HAMMING CODE USING VERILOG HDL

|  |  |
| --- | --- |
| **TESTBENCH:** | **DESIGN**: |
| module hamming\_tb();  reg [3:0] d;  wire [6:0] y;  reg [0:6] c;  wire [0:2]s;  wire [0:3]d1;  reg [0:6] c1;  hamming\_encoder a1(d,y);  initial  begin  #0d=4'b0011;  $monitor("The Data Given=%b",d,,"\nThe Encoded Data is=%b",y);  end  hamming\_decoder a2(.d1(d1),.s(s),.y(c),.c1(c1));  initial  c1=7'b1011010;  initial  begin  #5 $monitor("\nThe Data Given For Decoding=%b\n",c1,"\nNow by Comparing with the Encoded Data:");  end  initial  begin  #10 c=y;  end  always@ (c)  begin  if(c1==c)  begin  #1 $monitor("The Encoded Data is Right");  end  else  begin  $monitor("The Encoded Data is Wrong\n\nHence the correct ENCODED Data is:%b",c);  end  end  initial  begin  #10 $monitor("\nData bits=%b,Encoded data=%b",d,y);  #10$monitor("\nAFTER DECODING,\nParity Bits=%b\nData Bits(Decoded)=%b",s,d1);  end  initial  #20$finish;    endmodule | module hamming(d,y,c,s,d1);    input [3:0] d;  output [6:0] y;  input [0:6] c;  output [0:2]s;  output [0:3]d1;    wire [0:2]s;  wire [0:3]d1;  reg [0:6] c1;    hamming\_encoder a1(d,y);  hamming\_decoder a2(.d1(d1),.s(s),.y(c),.c1(c1));  endmodule  module hamming\_encoder(d,y);    input [3:0] d;  output [6:0] y;    assign y[0]=d[0];  assign y[1]=d[1];  assign y[2]=d[2];  assign y[3]=d[2]^d[1]^d[0];  assign y[4]=d[3];  assign y[5]=d[3]^d[1]^d[0];  assign y[6]=d[3]^d[2]^d[0];  endmodule    module hamming\_decoder(.d1(d1),.s(s),.y(c),.c1(c1));  input [0:6] c;  output [0:2]s;  output [0:3]d1;  input [0:6] c1;  wire [0:2]s;  wire [0:3]d1;  wire [0:6] c1;    assign s[0] = c[0];  assign s[1] = c[1];  assign s[2] = c[3];  assign d1[0] = c[2];  assign d1[1] = c[4];  assign d1[2] = c[5];  assign d1[3] = c[6];  endmodule |