



第6章

Quartus II应用深入

6.1 时序电路硬件设计与仿真 示例

6.1.1 编辑电路、创建工程和仿真测试

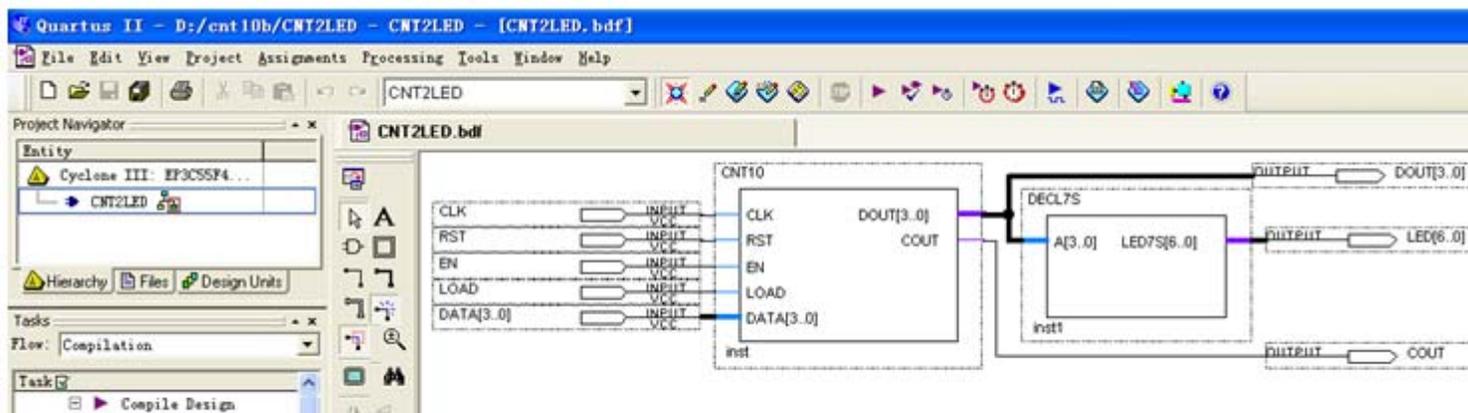


图 6-1 十进制计数器设计示例电路

6.1 时序电路硬件设计与仿真

示例

6.1.1 编辑电路、创建工程和仿真测试

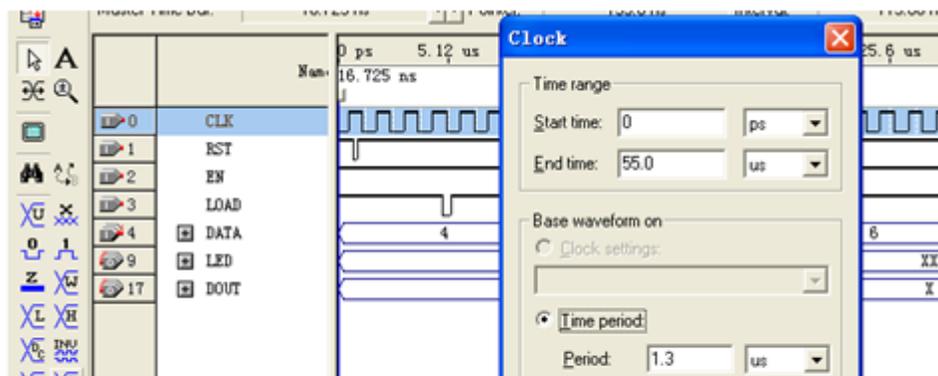


图 6-2 设置时钟 CLK 的周期

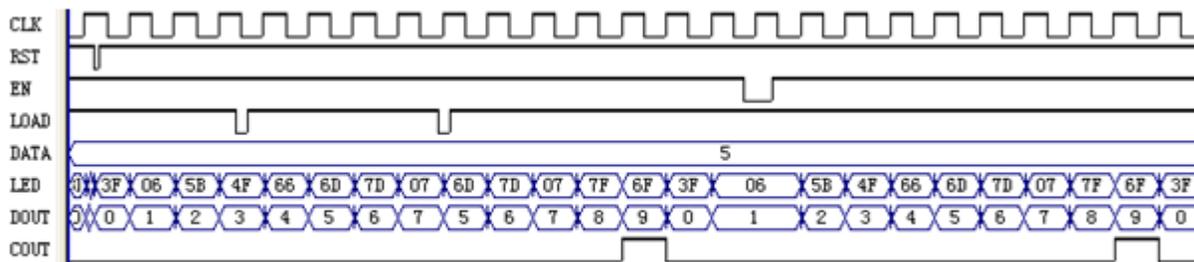


图 6-3 仿真波形输出报告 Simulation Report

6.1 时序电路硬件设计与仿真 示例

6.1.2 FPGA硬件测试

表 6-1 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	

6.2 SignalTap II 的使用方法

1. 打开SignalTap II编辑窗口
2. 调入待测信号

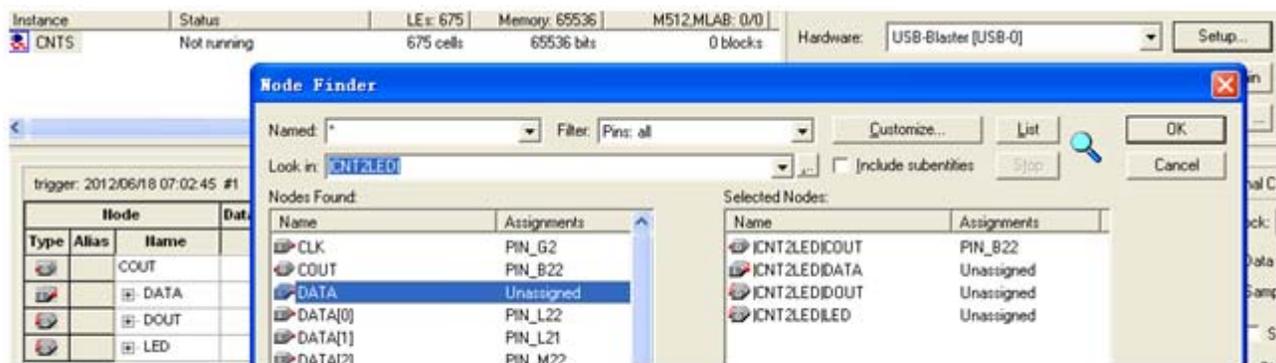


图 6-4 输入逻辑分析仪测试信号

6.2 SignalTap II 的使用方法

3. SignalTap II 参数设置

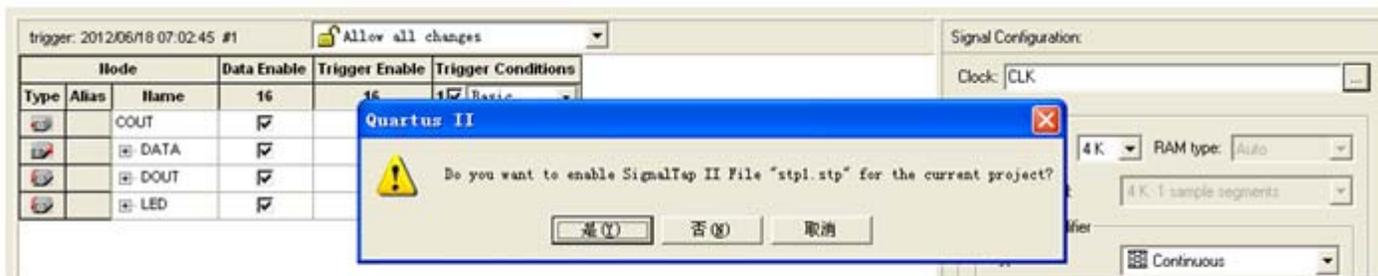


图 6-5 SignalTap II 编辑窗口



图 6-6 设置 EN 为触发信号

6.2 SignalTap II 的使用方法

4. 文件存盘

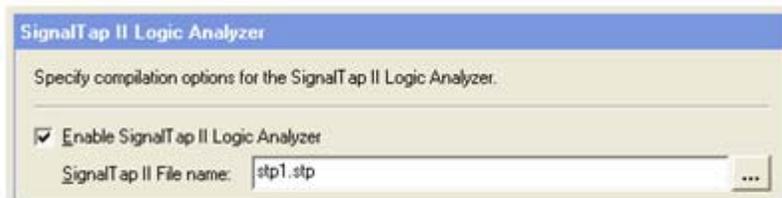


图 6-7 选择或删除 SignalTap II 文件加入综合编译

5. 编译下载

6.2 SignalTap II 的使用方法

6. 启动SignalTap II进行采样与分析

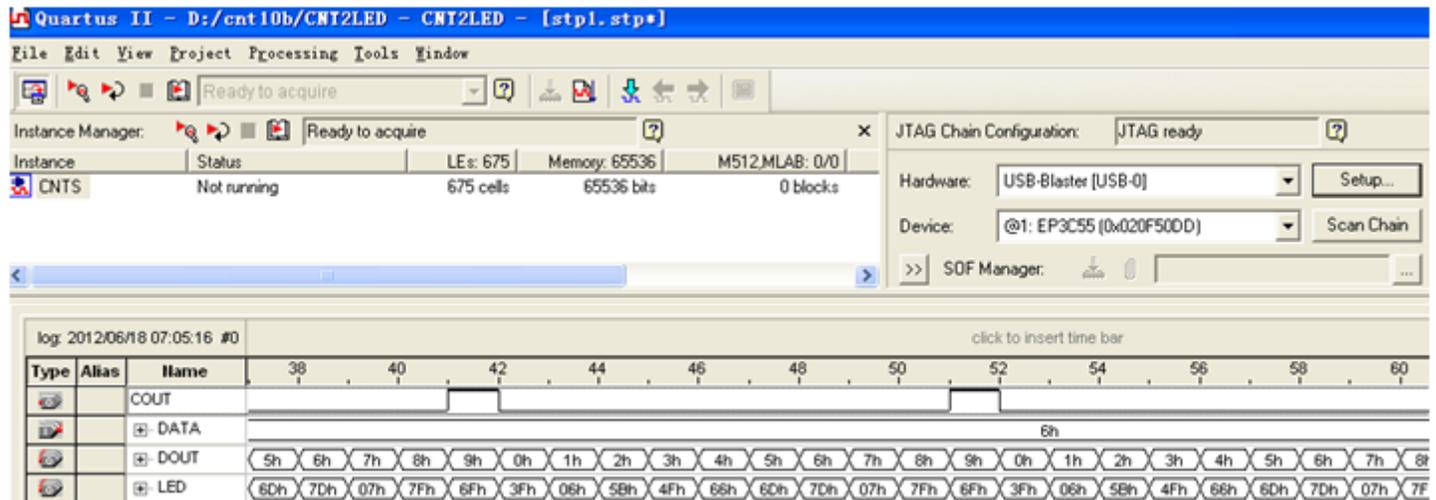


图 6-8 SignalTap II 实时数据采集显示界面

6.2 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 "G21"; -- 逻辑分析仪采样时钟

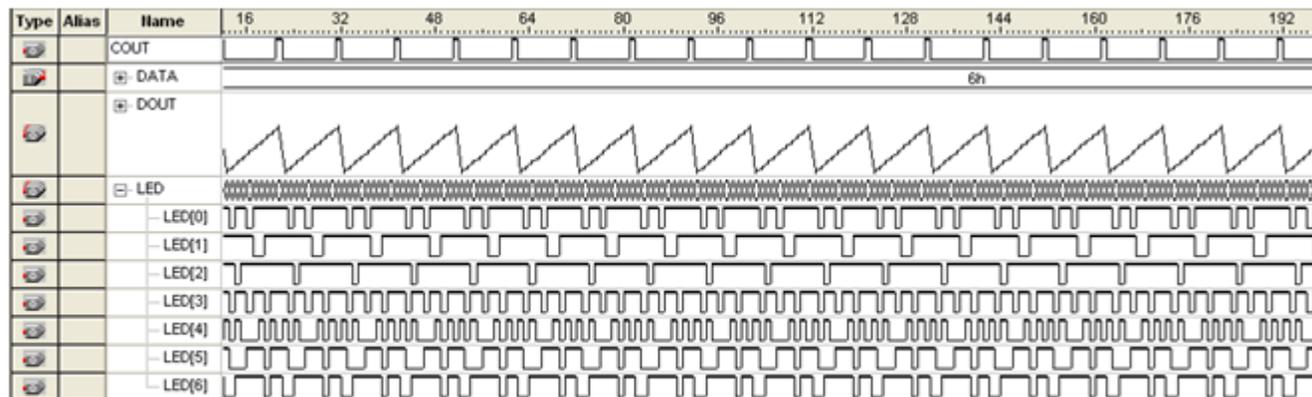


图 6-9 SignalTap II 数据窗口显示对硬件系统实时测试采样后的信号波形

7. SignalTap II 的其他设置和控制方法

6.3 编辑SignalTap II的触发信号

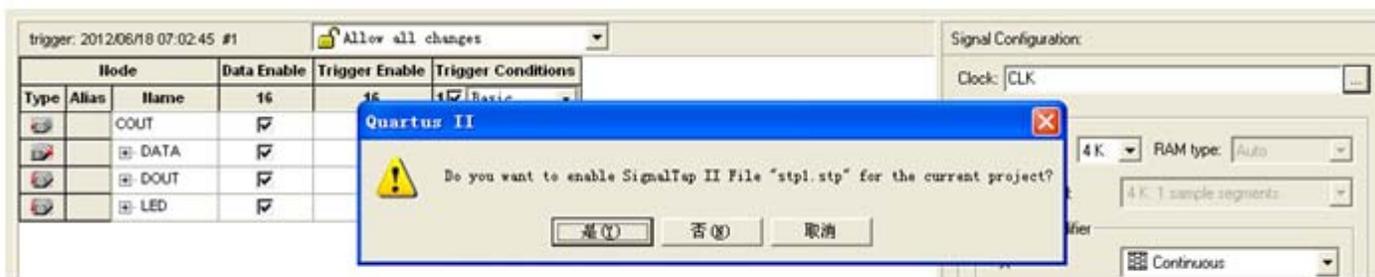


图 6-5 SignalTap II 编辑窗口

6.4 Fitter Settings项设置

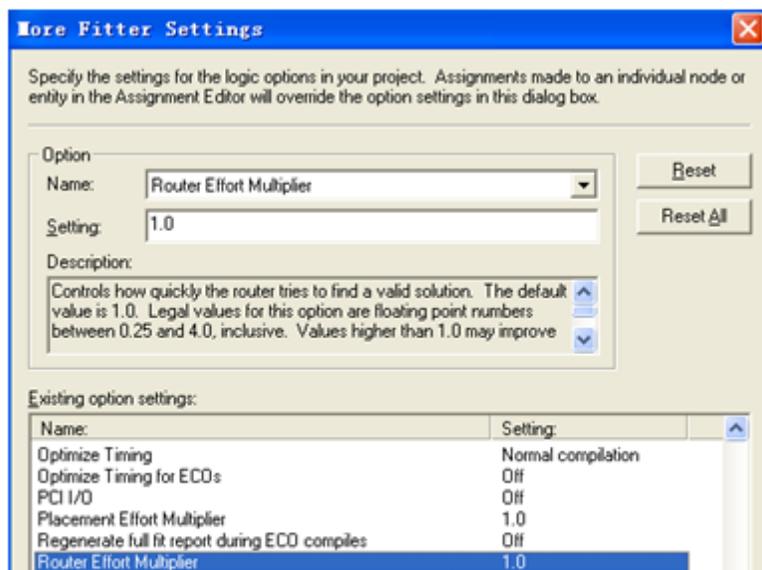


图6-10 布线倍增器优化程度指数选择

6.5 功能块Chip Planner应用

6.5.1 Chip Planner应用流程说明

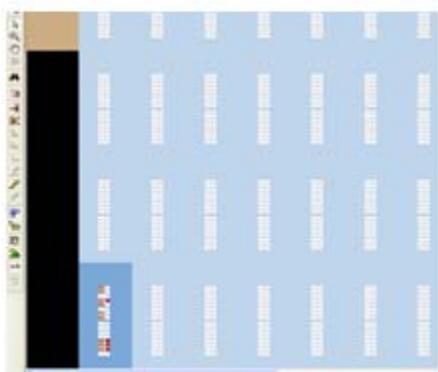


图 6-11 左下侧是已占用的 LAB

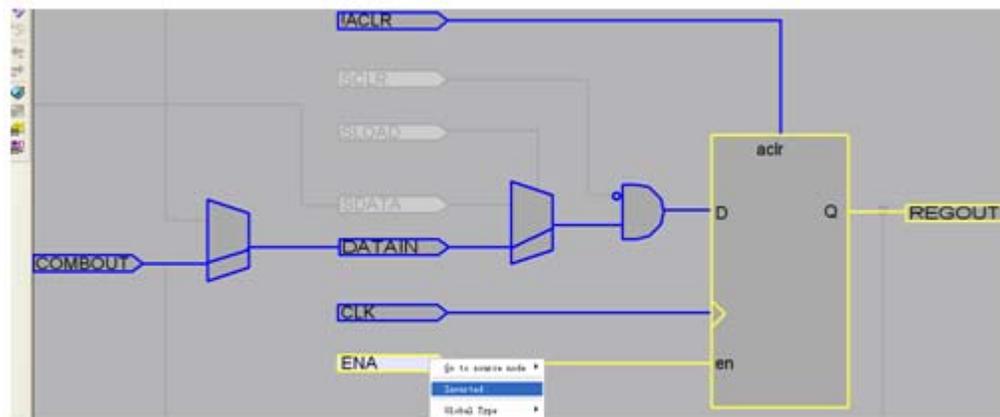


图 6-12 Resource Property Editor 的门级原理图编辑窗口

6.5 功能块Chip Planner应用

6.5.2 Chip Planner说明

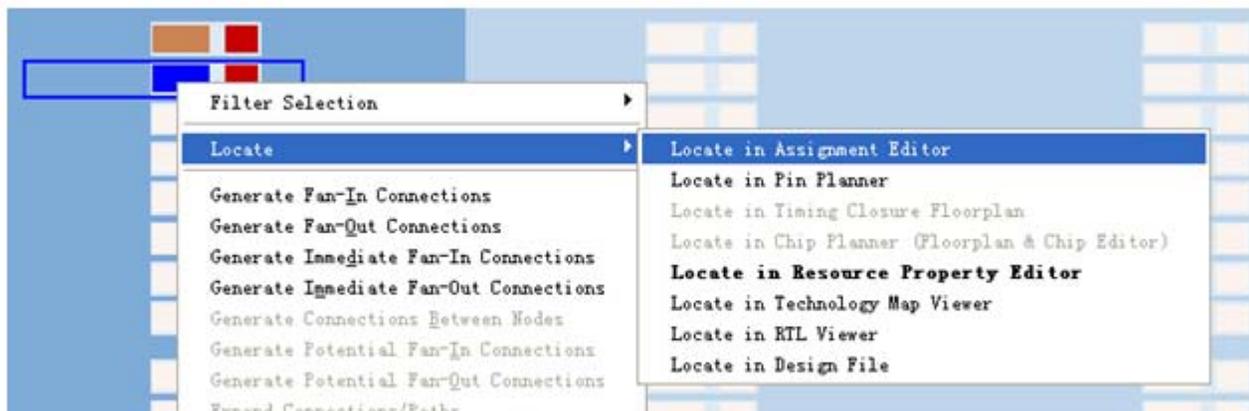


图 6-13 Locate 多项选择

6.5 功能块Chip Planner应用

6.5.2 Chip Planner说明

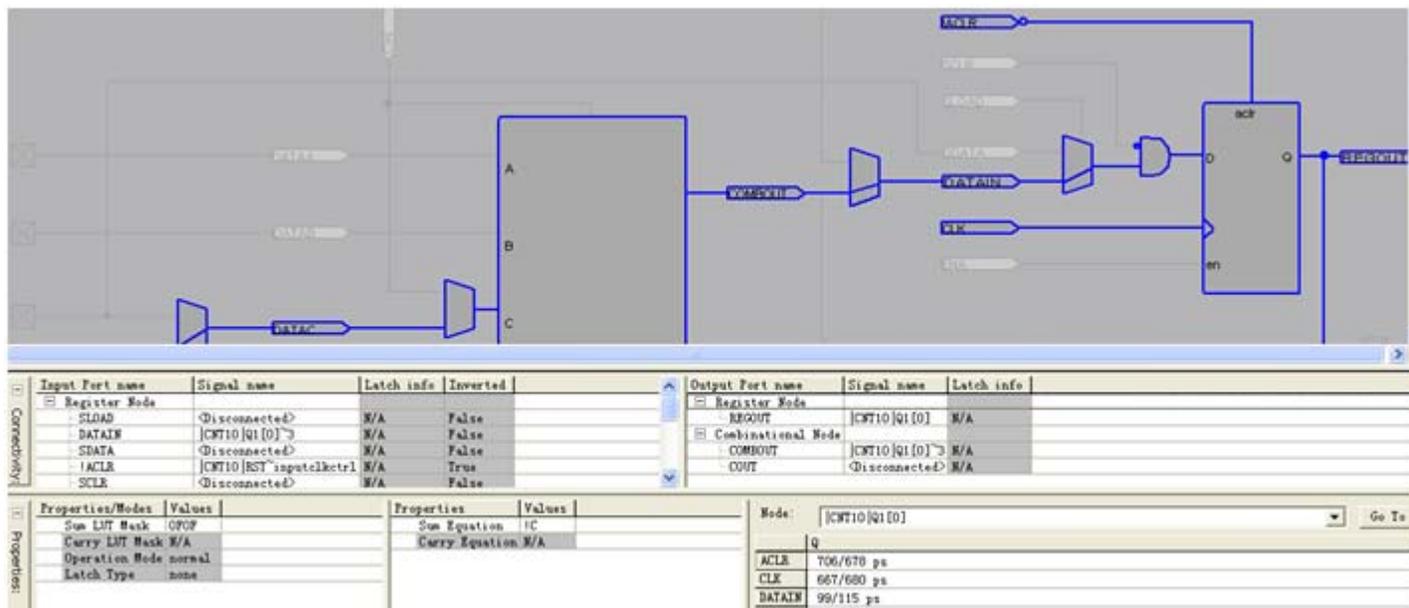


图 6-14 打开属性和端口连接窗口

6.6.1 Synplify使用流程

1. 启动Synplify

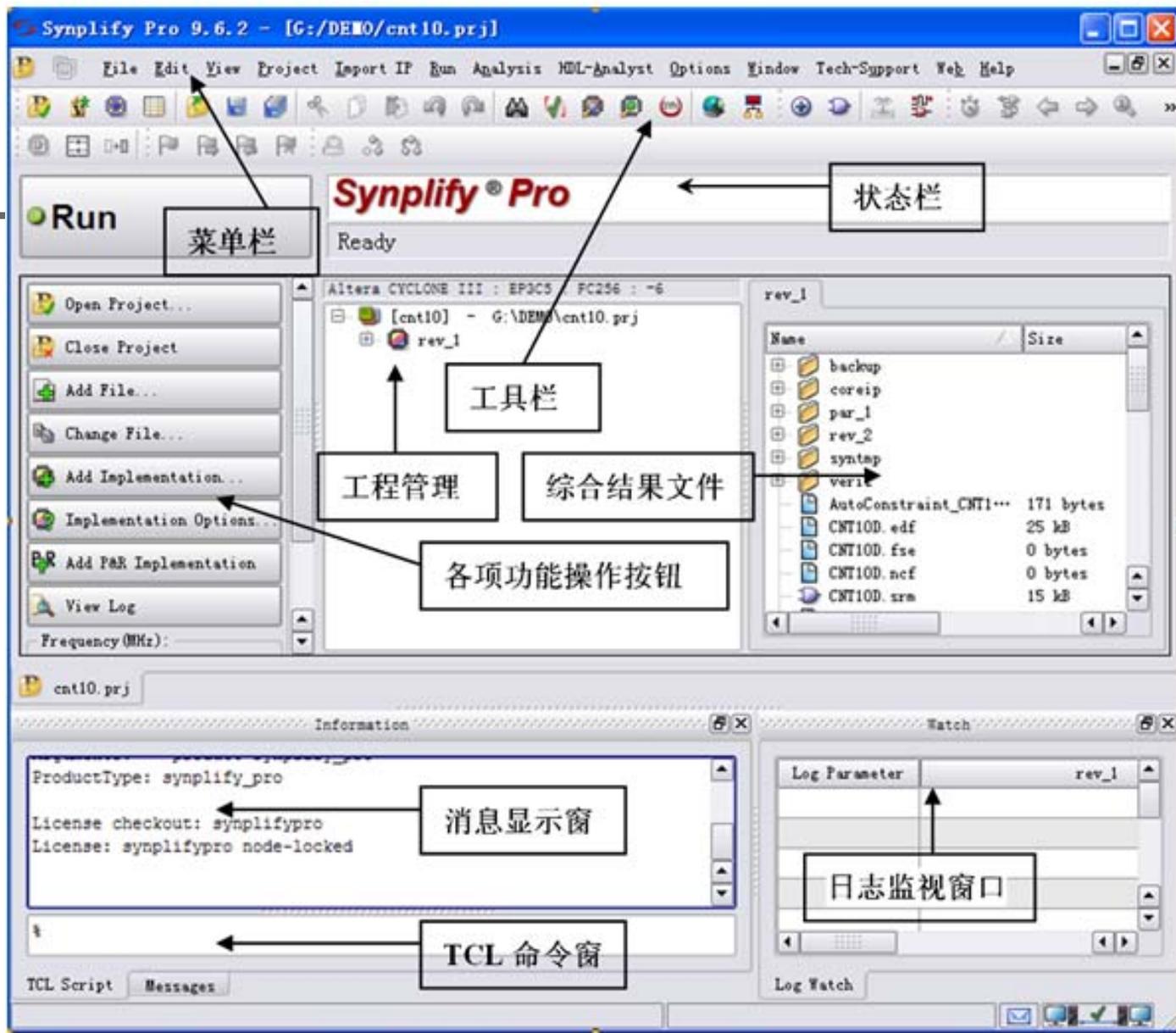


图6-15 Synplify Pro启动后界面

6.6 Synplify的应用及接口方法

6.6.1 Synplify使用流程

2. 创建工程

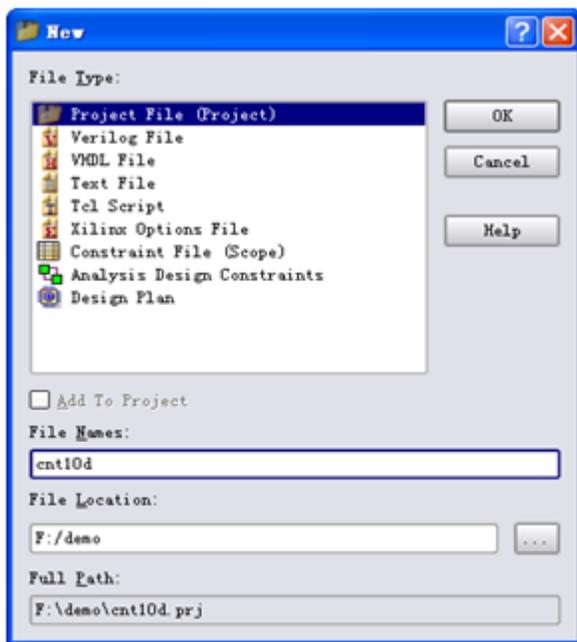


图6-16 创建工程

6.6 Synplify的应用及接口方法

6.6.1 Synplify使用流程

3. 加入源文件
4. 选择顶层文件
5. 设置工程属性

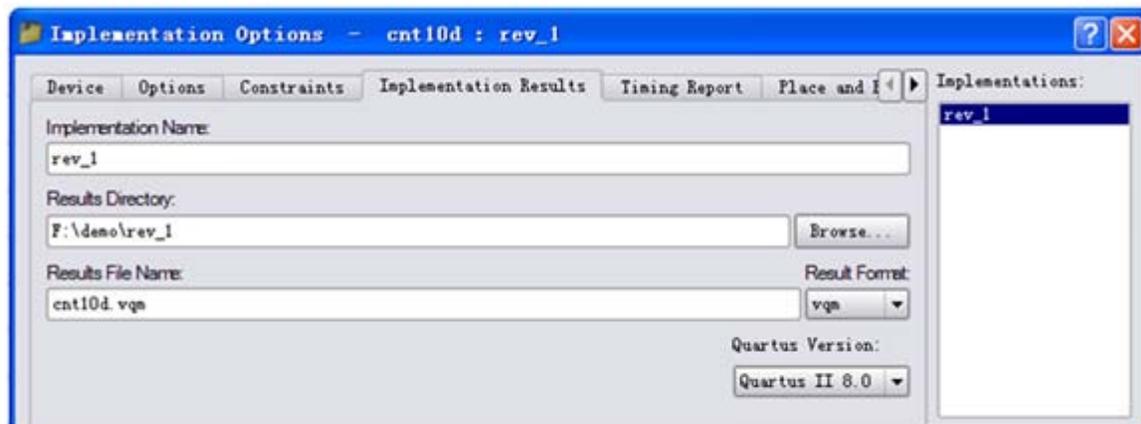
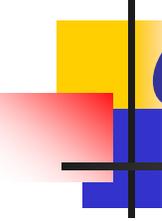


图6-17 综合目标详细信息



6.6 Synplify的应用及接口方法

6.6.1 Synplify使用流程

6. 综合前设置约束

7. 综合

8. 检测结果

6.6 Synplify的应用及接口方法

6.6.2 Synplify与Quartus II接口

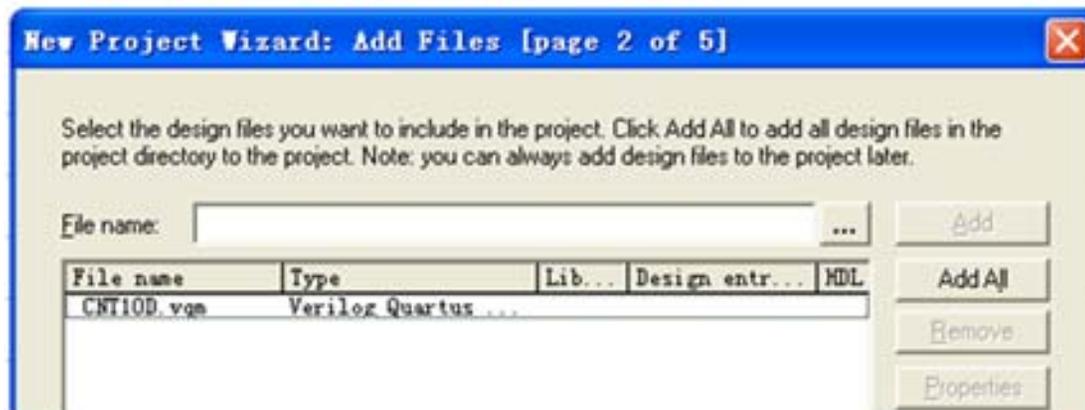


图 6-18 加入 Cnt10d.vqm 文件

6.6 Synplify的应用及接口方法

6.6.2 Synplify与Quartus II接口

1. Synplify软件路径设置

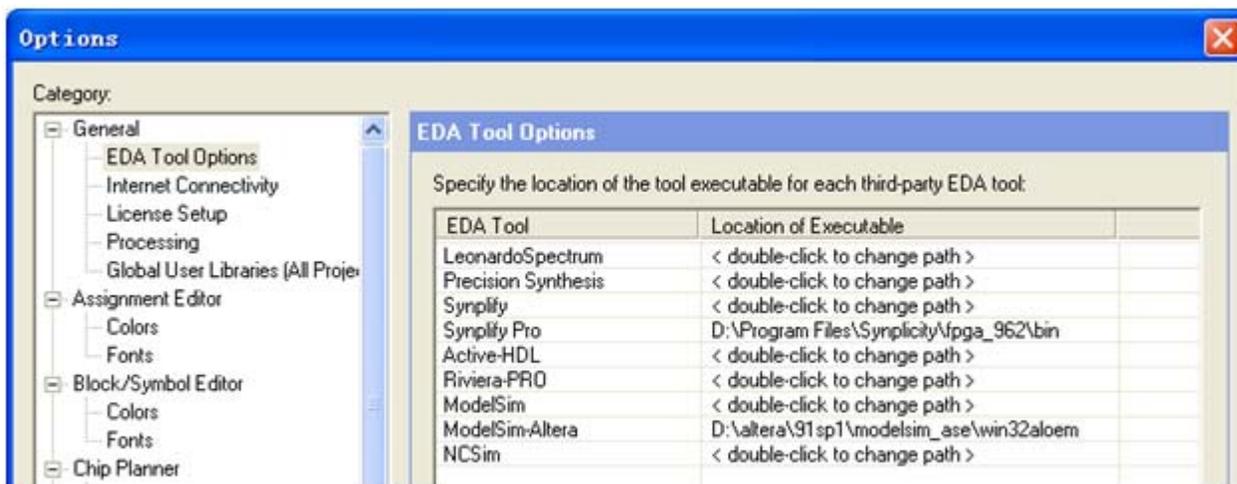


图6-19 Synplify软件路径设置

6.6 Synplify的应用及接口方法

6.6.2 Synplify与Quartus II接口

2. 设置Synplify Pro综合器

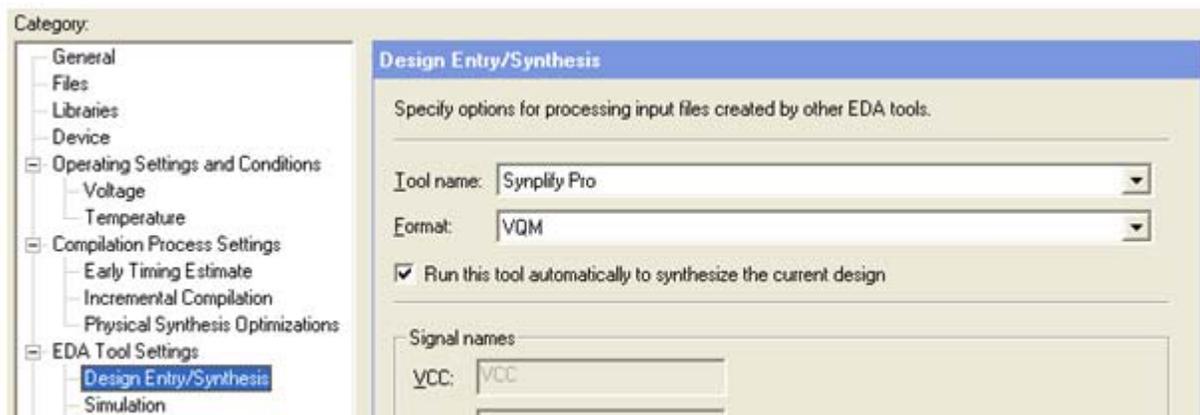


图6-20 设置综合器

实验与设计

6-1. 应用宏模块设计数字频率计

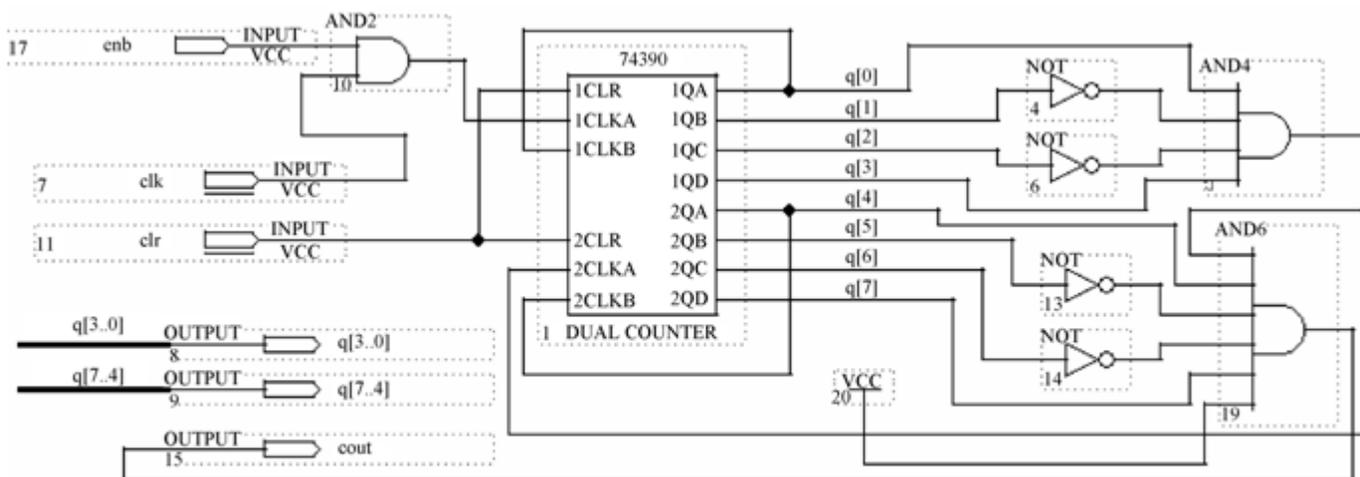


图 6-21 含有时钟使能的 2 位十进制计数器

实验与设计

6-1. 应用宏模块设计数字频率计

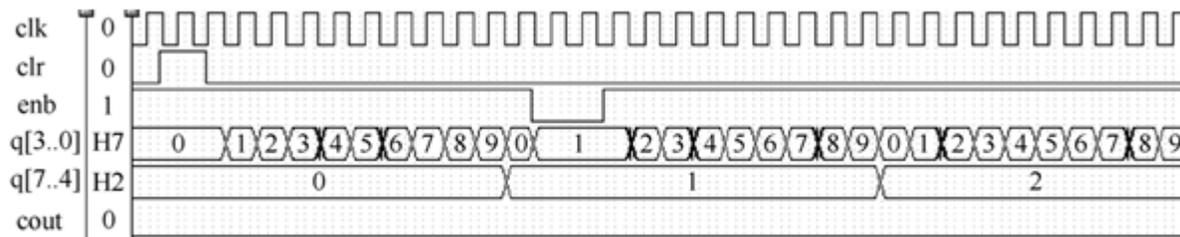


图 6-22 两位十进制计数器工作波形

实验与设计

6-1. 应用宏模块设计数字频率计

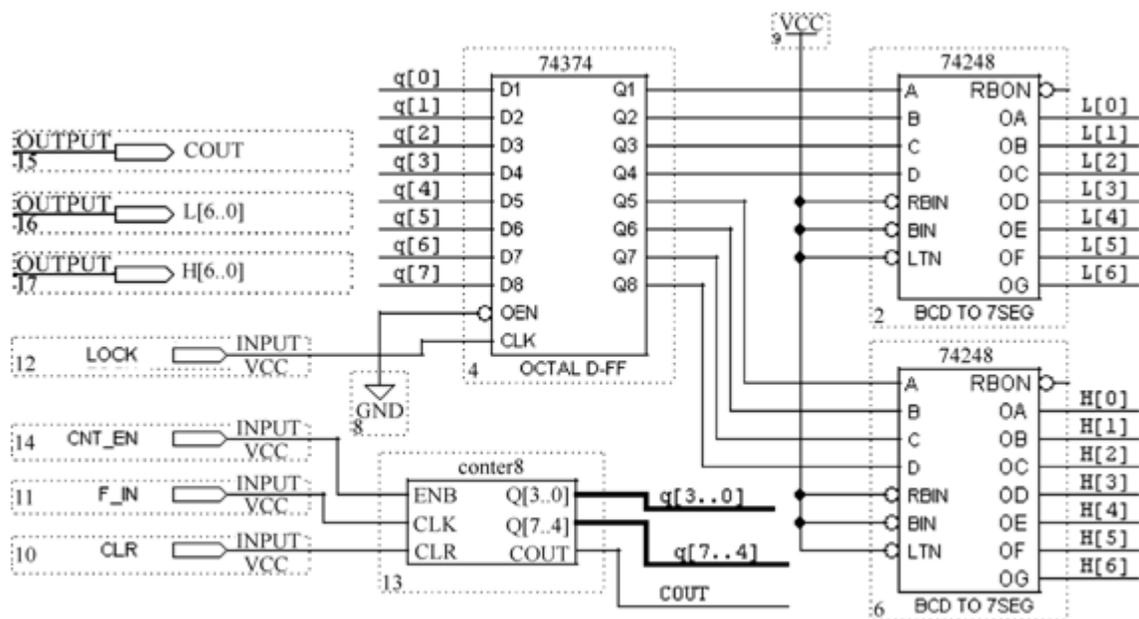


图 6-23 2 位十进制频率计顶层设计原理图文件

实验与设计

6-1. 应用宏模块设计数字频率计

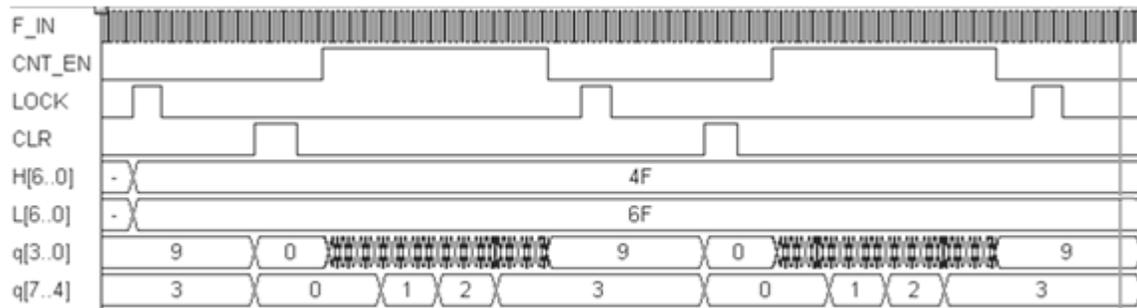


图 6-24 两位十进制频率计测频仿真波形

实验与设计

6-1. 应用宏模块设计数字频率计

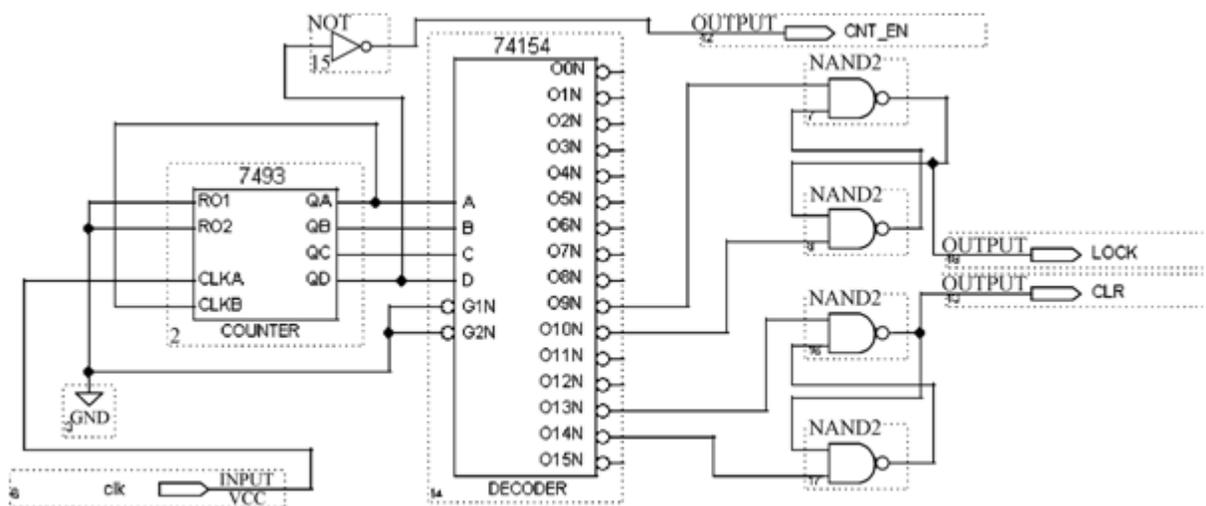


图 6-25 测频时序控制电路

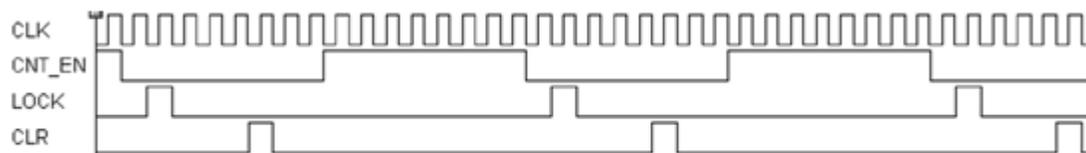


图 6-26 测频时序控制电路工作波形

实验与设计

6-1. 应用宏模块设计数字频率计

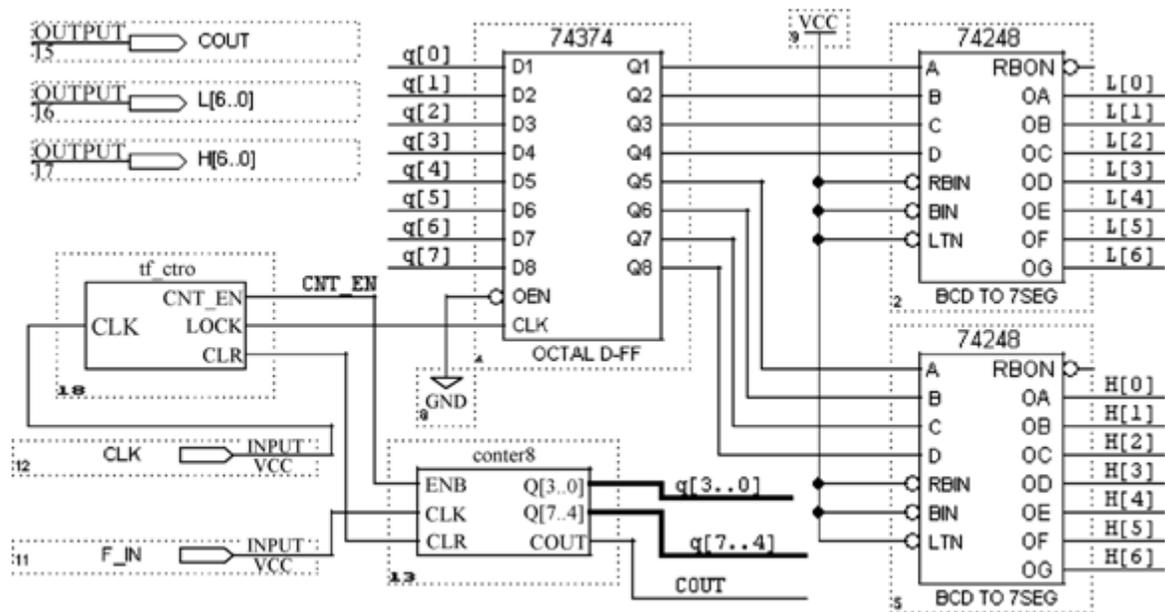


图 6-27 频率计顶层电路原理图

实验与设计

6-1. 应用宏模块设计数字频率计

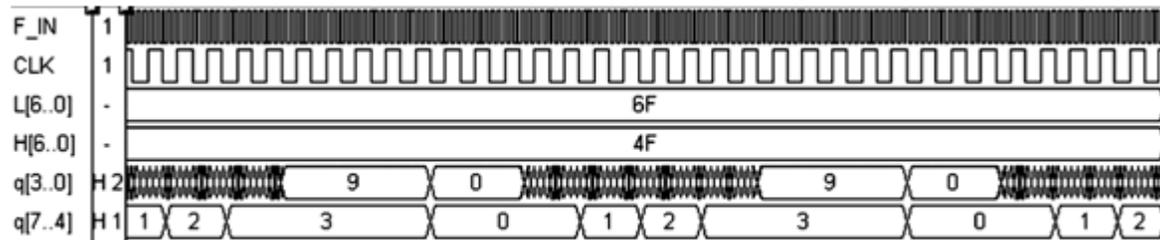


图 6-28 频率计工作时序波形

6-2. 计数器设计实验

实验与设计

6-3 数码扫描显示电路设计

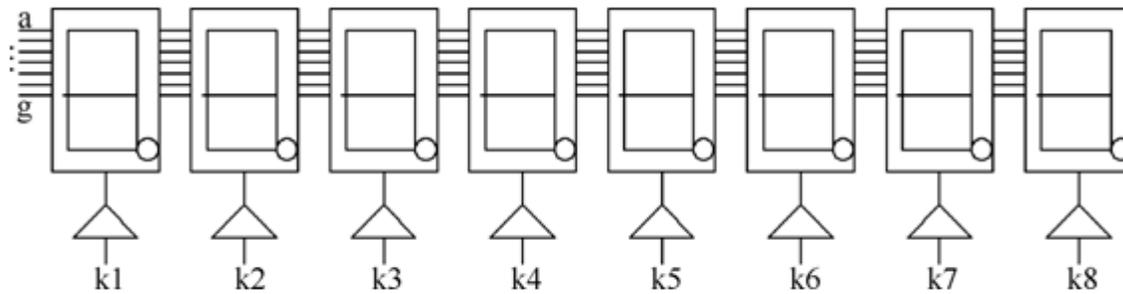


图 6-29 8 位数码扫描显示电路

实验与设计

6-4 串行静态显示控制电路设计

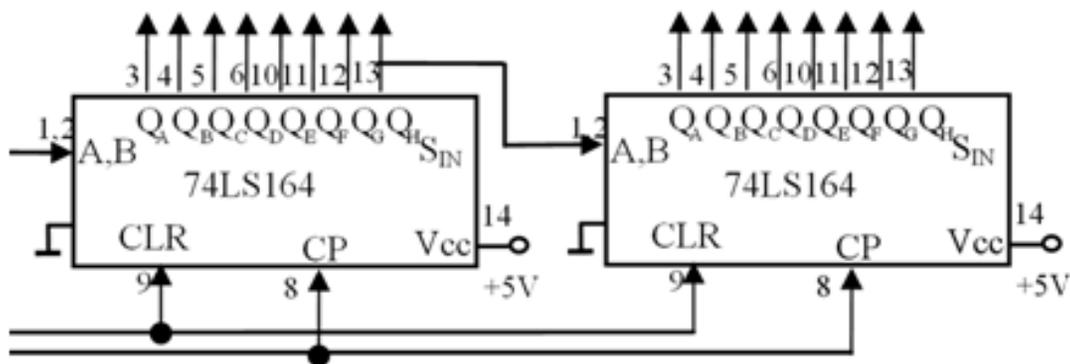


图 6-30 串/并转换数码管静态显示电路

实验与设计

6-5 基于VHDL代码的频率计设计

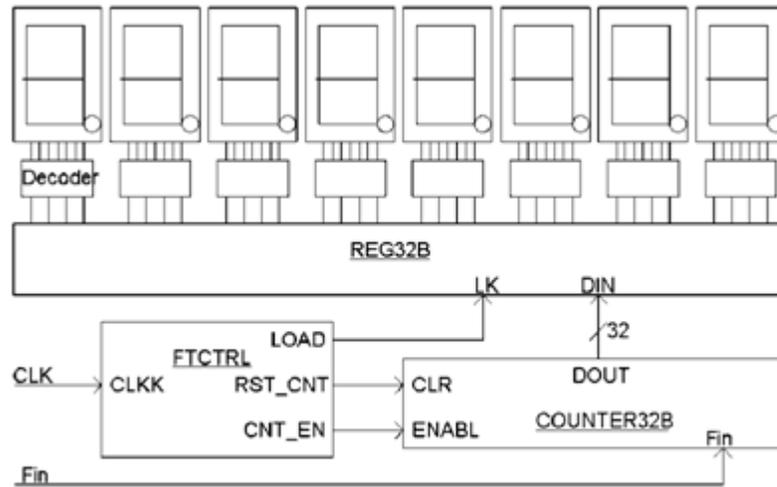


图 6-31 频率计电路框图

实验与设计

6-5 基于VHDL代码的频率计设计

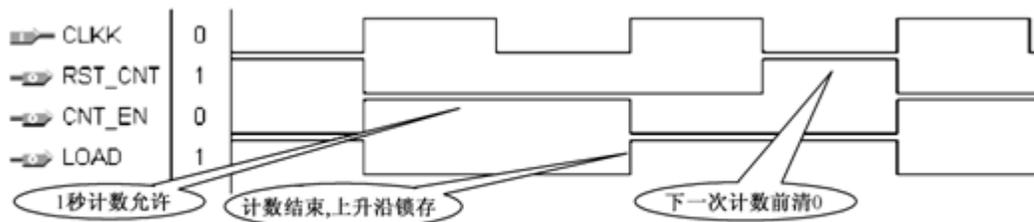


图 6-32 频率计测频控制器 FTCTRL 测控时序图

【例 6-1】

```
LIBRARY IEEE; --测频控制电路
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY FTCTRL IS
    PORT (CLKK : IN STD_LOGIC; -- 1Hz
          CNT_EN, RST_CNT : OUT STD_LOGIC; -- 计数器时钟使能和计数器清零
          Load : OUT STD_LOGIC ); -- 输出锁存信号
END FTCTRL;
ARCHITECTURE behav OF FTCTRL IS
    SIGNAL Div2CLK : STD_LOGIC;
BEGIN
    PROCESS( CLKK ) BEGIN
        IF CLKK'EVENT AND CLKK='1' THEN Div2CLK<=NOT Div2CLK;--1Hz 时钟 2 分频
        END IF;
    END PROCESS;
    PROCESS (CLKK, Div2CLK) BEGIN
        IF CLKK='0' AND Div2CLK='0' THEN RST_CNT<='1';-- 产生计数器清零信号
        ELSE RST_CNT <= '0'; END IF;
    END PROCESS;
    Load <= NOT Div2CLK; CNT_EN <= Div2CLK;
END behav;
```

实验与设计

6-6 VGA彩条信号显示控制电路设计

时钟频率 (lock frequency) : 25.175 MHz (像素输出的频率)

行频 (Line frequency) : 31469 Hz

场频 (Field frequency) : 59.94 Hz (每秒图像刷新频率)

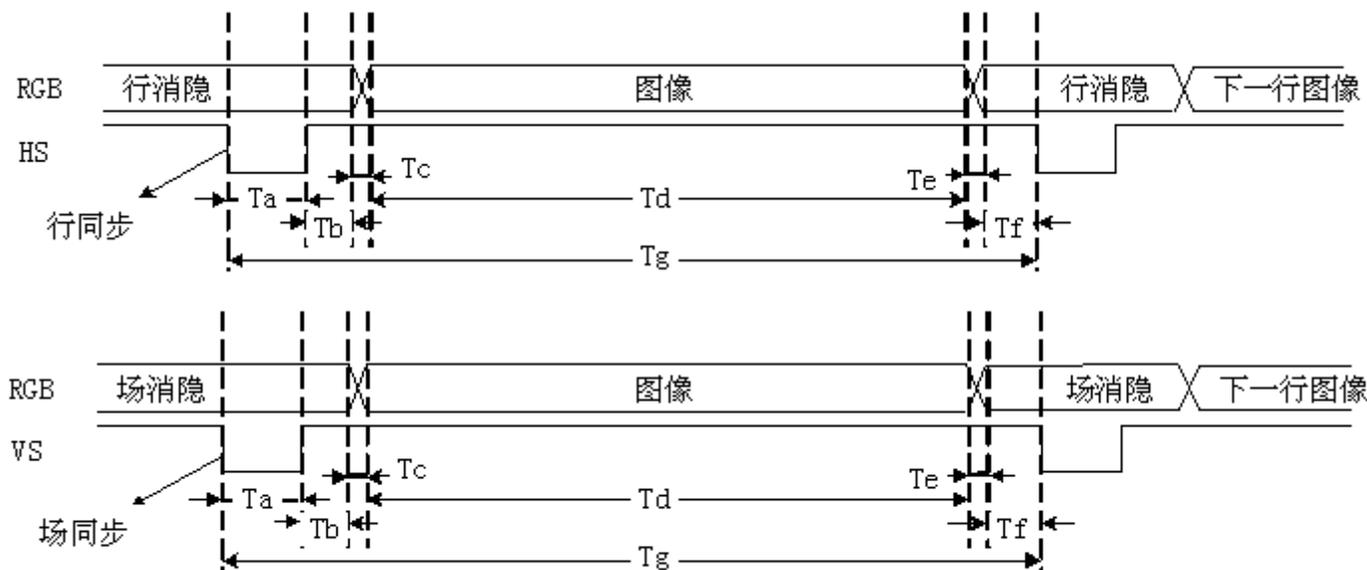


图 6-33 VGA 行扫描、场扫描时序示意图

实验与设计

6-6 VGA彩条信号显示控制电路设计

表 6-2 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 6-3 场扫描时序要求：(单元：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

6-6 VGA彩条信号显示控制电路设计

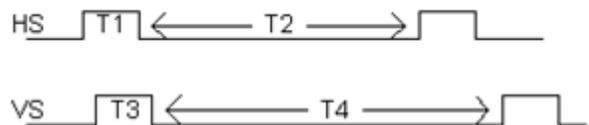


图 6-34 HS 和 VS 的时序图

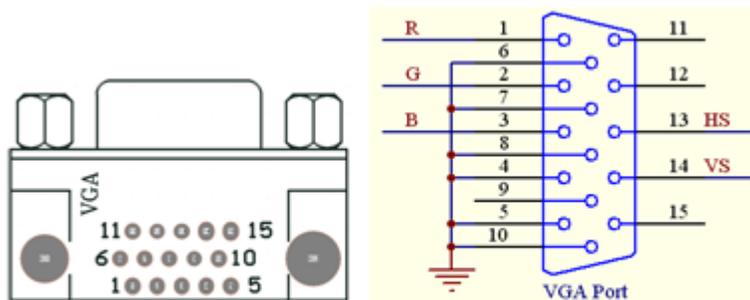


图 6-35 VGA 接口电路图，左接口从上往下看

实验与设计

6-6 VGA彩条信号显示控制电路设计

表 6-4 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 6-5 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

【例 6-2】

```
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, 兰控制
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 FS : 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
```

接下页

```

IF MD'EVENT AND MD = '0' THEN
    IF MMD = "10" THEN MMD <= "00";
    ELSE MMD <= MMD + 1; END IF;           -- 三种模式
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 FS = 13 THEN FS <= "0000"; ELSE FS <= (FS + 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

```

接下页

```

    IF CCLK'EVENT AND CCLK = '0' THEN
        IF LL=481 THEN LL<="000000000"; 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;

```