

# 计算机组成原理(下)

王 诚 编著

清华大学出版社

(京)新登字 158 号

## 内 容 简 介

本册共两章。第 4 章讲多级结构的存储器系统;第 5 章讲输入/输出设备与输入/输出系统。

本书是中央电大计算机专业的本科生教材。也可供其他高等院校用作计算机组成原理的教材。还可供从事计算机事业的工程技术人员参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: 计算机组成原理(下)

作 者: 王 诚

出版者: 清华大学出版社(北京清华大学学研楼,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 清华大学印刷厂

发行者: 新华书店总店北京发行所

开 本: 787× 1092 1/16 印张: 5.75 字数: 136 千字

版 次: 2000 年 5 月第 1 版 2000 年 5 月第 1 次印刷

书 号: ISBN 7-302-01314-4/TP·506

印 数: 0001~7600

定 价: 8.00 元

# 目 录

第 4 章 多级结构的存储器系统.....	201
本章主要教学内容.....	201
4.1 多级结构的存储器系统概述 .....	201
4.2 主存储器部件的组成与设计 .....	203
4.2.1 主存储器概述.....	203
4.2.2 动态存储器的记忆原理和读写过程.....	204
4.2.3 教学计算机的内存储器组成与设计.....	207
4.2.4 主存储器实现与应用中的几项技术.....	214
4.3 高速缓冲存储器部件 .....	215
4.3.1 高速缓冲存储器的运行原理.....	216
4.3.2 cache 的 3 种映像方式 .....	217
4.3.3 cache 存储器实用中的几个问题 .....	219
4.4 虚拟存储器部件 .....	223
4.4.1 段式存储管理.....	224
4.4.2 页式存储管理.....	226
4.5 外存设备与磁盘阵列技术 .....	228
4.5.1 外存设备概述.....	228
4.5.2 磁盘设备的组成与运行原理.....	231
4.5.3 磁带机设备组成简介.....	237
4.5.4 光盘设备的组成与运行原理.....	239
4.5.5 磁盘阵列技术与容错支持.....	240
习题与思考题.....	243
第 5 章 输入/输出设备与输入/输出系统.....	246
本章主要教学内容.....	246
5.1 输入/输出设备.....	247
5.1.1 输入/输出设备概述 .....	247
5.1.2 点阵式设备运行原理.....	247
5.1.3 显示器设备的组成与运行原理.....	249
5.1.4 打印机设备的组成与运行原理.....	256
5.1.5 计算机键盘运行原理.....	260
5.2 输入/输出系统.....	262
5.2.1 输入/输出系统概述 .....	262

5.2.2	计算机总线.....	263
5.2.3	输入/输出接口概述 .....	269
5.2.4	串行接口的内部组成与传送协议.....	271
5.2.5	并行接口芯片与并行输入/输出操作 .....	277
5.2.6	常用的输入/输出方式 .....	280
	习题与思考题.....	287

# 第 4 章 多级结构的存储器系统

## 本章主要教学内容

主存储器, 又称内存储器, 是传统计算机硬件系统的五大功能部件之一, 用于存储处在运行中的程序和相关数据, 其存储容量与读写速度等指标, 对计算机总体性能有重大影响。因此, 在现代的计算机系统中, 通常总是采用由三种运行原理不同, 性能差异很大的存储介质分别构建高速缓冲存储器、主存储器和虚拟存储器, 再将它们组成三级结构的统一管理、调度的一体化存储器系统。由高速缓冲存储器缓解主存储器读写速度慢, 不能满足 CPU 运行速度需要的矛盾; 用虚拟存储器(快速磁盘上的一片存储区)更大的存储空间, 解决主存储器容量小, 存不下更大程序与更多数据的难题。显而易见, 三级结构的存储器系统, 是围绕主存储器来组织和运行的, 就是说, 设计与运行程序是针对主存储器进行的, 充分表明主存储器在计算机系统中举足轻重的地位。

本章将首先介绍多级存储器系统的基本组成, 各级存储器所用介质的特性, 多级结构存储器结构应满足的原则, 以及它得以高效运行的原理; 其次分别介绍主存储器、高速缓冲存储器、虚拟存储器各自的组成与运行原理; 最后则介绍外存储设备(磁盘、磁带、光盘)的组成与使用, 并顺便介绍一些有关磁盘阵列的容错方面的知识。

本章概念性知识较多, 因此, 布置的作业较少; 教学实验是使用教学计算机系统来完成扩展由静态存储器芯片组成的主存储器的容量, 量测有关内存读写操作的时序波形等。

为了帮助记忆, 可否用如下几句话粗略概括本章教学内容纲要, 供同学们参考。

存储器, 三级穿, 局部、一致且包含, 提速、主体、扩容量, 缓存、主存、虚存盘; 字位扩展、体交叉, 完全、直接、组相联, 段表、页表和快表, 盘、带、阵列容错连。

### 4.1 多级结构的存储器系统概述

当前流行的计算机系统中, 广泛采用由三种运行原理不同、性能差异很大的存储介质, 来分别构建高速缓冲存储器、主存储器和虚拟存储器, 再将它们组成通过计算机硬软件统一管理与调度的三级结构的存储器系统, 如图 4.1 所示。

这种三级结构的存储器系统, 是围绕读写速度尚可、存储容量适中的主存储器来组织和运行的。用高速缓冲存储器缓解主存读写速度慢、不能满足 CPU 运行速度需要的矛盾; 用虚拟存储器更大的存储空间, 解决主存容量小、存不下规模更大的程序与更多数据的难题, 从而达到使整个存储器系统有更高的读写速度、尽可能大的存储容量、相对较低的制造与运行成本。高速缓冲存储器的问题是容量很小, 虚拟存储器的问题是读写速太慢。追求整个存储器系统有更高的性能/价格比的核心思路, 在于使用中充分发挥三级存储器各自的优势, 尽量避其短处。很显然, 存储相同数量的信息, 使用读写速度快的存储介

图 4.1 三级结构的存储器系统

质时,其价格通常总是要高一些。

这种三级结构的存储器系统的运行原理,或者说它可以有良好的性能/价格比,是建立在程序运行的局部性原理之上的。程序运行的局部性原理主要体现在以下 3 个方面:

- (1) 时间方面:在一小段时间内,最近被访问过的程序和数据很可能再次被访问。
- (2) 空间方面:这些最近被访问过的程序和数据,往往集中在一小片存储区域中。
- (3) 指令执行顺序方面:指令顺序执行比转移执行的可能性要大(大约为 5 : 1)。

这样就有可能把要使用的程序和数据,按其使用的急迫性和频繁程度,分时间段、分批量,合理地调入存储容量不同、读写速度不同的存储器部件中,并由计算机硬件、软件自动地统一管理调度。就是说,把 CPU 最近一小段时间要频繁、高速使用的信息存储在高速缓冲存储器中,可以快速完成读写操作,不至于拖慢 CPU 的运行速度,问题是信息数量不能太多,但在这一小段时间内也算够用了;把在略长一段时间内要用的较多信息存放在主存储器中,只是在 CPU 从高速缓冲存储器中找不到要用的信息时,才读速度较慢的主存储器,用时会长些,但找到这一信息的概率会大得多,在把得到的信息读入 CPU 的同时,还顺便将其写入到高速缓冲存储器中,并标明这一信息来自主存储器的哪个存储单元,下次再用到这一信息时,就不必再去读速度较慢的主存储器,而是快速地从高速缓冲存储器中直接得到;把那些暂时可以先不使用的信息保存在容量非常大的虚拟存储器中,用到时再从那里以更大的批量读入主存储器,读入的速度会慢得多,但确保再大的程序和再多数据总能分时、分批量地调入主存储器,且保证其正常运行。

在三级结构的存储器系统中,这三级不同的存储器中存放的信息必须满足如下两个原则:

(1) 一致性原则:同一个信息会同时存放在几个级别的存储器中,此时,这一信息在几个级别的存储器中必须保持相同的值。

(2) 包含性原则:处在内层(更靠近 CPU)存储器中的信息一定被包含在各外层的存储器中,即内层存储器中的全部信息一定是各外层存储器中所存信息中一小部分的副本。这是保证程序正常运行、实现信息共享、提高系统资源利用率所必须的,反之则不成立。例如,高速缓冲存储器中的信息,肯定也存放在主存储器中,还存放在虚拟存储器中,但主存储器中的非常多的信息不会同时在高速缓冲存储器中,虚拟存储器中的更多的信息也不

会同时出现在主存储器中。

三级不同的存储器,是用读写速度不同、存储容量不同、运行原理不同、管理使用办法也不尽相同的不同存储介质实现的。高速缓冲存储器使用静态存储器芯片实现,主存储器通常使用动态存储器芯片实现,而虚拟存储器则使用快速磁盘设备上的一片存储区。前者是半导体电路器件,以数字逻辑电路方式进行读写,后者则是在磁性介质层中通过电磁转换过程完成信息读写。表 4.1 以 1993 年在大型计算机上所用的存储器为例,给出它们的主要特性和性能指标。

表 4.1 1993 年在大型计算机上的存储器系统

存储器类型	存取时间	存储容量	存储成本	设备工艺	传送单位	带宽	分配管理
CPU	100ns	512B	1800 美分/KB	ECL	4~8B	400~800	编译器
缓存	(20~40)ns	128KB	72	256Kb SRAM	32B	250~400	硬件
主存	60~100ns	812MB	5.6	4Mb DRAM	0.5~1KB	80~33	O.S
虚存	10~20ms	60~228GB	0.23	1GB Disc	5~512KB	3~5	O.S/用户
后援	2~20min	512GB~2TB	0.01	5GB Tape	后援	0.18~0.23	O.S/用户

这里还给出了中央处理机(CPU)和后援存储器(磁带)的有关参数,便于对比,但通常不会把它们作为三级结构的存储器的组成部分来看待。

下面将分别介绍主存储器、高速缓冲存储器和虚拟存储器的有关知识,包括用到的半导体芯片的内部组成,3种存储器部件的组成和运行原理,以及使用中应该考虑的有关问题等。

## 4.2 主存储器部件的组成与设计

### 4.2.1 主存储器概述

主存储器是计算机硬件系统中的五大功能部件之一,用于存放正在运行中的程序和有关数据。它的读写速度和存储容量,对计算机系统的运行性能有至关重要的影响,经常成为影响系统运行性能的瓶颈。读写速度,通常用读、写一个存储单元必须的时间度量。例如 60ns,连续两次读写必需的时间间隔被称为存储周期,考虑到线路恢复的延迟问题,它应略大于一次主存读、写所用的时间。存储容量,通常用构成存储器的字节(8 bits)数或字数表述,一个存储字通常由 2, 4, 8 个字节组成。多数计算机都能在逻辑上同时支持按字或字节读写主存储器。

主存储器通过地址总线、数据总线、控制总线与计算机的 CPU 和外围设备连接在一起,如图 4.2 所示。

地址总线用于选择主存储器的一个存储单元(字或字节),其位数决定了可以访问的存储单元的最大数目,称为最大可寻址空间。例如,当按字节寻址时,20 位的地址可以访问 1MB 的存储空间,32 位的地址可以访问 4GB 的存储空间。

数据总线用于在计算机各功能部件之间传送数据,数据总线的位数(总线的宽度)与总线时钟频率的乘积,正比于该总线所支持的最高数据吞吐(输入/输出)能力。

控制总线用于指明总线的工作周期类型和本次入/出完成的时刻。总线的工作周期可以包括主存储器读周期,主存储器写周期,I/O设备读周期,I/O设备写周期,即区分要用哪个部件(主存或I/O设备)和操作的性质(读或写);还有直接存储器访问(DMA)总线周期等。若在计算机系统中使用了不同读写速度的主存储器,在CPU发出读写主存储器的命令后,它不能知晓读写操作完成的时刻,这是由被读写的存储器(或外围设备)本身的运行速度决定的,此时可以让主存储器本身提供读写完成的回答信号(Ready),CPU通过检测该信号来得知本次读写完成的时刻;若为读操作,有了该回答信号后,CPU就可以接收已读出的数据,如图4.2所示,这被称为CPU和主存储器按异步方式运行。

图 4.2 主存储器与其他部件的连接关系

主存储器可以由只读存储区(ROM)和读写存储区(RAM)两部分组成,是分别采用ROM和RAM存储器芯片实现的。ROM存储区用来储存内容固定不变的程序和数据,例如操作系统的内核部分,系统刚加电时运行的硬件诊断程序等。

从所用的半导体生产工艺区分,存储器芯片又可以分为静态存储器(SRAM)和动态存储器(DRAM)两种类型。由于动态存储器集成度高,生产成本低,被广泛地用于实现要求更大容量的主存储器。静态存储器读写速度快,生产成本高,通常多用其实现容量可以较小的高速缓冲存储器。静态存储器和动态存储器的不同之处主要表现如表4.2所示:

表 4.2 静态存储器和动态存储器主要性能比较

主要性能	SRAM	DRAM
存储信息	触发器	电容
破坏性读出	非	是
需要刷新	非	需要
行列地址	同时送	分两次送
运行速度	快	慢
集成度	高	低
发热量	大	小
存储成本	高	低

#### 4.2.2 动态存储器的记忆原理和读写过程

动态存储器,是用金属氧化物半导体(MOS)的单个MOS管来存储一个二进制位(bit)信息的。信息被存储在MOS管T的源极的寄生电容 $C_s$ 中,例如,用 $C_s$ 中存储电荷,若有电荷表示1,无电荷则表示0,如图4.3所示。

这一存储单元的运行情况是:当字线为高电平时,MOS管T导通,电源经电容与位线连通,依据位线上的电位是高还是低,以及电容 $C_s$ 上有无电荷的不同组合情况,MOS管T中会有电流流过或无电流流过两种不同状态;当字线为低电平时,MOS管T截止,电容 $C_s$ 上有无电荷的情况不会发生变化。

对这种单管存储单元的读写过程简述如下:

写数据时,先使字线为高电平,则 T 管导通,若数据线为低电平(写 1)且电容  $C_s$  中未存储电荷,则电源  $V_{DD}$  将向电容充电,使  $C_s$  中存储上一定数量的电荷,表示 1 信号已写入存储单元;若数据线为高电平(写 0)且电容  $C_s$  中已存储有电荷,则将会使电容  $C_s$  完成放电,使  $C_s$  中无存储电荷,表示 0 信号已写入存储单元;如写入的数据与  $C_s$  中原存储信息相同,则  $C_s$  中原存储有无电荷的情形不会发生变化。

图 4.3 MOS 单管存储单元线路

读数据时,先使数据线预充电至高电平,当字线的高电平到来后,则 T 管导通,若电容  $C_s$  中原存储有电荷(存储 1 信号),电容  $C_s$  就要放电,则会使数据线的电位由高变低;若  $C_s$  中未存储有电荷(存储 0 信号),数据线的电位不会变化。如果在数据线的一端接上一个高灵敏度的放大器,就可以检测出数据线上的这种不同的变化情况,从而区分出读出来的数据是 1 还是 0。请注意两件事情:一是读操作过程会使电容  $C_s$  中原存储的电荷(存储 1 信号)丢失,这就是通常说的动态存储器的破坏性读出的含意。为了保持原记忆内容,必须在读操作之后立刻跟随一次写回操作,这被称为预充电延迟。在预充电延迟完成之前,是不能开始下一次的读操作的,动态存储器的读写周期显然比它的数据读出时间长得多。二是向动态存储器的存储单元提供地址,是按先送行地址和再送列地址的次序分别完成的,由存储器芯片本身的时序信号控制完成的。另外一定要分别处理行列地址的原因,就是对动态存储器必须定期刷新。例如,每 2ms 应对所有存储单元至少进行一次刷新操作,来补偿电容  $C_s$  漏电现象造成的电荷损失,确保所存储的信息不会丢失,这种刷新操作是以行为单位执行的,此时只需提供行地址即可,与列地址无关。

由于动态存储器的读出信号很小,接在位线上的读出放大器必须有足够高的灵敏度,通常用具有极强正反馈能力的触发器线路实现。从原理上讲,触发器总应处在两个稳定状态中的一个状态,但若用某种办法首先使触发器的两个输出端保持同一电位。例如,将其两个输出端短接起来,则触发器的两个输出端即为同一电位;在突然断开这一连接关系的同时,再引入一个哪怕是很小的电位变化到该触发器的某一个输出端,必将引起触发器朝

图 4.4 用触发器实现的放大器电路

确定的方向翻转, 翻转后的触发器的状态即能表示读出的数据值。动态存储器的每一列都有自己的读出放大器, 这一运行过程, 如图 4.4 所示。

为了使位线的寄生电容对触发器两个输出端的影响相同(保持平衡), 也为了克服读出 0 信号时位线上无电位变化(即无电荷给触发器)的矛盾, 把存储器阵列中的所有列分成相等的两部分, 分别接在触发器两个输出端, 并在各端引入一个参考单元, 其电容上写入的内容总是 1, 其电荷量为存储器存储单元电荷量的  $1/2$ ; 在执行存储单元写入操作时, 送给接在触发器两个输出端的列的信号的高低电位情况应该相反。这样一来, 不管读出的信息是 0 还是 1, 也不管读出的存储器的存储单元接在触发器的哪个输出端, 触发器一定朝所设计的方向翻转, 即一定能区分出读出的数据是 0 还是 1, 这一运行过程, 如图 4.5 所示。

图 4.5 动态存储器的读写线路

动态存储器读写过程中, 行、列地址的建立时间, 读写信号的建立时间, 得到读出数据、提供写入数据的时间之间, 要满足规定的条件(技术指标)。我们不能在这里详细讨论这些线路与信号时序问题, 只能给出一些结论性的内容。

CPU 向动态存储器送地址, 是一次完成的, 由存储器芯片内部的行地址和列地址锁存器线路分先后接受行、列地址;

执行读操作时, 为了保证正常读出数据, 应在列地址锁存前建立读写命令信号/ $\overline{WE}$  (为高电位), 应在列地址锁存信号撤消后再结束/ $\overline{WE}$  信号, 读出数据送到  $D_{out}$  引脚。

执行写操作时, 读写命令信号/ $\overline{WE}$  (为低电位), 应在列地址锁存前建立, 写入数据  $D_{in}$  也应在列地址锁存前建立, / $\overline{WE}$  和写入数据应保持一定的时间。

动态存储器是破坏性读出, 读操作之后必须跟着执行写操作, 此时/ $\overline{WE}$  为先高(执行读)后低(执行写), 这一跳变时间要保证读、写操作的正常执行, 如图 4.6 所示。

图 4.6 动态存储器的读-回写工作方式时序波形图

### 4.2.3 教学计算机的内存储器组成与设计

#### 1. 静态存储器的存储原理和芯片内部结构

静态存储器(SRAM)是用触发器线路记忆与读写数据的,通常用 6 个 MOS 管组成存储一位二进制信息的存储单元。MOS 管  $T_1$  和  $T_3$ 、 $T_2$  和  $T_4$  分别组成一个反相器,两个反相器的输入、输出交叉耦合组成一位触发器,用于记忆一位二进制信息,如图 4.7 所示。例如, $T_1$ 管处于导通状态( $T_2$ 管一定处于截止状态),即它的输出端为低电平,则表示存储的是 1 信号,反之, $T_2$ 管处于导通状态,表示存储的是 0 信号。MOS 管  $T_5$  和  $T_6$  用于完成对该存储单元的读写操作,两个 MOS 管的一个极分别与触发器的两个输出端连接,两个管的栅极和字线连接,两个管的另外一个极分别与两条位线(位线 1、位线 2)连接。两条位线用来传送读、写的数据信号,字线用来提供把触发器与位线连通或是断开的控制信号。这一存储单元的读写过程如下:

读操作时,先使二位线充电至高电平,当字线送来高电平时,MOS 管  $T_5$  和  $T_6$  将导通,使触发器的两个输出端与位线 1 和位线 2 连通。若触发器存储的是 1 信号,即  $T_1$  管处于导通状态,它的输出端为低电平,则位线 1 就会经 MOS 管  $T_5$  产生流向  $T_1$  管的电流,从而在位线 1 上出现一个负脉冲,而位线 2 上就不会出现负脉冲;反之,若触发器存储的是 0 信号,即  $T_2$  管处于导通状态,它的输出端为低电平,则位线 2 就会经 MOS 管  $T_6$  产生流向  $T_2$  管的电流,从而在位线 2 上出现一个负脉冲,而位线 1 上就不会出现负脉冲;这样,就可以通过检查哪一条位线上出现一个负脉冲来判断触发器状态,即区分读出来的信号是 1 或是 0。

图 4.7 MOS 静态存储器的存储

写操作时,通过二条位线提供写入的数据信号。例如,写入 1 信号时,在位线 1 送低电平信号,在位线 2 送高电平信号,当字线送来高电平时, MOS 管  $T_5$  和  $T_6$  将导通,使触发器状态保持不变(已存储 1 信号),或使触发器状态翻转为 1 状态(原存储的是 0 信号)。要写入 0 信号,则需在位线 1 送高电平信号,在位线 2 送低电平信号。

要用这样的存储单元组成容量更大的一位存储器芯片,例如  $2048 \times 1$  的芯片,则需将它们组织成  $64 \times 32$  的矩阵形式,每个存储单元被连接到不同字线、位线的交叉点处,并加进读写控制电路,用地址译码器提供字线、位线选择信号,如图 4.8 所示。

图 4.8 2K 静态存储器芯片逻辑组成框图

为访问 2048 个存储单元,要用 11 位地址  $A_0 \sim A_{10}$ ,可以把  $A_0 \sim A_5$  用作 6 位行地址,经 X 译码器产生 64 个译码信号来选择 64 行;把  $A_6 \sim A_{10}$  用作 5 位列地址,经 Y 译码器产生 32 个译码信号来选择 32 列。这样,用 11 位地址  $A_0 \sim A_{10}$  就可以选中行、列交叉处的一个存储单元。为了用 Y 的译码信号选择一列,在每个存储单元处都要加两个 MOS 管  $T_7$  管、 $T_8$  管,用于选择把指定列的全部存储单元的  $T_5$  管、 $T_6$  管与该列的位线 1、位线 2 连通,而其他各列的全部存储单元都与对应列的位线 1、位线 2 断开。当一个存储单元被选中,它的字线选择将把该存储单元的  $T_5$  管、 $T_6$  管导通,列线选择将把该存储单元的  $T_7$  管、 $T_8$  管导通。若写允许/ $\overline{WE}$  信号为 0(低电平),执行写操作,把写入数据  $D_{IN}$  经  $T_5$  管、 $T_6$  管和  $T_7$  管、 $T_8$  管写入触发器;若/ $\overline{WE}$  信号为 1(高电平),执行读操作,触发器的状态经  $T_5$  管、 $T_6$  管及  $T_7$  管、 $T_8$  管和位线 1、位线 2 送到读出放大器,得到读出数据信号 DOUT。/ $\overline{CS}$  是片选信号,用于实现静态存储器的字数扩展控制。加上  $T_7$  管、 $T_8$  管后,一个存储单元总计用 8 个 MOS 管。

若一个存储器芯片的容量为  $2048 \times 8$ ,概念上,可以想象为一个芯片内有 8 套如图 4.8 所示的线路,实现中可以进行适当简化。

## 2. 教学计算机内存存储器的组成与设计

教学计算机采用单总线结构, 16 位的地址总线, 16 位的数据总线和简化的控制总线。考虑到扩展内存容量, 扩展的输入/输出接口, 中断和 DMA 接口等教学实验的实际需要, 实现中通过一条 50 芯的扁平电缆把地址总线的输出信号, 数据总线的输入/输出信号, 控制总线的信号引出教学计算机, 以便与支持扩展的教学实验板相连接。出于教学机本身的器件安全需要, 我们通过两片 74LS245 器件把数据总线隔断为内部总线 IB 与外部总线 DB 两部分。下面分别介绍它们的设计与功能。

### (1) 地址总线

地址总线 (记为 AB15 ~ AB0) 统一由地址寄存器 AR 驱动, 而地址寄存器只接收由 ALU 输出的信息。这是由教学计算机本身结构的特点所决定的。

地址总线要提供读写内存用的 16 位的地址, 读写输入/输出接口用的 8 位(地址总线低 8 位) 的入/出端口地址。地址总线的位数决定了可寻址的最大内存空间, I/O 端口地址的位数影响可接入系统的外设数目。教学机的指令格式和教学机本身的结构特性, 决定了将送往地址寄存器的地址信息只能经 ALU 送出。这是因为:

程序计数器 PC, 是用运算器内部的一个原为通用目的的寄存器实现的, 读取指令时, 只能将 PC 的内容经 ALU 送出。为读取双字指令的第二个字亦是如此。

教学机中的内存读写指令, 无论使用哪种寻址方式, 包括寄存器间接寻址, 变址寻址, 立即数寻址, 堆栈寻址(堆栈指针 SP 也是用运算器内的原为通用目的的一个寄存器实现的), 绝对地址寻址等, 寻址计算后的实际地址都是通过 ALU 送出的。

输入/输出指令中的入/出端口地址, 给出在指令寄存器的低位字节, 也是经 ALU 送往 AR。这与相对转移指令寻址计算时, 把在 IR 低位字节中的偏移值 offset 送 ALU 合用相同的通路。绝对地址寻址时, 也是把读出的地址先送 ALU, 这就与把内存读出的数据送运算器中通用寄存器合用了相同的一条传送路径, 以尽量减少所用的器件数量, 简化设计, 降低成本。

16 位地址总线的内容, 其高位将送往一个译码器线路, 用以产生主存芯片的片选信号, 其低位送到每个内存芯片, 用于选择芯片内的一个字节单元。

8 位入/出端口地址信息(在地址总线低 8 位) 中的几位将送到一个译码器线路, 用于产生入/出接口芯片的片选信号, 最低的几位将按接口芯片的要求送到芯片的相应引脚, 以选择接口内不同的寄存器, 更进一步的细节将在稍后部分另行说明。

### (2) 控制总线

控制总线的信号主要是经一片双 2-4 的译码器器件 74LS139 给出的。

控制总线的基本功能, 是用来指明总线周期的类型和本次入/出操作完成的时刻。教学机中的基本总线周期类型包括: 内存写, 内存读, 外设(接口) 写, 外设(接口) 读四类。分别用 /MMW, /MMR, /IOW, /IOR 四个信号标记, 还用 /MMREQ 和 /IOREQ 区分是内存工作还是外设工作。为支持动态微程序设计, 而设计了一个特别的用于写控存的总线周期, 用 SWA 标记。这些信号的产生是通过微指令中的三位控制码来确定的, 更深一步说, 这三位控制码又是依据内存读写、I/O 读写、写控存操作的需求来赋值的, 即 /MIO,

REQ, /WE 取值为 000, 001, 010, 011, 10X, 11X 时, 分别表示内存写, 内存读, I/O 写, I/O 读, 不操作(NC) 和控存写操作功能。

为简化设计与降低实现成本, 读写内存和读写外设接口均选用了与 CPU 操作同步运行的方案, 即内存和外设接口均可以在零等待状态完成读写操作, 故不再需要单独给出本次读写操作完成时刻的信息。

### (3) 数据总线

数据总线是在计算机各功能部件之间完成数据传送的线路, 它的工作速度(频率) 和位数(宽度) 决定了总线上最大的数据传送率。

在教学计算机的实现中, 出于保护教学机主要部件 CPU 的需要, 把教学机的数据总线通过双向三态门电路分割成内部数据总线 IB 和外部数据总线 DB 两部分。外部总线用于连接安装在教学机主板上的内存芯片和串行接口芯片的数据线引脚, 也通过扁平电缆去连接扩展实验板上的存储器芯片和其他入/出接口芯片的数据线引脚。

内部总线除接收教学机主板上 16 位数据开关的输入之外, 重点用于 CPU 内部信息传送, 包括接收指令寄存器 IR 的低位字节的内容, 状态寄存器的内容, 运算器输出的运算结果的内容, 中断向量寄存器的内容。内部总线上的信息可以送往指令寄存器 IR, 状态寄存器, 运算器的数据输入端 D, 和为了写控制存储器专门设置的四组寄存器 LDR。

设计和总线使用的核心技术, 是要保证在任何时刻只能把一组数据发送到总线上, 却允许一个和多个部件同时接收总线上的信息。为此, 对多个申请发送数据到总线上的部件, 必须进行申请仲裁, 确保它们通过分时共享总线。所用的电路通常为带有高阻状态输出的三态门电路, 如图 4.9 所示。

输入端标记为 in, 输出端标记为 out, /C 为控制端。当 /C 为低电平时, 该线路的功能是:  $out = in$ ; 当 /C 高电平时, 输出 out 则为高阻状态。我们很容易用这样的线路构建数据总线。如图 4.10 所示。

图 4.9 三态门电路

图 4.10 用三态门构建总线

把不同部件的输出经三态缓冲门连接到总线上, 通过一组控制信号  $/C_i (i = 1, 2, 3, \dots)$  完成选择送入总线的的数据。若做到在任何时刻, 在多个控制信号中只有一个为低电平, 而其他信号均为高电平, 则确保了多个部件能在不同的时刻分别向总线发送数据。若有多于 1 个的控制信号同时为低电平, 则是严重错误, 会造成多个缓冲门的输出短路, 使总线上的信息混乱, 甚至烧坏器件。为此, 用译码器线路给出这类控制信号是一种比较理想的方案。教学机中就使用三位编码 DC1, 并通过一片 3-8 译码器芯片来提供一组控制信号, 以控制不同部件向内部总线发送数据。

从总线上向外传送数据,重点是解决线路的负载与驱动能力问题。由于总线驱动的部件可能较多,又往往有较长的传送路径,要求总线所用器件有比较强的驱动能力。

#### (4) 系统时钟及时序

计算机各功能部件都是在时钟信号的“驱动”下一步一步地协调运行并执行各自的功能。从系统运行速度上考虑,总是希望在系统能正常运行的前提下,使系统时钟频率尽量高一些。

教学计算机不必追求运行速度。从支持串行接口芯片 Intel8251 所要求的时钟频率考虑,教学机选用了 1.8432 MHz 的晶振,3 分频后用 614.4 KHz 的时钟作为系统主时钟,其波形如图 4.11 所示。既用于驱动 CPU,也用于驱动 I/O 总线,并保持 CPU 与内存和 I/O 读写的同步运行。

图 4.11 系统时钟

CPU 内部的某些寄存器,通常用该时钟信号结束时的上升沿完成接收操作,包括微指令寄存器 PLR,这意味着每个时钟脉冲时间对应一条微指令的时间,即一个微周期,或称一个 CPU 周期。只有运算器内的通用寄存器是用时钟脉冲的低电平接收输入的。

在执行内存或 I/O 读写操作时,每个总线周期由两个时钟时间组成。第一个时钟时间,被称为地址时间,用于传送内存地址或 I/O 端口地址;第二个时钟时间,被称为数据时间,用于读写数据。由于所选的时钟频率相当低,一次数据时间足以完成读写操作,这就是常说的在总线的零等待状态完成数据读写。在另外一些计算机中,若一次数据时间不能完成内存或 I/O 设备的读写操作,就可以增加 1 到多个数据时间(多个时钟脉冲时间),这增加了的数据时间被称为总线的等待状态,它降低了系统的输入/输出能力。

#### (5) 静态存储器的字位扩展

教学计算机的内存储器用静态存储器芯片实现,由 2K(4K)字的 ROM 区和 2K 字的 RAM 区组成。内存字长 16 位,按字寻址方式读写。

实现中选用的是每片 2048 个存储单元,每个单元 8 位长度的 74LS2716 (ROM)、和每片 2048 个存储单元,每个单元 8 位长度 74LS6116 (RAM) 静态存储器芯片。为组成 16 位的存储器,必须用两片芯片完成存储器的字长扩展;为达到 4K 字的内存容量,还必须用两片芯片实现存储器的存储单元数量扩展。也就是说,用两片 74LS2716 芯片组成 2K 字的 ROM 存储区,地址分配在 0 ~ 2047 范围内,用两片 74LS6116 芯片组成 2K 字的 RAM 存储区,地址分配在 2048 ~ 4095 范围内。具体设计如图 4.12 所示。

横向的两个芯片用于用 8 位的芯片构成 16 位的存储器,它们是并列关系,其入/出数据线分别作为存储器的高低位字节的数据线,并同时完成读写操作。

纵向的两个芯片用于用 2KB 容量存储器芯片构成 4KB 存储单元的存储器,其数据入/出线对应位必须连接在一起,并使它们分时工作。当读写前 2K 单元的内容时,2716 芯

片正常运行,而 6116 芯片的数据线处于高阻状态;反之,当读写后 2K 单元内容时,也必须使 2716 的数据线处于高阻状态。

存储器芯片正常运行与否,取决于送给它的片选信号/ $\overline{CS}$ 。当/ $\overline{CS}$  信号为低,芯片被选中,该芯片的数据线与芯片内被选中的存储单元连通,能执行读写操作;当其为高,芯片的数据线处于高阻状态,逻辑上等于从系统中断离出来,不能进行读写。这意味着多个存储器芯片的数据线的对应位是可以直接连接在一起的,但需确保在任何时刻送给这些芯片的片选信号/ $\overline{CS}_i$ ( $i=1, 2, \dots$ ) 中只能有一个为低,其他均应为高。

图 4.12 教学计算机的存储器组成、设计

为读存储器,还必须向其提供被读内存单元的地址和读操作命令;为写存储器,必须向其提供要写存储单元的地址,要写入的数据和写操作命令。

综上所述,存储器读写涉及到地址信息,读出与写入的数据内容,片选信号和读写操作命令信号,它们之间在时序配合上要满足某些条件。例如,有了稳定的地址与片选信号才可以读,有了稳定的地址和写入的数据,再有了片选信号才能再给出写命令,以便保证无误的写操作。此外,这些信号应有一定的持续时间,以保证读写操作得以正常完成。

在教学计算机的实现中,为读写由 2048 个存储单元组成的存储器芯片内的一个单元,需把内存地址总线的低 11 位地址  $AB_{10} \sim AB_0$  送到每个芯片的地址引脚。为保证构成一个字的两个存储器芯片同时工作,必须向它们提供相同的读写控制信号和片选信号。对用于存储单元容量扩展的不同芯片,由于它们的数据线各相应位已直接连接在一起,就必须用互斥的片选信号来保证这些芯片分时运行。这些片选信号,是用地址总线高位的某些位经译码器电路提供出来的。例如,用地址总线的  $AB_{13}, AB_{12}$  和  $AB_{11}$  这三位经 3-8 译码器 74LS138 器件给出/ $\overline{CS}_0, \overline{CS}_1, \dots, \overline{CS}_7$  共 8 个信号。分别用于选择每 2K 字的存储器芯片执行读写操作。例如,我们把/ $\overline{CS}_0$  接到两片 2716 芯片的  $\overline{CS}$  引脚,把/ $\overline{CS}_1$  接到两片 6116 芯片的/ $\overline{CS}$  引脚,故 ROM 是教学机内存的前 2K 字, RAM 为后 2K 字。/ $\overline{CS}_2 \sim \overline{CS}_7$  留给扩展内存容量实验时使用。这里要强调指出:该译码器能否正常译码还要受内存读写请求的信号 / $\overline{MMREQ}$  的控制,以确保只有在有内存读写时才应该用内存片选信号

选择相应的内存芯片;以保证内存读写与 I/O 端口读写的互斥性,二者的数据线与地址线都分别接到了教学机的外部总线 DB 与地址总线 AB 上。简单推理也很容易想到,I/O 端口读写操作也存在类似的片选信号的处理问题。当然,既不读写内存,也不读写 I/O 端口是经常遇到的情形,此时,当然不应向内存与 I/O 端口给出任何一个片选信号。

对内存 ROM 区,只给出 /CS 信号即可,不再使用读写命令信号。

对内存 RAM 区,在给出 /CS 片选信号的同时,还需给出读写操作命令信号 /WE, /WE 为低是写,为高是读。6116 芯片还有一个 /OE 控制信号,我们已将其接地。

#### (6) 在按字寻址的存储器系统中实现按字节读写

教学计算机的内存储器是按字方式寻址的,每次读写操作都以字为单位进行,但在处理字符与字符串数据时,又必须支持字节寻址,以便在一个内存字中保存两个字符。在教学机中,一个字中的两个字符的次序是高位字节存前边(左侧)的字符,低位字节存后面(右侧)的字符。例如,“ABCD”四个字符,通常是用如图 4.13 所示的方式进行存储。

这表明字节地址中偶数地址对应字存储单元的高位字节,奇数地址对应字存储单元的低位字节。请注意,字节地址和字地址有如下关系:

一个字的高位字节的字节地址为该内存单元的字地址的 2 倍;一个内存字中的两个字节的字节地址的关系是:低位字节的字节地址为高位字节的字节地址值加 1。

图 4.13 字符存储  
和字节地址

有了上述说明,也就找到了在硬件按字寻址的存储器系统中,用软件实现的按字节寻址来完成字节读写的办法,有关思路说明如下:

字地址用 15 位表示时,字节地址就要使用 16 位。若使用 16 位的字节地址值不断加 1,则可以使其指向内存的每个字节单元。每加两次,则会从一个字指向另一个字。如果用 16 位的字节地址完成增 1 计数,用 16 位的字节地址的高 15 位进行内存读写,用字节地址的最低 1 位的值为 0 还是为 1 区分读写的字节应是高位字节(0)还是低位字节(1),全部问题就都解决了。

具体处理办法,可以采用通过寄存器与 C 进位触发器的循环移位指令来处理。若 R<sub>i</sub> 中存的是一个字地址(假定该地址最高一位的值为零且 C 的值也为零),在与 C 循环左移一位后,即得到该字中的头一个字节(高位)单元的字节地址,加 1 后则为后一个字节(低位)单元的字节地址。再加 1 则指向了下一个字的高位字节,如此等等。

有了字节地址,要实现读字节操作,首先应将保存有该字节地址的寄存器与 C 循环右移,则字节地址的最低 1 位(标记高低位字节)移入 C 触发器。而寄存器中则是包含了被读字节的那个内存字的字地址,首先用其实现按字读,再用保存在触发器 C 中的值从读出的一个字中选择相应一个字节就可以了。

写一个字节内容到内存中稍微复杂些。首先应把准备写入内存的一个字节保存在一个通用寄存器的相应字节中,并将该寄存器的另一个字节的值清为 0;再用相应的字地址读出有关内存字的内容,并清 0 其中要替换为新内容的那个字节,把另一个字节的值通过 OR 操作与被写字节拼装成一个字,然后写回相应内存的字单元即可。请注意,切不可用一个字节的内容直接进行内存字的写操作,因为这会破坏原内存字单元中的另一个字节