

哈尔滨工业大学

2008 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	总分
分数	20	9	16	30	15	32	9	9	10	150

I. 数据结构部分（共 75 分）

一、填空题：（每小题 2 分，共 20 分）

1. $n > 20$ 2. $27/7$ 3. $AB \& \& EF > ! ||$ 4. $(R+N-F)\%N$ 5. 18,10,5,73,68,27,25,41,32,99
 6. $m - 2m - 1$ 7. 不同关键字具有相同的散列地址的现象 在处理冲突的过程中
 出现非同义词之间对一个散列地址争夺的现象 8. 4 9. 克鲁斯卡尔(Kruskal)
 10. 12,17,35,39,21,43

二、判断题：（每空 1 分，共 9 分）

1. (X) 2. (✓) 3. (X) 4. (✓) 5. (X)
 6. (✓) 7. (X) 8. (X) 9. (X)

三、问答题（每题 8 分，共 16 分）

1. 快速排序时间代价、空间代价较低，但是不稳定，而归并排序是稳定的。由于大多数情况下排序是按记录的主关键字进行的，则所用的排序方法是否稳定无关紧要，而由于计算机内存及运算速度的限制，对时间代价和空间代价的要求较高，故快速排序通常情况下被广泛采用而非归并排序。

$$2. (1) ASL_{succ} = (1 \times \frac{1}{2} + 2 \times \frac{1}{2^2} + 3 \times \frac{1}{2^3} + \dots + n \times \frac{1}{2^n}) \quad \text{①式}$$

$$\frac{1}{2} ASL_{succ} = (0 + \frac{1}{2^2} + 2 \times \frac{1}{2^3} + \dots + (n-1) \times \frac{1}{2^n} + n \times \frac{1}{2^{n+1}}) \quad \text{②式}$$

$$\text{①} - \text{②} \rightarrow \frac{1}{2} ASL_{succ} = (\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \dots + \frac{1}{2^n} - n \times \frac{1}{2^{n+1}})$$

$$ASL_{succ} = (1 - \frac{1}{2^n} - \frac{n}{2^{n+1}}) \times 2$$

$$(2) ASL_{unsucc} = (n+1) \times \frac{1}{2} + (n+1) \times \frac{1}{2^2} + \dots + (n+1) \times \frac{1}{2^n}$$

$$= (n+1)(1 - \frac{1}{2^n})$$

四、算法设计题（每题 15 分，共 30 分）

1. 采用层次遍历思想，设置变量 level 记录当前结点所在层数，设置变量 last 指向当前层最后结点，每次层次遍历出队时与 last 比较，若两者相等，那么层数加 1，并让 last 指向下一层最右结

点，当遍历到此左右孩子均为空且near初值为0时，给near赋值。当遍历完树给far值level

```
typedef struct {
    ElementType data;
    BiNode * lchild, * rchild;
}BiNode;
void FarNear(BiTree T)
{
    int far = 0, near = 0;
    if (T){
        int front = -1, rear = -1;
        int last = 0, level = 0; //last指向下一层第一个结点的位置
        BiTree Q[Maxsize];
        Q[rear++] = T;
        BiTree p;
        while (front < rear){
            p = Q[++front];
            if (p->lchild)
                Q[++rear] = p->lchild;
            if (p->rchild)
                Q[++rear] = p->rchild;
            if (front == last){
                level++;
                last = rear;
            }
            if (p->lchild == NULL && p->rchild == NULL && near == 0)
                near = level;
        }
        far = level;
    }
}
```

2. 算法思想：设定边权全为1，求关键路径，即可包含边最多。

语言描述：

```
Status TopologicalOrder(ALGraph G, Stack &T){
    FindInDegree(G, indegree);
    InitStack(T); count = 0; ve[0...G.vexnum - 1] = 0;
    while (!StackEmpty(S)){
        Pop(S, j); Push(T, j); ++count;
        for (p = G.vertices[j].firstarc; p; p = p->nextarc){
            k = p->adjvex;
            if (--indegree[k] == 0) Push(S, k);
            if (ve[j] + *(p->info) > ve[k]) ve[k] = ve[j] + *(p->info);
        }
    }
    if (count < G.vexnum)
```



```
    return ERROR;
else
    return OK;
}

Stack CriticalPath(ALGraph G){
    if (!TopologicalOrder(G, T))
        return ERROR;
    vl[0...G.vexnum - 1] = ve[G.vexnum - 1];
    while (!StackEmpty(T))
        for (Pop(T, j), p = G.vertices[j].firstarc; p; p = p->nextarc){
            k = p->adjvex;
            dut = *(p->info);
            if (v[k] - dut < vl[j])
                vl[j] = v[k] - dut;
        }
    for (j = 0; j < G.vexnum; ++j)
        for (p = G.vertices[j].firstarc; p; p = p->nextarc){
            k = p->adjvex; dut = *(p->info);
            ee = ve[j];
            el = vl[k] - dut;
            tag = (ee == el) ? '*' : ' ';
            printf(j, k, dut, ee, el, tag);
        }
}
```

II. 计算机组成原理部分 (共 75 分)

五、填空题 (每空 1 分, 共 15 分)

1. 判优控制和通信控制 链式查询 计数器定时查询 独立请求
2. 78 56 34 12H
3. 时分复用
4. 9
5. 指令执行周期末 指令存取周期结束
6. $2^{-127} \times 2^{-23}$ $-2^{-127}(1-2^{-23})$ $2^{-127} \times (1-2^{-23})$ $2^{-128} \times (2^{-1} + 2^{-23})$
7. 加减交替法 符号位

六、简答题 (每小题 8, 共 32 分)

1. DMA 方式的主要特点：I/O 和 CPU 并行工作；主存与 I/O 接口间有一条直接数据通路；不中断现行程序，无需保护现场、恢复现场；当 DMA 请求占用总线控制权时，若采用周期挪用的方式，CPU 暂停一个存取周期访问主存，但可继续自身内部的操作，即传送和主程序是并行的。

以数据输入为例，具体操作如下：

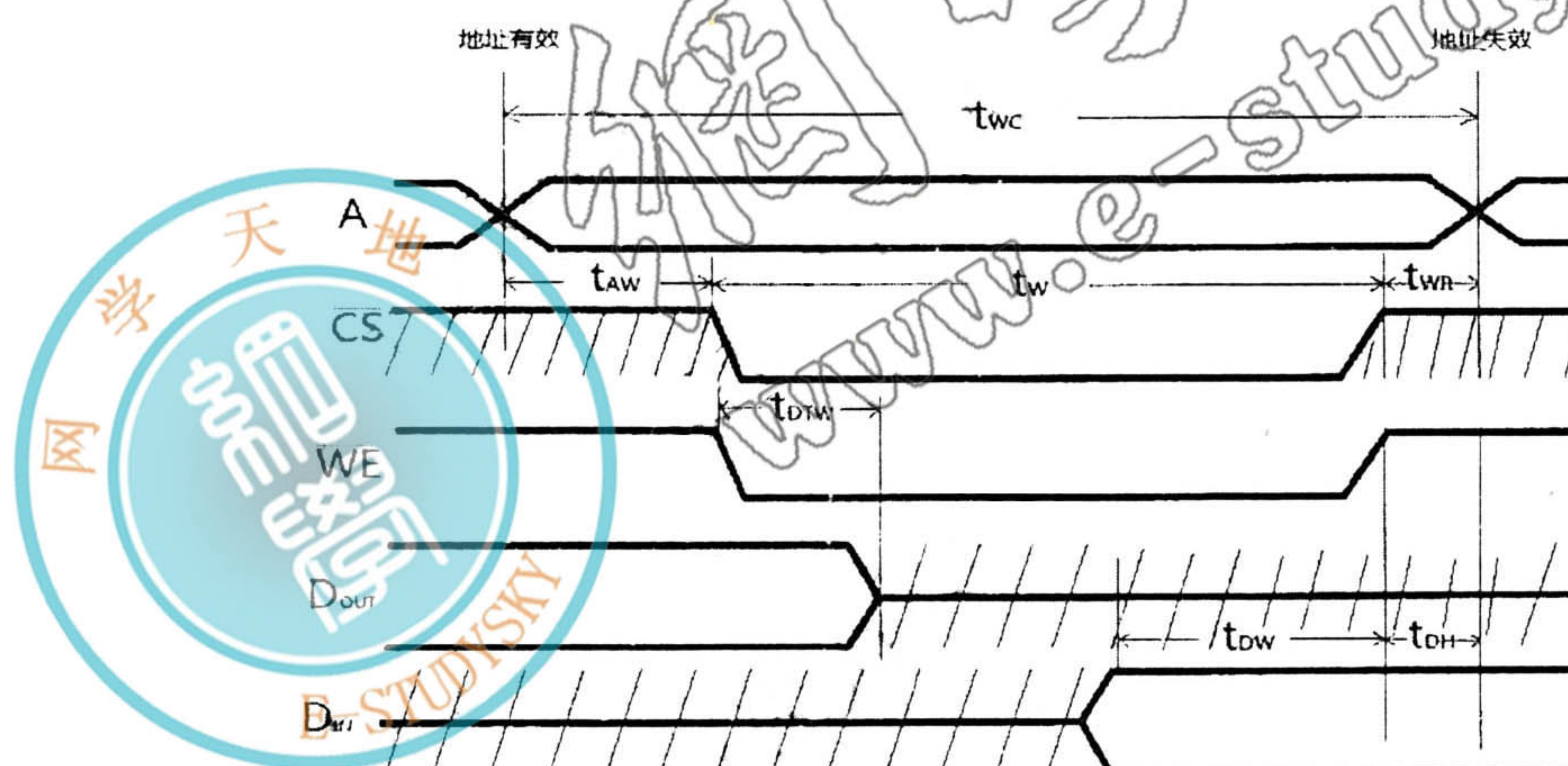
- ① 从设备读取一个字到 MDA 的数据缓冲寄存器 BR 中，表示数据缓冲寄存器“满”（如果 I/O 设备是面向字符的，则一次读入一个字节，组装成一个字）；
- ② 设备向 DMA 接口发请求（DREQ）
- ③ DMA 接口向 CPU 申请总线控制权（HRQ）
- ④ CPU 发回 HLDA 信号表示允许将总线控制权交给 DMA 接口
- ⑤ 将 DMA 主存地址寄存器中的主存地址送地址总线
- ⑥ 通知设备已被授予一个 DMA 周期（DACK），并为交换下一个字做准备；
- ⑦ 将 DMA 数据缓冲寄存器的内容送数据总线
- ⑧ 命令存储器作写操作
- ⑨ 修改主存地址和字计数值
- ⑩ 判断数据块是否传送结束，若未结束，则继续传送；若已结束（字计数器溢出），则向 CPU 申请程序中断，标志数据块传送结束。

2. ①地址信号给出过晚

②因为是写入操作应给出 \overline{WE} 信号而非 R/\overline{W} 信号，否则会造成误写

③写操作必须给出 D_{IN} 和 D_{OUT} 两个信号，指出数据来源及数据去路

正确时序图如下：



3. 单重分组跳跃进位链是组内并行、组件串行的进位链。多重分组跳跃进位链是组内并行，组件并行的进位链。

大组中产生的进位： C_0 、 C_3 、 C_7 、 C_{10} 。

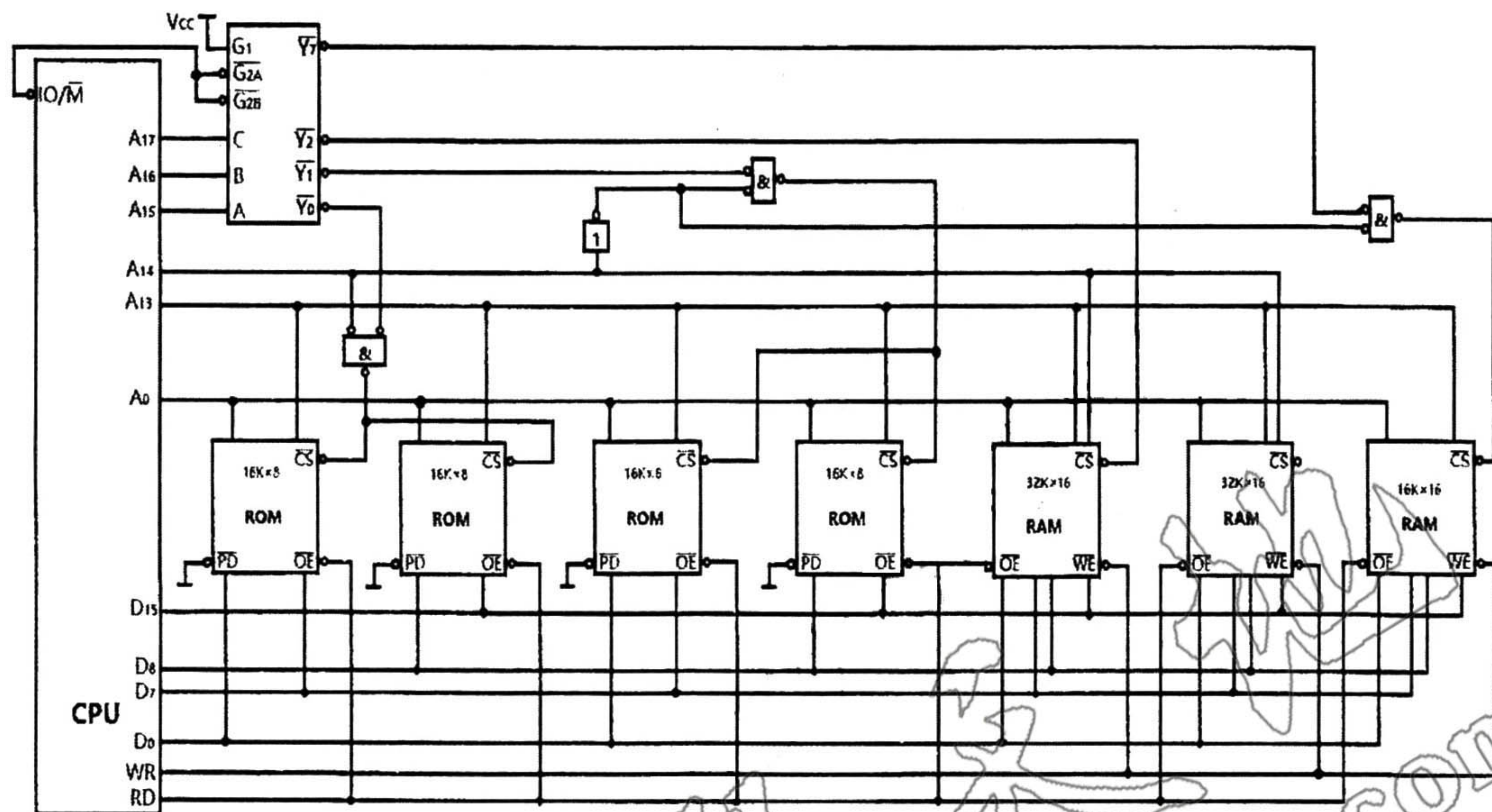
两种方案产生全部进位的时间是一致的。3、5、3、5分组的进位时间： $2.5t_y \times 3 = 7.5t_y$ 。

4、4、4、4分组的进位时间： $2.5t_y \times 3 = 7.5t_y$ 。可见两种分组方案最长加法时间相同。双重分组并行进位链的最长进位时间只与组数和级数有关，而与组内位数无关。

4.

[illegible]

(2)CPU 与存储器的连接图为：



九、控制单元设计题(共 10 分)

1. 机器周期和节拍(状态)组成了多级时序系统。

2. 取址周期：

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

执行周期：

T_0 Ad(IR) \rightarrow MAR $1 \rightarrow R$
 T_1 M(MAR) \rightarrow MDR $0 \rightarrow Q_{n+1}$ $0 \rightarrow A$
 T_2 MDR $\rightarrow Q_{0-n}$
 T_0^* $\bar{Q}_n Q_{n+1}(A+X) + Q_n \bar{Q}_{n+1}(A+\bar{X}+1) + \bar{Q}_n \bar{Q}_{n+1}A + Q_n Q_{n+1}A \rightarrow A$
 T_1^* L(A//Q) \rightarrow R(A//Q)

3. 中央控制节拍包括取值阶段所有节拍和执行阶段 T_0 、 T_1 、 T_2 3 个节拍，完成取指令和取操作数及乘法运算前的准备工作。

局部控制节拍是执行阶段的 T_0^* 、 T_1^* 节拍，其中 T_0^* 为重复加操作，受 Q 寄存器的末尾两位 Q_n 和 Q_{n+1} 控制，最多执行 $n+1$ 此； T_1^* 多为移位操作共执行 n 次。