

第6章

Quartus II应用深入

6.1 SignalTap II的使用方法

1. 打开SignalTap II编辑窗口
2. 调入待测信号

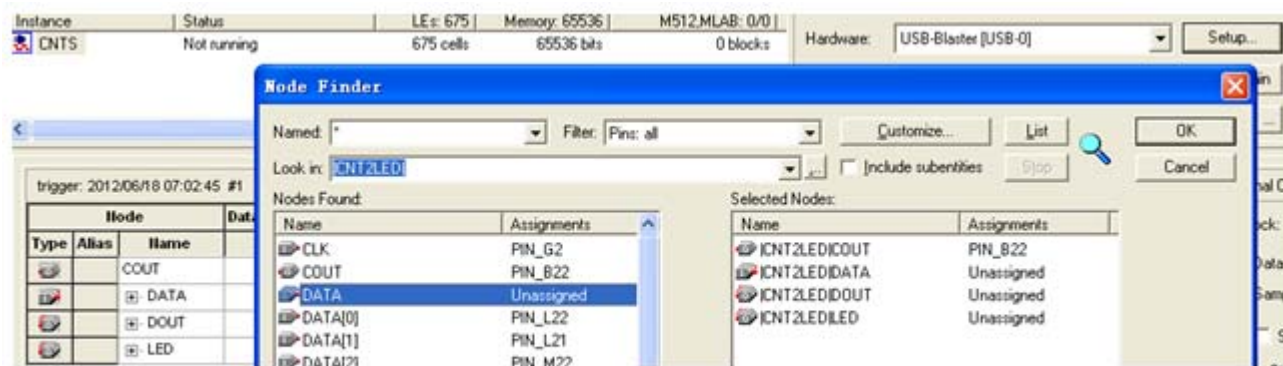


图 6-1 输入逻辑分析仪测试信号

6.1 SignalTap II 的使用方法

3. SignalTap II 参数设置



图 6-2 SignalTap II 编辑窗口

6.1 SignalTap II 的使用方法

4. 文件存盘

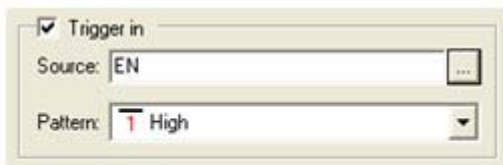


图 6-3 设置 EN 为触发信号

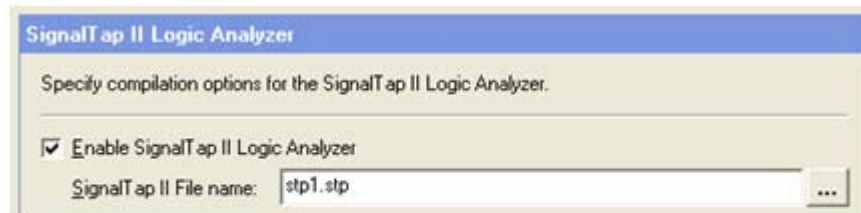


图 6-4 选择或删除 SignalTap II 文件加入综合编译

5. 编译下载

6.1 SignalTap II 的使用方法

6. 启动SignalTap II进行采样与分析

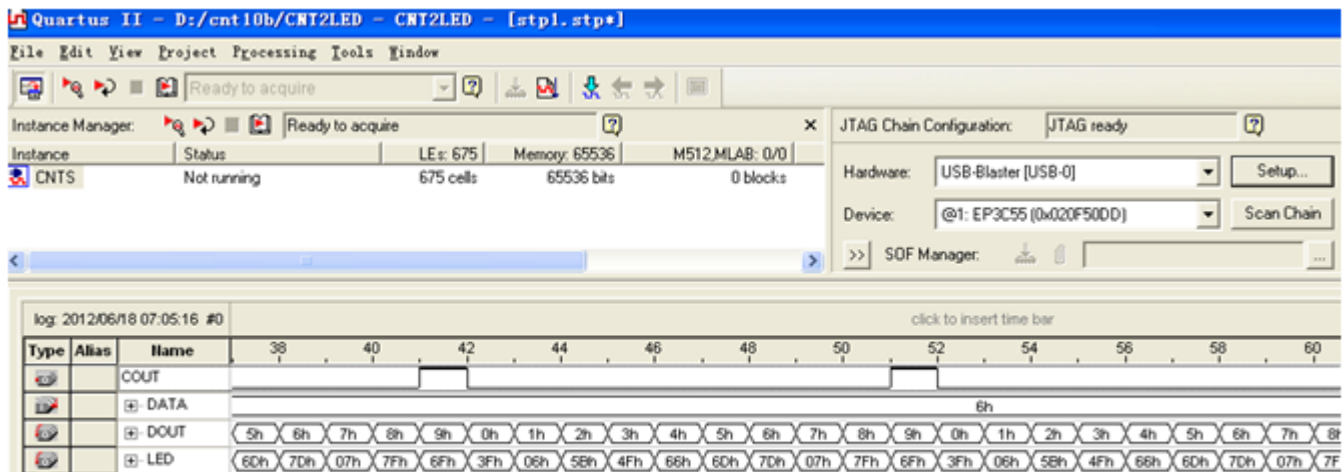


图 6-5 SignalTap II 实时数据采集显示界面

6.1 SignalTap II 的使用方法

6. 启动SignalTap II进行采样与分析

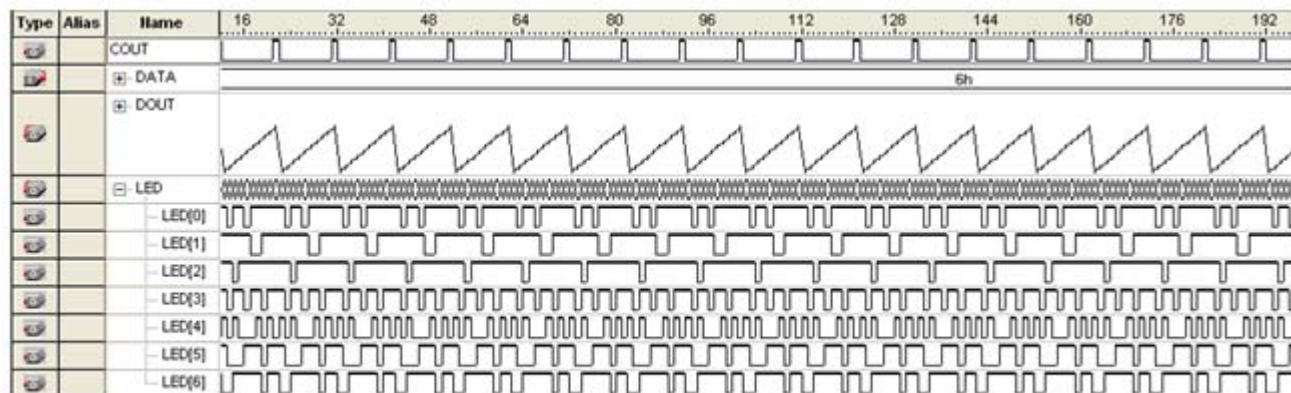


图 6-6 SignalTap II 数据窗口显示对硬件系统实时测试采样后的信号波形

7. SignalTap II 的其他设置和控制方法



6.2 编辑SignalTap II的触发信号

SignalTap II的触发信号也可单独设置或编辑，其触发控制逻辑也可根据实际需要由用户自行编辑。

6.3 Fitter Settings项设置

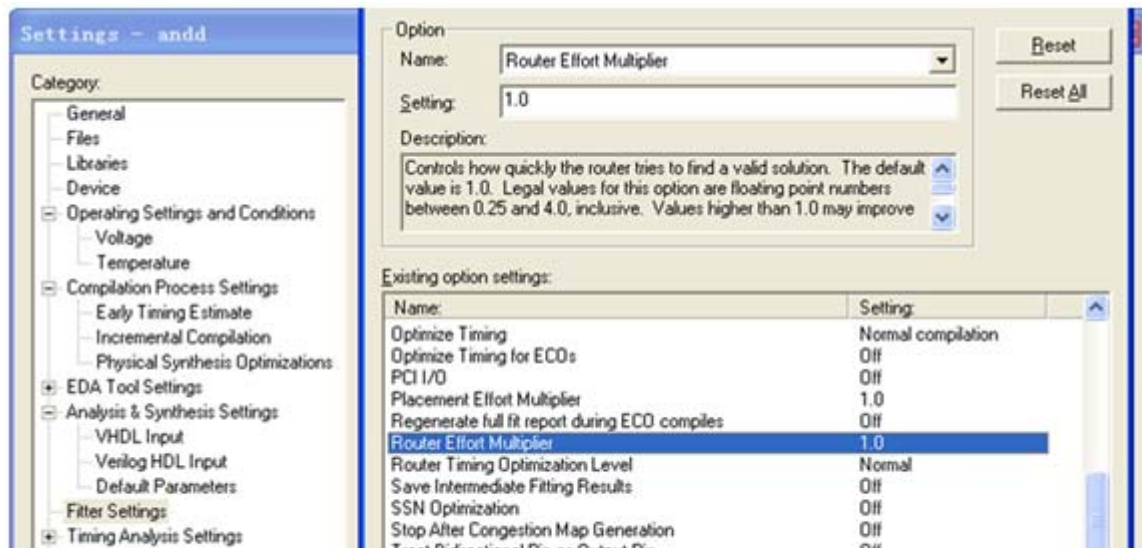


图 6-7 布线倍增器优化程度指数选择

6.4 功能块Chip Planner应用

6.4.1 Chip Planner应用流程说明

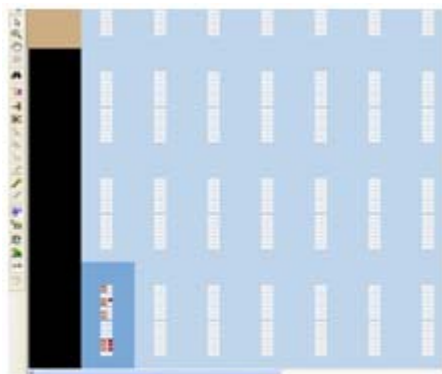


图 6-8 左下侧是已占用的 LAB

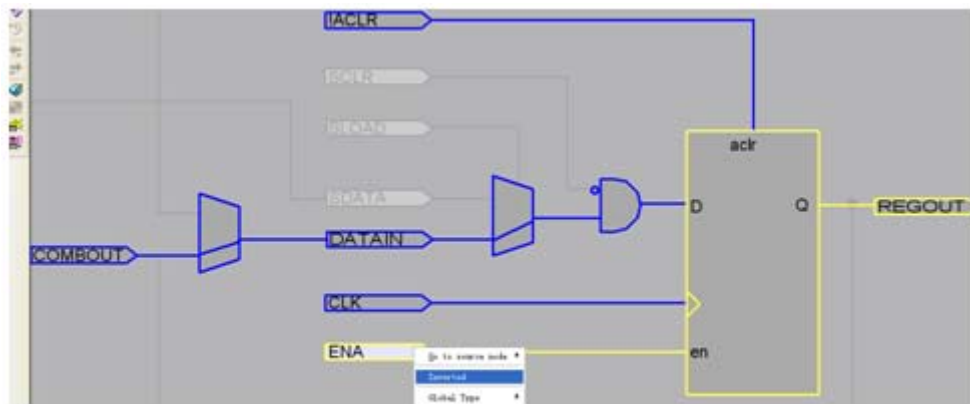


图 6-9 Resource Property Editor 的门级原理图编辑窗口

6.4 功能块Chip Planner应用

6.4.2 Chip Planner说明

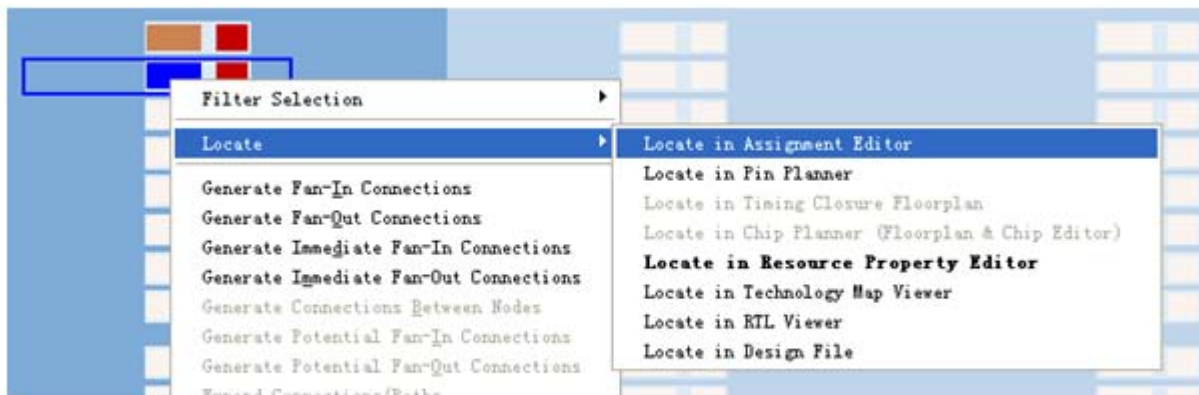
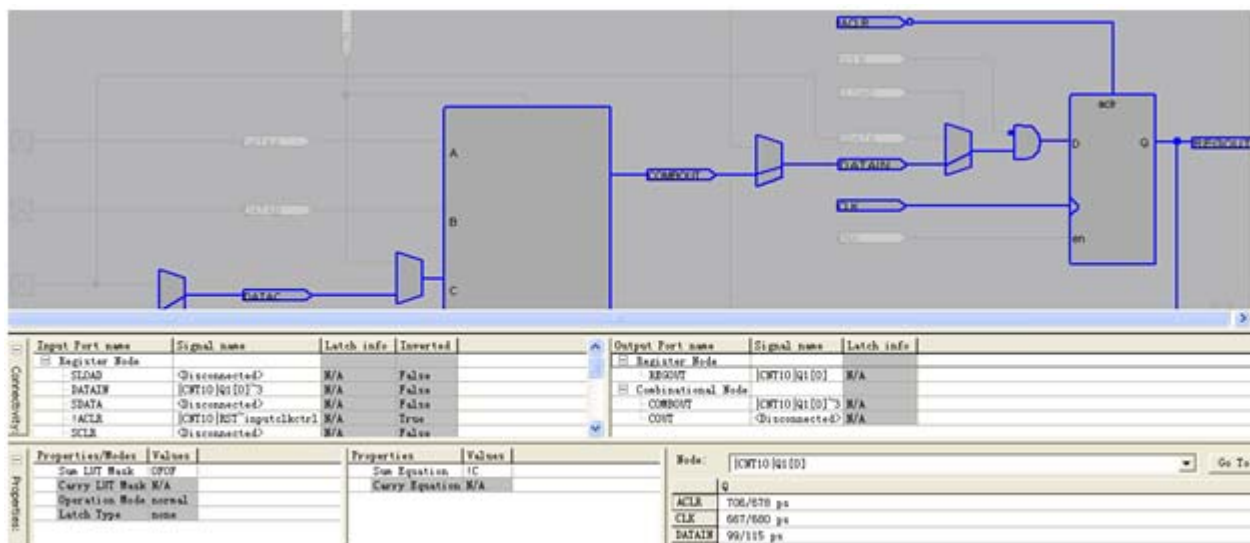


图 6-10 Locate 多项选择

6.4 功能块Chip Planner应用

6.4.2 Chip Planner说明



The screenshot displays the Chip Planner interface. The top portion shows a logic diagram with several input signals (DATA0, DATA1, DATA2, DATA3, DATA4, DATA5, DATA6, DATA7, DATA8, DATA9, DATA10, DATA11, DATA12, DATA13, DATA14, DATA15) connected to a logic block. The bottom portion shows the Property Windows, which are divided into several sections:

- Connectivity:** A table showing input and output port names, signal names, latch info, and inverted status.
- Properties:** A table showing properties and values for the logic block.
- Equation Editor:** A section for defining equations.
- Timing:** A section showing timing information for the logic block.

Input Port name	Signal name	Latch info	Inverted
Register Mode			
SD0A0	<Disconnected>	N/A	False
DATA0	[CMT10]Q[10] ?	N/A	False
SD0A1	<Disconnected>	N/A	False
SD0A2	<Disconnected>	N/A	False
DATA1	[CMT10]Q[11] ?	N/A	True
SD0A3	<Disconnected>	N/A	False
SD0A4	<Disconnected>	N/A	False

Property/Mode	Value
Sum LUT Mask	0x0F
Carry LUT Mask	N/A
Operation Mode	normal
Latch Type	none

Property	Value
Sum Equation	1C
Carry Equation	N/A

Mode:	[CMT10]Q[10]	Go To
Q		
ACL0	706/878 ps	
CLK	667/600 ps	
DATA0	99/115 ps	

图 6-11 打开属性和端口连接窗口

6.5 Synplify的应用及接口方法

6.5.1 Synplify使用流程

1. 启动Synplify

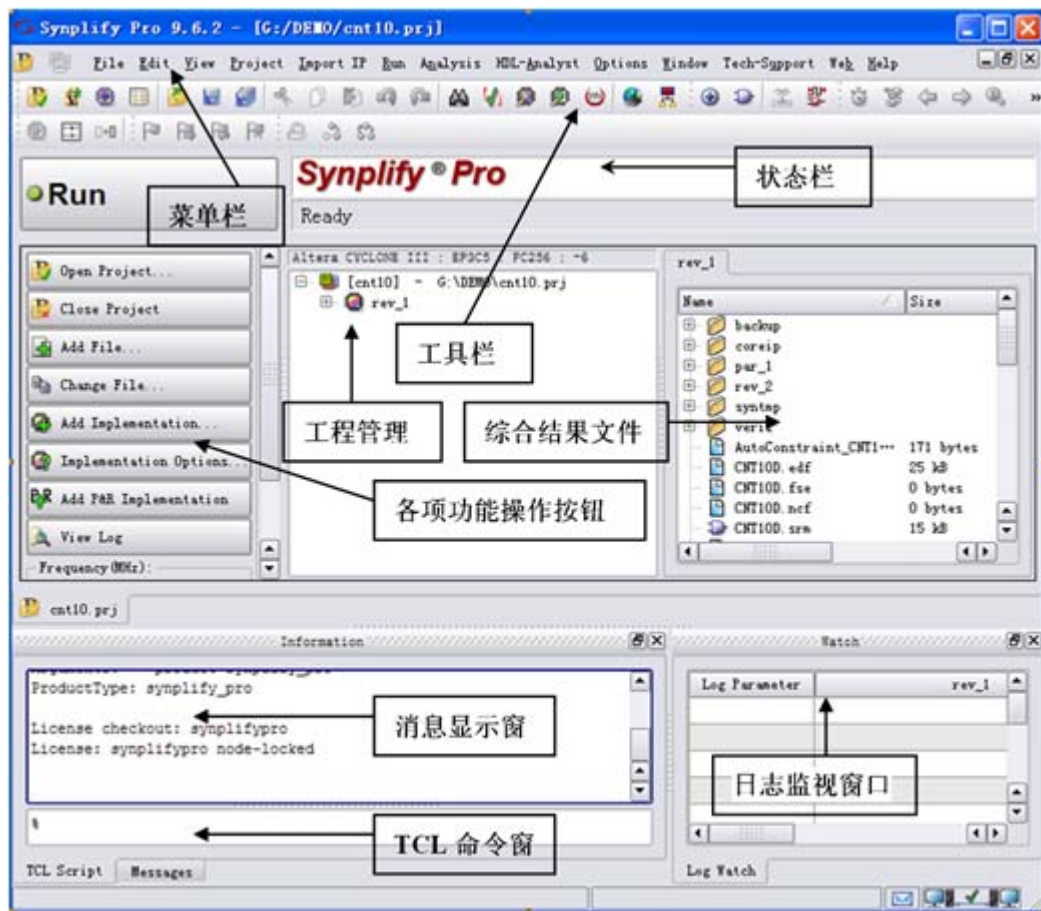


图6-12 Synplify Pro启动后界面

6.5 Synplify的应用及接口方法

6.5.1 Synplify使用流程

2. 创建工程

3. 加入源文件

4. 选择顶层文件

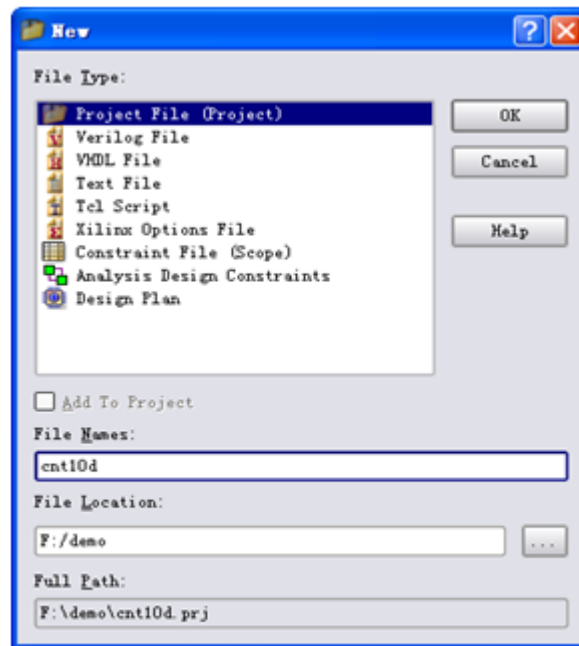


图6-13 创建工程

6.5 Synplify的应用及接口方法

6.5.1 Synplify使用流程

5. 设置工程属性

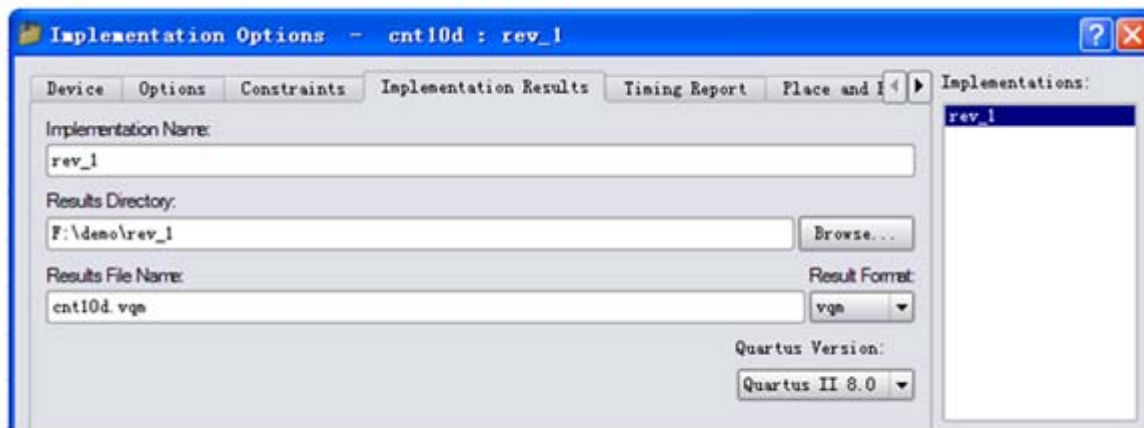
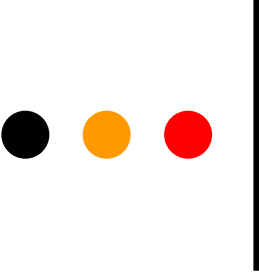


图6-14 综合目标详细信息



6.5 Synplify的应用及接口方法

6.5.1 Synplify使用流程

6. 综合前设置约束

7. 综合

8. 检测结果

6.5 Synplify的应用及接口方法

6.5.2 Synplify Pro与Quartus II接口

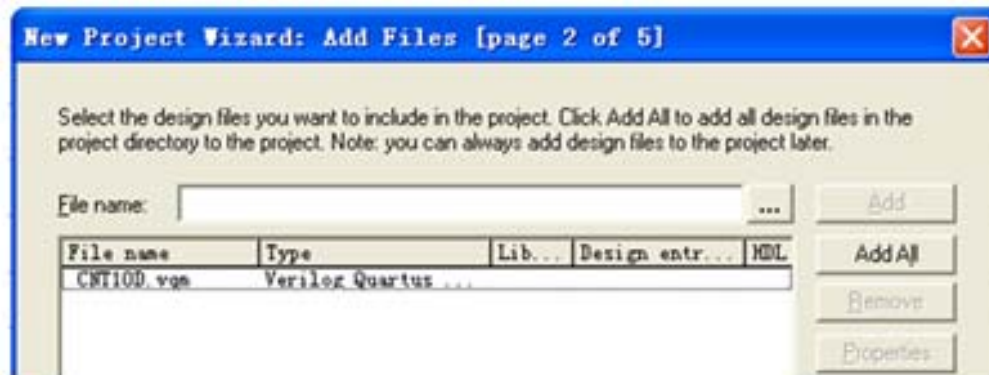


图 6-15 加入 Cnt10d.vqm 文件

6.5 Synplify的应用及接口方法

6.5.2 Synplify Pro与Quartus II接口

1. Synplify软件路径设置

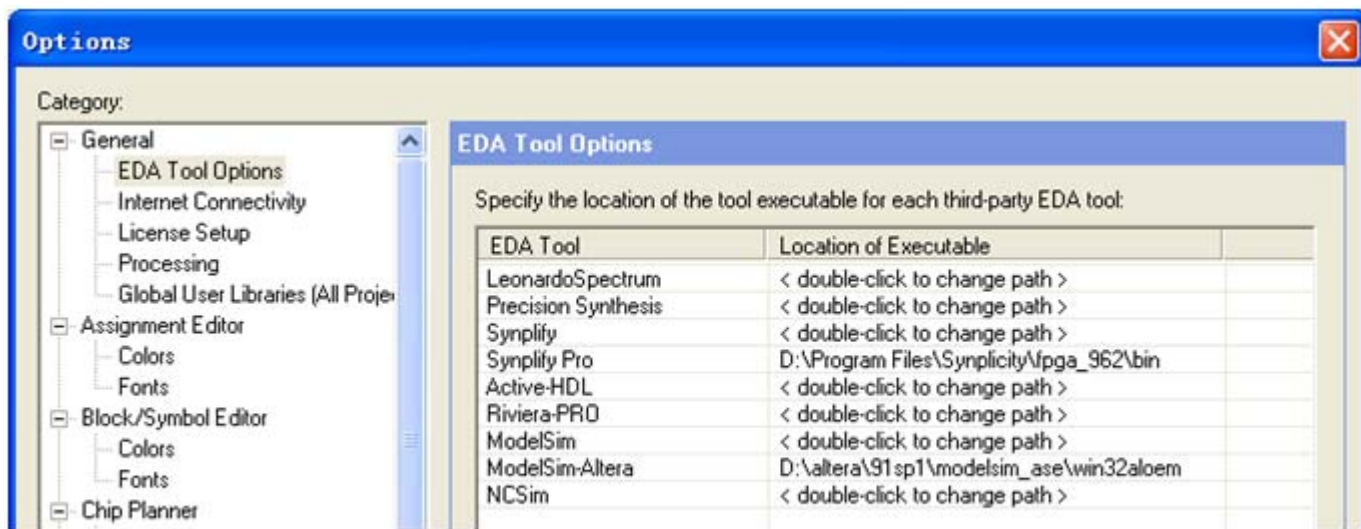


图6-16 Synplify软件路径设置

6.5 Synplify的应用及接口方法

6.5.2 Synplify Pro与Quartus II接口

2. 设置Synplify Pro综合器

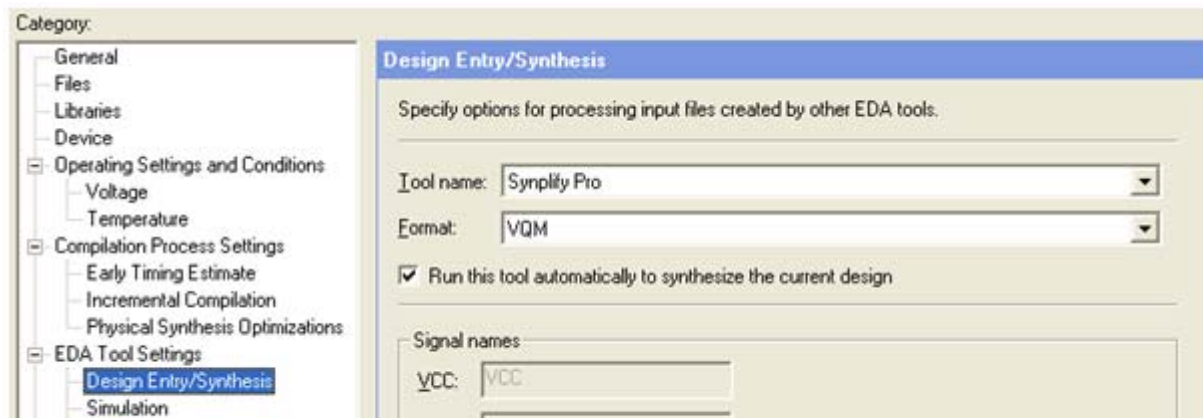


图 6-17 设置综合器

实验与设计

6-1 VGA彩条信号显示控制电路设计

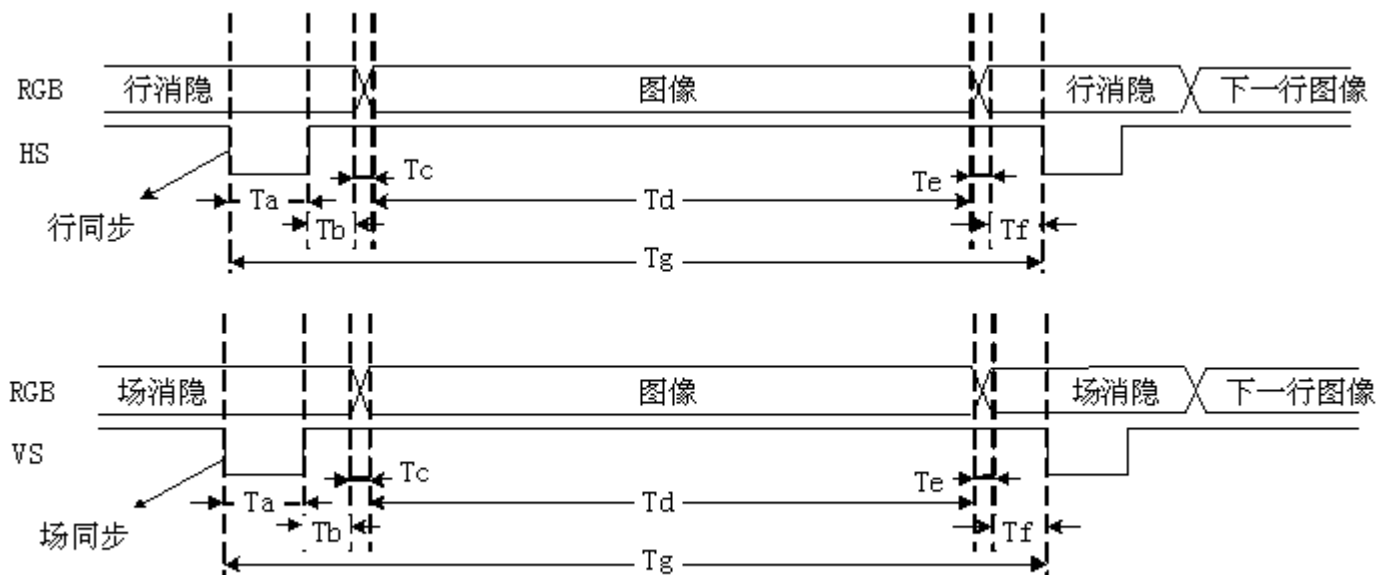
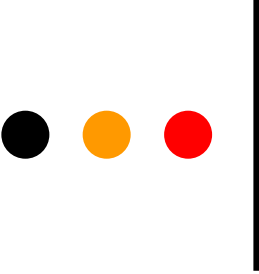


图 6-18 VGA 行扫描、场扫描时序示意图



实验与设计

6-1 VGA彩条信号显示控制电路设计

表 6-1 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 6-2 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

6-1 VGA彩条信号显示控制电路设计

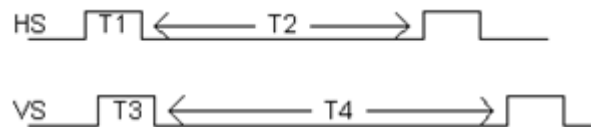


图 6-19 HS 和 VS 的时序图

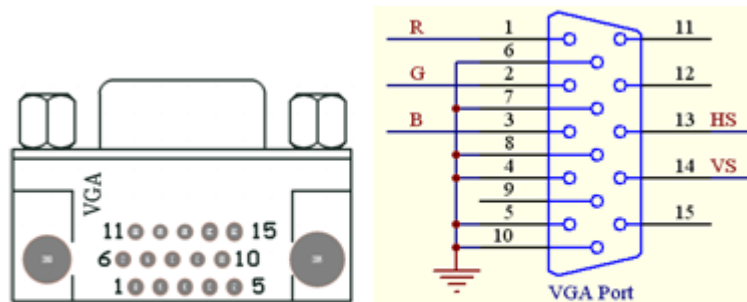


图 6-20 VGA 接口电路图，左接口从上往下看

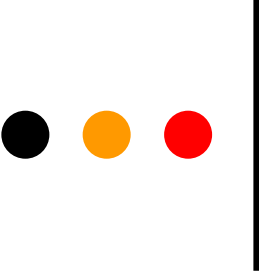
【例 6-1】

```
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
        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
    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 ;

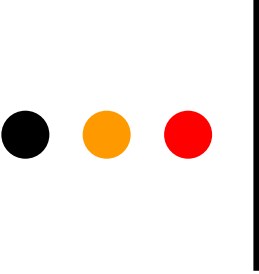
```



实验与设计

6-1 VGA彩条信号显示控制电路设计

```
        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

6-1 VGA彩条信号显示控制电路设计

表 6-3 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 6-4 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

6-3 半整数与奇数分频器设计

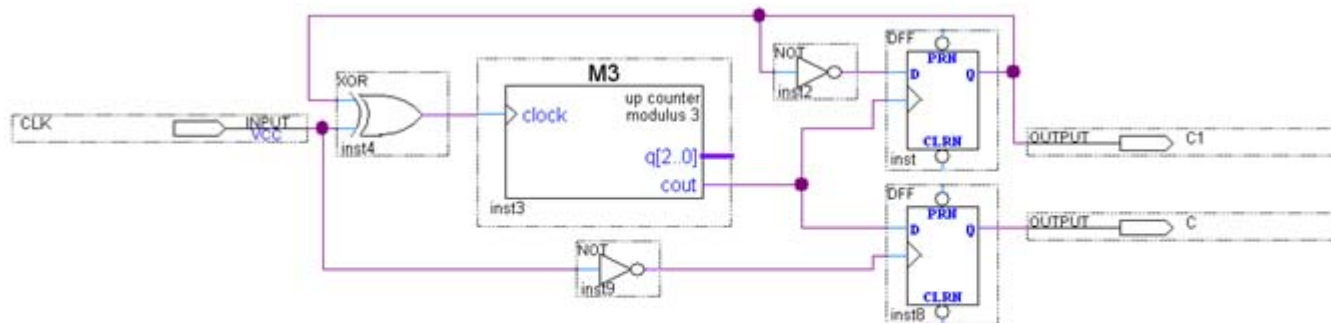


图 6-23 占空比为 50%的任意奇数次分频电路

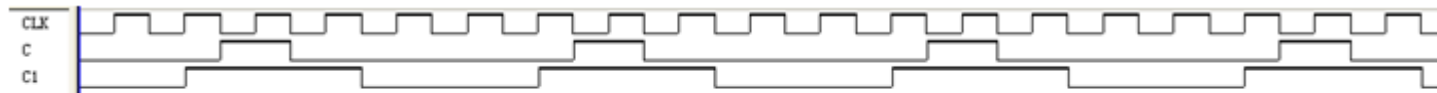
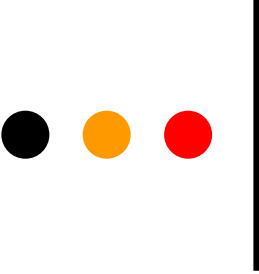


图 6-24 图 6-23 电路的仿真波形



实验与设计

6-3 半整数与奇数分频器设计

【例 6-2】 占空比为 50%的任意奇数次 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

6-3 半整数与奇数分频器设计



图 6-25 占空比为 50%的任意奇数次分频电路

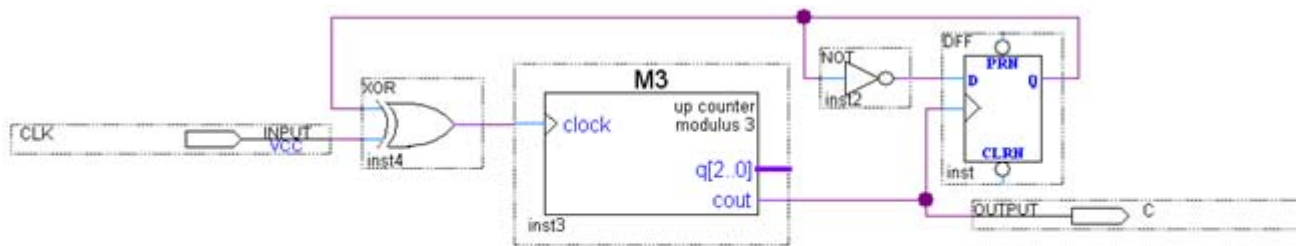


图 6-26 任意半整数分频电路



图 6-27 图 6-26 电路仿真波形图

实验与设计

6-4 基于Verilog代码的频率计设计

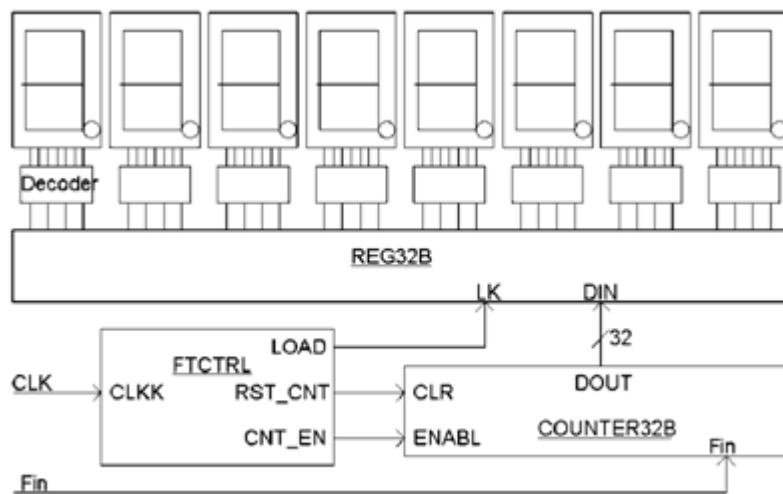


图 6-28 频率计电路框图

实验与设计

6-4 基于Verilog代码的频率计设计

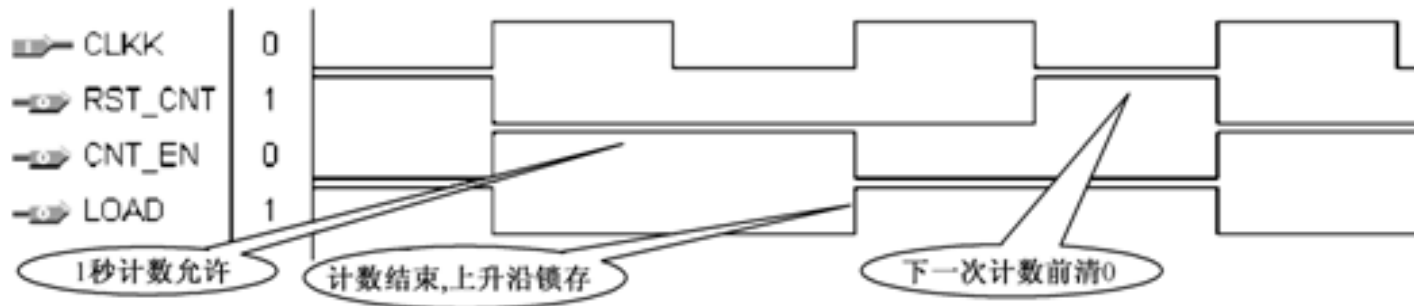
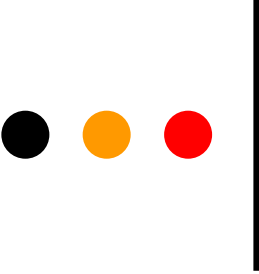


图 6-29 频率计测频控制器 FTCTRL 测控时序图



实验与设计

6-4 基于Verilog代码的频率计设计

【例 6-3】

```
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
```