

**Department of Electrical Engineering**

**Lab Report 2: Logic Circuits and Schematic Capture**

Name: Socheath Sok, ID: 014470701

Partner: Ashley Tran

Group: 11

Professor: Dr. Ray Wang

Class: EE 301

Date Performed: September 20, 2017

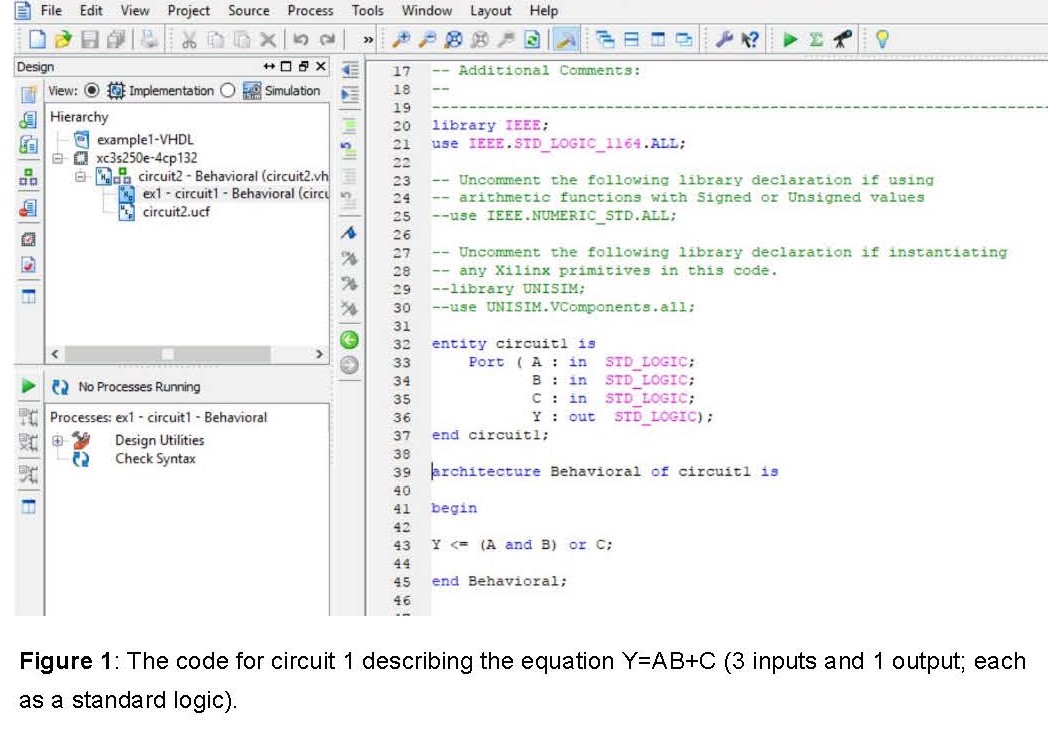
**Abstract**

The goal of this lab is to use the Xilinx ISE design software to create logic circuits in VHDL format and used a tool called ‘Schematic Capture’ to produce diagrams that represent the circuits. The lab was divided into two part: circuits designing and schematic creation. For the circuiting designing, we created a code for a simple boolean equation and used it as a component in the next circuit code. Similarly, for the schematic, we created a diagram for one circuit, and used it to create a more complicated one. After the completion of each part, we checked the functions with the Basys2 and got results that agree with the truth table for both parts.

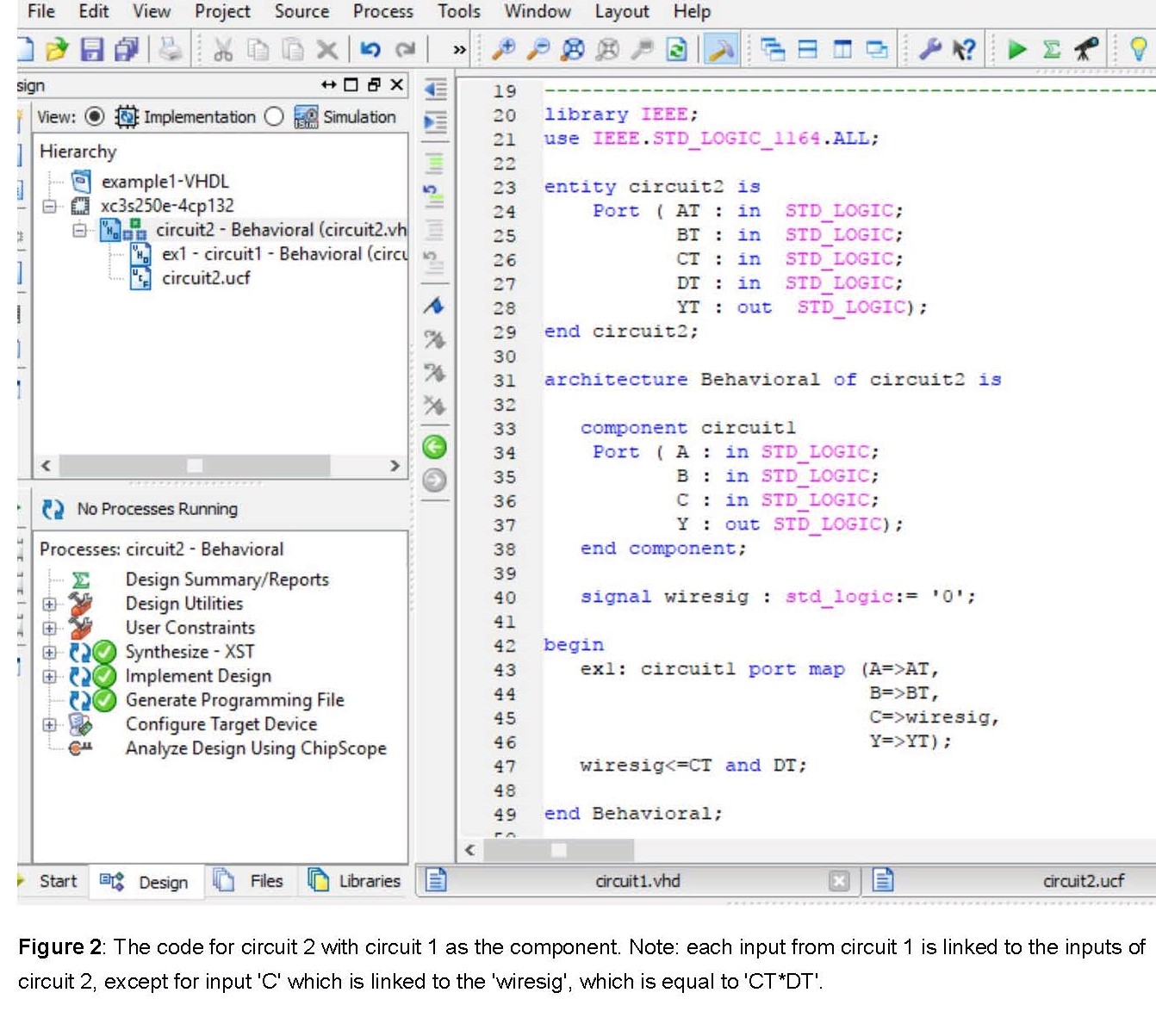
**Introduction**

1. **Part 1: Designing Logic Circuits, Simulation, and Implementation**

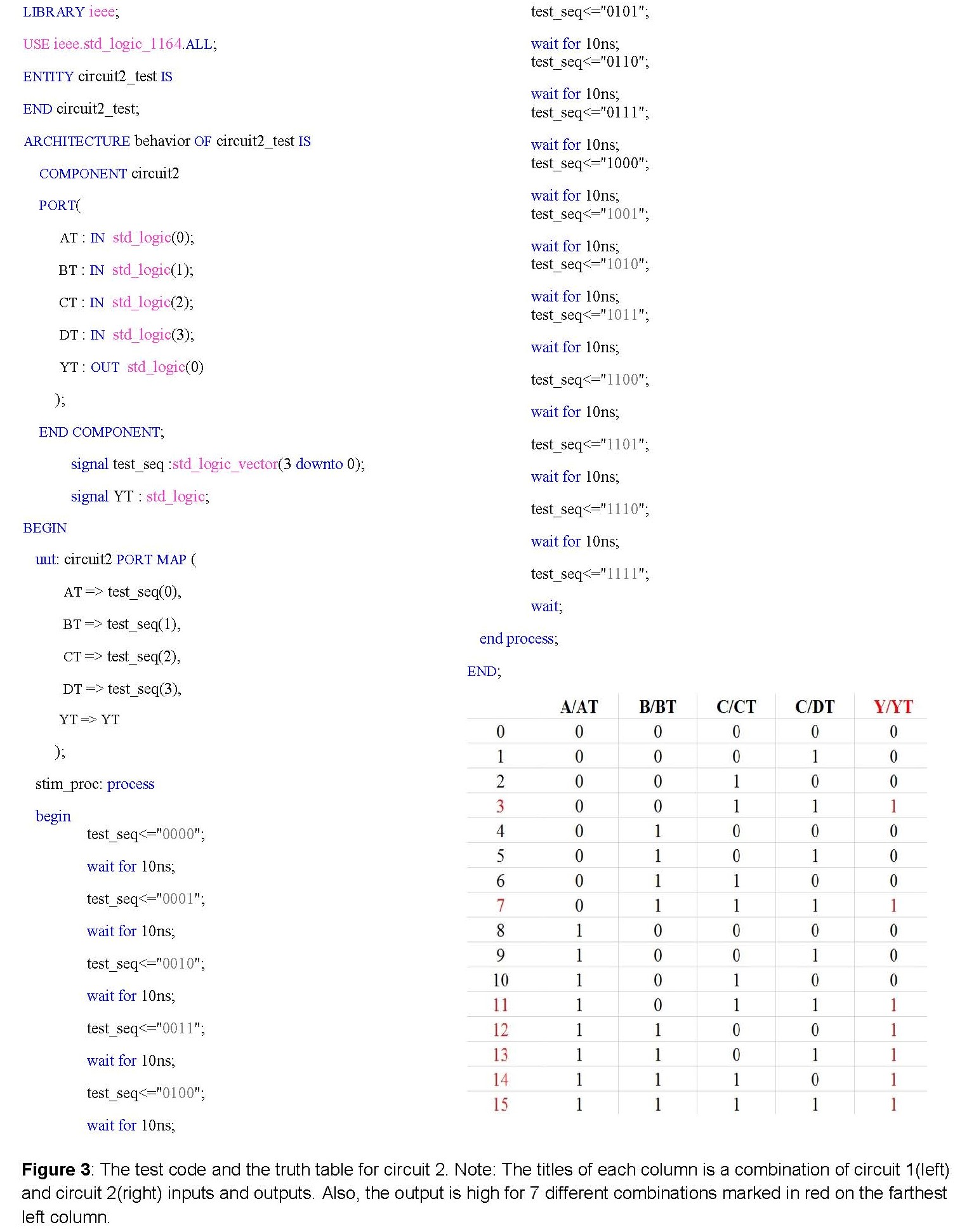
We started by creating a new project and defined the properties of the Basys2. Next, we added a VHDL module named ‘circuit 1’ with port names A, B, and C as the inputs and Y as the output. With our skeleton code, we added the boolean expression, ‘Y <= (A and B) or C;’, checked for syntax error and the result is shown in **Figure 1**.

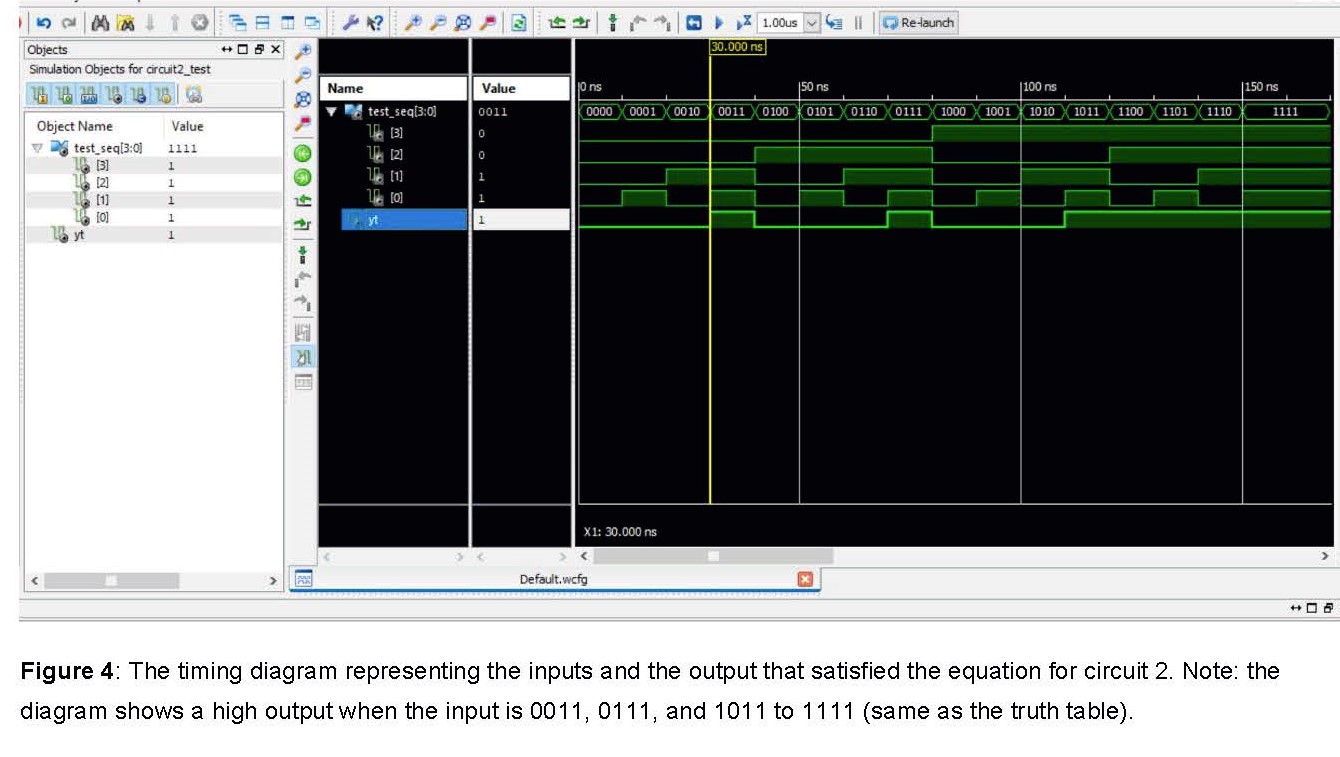


To show that the VHDL module for circuit 1 can be used as a structural component for other project, we added another source named ‘circuit 2’ which has four inputs (AT, BT, CT and DT) and one output (YT), and defined circuit 1 as the component. Additionally, we added a standard logic signal called ‘wiresig’ with initialization value of zero. In the ‘Instantiation of the structural component’ section, we connected inputs of circuit 1 to those of circuit 2, with the exception of input C, which is connect to the wiresig signal defined as (CT and DT). The finished code is shown in **Figure 2** below.

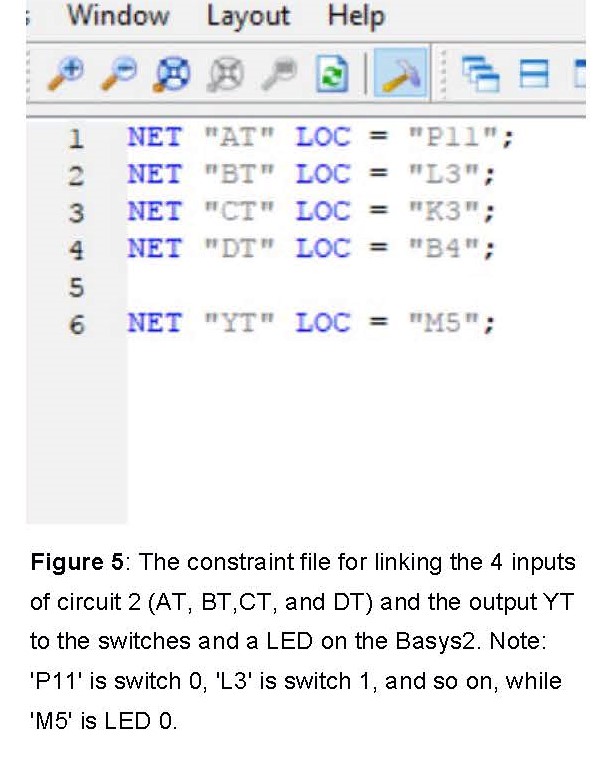


For the simulation, we needed a truth table to help create the test code, which we did by following the most of the steps done in lab 1. In the ‘Port’ section, instead of defining the inputs as a vector, we defined them as a standard logic, so we had to give each of them a value (0 to 3) and assigned each one to a test sequence. As for the ‘stimuli process’, we increased the test sequence command to 16 to represent each input combination in the table and the time required to 180ns. The final test code along with the truth table for circuit 2 can be seen in **Figure 3** while the timing diagram is shown in **Figure 4**.



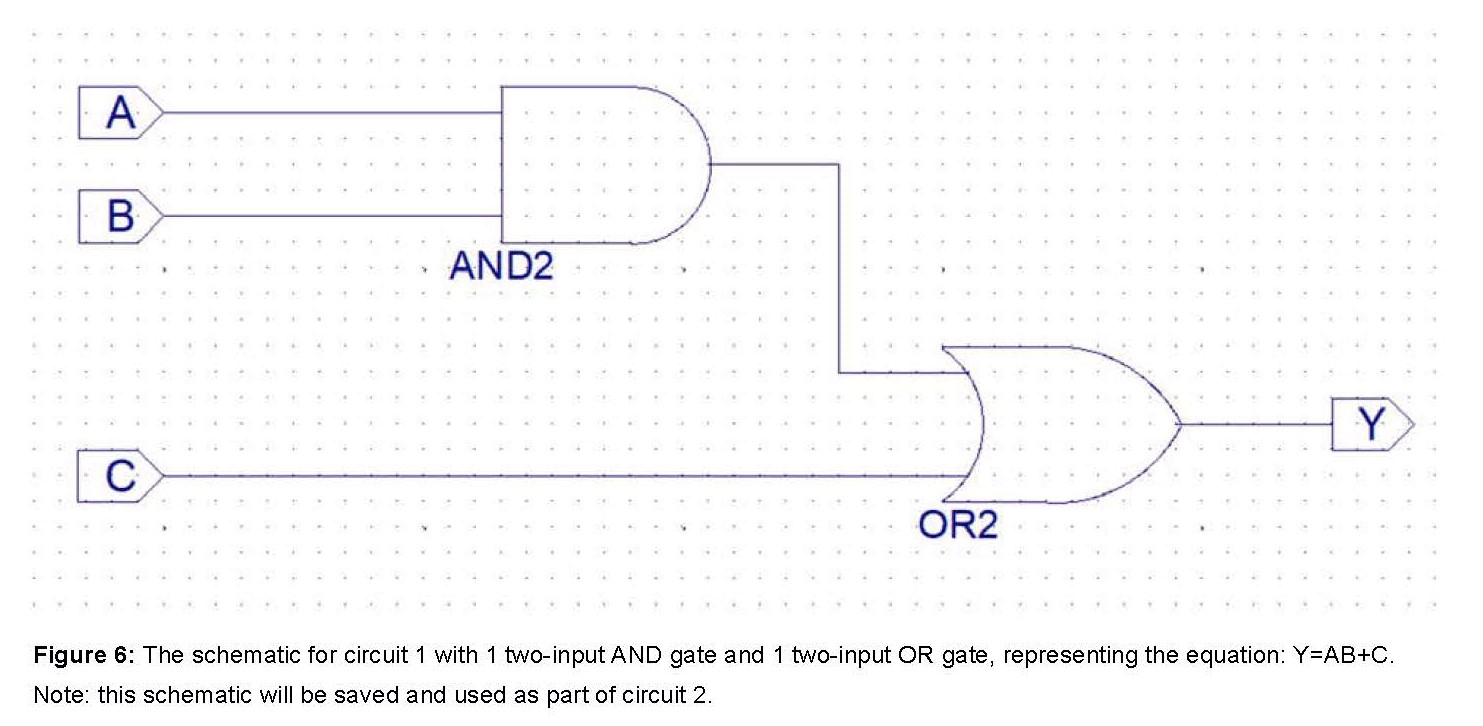


Since there is no problem with our code, we moved on to creating the constraint file in **Figure 5**. We assigned the 4 inputs to switch 0 to 3 and the output to LED 0 on the Basys2 and ran the file. After clearing all the errors, we generated and connected the programming file of circuit 2 to the board and found that the LED lighted up for input 3,7, and 11 to 15, which in accordance to our truth table in **Figure 3**.

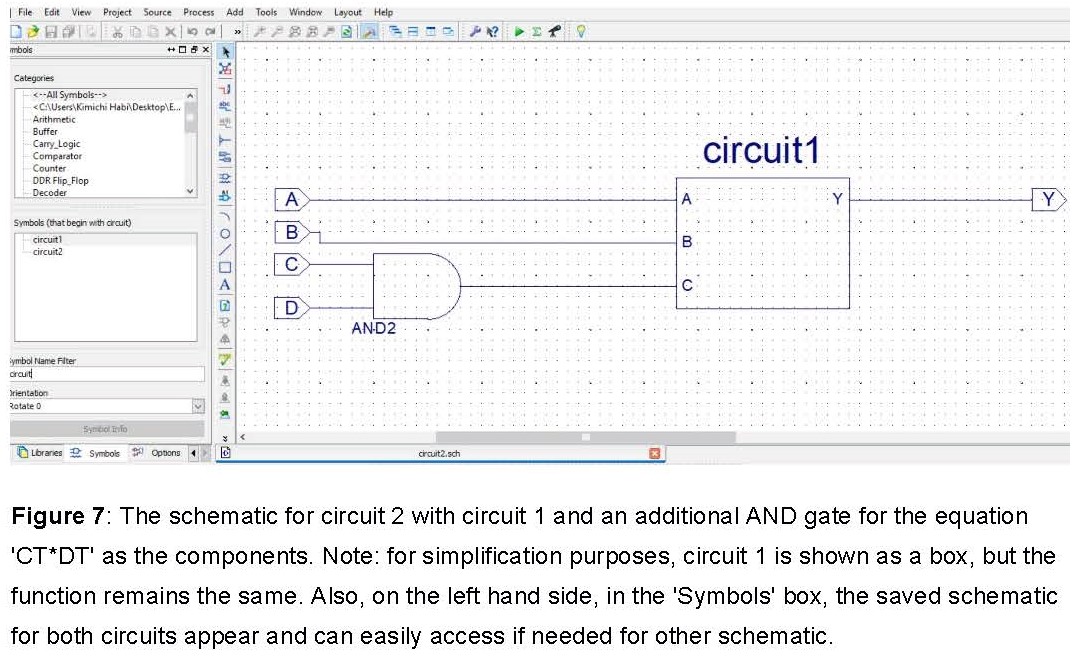


1. **Part 2: Schematic Capture**

Like the previous part, we created another new project and followed the same step, but this time the ‘Top-level source type’ is ‘Schematic’. Next, we added Schematic as our new source and called it ‘circuit 1’ because we are using the circuit 1’s equation to create our diagram. In the ‘Schematic Editor Window’, we selected the a two-input AND gate and a two-input OR gate, identified the inputs and output, and connected them based on the the equation Y=AB+C. The diagram for circuit 1 in **Figure 6** was saved and made into a macro symbol for future use in the ‘Symbol Wizard’ section of the Tool function.



Lastly, we added another schematic source (‘circuit 2’) based on the equation Y=AB+CD, where AB+C is the equation that defined the schematic of circuit 1. In the ‘symbols’ section of the editor window, we searched for circuit 1 and used it to create a diagram for circuit 2. The final schematic along with the macro symbols for both circuit can be seen in **Figure 7**. To check our work, we copied the constraint file in **Figure 5**, and implemented it. The result shows that the programming file for the schematic works exactly like the one in Part 1.



**Discussion**

The actual coding for both circuits in Part 1 were rather simple and straightforward. However, for the test code, it took us a while to figure everything out because we were not sure what to change for the ‘Port’ and ‘Port map’ sections. As for the second part, since the tutorial shows an older version of the program, we had to spend some times getting to know all the basic function needed for this lab before actually doing it. Nevertheless, we managed to avoid mistakes from previous lab, which, overall, helped us finished this lab faster. After completing this lab, we learnt how to use existing codes and schematic files as part of a new project. This can save use a lot of time and avoid unnecessary repetition which can create more errors or just slow us down in future labs.