

Robin Andersson
961102-5439

D0011E

Lab 1

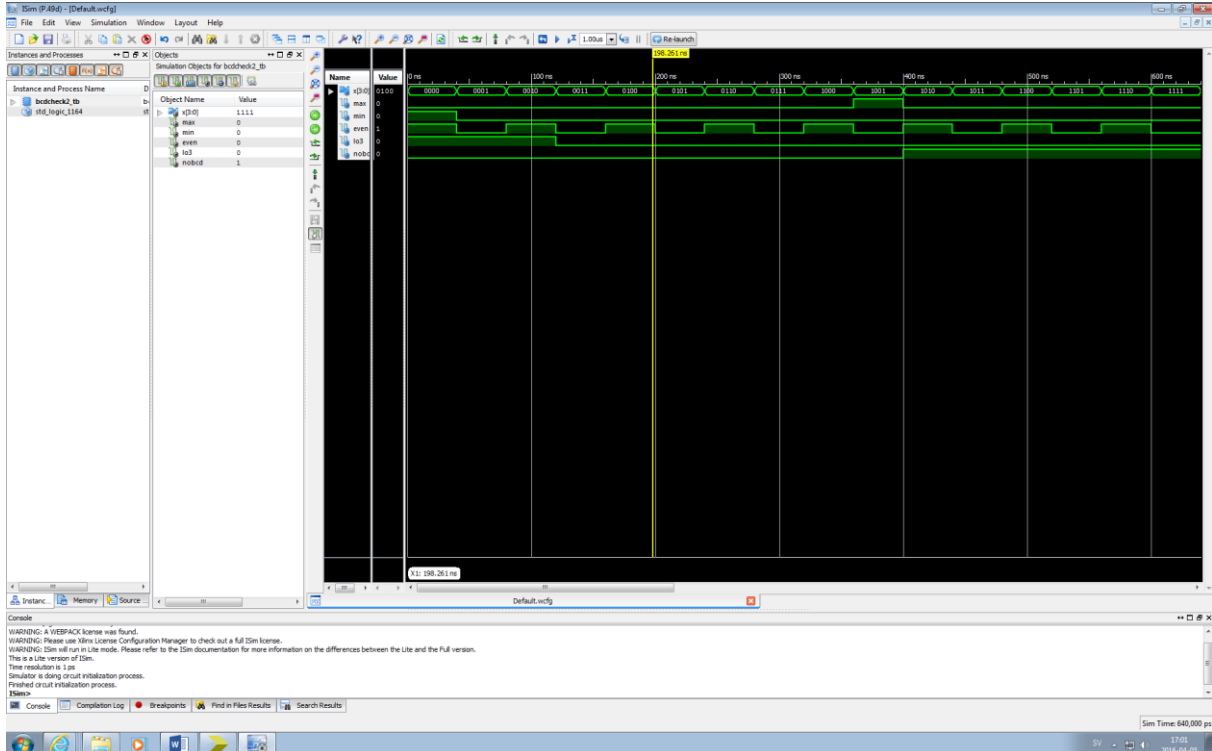
Lab 1

The truth table.

| Input (x) | max | min | even | lo3 | noBCD |
|-----------|-----|-----|------|-----|-------|
| 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 1 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 1 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 1 | 0 | 0 |
| 9 | 1 | 0 | 0 | 0 | 0 |
| 10 (A) | 0 | 0 | 1 | 0 | 1 |
| 11 (B) | 0 | 0 | 0 | 0 | 1 |
| 12 (C) | 0 | 0 | 1 | 0 | 1 |
| 13 (D) | 0 | 0 | 0 | 0 | 1 |
| 14 (E) | 0 | 0 | 1 | 0 | 1 |
| 15 (F) | 0 | 0 | 0 | 0 | 1 |

The number that I got was the number four.

The screenshot that I took in part four of the lab.



The signal output

When the signal output for max is decided the program checks if the input is nine and if that is true then the signal output for max becomes one, if the number is not nine then the signal output for max becomes zero. Therefore, the signal output for max with the input four is zero.

To decide the signal output for min the program checks if the input is zero, if that is true then the signal output is one otherwise the signal output is zero. Therefore, the signal output for min with the input four is one.

The signal output for even is decided by converting the last bit in the input. If the last number is zero then the signal output is one and if the last number is one then the signal output is zero. Since four is written as 0100_2 then the last bit is zero, therefore the signal output is one.

When the signal output for l03 is decided the program checks if the input lies between the numbers zero and two and if that is true then the signal output is one otherwise it is zero. Since four is greater than two then the signal output must be zero.

When it comes to the output for noBCD the signal output is one if the input lies between ten and fifteen, otherwise the signal output is zero. Therefore, since the input four is less than ten the signal output is zero.

Comparing the code

The first difference between the codes is that bcdcheck2 stores the value of the input in a vector while bcdcheck stores each bit in a separate variable. The next difference is when the output signals are decided, bcdcheck2 only looks at which number the input is while bcdcheck has to compare every bit in the input to get the same output as bcdcheck2. This is therefore the reason why bcdcheck has to use so many AND and NOT operators, because if we compare the code for max between the two then bcdcheck2 only checks if the number is nine, while bcdcheck has to make the output become one when the number is nine by using a combination of AND and NOT combinations. The code is then similar for the other output signals.

When it comes to the difference between the wave windows then the only difference is that bcdcheck2 saves the number as a nibble while bcdcheck saves the number in bits to different variables which you can see by every variable of the bits having its own wave. Otherwise the wave window looks the same for bcdcheck and bcdcheck2.