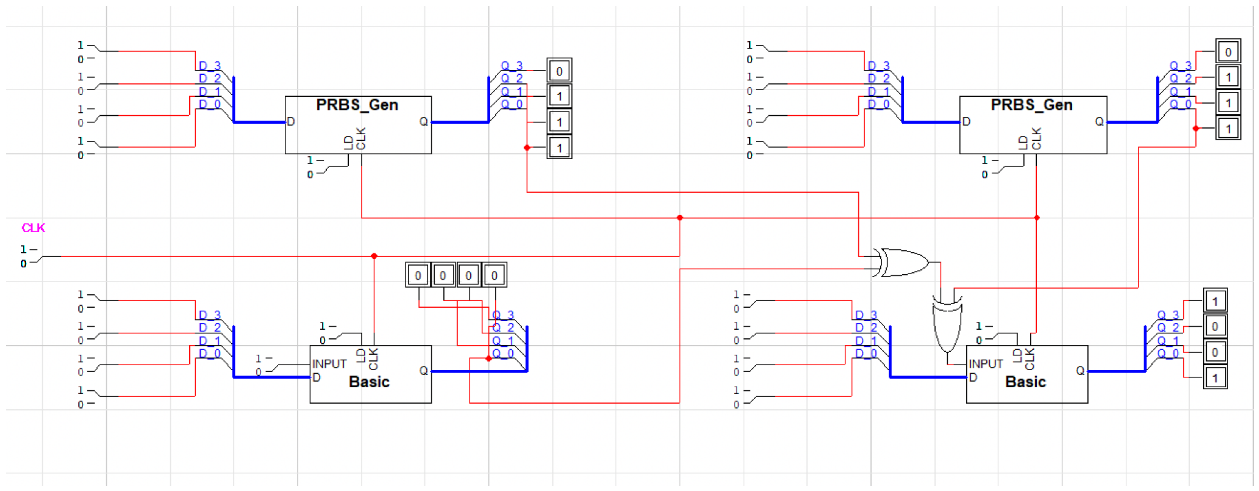
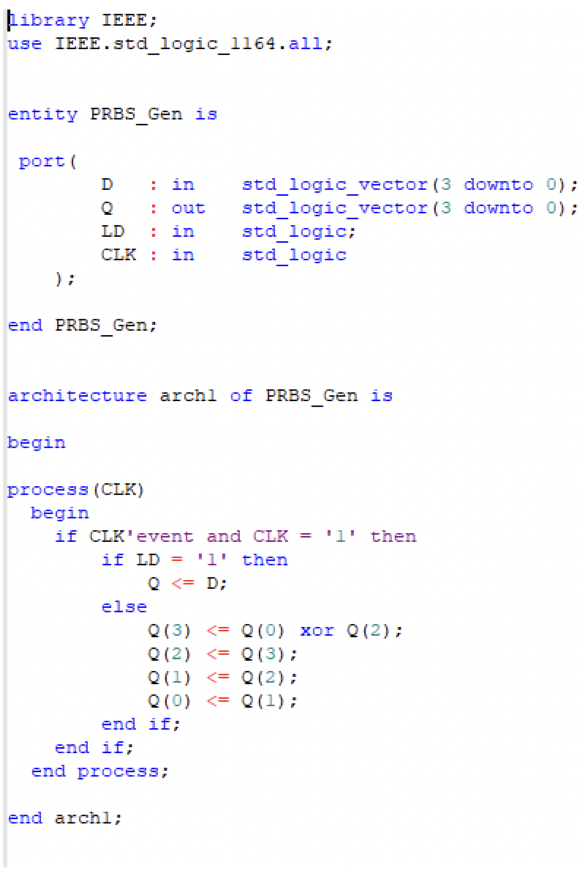
Benny Chen
4/3/2022
Lab 8

Theory

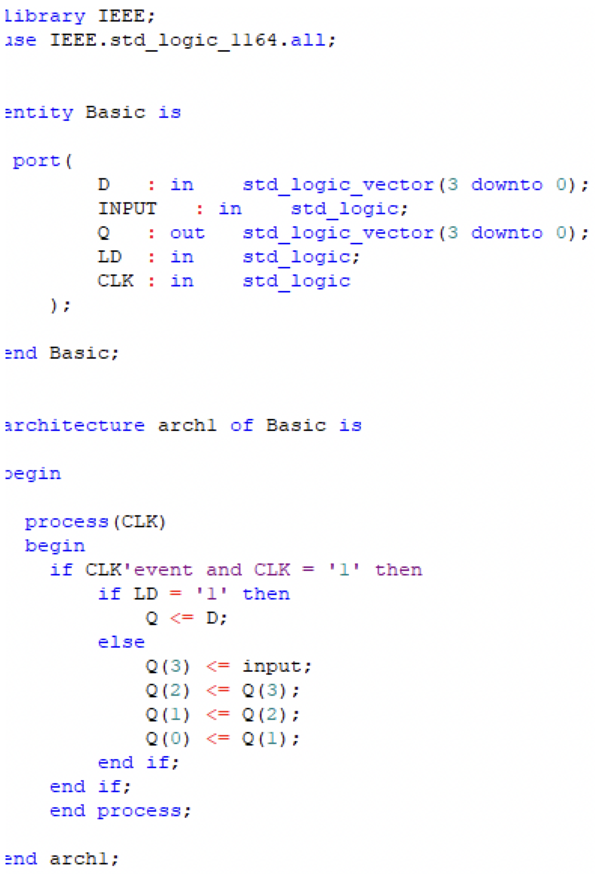
In this lab we created a PRBS generator, or a pseudo random binary sequence generator. A PRBS generator takes in a message then encodes it with a series of bits to output a series of "random" bits which then can be decoded to figure out the message. This can be used to send encrypted messages to places discreetly. Another encrypting method known as OTP or one time pad could be used to encrypt messages, however the encrypted messages would be double the length of the actual message itself. A PRBS on the other hand should have significantly less amount of the encrypted message and would only need the initial PRBS state to encrypt. Processes in VHDl are very important as they handle inputs into the code and trigger functions for it. We use this to take in the clock input to trigger events in the code from the clock.

Deliverables

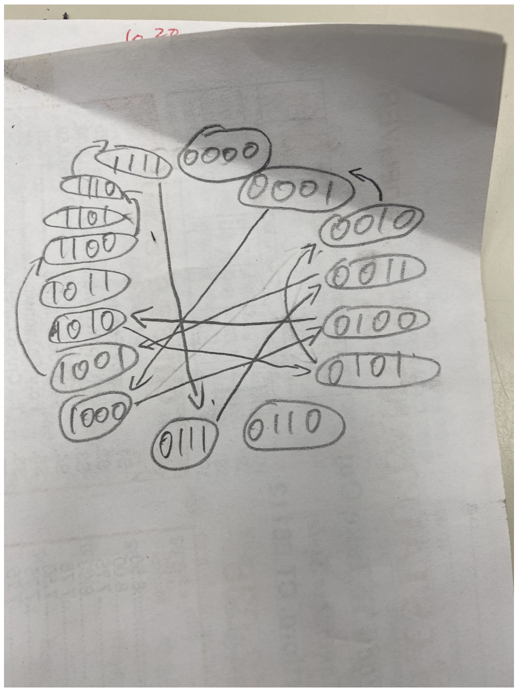




```vhdl
library IEEE;
use IEEE.std_logic_1164.all;


entity PRBS_Gen is

 port(
        D   : in    std_logic_vector(3 downto 0);
        Q   : out   std_logic_vector(3 downto 0);
        LD  : in    std_logic;
        CLK : in    std_logic
    );

end PRBS_Gen;


architecture arch1 of PRBS_Gen is

begin

process(CLK)
  begin
    if CLK'event and CLK = '1' then
        if LD = '1' then
            Q <= D;
        else
            Q(3) <= Q(0) xor Q(2);
            Q(2) <= Q(3);
            Q(1) <= Q(2);
            Q(0) <= Q(1);
        end if;
    end if;
  end process;

end arch1;
```



```vhdl
library IEEE;
se IEEE.std_logic_1164.all;


ntity Basic is

 port(
        D   : in    std_logic_vector(3 downto 0);
        INPUT   : in    std_logic;
        Q   : out   std_logic_vector(3 downto 0);
        LD  : in    std_logic;
        CLK : in    std_logic
    );

nd Basic;


rchitecture arch1 of Basic is

egin

  process(CLK)
  begin
    if CLK'event and CLK = '1' then
        if LD = '1' then
            Q <= D;
        else
            Q(3) <= input;
            Q(2) <= Q(3);
            Q(1) <= Q(2);
            Q(0) <= Q(1);
        end if;
    end if;
    end process;

nd arch1;
```



Discussion

In this lab we created a circuit using VHDL. I learned alot about how a PRBS generator works and how to write VHDL code. I don't really like VHDL though I would still need to learn it as it's similar to MyHDL.

Questions

1. Why can't we include the all-zeroes (0000) state in our PRBS generator? Can you think of design that would allow us to include this state? Address any potential shortcomings your suggestion has?

We can't include an all zero state in out prbs generator as when XORing, the result would be a 0 making the encrypted bit again all zeroes. To solve this problem we can add a NOT gate at some of the bits to have a one of the bits at least be a 1 or instead of XORing, we XNOR instead.

2. Given the state diagram that you created in the deliverables section above, create a transition table which models the diagram accurately.

Current	Next
0001	1000
1000	0100
0100	1010
1010	0101
0101	0010
0010	0001
1110	1111
1100	1110
1001	1100
0011	1001
0111	0011

1111	0111