module primary\_lsfr15 (

input clk,

input reset,

input write,

input pushin,

input [115:0] InitialData15,

output [115:0] rnd1

);

//Linear feedback shift registers

reg [115:0] lfsr15, random\_next1, random\_done1;

//Count for the number of shifts

reg [3:0] count1, count\_next1;

always @ (posedge clk or posedge reset)

begin

if (reset)

begin

lfsr15 <= #1 0;

//case1

//lfsr1 <= 185'h4751245563371bb82b2b5aacd05678a1b17e06c62eb0dace; //An LFSR cannot have an all 0 state, thus reset to 4751245563371bb82b2b5aacd05678a1b17e06c62eb0dace

end

else

begin

if (write)

begin

lfsr15 <= InitialData15;

//case2

//lfsr1 <= 185'h08AAC66E37215874F559A0ACF14362FC0D24CD61E1D5512;

count1 <= 0;

end

else if (pushin)

begin

lfsr15 <= #1 random\_next1;

count1 <= #1 count\_next1;

end

end

end

always @ (\*)

begin

//-----------Combinational code for shift register 1 --> 13 bits ----------//

random\_next1 = lfsr15; //default state stays the same

count\_next1 = count1;

random\_done1 = 0;

random\_next1 = { (lfsr15[105:95]), (lfsr15[115]^lfsr15[94]) ,(lfsr15[114]^lfsr15[93]) ,(lfsr15[113]^lfsr15[92]) ,(lfsr15[112]^lfsr15[91]) ,(lfsr15[111]^lfsr15[90]) ,

(lfsr15[110]^lfsr15[89]) ,(lfsr15[109]^lfsr15[88]) ,(lfsr15[108]^lfsr15[87]) ,(lfsr15[107]^lfsr15[86]) ,(lfsr15[106]^lfsr15[85]) ,

(lfsr15[84:27]), (lfsr15[115]^lfsr15[26]) ,(lfsr15[114]^lfsr15[25]) ,(lfsr15[113]^lfsr15[24]) ,(lfsr15[112]^lfsr15[115]^lfsr15[23]),

(lfsr15[111]^lfsr15[114]^lfsr15[22]), (lfsr15[110]^lfsr15[113]^lfsr15[21]), (lfsr15[109]^lfsr15[112]^lfsr15[20]), (lfsr15[108]^lfsr15[111]^lfsr15[19]),(lfsr15[107]^lfsr15[110]^lfsr15[18]),

(lfsr15[106]^lfsr15[109]^lfsr15[17]), (lfsr15[108]^lfsr15[16]) ,(lfsr15[107]^lfsr15[15]) ,(lfsr15[106]^lfsr15[14]) , (lfsr15[13:0]), (lfsr15[115:106]) };

count\_next1 = count1 + 1;

if (count1 == 1)

begin

count1 = 0;

random\_done1 = lfsr15; //assign the random number to output after 13 shifts

end

//--------------------------------------------End of combination logic for shift register 1----------------------------------//

end

assign rnd1 = lfsr15;

endmodule