module primary\_lsfr7 (

input clk,

input reset,

input write,

input pushin,

input [211:0] InitialData7,

output [211:0] rnd1

);

//Linear feedback shift registers

reg [211:0] lfsr7, 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

lfsr7 <= #1 0;

//case1

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

end

else

begin

if (write)

begin

lfsr7 <= InitialData7;

//case2

//lfsr1 <= 185'h08AAC66E37215874F559A0ACF14362FC0D24CD61E1D5512;

count1 <= 0;

end

else if (pushin)

begin

lfsr7 <= #1 random\_next1;

count1 <= #1 count\_next1;

end

end

end

always @ (\*)

begin

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

random\_next1 = lfsr7; //default state stays the same

count\_next1 = count1;

random\_done1 = 0;

random\_next1 = { (lfsr7[198:176]), (lfsr7[211]^lfsr7[175]) ,(lfsr7[210]^lfsr7[174]) ,(lfsr7[209]^lfsr7[173]) ,(lfsr7[208]^lfsr7[172]) ,(lfsr7[207]^lfsr7[171]) ,

(lfsr7[206]^lfsr7[170]) ,(lfsr7[205]^lfsr7[169]) ,(lfsr7[204]^lfsr7[168]) ,(lfsr7[203]^lfsr7[167]) ,(lfsr7[202]^lfsr7[166]) ,

(lfsr7[201]^lfsr7[165]) ,(lfsr7[200]^lfsr7[164]) ,(lfsr7[199]^lfsr7[163]) , (lfsr7[162:36]), (lfsr7[211]^lfsr7[35]) ,(lfsr7[210]^lfsr7[34]) ,

(lfsr7[209]^lfsr7[33]) ,(lfsr7[208]^lfsr7[32]) ,(lfsr7[207]^lfsr7[31]) ,(lfsr7[206]^lfsr7[30]) ,(lfsr7[205]^lfsr7[29]) ,

(lfsr7[204]^lfsr7[211]^lfsr7[28]), (lfsr7[203]^lfsr7[210]^lfsr7[27]), (lfsr7[202]^lfsr7[209]^lfsr7[26]), (lfsr7[201]^lfsr7[208]^lfsr7[25]),

(lfsr7[200]^lfsr7[207]^lfsr7[24]), (lfsr7[199]^lfsr7[206]^lfsr7[23]),

(lfsr7[205]^lfsr7[22]) ,(lfsr7[204]^lfsr7[21]) ,(lfsr7[203]^lfsr7[20]) ,(lfsr7[202]^lfsr7[19]) ,(lfsr7[201]^lfsr7[18]) ,

(lfsr7[200]^lfsr7[17]) ,(lfsr7[199]^lfsr7[16]) , (lfsr7[15:0]), (lfsr7[211:199]) };

count\_next1 = count1 + 1;

if (count1 == 1)

begin

count1 = 0;

random\_done1 = lfsr7; //assign the random number to output after 13 shifts

end

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

end

assign rnd1 = lfsr7;

endmodule