

1. 阅读下面的 Verilog HDL 代码，说明模块 ABC 描述的电路的逻辑功能并画出其逻辑符号：

```
module ABC (CP, S, SD, D, Q);  
    input CP, S, SD;  
    input[7:0] D;  
    output reg[7:0] Q;  
    always @(posedge CP, S)  
        if (S) Q<=D;  
        else begin  
            Q[0]<=Q[1];Q[1]<=Q[2];Q[2]<=Q[3];Q[3]<=Q[4];  
            Q[4]<=Q[5];Q[5]<=Q[6];Q[6]<=Q[7];Q[7]<=SD;  
        end  
endmodule
```

**参考解答：**具备异步置数功能的 8 位移位寄存器，其中：CP 为时钟输入，上升沿有效；S 为同步置数控制输入，高电平有效；SD 为移位数据输入；D 为同步置数数据输入；Q 为状态输出。

**逻辑符号略！**

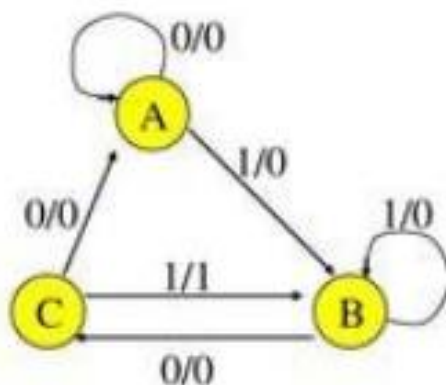
2. 阅读下面的 Verilog HDL 代码，说明模块 ABC 描述的电路的逻辑功能并画出其逻辑符号：

```
module ABC (CP, S, D, Q, B0);  
    input CP, S;  
    input [7:0] D;  
    output reg [7:0] Q;  
    output reg B0;  
    always @(posedge CP, S)  
        if (S) Q<=D;  
        else begin  
            if (Q==8' H01) B0<=1; else B0<=0;  
            Q<=Q-1;  
        end  
endmodule
```

**参考解答：**具备异步置数功能的 8 位 2 进制减 1 计数器，其中：CP 为时钟输入，上升沿有效；S 为同步置数控制输入，高电平有效；D 为同步置数数据输入；Q 为计数状态输出；Z 为借位输出，计数值为 0 时产生借位。

**逻辑符号略！**

3. 用 Verilog HDL 描述下图所示的同步时序逻辑电路：



**参考解答：**为避免挂起，无效状态的次态为 A；无效状态下的输出为 0。

```
module ABC (CP, X, Q, Z);
    input CP, X;
    output reg[1:0] Q;
    output reg Z;
    parameter A=2' b00, B=2' b01, C=2' b10, D=2' 11; //状态编码
    assign Z=((Q==C)&&X);
    always @(negedge CP) begin
        case (Q)
            A:if (X) Q<=B;
            B:if (!X) Q<=C;
            C:if (X) Q<=B;else Q<=A;
            D:Q<=A;
        endcase
    end
endmodule
```

4. 用 Verilog HDL 描述下表所示的同步时序逻辑电路：

| 现态<br>$Q_2^n Q_1^n Q_0^n$ | 次态<br>$Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$ | 输出<br>$CO$ | 现态<br>$Q_2^n Q_1^n Q_0^n$ | 次态<br>$Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$ | 输出<br>$CO$ |
|---------------------------|---------------------------------------|------------|---------------------------|---------------------------------------|------------|
| 000                       | 010                                   | 0          | 100                       | 110                                   | 0          |
| 001                       | ddd                                   | d          | 101                       | ddd                                   | d          |
| 010                       | 100                                   | 0          | 110                       | 111                                   | 0          |
| 011                       | ddd                                   | d          | 111                       | 000                                   | 1          |

**参考解答：**为避免挂起，无效状态的次态为 000；无效状态下的输出为 0。

```

module ABC (CP, Q, CO) ;
    input CP;
    output reg[2:0] Q;
    output CO;
    assign CO=Q[2]&Q[1]&Q[0];
    always @(negedge CP) begin
        case (Q)
            3' b000:Q<=3' b010;
            3' b010:Q<=3' b100;
            3' b100:Q<=3' b110;
            3' b110:Q<=3' b111;
            3' b111:Q<=3' b000;
            default:Q<=3' b000;
        endcase
    end
endmodule

```

```
        endcase
    end
endmodule
```

5. 设计一个 16 进制加 1 计数器，要求：先设计一个 4 进制加 1 同步计数器，然后利用它设计 16 进制加 1 计数器。

**参考解答：**

```
module Counter4(CP, Q, CO);
    input CP;
    output reg[1:0] Q;
    output CO;
    assign CO=Q[1]&Q[0];
    always @(negedge CP) begin Q<=Q+1; end
endmodule
```

```
module Counter16(CP, Q, CO);
    input CP;
    output[3:0] Q;
    output CO;
    wire CC
    Counter4 myCounter40(CP, Q[1:0], CC); //低位片
    Counter4 myCounter41(~CC, Q[3:2], CO); //高位片
```

endmodule