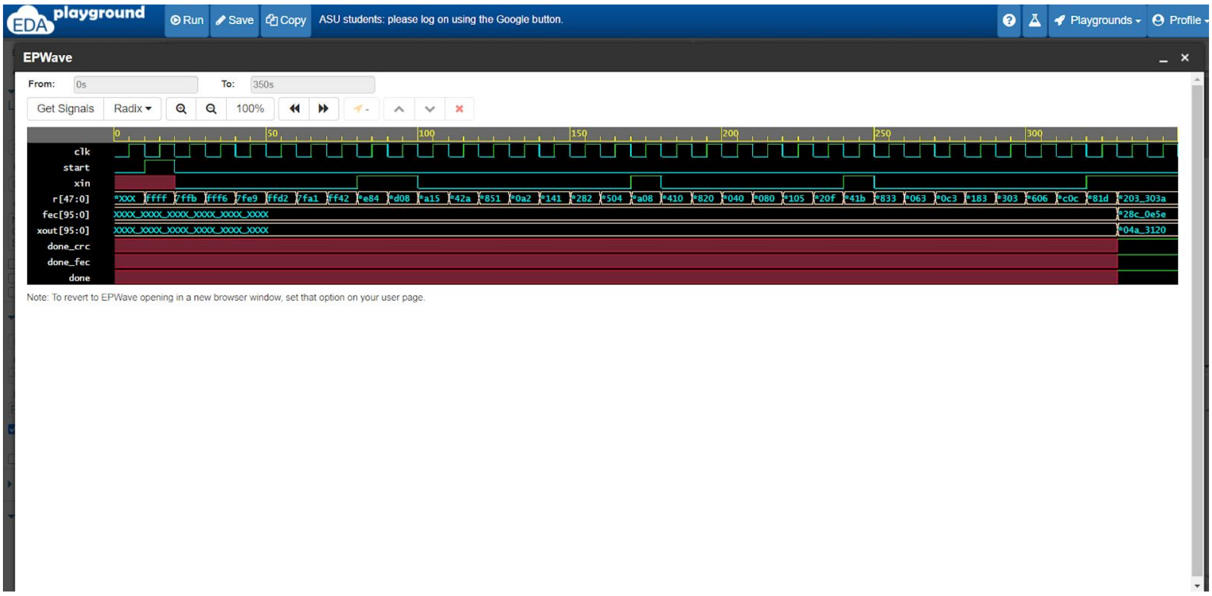
```
//CODE
//DESIGN
//32 clock cycles
module dcs(
  input start,xin,clk,
  output done,
  output [95:0]xout);
  wire done_fec,done_crc;
  wire [47:0]r;
  wire [95:0]fec;
  crc crc1(clk,start,xin,done_crc,r);
  fec fec1(done_crc,r,done_fec,fec);
  interleaver interleaver1(done_fec,fec,done,xout);
endmodule
module crc(
  input clk,start,x16,
  output reg done,
  output reg [47:0]r);
  reg state;
  integer count;
  parameter active=1,idle=0;
  always @(negedge clk) begin
    if(start==1) begin
      state<=active;r<={32'h0,16'hffff};count<=1;
    end
    if(done==1) begin
      state<=idle;
    end
  end
  always @(negedge clk) begin
    case(state)
      active: begin
        if(count<=32) begin
          r[47:16]<={r[47:16],x16};
          r[0]<=r[15]^x16;
          r[1]<=r[0];
          r[2]<=r[1]^r[15]^x16;
          r[14:3]<=r[13:2];
          r[15]<=r[14]^r[15]^x16;
          count<=count+1;
        end
      end
    endcase
    if(count==32) begin
      done<=1;
    end
  end
endmodule
```

```
module fec(
  input start,
  input [47:0]x,
  output reg done,
  output reg [95:0]fec);
  integer i;
  reg [47:0]P1,P0;
  always @(*) begin
    if(start) begin
      for(i=47;i>=0;i=i-1) begin
         if(i==47) begin
           P0[i]=x[i]; P1[i]=x[i];
           fec={fec,x[i],x[i]};
         end
        else if(i==46) begin
          P0[i]=x[i]^x[i+1]; P1[i]=x[i];
        end
        else if(i==45) begin
          P0[i]=x[i]^x[i+1]^x[i+2];          P1[i]=x[i]^x[i+2];
        end
        else begin
          P0[i]=x[i]^x[i+1]^x[i+2]^x[i+3];
          P1[i]=x[i]^x[i+2]^x[i+3];
        end
        fec={fec,P1[i],P0[i]};
      end
      done<=1;
    end
  end
endmodule
module interleaver(
  input start,
  input [95:0]fec,
  output reg done,
  output reg [95:0]xout);
  integer j,k,l;
  always @(*) begin
    k=95;
    for(j=0;j<4;j=j+1) begin
      for(l=j*2;l<95;l=l+8) begin
        xout[k]=fec[l+1];
        xout[k-1]=fec[l];
        k=k-2;
      end
    end
    done<=1;
  end
endmodule
```

```
//testbench
module dcs_test;
  reg clk,start,x;
  wire done;
  wire [95:0]out;
  dcs dcs1(start,x,clk,done,out);
   initial begin
    clk=0;
    forever #5 clk=~clk;
  end
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars(0,dcs1);
  end
  initial fork
    start=0;
    #10 start=1;
    #20 start=0;
    #20 x=0;
    #80 x=1;
    #100 x=0;
    #170 x=1;
    #180 x=0;
    #240 x=1;
    #250 x=0;
    #320 x=1;
    #350 $finish;
    $monitor("xout=%96h",out);
  join
endmodule
```

//WAVEFORM



ASU students: please log on using the Google button.

From: 0s To: 350s

Note: To revert to EPWave opening in a new browser window, set that option on your user page.

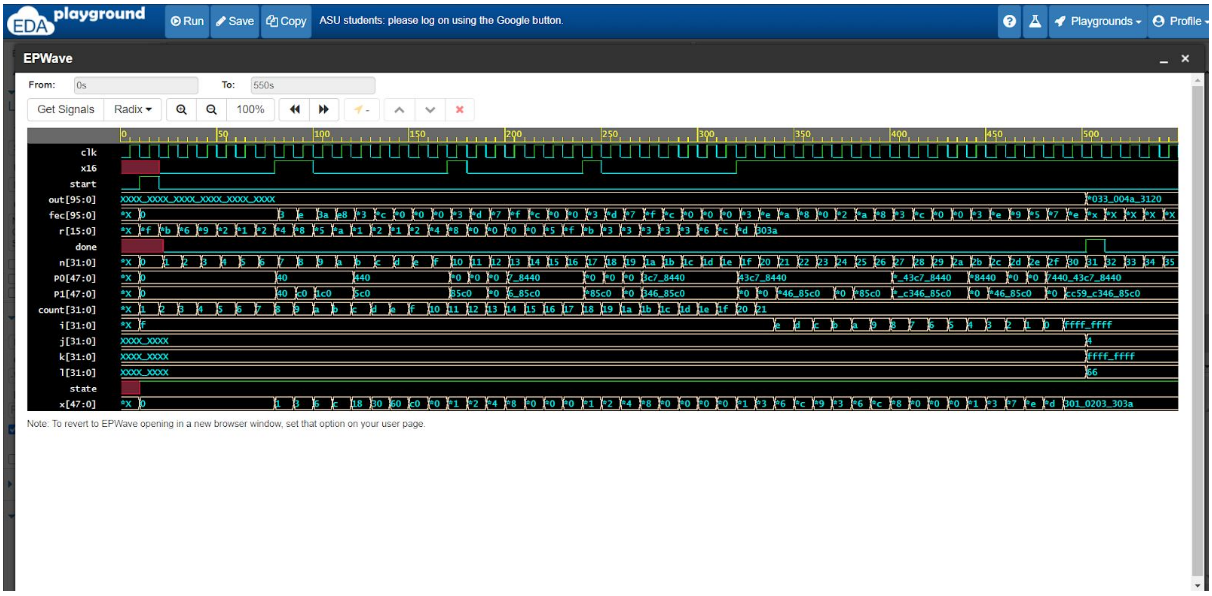
```
//48 clock cycles
//design
module DCS(
  input start,clk,x16,
  output reg done,
  output reg [95:0]out);
  integer count,n,i,j,k,l;
  reg [15:0]r;
  reg [95:0]fec;
  reg [47:0]x;
  reg [47:0]P1,P0;
  reg state;
  parameter idle=0,active=1;
  always @(negedge clk) begin
    if(start==1) begin
      state<=active;
      r<=16'hffff; count<=1;x<=0; fec<=0; P1<=0; P0<=0; n<=0;i<=15;
    end
  end
  always @(negedge clk) begin
    case(state)
      active: begin//CRC
        if(count<=32) begin
        r[0]<=r[15]+x16;
        r[1]<=r[0];
        r[2]<=r[1]+r[15]+x16;
        r[14:3]<=r[13:2];
        r[15]<=r[14]+r[15]+x16;
        count<=count+1;
        end
        if(count<=32) begin//FEC
          x<={x,x16};
        end
        else begin
          if(i>=0) begin
            x<={x,r[i]};
            i<=i-1;
          end
        end
        #1P0[n]<=x[0]^x[1]^x[2]^x[3];
        P1[n]<=x[0]^x[2]^x[3];
        #1fec<={fec,P1[n],P0[n]};
        n<=n+1;
        done<=(n==47);
        if(n==48) begin//INTERLEAVER
          k=95;
          for(j=0;j<4;j=j+1) begin
            for(l=j*2;l<95;l=l+8) begin
```

```
              out[k]=fec[l+1];
              out[k-1]=fec[l];
              k=k-2;
            end
          end
        end
      end
      idle: begin
        out<=0; r<=0;
      end
    endcase
  end
endmodule
```

//testbench

```
module test;
  reg clk,start,x;
  wire done;
  wire [95:0]out;
  DCS dcs1(start,clk,x,done,out);
  initial begin
    clk=0;
    forever #5 clk=~clk;
  end
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars(0,dcs1);
  end
  initial fork
    start=0;
    #10 start=1;
    #20 start=0;
    #20 x=0;
    #80 x=1;
    #100 x=0;
    #170 x=1;
    #180 x=0;
    #240 x=1;
    #250 x=0;
    #320 x=1;
    #550 $finish;
    #492 $monitor($time,"%96h %96h",dcs1.fec,out);
  join
endmodule
```

//waveform



ASU students: please log on using the Google button.

From: 0s To: 550s

Note: To revert to EPWave opening in a new browser window, set that option on your user page.