1.仔细阅读下面的 Verilog HDL 代码并回答后面的问题：

```
module ABC(A,B,Cin,S,Cout);
  input[3:0] A,B;
  input Cin;
  output[3:0] S;
  output Cout;
  assign {Cout,S}=A+B+Cin;
endmodule

module XYZ (A,B,Cin,S,Cout);
  input[7:0] A,B;
  input Cin;
  output[7:0] S;
  output Cout;
  wire CC;
  ABC ABC1(A[3:0],B[3:0],Cin,S[3:0],CC);
  ABC ABC2(A[7:4],B[7:4],CC,S[7:4],Cout);
endmodule
```

1）说明模块 ABC 描述的电路的逻辑功能并画出其逻辑符号；

参考解答：4 位全加器，其中：A,B 为数据输入；Cin 为进位输入；S 为数据输出；Cout 为进位输出。

逻辑符号略!

2）说明模块 XYZ 描述的电路的逻辑功能并画出其逻辑电路。

参考解答：8 位全加器，其中：A,B 为数据输入；Cin 为进位输入；S 为数据输出；Cout 为进位输出。

逻辑符号略!

2.说明模块 ABC 所描述的电路的逻辑功能并画出其逻辑符号：

```
module ABC(A,B,GT,ET,LT);
  input[3:0] A,B;
  output reg GT,ET,LT;
  always @(A,B) begin
    if (A>B) begin GT=1;ET=0;LT=0; end
    else if (A==B) begin GT=0;ET=1;LT=0; end
    else begin GT=0;ET=0;LT=1; end
  end
endmodule
```

参考解答：4 位数值比较器，其中：A,B 为数据输入；GT 为大于输出，ET 为等于输出，LT 为小于输出，均为高电平有效。

逻辑符号略!

3.说明模块 ABC 所描述的电路的逻辑功能并画出其逻辑符号：

```
module ABC(EN,A,Y);
  input EN;
```

```
  input[2:0] A;
  output reg[7:0] Y;
  always @(EN,A) begin
    if (!EN)
      case (A)
        0:Y=8’b11111110;
        1:Y=8’b11111101;
        2:Y=8’b11111011;
        3:Y=8’b11110111;
        4:Y=8’b11101111;
        5:Y=8’b11011111;
        6:Y=8’b10111111;
        7:Y=8’b01111111;
      endcase
    else Y=255;
  end
endmodule
```

参考解答：3 线-8 线译码器，其中：EN 为使能输入，低电平有效；A 为地址输入；Y 为译码输出，低电平有效。

逻辑符号略!

4. 说明模块 ABC 所描述的电路的逻辑功能并画出其逻辑符号：

```
module ABC(A,Y,V);
  input[3:0] A;
  output reg[1:0] Y;
  output reg V;
  always @(A) begin
    if (A[3]==0) begin Y=2’b00;V=0;end
    else if (A[2]==0) begin Y=2’b01;V=0;end
    else if (A[1]==0) begin Y=2’b10;V=0;end
    else if (A[0]==0) begin Y=2’b11;V=0;end
    else begin Y=2’b11;V=1;end
  end
endmodule
```

参考解答：4 线-2 线优先编码器，其中：A 为待编码的输入，低电平有效；Y 为编码输出，低电平有效；V 为 A 是否全部无效的标志输出，高电平有效。

逻辑符号略!

5.用 Verilog HDL 设计一个 3 线-8 线译码器，要求：先设计一个带使能输入（低电平有效）的 2 线-4 线译码器，然后利用此 2 线-4 线译码器设计一个 3 线-8 线译码器。

参考解答：

2 线-4 线译码器

```
module Decoder24(EN,A,Y);
  input EN;
  input[1:0] A;
  output reg[3:0] Y;
  always @(EN,A) begin
    if (!EN) begin
      case (A)
        2’b00:Y=1110;
        2’b01:Y=1101;
        2’b10:Y=1011;
        2’b11:Y=0111;
      endcase
    end
    else Y=4’b1111;
  end
endmodule
```

3 线-8 线译码器

```
module Decoder38(A,Y);
  input[2:0] A;
  output[7:0] Y;
  wire EN0,EN1;
```

```
  assign EN0=A[2];
  assign EN1=~A[2];
  Decoder24 myDecoder240(EN0,A[1:0],Y[3:0]);
  Decoder24 myDecoder241(EN1,A[1:0],Y[7:4]);
endmodule
```

6.用 Verilog HDL 设计一个 8 线-1 线选择器，要求：先设计一个带使能输入（低电平有效）的 4 线-1 线选择器，然后利用此 4 线-1 线选择器设计一个 8 线-1 线选择器。

参考解答：

4 线-1 线选择器

```
module Multiplexer41(EN,D,A,Y);
  input EN;
  input[3:0] D;
  input[1:0] A;
  output reg Y;
  always @(EN,D,A) begin
    if (!EN) begin
      case (A)
        2’b00:Y=D[0];
        2’b01:Y=D[1];
        2’b10:Y=D[2];
```

```
        2’b11:Y=D[3];
      endcase
    end
    else Y=0;
  end
endmodule
```

8线-1线选择器

```
module Multiplexer81(D,A,Y);
  input[7:0] D;
  input[2:0] A;
  output Y;
  wire EN0,EN1;
  assign EN0=A[2];
  assign EN1=~A[2];
  wire Y0,Y1;
  Multiplexer41 myMultiplexer410(EN0,D[3:0],A[1:0],Y0);
  Multiplexer41 myMultiplexer411(EN1,D[7:4],A[1:0],Y1);
  assign Y=Y0|Y1;
endmodule
```

或者：

```
module Multiplexer81(D,A,Y);
  input[7:0] D;
  input[2:0] A;
  output Y;
  wire EN0,EN1;
  assign EN0=0;
  assign EN1=0;
  wire Y0,Y1;
  Multiplexer41 myMultiplexer410(EN0,D[3:0],A[1:0],Y0);
  Multiplexer41 myMultiplexer411(EN1,D[7:4],A[1:0],Y1);
  assign Y=A[2]?Y1:Y0;
endmodule
```