	0	0	0
1A	1	1	0	1	0	0	0	0	0	0
1B	1	1	0	1	1	8	1	0	0	0
1C	1	1	1	0	0	8	1	0	0	0
1D	1	1	1	0	1	0	0	0	0	0
1E	1	1	1	1	0	0	0	0	0	0
1F	1	1	1	1	1	0	0	0	0	0

Note that this solution assumes “1” for the data value of address 8.

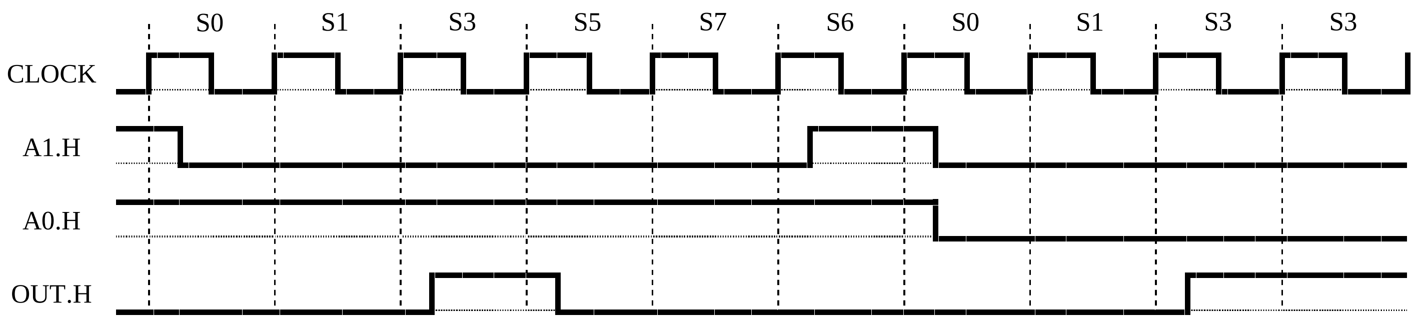
b) State Diagram



This is a Moore machine because the output, OUT, depends only on the present state, $Q_2Q_1Q_0$.

c) Timing Diagram

Initially, assume we are in state 0.



2. Instruction Execution (45pts)

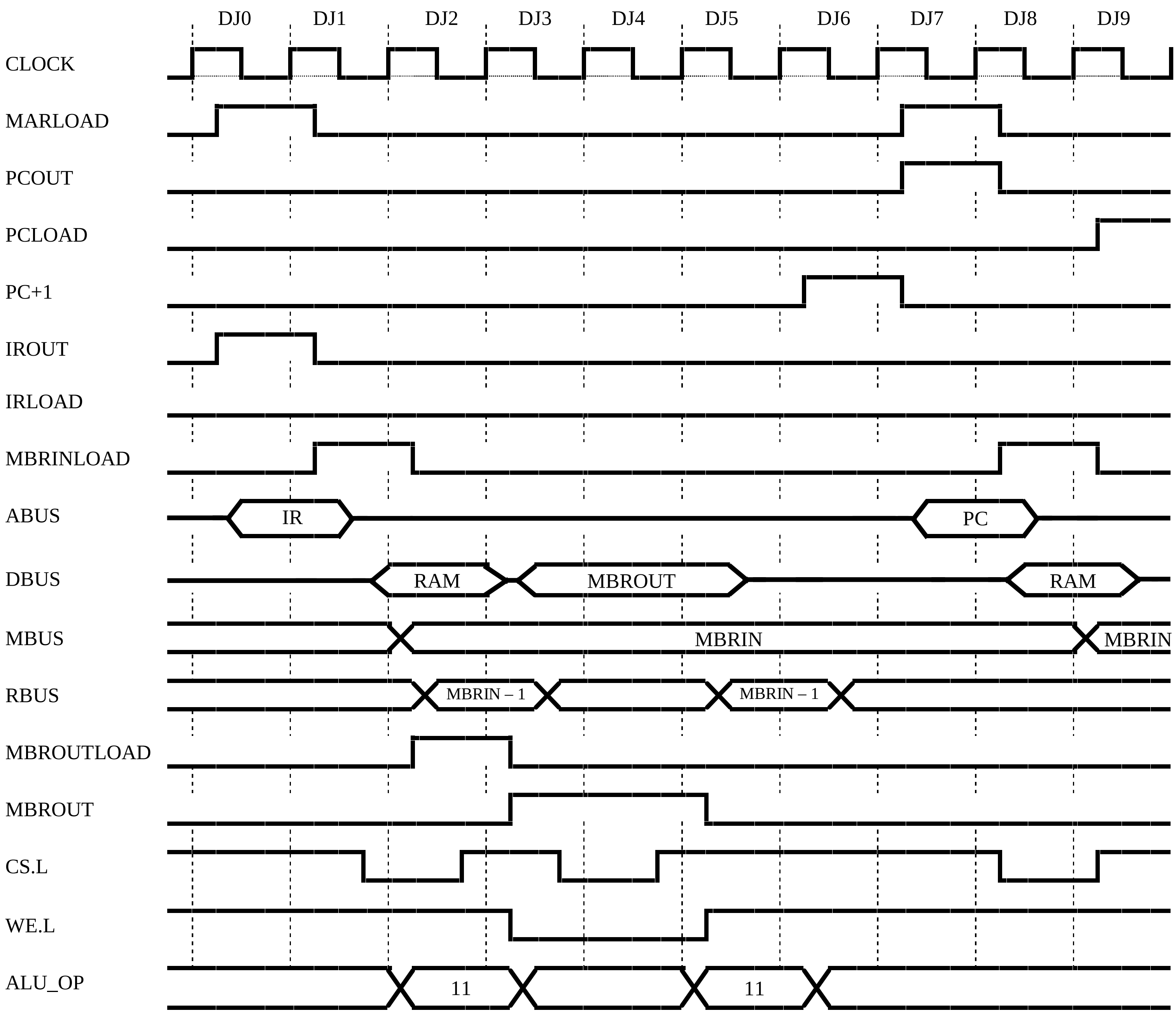
a) Change reserved instruction for ALU to be Y = B – 1.
Also, add a 16 bit wide NOR gate on output of Y to detect Y == 0. Note that this is NOT an extra ALU operation.

b)

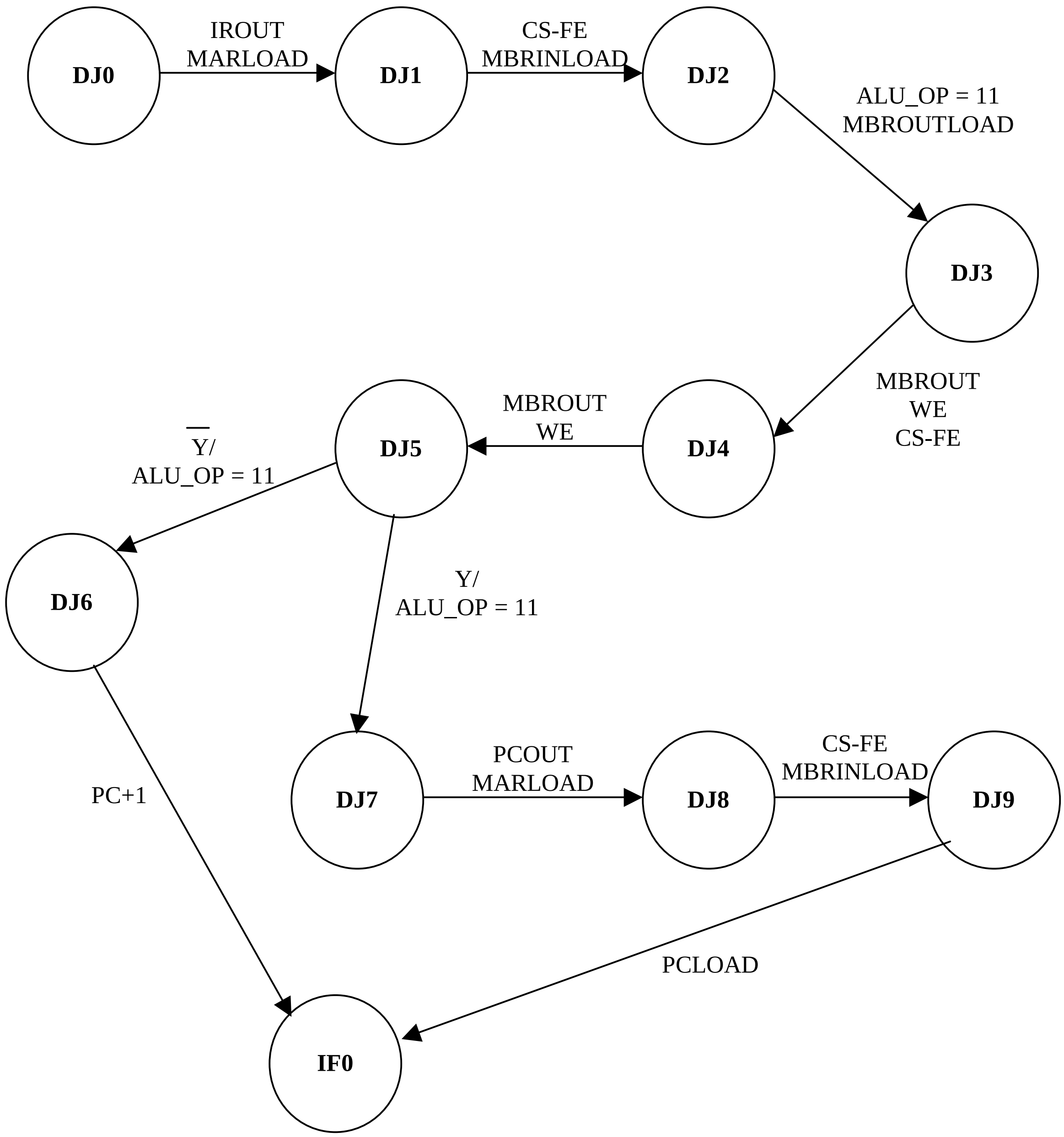
Operation	State	Register Transnfer Notation	Microoperation
fetch	IF0	PC -> MAR, PC+1 -> PC	PCOUT, MARLOAD, PC+1
		PC is now pointing to the second word of DJNZ instruction (brn-addr)	
	IF1	RAM[MAR] -> MBRIN	CS-FE, MBRINLOAD
	IF2	MBRIN -> IR	IRLOAD
decode	DJ0	IR -> MAR	IROUT, MARLOAD
	DJ1	RAM[MAR] -> MBRIN	CS-FE, MBRINLOAD
		Decrement mem_addr (RAM[IR<13:0>])	
	DJ2	MBRIN – 1 -> MBROUT	ALU_OP -> Y = B – 1 MBROUTLOAD
		Put the result back into RAM using a two-cycle write	
	DJ3	MBROUT -> RAM[MAR]	MBROUT, WE, CS-FE
	DJ4		MBROUT, WE
		Decide if we should branch or not	
	DJ5	If Y ≠ 0, goto DJ7	ALU_OP -> Y = B – 1
		Y = 0, so just execute the next intruction	
	DJ6	PC+1 -> PC, goto IF0	PC+1
		Y ≠ 0, so branch	
	DJ7	PC -> MAR	PCOUT, MARLOAD
	DJ8	RAM[MAR] -> MBRIN	CS-FE, MBRINLOAD
	DJ9	MBR -> PC, goto IFO	PCLOAD

Note that the fetch operation is provided for completeness.

c) Only the CS, WE, and ALU_OP signals during the decode operation is shown in this timing diagram:

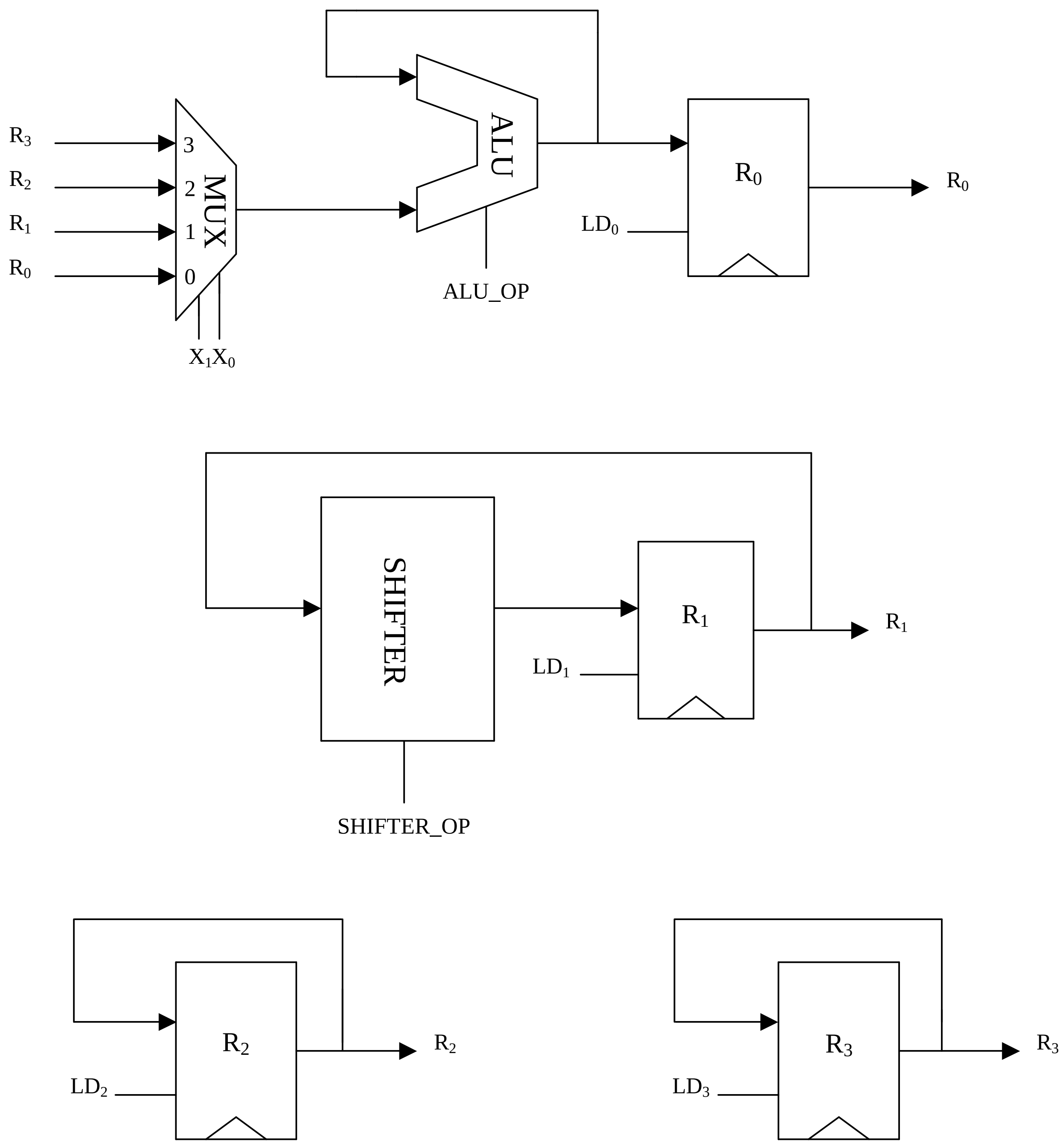


d) Decode execution sequence:



3. Katz Ch.11, Problem 6 (35pts)

a) Using the point-to-point register interconnection scheme in Figure 11.8, the following datapath is appropriate:

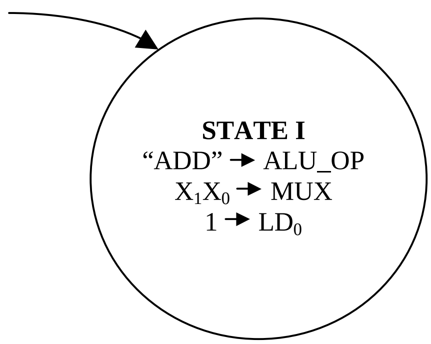


Note that in this datapath there are no input muxes on R_1, R_2 or R_3. Since the question is not very specific on R_2 and R_3, it's okay if you included muxes in your datapath.

b)

Register Transfer Notation	**Microoperation**
$ALU(R_0, R[X_1X_0]) \rightarrow R_0$	$ALU\_OP \rightarrow ALU$ $X_1X_0 \rightarrow MUX$ $1 \rightarrow LD_0$
$SHIFTER(R_1)$	$SHIFTER\_OP \rightarrow SHIFTER$ $1 \rightarrow LD_1$
$R_2 \rightarrow R_2$	$1 \rightarrow LD_2$
$R_3 \rightarrow R_3$	$1 \rightarrow LD_3$

c) One state is required to execute ADD.



d) One state is required to execute RSR.