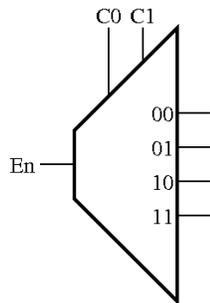


COSC 065 - Hardware Systems
Homework #2 - Sequential Logic

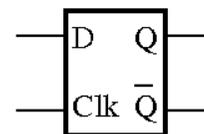
Due: 2008 September 19, beginning of lecture

This assignment is to be completed individually.

1. Lab question 1: Give a complete schematic diagram for building an XOR (exclusive OR) gate entirely from 74HC00 NAND gates. Include power, ground, switches, LEDs for both the input and output signals, and pin numbers for the logic gates. Use inverters to turn the LED's on when displaying a "1", as in the Lab #1 handout.
2. Lab question 2: Give a truth table and gate level diagram for building a *half adder* out of 74HC00 NAND gates. The half adder takes two inputs (x and y ,) and calculates two outputs (C , for "carry", and S , for "sum",) that are the result of adding the two input bits. You may omit the switch and LED circuits from your schematic, but assign pin numbers to the gates so that you can more quickly build this circuit in lab.
3. A *register* is two or more *D Flip-flops* connected to the same clock (enable) input. A *shift register* is an n -bit register that shifts its stored data by one bit position for each cycle of the clock. Shift registers can be used to perform multiplication, division, and serial-to-parallel conversion, among many other tasks. Show how to wire up a 4-bit shift register using D flip-flops.
4. In lecture, we constructed a ripple counter using a series of T flip-flops, each one with its output running into the input of the next in line. This circuit is problematic in practice, because when large counters roll over, it takes measurable time for the transition to propagate (ripple) all the way down to the last T flip-flop in line. Show how you can build an equivalent 4-bit counter out of D flip-flops, with all of the D flip-flops connected to the same clock input. (This is called a "*synchronous counter*".)
5. In class and in the textbook, we briefly examined an integrated circuit called a "*demultiplexor*", or "*demux*", which takes n control inputs to select exactly one one of its 2^n outputs. In practice, demultiplexor chips have an additional input pin, called "*enable*," (En). When *enable* is set to true, the demultiplexor operates as normally. When *enable* is set to false, none of the outputs is true, regardless of the status of the three control inputs. The presence of this enable input allows demultiplexors to be *cascaded* into larger combinations. Using only 2-to-4 demuxes with an enable input, show how you can construct a larger, 4-to-16 demultiplexing circuit. Use the 2-4 demux below.



2-4 Demux:



D flip-flop: