

1. 由 4 个 JK 触发器构成的二进制减法计数器初值为  $Q_3^n Q_2^n Q_1^n Q_0^n = 1110$ ，经过 2017 个脉冲计数后，计数器状态为  $Q_3^{n+1} Q_2^{n+1} Q_1^{n+1} Q_0^{n+1} = (\underline{0011})_2$
  2. 用 Verilog 语法表达的 8'h34 数对应的 8421BCD 码为(1001 0010)<sub>8421BCD</sub>
  3. Verilog 代码中 assign Y = (SEL == 1) ? D1 : D0 表述的逻辑功能为 2 选 1 的多路选择器。
  4. 将周期性正弦波信号转换为同频率的数字脉冲波形可用 施密特触发器。
  5. 谐振频率为 11.0592MHz 的石英晶体振荡器，当用作 38.4kbps 的二进制串行通信输出时钟时，至少需用 9 个触发器来构造分频器。
  6. 状态机实现中，若状态表达采用 8 位二进制数构成的独热码(one-hot code)，最多可提供 8 个状态。
  7. 逻辑函数  $F = \overline{AC(B+D)} + \overline{BC}$ ，则  $F \oplus 1 = \underline{\overline{AC(B+D)} + \overline{BC}}$ 。
  8. Verilog always 语句中  $Q[3:0] <= \{Q[2:0], Q[3]\}$  实现的操作功能是 4 位循环移位。
  9. 同一逻辑函数中同下标的最小项  $m_i$  和最大项  $M_i$  的关系为  $m_i = \underline{m_i} = \underline{M_i}$ 。
  10. 以 4 倍采样频率采集的一个完整波形周期序列由 200 个 16bit 采样数字样点值构成。当以每个样点数据 2us 更新速率循环写入由 +5V 供电的 1MSPS D/A 转换器时，D/A 输出端得到的无失真模拟信号重复频率为 2.5KHz。
- 2、用代数法将  $1 - \overline{(\overline{A+B}) + (\overline{A+B})} + \overline{(\overline{A \cdot B})(\overline{A \cdot B})}$  化简为最简与或式。(注：无推导步骤，不扣分)

## 二、化简题（共 10 分）

得分

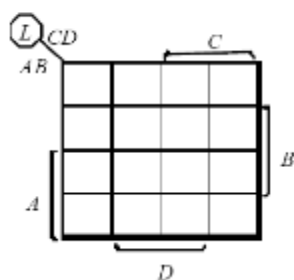
1、代数化简  $F = \overline{A+B+C} + (A \oplus B \oplus C) + ABC$

注：无步骤，不给分。

2、使用卡诺图化简  $F$  为最简与或式。  $F(A,B,C,D) = \sum m(1,2,4,5,6,9,11,13,15)$

注：不画圈，不给分。

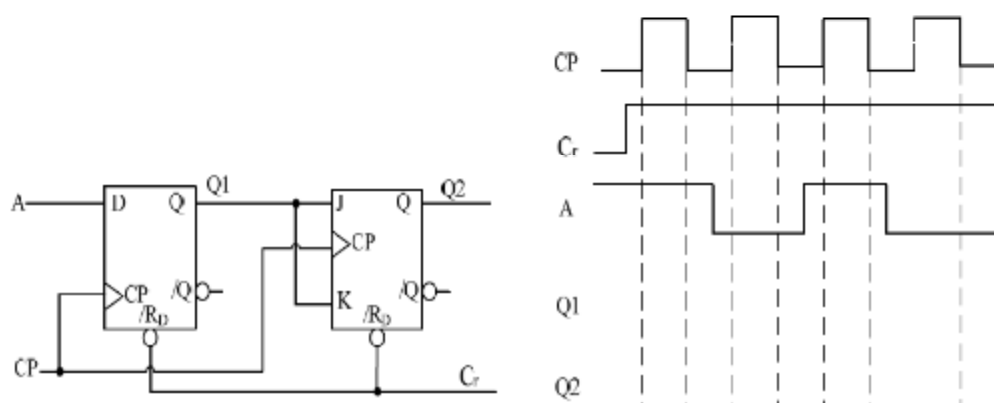
2、已知函数  $L$  的逻辑功能为：AB-00 时， $L = C + D$ ；AB-01 时， $L = \overline{CD}$ ；AB-10 时， $L = C \oplus D$ ；AB-11 时， $L$  为任意项。使用卡诺图法求逻辑函数  $L$  的“最简与或式”（备注：不画圈，不给分）。



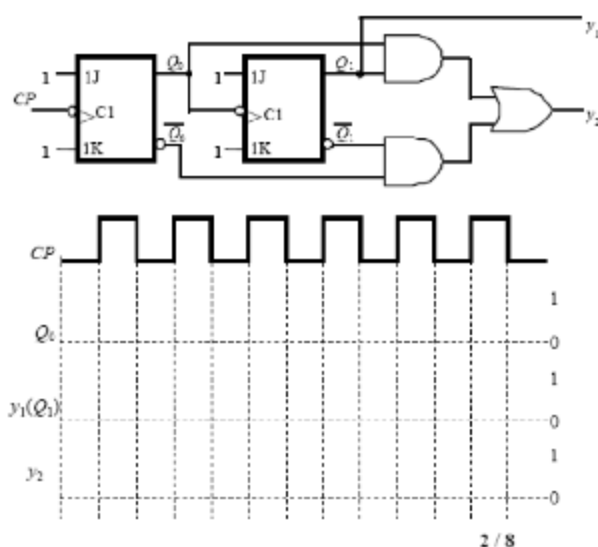
## 三、请画出 Q1、Q2 的波形（共 8 分）

得分

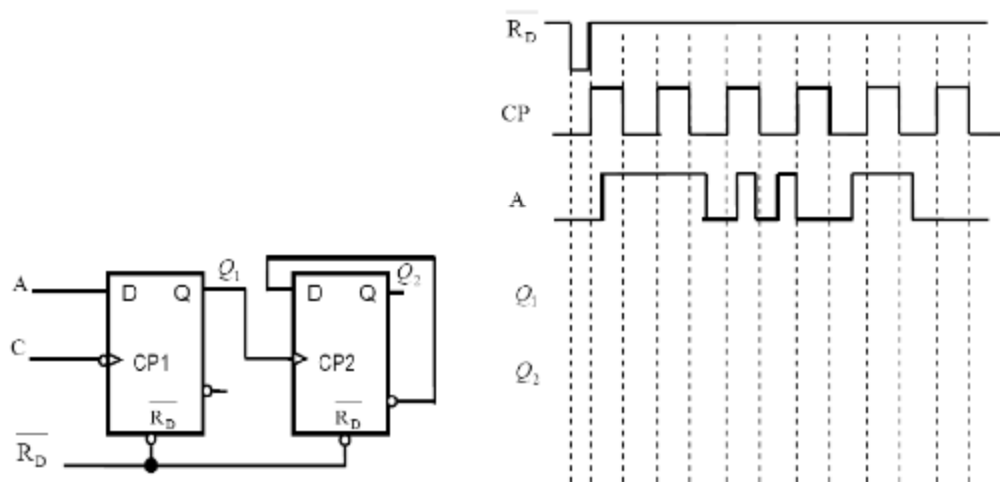
注： $\overline{R_D}$  均为异步清零信号。



- 三、 由 2 个边沿 JK 触发器（备注：初态都为 0）、2 个与门和 1 个或门组成的两相脉冲源电路如图所示，试对应时钟脉冲 CP 画出图中 Q0、Q1 和 y1、y2 的波形。（共 8 分）

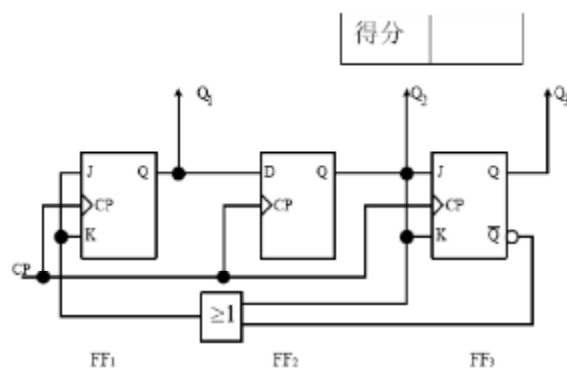


- 三、 请根据图 3 电路和输入波形，画出对应的 Q1、Q2 波形（ $\overline{R_d}$  是异步清零输入端）（10 分）



## 五、分析时序电路（共 14 分）

(1) 写出驱动方程、状态方程；



(2) 完成状态转换表；

$Q_3^n Q_2^n Q_1^n$	$Q_3^{n+1} Q_2^{n+1} Q_1^{n+1}$
0 0 0	
0 0 1	
0 1 0	
0 1 1	
1 0 0	
1 0 1	
1 1 0	
1 1 1	

(3) 画出状态转换图，检查能否自启动；

## 五、 Verilog HDL 代码分析（共 15 分）

得分

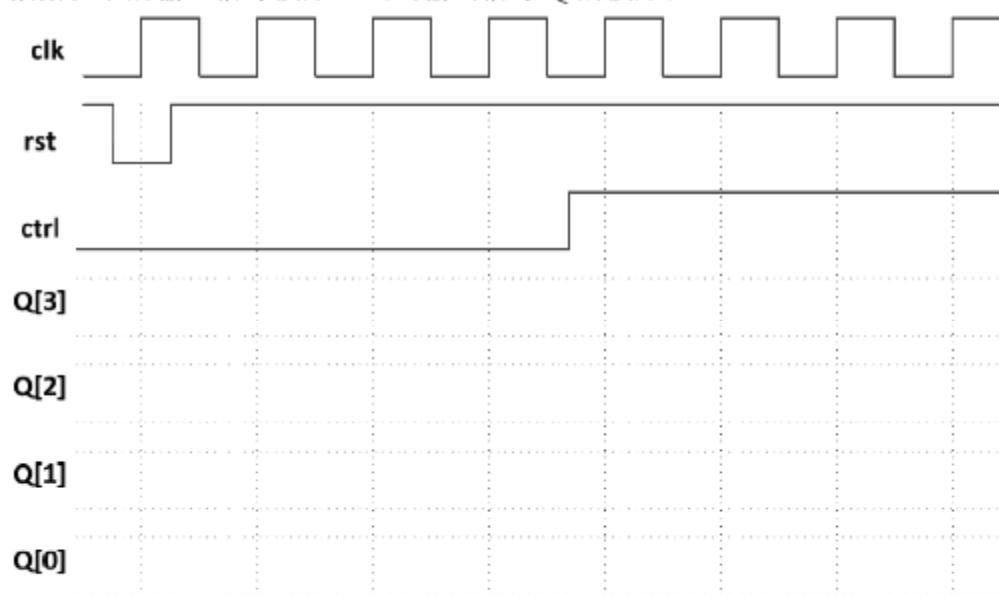
```
module test( clk , rst , ctrl , Q);
input clk , rst , ctrl;
output[3:0] Q;
reg[1:0] iCnt;
```

```
always@(posedge clk)
if(!rst)    iCnt <= 2'b00;
else if(!ctrl)    iCnt <= iCnt + 1'b1;
else    iCnt <= iCnt - 1'b1;
```

```
always@(iCnt)
case (iCnt)
2'b00 :    Q = 4'b1110;
2'b01 :    Q = 4'b1101;
2'b10 :    Q = 4'b1011;
2'b11 :    Q = 4'b0111;
default :    Q = 4'b1111;
```

```
endmodule
```

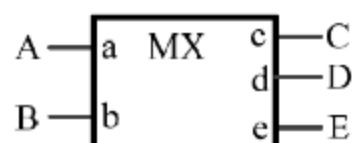
2、根据以下的输入信号波形，画出输出信号 Q 的波形；



3、简要说明程序中两个 always 语句的功能（注：分别写在二个虚线圆角矩形框里）

三、芯片 MX 的功能由如下 Verilog HDL 程序描述。若在其输入管脚上加入的 A,B 信号的波形如图所示，设电路的初始状态为 C=0, D=1, 请在图三中画出对应的输出信号 C,D,E 的波形。（共 8 分）

得分



```
module MX(a, b, c, d, e);
input a, b;
output c, d, e;
reg [1:0] state;
parameter S0=2' b00, S1=2' b01,
          S2=2' b10, S3=2' b11;
assign {c,d} = state;
assign e = c^d;
```

```
always @(posedge a or negedge b)
if ( ~ b) state <= S0;
else
case (state)
S0: state <= S2;
S1: state <= S3;
S2: state <= S3;
S3: state <= S0;
endcase
endmodule
```

## 六、 时序电路设计题（共 15 分）

得分	
----	--

用 4 位二进制同步计数器 74HC161，完成下列电路。（74HC161 功能表见附录）。

- (1) 不增加其他电路元件，直接在图 6-1 电路上完成电路连线，并在括号中标注高、低电平，完成 0,1,……10,0,1……的循环计数。（可用 0, 1 分别标注低电平和高电平）

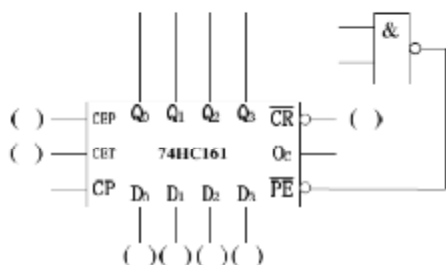


图 6-1

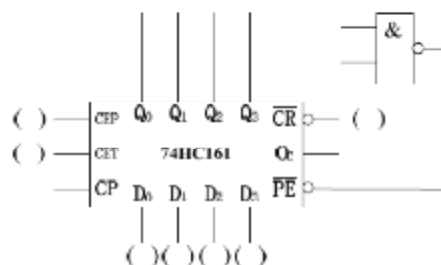


图 6-2

- (2) 不增加其他电路元件，直接在图 6-2 电路上完成电路连线，并在括号中标注高、低电平，完成 0,1,2,3,0,1,2,3……的循环计数。（可用 0, 1 分别标注低电平和高电平）
- (3) 若需要实现 0,1,2,3,0,1,……10,0,1,2,3,0,1,……10……的循环计数，应如何设计电路？请简要阐述设计思路。（如果必要,可用电路框图辅助说明设计原理，不需要画出详细电路实现）

## 六、综合题（10 分）

双向移位寄存器 74HCT194 和 16X4 PROM 组成如图 7 电路：

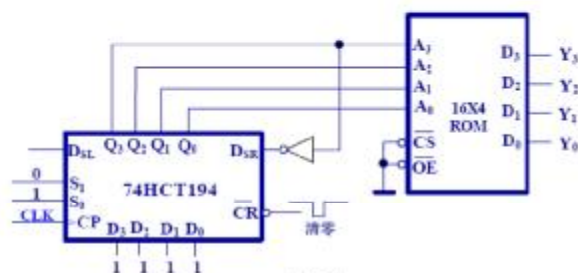


图 7

- (1) 其中移位寄存器构成什么电路？根据图 7 中所示条件画出其工作状态图：

$Q_3Q_2Q_1Q_0 \rightarrow$

1. 逻辑电路 MX 的功能由如下 Verilog HDL 代码给出，请完成以下步骤：

- 1) 写出 Y 的最小项表达式（包含无关项）；
- 2) 利用卡诺图将 Y 化简为最简与或式（不画圈不给分）。

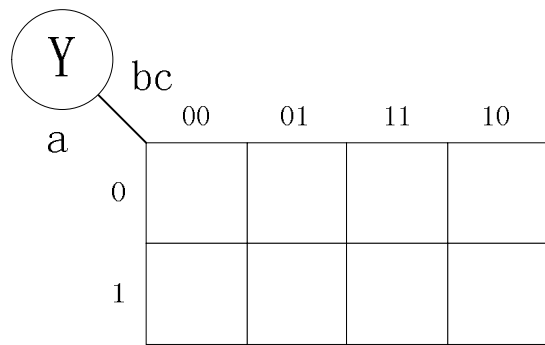
1)  $Y =$

```

module MX(a, b, c, Y);
input a, b, c;
output Y;
reg Y;
always @(a or b or c)
  case ({a,b,c})
    3'b000: Y = 1'b1;
    3'b001: Y = 1'b1;
    3'b010: Y = 1'b0;
    3'b011: Y = 1'b0;
    3'b100: Y = 1'b1;
    3'b101: Y = 1'b1;
    default: Y = 1'bx;
  endcase
endmodule

```

2)



$$Y = m(0, 1, 4, 5) + d(6, 7)$$

## 五、时序电路分析（共 14 分）

已知如下 Verilog HDL 描述的电路：

```

module logic_c(output Y,
  output [1:0] Q, input X, CLK, CLR);
  wire [1:0] A;
  flip_flop ff0(Q[0], A[0], CLK, CLR);
  flip_flop ff1(Q[1], A[1], CLK, CLR);
  assign A[1] = ~(Q[0] | X) ^ Q[1];
  assign A[0] = X ^ Q[0];
  assign Y = Q[1] & Q[0];
endmodule

```

```

module flip_flop(output Q,
  input A, CP, CR);
  reg Q;
  always @(negedge CP, negedge CR)
    if(~CR) Q <= 0;
    else Q <= A;
endmodule

```

1. 简述 flip\_flop 模块功能（2 分）

2. 写出 logic\_c 电路的 Y 端的输出方程（2 分）

3. 填写以下 logic\_c 电路状态方程  $Q_i^{n+1}(X, Q_1, Q_0)$  的最小项表达式（4 分）

$$Q_1^{n+1} = \sum m(\quad), \quad Q_0^{n+1} = \sum m(\quad)$$

4. 完成以下 logic\_c 电路的状态图（4 分）

