Professor Fearing EECS150/Problem Set 7 Fall 1998

Due at 10 am, Fri. Oct. 16 in box outside 218 Cory Hall

Reading Katz: 4.2.5, 7.6, 10.1.1

(This problem set may be done individually or by a group of two people maximum. A group of two should hand in a single copy of the solution with both names affixed).

1. Termination and Reflections (20 pts)

A cable of length 2 m with propagation velocity of $2 \times 10^8 ms^{-1}$ and impedance $Z_o = 100$ ohms has a driver at the transmitting end with source impedance Z_S of 25 ohms, and a load impedance Z_L of 10^5 ohms. Initally, the bus is at 0 V. At time $t = 0ns$ a pulse wave of amplitude 4.0V starts propagating from transmitter to receiver.

a. Sketch the voltage along the bus at 3, 13, 23, 33 and 53 ns.

b. Sketch the voltage versus time at the receiver end of the wire for the first 60 ns.

c. If the load was an IPAD/IBUF combination on a Xilinx 4005E, and the input threshold were 1.2 V, sketch the digital output of the IBUF. Is there a glitch? If so, explain why.

2. ROM Based FSM Design (20 pts)

Design a Moore FSM that counts nickels (N), Dimes (D) and Quarters (Q) and will assert an output OPEN for one clock cycle when at least $0.35 have been inserted. After global Reset or OPEN, the system returns to a waiting state. The inputs N.H, D.H, and Q.H are synchronized to the clock.

a. Draw a state diagram and state table, and design the FSM using FDR type D flip flops in a ROM based FSM. (Hint: this problem takes less than 5 flip flops). Make sure your FSM is self-starting. Both NSD and OD should be in the same ROM. List ROM contents in Hexadecimal.

b. What size ROM is needed? What fraction of the XC4005E would be used up for this ROM size?

3. Clock Design (20 pts)

You are given a 16 MHz clock signal. Generate a glitch free clock signal with average frequency of (4/13) 16 MHz $\approx$ 4.92 MHz. Estimate CLB cost of your design.

4. Optical Flow Algorithm Design (40 pts)

In this problem, you will prototype an algorithm you might decide to use in the final project. (It is much simpler to develop and debug this algorithm using a high level language without any of the timing, space, and real-time debugging limitations of the FPGA.) A set of test images have been captured using the same camera and hardware you will be using in the project and are accessible from the class homepage. The camera moves slightly between each frame. Each frame is represented as an array of `unsigned char image0[NROWS][NCOLS]` . Your algorithm should estimate the horizontal velocity of the image (in pixels/frame) at 8 locations on a row: [100][20], [100][40], ... [100][160]. While there is a very large variety of possible algorithms, you should try something simple, which could be implemented in the 4005 FPGA.

a. Explain the algorithm you want to try, e.g. mathematics and pseudo-code.

b. Show C-code and results for the test images at the 8 locations.

c. Outline how your algorithm might be realized in Xilinx. In particular, how much math computation do you need, and how much memory is used. Which steps in the algorithm will need to be rethought as they will be too complicated to implement? (Hint: floating point is not that easy to fit in Xilinx).

If you come up with a particularly clever algorithm, you may want to get credit for it ...