INF2270, another exercise in combinational logic

P. Häfliger

February 16, 2010

Abstract

In this exercise you will have a closer look at a part of the mandatory exercise in task 1 and another combinational logic circuit in task 2.

Task 1

Build and simulate a 4 bit decoder. In the mandatory exercise you will need this circuit to decode the number you want to display on the 7-segment LED. You can use the 16-bit 'one-hot' output of the decoder to selectively activate the appropriate segments of the LED display.

Note that in designs that are easiest to conceive as sequence of operations (like, decoder → 7-segement encoder) it is possible to optimize the combinational logic by combining those two steps into one. However, this is not recomended in most cases: it does not make the resulting circuit more understandable and is hard to do. Only if one desperately needs to place the result into a very small area or a very small programmable logic device can the effort be justified. For these exercises and the mandatory exercise, however, this is neither expected nor recommended.

In order to test your circuit you may design it with the ISE design suite (see the mandatory exercise oblig1.pdf for an introduction to the tool). For your convenience, a skeleton setup for your design can be found under

```
~inf2270/programmer/Decoder
```

.

Task 2

Design and simulat a comparator for two 4 bit two's complement input numbers a and b. There three output bits. One should be high if a>b another if a=b and the last if a<b.

Hint:

1. Do not optimize this for specifiacally 4 bits. Instead compose your solution of 4 identical cells (possibly with the exception of the most significant bit

(MSB) for two's complement numbers), such that you can easily extend your solution to an arbitrary number of bits.

2. Try to do this for unsigned binaries first and then think what you will have to change concerning the first 'sign' bit for two's complement.

In order to test your circuit you may design it with the ISE design suite (see the mandatory exercise oblig1.pdf for an introduction to the tool). For your convenience, a skeleton setup for your design can be found under

```
~inf2270/programmer/Comparator
```

.