Professor Fearing EECS150/Problem Set 7 Fall 1999

Due at 10 am, Fri. Oct. 22 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. ROM Based FSM Design (20 pts)

Design a Mealey FSM that counts nickels (N), Dimes (D) and Quarters (Q) and will assert an output OPEN for one clock cycle when at least $0.40 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?

2. Counter Design (30 pts)

Design a counter sub-system using two X74-163 4 bit loadable synchronous counters and a simple control FSM which has the following behavior. Counter and FSM are driven by an 11.2 MHz clock. On a falling edge on input line START.H, the counter counts 231 steps in the following sequence $220_{TEN}, 221, ..., 255, 0, 1, ...194, 195$, then halts until the next falling edge on START.H. Each step is 267 ns apart (3 clock cycles). Draw a schematic of the counters and FSM, state diagram for the control FSM, and functional timing diagram which shows six 11.2 MHz clock cycles each at the startup and end of the counting cycle.

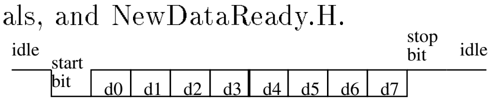
3. Asynchronous Serial Receiver Design (50 pts)

The universal asynchronous receiver transmitter or UART is at the heart of many serial communications connections, such as the serial port on an IBM PC. A transmission character is composed of an 8 bit information byte with a start bit before the first data bit, and a stop bit after the last data bit as shown in the figure below. The line may or may not be idle (at the 1 level) in between characters, with a start bit directly following a stop bit.

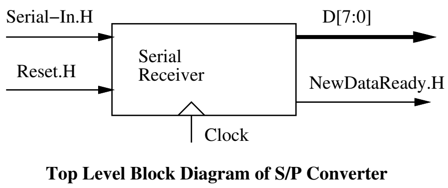
Design a 9600 baud 8 bit serial-to-parallel converter which has inputs of SerialIn.H and a synchronous RESET.H. The clock is 1.000 MHz and drives all clocked components. The outputs are the 8 data bits D0...D7, and NewDataReady.H. NewDataReady.H can be asserted until the next start bit is read.

The method for sampling the serial data is as follows: wait for SerialIn.H to go low (signalling a start bit), then wait four clock periods and check that the data line is still low. (Hint: preload a counter....) By shifting every 8 clock periods thereafter, you will sample each serial bit roughly in the middle of a bit. All sequential devices must be driven from the same clock, but you can use the CE input on SR4RLE to enable shifting. (Hint: if the shift register is preset to all ones before shifting, then the start bit may be used to indicate that all ten bits (start + 8 data + stop) have been shifted in.)

Show a detailed block diagram of the data path (to the level of Xilinx library components), a state diagram for the FSM controller, and a functional timing diagram showing clock, SerialIn.h, other control signals, and NewDataReady.H.



Serial Transmission Format



Top Level Block Diagram of S/P Converter