Electronics Club

Task-4

Chirag Hegde

Design.v:

```
module transmissionBlock(
input clk,
input data, start,
output reg [95:0] fec,
output reg done,
output reg [6:0] cycleCounter
);
  reg [5:0] count=0;
  reg [3:0] shift;
  reg [15:0] r,temp;

  always @(posedge clk, posedge start) begin
    cycleCounter=cycleCounter+1;
    if(start) begin
      done=0;
      r=16'hFFFF;
      fec=96'd0;
      count=49;
      cycleCounter=-1;
      shift=4'b0000;
    end

    else if(count>17) begin
      //CRC
      r[15] <= r[14]+r[15]+data;
      r[14:3] <= r[13:2];
      r[2] <= r[1]+data+r[15];
      r[1] <= r[0];
      r[0] <= data+r[15];

      //FEC
      shift[3:1] <= shift[2:0];
      shift[0]<=data;

      fec[0] <= shift[0]+shift[1]+shift[2]+shift[3];
      fec[1] <= shift[0]+shift[2]+shift[3];
      fec[95:2]<=fec[93:0];

      count<=count-1;

    end
```

```
    else if (count<=17 && count!=0) begin
      temp=r[15];
      shift[3:1] <= shift[2:0];
      shift[0]<=temp;

      fec[0] <= shift[0]+shift[1]+shift[2]+shift[3];
      fec[1] <= shift[0]+shift[2]+shift[3];
      fec[95:2]<=fec[93:0];
      r[15:1]<=r[14:0];
      count<=count-1;
      if(count==1) begin
      done<=1;
      end
    end
  end
endmodule

module interleavingModule(
  input [95:0] fec,
  output [95:0] int_fec
);
       fourByteInterleaver T1(fec[31:0],int_fec[31:0]);
       fourByteInterleaver T2(fec[63:32],int_fec[63:32]);
       fourByteInterleaver T3(fec[95:64],int_fec[95:64]);
endmodule

module fourByteInterleaver(
  input  [31:0] inFourBytes,
  output  [31:0] outFourBytes
);
  wire [7:0] byte0;
  wire [7:0] byte1;
  wire [7:0] byte2;
  wire [7:0] byte3;
  wire [7:0] out0;
  wire [7:0] out1;
  wire [7:0] out2;
  wire [7:0] out3;

  assign byte0 = inFourBytes[7:0];
  assign byte1 = inFourBytes[15:8];
  assign byte2 = inFourBytes[23:16];
  assign byte3 = inFourBytes[31:24];
  assign out3 = {byte0[1:0],byte1[1:0],byte2[1:0],byte3[1:0]};
  assign out2 = {byte0[3:2],byte1[3:2],byte2[3:2],byte3[3:2]};
  assign out1 = {byte0[5:4],byte1[5:4],byte2[5:4],byte3[5:4]};
  assign out0 = {byte0[7:6],byte1[7:6],byte2[7:6],byte3[7:6]};
  assign outFourBytes = {out3,out2,out1,out0};

endmodule
```

Testbench.v:

```
module t_transmissionBlock;
  reg clk;
  reg data, start;
  wire [95:0] fec;
  wire [95:0] interleaved_fec;
  wire done;
  wire [6:0] cycleCounter;
  wire count;

  reg [31:0] input_data = 32'h03_01_02_03;

  integer k;
  initial  forever #5 clk = ~clk ;

  transmissionBlock M1
(.clk(clk),.data(data),.start(start),.fec(fec),.done(done),.cycleCounter(cycleCoun
ter));

  interleavingModule M2(.fec(fec),.int_fec(interleaved_fec));

  initial begin
    clk=0;
    start=0;
    #1;
    start=1;
    #4 start=0;
    //#10;

    data=input_data[31];
    for(k=0;k<32;k=k+1)
      begin
        #10;
        input_data = input_data<<1;
        data=input_data[31];
      end
    end

  initial begin
    @(posedge done) begin
      #10 $finish;
    end
  end
```

```
initial begin
     $monitor("%h %h %d",fec,interleaved_fec,cycleCounter);
  end
endmodule
```

Waveforms: