第一章 引言

1. 什么是ISA？

指令集体系结构，ISA（Instruction Set Architecture）是在机器上运行的所有软件和执行这些软件的硬件之间的协定接口。【ISA实现了人机对话】

2. 计算机的硬件和软件等效原理？

硬件和软件等效原理：任何可以利用软件实现的事情可以利用硬件来实现。反之，任何可以利用硬件来实现的时间也同样可以利用软件来实现。

3. 列举计算机的三种基本部件（在最基本的层次上）

①用来解释和执行程序的处理器 ②用来存储数据和程序的存储器 ③与外界进行数据传输的机制

4. 通常用来测量计算机时钟频率的单位是什么？

时钟频率的度量单位为每秒周数，或称为赫兹。

5. 列举两种类型的计算机存储器

RAM（随机存取存储器）/ROM（只读存储器） 主存/硬盘

6. 解释SSI、MSI、LSI、VLSI的区别

小规模集成电路（SSI）：每块芯片上只有10-100个元件；中规模集成电路（MSI）：每块芯片上有100-1000个元件；大规模集成电路（LSI）：每块芯片上集成的元件多达1000-10000个；超大规模集成电路（VLSI）：每块芯片上集成的元件超过10000。

7. 冯·诺依曼体系结构与以前的计算机体现结构有什么不同？

在最早期的电子计算机中，编程就是利用各种导线进行接插连线，由于没有计算机的分层结构，对早期的计算机进行编程工程浩大。而冯·诺依曼体系结构是①三大硬件系统（中央处理器CPU（包含一个控制单元、一个算术逻辑单元、若干个寄存器）、主存储器系统Main-memory system、输入输出系统I/O）构成②具有执行顺序指令的处理能力，③在主存储器与CPU的控制单元之间，包含一条物理上的或者是逻辑上的单一信道，可以强制改变指令和执行的周期【冯·诺依曼瓶颈】的存储器体系结构，其工作原理是取指-译码-执行。

8. 取指-译码-执行周期（fetch-decode-execute cycle）的工作原理是怎样的？

①控制单元从计算机的存储器中提取下一条程序指令，并利用程序计数器来决定这条指令的所在位置 ②对提取的指令进行译码，变成算术逻辑单元（ALU）能够理解的一种语言 ③从存储器中取出执行指令所需的各种操作数的数据，并把它们放入CPU的寄存器中 ④ALU执行指令，并将执行的结果放到寄存器或存储器中。

* 晶体管（transistor）是英文单词转移电阻（transfer resistor）的缩写。事实上，晶体管是一种利用固体硬件制造的真空三极管。
* 摩尔定律：硅芯片的密度每18个月翻一番。
* 计算机的分层组织结构：

6、用户层，由各种应用任务组成。

5、高级语言层，由各种高级编程语言组成

4、汇编语言层，包含某种类型的汇编语言。

3、系统软件层，主要处理操作系统的命令。

2、指令集体系结构（Instruction Set Architecture），或者称为机器层，由特殊的计算机系统结构所能识别的机器语言组成。

1、控制层。在这一层次上，控制单元将确保正确地译码并执行指令，并且适时地将数据传送到正确的位置。

0、数字逻辑层，在这里我们面对的是计算机系统的物理构成：各种逻辑门和引线。

第四章 MARIE简单计算机模型

1. CPU的主要功能是什么？

中央处理器的任务就是提取程序指令，并对指令进行译码，然后按程序规定的顺序对正确的数据执行各种操作。

2. 控制单元的任务是什么？

控制单元负责 ①监视所有指令执行和各种信息的传送过程 ②从内存提取指令，对这些指令进行译码，确保数据适时地出现在正确的地方 ③通知ALU应该使用哪一个寄存器，执行哪一些中断服务程序，以及对所需执行的各种操作接通ALU中正确的电路 ④使用程序计数器来寻找下一条要执行的指令的位置，并使用一个状态寄存器来存放默写特殊的操作状态（溢出、进位、借位等）。

3. 寄存器安置在什么位置？有何种不同类型的寄存器？

寄存器位于处理器内部。包括：存储信息的寄存器、进行数值移位的寄存器、进行数值比较的寄存器和计数寄存器、还有“中间结果”寄存器【用来存储临时数据值】、变址寄存器【用来控制程序的循环操作】、堆栈指针寄存器【用于管理所处理的信息堆栈】、状态寄存器【用于保持各种工作状态或操作模式（如溢出、进位或一些零条件等）】、通用寄存器【程序员可以访问的寄存器】。

4. 什么是总线周期？

总线周期是完成总线信息传送所需的时钟脉冲间的时间间隔。

5. 列举出三种不同类型的总线，并指出这些总线在计算机中的位置

典型的总线包括数据总线【传递在计算机不同位置之间移动的信息】、地址总线【指出数据读写的位置】、控制总线【指示哪个设备允许使用总线，以及使用总线的目的；传递有关总线的请求、中断和时钟同步信号的响应】、电源线。

按种类：处理器-内存总线（处理器与机器匹配内存系统之间）、I/O总线（连接各种不同的带宽的各种设备）、底板总线（将计算机的所有部件在主板上连接起来）。

6. 总线仲裁的4种方式是什么？

①菊花链仲裁方式 ②集中式平行仲裁方式 ③采用自选择的分配式仲裁方法 ④采用冲突检测的分配式仲裁方法

7. 说明取址-译码-执行周期的各个步骤。

CPU首先提取一条指令，即将指令从主存储器转移到指令寄存器；接着对指令进行译码，即确定指令的操作码和提取执行该指令需要的数据（并且把它们放到寄存器中）；然后执行这条指令，即执行指令规定的各种操作（并且将执行结果存放到寄存器或者存储器中）

8. 中断驱动的I/O是如何工作的？

CPU收到一个来自I/O设备的中断信号，表示输入或输出的准备工作已经完成。利用某些方法可以使CPU从正常的取指-译码-执行周期转去“识别”这个中断请求。

9. 解释硬连线控制和微编程控制的区别（控制单元的两种设计方式）

定义？

硬连线控制的速度快，缺点是指令集和控制逻辑通过特殊的电路直接连接，使得电路变得比较复杂，设计或者修改都比较困难。微编程（利用某种低级语言进行编写）控制的优点是如果指令集需要修改，只需要简单地更新微程序，而实际的硬件部分不需要做任何改变，缺点是所有指令都需要经过一次额外的翻译过程，这样会减慢整个程序的执行速度。

10. 编译程序的任务

使用助记符号将汇编语言转换成机器语言。

* CPU的组成部分：
  + 寄存器是一种存储二进制数据的硬件设备。用来存储如地址、程序计数器或者是程序执行所需要的数据等各种数据。
  + 算术逻辑单元（Arithmetic Logic Unit，ALU）在程序执行过程中用来进行逻辑运算和算术运算。
  + 控制单元（Control Unit）见第2题。
* 总线（bus）是一组导电线路的组合，它作为一个共享和公用的数据通道将系统内的各个子系统连接在一起。【CPU通过总线与其他组件通信】总线可以是点到点的，也可以是共享通道。
* 中断（interrupt），改变（或中断）系统正常执行流程的各种事件，包括I/O请求、出现算术错误、算术溢出、硬件故障、用户定义的中断点、页面错误、非法指令等。
* 大部分计算机采用的中断处理方法是：在机器的每一个取指-译码-执行周期的开始处，先检查是否有中断请求存在。如果有，CPU先处理中断任务，中断处理完成后，CPU会继续进行机器的指令执行过程；如果没有，CPU则进行正常的程序执行过程。
* 两种方法正确设置各条控制线：优缺点见第9题

1. 硬连线控制，从物理上将各条控制线与实际的机器指令连接起来。

2. 微编程控制，即用软件来进行控制。所有的机器指令都被放置到一个专用的程序（微程序）中，然后将这些指令转换成规定的控制信号。

* 两种编址方式：

1. 按字节编址，每一个字节（8为二进制数）都有一个唯一的地址。

2. 按字编址，每一个机器字（不是一个字节，可能是2个或者4个等）都有一个唯一的地址。

* 时钟周期与时钟频率互为倒数。1MHz = 1 million cycles per second

CPU time =

第五章 指令系统体系结构

1. 什么是扩展操作码？

指令总长度固定的体系结构，但是却可以允许操作数域的位数根据需要而改变。

2. 设计指令系统可以采用的三种体系结构

堆栈体系结构，累加器体系结构，通用寄存器（GPR）体系结构

3. 固定长度和可变长度的指令结构各有什么优点？

固定长度：指令执行速度快，但是会浪费一些存储空间。

可变长度：可以节约存储空间，但是译码会变得比较复杂

4. 什么是寻址方式？

寻址方式是指定指令中操作数的位置的方法。

5。 解释有关流水线体系结构的基本概念

有些CPU会将取址-译码-执行周期分为一些较小的步骤，其中某些较小的步骤可以并行执行。这种时间上的交替可以加快CPU的执行速度。这种方法称为流水线。

* 计算机指令由操作码和操作数组成。操作码指定要执行的操作类型，操作数指出数据所处的寄存器和内存单元。
* 衡量指令体系结构（ISA）的性能的因素：
* 程序执行指令时占用内存空间的大小
* 指令系统的复杂程度，主要指指令执行所需要的译码数量和指令执行的任务的复杂性
* 指令的长度
* 指令系统中指令的总数目
* 三种设计CPU存储方式（区分指令系统的体系结构的基本方法）：
* 堆栈体系结构，使用一个堆栈来执行各种指令，而且指令那个的操作数就隐含地存放在堆栈的顶部。具有好的编码密度和一个简单的表达式估值模型。但是由于不能对堆栈进行随机访问，很难产生高效率的编码。【适用于存储器速度快】
* 累加器体系结构：将其中一个操作数隐含在累加器中。大幅降低了机器的内部复杂性，而且允许使用非常短的指令。但是由于累加器只是临时存储，对存储器访问非常频繁。
* 通用寄存器（GPR）体系结构：采用多个通用寄存器组，访问速度比存储器快得多，非常方便编译器进行处理，并且可以十分有效和高效地使用。【适用于存储器速度慢】
* 寻址方式：
* 立即寻址（immediate addressing），在指令中操作代码后面的数值会被立即引用。
* 直接寻址（direct addressing），在指令中直接指定要引用的数值的存储器地址。
* 间接寻址（indirect addressing），地址域中的二进制数用来指定一个存储器地址，该地址中的内容将被用作一个指针。操作数的有效地址就是通过访问这个存储器地址来获取的。
* 寄存器寻址（register addressing），采用一个寄存器来代替存储器存储操作数。
* 变址寻址（indexed addressing），一个贬值寄存器用来存储一个偏移量，将这个偏移量与操作数相加，就产生了指令所要求的有效地址。基址寻址（based addressing），使用基地址寄存器代替变址寄存器
* 堆栈寻址（stack addressing）
* 指令类型
* 数据移动指令，最常用
* 算术运算指令
* 逻辑指令
* 输入输出指令
* 控制指令：分支转移、跳过和进程调用等。

第六章 存储器

1. SRAM（静态随即访问存储器）和DRAM哪一个速度更快？

SRAM不需要刷新电路就可以更新数据，速度更快。

2. 直接映射高速缓存中地址的三个域分别是什么？如何使用这些域来访问存放在高速缓存中的一个数据字？

字域【有时又称为偏移量域，用来唯一地识别和确定来自某个指定的数据块中的数据字】，块域【必须选择一个唯一的高速缓存块】，标记域【在复制主存储器中的数据块到高速缓存时，标记会被一起存储在告诉缓存中，并且通过标记可以唯一识别和确定该数据块】。

CPU生成主存储器地址，访问与块域对应的高速缓存块，比对标记域与高速缓存标记，相同则取出为字域偏移量的数据返回CPU。

3. 组关联告诉缓存的地址分成哪三个域？如何使用这些域来访问存放在高速缓存中的一个数据字？

标记域、组域、字域。CPU生成主存储器地址，访问与组域对应的高速缓存块组，比对标记域与高速缓存标记，相同则取出为字域偏移量的数据返回CPU。

4. 严格来说，什么是有效存取时间（EAT）？

有效存取时间（Effective access time），使用命中率与相连存储器层次的相对访问时间产生的加权平均。

5. 说明如何推导有效存取时间公式

EAT = H·Accessc + (1-H)·AccessMM。

其中H为高速缓存命中率，Accessc是高速缓存访问时间，AccessMM是主存储器访问时间。

6. 什么是脏块？

脏块（dirty block）是指高速缓存中已经被修改过的数据块。

* 存储器分层结构系统的基本类型包括：寄存器、高速缓存、主存储器和辅助存储器。
* 引用的局部性有三种基本形式：
* 时间局部性（Temporal locality），最近访问过的内容很可能在不久的将来再次被访问
* 空间局部性（Spatial locality），对存储器地址空间的访问形成团簇的集中倾向。
* 顺序局部性（Sequential locality），访问存储器的指令倾向于按顺序执行。
* 全关联高速缓存，不为主存储器中的数据块唯一指定在高速缓存中的存放位置，而是允许主存储器中的数据块可以存放到高速缓存的任意位置。

第七章 输入/输出和存储系统

1. 用文字描述Amdahl定理

计算机系统整体性能的提升（称为加速率，speedup）取决于某个特定部件本书呢的加速率和该部件在系统中的使用率。

S =

其中，S代表系统整体性能的加速率；f表示部件在系统中的使用率；k表示新部件的加速率。

2. 什么是加速率？

加速率（speedup）是计算机整体性能的提升。

3. 解释程序控制的I/O与中断控制的I/O有什么不同

使用程序控制方式的CPU需要持续检查其附属设备是否有任何输入请求，CPU会持续处于一个忙等待（busy wait）的循环中，直到开始某个I/O请求。如果没有I/O任务需要处理，CPU就无法从事任何有用的工作。

中断控制的I/O在有数据发送需求时有外部设备来通知CPU。如果没有外部设备发出服务请求来中断CPU，CPU就可以继续执行其他任务。

Ps：相似之处：它们都可以对I/O服务程序进行修改来适应外部硬件设备的改变。

4. 直接存储器访问（DMA）的工作原理是什么？

存储器中装入了所要求的数值后，CPU就会发送信号到DMA子系统。由DMA来负责I/O的细节过程，而CPU会继续执行其下一个任务。在完成I/O处理后（或者是由于错误而结束I/O操作），DMA子系统会发送另一次中断请求通知CPU。

5. 通道控制的I/O与中断控制的I/O有什么不同？

用中断控制的I/O相比，通道控制的I/O只需要很少CPU参与。

6. 通道控制的I/O与DMA有何相似之处？

相同之处：一个IOP必须从CPU中窃取存储器周期。

Ps：不同之处是通道控制的I/O系统都配备单独的总线，用来帮助隔离主机和I/O操作。通道控制的I/O有I/O处理器来控制，具有执行程序的能力。

7. 什么是寻道时间？

寻道时间（seek time）是指磁盘驱动臂定位到指定的磁道上所需的时间。

8. 什么是旋转延迟？

旋转延迟（Rotational delay）是读写头定位到指定的扇区上方所需的时间。

寻道时间+旋转延迟=存取时间， 存取时间+读取时间=传输时间

9. 哪一种RAID技术能提供最佳性能？

RIDO-0不提供冗余，所以具有最佳性能。

10. 哪一种RAID技术最经济，又能提供足够的冗余量？

RAID-5以最少的成本代价提供了最佳的保护。

11. 哪一种RAID技术使用磁盘镜像组？

RAID-1是所有的RAID中具有最佳失效保护的一种方案。RAID-1在每一次写入数据时，都会将数据复制到磁盘驱动器中，称为镜像盘的第二组磁盘上。

* 4种I/O的控制方法
* 程序控制的I/O，每个I/O设备至少一个专用的寄存器。CPU会持续不断地监视每个寄存器，等待数据的到达。这种方法被称为轮询（polling）。
* 中断控制的I/O，CPU不再持续地查询其附属设备是否有任何输入请求，而是在有数据发送需求时由外部设备来通知CPU。
* 直接存储器存取，CPU必须为DMA控制器提供要传输数据字节的地址、字节数，以及目标设备或存储器地址。一旦存储器中装入了所要求的数值后，CPU就会发送信号到DMA子系统。由DMA来负责I/O的细节过程，而CPU会继续执行其下一个任务。在完成I/O处理后（或者是由于错误而结束I/O操作），DMA子系统会发送另一次中断请求通知CPU。
* 通道控制的I/O（channel I/O），一个或多个I/O处理器【IOP，这些小CPU是专门为优化I/O设计的，不同于DMA的控制电路，IOP具有执行能力，包括执行算术逻辑指令和分之转移指令】可以控制多条不同的I/O路径，这些路径就是通道路径。
* RAID-3最适合需要读写大块数据块的情况。
* RAID-5在所有的奇偶校验模型中提供了最佳的读操作吞吐量，并且在写操作时也具有令人满意的吞吐量。
* RAID-6提供了解决大量磁盘时效问题的一种经济使用的解决方案。

第九章 可选择的体系结构

1. Flynn分类法是基于哪两大特性来分类计算机体系结构的？

指令数目和流入处理器的数据流的数目。

四种组合：SISD（单指令流，单数据流），SIMD（单指令流，多数据流），MISD（多指令流，单数据流），MIMD（多指令流，多数据流）

Flynn分类方法的缺点：①对于MISD机器，能够运行的应用程序很少。

②Flynn假定并行执行都是同构的，然而一组处理器的集合可能是同构，也可能是异构的。

* RISC，精简指令集；CISC，复杂指令集；VLIW，超长指令字
* 通过基本的计算机性能等式来量化RISC和CISC的差别：

第十章 性能度量和分析

1. 解释程序或系统属于存储器约束的含义。本章还讨论了其他什么类型的约束？

CPU优化——最大限度地提高CPU所执行的各种操作的速度和效率。

存储器优化——最大幅度地提升代码的存储器管理的效率。

I/O优化——最大限度地增强输入/输出操作的效率。

CPU约束，存储器约束，I/O约束

2. 关于性能优化，Amdahl定律告诉了我们什么？

Amdahl定律指出计算机通过使用某种较快的执行模式可以获得的性能改善程度，受到这种快速模式所占用的系统时间的比例的限制：

S =

其中，S代表系统整体性能的加速率；f表示部件在系统中的使用率；k表示新部件的加速率。