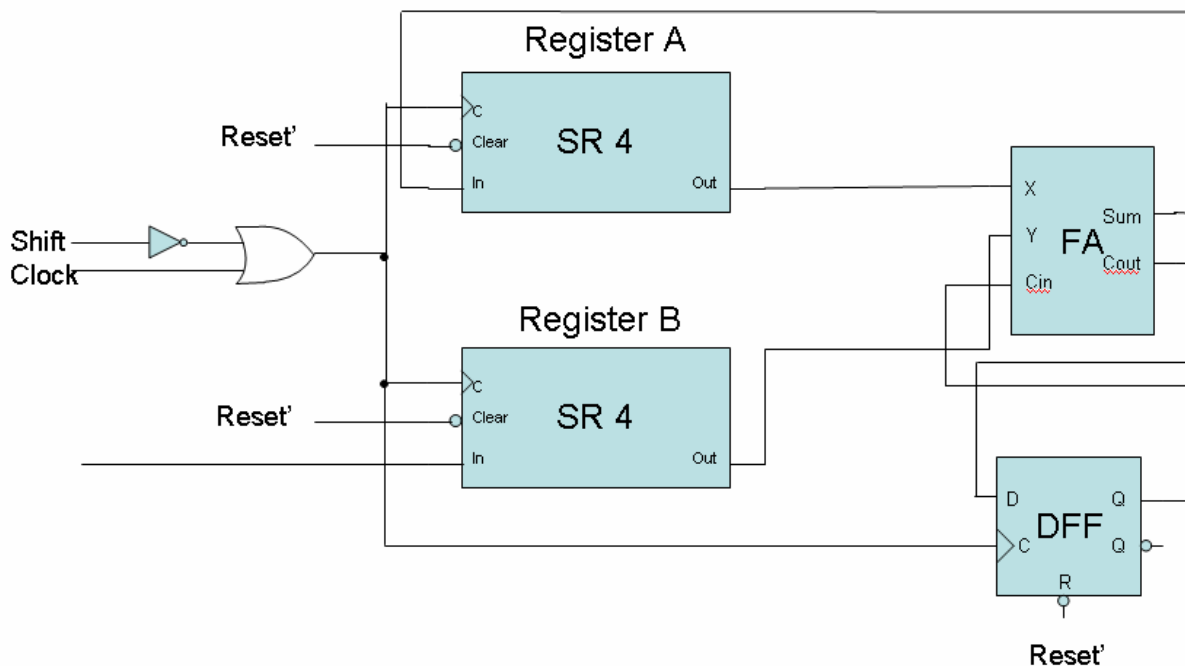


CSE 260M - Homework 10

Due November 15, 2006

1. Design an 8-bit register that has parallel load and has a synchronous reset (reset=0 resets the register). Implement your design using the Xilinx schematic tools.
2. Draw the logic diagram (using the Xilinx schematic tools) of a 4 bit register with mode selection inputs S_1 and S_0 . When $S_1S_0=00$, there is to be no change in the register contents. When $S_1S_0=01$, the value of the register is to be changed to zero. When $S_1S_0=10$ the register is to load a new value. When $S_1S_0=11$, the value is to be set to 15 (1111_2). All register changes should be synchronous to clock (i.e. don't use asynchronous reset/clear). Your logic diagram should use only edge-triggered D flip flops and simple gates (AND, OR, NOT). Do **not** gate the clock signal.
3. Design a circuit that has a 2-bit input that is a binary number and an output that goes high when the input is the largest input that has been seen since the last reset. The output should go high as soon as possible. Reset will reset the machine such that an input of 00 will be the largest input seen so far. When reset is high, however, the output should always be low. Draw a state diagram for this circuit and determine the next state and output equations.
4. The following figure is a design for a serial accumulator. This design has a serious flaw; It gates the clock! Redesign the circuit so that it does not gate the clock. Implement your design in VHDL and simulate adding the following numbers: 1, 2, 3, 4. Start by building a 4-bit register and a full adder in VHDL and then use these in your final design. (In other words, implement the block diagram.) Also be sure to reset your registers before you start entering the data. (NOTE: You might want to add a parallel output to Register A to make the result more obvious. Also think about how many times you need to clock this circuit to add four, 4-bit numbers.) **Turn in your VHDL code and simulation output clearly showing the final result in Register A. Select the *unsigned* radix for the output to show the result in decimal.**



5. Again looking at the figure in problem #4, what is the maximum frequency you could clock this circuit? Assume that the clock is **not** gated and that the registers are built as you did them. Further assume that flip-flops have a propagation delay (clock-to-output) of 5 ns, a setup time of 2 ns and a hold time of 1 ns; each logic gate has a delay of 3 ns (AND/OR/XOR are all the same) and that you only have only 2-input gates; the registers have one gate (AND gate) at the input to the flip-flops, as well (prove this to yourself!). You need to think about how the FA is built!
6. Design a counter that counts 1,2,3,5,8,13 and then repeats. Show the equations for each flip-flop (next state equations). You don't have to draw the circuit. Don't worry about reset.
7. A 4 bit twisted ring counter is a sequential circuit which produces the following sequence of output values: 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001 and then repeats. Design a circuit for a 4 bit twisted ring counter that uses four D flip flops. Draw a state transition diagram, a state table and a schematic for your circuit. Design an alternate implementation using just three flip flops and draw a state transition diagram, state table and a schematic for your circuit. If your designs are extended to implement an n bit twisted ring counter, how many flip flops are required using each of the two approaches. In what situations would you prefer the first method? In what situations would you prefer the second?