COEN 313
Digital Systems Design II-Winter 2020
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.

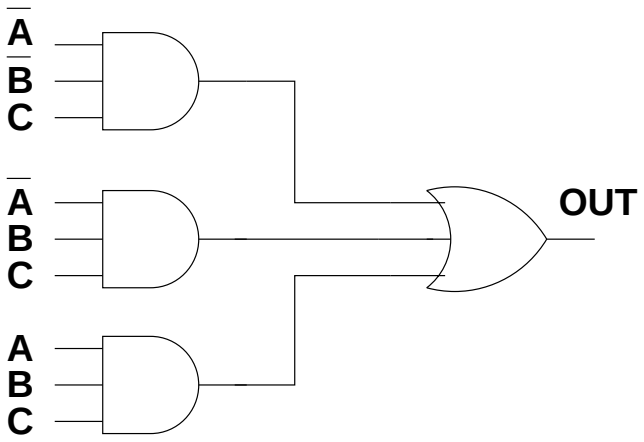

Figure 1: Two-level sum of minterms hardware implementation.

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). Simulate your design with the Modelsim simulator to verify correct functioning for all 8 input combinations. Synthesize and Implement your VHDL code with Vivado. Generate the bitstream and program the FPGA board with the Xilinx Vivado Hardware Manager 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;
```

Generating Schematic Views of a design

The Xilinx Vivado tools are capable of producing hardware schematic views of a synthesized and implemented design. The tool can also generate an RTL (register transfer level) elaborated view based upon the input VHDL source files used in the design. The procedure is briefly described here.

Elaborated View:

From the main Vivado window, select Open Elaborated Design from the RTL ANALYSIS. Refer to Figure 2.

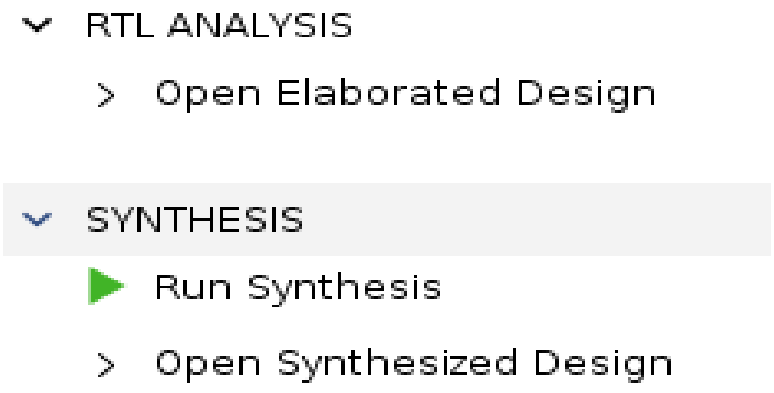

Figure 2: Selecting Open Elaborated Design.

A schematic diagram showing the inferred hardware will be generated and shown in the right hand pane of the Vivado Window as in Figure 3.

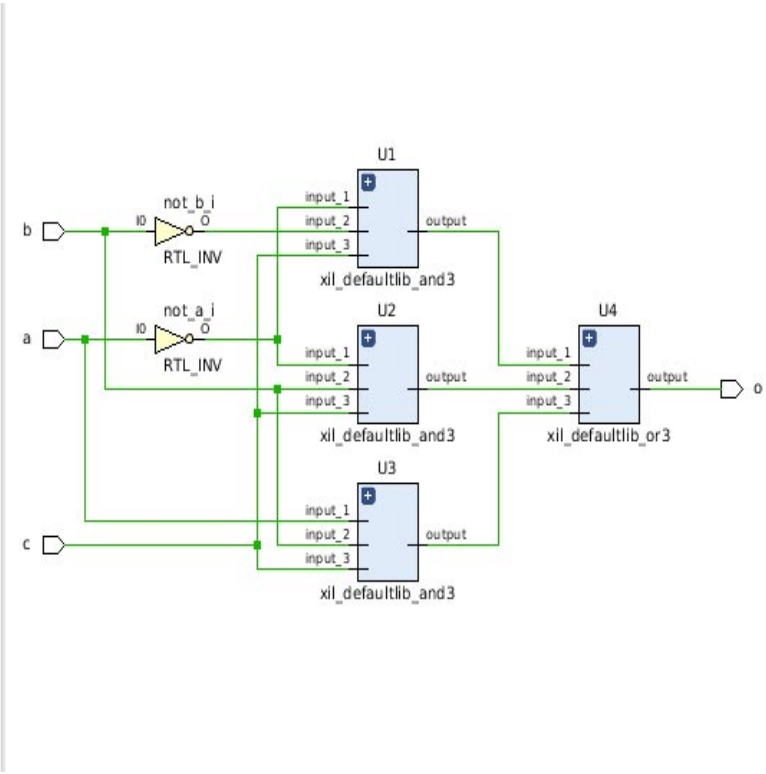

Figure 3: Elaborated Schematic.

Implemented Design:

The schematic diagram generated after the design has been synthesized and implemented by the Xilinx Vivado tools indicates how the design is implemented using the available logic elements within the Xilinx device. Xilinx FPGAs use RAM based LUTs (Look Up Tables) to implement Boolean logic.

Select Open Implemented Design from the IMPLEMENTATION portion of the Vivado main window as indicated in Figure 4.

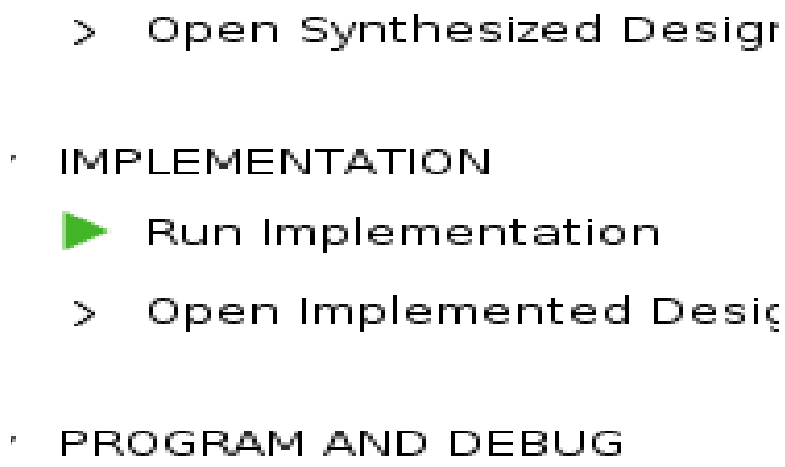

Figure 4: Opening the Implemented Design.

The implemented design will appear in the right hand part of the Vivado window and will show a representation of the internal organization of the FPGA device (in terms of LUTs, IO BUFS, and routing resources) as indicated in Figure 5.

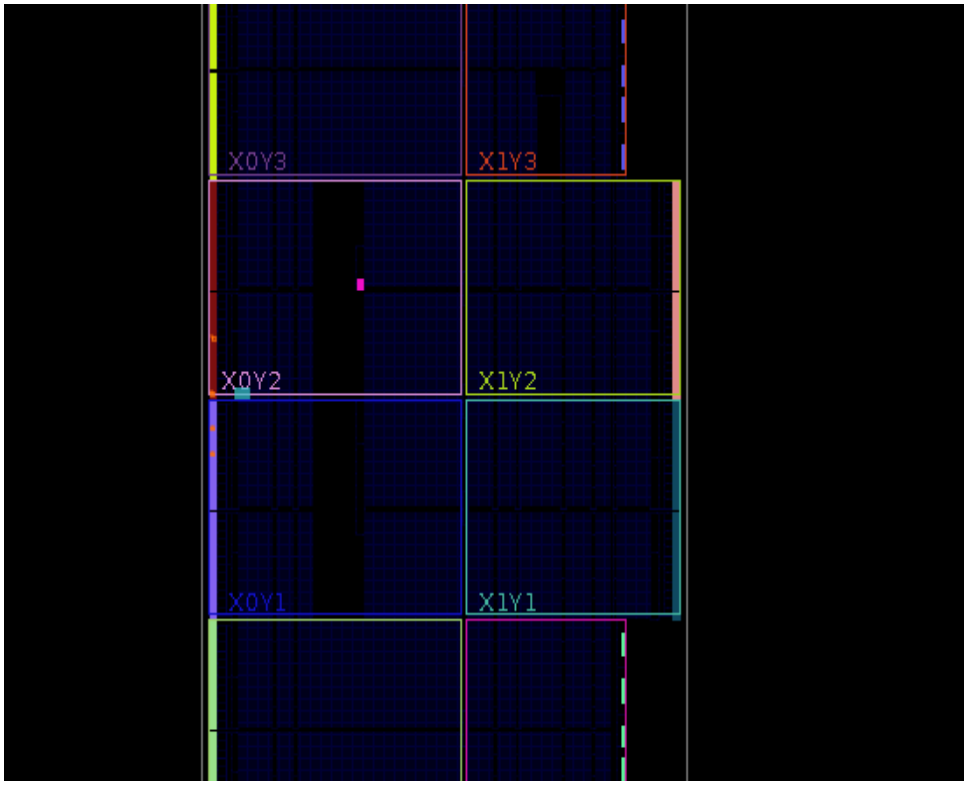

Figure 5: Default implemented design view.

To change to a schematic diagram representation of the implemented design select the Schematic (F4) icon listed under the “Sources” button in the IMPLEMENTED DESIGN pane (Figure 6). It is the second choice on the right and resembles a logic inverter on top and a wire on the bottom.

Figure 6: Selecting Schematic (F4).

The schematic will be now displayed as shown in Figure 7.

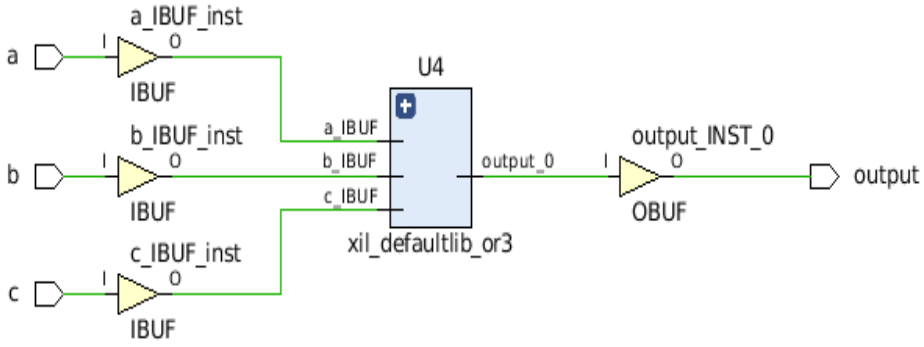

Figure 7: Schematic view of the implemented design.

To obtain the truth table that the LUT is implementing, double click the box labelled “xil_defaultlib_or3” which appears in the middle part of the schematic. Next, double click on the LUT3 box which appears in an outer box. Refer to Figure 8.

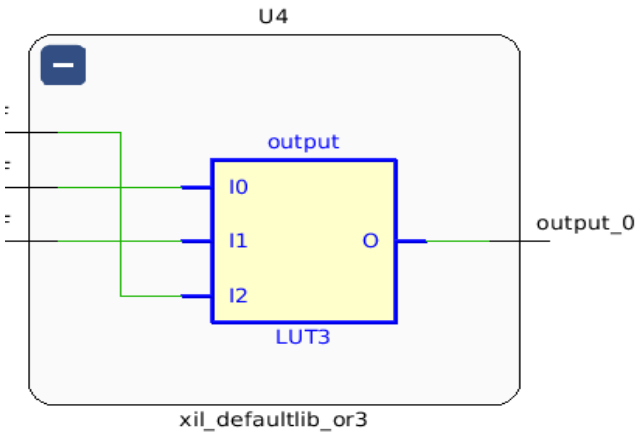

Figure 8: Selecting the LUT3 .

In the Cell Properties panel select the Show List icon (the last one on the bottom right resembling 4 horizontal grey lines) on the bottom right hand part of the panel and scroll to select the choice Truth Table as in Figure 9.

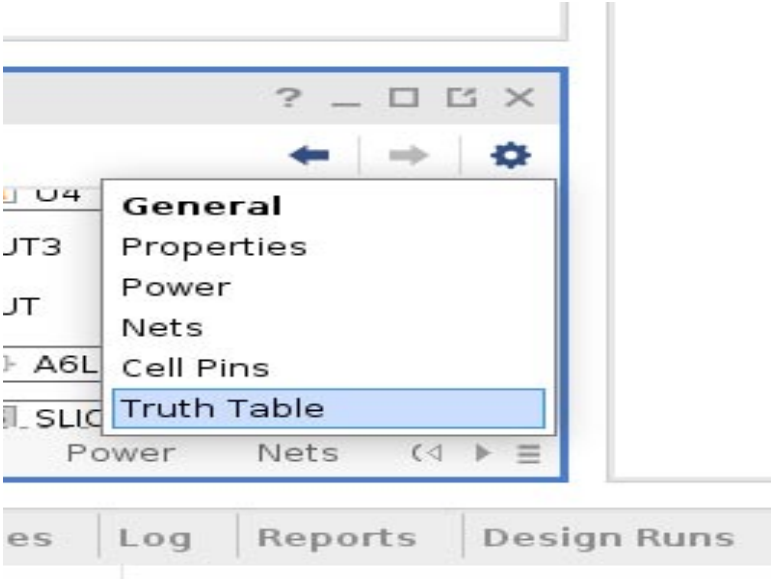

Figure 9: Selecting Truth table.

The Truth Table that the LUT has been programmed to implement together with the Boolean equation corresponding to the truth table appears in the Cell Properties (Figure 10). Note that the columns of the truth table are labelled I2 I1 I0 representing the three inputs to the LUT. You will have to refer to the schematic diagram to determine which top-level port has been mapped to each input of the LUT.

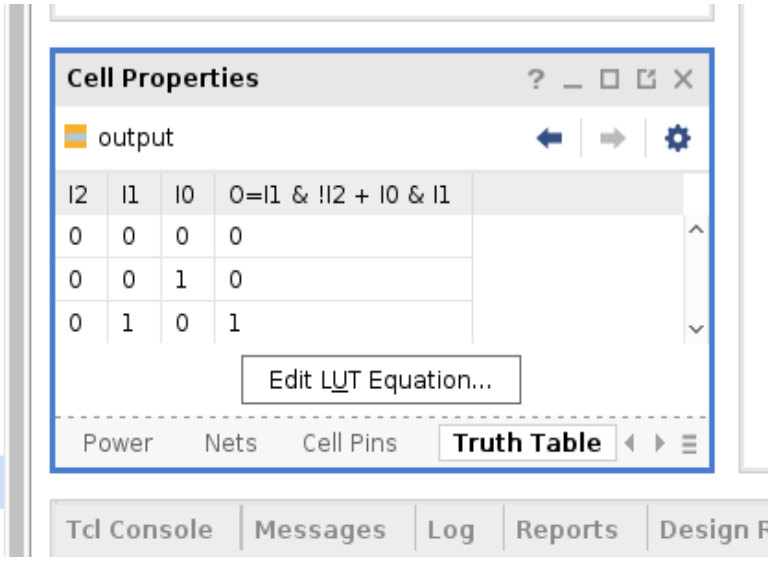

Figure 10: Truth Table and Boolean equation within the Cell Properties.

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 Vivado and compare the resulting RTL elaborated and Implemented schematic diagrams with that of the original design. Comment on any differences/similarities among the schematics. You do not have to download this version of the circuit to the FPGA board.

2. 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. Do the two RTL elaborated schematics indicate whether the synthesis tool has performed any logic minimization?

5. Determine whether the following VHDL code results in RTL elaborated and Implemented schematics which are indicative of logic minimization having been performed:

```
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 Elaborated and Implemented schematic diagrams as produced by the Vivado tool. (PDFs of the schematics may be obtained by selecting right clicking the background within a schematic and selecting the “Save as PDF” choice.)

3. VHDL code for both versions of the design.

4. Answers to the questions.

T. Obuchowicz
February 1, 2018

Revised: Jan. 31, 2020 for Vivado tools to generate schematics instead of Mentor Precision RTL.