

2009 年全国硕士研究生入学统一考试

计算机科学与技术学科联考计算机学科专业基础综合试题

一、单项选择题：第 1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项最符合试题要求。

1. 为解决计算机主机与打印机之间速度不匹配问题，通常设置一个打印数据缓冲区，主机将要输出的数据依次写入该缓冲区，而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是_____。

- A. 栈 B. 队列 C. 树 D. 图

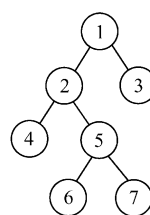
2. 设栈 S 和队列 Q 的初始状态均为空，元素 a, b, c, d, e, f, g 依次进入素出栈后立即进入队列 Q，且 7 个元素出队的顺序是 b, d, c, f, e, a, g，则栈 S 是_____。

- A. 1 B. 2 C. 3 D. 4

3. 给定二叉树如图 A-1 所示。设 N 代表二叉树的根，L 代表根结点的左子树，的右子树。若遍历后的结点序列是 3, 1, 7, 5, 6, 2, 4，则其遍历方式是_____。

- A. LRN B. NRL C. RLN D. RNL

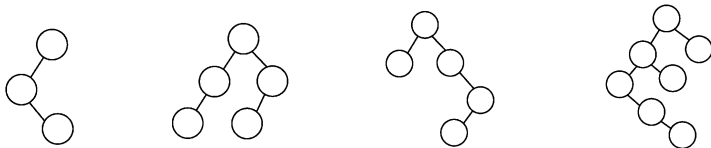
4. 下列二叉排序树中，满足平衡二叉树定义的是_____。



栈 S。若每个元的容量至少

R 代表根结点

图 A-1



- A. B. C. D.

5. 已知一棵完全二叉树的第 6 层（设根为第 1 层）有 8 个叶结点，则该完全二叉树的结点个数最多是_____。

- A. 39 B. 52 C. 111 D. 119

6. 将森林转换为对应的二叉树，若在二叉树中，结点 u 是结点 v 的父结点的父结点，则在原来的森林中，u 和 v 可能具有的关系是_____。

- I. 父子关系 II. 兄弟关系
III. u 的父结点与 v 的父结点是兄弟关系
A. 只有 II B. I 和 II C. I 和 III D. I、II 和 III

7. 下列关于无向连通图特性的叙述中，正确的是_____。

- I. 所有顶点的度之和为偶数
II. 边数大于顶点个数减 1
III. 至少有一个顶点的度为 1
A. 只有 I B. 只有 II C. I 和 II D. I 和 III

8. 下列叙述中，不符合 m 阶 B 树定义要求的是_____。

- A. 根结点最多有 m 棵子树 B. 所有叶结点都在同一层上
C. 各结点内关键字均升序或降序排列 D. 叶结点之间通过指针链接

9. 已知关键字序列 5, 8, 12, 19, 28, 20, 15, 22 是小根堆（最小堆），插入关键字 3，调整后得到的小根堆是_____。

- A. 3, 5, 12, 8, 28, 20, 15, 22, 19
B. 3, 5, 12, 19, 20, 15, 22, 8, 28
C. 3, 8, 12, 5, 20, 15, 22, 28, 19

D. 3, 12, 5, 8, 28, 20, 15, 22, 19

10. 若数据元素序列 11, 12, 13, 7, 8, 9, 23, 4, 5 是采用下列排序方法之一得到的第二趟排序后的结果, 则该排序算法只能是_____。

A. 冒泡排序 B. 插入排序 C. 选择排序 D. 二路归并排序

11. 冯·诺依曼计算机中指令和数据均以二进制形式存放在存储器中, CPU 区分它们的依据是_____。

A. 指令操作码的译码结果 B. 指令和数据的寻址方式
C. 指令周期的不同阶段 D. 指令和数据所在的存储单元

12. 一个 C 语言程序在一台 32 位机器上运行。程序中定义了三个变量 x、y 和 z, 其中 x 和 z 为 int 型, y 为 short 型。当 x=127, y=-9 时, 执行赋值语句 z=x+y 后, x、y 和 z 的值分别是_____。

A. x=0000007FH, y=FFF9H, z=00000076H
B. x=0000007FH, y=FFF9H, z=FFFF0076H
C. x=0000007FH, y=FFF7H, z=FFFF0076H
D. x=0000007FH, y=FFF7H, z=00000076H

13. 浮点数加、减运算过程一般包括对阶、尾数运算、规格化、舍入和判溢出等步骤。设浮点数的阶码和尾数均采用补码表示, 且位数分别为 5 位和 7 位 (均含 2 位符号位)。若有两个数 $X=2^7 \times 29/32$, $Y=2^5 \times 5/8$, 则用浮点加法计算 X+Y 的最终结果是_____。

A. 00111 1100010 B. 00111 0100010
C. 01000 0010001 D. 发生溢出

14. 某计算机的 Cache 共有 16 块, 采用 2 路组相联映射方式 (即每组 2 块)。每个主存块大小为 32B, 按字节编址。主存 129 号单元所在主存块应装入到的 Cache 组号是_____。

A. 0 B. 1 C. 4 D. 6

15. 某计算机主存容量为 64KB, 其中 ROM 区为 4KB, 其余为 RAM 区, 按字节编址。现要用 2K×8 位的 ROM 芯片和 4K×4 位的 RAM 芯片来设计该存储器, 则需要上述规格的 ROM 芯片数和 RAM 芯片数分别是_____。

A. 1、15 B. 2、15 C. 1、30 D. 2、30

16. 某机器字长为 16 位, 主存按字节编址, 转移指令采用相对寻址, 由两个字节组成, 第一字节为操作码字段, 第二字节为相对位移量字段。假定取指令时, 每取一个字节 PC 自动加 1。若某转移指令所在主存地址为 2000H, 相对位移量字段的内容为 06H, 则该转移指令成功转移后的目标地址是_____。

A. 2006H B. 2007H C. 2008H D. 2009H

17. 下列关于 RISC 的叙述中, 错误的是_____。

A. RISC 普遍采用微程序控制器
B. RISC 大多数指令在一个时钟周期内完成
C. RISC 的内部通用寄存器数量相对 CISC 多
D. RISC 的指令数、寻址方式和指令格式种类相对 CISC 少

18. 某计算机的指令流水线由四个功能段组成, 指令流经各功能段的时间 (忽略各功能段之间的缓存时间) 分别为 90ns、80ns、70ns、和 60ns, 则该计算机的 CPU 时钟周期至少是_____。

A. 90ns B. 80ns C. 70ns D. 60ns

19. 相对于微程序控制器, 硬布线控制器的特点是_____。

A. 指令执行速度慢, 指令功能的修改和扩展容易
B. 指令执行速度慢, 指令功能的修改和扩展难
C. 指令执行速度快, 指令功能的修改和扩展容易
D. 指令执行速度快, 指令功能的修改和扩展难

20. 假设某系统总线在一个总线周期中并行传输 4B 信息, 一个总线周期占用 2 个时钟周期, 总线时钟频率为 10MHz, 则总线带宽是_____。

A. 10MB/s B. 20MB/s C. 40MB/s D. 80MB/s

21. 假设某计算机的存储系统由 Cache 和主存组成, 某程序执行过程中访存 1000 次, 其中访问 Cache 缺失 (未命中) 50 次, 则 Cache 的命中率是_____。

A. 5% B. 9.5% C. 50% D. 95%

38. 主机甲与主机乙之间已建立一个 TCP 连接, 主机甲向主机乙发送了两个连续的 TCP 段, 分别包含 300B 和 500B 的有效载荷, 第一个段的序列号为 200, 主机乙正确接收到两个段后, 发送给主机甲的确认序列号是_____。

- A. 500 B. 700 C. 800 D. 1000

39. 一个 TCP 连接总是以 1KB 的最大段长发送 TCP 段, 发送方有足够多的数据要发送。当拥塞窗口为 16KB 时发生了超时, 如果接下来的 4 个 RTT (往返时间) 时间内的 TCP 段的传输都是成功的, 那么当第 4 个 RTT 时间内发送的所有 TCP 段都得到肯定应答时, 拥塞窗口大小是_____。

- A. 7KB B. 8KB C. 9KB D. 16KB

40. FTP 客户和服务端间传递 FTP 命令时, 使用的连接是_____。

- A. 建立在 TCP 之上的控制连接 B. 建立在 TCP 之上的数据连接
C. 建立在 UDP 之上的控制连接 D. 建立在 UDP 之上的数据连接

二、综合应用题: 第 41~47 题, 共 70 分。

41. (10 分) 带权图 (权值非负, 表示边连接的两顶点间的距离) 的最短路径问题是找出从初始顶点到目标顶点之间的一条最短路径。假设从初始顶点到目标顶点之间存在路径, 现有一种解决该问题的方法:

- ① 设最短路径初始时仅包含初始顶点, 令当前顶点 u 为初始顶点;
- ② 选择离 u 最近且尚未在最短路径中的一个顶点 v , 加入到最短路径中, 修改当前顶点 $u=v$;
- ③ 重复步骤②, 直到 u 是目标顶点时为止。

请问上述方法能否求得最短路径? 若该方法可行, 请证明之; 否则, 请举例说明。

42. (15 分) 已知一个带有表头结点的单链表, 结点结构为:

data	link
------	------

假设该链表只给出了头指针 list。在不改变链表的前提下, 请设计一个尽可能高效的算法, 查找链表中倒数第 k 个位置上的结点 (k 为正整数)。若查找成功, 算法输出该结点的 data 域的值, 并返回 1; 否则, 只返回 0。要求:

- 1) 描述算法的基本设计思想。
- 2) 描述算法的详细实现步骤。

3) 根据设计思想和实现步骤, 采用程序设计语言描述算法 (使用 C、C++ 或 Java 语言实现), 关键之处请给出简要注释。

43. (8 分) 某计算机的 CPU 主频为 500MHz, CPI 为 5 (即执行每条指令平均需 5 个时钟周期)。假定某外设的数据传输率为 0.5MB/s, 采用中断方式与主机进行数据传送, 以 32 位为传输单位, 对应的中断服务程序包含 18 条指令, 中断服务的其他开销相当于 2 条指令的执行时间。请回答下列问题, 要求给出计算过程。

1) 在中断方式下, CPU 用于该外设 I/O 的时间占整个 CPU 时间的百分比是多少?

2) 当该外设的数据传输率达到 5MB/s 时, 改用 DMA 方式传送数据。假定每次 DMA 传送块大小为 5000B, 且 DMA 预处理和后处理的总开销为 500 个时钟周期, 则 CPU 用于该外设 I/O 的时间占整个 CPU 时间的百分比是多少 (假设 DMA 与 CPU 之间没有访存冲突)?

44. (13 分) 某计算机字长为 16 位, 采用 16 位定长指令字结构, 部分数据通路结构如图 A-2 所示, 图中所有控制信号为 1 时表示有效、为 0 时表示无效。例如, 控制信号 MDRinE 为 1 表示允许数据从 DB 打入 MDR, MDRin 为 1 表示允许数据从内总线打入 MDR。假设 MAR 的输出一直处于使能状态。加法指令“ADD (R1), R0”的功能为 $(R0) + ((R1)) \rightarrow (R1)$, 即将 R0 中的数据与 R1 的内容所指主存单元的数据相加, 并将结果送入 R1 的内容所指主存单元中保存。

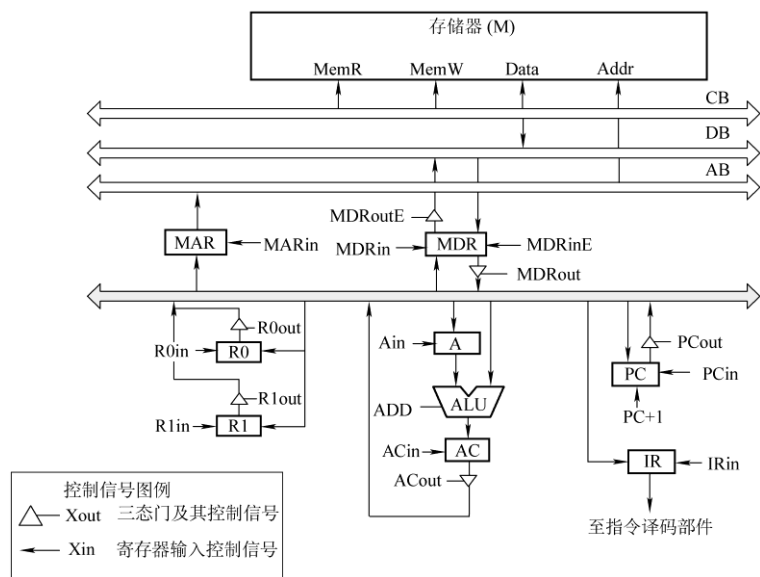


图 A-2

表 A-1 给出了上述指令取指和译码阶段每个节拍（时钟周期）的功能和有效控制信号，请按表中描述方式用表格列出指令执行阶段每个节拍的功能和有效控制信号。

表 A-1

时钟	功能	有效控制信号
C1	$MAR \leftarrow (PC)$	PCout, MARin
C2	$MDR \leftarrow M(MDR)$ $PC \leftarrow (PC) + 1$	MemR, MDRinE, PC+1
C3	$IR \leftarrow (MDR)$	MDRout, IRin
C4	指令译码	无

45. (7 分) 三个进程 P1、P2、P3 互斥使用一个包含 N (N>0) 个单元的缓冲区。P1 每次用 produce() 生成一个正整数并用 put() 送入缓冲区某一空单元中；P2 每次用 getodd() 从该缓冲区中取出一个奇数并用 countodd() 统计奇数个数；P3 每次用 geteven() 从该缓冲区中取出一个偶数并用 counteven() 统计偶数个数。请用信号量机制实现这三个进程的同步与互斥活动，并说明所定义信号量的含义（要求用伪代码描述）。

46. (8 分) 请求分页管理系统中，假设某进程的页表内容见表 A-2。

表 A-2

页号	页框 (Page Frame) 号	有效位 (存在位)
0	101H	1
1		0
2	254H	1

页面大小为 4KB，一次内存的访问时间为 100ns，一次快表 (TLB) 的访问时间为 10ns，处理一次缺页的平均时间为 10^8 ns (已含更新 TLB 和页表的时间)，进程的驻留集大小固定为 2，采用最近最少使用置换算法 (LRU) 和局部淘汰策略。假设①TLB 初始为空；②地址转换时先访问 TLB，若 TLB 未命中，再访问页表 (忽略访问页表之后的 TLB 更新时间)；③有效位为 0 表示页面不在内存中，产生缺页中断，缺页中断处理后，返回到产生缺页中断的指令处重新执行。设有虚地址访问序列 2362H、

1565H、25A5H，请问：

- 依次访问上述三个虚地址，各需多少时间？给出计算过程。
- 基于上述访问序列，虚地址 1565H 的物理地址是多少？请说明理由。

47. (9 分) 某网络拓扑如图 A-3 所示，路由器 R1 通过接口 E1、E2 分别连接局域网 1、局域网 2，通过接口 L0 连接路由器 R2，并通过路由器 R2 连接域名服务器与互联网。R1 的 L0 接口的 IP 地址是 202.118.2.1，R2 的 L0 接口的 IP 地址是 202.118.2.2，L1 接口的 IP 地址是 130.11.120.1，E0 接口的 IP 地址是 202.118.3.1，域名服务器的 IP 地址是 202.118.3.2。

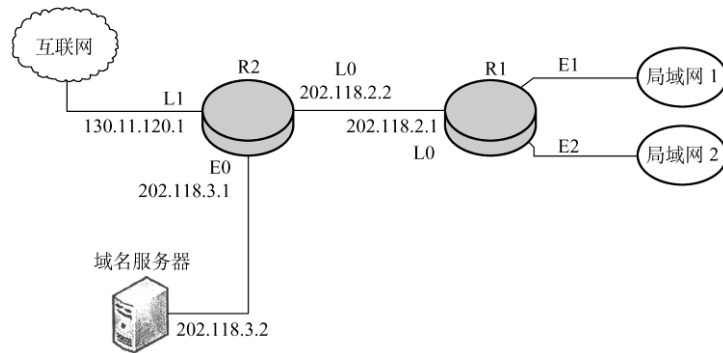


图 A-3

R1 和 R2 的路由表结构为:

目的网络 IP 地址	子网掩码	下一跳 IP 地址	接口
------------	------	-----------	----

- 1) 将 IP 地址空间 202.118.1.0/24 划分为 2 个子网，分别分配给局域网 1、局域网 2，每个局域网需分配的 IP 地址数不少于 120 个。请给出子网划分结果，说明理由或给出必要的计算过程。
- 2) 请给出 R1 的路由表，使其明确包括到局域网 1 的路由、局域网 2 的路由、域名服务器的主机路由和互联网的路由。
- 3) 请采用路由聚合技术，给出 R2 到局域网 1 和局域网 2 的路由。

2009 年计算机学科专业基础综合试题参考答案

一、单项选择题

- | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 1. B | 2. C | 3. D | 4. B | 5. C | 6. B | 7. A | 8. D |
| 9. A | 10. B | 11. C | 12. D | 13. D | 14. C | 15. D | 16. C |
| 17. A | 18. A | 19. D | 20. B | 21. D | 22. A | 23. D | 24. D |
| 25. C | 26. A | 27. C | 28. B | 29. A | 30. A | 31. B | 32. A |
| 33. B | 34. B | 35. C | 36. A | 37. D | 38. D | 39. C | 40. A |

二、综合应用题

41. 解答:

该方法不一定能 (或不能) 求得最短路径。

举例说明:

图 A-4 中, 设初始顶点为 1, 目标顶点为 4, 欲求从顶点 1 到顶点 4 之间的最短路径, 显然这两点之间的最短路径长度为 2。利用给定方法求得的路径长度为 3, 但这条路径并不是这两点之间的最短路径。

图 A-5 中, 设初始顶点为 1, 目标顶点为 3, 欲求从顶点 1 到顶点 3 之间的最短路径。利用给定的方法, 无法求出顶点 1 到顶点 3 的路径。

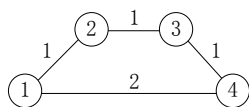


图 A-4

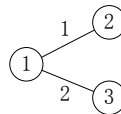


图 A-5

42. 解答:

1) 算法的基本设计思想:

问题的关键是设计一个尽可能高效的算法, 通过链表的一趟遍历, 找到倒数第 k 个结点的位置。算法的基本设计思想: 定义两个指针变量 p 和 q , 初始时均指向头结点的下一个结点 (链表的第一个结点)。 p 指针沿链表移动, 当 p 指针移动到第 k 个结点时, q 指针开始与 p 指针同步移动; 当 p 指针移动到最后一个结点时, q 指针所指示结点为倒数第 k 个结点。以上过程对链表仅进行一遍扫描。

2) 算法的详细实现步骤:

① $\text{count}=0$, p 和 q 指向链表表头结点的下一个结点;

② 若 p 为空, 转⑤;

③ 若 count 等于 k , 则 q 指向下一个结点; 否则, $\text{count}=\text{count}+1$;

④ p 指向下一个结点, 转②;

⑤ 若 count 等于 k , 则查找成功, 输出该结点的 data 域的值, 返回 1; 否则, 说明 k 值超过了线性表的长度, 查找失败, 返回 0;

⑥ 算法结束。

3) 算法实现:

```
typedef int ElemType;           //链表数据的类型定义
typedef struct LNode{           //链表结点的结构定义
    ElemType    data;           //结点数据
    struct LNode *link;         //结点链接指针
} *LinkList;
int Search_k(LinkList list,int k){
```



```

//查找链表 list 倒数第 k 个结点, 并输出该结点 data 域的值
LinkedList p=list->link,q=list->link;           //指针 p、q 指示第一个结点
int count=0;
while(p!=NULL){                                  //遍历链表直到最后一个结点
    if(count<k) count++;                          //计数, 若 count<k 只移动 p
    else q=q->link;p=p->link;                      //之后让 p、q 同步移动
} //while
if(count<k)
    return 0;                                     //查找失败返回 0
else {
    printf("%d",q->data);                          //否则打印并返回 1
    return 1;
}
} //Search_k

```

提示: 算法程序题, 如果能够写出数据结构类型定义, 正确的算法思想都会至少给一半以上分数, 如果能用伪代码写出自然更好, 比较复杂的地方可以直接用文字表达。

43. 解答:

1) 按题意, 外设每秒传送 0.5MB, 中断时每次传送 4B。中断方式下, CPU 每次用于数据传送的时钟周期为 $5 \times 18 + 5 \times 2 = 100$ 。

为达到外设 0.5MB/s 的数据传输率, 外设每秒申请的中断次数为 $0.5\text{MB}/4\text{B} = 125\,000$ 。

1s 内用于中断的开销为 $100 \times 125\,000 = 12\,500\,000 = 12.5\text{M}$ 个时钟周期。

CPU 用于外设 I/O 的时间占整个 CPU 时间的百分比为 $12.5\text{M}/500\text{M} = 2.5\%$ 。

2) 当外设数据传输率提高到 5MB/s 时, 改用 DMA 方式传送, 每次 DMA 传送 5 000B, 1s 内需产生的 DMA 次数为 $5\text{MB}/5\,000\text{B} = 1\,000$ 。

CPU 用于 DMA 处理的总开销为 $1\,000 \times 500 = 500\,000 = 0.5\text{M}$ 个时钟周期。

CPU 用于外设 I/O 的时间占整个 CPU 时间的百分比为 $0.5\text{M}/500\text{M} = 0.1\%$ 。

44. 解答:

一条指令的执行过程通常由取指、译码和执行 3 个步骤完成, 本题中取指用 3 个节拍、译码用 1 个节拍, 执行加法运算并把结果写入主存如何完成呢? 包括划分执行步骤、确定完成的功能、要提供的控制信号, 这是本题要测试的内容。为回答这个问题, 首先要看清图 A-2 中给出的部件组成情况和信息传送的路径。

要完成的功能是 $(R0) + ((R1)) \rightarrow (R1)$, 从图 A-2 中看到:

1) R0、R1 都有送自己的内容到内总线的路径, 控制信号分别是 R0out 和 R1out。

2) ALU 加运算, 2 个数据由工作寄存器 A 和内总线提供, 控制信号是 Add; A 只接收内总线的内容, 控制信号是 Ain; 结果需存 AC, 控制信号是 ACin; AC 的内容可送内总线, 控制信号是 ACout。

3) PC 可接收内总线的内容, 还可增 1, 控制信号是 PCin 和 PC+1, PC 的内容可送内总线, 控制信号是 PCout。

4) 指令寄存器 IR 可接收内总线的内容, 控制信号是 IRin。

5) 读/写存储器时, 地址由 MAR 经 AB 提供, MAR 只接收总线上的信息, 控制信号是 MARin。

6) 读存储器, 提供读命令 MemR, 并通过 DB 送入 MDR, 控制信号是 MDRinE; MDR 的内容可送入总线, 控制信号是 MDRout。

7) 写存储器, 提供写命令 MemW, 数据由 MDR 通过 DB 送到存储器的数据引脚, 控制信号是 MDRoutE。

然后是划分执行步骤、确定每步完成的功能、需要提供的控制信号。这是由指令应完成的功能、计算机硬件的实际组成情况和信息传送的可用路径共同决定的, 基本原则是步骤越少越好。硬件电路要能支持, 可以有多种方案, 解题时应参照给出的答题格式, 即取指和译码阶段的表 A-1 的内容, 但不必把表已有的内容再抄一遍。

划分指令执行步骤、确定每步完成的功能、给出需要提供的控制信号。

请注意, $(R0) + ((R1))$ 表示: R0 寄存器的内容与 R1 作地址从主存中读出来的数据完成加法运算; 而 $\rightarrow (R1)$ 表示把 R1 的内容作为主存储器的地址完成写主存操作。为防止出现误解, 题中还特地对此作了文字说明。这条指令的功能是先到主存储器取一个数, 之后运算, 再将结果写回主存储器。

1) 执行相加运算, 需把存储器中的数据读出, 为此首先送地址, 将 R1 的内容送 MAR, 控制信号是 R1out、MARin。

2) 启动读主存操作, 读出的内容送入 MDR, 控制信号是 MemR、MDRinE。还可同时把 R0 的内容经内总线送入

A, 用到的控制信号是 R0out、Ain。

3) 执行加法运算, 即 A 的内容与 MDR 的内容相加, 结果保存到 AC, 控制信号是 MDRout、Add、ACin。

4) 要把 AC 的内容写入主存, 由于 R1 的内容已经在 MAR 中, 地址已经有了, 但需要把写入的数据 (已经在 AC 中) 经内总线送入 MDR, 控制信号是 ACout、MDRin。

5) 给出写主存的命令, 把 MDR 的内容经 DB 送存储器的数据线引脚, 执行写操作, 控制信号是 MDRoutE、MemW。

这几个步骤是有先后次序的, 前面的完成了, 下一步才可以执行, 保证了不会产生硬件线路的冲突。请注意, 使用最为频繁的是内总线, 它在任何时刻只能接收一个输入数据, 并且向内总线发送信息的电路只能以三态门器件连接到内总线, 5 个向内总线发送信息的控制信号 (ACout、PCout、R0out、R1out、MDRout) 最多只能有一个为 1, 其他 4 个必须全为 0, 或者 5 个全为 0。

仔细看一下, 发现可以把第 2) 个步骤的操作划分到两个步骤中完成, 一个步骤中安排 MDR 接收从存储器中读出的内容, 到另外一个步骤实现 R0 的内容送入 A, 这多用了个操作步骤, 指令的执行速度会变慢。有些解题者在写存储器之前, 还会再执行一次把 R1 的内容送 MAR, 尽管无此必要, 但不属于原理上的错误。

当然还可以有其他的设计结果。

解题时这些叙述内容不必写出来 (这里写出这些内容是帮助大家领会本题要测试的知识点和指令的执行过程), 直接按照已经给出的表格的形式、提供的填写办法把设计的表格及其内容填好就可以了。

请注意, 题目表格内容 (告诉你答题的格式和答题内容的表达方式) 与你答题的表格内容合在一起才是这条指令完整的执行过程, 千万不要产生任何错觉。

参考答案一见表 A-3。

“ $A \leftarrow (R0)$ ”也可在 C7 “ $AC \leftarrow (MDR) + (A)$ ”之前单列的一个时钟周期内执行。

参考答案二见表 A-4。

表 A-3

时钟	功能	有效控制信号
C5	$MAR \leftarrow (R1)$	R1out, MARin
C6	$MDR \leftarrow M(MAR)$ $A \leftarrow (R0)$	MemR, MDRinE, R0out, Ain
C7	$AC \leftarrow (MDR) + (A)$	MDRout, Add, ACin
C8	$MDR \leftarrow (AC)$	ACout, MDRin
C9	$M(MAR) \leftarrow (MDR)$	MDRoutE, MemW

表 A-4

时钟	功能	有效控制信号
C5	$MAR \leftarrow (R1)$	R1out, MARin
C6	$MDR \leftarrow M(MAR)$	MemR, MDRinE
C7	$A \leftarrow (MDR)$	MDRout, Ain
C8	$AC \leftarrow (A) + (R0)$	R0out, Add, ACin
C9	$MDR \leftarrow (AC)$	ACout, MDRin
C10	$M(MAR) \leftarrow (MDR)$	MDRoutE, MemW

45. 解答:

定义信号量 odd 控制 P1 与 P2 之间的同步; even 控制 P1 与 P3 之间的同步; empty 控制生产者与消费者之间的同步; mutex 控制进程间互斥使用缓冲区。程序如下:

```
semaphore odd=0,even=0,empty=N,mutex=1;
P1 ( )
{
    x=produce();           //生成一个数
    P(empty);             //判断缓冲区是否有空单元
    P(mutex);             //缓冲区是否被占用
    Put();
    V(mutex);             //释放缓冲区
    if (x%2==0)
        V(even);          //如果是偶数, 向 P3 发出信号
    else
        V(odd);           //如果是奇数, 向 P2 发出信号
}
P2 ( )
{
    P(odd);               //收到 P1 发来的信号, 已产生一个奇数
    P(mutex);             //缓冲区是否被占用
    getodd();
    V(mutex);             //释放缓冲区
    V(empty);             //向 P1 发信号, 多出一个空单元
}
```

```

        countodd();
    }
    P3( )
    {
        P(even);           //收到 P1 发来的信号, 已产生一个偶数
        P(mutex);         //缓冲区是否被占用
        geteven();
        V(mutex);          //释放缓冲区
        V(empty);          //向 P1 发信号, 多出一个空单元
        counteven();
    }

```

46. 解答:

1) 根据页式管理的工作原理, 应先考虑页面大小, 以便将页号和页内位移分解出来。页面大小为 4KB, 即 2^{12} , 则得到页内位移占虚地址的低 12 位, 页号占剩余高位。可得三个虚地址的页号 P 如下 (十六进制的一位数字转换成 4 位二进制, 因此, 十六进制的低三位正好为页内位移, 最高位为页号):

2362H: P=2, 访问快表 10ns, 因初始为空, 访问页表 100ns 得到页框号, 合成物理地址后访问主存 100ns, 共计 $10\text{ns}+100\text{ns}+100\text{ns}=210\text{ns}$ 。

1565H: P=1, 访问快表 10ns, 落空, 访问页表 100ns 落空, 进行缺页中断处理 10^8ns , 访问快表 10ns, 合成物理地址后访问主存 100ns, 共计 $10\text{ns}+100\text{ns}+10^8\text{ns}+10\text{ns}+100\text{ns}=100\ 000\ 220\text{ns}$ 。

25A5H: P=2, 访问快表, 因第一次访问已将该页号放入快表, 因此花费 10ns 便可合成物理地址, 访问主存 100ns, 共计 $10\text{ns}+100\text{ns}=110\text{ns}$ 。

2) 当访问虚地址 1565H 时, 产生缺页中断, 合法驻留集为 2, 必须从页表中淘汰一个页面, 根据题目的置换算法, 应淘汰 0 号页面, 因此 1565H 的对应页框号为 101H。由此可得 1565H 的物理地址为 101565H。

47. 解答:

1) CIDR 中的子网号可以全 0 或全 1, 但主机号不能全 0 或全 1。

因此若将 IP 地址空间 202.118.1.0/24 划分为 2 个子网, 且每个局域网需分配的 IP 地址个数不少于 120 个, 子网号至少要占用一位。

由 $2^6-2 < 120 < 2^7-2$ 可知, 主机号至少要占用 7 位。

由于源 IP 地址空间的网络前缀为 24 位, 因此主机号位数+子网号位数=8。

综上可得主机号位数为 7, 子网号位数为 1。

因此子网的划分结果为子网 1: 202.118.1.0/25, 子网 2: 202.118.1.128/25。

地址分配方案: 子网 1 分配给局域网 1, 子网 2 分配给局域网 2; 或子网 1 分配给局域网 2, 子网 2 分配给局域网 1。

2) 由于局域网 1 和局域网 2 分别与路由器 R1 的 E1、E2 接口直接相连, 因此在 R1 的路由表中, 目的网络为局域网 1 的转发路径是直接通过接口 E1 转发的, 目的网络为局域网 2 的转发路径是直接通过接口 E2 转发的。由于局域网 1、2 的网络前缀均为 25 位, 因此它们的子网掩码均为 255.255.255.128。

根据题意, R1 专门为域名服务器设定了一个特定的路由表项, 因此该路由表项中的子网掩码应为 255.255.255.255。对应的下一跳转发地址是 202.118.2.2, 转发接口是 L0。

根据题意, 到互联网的路由实质上相当于一个默认路由, 默认路由一般写作 0/0, 即目的地址为 0.0.0.0, 子网掩码为 0.0.0.0。对应的下一跳转发地址是 202.118.2.2, 转发接口是 L0。

综上可得到路由器 R1 的路由表如下。

若子网 1 分配给局域网 1, 子网 2 分配给局域网 2, 见表 A-5。

表 A-5

目的网络 IP 地址	子网掩码	下一跳 IP 地址	接口
202.118.1.0	255.255.255.128		E1
202.118.1.128	255.255.255.128		E2
202.118.3.2	255.255.255.255	202.118.2.2	L0
0.0.0.0	0.0.0.0	202.118.2.2	L0

若子网 1 分配给局域网 2，子网 2 分配给局域网 1，见表 A-6。

表 A-6

目的网络 IP 地址	子网掩码	下一跳 IP 地址	接口
202.118.1.128	255.255.255.128		E1
202.118.1.0	255.255.255.128		E2
202.118.3.2	255.255.255.255	202.118.2.2	L0
0.0.0.0	0.0.0.0	202.118.2.2	L0

3) 局域网 1 和局域网 2 的地址可以聚合为 202.118.1.0/24，而对于路由器 R2 来说，通往局域网 1 和局域网 2 的转发路径都是从 L0 接口转发，因此采用路由聚合技术后，路由器 R2 到局域网 1 和局域网 2 的路由，见表 A-7。

表 A-7

目的网络 IP 地址	子网掩码	下一跳 IP 地址	接口
202.118.1.0	255.255.255.0	202.118.2.1	L0