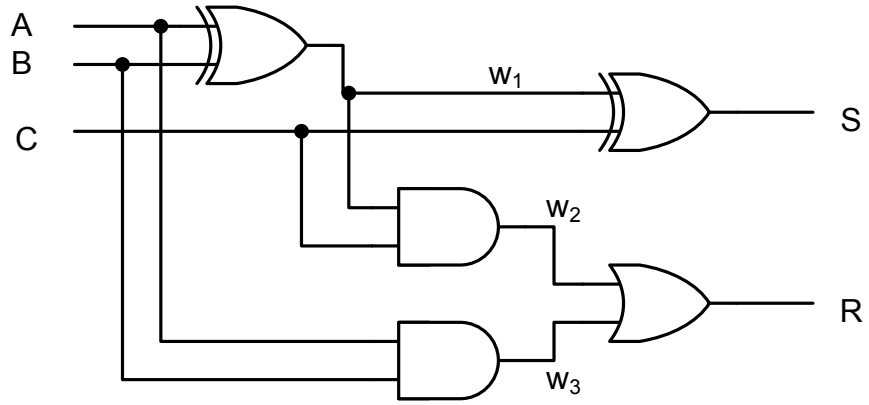
University of Maryland Baltimore County
Department of Computer Science and Electrical Engineering

CMPE 212L, Principles of Digital Design Laboratory

Experiment #6 Friday 3/04/2016

Objective

For complex logic circuits it is not easy to connect ICs in the breadboard and verify its functionality and it becomes tedious for more complex circuits. For these circuits, simulation tools would be the most appropriate choice. Hardware Description Languages (HDL) are used as a means for specifying the circuit. In this lab, you will be experiencing the using of HDL in validating your design. Verilog is to be used to design the following circuit and to verify its truth table using a test bench. Refer to the lab discussion slide for examples.



In the above figure, *A*, *B*, and *C* are 1-bit input ports while *S* and *R* are 1-bit output ports. The symbols w_1, w_2, w_3 refer to the interconnect wires that run between 2 gates. You should declare them as 'wire' in your Verilog code. The circuit should be written purely in 'Structural Model' and DO NOT use behavioral model. The code should be verified using a test bench covering all the test cases. Try to use "for loops" to cover the test cases, again refer to the discussion slides.

Verilog has inbuilt logic gates and their port mapping are defined as below.

Gate	2-input port map	3-input port map
AND	and(out,in1,in2)	and(out,in1,in2,in3)
OR	or(out,in1,in2)	or(out,in1,in2,in3)
XOR	xor(out,in1,in2)	xor(out,in1,in2,in3)
More gates for future use		
NAND	nand(out,in1,in2)	nand(out,in1,in2,in3)
NOR	nor(out,in1,in2)	nor(out,in1,in2,in3)
XNOR	xnor(out,in1,in2)	xnor(out,in1,in2,in3)
NOT	not(out,in) - (one input only)	

```
module sampleTestbench();
  reg [2:0] switches;      // 3 inputs for A, B, and C
  wire [1:0] out;          // 2 outputs for S and R

  // instantiate the module for lab6 (you'll have to write a module for lab6)
  lab6 labby(out[1],out[0],switches[2],switches[1],switches[0]);

  initial
   begin
        switches = 3'b111;
        $display("\t\t    switches=ABC, out\n");
        #45 $finish;
   end

  always
   begin
        #5 switches = switches + 3'b001;
        $monitor($time, " switches=%b, out=%b", switches, out);
        // $monitor displays the time and the string everytime
        // 'switches' or 'out' changes value.

   end
endmodule
```