Problem 1 (16 points)

List the ROM contents in **hexadecimal** to implement the FSM shown below. The input A is synchronized. The states are assigned numerical order, e.g., for state S4, $Q_3Q_2Q_1Q_0 = 0100_2$. (Follow normal state diagram assumptions: holding in the same state is implicit, etc.).

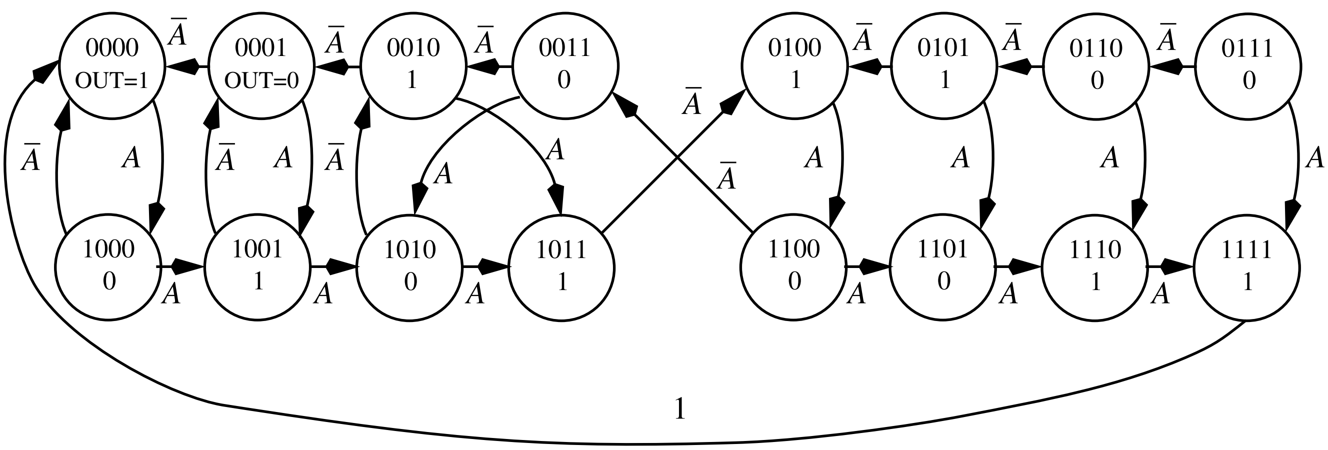
Fill in ROM contents in hexadecimal. (Binary answers will receive no credit.)

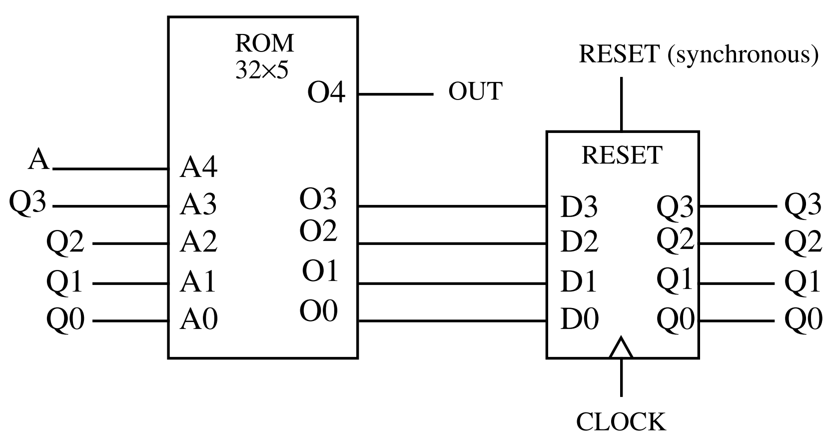
Address	**Data**
0	
1	
2	
3	
4	
5	
6	
7	

Address	**Data**
8	
9	
A	
B	
C	
D	
E	
F	

Address	**Data**
10	
11	
12	
13	
14	
15	
16	
17	

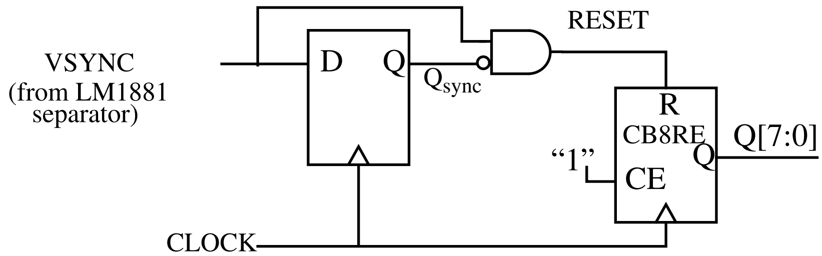
Address	**Data**
18	
19	
1A	
1B	
1C	
1D	
1E	
1F	



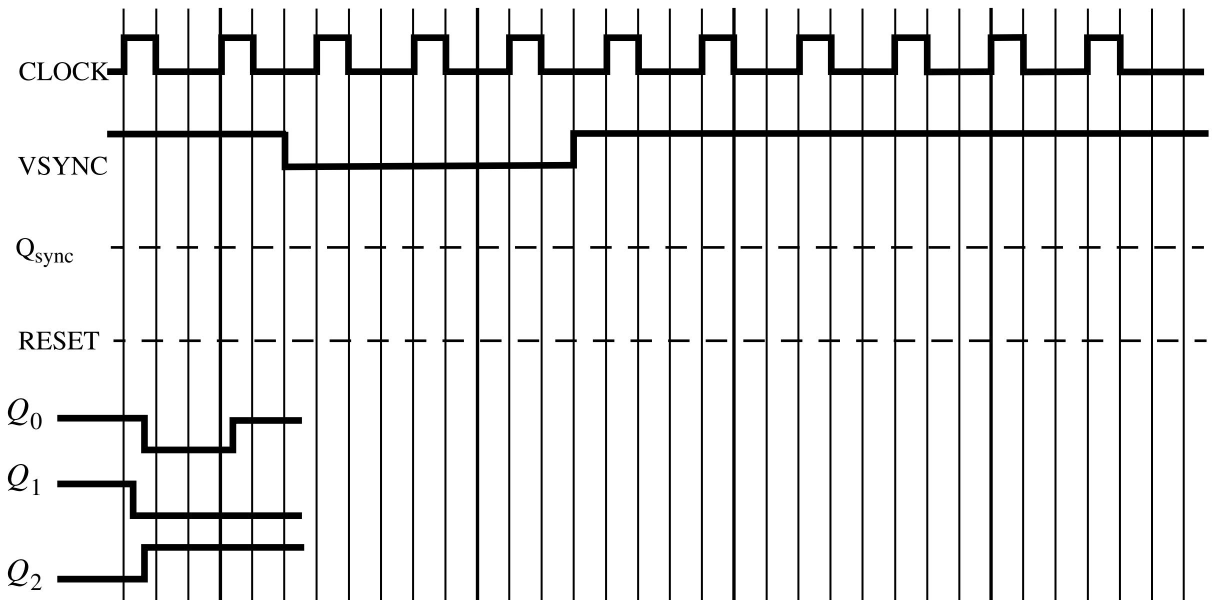


Problem 2 (20 points)

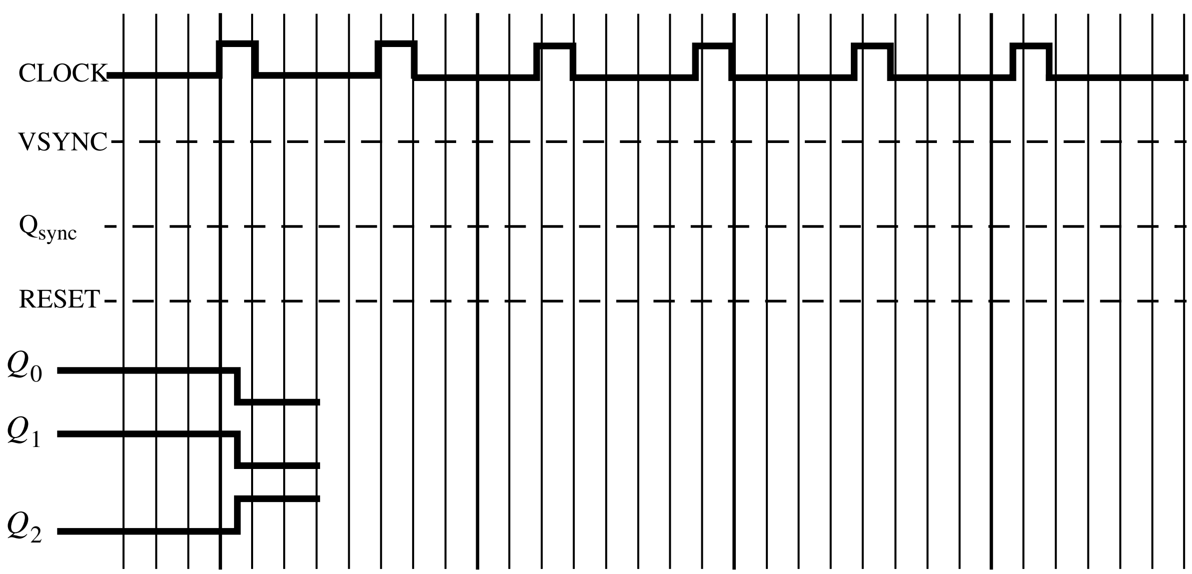
Your coworker has designed the following subsystem to reset your vertical counters in your project. (The CBBRE is an 8-bit binary counter with synchronous reset)



[7 pts.] **a)** Draw a timing diagram, assuming proper operation.



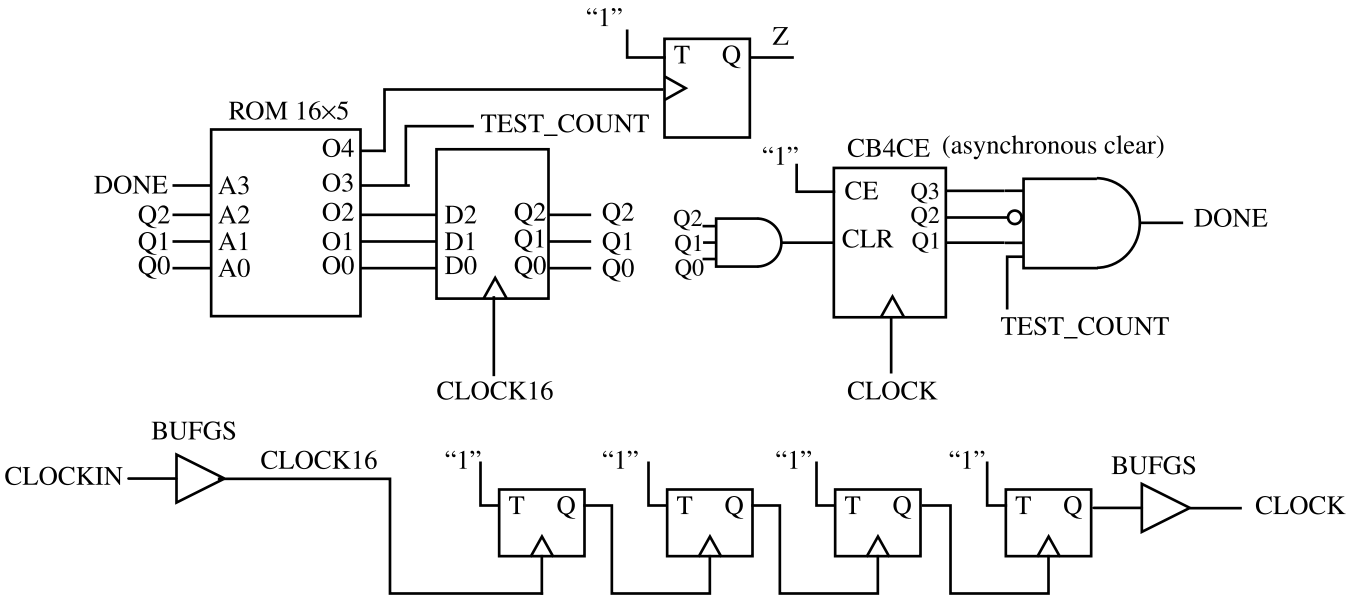
[7 pts.] **b)** You observe inconsistent operation of the circuit. Show with a timing diagram what the cause of the problem may be.



[6 pts.] **c)** Explain how the problem could be avoided.

Problem 3 (15 points)

Your coworker has been laid off and you have been given their Xilinx XC4000 design to get working. Before debugging, you look for any synchronous design rule violations. Find these violations, circle and number the problem areas on the diagram, and briefly explain the problem in a complete sentence.



1.

2.

3.

4.

5.

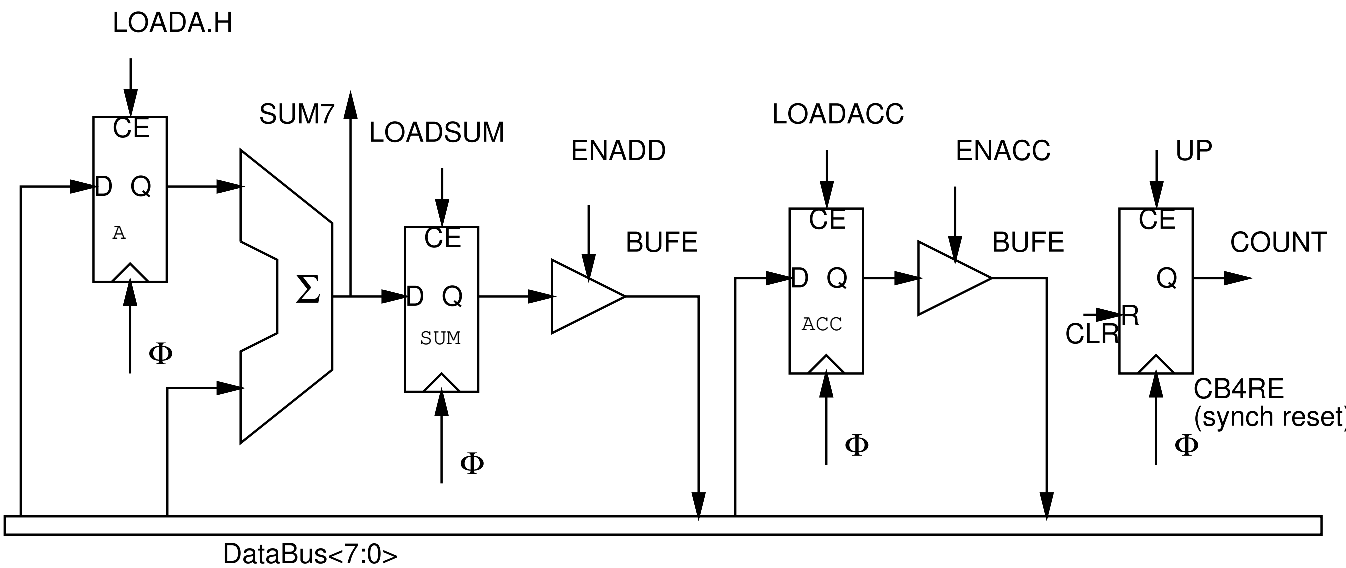
Problem 4 (30 points)

You are given the 8-bit data path below. You need to design control signals and an FSM controller that could implement an algorithm which performs a division by repeated subtraction. For example, 7 divided by 3 would be calculated by subtracting 3 from 7 twice, leaving a remainder of 1. Initially, the A register has the two's complement of the value to be subtracted (e.g., $-3 = 1111 \quad 1101_2$), and the ACC register has the numerator of the fraction (e.g., 7). SUM7 is the sign bit. In register transfer notation, the system performs the following operations:

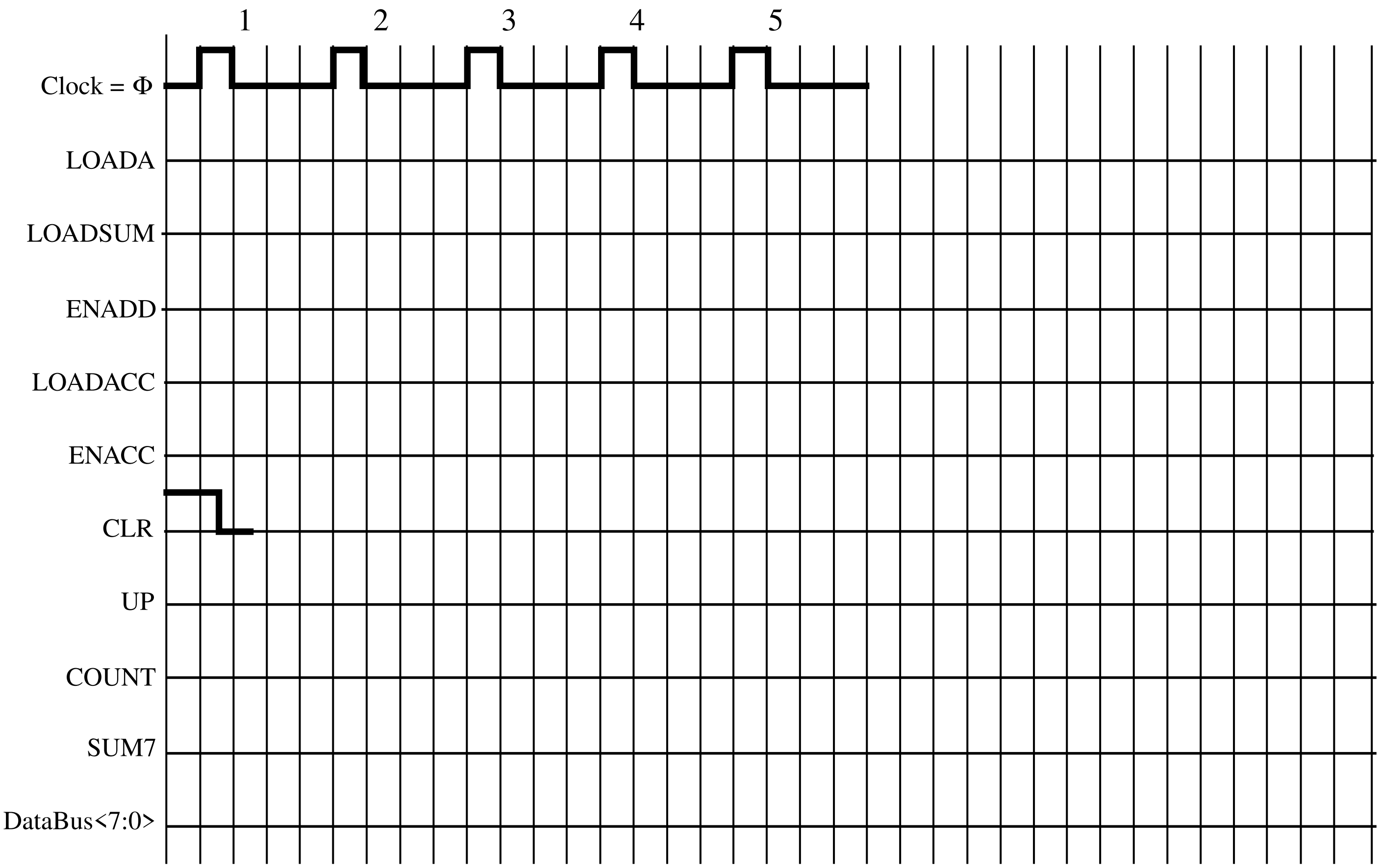
clock 1: $0 \rightarrow$ COUNT

clock 2: if (SUM7≠1) {
ACC - A $\rightarrow$ SUM
COUNT+1 $\rightarrow$ COUNT} else done

clock 3 to clock N: while (SUM7≠1) {
SUM-A $\rightarrow$ SUM
COUNT+1 $\rightarrow$ COUNT}



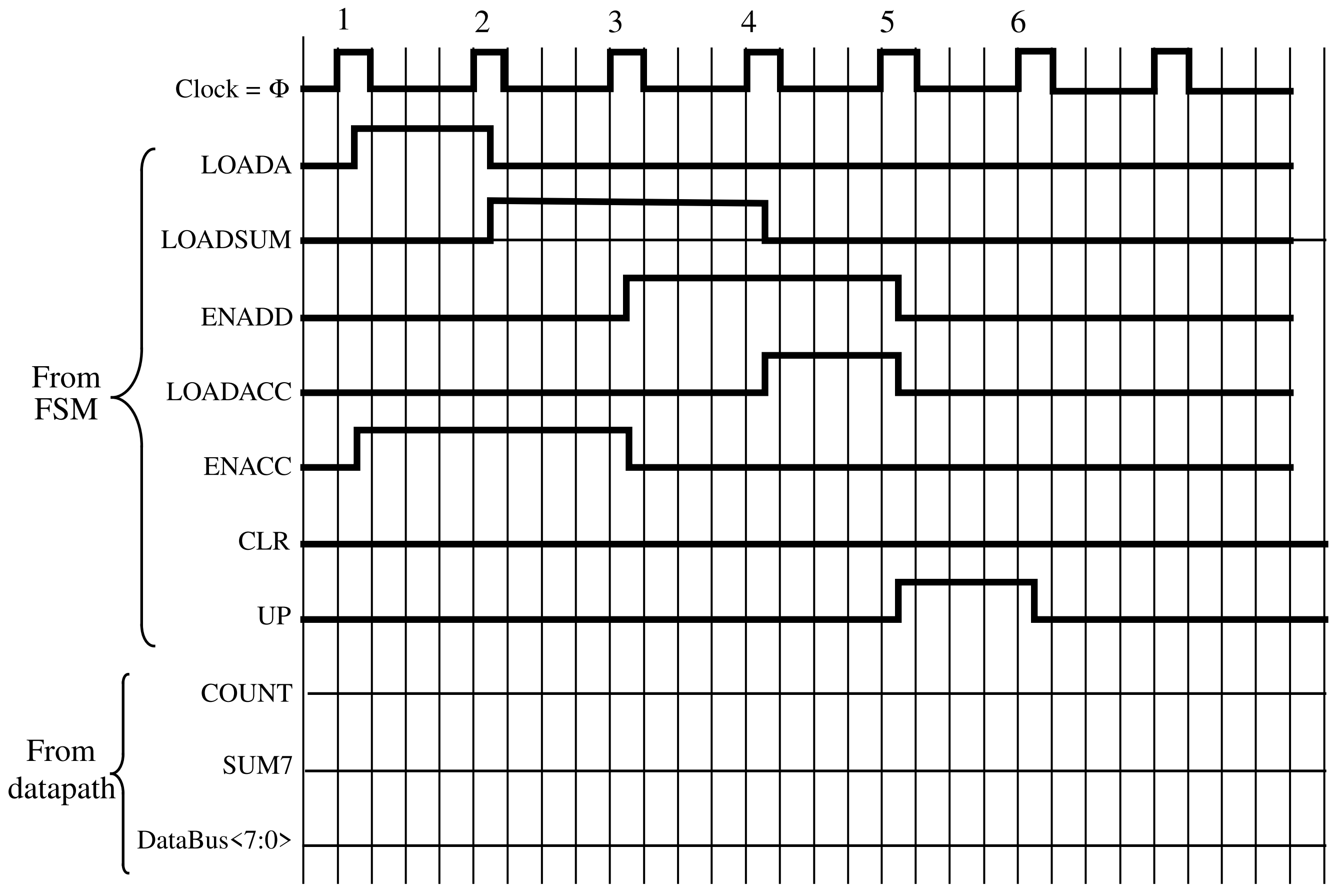
[20 pts.]4a) Complete the timing diagram for the algorithm described, assuming that the control FSM is a Mealey machine, using rising edge triggered FF. Reg $A = 1111 \quad 1101_2$ and $\mathrm{ACC} = 0000 \quad 0111_2$.



[10 pts.] 4b) Draw a state diagram for a Mealey machine which will generate the proper control signals for the data path for the division operation described. You may assume that A and ACC have already been loaded. Symbolically label outputs and transitions.

Problem 5a (9 points)

For the data path of Problem 4, a FSM generates the following control signals:



List, in register transfer notation, the operation which is occurring on the rising edge of the clock cycle.

Clock edge 1: RTN ______________________________

Clock edge 2: RTN ______________________________

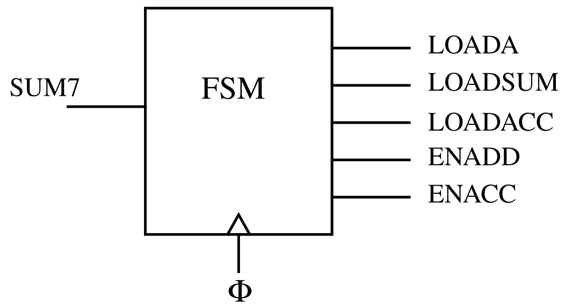
Clock edge 3: RTN ______________________________

Clock edge 4: RTN ______________________________

Clock edge 5: RTN ______________________________

Clock edge 6: RTN ______________________________

[10 pts.] 5b) A finite state machine, running on clock Φ generates control signals LOADA, LOADSUM, LOADACC, ENADD, ENACC as shown in the figure in part 5a, above. The FSM has input of the signal SUM7.



The FSM is a Moore machine. Outputs are stable t_{FSMOUT} after the rising edge of the clock. SUM7 must be stable t_{FSMIN} before the rising edge of the clock. The datapath has the following specifications:

Parameter	Specification
t_{az}	time from BUFE disable to tristate
t_{za}	time from BUFE enable to active output
t_p	propagation delay through tristate
t_{set}	setup time from CE on register to rising edge of clock
t_{cko}	delay from clock edge to output of register $< t_{\mathrm{FSMOUT}}$
t_{hold}	hold time for registers
t_{add}	propagation delay through adder

What is the minimum clock period for proper operation using the timing diagram of part 5a (in algebraic form)?

min period = ________________________________