

目 录

前言	i
第一章 微型计算机基础	1
1.1 微型计算机的发展概况	1
1.2 单片微型计算机的结构	3
1.3 四位、八位和十六位单片微型计算机	15
1.4 专用微型计算机	6
1.5 微型计算机的系统设计	9
1.6 未来发展趋势	16
小结	16
第二章 Motorola 公司的 MC6801 和 MC6805 系列	17
2.1 历史背景	17
2.2 MC6801 系列	19
2.3 MC6801 的软件概况	26
2.4 MC6801 的扩充模式	31
2.5 MC6801 系列的其他产品	32
2.6 MC6805 系列	34
2.7 实例分析——MC6805P2 用于汽车速度表/转速表/里程表的控制	41
2.8 未来发展趋势	47
小结	49
第三章 Texas Instruments 公司的单片微型计算机	50
3.1 一个廉价的四位单片微型计算机系列——TMS1000	50
3.2 TMS7000 八位单片微型计算机系列	60
3.3 实例分析——TMS7000 用于语音合成接口	63
小结	68
第四章 Zilog 公司的 Z8 系列	69
4.1 概述	69
4.2 Z8 单片微型计算机结构	70
4.3 寻址方式	86
4.4 指令系统	91
4.5 Z8 单片微型计算机系列	93
4.6 Z8671 单片微型计算机	94
4.7 Z8 系列的外围器件和 Z 总线	96
4.8 通用外围设备控制器	99
4.9 实例分析——巡回检测装置	102

4.10 未来器件的发展	118
小结	119
第五章 National Semiconductor 公司的 COPS400 微控制器	120
5.1 回顾与展望	120
5.2 COPS 微控制器系列	121
5.3 COPS 微控制器的结构	126
5.4 COPS 的指令系统	134
5.5 实例分析——数字式电视调谐系统	140
小结	144
第六章 Mostek 公司的十六位微型计算机——MK68200	145
6.1 概述	145
6.2 寄存器与存储器	146
6.3 I/O 特点	147
6.4 指令系统	153
6.5 实例分析——分布式控制	156
小结	160
第七章 M68 系列单片微型计算机的应用	161
7.1 EPROM 型 CMOS 单片微型计算机 MC1468705G2	161
7.2 单片微型计算机 MC6801 监控程序 LILbug 说明及程序清单	170
7.3 单片微型计算机 MC146805 E2 监控程序 CBUG05	202
7.4 单片微型计算机 MC146805 F2 L1 监控程序	227
7.5 单片微型计算机 MC146805 G2 L1 监控程序	238
7.6 单片微型计算机 MC146805 E2 在辐射监测中的应用	247
7.7 单片微型计算机 MC6801 构成低成本终端	267
7.8 单片微型计算机 MC6805 在低压抑制器中的应用	282
7.9 单片微型计算机 MC6805/MC146805 构成通用频率计数器	297
7.10 单片微型计算机 MC146805F2 在数字锁中的应用	302
7.11 单片微型计算机 MC146805 E2 在电子恒温装置中的应用	309
7.12 单片微型计算机 MC146805 G2 在自行车计算机中的应用	329
7.13 单片微型计算机 MC6805R2()1 在 A/D 变换系统中的应用及程序清单	359
7.14 单片微型计算机 MC6805 T2 L1 在锁相环系统中的应用及程序清单	395
参考文献	440

第一章 微型计算机基础

1.1 微型计算机的发展概况

本书所讨论的单片微型计算机(简称单片机)是本世纪两项最重要的发明——数字计算机和集成电路技术发展的最新产物。

数字计算机的发展从最早的机械式运算器开始,已经经历了几百年。目前普遍认为计算机发展的第一个重要里程碑是1837年 Charles Babbage 设计的“分析器”。这台机器虽然没有最后完成,但它已具备了现代计算机的许多主要特点。然而,几百年以后现代数字计算机的发展才真正开始。

机械式运算器的最早发展成果是1944年 Howard Aiken 为 IBM 设计的“马克1号”。1946年 John Manly 制造了第一台电子计算机“ENIAC”(Electronic Numerical Integrator And Calculator)。这台原始机器由18000只真空管组成,重30吨,而且速度很慢。与此同时, John Von Neumann 等人在普林斯顿设计了一台计算机——EDVAC (Electronic Discrete Variable Automatic Computer)。这台机器直到1951年才最后完成。在这台计算机中, Von Neumann 发展了 Aiken 的程序与数据分别存贮的思想,引进了一个新的重要概念:存贮程序。即将程序和数据以同样的格式存贮在机器中,这样就可以由机器本身去自动执行指令。计算机的这一功能导致了高级程序语言的产生和发展。

目前,单片机的结构具有两种类型:一种是将程序和数据存贮器分开的结构,即为 Harvard 结构(见图1.1)。另一种在很大程度上采用了通用计算机和微处理器的基本原理,对程序和数据存贮器不作逻辑上的区分,即为 Princeton 结构(见图1.2)。

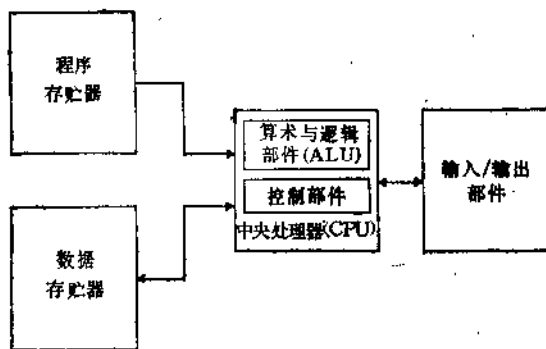


图 1.1 Harvard 结构

随着晶体管技术的出现,产生了第二代计算机(1948年)。在60年代,由于大型计算机的发展而产生了日益复杂的操作系统,使计算机进入第三代。继此之后的发展是以集成电路技术为基础的。早期集成电路的集成度较低,只有少量元件(约20个);随着制

作技术的发展,其集成度越来越高。从小规模集成电路 (SSI) 发展到中规模集成电路 (MSI) (集成了几百个晶体管)。1970 年又出现了大规模集成电路 (LSI) (集成了几千

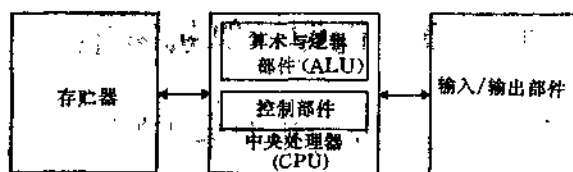
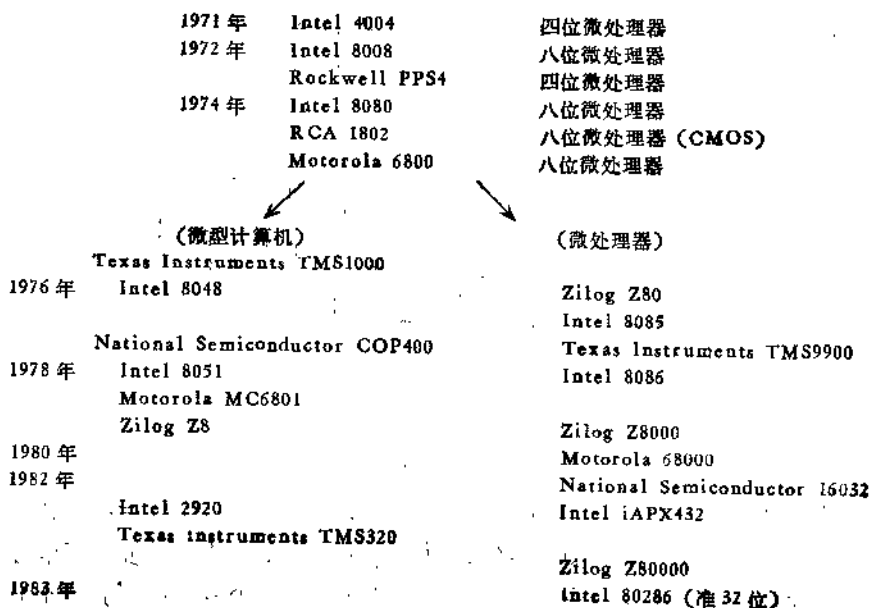


图 1.2 Princeton 结构

个晶体管)。由于集成电路的采用,使计算机进入了第四代。目前,由日本发起研究的第五代计算机,是采用超大规模集成电路 (VLSI) 和人工智能方法的新型计算机,现已成为世界范围内的研究课题。

70 年代初期出现的大规模集成电路技术,使我们能够采用少量器件来制造小型计算机。对于这一点,Marcian E'Ted'Hoff Jr 深信不疑。人们普遍认为他是微处理器的发明者,当时他正在 Intel 公司进行可编程运算器的设计,并研制出 Intel 4004 四位微处理器。这一产品及 EPROM 器件的出现,对微处理器的发展起到了关键性的作用。

1972 年 Intel 公司推出了八位微处理器 8008,随后于 1974 年又生产了 Intel 8080。1976 年 Zilog 公司也相继推出了 Z80 微处理器。这些产品目前已被广泛用于个人计算机。在同一时期内,Texas Instruments 公司生产了四位单片机 TMS1000,该产品被广泛用于日用消费品和常用设备中;通常是用于那些逻辑上可以替换的部件中,因此这种产品又称为微控制器。这类应用领域的开发标志着微型计算机和微处理器分支的开始。目前,生产厂家面临的最大问题是如何最有效地利用集成电路。在给定的技术条件下,可以根据不同情况设计出不同的计算机。这些机器有的功能较强;有的功能较弱,需要增加一些专用部件才能形成一个完整的系统。图 1.3 给出当代微型计算机的发展概况。



1984 年
1985 年

Motorola MC68010 (准 32 位)
MC68020 (准 32 位)
NS 32032 (32 位)
Intel 80386 (32 位)

图 1.3 微型计算机的发展

1.2 单片微型计算机的结构

就字面意思来说,单片机就是将计算机的所有组成部分均制作在一个芯片上。图 1.4 给出了单片机的基本结构。

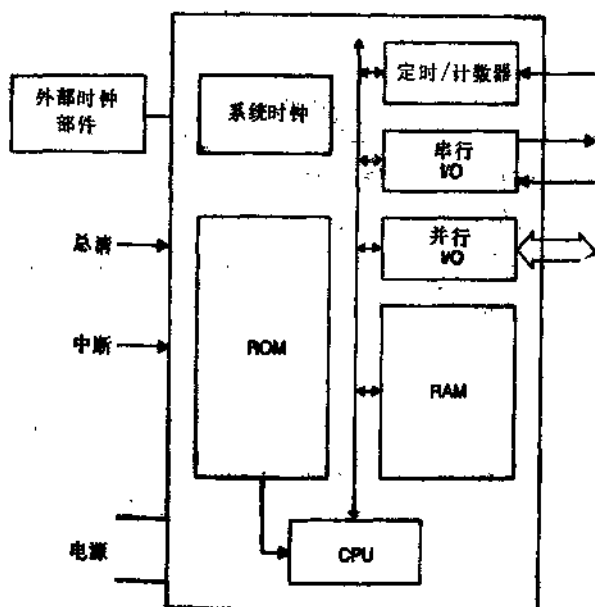


图 1.4 单片机的基本结构

1.2.1 只读存储器

只读存储器 (ROM) 是用来存放那些永久不变的应用程序的。许多微型计算机和微控制器都需要大量的应用程序,因此在生产芯片时将这些程序做到 ROM 中是非常经济的。芯片一旦做成,程序就不能更改;所以需要严格的方法并使用具有硬件仿真能力和高性能的软件开发系统来进行 ROM 代码的开发。

一些半导体生产厂家在其系列产品中为用户设有可编程空间,所以提供了附加 ROM 的选择。这类产品通常可以在微处理器模式下工作,利用 I/O 总线 和数据总线对外部存储器进行存取。尽管它们的 I/O 通道和外部电路受到一定限制,但其功能仍与单片机相似。这种器件在其他工业产品的电路中被广泛采用。这时 ROM 的开发成本虽然很高,但在 I/O 通道和其他芯片的使用问题上与通常以微处理器为基础的电路相比较,仍有很大的节省。集成电路产品的进一步发展是采用各种 EPROM 插件或直接使用

EPROM 来代替 ROM。这种 EPROM 产品比同等 ROM 产品的成本自然要高,但电路上是完全等价的。EPROM 产品在小批量应用中有着特殊的吸引力,它显示了单片器件的优越性,为用户提供了极其灵活的编程能力。

1.2.2 随机存储器

随机存储器 (RAM) 用来存放程序运行过程中的工作变量和数据,它的大小随产品类型而有所不同,但其字节宽度均与处理器的字节宽度相同(四位、八位或十六位)。某些特殊寄存器(如栈指针或定时寄存器等)在逻辑上通常被划分到 RAM 范围内。在 Harvard 结构的微型计算机中,一般将 RAM 看作是寄存器的集合。在大多数情况下,微处理器系统中的 RAM 和寄存器在物理上并没有差别,因此没有必要将它们区分开来。

1.2.3 中央处理器

单片机的中央处理器 (CPU) 与其他任何微型计算机的中央处理器很相似。由于在许多应用领域中大都包括二-十进制 (BCD) 数据的处理(如数字显示),因此一般要求中央处理器能够方便地处理这种类型的数据;同时还要求能对存储器和 I/O 端口的每一位状态进行灵活的测试、置位和清除操作。这是因为在许多控制应用中需要对各种输入/输出信号线进行读/写,而这些信号线本身在应用时连接着“两态”信号控制的设备,如开关、恒温器、继电器、阀门、马达等的接口。

1.2.4 并行输入/输出端口

在不同的微型计算机中,并行接口各不相同。一般情况下系统对并行接口中的一位或所有位均提供选择功能,以确定哪些是输入线哪些是输出线。在图 1.5 所示的 Intel 8048/8051 双向 I/O 端口内部电路图中,通过向管脚写“1”,可以使 Q2 接通而产生与

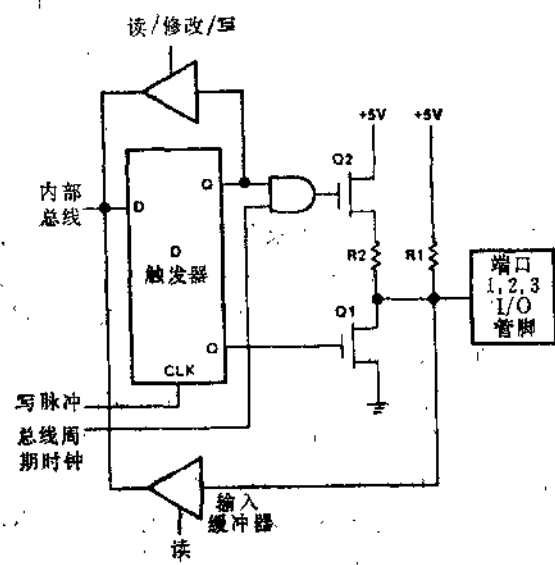


图 1.5 Intel 产品系列中采用的双向 I/O 端口电路

一个 TTL 兼容的输出高电平。该管脚如果作为输入端,则利用 R1 来进行保持;R1 大约为 $50k\Omega$,足以保证当外部信号加到管脚上时能够将其拉低。因此,若输出一个“1”使 Q1 断开,就可将该管脚作为输入端使用。

有些 I/O 线可以直接与外围设备相连(如发光显示器),还有的 I/O 接口可提供足够的电流以驱动其他设备。

有些器件要求 I/O 端口可以象系统总线那样进行设置,以便进行存储器及 I/O 端口的扩充。这种要求在产品的系列化方面很有用处,因为在系统扩充时有可能进行存储器的扩充,但不希望对现有的软件作较大的修改。

1.2.5 串行输入/输出端口

主机与终端设备的串行通信只需提供少量传输线就可以实现。这种通信可以作为专用功能器件的接口,还可以用来把几台微型计算机连接在一起。无论采用同步还是异步通信方式,都需要用规程来提供信息成帧的格式(起止形式)。这些格式可以用硬件或通用(同步)异步收发器 [U(S)ART] 来实现。因此处理器(或应用程序员)可不必进行这些低层费时的具体操作,而只需进行一些选择,如波特率和停止位的选择以及发送(接收)缓冲器的读/写操作。串行数据的处理一般都是以适当的模式由硬件来完成的。

1.2.6 计数器

在单片机的许多应用中都需要进行延时时间的精确计算。这可以通过计算程序各个分支的运行时间来实现,但除了少数简单程序外,大多数情况下其效率是不高的。最好的办法是采用计数电路,它可以独立地计时并能在预置时间到达时产生中断请求。这种计时器通常按照所要求的时间进行预置,并对所预置的数值进行递减;待减至零时,则产生中断。较高级的定时器还可以自动进行预置数的重新装入。这样,在需要连续中断的情况下(如时钟中断),就不用程序员反复对计时器进行赋值。与计时器相配合的还有一个事件计数器,一般提供一个专门的输入端直接与计数器相连。

1.2.7 定时部件

大多数微处理器中的时钟电路定时部件都很简单。如果要求机器进行最大限度地操作,那么所使用的晶体应该保证能够提供所需的最大频率,但不能超过。许多时钟电路是由电阻、电容组成的,这样可以由外部电路进行驱动,成本很低。当微处理器需要进行外部同步时,这种电路非常有用。

1.3 四位、八位和十六位单片微型计算机

微处理器和微型计算机的分类通常都是以其内部数据通道的字符宽度来进行的。字符宽度在机器操作过程中起着重要的作用。尽管目前还没有三十二位的单片机,但这只不过是时间问题,这种功能很强的设备将成为价廉物美的个人计算机以及小型商业计算机的关键部件。

微处理器中的数据通道宽度表示机器在运行时并行处理的数据位数。事实上,字符

宽度与电路的复杂程度成正比。因此,四位机的开发费用要比八位和十六位机的开发费用小得多。但四位机不能适应所有应用领域。

在逻辑上,任何设计合理的通用计算机结构,无论其数据宽度如何,对所有可计算的问题都能够处理。关键问题是所花费的时间和程序设计的复杂性不同。

四位机广泛用于计算器和小型控制器中。这种类型的单片机每次可以处理一个BCD编码的十进制数。凡是超过一位十进制数的数据必须用串行方式处理,所以按这种数字位的安排方式进行I/O处理也是很容易的。但是,只用四位码来进行地址处理操作就非常困难了。因此,这类单片机只适合于那些低档的,数据量较小而时间要求又不高的应用领域。由于四位机可满足许多应用要求,所以这类机器在微型计算机中是最多的一类,如Texas Instruments公司的四位单片机TMS1000已经生产了六千多万块。

对于许多要求较高的应用,八位机在其性能和电路使用配置方面都做到了较好的兼顾。八位字长一次可以同时处理两个BCD码,并能方便地处理通用终端上常用ASCII码的字符数据。若采用合适的算法,八位机还可以进行十六位数据的处理。

八位字长对于指令系统是很方便的,使用一个或两个字节就可以表示操作码和地址信息。因此可以对严格的Harvard结构进行修改,以使程序存储器与数据存储器彼此进行灵活的操作。

十六位微型计算机与小型计算机一样,具有通用计算能力以及规则的结构和指令系统,如Zilog公司的Z8000, Motorola公司的MC68000和National Semiconductor公司的16032。十六位字长一次可以同时处理两个ASCII码或四个BCD码。

十六位字长很适于信号数据的处理。通常每位数字位可表示的信噪比为6dB,所以八位字长最大只能表示50dB的信噪比,这对于模拟电路来说是不太合适的。虽然对大多数应用来说,十二位字长就足够了(约70dB),但仍希望采样信号有较大的范围,而且信号比例尺可连续改变。由于十六位微处理器有较宽的数据范围,所以很容易直接处理这样的数据。当然使用八位或四位微处理器也可以进行十六位或十二位数据的处理,但在时间上要增加一定的开销。这里应该注意,对双倍字长的数据处理并不是简单地对一个数据的两半分别进行处理,其中还应包括每部分数据中中间进位的处理和各种标志的管理。因此,处理双倍字长的数据比处理单字长的数据要多花费两倍的时间。

三十二位微处理器具有处理浮点数据和极大的寻址能力,还具有很强的综合处理能力。

1.4 专用微型计算机

对于某些应用领域,通用结构是无能为力的。这就导致了一类新型计算机的产生,它们的设计是针对某种具体功能进行的,其中信号处理就是这样一个领域。目前已有许多微型计算机适用于这种领域,如通信和测试系统等。

模拟信号的数字处理就是对模拟信号的数字表示进行处理。对于这种实时处理所要求的操作,通用结构尤不适用。在这种操作中,对乘法功能有着特殊的要求。例如,数字滤波器一般由信号的迭加来实现,即将上次存储的数值按一定的系数加权进行处理。因此,在两个采样时间间隔内,就需要对数据进行多次乘法处理。

1.4.1 Intel 2920 单片微型计算机

Intel 2920 是单片机发展的开端,它是由通用结构向专用器件发展的第一个转折点。在它以后确实出现了许多这样的处理器件。Intel 2920 的结构是专门为处理高速实时模拟信号而设计的。图 1.6 给出 Intel2920 的框图;图 1.7 给出其模拟部分的详细框图,它提供了四个模拟输入和八个模拟输出。通过一个九位数/模转换器的所有信号都被转接输出。图 1.8 示出 Intel 2920 的算术处理部件和数据存贮器的结构。这部分很象 Am2901 位片处理器中的 ALU 和快速存贮器部件,只不过它是二十五位字长的。

Intel 2920 的 RAM、移位器、ALU 和模拟部分的控制是按微程序方式并行实现的。实际上,Intel 2920 的基本指令就是微指令,它不需再译成更低一级的代码,所有算术运算都是以二的补码进行的(移位时不带符号位)。

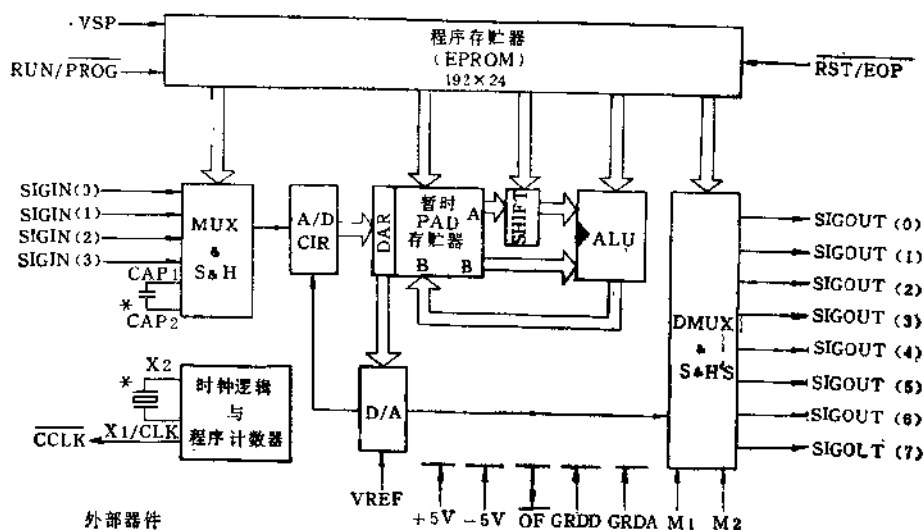


图 1.6 Intel 2920 方框图

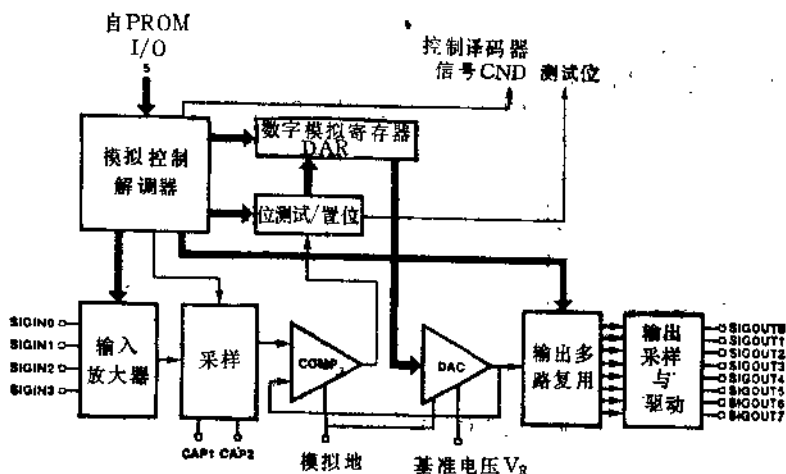


图 1.7 Intel 2920 模拟部分框图

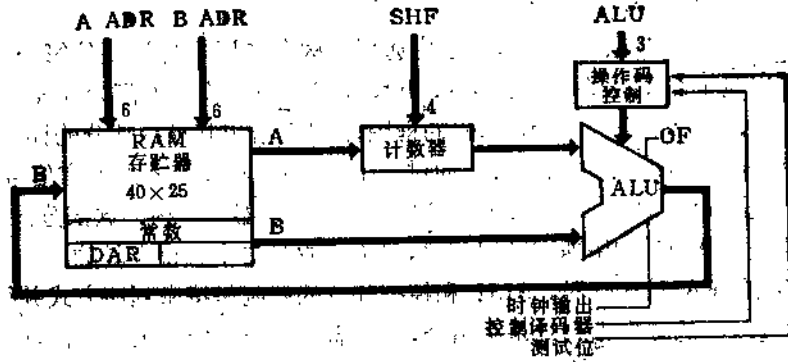


图 1.8 Intel 2920 算术处理部件与数据存储器的结构

Intel 2920 的硬件中虽然没有包括乘法,但一般的常数乘法可以通过移位迭加来实现。在进行一次移位操作时,可使操作数从高两位到低十三位任意移位,并允许无条件转移(这在信号处理中一般不需要)。一个采样周期包含 192 条指令,每条指令大约需要 400ns,因此采样频率大约为 13kHz。

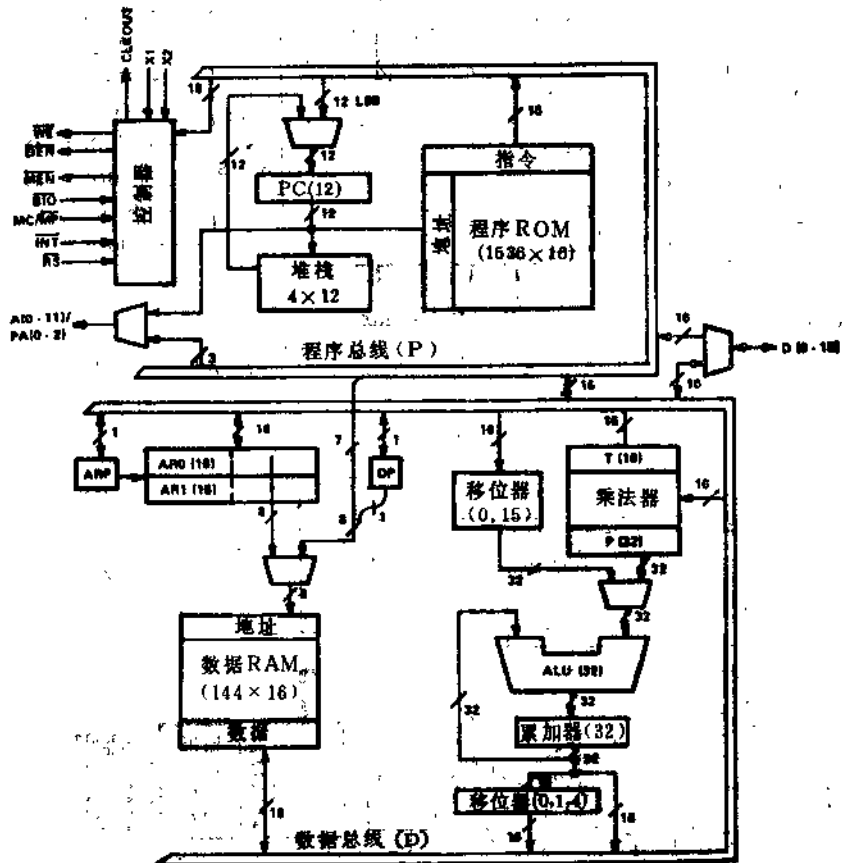


图 1.9 TMS 320 结构

Intel 2920 的某些应用领域如下:

- (1) 滤波——大约 40 个极点;
- (2) 波形发生器——外部控制的任意函数;
- (3) 低频调制解调器——AM, FM 或 PM;
- (4) 非线性方程——整流器、限流器、放大器等;
- (5) 其他——锁相环、自适应滤波器、线性电路。

1.4.2 Texas Instruments 公司的 TMS 320

TMS 320 是具有处理实时信号结构的另一种微处理器,其主要特性是有一个硬件的十六位并行乘法器和一个三十二位累加器,如图 1.9 所示。还有一个循环移位器,其功能与 Intel 2920 在进行乘法时所作的一次操作多次移位相类似。TMS 320 没有模拟信号接口,但它具有一个灵活的功能很强的指令系统,因此它适合于数字滤波、傅里叶分析、语音分析、合成以及图象处理等应用领域。

1.5 微型计算机的系统设计

1.5.1 功能描述

常言说,“三思而后行”;但我们在进行微型计算机的系统设计时,总是把自己还相当模糊的概念和想法强加给生产厂。显然,生产厂一般很清楚自己的产品可以干什么用最好,而对这种器件的应用情况用户却往往并不太了解,尤其在用户对产品有充分选择余地时,情况就更加严重。我们的产品是面向市场的,而在市场方面,用户恐怕比生产厂了解得更清楚。

因此在开始进行设计时,第一步首先要以外行人的角度提出一些问题,以确定所要求的是什。在这一阶段,对技术方面的要求不很严格,所提出的问题一般是:

- (1) 用户对机器的反应如何;
- (2) 轻便性如何;
- (3) 耐久性如何;
- (4) 价钱如何。

这些问题是由用户市场管理部门提出的,其中用户对机器的反应是至关重要的,它直接关系到产品的成败,因此有必要将典型用户的反应记录下来。由于真正关键的因素不可能无限多,所以在提交给用户的一系列方案中,只能选出一种(或两种)方案。通过这种方法可以定义出较为理想的机器。然后再考虑在工程要求的条件下对这一阶段所确定的参数进行适当的修改,但最初对产品的定义提供了一个标准。在产品最终定型时,我们有必要检查一下它是否与最初的定义相吻合。

第二步则开始定义全部功能。这些功能是根据上述用户指标的要求得到的,但必须更加精确。然后技术人员将对这些功能的定义进行分析处理,他们的工作就是一丝不苟地实现这些功能。特别是当技术人员不了解市场管理部门的情况时,更不能有丝毫偏差。在具体实现中,由于工程上的原因有可能对功能指标作一些修改,但这种变动必须是微小的,并要以文字的形式记录下来附在功能指标之后。在微型计算机系统设计中,功能指标

所应包括的项目如下:

(1) 环境条件:

- 1) 工作温度*;
- 2) 存放温度;
- 3) 湿度和耐腐蚀性;
- 4) 震动;
- 5) 用户方面的使用情况。

(2) 必要的安全标准*。

(3) 可维护性/模块性。

(4) 用户软件接口的全部功能。

(5) 所有硬件接口的定义。

(6) 显示器的尺寸、颜色、亮度。

(7) 键盘的尺寸。

(8) 体积和重量*。

(9) 电池寿命/耗电量*。

(10) 故障形式。

(11) 样机要求:

- 1) 形状;
- 2) 功能;
- 3) 形状和功能。

(12) 最高价钱。

(13) 其他有助于产品改进的意见(这可能影响最初的规定)。

上述标有*号的部分是特别重要的项目, 这些项目关系到产品的成败, 其中有些要求直到设计周期的最后阶段也未必能找到一种折衷方案来满足。比如, 要求系统在高于100℃的环境下运行, 这对所有微型计算机系统来说是不可能的。

对生产厂将要开始的设计工作应该进行很好的可行性研究。这项研究的目的是证明生产出的产品是否能真正达到设计所要求的功能(生产厂不愿花费大量人力物力去生产一个只在理论上战胜竞争对手的产品)。在可行性研究所得的结论中, 制定的功能指标是很严格的, 技术人员对项目的认可则表示这是更高一级的确认。但在这一阶段仍可能做某些让步, 因为稍一疏忽就会影响到项目的生命力。这时先不用去考虑价格问题, 因为半导体生产厂设计产品的最终目的是为了出售, 而且他们最关心的也是对那些还没有在功能指标中作出任何规定的参数, 并希望保证有一个合理的数值范围。

1.5.2 定制部件的方法

考虑定制掩膜部件的原因不外乎是成本问题。花费极少的资金并采用简单的电路图对生产厂无疑是很有吸引力的, 然而在许多情况下生产总成本并没有多大变化, 问题是生产成本变得越来越不直观。那么如何估计这部分隐含的成本, 进而确定是否定制掩膜部件就成为要认真考虑的问题。

在进行所有电器产品的设计时, 除了上述技术指标的讨论外, 还应考虑一些市场情

况,即所要求产品的数量,要使产品的成本除以售出产品的数量达到最小。下面举一个加热控制器的例子,如果只生产一种这样的产品,那么无论怎样生产,其开发费用也将肯定超过产品部件的成本。这时的最好办法是以单板形式销售并采用高级语言来进行程序设计。然而,若要大批量生产,则开发费用就显得微不足道了。这时产品部件的成本就是主要因素。正是在这方面,单片机才显示出其真正的优越性。如果最初假设出售两万个产品,那么两万个以上的产品就有利可图了,但定制产品部件仍不可避免地存在着危险。

资金流动

从广义来说,开始选择掩膜 ROM 的投资是很值得的。在产品大量生产出来之后,这部分投资就可收回。但在第一个样机生产出来之前是必须进行投资的。这笔钱由两部分组成:一是用户方面的,如代码的开发、仿真和调试;二是生产厂方面的,如生产产品时所消耗的材料和人力。一般生产厂在用户拿到样机时就需支付这部分费用,而不管用户是否有批量的要求。

最小批量

生产厂在接到程序代码之后,便生产出一批样机。从这批样机开始就能生产出成千上万个芯片。生产厂一般将这批样机的费用都记在用户的帐上(不过集成电路的大部分成本是在封装上面),而且在接到批量订货之前,一般用户只希望把尽量多的电路封装在样片中。一旦接到审定的结果,厂方就可以很容易地进行大批量的封装生产。从这时起投资才真正开始发挥经济效益,这也是为什么生产厂一定要求最小批量订货的缘故。

定型样机

采用定制部件的动机之一就是要减少部件的数量。在一般情况下,可重编程的 EPROM 部件要么需要大量的器件(如分离 EPROM);要么采用昂贵的分段封装,将 EPROM 做在单片机芯片的上部,两个罗列起来。这两种情况都需要增加许多费用来添置那些仿真最终产品所需的单元部件。另外,可编程方案(EPROM)比定制部件方案要消耗更多的电能,因此还可能要求附带一个电池包(或电缆)。所有这一切都说明,先进行小批量生产,检查一下市场反应,然后再决定是转人生产掩膜成品,还是重新进行设计,这些工作都不是轻而易举的(但在多数情况下不是不可能的)。

固定性

正如上面讨论过的那样,从工程生产的角度来说,要使产品的设计随着市场的变化而进行修改是很困难的,但同时又引出了许多很好的想法,比如用掩膜来固定程序。可这是相当冒险的,因为这意味着把许多没有完全试验过的新方法付诸实践。一旦将程序做进硅片,那么软件中存在的微小错误就可能是致命的。解决的办法只能是重新做,或增加一个硬件装置(成功的可能性很小,即使可能也要增加成本)。提供完全固定的 ROM 代码,软件编制人员必须保证代码的精确性(这对低档的微型计算机机器码来说是相当困难的),硬件设计人员也必须加倍注意。尤其在时间要求很短的情况下更要注意(硬件仿真的时间一般很短,有时甚至当软件要进行掩膜处理时硬件仍在设计中),问题发现得越晚,

所花的代价就越大。

这些看起来也许令人担忧，这里有必要提醒读者：定制掩膜微型计算机代表了一个完全通用的器件，它可以完成成千上万个晶体管的功能，其价格却只相当于五、六个 TTL 器件。一个真正设计良好的微型计算机可以完成所有必要的功能，并保证使所有投资都发挥效益。因此，在今后相当长的一段时间内，单片机一定会继续保持着优势。

1.5.3 微型计算机的最终设计任务

在决定要进行单片机的设计之后，下一步的任务就是选择一种特定类型，使所有的设计要求都达到最便宜的标准。设计者不仅需要研究竞争对手的想法，而且必须从那些与对手明显不同的角度来评价处理器的性能。在进行设计时，单片机的处理能力通常是需要考虑的因素，但不是主要因素。下面给出在评价一个给定产品时应考虑的一些因素。

输入和输出部件

当希望降低产品的成本而要去掉一些部件时，首先想到的是输入/输出部分。因为输入/输出部分不仅随着管脚的增加而使组件成本提高，而且还会增加一些测试工作。因此管脚越少，产品的成本就越低。要保证设计产品的成本最低，首先应确定输入/输出管脚的最低需要量。在本书的其他部分将谈到，在微型计算机中以分别密接的连接方式工作时就不要求有自己的输入端，除非输入端上的信号变化非常快。同样，我们也可以利用视觉的滞留特性对输出管脚按多路复用方式来驱动显示器。许多输入事件可以由轮转程序进行处理，对那些要求快速响应的输入事件（小于 500ns），有必要提供中断输入端以中断微型计算机的一般工作。但为了节省管脚，有些器件也没有提供。

如果需要增加外设微处理器芯片来完成系统功能，就必须提供总线接口。但这需要增加相当数量的管脚，因此许多单片机不提供总线接口。有些器件即使作为选择功能而提供总线，也不可避免地增加了一定数量的管脚。因此，有时甚至明显要求设置总线的器件，在仔细设计后也可以取消。比如在 PMOS 微型计算机中，需要一个小规模的静态存储（128 位）。如果使用常规的 CMOS 128 × 1 的 ROM，则需要七条地址线，还要加上数据输入、数据输出和读/写线；如果采用移位寄存器来代替真正的随机存储器，那么只需要数据输入、数据输出和时钟三条线就足够了。

代码尺寸

单片机的系列产品随着 I/O、RAM 和 ROM 数量的不同而各异，当然价格也随之不同。由于系统的扩充或许多新功能的增加，系统程序的大小往往会超出原来的 1024 字节。这就需要再增加一个存储器，这样就有可能使产品的成本增加 50%。在这种情况下，增加的成本太高。如果基本系统程序本身就需要 1K 多的空间，那么索性做成 2K 字节的存储器，这样就很容易增加功能。就硅片的工艺水平来说，RAM 的制作比 ROM 的开销要大得多。因此，单片机一般只提供少量的 RAM 空间。另外，增加功能还会带来其他许多变化，所以在考虑单片机的容量时，RAM 空间也要考虑在内。因此，在打算增加新的功能时，必须进行总体考虑，看其是否合算。

处理能力

一般说来,微控制器的应用不受其输入/输出速率的限制。

实际上,现实中的处理速度比微型计算机的代码执行速度要慢得多。然而仍有两点需要注意,一是“除以 n ”的问题。假设要求进行音乐音阶方波的输出,一般是采用快速计数器对预置的值进行计数(通常是递减)而产生输出,并不断重复这一过程。在比率 100:106 的频率处存在一个半音阶(接近 50:53);也就是说,对于一个给定的音阶和一个低半音的音阶,其处理过程是,一个循环必须比另一个循环快 100 倍左右(即每个周期执行两遍循环)。对于一个 1kHz 的音阶,递减跳转过程必须在 10ns 内完成,这对于大部分单片机来说是不可能的。因此,在实际应用中,高音区通常允许有一定的“自由度”。

另一个是有关中断的问题。在某时刻,有些情况不允许中断,而另一些情况却需要中断。例如在多路复用条件下进行显示时,就不希望进行其他操作,否则屏幕就会闪烁不定(产生闪烁是由于滞留时间,或由于切换频率)。如果必须进行其他操作时,那么必须使操作时间尽量缩短,否则显示就得不到足够的时间,使屏幕变得昏暗。多位复用驱动的数字越多(发光二极管显示),问题就越严重。如果没有大峰值电流显示器驱动(一般采用昂贵的驱动晶体管)来保证平均电流达到合理水平,那么当数字超过十二个以后,问题就根本无法解决。

实时程序

许多应用系统都要求微型计算机能够自动计时,或能够在指定的时间间隔内做某件事。但这又会产生一些问题。当微型计算机执行代码时,一般在不同的环境下执行不同的分支程序。例如加热控制器,当房间温度足够时执行一个分支程序,而当需要增加温度时又执行另一个分支程序。一般情况下,这两部分程序的执行时间是不同的。如果程序每单位时间内处理一次时钟,那么在某个分支程序内就有可能使时钟丢失或增加。时间越长,这一误差就会越大。一种解决的办法是将时间短的分支程序延长,使两个分支程序执行的时间相等。但在实际应用中,由于分支程序很多,所以这种方法是很不方便的。计时的要求还会产生另一些问题。如果要求微型计算机在不同的温度或电池能量等情况下都要保持恒定不变,就要求使用晶体或陶瓷振荡器来取代廉价的电阻电容网络。

更好的办法是使程序的执行速度快于需要,然后等待外部事件去处理时钟。主驱动设备经常采用这种方法,它使用一个主频率作为时钟源。如果采用内部时钟,则要求使用更精确的晶体作为微型计算机的时钟源。

耗电量

有些单片机的应用要求很低的能耗。一是希望系统能长时间运行而不需更换主电源;二是因为系统本身就很很小(如袖珍式装置)。根据前面所述,自然会想到使用 CMOS 器件。然而 CMOS 器件在运行时的耗电量并不是很低的(但低于 PMOS),一般采用“断电”模式来进一步降低能耗。除了 RAM 中的程序等工作单元和时钟加电以外,芯片上的大部分电源都被切断。通过外部事件(如按键)或定时器的中断(隔一段时间后),可以再将机器“唤醒”。要使平均能耗达到最低,重要的是将“唤醒”处理过程尽量缩短。另外,

CMOS 的耗电量与系统时钟频率成正比。因此,在保证正确操作的前提下,应尽可能地降低系统的工作频率。同时 CMOS 器件可以在很大的电压范围内工作,而耗电量又与所提供的电压成正比。所以对电流的消耗以及电池的尺寸都要给予重视。例如在电子表中,操作电压只有 1.5V;因此需要特殊的制造技术才能满足这一要求。一般单片机的操作电压在 3V 以下的很少。

1.5.4 选择最合适的技术

PMOS 工艺的生产技术已有十二年的历史(到 1984 年止)。尽管有比它更好的工艺(如 NMOS 和最近出现的 CMOS 工艺)在性能方面已经取代了它,但 PMOS 并没有被淘汰。它仍占有工艺要求最简单,价格最低廉和成品率最高的优势,并尤其适用于四位机和那些对速度要求不高的应用领域,如语音合成系统等。目前,采用先进的工艺,可以使 PMOS 器件在很大电压范围内和很小的电流环境下工作,因此能用 9V 电池来直接驱动。也可以在其输出端上加高电压来驱动电子管荧光显示器。

NMOS 工艺十年前就已成熟,当时的工艺技术已能生产 n 沟道晶体管。由于 n 沟道中的电子漂移速度比 p 沟道中的空穴漂移速度快得多,而限制器件速度的主要原因是芯片上的走线电容和栅极电容的充放电时间,所以采用 n 沟道技术可以提高速度。不过这是以增加能耗作为代价的。因为 NMOS 器件可以提供用于存储器 and 微处理器所需逻辑与 TTL 兼容的 5V 电源,所以大多数八位微处理器都采用 NMOS 器件。随着工艺技术的不断改进,已经能将 ROM, RAM 和 I/O 设备做到芯片中,而所生产的器件同样具有速度高的特点。但由于能耗的增加,使用电池进行工作就很困难。

CMOS 技术是理论上首先提出的半导体技术,但直到能够生产 NMOS 之后,CMOS 的生产才成为可能。由于早期的 CMOS 需要大面积的硅片,所以使其应用范围限制在小规模集成电路逻辑中(即 RCA4000 CMOS 逻辑)。随着 CMOS 技术及掩膜质量和处理工艺的改进,近几年已可以完全采用 CMOS 来进行微型计算机的设计。从广义来讲,这些产品扩充了已有的四位机和八位机系列,使得生产厂和用户的开发费用都有所下降。CMOS 的特点是具有大范围的工作电压和很小的耗电量。它的第一个优点是由其电路的对称特性决定的。CMOS 的输入门槛电压是采用总电压的某个百分比,而不是绝对电压。第二个优点是由于在电源线路之间采用了两个互补晶体管。在任一时刻只有一个是开通的(不包括从一个状态向另一个状态的切换瞬间),这一点不同于 PMOS 和 NMOS(这里

表 1.1 制造技术的比较

相对数的比较内容	PMOS (金属栅)	NMOS (硅栅) (耗尽负载)	CMOS
工艺步骤	1	1.2	1.8
时钟速度	1	10	(2 个金属栅)
耗电量	10	50	1
成本	1	1.3	2.0
工作电压范围	好	差	很好
$(V_{最大} - V_{最小}) / V_{有效}$	30%	10—20%	75%
显示驱动	发光二极管真空荧光显示器	发光二极管	液晶显示器

两个都是开通的)。因此,当机器处于某个逻辑状态时,只有一个电流流过。即便在有大电源供电的系统中,CMOS也显示出优越性:因为它经常处在“断电”或“暂停”状态,所以至少在再次来电前,对 RAM 工作区的保护省电。表 1.1 给出这三种器件的主要特性比较。

1.5.5 ROM 代码的开发和交货周期

在用户提出最终的掩膜 ROM 软件之后,第一步(也是最重要的一步)就是复制一个软件的副本。以便由于生产线的变化或掩膜容量的改变而不得不重新组装器件时,使保留的数据仍然有效。

然后将软件的目标码版本转换到“栅极布局板”程序中。这是一个标准格式,它用来表示做到最终产品 ROM 中的程序结构,而且很大程度上独立于具体的器件。通过对输入到“栅极布局”程序的数据进行处理,计算出软件所需栅极的位置并产生相应的几何数据。最后将这些几何数据与在器件的非程序区中固定的几何位置合并,做出最后的掩膜。

实际上,由定义的那些场效应晶体管栅极掩膜的几何位置来产生程序。那些晶体管的栅极氧化层很薄,能在一定的逻辑信号下开通。而那些没有定义的晶体管则具有很厚的场氧化层,它在任何正常情况下都不会开通。但在“低门槛电压”技术中,光有这层氧化膜是不够的。对那些不需要的晶体管,还要再进行一次离子注入处理,以保证它永远不会导通。这部分附加的掩膜处理工作是自动进行的,但与软件的处理过程一样,因此自然要增加一定的成本开销。

人们一般认为,这种用户定做的软件就是一个制作掩膜的处理阶段。其实不然,这种工艺处理在相当早的切片(直径 100mm)生产过程中就已进行了。而其他的掩膜处理,如双极 ROM 的处理,一般在工艺过程的“接触”阶段进行,这时芯片的生产已接近完成。因此,通过生产大量半成品的切片来提高样机的交货速度是不可取的。

提高首批样机交货速度的最佳方法是,在“结构生成”磁带交到掩膜工序的同时将切片放入生产线。也就是说,在需要对切片进行编程时,掩膜好的产品正好到达生产线。为了加速样机的生产,只需生产小批量的切片。然而,这部分切片也要分两批进行工艺处理,以防由于设备的故障或操作错误而报废全部材料。

在处理切片的同时,“自动测试程序”也开始进行。利用最初的“栅极分布程序”产生一个测试程序,对处理过的切片进行“探针”测试。在产品装配完成之后,还要进行器件的最终测试。所有器件产品都要经过这两种测试。在上述过程结束之后,还要对最终产品进行抽样测试(探针测试和最终测试),以确保产品质量。

下一步将完成的切片进行封装,以提供最终定做的样品。一般样品均采用陶瓷封装,虽然成本很高,但对于小批量的样品生产还是很方便的。目前采用的塑料封装自动生产线对于小批量的生产(小于 1000 个器件)是不合算的。

最后,对封装好的器件再进行一次最终测试就可以交给用户了。由于在最初程序到生产出实际产品之间采用了高度的自动化工艺,所以产品可以马上投入应用,最多是对某些元件值作些微小的调整(这些值是从在软件开发时用来仿真微型计算机功能的硬件设备中得到的)。