数字系统设计导论

赵志熙 编

中国铁道出版社

数字系统设计导论

赵志熙 编

中国铁道出版社
1985年·北京

内 容 简 介

本书以寄存器传送为核心，以符号语言——硬件程序为媒介，阐明数字系统（包括可编程序的系统）的设计方法。在硬件程序设计中引用了电子数字计算机程序（软件程序）设计的思路和技巧，以便使硬件程序和软件程序两种设计具有统一的逻辑基础。

全书共十三章大致分为六个部分。第一、二部分介绍数字系统的硬件程序设计方法。第三部分阐明由硬件程序编译成数字系统逻辑电路的方法和步骤。第四部分以小型假想的数字计算机为例介绍可编程序数字系统的设计。第五部分讨论系统之间数据传送、通信以及接口设计问题。第六部分介绍了微处理器原理及其程序设计。为设计以微处理器为核心的数字系统打下基础。

本书是高校控制类专业高年级学生选修教材，也可供从事数字系统设计人员和计算机专业师生参考。

数字系统设计导论
赵志民 编
中国铁道出版社出版
新华书店北京发行所发行
各地新华书店经售
中国铁道出版社印刷厂印
开本：787×1092毫米 1/16 印张：10 字数：229千
1985年8月 第1版 1985年8月 第1次印刷
印数：0001—6,000册 定价：2.05元

前　言

数字系统是对数字形式的信息进行加工和运算的系统总称。随着数字技术和数字计算机技术的迅速发展，数字系统将在很多领域中日益得到应用和推广。目前关于数字系统的书籍，大多是就某种系统的工作原理进行描述，而提供设计方法的尚少。编写这本书是想对数字系统的设计方法作些介绍。

从数字系统的发展进程来看，在通用的、商品化的微处理器出现之前，绝大多数数字系统是由分离元件和小规模集成电路构成的，系统的功能是由逻辑电路直接实现的。对于这类系统一般称做硬接式数字系统。在通用的微处理器出现之后，愈来愈多地较为复杂的数字系统采用了微处理器。这类系统的功能主要依赖于微处理器所能执行的指令程序来完成。这种以微处理器为基础的数字系统也可称做微处理器化的数字系统。目前这类系统正处在方兴未艾的阶段。随着大规模集成工艺的发展，当"非通用"的大规模集成电路的生产在经济上成为合理、可行时，数字系统将进入采取专用的大规模集成电路的新阶段。这类数字系统的功能将主要依靠逻辑电路直接实现。

如上所述，数字系统的功能或者由逻辑电路直接实现，或者由指令程序实现，因此数字系统的设计应包括逻辑电路的设计和指令程序设计两部分。鉴于有关程序设计书籍较多，所以在本书以较大的篇幅着重阐明数字系统逻辑电路的设计方法。为了阐述方便起见，以后将称数字系统的指令程序为软件，称逻辑电路为硬件。于是，逻辑电路的设计就是硬件设计。

在本书中，试图以较简单的系统为例，在系统的硬件设

计中以“寄存器传送”为核心，以程序（指软件）设计的原理和技巧为指导，把系统的硬件设计首先归结为“硬件程序”设计，然后再根据硬件程序翻译成系统的逻辑电路。

实际上，对数字系统逻辑电路的描述，大部分内容是关于数字形式的各种信息（以后通称为数据），如数字量，逻辑量，字符以及指令等如何在各寄存器之间传送及受到处理的。为了方便起见，以后把数据在寄存器之间的传送称做寄存器传送。对于寄存器传送，可用自然语言描述，也可用形式简单而意义确切的符号语言表达。自七十年代以来，用寄存器传送的符号语言描述数字系统的书籍愈来愈多。尽管所使用的符号语言直到今天还没有形成象FORTRAN那样公认的标准语言，但无疑地日益受到重视。用寄存器传送的符号语言阐明数字系统的设计是可能的而且是可取的。描述数字系统的完整的符号语言可称做硬件程序。硬件程序不仅能将设计思路表达清楚，而且形式简明，在设计过程中便于修改。更可贵的是能根据硬件程序比较容易地译成系统的逻辑电路。

硬件程序的书写形式和设计方法都类似于编写电子数字计算机的指令程序。对于读者来说，如果已具有程序（指软件程序）设计知识，将有助于硬件程序设计。反之，若掌握了硬件程序设计方法，也有助于软件程序设计。以统一的逻辑思维指导两种程序设计是本书的主导思路。

为了讨论微处理器化的数字系统设计问题，并使读者进一步理解同样的系统功能既可由硬件实现也可由软件实现，以便在设计数字系统时能灵活运用。本书以8080微处理器为例，简要地介绍了以微处理器构成数字系统的基本方法以及简单程序设计举例。

本书的内容大体分成六个部分共十三章。

第一部分包括一、二章。着重讨论数字系统中数据在寄

存器之间的传送和被处理过程，相应的基本逻辑电路以及与基本逻辑电路相对应的描写符号——“传送语句”，“连接语句”以及“分支语句”。这些语句是构成硬件程序的基本语句。

第二部分包括三、四两章。着重讨论数字系统的硬件程序设计方法。其中包括以存储器为主体的硬件程序。

第三部分是五、六两章。讨论由硬件程序翻译成逻辑电路的具体方法和步骤。其中包括硬接线式控制器设计方法，微程序控制器设计方法以及数据处理部分的设计方法。

第四部分包括七、八两章。以一个假想的指令很少的数字计算机——模型机为例，阐明在给定指令的情况下可编程序的数字系统的设计方法。通过这两章希望使读者能深入理解计算机的硬件与软件的内在联系，进一步掌握硬件程序设计方法。

第五部分包括九至十一章。仍以模型机为例讨论了系统与系统之间的连接和通信方式，模型机的输入输出系统，中断系统以及接口的设计方法。

第六部分包括十二、十三章。这部分讨论了以微型计算机为主体的数字系统的设计问题。考虑到对模型机的设计已有较全面的了解，对于学习微处理器已有了基础。这就可能以较少的篇幅对微型计算机原理进行介绍。这一部分以8080微型计算机为例，介绍了微型计算机的组成，指令系统以及简单的程序设计。

本书力求以通俗易懂的语言阐明数字系统的设计方法，并以若干举例说明设计过程。有意避开逻辑上的论证，因此有些地方可能不够严谨。另外，限于水平，谬误不妥之处在所难免，况且有些举例是杜撰的，可能漏洞很多。希读者批评指正。

本书在定稿时，经北京工业大学沈以清同志审阅，提出了宝贵的意见，在此谨致以诚挚的谢意。

目　录

第一章　系统概述及符号约定

第一节　数字系统概述

在数字系统领域里，“数字”一词是指信息的一种表达形式。它是由有限个多值信号或状态所组成。目前，实际应用的信号或状态只限于两个值，即 0 和 1 。以数字形式表达的信息，就其具体内容而言可以是数学中的常量和变量，可以是逻辑值。可以是字符（如 A、B、C、…），也可以是命令与指令。但它们的形式统统表现为由 0 和 1 组成的有限序列，这种数字形式的信息统称做数据。任何用来传送和处理数据的设备统称做数字系统。系统一词往往给人以大而复杂的感觉，但从上述定义出发，一个数字系统也可能是小而简单的系统。即便是一个大而复杂的系统，也可能按其功能划分为若干个子系统。对于每个子系统可以独立地进行考查和设计。

各种数字系统规模有大小，功能有简繁。所用的技术也

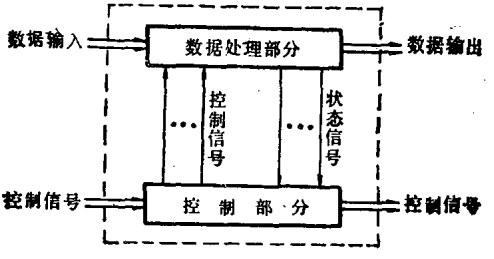

图　1—1

不一致。因此，它们的具体结构也不一样。但从功能角度来看，一个最一般的数字系统模型由两大部分组成，即数据处理部分和控制部分。如图 1 — 1 所示。在有些场合，将一个较复杂系统的输入部分和输出部分独立地划分开来，如图 1 — 2 所示。但可将这两部分看成是该系统的两个子系统。

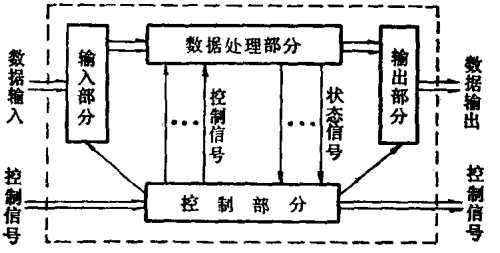

图 1 — 2

而每个子系统本身仍可用图 1 — 1 来表示。因此，图 1 — 1 是数字系统的最基本的模型。

在系统模型中，数据处理部分主要由记忆部件，功能部件（网络）以及控制门所组成。这里所说的记忆部件是指那些能保存数据的部件，如寄存器、锁存器、触发器以及存储器等。功能部件，或称功能网络，是指那些能完成算术运算，逻辑运算，数字变换等等功能的组合逻辑网络。而控制门则是指那些控制数据在各记忆部件之间传输的门电路。

一个数字系统可能对数据进行复杂的处理。但是，不管复杂程度如何，重要的是数据必须能在各记忆部件之间进行传送，特别是在寄存器之间的传送。数据是在传送过程中受到处理的。这也就是说，在寄存器之间的数据通道上设置某种功能部件，当数据经由该部件而传送时就受到了相应的处理。

一个数据由一个寄存器传到另一个寄存器的过程称做一

步或一拍。以后将要指出，在一拍中只能对数据进行一些最基本的处理。例如就算术运算而言，仅能进行一次加法运算或减法运算。虽然在一拍中也能进行一次乘法运算，但考虑到所需的硬件（电子电路）比较复杂而且昂贵，所以除了特别强调快速运算的场合外，是很少采用的。至于除法运算，一般是不容易在一拍内完成的。就逻辑运算而论，为了简化设备起见，在一拍内也仅进行一次“逻辑与”或“逻辑或”一级的运算。因此，要对诸多数据进行复杂的处理，一般总是将复杂的运算化成若干步简单的运算，并按规定的时间顺序加以执行，从而完成预定的任务。由此看来，实现数据在寄存器之间传送的电路环节是构成数据处理部分的基本环节。掌握这些基本环节是设计数字系统的基础。

既然数据需在寄存器之间经过多次传送才能完成处理任务，那么，在系统中就需包括控制部分来控制数据的传送。该控制部分通常称做顺序控制器，简称为控制器。事实上，数据在寄存器之间的传送是在电位（或电平）控制下进行的。每传送一次就需一拍电位，多次传送就需多拍电位。控制器就是产生多拍电位——电位序列的时序电路。控制器的输入信号包括两部分。一部分是来自系统外部的控制信号，另一部分是来自系统内部（数据处理部分）的状态信号。根据这些信号，控制器产生不同的电位序列。

控制器产生的电位序列除了用以控制数据传送外，根据需要也可向系统外部输出某些电位，以构成其它系统的输入控制信号。

根据以上分析可以看出，要设计一个数字系统，首先应根据给定的任务估计在系统的数据处理部分中应设置哪些记忆部件和功能部件，在这个基础上再仔细思考数据需要经过哪些寄存器传送才能实现对它的处理。这一思考过程是设计

数字系统的关键一环。如果能用一种简明而又精确的符号语言将这一思考过程编写出来，具体说，能将寄存器传送内容和先后顺序用符号语言编写出来，则是十分有用的。这种符号语言必须具有这样性质，即能根据它比较容易地翻译成数据处理部分的具体逻辑电路。这种符号语言我们称它为硬件程序。

硬件程序不仅是设计数据处理部分的依据，由于它包含了在什么条件下在什么时机需向有关控制门发送控制电位（电平）的全部信息。因此，硬件程序又是设计数字系统控制器的依据。以后将会看到，控制器可由硬件程序直接翻译而得。因此，数字系统设计的重点在于：在深入理解寄存器传送的基础上，学会硬件程序的设计。

第二节　系统的组成部件及其符号约定

数字系统是由一些基本的逻辑部件和记忆部件所组成的。对于数字系统的设计者来说，掌握这些基本部件的工作原理是必需的。但是在设计一个系统时，设计者可以把每个部件看成是具有某些性能的“暗盒”，而无需探讨它的内部。凡在用到它们的地方仅以方框和约定的符号表示出来就可以了。另外，由于篇幅的限制，所以在这一节里着重阐明对有关部件的符号表示和约定。

一、逻辑部件

从工艺角度来看，逻辑部件有DTL（二极管-三极管逻辑）、TTL（三极管-三极管逻辑）、MOS（金属-氧化物-半导体场效应管）以及ECL(发射极耦合逻辑)等集成电路。它们在工作速度，价格，功耗，尺寸以及受环境温度的影响方面各有不同。但就逻辑功能而论，它们都是接收输入

信号产生输出信号。其中输入信号是以电平表示的逻辑变量，而输出信号则是以电平表示的输入变量的逻辑函数。逻辑部件的功能就是处理输入信号和产生输出信号。而输出仅仅是输入的逻辑函数，与其内部状态无关。

最基本的逻辑部件是常用的逻辑门。在图 1—3 中给出了一些逻辑门的图形和符号。图中包括 4 个逻辑运算符：逻辑与“∧”，逻辑或“∨”，逻辑非“—”以及逻辑异或“⊕”。根据这些运算符的意义，相应的逻辑图形的意义也就清楚了。

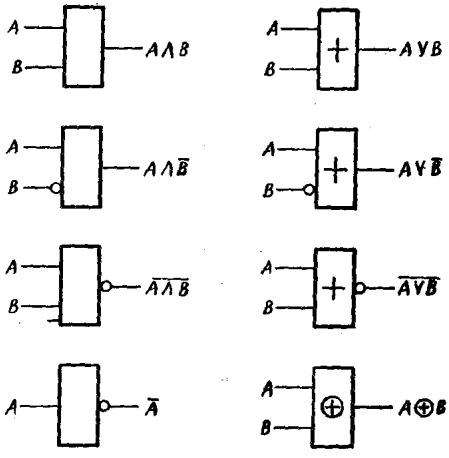

图 1—3

在实现某一数字系统时，应尽可能减少所用逻辑门的类型，以便采用定型的产品。因此，由逻辑图转换为具体电路时，要经过适当的等效逻辑变换。例如变换成主要由“与或非”部件组成的电路。但这属于电路一级的设计内容，在本书中省略了这一部分。

随着集成技术的发展，一些较复杂的组合逻辑网络，例如译码网络，编码网络，加法网络，逻辑运算网络，比较网络等等，都可集成于一块小的集成片上。因此，也可把它们看成是构成数字系统的部件。这类部件，称它为功能部件。在数字系统的逻辑图中，用一个方框来代表功能部件并在框内以文字符号注明它的功能。在图 1—4 (*a*)和(*b*)中画出了

加法器（*ADDER*）和逻辑与（*AND*）两个功能部件的图形。

在框图中，部件的输入和输出均以双线表示。这指明输入信号和输出信号是由若干码位组成的，或者说，数字信号是以并行形式传送的。为了绘图方便起见，在不致引起混淆的地方，也可用单线表示。

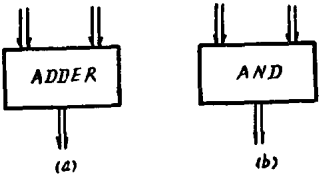

图 1—4

二、记忆部件

这里所谓的记忆部件，包括触发器，寄存器和存储器等三类部件。在实际的数字系统中，所用的触发器的类型是各式各样的。但为了把精力集中于阐明系统的一般设计方法上，在本书中以使用D触发器为主。也就是说，无论是单个触发器，还是由触发器组成的寄存器，大都采用D触发器。

D触发器有电平控制的和脉冲控制的两类。

图 1—5 是由电平（幅度）控制的D触发器的图形符号和它的输入输出时间图。其中x是数据输入端D的输入信号

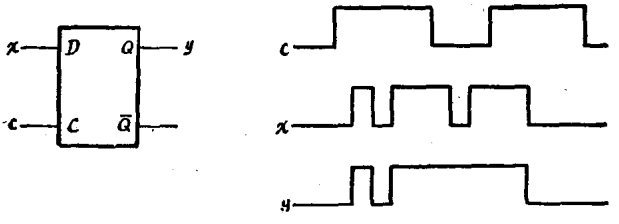

图 1—5

（此处是任意给定的）。C是时钟输入端C的控制电平。y

是触发器输出端Q的输出信号。电平控制触发器的工作特点是当控制信号C处在高电平期间时，Q端的输出电平随D端输入电平的变化而变化，即输入为高电平时，输出也为高电平，反之亦然。当控制信号处于低电平期间时，Q端的输出不再随D端输入信号而变化。在此期间Q端电平是维持在高电平还是低电平，决定于控制信号降为低电平的瞬间输入信号的电平。若此时输入信号为高电平，则Q端就维持在高电平；若输入信号为低电平，则Q端就维持低电平。一般称这类触发器为锁存触发器。

脉冲控制的D触发器有脉冲正跳变触发的和脉冲负跳变触发的两类。我国目前生产的D触发器以正跳变触发的较多。而J-K触发器一般为负跳变触发的。为了论述方便，不致在触发方式上多费笔墨，因此以后均以负跳变触发的D触发器为主。

图 1 — 6 (a)是脉冲负跳变触发的D触发器的图形符号。

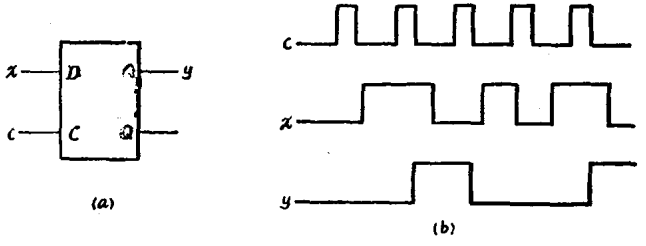

图 1 — 6

它与电平控制的触发器图形没有什么区别。但输入和输出的时间关系却不同。从图 1 — 6 (b)的时间图中可以看出(输入信号是任意给定的)，只有在时钟脉冲（C端）负跳变时，触发器状态才随D端的输入信号而改变。

无论是哪一种触发器，按一般约定，当Q端输出高电平

时，触发器是在“1”的状态，反之则为“0”状态。

在数字系统中，一般用单个触发器反映某一事件的状态。所以往往称这类触发器为状态标志或标志器。对于具有这种功能的触发器，[3]以后约定用小写英文字母或字母串给它命名。例如s，pe，$busy$等都可理解为触发器的名称。

当把若干个触发器组成一体，用以寄存数据时，则称这种触发器组为寄存器。不过，常称由电位控制的D触发器构成的寄存器为锁存器。

关于寄存器和锁存器的图形符号有时用触发器组来表示，有时用一个方框来表示。这视需要而定，这里不再一一举例了。关于寄存器的名称，约定以大写英文字母或字母串表示。例如A，AC，DR，$MASK$等等都可作为寄存器的名称。必须注意，符号AC是一个不容分割的字母串，它代表一个寄存器。若写成A，C则分别代表两个寄存器了。

在有的文献中，若以AR代表寄存器，则用（AR）表示寄存器的内容。在本书中为了简便起见，不采取这种规定。而是约定：当AR独立存在时，它代表寄存器名称；当在传送语句、连接语句或条件语句中出现时，它代表寄存器的内容。实践证明，如此处理不致混淆不清。

在数字系统中，往往需要对寄存器的一部分内容进行处理。这就需要能以符号形式简洁地表达寄存器的某一位或某几位。为此，这里约定：寄存器的最低有效位（通常为最右的一位）指定为0号位，简称为0位。其它各位由低而高的顺序号为1，2，3，…。若A为某一寄存器，以A_i代表A的第i位，以$A_{i:j}$代表A的第i位到第j位。例如有一个寄存器IR共有16位。按以上约定，其最低有效位为IR_0，最高有效位则是IR_{15}。而$IR_{3:7}$代表IR的第3位到第7位（共5位）。