EG3040 Programmable Electronics lab week 3
Designing a half adder in VHDL

1 Designing a half adder

Over the term, we will build up a portfolio of working examples of VHDL code on the FPGA board. The purpose of this weeks lab is to start thinking about hardware design using VHDL. Today you will make your first chip.

2 About this exercise

The half adder is a combinatorial circuit which is used to add two 1-bit binary digits. It produces two 1-bit binary outputs—sum and carry out. The half adder circuit consists of one XOR gate and one AND gate. The truth table with the circuit diagram of half adder is shown in figure 1.

2.1 Design guidance

Last week we learnt about entity and architecture blocks in a VHDL program. Entity declarations describe the interface of the component, i.e. input and output ports. In figure 1(a) the interface of a half adder is described where the input ports are *in1* and *in2* and the output ports are *sum* and *cout*. Architecture declarations describe what the circuit actually does—it implements the associated entity. All the statements in an architecture block execute concurrently and order independently. We were also introduced to the dataflow level, where we specify the data flows between hardware registers. The main feature of this level is concurrent assignment statements. In figure 1(c) the XOR and AND gates operate side by side where XOR gate will generate sum and AND gate will generate carry out (*cout*). We can implement these gates using concurrent assignment statements. For example the XOR gate can be implemented as *sum* <= *in1* XOR *in2* for *sum* and AND gate can be implemented as *cout* <= *in1* AND *in2* for carry out.

You can try to create your own VHDL file from scratch (including the relevant libraries and so forth), or you may wish to use the example file you were given last week as a template to get started with.

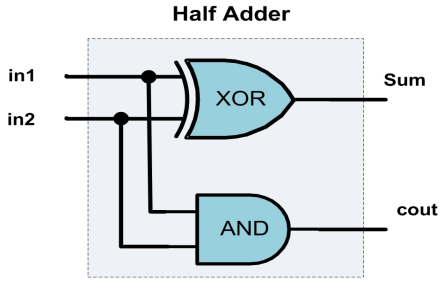
3 Test your code in the simulator

Register Transfer Level (RTL) is the name given to the actual format we download onto the FPGA. Once we have completed our RTL design, we need to test

(a)

in1	in2	sum	cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b)



(c)

Figure 1: Half adder design shows (a) block diagram with Input/Output ports, (b) truth table and (c) the circuit diagram.

its functionality. For this purpose we build a test bench in VHDL where we run the simulation to test the functionality of our RTL design under all the possible test cases. Throughout the term we will use the Modelsim-Altera tool for the simulation. For this lab exercise we will provide you a test bench for a half adder design. You can download it from your blackboard. You need to add it as a source file in your project (just as you did last week) and associate it with simulation only. You should properly investigate the functional simulation of your RTL design. Make sure there are no bugs in your RTL design!

4 Download the code onto the board

Finally, we are in a position to download the compiled bit file (the file that contains the FPGA configuration) onto the board. There are a number of

possible ways of doing this: we will be downloading straight onto the FPGA using the Quartus II tool via a USB link (just as you did last week). Don't forget to include the pin configuration file that you qere supplied with last week!

You might want to get really adventurous! For the advanced part of this lab you can try to connect your adder up to switches and LEDs on the board for input and output.

If you encounter any difficulties, please ask the demonstrators.