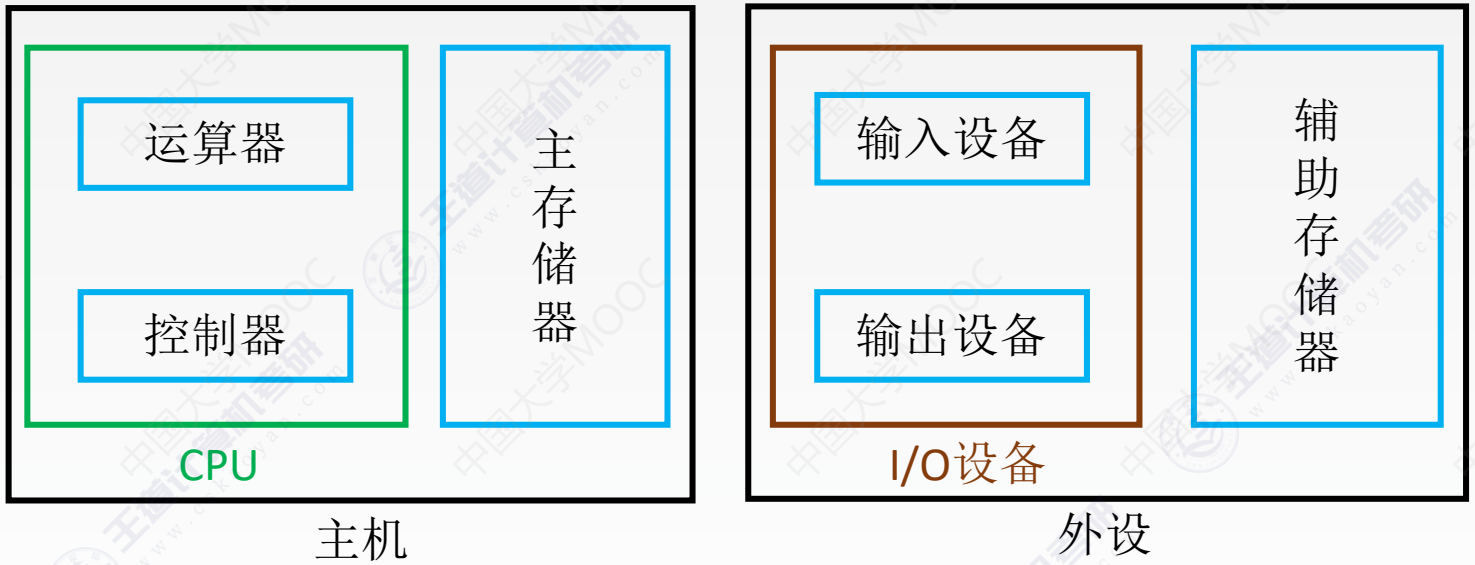
王道考研——组成原理

WWW.CSKAOYAN.COM

第五章 中央处理器

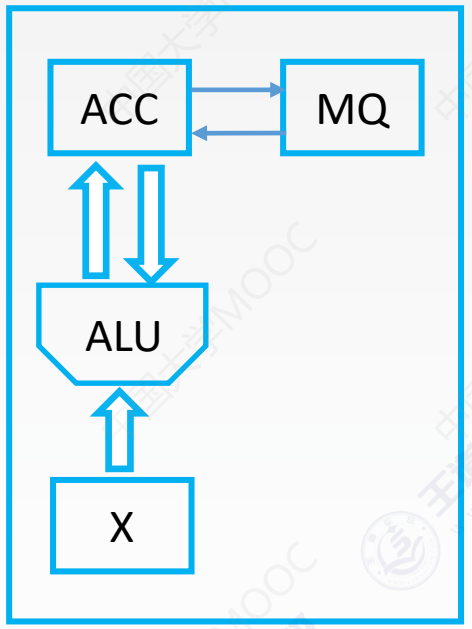
王道24考研交流群：769832062

现代计算机的结构



运算器的基本组成

运算器



运算器：用于实现算术运算（如：加减乘除）、逻辑运算（如：与或非）

ACC：　累加器，用于存放操作数，或运算结果。
MQ：　乘商寄存器，在乘、除运算时，用于存放操作数或运算结果。
X：　通用的操作数寄存器，用于存放操作数
ALU：　算术逻辑单元，通过内部复杂的电路实现算数运算、逻辑运算

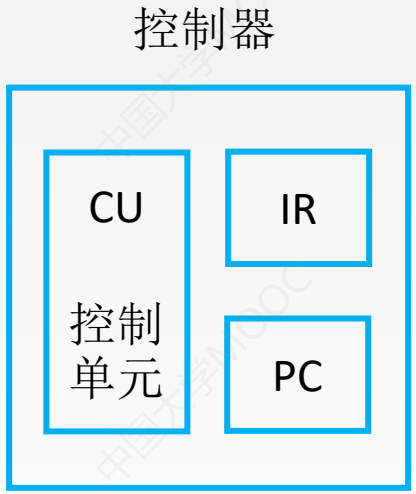
	加	减	乘	除
ACC	被加数、和	被减数、差	乘积高位	被除数、余数
MQ			乘数、乘积低位	商
X	加数	减数	被乘数	除数

Accumulator

Multiple-Quotient Register

Arithmetic and Logic Unit

控制器的基本组成



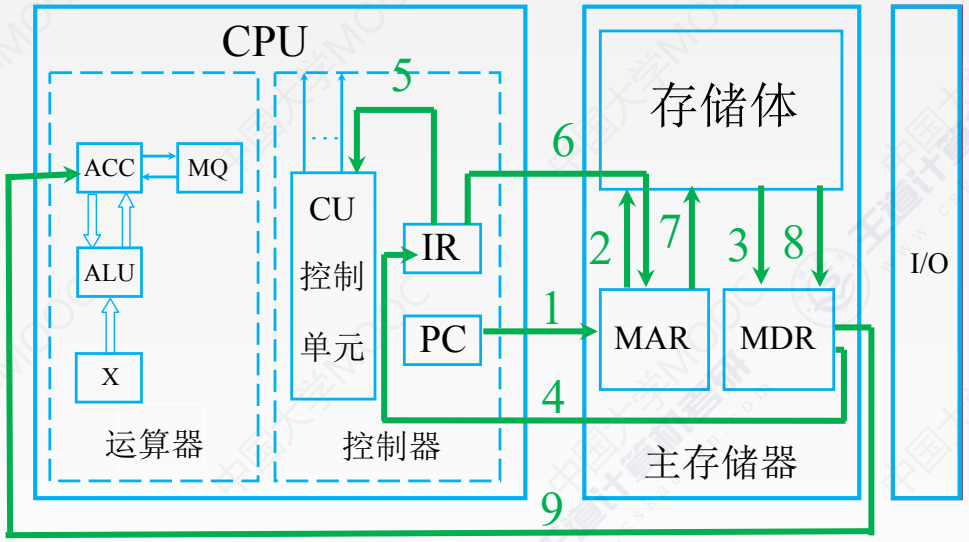
CU：控制单元，分析指令，给出控制信号
IR：指令寄存器，存放当前执行的指令
PC：程序计数器，存放下一条指令地址，有自动加1功能

Control Unit
Instruction Register
Program Counter

完成一条指令
- 取指令 —— PC
- 分析指令 —— IR
- 执行指令 —— CU

（取指令、分析指令 → 取指；执行指令 → 执行）

计算机的工作过程



PC

主存地址	指令		注释
	操作码	地址码	
0	000001	0000000101	取数*a*至ACC
1	000100	0000000110	乘*b*得*ab* ,存于ACC中
2	000011	0000000111	加*c*得*ab+c* ,存于ACC中
3	000010	0000001000	将*ab+c* ,存于主存单元
4	000110	0000000000	停机
5	*0000000000000010*		原始数据*a=2*
6	*0000000000000011*		原始数据*b=3*
7	*0000000000000001*		原始数据*c=1*
8	*0000000000000000*		原始数据*y=0*

初：(PC)=0，指向第一条指令的存储地址

#1：(PC)→MAR，导致(MAR)=0

#3：M(MAR)→MDR，导致(MDR)=000001 0000000101

#4：(MDR)→IR，导致(IR)=000001 0000000101

#5：OP(IR)→CU，指令的操作码送到CU，CU分析后得知，这是“取数”指令

#6：Ad(IR)→MAR，指令的地址码送到MAR，导致(MAR)=5

#8：M(MAR)→MDR，导致(MDR)=*0000000000000010=2*

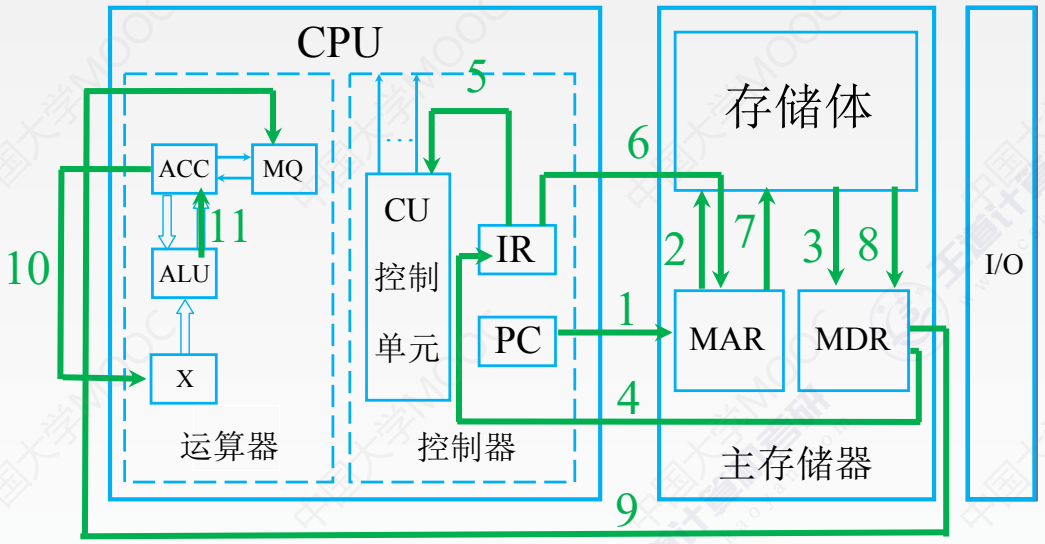
#9：(MDR)→ACC，导致(ACC)=*0000000000000010=2*

取指令（#1~#4）

分析指令（#5）

执行取数指令（#6 ~ #9）

计算机的工作过程



PC

主存地址	指令		注释
	操作码	地址码	
0	000001	0000000101	取数*a*至ACC
1	000100	0000000110	乘*b*得*ab* ,存于ACC中
2	000011	0000000111	加*c*得*ab+c* ,存于ACC中
3	000010	0000001000	将*ab+c* ,存于主存单元
4	000110	0000000000	停机
5	*0000000000000010*		原始数据*a=2*
6	*0000000000000011*		原始数据*b=3*
7	*0000000000000001*		原始数据*c=1*
8	*0000000000000000*		原始数据*y=0*

上一条指令取指后PC自动+1，(PC)=1；执行后，(ACC)=2

#1：(PC)→MAR，导致(MAR)=1

#3：M(MAR)→MDR，导致(MDR)=**000100 0000000110**

#4：(MDR)→IR，导致(IR)= **000100 0000000110**

#5：OP(IR)→CU，指令的操作码送到CU， CU分析后得知，这是“乘法”指令

#6：Ad(IR)→MAR，指令的地址码送到MAR，导致(MAR)=6

#8：M(MAR)→MDR，导致(MDR)=***0000000000000011=3***

#9：(MDR)→MQ，导致(MQ)=***0000000000000011=3***

#10：(ACC)→X，导致(X)=2

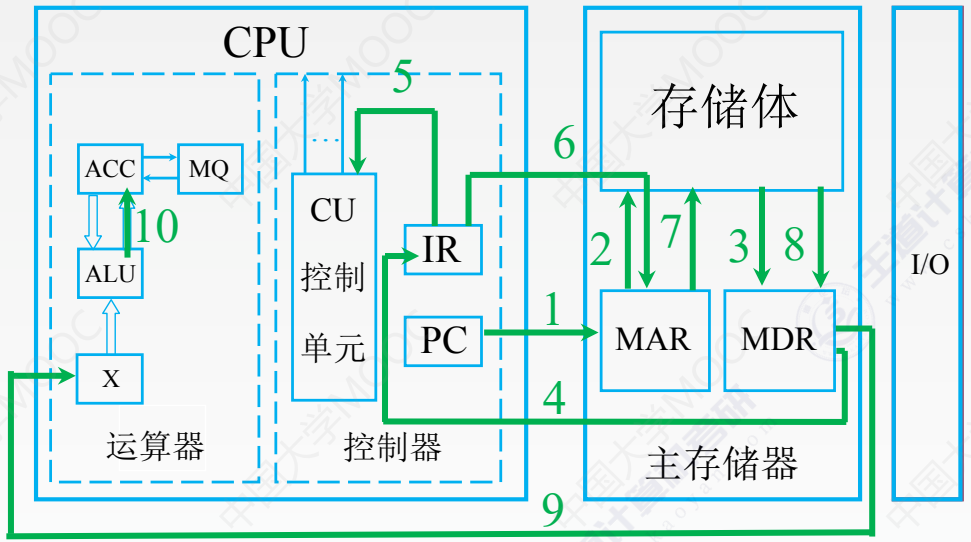
#11：(MQ)*(X)→ACC，由ALU实现乘法运算，导致(ACC)=6，如果乘积太大，则需要MQ辅助存储

取指令（#1~#4）

分析指令（#5）

执行乘法指令（#6 ~ #11）

计算机的工作过程



PC

主存地址	指令		注释
	操作码	地址码	
0	000001	0000000101	取数*a*至ACC
1	000100	0000000110	乘*b*得*ab* ,存于ACC中
2	000011	0000000111	加*c*得*ab*+*c* ,存于ACC中
3	000010	0000001000	将*ab*+*c* ,存于主存单元
4	000110	0000000000	停机
5	*0000000000000010*		原始数据*a=2*
6	*0000000000000011*		原始数据*b=3*
7	*0000000000000001*		原始数据*c=1*
8	*0000000000000000*		原始数据*y=0*

上一条指令取指后(PC)=2，执行后，(ACC)=6

#1：(PC)→MAR，导致(MAR)=2

#3：M(MAR)→MDR，导致(MDR)= 000011 0000000111

#4：(MDR)→IR，导致(IR)= 000011 0000000111

#5：OP(IR)→CU，指令的操作码送到CU， CU分析后得知，这是“加法”指令

#6：Ad(IR)→MAR，指令的地址码送到MAR，导致(MAR)=7

#8：M(MAR)→MDR，导致(MDR)=*0000000000000001=1*

#9：(MDR)→X，导致(X)=*0000000000000001=1*

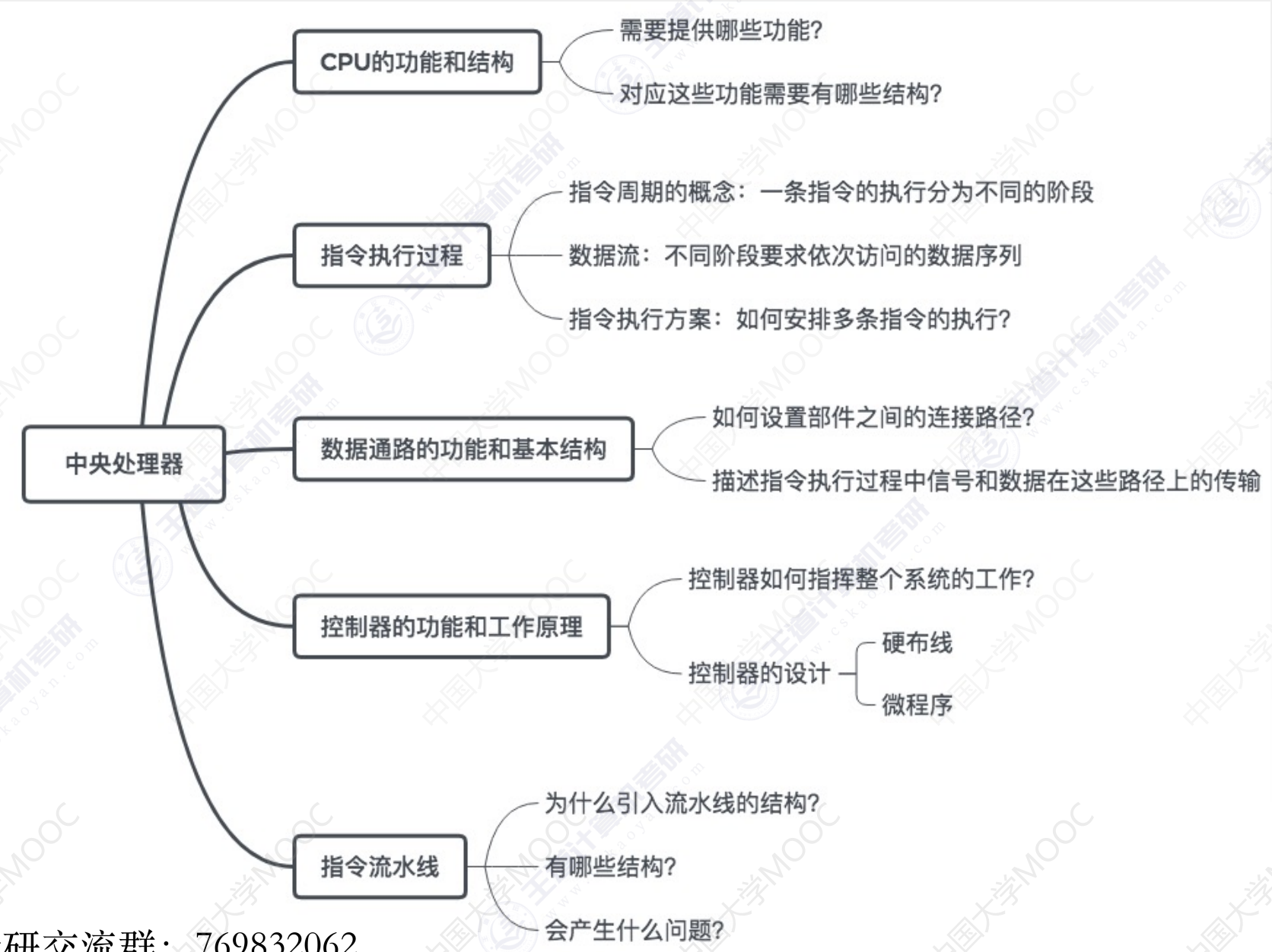
#10：(ACC)+(X)→ACC，导致(ACC)=7，由ALU实现加法运算

取指令（#1~#4）

分析指令（#5）

执行加法指令（#6 ~ #10）

本章总览



本节内容

中央处理器

CPU的功能和基本结构

CPU的功能

1. **指令控制**。完成取指令、分析指令和执行指令的操作，即程序的顺序控制。

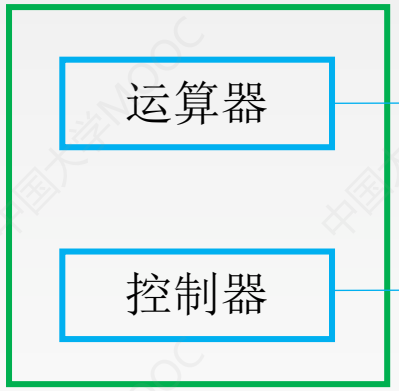
2. **操作控制**。一条指令的功能往往是由若干操作信号的组合来实现的。CPU管理并产生由内存取出的每条指令的操作信号，把各种操作信号送往相应的部件，从而控制这些部件按指令的要求进行动作。

3. **时间控制**。对各种操作加以时间上的控制。时间控制要为每条指令按时间顺序提供应有的控制信号。

4. **数据加工**。对数据进行算术和逻辑运算。

5. **中断处理**。对计算机运行过程中出现的异常情况和特殊请求进行处理。

运算器和控制器的功能



运算器 → 对数据进行加工

控制器 → 协调并控制计算机各部件执行程序的指令序列，
基本功能包括取指令、分析指令、执行指令

取指令：自动形成指令地址；自动发出取指令的命令。

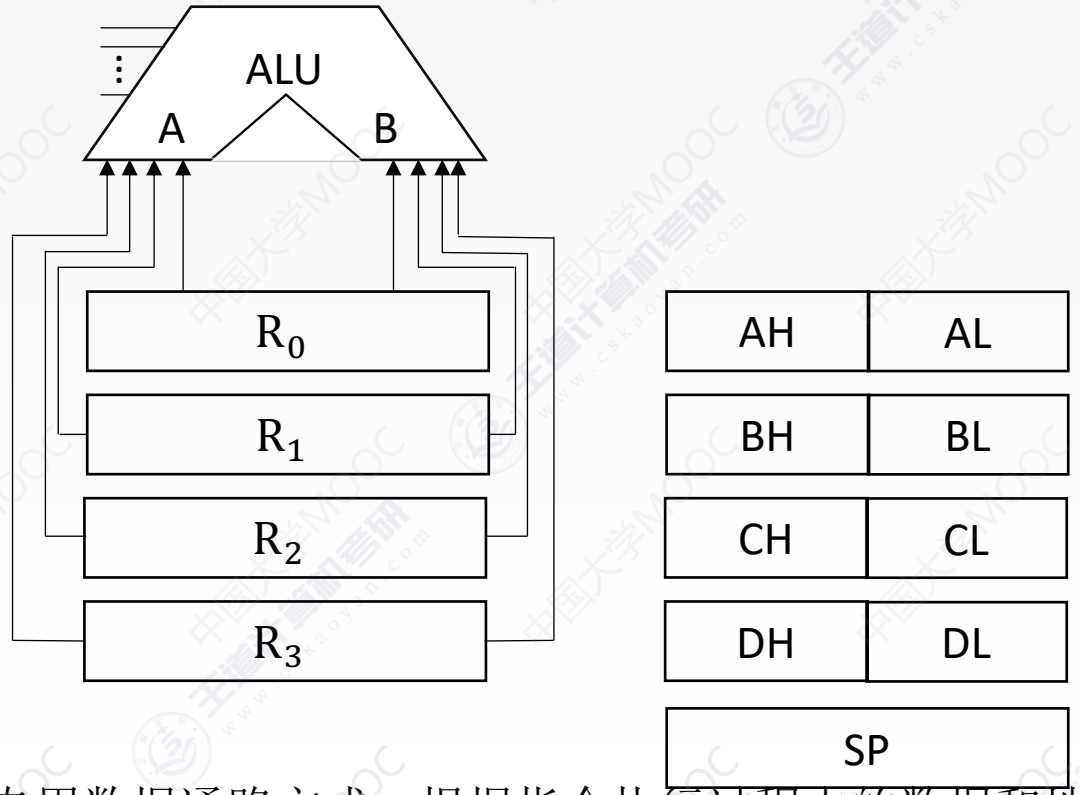
分析指令：操作码译码(分析本条指令要完成什么操作)；
产生操作数的有效地址。

执行指令：根据分析指令得到的“操作命令”和“操作数地址”，
形成操作信号控制序列，控制运算器、存储器以及I/O
设备完成相应的操作。

中断处理：管理总线及输入输出；处理异常情况(如掉电)和特殊请
求(如打印机请求打印一行字符)。

运算器的基本结构

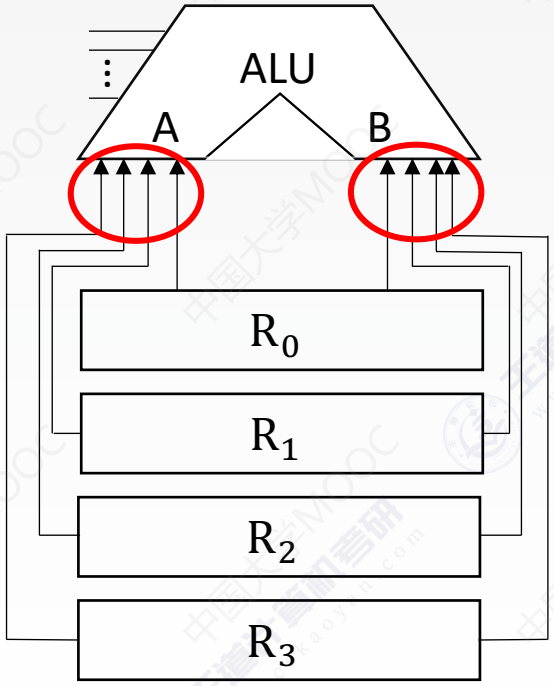
1. 算术逻辑单元：主要功能是进行算术/逻辑运算。
2. 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。



专用数据通路方式：根据指令执行过程中的数据和地址的流动方向安排连接线路。

运算器的基本结构

1. 算术逻辑单元：主要功能是进行算术/逻辑运算。
2. 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。



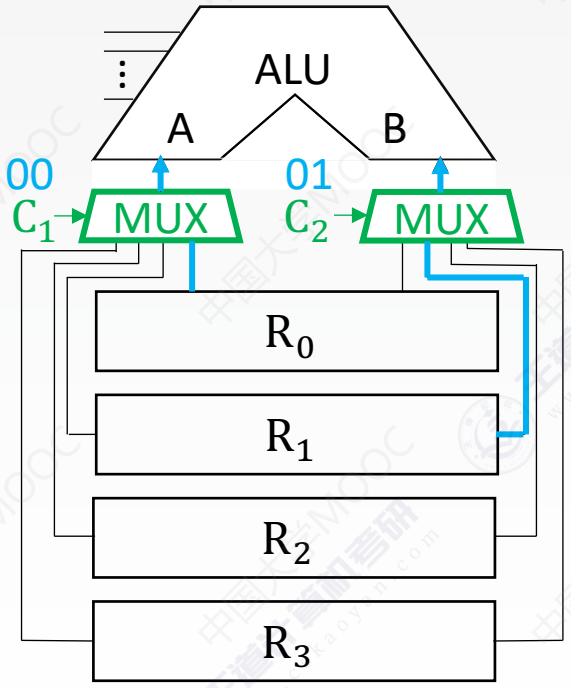
如果直接用导线连接，相当于多个寄存器同时并且一直向ALU传输数据

解决方法1. 使用多路选择器

专用数据通路方式：根据指令执行过程中的数据和地址的流动方向安排连接线路。

运算器的基本结构

1. 算术逻辑单元：主要功能是进行算术/逻辑运算。
2. 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。



如果直接用导线连接，相当于多个寄存器同时并且一直向ALU传输数据

解决方法1. 使用多路选择器

根据控制信号选择一路输出

解决方法2. 使用三态门

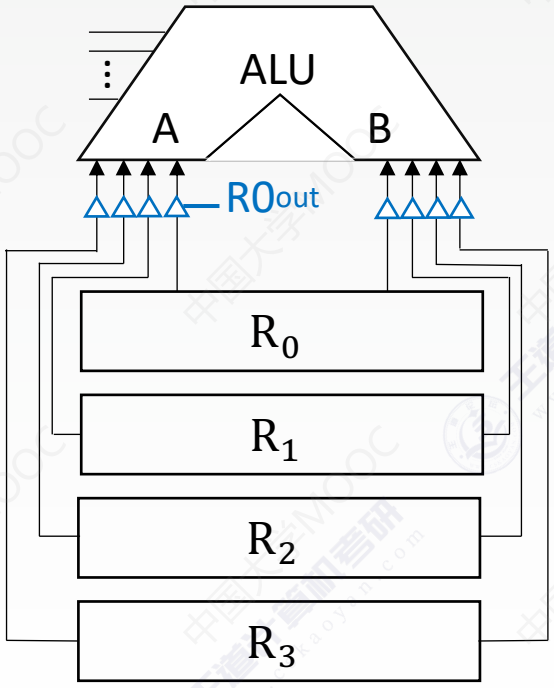
可以控制每一路是否输出

专用数据通路方式：根据指令执行过程中的数据和地址的流动方向安排连接线路。

运算器的基本结构

1. 算术逻辑单元：主要功能是进行算术/逻辑运算。
2. 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。

CPU内部单总线方式：将所有寄存器的输入端和输出端都连接到一条公共的通路上。



如果直接用导线连接，相当于多个寄存器同时并且一直向ALU传输数据

解决方法1. 使用多路选择器

根据控制信号选择一路输出

解决方法2. 使用三态门

可以控制每一路是否输出

如：ROout为1时R_0中的数据输出到A端，

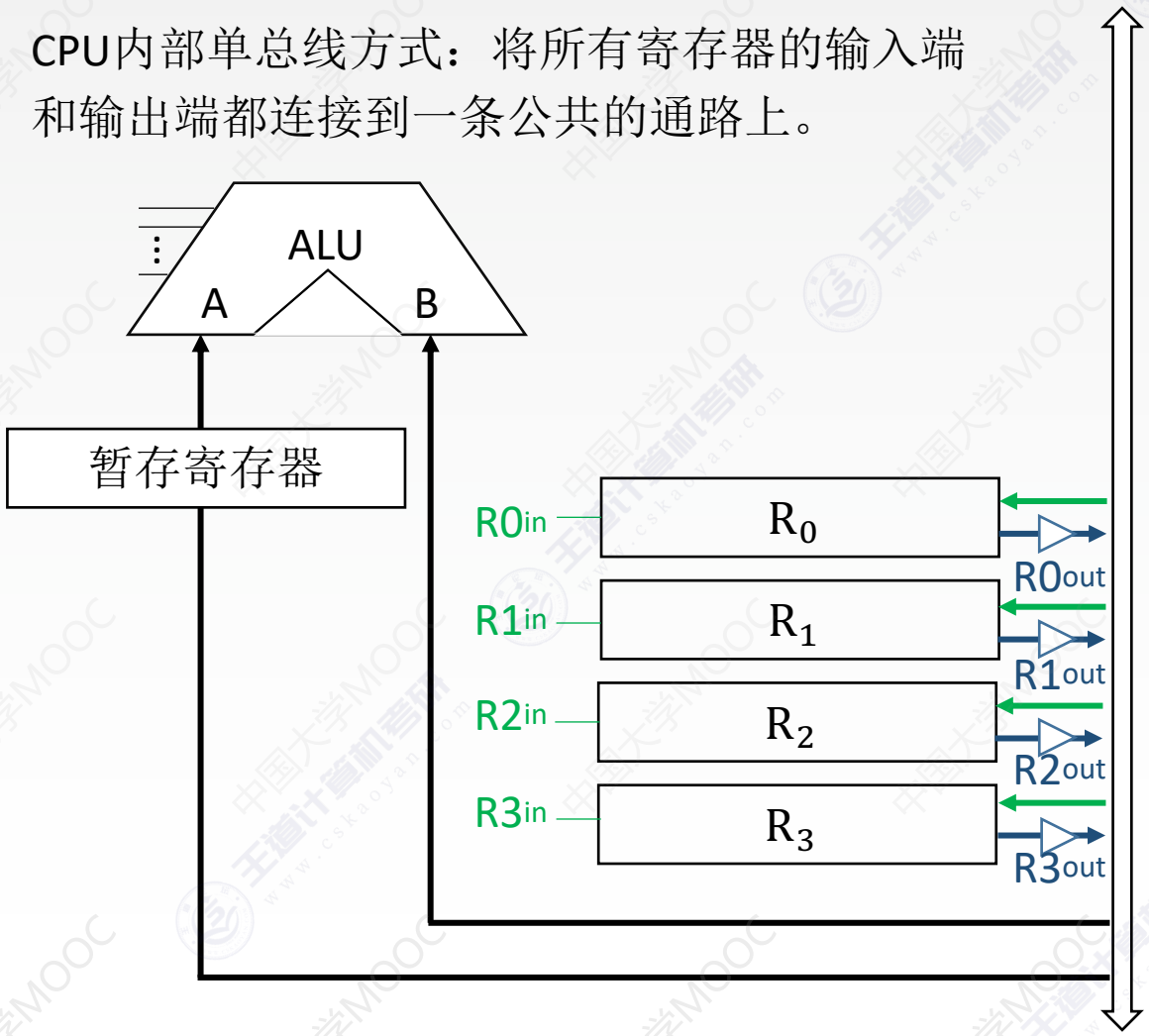
ROout为0时R_0中的数据无法输出到A端

性能较高，基本不存在数据冲突现象，但结构复杂，硬件量大，不易实现。

专用数据通路方式：根据指令执行过程中的数据和地址的流动方向安排连接线路。

运算器的基本结构

CPU内部单总线方式：将所有寄存器的输入端和输出端都连接到一条公共的通路上。

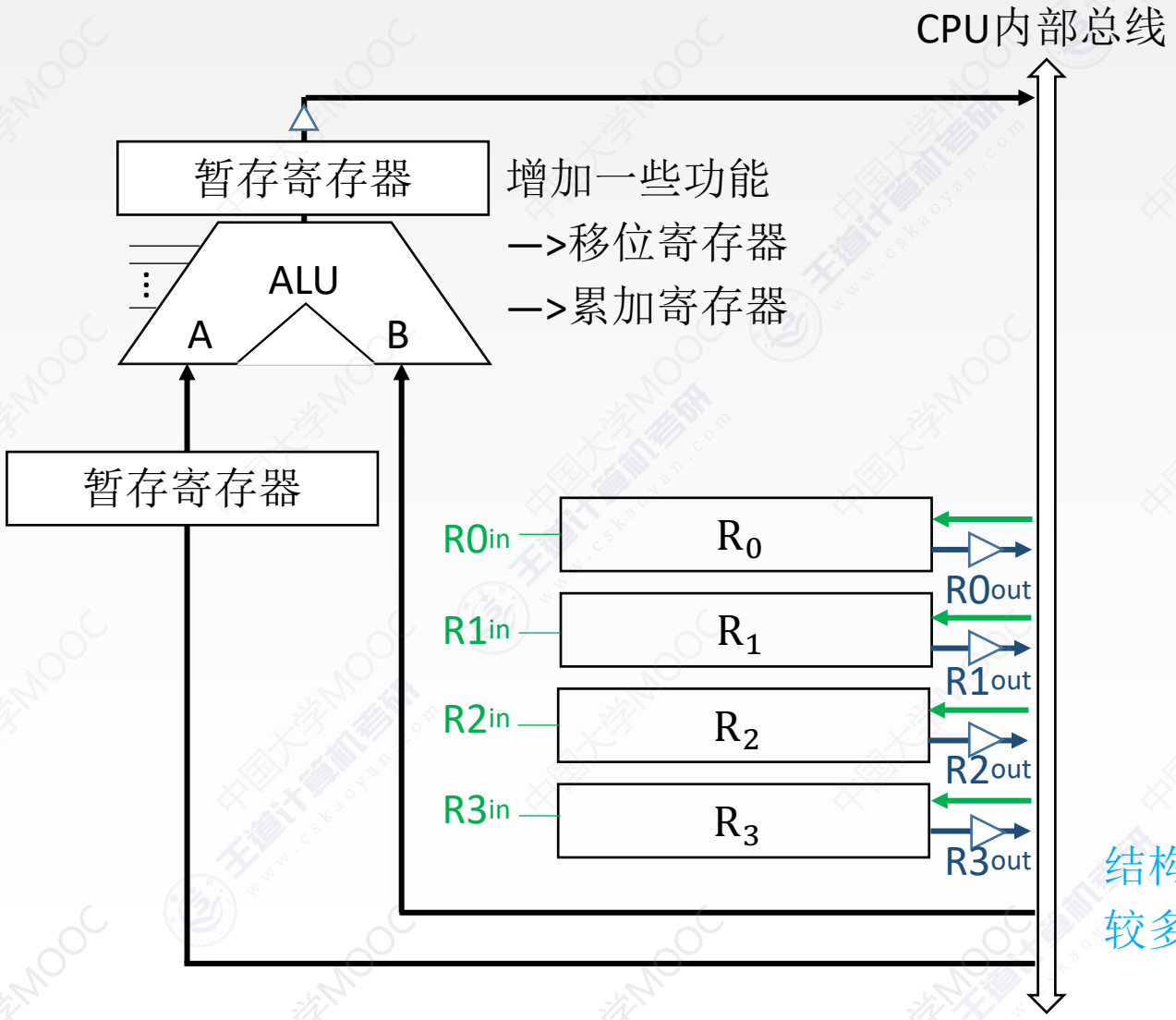


1. 算术逻辑单元：主要功能是进行算术/逻辑运算。
2. 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。
3. 暂存寄存器：用于暂存从主存读来的数据，这个数据不能存放在通用寄存器中，否则会破坏其原有内容。

如：两个操作数分别来自主存和R_0，最后结果存回R_0，那么从主存中取来的操作数直接放入暂存器，就不会破坏运算前R_0的内容。

结构简单，容易实现，但数据传输存在较多冲突的现象，性能较低。

运算器的基本结构

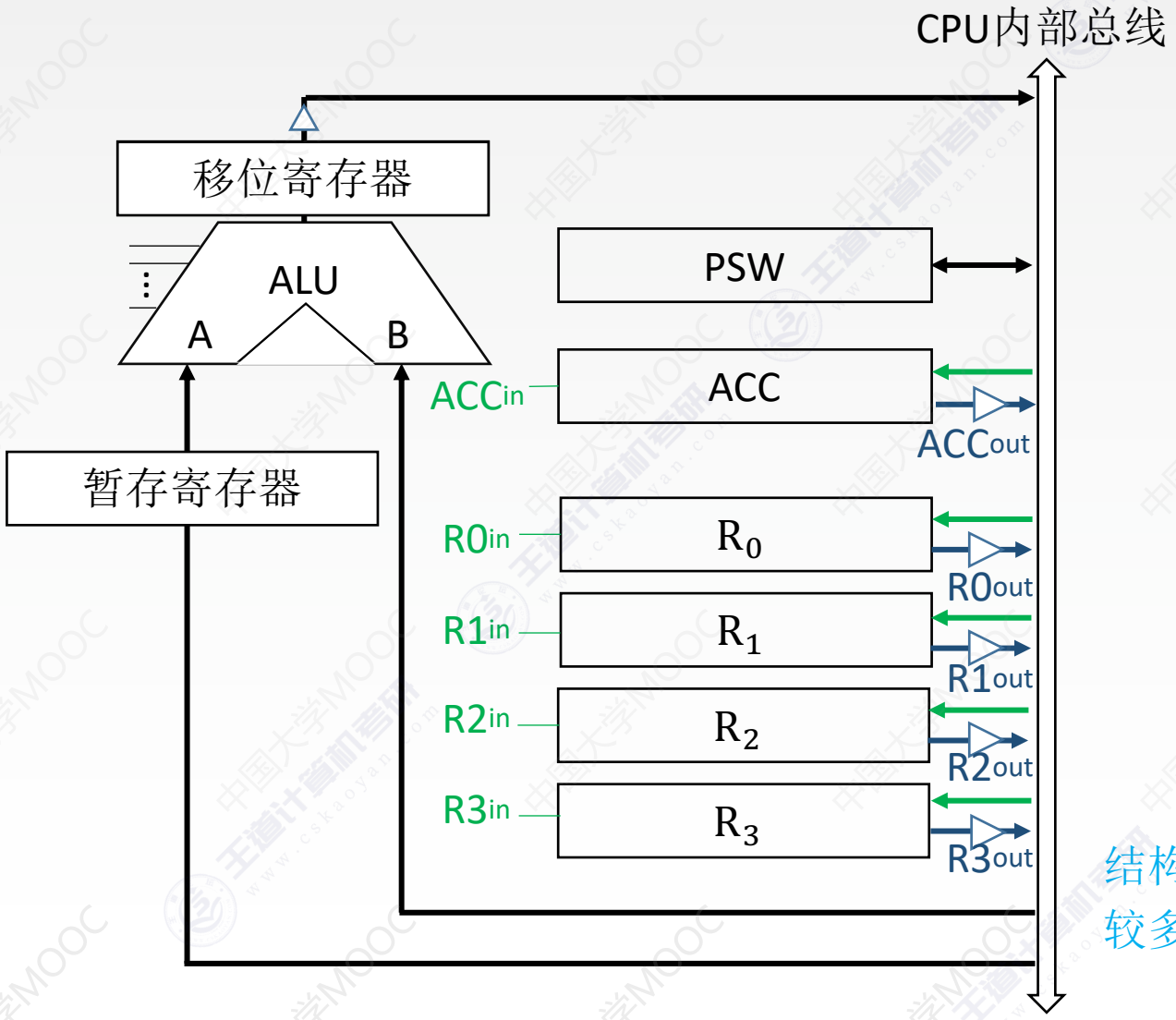


1. 算术逻辑单元：主要功能是进行算术/逻辑运算。
2. 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。
3. 暂存寄存器：用于暂存从主存读来的数据，这个数据不能存放在通用寄存器中，否则会破坏其原有内容。

如：两个操作数分别来自主存和R_0，最后结果存回R_0，那么从主存中取来的操作数直接放入暂存器，就不会破坏运算前R_0的内容。

结构简单，容易实现，但数据传输存在较多冲突的现象，性能较低。

运算器的基本结构

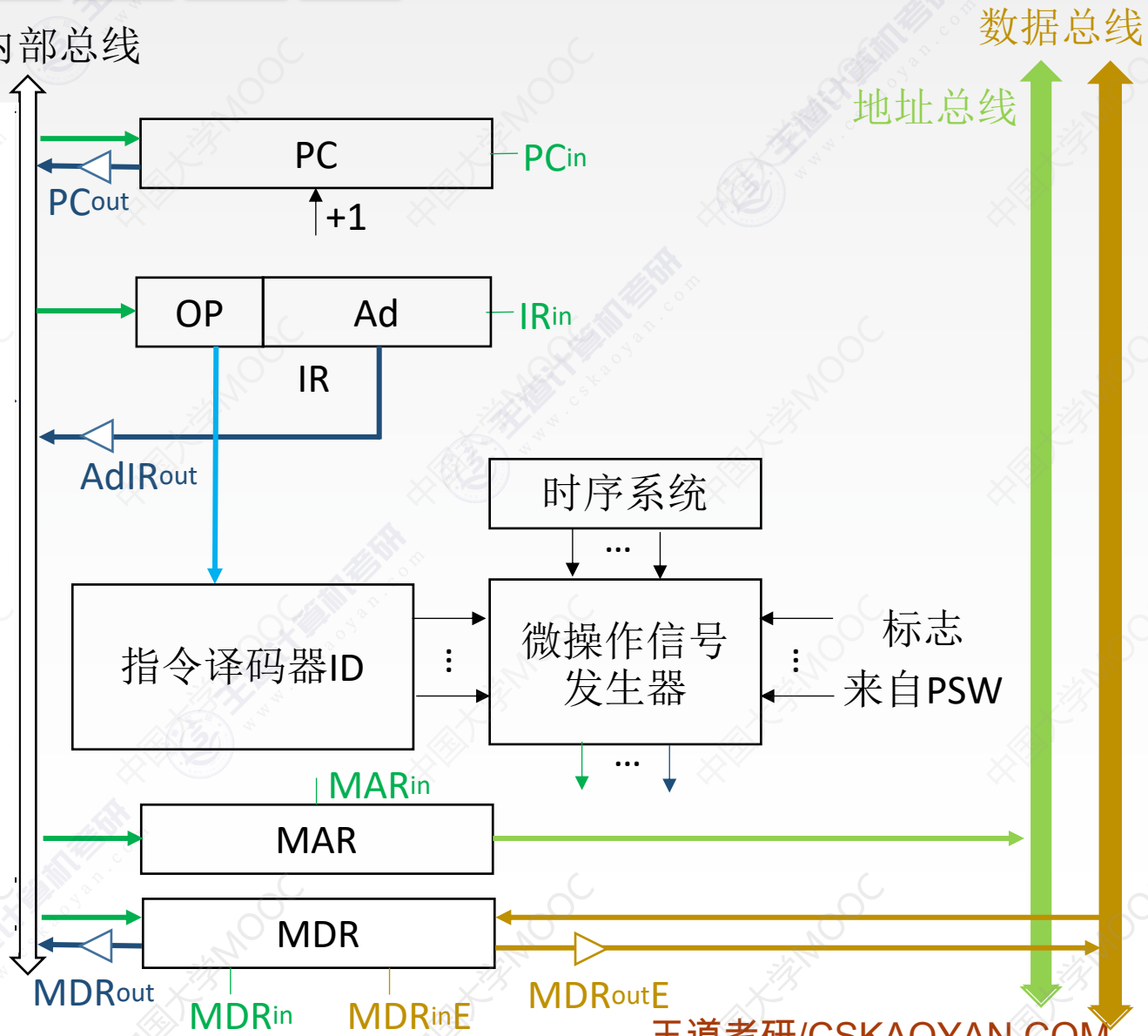


1. 算术逻辑单元：主要功能是进行算术/逻辑运算。
2. 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。
3. 暂存寄存器：用于暂存从主存读来的数据，这个数据不能存放在通用寄存器中，否则会破坏其原有内容。
4. 累加寄存器：它是一个通用寄存器，用于暂时存放ALU运算的结果信息，用于实现加法运算。
5. 程序状态字寄存器：保留由算术逻辑运算指令或测试指令的结果而建立的各种状态信息，如溢出标志（OP）、符号标志（SF）、零标志（ZF）、进位标志（CF）等。PSW中的这些位参与并决定微操作的形成。
6. 移位器：对运算结果进行移位运算。
7. 计数器：控制乘除运算的操作步数。

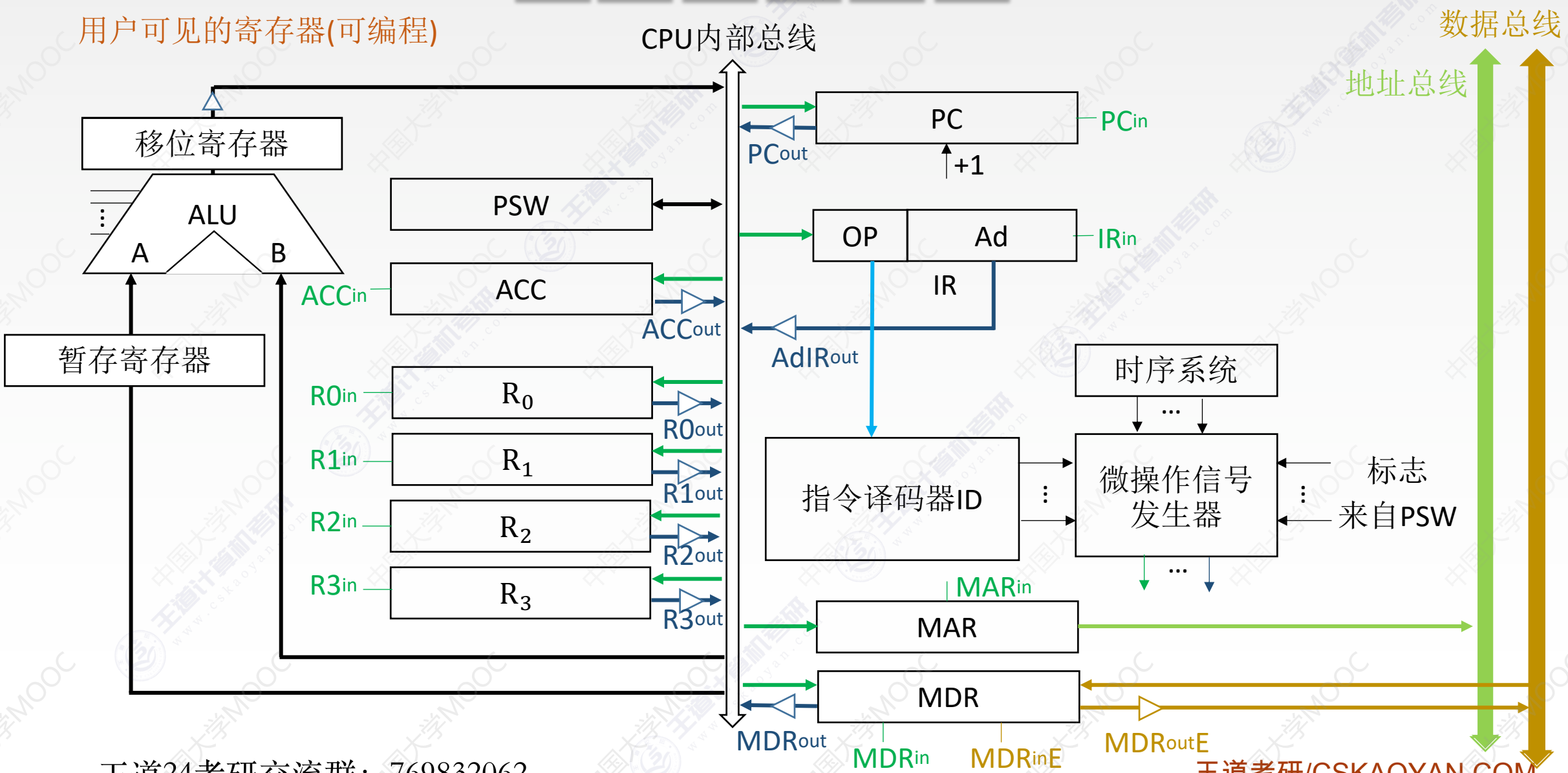
结构简单，容易实现，但数据传输存在较多冲突的现象，性能较低。

控制器的基本结构

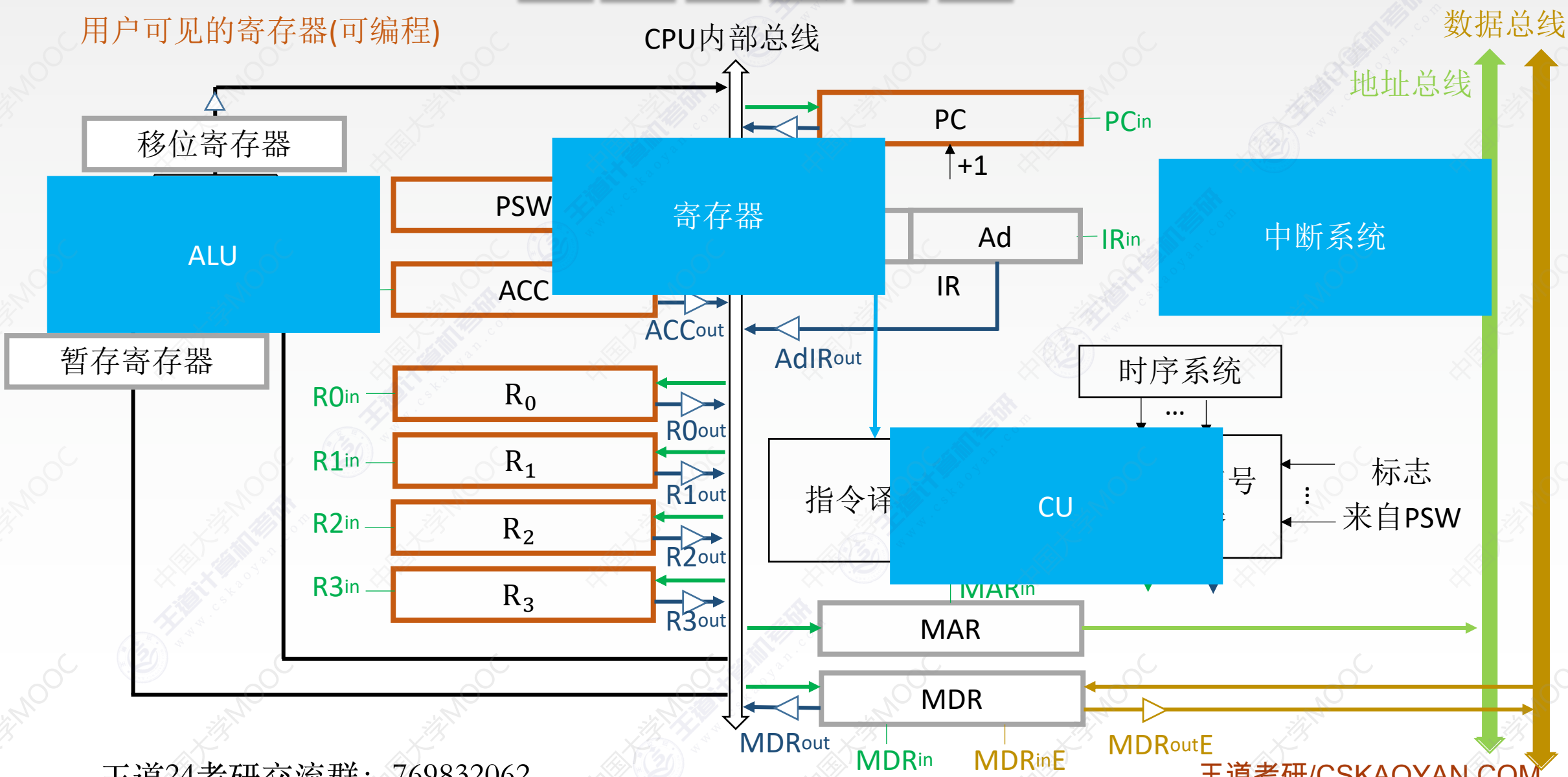
1. 程序计数器：用于指出下一条指令在主存中的存放地址。CPU就是根据PC的内容去主存中取指令的。因程序中指令（通常）是顺序执行的，所以PC有自增功能。
2. 指令寄存器：用于保存当前正在执行的那条指令。
3. 指令译码器：仅对操作码字段进行译码，向控制器提供特定的操作信号。
4. 微操作信号发生器：根据IR的内容（指令）、PSW的内容（状态信息）及时序信号，产生控制整个计算机系统所需的各种控制信号，其结构有组合逻辑型和存储逻辑型两种。
5.时序系统：用于产生各种时序信号，它们都是由统一时钟（CLOCK）分频得到。
6. 存储器地址寄存器：用于存放所要访问的主存单元的地址。
7. 存储器数据寄存器：用于存放向主存写入的信息或从主存中读出的信息。



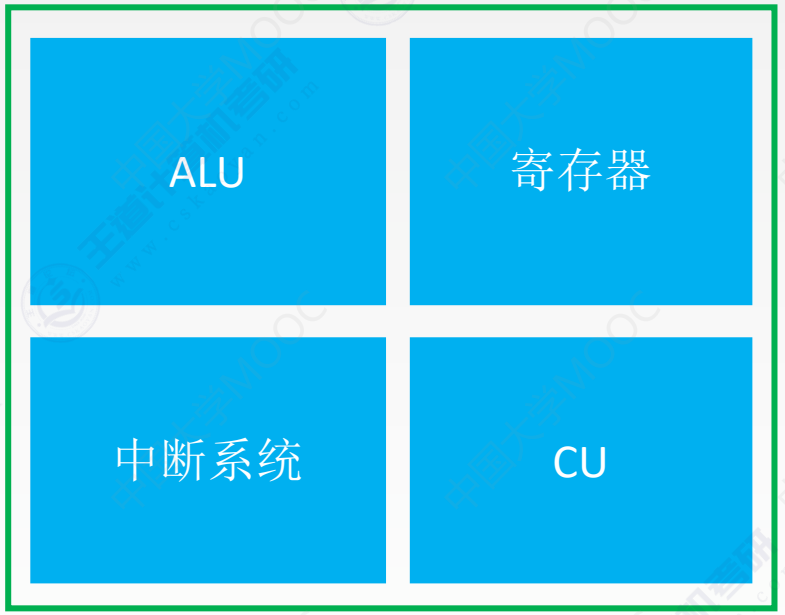
CPU的基本结构



CPU的基本结构



CPU的基本结构



本节回顾