Ege Ozan Özyedek
Experiment 2 – Lab 2
10.10.2018
EEE102 – 02

- The Design Methodology

In this lab, a combinational circuit that resembled a problematic production phase of an electronics company was set out to be created. The circuit had 3 variables/inputs which were Supply (symbolizing supply shortages), Analysis (symbolizing issues that come from wrong analysis) and Help (symbolizing another manufacturer helping to resolve the possible issues). Their binary states are as follows

Supply
- 0: No supply shortages/issues
- 1: Supply issues exist which may cause production delay

Analysis
- 0: No analysis issues
- 1: Analysis issues exist which may cause production delay

Help
- 0: Help does not exist
- 1: Help exists which can resolve one existing issue

Production
- 0: Production will be finished on time
- 1: Production will be delayed

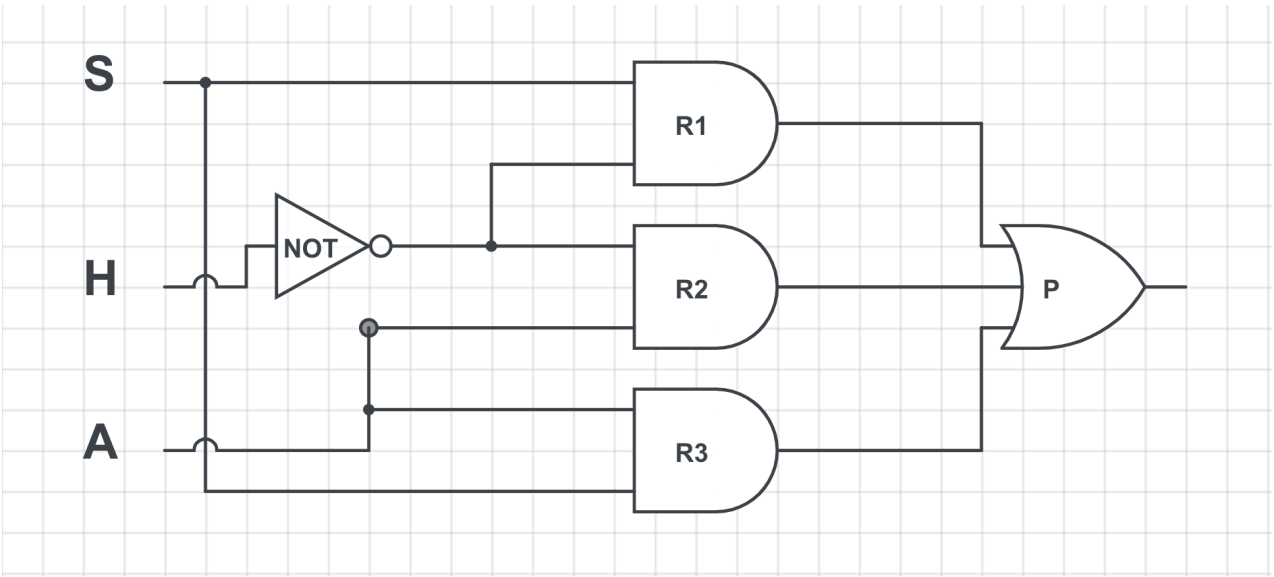

Figure 1: Planned Combinational Circuit of Production Delay

Ege Ozan Özyedek
Experiment 2 – Lab 2
10.10.2018
EEE102 – 02

Table 1: The Truth Table of The Combinational Circuit

S	H	A	P	Minterm
0	0	0	0	m_0
0	0	1	1	m_1
0	1	0	0	m_2
0	1	1	0	m_3
1	0	0	1	m_4
1	0	1	1	m_5
1	1	0	0	m_6
1	1	1	1	m_7

- Results

First off, we can use the truth table, minterms and SOP to find the formula of P:

$$P = m_1 + m_4 + m_5 + m_7$$

$$P = (\bar{S}.\bar{H}.A) + (S.\bar{H}.\bar{A}) + (S.\bar{H}.A) + (S.H.A)$$

$$P = \bar{H}.\ (\bar{S}.A + S.\bar{A}) + S.A\ .(H + \bar{H})$$

Since $(H + \bar{H}) = 1$ we find the formula as

$$P = \bar{H}.\ (\bar{S}.A + S.\bar{A}) + S.A$$

Using Vivado, the planned combinational circuit was created. The code which embody the circuit design is as follows:

```
production <= (((not Help) and Supply) or (Supply and Analysis) or
(Analysis and (not Help)));
```

This code turns into the RTL schematic below

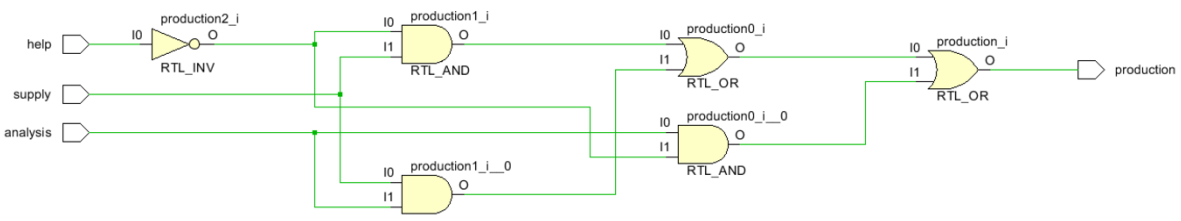

Figure 2: RTL Circuit Schematic

Ege Ozan Özyedek
Experiment 2 – Lab 2
10.10.2018
EEE102 – 02

The original design shows the OR-gate at the end having three inputs. However, the schematic generated from the code has 2 OR-gates. One connects with the other to create the same circuit at the end. This might be because there exist no three-input gates on Vivado.

After the coding of the circuit, a test bench was created. The code for this can be seen at the end of this report under Appendices. The simulation showcased all possibilities that might occur in this circuit design. A screenshot of the test bench can be seen below with all possible binary inputs and outputs.

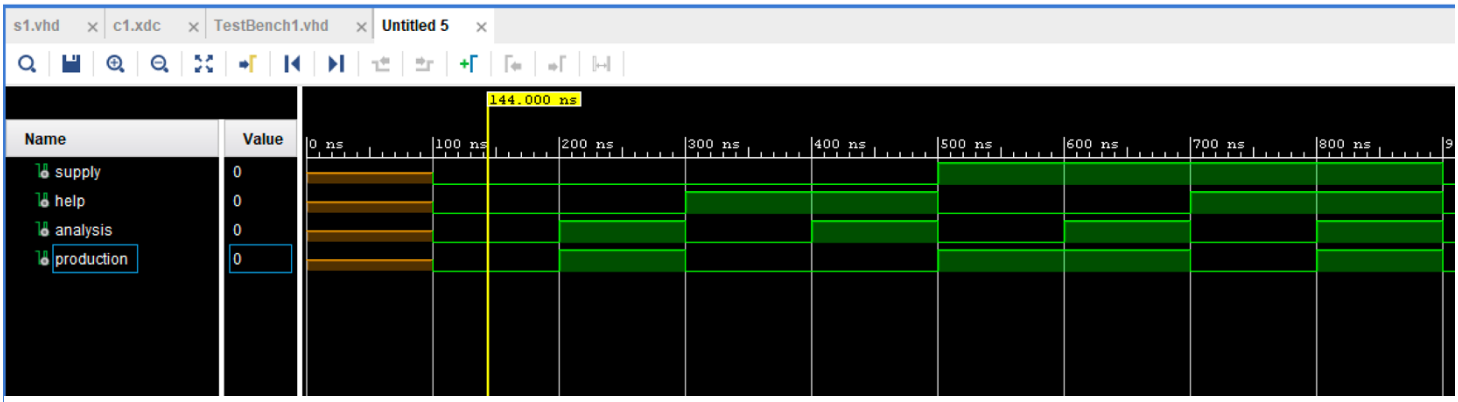

Figure 3: Test Bench Simulation which shows all possible binary inputs and their outputs

After testing, the design code was implemented on Basys 3. Three switches were used to create three binary input variables and one LED was used to indicate the binary output value. All possibilities can be seen on the pictures below. For context

- Analysis: V17 (Switch)
- Help: V16 (Switch)
- Supply: W16 (Switch)
- Production: U16 (LED)

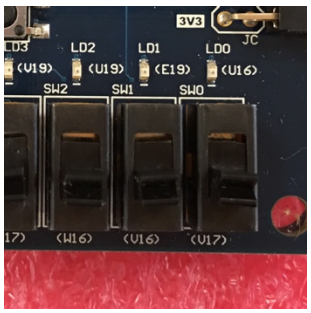

Figure 4: S:0 H:0 A:0 P:0

Figure 5: S:0 H:0 A:1 P:1

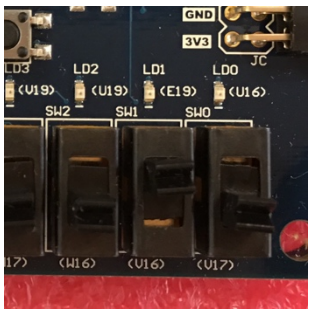

Figure 6: S:0 H:1 A:0 P:0

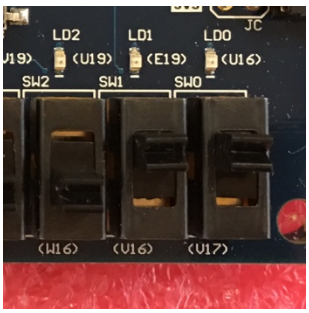

Figure 7: S:0 H:1 A:1 P:0

Ege Ozan Özyedek
Experiment 2 – Lab 2
10.10.2018
EEE102 – 02

Figure 8: S:1 H:0 A:0 P:1

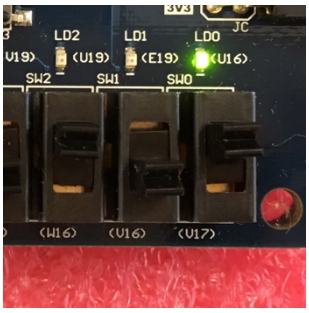

Figure 9: S:1 H:0 A:1 P:1

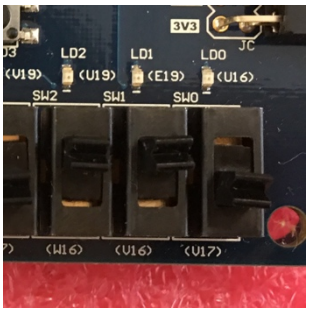

Figure 10: S:1 H:1 A:0 P:0

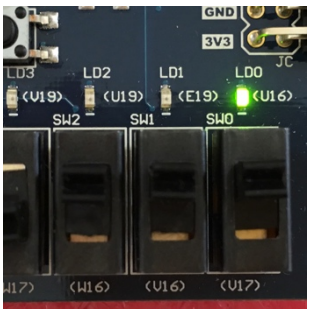

Figure 11: S:1 H:1 A:1 P:1

- Conclusions

  This experiment has familiarized me with the VHDL language and has taught me how to use Vivado and how to program onto Basys 3. This was a good start in learning how to program different types of circuit boards and how to link code with physical attributes. It was also a simple introduction to VHDL which might help me in my future with digital design. Although all of my results were as expected, I did have a slight difference between my planned combinational circuit and the RTL Schematic. Vivado couldn't draw (or comprehend?) a three input OR-gate so it divided it to two OR-gates. The final results are the same and logically it is the same but its worth noting.

- Appendices

  VHDL Design Code (used to design the logic circuit)

```
----------------------------------------------------
-- Company: Bilkent University
-- Engineer: Electrical and Electronics
-- Create Date: 09/10/2018 09:12:47 PM
-- Design Name: Production
-- Module Name: s1 - Behavioral
-- Project Name: Lab 02
-- Target Devices: Basys 3
-- Description: VHDL code which creates a logic circuit
that resembles a production phase of a electronics
company.
----------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity s1 is
    Port ( supply : in STD_LOGIC;
           help : in STD_LOGIC;
           analysis : in STD_LOGIC;
           production : out STD_LOGIC);
end s1;

architecture Behavioral of s1 is

begin

production <= (((not Help) and Supply) or (Supply and
Analysis) or (Analysis and (not Help)));

end Behavioral;
```

Ege Ozan Özyedek
Experiment 2 – Lab 2
10.10.2018
EEE102 – 02

VHDL Code for the Test Bench Simulation

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity testBench1 is
end TestBench1;
architecture Behavioral of TestBench1 is
COMPONENT s1
PORT( supply : IN STD_LOGIC;
help : IN STD_LOGIC;
analysis : IN STD_LOGIC;
production : OUT STD_LOGIC);

END COMPONENT;

SIGNAL supply : STD_LOGIC;
SIGNAL help : STD_LOGIC;
SIGNAL analysis : STD_LOGIC;
SIGNAL production : STD_LOGIC;

BEGIN
UUT: s1 PORT MAP(
supply => supply,
help => help,
analysis => analysis,
production => production);

TestBench1 : PROCESS BEGIN

wait for 100 ns;
supply<='0';
help<='0';
analysis<='0';

wait for 100 ns;
supply<='0';
help<='0';
analysis<='1';

wait for 100 ns;
supply<='0';
help<='1';
analysis<='0';
```

VHDL Constraints Code (used to set up switches and LED on Basys 3)

```
set_property PACKAGE_PIN V17 [get_ports {supply}]
    set_property IOSTANDARD LVCMOS33 [get_ports {supply}]
set_property PACKAGE_PIN V16 [get_ports {help}]
    set_property IOSTANDARD LVCMOS33 [get_ports {help}]
set_property PACKAGE_PIN W16 [get_ports {analysis}]
    set_property IOSTANDARD LVCMOS33 [get_ports
{analysis}]
set_property PACKAGE_PIN U16 [get_ports {production}]
    set_property IOSTANDARD LVCMOS33 [get_ports
{production}]
```