

# GAL22V10 在接口电路中的应用

张旭辉<sup>1</sup>, 马宏伟<sup>1</sup>, 尹 航<sup>2</sup>

(西安科技学院机械工程系, 西安 710054; 2. 西安德赛控制系统责任有限公司)

**[摘 要]** 本文介绍了可编程逻辑器件的发展现状及其在接口电路地址译码中的应用, 在基于 PC 机的多轴步进电机控制卡设计中用 GAL22V10 实现了在接口电路中设置地址、控制总线缓冲和隔离以保护计算机的功能。从使用效果看, 不仅减小了系统体积, 简化了系统设计, 而且减少了级延迟, 提高了系统处理速度, 有助于保证控制卡的实时性要求。

**[关键词]** PLD; 接口电路; ABEL

**[中图分类号]** TN43 **[文献标识码]** B **[文章编号]** 1000-0682(2002)01-0049-03

## The application of the GAL22V10 to an interface circuit

ZHANG Xu-hui<sup>1</sup>, MA Hong-wei<sup>1</sup>, YIN Hang<sup>2</sup>

(1. Mechanical Engineering Department of Xi'an Institute of Science and Technology, Xi'an 710054, China; 2. Xi'an Teser Control System Co., Ltd.)

**Abstract:** The paper presents the use of GAL22V10 in the interface circuit address coding of a PC-based multi-axis stepper control system, and discusses the design of the interface circuit and the characteristic of the GAL device. The actual working results show that the use of GAL22V10 is effective not only in reducing the bulky system and simplifying the system design, but also shortening the delay-time, thus improving the processing efficiency.

**Key words:** PLD, Interface circuit; ABEL

## 1 引言

在接口电路设计中, I/O 端口译码必不可少。IBM PC/XT/AT 微机系统支持的端口地址数为 1024 个, 地址范围从 0 ~ 3FFH, 占有地址线 A0 ~ A9。但是由于系统板占用了相当部分端口地址, 用户开发的扩展板一般只可使用 00H 以后注明保留或者用户根据自己本机情况知道的确未占用的 I/O 端口地址。译码的方法有多种, 如采用一般的组合逻辑电路译码; 采用译码器芯片 (如 74LS138); 利用开关式地址可选译码器方法使扩展卡的口地址能够适应不同的地址分配场合; 还有应用专用译码技术及芯片, 如可编程逻辑器件 PLD (Programmable Logic Device)。PLD 是专用集成电路的一个分支, 它以其灵活的现场可编程特性, 较高的功能集成度和较快的速度成为研制和设计数字系统的最理想选择。目前 PLD

器件已经在计算机硬件、工业控制、数字电路系统等众多领域广泛应用。PLD 器件用于地址译码时, 基本功能就是取代前几种译码电路, 有一个 PLD 器件完成译码任务。本文主要讨论在研制中使用 GAL22V10 进行 I/O 端口控制和地址译码的技术。

## 2 地址译码技术及控制卡电路逻辑设计

在设计接口电路时, 主要考虑 3 方面因素: (1) 总线负载。当 CPU 读插件板上的内存或接口时, 内存芯片或接口芯片将数据传送到系统总线的的数据总线上, 此时数据总线上的所有负载都将成为内存芯片或接口芯片的负载。为保证总线的正常工作, 在接口电路中要增加双向数据驱动; (2) 总线竞争。PC 机属于独立式 I/O 接口寻址方式, 对一个地址, 计算机可以有 I/O 读写、DMA 读写和存储器读写, 它们地址译码是相同的, 要加以区别。解决办法是在译码中加入 AEN 信号, 避免 DMA 操作不会选通 I/O 地址。对存储器地址和 I/O 接口地址相同而产生的混淆, 但这种情况下, 在硬件电路上要对这两种存取进行区别, 避免总线竞争的原则就是只有当 CPU 读接口卡时, 才允许通向系统数据总线的三态

收稿日期: 2001-09-25

作者简介: 张旭辉 (1972-), 男, 硕士, 研究方向: 智能化检测与质量控制。

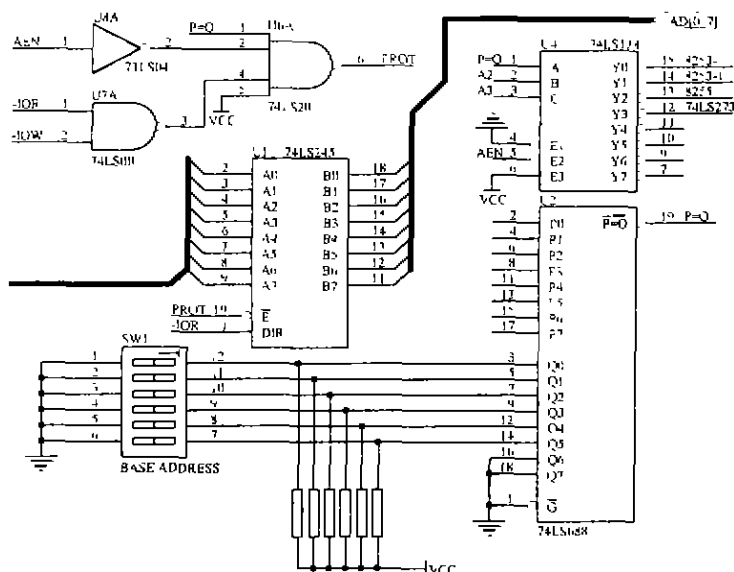


图 2-1 多轴步进电机控制卡译码电路

门导通,其他任何时刻这些三态门必须呈现高阻状态。

(3)接口保护。接口电路还应该考虑由于接口电路出现故障或者工作时的误动作对计算机造成的损坏。

在基于 ISA 总线多轴联动步进电机控制卡设计中我们原采用如图 2-1 所示电路原理图。

## 2.1 缓冲器及保护功能设计

用八同相双向三态缓冲器/驱动器芯片 74LS245 缓冲 ISA 总线扩展槽与各器件间的 8 位数据信号。在这个原理图中, 74LS245 不但起缓冲、隔离作用, 还有一定的保护和控制作用。计算机读控制信号  $\overline{\text{IOR}}$  (低电平有效) 连接到 74LS245 的 DIR (方向控制端), 而门控信号  $\overline{\text{G}}$  接信号 PROT。PROT 信号是由 I/O 端口读写信号  $\overline{\text{IOR}}$ 、 $\overline{\text{IOW}}$  和口地址译码信号  $P=Q$  产生的信号。当读有效为低电平时, 控制卡的数据可以通过 74LS245 输入到计算机; 读有效为高电平时, 计算机侧的数据输出到控制卡。PROT 实现只有计算机与控制卡交换数据时, 选通 74LS245 的门控信号  $\overline{\text{G}}$ , 使之三态门打开。例如调整口地址译码电路使控制卡口地址为 300 ~ 30FH, 只有主机对这个范围的端口地址进行 I/O 指令是, 才选通  $\overline{\text{G}}$ , 否则 74LS245 处于高阻状态, 这时端口不对数据线产生任何影响。引入信号  $\overline{\text{IOR}}$  和  $\overline{\text{IOW}}$  避免了计算机 CPU 访问地址为 300 ~ 30FH 的存储器时误动作打开 74LS245 三态门。

## 2.2 端口地址译码电路

使用了比较器 74LS688, 3~8 译码器 74LS138 以及拨码开关 SW 等器件共同组成口地址译码器。SW 可设定 00H~3FH 间的 16 个连续口地址位置(译码地址线 A9~A4), 再由 A2、A3 与 74LS688 输出

P=Q 信号通过 74LS138 译码器把 16 个口地址分为四段,分别选通两个 8253 和 8255,最后地址线 A0、A1 与 8253 的片选线一起确定 8253 的地址。该电路使控制卡的地址设置在比较大的范围内进行调整,保证卡上用到地址不与 PC 机内其他板卡的地址相冲突,避免不可预计的事情发生。

### 3 GAL22V10 的性能特点

历史上可编程逻辑器件经历了从 PROM、PLA、PAL、GAL、EPLD (Erasable Programmable Logic Device) 到 CPLD (Complex Programmable Logic Device 即复杂的 PLD) 和 FPGA (Field Programmable Gate Array) 的发展过程。GAL (Generic Array Logic) 器件被认为是低密度 PLD 与高密度 PLD 的分界点, 即标准门 500 ~ 750 门之间。GAL 器件是美国 Lattice 公司于 80 年代率先推出的一种高性能的理想 PLD 产品。GAL 器件采用 E<sup>2</sup>CMOS 工艺, 可以进行反复的编程和擦除, 而且具有低功耗、高速的特点。常用 GAL 器件有 GAL16V8、GAL20V8、GAL22V10 等。GAL16V8、GAL20V8 的使用比较广泛。但对于某些应用场合, 这两种 GAL 芯片往往不能满足需要, 其主要不足之处: 乘积项不能超过 8 个; 能配置的输出引脚最多只有 8 个; 不能对寄存器进行复位或置位操作; 对反馈结构的限制比较多, 有些引脚不能反馈等。GAL22V10 是一种通用型 GAL 器件, 它可以从某种程度上解决以上提到的 GAL16V8 和 GAL20V8 不足之处。GAL22V10 内部共有 132 个与门, 而且输出管脚上的与门不是平均分配; 逻辑设计表达式中“与”项最多为 22 个变量的逻辑乘; GAL22V10 的输出宏单元比较简洁, 灵活, 容易

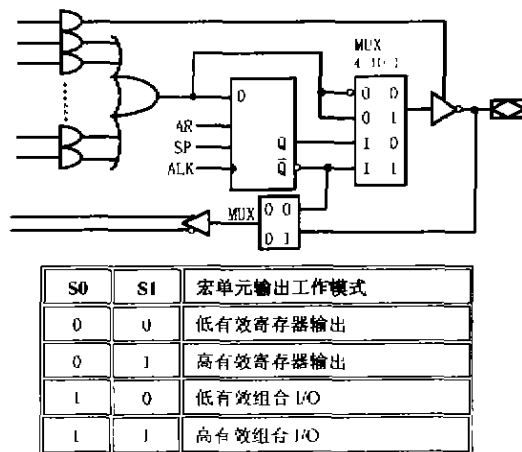


图 3—1 GAL22V10 输出宏单元

使用,没有像 GAL20V8 在不同输出模式下输出反馈的种种限制。GAL22V10 的输出宏单元(OLMC)见图 3—1。每个输出宏单元由各自的 S0 和 S1 控制,可以有 4 种不同的输出工作模式。可编程的置位和复位可以由二个与门进行统一的置位和复位控制。

## 4 设计实例

GAL 22V10 的逻辑设计软件可用 ABEL(3.0 版以上)或其它高级逻辑设计语言。如 Protel 公司的 Protel 98/99 都集成了一个 CUPL 逻辑设计语言。CUPL 语言也是一种编译型硬件描述语言 VHDL (Hardware Description Language), 支持所有的 PLD 器件。Protel99 的 PLD 设计工具支持实时检测功能, 在汇编和仿真的过程中能进行自动检测, 并给出错误提示。通过其中的 PLD-CUPL Wizard 可以很快捷的建立基于原理图或者文本方式的 PLD 设计文件, 对这个文件进行编译, 就可以得到符合工业标准的 JEDEC 文件(\*.JED)。

ABEL 语言是美国 DATA I/O 公司的一种高级编译型逻辑设计语言, 它以其方便、灵活、易于掌握的特点, 深受硬件逻辑设计者的喜爱。利用它高度结构化的语言、灵活多样的逻辑描述形式可以很容易地编写出 Abel 原文件, 编译过程对用户的逻辑设计进行语法检查、逻辑化简、自动生成符合 JEDEC 标准的 JED 文件 (\* .JED), 而且具有模拟仿真功能。设计者可以根据模拟的结果修改设计中的错误, 或发现可能产生故障的隐患, 以提高设计的可靠性。ABEL 软件是通用的 PLD 器件设计软件, 高版本的 ABEL 几乎可以支持任何一种 PLD 器件。

不论选用那一种设计语言,最后都要生成 JEDEC 格式文件,才能通过编程器烧入 PLD 器件。下面以对端口 360H,364H,368H,36CH 的读控制的地址译码为例,说明 GAL22V10 在端口地址译码电路中的应用。首先确定与图 2—1 原理图功能相同的逻辑表达式,在字处理软件(如写字板、记事本、EDIT 等)中编写的 PLD 设计文件(Control. abl),然后在 ABEL 中编译,生成 JED 文件(U1.jed)。若编译通过而且经过模拟仿真与设计的逻辑功能相符合,没有逻辑错误,就可以用西尔特公司的 SnperPRO/L 编程器把 JED 文件写入(“烧入”)GAL22V10 芯片中。我们设计的逻辑通过仿真,与预期要求功能相符。仿真文件内容如下(control.sim):

Simulate ABEL( tm)3.00b

File: '910. out' Module: 'decode' Device: 'U22' Part: 'P22V10'

Use : STEPPER CONTROLLER

Device: GAL22V10

Date : 06/01/200

[illegible]

5 out of 5 vectors passed

由于执行 `abel` 编译命令,只产生简单的仿真通过报告,没有详细的各信号时序图,所以要生成详细时序图,必须用仿真命令(`simulate.exe`)。执行下面命令就得到 `control.sim`。

C:/Abel > simulate control. out control. sim-T4-  
W1,7,8,9,10,15,16,17,18

设计过程中应注意几个问题:

1)信号极性的设置。在管脚说明时,可以定义信号为低电平有效(用一个“!”运算符说明),ABEL自动控制极性进行逻辑转换。如果一个信号被说明为低电平有效,它将在随后的设计描述中遇到时自动取反。取反运算在任何用到该信号时(包括作为输入、输出或测试向量时)都被执行。

2) GAL22V10 与门不是平均分配给每个管脚,而是按中间输出引脚多,两侧输出引脚少的规律分布。如 17 脚和 20 脚各有 14 个与门输入,18 脚和 19 脚各有 16 个与门输入。设计时可以根据输出信号逻辑等式中所含的乘积项数目分配不同的管脚,将复杂一些的输出向量定义在 18、19 引脚。这是解决编译时的提示“乘积项过多”的有效途径。

3)同一逻辑函数可以用多个不同方程式表示。要尽量采用乘积项少的方程式。另外,管脚属性的正确选择可以优化设计,减少乘积项。

## 5 结论

GAL 器件的使用,使得所设计的多轴联动步进电机控制卡口地址选择范围广,调整方便,而且具有总线缓冲、隔离控制及保护计算机的作用。用一片芯片代替了几个芯片,这不仅减小了系统体积,简化了系统设计,而且减少了级延迟,提高了系统处理速度。

[参考文献]

- [1] 薛宏熙,等. MACH 可编程逻辑器件及其开发工具(第二版)[M]. 清华大学出版社, 1998
- [2] Databook of GAL22V10. LATTICE SEMICONDUCTOR CORP [OL]. Mar 2001. <http://www.latticesemi.com>