

第5章

时序电路的Verilog设计

5.1 基本时序元件的Verilog表述

5.1.1 基本D触发器及其Verilog表述

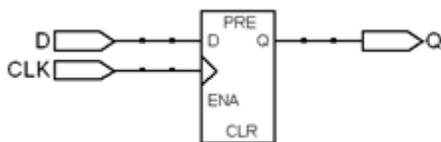


图 5-1 边沿触发型 D 触发器

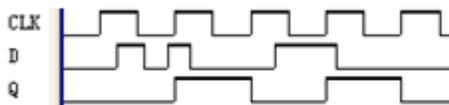


图 5-2 D 触发器时序波形

【例 5-1】

```
module DFF1(CLK,D,Q);  
    output Q ;  
    input  CLK, D ;  
    reg Q;  
    always @(posedge CLK )  
        Q <= D;  
Endmodule
```

5.1 基本时序元件的Verilog表述

5.1.2 用UDP表述D触发器

【例 5-2】

```
primitive EDGE_UDP(Q,D,CLK,RST);
input D,CLK,RST; output Q; reg Q;
table // D CLK RST : Q : Q+
    0 (01) 0 : ? : 0;
    1 (01) 0 : ? : 1;
    ? (1?) 0 : ? : -;
    ? (?0) 0 : ? : -;
    1 0 1 : ? : 0;
    1 1 1 : ? : 0;
    0 0 1 : ? : 0;
    0 1 1 : ? : 0;
endtable
endprimitive
```

【例 5-3】

```
module DFF_UDP (Q,D,CLK,RST) ;
input D,CLK,RST;
output Q;
EDGE_UDP U1(Q,D,CLK,RST);
Endmodule
```

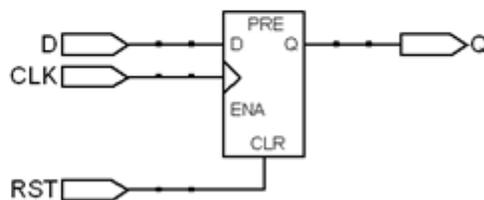


图 5-3 含异步复位的边沿触发型 D 触发器

5.1 基本时序元件的Verilog表述

5.1.3 含异步复位和时钟使能的D触发器及其Verilog表述

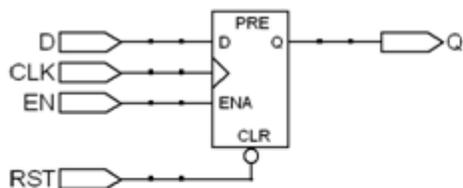


图 5-4 含使能和复位控制的 D 触发器

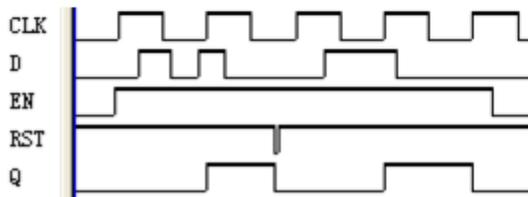


图 5-5 图 5-4 的 D 触发器的时序图

【例 5-4】

```
module DFF2 (CLK, D, Q, RST, EN) ;
    output Q;
    input CLK, D, RST, EN;
    reg Q;
    always @(posedge CLK or negedge RST)
        begin
            if (!RST) Q <= 0;
            else if (EN) Q <= D;
        end
endmodule
```

5.1 基本时序元件的Verilog表述

5.1.4 含同步复位控制的D触发器及其Verilog表述

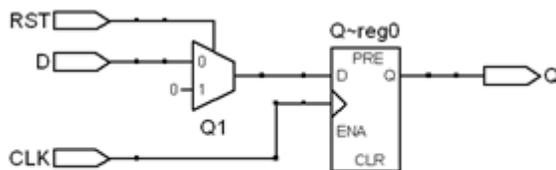


图 5-6 含同步清 0 控制的 D 触发器

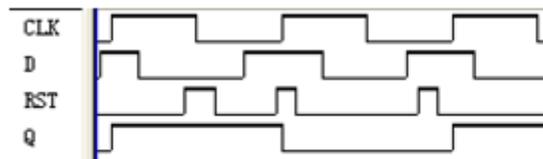


图 5-7 含同步清 0 控制 D 触发器的时序图

5.1 基本时序元件的Verilog表述

5.1.4 含同步复位控制的D触发器及其Verilog表述

【例 5-5】

```
module DFF3 (CLK, D, Q, RST);
    output Q;
    input CLK, D, RST;
    reg Q;
    always @(posedge CLK)
        if (RST==1) Q = 0;
        else if (RST==0) Q = D;
endmodule
```

【例 5-6】

```
module DFF1 (CLK, D, Q, RST);
    output Q; input CLK, D, RST;
    reg Q, Q1; //注意定义了Q1信号
    always @(RST) //纯组合过程
        if (RST==1) Q1=0; else Q1=D;
    always @(posedge CLK)
        Q <= Q1;
endmodule
```

【例 5-7】

```
module DFF2 (input CLK, input D, input RST, output reg Q);
    always @(posedge CLK) Q <= RST ? 1'b0 : D;
endmodule
```

5.1 基本时序元件的Verilog表述

5.1.5 基本锁存器及其Verilog表述

【例 5-8】

```
module LATCH1 (CLK, D, Q);  
    output Q ; input CLK, D;  
    reg Q;  
    always @(D or CLK)  
        if (CLK) Q <= D;  
endmodule
```

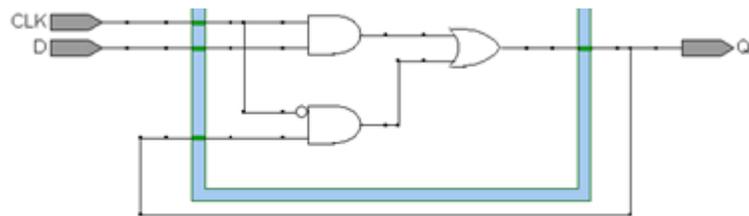


图 5-9 锁存器模块内部逻辑电路

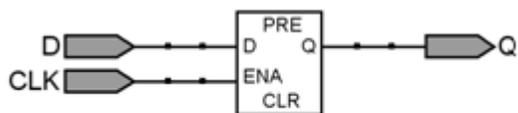


图 5-8 锁存器模块

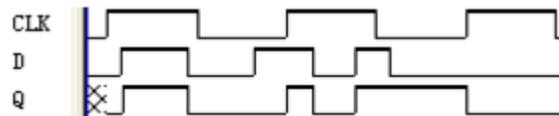


图 5-10 例 5-8 锁存器的时序波形

5.1 基本时序元件的Verilog表述

5.1.6 含清0控制的锁存器及其Verilog表述

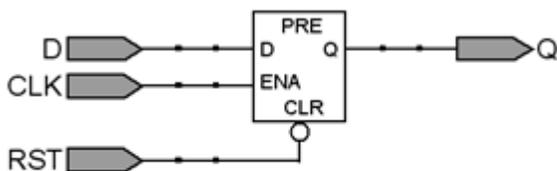


图 5-11 含异步清 0 的锁存器

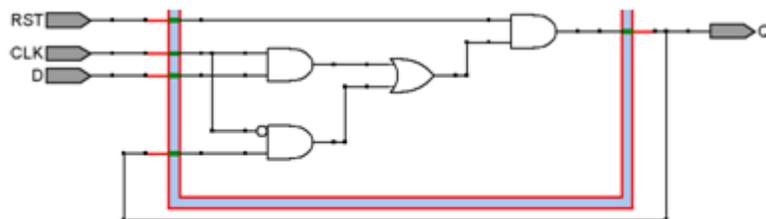


图 5-12 含异步清 0 锁存器的逻辑电路图

5.1 基本时序元件的Verilog表述

5.1.6 含清0控制的锁存器及其Verilog表述

【例 5-9】

```
module LATCH2 (CLK,D,Q,RST);  
    output Q ;    input CLK,D,RST;  
    assign Q = (!RST)? 0:(CLK ? D:Q);  
endmodule
```

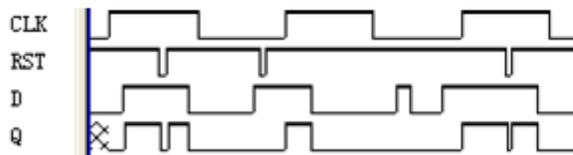


图 5-13 含异步清 0 的锁存器的仿真波形

【例 5-10】

```
module LATCH3 (CLK,D,Q,RST);  
    output Q ;  
    input CLK,D,RST;  
    reg Q;  
    always @(D or CLK or RST)  
        if(!RST) Q<=0;  
        else if(CLK) Q<=D;  
endmodule
```

5.1 基本时序元件的Verilog表述

5.1.7 异步时序电路的Verilog表述特点

【例 5-11】

```
module AMOD(D,A,CLK,Q);  
    output Q; input A,D,CLK;  
    reg Q,Q1;  
    always @(posedge CLK) Q1 = ~(A|Q);  
    always @(posedge Q1) Q = D;  
endmodule
```

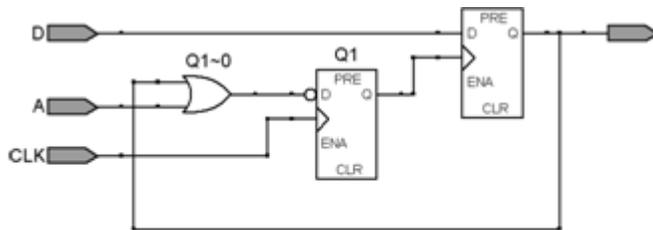


图 5-14 例 5-11 的时序电路图

● ● ● | 5.1 基本时序元件的Verilog表述

5.1.8 时钟过程表述的特点和规律

`always @(posedge CLK or RST), 或 always @(posedge CLK or negedge RST or A)`

```
always @ (posedge CLK or negedge RST) begin if (! RST) ...
```

```
always @ (posedge CLK or negedge RST) begin if (RST==0) ...
```

```
always @ (posedge CLK or negedge RST) begin if (! RST==1) ...
```

```
always @(posedge CLK or DIN)
```

5.1 基本时序元件的Verilog表述

5.1.8 时钟过程表述的特点和规律

【例 5-12】

```
module DFF5 (CLK, D, Q, RST, DIN, OUT);  
output Q, OUT; input CLK, D, RST, DIN;  
    reg Q, OUT;  
    always @(posedge CLK ) begin  
        OUT = !DIN ;  
        if (RST==1) Q=0;  
        else if(RST==0) Q=D; end  
endmodule
```

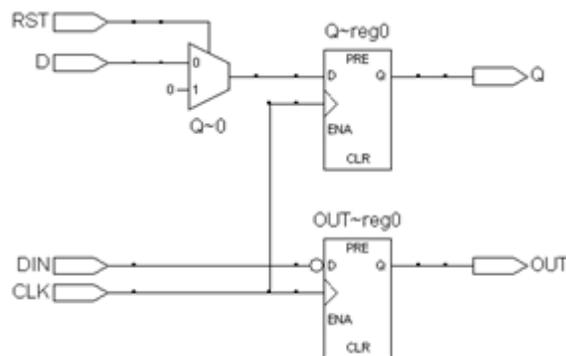


图 5-15 例 5-12 的 RTL 图

5.2 二进制计数器及其Verilog表述

5.2.1 简单加法计数器及其Verilog表述

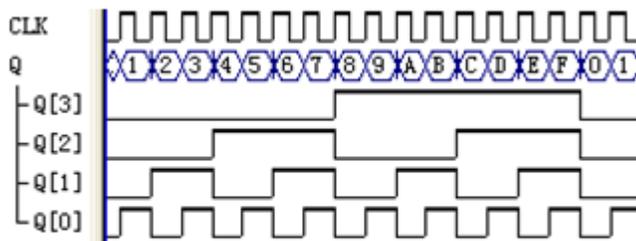


图 5-16 4 位加法计数器工作时序

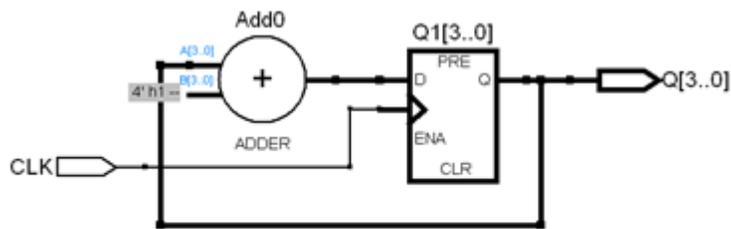


图 5-17 4 位加法计数器 RTL 电路图

5.2 二进制计数器及其Verilog表述

5.2.1 简单加法计数器及其Verilog表述

【例 5-13】

```
module CNT4(CLK,Q);  
    output [3:0] Q;    input  CLK;  
    reg [3:0] Q1 ;  
    always @(posedge CLK)  
        Q1 <= Q1+1 ;  
    assign Q=Q1;  
endmodule
```

【例 5-14】

```
module CNT4 (CLK,Q);  
    output [3:0] Q ;  
    input  CLK;  
    reg [3:0] Q ;  
    always @(posedge CLK)  
        Q <= Q+1 ;  
endmodule
```

5.2 二进制计数器及其Verilog表述

5.2.2 实用加法计数器设计

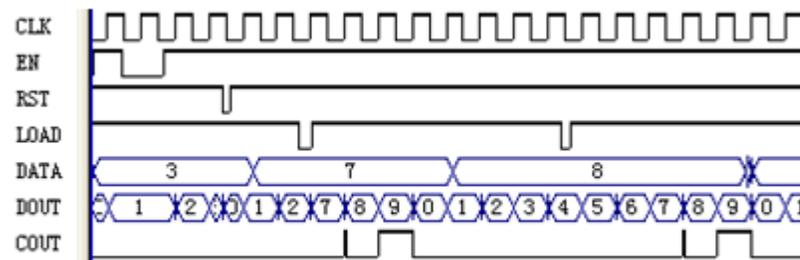


图 5-18 例 5-15 的仿真波形图

5.2 二进制计数器及其Verilog表述

5.2.2 实用加法计数器设计

【例 5-15】

```
module CNT10 (CLK,RST,EN,LOAD,COUT,DOUT,DATA);
    input CLK,EN,RST,LOAD ;      // 时钟, 时钟使能, 复位, 数据加载控制信号;
    input [3:0] DATA ;          // 4 位并行加载数据
    output [3:0] DOUT ;          // 4 位计数输出
    output COUT ;                // 计数进位输出
    reg [3:0] Q1 ;               reg COUT ;
    assign DOUT = Q1;            // 将内部寄存器的计数结果输出至 DOUT
    always @(posedge CLK or negedge RST) //时序过程
        begin
            if (!RST) Q1 <= 0;    //RST=0 时, 对内部寄存器单元异步清 0
            else if (EN) begin    //同步使能 EN=1, 则允许加载或计数
                if (!LOAD) Q1<=DATA; //当 LOAD=0, 向内部寄存器加载数据
                else if (Q1<9) Q1 <= Q1+1; //当 Q1 小于 9 时, 允许累加
                else Q1 <= 4'b0000; end //否则一个时钟后清 0 返回初值
            end
        end
    always @(Q1) //组合过程
        if (Q1==4'h9) COUT = 1'b1; else COUT = 1'b0;
endmodule
```

5.2 二进制计数器及其Verilog表述

5.2.2 实用加法计数器设计

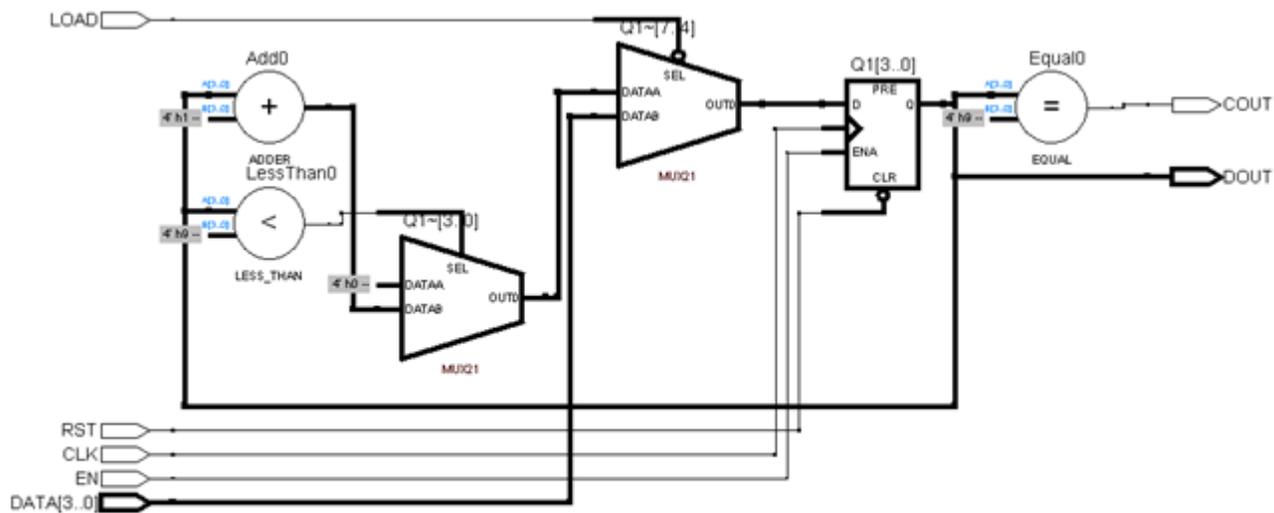


图 5-19 QuartusII 对例 5-15 综合和后得到的 RTL 电路图

5.3 移位寄存器的Verilog表述与设计

5.3.1 含同步预置功能的移位寄存器设计

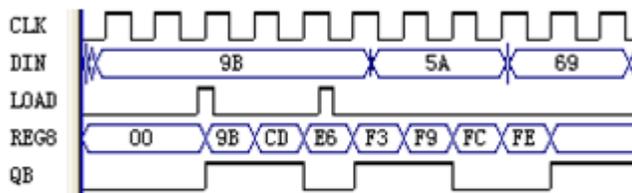


图 5-20 例 5-16 的工作时序图

【例 5-16】

```
module SHFT1(CLK,LOAD,DIN,QB);  
    output QB; input CLK,LOAD; input[7:0] DIN; reg[7:0] REG8;  
    always @(posedge CLK )  
        if (LOAD)    REG8<=DIN ; else REG8[6:0]<=REG8[7:1];  
    assign QB = REG8[0] ;  
endmodule
```

● ● ● | 5.3 移位寄存器的Verilog表述与设计

5.3.2 使用移位操作符设计移位寄存器

【例 5-17】

```
module SHIF4 (DIN,CLK,RST,DOUT);  
  input CLK,DIN,RST;  output DOUT;  reg [3:0] SHFT;  
  always@(posedge CLK or posedge RST)  
    if(RST) SHFT<=4'B0;  
    else begin SHFT <=(SHFT >> 1); SHFT[3] <= DIN; end  
  assign DOUT = SHFT[0];  
endmodule
```

5.4 可预置型计数器设计

5.4.1 同步加载计数器

【例 5-18】

```
module FDIV0 (input CLK,RST, input[3:0]D, output PM, output[3:0]DOUT);
  reg [3:0] Q1;    reg FULL;
  (* synthesis,keep *) wire LD ; //设定 LD 为仿真可测试属性
  always @(posedge CLK or negedge RST )
    if (!RST) begin Q1<=0 ; FULL<=0; end
    else if (LD) begin Q1<=D ; FULL<=1; end
    else begin Q1 <= Q1+1; FULL<=0 ; end
  assign LD=(Q1==4'B1111); assign PM=FULL; assign DOUT=Q1 ;
endmodule
```

5.4 可预置型计数器设计

5.4.1 同步加载计数器

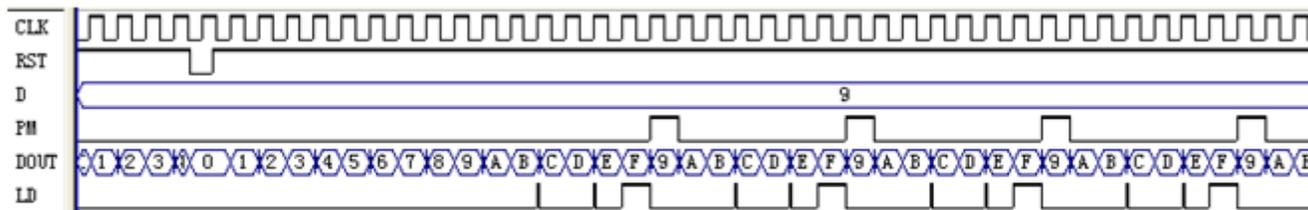


图 5-21 22 例 5-18 的仿真波形，同步加载模式（目标器件 EP1C3）

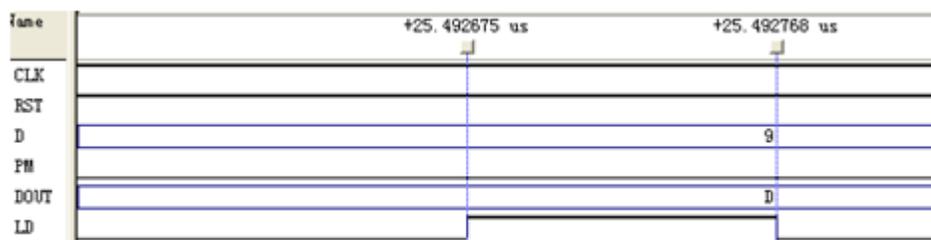


图 5-22 图 5-21 中毛刺脉冲展开后的时序，毛刺脉宽约 0.1ns

5.4 可预置型计数器设计

5.4.2 异步加载计数器

【例 5-19】

```
module fdiv1 (CLK, PM, D, DOUT, RST );
    input CLK; input RST; input[3:0] D; output PM; output[3:0] DOUT;
    reg [3:0] Q1; reg FULL;
    (* synthesis,probe_port,keep *) wire LD ;
    always @(posedge CLK or posedge LD or negedge RST )
        if (!RST) begin Q1<=0 ; FULL<=0; end
    else if (LD) begin Q1<=D ; FULL<=1; end
        else begin Q1<=Q1+1; FULL<=0 ; end
    assign LD=(Q1==4'B0000);
    assign PM=FULL;
    assign DOUT=Q1 ;
endmodule
```

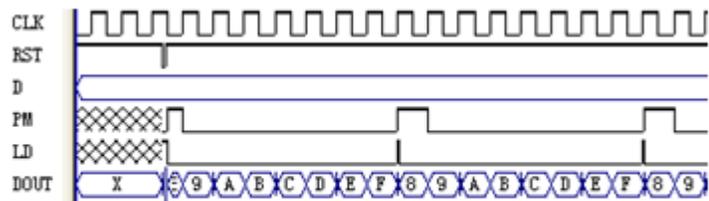


图 5-23 例 5-19 的仿真波形(目标器件 EP3C55)

5.4 可预置型计数器设计

5.4.3 异步清0加载计数器

【例 5-20】

```
module fdiv1 (CLK,PM,D );
input CLK; input[3:0] D;    output PM ; reg FULL;
(* synthesis,probe_port,keep *) reg [3:0] Q1;
(* synthesis,probe_port,keep *) wire RST ;
always @(posedge CLK or posedge RST)
    if (RST) begin Q1<=0;
                FULL<=1; end
    else begin Q1<=Q1+1;
              FULL<=0 ; end
assign RST = (Q1==D);
assign PM = FULL ;
endmodule
```

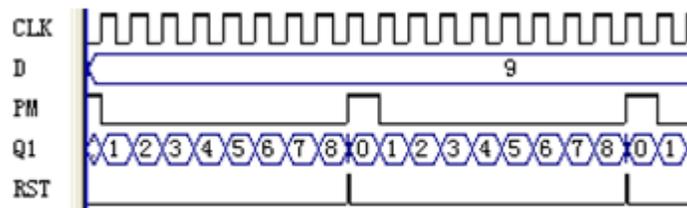


图 5-24 例 5-20 的仿真波形(EP3C55)

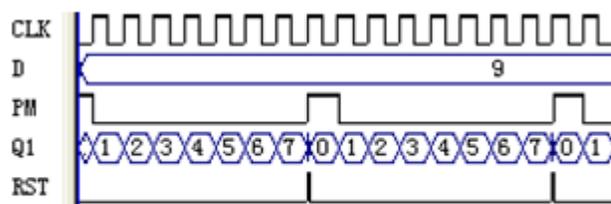


图 5-25 例 5-20 的仿真波形(EP1C3)

5.4 可预置型计数器设计

5.4.3 异步清0加载计数器

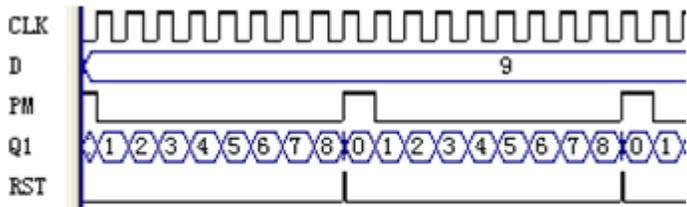


图 5-24 例 5-20 的仿真波形(EP3C55)

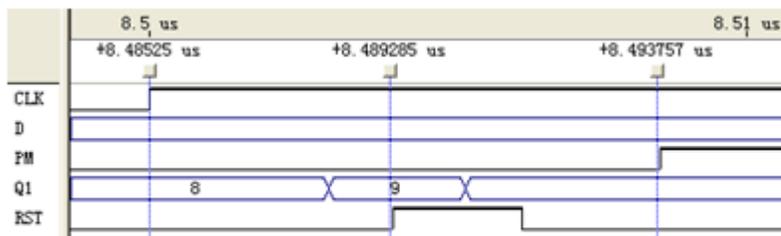


图 5-26 图 5-24 中的 RST 信号展开后的时序

5.4 可预置型计数器设计

5.4.4 同步清0加载计数器

```
always @(posedge CLK)
```

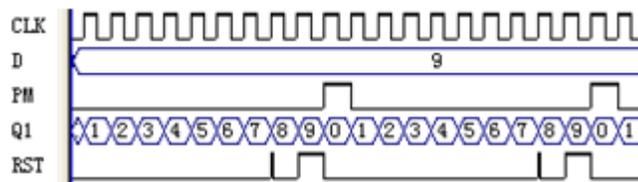


图 5-27 同步清 0 加载计数器时序波形

5.5 时序电路硬件设计与仿真示例

5.5.1 编辑电路、创建工程和仿真测试

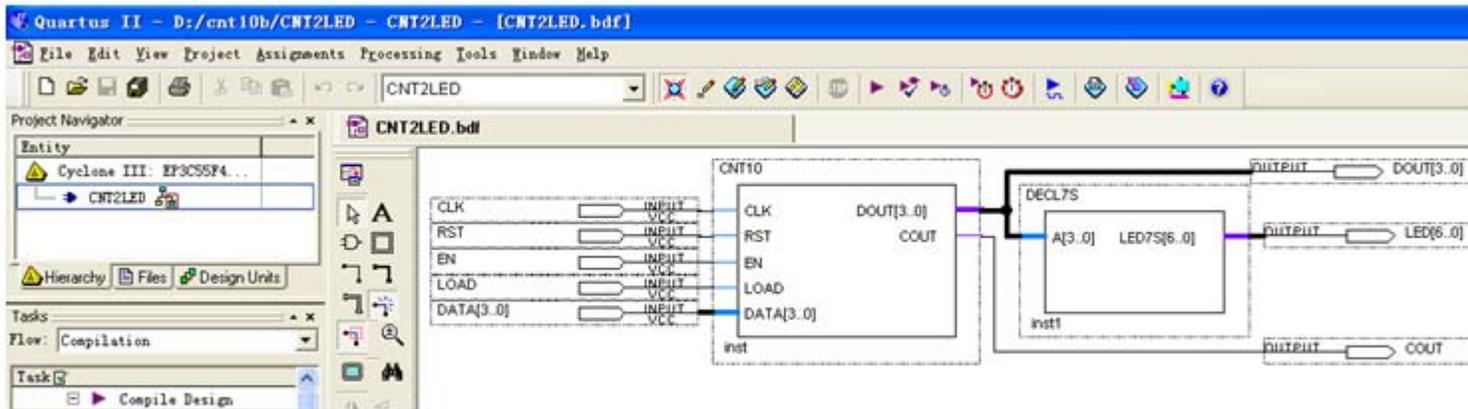


图 5-28 十进制计数器设计示例电路 CNT2LED

5.5 时序电路硬件设计与仿真示例

5.5.1 编辑电路、创建工程和仿真测试

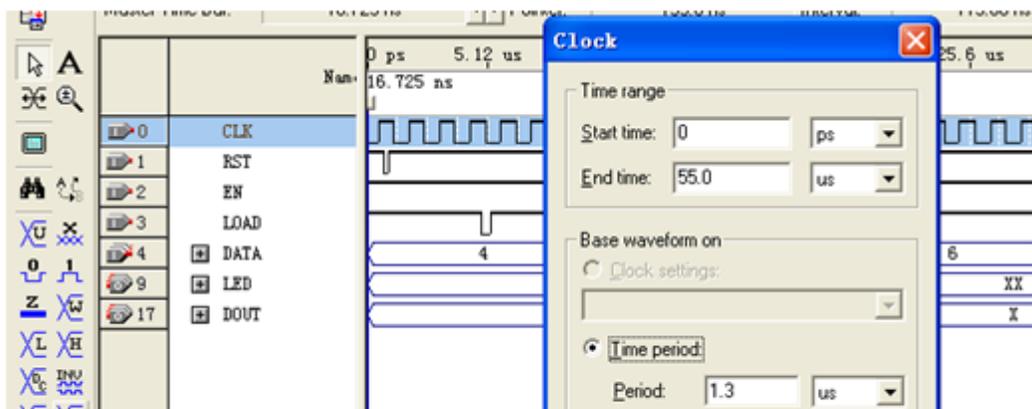


图 5-29 设置时钟 CLK 的周期

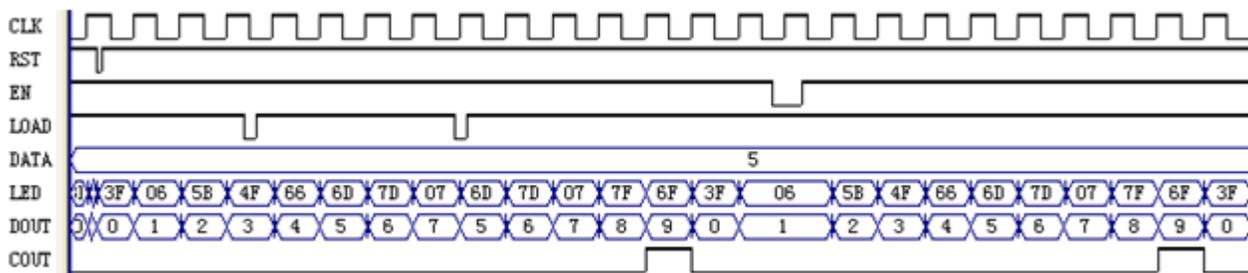


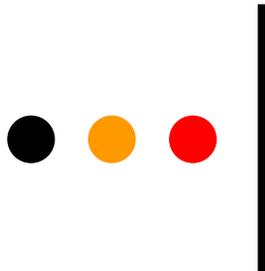
图 5-30 仿真波形输出报告 Simulation Report

5.5 时序电路硬件设计与仿真示例

5.5.2 FPGA硬件测试

表 5-1 基于 EP3C55F484 的 55F+ 开发板的引脚锁定表

CLK	RST	EN	LOAD	COUT	DATA(3)	DATA(2)	DATA(1)	DATA(0)
键 1,K1	键 2,K2	键 3,K3	键 4,K4	LED1	拨码 4	拨码 4	拨码 4	拨码 4
Pin:AA3	Pin:AB3	Pin:V3	Pin:Y7	Pin:B22	Pin:N22	Pin:M22	Pin:L21	Pin:L22
LED[6..0]	LED(6)	LED(5)	LED(4)	LED(3)	LED(2)	LED(1)	LED(0)	
数码管	LED: g	LED: f	LED: e	LED: d	LED: c	LED: b	LED: a	
引脚	Pin:G3	Pin:B2	Pin:E3	Pin:D8	Pin:A3	Pin:C6	Pin:H21	



习 题

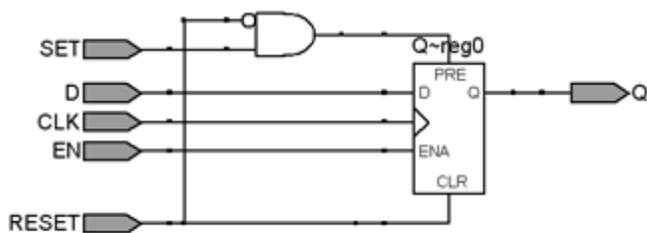


图 5-31 RTL 图 1

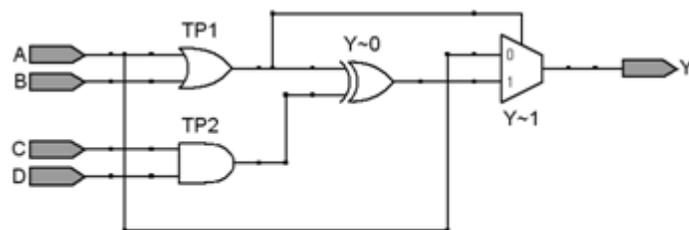


图 5-32 RTL 图 2

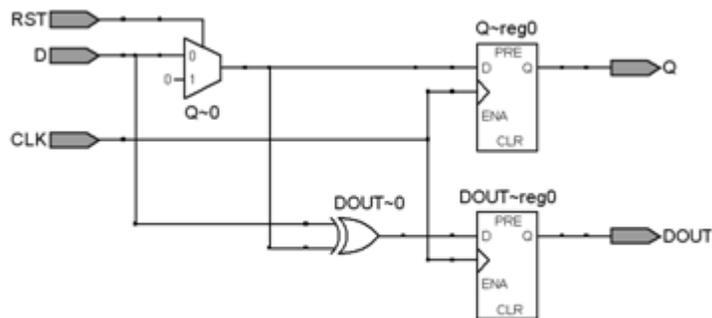


图 5-33 RTL 图 3

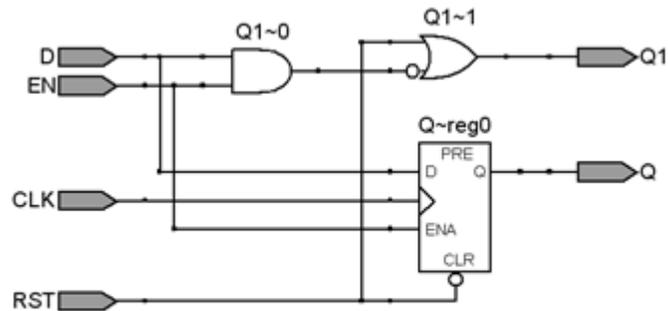


图 5-34 RTL 图 4

实验与设计

5-1. 应用宏模块设计数字频率计

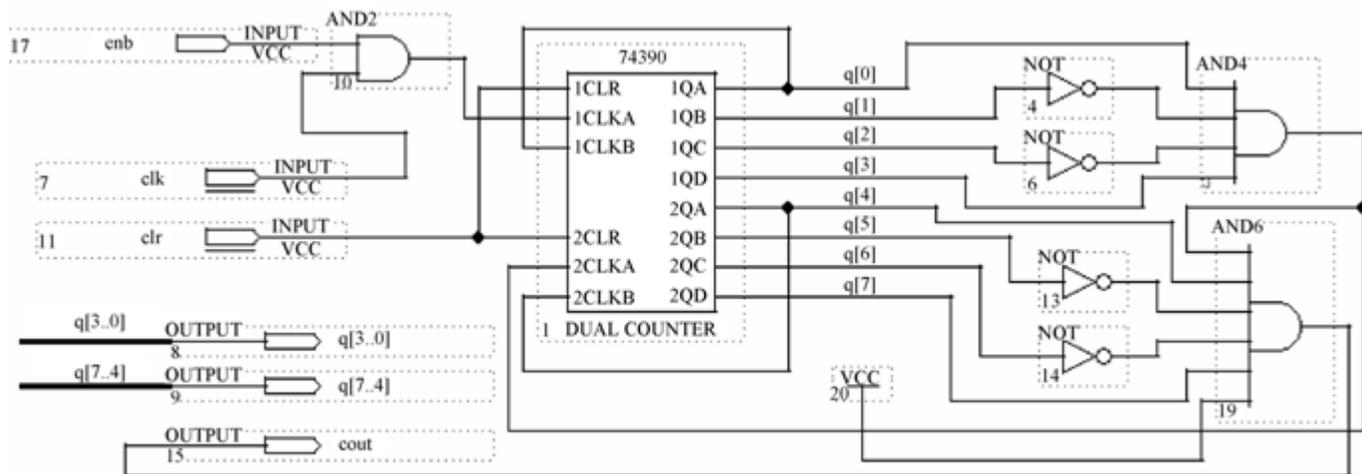


图 5-35 含有时钟使能的 2 位十进制计数器

实验与设计

5-1. 应用宏模块设计数字频率计

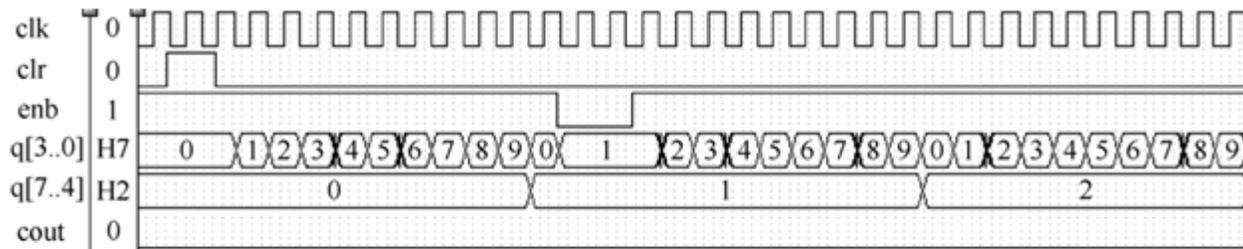


图 5-36 两位十进制计数器工作波形

实验与设计

5-1. 应用宏模块设计数字频率计

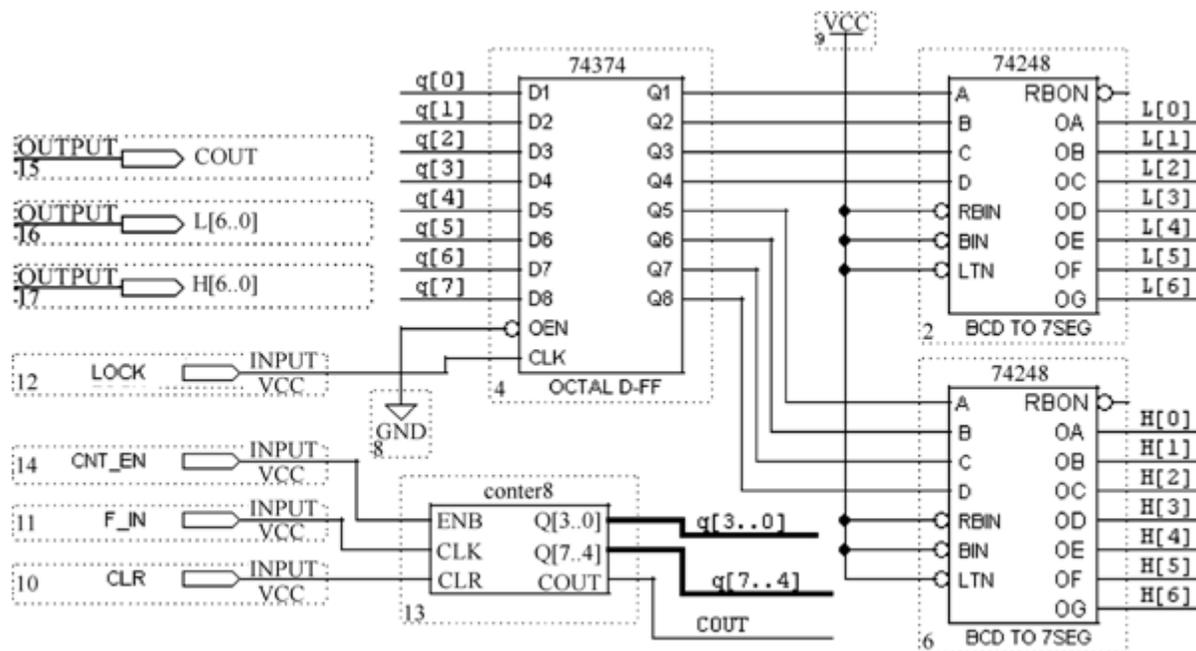


图 5-37 2 位十进制频率计顶层设计原理图文件

实验与设计

5-1. 应用宏模块设计数字频率计

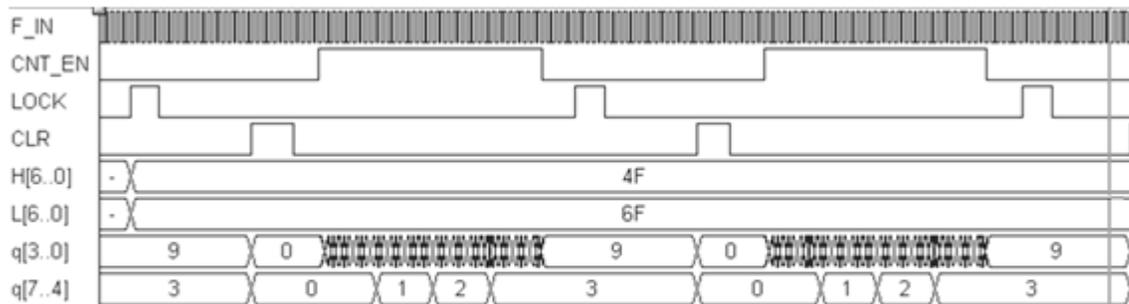


图 5-38 两位十进制频率计测频仿真波形

实验与设计

5-1. 应用宏模块设计数字频率计

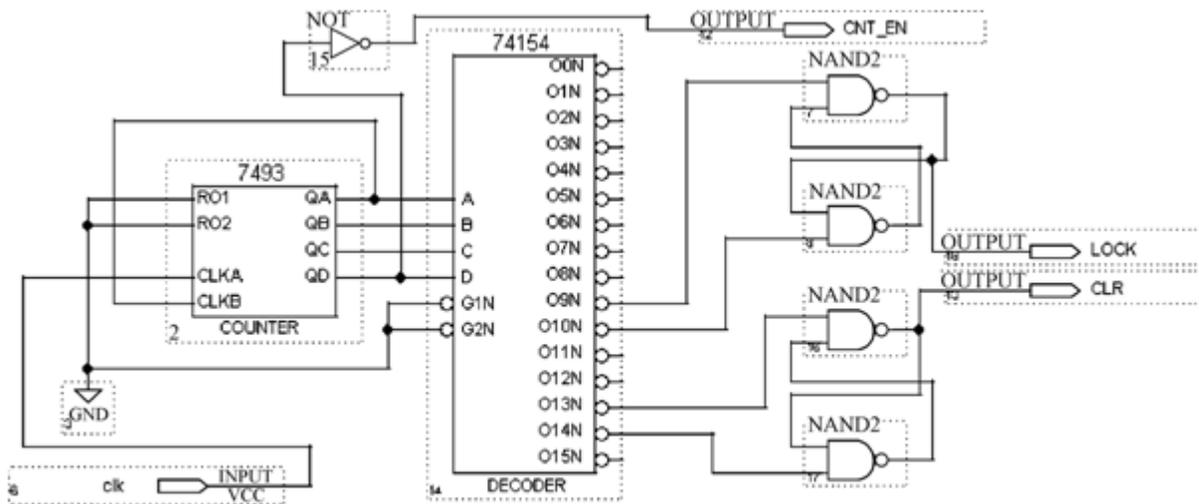


图 5-39 测频时序控制电路

实验与设计

5-1. 应用宏模块设计数字频率计

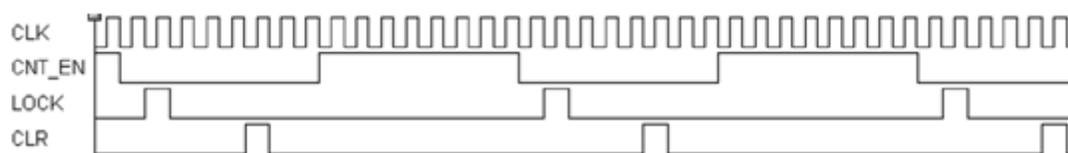


图 5-40 测频时序控制电路工作波形

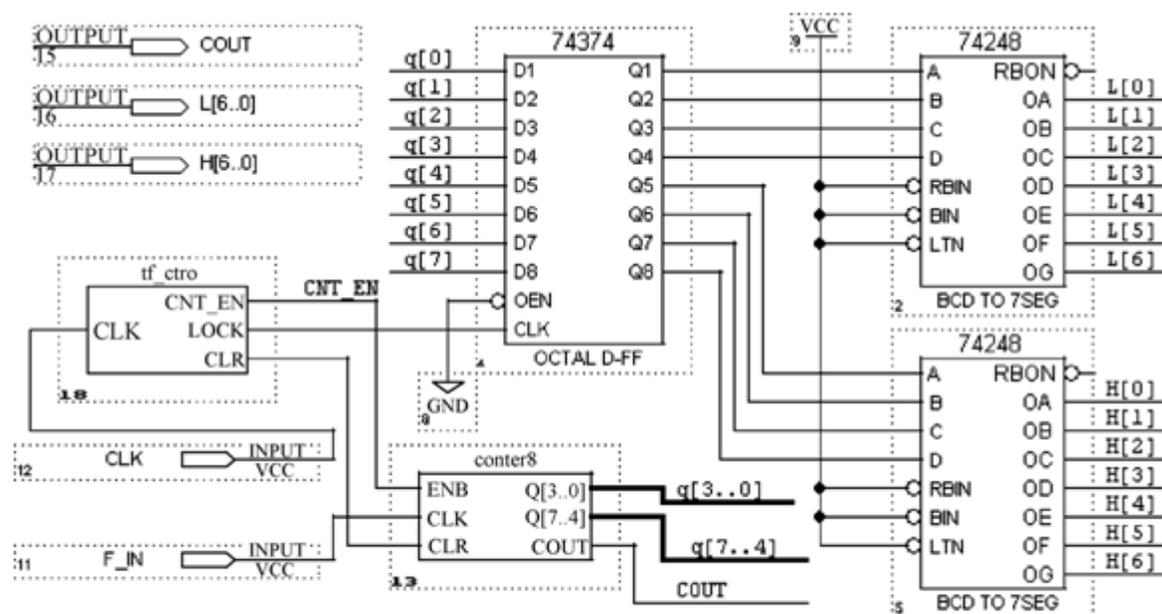


图 5-41 频率计顶层电路原理图

实验与设计

5-1. 应用宏模块设计数字频率计

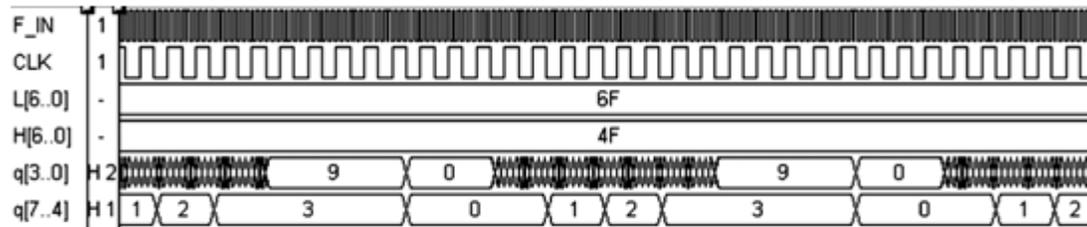


图 5-42 频率计工作时序波形

实验与设计

5-2. 计数器设计实验

5-3 数码扫描显示电路设计

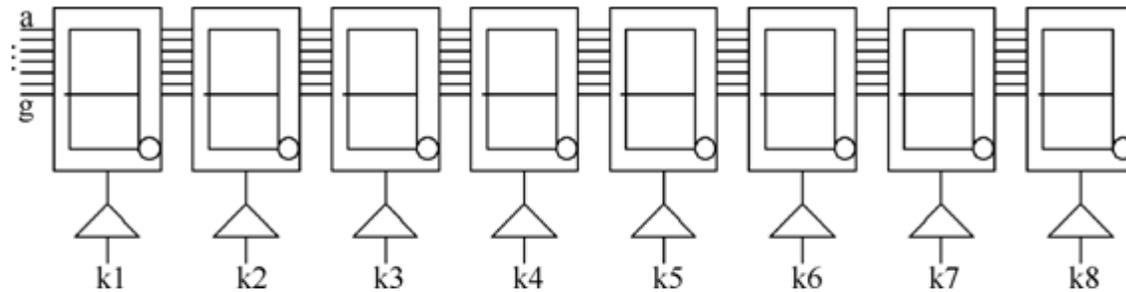


图 5-43 8 位数码扫描显示电路

实验与设计

5-4 模可控计数器设计

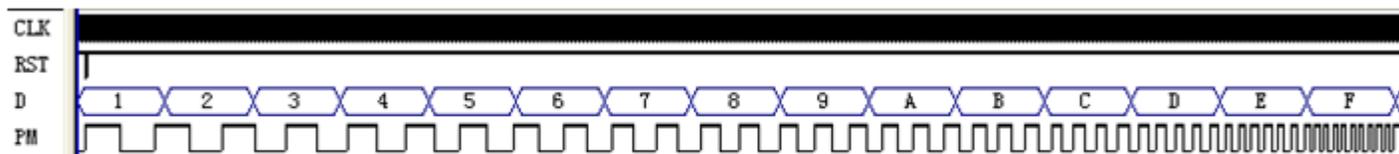


图 5-44 针对不同预置数，占空比均衡后的分频器输出

5-5 串行静态显示控制电路设计

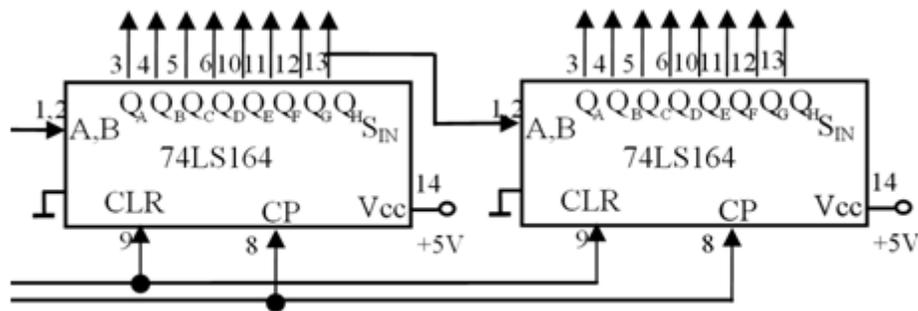
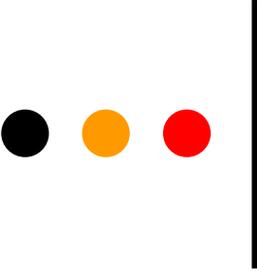


图 5-45 串/并转换数码管静态显示电路



实验与设计

5-6 高速硬件除法器设计

【例5-21】

```
module DIV16 (input CLK, input [15:0] A,B, output reg [15:0] QU,RE);
    reg [15:0] AT,BT,P,Q; integer i;
    always @(posedge CLK) begin
        AT = A ; BT = B; P = 16'H0000; Q = 16'H0000 ;
        for (i=15; i>=0; i=i-1)
            begin
                P = {P[14:0], AT[15]};
                AT = {AT[14:0], 1'B0} ; P=P-BT;
                if (P[15]==1) begin Q[i]=0; P = P+BT ; end
                else Q[i]=1 ;
            end
        end
        always @( * ) begin QU = Q; RE = P ; end
endmodule
```

5-7 不同类型的移位寄存器设计