第3章 可编程技术实验设计

数字电路的 VHDL 程序设计完成后，必须利用 EDA 软件中的综合器、适配器、时序仿真器和编程器等工具进行相应的处理和下载，才能使此项设计在 FPGA 上完成硬件实现并能进行硬件测试。在 EDA 工具的设计环境中，有多种途径来完成目标电路系统的表达和输入方式，如 HDL 的文本输入方式、原理图输入方式、状态图输入方式、波形输入方式、Matlab 的模型输入方式，以及混合输入方式。本章将详细介绍利用 Quartus Ⅱ 和 FPGA 实验台进行逻辑电路设计的实验方法和步骤。本章设计了一些可编程技术基础实验，通过实验可以更好地掌握 FPGA 开发流程，掌握 Quartus Ⅱ 软件和 GW48-CP++实验台的使用方法，学会用 VHDL 和原理图方法设计逻辑电路，掌握用层次化设计方法设计复杂电路的能力。本章的实验侧重点是软硬件开发环境的熟悉，每个实验都提供了详细的实验步骤和操作说明，是后续实验的基础。

3.1 可编程技术基础实验

3.1.1 熟悉 FPGA 软硬件开发环境

1. 实验目的

(1) 熟悉 Quartus Ⅱ 开发环境及开发流程。

(2) 掌握 Quartus Ⅱ 中 VHDL 文本输入设计方法。

(3) 熟悉 FPGA 实验台的功能和使用方法。

2. 实验内容

利用 Quartus Ⅱ 完成 2 选 1 多路选择器的文本编辑输入(mux21a.vhd)和仿真测试等步骤，给出 2 选 1 多路选择器仿真波形。最后在实验系统上进行硬件测试，验证 2 选 1 多路选择器的功能。

3. 实验环境

PC	1台
Quartus Ⅱ软件环境	1套
GW48-CP++实验台	1台

4. 实验原理与电路

本实验利用VHDL设计2选1多路选择器，其电路示意图和原理图分别见2.1节的图2-1和图2-2，真值表见2.1节的表2-8。利用Quartus Ⅱ设计2选1多路选择器主要设计步骤如图3-1所示。

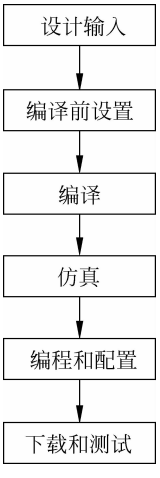


图3-1 2选1多路选择器主要设计步骤

(1) 设计输入。主要完成2选1多路选择器的逻辑电路描述，2选1多路选择器的真值表如表2-8所示，采用VHDL语言设计2选1多路选择器，VHDL代码参见2.1节中的程序P2-1。

(2) 编译前设置。主要完成目标芯片的选择，即制定实验台上的FPGA芯片型号。GW48-CP++实验台上的芯片型号为EP3C40Q240，在建立工程时可以指定芯片型号，也可以在建立工程后通过主菜单Assignment→Device设置。

(3) 编译。主要实现逻辑电路的排错，完成电路的综合、布局、布线等工作。在编译过程中可以输出电路中存在的错误，需要改正后，才能编译成功。

(4) 仿真。完成对电路的功能测试，通过建立波形文件设置电路的输入信号，运行仿真观察波形的输出结果，检查电路的逻辑器功能是否正确。仿真的关键是在波形图上合理的设置输入信号的值，可以按照真值表设置输入信号的值，仿真后，按照输入信号和电路的逻辑自动在波形图上产生相应的输出信号，然后按照真值表的理论值验证仿真波形图上输入/输出信号的值是否正确。仿真的时间可以根据实际需求以及电路功能的复杂程度设置，比较简单的电路，可以设置较短的仿真时间，即可完成仿真电路的功能。仿真时间的设置可通过主菜单Edit→End Time来设置。一般设置μs数量级就可以满足要求。

(5) 编程和配置。Quartus Ⅱ编译电路成功后，自动将布局布线的电路，逻辑单元和引脚分配转换为该电路的编程图像，即生成电路的Programmer对象文件(.pof)或SRAM对象文件(.sof)。在生成编程图像之前需要对电路的输入信号进行引脚分配，即使电路的每个输入/输出信号与FPGA上的引脚相连。以便后续利用实验台的按键和数码管等外设对电路进行测试。引脚锁定和实验台的使用方法见2.3.5节。

(6) 下载和测试。生成Sof文件后，启动编程器Programmer，选择ByteBlasterMV[Lpt1]编程方式(计算机与实验台采用LPT数据线连接)，然后可以将电路的Sof文件下载到实验台的FPGA芯片中。下载之前需要保证实验台与计算机已用数据线连接，并且实验台的电源已经打开。下载之后，可以利用实验台的按键和数码管等外设对电路进行测试。

5. 实验任务与步骤

1）建立工作库文件夹和编辑设计文件

首先建立工作库目录，以便存储工程项目设计文件。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的所有设计文件的文件夹。此文件夹将被 EDA 软件默认为工作库(Work Library)。一般来说，不同的设计项目最好放在不同的文件夹中，而同一工程的所有文件都必须放在同一文件夹中。注意不要将文件夹设在计算机已有的安装目录中，更不要将工程文件直接放在安装目录中。在建立了文件夹后就可以将设计文件通过 Quartus Ⅱ的文本编辑器编辑并存盘，详细步骤如下。

(1) 新建一个文件夹。利用资源管理器新建一个文件夹，如 D:\muxfile，注意文件夹名不能用中文。

(2) 输入 VHDL 源程序。打开 Quartus Ⅱ软件，选择主菜单 File→New，在 New 窗中的 Design Files 中选择编译文件的语言类型，这里选择 VHDL Files，如图 3-2 所示。然后在 VHDL 文本编译窗口中输入如图 3-3 所示的 VHDL 程序。

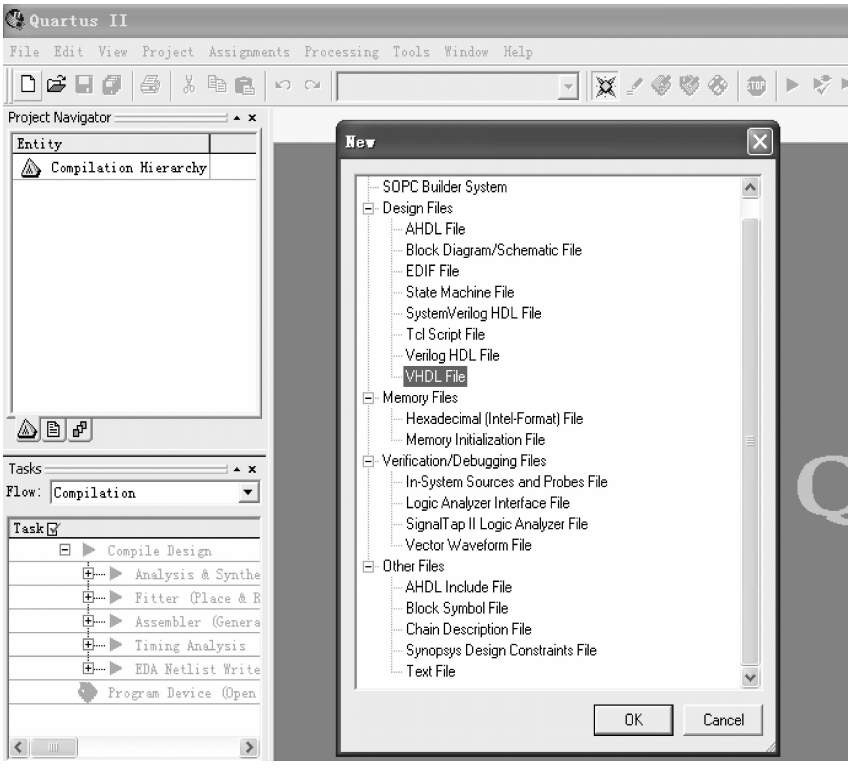


图 3-2 选择编辑文件的语言类型

(3) 文件存盘。选择主菜单 File→Save As，找到已设立的文件夹 D:\muxfile，存盘文件名应该与实体名一致，即 mux21a. vhd，实体名将在建立工程时指定。当出现 Do You Want To Create 提示信息时，若选择“否”，可按以下的方法进入创建工程流程；若选择“是”，则直接进入创建工程流程。

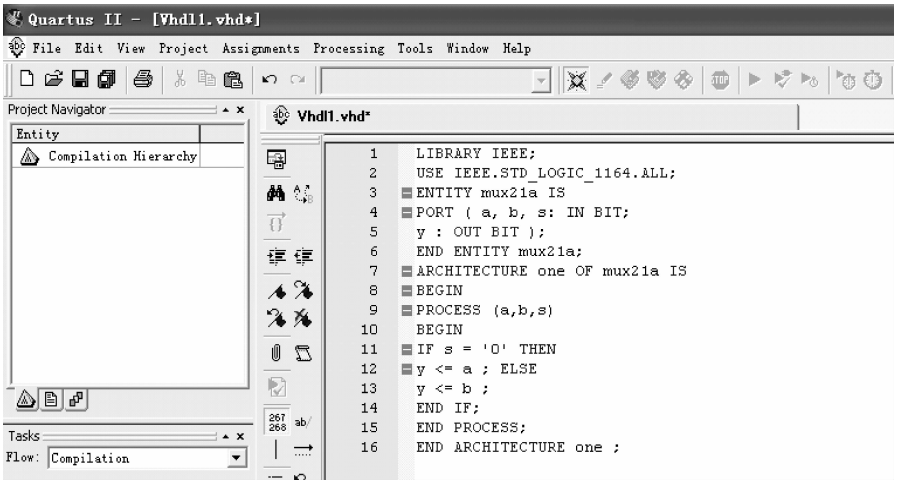


图 3-3 编辑输入设计文件

2）创建工程

使用 New Project Wizard 可以为工程指定工作目录、分配工程名称以及指定最高层设计实体的名称，还可以指定要在工程中使用的设计文件、其他源文件、用户库和 EDA 工具，以及目标器件系列和具体器件等。

在此要利用 New Project Wizard 创建此设计工程，即令 mux21a. vhd 为工程，并设定此工程一些相关的信息，如工程名、目标器件、综合器、仿真器等。步骤如下。

（1）建立新工程管理窗。选择主菜单 File→New Project Wizard，即弹出“工程设置”对话框，如图 3-4 所示。单击此对话框最上一栏右侧的按钮，找到文件夹 D:\muxfile，选中已存盘的文件 mux21a. vhd，再单击“打开”按钮，即出现如图 3-4 所示的设置情况。其中第一行表示工程所在的工作库文件夹；第二行表示此项工程的工程名，此工程名可以取任何其他的名，通常直接用实体名作为工程名；第三行是实体名。

（2）将设计文件加入工程中。单击“工程设置”对话框（图 3-4）下方的 Next 按钮，在弹出如图 3-5 所示的“添加文件”对话框中，将与此工程相关的所有 VHDL 文件加入此工程中。工程的文件加入的方法有两种：第一种是单击右边的 Add All 按钮，将设定的工程目录中的所有 VHDL 文件加入到工程文件栏中；第二种方法是单击“浏览”按钮，从工程目录中选出相关的 VHDL 文件，单击 Add 按钮将文件加入到工程。

（3）选择目标芯片。单击“添加文件”对话框（图 3-5）中的 Next 按钮，弹出如图 3-6 所示的“器件设置”对话框，首先在 Family 栏选择芯片系列，在此选择 Cyclone Ⅲ系列，在窗口右侧设置 Pin Count 为 240，这时列表中显示了 3 种具有 240 个引脚的芯片，这里选择 EP3C40Q240C8。此芯片即是 GW48-CP++实验台上的 FPGA 芯片。只有指定实验台上的目标芯片，后续才允许将电路下载到实验台的 FPGA 芯片中，不同的实验台或开发板上的 FPGA 芯片不同，需要根据实际情况选择目标芯片。

（4）选择仿真器和综合器类型。单击如图 3-6 所示的对话框中的 Next 按钮，这时弹出“EDA 工具设置”对话框，在此对话框可以选择仿真器和综合器类型，如果都是选择默认的 None 项，表示都选择 Quartus Ⅱ中自带的仿真器和综合器，因此，在此都选择默认项“None”。

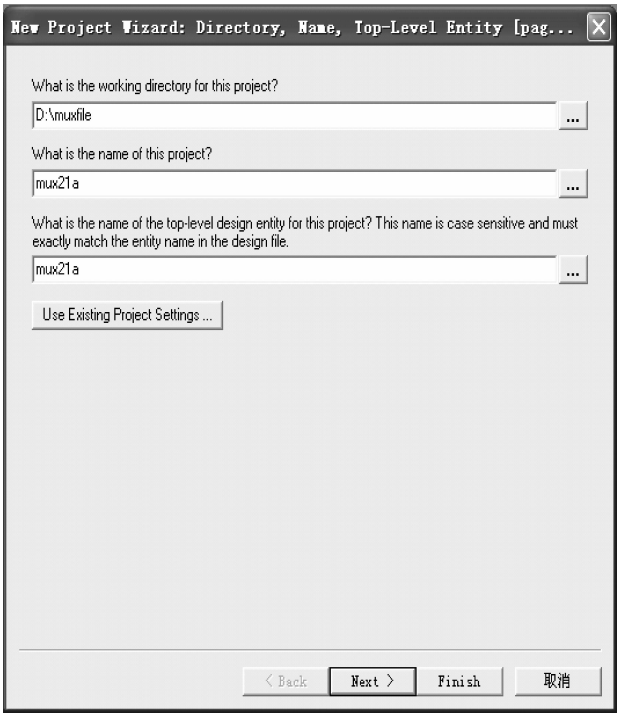


图 3-4　创建工程 mux21a

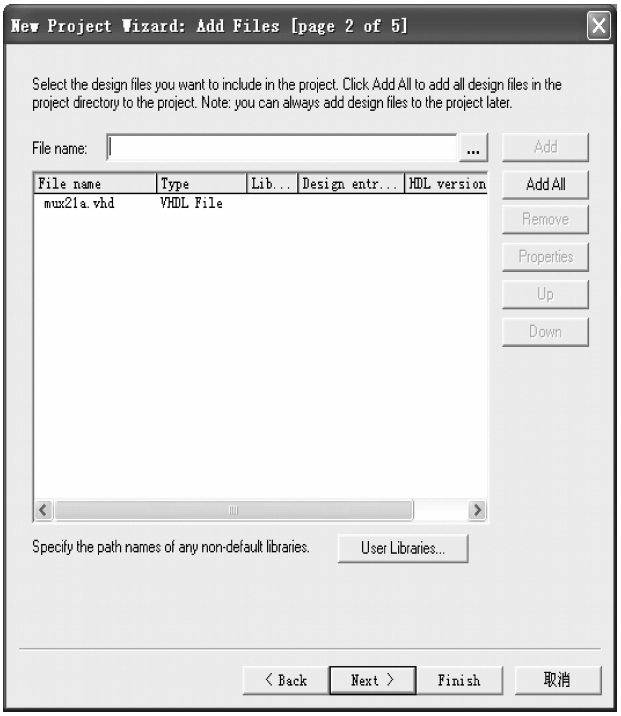


图 3-5　将相关文件加入工程

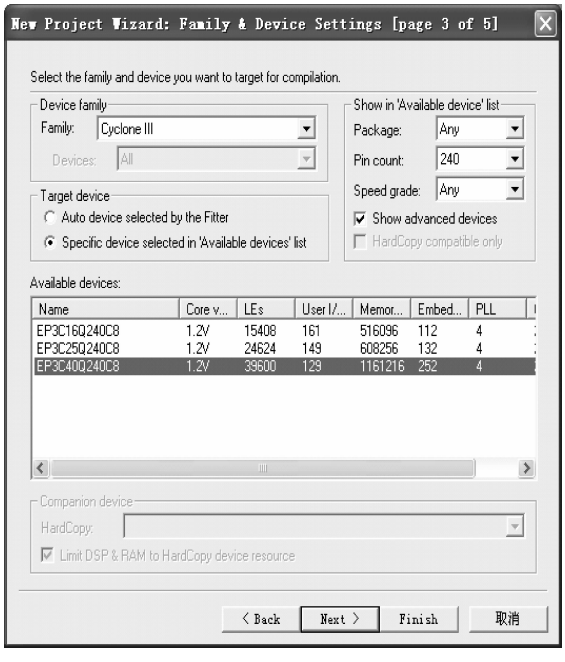


图 3-6 选择目标器件

(5) 结束设置。继续单击 Next 按钮，弹出如图 3-7 所示的“工程信息报告”对话框，显示了工程的实体名、目标芯片等信息。最后单击 Finish 按钮，即已设定好此工程，弹出如图 3-8 所示的窗口，此工程管理窗口(Project Navigator)主要显示工程项目的层次结构和各层次的实体名。

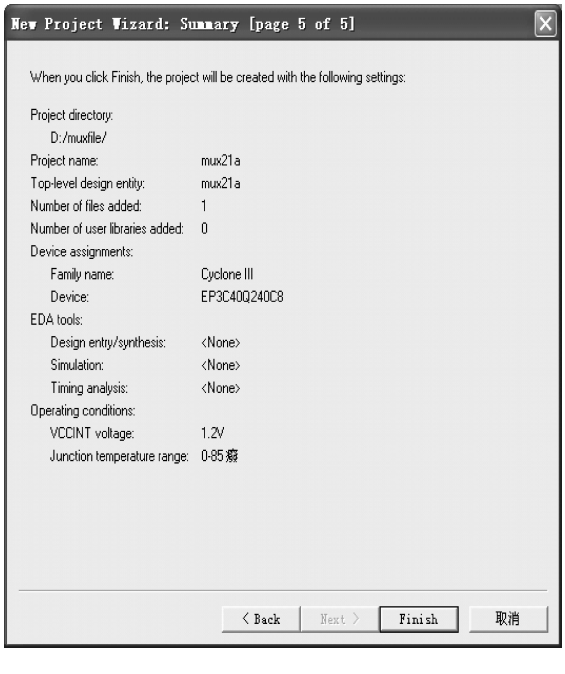


图 3-7 工程报告信息

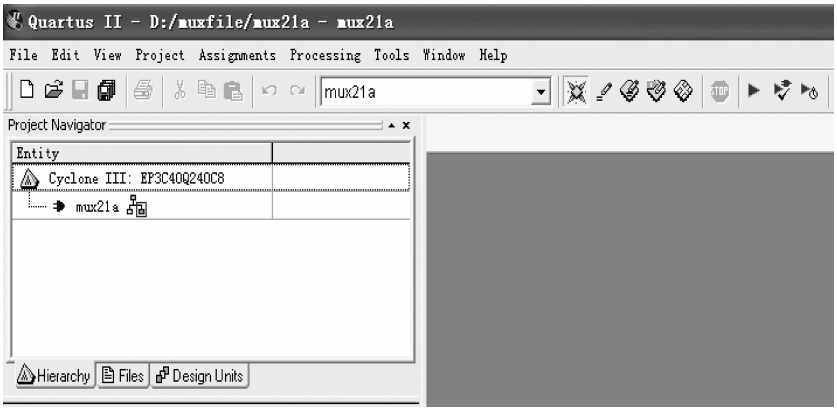


图 3-8 “工程管理”窗口

建立工程后，可以使用 Settings 对话框（Assignments 菜单）的 Add/Remove 页，在工程中添加和删除、设计其他文件。在执行 Quartus Ⅱ的 Analysis & Synthesis 期间，Quartus Ⅱ将按 Add/Remove 页中显示的顺序处理文件。

利用“工程管理”窗口（Project Navigator）可以方便地管理工程文件，双击 Project Navigator 窗口的实体名 mux21a，可以打开顶层实体的设计文件，即 mux21a. vhd。如图 3-9 所示。单击 Project Navigator 窗口的 Files 页面，如图 3-10 所示，可以看到工程中所有的文件，这里只有 mux21a. vhd，以后建立的其他文件也会出现在此页面。如果主窗口中没有显示 Project Navigator 窗口（可能不小心关闭了），可以选择主菜单 View→Utility Windows→Project Navigator，或者按 Alt＋0 组合键打开“工程管理”窗口。

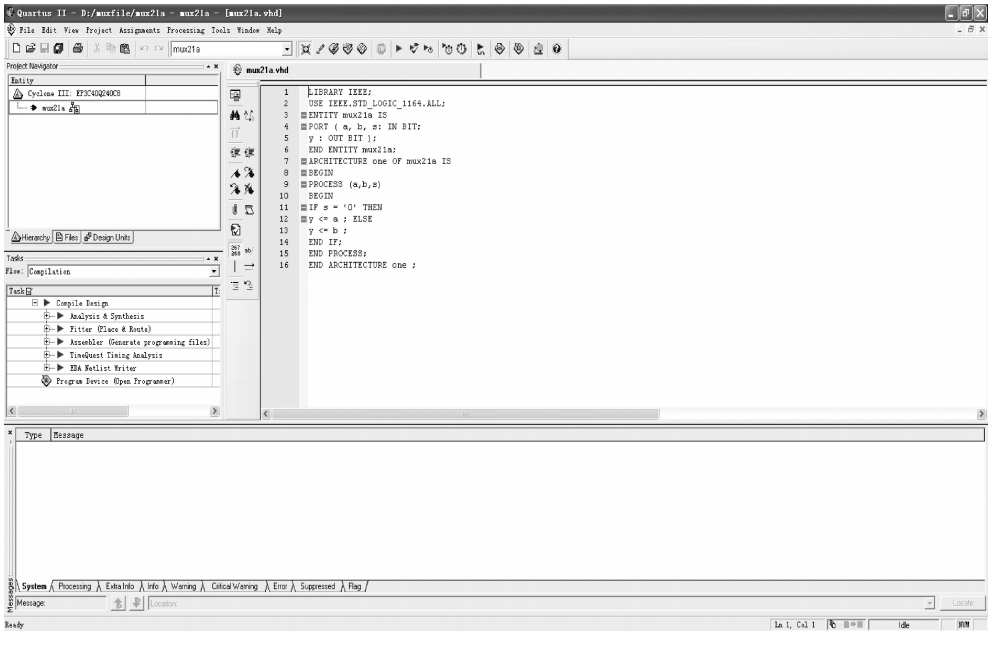


图 3-9 利用“工程管理”窗口打开顶层实体设计文件

3）编译前设置

在对工程进行编译处理前，必须做好必要的设置。

如果在建立工程时已经选择目标芯片可忽略此步骤。目标芯片的选择也可以这样来实现：选择主菜单 Assignments→Settings 项，在弹出的对话框中选择 Compiler Settings 项下的 Device，选择目标芯片 EP3C40Q240C8，如图 3-11 所示。首先在 Show in Available devices' list 栏中分别选择 Package 为“PQFP”、Pin count 为“240”、Speed grade 为“8”，然后在 Available devices 列表中选择目标芯片“EP3C40Q240C8”。

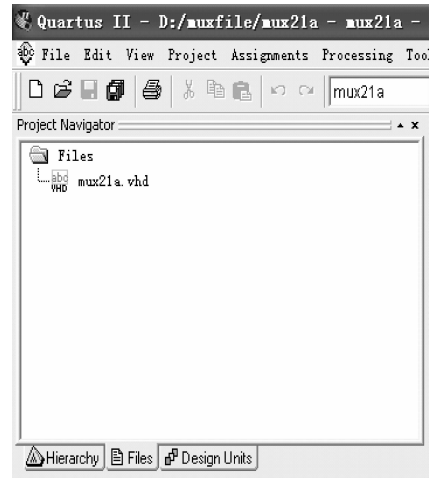


图 3-10 “工程管理”窗口的工程文件列表

4）编译

Quartus Ⅱ编译器是由一系列处理模块构成的，这些模块负责对设计项目的检错、逻辑综合和结构综合，即将设计项目适配进 FPGA/CPLD 目标器中，同时产生多种用途的输出文件，如功能和时序仿真文件、器件编程的目标

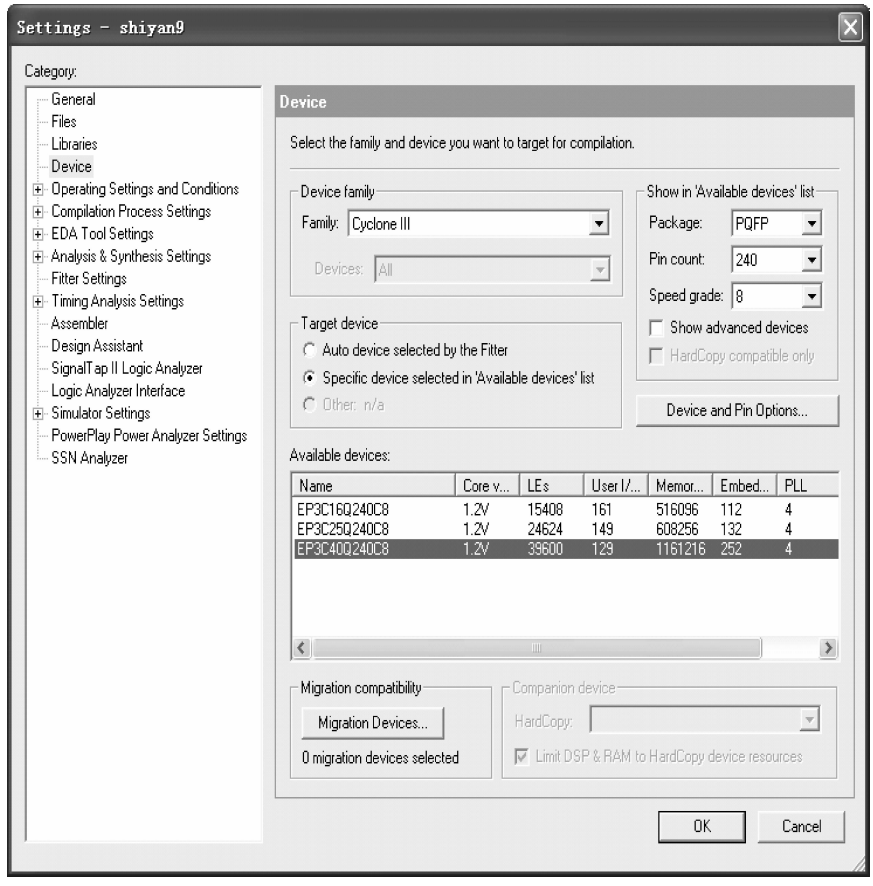


图 3-11 选定目标器件

文件等。编译器首先从工程设计文件间的层次结构描述中提取信息，包括每个低层次文件中的错误信息，供设计者排除，然后将这些层次构建产生一个结构化的以网表文件表达的电路原理图文件，并把各层次中所有的文件结合成一个数据包，以便更有效地处理。

下面首先选择主菜单 Processing→Start Compilation 项，启动全程编译。注意这里所谓的编译(Compilation)包括 Quartus Ⅱ对设计输入的多项处理操作，其中包括排错、数据网表文件提取、逻辑综合、适配、装配文件(仿真文件与编程配置文件)生成，以及基于目标器件的工程时序分析等。如果工程中的文件有错误，启动编译后在下方的 Processing 栏中会显示出来，如图 3-12 所示。对于 Processing 栏显示出的语句格式错误，可双击此条文，即弹出对应的 VHDL 文件，在深色标记条处(或附近)可发现文件中的错误。改正错误，再次进行编译直至排除所有错误。注意，如果发现报出多条错误信息，每次只要检查和纠正最上面报出的错误，因为许多情况下，都是由于某一种错误导致了多条错误信息报告。如果主窗口的 Processing 栏中没有显示 Messages 窗口(可能不小心关闭了)，可以选择主菜单 View→Utility Windows→Messages，或者按 Alt＋3 组合键打开 Messages 窗口。

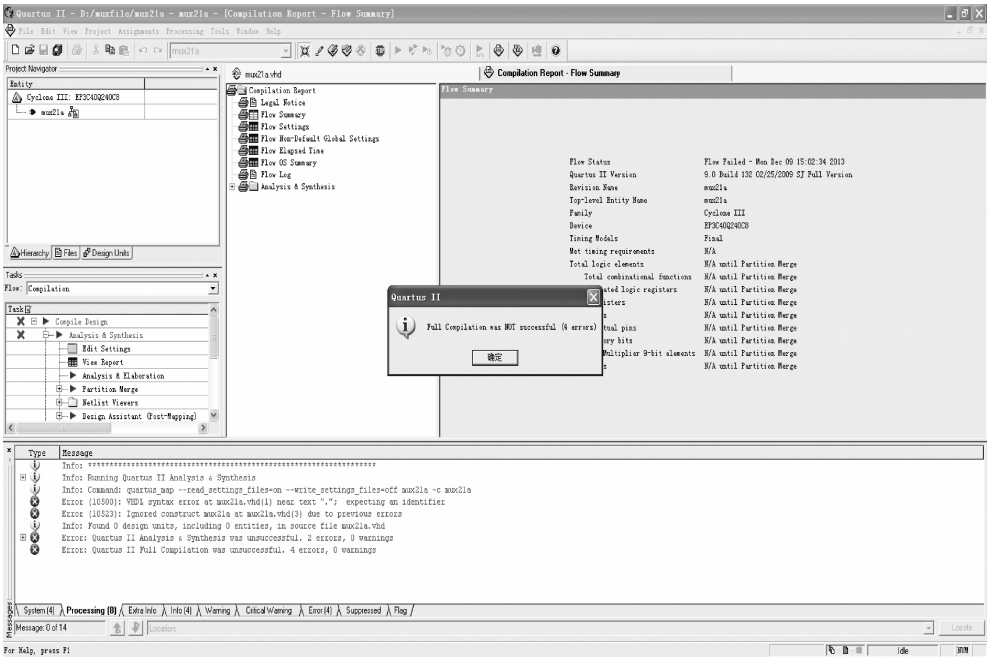


图 3-12　全程编译后出现报错信息

如果编译成功，可以见到如图 3-13 所示的信息。Compilation Report 栏是编译报告项目选择主菜单，单击其中各项可以详细了解编译与分析结果(默认显示 Flow Summary 项的信息)。Flow Summary 窗口显示硬件耗用统计报告，其中报告了当前工程耗用了一个逻辑宏单元、4 个引脚，0 个内部 RAM 位等。

如果单击 TimeQuest Timing Analyzer 项的“＋”号，则能通过单击下面列出的各项目看到当前工程所有相关时序特性报告。

如果单击 Fitter 项的“＋”号，则能通过单击下面列出的各项看到当前工程所有相关硬件特性适配报告。

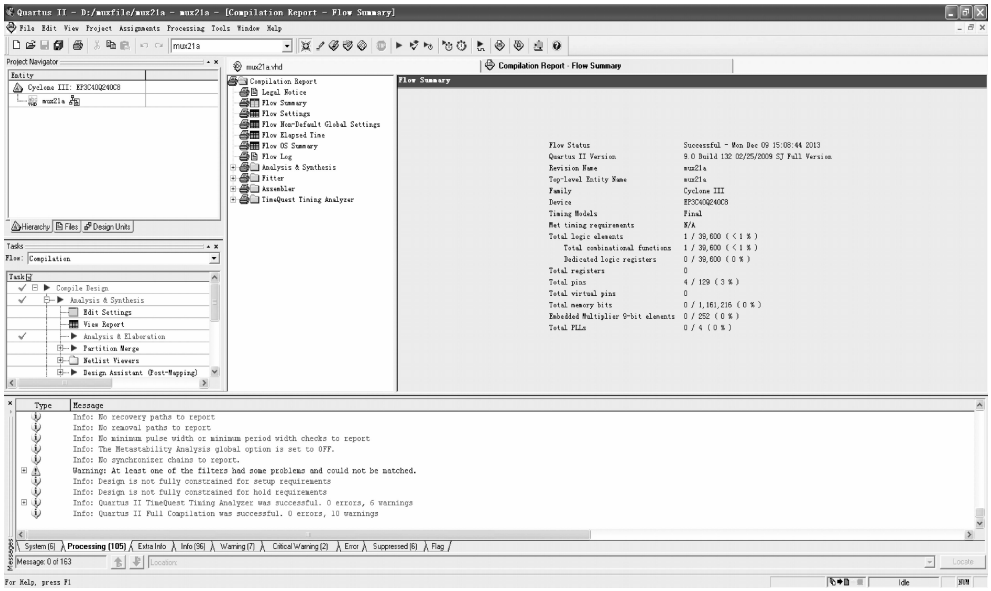


图 3-13　编译成功信息

5）仿真

对工程的编译通过后，必须对其功能和时序性质进行仿真测试，以了解设计结果是否满足原设计要求。以 VWF 文件方式的仿真流程的详细步骤如下。

（1）打开波形编辑器。选择主菜单 File→New 项，在 New 对话框中选择 Verification/Debugging Files 中的 Vector Waveform File（图 3-14），单击 OK 按钮，即出现空白的波形编辑器（图 3-15）。注意可以选择主菜单 View→Full Screen 或者单击工具栏上的“全屏窗口”按钮将窗口扩大或缩小，以利于观察仿真波形。

（2）设置仿真时间区域。在选择主菜单 Edit→End Time 项，在弹出的对话框中的 Time 文本框中输入“50”，单位选择“μs”，如图 3-16 所示，即整个仿真域的时间设定为 50μs，单击 OK 按钮，结束设置。对于时序仿真来说，将仿真时间轴设置在一个合理的时间区域上十分重要。通常设置的时间范围在数十微秒间。如果电路功能比较复杂，需要验证的功能较多，则需要适当加长仿真时间。仿真时间设置越长，将会导致仿真的耗时越长，如果设置为几秒，则实际仿真电路需要的时间可能需要几个小时，因此仿真时间设置一定要合理。

（3）存盘波形文件。选择 File 中的 Save As，将以名为 mux21a.vwf（默认名）的波形文件存入文件夹 D:\muxfile 中。

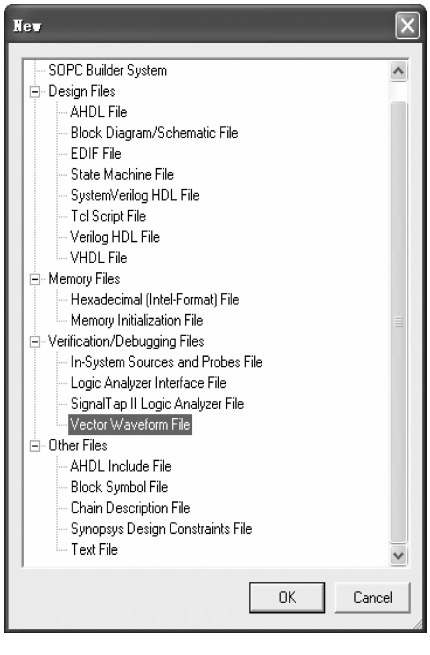


图 3-14　建立矢量波形文件

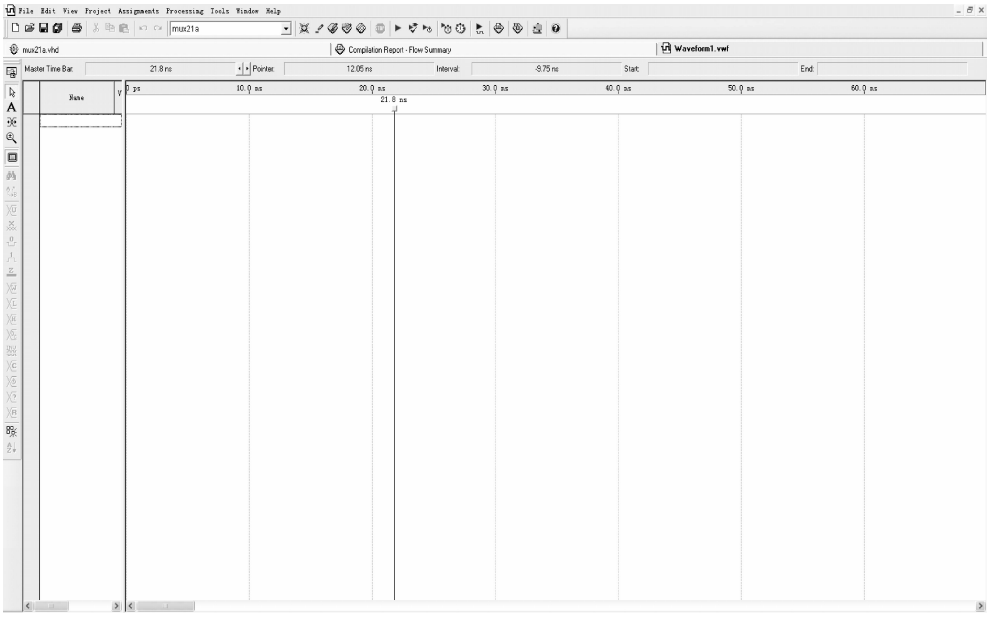


图 3-15 波形编辑器

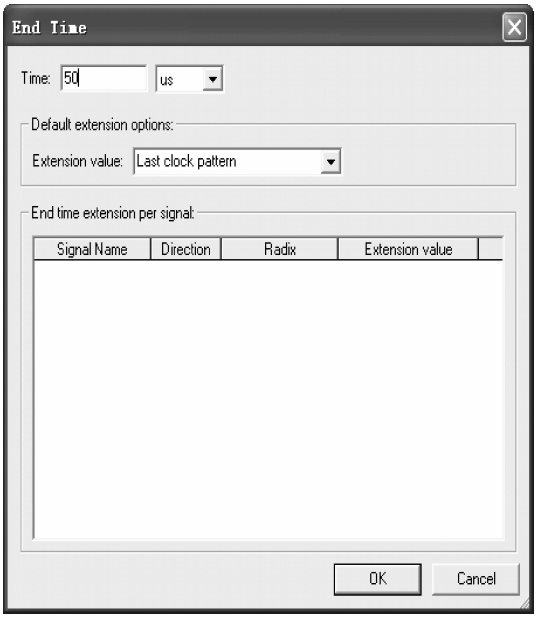


图 3-16 设置仿真时间长度

(4) 输入信号节点。将工程 mux21a 的端口信号节点选入波形编辑器中，主要有以下两种方法(可以任选一种)。

方法一：首先选择主菜单 View→Utility Windows→Node Finder 选项，出现如图 3-17 所示的 Node Finder 窗口。单击 List 按钮，可以在下面的 Node Found 栏中看到 4 个引脚。可以用鼠标左键单击 Node Found 栏中的引脚，拖曳到 mux21a. vwf 编辑窗口中的 Name

栏，添加完引脚的 mux21a.vwf 编辑窗口如图 3-18 所示。

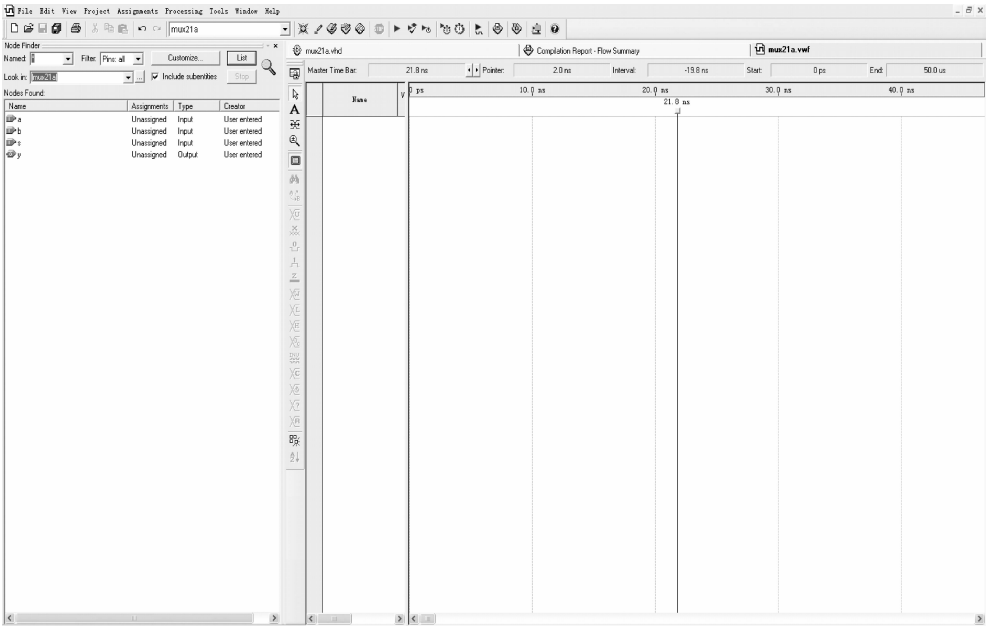


图 3-17 加入输入/输出节点窗口(Node Finder 窗口)

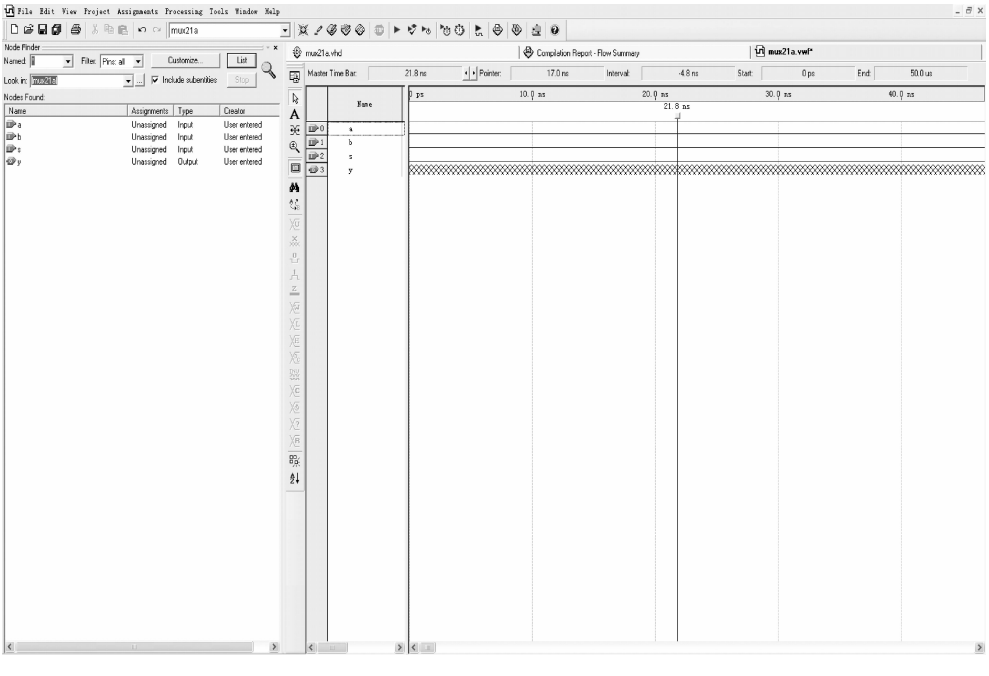


图 3-18 添加完引脚的波形编辑窗口

方法二：双击图 3-15 波形编辑器左侧栏(Name 栏)的空白处，在弹出的窗口中单击 Node Finder 按钮后出现如图 3-19 所示的对话框。在 Filter 框中选择“Pins：All”，然后单击

List 按钮。于是在左侧 Nodes Found 栏中出现了设计中的 mux21a 工程的所有端口引脚名(如果此对话框中的 List 不显示,需要重新编译一次,即选择主菜单 Processing→Start Compilation,然后重复以上操作过程)。用鼠标单击窗口中间的 » 按钮,可将 Nodes Found 栏中所有的节点加入到右侧 Selected Nodes 栏中,也可以先选中 Nodes Found 栏中的某个信号,单击 › 按钮,将其加入到 Selected Nodes 栏中。最后单击 OK 按钮,完成仿真波形节点信号的添加。

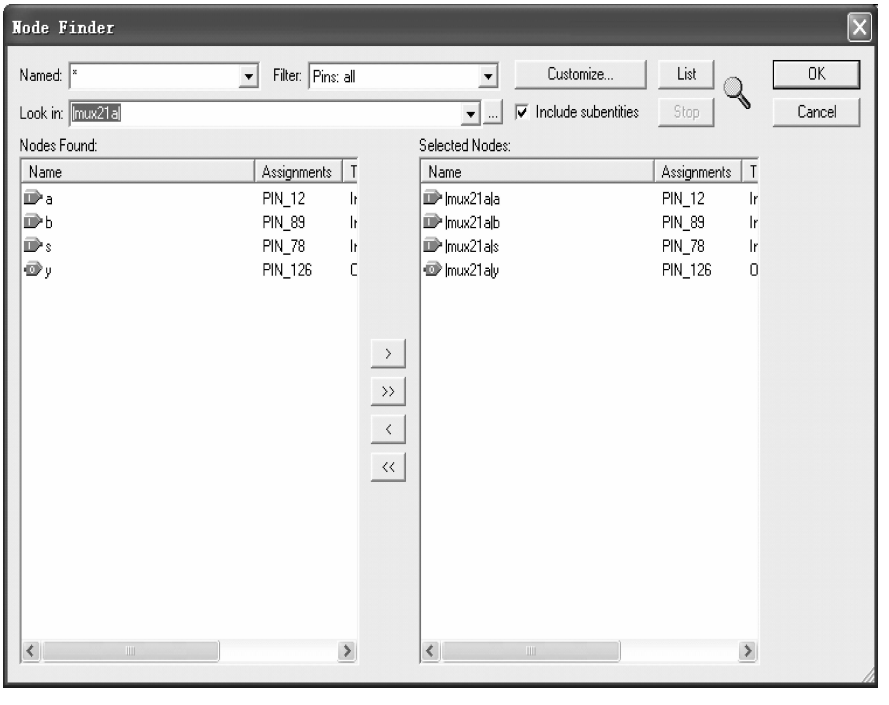


图 3-19 准备向波形编辑器输入信号节点

(5) 编辑输入波形。单击波形窗口左侧的"全屏显示"按钮,使波形编辑器全屏显示,并单击"放大缩小"按钮 后,在波形编辑区域用鼠标左键单击(放大)或右键单击(缩小),使仿真坐标处于适当位置。这时仿真时间横坐标设定在数十微秒数量级。

在 Name 栏中用鼠标左键单击端口信号 a,使之变蓝色,单击编辑器工具栏上的"高电平"按钮,使 a 信号设置为高电平。编辑器工具栏上的按钮与主菜单 Edit→Value 的功能一致,其常用功能说明如图 3-20 所示。利用编辑器工具栏可以方便地对输入信号进行编辑,设置不同的值。同理设置 b 信号为低电平(默认为低电平),然后按下面方法手动设置 s 信号先为高电平,一段时间后变成低电平,首先单击波形编辑器工具栏上的选择工具(白色箭头)使得鼠标处于选择状态,然后用鼠标左键单击 s 信号的波形(不是 Name 栏中的 s 信号,而是信号后面的直线),向后拖动一段,大约 25μs,即选中这段波形(变成蓝色),如图 3-21 所示,再单击工具栏上的"高电平"按钮,设置这段 s 信号为高电平。编辑完成的波形图如图 3-22 所示。

(6) 仿真器参数设置(可以选择默认设置,忽略此步骤)。选择主菜单 Assignment→Settings,在 Settings 窗口中选择 Category→Simulator Settings,如图 3-23 所示。可以设置仿真模式 Time Mode,默认为 Time。还可以进行毛刺检测 Glitch Detection(1ns 宽度)等设置。

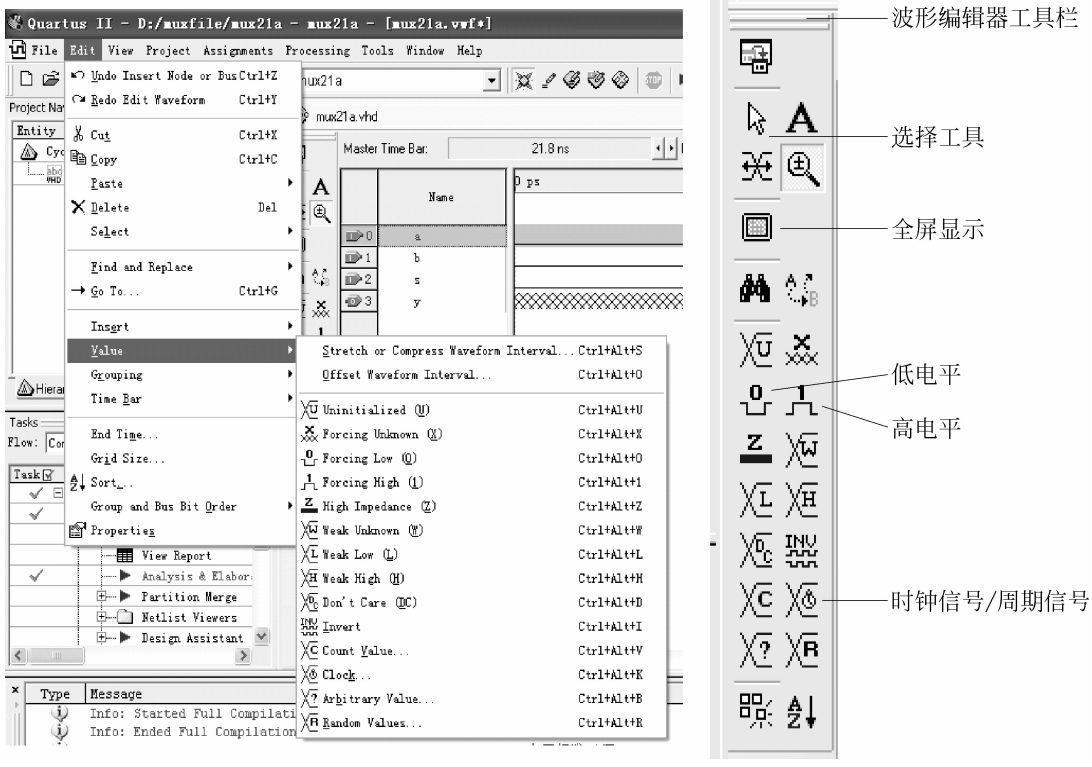


图 3-20 波形编辑器工具栏说明

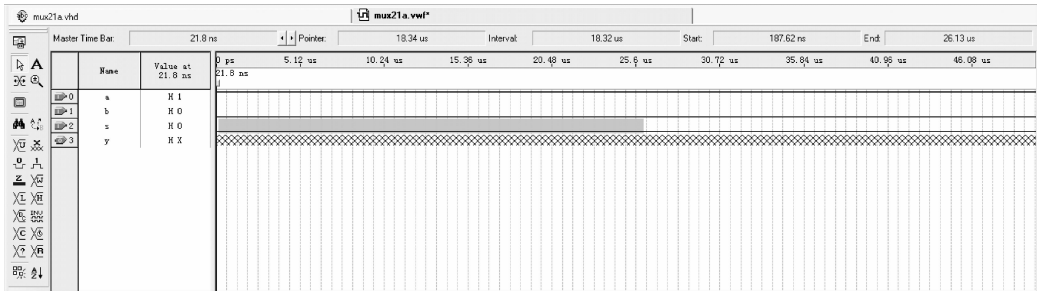


图 3-21 手动设置 s 信号

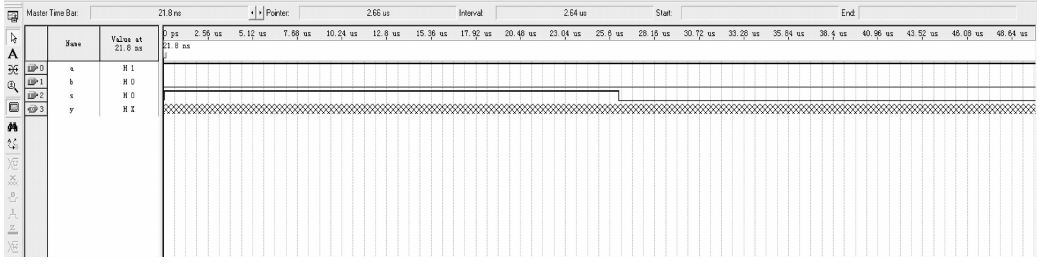


图 3-22 仿真前波形

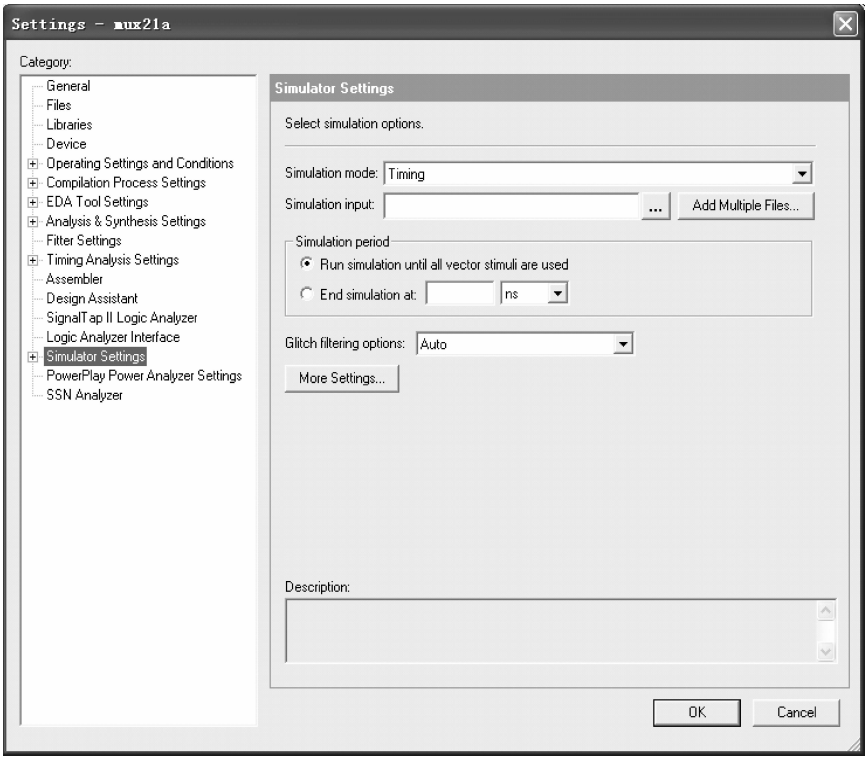


图 3-23　设置仿真参数

(7) 启动仿真器。所有设置完毕，在选择主菜单 Processing→Start Simulation，或者单击 Quartus Ⅱ工具栏上的 按钮，开始仿真，直到出现 Simulator Was Successful，仿真结束。

(8) 观察仿真结果。仿真波形文件 Simulation Waveforms 窗口通常会自动弹出，如图 3-24 所示。如果不小心关闭了 Simulation Waveforms 窗口，可以重新启动仿真器，或者选择主菜单 Processing→Simulation Report，打开仿真波形报告。

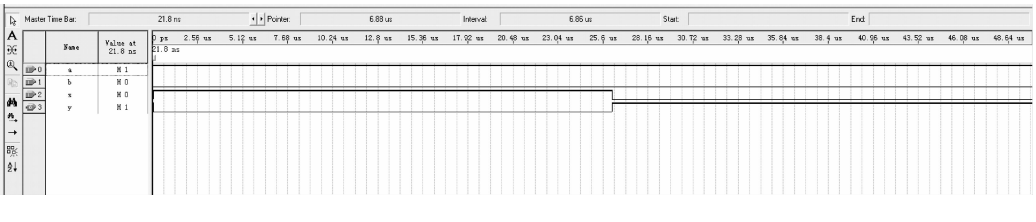


图 3-24　mux21a 仿真结果

从波形图可以看出，mux21a 电路共仿真运行了 50μs，前 25μs 这段时间 s 信号为高电平时，y 选择输出 b 的信号，即低电平，当 s 信号为低电平时，选择输出 a 的信号，即高电平，证明了此 2 选 1 选择器(mux21a)的“2 选 1”功能。

注意：Quartus Ⅱ的仿真波形文件中，波形编辑文件(*.vwf)与波形仿真报告文件(Simulation Report)是分开的。如果在启动仿真(Processing→Run Simulation)后，并没有出现仿真完成后的波形图，而是出现文字“Can't Open Simulation Report Window”，但报告

仿真成功，则可自己打开仿真波形报告，选择主菜单 Processing→Simulation Report。

如果无法展开波形显示时间轴上的所有波形图或者波形图上一片空白，可以单击 Simulation Waveforms 窗口左侧工具栏上的“放大缩小”按钮，用鼠标在波形上左键单击或右键单击，对波形图进行放大和缩小。如果波形无法放大或缩小，可以用鼠标右键单击波形编辑窗口中任何位置，这时再选择弹出窗口的 Zoom 项，在出现的下拉菜单中选择 Fit in Window 选项(图 3-25)。如果即使缩小波形也不能显示电路的全部功能，可能是仿真时间太短，参考步骤(2)设置仿真时间区域，设置合理的仿真时间。

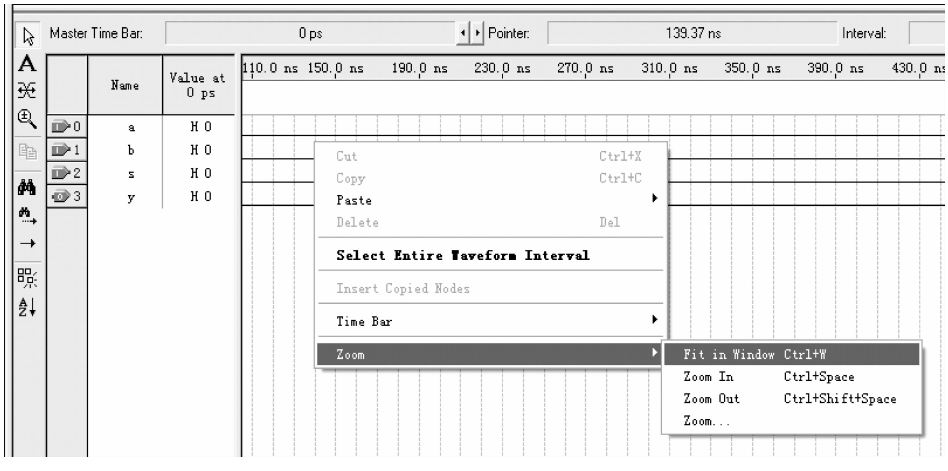


图 3-25 使波形显示适合窗口

(9) 在上面的仿真中，a、b 的信号在整个仿真过程中是不变的，恒为高电平或者低电平，为了进一步练习编辑输入波形的方法，下面重新编辑输入波形，修改 a、b 信号为周期变化的信号，再进行一次仿真，查看仿真结果。

首先打开仿真前波形文件“mux21a. vwf”，可以选择主菜单 File→Open，选择打开工程所在目录下的 mux21a. vwf 或者在 Project Navigator 窗口的 Files 页面(图 3-26)中双击打开 mux21a. vwf。然后在波形编辑器的 Name 栏中用鼠标左键单击端口信号 a，使之变蓝色，再选择主菜单 Edit→Value→Clock 选项，或者在波形编辑器工具栏上单击“时钟信号”按钮(图 3-20)在弹出的 Clock 窗口中设置 clk 的周期为 3μs；Duty Cycle 是占空比，可选择“50”，即 50%占空比，如图 3-27 所示。同样方法设置端口信号 b 的周期为 6μs，前面步骤(5)中，s 信号在整个仿真周期变化了一次，先高后低，下面设置 s 信号再变化一次，可以采用手动方法设置(参考步骤(5))，也可以同 a 信号一样利用时钟信号方法来设置，如设置时钟周期为 25μs。编辑好的波形如图 3-28 所示，然后保存文件。

参考步骤(7)，重新启动仿真器，出现仿真结果如图 3-29 所示。从波形图可以看出，当 s 信号为高电平时，y 的波形与 b 一致，即 y 输出 b 的信号，当 s 信号为低电平时，y 的波形与 a 一致，即 y 输出 a 的信号，证明了此 2 选 1 选择器(mux21a)的“2 选 1”功能。

在输出波形图上，利用“标线”可以定位任意时刻的仿真情况，如图 3-29 所示，在 10μs 时刻，各个信号的值为 a=0、b=1、s=0、y=1，与 2 选 1 选择器的真值表(表 2-8)一致。注意：单击工具栏上的箭头按钮使鼠标处于选择状态，可以拖动“标线按钮”到指定时间位置。

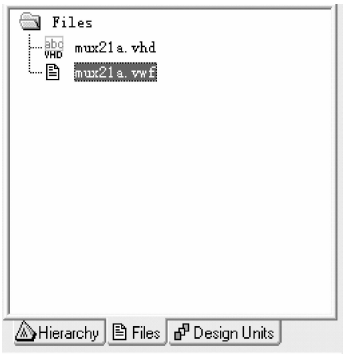


图 3-26 在工程管理器中选择打开仿真前波形文件 mux21a. vwf

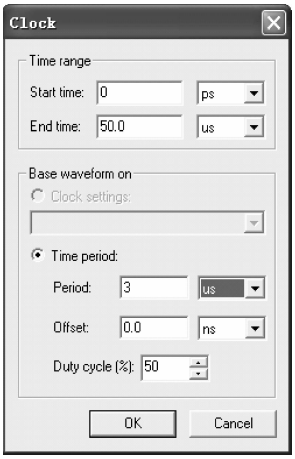


图 3-27 设置时钟信号的周期

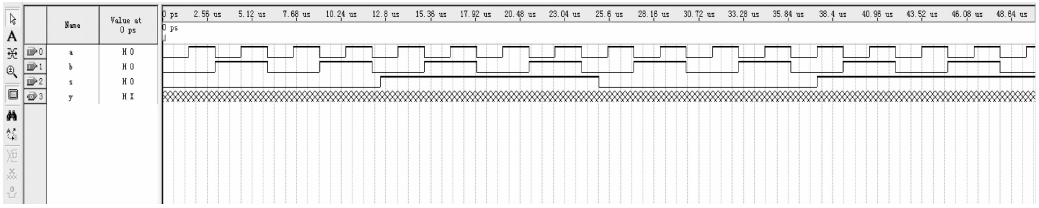


图 3-28 mux21a 仿真前波形(时钟信号)

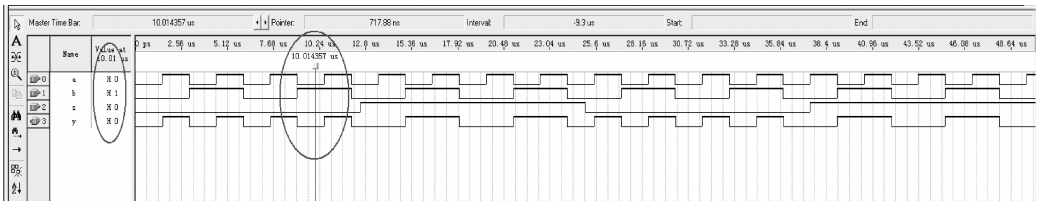


图 3-29 仿真后波形(时钟信号)

6) 应用 RTL 电路图观察器

Quartus Ⅱ中的 RTL Viewer 工具可生成硬件描述语言或网表文件(VHDL、Verilog、BDF、TDF、EDIF、VQM)对应的 RTL 电路图。通常,VHDL 代码或原理图经过编译后,功能仿真之前,可以利用 RTL Viewer 分析电路逻辑行为是否符合用户的要求。方法如下。

选择主菜单 File→Open,选择打开工程所在目录下的 mux21a. vhd 或者在 Project Navigator 窗口的 Files 页面(图 3-26)中双击打开 mux21a. vhd。选择主菜单 Tools→Netlist Viewer→RTL Viewer 项,可以打开 mux21a 工程各层次的 RTL 电路图(图 3-30)。双击图形中有关模块,或选择左侧各项,可逐层了解各层次的电路结构。

7) 编程配置、下载与硬件测试

为了能对此选择器进行硬件测试,应将其输入/输出信号锁定在芯片确定的引脚上,编

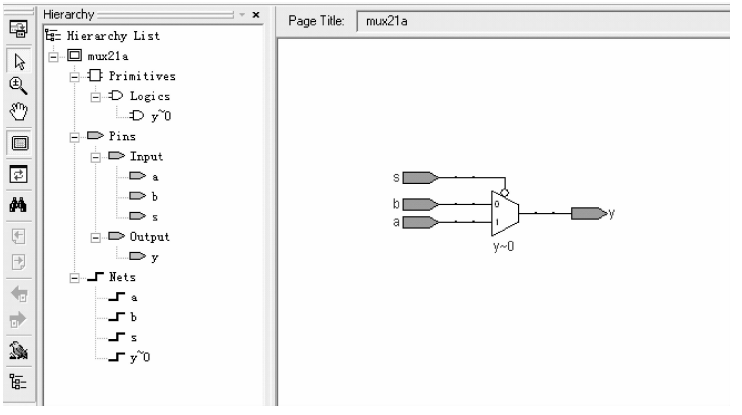


图 3-30 mux21a 工程的 RTL 电路图

译后下载。当硬件测试完成后，还必须对配置芯片进行编程，完成 FPGA 的最终开发。

(1) 引脚锁定。为了能对选择器进行硬件测试，应将其输入/输出信号锁定在芯片确定的引脚上。在此选择 GW48-CP++实验台的电路模式 No.5，即用键 1 控制信号 s；a、b 分别接 Clock5 和 Clock0；输出信号 y 接扬声器 Speaker。根据上面确定的实验模式锁定 mux21a 在目标芯片中的具体引脚。通过查阅 2.3.3 节的图 2-46 和 2.3.4 节的芯片引脚分配表(表 2-14)来确定。mux21a 工程的引脚锁定方案如表 3-1 所示。

表 3-1 mux21a 引脚锁定方案

输入/输出信号	外　　设	引 脚 名 称	引　脚　号
s	按键 1	PIO0	18
a	Clock5	Clock5	150
b	Clock0	Clock0	152
y	LED 灯 1	Speaker	13

打开 mux21a 工程(如果刚打开 Quartus Ⅱ，应选择主菜单 File→Open Project 项，并单击工程文件 mux21a)，选择主菜单 Assignments→Pin 项，弹出如图 3-31 所示的对话框，在 Location 处填写每个信号的引脚号，引脚锁定完毕如图 3-32 所示，最后存盘，关闭对话框。引脚锁定后，必须再编译一次(选择主菜单 Processing→Start Compilation)，将引脚锁定信息编译进下载文件中。如果图 3-31 中没有 Location 列，则使用鼠标右键单击 Node Name 栏，在弹出右键菜单中选择 Customize Columns 选项，如图 3-33 所示。然后按照图 3-34 和图 3-35 添加 Location 列。

(2) 选择编程器。用数据线连接 GW48-CP++实验台与计算机，并打开实验台电源。在 Quartus Ⅱ中选择主菜单 Tool→Programmer 项，弹出如图 3-36 所示的编程窗口。如果窗口上方显示 ByteBlasterMV[LPT1]，则忽略以下步骤，否则如果显示 No Hardware 或其他方式，则单击编程窗口上的 Hardware Setup 按钮，即弹出 Hardware Setup 对话框，如图 3-37 所示；选择此框的 Hardware Settings 选项卡，单击 Add Hardware 按钮，在弹出的窗口中选择 ByteBlaster，单击 OK 按钮，再双击 ByteBlasterMV 选项，使 Currently selected hardware 右侧显示 ByteBlasterMV[LPT1]，如图 3-38 所示，单击 Close 按钮，关闭对话框即可。这时应该在编程窗口右上显示出编程方式“ByteBlasterMV[LPT1]”，如图 3-39 所示。

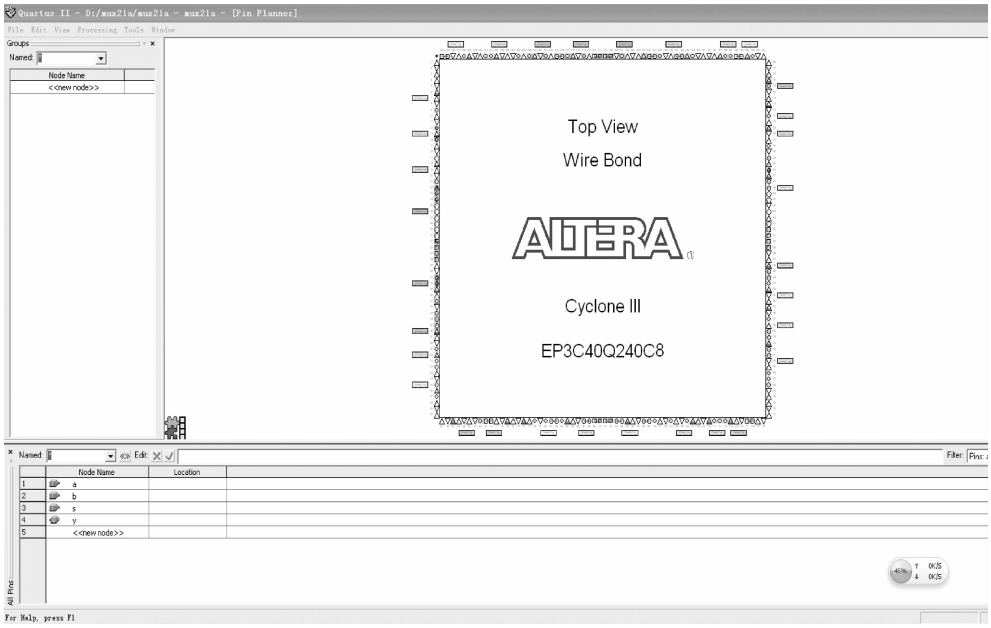


图 3-31　引脚锁定编辑器 Pin Planner 窗口

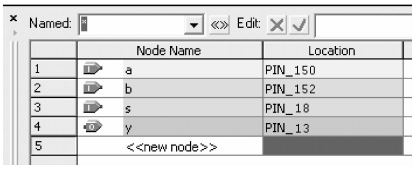


图 3-32　引脚锁定完毕

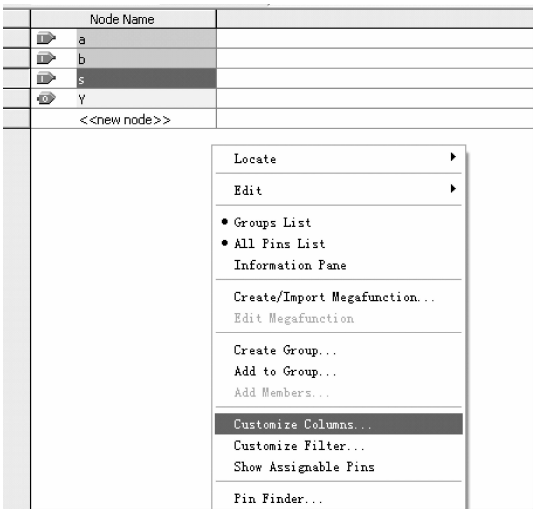


图 3-33　自定义列命令

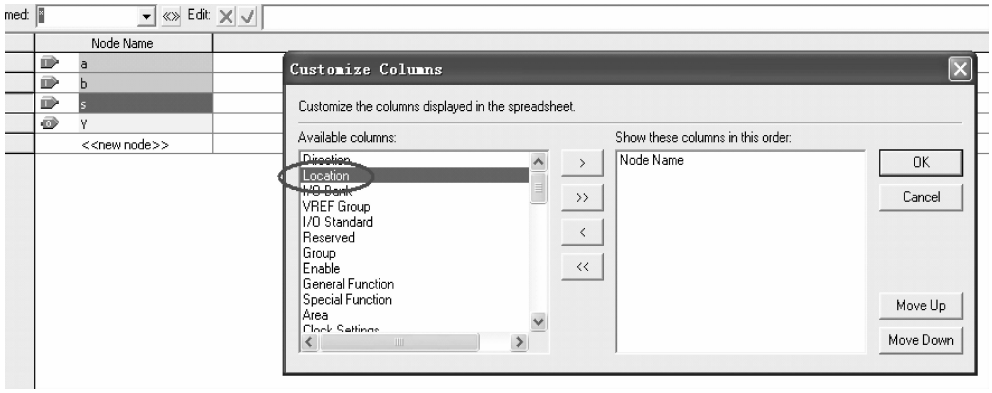


图 3-34 “添加 Location”窗口

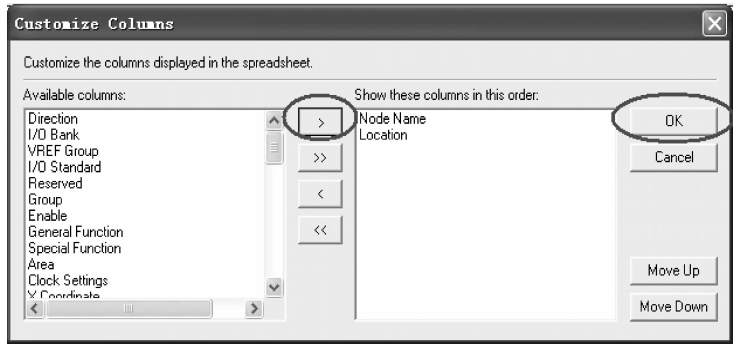


图 3-35 添加 Location 列

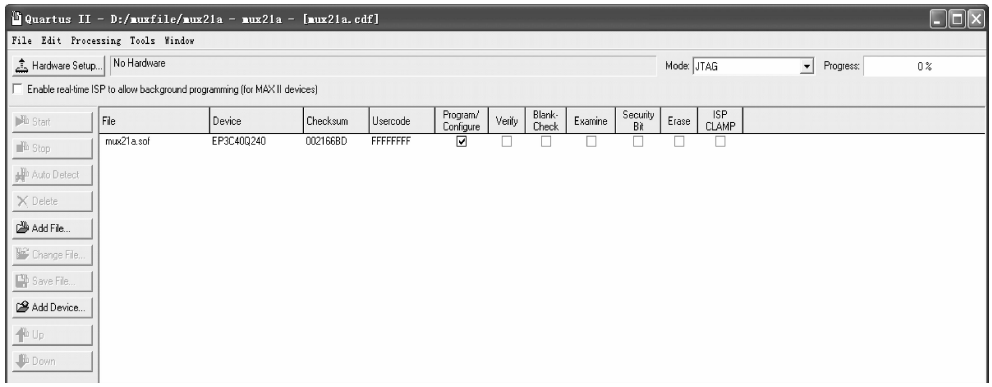


图 3-36 编程窗口

(3) 选择编程模式及配置并进行编程。为了将编译产生的下载文件配置进 FPGA 中进行测试,要将系统连接好,上电,然后在图 3-39 所示的编程窗口中,在 Mode 栏中有 3 种编程模式可以选择:JTAG、Passive Serial 和 Active Serial。为了直接对 FPGA 进行配置,在编程窗口的编程模式 Mode 中选择“JTAG”(默认),并选中下载文件右侧的第一小方框,如图 3-39 所示。注意要仔细核对下载文件路径与文件名。如果此文件没有出现或有错,单击左侧 Add File 按钮,手动选择配置文件 mux21a.sof。如果出现多个 mux21a.sof 文件,删除

图 3-37 Hardware Setup 对话框

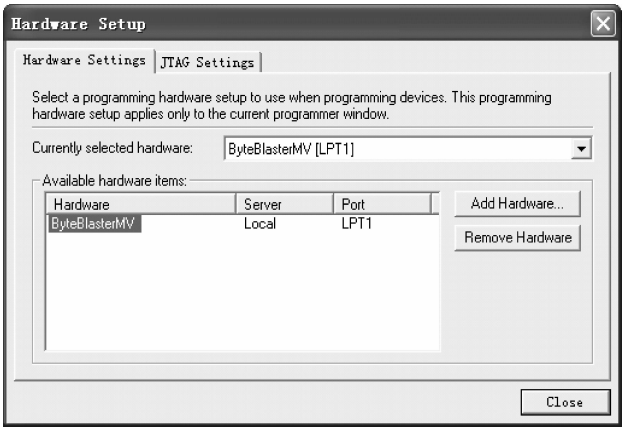


图 3-38 选择编程方式

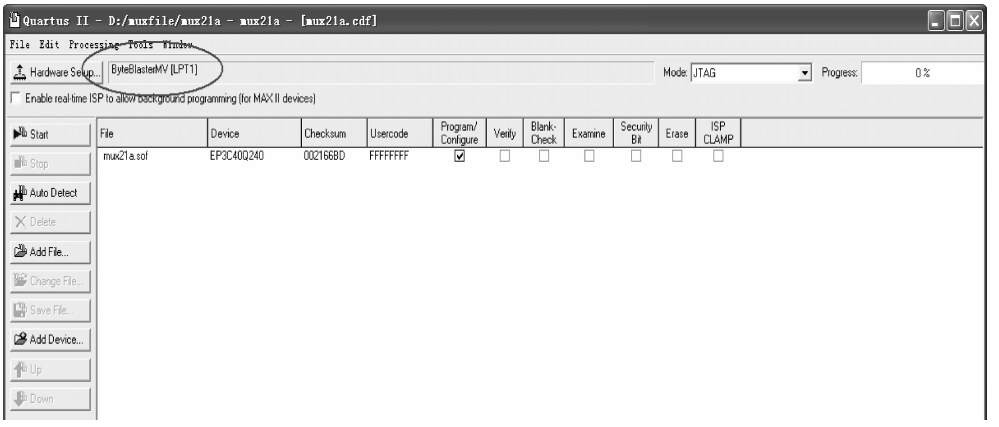


图 3-39 选择编程方式后的编程器窗口

多余的其他路径下面的 Sof 文件，确保 mux21a.sof 是想要下载到实验台测试的电路。最后单击下载标符 Start 按钮，即进入对目标器件 FPGA 的配置下载操作。当 Progress 栏中显示出 100%，以及在底部的处理栏中出现 Configuration Succesful 时，表示编程成功，如图 3-40 所示。单击下载标符 Start 按钮后，Progress 栏中一直显示为 0%，并且在 Quartus Ⅱ 主窗口下面的 Message 窗口的 System 栏显示出错信息，如图 3-41 所示。则说明下载失败，一般是下面的一个或多个原因：①没有选择 ByteBlasterMV[LPT1]编程器；②没有打开实验台电源；③数据线与计算机并口松动；④数据线与实验台连接不正确（正确的连接方式：数据线端口中间有凸起的一面向上）没有连接好的缘故。

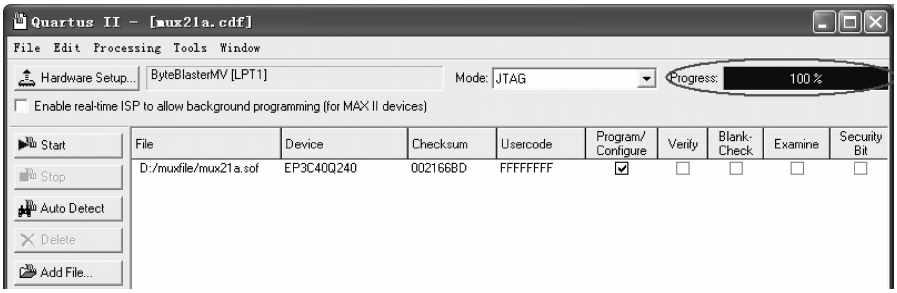


图 3-40　编程成功

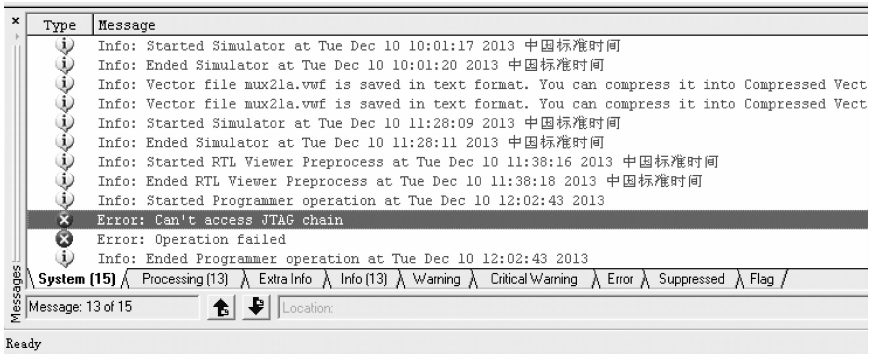


图 3-41　编程失败

（4）硬件测试。下载后，按动实验台左下角的“模式选择”按钮，选择实验电路模式“No. 5”，在实验台右上角的时钟区，用短路帽设定 Clock5 和 Clock0 的频率分别为 1024Hz 和 256Hz。注意：一个时钟上不能同时选择两个频率，即不能同时放两个短路帽。当用键 1 输入高电平时，扬声器发出 256Hz 低频声，当用键 1 输入低电平时，扬声器发出 1024Hz 高频声。

思考：扬声器为什么会发出不同的声音，分析电路的功能和原理。

3.1.2　利用 VHDL 设计十进制加法计数器

1. 实验目的

（1）熟悉 Quartus Ⅱ 开发环境及开发流程。

(2) 掌握 Quartus Ⅱ 中 VHDL 文本输入设计方法。

(3) 熟悉 FPGA 实验台的功能和使用方法。

2. 实验内容

在 QuartusⅡ中,利用 VHDL 设计一个带有异步复位和同步时钟使能的十进制加法计数器,完成仿真,最后在实验台上进行硬件测试,验证计数器的功能。

3. 实验环境

PC	1 台
Quartus Ⅱ 软件环境	1 套
GW48-CP++实验台	1 台

4. 实验原理与电路

带有异步复位和同步时钟使能的十进制加法计数器是比较常见的时序逻辑电路,所谓同步或异步都是相对于时钟信号而言的。不依赖于时钟而有效的信号称为异步信号,否则称为同步信号。计数器状态转移图如图 3-42 所示,状态转移表如表 3-2 所示。计数器元件符号如图 3-43 所示。其中,clk 为时钟脉冲,rst 为异步复位信号,当 rst 为高电平,计数器清零,en 为同步使能信号,高电平有效,cq[3..0]为计数器 4 位输出,cout 为进位标志,当计数器计数到 9 时,cout 为 1。

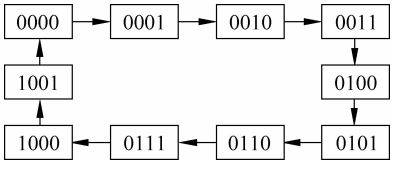


图 3-42 计数器状态转移图

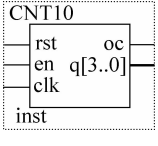


图 3-43 计数器元件符号

表 3-2 计数器状态转移表

Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

5. 实验任务与步骤

1）建立工程

首先建立一个文件夹，以便将所有设计文件存放于该文件夹中，如 D:\cnt10。注意文件夹名不要用中文，文件夹名称中不要含有空格。然后运行 Quartus Ⅱ软件，选择主菜单 File→New Project Wizard，利用向导创建工程。如图 3-44 所示，指定工程存放目录为 D:\cnt10，工程名和实体名可取 cnt10。其他设置选项可以保持默认值，创建工程以后再设置。注意第 3 行顶层实体名称一旦确定，后续的顶层实体文件名必须与此保持一致。

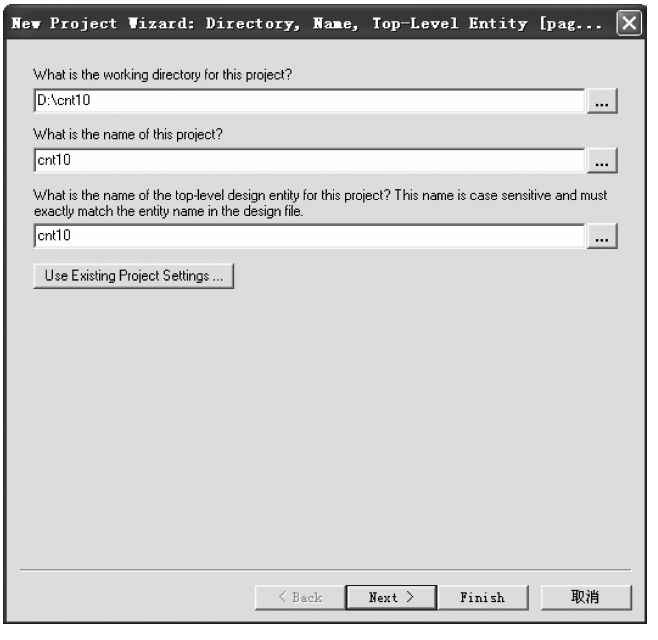


图 3-44　创建工程 cnt10

2）VHDL 设计输入

（1）新建 VHDL 文件，打开文本编辑器。具体步骤：主菜单 File→New，弹出对话框，选择 VHDL Files 文件类型，如图 3-45 所示。完成后，即打开框图编辑器，并生成一个空白的框图文件，保存该文件为 cnt10.vhd。

注意：该文件作为工程的设计文件，其名称要和实体名一致，实体名在创建工程时已经指定。

（2）利用文本编辑器在 cnt10.vhd 文件中录入程序 P3-1 的代码。然后选择主菜单 File→Save，保存文件 cnt10.vhd。

程序 P3-1　cnt10.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity cnt10 is
```

```
    port( rst,en,clk:in std_logic;
              oc:out std_logic;
              q:out std_logic_vector(3 downto 0));
end cnt10;
architecture bhv of cnt10 is
begin
  process(rst,en,clk)
  variable tmp:std_logic_vector(3 downto 0);
  begin
    if rst = '1' then
      tmp := "0000";
      oc <= '0';
    elsif en = '0' then null;
    elsif clk'event and clk = '1' then
      if tmp = 9 then oc <= '1';
        tmp := "0000";
        else tmp := tmp + 1;
        oc <= '0';
      end if;
  end if;
  q <= tmp;
  end process;
end bhv;
```

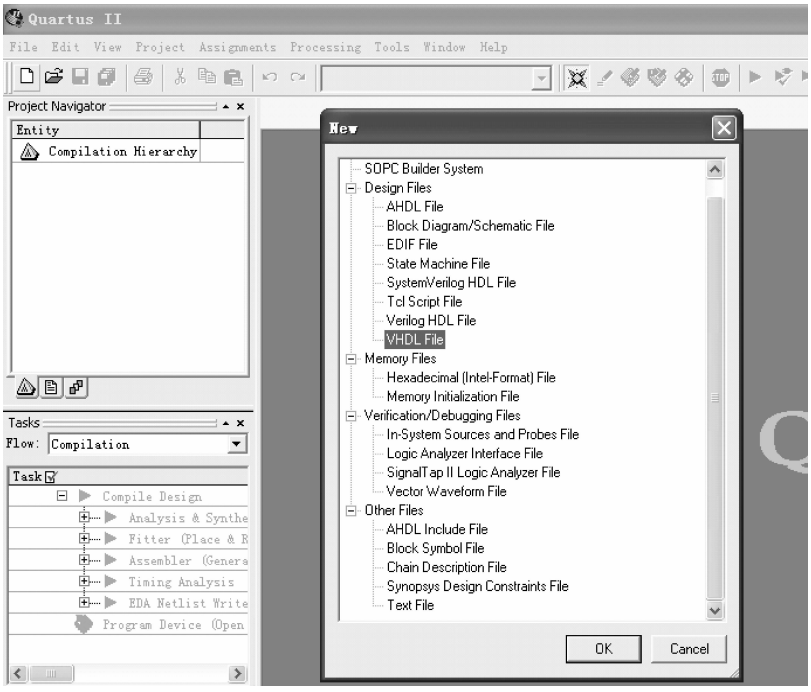


图 3-45　选择编辑文件的语言类型

3）器件选择、编译

选择主菜单 Assignments→Device 项，对器件进行配置，选择 Cyclone Ⅲ系列 EP3C40Q240C8 型号。选择主菜单 Processing→Start Compilation 或者单击工具栏按钮▶，对整个工程设计进行编译。如果 cnt10. vhd 中的逻辑电路有错误，在下方的 Processing 处理栏中会显示出来。双击错误提示，定位出错的位置，改正错误后再编译工程，直到编译成功。

4）仿真

对工程的编译通过后，必须对其功能和时序性质进行仿真测试，以了解设计结果是否满足原设计要求。以 VWF 文件方式的仿真流程的详细步骤如下。

（1）打开波形编辑器。选择主菜单 File→New 项，在 New 窗口中选择 Verification/Debugging Files 中的 Vector Waveform File，单击 OK 按钮，即出现空白的波形编辑器。

（2）设置仿真时间区域。在主菜单 Edit→End Time 项，在弹出的窗口中的 Time 窗口中输入合适的数字，即仿真时间，如 100，单位选择“μs”。

（3）存盘波形文件。选择 File 中的 Save As，将以名为 cnt10. vwf（默认名）的波形文件存入文件夹 D:\cnt10 中。

（4）输入信号节点。将工程 cnt10 的端口信号节点选入波形编辑器中。方法是首先选择主菜单 View→Utility Windows→Node Finder 选项。在 Filter 框中选择 Pins：All，然后单击 List 按钮。于是在左侧 Node Found 栏中出现了设计中的 mux21a 工程的所有端口引脚名（如果此对话框中的 List 不显示，需要重新编译一次，即选择 Processing→Start Compilation，然后重复以上操作过程）。可以用鼠标左键单击 Node Found 栏中的引脚，拖曳到 cnt10. vwf 编辑窗口中的 Name 栏。

（5）编辑输入波形。单击波形窗口左侧的全屏显示按钮，使全屏显示，并单击“放大缩小”按钮🔍后，在波形编辑区域用鼠标左键单击（放大）或右键单击（缩小），使仿真坐标处于适当位置。这时仿真时间横坐标设定在数十微秒数量级。单击波形编辑器窗口 Name 栏的时钟信号名 clk，使之变成蓝色条，再单击左列波形编辑器工具栏的“时钟信号”按钮（见图 3-20），在 Clock 窗口中设置 clk 的时钟周期为 2μs；Clock 窗口中的 Duty Cycle 是占空比，默认为“50”，即 50%占空比。然后再分别设置 en 和 rst 的电平。编辑完成的波形图如图 3-46 所示。

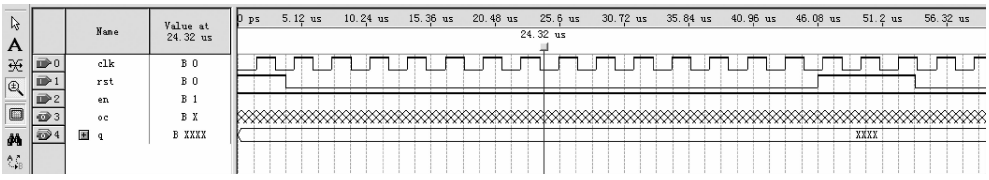


图 3-46 计数器仿真前波形图

（6）总线数据格式设置。单击如图 3-46 所示的输出信号 cq 左侧的“＋”，则能展开此总线中的所有信号；如果双击此“＋”号左侧的信号标记，将弹出对该信号数据格式设置的对话框。在该对话框的 Radix 栏有 4 种选择，这里可选择无符号十进制整数“Unsigned Decimal”表达方式。最后对波形文件再次存盘。

（7）仿真器参数设置（可以选择默认设置，忽略此步骤）。选择主菜单 Assignment→

Settings，在 Settings 窗口中选择 Category→Simulator Setting。可以设置仿真模式“Time Mode”，默认为“Time”。还可以进行毛刺检测“Glitch Detection”(1ns 宽度)等设置。

(8) 启动仿真器。所有设置完毕，在选择主菜单 Processing→Start Simulation，或者单击 Quartus Ⅱ工具栏上的 按钮，开始仿真，直到出现“Simulator Was Successful”，仿真结束。

(9) 观察仿真结果。仿真波形文件 Simulation Waveforms 窗口通常会自动弹出，如图 3-47 所示。

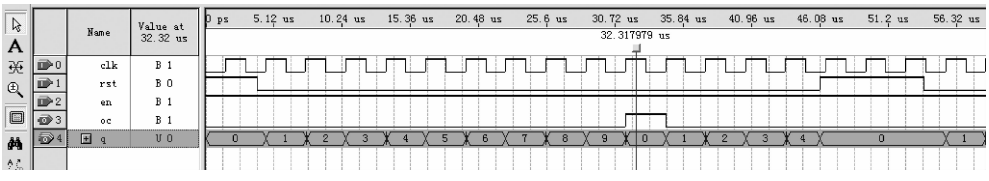


图 3-47 计数器仿真输出波形图

注意：Quartus Ⅱ的仿真波形文件中，波形编辑文件(*.vwf)与波形仿真报告文件(Simulation Report)是分开的。如果在启动仿真(Processing→Run Simulation)后，并没有出现仿真完成后的波形图，而是出现文字“Can't Open Simulation Report Window”，但报告仿真成功，则可自己打开仿真波形报告，选择主菜单 Processing→Simulation Report。

如果无法展开波形显示时间轴上的所有波形图或者波形图上一片空白，可以单击 Simulation Waveforms 窗口左侧工具栏上的“放大缩小”按钮，用鼠标在波形上左键单击或右键单击，对波形图进行放大和缩小。如果波形无法放大或缩小，可以用鼠标右键单击波形编辑窗口中任何位置，这时再选择弹出窗口的 Zoom 项，在出现的下拉菜单中选择 Fit in Window。如果即使缩小波形也不能显示电路的全部功能，可能是仿真时间太短，参考步骤(2)设置仿真时间区域，设置合理的仿真时间。

5) 应用 RTL 电路图观察器

Quartus Ⅱ可实现硬件描述语言或网表文件(VHDL、Verilog、BDF、TDF、EDIF、VQM)对应的 RTL 电路图的生成。方法如下。

选择主菜单 Tools→Netlist Viewer→RTL Viewer 项，可以打开 cnt10 工程各层次的 RTL 电路图，如图 3-48 所示。双击图形中有关模块，或选择左侧各项，可逐层了解各层次的电路结构。由图 3-48 可以看出，电路主要由一个比较器、一个加法器、4 个多路选择器和一个 4 位锁存器组成。

对于较复杂的 RTL 电路，可利用功能过滤器“Filter”简化电路。即用鼠标右键单击该模块，在弹出的下拉菜单中选择 Filter 项的 Sources 或 Destinations，由此产生相应的简化电路。

如果选择主菜单 Tools→Netlist Viewer→Technology Map Viewer 项，可以看到本工程的硬件电路图。

6) 引脚锁定、下载与硬件测试

在此选择 GW48-CP++实验台的电路模式 No. 5，根据 EP3C40Q240C8 芯片，通过查找引脚分配表(表 2-14)，确定计数器的引脚锁定方案如表 3-3 所示。主频时钟 clk 接

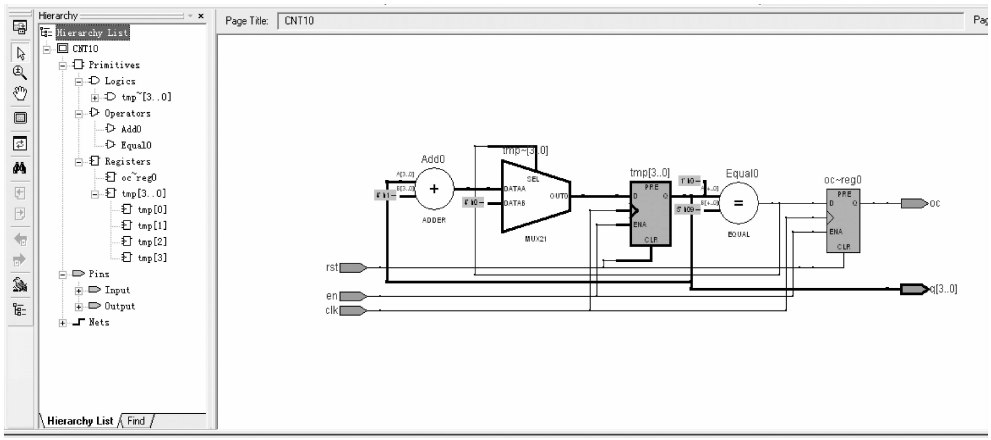


图 3-48 cnt10 工程的 RTL 电路图

Clock0(第 152 脚,可接在 4Hz 上);q[3..0]从高位到低位一次分配引脚号。如果高低位错乱,数码管将不会显示正确的数值。

表 3-3 计数器引脚锁定方案

输入/输出信号	外 设	引脚名称	引 脚 号
clk	Clock0	Clock0	152
rs	按键 2	PIO1	18
en	按键 3	PIO2	21
q[3..0]	数码管 1	PIO19、PIO18、PIO17、PIO16	69,63,57,56
oc	LED 灯 1	PIO8	44

选择主菜单 Assignments→Pin 项,弹出如图 3-31 所示的对话框,在 Location 处填写每个信号的引脚号,引脚锁定完毕如图 3-49 所示。如果窗口中没有出现 Location 列,参照 3.1.1 步骤 7)引脚锁定部分添加 Location 列,最后进行存盘,关闭对话框。引脚锁定后,必须再编译一次。

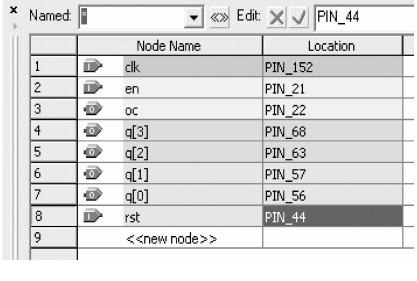


图 3-49 计数器引脚锁定

在 Quartus Ⅱ中选择主菜单 Tool→Programmer 项,单击编程窗口上的 Hardware Setup 按钮,选择编程器“ByteBlasterMV[LPT1]”,具体方法参见 3.1.1 的实验步骤 7)中的“选择编程器”部分。

单击下载标符 Start 按钮,即进入对目标器件 FPGA 的配置下载操作。注意要仔细核对下载文件路径与文件名。如果此文件没有出现或有错,单击左侧 Add File 按钮,手动选择配置文件 cnt10.sof。当 Progress 显示出 100%,以及在底部的处理栏中出现 Configuration Successful 时,表示编程成功。

下载后,按动实验台左下角的“模式选择”按钮,选择实验电路模式“No.5”,控制按键 1、

2、3，使计数器工作，观察数码管 1 和 LED 灯的输出，验证计数器的功能。

7）SignalTap Ⅱ实时测试

随着逻辑设计复杂性的不断增加，仅依赖于软件方式的仿真测试来了解设计系统的硬件功能已远远不够了，而不断需要重复进行硬件系统的测试也变得更为困难。为了解决这些问题，设计者可以将一种高效的硬件测试手段和传统的系统测试方法相结合来完成。这就是嵌入式逻辑分析仪的使用。它可以随设计文件一并下载于目标芯片中，用以捕捉目标芯片内部信号节点处的信息，而又不影响原硬件系统的正常工作。

这就是 Quartus Ⅱ中 SignalTap Ⅱ的目的。在实际监测中，SignalTap Ⅱ将测得的样本信号暂存于目标器件中的嵌入式 RAM（如 ESB、M4K）中，然后通过器件的 JTAG 端口将采得的信息传出，送入计算机进行显示和分析。

嵌入式逻辑分析仪 SignalTap Ⅱ允许对设计中的所有层次模块的信号节点进行测试，可以使用多时钟驱动，而且还能通过设置以确定前后触发捕捉信号信息的比例。

以下步骤描述设置 SignalTap Ⅱ文件和采集信号数据的基本流程。

（1）建立新的 SignalTap Ⅱ文件。

（2）向 SignalTap Ⅱ文件添加实例，并向每个实例添加节点。可以使用 Node Finder 中的 SignalTap Ⅱ滤波器查找所有预综合和布局布线后的 SignalTap Ⅱ节点。

（3）给每个实例分配一个时钟。

（4）设置其他选项，如采样深度和触发级别，并将信号分配给数据/触发输入和调试端口。

（5）如果必要，可指定 Advanced Trigger 条件。

（6）编译设计。

（7）对器件进行编程。

（8）在 Quartus Ⅱ软件中或使用外部逻辑分析仪或示波器采集和分析信号数据。

可以使用以下功能设置 SignalTap Ⅱ逻辑分析仪。

（1）多个逻辑分析仪：SignalTap Ⅱ逻辑分析仪在每个器件中支持逻辑分析仪的多个嵌入式实例。可以使用此功能为器件中的每个时钟域建立单独且唯一的逻辑分析仪，并在多个嵌入式逻辑分析仪中应用不同的设置。

（2）实例管理器：实例管理器允许在多个实例上建立并执行 SignalTap Ⅱ逻辑分析。可以使用它在 SignalTap Ⅱ文件中建立、删除和重命名实例。实例管理器显示当前 SignalTap Ⅱ文件中的所有实例、每个相关实例的当前状态以及相关实例中使用的逻辑元素和存储器比特的数量。实例管理器可以协助检查每个逻辑分析仪在器件上要求的资源使用量。可以选择多个逻辑分析仪以及选择 Run Analysis（Processing 菜单）来同时启动多个逻辑分析仪。

下面介绍如何利用 SignalTap Ⅱ测试上面的计数器电路的方法。

（1）打开 SignalTap Ⅱ编辑窗口。选择主菜单 File→New 项，在 New 窗口中选择 Verification/Debugging Files 中的 SignalTap Ⅱ File，单击 OK 按钮，即出现 SignalTap Ⅱ编辑窗口（如图 3-50 所示）。

（2）调入待测信号。首先单击上排的 Instance 栏内的 auto_signaltap_0，更改此名，如 cnts，这是其中一组待测信号名。为了调入待测信号名，在 cnts 栏下的空白处双击，即弹出

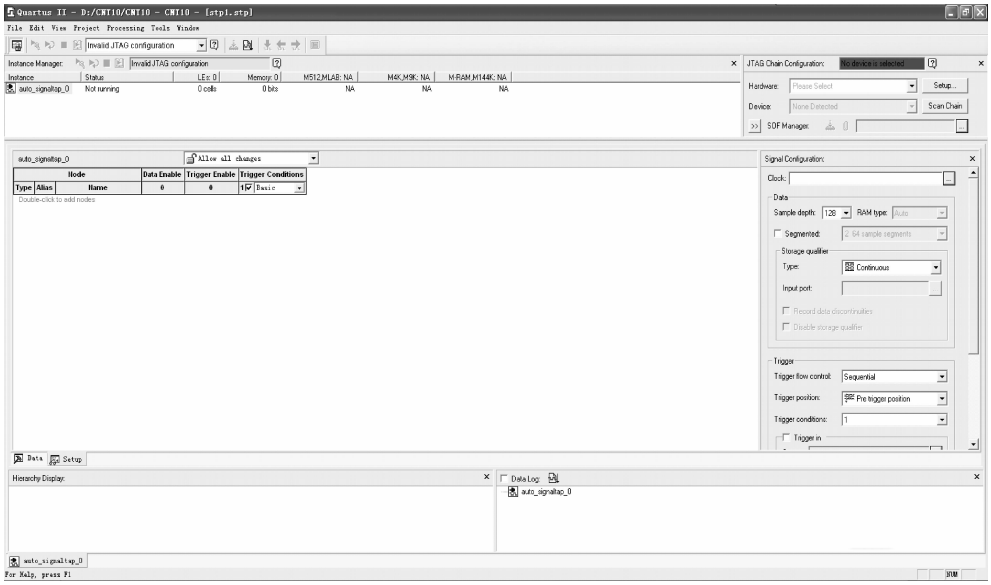


图 3-50 SignalTap Ⅱ编辑窗口

Node Finder 窗口，单击 List 按钮，即在左栏出现此工程相关的所有信号，包括内部信号。选择需要观察的信号名：4 位输出总线信号 q、内部 4 锁存器总线 tmp 信号和 oc。单击 OK 按钮后即可将这些信号调入 SignalTap Ⅱ信号观察窗，如图 3-51 所示。注意不要将工程的主频时钟信号 clk 调入信号观察窗，因为在本项设计中打算调用本工程的主频时钟信号 clk 兼作逻辑分析仪的采样时钟。此外如果有总线信号，只需调入总线信号名即可；慢速信号可不调入；调入信号的数量应根据实际需要来决定，不可随意调入过多的、没有实际意义的信号，这会导致 SignalTap Ⅱ无谓地占用芯片内过多的资源。

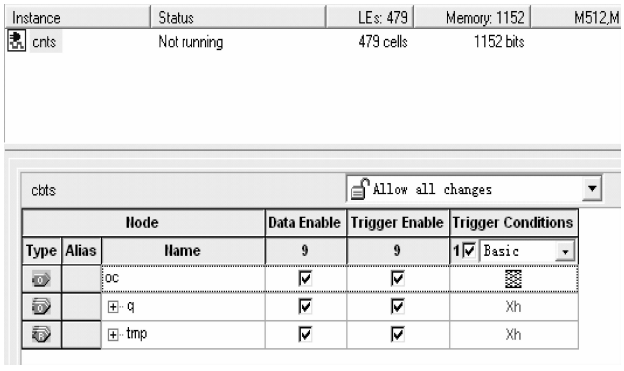


图 3-51 调入待测信号

(3) SignalTap Ⅱ参数设置。首先输入逻辑分析仪的工作时钟信号 Clock。在 SignalTap Ⅱ编辑窗口右侧单击 Clock 栏右侧的“…”按钮，即出现 Node Finder 窗口，选中工程的主频时钟信号 clk 作为逻辑分析仪的采样时钟；接着在 Data 框的 Sample 栏选择采样深度为 1Kb。注意这个深度一旦确定，则 cnts 信号组的每一位信号都获得同样的采样深

度，所以必须根据待测信号采样要求、信号组总的信号数量，以及本工程可能占用 ESB/M4K 的规模，综合确定采样深度，以免发生 M4K 不够用的情况。

然后是根据待观察信号的要求，在 Trigger 框的 Trigger Position 栏设定采样深度中起始触发的位置，如选择前点触发 Pre Trigger Position。

最后是触发信号和触发方式选择。这可以根据具体需求来选定，在 Trigger 框的 Trigger Conditions 栏选择“1”；选中 Trigger in 复选框，并在 Source 栏选择触发信号。在此选择 cnts 工程中的 en 作为触发信号；在 Pattern 栏选择上升沿触发方式(Rising Edge)，即当测得 en 的上升沿后，SignalTap Ⅱ在 clk 的驱动下根据设置 cnts 信号组的信号进行连续或单次采样。SignalTap Ⅱ参数设置后如图 3-52 所示。

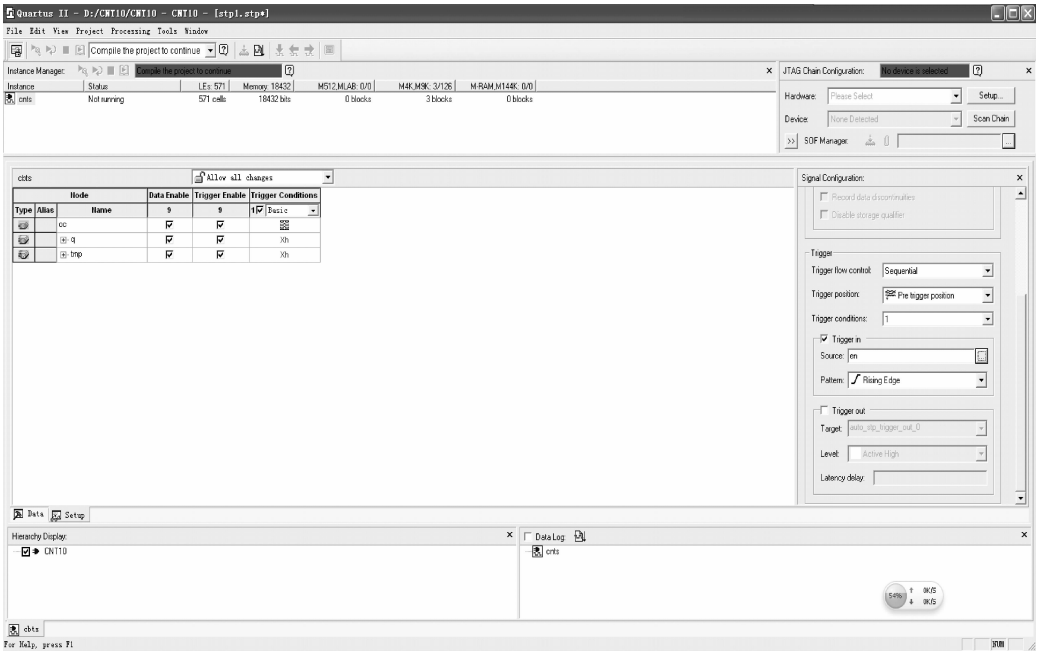


图 3-52 SignalTap Ⅱ 参数设置

(4) 文件存盘。选择主菜单 File→Save As 项，输入此 SignalTap Ⅱ文件名为 cnt10.stp1(默认名)。单击 Save 按钮后，将出现一个提示“Do You Want To Enable SignalTap Ⅱ”，应该单击 Yes 按钮。表示同意再次编译时将此 SignalTap Ⅱ文件(核)与工程(CNT10)捆绑在一起综合/适配，以便一同被下载进 FPGA 芯片中去完成实时测试任务。

如果单击 No 按钮，则必须自己去设置，方法是选择主菜单 Assignments 中的 Settings 项，在其 Category 栏中选择 SignalTap Ⅱ Logic Analyzer，在右侧窗口 SignalTap Ⅱ File Name 栏中选中已存盘的 SignalTap Ⅱ File 文件名，如 cnt10.stp1，并选中 Enable SignalTap Ⅱ Logic Analyzer 复选框，如图 3-53 所示。单击 OK 按钮即可。

但应该特别注意，当利用 SignalTap Ⅱ将芯片中的信号全部测试结束后，如在构成产品前，不要忘了将 SignalTap Ⅱ从芯片中除去。方法也是在图 3-53 所示的窗口中关闭 Enable SignalTap Ⅱ Logic Analyzer 项，再编译、编程一次即可。

(5) 编译下载。首先选择主菜单 Processing 的 Start Compilation 项，启动全程编译；

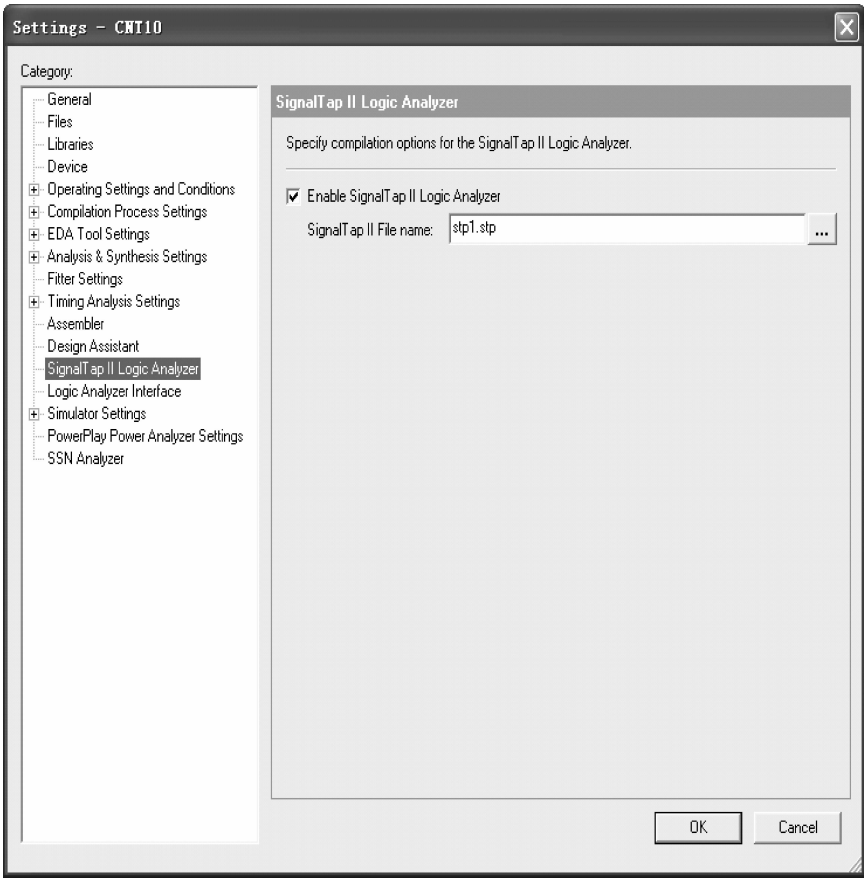


图 3-53　设定 SignalTap Ⅱ 与工程一同综合适配

然后选择主菜单 Tool→Programmer 项，单击下载标符 Start 按钮，下载 Sof 文件到目标器件。下载之前保证对工程进行了引脚分配，并且编程器选择了 ByteBlasterMV 方式。

然后选择主菜单 Tools 中的 SignalTap Ⅱ Analyzer，打开 SignalTap Ⅱ。接着打开实验开发系统的电源，连接 JTAG 口，设定通信模式。单击图 3-52 右上角的 Setup 按钮，选择硬件通信模式 ByteBlasterMV。然后单击下方的 Device 栏的 Scan Chain 按钮，对实验板进行扫描。如果在 Device 栏中出现板上 FPGA 的型号名，如图 3-54 所示，表示系统 JTAG 通信情况正常，可以进行测试。

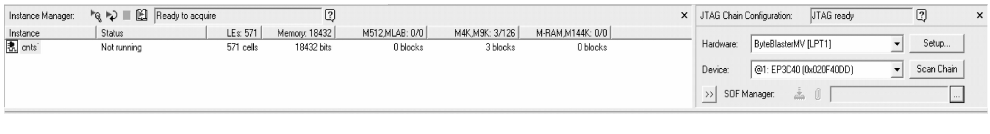


图 3-54　SignalTap Ⅱ 与实验台通信准备测试

设定实验台上的模式(模式 No. 5)和控制按键 1、2、3，使控制信号 en＝1、rst＝0，使计数器工作，clk 频率可在 Clock0 处设 65 536Hz 或更高。

(6) 启动 SignalTap Ⅱ进行采样与分析。如图 3-54 所示，单击 Instance 名“cnts”，再选

择主菜单 Processing Autorun Analysis 项或者单击工具栏上的 Autorun Analysis 按钮，启动 SignalTap Ⅱ。然后按键 1(en)，使由低到高，产生一个上升沿，作为 SignalTap Ⅱ的采样触发信号，这时就能在 SignalTap Ⅱ数据窗口通过 JTAG 口观察到来自实验板上 FPGA 内部的实时信号(如图 3-55 所示，用鼠标的左键或右键放大或缩小波形)。数据窗口的上沿坐标是采样深度的二进制位数，全程是 1024 位(前位触发在 12%深度处)。用鼠标右键单击 q 信号，可以利用右键菜单 Bus Display Format 设置数据显示的进制，如选择无符号十进制显示(Unsigned Decimal)。

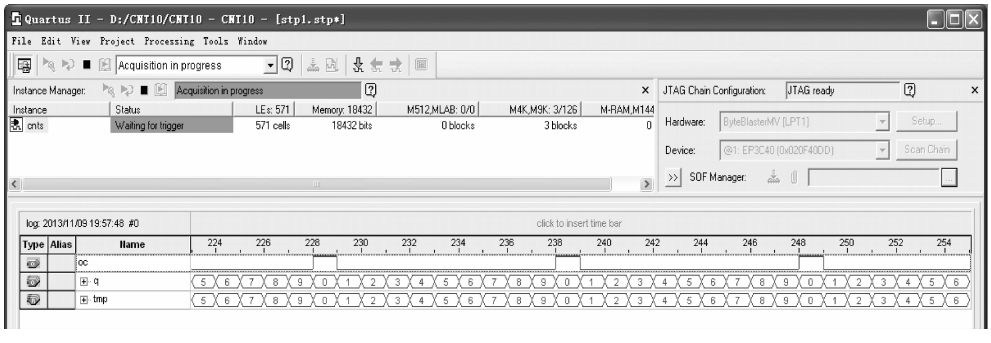


图 3-55 SignalTap Ⅱ采样已被启动

如果单击总线信号 tmp(图 3-55)左侧的“+”号，可以展开此总线信号，同时可单击左右键来控制数据的展开和收缩。此外，如果希望观察到将要形成模拟波形的数字信号波形，可以用鼠标右键单击所要观察的总线信号名(如 cqi)，在弹出的下拉菜单中选择总线显示模式“Bus Display Format”为“Line Chart”，即可获得如图 3-56 所示的“模拟信号波形”。

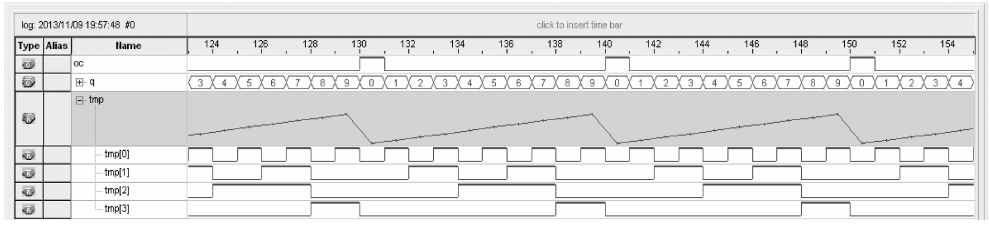


图 3-56 SignalTap Ⅱ 数据窗口设置后的信号波形

注意：在以上给出的示例中，为了便于说明，SignalTap Ⅱ的采样时钟选用了被测电路的工作时钟。但在实际应用中，多数使用独立的采样时钟，这样就能采集到被测系统中的慢速信号，或与工作时钟相关的信号(包括干扰信号)。

为 SignalTap Ⅱ提供独立时钟的方法是在顶层文件的实体中增加一个时钟输入端口，如“Logc_clk：In std_logic”；在此实体中不必对其功能和连接具体定义，而在 SignalTap Ⅱ的参数设置中则可以选择 Logc_clk 为采样时钟。

(7) SignalTap Ⅱ的其他设置和控制方法。以上示例仅设置了单一嵌入式测试模块“cnts”，其采样时钟是 clk。事实上可以设置多个嵌入式测试模块(Instance)。可以使用此功能为器件中的每个时钟域建立单独且唯一的逻辑分析仪测试模块，并在多个测试模块中

应用不同的时钟和不同的设置。

Instance 管理器允许在多个测试模块上建立并执行 SignalTap Ⅱ逻辑分析。可以使用它在 SignalTap Ⅱ文件中建立、删除和重命名测试模块。Instance 管理器显示当前 SignalTap Ⅱ文件中的所有测试模块、每个相关测试模块的当前状态以及相关实例中使用的逻辑元素和存储器耗用量。测试模块管理器可以协助检查每个逻辑分析仪在器件上要求的资源使用量。可以选择多个逻辑分析仪及选择 Run Analysis (Processing 菜单)来同时启动多个逻辑分析仪。此外,SignalTap Ⅱ的采样触发器采用逻辑级别或逻辑边缘方面的逻辑事件模式,支持多级触发、多个触发位置、多个段以及外部触发事件。

可以使用 SignalTap Ⅱ逻辑分析仪窗口中的 Signal Configuration 面板设置触发器选项。可以给逻辑分析仪配置最多 10 个触发器级别,使用户可以只查看最重要的数据。可以指定 4 个单独的触发位置:前、中、后和连续。触发位置允许指定在选定测试模块中、在触发器之前和触发器之后应采集的数据量。分段的模式允许通过将存储器分为密集的时间段,为定期事件捕获数据,而无须分配大采样深度,从而节省硬件资源。

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

3.1.3 利用原理图设计 2-4 译码器

1. 实验目的

(1) 进一步熟悉 Quartus Ⅱ开发环境和设计开发流程。

(2) 掌握利用框图设计输入。

(3) 熟悉 FPGA 实验台的功能和使用方法。

2. 实验内容

利用 Quartus Ⅱ元器件库中的基本单元,设计一个 2-4 译码器,并生成用户自定义框图符号。最后在实验台上进行硬件测试,验证 2-4 译码器的功能。

3. 实验环境

PC	1 台
Quartus Ⅱ软件环境	1 套
GW48-CP++实验台	1 台

4. 实验原理与电路

利用 EDA 工具进行原理图输入设计的优点是,设计者不必具备许多诸如编程技术、硬件描述语言等知识就能迅速入门,完成较大规模的电路系统设计。Quartus Ⅱ提供了直观便捷和操作灵活的原理图输入设计功能,同时还配备了更丰富的适用于各种需要的元件库,其中包含基本逻辑元件库(如与非门、反向器、D 触发器等)、宏功能元件(包含了几乎所有 74 系列的器件),以及类似于 IP 核的参数可设置的宏功能块 LPM 库。Quartus

Ⅱ同样提供了原理图输入多层次设计功能，使得用户能设计更大规模的电路系统。与传统的数字电路实验相比，Quartus Ⅱ提供原理图输入设计功能具有不可比拟的优势和先进性。

(1) 能进行任意层次的数字系统设计。传统的数字电路实验只能完成单一层次的设计。

(2) 对系统中的任一层次，或任一元件的功能能进行精确的时序仿真，精度达 0.1ns，因此能容易地发现对系统可能产生不良影响的竞争冒险现象。

(3) 通过时序仿真，能迅速定位电路系统的错误所在，并随时纠正。

(4) 能对设计方案进行随时更改，并储存设计过程中所有的电路和测试文件入档。

(5) 通过编译和下载，能在 FPGA 或 CPLD 上对设计项目随时进行硬件测试验证。

(6) 如果使用 FPGA 和配置编程方式，将不会有器件损坏和损耗的问题。

(7) 符合现代电子设计技术规范。传统的数字电路实验利用手工连线的方法完成元件连接，容易对学习者产生误导，以为只要将元件间的引脚用引线按电路图连上即可，而不必顾及引线的长短、粗细、弯曲方式、可能产生的分布电感和电容效应，以及电磁兼容性等十分重要的问题。

在 Quartus Ⅱ中，框图编辑器用于以原理图和流程图的形式输入和编辑图形设计信息。框图编辑器主要功能包括：对 Altera 提供的宏功能模块进行例化，插入框图和基本单元符号，从框图或框图设计文件中建立文件。新建一个框图设计文件(Block Diagram/Schemetic File)，即可打开框图编辑器，利用元器件库完成电路的原理图设计。

Quartus Ⅱ框图编辑器读取并编辑框图设计文件。框图编辑器中的每个框图设计文件包含代表设计中逻辑的框图和符号。利用框图编辑器，可将每一个流程图、原理图或者符号代表的设计逻辑合并到工程中。可以在修改框图和符号时更新设计文件。利用框图编辑器，可以插入框图和基本单元符号，称为框图的矩形符号代表设计实体以及相应的信号，这在从上到下的设计中很有用。框图之间是用代表相应信号连接关系的管道连接起来的。可以仅将框图作为设计，也可以将其与其他原理图元素相结合作为用户设计。

框图编辑器还提供有助于在框图设计文件中连接框图和基本单元(包括总线和节点连接映射)的一组工具。可以更改框图编辑器的显示选项，例如根据设计者的偏好更改导向线和网格间距、橡皮线、颜色和像素、缩放以及不同的框图和基本单元属性。

Quartus Ⅱ软件提供可在框图编辑器中使用的各种逻辑功能符号，包括基本单元、参数化模块库(LPM)函数和其他宏功能模块。设计者可以使用这些逻辑功能符号(元器件)来完成框图设计文件，即完成电路逻辑功能设计。在 Quartus Ⅱ中，元器件库以目录树的形式组织，如图 3-57 所示。门电路在元器件树形目录的位置是 Primitives/Logic。元器件库中的门电路名称及符号如表 3-4 所示。其中“Megafunctions”库中包含宏功能模块，“Others”目录项下 Muxplus2 库中包含基本单元和 74 系列器件等，可以在框图设计文件中使用这些元器件符号。利用 Quartus Ⅱ框图编辑器和元器件库可以方便地完成各种复杂逻辑电路设计。

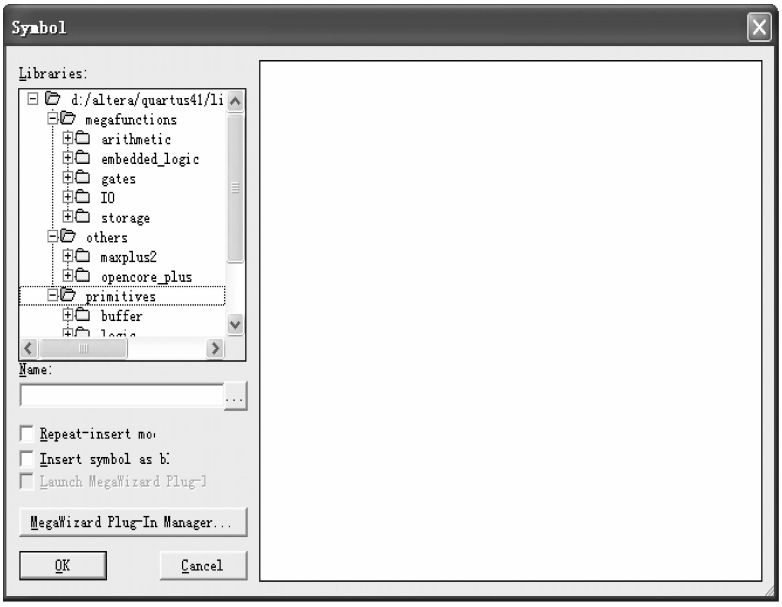
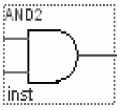
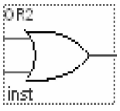
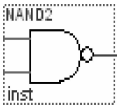
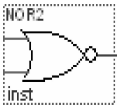
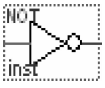
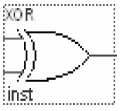


图 3-57　元器件库目录组织

表 3-4　元器件库中的门电路名称及符号

门　电　路	元器件名称	Altera 元器件符号
与门	AND2	
或门	OR2	
与非门	NAND2	
或非门	NOR2	
非门	NOT	
异或门	XOR	

2-4 译码器的元件符号如图 3-58 所示，真值表如表 3-5 所示。

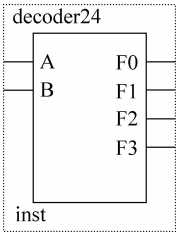


图 3-58　2-4 译码器元件符号

表 3-5　2-4 译码器的真值表

输　入		输　出			
A	B	F3	F2	F1	F0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

5. 实验任务与步骤

1）建立工程

首先建立一个文件夹，以便将所有设计文件存放于该文件夹中，如 D:\decoder24。注意：文件夹名不要用中文，文件夹名称中不要含有空格。然后运行 Quartus Ⅱ 软件，选择主菜单 File→New Project Wizard，利用向导创建工程。如图 3-59 所示，指定工程存放目录为 D:\decoder24，工程名和实体名可取 decoder24。其他设置选项可以保持默认值，创建工程以后再设置。

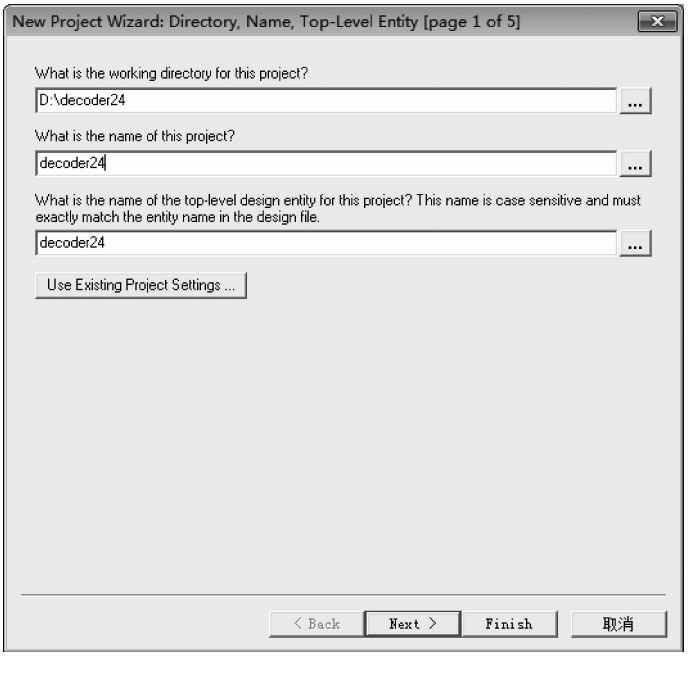


图 3-59　创建工程 decoder24

2）框图设计输入

（1）新建框图文件，打开框图编辑器。具体步骤：主菜单 File→New，弹出 New 对话框，选择 Block Diagram/Schema Tic File 文件类型，如图 3-60 所示。完成后，即打开框图编

辑器，并生成一个空白的框图文件。保存该文件为 decoder24.bdf。

注意：该文件作为工程的设计文件，其名称要和实体名一致，实体名在创建工程时已经指定。

(2) 从元器件库中调用元器件加入到框图文件，完成设计 2-4 译码器电路。具体步骤：选择主菜单 Edit→Insert Symbol 或者单击工具栏上的按钮 或者在编辑窗口中的任何一个位置上右击，将出现快捷菜单，选择其中的输入元件项 Insert Symbol，于是将弹出如图 3-61 所示的“输入元件”对话框。从 Megafunctions\Arithmetic 库中选择 and2，可以在过滤器中输入符号名称，如 and2，所需“与门”的框图符号会自动出现在如图 3-61 所示 Symbol 对话框右侧栏中。然后单击 Symbol 对话框的 OK 按钮，在框图编辑器中，单击鼠标左键即可将元件调入编辑窗口中。可以连续单击鼠标左键加入多个与门，如果不需再添加，可以按键盘上的 Esc 键，或者单击鼠标右键，在弹出的右键菜单中选择 Cancle 选项即可退出元件输入状态，进行其他操作。同理加入其他需要的门电路，基本门电路名称如表 3-4 所示。然后加入输入/输出引脚，输入引脚名称为“Input”，输出引脚为“Output”。输入/输出引脚在元器件树形目录的位置是 primitives/pin。将 2-4 译码器所需的元件符号加入到框图文件后，结果如图 3-62 所示。

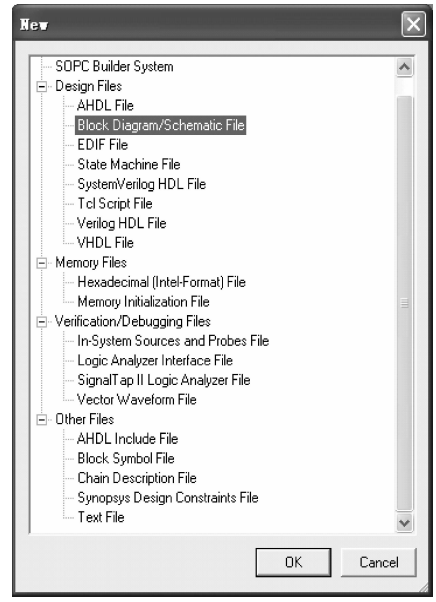


图 3-60　选择编辑文件的语言类型

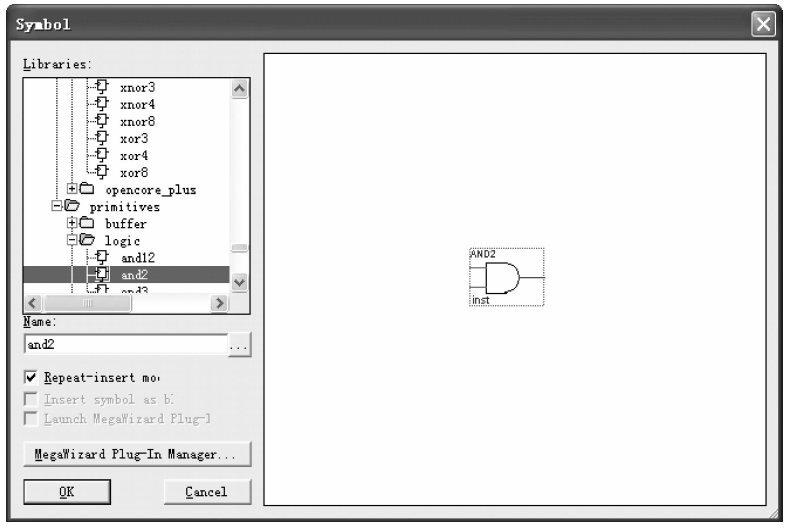


图 3-61　利用过滤器快速搜索框图符号

(3) 双击输入/输出端口 pin_name 文字处，使其变黑色，再用键盘分别输入各引脚名：A、B、F0、F1、F2 和 F3。也可以双击输入/输出端口符号，在弹出的 Pin Properties 窗口中重新命名端口名称。然后利用工具栏上的符号按钮 (Orthogonal Node Tool) 和

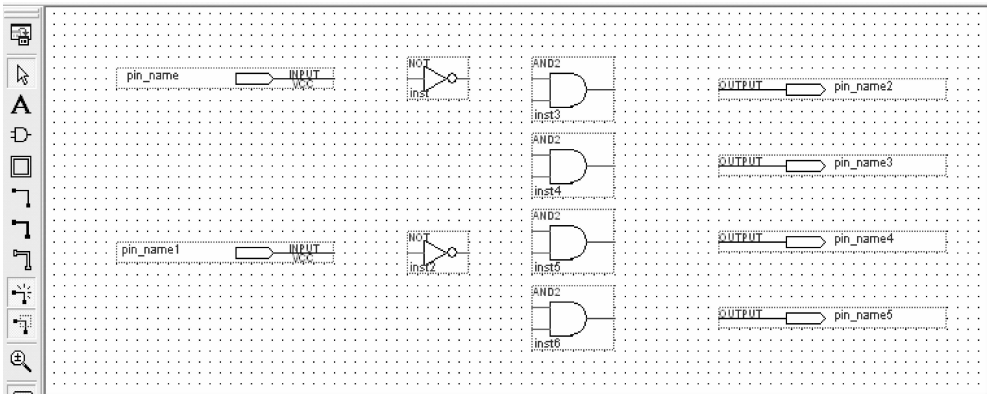


图 3-62 放置元件符号

(Orthogonal Bus Tool)将各个框图符号连接起来,具体方法是单击元件符号的输入/输出信号,用拖动连线的方法连接好各个元件,如图 3-63 所示。

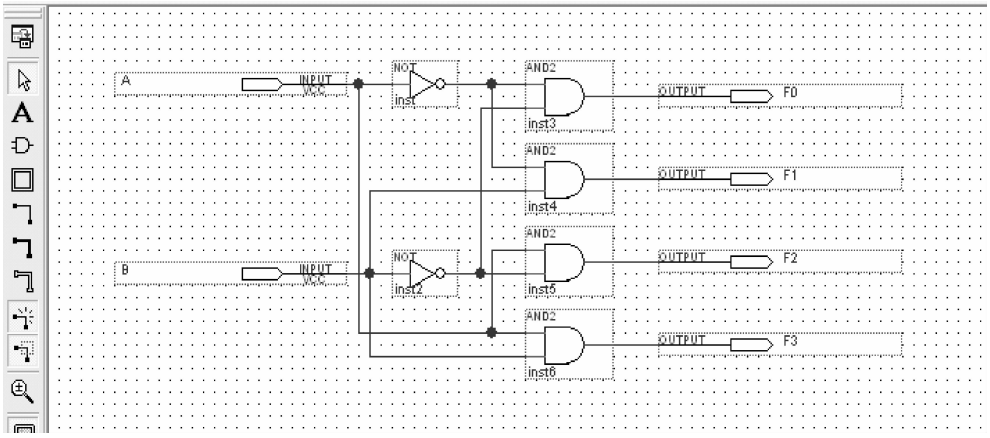


图 3-63 符号的连接与命名

(4) 完成 2-4 译码器逻辑电路设计后,选择主菜单 File→Save,保存框图文件为 D:\decoder24\decoder24.bdf。

3) 器件选择、编译

选择主菜单 Assignments→Device,对器件进行配置,选择 Cyclone Ⅲ 系列 EP3C40Q240C8 型号。然后选择主菜单 Processing→Start Compilation 或者工具栏的按钮 ▶,对整个工程设计进行编译。如果 decoder24.bdf 中的逻辑电路有错误,在下方的 Processing 处理栏中会显示出来。双击错误提示,定位出错的位置,改正错误后再编译工程,直到编译成功。

4) 仿真

(1) 打开波形编辑器。选择主菜单 File→New 项,在 New 对话框中选 Verification/Debugging Files 中的 Vector Waveform File,单击 OK 按钮,即出现空白的波形编辑器。

(2) 在主菜单 Edit→End Time 项,在弹出的对话框中的 Time 栏中输入合适的数字,即

仿真时间，如 100，单位选择“μs”。

(3) 存盘波形文件。选择 File 中的 Save As，将以名为 decoder24.vwf(默认名)的波形文件存入文件夹 D:\decoder24 中。

(4) 输入信号节点。将工程 decoder24 的端口信号节点选入波形编辑器中，主要有以下两种方法。

方法一：首先选择主菜单 View→Utility Windows→Node Finder 选项。在 Node Finder 窗口中单击 List 按钮。于是在左侧 Node Found 栏中出现了设计中的 decoder24 工程的所有端口引脚名(如果此对话框中的 List 不显示，需要重新编译一次，即选择 Processing→Start Compilation，然后重复以上操作过程)。可以用鼠标左键单击 Node Found 栏中的引脚，拖曳到 decoder24.vwf 编辑窗口中的 Name 栏。

方法二：双击波形编辑器左侧栏(Name 栏)的空白处，弹出对话框如图 3-64 所示。在 Filter 框中选择“Pins：All”，然后单击 List 按钮。于是在左侧 Node Found 栏中出现了设计中的 mux21a 工程的所有端口引脚名(如果此对话框中的 List 不显示，需要重新编译一次，即选 Processing→Start Compilation，然后重复以上操作过程)。用鼠标单击窗口中间的按钮 »，可将 Node Found 栏中所有的节点加入到右侧 Selected Nodes 栏中。最后单击 OK 按钮，完成仿真波形节点信号的添加。

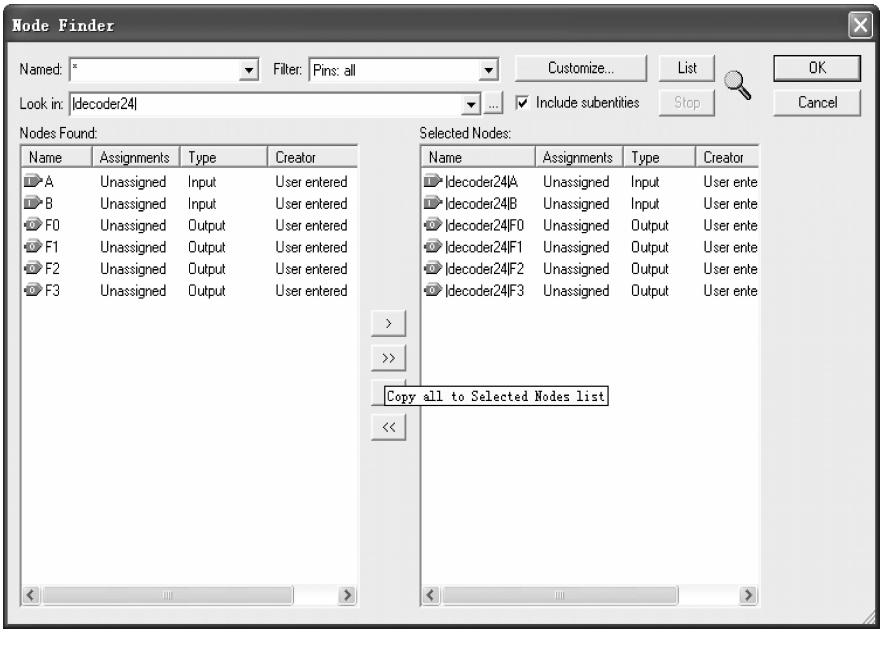


图 3-64 添加输入/输出信号

(5) 编辑输入波形。单击波形窗口左侧的“全屏显示”按钮，使全屏显示，并单击“放大缩小”按钮 后，在波形编辑区域用鼠标左键单击(放大)或右键单击(缩小)，使仿真坐标处于适当位置。这时仿真时间横坐标设定在数十微秒数量级。按照 2-4 译码器的真值表(表 3-5)设计输入波形，编辑输入波形有以下两种方法。

第一种方法：首先单击波形编辑器工具栏上的选择工具(白色箭头)使得鼠标处于选择

状态，然后用鼠标左键单击 A 信号的波形（不是 Name 栏中的 A 信号，而是信号后面的直线），向后拖动一段，即选中这段波形（变成蓝色），再单击波形编辑器工具栏（图 3-20）上的“高电平”按钮，设置这段 A 信号为高电平，如图 3-65 所示。利用这种方法按照图 3-66 完成波形图的编辑。

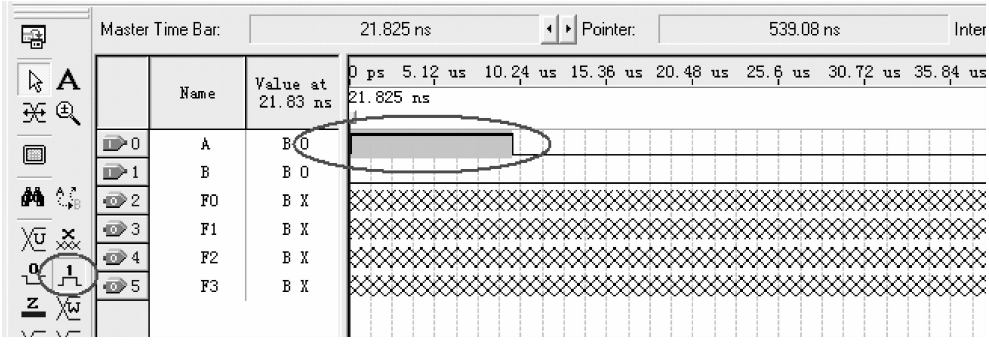


图 3-65 手动设置 A 信号

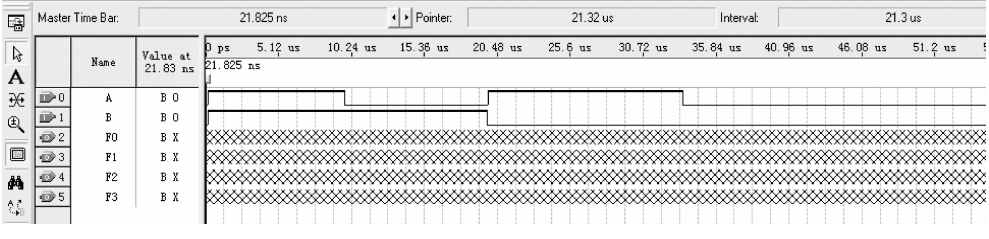


图 3-66 2-4 译码器仿真输入波形图（一）

第二种方法：在波形编辑器的 Name 栏中单击端口信号 A，使之变蓝色，再选择主菜单 Edit→Value→Clock 选项，或者在波形编辑器工具栏上单击“时钟信号”按钮（图 3-20），在弹出的 Clock 窗口中设置时钟周期为 10μs；Clock 窗口中的 Duty Cycle 是占空比，可选择“50”，即 50%占空比；同样方法设置端口信号 B 的周期为 20μs，信号 A 和 B 的信号为两倍关系，这样会出现信号 A 和 B 的 4 种组合情况，如图 3-67 所示。

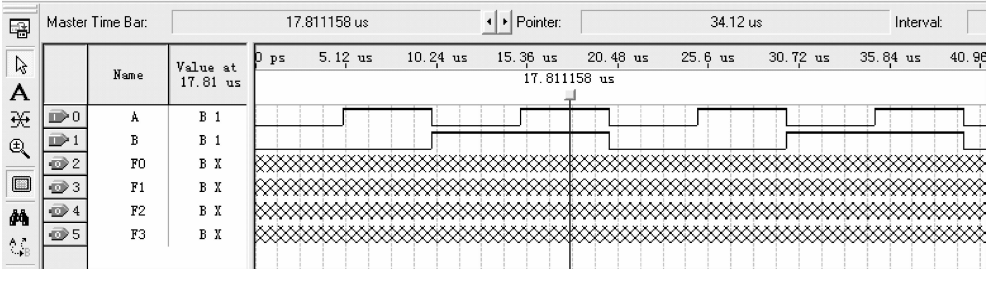


图 3-67 2-4 译码器仿真输入波形图（二）

（6）启动仿真器。所有设置完毕，在选择主菜单 Processing→Start Simulation，或者单击 Quartus Ⅱ 工具栏上的 按钮，开始仿真，直到出现 Simulator Was Successful，仿真结束。

(7) 观察仿真结果。仿真波形文件 Simulation Waveforms 窗口通常会自动弹出，如图 3-68 所示。如果不小心关闭了 Simulation Waveforms 窗口，可以重新启动仿真器，或者选择主菜单 Processing→Simulation Report，打开仿真波形报告文件。注意，Quartus Ⅱ的仿真波形文件中，波形编辑文件(*.vwf)与波形仿真报告文件(Simulation Report)是分开的。如果在启动仿真(Processing→Run Simulation)后，并没有出现仿真完成后的波形图，而是出现文字“Can't Open Simulation Report Window，”但报告仿真成功，则可自己打开仿真波形报告，选择主菜单 Processing→Simulation Report。

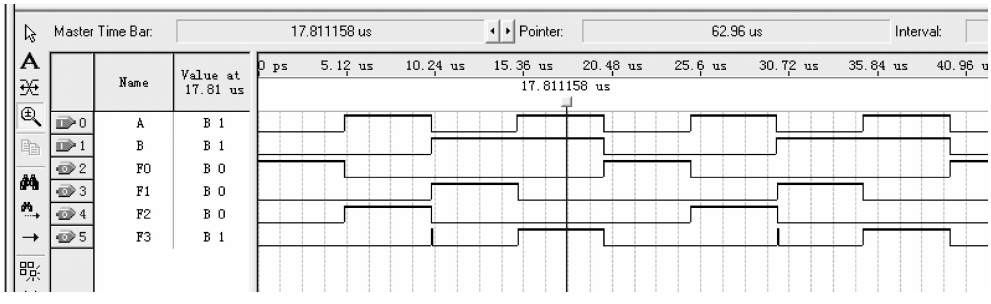


图 3-68　2-4 译码器仿真结果

如果无法展开波形显示时间轴上的所有波形图或者波形图上一片空白，可以单击 Simulation Waveforms 窗口左侧工具栏上的“放大缩小”按钮，用鼠标在波形上左键单击或右键单击，对波形图进行放大和缩小。如果波形无法放大或缩小，可以使用鼠标右键单击波形编辑窗口中任何位置，这时再选择弹出窗口的 Zoom 项，在出现的下拉菜单中选择 Fit in Window。如果即使缩小波形也不能显示电路的全部功能，可能是仿真时间太短，参考前面讲解的设置仿真时间区域，设置合理的仿真时间。

5) 利用 RTL 观察器查看综合后的电路图

Quartus Ⅱ可实现硬件描述语言或网表文件(VHDL、Verilog、BDF、TDF、EDIF、VQM)对应的 RTL 电路图的生成，方法如下。

选择主菜单 File→Open，打开工程所在目录下的 decoder24.bdf 或者在 Project Navigator 窗口的 Files 页面中双击 decoder24.bdf。选择主菜单 Tools→Netlist Viewer→RTL Viewer 项，可以打开 2-4 译码器工程各层次的 RTL 电路图，如图 3-69 所示。双击图形中有关模块，或选择左侧各项，可逐层了解各层次的电路结构。

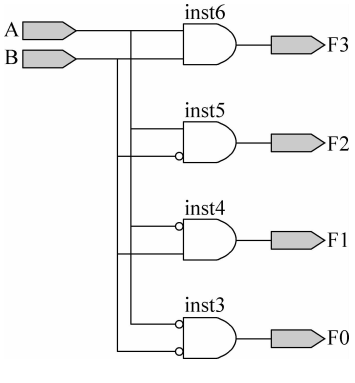


图 3-69　2-4 译码器 RTL 级电路图

6) 引脚锁定与 Sof 文件下载

选择模式 No.5，参照 2.3.4 节的引脚分配表(表 2-14)，设计 2-4 译码器的引脚锁定方案，填写表 3-6。在 Quartus Ⅱ中，选择主菜单 Assignment→Pin 项，在弹出的窗口中，将 2-4 译码器各个信号名称和对应的引脚号分别填到 Node Name 和 Location 中，完成电路的引脚锁定。如果窗口中没有 Location 列可以参考 3.1.1 实验步骤 7)添加 Location 列。引脚锁定完毕后，选择主菜单 Processing→Start Compilation，编译整个工程。

然后用数据线连接 GW48-CP++实验台与计算机，并打开实验台电源。然后选择主菜单 Tools→Programmer，选择编程方式为“ByteBlasterMV[LPT1]”，下载 Sof 文件到 GW48-CP++实验台，具体方法参考 3.3.1 节中实验步骤。

在实验台上，通过按键 1 和 2 输入数据，观察 LED 灯的显示完成 2-4 译码器的功能测试，记录 2-4 译码器输入/输出结果，由实验结果列出真值表，并将其与理论真值表进行比较是否一致。

表 3-6　2-4 译码器引脚锁定方案表

输入/输出信号	外　　设	引 脚 名 称	引　脚　号
A	按键 1		
B	按键 2		
F3	LED 灯 4		
F2	LED 灯 3		
F1	LED 灯 2		
F0	LED 灯 1		

3.1.4　Altera 元器件库宏功能模块

1. 实验目的

(1) 进一步熟悉 Quartus Ⅱ开发环境和设计开发流程。

(2) 掌握利用框图设计输入。

(3) 熟悉 FPGA 实验台的功能和使用方法。

2. 实验内容

利用 Quartus Ⅱ框图编辑器建立设计输入文件，插入 Altera 元器件库中的宏功能模块 lpm_counter(计数器)，并对其进行配置。了解 lpm_counter 的工作原理。

3. 实验环境

PC	1 台
Quartus Ⅱ软件环境	1 套
GW48-CP++实验台	1 台

4. 实验原理与电路

Quartus Ⅱ不仅提供了基本逻辑元件库(如与非门、反向器、D 触发器等)、宏功能元件(包含了几乎所有 74 系列的器件)，还提供了参数可设置的宏功能块 LPM 库。利用框图编辑器，可以对 Altera 提供的宏功能模块进行例化，利用 Megawizard Plug-In Manager(Tools 菜单)可以建立或修改包含宏功能模块自定义变量的设计文件。这些自定义宏功能模块变

量是基于 Altera 提供的包括 LPM 功能在内的宏功能模块。宏功能模块以框图设计文件中的框图表示。利用 Quartus Ⅱ 框图编辑器和宏功能模块可以方便地完成各种复杂逻辑电路设计。

宏功能模块 lpm_counter 是一个计数器电路，其框图如图 3-70 所示。lpm_counter 有以下 4 组信号。

clock——输入时钟脉冲。

data[7..0]——lpm_counter 的 6 位数据端。

q[7..0]——lpm_counter 的 8 位数据输出端。

aclr——异步清零信号，高电平有效。

aload——异步加载数据信号，aload 是计数器控制端，当 aload 为高电平时，计数器处于并行置数状态，即使 q[7..0]输出 data[7..0]的数据。当 aload 为低电平时，计数器处于计数状态，即每次 clock 上升沿到来时，q[7..0]的值自动加 1。

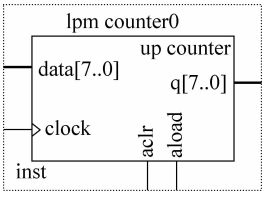


图 3-70 lpm_counter 框图

5. 实验任务与步骤

1）建立工程

首先建立一个文件夹，以便将所有设计文件存放于该文件夹中，如 D:\count。运行 Quartus Ⅱ 软件，选择主菜单 File→New Project Wizard，利用向导创建工程。指定工程存放目录为 D:\count，工程名和实体名可取 count。其他设置选项可以保持默认值，创建工程以后再设置。

2）框图设计输入

（1）新建框图文件，打开框图编辑器。具体步骤：主菜单 File→New，弹出 New 对话框，选择 Block Diagram/Schema Tic File 文件类型。完成后，即打开框图编辑器，并生成一个空白的框图文件。保存该文件为 count. bdf。

注意：该文件作为工程的设计文件，其名称要和实体名一致。

（2）从元件库中插入一个 8 位计数器，具体步骤：选择主菜单 Edit→Insert Symbol 或者单击工具栏上的按钮，从 Megafunctions\Arithmetic 库中选择 lpm_counter，如图 3-71 所示。单击 OK 按钮后，出现 megawizard Plug-In Manager 窗口，接下来设置计数器的参数：8 位（默认值），递增（默认值），连续单击几次 Next 按钮，直到出现输入设置窗口如图 3-72 所示。选择异步清零和置数信号。最后单击 Finish 按钮后生成 lpm_counter 符号，lpm_counter 对应的 VHDL 文件保存路径和文件名为 D: \count\lpm_counter0（默认值）。然后将 lpm_counter 加入到框图编辑器空白的 count. bdf 文件中。

（3）然后为 lpm_counter 添加输入/输出引脚，输入引脚名称为 Input，输出引脚名称为 Output，可在元器件库中查询。可以在过滤器中输入符号名称，如 Input，则所需框图符号会自动出现。可以利用工具栏上的“符号”按钮（Orthogonal Node Tool）和（Orthogonal Bus Tool）将 lpm_counter 的各个框图符号连接起来，完成逻辑电路设计。

注意：总线类信号 Data[7..0]和 Q[7..0]需要用（Orthogonal Bus Tool）与输入/输出引脚连接，即用“粗线”与输入/输出引脚相连。

（4）选择主菜单 File→Save，保存文件 D:\count\count. bdf。

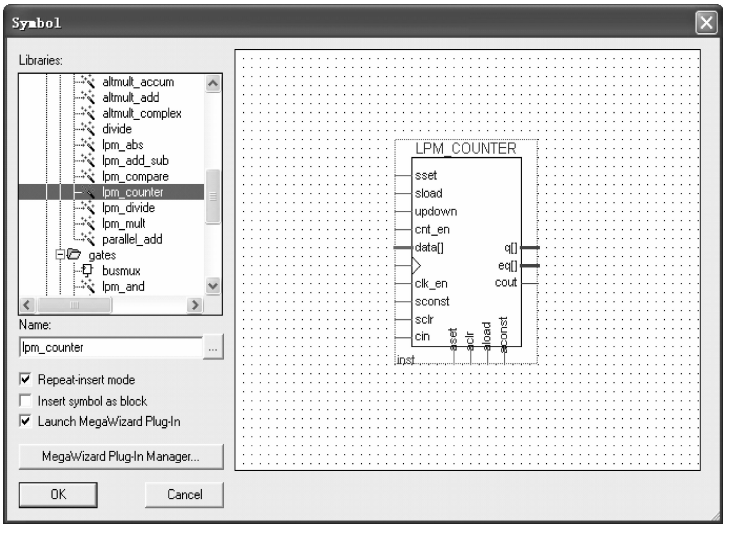


图 3-71 插入 lpm_counter 元件

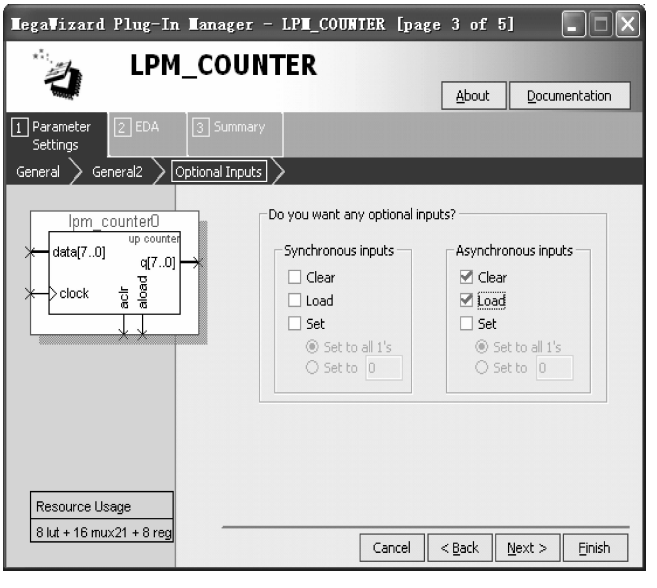


图 3-72 设置 lpm_counter 输入

3）器件选择、编译

选择主菜单 Assignments→Device，对器件进行配置，选择 Cyclone Ⅲ系列 EP3C40Q240C8 型号。选择主菜单 Processing→Start Compilation 或者工具栏按钮 ▶，对整个工程设计进行编译。选择主菜单 File→New 项，在 New 窗口中选 Verification/Debugging Files 中的 Vector Waveform File，建立波形文件。

4）仿真

（1）选择主菜单 View→Utility Windows→Node Finder 选项，将 count 的全部输入/输出信号加入到波形编辑器。具体方法参考 3.3.1 节中的仿真部分。

注意：如果加入了 data[7..0]和 q[7..0]两个信号，可以不用分别加入 data[7]，data[6]，…，data[0]，q[7]，q[6]，…，q[0]，如图 3-73 所示。

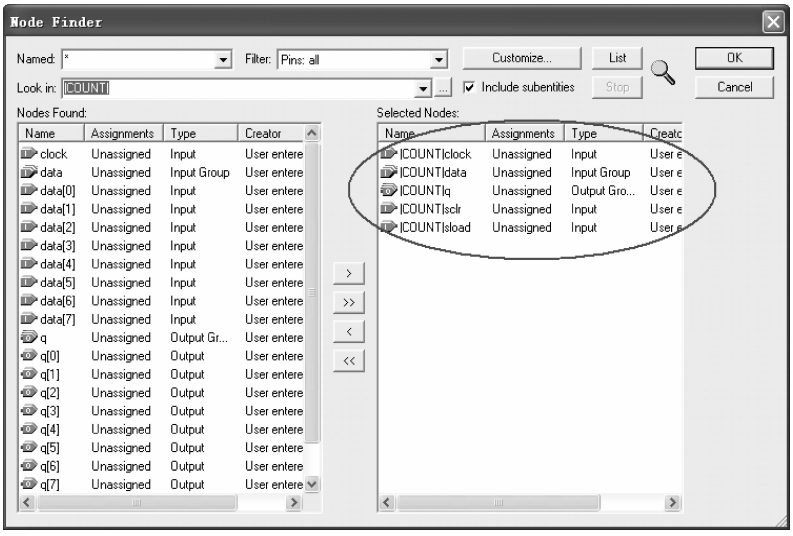


图 3-73　加入输入/输出引脚到波形图

(2) 编辑输入波形。单击波形窗口左侧的“全屏显示”按钮，使全屏显示，并单击“放大缩小”按钮 后，在波形编辑区域用鼠标左键单击(放大)或右键单击(缩小)，使仿真坐标处于适当位置。这时仿真时间横坐标设定在数十微秒数量级。按照 count 的真值表设计输入波形，选择主菜单 Processing→Start Simulation，或者单击 Quartus Ⅱ工具栏上的 按钮，开始仿真，仿真成功后弹出计数器工程 count 的仿真波形如图 3-74 所示。图中只给出了计数功能的仿真，请参考此波形图，仿真计数器的置数功能，即设置 load 和 data 输入端的信号，给计数器加载数据，观察计数器的输出信号 q。

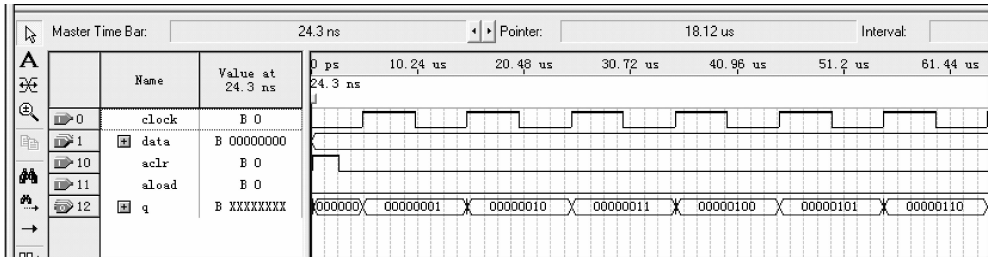


图 3-74　计数器工程 count 的仿真波形

5) 引脚锁定与 Sof 文件下载

选择模式 No.5，参照 3.1.1 节引脚锁定方法，设计计数器工程 count 的引脚锁定方案，填写表 3-7。在 Quartus Ⅱ中，选择主菜单 Assignment→Pin，在弹出的窗口中，将计数器工程 count 各个信号名称和对应的引脚号分别填到 Node Name 和 Location 中，完成电路的引脚锁定。引脚锁定完毕后，选择主菜单 Processing→Start Compilation，编译整个工程。然后选择主菜单 Tools→Programmer，下载 Sof 文件到 GW48-CP++实验台，在实验台上，测

试计数器的功能，记录输出结果，由实验结果列出真值表，并将其与理论真值表进行比较是否一致。

表 3-7 2-4 译码器引脚锁定方案表

输入/输出信号	外　　设	引 脚 名 称	引　脚　号
clk	Clock0		
data[7]	按键 2		
data[6]	按键 2		
data[5]	按键 2		
data[4]	按键 2		
data[3]	按键 1		
data[2]	按键 1		
data[1]	按键 1		
data[0]	按键 1		
aload	按键 3		
aclr	按键 4		
q[7]	LED 灯 7		
q [6]	LED 灯 6		
q [5]	LED 灯 5		
q [4]	LED 灯 4		
q [3]	LED 灯 3		
q [2]	LED 灯 2		
q [1]	LED 灯 1		
q [0]	LED 灯 20		

注意：用数据线连接 GW48-CP++实验台与计算机，并打开实验台电源。并且选择编程方式为 ByteBlasterMV[LPT1]，具体方法参考 3.1.1 节中的选择编程器部分。

3.1.5 元器件自定义与使用

1. 实验目的

(1) 进一步熟悉 Quartus Ⅱ开发环境和设计开发流程。

(2) 掌握利用框图设计输入。

(3) 熟悉 FPGA 实验台的功能和使用方法。

2. 实验内容

(1) 修改 Quartus Ⅱ元器件库中的 74273 寄存器的逻辑功能和外观，生成用户自定义框图符号。

(2) 利用计数器、2-4 译码器和 74273 寄存器设计一个简单的逻辑电路，进行软件仿真，并在实验台上验证实验结果。

3. 实验环境

PC　　1 台

Quartus Ⅱ软件环境　　1套
GW48-CP++实验台　　1台

4. 实验原理与电路

在 Quartus Ⅱ中，元器件库 Others 目录项下 Muxplus2 库中包含基本单元和 74 系列器件等，可以在框图设计文件中使用这些元器件符号。利用 Quartus Ⅱ框图编辑器和元器件库可以方便地完成各种复杂逻辑电路设计。元器件库中的寄存器 74273 是比较常用的寄存器，可以暂存 8 位数据。其框图如图 3-75 所示。本实验的目的是修改 74273 的外观，使其数据输入端和输出端变成总线型端口，然后生成自定义元器件符号，如图 3-76 所示。

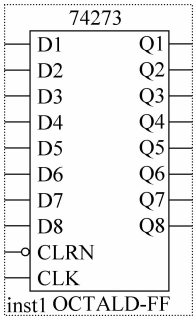


图 3-75　74273 元件符号

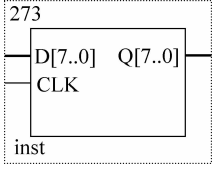


图 3-76　修改后的 74273 元件符号

5. 实验任务与步骤

1）建立工程

首先建立一个文件夹，以便将所有设计文件存放于该文件夹中，如 D:\lab0。运行 Quartus Ⅱ软件，选择主菜单 File→New Project Wizard，利用向导创建工程。指定工程存放目录为 D:\lab0，工程名和实体名可取 lab0。其他设置选项可以保持默认值，创建工程以后再设置。

2）框图设计输入

(1) 新建框图文件，打开框图编辑器。具体步骤：选择主菜单 File→New，弹出 New 对话框，选择 Block Diagram/Schema Tic File 文件类型。完成后，即打开框图编辑器，并生成一个空白的框图文件。保存该文件为 lab0.bdf。

注意：该文件作为工程的设计文件，其名称要和实体名一致。

(2) 修改元件库中现有元件逻辑功能和外观，生成用户自定义框图符号

用户可以在库中元件的基础上修改元件的逻辑功能和外观生成符合需要的用户自定义元件，生成用户定义的框图符号。在 lab0.bdf 文件中插入 Others\Muxplus2 库中的寄存器元件 74273。然后双击 74273 框图符号，打开 74273.bdf 文件(Read Only)，然后可以修改 74273 的逻辑电路，去掉清零输入端，使 74273 具有一个 8 位输入引脚和一个 8 位输出引脚。如图 3-77 所示。通过右键菜单中的 Properties 菜单项可以设置各个框图符号的名称。完成后将 74273.bdf 另存为 D:\lab0\273.bdf。利用 Create/Update 命令生成 273.bsf 文件。将 273 元件符号插入 lab0.bdf 文件中，将原来的 74273 符号删除。按照图 3-78 完成电

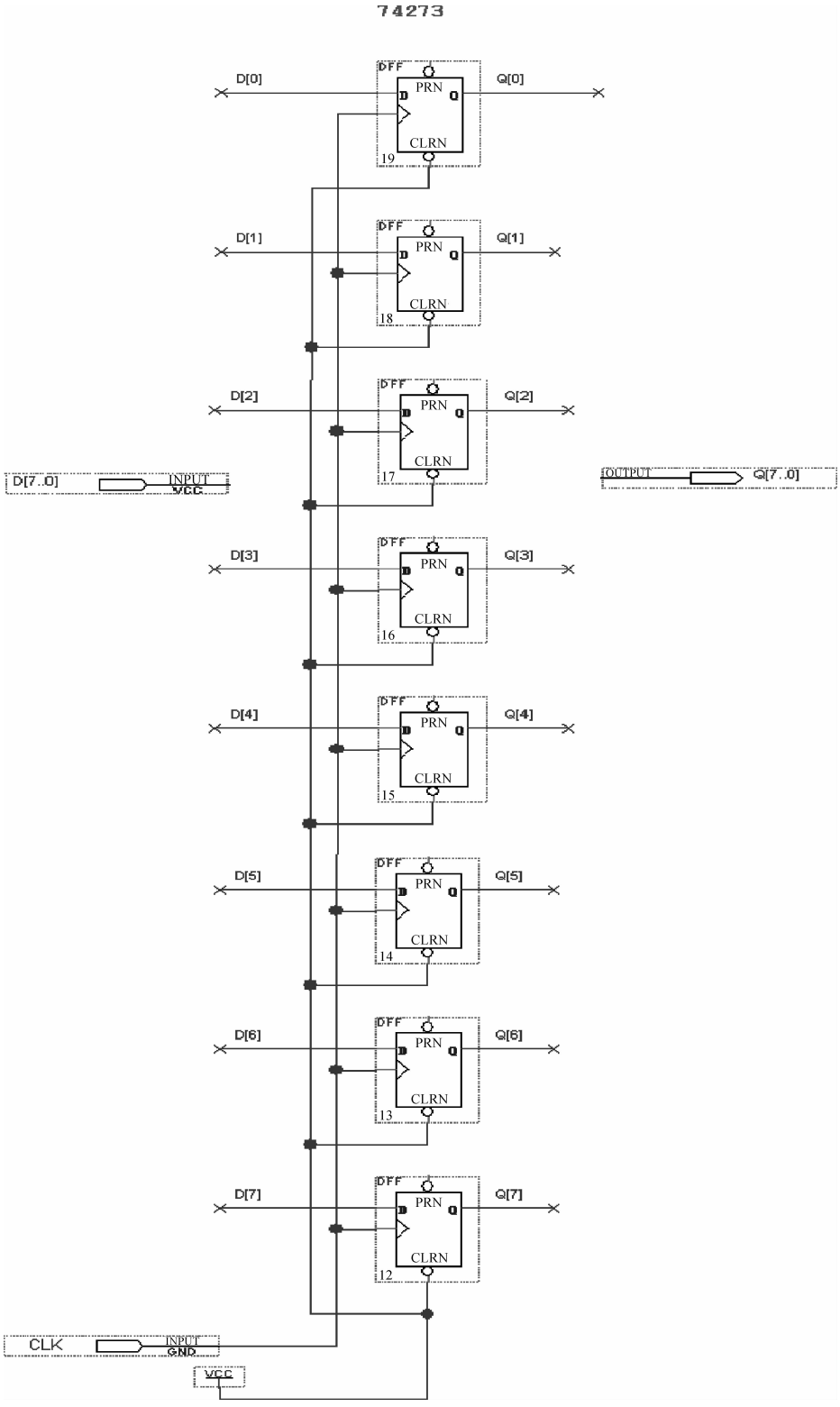


图 3-77 修改后的 273 逻辑电路图

路的设计，保存 lab0.bdf 文件。至此，得到一个简单的逻辑电路。

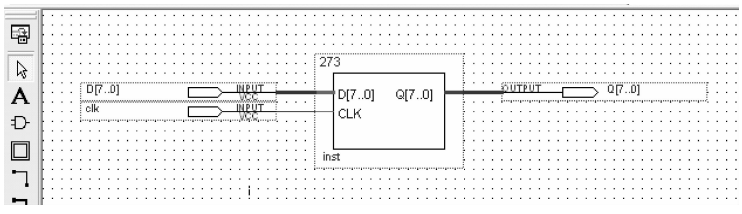


图 3-78　273 实验原理图

3）器件选择、编译与仿真

选择主菜单 Assignments→Device，对器件进行配置，选择 Cyclone Ⅲ 系列 EP3C40Q240C8 型号。选择主菜单 Processing→Start Compilation 或者工具栏按钮 ▶，对整个工程设计进行编译。选择主菜单 File→New，在 New 对话框中选择 Other Files 中的 Vector Waveform File，建立仿真波形文件(.vwf)，选择主菜单 View→Utility Windows→Node Finder 选项。将 lab0 的全部输入/输出信号加入到波形编辑器。选择主菜单 Processing→Start Simulation，或者单击 Quartus Ⅱ工具栏上的 按钮，开始仿真。仿真要求：①设置 D 为 00000011，当 clk 上升沿时，Q 输出为 000000011，寄存器 273 实现了暂存数据的功能；②使 data 不断变化，查看输出 Q 的变化情况，分析寄存器 273 的工作原理。

表 3-8　寄存器 273 引脚锁定方案表

输入/输出信号	外　　设	引脚名称	引　脚　号
clk	Clock0		
D[7]	按键 2		
D[6]	按键 2		
D[5]	按键 2		
D[4]	按键 2		
D[3]	按键 1		
D[2]	按键 1		
D[1]	按键 1		
D[0]	按键 1		
Q[7]	LED 灯 7		
Q[6]	LED 灯 6		
Q[5]	LED 灯 5		
Q[4]	LED 灯 4		
Q[3]	LED 灯 3		
Q[2]	LED 灯 2		
Q[1]	LED 灯 1		
Q[0]	LED 灯 0		

4）引脚锁定与 Sof 文件下载

选择模式 No.0，参照引脚锁定表，设计寄存器 273 的引脚锁定方案，填写表 3-8。在 Quartus Ⅱ中，选择主菜单 Assignment→Pin，在弹出的窗口中，将寄存器 273 各个信号名称和对应的引脚号分别填到 Node Name 和 Location 中，完成电路的引脚锁定。引脚锁定完毕后，选择主菜单 Processing→Start Compilation，编译整个工程。然后选择主菜单 Tools→Programmer，下载 Sof 文件到 GW48-CP++实验台，在实验台上，完成寄存器 273 的功能测试，记录寄存器 273 输入/输出结果，由实验结果列出真值表，并将其与理论真值表进行比较是否一致。

注意：用数据线连接 GW48 实验台与计算机，并打开实验台电源。并且选择编程方式为“ByteBlasterMV[LPT1]”，具体方法参考 2.5.1 中实验步骤 7）中选择编程器部分。在实验台上，通过按键输入数据，观察数码管的显示完成电路功能验证。具体验证过程：利用按键 1 和 2 设置 D[7..0]，按动按键 3 给寄存器 273 一个脉冲，查看数码管的显示，是否与 D 相同。改变 D，查看数码管的显示，测试 273 的功能。

3.1.6 利用原理图层次化方法设计 1 位全加器

1. 实验目的

（1）进一步熟悉 Quartus Ⅱ开发环境和设计开发流程。
（2）掌握和熟悉 Quartus Ⅱ软件原理图设计数字电路的方法。
（3）掌握 Quartus Ⅱ软件的层次化设计方法及步骤。
（4）熟悉 FPGA 实验台的功能和使用方法。

2. 实验内容

利用原理图输入法及层次化方法设计一位全加器，完成软件仿真，并在 FPGA 实验台上验证实验结果。

3. 实验环境

PC	1 台
Quartus Ⅱ软件环境	1 套
GW48-CP++实验台	1 台

4. 实验原理与电路

在 Quartus Ⅱ中，可以采用层次化设计方法设计数字电路。顶层实体可以包含多个底层的电路模块，层次化设计的示意图如图 3-79 所示。层次化设计的核心思想是“模块化”和“元件复用”。模块化是将一个数字系统划分为几个模块，每个模块可由更小的模块实现。每个实体都可以看成是上层

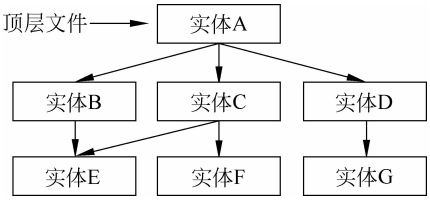


图 3-79 层次化设计示意图

实体中的一个模块或元件，每个模块可以采用 VHDL 或者框图输入法进行设计。元件复用指的是每个模块或元件可以被不同的实体进行多次调用。元件复用减轻了设计者的工作量，并且使程序更加结构化，增加了程序的可读性。

本实验将学习如何采用层次化方法设计全加器。一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。然后在顶层实体设计文件中利用两个半加器和或门设计一位全加器。半加器的元件符号见 2.1.5 节图 2-6，全加器的元件符号见 2.1.5 节图 2-7。下面的实验将完成如何使用层次化方法利用半加器 h_adder 来设计全加器 f_adder。

5. 实验任务与步骤

下面将给出使用原理图输入的方法进行底层元件设计和层次化设计的主要步骤。每个步骤的详细操作参见 3.1.1 节和 3.1.3 节的实验步骤。

1）建立工程

首先建立一个文件夹，如 D:\adder。工程名和实体名可取 adder。

2）框图设计输入

（1）新建框图文件。选择主菜单 File→New，在弹出的 New 对话框中选择 Device Design Files 页的原理图文件编辑输入项 Block Diagram/Schematic File，单击 OK 按钮后将打开原理图编辑窗口。

（2）在编辑窗口中的任何一个位置上右击，将出现快捷菜单，选择其中的输入元件项 Insert Symbol，弹出“元器件”对话框，从 Libraries\Primitives\Logic 和 Libraries\Primitives\Pin 下选择基本门电路和输入/输出引脚。完成设计半加器电路，如图 3-80 所示。

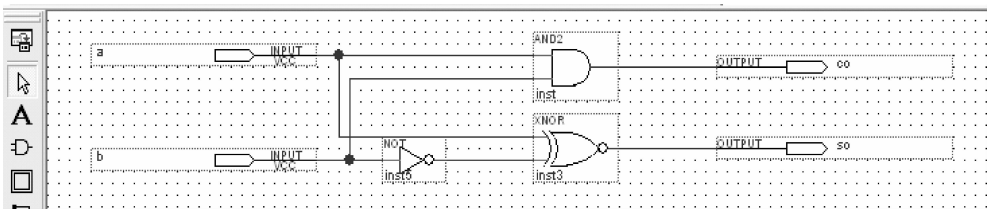


图 3-80　半加器原理图

（3）选择主菜单 File→Save As，选择已建立的工程目录 D:\adder，将已设计好的原理图文件取名为 h_adder.bdf（注意默认的后缀是.bdf，文件名为 h_adder），并存盘在此文件夹内。

3）将设计项目设置成可调用的元件

为了构成全加器的顶层设计，必须将以上设计的半加器 h_adder.bdf 设置成可调用的元件。在打开半加器原理图文件 h_adder.bdf 的情况下，选择主菜单 File 中的 Create/Update→Create Symbol Files For Current File 项，即可将当前文件 h_adder.bdf 变成一个元件符号存盘，以待在高层次设计中调用。

4）设计全加器顶层文件

为了建立全加器的顶层文件，必须再建立一个原理图编辑窗口，方法同前，即再次选择

主菜单 File→New 原理图文件编辑输入项 Block Diagram/Schematic File。

在新打开的原理图编辑窗口双击，在弹出的窗口中选择 h_adder. bdf 元件所在的路径 D:\adder，调出元件，连接好全加器电路图(图 3-81)。

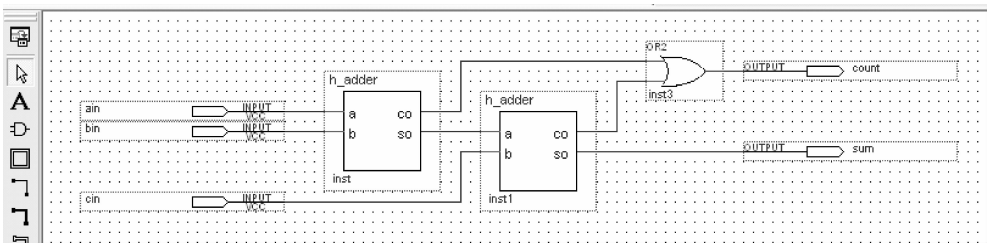


图 3-81 全加器原理图

以 f_adder. bdf 名将此全加器设计存在 D:\adder 的文件夹中。该文件即为工程 f_adder 的顶层实体文件。

5) 编译、仿真

选择主菜单 Assignments→Device，对器件进行配置，选择 Cyclone Ⅲ 系列 EP3C40Q240C8 型号。选择主菜单 Processing→Start Compilation 或者单击工具栏按钮 ▶，对整个工程设计进行编译。选择主菜单 File→New 项，在 New 对话框中选择 Other Files 中的 Vector Waveform File，建立仿真波形文件(. vwf)，选择主菜单 View→Utility Windows→Node Finder 选项。将 adder 的全部输入/输出信号加入到波形编辑器。选择主菜单 Processing→Start Simulation，或者单击 Quartus Ⅱ 工具栏上的 按钮，开始仿真，仿真成功后弹出全加器工程 f_adder 的仿真波形如图 3-82 所示。

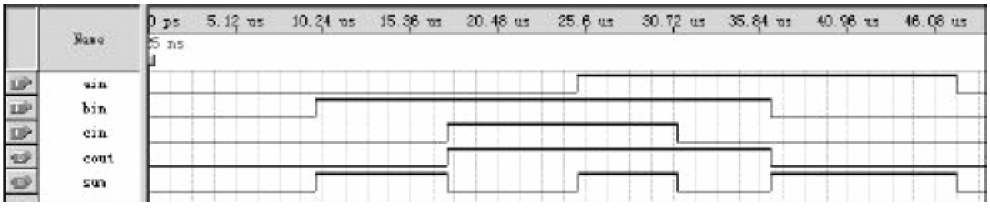


图 3-82 全加器工程 h_adder 的仿真波形

6) 引脚锁定与 Sof 文件下载

选择模式 No. 5，参照引脚锁定表，设计全加器的引脚锁定方案，填写表 3-9。在 Quartus Ⅱ 中，选择主菜单 Assignment→Pin，在弹出的窗口中，将全加器各个信号名称和对应的引脚号分别填到 Node Name 和 Location 中，完成电路的引脚锁定。引脚锁定完毕后，选择主菜单 Processing→Start Compilation，编译整个工程。然后选择主菜单 Tools→Programmer，下载 Sof 文件到 GW48-CP++实验台，在实验台上，通过按键 1、按键 2 和按键 3 的输入数据，观察 LED 灯的显示，完成全加器的功能测试，记录输入/输出结果，由实验结果列出真值表，并将其与理论真值表进行比较是否一致。

注意：用数据线连接 GW48 实验台与计算机，并打开实验台电源。并且选择编程方式为“ByteBlasterMV[LPT1]”，具体方法参考 3.3.1 节中的选择编程器部分。

表 3-9 全加器引脚锁定方案

输入/输出信号	外　　设	引脚名称	引　脚　号
ain	按键 1	PIO0	
bin	按键 2	PIO1	
cin	按键 3	PIO2	
cout	LED 灯 1	PIO8	
sum	LED 灯 2	PIO9	

3.1.7 Quartus Ⅱ的混合输入及层次化设计

1. 实验目的

(1) 进一步熟悉 Quartus Ⅱ开发环境和设计开发流程。
(2) 掌握和熟悉 Quartus Ⅱ软件的混合输入法设计数字电路的方法。
(3) 掌握 Quartus Ⅱ软件的层次化设计方法及步骤。
(4) 掌握和熟悉 Quartus Ⅱ软件的设计输入、编译、仿真以及下载。
(5) 熟悉 FPGA 实验台的功能和使用方法。

2. 实验内容

利用 2 选 1 选择器(见 3.1.1 节)、2-4 译码器(见 3.1.3 节)、计数器(见 3.1.4 节)和 74273 寄存器(见 3.1.5 节)采用层次化设计方法设计一个简单的逻辑电路,要求利用 2-4 译码器分别控制 4 个 74273 寄存器工作。寄存器的输入有两个来源:可以直接通过输入端口置数,还可以来源于计数器的输出。寄存器的输入源由 2 选 1 选择器控制。完成软件仿真,并在实验台上验证实验结果。

3. 实验环境

PC　　1 台
Quartus Ⅱ软件环境　　1 套
GW48-CP++实验台　　1 台

4. 实验原理与电路

在 Quartus Ⅱ中,数字系统采用层次化设计,可以混合使用原理图和硬件描述语言来设计电路,元器件库 Others 目录项下 Muxplus2 库中包含基本单元和 74 系列器件等,可以在框图设计文件中使用这些元器件符号。利用 Quartus Ⅱ框图编辑器和元器件库可以方便地完成各种复杂逻辑电路设计。

5. 实验任务与步骤

1) 建立工程

首先建立一个文件夹,以便将所有设计文件存放于该文件夹中,如 D:\lab1。运行

Quartus Ⅱ软件，选择主菜单 File→New Project Wizard，利用向导创建工程。指定工程存放目录为 D:\lab1，工程名和实体名可取 lab1。其他设置选项可以保持默认值，创建工程以后再设置。下面通过设计一个简单逻辑电路来掌握框图输入设计方法。

2）框图设计输入

（1）生成用户自定义框图符号。分别将 2 选 1 选择器（见 3.3.1 节）、2-4 译码器（见 3.1.3 节）和 273 寄存器的设计文件 mux21a.vhd、decoder24.bdf、273.bdf 复制到 lab1 文件夹。

打开 mux21a.vhd，选择主菜单 File 中的 Create/Update→Create Symbol Files For Current File 项，即可将当前文件 mux21a.bdf 变成一个元件符号，即 mux21a.bsf 文件，以待在高层次设计中调用。同理，生成 decoder24.bsf 和 273.bsf。

（2）建立顶层实体文件。选择主菜单 File→New，弹出 New 对话框，选择 Block Diagram/Schema Tic File 文件类型。完成后，即打开框图编辑器，并生成一个空白的框图文件，保存该文件为 lab1.bdf。

注意：该文件作为工程的设计文件，其名称要和实体名一致。

打开 lab1.bdf，单击主菜单 Edit→Insert Symbol 或者单击工具栏上的按钮，在元件库中 mux21a、decoder24、273 和计数器 lpm_counter 等元件，按照实验内容要求设计电路，如图 3-83 所示。基本功能：4 个寄存器由 2-4 译码器进行选择，寄存器输入数据由计数器输出确定。

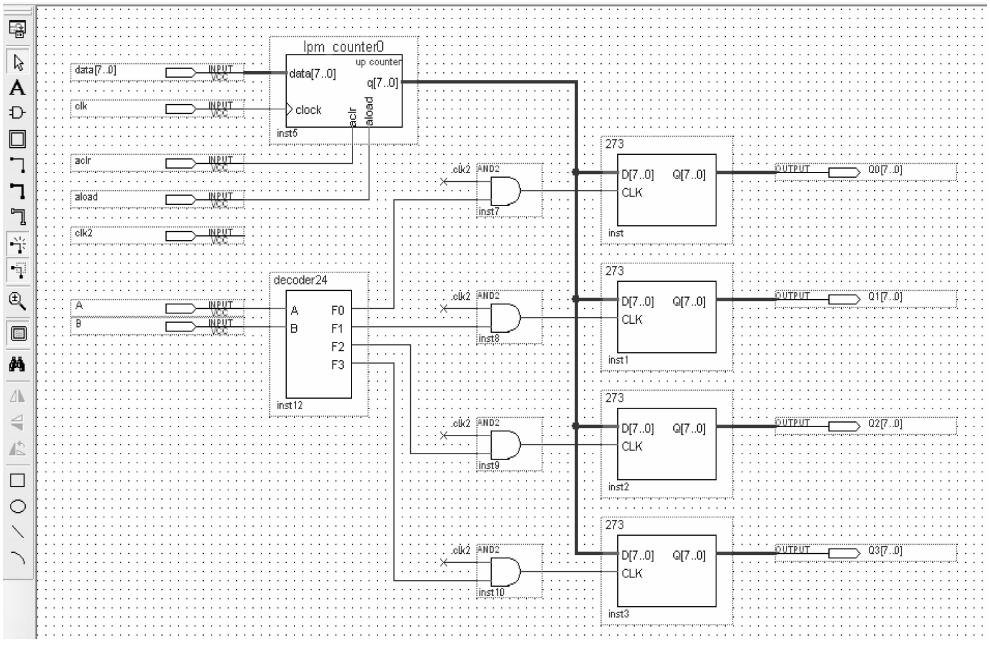


图 3-83　逻辑电路图

3）器件选择、编译与仿真

选择主菜单 Assignments→Device，对器件进行配置，选择 Cyclone Ⅲ系列 EP3C40Q240C8 型号。选择主菜单 Processing→Start Compilation 或者单击工具栏按钮，对整个工程设计进行编译。选择主菜单 File→New（Vector Waveform File），建立仿真波形文件（.vwf），

参考图 3-84，自主设计仿真输入波形，然后对逻辑电路进行仿真，观察仿真波形，记录分析实验结果。参考仿真方案：①给 4 个寄存器置入十六进制数据 11，使得 11 依次从 Q2、Q3、Q1、Q0 端口输出；②依次给 4 个寄存器置入十六进制数据 01、02、03、04，并输出到 Q 端口。

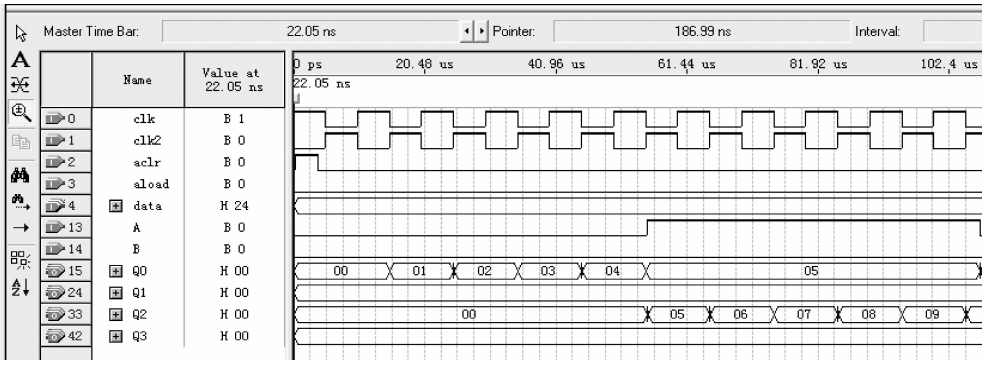


图 3-84　仿真结果

4）引脚锁定与 Sof 文件下载

选择模式 No. 0，按照表 3-10 的要求进行引脚锁定，结合图 2-41 和引脚分配表（表 2-14）来分配引脚，填充表 3-10 的空白内容，并在 Quartus Ⅱ 软件中完成引脚锁定，最后选择主菜单 Tools→Programmer，下载 lab1. sof 文件到 GW48-CP++实验台。下载之前要连接 GW48-CP++实验台与计算机，并打开实验台电源。在实验台上，通过按键输入数据，观察数码管的显示完成电路功能验证。具体验证过程：按键 4、3 取值 00、01、10、11，按键 6 分别取值 0、1，利用计数器向寄存器输入数据，观察寄存器输出情况，是否与理论值相同。

表 3-10　工程 Lab1 引脚锁定方案

输入/输出信号	外　　设	引 脚 名 称	引　脚　号
clk2	按键 8		
clk	按键 7		
aload	按键 6		
aclr	按键 5		
B	按键 4		
A	按键 3		
data[7..4]	按键 2		
data[3..0]	按键 1		
Q3[7..4]	数码管 8		
Q3[3..0]	数码管 7		
Q2[7..4]	数码管 6		
Q2[3..0]	数码管 5		
Q1[7..4]	数码管 4		
Q1[3..0]	数码管 3		
Q0[7..4]	数码管 2		
Q0[3..0]	数码管 1		

3.2 可编程技术实验常见问题与解决方法

3.2.1 Quartus Ⅱ软件使用常见问题与解决方法

(1) 问题描述：Quartus Ⅱ软件工具栏按钮全部为灰色不能对电路进行编译或仿真。

解决方法：打开已经存在的工程，必须选择主菜单 File→Open Project，如图 3-85 所示，而不是打开一个文件(Open File)。如果只打开一个设计文件，是不能编译或仿真的。必须打开工程，然后进行操作。

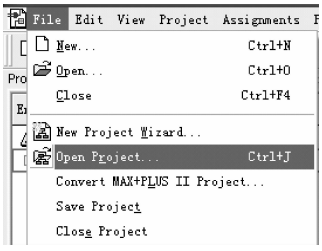


图 3-85 打开已经存在的工程

(2) 问题描述：Quartus Ⅱ软件发生异常，或者 Quartus Ⅱ软件主窗口的一些窗口关闭了。

解决方法：选择主菜单 Tool→Customize，在弹出的窗口中选择 Quartus Ⅱ，单击 Apply 按钮，再重新启动 Quartus Ⅱ软件，可恢复 Quartus Ⅱ软件最初的设置和窗口布局。

(3) 问题描述：Error：Thank You For Using Thequartus Iisoftware. Your 30-Day Evaluation Period Has Expired。

解决方法：Quartus Ⅱ软件 30 天试用过期，需要获取注册文件，对软件进行注册才能继续使用。

3.2.2 “编译”过程常见问题与解决方法

(1) 问题描述：Error：Top-Level Design Entity "Mux21a" Is Undefined。

解决方法：顶层实体 mux21a 没有定义，在创建工程的时候如果指定实体名为“mux21a”，在文本编辑器或者框图编辑器中完成设计输入后，必须以文件名 mux21a. vhd 或者 mux21a. bdf 命名。文件名必须与实体名一致。并且 mux21a. vhd 文件代码中的实体名也必须为 mux21a，即...entity **mux21a** is port (a, b, s: in bit; y : out bit); end entity **mux21a**; architecture one of mux21a is ...

还可以通过以下方法解决该问题：如果实体名为 lab0，顶层实体设计文件为 mux21a. vhd，那么可以在 Projector Navigator 窗口中的“Files”页右击 mux21a. vhd 文件，在弹出菜单中选择 Set as Top-Level Entity 选项，如图 3-86 所示。

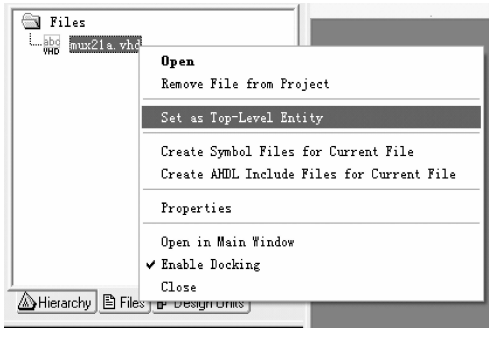


图 3-86 设置顶层实体

(2) 问题描述：编译不成功，但没有显示错误提示窗口。

解决方法：选择主菜单 View→Utility Windows→Messages，打开消息窗口，即可在消息窗口中看到出错信息提示，双击错误提示，定位电路出错的位置，进行修改直至编译成功。

(3) 问题描述：Error：Bus Name Allowed

Only On Bus Line -- Pin "Q[7..0]"。

解决方法：引脚 Q[7..0]是总线类型信号，需要用 Orthogonal Bus Tool 与器件的信号相连，即用“粗线”而不是“细线”连接信号和引脚。

(4) 问题描述：Error：Block Or Symbol Of Type Lpm_Counter0 And Instance "Inst" Overlaps Another Pin，Block，Or Symbol。

解决方法：框图中的 Lpm_Counter0 器件叠加在一起，需要删除多余的 Lpm_Counter0。

3.2.3 “仿真”过程常见问题与解决方法

(1) 问题描述：编译成功后，运行仿真，弹出提示框如图 3-87 所示。

图 3-87 提示设置时钟周期不合法

解决方法：运行仿真之前必须建立波形文件(*.vwf)，加入输入/输出引脚，设置输入信号，然后运行仿真查看输出波形。如果已经建立了波形文件，必须保证其文件名与实体名一致。实体名在创建工程时指定，在 Project Navigator 窗口中可以查看该工程的实体名。如图 3-88 所示，查看工程实体名为 RAM，那么对应的波形图名称为 RAM.vwf。

(2) 问题描述：运行仿真后，输出波形与期望的不一致。

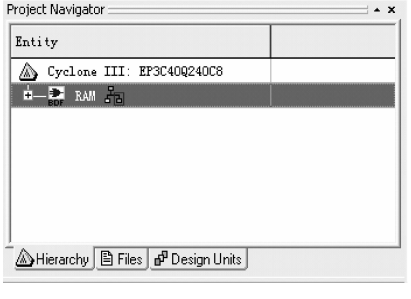


图 3-88 在“工程管理”窗口查看工程实体名

解决方法：如果波形都为直线，可以利用波形编辑器工具栏上的“放大缩小”工具 来对波形进行缩小操作，使波形全部显示。

如果输出的波形中各个输入信号与仿真前编辑的波形图不一致，可能是工程存在多个波形文件(*vwf)，Quartus Ⅱ只仿真文件名与实体名相同的波形文件(最初创建默认保存文件名为实体名)。在工程所在的文件夹删除多余的波形文件，只保留文件名与实体名相同的波形文件，在此文件上编辑输入波形再仿真。

如果无上述两种问题，只能重新编辑波形输入文件，合理设置各个输入信号再运行仿真，直至仿真正确。

(3) 问题描述：无法利用波形编辑器工具栏上的“放大缩小”工具 来对波形进行缩小操作。

解决方法：右键单击波形图，在弹出的菜单中选择 Zoom→Fit In Window 或者按 Ctrl+W 组合键使波形图完整显示在窗口中，然后再利用放大缩小工具调整波形图的显示大小。

(4) 问题描述：给信号设置时钟周期为 10μs 时，弹出提示框，如图 3-89 所示。

解决方法：说明设置的周期不合理，超出了仿真时间，需要加长仿真时间，选择主菜单

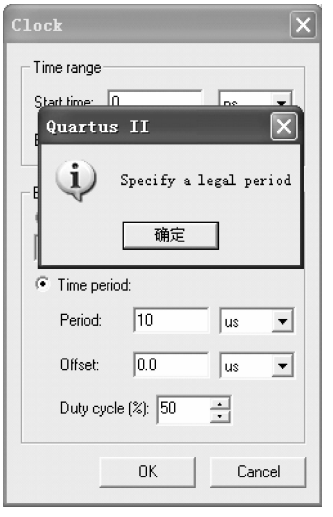


图 3-89 设置时钟周期不合法提示

Edit→End Time，设置合理的仿真时间（大于 10μs），一般需要根据电路的实际情况设置一个合理的仿真时间，一般几百 μs 或 1ms 足够。

（5）问题描述：仿真电路时，弹出提示框，如图 3-90 所示。

解决方法：Quartus Ⅱ只仿真文件名与实体名相同的波形文件。如果波形文件的名字与实体名不一致，Quartus Ⅱ提示没有找到波形文件。所以将现有的波形文件名称改成实体名。

图 3-90 提示波形文件不存在

3.2.4 “编程下载”过程常见问题与解决方法

问题描述：在编程器 Programmar 窗口中单击下载标符 Start 按钮后，Progress 栏一直显示为 0%，并且在 Quartus Ⅱ主窗口下面的 Message 窗口 System 栏显示出错信息：

```
Error: Can't Access Jtag Chain
Error: Operation Failed
```

解决方法：说明下载失败，一般可通过下面的一个或多个方法解决：①选择 ByteBlasterMV[LPT1]编程器；②打开实验台电源；③用数据线连接计算机与实验台，并且保证与实验台连接的数据线端口中间有凸起的一面向上。

3.2.5 “实验台操作”过程常见问题与解决方法

(1) 问题描述：下载时序电路到实验台，时钟信号与锁定引脚为 152，即实验台 Clock0。按动按键设置电路的输入信号，LED 灯没有输出信号。

解决方法：按动实验台左下角的模式选择按键选择正确的实验台模式，确定实验台的模式和引脚锁定时使用的模式相同。确定实验台上的 Clock0 上有短路帽插在选定的频率引脚上。否则无时钟脉冲信号。

(2) 问题描述：电路仿真正确，但下载电路到实验台后，数码管显示数值不正确。

解决方法：数码管引脚锁定不正确。数码管的 4 个 PIO 号的高低位需要与 4 位输出信号的高低位一一对应，顺序不能颠倒。数码管的使用方法见 2.3.5 节。