

2020 ~ 2021 学年

数
字
逻
辑
电
路
实
验
A

东南大学电工电子实验中心
2020年10月

目 录

教学计划和要求	1
基本情况:	1
教学要求:	1
预习要求:	1
实验要求:	1
实验报告要求:	2
实验教学计划	2
评分和考核方法	2
慕课	3
本学期开放时间安排	3
课程准备事项	4
实验预习	4
准备导线	4
准备面包板	4
准备工具	5
下载并安装相关软件	5
预习报告样例	6
实验内容	6
实验设计方案	6
测试方案	7
电路搭接调试方法样例	9
分析电路	9
合并器件	9
查询数据手册	9
根据数据手册选择器件, 分配管脚	10
元件布局	11
布线	11
通电前检查	13
通电验证	13
故障调试	13
故障举例:	14
实验一 组合逻辑电路	15
学习目标:	15
时间要求:	15
预备知识:	15
预习要求	16
必做实验	16
选作实验	17
实验二 可编程数字逻辑设计基础	18
学习目标	18

时间要求	18
预备知识	18
预习要求	19
必做实验	20
选做实验	20
实验三 时序逻辑电路	21
学习目标	21
时间要求	21
预备知识	21
预习要求	21
必做实验	21
实验四 FPGA 时序逻辑设计	23
学习目标	23
时间要求	23
预备知识	23
预习要求	29
必做实验	29
实验五 FPGA 小系统设计	30
学习目标	30
时间要求:	30
预备知识	30
预习要求	31
必做实验	31
附录一 Fritzing 简明操作指南	32
附录二 Quartus II 简明操作指南	35
一、 设计项目输入	35
1.1 建立工程项目	35
1.2 建立原理图输入文件	36
二、 设计项目处理	38
三、 设计项目校验	39
3.1 建立输入激励波形文件 (.vmf):	39
3.2 为输入信号建立输入激励波形:	39
3.3 功能仿真	40
3.4 时序仿真	40
四、 器件编程	40
4.1 Quartus II 器件编程	40

教学计划和要求

基本情况:

- 1、总学时: 32 学时
- 2、学时比例: 1 (课内): 1.5 (课外)
- 3、学 分: 1 学分
- 4、适用范围: 2019 级 6、8、11、22 系学生
- 5、先修课程: 高等数学、物理、数字逻辑电路
- 6、时 间: 2020.11~2021.1
- 7、实验教材: 《数字逻辑设计实践》徐莹隽、常春等编著, 高等教育出版社

教学要求:

预习要求:

- 1) 进实验室前**必须**完成本次实验预习报告, 内容包括
 - a) 实验电路功能描述、系统框图和详细设计过程
 - b) 实验电路原理图
 - c) 实验电路硬件连接图
 - d) 如果实验有预习思考题, 提交预习思考题答案
- 2) 预习报告写在实验报告的实验原理部分, 如没有把握, 也可先写在其他纸上, 实验完成后誊写到实验报告上
- 3) 实验电路**必须**在进实验室前完成搭接, 实验室内以电路调试为主
- 4) 电工电子实验中心的网站地址为: <http://10.1.8.30/LimsCMS/>, 预习时请访问该平台以了解您准备完成的实验内容是否有变动、提示或其他通知。

实验要求:

- 1) 实验采用集中授课+开放实验模式, 第一次实验时请仔细阅读开放实验规章制度, 并在以后的实验中认真遵守
- 2) 每次实验要带校园卡、元器件、单股连接线、丁字线和所发的面包板。丁字线、面包板请妥善保管, 勿丢失或损坏, 否则将照价赔偿
- 3) 在进入实验室时, 请在刷卡机上出示您的校园卡, 在刷卡机分配的实验室和实验座位上完成实验, 严禁串座
- 4) 实验前先检查自己座位上的仪器, 如有缺失或损坏, 请及时通知指导老师处理。实验过程中如果发生仪器故障, 应该和值班教师检查确认后, 用备用仪器更换, 严禁自己用其他实验座位上的仪器更换
- 5) 断电按实验设计方案搭接电路, 认真检查确保无误后方可通电测试
- 6) 认真记录实验数据和实验波形, 所有数据和波形都要分析判断, 确认正确
- 7) 要求实物验收的实验内容, 完成并记录所有实验数据后, 提请指导老师验收, 验收通过后方可拆除电路结束实验
- 8) 实验过程中遇到故障要独立思考, 耐心查找故障原因并排除, 记录故障现象、排除故障的过程和方法
- 9) 实验中若发生异常现象, 应立即切断电源, 并通知指导老师处理。如有元器件损坏, 可到仪表室购买

- 10) 开放实验每次至少要完成一项实验内容，单次实验时间不少于 1 个小时
- 11) 实验中请勿随意离开实验室，如确实有特殊情况请向指导教师请假
- 12) 实验完成后请将仪器归位并关闭仪器电源、整理线缆、打扫干净实验桌面，然后刷卡下机

实验报告要求：

1. 实验报告应该包括以下几个部分
 - ✧ 实验目的和要求
 - ✧ 实验原理，包括实验电路的设计过程、系统框图、原理图、硬件连接图、测试方案等，要求在实验前完成
 - ✧ 实验仪器，实验中用到的仪器设备
 - ✧ 实验记录，记录实验具体步骤、原始数据、实验过程、实验中遇到的故障现象、排除故障的过程和方法等
 - ✧ 实验分析，对实验结果进行分析比对
 - ✧ 实验小结，总结实验完成情况，对设计方案和实验结果做必要的讨论，简述实验收获和体会
 - ✧ 实验思考题，如有，提交思考题答案
 - ✧ 参考资料，记录实验过程阅读的有关资料，包含资料名称、作者等
2. 数字电路实验结果验证一般包括真值表验证、波形验证、功能验证
 - ✧ 真值表验证结果，要求用表格记录
 - ✧ 波形验证结果，要求记录在坐标纸上，记录的时候注意多路波形之间的时序关系
 - ✧ 功能验证结果，要求用表格记录
3. 实验报告必须在指定时间完成并提交

实验教学计划

周次	内容	实验模式	学时	教材
06	组合逻辑电路	集中授课	3	第 1、2、3 章
07		开放实验	3	
08	可编程数字逻辑设计基础	集中授课	3	第 2、7 章
9	时序逻辑电路	集中授课	3	第 4 章
10		开放实验	3	
11		集中验收	3	
12	FPGA 时序逻辑设计	集中授课	3	第 4、7 章
13	FPGA 系统设计	集中授课	3	第 6、7 章
14		集中验收	3	
15		集中验收	3	
16	期末考试	集中上课	3	

评分和考核方法

实验过程采用积分考核方式，总积分为 600，实验总评采用等级制，等级和积分对应关系为：优（540

- 600), 良 (480 - 539), 中 (420 - 479), 及格 (360 - 419), 不及格 (0 - 359), **期末卷面成绩低于 45 分, 总成绩不及格**。具体积分分布如下表所示

实验名称	预习报告	预搭接	实验验收	实验报告
组合逻辑电路	10	5	25	20
可编程数字逻辑设计基础	10	0	30	20
时序逻辑电路 (第 9 周实验)	10	5	20	20
时序逻辑电路 (第 10 周实验)	10	5	20	
时序逻辑电路 (第 11 周实验)	10	5	30	
FPGA 时序逻辑设计	15	0	30	20
FPGA 系统设计	20	0	80	20
期末考试	120			
其他 (实验态度、考勤、违纪违规)	40			

慕课

课程已经上线[中国大学 MOOC](#)平台, 可直接点击链接加入或扫描下面课程二维码加入课程学习, 为了便于管理, 昵称建议设置为 “seu+学号 +姓名”。



本学期开放时间安排

1、开放实验时间

- ◇ 周一 14:00 ~ 21:00
- ◇ 周二 09:30 ~ 21:00
- ◇ 周三 09:30 ~ 21:00
- ◇ 周四 09:30 ~ 21:00
- ◇ 周五 09:30 ~ 16:30
- ◇ 周六 09:30 ~ 16:30

2、法定假期和校历规定的假期, 实验室不开放

3、开放时间临时变更, 将在实验中心主页上提前通知, 来实验室前最好访问实验中心主页确认是否开放

课程准备事项

在首次课前，每位同学必须完成以下准备工作

实验预习

- 1、完成实验 1 所有必做实验的设计，并将设计方案、原理图、硬件连接示意图等写在预习报告上或者实验报告的原理部分
- 2、至少完成实验 1 中“数值判别电路”的电路搭接，如有能力，可将所有电路都预搭接好

准备导线

面包板搭试电路时，使用的是直径为 0.6mm 左右的单股导线。在课程开始前，请按下表准备好导线，所有导线两侧线头都要剥好，线头剥离长度约为 6mm 左右，线头剪成 45° 斜口。这样有利于线头全部插入底板以保证接触良好，同时裸线又不会露在外面与其它导线短路。表格内的线长不包含两边线头的长度，也可自行从市场上购买成品面包板用跳线，如下图所示。

第一次上课时，指导老师将检查导线准备情况，检查结果计入第一次实验的预习成绩。

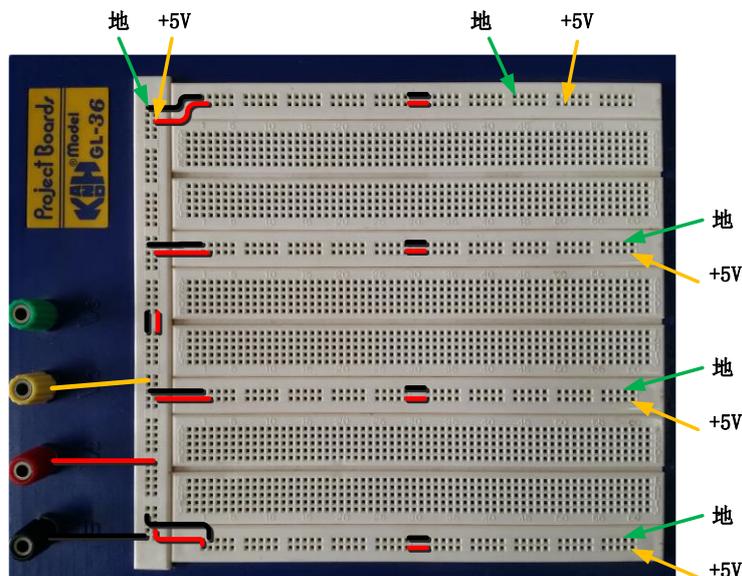
线长 (mm)	数量
2.5	6
5	6
7.5	6
10	4
20	4
50	4



准备面包板

我们实验主要通过面包板完成电路搭试，在首次实验课前，请按下图所示连接好跳线。请根据面包板的结构，自己分析一下，这样的跳线有什么好处。

第一次上课时，指导老师将检查面包板准备情况，检查结果计入第一次实验的预习成绩



准备工具

为了提高实验效率，我们建议自备以下实验工具：

剥线钳：用于剥除电线头部的表面绝缘层，常用的剥线钳有两种，如图所示，两种功能基本相同，下面的那种更省力些。剥线钳使用时要掌握以下要点：

1. 根据导线直径，选好剥线钳刀片的孔径。
2. 根据缆线的粗细型号，选择相应的剥线刀口
3. 将准备好的电缆放在剥线工具的刀刃中间，选择好要剥线的长度
4. 握住剥线工具手柄，将电缆夹住，缓缓用力使电缆外表皮慢慢剥落
5. 松开工具手柄，取出电缆线，这时电缆金属整齐露出外面，其余绝缘塑料完好无损



镊子：我们平时搭电路时，如果用手直接拿住导线往插孔里插，很容易造成导线弯曲，而用镊子夹住导线或元器件的引脚送入到指定的面包板插孔，导线则不易弯曲。镊子也常用于折弯、捋直导线或元件管脚，起拔集成电路等。镊子的种类很多，我们实验中一般选择左图所示的尖头镊子。

剪刀或斜口钳：用来剪断导线、元件脚，其中斜口钳切断能力要比剪刀强很多，剪切不费力。斜口钳剪线时，应将线头朝向下，以防止断线时伤及眼睛或其他同学，不要用来剪断铁丝或其他金属的物体，也不要用来剪线径过粗的导线，以免损伤刃口。斜口钳在很多场合也可以用指甲钳代替。



DIP 封装集成电路起拔器：用于将双列直插封装的集成电路从面包板上取下，同时不损伤器件。使用时，将起拔器的两个爪，插入到集成电路的两侧，然后握紧夹子，轻轻上提将集成电路拔出。无该工具时，可用镊子代替将镊子插入面包板的中间槽中，在集成电路的两头来回轻轻地往上撬，切不可仅在一头用力撬，那样容易将集成电路的引脚折弯。

下载并安装相关软件

数字逻辑设计实践课程要用到的软件包括电路仿真软件 Multisim 2013，Intel 公司 FPGA 开发平台 Quartus II 以及虚拟面包板软件 fritzing。请自行下载安装，或上课时到指导老师处拷贝。

- 1、 易派实验箱相关软件和资料 [校内网盘下载](#)
- 2、 Multisim 2013 (文件大小 552M) [校内网盘下载](#)
- 3、 Quartus 9.1 sp2 Web Edition [校内网盘下载](#)
- 4、 fritzing.0.9.3b 64bit: [校内网盘下载](#)
- 5、 fritzing.0.9.3b 32bit: [校内网盘下载](#)

预习报告样例

实验内容

人类有四种主要血型：A, B, AB 和 O 型。为了避免输血反应，输血者和受血者之间要遵循下列原则：

- ✧ 同一血型之间可以相互输血。
- ✧ AB 型受血者可以接受任何血型的输血。
- ✧ O 型输血者可以给任何血型的受血者输血。

实验设计方案

1. 输入、输出信号编码

输入信号：用二位二进制数 G_1 、 G_0 代表输血者的 4 种血型， R_1 、 R_0 代表受血者的 4 种血型，编码代表的具体意义如下表所示

输出信号：S 代表是否满足输血/受血条件，“1”满足，“0”不满足

2. 列出真值表

根据题目要求，列出真值表

G_1	G_0	R_1	R_0	S	G_1	G_0	R_1	R_0	S
0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	1	1	0	1	0	1
0	0	1	1	1	1	0	1	1	1
0	1	0	0	0	1	1	0	0	0
0	1	0	1	1	1	1	0	1	0
0	1	1	0	0	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1

表 2 实验真值表

输血者			受血者		
G_1	G_0	血型	R_1	R_0	血型
0	0	O 型	0	0	O 型
0	1	A 型	0	1	A 型
1	0	B 型	1	0	B 型
1	1	AB 型	1	1	AB 型

表 1 输入输出信号编码

G_1G_0	00	01	11	10
R_1R_0 00	1	0	0	0
01	1	1	0	0
11	1	1	1	1
10	1	0	0	1

图 2 卡诺图

3. 逻辑化简

根据真值表画出卡诺图，如图 2 所示，化简得到与或非表达式如式 1，考虑到只有与非门器件，转化，得到与或非表达式 2。

$$S = \overline{G_1} \overline{G_0} + \overline{G_1} R_0 + \overline{G_0} R_1 + R_1 R_0 \quad (1)$$

$$S = \overline{\overline{G_1} \overline{G_0} \bullet \overline{G_1} R_0 \bullet \overline{G_0} R_1 \bullet R_1 R_0} \quad (2)$$

4. 逻辑电路图

根据逻辑函数表达式 (1)，绘制出电路原理图如图 3 所示（反变量所需的非门未画出），其中输入信号的原变量从实验箱上的逻辑电平开关获得，反变量需要另外接非门来实现。

5. 硬件连接示意图

根据电路原理图，查询教材附录“常用集成电路型号和引脚图”，可知，硬件电路需要 4 个两输入与非门，可以选择 1 片 74HC00 实现；需要 1 个四输入与非门，可以选择 1 片 74HC20 实现。硬件连接示意图如图 4 所示，并将管脚分配反标注到电路原理图中。也可利用 Fritzing 软件绘制硬件连接示意图，Fritzing 软件的具体使用方法参看附录一。

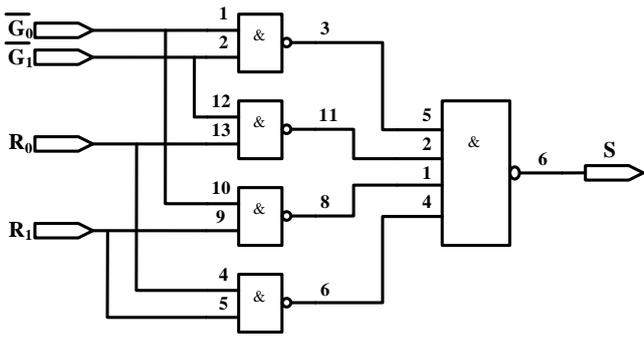


图 3 电路原理图

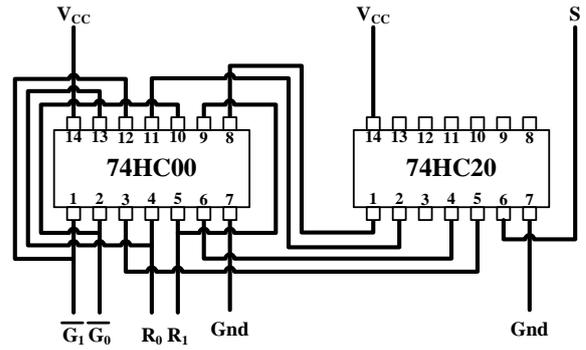


图 4 硬件连接图

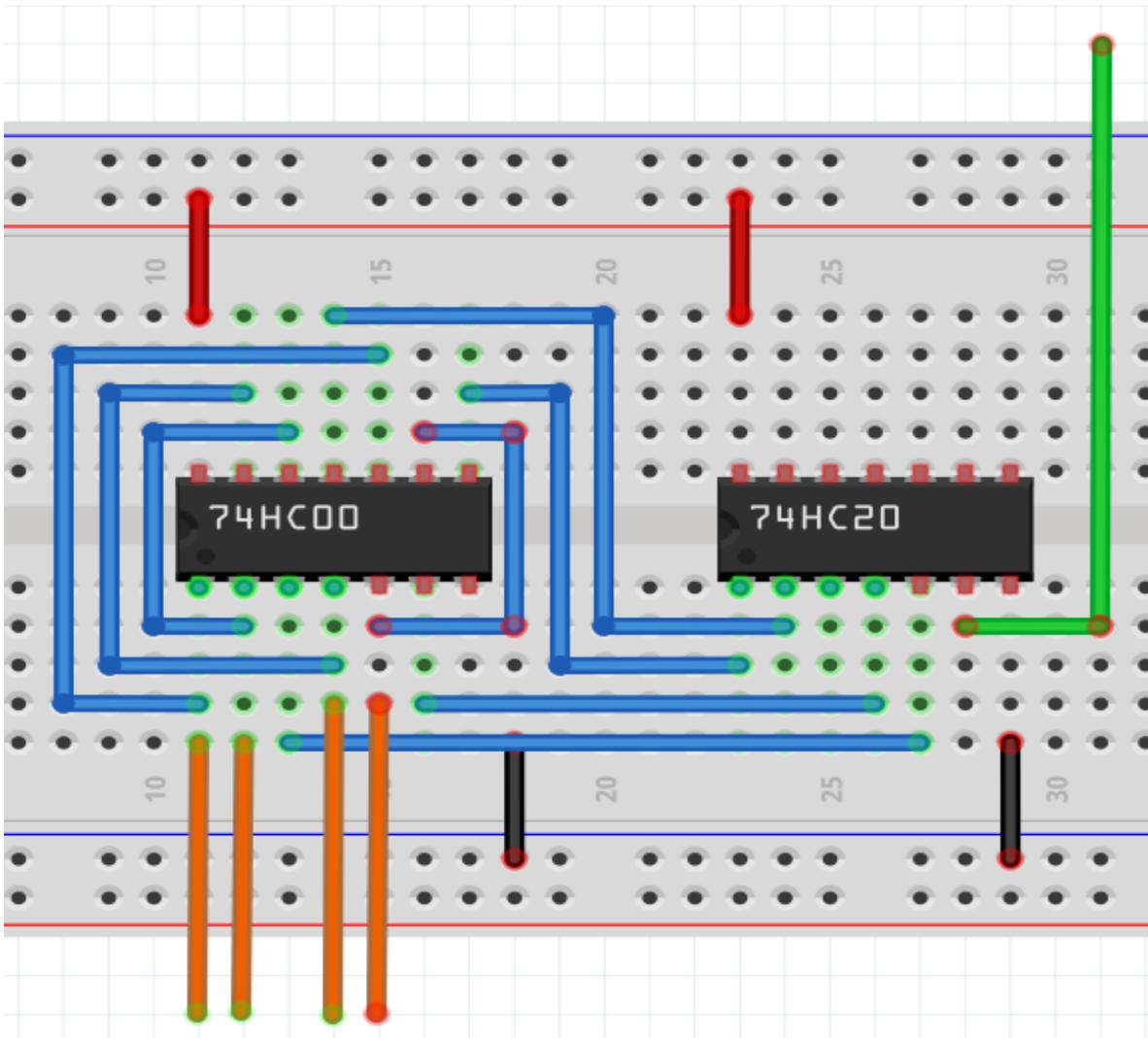


图 5 用 Fritzing 软件绘制的硬件连接图

测试方案

4 个输入信号，用实验箱上的逻辑电平开关实现，1 个输出端连接到实验箱上的 LED，按照真值表的要求，拨动逻辑电平开关改变输入信号值，遍历 16 种输入组合，并观察输出信号值，输出 LED 亮则输出为 1，灭则输出为 0，将测试结果填入表 3。

G1	G0	R1	R0	S	测试结果	G1	G0	R1	R0	S	测试结果
0	0	0	0	1		1	0	0	0	0	
0	0	0	1	1		1	0	0	1	0	
0	0	1	0	1		1	0	1	0	1	
0	0	1	1	1		1	0	1	1	1	
0	1	0	0	0		1	1	0	0	0	
0	1	0	1	1		1	1	0	1	0	
0	1	1	0	0		1	1	1	0	0	
0	1	1	1	1		1	1	1	1	1	

表 3 实验结果测试记录表

电路搭接调试方法样例

我们以教材 1.8 节电路为例,来介绍如何正确搭接和调试电路,实验电路如图 1 所示:

分析电路

分析可知,图中包含了 3 个 2 输入与非门、1 个非门、1 个 3 输入与非门,查询教材附录“常用集成电路型号和引脚图”,可知需要用到以下器件

- ◇ 7400 2 输入与非门
- ◇ 7404 非门
- ◇ 7420 4 输入与非门

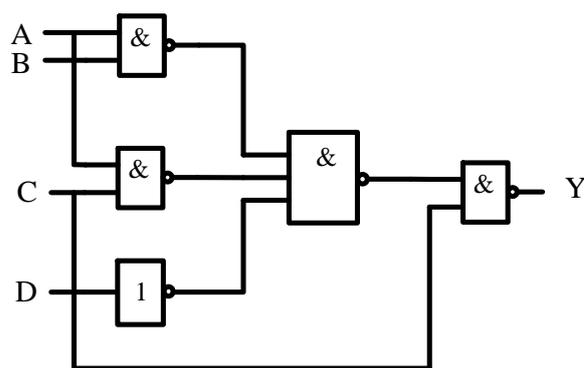
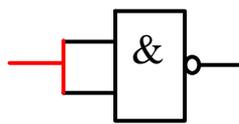


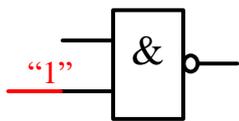
图 1 实验电路原理图

合并器件

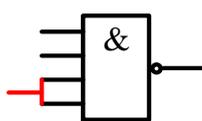
为了减少所用器件种类,可以根据门电路特点,合并一些门,其中非门可以用 2 输入与非门代替,有以下两种方法,如图 2 所示。因为没有 3 输入与非门,可以用 4 输入与非门代替 3 输入与非门,常用的也有两种方法,如图 3 所示,经过合并后的电路如图 4 所示



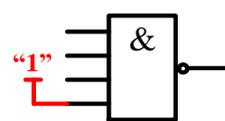
(a) 两输入脚短接



(b) 一个脚强制接高电平



(a) 两输入脚短接



(b) 空闲脚强制接高电平

图 2 非门的常用替换方法

图 3 3 输入与非门常用替换方法

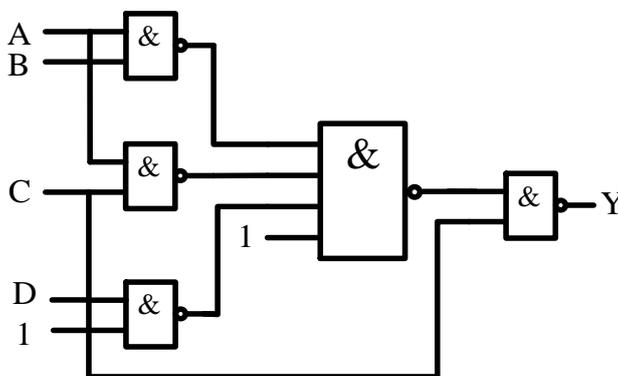


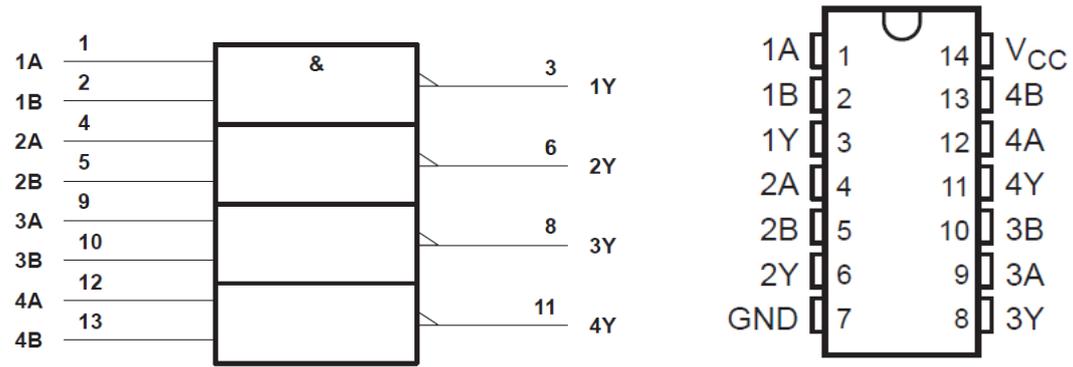
图 4 合并后的电路原理图

查询数据手册

集成电路的数据手册给出了芯片的所有参数指标以及使用方法,熟练阅读数据手册是一个电子设计人员的必备技能。数据手册的获取途径包括:

- ◇ 芯片制造商的官方网站
- ◇ 专门的 Datasheet 下载网,例如 <http://www.alldatasheet.com/> 和 <http://www.icpdf.com/>
- ◇ 直接用网络搜索引擎,如 Google、Baidu 等
- ◇ 为了提高搜索准确度,最好在搜索的关键词后面加上 pdf,如搜索“74HC00”的数据手册,可用关键词“74HC00 pdf”

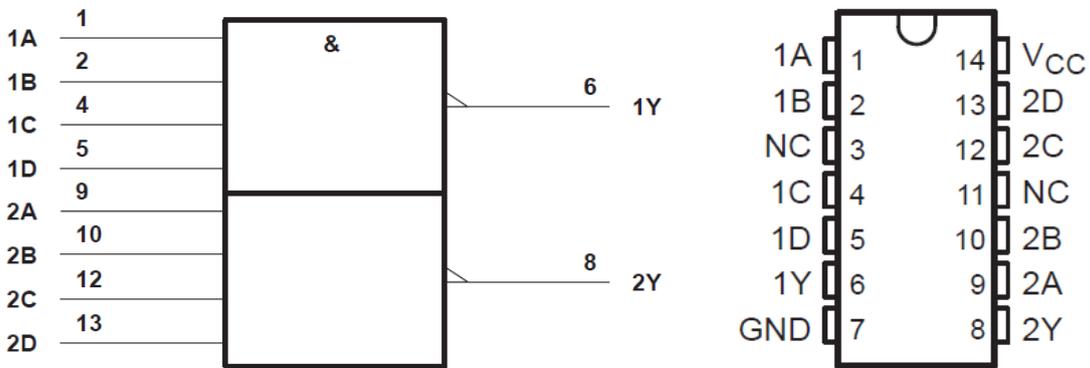
通过查询数据手册，可知 74HC00 和 74HC20 的元件符号和管脚封装如图 5 和图 6 所示。



(a) 元件符号

(b) 管脚封装

图 5 74HC00 4-2 输入与非门



(a) 元件符号

(b) 管脚封装 (NC 表示无连接)

图 5 74HC20 2-4 输入与非门

根据数据手册选择器件，分配管脚

根据数据手册的器件功能和管脚排列，合理分配逻辑门和管脚，然后在原理图上加上文字说明及数字标号，作为实验接线的依据。如图 6 所示。U1 代表 7400，U2 代表 7420。在每个门的输入、输出端标注所用的器件的引脚号。

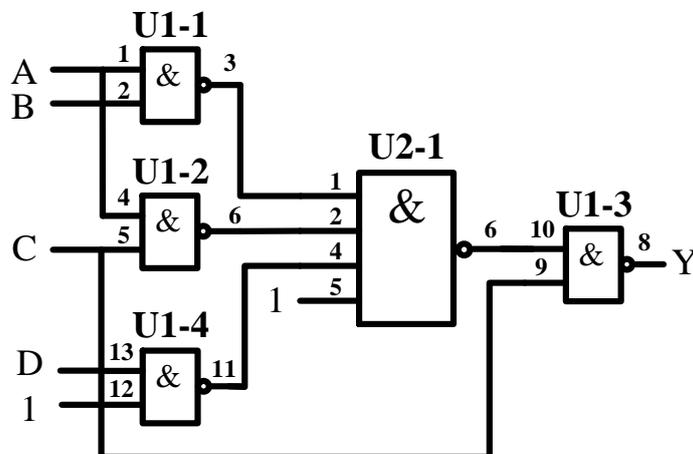


图 6 已标注管脚的电路原理图

元件布局

元器件布局要合理，信号流清晰，连线便捷，集成电路跨插在宽条面包板的凹槽上，两排引脚分别插在凹槽上下两侧的小孔中。插入集成电路之前先看清方向，一般使芯片缺口在左边，芯片第一脚在左下方。插入前微调管脚的角度，使管脚和集成电路本体之间呈近 90 度，所有管脚对准面包板对应孔位，然后均匀用力按下，确保芯片引脚能与面包板内的金属条可靠接触。如图 7 所示。

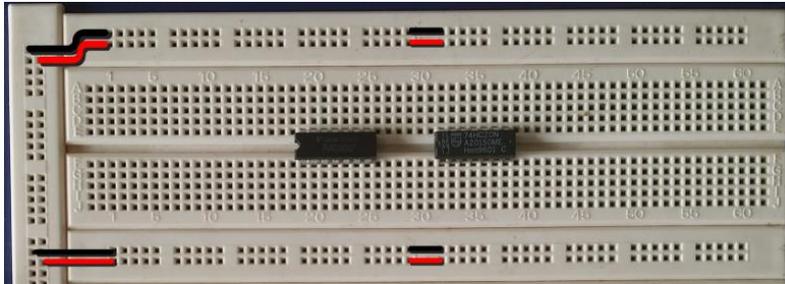


图 7 元件布局

布线

布线时使用直径为 0.6mm 左右的单股导线，使用前要用剥线钳剥离导线两侧线头，线头剥离长度约为 6mm 左右，线头剪成 45° 斜口。布线要求走线整齐，清楚，切忌混乱，电源、地和信号尽量使用不同颜色的导线，以示区别。导线应在集成电路芯片周围走线，切忌在集成块上方悬空跨过避免导线之间的互相交叉重叠，也不要过多地遮盖其他插孔，所有走线尽可能贴近面包板表面

在合理布线的前提下，导线尽可能短些布线顺序一般为先布电源线和地线，再布固定电平的规则线，最后按照信号流程逐级连接各逻辑控制线。

- 1、**布电源线**：7400 和 7420 的 14 脚是电源脚，用导线将 14 脚连接 5V 电源，如图 8 中红线所示

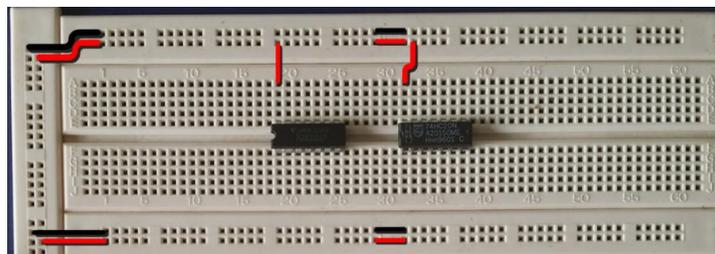


图 8 布电源线

- 2、**布地线**：7400 和 7420 的 7 脚是地线脚，用导线将 7 脚连接电源地线，如图 9 中黑线所示

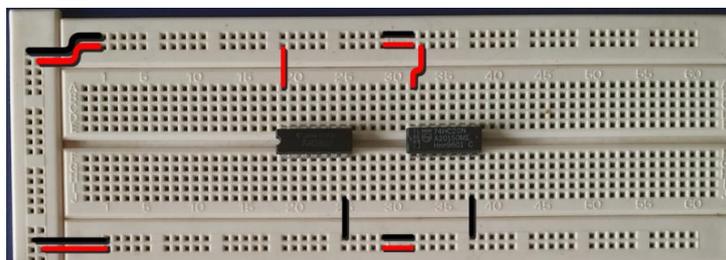


图 9 布地线

- 3、**布固定电平线**：在设计中有两个固定电平管脚，分别为 U1 的 12 脚和 U2 的 5 脚，要求置“1”，可将这些管脚直接和 5V 电源相连，如图 9 所示。如要求是置“0”，则可将这些管脚和电源地相连。

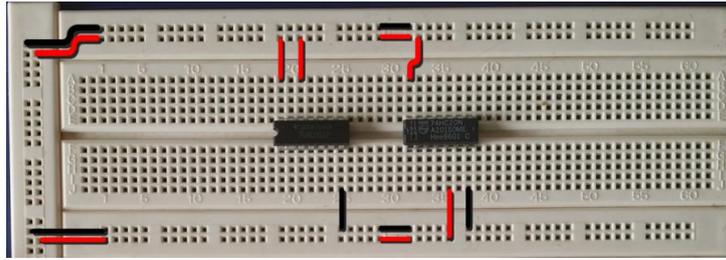


图 9 布地线

- 4、布信号线：按信号流向布线，先布最左侧的线，如图 10 所示所示；再布中间的线，如图 11 所示，最后布最右侧的线，如图 12 所示

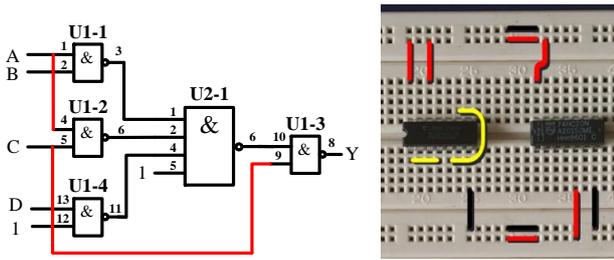


图 10 布最左侧信号线

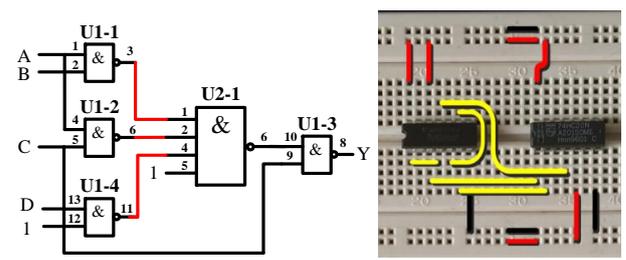


图 11 布中间信号线

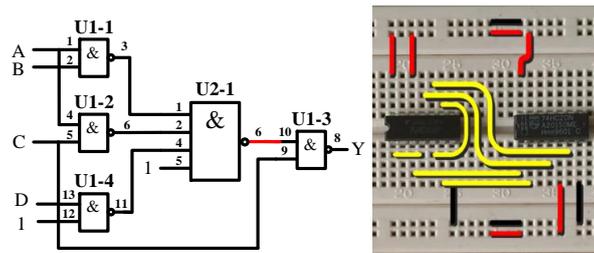


图 12 布最右侧信号线

- 5、连接输入信号，将输入信号 A、B、C、D 对应管脚，用一头直一头弯的丁字线，与实验箱上的逻辑电平开关连接，如图 13 所示

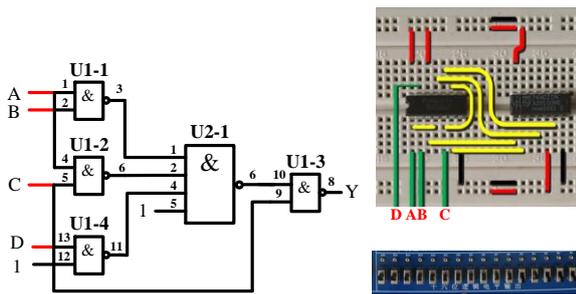


图 13 连接输入信号

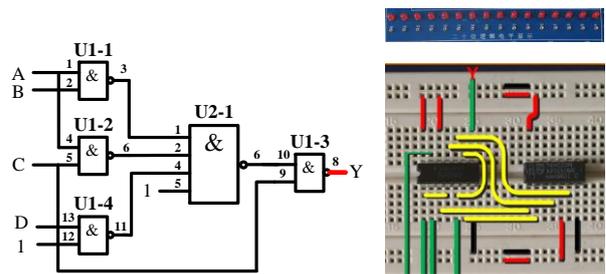


图 14 连接输出信号

- 6、连接输出信号，将输出 Y 对应管脚，同样用丁字线，与实验箱上发光二极管连接，如图 14 所示

通电前检查

在通电验证前，必须仔细检查，主要包括

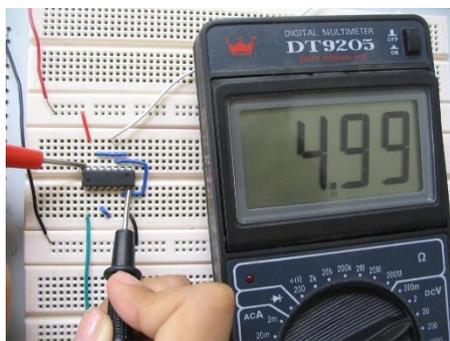
- ✧ 集成芯片是否选择正确，方向是否插对，引脚有无折弯、互碰情况
- ✧ 集成电路多余输入端处理是否正确，是否有两个以上输出端错误地连在一起
- ✧ 布线是否合理，是否有相碰短路现象
- ✧ 电源线与地线在内的连线是否有漏线与错线
- ✧ 用万用表“欧姆 $\times 10$ ”挡，测量实验电路的电源端与地线端之间的电阻值，排除电源与地线的短路现象
- ✧ 用万用表的“直流电压挡”测量直流稳压电源输出电压是否为所需值

通电验证

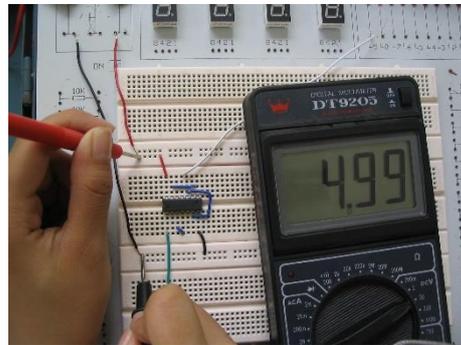
接通电源，根据真值表，按顺序，拨动逻辑电平开关，改变输入值从“0000”到“1111”，观察输出 LED 指示，LED 灭则输出“0”，LED 亮则输出“1”。记录测试结果。

故障调试

如果测试该电路时，某种输入下的输出与理论分析有悖，则要排除故障。排除故障的第一步是要确定**集成电路上所加的电源电压是否正确**，可以用万用表的直流电压档测量。可靠的检查方法是如图 15 (a) 那样，用万用表的测试表笔直接测量集成芯片电源端和地线两引脚之间的电压。这种方法可以检查出断线、引线虚接、因芯片管脚折断或折弯而未能插入实验底板的情况。图 15 (b) 中的测量方法，虽然能判断出电源是否已经接到面包板的窄条上，但如果从窄条接到集成电路管脚的连线有问题或者集成电路没有插好，集成电路还是不能正常工作。



(a) 正确的电源测量方法



(b) 错误的电源测量方法

图 15 用万用表测量电源电压

对于组合电路，可根据逻辑表达式或真值表由前向后逐级检查。但更快的检查方法，应该是由后向前逐级检查。测试方法是用万用表测量每个节点的电压。测量的时候一定要注意尽量**直接测量集成电路的管脚上的电压**，理由和前面测电源电压一样。

为了方便调试节点电压，我们在面包板上，用单股导线将黑色接线柱和绿色或黄色接线柱连接起来，如图 16 所示。这样在用万用测量节点电压的时候，可以空出一只手出来。

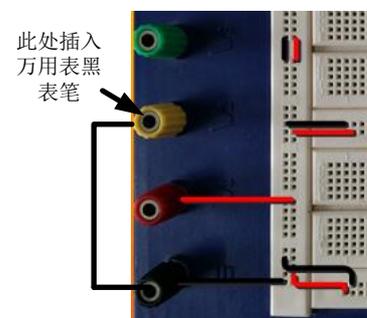


图 16 连接黑色和黄色接线柱

故障举例:

故障现象

输入逻辑开关“ABCD”置在“0011”状态,根据真值表,输出应为“0”,即逻辑电平指示灯应该灭,实际输出逻辑电平指示灯是亮,即输出为“1”,电路存在故障。

排除故障

用万用表从后向前测各点的电平并与理论值比较

1、测量最后一级 2 输入与非门 (U1-3)

理论分析:

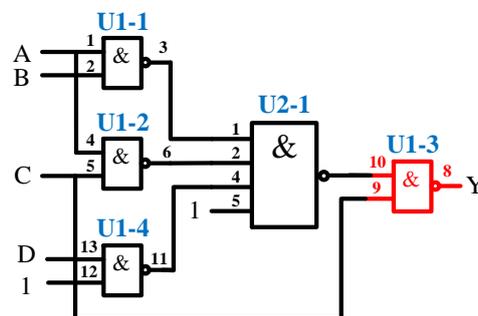
- ✧ 输出 U1-3 8 脚应为“0”
- ✧ 输入 U1-3 的 9、10 脚应为“1”

万用表实测

- ✧ U1-9 脚对地电压,约为 5V,等效逻辑“1”,信号正确
- ✧ U1-10 脚对地电压,约为 0V,等效逻辑“0”,信号错误

结论

- ✧ 因为 U1-10 脚连接到 4 输入与非门输出 (U2-1 6 脚),排查 U2-1



2、排查 4 输入与非门 U2-1

理论分析

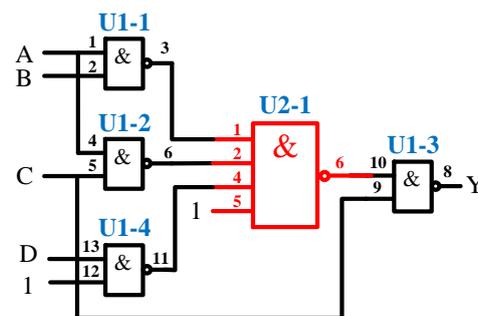
- ✧ 如 U2-6 脚为“1”,则 U2-6 脚和 U1-10 之间的连线有问题
- ✧ 如 U2-6 脚为“0”,则需检查 U2-1 的 4 个输入信号
- ✧ 根据原理图, U2-1、2、5 脚应为都“1”, 4 脚应为“0”

万用表实测

- ✧ U2-1、2、5 脚对地电压,约为 5V,等效逻辑“1”,信号正常
- ✧ U2-4 脚对地电压,约为 0~1V,等效逻辑“X”,信号错误

结论

- ✧ 因为 U2-4 脚信号连接到 2 输入与非门输出(U1-4 11 脚),需排查 U1-4



3、排查 2 输入与非门 U1-4

理论分析

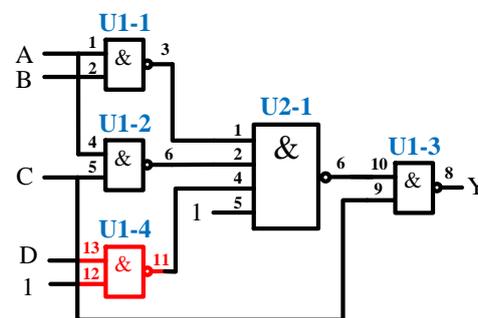
- ✧ 如 U1-11 脚为“0”,则 U1-11 脚和 U2-4 脚之间的连线有问题
- ✧ 如 U1-11 脚为“1”,则需检查 U1-4 的 2 个输入信号

万用表实测

- ✧ U1-11 脚对地电压,约为 0V,等效逻辑“0”,信号正常

结论

- ✧ U1-11 脚和 U2-4 脚之间的连线有问题



实验一 组合逻辑电路

学习目标:

- 1、认识数字集成电路，能识别各种类型的数字器件和封装
- 2、掌握小规模组合逻辑和逻辑函数的工程设计方法
- 3、学习 Multisim 逻辑化简操作和使用方法
- 4、掌握常用中规模组合逻辑器件的功能和使用方法
- 5、学习查找器件资料，通过器件手册了解器件。
- 6、了解面包板的基本结构、掌握面包板连接电路的基本方法和要求
- 7、了解实验箱的基本结构，掌握实验箱电源、逻辑开关和 LED 电平指示的用法
- 8、学习基本的数字电路的故障检查和排除方法

时间要求:

- 1、实验时间：第 6、7 周
- 2、报告提交：第 8 周课内

预备知识:

1、实验教材

本实验内容相关知识，请参看实验教材的 1.1 节、1.2 节、1.6 节、1.7 节、1.8 节、2.2 节、2.8 节、2.10 节、3.1 节、3.2 节、3.3 节。

2、利用 Multisim 软件实现自动逻辑化简

电子电路仿真软件已经深入到了电子设计的每个环节，大量传统的理论计算，都可以用仿真软件来实现。在实际工作中，数字电路的设计绝大部分都是通过计算机软件辅助实现的。Multisim 是电路仿真软件的一种，它可以对模拟电路和数字电路进行精确仿真。在这里我们介绍 Multisim 软件中的逻辑转换器功能，它可以将分析好的真值表转换成逻辑表达式和电路原理图，将我们从繁重的电路化简工作中解脱出来。

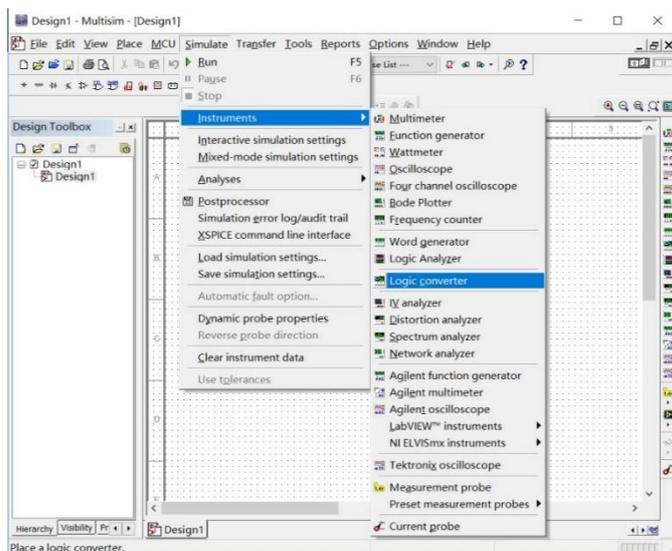


图 1.1 Multisim 界面

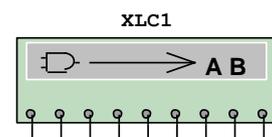


图 1.2 逻辑转换器

打开 Multisim 软件，如图 1.1 所示，在菜单栏中选择 Simulate→Instruments→Logic Converter 菜单项打开如图 1.2 所示的仪器，双击，打开图 1.3 所示的界面。以前面预习报告样例中的血型配对为例，有四个输入

变量，用鼠标点击，选中 A、B、C、D 四个输入，显示从“0000”到“1111”16 个值，在右侧用鼠标点击对相应的输出，按要求设置成 0、1 或 x。点击右侧 $\overline{1011} \rightarrow A\overline{B}$ 按钮，可以将真值表转换成最小项。点击 $\overline{1011} \rightarrow A\overline{B}$ 按钮，则可以将真值表转换成最简与或表达式。转换结果显示在逻辑转换器底部窗口中，化简结果“A'B'+A'D+B'C+CD”和用卡诺图化简的结果是一样的。点击 $A\overline{B} \rightarrow \rightarrow$ 按钮，可以自动生成与或电路如图 1.4 所示，点击 $A\overline{B} \rightarrow \text{NAND}$ 按钮，可以自动生成与非电路如图 1.5 所示。注意，自动生成的电路图，都是 2 输入门。

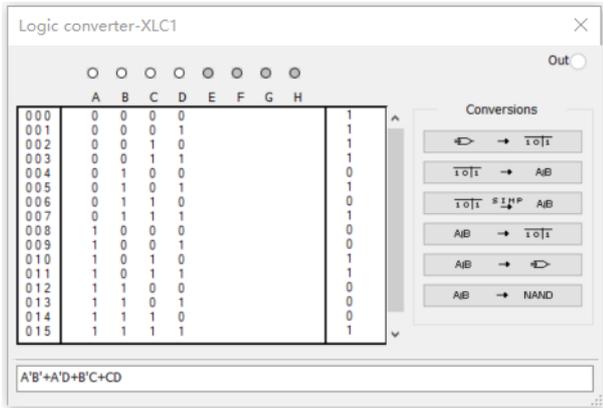


图 1.3 真值表输入界面

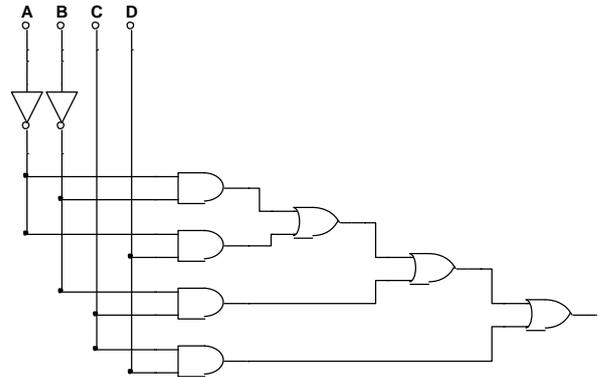


图 1.4 自动生成的与或电路图

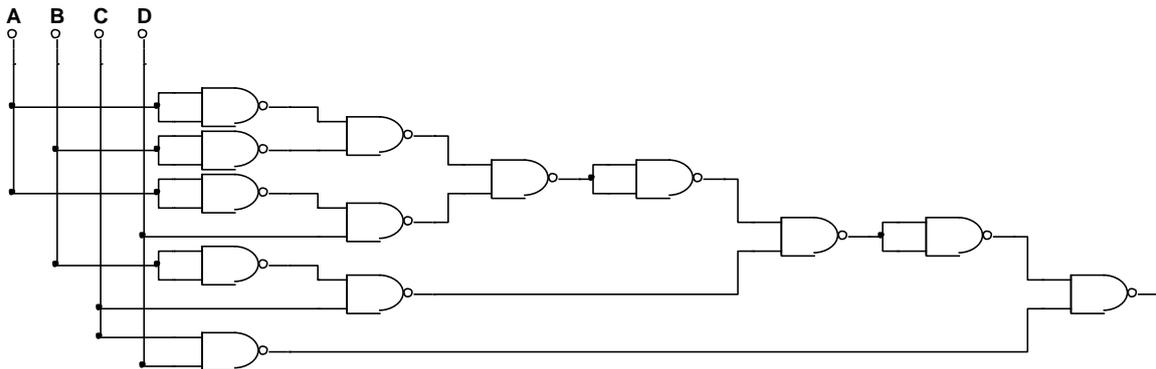


图 1.5 自动生成的与非电路图

3、参考资料：

“如何读懂数字逻辑器件数据手册” (pdf 文档)

“Multisim 软件使用手册” (pdf 文档)

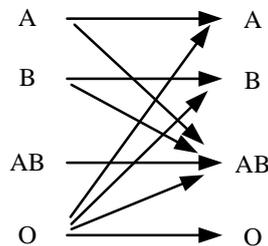
预习要求

- 1、根据“课程准备事项要求”，完成单股线的准备和面包板常用连接线的安装
- 2、访问 MOOC 平台，预习第二章和第三章与本实验相关的内容
- 3、第 6 周课前完成必做实验 1 和实验 2 电路设计，将设计方案、原理图、硬件连接示意图写在实验报告的原理部分，
- 4、完成“数值判别电路”的电路搭接，如果有余力也可预搭内容 2 电路；

必做实验

- 1、数值判别电路（第 6 周实验课内指导教师验收）

- a) 用门电路设计一个组合逻辑电路，它接收一位 8421BCD 码 $B_3B_2B_1B_0$ ，仅当 $2 < B_3B_2B_1B_0 < 7$ 时输出 Y 才为 1
- b) 用门电路设计一个组合逻辑电路，它接收 4 位 2 进制数 $B_3B_2B_1B_0$ ，仅当 $2 < B_3B_2B_1B_0 < 7$ 时输出 Y 才为 1
- 2、用三种方案设计实现 3 位二进制原码转补码电路（3 位二进制数仅考虑 0 和负数，且已省去符号位）：
- a) 全部用门电路实现
- b) 用数据选择器 74151+门电路实现
- c) 用三八译码器 74138+门电路实现
- 提示：根据本学期发的器件完成设计，异或逻辑可以直接选用 7486
- 3、人类有四种血型：A、B、AB 和 O 型。输血时，输血者与受血者必须符合下图的规定，否则有生命危险，利用数据选择器和最少数量的与非门，完成血型配对任务。（设计方案可参看数字逻辑电路实践教材 Page 86）



注：内容 2 和内容 3 为开放实验，可自己任意选择合适的时间到实验室完成实验，**每项实验结果都需要用手机拍摄视频上传到各自班级 QQ 群的 QQ 作业中**

选作实验

- 1、设计一个保险箱的数字密码锁，该锁有 4 位密码输入端 (A_0, A_1, A_2, A_3) 和一个开箱钥匙信号输入端 (E)。当用钥匙开箱时 ($E=1$)，如果输入密码 = “1011”，保险箱被打开 ($Z_1=1$)；如果不符，电路将发生报警信号 ($Z_2=1$)。要求使用最少数量的与非门实现电路，检测并记录实验结果。

实验二 可编程数字逻辑设计基础

学习目标

- 1、了解可编程数字系统设计的流程
- 2、掌握 Quartus 软件的使用方法
- 3、掌握竞争和冒险的基本概念和电路时延分析方法

时间要求

- 1、实验时间：第 8 周
- 2、报告提交：第 9 周课内

预备知识

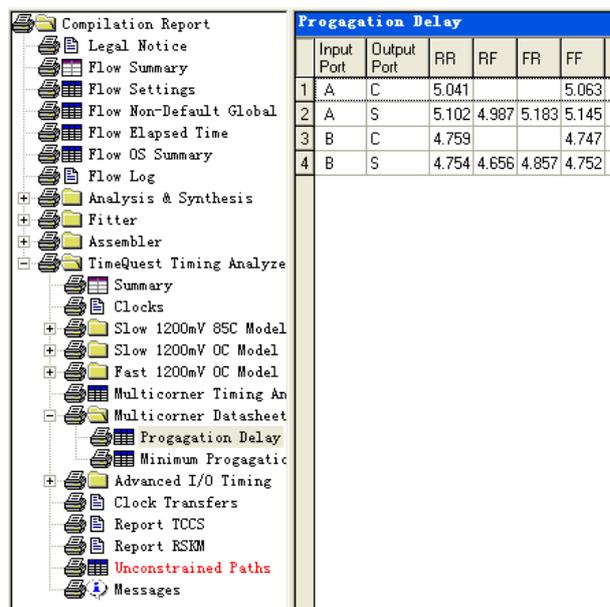
1、关于 Quartus 软件

Quartus 是 Altera（现已被 Intel 收购）公司可编程逻辑器件开发平台，根据功能的不同，可分为订购版（15.1 版本后改为专业版和标准版）和网络版（15.1 后称为精简版），可以直接在 Intel [网站下载](#)，其中网络版是免费的，只要注册即可下载使用。由于 Quartus 9.1 以后的版本取消了内置仿真器，需要用硬件描述语言写 Testbench 调用第三仿真器，不太适合本学期实验。如需要在自己电脑上安装，建议大家下载安装 Quartus 9.1 sp2 Web 版，使用方法参看附录一和教材第 7 章。

2、逻辑电路时延分析方法

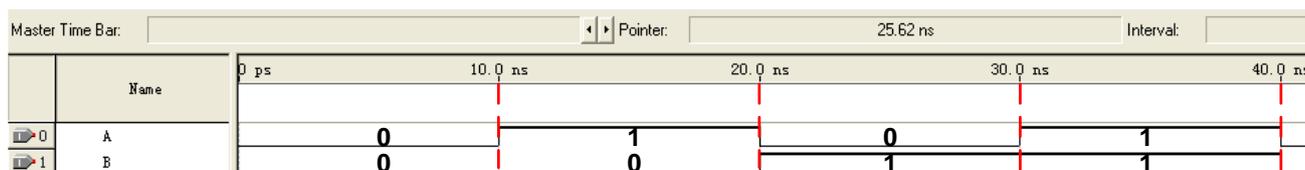
下面以 1 位半加器为例，介绍一下时序仿真和电路时延分析方法。半加器有 2 个输入 A、B，两个输出 S、C。

- a) 点击工具条  按钮，编译项目，生成 Compilation Report
- b) 在工作区右栏 Report 列表中，点击“TimeQuest Timing Analyze”→“Multicorner Datasheet”→“Proagation Delay”项，查看电路时延信息，如图 2.1 所示。
- c) 以时延信息表第 2 行为例，RR=5.102ns 表示当输入端口 A 从“0”→“1” (Rising)后 5.102ns，输出端口“S”从“0”→“1” (Rising)；
- d) RF=4.987ns 表示当输入端口 A 由“0”→“1” (Rising)后 4.987ns，输出端口“S”从“1”→“0” (Falling)；同样可以得到 FR 和 FF 的值。



	Input Port	Output Port	RR	RF	FR	FF
1	A	C	5.041	5.063		
2	A	S	5.102	4.987	5.183	5.145
3	B	C	4.759			4.747
4	B	S	4.754	4.656	4.857	4.752

图 2.1 电路时延信息



e) 图 2.2 输入激励信号

- f) 新建“Vector Waveform File”设置输入激励文件，为了仿真全部功能，同时为了更方便的查看时延，

输入激励文件中信号 A 可选用周期为 20ns 的时钟信号，B 选用周期为 40ns 的时钟信号，如图所示。

- g) 选择“Processing”→“Simulator Tool”打开仿真工具界面，选择 Timing 仿真，打开仿真结果，右键点击波形区域，选择“Time Bar”→“Insert Time Bar”或者“Time Bar”→“Time Bar Organizer”增加两条“Time Bar”。如图所示。

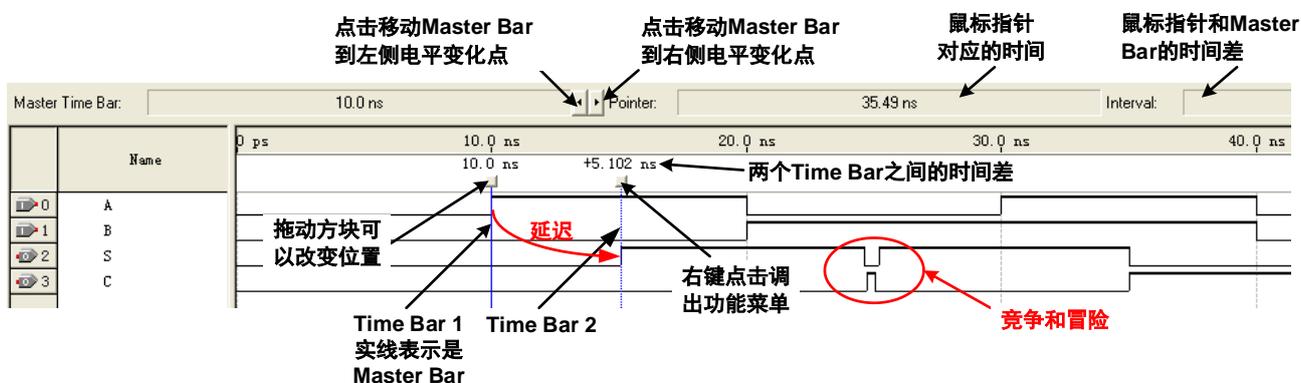
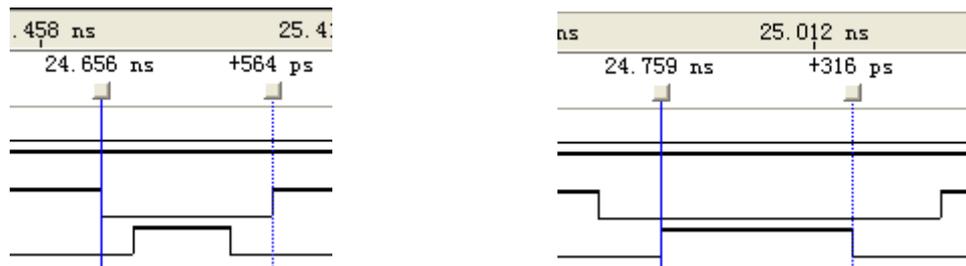


图 2.3 仿真波形时延分析

- h) 从图中可以看到，输入信号 A 从“0”→“1”，输出信号 S 并不立刻变化，有一定的延迟。将两条“Time Bar”轮流设为“Master Time Bar”，点击 ，使 Time Bar 1 卡在输入 A 信号的第一个上升沿，Time Bar 2 卡在输出 S 信号的第一个上升沿，可以测出，输入 A 从“0”→“1”后 5.102ns 输出 S 从“0”→“1”，这和前面查看时延信息表得到的结果一致。
- i) 从图中可以看到，当输入 BA 从“01”→“10”时，输出信号 S 出现了一个很短的低电平，输出 C 则出现了一个很短的高电平，这表明出现了竞争和冒险现象。BA 从“01”→“10”是两个信号同时变化，而从时延信息表可知 A→S 的 FR=5.183ns，B→S 的 RF=4.656ns，也就是说，由于 B 信号变化的快，B 因此“0”→“1”，A 还未从“1”→“0”，短时间输入端出现了“11”状态，根据逻辑函数，此时 S 输出 0，C 输出 1。等到 A 从“1”→“0”后，输入稳定为“10”，根据逻辑函数，S 输出 1，C 输出 0。用 g 中的方法我们可以测出 S 和 C 的毛刺宽度如图所示。



3、参考资料

- “Cyclone II 器件数据手册” (pdf 文档)
 “Cyclone III 器件数据手册” (pdf 文档)

预习要求

- 1、访问 MOOC 平台第二章“竞争和冒险现象”和第五章，预习和本实验相关的内容
- 2、学习实验教材 2.9 节“竞争和冒险现象”；
- 3、学习实验教材第 7 章“可编程数字系统设计基础”；
- 4、参考附录“Quartus II 使用简介”，第 8 周课前完成必做实验的设计方案，用 Quartus 原理图输入法完成必做内容的原理图绘制，并将设计文件拷贝带至实验室，上课前交由指导教师检查

必做实验

基础部分 (70%)

- 1、观察并记录实验箱上的 FPGA 型号，新建一个 Project，器件选用实验箱上的 FPGA；
- 2、用“AND2”和“XOR”器件设计一个 1 位半加器，并用功能仿真进行验证；
- 3、点击“File”→“Create/Update”→“Create Symbol File for Current File”菜单项，将 1 位半加器封装成元件。新建一个原理图文件，调用 2 个半加器实现一个 1 位全加器，并用功能仿真进行验证；
- 4、将 1 位全加器封装成元件，新建原理图文件，调用 4 个全加器实现一个 4 位行波加法器，用功能仿真进行验证，用“Tools”→“Netlist Viewers”→“RTL Viewer”查看电路综合结果；
- 5、适配编译，用“Tools”→“Netlist Viewers”→“Technology Map Viewer”查看电路 Map 结果；用“Tools”→“Chip Planner”查看器件适配结果；
- 6、将 4 位全加器下载到实验箱，连接逻辑电平开关进行功能验证（课内验收）；
- 7、对 1 位半加器，进行时序仿真，并做时延分析，可参考预备知识相关内容；
- 8、对 1 位全加器，进行时序仿真，并做时延分析，要求：
 - a) 测量 A 第 1~4 个上升沿到对应的 S 输出之间的延迟时间；
 - b) 对输出 S 的毛刺进行测量和分析；
 - c) 对输出 C 的毛刺进行测量和分析；
 - d) 对测得的时延结果进行分析
- 9、对 4 位全加器，进行时序仿真，并做时延分析，要求：
 - a) 测量 $C_{in}=“0”$ ， $B=“0111”$ ，A 从“0000”→“0001”所对应输出 S_3 的时延；
 - b) 测量 $C_{in}=“1”$ ， $B=“1000”$ ，A 从“0000”→“0111”所对应输出 C_{out} 的时延；
 - c) 对测得的时延结果进行分析

扩展部分 (30%)

- 1、参看 7.6 节“总线 (Bus) 功能”，将 4 位全加器的输入输出信号用总线的方式引出，并封装成元件；
- 2、利用封装好的 4 位全加器，实现 4 位全减器。其中输入为 4 位二进制无符号数，输出结果为 5 位二进制原码，其中最高位为符号位。电路算法要求采用二进制求补相加方式。
- 3、将 4 位全减器下载到实验箱，连接逻辑电平开关进行功能验证（课内验收，课内来不及验收的可提交实验结果视频到各自班级 QQ 群的 QQ 作业中）；

选做实验

- 1、调用 4 个 4 位全加器模块，实现 1 个 16 位全加器，做时序仿真，测量 $C_{in}=“1”$ ， $B=(8000)_{16}$ ，A 从 $(0000)_{16} \rightarrow (7FFF)_{16}$ 所对应输出 C_{out} 的时延；
- 2、查找资料，修改 16 位全加器的结构，选择更合理的方案，降低延迟，提高工作频率。

实验三 时序逻辑电路

学习目标

- 1、掌握时序逻辑电路的一般设计过程
- 2、掌握时序逻辑电路的时延分析方法，了解时序电路对时钟信号相关参数的基本要求
- 3、掌握时序逻辑电路的基本调试方法
- 4、熟练使用示波器和逻辑分析仪观察波形图

时间要求

- 1、实验时间：第 9、10、11 周
- 2、报告提交：第 12 周

预备知识

实验教材： 第 4 章

预习要求

- 1、访问 MOOC 平台第四章，预习和本实验相关的内容

2、广告流水灯（第 9 周）

- a) 用 D 触发器 7474 分别设计一个模 8 异步行波计数器和模 8 同步计数器，电路包含一个输出信号 F，当计数器计数值为“7”的时候， $F = 1$ ，其他计数值则 $F = 0$ 。在 Quartus 中进行时序仿真验证，并对两个仿真结果进行比较和分析
- b) 完成广告流水灯的设计，包含详细的设计过程和电路原理图
- c) 完成广告流水灯的硬件电路搭接

3、序列发生器（第 10 周）

- a) 分别用集成计数器 74161 的同步置“0”和异步清“0”功能实现模 10 计数器，在 Quartus 中进行时序仿真验证，并分析比较两种方法的区别
- b) 分别用集成计数器 74161 和 4 位双向移位寄存器 74194 实现图 3.1 所示的环形计数器，电路必须能自启动，并在 Quartus 中进行时序仿真验证
- c) 完成两种方法实现序列发生器的设计方案，包含详细的设计过程和电路原理图
- d) 完成两种方案序列发生器的硬件电路搭接

4、4 位并行输入-串行输出曼切斯特编码电路（第 11 周）

- a) 完成 4 位并行输入-串行输出曼切斯特编码设计方案、包含详细的设计过程和电路原理图
- b) 自行设计合理的电路验证方案
- c) 完成 4 位并行输入-串行输出曼切斯特编码硬件电路搭接

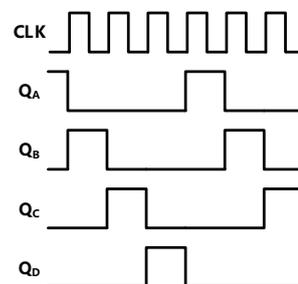


图 3.1 环形计数器波形

必做实验

1、广告流水灯（第 9 周课内验收）

用触发器、组合函数器件和门电路设计一个广告流水灯，该流水灯由 8 个 LED 组成，工作时始终为 1 暗 7 亮，且这一个暗灯循环右移。

- (1) 写出设计过程，画出设计的逻辑电路图，按图搭接电路
- (2) 将单脉冲加到系统时钟端，静态验证实验电路
- (3) 将 TTL 连续脉冲信号加到系统时钟端，用示波器观察并记录时钟脉冲 CP、触发器的输出端 Q2、Q1、Q0 和 8 个 LED 上的波形。

2、序列发生器（第 10 周课内验收）

分别用 MSI 计数器和移位寄存器设计一个具有自启动功能的 01011 序列信号发生器

- (1) 写出设计过程，画出电路逻辑图
- (2) 搭接电路，并用单脉冲静态验证实验结果
- (3) 加入 TTL 连续脉冲，用示波器观察观察并记录时钟脉冲 CLK、序列输出端的波形。

注：内容 2 为开放实验，可自己任意选择合适的时间到实验室完成实验，每项实验结果都需要用手机拍摄视频上传到各自班级 QQ 群的 QQ 作业中。

3、4 位并行输入-串行输出曼切斯特编码电路（第 11 周课内验收，基础部分占 70%，扩展部分占 30%）

在电信与数据存储中，曼彻斯特编码(Manchester coding)，又称自同步码、相位编码(phase encoding, PE)，它能够用信号的变化来保持发送设备和接收设备之间的同步，在以太网中，被物理层使用来编码一个同步位流的时钟和数据。曼彻斯特编码用电压的变化来分辨 0 和 1，从高电平到低电平的跳变代表 0，而从低电平到高电平的跳变代表 1。信号的保持不会超过一个比特位的时间间隔。即使是 0 或 1 的序列，信号也将每个时间间隔的中间发生跳变。这种跳变将允许接收设备的时钟与发送设备的时钟保持一致，图 3.2 为曼切斯特编码的例子。

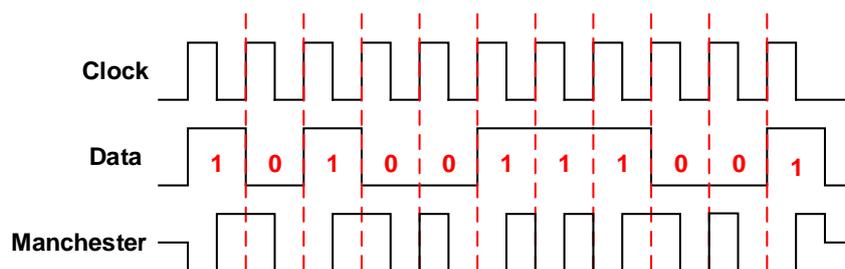


图 3.2 曼切斯特编码

设计一个电路，它能自动加载 4 位并行数据，并将这 4 位数据逐个串行输出（高位在前），每个串行输出位都被编码成曼切斯特码，当 4 位数据全部传输完成后，重新加载新数据，继续传输，如图 3.3 所示。

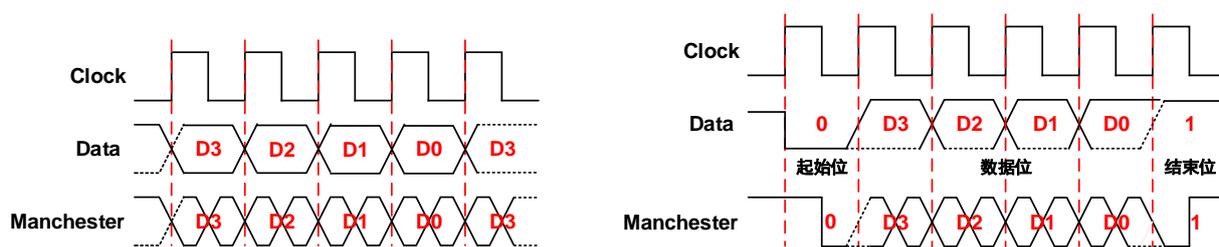


图 3.3 基础部分波形

3.4 扩展功能波形

- (1) 写出设计过程，画出电路逻辑图，设计时要注意**不允许手动加载数据**。
- (2) 加入 TTL 连续脉冲，用示波器观察观察并记录时钟脉冲 CLK、串行数据输出端的波形。
- (3) 给串行数据增加起始位和结束位，其中起始位为“0”，结束位为“1”，起始和结束位同样要编码成曼切斯特码，波形图参看图 3.4（**扩展部分，选作**）

实验四 FPGA 时序逻辑设计

学习目标

- 1、进一步学习时序逻辑电路的分析和设计方法，掌握状态机设计方法，
- 2、了解可编程数字系统设计的流程，掌握 Quartus II 软件的使用方法
- 3、掌握原理图输入方式设计时序逻辑电路的方法和流程

时间要求

- 1、实验时间：第 12 周
- 2、报告提交：第 13 周

预备知识

1、实验教材：

第 4 章、第 7 章

2、状态机（更详细的内容可以参看教材 4.1.5、4.5.2 第 5 部分、6.2.4）

时序逻辑电路又称为有限状态机（FSM, Finite State Machine）。根据电路的输出变量、输入变量及状态变量（触发器输出）之间的关系，有限状态机可分为米里型（Mealy）状态机和摩尔型（Moore）状态机两大类。摩尔型状态机的输出只与触发器的原输出状态有关，而米里型状态机的输出既与触发器的原输出状态有关，还与输入变量的状态有关。

状态机的特性取决于输入、输出和触发器的状态。触发器的状态有**现态**和**次态**两种，输入信号作用前或下一个时钟边沿到来之前的触发器状态称为现态，用 Q^n 和 \overline{Q}^n 表示；输入信号作用后或下一个时钟边沿到来之后的触发器状态称为次态，用 Q^{n+1} 和 \overline{Q}^{n+1} 表示。

输入、输出和触发器的状态三者之间的关系可用状态方程、状态图、状态表和时序图来表示。四种描述方法从不同侧面突出了时序电路逻辑功能的特点，它们本质上是相通的，可以互相转换，在实际工作中，可根据具体情况选用。下面我们以一个“1110”序列检测器为例，了解一下状态机的相关概念

序列检测器用于检测一组或多组由二进制码组成的脉冲序列信号，当检测器连续收到一组串行二进制码，且这组码与检测器中预先设置的码相同，则输出 1，否则输出 0。序列检测的关键在于正确码的收到必须是连续的，这就要求检测器必须记住前面已测到的正确码，直到在连续的检测中所收到的每一位码都与预置码相同。在检测过程中，任何一位不相等都将回到初始状态重新开始检测。分析可得“1110”序列检测器的判断流程如图 4.1 所示。根据工作流程分析绘制状态转移图如图 4.2 所示，其中 X 为输入信号，Z 为输出信号。

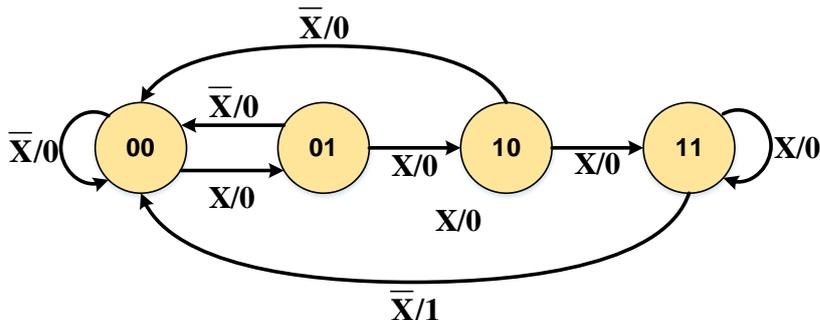


图 4.2 “1110” 序列检测器状态转移图

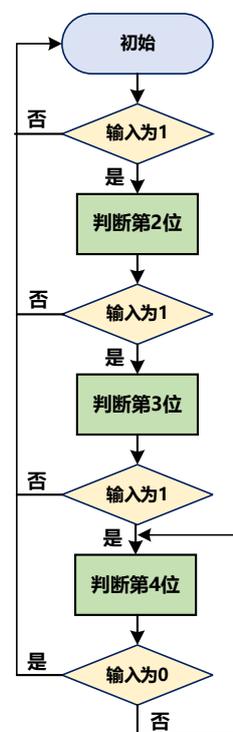


图 4.1 序列检测器流程图

状态机设计时需要状态进行编码，常用的状态编码有二进制码、格雷码以及 One Hot 码。

二进制码是最常用的一种状态分配法，其状态是用触发器状态的不同组合来表示的， N 个触发器有 2^N 种状态组合，状态数 M 和触发器的级数 N 一般要满足式 $2^{N-1} < M \leq 2^N$ ；然后给每个状态规定对应的触发器状态组合。从 2^N 个状态中取出 M 个状态的组合可以有多种不同的分配方案，而每个方案中 M 个状态的排列顺序又有许多种。不同的分配方案，所设计出来的电路是不一样的，复杂程度也不一样。如果分配方案选的好，设计出来的电路就很简单，否则电路就会复杂得多。

格雷码是二进制码的一个特例，由于格雷码是一种安全编码，当状态顺序改变时，二进制数值中只有一位发生变化，不易产生竞争和冒险，有利于提高系统的稳定性。

One-Hot 码，又称为一位有效编码，采用 N 个触发器来对 N 个状态进行编码，每个状态都由独立的触发器确定，并且在任意时候只有一位有效。**One-Hot** 编码使逻辑实现更简洁，因为一个状态只需要用一位来指示，该位为 1 对应唯一的状态，判断当前状态是否为该状态，只需判断该状态位是否为 1；如果状态寄存器输入端该位为 1，则下一状态将转移为该状态，因此判断下一状态是否为该状态，只需判断表示下一状态的信号中该位是否为 1 即可。

对于“1110”序列检测器，选择不同方式进行状态编码的结果如表 4.1 所示，对应的状态转移表如表 4.2 所示。为了便于后面用计数器实现状态机，二进制编码根据状态顺序进行编码。

表 4.1 “1110”序列检测器状态编码表

状态名	二进制码		格雷码		One-Hot 码			
	Q ₁	Q ₀	Q ₁	Q ₀	Q _A	Q _B	Q _C	Q _D
S ₀	0	0	0	0	1	0	0	0
S ₁	0	1	0	1	0	1	0	0
S ₂	1	0	1	1	0	0	1	0
S ₃	1	1	1	0	0	0	0	1

表 4.2 “1110”序列检测器状态转移表

文字描述				二进制编码				One-Hot 编码							
现态	次态	条件	输出	现态		次态		现态				次态			
				Q ₁	Q ₀	Q ₁	Q ₀	Q _A	Q _B	Q _C	Q _D	Q _A	Q _B	Q _C	Q _D
S ₀	S ₀	\bar{X}	0	0	0	0	0	1	0	0	0	1	0	0	0
	S ₁	X	0												
S ₁	S ₀	\bar{X}	0	0	1	0	0	0	1	0	0	1	0	0	0
	S ₂	X	0												
S ₂	S ₀	\bar{X}	0	1	0	0	0	0	0	1	0	1	0	0	0
	S ₃	X	0												
S ₃	S ₀	\bar{X}	1	1	1	0	0	0	0	0	1	1	0	0	0
	S ₃	X	0												

无论是二进制编码还是 One-Hot 编码，在状态机实现的时候一般都可选择触发器、集成计数器或移位寄存器方案。

对于 **One-Hot 编码状态机**，当用 D 触发器实现时，需要求出各触发器的激励方程或状态方程，其每一个次态都是现态与上各个转移条件之或，即：

$$NS = \Sigma(C \cdot PS)$$

其中，PS 是箭头的根状态（现态），NS 是箭头的目的状态（次态），C 是转移条件。对于图 4.2 有：

$$S_0^n = \bar{X} \cdot S_0^{n-1} + \bar{X} \cdot S_1^{n-1} + \bar{X} \cdot S_2^{n-1} + \bar{X} \cdot S_3^{n-1}$$

由于 One-Hot 编码状态和触发器是一一对应的，所以可得

$$Q_A^n = \bar{X} \cdot Q_A^{n-1} + \bar{X} \cdot Q_B^{n-1} + \bar{X} \cdot Q_C^{n-1} + \bar{X} \cdot Q_D^{n-1}$$

同理：

$$Q_B^n = X \cdot Q_A^{n-1} \quad Q_C^n = X \cdot Q_B^{n-1} \quad Q_D^n = X \cdot Q_C^{n-1} + X \cdot Q_D^{n-1}$$

输出 Z 为：

$$Z = \bar{X} \cdot Q_D^{n-1}$$

对于带使能端的 1-2 译码器来说，其逻辑函数为：

$$D_1 = A_0 EN \quad D_0 = \bar{A}_0 EN$$

所以 1 个转移条件变量和现态相与可以用带使能端的 1-2 译码器来实现，同样如果有 2 个转移条件则可用带使能端的 2-4 译码器来实现，3 个转移条件则可用带使能端的 3-8 译码器来实现，以此类推。由此绘制出电路原理示意图如图 4.4 所示。最后用 74 系列器件实现这个状态机，因为 74 系列没有专门的 1-2 译码器，可以选择了 74139（2-4 译码器）来代替，具体电路图如图 4.5 所示。一定要注意 One-Hot 状态机要增加自启动电路，74139 的输出是低电平有效，另外 Z 是寄存输出。

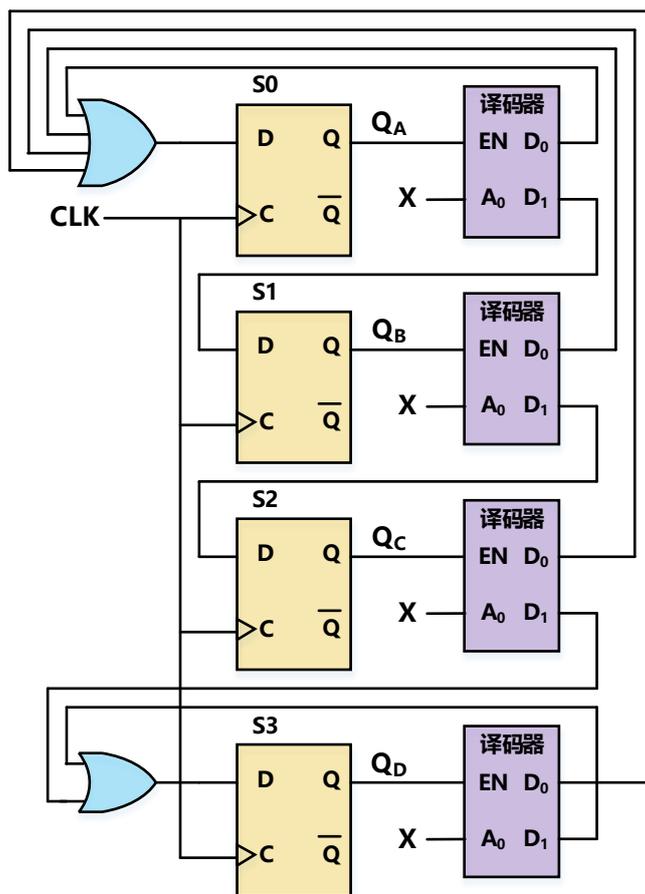


图 4.4 “1110” 序列检测器电路原理示意图

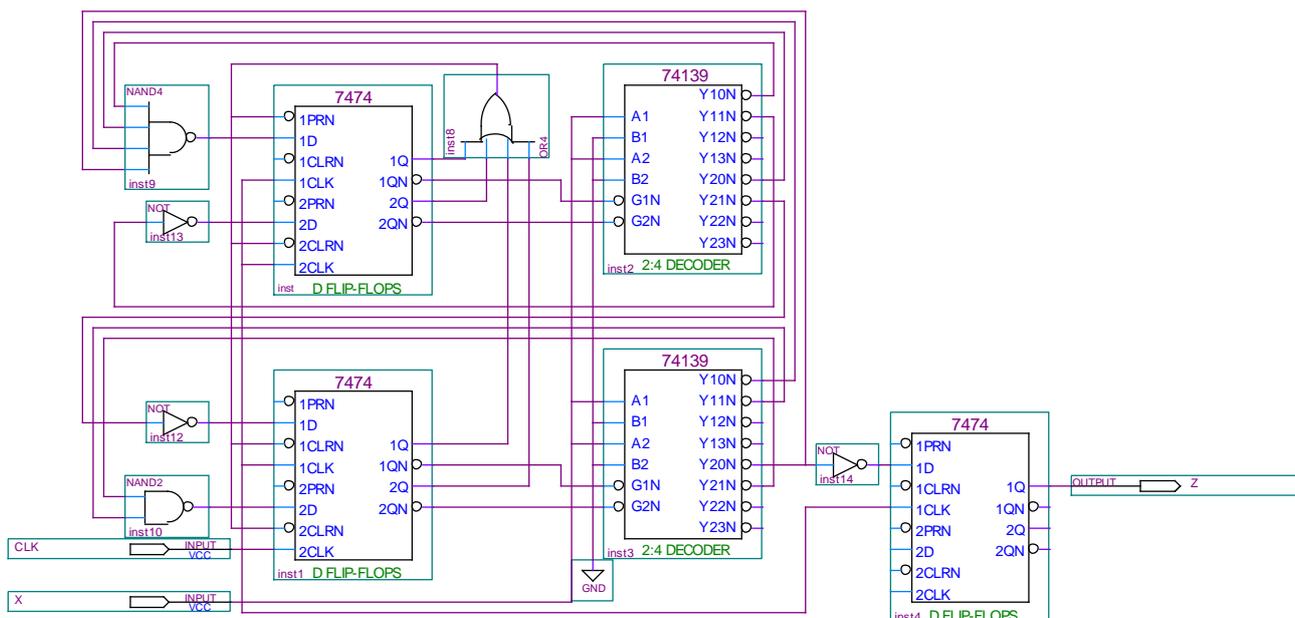


图 4.5 One-Hot 编码“1110”序列检测器电路原理图

用集成计数器法实现 One-Hot 编码状态机时，一般会用译码器将二进制计数值转成 One-Hot 码。而用移位寄存器法实现 One-Hot 编码状态机时，则只要将移位寄存器设计成具有自启动功能的环形计数器就能产生 One-Hot 码了（可参考实验三的相关预习题）。

而对于二进制编码状态机，当用 D 触发器实现时，因为 Q_1 、 Q_0 都不止一个状态为 1，不能象 One-Hot 编码那样直接将触发器代入，但可以对照表 4.2，将 Q_1 、 Q_0 对应为“1”的转移条件的 C 和 PS 的积相加获得状态方程，结果如下：

$$Q_0^n = X \cdot \overline{Q_1^{n-1}} \cdot \overline{Q_0^{n-1}} + X \cdot Q_1^{n-1} \cdot \overline{Q_0^{n-1}} + X \cdot Q_1^{n-1} \cdot Q_0^{n-1}$$

$$Q_1^n = X \cdot \overline{Q_1^{n-1}} \cdot Q_0^{n-1} + X \cdot Q_1^{n-1} \cdot \overline{Q_0^{n-1}}$$

用集成计数器功能块实现时，其设计过程相比用触发器实现要复杂些。从状态图可以看到，在二进制编码的情况下，其各个状态间的迁移有两种情况，一为顺序转移，即与计数器自身规律相一致的转移，如 $S_0 \rightarrow S_1$ ， $S_1 \rightarrow S_2$ 等；另一个是非顺序转移，如 $S_1 \rightarrow S_0$ ， $S_3 \rightarrow S_1$ 等。还有一种保持的情况，如 S_0 在输入 $X=0$ 时的行为。显然可以考虑用计数器的计数、保持和置数等功能来实现这些控制关系。下面我们用集成计数器 74161 来完成具体的设计：

先将状态图划分为顺序转移和非顺序转移。对于 74161 计数器，使能端 ENT 和 ENP 都为“1”时，顺序计数，从而实现顺序转移。使能端 ENT 和 ENP 都为“0”时，保持当前状态，故保持状态也可以画在顺序迁移图上，如图 4.6 所示。

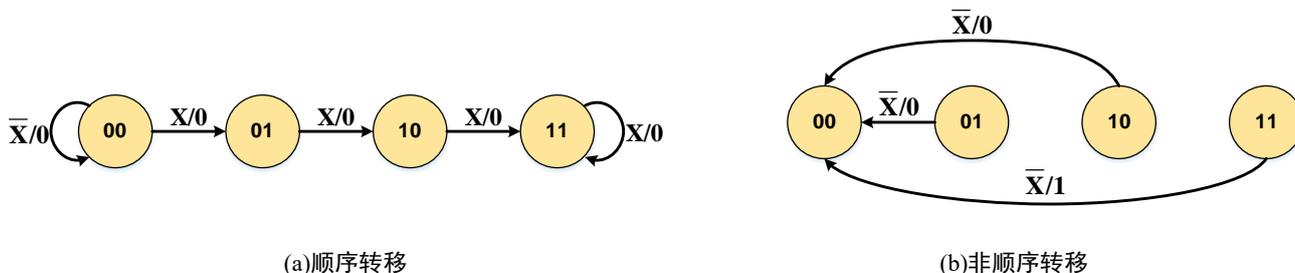


图 4.6 划分后的状态图

根据划分后的状态图以及 74161 的功能，可修改状态转移表如表 4.3 所示。

表 4.3 74161 实现二进制编码的“1110”序列检测器状态转移表

文字描述				二进制编码				74161 集成计数器						
现态	次态	条件	输出	现态		次态		功能	端口					
				Q ₁	Q ₀	Q ₁	Q ₀		EN	LD	D ₃	D ₂	D ₁	D ₀
S ₀	S ₀	\bar{X}	0	0	0	0	0	保持	0	1	x	x	x	x
	S ₁	X	0			0	1	计数	1	1	x	x	x	x
S ₁	S ₀	\bar{X}	0	0	1	0	0	置数	0	0	0	0	0	0
	S ₂	X	0			1	0	计数	1	1	x	x	x	x
S ₂	S ₀	\bar{X}	0	1	0	0	0	置数	0	0	0	0	0	0
	S ₃	X	0			1	1	计数	1	1	x	x	x	x
S ₃	S ₀	\bar{X}	1	1	1	0	0	置数	0	0	0	0	0	0
	S ₃	X	0			1	1	保持	0	1	x	x	x	x

根据状态转移表求得状态转移方程如下：

$$EN = X \cdot \bar{Q}_1^{n-1} \cdot \bar{Q}_0^{n-1} + X \cdot \bar{Q}_1^{n-1} \cdot Q_0^{n-1} + X \cdot Q_1^{n-1} \cdot \bar{Q}_0^{n-1}$$

$$LD = \bar{Q}_1^{n-1} \cdot \bar{Q}_0^{n-1} + X \cdot \bar{Q}_1^{n-1} \cdot Q_0^{n-1} + X \cdot Q_1^{n-1} \cdot \bar{Q}_0^{n-1} + X \cdot Q_1^{n-1} \cdot Q_0^{n-1}$$

$$D_3 = 0 \quad D_2 = 0 \quad D_1 = 0 \quad D_0 = 0$$

$$Z = \bar{X} \cdot Q_1^{n-1} \cdot Q_0^{n-1}$$

实现这些计数器控制信号，可以选择译码器或数据选择器，电路原理图如图 4.7 所示。

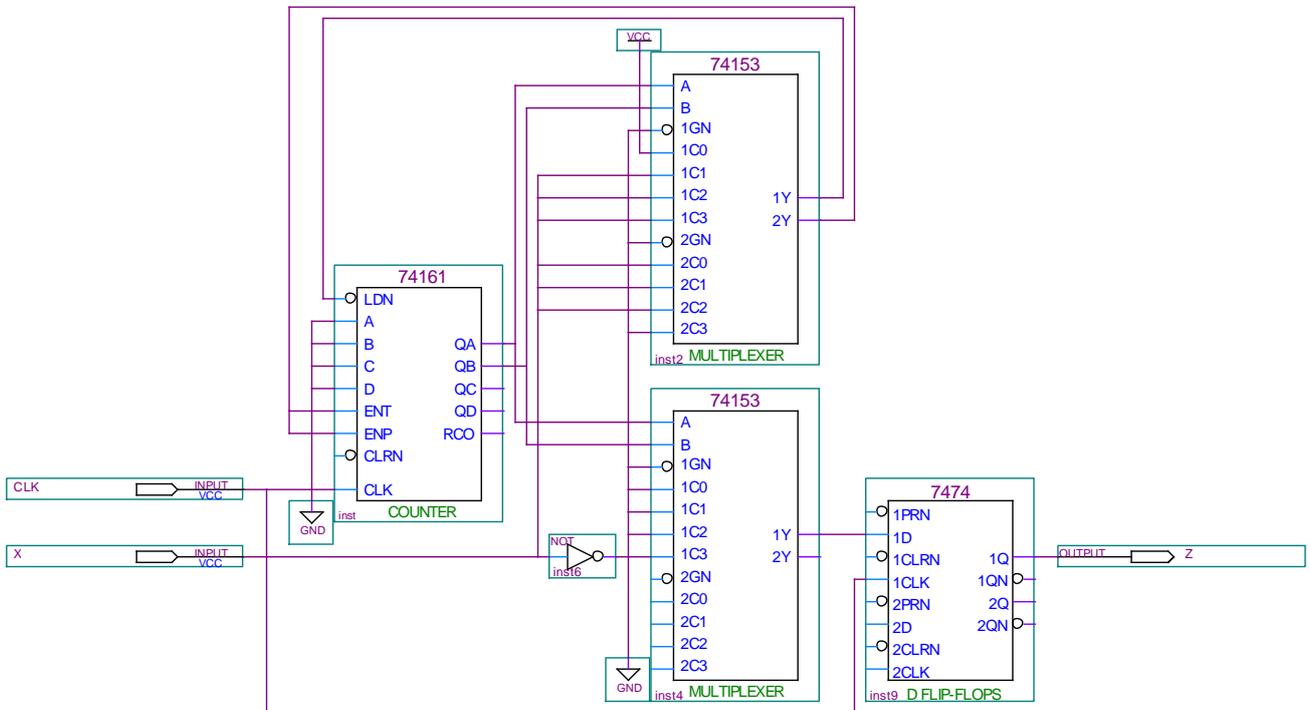


图 4.7 二进制编码“1110”序列检测器电路原理图

3、算法状态机 (ASM Algorithmic State Machine)

算法状态机方法是设计有限状态机的一种方法。在数字电路设计中，算法状态机图是对时序逻辑状态转移的一种图形描述。在功能上，算法状态机图与状态图类似，但是算法状态机类似于传统的计算机流程图，更容易理解和掌握。算法状态机图基本元素包括状态框、判断框、条件输出框以及输入输出路径。由于摩尔型有限状态机的输出只与当前的状态有关，因此其输出情况被标注在状态框内部，而米利型有限状态机的状态块中不会标注输出情况。对于上面的“1110”序列检测器，我们可以画出 ASM 图如图 4.8 所示。将系统中的数据通路去除，即去掉所有条件输出，只保留控制通路，可得控制通路的 ASM 图如图 4.9 所示

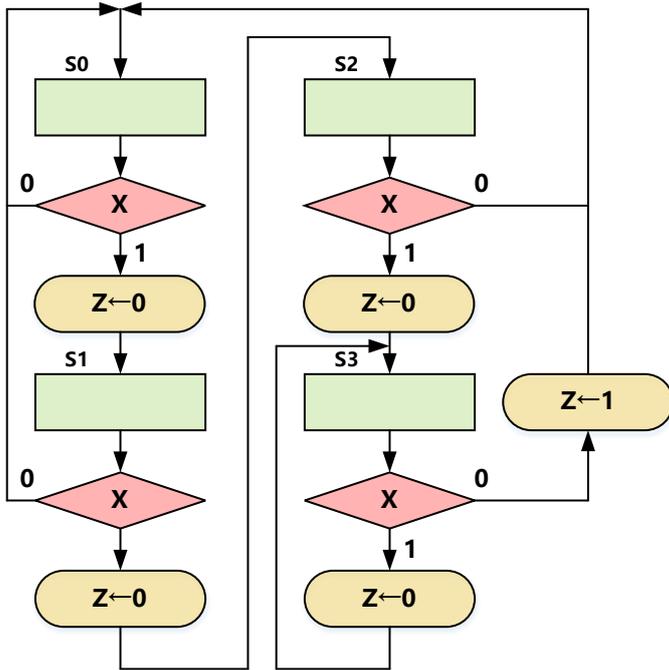


图 4.8 “1110”序列检测器电路 ASM 图

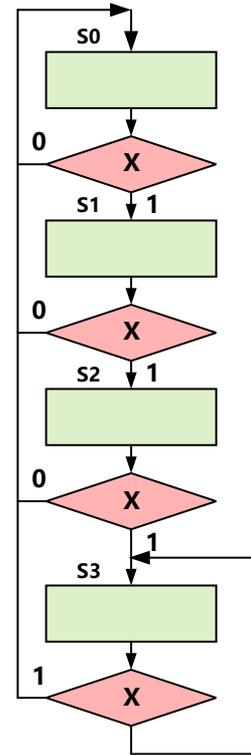


图 4.9 “1110”序列检测器电路 ASM 控制通路

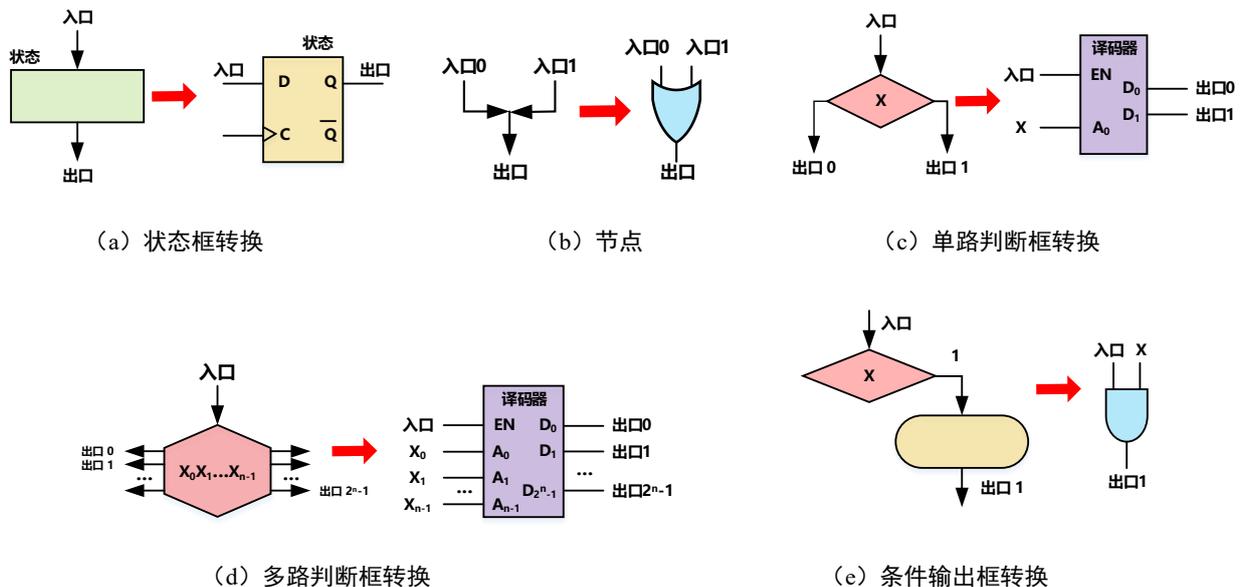


图 4.10 采用 one hot 编码的控制单元转换规则

实验五 FPGA 小系统设计

学习目标

1. 综合前面所学的各项内容
2. 了解掌握数字系统设计的流程和方法
3. 掌握复杂电路连接和调试技能

时间要求:

1. 实验时间: 第 13、14、15 周
2. 报告提交: 第 16 周课内

预备知识

1. 实验教材: 第 5、7 章
2. 二进制串行乘法实现方法

二进制乘法可以采用移位相加的方法。即用乘数的各位数码, 从低位开始依次与被乘数相乘, 每相乘一次得到的积称为部分积, 将第一次(由乘数最低位与被乘数相乘)得到的部分积右移一位并与第二次得到的部分积相加, 将加得的和右移一位再与第三次得到的部分积相加, 再将相加的结果右移一位与第四次得到的部分积相加, ……直到所有的部分积都被加过一次。

以被乘数($M_3M_2M_1M_0$)和乘数($N_3N_2N_1N_0$)分别为 1101 和 1001 为例, 其计算过程如下:

1 1 0 1	
× 1 0 0 1	
0 1 1 0 1	N ₀ 与被乘数相乘的部分积
0 1 1 0 1	部分积右移一位
+ 0 0 0 0	N ₁ 与被乘数相乘的部分积
0 0 1 1 0 1	两个部分积之和
0 0 1 1 0 1	部分积之和右移一位
+ 0 0 0 0	N ₂ 与被乘数相乘的部分积
0 0 0 1 1 0 1	与前面部分积之和相加
0 0 0 1 1 0 1	部分积之和右移一位
+ 1 1 0 1	N ₃ 与被乘数相乘的部分积
0 1 1 1 0 1 0 1	与前面部分积之和相加
0 1 1 1 0 1 0 1	部分积之和右移一位 (最终乘积)

可以用二个寄存器 A、B 分别存放被乘数 M、乘数 N, 累加器 Y 存放部分积。A 和 B 中的数据在加法

器中相加后送入 Y 中。A 和 Y 相级联又构成了一个 8bit 的移位寄存器，当它工作于移位模式时，可以实现数据的右移。由于乘数的每一位不是 0 就是 1，对应的部分积不是 0 就是被乘数本身，所以实际作部分积相加这一步时，只要判断乘数的对应位，如该位为 1，则将累加器中的数据加上被乘数再移位；如该位为 0，就不加被乘数而直接移位。

运算时首先将累加 Y 清零，并将被乘数 M 和乘数 N 分别存入寄存器 A 和 B，然后依据寄存器 B 中最右一位 B_0 (数据 N_0) 确定第一个部分积。将此部分积送入 A 累加器以后，将 Y 连同寄存器 B 右移一位，部分积的最低位被移进寄存器 B 的最左位，乘数的最低位 N_0 被移出寄存器 B，而乘数的次低位 N_1 被移至寄存器 B 的 B_0 位。第二次仍然依据 B_0 位的数据 (N_1) 来确定第二个部分积，将部分积与累加器中的数据相加后右移一位， N_1 又被移出寄存器，数据 N_2 被移到 B_0 位置， \dots 。这样，经过 4 次部分积相加位的操作，完成 1 次乘法运算，乘数 N 恰好被移出寄存器 B，寄存器 B 中保存的就是运算积的低 4 位数据。移位相加的次数应用一个计数器来控制，每移位一次，计数器计一个数。当计数器计得 4 个数时，发出一个信号，使电路停止操作，并输出运算结果

预习要求

1. 访问 MOOC 平台第六章，预习和本实验相关的内容
2. 查找二进制加减运算相关
3. 查找二进制码转 BCD 码硬件算法相关资料
4. 提前完成预习报告，包括系统设计过程、顶层模块框图、单元模块框图、状态图、ASM 图、状态表、状态方程、模块电路原理图等

必做实验

1. 实验基础部分(70%)

设计一个简易计算器，它具有下列运算功能：

- (1) 一位十进制数的相加；
- (2) 一位十进制数的相减；
- (3) 一位十进制数的相乘，必须采用串行乘法实现；
- (4) 数值和运算符用 4×4 键盘输入（实验室提供接口程序），其中 A 为“+”，B 为“-”，C 为“×”，E 为“=”
- (5) 数值用数码管以十进制形式显示，以加法为例，初始时显示全“0”，先输入被加数，再输入运算符，按下运算符键后，数码管显示全“0”，再输入加数，方法和前面一样，最后按下“=”，数码管显示运算结果

2. 实验扩展部分(30%)

- (1) 将电路功能扩展为两位十进制数的相加、相减和相乘。
- (2) 其他自选功能

3. 实验要求：

- (1) 根据设计要求划分设计层次、单元模块和接口信号，在预习报告上记录设计过程，绘制系统框图，每个模块的状态转移图或 ASM 图，并设计验证方案。
- (2) 用原理图输入法设计所有单元模块并编译，分析编译时产生的错误和警告信息
- (3) 对所有的单元模块进行功能仿真，并记录和分析全部仿真结果
- (4) 在顶层文件中连接全部单元模块并编译、综合、分配管脚和适配。
- (5) 对整个系统进行时序仿真，并记录和分析仿真结果。
- (6) 将仿真正确的设计下载到实验箱上，连接输入输出设备和示波器进行板级验证

附录一 Fritzing 简明操作指南

Fritzing 是一款完全免费的、绿色开源的电子设计自动化软件，下面简单介绍一下如何利用该软件绘制电路硬件连接示意图。在使用前需要根据操作系统版本下载对应的 [32 bit Fritzing](#) 软件或 [64 bit Fritzing](#) 软件。下载完成后在下载目录里找到压缩包，解压后打开目录里的“Fritzing.exe”，即可运行 Fritzing。Fritzing 有 4 种视图，分别是面包板、原理图、PCB、code，单击切换到“面包板”视图，如图 1 所示，需要绘制硬件连接图的电路原理图如图 2 所示。

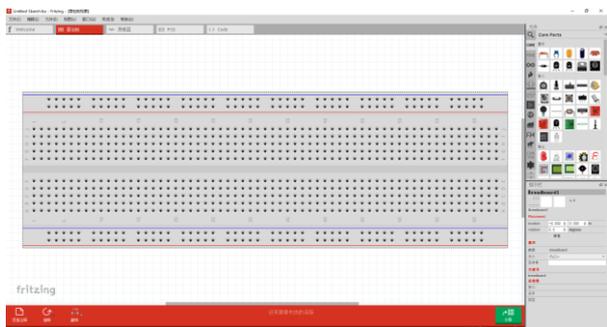


图 1 Fritzing 面包板视图

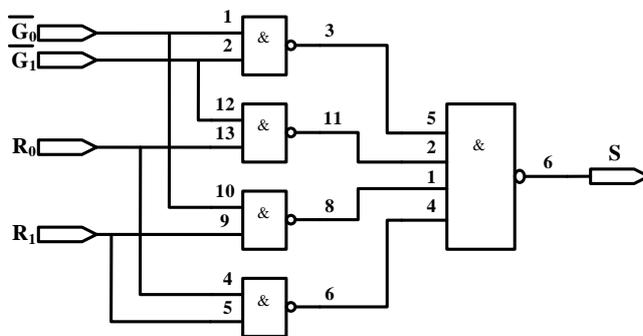


图 2 电路原理图

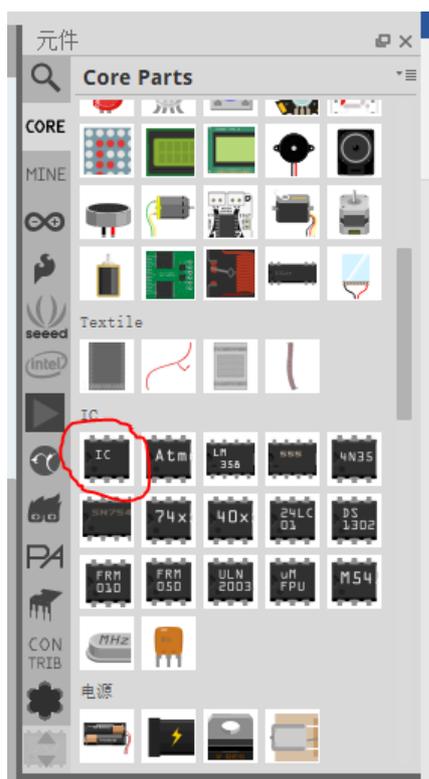


图 3 IC 器件



图 4 指示栏

- 1、在右侧“元件”栏中，选中“CORE”部件列表，下拉，找到“IC”器件。如图 3 所示，拖动“IC”器件到面包，并放在合适的位置，如图 4 所示。
- 2、修改引脚数为“14”，修改芯片标签为“74HC00”，电路如图 5 所示。
- 3、用鼠标点中按“Del”键删除图中不用的连线，如图 6 所示
- 4、重复 1 到 3，放置元件 74HC20，如图 7 所示

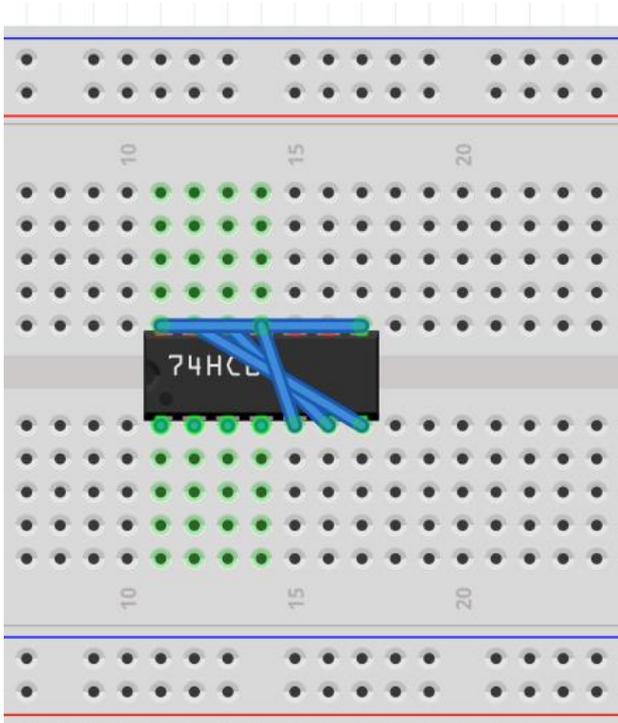


图 5 修改后的电路

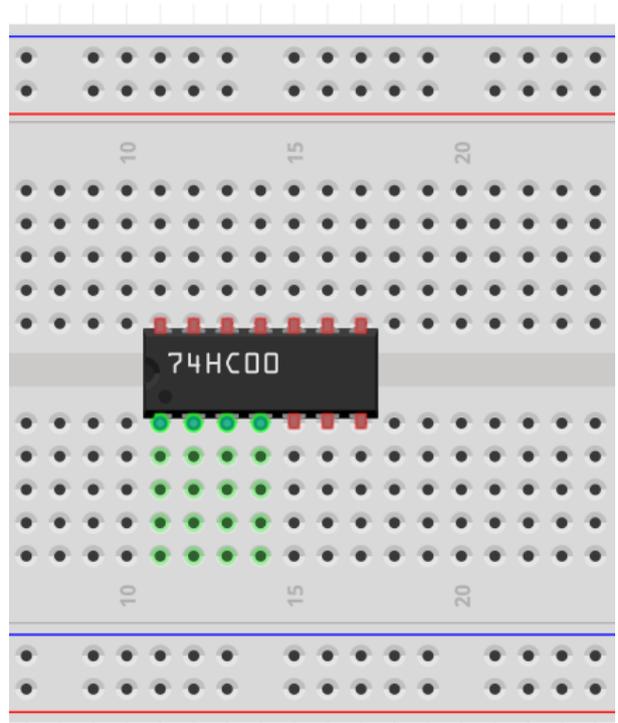


图 6 删除连线后的电路

- 5、首先连接 7400 的 2 脚和 12 脚，鼠标移到准备连线的起点，按住左键，将鼠标移动到连线的终点，松开，如图 8 所示
- 6、鼠标点住连线，根据需要，修改连线形状，修改后的电路如图 9 所示
- 7、重复上述步骤，绘制全部电路，如图 10 所示。
- 8、添加电源和地线，可以右键点击连线，修改连线颜色，如图 11 所示
- 9、添加输入输出信号，如图 12 所示
- 10、 点击文件菜单，选择保存电路文件。

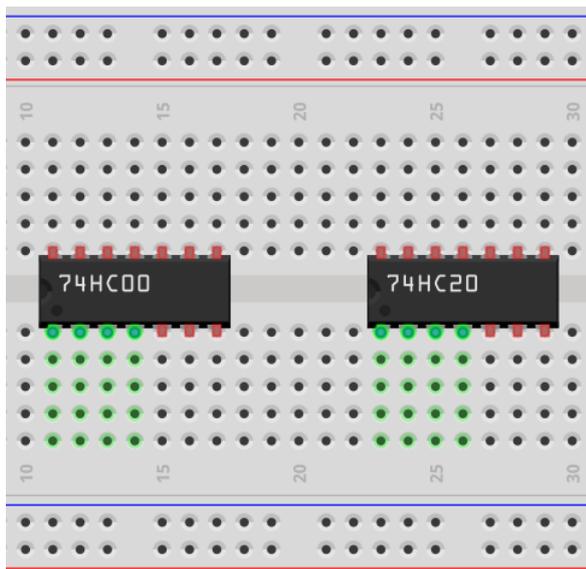


图 7 添加 74HC20 后的电路

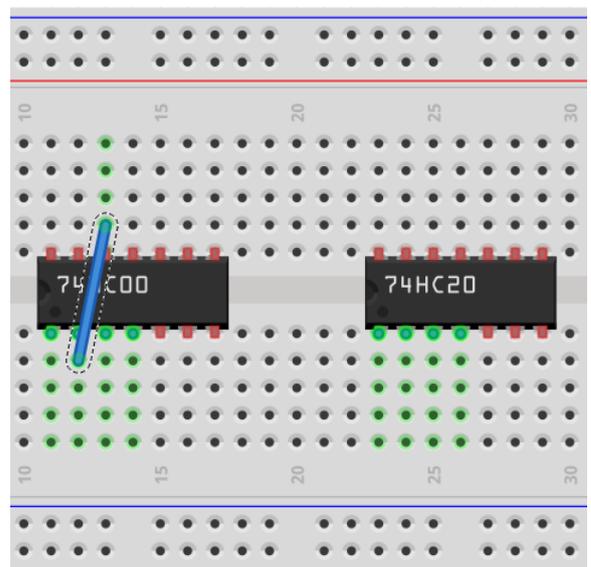


图 6 连线后的电路

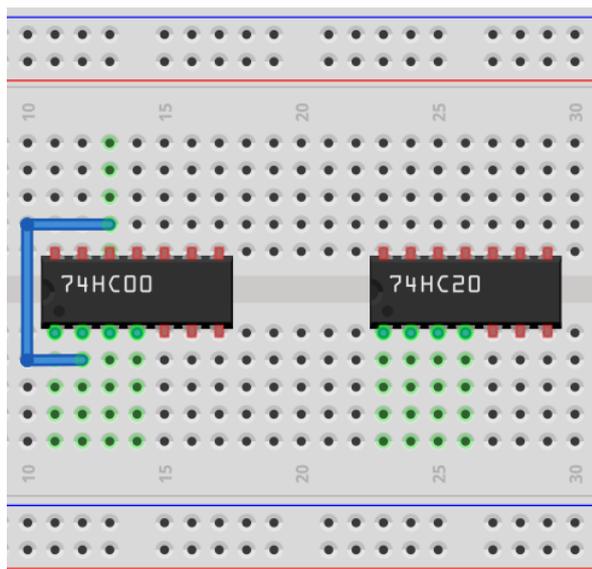


图 9 修改连线后的电路

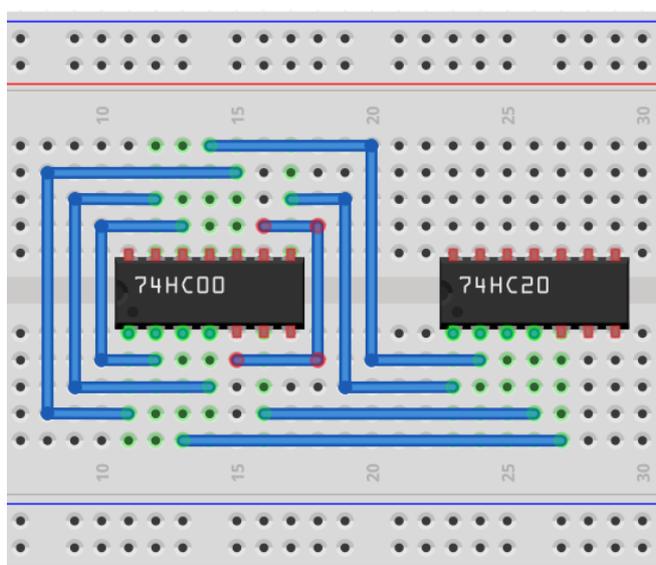


图 10 完成信号连线后的电路

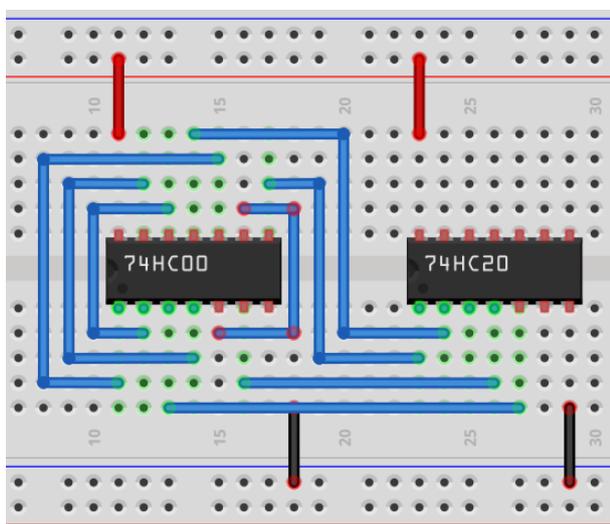


图 11 添加电源和地线后的电路

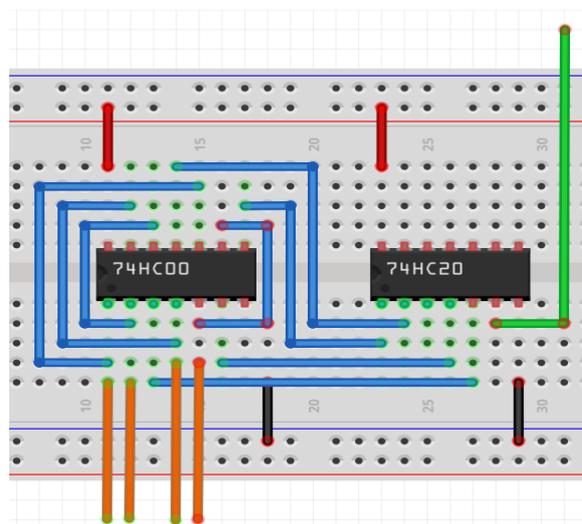


图 10 完成信号连线后的电路

附录二 Quartus II 简明操作指南

在前面的实验里，我们所有的实验都是基于这样一个观点，即将一个数字系统划分成合适利用已有的中小规模数字集成电路的功能的模块，然后将这些集成电路通过外部引线连接起来。现在开始我们将研究用可编程逻辑器件(PLD/FPGA)来进行数字系统设计。

可编程逻辑器件是一种大规模的集成电路，其内部预置了大量易于实现各种逻辑函数的结构，同时还有一些用来保持信息或控制连接的特殊结构，这些保持的信息或连接确定了器件实现的实际逻辑功能，当改变这些信息或连接时器件的功能也将随之改变。可编程逻辑器件的设计过程和传统的中小规模数字电路设计也不一样，可编程数字系统，无论是 CPLD 还是 FPGA 器件都需要利用软件工具来进行设计。可编程数字系统设计总体上一般可以分为设计输入、项目处理、设计校验和器件编程这四个主要过程。下面我们将一个简单的模 60BCD 计数器为例，说明可编程数字系统设计的基本流程、概念和方法，掌握 Quartus II 软件的基本功能和操作，了解原理图输入方式的设计全过程。

一、设计项目输入

设计输入是设计者对系统要实现的逻辑功能进行描述的过程。设计输入有多种表达方式，本次我们主要学习图形输入法。

1.1 建立工程项目

1. 打开 Quartus II，在 File 菜单中选择 New Project Wizard 项，将出现工程项目建立向导对话框。
2. 点击“Next”，进入到相应的对话框，在最上面的文本输入框中输入项目所在的目录名（注意：不能用中文名，下同），在中间的文本输入框中输入项目名称，在图 1 中中额的文本输入框中输入最顶层模块的名称。



图 1 项目名称输入

3. 点击“Next”，进入到设计文件选择对话框，由于在本例中还没有任何设计文件，所以不选择任何文件。
4. 点击“Next”，进入到图 2 所示的器件选择对话框，在“Family”下拉菜单中选择“CycloneIII”，在“Available Devices”列表栏中选择“EP3C5E144C8”。
5. 点击“Next”进入到第三方 EDA 工具选择对话框，在这个界面我们可以选择第三方的综合工具、仿真工具和时延分析工具。由于在本例中我们的综合、仿真和时延分析都采用 Quartus II 内置的工具，所以在这个页面不作任何选择。
6. 点击“Next”进入到 “Summary”对话框，在这个窗口列出了前面所作设置的全部信息。点击“Finish”完成

工程项目建立过程，回到主窗口。

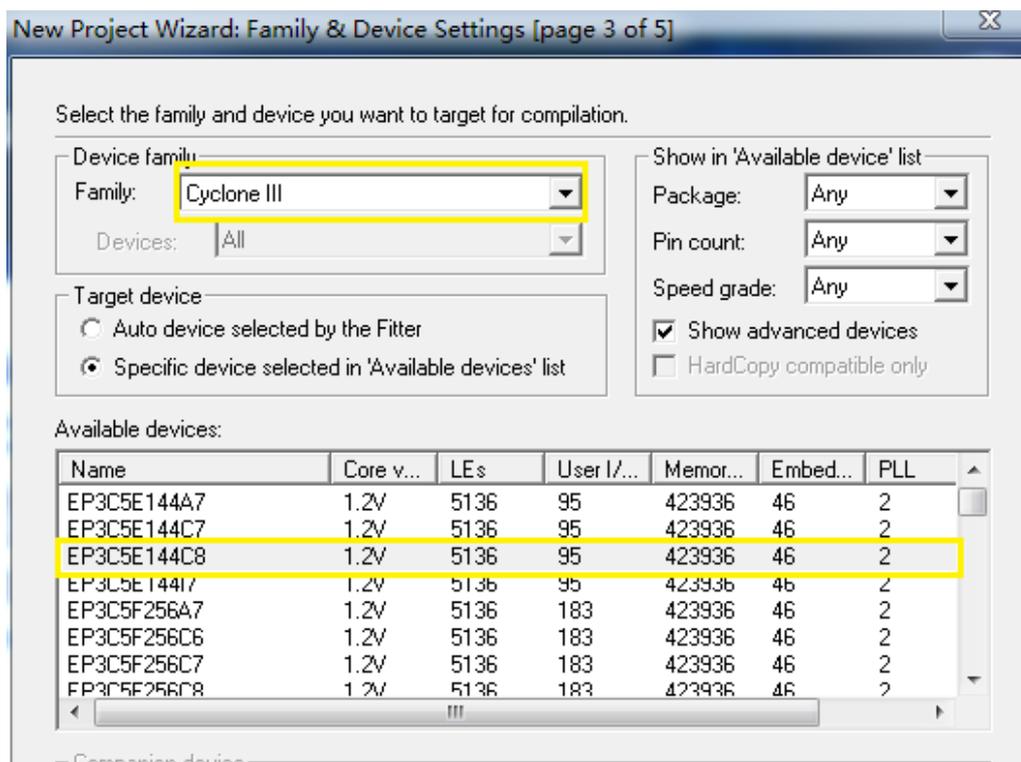


图 2 器件选择对话框

1.2 建立原理图输入文件

在 Quartus II 中我们可以利用 Block Editor 以原理图的形式进行设计输入和编辑。Block Editor 可以读取并编辑后缀名为“.bdf”的原理图设计文件以及在 MAX+PLUS II 中建立的后缀为“.gdf”的原理图输入文件。

1. 在 File 菜单中选择 New 项，将出现新建文件对话框。选择“Design File/Block Diagram/Schematic File”项。
2. 点击“OK”，在主界面中将打开“Block Editor”窗口。“Block Editor”包括主绘图区和主绘图工具条两部分。主绘图区是用户绘制原理图的区域，绘图工具条包含了绘图所需要的一些工具。简要说明如下：

 **选择工具：**用于选择图中的器件、线条等绘图元素；

 **插入器件：**从元件库内选择要添加的元件

 **插入模块：**插入已设计完成的底层模块；

 **正交线工具：**用于绘制水平和垂直方向的连线；

 **正交总线工具：**用于绘制水平和垂直方向的总线；

 **打开/关闭橡皮筋连接功能：**按下，橡皮筋连接功能打开，此时移动元件连接在元件上的连线也跟着移动，不改变同其他元件的连接关系；

 **打开/关闭局部正交连线选择功能：**按下时打开局部正交连线选择功能，此时可以通过用鼠标选择两条正交连线的局部；

 **放大和缩小工具：**按下时，点击鼠标左键放大，右键缩小显示绘图工作区；

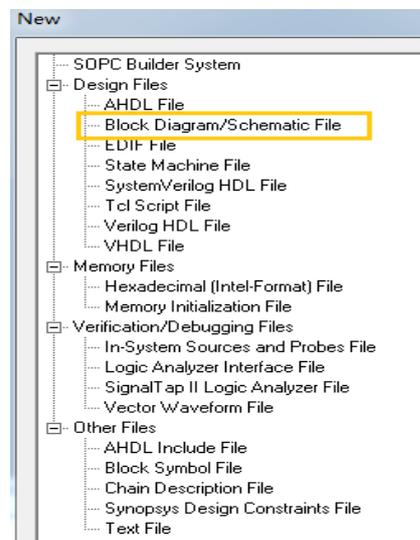


图 3 新建原理图文件

-  **全屏显示**: 将当前主窗口全屏显示;
-  **垂直翻转**: 将选中的元件或模块进行垂直翻转;
-  **水平翻转**: 将选中的元件或模块进行水平翻转;
-  **旋转 90 度**: 将选中的元件或模块逆时针方向旋转 90 度;

3. 元件的添加:

在主绘图区双击鼠标左键, 弹出图 4 所示的 Symbol 对话框, 在 name 栏输入需添加的元件, 如 7400 或 nand2 (二输入与非门), not (非门), vcc (5v 电源、高电平), gnd (接地、低电平), input (输入引脚), output (输出引脚) 等, 回车或点击 ok, 此时在鼠标光标处将出现该元件图标, 并随鼠标的移动而移动, 在合适的位置点击鼠标左键, 放置一个元件。也可以利用插入器件工具  来添加元器件, 方法类似。

在“Libraries”栏中显示目前你已经安装的元件库, 一般缺省会有 mega functions、others 和 primitives 这三个库。其中 mega functions 是参数化模块库, 包含了一些参数可调、功能复杂的高级功能模块; others 库中则包含了原来 MAX+PLUS II 中的部分器件库, 其中包括了大部分的 74 系列中规模逻辑器件; primitives 库是基本库包含一些基本的逻辑器件, 如各种门、触发器等。

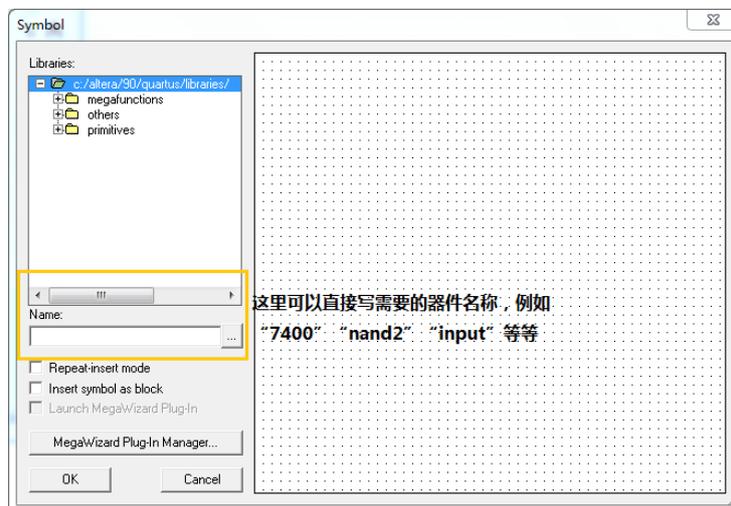


图 4 找出需要的器件名称

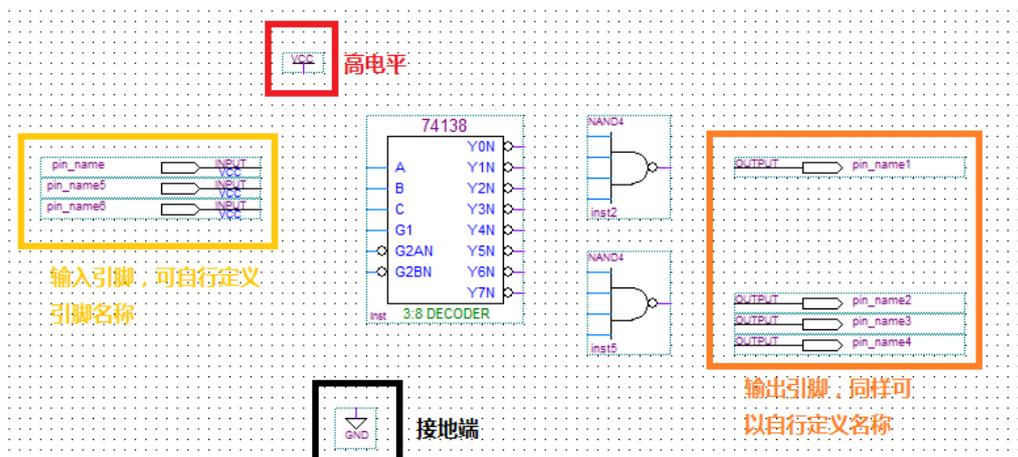


图 5 将器件放置在工作区

4. **命名输入输出引脚**: 双击输入输出引脚的“PIN_NAME”, 输入自己定义的名字即可。
5. **器件的连接和修改**: 连接元器件的两个端口时, 先将鼠标移到其中一个端口上, 这时鼠标指示符自动变

为“+”形状，然后一直按住鼠标的左键并将鼠标拖到第二个端口，放开左键，则一条连接线被画好了。如果需要删除一根连接线，可单击这根连接线使其成高亮线，然后按键盘上的“Delete”键即可。

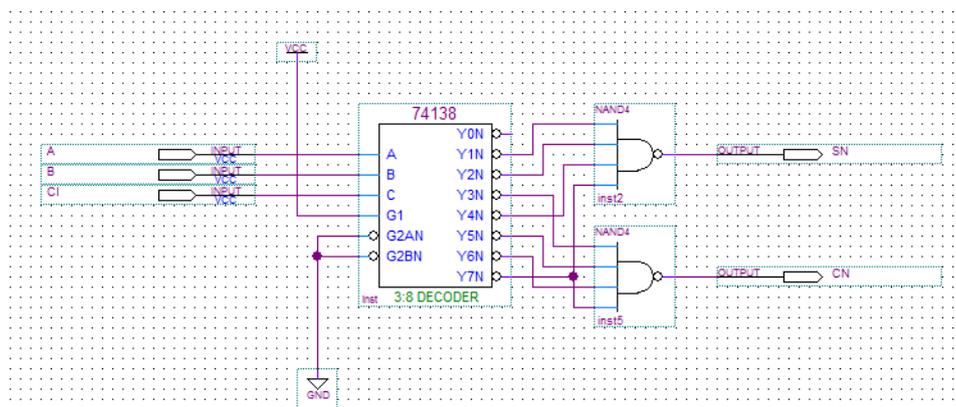


图 6 连接好线和输入输出端口的原理图

- 注：1、连线时走线请避开元件或端口的轮廓虚线；（图中浅绿色虚框）
2、连线中线段可交叉，但如果出现连接点则表示两线段相连相通；
3、输入端通常不可悬空。

6. **保存文件：**从“File”菜单下选择“Save”，出现文件保存对话框。单击“OK”，使用默认的文件名存盘。默认的文件名为项目顶层模块名加上“.bdf”后缀。

二、设计项目处理

在完成输入后，设计项目必须经过一系列的编译处理才能转化为可以下载到器件内的编程文件。

1. 点击主工具栏上的 按钮，开始“Analysis and Synthesis”编译过程。注意应该将要编译的文件设置成顶层文件才能对它进行编译，设置方法为：点击左边 Project Navigator/files，打开 files/Device Design Files，选中要编译的 bdf 文件，点击鼠标右键，在图 7 所示弹出的菜单中选择 Set as Top-level Entity。
2. 在项目处理过程期间，所有信息、错误和警告将会在自动打开的信息处理窗口中显示出来。如果有错误或警告发生，双击该错误或警告信息，就会找到该错误或警告在设计文件中的位置。其中错误必须要修改，否则无法执行后续的项目处理，对于警告则要分情况处理。
3. **分配引脚：**Analysis and Synthesis 全部通过后，为了把我们的设计下载到实际电路中进行验证，还必须把设计项目的输入输出端口和器件相应的引脚绑定在一起。有两种方法可以实现这个过程，一种是给引脚分配信号，另一种则是给信号分配管脚。在此只介绍给信号分配管脚的方法：

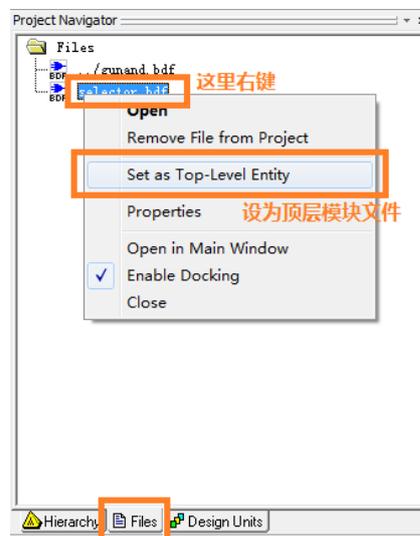


图 7 设置项目顶层文件

选择菜单 Assignments→Pins, ”Assignments Editor”窗口。选择菜单 View→Show All Known Pin Names, 此时编辑器将显示所有的输入输出信号，其中“Node Name”列是信号列，“Location”列是引脚列，“General Function”列显示该引脚的通用功能。对于一个输入输出信号，双击对应的“Location”列，在弹出的下

拉列表框内选择需要绑定的管脚号。完成所有引脚的绑定，保存修改，此时原理图设计文件将给输入输出端口添加引脚编号。

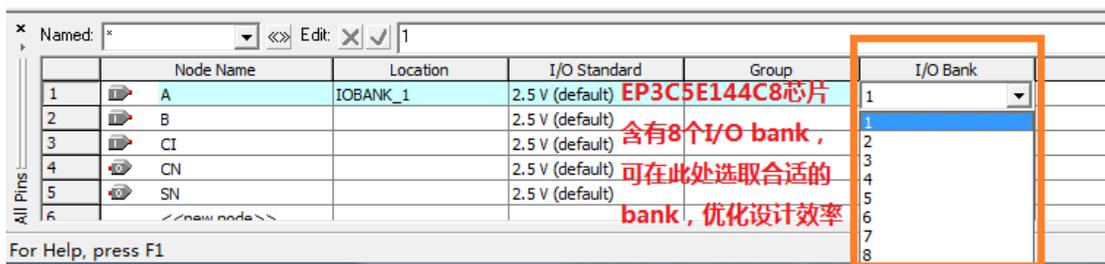


图 8 根据需要选取合适 I/O bank



图 9 给信号分配管脚

4. 布局布线、生成编程文件和时序分析：Analysis and Synthesis 和管脚分配完成后，可以点击  进行全编译

三、设计项目校验

在完成设计输入和编译后，我们可以通过软件来检验设计的逻辑功能和计算设计的内部定时是否符合设计要求。常见的设计项目校验包括功能仿真、定时分析和时序仿真。

3.1 建立输入激励波形文件 (.vmf):

在做仿真之前，必须先建立波形激励文件，具体步骤如下：

- 在“File”菜单中选择“New”打开新建文件对话框，在“Verification/Debugging Files”中选择“Vector Waveform File”项后选择“OK”。
- 编辑器窗口的节点名称栏(Name)空白处单击鼠标右键，在该菜单中选择“Insert Node or Bus...”项，弹出 Insert Node or Bus 对话框，点击“Node Finder”按钮，打开“Node Finder”对话框，单击“List”按钮可以在“Nodes Found”栏中看到在设计中的所有输入/输出信号，当选中信号时，蓝色高亮，表示被选中。单击“>”按钮可将选中的信号移动到“Selected Nodes”区，表示可对这些信号进行观测。点击“OK”按钮，回到“Insert Node or Bus”对话框，再点击该对话框的“OK”按钮。
- 从菜单“File”中选择“Save”，将此波形文件保存为默认名，扩展名“.vmf”表示仿真波形激励文件。

3.2 为输入信号建立输入激励波形:

在波形文件中添加好输入/输出信号后，就可开始为输入信号建立输入激励波形。

- 在“Tools”菜单中选择“Options”项，打开参数设置对话框，选择“Waveform Editor”项设置波形仿真器参数。在这个对话框里我们设置“Snap to grid”为不选中，其他为缺省值即可。
- 从菜单“Edit”下选择“End Time”项，弹出终止时间设定对话框，根据设计需要设置仿真终止时间。

3. 利用波形编辑器工具栏提供的工具为输入信号赋值，工具栏中主要按钮的功能介绍如下：



放大和缩小工具：利用鼠标左键放大/右键缩小显示仿真波形区域；



全屏显示：全屏显示当前波形编辑器窗口；



赋值“0”：对某段已选中的波形，赋值‘0’，即强 0；



赋值“1”：对某段已选中的波形，赋值‘1’，即强 1；



时钟赋值：为周期性时钟信号赋值；

4. 用鼠标左键单击“Name”区的信号，该信号全部变为黑色，表示该信号被选中。用鼠标左键单击  按钮即可将该信号设为“1”。**设置时钟信号方法：**选中信号，单击工具条中的  按钮打开 Clock 对话框，输入所需的时钟周期，单击“OK”关闭此对话框即可生成所需时钟。
5. 选择“File”中“Save”存盘。到此完成激励波形输入。

3.3 功能仿真

可编程系统的仿真一般分为功能仿真和时序仿真。其中功能仿真，主要是检查逻辑功能是否正确。功能仿真方法如下：

1. 在“Processing”菜单下选择“Simulator Tool”项，打开“Simulator Tool”对话框。在“Simulator Mode”下拉列表框中选择“Functional”项，在“Simulation input”栏中指定波形激励文件。单击“Generator Functional Simulator Netlist”按钮，生成功能仿真网表文件。
2. 仿真网表生成成功后，点击“Start”按钮，开始功能仿真。仿真计算完成后，点击“Report”按钮，打开仿真结果波形。
3. 观察输出波形，检查是否满足设计要求。

3.4 时序仿真

时序仿真则是在功能仿真的基础上利用在布局布线中获得的精确延时参数进行的精确仿真，一般时序仿真的结果和实际结果非常的接近，但由于要计算大量的时延信息，仿真速度比较慢。时序仿真的详细步骤如下：

1. 在“Simulator Tool”对话框的“Simulator Mode”下拉列表框中选择“Timing”项，在“Simulation input”栏中指定波形激励文件。
2. 点击“Start”按钮，开始时序仿真。仿真计算完成后，点击“Report”按钮，打开和功能仿真类似仿真结果波形。

四、 器件编程

器件编程是使用项目处理过程中生成的编程文件对器件进行编程的，在这个过程中可以对器件编程、校验、试验，检查是否空白以及进行功能测试。

4.1 Quartus II 器件编程

1. 用下载电缆将计算机并口和实验设备连接起来，接通电源。
2. 选择 Tools→Programmer 菜单，打开 Programmer 窗口。
 - a) 在开始编程之前，必须正确设置编程硬件。点击“Hardware Setup”按钮，打开硬件设置窗口。
3. 点击“Add Hardware”打开硬件添加窗口，在“Hardware type”下拉框中选择“Usb Blaster”，点击 OK 按钮确认，关闭 Hardware Setup 窗口，完成硬件设置。

- 将 Program/Configure 选中。
- 点击“Start”按钮，开始编程。

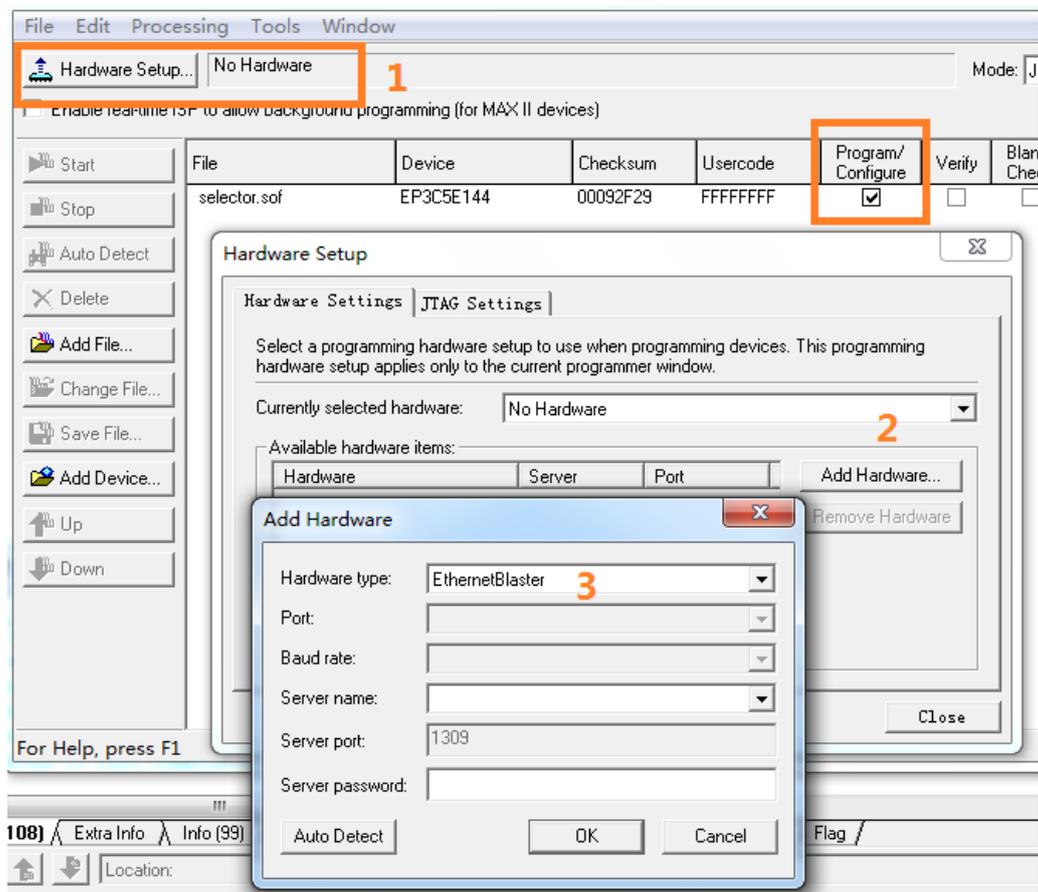


图 10 下载电缆设置