

微处理器系统设计导论

〔美〕哈利·加兰

著

孙公达 古新生

蔡和刚译

蔡希琨

校

微处理器系统设计导论

(美) 哈利·加兰 著

孙公达、古新生、蔡和刚 译

蔡希尧 校

陕西科学技术出版社

INTRODUCTION TO MICROPROCESSOR
SYSTEM DESIGN

HARRY GALAND

1979 by McGraw-Hill, Inc.

微处理器系统设计导论

[美] 哈利·加兰 著

孙公达、古新生、蔡和刚 译

蔡希尧 校

陕西科学技术出版社出版

(西安北大街131号)

陕西省新华书店发行 陕西省建工局印刷厂印刷

开本787×1092 1/32 印张7 插页2 字数118,000

1981年3月第1版 1981年3月第1次印刷

印数1—3,000

统一书号: 15202·21 定价: 0.80元

前 言

微处理器作为电路元件已使设计的电子电路系统发生重大的变化。目前，微处理器已广泛用于袖珍计算器、实验室仪器与仪表、消费性产品、飞机飞行控制系统、商业计算机系统等领域中。在上述每一个领域以及在许多其他的应用方面，微处理器对系统设计均产生了重要的影响。

本书在考察微处理器本质问题的同时，对微处理器怎样进行工作以及在系统设计中如何应用也加以讨论。重点放在对微处理器为基础的设计有所理解并能实际应用。书中还引用一些专门的设计例子来阐明基本概念。

为了有效地应用微处理器，需要了解微处理器的硬件和软件两方面的知识。因此，本书讨论了这两方面的课题，而每一方面都假定读者已具备了某些初步的基础知识。硬件各章的基础知识是要求具备在大学学习一年电子学课程的水平或与之相当的训练，应当熟悉基本的晶体管电路、逻辑门电路和电源。软件各章要求对计算机程序设计有初步的入门，熟悉存储程序的概念以及具有二进位数制的知识。

本书第一章叙述什么是微处理器及其如何使用。第二章叙述制造微处理器所采用的集成电路技术。第三章介绍几种最流行的微处理器，包括最新的“单片”计算机和功能更强的十六位微处理器。第四章叙述使用微处理器所需的电路，而更进一步的硬件概念——中断、串行接口和直接存储器存取（DMA）则在第五章讨论。

微处理器的硬件一旦确定之后，系统的操作就严格地取决于系统软件。关于微处理器的软件在第六章介绍，第七章介绍高级语言。

应用微处理器实现算术运算的内容在第八章叙述。第九章则讨论如何在数字微处理器中使用模拟信号的技术问题。最后，在第十章阐述可大大扩展微处理器使用范围的标准接口。

本书应结合制造厂家对其微处理器的详细说明书来进行学习与使用。

哈利·加兰

译 者 的 话

这是美国 1979 年出版的有关微处理器系统设计方面的一本普及技术书，取材于目前最先进的微处理器，现作为斯坦福大学的教学用书。作者哈利·加兰是美国克罗门科微处理器公司经理，斯坦福大学讲师。

该书内容系统具体，讲述由浅入深，结合实际通俗易懂。书中除框图和表格外，并在每章后附有适量的练习及解答。很适合具有电路基本知识和计算机一般概念、而希望对微处理器硬件、软件及系统结构有进一步全面了解的读者自学；也可作为大专院校有关专业的教学用书。对于微处理器的操作使用人员、系统设计人员和有关科技人员，本书有一定参考价值。

对原书中的个别错误，在译文中已作改正。为了便于阅读，并附了少量译者注释。

本书承蒙美国宾夕法尼亚大学计算机与信息科学系卡尔教授来西北电讯工程学院讲学时热情推荐，在此表示感谢。译文由西北电讯工程学院蔡希尧教授校阅。

由于我们水平有限，翻译不当之处，请读者予以批评指正。

1979年12月

目 录

前 言

1

导 论

理想的微处理器 (1) 数据总线 (3) 地址总线 (8)
控制总线 (13) 通用微处理器 (16) 小结 (16) 练习 (17)

2

微处理器的工艺技术

双极型工艺技术 (23) MOS工艺技术 (31) 小结 (35)
练习 (35)

3

微处理器的进展

8008 (38) 8080 (42) z80 (47) 8748 (52) 8086 (54)
z8000 (59) 小结 (62) 练习 (63) 参考资料 (63)

4

基本的微处理器硬件

电源 (65) 时钟 (68) 逻辑门 (70) 锁存器 (75) 缓冲器 (75)
存储器 (80) 最小的微处理器系统 (84) 单板计算机 (87)
小结 (88) 练习 (88)

5

微处理器系统的扩展

数据传送的同步 (91) 中断 (96) 数据的串行传输 (98)
直接存储器存取 (100) 存储器——映射输入/输出 (102)
存储体选择 (103) 地址预测 (104) 小结 (106) 练习 (106)

6

微处理器的机器语言

数据的二进制表示 (108) 机器语言的程序设计 (115) 机器周期 (126) 寄存器 (128) 寻址方式 (129) 堆栈操作 (130)
子程序 (132) 练习 (134)

7

汇编语言和高级语言

汇编语言 (138) PL/M (144) BASIC (147) 其他高级语言 (149) 小结 (150) 练习 (151)

8

微处理器的算术运算

加法 (153) 减法 (155) 乘法 (156) 除法 (162) 小结 (162)
练习 (162)

9

模拟接口

数-模转换 (165) 软件定时转换 (168) 软件逐次逼近转换 (170)
硬件逐次逼近转换 (174) 小结 (176) 练习 (177)

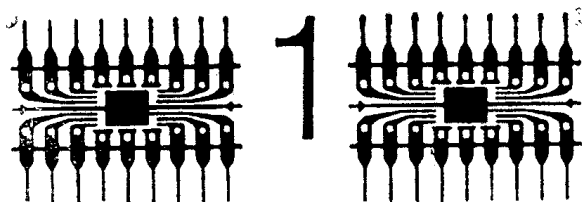
10

标准接口

S—100 微计算机总线 (178) IEEE488接口总线 (192)
RS—232 通信接口 (196) 小结 (201) 练习 (201)

奇数题答案 (203)

索引 (210)



导 论

今天，微处理器尽管有许多不同的型式，但它们却具有许多共同的特点。事实上，由于最近几年微处理器设计所取得的进展，它们已日益成为独立完整的理想的“单片式计算机”。这种计算机是在单块硅片上做成的单块集成电路。它能够执行预先存放的程序，并且把它设计成很容易同外部设备相连接。

在这一章中，我们要介绍理想微处理器的概念，引入一些在讨论微处理器时要用到的定义和习惯用语，并由理想微处理器的概念，给出一个微处理器的通用模型。头脑中有了这样一个通用的模型，对以后各章中新介绍的各种微处理器的工作原理，就容易理解了。

理想的微处理器

首先让我们来看看什么叫做“理想的微处理器”。如图1-1所示，一个理想的微处理器具有 N 条输入线和 M 条输出

线。由于微处理器是一种数字设备，对于每一条输入线，只允许加入两种电平；对于每一条输出线，同样的也只能出现两种电平。这两种电平被分别叫做逻辑0和逻辑1，并以二进制数0和1（称为二进制数或叫做位）作为它们的表示符号。

输入线上的信号是输入到微处理器去的数据。这些数据可以来自开关、传感器、模拟-数字转换器、键盘或者其他类似设备。在这一理想微处理器的内部，存放微处理器的程序。它是一组有顺序的指令，用来确定如何处理输入的数据，以及确定应该送什么样的信息到输出线上，以作为这一输入数据的处理结果。输出线可以接到传动装置、数字显示器、数字-模拟转换器、打印机、报警器或其他任何一种输出设备上。

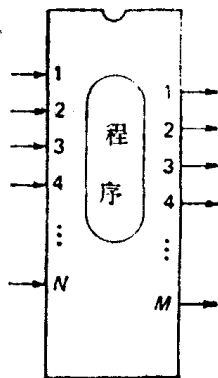


图 1-1 理想的微处理器

从概念上来说，微处理器是一个数字设备，它从一定数目的输入线上接收数据，按照所存放的程序来处理数据，并产生出一定数量的输出信号作为该数据的处理结果。任何一个给定的时刻，在微处理器输出线上的逻辑电平取决于下述两个因素：

1. 在该时刻之前，输入到微处理器的全部信息情况；
2. 微处理器所存放的程序。

微处理器能够在许多方面得到应用的关键就在于相同的或非常类似的硬件，可以通过编制程序的办法，来满足大量不同的应用。

作为使用微处理器的一个具体例子，让我们来看看十字路口上交通指示灯的控制情况。微处理器的每一条输入线接到路口不同的传感器上，当一辆机动车在一个传感器上通过时，对应的输入线上就出现逻辑 1；否则，输入线就处于逻辑 0。微处理器的每一条输出线控制十字路口上相应的一个交通指示灯。逻辑 1 使灯亮，逻辑 0 使灯灭。而所设计的微处理器程序，则使得十字路口的交通能够安全畅通。微处理器的其他输入端，则可用来指示每天的时间或道路的情况和特征等。在一个十字路口上的微处理器的某些输出信息，还可用来作为另一十字路口上的微处理器的输入信息。只要有一个足够完善的微处理器程序，在控制交通指示灯时就可以把上述所有因素都考虑进去。

数 据 总 线

与理想的微处理器不同，当 N 条输入线和 M 条输出线的数目很大时，实际的微处理器就可能做不到。因为任何实际的集成电路的封装，可供利用的引出线数目总是有限的。对于大多数微处理器来说， N 等于 M 。这一数值称为微处理器的数据通道宽度或字长。微处理器就是用数据通道宽度这一最常用的参数来进行分类的。用作与微处理器来往传送数据的线，集中起来称为数据总线。

图 1-2a 表示一个 8 位微处理器，它有 8 位宽度的数据通道（即 $N = M = 8$ ）。这一微处理器每一次只对 8 位数据进行操作。我们把这个 8 位的数据字定义为一个“字节”。8 位数据总线的简略表示如图 1-2b 所示。4 位微处理器如图

1-3 所示，一个 4 位的数据字定义为一个“半字节”。

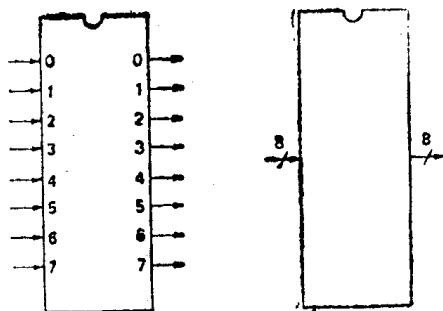


图 1-2 8 位微处理器。每个数据字包含 8 位或一个字节

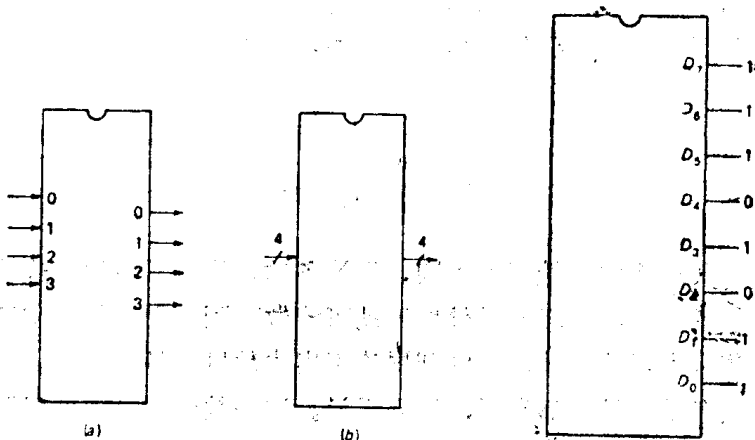


图 1-3 4 位微处理器。每个数据字包含 4 位或一个“半字节”。

图 1-4 8 位微处理器的数据总线。D₇ 是最高位 (MSB)，D₀ 是最低位 (LSB)。

任一给定时间，在微处理器数据线上的逻辑电平确定一具体的数据字。如图 1-4 所示的 8 位微处理器的情况，数据字是由 8 个二进制位数 D₀ 到 D₇ 所组成。D₀ 称为最低位。

或 LSB, D7 称为最高位或 MSB。在数据总线上的数据字, 可以用几种不同方法来表示它的数值。最简单的方法是作为一个二进制数, 用二进制来表示。用这种表示方法, 图 1-4 中所示的 8 位数据字可写成 11101011。为了表示这是一个二进制数, 我们可以用字母 B 或者下标 2 放在它的尾部, 如下所示:

11101011B 或 11101011₂

表 1-1 八进制的数

二 进 制 数			八 进 制 数
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

上述数据字也可表示为八进制数。在八进制表示中, 按照表 1-1, 可把每三位一组的二进制数指定为 0 到 7 之间的某一相应数。利用该表可找出上述二进制字的八进制等价值为:

1 1 1 0 1 0 1 1	二进制
3 5 3	八进制

353 是二进制数 11101011 的八进制等价值。为了标明这个数是用八进制表示的, 我们可以用字母 Q 或者下标 8 放在它的尾部, 如下所示:

353Q 或 353。

数据字的第三种表示方法（这是微处理器中最常用的一种方法）是把它作为十六进制数。在十六进制表示中，按照表 1-2，可把每四位一组的二进制数指定为十六进制的某一相应的字符。因此，上述二进制数的十六进制等价值可得出为：

1 1 1 0 1 0 1 1	二 进 制
E B	十 六 进 制

EB 是二进制数 11101011 和八进制数 353 的十六进制的等价值。“十六进制”一词通常可简略地用 hex 缩写来表

表 1-2 十六进制的数

二 进 制 数				十 六 进 制 数
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

示。为了标明这个数是十六进制表示的，我们可以用字母 H 或者下标 16 放在它的尾部，即：EBH 或 EB_{16} 。

实际中限制微处理器的最主要因素是：在一块经济合算的集成电路封装中，引出线的数目是有限的。为了节省所需的引出线数量，许多微处理器都把同一引出线既用作输入数据总线，又用作输出数据总数。在任一给定的时间，这些引出线或者被用作输入或者被用作输出，但它们决不能同时用于二个方面。图 1-5 中以图形的方法表示了一个 4 位微处理器，它具有双向数据总线。图 1-5(a) 表示数据总线处于输入方式，图 1-5(b) 表示该总线处于输出方式。

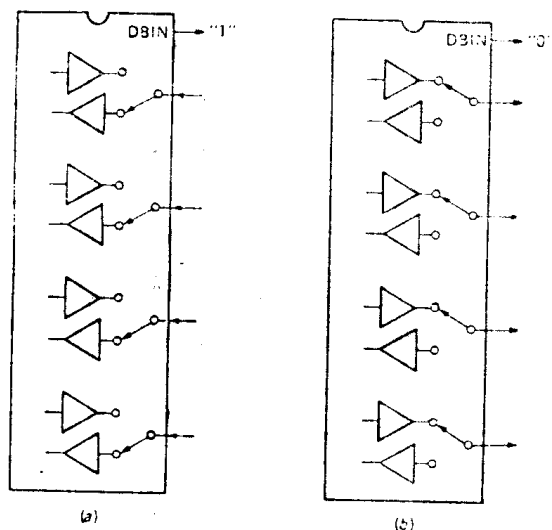


图 1-5 一个表示双向数据总线工作的 4 位微处理器。

(a) 输入方式的数据总线；(b) 输出方式的数据总线。

图 1-5 中还给出了微处理器的一个专门称为 DBIN 的

控制输出端，利用它可告诉外部电路：数据总线这时是处于输入方式还是处于输出方式。DBIN 处于高电平(逻辑1)，表明为输入方式；处于低电平(逻辑0)则表明为输出方式。

图 1-6 给出 N 位微处理器双向数据总线的表示方法。

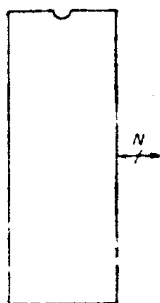


图 1-6 具有双向数据总线的 N 位微处理器。

地 址 总 线

对于理想微处理器来说，输出数据可以是在此之前的全部数据的函数，这一函数的特性取决于微处理器的程序。理想微处理器被认为内存储器的容量是无限的；而在实际的微处理器中，可用于存储数据和程序的内存储器容量却是受限制的。这样一来，微处理器就必须频繁地与外存储器打交道。一般来说，微处理器必须既能把信息存放到外存储器中，又能从外存储器中取出信息。我们把信息存放到存储器中去的过程称为写存储器，而把信息从存储器中取出的过程称为读存储器。

信息是存放在存储器的一组存储单元中，每一存储单元构成存储器的一个字。存储器的字长取决于微处理器的数据通道宽度。例如，一个 8 位的微处理器就要求每一存储单元能存放一个 8 位的数据字或者一个字节。如果是一个 4 位的微处理器就要用结构不同的存储器，使得每一存储单元能存

放一个 4 位字或者一个“半字节”。

存储器中的每一单元都有各自的存储地址，除非有另外的说明，这一地址今后都用十六进制来表示。

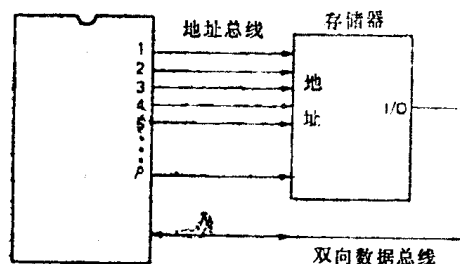


图 1-7 用于选择外部存储器中存储单元的地址总线。

在对存储单元进行读或写之前，微处理器首先应当选择好所要的存储器地址。有些微处理器是刚好在存储器的读或写操作之前，将这一地址信息输出到数据总线上。但是大多数微处理器有专门的地址总线，如图 1-7 所示。地址总线的每一条线上或者为逻辑 1 电平，或者为逻辑 0 电平。由于每条线只能有两种状态，因而一个有 P 条地址线的微处理器，它能够对 2^P 个存储单元编排各不相同的地址。表 1-3 中给出了 P 的数值从 0 到 20 这一范围内， 2^P 所具有的数。

表 1-3 2^P 的乘方

P	2^P
0	1
1	2
2	4
3	8
4	16
5	32
6	64