EDA技术实用教程

第5章

时序电路的VHDL设计

5.1 基本时序元件的VHDL表述

5.1.1 D触发器的VHDL表述

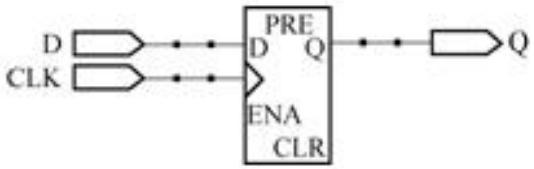


图 5-1 D 触发器模块图

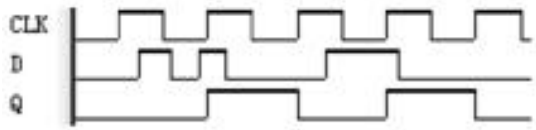


图 5-2 D 触发器时序波形

5.1 基本时序元件的VHDL表述

5.1.1 D触发器的VHDL表述

【例 5-1】

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
ENTITY DFF1 IS
  PORT (CLK, D : IN STD_LOGIC ;
              Q : OUT STD_LOGIC );
 END ;
 ARCHITECTURE bhv OF DFF1 IS
  SIGNAL Q1 : STD_LOGIC ;
  BEGIN
   PROCESS (CLK,Q1)    BEGIN
     IF  CLK'EVENT AND CLK = '1'
            THEN  Q1 <= D ;
    END IF;
    END PROCESS ;
       Q <= Q1 ;
END bhv;
```

5.1 基本时序元件的VHDL表述

5.1.1 D触发器的VHDL表述

1. 上升沿检测表达式和信号属性函数EVENT

<信号名>'EVENT

2. 不完整条件语句与时序电路

【例 5-2】

```
ENTITY COMP_BAD IS
  PORT( a, b  : IN BIT;   q  : OUT BIT    );
  END ;
ARCHITECTURE one OF COMP_BAD IS
  BEGIN
CMP:  PROCESS (a,b)  BEGIN        -- CMP 是当前进程的标号或名称，不参与综合
   IF  a>b  THEN  q<='1' ;
ELSIF a<b  THEN  q<='0' ; END IF;-- 注意未提及当 a=b 时，q 作何操作
  END PROCESS ;
END ;
```

5.1 基本时序元件的VHDL表述

5.1.1 D触发器的VHDL表述

2. 不完整条件语句与时序电路

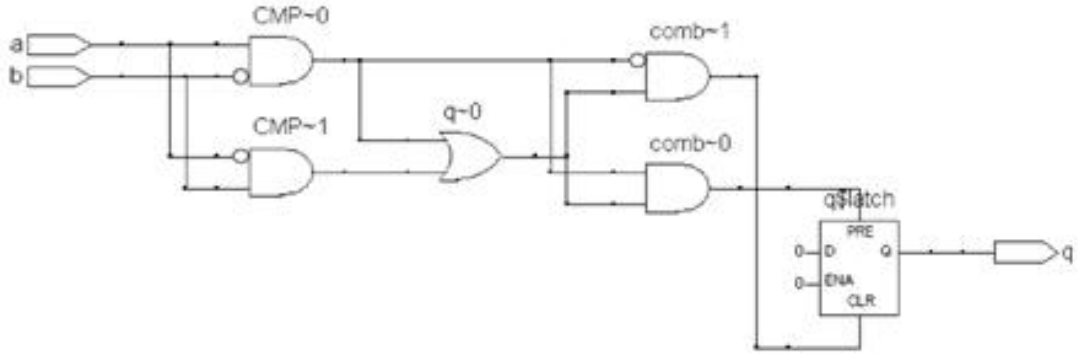


图 5-3 例 5-2 综合后的 RTL 电路图

【例 5-3】

```
IF a>b THEN q <= '1'; ELSE q<='0' ; END IF;
```

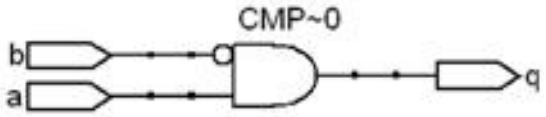


图 5-4 例 5-3 的 RTL 电路图

5.1 基本时序元件的VHDL表述

5.1.2 含异步复位和时钟使能的D触发器的VHDL表述

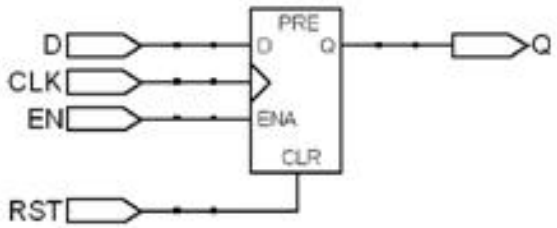


图 5-5 含使能和复位控制的 D 触发器

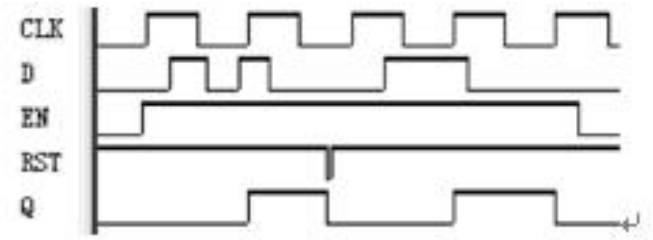


图 5-6 图 5-5 的 D 触发器的时序图

5.1 基本时序元件的VHDL表述

5.1.2 含异步复位和时钟使能的D触发器的VHDL表述

【例 5-4】

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
ENTITY DFF2 IS
  PORT (CLK,RST,EN,D : IN STD_LOGIC;     Q : OUT STD_LOGIC );
END ;
ARCHITECTURE bhv OF DFF2 IS
    SIGNAL Q1 : STD_LOGIC ;
    BEGIN
   PROCESS (CLK,Q1,RST,EN)  BEGIN
     IF RST='1' THEN Q1<='0'; ELSIF  CLK'EVENT AND CLK = '1'  THEN
       IF  EN='1'  THEN   Q1<=D ;  END IF;
     END IF;
    END PROCESS ;
      Q <= Q1 ;
END bhv;
```

5.1 基本时序元件的VHDL表述

5.1.3 含同步复位控制的D触发器的VHDL表述

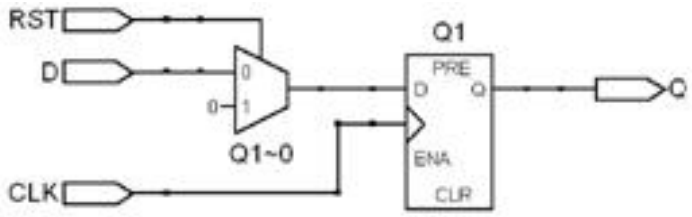


图 5-7 含同步清 0 控制的 D 触发器

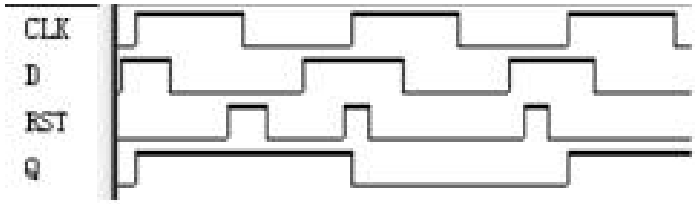


图 5-8 含同步清 0 控制 D 触发器的时序

5.1 基本时序元件的VHDL表述

5.1.3 含同步复位控制的D触发器的VHDL表述

【例 5-5】

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
ENTITY DFF3 IS
  PORT (CLK,RST,D : IN STD_LOGIC;    Q : OUT STD_LOGIC );
 END ;
 ARCHITECTURE bhv OF DFF3 IS
  SIGNAL Q1 : STD_LOGIC ;
  BEGIN
   PROCESS (CLK,Q1,RST)   BEGIN
   IF  CLK'EVENT AND CLK = '1'  THEN
       IF RST='1' THEN Q1<='0'; ELSE  Q1 <= D;  END IF;
   END IF;
    END PROCESS ;
Q <= Q1 ;
END bhv;
```

5.1 基本时序元件的VHDL表述

5.1.4 基本锁存器的VHDL表述

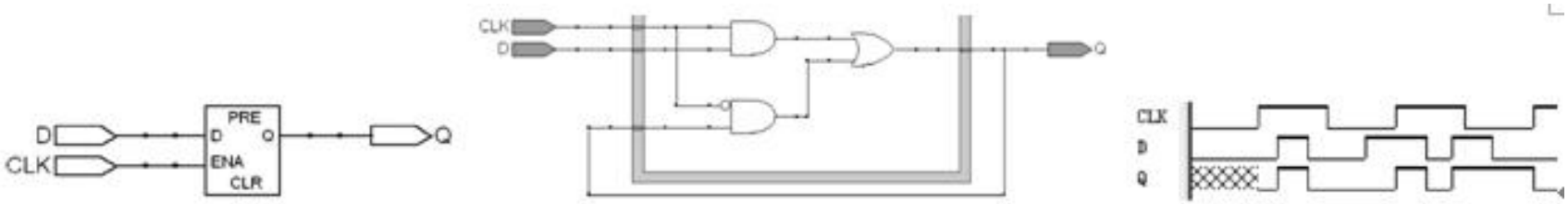


图 5-9　基本锁存器模块、内部电路结构以及锁存器时序波形图

5.1 基本时序元件的VHDL表述

5.1.4 基本锁存器的VHDL表述

【例 5-6】

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
ENTITY LTCH2 IS
  PORT (CLK,D : IN STD_LOGIC;   Q : OUT STD_LOGIC);
 END ;
 ARCHITECTURE bhv OF LTCH2 IS
   BEGIN
   PROCESS (CLK, D)   BEGIN
      IF  CLK='1'  THEN  Q <= D;  END IF;
   END PROCESS ;
END  bhv;
```

5.1 基本时序元件的VHDL表述

5.1.4 基本锁存器的VHDL表述

【例 5-7】

```
PROCESS (CLK)      BEGIN
    IF  CLK='1'  THEN  Q <= D;
    END IF;
  END PROCESS ;
```

5.1 基本时序元件的VHDL表述

5.1.5 含清0控制的锁存器的VHDL表述

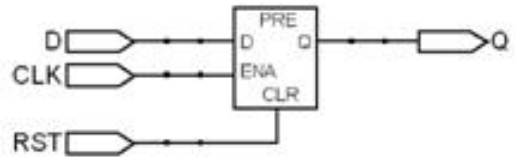


图 5-10　含异步清 0 的锁存器

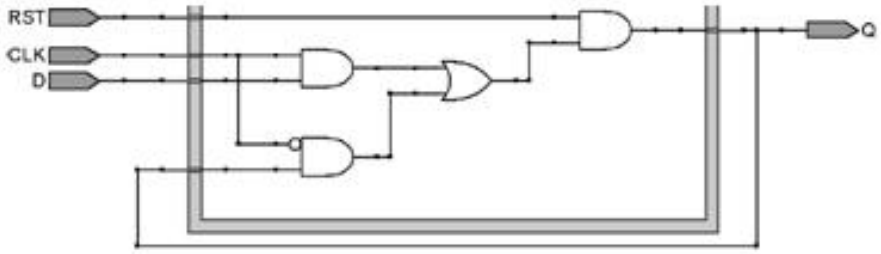


图 5-11　含异步清 0 锁存器的逻辑电路图

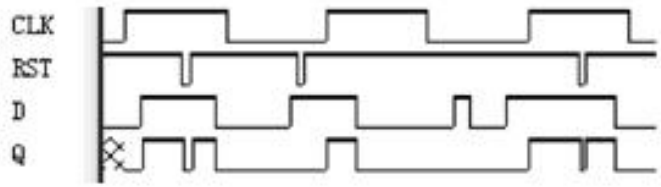


图 5-12　含异步清 0 的锁存器的仿真波形

5.1 基本时序元件的VHDL表述

5.1.5 含清0控制的锁存器的VHDL表述

【例 5-8】

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
ENTITY LTCH3 IS
  PORT (CLK,D,RST : IN STD_LOGIC;    Q : OUT STD_LOGIC );
 END ;
 ARCHITECTURE bhv OF LTCH3 IS
   BEGIN
   PROCESS (CLK,D,RST)    BEGIN
      IF RST='1' THEN Q<='0';
        ELSIF  CLK = '1'  THEN  Q <= D;  END IF;
   END PROCESS ;
END bhv;
```

5.1 基本时序元件的VHDL表述

5.1.6 VHDL实现时序电路的不同表述

```
CLK'EVENT AND (CLK='1') AND (CLK'LAST_VALUE='0')
```

【例 5-9】

```
IF  (CLK'EVENT AND CLK='1') AND (CLK'LAST_VALUE='0')
    THEN  Q <= D ;    --确保 CLK 的变化是一次上升沿的跳变
END IF;
```

【例 5-10】

```
IF  CLK='1' AND CLK'LAST_VALUE ='0'  THEN  Q <= D;  END IF;
```

5.1 基本时序元件的VHDL表述

5.1.6 VHDL实现时序电路的不同表述

【例 5-11】

```
IF  rising_edge(CLK) -- 注意使用此函数必须打开 STD_LOGIC_1164 程序包
    THEN  Q1 <= D ;
END IF;
```

【例 5-12】

```
WREG: PROCESS  BEGIN
       wait until CLK = '1'   ;     --利用 wait 语句
        Q <= D ;
      END PROCESS;
```

```
G1 : BLOCK (CLK'EVENT AND clk='1')
begin  q<=GUARDED d;    END BLOCK G1;
```

5.1 基本时序元件的VHDL表述

5.1.7 双边沿触发时序电路设计讨论

在同一进程中同一信号的双边沿操作

```
PROCESS (CLK)     BEGIN
   IF  RISING_EDGE(CLK)  THEN
       Q1 <= Q1 + 1 ;
 ELSIF  FALLING_EDGE(CLK)  THEN
       Q1 <= Q1 + 1 ;
   END IF;
END PROCESS ;
```

在不同进程中同一信号的双边沿操作

```
PROCESS (CLK)   BEGIN
   IF  RISING_EDGE(CLK)  THEN
      Q1 <= Q1 + 1 ;   END IF;
END PROCESS ;
PROCESS (CLK)  BEGIN
    IF  FALLING_EDGE(CLK)  THEN
      Q1 <= Q1 + 1 ;    END IF;
END PROCESS ;
```

5.2 计数器的VHDL设计

5.2.1 4位二进制加法计数器设计

【例 5-13】

```
ENTITY CNT4 IS
  PORT ( CLK : IN BIT ;
         Q  : BUFFER INTEGER RANGE 15 DOWNTO 0 );
  END ;
ARCHITECTURE bhv OF CNT4 IS
  BEGIN
  PROCESS (CLK) BEGIN
    IF  CLK'EVENT AND CLK = '1'  THEN   Q<=Q+1 ;   END IF;
  END PROCESS ;
END bhv;
```

5.2 计数器的VHDL设计

5.2.2 计数器更常用的VHDL表达方式

【例 5-14】

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
USE IEEE.STD_LOGIC_UNSIGNED.ALL ;
ENTITY CNT4 IS
PORT (CLK : IN STD_LOGIC ;
       Q  : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) ) ;
END ;
ARCHITECTURE bhv OF CNT4 IS
   SIGNAL Q1 : STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
  PROCESS(CLK)  BEGIN
     IF  CLK'EVENT AND CLK = '1'  THEN  Q1<=Q1+1;  END IF;
  END PROCESS ;
     Q <= Q1 ;
END bhv;
```

5.2 计数器的VHDL设计

5.2.2 计数器更常用的VHDL表达方式

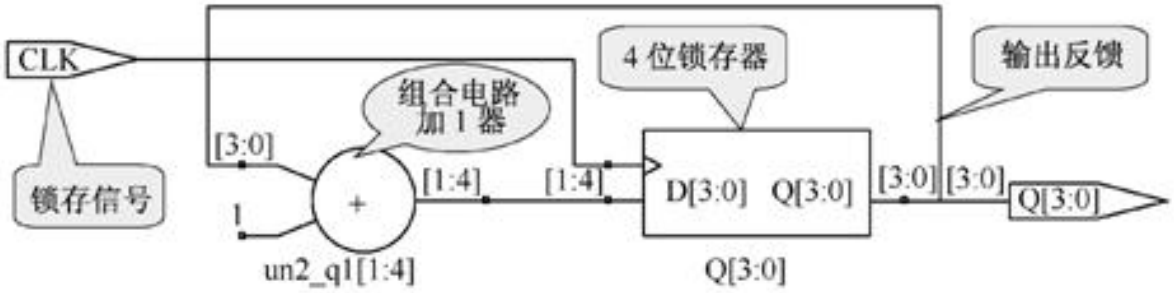


图 5-13　4 位加法计数器 RTL 电路

5.2 计数器的VHDL设计

5.2.2 计数器更常用的VHDL表达方式

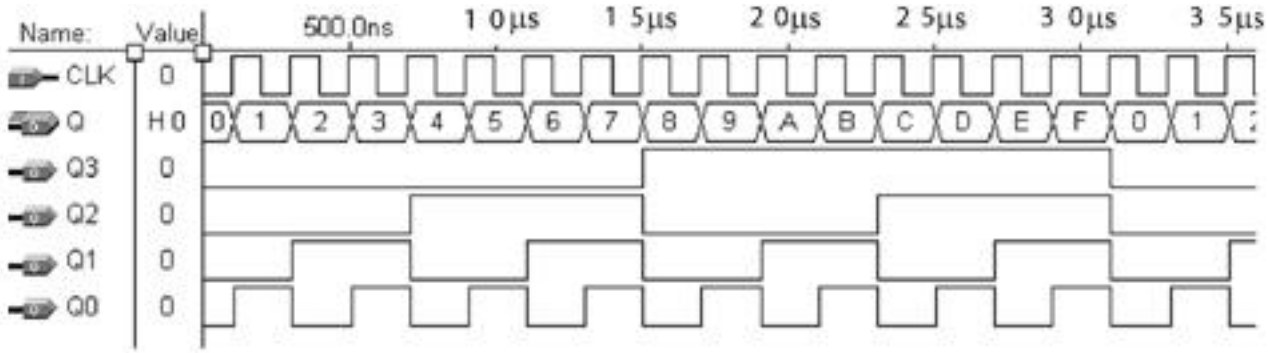


图 5-14　4 位加法计数器工作时序

5.2 计数器的VHDL设计

5.2.3 实用计数器的VHDL设计

【例 5-15】

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY CNT10 IS
     PORT (CLK,RST,EN,LOAD : IN STD_LOGIC;
           DATA : IN STD_LOGIC_VECTOR(3 DOWNTO 0);  --4位预置数
           DOUT : OUT STD_LOGIC_VECTOR(3 DOWNTO 0); --计数值输出
           COUT : OUT STD_LOGIC);                   --计数进位输出
END CNT10;
ARCHITECTURE behav OF CNT10 IS
BEGIN
  PROCESS(CLK, RST, EN, LOAD)
    VARIABLE  Q : STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
   IF RST='0' THEN   Q := (OTHERS=>'0'); --复位低电平时，计数寄存器清0
     ELSIF CLK'EVENT AND CLK='1' THEN    --测试时钟上升沿
       IF EN='1' THEN                    --计数使能高电平，允许计数
          IF (LOAD='0') THEN  Q := DATA; ELSE  --预置控制低电平，允许加载
             IF Q<9 THEN   Q := Q + 1;     --计数小于9，继续累加
                ELSE   Q := (OTHERS=>'0'); --否则计数清0
             END IF;
          END IF;
       END IF;
   END IF;
   IF Q="1001" THEN COUT<='1';             --当计数为9时，进位输出1
            ELSE    COUT<='0';  END IF;    --否则进位输出0
       DOUT <= Q;                          --计数寄存器的值输出端口
 END PROCESS;
END behav;
```

5.2 计数器的VHDL设计

1．程序分析

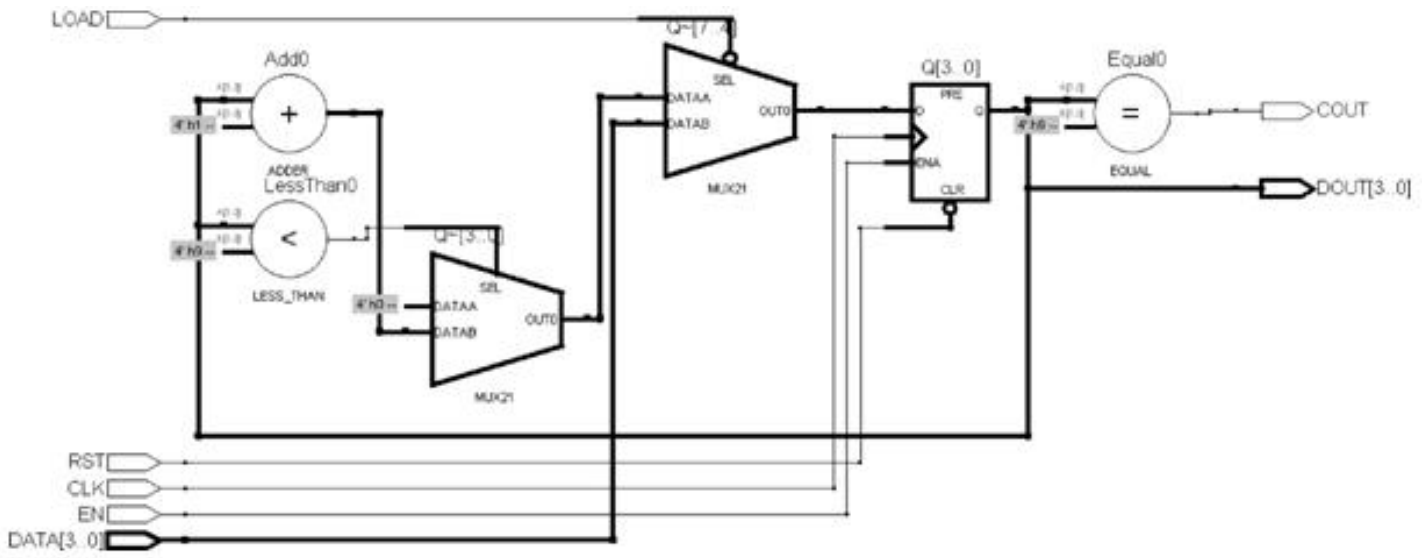


图 5-15　例 5-15 的 RTL 电路图

5.2 计数器的VHDL设计

1．程序分析

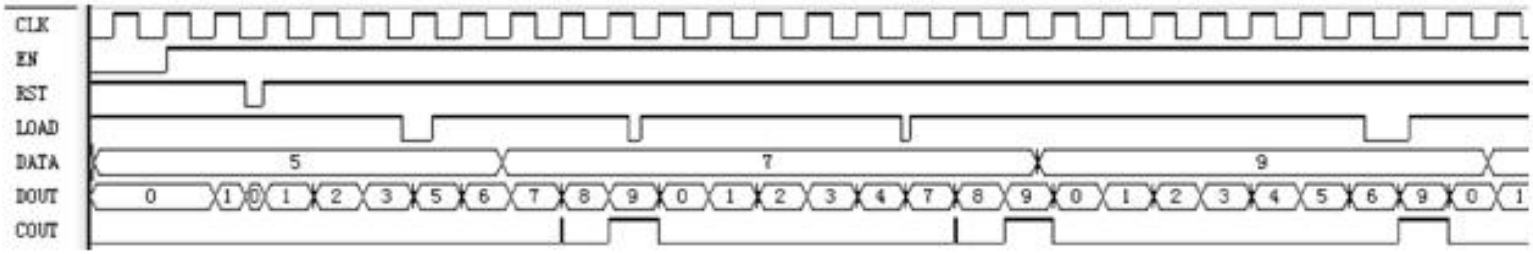


图 5-16　例 5-15 的时序仿真波形图

2．时序模块中的同步控制信号和异步控制信号的构建

5.2 计数器的VHDL设计

3．另一种描述方式

【例 5-16】

```
SIGNAL  Q : STD_LOGIC_VECTOR(3 DOWNTO 0);
...
   REG:  PROCESS(CLK, RST, EN,Q,LOAD)   BEGIN     --时序进程
       IF RST='0' THEN   Q <= (OTHERS=>'0') ;
         ELSIF CLK'EVENT AND CLK='1' THEN
           IF EN='1' THEN
             IF (LOAD='0') THEN  Q<=DATA; ELSE
               IF Q<9 THEN   Q<=Q+1;  ELSE   Q<=(OTHERS=>'0');  END IF;
END IF;     END IF;     END IF;
      END PROCESS;
         DOUT <= Q;
    COM:  PROCESS(Q)   BEGIN                          --组合进程
          IF Q="1001" THEN COUT<='1';  ELSE   COUT<='0';  END IF;
     END PROCESS;
```

5.3 移位寄存器的VHDL设计

```
REG8(6 DOWNTO 0)<= REG8(7 DOWNTO 1) ;
```

【例 5-17】

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY SHFT IS
   PORT ( CLK, LOAD : IN STD_LOGIC;  QB : OUT STD_LOGIC
               DIN  : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
               DOUT : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) );
END SHFT;
ARCHITECTURE behav OF SHFT IS
 SIGNAL REG8 : STD_LOGIC_VECTOR(7 DOWNTO 0);
BEGIN
   PROCESS (CLK, LOAD)       BEGIN
     IF CLK'EVENT AND CLK = '1' THEN
        IF LOAD = '1' THEN  REG8 <= DIN;  --由（LOAD='1'）装载新数据
          ELSE  REG8(6 DOWNTO 0) <= REG8(7 DOWNTO 1);
        END IF;
     END IF;
    END PROCESS;
       QB <= REG8(0);     DOUT<=REG8;
END behav;
```

5.3 移位寄存器的VHDL设计

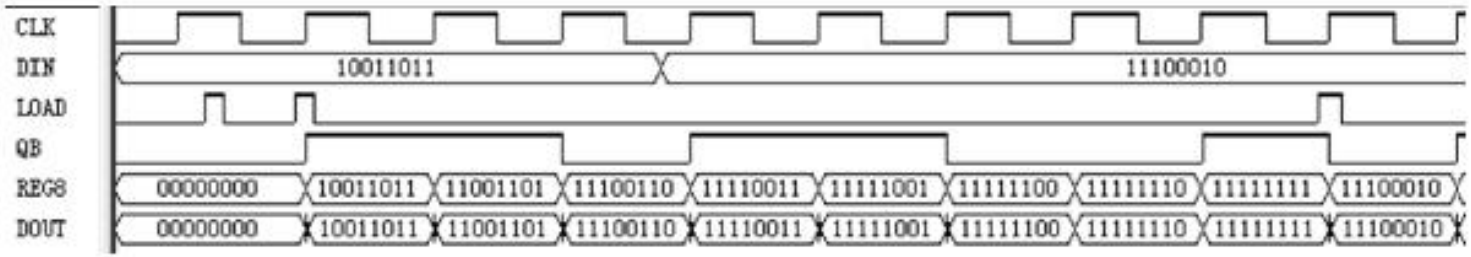


图 5-17　例 5-17 的工作时序

5.4 属性描述与定义语句

属性测试项目名'属性标识符

1. 信号类属性

```
NOT(clock'STABLE AND clock ='1')
(clock'EVENT AND clock ='1')
```

2. 数据区间类属性

```
SIGNAL range1 : IN STD_LOGIC_VECTOR (0 TO 7);
...
FOR i IN range1'RANGE LOOP
```

5.4 属性描述与定义语句

3．数值类属性

```
PROCESS (clock, a, b);
TYPE obj IS ARRAY (0 TO 15) OF BIT ; --定义obj为BIT数组类型
SIGNAL ele1, ele2, ele3, ele4    : INTEGER ;
BEGIN
 ele1 <= obj'RIGHT ; --测得数据类型obj的最右侧位是第15位
 ele2 <= obj'LEFT  ; --测得数据类型obj的最左侧位是第0位
 ele3 <= obj'HIGH  ; --测得数据类型obj的最高位是第15位
 ele4 <= obj'LOW   ; --测得数据类型obj的最低位是第0位
...
```

5.4 属性描述与定义语句

3．数值类属性

【例 5-18】

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY parity IS
   GENERIC (bus_size : INTEGER := 8 );
   PORT (input_bus : IN STD_LOGIC_VECTOR(bus_size-1 DOWNTO 0);
         even_numbits, odd_numbits : OUT STD_LOGIC ) ;
END parity ;
ARCHITECTURE behave OF parity IS
BEGIN
PROCESS (input_bus)
   VARIABLE temp: STD_LOGIC;
BEGIN
   temp := '0';
   FOR i IN input_bus'LOW TO input_bus'HIGH LOOP
   temp := temp XOR input_bus(i) ;
   END LOOP ;
   odd_numbits <= temp ;    even_numbits <= NOT temp;
END PROCESS;
END behave;
```

5.4 属性描述与定义语句

4. 数组属性'LENGTH

```
TYPE arry1 ARRAY (0 TO 7) OF BIT;
VARIABLE wth1 : INTEGER;
...
wth1: =arry1'LENGTH; -- wth1 = 8
```

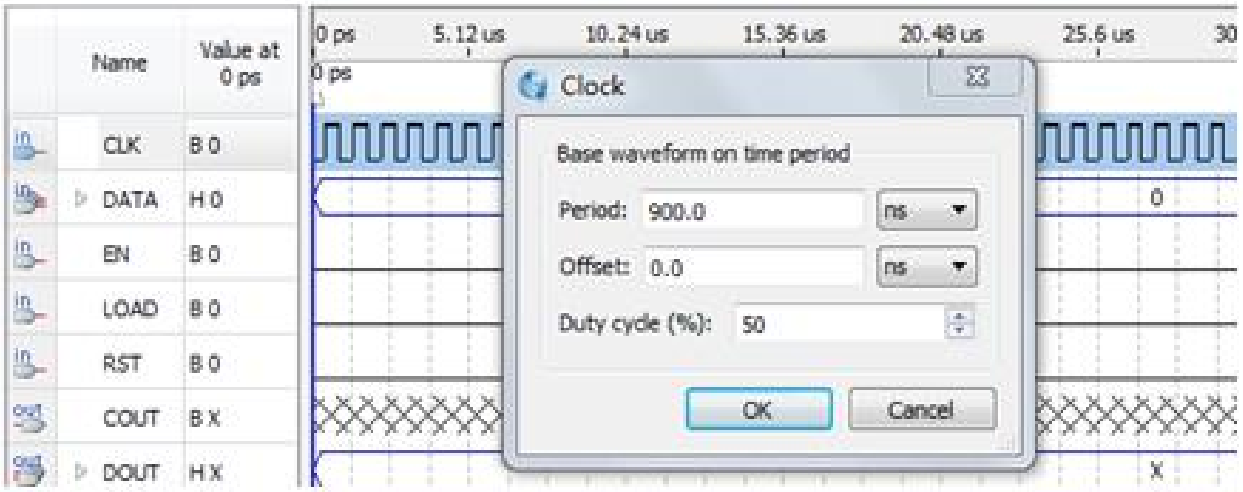
5. 用户定义属性

```
ATTRIBUTE 属性名 : 数据类型;
ATTRIBUTE 属性名 OF 对象名 : 对象类型IS 值;
```

```
LIBRARY synplify;
USE synplicity.attributes.all;
```

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



5-18 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 A.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)	合并	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

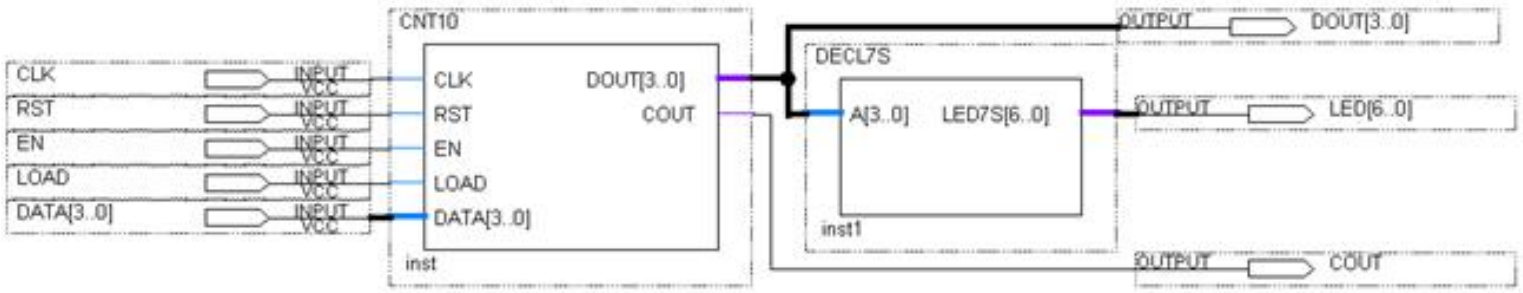


图 5-19　十进制计数器设计示例电路

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

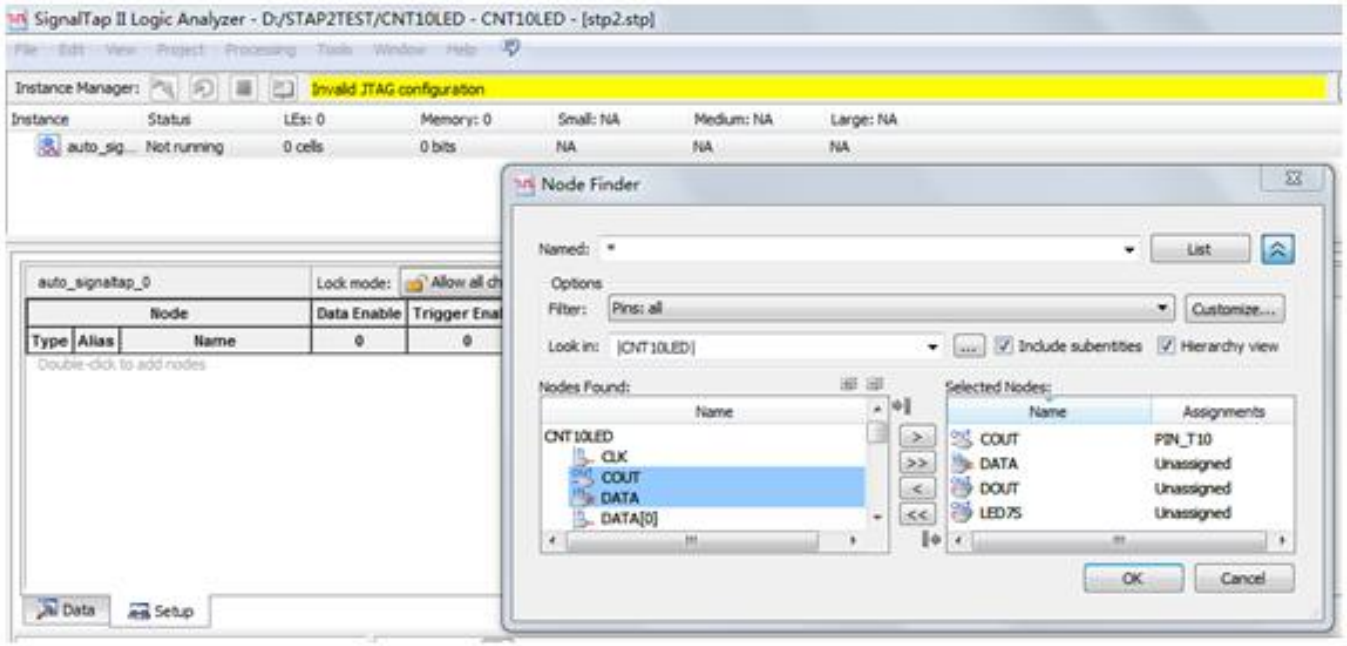


图 5-20 输入逻辑分析仪测试信号

5.6 SignalTap II的使用方法

2．调入待测信号

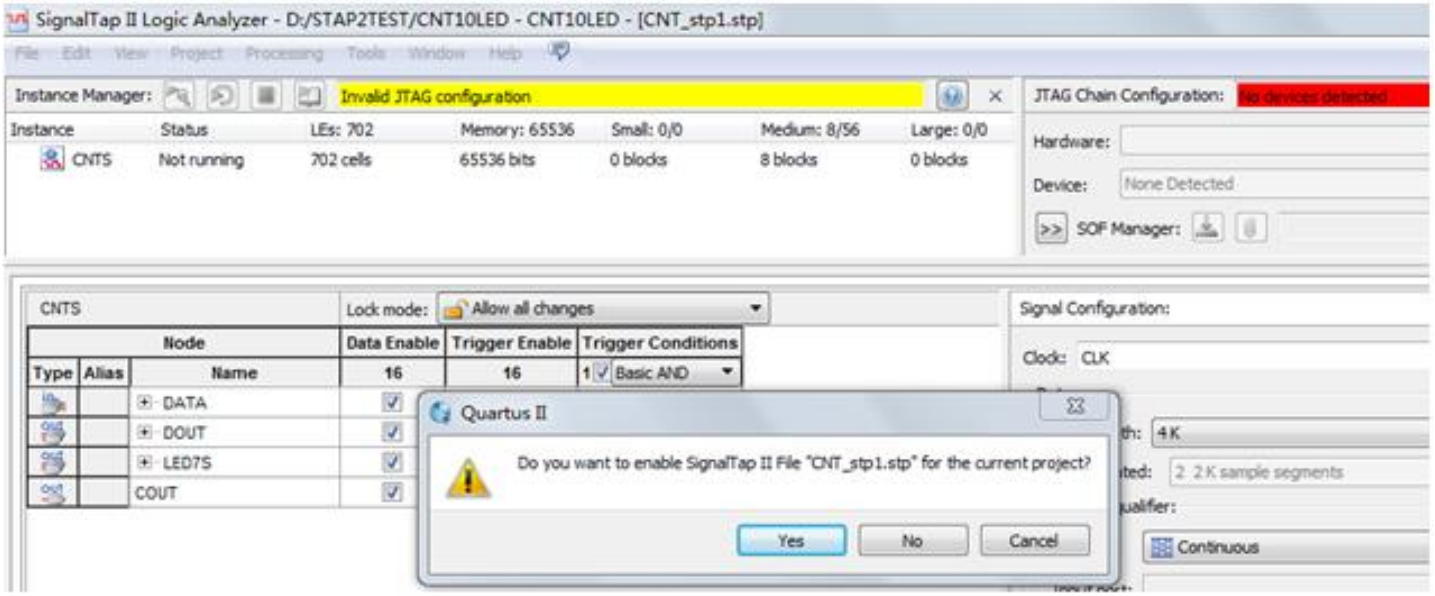


图 5-21　SignalTap II 编辑窗口

5.6 SignalTap II的使用方法

3. SignalTap II参数设置

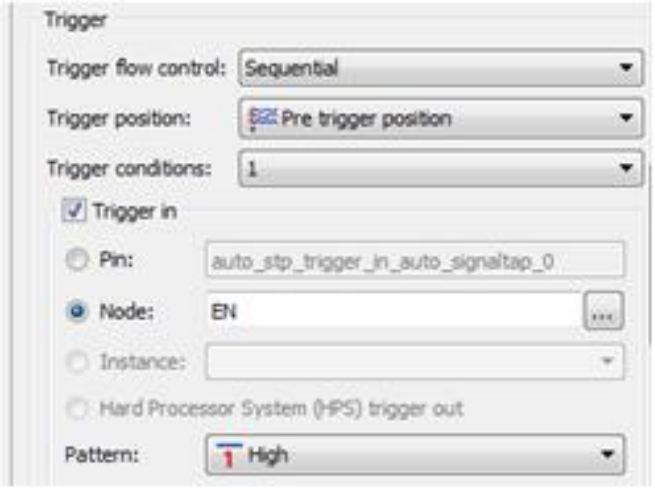


图 5-22 设置 EN 为触发信号

5.6 SignalTap II的使用方法

4．文件存盘

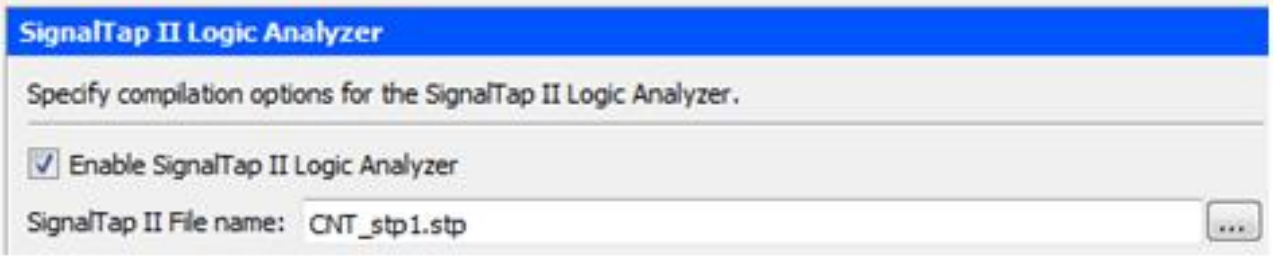


图 5-23 选择或删除 SignalTap II 文件加入综合编译

5.6 SignalTap II的使用方法

5. 编译下载

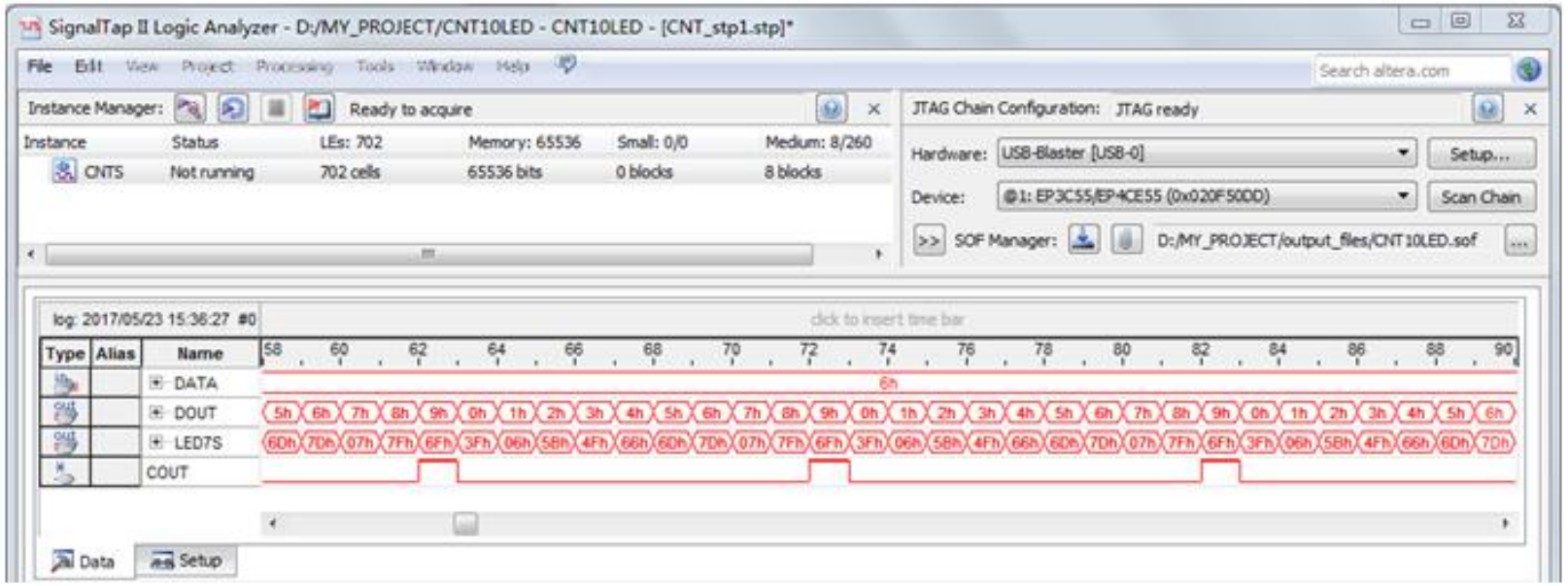


图 5-24 SignalTap II 实时数据采样显示界面

5.6 SignalTap II的使用方法

6. 启动SignalTap II进行采样与分析

```
ARCHITECTURE ONE OF xxx IS
attribute chip_pin of CLK0 : signal is "B11"; -- 逻辑分析仪采样时钟
```

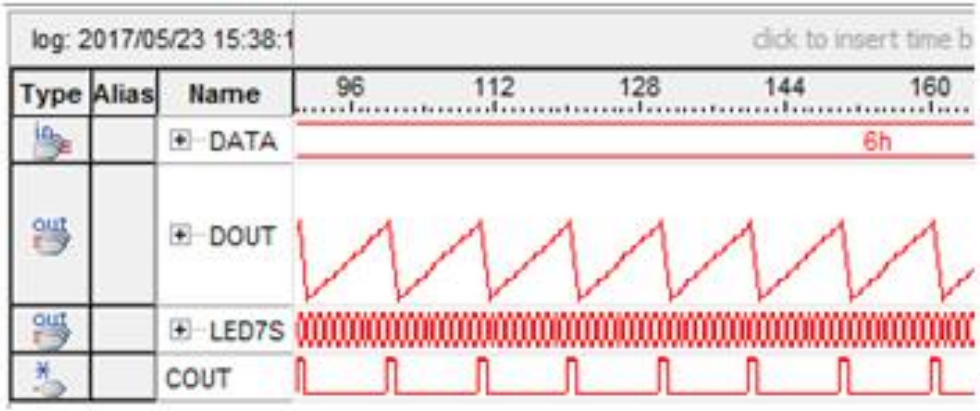


图 5-25　改变 DOUT 数据显示的方式

5.6 SignalTap II的使用方法

6．启动SignalTap II进行采样与分析

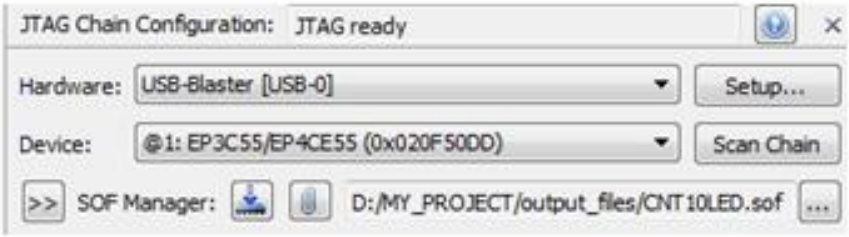


图 5-26　扫描 FPGA，并下载 SOF 文件

7．SignalTap II的其他设置和控制方法

5.7 编辑SignalTap II的触发信号

习　题

5-9 分别给出**RTL**图（图**5-27**和图**5-28**）的**VHDL**描述，注意其中的**D**触发器和锁存器的表述。

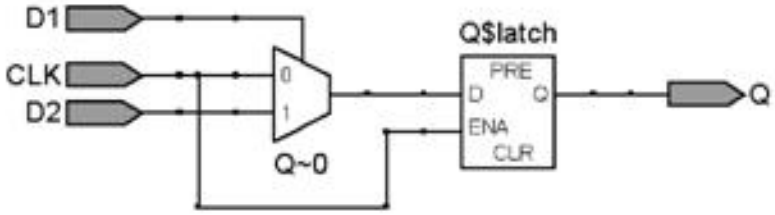


图 5-27　RTL 图 1

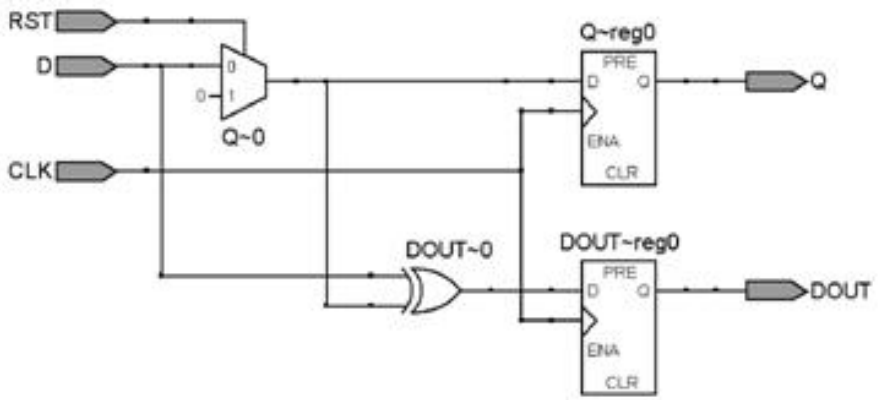


图 5-28　RTL 图 2

实验与设计

5-1 高速硬件除法器设计

【例 5-19】

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity DIV16 is
port(CLK : in std_logic;  A,B : in std_logic_vector(15 downto 0);
   QU,RE  : out std_logic_vector(15 downto 0));
end DIV16;
architecture rtl of DIV16 is
begin
  process(CLK)
    variable AT, BT,P,Q  : std_logic_vector(15 downto 0);
  begin
     if  rising_edge(CLK)  then   AT:=A;  BT:=B;
        P:="0000000000000000";   Q:="0000000000000000";
         for i in  QU'range  loop
            p := P(14 downto 0) & AT(15);
            AT:=AT(14 downto 0)&'0';  P:=P-BT;
               if P(15)='1' then Q(i):='0'; P:=P+BT ;
                  else Q(i):='1';  end if;
         end loop;   end if ;
               QU <= Q;  RE <= P ;
   end process;
end rtl ;
```

实验与设计

5-2 移位相加型8位硬件乘法器设计

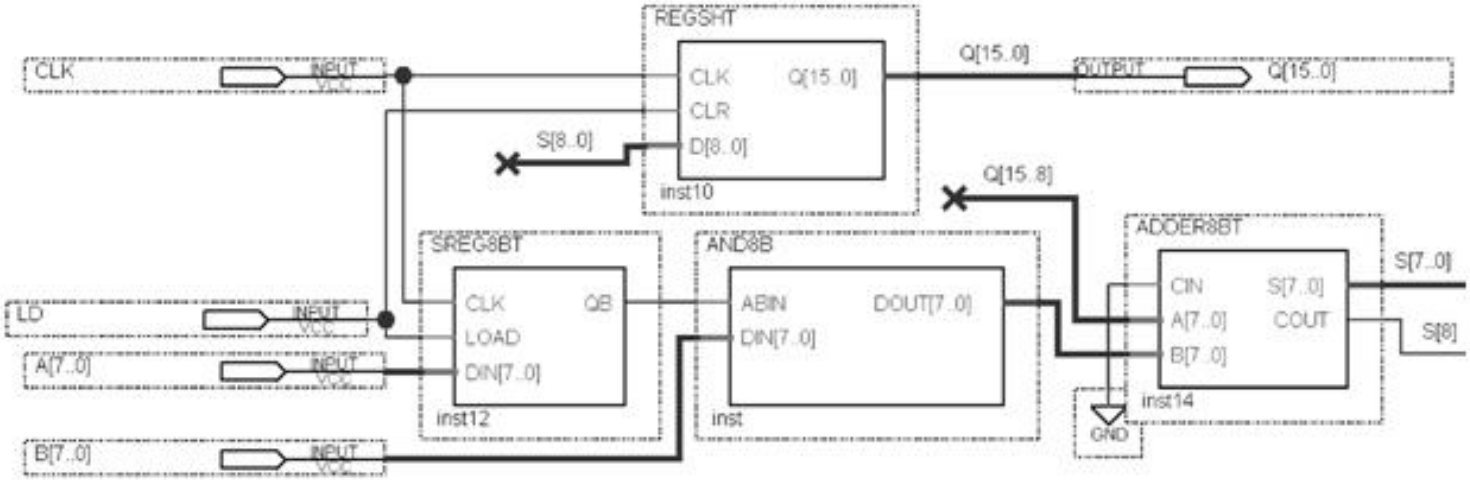


图 5-29　8 位乘法器逻辑原理图

实验与设计

5-3 半整数与奇数分频器设计

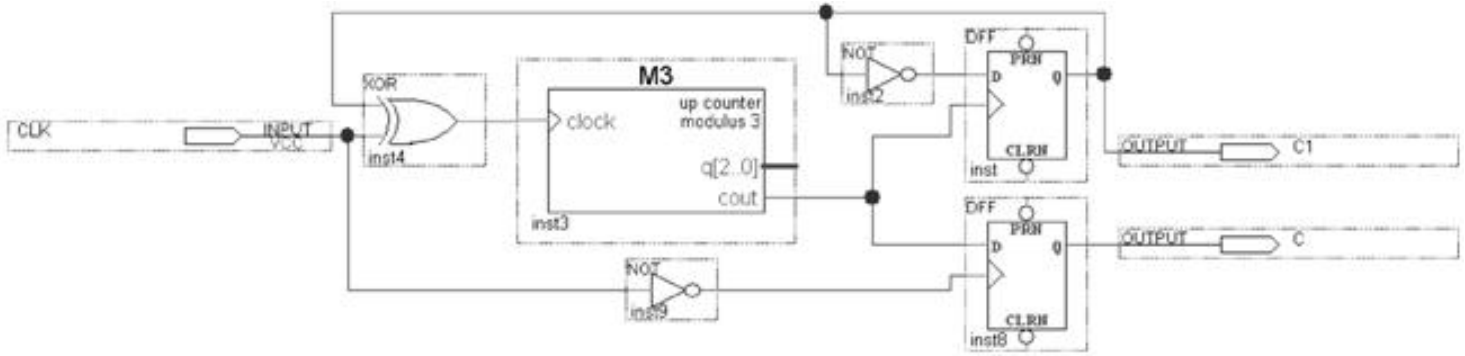


图 5-30 占空比为 50%的任意奇数次分频电路

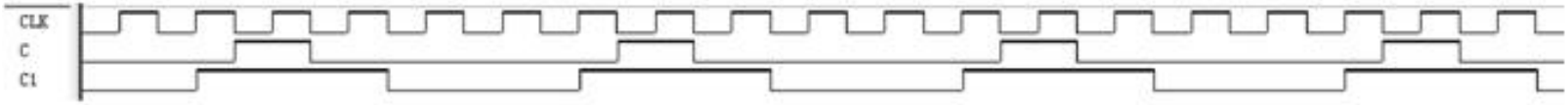


图 5-31 图 5-30 电路的仿真波形

实验与设计

5-3 半整数与奇数分频器设计

【例 5-20】占空比为 50%的任意奇数次 5 分频电路

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
USE IEEE.STD_LOGIC_UNSIGNED.ALL ;
ENTITY DIV IS
    PORT (CLK : IN STD_LOGIC ;  K_OR,K1,K2 : OUT STD_LOGIC) ;
END ;
ARCHITECTURE bhv OF DIV IS
   SIGNAL C1,C2 : STD_LOGIC_VECTOR(2 DOWNTO 0) ;
   SIGNAL M1,M2 : STD_LOGIC ;
 BEGIN
PROCESS(CLK,C1)    BEGIN
  IF  RISING_EDGE(CLK) THEN
  IF (C1="100") THEN  C1<="000";  ELSE C1<=C1+1; END IF;
  IF (C1="001") THEN  M1<=NOT M1;  ELSIF (C1="011") THEN  M1<=NOT M1;
  END IF;    END IF;
END PROCESS;
PROCESS(CLK,C2)   BEGIN
  IF  FALLING_EDGE(CLK) THEN
  IF (C2="100") THEN  C2<="000";  ELSE C2<=C2+1; END IF;
  IF (C2="001") THEN  M2<=NOT M2;  ELSIF (C2="011") THEN  M2<=NOT M2;
  END IF;  END IF;
END PROCESS;
  K1 <= M1;  K2 <= M2; K_OR <= M1 OR M2;
END bhv;
```

实验与设计

5-3 半整数与奇数分频器设计

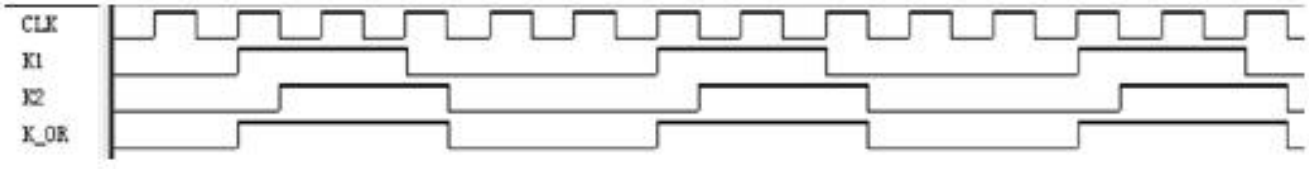


图 5-32　占空比为 50%的任意奇数次分频电路

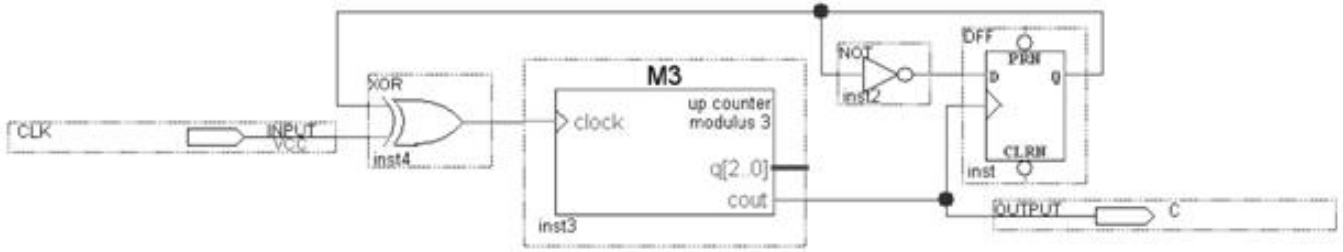


图 5-33 任意半整数分频电路

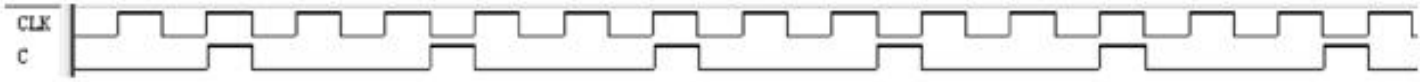


图 5-34 图 5-33 电路仿真波形图

实验与设计

5-6 串行静态显示控制电路设计

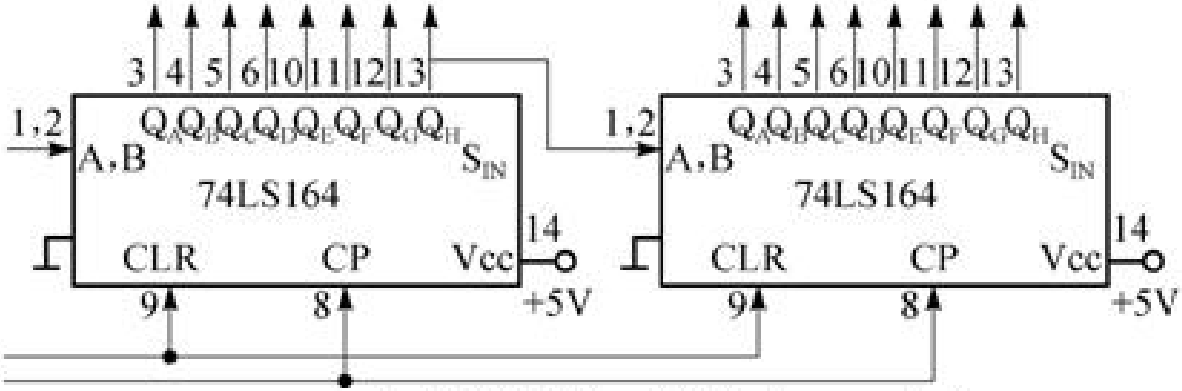


图 5-35 串/并转换数码管静态显示电路

实验与设计

5-7 VGA彩条信号显示控制电路设计

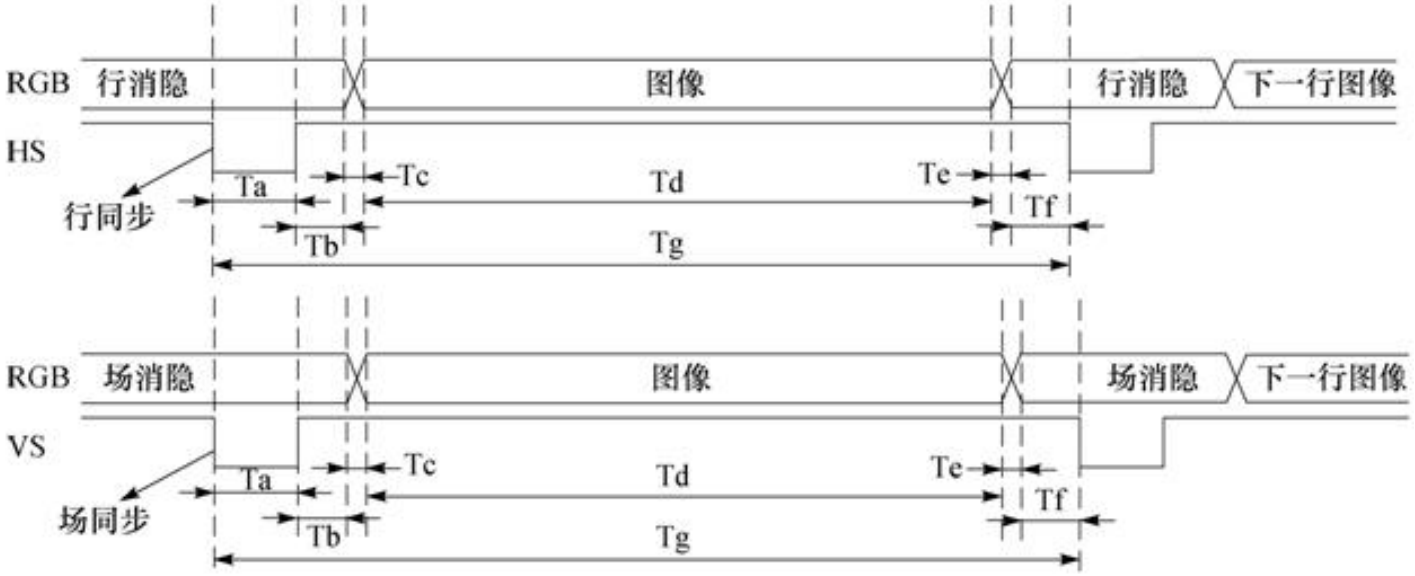


图 5-36　VGA 行扫描、场扫描时序示意图

实验与设计

5-7 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）

		行同步头			行图像		行周期
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

5-7 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

5-7 VGA彩条信号显示控制电路设计

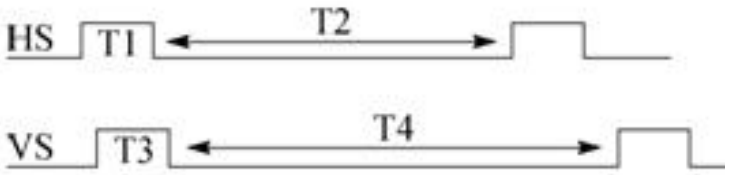


图 5-37 HS 和 VS 的时序图

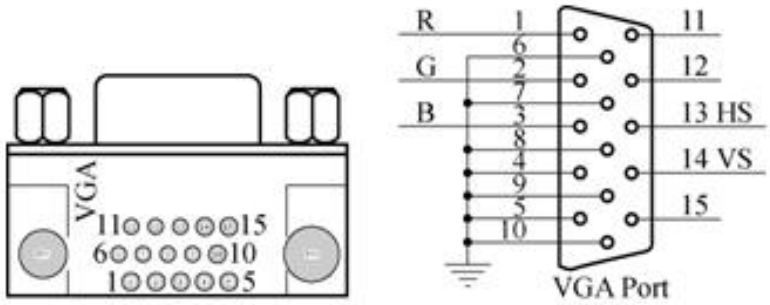


图 5-38 VGA 接口电路图，左接口从上往下看

【例 5-21】

```
LIBRARY IEEE;   -- VGA显示器彩条发生器
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY COLOR IS
   PORT (     CLK, MD : IN STD_LOGIC;
      HS, VS, R, G, B : OUT STD_LOGIC  ); -- 行场同步及红，绿，蓝控制
END COLOR;
ARCHITECTURE behav OF COLOR IS
   SIGNAL HS1,VS1,FCLK,CCLK     : STD_LOGIC;
   SIGNAL MMD : STD_LOGIC_VECTOR(1 DOWNTO 0);-- 方式选择
   SIGNAL PS : STD_LOGIC_VECTOR (3 DOWNTO 0);
   SIGNAL CC : STD_LOGIC_VECTOR(4 DOWNTO 0); -- 行同步/横彩条生成
   SIGNAL LL : STD_LOGIC_VECTOR(8 DOWNTO 0); -- 场同步/竖彩条生成
   SIGNAL GRBX : STD_LOGIC_VECTOR(3 DOWNTO 1);-- X横彩条
   SIGNAL GRBY : STD_LOGIC_VECTOR(3 DOWNTO 1);-- Y竖彩条
   SIGNAL GRBP : STD_LOGIC_VECTOR(3 DOWNTO 1);
   SIGNAL GRB  : STD_LOGIC_VECTOR(3 DOWNTO 1);
BEGIN
   GRB(2) <= (GRBP(2) XOR MD) AND HS1 AND VS1;
   GRB(3) <= (GRBP(3) XOR MD) AND HS1 AND VS1;
   GRB(1) <= (GRBP(1) XOR MD) AND HS1 AND VS1;
   PROCESS( MD )    BEGIN
      IF MD'EVENT AND MD = '0' THEN
         IF MMD = "10" THEN  MMD <= "00";
          ELSE   MMD <= MMD + 1; END IF;       -- 3种模式
      END IF;
   END PROCESS;
   PROCESS( MMD )    BEGIN
      IF MMD = "00" THEN     GRBP <= GRBX;     -- 选择横彩条
      ELSIF MMD = "01" THEN  GRBP <= GRBY;     -- 选择竖彩条
      ELSIF MMD = "10" THEN  GRBP <= GRBX XOR GRBY; -- 产生棋盘格
       ELSE  GRBP <= "000";  END IF;
   END PROCESS;
   PROCESS( CLK )    BEGIN
      IF CLK'EVENT AND CLK = '1' THEN -- 13MHz 13分频
         IF PS = 13 THEN PS <= "0000";  ELSE PS <= (PS + 1); END IF;
      END IF;
```

```
    END PROCESS;
    FCLK <= FS(3); CCLK <= CC(4);
    PROCESS( FCLK )    BEGIN
       IF FCLK'EVENT AND FCLK = '1' THEN
          IF CC = 29 THEN  CC <= "00000";  ELSE  CC<=CC+1;   END IF;
       END IF;
    END PROCESS;
    PROCESS( CCLK )    BEGIN
       IF CCLK'EVENT AND CCLK = '0' THEN
          IF LL=481 THEN  LL<="0000000000";  ELSE  LL<=LL+1; END IF;
       END IF;
    END PROCESS;
    PROCESS( CC,LL )    BEGIN
       IF CC > 23 THEN  HS1 <= '0';  --行同步
       ELSE HS1 <= '1';  END IF;
       IF LL > 479 THEN  VS1 <= '0'; --场同步
       ELSE  VS1 <= '1';   END IF;
    END PROCESS;
    PROCESS(CC, LL)      BEGIN
          IF CC < 3  THEN GRBX <= "111";  -- 横彩条
       ELSIF CC < 6  THEN GRBX <= "110";
       ELSIF CC < 9  THEN GRBX <= "101";
       ELSIF CC < 13 THEN GRBX <= "100";
       ELSIF CC < 15 THEN GRBX <= "011";
       ELSIF CC < 18 THEN GRBX <= "010";
       ELSIF CC < 21 THEN GRBX <= "001";
        ELSE GRBX <= "000";         END IF;
          IF LL <  60 THEN GRBY <= "111";  -- 竖彩条
       ELSIF LL < 130 THEN GRBY <= "110";
       ELSIF LL < 180 THEN GRBY <= "101";
       ELSIF LL < 240 THEN GRBY <= "100";
       ELSIF LL < 300 THEN GRBY <= "011";
       ELSIF LL < 360 THEN GRBY <= "010";
       ELSIF LL < 420 THEN GRBY <= "001";
       ELSE GRBY <= "000";         END IF;
    END PROCESS;
    HS<=HS1 ; VS<=VS1 ;R<=GRB(2) ;G<=GRB(3) ; B<=GRB(1);
END behav;
```

实验与设计

5-8 基于VHDL代码的频率计设计

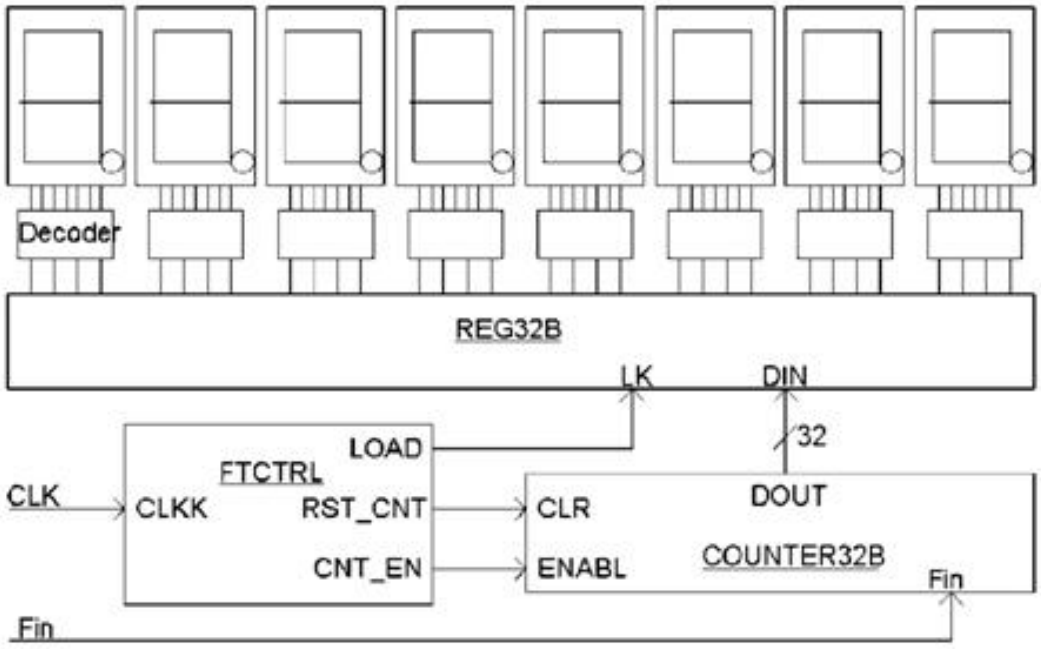


图 5-39 频率计电路框图

实验与设计

5-8 基于VHDL代码的频率计设计

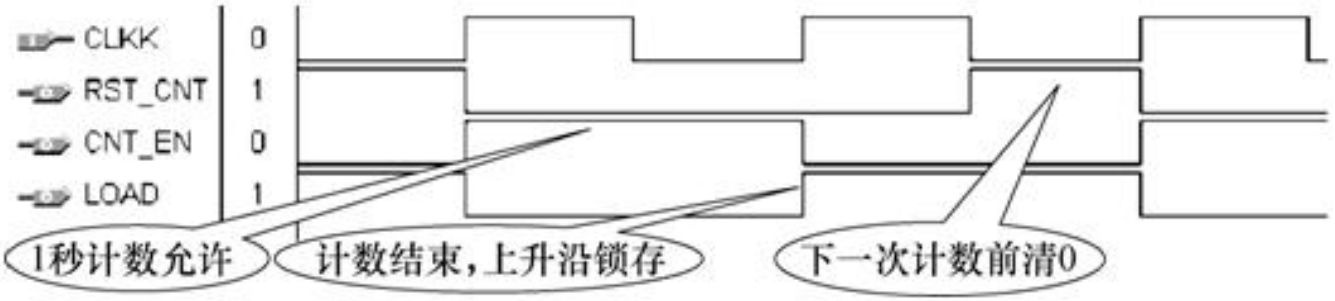


图 5-40 频率计测频控制器 FTCTRL 测控时序图