《FPGA 应用开发入门与典型实例》（修订版）

作者：华清远见

第 2 章　从零开始设计 FPGA 最小系统

本章目标

- 了解 FPGA 硬件系统的调试步骤和方法
- 了解 FPGA 常用接口电路的原理和设计
- 掌握 FPGA 最小系统的概念和组成电路

专业始于专注 卓识源于远见

2.1 FPGA 最小系统的概念

FPGA 最小系统是可以使 FPGA 正常工作的最简单的系统。它的外围电路尽量最少，只包括 FPGA 必要的控制电路。

一般所说的 FPGA 的最小系统主要包括：FPGA 芯片、下载电路、外部时钟、复位电路和电源。如果需要使用 NIOS II 软嵌入式处理器还要包括：SDRAM 和 Flash。一般以上这些组件是 FPGA 最小系统的组成部分。红色飓风 II 代 Altera 开发板功能框图如图 2.1 所示。

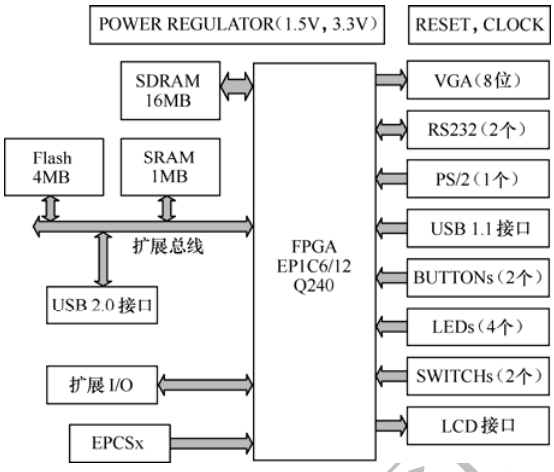


图 2.1 红色飓风 II 代 Altera 开发板功能框图

红色飓风 II 代 Altera 开发板的总线接口信号如图 2.2 所示。

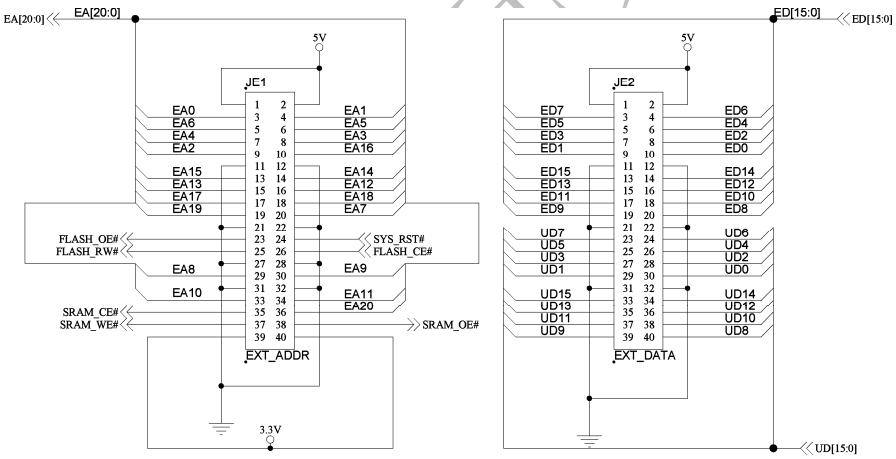


图 2.2 开发板的总线扩展接口

2.2 FPGA 最小系统电路分析

2.2.1 FPGA 管脚设计

FPGA 的管脚主要包括：用户 I/O（User I/O）、配置管脚、电源、时钟及特殊应用管脚等。其中有些管脚可有多种用途，所以在设计 FPGA 电路之前，需要认真的阅读相应 FPGA 的芯片手册。

下面以 Altera 公司的 Cyclone 系列 FPGA 为例，介绍 FPGA 的各种功能管脚。

（1）用户 I/O。

I/O*num*（LVDS*num*n）：可用作输入或输出，或者双向口，同时可作为 LVDS 差分对的负端。其中 *num* 表示管脚序号。

一般在绘制 FPGA 原理图时，将同一种功能和用途的管脚放在一个框图中，如图 2.3 所示是用户 I/O 的原理图。

（2）配置管脚。

- MSEL[1..0]：用于选择配置模式。FPGA 有多种配置模式，比如主动、被动、快速、正常、串行、并行等，可以此管脚进行选择。
- DATA0：FPGA 串行数据输入，连接至配置器件的串行数据输出管脚。
- DCLK：FPGA 串行时钟输出，为配置器件提供串行时钟。
- nCSO（I/O）：FPGA 片选信号输出，连接至配置器件的 nCS 管脚。
- ASDO（I/O）：FPGA 串行数据输出，连接至配置器件的 ASDI 管脚。
- nCEO：下载链器件使能输出。在一条下载链（Chain）中，当第一个器件配置完成后，此信号将使能下一个器件开始进行配置。下载链的最后一个器件的 nCEO 应悬空。

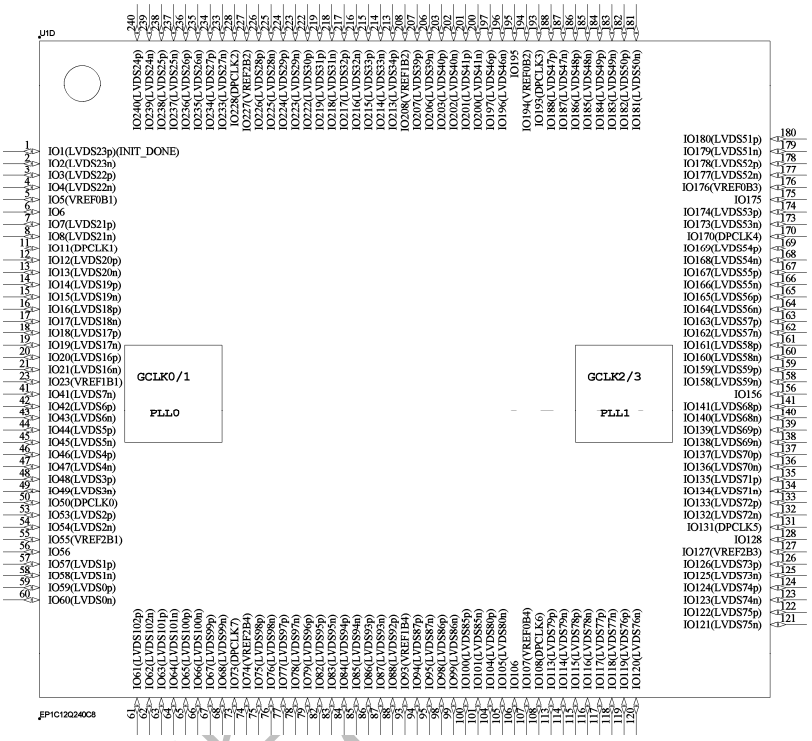


图 2.3　FPGA 用户 I/O 原理图

- nCE：下载链器件使能输入，连接至上一个器件的 nCEO。下载链第一个器件的 nCE 接地。
- nCONFIG：用户模式配置起始信号。
- nSTATUS：配置状态信号。
- CONF_DONE：配置结束信号。

如图 2.4 所示是 FPGA 配置管脚原理图。

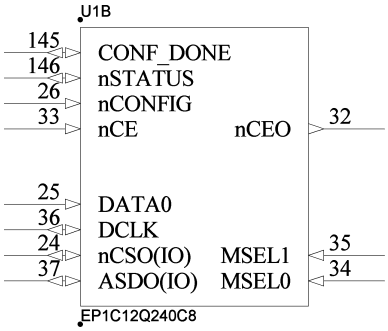


图 2.4　FPGA 配置管脚原理图

（3）电源管脚。

- VCCINT：内核电压。通常与 FPGA 芯片所采用的工艺有关，例如 130nm 工艺为 1.5V，90nm 工艺为 1.2V。
- VCCIO：端口电压。一般为 3.3V，还可以支持选择多种电压，如 5V、1.8V、1.5V 等。

- VREF：参考电压。
- GND：信号地。

如图 2.5 所示是 FPGA 电源管脚原理图。

（4）时钟管脚。

- VCC_PLL：锁相环管脚电压，直接连 VCCIO。
- VCCA_PLL：锁相环模拟电压，一般通过滤波器接到 VCCINT 上。
- GNDA_PLL：锁相环模拟地。
- GNDD_PLL：锁相环数字地。
- CLK*num*(LVDSCLK*num*p)：锁相环时钟输入。支持 LVDS 时钟输入，p 接正端，num 表示 PLL 序号。
- CLK*num*(LVDSCLK*num*n)：锁相环时钟输入。支持 LVDS 时钟输入，n 接负端，num 表示 PLL 序号。
- PLL*num*_OUTp(I/O)：锁相环时钟输出。支持 LVDS 时钟输入，p 接正端，num 表示 PLL 序号。
- PLL*num*_OUTn(I/O)：锁相环时钟输出。支持 LVDS 时钟输入，n 接负端，num 表示 PLL 序号。

如图 2.6 所示是 FPGA 时钟管脚原理图。

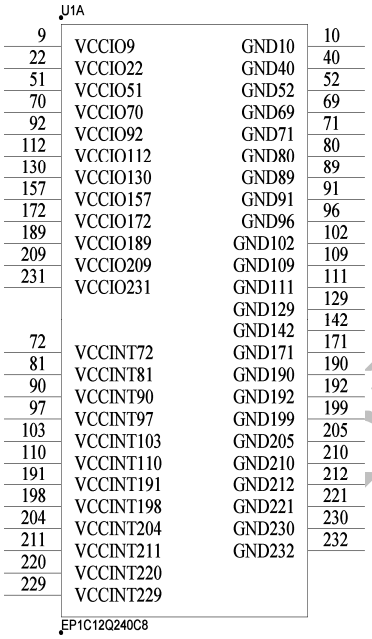


图 2.5　FPGA 电源管脚原理图

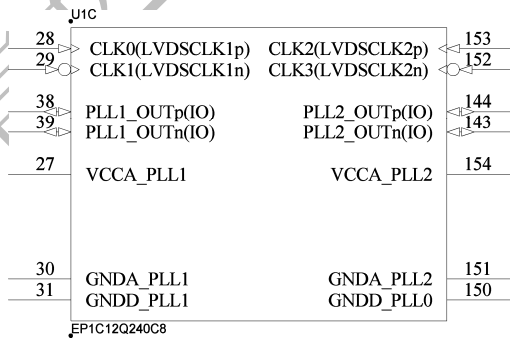


图 2.6　FPGA 时钟管脚原理图

另外，FPGA 的管脚中，有一些是全局时钟，这些管脚在 FPGA 中已经做好了时钟树。使用这些管脚作为关键时钟或信号的布线可以获得最佳性能。

（5）特殊管脚。

- VCCPD：用于选择驱动电压。
- VCCSEL：用于控制配置管脚和锁相环相关的输入缓冲电压。
- PORSEL：上电复位选项。
- NIOPULLUP：用于控制配置时所使用的用户 I/O 的内部上拉电阻是否工作。
- TEMPDIODEn/p：用于关联温度敏感二极管。

2.2.2　下载配置与调试接口电路设计

FPGA 是 SRAM 型结构，本身并不能固化程序。因此 FPGA 需要一片 Flash 结构的配置芯片来存储逻辑配置信息，用于进行上电配置。

以 Altera 公司的 FPGA 为例，配置芯片分为串行（EPCSx 系列）和并行（EPCx 系列）两种。其中 EPCx

系列为老款配置芯片，体积较大，价格高。而 EPCSx 系列芯片与之相比，体积小、价格低。
另外，除了使用 Altera 公司的配置芯片，也可以使用 Flash+CPLD 的方式去配置 FPGA。
在把程序固化到配置芯片之前，一般先使用 JTAG 模式去调试程序，也就是把程序下载到 FPGA 芯片上运行。虽然这种方式在断电以后程序会丢失，但是充分利用了 FPGA 的无限擦写性。
所以一般 FPGA 有两个下载接口：JTAG 调试接口和 AS（或 PS）模式下载接口。所不同的是前者下载至 FPGA，后者是编程配置芯片（如 EPCSx），然后再配置 FPGA。
如图 2.7 和图 2.8 所示分别是 JTAG 模式和 AS 模式的电路原理图。

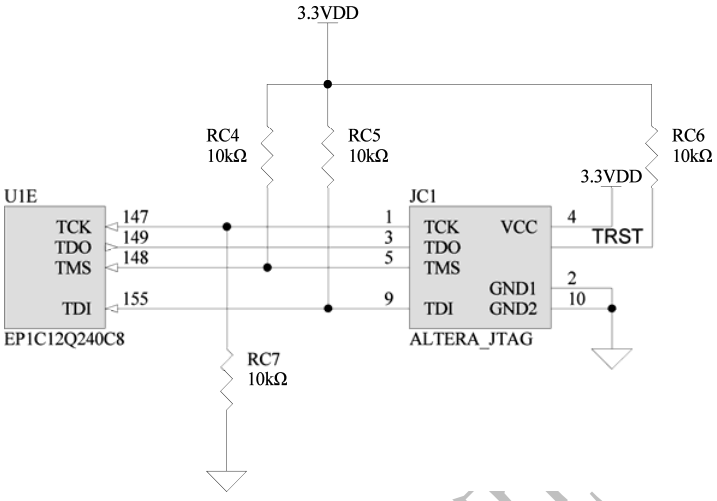


图 2.7　JTAG 模式原理图

2.2.3　高速 SDRAM 存储器接口电路设计

SDRAM 可作为软嵌入式系统的（NIOS II）的程序运行空间，或者作为大量数据的缓冲区。SDRAM 是通用的存储设备，只要容量和数据位宽相同，不同公司生产的芯片都是兼容的。
一般比较常用的 SDRAM 包括现代 HY57V 系列、三星 K4S 系列和美光 MT48LC 系列。例如，4M×32 位的 SDRAM，现代公司的芯片型号为 HY57V283220，三星公司的为 K4S283232，美光公司的为 MT48LC4M32。这几个型号的芯片可以相互替换。
SDRAM 典型电路如图 2.9 所示。

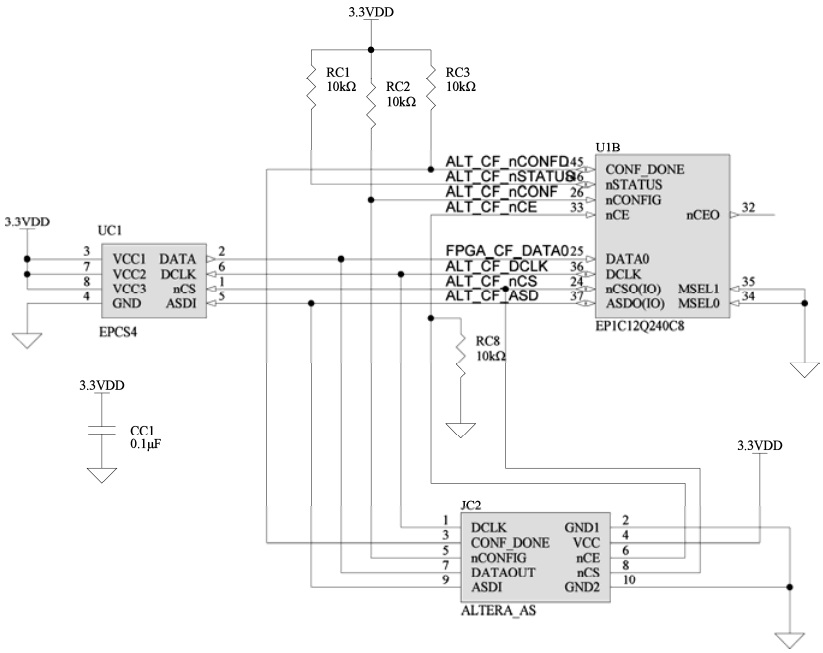


图 2.8　AS 模式原理图

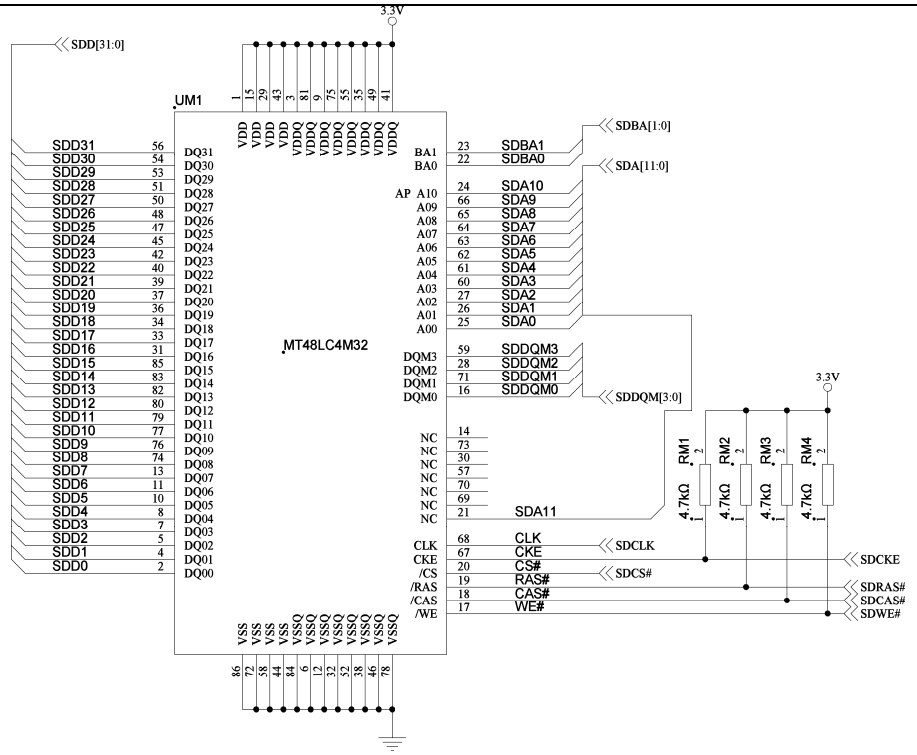


图 2.9　SDRAM 典型电路

2.2.4　异步 SRAM（ASRAM）存储器接口电路设计

由于 ASRAM 的读写时序相对比较简单，因此一般使用 SRAM 作为数据的缓冲，但其成本相对 SDRAM 高。而且作为异步设备，ASRAM 对于时钟同步的要求也不高，可以在低速下运行。ASRAM 主要为 8 位和 16 位数据宽度，用户可根据需要进行选择。

ASRAM 的典型电路如图 2.10 所示。

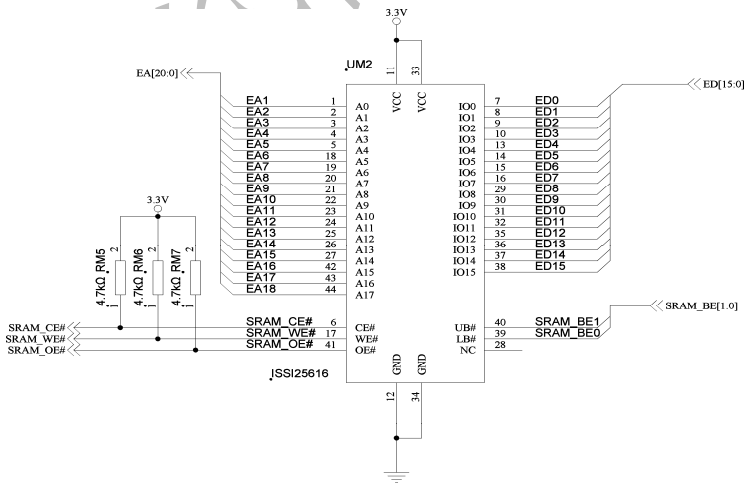


图 2.10　ASRAM 典型电路

2.2.5　Flash 存储器接口电路设计

Flash 可作为软嵌入式系统的程序存储空间，或者作为程序的固件空间。最常使用的是 AMD 公司或者 Intel 公司的 Flash。在小容量的 Flash 选择上，AMD 公司的 Flash 性价比较高，而高容量的 Flash 选择上，Intel 公司的 Flash 性价比较高。

Flash 同样也可以通过设置实现 8 位和 16 位的数据位宽，下面是几种典型的 Flash 应用。

16 位模式下的（AMD）Flash 连接如图 2.11 所示。

8 位模式下的（AMD）Flash 连接如图 2.12 所示。

8 位模式下（Intel）Flash 连接如图 2.13 所示。

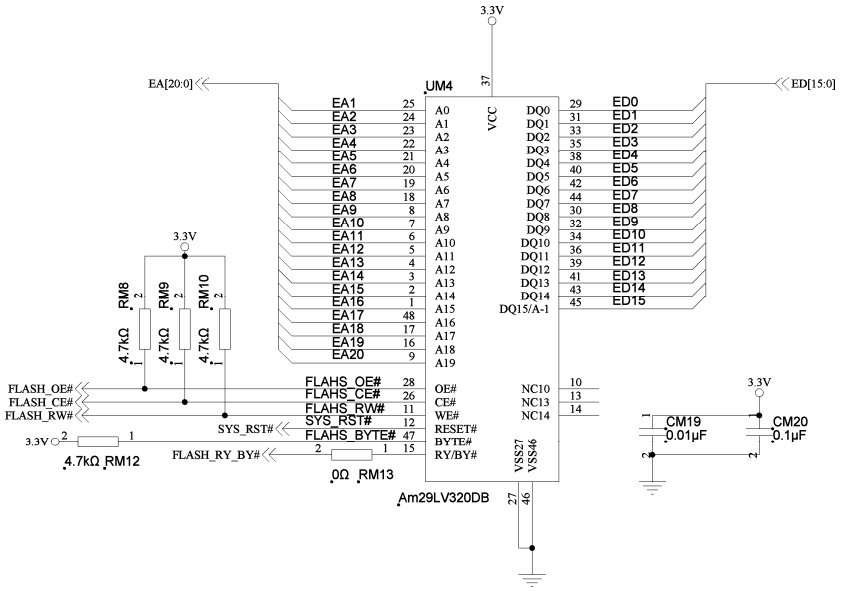


图 2.11　16 位模式下（AMD）Flash 连接

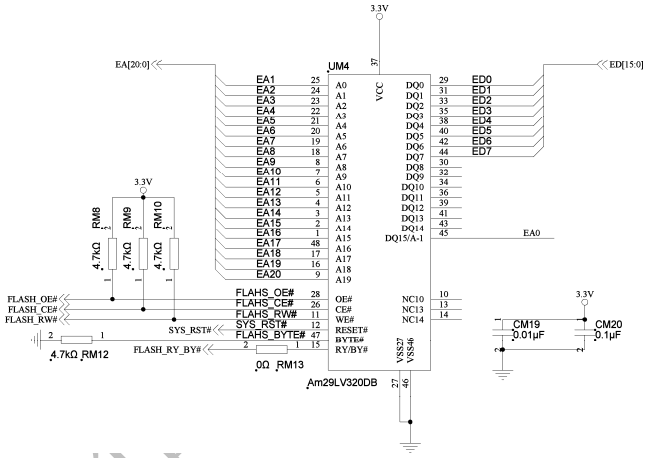


图 2.12　8 位模式下（AMD）Flash 连接

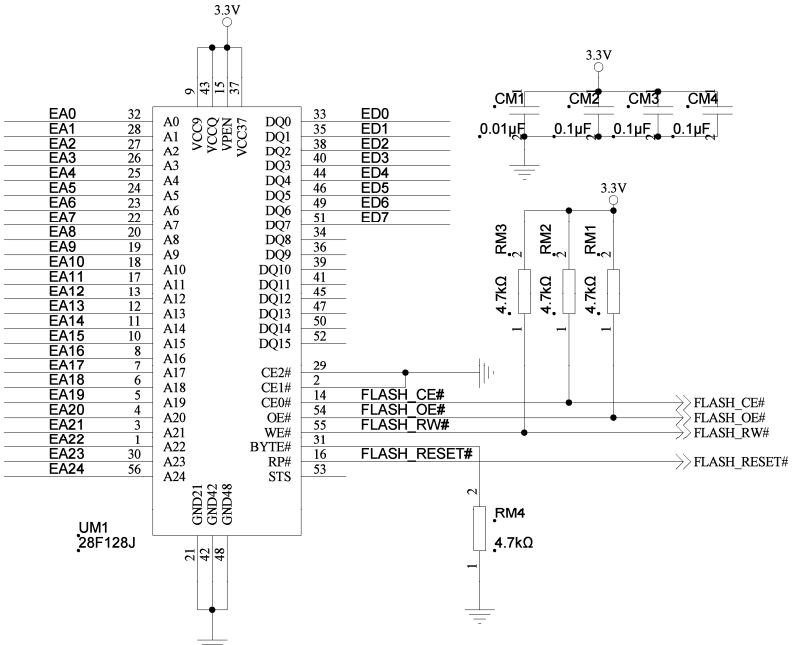


图 2.13　8 位模式下（Intel）Flash 连接

2.2.6 开关、按键与发光 LED 电路设计

发光 LED 参考电路如图 2.14 所示。

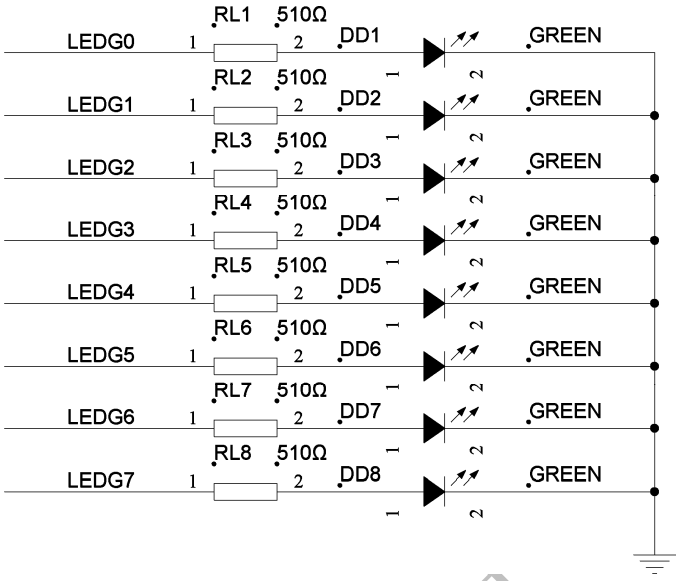


图 2.14 数码管参考电路

拨码开关参考电路如图 2.15 所示。

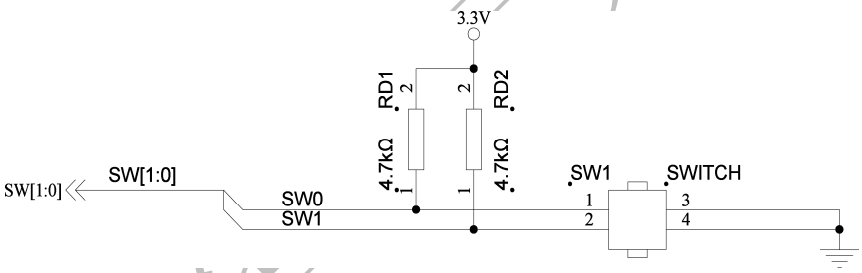


图 2.15 拨码开关参考电路

按键开关参考电路如图 2.16 所示。

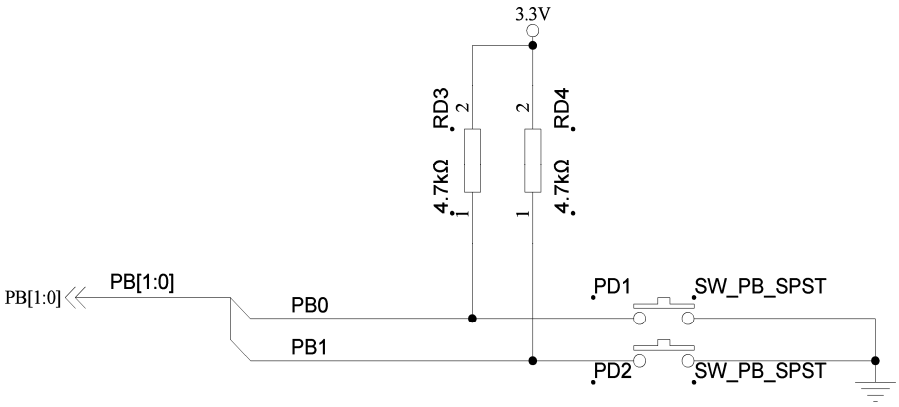


图 2.16 按键开关参考电路

2.2.7 VGA 接口电路设计

红色飓风开发板提供了 VGA 显示功能与接口，可以用普通的 VGA 电缆连接到计算机的显示器上。VGA 连接器定义如图 2.17 所示。

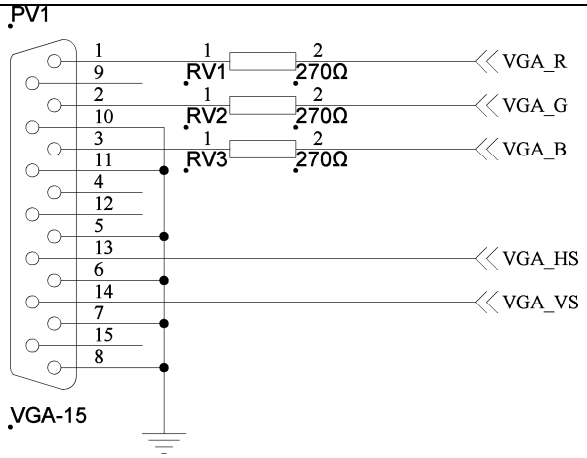


图 2.17　VGA 连接器定义

包括的信号有 Red（R）、Green（G）、Blue（B）、Horizontal Sync（水平扫描 HS）以及 Vertical Sync（垂直扫描 VS）。系统结构示意图如图 2.18 所示。

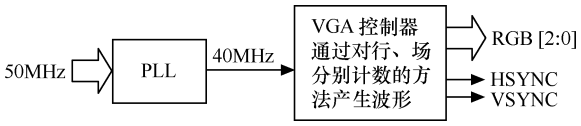


图 2.18　VGA 接口结构示意图

2.2.8　PS/2 鼠标及键盘接口电路设计

早期的 PS/2 鼠标及键盘采用 5V 电压标准，目前的 PS/2 鼠标及键盘主要采用 3.3V 电压标准，如图 2.19 所示的参考电路可以实现对两种标准的兼容。

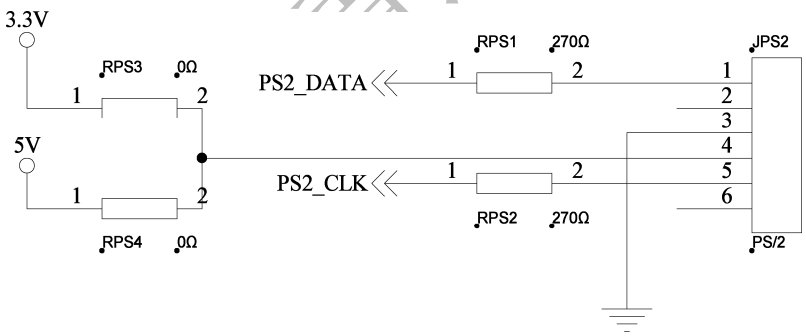


图 2.19　PS/2 参考电路

2.2.9　RS-232 串口

RS-232 接口定义如表 2.1 所示。

表 2.1　　RS-232 接口定义表

25 芯	9 芯	信号方向来自	缩　　写	描　　述
2	3	PC	TXD	发送数据
3	2	调制解调器	RXD	接收数据
4	7	PC	RTS	请求发送
5	8	调制解调器	CTS	允许发送

6	6	调制解调器	DSR	通信设备准备好
7	5		GND	信号地
8	1	调制解调器	CD	载波检测
20	4	PC	DTR	数据终端准备好
22	9	调制解调器	RI	响铃指示器

DTE DCE 设备信号电流方向如表 2.2 所示。

表 2.2　　DTE DCE 设备信号电流方向表

9 芯 DTE	25 芯 DTE	电 流 方 向	缩写 DCE	描述 DCE
3	2	DTE→DCE	2	3
2	3	DTE←DCE	3	2
7	4	DTE→DCE	4	7
8	5	DTE←DCE	5	8
6	6	DTE←DCE	6	6
5	7	DTE←DCE	7	5
1	8	DTE←DCE	8	1
4	20	DTE→DCE	20	4
9	22	DTE←DCE	22	9

RS-232 参考电路如图 2.20 所示。

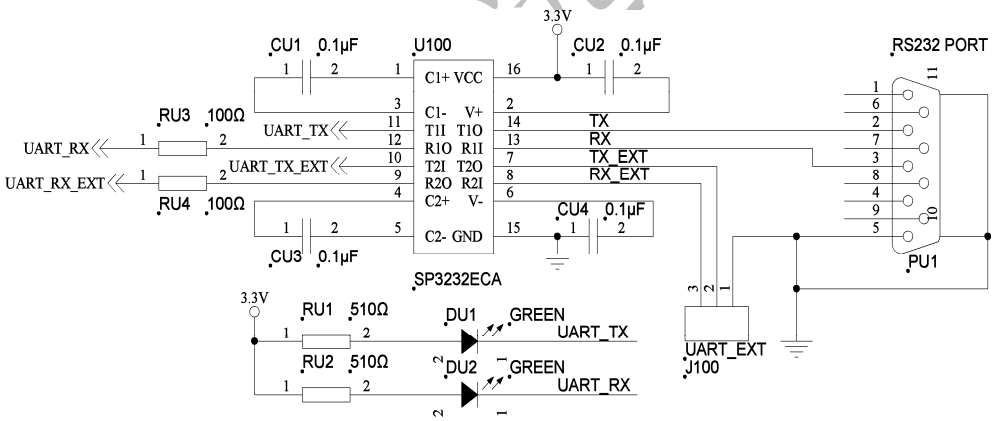


图 2.20　RS-232 参考电路

2.2.10　字符型液晶显示器接口电路设计

字符型液晶显示器电路原理图如图 2.21 所示。

第 1 脚：VSS 为地电源。

第 2 脚：VDD 接 5V 正电源。

第 3 脚：V0 为液晶显示器对比度调整端，接正电源时对比度最弱，接地电源时对比度最高，对比度过高时会产生“鬼影”，使用时可以通过一个 10kΩ的电位器调整对比度。

第 4 脚：RS 为寄存器选择线，高电平时选择数据寄存器低电平时选择指令寄存器。

第 5 脚：RW 为读写信号线，高电平时进行读操作，低电平时进行写操作。当 RS 和 RW 共同为低电平时可以写入指令或者显示地址，当 RS 为低电平、RW 为高电平时可以读忙信号，当 RS 为高电平、RW 为低电平时可以写入数据。

第 6 脚：E 端为使能端，当 E 端由高电平跳变成低电平时，液晶模块执行命令。

第 7～14 脚：D0～D7 为 8 位双向数据线。

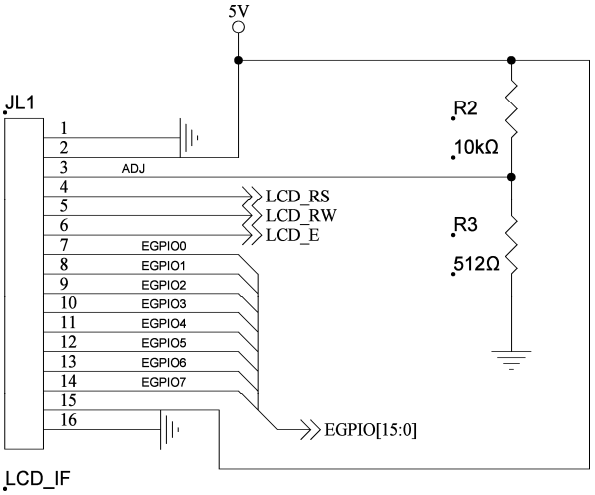


图 2.21　字符型液晶显示器电路原理图

2.2.11　USB 2.0 接口芯片 CY7C68013 电路设计

通过开发系统上的 USB 接口，可以用 USB 电缆直接将系统与主机（比如 PC 机）相连。USB 接口使用了 Cypress 公司的 CY7C68013 芯片来实现，其电路原理图如图 2.22 所示。

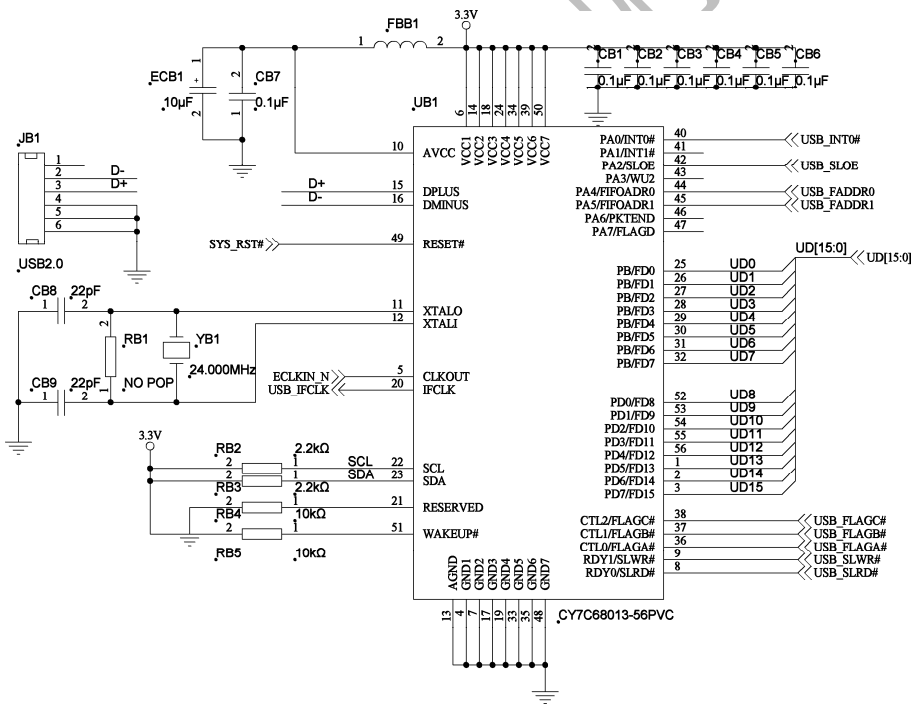


图 2.22　CY7C68013 原理图

1．芯片介绍

Cypress Semiconductor 公司的 EZ-USB FX2 是世界上第一款集成 USB 2.0 的微处理器。它集成了 USB 2.0 收发器、SIE（智能串行引擎）、增强的 8051 微控制器和可编程的外围接口。FX2 这种独创性结构可使数据传输率达到 56MB/s，即 USB 2.0 允许的最大带宽。

在 FX2 中，智能 SIE 可以硬件处理许多 USB 1.1 和 USB 2.0 协议，从而减少了开发时间和确保了 USB 的兼容性。GPIF（General Programmable Interface）和主/从端点 FIFO（8 位或 16 位数据总线）为 ATA、UTOPIA、EPP、PCMCIA 和 DSP 等提供了简单和无缝连接接口。

CY7C68013 集成了以下特性。

（1）USB 2.0 收发器、SIE（智能串行引擎）和增强性 8051 微处理器。

（2）软件运行。8051 程序从内部 RAM 开始运行，可借助下列几种方式进行程序装载。

- 通过 USB 下载。
- 从 EEPROM 中装载。
- 通过外部存储器设备。

（3）4 个可编程（BULK/INTERRUPT/ISOCHRONOUS）端点，可选双缓冲、三缓冲和四缓冲。

（4）8 位或 16 位外部数据接口。

（5）通用可编程接口（GPIF）。

- 可以直接连接到并口，分为 8 位和 16 位。
- 可编程波形描述符和配置寄存器。
- 支持多个 Ready 输入和 Control 输出。

（6）集成标准 8051 内核，且具有下列增强特性。

- 可以达到 48MHz 时钟。
- 每条指令占 4 个时钟周期。
- 2 个 UARTS。
- 3 个定时/计数器。
- 扩展的中断系统。
- 两个数据指针。

（7）采用 3.3V 电源系统。

（8）矢量 USB 中断。

（9）独立的数据缓冲区供 SETUP 和 DATA 包控制传输。

（10）集成 I^2C 控制器，运行速度可达 100kHz。

（11）4 个 FIFO，可与 ASIC 和 DSP 等无缝连接。

（12）专门的 FIFO 和 GPIF 自动矢量中断。

（13）可用于 DSL Modems、ATA 接口、相机、Home PNA、WLAN、MP3 播放器、网络等。

2．USB 启动方式和枚举

上电时，内部逻辑会检查连接到 I^2C 总线上的 EEPROM 中的第一个字节（0xC0 或 0xC2）。如果是 0xC0，就会使用 EEPROM 中的 VID/PID/DID 来替代内部存储值；如果是 0xC2，内部逻辑就会把 EEPROM 中的内容装入到内部 RAM 中；如果没有检查到 EEPROM，FX2 就会使用内部存储的描述符来枚举。其缺省值是 0x04B4/ 0x8613/ 0xxxyy。

当首次插入 USB 时，FX2 会通过 USB 电缆自动枚举并下载固件和 USB 描述符表。然后 FX2 将再次枚举，通过下载的信息来定义设备。这两个步骤就叫做重枚举，当设备插入时它们就立即执行。

3．程序/数据存储器

（1）内部数据 RAM。

FX2 的内部数据 RAM 被分成 3 个不同的区域：低（LOW）128 字节，高（Upper）128 字节和特殊功能寄存器（SFR）空间。低 128 字节和高 128 字节是通用 RAM，SFR 包括 FX2 控制和状态寄存器。

（2）外部程序存储器和数据存储器。

FX2 有 8KB 片上 RAM（位于 0x0000～0x1FFF 范围内）和 512 字节 Scratch RAM（位于 0xE000～0xE1FF）。尽管 Scratch RAM 从物理上来说位于片内，但是通过固件可以把它作为外部 RAM 一样来寻址。FX2 保留 7.5KB（0xE200～0xFFFF）数据地址空间作为控制/状态寄存器和端点缓冲器。

注意 只有数据内存空间保留，而程序内存（0xE000～0xFFFF）并不保留。

4．端点缓冲区

FX2 包含 3 个 64 字节端点缓冲区和 4KB 可配置成不同方式的缓冲，其中 3 个 64 字节的缓冲区为 EP0、EP1IN 和 EP1OUT。

EP0 作为控制端点用，它是一个双向端点，既可为 IN 也可为 OUT。当需要控制传输数据时，FX2 固件读写 EP0 缓冲区，但是 8 个 SETUP 字节数据不会出现在这 64 字节 EP0 端点缓冲区中。

EP1IN 和 EP1OUT 使用独立的 64 字节缓冲区，可配置为 BULK、INTERRUPT 或 ISOCHRONOUS 传输方式，这两个端点和 EP0 一样只能被固件访问。这一点与大端点缓冲区 EP2、EP4、EP6 和 EP8 不同，这 4 个端点缓冲区主要用来和片上或片外进行高带宽数据传输而无需固件的参与。EP2、EP4、EP6 和 EP8 是高带宽、大缓冲区，它们可被设置成不同的方式来适应带宽的需求。

5．外部 FIFO 接口

EP2、EP4、EP6 和 EP8 大端点缓冲区主要用来进行高速（480Mbit/s）数据传输。可以通过 FIFO 数据接口与外部 ASIC 和 DSP 等处理器无缝连接来实现高速数据传输。它具有的通用接口有：Slave FIFO 或 GPIF（内部主）、同步或异步时钟、内部或外部时钟等。

6．中断资源

FX2 的中断结构是在一个标准 8051 单片机的基础上增强和扩展了部分中断资源，中断资源如表 2.3 所示。

表 2.3 FX 中断资源表

FX2 中断	中 断 来 源	中 断 向 量	优 先 级
IE0	INT0 Pin	0x0003	1
TF0	Timer0 Overflow	0x000B	2
IE1	INT1 Pin	0x0013	3
TF1	Timer1 Overflow	0x001B	4
RI_0 & TI_0	USART0 Rx & Tx	0x0023	5
TF2	Timer2 Overflow	0x002B	6
Resume	WAKEUP/WU2 Pin	0x0033	0
RI_1 & TI_1	USART1 Rx & Tx	0x003B	7
USBINT	USB	0x0043	8
I2CINT	I^2C BUS	0x004B	9
IE4	GPIF/FIFOs/INT4 Pin	0x0053	10
IE5	INT5 Pin	0x005B	11
IE6	INT6 Pin	0x0063	12

其中 27 个 USB 请求共享 USB 中断，14 个 FIFO/GPIF 源共享 INT4。

芯片的详细介绍与使用方法可参看芯片数据手册。

2.2.12 电源电路设计

电源是整个系统能够正常工作的基本保证，如果电源电路设计的不好，系统有可能不能工作，或者即使能工作但是散热条件不好，导致系统不稳定等异常情况。所以如何选用合适的电源芯片，以及如何合理地对电源进行布局布线，都是值得下大功夫研究的。

在选用电源之前要仔细阅读 FPGA 的芯片手册，一般来说 FPGA 用到的管脚和资源越多，那么所需要的电流就越大，当电路启动时 FPGA 的瞬间电流也比较大。通过数据手册中提供的电气参数，确定 FPGA 最大需要多大的电流才能工作。

下面是几种常使用的 FPGA 参考电源。

- AS117 可以提供 1A 电流，线型电源（适用 144 管脚以下、5 万逻辑门以下的 FPGA）。
- AS2830(或 LT1085/6)可以提供 3A 电流，线性电源(适用 240 管脚以下、30 万逻辑门以下的 FPGA)。
- TPS54350 可以提供 3A 电流，开关电源（适用大封装大规模的高端 FPGA）。

AS2830 电源应用电路如图 2.23 所示。

对于线性电源芯片，输出电压和输入电压的关系为：Vout= (1+RP3/RP2)×Vref。

Vref 一般是 1.25V，输出假定输入 Vin 为 5V，Vout 为 1.5V，那么 RP2/RP3=1/5，而 RP3 一般要求 100～150Ω，那么可以选 RP3=100Ω，RP2=500Ω。如果采用了固定电平输出的芯片，只需要把 RP3 焊 0Ω，RP2 不焊接即可。

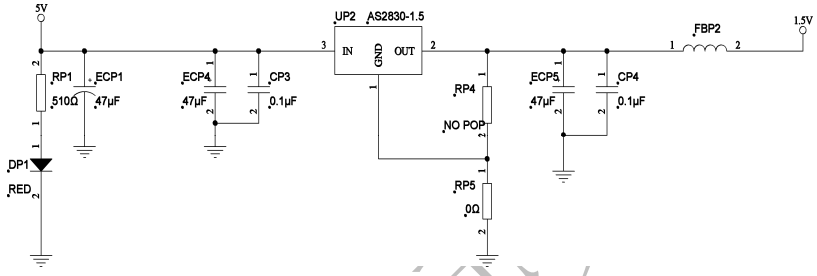


图 2.23　AS2830 电源应用电路

2.2.13　复位电路设计

一般复位电路采用的是低电平复位，只有个别单片机采用高电平复位方式。

常见的电平复位电路分为芯片复位和阻容复位。前者的复位信号比较稳定，而后者容易出现抖动。因此在成本允许的范围内我们一般推荐使用芯片复位。

常用的芯片复位有 MAX708S/706S 系列，它们可提供高、低电平两种复位方式和电源监控能力（监控电源电压低到一定程度自动复位）。

IMP811 是一款比较低廉的复位芯片，只有低电平复位功能，但是其体积非常小。

阻容复位典型连接电路如图 2.24 所示。

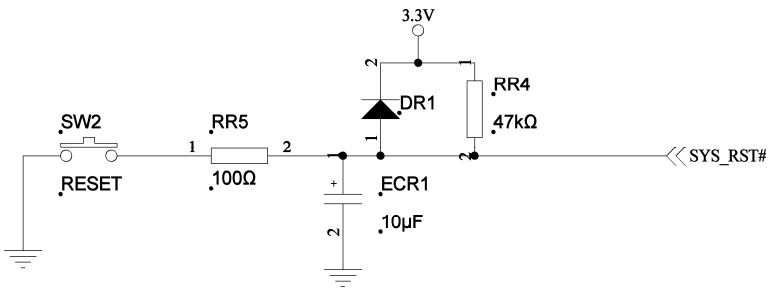


图 2.24　阻容复位典型连接电路

MAX708S 典型连接电路如图 2.25 所示。

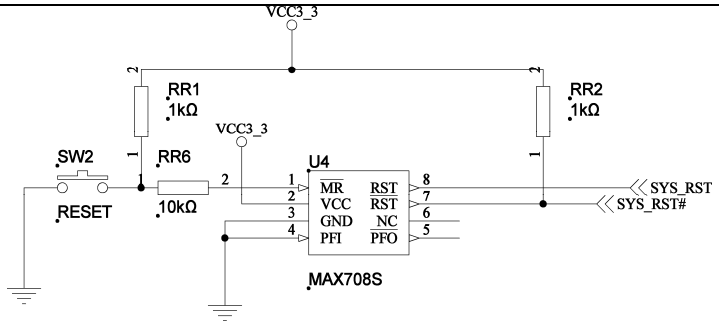


图 2.25　MAX708S 典型连接电路

IMP811 典型连接电路如图 2.26 所示。

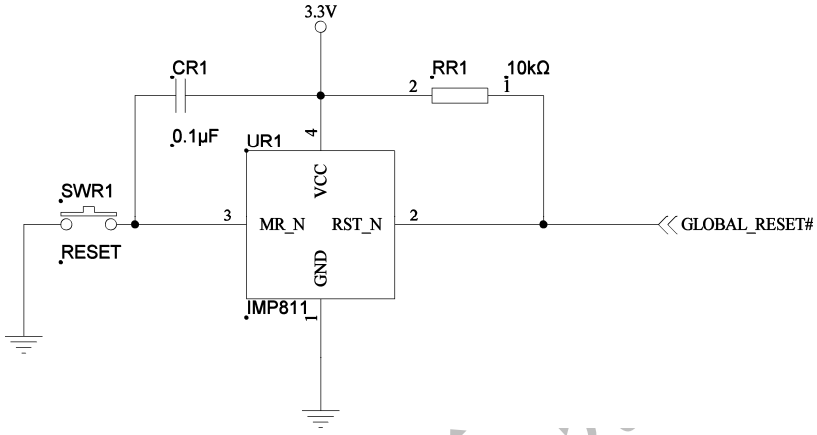


图 2.26　IMP811 典型连接电路

2.2.14　时钟电路设计

时钟电路典型连接如图 2.27 所示。

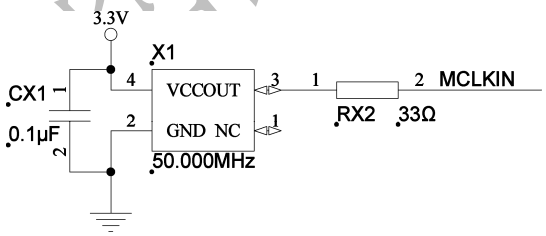


图 2.27　时钟电路典型连接

2.3　FPGA 硬件系统的设计技巧

FPGA 的硬件设计不同于 DSP 和 ARM 系统，比较灵活和自由。只要设计好专用管脚的电路，通用 I/O 的连接可以自己定义。因此，FPGA 的电路设计中会有一些特殊的技巧可以参考。

2.3.1　管脚兼容性设计

前面的内容提到过，FPGA 在芯片选项的时候要尽量选择兼容性好的封装。那么，在硬件电路设计时，就要考虑如何兼容多种芯片的问题。

例如，红色飓风 II 代－Altera 的开发板就是兼容了 EP1C6Q240 和 EP1C12Q240 两个型号的 FPGA。这两个芯片有 12 个 I/O 管脚定义是不同的。在 EP1C6Q240 芯片上，这 12 个 I/O 是通用 I/O 管脚，而在 EP1C12Q240 芯片上，它们是电源和地信号。

为了能保证两个芯片在相同的电路板上都能工作，我们就必须按照 EP1C12Q240 的要求来把对应管脚连接到电源和地平面。因为，通用的 I/O 可以连接到电源或者地信号，但是电源或者地信号却不能作为通用 I/O。在相同封装、兼容多个型号 FPGA 的设计中，一般的原则就是按照通用 I/O 数量少的芯片来设计电路。

2.3.2 根据电路布局来分配管脚功能

FPGA 的通用 I/O 功能定义可以根据需要来指定。在电路图设计的流程中，如果能够根据 PCB 的布局来对应的调整原理图中 FPGA 的管脚定义，就可以让后期的布线工作更顺利。

例如，如图 2.1 所示，SDRAM 芯片在 FPGA 的左侧。在 FPGA 的管脚分配的时候，应该把与 SDRAM 相关的信号安排在 FPGA 的左侧管脚上。这样，可以保证 SDRAM 信号的布线距离最短，实现最佳的信号完整性。

2.3.3 预留测试点

目前 FPGA 提供的 I/O 数量越来越多，除了能够满足设计需要的 I/O 外，还有一些剩余 I/O 没有定义。这些 I/O 可以作为预留的测试点来使用。

例如，在测试与 FPGA 相连的 SDRAM 工作时序状态的时候，直接用示波器测量 SDRAM 相关管脚会很困难。而且 SDRAM 工作频率较高，直接测量会引入额外的阻抗，影响 SDRAM 的正常工作。

如果 FPGA 有预留的测试点，那么可以将要测试的信号从 FPGA 内部指定到这些预留的测试点上。这样既能测试到这些信号的波形，又不会影响 SDRAM 的工作。

如果电路测试过程中发现需要飞线才能解决问题，那么这些预留的测试点还可以作为飞线的过渡点。

2.4 FPGA 硬件系统的调试方法

随着 FPGA 芯片的密度和性能不断提高，调试的复杂程度也越来越高。BGA 封装的大量使用更增加了板子调试的难度。所以在调试 FPGA 电路时要遵循一定的原则和技巧，才能减少调试时间，避免误操作损坏电路。

一般情况下，可以参考以下步骤进行 FPGA 硬件系统的调试。

（1）首先在焊接硬件电路时，只焊接电源部分。使用万用表进行测试，排除电源短路等情况后，上电测量电压是否正确。

（2）然后焊接 FPGA 及相关的下载电路。再次测量电源地之间是否有短路现象，上电测试电压是否正确，然后将手排除静电后触摸 FPGA 有无发烫现象。

如果此时出现短路，一般是去耦电容短路造成，所以在焊接时一般先不焊去耦电容。FPGA 的管脚粘连也可能造成短路，这时需要对比电路图和焊接仔细查找有无管脚粘连。

如果出现电压值错误，一般是电源芯片的外围调压电阻焊错，或者电源的承载力不够造成的。若是后者，则需要选用负载能力更强的电源模块进行替换。如果 FPGA 的 I/O 管脚与电源管脚粘连，也可能出现电压值错误的现象。

如果出现 FPGA 发烫，一般是出现总线冲突的现象。这种情况下需要仔细检查外围总线是否出现竞争问题。特别是多片存储器共用总线时候，比如 ASRAM 和 Flash 芯片复用一套总线，如果片选信号同时有效就出现总线的冲突。

（3）以上步骤均通过后，将电路板上电运行。然后把下载电缆接到 JTAG 接口上，在主机中运行 Quartus II 软件，并打开 Programmer 编程器，单击其中的“Auto Detect”按钮进行 FPGA 下载链路自动检测。若能正确检测到 FPGA，说明配置电路是正确连接的。

自动检测 FPGA 下载链路如图 2.28 所示。

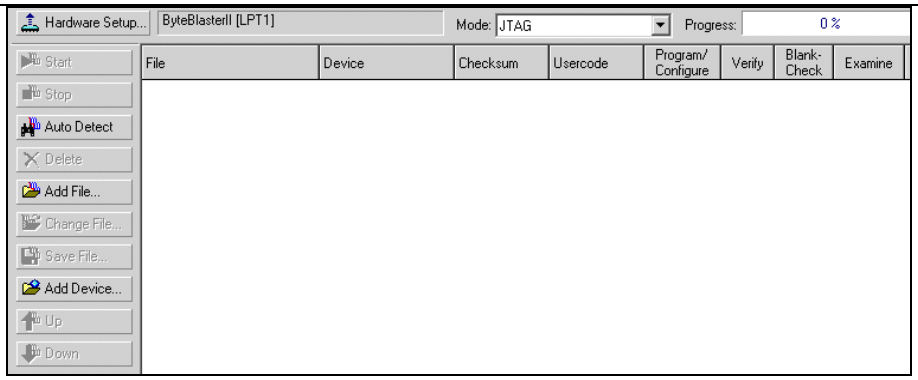


图 2.28　自动检测 FPGA 下载链路

（4）焊接时钟电路、复位电路及数码管电路，并向 FPGA 下载一个数码管跑马灯程序。若程序能够正确运行，说明 FPGA 已经可以正常工作了。

（5）最后焊接所有其他电路，并进行整体功能测试。

2.5　典型实例 1：在 Altera 的 FPGA 开发板上运行第一个 FPGA 程序

2.5.1　实例的内容及目标

1．实例的主要内容

本节旨在通过给定的工程实例——“蜂鸣器播放梁祝音乐”来熟悉 Altera Quartus II 软件的基本操作、设计、编译及仿真流程。同时使用基于 Altera FPGA 的开发板将该实例进行下载验证，完成工程设计的硬件实现，熟悉 Altera FPGA 开发板的使用及配置方式。

在本节中，将主要讲解下面一些知识点。

- Quartus II 工程创建及属性设置。
- Quartus II 源文件设计输入方式。
- Quartus II 约束设计。
- Quartus II 工程编译。
- Quartus II 功能仿真。
- Quartus II 时序仿真。
- Quartus II 硬件下载。

通过这些知识点，按照下面提供的训练流程，读者可以迅速地掌握使用 Quartus II 软件进行 FPGA 开发的方法。

2．实例目标

通过详细的流程讲解，读者应达到下面的目标。

- 熟悉 Quartus II 软件的操作环境。
- 熟悉 Quartus II 软件开发 FPGA 的基本流程。
- 可独立使用 Quartus II 软件开发新工程。

2.5.2　平台简介

本实例基于红色飓风 II 代 Altera 板，在此开发板上集成了 Altera 的 Cyclone 一代 FPGA 及相关的丰富外设资源。

此开发板更加详细的信息、扩展附件及使用方法，可以到红色飓风的官方网站：http://www.fpgadev.com 获取更多的信息。

本实例将在这个开发板上对 Altera 的 FPGA 设计流程做一个全面的介绍，依照此例程的流程便可在该开发板上运行一个 FPGA 程序。

如图 2.29 所示为此开发板的外观图。

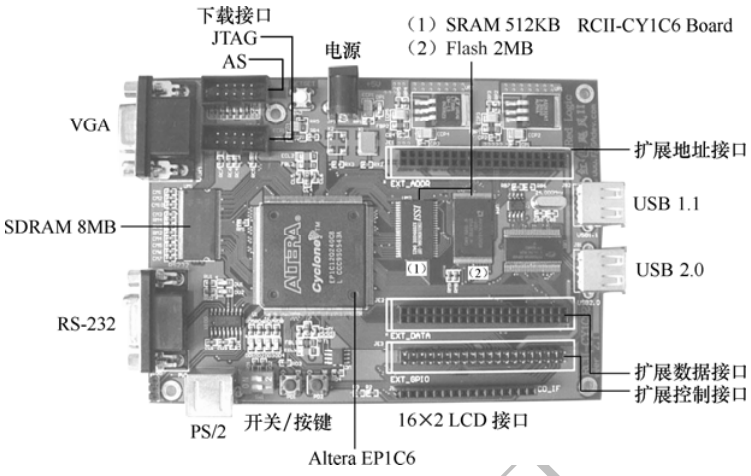


图 2.29　红色飓风 II 代 Altera 板外观图

2.5.3　实例详解

本节将使用图解的方式将整个流程一步一步展现给读者，使读者能够轻松掌握 Quartus II 的开发流程。

1．工程创建及属性设置

（1）启动 Quartus II 软件。

安装 Quartus II 软件后，在桌面或者程序组中启动 Quartus II 软件。

（2）打开新工程向导。

启动软件后，选择“File”菜单的“New Project Wizard”选项，打开新建工程向导，如图 2.30 所示。在新建工程向导的 “介绍”页面中，单击“Next”按钮进入下一页。

（3）设置工程属性。

如图 2.31 所示，在新建工程向导的第一页对工程工作目录、工程名称以及顶层模块名称进行设置。

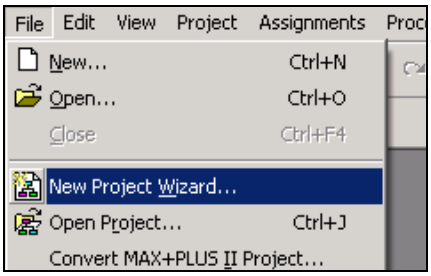


图 2.30　新建工程

图 2.31　设置工程属性

（4）添加设计文件。

在新建工程向导的第二页选择为工程添加设计文件，如图 2.32 所示。

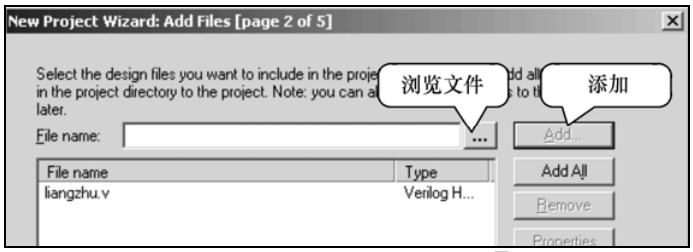
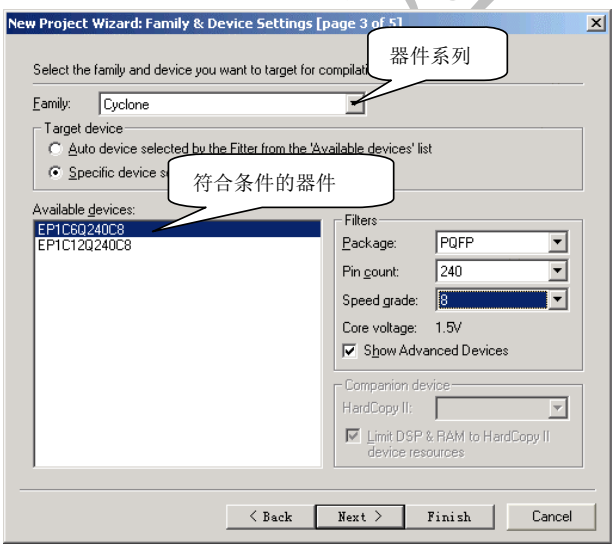


图 2.32　添加设计文件

（5）选择 FPGA 器件。

在新建工程向导的第三页，为工程配置相应的器件型号和参数，如图 2.33 所示。选取的器件型号将在完全编译时将工程设计映射到对应的器件逻辑资源上。



封装
管脚数
速度等级

图 2.33　选择器件型号

（6）完成工程创建。

配置完器件属性后，选择 Quartus II 默认的综合工具、仿真工具及时序分析工具，完成工程的创建。

选择软件左侧的工程浏览器的  按钮来管理已添加的文件。如图 2.34 所示，左边为工程层次窗口，右边为设计文件窗口。

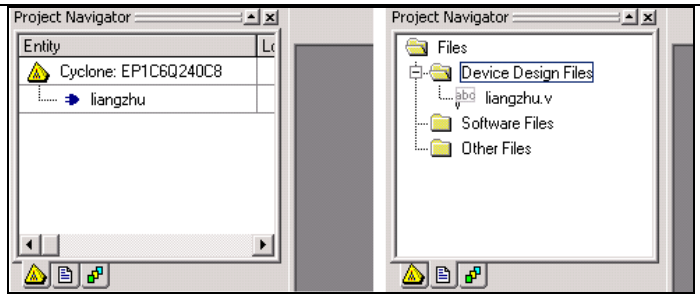


图 2.34　工程结构窗口

2．设计输入

（1）添加设计文件。

如果在创建工程时没有为工程添加设计文件，可以选择“Project”菜单的“Add/Remove Files in Project”选项，为工程添加设计文件。在本实例中，读者可向工程添加实例代码中的 liangzhu.v 文件。

添加后，在工程浏览器中双击 liangzhu.v 图标，即可查看该 Verilog 设计文件，如图 2.35 所示。

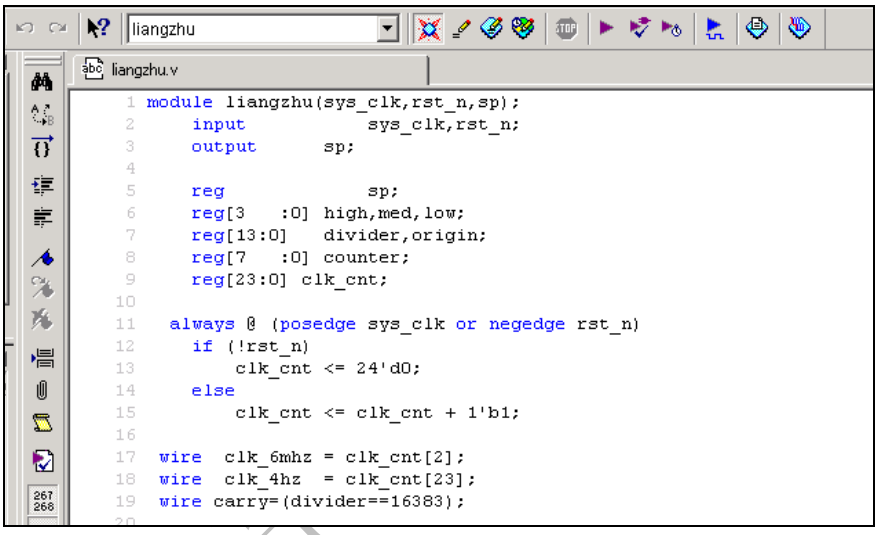


图 2.35　Verilog 设计文件

（2）配置器件属性。

同样的，如果在创建工程时没有为工程配置器件型号及属性，可以在工程浏览器的工程实体图标上单击右键，选择“Device”选项，为工程配置器件属性，如图 2.36 所示。

若在建立工程时已经配置好，则可单击“Device & Pin Options…”按钮，进一步设置器件的相关属性，如图 2.37 所示。

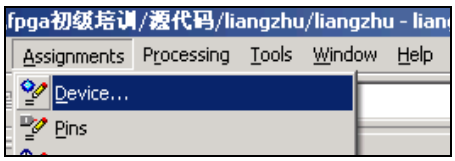


图 2.36　器件选择

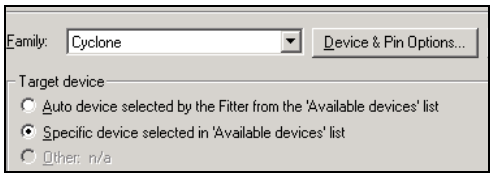


图 2.37　器件属性配置

在如图 2.38 所示的对话框中，可对 FPGA 所使用的配置芯片及未用管脚等进行配置。

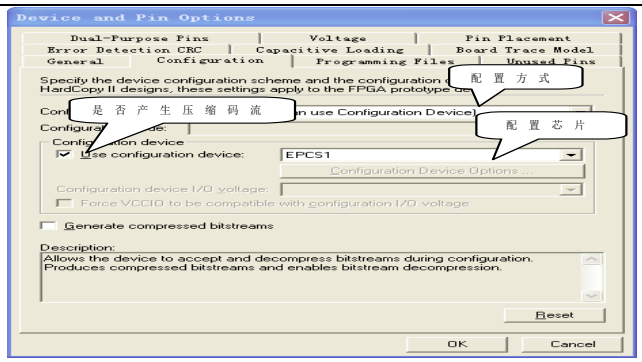


图 2.38　指定配置模式

选择“Unused Pins”选项卡，将不使用的管脚配置为三态，如图 2.39 所示。

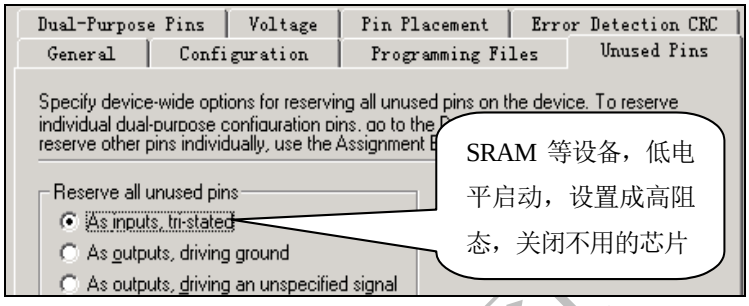


图 2.39　未用管脚置三态

（3）分析工程。

编写设计输入文件后，首先要进行工程分析，目的是为了检查设计输入的语法。单击“Start Analysis & Synthesis”按钮分析工程，如图 2.40 所示。

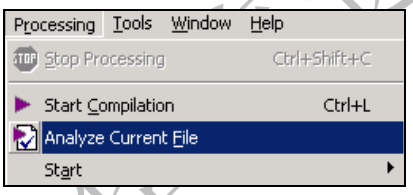


图 2.40　分析工程选项

查看信息栏，修改所有出现的错误，直到分析通过，如图 2.41 所示。

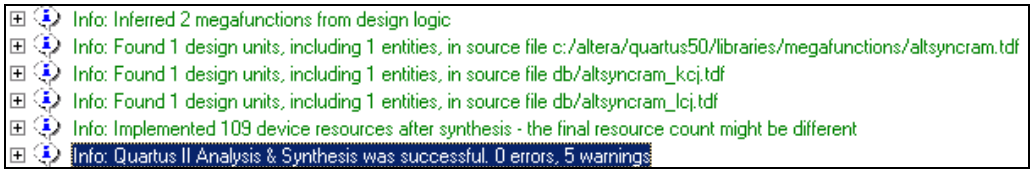


图 2.41　分析工程结果，没有报错

3．约束设计

（1）管脚分配。

管脚分配的作用在于将设计输入文件的端口与实际的器件进行映射，实现设计输入模块端口在实际器件管脚上的实例化。在 Quartus II 软件中可以在管脚分配主窗口中对管脚的分配进行设置，如图 2.42 所示。

	To	Location	I/O Bank	I/O Standard	General Function
1	rst_n	PIN_131	3	LVTTL	Row I/O
2	sp	PIN_62	4	LVTTL	Column I/O
3	sys_clk	PIN_153	3	LVTTL	Dedicated Clock
4	<<new>>	<<new>>			

图 2.42　管脚分配

通过工具栏中的快捷按钮可以帮助用户快速的进行管脚分配，如图 2.43 所示。

通过管脚分配主窗口中的管脚信息可以看到管脚的 I/O 种类、管脚的序号、Bank 所在的位置、I/O 使用的电压标准等，如图 2.44 所示。

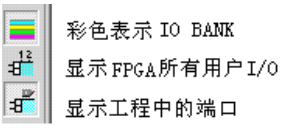


图 2.43　指定管脚快捷按钮

1	rst_n	PIN_131
2	sp	PIN_62
3	sys_clk	PIN_153

图 2.44　管脚信息

（2）其他约束。

除了对工程的管脚进行约束外，Quartus II 软件还允许用户对其他的一些约束进行设置，例如面积约束、速度约束、时钟约束、资源约束等。这些约束都属于较为严格的工程设置，在本实战训练中无需考虑。

4．编译工程

（1）完全编译。

有了完整的设计输入、完整的约束条件后，就可以对工程进行完全编译了，如图 2.45 所示是完全编译的选项。

根据工程复杂度的不同，Quartus II 在进行完全编译时所消耗的时间差异也很大。在工程浏览器中显示了编译的类别及进度，帮助用户了解编译的进程。

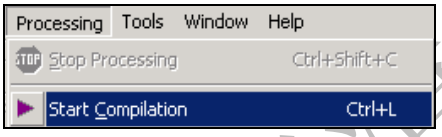


图 2.45　完全编译选项

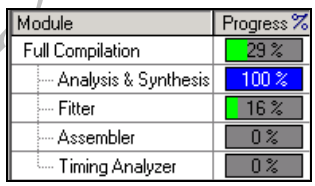


图 2.46　编译状态

若在编译过程中出现错误提示，用户可在信息栏中查看错误的信息，修改所有存在的错误后重新进行编译，直到能够无错误地完成编译，如图 2.47 所示。

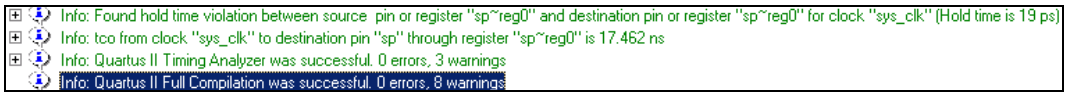


图 2.47　编译结果信息栏

（2）编译报告。

Quartus II 编译结束后，为用户提供了一个完整而详细的编译报告。通过该报告，用户可以查看工程使用资源的情况及系统可以达到的性能，如图 2.48 所示。

5．功能仿真

（1）建立仿真文件。

选择 Quartus II 软件“File”菜单的“New”选项，打开新建其他文件对话框，选择新建波形图文件，如图 2.49 所示。

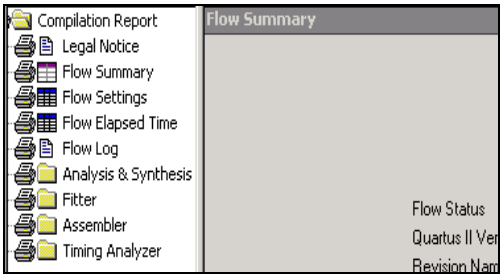


图 2.48　编译结果报告

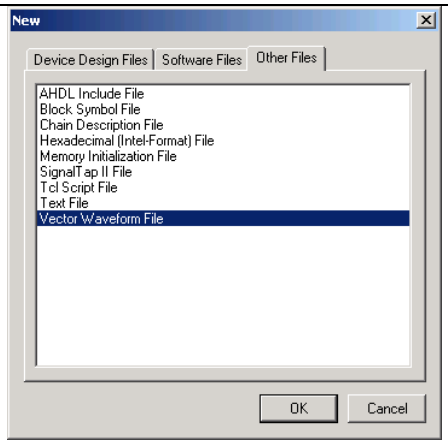


图 2.49　新建波形图

（2）添加观察信号。

用户可以通过双击波形窗口中的空白区域为工程添加需要观察的信号，如图 2.50 所示。

图 2.50　仿真波形窗口

在打开的插入节点或总线对话框中，打开“Node Finder…”（节点查找器），如图 2.51 所示。通过节点查找器，用户可以方便地选择需要观察的信号。

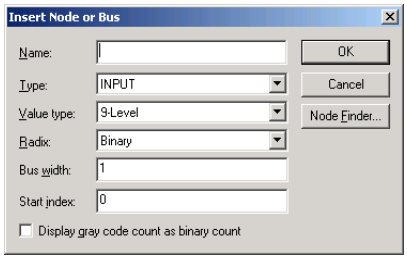


图 2.51　插入节点或总线对话框

选择 Filter：“Pins：all”，然后点击 list 列出所用输入/输出端口，如图 2.52 所示。

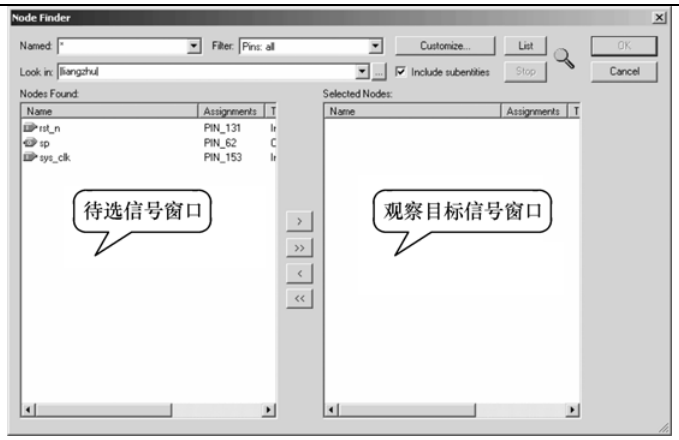


图 2.52　节点查找器

选中所有信号，单击 > 按钮，即可将选中信号加入观察目标窗口中，如图 2.53 所示。

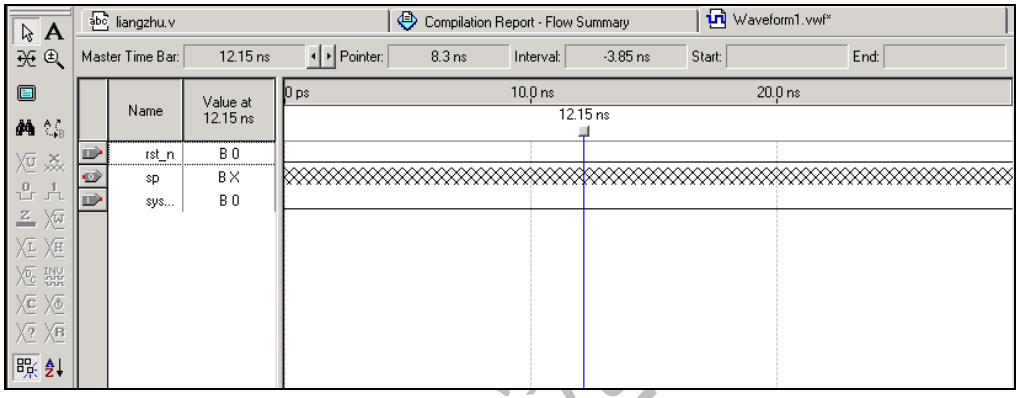


图 2.53　已添加好的信号

（3）设置仿真时间最小间隔。

考虑到硬件环境提供的是 50MHz 的时钟，即时钟周期为 20ns，设置“Grid Size”为 20ns。选择“Edit”菜单的“Grid Size”选项进行设置，如图 2.54 所示。

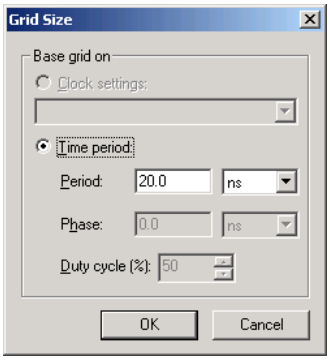


图 2.54　设置仿真时间间隔

（4）设置仿真时间长度。

选择“Edit”菜单的“End Time”选项，设置仿真时间长度为 10ms，如图 2.55 所示。

（5）添加激励信号。

选中 sys_clk 信号，单击时钟按钮，将该信号设置为时钟波形，周期为 20ns，如图 2.56 所示。此信号代表了系统时钟。

图 2.55　设置仿真结束时间

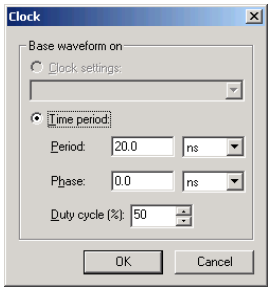


图 2.56　设置时钟

选中 rst_n 信号，单击高电平按钮，将该信号设置为 1。此信号代表了复位信号。

添加激励信号后，选择保存文件，此时的波形如图 2.57 所示。

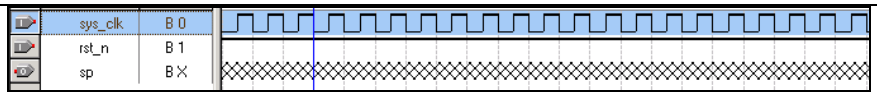


图 2.57　以设置好的输入波形

（6）生成功能仿真网表。

选择“Tools”菜单的“Simulator Tool”选项，打开仿真器，如图 2.58 所示。

将仿真器的仿真模式设置为“Functional”（功能仿真），如图 2.59 所示。

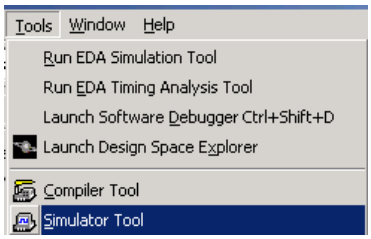


图 2.58　仿真器选项

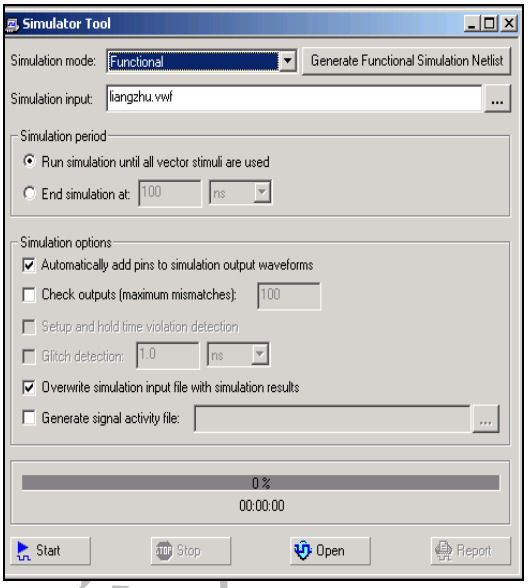


图 2.59　仿真对话框

单击“Generate Functional Simulation Netlist”按钮产生仿真网表。仿真网表是将工程设计文件进行编译及映射后生成的用于进行仿真的文件，仿真器根据仿真网表进行仿真，直接反应了工程设计文件的真实情况。

（7）开始功能仿真。

单击“Start”按钮开始进行功能仿真，如图 2.60 所示，可以看到仿真的进度，根据工程的复杂度，仿真过程所消耗的时间也有所不同。

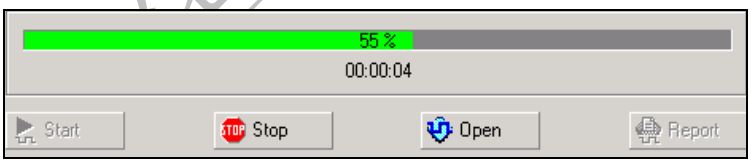


图 2.60　仿真进度

仿真完成后单击“open”按钮打开仿真结果，如图 2.61 所示。

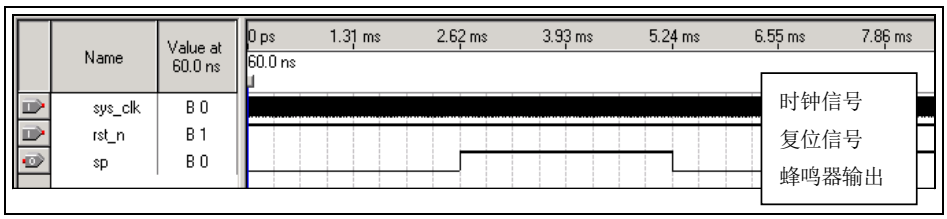


图 2.61　仿真波形结果

6．时序仿真

功能仿真后，如果波形没有问题，开始做时序仿真，检查波形延时对设计是否有影响。

选择“Tools”菜单的“Simulator Tool”选项，打开仿真器，选择仿真模式为“Timing”，即时序仿真模式，如图 2.62 所示。

图 2.62　选择时序仿真

单击“Start”按钮，开始时序仿真。时序仿真比功能仿真要慢一些。
仿真完成后，查看仿真结果。通过波形可以看到产生了 7.06ns 的延时，如图 2.63 所示。

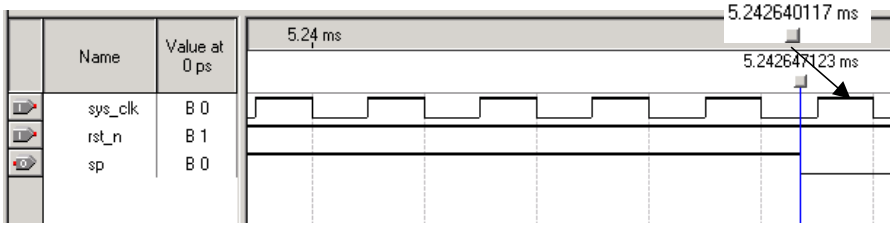


图 2.63　查看仿真结果

可以看到，功能仿真并不包含延迟，而时序仿真则会根据具体的器件参数配置及资源使用情况将延迟仿真出来。功能仿真主要用于验证工程设计文件逻辑的正确性，而时序仿真更能体现真实的硬件运行过程中设计文件的执行过程。

7．下载程序

仿真验证结束后，用户就可以将工程下载到实际的开发板上进行验证了。
选择“Tools”菜单的“Programmer”选项，打开下载器界面，如图 2.64 所示。

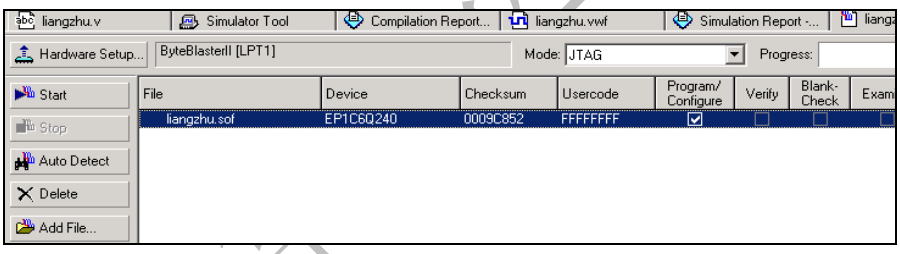


图 2.64　下载界面

首先单击“Hardware Setup”按钮，选择下载电缆，如 ByteBlasterII（LPT1），即打印机接口下载线，然后选择下载模式为 JTAG（在线调试）模式或者 AS（固化至 EPCS 配置芯片）模式，并选择对应的下载文件。连接好下载电缆后，选中“Program/Configure”即可下载。

8．结果检验

下载结束后，用户应该可以在开发板的蜂鸣器上听到演奏的梁祝乐曲。如果采用的是 JTAG 下载模式，那么将开发板断电后，重新上电，则乐曲不会再继续演奏；若使用的是 AS 下载模式，则乐曲还会再继续演奏。
若下载后没有听到乐曲，用户应检查几个主要容易出问题的步骤。例如，设计输入是否完整，管脚是否分配，下载电缆是否正确连接等。

2.5.4　小结

上述流程就是一个完整的 FPGA 设计流程。虽然此实例实现的功能比较简单，但对于初学者来说，是一个不错的入门实例。其主要目的是让初学者对 FPGA 的设计有一个初步的了解。通过该实例来熟悉 Quartus II

软件的使用，为今后的学习打下基础。
在实际的 FPGA 设计中，在每个阶段都会遇到很多问题。为了实现最终的目标，需要充分地利用 Quartus II 设计软件。这需要一个比较长时间的学习和积累，希望通过这个实例能让初学者更快地入门。

2.6 典型实例 2：在 Xilinx 的 FPGA 开发板上运行第一个 FPGA 程序

2.6.1 实例的内容及目标

1．实例的主要内容

本节旨在通过给定的工程实例——“按键开关控制 LED”来熟悉 Xilinx ISE 软件的基本操作、设计、编译及仿真流程。同时使用基于 Xilinx FPGA 的开发板将该实例进行下载、验证及调试，完成工程设计的硬件实现，熟悉 Xilinx FPGA 开发板的使用及配置方式。
在本训练中设计软件采用 ISE 7.1i，实现功能是利用 4 个按键开关来控制 8 个 LED 灯。具体的显示方案是由 4 个按键开关控制 8 个 LED 灯，根据按键开关按下的不同，会有不同的灯点亮。
在本训练中，将主要讲解下面一些知识点。

- Xilinx ISE 工程创建及属性设置。
- Xilinx ISE 源文件设计输入方式。
- Xilinx ISE 综合。
- Xilinx ISE 行为仿真。
- Xilinx ISE 约束设计。
- Xilinx ISE 布局布线。
- Xilinx ISE 时序仿真。
- Xilinx ISE 硬件下载。

通过这些知识点，按照下面提供的训练流程，读者可以迅速地掌握使用 Xilinx ISE 软件进行 FPGA 开发的方法。

2．实例目标

通过详细的流程讲解，读者应达到下面的目标。

- 熟悉 Xilinx ISE 软件的操作环境。
- 熟悉 Xilinx ISE 软件开发 FPGA 的基本流程。
- 可独立使用 Xilinx ISE 软件开发新工程。

2.6.2 平台简介

本实例基于红色飓风 II 代 Xilinx 板，此开发板集成了 Xilinx 的 Spartan3 40 万门 FPGA 及相关的丰富外设资源。
此开发板更加详细的信息、扩展附件及使用方法，可以到红色飓风的官方网站：http://www.fpgadev.com 获取更多的信息。
本实例将通过一个简单的实例在这个开发板上对 Xilinx 的 FPGA 设计流程做一个全面的介绍，依照此例程

的流程便可在该开发板上运行一个 FPGA 程序。
如图 2.65 所示为此开发板的一个外观图。

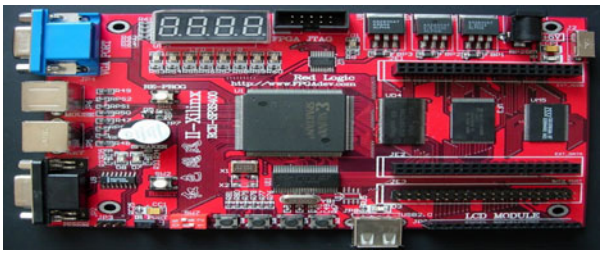


图 2.65　红色飓风 II 代 Xilinx 板外观图

2.6.3　实例详解

本节同样将使用图解的方式将整个流程一步一步展现给读者，使读者能够轻松掌握 ISE 的开发流程。

1．工程创建及属性设置

（1）启动 Project Navigator。
安装好 ISE7.1 套件后，可以按照以下的方法打开 Project Navigator。
① 在桌面上面双击 Xilinx ISE 7.1i 的快捷方式图标，如图 2.66 左图所示。
② 选择“开始”→“所有程序”→“Xilinx ISE 7.1i”→“Project Navigator”，打开 Project Navigator，如图 2.66 右图所示。
打开后的 Project Navigator 界面如图 2.67 所示。可以看到，Project Navigator 的界面是一个标准的 Windows 软件视窗。分别包括标题栏、菜单栏、工具栏、工程浏览器、进度浏览器、主工作区、信息栏、状态栏。

图 2.66　ISE 的启动

图 2.67　Project Navigator 主界面

（2）创建一个新的工程。

单击“File”→“New Project…”，弹出如图 2.68 所示对话框。

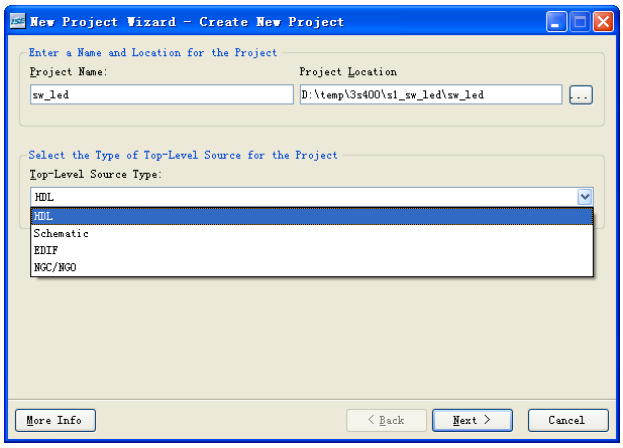


图 2.68　新建工程对话框

在这里填写将要创建的工程的名称（Project Name）、路径（Project Location）和工程的顶层模块类型（Top-Level Module Type）。

顶层模块类型我们主要使用前面两种：HDL(Hardware Design Language)硬件设计语言模式和 Schematic 原理图模式，这里选择 HDL。

（3）设置工程属性。

填写好后，单击“Next”按钮，在如图 2.69 所示的对话框中设置工程的属性。

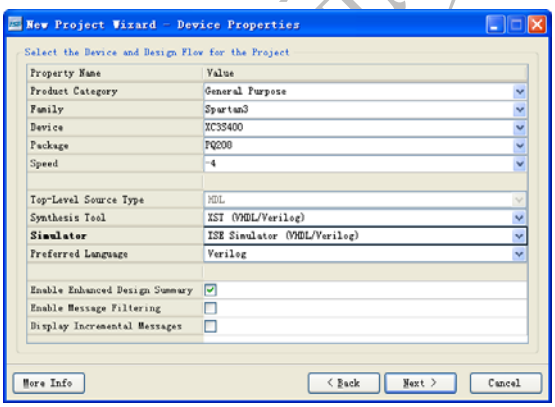


图 2.69　工程属性对话框

在这个对话框可设置的属性定义，如表 2.4 所示。

表 2.4　　属性定义

Device Family	使用的 FPGA 的种类
Device	使用的 FPGA 的型号
Package	使用的 FPGA 的封装
Speed Grade	使用的 FPGA 的速度级别
Top-Level Module Type	顶层模块类型
Synthesis Tool	综合工具
Simulator	仿真工具
Generated Simulation Language	仿真模块语言类型

本实例选用的都是 ISE 自己带的综合工具和仿真工具，这里也可以选择第三方的应用软件，如图 2.70 和图 2.71 所示。

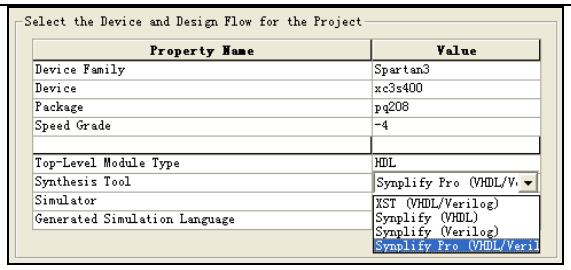


图 2.70　工程属性对话框选择第三方综合工具

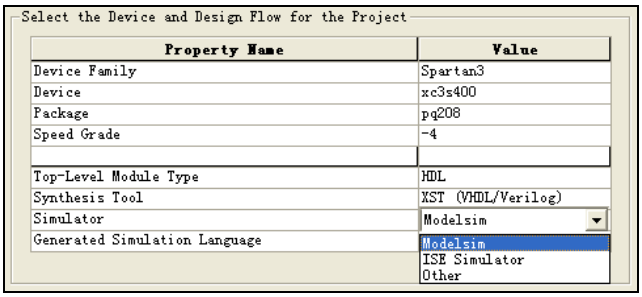


图 2.71　工程属性对话框选择第三方仿真工具

ModelSim 和 Synplify Pro 是比较通用的第三方仿真和综合软件。在这里如果我们选择了使用第三方的软件进行综合及仿真的话，在后面执行相应步骤的时候 ISE Navigator 就会自动寻找并打开相应的软件。

（4）添加设计文件。

填写好 FPGA 型号和使用的综合，仿真软件后，单击“Next”按钮打开如图 2.72 所示的创建源文件对话框。

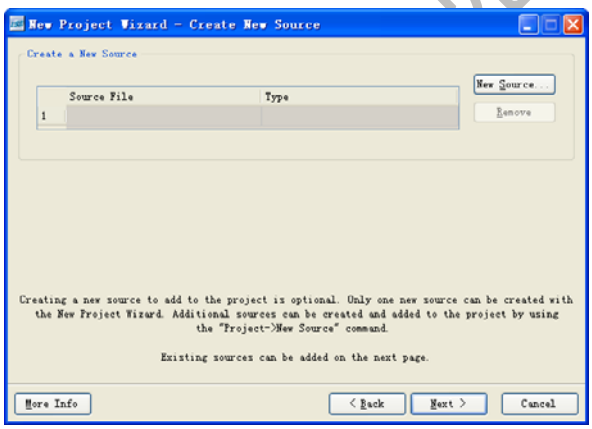


图 2.72　创建源文件对话框

在这个对话框里面，允许用户为即将建立的工程创建一个新的源文件，并且这里只可以创建一个源文件。其他的要在工程建立以后创建，也可以在创建工程以后再建立所有的源文件。直接单击“Next”按钮，进入如图 2.73 所示的添加源文件对话框。

在这个对话框里面为即将建立的工程添加已经存在的源文件。如果没有现成的源文件，直接单击“Next”按钮，完成新工程的创建。

（5）完成工程创建。

在如图 2.74 所示的新工程信息对话框中，列出了新建工程的相关参数及属性。

在这个对话框里面显示将要创建的工程的全部信息，确认无误后单击“Finish”按钮，Project Navigator 将自动创建一个名为 sw_led 的工程。如图 2.75 所示是新工程的界面。

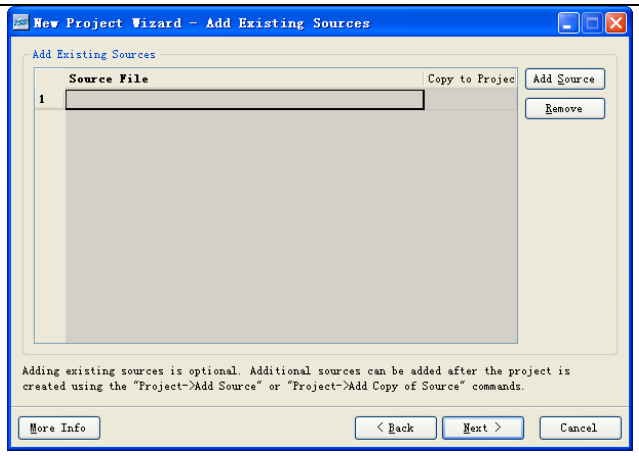


图 2.73　添加源文件对话框

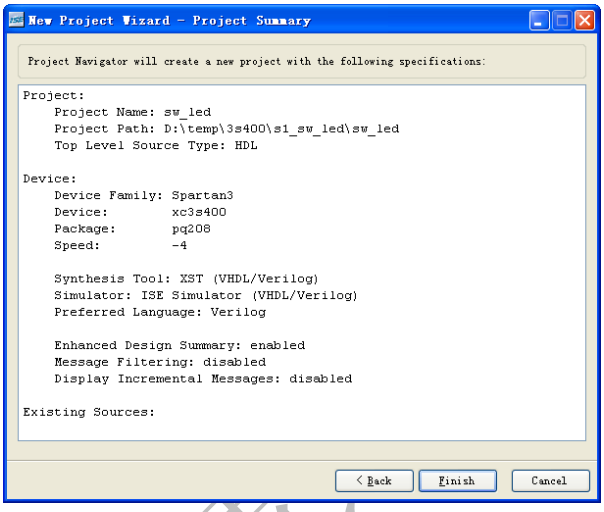


图 2.74　工程信息对话框

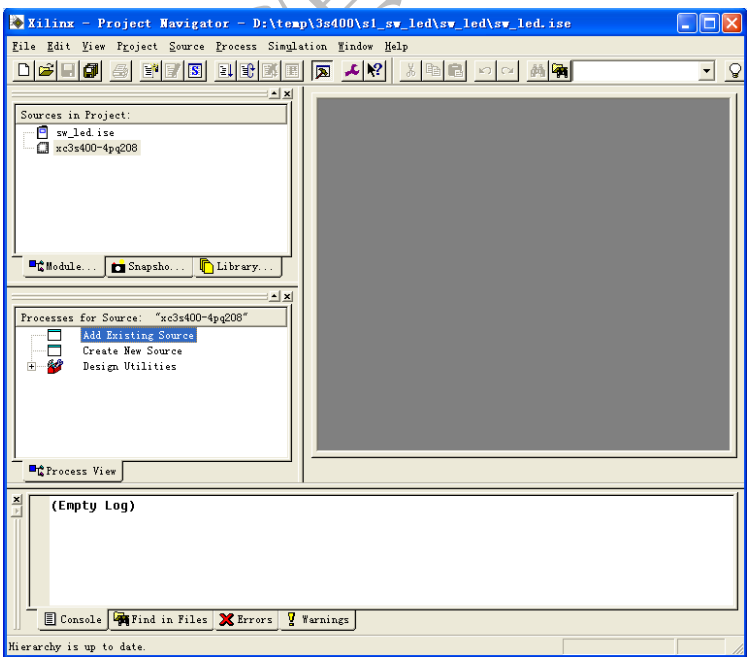


图 2.75　工程创建后的界面

2．设计输入

创建好工程以后就要为工程添加源文件了，具体方法如下。

为工程添加源文件有两种方式，可以双击“Process View”对话框里面的“Create New Source”，也可以在“Module View”对话框里面的“xc3s400-4pq208”图标上面单击鼠标右键，选择“New Source…”选项，如图 2.76 所示。

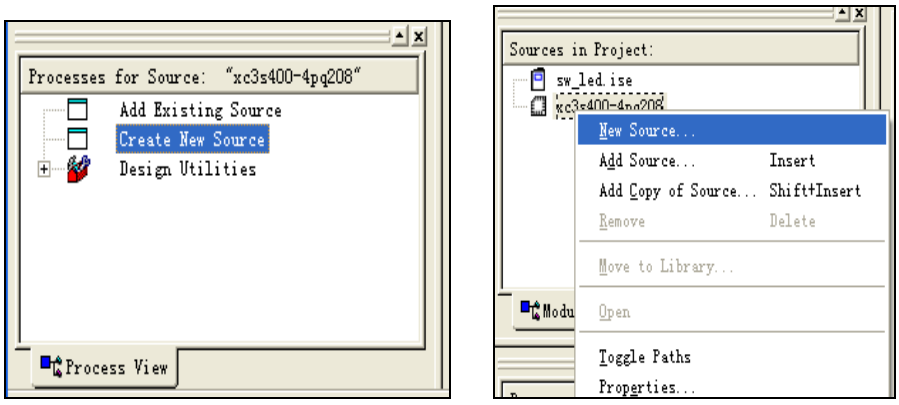


图 2.76　添加源文件的两种方式

选择“New Source”，弹出如图 2.77 所示的新建源文件对话框。

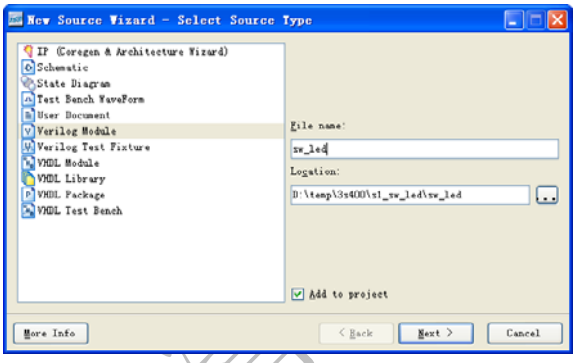


图 2.77　新建资源对话框

在右面的“File Name”栏里面填写要生成的源文件的名字，在“Location”栏填写源文件保存的路径，一般位于工程文件夹里面，没有特殊需要不必更改。另外一定要选择“Add to project”选项，然后在左边的一排图标里面选择源文件的类型后单击“Next”按钮，进入如图 2.78 所示的 Verilog 源定义对话框。

可以在上面的对话框里面输入源文件的模块名称和管脚定义，也可以先不输入，后面写程序的时候自己输入。单击“Next”按钮，完成源文件的创建，在如图 2.79 所示的对话框中列出了新建源文件的一些信息。确认信息无误后，单击“Finish”按钮，ISE 将生成名为 sw_led.v 的源文件，如图 2.80 所示。

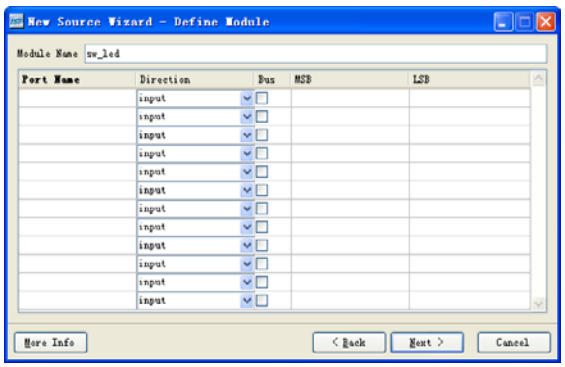


图 2.78　新建 Verilog 文件设置对话框

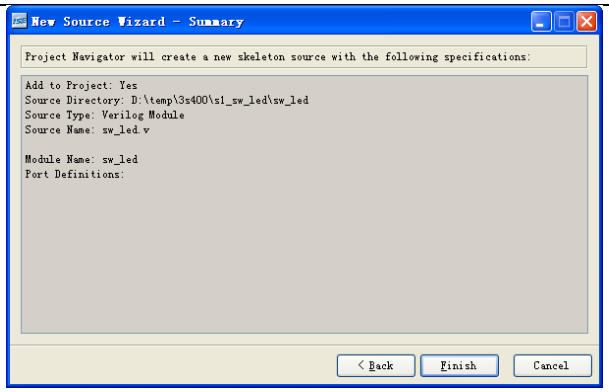


图 2.79　新建源文件信息对话框

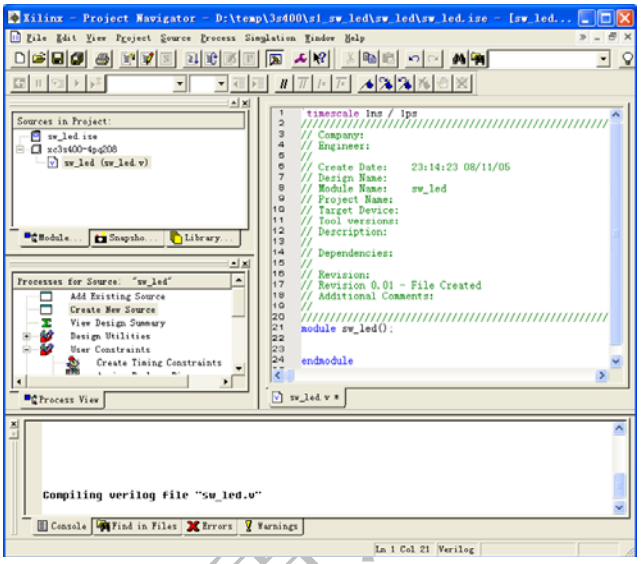


图 2.80　添加新资源后工程界面

用户可以在工作区中开始进行设计的输入，也可参考实例代码中的例程，将代码直接复制到新建的 Verilog 文件下。输入好程序以后，保存源文件，完成设计的输入。

3．综合及仿真

（1）综合。

在“Process View”对话框里面双击“Synthesize – XST”，开始进行工程的综合，如图 2.81 所示。

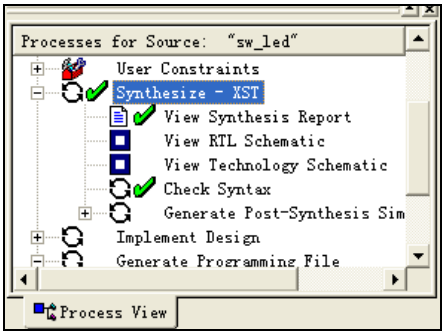


图 2.81　用 ISE 综合工具进行综合

综合主要检查源文件程序里面的语法错误（Check Syntax），双击“View Synthesis Report”可以观察综合的结果报告，如图 2.82 所示。

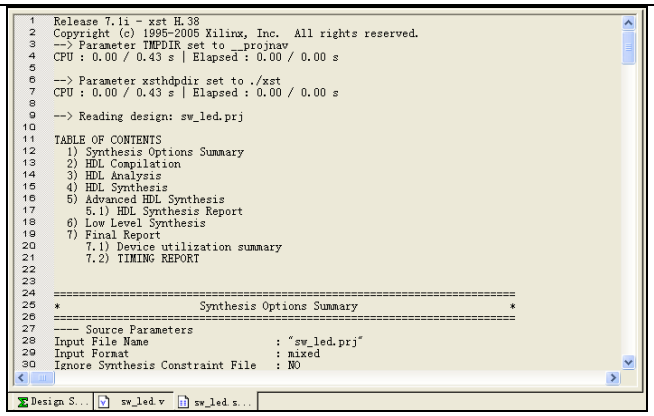


图 2.82　综合报告

如果在这步发现源程序的设计语言有语法毛病，就会弹出 Error 警告。这样就可以根据报错的位置，在源程序里面查找错误位置，修改好以后重新进行综合，直至无错误为止。

（2）综合后仿真。

这里的仿真是综合后的仿真，也可以称为功能仿真。仿真时并不考虑工程的约束条件及器件的资源使用状况，因此不会产生仿真延迟。这里使用的是 ISE 自带的仿真工具进行综合后仿真。下面是仿真的步骤。

① 产生仿真文件。

如图 2.76 的方式，在“New Source”对话框里面选择创建 Test Bench Wavefrom 文件，如图 2.83 所示。单击“Next”按钮，在如图 2.84 所示对话框中为 Test Bench 指定源文件。如图所示在源文件对话框中只有一个文件，这是因为我们只为工程添加了一个设计文件。如果我们为工程添加了几个设计文件，在这个对话框中，将出现多个源文件。此时用户可以选择需要进行仿真的源文件即可。

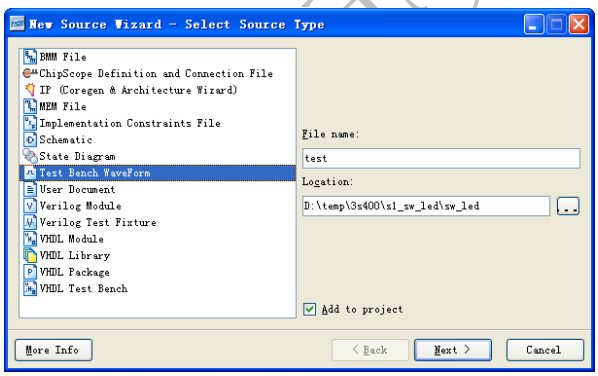


图 2.83　新建 Test Bench 对话框

在本训练中，将要进行仿真的源文件是 sw_led，如图 2.84 所示。单击“Next”按钮完成仿真文件的产生，如图 2.85 所示。

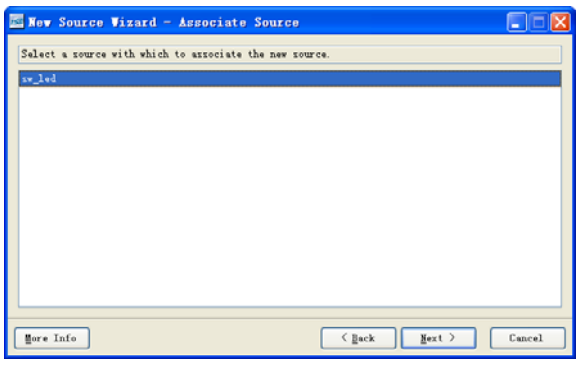


图 2.84　为 Test Bench 指定源文件

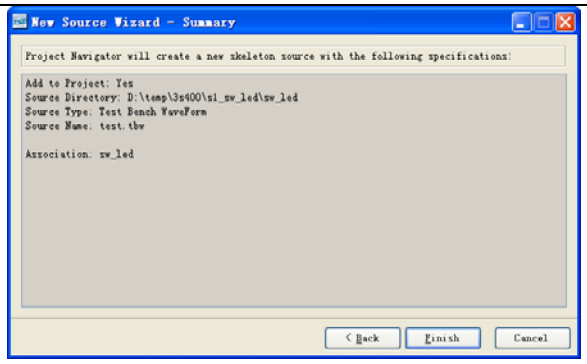


图 2.85　Test Bench 信息对话框

单击“完成”按钮，ISE 将创建名为 test.tbw 的仿真文件，同时弹出如图 2.86 所示的对话框。在该对话框中，用户可以对仿真的时间参数进行设置。

在这个对话框中，包括时钟信息、时钟时间信息、全局信号、仿真时间等多个参数都可以进行设置。按照仿真需求修改好时间参数以后，单击“OK”按钮就能在 ISE 的工作区里面看见新建的波形文件了，如图 2.87 所示。

在波形文件里面，clk 是时钟信号，在前面的时间参数设计里面已经设置好了。用户若需要对时钟进行更改，可选择执行“Test Bench”→“Rescale Timing…”命令，弹出修改时间参数的对话框，如图 2.88 所示。

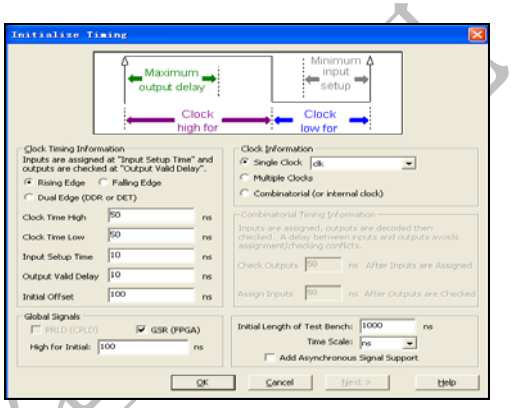


图 2.86　仿真时间参数设置对话框

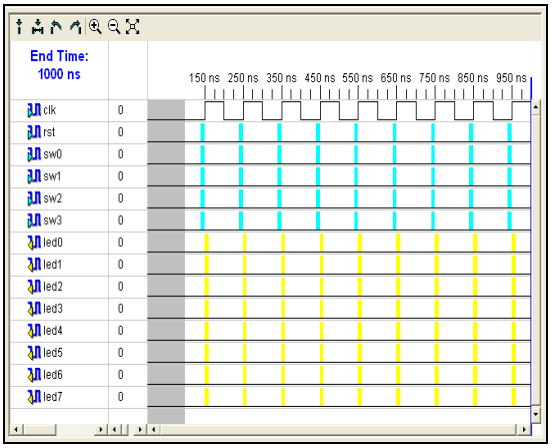


图 2.87　时钟波形

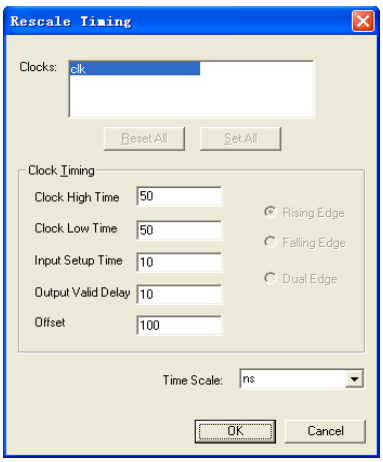


图 2.88　修改时间参数对话框

同样也可以执行“Test Bench”→“Set End of Test Bench…”，在弹出的对话框里面修改仿真波形的截止时间，如图 2.89 所示。

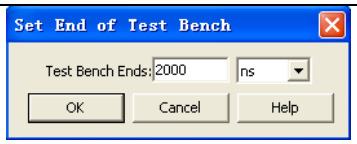


图 2.89　截止时间设置对话框

单击“OK”按钮就会发现仿真波形的时间长度由 1000ns 变成了 2000ns。

② 添加仿真激励。

要进行仿真，还需要为仿真添加激励。这里可以通过点击波形图中的蓝色方块来设置输入波形电平的高低，如图 2.90 所示。

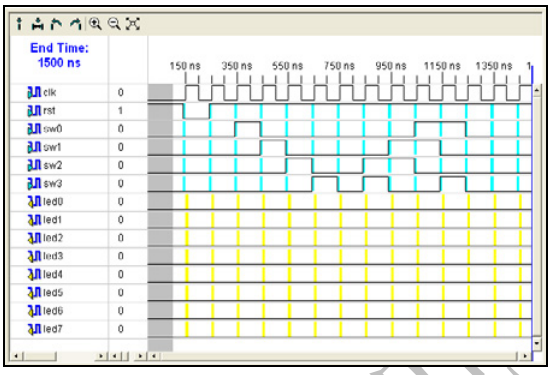


图 2.90　设置输入电平

设置好以后一定不要忘记保存波形文件。保存以后就会在工程浏览器的当前工程的子目录下看见刚生成测试的文件，如图 2.91 所示。

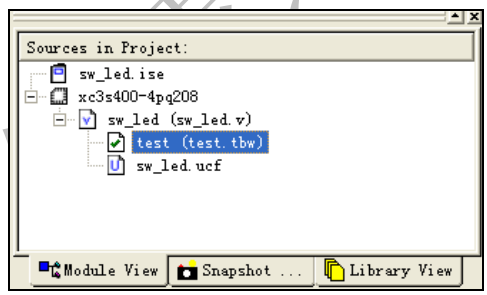


图 2.91　保存仿真文件到工程

③ 仿真。

首先在工程浏览器的【Module View】里面选择“test（test.tbw）”。

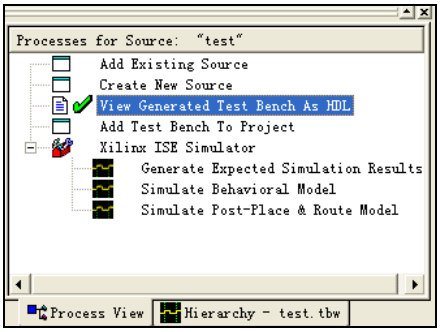


图 2.92　由波形文件生成测试代码

然后在进度浏览器的“Process View”对话框里面双击“View Generated Test Bench As HDL”图标，就能看见 ISE 根据刚才设置的波形文件自动生成的测试文件，如图 2.93 所示。

```
////////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2003 Xilinx, Inc.
// All Right Reserved.
////////////////////////////////////////////////////////////////////////////////
//   ____  ____
//  /   /\/   /
// /___/  \  /    Vendor: Xilinx
// \   \   \/     Version : 7.1i
//  \   \         Application : ISE Foundation
//  /   /         Filename : test.tfw
// /___/   /\     Timestamp : Mon Oct 02 09:52:07 2006
// \   \  /  \
//  \___\/\___\
//
//Command:
//Design Name: test
//Device: Xilinx
//
 timescale 1ns/1ps

module test;
    reg clk = 1'b0;
    reg rst = 1'b1;
    reg sw0 = 1'b0;
    reg sw1 = 1'b0;
    reg sw2 = 1'b0;
    reg sw3 = 1'b0;
    wire led0;
    wire led1;
    wire led2;
```

图 2.93　生成的测试代码

在进度浏览器中双击【Generate Expected Simulation Result】，弹出两个对话框，都选择是（Y），将看见生成的预期输出波形，如图 2.94 所示。

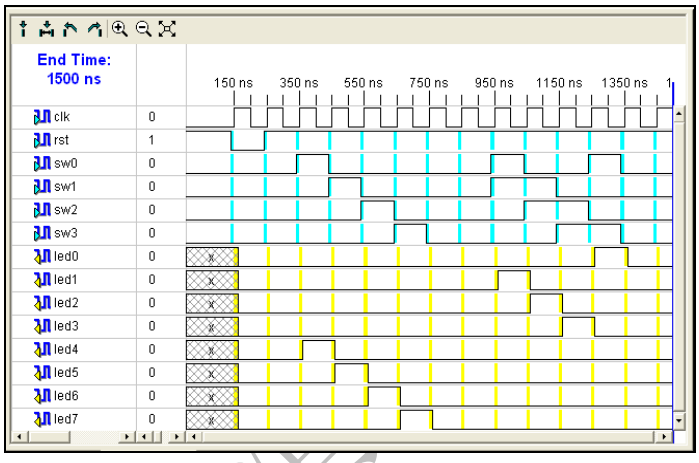


图 2.94　预期仿真波形

在进度浏览器中双击“Add Test Bench to Project”，将测试波形添加到工程里面。将会看到工程浏览器中 test 文件的图标由 test (test.tbw) 变成 test (test_tb.v) 。

如图 2.95 所示，在进度浏览器中双击“Simulate Behavioral Model”将对工程进行行为仿真，仿真结果如图 2.96 所示。

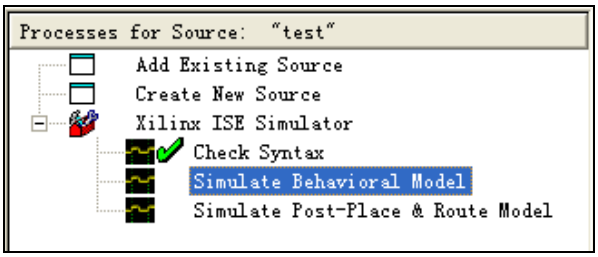


图 2.95　运行功能仿真

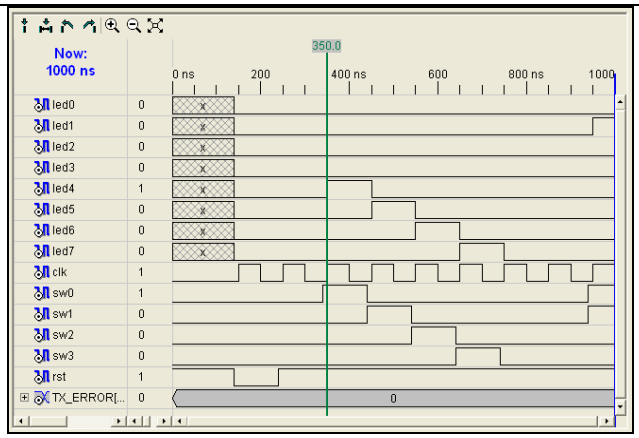


图 2.96　功能仿真结果

从图中可以看出功能仿真结果符合设计要求，可以继续进行下面的设计。如果此处仿真发现设计功能不符合要求，则要对原代码进行修改，直到仿真结果符合设计要求为止。

4．约束设计

在源程序输入完成以后，就可以设置约束，规划布局布线了。

约束有很多种：时序约束（Timing Constrains）、管脚约束（Assign Package Pins）以及面积约束（Area Constraints）等。

如图 2.97 所示是 ISE 进度浏览器中包含的用户约束的选项。

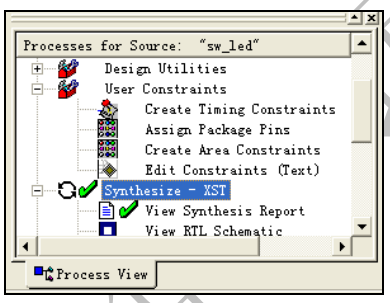


图 2.97　设置约束

（1）时序约束。

时需约束主要约束设计的时序和时钟频率，双击图 2.97 中的“Create Timing Constraints”图标，ISE 将打开约束编辑器，如图 2.98 所示。

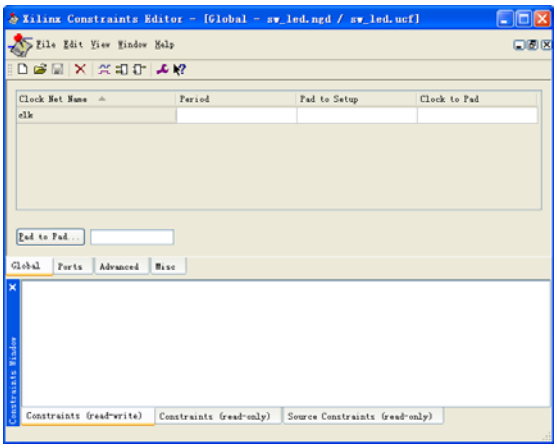


图 2.98　时序约束界面

在约束编辑器里面有 4 个复选页，定义如表 2.5 所示。

表 2.5　　属性定义

属　性	定　义
Global	用于附加全局约束，包括周期约束，输入延迟约束和输出延迟约束
Ports	附加端口约束，可以对每一个端口进行单独设置，包括物理位置、输入延迟和输出延迟。同时，也可以设置分组，进行分组约束
Advanced	附加分组约束（TNM/TNM_NET/TIMEGRP）和时序约束（FROM_TO/TIG O/OFFSET）等
Misc	附加专用约束，包括电压、初始值等

注意　附加约束的原则是先加全局约束后加分组约束。

在本实例中只做简单的时钟约束。在 clk 的 period 中输入：20ns HIGH 50%，即设置时钟的周期和占空比。设置后保存，完成时序约束。

（2）管脚约束。

管脚约束即约束工程设计源文件与选定器件对应的输入/输出管脚属性。双击图 2.97 中的“Assign Package Pins”图标，ISE 将打开 PACE 工具，如图 2.99 所示。

通过图 2.99 的“Design Object List”对话框里面的 Loc 列，可以为设计添加输入/输出管脚，添加形式为“Pxxx”或者“pxxx”。其中 P/p 代表 Pin，xxx 是数值。

（3）面积约束。

面积约束目的在于规划 FPGA 里面的逻辑使用大小。双击图 2.97 中的“Create Area Constraints”图标，ISE 同样打开 PACE 工具，面积约束和管脚约束用的是一个界面。在面积约束里面我们可以对设计使用的资源面积加以规划，控制使用资源在 FPGA 里面的位置。

这里只使用到管脚约束，按照按键和 LED 灯与 FPGA 的连接，输入管脚分配，然后保存分配。在保存选项中，选择“XST Optional”，如图 2.100 所示，关闭对话框，完成约束设计。

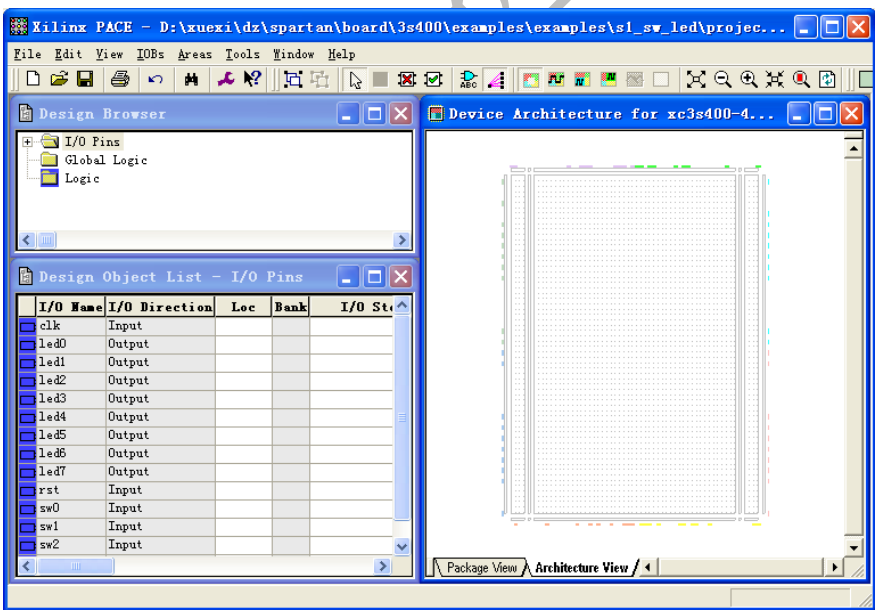


图 2.99　管脚约束工具 PACE

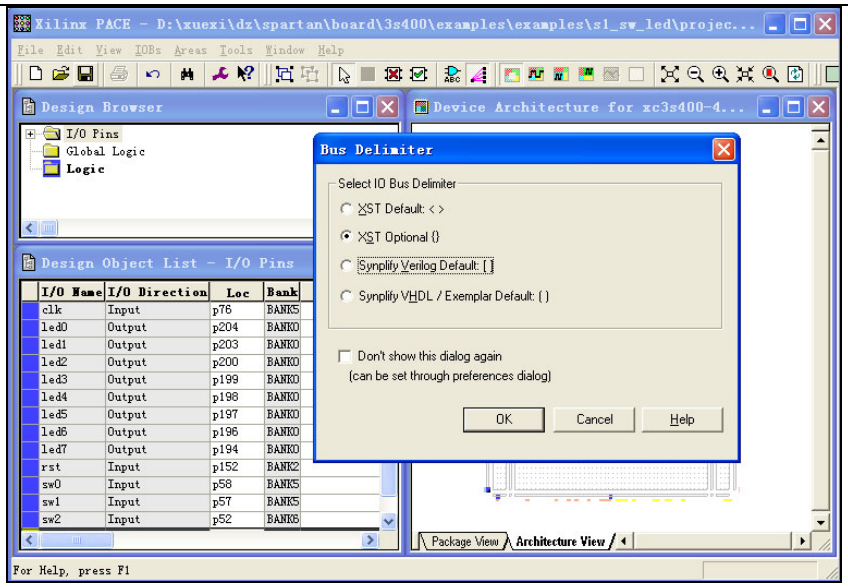


图 2.100　保存管脚约束结果

5．布局布线

设置好了各种约束以后就可以进行布局和布线了。双击进度浏览器中的“Implement Design”，开始进行布局布线实现。与综合相同，如果在“Implement Design”的右边出现绿色的对号，说明已经完成布局布线，没有发现错误，如图 2.101 所示。

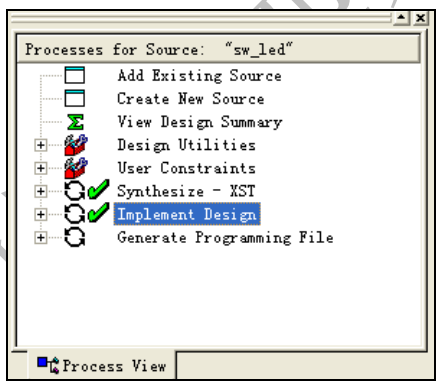


图 2.101　运行布局布线

如果实现过程中出现错误，在信息栏中根据错误的提示，修改相关的约束设计，再进行布局布线的实现，直至没有错误为止。

6．布局布线后仿真

在工程浏览器中选择测试文件，双击进度浏览器中“Implement Design”下面的“Simulate Post-Place & Route Model”，对工程进行布局布线后仿真，仿真结果如图 2.102 所示。

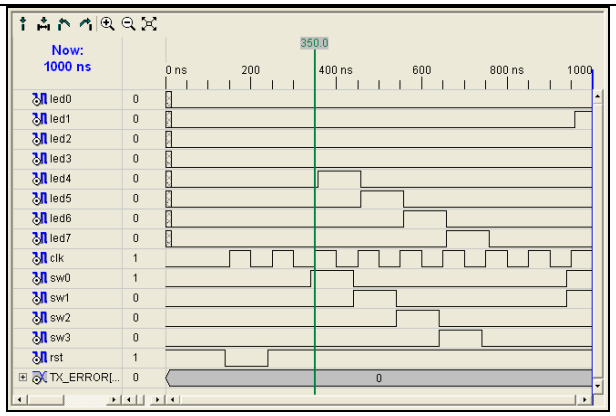


图 2.102　时序仿真结果

通过与行为仿真结果图 2.96 对比，可以发现，布局布线后仿真加入了布线的延迟，输出的结果相对于时钟有相应的延迟。在工作区里面单击鼠标右键，在弹出的菜单里面选择“Add Measure”选项，可以通过放大工作区和拖动时间测量游标，读取时间延迟，如图 2.103 所示。

由上图可以看出，尽管输出相对于时钟存在一定的延时，但输出结果是正确的，符合设计要求，可以进行下载和在线调试了。

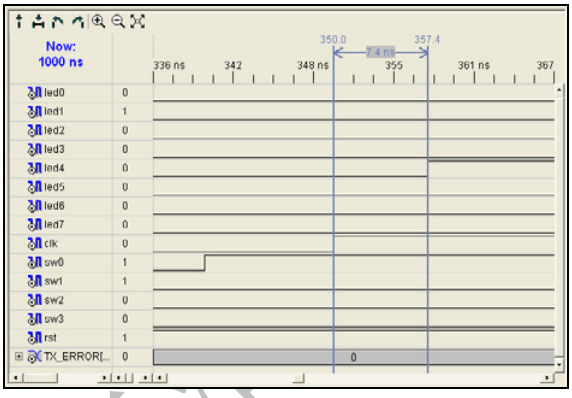


图 2.103　测量时间延迟

7．下载程序

布局布线完成以后就可以将程序下载到开发板上进行调试。开发板提供两种下载方式：JTAG 模式和 PROM 模式。区别在于 JTAG 模式是用于在线调试使用的，断电后不保留程序；而 PROM 模式则是将程序烧写在存储器里面，上电后自动加载。

（1）JTAG 模式下载。

如图 2.104 所示，在进度浏览器中“Generate Programming File”下面可以看到有 3 个选项，分别是编程文件生成报告、生成编程文件和使用 iMPACT 配置器件。

在“Generate Programming File”图标上面单击右键，选择“Properties…”，弹出如图 2.105 所示的属性对话框，在其中可以设置下载文件的属性。

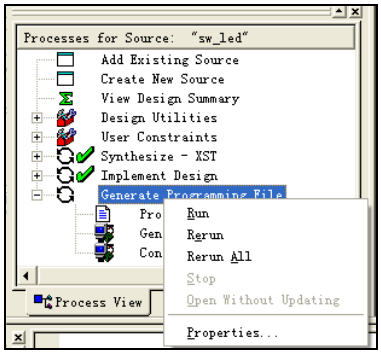


图 2.104 生成下载文件

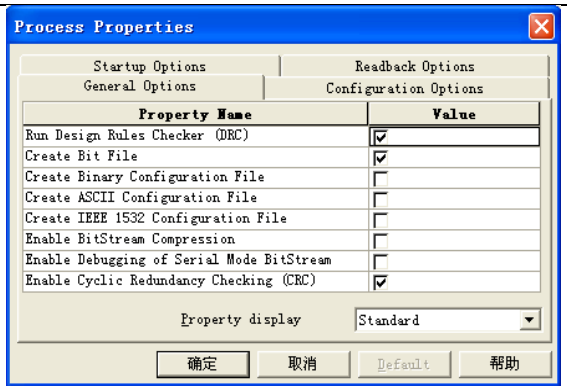


图 2.105 “General Options”选项卡

在该对话框中有 4 个选项卡，可以对相应的下载参数进行设置。

“General Options”选项卡里面的“Create Bit File”选项生成用于 JTAG 模式下载的二进制下载文件。确定选择该项，如图 2.105 所示。

在“Startup Options”选项卡里面将“FPGA Start-Up Clock”后面的内容改为“JTAG Clock”，如图 2.106 所示，在这里要根据不同的下载模式正确地选择时钟。

在“Readback Options”选项卡里面将“Security”后面的内容置为“Enable Readback an Reconfiguration”，如图 2.107 所示，然后单击“OK”按钮，完成下载的属性设置。

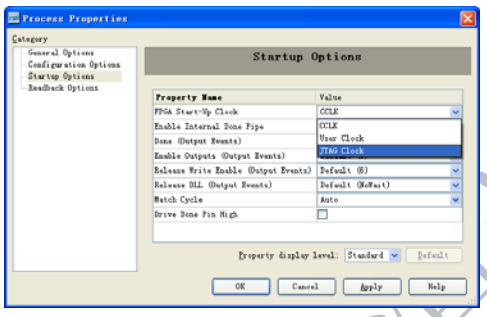


图 2.106 “Startup Options”选项卡

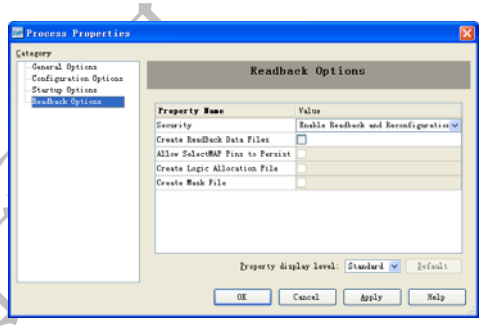


图 2.107 “Readback Options”选项卡

如图 2.108 所示，双击进度浏览器里面的“Generate Programming File”图标，ISE 将生成供 JTAG 模式下载的 bit 文件。“Generate Programming File”左边的绿色对号说明已经成功生成了下载 文件。

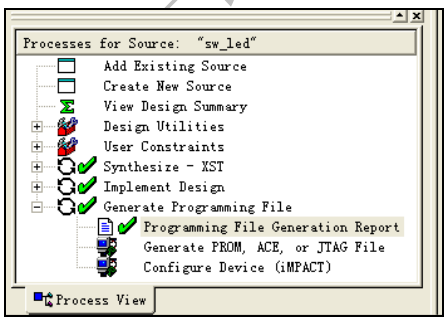


图 2.108 生成下载文件

双击“Generate Programming File”下面的“Configure Device（iMPACT）”图标，打开如图 2.109 所示的 iMPACT 工具。

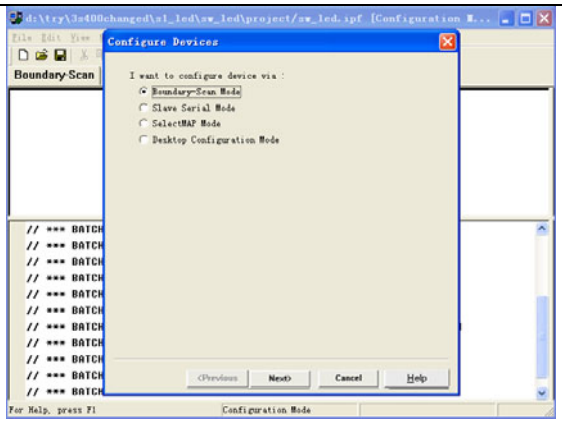


图 2.109　iMPACT 工具

在如图 2.109 所示的对话框里面选择器件配置的模式，这里选择边界扫描模式“Boundary-Scan Mode”，单击“下一步”按钮，如图 2.110 所示。

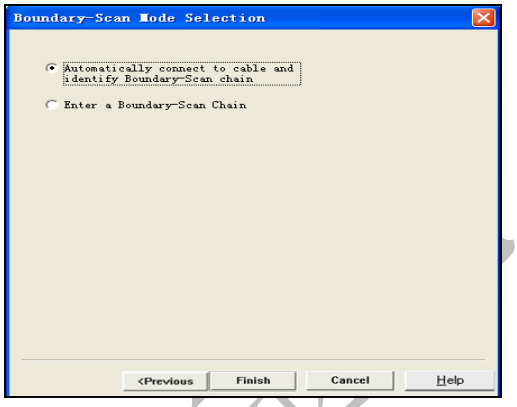


图 2.110　边界扫描模式选择对话框

选择自动加载边界扫描链路，单击完成。

由于开发板上面的 JTAG 和 PROM 下载是在一条链路上的，所以会弹出如图 2.111 所示的对话框，单击“确定”按钮就能看见 ISE 扫描到的开发板上面的下载链，如图 2.112 所示。

图 2.111　边界扫描检测结果对话框

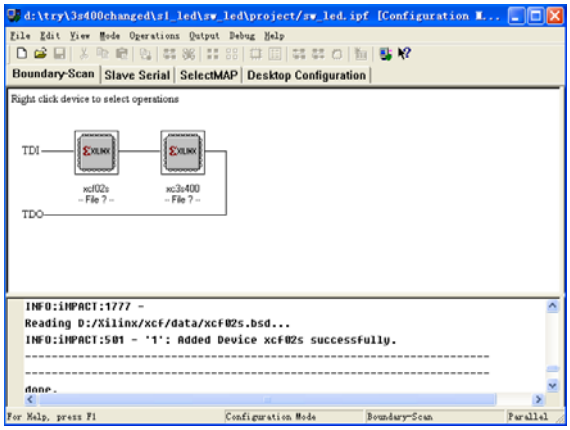


图 2.112　ISE 扫描下载链结果

其中下面标有 xcf02s 的图标代表的是 PROM，下面标有 xc3s400 的图标代表 FPGA。

双击 FPGA 图标，在弹出的对话框里面选择刚才生成的二进制下载文件（后缀为.bit）。配置好以后右键单

击 FPGA 图标，在弹出的菜单里面选择“Program…”选项，如图 2.113 所示。此时将会出现如图 2.114 所示的下载属性设置对话框。

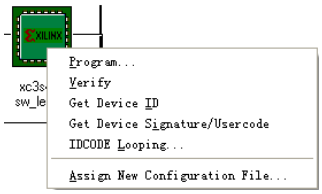


图 2.113　设置配置文件

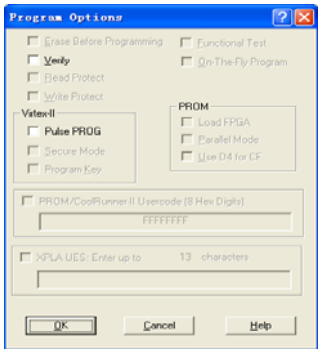


图 2.114　配置设置对话框

单击“OK”按钮就可以开始下载了。若下载成功将会有如图 2.115 所示的下载成功提示。

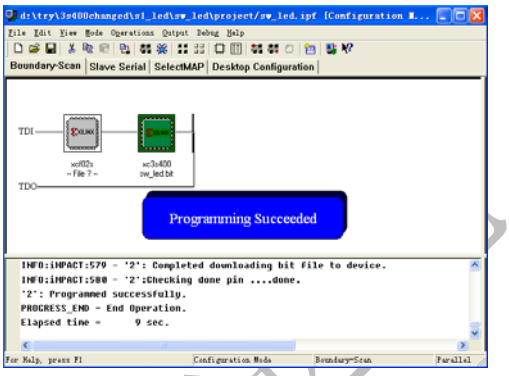


图 2.115　JTAG 下载成功

（2）PROM 模式下载。

PROM 模式下载是把程序存在一块存储器里面（开发板上面使用的是 Xilinx 公司的 xcf02s）。当开发板上电的时候，FPGA 就自动加载存储器里面的程序。下面是 PROM 模式下载的具体操作方法。

要进行 PROM 模式下载，首先要生成相应的二进制下载文件，双击进程浏览器中“Generate Programming File”下面的“Generate PROM, ACE, or JTAG File”图标，弹出如图 2.116 的对话框。

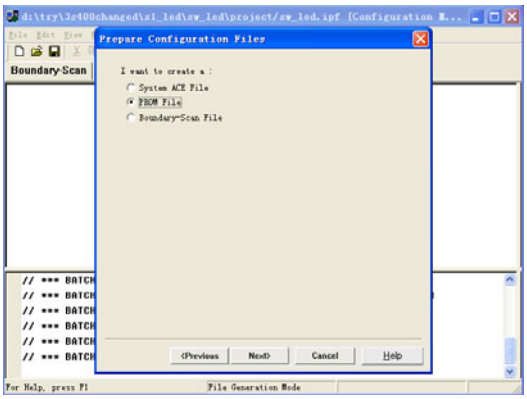


图 2.116　生成 PROM 文件对话框

选择“PROM File”选项，单击“下一步”按钮，打开如图 2.117 所示的 PROM 文件配置对话框。

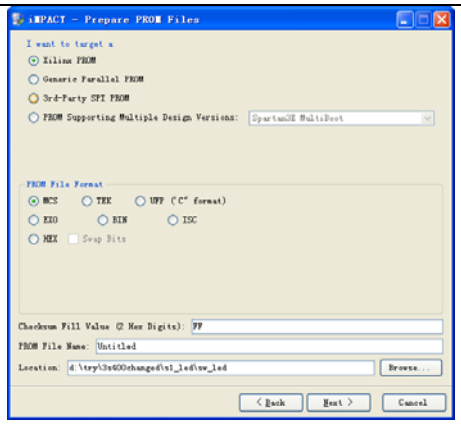


图 2.117　PROM 文件设置对话框

在该对话框中，选择生成一个 Xilinx PROM，后缀选择 MCS，文件格式选择 FF（十六进制），在“PROM File Name”后面填写生成的 PROM 下载文件的名称，在“Location”后面填写 PROM 文件保存地址。配置后单击“Next”按钮，出现 PROM 器件配置对话框，如图 2.118 所示。

选择 PROM 的型号，要与开发板上面使用的对应。这里选择 xcf02s，单击“Next”按钮。

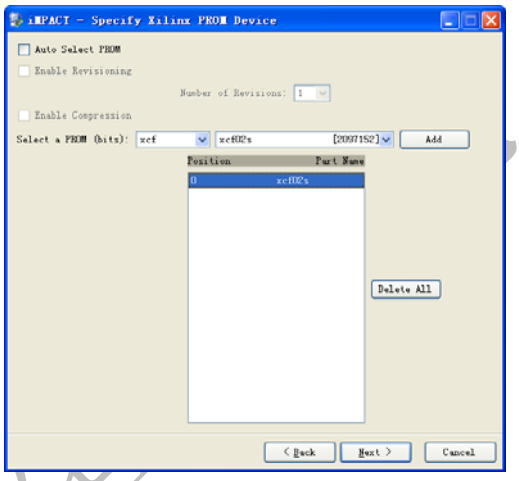


图 2.118　选择 PROM 型号

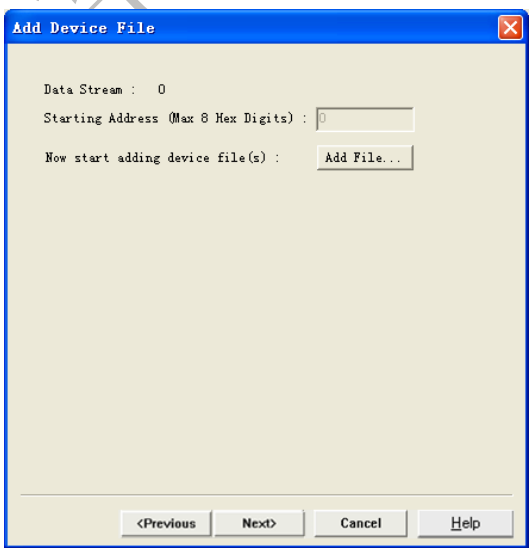


图 2.119　为生成 PROM 文件添加二进制文件

在出现的如图 2.119 所示对话框中单击“Add File…”按钮，在弹出的对话框里面选择 JTAG 下载的时候使用的二进制文件（后缀为.bit），ISE 将通过这个二进制文件生成 PROM 文件。

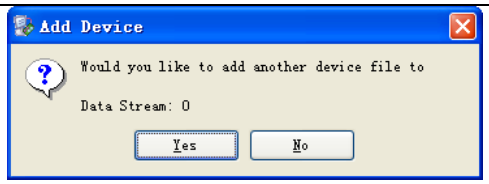


图 2.120　是否还要加载一个设计文件对话框

选择完一个二进制文件后，会弹出如图 2.120 所示的对话框，询问是否还要加载一个设计文件。单击“NO”按钮，完成 PROM 模式下载的配置。

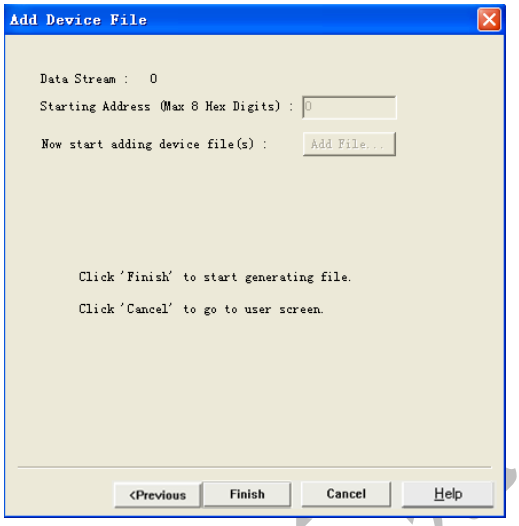


图 2.121　生成 PROM 文件

在出现的图 2.121 中单击“Finish”按钮，ISE 就会自动为我们生成后缀为.MCS 的十六进制 PROM 下载文件。生成成功后，将出现如图 2.122 所示的提示。

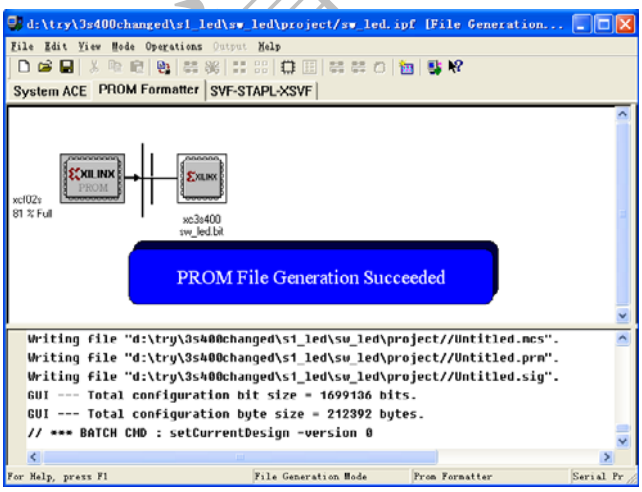


图 2.122　PROM 文件生成成功提示

生成了 PROM 的下载文件以后，再按照 JTAG 下载的方法，扫描下载链。将生成的.MCS 文件加载到 PROM 图标中去，然后再在 PROM 图标上面点击右键，选择“Program…”选项，如图 2.123 所示。

如图 2.124，在出现的下载属性配置对话框中选择相应的选项后，单击“OK”按钮就开始 PROM 的下载了（PROM 下载相对比较慢，并且最好先擦除 PROM 里面的内容再下载）。成功下载会出现如图 2.125 所示的提示。

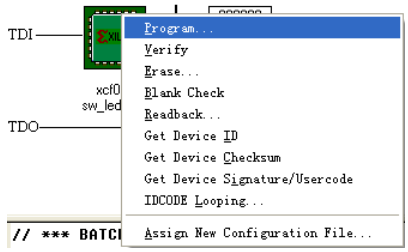


图 2.123　下载 PROM 器件

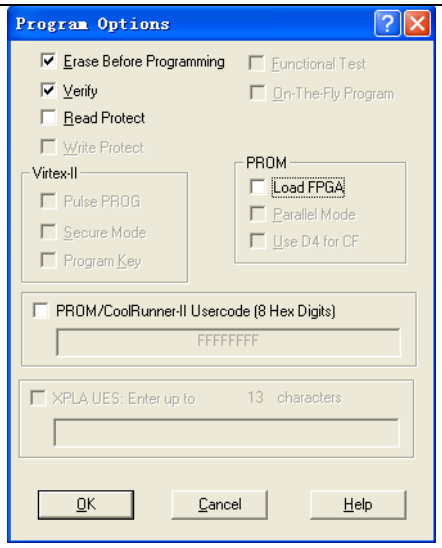


图 2.124　下载 PROM 器件设置对话框

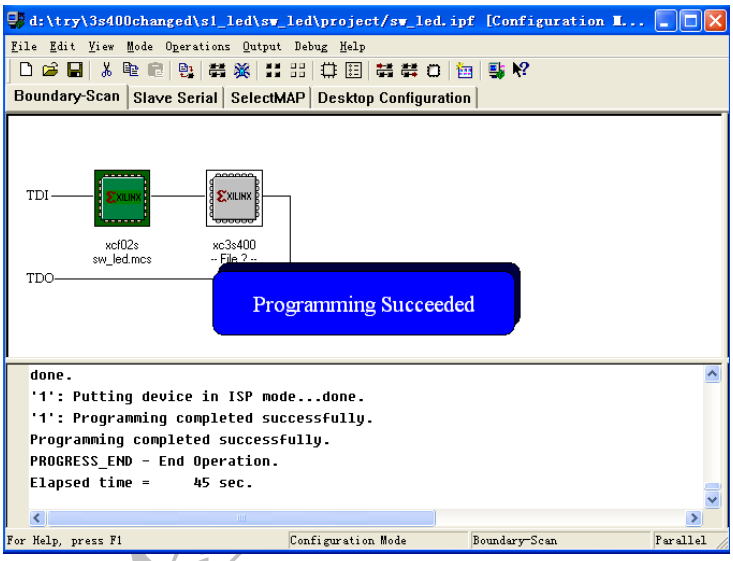


图 2.125　完成对 PROM 器件的烧写

需要注意的是，如果选用 JTAG 下载模式，则下载成功后，FPGA 就可实现预想的功能。如果选用 PROM 下载模式，下载成功后，要重新上电后 FPGA 才能正常工作。

因为 PROM 直接方式是将文件下载到 PROM 器件中，并没有直接下载到 FPGA 里，要重新上电后由 FPGA 自动从 PROM 器件中加载配置文件，才能实现预想的功能。

当然用户也可以在如图 2.124 所示的属性配置中选择“Load FPGA”选项。这样 PROM 下载完成后，下载的逻辑将自动加载至 FPGA 内，而无需重新上电。

2.6.4　小结

上述步骤就是利用 ISE 进行的一个完整的 FPGA 设计流程。虽然例程实现的功能比较简单，但对于初学者来说，是一个不错的入门实例。此实例的主要目的是让初学者对 FPGA 的设计有一个初步的了解，并熟悉 ISE 软件的使用，为今后的学习打下基础。

联系方式

集团官网：www.hqyj.com 嵌入式学院：www.embedu.org 移动互联网学院：www.3g-edu.org

企业学院：www.farsight.com.cn 物联网学院：www.topsight.cn 研发中心：dev.hqyj.com

集团总部地址：北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址：北京市海淀区西三旗悦秀路北京明园大学校区，电话：010-82600386/5

上海地址：上海市徐汇区漕溪路银海大厦 A 座 8 层，电话：021-54485127

深圳地址：深圳市龙华新区人民北路美丽 AAA 大厦 15 层，电话：0755-22193762

成都地址：成都市武侯区科华北路 99 号科华大厦 6 层，电话：028-85405115

南京地址：南京市白下区汉中路 185 号鸿运大厦 10 层，电话：025-86551900

武汉地址：武汉市工程大学卓刀泉校区科技孵化器大楼 8 层，电话：027-87804688

西安地址：西安市高新区高新一路 12 号创业大厦 D3 楼 5 层，电话：029-68785218