

2014年

合工大850专业课考试真题解析

@zengtao

2020-11

Version: 1.0.1

Update date: 2020-11-27

Part 1:

Data Structure

选择题1、对于顺序存储的线性表，访问结点和增加节点时间复杂度（），删除结点的时间复杂度为（）

A. $O(n)$ $O(n)$

B. $O(n)$ $O(1)$

C. $O(1)$ $O(n)$

D. $O(1)$ $O(1)$

C

选择题2、一棵左右子树均不空的二叉树在先序线索化后,其中空的链域的个数是()

A. 0

B. 1

C. 2

D. 不确定

B

选择题3、在图采用邻接表存储时，求最小生成树的Prime算法的时间复杂度为()

A. $O(n)$

B. $O(n+e)$

C. $O(n^2)$

D. $O(n^3)$

C

选择题4、 n 个顶点无向连通图最少有()条边

- A. $O(n^2)$
- B. $O(n)$
- C. $O(n-1)$
- D. $O(\log n)$

C

选择题5、 对序列{15, 9, 7, 8, 20, -1, 4}进行排序, 进行一趟后数据的排列变为{4, 9, -1, 8, 20, 7, 15} 则采用的是 () 排序

A、选择

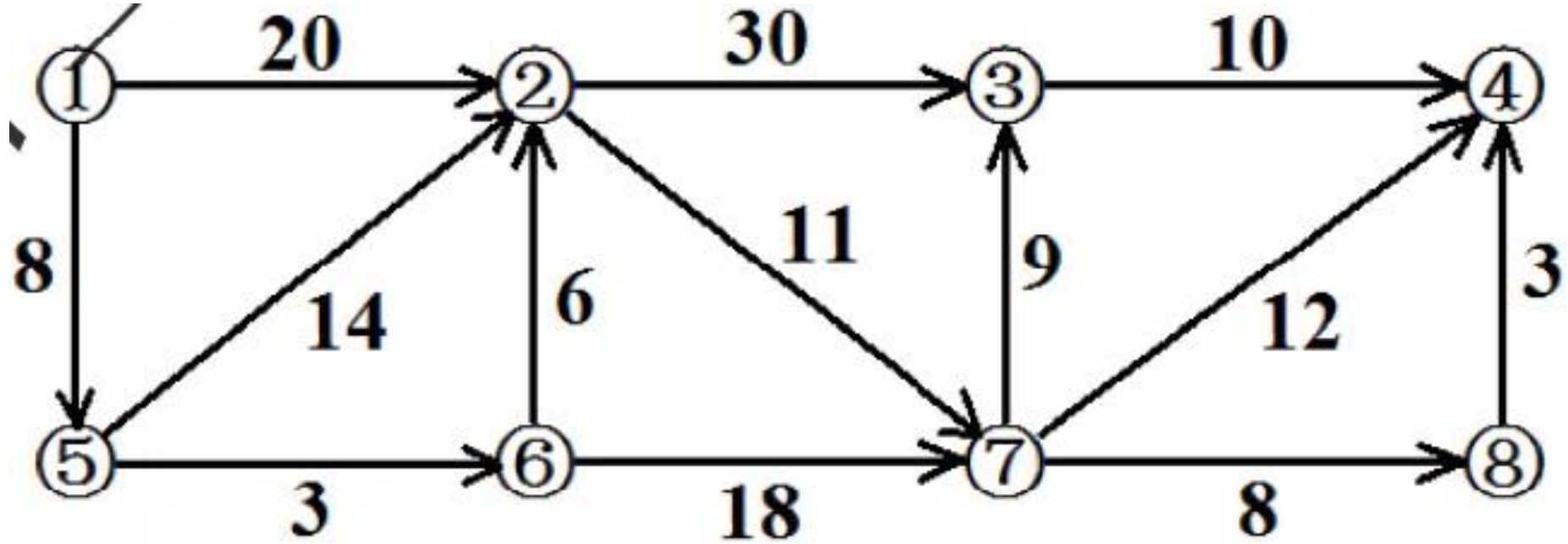
B、快速

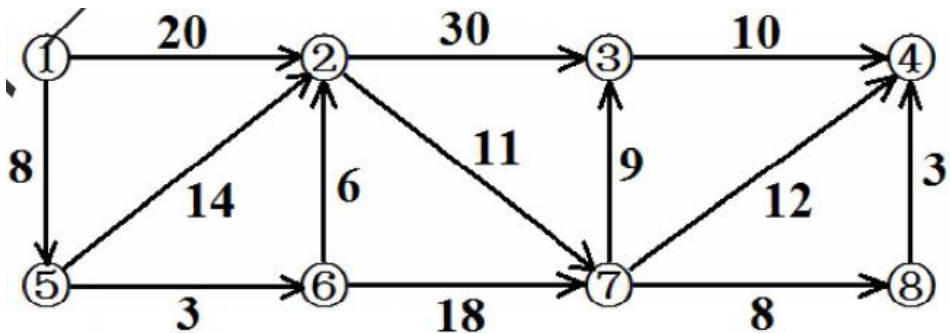
C、希尔

D、冒泡

C

应用题1、求出下图中从顶点1到其余各顶点的最短路径





初始化，顶点1到所有顶底的距离为

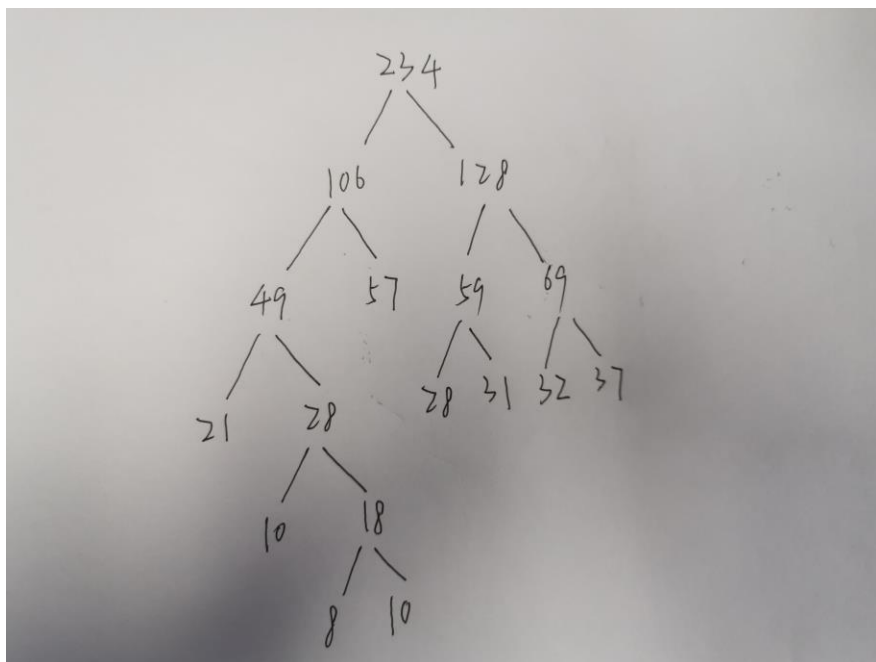
顶点	1	2	3	4	5	6	7	8
距离	0	20	∞	∞	8	∞	∞	∞

接下来，进行 $n - 1$ 趟选择，每次选择距离最小对应的顶点，并根据该顶点来更新到其他顶点的距离

顶点	1	2	3	4	5	6	7	8
距离	0	17	37	39	8	11	28	36

应用题2、对下面数据表，构建Huffman树，并计算其带权路径长度

(8, 10, 21, 32, 57, 31, 28, 37, 10)



方法一: $WPL = 57 \times 2 + (21 + 28 + 31 + 32 + 37) \times 3 + 10 \times 4 + (8 + 10) \times 5 = 691$

方法二: $WPL = 234 + 106 + 128 + 49 + 59 + 69 + 28 + 18 = 691$

应用题3、给出下列程序，写出print(5) 的运行结果

```
void print(int n) {  
    if (n == 0)  
        return;  
    else {  
        print(n - 1);  
        for (int i = 0; i < n; ++i)  
            cout << n << " ";  
        cout << endl;  
    }  
}
```

运行结果:

1

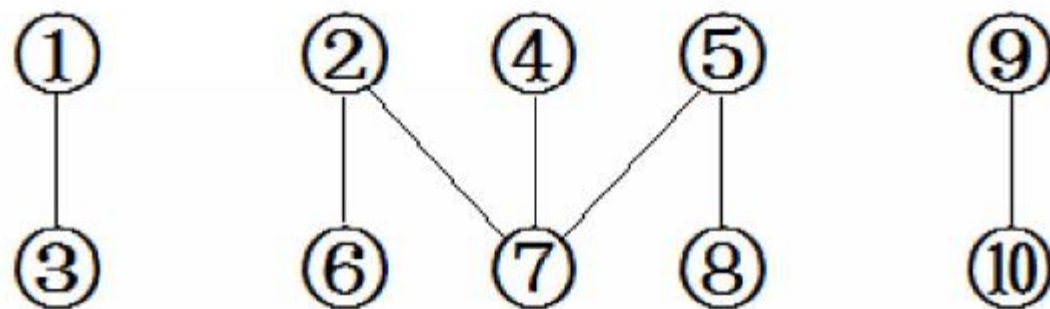
2 2

3 3 3

4 4 4 4

5 5 5 5 5

编程题1、设计算法以实现对无向图G的深度遍历，要求：将每一个连通分量中的顶点以一个表的形式输出。例如，下图的输出结果为：(1, 3) (2, 6, 7, 4, 5, 8) (9, 10)



```

void DFS(graph g, int v, bool visited[], bool *isFirstVertex) {
    visited[v] = true;
    if (*isFirstVertex) {
        printf("%c", getVertexValue(g, v));
        *isFirstVertex = false;
    } else {
        printf(",%c", getVertexValue(g, v));
    }

    for (int i = firstNeighbor(g, v); i >= 0; i = nextNeighbor(g, v, i)) {
        if (!visited[i]) {
            DFS(g, i, visited, isFirstVertex);
        }
    }
}

```

稍微修改后的深度优先遍历算法

```

void componentTraversal(graph g, bool visited[], bool *isFirstVertex) {
    int i;
    for (i = 0; i < g->numVertex; i++) {
        if (!visited[i]) {
            *isFirstVertex = true;
            putchar('(');
            DFS(g, i, visited, isFirstVertex);
            putchar(')');
            putchar('\n');
        }
    }
}

```

一次深度优先遍历对应一个连通分量

Reference code: https://github.com/erqitao/data-structure-code-answer/blob/master/exam_2014/adjGraph.c

编程题2、设计一个**非递归**算法输出二叉树先序序列中最后一个结点的值

编程题2、设计一个**非递归**算法输出二叉树先序序列中最后一个结点的值

```
binTree preOrderLast(binTree root) {  
    while (root) {  
        while (root->rchild != NULL) {  
            root = root->rchild;  
        }  
  
        if (root->lchild != NULL) {  
            root = root->lchild;  
        } else {  
            break;  
        }  
    }  
    return root;  
}
```

一直往二叉树的最右边走

直到遇到叶子节点

Reference code: https://github.com/erqitao/data-structure-code-answer/blob/master/exam_2014/binTree.c

编程题3、给定顺序表，写出快速排序算法实现过程，并给出其最好时间性能分析


```

int partition(int ary[], int low, int high) {
    int pi, pj, pivot;
    pi = low;
    pj = high;
    pivot = ary[low];

    while (pi < pj) {
        while (pi < pj && ary[pj] >= pivot)
            pj--;

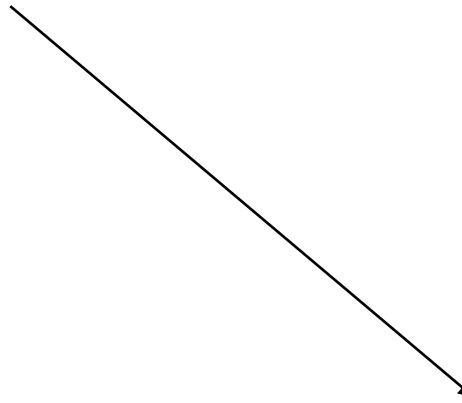
        if (pi < pj) {
            ary[pi] = ary[pj];
            pi++;
        }

        while (pi < pj && ary[pi] <= pivot) {
            pi++;
        }
        if (pi < pj) {
            ary[pj] = ary[pi];
            pj--;
        }
    }
    ary[pi] = pivot;

    return pi;
}

```

快速排序时间复杂度: $O(n \log n)$



```

void quickSort(int ary[], int low, int high) {
    if (low >= high)
        return;
    int mid = partition(ary, low, high);
    quickSort(ary, low, mid - 1);
    quickSort(ary, mid + 1, high);
}


```


Reference code: https://github.com/erqitao/data-structure-code-answer/blob/master/exam_2014/quickSort.c

See






https://github.com/erqitao/data-structure-code-answer/tree/master/exam_2014

for more details.

 master ▾ [data-structure-code-answer](#) / [exam_2014](#) /

 **erqitao** create a new repository

..

 a.exe	create a new repository
 adjGraph.c	create a new repository
 binTree.c	create a new repository
 program_questions_2014.png	create a new repository
 quickSort.c	create a new repository

Part 2:

Principles of Computer Organization

选择题1、下列选项中，哪一个不是描述计算机运算速度的指标

A. MIPS

B. CPI

C. MAR

D. PFLOPS

C

选择题2、11/64的二进制表示是

A. 0.001011 B. 0.010110 C. 0.101101 D. 0.011001

A

选择题3、在基数为4的浮点数中，用二进制表示的尾数左移4位，为了保持数值不变，阶码要

A. 减 1 B. 减 2 C. 减 3 D. 减 4

B

选择题4、偶校验法能发现几个二进制位出错

A. 偶数 B. 2 C. 奇数 D. 任意

C

选择题5、微程序控制器中，机器指令与微指令的关系是

- A. 每一条机器指令由一条微指令来执行
- B. 每一条机器指令由若干条微指令来执行
- C. 每一段机器指令组成的程序可由一条微指令来执行
- D. 一条微指令由若干条机器指令组成

B

选择题6、下列说法中不正确的是

- A. 同步传输中各部件用一个公共的时钟
- B. 异步传输中各部件不用时钟信号
- C. 同步通信中各部件的速度应差别不大
- D. 异步通信中各部件的速度可以有较大差别

B

填空题1、计算机硬件能直接识别并执行的是 ? 语言

机器语言

填空题2、变址寄存器内容由 ? 确定，在程序执行过程中 ? (可变/不变)

用户


可变

填空题3、在整数定点机中，若寄存器的内容为A0H，当它代表原码时，对应的十进制数为 ？

当它代表补码时，所对应的十进制数为 ？



-96



-32

填空题4、有一个4体低位交叉编址的存储器，假设单个存储体的存取周期为20ns，访问一次单个存储体可以存取一个存储字。现CPU每隔1/4存取周期启动一个存储体，则依次访问100个连续的存储字需 ? ns

$$20 + 99 \times 5 = 20 + 495 = 515$$

填空题5、某机有五级中断，优先级从高到低为L1 → L2 → L3 → L4 → L5。若将屏蔽字修改，改后L1级中断的屏蔽字为10011，L2级中断的屏蔽字为11111，L3级中断的屏蔽字为10111，L4级中断的屏蔽字为00010，L5中断的屏蔽字为00011，则修改后的处理优先级顺序从高到低为

L2 L3 L1 L5 L4

应用题1:

假定某机主存空间大小2GB，按字节编址。Cache的容量128KB，块大小为16个字，每个字32位，采用四路组相联映射方式。

- (1) 画出主存地址字段中各段的位数，指明其含义；
- (2) 设Cache初态为空，依次读主存第0, 1, 2, ..., 99号单元读出100个字（主存一次读出一个字，并重复按此次序读10次，命中率为多少？
- (3) 若主存的有取周期是Cache的6倍，则采用Cache后存储器的速度提高了多少？

应用题1:

假定某机主存空间大小2GB，按字节编址。Cache的容量128KB，块大小为16个字，每个字32位，采用四路组相联映射方式。

(1) 画出主存地址字段中各段的位数，指明其含义；

解:

主存空间大小 $2\text{GB} = 2^{31} \text{ B} \rightarrow$ 主存地址线个数 31位

$$\text{Cache块个数} = \frac{128\text{KB}}{16 \times 4\text{B}} = 2\text{K} = 2048$$

$$\text{Cache组个数} = 2048 / 4 = 512 = 2^9$$

标记	组地址	块内地址
16	9	6

块大小 $16 \times 4 = 64 \text{ B}$

$64 = 2^6$ ，块内地址6位

Cache组地址9位

应用题1:

假定某机主存空间大小2GB，按字节编址。Cache的容量128KB，块大小为16个字，每个字32位，采用四路组相联映射方式。

(2) 设Cache初态为空，依次读主存第0, 1, 2, ..., 99号单元读出100个字（主存一次读出一个字，并重复按此次序读10次，命中率为多少？

解:

主存第0, 1, 2, ..., 15号字在第一个块内

主存第16, 17, 18, ..., 31号字在第二个块内

主存第32, 33, 34, ..., 47号字在第一个块内

主存第48, 49, 50, ..., 63号字在第一个块内

主存第64, 65, 66, ..., 79号字在第一个块内

主存第80, 81, 82, ..., 95号字在第一个块内

主存第96, 97, 98, 99号字在第一个块内

一共占用7个块

对应7个块，只有加载该块的第一个字时才会不命中，其他情况均命中

共计访存 $100 \times 10 = 1000$ 次，不命中只有7次，故而命中率为

$$\frac{1000 - 7}{1000} \times 100\% = 99.3\%$$

应用题1:

假定某机主存空间大小2GB，按字节编址。Cache的容量128KB，块大小为16个字，每个字32位，采用四路组相联映射方式。

(3) 若主存的有取周期是Cache的6倍，则采用Cache后存储器的速度提高了多少？

解:

假设访问Cache时间为 T ，则访问主存的时间为 $T_m = 6T$

由(2)可知，Cache命中率为99.3%，那么可以得到，采用Cache后，平均访问时间为

$$T_c = 0.993 \times T + 0.007 \times 6T = 1.035T$$

$$\frac{T_m}{T_c} = \frac{6T}{1.035T} \approx 5.8 \quad \longrightarrow \quad \text{速度提高了 } 5.8 - 1 = 4.8 \text{ 倍}$$

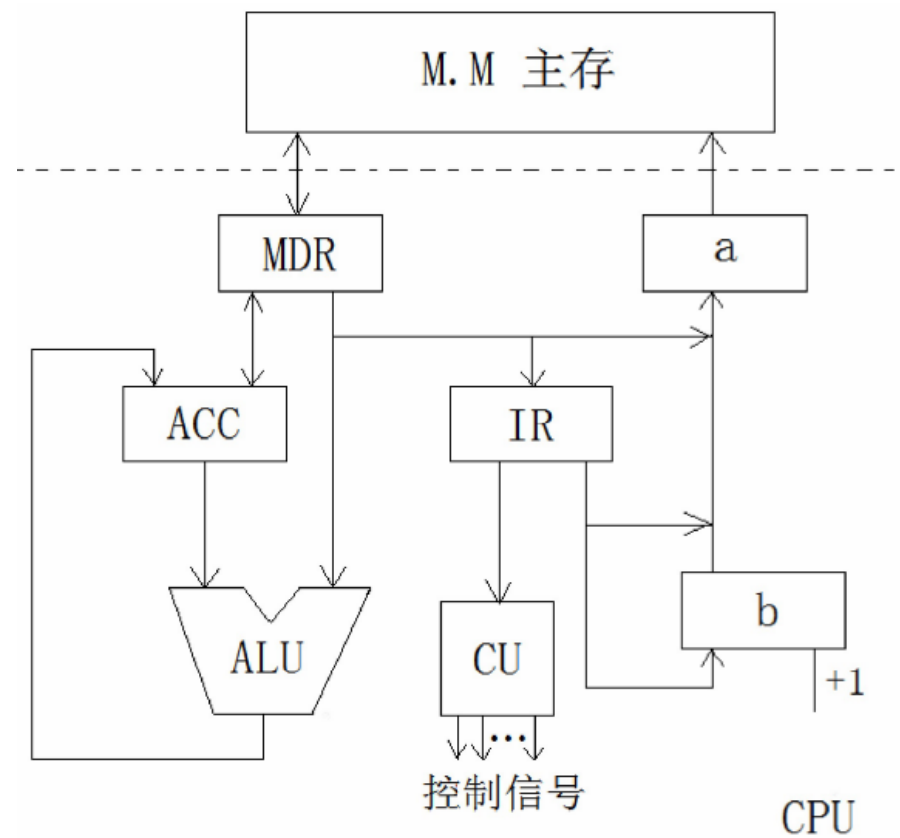
应用题2、

已知某模型机字长=存储字长=指令字长，部分数据通路如右图所示，各部分之间的连线表示数据通路，箭头表示信息传递方向。

要求：

- (1) 标明a、b两个寄存器的具体名称；
- (2) 若存储器容量为 $64M \times 32$ 位，指出图中各寄存器的位数；
- (3) 写出控制器完成指令（包括取指阶段）**ADD ACC, [mem]**发出的全部控制信号（用符号表示，如 **MDR→IR**）。

注：指令**ADD ACC, [mem]**的功能是将mem所指的主存单元内容与ACC的内容相加，结果存回ACC。

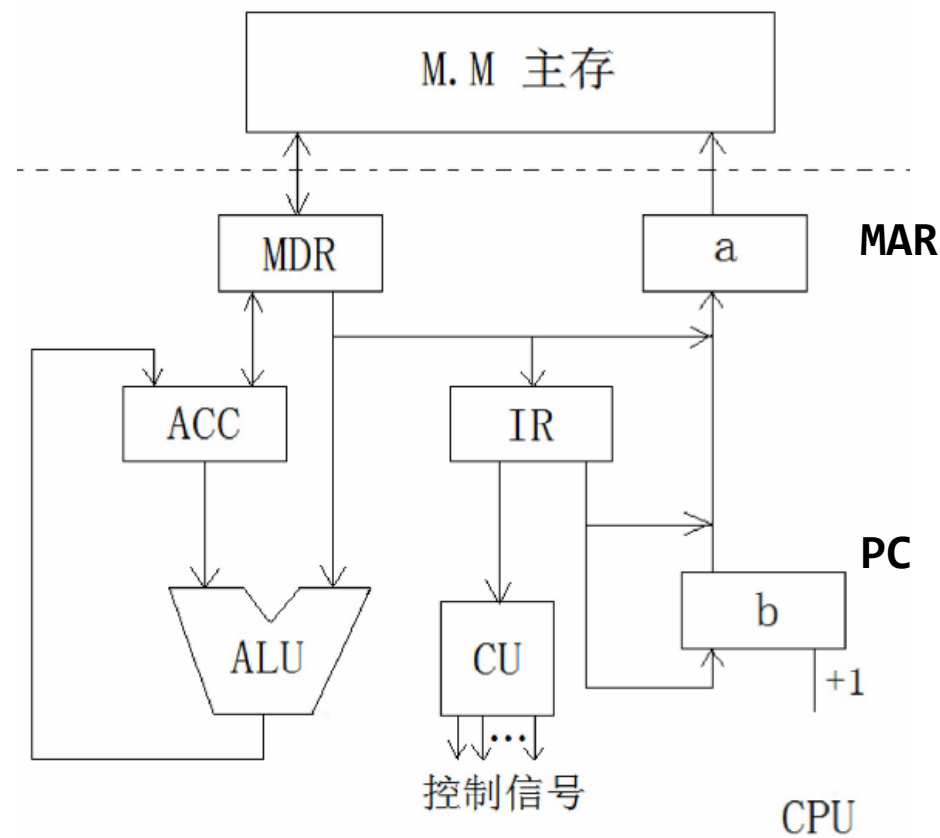


应用题2、

已知某模型机字长=存储字长=指令字长，部分数据通路如右图所示，各部分之间的连线表示数据通路，箭头表示信息传递方向。

要求：

(1) 标明a、b两个寄存器的具体名称；



已知某模型机字长=存储字长=指令字长，部分数据通路如右图所示，各部分之间的连线表示数据通路，箭头表示信息传递方向。

(2) 若存储器容量为 $64\text{M} \times 32$ 位, 指出图中各寄存器的位数;

The diagram illustrates the internal structure of a CPU and its connection to Main Memory (M.M 主存). The CPU is divided into several functional units:

- M.M 主存 (Main Memory):** Located at the top, it is connected to the CPU via a dashed line.
- MAR (Memory Address Register):** Labeled 'a', it is a 26-bit register that receives data from the M.M 主存 and sends it to the MDR.
- MDR (Memory Data Register):** Labeled '32', it is a 32-bit register that receives data from the MAR and sends it to the ALU and the CU.
- ACC (Accumulator):** Labeled '32', it is a 32-bit register that receives data from the ALU and sends it back to the ALU.
- IR (Instruction Register):** Labeled '32', it receives data from the MDR and sends it to the CU.
- ALU (Arithmetic Logic Unit):** A trapezoidal unit that receives data from the ACC and the MDR, performing arithmetic and logical operations.
- CU (Control Unit):** A rectangular unit that receives data from the IR and sends control signals (控制信号) to the other units.
- PC (Program Counter):** Labeled 'b', it is a 26-bit register that receives data from the CU and sends it to the MAR. It also has a '+1' increment input.

The CPU is labeled 'CPU' at the bottom right.

应用题2、

已知某模型机字长=存储字长=指令字长，部分数据通路如右图所示，各部分之间的连线表示数据通路，箭头表示信息传递方向。

要求：

(3) 写出控制器完成指令（包括取指阶段）ADD ACC, [mem]发出的全部控制信号

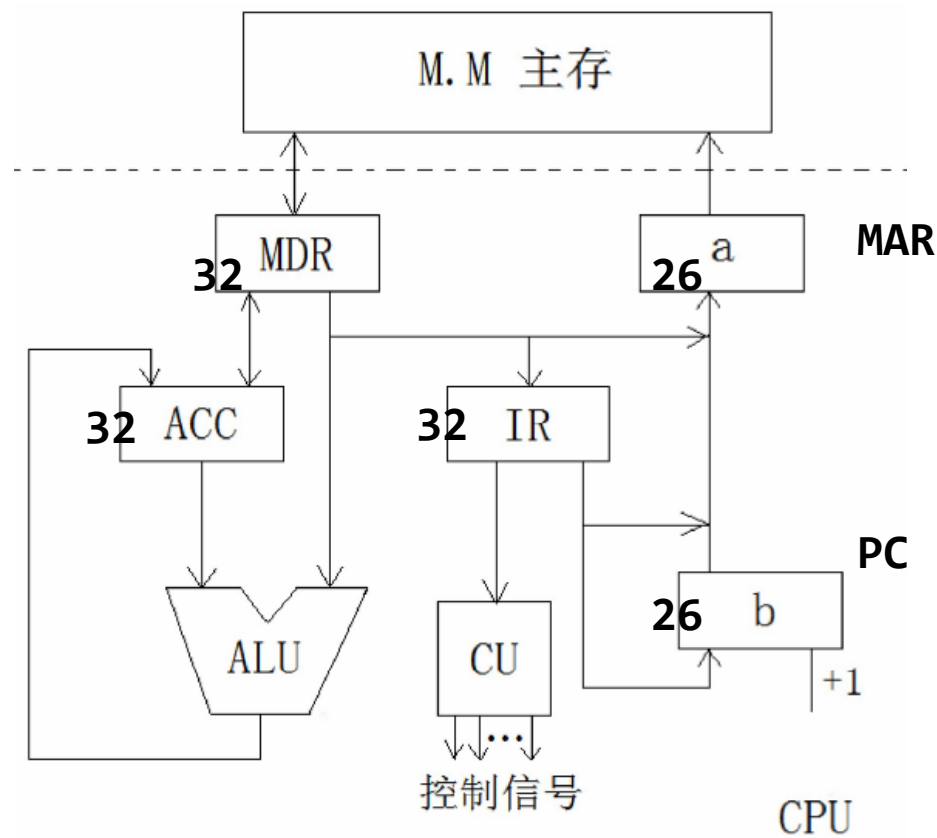
解：

$Ad(IR) \rightarrow MAR$

$1 \rightarrow R$

$M(MAR) \rightarrow MDR$

$(ACC) + (MDR) \rightarrow ACC$



答案仅供参考，欢迎探讨交流 :)

邮箱: erqitao@qq.com

QQ: 204370849

@zengtao

2020-11

考研加油！