**第1章 计算机系统概述**

1.3 计算机的硬件组成

手机屏幕的截图

描述已自动生成【硬件结构】冯诺依曼计算机由运算器、控制器、存储器、输入设备和输出设备组成，运算器、控制器合称为中央处理器。

【存储程序方式】程序和数据预先存放在存储器中，机器工作时，自动、逐条地从存储器中取出指令并执行。

【存储器结构】存储器为由定长单元组成的一维线性空间，按地址进行访问。

图示, 示意图

描述已自动生成【主机】CPU+主存储器。

【外部设备】输入设备+输出设备+辅助存储器。

【CPU】运算器+控制器；运算器由ALU、FPU、寄存器等组成；控制器由PC、IR、ID、CU等组成。

【ALU】算术逻辑单元，处理整型数据。

【FPU】浮点运算部件，处理浮点数据。

【PC】程序计数器，存放作为循环变量的指令地址。

【IR】指令寄存器，存放当前指令的内容。

【ID】指令译码器，分析（识别）IR中当前指令的操作类型及操作数信息。

【CU】控制单元，产生相应的部件控制信号，来控制相关部件实现指令的功能。

图表, 箱线图

描述已自动生成【存储元】存储一位二进制信息。

【存储单元】由多个存储元构成。

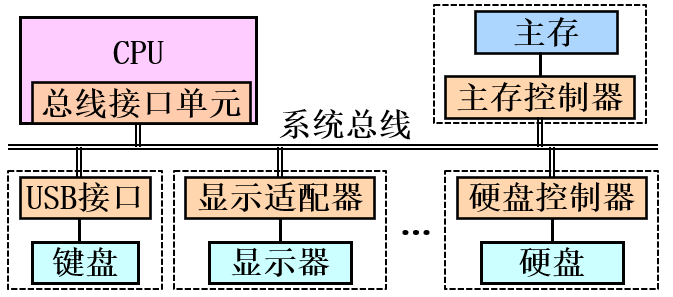
【存储矩阵】

【存储单元长度】存储单元所能存放的二进制位数。

【存储单元地址】

【存储字】存储单元所存放的信息。

【存储字长】存储单元所能存放的二进制位数。

【总线】设备（或部件）间进行信息传输的一组公共信号线，总线上的设备通过地址（设备编号）来进行识别。互连特点：分时传输、可扩展性好。

【主设备】（发起总线操作的设备）通过控制线发出操作命令。

【从设备】（响应总线操作的设备）通过状态线反馈传输状态。

【I/O接口】由于不同外设的速度、设备接口都有所不同，为了将外设连接到总线上，需要在外设与总线间增设连接电路，这个连接电路被成为I/O接口。I/O接口的主要功能是实现数据缓冲、格式转换、通信控制，以协调总线与外设在传输速度、数据格式、传输协议（操作时序）方面的差异。I/O接口又常被成为（设备）适配器或（设备）控制器。

【系统总线】连接计算机各大部件的总线；通常由ABus、DBus、CBus三种信号线组成。

【ABus】地址总线，用于选择传输的目标设备。

【DBus】数据总线，用于承载传输的内容。

【CBus】控制总线，用于控制传输的过程。

【实际机器】所编写程序可以被硬件识别和直接执行的机器成为实际机器。包括机器语言级机器、微程序级机器。

【虚拟机器】其余的称为虚拟机器。包括高级语言级机器、汇编语言级机器、操作系统级机器。

【翻译】先将程序转换为低级机器上的等效程序，再在低级机器上仪器运行。

【解释】对程序中的每一条语句，都转去执行低级机器上的一段等效程序，循环往复地解释并执行每一条语句，直到程序执行结束。

【语句】高级语言程序的基本单位，使用符号表示操作、操作数或指令地址，效率更高、通用性更好。语句表示的命令硬件无法识别。

【指令】汇编语言的命令。符号表示的命令与指令一一对应。

【计算机系统结构】机器语言程序员或编译程序编写者所看到的计算机系统的属性，包括概念性结构和功能特性两个方面。其内容主要有指令集、数据表示、寻址方式、存储系统、I/O结构等，这些属性都是由硬件或固件完成的功能。计算机系统结构主要研究计算机系统中软硬件的交界面。由于软硬件交界面的核心是指令集，因此，计算机系统结构又常称为指令集结构（ISA）。

【计算机组成】计算机硬件设计人员所看到的计算机系统的属性，它包含了许多对程序员来说是透明的硬件属性，包括专用部件及数据通路宽度设计、功能部件并行度、控制机构组成方式等。计算机组成主要研究如何合理地逻辑实现计算机系统结构分配给硬件的功能。

【计算机实现】计算机制造人员所看到的计算机系统的属性。它包括部件的物理结构、器件的集成度、模块的划分与连接、信号传输、电源、冷却及组装技术等。计算机实现主要研究器件技术和微组装技术，即如何高效地物理实现计算机组成所要求的功能。

【逻辑地址】由于程序需要存放在主存中才能执行，而主存是按地址访问的一维线性空间，因此程序的空间需要按主存单元长度进行编址，由于生成程序时不知道程序将来在主存中的地址，通常从0开始编址，称为逻辑地址。

【物理地址】程序必须放在主存中才能执行，此时的指令地址需要用主存地址来表示。因此指令地址、数据地址有双重含义，在程序中的地址称为逻辑地址，在主存中的地址称为物理地址。程序执行时所指的地址，通常都是物理地址。

【机器字长】机器字长指CPU一次能处理数据的二进制位数，又称为CPU字长。CPU中的运算有定点运算、浮点运算两大类，机器字长指的是一次定点整数运算的位数。

【CPU主频】指令执行过程由若干个步骤及相应的操作组成，每个操作何时开始、延迟时长都要有相应的时序信号进行同步。CPU使用时钟脉冲信号来进行操作定时，CPU中用于最基本操作定时的时钟脉冲信号，称为CPU的主时钟信号，主时钟信号的宽度称为时钟周期。CPU主时钟信号的频率称为CPU主频或主频，又称为时钟频率。

【主存编址单位】一个主存单元所能存放信息的位数，又称主存单元长度。因此，主存容量=主存单元长度×主存单元个数。

【主存地址空间】主存最大容量时的主存单元地址构成的空间。

【响应时间】又称执行时间，指一个任务从提交到完成所花的全部时间，包括CPU运算、主存访问、I/O操作（如磁盘访问）、操作系统开销等所有的时间。响应时间可以用 来表示，其中 称为CPU时间，指CPU用在程序执行上的时间，包括数据运算、访问主存等时间； 称为等待时间或其他时间，指在完成任务过程中除CPU执行程序外的时间，包括等待I/O操作完成、操作系统开销等时间。

【CPI】假设CPU为完成某任务共执行了 条指令，CPU的时钟周期为 ，则有

式中，CPI指一条指令执行所需的时钟周期数。对特定的指令而言，其CPI为执行该指令所需的时钟周期数，通常是一个确定值。对一个程序或一台机器而言，其CPI为程序或机器中的所有指令执行所需的平均时钟周期数。

【吞吐率】又称吞吐量，指单位时间内计算机完成的总工作量。吞吐率可以用 来表示，其中总时间指第一个任务提交到最后一个任务完成的响应时间。

【MIPS】对于给定的n个程序，MIPS的定义为：

【MFLOPS】对于给定的n个程序，MFLOPS的定义为：

**第2章 数据的表示与运算**

【数制】进位计数制又称进制，是用一组固定的符号和统一的规则来表示数值的方法。有数码、基数和位权3个参数，数码为计数的符号，基数为进位的单位，位权为不同数码位的权值。

【机器数】计算机内部用编码表示的数称为机器数。

【真值】对应地，把数学上的数称为真值。

【原码】原码的编码思想是：机器数的最高位为符号位（0表示正、1表示负），其余各位为真值的绝对值。

【补码】补码的编码思想是：正数的补码为其本身，负数的补码为其正补数。

【反码】反码的编码思想与补码一致，只是模比补码少1。反码是一种过渡性编码，常用作补码变换的中间表示形式。

【移码】移码的编码思想是：机器数等于真值加上一个常数。同一真值的移码和补码仅最高位不同、其余位完全相同。

【模】模运算系统中，模是指计量器的容量（技术范围）。

【同余】若 、、 满足如下关系：，则记为 ，则称 和 为模 的同余。

【补数】数学上，若 、、 满足关系：，则称 、 互为模 的补数。

【BCD码】用二进制形式表示十进制数的编码方式，称为二-十进制编码（Binary Coded Decimal, BCD）方式。

【交换码】交换码主要用于字符的索引、传送，编码为字符在字符集中的序号，如字符‘A’的ASCII码为65。交换码的长度只与字符集大小有关。

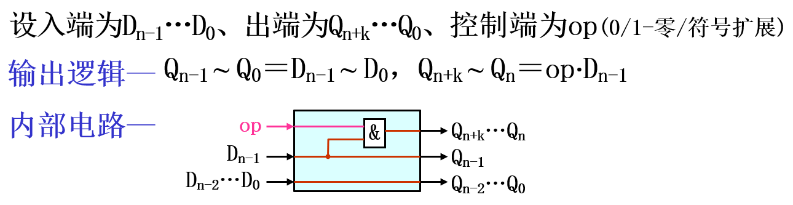
【内码】内码主要用于字符的存储和处理，编码由交换码及扩展码组成，扩展码可用于支持多个字符集，使编码长度为主存单元长度的倍数。内码的长度与字符集大小、支持的字符集个数、主存单元长度有关。

【整数的表示.定点表示法】定点表示法约定数据中的小数点位置固定不变，这种数据格式称为定点格式。采用定点格式表示的数称为定点数。通常，小数点位置只约定在最低位之后图示

描述已自动生成或最高数值位之前。针对不同的数据类型，定点数的表示格式共有3种。

图表

描述已自动生成【整数的表示.整数的表示】整数的表示方法中，进制通常为二进制，格式为定点格式，编码方式种，无符号整数为无符号编码，有符号整数为补码。以IA32为例，整数共表示为6种数据类型，分别为8/16/32位的有符号整数和8/16/32位的无符号整数。

【整数的表示.整数的类型转换】位数相同的类型转换，无须进行任何运算，数据运算方法可以改变位值的解释。位数减小的类型转换，需要通过阶段运算实现，即保留数据的低位部分。位数增加的类型转换，需要通过位扩展运算实现，为扩展运算有零扩展（Zero Extension）、符号扩展（Sign Extension）两种类型。

手机屏幕的截图

中度可信度描述已自动生成【实数的表示.浮点表示法】任意一个实数都可描述成 形式，其中 称为尾数（mantissa）， 称为指数（exponent）或阶， 称为基数，基是尾数的进制。通常，尾数为纯小数，基为二进制，即 。浮点数的阶大于最大阶码时称为上溢，上溢是真正的溢出，应进行溢出处理；浮点数的阶小于最小阶码时称为下溢，下溢可以认为是机器零，不是溢出。

表格

描述已自动生成【实数的表示.浮点数的规格化】浮点数中，有效数字的位数越多，浮点数的精度就越高，有效位数的最大值由浮点数的尾数决定，为了提高浮点数的精度，减少浮点运算过程中的精度损失，尾数中有效数字的尾数应最大化，即尾数的最高数值位为有效数字。通常，将尾数最高数值位为非0的浮点数称为规格化数。理论上，基数 时，规格化数的尾数 应当满足条件：。左规是指将尾数左移1位，同时将阶码减1；右规是指将尾数右移一位，同时将阶码加1。判断浮点数是否为规格化数的方法：不管尾数 用何种方式（原码、补码）表示，如果 乘2会发生上溢出，那么该浮点数是规格化数。

表格

低可信度描述已自动生成【实数的表示.IEEE754】尾数 为原码定点小数，改变了点位置；阶码 为移码定点整数，改变了偏移值。

【定点数的运算.加减运算】有符号加减运算溢出的特征是：同号相加或异号相减时，结果符号与被加（减）数符号不同。有符号运算结果是否溢出用OF（Overflow Flag）表示。无符号加减运算的溢出条件是：加法时最高位有进位，减法时最高位有借位。无符号运算的溢出称为有进位/借位，用符号CF（Carry Flag）表示。

【定点数的运算.移位运算】逻辑移位的数据为无符号数，无符号数乘以2时末位补0、除以2时首位补0。因此，逻辑移位运算规则为：机器数整体移位，移出的数据丢弃，出现的空位补0。算数移位的数据为有符号数。补码移位运算规则为：机器数整体移位，移出的数据丢弃，左移的空位补0、右移的空位补符号位。

【定点数的运算.乘法运算】机器乘法的实现思路是：（1）乘法运算用循环的加法运算及右移运算实现。（2）每次相加时，根据乘数当前位的值，确定是加被乘数还是加0。（3）每次右移时，加法的进位、部分积高位、部分积低位一起右移。 位无符号乘法的运算规则为：循环进行 次判断-加法-移位操作，每次循环中，判断操作确定加数的值，加法操作求出部分积，移位操作将加法进位及部分积一起右移。原码乘法的符号位与数值位需分开计算，运算分为计算乘积符号、计算乘积数值两个步骤，计算乘积数值的方法就是无符号乘法运算。 位补码的运算规则为：循环进行 次判断-加法-移位操作，每次循环的操作如表所示：

|  |  |  |
| --- | --- | --- |
| **判断位** | **加法** | **移位** |
| 00或01 |  | 算术右移一位 |
| 01 |  |
| 10 |  |

算术右移时不需带进位一起移位，因为加法运算不会产生溢出。

【浮点数的运算.加减运算】（1）对阶。对阶的目的是使两个浮点数的小数点对齐，即两个浮点数的阶相同；对阶的规则是小阶对大阶，用大阶作为运算结果的阶，小阶浮点数的尾数右移，右移位数等于两个数的阶差。尾数右移为算术移位，移出的数值位进入附加位。（2）尾数加减。对阶后两个数的阶相同，尾数可直接进行加减运算，运算时附加位一同参与。（3）规格化。

【运算器的组成】运算器是CPU中所有运算部件的统称，它负责所有数据的运算以及运算结果的临时存放。运算部件通常由基本运算功能电路、扩展运算功能电路（如移位器）、暂存数据的寄存器、选择数据的多路选择器，以及内部控制逻辑等构成。

【运算部件的互连】（1）总线互连。总线互连指各个部件的输入端及输出端通过一条或几条总线连接起来。由于每条总线上都连接有多个输出端，因此数据传送需要分时继续进行。总线结构的运算器中，每个部件的输出端需要通过三态门连接到总线上，以避免输出信号冲突。当部件的不同输入端，或输入端及输出端连接在同一条总线上时，入端之前或出端之后必须增设锁存器，以避免端口间的数据干扰。（2）点点互连。点点互连指不同部件的输入端与输出端之间都根据需要使用信号线连接，因此，不同部件间的数据传送可以同时进行。当部件的输入端数据来自多个部件的输出端时，需要设置多路选择器来实现直接连接。总线互连方式成本低、性能差，点点互连方式正好相反。

**第3章 存储系统**

【存取时间】又称访问时间，指存储器从启动一次存储器操作（读或写）到完成该操作所需的时间。存取时间有读出时间、写入时间两种。读出时间是存储器从收到有效地址开始，到数据送到引脚的全部时间；写入时间是存储器从收到有效地址开始，到数据写到所选地址的全部时间。

【存取周期】指存储器连续进行两次存储器操作的最短间隔时间。

【传输速度】传输速度通常用存储器带宽表示。存储器带宽指存储器被连续访问时，可以提供数据的速率，即单位时间内存储器最多可以传送的信息量，单位通常是Mbps、MB/s等。

【程序访问的局部性原理】程序访问的局部性原理是指，程序运行时，指令和数据访问所呈现出的相对簇聚的现象，局部性有时间局部性、空间局部性两个方面。时间局部性指最近访问过的信息，在不久将会被再次访问。空间局部性指与最近访问信息相邻的信息，在不久将会被访问。

手机屏幕截图

描述已自动生成图示

描述已自动生成【SRAM芯片的组成】包含存储矩阵、地址译码器、I/O门、读写电路、控制电路等。

【SRAM芯片的引脚组成】有地址引脚、数据引脚、控制引脚（片选及读写引脚）组成。地址引脚的个数与芯片容量、存储单元长度都有关系。假设SRAM芯片的存储单元长度为 位，容量位 位，地址引脚为 个，则有 ，。

图示

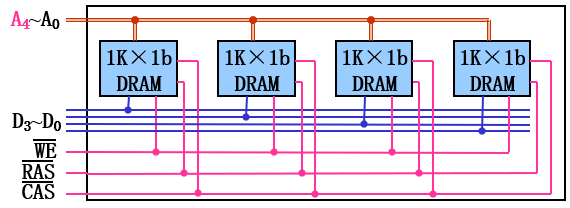
描述已自动生成【DRAM芯片的引脚组织、组成】DRAM芯片采用地址引脚复用技术，即地址由行地址、列地址组成，地址分两次送入DRAM，地址引脚个数只有地址位数的一半。为了实现地址分两次传输，DRAM设置有行选通信号RAS#、列地址选通信号CAS#，来指明地址引脚上的当前地址类型。进一步地，使RAS#在整个读写周期中全部有效，可以代替片选信号CS#。DRAM芯片主要由存储矩阵、地址译码器、读写电路组成，与SRAM相比增加了地址锁存器、时序控制电路、再生电路。

【DRAM芯片的刷新方式】DRAM采用行刷新方式后，刷新操作实际上是在一个刷新周期内，图片包含 游戏机, 天线, 物体, 钟表

描述已自动生成对所有行逐行进行刷新操作。（1）集中式刷新。（2）分散式刷新。（3）异步式刷新。异步式刷新指各行的刷新操作均匀分散在整个刷新周期中进行。异步方式中，每过一定时间进行一次刷新操作，刷新周期结束时刚好刷新一遍。

【主存的逻辑设计.位扩展法】

图示

描述已自动生成

【主存的逻辑设计.字扩展法】

图示, 示意图

描述已自动生成

手机屏幕的截图

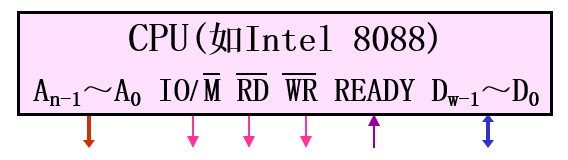
描述已自动生成

【主存的逻辑设计.字位扩展法】

图片包含 表格

描述已自动生成图示

描述已自动生成

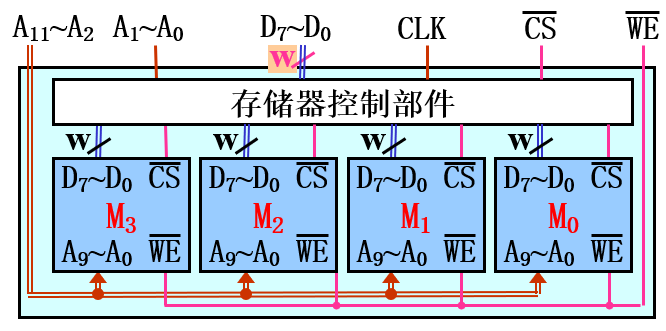
【CPU的存储器接口】包含地址引脚、数据引脚、控制/状态引脚。其中，IO/M#用于表示操作的目标部件类型（高电平为I/O设备、低电平为主存），RD#及WR#表示操作类型（读/写/空闲），READY用于表示操作的完成状态（高电平为已完成、低电平为未完成）。

【主存与CPU的连接】（1）数据线。主存的数据线需要与CPU的数据线一一连接。（2）地址线。主存的地址线与CPU地址线的低位信号线连接，CPU地址线的高位信号线用来选择主存，即高位地址线的值为0时是对主存进行操作。（3）控制线。主存的WE#线与CPU的WR#线连接。主存的CS#线应与CPU的控制线、地址线连接。

【SDRAM】（Synchronous DRAM，同步DRAM）SDRAM的所有操作都在时钟信号控制下进行，在确定的几个时钟周期后给出相应，CPU及DRAMC在其期间无需等待。SDRAM还支持突发（burst）传输模式，在第一个数据被访问后，可以连续传输多个数据。通常将连续传输的数据个数称为突发长度（Burst Length, BL），常规传输看作 的突发传输。突发传输模式对一次需要访问多个连续存储单元的访存操作很有效，是SDRAM相对异步DRAM的最大优点。

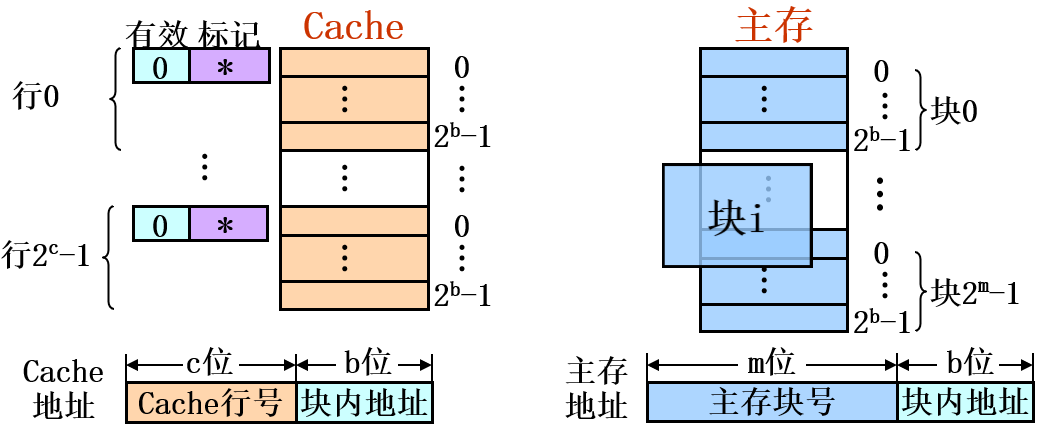
【DDR SDRAM】（Double Data Rate SDRAM，双倍数据速率SDRAM）DDR SDRAM能够在时钟脉冲的上升沿、下降沿分别进行数据传输。DDR SDRAM存储阵列中单元宽度为I/O宽度的两倍，在I/O缓冲器中增设预取缓冲区，读操作时将缓冲区中数据拆分后分时送出，写操作时数据填满缓冲区后一起写入。可见，DDR SRAM存取速度是SDRAM的2倍。

图示

描述已自动生成【多体交叉存储器】多体存储器指存储器由多个容量相同的存储体组成。交叉访问方式指轮流访问各个存储体，多个数据分时I/O，多体交叉存储器结构如图所示。交叉访问通常采用轮流启动方式（流水访问方式），即每隔 个存取周期（）启动一个存储体，经过 时间后，每隔 即可读出或写入一个数据。

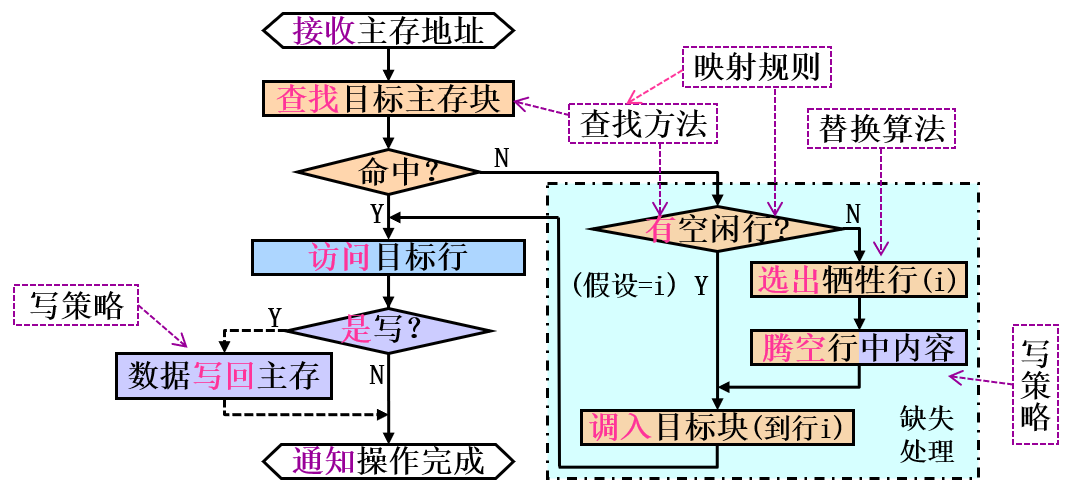
【Cache的性能指标】衡量Cache性能的指标主要有命中率及平均访问时间。若访存的信息已经在Cache中，则称为命中（Hit），否则称为确实（Miss）。访存操作在Cache中命中的概率称为命中率，等于命中次数与访存总次数的比值，常用 表示。Cache命中时，Cache完成访存操作所用的全部时间 称为命中时间（Hit Time）；缺失时，Cache需要先与主存交换信息，再完成访存操作，所用的时间由主存访问时间 和命中时间 组成，这个主存访问时间 称为缺失损失或缺失开销。平均访问时间 指Cache-主存层次完成访存操作平均所用的时间，则

【Cache的存储空间管理.Cache-主存的信息交换单位】Cache-主存之间的信息交换单位为块（Block），块由多个相邻的存储字（主存字）组成，大小固定。块大小是由计算机结构确定的，通常为8个机器字长。

【Cache的存储空间管理.Cache-主存的信息交换管理】主存空间、Cache的数据空间划分为若干个大小为块大小的区域，主存中的区域称为主存块，Cache中的区域称为缓存块。Cache中的每一个缓存块都要对应有一些管理信息，通常将存放缓存块和相应管理信息的存储空间称为行（Line）或槽（Slot）。可见，主存地址由主存块号、块内地址组成，Cache地址由Cache行号、块内地址组成。为了表示Cache行中的数据是否有效，每个行需要设置一个有效位（Valid bit）字段，常用V表示。有效位V=1表示该行的数据有效，V=0表示该行空闲。为了表示Cache行中的数据来自哪个主存块，每个行需要设置一个标记（Tag）字段。这个标记通常是主存块号的一部分，这样Cache行与主存块就有了对应关系。

手机屏幕的截图

中度可信度描述已自动生成【Cache的存储空间管理.Cache的内部结构】Cache的存储空间可以组织成行的数组，每个Cache行中包含有效位、标记等管理信息，以及缓存块信息。其中 ，。

【Cache的工作原理】Cache完成访问的过程分三步：（1）将主存地址变换成Cache地址，即查找请求字所在的Cache行，缺失时需要先指定一个空闲行，再调入目标主存块；若没有空闲行，还需要先找出一个牺牲行并腾空其中内容。（2）读/写Cache存储阵列，完成访问要求。（3）在适当时候将所写数据写回主存。

【Cache的地址映射】

（1）直接映射

手机屏幕截图

描述已自动生成

（2）全相联映射

图片包含 图示

描述已自动生成

（3）组相联映射

图示

描述已自动生成

（4）小结

表格

描述已自动生成

【Cache的替换算法】（1）随机算法（Random， RAND）。随机算法的基本思想是在候选行中随机选择一个牺牲块。算法实现时，整个Cache只需要一个随机数发生器。（2）先进先出算法（First In First Out，FIFO）。先进先出算法的基本思想是选择最早调入的块作为牺牲块。算法实现时，需要在候选行的每一行设置一个计数器，用各个计数器的值来表示候选行中各个块的调入顺序，替换时选择计数值最大的行作为牺牲行，每当有块被调入时更新所有计数器的值。（3）最近最少使用算法（Least Recently Used，LRU）。最近最少使用算法的基本思想是，选择近期最少使用的块作为牺牲块。这种算法的选择较好地反映了程序访问局部性。算法实现时，硬件配置与FIFO算法完全相同，但处理方法不同，计数器的更新方法为：计数值比访问行小的计数器加1，被访问行的计数器清0。Cache进行LRU算法组织时，通常在各个Cache行的管理信息中设置“LRU位”，放在Cache行的状态字段图示

描述已自动生成中，“LRU位”的位数就是计数器的位数。

图示

描述已自动生成【Cache的写策略】（1）全写法（Write Through）。全写法的基本思路是，写命中时，将数据写入Cache，同时写入主存；写缺失时，通常直接将数据写入主存，而不将目标主存块调入Cache。由于Cache与主存保持了一致性，全写法在替换时，只需将牺牲行抛弃即可（有效位清零），而无需将其写入主存。（2）写回法（Write Back）。写回法的基本思路是，写命中时，只将数据写入Cache，不写入主存；写缺失时，通常先将目标主存块调入Cache，再将数据写入Cache；仅当Cache行中主存块被替换、且主存块被修改过时，才将该行暂存的主存块写回主存。由于仅当牺牲行中内容被改写过时，才将该主存块写回主存，因此，每个Cache行的管理信息中，需要设置一个修改为（Dirty bit），又称脏位，表示该行是否被改写过。因此，Cache在处理写操作时，需要将脏位置1；再进行替换处理时，需要判断脏位，确定是否将该缓存块写入主存。

【虚拟存储器】

**第4章 指令系统**

【指令的操作】（1）数据传送及输入/输出。（2）算术逻辑运算与移位运算。为了支持关系运算的实现，算术运算应该产生结果的状态标记（CF/ZF/OF/SF），并保存到状态寄存器中，供其他指令使用。（3）转移控制。

【指令格式】由指令功能可知，机器指令包含的信息由操作功能、操作数类型、各个源操作数地址、目的操作数地址、下条指令地址。因此，指令格式通常由操作码和地址码两个部分组成。其中，操作码用来表示指令的操作类型及格式信息，操作类型包括操作功能及操作数类型，格式信息包括地址码个数、目的操作数地址位置；地址码用来表示各个操作数地址及下条指令地址。

表格

描述已自动生成【操作数的存放方式】（1）数据在寄存器中的存放方式。数据有多种长度，不同长度的数据存放在寄存器中时，数据地址都用 位寄存器地址表示，以便统一处理。当数据长度等于寄存器长度的长数据存放时，数据占用一个寄存器的全部位数。数据长度小于寄存器长度的短数据存放时，有寄存器低端、部分寄存器两种存放方案。（2）数据在存储器中的存放方式。由于存储单元长度是最短数据的长度，因此，数据在存储器中存放时，其内容可能存放在多个连续的存储单元中，数据地址用第一个存储单元的地址 来表示。数据是一串0/1序列，通常用最低有效位（Least Significant Bit，LSB）及最高有效位（Most Significant Bit，MSB）来表示数据的最低位及最高位。若数据以字节为排列单位，则LSB表示最低有手机屏幕截图

描述已自动生成效字节，MSB表示最高有效字节。端序（Endian）指数据中各个字节在存储器中的字节顺序，有大端（big endian）和小端（small endian）两种方式。对齐（Alignment）指数据在存储器中的位置限制，有不对齐和对齐两种方式。不对齐方式指数据可以从任意位置开始存放，对齐方式指数据必须从特定位置开始存放，默认采用边界对齐图示, 表格

描述已自动生成方式。边界对齐指数据地址为数据长度的倍数，即长度为 字节的数据，数据地址的最低 位必须为0。（3）数据在指令寄存器中的存放方式。指令寄存器用来存放当前指令，常数操作数通常直接放在指令中，执行时直接获取；操作数为变量时，用地址码表示。常数在指令中存放时，端序通常与在存储器中存放相同，并采用不对齐方式存放，以缩短指令字长。

【指令寻址方式】指令寻址方式指形成下条指令地址的方法，有顺序寻址和跳跃寻址两大类。顺序寻址中，下条指令地址由PC加一形成，其中“1”为当前指令所占存储单元个数。跳跃寻址中，下条指令地址由当前指令的地址码形成，即下条指令地址由寻址函数f产生。不同寻址方式的f不同，寻址方式通常有直接寻址、相对寻址等。识别指令寻址方式时，可以通过操作码来区分是顺序寻址还是跳跃寻址，跳跃寻址中的具体寻址方式，再通过地址码中的信息进行区分。寻址方式的识别在译码阶段完成，寻址方式的实现在执行阶段完成。

【数据寻址方式】（1）立即寻址。

形状, 矩形

描述已自动生成

（2）寄存器寻址。

图片包含 QR 代码

描述已自动生成

（3）直接寻址。

图片包含 形状

描述已自动生成

（4）间接寻址。

图片包含 图示

描述已自动生成

（5）寄存器间接寻址。

图示

中度可信度描述已自动生成

（6）变址寻址。

手机屏幕的截图

中度可信度描述已自动生成

（7）基址寻址。基址寻址方式中，基址寄存器中为基址地址，A为操作数地址相对于基址地址的偏移量，而变址寻址方式正好相反。

图示

描述已自动生成

（8）相对寻址。

图示

描述已自动生成

【指令系统的发展】CISC侧重增强指令功能，减少程序中指令数来提高性能。因而，CISC指令系统较复杂，指令种类多、寻址方式多、指令格式多（变长结构）。RISC侧重简化指令功能，通过减小指令执行时间来提高性能。因而，RISC指令系统很简单，指令种类少、寻址方式少、指令格式规整（定长结构）。

**第5章 中央处理器**

【CPU的功能】(1)指令控制。控制指令的执行顺序。（2）操作控制。产生指令执行所需的操作控制信号。（3）时间控制。控制操作控制信号的时序。（4）数据加工。实现指令约定的数据运算。（5）外部访问。实现对存储器、外设的访问。（6）异常及中断处理。实现异常及中断的检测及处理。

手机屏幕的截图

中度可信度描述已自动生成【CPU的组成.基本组成】（1）为了实现指令控制，控制器中需要设置用于存放指令地址的程序计数器PC、用于存放指令内容的指令寄存器IR，以及用于分析指令的指令译码器ID。（2）为了实现操作控制和时间控制，控制器中需要设置能够产生各种时序信号的电路，以及能够产生操作控制信号的电路，操作控制信号是基于时序信号产生的。（3）为了实现数据加工，需要设置用于完成算术和逻辑运算的ALU、完成浮点运算的FPU、用于存放结果的寄存器组、用于存放结果状态的状态寄存器等部件。（4）为了实现外部访问，需要设置能够进行总线传输控制的总线逻辑电路、与ALU等交互的缓冲寄存器，以及实现地址变换的存储器管理单元MMU。（5）为了实现异常及中断处理，控制器中需要设置用于异常及中断处理的中断机构部件。因此，CPU由运算器、总线接口单元BIU、存储器管理单元MMU、指令部件、控制单元CU以及中断机构6个部分组成。

【CPU的组成.寄存器组织】CPU中的寄存器可以分为两大类：一类是用户可见寄存器，用户编程时可以使用这类寄存器来实现定制功能；另一类是系统专用寄存器，用于控制CPU的操作及工作模式，只有一部分可以被操作系统程序调用。

【CPU的组成.寄存器组织.用户可见寄存器】（1）数据寄存器。专门用于存放指令的操作数，其位数等于机器字长。通常，将既存放源操作数又存放目的操作数的寄存器称为累加寄存器（AC）。（2）地址寄存器。专门用于存放操作数地址或指令地址，如用于寄存器间接寻址、基址寻址、变址寻址的寄存器。其位数等于程序逻辑地址长度，或段内地址长度。（3）通用寄存器（General Purpose Register，GPR）。既可用于存放数据，又可用于存放地址。（4）状态寄存器（Program Status Register，PSR）。用于存放程序运行的状态，又称为标志寄存器。程序运行状态包括运算结果标志及执行方式标志，其内容常称为程序状态字（Program Status Word，PSW）。运算结果标志由ALU产生，如ZF/CF/OF/SF等，常用作分支等指令的测试条件，故又称为条件吗。执行方式标志用于控制程序如何执行，如跟踪标志TF、中断允许标志IF等，TF=1时程序处于单步执行方式。

图示

描述已自动生成【CPU的组成.寄存器组织.专用寄存器】（1）程序寄存器（PC）。专门用于存放指令地址，指令地址用作程序执行过程的循环变量。（2）指令寄存器（IR）。专门用于存放当前指令内容。（IR）在取指令结果写入前，存放的是上调指令的内容。（3）存储器地址寄存器（Memory Address Register，MAR）。用于存放CPU外部访问的部件地址，如存储单元地址或I/O设备地址。（4）存储器数据寄存器（Memory Data Register，MDR）。用于存放CPU从外部读出的数据或欲向外部写入的数据。（5）控制寄存器。用于控制CPU的工作模式及操作方式，如许村的管理方式、Cache的写策略等。

【CPU的工作流程】CPU的工作流程由循环的指令周期、中断周期组成。CPU的六大功能中，数据加工、外部访问在指令周期中完成，异常及中断处理通过中断响应、多个指令周期实现，指令周期、中断周期由多种操作序列组成。而指令控制、操作控制、时间控制由控制器实现。

【CPU的工作原理】控制单元循环地产生工作流程所需的操作控制信号，数据通路实现工作流程所需的操作。

【指令的执行过程.指令执行步骤】指令执行过程有取指令、分析指令、执行指令3个步骤，由于操作数可以存放在寄存器、存储器等部件中，可能支持多种寻址方式，因此，指令执行过程可分为取指令、指令译码、取操作数、数据操作、保存结果、计算指令地址6个步骤。手机屏幕的截图

中度可信度描述已自动生成（1）取指令。用(PC)作为地址读存储器，访存结果（当前指令内容）写入IR。对于变长指令字格式，可能需要多次访存才能取到完整的指令字。（2）指令操作译码。对当前指令的操作码及寻址方式进行译码，输出信号供CU用来产生各种操作控制信号，地址码的地址参数直接从指令字中提取，供寻址方式的地址计算部件使用。（3）取操作数。进行源操作数地址计算，并用计算结果作为部件地址读取操作数。若为寄存器操作数，直接从寄存器组中读取；若为立即数，直接从IR中读取；若为存储器操作数，则需要一次或多次访存读取。（4）数据操作。在ALU等运算部件中进行数据运算。（5）保存结果。根据目的操作数寻址方式的要求，进行目的操作数地址计算，并用计算结果作为部件地址写入操作结果。写入方法与取操作数相同，指示操作类型由读改为写。（6）指令地址计算。根据指令寻址方式的要求，进行下条指令地址的计算，计算结果写入到PC中。若当前指令为顺序性指令，则PC=(PC)+“1”；若当前指令为转移型指令，且指令发生转移时，PC=转移目标地址（计算结果）。计算指令地址中的PC=(PC)+“1”操作常称为PC增量操作，为了缩短指令周期，PC增量操作通常放在取指令步骤中完成，而转移型指令的下条指令地址计算通常放在取操作数、数据操作步骤中完成。

【数据通路的部件组成】取址阶段的操作由PC、IR、MEM、MAR、MDR实现。PC的增量操作为PC=(PC)+“1”时，可用PC实现，此时PC由计数器组成；否则为PC=(PC)+n，需要使用Adder实现。执行指令阶段的操作由GPRs、ExtU、ALU、PSR、MAR、MDR、MEM实现。其中，GPRs、MEM保存源/目的操作数，ExtU实现相对寻址的符号扩展，ALU实现算术运算，PSR保存状态标识。

手机屏幕截图

低可信度描述已自动生成图示, 示意图

描述已自动生成【数据通路的部件互联】（1）总线结构数据通路。总线结构指多个部件输出端通过同一个信号线连接到其他部件输入端的互联方式。总线结构中，所有部件的输出端必须通过三态门连接到总线。组合逻辑部件只有一个输入端或输出端可以直接连接到总线上，因此可以在在入端增设锁存器来避免发生信号冲突。（2）专用结构数据通路。专用结构指部件每个输入端都通过不同信号线连接到其他部件输出端的互连方式，即部件之间根据需要直接互连，又称点点结构。由于同一个输入端连接有多个部件输出端，为了同一时刻只接收一个部件输出端的数据，部件的每个输入端都必须通过多路选择器连接不同的部件输出端，部件输入端只与一个部件输出端连接时除外。总线结构互联简单、分时传送，只能用于多周期CPU；专用结构互联复杂、同时传送，可以用于单周期CPU。

【数据通路的微操作及其控制】原子操作指不能再细化的操作，通常，将CPU内部的原子操作称为微操作（μOP），将实现μOP的部件控制信号称为微操作控制信号，又称为微操作命令（μOPCmd），将完成一个μOP的时间（或启动两个相邻μOP的间隔时间）成为一个节拍，多个μOP可以通过不同的节拍信号进行时序控制，来形成μOP序列。CPU中的μOP主要有寄存器间传送、存储器读、存储器写、算逻运算4种。

（1）寄存器间传送。

图示, 示意图

描述已自动生成

(2)存储器读/写

手机屏幕截图

描述已自动生成

手机屏幕截图

描述已自动生成

（3）算逻运算

图示, 示意图

描述已自动生成

【数据通路的设计】单周期CPU设计简单，指令周期由一个时钟周期组成，数据通路只能采用专用结构，以减少μOP序列的部署；并且所有部件都不能复用，使用次数超过一次的部件都要重复配置。多周期CPU性能较好，指令周期由多个时钟周期组成，数据通路可以采用总线结构或者专用结构；部件在指令周期中可以复用，不同指令周期的周期数可以不同。

手机屏幕截图

描述已自动生成【控制器的基本组成】CPU的六大功能，中，除数据加工、外部访问外的所有功能都由控制器来实现，包括指令控制、操作控制、时间控制、异常及中断处理，控制器由指令部件、控制单元CU、中断机构组成。

【控制器的工作原理】CPU的工作流程由循环的指令周期、中断周期组成，工作流程中的所有操作都通过μOP序列来表示。因此，控制器的主要功能是循环地、有序地产生CPU工作流程所对应μOP序列的μOP控制信号。

【控制器的类型】控制器的核心是CU，根据CU中μOP控制信号的产生方法不同，控制器有硬布线控制器（hardwired controller）、微程序控制器（microprogrammed controller）两种类型。硬布线控制器中，采用有限状态机方法来描述CPU工作流程所需要的μOP控制信号；采用组合逻辑电路来产生当前状态的μOP控制信号，不同状态用不同的时序信号来表示。因此，时序信号形成电路中，时序信号的循环周期为每个μOP序列的步数，μOP控制信号的形成与时序信号有关。微程序控制器中，采用微程序方法来描述CPU工作流程所需要的μOP控制信号，每一条微指令描述一个状态的μOP控制信号；微程序存放在专门的存储器中，通过执行微指令来产生当前状态的μOP控制信号，不同状态通过微指令的串行执行来表示。因此，时序信号形成电路中，时序信号的循环周期为一个微指令周期，相当于一个μOP时延，μOP控制信号的形成与时序信号无关。

【时序信号的组成】CPU的工作流程可划分为若干个机器周期（又称CPU周期），每个机器周期完成一个基本功能，如取指令、取操作数、数据操作、存操作数、中断响应等。一个机器周期可能需按序完成多个μOP，因而，每个机器周期可以划分成若干个节拍，每个μOP在一个节拍内完成，如单总线数据通路的取指令机器周期包含3个节拍。为了实现μOP的功能，每个节拍内可能需要同步脉冲来配合工作，不同μOP的同步脉冲边沿在节拍中的位置可能不同，因而，每个节拍可能包含若干个工作脉冲。工作脉冲的个数与节拍的宽度无关，仅与各μOP的特性有关。因此，CPU中的时序信号有机器周期、节拍、工作脉冲三种类型，不同指令周期所含的机器周期数、每个机器周期所含的节拍数都可能不同。

【时序信号的定时方式】CPU工作流程中的所有操作都可以通过μOP序列来表示，不同μOP的时延可能不同。μOP的定时方式指μOP的时长控制方法，又称为控制器的控制方式。常见的控制方式有同步控制、异步控制、联合控制三种。（1）同步控制方式。同步控制方式中，每个μOP的时序都完全受统一的基准时钟信号所控制，即每个μOP都在一个时钟周期内完成。基准时钟信号指频率固定的时钟脉冲信号，称为主时钟脉冲信号。主时钟脉冲信号的宽度称为时钟周期，其频率称为CPU主频。同步控制方式中，时钟周期为所有μOP时延的最大值，每个节拍周期等于一个时钟周期。（2）异步控制方式。异步控制方式不存在基准时钟信号，每个μOP的时序只受专门的联络信号控制，即CU发出μOP控制信号后，等待对应部件完成操作，收到其返回的应答信号（如操作完成）后，CU才可以发出下个μOP控制信号。异步控制方式中，每个节拍周期的值都不固定，完全取决于应答信号何时到达。（3）联合控制方式。联合控制方式是同步控制与异步控制的结合，每个μOP的时序受基准时钟信号及应答信号的共同控制，即μOP的定时以同步控制方式为基础，可以支持异步控制方式。联合控制方式中，同步控制时的一个节拍周期等于一个时钟周期，异步控制时的一个节拍周期等于多个时钟周期，所需时钟周期数取决于应答信号的时延。同步控制与异步控制的方式转换通常采用延长节拍的方法实现。

【异常及中断的基本概念】CPU执行程序过程中，可能会遇到一些特殊情况，必须进行相应处理后，才能继续执行当前程序。这些改变程序正常执行顺序的特殊情况常称为事件（Event），根据事件发生的位置，这些事件可分为异常（Exception）、中断（Interrupt）两大类。异常和中断事件的处理都是通过执行相应的处理程序来实现的。

【异常及中断的基本概念.异常】异常指由CPU内部执行指令所引起的意外事件，又称内部异常或程序性异常。异常由执行指令引起，异常一旦发生，程序就无法继续执行，必须立即进行处理。按照异常的报告及返回方式，异常可分为故障（Fault）、陷阱（Trap）和终止（Abort）3类。

手机屏幕截图

描述已自动生成

【异常及中断的基本概念.中断】中断指由CPU外部的设备产生的请求事件，又称外部中断。外设在完成操作或处于某些状态时，会向CPU发出中断请求，这就产生了中断事件。根据中断事件的紧急程度，中断有可屏蔽中断、不可屏蔽中断两种。可屏蔽中断指可以暂不处理（或稍后处理）的中断，如键盘中断、打印机中断等设备请求；不可屏蔽中断指必须立即处理的中断，如电源故障、线路故障、存储器校验错等硬件错误。CPU的状态寄存器中设置有一个“中断允许”标志（常记为IF），用IF=0、IF=1分别表示当前处于屏蔽中断、允许中断状态。

表格

描述已自动生成

【异常及中断的处理过程】异常或中断事件发生时，事件处理过程都由响应、处理、返回3个环节组成。响应指CPU从当前程序转到处理程序的过程，处理指CPU执行处理程序处理事件的过程，返回指CPU从处理程序返回当前程序的过程，异常中的故障、终止可能不返回当前程序的过程，直接终止当前程序的执行或重新启动计算机。响应需要完成保存断点及程序状态、关中断、识别事件类型并转入处理程序这三个任务。（1）保存断点及程序状态。断点、程序状态（如PSW）必须保存到特定寄存器中，返回时再恢复到PC、状态寄存器PSR中；其余用户可见寄存器（如GPRs）则由事件处理程序负责保存与恢复。同时，异常类型号也必须保存到专用寄存器中。（2）关中断。新的事件仅为中断事件，使CPU处于屏蔽中断状态（IF=0）即可实现。由于返回时会恢复PSW，IF亦会恢复为程序的原来状态。（3）识别事件类型并转入处理程序。这项工作的目标是将准备处理的事件处理程序入口地址写入PC。由于CPU同时只能处理一个事件，因此需要先识别事件类型（找出最紧急的事件），再获得相应处理程序入口地址，最后写入PC。识别事件类型、获得处理程序入口地址的方法有非向量方式、向量方式两种。非向量方式指所有时间共用一个处理程序，该处理程序入口地址是固定的，响应时将该处理程序入口地址写入PC，识别事件类型、获得处理程序入口地址安排在事件处理时（处理程序中）进行，故又称软件识别方式。向量方式指每个事件有一个处理程序，所有处理程序的入口地址存放在一个管理表（常称为中断向量表）中，响应时先找出最紧急的时间，再用该事件类型号查中断向量表，来获得相应的处理程序入口地址，故又称硬件识别方式。

**第6章 总线及互连**

【总线的分类.按信号线功能分类】（1）数据总线（Data Bus，DBus）。数据总线用来承载设备间传输的数据内容，是双向传输线。数据总线的位数称为数据总线宽度。（2）地址总线（Address Bus，ABus）。地址总线用来指出传输的数据所在的主存单元地址或外设地址。地址总线是单向传输线，只有主设备才会发出地址信息。地址总线的位数称为地址总线宽度，它决定了可寻址的地址空间大小。（3）控制总线（Control Bus，CBus）。控制总线用来传输控制过程中主/从设备如何使用地址总线和数据总线。由于信息传输是一个交互过程，一部分信号（如传输命令）由主设备发出，另一部分信号（如完成状态）由从设备发出。因此，控制总线的信号线有控制线、状态线两种类型，它们都是单向传输线。

【总线的分类.按连接部件分类】（1）片内总线。指芯片内部的总线，用于连接芯片内部的元器件，片内总线只含数据总线。（2）系统总线（System Bus）。指计算机内部连接CPU、主存、外设等主要部件的总线。系统总线大都由数据总线、地址总线及控制总线组成。

图示

描述已自动生成

（3）通信总线。指连接主机与外设之间、计算机之间的总线。通信总线通常只有数据线和控制线，有时只有数据线，控制信息通过数据线来传送。

【总线的特性】

图示

描述已自动生成

【总线的性能指标】（1）总线宽度。总线宽度指数据总线的位数，它反映了可同时传输的二进制位数。（2）总线带宽。总线带宽指总线的最大数据传输率，即总现在进行数据传输时，单位时间内最多可传输的数据位数，通常用Mbps或MBps表示。总线的数据传输率可以表示为总线宽度乘数据传输次数/秒。（3）总线负载能力。总线负载能力指总线信号的电平保持在有效范围内时，所能连接部件或设备的数量，常用“个”表示。

【总线的操作过程】（1）申请及分配阶段。需要使用总线的主设备向总线仲裁器提出请求，总线仲裁器确定哪个申请者获得下一个总线传输周期的总线使用权。总线仲裁其应该在有总线请求、总线空闲时进行仲裁。仲裁的结果为某个主设备的BG信号有效。为了实现总线分时共享，要求主设备在完成数据传输时，应撤销总线使用请求。（2）寻址阶段。总线传输周期中，获得总线使用权的主设备通过总线发出本次访问的从设备地址和操作命令，以选择及启动参与本次传输的从设备。（3）传送数据阶段。主设备和从设备进行数据交换，数据由源设备发出，经数据总线流入目标设备。读操作时，源设备为从设备、目标设备为主设备，写操作时刚好相反。（4）结束阶段。数据交换完成后，主设备、从设备都需要从总线上撤销自己所发出的信号，来让出总线使用权。

图示, 示意图

描述已自动生成【总线的集中式仲裁】（1）链式查询方式。链式查询方式的基本思想是，仲裁通过自动轮询各个主设备来实现，有总线请求、被询问的主设备获得总线使用权。仲裁在有总线请求、总线空闲时开始，在总线忙时结束。这种仲裁方式实现的是固定优先级策略，优先级由主设备与总线仲裁器的距离决定。链式查询仲裁方式的特点是，仲裁信号线最少（2根），但不能保证公平性，对电路故障很敏感，容易产生断链现象。

图示, 示意图

描述已自动生成（2）计数器定时查询方式。计数器定时查询方式的仲裁信号线与链式查询基本相同，不同的是用设备号总线（log2n根）代替菊花链连接，以避免断链现象。

图示

描述已自动生成（3）独立请求方式。独立请求方式的基本思想是，根据请求引脚的连接次序来进行仲裁，因而无需询问主设备，仲裁结果直接发送到相应的主设备。独立请求方式的特点是仲裁速度快，可以保证公平性，可以实现隐藏式仲裁，但仲裁信号线数量最多（2n根），仲裁电路较复杂。

**第7章 输入输出系统**

【外设与主机的联系.连接方式】外设与主机的连接方式有分散连接和总线连接两种，目前全部采用总线连接方式。采用总线方式互联时，外设必须通过I/O接口与总线连接，每个I/O接口连接一个外设。I/O接口一方面按照总线标准连接到总线，以满足总线操作要求；另一方面按照设备接口约定连接到外设，以满足具体设备的传输与控制要求。采用总线方式互连时，主设备（如CPU）按地址访问（输入或输出）I/O接口，I/O接口使用寄存器暂存相关信息，等到设备就绪时再转发主设备的操作。

【外设与主机的联系.编址方式】主存、外设都连接在总线上，主存地址为主存单元的地址，外设地址为I/O端口的地址，必须使用一定的方法来区分两类地址。外设的编址方式图示

描述已自动生成通常有统一编址、独立编址两种。（1）统一编址。统一编址指I/O端口与主存单元统一编址，共用一个地址空间，地址范围必须不重叠，又称存储器映像方式。统一编址方式下，指令系统无需设置I/O指令，CPU访问外设（I/O操作）可以借用访存指令来实现。因而，总线的控制信号线只需要有存储器读（MemR#）、存储器写（MemW#）即可。统一编址方式的特点是不增加机器指令数，但主存空间变小、不易扩展，地址译码也因全部地址线都参与而变得复杂。图示

描述已自动生成（2）独立编址方式。独立编址指I/O端口与主存单元都从零开始编址，使用不同的地址空间。独立编址方式下，指令系统需要增设I/O指令，CPU访问外设通过I/O指令来实现，访问主存通过访存指令来实现。因而，总线的控制信号线需要有MemR#、MemW#、IOR#、IOW#四种。独立编址方式的特点是主存和I/O空间互不影响、易扩展，地址译码简单，但需要增加两条机器指令。目前使用较多的是独立编址方式。

【外设与主机的联系.目标设备识别】总线事务由地址期、数据期组成，总线标准要求从设备（如外设）在地址期主动识别自身是否为总线事务的目标从设备，被选中时响应总线事务。连接在总线上的每个外设都有一个唯一的设备号（设备ID），其所连接的I/O接口使用硬件（如寄存器）保存了这个设备号。CPU进行I/O操作时，通过I/O指令的地址码字段指明了目标设备地址，通过BIU发起相应的I/O总线事务。外设（I/O接口）识别自身是否被选中的方法是，一直监视总线状态，当有I/O总线事务时，将总线上地址与自身设备ID进行比较，来判断自身是否为I/O总线事务的目标从设备。

【外设与主机的联系.联络方式】（1）数据传送方式。外设的数据传送方式有无条件传送、条件传送两种类型。无条件传送方式也称同步传送方式，主机可以直接与外设进行数据传送，常用于简单设备。条件传送方式又称异步传送方式，传送前需要先启动外设，等到设备就绪时，才能进行数据传送，常用于复杂设备。（2）外设与I/O接口的联络方式。外设与I/O接口的联络方式指信息传送时信号的定时方式，取决于外设的数据传送方式，不同的联络方式所需的联络信号线不同。对于无条件传送方式而言，外设随时可以接收或发送数据、无需预先联络，常称为立即响应方式，外设与I/O接口的连接线只有数据信号线。对于条件传送方式而言，并行传输设备通常采用异步联络方式，设备与I/O接口的连接线有数据线（n位）、请求/应答线等；串行传输设备可采用异步联络方式或同步联络方式，与I/O接口的连接线有数据线（1位）、请求/应答先同步时钟线等。

【I/O的传送控制方式.程序直接控制I/O】程序直接控制I/O方式指I/O完全依靠程序来实现，又称程序控制I/O方式，有程序查询、直接传送两种类型，分别适用于条件传送方式、无条件传送方式。程序查询方式又称轮询（Polling）方式，指CPU启动设备后，不断查询设备是否已做好传送准备，只有在设备就绪时，才进行数据传送。直接传送方式指CPU无需启动设备及查询设备状态，就可以直接进行数据传送，可以看作程序查询方式的特例。

【I/O的传送控制方式.程序中断I/O方式】又称中断驱动I/O（interrupt driven I/O）方式，指CPU启动设备后，继续执行现行程序，外设准备就绪后主动向CPU提出中断请求，CPU响应中断请求，转去执行中断服务程序进行数据传送，需要时可以再次启动设备，然后返回现行程序继续执行。

【I/O的传送控制方式.直接存储器存取方式】程序直接控制方式、程序中断方式都只能实现CPU与外设间的I/O，批量传输的数据都需要缓存在主存中，每个数据还需要再用一条指令传送到主存中，因此I/O性能不够理想。直接存储器存取（Direct Memory Access，DMA）方式指外设与主存间可以直接进行数据传送，传送由专用硬件（DMA接口）控制总线来实现，而不需要CPU干预。DMA方式每次由DMA接口实现一批数据（如4KB）的传送，期间CPU可以与外设并行工作，只有传送准备、结束处理工作还是由CPU负责，因此CPU工作效率极高。

【I/O的传送控制方式.通道方式和I/O处理机方式】通道方式是DMA方式的进一步发展，通道硬件能够实现DMA方式中由CPU负责的外设管理、外设状态检测、传送异常检测等工作，CPU只需执行I/O指令启动某个通道即可，进一步减少了I/O所占的CPU时间。I/O处理机（IOP）方式是通道方式的进一步发展，I/O处理机能够完成的通道方式中由CPU负责的码制变换、格式处理、数据校验等工作，更进一步减少了I/O所占CPU时间。