Solution for Assignment 5

CSE 370

Part c)

({b,c} > {y,z});

!!. Yes the answer is exactly the same as one in the previous case!!. While a truth table can easily be made to prove that yes indeed the expression is true, it is more important to have the right "intuition" too. First of all, lets see why is it **{b,c} > {y,z}** when **a=x=0**. Basically we want the function **abc>xyz** WITH sign. So when a=x=0 it reduces to the above expression. When a=x=1 then we need want |bc| < |yz|. However in 2-complement notation **101<110 (-3 <-2)** even though as unsigned number 01<10 and we are dealing with negative numbers. So when negative numbers are involved there is another "negation" introduced w.r.t unsigned number namely that even though unsigned 01 < unsigned 10, |101| > |110| (|-3| > |-2|). So this double negation in fact gives the condition for |bc| < |yz| when abc<0,xyz<0 zero as **unsigned ({b,c} > {y,z}).**

Part d)

See the attached Design Works File. There are two main blocks used. One just compares to see whether **abc=xyz** and other takes **pq** and **mn** as inputs and has two outputs. Also consider the 3-input NOR in the design as a 3-input AND gate with ALL the inputs negated (A?B?C? =(A+B+C)?) to understand it better. Remember while using the Not of Uniwarn, be careful about the case when the two speeds are equal because in that case the Uniwarn will output zero and hence NOT of that is **1.** But even if Train A is further **east** of the Train Z, there is no collision if there speeds are same, and the output of the Bi-directional Warning should be zero.

Part e)

The Truth Table for **unsigned ({b,c} > {y,z})** is as follows

b	c	y	z	bc>yz
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Minimizing it we get

unsigned ({b,c} > {y,z}) = by?+cy?z?+bcz?

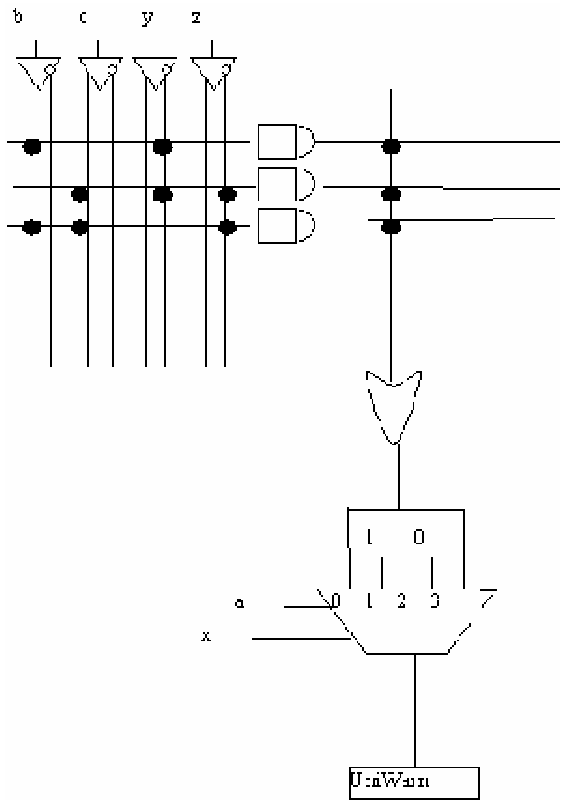
This is implemented in the PLA below. Now using this and considering the four cases in part c) we get the following implementation using a multiplexer. This is just a simple way of expressing that if-then logic!!

If a=0,x=0 (which is 0 pin in the mux) then output = **unsigned bc>yz**

If a=0,x=1 (which is 1 pin in the mux) then output = **1**

If a=1,x=0 (which is 2 pin in the mux) then output = **0**

If a=1,x=1 (which is 3 pin in the mux) then output = **unsigned bc>yz**



Other implementations

One Other implementation to the UniWarn problem could be using the two Absolute Module in the MidTerm and then using logic similar to above to solve it.

That is, first you find the absolute of the two speeds **abc** and **xyz.** In case the two speeds have different sign the case is same as above. In case both speeds are positive the above logic for **bc>yz** is used so even the third case remains same. However when both the speeds are **negative** then the logic does not remains the same as now we have the absolute values so **unsigned bc <yz** needs to computed now. This means we need to implement **two** K-maps.

Similarly there could another implementation to Bi-directional Warn system using UniWarn. Instead of using the the negation of output of UniWarn (See Designwork file) one could rather multiplex at the **input** of Uniwarn by giving it **abc & xyz** as input when Train A is further west and **xyz & abc** as input when Train Z is further west !!. In that case we don?t need to worry about the equal speeds case as above (why ??).