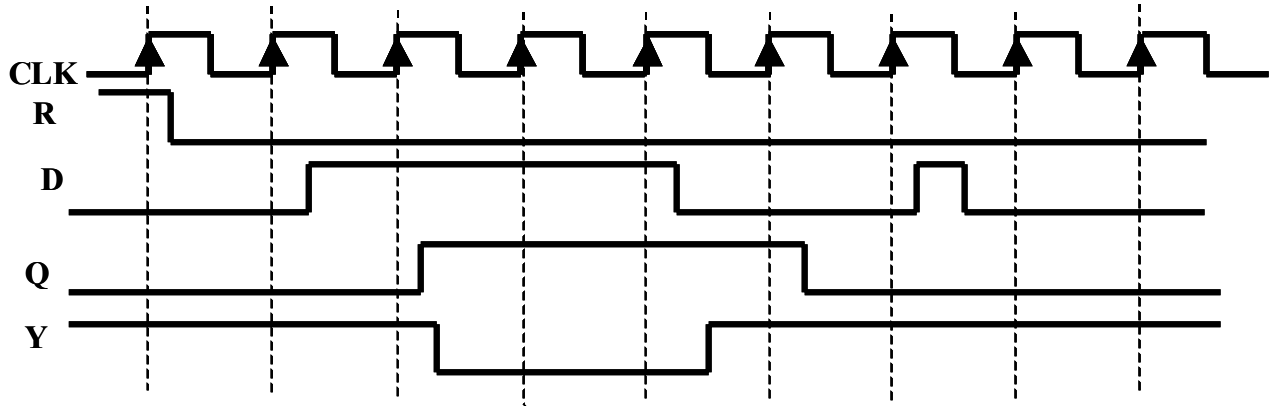
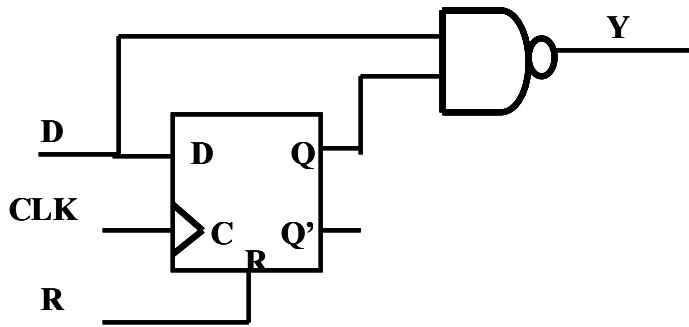


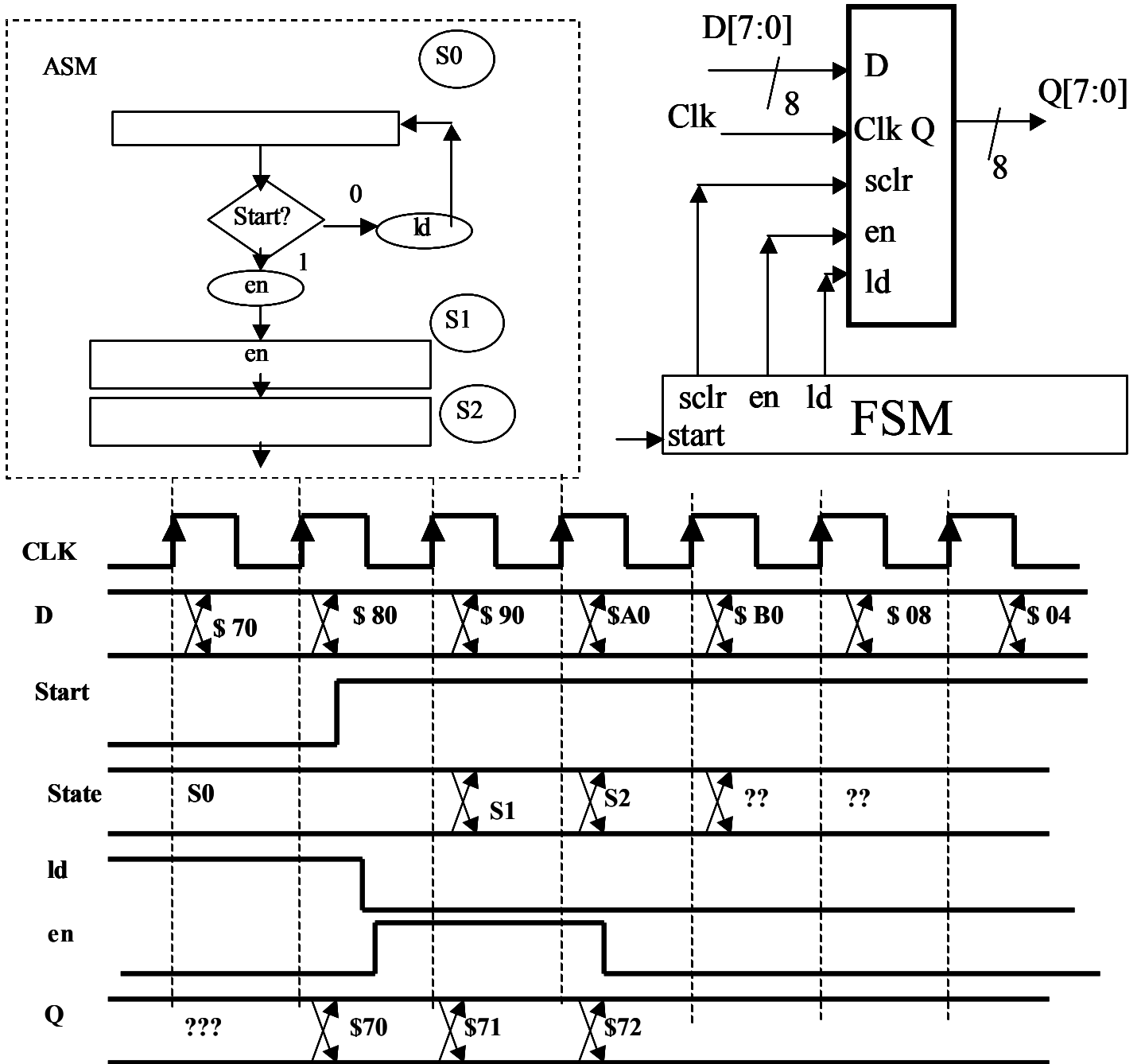
SSN: _____ (no names please)

For any partial credit, you must show your work.

1. (10 pts) On the diagram below, complete the timing diagram for the Y output for all clock cycles.



2. (15 pts) On the waveforms below, complete the waveforms for State, ld, en, and Q. The FSM is controlling an UP counter .



3. (20 pts) For the figure below:

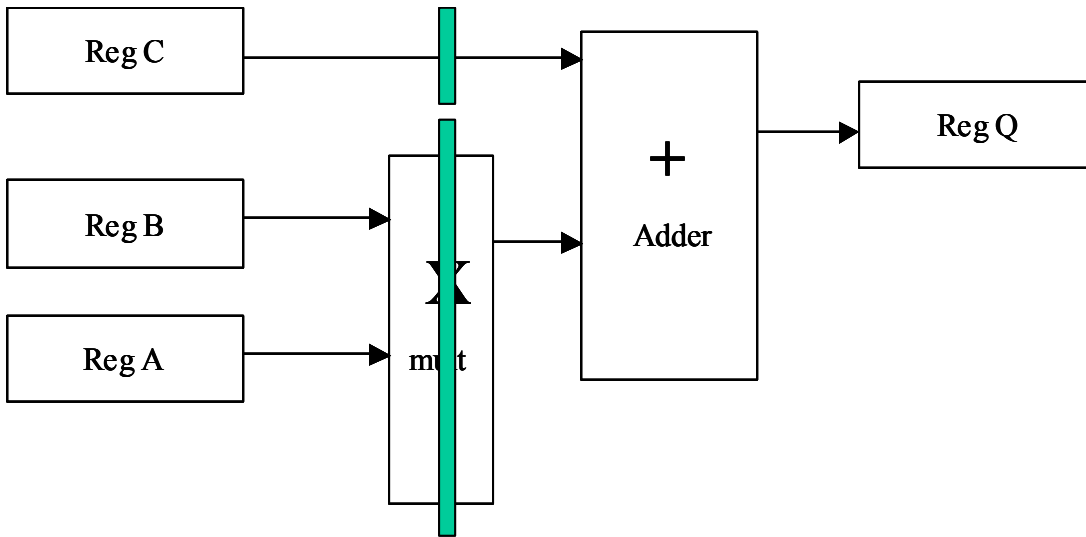
a. Give the maximum register-to-register delay. Show your work.

$$T_{cq} + \text{mult delay} + \text{add delay} + t_{su} = 3 + 18 + 7 + 1 = 29 \text{ ns}$$

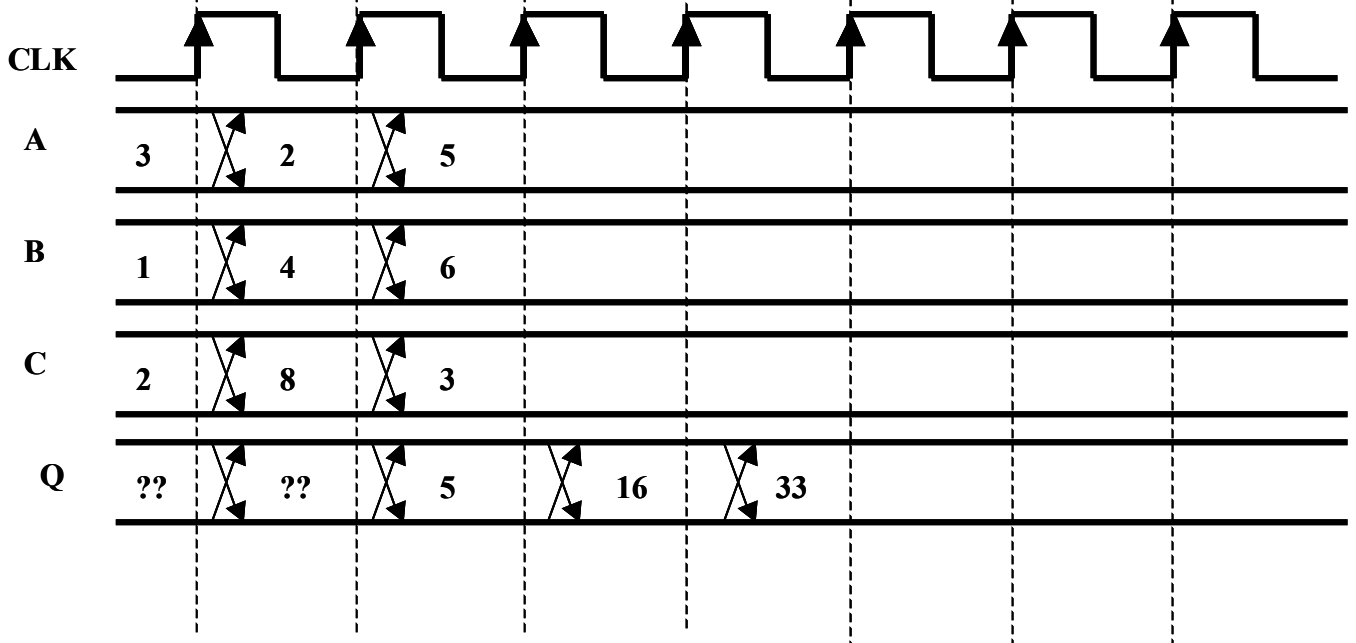
b. Modify the diagram to add one level of pipelining but still maintain the same functionality. Add the pipeline stage in the place that will improve the register-to-register delay the most. *Compute the new maximum register-to-register.* Assume that adding a pipeline registers to any functional unit (adder or multiplier) breaks the combinational delay path in the unit exactly in half.

$$T_{cq} + \frac{1}{2} \text{ mult delay} + \text{add delay} + t_{su} = 3 + 9 + 7 + 1 = 19 \text{ ns}$$

c. With the pipeline stage added, complete the 'Q' waveform shown below. Input registers change values as shown, assume Reg Q is loaded every clock cycle. All waveforms represent register outputs.

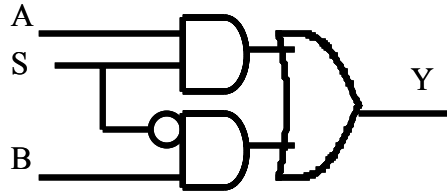


Mult Delay = 18 ns, Adder delay = 7 ns, $T_{cq} = 3$ ns, Setup time = 1 ns, Hold time = 2 ns



5. (10 pts) Draw the GATE LEVEL logic generated for the VHDL code shown below (A, B, Y are all single-bit signals). Your schematic must be composed of GATES (i.e, nands, nors, ands, ors, xors, etc...).

$Y \leq A$ when (S = '1') else B;



6. (10 pts) We looked a couple of SRAM based FPGAs from Xilinx and Altera. What was the basic mechanism for implementing a combinational logic function? Give one other feature/function that was included in both of the basic cells from Xilinx and Altera.

Both Xilinx and Altera use 4-input Look Up tables (LUTs). Both basic cells had a DFFs, also had fast carry logic.

7. (10 pts) Name two functions of a PLL (Phase Locked Loop)

*Internal/External Clock synchronization (eliminates skew from I/O buffer delay)
Clock multiplication*