module dataSelectLFSR (

input clk,

input reset,

input write,

input pushin,

input [6:0] data,

input [63:0] initialData,

output [63:0] rnd1

);

//Linear feedback shift registers

reg [63:0] dataSelect, 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

dataSelect <= #1 0;

//case1

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

end

else

begin

if (write)

begin

dataSelect <= #1 initialData;

//case2

//lfsr1 <= 185'h08AAC66E37215874F559A0ACF14362FC0D24CD61E1D5512;

count1 <= #1 0;

end

else if (pushin)

begin

dataSelect <= #1 random\_next1;

count1 <= #1 count\_next1;

end

end

end

always @ (\*)

begin

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

random\_next1 = #0 dataSelect; //default state stays the same

count\_next1 = #0 count1;

random\_done1 = #0 0;

random\_next1 = #0 { (dataSelect[56]^dataSelect[63]^dataSelect[58]^dataSelect[60]^dataSelect[62]) ,

(dataSelect[55]^dataSelect[62]^dataSelect[57]^dataSelect[59]^dataSelect[61]^dataSelect[63]) ,

(dataSelect[54]^dataSelect[63]^dataSelect[61]^dataSelect[63]) ,(dataSelect[53]^dataSelect[63]^dataSelect[60]) ,

(dataSelect[52]^dataSelect[62]^dataSelect[59]^dataSelect[63]) ,(dataSelect[51]^dataSelect[61]^dataSelect[58]^dataSelect[62]) ,

(dataSelect[50]^dataSelect[60]^dataSelect[57]^dataSelect[61]^dataSelect[63]) ,(dataSelect[49]^dataSelect[59]^dataSelect[58]) ,

(dataSelect[48]^dataSelect[58]^dataSelect[57]) ,(dataSelect[47]^dataSelect[57]^dataSelect[58]^dataSelect[60]^dataSelect[62]) ,

(dataSelect[46]^dataSelect[61]^dataSelect[58]^dataSelect[60]^dataSelect[62]^dataSelect[57]^dataSelect[59]) ,

(dataSelect[45]^dataSelect[61]^dataSelect[62]^dataSelect[57]^dataSelect[59]) ,(dataSelect[44]^dataSelect[61]^dataSelect[62]) ,

(dataSelect[43]^dataSelect[60]^dataSelect[61]^dataSelect[63]) ,(dataSelect[42]^dataSelect[59]^dataSelect[60]^dataSelect[62]) ,

(dataSelect[41]^dataSelect[58]^dataSelect[59]^dataSelect[61]^dataSelect[63]) ,

(dataSelect[40]^dataSelect[57]^dataSelect[58]^dataSelect[60]^dataSelect[62]) ,

(dataSelect[39]^dataSelect[58]^dataSelect[60]^dataSelect[62]^dataSelect[57]^dataSelect[59]^dataSelect[61]) ,

(dataSelect[38]^dataSelect[61]^dataSelect[62]^dataSelect[57]^dataSelect[59]) ,(dataSelect[37]^dataSelect[61]^dataSelect[62]) ,

(dataSelect[36]^dataSelect[61]^dataSelect[60]) ,(dataSelect[35]^dataSelect[60]^dataSelect[59]) ,

(dataSelect[34]^dataSelect[63]^dataSelect[58]^dataSelect[59]) ,(dataSelect[33]^dataSelect[58]^dataSelect[62]^dataSelect[57]) ,

(dataSelect[32]^dataSelect[57]^dataSelect[61]^dataSelect[58]^dataSelect[62]^dataSelect[60]) ,

(dataSelect[31]^dataSelect[61]^dataSelect[57]^dataSelect[59]^dataSelect[58]^dataSelect[62]) ,

(dataSelect[30]^dataSelect[61]^dataSelect[62]^dataSelect[57]) ,

(dataSelect[29]^dataSelect[62]^dataSelect[61]^dataSelect[63]^dataSelect[58]) ,

(dataSelect[28]^dataSelect[61]^dataSelect[60]^dataSelect[62]^dataSelect[57]) ,

(dataSelect[27]^dataSelect[62]^dataSelect[59]^dataSelect[61]^dataSelect[63]^dataSelect[58]) ,

(dataSelect[26]^dataSelect[57]^dataSelect[58]^dataSelect[60]^dataSelect[62]^dataSelect[61]) ,

(dataSelect[25]^dataSelect[58]^dataSelect[62]^dataSelect[57]^dataSelect[59]^dataSelect[61]^dataSelect[63]) ,

(dataSelect[24]^dataSelect[57]^dataSelect[61]^dataSelect[63]) ,(dataSelect[23]^dataSelect[63]^dataSelect[58]) ,

(dataSelect[22]^dataSelect[63]^dataSelect[62]^dataSelect[57]) ,(dataSelect[21]^dataSelect[61]^dataSelect[60]^dataSelect[58]) ,

(dataSelect[20]^dataSelect[60]^dataSelect[57]^dataSelect[59]^dataSelect[63]) ,(dataSelect[19]^dataSelect[60]^dataSelect[59]) ,

(dataSelect[18]^dataSelect[58]^dataSelect[59]^dataSelect[63]) ,(dataSelect[17]^dataSelect[62]^dataSelect[57]^dataSelect[58]) ,

(dataSelect[16]^dataSelect[61]^dataSelect[57]^dataSelect[58]^dataSelect[60]^dataSelect[62]) ,

(dataSelect[15]^dataSelect[58]^dataSelect[62]^dataSelect[57]^dataSelect[59]^dataSelect[61]^dataSelect[63]) ,

(dataSelect[14]^dataSelect[57]^dataSelect[61]^dataSelect[63]) ,(dataSelect[13]^dataSelect[58]) ,

(dataSelect[12]^dataSelect[63]^dataSelect[57]) ,(dataSelect[11]^dataSelect[63]^dataSelect[58]^dataSelect[60]) ,

(dataSelect[10]^dataSelect[62]^dataSelect[57]^dataSelect[59]) ,(dataSelect[09]^dataSelect[61]^dataSelect[60]^dataSelect[62]) ,

(dataSelect[08]^dataSelect[61]^dataSelect[60]^dataSelect[59]) ,(dataSelect[07]^dataSelect[60]^dataSelect[59]^dataSelect[58]) ,

(dataSelect[06]^dataSelect[61]^dataSelect[58]^dataSelect[57]^dataSelect[59]^dataSelect[61]^dataSelect[63]) ,

(dataSelect[05]^dataSelect[60]^dataSelect[57]) ,(dataSelect[04]^dataSelect[59]^dataSelect[58]^dataSelect[60]^dataSelect[62]) ,

(dataSelect[03]^dataSelect[61]^dataSelect[58]^dataSelect[57]^dataSelect[59]) ,(dataSelect[02]^dataSelect[62]^dataSelect[57]) ,

(dataSelect[01]^dataSelect[61]^dataSelect[63]^dataSelect[58]^dataSelect[60]^dataSelect[62]) ,

(dataSelect[00]^dataSelect[61]^dataSelect[60]^dataSelect[62]^dataSelect[57]^dataSelect[59]) ,

(dataSelect[61]^dataSelect[62]^dataSelect[59]^data[0]) ,

(dataSelect[58]^dataSelect[60]^dataSelect[61]^dataSelect[63]^data[1]) ,

(dataSelect[57]^dataSelect[59]^dataSelect[60]^dataSelect[62]^data[2]) ,

(dataSelect[60]^dataSelect[62]^data[3]^dataSelect[59]^dataSelect[61]^dataSelect[63]) ,

(dataSelect[59]^dataSelect[61]^dataSelect[63]^data[4]^dataSelect[58]^dataSelect[60]^dataSelect[62]) ,

(dataSelect[58]^dataSelect[60]^dataSelect[62]^data[5]^dataSelect[57]^dataSelect[59]^dataSelect[61]^dataSelect[63]) ,

(dataSelect[57]^dataSelect[59]^dataSelect[61]^dataSelect[63]^data[6]) };

count\_next1 = #0 count1 + 1;

if (count1 == 1)

begin

count1 = #0 0;

random\_done1 = #0 dataSelect; //assign the random number to output after 13 shifts

end

else

begin

//random\_done1 = random\_done1;

//dataSelect = dataSelect;

end

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

end

assign rnd1 = dataSelect;

endmodule