COEN 313
Digital Systems Design II Winter 2019
LAB 2: Structural and Concurrent VHDL

The purpose of this lab is to become acquainted with structural and concurrent VHDL. A secondary purpose is to become acquainted with different VHDL coding styles and to gain insight in combinational logic minimization as performed by logic synthesis tools.

Introduction

Consider the following truth table defining a three variable Boolean function:

Table 1: Some arbitrary Boolean Function.

A	B	C	OUT
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

The function may be expressed in *sum-of-minterms* form as: OUT = $\overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + A\,B\,C$. The gate level hardware implementation of the function is shown in Figure 1.

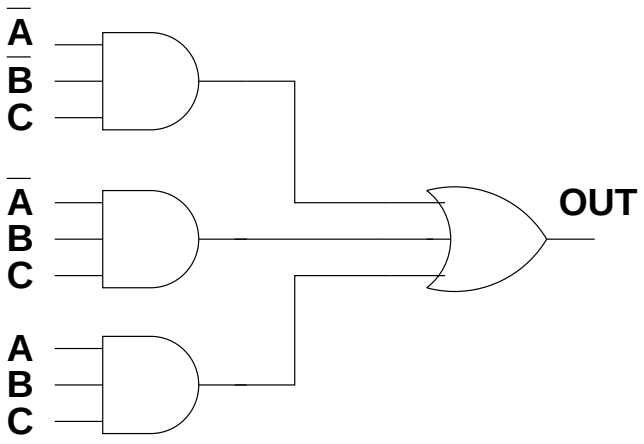

Figure 1: Two-level sum of minterms hardware implementation.

Procedure

Employ **structural** VHDL in your design by implementing the circuit shown in Figure 1 using **port map** statements to implement the AND and OR gates. Use **concurrent signal assignment statements** to obtain the required negated values of the input signals. Write entity-architecture descriptions for the AND and OR gate components using concurrent VHDL (i.e. the `and` operator for the AND gate). You may use a concurrent signal assignment statement in your top-level code to negate the output which drives the LED output. Simulate your design with the Modelsim simulator to verify correct functioning for all 8 input combinations. Synthesize your VHDL code with Precision RTL and obtain the RTL schematic diagram and the Tech schematics produced by the synthesis tool. Program the FPGA board with the Xilinx ISE tool. Demonstrate the operation of the design by downloading your synthesized code to the FPGA demonstration board. Use the following VHDL entity specification:

```
library IEEE;
use IEEE.std_logic_1164.all;

entity sum_of_minterms is
port( a,b,c           : in std_logic;
      output          : out std_logic);
end sum_of_minterms;
```

Questions

1. Rewrite the VHDL code for the `sum_of_minterms` entity making use of only CSA statements (no port maps). Re-synthesize your design with Precision RTL and compare the resulting RTL and Tech schematic diagrams with that of the original design. Comment on any differences/similarities between the two schematics. You do not have to download this version of the circuit to the FPGA board.

2. The Tech Schematic diagram indicates how the design is implemented using the available logic elements within the FPGA device. Xilinx FPGA devices use RAM based LUTs (Look Up Tables) to implement Boolean logic. Open up the Tech schematic of both versions of the design (the port map and the CSA only versions). One may determine the Boolean function a LUT has been programmed to implement by hovering on the outline of the LUT with the mouse cursor. Alternatively, the LUT information may be graphically displayed by right-clicking on the white background in the Tech Schematic window and selecting the **Show Lut Information** choice from the pop-up selection. The box representing the LUT will now contain logic gates which correspond to the Boolean function the LUT has been programmed to implement. Determine the Boolean function which the LUT implements in **both** versions of the VHDL code. Is the function equal to the original sum-of-minterms expression described by the VHDL code?

3. Comment on the Boolean function the LUT implements. Why do you think it is not equal to simply the minimized form of the function obtained by mapping the 1s in the K-map corresponding to the truth table? What does the Boolean function actually represent? (Consider mapping the

0s of the K-map). Has any logic minimization been performed by the sythesis tool on the orginal VHDL code?

Hint: What do you think the LUT Boolean function would be if we did **not** have to **negate** the top-level output port due to the active-low LEDs on rhe Xilinx XUPV2Pro development board? A sure fire way of obtaining the correct answer is to rewrite the VHDL code omitting the negation of the output port and observing the synthesized results...

4. Do the two RTL schematics indicate whether the synthesis tool has performed on logic minimization?

5. Determine whether the following VHDL code results in a RTL schematic which is indicative of logic minimization has take place:

```
library IEEE;
use IEEE.std_logic_1164.all;

entity keith is
port( input_1       : in std_logic;
      output        : out std_logic);
end keith;

architecture rtl of keith is

signal first , second : std_logic;

begin

first <= not input_1 ;
second <=  not first ;
output <= second;

end rtl ;
```

Deliverables

1. Modelsim simulation results for the port map version of the design.

2. RTL and Technology view schematic diagrams as produced by Precision RTL.

3. VHDL code for both versions of the design.

4. Answers to the questions.

T. Obuchowicz
February 1, 2018