1.任何通用计算机的 4 个主要部件是什么？

存储器：存储数据和指令
运算器：实现二进制数据的运算
控制器：译码和指令的执行
输入输出设备：由控制器操作

2.

基准程序	处理器		
	X	Y	Z
1	20	10	40
2	40	80	20

（a） 首先将 X 作为参考机器，然后将 Y 作为参照机器，分别为每个系统计算算术平均值。辩论直观上该三台机器有大致相等的性能以及算术平均值给出了误导性的结果。

（b） 首先将 X 作为参考机器，然后将 Y 作为参照机器，分别为每个系统计算几何平均值。说明这些结果比算术平均更现实。

将 X 作为参考机器：

基准程序	处理器		
	X	Y	Z
1	1	2.0	0.5
2	1	0.5	2.0
算术平均值	1	1.25	1. 25
几何平均值	1	1	1

将 Y 作为参考机器：

基准程序	处理器		
	X	Y	Z
1	0. 5	1	0.25
2	2.0	1	4.0
算术平均值	1.25	1	2.125
几何平均值	1	1	1

（a）
对于基准 1，机器 Y 的速度是机器 X 的两倍，但是对于基准 2，速度是机器 X 的一半。对于基准测试 1，机器 Z 的速度是 X 的一半。但是对于基准 2 的速度是它的两倍。
直观地说，这三种机器是等价的。
如果将 X 作为参考机器并计算速度算术平均值，我们发现 Y 和 Z 比 X 快 25%。
如果将 Y 作为参考机器并计算速度的算术平均值，我们发现 X 比 Y 快 25% ，Z 比 Y 快两倍。

因此，算术平均值在这里并不具有参考价值。

（b）
当使用几何平均值时，这三台机器在将 X 作为参考机器时性能相同，在将 Y 作为参考机器时性能也相同。这些结果更符合我们的直觉。

3.一个处理器访问主存的平均访问时间为 T2。-个容量比较小的 cache 存储器插在处理器与主存之间。cache 的访问速度比主存快很多，即 T1<T2。 任何时候，cache 保存主存的一部分拷贝，以便在不久的将来 CPU 最可能访问的字在 cache 中。假设处理器访问的下一个字在 cache 中的可能性为 H,即命中率为 H。

(a)对于任何单个存储器访问，在 cache 中访问该字与在主存中访问该字的理论加速比是多少?
加速比=访问主存的平均访问时间/访问 cache 的平均访问时间=**T2/T1**

(b) 假设 T 为平均访问时间，将 T 表示为 T1、T2 和 H 的函数，总加速比与 H 的函数是什么?
加速比=不使用 cache 进行访问的执行时间/使用 cache 进行访问的执行时间
=T2/T
=T2/(H*T1+(1-H)*T2)
=**H*T1+(1-H)*T2**

(e) 实际上，系统被设计为处理器必须首先访问 cache，并决定该字是否在 cache 中，如果该字不在 cache 中,然后才访问主存。因此，当 cache 访问失效(与“命中”相反)时，存储器的访问时间是 T1+ T2。将 T 表示为 T1，T2 和 H 的函数。现在计算其加速比，并与(b)中的结果进行比较。
T=H*T1+(1-H)*(T1+T2)=T1+(1-H)*T2
加速比=T2/T=T2/（T1+(1-H)*T2）
=**1/（(1-H）+T1/T2）**

加速比**比(b)中结果小**。

4.与单总线相比，使用多总线有什么好处?

1、单总线连接设备多的话，传输延迟越大。而这个延迟决定了设备协调总线使用所花费的时间。当总线控制频繁地由一个设备传递到另一个设备时，传输延迟明显的影响性能。而多总线传输延迟短。
2、当聚集的传输请求接近总线容量，总线成为瓶颈。通过提高总线的数据传输率或使用更宽的总线，虽然可以能够缓解。但是挂接设备产生的数据传输率增加更快，这是单一总线的失败，而多总线可以缓冲这些传输。
3、同时允许系统支持更广泛更多的 I/O 设备(总线与 I/O 设备的速度容易匹配)。

5.列出并简要定义指令执行的可能状态。

7 种可能状态如下：
指令地址计算：决定下一条要执行的指令的地址。通常是将一个固定的值与前一条指令的地址相加。
读取指令：将指令从内存单元读到 CPU 中。
指令操作译码：分析指令，以决定执行何种操作及其所用的操作数。
操作数地址计算：如果操作包含对存储器或通过 I/O 的操作数的访问，那么需决定操作数的地址。
取操作数：从存储器或从 I/O 中的读取操作数。
数据操作：完成指令所给出的操作。
存储操作数：将结果写入存储器或输出到 I/O 。

6.一个假想的 32 位微处理器采用 32 位指令格式，这种指令有两个部分：第 1 个字节包含操作码，其余部分是立即操作数或操作数的地址。

（a） 最大可直接寻址的存储器容量是多少？
24 位的操作数地址，共有 2^24=16M 个字节可以直接寻址。

（b） 讨论下面的微处理器总线对系统的影响：
（1） 32 位局部地址总线和 16 位局部数据总线。
32 位地址总线可以寻址 4G 的物理内存空间，浪费 8 根地址线， 16 位 数据总线导致取 1 条指令需要 2 次内存访问，会降低系统的性能。

（2） 16 位局部地址总线和 16 位局部数据总线。
16 位地址总线只能寻址 64K 的物理内存空间，不能支持 16M 内存空间， 16 位数据总线导致取 1 条指令需要 2 次内存访问，会降低系统的性能。

（c） 程序计数器和指令寄存器需要多少位？
操作数地址共有 24 位，因此程序计数器需要至少 24 位，由于采用 32 位指令格 式，故指令寄存器为 32 位。

7.顺序存取，直接存取、关联存取和随机存取 4 者何不同？
顺序存取：存储器组织成许多称为记录的数据单位，它们以特定的线性顺序方式存取。存储的地址信息用于分隔记录和帮助检索。采用共享读写机构，经过一个个的中间记录，从当前的存储位置移动到所要求的位置，因此存取不同数据之间相差很大（依赖于前面存取的序列）。

直接存取：同顺序存取一样，直接存取也采用共享读写机构。但是，单个的数据块或记录有基于物理存储位置的惟一地址。通过采用直接存取到达所需块处，然后在块中顺序搜索，计数或等待，最终到达所需的存储位置来完成存取，同样，存取时间也是可变的（依赖于前面存取的序列）。

随机存取：存储器中每一个可寻址的存储位置有惟一的物理编排的寻址机制。存取给定存储位置的时间是固定的，不依赖于前面存取的序列。因此任何存储位置可以随机选取，直接寻址和存取。 主存储器系统和某些 cache 系统采用随机存取。
关联存取：例如 cache 先直接定位数据集 set ，集内并行比较标签 tag 查找。存取时间是固定的，不依赖于前面存取的序列。

8.考虑一台机器，其主存可以按字节寻址，容量为 2^16 字节，块的大小是 8 字节，假设它使用一个包含 32 行的直接映射式 cahce 。

（a）16 位存储器地址如何划分标记、行号和字节号？
字节号:2^3=8, 行号:2^5=32, 标记：16-5-3=8 。

（b）如下地址将存入 cache 的哪些行？
0001 0001 0001 1 011 第 3 行
1100 0011 0011 0 100 第 6 行
1101 0000 0001 1 101 第 3 行
1010 1010 1010 1 010 第 21 行

(c)假设地址 0001 1010 0001 1010 的字节内容存入 cache 那么与它同存一行的其他的字节的地址各是什么?
0001 1010 0001 1000,
0001 1010 0001 1001,
0001 1010 0001 1110,
0001 1010 0001 1111

(d)存储器总共有多少字节能保存于 cache 中？
32*8=256 字节

(e)为何标记保存在 cache 中？
多个内存块可以装载到同一个 cache 行中，结合标记可以识别一个 cache 行中装载的是哪一个内存块的数据。

9.考虑动态 RAM 每毫秒必须刷新 64 次，每次刷新操作需要 150NS，一个存储器周期需要 250NS 问存储器总操作时间的百分之多少必须用于刷新？

刷新时不能进行动态 RAM 存取，其它时间都可以存取动态 RAM。
1 毫秒中储存周期中刷新的时间为 64 150 ns = 9600 ns.
刷新占总储存周期的时间百分比为 (9.6 10^(-6) s)/10^(-3) s = 0.0096 = 0.96%

10.定义寻道时间、旋转延迟、存取时间和传送时间四个术语。
寻道时间：在可移动磁道系统中，磁头定位到该磁道所花的时间称为寻道时间。
旋转延迟：一旦磁道选定，磁盘控制器将处于等待状态，直到相关扇区旋转到磁

头可读写的位置，这段时间称为旋转延迟。
存取时间：寻道时间和旋转延迟的总和称为存取时间。
传送时间：待磁头定位后，扇区旋转到磁头下面时就可以完成读写操作，这就是整个操作的传送部分，传送所需时间称为传送时间。

11.RAID 系统中如何实现冗余。

数据分布在一组物理磁盘上，冗余磁盘容量用于存储奇偶校验信息，保证磁盘万一损坏时能恢复数据。
在 RAID1 中,采用简单的备份所有数据的方法来实现冗余。
在 RAID2 中，采用了海明码计算模式。
在 RAID2-5 中，采用了某些形式的奇偶校验计算模式.
在 RAID6 中，采用了双奇偶校验计算模式。

12.正上溢，指数上溢和有效数上溢 overflow ，三者的区别是什么？

正上溢：当机器 （ 浮点 / 整数 ） 数值大于最大正数 A 值，或小于最小负数 B 值时。
指数上溢指一个浮点数的正指数超过了最大允许指数值。
有效数上溢是指 2 个同符号数相加，符号位进位 （ 丢失 ） 。

13.以 IEEE 32 位浮点格式表示如下数：

（a） -5
1 10000001 01000000000000000000000

（b）-6
1 10000001 10000000000000000000000

（c）-1.5
1 01111111 10000000000000000000000

（d）384
384 =110000000 = 1.1*2^1000
127 + 8 = 135 = 10000111
0 10000111 00000000000000000000000

（e）1/16
1/16=0.0001 = 1.0*2^(-100)
127 - 4 = 123 = 01111011
0 01111011 00000000000000000000000

(f) -1/32
-1/32 = -0.00001 = -1.0 × 2^(-101)
127 - 5 = 122 = 01111010
0 01111010 00000000000000000000000

14. 列出为过程返回保存返回地址的三种可能位置。

寄存器，被调过程开始处，堆栈顶部。

15. 列出并简要说明指令流水线处理条件转移指令的几种方式。

多个指令流、预取转移目标、循环缓冲器、转移预测、延迟转移。

多个指令流：复制流水线的初始部分，并允许流水线同时取这两条指令，使用两个指令流。

预取转移目标：识别出一个条件转移指令时，
除了取此转移指令后的指令外，转移目标处的指令也被取来。这个目标被保存，直到转移指令被执行。

循环缓冲器：由流水线指令取阶段维护的一个容量小的但极高速的存储器，用来存储一段连续的指令。

转移预测：根据程序指示或执行历史预测一条转移路径，提前转移执行。

延迟转移：自动重排程序中的指令，以致一条转移指令出现在实际所要求的位置之后。无论转移是否发生，转移之后的指令总会执行。利用转移指令直到下一条指令之后才产生影响的特点，在转移指令之后安排一条有用指令来代替仅为延迟的空操作。