

考试科目代码及名称 833 真题回忆版

考试时间 2021 年 12 月 26 号下午 (3 小时)

答题要求：所有答案（填空题按照标号写）必须写在答题纸上，
写在试卷上一律作废，准考证号写在指定位置！

一、选择题

1. ENIAC 所用的主要元件是 ()
 - A、集成电路
 - B、晶体管
 - C、电子管
 - D、以上各项都不对
2. 所谓三总线结构的计算机是指 ()
 - A、地址线、数据线和控制线三组传输线
 - B、I/O 总线、主存总线和 DMA 总线三组传输线
 - C、I/O 总线、主存总线和系统总线三组传输线
 - D、I/O 总线、主存总线和地址总线
3. 某计算机字长是 16 位，它的存储容量是 1MB，按字编址，它的寻址范围是 ()
 - A、512K
 - B、1M
 - C、512KB
 - D、1MB
4. 在磁表面存储器的记录方式中 ()
 - A、不归零制和归零制的记录密度是一样的
 - B、不归零制有自同步能力
 - C、不归零制的记录方中不需要同步信号，故记录密度比归零制高
 - D、不归零制记录方式由于磁头线圈中始终有电流，因此抗干扰性能好
5. CPU 通过 () 启动通道。
 - A、执行通道命令
 - B、执 I/O 指令
 - C、发出中断请求
 - D、发出 DMA 请求
6. 21. 设 $[x]_2 = 1.X_1X_2X_3X_4$ 当满足下列 () 时， $x > -1/2$ 成立。
 - A、 X_1 必须为 1， $X_2 \sim X_4$ 至少有一个为 1
 - B、 X_1 必须为 1， $X_2 \sim X_4$ 任意
 - C、 X_1 必须为 0， $X_2 \sim X_4$ 至少有一个为 1
 - D、 X_1 必须为 0， $X_2 \sim X_4$ 任意
7. 在下述有关不恢复余数法何时需恢复余数的说法中，() 是正确的。
 - A、最后一次余数为正时，要恢复一次余数
 - B、最后一次余数为负时，要恢复一次余数

- C、最后一次余数为 0 时，要恢复一次余数
D、任何时候都不恢复余数
8. 堆栈寻址方式中，设 A 为累加器，SP 为堆栈指示器，M_{gs} 为 SP 指示的栈顶单元，如果进栈操作的动作顺序是
(SP) - 1 → SP
(A) → M_{sp}
那么出栈操作的动作顺序应为 ()
A、(M_{sp}) → A, (SP) + 1 → SP
B、(M_{sp}) → A, (SP) - 1 → SP
C、(SP) + 1 → SP, (M_{sp}) → A
C、(SP) - 1 → SP, (M_{sp}) → A
9. 中断向量可提供 ()
A、被选中设备的地址
B、传送数据的起始地址
C、中断服务程序入口地址
D、主程序的断点地址
10. 水平型微指令的特点是 ()
A、一次可以完成多个操作
B、微指令的操作控制字段不进行编码
C、微指令的格式简短
D、不易并行处理
11. 某机有四级中断，优先级从高到低为 1 → 2 → 3 → 4。若将优先级顺序修改，改后 1 级中断的屏蔽字为 1011，2 级中断的屏蔽字为 1111，3 级中断的屏蔽字为 0011，4 级中断的屏蔽字为 0001，则修改后的优先顺序从高到低为 ()
A、3—2—1—4
B、1-3—>4→2
C、3→1→2→4
D、2→1—3s4
12. 下列 () 总线是显示卡专用的局部总线。
A、USB
B、AGP
C、SAS
D、PCI
13. 下面关于数据结构的说法，正确的是 ()
A. 逻辑上相邻的元素在存储位置上一定相邻
B. 算法的实现依赖于采用的逻辑结构
C. 数据的逻辑结构分为线性结构和非线性结构
D. 以上说法都正确
14. 带尾指针的单循环链表在其表尾进行插入结点和删除尾结点的操作，设链表长度为 n，则两种操作的时间复杂度为 ()
A. O(1)、O(1) B. O(1)、O(n) C. O(n)、O(1) D. O(n)、O(n)
15. 若 front 和 rear 分别表示链接队列的队头指针与队尾指针，则向队列中插入一个由 p 指的新元素的过程是依次执行 ()

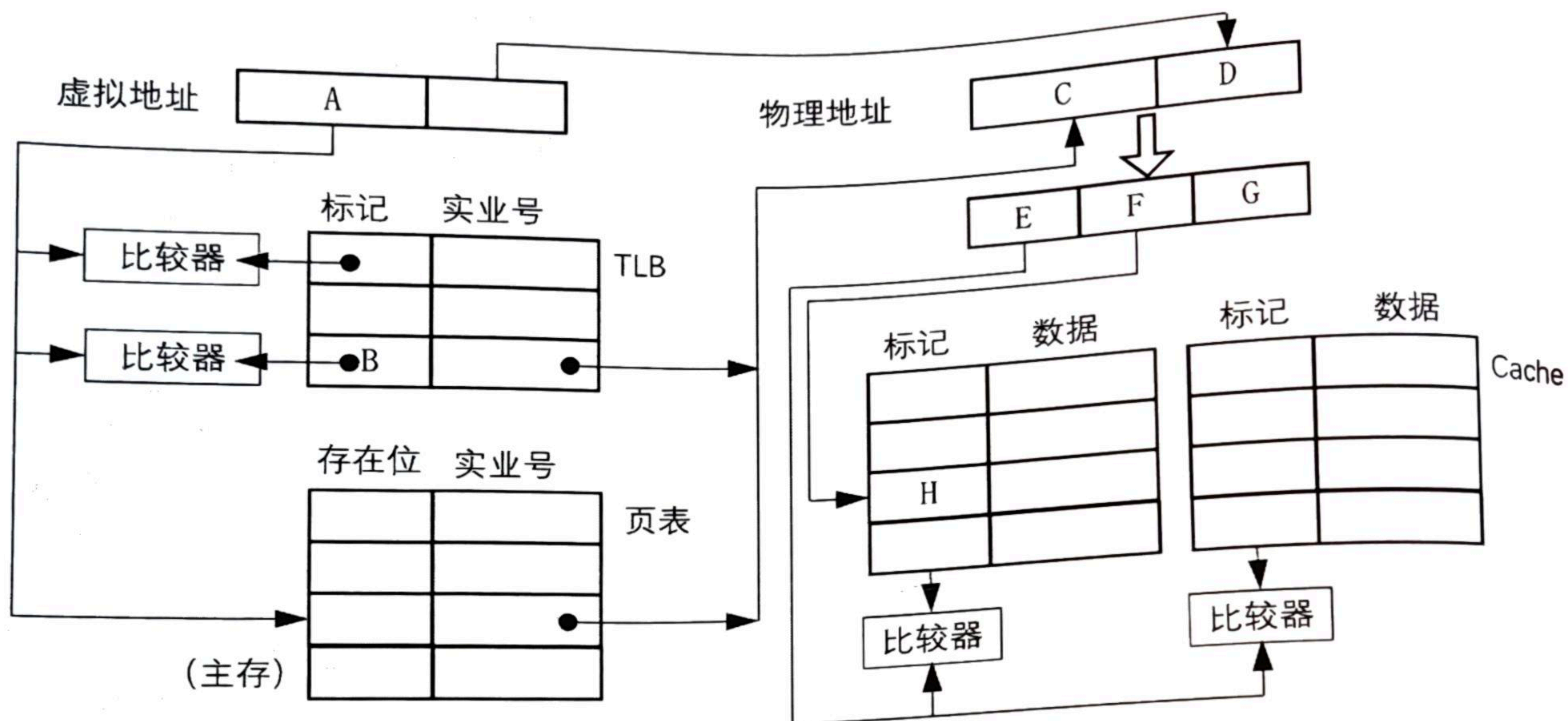
- A. $p \rightarrow \text{link} = \text{rear}; \text{rear} = p;$
 B. $p \rightarrow \text{link} = \text{front}; \text{front} = p.$
 C. $\text{rear} \rightarrow \text{link} = p; \text{rear} = p;$
 D. $\text{front} \rightarrow \text{link} = p; \text{front} = p;$
16. 广义表 $L = (a, ((c, d, (e, f)), g))$, 取出元素 g 的运算是 ()
 A. $\text{tail}(\text{tail}(L))$
 B. $\text{tail}(\text{head}(\text{tail}(L)))$
 C. $\text{head}(\text{tail}(\text{tail}(\text{tail}(L))))$
 D. $\text{head}(\text{tail}(\text{head}(\text{tail}(L))))$
17. 已知模式串 $P = \text{"abababababab"}$, 则其 Next 函数值为 ()
 A. 0 1 1 2 3 1 2 3 2 3 2 3 1 2
 B. 0 1 1 2 3 1 2 3 4 5 4 3 1 2
 C. 0 1 1 2 3 1 2 3 4 5 4 5 6 2
 D. 0 1 1 2 3 1 2 3 4 5 4 5 6 7
18. 若二叉树的先序序列和后序序列正好相同, 则该二叉树一定是 () 的二叉树。
 A. 任一结点无左孩子 B. 任一结点无右孩子
 C. 空或只有一个结点 D. 高度等于其结点数
19. 设有两个无向图 $G = (V, E)$, G' 是 G 的生成树, 以下说法不正确的是 ()
 A. G' 是 G 的子图 B. G' 是 G 的极大连通子图
 C. G' 是 G 的无环子图 D. G' 是 G 的极小连通子图
20. 下列说法不正确的是 ()
 A. 哈夫曼树的形态不是唯一的, 但对具有一组权值的各哈夫曼树的 WPL 值是唯一的
 B. 一棵树采用孩子兄弟表示法所建立的存储结构与它所对应的二叉树的二叉链表存储结构式完全相同的。
 C. n 个顶点连通图 G 的生成树必定包含且仅包含 G 的 $n-1$ 条边
 D. 有向图的邻接矩阵一定是对称矩阵
21. 高度为 5 的 5 阶 B-树至少有 (C) 个结点, 至多有 () 个结点。
 A. 8 1, 3 1 2 4 B. 8 2, 3 1 2 4
 C. 8 1, 7 8 1 D. 8 2, 7 8 1
22. 下列排序算法中, 能够在一趟排序后选出一个元素放在最终位置的是 ()
 A. 归并排序 B. 堆排序 C. 直接插入排序 D. 基数排序

二、简要分析题

- 指令字中有哪些字段? 各有何作用? 如何确定这些字段的位数?
- 在浮点补码加减运算中, 当尾数运算结果的符号位为 01 或 10 时, 即表示运算结果溢出。这种说法是否正确, 为什么?
- 什么是“程序访问的局部性”? 存储系统中哪一级采用了程序访问的局部性原理?

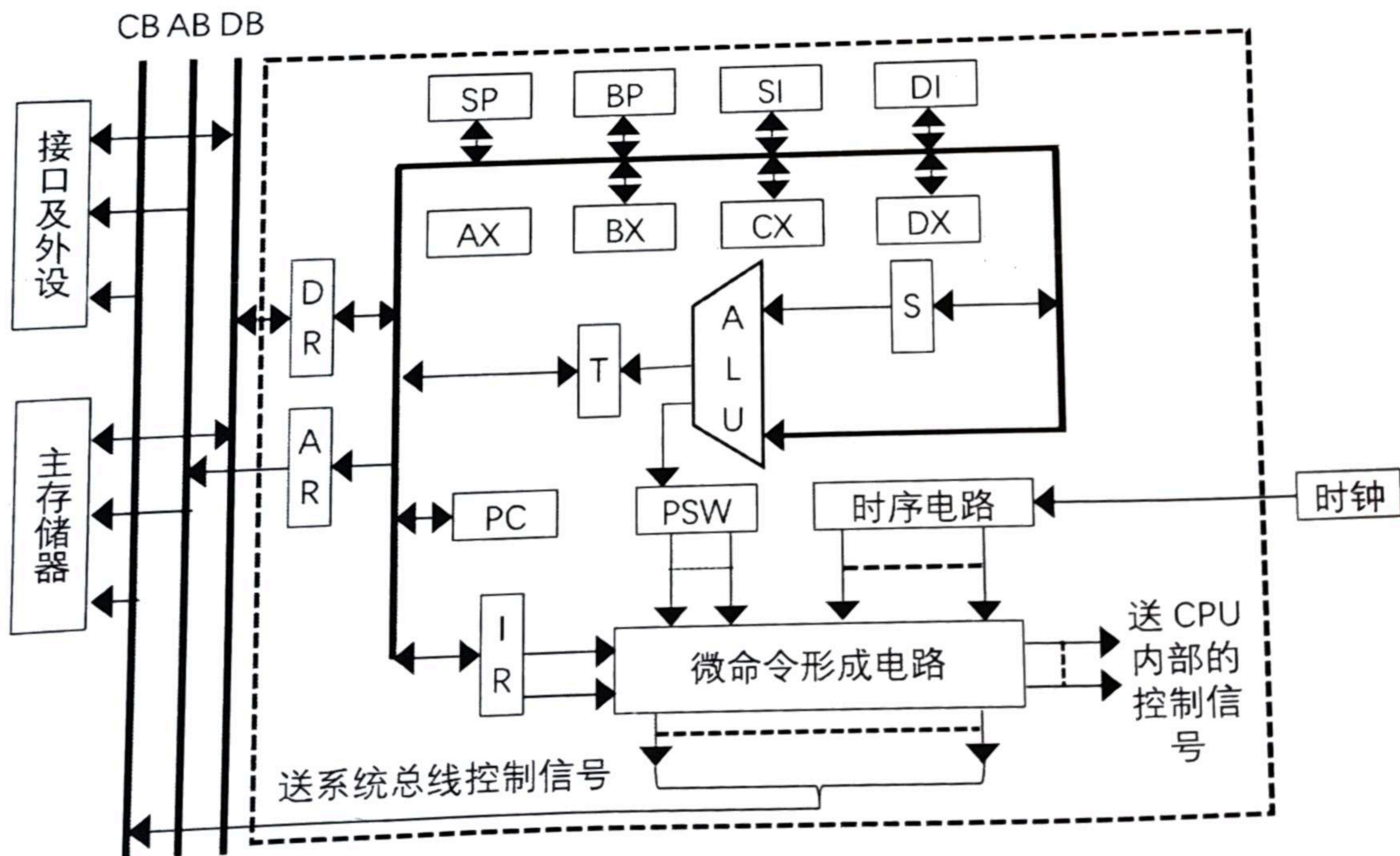
三、分析设计题

- 已知二进制数 $x = -0.1011$, $y = -0.1101$, 用补码布斯法计算 $[x * y]$ 补。
- 某计算机采用页式虚拟存储管理方式, 按字节编址, 虚拟地址为 32 位, 物理地址为 26 位, 页大小为 4KB; TLB 采用全相联映射: cache 数据区大小为 64KB, 按 2 路组相联方式组织, 主存块大小为 64B。下图为存储访问过程的示意图。请回答下列问题。



- (1) 图中 A~G 的位数各是多少？分别存放的是什么信息？
- (2) 将块号 4102 的主存块装入到 Cache 中时，所映射的 Cache 组号是多少？对应的 H 字段内容是什么？
- (3) Cache 缺失处理的时间开销大还是缺页处理的时间开销大？为什么？

3、



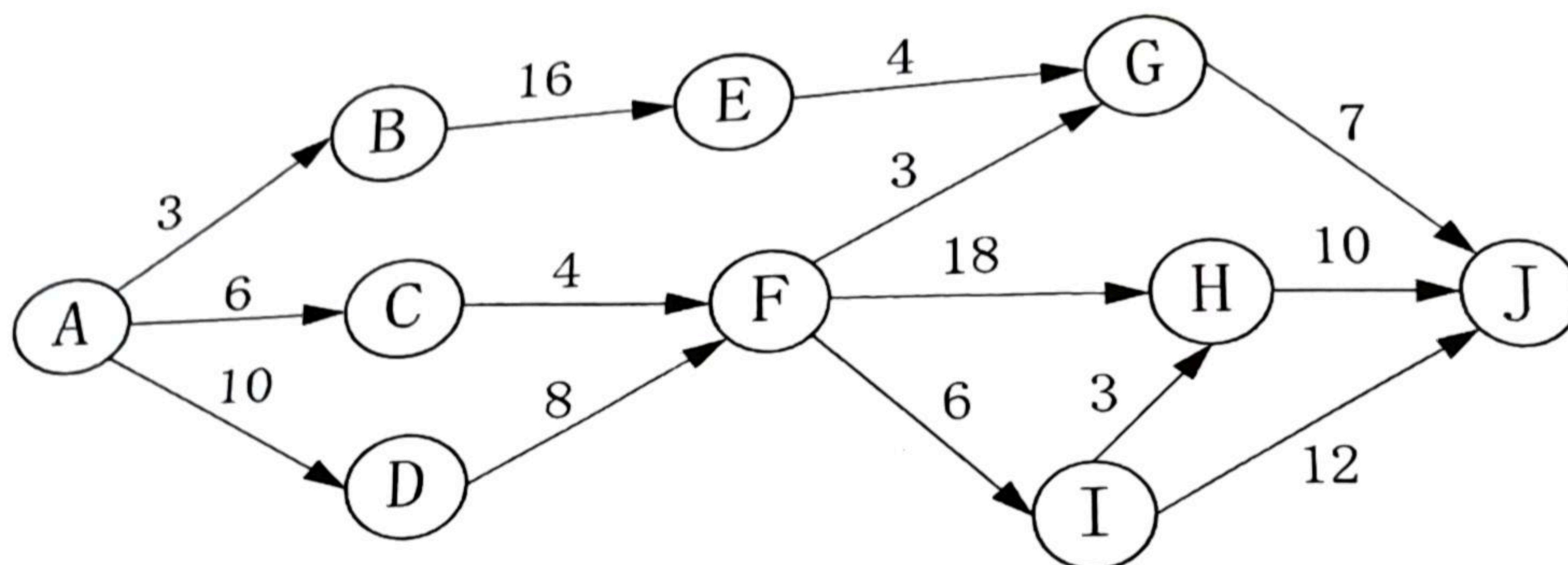
3、

取指令操作样例		PCout, Arin
T1	AR←PC	ARout, Mread, DRSin
T2	DR←Memory[AR]	PC+1, DRlout, IRin
T3	PC←PC+1, IR←AR	

- (1) ADD AX, (1000H) 加法指令, 源操作数间接寻址
 (2) JZ offset, offset 为偏移量, JZ 无条件跳转指令

一、综合题

1. (14 分) 如下图所示 AOE 网, 求出所有的关键活动并给出关键路径。



2. (16 分) 设计一个算法, 判断一个二叉树是否是二叉排序树。

二叉树类型定义如下:

```

typedef struct node
{
    int key;
    struct node *lchild, *rchild;
} bitree;
  
```

3. (9 分) 判别以下序列是否为小顶堆。如果不是, 则把它调整为小顶堆 (要求记录交换次数最少)

- (1) (100, 86, 48, 73, 35, 39, 42, 57, 66, 21);
 (2) (12, 14, 33, 65, 24, 56, 48, 92, 86, 33);
 (3) (05, 56, 20, 23, 40, 38, 29, 61, 35, 76, 28, 100)。

4. (16 分) 设计算法将两个带头结点的升序链表合并为一个新的降序链表并返回, 要求算法原地工作。

链表类型如:

```

typedef struct Node
{
    int data
    struct Node *next;
} LNode, Linklist;
  
```