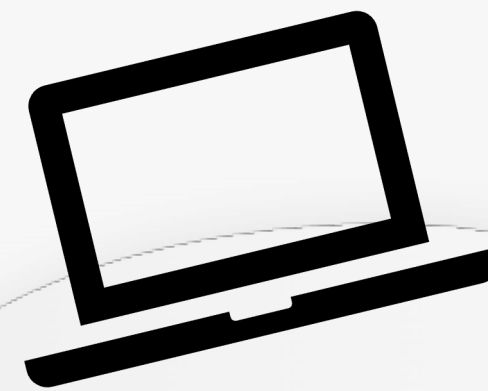


船说-计算机基础：



CPU执行程序

CPU设计与结构



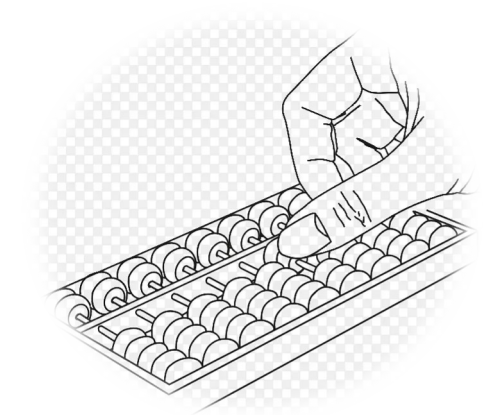


1.CPU设计与结构



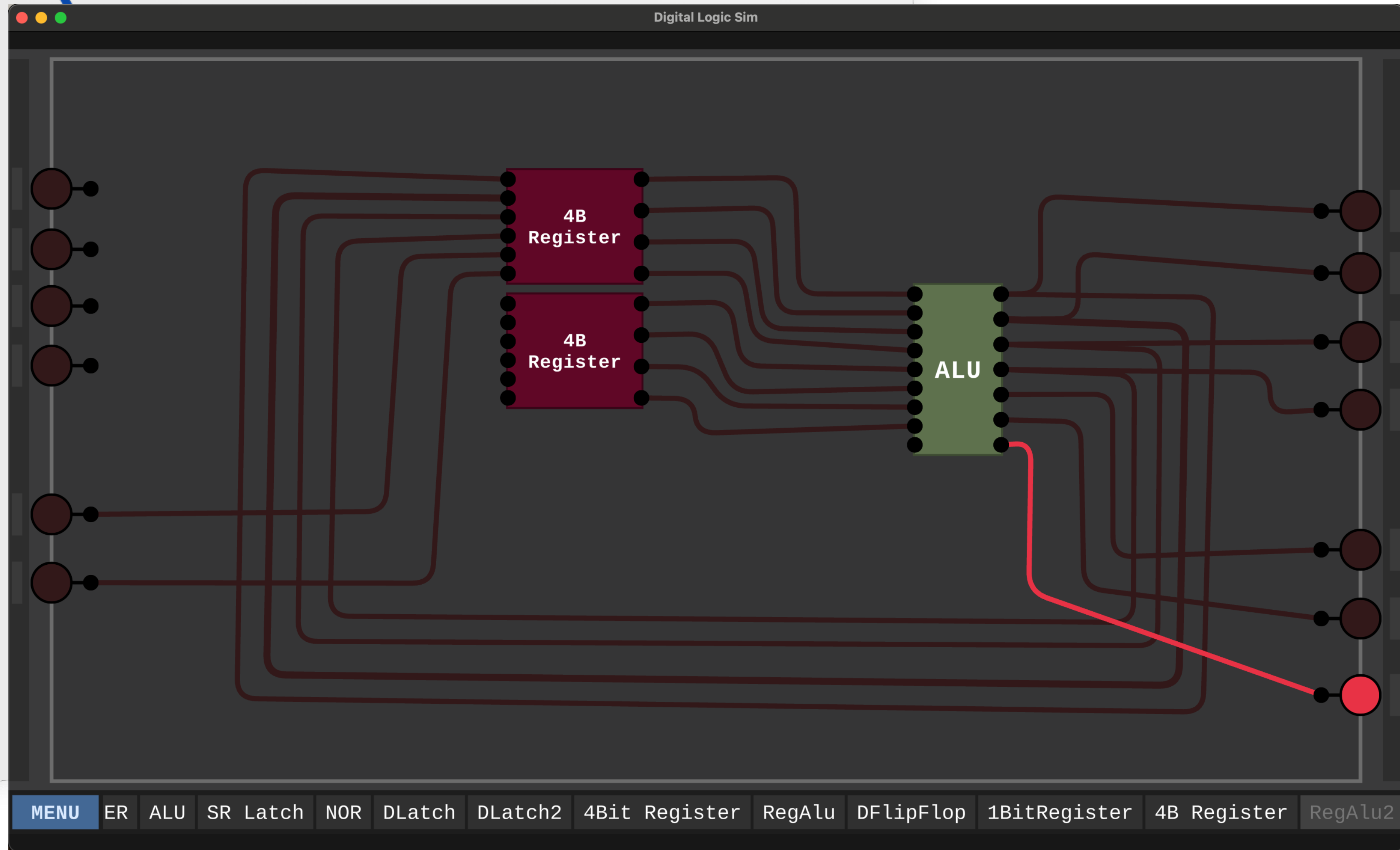
1.5 CPU如何执行程序

- 01 运算数据来源于编译生成的可执行机器码
- 02 CPU执行指令的过程
- 03 汇编指令初识





运算数据哪里来？





运算数据哪里来？

```
1  /* Type your code here, or load an example
2  int add() {
3      int x = 2;
4      int y = 3;
5      return x + y;
6  }
7
8  int main()
9  {
10     int a;
11     a = add();
12 }
```

```
song@df 1.cpu_structure % gcc -o test 1_1_add.c
```

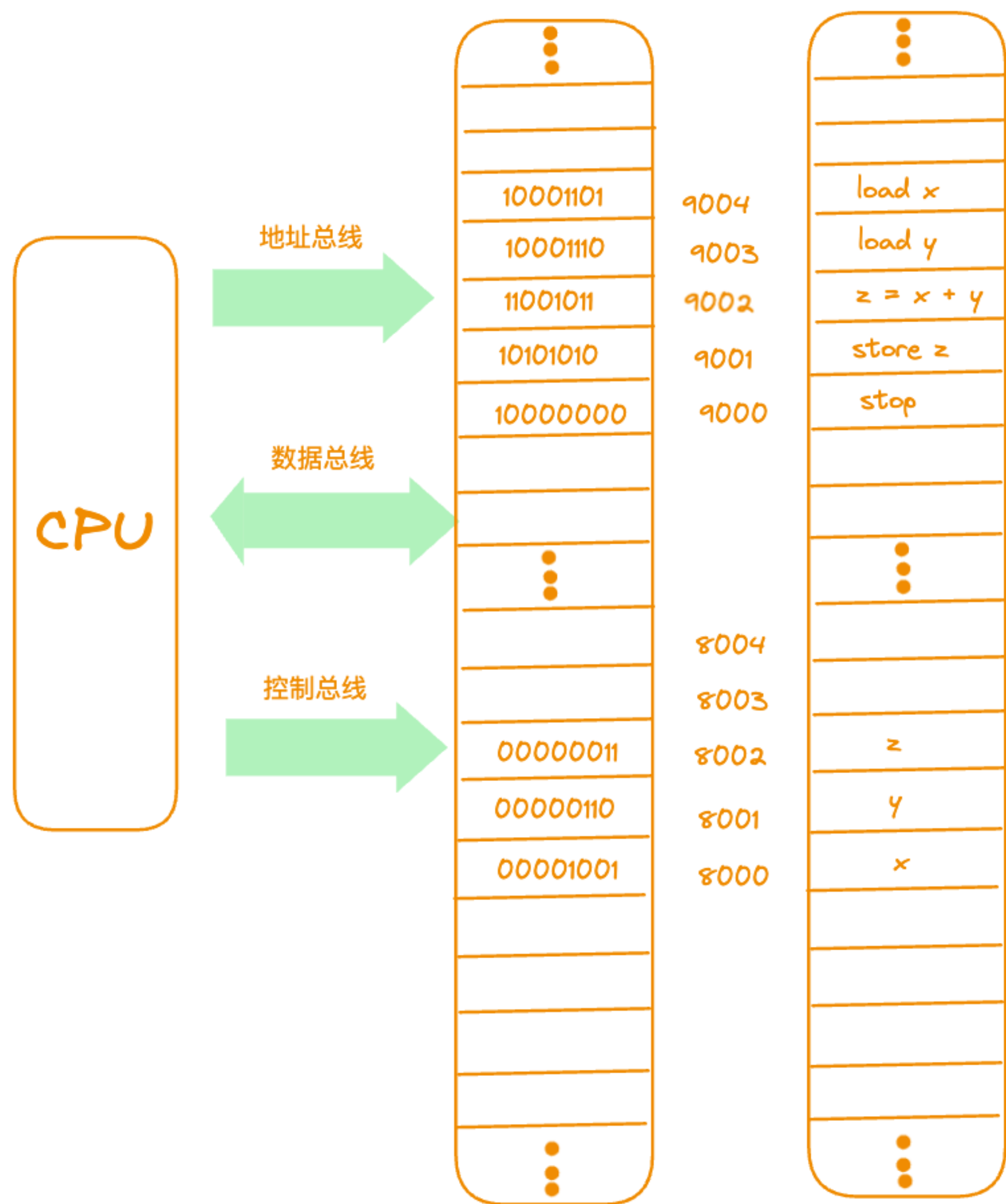
```
song@df 1.cpu_structure % hexdump test
```

00000000	facf	feed	000c	0100	0000	0000	0002	0000
00000010	0010	0000	02e8	0000	0085	0020	0000	0000
00000020	0019	0000	0048	0000	5f5f	4150	4547	455a
00000030	4f52	0000	0000	0000	0000	0000	0000	0000
00000040	0000	0000	0001	0000	0000	0000	0000	0000
00000050	0000	0000	0000	0000	0000	0000	0000	0000
00000060	0000	0000	0000	0000	0019	0000	00e8	0000
00000070	5f5f	4554	5458	0000	0000	0000	0000	0000
00000080	0000	0000	0001	0000	4000	0000	0000	0000
00000090	0000	0000	0000	0000	4000	0000	0000	0000
00000a00	0005	0000	0005	0000	0002	0000	0000	0000
00000b00	5f5f	6574	7478	0000	0000	0000	0000	0000
00000c00	5f5f	4554	5458	0000	0000	0000	0000	0000
00000d00	3f54	0000	0001	0000	004c	0000	0000	0000
00000e00	3f54	0000	0002	0000	0000	0000	0000	0000
00000f00	0400	8000	0000	0000	0000	0000	0000	0000
00001000	5f5f	6e75	6977	646e	695f	666e	006f	0000
00001100	5f5f	4554	5458	0000	0000	0000	0000	0000
00001200	3fa0	0000	0001	0000	0060	0000	0000	0000



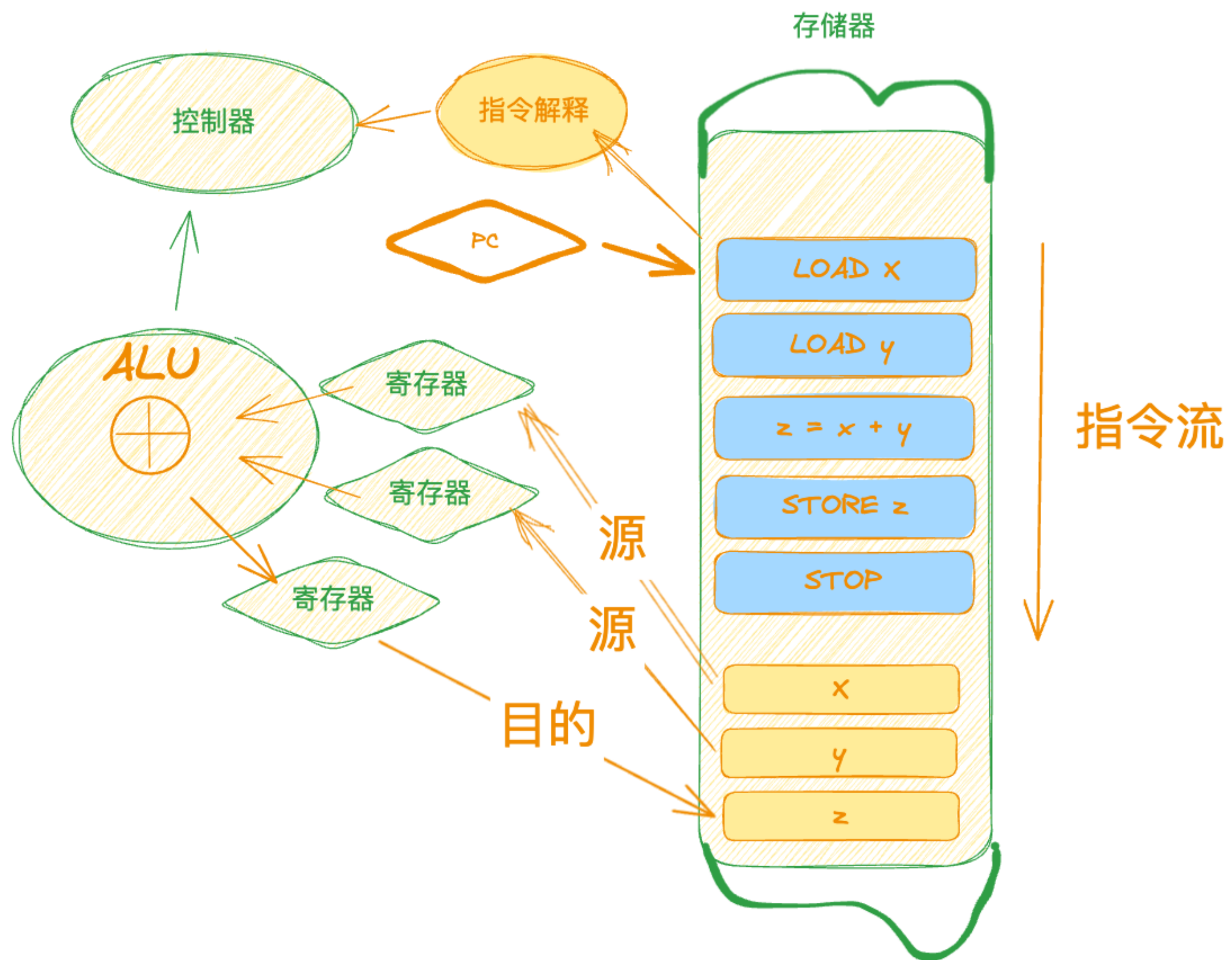


程序及其数据





运算数据来自存储器





指令格式



Operation Address1, Address2, Address3





Register Transfer Language(RTL)

$$[8000] = 100$$

$$[8000] = [8000] + 1$$

解释下面表述的含义：

a. $[2000] = 5$

b. $[2000] \leftarrow 6$

c. $[2000] \leftarrow [6]$





三地址指令格式

Operation Address1, Address2, Address3

Address2和Address3是源数据地址，
Address1是目的数地址





三地址指令执行



ADD P , Q , R





三地址指令执行



ADD

P ,

Q ,

R

10	011	010	001
----	-----	-----	-----





三地址指令执行

ADD P , Q , R

10	011	010	001
----	-----	-----	-----

00 Load

01 Store

10 Add

11 Sub

000

001

010

011

123

710

833

R

Q

P

加法
器





两地址指令格式



Operation Address1, Address2

Address2是源数据地址，Address1是源数据地址也是目的数地址





单地址指令格式



Operation Address

常见的测试指令，跳转指令。





计算机的六大类指令



MOV A,B
LOAD A,B
STORE A,B
ADD A,B
TEST A
BEQ Z





处理分支的汇编指令



```
2
3  int main()
4  {
5      int x;
6      x = 3;
7
8      return 0;
9  }
10
```

```
1  main:
2      sub    sp, sp, #8
3      mov    r0, #0
4      str    r0, [sp, #4]
5      mov    r1, #3
6      str    r1, [sp]
7      add    sp, sp, #8
8      bx     lr
```

在线编译器：<https://godbolt.org/>

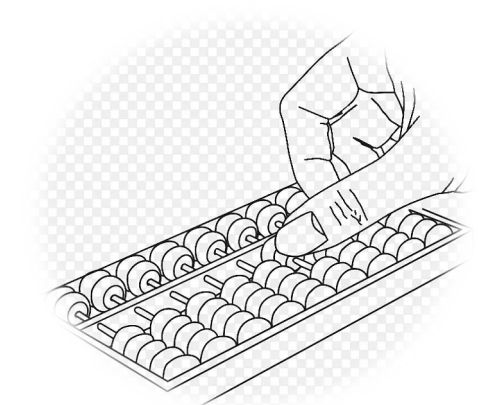




总结

1.5 CPU执行程序

1. CPU只能执行机器码
2. 高级语言程序需要编译生成机器码才能执行
3. CPU所有执行指令都来自存储器
4. 指令中包含操作和数据
5. 执行过各需要借助CPU内的寄存器
6. CPU内部的PC寄存器的特殊作用





欢迎参与学习

WELCOME FOR YOUR JOINING

嵌入式宋船长