

哈尔滨工业大学

2005 年硕士研究生入学考试试题参考答案

考试科目：计算机专业基础

报考专业：计算机科学与技术

考试科目代码：[424]

主观问答题，可根据考生表述明确与否酌情给分。

题号	一	二	三	四	五	六	七	八	九	十	总分
分数	10	8	8	22	27	30	5	20	8	12	150

I. 数据结构部分（共 75 分）

一、填空题（每小题 1 分，共 10 分）

1. 15 2. 关键路径 3. $(n+1)/2$ $n+1$ 4. 2^h-1 2^{h-1} 5. Huffman 6. 归并
7. $n-e$ 8. $m-n$

二、单选题（每空 1 分，共 8 分）

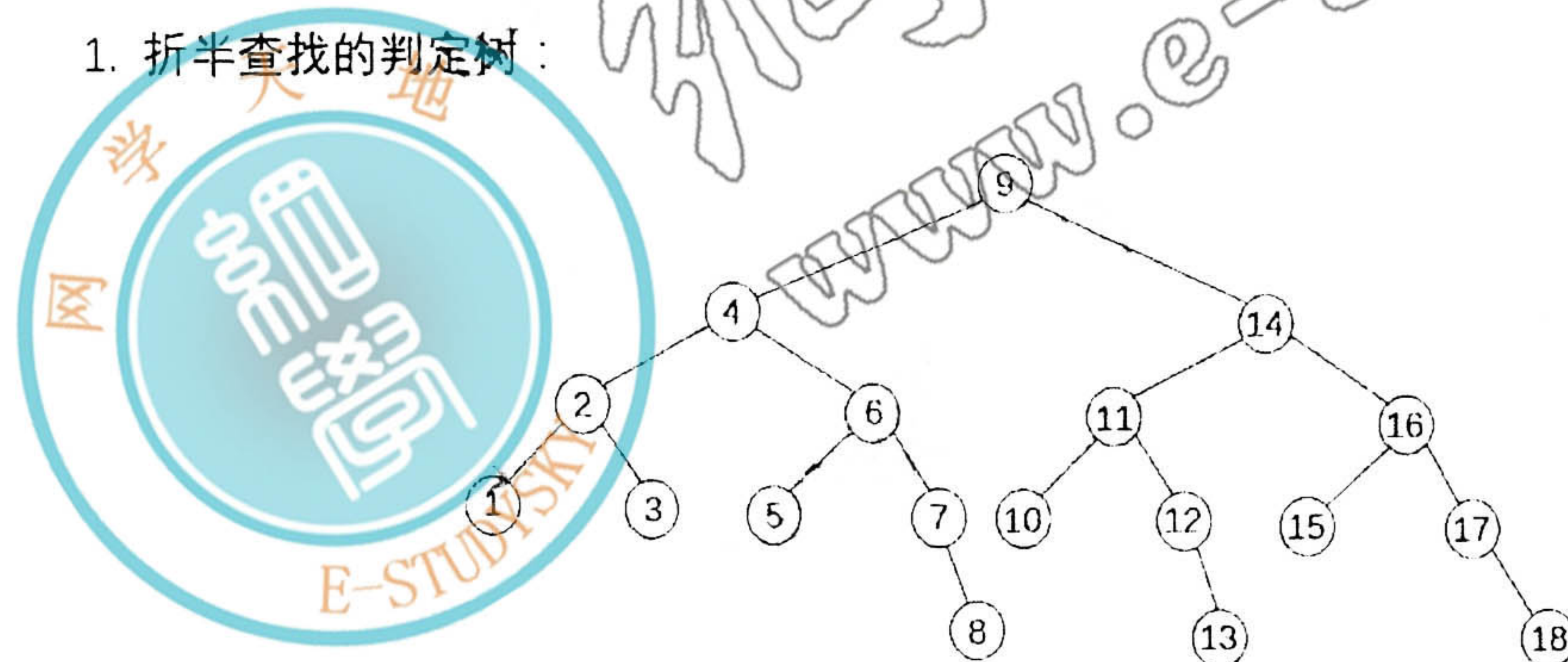
1. C 2. B 3. B 4. B 5. A 6. C 7. C 8. C

三、判断题（每题 1 分，共 8 分）

1. (✓) 2. (X) 3. (X) 4. (✓) 5. (X)
6. (✓) 7. (X) 8. (X)

四、简答题（共 22 分）

1. 折半查找的判定树：



$$ASL = (1 + 2 \times 2 + 3 \times 4 + 8 \times 4 + 5 \times 3) \times \frac{1}{18} \approx 3.6$$

查找失败的比较次数为 5。

2. (1) A、D

(2) 将(1)中的序列看成是一个字符序列，则从起始位置开始，依次到序列中的任何一个位置处，都可记录此时的字符"I"的个数 N_i 和 O 的个数 N_o ，如果在任一位置处均有 $N_i \geq N_o$ 且对于整个串有 $N_i = N_o$ ，则是合法的。

3. DFS: V1、V2、V4、V3 BFS: V1、V2、V3、V4

DFS 的基本思想：

设有一无向图 G ，其中所有顶点都标记为“未被访问过”。选择 G 中的一个顶点 V 作为搜索起点，将 V 标记为“已访问过”。然后递归地搜索和 V 相邻但没有被访问过的顶点。当访问完从顶点 V 所能到达的所有顶点之后，如果 G 中还有未被访问过的顶点，则再选一个没有被访问过的顶点作为新的搜索起点。重复上述过程，直到访问完 G 中的所有顶点为止。

五、算法设计题（共 27 分）

1. 算法思想：

设所要插入的堆为大顶堆，将要插入的 key 值放在堆数组的最后，然后与其父结点进行比较，若 key 大于父结点，则交换两者的位置，否则插入成功。

算法描述：

`void HeapInsert(R, key) // 堆中开始有 n 个元素`

```
{
    int i, j;
    ElemType temp;
    j = n + 1;
    i = j / 2;
    R[i] = key;
    while (j > 1) {
        if (R[j] > R[i]) {
            temp = R[i];
            R[i] = R[j];
            R[j] = temp;
            j = i;
            i = j / 2;
        }
        else
            return;
    }
}
```

该算法的时间复杂度为 $O(n \log_2 n)$ 。

2. 算法思想：

首先将所有记录的 $count$ 字段全部初始化为 -1，每次找到一个最小的记录，然后修改记录 $count$ 值，并输出此元素，直到所有元素输出。

算法描述：

```
void Sort(node R[], int n) {
    int Index = 0, num = 0;
    node min;
    for (int i = 0; i < n; i++)
        R[i].count = -1;
    min.count = -1;
    while (num < n) {
```



```
for (int i = 0; i < n; i++){
    if (R[i].count == -1 && min.count == -1){
        min = R[i];
        Index = i;
        min.count = 0;
    }
    if (R[i].key > min){
        min = R[i];
        Index = i;
    }
}
OutPut(R[Index]);
R[Index].count = ++num;
min.count = -1;
}
```

II. 计算机组成原理部分（共 75 分）

六、填空题（每空 1 分，共 30 分）

1. 总线控制器 总线判优控制 总线通信控制 信息传送控制 即通信双方如何
获知开始/结束以及双方如何协调配合 半同步通信
2. 指令周期 机器周期 时钟周期
3. 程序查询 程序中断 DMA
4. 4M 1M
5. 620μs 630μs
6. Cache 结构 快速进位 指令流水
7. 0400 F6FF
8. FF.FFFFFFFF $(1-2^{-23}) \times 2^{127}$ $-(2^{-1}+2^{-23}) \times 2^{-128}$
9. 组合逻辑 硬连线 微程序 存储
10. 14 $2^{32}=4G$

七、选择题（每题 1 分，共 5 分）

1. D 2. C 3. B 4. B 5. C

八、简答与计算题(共 20 分)

1. 不能得出 A 机比 B 机快的结论。因为 MIPS 值是对该机器指令系统的全部指令周期时间按算术平均或者加权平均方式求得，基准程序也是基于这种算法加以编制、进行调试的。但在执行具体程序时，其使用的指令集以及各种指令的使用频度会有很大不同，B 机完全有可能比 A 机提前完成程序。

2. DMA 接口电路中有以下 4 个寄存器：

主存地址寄存器 AR，用于存放主存中需要交换数据的地址，若主存容量为 $1M \times 16$ 位，由于 $1M=2^{20}$ ，故 AR 应为 20 位。

字计数器 WC，用于记录传送数据的总字数，若传送最大批量为 $512B=256$ 字，由于 $256=2^8$ ，故 WC 应为 8 位。

数据缓冲寄存器 BR，用于暂存每次传送的字数据，BR 应为 16 位。

设备地址寄存器 DAR，用于存放设备码或表示设备信息存储区的寻址信息。若外设地址空间为 $256=2^8$ ，则设备地址码长 8 位，即 DAR 应为 8 位。

3. 设采用原码形式机器数，加减交替除法

$$x=0.1010 \quad [X]_{\text{原}}=0.1010 \quad x^*=0.1010 \quad y=-0.1101 \quad [Y]_{\text{原}}=1.1101 \quad y^*=0.1101$$

$$[-y^*]_{\text{补}}=1.0011 \quad \text{商符 } x_0 \oplus y_0 = 0 \oplus 1 = 1$$

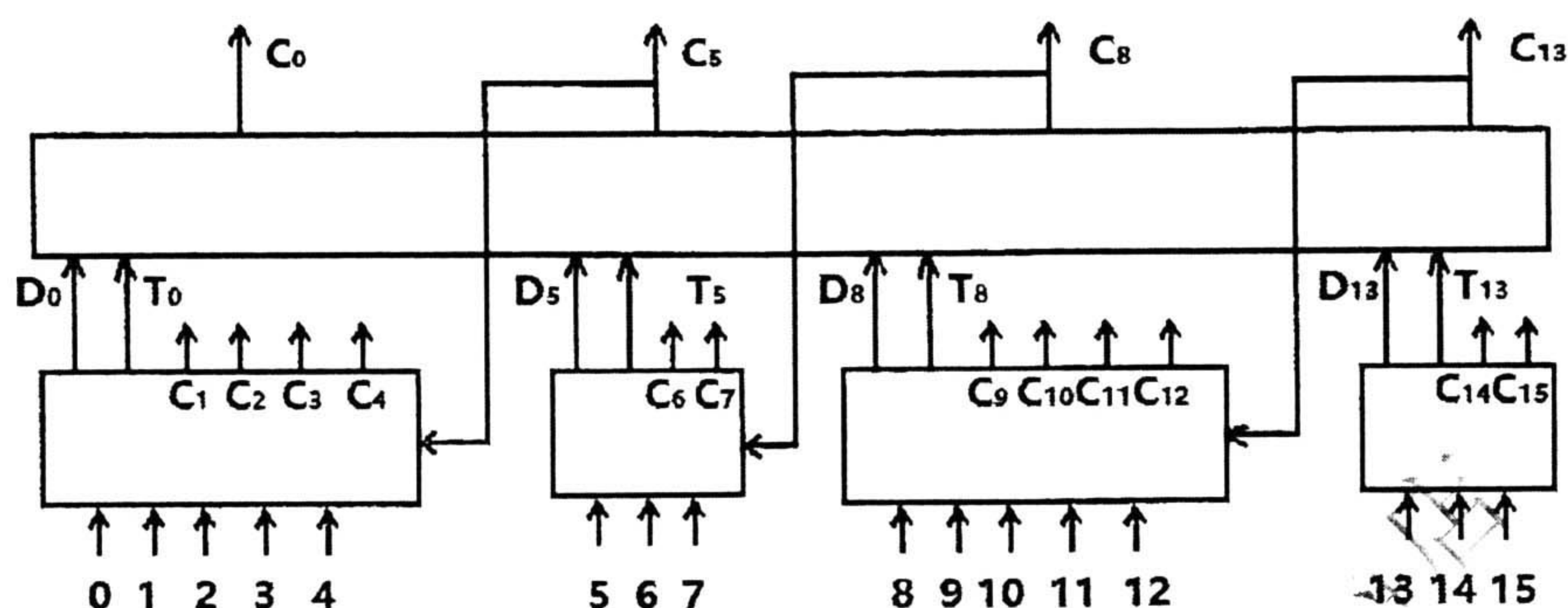
被除数(余数)	商	说明
0. 1 0 1 0	0 0 0 0 x	
+ 1. 0 0 1 1		$+[-y^*]_{\text{补}}$ ，减除数
1. 1 1 0 1	0	余数为负，商0
1. 1 0 1 0	0 x	$\leftarrow 1$ 位
+ 0. 1 1 0 1		$+ [y^*]_{\text{原}}$ ，减除数
0. 0 0 1 1	0 1	余数为正，商1
0. 1 1 1 0	0 1 x	$\leftarrow 1$ 位
+ 1. 0 0 1 1		$+ [-y^*]_{\text{补}}$ ，减除数
0. 0 0 0 1	0 1 1	余数为正，商1
0. 0 0 1 0	0 1 1 x	$\leftarrow 1$ 位
+ 1. 0 0 1 1		$+ [-y^*]_{\text{补}}$ ，减除数
1. 0 1 0 1	0 1 1 0	余数为负，商0
0. 1 0 1 0	0 1 1 0 x	$\leftarrow 1$ 位
+ 0. 1 1 0 1		$+ [y^*]_{\text{原}}$ ，减除数
1. 0 1 1 1	0 1 1 0 0	余数为负，商0
+ 0. 1 1 0 1		$+ [y^*]_{\text{原}}$ ，加除数
0. 0 1 0 0		恢复余数

商：1.1100 余：0.00000100

$$x/y = -0.1100 + (0.00000100)/(-0.1101)$$

4. CPU 进入中断响应周期要完成保护程序断点、寻找服务程序入口地址、关中断等操作。这些操作由中断隐指令完成，即由硬件自动完成。

5. 双重分组跳跃进位链是将 n 位全加器分成几个大组，大组内有包含若干小组，每个大组内所包含的各个小组的最高进位是同时产生的，大组与大组之间采用串行进位。因为各个小组的最高进位是同时产生的，小组内的其他进位也是同时产生的(小组内的其他信号与小组的最高位进位并不是同时产生的)，故这种进位链又有“组(小组)内并行，组(小组)间并行之称”，并且其速度较快。



$$C_{13} = D_{13} + T_{13}C_{16}$$

$$C_8 = D_8 + T_8C_{13} = D_8 + T_8D_{13} + T_8T_{13}C_{16}$$

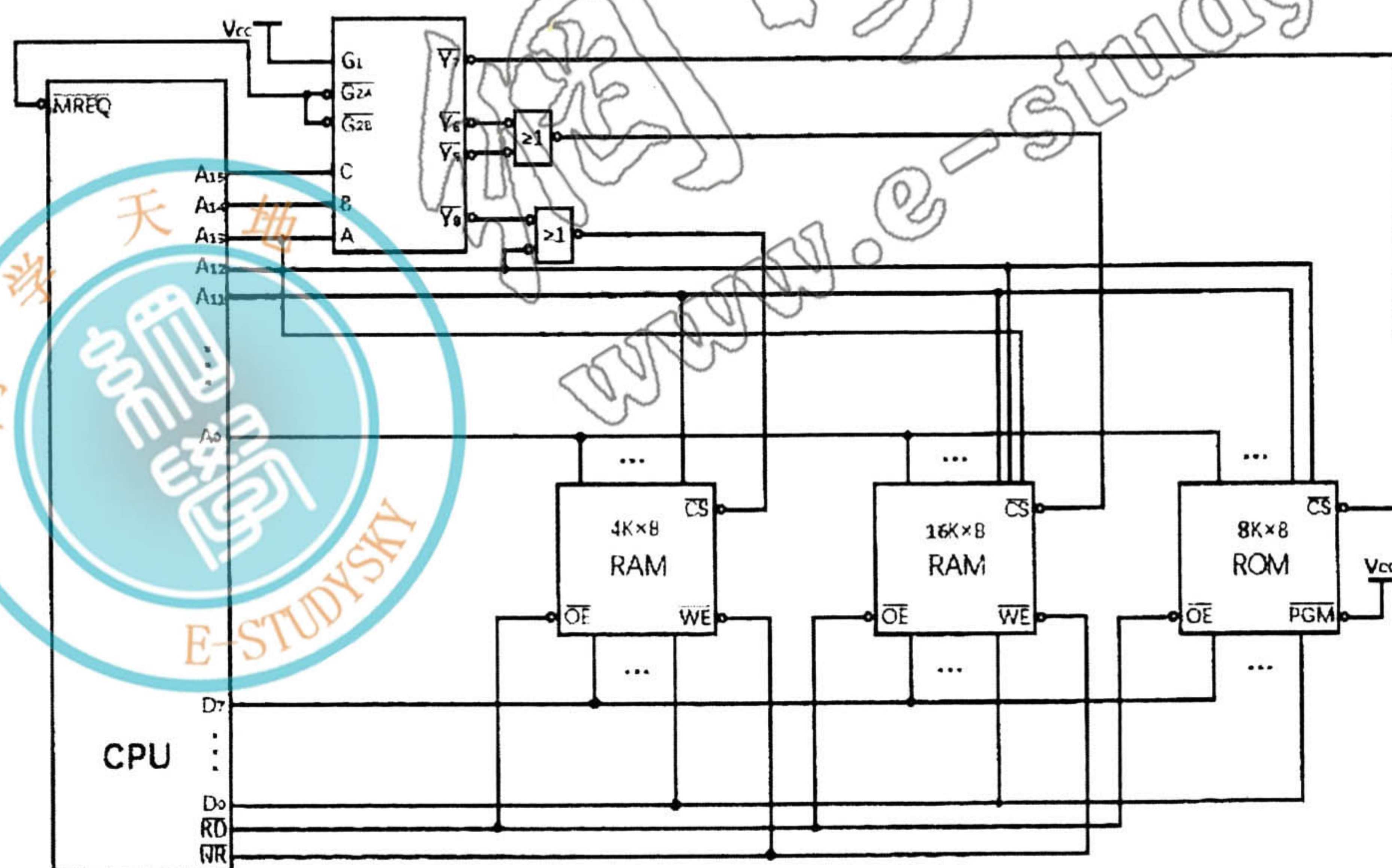
$$C_5 = D_5 + T_5C_8 = D_5 + T_5D_8 + T_5T_8D_{13} + T_5T_8T_{13}C_{16}$$

$$C_0 = D_0 + T_0C_5 = D_0 + T_0D_5 + T_0T_5D_8 + T_0T_5T_8D_{13} + T_0T_5T_8T_{13}C_{16}$$

九、

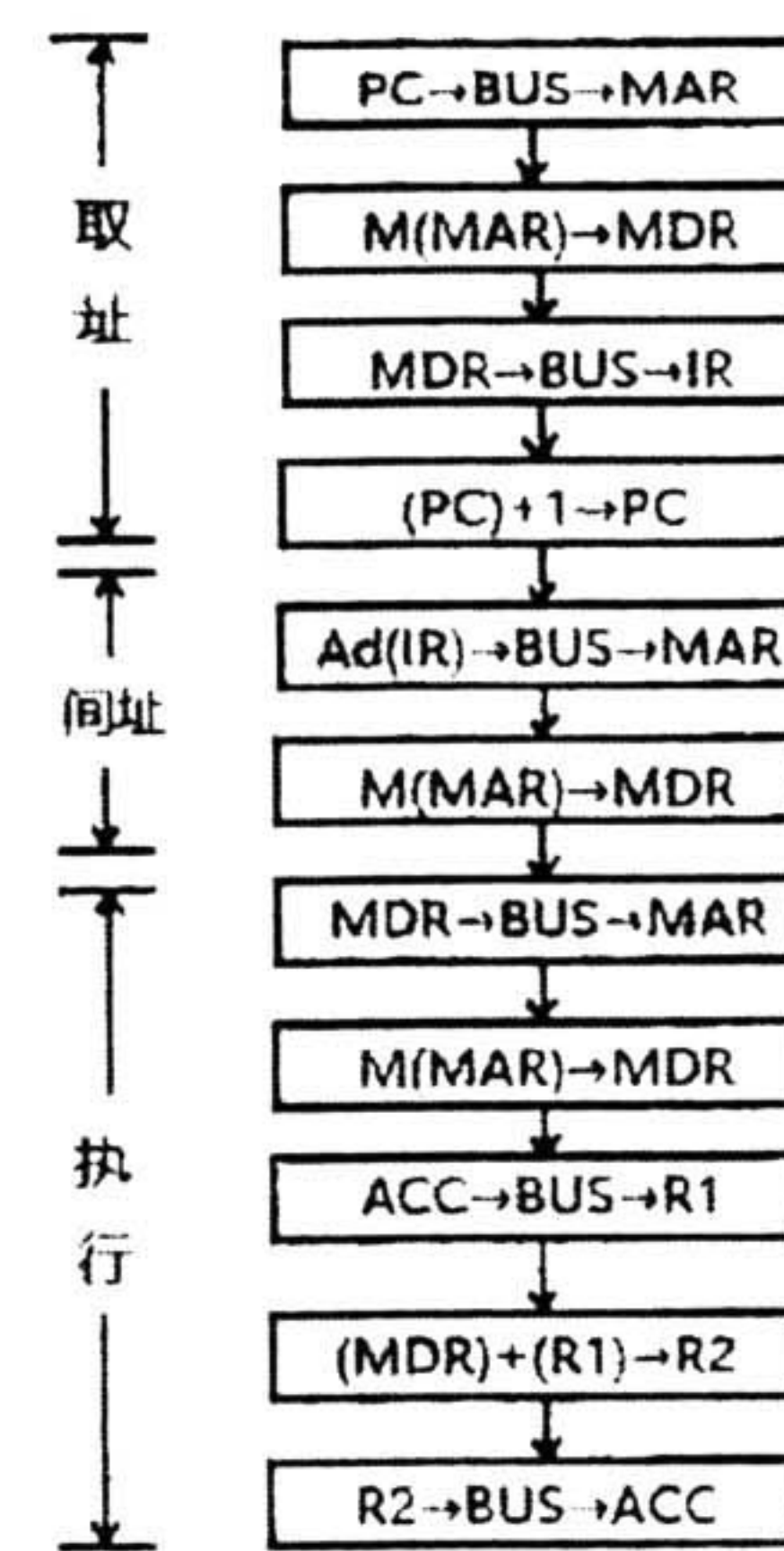
根据主存地址空间分配，最小 4K 地址空间为系统程序工作区，选用 1 片 4K×8 位 RAM；
最大 8K 为系统程序区，选用 1 片 8K×8 位 ROM；与系统相邻的 16K 为用户程序区，选用 1 片 16K×8 位 RAM 芯片。各芯片地址如下：

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4K×8位RAM
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	16K×8位RAM
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	8K×8位ROM
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	



十、综合题 (12 分)

1.



2. 取址阶段：

T_0 $PC \rightarrow MAR$ $1 \rightarrow R$
 T_1 $M(MAR) \rightarrow MDR$ $(PC) + 1 \rightarrow PC$
 T_2 $MDR \rightarrow IR$ $OP(IR) \rightarrow ID$

间址阶段：

T_0 $Ad(IR) \rightarrow MAR$ $1 \rightarrow R$
 T_1 $M(MAR) \rightarrow MDR$

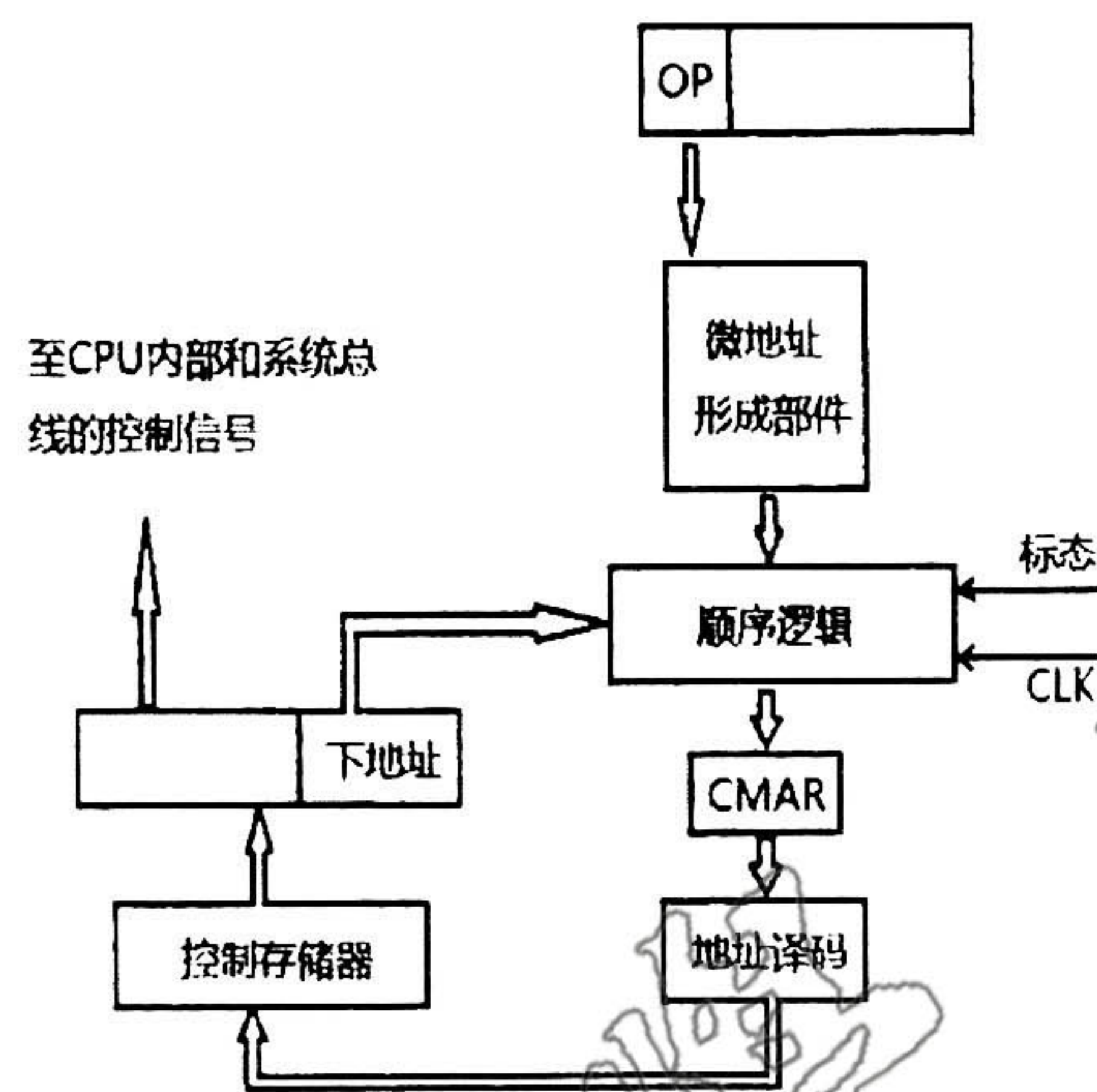
执行阶段：

T_0 $MDR \rightarrow MAR$ $1 \rightarrow R$
 T_1 $M(MAR) \rightarrow MDR$ $ACC \rightarrow R1$
 T_2 $(MDR) + (R1) \rightarrow R2$
 T_3 $R2 \rightarrow ACC$

4. CU 采用微程序设计，需增加给出下条微指令地址的微操作命令，即：

$Ad(CMDR) \rightarrow CMAR$

$OP(IR) \rightarrow \text{微地址形成部件} \rightarrow CMAR$



由图可知：若 CU 采用微程序控制设计，则 CU 中应包含控制存储器、CMAR、CMDR、地址译码器、微地址形成部件、顺序逻辑、时钟系统提供 CLK、提供标态的程序状态字寄存器 PSW、程序计数器 PC、指令寄存器 IR 等。

