* **The Design Methodology**

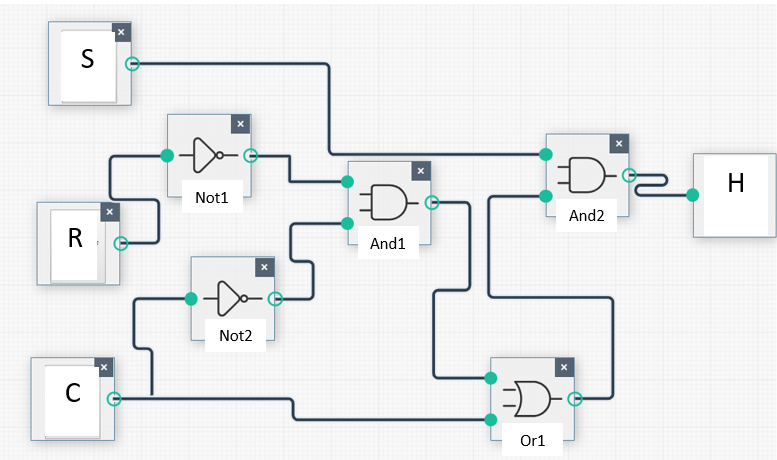
In my case output is entrance to bank vault at the gas station and inputs are S(supervisor), C(cashier) and R(rookie). Only the S(supervisor) has access the bank vault so if someone else wants to get in the vault they must be together with the S(supervisor). Also, R(rookie) always must be with the C(cashier) to be monitored. If R(rookie) is in the vault, then C(cashier) must be in the vault as well as the supervisor. To provide these conditions first I designed truth table accordingly. Then I produce the logic expression from the sum of products method.

* **Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S | C | R | H(S,C,R) | Minterms |
| 0 | 0 | 0 | 0 |  |
| 0 | 0 | 1 | 0 |  |
| 0 | 1 | 0 | 0 |  |
| 0 | 1 | 1 | 0 |  |
| 1 | 0 | 0 | 1 |  |
| 1 | 0 | 1 | 0 |  |
| 1 | 1 | 0 | 1 |  |
| 1 | 1 | 1 | 1 |  |

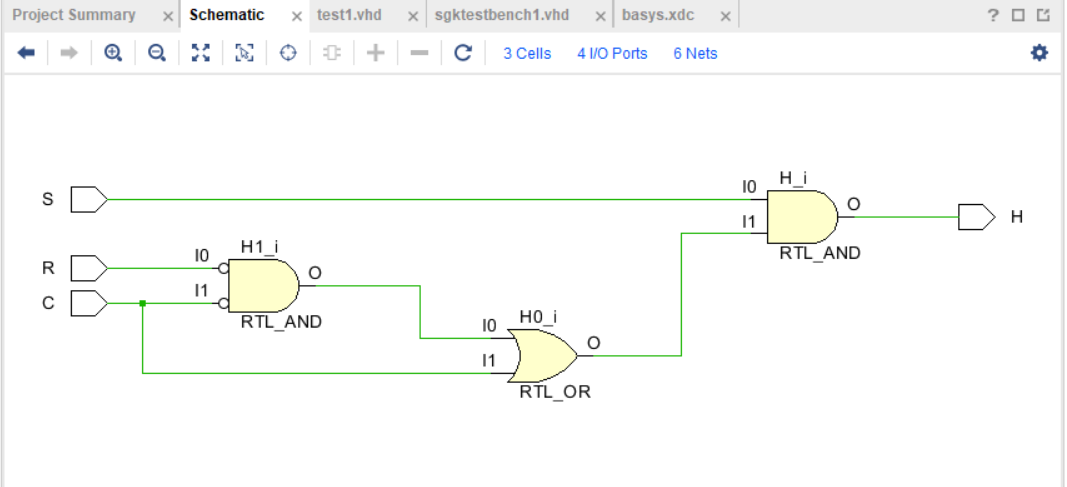
Truth Table

Sum of Products(minterms)

**Logic Schematic Diagram**

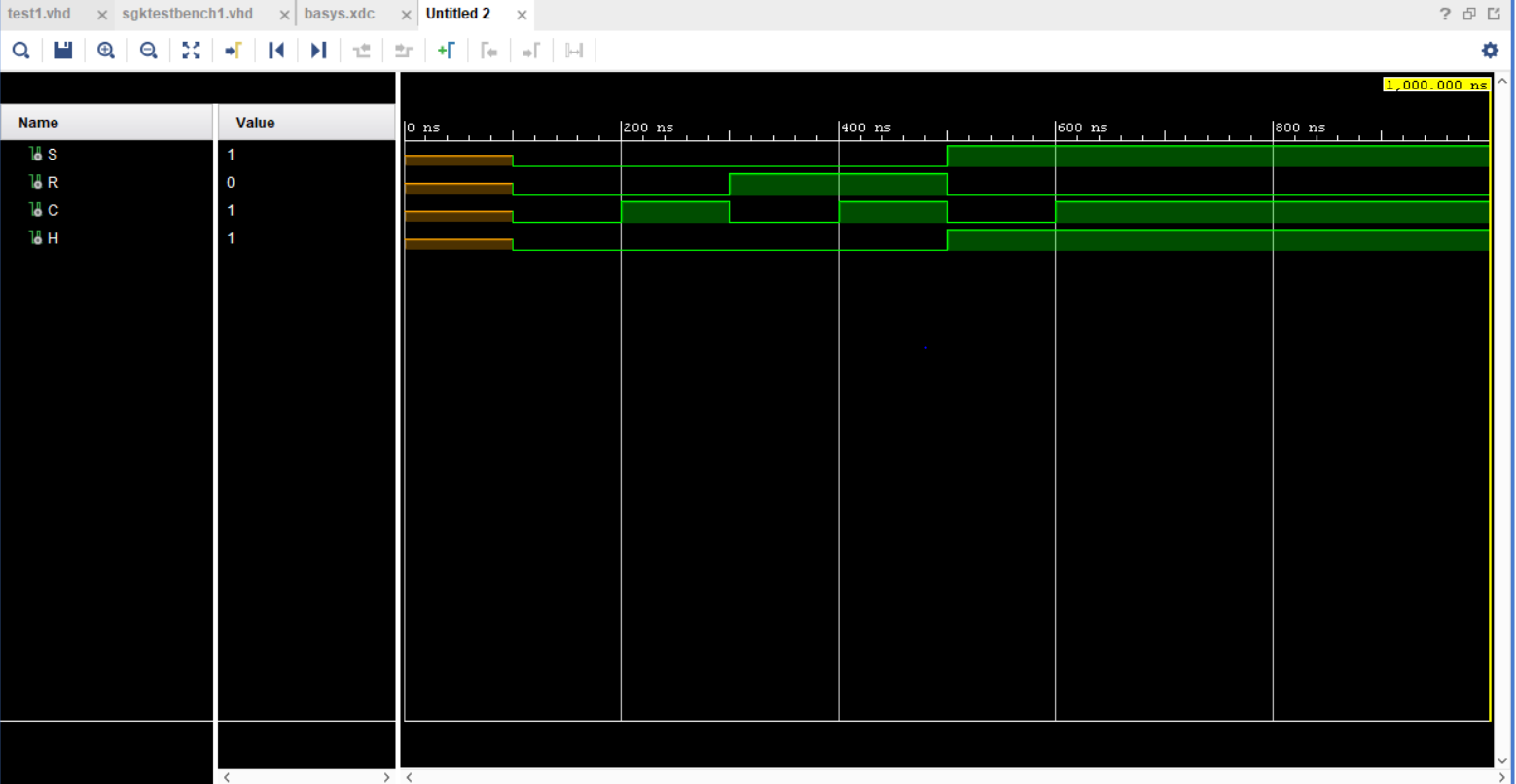
* **Circuit Diagram:**

Program uses bubble to bubble logic to show the diagram of my logic expression. Displayed diagram a bit different but truth table is same with mien.

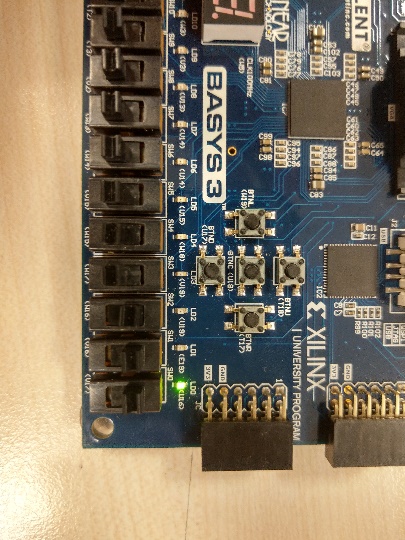
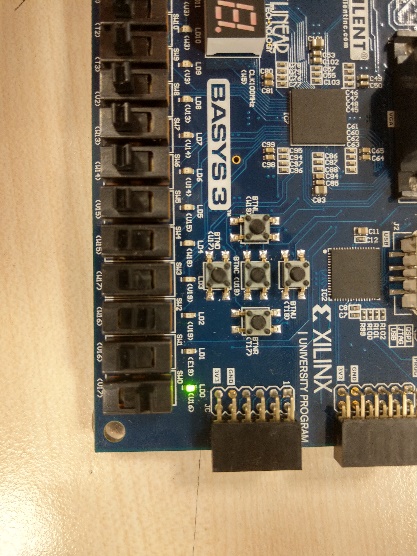


* **Simulation:**

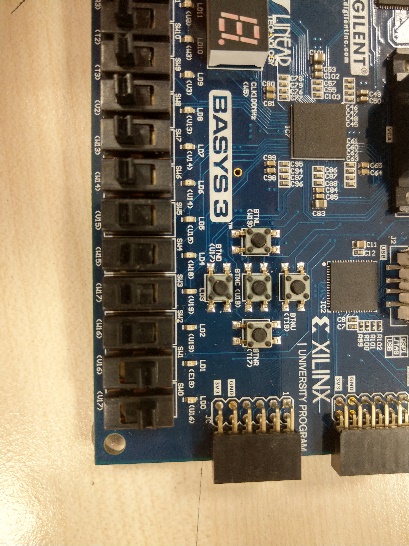
I entered 8 different case and they are all matching with my truth table.



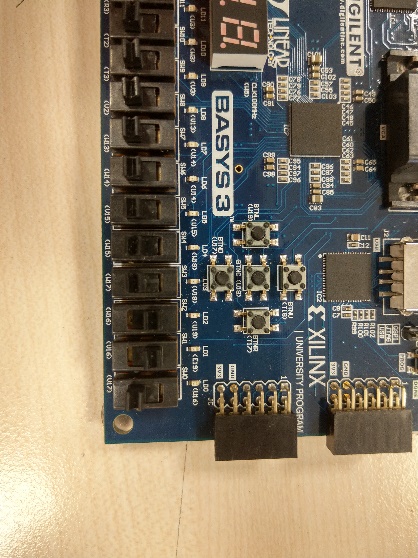
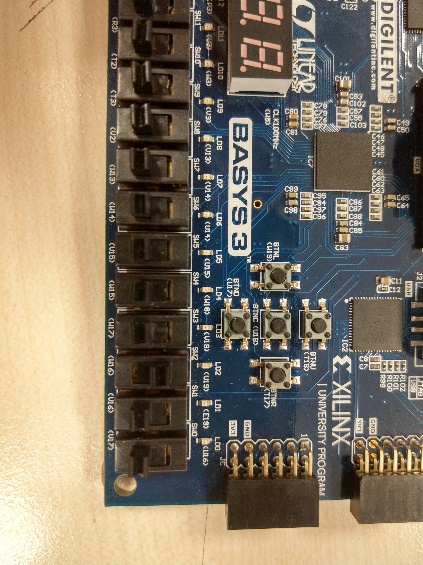
These are photos of the all possible 8 scenerios:

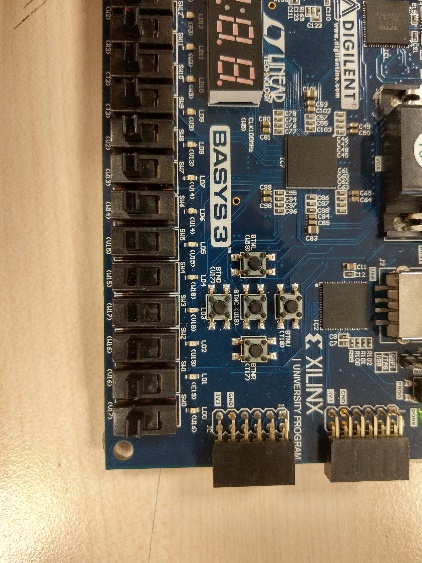
Case8 Case7

Case6 Case5

Case4 Case3

Case2 Case1

* **Conclusion:**

After doing this lab, with help of the tutorial that uploaded to the Moodle I learnt how to use Vivado program such as adding design sources, simulation sources, adding constraints, how to write pin configuration, generating Bitstreams to use on Basys3 and most important thing I learnt is writing codes in VHDL code language. To conclude, with all of these I complete Lab02 assignment by myself.

* **Appendices :**

**VHDL code:**

* **Design sources:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity test1 is

Port ( S : in STD\_LOGIC;

R : in STD\_LOGIC;

C : in STD\_LOGIC;

H : out STD\_LOGIC);

end test1;

architecture Behavioral of test1 is

begin

H <= S and ((not R and not C) or C);

end Behavioral;

* **Simulation sources:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity sgktestbench1 is

end sgktestbench1;

architecture Behavioral of sgktestbench1 is

component test1

Port ( S : in STD\_LOGIC;

R : in STD\_LOGIC;

C : in STD\_LOGIC;

H : out STD\_LOGIC);

end component;

SIGNAL S : STD\_LOGIC;

SIGNAL R : STD\_LOGIC;

SIGNAL C : STD\_LOGIC;

SIGNAL H : STD\_LOGIC;

BEGIN

UUT: test1 PORT MAP(

S => S,

R => R,

C => C,

H => H

);

testBench1 : PROCESS

BEGIN

wait for 100 ns;

S<='0';

R<='0';

C<='0';

wait for 100 ns;

S<='0';

R<='0';

C<='1';

wait for 100 ns;

S<='0';

R<='1';

C<='0';

wait for 100 ns;

S<='0';

R<='1';

C<='1';

wait for 100 ns;

S<='1';

R<='0';

C<='0';

wait for 100 ns;

S<='1';

R<='0';

C<='1';

wait for 100 ns;

S<='1';

R<='1';

C<='0';

wait for 100 ns;

S<='1';

R<='1';

C<='1';

END PROCESS;

end Behavioral;

* **Constrains:**

set\_property PACKAGE\_PIN V17 [get\_ports {S}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {S}]

set\_property PACKAGE\_PIN V16 [get\_ports {R}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {R}]

set\_property PACKAGE\_PIN U16 [get\_ports {H}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {H}]

set\_property PACKAGE\_PIN W16 [get\_ports {C}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {C}]

connect\_debug\_port dbg\_hub/clk [get\_nets <clock\_net\_name>]