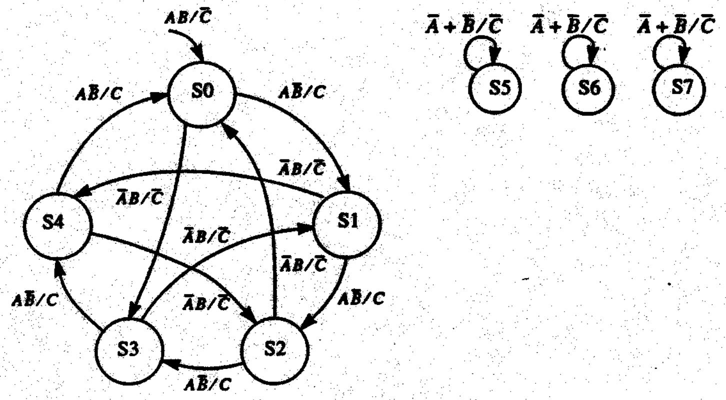
Problem 1 (10 points) KEY

Fall 1998 mean = 67
s.d. = 15

Complete the state table for the Mealy-type FSM: Holding in a state is implicit. Unspecified outputs are implicitly unasserted.



Inputs A B	Present State	Next State	Output
0 0	S0	S0	0
0 1		S3	0
1 0		S1	1
1 1		S0	0
0 0	S1	S1	0
0 1		S4	0
1 0		S2	1
1 1		S0	0
0 0	S2	S2	0
0 1		S0	0
1 0		S3	1
1 1		S0	0
0 0	S3	S3	0
0 1		S1	0
1 0		S4	1
1 1		S0	0
0 0	S4	S4	0
0 1		S2	0
1 0		S0	1
1 1		S0	0
0 0	S5	S5	0
0 1		S5	0
1 0		S5	0
1 1		S0	0
0 0	S6	S6	0
0 1		S6	0
1 0		S6	0
1 1		S0	0
0 0	S7	S7	0
0 1		S7	0
1 0		S7	0
1 1		S0	0

Problem 2 (14 points) KEY

[4 pts.] a) Find the minimal sum-of-products form for

$$Y = \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D} + \bar{A}B(\bar{C}D + CD) + AB(CD + \bar{C}D)$$

$$\boxed{Y = B\cdot D + \bar{B}\cdot\bar{D}}$$

CD \ AB	00	01	11	10
00	1			1
01		1	1	
11		1	1	
10	1			1

[4 pts.] b) Find the minimal product-of-sums form for

$$Y = A\cdot\bar{D} + \bar{A}\bar{B}\cdot D$$

$$\boxed{Y = (\bar{A}+\bar{D})(A+D)(A+\bar{B}) \quad \text{or} \quad (\bar{A}+\bar{D})(A+D)(\bar{B}+\bar{D})}$$

A \ BD	00	01	11	10
0	0	1	0	0
1	1	0	0	1

$\bar{Y} = \bar{A}\bar{D} + A\cdot D + \bar{A}\cdot B$

or $\bar{Y} = A\cdot D + B\cdot D + \bar{A}\bar{D}$

[6 pts.] c) *State minimization.* For the following state table, determine which states are equivalent.

Present State	Input	Output	Next State
S0	0	0	S0
S0	1	0	S1
S1	0	1	S2
S1	1	0	S6
S2	0	0	S3
S2	1	0	S6
S3	0	0	S2
S3	1	0	S6
S4	0	1	S4
S4	1	0	S4
S5	0	1	S5
S5	1	0	S4
S6	0	1	S3
S6	1	0	S2

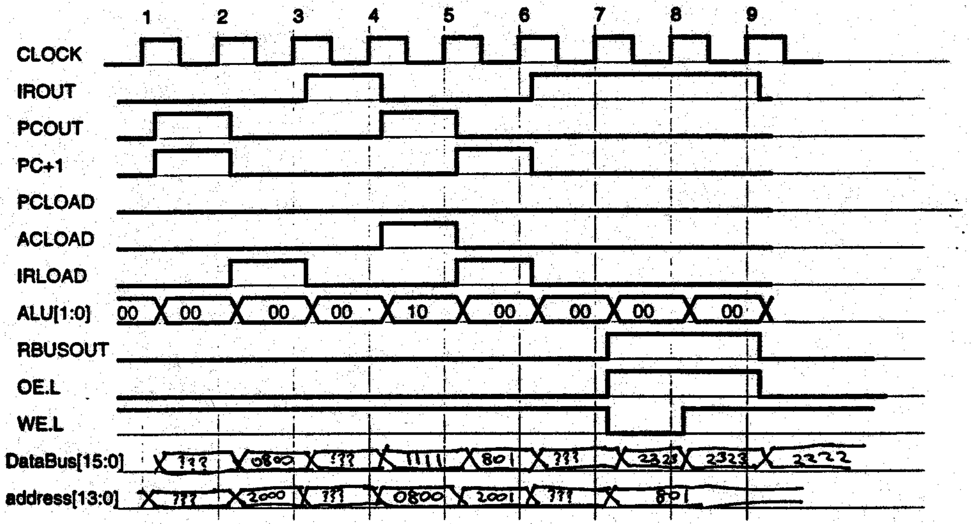
	S0	S1	S2	S3	S4	S5
S1	X					
S2	S0 S3 S1 X S6	X				
S3	S0 S2 S1 X S6	X	S3 S2 S6 S6			
S4	X	S2 S4 S6 X S4	X	X		
S5	X	S2 S5 S6 X S4	X	X	S4 S5 S4 S4	
S6	X	S2 S3 S6 X S2	X	X	S4 S3 S4 X S2	S5 S3 S4 X S2

$$\boxed{\begin{array}{l} S3 = S2 \\ S5 = S4 \end{array}}$$

Problem 3 (24 points) KEY

[8 pts.] a) Using the data path on page 8, and the timing diagram below, list in register transfer notation (RTN), the operation which is occurring on the rising edge of the clock. (Note: This timing diagram is independent of the controller on page 9.)

clock edge 1: RTN garbage → MAR
clock edge 2: RTN PC → MAR, pc+1 → PC   PC=2001, MAR=2000
3: RTN RAM[MAR] → IR, garbage → MAR   IR=0800   MAR=???
4: RTN IR → MAR   MAR=0800
5: RTN PC → MAR, RAM[MAR] plus AC → AC   MAR=2001, RAM[0800]+AC → AC   AC=2323
6: RTN PC+1 → PC, RAM[MAR] → IR, garbage → MAR   RAM[2001] → IR=801
7: RTN IR → MAR   MAR=801
8: RTN IR → MAR, AC → RAM[MAR]



(CS.L tied to ground)
MARLOAD=1

KEY

[8 pts.] b) The initial values for the registers are: PC = 0 × 2000, IR = 0 × 0800, AC = 0 × 1212. The RAM contents are:

Address	Data
0 × 0800	0 × 1111
0 × 0801	0 × 2222
0 × 0802	0 × 3333
0 × 2000	0 × 0800
0 × 2001	0 × 0801
0 × 2002	0 × 0802

Complete the timing diagram above for the Data Bus and address bus. If value is undefined, indicate by ???. Also indicate tri-state (HiZ) conditions.

[3 pts.] c) Assuming WE is clean, considering timing skew and possible glitches, is there a potential problem near the rising edge of clock 7? If so, briefly state the problem.

There is a potential problem since the address lines are changing just after the rising edge of clock 7. It is possible that data could be written to multiple addresses. Also, there is the possibility of bus conflict due to OE.L and RBusOUT being asserted at the same time.

[3 pts.] d) Assuming WE is clean, considering timing skew and possible glitches, is there a potential problem near the rising edge of clock 8? If so, briefly state the problem.

ALU control lines and RBusOUT control line may have glitches after the rising edge of clock 8. If the WE is delayed by timing skew, incorrect data could be written to RAM.

[2 pts.] e) Is the memory address register needed for this design? Explain why or why not.

Yes. Otherwise the address lines would not be held stable during a write cycle. At clock edge 6, IR → MAR needs to load the address register a clock cycle before WE is asserted to prevent the problem mentioned in part c) above.

Problem 4 Microprogramming (10 points) KEY

Using the data path on page 8 and microprogrammed controller on page 9, write a microprogram, in symbolic form, to execute the following algorithm: (Recall AC<15> is bit 15 of AC.)

```
IR → MAR
while (AC<15>=0)
{RAM [MAR] plus AC → AC, IR → MAR
 PC+1 → PC, IR → MAR}
AC → RAM [MAR]
PC → MAR
RAM [MAR] → IR
```

MAR needs to be loaded with IR throughout this section.

You have two micro-instructions available, DO and JMP.

Label	Operation	Comment
	Do IRoUT	IR→MAR
	JMP(AC<15>) NEXT	garbage→MAR during JMP
LOOP	Do IRoUT	IR→MAR
	Do IRoUT, ACLOAD, Y=A+B, READ	RAM[MAR]+AC→AC
	Do IRoUT, PC+1	PC+1→PC, IR→MAR
	JMP (AC<15>) LOOP	(can read and PC+1 on same clock if desired, since OE is asserted by default.
NEXT	Do IROUT	
	Do IRoUT, Y=A, WRITE0	AC→RAM[MAR]
	Do PCouT, Y=A, WRITE1	PC→MAR
	Do IRLOAD, READ	RAM[MAR]→IR

Problem 5 (12 points)

Using the data path on page 8, draw a state diagram(s) for a Moore-type FSM(s) which will control the following operations:

1) IR → MAR

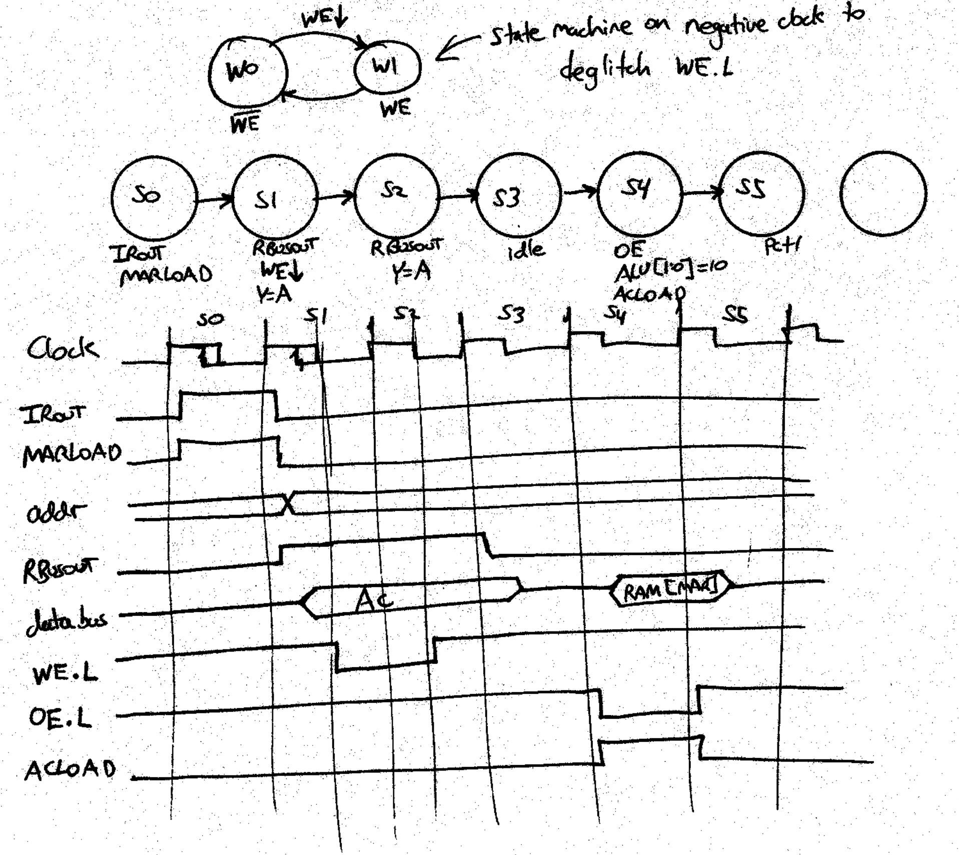
2) AC → RAM [MAR]

3) AC plus RAM [MAR] → AC

4) PC+1 → PC

Necessary control signals:
IRouT, MARLOAD, RBUSOUT, OE, WE, PC+1, ACLOAD, ALU[1:0]

Your state diagram must show all asserted control signals and your design should tolerate timing skew without causing bus conflicts or incorrect memory operations. Explicitly show deglitching states.



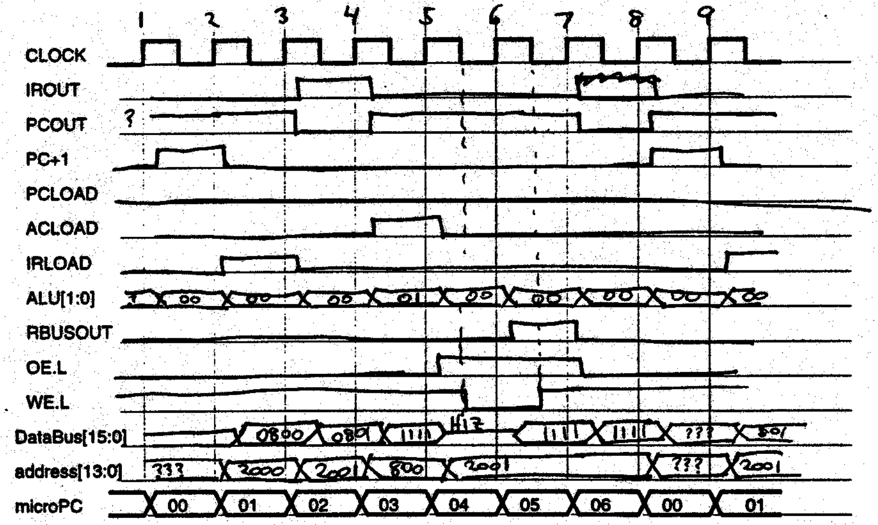
State machine on negative clock to deglitch WE.L

Problem 6 FSM Microprogram Analysis (30 points) KEY

Complete the timing diagram for the computer data path on page 8 and the microprogrammed controller shown on page 9. All components are synchronous. Partial microprogram ROM contents in Hexadecimal are:

Address	Hex Data	SRC	DST	ALU	MEM				
0	0×73	1	11	00	11	PCOUT	NOP	Y=A	PC+1
1	0×60	1	10	00	00	PCOUT	IRLOAD	Y=A	READ
2	0×30	0	11	00	00	IROUT	NOP	Y=A	READ
3	0×44	1	00	01	00	PCOUT	ACLOAD	Y=B	READ
4	0×71	1	11	00	01	PCOUT	NOP	Y=A	WRITE0
5	0×72	1	11	00	10	PCOUT	NOP	Y=A	WRITE1
6	0×E0	JMP(true)	000						

The initial values for the data path registers are: PC = 0×2000 IR = 0×0900, AC = 0×1212. The RAM contents are: address 0×0800 = 0×1111; address 0×0801 = 0×2222; address 0×0802 = 0×3333; address 0×2000 = 0×0800; address 0×2001 = 0×0801; address 0×2002 = 0×0802.



(CS.L tied to ground) MARLOAD=1

↑Write 1111 to RAM[2001]

clock edge

3) RAM[2000] → IR IR=0800
PC→MAR

4) IR→MAR

5) PC→MAR, RAM[800] →AC (AC=1111)