CSE-2301

Lab 8 PRBS

Gabriel Kinney

Theory

A PRBS stands for pseudo random binary sequence, which is a series of binary numbers that do not have an obvious pattern, and is used for encryption. It works by sending a series of bits through a shift register, with some of the bits being xor with each other and fed back into the input of the register. This creates a semi random sequence that can be used to encode and decode binary information as although apparently random, 2 PRBS will produce the same output sequence given the same initial conditions. This could be used for encrypted communication by sending the encrypted data while having an identical PRBS on the receiving end with an agreed upon initial state.

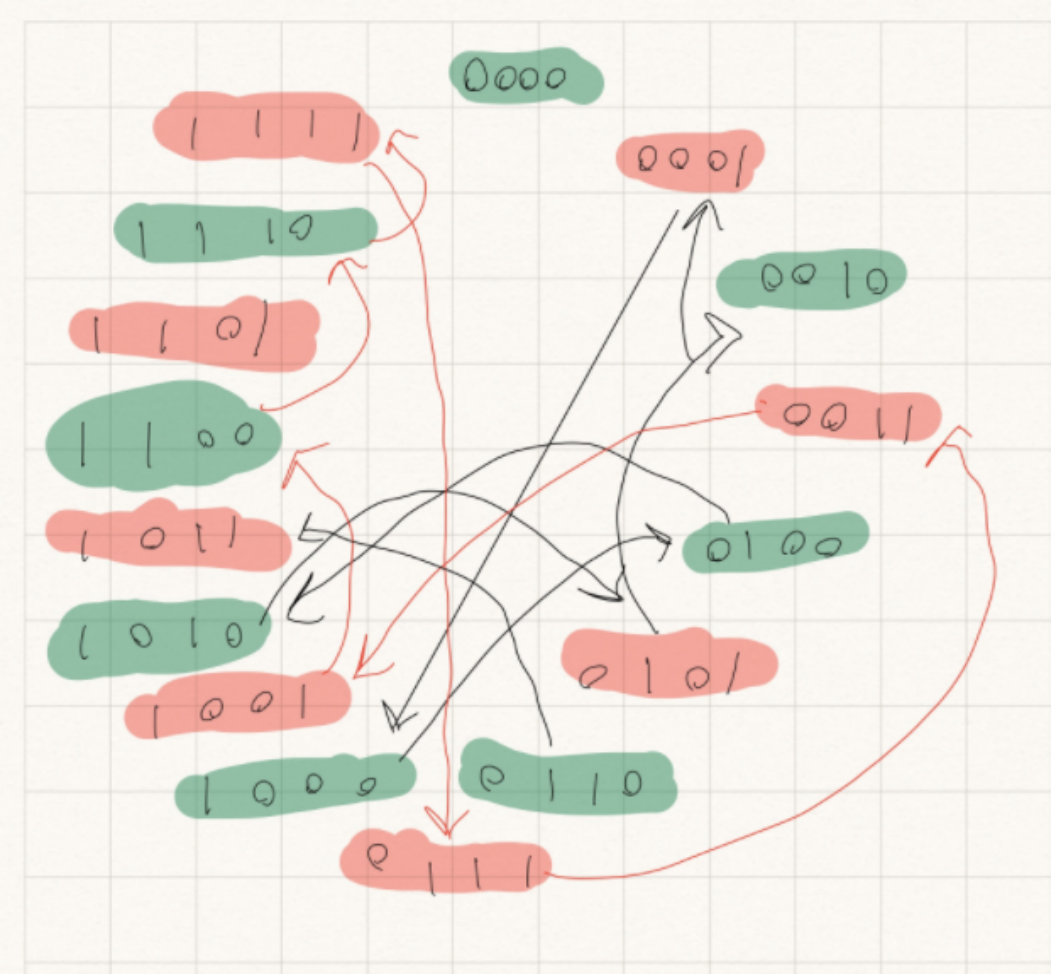
An OTP system must have the bits selected and transmitted prior to the message being transmitted, this would mean that the total size of the transmitted information is more than double the actual message. A PRBS on the other hand does not require significantly more information, and would only require the initial state of the PRBS in addition to then encrypted message bits.

Processes are important because there has to be a defined trigger for the code within the process to input. Without a process it would be impossible to initiate the functionality on the clock of the circuit, and processes allow the clock to be taken as an input.

Deliverables

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


entity Basic_reg_2 is

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

end Basic_reg_2;


architecture arch1 of Basic_reg_2 is

begin

  -- Your VHDL code defining the model goes here
process (CLK, LD)
 begin
 if (CLK'event and CLK='1') then
 Q(0) <= Input;
 for I in 1 to 3 loop
    Q(I) <= Q(I-1);
 end loop;
 end if;
             --30

 if (LD='1') then
 Q(0) <= D(0);
 Q(1) <= D(1);
 Q(2) <= D(2);
 Q(3) <= D(3);
 end if;
 end process;
end arch1;
```

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


entity SRBS 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 SRBS;


architecture arch1 of SRBS is

begin
process (CLK, LD)
 begin

 if (CLK'event and CLK='1') then
 Q(0) <= Q(1);
 Q(1) <= Q(2);
 Q(2) <= Q(3);
 Q(3) <= Q(0) xor Q(2);

 end if;


 if (LD='1') then
 Q(0) <= D(0);
 Q(1) <= D(1);
 Q(2) <= D(2);
 Q(3) <= D(3);
 end if;
 end process;


end arch1;
```



Discussion

In this lab we learned how to use VHDL to mimic and execute the functionality of physical circuits. VHDL is useful for creating complex systems without having to use large amounts of logic gates. We also learned about encryption in the form of PRBS which is a widely used form of encryption.

Practice questions

In the implemented design the all 0 state would not work because as the bits are passed down the register, the results of the XOR would always be 0 meaning the sequence would forever output 0s. One possible solution could be to take the inverse of one of the bits which would always introduce a 1 at some point, however it is possible that this would lead to a more predictable shorter loop of numbers.