

We are generating a 5-to-32 decoder hierarchically in Verilog. We have generated 2-to-4 decoder, 3-to-8 decoder defined as follows:

```
module decoder2_4(  
  input [1:0] in,  
  output [3:0] out  
);  
  
  reg [3:0] out;  
  
  always @(in) begin  
    case (in)  
      2'b00: out = 4'b0001;  
      2'b01: out = 4'b0010;  
      2'b10: out = 4'b0100;  
      2'b11: out = 4'b1000;  
    endcase  
  end  
  
endmodule  
  
module decoder3_8(  
  input [2:0] in,  
  output [7:0] out  
);  
  
  wire [3:0] out_low;  
  wire [3:0] out_high;  
  
  decoder2_4 decoder_low (  
    .in(in[1:0]),  
    .out(out_low)  
  );  
  
  decoder2_4 decoder_high (  
    .in(in[1:0]),  
    .out(out_high)  
  );  
  
  assign out[3:0] = (in[2] == 0) ? out_low : 4'b0000;  
  assign out[7:4] = (in[2] == 1) ? out_high : 4'b0000;
```

endmodule

Please include the previous module(s) in your response and use them to hierarchically generate a 5-to-32 decoder defined as:

```
module decoder5_32(in[4:0], out[31:0]);  
//Insert code here  
endmodule
```

```
module decoder2_4(  
    input [1:0] in,  
    output [3:0] out  
);  
  
    reg [3:0] out;  
  
    always @(in) begin  
        case (in)  
            2'b00: out = 4'b0001;  
            2'b01: out = 4'b0010;  
            2'b10: out = 4'b0100;  
            2'b11: out = 4'b1000;  
        endcase  
    end  
endmodule  
  
module decoder3_8(  
    input [2:0] in,  
    output [7:0] out  
);  
  
    wire [3:0] out_low;  
    wire [3:0] out_high;  
  
    decoder2_4 decoder_low (  
        .in(in[1:0]),  
        .out(out_low)  
    );  
  
    decoder2_4 decoder_high (  
        .in(in[1:0]),  
        .out(out_high)  
    );  
  
    assign out[3:0] = (in[2] == 0) ? out_low : 4'b0000;  
    assign out[7:4] = (in[2] == 1) ? out_high : 4'b0000;  
endmodule  
  
module decoder5_32(  
    input [4:0] in,  
    output [31:0] out  
);  
  
    wire [7:0] out0, out1, out2, out3;
```

```

decoder3_8 decoder0 (
    .in(in[2:0]),
    .out(out0)
);

decoder3_8 decoder1 (
    .in(in[2:0]),
    .out(out1)
);

decoder3_8 decoder2 (
    .in(in[2:0]),
    .out(out2)
);

decoder3_8 decoder3 (
    .in(in[2:0]),
    .out(out3)
);

assign out[7:0]    = (in[4:3] == 2'b00) ? out0 : 8'b00000000;
assign out[15:8]   = (in[4:3] == 2'b01) ? out1 : 8'b00000000;
assign out[23:16]  = (in[4:3] == 2'b10) ? out2 : 8'b00000000;
assign out[31:24]  = (in[4:3] == 2'b11) ? out3 : 8'b00000000;

endmodule

```