Professor Fearing EECS150/Problem Set 5 Fall 1999

Due at 10 am, Fri. Oct. 1 in box outside 218 Cory Hall

Quiz 1. 2-330 pm Tues. Oct. 5, 10 Evans (no late HW due to solution handout Fri. pm)

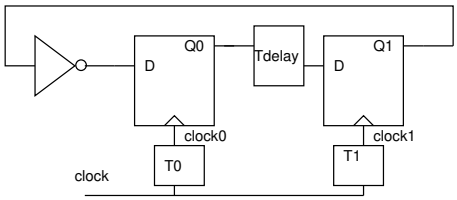
Reading Katz: 4.2

1. Glitches (20 pts)

Consider the x74_163 component from the Xilinx library, with the counter counting 0,1,2,...,14,15,0,1, ... (Also see text p. 353-356). An AND4B2 gate is used to detect state 5 ($FIVE = \overline{Q_3}Q_2\overline{Q_1}Q_0$). The delay from the output of each flip-flop in the counter to the AND4B2 gate driving FIVE depends on the layout chosen by the xilinx place and routing software. Assuming variable timing delays, show with a timing diagram all places where glitches could be generated on transitions. What would happen if the FIVE signal were used to drive the asynchronous reset input of another counter?

2. Clock Skew (30 pts)

For this problem consider the parameters of the XC4000E-2 $T_{ICK} = T_{setup} = 2.4$ ns min, $T_{hold} = T_{CKI} = 0$ ns, $T_{CKO} < 2.8$ ns, $T_{delay} = 60$ ns and inverter plus interconnect propagation 4 ns. For each question below, justify your answer with a timing diagram.



a. With delay $T_0 = T_1 = 0$ ns, what is the maximum clock speed?

b. What is the maximum $T_1 - T_0$ for proper operation?

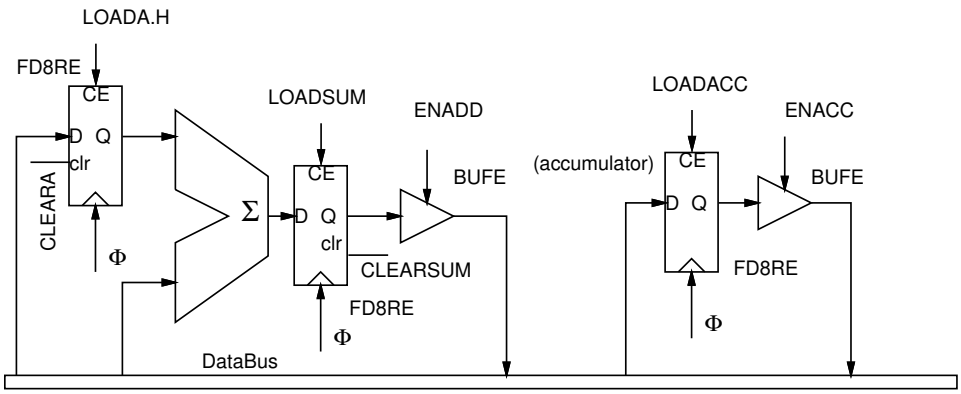
c. What is the minimum $T_1 - T_0$ for proper operation?

d. In Xilinx XC4000 designs, how can you prevent indeterminate delays on the clock line due to random routing?

3. Controller Design and Tri-state busses (50 pts)

Assume a delay of 20 ns from E being deasserted to the buffer being tri-stated, and a delay of 8 ns from E being asserted to a buffer being active. The data path is 8 bits wide.

a. Determine a set of control signals which will exchange the contents of ACC and SUM. (Contents of register A can be discarded) Draw a functional timing diagram for the control and clock signals LOADA, LOADSUM, ENADD, LOADACC, ENACC, CLEARA, CLEARSUM, and Φ as well as the data bus.

b. Determine a set of control signals which will perform the operation `ACC = 3 * ACC + 2 * SUM;`, and draw a functional timing diagram. Make sure that there are no bus conflicts, and that you allow reasonable setup and hold times for all devices.

c. Design a controller circuit, which given a START signal will perform the `ACC = 3 * ACC + 2 * SUM;` operation then assert DONE until the next START signal is input. (Hint: use counter plus decoder plus appropriate logic. ACC and SUM are already initialized.) Show state and logic diagrams for controller.