

# 第2章

## 门电路和组合逻辑电路

---

集成逻辑门电路是数字电路的基础，本章在简述集成逻辑门电路特点的基础上，指导学生完成门电路延迟时间测试、三态门构成公共总线等实验。

组合逻辑电路的特点是，电路任意时刻的稳定输出仅取决于该时刻的输入信号，而与电路原来的状态无关。组合逻辑电路实验是数字电路实验的重要部分。本章简述译码器、编码器、数据选择器及运算部件等组合逻辑部件的基本工作原理，给出针对相应实验的具体指导。

### ▶▶ 2.1 门电路特性研究实验

本实验是数字电路的入门实验。通过实验，使学生理解基本 TTL 电路的工作特性，掌握根据实验目的设计实验电路的方法，培养学生分析问题及解决问题的能力。

#### 2.1.1 实验目的

- (1) 掌握门电路的主要特性及逻辑功能，熟悉集成电路器件的引脚和用法。
- (2) 掌握门电路延迟时间的测量方法。
- (3) 掌握门电路延迟时间对电路的影响。
- (4) 掌握三态门构成公共总线的特点和方法。
- (5) 了解“线与”的概念。

#### 2.1.2 实验器件

- (1) 二输入四与非门 74LS00 2 片
- (2) 二输入四异或门 74LS86 1 片
- (3) 三态输出的四总线缓冲门 74LS125 1 片

#### 2.1.3 实验要求

- (1) 测试与非门传输延迟时间  $t_{pd}$ 。



- (2) 测试延迟时间  $t_{pd}$  产生的尖峰信号。
- (3) 设计一个电路，消除尖峰干扰的影响并分析尖峰干扰的原因和消除的方法。
- (4) 测试三态门的逻辑功能，设计一个用 4 个三态门构成的单向公共总线，实现“线与”功能。

## 2.1.4 实验原理

### 1. 集成电路简介

按照集成逻辑门组成的有源器件的不同，集成电路可分为两大类：一类为双极型晶体管集成电路，它主要有晶体管 - 晶体管逻辑门 (Transistor Transistor Logic, TTL)、射极耦合逻辑门 (Emitter Coupled Logic, ECL) 和集成注入逻辑门 (Integrated Injection Logic, I<sup>2</sup>L) 等几种类型。另一类为金属 - 氧化物 - 半导体场效应晶体管 (Metal Oxide Semiconductor, MOS) 集成电路，它又可分为 NMOS (N 沟道增强型 MOS 管构成的逻辑门)、PMOS (P 沟道增强型 MOS 管构成的逻辑门) 和 CMOS (利用 PMOS 管和 NMOS 管互补电路构成的门电路，故又称为互补 MOS 门) 等几种类型<sup>[3]</sup>。

目前，数字系统中普遍使用 TTL 和 CMOS 集成电路。TTL 集成电路工作速度快、驱动能力强、但功耗大、集成度低；CMOS 集成电路集成度高、功耗低，其缺点是工作速度略慢。超大规模集成电路基本上都是 MOS 集成电路，本实验中采用的是 TTL 集成电路。

在 TTL 门的内部电路中，由于三极管和二极管从导通变为截止或从截止变为导通都需要一定的时间，而且还存在二极管、三极管以及电阻、连接线等的寄生电容，所以理想的矩形波电压信号经过门电路后，其输出电压波形要比输入信号滞后，而且波形的上升沿和下降沿要变坏。

在理论上讨论组合逻辑设计的时候，总是假定输入和输出都处于稳定的逻辑电平上。在实际设计中，由于存在门的传输延时、线延时，可能出现偏离理想状态的情况，使得在输入信号逻辑电平发生跳变时，按照理想情况设计的逻辑电路产生错误输出。

### 2. 竞争和冒险

一个门电路的两个输入信号同时向相反的逻辑电平跳变的现象称为竞争。这个跳变可以是一个从逻辑 1 跳变到逻辑 0，另一个从逻辑 0 跳变到逻辑 1，也可以是两个同时从 0 跳变到 1，或从 1 跳变到 0。由于竞争可能会在电路输出端产生不希望出现的尖峰脉冲，称之为冒险。竞争 - 冒险现象产生的原因主要有两个方面：一是信号上升时间和下降时间不同引起的竞争 - 冒险；二是不同传输路径引起的竞争 - 冒险。

### 3. 三态门

在数字系统和计算机电路应用中，有时需要将逻辑门的输出直接连接在一起，以简化电路的结构或控制的便利，如共用数据通道、输出钳位等。但是，普通 TTL 逻辑门的输出是不能直接连接在一起的。

使逻辑门的输出能连接在一起需要解决两个问题：一是器件正常工作不受影响；二是输出连接在一起后的逻辑状态是明确的。解决这两个问题，现在一般采用输出门均截止或集电极开路输出方式。前者增加了一个输出均截止的状态 (高阻态)，称为三态门；后者去掉了集电极直接输出，称为集电极开路门 (简称 OC 门)。

三态门的主要用途之一是实现总线传输，即用一个传输通道 (称为总线)，以选通方式传送



多路信息。使用时，要求只有需要传输信息的那个三态门的控制端处于使能状态，其余各门皆处于静止状态。若同时有两个或两个以上三态门的控制端处于使能状态，则会出现与普通 TTL 门线与运用时的同样问题，因而是绝对不允许的<sup>[4]</sup>。

### 2.1.5 实验预习

- (1) 了解实验箱基本构成（参考 1.3.1 节的实验箱简介）。
- (2) 熟悉示波器的基本使用方法（参考 1.3.2 节的示波器简介）。
- (3) 了解 74LS00 芯片、74LS86 芯片和 74LS125 芯片（参考附录 A），重点是各个引脚的定义。
- (4) 参考图 2-2 实现测试门电路延迟时间的电路、思考如何设计用 74LS00 芯片和 74LS86 芯片实现测试尖峰信号的电路、如何用 74LS125 芯片构成单向公共总线。

### 2.1.6 实验过程及结果分析

- (1) 熟悉示波器的基本用法。重点是如何确定矫正初始化各个调节参数。
- (2) 测试与非门传输延迟时间  $t_{pd}$ 。

① 理解 TTL 门电路延迟时间  $t_{pd}$  的概念，测试电路如图 2-1 所示。

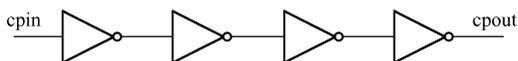


图 2-1 测试门电路延迟时间的电路

用 74LS00 芯片实现的参考测试电路如图 2-2 所示。

其中，1-6，8-13 表示 74LS00 的引脚编号。

注意：7 号引脚（GND）和 14 号引脚（VCC）的正确接入。

② 在示波器上调出类似图 2-3 的波形，并进行分析与测试。

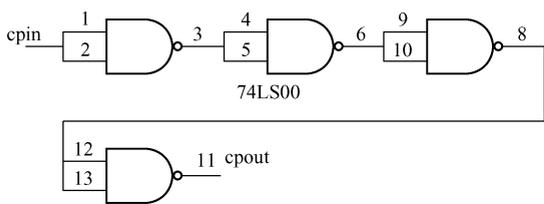


图 2-2 用 74LS00 芯片实现的参考测试电路

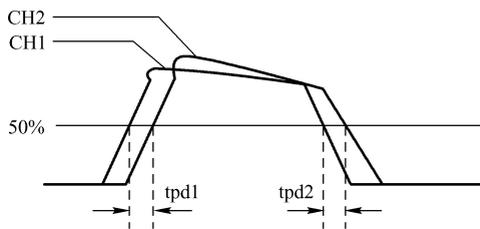


图 2-3 测试延迟时间的波形图

测试结果：

$$t_{pd} = (t_{pd1} + t_{pd2}) / 2$$

其中， $t_{pd1}$ 、 $t_{pd2}$  由示波器读出。一个门电路的延时时间为  $t_{pd}/4$ 。

注意：与非门门电路内部也是由多个二极管、三极管、MOS 管等组成的，由于极间电容分布电感以及传输时间等参数的影响最终导致延时，而极间电容分布电感以及每个二极管、三极管、MOS 管的参数不可能做到完全一样，所以每个门电路的延时时间不一样。

(3) 组建电路并根据实验要求测试尖峰信号，参考电路如图 2-4 所示。

电路中，cpin 连接输入脉冲，由输出 vout 观察尖峰信号。



(4) 思考图 2-4 电路中尖峰信号产生的原因。考虑如何消除尖峰信号。

(5) 三态门逻辑功能。

① 芯片 74LS125 的三态门是低电平有效，当  $C = 0$  时导通， $C = 1$  时截止，处于高阻态。图 2-5 是三态门的逻辑符号。

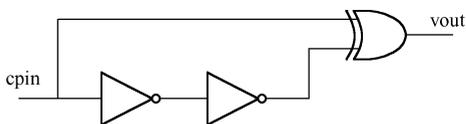


图 2-4 测试尖峰信号电路

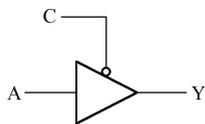


图 2-5 三态门的逻辑符号

② 在面包板上测试三态门逻辑功能。

注意：三态门有两种。第一种是本实验所用的 74LS125，是低电平有效的传输门；第二种是高电平有效的非门。

(6) 用 4 个三态门构成单向公共总线，测试并分析。参考电路如图 2-6 所示。

① 用芯片 74LS125 连接测试电路。

② 为了便于测试，4 个三态门输入端可分别接“0”、“1”、“cp1”、“cp2”。考虑总线的工作特性，工作时， $C1 \sim C4$  只有一个为 0，其余为 1。

### 2.1.7 实验报告及思考题

(1) 通过在示波器上观察“测试与非门传输延迟时间  $t_{pd}$ ”的实验结果，得到什么结论？

(2) 举例说明如何消除尖峰信号的影响（描述原理、给出电路图、比较消除前和消除后的实验现象）。

(3) 如果要构成双向公共总线，该如何设计电路（描述原理、给出电路图）？

### 2.1.8 扩展实验及思考

扩展实验由同学们在课堂完成必做实验后进行，有兴趣且学有余力的同学也可在课下进行。

(1) 学习 Quartus 软件的基本用法（参考附录 B）。

(2) 熟悉利用 Quartus 软件“图形块输入”建立电路的基本方法。

并依此方法重做 2.1.6 节中的 (3) 和 (4)。可适当增加延时电路，根据实验结果，你发现什么问题？

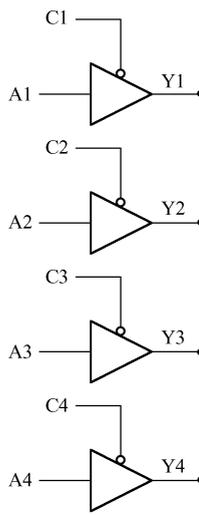


图 2-6 测试用 4 个三态门构成的单向公共总线

## 2.2 组合逻辑电路实验——译码器与编码器的设计

译码器在数字系统中有广泛的用途，不仅用于代码的转换，终端的数字显示，还用于数据分配、存储器寻址和组合控制信号等。

编码器的功能恰好与译码器相反，在输入或输出信号需要编码的场合有着非常广泛的应用。



### 2.2.1 实验目的

- (1) 加深理解译码器和编码器的工作原理。
- (2) 熟悉 VHDL 实现译码器和编码器的方法。
- (3) 学会使用 EDA 软件——Quartus。

### 2.2.2 实验要求

- (1) 设计 4-16 译码器，要求：
  - ① 具有使能端。使能端有效时，译码器根据输入状态，使输出通道中相应的一路有信号输出（为 0'），其余为 1'；使能端无效时，译码器被禁止，所有输出无效（为 1'）。
  - ② VHDL 编程实现，在 Quartus II 环境中进行仿真。
- (2) 设计 BCD 码编码器，要求：
  - ① 具有使能端。使能端有效时，编码器处于正常工作状态，根据输入（10 个数码之一），产生 4 位输出。
  - ② VHDL 编程实现，Quartus II 环境中仿真，下载到实验箱验证。
  - ③ 实验箱的输入开关、数码管，以及设计的编码器三者连接，在数码管上观察实验结果。

### 2.2.3 实验原理

译码器是一个多输入、多输出的组合逻辑电路，它的作用是对给定的代码进行“翻译”，变成相应的状态，使输出通道中相应的一路有信号输出。译码器的种类很多，常见的有二进制译码器、二-十进制译码器和数字显示译码器。

二进制译码器能将  $n$  个输入变量变换成  $2^n$  个输出，且输出与输入变量构成的最小项具有对应关系，如 3-8 译码器（真值表如表 2-1 所示）、4-16 译码器。

二-十进制译码器能将 4 位 BCD 码的 10 组代码翻译成与 10 个十进制数字符号对应的输出信号。

数字显示译码器是不同于上述译码器的另一种译码器。在数字系统中，通常需要将数字量直观地显示出来，一方面供人们直接读取处理结果，另一方面用以监视数字系统工作情况。因此，数字显示电路是许多数字设备不可缺少的部分。

表 2-1 3-8 译码器真值表

| 使能端   |       |       | 代码输入  |       |       | 译码器输出 |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $G_1$ | $G_2$ | $G_3$ | $A_2$ | $A_1$ | $A_0$ | $Y_7$ | $Y_6$ | $Y_5$ | $Y_4$ | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |
| ×     | 1     | ×     | ×     | ×     | ×     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |
| ×     | ×     | 1     | ×     | ×     | ×     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |
| 0     | ×     | ×     | ×     | ×     | ×     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |
| 1     | 0     | 0     | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 0     |



| 使能端   |       |       | 代码输入  |       |       | 译码器输出 |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $G_1$ | $G_2$ | $G_3$ | $A_2$ | $A_1$ | $A_0$ | $Y_7$ | $Y_6$ | $Y_5$ | $Y_4$ | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |
| 1     | 0     | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 0     | 1     |
| 1     | 0     | 0     | 0     | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 0     | 1     | 1     |
| 1     | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 0     | 1     | 1     | 1     |
| 1     | 0     | 0     | 1     | 0     | 0     | 1     | 1     | 1     | 0     | 1     | 1     | 1     | 1     |
| 1     | 0     | 0     | 1     | 0     | 1     | 1     | 1     | 0     | 1     | 1     | 1     | 1     | 1     |
| 1     | 0     | 0     | 1     | 1     | 0     | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 1     |
| 1     | 0     | 0     | 1     | 1     | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |

常用的数码管由七段或八段构成字形，与其相对应的有七段数字显示译码器和八段数字显示译码器。中规模集成电路 74LS47，是一种常用的七段显示译码器，该译码器能够驱动七段显示器显示 0~15 共 16 个数字的字形，其输入  $A_3$ 、 $A_2$ 、 $A_1$  和  $A_0$  接收 4 位二进制码，输出  $Q_a$ 、 $Q_b$ 、 $Q_c$ 、 $Q_d$ 、 $Q_e$ 、 $Q_f$  和  $Q_g$  分别驱动七段显示器的 a、b、c、d、e、f 和 g 段。

所谓编码，是用文字、符号或者数字表示特定对象的过程（即用二进制代码表示不同事物）。根据编码的不同，可将编码器分为二进制编码器和二-十进制编码器两种。二进制编码器是用  $n$  位二进制代码对  $N=2^n$  个信号进行编码的电路，如 3 位二进制编码器（8 线-3 线）；二-十进制编码器是用 4 位二进制代码对 0~9 十个信号进行编码的电路（如 8421BCD 编码器）。根据编码信号是否存在优先级，可分为普通编码器和优先编码器两种。

## 2.2.4 实验预习

- (1) 熟悉各种译码器（尤其是二进制译码器）的功能表。
- (2) 熟悉编码器（尤其是二-十进制编码器）功能。
- (3) 了解实验箱上“数码管区”的功能。
- (4) 了解 Quartus II 的使用（包括建立工程、输入设计文件、仿真、下载等）。
- (5) 设计实现电路。
- (6) 复习 VHDL 语法，构思本实验程序框架。

## 2.2.5 实验过程及结果分析

- (1) 在 Quartus II 下实现 4-16 译码器。
  - ① 根据真值表编写 VHDL 代码。
  - ② 创建工程。
  - ③ 新建 VHDL 文件。
  - ④ 编译、仿真、观察仿真结果。
- (2) 在 Quartus II 下实现 BCD 码编码器，并在实验箱观察运行结果。
  - ① 确定输入（10 个，分别表示 0~9 十个数码之一）和输出（根据实验箱特点，需要 4 位输出，对应为输入十进制数的 BCD 码）。明确输入和输出的关系，编写 VHDL 代码。



- ② 创建工程。
- ③ 新建 VHDL 文件。
- ④ 编译、定义引脚。
- ⑤ 仿真、电路烧录到 EPM7128SLC - 15 芯片。

⑥ 以实验箱逻辑电平区的拨动开关为输入（分别假定每个开关为一个十进制数）、LED 指示灯为输出（每个指示灯为 BCD 码的 1 位），连接电路。

⑦ 以实验箱逻辑电平区的拨动开关为输入、数码管区的共阳极数码管为输出，观察运行结果。

### 2.2.6 实验报告及思考题

(1) 分析 4-16 译码器的仿真结果。

(2) 在自己设计的 4-16 译码器的基础上，实现一个三输入的逻辑函数（自己设定），需要再添加什么电路，电路与 4-16 译码器应该怎么连接，用 VHDL 实现是如何做的？

(3) 说明 BCD 码编码器引脚定义、烧录过程、电路功能检验结果，在数码管上观察结果。

(4) 画出以拨动开关为输入，数码管为输出，自己设计的 BCD 码编码器为中心所连接的整个电路的逻辑框图。

### 2.2.7 扩展实验

如果以实验箱键盘区按键作为输入，数码管区的数码管作为输出，连接输入/输出的电路该如何设计？

## ▶▶ 2.3 组合逻辑电路实验——数据选择器的设计

数据选择器是在地址选择信号的控制下，分时地从多路输入数据中选择一路作为输出的电路，它是目前逻辑设计中应用十分广泛的逻辑部件，有 2 选 1、4 选 1、16 选 1 等类型。

### 2.3.1 实验目的

- (1) 掌握数据选择器的逻辑功能和特点。
- (2) 熟悉 VHDL 实现数据选择器的方法。
- (3) 进一步熟悉 Quartus 的使用。

### 2.3.2 实验要求

设计 4 路数据选择器，要求：

(1) 输入：

① 4 个数据，每个数据是 16 位的二进制数（用 `std_logic_vector(15 downto 0)` 数据类型定义）。选择控制端 2 位，根据控制端的二进制编码，从 4 个输入数据中选择一个需要的数据送到输



出端。

② 使能端，当使能端有效时，输入的 4 个数据中的某个数据输送到输出端，当使能端无效时，输出为高阻态“ZZZZZZZZZZZZZZZZ”。

(2) 输出：一个 16 位的二进制数或者高阻态。

### 2.3.3 实验原理

数据选择器又叫多路开关，它在选择控制（地址码）的控制下，从几个数据输入中选择一个并将其送到一个公共的输出端。数据选择器的功能类似一个多掷开关。4 选 1 数据选择器原理示意图如图 2-7 所示。在 VHDL 语言中，可以用 case 语句来生成一个数据选择器。

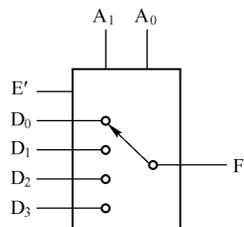


图 2-7 4 选 1 数据选择器原理示意图

### 2.3.4 实验预习

- (1) 熟悉多路选择器功能。
- (2) 复习 VHDL 语法，构思本实验程序框架。

### 2.3.5 实验过程及结果分析

按照实验要求，在 Quartus II 下实现 4 路数据选择器。

- (1) 创建工程。
- (2) 新建 VHDL 文件。
- (3) 编译、仿真、观察仿真结果。

### 2.3.6 实验报告及思考题

- (1) 分析 4 路数据选择器的仿真结果。
- (2) 想一想自己设计的 4 路数据选择器能应用在哪里？

### 2.3.7 扩展实验

将 4 路数据选择器改为“输入 4 个数据，每个数据是 1 位的二进制数”，怎么利用这个数据选择器实现 3 输入逻辑函数？

## ▶▶ 2.4 组合逻辑电路实验——运算部件的设计

全加器是数字系统尤其是计算机中最基本的运算单元电路，其主要功能是实现二进制数算术加法运算。

算术逻辑单元（Arithmetic Logical Unit, ALU）是 CPU 运算的核心。ALU 是一种功能较强的组合电路，其基本组合逻辑结构是超前进位加法器。



### 2.4.1 实验目的

- (1) 掌握全加器的特点及设计方法。
- (2) 掌握超前进位的工作原理。
- (3) 掌握超前进位并行加法器的设计方法。
- (4) 熟悉 VHDL 模块化设计方法。
- (5) 熟悉 ALU 的设计方法。

### 2.4.2 实验要求

- (1) 基于 Quartus II，设计实现一位全加器。
- (2) 利用 VHDL 模块化设计方法，以设计完成的 1 位全加器为基础，设计实现 4 位并行进位加法器。
- (3) 设计实现能完成 8 种算术运算和 8 种逻辑运算的 16 位 ALU，要求：
  - ① 具有 4 位的功能选择端，其中的 1 位用来选择算术运算/逻辑运算，其余 3 位具体给出是哪一种算术/逻辑运算。8 种算术运算中必须包含加法、减法、加 1、减 1 四种运算，其余自定。8 种逻辑运算必须有与、或、非和传递四种运算，其余自定。固定的 8 种运算功能如表 2-2 所示。

表 2-2 运算功能

| 运 算 | 操 作                         | 对标志位 Z 和 C 的影响 |
|-----|-----------------------------|----------------|
| 加法  | result $\leftarrow$ A + B   | 影响标志位 Z 和 C    |
| 减法  | result $\leftarrow$ A + 1   | 影响标志位 Z 和 C    |
| 加 1 | result $\leftarrow$ A - B   | 影响标志位 Z 和 C    |
| 减 1 | result $\leftarrow$ A - 1   | 影响标志位 Z 和 C    |
| 与   | result $\leftarrow$ A and B | 影响标志位 Z        |
| 或   | result $\leftarrow$ A or B  | 影响标志位 Z        |
| 非   | result $\leftarrow$ not B   | 影响标志位 Z        |
| 传递  | result $\leftarrow$ B       | 不影响标志位 Z 和 C   |

其中，A、B 是参与运算的两个 16 位操作数，result 是运算结果（16 位），Z 是零标志位，当运算结果 result = "0000000000000000" 时，Z = 1'，否则 Z = 0'。C 为进位标志位，当运算结果向高位（第 16 位）有进位时，C = 1'，否则 C = 0'。

- ② 用 VHDL 编程实现，在 Quartus II 下编译并仿真。

### 2.4.3 实验原理

所谓全加器是指既考虑低位进位，又考虑对高位进位的加法器，完成 1 位全加算术运算功能的逻辑电路称为 1 位全加器，其输入为  $A_i$ 、 $B_i$ 、 $C_{i-1}$ ，输出为  $S_i$  和向高位的进位  $C_i$ ， $S_i$  和  $C_i$  的



辑表达式为：

$$S_i = (A_i \oplus B_i) \oplus C_{i-1}$$

$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1}$$

真值表如表 2-3 所示。

表 2-3 1 位全加器真值表

| $A_i$ | $B_i$ | $C_{i-1}$ | $S_i$ | $C_i$ |
|-------|-------|-----------|-------|-------|
| 0     | 0     | 0         | 0     | 0     |
| 0     | 0     | 1         | 1     | 0     |
| 0     | 1     | 0         | 1     | 0     |
| 0     | 1     | 1         | 0     | 1     |
| 1     | 0     | 0         | 1     | 0     |
| 1     | 0     | 1         | 0     | 1     |
| 1     | 1     | 0         | 0     | 1     |
| 1     | 1     | 1         | 1     | 1     |

用  $n$  个 1 位全加器实现两个  $n$  位操作数各位同时相加，这种加法器称为并行加法器。并行加法器中传递进位信号的逻辑线路称为进位链。设两个  $n$  位操作数为： $A = A_{n-1}A_{n-2}\cdots A_i\cdots A_0$ ， $B = B_{n-1}B_{n-2}\cdots B_i\cdots B_0$ ，定义辅助函数  $G_i$  和  $P_i$ 。

$$G_i = A_i B_i$$

$$P_i = A_i \oplus B_i$$

$G_i$  称为进位产生函数，其含义：若该位两个输入端  $A_i$  和  $B_i$  均为 1，则必向高位产生进位，此分量与低位进位无关； $P_i$  称为进位传递函数，其含义：当  $P_i = 1$  时，如果低位有进位，则本位必产生进位，即低位传来的进位能越过本位而向更高位传递。因此，

$$C_i = G_i + P_i C_{i-1}$$

串行进位（也叫行波进位）的并行加法器的进位信号逐位形成。其进位信号的逻辑表达式为：

$$C_1 = G_1 + P_1 C_0$$

$$C_2 = G_2 + P_2 C_1$$

.....

$$C_n = G_n + P_n C_{n-1}$$

4 位串行进位的并行加法器逻辑图如图 2-8 所示。

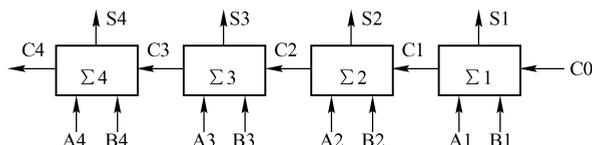


图 2-8 4 位串行进位的并行加法器逻辑图

从串行进位加法器的逻辑图和其进位信号表达式可以看出，这种加法器具有节省器件、成本低的优点，但存在延迟时间长的缺陷。



并行进位（也叫超前进位）的并行加法器的进位信号同时形成。它具有运算速度快的优点。其进位信号的逻辑表达式为：

$$C_1 = G_1 + P_1 C_0$$

$$C_2 = G_2 + P_2 G_1 + P_2 P_1 C_0$$

.....

$$C_n = G_n + P_n G_{n-1} + \dots + P_n P_{n-1} \dots P_2 P_1 C_0$$

4 位先行进位加法器逻辑图如图 2-9 所示。

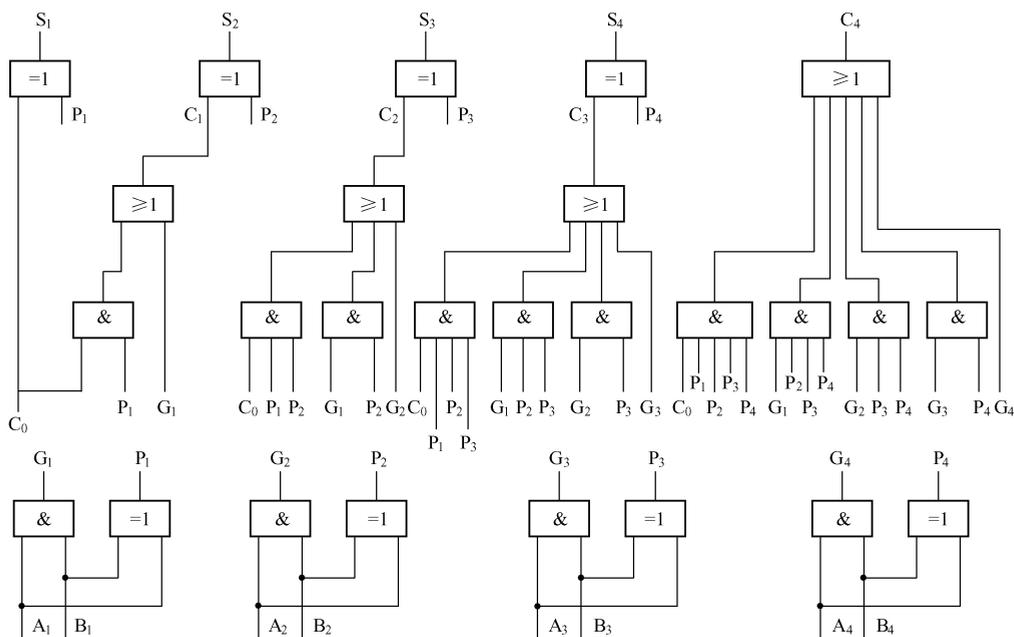


图 2-9 4 位先行进位加法器逻辑图

算术逻辑运算部件 ALU 是一种功能较强的组合逻辑电路，主要完成二进制代码的定点算术和逻辑运算，有时也叫多功能函数发生器，其所完成的算术运算主要包括定点加、减运算等，完成的逻辑运算主要包括逻辑与、或、非、异或等。ALU 的基本逻辑结构是超前进位加法器，它是通过改变加法器的  $G_i$  和  $P_i$  来获得多种运算能力的。

#### 2.4.4 实验预习

- (1) 熟悉 1 位全加器的功能和逻辑结构。
- (2) 充分理解超前进位加法器的工作原理，熟悉其逻辑结构。
- (3) 熟悉 VHDL 模块化设计方法、元件例化语句等。
- (4) 理解并熟悉 ALU 的功能特点。
- (5) 了解 Quartus II 里的 RTL 阅读器。

#### 2.4.5 实验过程及结果分析

- (1) 用 VHDL 设计实现 1 位全加器，在 Quartus II 下编译、仿真。



(2) 设计实现 4 位超前进位的并行加法器。

① 电路输入为两个 4 位的二进制数 A、B 及低位进位  $C_0$ ，输出为 4 位的加法结果 result 及向高位的进位  $C_4$ 。

② 设计超前进位电路，采用 VHDL 模块化设计方法（例如应用元件例化语句），利用已设计的 1 位全加器，完成设计目标。

③ 用 Quartus II 编译通过后，使用 RTL 阅读器观察模块连接情况，仿真观察结果。

(3) 设计算术逻辑运算部件 ALU。

① 按实验要求 3，用 VHDL 编程实现 ALU。

② 用 Quartus II 编译，仿真。

### 2.4.6 实验报告及思考题

(1) 在设计中遇到什么问题？是怎么解决的？

(2) 给出 1 位全加器功能仿真和时序仿真的结果，分析两种结果的不同。

(3) 如何用两个 4 位超前进位的并行加法器构成一个 8 位并行加法器？

(4) 分析 ALU 功能仿真波形。

(5) 如何用加法器实现减法运算？如何判断和的正负及溢出？

(6) 是否可以以加法器为核心实现简单的算术逻辑运算单元？

### 2.4.7 扩展实验

利用 4 位全加器设计  $4 \times 4$  乘法器。