第一章　计算机系统概论

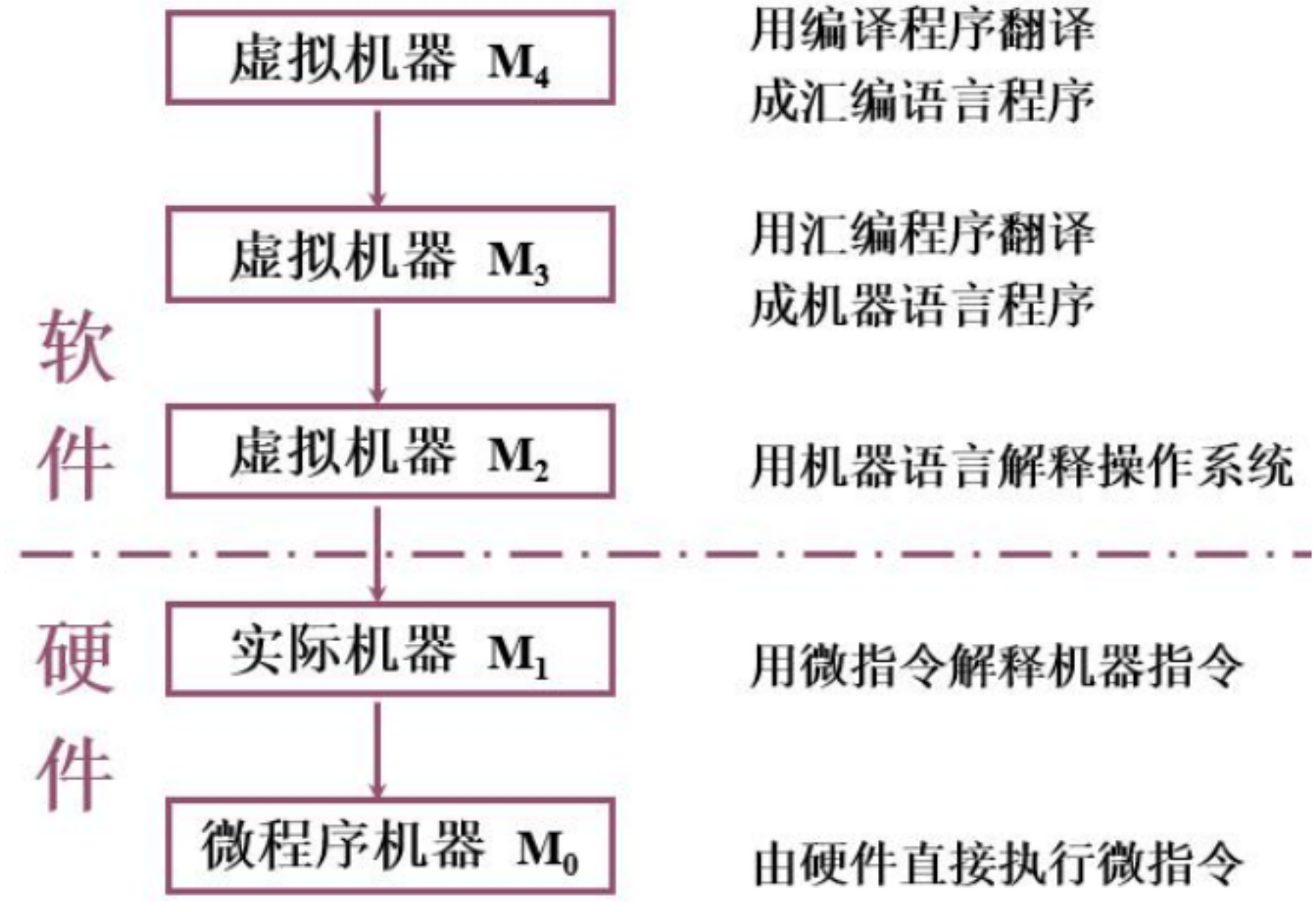
理解：计算机软硬件概念；　　P3

所谓“ 硬件 ”，是指计算机的实体部分，它由看得见摸得着的各种电子元器件，各类光、电、机设备的实物组成，如主机、外部设备等。

所谓“ 软件 ”，它看不见摸不着，由人们事先编制的各种具有各类特殊功能的程序组成。

理解：计算机系统的层次结构；

1.



掌握：计算机的基本组成；冯　　诺依曼计算机的特点；

计算机组成是指如何实现计算机体系结构所体现的属性，　　它包含了许多对程序员来说是透明的硬件细节。　P7

冯诺依曼计算机的特点　　P8

计算机由运算器、存储器、控制器、输入设备和输出设备五大部件组成。

指令和数据均用二进制数表示。

指令由操作码和地址码组成，　　操作码用来表示操作的性质，　　地址码用来表示操作数在存储器中的位置。

指令在存储器内按顺序存放。通常，指令是顺序执行的，在特定条件下，可根据运算结果或根据设定的条件改变执行顺序。

机器以运算器为中心，输入输出设备与存储器间的数据传送通过运算器完成。

掌握：高级语言、汇编语言、机器语言各自的特点

高级语言：这类语言对问题的描述十分接近人们的习惯，并且还具有较强的通用性。

汇编语言：　用符号表示操作，　　并用符号表示指令或数据所在存储单元的地址，　　使程序员可以不再使用繁杂而又易错的二进制代码来编写程序。

机器语言：　用户必须用二进制代码来编写程序。　　要求程序员对他们所使用的计算机硬件及其指令系统十分熟悉，编写程序难度很大，操作过程也极容易出错。

掌握：计算机的硬件框图及工作过程；

2014-1-13

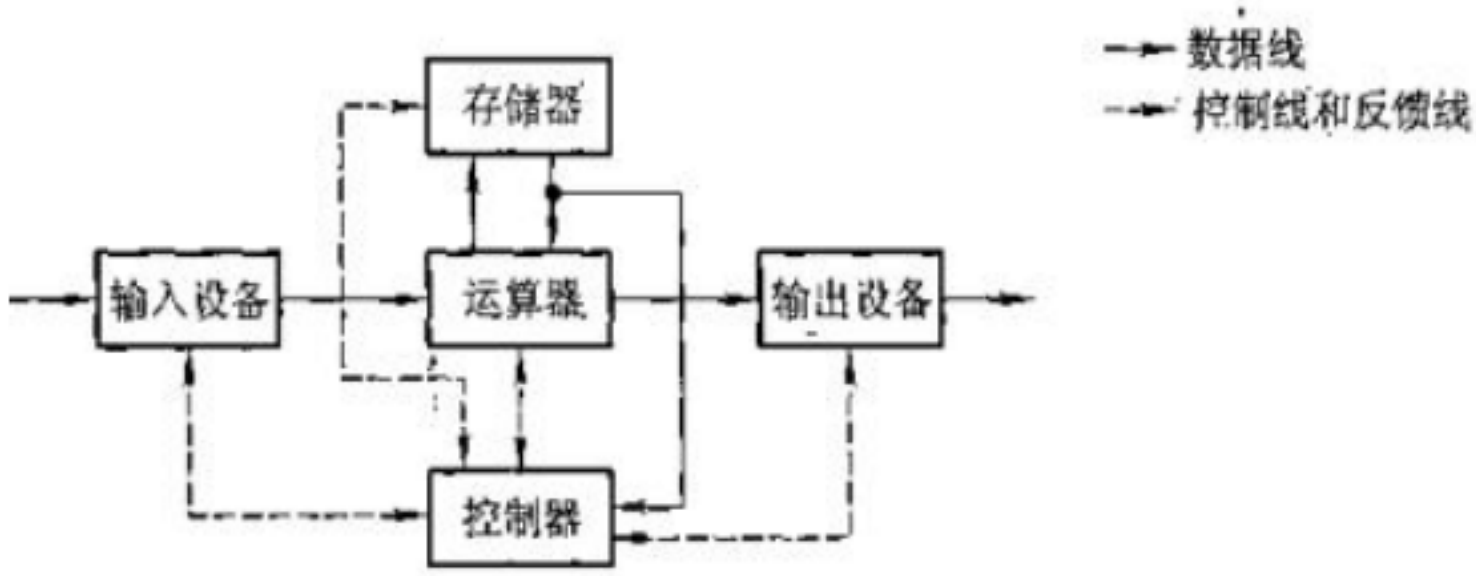


图 1.7　典型的冯·诺依曼计算机结构框图

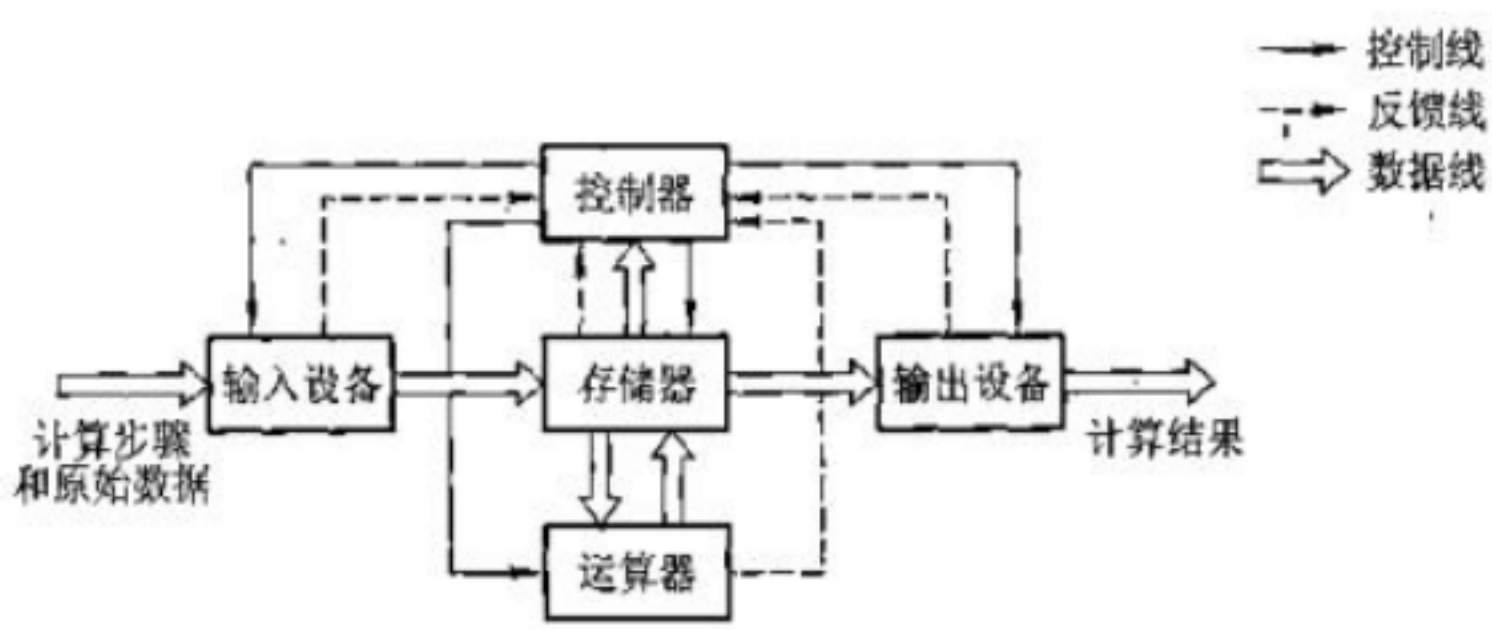


图 1.8　以存储器为中心的计算机结构框图

掌握：计算机硬件的主要技术指标。

机器字长：指　CPU 一次能处理的数据的位数，通常与　　CPU的寄存器位数有关。

存储容量 =存储单元个数　X 存储字长。

运算速度：

第二章　计算机的发展及应用

了解：计算机的产生、发展及应用

1.电子管计算机、晶体管计算机、集成电路计算机、大规模和超大规模集成电路计算机。

掌握：摩尔定律及其应用范围

第三章　系统总线　12（选择、填空）

理解： 总线的基本概念；　总线上信息的传送　（点对点传输、　广播传输、　串行传输、　并行传输）

1.总线：是连接多个部件的信息传输线，是各部件共享的传输介质。　　P41

理解：总线的分类；

2014-1-13

总线的分类： 按数据传送方式可分为并行传输总线和串行传输总线。 在并行传输总线中， 又可按传输数据宽度分为 8 位、 16 位、 32 位、 64 位等传输总线。若按总线的适用范围划分，则又有计算机（包括外设）总线、测控总线、网络通信总线等。

片内总线：是指芯片内部的总线，如在 CPU 芯片内部，寄存器与寄存器之间、寄存器与算逻单元 ALU 之间都由片内总线连接。

系统总线：是指 CPU、主存、 I/O 设备（通过 I/O 接口）各大部件之间的信息传输线。又称板级总线或板间总线。

按系统总线传输信息的不同，又可分为三类：数据总线、地址总线和控制总线。

了解：总线特性、性能指标、总线标准；

总线特征： P45

机械特征： 之宗现在机械连接方式上的一些性能， 如插头与插座使用的标准， 他们的几何尺寸、形状、引脚的个数以及排列的顺序，接头处的可靠接触等。

电气特征： 是指总线的每一根传输线上信号的传递方向和有效的电平范围。 通常规定由 CPU 发出的信号称为输出信号，送入 CPU 的信号成为输入信号。

功能特性：是指总线中每根传输线的功能，例如：地址总线用来指出地址码：数据总线用来传递数据；控制总线发出控制信号等。

时间特性：是指总线中的任一根线在什么时间内有效。

性能指标：总线宽度、总线带宽、时钟同步、总线复用、信号线数、总线控制方式、其他指标（负载能力、电源电压、总线宽度能否拓展）等。 P46

总线标准：可视为系统与各模块、模块与模块之间的一个互连的标准界面。 P47

目前流行的总线标准： ISA 总线、EISA 总线、 VESA 总线、 PCI 总线、 AGP 总线（显卡）、RS-232C 总线、 USB 总线。 P48

理解：总线结构；

1.总线结构通常可分为单总线结构和多总线结构两种。 P52

掌握：总线的判优控制（链式、计数器、独立请求）

总线判优控制可分为集中式和分布式两种，前者将控制逻辑集中在一处（如在 CPU 中），后者将控制逻辑分散在与总线连接的各个部件或设备上。 P57

常见的集中控制优先权仲裁方式：链式查询（最不稳定） 、计数器定时查询和独立请求方式（最稳定）。

掌握：总线通信控制的四种方式， 会计算波特率。

总线通信控制的四种方式：同步通信、异步通信、半同步通信和分离式通信。

计算波特率：

例 3.1 假设总线的时钟频率为 100 MHz，总线的传输周期为 4 个时钟周期，总线的宽度为 32 位，试求总线的数据传输率。若想提高一倍数据传输率，可采取什么措施？

解：根据总线时钟频率为 100 MHz，得

1 个时钟周期为 1/100 MHz = 0.01 μs

总线传输周期为 0.01 μs × 4 = 0.04 μs

由于总线的宽度为 32 位 = 4 B（字节）

故总线的数据传输率为 4 B/(0.04 μs) = 100 MBps

若想提高一倍数据传输率，可以在不改变总线时钟频率的前提下，将数据线的宽度改为 64 位，也可以仍保持数据宽度为 32 位，但使总线的时钟频率增加到 200 MHz。

例 3.2 在异步串行传输系统中，假设每秒传输 120 个数据帧，其字符格式规定包含 1 个起始位、7 个数据位、1 个奇校验位、1 个终止位，试计算波特率。

解:根据题目给出的字符格式,一帧包含 1 +7 +1 +1 =10 位

故波特率为(1 +7 +1 +1) ×120 =1 200 bps =1 200 波特

掌握概念：总线复用、异步通信

总线复用： 一条信号线上分时传送两种信号。 例如， 通常地址总线与数据总线在物理上是分开的两种总线，地址总线传输地址码，数据总线传输数据信息。为了提高总线的利用率，优化设计， 特将地址总线和数据总线共用一组物理线路， 在这组物理线路上分时传输地址信号和数据信号，即为总线的多路复用。 P46

异步通信： 异步通信克服了同步通信的缺点， 允许各模块速度的不一致性， 给设计者充分的灵活性和选择余地。 它没有公共的时钟标准， 不要求所有部件严格的统一操作时间， 而是采用应答方式（又称握手方式） ，即当主模块发出请求信号时，一直等待从模块反馈回来“ 响应 ”信号后，才开始通信。当然，这就要求主、从模块之间增加两条应答线。

异步通信的应答方式又可分为不互锁、半互锁和全互锁三种类型。 P61

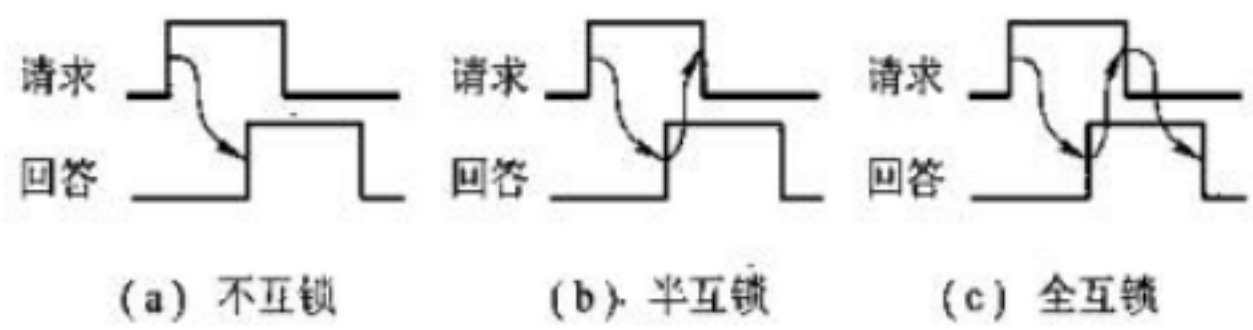


图 3.18 异步通信中请求与回答的互锁

第四章 存储器 26（选择、填空、简答或计算、设计）

掌握：存储器分类和存储器的层次结构（速度、容量、价格） ；

存储器分类： P68

按存储介质分类：半导体存储器、磁表面存储器、磁芯存储器、光盘存储器。

按存取方式分类：随机存储器 RAM、只读存储器 ROM、串行访问存储器

按在计算机中的作用分类：主存储器、辅助存储器、缓冲存储器。

层次结构：存储系统层次结构主要体现在缓存 -主存和主存 -辅存这两个存储层次上。 P71

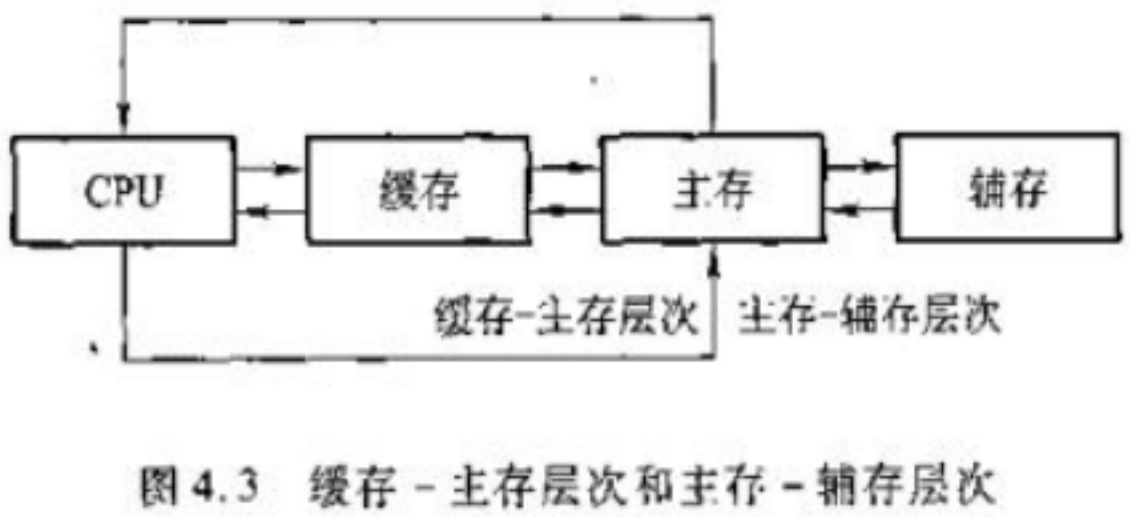


图 4.3 缓存 - 主存层次和主存 - 辅存层次

掌握：主存储器的基本组成、性能指标（容量、速度、带宽）

主存储器： 现代计算机的主存都由半导体集成电路构成， 图中的驱动器、 译码器和读写电路均制作在存储芯片中，而 MAR 和 MDR 制作在 CPU 芯片内。 P73

2014-1-13

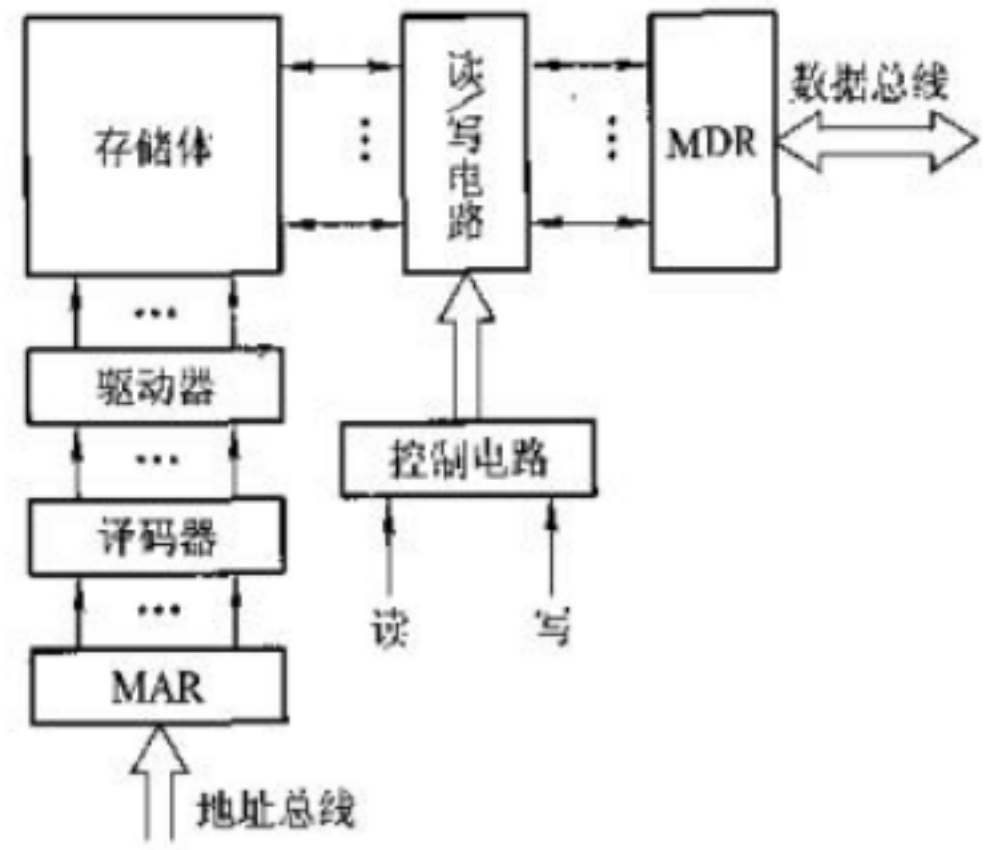


图 4.4　主存的基本组成

性能指标：速度、容量和每位价格。

掌握：半导体存储芯片的结构和译码驱动方式（线选法、重合法）　　P76

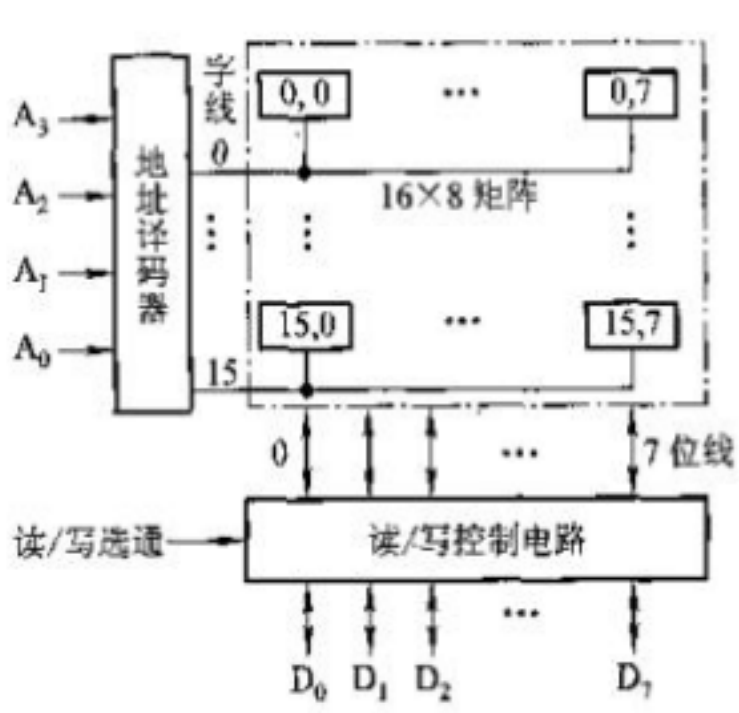


图 4.9　16 × 1 字节线选法结构示意图

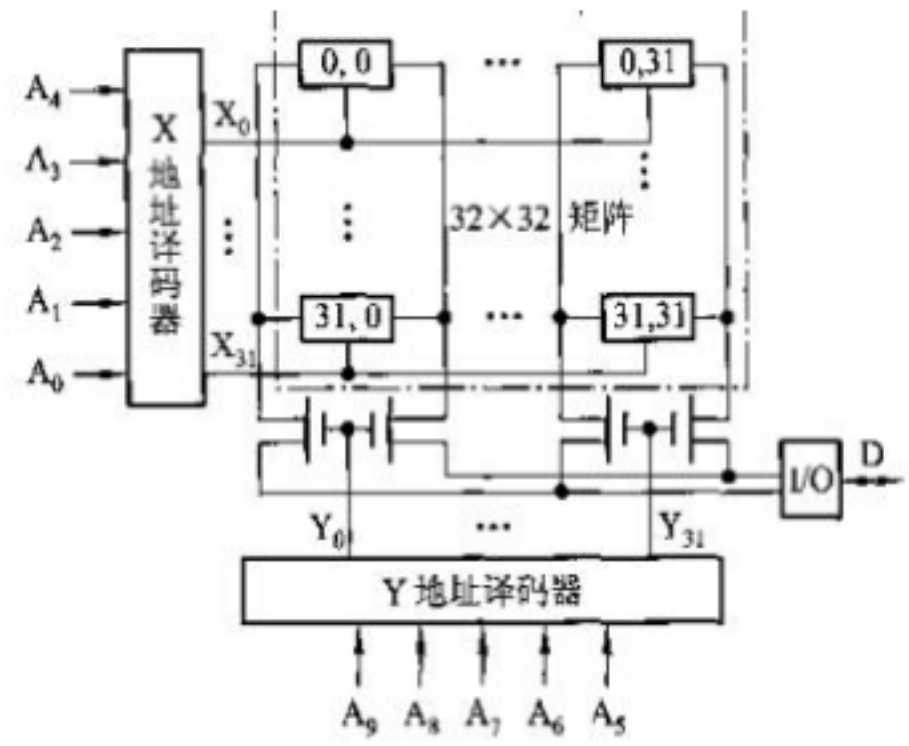


图 4.10　1 K × 1 位重合法结构示意图

理解： SRAM 和 DRAM 的读写原理

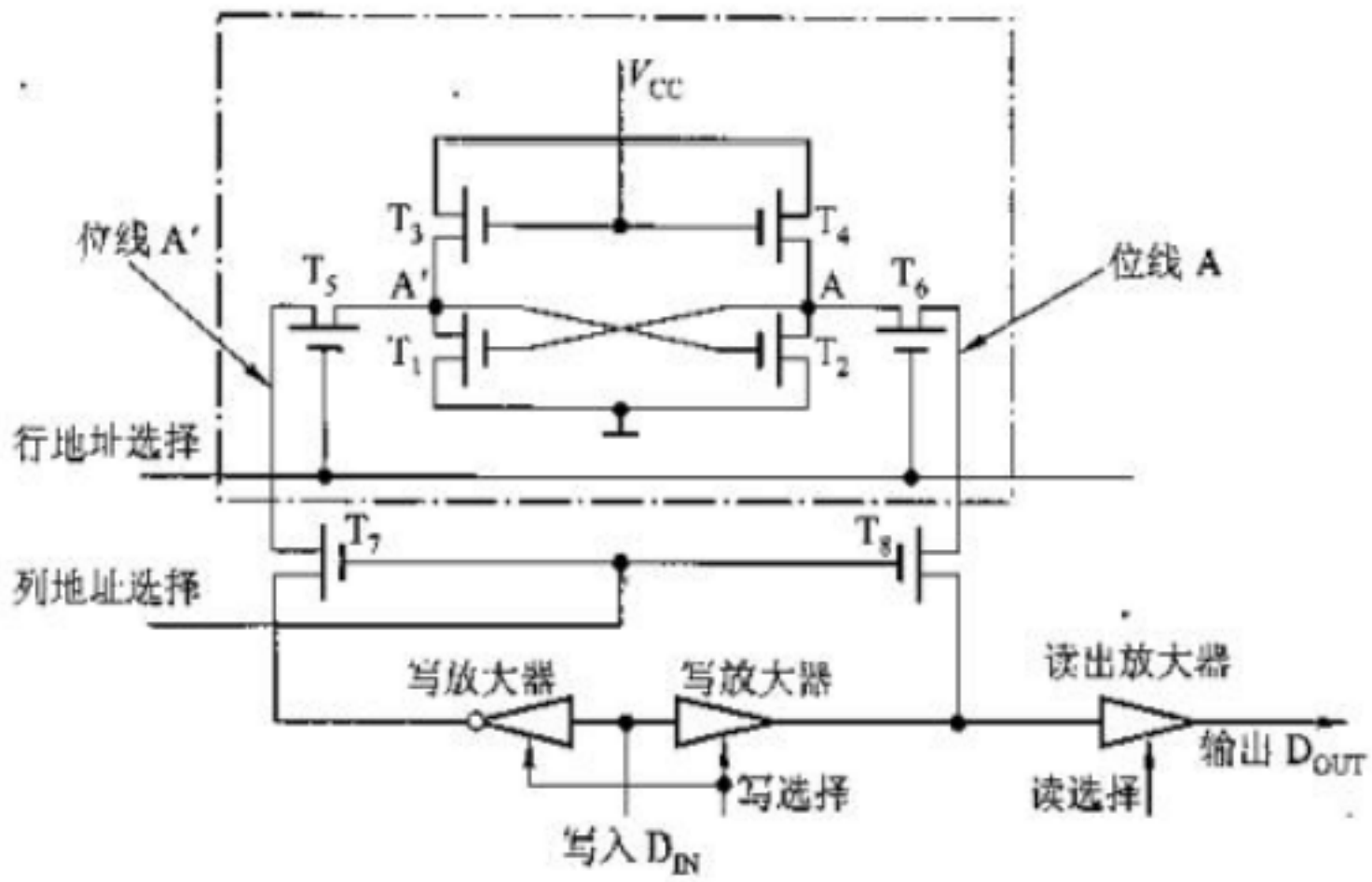


图 4.11　静态 RAM 的基本单元电路

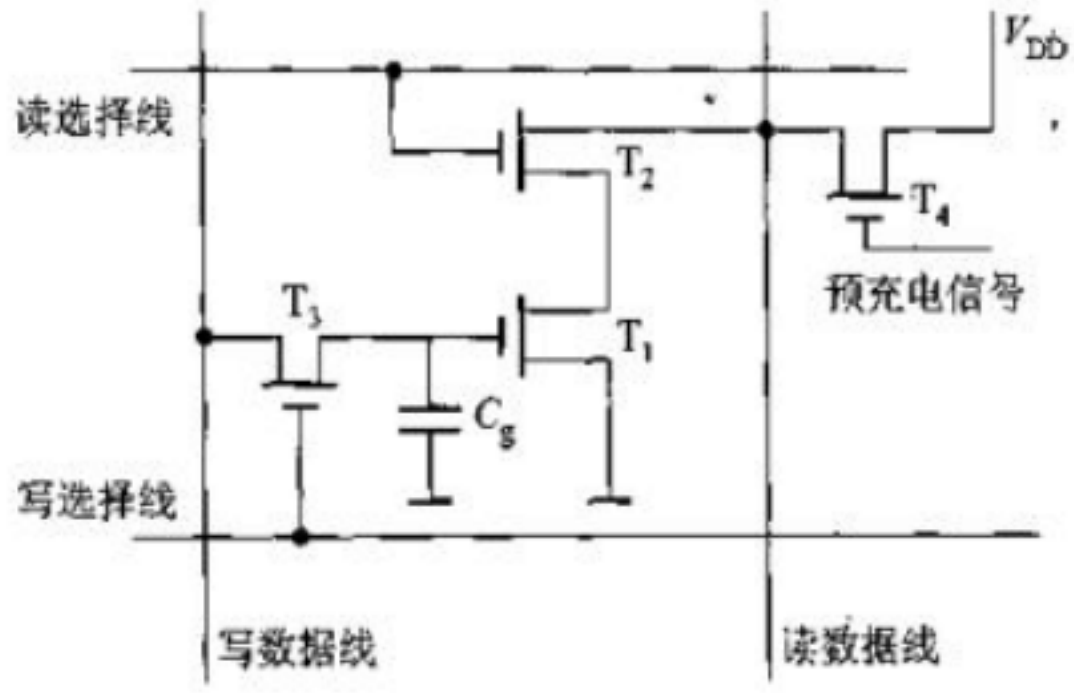


图 4.17　三管 MOS 动态 RAM 基本单元电路

掌握： DRAM 的刷新方式（集中、分散、异步）

集中刷新是在规定的一个刷新周期内,　　对全部存储单元集中一段时间逐行进行刷新,　　此刻必须停止读 /写操作。

集中刷新是在规定的一个刷新周期内,对全部存储单元集中一段时间逐行进行刷新,此刻必须停止读/写操作。例如,对 128×128 矩阵的存储芯片进行刷新时,若存取周期为 0.5 μs,刷新周期为 2 ms(占 4 000 个存取周期),则对 128 行集中刷新共需 64 μs(占 128 个存取周期),其余的 1 936 μs(共 3 872 个存取周期)用来读/写或维持信息,如图 4.24 所示。由于在这 64 μs 时间内不能进行读/写操作,故称为“死时间”,又称访存“死区”,所占比率为 128/4 000 × 100% = 3.2%,称为死时间率。

分散刷新：是指对每行存储单元的刷新分散到每个存取周期内完成。

异步刷新：

2014-1-13

异步刷新是前两种方式的结合，它既可缩短“死时间”，又充分利用最大刷新间隔为2 ms的特点。例如，对于存取周期为0.5 μs，排列成128×128的存储芯片，可采取在2 ms内对128行各刷新一遍，即每隔15.6 μs(2 000 μs÷128≈15.6 μs)刷新一行，而每行刷新的时间仍为0.5 μs，如图4.26所示。这样，刷新一行只停止一个存取周期，但对每行来说，刷新间隔时间仍为2 ms，而“死时间”缩短为0.5 μs。

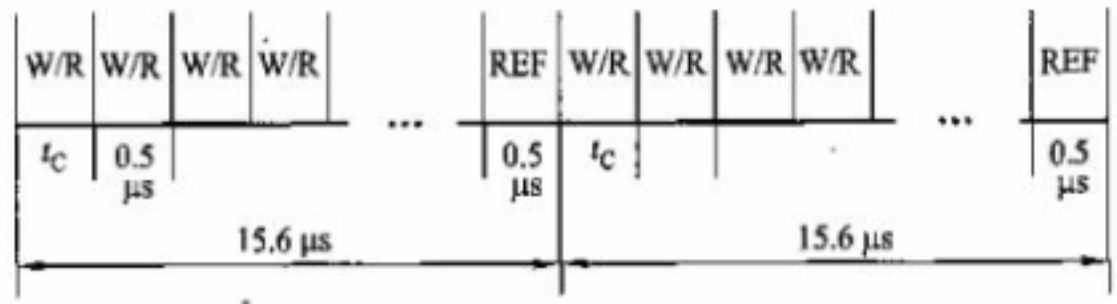


图4.26 异步刷新时间分配示意图

如果将动态RAM的刷新安排在CPU对指令的译码阶段，由于这个阶段CPU不访问存储器，所以这种方案既克服了分散刷新需独占0.5 μs用于刷新，使存取周期加长且降低系统速度的缺点，又不会出现集中刷新的访存“死区”问题，从根本上提高了整机的工作效率。

了解：只读存储器、可区分不同 ROM类型的使用特性（掌握如下缩写对应的中文 MROM、PROM、EPROM、EEPROM、FlashROM）

掩模 ROM（MROM）：用户无法改变原始状态。

PROM：是可以实现一次性编程的只读存储器，不得再修改。

EPROM：是一种可擦除可编程的只读存储器。 （紫外线照射只能一次全部擦除或者用电气方法可局部擦写）。

EEPROM: 电可擦除只读存储器。

FlashROM：闪存。

掌握：存储器容量与寻址范围计算

位扩展：是指增加存储字长，例如 2片1K*4位芯片可组成 1K*8位的存储器。

字扩展：是指增加存储器字的数量，例如 2片1K*4位芯片可组成 2K*4位的存储器。

字、位扩展： both

掌握：存储器与 CPU的连接（会设计、会画图） ；

12. 画出用 1024×4位的存储芯片组成一个容量为 64K×8位的存储器逻辑框图。 要求将 64K分成 4个页面， 每个页面分 16组，指出共需多少片存储芯片。解： 设采用 SRAM芯片， 则：

总片数 =（64K×8位） /（1024×4位）=64×2=128片

题意分析：本题设计的存储器结构上分为总体、页面、组三级，因此画图时也应分三级画。

首先应确定各级的容量：

页面容量 =总容量 /页面数 =64K×8/4 =16K×8位， 4片 16K×8字串联成 64K×8位

组容量 =页面容量 /组数 =16K×8位/16=1K×8位， 16片 1K×8位字串联成 16K×8位

组内片数 =组容量 /片容量 =1K×8位/1K×4位 =2片，两片 1K×4位芯片位并联成 1K×8位

存储器逻辑框图：

掌握：存储器的校验（奇偶校验、 CRC校验）；

掌握：顺序存储、交叉存储带宽计算

2014-1-13

$$t_2 = nT$$

例 4.6 设有 4 个模块组成的四体存储器结构，每个体的存储字长为 32 位，存取周期为 200 ns。假设数据总线宽度为 32 位，总线传输周期为 50 ns，试求顺序存储和交叉存储的存储器带宽。

解：顺序存储（高位交叉编址）和交叉存储（低位交叉编址）连续读出 4 个字的信息量是 32 × 4 = 128 位。

顺序存储存储器连续读出 4 个字的时间是

$$200\ \text{ns} \times 4 = 800\ \text{ns} = 8 \times 10^{-7}\ \text{s}$$

交叉存储存储器连续读出 4 个字的时间是

$$200\ \text{ns} + 50\ \text{ns} \times (4-1) = 350\ \text{ns} = 3.5 \times 10^{-7}\ \text{s}$$

顺序存储器的带宽是

$$128/(8 \times 10^{-7}) = 16 \times 10^{7}\ \text{bps}$$

交叉存储器的带宽是

$$128/(3.5 \times 10^{-7}) = 37 \times 10^{7}\ \text{bps}$$

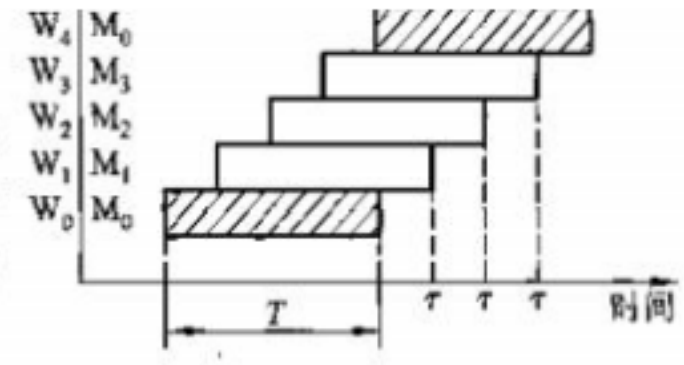


图 4.45 四体低位交叉编址存储器流水线工作方式示意图

掌握： Cache的基本结构及工作原理、 Cache-主存地址映射（直接、全相联、组相联） ；

掌握： Cache的命中率、平均访问时间、 Cache -主存系统效率的计算

例 4.7 假设 CPU 执行某段程序时，共访问 Cache 命中 20 00 次，访问主存 50 次。已知 Cache 的存取周期为 50 ns，主存的存取周期为 200 ns。求 Cache - 主存系统的命中率、效率和平均访问时间。

解：(1) Cache 的命中率为

$$2000/(2000+50) = 0.97$$

(2) 由题可知，访问主存的时间是访问 Cache 时间的 4 倍（200/50 = 4）。

设访问 Cache 的时间为 t，访问主存的时间为 $4t$，Cache - 主存系统的访问效率为 e，则

$$e = \frac{\text{访问 Cache 的时间}}{\text{平均访问时间}} \times 100\%$$

$$= \frac{t}{0.97 \times t + (1-0.97) \times 4t} \times 100\% = 91.7\%$$

(3) 平均访问时间为

$$50\ \text{ns} \times 0.97 + 200\ \text{ns} \times (1-0.97) = 54.5\ \text{ns}$$

理解： Cache的替换算法；

第五章 输入输出系统（选择）

了解：输入输出系统的发展概况及组成、 I/O 与主机的编址方式、传送方式、联络方式以及设备寻址；

早期阶段： I/O 设备与主存交换信息都必须通过 CPU。

接口模块和 DMA 阶段：这个阶段 I/O 设备通过接口模块与主机连接，计算机系统采用了总线结构。

具有通道结构的阶段：大中型机中采用 I/O 通道的方式来进行数据交换。

具有 I/O 处理机的阶段：

输入输出系统的组成：由 I/O 软件（ I/O 指令和通道指令）和 I/O 硬件组成。

2014-1-13

I/O 软件主要任务：
将用户编制的程序（或数据）输入主机内。
将运算结果输送给用户。
实现输入输出系统与主机工作的协调等。
I/O 设备编址方式：通常将 I/O 设备码看做地址码，对 I/O 地址码的编址可采用两种方式：统一编址或不统一编址。统一编址就是将 I/O 地址看作是存储器的一部分。不统一编址就是指 I/O 地址和存储器地址是分开的，所有对 I/O 设备的访问必须有专用的 I/O 指令。
设备寻址：由于每台设备都赋予一个设备号，因此，当要启动某一设备时，可由 I/O 指令的设备码字段直接指出该设备的设备号。 通过接口电路中的设备选择电路， 便可选中要交换信息的设备。
传送方式： 在同一瞬间， n 位信息同时从 CPU输出至 I/O 设备，或由 I/O 设备输入至 CPU, 这种传送方式称为并行传送。 其特点是传送速度较快， 但要求数据线多。 若在同一瞬间只传送一位信息，在不同时刻连续逐位传送一串信息，这种传送方式成为串行传送。
联络方式：立即响应方式、异步工作采用应答信号联络、同步工作采用同步时标联络。
理解： I/O 接口的功能及基本组成；程序查询方式的工作原理及程序查询接口电路；程序中断方式的工作原理（中断向量的作用）及程序中断接口电路、 DMA 方式的特点。
第六章 计算机的运算方法 17 （选择、填空、判断、简答或计算）
掌握：计算机中有符号数（原码、补码（变形补码） 、反码、移码）和无符号数的表示；
原码：符号位用 0 和 1 表示，数值位即真值的绝对值。
反码：符号位不变，原码的数值位按位取反。
补码：符号位不变，反码的数值位 +1。
移码： 例如比较 21 和-21 两个数， 两个数的反码分别为 10101 和-10101， 两个数的补码分别为 10101 和 101011，直接比较补码会出现 -21>21，所以给其反码加上 2 的 5 次方，得 10101+100000=110101， -10101+100000=001011，再比较即可得 21>-21.由此移码就是 [X]移=2n+x（2n>x≥-2n）
掌握：计算机中数的定点表示和浮点表示 （精度、数值范围与尾数及阶码的关系、 溢出条件），浮点数的规格化， IEEE754标准；
掌握： 定点运算 （算术移位和逻辑移位、 补码加减、 原码一位乘法、 补码一位乘法 （校正法、Booth 算法） 、 定点四则运算的硬件实现） ；
掌握：浮点四则运算（加、减、乘、除） ；

1. 对阶

对阶的目的是使两操作数的小数点位置对齐，即使两数的阶码相等。为此，首先要求出阶差，再按小阶向大阶看齐的原则，使阶小的尾数向右移位，每右移一位，阶码加 1，直到两数的阶码相等为止。右移的次数正好等于阶差。尾数右移时可能会发生数码丢失，影响精度。

例如，两浮点数 $x=0.1101\times2^{01}$，$y=(-0.1010)\times2^{11}$，求 $x+y$。

首先写出 x,y 在计算机中的补码表示。

$[x]_{补}=00,01;\ 00.1101$，$[y]_{补}=00,11;\ 11.0110$

在进行加法前，必须先对阶，故先求阶差：

$$[\Delta_j]_{补}=[j_x]_{补}-[j_y]_{补}=00,01+11,01=11,10$$

即 $\Delta_j=-2$，表示 x 的阶码比 y 的阶码小，再按小阶向大阶看齐的原则，将 x 的尾数右移两位，其阶码加 2，

得 $$[x]'_{补}=00,11;\ 00.0011$$

此时，$\Delta_j=0$，表示对阶完毕。

2. 尾数求和

将对阶后的两个尾数按定点加(减)运算规则进行运算。

如上例中的两数对阶后得

$$[x]'_{补}=00,11;\ 00.0011$$
$$[y]_{补}=00,11;\ 11.0110$$

则 $[S_x+S_y]_{补}$ 为

$$\begin{array}{rl} 00.0011 & [S_x]'_{补} \\ +11.0110 & [S_y]_{补} \\ \hline 11.1001 & [S_x+S_y]'_{补} \end{array}$$

即 $$[x+y]_{补}=00,11;\ 11.1001$$

3. 规格化

由 6.2.2 节可知，当基值 $r=2$ 时，尾数 S 的规格化形式为

$$\frac{1}{2}\leqslant|S|<1 \tag{6.19}$$

如果采用双符号位的补码，则

当 $S>0$ 时，其补码规格化形式为

$$[S]_{补}=00.1\times\times\cdots\times \tag{6.20}$$

当 $S<0$ 时，其补码规格化形式为

$$[S]_{补}=11.0\times\times\cdots\times \tag{6.21}$$

可见，当尾数的最高数值位与符号位不同时，即为规格化形式，但对 $S<0$ 时，有两种情况需特殊处理。

① $S=-\frac{1}{2}$，则 $[S]_{补}=11.100\cdots0$。此时对于真值 $-\frac{1}{2}$ 而言，它满足式(6.19)，对于补码

($[S]_{补}$)而言,它不满足于式(6.21)。为了便于硬件判断,特规定 $-\frac{1}{2}$不是规格化的数(对补码而言)。

② $S=-1$,则$[S]_{补}=11.00\cdots0$,因小数补码允许表示 -1,故 -1 视为规格化的数。

当尾数求和(差)结果不满足式(6.20)或式(6.21)时,则需规格化。规格化又分左规和右规两种。

(1) 左规

当尾数出现 00.0×××…×或 11.1×××…×时,需左规。左规时尾数左移一位,阶码减 1,直到符合式(6.20)或式(6.21)为止。

如上例求和结果为

$$[x+y]_{补}=00,11;\ 11.1001$$

尾数的第一数值位与符号位相同,需左规,即将其左移一位,同时阶码减 1,得

$$[x+y]_{补}=00,10;\ 11.0010$$

则

$$x+y=(-0.1110)\times2^{10}$$

(2) 右规

当尾数出现 01.×××…×或 10.×××…×时,表示尾数溢出,这在定点加减运算中是不允许的,但在浮点运算中这不算溢出,可通过右规处理。右规时尾数右移一位,阶码加 1。

($[S]_{补}$)而言,它不满足于式(6.21)。为了便于硬件判断,特规定 $-\frac{1}{2}$不是规格化的数(对补码而言)。

② $S=-1$,则$[S]_{补}=11.00\cdots0$,因小数补码允许表示 -1,故 -1 视为规格化的数。

当尾数求和(差)结果不满足式(6.20)或式(6.21)时,则需规格化。规格化又分左规和右规两种。

(1) 左规

当尾数出现 00.0×××…×或 11.1×××…×时,需左规。左规时尾数左移一位,阶码减 1,直到符合式(6.20)或式(6.21)为止。

如上例求和结果为

$$[x+y]_{补}=00,11;\ 11.1001$$

尾数的第一数值位与符号位相同,需左规,即将其左移一位,同时阶码减 1,得

$$[x+y]_{补}=00,10;\ 11.0010$$

则

$$x+y=(-0.1110)\times2^{10}$$

(2) 右规

当尾数出现 01.×××…×或 10.×××…×时,表示尾数溢出,这在定点加减运算中是不允许的,但在浮点运算中这不算溢出,可通过右规处理。右规时尾数右移一位,阶码加 1。

例 6.29 已知两浮点数 $x=0.1101\times2^{10}$,$y=0.1011\times2^{01}$,求 $x+y$。

解:x、y 在机器中以补码表示为

$$[x]_{补}=00,10;\ 00.1101$$

$$[y]_{补}=00,01;\ 00.1011$$

① 对阶:

$$[\Delta_j]_{补}=[j_x]_{补}-[j_y]_{补}$$

理解： ALU的工作原理；

理解：进位链结构。

2014-1-13

第七章　指令系统　14　（选择、填空、判断）

掌握：机器指令的一般格式；扩展操作码技术；

机器指令的一般格式：指令由操作码和地址码两部分组成。

例 7.1　假设指令字长为 16 位，操作数的地址码为 6 位，指令有零地址、一地址、二地址三种格式。

(1) 设操作码固定，若零地址指令有 P 种，一地址指令有 Q 种，则二地址指令最多有几种？

(2) 采用扩展操作码技术，若二地址指令有 X 种，零地址指令有 Y 种，则一地址指令最多有几种？

解：(1) 根据操作数地址码为 6 位，则二地址指令中操作码的位数为 $16-6-6=4$。这 4 位操作码可有 $2^4=16$ 种操作。由于操作码固定，则除去了零地址指令 P 种，一地址指令 Q 种，剩下二地址指令最多有 $16-P-Q$ 种。

(2) 采用扩展操作码技术，操作码位数可变，则二地址、一地址和零地址的操作码长度分别为 4 位、10 位和 16 位。可见二地址指令操作码每减少一种，就可多构成 2^6 种一地址指令操作码；一地址指令操作码每减少一种，就可多构成 2^6 种零地址指令操作码。

因二地址指令有 X 种，则一地址指令最多有 $(2^4-X)\times 2^6$ 种。设一地址指令有 M 种，则零地址指令最多有 $[(2^4-X)\times 2^6-M]\times 2^6$ 种。

根据题中给出零地址指令有 Y 种，即

$$Y=[(2^4-X)\times 2^6-M]\times 2^6$$

则一地址指令

$$M=(2^4-X)\times 2^6-Y\times 2^{-6}$$

在设计操作码不固定的指令系统时，应尽量考虑安排指令使用频度（即指令在程序中出现的概率）高的指令占用短的操作码，对使用频度低的指令可占用较长的操作码，这样可以缩短经常使用的指令的译码时间。当然，考虑操作码长度时也应考虑地址码的要求。

了解：操作数类型和操作类型；

操作数类型：地址、数字、字符、逻辑数据等。

操作类型：数据传送、算术逻辑操作、移位、转移（无条件转移、条件转移、调用与返回、陷阱与陷阱指令）、输入输出、其他（等待指令、停机指令、空操作指令、开中断指令、关中断指令、置条件码指令等）。

二地址指令：存储器 -存储器型、寄存器 -寄存器型、存储器 -寄存器型执行速度区别

掌握：寻址方式；程序计数器的作用

寻址方式：指令寻址和数据寻址两类。

指令寻址分为顺序寻址和跳跃寻址两种。

数据寻址：

立即寻址：操作数本身设在指令字内，即形式地址 A 不是操作数的地址，而是操作数本身，又称为立即数。它的优点在于只要取出指令，便可立即获得操作数，这种指令在执行阶段不必再访问存储器。

直接寻址：指令字中的形式地址 A 就是操作数的真实地址。它的优点是寻找操作数比较简单，也不需要专门计算操作数的地址，在指令执行阶段对主存只访问一次。他的缺点在于 A 的位数限制了操作数的寻址范围，而且必须修改 A 的值，才能修改操作数的地址。

隐含寻址：是指指令字中不明显地给出操作数的地址，其操作数的地址隐含在操作码或某个寄存器中。由于隐含寻址在指令字中少了一个地址，因此，这种寻址方式有利于缩短指令字长。

2014-1-13

间接寻址： 有效地址是由形式地址间接提供的。与直接寻址相比，它扩大了操作数的寻址范围，并且便于编制程序。

寄存器寻址：在寄存器寻址的指令字中，地址码字段直接给出了寄存器的编号，即 EA=Ri。其操作数在由 Ri 所指定的寄存器内。由于操作数不在贮存中，故寄存器寻址在指令执行阶段无须访存，减少了执行时间。由于地址字段只需指明寄存器编号，故指令字较短，节省了存储空间，因此寄存器寻址在计算机中得到了广泛应用。

寄存器间接寻址：

基址寻址： 基址寻址须设有基址寄存器 BR，其操作数的有效地址 EA 等于指令字中的形式地址与基址寄存器中的内容相加。

变址寻址

相对寻址：相对寻址的有效地址是将程序计数器 PC的内容与指令字中的形式地址 A 相加而成。

堆栈寻址

程序计数器的作用：用来存放下一条指令的地址的。当执行一条指令时，首先需要根据 PC 中存放的指令地址，将指令由内存取到指令寄存器中，此过程称为“ 取指令 ” 。与此同时，PC中的地址或自动加 1 或由转移指针给出下一条指今的地址。此后经过分析指令，执行指令。完成第一条指令的执行，而后根据 PC 取出第二条指令的地址，如此循环，执行每一条指令！

了解： RISC技术。

1.RISC即精简指令集系统计算机。 RISC技术是用 20%的简单指令的组合来实现不常用 80%的那些指令的功能。 在提高性能方面， RISC技术还采用了许多有效措施， 最有效的方法就是减少指令的执行周期数。

第八章 CPU的结构和功能 17 （选择、填空、判断、简答或计算）

掌握： CPU 的结构；

CPU实质包括运算器和控制器两大部分。

控制器基本功能：取指令、分析指令、执行指令。此外控制器还必须能控制程序的输入和运算结果的输出以及对总线的管理，甚至能处理机器运行过程中出现的异常情况和特殊请求，即处理中断的能力。

总之， CPU必须具有控制程序的顺序执行（称指令控制） 、产生完成每条指令所需的控制命令（称操作控制） 、对各种操作加以时间上的控制（称时间控制） 、对数据进行算术运算和逻辑运算（数据加工）以及处理中断等功能。

掌握：指令周期（概念、数据流） ；

指令周期： CPU没取出并执行一条指令所需的全部时间称为指令周期，也即 CPU完成一条指令的时间。

数据流：

2014-1-13

1. 取指周期的数据流

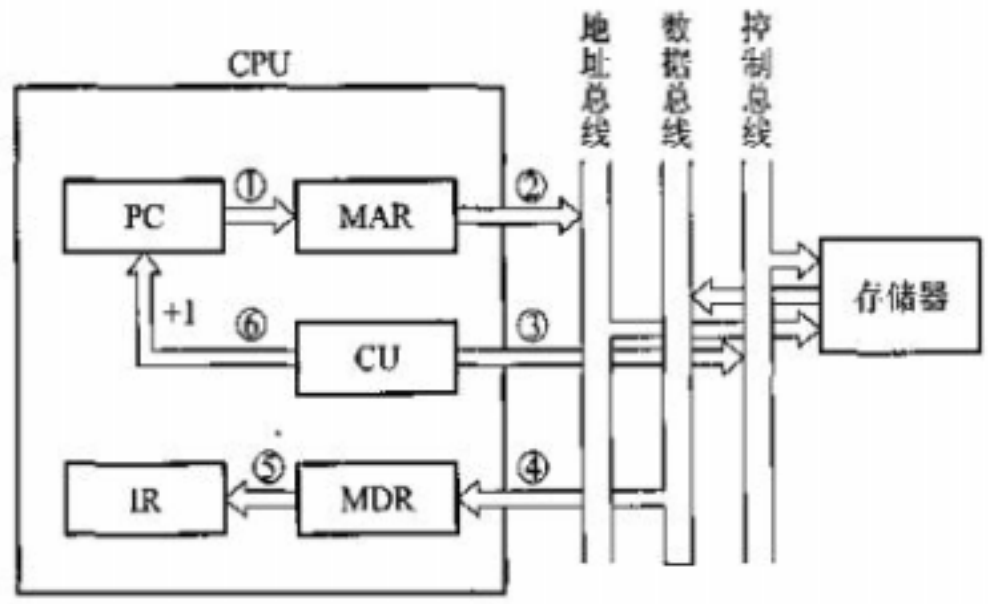


图 8.10 取指周期数据流

图 8.10 所示的是取指周期的数据流。PC 中存放现行指令的地址，该地址送到 MAR 并送至地址总线，然后由控制部件 CU 向存储器发读命令，使对应 MAR 所指单元的内容（指令）经数据总线送至 MDR，再送至 IR，并且 CU 控制 PC 内容加 1，形成下一条指令的地址。

2. 间址周期的数据流

间址周期的数据流如图 8.11 所示。一旦取指周期结束，CU 便检查 IR 中的内容，以确定其

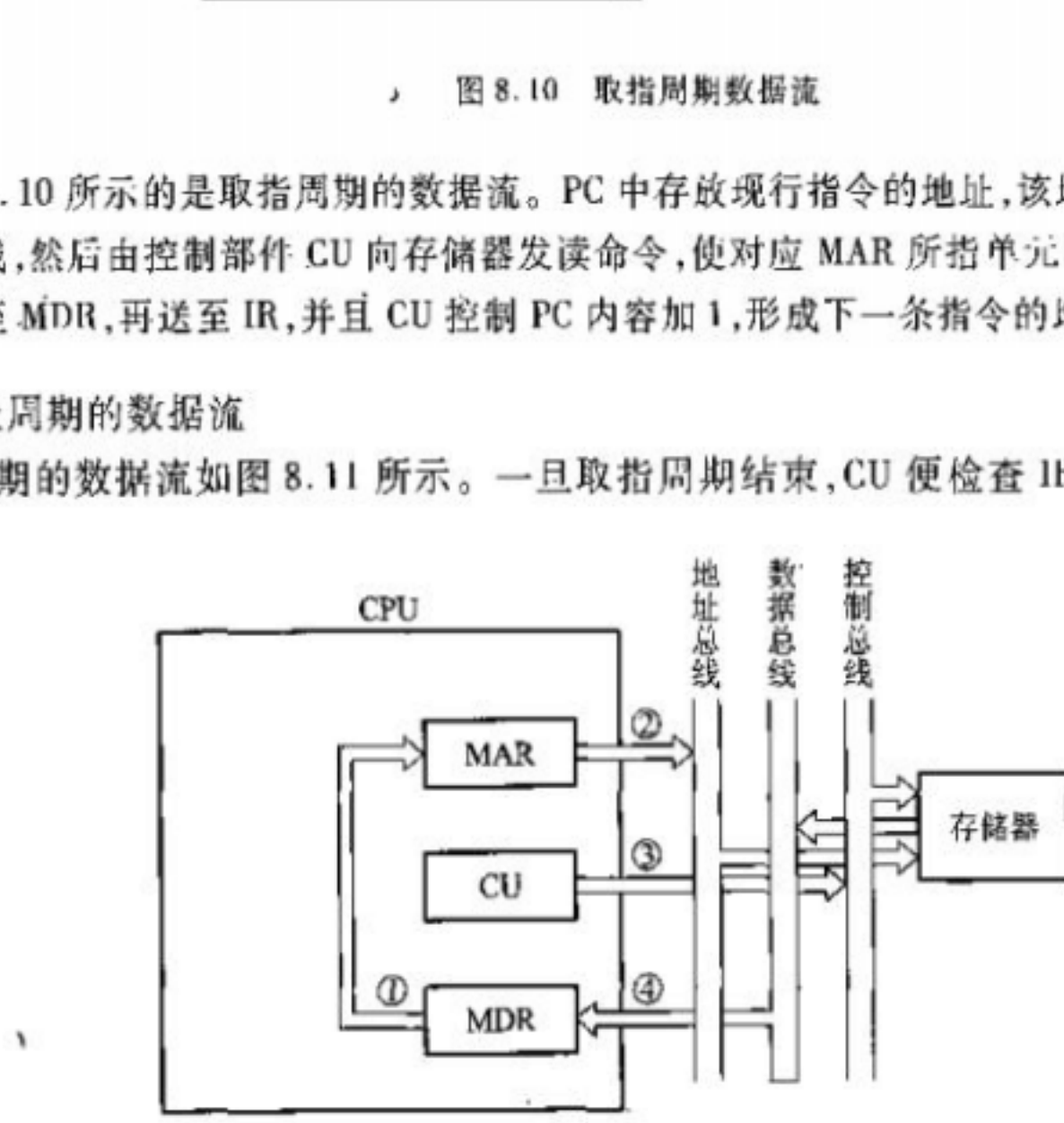


图 8.11 间址周期数据流

是否有间址操作，如果需要间址操作，则 MDR 中指示形式地址的右 N 位（记作 Ad(MDR)）将被送到 MAR，又送至地址总线，此后 CU 向存储器发读命令，以获取有效地址并存至 MDR。

4. 中断周期的数据流

CPU 进入中断周期要完成一系列操作(详见 9.1 节),其中 PC 当前的内容必须保存起来,以待执行完中断服务程序后可以准确返回到该程序的间断处,这一操作的数据流如图 8.12 所示。

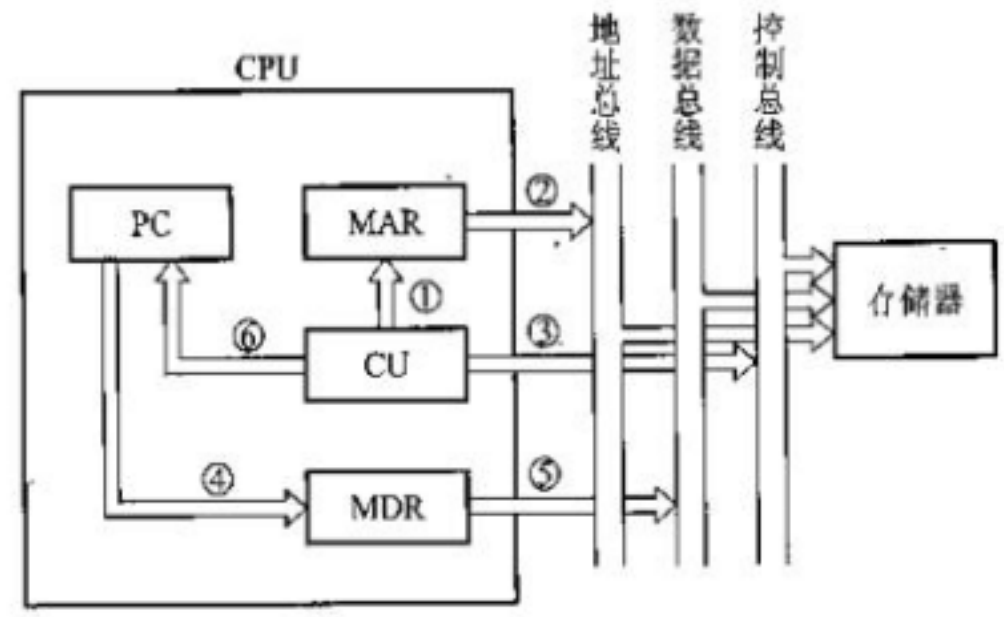


图 8.12 中断周期数据流

图中由 CU 把用于保存程序断点的存储器特殊地址(如栈指针的内容)送往 MAR,并送到地址总线上,然后由 CU 向存储器发写命令,并将 PC 的内容(程序断点)送到 MDR,最终使程序断点经数据总线存入存储器。此外,CU 还需将中断服务程序的入口地址送至 PC,为下一个指令周期的取指周期做好准备。

掌握：指令流水的相关内容（实际吞吐率　　、加速比　，影响指令流水线性能的因素　　<结构相关、数据相关、控制相关　　>）；

吞吐率：在指令级流水线中，吞吐率是指单位时间内流水线所完成指令或输出结果的数量。

最大吞吐率是指流水线在连续流动到达稳定状态后所获得吞吐率。

实际吞吐率是指流水线完成　　n 条指令的实际吞吐率。

加速比：

2. 加速比(Speedup Ratio)

流水线的加速比是指 m 段流水线的速度与等功能的非流水线的速度之比。如果流水线各段时间均为 Δt,则完成 n 条指令在 m 段流水线上共需 $T = m \cdot \Delta t + (n-1)\Delta t$ 时间。而在等效的非流水线上所需时间为 $T' = nm\Delta t$。故加速比 S_p 为

$$S_p = \frac{nm\Delta t}{m\Delta t + (n-1)\Delta t} = \frac{nm}{m+n-1} = \frac{m}{1+(m-1)/n}$$

可以看出,在 $n \gg m$ 时,S_p 接近于 m,即当流水线各段时间相等时,其最大加速比等于流水线的段数。

效率：是指流水线中各功能段的利用率。

2014-1-13

3. 效率(Efficiency)

效率是指流水线中各功能段的利用率。由于流水线有建立时间和排空时间,因此各功能段的设备不可能一直处于工作状态,总有一段空闲时间。图 8.18 是 4 段($m=4$)流水线的时空图,各段时间相等,均为 Δt。图中 $mn\Delta t$ 是流水线各段处于工作时间的时空区,而流水线中各段总的时空区是 $m(m+n-1)\Delta t$。通常用流水线各段处于工作时间的时空区与流水线中各段总的时空区之比来衡量流水线的效率。用公式表示为

$$E=\frac{mn\Delta t}{m(m+n-1)\Delta t}=\frac{n}{m+n-1}=\frac{S_p}{m}=T_p\Delta t$$

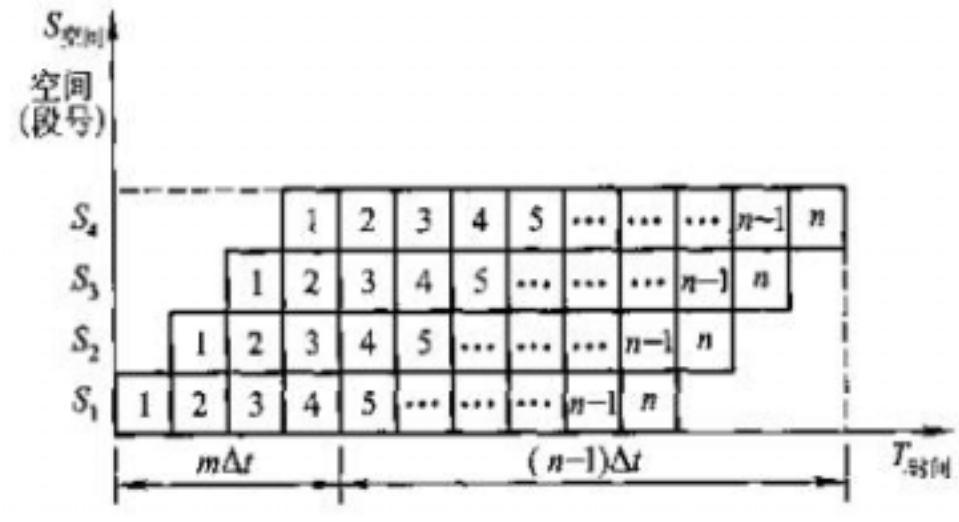


图 8.18 各段时间相等的流水线时空图

掌握： 中断系统的相关内容（中断响应、 中断向量、 开中断、 关中断、 中断屏蔽、 中断返回、响应优先级、处理优先级） 。

中断响应 ： 中断响应是当中央处理机发现已有中断请求时，中止，保存现行程序执行，并自动引出中断处理程序的过程。 中断响应是解决中断的发现和接收问题的过程， 是由中断装置完成的。中断响应是硬件对中断请求作出响应的过程，包括识别中断源，保留现场，引出中断处理程序等过程。

中断向量： 早期的微机系统中将由硬件产生的中断标识码 （中断源的识别标志， 可用来形成相应的中断服务程序的入口地址或存放中断服务程序的首地址） 称为中断向量。 硬件向量法，就是利用硬件产生向量地址，再由向量地址找到中断服务程序的入口地址。

开中断：开中断就是指系统可以在连续运行是中断，去运行中断服务函数。

关中断：就是指关闭系统中断，不允许系统打断连续的运行。

中断屏蔽：

中断返回：

响应优先级：

处理优先级：

了解：流水线多发技术（超标量技术 、超流水线技术、超长指令字技术）的特点

超标量技术： 是指在每个时钟周期内可同时并发多条独立指令， 即以并行操作方式将两条或两条以上指令编译并执行。

超流水线技术： 是将一些流水线寄存器插入到流水线段中， 好比将流水线再分段， 与超标量计算机一样，硬件不能调整指令的执行顺序呢，靠编译程序解决优化问题。

超长指令字技术：超长指令字（ VLIW）技术和超标量技术都是采用多条指令在多个处理部件中并行处理的体系结构， 在一个时钟周期内能流出多条指令。 但超标量的指令来自同一标准的指令流吗， VLIW 则是由编译程序在编译时挖掘出指令间潜在的并行性后，把多条能并行操作的指令组合成一条具有多个操作码字段的超长指令，由这条超长指令控制 VLIW 机中多个独立工作的功能部件， 由每一个操作码字段控制一个功能部件， 相当于同时执行多条指

2014-1-13

令。 VLIW 较超标量具有更高的并行处理能力，但对于优化编译器的要求跟高，对 cache 的容量要求更大。

第九章 控制单元 8 （选择、填空、判断）

掌握：微操作命令的分析（按取指周期、间指周期、执行周期和中断周期分析不同指令的微操作命令）；

控制单元具有发出各种微操作命令序列的功能。

取指周期：

现行指令地址送至存储器地址寄存器： PC->MAR。

向主存发送读命令，启动主存作读操作： 1->R.

将 MAR（通过地址总线）所指的主存单元中的内容（指令）经数据总线读至 MDR 内：M（MAR）->MDR.

将 MDR 内容送至 IR：MDR->IR.

指令的操作码送至 CU 译码： OP(IR)->CU。

形成下一条指令的地址： （PC）+1->PC。

间址周期：完成取操作数有效地址的任务

将指令的地址码部分（形式地址）送至存储器地址寄存器： Ad（IR）->MAR.

向主存发送读命令，启动主存作读操作： 1->R。

将 MAR（通过地址总线）所指的主存单元中的内容（有效地址）经数据总线读至 MDR 内：M（MAR）->MDR.

将有效地址送至指令寄存器的地址字段： MDR->Ad（IR）。

执行周期：

非访存指令：这类指令在执行周期不访问存储器。

访存指令：这类指令在执行阶段都需要访问存储器。考虑直接寻址。

转移类指令：执行期间不访问存储器

具体指令看书。

中断周期：有请求中断事件发生，则进入中断周期。

将特定地址 0 送至存储器地址寄存器： 0->MAR。/如果是断点存入栈堆，而且进栈是先修改指针，后存入数据： （SP）-1->SP,且 SP->MAR。

向主存发写命令，启动存储器做写操作： 1->W。

将 PC的内容（程序断点）送至 MDR： PC->MDR。

将 MDR 的内容 （程序断点） 通过数据总线写入到 MAR（通过地址总线） 所示的主存单元（ 0 地址单元）中： MDR->M （MAR）。

将向量地址形成部件的输出送至 PC：向量地址 ->PC。

关中断，将允许中断触发器清零： 0->EINT。

理解：控制单元的功能；多级时序系统（时钟周期、机器周期、指令周期、平均指令执行速度 MIPS 计算）；控制方式。

时钟周期： CPU最小的时间单位。

机器周期：在计算机中，为了便于管理，常把一条指令的执行过程划分为若干个阶段，每一阶段完成一项工作。例如，取指令、存储器读、存储器写等，这每一项工作称为一个基本操作。完成一个基本操作所需要的时间称为机器周期。 一般一个机器周期由若干个时钟周期组成。

指令周期： 指令周期是执行一条指令所需要的时间， 一般由若干个机器周期组成， 是从取指令、分析指令到执行完所需的全部时间。

MIPS 计 算 ：

2014-1-13

例 9.3 设某计算机的 CPU 主频为 8 MHz，每个机器周期平均含 2 个时钟周期，每条指令的指令周期平均有 2.5 个机器周期，试问该机的平均指令执行速度为多少 MIPS？若 CPU 主频不变，但每个机器周期平均含 4 个时钟周期，每条指令的指令周期平均有 5 个机器周期，则该机的平均指令执行速度又是多少 MIPS？由此可得出什么结论？

解：由于主频为 8 MHz，所以时钟周期为 1/8 = 0.125 μs，机器周期为 0.125 × 2 = 0.25 μs，指令周期为 0.25 × 2.5 = 0.625 μs。

① 平均指令执行速度为 1/0.625 = 1.6 MIPS。

② 若 CPU 主频不变，机器周期含 4 个时钟周期，每条指令平均含 5 个机器周期，则指令周期为 0.125 × 4 × 5 = 2.5 μs，故平均指令执行速度为 1/2.5 = 0.4 MIPS。

③ 可见机器的运行速度并不完全取决于主频。

此外，机器的运行速度还和其他很多因素有关，如主存的运行速度、机器是否配有 Cache、总线的数据传输率、硬盘的运行速度以及机器是否采用流水技术等。机器速度还可以用 MIPS（执行百万条指令数每秒）和 CPI（执行一条指令所需的时钟周期数）来衡量。

控制方式：同步控制、异步控制、联合控制、人工控制。

同步控制方式： 是指， 任何一条指令或指令中任何一个微操作的执行都是事先确定的， 并且都是受统一基准时标的时序信号所控制的方式

采用定长的机器周期；

采用不定长的机器周期；

采用中央控制和局部控制相结合的方式。

人工控制方式：

Reset 键

连续或单条执行转换开关

符合停机开关

第十章 控制单元的设计 2 （判断）

理解：组合逻辑设计方法；

掌握：微程序设计思想：安排微操作时序的原则、微程序控制单元框图及工作原理、微指令的编码方式、微指令地址的形成方式、微指令格式（水平型微指令、垂直型微指令各自的特点）。

掌握：机器指令与微指令的关系、微指令序列地址形成方式中的断定方式。

掌握：微指令存储于控制存储器；控制存储器位于控制单元内，控制单元位于 CPU内。

2014-1-13