

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

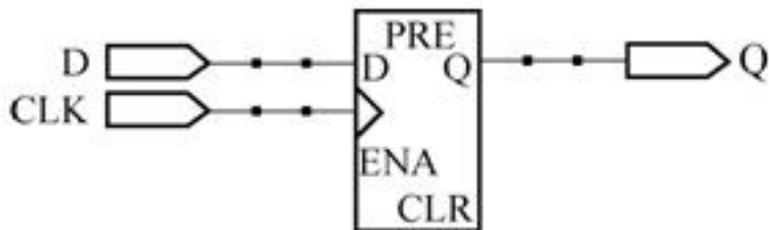


图 5-1 D 触发器模块图

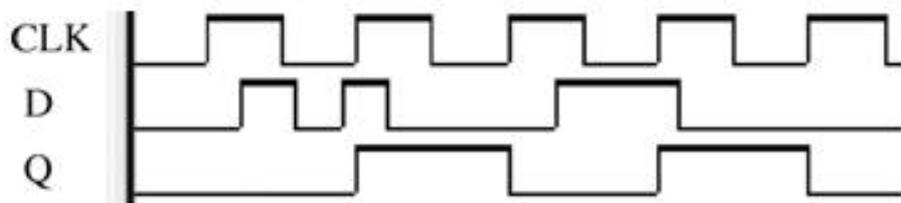


图 5-2 D 触发器时序波形

● ● ● | 5.1 基本时序元件的Verilog表述

5.1.1 基本D触发器

【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 含异步复位和时钟使能的D触发器

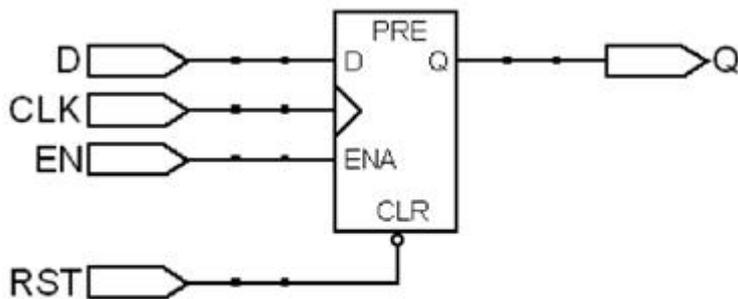


图 5-3 含使能和复位控制的 D 触发器

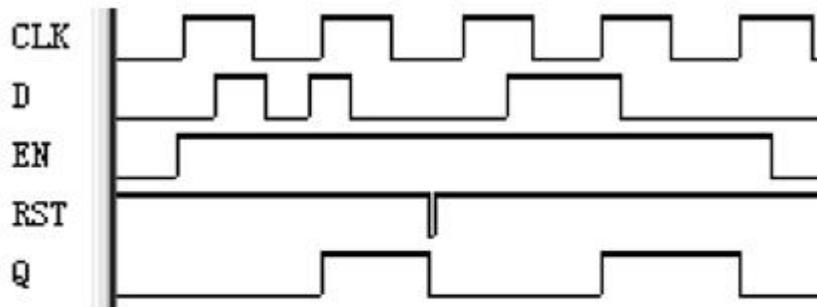


图 5-4 D 触发器的时序波形

● ● ● | 5.1 基本时序元件的Verilog表述

5.1.2 含异步复位和时钟使能的D触发器

【例 5-2】

```
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

5.1.3 含同步复位控制的D触发器

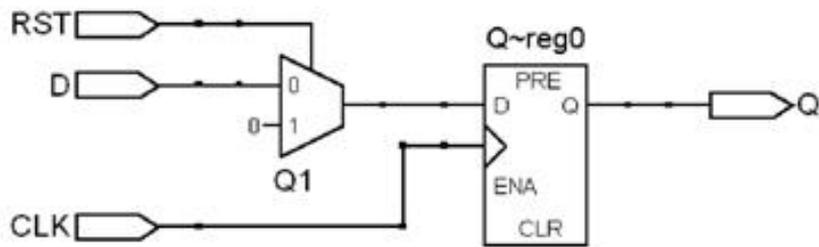


图 5-5 含同步清 0 控制的 D 触发器

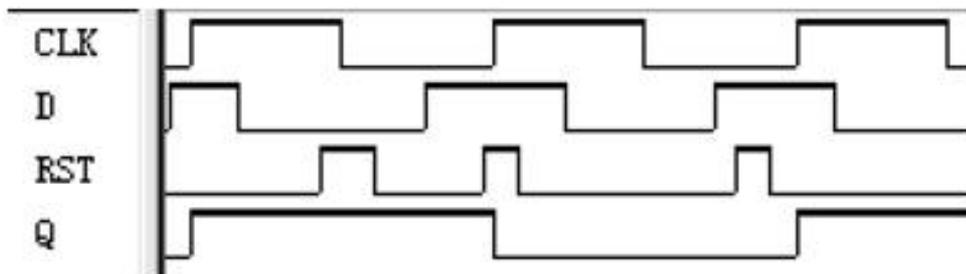


图 5-6 含同步清 0 控制 D 触发器的时序波形

5.1 基本时序元件的Verilog表述

5.1.3 含同步复位控制的D触发器

【例 5-3】

```
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 5-5】

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

【例 5-4】

```
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
```

5.1 基本时序元件的Verilog表述

5.1.4 基本锁存器

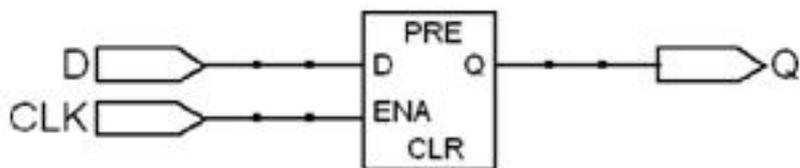


图 5-7 锁存器模块

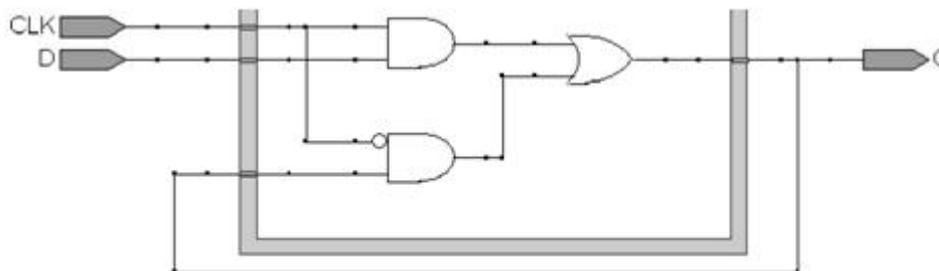


图 5-8 锁存器模块内部逻辑电路

5.1 基本时序元件的Verilog表述

5.1.4 基本锁存器

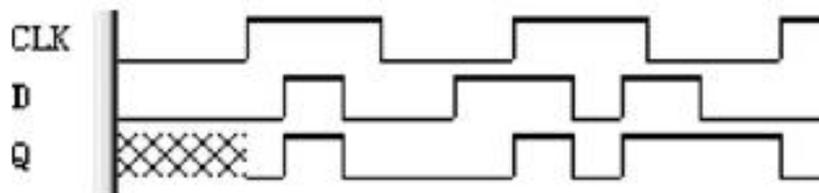


图 5-9 锁存器的时序波形

【例 5-6】

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

5.1 基本时序元件的Verilog表述

5.1.5 含清0控制的锁存器

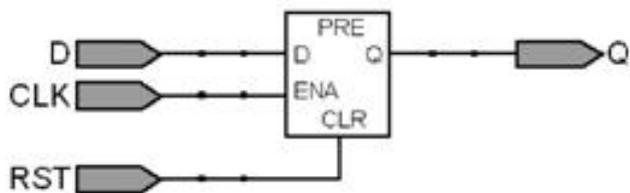


图 5-10 含异步清 0 的锁存器

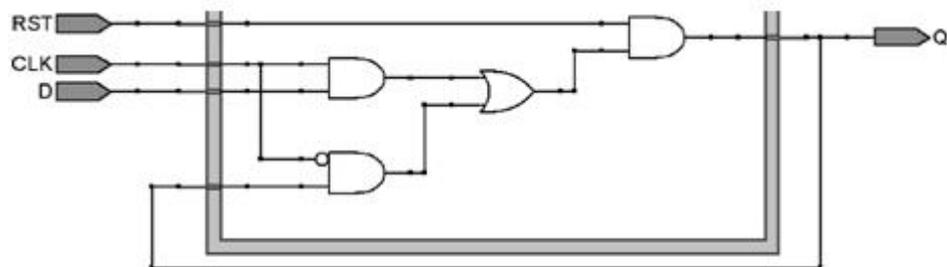


图 5-11 含异步清 0 锁存器的逻辑电路图

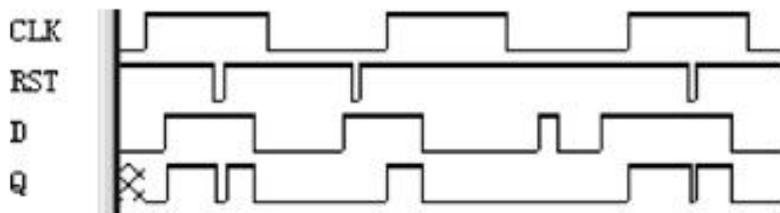


图 5-12 含异步清 0 的锁存器的仿真波形

● ● ● | 5.1 基本时序元件的Verilog表述

5.1.5 含清0控制的锁存器

【例 5-7】

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

【例 5-8】

```
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

5.1.6 异步时序电路的Verilog表述特点

【例 5-9】

```
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
```

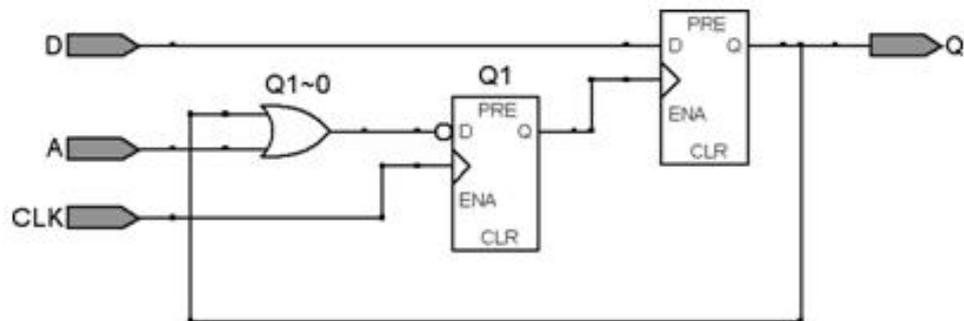


图 5-13 例 5-9 的时序电路图

5.1 基本时序元件的Verilog表述

5.1.7 时钟过程表述的特点和规律

【例 5-10】

```
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
```

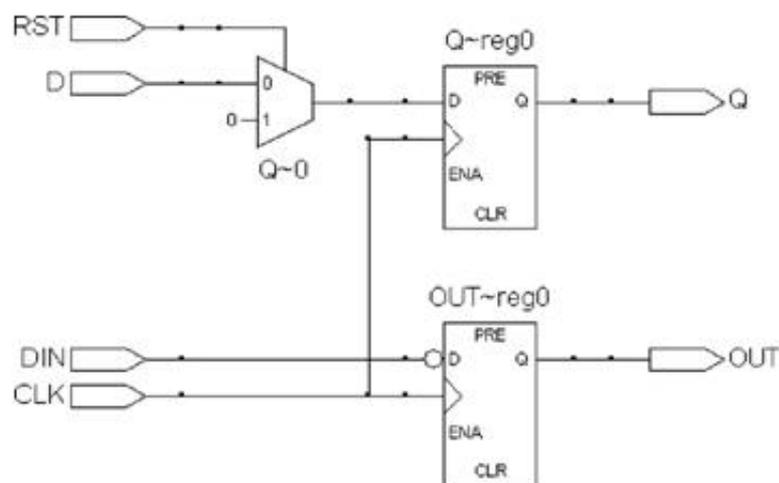


图 5-14 例 5-10 的 RTL 图

5.2 二进制计数器的Verilog表述

5.2.1 简单加法计数器

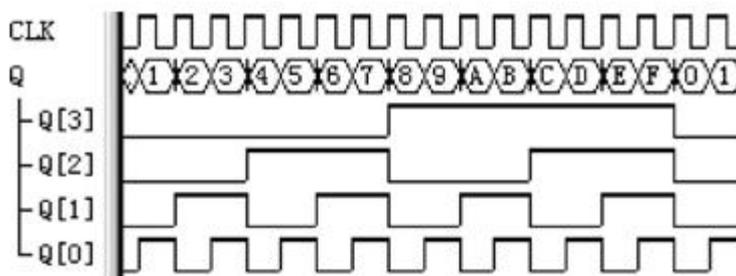


图 5-15 4 位加法计数器工作时序图

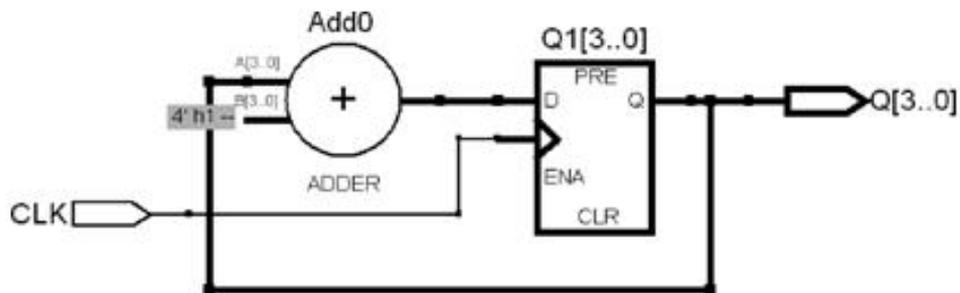


图 5-16 4 位加法计数器 RTL 电路图

5.2 二进制计数器的Verilog表述

5.2.1 简单加法计数器

【例 5-11】

```
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 5-12】

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

5.2 二进制计数器的Verilog表述

5.2.2 实用加法计数器设计

【例 5-13】

```
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
    always @(Q1) //组合过程
        if (Q1==4'h9) COUT = 1'b1; else COUT = 1'b0;
endmodule
```

5.2 二进制计数器的Verilog表述

5.2.2 实用加法计数器设计

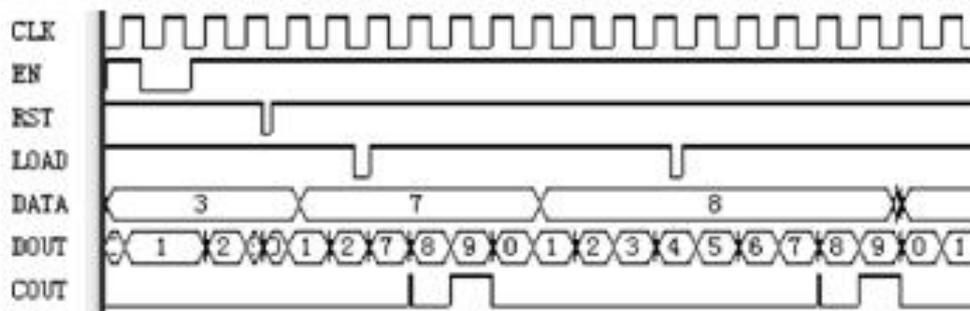


图 5-17 例 5-13 的仿真波形图

5.2 二进制计数器的Verilog表述

5.2.2 实用加法计数器设计

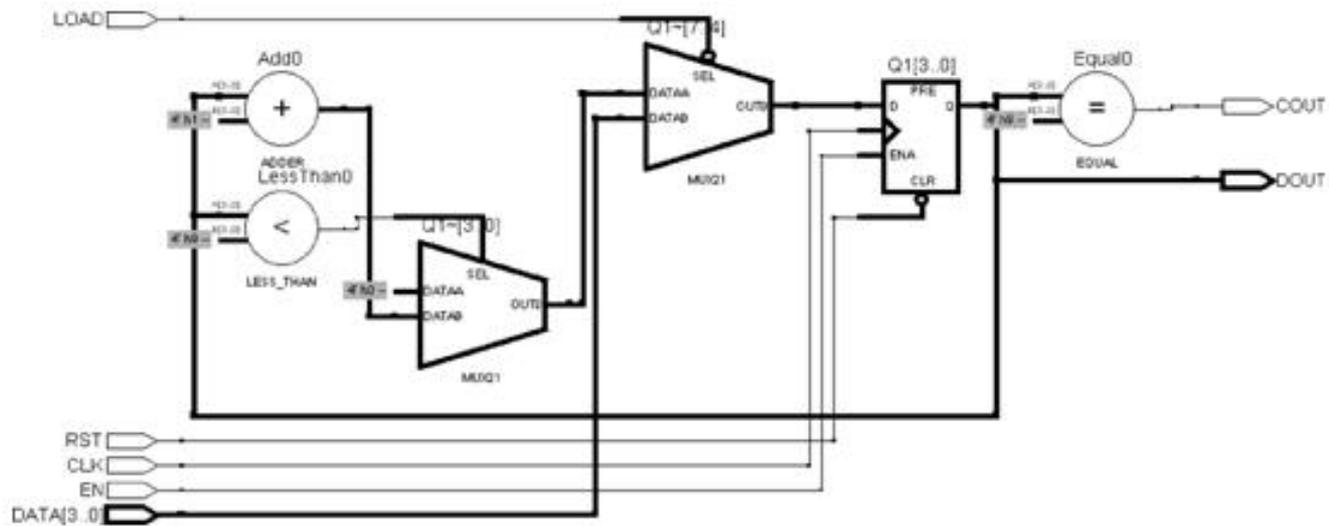


图 5-18 Quartus 对例 5-13 综合和后得到的 RTL 电路图

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

【例 5-14】

```
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
```

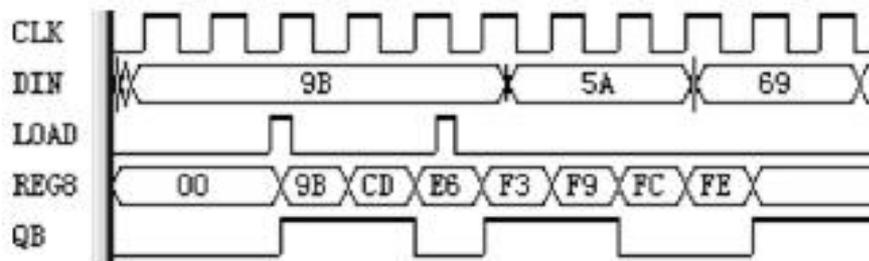


图 5-19 例 5-14 的工作时序图

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 5-15】

```
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 自动预置型计数器设计

5.4.1 同步加载计数器

【例 5-16】

```
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 自动预置型计数器设计

5.4.1 同步加载计数器

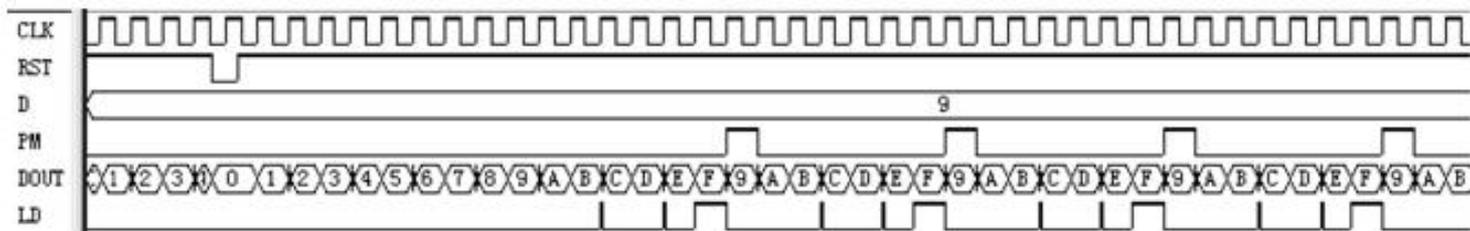


图 5-20 例 5-16 的仿真波形，同步加载模式（目标器件 EP1C3）

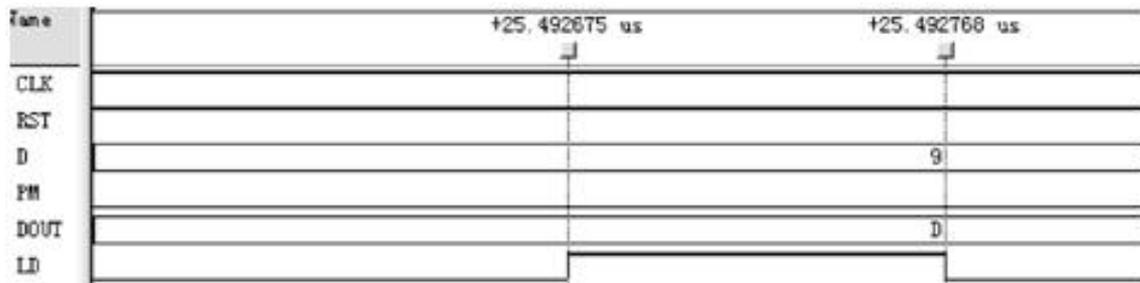


图 5-21 图 5-20 中毛刺脉冲展开后的时序，毛刺脉宽约 0.1ns

5.4 自动预置型计数器设计

5.4.2 异步加载计数器

【例 5-17】

```
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
```

5.4 自动预置型计数器设计

5.4.2 异步加载计数器

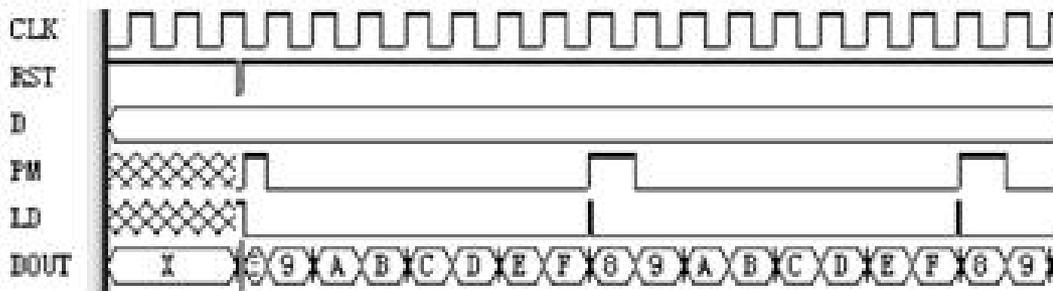


图 5-22 例 5-17 的仿真波形 (EP4CE55)

5.4 自动预置型计数器设计

5.4.3 异步清0加载计数器

【例 5-18】

```
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
```

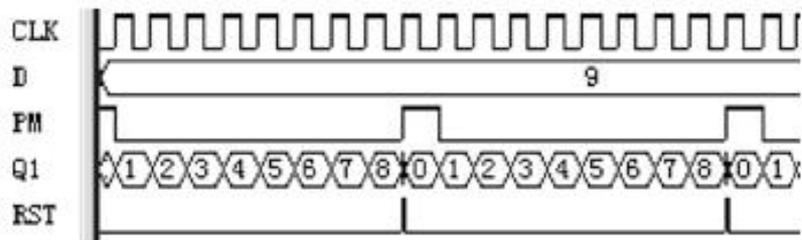


图 5-23 例 5-18 的仿真波形 (EP4CE55)

5.4 自动预置型计数器设计

5.4.4 同步清0加载计数器

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

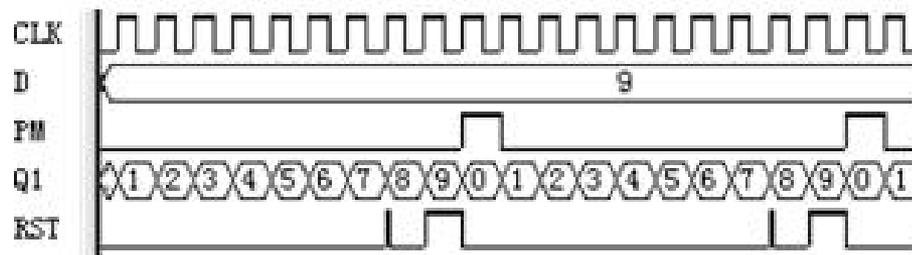


图 5-26 同步清 0 加载计数器时序波形

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

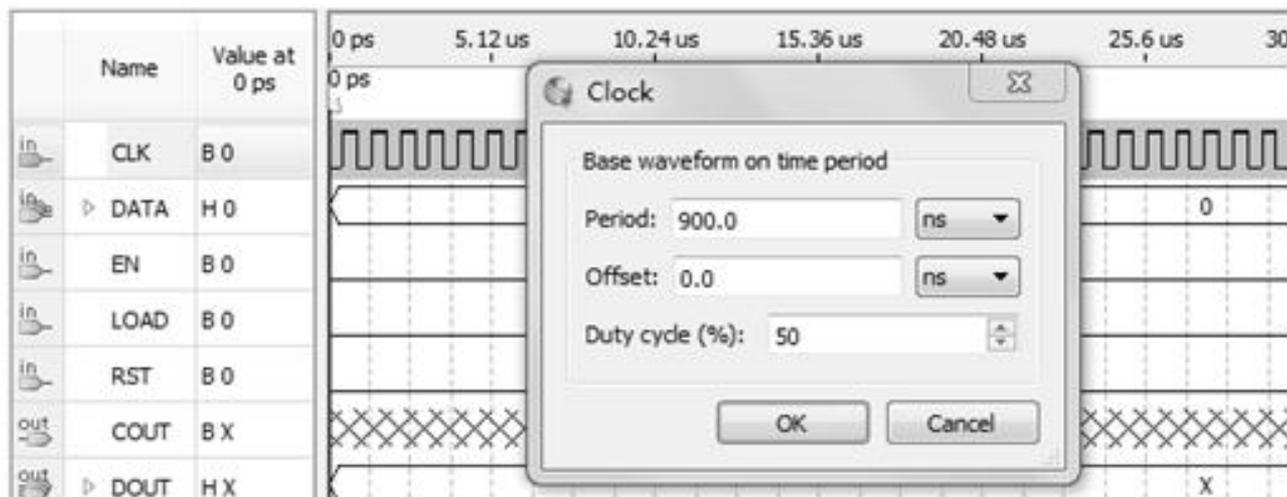


图 5-27 设置时钟 CLK 的周期

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 EP4CE55F23C8 的引脚锁定情况 (可通过附录 F.4 的列表获得)

计数器信号名	CLK	EN	LOAD	RST	DATA (3)	DATA (2)	DATA(1)
模式 0 电路控制	键 8	键 7	键 6	键 5	键 1 : D4	键 1 : D3	键 1 : D2
模式 0 电路信号	PIO7	PIO6	PIO5	PIO4	PIO11	PIO10	PIO9
对应 FPGA 引脚	AB6	Y7	AA6	AB3	AB5	AA3	W2
计数器信号名	DATA (0)	COUT	DOUT (3)	DOUT (2)	DOUT (1)	DOUT (0)	
模式 0 电路控制	键 1 : D1	数码 2 : a 段	数码 1	数码 1	数码 1	数码 1	
模式 0 电路信号	PIO8	PIO20	PIO19	PIO18	PIO17	PIO16	
对应 FPGA 引脚	U2	AA1	V2	W1	R2	U1	

5.6 SignalTap II的使用方法

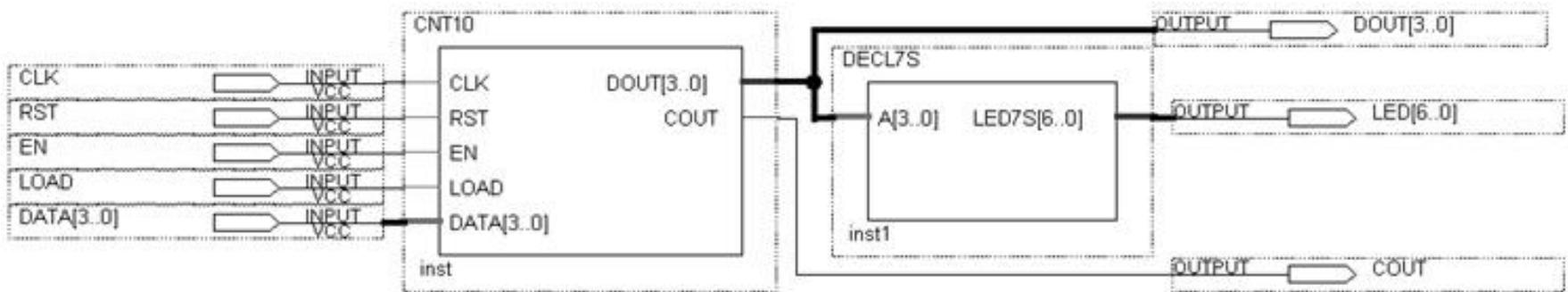


图 5-28 十进制计数器设计示例电路

5.6 SignalTap II的使用方法

1. 打开SignalTap II编辑窗口
2. 调入待测信号

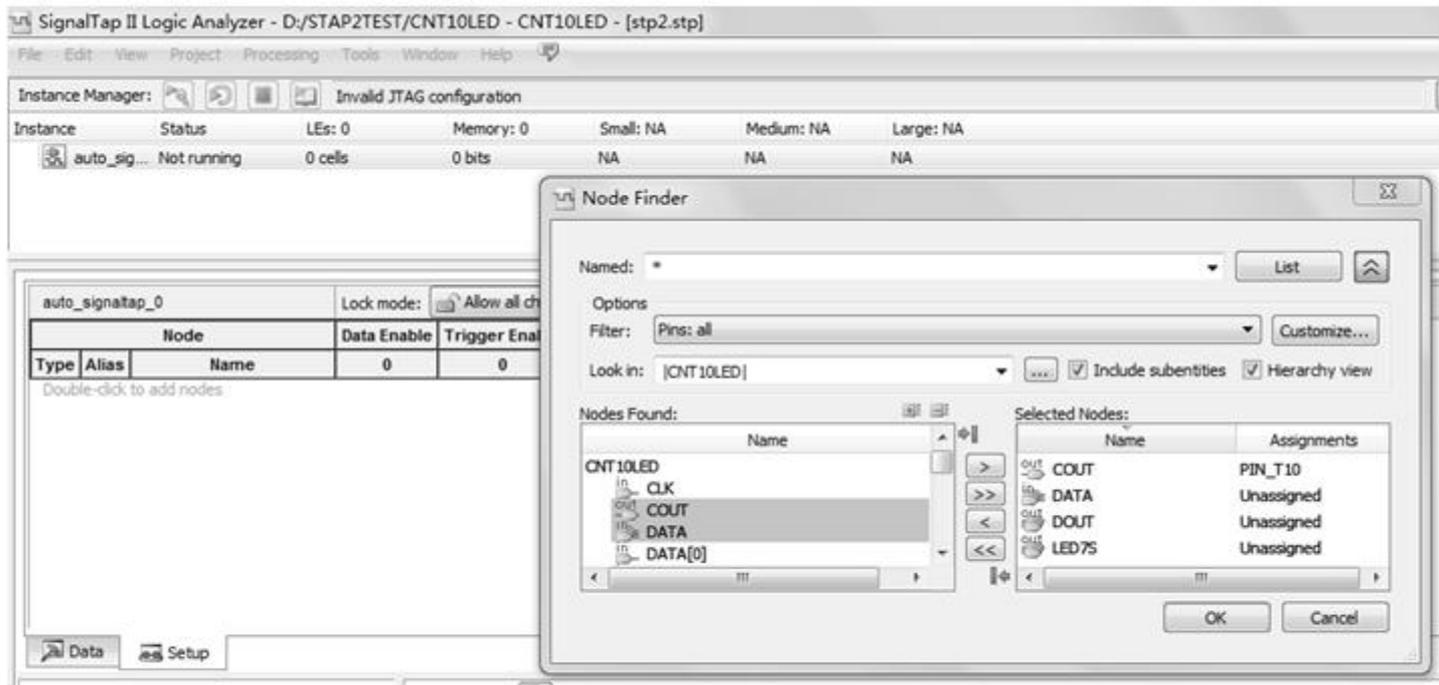


图 5-29 输入逻辑分析仪测试信号

5.6 SignalTap II 的使用方法

3. SignalTap II 参数设置

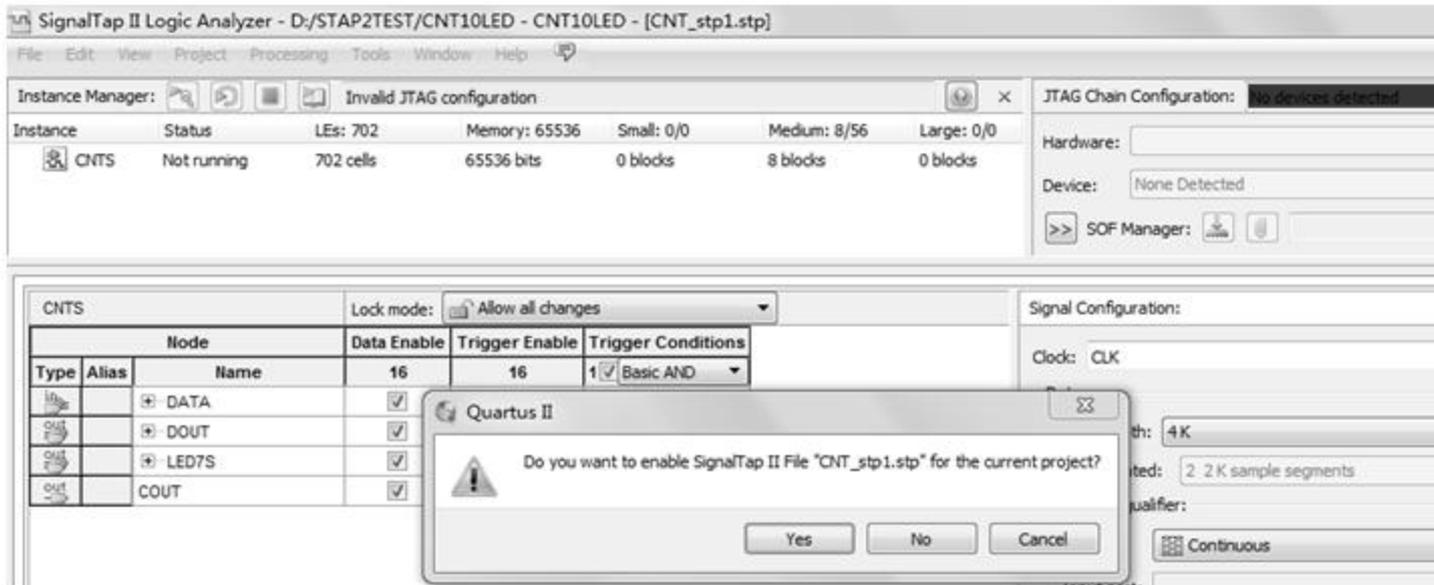


图 5-30 SignalTap II 编辑窗口

5.6 SignalTap II的使用方法

3. SignalTap II参数设置

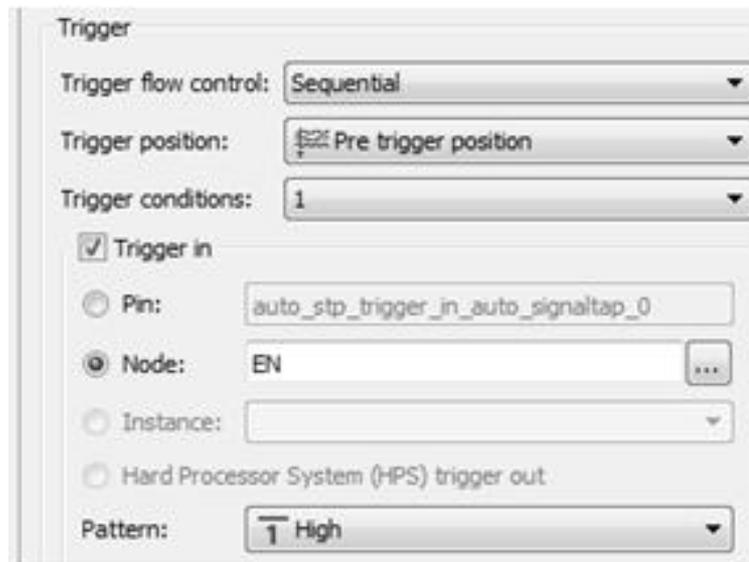


图 5-31 设置 EN 为触发信号

5.6 SignalTap II的使用方法

4. 文件存盘

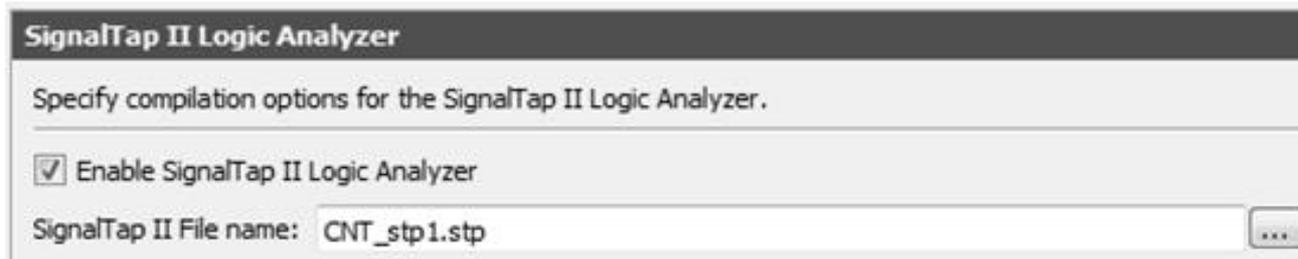


图 5-32 选择或删除 SignalTap II 文件加入综合编译

5.6 SignalTap II 的使用方法

5. 编译下载

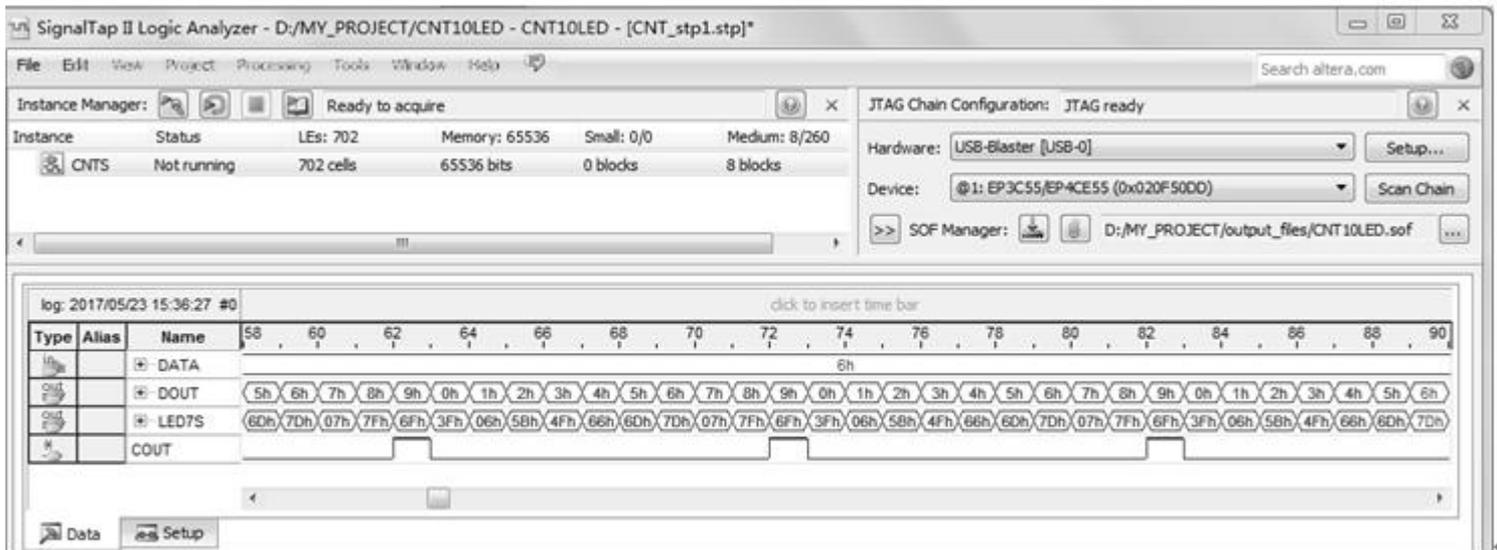


图 5-33 SignalTap II 实时数据采集显示界面

5.6 SignalTap II的使用方法

6. 启动SignalTap II进行采样与分析

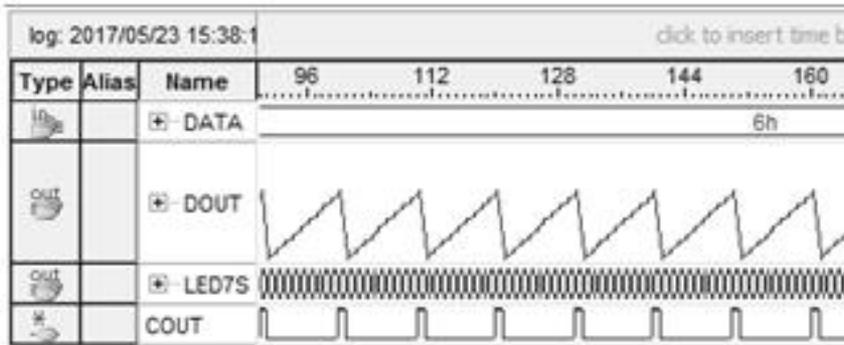


图 5-34 改变 DOUT 数据显示的方式

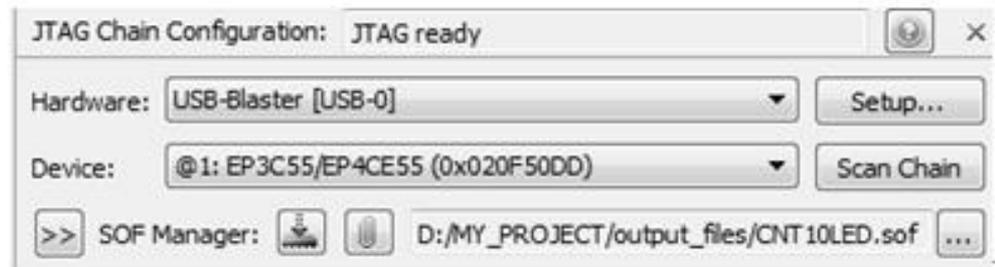
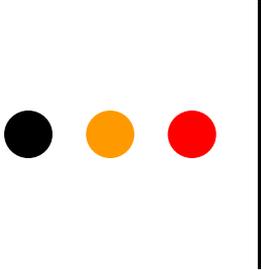


图 5-35 扫描 FPGA，并下载 SOF 文件



5.6 SignalTap II的使用方法

6. 启动SignalTap II进行采样与分析

```
input CLK    /* synthesis chip_pin = "G22" */ ;    //计数器工作时钟
input CLK0   /* synthesis chip_pin = "B11" */;     //逻辑分析仪采样时钟
```

7. SignalTap II的其他设置和控制方法

5.7 编辑SignalTap II的触发信号

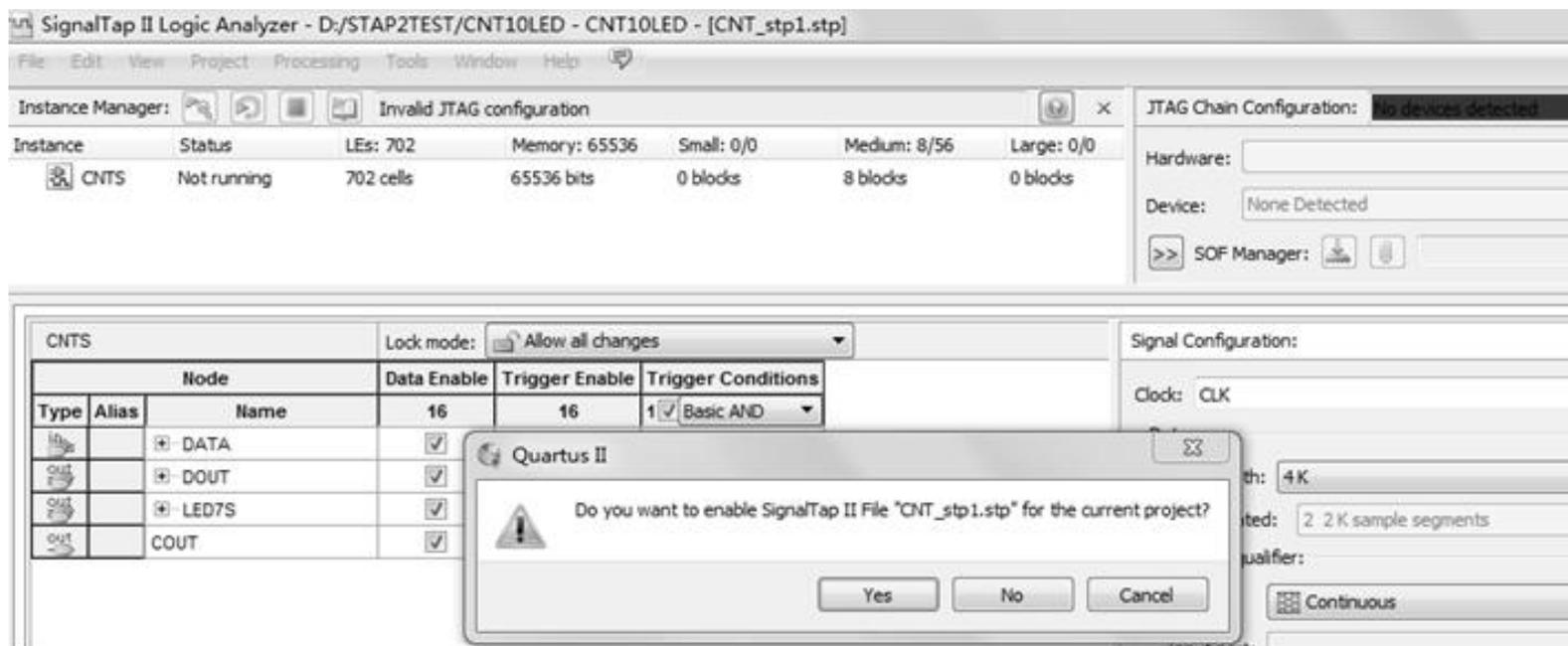


图 5-30 SignalTap II 编辑窗口

习 题

5-6 分别给出以下四个RTL图（图5-36~图5-39）的Verilog描述，注意其中的D触发器和锁存器的表述。对于图5-39的电路，分别使用if语句和条件操作语句完成表述。

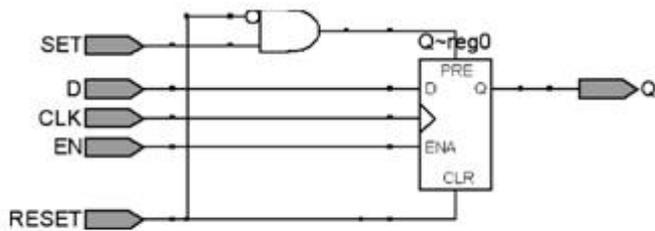


图 5-36 RTL 图 1

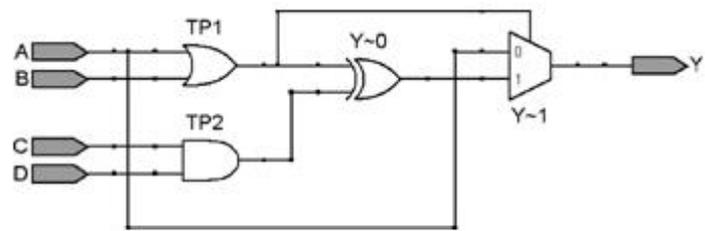


图 5-37 RTL 图 2

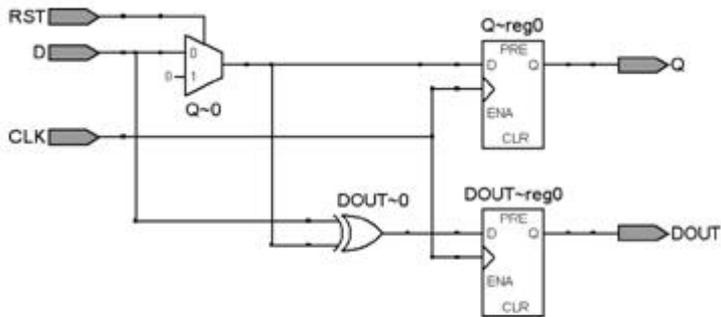


图 5-38 RTL 图 3

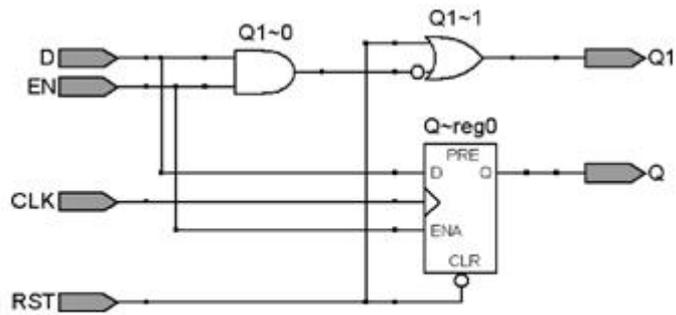


图 5-39 RTL 图 4

实验与设计

5-2 数码扫描显示电路设计

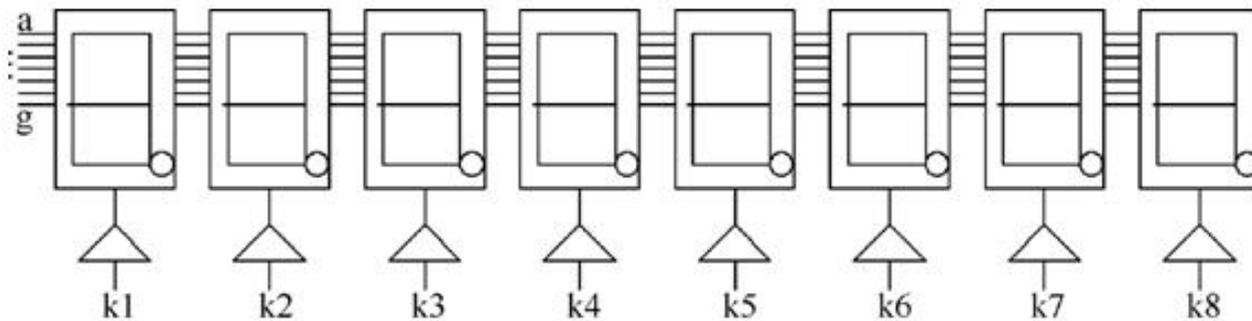


图 5-40 8 位数码扫描显示电路

实验与设计

5-1. 应用宏模块设计数字频率计

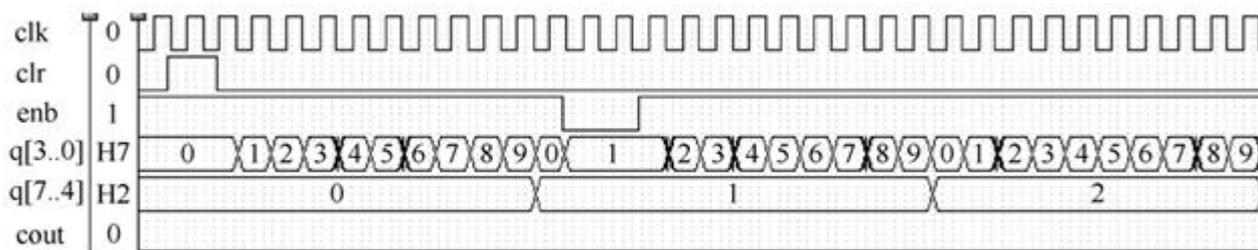


图 5-36 两位十进制计数器工作波形

实验与设计

5-1. 应用宏模块设计数字频率计

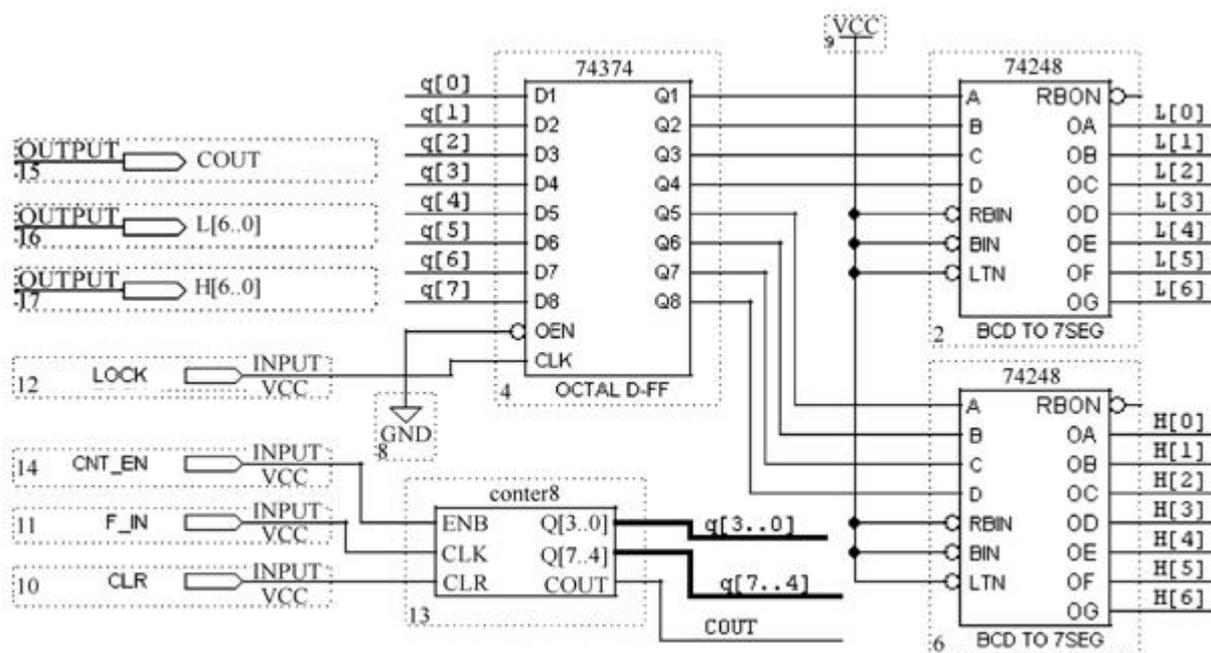
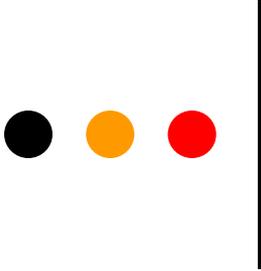


图 5-37 2 位十进制频率计顶层设计原理图文件



实验与设计

5-3 高速硬件除法器设计

【例 5-19】

```
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
    end
    always @(*) begin  QU = Q;  RE = P; end
endmodule
```

实验与设计

5-5 模可控计数器设计

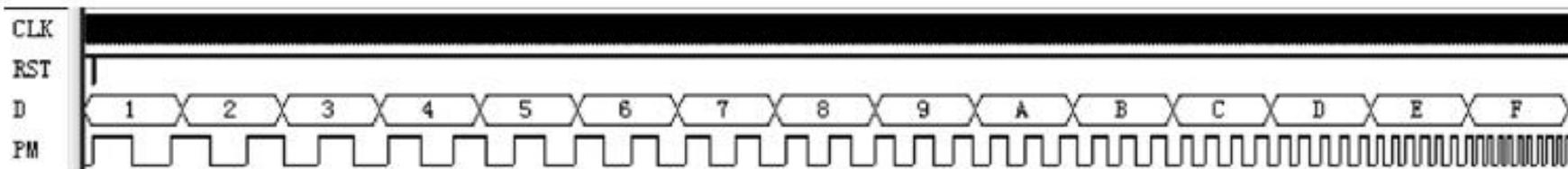


图 5-41 针对不同预置数，占空比均衡后的分频器输出

实验与设计

5-6 移位相加型8位硬件乘法器设计

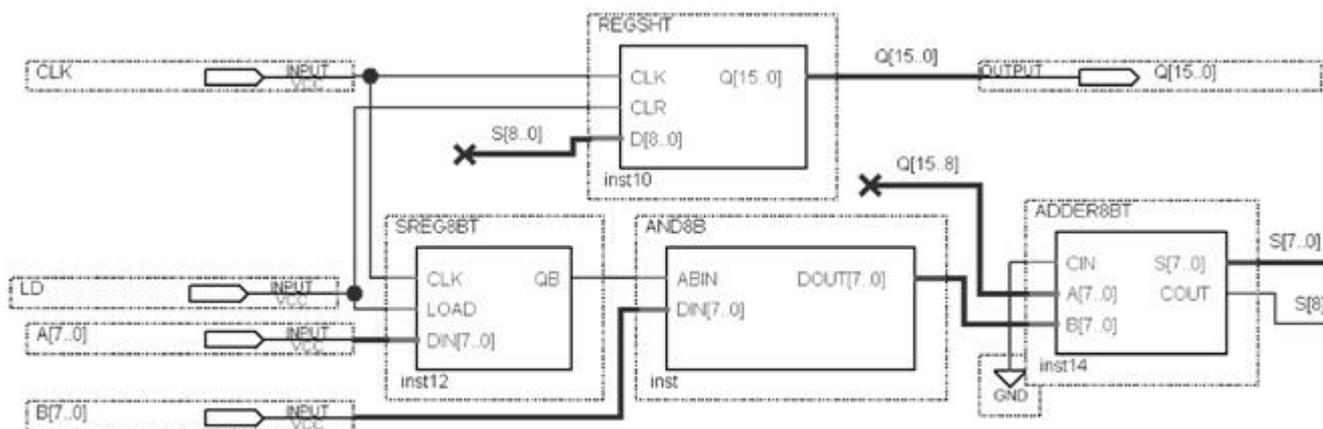


图 5-42 8 位乘法器逻辑原理图

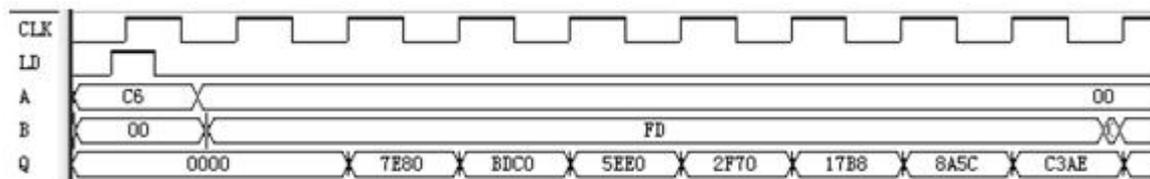


图 5-43 8 位移位相加乘法器运算逻辑波形图

实验与设计

5-7 半整数与奇数分频器设计

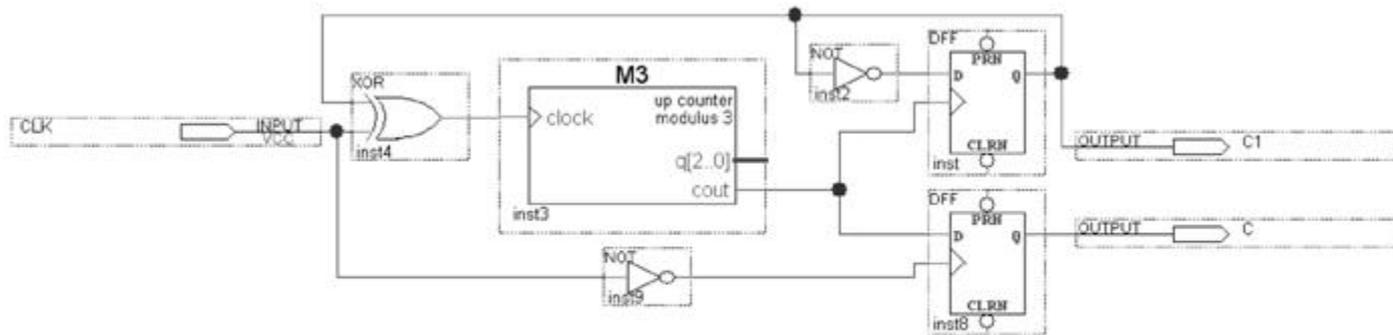


图 5-44 占空比为 50%的任意奇数次分频电路

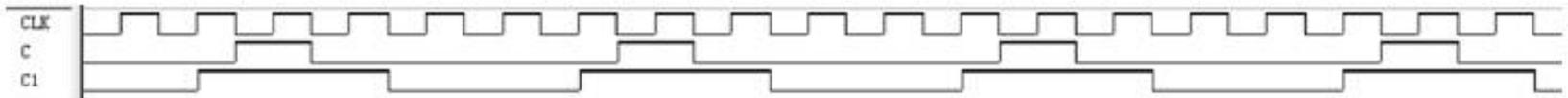
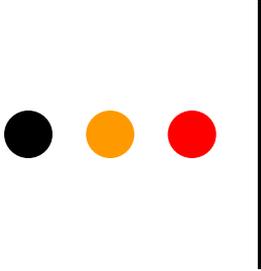


图 5-45 图 5-44 电路的仿真波形



实验与设计

5-7 半整数与奇数分频器设计

【例 5-20】 占空比为 50% 的奇数次 5 分频电路。

```
module FDIV3 (input CLK, output K_OR,K1,K2);
    reg[2:0] C1,C2; reg M1, M2;
    always @(posedge CLK) begin
        if(C1==4) C1<=0; else C1<=C1+1;
        if(C1==1) M1<=~M1; else if(C1==3) M1=~M1; end
    always @(negedge CLK) begin
        if(C2==4) C2<=0; else C2<=C2+1;
        if(C2==1) M2<=~M2; else if(C2==3) M2=~M2; end
    assign K1 = M1; assign K2 = M2;
    assign K_OR = M1 | M2;
endmodule
```

实验与设计

5-7 半整数与奇数分频器设计

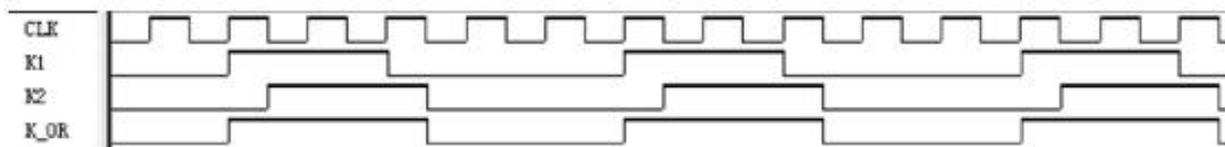


图 5-46 占空比为 50%的奇数次 5 分频电路仿真波形

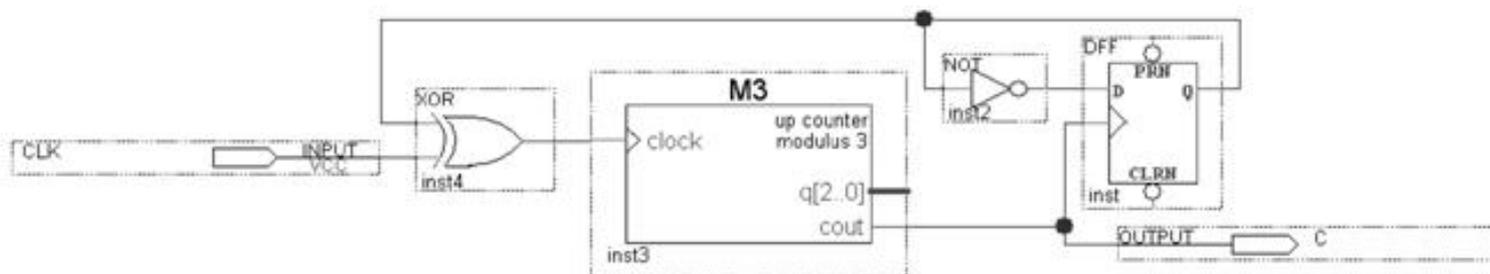


图 5-47 任意半整数分频电路

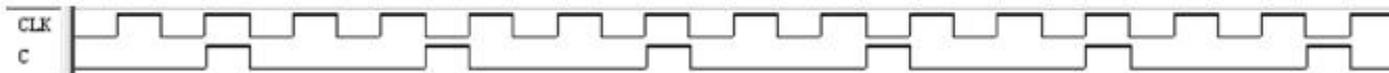
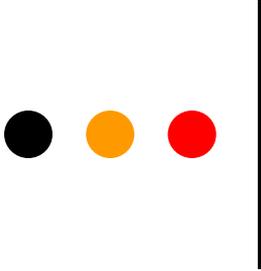


图 5-48 图 5-47 电路的仿真波形



实验与设计

5-8 基于Verilog代码的频率计设计

【例 5-21】

```
module FTCTRL (CLKK, CNT_EN, RST_CNT, LOAD);
    input CLKK;          output CNT_EN, RST_CNT, LOAD;
    wire CNT_EN, LOAD;   reg RST_CNT, Div2CLK;
    always @(posedge CLKK)
        Div2CLK <= ~Div2CLK;
    always @(CLKK or Div2CLK) begin
        if (CLKK==1'b0 & Div2CLK==1'b0) RST_CNT <= 1'b1;
        else RST_CNT <= 1'b0;          end
    assign LOAD = ~Div2CLK;
    assign CNT_EN = Div2CLK;
endmodule
```

实验与设计

5-8 基于Verilog代码的频率计设计

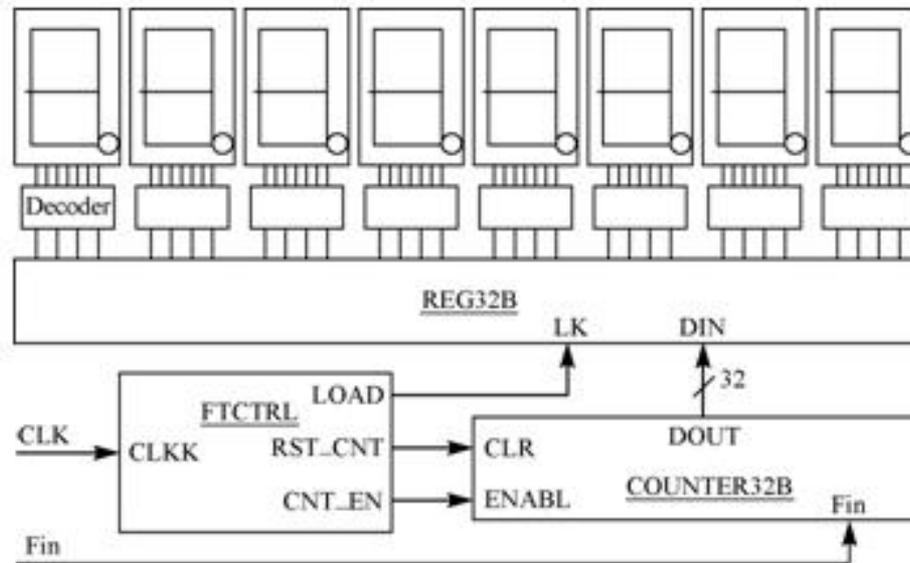


图 5-49 频率计电路框图

实验与设计

5-8 基于Verilog代码的频率计设计

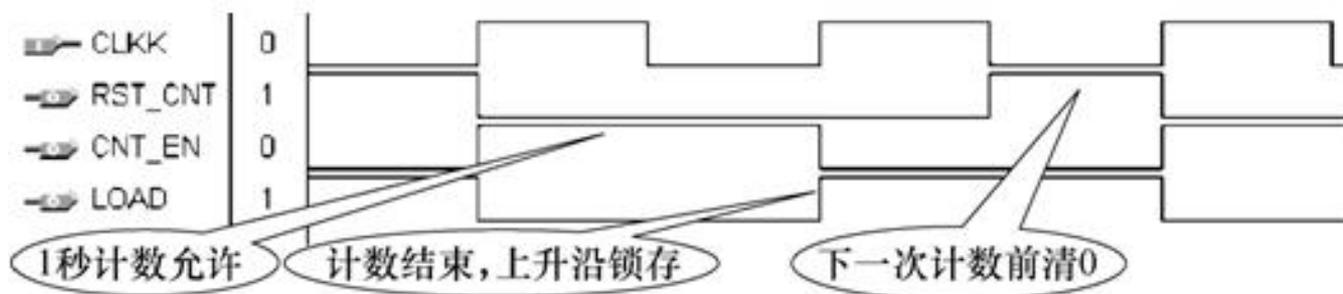


图 5-50 频率计测频控制器 FTCTRL 测控时序图



实验与设计

5-9 VGA彩条信号显示控制电路设计

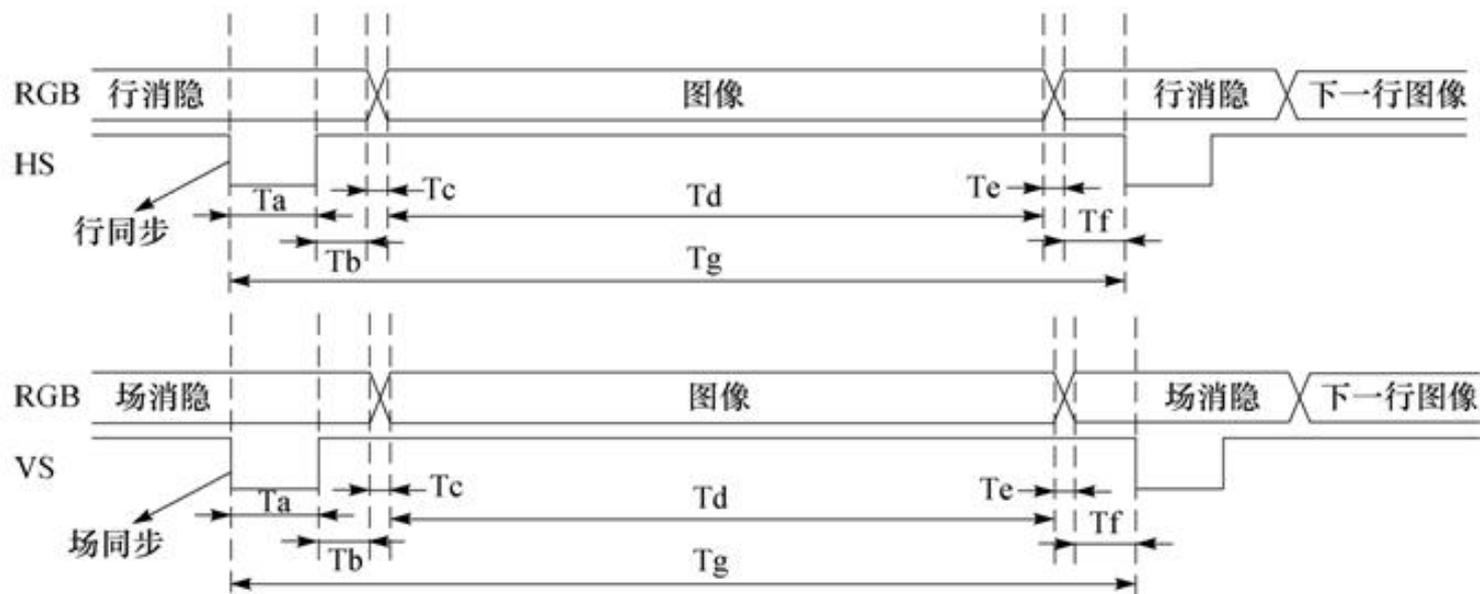
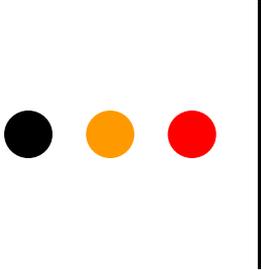


图 5-51 VGA 行扫描、场扫描时序示意图



实验与设计

5-9 VGA彩条信号显示控制电路设计

表 5-2 行扫描时序要求 (单位 : 像素 , 即输出一个像素 pixel 的时间间隔)

		行同步头			行图像		行周期
对应位置	Tf	Ta	Tb	Tc	Td	Te	Tg
时间 (pixels)	8	96	40	8	640	8	800

表 5-3 场扫描时序要求 (单位 : 行 , 即输出一行 line 的时间间隔)

		场同步头			场图像		场周期
对应位置	Tf	Ta	Tb	Tc	Td	Te	Tg
时间 (lines)	2	2	25	8	480	8	525



实验与设计

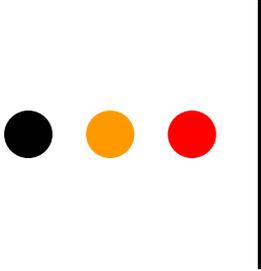
5-9 VGA彩条信号显示控制电路设计

表 5-4 颜色编码

颜色	黑	蓝	红	品	绿	青	黄	白
R	0	0	0	0	1	1	1	1
G	0	0	1	1	0	0	1	1
B	0	1	0	1	0	1	0	1

表 5-5 彩条信号发生器 3 种显示模式

1	横彩条	1: 白黄青绿品红蓝黑	2: 黑蓝红品绿青黄白
2	竖彩条	1: 白黄青绿品红蓝黑	2: 黑蓝红品绿青黄白
3	棋盘格	1: 棋盘格显示模式 1	2: 棋盘格显示模式 2



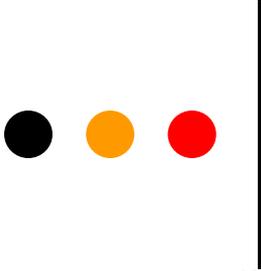
实验与设计

5-9 VGA彩条信号显示控制电路设计

【例 5-22】

```
module VGA_COLOR_LINE (CLK,MD,HS,VS,R,G,B); //VGA 显示器 彩条发生器
    input CLK , input MD ;
    output HS , VS , R , G , B;

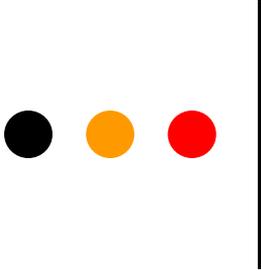
    wire R,G,B,VS,HS;           //红 , 绿 , 蓝信号 , 和场同步 , 行同步信号
    wire FCLK, CCLK;
    reg HS1, VS1; reg[1:0] MMD; reg[4:0] FS;
    reg[4:0] CC;                 //行同步 , 横彩条生成
    reg[8:0] LL;                 //场同步 , 竖彩条生成
    reg[3:1] GRBX,GRBY,GRBP;     //X 横彩条 , Y 竖彩条
    wire[3:1] GRB;
    assign GRB[2] = (GRBP[2] ^ MD) & HS1 & VS1;
    assign GRB[3] = (GRBP[3] ^ MD) & HS1 & VS1;
    assign GRB[1] = (GRBP[1] ^ MD) & HS1 & VS1;
    always @(posedge MD) begin
        if (MMD==2'b10) MMD<=2'b00; else MMD<=MMD+1; end //3 种模式
    always @(MMD) begin
        if (MMD == 2'b00) GRBP <= GRBX;           //选择横彩条
```



实验与设计

```
else if (MMD == 2'b01) GRBP <= GRBY;           //选择竖彩条
else if (MMD == 2'b10) GRBP <= GRBX ^ GRBY;   //产生棋盘格
else GRBP <= 3'b000;   end

always @(posedge CLK ) begin                    //20MHz 21 分频
    if (FS==20) FS<=0; else FS<=(FS+1); end
always @(posedge FCLK) begin
    if (CC==29) CC<=0; else CC<=CC+1;   end
always @(posedge CCLK) begin
    if (LL==481) LL<=0; else LL<=LL+1;   end
always @(CC or LL) begin
    if (CC > 23) HS1<=1'b0; else HS1<=1'b1;           //行同步
    if (LL > 479) VS1<=1'b0; else VS1<=1'b1; end     //场同步
always @(CC or LL) begin
    if (CC < 3) GRBX <= 3'b111;                       //横彩条
    else if (CC < 6) GRBX <= 3'b110;
    else if (CC < 9) GRBX <= 3'b101;
    else if (CC < 12) GRBX <= 3'b100;
    else if (CC < 15) GRBX <= 3'b011;
    else if (CC < 18) GRBX <= 3'b010;
    else if (CC < 21) GRBX <= 3'b001;
    else GRBX <= 3'b000;
    if (LL < 60) GRBY <= 3'b111;                       //竖彩条
```



实验与设计

5-9 VGA彩条信号显示控制电路设计

```
    else if (LL < 120)  GRBY <= 3'b110;
    else if (LL < 180)  GRBY <= 3'b101;
    else if (LL < 240)  GRBY <= 3'b100;
    else if (LL < 300)  GRBY <= 3'b011;
    else if (LL < 360)  GRBY <= 3'b010;
    else if (LL < 420)  GRBY <= 3'b001;
    else    GRBY <= 0;    end

    assign HS = HS1;    assign FCLK = FS[3];
    assign HS = HS1;    assign VS = VS1 ;
    assign R = GRB[2];  assign G = GRB[3];
    assign B = GRB[1];  assign CCLK = CC[4];
endmodule
```

实验与设计

5-9 VGA彩条信号显示控制电路设计

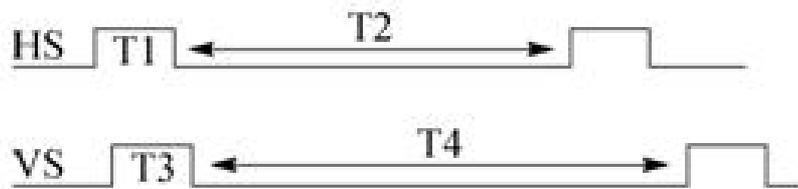


图 5-52 HS 和 VS 的时序图

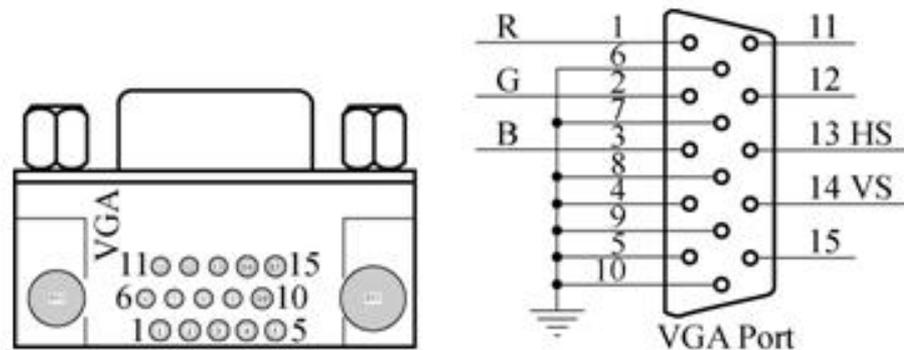


图 5-53 VGA 接口电路图，左接口从上往下看