

# 数 字 逻 辑

## 实 验 指 导

黑龙江大学计算机科学技术学院、软件学院

2024 年 9 月

# 目 录

<b>第一章</b>	<b>KX-DG3B 数字电路实验系统概述</b>	<b>-----1</b>
1.1	KX-DG3B 数字电路实验系统概述	-----1
1.2	KX-DG3B 数字电路实验系统支持的实验项目	-----1
<b>第二章</b>	<b>实验指导</b>	<b>-----2</b>
2.1	KX-DG3B 数字电路实验系统使用介绍及基本 TTL 器件功能测试	--2
2.2	使用 Quartus II 设计数字电路	-----5
2.3	设计三变量多数表决器	-----19
2.4	设计一位全加减器	-----20
2.5	译码器相关设计	-----21
2.6	中期测验	-----22
2.7	触发器特性学习及相关设计	-----22
2.8	设计计数器	-----23

## 实验要求：

1. 实验之前认真预习，写出预习报告，否则实验效率会很低，实验报告格式见实验报告封皮；

2. 实验过程当中，要小心仔细，谨防损坏设备，分析遇到的各种现象，判断结果是否正确，记录运行结果；

3. 实验之后，认真完成并提交实验报告（下课前），包括实验结果、对遇到的各种现象的分析、自己在实验中的心得体会与收获。

## 实验评分标准：

本实验课为非独立设课，满分 20 分，其中：

1. 实验 1、2 为学习实验操作过程，属于验证性实验，不记成绩；
2. 实验 3、4、5、7、8 为 10 分，其中预习情况 5 分、操作能力和实验报告 5 分；
3. 实验 6（中期测验）10 分。
4. 实验 3、4、5、7、8 为 10 分，共 50 分，折合为 10 分，实验 6 单独计 10 分，总计 20 分。

# 第一章 KX-DG3B 数字电路实验系统概述

## 1.1 KX-DG3B 数字电路实验系统概述

KX-DG3B 是由杭州康芯电子有限公司研制的数字电路实验系统，主要用于数字逻辑、数字电路等课程的硬件教学实验，该系统可分为两大互为支持的实验区：

1. 基于传统的手工设计技术之原理验证与原理设计性实验区；
2. 基于现代自动设计技术之自主设计与自主创新实验区。

其中自主设计与自主创新实验区主要由含 20 万逻辑门的 Cyclone II 大规模 FPGA EP2C5 构成。它含有 4608 个 LEs、5064 个触发器、12 万个可编辑 RAM/ROM 存储单元、2 个参数可设置型嵌入式锁相环（可倍频至 400MHz）、26 个 9bit×9bit 可编辑硬件数字乘法器。而在 Quartus II 设计平台中，含有几乎所有类型的 74 系列器件的库，给电路设计带来极大的方便。

## 1.2 KX-DG3B 数字电路实验系统支持的实验项目

实验系统可完成三类实验项目：

1. 利用传统器件（如 74 系列器件）以传统的方式完成数字电路实验。
2. 作为向自主设计平台过渡训练，利用 Quartus II 在 FPGA 上重复完成以上所有传统数字实验，但利用先进的时序仿真器来测试电路模型，包括竞争冒险现象等等。
3. 利用 DDS 函数信号发生器、Quartus II、时序仿真工具、嵌入式逻辑分析仪 Signal Tap II、在系统读写器 In-System Memory Content Editor 等实现自主设计，培养自主设计和自主创新能力。使学生在数字电路学习阶段就能在原理认知、动手能力、实践水平和创新意识方面较之传统实验方式的训练有质的飞跃，为后期的学习打好坚实的理论与实践基础。

除此之外，还能实现步进电机控制、直流电机控制、七段译码器设计、简易波形发生器设计、基于 DDS 的信号发生器、状态机控制 ADC 采样、6 位普通频率计设计、模型数字电子琴设计、乐曲自动演奏电路设计等其它扩展实验项目。

## 第二章 实验指导

### 2.1 KX-DG3B 数字电路实验系统使用介绍

#### 及基本 TTL 器件功能测试

##### 实验目的：

1. 学习如何使用 KX-DG3B 数字电路实验系统；
2. 测试基本 TTL (Transistor-Transistor-Logic) 器件的逻辑功能。

##### 实验内容：

1. 学习 KX-DG3B 的使用方法和注意事项；
2. 测试 74LS00 的逻辑功能；
3. 测试 74LS02 的逻辑功能；
4. 测试 74LS86 的逻辑功能；
5. 分析多路开关的功能。

##### 实验步骤

###### 一. KX-DG3B 的使用方法和注意事项：

1. KX-DG3B 构成复杂，功能多样，使用时需要格外谨慎小心，以免损坏设备；
2. KX-DG3B 上有许多开关、按键、旋钮和跳线，如非必要，不要乱动，以免给实验者自己造成困难。
3. KX-DG3B 与计算机主机通过 JTAG PORT——USB BLASTER——USB PORT 连接，驱动程序已安装好，实验完成时不用拆下，关闭 KX-DG3B 和计算机电源即可；

###### 二. 基本 TTL 器件功能测试

1. 在 KX-DG3B 上找到 74LS00 芯片；
2. 对照 74LS00 的管脚图学习各个管脚功能；
3. KX-DG3B 上的 14 管脚插座的 14 脚、7 脚已连接好+5V 和 GND，所以只需要选取一个逻辑门的输入输出管脚，把输入管脚连接在控制板的数据开关 L1~L12 上，输出管脚接逻辑笔观察结果；
4. 用开关控制不同的输入组合，逻辑笔观察输出结果，填表 1-1；
5. 在 KX-DG3B 上找到 74LS02、74LS86，测试器件功能并填表 1-2、1-3。

**注意：**KX-DG3B 上的 16、20 管脚插座没有接好+5V 和 GND，需要实验者自行连接（注意芯片管脚与管脚插座的对应关系！！）。

###### 二. 分析图 1-4 所示多路开关，写出输出表达式并填表 1-4：

实验结果

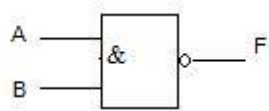


图 1-1

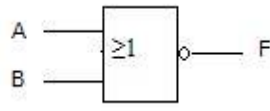


图 1-2

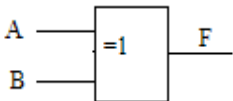


图 1-3

表 1-1

A	B	$F = \overline{AB}$
0	0	
0	1	
1	0	
1	1	
0		
1		

表 1-2

A	B	$F = \overline{A+B}$
0	0	
0	1	
1	0	
1	1	
0		
1		

表 1-3

A	B	$F = A \oplus B$
0	0	
0	1	
1	0	
1	1	
0		
1		

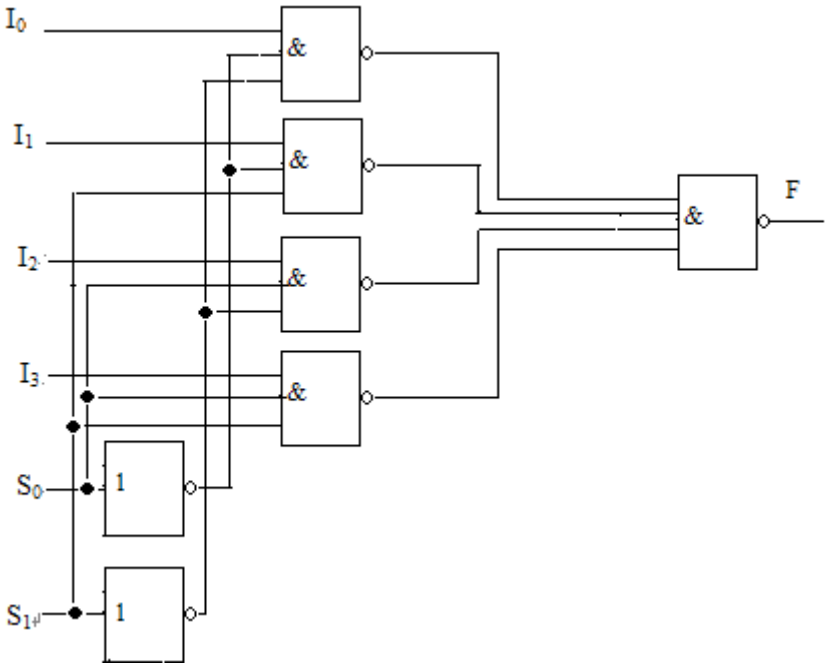





图 1-4 多路开关

输出 F=\_\_\_\_\_

表 1-4 多路开关功能表

S <sub>0</sub>	S <sub>1</sub>	输 入	输 出 F	不起作用端	何路选通
0	0	I <sub>0</sub> 0 1 			
0	1	I <sub>1</sub> 0 1 			
1	0	I <sub>2</sub> 0 1 			
1	1	I <sub>3</sub> 0 1 			

功能总结：

74LS00：

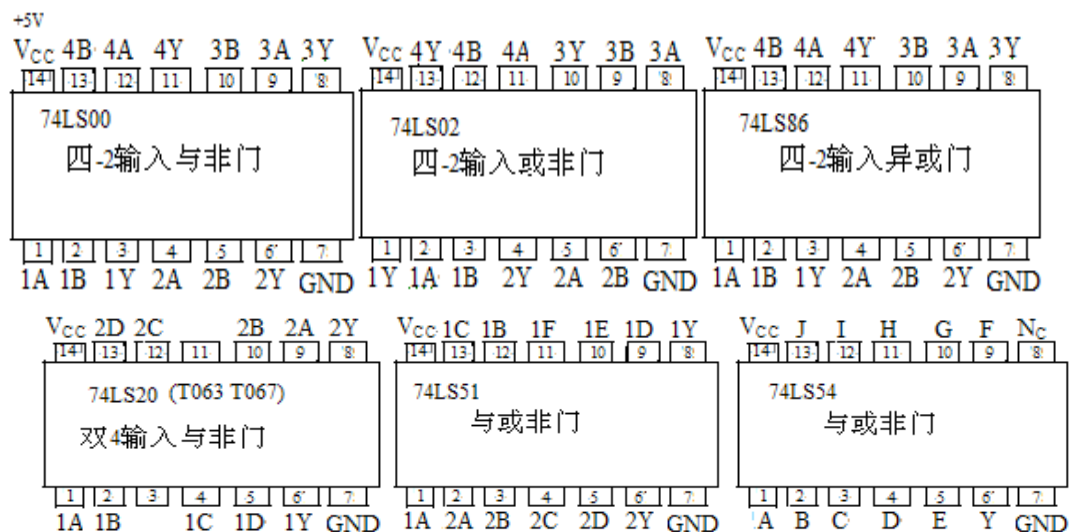
74LS02：

74LS86：

多路开关功能：

管脚图：（其中后三个芯片功能表达式：74LS20： $Y=\overline{ABCD}$ ；

74LS51： $1Y=\overline{ABC+DEF}$ ； $2Y=\overline{AB+CD}$ ；74LS54： $Y=\overline{AB+CDE+FGH+IJ}$ ）



## 2.2 使用 Quartus II 设计数字电路

### 实验目的：

学习使用 Quartus II 设计数字电路；

### 实验内容：

- 1. 学习如何使用 Quartus II；
- 2. 用 Quartus II 设计一个简单电路。

### 实验步骤

#### 一. Quartus II 软件使用

##### 1. Quartus II 简介

Quartus II 是美国 Altera 公司提供的大规模 PLD 开发集成环境。Altera 是世界上最大的可编程逻辑器件供应商之一。Quartus II 可以完成应用于 FPGA/CPLD 的 EDA 开发整个流程，它提供了一种与结构无关的设计环境，使设计者能方便地进行设计输入、快速处理和器件编程。图 2-1 是 EDA 设计的一般流程：

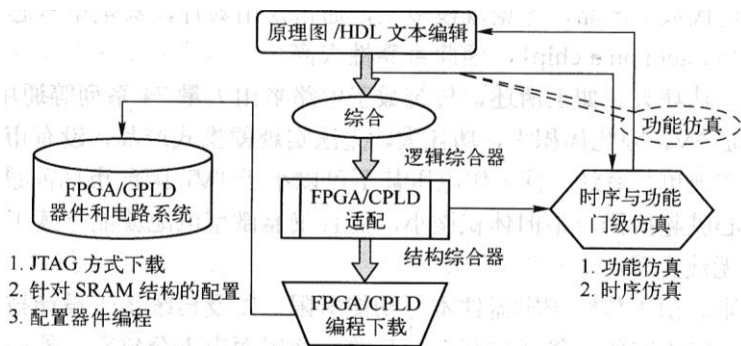


图 2-1 应用于 FPGA/CPLD 的 EDA 开发流程

Quartus II 提供了能满足各种特定设计的需要、也是单芯片可编程系统设计的综合性环境。Quartus II 也完全支持 VHDL、Verilog HDL 的设计流程，其内部嵌有 VHDL、VerilogHDL 综合器。

图 2-2 形象地说明了针对软件语言的编译器及其编译结果和针对硬件语言的综合器及其综合结果之间的区别。

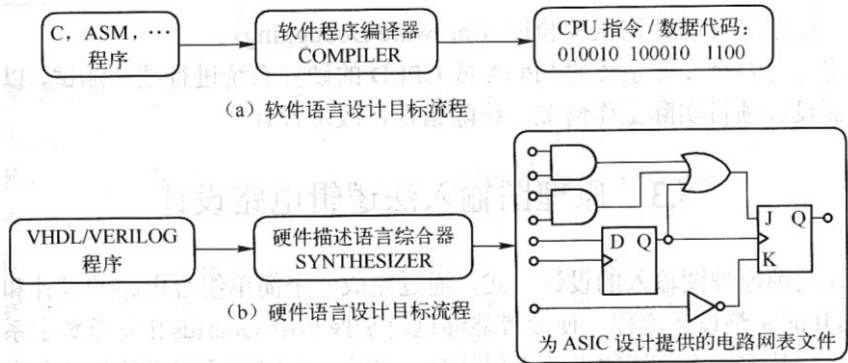


图 2-2 计算机软/硬件描述语言编译/综合工具的不同之处



如图 2-3 所示, Quartus II 包括模块化的编译器, 编译器包括的功能模块有分析/综合器、适配器、装配器、时序分析器、设计辅助模块、EDA 网表文件生成器、编辑数据接口等。

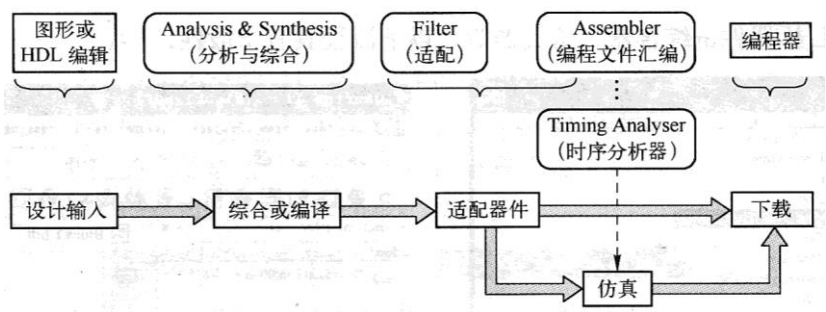


图 2-3 Quartus II 设计流程

图 2-3 中的上半部分是 Quartus II 编译设计主控界面, 它显示了 Quartus II 自动设计的各主要处理环节和设计流程, 包括设计输入编辑、设计分析与综合、适配、编程文件汇编 (装配)、时序参数提取以及编程下载几个步骤。图 2-3 下半部分的流程图, 是与此图上面的 Quartus II 设计流程相对照的标准的数字系统自动开发流程。

2. Quartus II 完整设计流程

本节使用纯原理图输入方式完成图 2-4 所示电路的设计, 详细介绍 Quartus II 的完整设计流程。

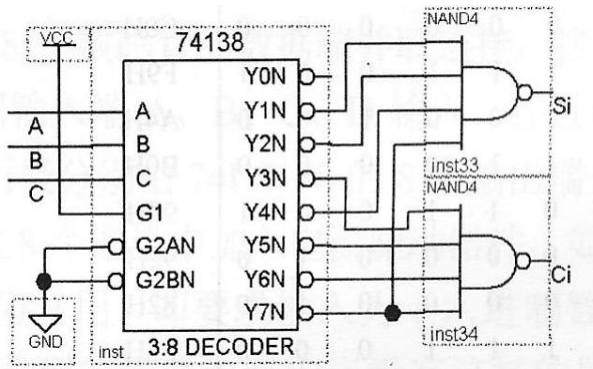


图 2-4 逻辑电路图

Quartus II 的完整设计流程有以下几步: 原理图输入和创建工程、全程编译、仿真、引脚锁定、下载编程。以下做详细介绍:

1) 原理图输入和创建工程

输入电路原理图前要建立工作库文件夹, 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 Quartus II 默认为工作库 (Work Library)。一般同一工程的所有文件都放在同一文件夹中。建立了文件夹后就可以通过 Quartus II 的原理图编辑器编辑电路并存盘了。

a) 新建一个文件夹。D:\TEST

注意: 文件夹最好位于硬盘某个分区的根目录下, 名字不能用中文, 最好不用数字!

b) 打开原理图编辑窗口  
打开 Quartus II，选择菜单  
File→New，选择 Design Files  
中的 Block Diagram/ Schematic  
File (图 2-5)，点 OK，即出现图  
2-6 所示原理图编辑窗口。

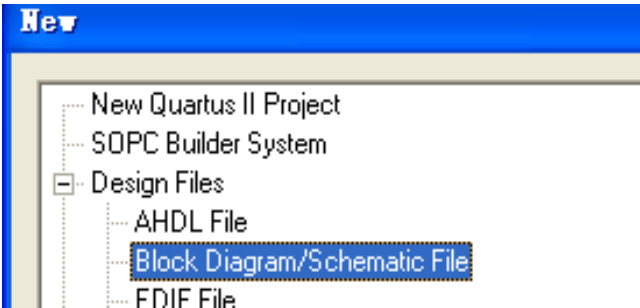


图 2-5 选择编辑文件类型

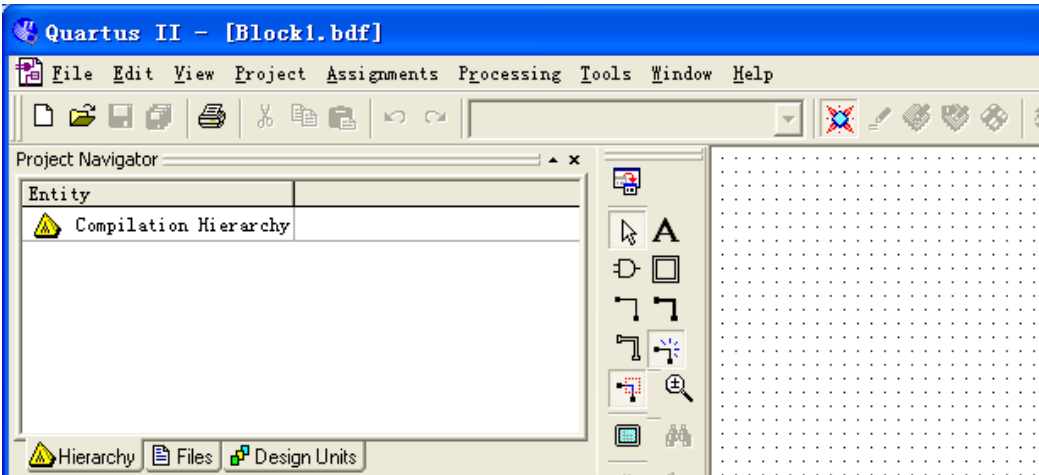


图 2-6 原理图编辑窗口

c) 编辑输入电路原理图  
双击原理图编辑窗口中任一空白点，将激活并弹出逻辑电路器件输入对话框：

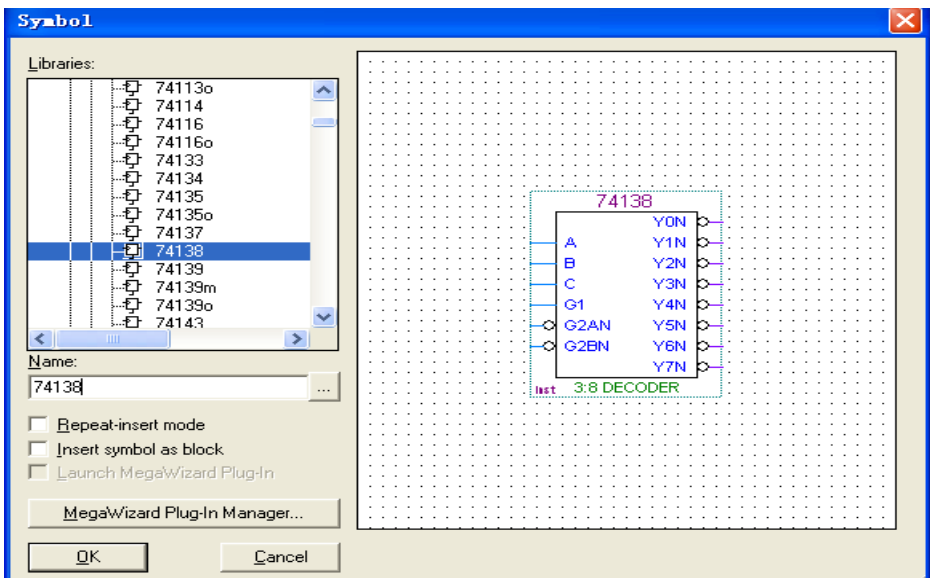


图 2-7 调入需要的宏功能元件 (Symbol) 74138

在 Name 框内输入所需元件名称，在此为 74138，这是一个 3-8 译码器。（由于仅考虑

器件的逻辑功能，诸如 74LS138、74HC138、74S138 等同类功能的器件一律命名为 74138)。单击 OK，即可将此元件调入编辑窗口中。

以同样方法调入两个 4 输入与非门（名称是 NAND4）、数个输入输出端口（INPUT 和 OUTPUT），高电平符号（VCC）、低电平符号（GND）。

其他门级元件的名称：2 输入与门是 AND2, 3 输入或门是 OR3, 反相器是 NOT, 异或门是 XOR 等。

用鼠标拖出连线将元件连接起来，如图 2-8 所示。输入输出端口的名称可以修改，双击端口元件，在弹出的对话框键入 AI、SO 等。鼠标双击元件 74138，会展开此元件的内部电路结构，或元件的功能描述（如果是用 HDL 描述的元件），以了解元件的内部结构。

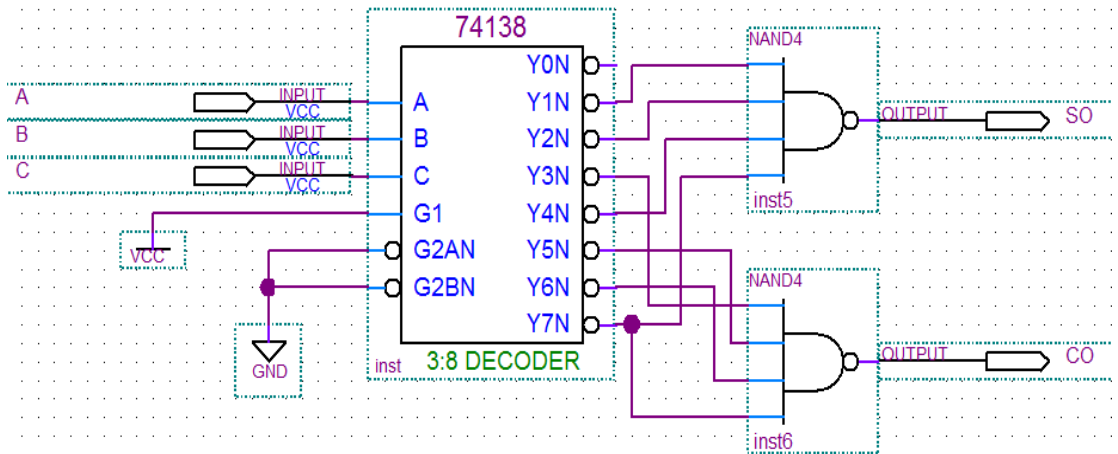
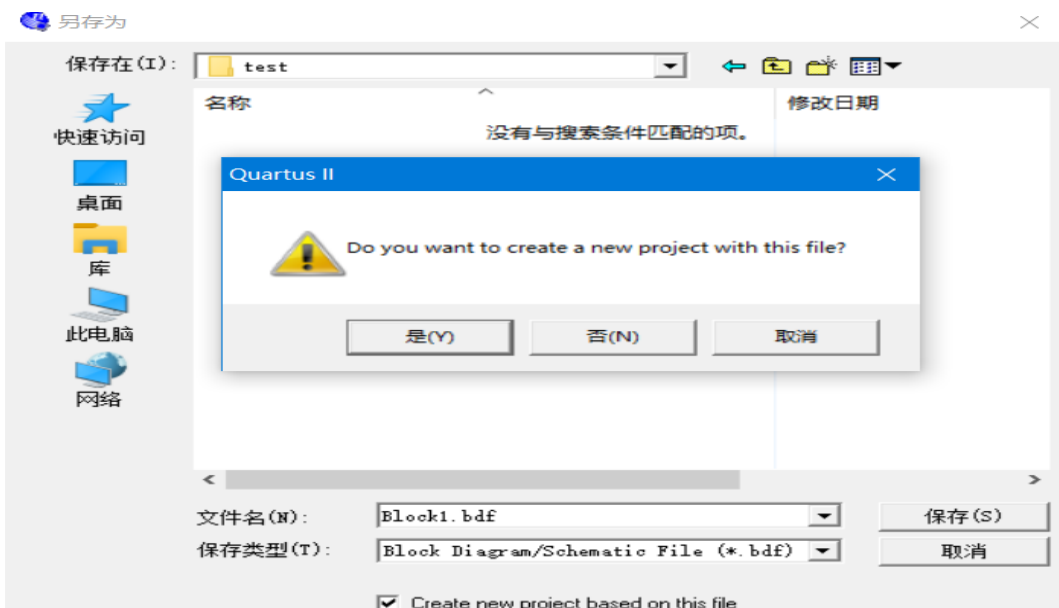


图 2-8 示例电路图

d) 文件存盘。

点存盘按钮，找到已建立的文件 D:\TEST，存盘文件名可以是默认的 Block1.bdf。点“保存”，出现问句“Do you want to create a new project with this file?”时，若单击“是”就直接进入创建工程流程；



如单击“否”，也可以按以下方法进入创建工程流程。

创建工程

进行一个项目设计前必须为此项目建立一个工程管理项目，简称为工程。使用 New Project Wizard 可以创建工程，即可令顶层设计 BLOCK1.bdf 为工程，并设定此工程的一些相关信息，如工程名、目标器件、综合器等，步骤如下：

1、打开建立新工程管理窗。Files→New Project Wizard，弹出工程设置对话框：

单击第一栏右侧的“...”按钮，找到建立的文件夹 D:\TEST，选中已存盘的文件 BLOCK1.bdf（一般应设顶层设计文件为工程）。单击“打开”，即出现如图 2-9 所示的设置情况。其中第一行 D:\TEST 表示工程所在的库文件夹；第二行 BLOCK1 表示此项工程的工程名（工程名可以取任何其他的名，也可以直接用顶层文件名作为工程名）；第三行是当前工程顶层文件的实体名，这是针对 HDL 输入而言的。对于原理图，即为 BLOCK1。如在存盘一步点了“是”，则略过斜体这段，后续前两部直接点 Next 即可。

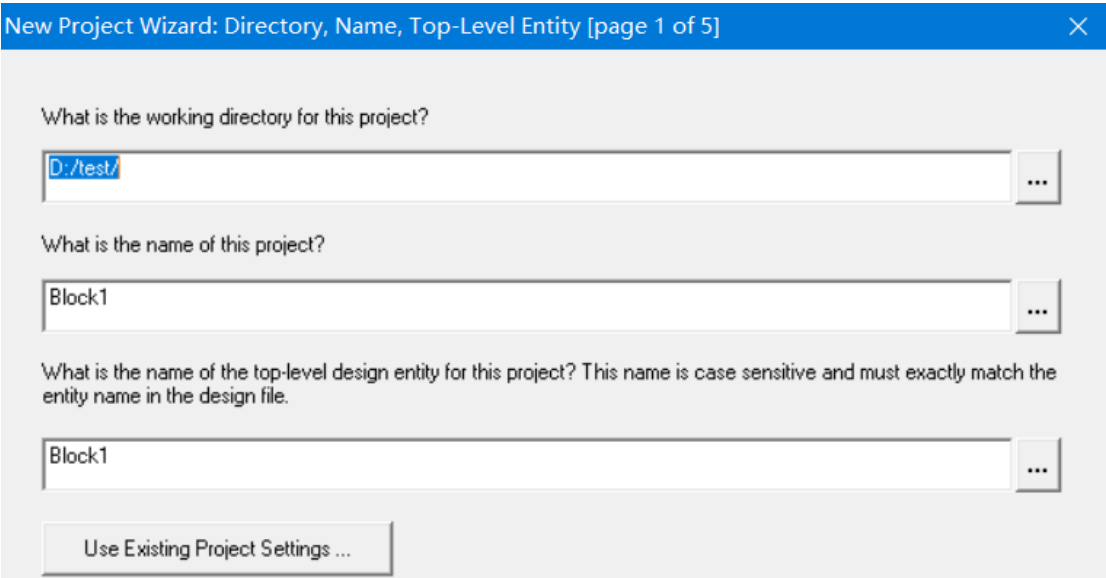


图 2-9 利用 New Project Wizard 创建工程 BLOCK1

2、将设计文件加入工程中。单击下方的 NEXT，在 Add Files 窗口中单击 File name 栏右侧按钮“...”，找到工程文件 BLOCK1.bdf，点击右侧的 Add 将该文件加入工程中：

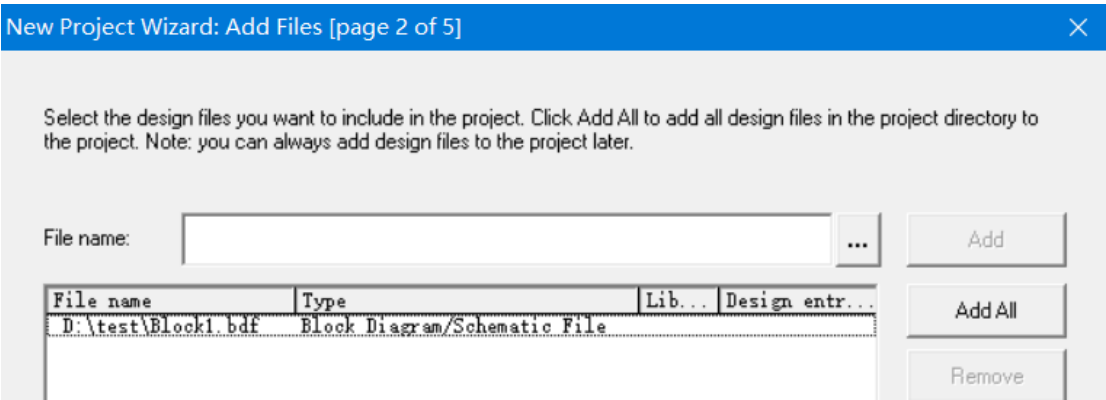


图 2-10 将所有相关文件加入工程

工程文件加入的方法有两种：第一种是单击 Add All 按钮，将设定目录中的所有相关文件加入工程；第二种是单击 Add 按钮，把从目录中选出的相关文件加入工程。

3、选择目标芯片。即选择当前设计的电路最终将实现于什么硬件芯片中。单击 Next 进入目标芯片选择窗口。首先在 Family 栏选择芯片系列为 Cyclone II 系列。这里选择的目标器件是 EP2C5T144C8，其中 EP2C5 表示 Cyclone II 系列及此器件的逻辑规模；T 表示 TQFP 封装；C8 表示速度级别。便捷的方法是通过图 2-11 所示窗口右边的三个选项栏进行过滤选择：选择 Package 为 TQFP；Pin count 为 144 和 Speed grade 为 8 级，选择图中四个器件的第一个 EP2C5T144C8，点 Next。

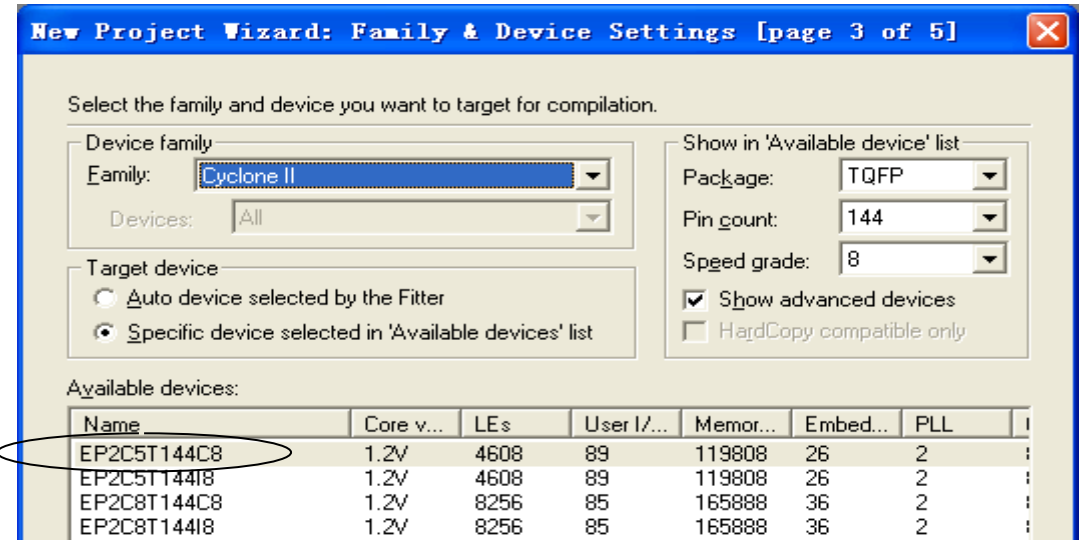


图 2-11 选择目标器件 EP2C5144C8 (注意::: 必须选中图中第一个器件!!!)

EP2C5T144C8 属 Altera 公司的 Cyclone II 系列 FPGA，其逻辑资源规模约达 20 万逻辑门，包含 4608 个逻辑宏单元、5064 个触发器、12 万个可编辑 RAM/ROM 存储单元、2 个参数可设置型嵌入式锁相环(可倍频至 400MHz)、26 个 9bit×9bit 可编辑硬件数字乘法器等。

4、工具设置。点 Next 都做默认选择，表示仅选择 Quartus II 自含的所有设计工具。

5、结束设置。点击 Finish。这一步是“工程设置统计”窗口，列出了与此项工程相关的设置情况。出现 BLOCK1 的工程管理窗，显示本工程项目层次结构（图 2-12）。

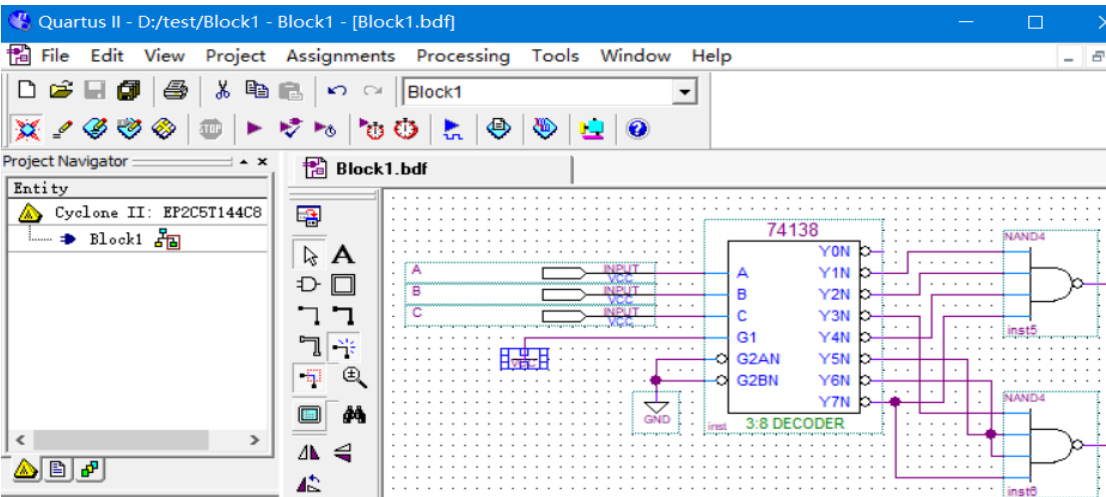


图 2-12 BLOCK1 工程管理窗

注意图 2-12 左上角的路径指示，它指出了当前工程所在位置和工程名，以及顶层工程文件名等。建立工程后，可使用 Assignments→Settings 对话框的 Files 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，将按 Files 页显示的顺序处理文件。

2) 全程编译

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、适配、输出结果的编辑配置，以及时序分析等。在这一过程中，为了把设计项目适配到 FPGA 目标器件中，将同时产生多种用途的输出文件，如功能和时序信息文件、器件编程的目标文件等。编译开始后，编译器首先检查出工程设计文件中可能的错误信息，供设计者排除，然后产生以网表文件表达的电路原理图文件。

点击工具栏上的紫色三角按钮，启动全程编译。（或 Processing→Start Compilation 项，或 ctrl+L）这里所谓的全程编译（Full Compilation）包括以上提到的 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、装配文件生成（仿真文件与编程配置文件），以及基于目标器件硬件性能的工程时序分析等。

如编译出错，查看工程管理窗下方的 Processing 栏中的编译信息。红色文字显示出错信息，蓝色警告可忽略。可单击此条文，弹出对应的顶层文件并用深色标记指出错误所在。改正错误，再次编译，直至排除所有错误。

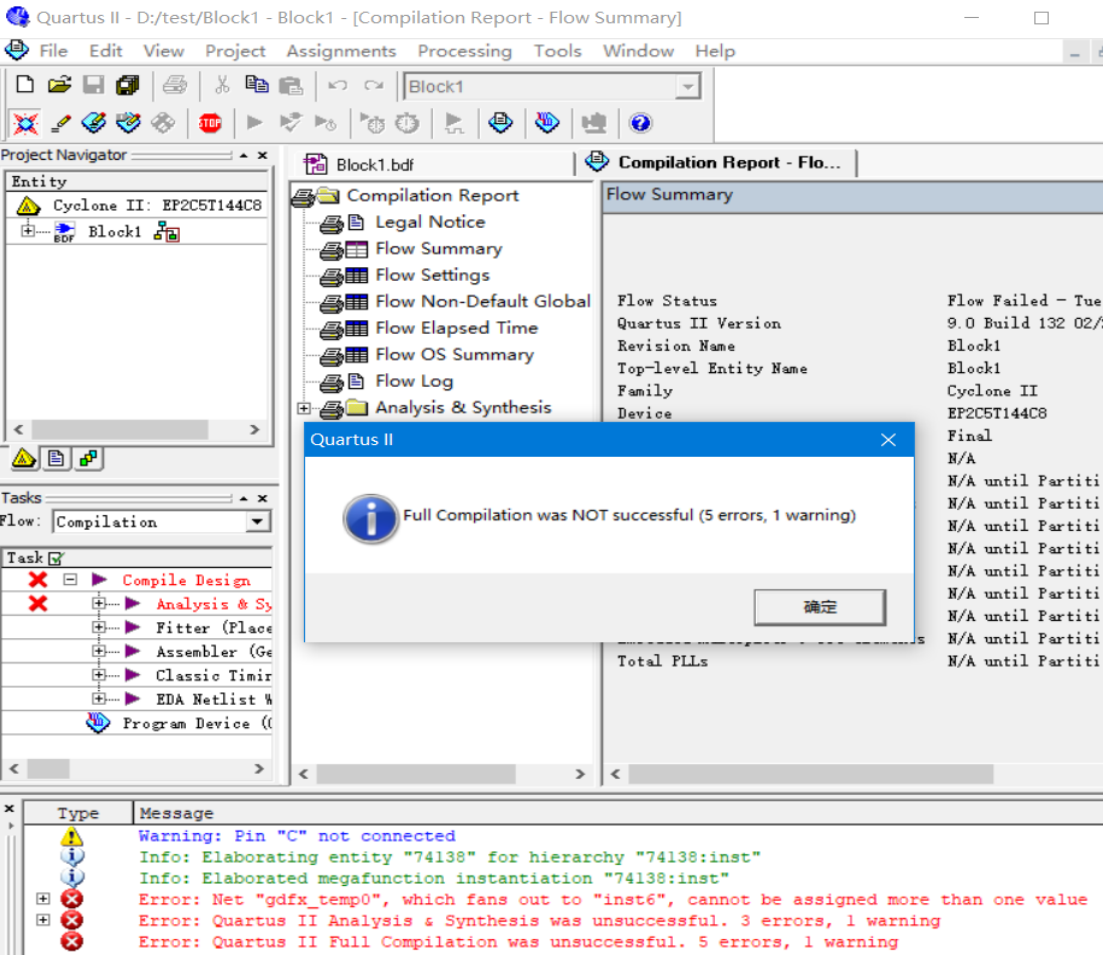


图 2-16 全程编译后出现报错信息

**注意：**如果发现报告多条错误信息，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。

Processing 栏出现的蓝色文字是 Warning（报警）信息，有的 Warning 信息不影响编译结果的正常功能。

如编译成功，会出现提示，工程管理窗显示出工程 BLOCK1 的层次结构和其中结构模块耗用的逻辑宏单元数；在此栏下是编译处理流程，包括数据网表建立、逻辑综合（Synthesis）、适配(Fittering)、配置文件装配(Assembling)和时序分析(Classic Timing Analysizing)等；最下栏是编译处理信息；中栏（Compilation Report）是编译报告项目选择菜单，单击其中各项可以详细了解编译与分析结果。

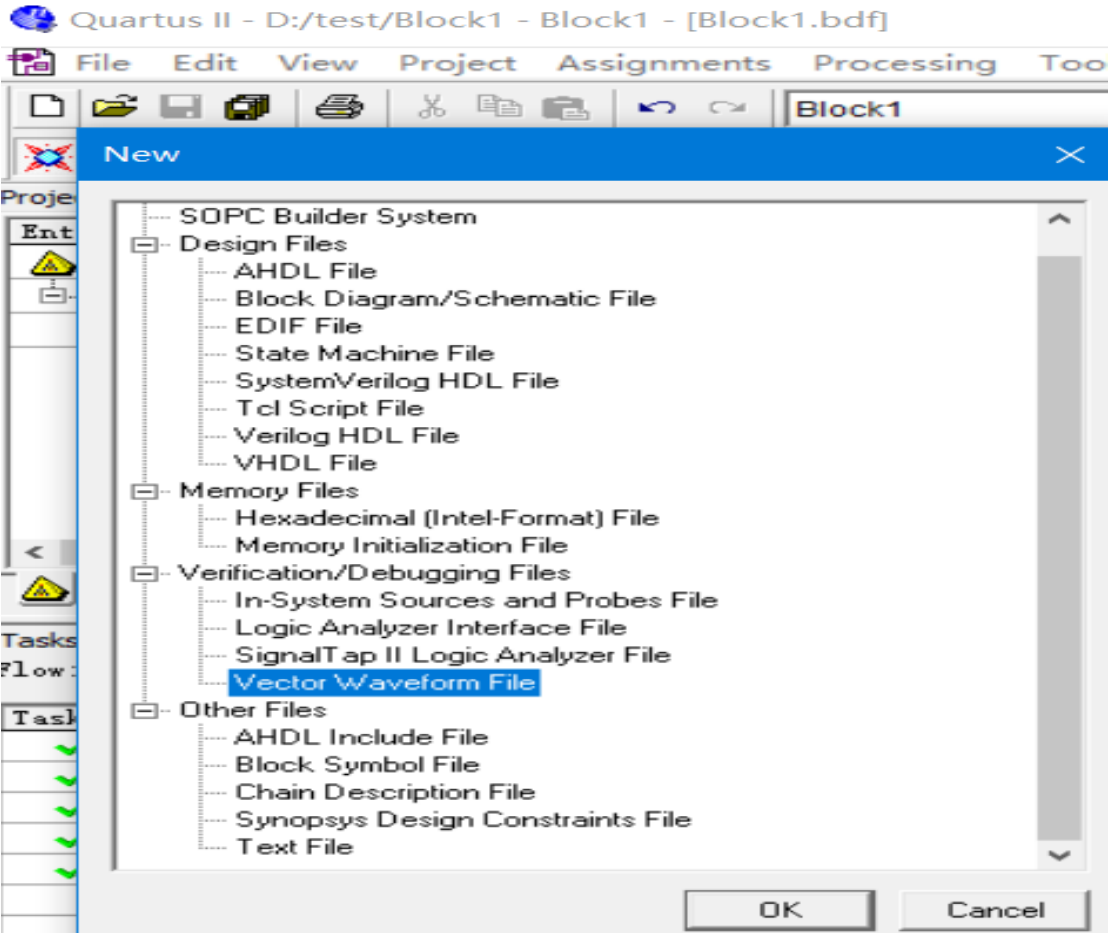
例如单击 Flow Summary 项，将在右栏显示硬件耗用统计报告，其中报告了当前工程耗用了 2 个逻辑宏单元、0 个内部 RAM 位等。

如果单击 Timing Analyzer 项的 tpd，则能看到当前工程电路的延时信息。

### 3) 仿真（逻辑功能测试）

编译成功后，需要通过仿真对电路功能和时序性质进行测试，以了解设计结果是否满足原设计要求，仿真流程如下：

a. 新建波形。点 New，选择 Vector Waveform File，单击 OK，即出现空白的波形编辑器，注意默认名字是 Waveform1.vwf。





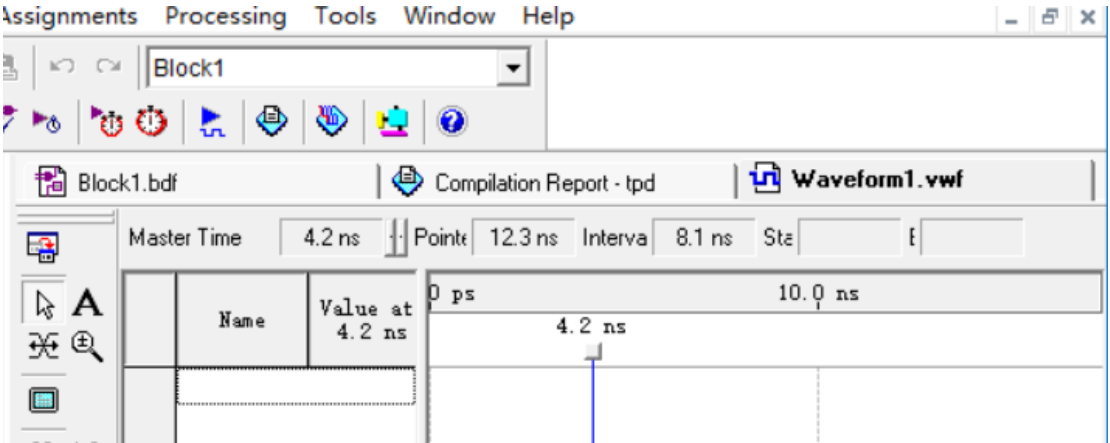


图 2-17 波形编辑器

b. 设置仿真时间。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：Edit→End Time，Time 栏输入 50，单位选“μs”，整个仿真域的时间即设定为 50μs，单击 OK 结束设置。

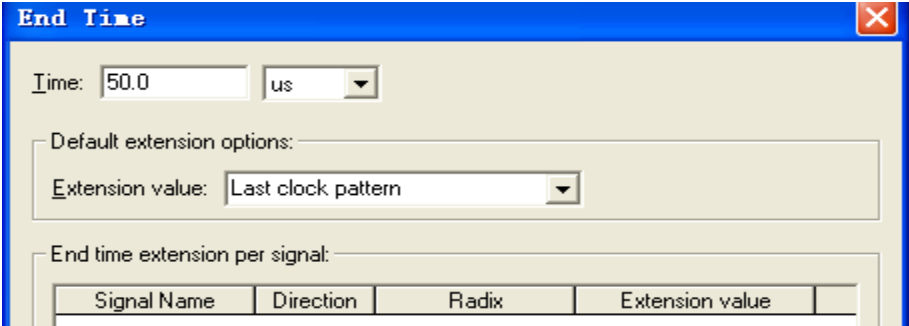


图 2-18 设置仿真时间长度

- c. 波形文件存盘。点保存按钮——保存，以默认名 BLOCK1.vwf 将波形文件存入 D:\TEST 中。
- d. 将工程 BLOCK1 的端口信号名选入波形编辑器中。View→Utility Windows→Node Finder (Alt+1)，弹出图 2-19 所示对话框，在 Filter 框中选 Pin: all（默认项），单击 List 按钮，在下方的 Nodes Found 窗口中出现设计中的 BLOCK1 工程所有端口名。（保持窗口浮动：右键点击窗口边框→不选 Enable Docking）

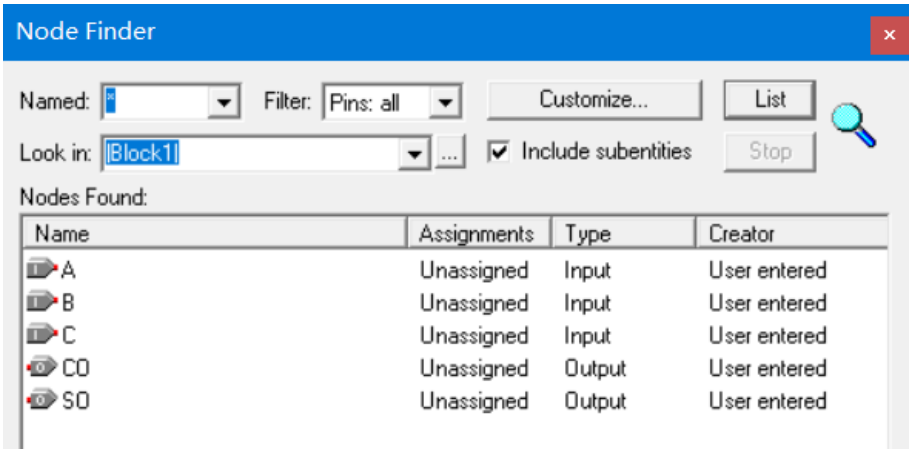


图 2-19 Node Finder 窗口



注意：如果点 List 不显示工程端口名，需要重新全程编译。

用鼠标将端口名 AI、BI、CI、CO、SO 拖到波形编辑窗。关闭 Node Finder 窗口。全屏显示波形窗，点击放大缩小按钮<sup>④</sup>后可用鼠标左键或右键调整编辑区，使仿真坐标处于适当位置，如图 2-20 所示，这时仿真时间横坐标设定在数十微秒数量级。

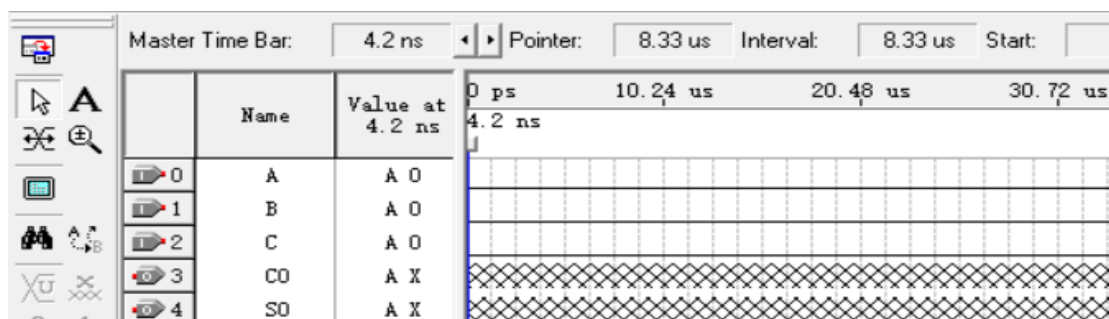


图 2-20 波形编辑窗口

加入端口名的另一种方法：双击波形窗口左侧空白处——弹出 Insert Node or Bus 窗口——点 Node Finder——点 List——点>>——点 OK——点 OK。

e. 编辑输入波形（输入激励信号）。即对输入引脚 AI、BI、CI 设置输入信号，以便通过观察输出信号来了解电路功能。单击<sup>⑤</sup>按钮，用鼠标拖动某个输入信号使之变蓝，单击左列高电平设置按钮<sup>①</sup>，如图 2-21。再次存盘。

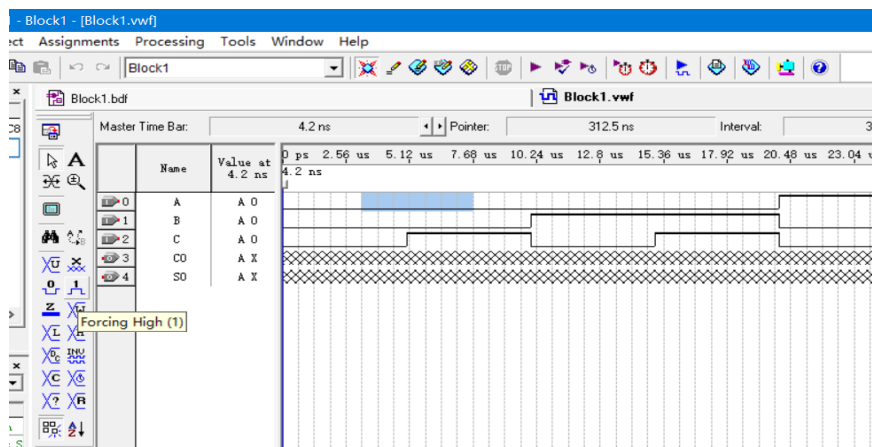


图 2-21 设置波形激励图

输入波形快速设置：右击波形编辑窗口——Zoom——Fit in Window，Name 栏右击输入端口名——Value——Count Value——选 Timing——Count every 处分别改成仿真时间的 1/2、1/4、1/8，单位选 us——确定。

f. 仿真参数设置。Assignments→Settings，选择 Category→Simulator Settings（图 5-23），右侧 Simulation mode 项选择 Timing（时序仿真）并选择仿真激励文件名 BLOCK1.vwf（默认）。选中 Run simulation until all vector stimuli are used（全程仿真）。

选择 Simulation Verification 栏，确认选定 Simulation coverage reporting; Glitch detection（毛刺检测）为 1ns 宽度。

g. 启动仿真器。点击蓝色三角，（或 Processing→Start Simulation 或 ctrl+I），出现 Simulator was successful。

h. 观察仿真结果。仿真波形文件 Simulation Report 通常会自动弹出，如图 2-22。

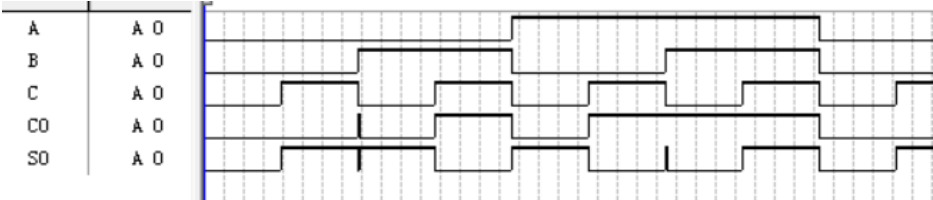



图 2-22 仿真波形输出

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) 是两个不同文件。

**手动打开仿真波形报告：**Processing→Simulation Report (ctrl+shift+r)；

**要展开时间轴上所有波形：**右键单击编辑窗→Zoom→Fit in Windows (ctrl+w)。

参考相应真值表，由图 2-23 的输入输出波形可知，电路功能符合设计要求。

**延时测量：**点击仿真报告窗口左侧工具栏的放大镜 (Zoom Tool)，在波形窗口某个输出信号电平变化的附近连续单击，即可放大波形，可以看到信号从 C 输入，从 SO 输出的延时情况，单击  按钮，对图中 C 上升沿处右键单击——Time Bar——Insert Time Bar——选中 Make master time bar——点 OK，即出现主时间标尺，同样方法在 SO 变化处插入副时间标尺（不选中 Make master time bar!!!），副时间标尺上方带有正（负）号的时间即为此电路当实现于 EP2C5T144C8 器件中两个指定端口的精确延时。这就是时序仿真的优势。

注：标尺会主动对齐整数时间，可用鼠标拖拽使之对齐信号变化处，同时计数。如不区分主、副时间标尺，则需要手动做差值计算。

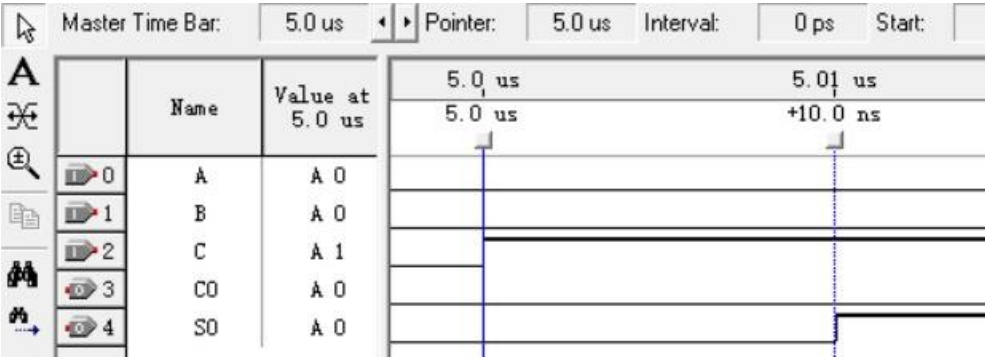


图 2-23 C 与 SO 的延时波形显示

如输出有毛刺出现，可在毛刺附近放大波形，然后按下图所示测量延时。

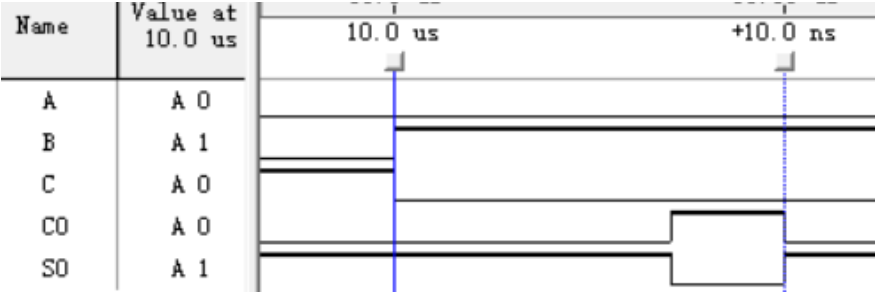


图 2-24 带有毛刺的延时测量

#### 4) 引脚锁定

前面的功能测试尽管实现了精确的时序分析，但仍然是基于计算机平台的，而最终和最重要的测试必须是基于硬件电路的测试。为此，应将其输入输出信号锁定在指定芯片（EP2C5144C8）确定的引脚上，编译后将设计文件下载于此芯片中，以便对电路设计进行硬件测试，最终确定本项设计是否满足要求。

假设要像图 2-25 那样锁定引脚，即 A、B、C、S0、C0 分别锁定在 75、79、87、52、121 引脚上：

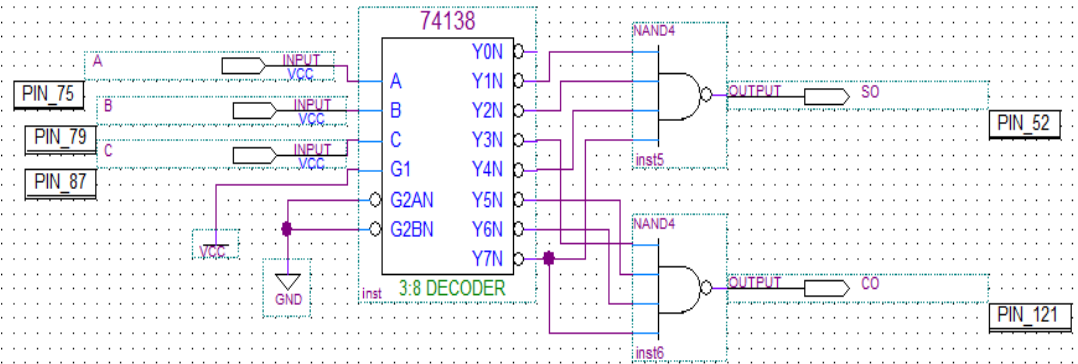


图 2-25 图 2-8 所示电路于 EP2C5144 内的引脚锁定情况

引脚锁定：

- a) Assignments→Assignments Editor，进入图 2-26 所示的编辑窗。
- b) 在 Category 栏中选 Pin。
- c) 双击 T0 栏的 new，在出现的下拉栏中选择要锁定的端口信号名；然后双击对应的 Location 栏的 new，在下拉栏中选择要锁定的器件引脚号，或直接键入引脚号并回车。

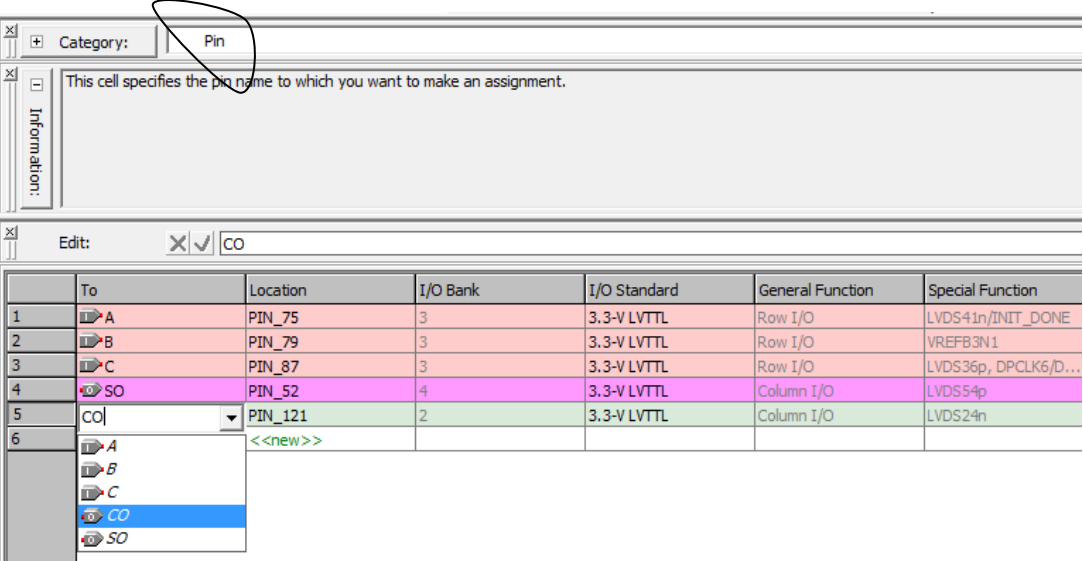


图 2-26 引脚锁定对话框

常用引脚——输入：75、79、87、92；输出：52、121、122。不能用 80、81、76。

- d) 全部引脚锁定后，必须再编译一次，才能将引脚锁定信息编译进编程下载文件中，

此后就可以将编译好的 SOF 文件下载到实验系统的 FPGA 中去了。

注意：Assignments Editor 窗口还能对引脚做进一步的设定，如 I/O Standard 栏可设置每一信号的 I/O 电压；Reserved 栏可对某些空闲的 I/O 引脚电气特性做设置；而在 Signal Probe 等选择栏，可对指定信号做探测信号的设定。这些一般不做设置，选择默认。

e) 引脚锁定的另一种方法——图形方式：Assignments→Pins，弹出引脚编辑窗 Pin Planner (ctrl+shift+n)，用鼠标将编辑窗左侧的信号名逐个拖入右侧器件对应引脚上即可。这种方法适合于引脚数量较少的目标器件。

### 5) 编程下载

引脚锁定并编译完成后，Quartus II 将生成多种形式的针对所选目标 FPGA 的编程文件，其中最主要的是 POF 和 SOF 文件。前者是编程目标文件，用于对配置器件编程；后者是静态 SRAM 目标文件，用于对 FPGA 直接配置，在系统直接测试中使用。这里首先将 SOF 文件通过 JTAG 口（是一个标准测试口）载入 FPGA 中，进行硬件测试，步骤如下：

a) 打开编辑窗和配置文件。连接好实验系统所需导线，打开电源，点击 Programmer 按钮(或 Tool→Programmer)，弹出图 2-27 所示编程窗。Mode 栏中有四种编程模式可以选择，默认选择 JTAG，并选中打勾下载文件右侧的第一个小方框 (Program/Configure)，仔细核对下载文件路径与文件名。如果文件没有出现或有错，单击左侧 Add Files 按钮，手动选择配置文件 BLOCK1.sof。

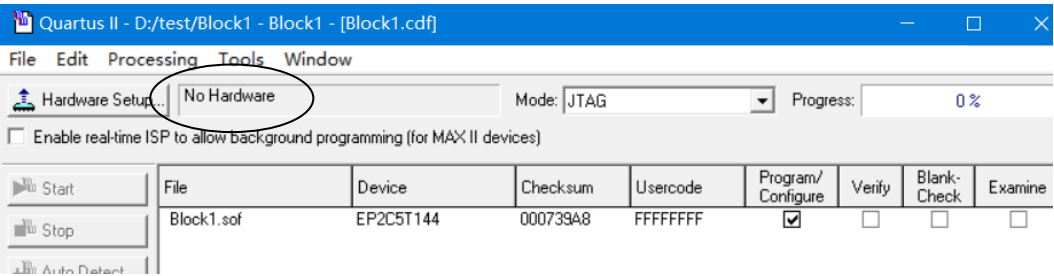


图 2-27 选择编程下载文件和下载模式

b) 设置编程器。如上图画圈处为 No Hardware，点击 Hardware Setup 按钮，在弹出的图 2-28 所示对话框中选择 Hardware Settings 页，可以看到当前可用的编程器，在 Currently selected hardware 右侧选择可用编程器，即 USB-Blaster，点 Close。如可选编程器为空，则需拔、插一次下载器左侧的 USB 连接线并重新打开 Hardware Setup 窗口。



图 2-28 选择编程器

- c) 下载。点击 Start 按钮，开始对目标器件 FPGA 的配置下载操作。当 Progress 显示出 100%，以及底部的处理信息栏中出现“Configuration Succeeded”时表示编程成功。
- d) 硬件测试。成功下载后，将已锁定的三个输入引脚端 75、79、81 分别与三个电平开关相连，将两个输出引脚端 52、121 分别与数码管输入端相连。改变实验系统上的输入信号，观察输出信号，对比图 2-22 的波形，验证电路功能。
- e) 编程配置器件（可选项）。由于是对 FPGA 进行设计，如果最终要实现产品级设计，还要完成对 Flash 配置器件的编程，即将以上测试用的配置文件 BLOCK1.sof 对应的编程文件 BLOCK1.pof 烧写到此 FPGA 专用 Flash 存储器中，如 EPCS1 等。
- 以上就是利用 Quartus II 进行电路设计的完整流程。

## 二. 用 Quartus II 设计简单电路

在掌握上述设计流程后，依照图 2-29 做电路设计，完成整个流程；也可以利用 Quartus II 完成实验一中的器件功能测试和多路开关的设计。

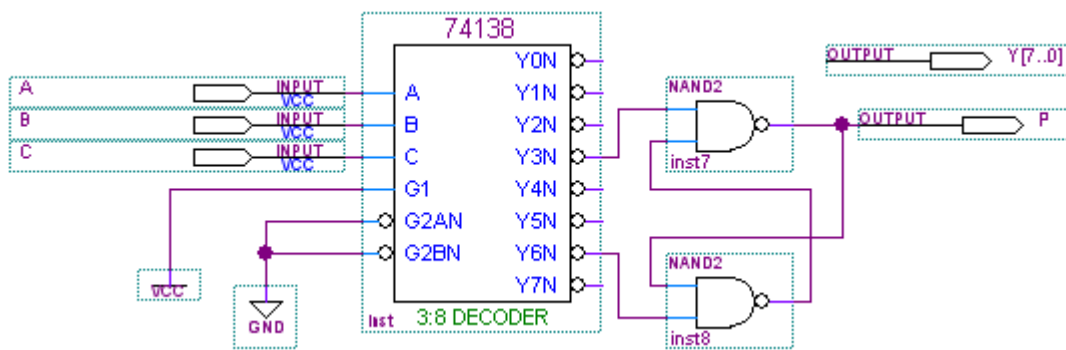


图 2-29 脉冲发生器示例电路图

## 实验结果：

用最简练的语言总结说明 Quartus II 设计电路的完整流程。

## 2.3 设计三变量多数表决器

### 实验目的：

通过设计三变量多数表决器掌握 Quartus II 的使用方法；

### 实验内容：

使用 Quartus II 设计一个三变量多数表决器（要求用与非门实现）。

### 实验步骤

1. 课前预习：
  - 1) 根据逻辑关系画出三变量多数表决器的真值表；
  - 2) 根据真值表得表达式，对表达式进行化简、变换，要求用与非门实现；
  - 3) 画出电路图。
2. 课堂利用 Quartus II 对课前设计好的电路进行输入、编译、仿真、引脚锁定、编程下载；
3. 硬件测试，利用实验系统上的数据开关做输入，数码管显示输出，验证设计是否符合要求，学习如何测量输出延时  $\Delta t$ 。
4. 如有空余时间，可尝试设计原/反码发生器（利用异或门实现）。
5. 将实验结果描述于实验报告上，完成实验报告。
6. 总结利用 Quartus II 设计数字电路的方法、步骤。

## 2.4 设计一位全加减器

### 实验目的：

通过设计一位全加减器掌握 Quartus II 的使用方法；

### 实验内容：

使用 Quartus II 设计一个一位全加减器（用一个 74LS00 和一个 74LS86 实现）。提示：可设置一控制变量 M，用来控制做加法或是减法。

### 实验步骤

1. 课前预习：
  - 1) 画出一位全加减器的真值表；
  - 2) 根据真值表画卡诺图；
  - 3) 根据卡诺图得表达式，对表达式进行化简、变换，要求用一个 74LS00 和一个 74LS86 实现；
  - 4) 画出电路图。
2. 课堂利用 Quartus II 对课前设计好的电路进行输入、编译、仿真、引脚锁定、编程下载；
3. 硬件测试，利用实验系统上的数据开关做输入，数码管显示输出，验证设计是否符合要求，测量输出延时  $\Delta t$ 。
4. 如有空余时间，可预习 3-8 线译码器 74138 的具体功能。
5. 将实验结果描述于实验报告上，完成实验报告。
6. 总结利用 Quartus II 设计数字电路的方法、步骤。

## 2.5 译码器相关设计

### 实验目的：

通过译码器相关设计熟练掌握 Quartus II 的使用方法；

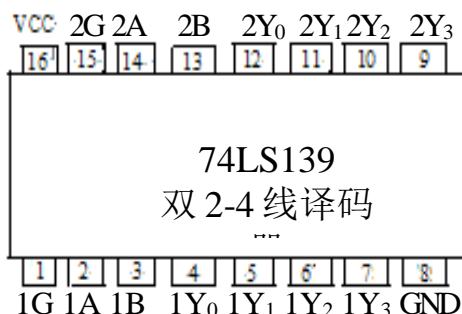
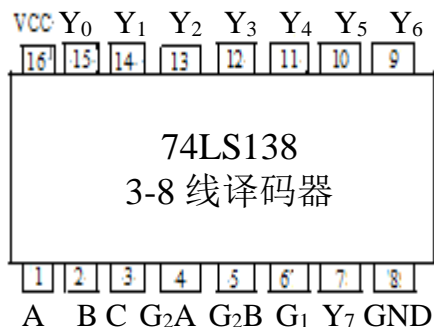
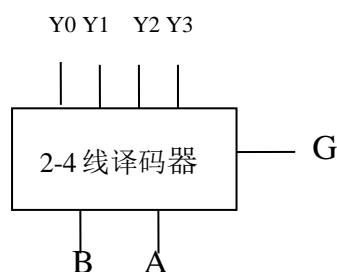
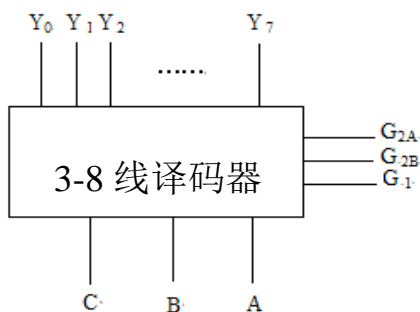
### 实验内容：

使用 Quartus II 设计全加器、全减器（用 74138 和与非门实现）。

### 实验步骤

1. 课前预习：
  - 1) 用译码器 74138 和与非门设计一位全加器和全减器。
  - 2) 画出电路图。
2. 课堂利用 Quartus II 对课前设计好的电路进行输入、编译、仿真、引脚锁定、编程下载。
3. 硬件测试，利用实验系统上的数据开关做输入，数码管显示输出，验证设计是否符合要求，测量输出延时  $\Delta t$ 。
4. 如有空余时间，可尝试用两个 3-8 线译码器构成 4-16 线译码器（或两个 2-4 构成一个 3-8）。
5. 将实验结果描述于实验报告上，完成实验报告。
6. 总结利用 Quartus II 设计数字电路的方法、步骤。

### 管脚图：





## 2.6 中期测验

### 2.7 触发器特性学习及相关设计

#### 实验目的：

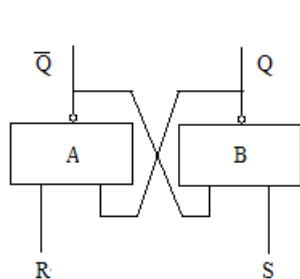
学习使用 Quartus II 设计时序逻辑电路；

#### 实验内容：

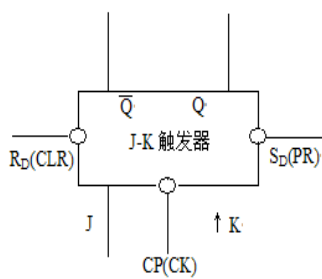
测试基本 RS 触发器、JK 触发器、D 触发器的逻辑功能，使用 Quartus II 设计一个移位寄存器。

#### 实验步骤

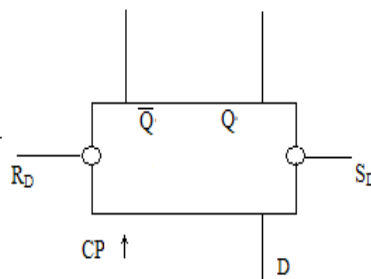
1. 课前预习：
  - 1) 分析 RS 触发器、JK 触发器、D 触发器的逻辑功能，写出表达式，画出逻辑关系表；
  - 2) 用 D 触发器设计一个 3 位的移位寄存器；
  - 3) 画出电路图。
2. 课堂利用 Quartus II：
  - 1) 分析 RS 触发器、JK 触发器、D 触发器的逻辑功能，对照验证逻辑关系表；
  - 2) 对课前设计好的移位寄存器电路进行输入、编译、仿真、引脚锁定、编程下载；
3. 硬件测试，利用实验系统上的数据开关做输入，时钟源或脉冲输出键做 CP，数码管显示输出，验证设计是否符合要求，测量输出延时  $\Delta t$ 。
4. 输入、验证预先设计好的移位寄存器。
5. 如有空余时间，可尝试用 D 触发器设计计数器、二分频、三分频电路。
6. 将实验结果描述于实验报告上，完成实验报告。



RS 触发器



JK 触发器



D 触发器

## 2.8 设计计数器

### 实验目的:

练习使用 Quartus II 设计时序逻辑电路;

### 实验内容:

使用 D 触发器设计一个同步 8421 BCD 码的十进制加法计数器。要求用两种方法 (D 触发器和 VHDL 语言) 分别实现。

### 实验步骤

#### 1. 课前预习:

两种方法实现设计:

- 1) 使用 D 触发器和一些门电路, 设计一个同步 8421 BCD 码的十进制加法计数器。
  - 2) 借助于 Quartus II, 使用一般模型理论和自动化设计技术在实验系统上的 FPGA 中实现相同功能;
2. 课堂利用 Quartus II 对课前设计好的电路进行输入、编译、仿真、引脚锁定、编程下载;
  3. 硬件测试, 利用实验系统上的数码管显示输出, 验证设计是否符合要求。
  4. 将实验结果描述于实验报告上, 完成实验报告。