**中国矿业大学计算机学院**

**课程实验报告**

课程名称: 计算机组成原理实验课

实验题目: 实验三 CPU与模型机实验

实验时间: 2021年1月10日

学生姓名: 李春阳

学 号: 10193657

专业班级: 信息安全2019-1班

任课教师 徐志鸥

评语与成绩:

**实验三** **CPU与简单模型机设计实验**

**一、实验目的**

（1）掌握一个简单CPU的组成原理。

（2）在掌握部件单元电路的基础上，进一步将其构造一台基本模型计算机。

（3）为其定义五条机器指令，编写相应的微程序并上机调试掌握整机概念。

1. **实验设备**

PC机一台，TD-CMA实验系统一套。

1. **CPU的组成及其工作原理**

1、CPU的组成及其工作原理（描述）

本实验要实现一个简单的CPU，并且在此CPU的基础上，继续构建一个简单的模型计算机。

CPU由运算器（ALU）、微程序控制器（MC）、通用寄存器（R0），指令寄存器（IR）、程序计数器（PC）和地址寄存器（AR）组成,如图5-1-1所示。这个CPU在写入相应的微指令后，就具备了执行机器指令的功能，但是机器指令一般存放在主存当中，CPU必须和主存挂接后，才有实际的意义，所以还需要在该 CPU 的基础上增加一个主存和基本的输入输出部件，以构成一个简单的模型计算机。

2、CPU的电路原理及数据通路图

A

B

ALU

R0

AR

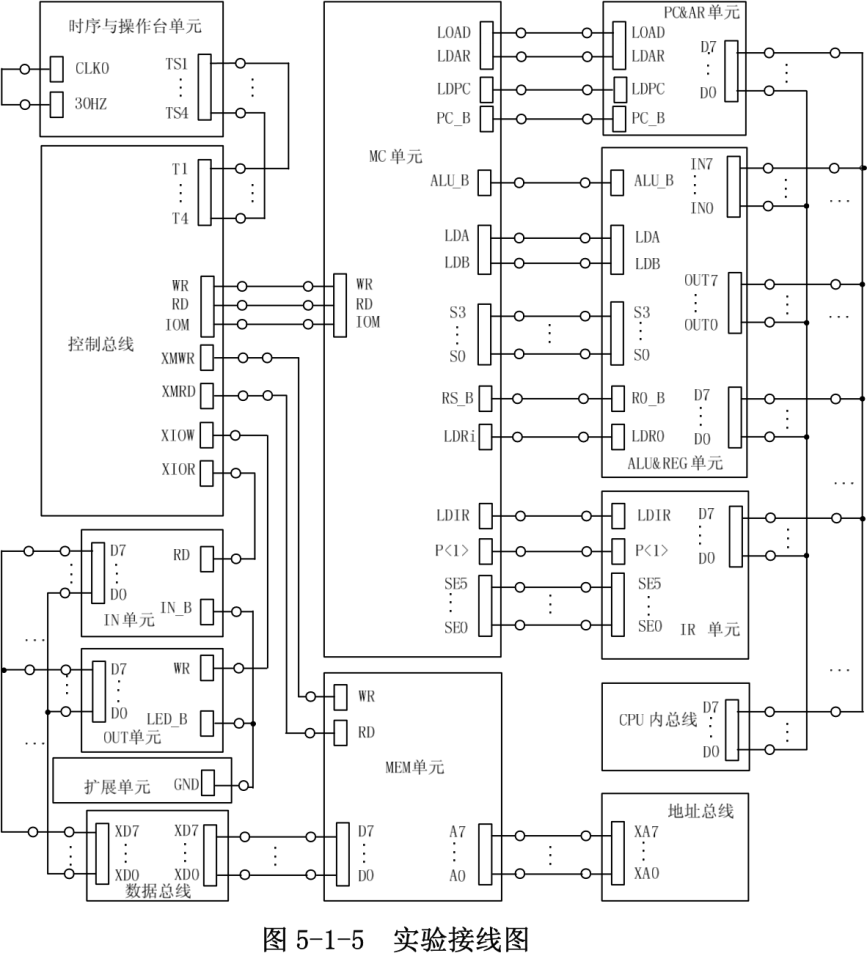
PC

IR

控制器

图5-1-1 基本CPU构成原理图

按图5-1-5连接实验线路。



除了程序计数器（PC），其余部件在前面的实验中都已用到，在此不再讨论。系统的程序计数器（PC）和地址寄存器（AR）集成在一片FPGA芯片中 。CLR连接至CON单元的总清端CLR，按下CLR按钮，将使PC清零，LDPC和T3相与后作为计数器的计数时钟，当LOAD为低时，计数时钟到来后将CPU内总线上的数据打入PC。

D7...D0

计数器L

CPU内总线

B7...B0

三态门

B7...B0

Q7...Q0

LDPC

T3

PC\_B

CPU内总线

B7...B0

...

...

CLR

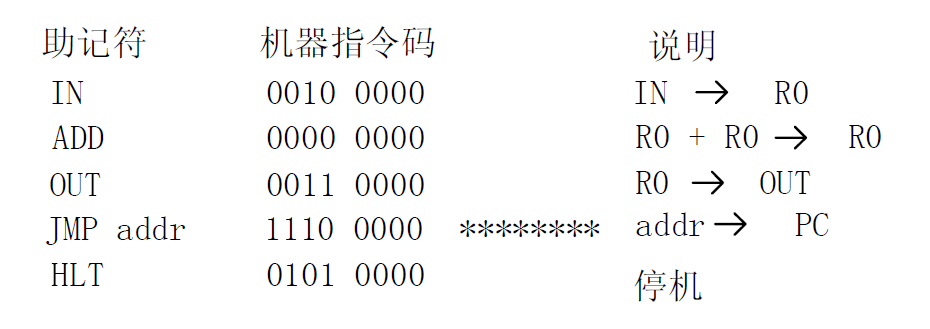
LOAD

...

图5-1-2 程序计数器(PC)原理图

3、解释涉及到的控制信号

本模型机和前面微程序控制器实验相比，新增加一条跳转指令JMP，共有五条指令：IN（输入）、ADD（二进制加法）、OUT（输出）、JMP（无条件转移），HLT（停机），其指令格式如下（高４位为操作码）：



其中JMP为双字节指令，其余均为单字节指令，\*\*\*\*\*\*\*\*为addr对应的二进制地址码。微程序控制器实验的指令是通过手动给出的，现在要求 CPU 自动从存储器读取指令并执行。根据以上要求，设计数据通路图，如图5-1-3所示。

本实验在前一个实验的基础上增加了三个部件，一是 PC（程序计数器），另一个是 AR（地址寄存器），还有就是 MEM（主存）。因而在微指令中应增加相应的控制位，其微指令格式如表5-1-1所示。

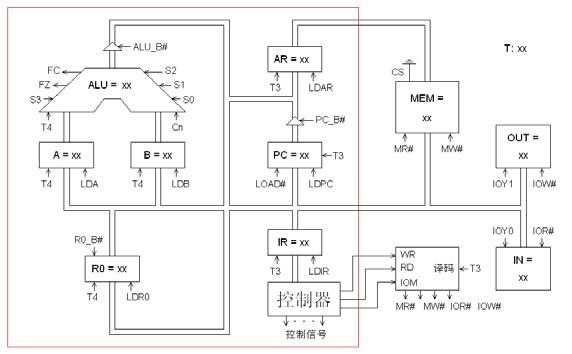
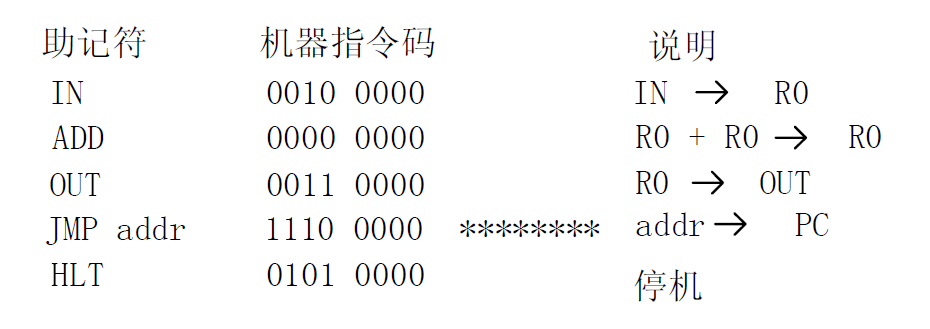


图5-1-3 数据通路图

4、指令与指令格式

本模型机和前面微程序控制器实验相比，新增加一条跳转指令JMP，共有五条指令：IN（输入）、ADD（二进制加法）、OUT（输出）、JMP（无条件转移），HLT（停机），其指令格式如下（高４位为操作码）：



其中JMP为双字节指令，其余均为单字节指令，\*\*\*\*\*\*\*\*为addr对应的二进制地址码。微程序控制器实验的指令是通过手动给出的，现在要求 CPU 自动从存储器读取指令并执行。

表5-1-1 微指令格式

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 22 | 21 | 20 | 19 | 18-15 | 14-12 | 11-9 | 8-6 | 5-0 |
| M23 | M22 | WR | RD | IOM | S3-S0 | A字段 | B字段 | C字段 | MA5-MA0 |

A字段 B字段 C字段

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 14 | 13 | 12 | 选择 |  | 11 | 10 | 9 | 选择 |  | 8 | 7 | 6 | 选择 |
| 0 | 0 | 0 | NOP | 0 | 0 | 0 | NOP | 0 | 0 | 0 | NOP |
| 0 | 0 | 1 | LDA | 0 | 0 | 1 | ALU\_B | 0 | 0 | 1 | P<1> |
| 0 | 1 | 0 | LDB | 0 | 1 | 0 | R0\_B | 0 | 1 | 0 | 保留 |
| 0 | 1 | 1 | LDR0 | 0 | 1 | 1 | 保留 | 0 | 1 | 1 | 保留 |
| 1 | 0 | 0 | 保留 | 1 | 0 | 0 | 保留 | 1 | 0 | 0 | 保留 |
| 1 | 0 | 1 | LOAD | 1 | 0 | 1 | 保留 | 1 | 0 | 1 | LDPC |
| 1 | 1 | 0 | LDAR | 1 | 1 | 0 | PC\_B | 1 | 1 | 0 | 保留 |
| 1 | 1 | 1 | LDIR | 1 | 1 | 1 | 保留 | 1 | 1 | 1 | 保留 |

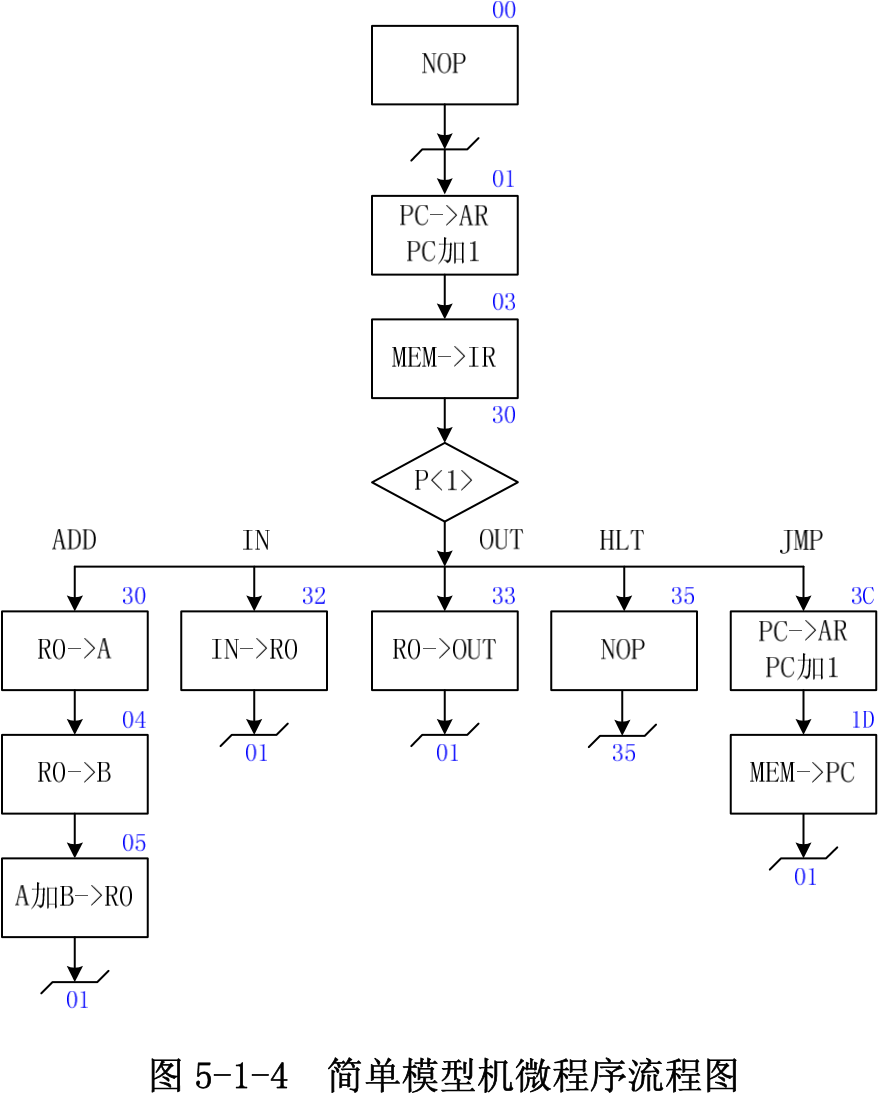
设计一段机器程序，要求从IN单元读入一个数据，存于R0，将R0和自身相加，结果存于 R0，再将R0的值送OUT单元显示。

根据要求可以得到如下程序，地址和内容均为二进制数。



5、画出相关指令得周期流程图

系统涉及到的微程序流程见图 5-1-4 所示，当拟定“取指”微指令时，该微指令的判别测试字段为P<1>测试。指令译码原理见图3-2-3所示，由于“取指”微指令是所有微程序都使用的公用微指令，因此P<1> 的测试结果出现多路分支。本机用指令寄存器的高6位（IR7—IR2）作为测试条件，出现５路分支，占用５个固定微地址单元，剩下的其它地方就可以一条微指令占用控存一个微地址单元随意填写，微程序流程图上的单元地址为16进制。



6、理解指令、微指令、程序、微程序

指令是计算机执行某种操作的命令。

微指令是一个单位时间内出现的一组微操作的描述语句

程序是一系列[机器指令](https://www.baidu.com/s?wd=%E6%9C%BA%E5%99%A8%E6%8C%87%E4%BB%A4&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYuWmkP1nLrjwBmvNbmvP90ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHTdrjnsPjTs)的有序集合，用于解决实际问题，有子程序、分支、循环等结构。

微程序是微指令的有序集合，用于实现机器指令的功能。。

**四、分析与比较与实验一、实验二的不同。**

实验三是从主存中一条一条取出机器指令去译码并执行，与前两个实验相比，本实验是构造一个简单的模型机而不是在模拟一个个简单的模块，不再注重部分部件的情况，而是从总体上看。

**五、实验结果与问题分析**

CPU由运算器（ALU）、微程序控制器（MC）、通用寄存器（R0），指令寄存器（IR）、程序计数器（PC）和地址寄存器（AR）组成,这个CPU在写入相应的微指令后，就具备了执行机器指令的功能，但是机器指令一般存放在主存当中，CPU必须和主存挂接后，才有实际的意义，所以还需要在该CPU的基础上增加一个主存和基本的输入输出部件，以构成一个简单的模型计算机。

**六、实验体会**

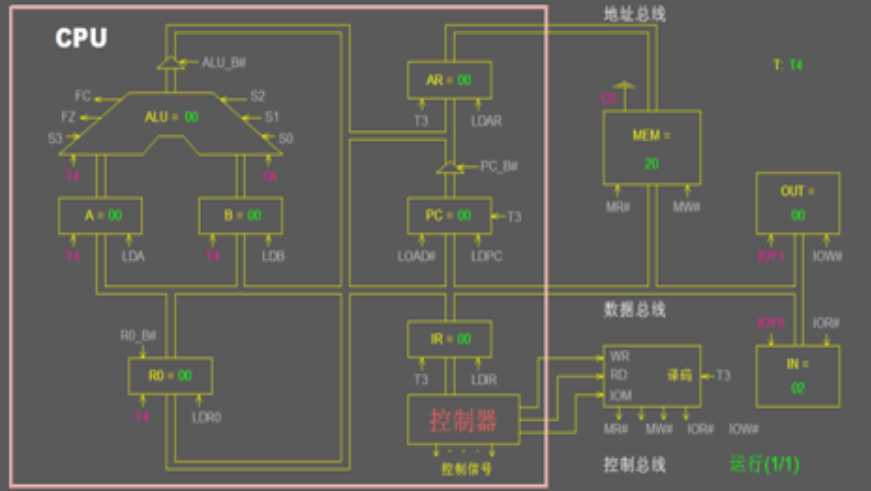
本实验有如何构建一个简单 CPU 的全部过程，并且在此 CPU 的基础上，继续构建一个简单的模型计算机。 CPU 由运算器、微程序控制器、通用寄存器，指令寄存器、程序计数器和地址寄存器组成。在 CPU 的基础上增加一个主存和基本的输入输出部件，就构成一个简单的模型计算机。 在掌握部件单元电路的基础上，进一步将其构造一台基本模型计算机。

**七、思考题**

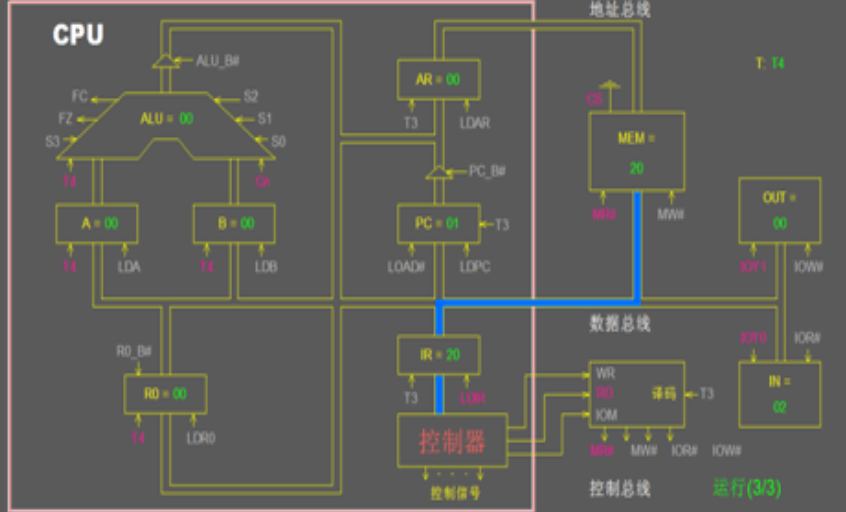
根据实验模型画出数据通路图，并解释每个部分作用和工作流程。

答：

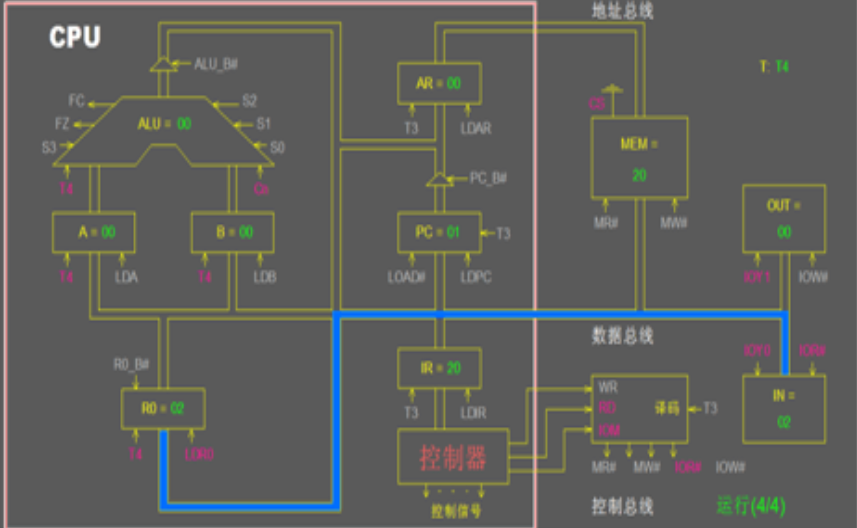
（1）从IN单元读入数据送至AR。首先，AR地址为00，计数器PC加1，机器指令代码送至MEM单元，MEM=20。



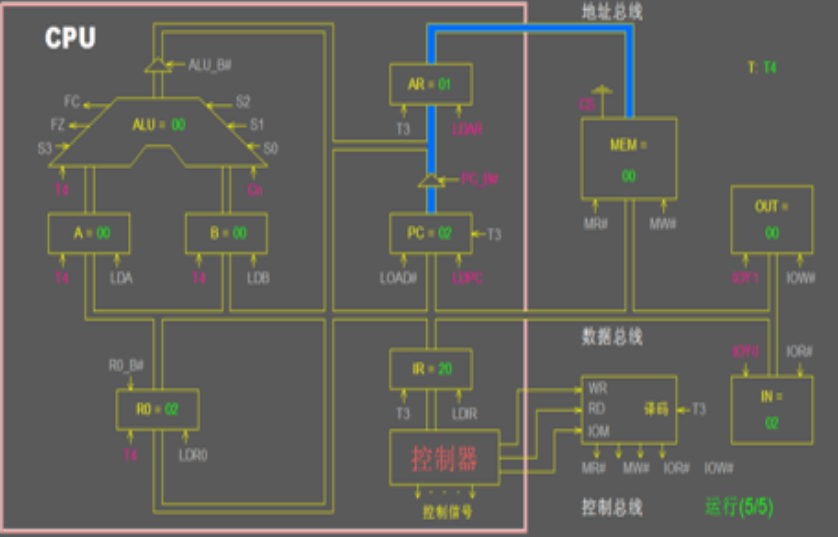
（2）MEM中的机器指令代码送至IR单元，再到控制器，进行译码操作。



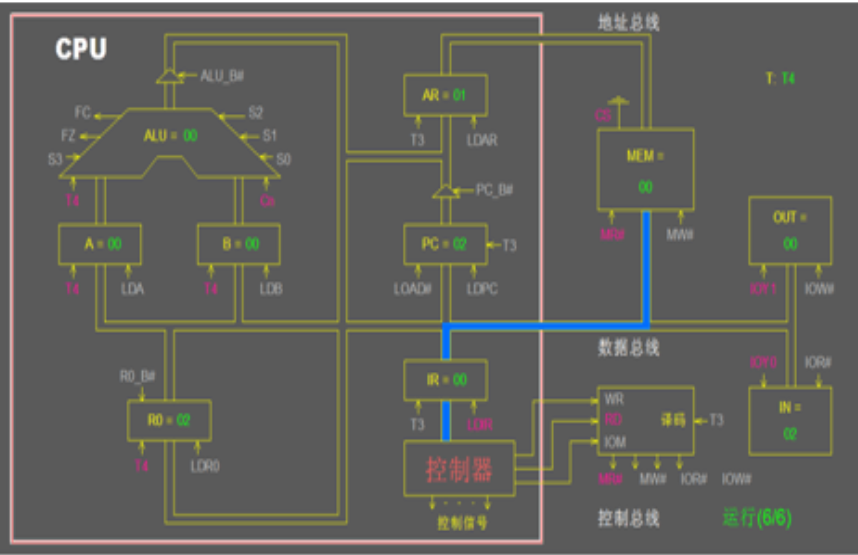
（3）RO寄存器读入IN单元输入的数据。



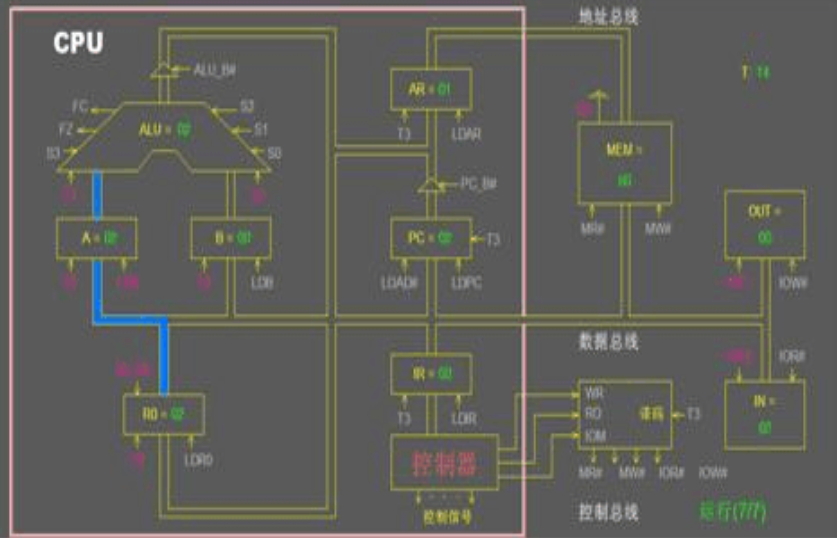
（4）AR地址为01，计数器PC加1，机器指令代码送至MEM单元，MEM=00

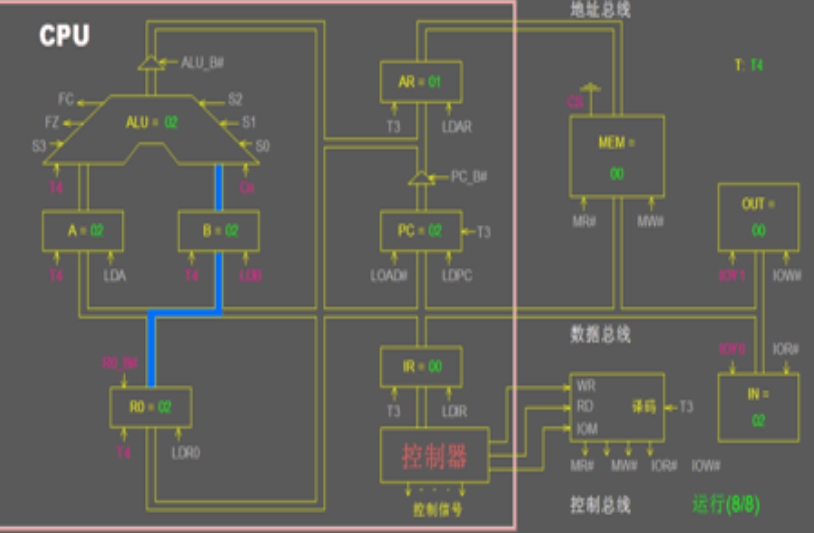


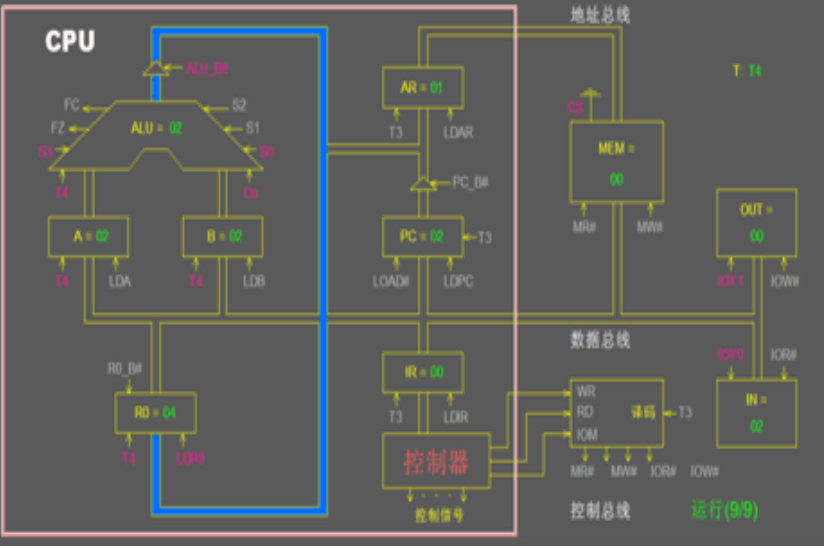
（5）MEM中的机器指令代码送至IR单元，再到控制器，进行译码操作。



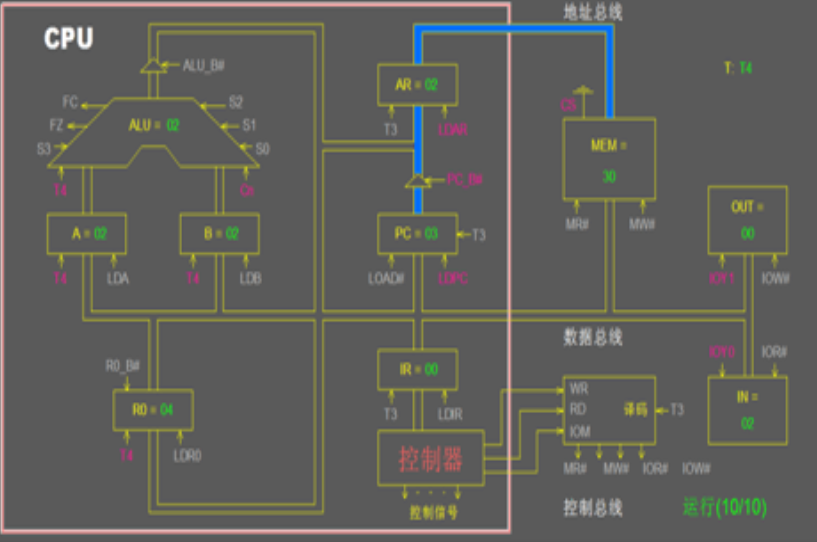
（6）将读入的数据送到A暂存器，再送到B暂存器。最后将RO和自身相加，结果送到RO。

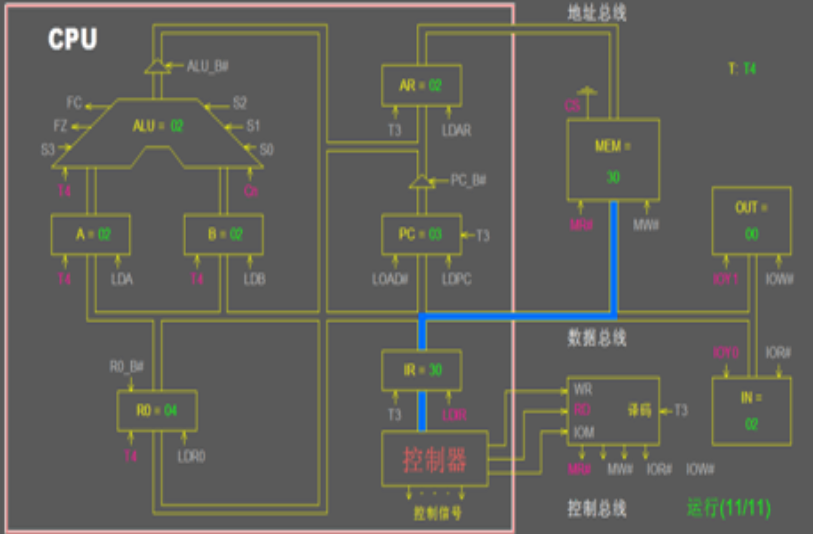


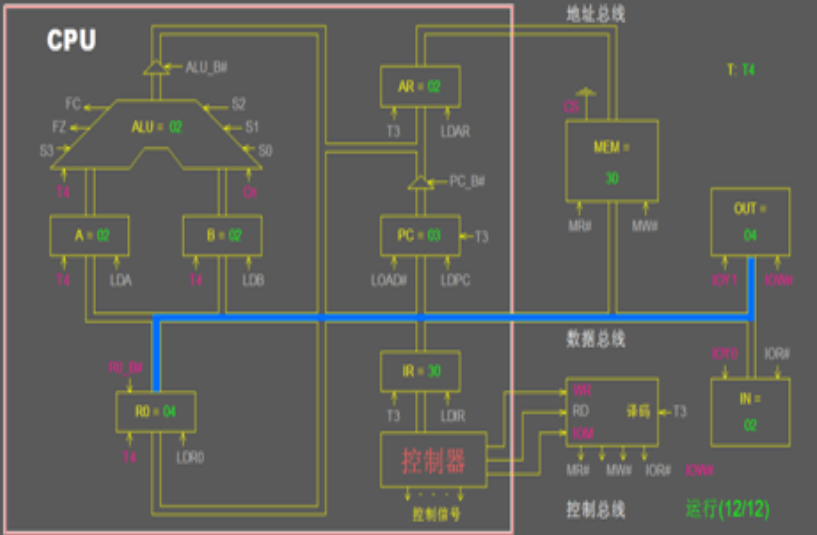




（7）进行计数译码操作，将RO的值送至OUT单元，输出。







（8）再进行计数译码，跳转到00，返回第一条机器指令

