

長沙工學院

计算机硕士考研 历年真题答案

九三西域府前進四委員會 編
The Ahead Four Committee of Jiusan Western Region

许可协议

CC BY-NC 4.0

本作品采用[知识共享署名-非商业性使用 4.0 国际许可协议](#)进行许可

你可以自由地

共享 - 在任何媒介以任何形式复制、分享本作品

演绎 - 修改、转换或以本作品为基础进行创作

但需遵守以下条例

署名 - 你必须给出原作者署名及其联系方式，并标明是否对原始作品进行修改

非商业性使用 - 不得将本作品用于商业目的

原作者

九三西域府 (tianwenzy@tianwenzy.com)

弟中弟 (chengyue997@163.com)

详细贡献

皮皮 (基础依托者)

九三西域府 (2019、2018、2017、2013、2012、2011、2008、2005 录入编辑排版)

弟中弟 (2021、2020、2004、2003)

charles meng (2016)

517 (2015、2014)

卢小帅 (2007、2006)

兰斯洛特 (2010)

纠正补充者: LYH、xc、小庄、21-计科-、y、WM、一起卷、华、lpl

素材版权

HarmonyOS Sans: 免费商用 华为&汉仪字库

狮尾 B2 加糖宋体: 免费商用 Max (个人)

教育部隸書: 免費商用 臺灣當局僑教育部

黃令東齊伋體: 免費商用 黃令東 (个人)

方正仿宋: 免费商用 方正字库

Asana: 免费商用 Apostolos Syropoulos (自由字体)

封面 logo: 九三西域府绘制

使用说明

- 本作页面布局已对 B4 进行优化，可直接拿本文件去打印店以 A4 或 B5 打印。
- 按照分年份的方式优先整理了 2003-2019 年的计算机组成原理真题。1996-2002 年的计算机组成原理真题由于没有其他参考答案，暂不整理。
- 对于不完善的真题，比如 2020 年、2021 年的，暂无法详细整理。
- 本答案合集是在“皮皮”整理的《2018-2003 组成原理真题答案》基础上，“九三西域府”进行了版面设计、排版、编辑录入；其他贡献者对错误的答案进行了校正；对矛盾不一的答案进行了统一；对多种说法的概念名词进行了统一；对过时的答案进行了更新；对争议的答案进行了讨论。
- 为了便于选择填空对答案，选填的答案用了黑体加粗红字标出，简答与大题不再做其他的标记处理
- 2019 年是自主整理的，每题都进行了详细的论证，之前年份不一定详细
- 本作对计算类题目尽可能给出有详细步骤和思路的解析。
- 本作对概念性题目尽量有理可依，用黄色高亮来标识出处。
- 对于涉及到的资料做如下的简称：
 - [慕课]中国大学 MOOC 联盟里刘芳、沈力等老师教授的视频及 PPT。
 - [爱课程]爱课程网上唐玉华等老师教授的视频及 PPT。
 - [绿书]大纲指定教材《计算机原理与设计》。
 - [白书]大纲指定教材《数据结构与算法（第 2 版）》。
 - [黑书]大纲指定教材《计算机组成与设计-硬软件接口》。
 - [CSAPP]经典书籍《深入理解计算机系统》。
- 对书籍的出处依据，会说明页行，行数是从顶部往下数的，如：绿书 P4 第 2 行
- 对视频的出处依据，会说明视频文件编号及时间轴坐标，如：慕课视频 2.2.2-2:51 处
- 对于 PPT 的出处依据，会说明何种 PPT 编号及页数，如：爱课程 PPT5.9 第 10 页
- 由于未来几乎不再可能出的名词解释，不会进行深的解析作答。
- 个人整理，用爱发电，难免会存在纰漏或错误。
如果对于某些答案有您自己的见解看法，欢迎指出。
- 个人精力所限，未来不会再更新

目录

2021	1
2020	7
2019	14
2018	19
2017	23
2016	27
2015	30
2014	33
2013	37
2012	41
2011	45
2010	48
2009	51
2008	52
2007	56
2006	60
2005	63
2004	67
2003	72

2021

一、选择

(暂无)

二、填空

1. **(索引存储)(散列存储)** 顺序存储要求逻辑上相邻的元素在物理位置上也相邻,链式存储则借助元素存储地址的指针来表示元素之间的逻辑关系。
2. **(可读性)(健壮性)** 一个好的算法具有正确性、可读性、健壮性以及满足效率与低存储量需求(姑且将该特性等同于题目中的高效性)
3. **(5051)** 一个长度为 n 的字符串其子串个数为: $n(n+1)/2+1$
4. **(不稳定)(稳定)**
5. **(C2420000H)(C1000000H)(C2620000H)(C2220000H)** 题目没说明一般都是短浮点数,则 $x = -48.5 = -110000.1 = -1.100001 \times 2^5$, 对规格化的二进制浮点数,整数部分的 1 将会被隐含表示,不会存储在 23 位的尾数内。在 IEEE754 标准中,短浮点数偏置值为 127,那么阶码 $= 127 + 5 = 132$, 转化为 2 进制后的阶码 $E = 1000\ 0100$, 至此,根据 IEEE754 短浮点数的格式,我们得到 x 的机器数:

m_s (数符)	E (阶码)	M (尾数)
1	1000 0100	1000 0100 0000 0000 0000 0000

同样的 $y = 9 = 1000 = 1.000 \times 2^3$, $E = 3 + 127 = 130 = 1000\ 0010$

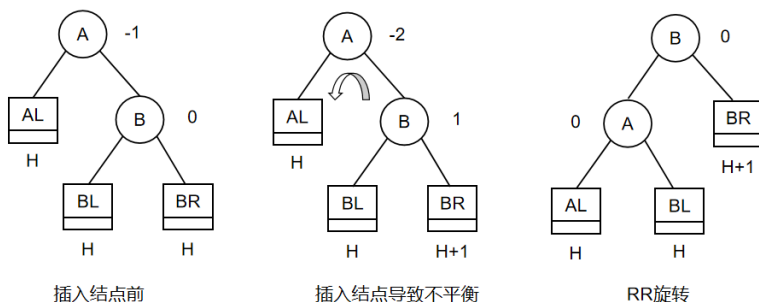
m_s (数符)	E (阶码)	M (尾数)
1	1000 0010	0000 0000 0000 0000 0000 0000

接下来对 $x+y$ 、 $x-y$ 不再赘述。

6. **(FFFFF800H)(FFF80000H)(FFFFFF80H)(OFFF80H)** 16 位补码 0xF800 显然是负数,对负数的扩展高位添 1,因此扩展后的 32 位机器数是 FFFF F800,对补码形式的负数,算术左移低位添 0,算术右移高位添 1,逻辑移位只添 0。
7. **(-1)(0)** 国防科大计算机考研的指定教材**白书**所教授的 $next[i]$ 求解方法等同于**王道/天勤**中的 $nextval[i]$,下标 i 从 0 开始(王道书籍一般从 1 开始)本人一般采用手算最长相等前后缀长度的方法。
 - ‘a’前后缀为空,最长相等前后缀长度=0;
 - ‘ab’的前缀{a}∩后缀{b}=∅,最长相等前后缀长度=0;
 - ‘abc’的前缀{a,ab}∩后缀{b,bc}=∅,最长相等前后缀长度=0;
 - ‘abca’的前缀{a,ab,abc}∩后缀{bca,ca,a}={a},最长相等前后缀长度=1;
 - ‘abcab’的前缀{a,ab,abc,abca}∩后缀{bcab,cab,ab,b}={ab},最长相等前后缀长度=1;
 - ‘abcabc’前缀{a,ab,abc,abca,abcab}∩后缀{bcabc,cabc,abc,bc,c}={abc},最长相等前后缀长度=1;
 - ‘abcabcd’的前缀{a,ab,abc,abca,abcab,abcabc}∩后缀{bcabcd,cabcd,abcd,bcd,cd,d}=∅,最长相等前后缀长度=0;
 具体的数组情况见下表:

编号	0	1	2	3	4	5	6
S	a	b	c	a	b	c	d
PM	0	0	0	1	1	1	0
next[i]	-1	0	0	0	1	1	1
nextval[i]	-1	0	0	-1	0	0	1

8. **(O(n2))**
9. **(10)** 相同的结点树时，深度最小的二叉树肯定是满二叉树，题目就是求 1000 个结点的满二叉树深度，深度为 n 的满二叉树结点数目为 $2^n - 1$ 。根据 $2^n - 1 = 1000$ ，知 $n = \lceil \log_2 1001 \rceil = 10$ 。
10. **(2n)** 扩充二叉树：在二叉树出现空子树的位置增加空树叶所形成的二叉树。记结论：外路径长度 E ，内路径长度 I ，内结点个数 n ，则 $E = I + 2n$
11. **(1) (逆时针) RR 平衡旋转 (左单旋转)**。由于在结点 A 的右孩子 (B) 的右子树 (BR) 上插入了新结点， A 的平衡因子由 -1 减至 -2 ，导致以 A 为根的子树失去平衡，需要一次向左的旋转操作。将 A 的右孩子 B 向左上旋转代替 A 成为根结点，将 A 结点向左下旋转成为 B 的左子树的根结点，而 B 的原左子树则作为 A 结点的右子树，如图所示：



12. **(2)** 无向图的所有顶点度之和等于边数的 2 倍，根据题设，该图所有顶点度之和为 30，未知度数的顶点个数 $= 10 - 4 - 3 = 3$ ，已知度数顶点的度之和 $= 4 \times 4 + 3 \times 3 = 25$ ，那么剩余的 3 个顶点的度数之和必须等于 $30 - 25 = 5$ ，设度为 2 的顶点个数为 x ，就有 $2x + (3 - x) = 5$ ，解得 $x = 2$ 。
13. (暂无)
14. **(10) (4) (1024)** $1M \times 4$ 位的芯片数据线应为 4 根，地址线应为 $\log_2 21M = 20$ 根，而 DRAM 采用地址复用技术，地址线是原来的 $1/2$ ，故地址线数为 10 根。题设计算机地址线 23 根，按 64 位字长编址，那么计算机容量为 $2^{23} \times 64 \text{bit} = 2^{29} \text{bit} = 26M$ ，则所需存储芯片个数为 $26M \times 64 / 1M \times 4 = 1024$ 。
15. **(90.91%) (181.82ns)**

Cache 命中率 = Cache 存取次数 / 总存取次数 $= 2000 / 2200 = 0.90909... \approx 90.91\%$

平均存取时间 = Cache 总存取时间 + 主存总存取时间 / 存取次数

$$= (2000 \times 100 \text{ns} + 200 \times 1000 \text{ns}) / 2200 = 181.818181... \approx 181.82 \text{ns}$$

(或直接用 Cache 存取时间 \times Cache 命中率 + 主存存取时间 \times 主存命中率，结果一样)

三、简答

1. 比较如下
- ① 存取 (读写) 方式：顺序表可以顺序存取，也可以随机存取，链表只能从表头顺序存取元素。例如在第 i 个位置上执行存或取的操作，顺序表仅需一次访问，而链表则需从表头开始依次访问 i 次。
- ② 逻辑结构与物理结构：采用顺序存储时，逻辑上相邻的元素，对应的物理存储位置也相邻。而采用链式存储时，逻辑上相邻的元素，物理存储位置则不一定相邻，对应的逻辑关系是通过

指针链接来表示的。

③查找、插入和删除操作：对于按值查找，顺序表无序时，两者的时间复杂度均为 $O(n)$ ；顺序表有序时，可采用折半查找，此时的时间复杂度为 $O(\log_2 n)$ 。对于按序号查找，顺序表支持随机访问，时间复杂度仅为 $O(1)$ ，而链表的平均时间复杂度为 $O(n)$ 。顺序表的插入、删除操作，平均需要移动半个表长的元素。链表的插入、删除操作，只需修改相关结点的指针域即可。由于链表的每个结点都带有指针域，故而存储密度不够大。

④空间分配：顺序存储在静态存储分配情形下，一旦存储空间装满就不能扩充，若再加入新元素，则会出现内存溢出，因此需要预先分配足够大的存储空间。预先分配过大，可能会导致顺序表后部大量闲置；预先分配过小，又会造成溢出。动态存储分配虽然存储空间可以扩充，但需要移动大量元素，导致操作效率降低，而且若内存中没有更大块的连续存储空间，则会导致分配失败。链式存储的结点空间只在需要时申请分配，只要内存有空间就可以分配，操作灵活、高效。

2. （讲一下基本思路，如果考试遇到最好写点代码）

采用二维数组存储的稀疏矩阵，对角线元素之和 $\sum_{i=0}^{n-1} A[i][i]$ 。

而采用三元组存储的稀疏矩阵，如果使用一维数组存储三元组，那么就需要遍历这个数组，在遍历每一个元素时确认一下行下标与列下标是否一致，一致则将数据域的数值加和，不一致则跳出，形如：

```
1. int sum = 0 ;
2. for(int i = 0 ; i < n ; i++)
3. {
4.     if(A[i].row == A[i].col)
5.         sum += A[i].data ;
6. }
```

如果采用十字链表存储，思路与一维数组相同，代码形式上虽然有所差别，但本质相同。

对于具有 m 个元素的 $n \times n$ 稀疏矩阵：

①采用二维数组存储的稀疏矩阵，由于拥有随机存取的特性，故存取一个元素的时间复杂度为 $O(1)$ ，存储一个二维数组的空间复杂度为 $O(n^2)$ 。

②采用三元组十字链表方式存储，首先建立每行（每列）只有头结点的空链表，并建立起这些头结点拉成的循环链表；然后每输入一个三元组 $(i, j, a[i][j])$ ，则将其结点按其列号的大小插入到第 i 个行链表中去，同时也按其行号的大小将该结点插入到第 j 个列链表中去。建立头结点循环链表时间复杂度为 $O(s)$ ， $s=n$ ，插入每个结点到相应的行表和列表的时间复杂度是 $O(m \times s)$ ，这是因为每个结点插入时都要在链表中寻找插入位置，所以总的时间复杂度为 $O(mn)$ 。如果我们输入三元组时是按以行为主序（或列）输入的，则每次将新结点插入到链表的尾部的时间复杂度为 $O(m)$ 。

3. 计算机中的流水线把一个重复的过程分解为若干子过程，每个子过程与其他子过程并行执行。由于采用流水线技术只需增加少量硬件就能把计算机的运算速度提高几倍，因此成为计算机中普遍使用的一种并行处理技术。

完成同样一批任务，不使用流水线所用的时间与使用流水线所用的时间之比，称为流水线的加速比。

设 T_0 表示不使用流水线时的执行时间，即顺序执行所用的时间； T_k 表示使用流水线时的执行时间，则计算流水线加速比（ S ）的基本公式为： $S = \frac{T_0}{T_k}$ 。

若流水线各段执行时间都相等，则一条 k 段流水线完成 n 个任务所需时间为 $T_k = (kn - 1)\Delta t$ 。而不使用流水线，即顺序执行 n 个任务时，所需的时间为 $T = kn\Delta t$ 。将 T_0 和 T_k 代入上式得实际

$$\text{加速比为 } S = \frac{kn\Delta t}{(k+n-1)\Delta t} = \frac{kn}{k+n-1}。$$

理论上连续输入的任务数 $n \rightarrow \infty$ 时，最大加速比为 $S_{\max} = k$ 。

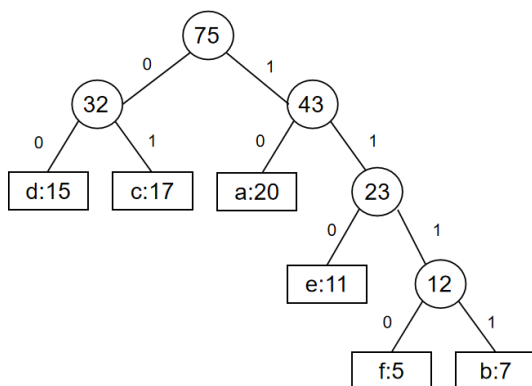
4. 区别如下

- ① 中断程序和正在执行的程序之间是相互独立的；子程序调用装入的子程序与 **cpu** 正在执行的子程序之间是同一程序的两个部分。
- ② 除了软中断，通常中断都是随机产生的；子程序调用是通过子程序调用指令引起的。
- ③ 中断服务程序的入口地址有中断向量地址指出；子程序调用的入口地址有程序调用指令指出。
- ④ 调用中断服务程序和子程序都需保护断点；前者有中断隐指令完成，后者是有指令本身完成。
- ⑤ 处理中断服务程序时，对多个同时发生的中断要进行判优，而调用子程序时必没有这种操作。

四、应用

1. 已知对字符的访问频率为 a:20、b:7、c:13、d:15、e:11、f:5

- (1) 构建哈夫曼树，要求左子结点权值小于右子节点，编码方式左 0 右 1，并求出编码后的信息长度

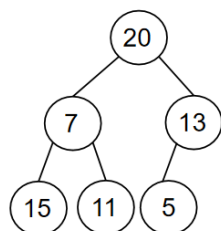


易知各字符编码分别为 a:10, b:1111, c:01, d:00, e:110, f:1110

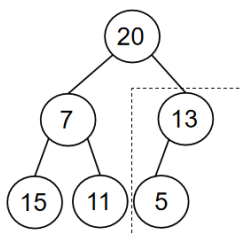
$$WPL = 2 \times (20 + 17 + 15) + 3 \times 11 + 4 \times (7 + 5) = 185$$

- (2) 初始序列为：{20, 7, 13, 15, 11, 5}

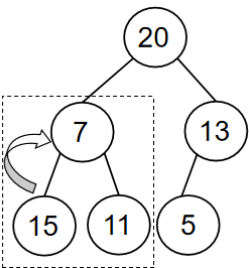
则初始堆为



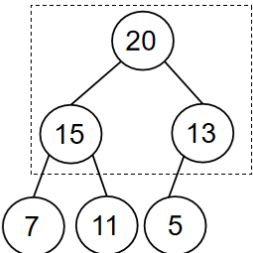
从第 $\lfloor n/2 \rfloor$ 个结点为根的子树依次向前进行筛选， $\lfloor n/2 \rfloor = 3$



此时由于第 3 个结点的子树满足堆定义，因此不发生交换



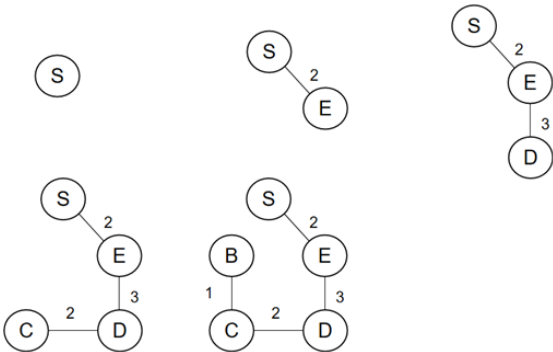
来到第 2 个结点，由于第 2 个结点的值小于左右子结点的值，发生交换



来到第 1 个结点，由于第 1 个结点的子树满足堆定义，因此不发生交换
至此该二叉树满足堆的定义。

2. 参考答案如下

- (1) **Prim** 从选中顶点开始，选择一个与当前顶点集合距离最近的顶点，并将该顶点和相应的边加入最小生成树中，以此类推，直至图中所有的顶点都并入集合，得到的 **T** 就是最小生成树，构造过程如下：



- (2) 变化过程如下

编号	length	pre	编号	length	pre	编号	length	pre
s → S	0	-1	s → S	0	-1	s → S	0	-1
s → B	4	S	s → B	4	S	s → B	4	S
s → C	∞	-1	s → C	8	E	s → C	7	B
s → D	∞	-1	s → D	∞	-1	s → D	11	C
s → E	3	S	s → E	3	S	s → E	3	S

3. 参考答案如下

- (1) 图中 a 为 MDR，c 为 IR，b 为 MAR，d 为 PC。
- (2) 指令 **LDA X** 的数据通路为：
X → MAR → 主存 → MDR → ALU → ACC
- (3) 指令 **ADD Y** 的数据通路为：
Y → MAR → 主存 → MDR → ALU
ACC → ALU

ALU → ACC

4. 参考答案如下

- (1) 已知计算机 M 采用 32 位定长指令字，即一条指令占 4B，观察表中各指令的地址可知，每条指令的地址差为 4 个地址单位，即 4 个地址单位代表 4B，一个地址单位就代表了 1B，所以该计算机是按字节编址的。
- (2) R2 中装的是 i 的值，循环条件是 $i < N$ (600)，即当 i 自增到不满足这个条件时跳出循环，程序结束，所以此时 i 的值为 600。
- (3) 存储器按字节编址，页大小 4KB， $\log_2 4K = 12$ ，故虚页号 12 位。
页内偏移字段 = $32 - 12 = 20$ 位。
TLB 中标记字段的内容为虚页号，故标记偏移字段有 20 位。
16 个页表项说明 TLB 有 16 行，采用 4 路组相联有 4 组，故组号 2 位。
- (4) Cache 块大小为 32B，块内地址字段为 $\log_2 32B = 5$ 位，数据 Cache 块 256 行，直接映射方式下对应行号字段 $\log_2 256 = 8$ 位，标记偏移字段 = $32 - 5 - 8 = 19$ 位；指令 Cache 共 16 行，直接映射方式下对应行号字段 $\log_2 16 = 4$ 位，标记偏移字段 = $32 - 5 - 4 = 23$ 位。
数据 Cache：0~4 位块内地址字段，5~12 位行号字段，13~31 位标记字段。
- (5) 指令 Cache 共有 16 行，每块 32 字节，所以 Cache 数据区的容量为 $16 \times 32B = 512B$ 。P 共有 6 条指令，占 24B，小于主存块大小 32B。其起始地址为 08048100H，对应一块的开始位置，由此可知所有指令都在一个主存块内。读取第一条指令时会发生 Cache 缺失，因此将 P 所在的主存块调入 Cache 的某一行，以后每次读取指令时，都能在指令 Cache 中命中。因此在 600 次循环中。只会发生 1 次访问缺失。所以指令 Cache 的命中率为 $(600 \times 6 - 1) / (600 \times 6) = 99.97\%$ 。
- (6) 指令 4 为加法指令，即对应 $sum += A[i]$ ，当数组 A 中元素的值过大时，会导致这条加法指令发生溢出异常，而指令 2、5 虽然都是加法指令。但它们分别为数组地址的计算指令和存储变量 i 的寄存器进行自增的指令，而 i 最大到达 600，所以它们都不会产生溢出异常。

只有访存指令可能产生缺页异常，即指令 3 可能产生缺页异常。

一开始数组并不在主存中，第一次访问数组时会读取磁盘，把 A 数组调入内存。而以后数组 A 的元素都在内存中，不会再读取磁盘，所以该程序一共读取磁盘一次。

每访问一次内存数据就会查一次 TLB，共访问数组 600 次，所以此时访问 600 次 TLB，还要考虑到第一次访问数组 A，即第一次访问 A[0]时会先查一次 TLB，然后产生缺页，处理完缺页中断后，会重新访问 A[0]，此时又查 TLB，所以访问 TLB 的次数一共是 601 次。

2020

一、选择

1. [D]

2. [A]

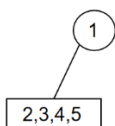
3. [A]分析一下这个代码,我们可以看到循环的条件是 $i+j \leq n$,而在循环体中, i 或 j 的值总会+1,因此我们可以将 $i+j$ 看做一个整体 k ,简化这个代码:

```
01. int k = 1;
02. while( k <= n){
03.     k++;
04. }
```

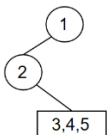
显然这个循环体会执行 n 次,即时间复杂度为 $O(n)$ 。

4. [C]树的先根、后根遍历等同于二叉树的先序、中序遍历,已知一棵二叉树的先序、中序遍历序列可以唯一确定一颗二叉树,因此我们可以先画出二叉树,再将其转化为树。

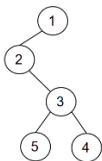
①先序序列为 12354,则 1 为根节点,中序序列为 25341,则 2, 3, 4, 5 四个节点都在 1 这个根节点的左子树上:



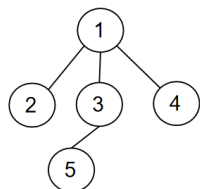
②对于 2,3,4,5 的先序序列为 2354,则 2 为根节点,中序序列为 2534,则 3,4,5 是 2 这个节点的右孩子:



③对于 3,4,5 的先序序列为 354,则 3 为根节点,中序序列 534,则 5 是 3 这个节点的左孩子,4 是右孩子:



根据二叉树转换为树的规则,转换为



至此,我们得到层次序列为 12345。

5. [C]争议 (经大家综合讨论后,普遍认为这道题应该缺了一个 $n=2$ 的某个幂次数的条件,希望

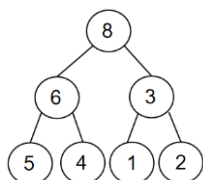
大家酌情参考)在采用双散列函数探测法时,Hash2 跟 Hash1 一样产生一个 $1 \sim n-1$ 之间的数。当 n 为素数时,Hash2(key)可以是 $1 \sim n-1$ 之间的任何数,当 n 是 2 的幂次数时,Hash2(key)可以是 $1 \sim n-1$ 之间的任何奇数。

在该知识点中,涉及到一个互质的概念,Hash2(key)所得到的值与 n 互质(互质是公约数只有 1 的两个整数,叫做互质整数),例如 2 的幂次数这类偶数的因子全是偶数,与任何奇数互质。鉴于选项中只有 C 是奇数,故选 C。

6. **[*(+)]**在完成对 5 的扫描后,操作数栈为 (8, 3, 5),此时运算符栈为 (*, (, +)

待处理序列	当前扫描	运算符栈	操作数栈	操作
8*(3+5)+4	8			8 压入操作数栈
*(3+5)+4	*		8	*压入运算符栈
(3+5)+4	(*	8	(压入运算符栈
3+5)+4	3	*, (8	3 压入操作数栈
+5)+4	+	*, (8, 3	+压入运算符栈
5)+4	5	*, (, +	8, 3	5 压入操作数栈

7. **[B]**大根堆的最大元素存放在根结点,且其任一非结点的值小于其双亲结点值。将选项作为二叉树层次序列画出二叉树,显然只有 B 满足大根堆定义。



8. **[A]**有 n 个顶点的强连通图最多有 $n(n-1)$ 条边,最少有 n 条边
9. **[A]**小端:较高的有效字节存放在较高的存储器地址,较低的有效字节存在较低的存储器地址。
大端:较高的有效字节存放在较低的存储器地址,较低的有效字节存放在较高的存储器地址。

地址	0x00	0x01	0x02	0x03
大端模式	12	34	56	78
小端模式	78	56	34	12

把 int 型变量赋值给 char 型变量时,由于 Int 型变量为 32 位,char 类型只有 8 位,直接将低 8 位截取,其余的丢弃,因此答案为 0x12。

10. **[C]**

11. **[D]** a: 42 28 00 00, 则 $[a]_2 = 0100\ 0010\ 0010\ 1000\ 0000\ 0000\ 0000\ 0000$
b: 42 08 00 00, 则 $[b]_2 = 0100\ 0010\ 0000\ 1000\ 0000\ 0000\ 0000\ 0000$

在 IEEE754 标准下分别为:

	m_s (数符)	E (阶码)	M (尾数)
a	0	1000 0100	0101 0000 0000 0000 0000 000
b	0	1000 0100	0001 0000 0000 0000 0000 000

显然,我们可以看到 a 与 b 同号,且阶码相同,直接将尾数相加得到 c 的表示:

	m_s (数符)	E (阶码)	M (尾数)
c	0	1000 0100	0110 0000 0000 0000 0000 000

那么 $[c]_2 = 0100\ 0010\ 0011\ 0000\ 0000\ 0000\ 0000\ 0000 = 0x42\ 30\ 00\ 00$ ，故选 D

12. **[C]** Cache 共 8 块、采用 2 路组相联，因此共分为 4 组，组号为 0, 1, 2, 3。每个主存块大小为 4B，因此第 58 号单元位于第 $58/4=14$ 块主存块中， $14\%4=2$ ，因此将映射到 Cache 第 2 组的任一字块当中。
13. **[D]** 计算机区分指令和数据有以下 2 种方法：
 - ①通过不同的时间段来区分指令和数据，即在取指令阶段（或取指微程序）取出的为指令，在执行指令阶段（或相应微程序）取出的即为数据。
 - ②通过地址来源区分，由 PC 提供存储单元地址的取出的是指令，由指令地址码部分提供存储单元地址的取出的是操作数。

通常完成一条指令可分为取指阶段、分析阶段和执行阶段。在取指阶段通过访问存储器可将指令取出；在执行阶段通过访问存储器可将操作数取出。这样，虽然指令和数据都是以 0、1 代码形式存在存储器中，但 CPU 可以判断出在取指阶段访存取出的 0、1 代码是指令；在执行阶段访存取出的 0、1 代码是数据。
14. **[D]**
15. **[B]** 在执行 A 指令时，PC 已经完成了自增操作，存储的是下一条指令的地址，因此当执行 A 指令发生中断时，PC 保存的地址为 A 的下一条指令 B 的地址 1001。
16. **[B]** 芯片容量为 2048×8 位，说明芯片容量为 2048B，且以字节为单位存取，即地址线 $\log_2(2048)=11$ 根。8 位则说明数据线要 8 根，加上 1 根片选线和 1 根读控制线 1 根写控制线，引脚数最小为 $11+8+1+1+1=22$ 根
注意：读/写控制线也可以共用一根，题目中没有特殊说明按两根算。
17. **[A]** 程序查询方式、程序中断方式、DMA 方式、通道方式对 CPU 的干扰依次递减。
18. **[暂无]**

二、 填空

1. **(顺序存储) (链式存储)**
2. **($q \rightarrow next = p \rightarrow next$) ($p \rightarrow next = q$)**
3. **($O(n^2)$) ($O(n^2)$)** 简单选择排序的元素间比较次数与序列的初始状态无关，因此时间复杂度始终是 $O(n^2)$ 。
4. **(10, 5, 20, 15, 30) (5, 10, 15, 20, 39)**
5. **(16, 20, 50, 80, 60)**
6. **(1023)** 若满二叉树的深度为 n ，那么总结点数为 $2^n - 1$
7. **(401)** n 个结点的二叉链表共有 $2n$ 个链域，除了根节点以外，其他每个节点都被一个链域所指向，因此用到的链域为 $n-1$ 个，即空链域个数为： $2n - (n-1) = n+1$ 个
8. **($2n+1$) ($2n+2$) (201) (49)**
9. **(100111) (true)** $\&$ 是位操作，操作数逐位进行“与”操作； $\&\&$ 是逻辑运算“与”，结果有编译器决定，一般认为 false 为 0，true 为 1 或非 0；
 $A=44 = 10\ 1100\ B=52=11\ 0100\ A\&B = 10\ 0111\ A\&\&B = true$
10. **(0) (2)** 采用手算最长相等前后缀长度的方法。
‘h’前后缀为空，最长相等前后缀长度=0；
‘he’的前缀{h}∩后缀{e}= ∅，最长相等前后缀长度=0；
‘heh’的前缀{h,he}∩后缀{eh,h}= {h}，最长相等前后缀长度=1；

‘hehe’的前缀{h,he,heh} ∩ 后缀{ehe,he,e}={he}, 最长相等前后缀长度=1;

‘heheh’的前缀{h,he,heh,hehe} ∩ 后缀{eh,eh,eh,h}={heh}, 最长相等前后缀长度=1;

‘hehehe’的前缀{h,he,heh,hehe,heheh} ∩ 后缀{eh,eh,eh,eh,e}={he,heh}, 最长相等前后缀长度=2;

‘hehehel’的前缀{h,he,heh,hehe,heheh,hehehe} ∩ 后缀{eh,eh,eh,eh,eh,e}={}, 最长相等前后缀长度=0;

‘hehehell’的前缀{h,he,heh,hehe,heheh,hehehe,hehehel} ∩ 后缀{eh,eh,eh,eh,eh,eh,e}={}, 最长相等前后缀长度=0;

‘hehehello’的前缀{h,he,heh,hehe,heheh,hehehe,hehehel,hehehell} ∩ 后缀{eh,eh,eh,eh,eh,eh,eh,e}={}, 最长相等前后缀长度=0;

编号	0	1	2	3	4	5	6	7	8
S	h	e	h	e	h	e	l	l	o
PM	0	0	1	1	1	2	0	0	0
next[i]	-1	0	0	1	1	1	2	0	0
nextval[i]	-1	0	-1	0	-1	0	2	0	0

11. **(取指) (执行) (100004)** 如果 k 段流水线中的各段执行时间均为 t, 那么执行 n 条指令的时间就是执行第一条指令的时间 kt, 加上其余 n-1 条指令各执行一段的时间 (n-1)t, 即 kt + (n-1)t。
12. **(速度快) (设计复杂、难以修改) (速度慢) (RISC) (CISC)**
13. **(4) (60) (16) (80)**

Ad	wait	Data
通过总线传输地址	准备数据	通过总线传输数据

对于异步 DRAM 芯片, 访问时间就是从行地址到达行地址引脚的时间起截至到数据被传输到数据引脚的时间段, 那么 DRAM 访问周期就是准备数据的这一时间段。

传送地址: 1 个总线周期=4 个时钟周期

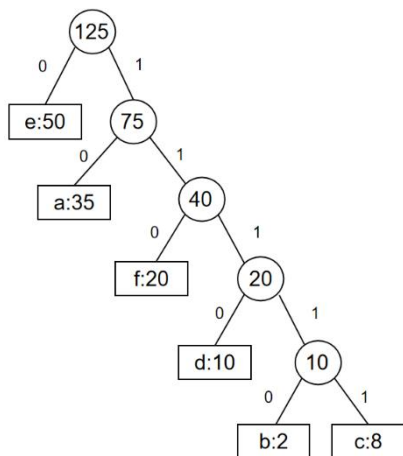
准备数据: DRAM 访问周期=60 个时钟周期

传输地址: 数据传输一次 4 个字, 一个字需一个总线周期, 4×32÷32=4 总线周期=16 时钟周期

总共花费: 4+60+16=80 个时钟周期

三、应用

1. (1)



$$\text{总长度} = 50 \times 1 + 35 \times 2 + 20 \times 3 + 10 \times 4 + (2 + 8) \times 5 = 270$$

(2) 证明：因为哈夫曼树只有度为 0 和 2 的节点，固设度为 2 的节点（分支节点）数量为 a ，由题意知度为 0 的节点（叶子节点）数为 n ；

那么，节点总数 $\text{sum} = a + n$

又因为在树中除了根节点，每一个节点都有一个父节点，对应一个度

总度数 = 总节点数 - 1

$2 \cdot a = \text{sum} - 1 = a + n - 1$ 得到 $a = n - 1$

即度分支节点数等于叶子节点数 - 1

2. (1) 链地址法

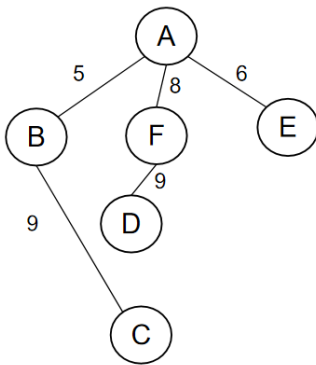
0	1	2	3	4	5	6	7	8	9	10	11
12	25			16			7	8	21	34	23
							19	32			

(2) 开放定址法

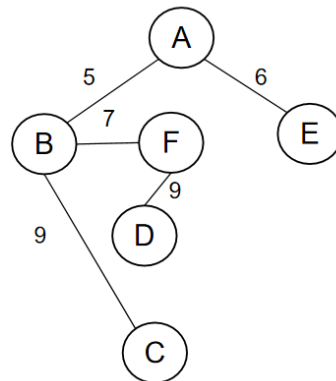
0	1	2	3	4	5	6	7	8	9	10	11
12	25	32	23	16			7	8	19	21	34
1	1	7	5	1			1	1	3	2	2

平均查找长度为： $(1 \times 5 + 2 \times 2 + 3 + 5 + 7) \div 10 = 2.4$

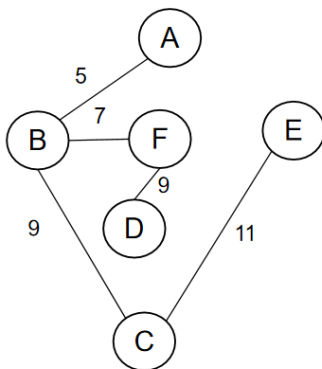
3. (1) 广度优先遍历树



深度优先遍历树



(2) 最小生成树



(3) 拓扑排序

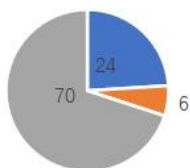
1. ABCDE
2. ABDCE
3. ACBDE

4. (1) 3.82

根据题目所述,只有 Load 与 Store 指令可以访存,显然这是一种 RISC 结构的指令集。为方便计算,假设共有 100 条指令需要执行,所求有效 CPI 即所有指令执行完毕所需总时钟周期数与指令总条数的比值,其中前者需要考虑数据存取缺失所额外产生的代价。

据题知,这 100 条指令中,有 30 条内存访问指令(由 24 条 Load 指令和 6 条 Store 指令组成),70 条非内存访问指令。指令的执行一般可以分为取指 IF、译码 ID、执行 EX、访存 MEM、写回 WB,对于非内存访问指令而言,它不需要访问内存,因此不会在 MEM 阶段产生数据存取缺失,因此只需要考虑在取指阶段发生的取数据缺失;而相比非内存访问指令,内存访问指令还需在 MEM 阶段访问内存,因此计算有效 CPI 时还要考虑在此阶段可能产生的数据存取缺失。

各指令数目



■ Load 指令 ■ Store 指令 ■ 非内存访问指令

设执行这 100 条指令所需的总周期数为 N , 则 $N = n_1 + n_2$, 其中 n_1 为理想条件下执行 100 条指令所需周期数, 即不考虑内存缺失, $n_1 = 100 \times 1.8 = 180$; 而 n_2 为执行这 100 条指令所产生的全部缺失代价, 可以分为三类: 一是非内存访问指令在取指阶段的指令 Cache 未命中所产生的缺失 I_1 ; 二是内存访问指令在取指阶段的指令 Cache 未命中所产生的缺失 I_2 ; 三是内存访问指令在访存阶段数据 Cache 未命中所产生的缺失 I_3 。同时由于数据存取的缺失代价是不同的, 内存访问指令中的 Load 指令(取数)与 Store 指令(存数)所产生的数据缺失代价也是不同的, 所以也应分开讨论。 $n_2 = I_1 + I_2 + I_3$ 。

I_1 : 非内存访问指令共 70 条, 指令 Cache 命中率 95%, 取指令缺失代价 20 个时钟周期, 故 $I_1 = 70 \times 5\% \times 20 = 70$;

I_2 : 内存访问指令共 30 条内存访问指令, 指令 Cache 命中率 95%, 取指令缺失代价 20 个时钟周期, 故 $I_2 = 30 \times 5\% \times 20 = 30$; I_1 与 I_2 也可合一计算, 因为所有指令都有可能因指令 Cache 未命中产生数据缺失;

I_3 : 30 条内存访问指令, 24 条 Load 指令和 6 条 Store 指令, 数据 Cache 命中率 80%。取数据缺失代价 20 个时钟周期, 存数据缺失代价 5 个时钟周期, 故 $I_3 = 24 \times 20\% \times 20 + 6 \times 20\% \times 5 = 96 + 6 = 102$;

故 $n_2 = I_1 + I_2 + I_3 = 70 + 30 + 102 = 202$; $N = n_1 + n_2 = 180 + 202 = 382$; $CPI = 3.82$ 。

不考虑缺失的总周期数 n_1	$100 \times 1.8 = 180$	
总缺失周期数 n_2	非内存访问指令取指令缺失周期数 I_1	$70 \times 5\% \times 20 = 70$
	内存访问指令取指令缺失周期数 I_2	$30 \times 5\% \times 20 = 30$
	内存访问指令取数据与存数据缺失周期数 I_3	$24 \times 20\% \times 20 + 6 \times 20\% \times 5 = 96 + 6 = 102$
总执行时钟周期数	$70 + 30 + 102 + 180 = 382$	

(2) 提高命中率：适当增加 Cache 块的大小、增加 Cache 容量；

降低缺失代价：设置多级 Cache。

(第四大题答案及解析由“xc”完成)

5. 由题得盘面数： $2 \times 4 = 8$ ，转速： $6000 \div 60 = 100\text{rpm}$ ，一个时钟周期为： $\frac{1}{2 \times 10^9} = 5 \times 10^{-7}\text{ms}$ 。

① 格式化容量 $= 8 \times 6000 \times 400 \times 512\text{kb} = 1200000\text{MB} = 1.14440918...T \approx 1.14T$

② 平均访问时间 = 平均寻道时间 + 平均旋转时间

$$= 8\text{ms} + 1 \div 100 \times 50\% = 13\text{ms}$$

③ 平均传输速率 $= 100 \times 400 \times 512\text{kb} = 20480000\text{kb/s} = 2.44140625...GB/s \approx 2.44GB/s$

④ 连续读取时间 = 平均访问时间 + 读取 8 个扇区的时间

$$= 13\text{ms} + 8 \times 1 \div 100 \div 400 \times 1000\text{ms} = 13.2\text{ms}$$

⑤ 随机读取时间 $= 8 \times (\text{平均访问时间} + \text{读取 1 个扇区的时间})$

$$= 8 \times (13 + 1 \div 100 \div 400 \times 1000) = 8 \times (13\text{ms} + 0.025\text{ms}) = 104.2\text{ms}$$

⑥ 磁盘转速为 100r/s 每秒读取 $100 \times 400 = 40000$ 个扇区，DMA 方式每次读取 8 个扇区，那么 1s 内要做 $40000 \div 8 = 5000$ 次 I/O 操作，即每 $1 \div 5000 = 0.2\text{ms}$ 处理一次 I/O 请求。

处理一次需要 $1000 \times 5 \times 10^{-7}\text{ms} = 5 \times 10^{-4}\text{ms}$ ，占用比例为 $5 \times 10^{-4} / 0.2 = 0.0025 = 0.25\%$

(或用总时间计算： $5000 \times 5 \times 10^{-4}\text{ms} / 1000\text{ms} = 0.25\%$)

6. (1) 读取寄存器在 ID 阶段



L3L4 在 ID 阶段读取寄存器时结果仍未写回，故 L1 与 L3、L2 与 L3、L3 与 L4 存在数据冲突。

(2) 在 add 指令的情况下，完成指令需要 4 个周期：取指-译码-执行-写回，增加一个空的访存周期，使流水线更加规整，所以访存周期没有执行操作。

所以，add 指令执行的具体情况：

取指：使用 PC 中的地址从存储器读取指令，然后将指令放入 IF/ID (题目中为 FBUF) 寄存器。

指令译码与寄存器堆的读取：把读出的寄存器的内容存入 ID/EX (题目中为 DBUF) 寄存器。

执行或地址计算：将计算结果存入 EX/MEM (题目中为 EBUF) 寄存器。

存储器访问：未执行任何操作，将 EX/MEM (题目中为 EBUF) 寄存器的值传到 MEM/WB (题目中为 MBUF) 寄存器。

写回：从 MEM/WB (题目中为 MBUF) 寄存器中读取数据并写回寄存器堆。

量化为表格，更加清晰直观：

	FBUF	DBUF	EBUF	MBUF
1 (取指)	空	空	空	空
2 (译码)	ADD 指令	空	空	空
3 (执行)	空	寄存器中取出的内容	空	空
4 (访存)	空	空	计算的结果	空
5 (写回)	空	空	空	计算的结果

(第六大题答案及解析由“LYH”完成)

2019

一、选择

1. **[A]**很简单的问题，不再赘述，绿书 P4 第 2 行。
2. **[B]**按照公式，CPU 执行时间 = $\frac{(\text{指令条数} \times \text{CPI})}{\text{主频}}$ ，则 $\text{CPI} = \frac{(\text{CPU 执行时间} \times \text{主频})}{\text{指令条数}} = \frac{(10 \times 2.5\text{G})}{1.25 \times 10^{10}} = 2$ ，按照公式 $\text{MIPS} = \frac{\text{主频}}{\text{CPI}} = \frac{2.5\text{G}}{2} = \frac{2.5 \times 10^3 \text{M}}{2} = 1.25 \times 10^3$
注：在性能相关的公式计算中，主频单位常为 MHz。
3. **[B]**绿书 P55、56 相关内容；慕课视频 2.4.8-4:26 处。
4. **[A]**慕课视频 2.2.2-2:51 处；绿书 P45 第六行。
5. **[B]**绿书 P51 第 10 行、第 24 行。也可以通过命名来区分，变址变，基址定。
6. **[C]**按照 IEEE754 标准，阶码 0、尾数非 0 表示规格化数。不要忽视 IEEE754 而直接按照普通的规格化数来分析。（群友：21-计科- 纠正）
7. **[B]**01 正溢，10 负溢。
8. **[B]**绿书 P134 第 4 行-第 7 行；慕课视频 3.4.2 - 2:31 处。
9. **[B]**绿书 P211 前 3 行。 $N = \lceil \log_2 64 + 1 \rceil = 7$ 。
10. **[D]**字母及其符号编码。BCD 仅编码数字；奇偶校验和汉明码为校验码。
11. **[A]**绿书 P237 第 15、16 行。
12. **[D]**DRAM 地址引脚复用，要除以 2，绿书 P243 第 10 行。 $8 + 24/2 = 20$ 。
13. **[C]**爱课程 PPT5.9 第 10 页。TLB 命中，页表一定命中，信息就在主存上，但不一定在 Cache 中。
14. **[C]**寻道时间与转速无关；存取速度取决于存取时间，存取时间由寻道时间+等待时间(查扇)+传输时间决定，和转速无单纯的线性关系。平均等待时间 = $1/2r$ ， r 转速。
15. **[B]**页面大小 1KB，则页内偏移是 10 位；虚拟地址空间 4GB，则地址为 32 位，构成是[虚页号(22)，页内偏移(10)]。虚拟地址 038F7F80H = 0000 0011 1000 1111 0111 1111 1000 0000，取高 22 位为 0000 0011 1000 1111 0111 11 = 0E3DFH，查页表标记发现存在，有效位 1，则命中，拿到页框号 00245H，与虚址的低 10 位拼接，即可得到最终实地址 0000 0000 0010 0100 0101 11 1000 0000 = 0091780H。
16. **[C]争议** 条带化就是 RAID0，不具备任何可靠性；磁盘镜像、奇偶校验都是提高可靠的方法；对于汉明校验，个人认为是可以的，爱课程 PPT6.8 第 14 页也谈到可以实现纠错一位检测两位的功能（起码是强过奇偶校验的）。尽管成本很大，且 RAID2 也不再使用（这也是很多人认为不选的一个原因），但不能忽视其提高可靠性的本质。
17. **[D]**通道是专门负责输入输出的处理机，与 CPU 并行工作，对 CPU 干扰最少，爱课程 PPT6.7 第 27 页。程序直接控制和程序直接中断都是以 CPU 为中心的 I/O 工作方式，对 CPU 干扰最多，尤其程序直接控制，与 CPU 串行。DMA 仅在开始结束时干扰 CPU。
18. **[B]**DMA 周期窃取是在执行访存指令中插入的，而中断请求是在指令周期结束时进行的，故不是一种中断请求（DMA 仅在开始结束时进行中断请求）；两数相加为 0 显然不产生中断；结果

溢出如果是下溢出当 0 处理，不中断（异常）。

19. **[A]** 尽管资料中没有明确的出处说明这一点，但显然统一编址的时候，主存的部分地址空间划给了 I/O，那么分辨 I/O 还是主存就主要看地址是哪的，所以仅需要区分地址码即可。
20. **[A]** 中断最重要的就是正确进入中断服务程序和正确返回断点处。

二、填空

1. **(控制器) (运算器) (存储器) (存储程序) (指令)**
前三空显然没什么问题；第四空，按照后面文字描述，可知是在解释“存储程序”原理，故可知答案；第五空，可以从爱课程 PPT1.2 第 8 页的最后两行推知，填“指令”。
2. **(机器数) (真值)。**
3. **(00110000) (10001111)** \wedge 是与操作， \oplus 是异或操作。
4. **(C0880000H) (C011000000000000H)** $-4.25 = -100.01 = -1.0001 \times 2^2$
第一问，根据 IEEE754 单精度的定义，[符号 s，阶码 e，尾数 f]。符号 s 为 1；阶码 e 加上偏置 127 是 129，即 10000001；尾数去掉隐含 1 后是 000100000000000000000000。则最终的 IEEE754 值是，1100 0000 1000 1000 0000 0000 0000 0000 = C0880000H。
第二问，根据 IEEE754 双精度的定义，[符号 s，阶码 e，尾数 f]。符号 s 为 1；阶码 e 加上偏置 1023 是 1025，即 10000000001；尾数去掉隐含 1 后是 000100.....(共 52 位)。则最终的 IEEE754 值是 1100 0000 0001 0001 00..... = C011000000000000H。
5. **(9) (8256)** 没找到相关出处，就参考了 2014 年 408 真题，去掉公共的 2 条微指令，每条指令是 4 条微指令，64 条指令就是 $64 \times 4 + 2 = 258$ 条微指令，需要 $\lceil \log_2 258 \rceil = 9$ 位。每条微指令 32 位，258 条就是 $258 \times 32 = 8256$ 位。
6. **(13) (5) (6) (128)** 主存空间为 16MB，按字节 B 编址，则 $\frac{2^{24}B}{B} \rightarrow 24$ ，地址长度为 24 位。Cache 块大小为 64B，即 2^6B ，块内偏移字段是 6 位。Cache 数据区总计 4KB，则 Cache 共有 $\frac{2^{12}B}{2^6B} = 2^6$ 行（块），按照二路组相联除以 2，则有 2^5 组，则索引字段为 5 位。 $24 - 6 - 5 = 13$ ，则标识字段为 13 位。Cache 分为两部分，标记阵列和数据阵列。一般标记阵列主要以主存标识字段和一位有效位构成，按照不同的要求还会有其他位。题目中谈到采用了 LRU 置换算法和写回策略，则应当有 $\log_2 2 = 1$ 位替换位和 1 位脏位。则本题的标记共是 $13 + 1 + 1 + 1 = 16$ 位。每行（块）Cache 对应一组标记，一共 2^6 行（块），则 $2^6 \times 16 \div 8 = 16 \times 8 = 128$ 字节，可知标记阵列共 128 字节。
7. **(30) (32)** 主存地址空间 1GB，即 $2^{30}B$ ，则 MAR 为 30 位。CPU 一次性读取最大 32 位，即机器字长，与 MDR 同宽，则 MDR 是 32 位。
8. **(集中式) (分散式)** 按理说应该还有第三问填透明式，但既然没给空，就优先填前两个。绿书 P276 小标题。
9. **(II) (III) (VII) (VI) (X)** 中断服务程序内的部分，自然从保护现场开始；由于是单级中断，自然不需要把隐指令阶段关上的中断打开，也不需要设置新屏蔽字，直接执行中断服务；执行结束后，由于之前并没有开中断也没有设置新屏蔽字，所以此处不关中断也不恢复屏蔽字，直接恢复现场；然后打开隐指令阶段关上的中断；然后返回。主体流程参考绿书 P340 图 6.15。
10. **(只读型) (追记型) (可擦写型)** 绿书 P298 小标题。

三、简答

1. 提高运算速度，可以使用主频高的 CPU、增加 CPU 寄存器数目、采取并行结构；提高存储速

度，可以使用高速器件、多体交叉编址、调整存储层次、使用高速缓存 Cache。

2. 结构冒险、数据冒险、分支冒险

结构冒险解决方法：插入气泡；修改指令使其延缓一个周期执行或插入 NOP。

数据冒险解决方法：插入气泡；插入 NOP；转发技术；编译优化指令顺序。

分支冒险解决方法：插入气泡；插入 NOP；分支预测；延迟分支。

3. 除数为 0：译码阶段 ID

结果溢出：执行阶段 EXE

无效指令地址：取指阶段 IF

缺页：取指阶段或访存阶段 IF、MEM

外部中断：写回阶段 WB

爱课程 PPT4.7 第 28 第 29 页（本题可以从这些操作的原理入手。除数为 0 肯定是在除之前发现到的，如果是在执行时检测则就太迟了，会造成系统宕机，故应当在译码的时候就检测；结果溢出必然是执行后才发现溢出；无效指令地址则是取指时期；缺页可以是指令访问主存访存时期，也可以是取指时期；外部中断则应该在指令执行的最后阶段，也就是写回时期。）

4. 使多块盘并行工作空间整合，提高传输速度，提高可靠性。

条带化：把数据分散到多个硬盘

大条区交叉分布减少了 IO 排队时间，具有较快的 IO 响应能力。

小条区交叉分布可获得较高的数据传输率。

爱课程 PPT6.8 第 12 页

5. 见下表，爱课程 PPT6.6 第 24 页。

	中断方式	DMA
数据传输	程序控制	硬件控制
响应时间	指令执行结束	存取周期结束
处理异常	能	不能
中断请求	传送数据	结束处理
优先级	低	高
I/O 设备	低速设备	高速设备

四、应用

1. 参考答案如下

(1) 8 位机器码，补码表示，即 1 位符号位，7 位数值位。

$$[x]_{\text{补}} = 0111111, [y]_{\text{补}} = (2^9 - 128) \text{MOD} 2^8 = 100000000 - 10000000 = 10000000$$

$$[-y]_{\text{补}} = (2^9 + 128) \text{MOD} 2^8 = 100000000 + 10000000 = 100000000$$

$$[\text{sum}]_{\text{补}} = [x + y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}} = 01111111 + 10000000 = 11111111$$

$$-[\text{sum}]_{\text{原}} = 2^9 - 11111111 = 100000000 - 11111111 = 00000001 = 1 \text{ (十进制)}$$

$\text{sum} = -1 \text{ (十进制)}$, $x + y = 127 - 128 = -1$, 无溢出

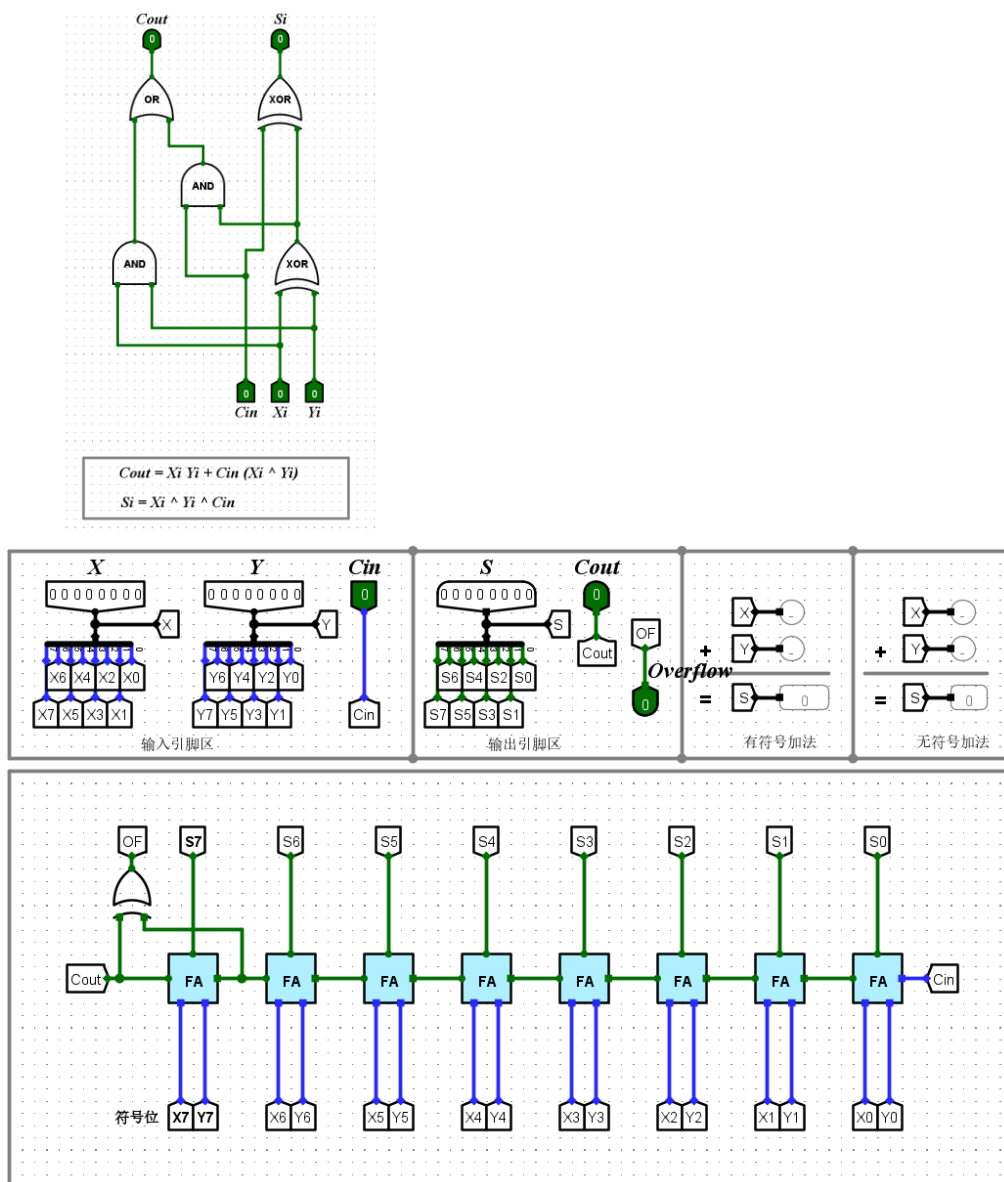
$$[\text{sub}]_{\text{补}} = [x - y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}} = 01111111 + 10000000 = 11111111$$

$$-[\text{sub}]_{\text{原}} = 2^9 - 11111111 = 100000000 - 11111111 = 00000001 = 1 \text{ (十进制)}$$

$\text{sub} = -1 \text{ (十进制)}$, $x - y = 127 + 128 = 255$, 溢出

(2) 上述溢出结果，其计算得到的正数已经超过了能表示的最大正数范围，属于正溢出，需要产生异常（内中断）。

(3) 逻辑图如下，分别为 1 位全加器和 8 位加法器



FA 单元即为 1 位全加器。

2. CSAPP 的 P282-P287 建议仔细阅读体会。

- (1) 要达到最大吞吐率的流水线设计，一个是要让流水段尽量规整；另一个，由于流水线寄存器的延迟达到了 20ps，是一个比较大的开销，所以也要让流水线寄存器尽可能多的低。给出的这些模块中，F 是 70ps 最长的段，所以应当以 70 的规模来划分。A、B+C、D+E、F，得到的段延迟是 60ps、70ps、55ps、70ps，根据划分插入流水段寄存器，得到段延迟是 80ps、90ps、75ps、90ps。则最大延迟为 $90ps \times 4 = 360ps$ 。

60ps	20ps	70ps	20ps	55ps	20ps	70ps	20ps
------	------	------	------	------	------	------	------

$$\text{吞吐率} = \frac{1}{90ps} \times \frac{1000ps}{1ns} \approx 11.1GIPS$$

- (2) 对于高级语言 $x=x+y$ ，需要用到三类指令都已经下面表格给出，分别由四条指令构成：取 x 、取 y 、相加、存回 x ，那么可以画出其流水线图

取 x	IF	ID	EX	MEM	WB			
取 y		IF	ID	EX	MEM	WB		
相加			IF	ID	EX	MEM	WB	
存回 x				IF	ID	EX	MEM	WB

但显然，对于“相加指令”的 EX 段，由于此时“取 y 指令”的 MEM 还未完成，此时拿不到 y 来相加，故构成了数据冒险。同样的，对于“存回 x 指令”的 EX 段，由于“相加指令”的 WB 段未完成，结果还未写回寄存器中，所以此时存回 EX 阶段也拿不到结果，故也发生了数据冒险。应当进行冒险的消除。由于题目要求未采用转发技术，所以只能进行硬件阻塞，通过插入气泡来延缓流水段若干个周期。结果如下

取 x	IF	ID	EX	MEM	WB									
取 y		IF	ID	EX	MEM	WB								
相加			IF				ID	EX	MEM	WB				
存回 x							IF				ID	EX	MEM	WB

一共需要 14 个时钟周期。此题与 2012 年 408 真题高度相似。

3. 参考答案如下

(1) 8 行 $256B/32B = 8$ 。

$s=64$ 时，访问序列为 $a[0]$ 、 $a[64]$ 、 $a[0]$，由于块是 32B，int 是 4B，所以一块只能放 0—7，第二块放 8-15.....。 $a[0]$ 所在是第 0 行， $a[64]$ 所在也是第 0 行，第一次必然不命中，其余都一直冲突，所以缺失率是 100%。 $s=63$ 时，访问序列为 $a[0]$ 、 $a[63]$ 、 $a[126]$， $a[0]$ 在第 0 行， $a[63]$ 在第 7 行， $a[126]$ 在第 7 行。第一次 $a[0]$ 不命中，其后每次都命中，第一次 $a[63]$ 不命中，其后每次的 $a[63]$ 和 $a[126]$ 都冲突，缺失率是 $2001/3000 \approx 66.7\%$ 。

(2) 4 组 $8/2 = 4$ 。由于是组相联，会在行内不冲突的映射。所以 $s=64$ 时，仅头两次缺失，之后全部命中，缺失率 $2/2000 = 0.1\%$ 。 $s=63$ 时，仅头三次缺失，之后全部命中，缺失率 $3/3000 = 0.1\%$ 。

4. 参考答案如下

(1) 91.55GB 由于并没有给出位密度，所以这里的磁盘容量只能是格式化后的容量。容量为 $= 8 \times 40000 \times 600 \times 512 \div 1024 \div 1024 \div 1024 \approx 91.55GB$ 。

(2) 8ms 不考虑数据传输时间，则平均存取时间只剩下了 = 平均等待时间 + 寻道时间 $= \frac{1}{2r} + 5ms = 0.5 \times \frac{60}{10000} s + 5ms = 3ms + 5ms = 8ms$ 。

(3) [台号，柱面号(16 位)，盘面号(3 位)，扇区号(10 位)] 600 个扇区： $\lceil \log_2 600 \rceil = 10$ 位；8 个磁头意味着 8 个盘面，3 位(计算方法同上)；40000 个磁道即 40000 个柱面，16 位(计算方法同上)。台号没有信息给出无法得知。

(4) 1.25% 算 I/O 处理时间占整个处理器时间的百分比，就是要看 1 秒内进行了多少次 I/O 操作，以此得到 I/O 所花费的总时钟周期。每磁道有 600 扇区，一次 DMA 传送了 8 个扇区，也就是磁头移动了整个磁道的 $8/600$ ，而磁头转一圈需要 $60/10000s$ ，那么传输 8 个扇区所需要的时间就是 $\frac{8}{600} \times \frac{60}{10000} = 0.08ms$ ，那么一秒进行了 $\frac{1000}{0.08} = 12500$ 次 I/O 操作，也就是花费了 12500000 个时钟周期，那么占比为 $\frac{12500000}{1G} = 1.25\%$ 。

2018

一、选择

1. **[D]**爱课程 PPT1.4 第 13 页。
2. **[C]**首先根据各类型指令的条数可算出各类型指令的占比，整数运算 $\frac{45000}{100000} = 0.45$ ，数据传送 $\frac{32000}{100000} = 0.32$ ，浮点运算 $\frac{15000}{100000} = 0.15$ ，控制传送 $\frac{8000}{100000} = 0.08$ 。平均 $CPI = 1 \times 0.45 + 2 \times 0.32 + 3 \times 0.15 + 2 \times 0.08 = 1.7$ 。则可以首先算出 $MIPS = \frac{\text{主频}}{CPI} = \frac{400}{1.7} \approx 235$ ，可直接选出答案 C。当然，程序执行时间 = $\frac{\text{程序指令总条数} \times CPI}{\text{主频}} = \frac{1.7 \times 100000}{400M} = 425\mu s$ 。注意，算 MIPS 时，主频默认单位为 MHz 且不写出。
3. **[D]**可结合慕课视频 1.4.3-4:42 处理解。
4. **[A]**显然这是个简单题，自行计算即可。2FH=47D，63O=51D，10010110B=150D，10010110BCD=96D。
5. **[B]**要使 $x > -\frac{1}{2}$ ，就首先不能让 x_1 为 1，否则 2^{-1} 已然不满足条件，排除 CD。对于剩下的 $x_2x_3x_4$ 来说，即便全 1，也不会比 2^{-1} 大，自然也就比 $-\frac{1}{2}$ 大。
6. **[C]**IEEE754 下的 C510000H=1100 0101 0001 0000 0000 0000 0000 0000，符号位为 1，负值。阶码为 10001010B=138D，去掉 127 的偏置为 11。尾数部分是 00100……，加上隐含 1 为 1.0010……，也即 1.125。最终真值为 -1.125×2^{11} 。
7. **[D]**对阶操作必然不会引起阶码的溢出。其余参考绿书 P145 开始的内容。
8. **[A]**爱课程 PPT1.2 第 14 页。
9. **[C]**小端存储，即数据的最低位为地址起始处。则 1000H 单元处的 23H 为低位数据，1001H 单元处的 F7H 为高位数据，原数据为 1111 0111 0010 0011B，恢复成原码是 1000 1000 1101 1101B=-2296D。同理 1002H 单元处开始的数据为 -21710D。
10. **[D]**显然是 D。慕课视频 4.2.1-3:05 处。
11. **[A]**变址寄存器内容是 0400H，形式地址是 5200H，所得实际地址是 5600H，即所取内容在 5600H 处，即 1200H。
12. **[B]**爱课程 PPT4.10 第 16 页。
13. **[D]**Load-Use 指令所产生的数据在访存阶段被取出，是无法通过转发发给下一条流水的。慕课视频 4.4.3-7:46 处。至少仅用 NOP 就可以消除所有指令。由于 Load-Use 指令产生的数据冒险无法通过转发消除，所以只能用 NOP 阻塞一个时钟周期。前面的分支指令不会产生数据以供后面的 ALU 指令使用，没什么问题。
14. **[A]**半导体存储器分为随机存储器、只读存储器、串行存储器，串行存储器是不能随机存取方式读写的。此点来源于网络百科。其次，ROM 作为半导体存储器，可以随机读，但不能随机写。绿书 P247 第 11 行。
15. **[B]**主存空间 1024MB = 2^{30} B，按字节编址，则是 2^{30} 个主存单元，则 MAR 是 30 位。一次读写操作最多 32 位，即是机器字长=存储字长，也即 MDR 位数。不要将编址位数和 MDR 位数搞混。

16. **[C]**Cache 失效后, 需要将主存一个块的内容载入到 Cache。存取和总线的宽度是 8B, 一共需要进行 8 次 8B 传输操作, 每次传送 8B 要配套一次送地址和一次读命令合计 2 个时钟周期; 以及配套 8 个时钟周期的准备, 共计 10 个周期。8 次就是 80 个时钟周期。
17. **[C]**外部中断, 主机 (CPU、主存构成主机) 以外的部件引起的中断。绿书 P332 第 27 行, 显然是 C。
18. **[A]**链式查询, 也即菊花链仲裁, 对电路故障很敏感。爱课程 PPT7.1 第 23 页。
19. **[D]**全串行模式下的 DMA 要接管对地址总线、数据总线及相关控制线的使用权, 即接管控制系统总线。绿书 P351 第 20 行。
20. **[A]**每个事务要 5 次 I/O 操作, 每次 I/O 操作要 10000 条指令, 则每个事务要 50000 条指令。处理器每秒执行 5 千万条指令, $50000000/50000 = 1000$, 从指令条数的角度来看, 每秒能最多执行 1000 个事务。但从 I/O 操作角度来看, 每秒最多完成 1000 次 I/O 操作, 也即 $1000/5 = 200$ 次事务。综合下来每秒钟处理事务最多 200。

二、填空

1. **(01000101) (00100110) (0)** 无符号数, $69=64+4+1=01000101$, $38=32+4+2=00100110$, 这种方法也是较为方便的十进制转二进制方法, 不要傻傻的除基取余算。不产生进位, 进位 0。
2. **(FFFFFBC0H) (-1088) (FFFFBC00H) (FFFFFFBCH) (0FFFFFFBCH)** 第一空将二进制写成十六进制数无需多言, 但注意不要把末尾的 H 忘掉。int 是 32 位的, 慕课视频 2.4.1-3:50 处。所以截取低 32 位是 1111 1011 1100 0000B, 且 C 语言中是补码表示, 则原码为 1000 0100 0100 0000B=-1088D。对其算术左右移的话, 左移补 0 右移补符。逻辑移位只补 0。
3. **(双稳态触发器) (栅极电容) (DRAM)** 爱课程 PPT5.2 第 6 页。后一问众所周知。
4. **(16) (9) (7)** 主存空间 4GB, 字节编址, 即 $2^{32}B$, 主存地址是 32 位。块大小是 128B, 即 2^7B , 块内偏移 7 位。Cache 数据区 64KB, $64KB/128B = 2^9$, 则索引字段 9 位, 剩余 16 位为主存标记字段。
5. **(主存) (Cache) (辅存)** 慕课视频 5.1.1-3:36 处。
6. **(64GB) (2^{24})** 虚址 36 位, 且字节编址, 则虚存空间大小为 $2^{36}B = 64GB$, 页面大小 4KB, 最多可有 $2^{36}/2^{12} = 2^{24}$ 页。
7. **(快表/TLB)**
8. **(总线) (外设) (主存)** 结合绿书 P351 第 20 行, 第一空倾向于填总线。绿书 P350 第 11 行。
9. **(请求) (允许响应) (结束)** 绿书 P337 第 4 行。
10. **(40) (2000) (1.67GB) (1.22GB)** 直径差是 100mm, 则最内最外磁道相差 50mm, 当中容纳了 2000 磁道, 道密度为 $\frac{2000}{50} = 40TPM$ 。所有盘面的相同磁道在逻辑上构成了一个柱面, 柱面数即为 2000。非格式化容量要按照位密度计算, $20 \times 2000 \times 45000B \div 1024 \div 1024 \div 1024 \approx 1.67GB$ 。格式化容量按照扇区算, $20 \times 2000 \times 64 \times 512B \div 1024 \div 1024 \div 1024 \approx 1.22GB$ 。

三、简答

1. 程序响应时间是指作业提交到完成所花的时间, 通常包括 CPU 执行时间和等待时间; 吞吐率是单位时间内处理请求的数量。
 - (1) 使用更快的处理器, 意为主频高的处理器, 会缩短程序执行时间, 从而减少程序响应时间。
 - (2) 优化编译器生成的代码使程序执行总时间减少, 缩短响应时间, 提高吞吐率。

2. len 是无符号 `unsigned` 类型，当 len 为 0 时，而具体执行却是 $len-1$ ，即 $11\dots 1$ ，使得循环不断执行，最终发生越界访问，发生存储器访问异常。修改方法：将 len 声明为 `int` 型，或循环测试条件改为 $i < len$ 。
3. 最少访存 0 次。当执行 `store` 指令时首先查找 TLB，然后再查找 Cache。若 Cache 命中，则在 Write Back 策略下，只将对应的 Cache 行修改并标记脏位，即完成全部操作，最好的情况下不访存。
4. 不一样。中断响应优先级是由硬件排队线路或中断查询程序的查询顺序决定的，不能动态改变；而中断处理优先级可以由中断屏蔽字来改变。
5. 总线突发传送是一种在一次总线事务过程中传输多个数据的方式，只要传送一个首地址，后面连续传送多个数据信息。适用于传送多个连续单元的数据。

四、应用

1. $[A]_{\text{补}} = 00.10101$, $[B]_{\text{补}} = 1.00101$, $[-A]_{\text{补}} = 11.01011$ ，过程如下

	部分积	乘数
	00.00000	1.001010
加 $[-A]_{\text{补}}$	+11.01011 11.01011	
右移	11.10101	11.00101
加 $[A]_{\text{补}}$	+00.10101 00.01010	
右移	00.00101	011.0010
加 $[-A]_{\text{补}}$	+11.01011 11.10000	
右移	11.11000	0011.001
加 $[A]_{\text{补}}$	+00.10101 00.01101	
右移	00.00011	10011.00
加 0 右移	00.00011	010011.0
加 $[-A]_{\text{补}}$	+11.01011 11.01110	010011.0

所以 $[A \times B]_{\text{补}} = 11.0111001001$

2. 参考答案如下
 - (1) 编址单位是字节。指令长是双字节，且 PC 自增量是 2。
 - (2) 是带符号比较。因为设有溢出标志 OF，而无符号比较的判断表达式中没有溢出标志 OF。偏移地址 `imm8` 表示相对指令条数，因为偏移量是 $\text{imm8} \times 2$ ，这才是相对地址。转移目标地址范围时 $(-128 \times 2) \sim (127 \times 2)$
3. 参考答案如下
 - (1) $C = A + B$; $D = E - F$; $H = C + D$
 - (2) 采用转发技术也会存在相关性
Load 2#, B 和 Add 3#, 1#, 2# 之间; Load 5#, F 和 Sub 6#, 4#, 5# 之间还存在相关性。因为这属于 Load-Use 数据冒险，无法通过转发技术避免，解决方法时插入空指令等 Load 完成后再 Use。(因为 Load 产生的数据来自四阶段的访主存，从主存写到寄存器，此时无法提前转发)
 - (3) 可将 Load 4#, E 插入到第一组数据相关之间。

Store C, 3#插入到第二组数据相关的指令之间。

修改后如下

```
Load 1#, A
Load 2#, B
Load 4#, E
Add 3#, 1#, 2#
Load 5#, F
Store C, 3#
Sub 6#, 4#, 5#
Store D, 6#
Add 7#, 3#, 6#
Store H, 7#
```

4. 参考答案如下

- (1) 页大小为 $4\text{KB} = 2^{12}\text{B}$ ，虚地址 42 位，则高 30 位为虚页号；低 12 位为页内地址；页表项至少要有 1 位装入位，以表示该页面是否已在主存中。由于虚页号 30 位，则页表项有 2^{30} 个，页表项为 4B，则页表大小是 $2^{30} \times 4\text{B} = 2^{32}\text{B}$ 。
- (2) TLB 采用二路组相联，则 256 项分为 128 个组，组索引需要 7 位，位于虚页号 30 位的低 7 位，高 32 位为标记字段。TLB 总标记大小为 $23\text{bit} \times 256$ 。
- (3) 1GB 物理内存为 2^{30}B 。Cache 块大小为 $64\text{B} = 2^6\text{B}$ ，需要 6 位。Cache 大小为 128KB，则有 $128\text{KB}/64\text{B} = 2\text{K} = 2^{11}$ ，四路组相联 $\frac{2^{11}}{4} = 2^9$ ，需要 9 位组索引。剩余 $30-6-9=15$ 位为标识位。[29...标识位...15, 14...组索引...6, 5...块内地址...0]

5. 参考答案如下

- (1) $8 \times 512 \times 64 \times 1\text{KB} = 256\text{MB}$
- (2) 平均存取时间=平均寻道时间+平均等待时间 $= 8\text{ms} + \frac{1}{2r} = 8\text{ms} + 0.5 \times 60/3600\text{s} \approx 16.33\text{ms}$
- (3) 因为是连续扇道连续磁道，所以有突发传输的特性，5MB 需要 5K 个扇区，80 磁道。仅第一个磁道需要寻道和等待，其他 79 个只需要道间移动，所以用时是 $16.33 + 79 \times 1.5 = 134.83\text{ms}$
- (4) $5\text{MB}/134.83 \approx 37.1\text{MB/s}$

2017

一、选择

1. **[D]**都存储在主存中。
2. **[D]**D 选项是 CISC 指令集的特征，二八原则。RISC 的所有指令使用频度都很高。
3. **[B]**显然选 B。
4. **[C]**00100100B=36D, 43O=35D, 32D=32D, 25H=37D。
5. **[B]**绿书 P84 第 4 行。
6. **[B]**首先排除原码。如果是反码，恢复成原码形式为 11111111，确实是-127。根据-127 的原码形式，可知其补码为 10000001。移码是在补码的基础上反符号得到的，绿书 P91，移码应当是 00000001。
7. **[D]**慕课视频 3.2.1-11:24 处。
8. **[C]**慕课视频 2.3.1-2:25 处。
9. **[C]**慕课视频 2.4.8-3:43 处。
10. **[C]**众所周知。
11. **[A]**爱课程 PPT4.10 第 16 页。
12. **[B]**不用管位扩展上的问题，直接看字扩展方面。1M 为一个整体，主存储器由 16 个 1M 构成。第一个 1M 的地址范围是 000000H-0FFFFFH，第二个 1M 的地址范围是 100000H-1FFFFFH，1C1F00H 在第一个 1M 之内，所在芯片最小地址应该是 100000H。
13. **[A]**
14. **[B]**采取组相联策略均不会冲突失效，则只需考虑直接映射情况。
给出的是存储单元号，按字节 B 编址，而按照块大小 16B，可求出选项中的单元处于哪个块。
A. 3、6 B. 3、19 C. 3、10 D. 2、3
采用直接映射，16 行 Cache，求出 Cache 块号
A. 3、6 B. 3、3 C. 3、10 D. 2、3
(以上编号均从 0 开始)
15. **[B]**Cache 访问情况如下 (*表示命中，_表示替换)

序列	3	5	7	3*	8	20_	7*	11_
行 0	3	3	3	3	3	3	3	11
行 1		5	5	5	5	20	20	20
行 2			7	7	7	7	7	7
行 3					8	8	8	8

 访问 8 次，命中 2 次，命中率 25%。
16. **[D]**通道主要用于大型计算机系统。
17. **[C]**中断的申请时间是随机的，而 CPU 是在统一的时刻即每条指令执行结束时发出中断查询信号，以获取中断请求。因此 CPU 响应中断的时刻是每条指令执行阶段的结束时刻。
18. **[D]**

19. **[D]**每秒 40MB, 每次 64bit=8B, 则每秒需要申请 5M 次中断, 每次中断需要处理 400 周期, 则共需要 $400 \times 5M = 2G$ 个周期, 占 CPU 周期的 100%。
20. **[B]**

二、 填空

1. **(表示范围) (精度)**
2. **(8) (8) (8) (9)** 绿书 P112 第 5 行。
3. **(6) (2) (8)**
4. **(85ns)**
5. **(4) (2) (1)**
6. **(直接映射) (组相联映射) (全相联映射)**
7. **(60%) (88ns)**
8. **(T0: PC->MAR) (T1: MAR->MDR, PC+1->PC) (T2: MDR->IR) (T0: SP->MAR) (T1: MAR->MDR) (T2: MDR->PC, SP+1->SP)**
9. **(总线) (I/O 设备) (主存)**
10. **(总线) (地址) (控制信号)**

三、 简答

1. ①指令是计算机执行某种操作的命令, 是计算机运行的最小功能单位。
②指令系统是计算机系统程序员接触到的计算机的所有功能, 包含了所有的机器指令, 还包括数据表示及寻址方式。
③指令周期是指 CPU 从主存取出并执行一条指令所需要的全部时间。
2. ①局部性原理包括时间局部性和空间局部性。时间局部性是指最近的未来用到的信息很可能是正在使用的信息, 因为程序在循环; 空间局部性是指最近的未来要用到的数据很可能与正在使用的数据在存储空间上是相邻的, 因为数据是以向量数组等形式簇聚的存储在一起。
②硬件利用局部性原理的最好例子就是 Cache。把程序中正在使用的部分存放在一个高速 Cache 中, 使得 CPU 的访存操作大多数情况下只访问 Cache, 从而减少程序执行时间。
3. ①关中断会进入不可响应中断的状态, 可以在保护中断现场时不被新的中断打断, 从而保证中断任务结束后能够正确返回中断处并继续正确执行。
②CPU 响应中断时有一次关中断过程, 是为了保存断点不被打断。在中断处理过程中也有一次关中断, 是为了恢复现场和屏蔽字时不被打断。
4. 不是。有些硬件系统不支持虚拟存储, 则不需要进行虚拟地址到物理地址的转换。
5. 提高总线宽度
优化总线仲裁算法, 减少仲裁时间
采用成组数据传输方式

四、 应用

1. 参考答案如下
(1) $X_{10} = (22/64)_{10} = (0.010110)_2 = 0.1011 \times 2^{-001}$, 则阶码 $E_x = -001$, 尾数 $M_x = 0.10110$ 然后按照题目要求写成补码形式
是 $[X]_{\text{补}} = \{[E_x]_{\text{补}}, [M_x]_{\text{补}}\} = \{1, 111; 0.10110\}$

同样 $Y_{10} = (-2.75)_{10} = (-010.11)_2 = -0.1011 \times 2^{+010}$ 则阶码 $E_Y = +010$ ，尾数 $M_Y = -0.10110$

是 $[Y]_{\text{补}} = \{[E_Y]_{\text{补}}, [M_Y]_{\text{补}}\} = \{0, 010; 1.01011\}$

(2) $[-E_Y]_{\text{补}} = 1, 101$

第一步求阶差， $\Delta E = [E_X]_{\text{补}} - [E_Y]_{\text{补}} = [E_X]_{\text{补}} + [-E_Y]_{\text{补}} = 1, 111 + 1, 101 = (1, 100)_{\text{补}} = 1, 101 = (-3)_{10}$ ，也即阶差为 3。

第二步对阶，X 的阶码加 3，尾数右移 3 位，对接过程中 X 的尾数有效位有所损失，此处采用恒置 1 法，对阶后 $[X]_{\text{补}} = \{0, 010; 0.00011\}$ 。

第三步尾数求和， $M_X + M_Y = 0.00011 + 1.01011 = 1.01110$ 。

最终答案为 $[X + Y]_{\text{补}} = \{0, 010; 1.01110\}$

2. 10-6 或 10-7, 12-6 或 12-7, 11-6 或 11-7。如果(5)加上多路选择器还可以有 10-5、12-5、11-5。

原解析：<https://learnerhub.net/#/spaces/715/questions/73>

严格来说，只有在有 mux 多路选择器的位置才可以进行数据定向，因为例如把 10 直接和 5 连接起来会造成电平信号的冲突。因为 5 是由 ID/EX 的寄存器直接输出的值，10 是 EX/MEM 的 O 寄存器直接输出的值，电平会撞在一起。

前置知识：理想流水线的基本工作原理，数据定向即数据旁路技术解决数据相关的基本原理，基本的 MIPS 指令（如 beq、add、addi、sw、lw）功能，和相应的流水段实现指令的大概过程，以上再国防科大的 MOOC 中都有详细解释，或参考相关博客或指令手册。

解决问题的基本思路就是，相邻流水段的数据相关就是相邻两条汇编指令的数据相关，写回段和执行段的数据相关是间隔一条无关指令的数据相关。

10-6:

ALUout 和 ALU 的上路输入产生数据相关，则可以

add r1, r2, r3 -> 10

add r4, r1, r5 -> 6

10-7:

与上面例子同理

10-5:

add \$t0, \$t1, \$t2 -> 10;

beq \$t0, \$zero, indexoutofbounds -> 5

indexoutofbounds 表示标号，是一个常数，也是指令的偏移地址部分。判 0 的逻辑在(5)位置，相对寻址的地址计算通过 ALU 完成。

11-5、11-6、11-7:

与上面同理，只要中间插入一个完全无关的指令即可

12-6:

load r2, (0)r1 -> 12

add r5, r6, r7

add r3, r2, r4 -> 6

12-7:

与上面同理

12-5:

load r2, (0)r1 -> 12

add r5, r6, r7

beq r2, \$zero, indexoutofbounds -> 5

12-14:

lw r1, (0)r2 -> 12

```
sw r1, (0)r3 -> 14
```

```
11-14:
```

```
add r1, r2, r3 -> 11
```

```
sw r1, (0)r4 -> 14
```

(本道大题答案及解析由“charles meng”整理)

3. 参考答案如下

- (1) 子程序调用指令被读取之前, PC 中存放的是该指令的地址, 内容为 3000H, SP 内容为 200H, 栈顶内容为 2000H。
- (2) 子程序调用指令执行后, 需要使用堆栈保存返回地址 $PC+1 \rightarrow (SP)$, 即栈顶内容为 $PC+1$, 也即 3002H, 而由于堆栈向上生长, $SP=SP+1=202H$, PC 内容为 1000H。
- (3) 子程序返回后, 取出栈顶内容保存的返回地址, $SP=SP-1=200H$, 栈顶内容为 2000H, PC 内容为将要执行的指令地址为 3002H。

4. 参考答案如下

- (1) 采用字节编址 (注意, 非字长编址, 某些答案是按照字长编址计算, 是错的), 容量为 $2^{24}B = 16MB$ 。
- (2) $\lceil 16M/512K \rceil = 2^5 = 32$, $\lceil 8/4 \rceil = 2$, $32 \times 2 = 64$ 片。
- (3) 存储器地址码需要地址线 24 位, 低 19 位高 5 位。片选译码的地址码位数是 $\lceil \log_2 M/m \rceil = \lceil \log_2 32 \rceil = 5$ 位。
- (4) 一共是 64 片负载, 一个 TTL 门电路可以驱动 8 个, $\lceil 64/8 \rceil = 8$, $\lceil 8/8 \rceil = 1$, $8+1=9$ 个。

5. 参考答案如下

- (1) 盘面数是 $5 \times 2 - 1 = 9$, 要去掉一个伺服面。柱面数是 $(12.4cm - 6.4cm) \times 2000/2 + 1 = 60001$ 。
- (2) 非格式化容量用位密度算, $(3 \times 64 \times 6000 \div 8) \times 60001 \times 9 \approx 72.42GB$
- (3) 磁盘传输速率为 $3 \times 64 \times 6000 \div 8 \frac{1}{r} = 16.47 MB/s$
- (4) 10 台需要台号 $\lceil \log_2 10 \rceil = 4$ 位, 64 扇区需要 $\lceil \log_2 64 \rceil = 6$ 位, 9 个盘面需要 $\lceil \log_2 9 \rceil = 4$ 位, 60001 柱面需要 $\lceil \log_2 60001 \rceil = 16$ 位, 地址方案为 [台号 4 位, 柱面号 16 位, 盘面号 4 位, 扇区号 6 位]。

2016

一、选择

1. **[D]**存储程序原理是冯诺依曼架构最本质的特征，绿书及 MOOC 都谈及了。
2. **[A]** 问的是处理器 P 的时钟周期数，直接主频 3GHz，排除 B、D。CPI 为 1.5，则程序指令数为 $\frac{3 \times 10^{10}}{1.5} = 2 \times 10^{10} = 20 \times 10^9$ 条。
3. **[A]**补码表示法的时候，符号位和最高数位的异或可以判断有符号数是否溢出。这个方法它需要的逻辑门较少，比较常用。
4. **[B]**可以写个程序验证一下，`printf("%08X\n", y);`。原理是，c 语言如果类型转换到的类型是有符号类型的话，例如 `int`，那么第一个需要考虑的是，不管转换前的类型是有符号，还是无符号。`int` 型能不能放得下之前的数据，意思就是原数据真值能不能保持不变放到新的数据类型里面，如果可以的话，原数据是保持不变的。例如本题，65530 是在 `int` 的类型范围之内的。那么，进行的是零扩展，而不是符号扩展，假如，题目本来是 `signed short`，为了尽量维持原数据不变，那么是需要符号扩展。如果不幸原来的数据范围超过了 `int`，那么是编译器实现（`compiler implementation`），意思就是编译器自己决定如何处理，c 语言没有规定。其实有一个小知识点是，本题那个 65530 是一个字面值常量，它的类型其实是 `int`。所以程序的第一步实际上已经发生了 `int` 到 `unsigned short` 的转换。类型转换具体可见：<http://docs.linuxtone.org/ebooks/C&CPP/c/ch15s03.html>。（linux c 内核一站式编程是一本关于 c 语言和 linux 编程的一本好的入门书）。
5. **[C]**常识性知识。
6. **[B]**立即数寻址，源操作数隐含在指令当中，而指令存放在存储器中，所以可以说立即数寻址的源操作数存在存储器。显然寄存器寻址源操作数在寄存器中。
7. **[C]**RISC 简单指令多，多数可在一个时钟周期内完成。RISC 采用组合逻辑控制比较多，组合逻辑控制的特点是，速度快，硬件资源需要多。微程序控制器因为需要访存微指令，所以速度慢，但是硬件资源需要少。RISC 指令集较为简单，硬件资源需要多一些也没有关系。RISC 内部通用寄存器数量一般来说比 CISC 多。
8. **[A]**应当让流水段大致相等，详情可看 CSAPP 的 P282~P287。
9. **[A]**数据旁路，即转发技术，是解决数据冒险的。
10. **[D]**基本概念。
11. **[C]**A 是望文生义。控制信号是译码产生的 B 错。垂直指令是将微命令统一编码的，每次只产生一个微操作，设置了微操作码字段。
12. **[D]**按字节编址，需要 ROM 芯片 $\frac{16K}{1K} = 16$ 片，需要 RAM 芯片 $\frac{48K}{4K} \times \frac{8}{4} = 24$ 片。
13. **[A]**ROM 也是随机存取。ROM 不可做高速缓存。ROM 不需要定时刷新。
14. **[C]**组相联映射，130 号单元在 $130 \div 64 = 2.03125$ ，也即第 2 块里。 $2 \bmod 4 = 2$ ，第 2 组。
15. **[]**408 同款争议题，不作论述，建议放弃此题
16. **[D]**TLB 放的是页表的副本，TLB 若命中，则页表必命中。
17. **[D]**带宽约为分辨率 \times 深度 \times 帧频 $\times 2 = 1920 \times 1080 \times 24 \times 60 \times 2 \approx 5972\text{Mbps}$ 。总带宽 50% 用

于刷新屏幕，则刷新屏幕所用的带宽要乘以 2。通信传输中的转换需要除 1000（有别于数据存储中除 1024）

18. **[B]** 平均存取时间 = 平均寻道时间 + 平均等待时间 + 传输时间 + 延迟 = $6\text{ms} + \frac{1}{2r} + \frac{4\text{KB}}{20\text{MB/s}} + 0.2\text{ms} = 6 + 6 + 0.2 + 0.2 = 12.4\text{ms}$ 。
19. **[C]** 中断向量表示的是中断服务程序入口地址。
20. **[D]** USB 是串行总线。热拔插特性（鼠标键盘），级联特性（USB 拓展坞）。

二、 填空

1. **(程序) (数据) (二进制)**
2. **(0000007FH) (FFFFFF7H) (00000076H)**
3. **(15392)**
4. **(不同) (相容微命令) (相同) (互斥微命令)**
5. **(BFE00000H) (BFH) (00H)**
6. **(主存) (操作码) (控制信号)**
7. **(逻辑) (逻辑地址位数)** 国防科大 2011 学年春学期《计算机原理》评分标准（群友 y 纠正）
8. **(10) (2)** 数据引脚只有一个的时候不能复用，具体参见《计算机组成原理学习要点与习题解析》（也就是那本**紫红色的书**）**p186**
9. **(一条指令执行结束) (存取周期结束)**
10. **(①) (⑤) (②) (③) (⑥) (②)**
11. **(4)**

三、 简答

1. 计算机是通过指令执行的时机不同来区分指令和数据的，在取指阶段，计算机从内存中取出的自然是指令；在执行阶段，指令执行需要操作数，此时计算机从内存中取出的是数据。指令周期是区分指令和数据的重要依据。
2. 一般是耗时最多的一段。指令流水各段长度不一的情况下，耗时较多的段将会令指令流水线无法连续不断的运行，会存在等待。消除指令流水线瓶颈的方法之一是重新设定和划分指令流水各段功能和时常，比如再次细分瓶颈段等。
3. Cache-主存层次的引入是基于程序的局部性原理考量的，可以改善 CPU 高速和主存低速不匹配的问题。该层次是由硬件实现的。引入主存-辅存的目的是解决主存空间不足，扩展主存地址空间。该层次是由软件实现的。
4. 点击鼠标会产生 I/O 中断；点击鼠标后打开程序，如果是第一次打开，需要从辅存调入内存中，会产生缺页中断。
5. 总线带宽 = 总线工作频率 × 总线宽度，所以要增加带宽，可以增加数据线数量增加总线宽度、提升时钟频率、采用成组传输数据的方式。

四、 应用

1. 参考答案如下
 - (1) 6 位 OP 字段，最多 $2^6 = 64$ 条指令；3 位寄存器编号，最多 $2^3 = 8$ 个寄存器；指令定长 32 位，所以 MAR 为 32 位；按字节编址，存储单元宽度是 8，所以 MDR 为 8 位。
(MDR 与存储字长同宽)

- (2) 转移指令的偏移量 Imm 是 16 位，补码表示，目标地址范围是 $-2^{15} \sim 2^{15} - 1$ 。
- (3) OP 字段是 001010, (R5)间接寻址 Md 是 10, R5 寄存器编号是 101, R4 是直接寻址 Ms 是 01, R4 寄存器编号是 100, 立即数 9ABCH 是 1001 1010 1011 1100。则 0010 1010 1010 1100 1001 1010 1011 1100 = 2AA59ABCH; 指令执行后源操作数和立即数相加，结果存放于目的操作数中，(R5)=R4+9ABC，也即 1234H+9ABCH=ACF0H，结果存到 5678H 地址所在的存储单元，主存 5678H 中的内容发生改变成为 ACF0H。
2. 为方便计算采取双符号位。 $[A]_{\text{补}} = 00001, 11.00010$; $[B]_{\text{补}} = 11110, 00.11011$ 。对阶后， $[B]_{\text{补}} = 00001, 00.00011$ 。然后进行尾数求差， $[A]_{\text{尾数}} + [-B]_{\text{尾数}} = 10.11111$ ，即 $[A - B]_{\text{补}} = 00001, 10.11111$ 。需要进行一次右规，得 $[A - B]_{\text{补}} = 00010, 11.01111$ 。恢复题目要求一位符号位是，0010, 1.0111。
3. 暂无。
4. 参考答案如下
- (1) 外设传输速率为 0.5MB/s，则每秒可传输 $\frac{0.5\text{MB}}{4\text{B}} = 125\text{K}$ 个 32 位，每 32 位需要 $18+2=20$ 条指令开销，耗时钟周期是 $20 \times 5 = 100$ ，故百分比为 $\frac{125\text{K} \times 100}{500\text{M}} = \frac{12.5}{500} = 2.5\%$
- (2) 当传输速率为 5MB/s 时， $\frac{5\text{MB}}{5000\text{B}} = 1000$ ，则每秒可进行 1000 次 DMA 传输，而 DMA 开销是每次 500 个时钟周期，共 $1000 \times 500 = 500000$ 时钟周期。百分比为 $\frac{500000}{500\text{M}} = 0.1\%$

2015

一、选择

1. **[B]** 每条指令运行时间 = $CPI \times \text{时钟周期}$, 则 A 每条指令运行时间为 3ns, B 为 2ns, 速度比 2:3。
2. **[D]** 系统吞吐率取决于单位时间内处理指令的数量, 吞吐率越高, 则表示单位时间内能执行的指令数越多, 程序运行越快。其余三项与执行时间没有必然的联系。
3. **[D]** -127 的补码为 81H, 若扩展为 32 位, 则需要前面添补符号位, 扩展后为 FFFFFFF81H。
4. **[A]** 大端存储是高字节存储在低地址的方法, 但与本题并无关系。无符号短型的 12345, 在 C 语言中以补码形式存储, $12345 = 8192 + 4096 + 32 + 16 + 8 + 1 \rightarrow 0011\ 0000\ 0011\ 1001$, 进行扩展时只需补 0 就行, 则 i2、i3、i4 还是 12345。
5. **[C]** $EA = (PC) + A$, 因为按字节编址, 转移指令由两个字节组成 (机器字长条件用不上, 莫盲目凑理由)。所以转移指令取出后的 PC 值为 $2000H + 2 = 2002H$ 。转移指令成功后目标地址为 $12H + 2002H = 2014H$ 。
6. **[A]** 操作码是一个指令字必有的。可以有零地址指令。
7. **[D]** 公共微操作一般包含取指周期、间址后期及中断周期一些微指令, 而 D 将 IR 中的指令传至 MAR 是执行阶段的微指令。
8. **[C]** SRAM 芯片地址不复用, 数据线复用 (即算的时候不用乘 2), 则 $4M = 2^{22}$, $22 + 4 = 26$ 根。
9. **[B]** 按字节编址, 主存块 32B, 则 1023 号单元是 $1023 \bmod 32 = 31$, 在第 31 号主存块。16 行二路组相联, $31 \bmod 8 = 7$, 在第 7 组 Cache。
10. **[C]** Cache 有专门的硬件机构, 所以硬件完成。缺页是一种软件机制。
11. **[D]** 页表仅标明当前存储空间情况, 与局部性无关。
12. **[D]** 辅存必定非易失, 硬盘、光盘等。
13. **[C]** 基本概念。
14. **[A]** 查询一次开销 500 个时钟周期, 每秒至少查询 200 次, 耗时占比为 $\frac{500 \times 200}{500M} = \frac{100000}{500M} = 0.02\%$ 。
15. **[A]** 基本概念。
16. **[D]** 屏蔽字就是为了屏蔽需要屏蔽的, 所以 I 和 II 不选。
17. **[C]** 统一编址用的是访存指令, 独立编址用的是 I/O 指令。
18. **[B]** 总线频率 20MHz, 一个总线周期占 2 个时钟周期, 也即每秒 10M 个总线周期, 一个总线周期传输 8B 信息, 每秒共传输 80MB 信息。
19. **[C]** 三总线结构是 I/O 总线、存储总线、DMA 总线。
20. **[C]** 一条总线上可有多多个主设备竞争使用权, A 错; 一个设备扮演的角色不是固定, 有时是主设备, 有时是从设备, B 错; 只有主设备能引起总线操作, D 错。

二、填空

1. (天河二号) (亿亿次)
2. (2000H) (3000H)

3. (0000007F) (127) (0000007F) (127) (00000080) (128) (00000000) (0) (从左到右从上到下)
4. (64) (16) (10) (15)
5. (全相联映射) (直接映射) (2)
6. (I/O 接口) (端口) (地址)
7. (周期挪用) (交替访存)
8. (小容量廉价磁盘) (整体的大容量磁盘) (备份冗余)
9. (结构冒险) (数据冒险) (分支冒险)

三、 简答

1. 采用二进制是依据电子器件的物理特性，只有 0 和 1 两个码，可以用电路通断来表示。在电脑内部信息以二进制方式存储，一条指令的执行的不同阶段可以区分出指令和数据。取指阶段从主存中取到的是指令，执行阶段从主存中取到的是数据。
2. ①对于结果为 $\pm 1x.xx\cdots x$ 的情况，需要进行右规。右规时，尾数右移一位，阶码加 1。右规操作可以表示为： $M_b \leftarrow M_b \times 2^{(-1)}$, $E_b \leftarrow E_b + 1$ 。右规时需要注意：尾数右移时，最高位 1 被移到小数点前一位作为隐藏位，最后一位移出时需要考虑舍入；阶码加 1 时，直接在末尾加 1。
②对于结果为 $\pm 0.00\cdots 01x\cdots x$ 的情况，需要进行左规。左规时，数值位逐次左移，阶码逐次减 1，直到将第一位 1 移到小数点左边。假定 k 为结果中 \pm 和左边第一个 1 之间连续 0 的个数，则左规操作可以表示为： $M_b \leftarrow M_b \times 2^k$, $E_b \leftarrow E_b - k$ 。左规时注意：尾数左移时数值部分最左 k 个 0 被移出，因此相对来说，小数点右移了 k 位。因为尾数相加时默认小数点位置在第一个数值位(隐藏位)之后，所以小数点右移 k 位后被移到了第一位 1 后面，这个 1 就是隐藏位；执行 $E_b \leftarrow E_b - k$ 时，每次都在末位减 1，一共减了 k 次。
3. 单周期处理器执行一个指令固定地用一个时钟周期，它能使设计得到简化。在计算机中这种能够在一个时钟周期内完成的指令也占了大部分，如此设计也有一定的合理性。此时每条指令的 CPI 是相同的，均为 1。单周期处理器简化了设计，但在性能上做了妥协。比起多周期处理器，单周期处理器缺乏灵活性，因为不管指令复杂度如何，都需要用相同的时间去执行，造成了时间上的浪费。
4. 不影响，在直接映射的情况下，Cache 块和主存块是一一对应好的，无论何种替换算法，产生替换时替换结果都是一样的。
5. 同步总线：总线系统有一个共所有设备使用的统一时钟，设备之间都按照约定俗成的时序进行操作和完成信息交换。同步总线的所有信号命令都是与总线时钟同步的，所以总线的使用设计调试都较为简单。所有设备的总线操作所需要的总线时钟都是整数，且总线操作周期需设计得满足最慢设备使用，从而使高速设备不能高效使用总线，造成时间浪费。
异步总线：总线系统没有统一的时钟，任何一件事发生都只能是前一个或一些时间的结束，所有设备都以“握手”的方式进行联系从而完成信息交换。总线对设备速度差异的适应性更强，慢速设备对快速总线的影响仅限于本次总线操作，不影响下一次快速总线设备操作，效率高。异步总线无统一的时钟，设计控制调试都比较复杂。

四、 应用

1. $[X]_{\text{补}} = 1101$, $[Y]_{\text{补}} = 0110$, $[-X]_{\text{补}} = 0011$, 按 booth 乘法，可以得出最终结果位 1,101110。具体步骤如下

部分积	乘数	附加位	说明
00,000	0110	0	$Y_n Y_{n+1} = 00$, 部分积右移一位
00,000	0011	0	$Y_n Y_{n+1} = 10$, $+[-X]_{\text{补}}$
+00,011			
00,011			部分积右移一位
00,001	1001	1	$Y_n Y_{n+1} = 11$, 部分积右移一位
00,000	1100	1	$Y_n Y_{n+1} = 01$, $+ [X]_{\text{补}}$
+11,101			
11,101	110		最后一步不移位, 得 $[xy]_{\text{补}}$

2. 参考答案如下

- (1) $5T + (n-1)T = 5T + (5-1)T = 9T$, 需要 9 个时钟周期。
- (2) 不相同, 此时寄存器 \$6 中内容已经发生了改变。
- (3) 存在数据冒险, 当 I3 指令与 I1 产生数据冒险, I3 的 ID 段与 I1 的 WB 冲突(此时寄存器 1 中数据还未写回)所以 I3 必须将 ID 阶段延。I4 由于 I3 还在 IF 段所以要延迟 1 个时钟周期。只能与上一指令的 ID 阶段并行。
- (4) 采用停顿、气泡的办法, 即将会产生冒险的阶段暂停合适的时钟周期, 以消除冒险。
- (5) 转发技术可以消除数据冒险。

3. 参考答案如下

- (1) 页大小 128B, 则 $\frac{128B}{B} = 2^7 \rightarrow 7$ 位页内偏移, 是 0~6; 剩余 9 位为虚拟页号, 是 7~15。16 项 TLB 二路组相联, $\frac{16}{2} = 8 = 2^3$, 组索引是 3 位, 是 7~9; 剩余部分为 TLB 标记, 是 10~15。
- (2) 同上, 页内偏移 7 位, 是 0~6; 则剩余 5 位是物理页号, 是 7~11。
- (3) 物理地址 12 位。块大小 8B, 则块内偏移是低 3 位, 0~2; 16 行 Cache 直接映射, 索引是 4 位, 3~6; 剩余 5 位为主存标记, 是 7~11。
- (4) 主存地址 062CH=0000 0110 0010 1100, 查 TLB 标记 00 0001=01H, 可知页框号是 19, 即 0001 1001; 有效位 1, 已被装入内存。拼接可得物理地址是 1 1001 010 1100, 也就是 1100 1010 1100。按物理地址的字段定义, 主存标记是 7~11 位, 也就是还是 19, 查 Cache 可知 19 标记在 Cache 第 5 行, 块内偏移是 100, 也就是第 4 个字, 取出 46。
(切记, 表里的值为 16 进制, 不可当作 10 进制去傻傻的转成 2 进制。)

4. 盘面数为 $2 \times 4 - 1 = 7$ 。柱面数为 $(8-4) \times 10 \div 2 \times 1000 + 1 = 20001$ 。

- (1) 非格式化容量用位密度算, 内直径 $\times \pi \times$ 位密度 \times 盘面数 \times 柱面数 $= 40 \times 3.14 \times 9000 \times 7 \times 20001 \div 8 \div 1024 \div 1024 \div 1034 \approx 18.4GB$
格式化容量用扇区算, 扇区大小 \times 每磁道扇区数 \times 柱面数 \times 盘面数 $= 512 \times 200 \times 20001 \times 7 \approx 13.35GB$
- (2) 平均存取时间 = 平均寻道时间 + 平均等待时间 $= 5ms + \frac{1}{2r} = 5 + \frac{60}{12000 \times 2} = 7.5ms$
- (3) DMA 平均传输长度是 8 扇区, 是一柱面的 $8/200$, 则一次 DMA 耗时是 $\frac{8}{200} \times \frac{60}{12000} = 0.2ms$, $\frac{1000ms}{0.2ms} = 5000$, 也就是 1s 有 5000 次 DMA, 则 $\frac{5000 \times 1000}{1 \times 10^9} = 0.5\%$ 。
(磁盘转一圈的时间 $t = 1/r$, r 是转速)
- (4) 转一圈要 $\frac{60}{12000} = 5ms$, 即 200 个扇区, 1 秒要转 200 圈, 总共中断次数是 $2000 \times 200 \times \frac{512B}{8B} = 2^8 \times 1000$ 次, 中断开销共 $2^8 \times 5 \times 1000000$ 个周期。中断开销周期占比是 $\frac{2^8 \times 5 \times 10^6}{10^9} = \frac{1280}{1000} = 128\%$, 不可行。

2014

一、选择

1. **[A]**将各数转为十进制后，分别是 153、151、152、152。
2. **[D]**阶码的位数影响浮点数范围，尾数的位数影响浮点数精度，基数则都影响。
3. **[B]**补码零表示是唯一的 0.0000。
4. **[D]**根据 Amdahl 定律，改进后的时间 $20 = \frac{80}{x} + 20$ ，x 也就是乘法运算要改进的倍数，几乎无穷了，所以乘法操作的时间无法估计。
5. **[C]**所谓间址，就是该寄存器、地址单元里存的不是操作数，存的是操作数地址。
6. **[B]**基本常识。
7. **[C]**在一个 CPU 周期中，一组实现一定功能的微命令的组合构成一条微指令，有序的微指令构成一段微程序，微程序的作用是实现一条对应的机器指令。
8. **[C]**需要注意的是，MIPS 机中寄存器内存放的数据是以补码形式存放的。
 $[s0 + s1]_{\text{补}} = [s0]_{\text{补}} + [s1]_{\text{补}} = 1000\ 0\ 0\ 0\ 0\ 0\ 0\ 0 + 1101\ 0\ 0\ 0\ 0\ 0\ 0\ 0$
 $= 10101\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = 0x50000000$
 $[s0 - s1]_{\text{补}} = [s0]_{\text{补}} + [-s1]_{\text{补}} = 1000\ 0\ 0\ 0\ 0\ 0\ 0\ 0 + 0011\ 0\ 0\ 0\ 0\ 0\ 0\ 0$
 $= 1011\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = 0xB0000000$
9. **[D]**
10. **[B]**32K/4K=8，8bit/4bit=2，需要 16 片。
11. **[B]**字长 32 位，按字编址也即按 4B 编址， $\frac{8MB}{4B} = 2M$ ，寻址范围是 2M。
12. **[B]**基本常识。
13. **[C]**组相联不会冲突失效，所以只考虑直接映射。各选项的单元号对 4 取余，可得所在的块号，分别是 A:13、25；B:12、36；C:14、71；D:11、59。然后块号再对 16 取余，可得所在 Cache 行号，分别是 A:13、9；B:12、4；C:14、7；D:11、11。再对 4 取余，可得所在块内地址，分别是 A:1、1；B:0、0；C:2、3；D:3、3。可知 C 选项不会引起冲突失效。
14. **[A]**处理器和内存以外的事件引起的中断为外中断。
15. **[C]**中断向量提供的是中断服务程序的入口地址。
16. **[D]**基本常识。
17. **[B]**CD-ROM 只能读不能写入。
18. **[D]**串行工作即程序查询方式。
19. **[A]**三总线结构是：I/O 总线、主存总线、DMA 总线。
20. **[B]**突发传送只第一次传输地址，之后不用，需要一个时钟周期，。传输 128 位数据，需要传 4 次 32 位，需要 4 个时钟周期。共计 5 个时钟周期， $\frac{5}{100M} = \frac{1}{20M} = 50ns$ 。

二、填空

1. **(00101111) (0) (1) (1) (0)**

2. (对阶) (规格化结果尾数)
3. (奇) (偶)
4. (寄存器寻址) (寄存器间接寻址) (立即数寻址)
5. (水平) (垂直) (垂直)
6. (取指) (执行) (中断)
7. (分散) (透明) 以 MOOC 为主。
8. (顺序) (直接)
9. (硬件) (硬件) (软件) 以 MOOC 为主。
10. (统一的时钟信号) (握手信号)
11. (总线主设备) (集中式仲裁) (分散式仲裁)

三、简答

1. len 是无符号 unsigned 类型，当 len 为 0 时，而具体执行却是 len-1，即 11.....1，使得循环不断执行，最终发生越界访问，发生存储器访问异常。修改方法：将 len 声明为 int 型，或循环测试条件改为 $i < \text{len}$ 。
2. 加法器是 ALU 中最核心的单源，其他减法乘法除法都可以转化为相应的加法运算和移位运算。定点乘法由左移和加法实现，定点除法由加法和右移实现，同样浮点运算也可以由定点的运算组合实现。
3. 可以消除结构冒险。因为在传统的五段流水线中，IF 和 MEM 的冲突经常发生，即某指令需要取指的同时，另一条指令在等待访存取操作数。由于两个操作不可同时进行，造成了流水线若干时钟周期的延迟。若分开存储，IF 和 MEM 可同时进行，避免了结构相关问题。
4. ①CPU 不需要访存，也即 IO 访存请求何 CPU 未发生冲突。②CPU 正在访存，则必须等待存取周期结束后，CPU 再将总线占有权让出。③CPU 同时也要访存，出现了冲突，此可 CPU 要暂时放弃总线占有权，即 DMA 具有优先权。
5. 影响同步总线带宽的主要因素是总线宽度和总线时钟频率。所以有两个措施：增大总线宽度、增加总线时钟频率。

四、应用

1. 商的符号是 $Q_0 = X_0 \oplus Y_0 = 1$ 。 $|[X]_{\text{原}}| = 0.1011$ ， $|[Y]_{\text{原}}| = 1.1101$ ， $[-Y]_{\text{补}} = 1.0011$ 。运算过程如下：

	被除数	商
	0.1011	0.0000
$+[-Y]_{\text{补}}$	1.0011	
	1.1110	0.0000
左移	1.1100	0.0000
$+ Y $	0.1101	
	0.1001	0.0001
左移	1.0010	0.0010
$+[-Y]_{\text{补}}$	1.0011	
	0.0101	0.0011

左移	0.1010	0.0110
$+[- Y]_{\text{补}}$	1.0011	
左移	1.1101	0.0110
$+ Y $	1.1010	0.1100
	0.1101	
	0.0111	0.1101

故商值 $Q' = 0.1101$ ，加上符号位 $[X/Y]_{\text{原}} = 1.1101$

2. 参考答案如下

(1) $10^7 \times 100\text{ps} = 1\text{ms}$

(2) 用加速比计算，可知快了 $\frac{10^7 \times 100}{10^7 \times 20 + 80} \approx 5$ 倍。

(3) 发生在 I2 取数时，此时 I1 还未写回寄存器（I2 到 ID 时，I1 还未到 WB 阶段）

指令编号	时钟周期								
	1	2	3	4	5	6	7	8	9
1	IF	ID	EX	MEM	WB				
2		IF	STALL(ID)	STALL	ID	EX	MEM	WB	
3					IF	ID	EX	MEM	WB

(4) 数据旁路技术，将前面计算结果（EX 后其实计算结果就出来了）采用硬件技术传送到其他需要的地方（只能在时空上向后）

指令编号	时钟周期							
	1	2	3	4	5	6	7	8
1	IF	ID	EX	MEM	WB			
2		IF	STALL	ID	EX	MEM	WB	
3				IF	ID	EX	MEM	WB

3. 参考答案如下

(1) 主存空间 64MB，按字节编址，则地址长度为 $\frac{64\text{MB}}{\text{B}} = 64\text{M} = 2^{26} \rightarrow 26$ 位。

块大小 64B，则块内偏移是 $\frac{64\text{B}}{\text{B}} = 64 = 2^6 \rightarrow 6$ 位。

Cache 数据区 4KB，块大小 64B，则有 $\frac{4\text{KB}}{64\text{B}} = \frac{2^{12}}{2^6} = 2^6$ 块，4 路组相联则是 $\frac{2^6}{4} = 2^4$ 组，4 位。

主存标记为 $26 - 6 - 4 = 16$ 位。

主存地址字段划分为

25.....10	9.....6	5.....0
主存标记 16 位	组索引 4 位	块内偏移 6 位

(2) Cache 数据区容量为 $4\text{KB} = 4096\text{B} = 32768\text{bit}$ 。

标记阵列每一行有 $16 + 1 + 1 + 2 = 20\text{bit}$ ，16 位主存标记，1 位有效位，1 位脏位，2 位 LRU 替换位。一共有 2^6 块，每一块对应一行标记，故标记阵列大小为 $20\text{bit} \times 2^6 = 1280\text{bit}$ 。Cache 总容量大小为 $32768 + 1280 = 34048\text{bit}$ 。

(3) 0~4344 有 4345 个单元，cache 大小 64， $4345/64 = 67.89$ 所以要访问前 68 个块（0~67），由于一共 64 块，所以会产生替换。第一轮访问：0~68 每一块都是第一次未命中，其中第 64~第 67 是因为替换所以未命中的。第二轮：访问第 0~第 3 时由于位置上是上一轮的 64~第 67，所以按照组相联第 0~第 3 块替换到第 1 行去，然后接下来的第 4~第 15 皆命中；然后第 16~第 19 替换到 2 行，第 20~第 31 皆命中；然后第 32~第 35 替换到第 3 行，第 36~第 47 皆命中；第 48~第 51 替换到第 0 行，第 52~第 63 皆命中；第

64~第 67 替换到第 1 行。以此看来，第一轮全不命中 68 次，第二轮不命中 20 次，以此类推后面的第三轮、第四轮、……、第十六轮都不命中 20 次。共有 $68 + 20 \times 15 = 368$ 次未命中。而又因为任何一个块如果不命中，那么必然是该块里的第一个单元不命中，载入该块后，块内其余单元都会命中。所以总共访问 4345 个单元，不命中单元是 368 个，命中率 $p = \frac{4345 \times 16 - 368}{4345 \times 16} \approx 99.47\%$

	第 0 行	第 1 行	第 2 行	第 3 行
0 组	0 (64) (48)	16 (0) (64)	32 (16)	48 (32)
1 组	1 (65) (49)	17 (1) (65)	33 (17)	49 (33)
2 组	2 (66) (50)	18 (2) (66)	34 (18)	50 (34)
3 组	3 (67) (51)	19 (3) (67)	35 (19)	51 (35)
4 组	4 4	20 20	36 36	52 52
5 组	5 5	21 21	37 37	53 53
6 组	6 6	22 22	38 38	54 54
7 组	7 7	23 23	39 39	55 55
8 组	8 8	24 24	40 40	56 56
9 组	9 9	25 25	41 41	57 57
10 组	10 10	26 26	42 42	58 58
11 组	11 11	27 27	43 43	59 59
12 组	12 12	28 28	44 44	60 60
13 组	13 13	29 29	45 45	61 61
14 组	14 14	30 30	46 46	62 62
15 组	15 15	31 31	47 47	63 63

(注：黑色为第一轮访问块的序列，红色为第二轮访问块的序列，被 () 包住意味发生了替换，未被包住则是命中。)

(4) 平均访问时间：

$$\text{命中开销} + (1 - p) \times \text{失效开销} = 1 + 100 \times (1 - p) = 1 + 0.0053 \times 100 = 1.53 \text{ 个周期}$$

4. 参考答案如下

- (1) 百分比为 $\frac{30 \times 400}{500M} = \frac{12}{5K} \approx 0.00006\%$
- (2) 每秒传输 50KB，需要传输 $\frac{50KB}{2B} = 25K$ 次，百分比为 $\frac{25K \times 400}{500M} = \frac{1}{50} = 2\%$
- (3) 每秒传输 4MB，需要传输 $\frac{4MB}{16B} = 0.25M$ 次，百分比为 $\frac{0.25M \times 400}{500M} = \frac{1}{5} = 20\%$
- (4) 第三种硬盘传输的开销占比过大，采用程序查询方式效率太低。

2013

一、选择

1. [C]冯诺依曼机的核心就是存储程序原理。
2. [A]
3. [C] $MIPS = \frac{\text{主频}}{CPI} = \frac{200}{2.5} = 80$ 。(注意, MIPS 的计算公式中主频单位是 MHz)
4. [A]P1 的 CPI 为 $1 \times 10\% + 2 \times 20\% + 3 \times 50\% + 4 \times 10\% = 2.8$; P2 的 CPI 为 $2 \times (10\% + 20\% + 50\% + 20\%) = 2$ 。P1 执行时间为 $\frac{100 \times 2.8}{1.5G} = 187ns$, P2 执行时间为 $\frac{100 \times 2}{1.5G} = 100ns$ 。
5. [C]
6. [D]-256 的补码为 1,111 1111 0000 0000=FF00H。符号扩展补 F, 为 FFFF FF00H。
7. [D]最常见的有零地址、一地址、二地址。所以都有可能。
8. [C]指令执行过程为: 先立即数寻址取出 100, 然后间接寻址取出以 R1 内容为地址的操作数, 加和后将结果放到 R4 中, 寄存器寻址。
9. [C]超标量流水线技术目的在于使每个时钟周期内可并发多条独立指令。
10. [A] IR 存放的是当前正在执行的指令。绿书 P8。(某答案谈译码器从 CU 中分离, 故选了 B, 但应试应当尽量参考指定教材, 绿书中并未将译码器 ID 从 CU 中分离, 而逐字来看绿书也并没细致说 IR 存放的是正在执行的指令的操作码。)
11. [C]字段直接编码发每个字段独立编码, 都需要保留一种编码表示空白操作, 实际微命令数是 8、4、14、27、16。 $\lceil \log_2 8 \rceil + \lceil \log_2 4 \rceil + \lceil \log_2 14 \rceil + \lceil \log_2 27 \rceil + \lceil \log_2 16 \rceil = 18$ 。
12. [C]首先, 按照 IEEE754 将 -6.75 表示出来, $-6.75 = -110.11 = -1.1011 \times 2^2$, 符号位 1, 阶码为 $127+2=129=1000\ 0001$, 尾数 1.1011, 按 IEEE754 写出来为 1 1000 0001 101100……, 也即 1100 0000 1101 1000 0000 0000 0000 0000=C0D80000H, 按照大端存储, 高位字节放在低地址处, 低地址 1000H 处放的应是高位字节 C0H, 1003H 处自然就是 00H。
13. [B]按字节编址, 4 根地址线, 一共 16 字节, 一个块 2 字节, 共 8 块。Cache 容量是 4 块, 第 7 块所在 Cache 是 $7 \bmod 4=3$, 第 3 号 Cache 块。
14. [B]Cache 调度由硬件完成工作, 页调度由系统软件完成工作。
15. [A]考察写策略的定义。
16. [D]中断向量地址是中断向量表所在的地址。(地址的地址)
17. [B]
18. [B]磁盘、光盘、U 盘都属于外部 I/O 设备, CPU 不能直接访问。
19. [A]格式化后的磁盘最小单位就是扇区。
20. [D]AGP 是局部总线, ATA 是并行的, USB 属于外部总线, 与通信总线还是存在一定区别。

二、填空

1. (每秒执行多少百万条指令) (每秒进行多少百万次浮点运算)
2. $(-2^{-2} \times 2^{-4}) (-2^3) ((2^{-1} + 2^{-2}) \times 2^3) (2^{-1} \times 2^{-4})$

3. (表示范围) (表示精度) (0,111 1101) (-25.75)
4. (90ns) (10/3)
5. (SRAM) (DRAM) (DRAM)
6. (40%) (18.8) 由于是组相联策略，可以求映射组号，看有没有出现 3 次以上的组号。访问主存情况如下： $6\%4 = 2$ 、 $2\%4 = 2$ 、 $5\%4 = 1$ 、 $4\%4 = 0$ 、 $3\%3 = 3$ 、 $7\%4 = 3$ 。同一组号最多出现 2 次，意味着没有发生替换，只考虑第一次不命中即可。按照题中访问地址流来看，第二次访问的 B2、B、B7、B6 都命中了，所以命中率 40%，平均时钟周期 $0.4 \times 2 + 0.6 \times 30 = 18.8$ 。
7. (相容) (互斥) (互斥) (相容)
8. (有中断请求) (允许中断)
9. (寄存器) (主存)
10. (事件) (同步) (异步) 绿书为主。
11. (北桥) (南桥) 当前计算机大多不再有北桥，建议放弃此题。

三、简答

1. 计算机通过指令周期的不同阶段来区分指令和数据。一条指令通常分为取指阶段和执行阶段等。在取指阶段访存取出指令，在执行阶段访存取出操作数。
2. 无符号算术运算中没有溢出，所有的无符号运算都是以 2 的 n 次方为模， n 是结果中的位数。有符号数运算结果有上溢和下溢，对于上溢程序退出进行中断并报错，对于下溢系统一般按零值处理。
3. 程序的指令都是有序的，指令的执行分为三个阶段：取指、分析、执行阶段中。根据程序计数器 PC 的值取出指令，然后进行分析译码，再取出数据进行执行操作。执行完毕后 PC 自增，开始执行下一条指令。当遇到跳转时，会进行中断保留现场，然后执行中断操作，中断返回后再继续按照原定的顺序执行。
4. 多级存储系统：Cache-主存系统，使用高速缓冲解决了主存和 CPU 速度不匹配的问题；主存-虚存系统，解决了主存容量问题。
多模块存储器：一种是单体多字存储器，另一种是多体并行存储器。两种方案都提高了存储器带宽，加快了存取速度。
5. 不是。
I/O 接口一般指 CPU 和 I/O 设备之间的连接部件。
I/O 端口一般指 I/O 接口中的各寄存器。为了便于程序对这些寄存器进行访问，通常给每个寄存器分配一个地址编号，这种编号被称为 I/O 端口地址，相应寄存器被称为 I/O 端口。
一个接口中往往包含若干个端口。

四、应用

1. $A = -0.1011$, $B = -0.0111$, $[A]_{\text{补}} = 11.0101$, $[B]_{\text{补}} = 11.1001$, $[-B]_{\text{补}} = 00.0111$
过程如下

	部分积	乘数
	00.0000	1.01010
加 $[-B]_{\text{补}}$	+00.0111	1.01010
	00.0111	1.01010
右移	00.0011	1 1.0101

加[B] _补	+11.1001	1	1.0101
	11.1100	1	1.0101
右移	11.1110	0 1	1.010
加[-B] _补	+11.1001	0 1	1.010
	00.0101	0 1	1.010
右移	00.0010	101	1.01
加[B] _补	+11.1001	101	1.01
	11.1011	101	1.01
右移	11.1101	1101	1.0
加[-B] _补	+00.0111	1101	1.0
	00.0100	1101	1.0

故[B × A]_补 = 00.01001101

2. 参考答案如下

(1) 16 根地址线，地址为 16 位

系统程序区：0~1023 地址空间：0000000000000000~0000001111111111

用户程序区：1024~3071 地址空间：0000010000000000~0000101111111111

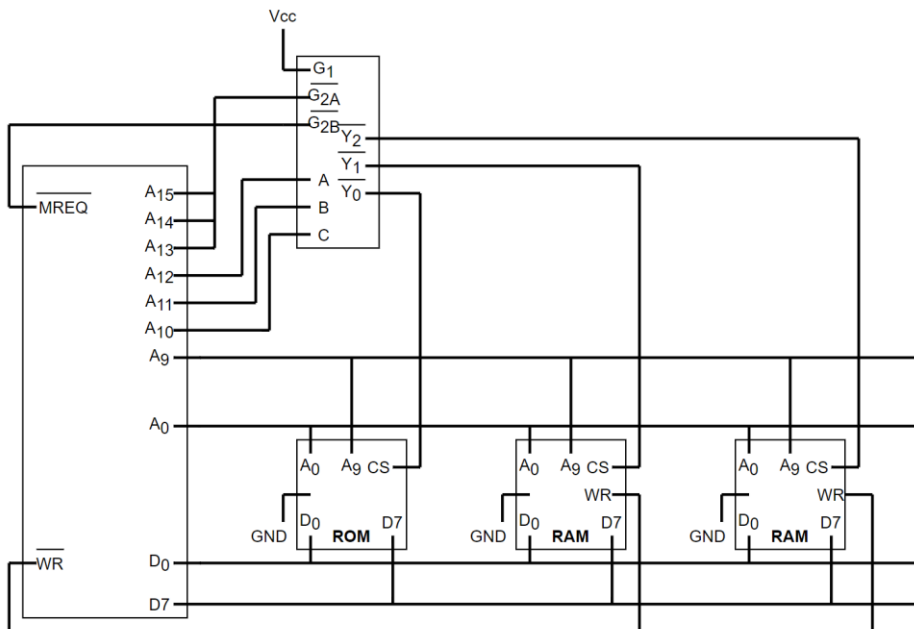
(1023 = 2⁹ + 2⁸ + + 2¹ + 2⁰，所以 0~9 位为 1，其余为 0，以此类推 3071 = 2048 + 1023 = 2¹¹ + 2⁹ + 2⁸ + + 2¹ + 2⁰，所以 0~9、11 位为 1，其余为 0。地址最右为 0)

(2) 数据线是 8 根，所以尽量选择 8 位的芯片

系统程序区是 1KB，且系统区为只读，所以选择 1K×8 位 ROM 一片

用户程序区是 2KB，且用户区为读写，所以选择 1K×8 位 RAM 两片

(3) 逻辑图如下



A₀~A₉、D₀~D₈、WR 对应连接，MREQ 和 A₁₅~A₁₃ 接 G_{2B} 和 G_{2A} 对译码器进行使能。

A₁₀、A₁₁、A₁₂ 接 A、B、C 作为输入，三个 Y 输出端分别接存储芯片片选端 CS。

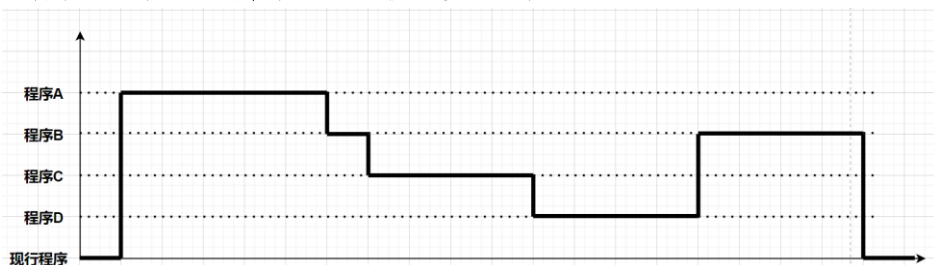
(4) 其他方案就是芯片的选择不同而已。在上一问基础上，可将 1K×8 位 RAM 两片替换位 1K×4 位 RAM 四片。

3. 参考答案如下

(1) 屏蔽码如下

中断源	屏蔽码			
A	1	1	1	1
B	0	1	0	0
C	0	1	1	1
D	0	1	0	1

(2) 硬件优先级是 $A > B > C > D$ ，当同时出现请求时，会先响应 A 中断，并屏蔽其他所有中断，直到 A 处理结束。按照硬件优先级顺序，B 先被响应。在执行 B 的时候，设置屏蔽字，因为屏蔽字设置后处理的优先级是 $C > D > B$ ，所以紧接着 B 中断就会被 C 打断。C 开始执行并设置屏蔽字，就会把 B 给屏蔽。C 处理完后处理 D，D 处理完后再执行 B 剩余的程序。直到结束。CPU 执行轨迹如下：



4. 参考答案如下

(1) 平均存取时间=平均寻道时间+平均旋转等待时间+传输时间+控制器开销

$$\text{磁盘 a: } 11 + 0.5 \times \frac{60}{6000} \times 1000 + \frac{1024\text{B}}{20\text{MB/s}} \times 1000 + 0.2 = 16.2512\text{ms}$$

$$\text{磁盘 b: } 9 + 0.5 \times \frac{60}{12000} \times 1000 + \frac{1024\text{B}}{40\text{MB/s}} \times 1000 + 0.1 = 11.6256\text{ms}$$

(2) 磁盘 a: $11 + 0.5 \times \frac{60}{6000} \times 1000 + \frac{1024\text{B}}{20\text{MB/s}} \times 1000 \times 2 + 0.2 = 16.3024\text{ms}$

$$\text{磁盘 b: } 9 + 0.5 \times \frac{60}{12000} \times 1000 + \frac{1024\text{B}}{40\text{MB/s}} \times 1000 \times 2 + 0.1 = 11.6512\text{ms}$$

(3) 根据(1)可知，磁盘读写的时间和控制器开销比重较小而寻道时间和等待时间比重较大；根据(2)的计算可知，即使加倍要传输的数据量，对总的平均读取时间影响小，说明磁盘性能主要因素在于寻道时间和磁盘转速。所以要从减少寻道时间和等待时间两方面改善。寻道时间取决于算法策略，可以对寻道算法进行改进；等待时间只与转速有关，可以提高磁盘转速。

2012

一、选择

1. **[B]**指令是计算机运行的最小功能单位，指令系统反映计算机基本功能。
2. **[D]**CPI 是一条指令执行的时钟周期数，A 执行一条指令耗时 $1.2\text{ns} \times 2 = 2.4\text{ns}$ ，B 执行一条指令耗时 $2\text{ns} \times 1 = 2\text{ns}$ 。速度比是 $\frac{1}{T_A} : \frac{1}{T_B} = \frac{T_B}{T_A} = 1.2$ ，B 比 A 快了 1.2 倍。
3. **[D]**相对寻址的有效地址 $EA=(PC)+A$ 。
4. **[C]**浮点数总位数不变情况下，阶码位数少，则尾数多，表示的范围变小，精度变高。
5. **[C]**寄存器不支持子程序嵌套和递归，子程序起始位置和栈顶不支持子程序递归。
6. **[B]**不透明也就是能看得见摸得着可操控，PC 是程序员可以操控的，通过一些程序控制指令来改变其值，其他寄存器都是 CPU 内部工作寄存器，程序员不可见。
7. **[B]**微命令是构成控制信号序列的最小单位。
8. **[D]**程序状态字寄存器是描述系统运行状态的部件。
9. **[A]**最短字长编码需要留一个编码表示空操作，实际操作数是 31 个，需要 5 位。
10. **[C]**SRAM 速度快，集成度低，功耗大；DRAM 速度相对慢，集成度高，功耗低。
11. **[B]**字长 32 位，按字编址即按 4B 编址， $512\text{MB}/4\text{B}=128\text{M}$ ，寻址范围是 $0\sim128\text{M}-1$ 。
12. **[C]** $64\text{K} = 2^{16} \rightarrow 16\text{位}$ ， $16+4=20$ 。
13. **[B]** $58 \bmod 8=2$ ，放在第 2 行。
14. **[A]**4GB，按字节编址， 2^{32} ，地址 32 位，逻辑空间范围是 $0\sim1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111$ ，也即 $0\sim\text{FFFF FFFFH}$
15. **[C]**平均等待时间是磁盘转速 2 倍的倒数，所以转速提高一倍，平均等待时间缩小一半。
16. **[C]**
17. **[A]**周期挪用中挪用的就是访存周期。
18. **[C]**CPU 对通道和外设的查询状态、启动、关闭都是通过 I/O 指令完成的。
19. **[D]**效率最高的 I/O 方式就是通道。（如果有外围处理机就选外围处理机）
20. **[B]**基本概念。

二、填空

1. **(硬件系统) (软件系统)**
2. **(辅存) (主存)**
3. **(10101011) (11010101)**
4. **(10001111) (10111111)**
5. **(56H) (34H)**
6. **(2000H) (3000H)**
7. **(控制) (数据)**

8. (机器周期) (取指)
9. (超流水线) (超标量)
10. (30) (32)
11. (匹配 CPU 和主存之间的速度) (增加空间降低成本协调速度)
12. (256TB) (512M)
13. (中断响应) (中断处理)
14. (统一编址) (独立编址)
15. (统一的时钟信号) (握手应答方式)

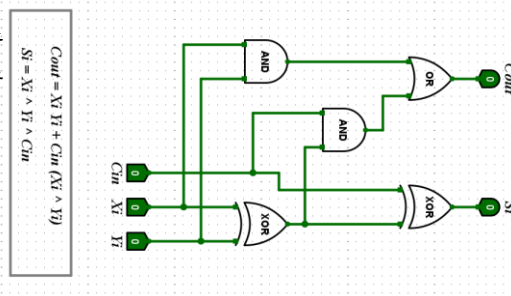
三、简答

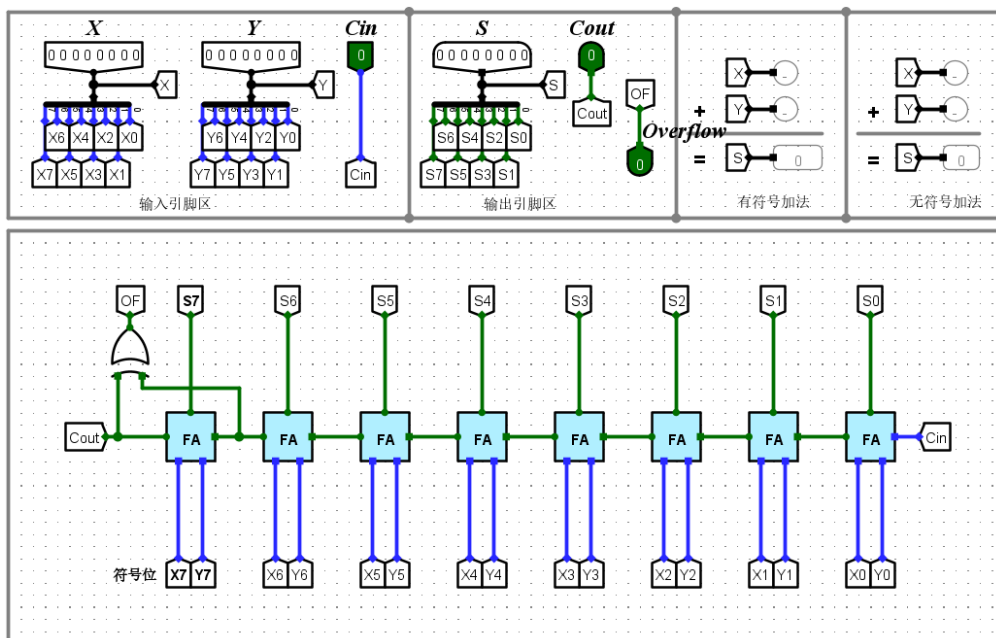
1. 二进制只有 0 和 1 两个数字，刚好和逻辑电路的高低电平对应，实现起来比较方便简单可靠，所以采用二进制编码信息。
2. 相同点：都要先产生转移的目的地址，然后修改 PC 值，使计算机在新的地址处取指执行。
不同点：调用指令和转移指令的区别在于子程序调用时需要保存返回地址。
返回地址不需要再指令中给出返回地址，因为在调用子程序时已经将返回地址存放在某个约定的地方，返回时就根据此处存放地址返回，约定地方包括：寄存器、子程序起始位置、栈顶等。
3. ①存取速度上寄存器非常快，主存相对较慢。②容量上寄存器一般很小，几 KB，而主存相对容量较大。③价格上寄存器很贵，主存相对较低。④寄存器一般在 CPU 内部，主存在 CPU 之外。⑤每个寄存器都有自己的名字，没有主存类似的地址编号。⑥寄存器多存放现行程序状态、中间值以及一些指令控制部件的信号，而主存主要存放整个程序的指令和数据。
4. Cache 对程序员来说是透明的。因为 Cache 存储器的地址变换和块替换均由硬件实现，程序员无法了解其工作过程。在执行第一个 IO 操作前，操作系统需要执行中断程序如保存断点、保存硬件的其他现场来保证存储访问的一致性。
5. 一种是组合逻辑，另一种是微程序。组合逻辑控制器速度更快。因为组合逻辑控制器由复杂的组合逻辑控制电路和一些触发器构成，速度取决于电路延迟，所以速度非常快。而微程序控制器采用了存储程序原理，所以每条指令都要从控制存储器中取一次，影响了速度。

四、应用

1. $[x]_{\text{补}} = 0,111\ 1111\ 1111\ 1110$, $[y]_{\text{补}} = 1,000\ 0000\ 0000\ 1010$
 - (1) 求和得 $10,111\ 1111\ 1111\ 0100$ ，符号位和最高位均产生进位，此时不会溢出，舍弃结果中最高位 1， $\text{sum}_{\text{补}} = 0,111\ 1111\ 1111\ 0100$ ，还原为十进制是 32756，正确。
 - (2) $[-y]_{\text{补}} = 0,000\ 0000\ 0000\ 1010$ ，与 $[x]_{\text{补}}$ 求和得 $10,000\ 0000\ 0000\ 1000$ ，数值最高位进位而符号位未进位，发生了溢出， $\text{sub}_{\text{补}} = 1,000\ 0000\ 0000\ 1000$ ，还原为十进制为 -32760，结果错误，验证溢出。
 - (3) 逻辑图如下

右为 1 位全加器
下为 8 位加法器

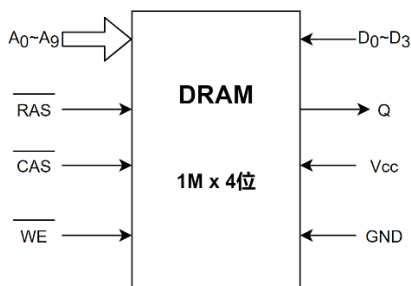




FA 单元为 1 位全加器。

2. 参考答案如下

- (1) DRAM 地址引脚复用，是 10 根；4 位芯片，数据引脚 4 根。
- (2) 引脚示意图如下



- (3) 按字节编址，所需芯片 $\lceil M/m \rceil \lceil N/n \rceil = \lceil 2^{24}/1M \rceil \lceil 8/4 \rceil = 16 \times 2 = 32$ 片
- (4) 存储器需要 24 位地址码，片选由高 4 位产生。
- (5) 32 片需要一级门电路 $\lceil 32/8 \rceil = 4$ 个，驱动中这 4 个一级门电路又需要 $\lceil 4/8 \rceil = 1$ 个二级门电路，一共需要 $4+1=5$ 个门

3. 参考答案如下

- (1) 未谈及位密度等，给了扇区信息，则磁盘容量指的是格式化容量，用扇区算。
容量 = $4 \times 20000 \times 2000 \times 512B \approx 76.29GB$
- (2) $\frac{1MB}{1KB} = 1024$ 个逻辑块
- (3) 一个逻辑块是 1KB，占 2 个扇区，这两个扇区是连续的。访问一个逻辑块的时间 = 平均旋转时间 + 平均等待时间 = $5ms + \frac{1}{2r} = 5ms + 3ms = 8ms$ 。
那么 1024 个逻辑块需要耗时 $8ms \times 1024 = 8192ms$ 。
- (4) 映射到同一磁道的连续扇区，从 0 扇区开始访问。则仅定位第一个扇区即可，直接寻道到该磁道，且无需查找扇区。只需 5ms 寻道时间。

4. 参考答案如下

- (1) 每次并行传输 4 个字节，总线宽度 32 位。

传输需要一个总线周期传地址，一个总线周期传数据，所以要 2 个总线周期

$$\text{总线带宽} = \frac{4\text{B}}{2 \div 66\text{MHz}} = 132\text{MB/s}。$$

- (2) 突发传送，仅传一次地址即可。32 个字节数据需要 8 个周期。一共 64 字节，也就是要 2 个总线周期传地址，16 个总线周期传数据，一共 18 个总线周期。

$$\text{总线带宽} = \frac{64\text{B}}{18 \div 66\text{MHz}} \approx 234.7\text{MB/s}$$

- (3) 总线带宽与总线宽度、工作频率、总线是否复用以及总线传输方式有关。

2011

一、选择

1. **[D]**二进制只有 0 和 1，刚好对应逻辑电路里的高低电平，实现起来可靠方便。
2. **[A]**IEEE754 中，阶码用移码表示，偏置 127。E=1 时最小， $1-127=-126$ ；E=254 最大， $254-127=127$
3. **[A]**
4. **[A]**统一编址下，用访存指令访问 I/O 端口，地址码可区分主存还是外设。
5. **[C]**常见程序控制类指令有分支、调用、跳跃、返回等指令。
6. **[D]**CISC 中访存指令不受限制，RISC 中只有 Load/Use 指令可以访存。
7. **[A]**时钟周期是计算机操作的最小单位时间。
8. **[D]**最终得到的操作数是 600，说明访问到的是寄存器，使用的是寄存器直接寻址方式。
9. **[D]**DRAM 芯片地址数据引脚都复用，地址引脚 12 根，数据引脚 4 根，共 16。
10. **[A]**DMA 的三种方法：CPU 停止访存、交替访问、周期挪用。
11. **[B]**字长 16 位，按字编址即按 2B 编址。存储容量 $8MB = \frac{8MB}{2B} = 4M$ ，寻址范围 $0 \sim 4M-1$ 。
12. **[A]**固定概念。
13. **[A]**主要就是速度快。
14. **[D]**磁阻磁头是按磁阻变化原理工作的。
15. **[A]**固定概念。

二、填空

1. (概念性结构和功能特性) (逻辑实现) (物理实现)
2. (单位时间内处理请求量) (用户发出一个请求，系统作出响应并给出结果的这段时间)
3. (相对寻址) (转移指令)
4. (12FCH) (3100H)
5. (可以同时出现) (不能同时出现) (相容微命令)
6. (同时性) (并发性)
7. (存储体) (寻址系统) (读写系统)
8. (独立编址) (统一编址)
9. (柱面号) (盘面号) (扇区号)
10. (形变型) (相变型) (磁光型)
11. (复用总线) (专用总线)
12. (字节多路通道) (数组多路通道) (选择通道)

三、简答

1. 第 1 层是微程序机器层，是一个实在的硬件层，它由机器硬件直接执行微指令。第 2 层是传统

机器语言层，也是一个实际的机器层，由微程序解释机器指令系统。第3层是操作系统层，它由操作系统程序实现。操作系统程序是由机器指令和广义指令组成的，这些广义指令是为了扩展机器功能而设置的，它是由操作系统定义和解释的软件指令组成，也称为混合层。第4层是汇编语言层，它为用户提供一种符号化的语言，借此可编写汇编语言源程序。这一层由汇编程序支持和执行。第5层是高级语言层，它是面向用户的，为方便用户编写应用程序而设置的，由各种高级语言编译程序支持和执行。高级语言层之上还有应用层，由实际解决问题的处理程序组成。

各层次之间关系紧密，下层是上层的基础，上层是下层的扩展。

2. 缺

3. 控制器的基本功能是对指令流和数据流实施正确的控制。由控制器形成受控部件的控制信号，控制器控制各种动作的执行顺序。

执行一条指令所需步骤：首先是取指令和分析指令。按照程序规定次序，从主存取出当前执行的指令，并送到控制器的 IR 中，对所取的指令进行分析，即根据指令中的操作码确定计算机应进行什么操作。其次是执行指令。根据指令分析结果，由控制器发出完成操作所需的一系列控制电位，以便指挥计算机有关部件完成这一操作，同时还为取下一条指令做好准备。

4. 相同点：都是计算机中的控制信息，计算机通过他们实现数值处理或非数值处理。不同点：指令由 CPU 执行并控制，通道命令字由通道控制并执行；指令所有计算机都由，通道命令字只有计算机配置通道才有；指令可以完成复杂的任务，复杂规整完备有效。通道命令字只规定通道及其连接外设的操作等，简单且具有局限性。

5. 单总线结构的缺点：①多设备竞争总线使用权，降低了计算机系统的效率。②多种速度差异很大的设备连接在一条总线上，整个系统性能会降低。③扩展设备的能力受限，计算机系统功能也受到限制。

可采用多总线结构或分成多级总线解决缺点。通过多条带宽不同的总线，将设备按速度分类接在不同的总线上。低速总线作为高速总线的一个设备工作，通过总线桥接器将不同速度的总线连接起来。可解决或缓解争夺使用权的矛盾，解决速度差异导致的低效工作，也解决了扩展能力受限的问题。

四、应用

1. 参考答案如下

- (1) R1 中内容 48488000H=0100 1000 0100 1000 1000 0000 0000 0000，真值为 1212710912
R2 中内容 C4C48000H=(1100 0100 1100 0100 1000 0000 0000 0000)₂※，真值-993755126
执行加法指令得 1 0000 1101 0000 1101 0000 0000 0000 0000，符号位和最高位均产生进位，此时不是溢出，将结果最高位 1 舍去，得 R3 中内容 0000 1101 0000 1101 0000 0000 0000 0000，真值是 218955776。

R1 执行前：1212710912，执行后：1212710912。

R2 执行前：-993755126，执行后：-993755126。

R3 执行前：0，执行后：218955776。

- (2) 按照 IEEE754 将寄存器中的数据还原为真值

R1: 0 10010000 1001000100……, $(1 + 2^{-1} + 2^{-4} + 2^{-8}) \times 2^{17}$, 真值为 20532

R2: 1 10001001 1000100100……, $-(1 + 2^{-1} + 2^{-5} + 2^{-8}) \times 2^{10}$, 真值为-1572
20532-1572=18960

R1 执行前：20532，执行后：20532

R2 执行前：-1572，执行后：-1572

R3 执行前：0，执行后：18960

2. 参考答案如下

- (1) 每条指令占 4 个存储单元, 字长为 32 位, 则每个存储单元是 8 位, 按字节编址。
(其实这样的解释难以自圆其说,CPU 字长 32 位并没有说是指令字长还是机器字长)
- (2) 该指令意思是 s3 寄存器中的值左移 2 位, 在二进制中意味着乘 4 操作。
- (3) 寄存器 t0 编号为 0, s6 编号为 14。在第三四句指令中均出现了 t0, 按照格式表可知编号为 0。在第二句出现了 s6, 按格式表可知是 14。
- (4) Exit 对应的地址应该是返回地址, 根据 JUMP 指令可知 Loop 起始为 20000, 则下一个地址是 20004。(逻辑上来看, 原先程序判 Loop 成功则进入子程序, 否则就会顺序执行, 所以 Loop 处紧接着就是 Exit 后)
- (5) 分支指令(不等转移)偏移量 8 位, 范围是 0~255; 跳转指令(跳转转移)目的地址是 24 位, 范围是 $0 \sim 2^{24} - 1$
- (6) 机器字长是 32 位, 根据 bne 指令, A[i] 的元素要和任意数据 k 进行比较, 可知元素占 4 字节。

3. 按字编址, 字长 64 位, 即按 8B 编址。

存储容量为 $2^{22} \times 8B = 2^{25}B = 32MB$

需要芯片 $\lceil M/m \rceil \lceil N/n \rceil = \lceil 2^{22}/128K \rceil \lceil 64/8 \rceil = 32 \times 8 = 256$ 片

用来做字扩展的片选译码地址码, 位数是 $\lceil \log_2 M/m \rceil = \log_2 32 = 5$ 位, 地址码的高 5 位。

4. 参考答案如下

- (1) 盘面数为 $6 \times 2 - 1 = 11$
柱面数位 $(12.9 - 6.1) \times 10 \div 2 \times 180 + 1 = 6121$
- (2) 非格式容量 = 内直径 $\times \pi \times$ 位密度 \times 盘面数 \times 柱面数 = $6.1 \times 10 \times 3 \times 5000 \times 11 \times 6121 \div 8 \approx 7.18GB$
- (3) 传输率 $f =$ 内直径 $\times \pi \times$ 位密度 \times 转速 = $6.1 \times 10 \times 3 \times 5000 \times \frac{7200}{60} \div 8 \approx 13.09MB/s$
- (4) 12 台磁盘机需要 4 位台号
6121 个柱面需要 13 位柱面号
1 个盘面需要 4 位盘面号
64 个扇区需要 6 位扇区号
地址方案为 [台号 4 位, 柱面号 13 位, 盘面号 4 位, 扇区号 6 位]

26……23	22……10	9……6	5……0
台号	柱面号	盘面号	扇区号

2010

一、选择

1. **[B]**基本概念。
2. **[A]**绿书相关章节。
3. **[B]**基本概念。
4. **[B]**执行阶段取出数据，取指阶段可能会取出地址（间址的时候）。
5. **[A]**变址寻址， $EA=6000H+2000H=8000H$ ， $8000H$ 处内容为 $0080H$ 。
6. **[B]**RISC 采用组合逻辑控制。
7. **[A]**指令周期按顺序分为取值周期、间址周期、执行周期和中断周期，故中断查询应该在最后。
8. **[A]**1 位符号位判断溢出的方法是：加数与被加数同号，结果与之异号则溢出。加数与被加数异号，不可能发生溢出。
9. **[B]**光盘、磁盘、磁带都只能按顺序访问。
10. **[A]**智能终端除了可进行数据输入输出，还具有数据处理功能，配备了处理机。
11. **[C]**SRAM 芯片地址引脚不复用，数据引脚复用。地址引脚是 22，数据引脚 4，共 26。
12. **[B]**速度：寄存器>Cache>主存>辅存。
13. **[C]**通道通过中断方式对 CPU 发出请求。
14. **[C]**统一编址时，使用访存指令进行输入输出。
15. **[B]**当 I/O 和 CPU 同时请求访存时，出现访存冲突，此时 CPU 暂时放弃总线占有权，由 I/O 设备挪用几个存储周期。

二、填空

1. **(存储程序概念)**
2. **(9002) (EFFE) (6FFE)**
3. **(11000110) (10000110) (11101111)**
4. **(机器数) (真值)**
5. **(数据总线) (控制总线) (地址总线)**
6. **(FF00) (F000) (FFF0) (FFFFFF00)**
7. **(标志)**
8. **(直接编码) (字段直接编码) (字段间接编码) (最短编码法)** 以最新的 MOOC 中的分类为主。
9. **(CPU 停止访存) (周期挪用) (交替访存)**
10. **(只读) (可擦写) (追记)**
11. **(留磁少) (导磁率高) (软磁)**

三、简答

1. 以二进制形式存放在主存。二进制只有 0 和 1 两个数字，刚好和逻辑电路的高低电平对应，实现起来比较方便简单可靠，所以采用二进制编码信息。计算机通过指令周期的不同阶段来区分指令和数据。一条指令通常分为取指阶段和执行阶段等。在取指阶段访存取出指令，在执行阶段访存取出操作数。
2. 相同点：都要先产生转移的目的地址，然后修改 PC 值，使计算机在新的地址处取指执行。
不同点：调用指令和转移指令的区别在于子程序调用时需要保存返回地址。
返回地址不需要再指令中给出返回地址，因为在调用子程序时已经将返回地址存放在某个约定的地方，返回时就根据这个地方存放的地址返回，约定地方包括：寄存器、子程序起始位置、栈顶等。
3. 程序计数器 PC 是控制器的组成部分。用于指出下一条指令在主存中的存放地址，CPU 就是根据 PC 的值作为地址访问存储器取得指令。由于程序指令大部分顺序执行，故 PC 有自增功能。某些情况下可以对 PC 进行赋值完成转移、子程序调用返回等程序浮动功能。
4. 响应中断的条件有 4 个：①有中断请求。②CPU 的一条指令执行完毕。③CPU 允许响应中断。④无更紧迫的任务或事件。
该做的工作有：执行中断隐指令，具体包含关中断、保存断点、引出中断服务程序。
5. ①集中式仲裁：设置集中式的仲裁电路，连接所有总线设备并按照某种策略选择其中一个总线主设备获得总线使用权。优点：仲裁过程以及总线设备接口简单。缺点：仲裁电路出现故障将导致整个系统瘫痪；扩展设备需要对仲裁电路进行较大的修改，难度较大。②分布式仲裁：所有主设备均设置自己的仲裁电路。当主设备发出请求时，各仲裁电路按照一定策略共同决定总线使用权。优点：线路可靠性高，扩展灵活，设备接插比较随意。缺点：确定总线主设备是否在正常工作，系统需要进行超时判断；由于每个主设备需要在其接口设计仲裁电路，导致设计的复杂性加大。

四、应用

1. 将运算的数据以二进制补码形式表示：
 $[A]_{\text{补}} = ([E_A]_{\text{补}} ; [M_A]_{\text{补}}) = (11, 100 ; 11.0011)$
 $[B]_{\text{补}} = ([E_B]_{\text{补}} ; [M_B]_{\text{补}}) = (11, 101 ; 00.1011)$
 求阶差， $[E_{A-B}]_{\text{补}} = [E_A]_{\text{补}} - [E_B]_{\text{补}} = [E_A]_{\text{补}} + [-E_B]_{\text{补}} = 11.100 + 00.011 = 11.111$ ，也就是-1，可知 B 的阶码较大，比 A 大 1。
 对阶，A 的阶向 B 看齐，尾数右移损失精度时采用 0 舍 1 入法。
 $[A']_{\text{补}} = ([E'_A]_{\text{补}} ; [M'_A]_{\text{补}}) = (11, 101 ; 11.1010)$
 尾数相加， $[M'_A]_{\text{补}} + [M_B]_{\text{补}} = 11.1010 + 00.1011 = 00.0101$
 此时 $[A + B]_{\text{补}} = 11, 101 ; 00.0101$
 最后进行规格化，由于尾数两符号位相同，尾数两符号位相同无溢出，但需规格化，所以
 $[A + B]_{\text{补}} = 11, 010 ; 00.1010$
2. $[A]_{\text{原}} = 0.0101$ ， $[B]_{\text{原}} = 1.1101$ ， $A* = 0.0101$ ， $B* = 0.1101$

部分积	乘数	说明
0.0000	1101	初始部分积置 0
+0.0101		乘数为 1，加 A*
0.0101		
0.0010	1 110	右移 1 位
0.0010	1 110	乘数为 0，加 0
0.0001	01 11	右移 1 位

+0.0101	01	11	乘数为 1, 加 A*
0.0110	01	11	
0.0011	001	1	右移 1 位
+0.0101	001	1	乘数为 1, 加 A*
0.1000	001		
0.0100	0001		右移 1 位

则 $A * \times B * = 0.01000001$

符号位为 $A_S \oplus B_S = 0 \oplus 1 = 1$, 则 $[A \times B]_{\text{原}} = 1.01000001$

3. 参考答案如下

- (1) 指令字长 32 位, 32 条指令需要 5 位 op 字段, 32 个通用寄存器需要 5 位 Ri 字段, 三种寻址方式需要 2 位, 剩余 20 位做形式地址。格式如下

5	5	2	20
操作码 op	寄存器编号 Ri	寻址方式 X	形式地址 A

- (2) 形式地址 20 位, 故直接寻址空间为 $-2^{19} \sim 2^{19} - 1$

寄存器 32 位宽, 寄存器间接寻址空间为 $-2^{31} \sim 2^{31} - 1$

4. 地址线 20 根, 按字编址, 字长 32 位, 则存储器容量为 $2^{20} \times 32\text{bit} = 2^{20} \times 4\text{B} = 4\text{MB}$ 。

需要芯片 $\lceil M/m \rceil \lceil N/n \rceil = \lceil 2^{20}/32\text{K} \rceil \lceil 32/8 \rceil = 32 \times 4 = 128$ 片

用来做字扩展的就是片选译码地址码, 位数是 $\lceil \log_2 M/m \rceil = \log_2 32 = 5$ 位, 地址码的高 5 位。

5. 参考答案如下

- (1) 盘面数为 $4 \times 2 - 1 = 7$

柱面数为 $(8.9 - 4.3) \times 10 \div 2 \times 100 + 1 = 2301$

- (2) 非格式容量 = 内直径 $\times \pi \times$ 位密度 \times 盘面数 \times 柱面数 = $4.3 \times 10 \times \pi \times 5000 \times 7 \times 2301 \div 8 \approx 1.21\text{GB}$

- (3) 传输率 $f =$ 内直径 $\times \pi \times$ 位密度 \times 转速 = $4.3 \times 10 \times 3 \times 5000 \times \frac{7200}{60} \div 8 \approx 9.23\text{MB/s}$

- (4) 12 台磁盘机需要 4 位台号

2301 个柱面需要 12 位柱面号

7 个盘面需要 3 位盘面号

64 个扇区需要 6 位扇区号

地址方案为 [台号 4 位, 柱面号 12 位, 盘面号 3 位, 扇区号 6 位]

24.....21	20.....9	8.....6	5.....0
台号	柱面号	盘面号	扇区号

2009

2009 年采用全国统考 408 试卷，无自命题

2008

一、选择

1. **[A]**
2. **[B]**变址间址寻址方式也就是先变址后间址的寻址方式。先变址，有效地址 $EA'=(X)+A$ ；后间址，有效地址 $EA=(EA')=((X)+A)$ 。
3. **[C]**基本常识。
4. **[B]**基本常识。
5. **[C]**一组实现一定功能的微命令构成一条微指令，有序微指令序列构成一段微程序，微程序的作用是实现一条对应的机器指令。
6. **[D]**01 正溢，10 负溢。
7. **[C]**采用最短字长编码，需要保留一个编码表示空操作，实际操作个数为 33 个。需要 6 位。
8. **[B]**由字面即可知道基址是不变的，变址是可变的。
9. **[A]**RAM 大多易失性，ROM 一般为非易失。
10. **[B]**MROM 为掩膜式只读存储器，其他均可多次写入。
11. **[B]**软磁特性、饱和磁感应强度大、导磁率高、剩磁小、高频特性好、硬度大、居里点高。
12. **[C]**某一时刻至多有一个总线主设备，但会有多个从设备。
13. **[C]**程序查询控制是效率最低的。
14. **[A]**统一编址采用访存指令，靠地址码区分。独立编址则有专门的 I/O 指令。
15. **[B]**CPU 对通道状态查询、启动和关闭都是通过 I/O 指令。

二、填空

1. **(32) (8) (24)**
2. **(1000 0111 1101 1000) (1111 1000 0010 1000) (0111 1000 0010 1000)**
3. **(0010 0011) (1001 0100) (1011 0111)**
4. **(加法器)**
5. **(转移) (子程序调用) (返回)**
6. **(微命令) (微操作)**
7. **(有中断请求) (CPU 允许响应) (CPU 的一条指令结束)**
8. **(两种稳定状态) (方便检测所处状态) (两种状态相互容易转换)**
9. **(串行) (并行)**
10. **(输出) (外存储器) (数模转换设备)**
11. **(硬件电路) (软件程序)**
12. **(螺线型) (线)**

三、 简答

- ①寄存器-寄存器指令系统中,运算指令不含存储器型操作数。机器在执行这类指令的过程中,只对寄存器中操作数进行操作,从寄存器取操作数,结果放到寄存器,不访存速度快。
②存储器-存储器指令系统中,部分运算指令的操作数都存放在存储器中,显然两个以上的操作数存放在存储器中,使得指令的执行需要4次以上的访存,即1次取指、2次读操作数、1次写操作数,指令执行速度很慢。
③寄存器-存储器指令的速度介于上述二者之间。
- 溢出是指运算结果超过了数据的表示范围。通常大于机器所能表示的最大正数为上溢,小于机器所能表示的数为下溢。补码定点乘法会出现溢出。若参加运算的两个数均为定点小数,即 $|A| \leq 1$, $|B| \leq 1$ 时, $|C| = |A| \times |B| \leq 1$ 。对于补码数据表示来说,当 $A=-1$, $B=-1$ 时, $C=1$ 这是唯一溢出的情况。
- 组合逻辑控制器速度更快。因为组合逻辑控制器由复杂的组合逻辑控制电路和一些触发器构成,速度取决于电路延迟,所以速度非常快。而微程序控制器采用了存储程序原理,所以每条指令都要从控制存储器中取一次,影响了速度。
- DMA 控制器试通过专门设计的硬件控制逻辑来实现对数据传送的控制。而通道实际上是一个具有特殊功能的处理器,有自己的指令和程序,通过执行一个通道程序实现对数据传送的控制,具有更强的独立处理数据输入输出能力。DMA 控制器通常仅能控制一台或少数几台同类设备,而通道可以同时控制许多台同类或不同类设备。
- 从规模上可分为片内总线、系统总线、设备总线、通信总线。片内总线:芯片内部设置的总线,用于完成芯片内部各功能单元的信息交换。系统总线:用于计算机内部各大功能部件的信息传输,因为这些部件都各自制作在不同的插板上,也成为板级总线。设备总线:又称为 IO 总线,用于连接各种外部设备。通信总线:又称为外部总线,用于完成计算机间、计算机与其他系统之间的信息通信。

四、 应用

- 首先将参加运算的数据二进制表示
 $[A]_{\text{补}} = ([E_A]_{\text{补}} ; [M_A]_{\text{补}}) = (11, 100 ; 11.1011)$
 $[B]_{\text{补}} = ([E_B]_{\text{补}} ; [M_B]_{\text{补}}) = (11, 010 ; 00.1011)$
 阶码相加, $[E_C]_{\text{补}} = [E_A]_{\text{补}} + [E_B]_{\text{补}} = 00, 010$
 尾数相乘, $[M_C]_{\text{补}} = [M_A]_{\text{补}} \times [M_B]_{\text{补}}$
 运算过程如下:
 $[M_A]_{\text{补}} = 11.1001$, $[-M_A]_{\text{补}} = 00.0111$, $[M_B]_{\text{补}} = 00.1101$

	部分积	乘数
	00.0000	0.11010
加 $[-M_A]_{\text{补}}$	+00.0111	0.11010
	00.0111	0.11010
右移	00.0011	1 0.1101
加 $[M_A]_{\text{补}}$	+11.1001	1 0.1101
	11.1100	1 0.1101
右移	11.1110	01 0.110
加 $[-M_A]_{\text{补}}$	+00.0111	01 0.110
	00.0101	01 0.110

右移	00.0010	101	0.11
加 0 右移	00.0001	0101	0.1
加 $[M_A]_{\text{补}}$	+11.1001	0101	0.1
	11.1010	0101	

$[M_C]_{\text{补}} = 11.10100101$, 需要左移一位进行规格化得 $[M_C]_{\text{补}} = 11.0100101$, 阶码减 1 得 $[E_C]_{\text{补}} = 00.001$, 采用 0 舍 1 入法进行舍入得 $[M_C]_{\text{补}} = 11.0101$

最终 $[C]_{\text{补}} = 00.001 ; 11.0101$

2. $[A]_{\text{补}} = 0.01011$, $[B]_{\text{补}} = 11.00101$, $[-B]_{\text{补}} = 00.11011$

余数	被除数	商	说明
	00.01011	0.00000	初始商置 0
	+11.00101	0.00000	异号, 加 $[B]_{\text{补}}$
$[R_1]_{\text{补}}$	11.10000	0.00001	余数除数同号商 1
	11.10000	0.00010	左移 1 位
	+00.11011	0.00010	加 $[-B]_{\text{补}}$
$[R_2]_{\text{补}}$	11.11011	0.00011	余数除数同号商 1
	11.10110	0.00110	左移 1 位
	+00.11011	0.00110	加 $[-B]_{\text{补}}$
$[R_3]_{\text{补}}$	00.10001	0.00110	余数除数异号商 0
	01.00010	0.01100	左移 1 位
	+11.00101	0.01100	加 $[B]_{\text{补}}$
$[R_4]_{\text{补}}$	00.00111	0.01100	余数除数异号商 0
	00.01110	0.11000	左移 1 位
	+11.00101	0.11000	加 $[B]_{\text{补}}$
$[R_5]_{\text{补}}$	11.10011	0.11001	余数除数同号商 1
	11.00110	1.10010	左移 1 位
	+00.11011	1.10010	加 $[-B]_{\text{补}}$
	00.00001	1.10010	

由于未除尽, 商为负, 需要修正商,

$[C]_{\text{补}} = 1.10010 + 0.00001 = 1.10011$

由于 $C_0=1$ 且余数与除数异号, 无需恢复余数

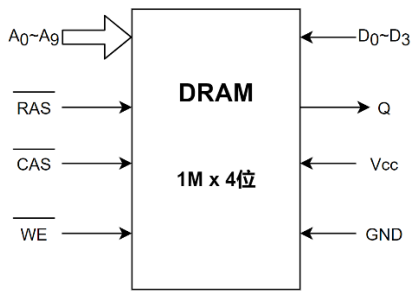
则 $[2^{-5}R_5]_{\text{补}} = 0.0000000001$

3. 参考答案如下

- (1) 子程序调用之前, PC 存放该指令的地址, 内容为 3000H; SP 为 200H; 栈顶内容 2000H。
- (2) 子程序调用执行之后, 堆栈保存返回地址, $(SP)=PC+1$, 栈顶内容是 3002H; 堆栈向上生长, $SP=SP+1=202H$; PC 内容为 1000H, 即子程序起始处。
- (3) 返回后, 取出栈顶保存内容, $SP=SP-1=200H$; 栈顶内容为 2000H; PC 为即将执行的指令地址 3002H。

4. 参考答案如下

- (1) $1M \times 4\text{bit}$ 的 DRAM 存储芯片, 地址引脚复用, 所以地址引脚是 10 根, 数据引脚 4 根。
- (2) 示意图如下 (注意 DRAM 芯片地址引脚复用, 要少画一半)



- (3) 需要 $\lceil M/m \rceil \lceil N/n \rceil = \lceil 8M/1M \rceil \lceil 64/4 \rceil = 8 \times 16 = 128$ 片。
 (4) $8M = 2^{23} \rightarrow 23$ 位地址码，其中片选需要 3 位。
 (5) 128 片需要一级门电路 $\lceil 128/8 \rceil = 16$ 个，驱动中这 16 个一级门电路又需要 $\lceil 16/8 \rceil = 2$ 个二级门电路，一共需要 $16+2=18$ 个门。

5. 参考答案如下

- (1) 盘面数位 $4 \times 2 - 1 = 7$
 柱面数为 $(13.2 - 6.2) \times 10 \div 2 \times 200 + 1 = 7001$
 (2) 非格式容量 = 内直径 $\times \pi \times$ 位密度 \times 盘面数 \times 柱面数 = $6.2 \times 10 \times 3 \times 6000 \times 7 \times 7001 \div 8 \approx 6.37GB$
 (3) 传输率 $f =$ 内直径 $\times \pi \times$ 位密度 \times 转速 = $6.2 \times 10 \times 3 \times 6000 \times \frac{7200}{60} \div 8 \approx 1.33MB/s$
 (4) 6 台磁盘机需要 3 位台号
 7001 个柱面需要 13 位柱面号
 7 个盘面需要 3 位盘面号
 32 个扇区需要 5 位扇区号
 地址方案为 [台号 3 位, 柱面号 13 位, 盘面号 3 位, 扇区号 5 位]

23……21	20……8	7……5	4……0
台号	柱面号	盘面号	扇区号

6. 提高运算器速度的方法：①运算电路特别是加法和移位用逻辑电路高速化
 提高存储器速度的方法：①采用多级存储系统，Cache-主存系统，可改善主存和 CPU 速度不匹配的问题。②采用多模块存储器，可以提高存储器带宽，加快存取速度。

2007

一、选择

1. **[B]**此题要熟悉指令取指和译码阶段的数据通路。
2. **[C]**变址相对寻址就是先变址后相对的寻址方式。先变址，有效地址 $EA'=(X)+A$ ；后相对，有效地址 $EA=EA'+PC=(X)+A+PC$ 。
3. **[D]**
4. **[B]**取指完成后 $PC+1$ ，存放下一条指令的地址。
5. **[C]**控制存储器用于存放微程序，在 CPU 内部，用 ROM 实现。
6. **[D]**
7. **[D]**
8. **[C]**符号位的第一位为正确的符号，第二位为运算得到的符号。负溢为：应该为负但所得为正。
9. **[C]**总线宽度又称总线位宽，通常指数据总线的根数，表示总线上能同时传输的数据位数。
10. **[A]**按字编址即按 4B 编址， $\frac{512MB}{4B} = 128M$ ，则寻址范围为 $0 \sim 128M-1$ 。
11. **[D]**通道是一个具有输入输出功能的处理器，独立于 CPU，不需要 CPU 的参与，最高效。
12. **[B]**通道程序是通道控制字的有序集合。
13. **[B]**双极型存储芯片，地址引脚和数据引脚都不复用，所以地址引脚 14 根，数据引脚 8 根。
14. **[C]**总线从设备是指被主设备访问的设备；总线主设备是能够申请并获得总线使用权的设备；总线源设备是发送数据的设备；总线目标设备是接收数据的设备。
15. **[D]**

二、填空

1. (取指) (译码) (执行)
2. (100 0000 0011) (000 0000 0011)
3. (1101 1110) (1001 0100)
4. $(2^{127} \times (1 - 2^{-23})) (-2^{-128} \times (2^{-1} + 2^{-23}))$
5. (累加器型) (堆栈型) (通用寄存器型)
6. (阶码相加) (尾数相乘) (规格化和舍入)
7. (全 1 熔断丝型) (全 0 肖基特二极管型)
8. (引入中断服务程序) (保存断点)
9. (容量) (速度) (多级)
10. (双端口存储器) (多模块存储器)
11. (可逆相变型介质) (磁光型介质)
12. (并行接口) (串行接口)
13. (固态) (动态)

三、 简答

1. 程序控制类指令的功能是改变程序执行的顺序。常见的控制类指令有：条件转移指令、无条件转移指令、子程序调用指令、返回指令、循环指令、跳跃指令。
2. 指令周期是指 CPU 从主存取出一条指令并执行所需要的全部时间。
时钟周期是计算机操作的最小单位时间，大小是 CPU 主频的倒数。
存储周期指连续两次启动存储器进行存取操作所需要的最小时间间隔。
指令周期常常包括若干机器周期，机器周期包含若干时钟周期。存储周期也包括若干时钟周期。指令周期里的取指、间址、中断和执行等阶段有时需要访存，即占用若干个存储周期。三者时间大小关系为时钟周期<存储周期<指令周期。
3. ①直接控制编码：微指令的微命令字段的每一位代表一个微命令，对应位设置 0 或 1 就代表选用或不选用某个微命令。无需进行译码，优点是简单、直观、执行速度快，操作并行性好。缺点是微指令字长过长， n 个微命令就要求微指令的操作字段有 n 位，所需控制存储器容量极大。
②分段直接编码：将指令的微命令字段分成若干小字段，把互斥性微命令组合在同一字段中，把相容性微命令结合在不同字段中，每个字段独立编码，每种编码代表一个微命令且各字段编码单独定义。这种方式可以缩短微指令字长，但要通过译码电路后再发出微命令，比直接控制编码慢。
③分段间接编码：一个字段的某些微命令需要另一个字段的某些微命令来解释，由于不是靠字段直接译码发出的微命令，故称为分段间接编码，又称隐式编码。这种方式可进一步缩短微指令字长，但因为削弱了微指令的并行控制能力，因此通常作为直接编码的一种辅助手段。
4. ①同步总线：总线系统有一个共所有设备使用的统一时钟，设备之间都按照约定俗成的时序进行操作和完成信息交换。同步总线的所有信号命令都是与总线时钟同步的，所以总线的使用设计调试都较为简单。所有设备的总线操作所需要的总线时钟都是整数，且总线操作周期需设计得满足最慢设备使用，从而使高速设备不能高效使用总线，造成时间浪费。
②异步总线：总线系统没有统一的时钟，任何一件事发生都只能是前一个或一些时间的结束，所有设备都以“握手”的方式进行联系从而完成信息交换。总线对设备速度差异的适应性更强，慢速设备对快速总线的影响仅限于本次总线操作，不影响下一次快速总线设备操作，效率高。异步总线无统一的时钟，设计控制调试都比较复杂。
5. 由接口寄存器、中断传送标准接口、DMA 控制逻辑三部分组成。
(一) DMA 控制逻辑：控制 DMA 的数据传送过程。由单位数据准备就绪、DMA 请求和响应等多个触发器，DMA 优先排队、DMA 向 CPU 发请求及回送命令等线路构成。
(二) 中断传输标准接口：一是当 WC 产生溢出说明一批数据传输完成或传输出现故障，由它向 CPU 发出中断请求，以便 CPU 完成 DMA 传送的结束处理；二是 DMA 传送的所有准备工作也借助它实现，如查询、启停外设、设置参数等。
(三) 接口寄存器：①数据缓冲寄存器 DBR：寄存要传送的单位数据。②地址缓冲寄存器 ABR：寄存主存缓冲区开始地址。③字计数器 WC：寄存传输长度，即要交换的单位数据个数。④状态寄存器 DSR：寄存设备及接口状态。⑤设备控制寄存器 DCR：寄存 CPU 启动接口及设备有关的控制命令。⑥设备内部地址寄存器 DAR：寄存设备内部地址。

四、 应用

1. 首先将参加运算中所有数据的二进制表示出来。

$$[A]_{补} = ([E_A]_{补} ; [M_A]_{补}) = (11, 100 ; 11.1011)$$

$$[B]_{补} = ([E_B]_{补} ; [M_B]_{补}) = (11, 010 ; 00.1011)$$

$$[-E_B]_{\text{补}} = 00,110$$

第二步求阶差： $[E_{A-B}] = [E_A]_{\text{补}} - [E_B]_{\text{补}} = [E_A]_{\text{补}} + [-E_B]_{\text{补}} = 11,100 + 00,110 = 00,010$ ，即 A 的阶码较大，比 B 的阶码大 2。

然后是对阶，阶码小的数 B 向阶码大的数 A 看齐，由于尾数右移时会损失精度，采用“0 舍 1 入”法， $[B']_{\text{补}} = ([E'_B]_{\text{补}} ; [M'_B]_{\text{补}}) = (11,100 ; 00.0011)$ 。（不舍入的尾数结果是 00.0010，由于舍掉的是 11，则 0 舍 1 入，为 00.0011）

接着尾数相加， $[M_A]_{\text{补}} + [M'_B]_{\text{补}} = 11.1011 + 00.0011 = 11.1110$

此时 $[A+B]_{\text{补}} = (11,100 ; 11.1110)$

最后是规格化，尾数左规三次，阶码减 3

所以 $[A+B]_{\text{补}} = (11,001 ; 11.0000) = -2^{-7}$ 。

2. 参考答案如下

(1) 设被乘数 $[A]_{\text{补}} = A_0.A_1A_2 \dots A_n$ ，乘数 $[B]_{\text{补}} = B_0.B_1B_2 \dots B_n$ ，其中 A_0 和 B_0 为符号位。

根据已知，补码一位乘法的统一公式为：

$$\begin{aligned} [C]_{\text{补}} &= [A \times B]_{\text{补}} = [A]_{\text{补}} \times (0.B_1B_2 \dots B_n) + [-A]_{\text{补}} \times B_0 \\ &= [A]_{\text{补}} \times (0.B_1B_2 \dots B_n) - [A]_{\text{补}} \times B_0 \\ &= [A]_{\text{补}} \times (-B_0 + B_12^{-1} + B_22^{-2} + \dots + B_n2^{-n}) \\ &= [A]_{\text{补}} \times [-B_0 + (B_1 - B_12^{-1}) + (B_2 - B_22^{-2}) + \dots + (B_n2^{-(n-1)} + B_n2^{-n})] \\ &= [A]_{\text{补}} \times [(B_1 - B_0) + (B_2 - B_1)2^{-1} + \dots + (B_{n+1} - B_n)2^{-n}] \end{aligned}$$

其中 $B_{n+1} = 0$

设 Z_i 表示第 i 步部分积，可得递推公式如下：

$$[Z_0]_{\text{补}} = 0$$

$$[Z_1]_{\text{补}} = 2^{-1} \{ [Z_0]_{\text{补}} + (B_{n+1} - B_n)[A]_{\text{补}} \}$$

$$[Z_2]_{\text{补}} = 2^{-1} \{ [Z_1]_{\text{补}} + (B_n - B_{n-1})[A]_{\text{补}} \}$$

... ..

$$[Z_i]_{\text{补}} = 2^{-1} \{ [Z_{i-1}]_{\text{补}} + (B_{n-i+2} - B_{n-i+1})[A]_{\text{补}} \}$$

... ..

$$[Z_n]_{\text{补}} = 2^{-1} \{ [Z_{n-1}]_{\text{补}} + (B_2 - B_1)[A]_{\text{补}} \}$$

$$[A \times B]_{\text{补}} = [Z_{n+1}]_{\text{补}} = [Z_n]_{\text{补}} + (B_1 - B_0)[A]_{\text{补}}$$

由此可见，开始时 $B_{n+1} = 0$ ，部分积初值 $[Z_0]_{\text{补}} = 0$ ，每一步乘法由 $(B_{i+1} - B_i)(i = 1, 2, \dots, n)$ 决定原部分积加上 $[A]_{\text{补}}$ 或加 $[-A]_{\text{补}}$ 或加 0，再右移一位得到新的部分积，以此重复 n 步。第 $n+1$ 步由 $(B_1 - B_0)$ 决定原部分积加上 $[A]_{\text{补}}$ 或 $[-A]_{\text{补}}$ 或 0，但不移位，即得 $[A \times B]_{\text{补}}$ 。

(2) $[A]_{\text{补}} = 0.11011$ ， $[-A]_{\text{补}} = 1.00101$ ， $[B]_{\text{补}} = 1.00111$

运算过程如下：

	部分积	乘数
	00.00000	1.001110
加 $[-A]_{\text{补}}$	+11.00101	1.001110
	11.00101	1.001110
右移	11.10010	1 1.00111
加 0 右移	11.11001	01 1.0011
加 0 右移	11.11100	101 1.001

加[A] _补	+00.11011	101	1.001
	00.10111	101	1.001
右移	00.01011	1101	1.00
加 0 右移	00.00101	11101	1.0
加[-A] _补	+11.00101		
	11.01010	11101	

所以 $[A \times B]_{补} = 11.0101011101$

3. 参考答案如下

- (1) 24 根地址线有 $16M$ 个地址，字长 32 位，按字编址，容量为 $16M \times 32bit \div 8 = 64MB$ 。
- (2) 所需 SRAM 存储芯片 $\lceil M/m \rceil \lceil N/n \rceil = 256$ 片。
- (3) 存储器地址码位数： $\lceil \log_2 16M \rceil = 24$ 位。
片选译码的地址码位数： $\lceil \log_2 M/m \rceil = \lceil \log_2 32 \rceil = 5$ 位。
- (4) 256 片需要一级门电路 $\lceil 256/8 \rceil = 32$ 个，驱动中这 32 个一级门电路又需要 $\lceil 32/8 \rceil = 4$ 个二级门电路，一共需要 $32+4=36$ 个门。

4. 参考答案如下

- (1) 盘面数为 $2 \times 6 - 1 = 11$ 面，去掉一个伺服面
柱面数为 $(12.9 - 6.1) \div 2 \times 10 \times 220 + 1 = 7481$
- (2) 由于未谈及扇区信息，则默认非格式化容量
则容量 $C_n = \text{内直径} \times \pi \times \text{位密度} \times \text{柱面数} \times \text{盘面数} = 6.1 \times 10 \times 3 \times 6000 \times 7481 \times 11 \div 8 \approx 10.52GB$ （注意，容量的换算是 1024，通信传输的换算是 1000）
- (3) 传输率 $f = \text{内直径} \times \pi \times \text{位密度} \times \text{转速} = 6.1 \times 10 \times 3 \times 6000 \times \frac{7200}{60} \div 8 \approx 15.7MB/s$
- (4) 读取 6000 字节数据的平均时间： $t_{总} = t_s + t_w + t_{rw}$ ，其中 $t_s + t_w = 10 + \frac{1}{120} \times \frac{1}{2} \times 1000$ ，
而 $t_{rw} = \frac{8000}{6.1 \times 10 \times 6000 \times 3 \div 8} \times \frac{1}{120} \times 1000$ ，则 $t_{总} \approx 19.03ms$ 。

5. 参考答案如下

(1) 屏蔽码如下

中断源	屏蔽码			
1	1	1	1	1
2	0	1	1	1
3	0	0	1	1
4	0	0	0	1

(2) 屏蔽码如下

中断源	屏蔽码			
1	1	1	1	1
2	0	1	0	0
3	0	1	1	1
4	0	1	0	1

6. ①改进计算机的体系结构，如采用并行处理、流水线技术。
- ②运算电路特别是加法和移位用逻辑电路的高速化。
- ③运算方法和逻辑结构的高速化，如进位信号的快速处理。
- ④存储系统的优化改进，如采用多模块存储器、多级 Cache 等。
- ⑤选择高速的外围设备使之与 CPU 速度匹配，选择高效的 I/O 方式。

2006

一、选择

1. **[D]**
2. **[D]**间址相对寻址方式也就是先间址后相对的寻址方式。先间址，有效地址 $EA'=(A)$ ；后相对，有效地址 $EA=EA'+(PC)=(A)+(PC)$ 。
3. **[C]**符号位参与位移，右移时其左端空出的位补入“符号位”。
4. **[B]**浮点数据由阶码和尾数构成，阶码的位数决定了数据的范围大小，尾数的位数决定了数据的精度。阶码位数和尾数位数之和为数据字长，固定字长的情况下，阶码位数多则尾数位数少，精度就低。
5. **[A]**
6. **[A]** $[X]_{补}$ 的符号位是 1，所以 x 是负数。 $[-1/2]_{补} = 1.1000$ ，采用补码表示，如果符号位相同，则数值越大，码值越大。所以要使 $x > \frac{-1}{2}$ 成立， X_1 必须为 1，且 $X_2X_3X_4X_5$ 至少有一个为 1。
7. **[B]**内存储器主要用于存放指令和数据；控制存储器存放指令对应的微程序；通用寄存器存放指令执行过程中所需的操作数和操作数地址信息；指令寄存器存放当前正在执行的指令。
8. **[C]**控制器由指令控制部件、地址形成部件和时序部件构成，其中指令控制部件又包括程序计数器 PC、指令寄存器 IR 和指令译码器 ID。C 不是控制器组成部分。
9. **[B]**
10. **[B]**按字编址，即按 8B 编址， $\frac{512MB}{8B} = 64M$ ，则寻址范围是 $0 \sim 64M-1$ 。
11. **[A]**不可屏蔽中断，即必须立即响应、不能回避和禁止的中断。如掉电中断。
12. **[B]**周期挪用 DMA 传送方式，每当设备准备好一个单位数据，CPU 仅暂停访问存储器一个访问周期，DMA 只用这个访问周期传送这个单位数据，传送完后又将访问控制权还给 CPU。
13. **[C]**程序中中断控制传送方式是设备准备好一个单位数据后，通过 CPU 运行中断服务程序实现单位数据传送。每传送一个单位数据，CPU 就运行一次中断服务程序，且数据传送的路径是从设备到 CPU 再到主存，或者方向相反，所以速度慢。而 DMA 是为克服程序中中断控制传送通过程序和数据传送的路径需通过 CPU 再送到主存这两个缺点而提出的。
14. **[C]**总线从设备是指被主设备访问的设备；总线主设备是能够申请并获得总线使用权的设备；总线源设备是发送数据的设备；总线目标设备是接收数据的设备。
15. **[B]**单总线结构的计算机系统中，外设和主存的存储单元是统一编址，所以可不专设 I/O 指令。

二、填空

1. **(地址寄存器) (变址寄存器) (加法器)**
2. **(1, 0101 1011) (0, 0101 1011)**
3. **(1000 1111) (1011 1111)**
4. **$(2^{-1} \times 2^{-128}) (-1 \times 2^{127})$**
5. **(缩短指令长度) (扩大寻址空间) (提高编程灵活性)**

6. (尾数调整) (阶码相减) (尾数相除)
7. (存储地址寄存器) (地址驱动器) (地址译码器)
8. (只读) (螺旋线) (形变)
9. (主存和 CPU 速度不匹配) (硬件) (容量和成本) (软件)
10. (片内总线) (系统总线) (I/O 总线)
11. (字节多路) (数组多路)

三、简答

1. 根据 PC 访问主存得到的数据被送往 IR, 可知从主存得到的数据是指令。从指令执行周期的角度来说, 控制器的取指周期取出的一定是控制信息, 其他周期取出的是数据信息。
2. 寄存器-寄存器指令系统中, 运算指令不含存储器型操作数。机器在执行这类指令的过程中, 只对寄存器中操作数进行操作, 从寄存器取操作数, 结果放到寄存器, 不访存速度快。
存储器-存储器指令系统中, 部分运算指令的操作数都存放在存储器中, 显然两个以上的操作数存放在存储器中, 使得指令的执行需要 4 次以上的访存, 即 1 次取指、2 次读操作数、1 次写操作数, 指令执行速度很慢。
3. ①明确设计环境, 包括指令系统、主存存储周期、时钟周期、部件结构等。②分析指令流程, 根据指令功能要求和部件间的数据通路, 为各种指令的微操作序列选定适当的节拍信号。③编制指令操作时间表, 不断安排、调整、完善微操作的时序、控制器的结构和数据通路的设置。④形成微操作表, 便于清洗有条理的进行微操作信号的综合。⑤进行微操作的逻辑综合, 形成对每个微操作对应的逻辑表达式及微操作控制部件的逻辑图。⑥产生各个微操作的逻辑线路。
4. 总线系统中, 用来管理“总线上设备和设备使用总线过程”的部件, 不一定是独立的控制器, 其组成可以分布到总线的各个部件或各个设备上。功能如下: ①总线系统资源的管理: 连接到总线上的所有部件, 如存储空间、IO 设备、通道、DMA 和中断接口等, 对其进行分配、选择、启动等管理。②总线系统的定时: 产生总线命令和各种定时信号。③总线的仲裁: 确定哪个设备获得总线使用权。④总线的连接: 要完成多种总线不同总线协议的转换和多条总线之间的连接。
5. 通常有硬件判优和软件判优两种方法。①软件判优是通过执行中断查询找到优先中断源, 并立即进入想要的中断服务程序。特点是仅需很少的硬件, 单需执行程序, 所有响应速度慢, 只适用于中断源少、速度慢的计算机系统。②硬件判优是通过硬件判优线路找到优先中断源, 速度快, 但需要额外的硬件。目前主要采用的有两种, 一种是最左判优线路, 是一种并行中断判优线路, 速度快, 费器材; 另一种是链型判优线路, 属于串行中断判优线路, 通常用于 IO 中断判优, 判优线路各环分布在相应的中断接口逻辑中, 即方便又节省器材。

四、应用

1. 首先将运算数据用二进制表示
 $[A]_{补} = ([E_A]_{补} ; [M_A]_{补}) = (0, 001 ; 1.000100)$
 $[B]_{补} = ([E_B]_{补} ; [M_B]_{补}) = (1, 110 ; 0.110110)$
 对阶: 将 B 对阶至 0,001, 即 +011,
 此时 B 为 $[B']_{补} = ([E_{B'}]_{补} ; [M_{B'}]_{补}) = (0, 001 ; 0.000111)$ (0 舍 1 入)
 尾数相减: $[M_A]_{补} + [-M_B]_{补} = 11.000100 + 11.111001 = 10.111101$, 右规一次
 得 $(0, 010 ; 11.011110)$, 舍入后为 $[A - B]_{补} = (0, 010 ; 11.011111)$ 。
2. $[A]_{补} = 111.100110$; $[-A]_{补} = 000.011010$; $[2A]_{补} = 111.001100$; $[-2A]_{补} = 000.110100$ 。因为

$n=6$ 为偶数, 所以 $[B]_{\text{补}}$ 需要 2 位符号位, $[B]_{\text{补}} = 11.100011$, 运算结果如下:

部分积		乘数	B_{n+1}
$[P_0]_{\text{补}}$	000.000000	1 1. 1 0 0 0 1 1 0	
$+ [-A]_{\text{补}}$	000.011010		
	000.011010		
$[P_1]_{\text{补}} \rightarrow 2$	000.000110	1 0 1 1. 1 0 0 0 1	
$+ [A]_{\text{补}}$	111.100110		
	111.101100		
$[P_2]_{\text{补}} \rightarrow 2$	111.111011	0 0 1 0 1 1. 1 0 0	
$+ [-2A]_{\text{补}}$	000.110100		
	000.101111		
$[P_3]_{\text{补}} \rightarrow 2$	000.001011	1 1 0 0 1 0 1 1. 1	
$[P_4]_{\text{补}}$	000.001011		
		$\rightarrow 2$ 1 1 1 1 0 0 1 0 1	

故 $[A \times B]_{\text{补}} = 0.001011110010$

3. 参考答案如下

- (1) 存储容量为 $32\text{bit} \times 2^{22} = 4\text{B} \times 4\text{M} = 16\text{MB}$
- (2) 需要 SRAM 芯片 $\lceil M/m \rceil \lceil N/n \rceil = \lceil 4\text{M}/128\text{K} \rceil \lceil 32/8 \rceil = 32 \times 4 = 128$ 片
- (3) 存储器地址码位数: $\lceil \log_2 4\text{M} \rceil = 22$ 位。
片选译码的地址码位数: $\lceil \log_2 M/m \rceil = \lceil \log_2 32 \rceil = 5$ 位。
- (4) 128 片需要一级门电路 $\lceil 128/8 \rceil = 16$ 个, 驱动中这 16 个一级门电路又需要 $\lceil 16/8 \rceil = 2$ 个二级门电路, 一共需要 $16+2=18$ 个门。

4. 参考答案如下

- (1) 盘面数 $2 \times 8 - 1 = 15$, 去掉一个伺服面
柱面数 $(12.9 - 6.1) \times 10 \div 2 \times 200 + 1 = 6801$
- (2) 非格式化盘组容量 $C_n = \text{内直径} \times \pi \times \text{位密度} \times \text{盘面数} \times \text{柱面数} = 6.1 \times 10 \times 3 \times 6000 \times 15 \times 6801 \div 8 \approx 13.04\text{GB}$
- (3) 传输率 $f = \text{位密度} \times \pi \times \text{内直径} \times \text{转速} = 6000 \times 3 \times 6.1 \times 10 \times \frac{7200}{60} \div 8 \approx 15.7\text{MB/s}$
- (4) 平均时间 $t_{\text{总}} = t_s + t_w + t_{rw} = 12 + \frac{1}{120} \times \frac{1}{2} \times 1000 + \frac{6000}{6.1 \times 10 \times 3 \times 6000 \div 8} \times \frac{1}{120} \times 1000 \approx 19.81\text{ms}$

5. 直接控制编码方式, 每一位编码一种微操作, $L=17$ 。

最短字长编码方式 $L = \lceil \log_2 N \rceil = 5$ 。

分段直接编码中, 由于每一组中需设置一个空白微操作, 实际微命令个数为 $7+1$ 、 $6+1$ 、 $4+1$, 则 $L = \lceil \log_2 8 \rceil + \lceil \log_2 7 \rceil + \lceil \log_2 5 \rceil = 9$ 。

6. (1) 提高记录密度, 增大存储容量的新技术措施: ①用高密度记录磁头和高编码效率的 RLLC 记录方式。②采用接触启停浮动磁头, 降低浮动高度, 使磁头和介质间距仅为 $0.1\mu\text{m}$ 。③采用高性能磁介质并使磁层尽量薄。④采用嵌入式伺服或光伺服技术, 提高道密度, 目前道密度为 $100 \sim 700\text{TPM}$ 。⑤采用 MR 和 GMR 磁头和部分响应最大似然 (PRML) 读信息技术, 可提高记录密度数十倍。

(2) 缩短存取时间、提高传输率采取的新技术措施: ①采用“液态动力轴电机”技术, 提高磁盘转速, 延长使用寿命。②从盘片厚度和盘片直径两个方面减小磁盘尺寸, 即增加转速又缩短磁头行程, 从而减少寻道时间。③在 HDC 中设置“磁盘 Cache”, 即在主存和磁盘之间设置高速缓存, 通常由 SRAM 或 DRAM 组成。速度与主存匹配, 容量在 1MB 到几 MB 之间。

2005

一、选择

1. **[C]**
2. **[A]** 相对间址寻址就是先相对后间址的寻址方式。先相对， $EA'=(PC)+A$ ；后间址， $EA=(EA')=((PC)+A)$
3. **[C]** 寄存器不支持子程序嵌套和递归，子程序起始位置不支持子程序递归。
4. **[C]** 跳跃指令执行后 PC 自增，本是要执行下一条，但因为跳跃执行了另一个，目标地址是下一条指令的地址。
5. **[C]** 指令系统是计算机程序员接触到的计算机的所有功能。
6. **[A]** 原码算术右移时符号位不参与，右移后小数点后补 0。补码算术右移时，符号位不参与移位，右移后小数点后补 1。
7. **[B]** 机器指令由微程序构成，微程序由微指令构成，微指令是微命令的序列。
8. **[B]** 字长 32 位，按字编址也即按 4B 编址。 $256MB/4B=64M$ ，寻址范围为 $0\sim 64M-1$ 。
9. **[B]** 规格化浮点数尾数部分的小数点后第一位和尾数符号必定异号。
10. **[C]** 双极型存储芯片 BiRAM，地址和数据引脚都不复用，所以和为 $16+8=24$ 。（数据引脚不复用要乘 2）
11. **[B]** 总线从设备就是被主设备访问的设备。
12. **[C]** 基础概念。（未来再考可能几乎为 0）
13. **[C]** 平均等待时间是二倍转速的倒数，所以转速提高一倍，平均等待时间缩小一半。
14. **[C]** CPU 对通道的启动和停止只能通过通道 I/O 指令实现。
15. **[A]** 中断最重要的就是正确进入中断服务程序和正确返回断点处。

二、填空

1. **(程序计数器 PC) (指令寄存器 IR) (指令译码器 ID)**
2. **(101 1100 0111) (001 1100 0111)**
3. **(指令长度) (程序总位数) (指令字所能表示的操作)**
4. **(子程序调用指令) (子程序返回指令)**
5. **(高速化运算电路) (改进体系结构) (高速化运算方法和传输结构)**
6. **(同步控制) (异步控制)**
7. **(只读) (追记) (可擦写)**
8. **(剩磁小) (易磁化) (导磁率高)**
9. **(专用型) (复用型)**
10. **(E2PROM) (EPROM) (Flash)**
11. **(中断请求) (中断判优) (中断响应) (中断处理)** 绿书 P333

三、 简答

1. 微命令试构成控制信号序列的最小单位，由控制部件向执行部件发出。微操作是执行部件接受微命令后所进行的最基本操作。微命令是微操作的控制信号，微操作是微命令的操作过程。微指令是一组实现一定操作功能的微命令组合，通常由二进制编码组成。微程序是由有序的微指令组成的，存放在控制存储器中。一条机器指令对应一段微程序。
2. 指令格式与功能不仅影响机器的硬件结构，也直接影响系统软件和机器的使用范围。从指令功能而言，指令即是程序员进行程序设计的最小单位，又是机器硬件能够理解并执行的语言，是设计硬件的基础。从指令格式而言，指令长度，操作码和地址码结构都直接影响计算机的速度、功能强弱、存储空间利用率及硬件设计复杂性和软件设计的方便性。
3. 将非规格化浮点数，经过尾数移位和阶码调整转换为规格化浮点数的过程称为浮点规格化。规格化使得每次计算后都能获得最大有效数字；使得浮点数的表示具有唯一性；为浮点数计算和程序设计提供方便。
4. 功能：快速查找并跟踪磁道；启动或者中途出错时能够迅速磁头归零；停机掉电的时候能快速退出盘面以保护磁头和盘面。性能要求：要求定位速度快，精度高。速度快可以保证系统运行速度加快，精度高可以保证读出正确的数据且提高道密度提高磁盘容量。
5. ①集中式仲裁：设置集中式的仲裁电路，连接所有总线设备并按照某种策略选择其中一个总线主设备获得总线使用权。优点：仲裁过程以及总线设备接口简单。缺点：仲裁电路出现故障将导致整个系统瘫痪；扩展设备需要对仲裁电路进行较大的修改，难度较大。②分布式仲裁：所有主设备均设置自己的仲裁电路。当主设备发出请求时，各仲裁电路按照一定策略共同决定总线使用权。优点：线路可靠性高，扩展灵活，设备接插比较随意。缺点：确定总线主设备是否在正常工作，系统需要进行超时判断；由于每个主设备需要在其接口设计仲裁电路，导致设计的复杂性加大。
6. （一）传送初始化。CPU 判断磁盘空闲繁忙或状态良好故障，若忙或故障就执行其他程序，否则向磁盘 DMA 接口寄存器传送如下参数并启动磁盘工作继续执行其他程序。内存始地址送至 ABR；传送数据个数的补码送至 WC；读盘命令送至 DCR；磁盘柱面号盘面号扇区号送至设备内部地址寄存器 DAR。
（二）数据传送。①磁盘按位读出信息并组装成一个单位数据后传送到 DBR，表示准备就绪。②磁盘向 DMA 接口发生传输请求 DREQ，DMA 接口向 CPU 申请总线控制权。③CPU 发回 HLDA 信号，表示取得总线控制权，磁盘可以挪用一個存储周期传送一个单位数据。CPU 建立一个 DMA 周期，DMA 周期完成接下来的工作。④ABR 中的内容传送至主存 MAR，将 DBR 中的单位数据送至主存 MBR，同时回送写存储器命令到主存。⑤启动主存写操作，将 MBR 中的数据写入 MAR 指定存储单元。⑥修改传输长度即 WC+1，给出下一个要传输单位数据的主存地址即 ABR+1，同时判断 WC 是否溢出。⑦若无溢出检查也无错误，重复执行①~⑥；否则向 CPU 发中断请求。
（三）DMA 结束处理。CPU 响应 DMA 的中断请求，转去执行中断服务程序，从 DMA 接口的 DSR 中取出状态，进行判断，若为传输错误引起的中断，则转错误诊断及处理程序；若为正常结束，则停磁盘。

四、 应用

1. 将被乘数和乘数转为规格化浮点数二进制
 $[A]_{补} = ([E_A]_{补} ; [M_A]_{补}) = (0110 ; 0.1101)$
 $[B]_{补} = ([E_B]_{补} ; [M_B]_{补}) = (1011 ; 1.0110)$

求积的阶码，运算过程中阶码取两位符号位

$[E_C]_{\text{补}} = [E_A]_{\text{补}} + [E_B]_{\text{补}} = 00110 + 11011 = 00001$ ，结果无溢出， $[E_C]_{\text{补}} = 0001$

	部分积	乘数	M_{Bn+1}
$[P_0]_{\text{补}}$	00.0000	1. 0 1 1 0 0	0
$[P_1]_{\text{补}} \rightarrow 1$	00.0000	0 1. 0 1 1 0	0
$+ [-M_A]_{\text{补}}$	11.0011		
	11.0011		
$[P_2]_{\text{补}} \rightarrow 1$	11.1001	1 0 1. 0 1 1	1
$[P_3]_{\text{补}} \rightarrow 1$	11.1100	1 1 0 1. 0 1	1
$+ [M_A]_{\text{补}}$	00.1101		
	00.1001		
$[P_4]_{\text{补}} \rightarrow 1$	00.0100	1 1 1 0 1. 0	0
$+ [-M_A]_{\text{补}}$	11.0011		
$[P_5]_{\text{补}}$	11.0111		
		$\rightarrow 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1.$	

故 $[M_C]_{\text{补}} = [M_A]_{\text{补}} + [M_B]_{\text{补}} = 1.01111110$

因 $[M_C]_{\text{补}}$ 已是规格化数，所以 $[C]_{\text{补}} = 0001, 1.01111110$

2. $[A]_{\text{补}} = 00.10101$, $[B]_{\text{补}} = 11.00011$, $B_0 = 1$, $[-B]_{\text{补}} = 00.11101$ ，运算过程如下：

余数和被除数	商
$[R_0]_{\text{补}} \quad 00.10101$	$A_{01} \neq B_0 \quad 0. 0 0 0 0 0$
$\leftarrow 1 \quad 01.01010$	$0. 0 0 0 0 0 \quad 0$
$+ [B]_{\text{补}} \quad 11.00011$	
$[R_1]_{\text{补}} \quad 00.01101$	$A_{01} \neq B_0$
$\leftarrow 1 \quad 00.11010$	$0. 0 0 0 0 \quad 0 \quad 0$
$+ [B]_{\text{补}} \quad 11.00011$	
$[R_2]_{\text{补}} \quad 11.11101$	$A_{01} = B_0$
$\leftarrow 1 \quad 11.11010$	$0. 0 0 0 \quad 0 \quad 0 \quad 1$
$+ [-B]_{\text{补}} \quad 00.11101$	
$[R_3]_{\text{补}} \quad 00.10111$	$A_{01} \neq B_0$
$\leftarrow 1 \quad 01.01110$	$0. 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0$
$+ [B]_{\text{补}} \quad 11.00011$	
$[R_4]_{\text{补}} \quad 00.10001$	$A_{01} \neq B_0$
$\leftarrow 1 \quad 01.00010$	$0. \quad 0 \quad 0 \quad 1 \quad 0 \quad 0$
$+ [B]_{\text{补}} \quad 11.00011$	
$[R_5]_{\text{补}} \quad 00.00101$	$\leftarrow 1 \quad 0. 0 1 0 0 0$
	商符变反 $1. 0 1 0 0 0$

由于未除尽，商为负（ $C_0 = 1$ ）且 $A_{01} \neq B_0$ ，故所得余数即为真余数，不必修正。

即 $[2^{-5}R_S] = 0.00101 \times 2^{-5} = 0.000000101$

3. 依题目可知加法器的级数为：

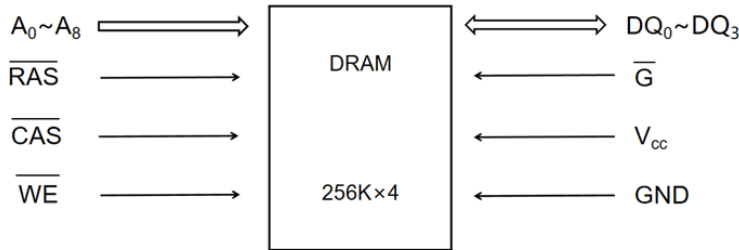
$N=6+2+4+6=18$ 级。信号时间裕量 $\Delta = 1/3 (n \times t_d) = 6 \times 0.1 = 0.6\text{ns}$

时间周期称 $T_{cp} \geq n \times t_d + \Delta = 18 \times 0.1 + 0.6 = 2.4ns$

所以时钟频率可取： $1/T_{cp} = 1/(2.4 \times 10^{-9}) \approx 416MHz$

4. 参考答案如下

(1) (注意 DRAM 芯片, 复用要少画一半引脚)



(2) 所需芯片数 $[M/m][N/n] = [2M/256K][64/4] = 8 \times 16 = 128$ 片。

(3) 存储器地址码位数： $\log_2(2M) = 21$ 位。

片选译码所需地址码位数： $\log_2[M/m] = \log_2[2M/256K] = \log_2 8 = 3$ 位。

(4) 为了驱动 128 片芯片负载, 需要 $[128/8] = 16$ 个一级门, 为了驱动这 16 个一级门, 又需要 $[16/8] = 2$ 个二级门, 一共需要 $16+2=18$ 个门。

5. 参考答案如下

(1) 盘面数为 $6 \times 2 - 1 = 11$

柱面数 $= (8.9 - 4.1) \times 10 \div 2 \times 150 + 1 = 3601$

(2) 非格式化容量 $=$ 内直径 $\times \pi \times$ 位密度 \times 盘面数 \times 柱面数 $= 4.1 \times 10 \times 3 \times 5000 \times 11 \times 3601 \div 8 \approx 2.84GB$

(3) 数据传输率为 $f =$ 位密度 $\times \pi \times$ 内直径 \times 转速 $= 5000 \times 3 \times 4.1 \times 10 \times \frac{7200}{60} \approx 8.78MB/s$

(4) 15 台磁盘机需要 4 位台号

3601 个柱面需要 12 位柱面号

11 个盘面需要 4 位盘面号

64 个扇区需要 6 位扇区号

地址方案为 [台号 4 位, 柱面号 12 位, 盘面号 4 位, 扇区号 6 位]

25.....22	21.....10	9.....6	5.....0
台号	柱面号	盘面号	扇区号

6. 参考答案如下

(1) 主存系统容量为 $2MB \times 32 = 64MB$

(2) 体地址 $d = A \bmod n = 3FFFFC3H \bmod 20H = 3$

体内地址 $I = [A/n] = [3FFFFC3H/20H] = 1FFFFEH$

其实也可以这样考虑, $3FFFFC3H=0011\ 1111\ 1111\ 1111\ 1010\ 0011$

32 个体, 需要 5 位来选体, 因为低位交叉, 那就是低 5 位, 000111, 所以体号是 3。

剩余部分是 $001\ 1111\ 1111\ 1111\ 1101=1FFFFEH$

(3) 按如上策略进行拼接。

$2FFFFH=0010\ 1111\ 1111\ 1111$, 体地址 16 是 10000

可得 $0000\ 0101\ 1111\ 1111\ 1111\ 0000=5FFFFH$

2004

一、选择

1. **[C]** 主机是由 CPU、内存储器及相应接口组成。
2. **[A]** 变址间址寻址方式也就是先变址后间址的寻址方式。先变址址,有效地址 $EA'=(X)+A$; 后间址,有效地址 $EA=(EA')=(A+(X))$ 。
3. **[D]** 溢出分为上溢和下溢。对于下溢出的判断,如果在求阶码和就判断,可能出现本不应为溢出而被误以为是下溢,错误的扩大了溢出范围;而如果在规格化过程中判断,就可能把本来是阶下溢错判为阶上溢了。对于上溢出的判断,如果在求阶码和后就判断,就可能出现本不应为溢出而被误以为是上溢,错误的扩大了溢出范围。因此,判断溢出应该在求阶码和且规格化之后进行。
4. **[C]** 由于加法器是整个运算器的核心,提高加法器的运算速度最直接的方法就是多位并行加法。本题中四个选项均对加法器的速度产生影响,进位传递延迟对并行加法器的影响最为关键。
5. **[B]** “中断屏蔽码”的作用是为了实现多重中断。其目的是封锁中断优先权不高于本级的中断源发出的中断请求,而高于本级的中断源可以发出中断请求。换句话说,“中断屏蔽码”的作用是暂时剥夺部分中断源发出中断请求的权利,使主机不能响应这些请求,所以选择 B。
6. **[A]** 最短字长编码这种编码法是将所有的微命令进行统一的二进制编码,用不同的码点组合表示不同的微指令,通过译码器产生微操作控制信号。由于在控制器中需要设置一个空微操作(什么也不做)所以参加编码的微命令个数是 $N+1$ 个。所以选择 A。
7. **[A]** 浮点数尾数的符号位代表数的符号,最小的负数是距离零点最远的负数,即绝对值最大的负数。B、D 选项尾数的符号位为 0,表示它们是正数。C 选项阶码的符号位为 1,表明其阶码为负值(距离零点近,比较大的复数)。所以选择 A。
8. **[D]** 非易失性存储器即掉电后信息也不丢失的存储器,ROM 和磁表面存储器 MSM 都是非易失性存储器,所以选择 D。
9. **[B]** 所谓字编址就是一个存储单元存放一个字长的信息,即一个地址存放一个字长的信息。存储容量除以字长得存储器地址个数,从而确定寻址范围。该题目中存储容量为 128MB,存储字长为 32 位即 4B, $128MB/4B=32M$ /即有 32M 个存储单元,寻址范围为 $0\sim 32M-1$,所以选择 B。
10. **[C]** SRAM 芯片数据线复用,即每位的输入输出使用同一数据线,字位结构为 $1M\times 4$ 位的 SRA 芯片,表示该芯片具有 1M 个存储单元,每个存储单元存放 4 位信息。则地址引脚需要 20 根 ($1M=2^{20}$),数据引脚要 4 根(每一位需要 1 根)共 24 根,所以选择 C。
11. **[A]** 统一编址下,内存专口开辟一块区域用于 I/O 寻址,使用统一的访存指令访问 I/O 端口,地址码可区分访问的是主存还是外设。
12. **[C]** 中断方式主要由程序实现。
13. **[C]** 对于高速外设进行数据传输时,通常成组传送,而低速外设通常为字节或字传送。选择通道和数组多路通道用于高速成组传送的外设。所以选择 C。
14. **[D]** 输入输出设备和外存储器等一起称之为外围设备。A、B、C 选项 都是外围设备的一类,但不是全部。D 选项正确。
15. **[A]** 从工作原理可将磁头分为感应式和磁阻式两类,感应式磁头依电磁感应定律工作,其组成

必须有读写线圈；磁阻式依磁阻元件通过磁场时，磁阻发生改变而工作，其中无线圈且仅能读信息。

二、 填空

1. (1010 0110) (0010 0110)
2. (1010 0111) (0000 1000)
3. (微操作控制字段) (顺序控制字段) 绿书 P204
4. (定时) (仲裁) (连接)
5. (微操作控制部件)
6. (0 舍 1 入) (恒置 1)
7. (尾数相加) (规格化尾数) (舍入)
8. (速度快) (价格低) (多级存储层次)
9. (电荷泄露) (刷新)
10. (形变型) (相变型) (磁光型)
11. (发光二极管) (阴极射线管显示器) (液晶显示器)
12. (地址缓冲寄存器) (字计数器) (数据缓冲寄存器) (设备控制与状态寄存器)

三、 简答

1. 按操作数存储位置可将指令分为堆栈型、累加型、通用寄存器型，对应的机器就是堆栈型指令机、累加型指令机、通用寄存器型指令机。累加型指令机中，所有的运算指针均对累加器中的操作数进行，累加器地址可以隐含，因此指令长度较短，但是累加器容易形成瓶颈。通用寄存器型指令机中，所有运算指针均对寄存器中的操作数进行，由于可以根据表达式分配寄存器，编译时容易实现表达式的运算，且运算迅速，同时操作所地址必须显式给出，因此指令长度较长。
2. ①同步总线：总线系统有一个共所有设备使用的统一时钟，设备之间都按照约定俗成的时序进行操作和完成信息交换。同步总线的所有信号命令都是与总线时钟同步的，所以总线的使用设计调试都较为简单。所有设备的总线操作所需要的总线时钟都是整数，且总线操作周期需设计得满足最慢设备使用，从而使高速设备不能高效使用总线，造成时间浪费。
②异步总线：总线系统没有统一的时钟，任何一件事发生都只能是前一个或一些时间的结束，所有设备都以“握手”的方式进行联系从而完成信息交换。总线对设备速度差异的适应性更强，慢速设备对快速总线的影响仅限于本次总线操作，不影响下一次快速总线设备操作，效率高。异步总线无统一的时钟，设计控制调试都比较复杂。
3. 补码定点整数的数据表示范围为： $-2^{15} \sim 2^{15} - 1$ ；补码浮点数的表示范围为： $-1 \times 2^{31} \sim (1 - 2^{-9}) \times 2^{31}$ 。补码定点整数表示的有效数字位为 15 位。补码浮点数表示的有效数字位为 9 位。不难看出，在电子器件用量相同或相近的情况下，浮点数据比定点数据表示的范围大，但数据精度不及定点的高。另外，采用浮点数据表示时，阶码和尾数的划分，要根据数据范围和精度的要求综合考虑。
4. 相同点：都是用来储存信息的，且以二进制的形式读取或写入。
不同点：主存容量小速度快成本高，CPU 可对其直接访问，通常为半导体存储器。辅存容量大速度慢成本低，需要通道或 DMA 等 IO 方式控制读写，通常为磁表面存储器、光盘存储器。
5. 中断：CPU 暂时中止现行程序，转去执行“处理随机发生的紧急事件或特殊请求”的程序，处理

完后自动返回被中止程序继续运行。

响应中断的条件有 4 个：①有中断请求。②CPU 的一条指令执行完毕。③CPU 允许响应中断。④无更紧迫的任务或事件。

6. （知识点过于老旧，已无再考可能，建议舍弃）

四、应用

1. 首先将被除数和除数表示出来： $[X]_{\text{移、补}} = E_x$ ， $M_x = 1100, 1.00110$ 。

由于 $|M_x| > |M_y|$ ，故需要对 X 进行尾数调整，即 $[X]_{\text{移、补}} = 1101, 1.1001$ 。

$[Y]_{\text{移、补}} = E_y$ ， $M_y = 0110, 0.10111$ ， $[E_y]_{\text{补}} = 1110$ ， $[-E_y]_{\text{补}} = 0010$ ， $[-M_y]_{\text{补}} = 1.01001$ ，商的阶码 $[E_Q]_{\text{移}} = [E_x - E_y]_{\text{移}} = [E_x]_{\text{移}} + [-E_y]_{\text{移}} = 1101 + 0010 = 1111$

采用加减交替法， $[M_x]_{\text{补}} = 11.10011$ ， $[M_y]_{\text{补}} = 00.10111$ ， $M_{y0} = 0$ ， $[-M_y]_{\text{补}} = 11.01001$

运算过程如下：

余数和被除数							
$+ [M_y]_{\text{补}}$	11.10011 00.10111	$M_{x01} \neq M_{y0}$	0	0	0	0	0
$[R_0]_{\text{补}}$ $\leftarrow 1$	00.01010 00.10100	$M_{x01} = M_{y0}$	0	0	0	0	1
$+ [-M_y]_{\text{补}}$	11.01001						
$[R_1]_{\text{补}}$ $\leftarrow 1$	11.11101 11.11010	$M_{x01} \neq M_{y0}$	0	0	0	0	1
$+ [M_y]_{\text{补}}$	00.10111						
$[R_2]_{\text{补}}$ $\leftarrow 1$	00.10001 01.00010	$M_{x01} = M_{y0}$	0	0	0	1	0
$+ [-M_y]_{\text{补}}$	11.01001						
$[R_3]_{\text{补}}$ $\leftarrow 1$	00.01011 00.10110	$M_{x01} = M_{y0}$	0	0	1	0	1
$+ [-M_y]_{\text{补}}$	11.01001						
$[R_4]_{\text{补}}$ $\leftarrow 1$	11.11111 11.11110	$M_{x01} \neq M_{y0}$	0	1	0	1	1
$+ [M_y]_{\text{补}}$	00.10111						
$[R_5]_{\text{补}}$	00.10101	$M_{x01} = M_{y0}$ $\leftarrow 1$	1	0	1	1	0

2. $[X]_{\text{补}} = 00.11011$ ， $[Y]_{\text{补}} = 11.0101$ ， $[-X]_{\text{补}} = 11.00101$ ，运算过程如下：

部分积		乘数						Y_{n+1}
$[P_0]_{\text{补}}$	$+ [-X]_{\text{补}}$	00.00000 11.00101	1	1	0	1	0	0
$[P_1]_{\text{补}}$	$\rightarrow 1$ $+ [X]_{\text{补}}$	11.00101 11.10010 00.11011	1	1	1	0	1	1
$[P_2]_{\text{补}}$	$\rightarrow 1$ $+ [-X]_{\text{补}}$	00.01101 00.00110 11.00101	1	1	1	1	0	0
$[P_3]_{\text{补}}$	$\rightarrow 1$ $+ [X]_{\text{补}}$	11.01011 11.10101 00.11011	1	1	1	1	0	1
$[P_4]_{\text{补}}$	$\rightarrow 1$ $+ [-X]_{\text{补}}$	00.10000 00.01000 11.00101	0	1	1	1	1	0
$[P_5]_{\text{补}}$	$\rightarrow 1$	11.01101 11.10110 11.10110	1	0	1	1	1	1
$[P_6]_{\text{补}}$			$\rightarrow 1$	0	1	0	1	1

所以 $[X \times Y]_{\text{补}} = 11.1011\ 0010\ 111$, $[X \times Y] = -0100\ 1101\ 001$ 。

3. 顺序-转移型的后继微地址的形成分顺序执行和转移执行两种情况, 顺序执行时后继指令地址由 μPC 加一个增量(通常为 1)产生, 转移时后继微指令由微指令的转移字段 BAF 指示的转移地址来源给出。转移地址的来源有以下三种: 由 BAF 确定的地址, 机器指令所对应的微程序的入口地址, 微子程序入口地址和返回地址。

断定型的后继微地址可由微程序设计者直接指定, 或者由微程序设计者指定的测试判别字段产生。微指令中非测试字段 HF 直接生成微地址码中的非测试地址, 测试控制字段 TCF 指出产生测试地址的测试条件, 根据测试结果形成微地址码中的测试地址 TF。

	顺序转移方式	断定方式
优点	微指令中 SCF 字段较短, 后继微地址产生机构比较简单	能以较短的 SCF 配合实现多路并行转移, 提高了微程序的执行效率和速度; 微程序在 M 中分配物理空间方便灵活
缺点	不利于解决两路以上的并行微程序转移, 从而不利于提高微程序的执行速度; 必须为每个分支安排转移流程, 微程序在 CM 中的物理分配不灵活	后继微地址码的生成机构比较复杂
适用范围	速度要求不高的小型、微型计算机	设计快速微程序控制器时使用

4. 参考答案如下(注, 题目图可能有错误, 引脚应是 $A_0 \sim A_9$)

- (1) SRAM 芯片, 其地址引脚不复用, 数据引脚复用。地址引脚是 $A_0 \sim A_9$, 有 2^{10} 个存储单元, 每个存储单元是 4bit, 容量为 $2^{10} \times 4\text{bit} = 512\text{KB}$ 。
- (2) ①所需芯片数: $\lceil M/m \rceil \lceil N/n \rceil = \lceil 16\text{K}/1\text{K} \rceil \lceil 32/4 \rceil = 16 \times 8 = 128$ 片。
②存储器地址码位数: $\log_2(16\text{K}) = 14$ 位。
片选译码所需地址码位数: $\log_2 \lceil M/m \rceil = \log_2 \lceil 16\text{K}/1\text{K} \rceil = \log_2 16 = 4$ 位。
存储器的低 10 位地址接芯片, 高 4 位地址 $A_{13}A_{12}A_{11}A_{10}$ 译码产生片选。
③为了驱动 128 片芯片负载, 需要 $\lceil 128/8 \rceil = 16$ 个一级门, 为了驱动这 16 个一级门, 又需要 $\lceil 16/8 \rceil = 2$ 个二级门, 一共需要 $16+2=18$ 个门。

5. 参考答案如下

- (1) 盘面数为 $4 \times 2 - 1 = 7$ 面, 去掉一个伺服面。
柱面数为 $(8.9 - 3.9) \times 10 \times 40 \div 2 + 1 = 1001$ 。
- (2) 没有谈及扇区信息, 则默认为非格式化容量, $C_n = \text{位密度} \times \pi \times \text{内直径} \times \text{盘面数} \times \text{柱面数} = 500 \times 3 \times 3.9 \times 10 \times 7 \times 1001 = 409909500\text{bit} \approx 48.86\text{MB}$
- (3) 数据传输率为 $f = \text{位密度} \times \pi \times \text{内直径} \times \text{转速} = 500 \times 3 \times 3.9 \times 10 \times \frac{7200}{60} = 7020000\text{bit/s} \approx 856.9\text{KB/s}$
- (4) 12 台磁盘机需要 4 位台号
1001 个柱面需要 10 位柱面号
7 个盘面需要 3 位盘面号
64 个扇区需要 6 位扇区号
地址方案为 [台号 4 位, 柱面号 10 位, 盘面号 3 位, 扇区号 6 位]

22……19	18……9	8……6	5……0
台号	柱面号	盘面号	扇区号

6. 参考答案如下

- (1) 显示器缓存容量位每一帧可显示的汉字所占容量： $25 \times 40 \times 24 \times 24 \div 8 = 72000\text{B}$ 。
- (2) 字符发生器 ROM 容量为 6000 个汉字乘以每个汉字容量：
 $6000 \times 24 \times 24 \div 8 = 432000\text{B}$
- (3) 设置点计数器、行计数器、字计数器、排计数器控制显存访问与屏幕扫描之间的同步。
 点计数器范围： $24+3=27$ 。
 行计数器范围： $24+8=32$ 。
 字、排计数器的范围不仅与扫描正程时间有关，而且与扫描逆程时间有关，因此计算较为复杂。
- 列方程 $\begin{cases} (40+x) \times 0.8 = 40 \\ (25+y) \times 0.8 = 25 \end{cases}$
- 解得 $x=10$, $y=65.25$, 则：
 字计数器范围 $= 40 + 10 = 50$ 。
 排计数器范围 $= 25 + 6.25 = 31.25$ 取 32。
 点频为 $50\text{HZ} \times 32 \text{排} \times 50 \text{字} \times 32 \text{行} \times 27 \text{点} = 6912000\text{HZ} = 69.12\text{MHZ}$ 。
 行频为 $50\text{HZ} \times 32 \text{排} \times 50 \text{字} \times 2 \text{行} = 256000\text{Hz}$ 。
 字频为 $50\text{HZ} \times 32 \text{排} \times 50 \text{字} = 8000\text{Hz}$ 。
 排频为 $50\text{HZ} \times 32 \text{排} = 1600\text{HZ}$ 。

2003

一、选择

1. **[B]** 间址变址寻址方式也就是先间址后变址的寻址方式。先间址,有效地址 $EA'=(A)$; 后变址,有效地址 $EA=EA'+(X)=(A)+(X)$ 。
2. **[B]** 使用扩展操作码的主要原则就是将使用频度高的指令分配短的操作码,而将使用频度低的指令分配较长的操作码,这样可以缩短程序的总位数,节省存储空间、译码时间,提高程序的执行速度。
3. **[B]** 程序控制指令主要包含四类。第一类:无条件转移、调用和返回指令;第二类:调用和返回指令;第三类:循环控制指令;第四类:软中断指令及中断返回指令。
4. **[C]** 表示结果为正数,无溢出;01 表示结果正溢出;10 表示结果负溢出;11 表示结果为负数,无溢出。
5. **[A]**
6. **[C]** 判断规格化浮点数的大小首先看符号,正数比负数大,然后比较阶码的大小,最后才是尾数大小的比较。此题中,A、B 选项对应的浮点数均为负数。由于采用移码表示阶码,C 选项的阶码为正数,而 D 选项的阶码为负数,故 C 表示的浮点数最大。
7. **[D]**
8. **[D]** 微程序控制器中,每条微指令都对应一段微程序。识别指令是通过对指令的操作码译码来完成的,故指令对应微程序的第二条微指令地址应该由指令操作码经过一定的变换形成。
9. **[A]** 从存储位元结构和使用的器件角度,双极型 RAM (BiRAM),MOS 动态 RAM (DRAM) 和 MOS 静态 RAM (SRAM) 三种类型。按速度排序,BiRAM 速度最快、SRAM 次之、DRAM 最慢;按集成度和功耗比较,BiRAM 集成度低、位功耗大,DRAM 集成度最高、位功耗最小,SRAM 则位居前两者之间。
10. **[B]** 字位结构为多位的 DRAM 存储芯片,地址引脚、数据引脚都复用。数据引脚复用的含义是,一位数据使用一个引脚既作输入用又作输出用,注意,要设置控制引脚 OE,用来确定何时作输入何时作输出使用。地址引脚复用的含义是,两位地址码串行、分时使用一个地址引脚,此时,要设置行、列地址选通两个控制引脚。据此,地址引脚 10 与数据引脚 4 之和应为 14,即选择 B。
11. **[D]**
12. **[B]** 随机访问存储器的特点是以存储单元为单位,可随时访问存储器的任何存储单元且访问所需时间都是相同的,与存储单元所处的物理位置无关。很显然,本题应选择主存储器。
13. **[D]** 在程序直接、程序中断、DMA、通道和外围处理机的五种 I/O 控制传送方式中,从前至后硬件逻辑一个比一个复杂,速度一个比一个快,对 CPU 的打扰一个比一个少。
14. **[A]** 按信息的可保存性,存储器可分为挥发性和非挥发性,即易失性和非易失性两类。挥发性存储器即断电后信息就丢失的存储器,BiRAM、DRAM、SRAM、CAM 都是挥发性存储器。
15. **[D]** 对于只读光盘、CD-R 光盘,其光道是从内向外伸展的-条螺旋线。它们使用的驱动电机是恒线速(CLV)盘,即角速度时刻变化而线速度保持不变,写入时,信息位的间隔也是等距的。而对于磁光盘光道为一组同心圆,它们使用的驱动电机是恒角速(CAV)的,即对于不同的光道,

线速度不同而角速度一样,导致内圈光道线速度小,外圈光道线速度大.写入时,信息位的间隔是不等距的.所以应选位记录密度相同的螺旋线。

二、 填空

1. (0100 1100) (1100 1100)
2. (0000 0000) (0000 0000)
3. (完备性) (有效性) (规整性)
4. (数据总线) (地址总线) (控制总线)
5. (1110 1010) (1010 1011)
6. (最短字长) (直接控制) (互斥微命令) (相容微命令)
7. (集中式刷新) (分散式刷新)
8. ($\lceil \log_2 m \rceil + \lceil \log_2 n \rceil$) ($\lceil A/n \rceil$) ($A \bmod n$)
9. (柱面号) (盘面号) (扇区号)
10. (忙/闲) (就绪/完成) 绿书 P343、P345
11. (击打式) (非击打式) (点阵打印机) (激光打印机) (此知识点过于老旧,基本不可能再考)

三、 简答

1. 一个特定的存储区或寄存器,一端是固定的,另一端是浮动的。所有数据的存入或取出,只能在浮动的一端(栈顶)进行,严格按照先进后出的原则存取。随机存储器中开辟一个区域作为堆栈叫软件堆栈,用寄存器构成的堆栈叫硬件堆栈。通常用来保护断点和现场。
2. ①总线主设备发出总线使用请求,并等待获得总线控制权。②经过总线控制器仲裁,对请求给出应答,表明该主动设备获得总线控制权。③主设备通过地址总线发出源设备的地址选择源设备(这句话我没读明白),通过控制总线向源设备发出读数据的总线命令。④从设备读出数据送到数据总线,同时发出撤销使用总线的请求,这设备从数据总线取得数据,表示设备本次使用总线完毕。⑤总线控制器收到撤销使用总线请求信号,收回总线控制权,发出总线撤销使用应答信号并释放总线。
3. 寻址技术又称寻址方式,是确定操作数地址的技术,是计算机中硬件对软件最早提供支持的技术之一。作用是扩大存储器访问范围、为高级语言和数据结构的实现提供支持、支持系统对程序的定位。
4. 功能上 Cache-主存解决的是速度问题,主存-辅存解决的是容量问题。实现技术上,Cache-主存为了解决速度问题,完全使用硬件,且对所有程序员透明;主存-辅存则是硬件和软件一起应用且以软件为主,仅对应用程序员透明。Cache 多以双极型存储器件构成,主存多由 MOS 器件构成,辅存多以磁盘光盘磁带为主。
5. 相同点:都是计算机中的控制信息,计算机通过他们实现数值处理或非数值处理。不同点:指令由 CPU 执行并控制,通道命令字由通道控制并执行;指令所有计算机都由,通道命令字只有计算机配置通道才有;指令可以完成复杂的任务,复杂规整完备有效。通道命令字只规定通道及其连接外设的操作等,简单且具有局限性。
6. 主要由以下四部分组成:①设备选择器:接收设备地址或编号进行译码,选中并控制相应外设及其接口实现 IO 操作。②数据缓冲器:寄存要输入输出的信息,如键盘、鼠标、扫描仪等。实际上是一个小容量存储器。③状态与控制命令寄存器 SCR:寄存两部分内容,一部分是 CPU 送来的控制命令,以指明接口及外设的操作方式;另一部分记录表示外设及接口的运行状态。

CPU 可根据接口实际运行状态对 IO 操作进行动态控制，或根据当前状态作出相应处理。④中断控制逻辑：实现程序中断方式的数据传送和完成外设的操作控制所必需的逻辑线路。形成送往 CPU 的中断信号、状态信号、中断类型码，接收中断响应信号，从而进入中断服务。

四、应用

1. 参考答案如下

(1) 设某一布的部分积为 $[P_i]_{\text{补}}$ ，据递推公式可得下一个部分积 $[P_{i+1}]_{\text{补}}$ ：

$$\text{则 } [P_{i+1}]_{\text{补}} = 2^{-1} \{ [P_i]_{\text{补}} + (Y_{n-i+1} - Y_{n-i})[X]_{\text{补}} \}$$

有了 $[P_{i+1}]_{\text{补}}$ ，又可以根据递推公式得到下一个部分积 $[P_{i+2}]_{\text{补}}$

$$\text{则 } [P_{i+2}]_{\text{补}} = 2^{-1} \{ [P_{i+1}]_{\text{补}} + (Y_{n-i} - Y_{n-i-1})[X]_{\text{补}} \}$$

将两个乘法合并成一步，即可从 $[P_i]_{\text{补}}$ 一步求得 $[P_{i+2}]_{\text{补}}$ ：

$$\begin{aligned} [P_{i+2}]_{\text{补}} &= 2^{-1} \{ [P_{i+1}]_{\text{补}} + (Y_{n-i} - Y_{n-i-1})[X]_{\text{补}} \} \\ &= 2^{-1} \left\{ 2^{-1} \{ [P_i]_{\text{补}} + (Y_{n-i+1} - Y_{n-i})[X]_{\text{补}} \} + (Y_{n-i} - Y_{n-i-1})[X]_{\text{补}} \right\} \\ &= 2^{-2} \{ [P_i]_{\text{补}} + (Y_{n-i+1} - Y_{n-i})[X]_{\text{补}} + 2(Y_{n-i} - Y_{n-i-1})[X]_{\text{补}} \} \\ &= 2^{-2} \{ [P_i]_{\text{补}} + (Y_{n-i+1} - Y_{n-i} + 2Y_{n-i} - 2Y_{n-i-1})[X]_{\text{补}} \} \\ &= 2^{-2} \{ [P_i]_{\text{补}} + (Y_{n-i+1} + Y_{n-i} - 2Y_{n-i-1})[X]_{\text{补}} \} \end{aligned}$$

(2) 当 $i=0$ 时，部分积为 $[P_0]_{\text{补}}$ ，其两位一乘的第一步部分积表达式为：

$$[P_2]_{\text{补}} = 2^{-2} \{ [P_0]_{\text{补}} + (Y_{n+1} + Y_n - 2Y_{n-1})[X]_{\text{补}} \}$$

(3) $[X]_{\text{补}} = 00.1111$ ， $[-X]_{\text{补}} = 11.00001$ ， $[2X]_{\text{补}} = 001.111110$ ；因为 $n=6$ 为偶数，所以 $[Y]_{\text{补}}$ 需要两位符号位， $[Y]_{\text{补}} = 11.000111$ 。运算过程如下：

部分积		乘数								Y_{n+1}
$[P_0]_{\text{补}}$	000.000000	1	1	0	0	0	1	1	1	0
$+[-X]_{\text{补}}$	111.000001									
$[P_1]_{\text{补}}$	111.000001									
$\rightarrow 2$	111.110000	0	1	1	1	0	0	0	1	1
$+ [2X]_{\text{补}}$	001.111110									
$[P_2]_{\text{补}}$	001.101110									
$\rightarrow 2$	000.011011	1	0	0	1	1	1	0	0	0
$[P_3]_{\text{补}}$	000.000110									
$\rightarrow 2$	000.000110	1	1	1	0	0	1	1	1	0
$+ [-X]_{\text{补}}$	111.000001									
$[P_4]_{\text{补}}$	111.000111									
$\rightarrow 2$	111.000111	1	1	1	1	0	0	0	1	1

$$\text{故 } [X \times Y]_{\text{补}} = 1.0001 \ 1111 \ 1001$$

2. 将被除数和除数表示出来： $[X]_{\text{原}} = 0.1001$ ， $[Y]_{\text{原}} = 1.1011$ ，原码除法中，商符为 $X_0 \oplus Y_0$ ，商值的计算是按照被除数和除数的绝对值计算的。

$||X|_{\text{原}}| = 00.1001$ ， $||Y|_{\text{原}}| = 00.1011$ ， $||-Y|_{\text{补}}| = 11.0101$ ，运算过程如下：

余数和被除数		商					
	00.1001		0	0	0	0	0
$+[-Y]_{\text{补}}$	11.0101						
$[R_0]_{\text{补}}$	11.1110						
$\leftarrow 1$	11.1100	$\leftarrow 1$	0	0	0	0	0
$+ Y $	00.1011						
$[R_1]_{\text{补}}$	00.0111						
$\leftarrow 1$	00.1110	$\leftarrow 1$	0	0	0	0	1
$+[-Y]_{\text{补}}$	11.0101						
$[R_2]_{\text{补}}$	00.0011						
$\leftarrow 1$	00.0110	$\leftarrow 1$	0	0	0	1	1
$+[-Y]_{\text{补}}$	11.0101						
$[R_3]_{\text{补}}$	11.1011						
$\leftarrow 1$	11.0110	$\leftarrow 1$	0	0	1	1	0
$+ Y $	00.1011						
$[R_4]_{\text{补}}$	00.0001	$\leftarrow 1$	0	1	1	0	0

3. 组合逻辑：①优点：速度快效率高，使用广泛。②缺点：系统复杂，设计复杂性较大，设计难以形式化，设计效率低。

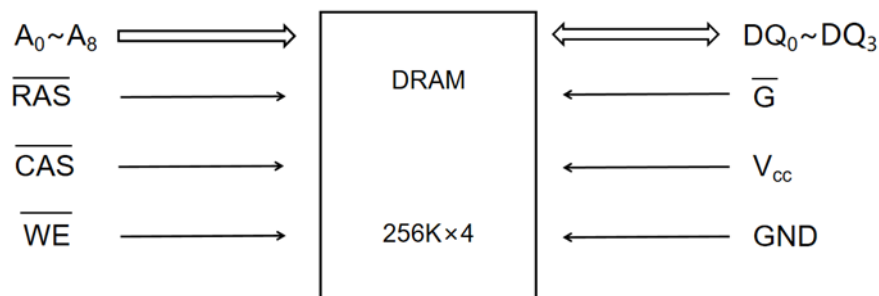
微程序：①优点：规整灵活可维护性好，设计思想简单，在廉价微处理器中广泛应用。②缺点：速度慢，几乎所有指令处理速度都一样。

4. 参考答案如下

(1) $\lceil 64/6 \rceil = 16$, $\lceil 4M/256K \rceil = 16$

共需要 $16 \times 16 = 256$ 片。

(2) 示意图如下（注意，DRAM 地址引脚复用，要少画一半）



(3) $4M = 2^{22}$, 地址码位数是 22。

位扩展用了 16 块芯片，作为片选译码的地址码需要 4 位。

(4) 需要一级 TTL 门 $\lceil 256/8 \rceil = 32$ 个。

为了驱动这 32 个 TTL 门又需要 $\lceil 32/8 \rceil = 4$ 个二级 TTL 门。

一共需要 $32+4=36$ 个。

5. 参考答案如下

(1) 盘面数是 $6 \times 2 - 1 = 11$ ，要去掉一个伺服面。

柱面数是 $(13.2\text{cm} - 6.2\text{cm}) \times 0.5 \times 10 \times 50 + 1 = 1751$ 。

(2) 因为没有谈及扇区的信息，所以此处容量为非格式化容量。

容量为 $C_n = \text{盘面数} \times \text{柱面数} \times \pi \times \text{内直径} \times \text{位密度}$

$$= 11 \times 1751 \times \pi \times 6.2 \times 10 \times 400 \div 8 = 179127300B \approx 170.83MB。$$

(cm 和 mm 的单位转换, 以及除 8 得单位 B 等操作自行注意, 培养意识)

- (3) 数据传输率为 $f = \text{位密度} \times \pi \times \text{内直径} \times \text{转速} = 400 \times 3 \times 62 \times r \div 8 \approx 1.064MB/s。$

转速 r 为 7200rpm, 也即 $\frac{7200}{60} = \frac{120}{1} = 120rps。$ (注意单位)

- (4) 10 台磁盘机需要 4 位台号。

1751 个柱面需要 11 位柱面号。

11 个盘面需要 4 位盘面号。

64 个扇区需要 6 位扇区号。

地址方案为 [台号 4 位, 柱面号 11 位, 盘面号 4 位, 扇区号 6 位]

24……21	20……10	9……6	5……0
台号	柱面号	盘面号	扇区号

6. 参考答案如下

- (1) 求 f_{BMC} , 先求笔绘仪、打印机、终端的数据传输率:

笔绘仪数据传输率: $f_1 = 1/(1000 \times 10^{-6}) = \frac{1}{1000} \times 10^6$

打印机的数据传输率: $f_2 = f_3 = f_4 = 1/(150 \times 10^{-6}) = \frac{1}{150} \times 10^6$

终端的数据传输率: $f_5 = f_6 = \dots = f_{12} = 1/(100 \times 10^{-6}) = \frac{1}{100} \times 10^6$

所以 $f_{BMC} = \sum_{i=1}^{12} f_i = f_1 + 3f_2 + 8f_3 = (\frac{1}{1000} + \frac{3}{150} + \frac{8}{100}) \times 10^6 = 0.101 \times 10^6 = 0.101MB/s$

- (2) 假定能连接 X 台终端

则有 $X = \lfloor (f_{MBC_{max}} - f_{MBC})/f_5 \rfloor = \lfloor (0.16 - 0.101)/0.01 \rfloor = 5$

同志们、朋友们：

回忆之前备考的时候，花几百块买毫无价值的几页 A4 纸、收到的各种资料打上了一层又一层的水印、资料被印在各种花花绿绿的纸上让人头晕目眩，这些情形还历历在目。

2020 年 11 月末，本人在做 2019 年组成原理真题时，因为当时没有值得借鉴的参考答案，群友也大多亦未动笔，或不敢确信自己的答案是否正确，所以导致了做完题后没法订正的局面。在此情形下，我不得不自己亲手做每一题，并按照指定教材绿书和黑书，对每一题进行了详细的解析。然后用有道云笔记的 **Markdow+Mathtype** 录入进去，通过外链分享给了群友，好让大家共同订正，也顺便指出我的答案错误不足之处。后来又考虑到部分同学不便使用有道云笔记，我又将其录入 **Word** 文档，导出为 **PDF** 上传到群里。

从那起，我就计划能够在考完后，抽空把历年真题的答案都整理一遍，排版好，以方便后来备考的同学使用。我更希望我所整理出来的资料，能够在各种素材合规、版权无争议的情况下，以 **CC 4.0** 的相关协议进行许可发行，让大家开源免费自由地下载。

在封面设计上，首先我重绘了一下更简洁的国防科大校徽，主要是将各种字符以线条代替；其次我用国防科大历史上的校名“长沙工学院”代替，目的都是为了避嫌，毕竟这是一所军校，不宜过于张扬。再者也是考虑到大家备考时期，不愿意轻易透露自己的备考目标，所以这样做也有“遮天蔽日”的意图。

在素材使用上，封面校徽是我自行设计的。各类字体也是经过反复验证后，确定是可商用免费字体后才使用的。（其中“隶书”这一字体，在声明版权所属时，我标之以“台湾当局伪教育部”，确保遵守《反分裂国家法》和《新华社新闻信息报道中的禁用词和慎用词》）

对于各类公式，我借助了 **MS Office** 自带的公式编辑器进行编辑。

对于一些逻辑电路图、芯片图、时间线图 etc，我借助 **ioDraw** 网站进行了绘制。

整个 **PDF** 没有打任何恶心的水印，只在开头做了基本的许可，以及页眉处做了标识。

在答案的整理上，我个人的精力和努力必然是有限的，初期工作只有我一人进行，要手打将近十万字符，工作量已然之大；加之初试结束后，知识点忘得特别快，还要回顾知识点，所以进度非常缓慢，我甚至一度想要停止这个计划，毕竟考研与我已经无关了。但好在弟中弟、charles meng、517、卢小帅等朋友进行了帮忙。尤其是弟中弟，先行整理了 03、04、21 的答案，非常认真详细，给了我极大的动力，使得我也继续鼓起劲头。

整体下来，《历年真题答案》质量还是相当高的，无论是从外观排版上，还是内在答案解析上，我相信各位备考同学使用它的时候，会是舒心悦快的。

一份这么详细的答案合集，拿去卖一份 100 块，说实话不过分，很多求学心切的人一定会为之付款的，但我不打算这样，还是将 **PDF** 分享出来，以便大家自行打印或平板上使用。作为计算机专业报考者和学习者，我推崇开源共享精神；作为国防科大的报考者，人民军队为人民的精神影响着我；最后，作为一名精神上的共产党人，红色信念要求我为人民服务。

现在停笔的时间，正是 2021 年 6 月 30 日晚上，写下这些文字的时候，我正听着《红旗颂》，旋律深入人心，从来没感到竟如此厚重。很快就是中国共产党成立一百周年的盛典了，在这伟大的时刻，我有幸能赶工完这部作品，分享给广大朋友们，以表达我新时代下以“开源自由免费共享”为核心内涵的“为人民服务”之理念。

祝愿大家备考顺利。

（本页在物理上的奇数页 83 页上，为了不影响您的备考学习，可以在打印的时候不打这一页，或者全本打印后撕下此页用作草稿纸；或者用 **PDF** 编辑工具删除此页。）