Comparator, Priority Encoder and Adder

Javier Mondragón Martin del Campo

Laboratory exercise number:

Lab. 03

Laboratory exercise name:

Comparator, Priority Encoder and Adder

|  |  |  |
| --- | --- | --- |
| Names: | Roll Number | Date |
| Javier Mondragón M. | A01365137 | 11/September/2019 |

Lab description:

The purpose of this laboratory is to learn how program new ways in VHDL for FPGA and practice using loops, signed and unsigned numbers and comparing operators. Also to learn how to simulate and test without using the FPGA.

The material used was:

ISE Project Naviator for the VHDL compiler and editor

Schematics, block diagrams and/or timing diagrams:

For the problem 5.8, the comparator, the following block was displayed in the problem, the logic behind it was with if statements and the output was an output port of 3 bits.

a(7:0)

b(7:0)

X1

X2

X3

sel

a>b

a=b

a<b

For the 5.2, the priority encoder, I used the following figure to create the code for the priority encoder, using if statements.

‘0’

‘1’

‘0’

‘0’

‘1’

‘1’

‘0’

‘1’

‘1’

‘0’

7

6

5

4

3 2

2 1

1 0

0

For the following figure, I used a for loop to implement the Adder, I had to use CIN and COUT as a port-vector with the same size as SUM.

CIN (Carry in)

a(7:0)

b(7:0)

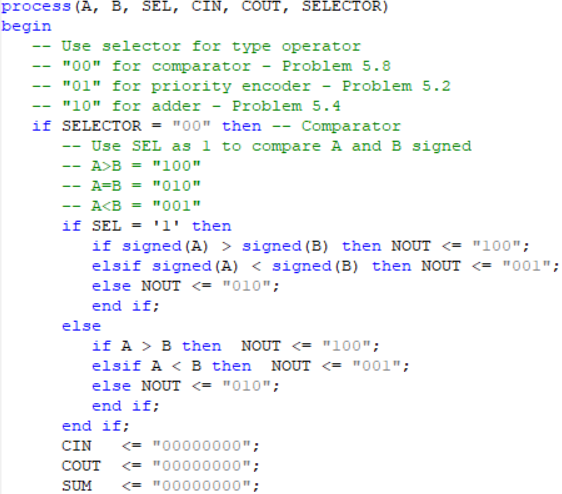
SUM(7:0)

COUT (Carry out)

+

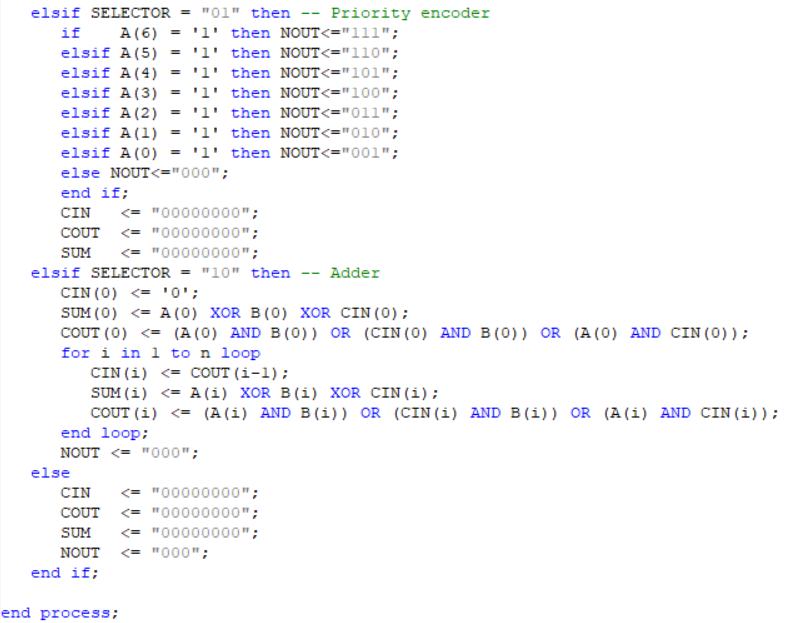
The code is displayed in the Image 1.1 and 1.2

Image 1.1



Comparator

Image 1.2



Adder

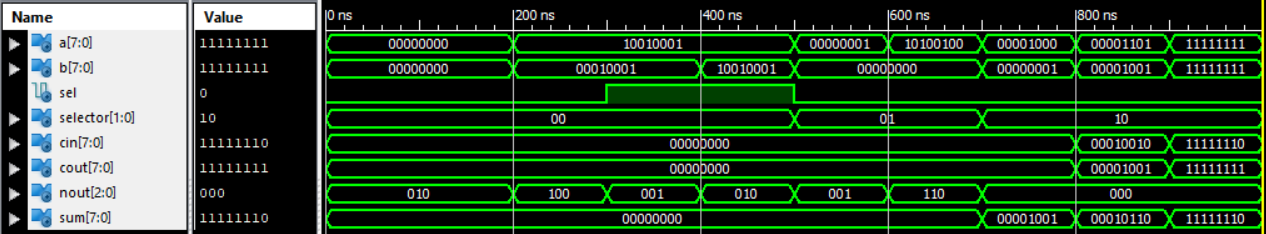
Results obtained:

The result was a successful comparator, priority encoder and an adder in the same MUX.

Evidence:

The selector is for the selection between comparator, priority encoder and an adder using “00” for Comparator, “01” for Priority encoder, “10” for Adder,

Image 2.1



Test results from Testbench

Conclusions:

The loops and conditionals facilitate the use of VHDL, making process faster and easier to scale. The simulation helps a lot for testing without having a FPGA and sometimes easier to see, also when your FPGA does not have enough physical ports to probe your program.

Problems encountered:

The for loop is designed to repeat hardware, not to repeat instructions and the ports must be threated as wires and not like variables.