

CSE370 Fall '99
Assignment 5
Distributed: 11/1/99
Due: 11/8/99

Reading:

Katz, Chapter 5 through 5.3.1, the rest of Chapter 5 is optional
Verilog Introduction: <http://www.eg.bucknell.edu/~cs320/1995-fall/verilog-manual.html>
Recommend sections 2.2-2.7
Design Works Howto: <http://...370/98au/admin/Tools/DesignWorks/Verilog.html>

Top-Down Design of a Bi-directional Collision Warning System.

In top-down design, it is not unusual to design and verify a system before the detailed design of some of its parts is complete. In this assignment, we will approximate that process for the Bi-directional Train Warning System of midterm infamy. The following steps constitute the basic top-down design process:

- A. Create a behavioral (high level language) model of the entire system based on the specification below. We are going to skip this step for now (**nothing to turn in**).
- B. Decide how to simplify the problem by breaking it into manageable sub-problems. In this case, please assume that you have decided to implement the bi-directional subsystem using the unidirectional warning system. The specification for the uni-directional warning system is the same as for the bi-directional warning system, except that we assume Train A is always further west than Train Z (**nothing to turn in yet**).
- C. Create and verify a behavioral model of the unidirectional subsystem for use in design and simulation of the complete bi-directional warning system. Most of the Verilog model is included below. Please complete the model and link it to a device called UniWarn in your library. Follow the attached instructions for doing this. Test your UniWarn device to your satisfaction, and turn in the completed Verilog model.

```
module UniWarn(a, b, c, x, y, z, warn);
    // I/O Port Declarations
    input a;
    input b;
    input c;
    input x;
    input y;
    input z;
    output warn;
    // internal declarations
    reg warn;

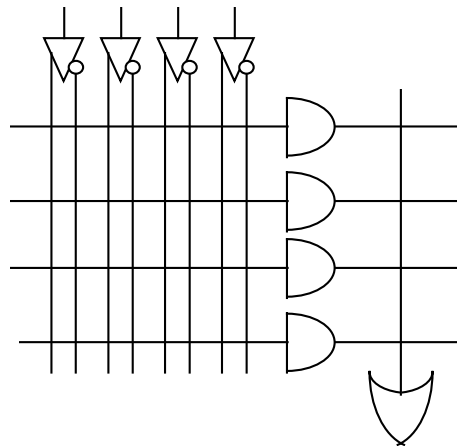
    // behavioral model for UniWarn system
    always @(a or b or c or x or y or z) begin
        if (a && ~x) warn = 0;
        if (~a && x) warn = 1;
        if (~a && ~x) warn = ({b,c} > {y,z});
        if (a && x) warn = _____; // please fill in
        // note that {b,c} is regarded as a 2-bit UNSIGNED
        // number by Verilog
    end
endmodule
```

D. Design the bi-directional system using UniWarn device, and any other sub-systems that your think will be useful. You may use any parts from the *primgate* or *primlog* libraries to implement the system. Turn in the DW schematic for the system, showing all levels of hierarchy as needed, along with a description of the design concept, and an explanation for each element of the design. Also turn in the DW spreadsheet showing the inputs and **verified expected** outputs for each of the following cases:

Train A		Train Z	
Speed	Pos	Speed	Pos
-3	0	+3	1
-3	1	+3	0
0	0	0	0
-2	2	-1	0
1	2	0	3
1	3	0	2
1	2	1	0

E. On paper, showing your work, complete the detailed design of the UniWarn sub-system using only the following hardware (hint: take a look at the Verilog model):

- One 4:1 multiplexor
- The PLA shown in the figure.



Appendix

Specification for the Bi-directional Collision Warning System

Train A and Train Z are on the same track. The velocity of each train is represented by a 3-bit 2's complement number from -3 to +3. The A and Z velocity inputs are $\{a,b,c\}$ and $\{x,y,z\}$ respectively. If the velocity is positive the train is traveling east, otherwise the train is traveling west. The position of each train is represented by a 2-bit unsigned number where 0 is furthest west and 3 is furthest east. The A and Z position inputs are $\{p,q\}$ and $\{m,n\}$. The system issues a warning (W) whenever the current velocities and positions of train A and train Z will result in a collision assuming no future velocity changes. There is no need to issue a warning if the trains are going the same speed in the same direction. However, a warning should be issued if the two trains are in the same position.

