

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。

一、实验设备

PC 机一台, TD-CMA 实验系统一套。

二、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 (RO), 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

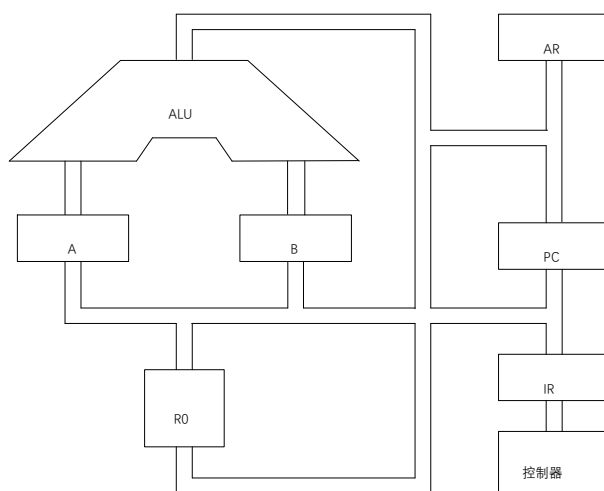


图 5-1-1 基本 CPU 构成原理图

按图 5-1-5 连接实验线路。

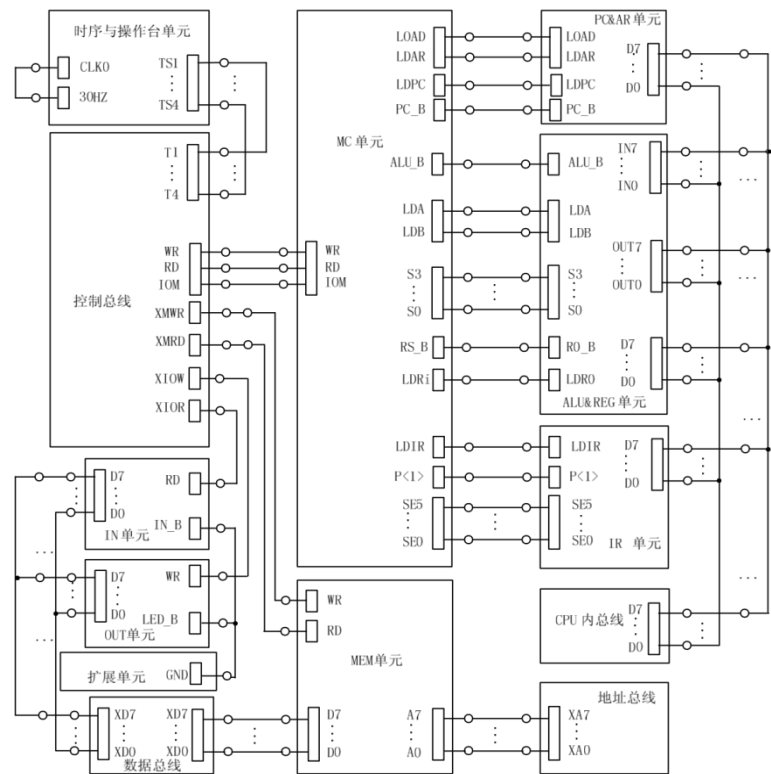


图 5-1-5 实验接线图

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。

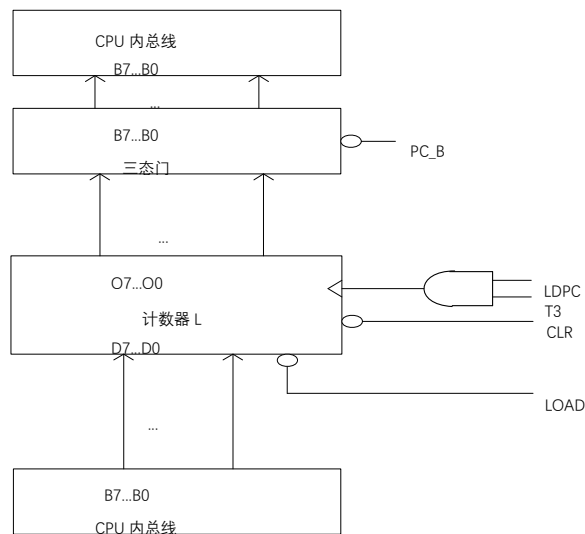


图 5-1-2 程序计数器(PC)原理图

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）：

助记符	机器指令码	说明
IN	0010 0000	IN → R0
ADD	0000 0000	R0 + R0 → R0
OUT	0011 0000	R0 → OUT
JMP addr	1110 0000 *****	addr → PC
HLT	0101 0000	停机

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。

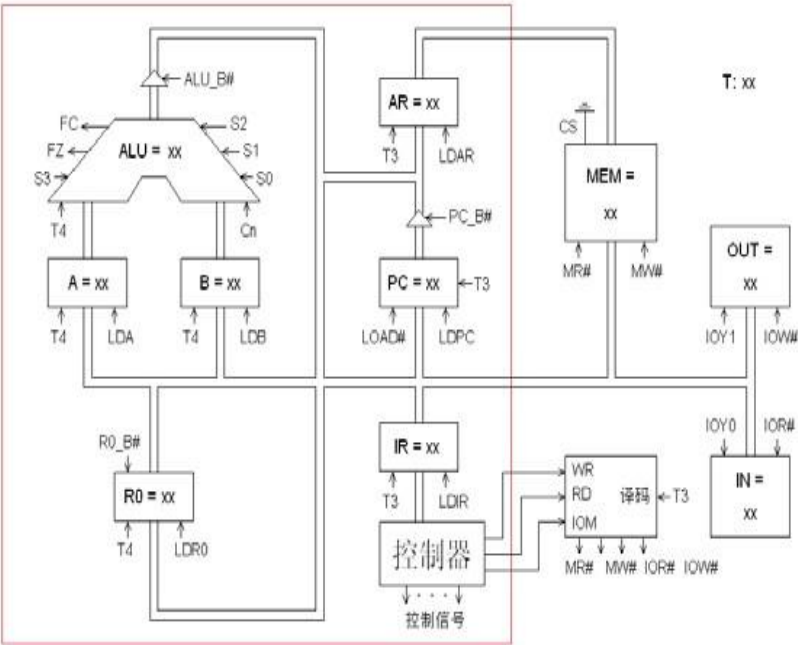


图 5-1-3 数据通路图

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）：

助记符	机器指令码	说明
IN	0010 0000	IN → R0
ADD	0000 0000	R0 + R0 → R0
OUT	0011 0000	R0 → OUT
JMP addr	1110 0000 *****	addr → PC
HLT	0101 0000	停机

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	
0	0	0	NOP		0	0	0	NOP	
0	0	1	LDA		0	0	1	ALU_B	
0	1	0	LDB		0	1	0	R0_B	
0	1	1	LDR0		0	1	1	保留	
1	0	0	保留		1	0	0	保留	
1	0	1	LOAD		1	0	1	保留	
1	1	0	LDAR		1	1	0	PC_B	
1	1	1	LDIR		1	1	1	保留	
8	7	6	选择		1	0	0	保留	
0	0	0	NOP		1	0	1	LDPC	
0	0	1	P<1>		1	1	0	保留	
0	1	0	保留		1	1	1	保留	
0	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。

地 址	内 容	助记符	说 明
00000000	00100000 ;	START: IN R0	从 IN 单元读入数据送 R0
00000001	00000000 ;	ADD R0, R0	R0 和自身相加，结果送 R0
00000010	00110000 ;	OUT R0	R0 的值送 OUT 单元显示
00000011	11100000 ;	JMP START	跳转至 00H 地址
00000100	00000000 ;		
00000101	01010000 ;	HLT	停机

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。

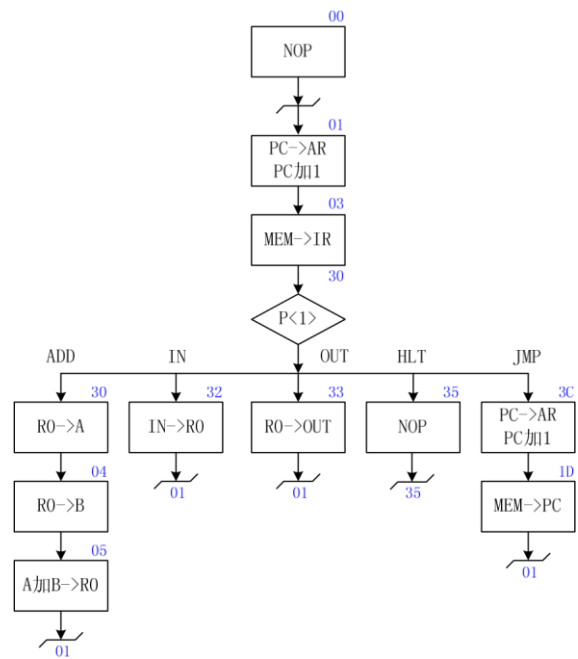


图 5-1-4 简单模型机微程序流程图

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（RO），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

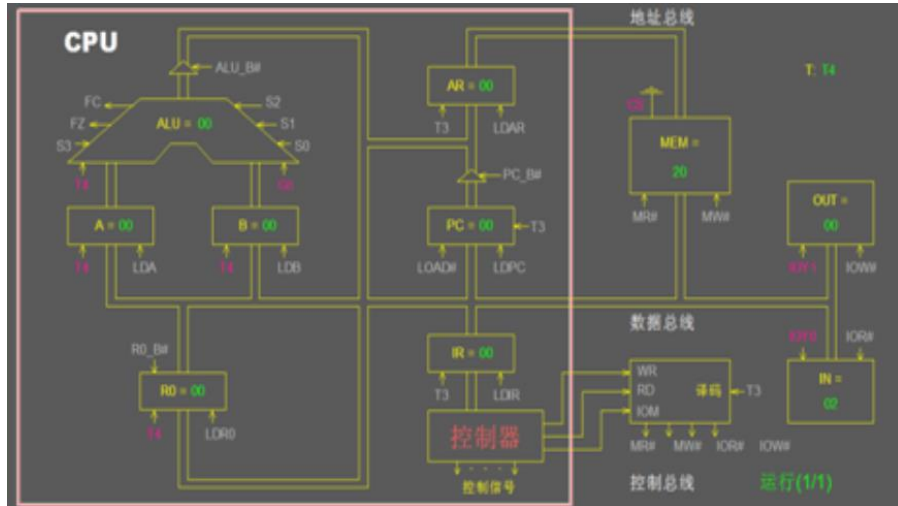
本实验有如何构建一个简单 CPU 的全部过程，并且在此 CPU 的基础上，继续构建一个简单的模型计算机。CPU 由运算器、微程序控制器、通用寄存器，指令寄存器、程序计数器和地址寄存器组成。在 CPU 的基础上增加一个主存和基本的输入输出部件，就构成一个简单的模型计算机。在掌握部件单元电路的基础上，进一步将其构造一台基本模型计算机。

七、思考题

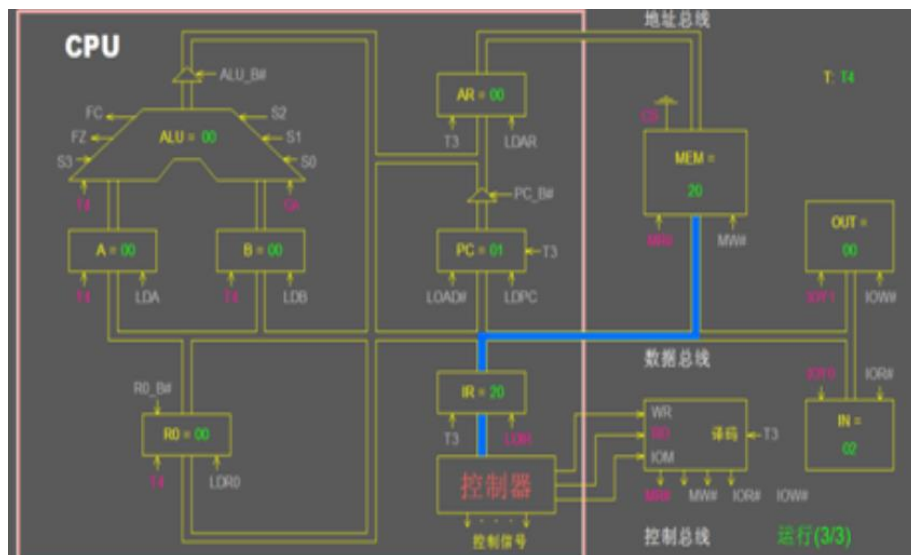
根据实验模型画出数据通路图，并解释每个部分作用和工作流程。

答：

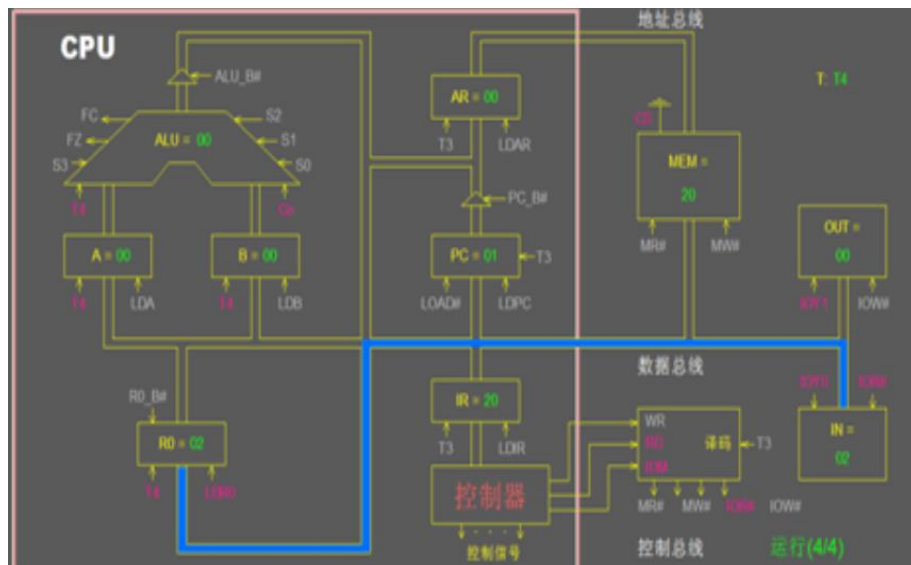
（1）从 IN 单元读入数据送至 AR。首先，AR 地址为 00，计数器 PC 加 1，机器指令代码送至 MEM 单元，MEM=20。



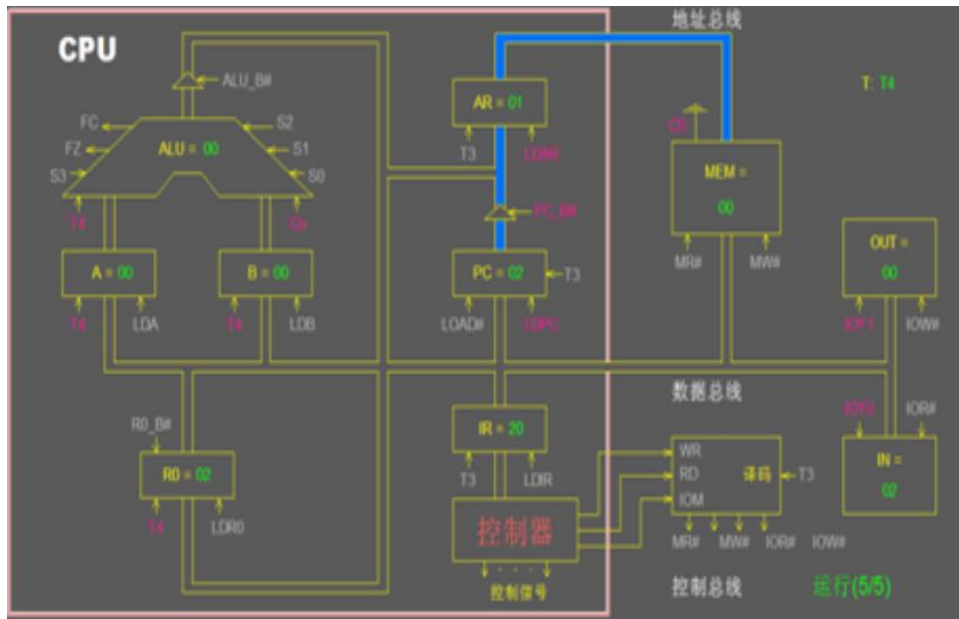
(2) MEM 中的机器指令代码送至 IR 单元，再到控制器，进行译码操作。



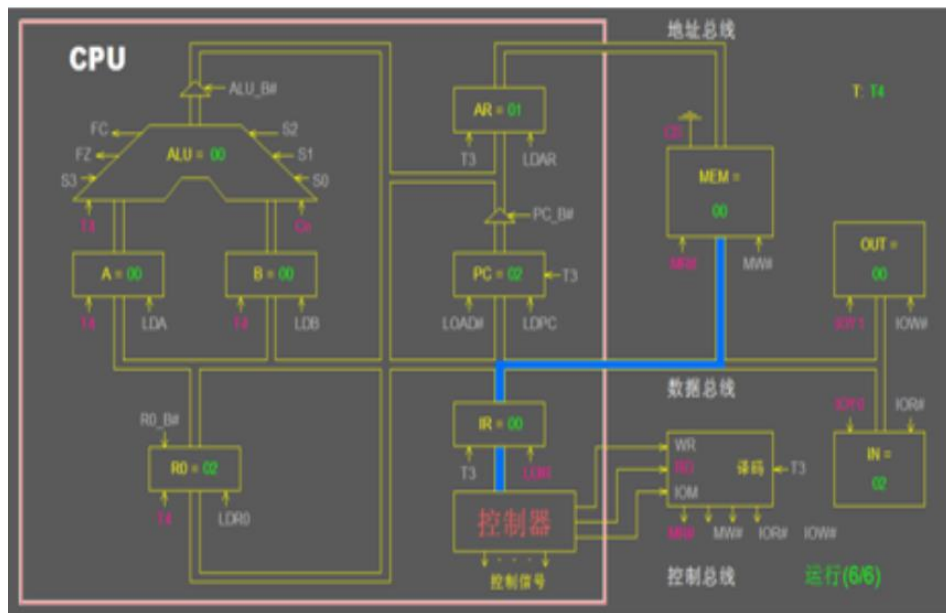
(3) RO 寄存器读入 IN 单元输入的数据。



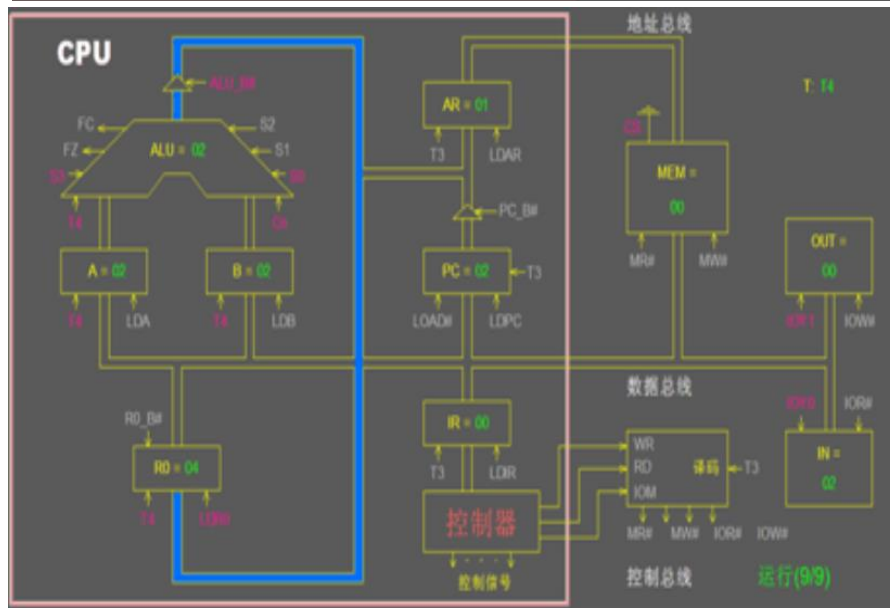
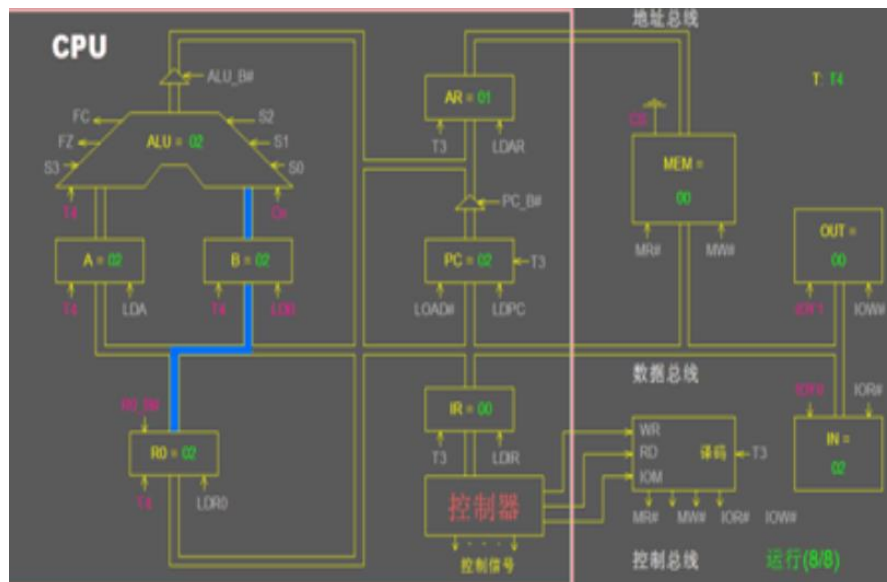
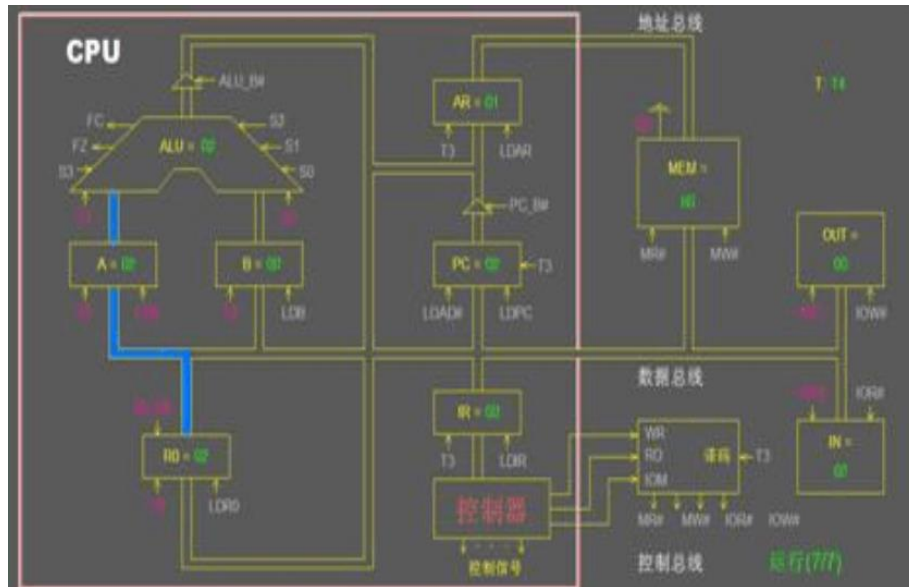
(4) AR 地址为 01，计数器 PC 加 1，机器指令代码送至 MEM 单元，MEM=00



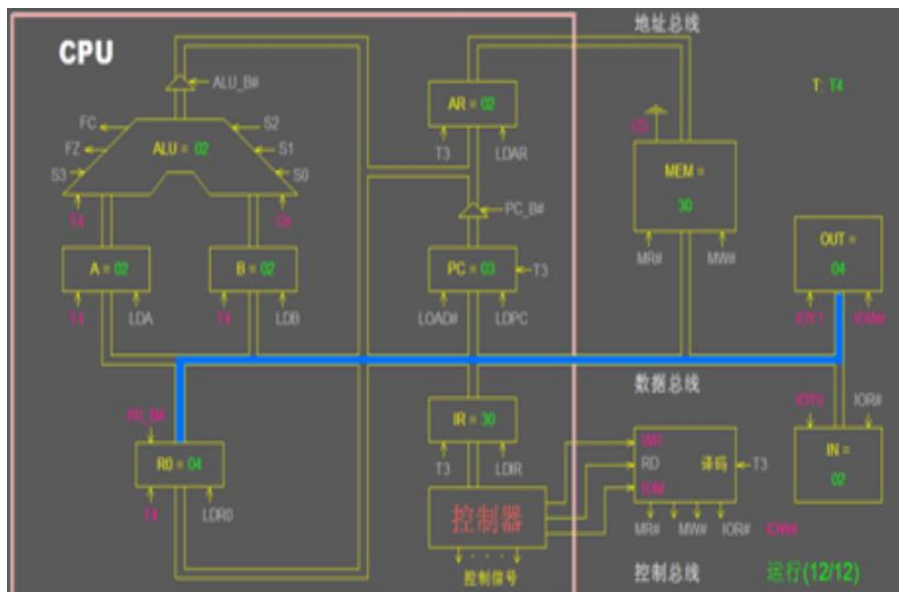
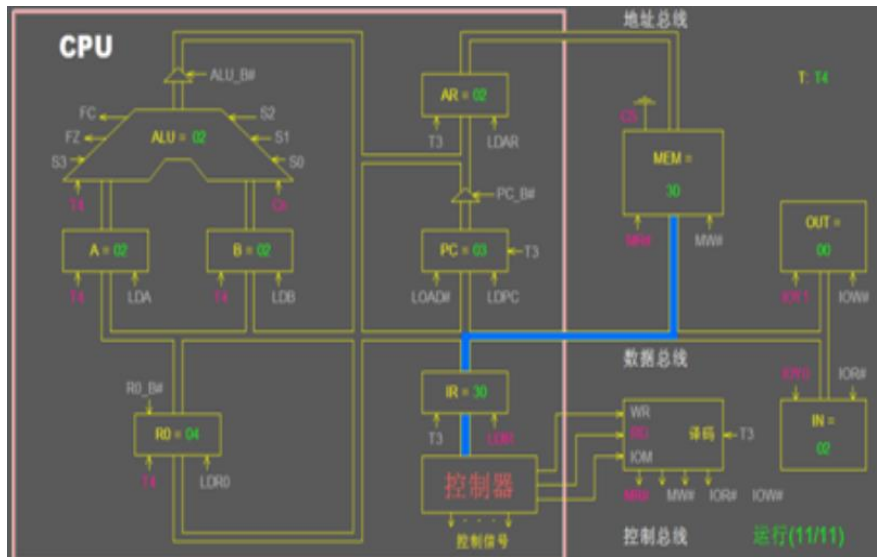
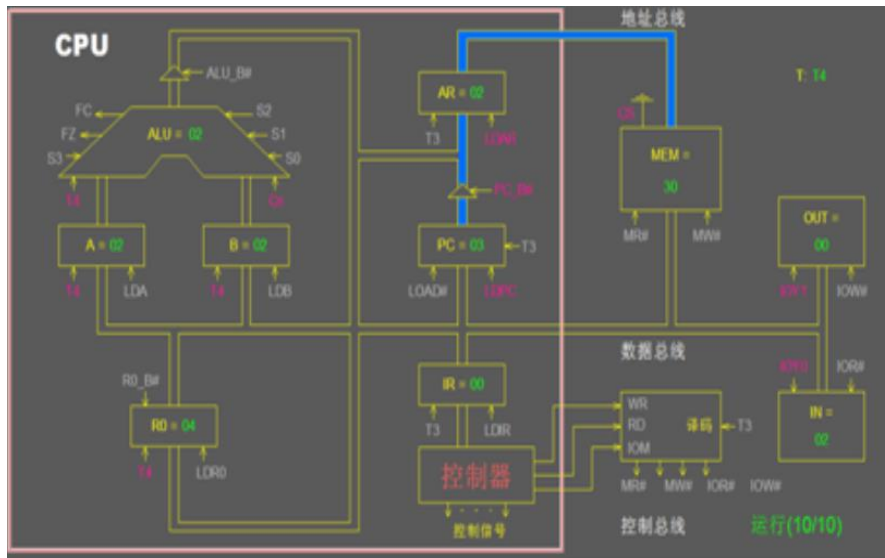
(5) MEM 中的机器指令代码送至 IR 单元，再到控制器，进行译码操作。



(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 R0 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

