Random Number Generator

Digital Circuit Design J Component Project

Slot: G1 & L43+L44

Faculty: Prof. Usha Rani S

Done by: Mrinmay Date(Reg no. 18BIS0147)

Shubhit Talus (Reg no. 17BIS0041)

Parth Aagarwal (Reg no. 17BIS0006)

Random Number Generator (**RNG**): is a device that generates a sequence of numbers or symbols that cannot be reasonably predicted better than by a random chance. Random number generators can be true **hardware random-number generators** (HRNG), which generate genuinely random numbers, or **pseudo-random number generators** (PRNG), which generate numbers that look random, but are actually deterministic, and can be reproduced if the state of the PRNG is known. chance.

Random number generators can be true hardware random-number generators (HRNG), which generate genuinely random numbers, or pseudo-random number generators (PRNG), which generate numbers that look random, but are actually deterministic, and can be reproduced if the state of the PRNG is known.

Code

```
module rand_num_generator
(
   input clk, reset;
   output [3:0] q;
   output [6:0] LED_out;
);

reg [3:0] r_reg;
wire [3:0] r_next;
wire feedback_value;
reg [6:0] LED_out;

always @(posedge clk, posedge reset)
begin
```

```
    if (reset)
        begin
        // set initial value to 1
        r_reg <= 1;
        end
    else if (clk == 1'b1)
        r_reg <= r_next;
end

//// N = 3
//// Feedback polynomial : x^3 + x^2 + 1
////total sequences (maximum) : 2^3 - 1 = 7
assign feedback_value = r_reg[3] ^ r_reg[2] ^ r_reg[0];
//// N = 4
//assign feedback_value = r_reg[4] ^ r_reg[3] ^ r_reg[0];
// N = 5, maximum length = 28 (not 31)
//assign feedback_value = r_reg[5] ^ r_reg[3] ^ r_reg[0];
assign r_next = {feedback_value, r_reg[3:1]};
assign q = r_reg;

always @(*)
```

```
begin
 case(q)
 4'b0000: LED_out = 7'b0000001; // "0"
 4'b0001: LED_out = 7'b1001111; // "1"
 4'b0010: LED_out = 7'b0010010; // "2"
 4'b0011: LED_out = 7'b0000110; // "3"
 4'b0100: LED_out = 7'b1001100; // "4"
 4'b0101: LED_out = 7'b0100100; // "5"
 4'b0110: LED_out = 7'b0100000; // "6"
 4'b0111: LED_out = 7'b0001111; // "7"
 4'b1000: LED_out = 7'b0000000; // "8"
 4'b1001: LED_out = 7'b0000100; // "9"
 default: LED_out = 7'b0000001; // "0"
 endcase
end
endmodule
```

Linear-Feedback Shift Register (LFSR): is a shift register whose input bit is a linear function of its previous state.

The most commonly used linear function of single bits is exclusive-or (XOR). Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value.

The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits that appears random and has a very long cycle.

Applications of LFSRs include generating pseudo-random numbers, pseudo-noise sequences, fast digital counters, and whitening sequences. Both hardware and software implementations of LFSRs are common.

We presented the random number generator using linear feedback shift register. For a single-bit random number generator, LFSR is the most effective method. When multiple bits are required, LFSR can be extended by utilizing extra time or extra circuitry. Cryptographic algorithms and communications protocols are based on random numbers generators.