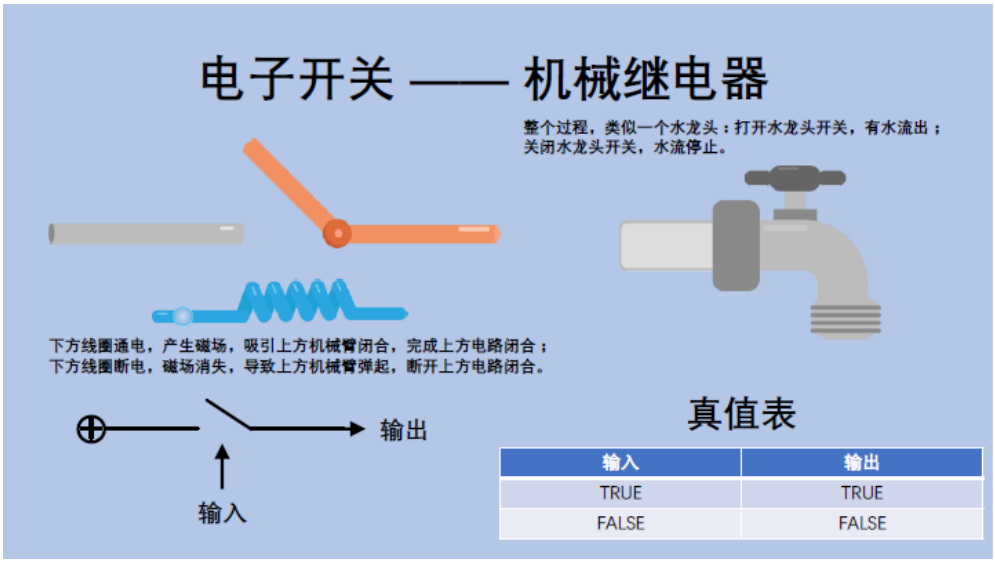
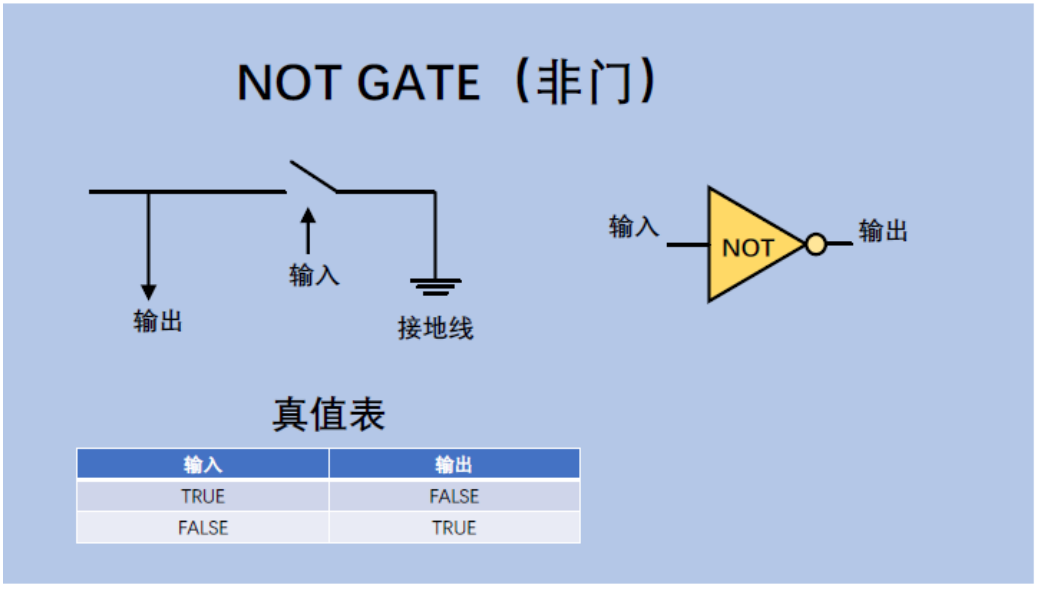
1、逻辑门

1. 电子开关——机械继电器

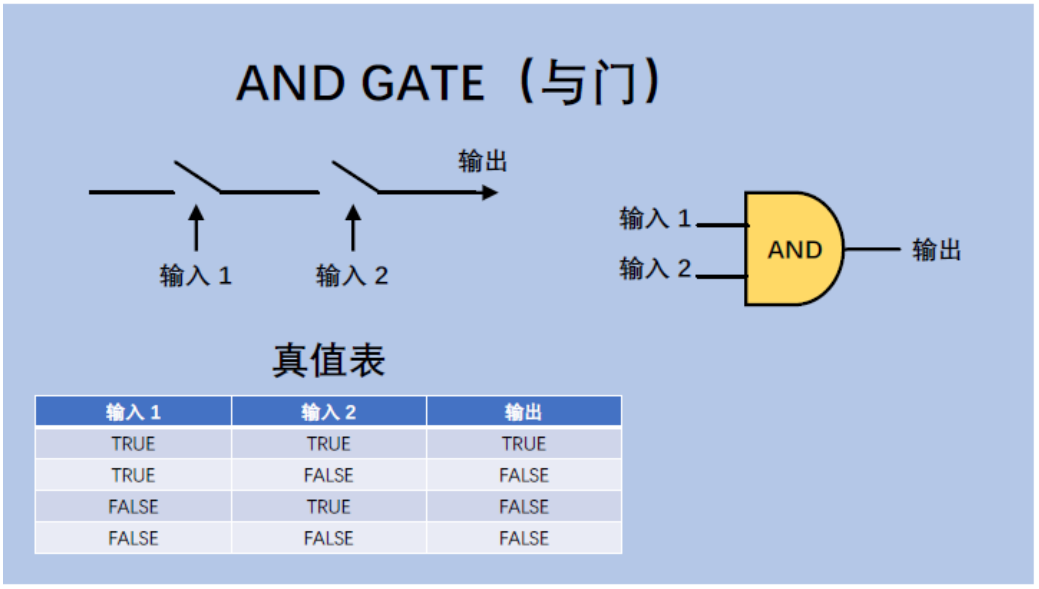


2. 门电路

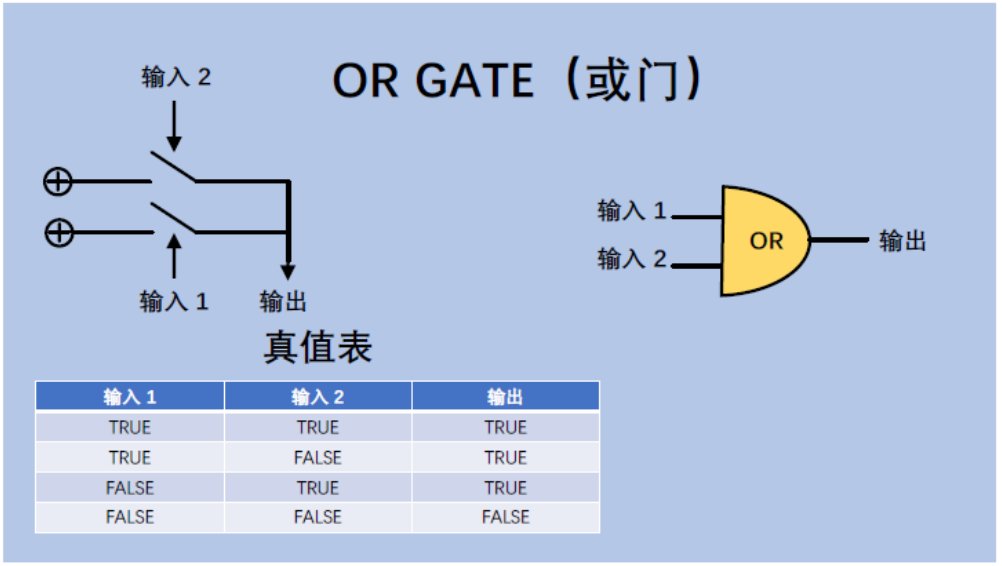
一个CPU上其实包含了很多的基本单元，每个单元其实就是一个“门电路”，“门电路”其实就是针对 1 位(bit) 的数据，进行逻辑运算。三种运算：与门、或门、非门。



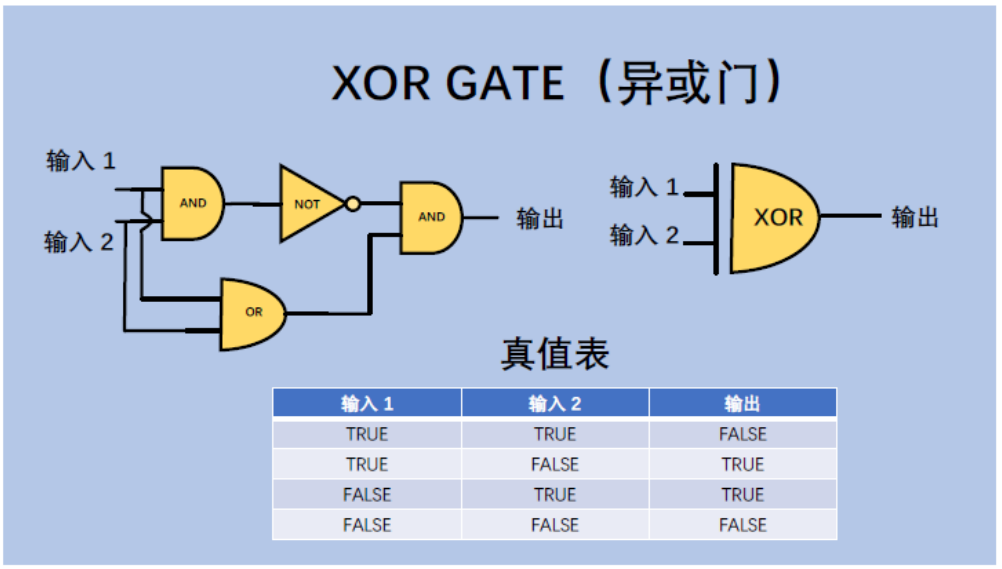
输入	输出
TRUE	FALSE
FALSE	TRUE



输入 1	输入 2	输出
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE



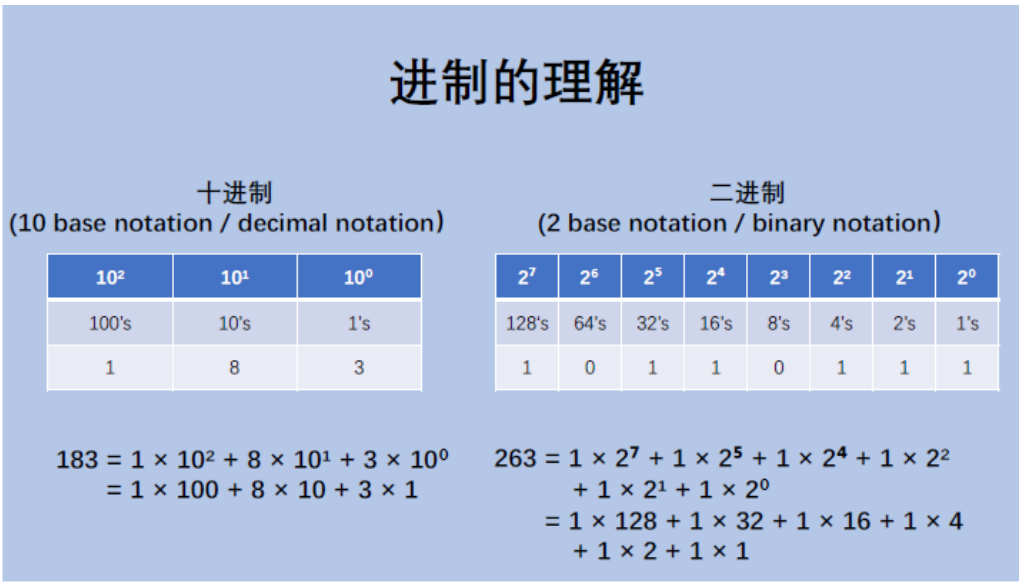
输入 1	输入 2	输出
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE



输入 1	输入 2	输出
TRUE	TRUE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

2、算数逻辑单元

1. 进制的理解



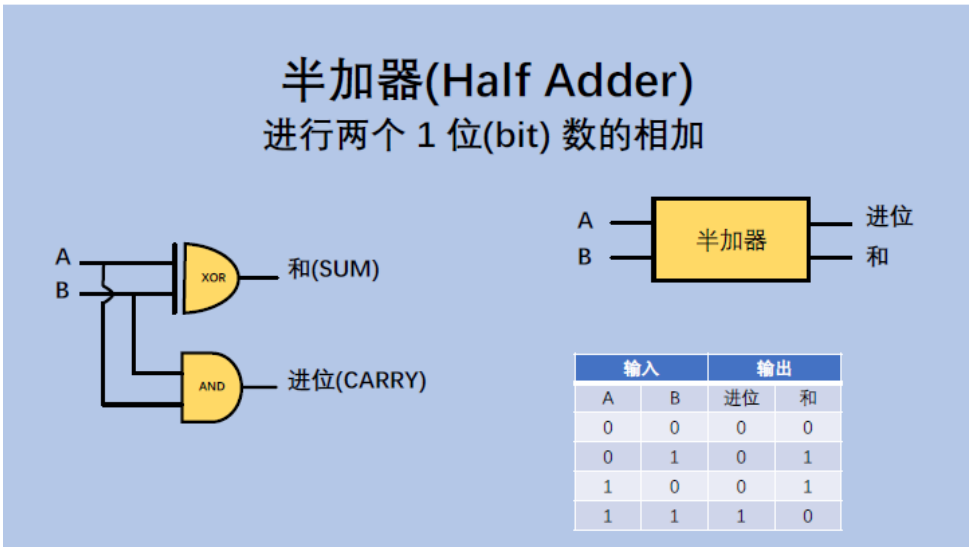
10^2	10^1	10^0
100's	10's	1's
1	8	3

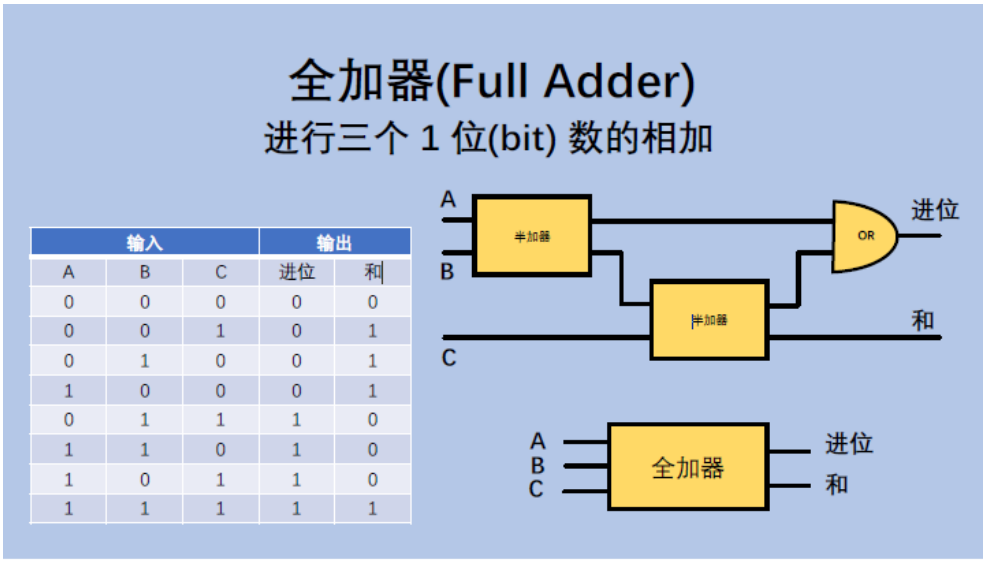
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128's	64's	32's	16's	8's	4's	2's	1's
1	0	1	1	0	1	1	1

2. 算数单元

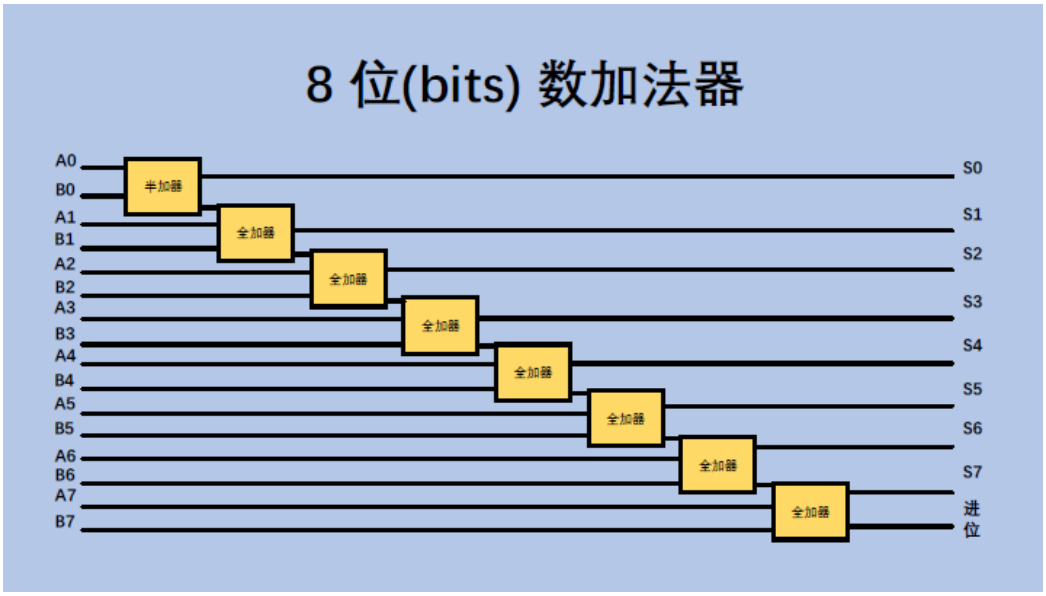
算数单元，负责计算机里的所有数字操作，比如四则运算。

CPU中有一个核心组件:ALU，就是通过集中“门电路”构成的。通过“门电路”构建出半加器（针对2bit相加=》和，是否进位）和全加器（针对3bit相加=》和，是否进位）





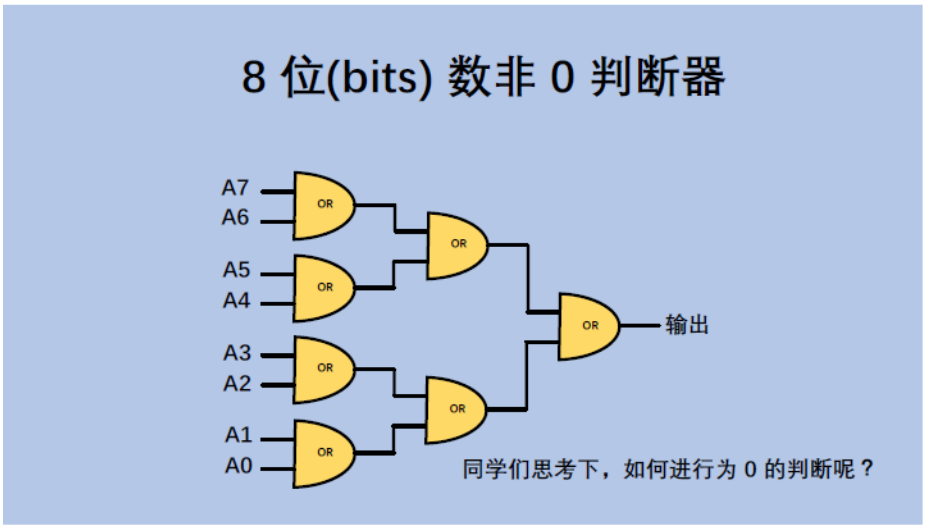
结合半加器和全加器，就能构建（针对8bits的数字进行相加的））的加法器（adder）。



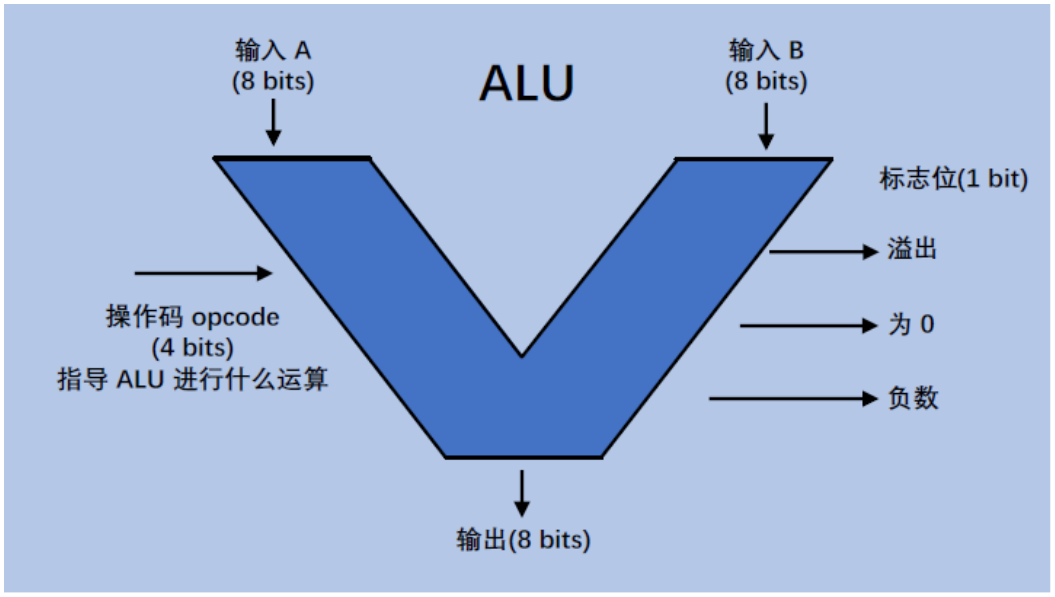
CPU =》 ALU =》 加法器 + 逻辑判断

3. 逻辑单元

逻辑单元主要用来进行逻辑操作，最基本的操作就是 与、或、非操作，但不只是一位(bit)数的比较。



4. ALU符号



3、寄存器和内存

1. 内存：平时说的内存
2. 外存：平时说的硬盘、软盘、光盘・・・・・・
3. 比较：
    1. 大小：内存比较小，外存比较大
    2. 访问速度：内存访问速度快，外存访问速度慢
    3. 价格：内存贵，外存便宜
    4. 持久化：内存没有持久化，外存数据是持久化的
    5. 随机访问：内存支持随机访问，随心所欲访问内存中任意地址的数据

外存支持随机访问，但是代价很大，外存擅长顺序访问

4.计算机存储数据，不仅仅是内存和外存能存储，CPU上也能存储数据（寄存器）。CPU的寄存器，存储空间更小，价格更高，访问速度更快。

4、控制单元

5、指令

所谓指令，即指导 CPU 进行工作的命令，主要有操作码 + 被操作数组成。

其中操作码用来表示要做什么动作，被操作数是本条指令要操作的数据，可能是内存地址，也可能是寄存器编号等。

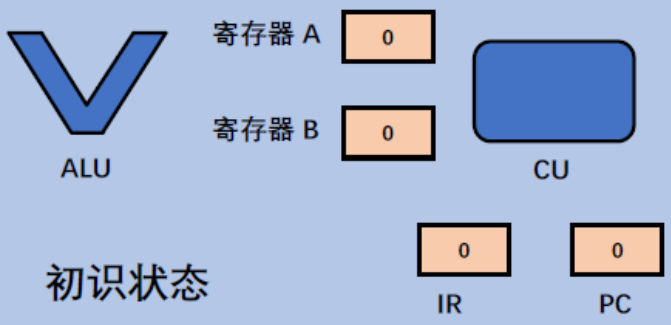
指令本身也是一个数字，用二进制形式保存在内存的某个区域中。

指令表(Instruction Table)

指令(instruction)	功能说明	4位 opcode	操作的地址或者寄存器
LOAD_A	从 RAM 的指定地址，将数据加载到 A 寄存器	0010	4 位 RAM 地址
LOAD_B	从 RAM 的指定地址，将数据加载到 B 寄存器	0001	4 位 RAM 地址
STORE_A	将数据从 A 寄存器写入 RAM 的指定地址	0100	4 位 RAM 地址
ADD	计算两个指定寄存器的数据的和，并将结果放入第二个寄存器	1000	2 位的寄存器 ID 2 位的寄存器 ID

6、CPU的基本工作流程

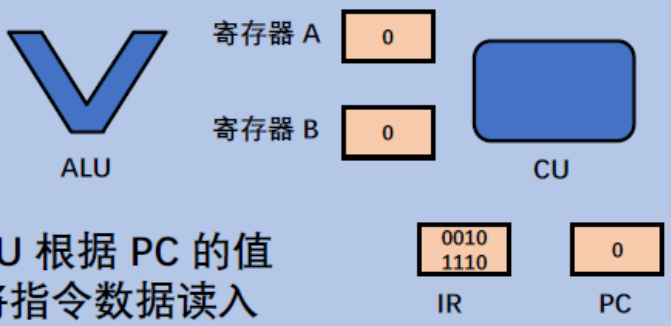
CU 和 ALU 的配合



初识状态

地址	数据
0	00101110
1	00011111
2	10000100
3	01001101
4	00000000
5	00000000
6	00000000
7	00000000
8	00000000
9	00000000
10	00000000
11	00000000
12	00000000
13	00000000
14	00000011
15	00001110

CU 和 ALU 的配合



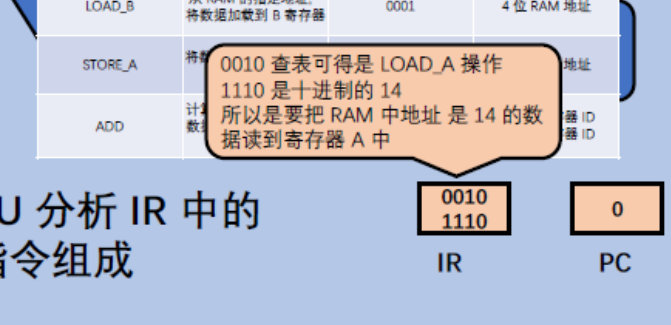
CU 根据 PC 的值
将指令数据读入
IR 中

地址	数据
0	00101110
1	00011111
2	10000100
3	01001101
4	00000000
5	00000000
6	00000000
7	00000000
8	00000000
9	00000000
10	00000000
11	00000000
12	00000000
13	00000000
14	00000011
15	00001110

CU 和 ALU 的配合

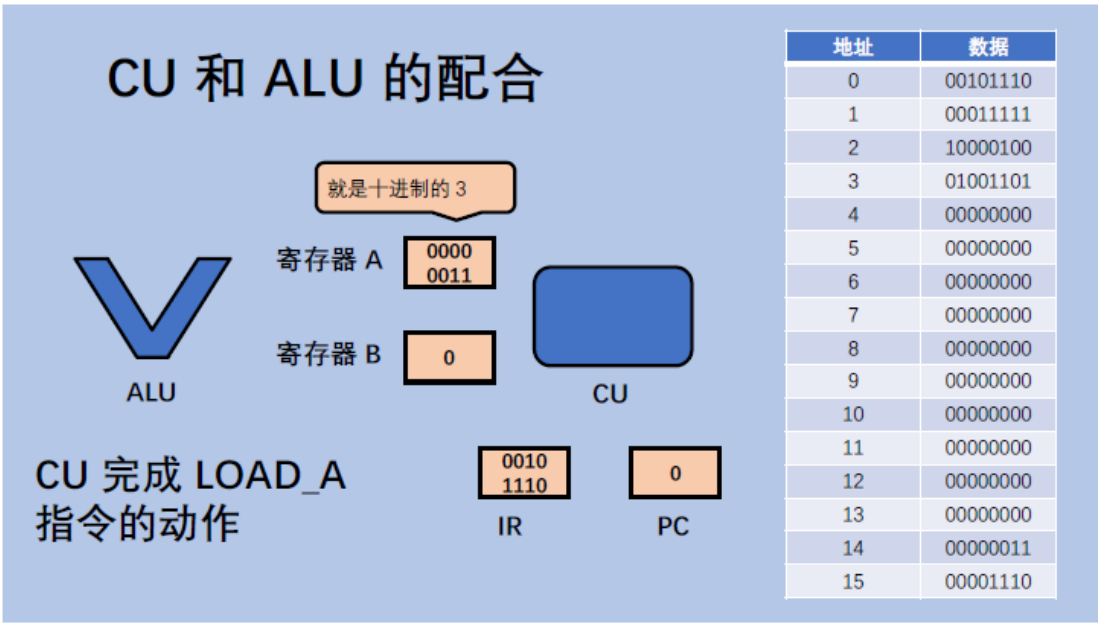
指令(instruction)	功能说明	4位 opcode	操作的地址或者寄存器
LOAD_A	从 RAM 的指定地址，将数据加载到 A 寄存器	0010	4 位 RAM 地址
LOAD_B	从 RAM 的指定地址，将数据加载到 B 寄存器	0001	4 位 RAM 地址
STORE_A	将		地址
ADD	计		器 ID
	数		器 ID

0010 查表可得是 LOAD_A 操作
1110 是十进制的 14
所以是要把 RAM 中地址 是 14 的数据读到寄存器 A 中

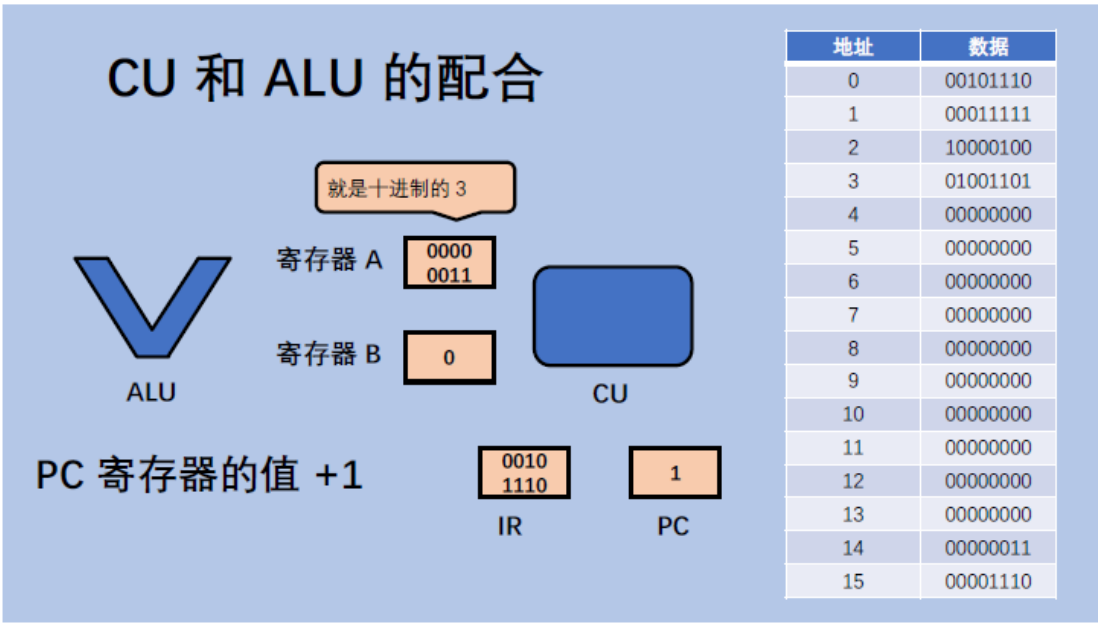


CU 分析 IR 中的
指令组成

地址	数据
0	00101110
1	00011111
2	10000100
3	01001101
4	00000000
5	00000000
6	00000000
7	00000000
8	00000000
9	00000000
10	00000000
11	00000000
12	00000000
13	00000000
14	00000011
15	00001110



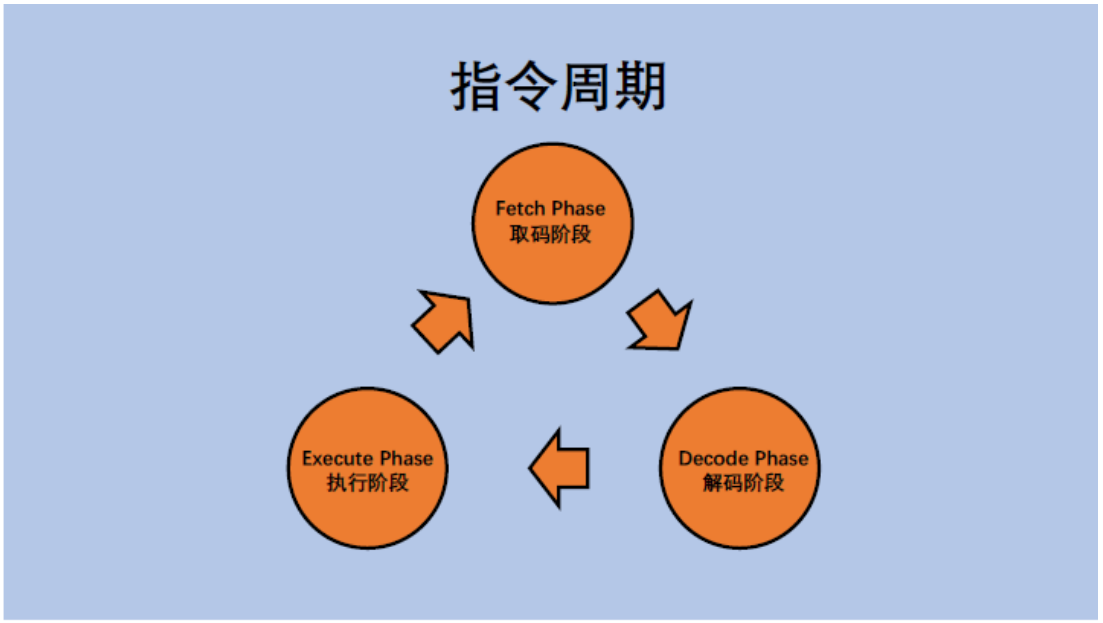
地址	数据
0	00101110
1	00011111
2	10000100
3	01001101
4	00000000
5	00000000
6	00000000
7	00000000
8	00000000
9	00000000
10	00000000
11	00000000
12	00000000
13	00000000
14	00000011
15	00001110



地址	数据
0	00101110
1	00011111
2	10000100
3	01001101
4	00000000
5	00000000
6	00000000
7	00000000
8	00000000
9	00000000
10	00000000
11	00000000
12	00000000
13	00000000
14	00000011
15	00001110

第一条指令的运行，其实没有用到我们之前制作的 ALU 部件，但这只是其中一些指令而已，大家尝试把剩余的 3 条指令自行运行一次，观察并理解这个过程。

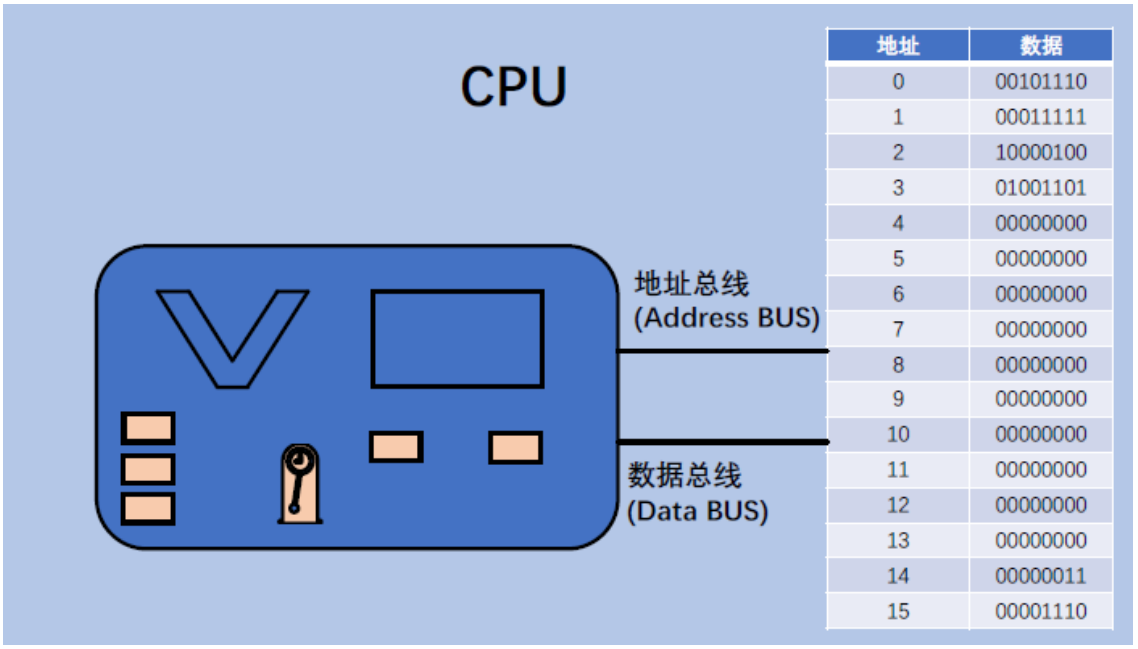
我们来总结下执行周期经过哪些阶段：



当然，电子计算机中的 CPU 可不像我们刚才那样，靠自己来驱动这个周期的运转，而是靠背后一个时钟来进行周期驱动的。

知乎问题：时钟频率是什么概念

最后，ALU + CU + 寄存器 + 时钟就组成了我们平时经常看到的一个词汇：中央处理器（Center ProcessUnit）简称 CPU。



地址	数据
0	00101110
1	00011111
2	10000100
3	01001101
4	00000000
5	00000000
6	00000000
7	00000000
8	00000000
9	00000000
10	00000000
11	00000000
12	00000000
13	00000000
14	00000011
15	00001110

7、总结

1. CPU 中的 PC 寄存器，是决定 CPU 要执行哪条指令的关键；
2. 指令是由 动作 + 操作对象组成
3. CPU 眼中只有指令，没有其他的概念