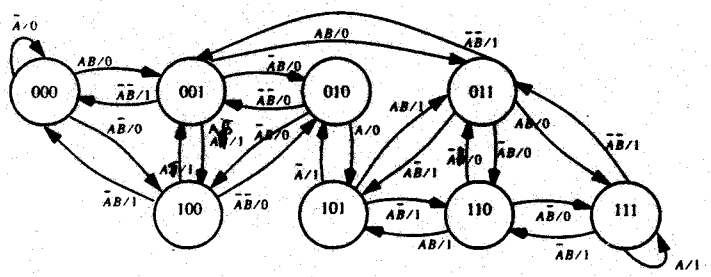
Problem 1 (16 points) KEY

Fall 199 Quiz 2. Ave. ≈ 50
max = 88

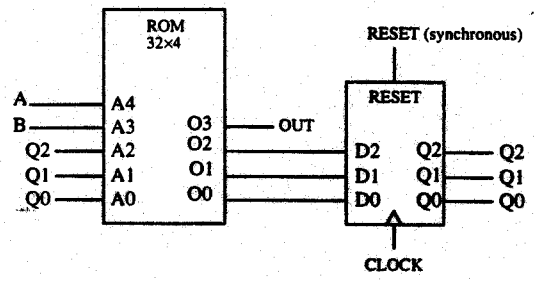
List the ROM contents in **hexadecimal** to implement the FSM shown below. The inputs A and B are synchronized. The states are assigned numerical order, e.g., for state S4, $Q_2Q_1Q_0 = 100_2$. (Follow normal state diagram assumptions: holding in the same state is implicit, etc.).

Fill in ROM contents in hexadecimal. (Binary answers will receive no credit.)

	Address	Data	Address	Data	Address	Data	Address	Data
S0	0	0	8	0	10	4	18	1
	1	8	9	2	11	C	19	3
	2	1	A	4	12	5	1A	5
	3	9	B	6	13	D	1B	7
	4	2	C	8	14	9	1C	9
	5	A	D	A	15	E	1D	B
	6	3	E	3	16	7	1E	D
S7	7	B	F	E	17	F	1F	F
		$\bar{A}\bar{B}$		$\bar{A}B$		$A\bar{B}$		AB

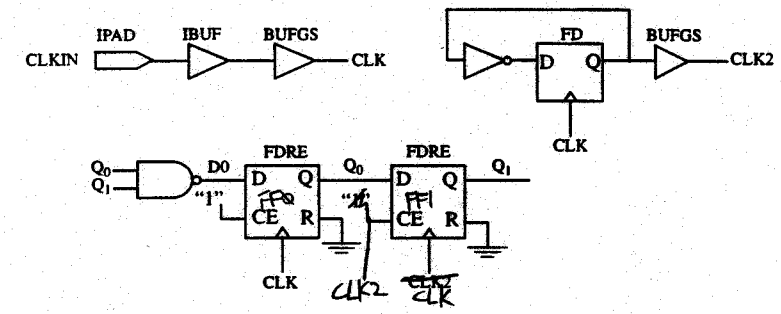


OUT	Q_2	Q_1	Q_0

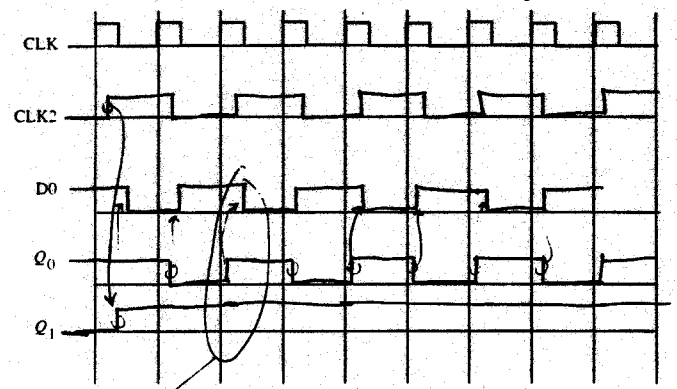


Problem 2 (24 points) KEY

Your lab partner has designed the following Xilinx XC4000 circuit and tells you it works fine in unit delay simulation, but won't work when downloaded to the Xilinx chip. Given CLKIN=1MHz, T_{ckomax}=2.8ns, T_{su}=2.4ns.



12
[8 pts.] a) Complete the timing diagram (considering propagation and interconnect delays) to show why the circuit does not operate as intended. Show all clock cycles



4 6
[8 pts.] b) Circle the problem area(s) in the timing diagram, and briefly explain in a full sentence what the problem is. CLK2 arrives at least T_{cko} after CLK, thus FF1 loads the next value of Q0 rather than the present value. (By the way, interconnect delay from Q of FD to BUFGS makes CLK2 have an even larger delay).

4 6
[8 pts.] c) Modify the design above so it will work as intended. Do not add any gates or FF.

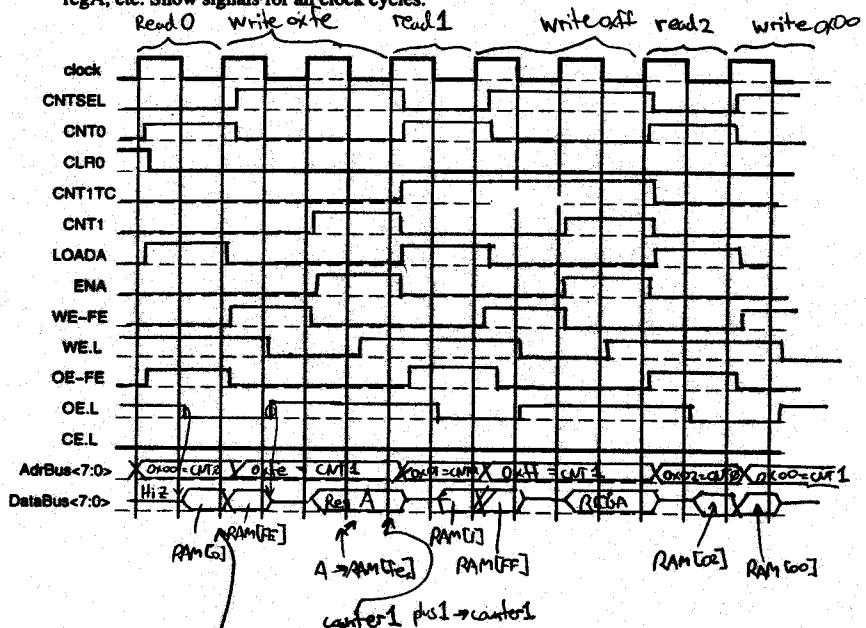
Use only CLK, and tie CE of FF1 to CLK2.

[20 pts.] 3a) For the datapath above, complete the functional timing diagram for a set of control signals that will perform a block move operation as described in the register transfer notation:

```
0 → COUNTER0
while (COUNTER1 <= 0xff)
{ RAM[COUNTER0] → A, COUNTER0+1 → COUNTER0
  A → RAM[COUNTER1]
  COUNTER1+1 → COUNTER1
}
done /* go to done state, no-op */
```

this was supposed to be <0xff, but with 9 bit counter...

The control signals are generated by a MOORE machine using rising edge-triggered FFs. COUNTER0 initially contains 0x00. COUNTER1 initially contains 0xfe. Insert idle states only as necessary, e.g., to prevent tri-state conflicts. Label address and databus contents, e.g., RAM[0], regA, etc. Show signals for all clock cycles.



RAM[0] → A
Counter0 plus 1 → counter0

read cycle [0] +6 pts
write cycle [0xfe] +8 pts
read [0x01] +2
write [0xff] +1
read [0x02] +1
write [0x00] +1
20

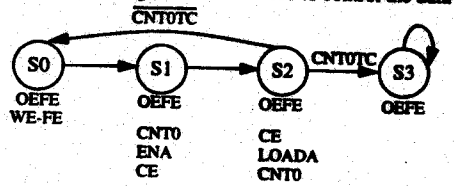
Problem 3 (cont.)

4 [4 pts.] b) Finite state machines typically generate glitches after the rising edge of the clock. Which control signal(s) generated by the FSM must be glitch free to ensure proper operation of the data path?

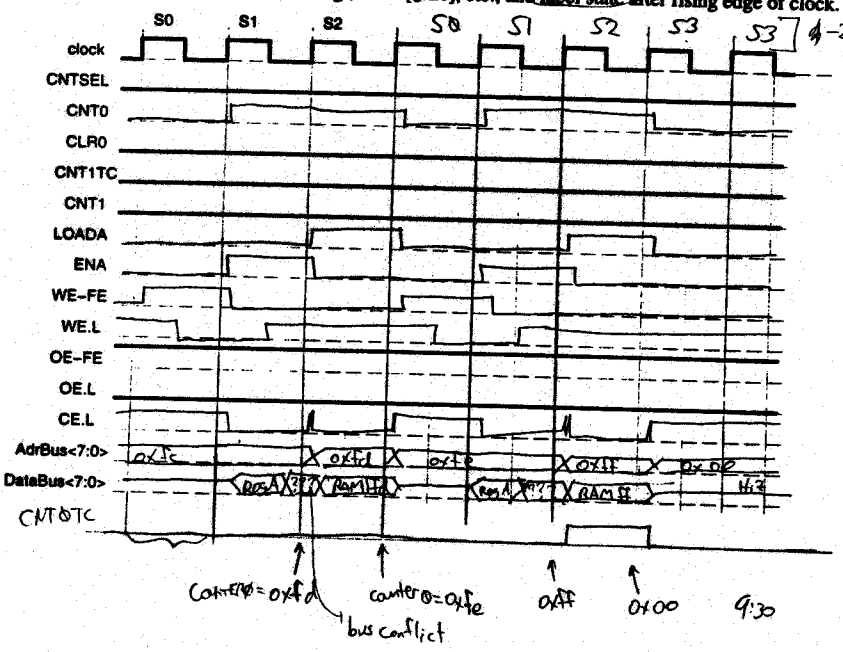
CNTSEL

-2 each wrong

5 [5 pts.] c) An FSM with state diagram below is used to control the data path shown above:



Complete the timing diagram for all clock cycles, assuming counter 0 initially contains 0xfc. Label address and databus contents, e.g., RAM[0xfc], etc., and label state after rising edge of clock.



-2 if states wrong

9.30

Problem 3 (cont.)

6 [7 pts.] **d) Using the state diagram from Prob. 3c, list in register transfer notation the operation(s) that occur during each state or at the next rising edge of the clock when in a particular state.**

1.5 S0: garbage → RAM[0xfc]

1.5 S1: reg A → RAM[0xfc], counter Q+1 → counter Q

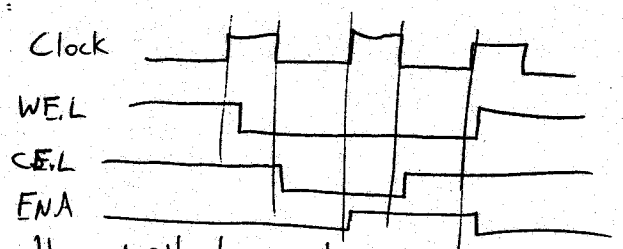
1.5 S2: counter Q+1 → counter Q, RAM[0xfd] → reg A

1.5 S3: Nop

5

[4 pts.] **e) There is at least one design flaw indicated by the timing diagram for Problem 3c. State the problem(s) and suggest a fix for each problem.**

There is a bus conflict in state 2 because both ENA and CE are asserted, i.e. the RAM is reading. If CE is not asserted in state 2, this would help, but there is still a possibility of bus conflict. The best solution would be to modify the write cycle to be:



then there won't be a bus conflict, by adding a negative edge triggered DFF to CE.L.

Problem 4 Short answers (15 points)

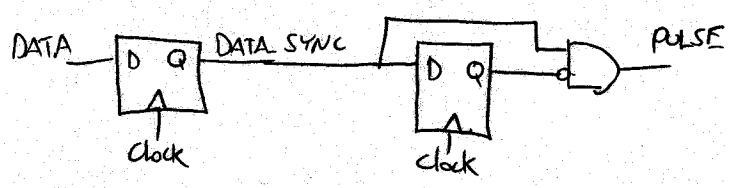
[3 pts.] **a) What connections are necessary for LED board operation for the project?**

3

PWR
GND
LEDclock
DATA
DISPLAY ENABLE

3

[3 pts.] **b) Draw a circuit that outputs a single pulse one clock period wide for every rising edge on an unsynchronized input DATA. DATA changes at less than 1/10 the clock rate.**



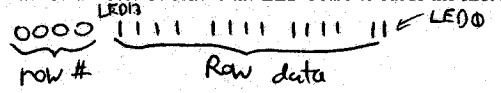
3

[3 pts.] **c) In the Xilinx FPGA, what is the difference between an OFD and a FD?**

OFD is part of IO Block and doesn't use any CLBs. OFD is connected to an OPAD, and has very small interconnect delay compared to connecting FD from a CLB to OBUF and OPAD.

3

[3 pts.] **d) What data would be sent to the LED board to cause the first row of LEDs to all be turned on?**



3

[3 pts.] **e) Why does the clock out to the LED board need to be as fast as possible?**

So that DISPenable can be asserted for as long as possible. Otherwise the display will be too dim if more time is spent sending data than displaying it.