



NP5WCLQ

位片式微处理器

严志华 王明馨 编译

南京大学出版社

位片式微处理器

严志华 王明馨 编译

南京大学出版社

1991·南京

内 容 简 介

本书内容主要介绍Am2900系列芯片的功能、ALU的演变、控制器的工作原理与组合方式。在第四章中介绍了微程序的设计方法和一个简单系统实例——一个供学习编写微程序用的学习型位片式微处理器系统。最后还简单介绍了一个用Am2900系列芯片组合的9080微处理器的模拟器的硬件方案及个别机器级指令的微程序实例。

本书可供大专院校师生、工程技术人员了解位片式微处理器的工作原理与系统设计方法。对于已经了解微处理器工作原理的人员,想进一步了解微处理器内部工作原理,本书对他们有一定的帮助。

位 片 式 微 处 理 器

严志华 王明馨 编译

南京大学出版社出版
(南京大学校内)

江苏省新华书店发行 武进第三印刷厂印刷
开 本: 787×1092 1/16 字 数: 184千
印 张: 7.5 印 数: 1—1500
1991年1月第1版 1991年1月第1次印刷

ISBN 7-305-00736-6

TP·24

定价: 4.35元

前 言

本书是根据对研究生讲课的讲义编写的。1987年夏,在太原山西大学举行的综合大学无线电专业教师计算机课程研讨班上作了介绍。内容编译自《Bit-Slice Design: Controllers and ALUs》、《AMDASM™ Reference Manual》、《An Emulation of the Am9080 A》及《Advanced Micro Devices Bipolar Microprocessor Logic and Interface Data Book》等书。

位片式微处理器是微处理器中的一个分支系列,它有别于固定指令系统式的微处理器和单片式微处理器,是一种处理位数可以由用户自己自由组合的系统。对Am2901或Am2903等芯片,可以组成4的整数倍位数,而系统的指令是由用户按需要以微指令方式来编制组合而成,可以固化在PROM或EPROM中,微指令的位数由对系统的处理速率、功能来确定。功能强和/或速率高,微指令的位数要多些;反之少些。

第四章中的Am2900系列学习机是徐州师范学院张安京同志在攻读硕士研究生学位,学习本课程时所做一个设计安装实验,系统能够正常运行。

微处理器的生产技术飞速发展,新的器件不断产生,位片器件也不例外。现在已经有32位的位片器件被开发出来的报道,本书中尚未编入这些内容。但作为教材,它对掌握微处理器的工作原理及位片式微处理器系统的设计方法,还是有它的作用。

由于编者水平有限,错误在所难免,恳请读者批评指正。

编 者

1988年12月于南京大学

目 录

前言	
第一章 绪论	(1)
1-1 处理器系统实施方案的选择	(1)
1-2 微程序设计	(2)
1-3 LSI(大规模 集成电路)的优点	(3)
1-4 Am2900系列	(4)
1-5 语言的相互关系	(5)
1-6 控制器的设计	(7)
第二章 ALU的演变	(11)
2-1 指令格式	(11)
2-2 控制单元功能	(11)
2-3 PC和MAR	(12)
2-4 改进ALU的速率	(13)
2-5 附加的灵活性	(16)
2-6 Am2901	(19)
第三章 简单控制器	(21)
3-1 顺序执行	(21)
3-2 多路时序	(22)
3-3 起始地址	(23)
3-4 映象可编程的只读存储器	(23)
3-5 无条件转移	(24)
3-6 条件转移	(25)
3-7 时间 考虑	(27)
3-8 子程序	(33)
3-9 嵌套子程序	(34)
3-10 Am2900 /11	(35)
3-11 堆栈容量	(36)
3-12 微程序存储器	(36)
第四章 微程序设计及学习型位片式微处理器举例	(39)
4-1 微指令的分类	(39)
4-2 微指令的编译法	(39)
4-3 微程序汇编语言	(41)
4-4 字符	(41)
4-5 项目	(41)
4-6 定义文件	(42)

4-7	定义语句	(43)
4-8	定义字	(43)
4-9	名字	(44)
4-10	字段	(44)
4-11	注释内容	(46)
4-12	汇编文件	(47)
4-13	汇编文件语句	(47)
4-14	程序计数器控制语句	(49)
4-15	常数定义语句	(49)
4-16	执行语句	(49)
4-17	自变量	(51)
4-18	页面与相对寻址	(54)
4-19	十六进制数的属性	(55)
4-20	Am2900系列学习机	(56)
第五章	控制器的改进	(63)
5-1	测试范围的扩充	(63)
5-2	Am29811	(64)
5-3	CASE语句(n通道转移控制单元: Am29803)	(72)
第六章	控制器的进一步改进	(74)
6-1	状态查询	(74)
6-2	中断服务	(74)
6-3	实施方案	(75)
6-4	Am2910	(77)
6-5	控制线	(80)
6-6	中断处理	(81)
6-7	Am2914	(82)
6-8	Am2914 的互连	(85)
6-9	机器级中断	(87)
第七章	算术与逻辑单元及其基本运算的改进	(88)
7-1	ALU的进一步增强	(88)
7-2	指令字段	(90)
7-3	指令系统的扩展	(90)
7-4	操作实例	(91)
7-5	算术运算的一般讨论	(94)
7-6	Am2901的乘法	(98)
7-7	Am2903的乘法	(100)
第八章	系统的连接	(101)
8-1	对Am2903便笺式寄存器的扩充	(101)
8-2	MUX 组合电路	(101)
8-3	状态寄存器	(103)
8-4	Am2904	(103)
8-5	一个Am9080A的模拟器	(106)

第一章 绪 论

逻辑电路的组合,在60年代中期采用的是小规模集成电路,有与非门、或非门、异或门和与非门等。在70年代初开始则采用中规模集成电路,有寄存器、译码器、多路转接器等及其他器件。70年代末以后,就有带片上暂存器、中断控制器、微程序时序器、只读存储器/可编程只读存储器的运算器和其它大规模集成电路,包括完整的单片式微处理器(控制器、运算器和寄存器)相应产生。在这几年来,逻辑电路的设计者们又使用了积木式电路。

小规模集成电路大约有2~10个门,中规模集成电路有20~100个门,大规模集成电路有200~1000个门,超大规模集成电路的门数就更多,例如Z8000微处理器就有17500个门。

1-1 处理器系统实施方案的选择

目前设计者在实施方案中有三种可选择的基本方式:(1)中小规模逻辑电路方案;(2)八位、十六位或更多位的MOS固定指令系统(FIS)微处理器方案;(3)可编微程序的位片式结构。

1-1-1 处理器系统的体系结构

任何固定指令系统的MOS微处理器都有它自己预先确定的内部体系结构,这就限制了设计的随意性。如果用MOS器件组成的体系结构能够满足要求,这是可行的。一个中小规模集成电路组成的结构允许设计者按预先确定的精确细节来完成体系结构。对于位片式微处理器而言,具有一定的限制。但系统的大部分结构,留给使用者通过选择指令及微程序来确定的。

1-1-2 容量

由于空间的限制,印刷电路板尺寸或板间距离(机器空间等)是常为设计者所关心的。对于一个典型的处理器单元,一个固定指令系统的MOS微处理器可以有2~6个芯片,对于用中、小规模集成电路器件的相同系统,要有100~500个芯片,但对于一个位片式则是折中容量,约在30~60个芯片。

1-1-3 字长

不管是哪种计算机、控制器、简单处理器,对于一个系统来说,字长是必须确定的工作。固定指令系统MOS微处理器的字长与设计目的是一致的,MOS器件有四位、八位或十六位或更多位字长,用中、小规模组成的系统可以有任何位字长,位片式微处理器允许有宽阔的变化范围。以2900系列为例,它所允许的字长在四位的倍数上来选择,当位片式微处理器不能方便地匹配时,能用中、小规模集成电路“插入”基本位片式结构来补充。

1-1-4 指令系统

在设计时,实施方案的选择对系统的指令会产生重大的影响。软件方面的投资通常远

远超过硬件的投资，它们的投资比率高于十比一左右。为了节省投资，考虑软件兼容性成为设计的首要目标，新的设计必须提供一个指令系统，FIS MOS微处理器有一个固定的指令系统，如果MOS微处理器的指令系统能满足设计系统的要求，就能采用一个以MOS微处理器为基础的系统。当前固定指令系统微处理器提供的是汇编语言，软件提供的为BASIC、PL/I、FORTRAN、PASCAL以及COBOL等语言，那么就需要以原有的指令系统为依据，再编写一个附加软件通过具体的MOS器件来执行。

中、小规模集成电路的设计能提供任何预期的指令系统，位片式设计也能提供任何预期指令系统的微程序设计。这两种处理方式的主要差别在于前者是由专用的硬件结构组成，但后者则由硬件和固件组成。

1-1-5 各种不同选择的比较

各种设计方案的比较在表1-1中摘要表示。当设计要求有高速率或长的字长等要求的系统时，固定指令系统的MOS器件就不能用。如果在设计时有元器件的数量或印刷电路板的尺寸限制，或者产品的体积不能满足中小规模集成电路的方案，这时最好的选择就是采用位片式微处理器的方案。对一个微程序化的位片式微处理器来说，设计方案的改进和变化，通常用改变PROM或修改可改写控制存储器来完成，用这种方法比用中、小规模集成电路芯片的组合逻辑电路的设计要容易。

表1-1 设计方案的比较

	中小规模集成电路	位片式器件	固定指令系统微处理器
体系结构	任何要求	伪灵活	预先设计
芯片数	500	50	2~6
字长	任何要求	2或4的倍数	4, 8, 16, 32
指令系统	可以用网络满足任何要求	可以用微程序满足任何要求	指令已确定，不能改
速率	100~200ns	100~200ns	1~2μs
设计时间	慢	快	快
排除故障	困难	开发系统辅助处理	开发系统辅助处理
文件编制	冗长乏味	通过微程序实现	软件是主要部分
改进	需要重新设计	容易做到，能预先规划	改换芯片型号
价格	最高	中等	最低

位片式器件可用于以下三个基本区域：具有长的字长的机器，带有特殊指令系统的机器和高速机器，最好是用于小体积但规格特殊的信号微处理器，它既需要高的速率又要长的字长，可变指令系统的小型计算机也发展到用位片式微处理器，它允许围绕一固定硬件结构，由用户按要求的指令系统编制微程序。

1-2 微程序设计

微程序设计是把硬件设计转变为软件设计。例如一个双极性肖特基TTL型机器是由位片式或中、小规模集成电路建成的，那么它的控制就必须是微程序的，这是目前公认为最好的方法，其原因有以下几点：

第一, 由存储器(可改写控制存储器, ROM, PROM或相当的器件)代替随机时序逻辑电路, 它可以使得在设计方面趋向更为合理的组织结构。

第二, 如果设备需要改进, 可以由现场工程师更换适当的PROM来完成, 这比更换硬件要容易得多, 而增加新的元器件到密集的印刷电路板上, 是很容易产生连接错误的。

第三, 对一个原始的设计能通过一个或几个PROM来简单地实现变动(改变微程序)。例如能按预先计划好的愿望, 在版本A的基础上简单地增加一个或两个PROM来改进为版本B, 这可简化生产过程。原始的单元应该是一个元器件安排得稀疏的印刷电路板, 这是用来为改进作准备的。在这种情况下, 可以用简单地增加PROM(或者根据需要可改用其它器件)来扩大系统。这种技术通常也可用在RAM的扩充上, 以定义文件和汇编源文件编制的微程序适用于固定形式的文件, 它与模块设计相配合, 微程序控制的应用可更清晰地地为文件编制提供了一个较好的机会。

最后, 诊断程序能包含在由最后系统所提供的PROM中, 并且能由现场工程师通过一个测试板调用和执行调试。有些诊断程序能以微程序方式被编入另一个系统中, 它们在系统正常运转时, 被例行地执行。对于更严格的试验, 能用一个特殊的测试存储器来简单地代替一个普通的PROM存储器。

1-3 LSI(大规模集成电路)的优点

假如由于速率上要求, 而选用双极性器件代替MOS器件, 有许多理由可以认为大规模集成电路比SSI/MSI要好。

首先, 成本低, LSI只需要较少的元器件, 因此LSI只需要较少插件和较少的机器空间, 也就会减少腐蚀机会, 以及较少的引出线之间的连接。这样, LSI就可以把更多的连接点转到芯片的内部去。

其次, 用LSI可以提高可靠性, 工作系统的故障大约有80%是由于印刷电路板腐蚀形成的损坏或是引出线的弯曲以及其它外部连接损坏等因素造成的。如果用SSI/MSI器件, 一个典型的控制器需要有300个16引出线的双列直插式器件, 总计要有4800个引出线。改用LSI的同样控制器只需要30个40引出线的双列直插式器件, 总计只有1200个引出线, 而其它连接点则可移至芯片里面。

本书准备介绍2900系列器件, 设计规则要求每一个设计必须用工业标准零件, Am2900系列是符合工业标准的双极性位片式器件, 这是一个大规模集成电路级组合的可编程微程序系列, 表1-2概括了它的优点。

表1-2 可编程微程序大规模集成电路的优点

更高级的组织结构
字段变换——可以象更换PROM那样简单
适配——可以象更换PROM那样简单
扩展——预先计划, 可以象加若干个PROM一样简单
较好的文件的编制
硬件与固件可以平行设计
大规模集成电路, 有较少的元件
大规模集成电路, 有较好的可靠性
诊断PROM能辅助调试、维护

1-4 Am2900系列

器件内容包括下述几方面：(1) 中央处理器——运算器以及便笺式寄存器部件：Am2901, Am2903及Am29203；(2) 微程序时序器及控制器：Am2909/Am2911 和 Am2910；(3) 双极型存储器：各种器件包括误差检测及校正控制器和支持器件 (Am2960系列)；(4) 中断控制器和支持器件：Am2914, Am2913及Am2902；(5) 总线 I/O：Am2950及支持器件；(6) DMA支持器件：Am2940及Am2942；(7) 由可编程微程序微循环支持的时序器：Am2925；(8) 主存储器程序控制器：Am2932；(9) 新的十六位的Am29116。

考虑一个基本计算机的简单方框，如图 1-1 所示。图内基本单元有：(1) CPU(中央处理器)包括 ALU 与便笺式寄存器，PC (程序计数器) 和 MAR (存储器地址寄存器)；(2) 主存储器，其中存放有效程序与数据；(3) 外围电路，包括备份存储器，输入输出接口；(4) 计算机控制单元，它还可以检查每一事件并且包括控制逻辑指令译码器和PROM，在CPU中处理数据和处理指令。

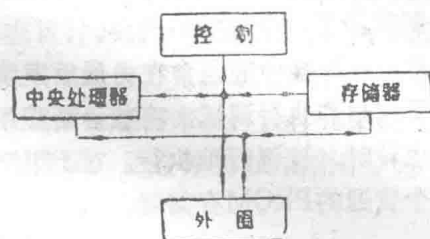


图1-1 简单系统方框图

从以上的简单叙述，我们可以归纳到图 1-2 所示的进展。一般通用计算机结构简图指出2900系列各种不同器件和它们的应用。

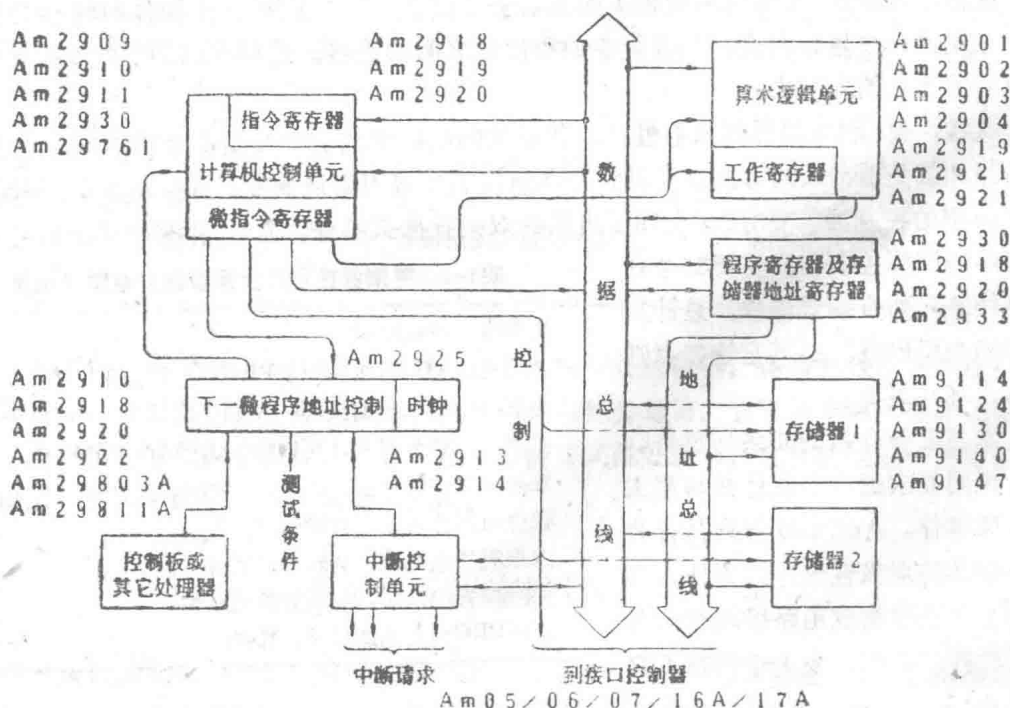


图1-2 通用计算机体系结构

1-5 语言的相互关系

程序编制的种类与源码有关——由使用者写出的某种编程语言，通过一个汇编或编译程序转为机器级的目标代码，这是机器级可执行的指令。编译程序是一种把高级语言程序翻译到目标代码的软件程序，而汇编程序是一种把汇编级源程序翻译到目标代码的软件程序，每一种计算机系统通常有好几个编译程序和一个汇编程序。

编译程序翻译的平均比率为4比1(每一个源指令有4个机器指令[FORTRAN IV]，或更高一些[FORTRAN VI, 6比1]，汇编程序几乎能够1比1地翻译，汇编语言是一种助记符形式的位模式机器指令，所有上述情况都由软件产生软件的方法来处理。如图1-3所示。

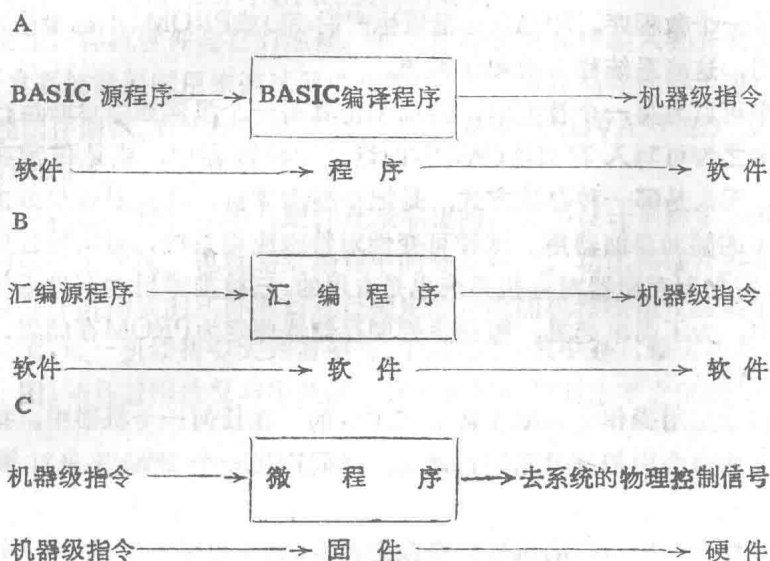


图1-3 语言关系

(A) 高级语言, (B) 汇编语言, (C) 微语言

1-5-1 高级语言

高级语言是一种完全自由格式，并且是一种有一些栏目安排上限制的语言，也就是它们在代码形式上用的是伪英语助记符，并有预先写的功能，它们的性能包括数值、循环、转移及子程序，着重于结构式的程序设计工具。例如IF—THEN—ELSE、CASE及PROCEDURE等语句。

1-5-2 汇编语言

汇编语言具有更加约束的格式，它需要一个精确的数据定义，并且可以通过编程器把程序装入存储器，又可用很多助记符式的指令，大部分指令语句是限制在一种操作上，因此翻译比率几乎是1比1。在通常情况下，汇编级程序设计员比用FORTRAN或BASIC的程序设计员要知道更多机器方面的使用知识。

1-5-3 机器级语言

机器级语言在软件级语言系统中是最紧密的，它们通常把指令、数据以及地址写成八进制或十六进制码，而实质上就是二进制码。它比汇编级语言需要更冗长的结构与调试过

程,也需要更严格的格式。它们需要程序设计员知道更具体的细节,这就决定于系统被编程序的复杂性。

1-5-4 微程序编制

机器级指令是由计算机控制单元(CCU)接收。在一个用微程序编程的机器中,每一机器指令(称为宏指令)被译码和对一个对应的微程序进行寻址,它以适当的顺序把所需要的物理控制讯号送到系统的其余部分,这使通过固件使微程序的软件指令转换为硬件控制活动。

1-5-5 微程序存储器

各种软件程序每时每刻都在很快地变化,而它们的数据也在不断改变,所以需要RAM作为它们的存储区,然而微程序一直可以保持不变。有一些机器可对几种不同的应用语言分别装入不同的微程序。

当系统中用了一个微程序,ROM(大批量生产时用)或PROM(小批量生产时用)都是用来存放微程序的,这些系统称为微程序系统。

当一个微程序可以为另一个替换时,例如去仿真另一个机器或微诊断运行时,这类微程序是存在一个称之为可写入存储器(WCS)的读/写存储器中,或是存放在系统主存储器中的某一部分。无论是哪一种存放方式,是把小型磁盘机、盒式磁带机或主存储器的一个区域作为微程序的辅助存储器用。这种可变性对特殊应用系统,像实验控制之类系统是有吸引力的,可改写控制存储器对样机系统也是有用的,已被推荐用在开发系统中。在贯穿本书的各个部分中,为了简单起见,微程序控制存储器假定为PROM存储器。

1-5-6 指令格式

每一个机器指令是用操作码和操作数形式组成的。在任何一个机器中,指令可以有几种不同格式,这些指令由控制单元进行译码,译码产生一个地址用来对微程序存储器读取微程序。

微程序是用来执行一个特定的机器指令或宏指令,它可以是一个或多个微指令组成(一个微指令假定执行一个微周期,这样的微指令称为微步),每一微指令的信息包含在各个不同字段内。在系统中,每一微指令字段指向或控制系统中的一个或多个特殊硬件单元。每一次,一个特定机器指令执行一个相同的微程序,有效微程序的特定的顺序构成特定程序的执行过程。

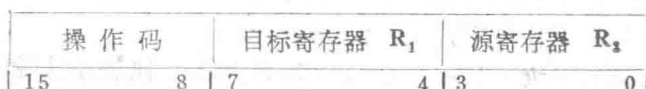
机器语言与硬件非常紧密,有非常严格的格式,它不用助记符,要求每一信息由程序设计员指定,包括程序地址和数据地址。一种机器指令格式的例子见图1-4A所示,在给定系统中通常存在很多机器指令格式。

微程序编制的格式是由程序员设计,一经选择,它就被固定下来。一个字段控制一个或几个特定的硬件单元,每一个字段位数的多少由各个被控制单元的所需要的硬件讯号来决定。简单而短的微程序能以位串方式记录,而样机的PROM由操作的PROM写入器产生。一个微指令格式的实例如图1-4B所示,多个微指令可以组成一个微程序。

1-5-7 开发系统

较长的微程序(微程序长度 >32 微字,而微字宽度大于16位)由开发系统处理较好,这些系统允许每一字段由助记符来规定,它是文件编制的助手(用标号例如ON、OFF表示比用0、1更接近于人类语言)。一旦这字段被确定,在处理中微代码(微程序、微指令)就能写

A 机器指令



B 微程序指令

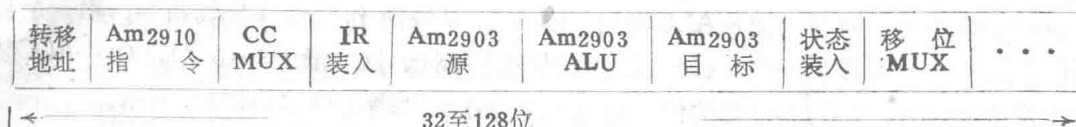


图1-4 格式举例 (A) 机器指令(寄存器寻址); (B) 微程序指令(Am2900系列)
(CC, 条件码测试; MUX, 多路开关; IR, 指令寄存器)

成助记符, 或多或少像一个伪汇编语言, 提供一个人们可读的文件编制, 于是开发系统可以用来编写微程序并且送入到自动PROM写入器。

开发系统允许样机硬件与它们连接, 并且把样机微程序装入到开发系统的可改写控制存储器中, 开发系统能同时用来调试硬件与固件, 可改写控制存储器, 可以用来代替样机系统的微程序控制存储器。对于2900系列而言, 开发系统为ADVANCED MICRO-COMPUTER的AMSYS29™。

微程序设计在程序设计级中与硬件是最靠拢的, 程序设计员必须知道被控制硬件中的每一部分中的每一个情况。也就是对编程中的分支方式、功能和硬件控制之间的相互变换是了解的。微程序所需要的实际数目是已确定的系统的机器级指令的函数, 对一般水平的计算机控制系统, 每一机器指令大约需要有4个微指令, 最少是1, 最大是16, 根据需要可以有明显的不同。图1-5是编程情况与所提供的每一种语言控制水平之间的相互关系。

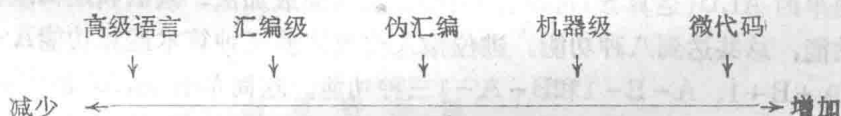


图1-5 语言间的关系

要求: (1) 硬件的详细知识; (2) 写程序时间; (3) 硬件与功能的控制

1-6 控制器的设计

这里介绍的计算机控制单元(CCU)的设计是应用于任何控制器的设计入门。

1-6-1 简单计算机

如果对图1-1的简单计算机作更进一步的简化, 则如图1-6所示, 其中: (1) CCU, (2) ALU及便笺式寄存器, (3) μ PC(程序计数器)和(4) MAR(存储器地址寄存器)。

存储器地址寄存器(MAR)是为外围设备和存储器寻址而设置的寄存器, 它的输出接至地址总线, CCU(计算机控制器)控制图中所示部分并且通过控制总线输出到外围设备,

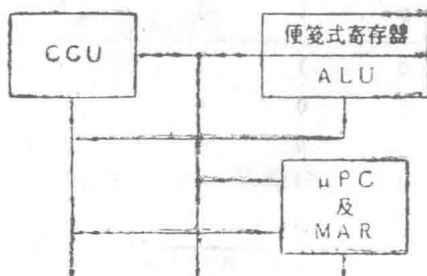


图1-6 计算机控制方框图

ALU(算术逻辑单元)通过数据总线接收并输出数据,地址可以通过数据总线装入μPC(程序计数器),这是一个十分基本的系统。

这系统的更进一步的简化如图1-7所示,这里是由CCU控制ALU和ACC(累加器,寄存器),而略去了其它的控制功能。这里输入只能通过ALU的通道A装入,通道B是通过ACC寄存器装入的,数据只能从ACC输出。CCU以某种操作码形式接收指令,用一个操作码作为输入,CCU必须产生:(1)适当的功能控制讯号(一般为3~6位),(2)ACC装入允许控制(两位),(3)进位位的值。对这一基本单元的微字格式如图1-8所示。

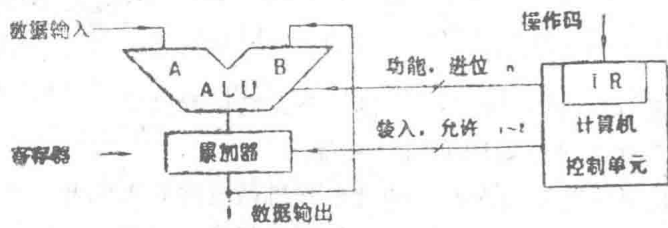


图1-7 简单时序系统(简单计算器SIMCOM)

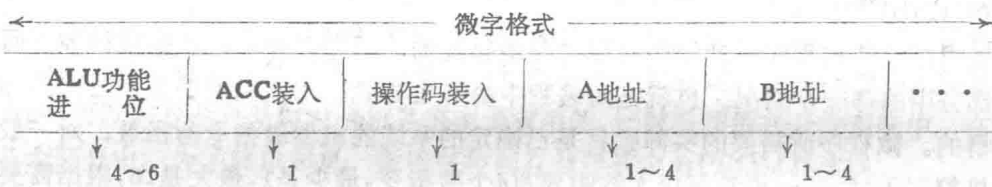


图1-8 CCU控制要求

一个简单的ALU(运算器)可以有3根控制线以完成加法、减法和逻辑或、与、异或以及相似的功能,总共达到八种功能。进位输入位允许使三种算术运算功能 $A+B$ 、 $A-B$ 和 $B-A$ 变为 $A+B+1$ 、 $A-B-1$ 和 $B-A-1$ 三种功能。这简单的ALU是支持如表1-3所示的机器级和汇编级指令加、减、或以及异或的。

表1-3 ALU功能

			ALU 功 能	
S_2	S_1	S_0	$C_{in}=0$	$C_{in}=1$
0	0	0	$A+B$	$A+B+1$
0	0	1	$B-A-1$	$B-A$
0	1	0	$A-B-1$	$A-B$
0	1	1	$A \vee B$	
1	0	0	$A \wedge B$	
1	0	1	$\overline{A \wedge B}$	
1	1	0	$A \vee \overline{B}$	
1	1	1	$\overline{A \vee B}$	

1-6-2 CCU的构造

CCU的工作是对接收到的操作码译码,而按规定对ALU-ACC进行控制,CCU的构

造是怎样的呢？有硬件方式或微程序方式两种。

• 硬件方式

如前所述，CCU 是由传统的时序电路网络构成，由一个SSI/MSI硬件单元组成。这一途径有如下优点：

- (1) 它是一习惯的设计；
- (2) 它可以是最低限度的(非冗长的，可试验的)方式；
- (3) 如果设计保持不变，这是合适的；
- (4) 它可以是一个最高速率的工具；
- (5) 在设计前确定高容量是可能的。

有如下缺点：

- (1) 有一漫长的设计时间；
- (2) 由于容量影响，文件编制难于产生和维护；
- (3) 全面地缩简掉多余部分是困难的；
- (4) 如果不缩简多余部分，又将干扰可调试性；
- (5) 改变设计要求部分或全部重新设计；
- (6) 当有竞争或冒险时，调试就困难；
- (7) 插件板大；
- (8) 连接点数量多(外部连接)；
- (9) 由于模块性，所以结构通常不再描述。

硬件控制是由指令寄存器(IR)、译码逻辑、时钟发生器网络和一个复杂的时序组合网络组成。网络的输出将成为系统其余部分的控制讯号，如图1-9所示。

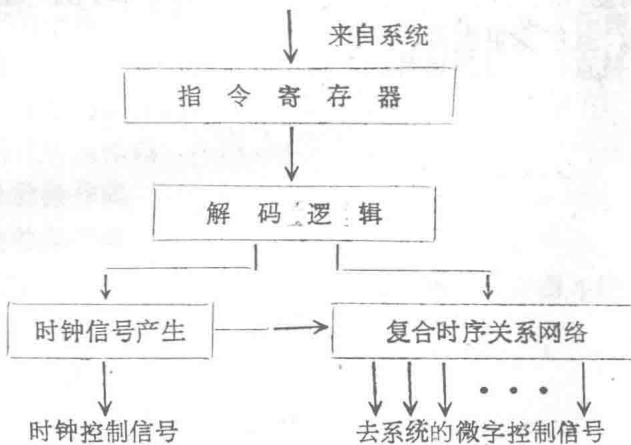


图1-9 典型的硬件CCU(SSI/MSI)

• 微程序设计方式

假使所需要的速率不能采用MOS固定指令系统的微处理器，则控制就能用微程序设计来实现。

最简单的微程序计算机控制单元需要有指令寄存器、译码逻辑、时钟源和一个以ROM或PROM组成的控制存储器。在控制器输出中，将包含有系统的其余部分的控制讯号。如果有一个不同于CCU的控制单元开发，那么这单元就能像一个寄存器、一个时钟源及一

个由PROM组成的控制存储器那样简单，如图1-10所示。

微程序控制单元结构有如下优点：

(1) 它运用于较高级设计；
(2) 微程序的编制允许有条理地达到设计要求；

(3) 它是一个紧凑的、积木式的(模块化的)物理单元(相当于SSI/MSI)；

(4) 它是一个灵活设计(微程序设计单元能完成不同控制功能和这些功能的不同变化，大多数情况下，不影响物理硬件)；

(5) 如果把一个适当结构的程序编制的方法用于产生微程序，对于系统运行的文件编制来说，这是一个比只用硬件的系统为好的系统；

(6) 诊断程序能以微程序方式设计进存储器本身或设计进用于故障寻找系统的特殊PROM；

(7) 它比SSI/MSI的设计时间短。

通过开发系统的辅助设计就能实现上述工作，因为微代码与硬件在一般情况下能同时开展，所以微程序设计也就能同时由二、三人同时合作进行工作。

• 成本核算

从发展来看，对于复杂的设计，把微程序用在控制单元设计方法上是有经济价值的，计算机PDP-11系列是这种趋向的一个较好的例子。

为了比较成本核算，设计了硬件逻辑的PDP-11/20，又设计了微程序化的PDP-11/60。虽然它们总体的“功能”(设计的复杂性)是差不多相同的，但PDP-11/60比PDP-11/20具有更多的控制功能，且成本更经济。

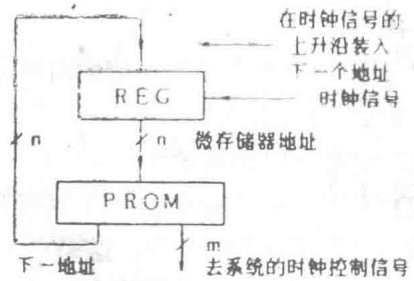


图1-10 最简单的微程序控制方式

第二章 ALU 的演变

前面简单地介绍了处理器的工作原理，关于CCU结构的详细讨论在后面再作叙述，在这一章中先讨论ALU的结构。

2-1 指令格式

如图 2-1 所示的 ALU 有：(1) 一个从 CCU 来的进位输入端和功能控制端的 ALU，(2) 一个来自 CCU 的装入允许控制端的 ACC(累加器)，这样一个简单系统能够支持 ADD, SUB, OR, EXOR, 累加器装入或通过等功能。假如数据只来自 ALU 的 A 通道，则指令有如下格式：

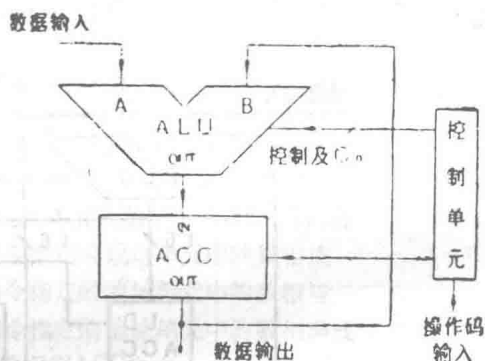


图2-1 简单系统

操 作 码	操 作 地 址
-------	---------

简单地址指令

这是多种方式中的一种，可以发生在多种不同的系统之间或任何一个系统中。

在一个系统中的指令通常有多种格式，其中几种较典型的是直接寻址：

- 带有隐含操作数的操作码
- 带有单一地址的操作码
- 带有两地址的操作码
- 带有多地址的操作码

直接寻址、变址寄存器寻址、间接寻址、相对寻址、寄存器寻址和这些寻址的组合这里就不提了。

2-2 控制单元功能

作为一个计算机控制器，CCU必须能取操作码到指令寄存器中去，并且要求判别指令是否包含地址部分。如果是包含的，则地址必须装入MAR(存储器地址寄存器)，那里就是用来从存储器取真正的数据的。在要讨论的简单系统中，操作码装入IR(指令寄存器)，同时地址则装入MAR。

如果需要从存储器中取出数据，操作码的译码将产生所需要的微指令去执行实际的取数。当所有的操作码都有效，ALU的功能继续执行。在要讨论的简单系统中，第二个操作数的取得像ALU的执行一样要产生在同一微周期内。