// reading MSD and CRC data from serial input

module read\_MSD\_input(

input clk ,

input rst ,

input ack ,

input in\_MSD\_CRC ,

output done ,

output [1147:0] MSD\_CRC

);

reg [1147:0] MSD\_CRC\_shift;

reg [10:0] shift\_cnt;

// generating done pulse and bulding MSD\_CRC data

assign MSD\_CRC = MSD\_CRC\_shift ;

assign done = (shift\_cnt == 1148) ;

// collecting inputs bit by bit through shift regisers

always @(posedge clk or posedge rst)

begin

if (rst)

MSD\_CRC\_shift <= 1148'd0 ;

else if( ack || (shift\_cnt < 1148 ) )

MSD\_CRC\_shift <= {MSD\_CRC\_shift[1146:0],in\_MSD\_CRC} ;

end

// making count of every shift

always @(posedge clk or posedge rst)

begin

if (rst)

shift\_cnt <= 11'd0;

else if ( ack )

shift\_cnt <= 11'd1;

else if (shift\_cnt != 0 && shift\_cnt <= 1148 )

shift\_cnt <= shift\_cnt + 1 ;

end

endmodule

module process\_parity1(

input clk ,

input rst ,

input mode ,

input done\_rd\_MSD ,

input [1147:0] MSD\_CRC ,

input ack ,

input in\_MSD\_CRC ,

output done\_p1 ,

output [2:0] PTAIL\_1 ,

output [2:0] TAIL\_1 ,

output [1147:0] PARITY\_1

);

reg [10:0] shift\_cnt;

reg [2:0] parity1\_shift\_reg;

reg [1147:0] parity1\_reg;

reg [2:0] ptail;

reg [2:0] tail ;

wire start\_shift;

wire in;

// mode 1:parallel 0: serial

assign start\_shift = mode ? ack : done\_rd\_MSD ;

assign done\_p1 = (shift\_cnt == 1151);

assign PTAIL\_1 = ptail;

assign TAIL\_1 = tail ;

assign PARITY\_1 = parity1\_reg;

assign in = mode ? in\_MSD\_CRC :

done\_rd\_MSD ? MSD\_CRC[0] :

(shift\_cnt < 1148)? MSD\_CRC[shift\_cnt]: parity1\_shift\_reg[2] ;

always @(posedge clk or posedge rst)

begin

if (rst)

parity1\_shift\_reg <= 3'd0;

else

parity1\_shift\_reg <= {parity1\_shift\_reg[1],parity1\_shift\_reg[0],(parity1\_shift\_reg[2]^parity1\_shift\_reg[1]^ in) };

end

always @(posedge clk or posedge rst)

begin

if (rst)

parity1\_reg <= 1148'd0;

else if( shift\_cnt < 1148 )

parity1\_reg <= {parity1\_reg[1146:0],(parity1\_shift\_reg[2]^parity1\_shift\_reg[1]^parity1\_shift\_reg[0]^in)};

end

always @(posedge clk or posedge rst)

begin

if (rst) begin

ptail <= 3'd0;

tail <= 3'd0;

end

else if( shift\_cnt >= 1148 && shift\_cnt < 1151 ) begin

ptail <= {ptail[1:0],(parity1\_shift\_reg[2]^parity1\_shift\_reg[1]^parity1\_shift\_reg[0]^in)};

tail <= {tail[1:0],in};

end

end

// making count of every shift

always @(posedge clk or posedge rst)

begin

if (rst)

shift\_cnt <= 11'd0;

else if (start\_shift)

shift\_cnt <= 11'd1;

else if (shift\_cnt != 0 && shift\_cnt <= (1148 + 3) )

shift\_cnt <= shift\_cnt + 1 ;

end

endmodule