

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；

2) 说明模块 XYZ 描述的电路的逻辑功能并画出其逻辑电路。

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

```

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;
            endcase
        else
            Y=8'b00000000;
        end
    end
endmodule

```

```

        5:Y=8' b11011111;
        6:Y=8' b10111111;
        7:Y=8' b01111111;
    endcase
    else Y=255;
end
endmodule

```

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

```

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。

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。