

83317

5087

756792

2648

微型计算机接口技术

吴报鑫 张华宋

37

48

上海交通大学科技交流室

一九八五年六月

上海兴华电脑技术应用服务部

为各地用户提供各种单板机、电脑配件、集成电路、电脑零件、接插件等。承接电脑应用项目设计及单板设计、编程、维修等技术服务。办理邮购，欢迎来人来函联系。

一、TP801 单板机，8085 单板机，TP801P 打印机，单板机专用电源，EPROM 擦除器，MCS-48 单片机及开发装置。

二、Z80 多功能组合模块，有 CPU 主控板、I/O 接口板、模数、数模转换板、内存扩充板、编程板等。用户可根据实际需要选用其中一块或几块组合成仪器或控制设备，具有成本低，组装方便等优点。单元板尺寸为 160mm×80mm。

三、8085 单板机控制的 EPROM 编程器，可对 2716、2732、2764、27128 编程，可键输入，也可拷制，每次编程任意字节，有出错检查。除编程外，仍可作单板机使用。

四、进口 74LS 系列电路，Z80 系列、INTEL 系列各种电脑芯片、接口电路、单片机电路、AD 公司、NATIONAL 公司 A/D、D/A 转换器等线性集成电路，78、79 系列三端稳压电路及一些 CD 系列 MOS 电路。备有各种器件手册可供选用。

五、进口运算放大器、比较器、驱动器、采样保持器、光电耦合器、及三极管、二极管、发光二极管、数码管、按键、晶体及软盘片等。

六、进口各种双列集成电路插座(8P--40P)，扁平电缆线及扁平电缆插头座(10P~60P)，印刷板插座(16P~100P)，各种规格航空插头座、实验板、通用印刷板等。

七、绘制印刷线路板设计用品：各种规格进口胶带、圆孔、双列集成电路管脚(8P~40P)、符号等。承接线路板底稿设计贴制。

八、与上海交通大学计算机系协作为用户提供各种软件，并指导上机操作使用。

本部地址：上海华山路 1898 号(靠近上海交通大学)

电话： 379163

联系人：马天涯 宋海志

前言

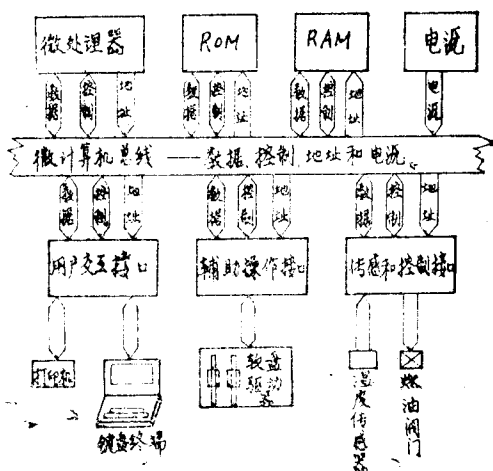
微型计算机是微型化了的计算机，它是大规模集成电路技术发展的结果。目前微型计算机的应用已深入到各个领域。大致包含以下四个方面：科学计算，企业管理，工业控制和智能化产品。

前二方面应用中，一般采用通用微型计算机系统，所需接口主要用于人机交换信息，例如键盘接口，显示接口，打印接口及软盘接口等。通常采用高级语言编制软件。

后二方面应用面广，目的各异。除了人机之间交换信息以外，还需在控制对象和计算机之间交换信息。所需接口常常要按特定控制对象或产品专门设计。较多地采用汇编语言或高级语言与汇编语言交叉应用。

一个典型的微型计算机系统硬件如图所示。一般包含微处理器、存储器、接口电路、总线、电源和各种外设。

微处理器即 CPU，是系统的控制和处理中心，将取自外部世界的信息进行存贮、处理后，再去控制客观世界。CPU 在工作时不断地与存储器，各种外部设备进行联系，交换信息。由于外部设备的种类有机械式、电动式、电子式；交换信息的型式有数字量，模拟量或开关量。信息传送速度可以高达每秒几十万 Bit，也可慢到分或秒的数量级。信息传送方式可以是并行方式或串行方式。因而 CPU 与外部设备交换信息比与存储器交换信息复杂得多。为了适应各种外部设备的需要，CPU 与外部设备之间需要有特定的硬件连接和相应的软件控制。



在微型机系统中，一个部件与另一些部件之间的联系，需要硬件的连接和软件的控制。完成这一任务的硬、软件综合称为接口。对这些硬、软件的设计称接口技术。

接口通常分为四种类型：用户交换接口，辅助操作接口，传感接口和控制接口。

用户交换接口的主要功能是将来自用户的数据，信息传送给微型机，或将来自微机的数据，信息传送给外部设备。常用接口有键盘接口、终端显示接口属这一

类。

辅助操作接口包括总线驱动器、总线接收器、时钟电路、磁带和磁盘系统接口。一般情况下，CPU 的扇出仅是 1~2 个 TTL 负载。如果 CPU 所带的存贮器或接口较多，则 CPU 的驱动能力将不足。为使信息有效地传送，需要加接总线驱动器、总线接收器。对于大容量的外存贮器如磁带、磁盘和软盘系统的接口一般也归于这一类。

传感接口通常用于微机控制和检测系统中。由于客观世界是一个模拟世界，微机要控制和检测的外部信息例如压力、温度、流量、形变等一般都是模拟量，而微机所能接受和处理的都是数字量，因此常常通过传感接口去监视、感受外部世界被检测或控制对象的变化，将这种变化转换成电压或电流变化，再进一步转换成微型机能接受的数字量。

控制接口是当微机进行处理、运算后，需要控制外部执行机构动作时，（例如驱动马达，启动阀门或点亮指示灯等）需要把数字量转换成模拟量，以实现对外部世界的控制。

接口技术主要解决以下几方面的问题：

(1) 输入：把外部设备送往微机的信息转换成与微机相容的格式。

(2) 输出：把微机送往外部设备的信息转换为与外设相容的格式。

这里的相容是指在信号类型，信号电平，信号传送方式等方面使外设与微机之间取得协调。

(3) 同步脉冲的产生：通常外部设备不止一个，微机在不同时刻需和不同外设取得联系，必须对外设寻址。一般需产生设备选择脉冲。以使相应外设与微机同步工作。

(4) 中断处理：检测并处理外部设备发到微机的中断请求信号。

实际上，一个完整的接口设计，包括机械的、电气的和功能的几方面要素：机械的包括接插件，电缆等；电气的包括发送与接收电路、信号形式、电平等；功能的要素包括接口的管理能力、接收、发送、控制等功能的逻辑，信息系统的编程等。

综上所述，为使微机有效地与客观世界或用户进行联系，充分发挥它的效能。接口设计是一项重要的工作。因而掌握接口技术，就成了推广应用微机的关键。

随着大规模集成电路工艺水平的不断提高，微型机接口芯片发展也很快。除了各种功能芯片外，已有各种功能模块出现。并进而又向着把 CPU，存贮器，各类接口制造在一块芯片上的单片机和专用微处理机发展。但是它们不可能全部替代从事微机开发应用中的接口设计工作。这些发展只是使接口设计中硬件设计的比重减少而软件设计比重相应增大。

目前，微机的应用方兴未艾，为使广大工程技术人员科技工作者及各类有志于微机应用的同志能在了解微机原理的基础上，尽快地掌握接口技术，把微机应用到生产实践、科学研究，技术改造等方面去。我们把近几年来在科研、教学中积累的有关接口技术的资料，经验加以总结，概括，提高，整理付印。

为使本书重点突出，避免篇幅冗长，书中对微处理器的原理、结构，指令系统，基本时序不再赘述。对书中阐述的各章内容力求避免泛泛介绍和广为罗列，而着重从应用的角度介绍接口技术中硬件和软件的典型设计方法。

由于作者水平有限，编写时间仓促，书中一定存在不少错误和缺点。恳请读者批评指正。

微型计算机接口技术

目 录

第一章 微型计算机的存储器及其接口方法	(1)
第一节 微型计算机存储器概述	(1)
第二节 存储器的空间分配及其与 CPU 的接口方法	(5)
第三节 存储器的扩充及总线驱动	(10)
第四节 动态 RAM 及其接口方法	(17)
第二章 微型计算机的输入/输出技术和中断处理	(22)
第一节 输入/输出的寻址方式	(23)
一、I/O 对应 I/O 寻址方式	(23)
二、存储器对应 I/O 寻址方式	(23)
三、Z80 输入输出指令组	(25)
第二节 输入/输出数据传送的控制方法	(28)
一、等待方式	(30)
二、查询方式	(32)
三、中断方式	(34)
四、直接存储器存取方式(DMA方式)	(35)
第三节 中断处理技术	(37)
一、Z80 中断系统组成	(39)
二、Z80 供中断系统使用的一些专用指令	(40)
三、中断系统的功能——优先级及嵌套	(41)
第四节 Z80 的中断方式	(45)
一、Z80 的 0 方式中断	(46)
二、Z80 的 1 方式中断	(55)
三、Z80 的 2 方式中断	(56)
四、不可屏蔽中断	(58)
第三章 微型计算机典型的外设接口及应用	(60)
第一节 并行接口及其应用	(61)
一、并行接口的分类	(61)
二、通用并行输入/输出接口——8212	(64)
三、Z80 并行输入/输出接口——Z80-PIO	(67)
四、PIO 的工作方式 0——输出工作方式及打印机接口	(72)
五、PIO 的工作方式 1——输入工作方式及光电阅读器接口	(77)
六、PIO 的工作方式 2——双向工作方式	(83)

七、PIO的工作方式3——位控工作方式	(83)
八、可编程的并行接口——8255A	(86)
九、七段显示器、键盘及终端显示接口	(91)
十、光电隔离技术	(102)
第二节 计数器/定时器电路(Z80-CTC)	(106)
一、Z80-CTC 结构	(106)
二、CTC 的编程	(108)
三、CTC 的工作方式	(110)
四、CTC 应用举例	(113)
第三节 串行通讯接口及其应用	(118)
一、串行通讯的基本概念	(118)
二、串行异步通讯的软件实现——电传打字机接口	(121)
三、Z80串行输入/输出器件——Z80-SIO的结构	(123)
四、Z80-SIO 的编程	(127)
五、通过 SIO 实现电传打字机接口	(130)
六、应用 SIO 实现盒式磁带录音机接口	(132)
第四节 直接存贮器存取(DMA)	(138)
一、Z80-DMA 的功能逻辑和引脚	(139)
二、Z80-DMA 的操作类型和操作方法	(141)
三、Z80-DMA 的控制命令及编程方法	(143)
四、Z80-DMA 的可读寄存器	(147)
五、Z80-DMA 的中断逻辑	(148)
六、Z80-DMA 的定时控制寄存器	(149)
第五节 软磁盘及硬磁盘接口	(150)
一、软磁盘接口	(150)
二、硬磁盘接口	(156)
第四章 数模及模数转换技术及其应用	(160)
第一节 数字模拟转换(D/A转换)	(160)
一、D/A 转换的基本原理	(160)
二、AD1408 D/A 转换器及其与 CPU 的接口	(162)
三、D/A 转换器的应用	(163)
四、8 位 CPU 与 12 位 D/A 转换器的接口方法	(168)
第二节 模拟数字转换(A/D 转换)	(170)
一、跟踪型 A/D 转换器	(171)
二、逐次逼近型 A/D 转换器	(173)
三、A/D 转换模块与 CPU 的接口	(175)
四、A/D 转换的分辨率和误差	(176)
五、双积分 A/D 转换及数字面板式仪表	(178)
六、并行 A/D 转换器(瞬时 A/D 转换器)	(181)

第三节 采样保持器和多路选择/分配器	(182)
一、采样保持器的原理及参数	(183)
二、多路选择/分配器	(185)
三、采样保持器与多路开关在系统中的安排	(186)
第四节 A/D 转换技术的应用—数据采集系统	(187)
一、微电脑控制皮肤温度测定仪中的数据采集系统	(193)
二、采用十二位 A/D 转换器的数据采集系统	(197)
第五章 微型计算机的总线	(188)
第一节 概述	(197)
第二节 芯片总线及其作用	(198)
第三节 内部总线——S-100 总线	(200)
第四节 外部总线	(204)
一、IEEE—488 总线标准及接口功能	(205)
二、EIA—RS232C 串行通讯标准	(209)
第六章 微型计算机应用系统的研制方法	(213)
第一节 应用系统的研制过程	(213)
一、硬件研制过程	(214)
二、软件研制过程	(214)
第二节 软件研制工具	(214)
一、编辑程序	(215)
二、汇编程序	(215)
三、调试程序	(216)
四、二进制装入程序	(216)
五、交叉汇编程序	(217)
六、模拟程序	(218)
第三节 硬件研制工具	(219)
一、微型计算机研制系统	(220)
二、联机仿真器	(223)
三、逻辑分析仪	(229)
附录A Z-80 指令系统	(234)
附录B 常见术语的英中对照	(237)

第一章 微型计算机的存贮器 及其接口方法

第一节 微型计算机存贮器概述

存贮器是计算机的重要组成部分,用于存贮二进制信息从而使计算机具有记忆功能。当使用者将有关指令、数据、信息分别存入存贮器后,计算机便可自动地、有条不紊地按人的意图去工作。

计算机的存贮器一般有两种,在主机内部的存贮器称内存贮器,又叫“内存”。通常内存容量小,速度较快,可与 CPU 直接交换信息。在主机外部的存贮器称外存贮器,又叫“外存”。它属于计算机的外部设备。常见的外存有软磁盘、硬磁盘,磁带机等。它们的特点一般是容量大,速度慢,不能直接与 CPU 交换信息,必须将外存的信息调入内存后才能被使用。

本章主要讨论微型计算机的内存贮器,有关外存的内容在以后章节中讨论。

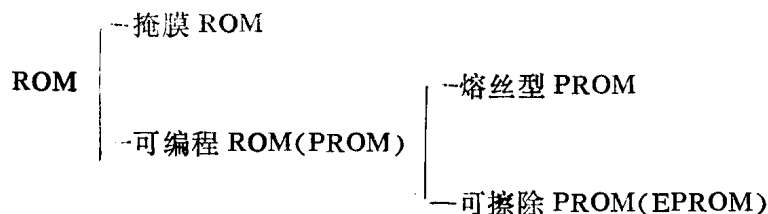
大型或小型计算机的内存以前通常采用磁芯存贮器。近年来,随着大规模集成电路的发展,在大型或小型计算机中已开始采用半导体存贮器。而微型计算机几乎全都采用半导体存贮器。这种存贮器具有体积小、重量轻、耗电少、速度快及非破坏性读出(即读出以后信息仍旧存在)等优点。

常见半导体存贮器可分为只读存贮器(Read Only Memory 简称 ROM)和随机存取存贮器(Random Access Memory 简称 RAM)。ROM 是只能读出的存贮器,ROM 中的信息在使用时是不能改变的,即使用时无法对它写入。当关断电源时 ROM 中的信息不会消失,通电后又可使用。因此,ROM 常用来存放永久性的程序,如监控程序,管理程序,调试、诊断程序,专用操作程序等。也可存放不变的常数或各种表格。RAM 中的信息在使用时可以被读出,也可以重新写入新的信息。所以 RAM 又称为读写存贮器。它常被用来存放需要改变的程序或数据。当切断电源后,RAM 中存放的信息在几毫秒后便消失,电源恢复后,信息不再恢复,必须重新写入。这是半导体 RAM 的一个缺点,即存在信息的易失性。

解决半导体 RAM 信息易失性问题的一个重要办法,是设置后备电源。当主电源停电时,备用电源的控制电路应保证在很短时间内把备用电源接到半导体 RAM 上去,使存在 RAM 中的信息不致丢失。一般的高能电池已足以作为备用电源了。除了使用备用电源以外,也可采用新的工艺,如 MNOS 工艺,生产非易失性的存贮器。这样,在一般的 RAM 中,设置一个 MNOS 工艺的存贮器。一旦主电源停电,就可以利用与主电源并联的电容器能在短时间存贮电荷的特性,在很短时间内把信息转存入非易失性的存贮器中。不过这种办法,目前还比较贵。

半导体存贮器,从工艺上可分为双极型和 MOS 型二类。若按存贮信息的方法来划

分, ROM 可分为:



掩膜 ROM 由制造厂按用户给定的信息要求, 掩膜制造而成, 封装后不能改写, 即用户只能读出无法改写。可编程 ROM(PROM)中信息可由用户根据需要写入。熔丝型 PROM 是用户把熔丝熔断或不熔断来表示“1”或“0”。信息写入后, 不能再改写。可擦除 PROM, 信息写入后, 可用电或紫外线擦除, 然后重新改写。但擦除或改写是需要设备和时间的, 所以它们在装上微机应用时仍是作只读存储器来用。

RAM 可分为静态 RAM 和动态 RAM。静态 RAM 的每一位信息存储在一个触发器中。电源存在时, 信息不会丢失。但电源失去几毫秒后, RAM 中信息就会消失。动态 RAM 是利用 MOS 晶体管栅极与基底之间的寄生电容上的电荷来存储信息, 这种电荷在几毫秒后便会漏失, 所以每隔一定时间必须重新写入一次以保持原来的信息, 这一重写过程称为刷新或再生。动态 RAM 的单位面积集成电路中存储信息较多, 存取速度也比较快, 功率耗散小, 但需附加刷新电路。所以一般大容量存储器采用动态 RAM, 小容量存储器则用静态 RAM。

适于与微处理机配套的半导体存储器分类, 归纳列入表 1—1:

表 1—1 半导体存储器的分类

按功能分类	按工艺分类	按电路性质分类
随机存取存储器 (RAM)	双极型	静态RAM
	MOS型	静态RAM
读/写存储器 (RWM)	双极型	动态RAM
		一次掩膜成型ROM
只读存储器 (ROM)	双极型	熔丝型PROM
		一次掩膜成型ROM
	MOS型	EAROM(电可改写的ROM)
		EPROM(可擦可编程的ROM)

存储器中存储信息的每一个单元, 必须有一个编号即有一个地址。我们是按地址号来选择合适的存储单元的。对具有 16 条地址线的微型机, 有 $2^{16} = 65536$ 即 64K 存储单元可供寻址。即 16 条地址线具备 64K 存储单元寻址能力。通常以 4 位十六进制数表示存储器地址号, 即 $0000_H \sim FFFF_H$ 。

存储器的容量一般用可寻址单元数 $\times$ 每单元数据位数表示。例如, 1024×1 ; 2048×1 ; 1024×4 ……等。常见的 RAM2114 的容量是 $1K \times 4$, 这表明一块 RAM 2114 芯片有 1024 个可寻址的存储单元, 每个存储单元有 4 位二进制数。即一块 RAM 2114 芯片可存储 $1024 \times 4 = 4096(4K)$ 位信息。这一芯片需具有 10 条地址线和 4 条双向数据线。又如 EPROM

2716 的容量是 $2K \times 8$ 。这表明一片 EPROM 2716 芯片有 2048 个可寻址的存储单元，每个存储单元有 8 位二进制数。即一块 EPROM 2716 可存储 $2048 \times 8 = 16384 (16K)$ 位信息。这一芯片需具有 11 条地址线和 8 条双向数据线。

对于同样存储容量的存储器芯片可以有不同的做法。例如，对于存储容量是 1024 (1K) 位信息的存储器，可以做成有 1K 个地址，每一地址只有一位二进制数，写成 1024×1 。也可以做成有 128 个地址，每一地址有 8 位二进制数，写成 128×8 。对于前一做法来说，1K 个地址要 10 个地址引脚，另外还要一个双向数据引脚，共计 11 个引脚。对于后一做法来说，128 个地址要 7 个地址引脚，还要 8 个双向数据引脚，共计 15 个引脚。由此可知存储器做成 1024×1 比做成 128×8 的引脚少。引脚少制造方便，所以存储器的数据往往不做成 8 位而做成 4 位或 1 位。字长不足 8 位的存储器与 8 位的 CPU 相连时，可把几片存储器并联成 8 位数据。

通常，存储器芯片中的存储单元排成矩阵形式，或者说排成阵列。例如，1024 个存储单元可排成 $32 \times 32 (=1024)$ 的矩阵。即 32 行 32 列的矩阵。2048 个存储单元可排成 $128 \times 16 (=2048)$ 的矩阵。即 128 行 16 列矩阵等等。对存储器的寻址是通过行地址选择线和列地址选择线的重叠来实现的，如图 1-1 所示。这样做可以节省译码和驱动电路。以

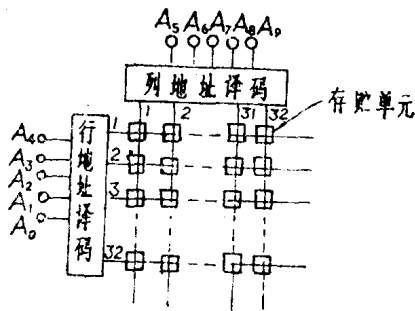


图 1-1 存储单元阵列的重叠选址法

1024 个存储单元而言，行地址译码器和列地址译码器各只需 32 条输出线。即总共译码输出是 64 条。若不采用阵列，则译码输出线就需要 1024 条。

各类存储器的芯片除了必须有地址线，数据线以外还需要有控制线。图 1-2 是 RAM 2114 的原理框图及引脚。由图可见，芯片共有 18 只引脚，除工作电源及地线外，还有 10 条地址线，4 条双向数据线，2 条控制线。其中 $\overline{WE}$ 为写入允许控制， $\overline{CS}$ 为片选控制。它们的逻辑关系如表 1-2 所示。

表 1-2 RAM 2114 控制脚 $\overline{CS}$ 和 $\overline{WE}$ 的逻辑关系

$\overline{CS}$	$\overline{WE}$	芯片状态
1	x	禁止
0	0	写入
0	1	读出

当 $\overline{CS}$ 是高电平时，芯片被禁止读写。当 $\overline{CS}$ 是低电平时，芯片才允许读写操作。当

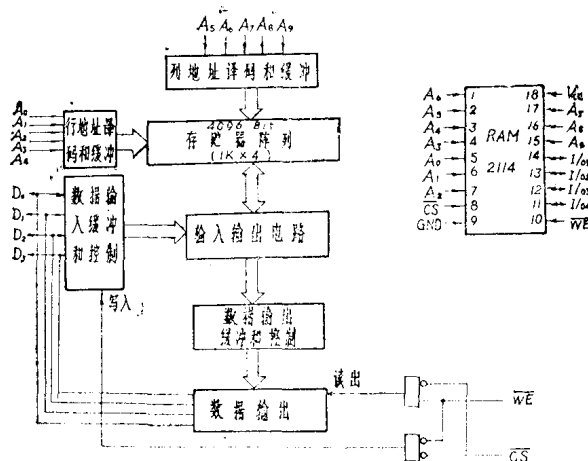


图 1—2 RAM-2114的原理框图及引脚

$\overline{WE}$ 是低电平时，芯片处在写入状态；当 $\overline{WE}$ 为高电平时，芯片处在读出状态。

图1—3a)示出了常用的 EPROM 2716 的原理框图及引脚。该芯片共有 24 条引脚，其

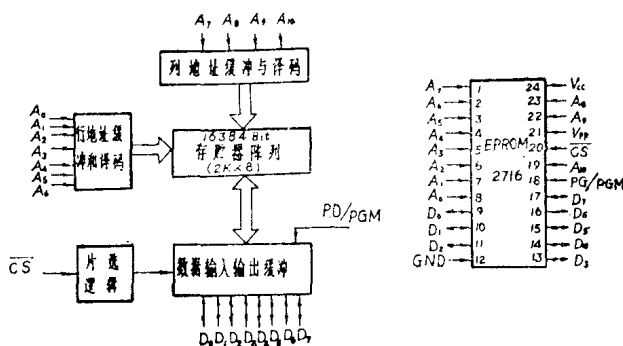
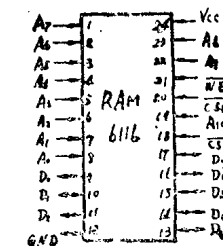


图 1—3a) EPROM 2716 的原理框图和引脚



b) RAM6116 的引脚图

中 11 条地址线，8 条双向数据线，11 条地址线中的 7 条用于行地址译码，以选择 128 行中的一行。4 条用于列地址译码，以选择 16 列中的一列。1 条片选线 $\overline{CS}$ 。 V_{PP} 引脚在读操作时接 +5V，在编程操作时接 +25V。 PD/PGM 引脚在读操作时可接低电平，在编程时需加上宽度为 50~55 毫秒的 TTL 高电平脉冲。在芯片未被选中时， PD/PGM 引脚可接高电平，这时芯片功耗可下降四分之三，由 525mW 下降为 132mW。 $\overline{CS}$ 、 V_{PP} 、 PD/PGM 引脚在不同工作方式时的状态归纳见表 1—3。

表 1—3 EPROM 2716 的工作方式选择

引脚	PD/PGM	$\overline{CS}$	V_{PP}	输出
工作方式				
读	低	低	+5V	输出
编程	50-55毫秒宽 TTL正脉冲	高	+25V	输入
未选中	无关 (若接高，则 可降低功耗)	高	+5V	高阻

图 1—3b) 示出了 RAM6116 的引脚图。RAM 6116 容量是 $2K \times 8$ 。每一块芯片可存贮 2K 字节。使用时, 片选端 $\overline{CS}_1$ 和 $\overline{CS}_2$ 均接向地址译码器; $\overline{WE}$ 端接向 $\overline{MEMW}$ 信号, 在写操作时 $\overline{WE}$ 为“0”, 读操作时 $\overline{WE}$ 为“1”。

RAM 6116 的一个重要特点, 是它的引脚数与 EPROM 2716 相同, 且它在使用时的引脚接法与 EPROM2716 读操作时引脚接法兼容。因此, 应用 RAM6116 调试好的程序, 可固化到 EPROM2716 中, 然后将 EPROM2716 替换 RAM6116 而不必更改任何接线。使用很方便, 因此, RAM6116 的应用已越来越广。

第二节 存贮器的空间分配及其与 CPU 的接口方法

CPU 对存贮器进行读写操作时, 首先由地址总线给出存贮单元的地址号, 然后给出相应的读或写控制信号, 最后才能在数据线上进行信息交换。所以 CPU 与存贮器接口时, 需要连接地址线、控制线和数据线。

8 位微处理器通常有 8 条数据线 ($D_0 \sim D_7$), 16 条地址线 ($A_0 \sim A_{15}$)。存贮器的每一存贮单元有一独立的地址, 用 4 位 16 进制数表示。对 16 条地址线的微机, 可选择 $2^{16} = 65536$ (即 64K) 地址, 即存贮器的地址可由 $0000_H \sim FFFF_H$ 。通常也称寻址能力是 64K。具有不同数量存贮单元的存贮器芯片与 CPU 连接时, 所需连接的地址线数目也不同。例如, 有 1K 存贮单元的芯片需用 $A_0 \sim A_9$ 十条地址线。有 2K 存贮单元的芯片需用 $A_0 \sim A_{10}$ 十一条地址线等等。

一般用于 8 位微机的存贮器字长也应该是 8 位。当存贮器字长不足 8 位时, 可用几片存贮器芯片组合成 8 位字长。例如, RAM2114 芯片是 $1K \times 4$, 只对应 4 位字长, 所以要用 2 片合在一起才能组成 8 位字长, 共同在 1K 地址范围内工作。由 2 片 2114 芯片共同形成 1K 容量 8 位字长的存贮器与 CPU 接口的示意图如图 1—4 所示。它们的地址线是并联在 CPU 的 $A_0 \sim A_9$ 地址线上。数据线分别接在 CPU 的 $D_0 \sim D_3$ 和 $D_4 \sim D_7$ 上。当地址线上出现某一地址时, 2 片 2114 选中同一单元地址, 它们的数据线共同形成一个字节, 在片选和读写信号控制下与 CPU 交换信息。对于具有 8 位数据线的存贮器。例如 2716, 每一块芯片的数据线就可单独实现与 CPU 数据接口来实现 8 位字长的读写操作。

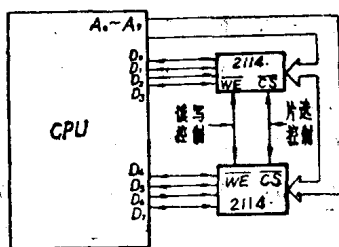


图 1—4 RAM2114 与 CPU 接口示意图

由于每一片存贮器寻址能力是有限的, 例如 2114 是 1K 寻址能力, 2716 是 2K 寻址能力。如果需实现 64K 可寻址的存贮器, 就需由多片存贮器芯片来提供。许多存贮器芯片都挂到 CPU 的地址总线 and 数据总线上。实际上, 存贮器是通过三态缓冲方式与总

线连接的。这种连接使得不进行读写操作的存储器与总线之间处于高阻抗状态。而只有被选中需要进行读写操作的存储器芯片才能在总线上与 CPU 交换信息。所谓选中的存储器是指它的片选端(CS)处在有效状态,即低电平。对存储器 CS 端的控制通常称为片选控制。片选控制一般有下列几种方法:

1. 线性片选法

图 1—5 中,有 4 块 2K×8 的存储器,每块芯片的 8 个数据端和数据总线 D₀~D₇ 相连。利用 11 条地址线 A₀~A₁₀ 对每一存储器中的 2K 个地址进行寻址。而地址线 A₁₁、A₁₂、A₁₃、A₁₄ 分别接到每一芯片的片选端 CS, CS 为低电平时,允许该芯片工作。例如当 A₁₁ 为低, A₁₂, A₁₃, A₁₄ 为高时,存储器 4 被选中,允许工作。其它三片存储器均无效。各芯片的寻址范围见表 1—4。

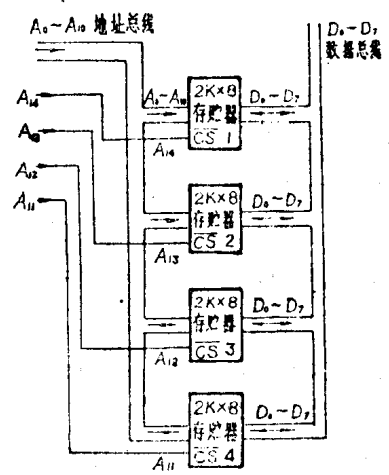


图 1—5 用线性片选法连接存储器

表 1—4 图 1—5 中各芯片的寻址范围

芯片		A ₁₄ A ₁₃ A ₁₂ A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	地 址
1	起始地址	0 1 1 1	0	0	0	0	0	0	0	0	0	0	0	3800H
	终了地址	0 1 1 1	1	1	1	1	1	1	1	1	1	1	1	3FFFH
2	起始地址	1 0 1 1	0	0	0	0	0	0	0	0	0	0	0	5800H
	终了地址	1 0 1 1	1	1	1	1	1	1	1	1	1	1	1	5FFFH
3	起始地址	1 1 0 1	0	0	0	0	0	0	0	0	0	0	0	6800H
	终了地址	1 1 0 1	1	1	1	1	1	1	1	1	1	1	1	6FFFH
4	起始地址	1 1 1 0	0	0	0	0	0	0	0	0	0	0	0	7000H
	终了地址	1 1 1 0	1	1	1	1	1	1	1	1	1	1	1	77FFH

用线性片选法寻址时,地址是不连续的。例如从芯片1的终了地址到芯片2的起始地址之间跳过了地址 $4000_H \sim 57FF_H$; 芯片2到3跳过了地址 $6000_H \sim 67FF_H$ 。这样就使可寻址的地址减少了,也给编程带来一定的困难,需要特别注意各地址段的首末地址。由于这种连接方法比较简单,对于存储量小的系统,适宜采用。附带说明一点,在上述例子中地址线 A_{15} 的状态未被限定,会使存储器地址出现重叠区。例如,当 A_{14} 为低, A_{13} , A_{12} , A_{11} 为高时, A_{15} 为0或1任意值时,存储器1都被选中。所以实际上第一块芯片的地址既可以是 $3800_H \sim 3FFF_H$ (这时 $A_{15} = 0$); 也可以是 $B800_H \sim BFFF_H$ (这时 $A_{15} = 1$)。因此使用线性片选法时,存储器芯片地址有很大的重叠区。但在实际使用时,只要我们了解这一点情况,并不会妨碍正常使用的。

2. 译码片选法

采用译码的方法进行片选控制时,地址线不是直接去控制存储器的 $\overline{CS}$ 端。而是通过将有关地址线接在译码器的输入端,由译码器的输出端去控制存储器的片选控制端 $\overline{CS}$ 。在存储器片选控制中常用的译码器是3—8译码器。常见型号如74LS138。它的封装引脚及逻辑真值表见图1—6所示。它有A, B, C三个输入端, $\overline{y_0} \sim \overline{y_7}$ 8个输出端。输出端低电平表示有效。A, B, C三端有8种组合,每一种情况对应于一个输出端为低电平。此外,还有三个控制端: G_1 , $\overline{G_{2A}}$, $\overline{G_{2B}}$, 只有当 G_1 为高电平, $\overline{G_{2A}}$ 和 $\overline{G_{2B}}$ 为低电平时,芯片才允许工作,否则芯片输出端均为高电平,而不受A、B、C端的控制。3—8译码器也称八中取一译码器。

$\overline{G_{2A}}$	$\overline{G_{2B}}$	C	B	A	输出
1	0	0	0	0	$\overline{y_0} = 0$ 其余为1
1	0	0	0	1	$\overline{y_1} = 0$ 其余为1
1	0	0	1	0	$\overline{y_2} = 0$ 其余为1
1	0	0	1	1	$\overline{y_3} = 0$ 其余为1
1	0	1	0	0	$\overline{y_4} = 0$ 其余为1
1	0	1	0	1	$\overline{y_5} = 0$ 其余为1
1	0	1	1	0	$\overline{y_6} = 0$ 其余为1
1	0	1	1	1	$\overline{y_7} = 0$ 其余为1
不是上述情况	X	X	X	X	全高输出为

A 1

B 2

C 3

$\overline{G_{2A}}$ 4

$\overline{G_{2B}}$ 5

G_1 6

$\overline{y_0}$ 7

GND 8

74LS138

16 V_{CC}

15 $\overline{y_1}$

14 $\overline{y_2}$

13 $\overline{y_3}$

12 $\overline{y_4}$

11 $\overline{y_5}$

10 $\overline{y_6}$

9 $\overline{y_7}$

图 1—6 74LS138真值表及引脚

用译码器进行存储器片选的连接法举例如图1—7所示。

根据3—8译码器的特性,当 A_{15} , A_{14} 都为“0”时,译码器被选中。当 A_{13} , A_{12} , A_{11} 都为“0”时,输出端 $\overline{y_0}$ 为“0”,其余输出端均为1。它使第一组存储器的 $\overline{CS}$ 有效。即这一组存储器被选中。其余存储器芯片均因 $\overline{CS}$ 无效而未被选中。可以理解,这样连接的第一组存储器芯片的寻址范围是 $0000_H \sim 07FF_H$ 。当 A_{13} , A_{12} , A_{11} 由全“0”依次变为全“1”时,8组存储器将依次被选中,它们的地址范围可用类似方法确定。

由于CPU的地址和数据总线既与存储器也与各种外设相连,只有在CPU发出 $\overline{MREQ}$ 信号时,才是与存储器交换信息。故在进行片选控制时,还必须引入 $\overline{MREQ}$ 信号。另外,一般RAM都有一个读/写控制端 $\overline{WE}$,当它的输入信号为低电平时,存储器实现写操

作，当它为高电平时，则实现读操作。可用 CPU 的 $\overline{WR}$ 信号作为存贮器的 $\overline{WE}$ 控制信号。在下面叙述的具体例子中会看到有关这一点的实现方法。

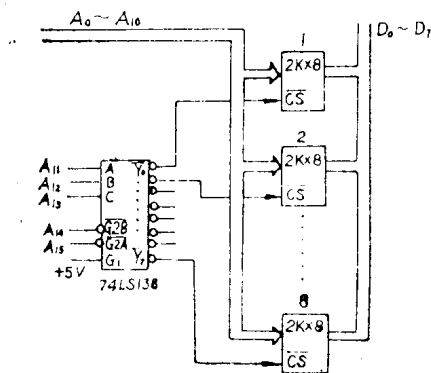


图 1—7 用译码器进行存贮器片选的连接

通常在单板微型机上可提供多到 16K 的存贮器。若需要的话，可对其余 48K，再进行外部扩充。下面举例具体说明单板机上 16K 存贮器容量分配及与 CPU 接口的具体方法。

设 16K 存贮器容量分配是：用三片 EPROM 2716，每片 2K 字节，共 6K；用八片 RAM2114，每四片是 2K 字节，共 4K。以上共占用 10K，其余 6K 留作机动。具体的地址分配如表 1—5 所示。

为了对这一地址分配中相应芯片进行片选控制，首先从相应地址范围的 16 条地址线状态找出每段地址的特征，然后寻求片选控制的具体方法。

表 1—6 示出了 0000H~3FFFH 这 16K 范围中相应每 2K 的地址线状态。从图中可发现如下规律：

- (1) 在这 16K 地址范围内 A_{15} 和 A_{14} 始终为“00”。并可推论得：

表 1—5 16K 存贮器容量分配举例

容量分配	芯 片 型 号	地 址 范 围
2K	EPROM2716(1片)	0000H~07FFH
2K	EPROM2716(1片)	0800H~0FFF
2K	EPROM2716(1片)	1000H~17FFH
2K	机 动	1800H~1FFFH
2K	RAM2114(4片)	2000H~27FFH
2K	RAM2114(4片)	2800~2FFFH
2K	机 动	3000H~37FFH
2K	机 动	3800H~3FFFH

在 4000H~7FFFH 的 16K 范围内 A_{15} 和 A_{14} 为“01”；
 在 8000H~BFFFH 的 16K 范围内 A_{15} 和 A_{14} 为“10”；
 在 C000H~FFFFH 的 16K 范围内 A_{15} 和 A_{14} 为“11”。

(2) 在 16K 地址范围中, 每 2K 地址范围的特征体现在地址线 A_{13} , A_{12} , A_{11} 三条线上。第一个 2K 区间, 这三条线是 000; 第二个 2K 区间, 这三条线是 001; ……第八个 2K 区间, 这三条线是 111。所以 A_{13} , A_{12} , A_{11} 三条地址线的状态是对每 2K 芯片进行片选控制的依据。

(3) 在每 2K 地址范围内, 又以地址线 A_{10} 的状态来划分前 1K 范围和后 1K 范围。例如在 $0000_H \sim 07FF_H$ 2K 范围中, 当 A_{10} 为“0”时, 对应 $0000_H \sim 03FF_H$ 这 1K 地址范围; 当

表 1—6 $0000_H \sim 3FFF_H$ 地址范围内每 2K 区间对应的地址线状态

十六进制	地 址 线 状 态															
地址范围	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0000_H }	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$07FF_H$	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
0800_H }	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
$0FFF_H$	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1000_H }	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
$17FF_H$	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
1800_H }	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
$1FFF_H$	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
2000_H }	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$27FF_H$	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
2800_H }	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
$2FFF_H$	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
3000_H }	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
$37FF_H$	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1
3800_H }	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
$3FFF_H$	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

A_{10} 为“1”时, 对应 $0400_H \sim 07FF_H$ 这 1K 地址范围。

(4) 每 1K 地址范围中, 地址线 $A_0 \sim A_9$ 可按每一具体地址给出相应的状态。

由地址线状态变化的规律可得到进行片选控制的具体方法。通常可采用前述的 3—8 译码器 74LS138 进行译码片选控制。对表 1—5 所示 16K 地址段中每 2K 地址范围的存储器进行片选控制时, 3—8 译码器输入端信号的具体接法如图 1—8 所示。

图 1—9 是对应于表 1—5 地址分配及具体所选用存储器芯片的具体接口连接图。图中

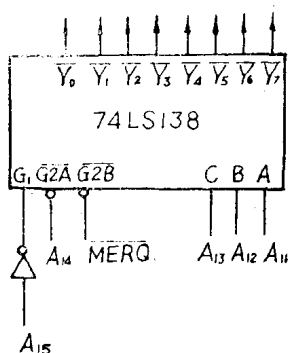


图 1—8 对0000H~3FFFH范围内每2K地址存储器片选时，3—8译码器的输入端信号接法

对 2114 每1K的片选端 $\overline{CS}$ 的控制是由 74LS138 的输出 $\overline{y}_4$ 、 $\overline{y}_5$ 和地址线 A_{10} 再作组合后形成的。

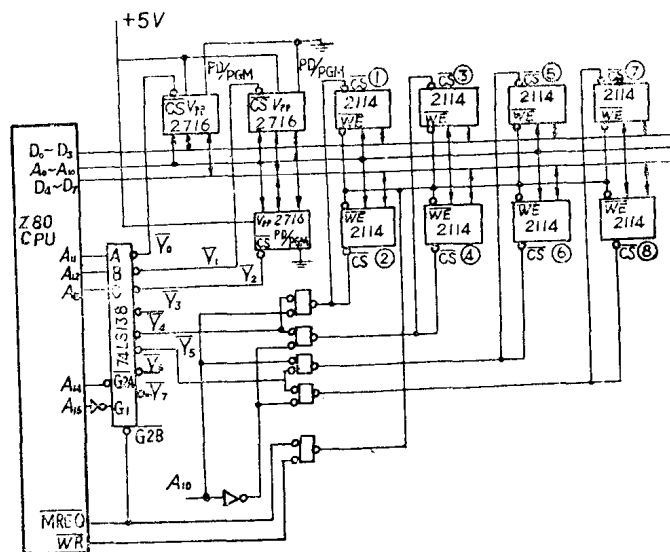


图 1—9 一个具体的存储器与CPU接口图

若 $\overline{y}_4$ 有效，且 A_{10} 地址线为“0”时，选中的是地址范围在 2000H~23FFH(1K)的 RAM-2114 ①、②二片芯片；若 $\overline{y}_4$ 有效且 A_{10} 地址线为“1”时，选中的就是地址范围在 2400H~27FFH(1K)的 RAM-2114 ③、④二片芯片。类似地，3—8译码器的 $\overline{y}_6$ 输出端与地址线 A_{10} 合起来选中 RAM-2114 芯片⑤、⑥或⑦、⑧。图中3—8译码器的 $\overline{y}_3$ 、 $\overline{y}_6$ 和 $\overline{y}_7$ 空着未接。因为表 1—5 规定相应这三个输出端的地址范围为机动。如果需要应用这三个地址范围时，可按图 1—9 类似方法接上 RAM 或 EPROM 等芯片。

第三节 存储器的扩充及总线驱动

微型计算机常用于某种专用系统，或对系统进行控制，或进行数据处理。这时，常将系统所配置的固定不变的程序存入 ROM 中；对试验性程序或一定时间后要加以更改的程序可写入 EPROM 中；在运行过程中要用到的数据，通常放在 RAM 中。系统所需存储器容量的大小取决于程序的复杂程度和数据的多少。如果设置在单板机上的 16K 存储器容量不能满足需要时，可以对 64K 容量中的其余 48K 作扩充连接。一般是以 16K 为一组加以