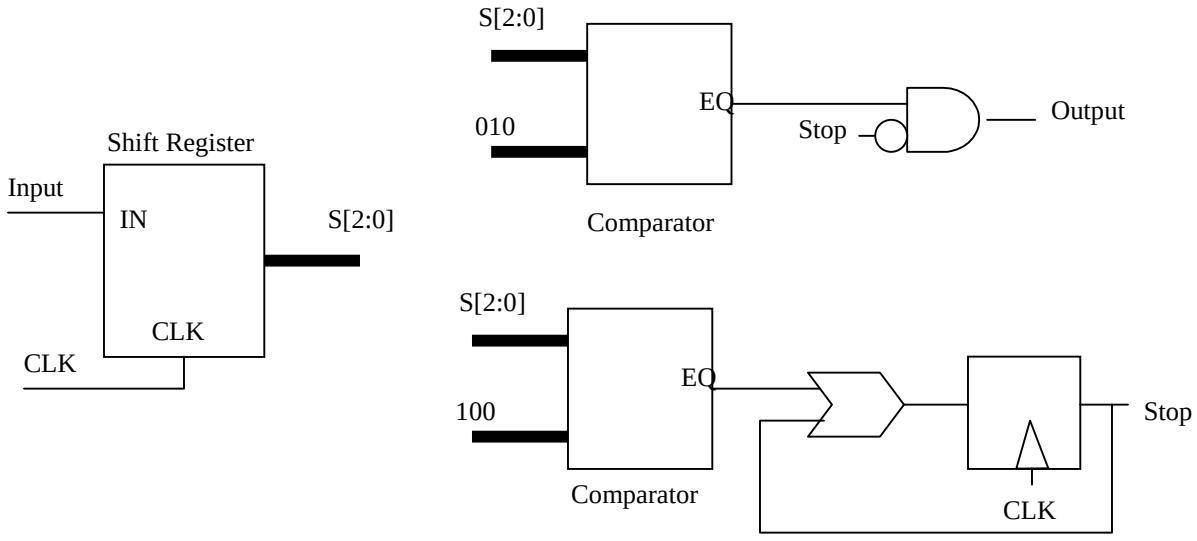
University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS 150 R. H. Katz
Fall 2000

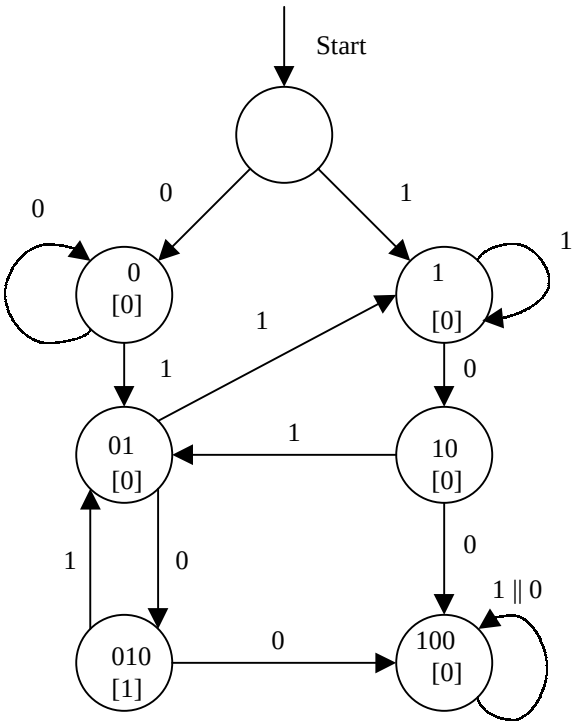
Problem Set # 8/9 (Assigned 2 November, Due 17 November)

1. Recall the sequence detector FSM described in lecture: the machine outputs a 1 whenever the input sequence …010… has been detected, as long as the string 100 has not been encountered. Implement this state machine with a trickier datapath: a shift register for the state register and a comparator to detect the critical sequences.
   (a) Design a datapath down to the block diagram level of its functional units to support this state machine. Define your register transfer operations, control signals, bus architecture, and timing behavior.

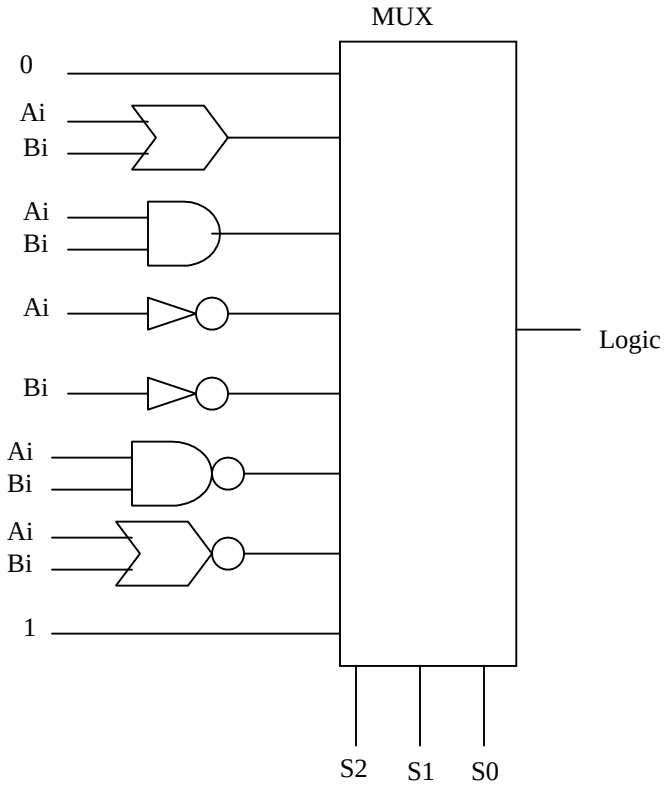


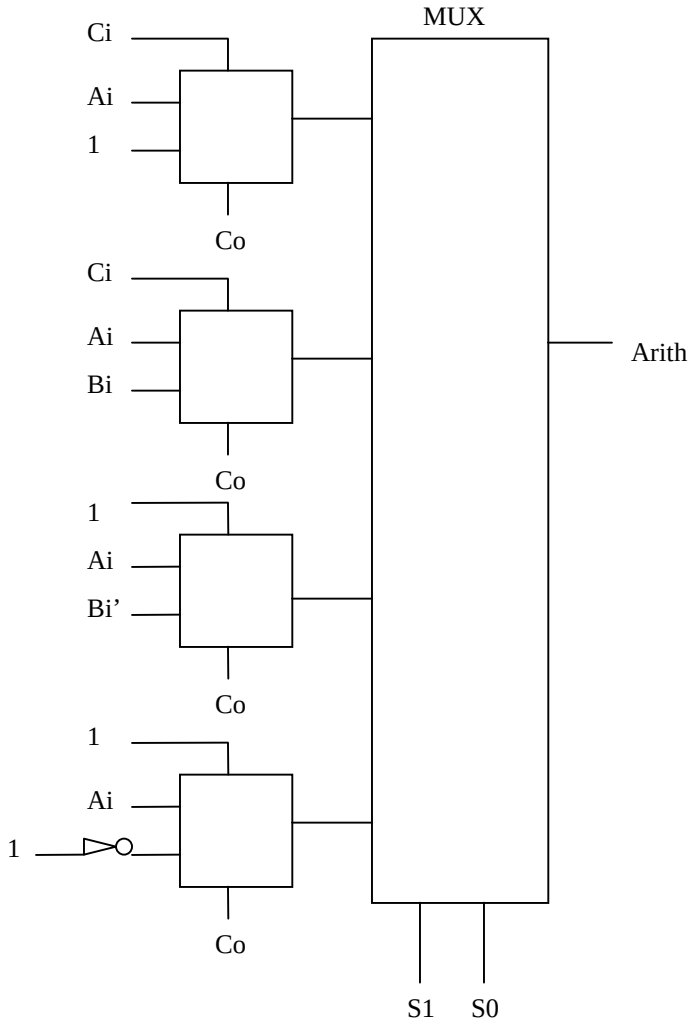
One input bit per clk cycle and a shift each clk cycle.

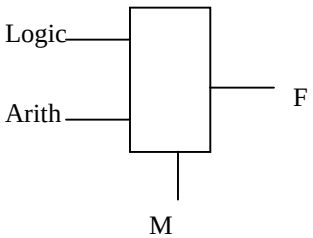
   (b) Design the state machine that implements the behavior of the specification above in terms of how it orchestrates the execution of the datapath you designed in part (a).



2. Design an Arithmetic Logic Unit to the following specification. The ALU has 4 control inputs: M, S_2, S_1, S_0, a carry-in and carry-out, and bit slice data inputs A_i and B_i (for each bit of the ALU). When M=1, the ALU is in arithmetic mode and when M=0, it is in logic mode. Develop a gate level implementation for a single bit of the ALU.




MUX
Ci
Ai
1
Co
Ci
Ai
Bi
Co
Arith
1
Ai
Bi’
Co
1
Ai
1
Co
S1
S0


Logic
Arith
F
M