

微型机 结构分析与模拟技术

叶梅龙 编著



北京理工大学出版社

1130
YML/1

微型机结构分析 与模拟技术

叶梅龙 编著



北京工业学院出版社

内 容 简 介

本书是专门用于超大规模集成电路和微处理器设计、解剖以及分析的寄存器传输级模拟技术。它以Z80机为对象,阐明AHPL语言对整机性寄存器级模拟的源描述书写方法和模拟器AHPLSIM86的使用方法。第一章介绍Z80机的组成和寄存器级硬件描述语言AHPL的语法;第二章根据Z80机指令的特点讨论其分类方法,以便于书写源描述;第三章给出各指令类的描述示例;最后一章介绍模拟器的使用方法。

本书可供从事超大规模集成电路和计算机设计自动化工作的科技人员参考,尤其适用于VLSI和微处理器解剖、分析的模拟。对于分析和模拟数字通讯、数字控制和数字检测等系统是同样适用的。本书也可以作为高等学校计算机工程专业高年级学生和研究生教学参考书。

微型机结构分析与模拟技术

叶梅龙 编著

•

北京工业学院出版社出版

新华书店北京发行所发行 各地新华书店经售

通县向阳印刷厂印刷

850×1168毫米 32开本 9印张 233千字

1987年10月第一版 1987年10月第一次印刷

印数: 1-5000册

统一书号: 15434.77 平装定价: 2.10元

精装定价: 5.10元

前 言

现代超大规模集成电路(VLSI)单个芯片上可容纳几千个,乃至上万个门,它们能够独立地构成一个相当规模的数字逻辑系统,例如单片CPU。因此,更先进的VLSI设计自动化系统应是自上而下地从寄存器传输级开始。这里首要的问题是运用寄存器级硬件描述语言对所设计的数字逻辑系统的基本组成、结构及其实现的功能进行准确地描述,然后通过模拟严格验证该描述所定义的功能是否已达到预期的要求,其中是否还存在设计中的疏忽和错误。而所谓寄存器传输级模拟是以数字系统或计算机指令的操作原理为基础,通过检查寄存器、存储器 and 计数器等部件中和引线、总线上数据传输的逻辑关系,达到验证指令操作,乃至整个系统工作正确性的目的。显然,这是从功能验证和分析出发考察数字系统的工作过程,并借以完成性能评价和功能测试码的生成等。模拟过程中输入、输出和内部传输的信息都是成组的数据,它们表现出“引线上的信息”或“寄存器中的内容”随时钟或节拍的变化而进行传输、运算和变化的情况,借以刻画数字系统的动作过程。因此,寄存器传输级模拟是验证数字系统的设计和解剖、分析超大规模集成电路或微处理器芯片的得力工具,它不但可用于补救编排指令操作表的疏忽,为交付生产的批准、验收提供依据,同时也是先进的VLSI和计算机设计自动化系统必不可少的一部分。

我们应用建立在PRIME 550机上的寄存器传输级模拟器AHPLSIM85,成功地完成了Z80微处理机的寄存器传输级模拟。使我们对Z80机各条指令的节拍安排、变址操作,ALU的工作特点,CPU的内总线结构以及特殊指令的实现方式等都有了更清楚的了解。本书就是详述这件工作,力图阐明以Z80机为典型的微机指令的操作流程。通过对Z80机的描述和模拟,阐明AHPL语言的描述方法及模拟器AHPLSIM85的使用方法。第

章介绍Z80机的组成以及AHPL硬述件描述语言的语法结构
1等；第二章从Z80机指令的特点出发，讨论如何将指令进行分类
目的是便于书写AHPL语言的源描述；第三章列举出各类指令
的源描述示例，便于读者更好地掌握AHPL语言，以书写被模
拟对象的源描述；第四章介绍该模拟器的命令文件的书写。同时
也讨论了出错信息的处理以及各类指令模拟结果的分析方法。最
后给出在“虚拟机器”上运行几个典型汇编程序的示例。

显然，任何模拟模型和实际的模拟对象之间总存在不同程度的
差异。这可能是由于建立模型总要有些假定的、理想化的条件
带来的，或者是由于对被模拟对象还不是完全清楚。本书所讨论
的对Z80机的模拟，必然有些内容也是猜测的，或者是虚拟的。
所以本书所提供的对Z80机AHPL语言的描述不可能和实物是
完全对应的。

此外须特别强调，掌握了Z80机的模拟，对于进一步解剖、
分析十六位机、三十二位机以及其它中、小型计算机是有直接参
考意义的。而作为讨论基本分析和模拟方法，以八位的Z80机为
例则更为简明扼要。并且这个方法对于分析和模拟数字通讯、数
字控制和数字检测等系统的数字部件的工作是同样适用的。

本书可供从事超大规模集成电路和数字计算机设计自动化工
作的同志参考；也可作高等院校计算机专业高年级学生和研究生
的教学参考书。

北京邮电学院沈树雍教授在百忙中审阅了本书的原稿，提出
了宝贵意见，改进和提高了原稿的质量。王占宁老师也阅读了原
稿，提出了很多可贵意见。参加Z80机模拟工作的有杨淮，马苏
拉、吴红梅、王林等同志。关于本书的撰写，刘明业同志曾提过
许多意见，并审阅了初稿。对上述各位同志，作者仅此一并深
谢。

由于本人水平所限，谬误在所难免。恳请批评指正。

叶梅龙

1985年初冬

目 录

第一章 微型机结构及其AHPL语言描述

§1.1 Z80 机的组成和结构	(1)
1. 总线与寄存器结构	(2)
2. AHPL 语言语法结构	(6)
§1.2 Z80CPU模块的组成	(10)
1. 说明部分	(10)
2. 标准函数单元	(14)
3. 时钟	(16)
§1.3 CONTROL 模块的组成	(16)
1. 控制矩阵	(16)
2. 说明部分	(19)
§1.4 MEMORYS模块的组成	(19)

第二章 Z80微机指令分类和译码

§2.1 指令编码的特点	(22)
§2.2 "000XX111"类	(26)
§2.3 "CB"类	(26)
§2.4 "ED"类	(27)
§2.5 "00XXX10X"、"1XXXX110" 及"10XXXXXX"类	(32)
§2.6 其余指令	(35)
1. "00"指令组	(37)
2. "01"指令组	(38)
3. "11"指令组	(38)

第三章 Z80微机操作控制的描述方法

§3.1 AHPL 语言的控制序列	(43)
-------------------	--------

1. 控制序列	(43)
2. 连接和传输	(46)
3. 模块描述的举例	(47)
§3.2 指令操作的分类	(58)
§3.3 移位与位操作指令类的描述	(69)
1. “CB”类指令	(69)
2. “000XX111”指令类	(80)
§3.4 “ED”指令类的描述	(88)
1. 数据块传送和输入指令	(88)
2. 数据块检索与输出指令	(94)
3. 自动输入和十六位加减指令	(101)
4. 自动输出和十六位寄存器与内存单元之间的传送 指令	(109)
5. 取补指令	(114)
6. 中断返回指令	(116)
7. 中断指令	(118)
8. 特殊传送指令和四位成组交换指令	(119)
9. 累加器A与I/O设备间的I/O指令	(126)
§3.5 八位算术逻辑指令类的描述	(128)
§3.6 交换、调用、返回及重新启动指令类的描述	(141)

第四章 指令的模拟方法

§4.1 命令文件的书写	(151)
1. 选择段	(151)
2. 时钟范围	(152)
3. 外部引线赋值及置初值	(152)
4. 输出、转贮及抑制	(153)
§4.2 书写AHPL源文件注意事项	(144)
§4.3 调试中的出错信息及其处理	(155)

§4.4 各类指令模拟结果的分析	(160)
1. 移位与位操作指令类举例.....	(160)
2. “ED”指令类举例.....	(162)
3. 八位算术逻辑指令类举例.....	(163)
4. 交换、调用、返回及重新启动指令类举例.....	(163)
§4.5 汇编程序的调试	(164)
§4.6 模拟器 AHPLSIM85的操作方法.....	(165)
附录 I Z80机指令系统表.....	(171)
附录 II Z80全机源描述.....	(176)
附录 III (a) 移位与位操作指令模拟示例	(248)
(b) “ED”指令模拟示例.....	(252)
(c) 八位算术逻辑指令模拟示例	(256)
(d) 交换、调用指令模拟示例	(260)
(e) 弹栈、压栈指令模拟示例	(266)
附录 IV 十六位数据和汇编程序的执行	(270)

第一章 微型机结构及其 AHPL语言描述

本章首先扼要讨论 Z80 机的组成和结构。然后用 AHPL 语言分别对其三个模块中的组成元素的名称、性质和规模进行说明,为以后引用这些元素提供确切的定义。在说明 Z80 机组成的同时将介绍 AHPL 语言的语法结构、各种说明语句的运用及标准函数单元的调用等。

§1.1 Z80机的组成和结构

本书的工作是以分析 Z80 机 CPU 芯片的结果所得到的硬件结构图(图 1.1)为依据的。

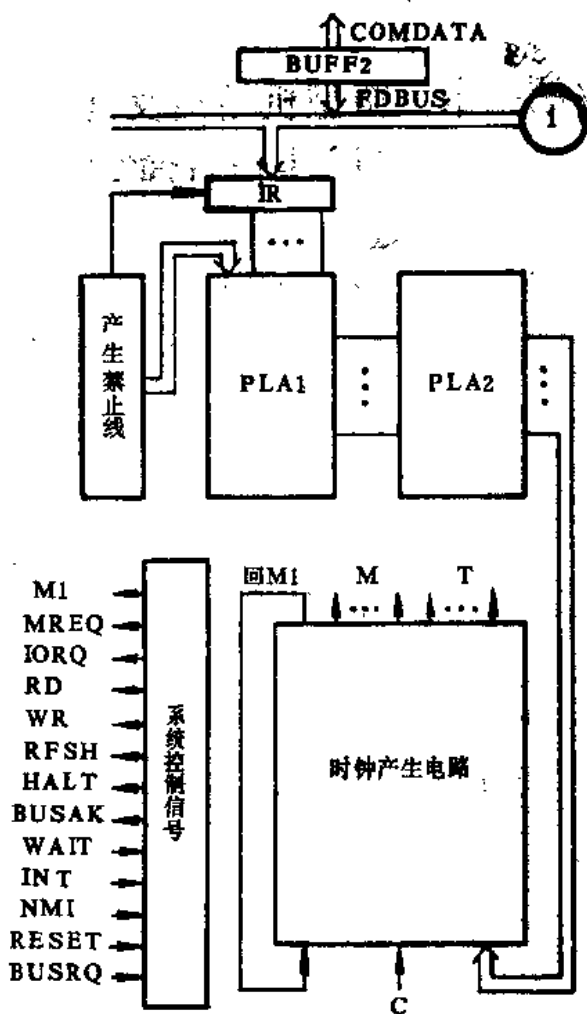
Z80 机 CPU 分成两个部分:控制部分和运算部分。

控制部分,用以产生 M1、MREQ、IORQ、RD、WR、RFSH、HALT 和 BUSAK 八个控制信号,由各个时钟控制 Z80 机 CPU 内的各部件、内存及 I/O 设备协调动作;

运算部分,是 Z80 机 CPU 的信息传输及运算部件,是执行指令的动作部分。

图 1.1 只给出了总线与寄存器结构,以及说明信息在 CPU 内的传输路线。我们感兴趣的也恰是 Z80 每条指令的执行过程,即所谓指令的数据流程。

模拟模型的建立是对实际系统抽象化的过程。为了对 Z80 机进行描述,首先必须对它的各组成部分(以后称为模块)及其对外连接和内部互相连接的关系有一个明确的概念。



(a)

图 1.1 Z80机CPU 硬件结构框图

1. 总线与寄存器结构

(1) 外部总线 共有两组：

COMADDR[16]：地址总线。共 16 位，用于传输地址，实现

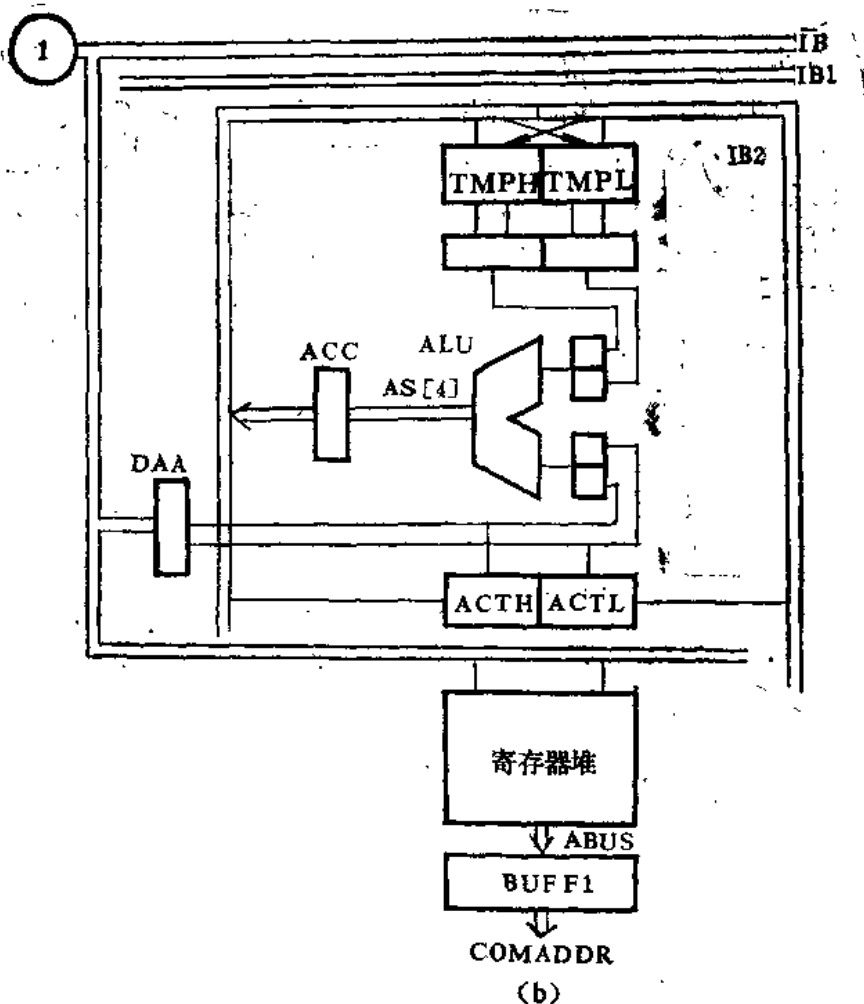


图1.1 Z80机CPU硬件结构框图

CPU 对内存或外设的访问。括号中的数字表示地址总线的位数，这是 AHPL 语言描述的书写方式。

COMDATA[8]：数据总线，用于CPU 与外部交换信息。

(2) 内部总线 共有六组

FDBUS[8]：数据缓冲器 BUFF2 与内部总线 IB 之间的引

线。

IB[8]：用于传输8位信息。

IB1[8]：用于移位等操作。

IB2[8]：实现与暂存器ACT、TMP交换信息。由于Z80机CPU中的ALU是四位运算器，因而IB2[8]分为高、低两部分IB2H和IB2L，并可进行半字交换传送。

ABUS[16]：寄存器堆(参见图1.2)到地址缓冲器BUFF1的引线。

AS[4]：ALU与ACC之间的引线。

(3) 外部输入引线 共有八根：

WAIT：等待信号线。用于协调CPU与内存或外设的速度。当被访问的内存或外设速度低于CPU时，向CPU发出WAIT信号，CPU按有关机器周期预定的节拍对WAIT进行检测；若WAIT有效，CPU进入等待，并继续对WAIT线进行检测，直至WAIT无效。例如访存周期的第二个节拍或访问外设的第三个节拍时，要对WAIT线进行检测。

RESET：复位信号线。当RESET有效(为低)时，CPU进行初始化，即程序计数器PC、再生地址计数器R、中断矢量寄存器I以及中断允许触发器IFF1和IFF2清零。每次加电时，CPU都能够自动完成一次复位。

BUSRQ：总线请求信号线。当采用DMA方式工作时，DMA控制器向CPU发送信号，请求CPU放弃对总线COMA、DDR和COMDATA的控制权。

INT：中断请求线。它为低电平时，表示外设向CPU申请中断。

NMI：不可屏蔽中断请求线。

CLOCK：时钟信号。它的周期T称为时钟周期，用它来同步模拟过程。

VC：电源线。

GRD: 地线。

(4) 输出引线 共有七条:

M1: 取指令周期信号。

RD: 读内存或外设信号。

WR: 写内存或外设信号。

MREQ: 存储器请求信号。当 MREQ 与 RD 同时有效时, 进行存储器读, 把内存数据送到 COMDATA 总线上; 当 MREQ 与 WR 同时有效时, 进行存储器写, 把 COMDATA 总线上的信息写入内存(见图 1.3)。

IORQ: I/O 请求信号, 请求 I/O 设备进行输入、输出操作(见图 1.3)。

RFSH: 刷新信号, 和 MREQ 一起用于刷新动态存储器。

HALT: 暂停状态线。通知内存或外设, 这时 CPU 只使内存刷新, 不进行其余操作。

(5) 缓冲器

BUFF1[16]: 地址缓冲器。暂存 CPU 提供的十六位二进制地址。

A[8]	F[8]
A'[8]	F'[8]
B[8]	C[8]
B'[8]	C'[8]
D[8]	E[8]
D'[8]	E'[8]
H[8]	L[8]
H'[8]	L'[8]
IXH[8]	IXL[8]
IYH[8]	IYL[8]
SPH[8]	SPL[8]
W[8]	Z[8]
I[8]	R[8]
PC[16]	
±1计数器	

图1.2 寄存器组

BUFF2[8]：数据缓冲器。暂存 CPU 与外部交换的 8 位二进制信息。

(6) 寄存器及寄存器堆

IR[8]：指令寄存器。存放现行指令操作码，以便进行译码。

ACC[4]：存放算术逻辑部件 ALU 产生的进位。

ASS：进位触发器。

IMF[2]：中断类型寄存器。

IFF1 和 IFF2：中断允许触发器。

I[8]：中断矢量寄存器。

R[8]：再生地址寄存器。提供内存刷新地址。

PC[16]：程序计数器。保存着下一条要执行的指令或多字节指令的下一字节地址。

AF、AF'、BC、BC'、DE、DE'、HL 和 HL'；其中 F 是标志寄存器，带' ' 的是辅助寄存器，不带' ' 的是主寄存器。主、辅寄存器之间可以实现快速切换，从而在中断时可以用辅助寄存器保护现场，以提高中断响应速度。还有变址寄存器 IX 和 IY、堆栈指示器 SP 以及暂存上述寄存器提供的地址的缓冲器 W 和 Z。

(7) 暂存器

ACT 和 TMP，存放操作数。在 Z80 机 CPU 中 ALU 是四位运算器，每次运算只能对 ACT、TMP 的高四位或低四位进行操作，所以 ACT 和 TMP 分别分成 ACTH、ACTL 和 TMPH、TMPL。

CHAF[16]、CHBC[16]、CHDE[16]、CHHL[16] 和 CHDH[16]：主、辅寄存器交换数据用的缓冲器。

RI[8]：中断矢量暂存器。用以暂存从外设读入的中断地址的低 8 位。

2. AHPL 语言语法结构

数字系统常包含若干互相连接的子系统。采用 AHPL 语言对

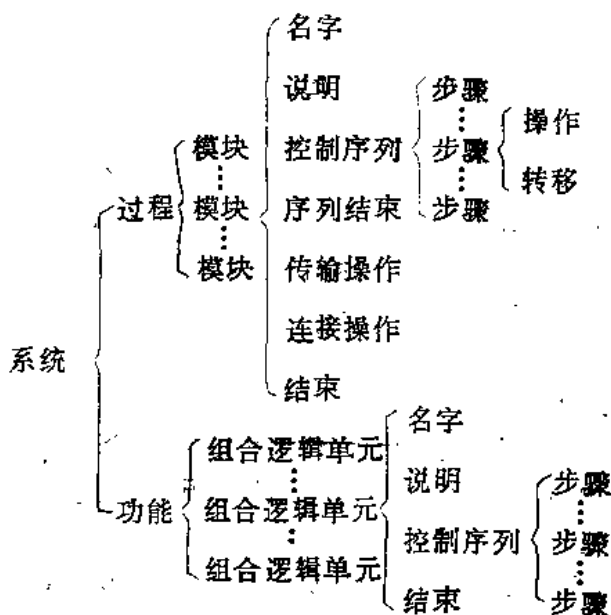
其进行描述时常将它们看成模块(MODULE)和单元(UNIT)。

这里的每个模块实际上是一个有限状态的自动机。AHPL语言用单元来描述功能的实现；用模块描述过程，借以实现运算进程的控制，于是便可以完成对数字系统的描述，整个AHPL语言的语法结构如表 1.1 所示。无论模块或单元的描述都包括命名、说明和控制序列等组成部分，命名用于区分不同的模块或单元，如 Z80机包括命名为 Z80CPU、CONTROL和MEMORYS的三个模块，指明所描述部件的组成元素的名称、性质和规模，如寄存器或引线的名称和维数。例如

EXBUSES, COMADDR[16], COMDATA[8].

说明 COMADDR 和 COMDATA 是外部总线，以 EXBUSES 标志；它们都是一维的，分别为十六位和八位。控制序列则描述操作控制的进程，它是由若干步骤(STEP)组成的。每个步骤的执行过程受时钟和状态控制。执行步骤包括操作和转移，操作是指数据传输和连接等，而转移是确定后继步骤。

表1.1 AHPL语言语法结构



一个数字系统，或其子系统，都可看作是一个简单的时序线路或一个自动机。Z80机的描述和模拟就是按三个自动机进行的。在AHPL描述中，时钟可以是真实存在的机器时钟，也可以是虚拟的机器周期脉冲信号或指令周期脉冲信号。在Z80机描述中，时钟是虚拟的。

在数字系统的AHPL语言描述中，变量说明必须根据被引用的情况分成局部的、半局部的和全局的。

局部变量的说明必须放在使用这些变量的模块里，而且只能在该模块内引用。这些被说明的局部变量有MEMORY(存储元件)、ONESHOTS(单稳触发器)、BUSES(总线)和CLUNITS(标准组合逻辑单元)。同一符号可在若干个模块里出现，但分别用于描述不同的器件。例如，在Z80CPU模块中，CLUNITS中的DCD1、DCD2等又在CONTROL模块中的CLUNITS中出现。虽然它们的符号名称相同，但对应完全不同的器件。

半局部变量描述了在模块之间单向传输的引线，必须在使用这些变量的模块里说明。同一个符号名称可能在若干个模块的说明中出现，并且具有相同的局部数值。这些被说明的半局部变量有INPUTS(输入)、OUTPUTS(输出)。例如控制指针MINT[6]和系统工作状态线READY，在Z80CPU模块中是作为输出变量来说明的，而在控制模块中则是作为输入变量来说明的；类似地，如M1、MREQ、…在控制模块中是作为输出变量来说明的，而在内存模块中则是作为输入变量来说明的。须注意，一个模块的输出不能与另一个模块的输出同名，但可作为多个模块的输入来说明。

全局变量是指那些能够具有外部数值的各模块的公有变量。它们只能在一个模块里说明，包括EXINPUTS和EXBUSES。例如在Z80CPU模块中已用EXBUSES说明了COMADDR[16]和COMDATA[8]是外部总线，即为全局变量，那么在MEMORYS模块的描述中将直接引用COMADDR[16]和COMD-

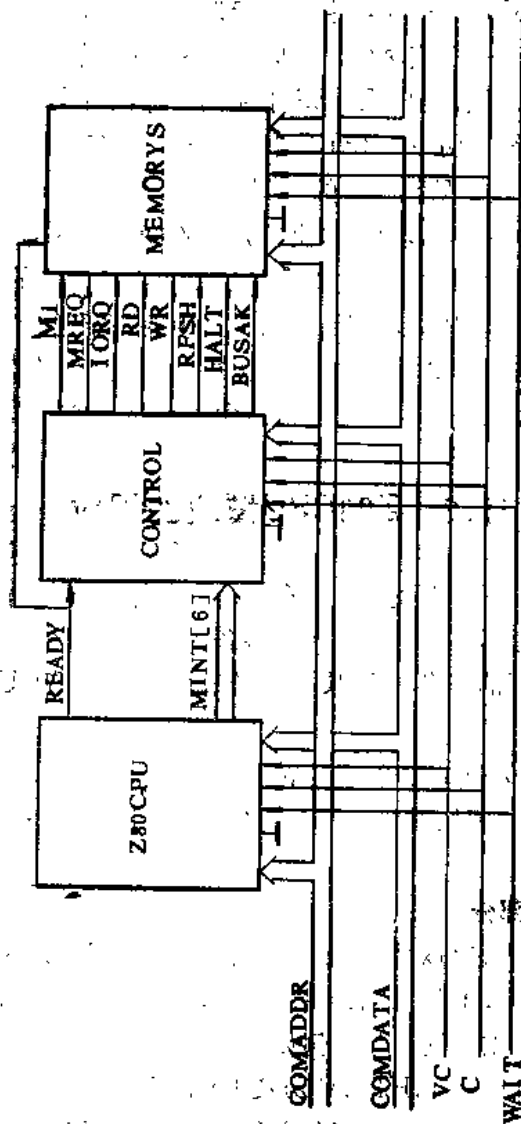


图1.3 模块间的联系