第3章 原理图输入设计方法

CHAPTER 3

利用 EDA 工具进行原理图输入，设计者能够利用原有的电路知识迅速入门，完成较大规模的电路系统设计，而不必具备许多诸如编程技术、硬件语言等新知识。Quartus Prime 的图形编辑器为用户提供所见即所得的设计环境，提供了功能强大、直观便捷和操作灵活的原理图输入设计功能，同时还配备了适用于各种需要的元件库，更为重要的是，Quartus Prime 还提供了原理图输入的多层次设计功能，使用户能设计更大规模的电路系统。与传统的数字电路设计相比，Quartus Prime 提供的原理图输入设计功能具有显著的优势。

3.1 原理图设计方法

以原理图进行设计的主要内容在于元件的引入与线的连接。当设计系统比较复杂时，应采用自顶向下的设计方法，将整个电路划分为若干相对独立的模块来分别设计。当对系统很了解且对系统速率要求较高时，或设计大系统中对时间特性要求较高的部分时，可以采用原理图输入方法。这种输入方法效率较低，但对于初学者来说入门方便，容易实现仿真，便于直观地对电路进行调整。

3.1.1 内附逻辑函数

在安装 Quartus Prime 软件时已有数种常用的逻辑函数安装在目录内，这些逻辑函数被称为原语(Primitive)和符号(Symbol)，也称为元件。在电路图编辑窗口中以元件引入的方式将需要的逻辑函数引入，各设计电路的信号输入引脚与信号输出引脚也需要以这种方式引入。有 megafunctions、others 和 primitives 3 个不同的目录分别放有不同种类的逻辑函数文件。

目录 megafunctions 下存放的是一些比较大的并可做参数设置的元件，使用中需要对其参数进行设置，在一些特殊的应用场合，可以调用该目录下的元件。目录 megafunctions 下还包括 IO、arithmetic、gates 和 storage 4 个子目录。

(1) IO 子目录下存放光纤通信接口(alt4gxb)、lvds 输入接口(altlvds_rx)、lvds 输出接口(altlvds_tx)等常用高速外设接口。

(2) arithmetic 子目录下存放整数加减乘除、开方等运算器，以及浮点数加减乘除、开方、倒数、对数、指数等运算器，如常用的整数的乘法器(lpm_mult)和除法器(lpm_divide)、浮点数的对数器(altfp_log)和倒数器(altfp_div)。

（3）gates 子目录下存放总线型的多位数据选择器（busmux）、三态门（lpm_bustri）等。

（4）storage 子目录下存放功能更多的 D 触发器（lpm_dff）、移位寄存器（lpm_shiftreg）等。

目录 megafunctions 下部分元件与 IP 目录下的核元件功能相同，名称上大小写不同，如移位寄存器，在 megafunctions→storage 子目录下有 lpm_shiftreg，而在 IP 目录下也有 LPM_SHIFTREG。在使用上，IP 目录下的核元件必须经过参数设定例化后才能使用，而 lpm_shiftreg 可以直接在原理图或 HDL 程序中调用，只要传递必要的参数即可。

目录 others 下存放的是数字电路中一些中规模器件库，包括常用的 74 系列逻辑器件等。将这些逻辑电路直接运用在逻辑电路图的设计上，可以简化许多设计工作。

目录 primitives 下存放的是数字电路中一些常用的基本元件库，primitives 下又包括 buffer、logic、other、pin 和 storage 5 个子目录。

（1）buffer 子目录下包含输入缓冲器（alt_in_buf）、输出缓冲器（alt_out_buf）、双向缓冲器（alt_inout_buf），以及三态门（tri）和连线（wire）。

（2）logic 子目录下有各种逻辑门电路，如与（and2）、或（or2）、非（not）、与非（nand2）、异或（xor）等，还有多输入端的门电路可选，如 4 输入的与非门（nand4）。

（3）other 子目录下包含高电平（vcc）、地（gnd）、标题（title）等元件或标识。

（4）pin 子目录下有双向（bidir）、输入（input）、输出（output）3 种端口。

（5）storage 子目录下包含各种触发器，如 D 触发器（dff）、JK 触发器（jkff）等。

3.1.2 编辑规则

在进行原理图设计时，经常需要对一些引脚、文件等进行编辑与命名，进行命名时必须按一定的规则进行。

1. 引脚（pin）名称

利用原理图进行设计时，经常需要用到输入/输出信号，就需要使用输入/输出引脚，此时必须对输入/输出引脚进行命名，命名时可采用英文字母 A～Z 或 a～z，阿拉伯数字 0～9，或是一些特殊符号如“/”“_”“－”等。例如，abc、d1、123_abc 等都可以命名。要注意英文字母的大小写代表的意义是相同的，也就是说，abc 与 ABC 所代表的是同样的引脚名称；还要注意名称所包括的英文字母长度不可以超过 32 个字符；另外，在同一个设计文件中不同的引脚名称不能重复。

2. 节点（node）名称

节点在图形编辑窗口中显示一条细线，它负责在不同的逻辑器件之间传送信号。也可以对节点进行命名，其命名规则与引脚名称相同，注意事项也相同。

3. 总线（bus）名称

总线在图形编辑窗口中显示一条粗线。一条总线代表很多节点的组合，可以同时传送多个信号。总线命名时，必须要在名称后面加上[$m..n$]表示一条总线内所含有的节点编号，m 和 n 都必须是整数，但谁大谁小均可，并无原则性规定。

4. 文件名称

原理图的文件名可以用任何英文名，扩展名为“.bdf”，文件名称小于或等于 32 个字符，扩展名并不包括在 32 个字符的限制之内。

5. 项目名称

一个项目(Project)包括所有从电路设计文件编译后产生的文件,这些文件是由Quartus Prime程序所产生的,有共同的文件名称,但其扩展名称各不相同,而项目名称必须与最高层的电路设计文件名称相同。

3.1.3　原理图编辑工具

下面介绍的是在原理图编辑时所用到的快捷工具按钮,熟悉这些工具的基本性能,可大幅提高设计时的速度。

(1) 选择工具:可以选取、移动、复制对象,为最基本且最常用的功能。

(2) 放大/缩小工具:可以放大/缩小所编辑的图形。

(3) 拖动工具:按住鼠标左键可以拖动整张原理图。

(4) A 文字工具:可以输入或编辑文字,例如,在指定名称或批注时使用。

(5) 添加元件工具:可以添加 megafunctions、others 和 primitives 目录下的或者本工程中自定义的元件。

(6) 添加输入/输出端口工具:可以添加输入、输出、双向端口,单击下拉三角可以选择不同端口。

(7) 画 block 工具:用于框图设计。

(8) 画直角节点连线工具:可以画出一条直角细线。

(9) 画直角总线连线工具:可以画出一条直角粗线。

(10) 画直角导管连线工具:可以画出一条直角空心粗线。

(11) 画节点斜线工具:可以画出一条任意斜率的细斜线。

(12) 画总线斜线工具:可以画出一条任意斜率的粗斜线。

(13) 画导管斜线工具:可以画出一条任意斜率的空心粗斜线。

(14) 画矩形工具:可以画出一个任意大小的矩形。

(15) 画椭圆工具:可以画出一个任意大小的椭圆。

(16) 画直线工具:可以画出一条任意长短的斜线。

(17) 画弧线工具:可以画出一条任意长短的弧线。

(18) 部分连线选择功能:使用选择工具选择连线时,可以选择部分连线。

(19) 使用橡皮筋绑定功能:可以使连线如橡皮筋一样,此时移动同连线相接的模块,连线也会随着移动而不会断开。

(20) 镜像/旋转功能:可以使某个元件左右镜像、上下镜像,以及逆时钟旋转90°。

3.1.4　原理图编辑流程

Quartus Prime 18的原理图编辑流程如下。

1. 建立工程

任何一项设计都是一项工程(Project),都必须首先为此工程建立一个放置与此工程相关文件的文件夹,此文件夹被EDA软件默认为工作库(Work Library)。一般而言,不同的设计项目最好放在不同的文件夹中。一个设计项目可以包含多个设计文件,这些文件包括

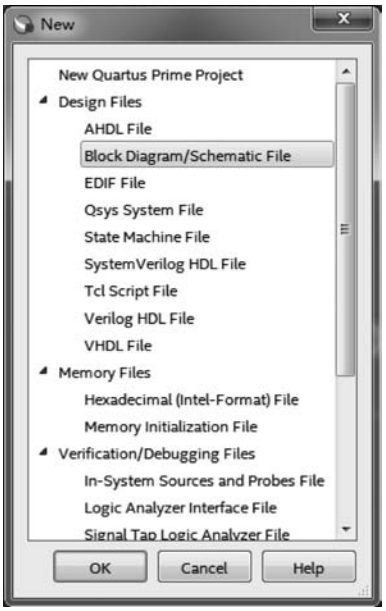


图 3-1　New 对话框

所有的层次设计文件和由设计者或 Quartus Prime 18 软件产生的副文件。必须注意文件夹名称不能用中文，且不可带空格。

2. 进入原理图设计系统

在主菜单上选择 File→New，或单击工具栏上的 图标，或利用快捷键 Ctrl＋N，在弹出的 New 菜单中选择 Block Diagram/Schemetic File 后单击 OK 按钮，如图 3-1 所示。这时将会出现一个 Block1.bdf 的无标题图形编辑窗口。

3. 输入元件

对于 Quartus Prime 18 软件而言，系统本身自带了不少元件，可以直接调用。调用方法如下。

（1）首先用鼠标左键选择工具栏上的 工具，或双击原理图空白处，将出现如图 3-2 所示的对话框。

（2）在对话框左上方的 Libaries 库中将出现 megafunctions、others 和 primitives 3 个目录，各目录下存放的主要元件已在 3.1.1 节中介绍。

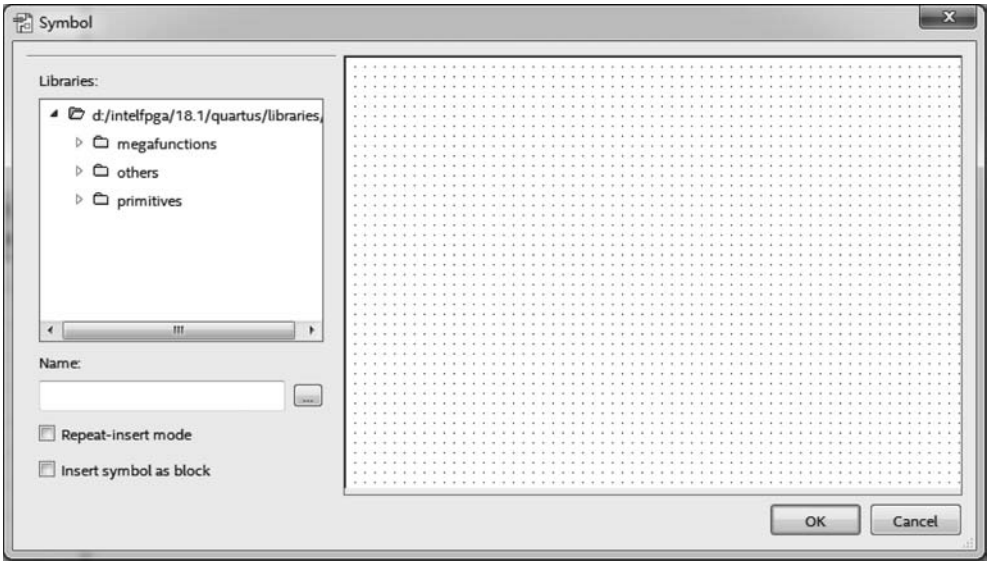


图 3-2　输入元件对话框

（3）选择要输入的元件，然后单击 OK 按钮确定。

若要输入的是 74 系列元件，则在 others→maxplus2 子目录下查找，也可以直接在 Name 文本框中输入型号名称，如要添加 4 位二进制计数器，在 Name 文本框中输入 74161，在对话框右侧元件图形框中将出现 74161 元件外形图（包含输入、输出引脚），单击 OK 按钮即可添加到原理图上。

对于参数可设置的元件，在原理图上添加该元件后，在元件的右上角会出现可重置的参

数列表，双击某参数，在弹出的对话框中即可进行参数设置。

4. 元件的编辑

元件被放置到原理图中后，还需要调整它们的位置，使其布局合理。常采用以下方法进行调整。

(1) 移动：用鼠标左键选中待移动的元件后，出现一个蓝色选择框，然后将其拖到合适的位置松开即可。若要同时移动多个元件，则在空白处按住鼠标左键后画出一个矩形框，把要移动的元件置于其中，然后用鼠标拖动即可；也可以按住 Ctrl 键同时选中多个元件后移动，移动时要松开 Ctrl 键，否则在移动的同时会复制出被选中元件。

(2) 旋转：当元件的摆放方向不理想时，可以通过旋转对其进行调整。其方法是用鼠标左键选中该元件后，右击出现快捷菜单，选择 工具(水平镜像)、 工具(上下镜像)、 工具(逆时针旋转 90°)进行调整，也可以在菜单 Edit 下进行同样的操作。

(3) 删除：选中要删除的元件后按 Delete 键即可，也可以在菜单操作方式下用 Edit→Delete 操作。如果要同时删除多个元件时，按上面讲的方法同时选中多个元件后按 Delete 键即可。

(4) 复制：当要放置多个相同的元件符号时，一般采用复制的方法。一种方法是选择菜单操作方式，用 Edit→Copy 进行复制，用 Edit→Paste 进行粘贴；另一种方法是选中要复制的元件后，按住 Ctrl 键再用鼠标进行拖动，这时元件边会出现一个小“＋”号；还可以通过右击弹出菜单来完成，或者利用快捷键 Ctrl＋C 复制、Ctrl＋V 粘贴。

5. 连线

放置好元件后，接下来就要实现对功能模块间逻辑信号的连接。有两种方法可以将元件的相应管脚连接起来，第 1 种方法是直接连接法，即通过导线将模块间对应的管脚直接连接起来。具体方法如下。

(1) 如果需要连接两个端口，将鼠标移到其中的一个端口，则鼠标变为“＋”形状。

(2) 一直按住鼠标的左键，将鼠标拖到待连接的另一个端口上。

(3) 放开左键，则一条连线画好了。

(4) 如果需要删除一根线，单击这根连线并按 Delete 键。

这种方法的优点是直观，但当模块比较多、管脚比较多时，会使原理图中连线繁杂，看起来很混乱。为了使图形文件连线明了简洁，就需要采用另一种方法，即标注连接法，在要连接的元件的管脚上做相同的标注，系统在编译时，会认为标注相同的地方在逻辑关系上是连接在一起的。

6. 命名

连线完成后，可以给引线端子和节点命名。

(1) 给引线端子命名：可以在引线端子的 PIN_NAME 处双击，然后输入名字。也可以在引线端子符号任意处右击，在弹出的快捷菜单中选择 Properties，然后输入名字，注意如果是总线型端子，则在名称后面要加上[$m..n$]。

(2) 给节点命名：选中需命名的线，右击，在弹出的快捷菜单中选择 Properties，然后输入名字即可，同样地，总线型节点在名称后面需加上[$m..n$]。

7. 总线

总线是一组相关的连线，总线的建立可以通过画线方式，只要在工具栏上选择 即可。

对 n 位宽的总线命名可以右击总线，在弹出的菜单中选择 Properties 并在弹出的对话框中进行命名，一般采用 $A[n-1..0]$形式，其中，单个信号用 A_{n-1}，…，A_2，A_1，A_0 形式，A_{n-1} 代表最高有效位，A_0 代表最低有效位。

8. 保存文件

选择 File→Save As 子菜单，或单击工具栏上的 图标，将出现“另存为”对话框，如图 3-3 所示，在“文件名”文本框内输入设计文件名，默认文件名为工程名，后缀名默认为 bdf，默认保存路径为当前工程文件夹，Add file to current project 复选框保持默认勾选状态，然后单击“保存”按钮即可保存文件。此时，保存文件的同时，也将该文件加入工程中，启动编译时会对加入工程的每个文件进行编译。

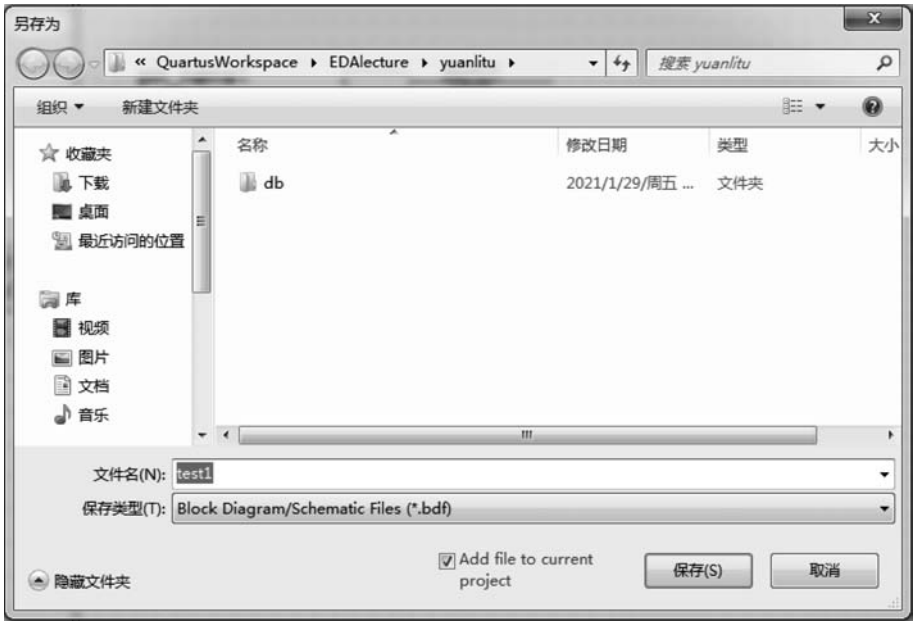


图 3-3 “另存为”对话框

9. 将当前设计文件设置成工程的顶层设计实体

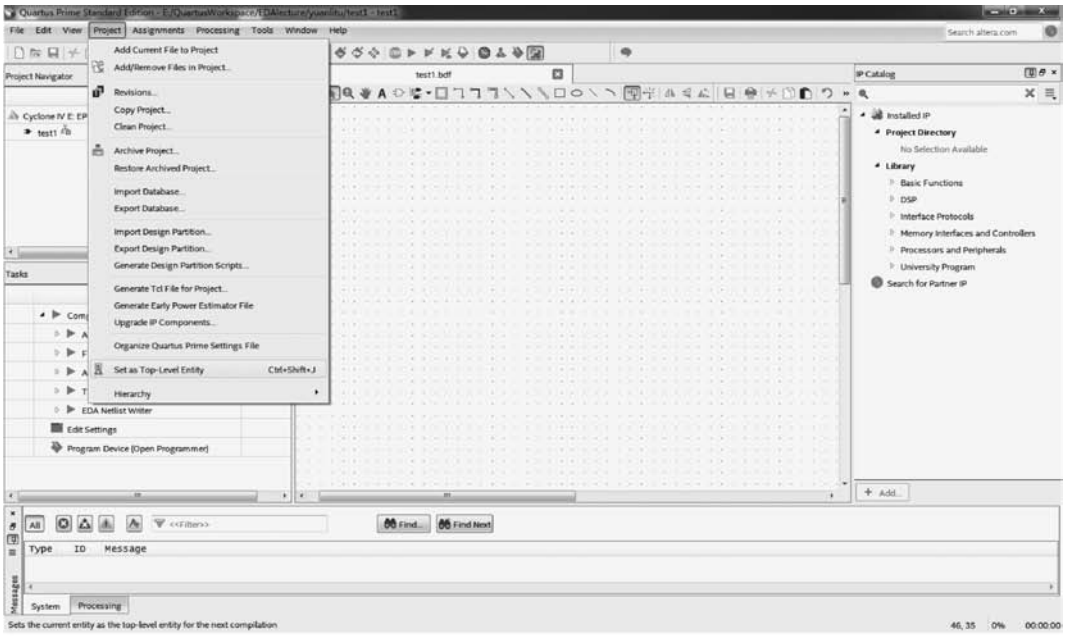
设计文件既可以是原理图也可以是 HDL 程序，将当前设计文件设定为工程的顶层设计实体有以下两个途径。

(1) 选择 Project→Set as Top-Level Entity，即将当前设计文件设置为顶层设计实体，如图 3-4(a)所示。

(2) 如果设计文件未打开，可先将工程导航区切换为 Files，然后从文件列表中找到已加入该工程的设计文件，右击该文件，从弹出的快捷菜单中选择 Set as Top-Level Entity，如图 3-4(b)所示，设定之前要确保该设计文件已经加入本工程。顶层设计名称在工程建立之初与工程名同名，指定了新的顶层设计实体后，顶层设计名称将跟随顶层设计文件名变换。

10. 创建元件

创建元件是建立一个新符号来代表当前设计文件，在其他高层设计文件中可以像调用一般元件一样直接调用它，类似其他软件的子电路生成功能。创建前，要首先用 File→Create/Update→Create Symbol File for Current File 子菜单，检查设计是否有错误，若正确



(a)

(b)

图 3-4　将当前设计文件设置成工程文件

无误，即可在当前工程路径文件夹中创建一个设计符号文件，扩展名为.bsf。创建完成后可以发现所设计的电路变成了一个具有输入和输出端口的元件，下次要用的时候直接调用就可以了。调用方法与添加元件相同，用鼠标左键选择工具栏上的 工具，或双击原理图空白处，弹出的对话框中左上方 Libaries 列表中不仅有 Quartus 自带的库元件，还有本工程中

自建的元件。这样可以大大减轻设计者的工作量，缩短设计开发周期。

3.1.5 设计项目的处理

Quartus Prime 18 编译器是一个高速自动化的设计处理器，能完成对设计项目的编译。它能够将设计文件转换成器件编程、仿真、定时分析所需要的输出文件，是 Quartus Prime 18 系统的核心。

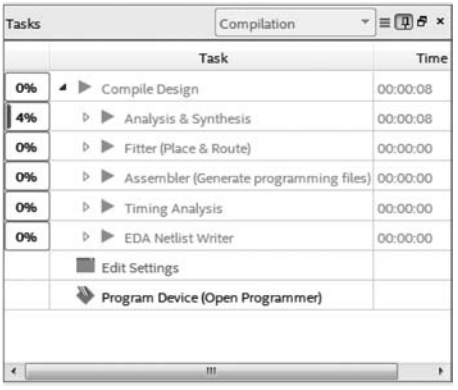


图 3-5 Quartus Prime 18 编译器窗口

1. 项目编译

1）启动编译器

在 Quartus Prime 18 菜单中选择 Processing→Start Compilation 项或单击工具栏上的 ▶ 按钮，则在主界面左中区域的 Tasks 区中看到编译进度条，如图 3-5 所示。

Quartus Prime 18 编译器将检查项目是否有错误，并对项目进行逻辑综合，然后配置到一个 Altera 器件中，同时产生编译文件、报告文件和仿真输出文件等。在编译器编译项目过程中，所有的信息、错误和警告将在自动打开的 Messages 信息处理窗口中显示出来。如果发现有错误，双击该错误，就能直接在设计编辑区域找到该错误在设计文件中所处的位置。

2）编译器的编译过程

任务区的编译窗口中的五个进程模块分别是：Analysis & Synthesis（分析与综合）、Fitter（适配器）、Assembler（装配器）、Timing Analysis（时序分析）、EDA Netlist Writer（EDA 网表生成器）。

编译过程描述如下。

（1）分析与综合：分析主要是检查 HDL 程序和原理图设计文件中的语法或电路设计错误。综合的任务是根据设计者逻辑功能的描述及约束条件如速度、功耗、成本、器件类型等，将用行为和功能层次表达的电子系统转换为低层次的、便于具体实现的逻辑电路的组合，给出满足要求的最佳实现方案，生成网表文件。

（2）适配器：将由综合器产生的网表文件针对某一指定的目标器件进行逻辑映射操作，包括底层器件配置、逻辑分割、优化、布局布线等操作。适配器将每个逻辑功能分配给最佳的逻辑单元位置，并选定相应的互连路径和引脚分配。

（3）装配器：将由适配器得到的器件、逻辑单元和引脚分配转换为器件的编程镜像，其形式是目标器件的在系统编程文件 SRAM Object File(.sof)和固化配置文件 Programmer Object File(.pof)。

（4）时序分析：分析寄存器到寄存器、寄存器到输出端、输入端到寄存器等路径上的延时。经过布局布线后的时序逻辑电路，其最佳状态应使各触发器的时钟信号上升沿到来之前和到来之后的一小段时间内，数据保持稳定不变，特别是总线上的各条路径。时序分析报告将给出布局布线后逻辑电路中各信号的延时情况及警告，用以给设计人员评估若有延时过大的路径是否会带来致命性的功能故障。

(5) EDA 网表生成器：生成与其他 EDA 工具配合使用的网表文件和其他输出文件，如用于功能或时序仿真的 VHDL Output 文件(.vho)和 Verilog Output 文件(.vo)，以及使用 EDA 仿真工具进行时序仿真时所需的 Standard Delay Format Output 文件(.sdo)。

3) 选择器件

在新建工程时即可指定目标器件，在工程建立完成后，在开始编译前，还可以更改目标器件，其方法是：在主界面 Assignments 菜单内选择 Device 项，或者双击主界面左上方工程导航区中当前选定的目标器件，将出现 Device 对话框，如图 3-6 所示；然后选择一个器件系列；再选择某一器件或 AUTO 自动选择；最后单击 OK 按钮。

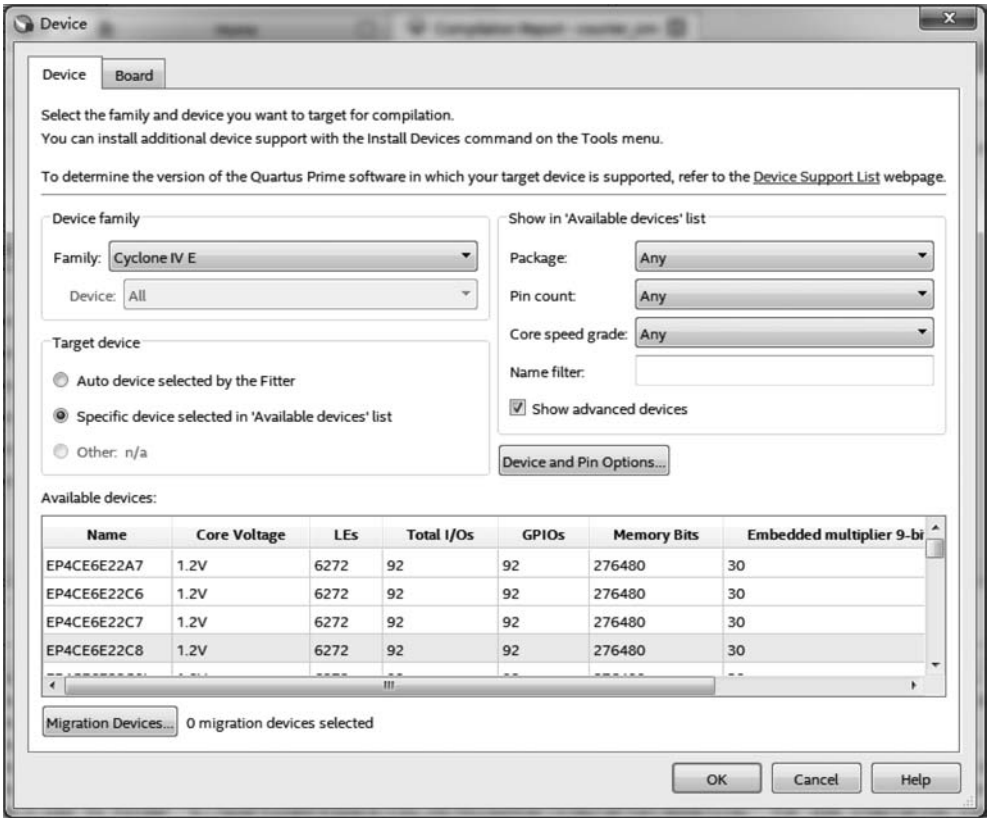


图 3-6　器件选择对话框

2. 引脚锁定

为了能对某设计进行硬件验证，应将顶层设计实体中的输入、输出、双向信号锁定在芯片确定的引脚上，并编译下载到 FPGA 上。在 Quartus Prime 18 已经打开某一工程并且完成第一次编译后，选择 Assignment→Pin Planner 子菜单，即可打开如图 3-7 所示的引脚锁定编辑窗口。

在 Pin Planner 窗口最下方的表格中即可完成引脚锁定的编辑。表中 Fitter Location 列已经显示锁定好的引脚，这只是在 Quartus Prime 对工程编译后自动对电路的输入、输出端给出引脚位置，并不是设计人员给出的引脚。双击表中 Location 栏对应的信号位置，手动输入对应的引脚，以“PIN_”开头并在其后跟引脚号，输入引脚号后按 Enter 键即可将一

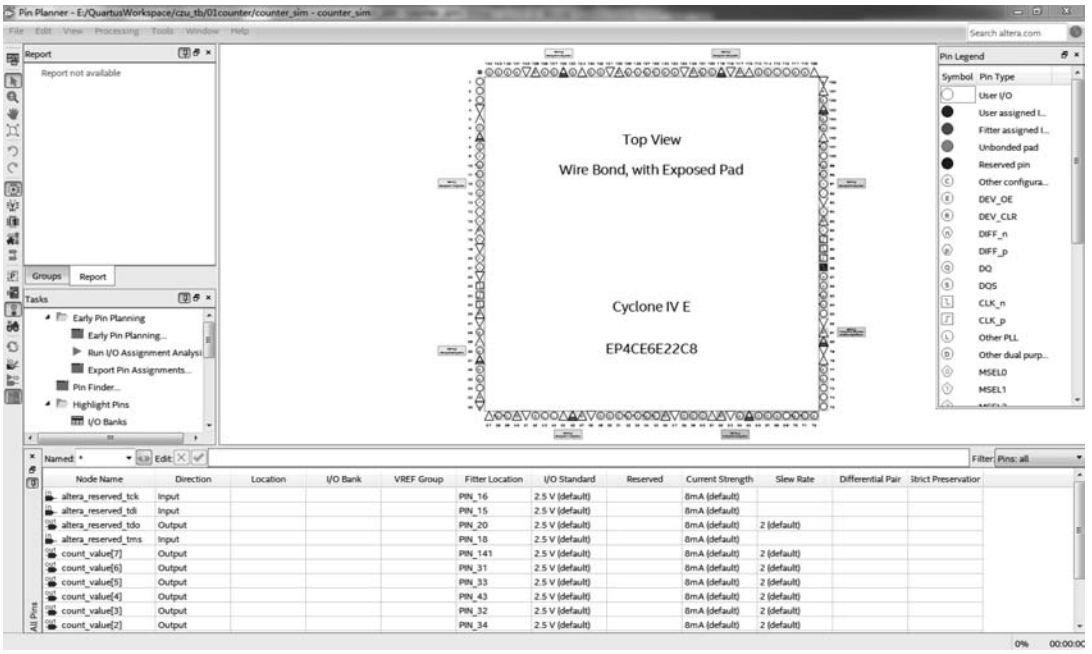


图 3-7　Pin Planner 编辑窗口

个端口的引脚锁定。注意，当输入所希望的引脚编号时，若发现其显示不出来，则说明此引脚不存在，或者此引脚只能作为输入口，不能作为输出口，再或者此引脚已被占用。带有 ARM 核的 FPGA，既有逻辑电路侧的 GPIO 引脚，也有处理器侧的 GPIO 引脚，两者不能混用。Pin Planner 即使接受此引脚名，也有可能在编译时报错，因为违反了引脚锁定规则。例如，一组 LVDS 差分总线输入端，同组的 LVDS 差分线必须指定在 FPGA 的同一 bank 内，否则编译会报错而无法生成 sof 下载文件。因此，建议开发人员在设计搭载 FPGA 的 PCB 电路板过程中，事先利用 Quartus Prime 软件进行引脚锁定实验，并完成编译，以免 PCB 板设计完成后再发现引脚锁定违反规则，导致 PCB 板报废。

只要引脚锁定发生变更，都必须重新编译后才能将引脚锁定信息编译进下载文件中。引脚数较少的 FPGA，其引脚号是纯数字，如图 3-7 所示的 Cyclone IV EP4CE6E22C8 型号芯片，共 144 个引脚，引脚号最大的是 PIN_144。而引脚数较多的 FPGA，尤以 BGA 封装的芯片，引脚号是字母加数字的组合，如 PIN_AB12 等。另外，在列表的 I/O Standard 列还可以设定引脚的电压标准，默认为 2.5V，最高 3.3V，最低 1.2V，如图 3-8(a)所示。同时，还可以在 Current Strength 列设定引脚的输出电流大小，默认为 8mA，基本可以点亮各种颜色的普通 LED 指示灯。这里，可以根据引脚外接电路的实际需要调节电流大小，如图 3-8(b)所示。

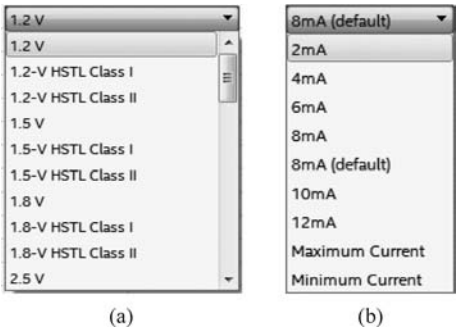


图 3-8　引脚电压和电流编辑窗口

上面提到在编译过程中的时序分析会分析寄存器的数据走线和时钟走线上的延时，为

了尽量减少时钟走线的延时，Quartus 提供了全局时钟走线的功能，即将设计中最主要使用的参考时钟设定为全局时钟，这样该时钟信号会通过 FPGA 片上的全局时钟网络走线，从而减少时钟线上延时，设定全局时钟的方法如下。

(1) 首先，在如图 3-7 所示的 Pin Planner 窗口指定时钟的引脚。

(2) 其次，单击 Assignments→Assignment Editor 子菜单，在 Assignment Editor 窗口列表中，将时钟的 Assignment Name 属性指定为 Auto Global Clock，如图 3-9 所示。

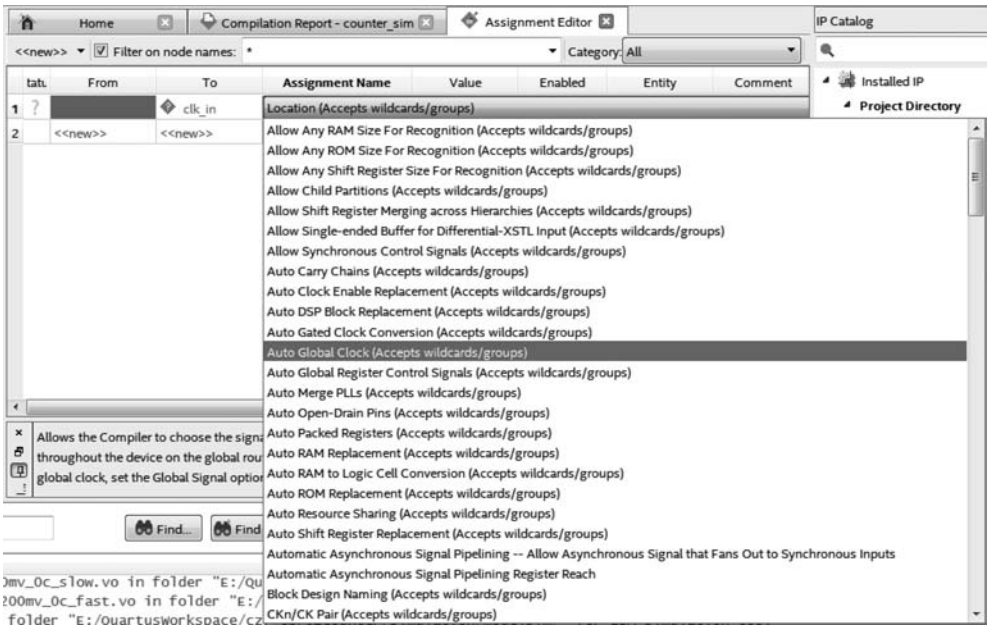


图 3-9　设定全局时钟

将设计中的参考时钟指定为全局时钟是最常用，也是最有效的时序约束方法。

3.1.6　设计项目的校验

Quartus Prime 的设计项目的校验包括设计项目的仿真(Simulation)和时序分析(Timing Analysis)两个部分。

1. 仿真

Quartus Prime 的仿真器(Simulator Waveform Editor)是一个测试电路的逻辑功能和内部时序的强大工具，可灵活地建立单个或多个器件的设计模型。一个设计项目完成输入和编译后只能保证为项目创建了一个编程文件，但还不能保证是否真正达到了设计要求，如逻辑功能和内部时序要求等，所以在器件编程之前还应进行全面模拟检测，以确保它在各种可能情况下的正确响应，这就是 Quartus Prime 的仿真器的作用。

仿真包括功能仿真和时序仿真，这两项工作在设计处理过程中同时进行。

功能仿真是在设计输入完成后，选择具体器件进行编译之前的逻辑功能验证，因此又称为前仿真。仿真前，要先利用波形编辑器或硬件描述语言等建立波形文件或测试向量，仿真结果将会生成报告文件和输出信号波形，从中便可以观察到各个节点的信号变化，若发现错误，则返回设计输入修改逻辑设计。

时序仿真是在选择了器件并完成布局、布线之后进行的时序关系仿真，因此又称为后仿真或延时仿真。由于不同器件的内部延时不一样，不同的布局、布线方案也给延时造成不同的影响，因此在设计处理后，对系统和各功能模块进行时序仿真，分析其时序关系，实际上也是与实际器件工作情况基本相同的仿真。

设计人员可利用 Quartus Prime 的仿真器进行功能和时序仿真。功能仿真是在不考虑器件延时的理想情况下仿真项目的逻辑功能，时序仿真是在考虑设计项目具体适配器件的各种延时情况下仿真设计项目的验证方法，不仅测试逻辑功能，还测试目标器件最差情况下的时间关系。在仿真过程中，需要给仿真器提供输入信号，仿真器将产生对应用于这些输入激励的输出信号，在时序仿真时，仿真结果与实际的可编程器件在同一条件下的时序关系完全相同。

使用 Quartus Prime 18 软件自带的仿真工具 Simulation Waveform Editor 进行仿真，其具体步骤如下。

1）创建仿真波形文件

（1）首先，打开设计项目。

（2）创建一个波形文件。选择 Quartus Prime 18 主界面的 File→New 子菜单，在弹出的对话框中找到验证与调试文件（Verification/Debugging Files）分类下的 University Program VMF，单击 OK 按钮打开仿真工具 Simulation Waveform Editor，将创建一个新的无标题波形文件，如图 3-10 所示。

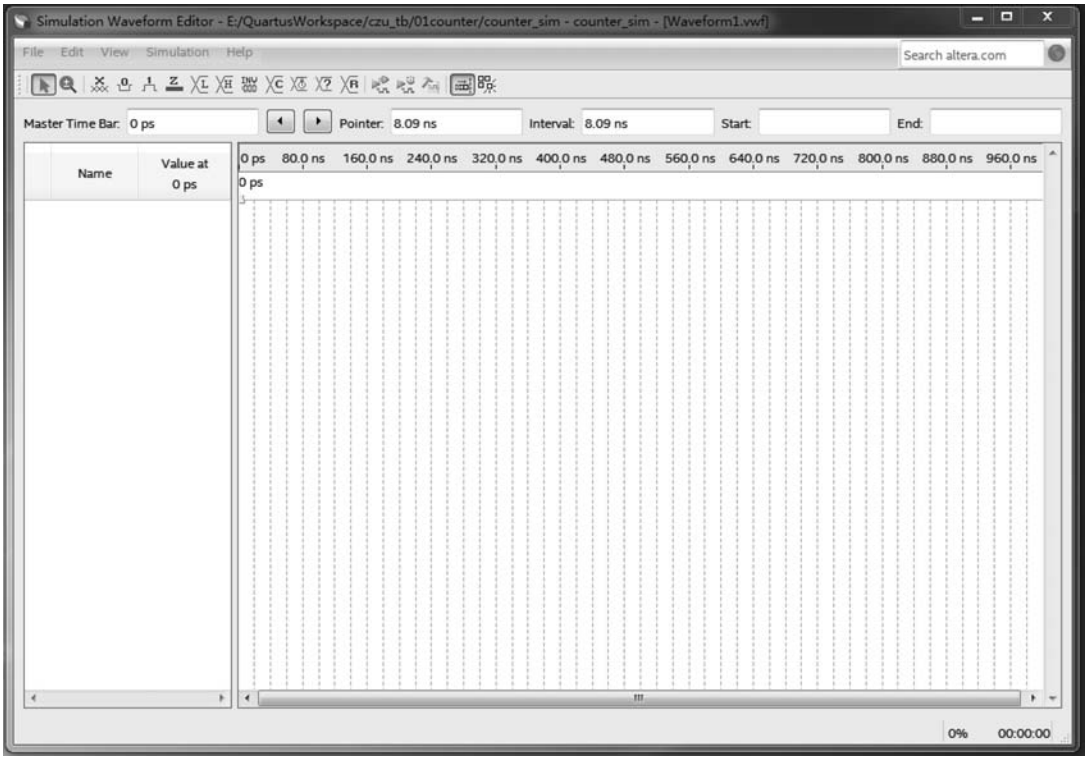


图 3-10　Simulation Waveform Editor 主界面

（3）存储波形文件。选择 Simulation Waveform Editor 主界面的 File→Save As 子菜

单，在 File Name 框中输入相应文件名，单击 OK 按钮存盘，文件后缀名为. vwf。

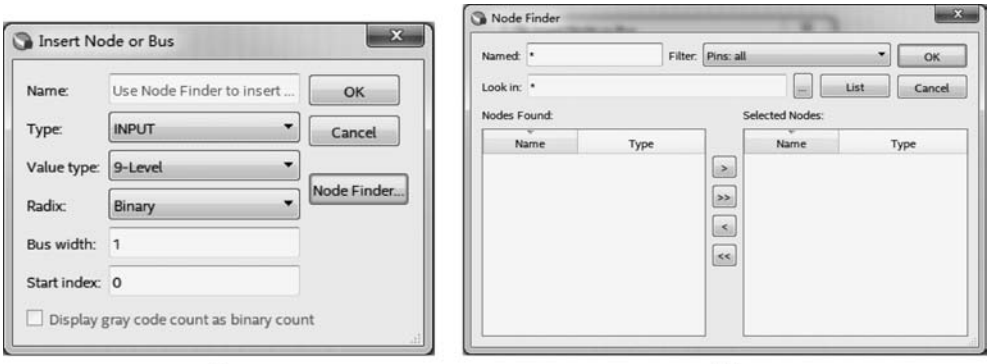
(4) 设定时间轴网格大小。选择 Simulation Waveform Editor 主界面的 Edit→Grid Size 子菜单，输入时间间隔(如 20ns)，单击 OK 按钮。通常用网格大小来表示在仿真过程中系统的最小单位时间。在对仿真波形文件中的输入时钟信号添加激励源时，对时钟的赋值是以网格时间为最小参考单位的，设计者只需填写时钟周期相对网格时间的倍数就行了。

(5) 设定时间轴长度。选择 Simulation Waveform Editor 主界面的 Edit→Set End Time 子菜单，并输入文件的结束时间，它决定在仿真过程中仿真器何时终止施加输入向量。对于比较简单的电路，取系统默认的仿真终止时间 1μs 就可以了，因为此时只需判断电路的逻辑功能关系是否正确。但对于一个复杂的电路而言，有时需要经过很多帧，这样，在进行时序仿真时就要设定较长的仿真时间。

2) 选择欲仿真的节点或总线

(1) 选择 Simulation Waveform Editor 主界面的 Edit→Insert→Insert Node or Bus 子菜单，出现如图 3-11(a)所示对话框。也可以在 Simulation Waveform Editor 主界面的左侧空白处右击，在弹出的快捷菜单中选择 Insert Node or Bus 选项，或者直接在该空白处双击鼠标左键。

(2) 在如图 3-11(a)所示对话框中，单击 Node Finder 按钮，弹出 Node Finder 对话框，如图 3-11(b)所示。



(a)　　(b)

图 3-11　节点输入对话框

(3) 在如图 3-11(b)所示对话框中选择要仿真的节点，先单击 List 按钮，在左边的 Nodes Found 列表中列出相关节点；如果单击 List 按钮后没有出现任何节点，则需重新选择 Filter 下拉框中的选项，Pin：all 表示只列出所有的输入/输出端子，此时可选择 Design Entity(all names)再单击 List 按钮。

(4) 在 Nodes Found 列表中选择需要仿真的节点，单击右移按钮(>)将它们移到右边的 Selected Nodes 列表中。

(5) 连续单击 Node Finder 对话框的 OK 按钮和 Insert Node or Bus 对话框的 OK 按钮，所要仿真的端子将出现在 Simulation Waveform Editor 主界面的左侧列表中，如图 3-12 所示。

如图 3-12 所示，所有未编辑的输入节点的波形都默认为逻辑低电平(0)，所有输出和隐

含节点波形都默认为未定义(×)逻辑电平。

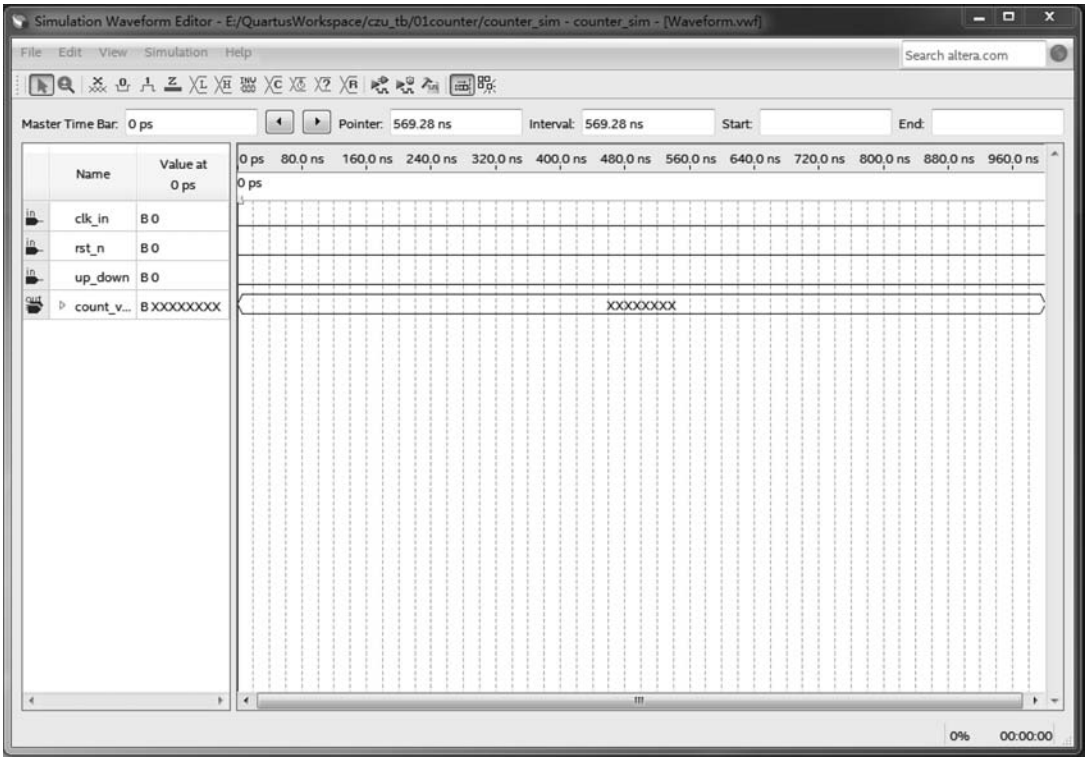


图 3-12　编辑仿真文件的端口和节点

3）编辑输入节点的仿真波形

首先介绍在波形编辑环境下，如图 3-10 所示的界面最左边常用控件按钮的功能。

：单击该按钮后，可以对选中的目标波形进行移动、剪切、复制、删除等操作。

：放大/缩小时间轴尺寸，单击放大，右击缩小。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的波形赋值为低电平（即逻辑“0”）。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的波形赋值为高电平（即逻辑“1”）。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的波形赋值为弱低电平。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的波形赋值为弱高电平。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的波形赋为不定态。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的波形赋为高阻态。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的波形进行逻辑取反操作。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的波形赋时钟信号。

：类似时钟赋值，先单击选择要编辑的波形，然后单击该按钮，可对选择的波形赋予指定周期的周期信号。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的总线赋任意值。

：先单击选择要编辑的波形，然后单击该按钮，可将选择的波形赋随机值。

：启动功能仿真。

：启动时序仿真。

：生成 ModelSim TestBench 文件(. vt)和脚本文件(. do)，". do"脚本文件可以在 ModelSim 中使用 do 命令调用。

将输入节点的某段用鼠标选中(变蓝)后，单击左边工具栏的有关按钮，即可进行低电平、高电平、任意、高阻态、反相和总线数据等各种设置。图 3-13 是进行节点波形输入的一个具体实例。

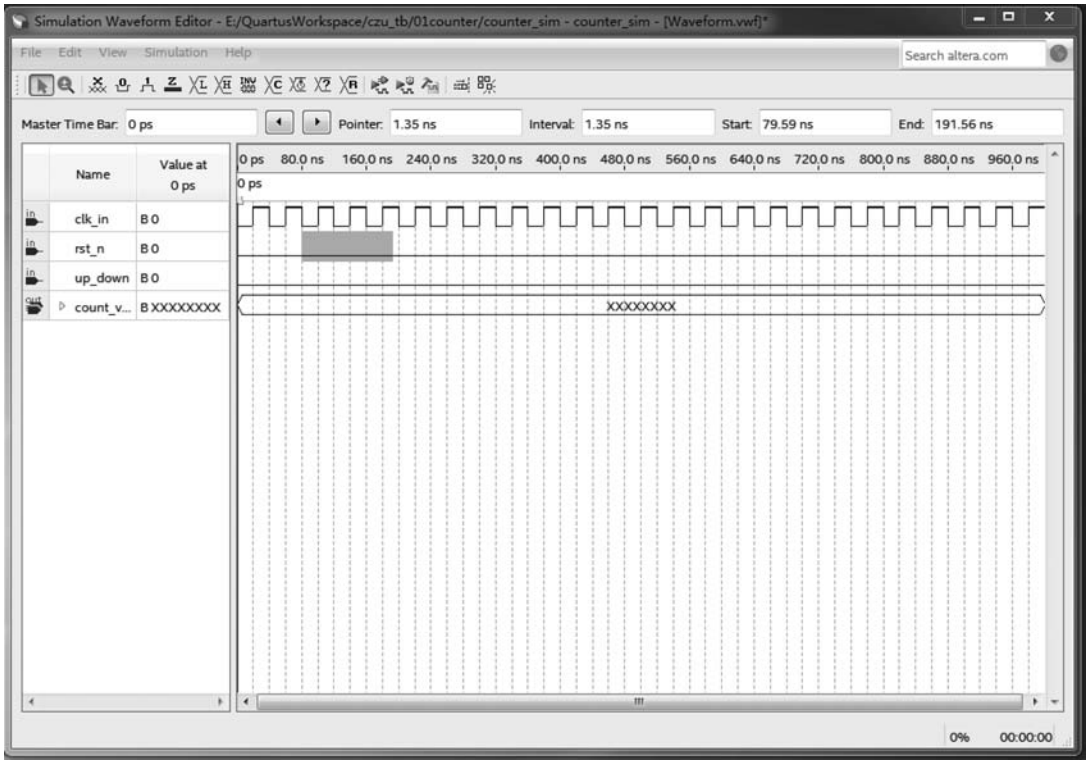


图 3-13　节点波形输入

4) 仿真

保存. vwf 文件后，在 Simulation Waveform Editor 主界面上单击工具栏上的 或 按钮，出现启动仿真进度条，如图 3-14 所示，若正确无误，将得到仿真波形。

5) 分析仿真结果

启动仿真后，若仿真过程各环节正确无误，将在另一个只读的 Simulation Waveform Editor 界面上得到仿真后的波形。在这里，主要观察输入和输出之间的逻辑关系是否符合设计要求。

2. 时序约束与时序分析

如前文所述，工程编译过程中已经包含时序分析(Timing Analysis)的环节，编译报告会给出时序分析的相关结果。如果工程中没有加任何时序约束，对于大部分设计来说，时序分析报告的分类标题通常会标红色，表示有时序方面的警告，如图 3-15 所示。这是因为，在默认情况下，Quartus Prime 软件会给所有没有被约束的时钟都设定为 1GHz 的时钟频率，

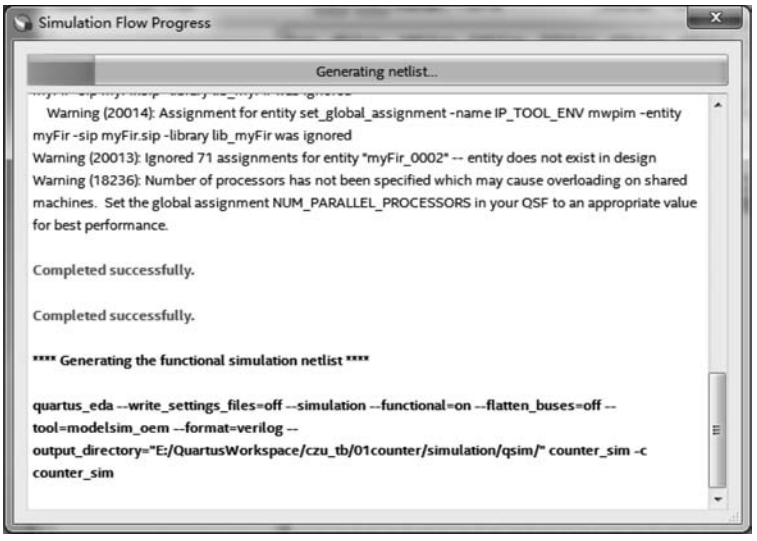


图 3-14　Simulation Waveform Editor 仿真器进度条窗口

所有的输入/输出的延迟都按 0 来计算。这显然不符合绝大多数设计的时序要求，所以有必要根据设计的特性，添加必要的时序约束。注意，在进行时序约束与分析之前，必须对顶层设计实体中的所有输入/输出端口进行引脚锁定，否则时序分析没有意义。

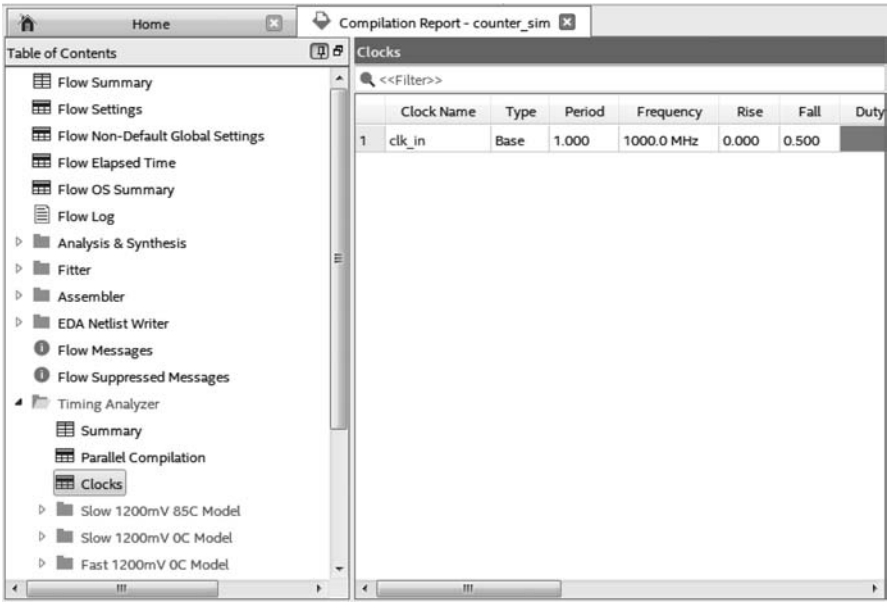


图 3-15　时序分析报告

时序分析支持 SDC（Synopsys Design Constraint）脚本输入，这种格式广泛引用于 ASIC 设计，它包含如下命令：Create_clock、Create_generated_clock、Set_input_delay、Set_output_delay、Set_false_path、Set_multicycle_path。每条命令都有详细的语法格式，用户不用去记住这些烦琐的语法，因为 Quartus Prime 的时序分析工具对每条命令都给出了图形

界面，用户只需要填写一些相应的参数，工具将自动生成对应的命令，而后可以写入 SDC 脚本文件。

在工程编译完成之后，可以利用 Quartus Prime 主界面上的 Tools→Timing Analyzer 来确定项目的性能。时序分析器是一个功能强大的、ASIC-style 的时序分析工具。采用 SDC 工业标准的约束、分析和报告方法来验证设计是否满足时序设计的要求。Timing Analyzer 主界面分成 5 个子窗口：Set Operating Conditions、Report、Task、Console、View panel，如图 3-16 所示。

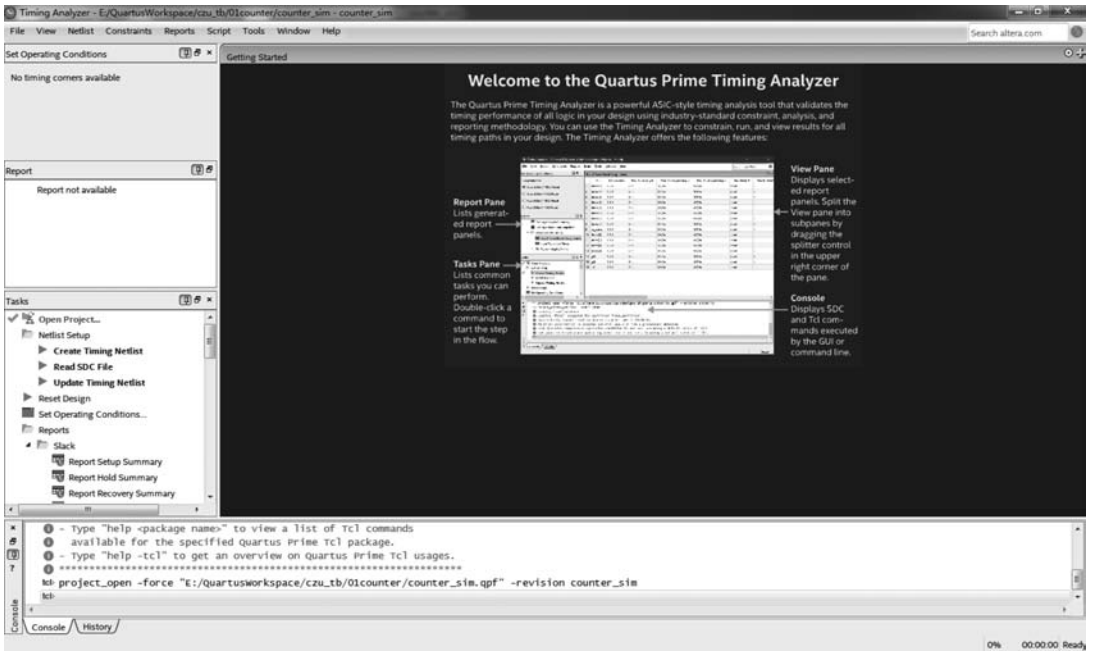


图 3-16　Timing Analyzer 主界面

在 Timing Analyzer 主界面上进行图形化时序约束的步骤如下。

（1）单击 Tasks 子窗口中的 Create Timing Netlist，或者从 Timing Analyzer 主界面上的 Netlist→Create Timing Netlist 创建时序网表；创建时序网表后，Set Operating Conditions 子窗口会显示若干个选项，它们分别表示如下含义。

① Slow 1200mV 85C Model：芯片内核供电电压 1200mV，工作温度 85℃情况下的慢速传输模型。

② Slow 1200mV 0C Model：芯片内核供电电压 1200mV，工作 0℃情况下的慢速传输模型。

③ Fast 1200mV 0C Model：芯片内核供电电压 1200mV，工作温度 0℃下快速传输模型。

（2）建立时钟约束，主要是给定时钟的频率、上升下降沿、占空比等参数，选择 Timing Analyzer 主界面上的 Constraints→Create Clock 子菜单，在弹出的对话框中进行设置，如图 3-17 所示。例如，设定 clk_in 时钟为 50MHz，占空比为 50%，关联顶层设计实体中的 clk_in 端口，SDC 脚本在 SDC command 文本框中同步生成，设定完成后点击 Run 按钮。

（3）设定输入/输出延时，选择 Constraints→Set Input Delay 子菜单，图 3-18 为复位输入信号的输入延时设定对话框，设定输入延时为 1ns；同样地，利用 Set Output Delay 设定

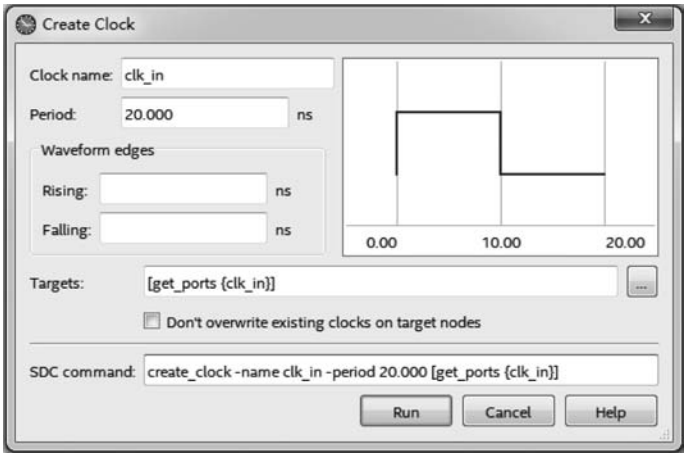


图 3-17　Create Clock 对话框

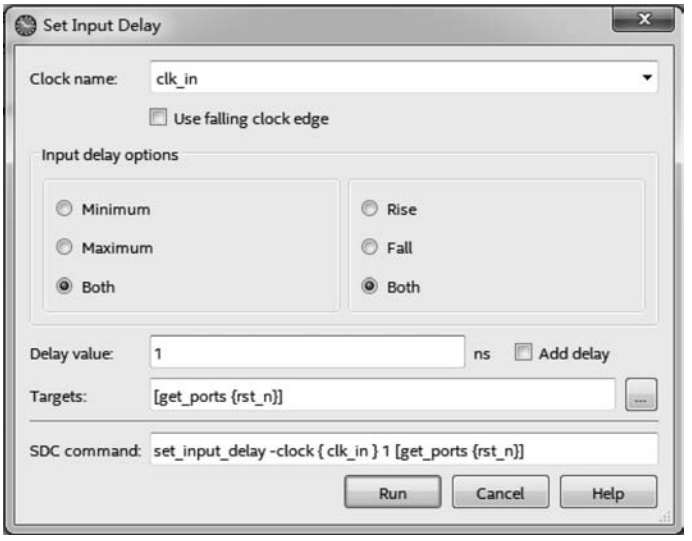


图 3-18　Set Input Delay 对话框

各输出端口的输出延时。

（4）对时钟和输入/输出延时约束设定完成后，选择 Constraints→Write SDC File 子菜单，指定文件名（后缀为. sdc），将上述约束项保存到文件中。SDC 文件为文本文件，可通过记事本或写字板等文本编辑查看写入 SDC 文件中的约束命令。

（5）通过 Tasks 子窗口中的 report 查看时序分析结果，如 Report→Slack 下的建立、保持时间梗概，Report→Datasheet 下的最高频率（Fmax）梗概，以及 Report→Custom Reports→Create Slack Histogram，选择 clk_in 后显示如图 3-19 所示的余量（Slack）直方图。

图 3-19 所示的直方图中，横轴为余量值，纵轴为路径数，该图中的 Slack 全是正值，表示在 clk_in 时钟下没有不满足约束的路径，若有 Slack 为负值的情况，则对应的纵向方块将用红色显示。

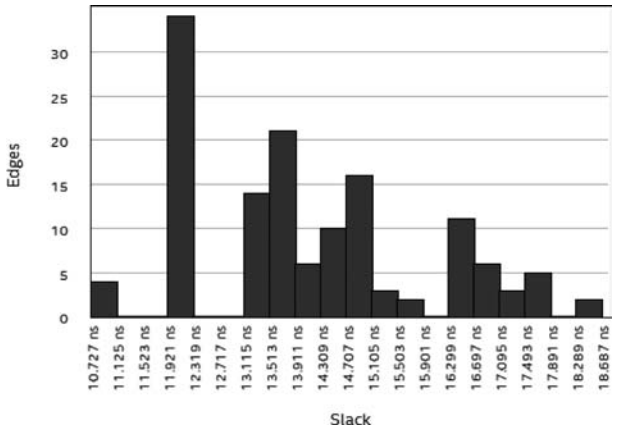
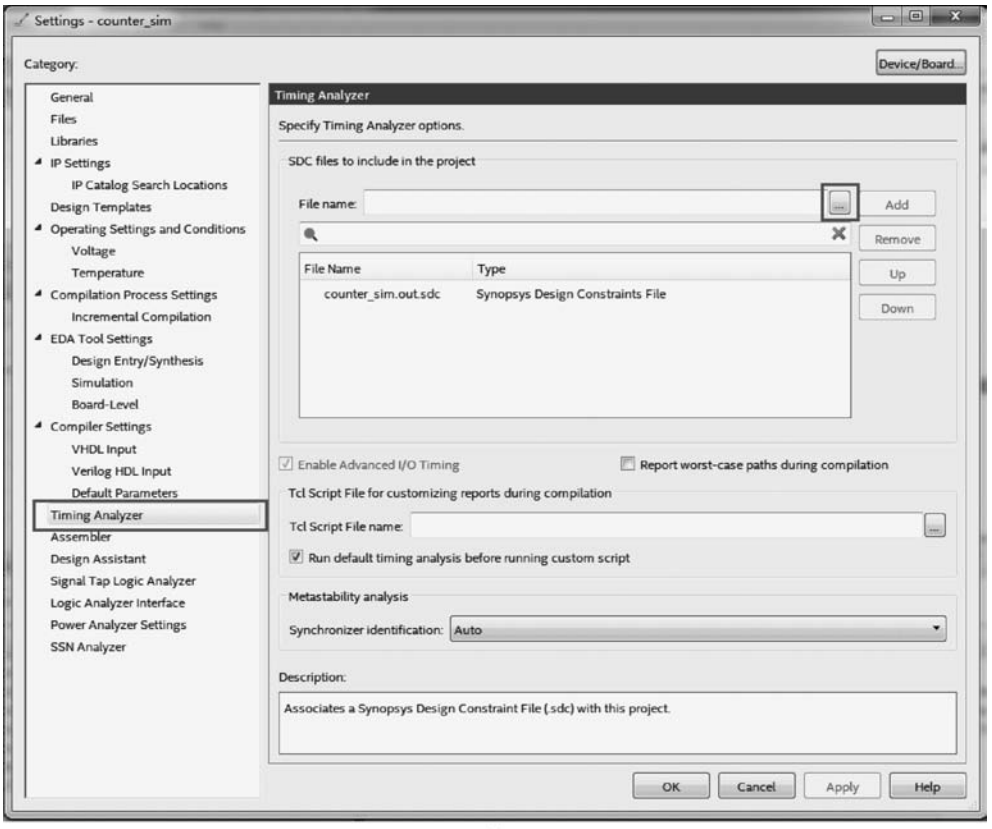


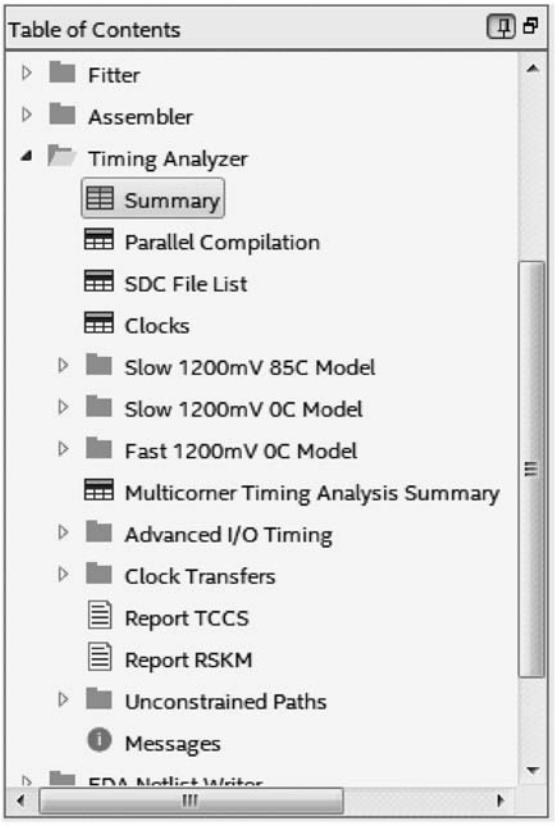
图 3-19　时序分析报告中的余量直方图

(6) 完成时序约束后，通过 Quartus Prime 主界面 Assignments→Settings 对话框中的 Timing Analyzer 选项指定.sdc 约束文件，如图 3-20(a)所示，再重新编译，即可在编译报告中看到“Timing Analyzer”不再显示红字，说明完全满足约束条件，如图 3-20(b)所示。



(a)

图 3-20　指定约束文件后时序分析检查全部通过



(b)

图 3-20 （续）

3.1.7 器件编程

编程是指将编程数据放到具体的可编程器件中去。当成功编译和仿真一个项目后，可以对一个器件进行编程并在实际电路中进行测试。每次上电后需要进行编程配置是基于SRAM工艺FPGA的一个特点，在FPGA内部有许多可编程的多路器、互连线节点和RAM初始化内容需要配置数据来控制。FPGA中的配置RAM就用来存放配置数据的内容。常利用USB-Blaster下载器和Quartus Prime编程器（Programmer）完成对FPGA器件的编程工作。

1. 项目编译

在编译过程中，Assemble将自动生成一个SRAM目标文件，此文件用于为某目标器件在系统编程，由于SRAM具有掉电后内容丢失的缺点，为了上电后无须人为干预即能自动配置FPGA，通常在电路板设计时，在FPGA的Altera专用的串行配置接口上连接一片非易失性的存储器，如EPROM芯片。若要给连接FPGA目标器件的EPROM芯片配置程序，需利用File→Convert Programming File子菜单，将根据已有的.sof文件手动生成一个编程目标文件（.pof），如图3-21所示。

需要进行如下几步操作来完成.pof文件的生成。

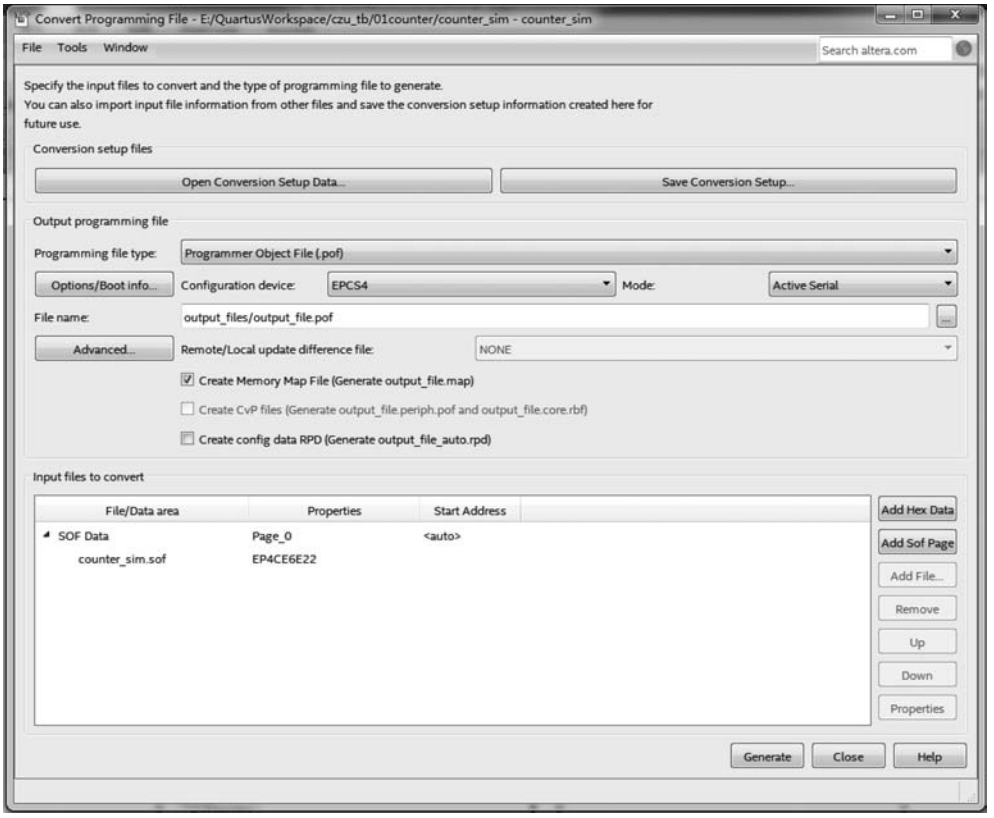


图 3-21　生成. pof 文件用于 EPROM 配置

(1) 在 Programming file type 选项中选择 Programmer Object File(. pof)。

(2) 在 Configuration device 选项中选择 PCB 上使用 EPROM 芯片,File name 文本框中自动生成路径和 output_file. pof 文件名。

(3) 在最下方的 Input files to convert 栏中,单击 SOF Data,然后单击右侧的 Add File 按钮选中在本工程路径\output_files 下已生成的. sof 文件。

(4) 最后单击右下方的 Generate 按钮,将弹出消息框完成. pof 文件的生成。

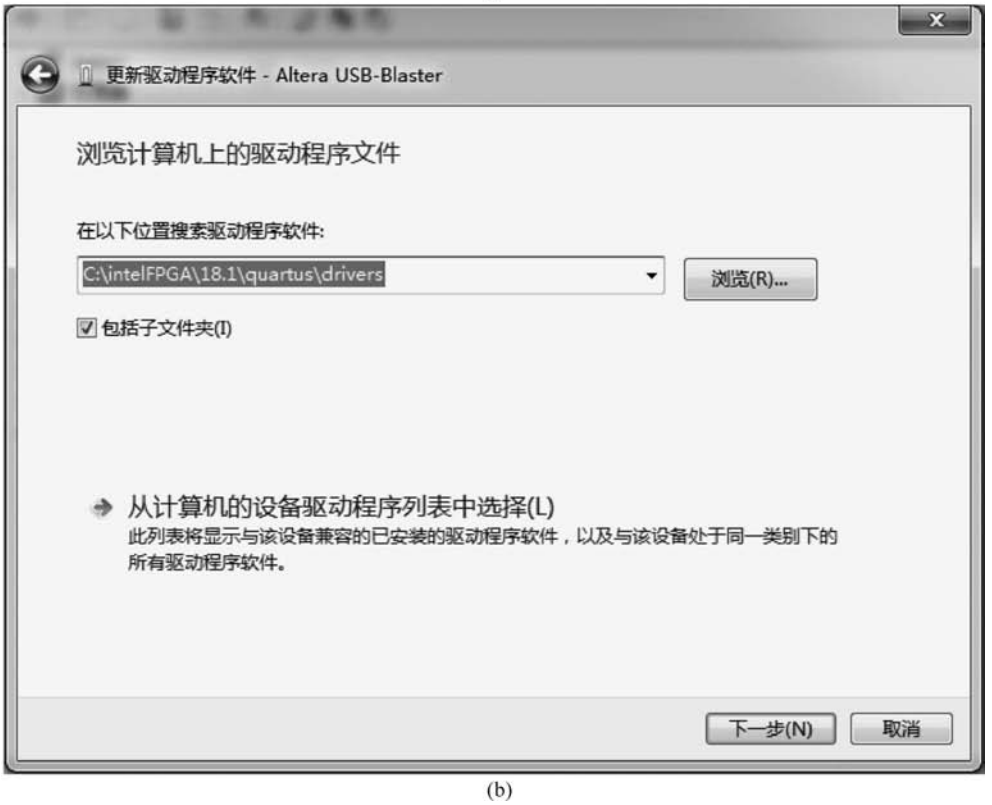
2. 安装 USB-Blaster 下载器驱动程序

将 USB-Blaster 下载器电缆一端安装在计算机 USB 接口上,USB-Blaster 下载器另一端的双排针(5×2)插头安装在装有可编程器件 PCB 板的相应物理接口上,PCB 板上的物理接口通常有金手指缺口,避免排针接反。PCB 板上用于 USB-Blaster 下载器的常见接口有 JTAG 和主动串行(AS)接口,JTAG 口用于对 FPGA 的 SRAM 在系统编程,下载. sof 文件; AS 接口则用于对 EEPROM 芯片进行配置,下载. pof 文件。USB-Blaster 下载器电缆第一次连接计算机 USB 口时,操作系统将提示发现新硬件,如图 3-22(a)所示。此时,需手动安装 USB-Blaster 下载器的驱动程序,将其驱动程序的路径指定为如图 3-22(b)所示文件夹。

3. 打开编程器

选择 Quartus Prime 主界面上的 Tools→Programmer 子菜单,或单击工具栏上的 按

(a)



(b)

图 3-22　USB-Blaster 安装驱动程序

钮，打开编程器，如图 3-23 所示。编程模式默认为 JTAG 方式，编程文件默认为在系统编程的.sof 文件。

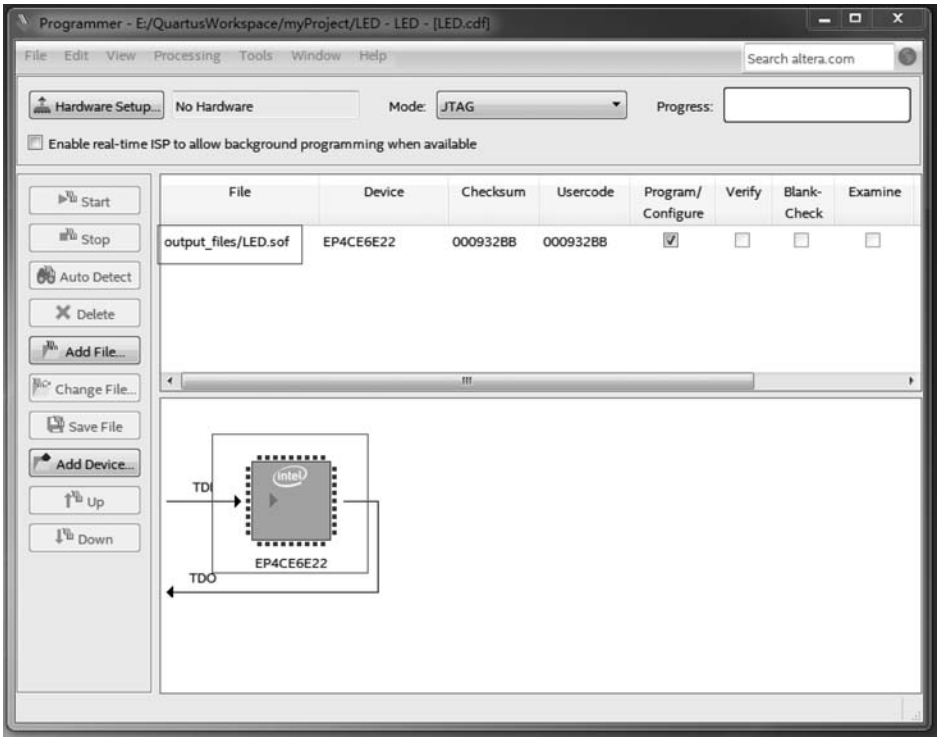


图 3-23　Quartus Prime 编程器窗口

编程器窗口中常用的功能说明如下。

Program/Configure：将一个编程文件中的数据编程到一个 FPGA 或 EPROM 器件中。

Verify：校验器件中的内容是否与当前编程数据内容相同。

Blank-Check：检查器件是否是空的或者已被擦除。

Examine：从器件中读取编程数据，勾选此选项时，其他选项均不能使用。

Security Bit：防止器件被读取编程数据或被再次编程，此选项仅针对 MAX3000 和 MAX7000 系列器件可用。

Erase：擦除 EPROM 器件中的数据。

ISP CLAMP：在系统编程时将所有 I/O 口钳制于静态状态。

4. 选择编程下载器

单击左上角的 Hardware Setup 按钮选择下载器，将出现 Hardware Setup 对话框，如图 3-24 所示，在该对话框中的 Currently selected hardware 下拉列表中选择 USB-Blaster [USB0]后，单击 OK 按钮。一台计算机可以连接多个 USB-Blaster 下载器。

5. 用 JTAG 在系统编程

将 USB-Blaster 下载器连接到 PCB 上的 JTAG 物理接口，完成 Hardware 选择以后，单击 Programmer 界面上的 Start 按钮，即可开始在系统编程，如图 3-25 所示。当 Progress 显示 100%时表示编程成功。

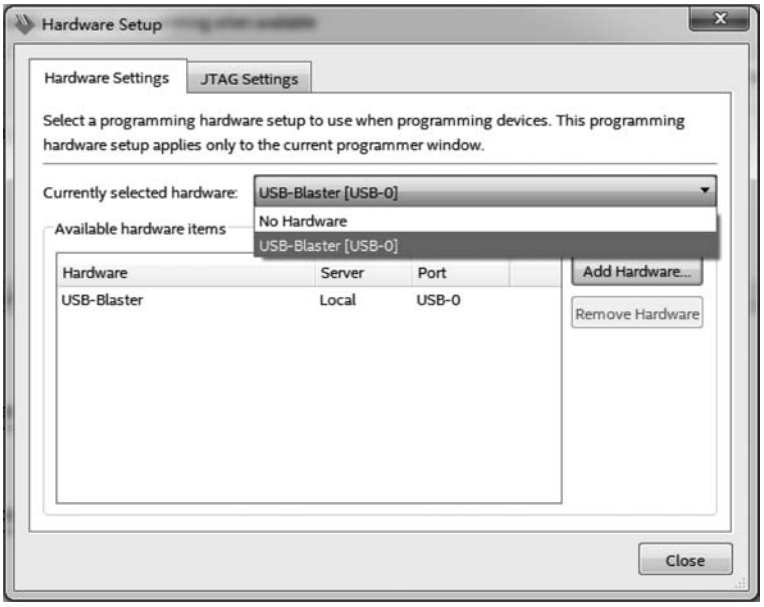


图 3-24　设定编程硬件对话框

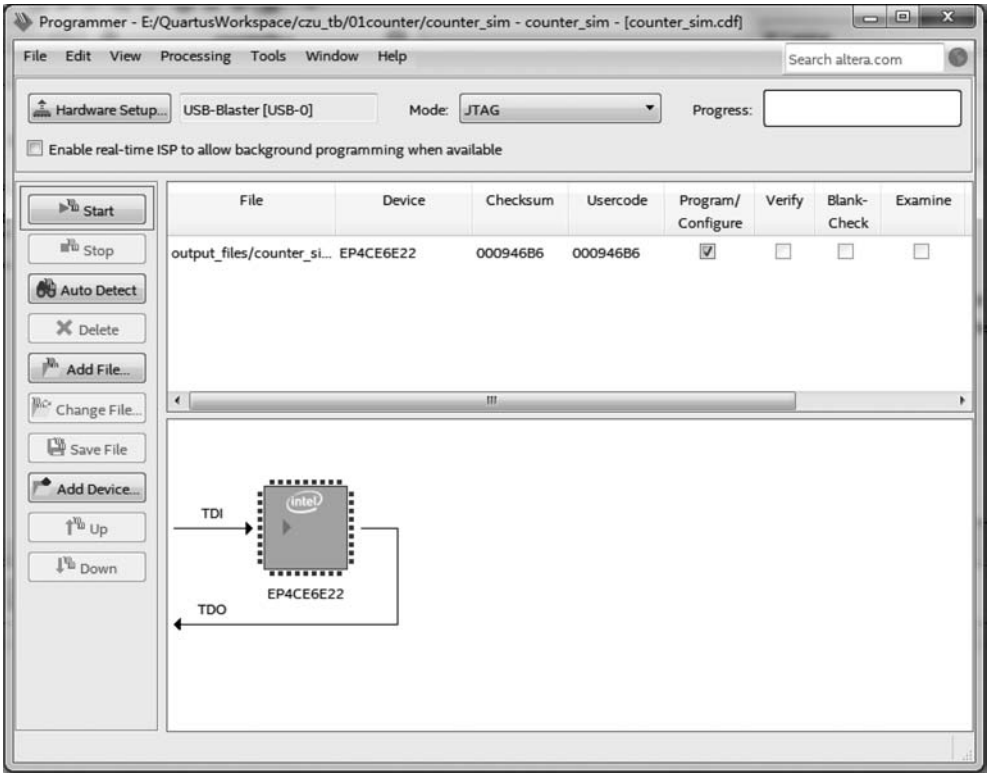


图 3-25　启动 JTAG 在系统编程

6. JTAG 口间接编程(用于 EPROM 芯片的编程配置)

由于 AS 直接模式下载设计文件时,需要复杂的保护电路,为了简化电路,省去 AS 物理接口,利用 JTAG 口将 FPGA 作为中转站也可以对 FPGA 上外挂的 EPROM 芯片进行编程配置,以实现 FPGA 上电自动配置的功能。为此,需首先根据.sof 文件生成 JTAG 间接配置文件(.jic)。同样地,先选择 File→Convert Programming File 子菜单,在弹出的对话框中进行如下设置,如图 3-26 所示。

(1) 在 Programming file type 选项中选择 JTAG Indirect Configuration File(.jic)。

(2) 在 Configuration device 选项中选择 PCB 上使用 EPROM 芯片,File name 文本框中自动生成路径和 output_file.jic 文件名。

(3) 在最下方的 Input files to convert 栏中,单击 Flash Loader,然后单击右侧的 Add Device 按钮,在弹出的 Select Devices 对话框中选中本工程中使用的 FPGA 芯片基本型号。

(4) 在最下方的 Input files to convert 栏中,单击 SOF Data,然后单击右侧的 Add Files 按钮,在弹出的 Select Input File 对话框中选中在本工程路径\output_files 下已生成的.sof 文件。

(5) 在最下方的 Input files to convert 栏中,单击在 SOF Data 下方已添加的.sof 文件,然后单击右侧的 Properties 按钮,在弹出的 SOF Properties 对话框中勾选 Compression,单击 OK 按钮,实现文件压缩功能。

(6) 最后单击右下方的 Generate 按钮,将弹出消息框完成.jic 文件的生成。

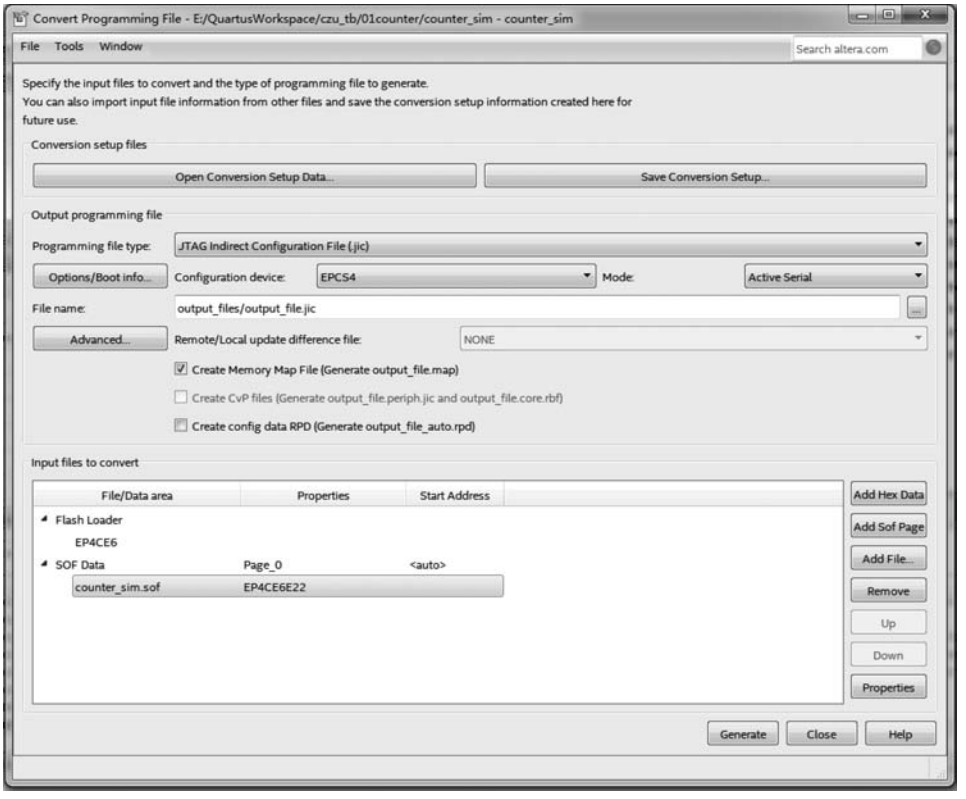


图 3-26　生成 JTAG 间接配置文件

生成.jic文件后，启动工具栏上的Programmer编程器，在Programmer界面上进行如下设置，如图3-27所示。

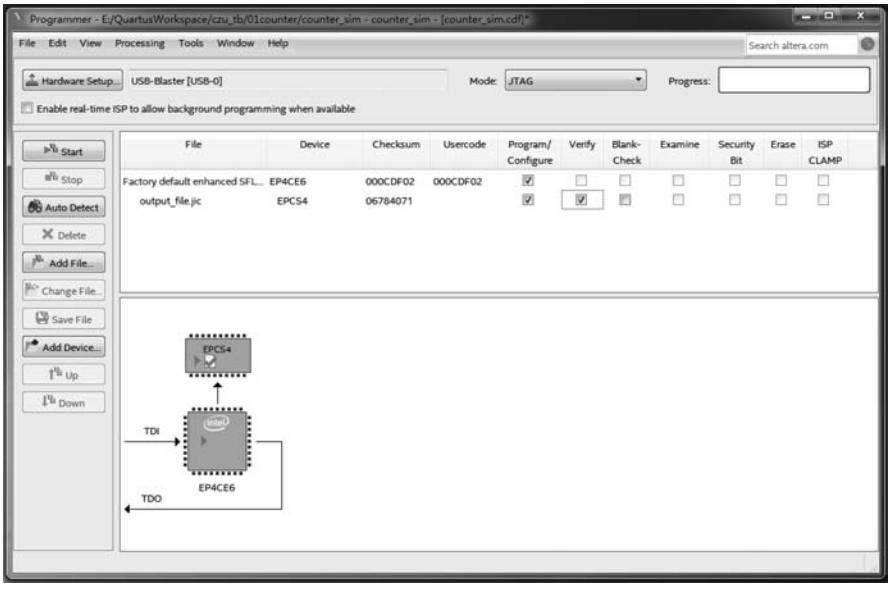


图3-27　JTAG间接配置文件的编程

（1）删除默认的.sof文件，鼠标左键选中.sof文件，单击左侧的Delete按钮。

（2）添加.jic文件，单击左侧Add File按钮，选中刚刚生成的.jic文件，添加后在界面下方会自动显示FPGA外挂EPROM的框图。

（3）勾选output_file.jic所在行的Program/Configure和Verify，然后单击左侧Start按钮。下载完成后，界面右上角的进度(Progress)会显示100%。

（4）可选步骤：选择File→Save子菜单，以将当前状态保存为.cdf文件，下次再打开Programmer工具时不用再重复以上操作。

3.2　1位全加器设计

通过3.1节的介绍，对原理图设计方法有了一定的了解，下面通过一个1位全加器的实例，进一步介绍原理图设计方法。1位全加器可以用两个半加器及一个或门连接而成，因此需要首先完成半加器的设计。以下将给出使用原理图输入的方法进行半加器底层元件设计和层次化设计全加器的主要步骤与方法，其主要流程与数字系统设计的一般流程基本一致。

3.2.1　建立文件夹

假设本项设计的文件夹取名为MY_PRJCT，在E盘中，路径为E:\MY_PRJCT。

3.2.2　输入设计项目和存盘

（1）打开Quartus Prime 18，首先利用新工程向导File→New Project Wizard新建工程。其次，选择File→New子菜单，在弹出对话框中选择框图/原理图文件(Block Diagram/

Schematic File)，单击 OK 按钮后将打开原理图编辑窗口。

(2) 在原理图编辑窗口中分别调入元件 1 个 and 与门、1 个 xor 异或门、2 个 input 端子和 2 个 output 端子，并按图 3-28 连接好。然后用鼠标分别在 input 和 output 的 PIN-NAME 上双击使其变蓝色，再用键盘分别输入各引脚名：a、b、so 和 co，如图 3-28 所示。

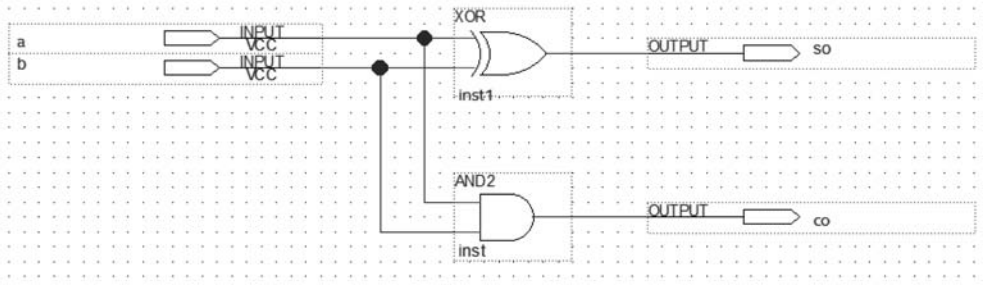


图 3-28　半加器 h_adder 原理图

(3) 选择 File→Save As 子菜单，将已设计好的原理图文件取名为 h_adder.bdf，该文件将自动保存在刚才建立的目录 E:\MY_PRJCT 下，并选择 Project→Add Current File to Project 将该原理图文件加入本工程。

3.2.3　将设计项目设置成工程文件

选择 Project→Set As Top-Level Entity 将设计项目 h_adder.bdf 设定为工程的顶层设计实体，此时工程导航区中 FPGA 器件型号下的顶层设计名会变更为 h_adder。

3.2.4　选择目标器件并编译

在新建工程时的新工程向导中就可以设定所选择的 FPGA 器件，新建完工程以后若想修改 FPGA 器件，可以选择 Assignments→Device 子菜单，或者直接双击在器件工程导航区中的 FPGA 器件型号，在弹出的对话框中更改 FPGA 器件型号。目标器件设定后，启动编译器，单击工具栏上的 ▶ 编译工具，或选择 Processing→Start Compilation 子菜单，此编译器的功能包括分析与综合、布局布线、生成编程文件、时序分析、生成 EDA 网表等所有环节。

3.2.5　时序仿真

接下来测试设计项目的正确性，即逻辑仿真，具体步骤如下。

(1) 建立波形测试文件。选择 File→New 子菜单，再选择 New 对话框中的 University Program VWF 项，打开 Simulation Waveform Editor 波形编辑窗口。

(2) 输入信号节点。在波形编辑窗口的 Edit→Insert→Insert Node or Bus 下拉菜单中单击 Node Finder 按钮，选择所需仿真的节点。在弹出的窗口中单击 List 按钮，这时左栏中将列出该项设计所有的引脚名称。利用中间的“>>”按钮将所有查找到的引脚添加到右栏中，然后单击 OK 按钮即可，如图 3-29 所示。

(3) 设置波形参量。如图 3-29 所示的波形编辑窗口中已经调入了半加器的所有节点信号，在为编辑窗口的半加器输入信号 a 和 b 设定必要的测试电平之前，首先设定相关的仿

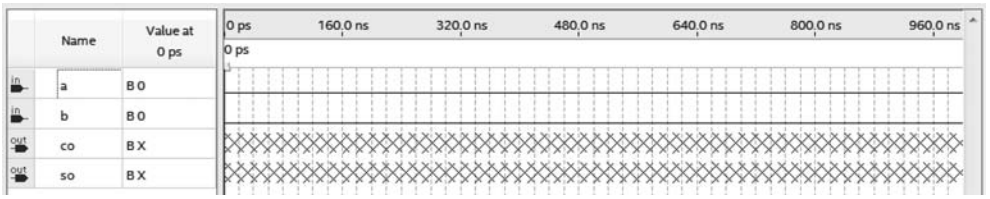


图 3-29　输入信号节点

真参数。在 Options 选项中取消勾选 Snap to Grid，以便能够任意设置输入电平位置，或设置输入时钟信号的周期。

（4）设定仿真时间宽度。选择 File 项及其 End time 选项，在 End time 选择窗口中选择适当的仿真时间域，如可选 4μs，以便有足够长的观察时间。

（5）加上输入信号。现在可以为输入信号 a 和 b 设定测试电平了。如图 3-30 中标出的那样，利用必要的功能键为 a 和 b 加上适当的电平，单击 时序仿真工具，以便仿真后能测试 so 和 co 输出信号。

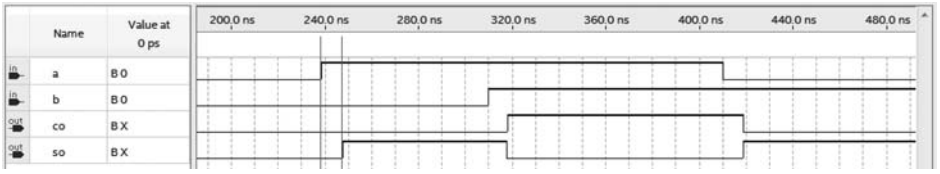


图 3-30　半加器 h_adder. bdf 的仿真波形

（6）仿真波形出现后，Simulation Waveform Editor 会给波形文件自动存盘，可以在工程文件夹中的\simulation\qsim 子文件夹中找到以“工程名＋当前系统时间. sim. vwf”方式命名的文件。

（7）观察分析波形。可以看出，图 3-30 显示的半加器的时序波形是正确的。还可以进一步了解信号的延时情况。图 3-30 中的两条竖线之间的时间间隔就是输入与输出波形间的延时量，延时大概在 10ns 左右。

为了精确测量半加器输入与输出波形间的延时量，可打开时序分析器，方法是选择 Tools→ Timing Analyzer 选项，双击 Tasks 子窗口中的 Reports→ Datasheet→ Report Datasheet 项，传输延时信息即刻显示在图表中，如图 3-31 所示。其中，RR 表示从上升沿到上升沿的最大延时，RF 表示从上升沿到下降沿的最大延时，FF 表示从下降沿到下降沿的最大延时，FR 表示从下降沿到上升沿的最大延时。

Propagation Delay

	Input Port	Output Port	RR	RF	FR	FF
1	a	co	9.030			9.116
2	a	so	10.446	10.348	10.638	10.617
3	b	co	8.722			8.791
4	b	so	10.076	9.995	10.303	10.215

图 3-31　延时时序分析窗口

（8）包装元件入库。在原理图文件 h_adder. bdf 打开的情况下，选择 File→ Create/

Update→Create Symbol Files for Current File 子菜单，生成 h_adder.bsf 图标文件，即将当前文件变成了一个包装好的单一元件，并被放置在当前工程路径指定的目录中，这样就可以在其他设计文件中调用 h_adder。

3.2.6　引脚锁定

如果以上的仿真测试正确无误，就应该将所进行的设计下载进选定的目标器件中，如目标器件为 EP4CE6E22C8，做进一步的硬件测试，以便最终了解设计项目的正确性。这就必须根据评估板、开发电路系统或 EDA 实验板的要求对设计项目输入/输出引脚赋予确定的引脚，以便能够对其进行实测。这里假设根据实际需要，要将半加器的 4 引脚 a、b、co 和 so 分别与目标器件 EP4CE6E22C8 的第 10、11、30 和 31 脚相接，操作如下。

(1) 选择 Assignments→Pin Planner 子菜单。

(2) 在 Pin Planner 窗口下方的 All pins 列表的 Location 项中将 a、b、co 和 so 分别设定为 PIN_10、PIN_11、PIN_30 和 PIN_31 引脚号，既可以手动输入，也可以从下拉列表中选定，如图 3-32 所示。

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate
in a	Input	PIN_10	1	B1_N0	PIN_10	2.5 V		8mA (default)	
in b	Input	PIN_11	1	B1_N0	PIN_11	2.5 V		8mA (default)	
out co	Output	PIN_30	2	B2_N0	PIN_30	2.5 V		8mA (default)	2 (default)
out so	Output	PIN_31	2	B2_N0	PIN_31	2.5 V		8mA (default)	2 (default)

图 3-32　半加器引脚锁定

(3) 特别需要注意的是，在锁定引脚后，必须再通过 Quartus Prime 18 的 Compiler 选项对工程重新编译一次，以便将引脚信息编入下载文件中。

3.2.7　编程下载

引脚锁定并重新编译后，就可进行编程下载，具体步骤如下。

(1) 用 USB-Blaster 的下载电缆插入计算机的 USB 口，USB-Blaster 的 JTAG 口(5×2 排针)与目标板连接好，并打开目标板电源，注意，为了更好地保护 FPGA 的 JTAG 口不被烧坏，应先插 JTAG 口，再给目标板上电。

(2) 选择 Quartus Prime 18 工具栏上的 Programmer 编程器工具，弹出编程器窗口，然后单击 Programmer 窗口左上角的 Hardware Setup 按钮，在 Hardware 设定下拉菜单中选择 USB-Blaster[USB0]。

(3) 单击 Programmer 编程窗口中自动出现的 *.sof 文件，并单击窗口左侧上方的 Start 按钮，向 EP4CE6E22C8 在系统下载配置文件，如果连线无误，程序下载完成后，窗口右上角的 Progress 会显示绿色的 100%。

3.2.8　设计顶层文件

可以将前面的工作看成是完成了一个底层元件的设计和功能检测，并被包装入库。现在利用已设计好的半加器，完成顶层项目全加器的设计，详细步骤可参考以下设计流程。

(1) 在原工程基础上，新建一个原理图文件，然后向新原理图中添加两个半加器元件 h_adder 和一个 2 输入端的或门 or2。这时，如果双击已添加的半加器元件 h_adder，即可弹

出半加器元件内部的原理图。

（2）完成全加器原理图设计，如图 3-33 所示，并以文件名 f_adder. bdf 存在同一工程目录中。

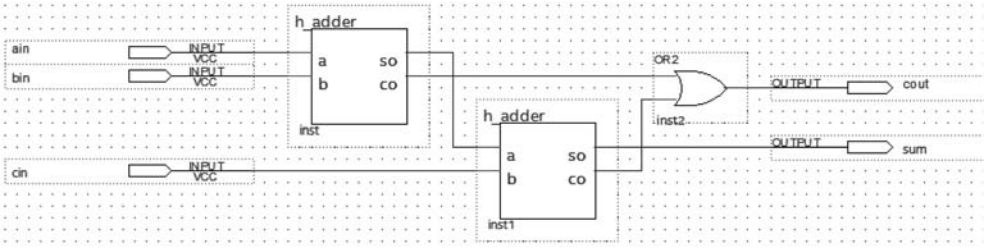


图 3-33　在顶层编辑窗口中设计全加器

（3）选择 Project→Add Current File to Project 子菜单，将当前文件 f_adder. bdf 加入 Project。

（4）选择 Project→Set As Top-Level Entity 子菜单，将 f_adder. bdf 设为顶层设计文件。

（5）重新编译工程，编译无误后建立波形仿真文件。

（6）对应 f_adder. bdf 的波形仿真文件如图 3-34 所示，参考图中设置输入信号 ain、bin 和 cin 的波形，启动功能仿真，观察输出波形的情况。

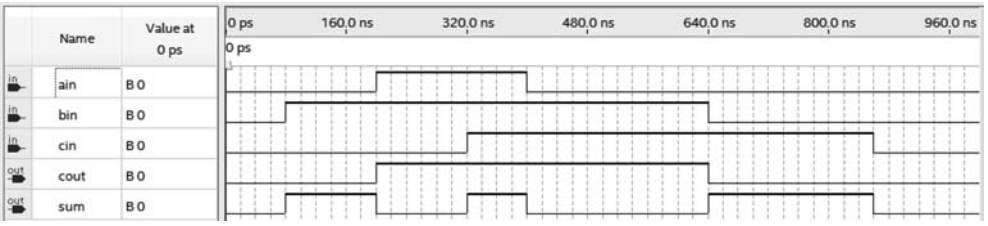


图 3-34　1 位全加器的时序仿真波形

（7）锁定引脚、编译并编程下载，可以硬件实测此全加器的逻辑功能。

3.3　数字电子钟设计

数字电子钟为计数器的综合应用，数字电子钟的秒针部分由六十进制计数器组成，分针部分也由六十进制计数器所组成。时针部分则可分为两种情况，12 小时制的为十二进制计数器，24 小时制的则为二十四进制计数器，在本例中采用十二进制计数器，分别说明如下。

3.3.1　六十进制计数器设计

1. 六进制计数器设计

要构成六十进制计数器，需要应用十进制计数器和六进制计数器，十进制计数器在基本的元件库中可以找到，而六进制计数器在基本的库中没有，所以首先介绍用 D 触发器设计具有使能与预置功能的六进制计数器。当使能输入端“en”为“1”时，计数器开始计数，当使

能输入端“en”为“0”时，计数器停止计数，保持原值。将具有使能功能的六进制计数器配合多路选择器的运用，可设计出含同步预置功能的六进制计数器，当预置控制端“load”为“0”时，会将输入数据送至触发器输入端，当预置控制端“load”为“1”时，计数器会停止预置。此计数器另有一串接进位端“co”可供多个计数器串接时进位使用。

1）数据选择器设计

数据选择器是一种数据处理的逻辑电路，可以在许多输入数据中选取一个并将它送至单一的输出线上。它主要分为三部分：控制线，数据线与输出线。例如，16 对 1 的数据选择器有 4 条控制线，16 条数据线，1 条输出线。在此，对 2 选 1 的数据选择器进行介绍。

2 选 1 的数据选择器的输入/输出引脚如下。

控制线 1 条定义为 s；数据输入线 2 条定义为 d0，d1；数据输出线 1 条定义为 y；其真值表如表 3-1 所示。

表 3-1　2 选 1 数据选择器真值表

控制线	输出线
s	y
0	d0
1	d1

新建一原理图文件，双击原理图空白处，添加参数化 MUX 元件，MUX 元件如图 3-35 所示，参数可重新配置，WIDTH 为输入数据端 data[]的位宽，WIDTHS 为输入选择端 sel[]的位宽。双击 Parameter 列表里的参数即可在弹出的对话框中设定参数。这里，只需设定 data[]的位宽为 2，sel[]的位宽通过 LOG2(WIDTH)自动算得为 1，再添加 d[1..0]和 s 的 input 端口和 y 输出端口即可构成 2 选 1 数据选择器。将该原理图文件保存为 mux2.bdf，并选择 File→Create/Update→Create Symbol File for Current File 子菜单为当前设计文件建立图标文件，即可在其他设计文件中调用 mux2 元件。

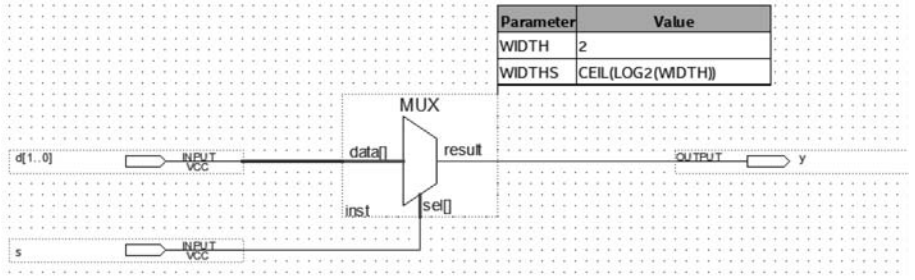


图 3-35　位宽可参数化的 2 选 1 数据选择器电路图

如果所需的数据选择器的数据输入端只有 1b 位宽，那么在原理图中可以直接添加库中“21mux”元件，21mux 的数据输入端为 A 和 B，选择端为 S，输出端为 Y，请读者自行尝试应用。

2）六进制计数器的真值表

六进制计数器的输入/输出引脚介绍如下。

脉冲输入端：clk。清除控制端：clrn。预置控制端：load。使能端：en。预置输入端：d_2、d_1、d_0。输出端：q_2、q_1、q_0。串接进位端 co。其真值表如表 3-2 所示。

表 3-2　六进制计数器真值表

上周期输出			控制线				输入值			输出		
q_2	q_1	q_0	clk	clrn	load	en	d_2	d_1	d_0	q_2	q_1	q_0
×	×	×	×	0	×	×	×	×	×	0	0	0
×	×	×	↑	1	0	×	a	b	c	a	b	c
q_2	q_1	q_0	↑	1	1	0	×	×	×	q_2	q_1	q_0
0	0	0	↑	1	1	1	×	×	×	0	0	1
0	0	1	↑	1	1	1	×	×	×	0	1	0
0	1	0	↑	1	1	1	×	×	×	0	1	1
0	1	1	↑	1	1	1	×	×	×	1	0	0
1	0	0	↑	1	1	1	×	×	×	1	0	1
1	0	1	↑	1	1	1	×	×	×	0	0	0

3）六进制计数器设计

在此利用 D 触发器设计，先设计含有使能输入的同步六进制计数器，再与 2 选 1 的多路选择器组合成含有预置与使能功能的六进制计数器。利用数字电路设计方法可设计出各触发器的 D 输入端的驱动方程分别为：

$$d_2 = E_n Q_1 Q_0 + \overline{E_n} Q_2 + Q_2 \overline{Q_0}$$

$$d_1 = Q_1 \overline{Q_0} + \overline{E_n} Q_1 + E_n \overline{Q_2} \cdot \overline{Q_1} Q_0$$

$$d_0 = E_n \overline{Q_0} + \overline{E_n} Q_0$$

$$\mathrm{co} = Q_2 Q_0 E_n$$

根据以上驱动方程可设计出如图 3-36 所示的电路图。图中很多连线使用标注的方式进行连接，如 clk、load、clrn 等。另外，mux2 的总线型输入端可以组合输入，如使用“d0，dx[0]”的方式，表示 mux2 的输入端 d[1..0]={d0，dx[0]}。

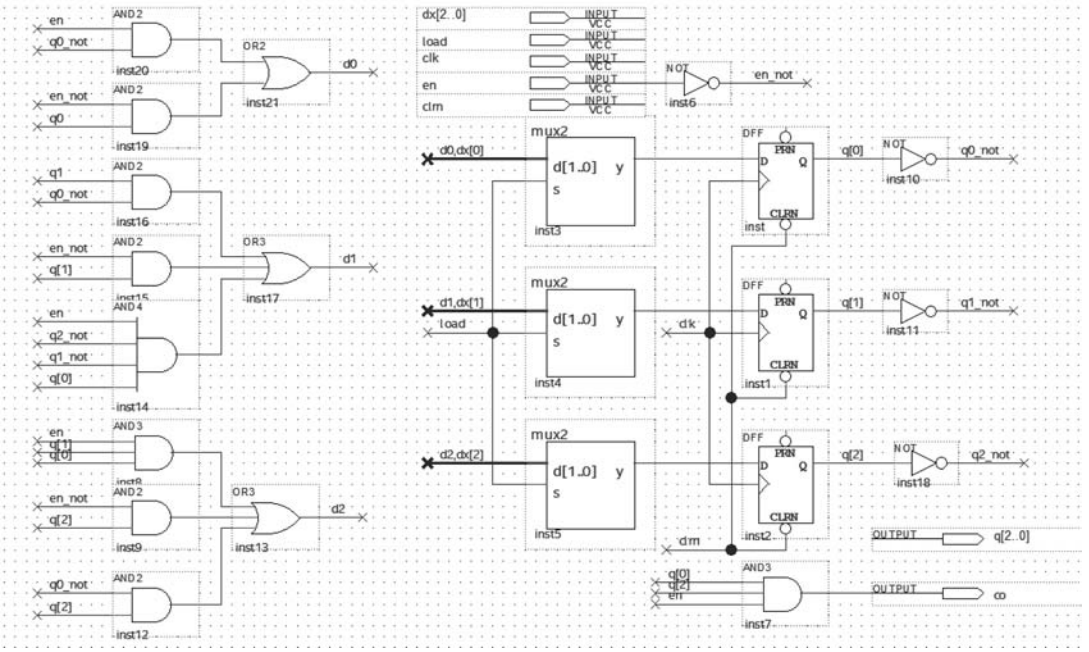


图 3-36　六进制计数器原理图 enldncout6_g.bdf

4）仿真六进制计数器

建立波形仿真文件，设置输入信号，得到如图 3-37 所示的仿真结果，可以看出，输出信号符合设计要求。

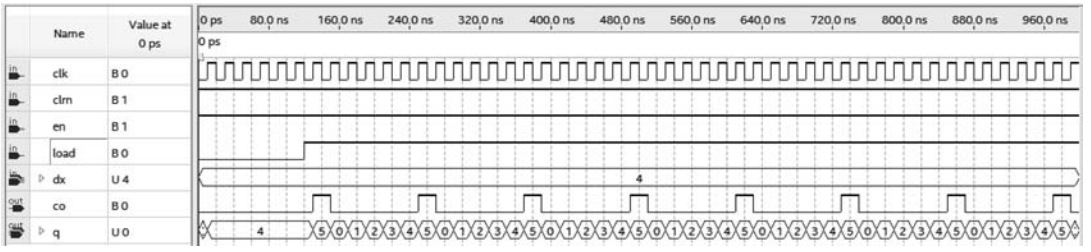


图 3-37　六进制计数器仿真结果

2. 六十进制计数器设计

1）六十进制计数器的真值表

六十进制计数器的输入/输出引脚介绍如下。

计数时钟输入端：clk。清零端(低电平使能)：clrn。预置控制端(低电平使能)：ldn。使能端：en。数据预置端：da[3..0]、db[2..0]。输出端：qa[3..0]、qb[2..0]。进位输出端 rco。其真值表如表 3-3 所示。

表 3-3　六十进制计数器真值表

控制端				十位预置	个位预置	十位输出	个位输出
clk	clrn	ldn	en	db[2..0]	da[3..0]	qb[2..0]	qa[3..0]
×	0	×	×	×	×	0	0
↑	1	0	×	b	a	b	a
↑	1	1	0	×	×	q(不变)	
↑	1	1	1	×	×	q=q+1(最高数到 59)	

2）六十进制计数器设计

利用十进制计数器 74160 组件与前面完成的六进制计数器 enldncout6_g 完成六十进制计数器电路图编辑结果如图 3-38 所示。

3）仿真六十进制计数器

建立波形仿真文件，设置输入信号，得到仿真结果如图 3-39 所示，可以看出，输出信号符合设计要求。

3.3.2　十二进制计数器设计

1. 十二进制计数器真值表

十二进制计数器的输入/输出引脚介绍如下。

计数时钟输入端：clk。清零端(低电平使能)：clrn。预置控制端(低电平使能)：ldn。使能端：en。数据预置端：da[3..0]、db。输出端：qa[3..0]、qb。其真值表如表 3-4 所示。

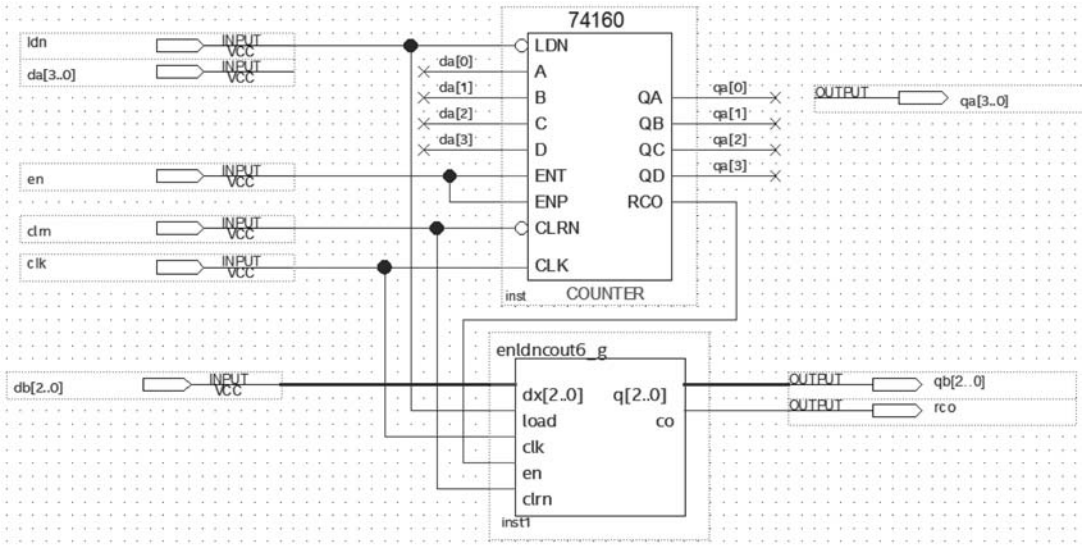


图 3-38　六十进制计数器原理图 enldncout60_g. bdf

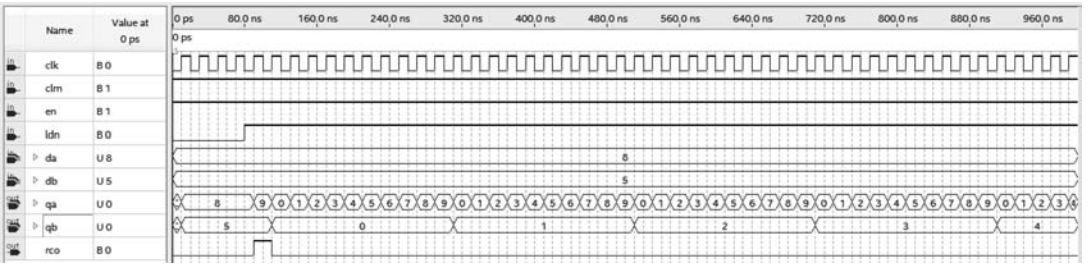


图 3-39　六十进制计数器仿真结果

表 3-4　十二进制计数器真值表

控制端				十位预置	个位预置	十位输出	个位输出
clk	clrn	ldn	en	db	da[3..0]	qb	qa[3..0]
×	0	×	×	×	×	0	0
↑	1	0	×	b	a	b	a
↑	1	1	0	×	×	q(不变)	
↑	1	1	1	×	×	q(不变)	
↑	1	1	1	×	×	q=q+1	

2. 十二进制计数器设计

1）二进制计数器的设计

十二进制计数器的十位需要二进制计数器，为此首先设计二进制计数器，如图 3-40 所示。

2）十二进制计数器的设计

运用十进制计数器 74160 器件与二进制计数器 enldncout2_g 可以完成十二进制计数

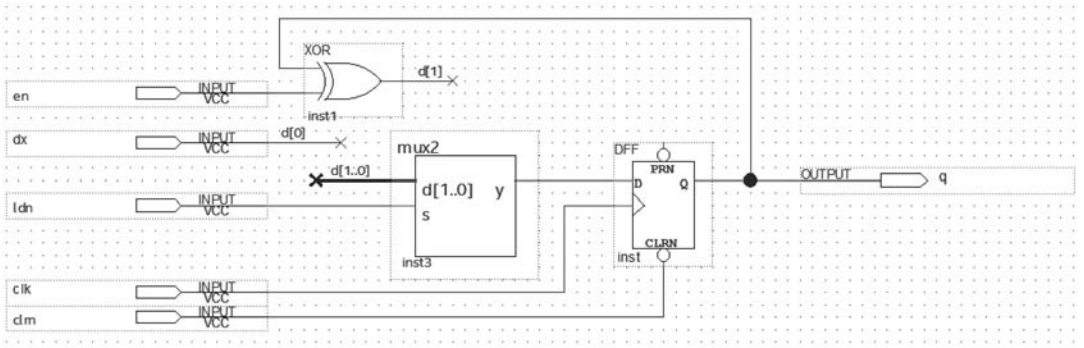


图 3-40　二进制计数器原理图 enldncout2_g. bdf

器的设计，电路图编辑如图 3-41 所示。

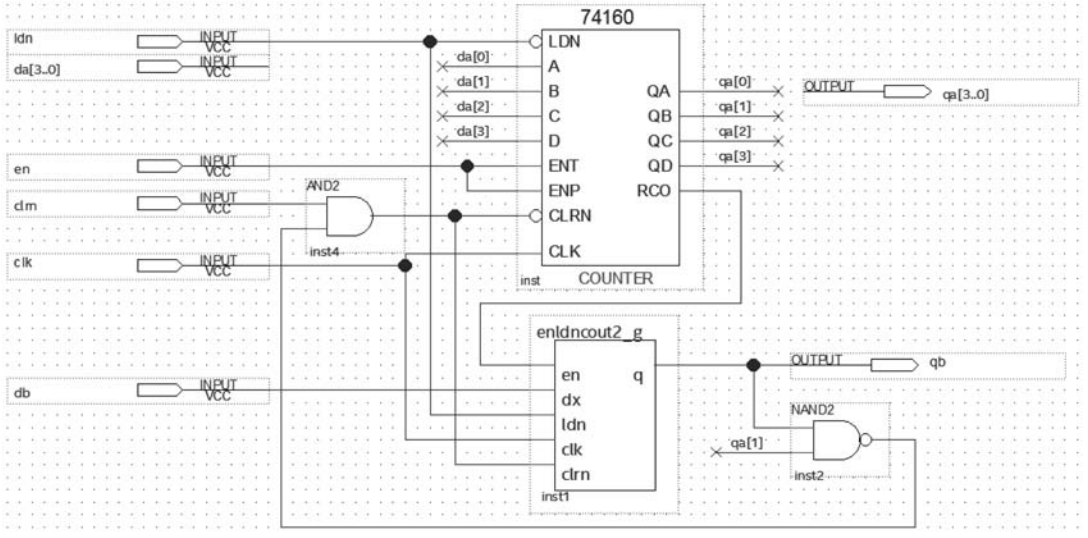


图 3-41　十二进制计数器原理图 enldncout12_g. bdf

3. 仿真十二进制计数器

建立波形仿真文件，设置输入信号，得到仿真结果如图 3-42 所示，可以看出，输出信号符合设计要求。

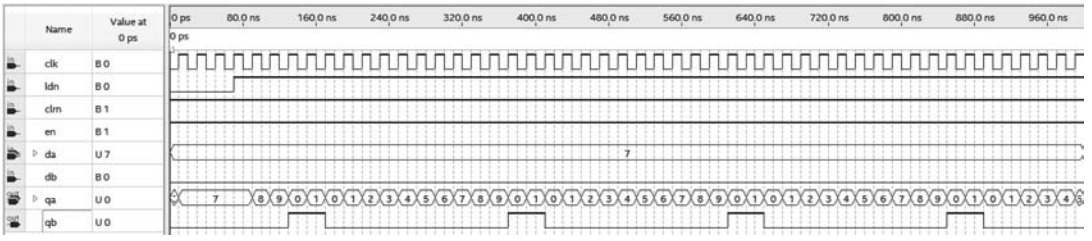


图 3-42　十二进制计数器仿真结果

3.3.3 数字电子钟顶层电路设计

1. 数字电子钟顶层电路设计

为简单起见，在此设计一个从0点0分0秒数到11点59分59秒的数字电子钟电路。其输入/输出引脚为：计数时钟输入端：clk。预置控制端：ldn。清零端：clrn。使能端：en。数据预置端：sa[3..0]、sb[2..0]、ma[3..0]、mb[2..0]、ha[3..0]、hb。输出端：qsa[3..0]、qsb[2..0]、qma[3..0]、qmb[2..0]、qha[3..0]、qhb。各引脚作用介绍如表3-5所示。

表3-5 数字电子钟数据脚位

	时针十位	时针个位	分针十位	分针个位	秒针十位	秒针个位
数据预置端	hb	ha[3..0]	mb[2..0]	ma[3..0]	sb[2..0]	sa[3..0]
时钟输出端	qhb	qha[3..0]	qmb[2..0]	qma[3..0]	qsb[2..0]	qsa[3..0]
计数器进制	十二进制计数器		六十进制计数器		六十进制计数器	
显示数字	00～11		00～59		00～59	

制作数字电子钟时、分、秒电路图如图3-43所示。

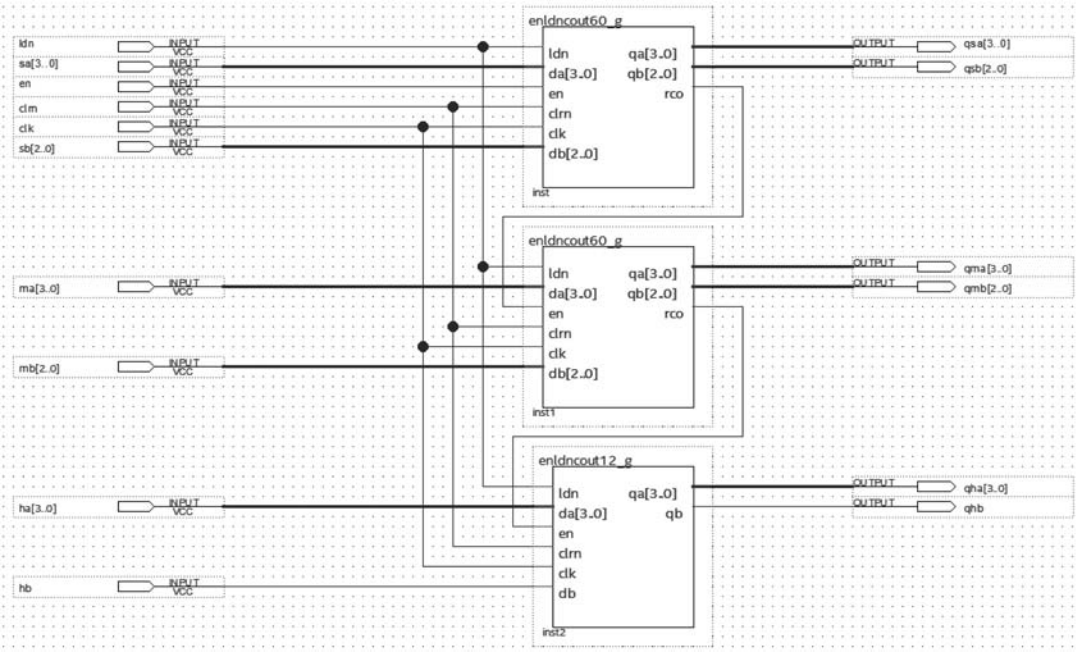


图3-43 电子钟时分秒计数器原理图 watch.bsf

2. 仿真数字钟

建立波形仿真文件，设置输入信号，得到仿真结果如图3-44所示，可以看出，输出信号符合设计要求。

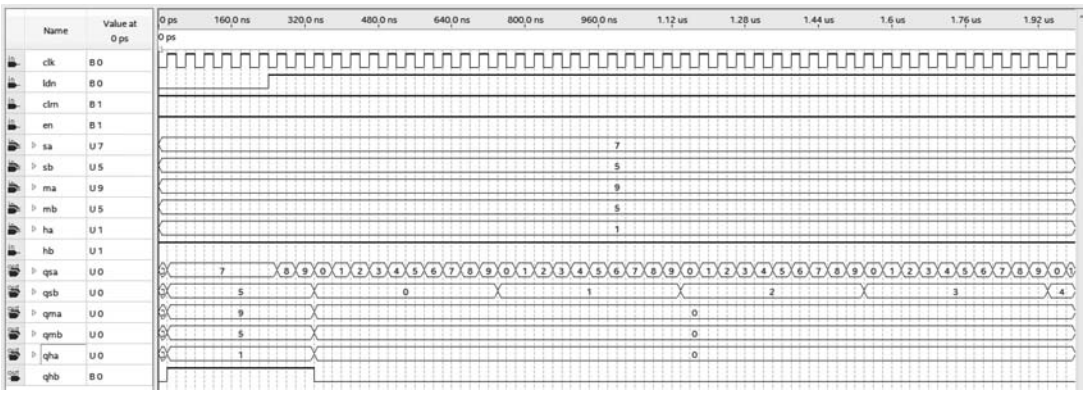


图 3-44　数字钟仿真结果

3.4　利用 LPM 兆功能块的电路设计

LPM(Library of Parameterized Modules,参数可设置模块库)是优秀的原理图设计人员智慧的结晶。具体地讲,一些模块的各种参数是由电路设计者为了适应设计电路的要求而定制的,通过修改 LPM 器件的某些参数,从而达到设计要求,使得基于 EDA 技术的电子设计的效率和可靠性有了很大的提高。

3.4.1　常用 LPM 兆功能块

作为 EDIF(电子设计交换格式)标准的一部分,LPM 形式得到了 EDA 工具的良好支持,LPM 中功能模块的内容丰富。Quartus Prime 对老版本的开发软件 Max+Plus II 和 Quartus II 提供的 LPM 中多种实用的 LPM 兆功能块进行了重新分类与整理。表 3-6 列出了 Quartus Prime 软件提供的主要的 LPM 兆功能块,功能比较复杂的兆功能块则划入了 IP 核中。常用的兆功能模块都可以在 mega-lpm 库中看到,每一模块的功能、参数含义、使用方法、硬件描述语言模块参数设置及调用方法都可以在 Quartus Prime 中的 Help 中查阅到,方法是在浏览器地址栏中输入 file:///C:/intelfpga/18.1/quartus/common/help/webhelp/index.htm#hdl/mega/mega_list_mega_lpm.htm,或者直接从文件系统中找到 Quartus 安装路径下的 htm 文件。以下将以基于 LPM_COUNTER 的数控分频器的设计为例说明 LPM 模块的原理图使用方法。

表 3-6　常用兆功能块

分　类	子　类	宏 单 元	注　释
兆功能函数(megafunctions)	IO	alt4gxb	光纤接口
		altlvds_rx	LVDS 输入接口
		altlvds_tx	LVDS 输出接口
		sld_virtual_jtag	虚拟 JTAG 接口
	算术运算(arithmetic)	altera_mult_add	乘加器
		altfp_abs	浮点求绝对值
		altmult_complex	复数乘法器

续表

分　类	子　类	宏 单 元	注　释
兆功能函数（megafunctions）	算术运算（arithmetic）	lpm_counter	计数器
		lpm_divide	除法器
	门电路（gate）	busmux	总线选择器
		lpm_bustri	总线三态门
		lpm_or	按位或
		mux	数据选择器
其他（others）	maxplus2	161mux	16 选 1 数据选择器
		4count	4 位二进制计数器
		7400	2 输入端与非门
		74160	十进制计数器
		7474	双路 D 触发器
	Opencore_plus	ocp_timeout_indicator	ocp 超时指示器
原语（primitives）	缓冲器（buffer）	alt_inbuf	输入缓冲器
		alt_outbuf	输出缓冲器
		alt_iobuf	双向缓冲器
		tri	三态门
	逻辑门（gate）	and12	12 输入端与门
		nand4	4 输入端与非门
		not	非门
		xor	异或门
	其他（other）	constant	常量
		vcc	高电平
		gnd	低电平
	引脚（pin）	bidir	双向端
		input	输入端
		output	输出端
	存储（storage）	dff	D 触发器
		dffea	带使能端和置数端的 D 触发器
		jkff	JK 触发器
		tff	T 触发器

3.4.2　基于 lpm_counter 的数据分频器设计

数控分频器的功能要求当在其输入端给定不同的数据时，其输出脉冲具有相应的对输入时钟的分频比。设计流程是首先按照 3.1.4 节的设计步骤，通过在原理图编辑窗口中调入兆功能元件，并按照图 3-45 的方式连接起来，其中，计数器 lpm_counter 元件的参数设置可按照以下介绍的方法进行。

用鼠标双击如图 3-45 所示的 LPM_COUNTER 右上角的参数显示文字，然后在弹出参数设置对话框中选择合适的参数，在窗口的 Ports 和 Parameters 栏中计数器各端口/参数的含义如下。

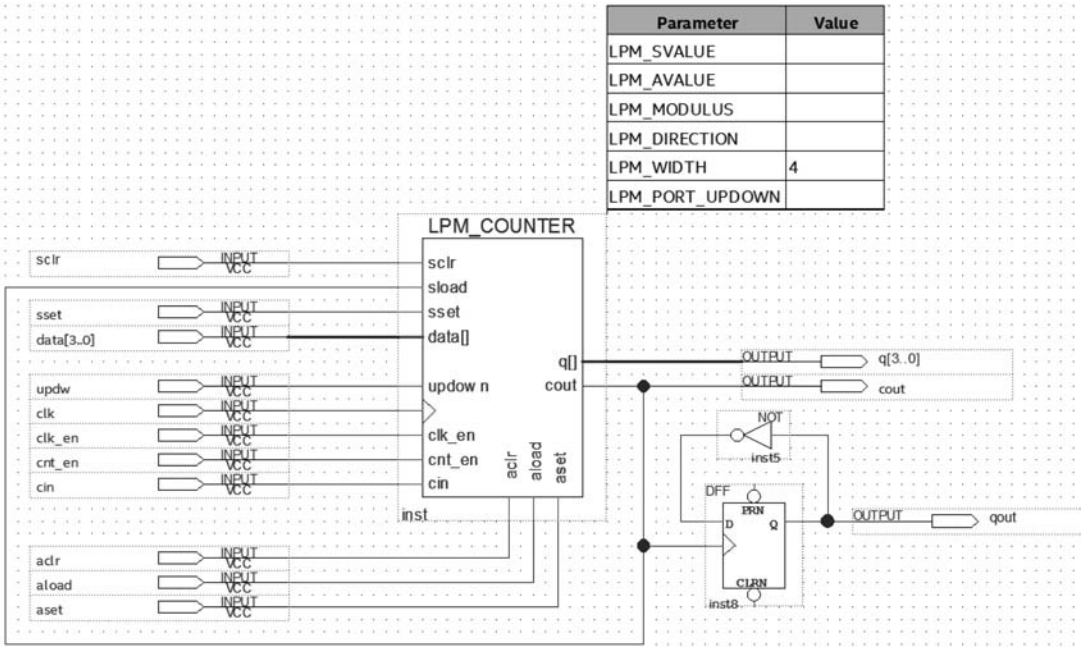


图 3-45　数控分频器电路原理图

1. Ports

sclr：同步清零。

sload：同步置数(置数值为 data[])。

sset：同步置位(计数器所有位全 1)。

data[]：置数的并行数据输入。

updown：计数器加减控制输入。

clock：上升沿触发计数时钟输入。

clk_en：高电平使能所有同步操作输入信号。

cnt_en：计数使能控制。

cin：最低进位输入，要使计数器正常计数，cin 必须为 1。

aclr：异步清零。

aload：异步置数(置数值为 data[])。

aset：异步置位(计数器所有位全 1)。

q[]：计数输出。

cout：计数进位或借位输出。

2. Parameters

LPM_SVALUE：sset 输入端值。

LPM_AVALUE：aset 输入端值。

LPM_MODULUS：计数器模值。

LPM_DIRECTION：计数器默认加计数/减计数。

LPM_WIDTH：计数器位宽。

LPM_PORT_UPDOWN：是否使能 updown 输入端。

设置情况如图 3-45 所示，计数器宽为 4，即 4 位计数器。工作原理如下。

当计数器计满“1111”时，由 cout 发出进位信号给并行加载控制信号 sload，使得 4 位并行数据 d[3..0]数据被加载进计数器中，此后计数器将在 d[3..0]数据的基础上进行加/减计数。如果是加法计数，则分频比为 R＝“1111”－d[3..0]＋1，即如果 d[3..0]＝12，则R＝4，即 clk 每进入 4 个脉冲，cout 输出一个脉冲；而如果作减法计数时，分频比为 R＝d[3..0]＋1，即如果 d[3..0]＝12，则 R＝13。图 3-46 是当 d[3..0]＝12 时的工作波形。

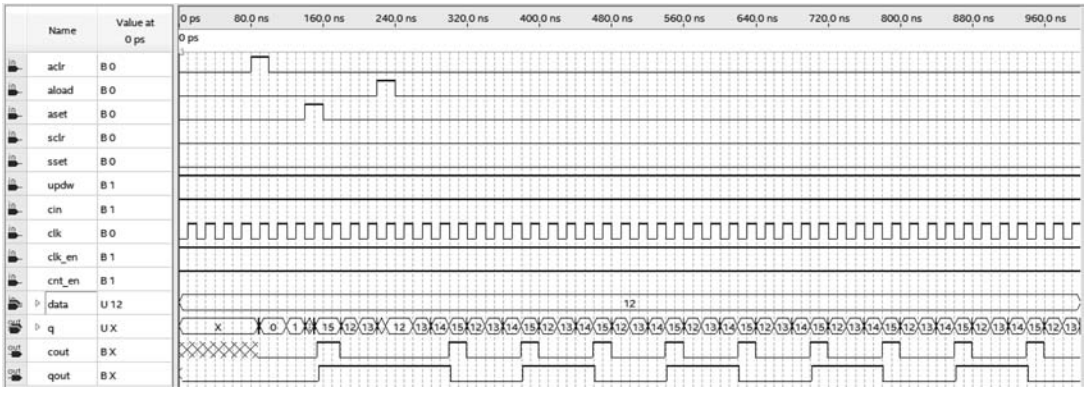


图 3-46　数控分频器工作波形

3.4.3　制作一个兆功能模块

Quartus Prime 把过去的 Max＋Plus II 和 Quartus II 软件版本中的兆功能库重新进行了整理，部分兆功能元件划入了 IP 核类中，IP 核使用时必须例化，也即根据 IP 核的模板，设定必要的参数，制作一个兆功能模块。下面以 LPM_COUNTER 为例，介绍该 IP 核例化的具体步骤。

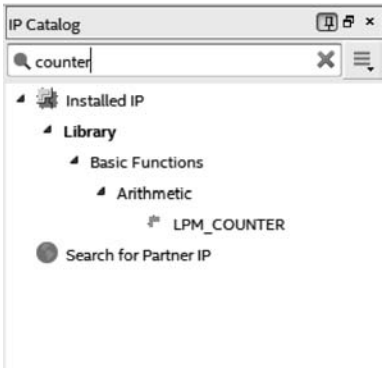


图 3-47　创建一个新的兆功能块

（1）在 Quartus Prime 18 主界面右侧的 IP catalog 子窗口上，输入“counter”，查找能匹配到的 IP 核，如图 3-47 所示，匹配到 LPM_COUNTER。如果主界面右侧没有 IP catalog 子窗口，选择主界面上 View→Utility→IP catalog 子菜单，调出 IP Catalog 子窗口。

（2）双击 LPM_COUNTER，弹出如图 3-48 所示的 IP 实例命名对话框，输入实例名称后，单击 OK 按钮进入参数设置界面。

（3）LPM_COUNTER 的参数设置界面是流水线式设置向导，图 3-49(a)为计数器位宽和加/减控制设置，图 3-49(b)为计数器模值与计数使能、低位进位输入等设置，图 3-49(c)为清零、置位、置数端的设置，如图 3-49 所示。

（4）所有参数设置完成后，单击 Next 按钮继续，弹出如图 3-50 所示的生成相关文件的对话框。“.inc”文件用于在 AHDL 程序中调用该计数器所需文件，如果不使用 AHDL 编

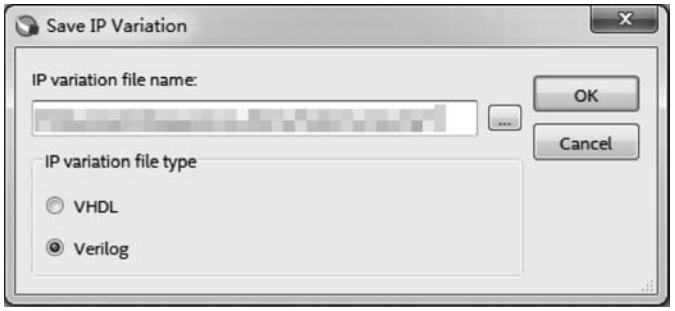


图 3-48　选择兆功能模块的类型并定义名称

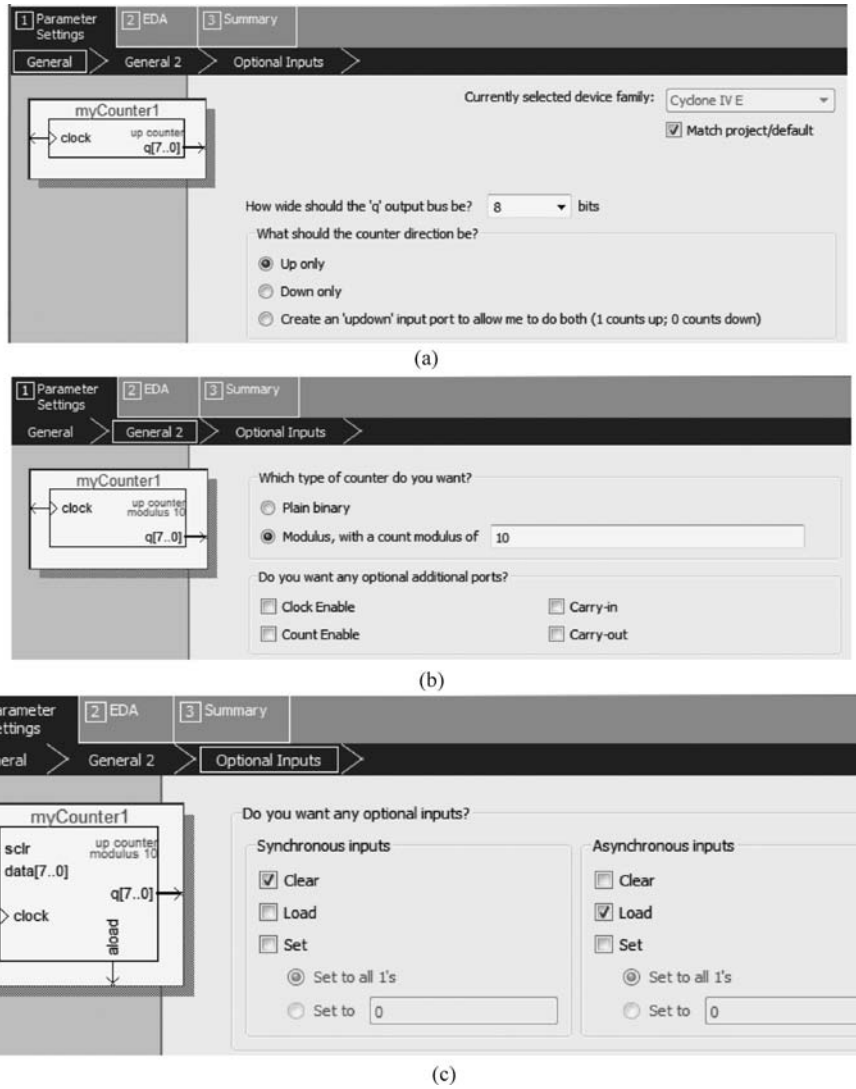


图 3-49　LPM_COUNTER 的参数设置界面

程，不用生成该文件；“.cmp”文件是 VHDL 程序的元件宣言文件，是 VHDL 程序中调用该计数器所需文件；“.bsf”文件为图标文件，原理图设计文件中调用该计数器时需要该文件；“*_inst.v”和“*_bb.v”这两个 verilog 程序文件分别是生成的实例文件和黑匣子文件。建议初学者全部勾选这些文件。

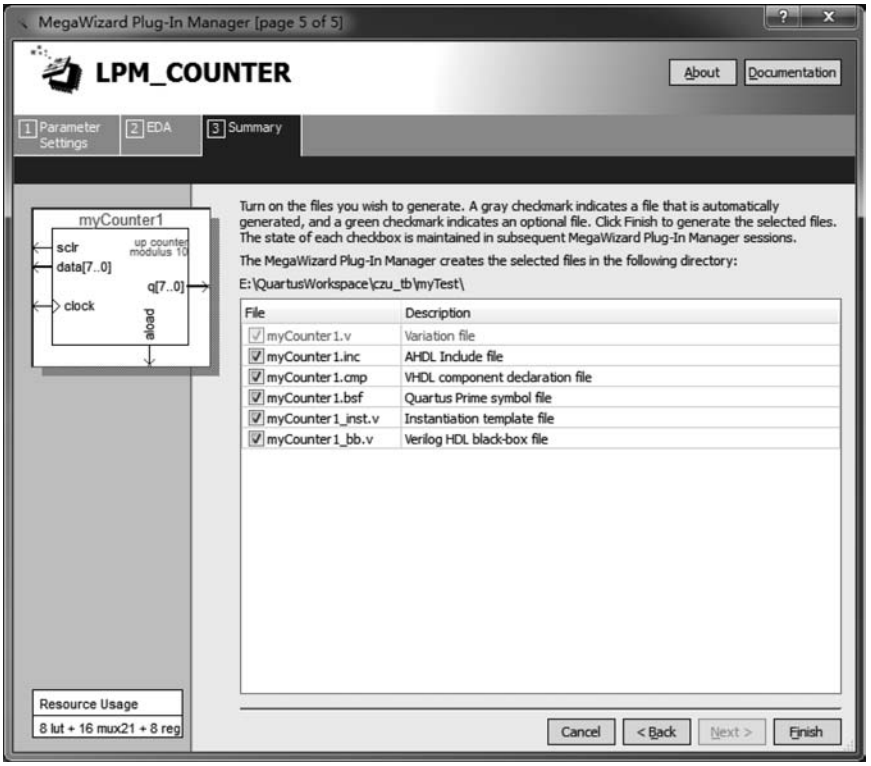


图 3-50　兆功能模块的汇总信息

（5）单击 Finish 按钮，即完成了计数器 IP 实例或称兆功能模块的制作。以后原理图设计和 HDL 代码编辑时就可以调用这个名为“myCounter1”的兆功能模块了。

如果后期要修改 myCounter1 的参数，若在某原理图中已添加 myCounter1 元件，双击 myCounter1 即可打开参数设置界面，或者从主界面左上角的工程导航区的下拉菜单中选择 IP 元件找到 myCounter1，双击即可打开其参数设置界面。

3.5　编译报告

当某个工程成功编译完成后，会得到 Quartus Prime 给出编译报告。在主界面，可以得到一个报告的梗概（Flow Summary），如图 3-51 所示。按 Ctrl＋R 快捷键或者选择 Processing→Compilation Report 也可以调出编译报告。

报告梗概（Flow Summary）从上到下依次给出了编译时间、Quartus 版本、工程名、顶层实体名、器件家族（Family）、器件具体型号（Device）、时序模型、总逻辑单元（LEs）使用量、总寄存器数（registers）、总管脚（pins）使用量、总虚拟管脚、总内存位使用量、嵌入式乘法器

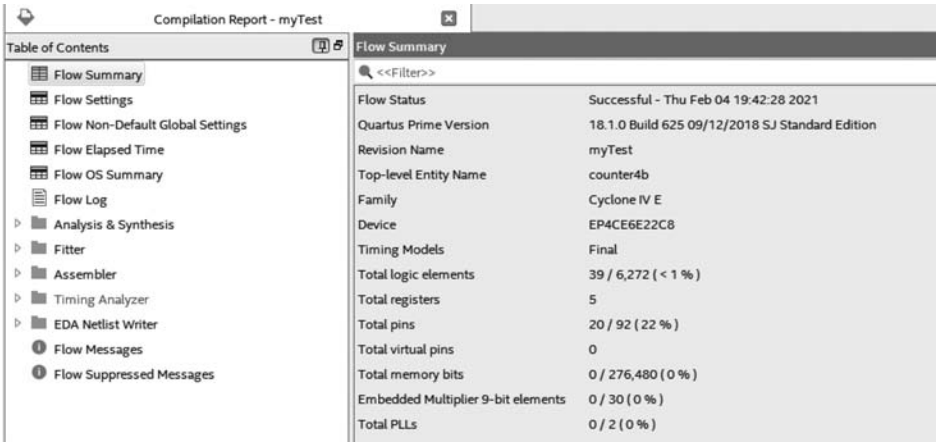


图 3-51　编译报告梗概

(9b)使用量、总锁相环(PLL)使用量等信息，需要特别关注片上逻辑资源的使用量，若某项资源不足时，则要对设计进行优化，以尽量适应所选择的 FPGA 器件。

Quartus 还给出了分类的详细报告(图 3-51 左侧 Table of Contents)。Flow Settings 给出了编译开始时间、任务、工程名等信息。Flow Non-Default Global Settings 列出了工程中一些全局设置值，如 testbench 文件名、testbench 的模块名等。Flow Elapsed Time 给出了大编译过程中各个环节所用的时间，如分析与综合、适配、组合、时序分析等。Flow OS Summary 给了计算机操作系统的版本以及处理器类型。我们重点应该关注仿真与综合项(Analysis & Synthesis)下的详细信息，仿真与综合项下包含各个实体的资源使用情况、片上 RAM 使用情况、IP 核使用情况等重要信息。根据顶层设计文件上各个实体的资源使用情况，可以有的放矢地进行裁剪，以达到减少片上资源的目标。图 3-52 为 3.4.2 节中的设计实例经过编译后查找表(LUT)、逻辑寄存器(LE)、片上 RAM(Memory bits)等各类资源使用量。

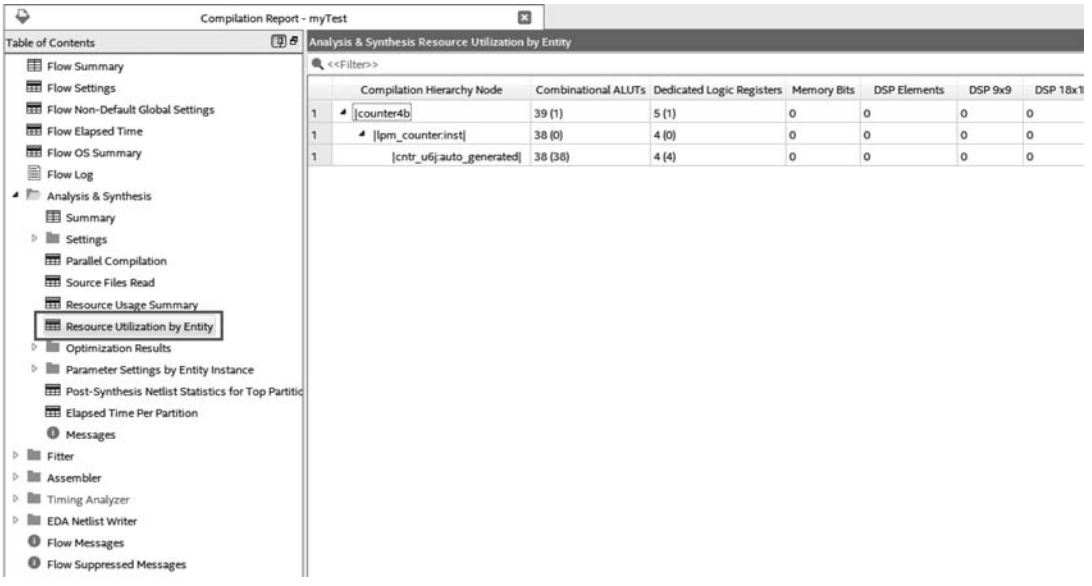
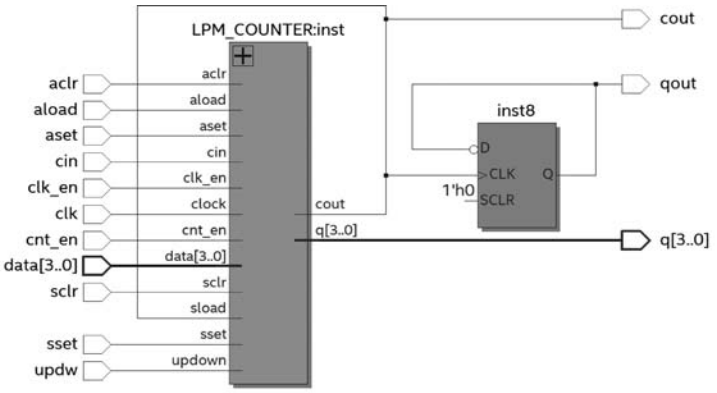


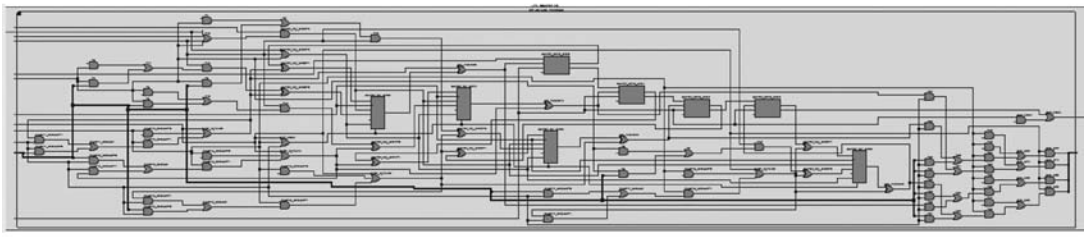
图 3-52　编译报告

编译报告中的 Timing Analyzer 分项通常以红色显示，表明时序不满足通常的约束条件，其原因已经在 3.1.6 节中说明，这时需要利用 Tools 下的 Timing Analyzer 工具进行必要的时序约束，这里不再赘述。

另外，设计者还可以通过 RTL Viewer 工具观察经过编译以后得到的寄存器传输级的实现电路图，选择 Tools→Netlist Viewer→RTL Viewer 子菜单，可以得到 3.4.3 节中实例的顶层设计的框图，如图 3-53(a)所示。双击 LPM_COUNTER：inst 还可以观察下一层的 RTL 图，如图 3-53(b)所示，在该层中便能够看到复杂的门级实现电路。



(a)



(b)

图 3-53　RTL 原理图

思考题与习题

1. 简述用原理图输入方式设计电路的详细流程。

2. 功能仿真和时序仿真有何区别？如何利用 Quartus Prime 18 进行这些仿真？

3. 如何设置仿真栅格时间及仿真终止时间？

4. 如何进行多层次的电路系统设计？

5. 设计一个 4 选 1 多路选择器，当选择输入端信号分别取“00”“01”“10”和“11”时，输出信号分别与一路输入信号相连。

6. 设计一个 7 人表决电路，参加表决者 7 人，同意为 1，不同意为 0，同意者过半则表决通过，绿指示灯亮；表决不通过则红指示灯亮。

7. 设计一个 8 位加法器电路。

8. 设计一个 4 位寄存器电路。

9. 设计一个异步清除 4 位同步加计数器电路。

10. 设计一个具有预置功能的三位数的十进制计数器电路。

11. 设计一个由两级 D 触发器组成的四分频电路。

12. 用 74194、74273、D 触发器等器件组成 8 位串入并出的转换电路，要求在转换过程中数据不变，只有当 8 位一组数据全部转换结束后，输出才变化一次。

13. 设计两位十进制频率计，F_IN 是待测频率信号(设其频率周期为 410ns)；CNT_EN 是对待测频率脉冲计数允许信号(设其频率周期为 32μs)，CNT_EN 高电平时允许计数，低电平时禁止计数。

14. 用两片 74160 设计计数长度为 60 的计数器 cnt60.bdf，并进行功能仿真。

15. 利用 LPM 模块，即 lpm_add_sub、busmux、lpm_latch 及其他模块构成一个可预置初值的减法计数器。