

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: 实验一 基本运算器实验

实验时间: 2020 年 12 月 20 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:

一、实验目的

- ## 二、实验设备

三、运算器的组成原理

运算器内部含有三个独立运算部件，分别为算术、逻辑和移位运算部件，要处理的数据存于暂存器 A 和暂存器 B，三个部件同时接受来自 A 和 B 的数据（有些处理器体系结构把移位运算器放于算术和逻辑运算部件之前，如 ARM），各部件对操作数进行何种运算由控制信号 S3…S0 和 CN 来决定，任何时候，多路选择开关只选择三部件中一个部件的结果作为 ALU 的输出。如果是影响进位的运算，还将置进位标志 FC，在运算结果输出前，置 ALU 零标志。ALU 中所有模块集成在一片 FPGA 中。

本实验的原理如图 1-1-1 所示。



逻辑运算部件由逻辑门构成，较为简单，而后面又有专门的算术运算部件设计实验，在此对这两个部件不再赘述。移位运算采用的是桶形移位器，一般采用交叉开关矩阵来实现，交叉开关的原理如图 1-1-2 所示。图中显示的是一个 4X4 的矩阵（系统中是一个 8X8 的矩阵）。每一个输入都通过开关与一个输出相连，把沿对角线的开关导通，就可实现移位功能，即：

(1) 对于逻辑左移或逻辑右移功能，将一条对角线的开关导通，这将所有的输入位与所使用的输出分别相连，而没有同任何输入相连的则输出连接 0。

(2) 对于循环右移功能，右移对角线同互补的左移对角线一起激活。例如，在 4 位矩阵中使用‘右 1’和‘左 3’对角线来实现右循环 1 位。

(3) 对于未连接的输出位，移位时使用符号扩展或是 0 填充，具体由相应的指令控制。使用另外的逻辑进行移位总量译码和符号判别。

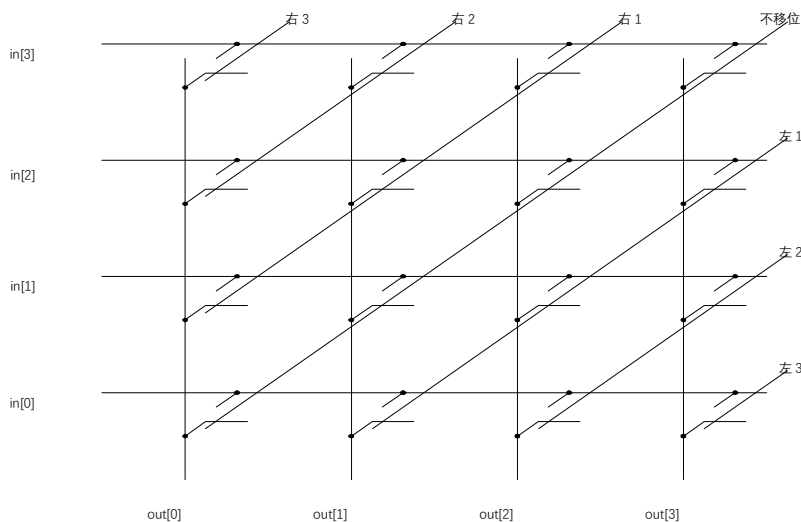


图 1-1-2 交叉开关桶形移位器原理图

运算器部件由一片 FPGA 实现。ALU 的输入和输出通过三态门 74LS245 连到 CPU 内总线上，另外还有指示灯标明进位标志 FC 和零标志 FZ。请注意：实验箱上凡丝印标注有马蹄形标记 ‘ \sqcup ’，表示这两根排针之间是连通的。图中除 T4 和 CLR，其余信号均来自于 ALU 单元的排线座，实验箱中所有单元的 T1、T2、T3、T4 都连接至控制总线单元的 T1、T2、T3、T4，CLR 都连接至 CON 单元的 CLR 按钮。T4 由时序单元的 TS4 提供，其余控制信号均由 CON 单元的二进制数据开关模拟给出。控制信号中除 T4 为脉冲信号外，其余均为电平信号，其中 ALU_B 为低有效，其余为高有效。

暂存器 A 和暂存器 B 的数据能在 LED 灯上实时显示，原理如图 1-1-3 所示（以

A0 为例，其它相同)。进位标志 FC、零标志 FZ 和数据总线 D7…D0 的显示原理也是如此。

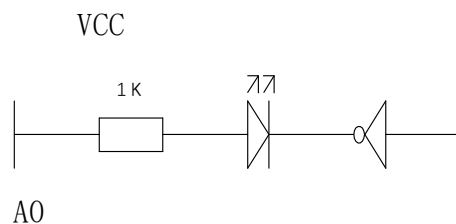


图 1-1-3 A0 显示原理图

ALU 和外围电路的连接如图 1-1-4 所示，图中的小方框代表排针座。

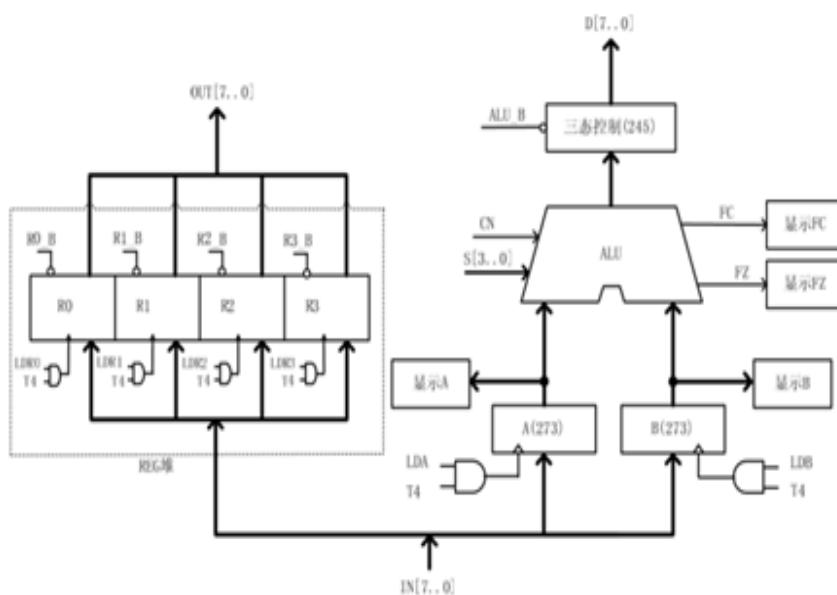


图 1-1-4 ALU 和外围电路连接原理图

按图 1-1-5 连接实验电路，

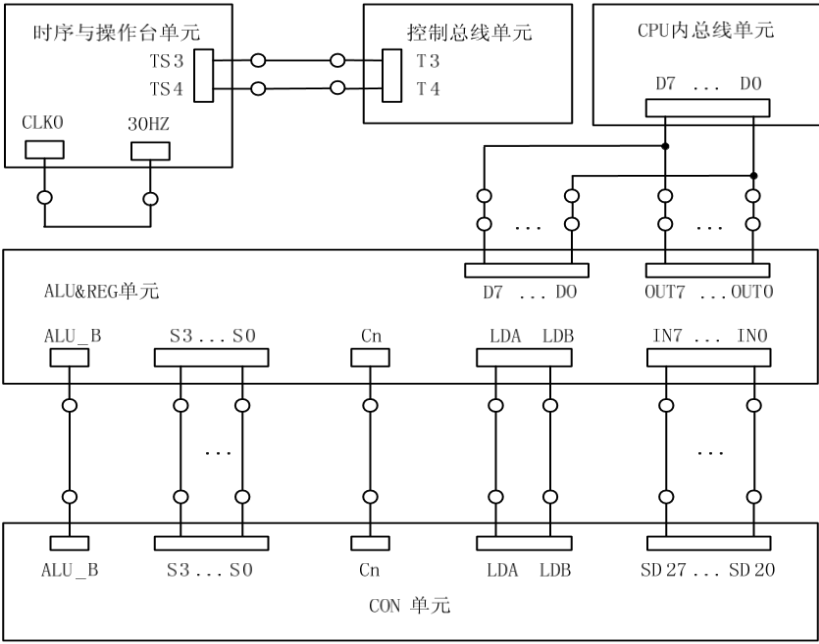


图 1-1-5 实验接线图

运算器实验的数据通路图，如图 1-1-6 所示。

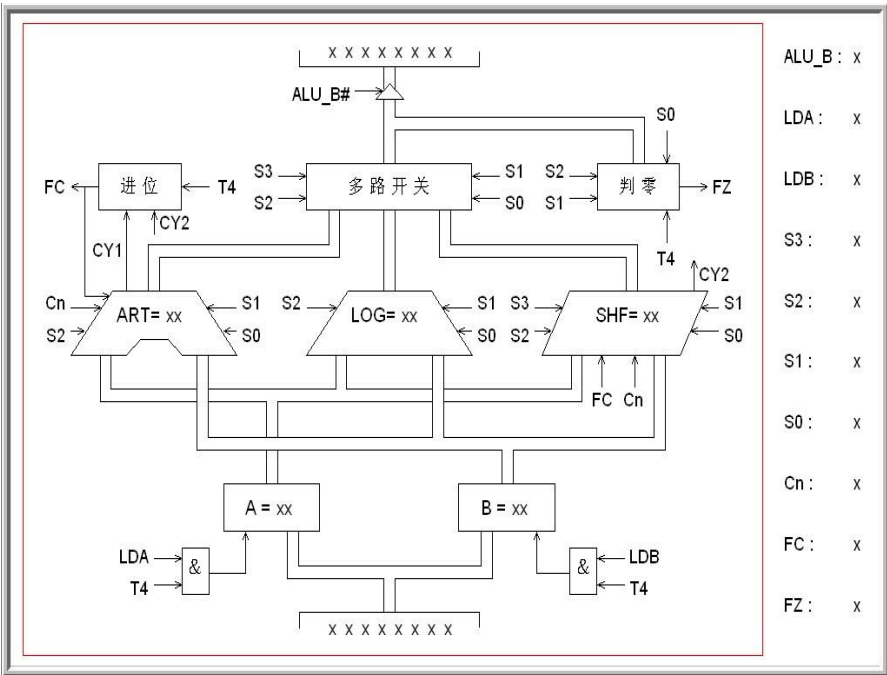


图 1-1-6

3、解释涉及到的控制信号

运算器的逻辑功能表如表 1-1-1 所示，其中 S3 S2 S1 S0 CN 为控制信号，FC 为进位标志，FZ 为运算器零标志，表中功能栏内的 FC、FZ 表示当前运算会影响到该标志。

表 1-1-1

运算类型	S3 S2 S1 S0	CN	功 能
逻辑运算	0000	X	F=A (直通)
	0001	X	F=B (直通)
	0010	X	F=AB (FZ)
	0011	X	F=A+B (FZ)
	0100	X	F=/A (FZ)
移位运算	0101	X	F=A 不带进位循环右移 B (取低 3 位) 位 (FZ)
	0110	0	F=A 逻辑右移一位 (FZ)
		1	F=A 带进位循环右移一位 (FC, FZ)
	0111	0	F=A 逻辑左移一位 (FZ)
		1	F=A 带进位循环左移一位 (FC, FZ)
算术运算	1000	X	置 FC=CN (FC)
	1001	X	F=A 加 B (FC, FZ)
	1010	X	F=A 加 B 加 FC (FC, FZ)
	1011	X	F=A 减 B (FC, FZ)
	1100	X	F=A 减 1 (FC, FZ)
	1101	X	F=A 加 1 (FC, FZ)
	1110	X	(保留)
	1111	X	(保留)

运算器的逻辑功能表如表 1-1-1 所示，其中 S3 S2 S1 S0 CN 为控制信号，FC 为进位标志，FZ 为运算器零标志，表中功能栏内的 FC、FZ 表示当前运算会影响到该标志。

*表中“X”为任意态。

四、运算器实验记录表与分析结论 1

1-1-2 运算结果表

运算类型	A	B	S3 S2 S1 S0	CN	结果
逻辑运算	65	A7	0 0 0 0	X	F=(65) FC=(0) FZ=(0)
	65	A7	0 0 0 1	X	F=(A7) FC=(0) FZ=(0)
	65	A7	0 0 1 0	X	F=(25) FC=(0) FZ=(0)
	65	A7	0 0 1 1	X	F=(E7) FC=(0) FZ=(0)
	65	A7	0 1 0 0	X	F=(9A) FC=(0) FZ=(0)
移位运算	65	A7	0 1 0 1	X	F=(CA) FC=(1) FZ=(0)
	65	A7	0 1 1 0		F=(32) FC=(1) FZ=(0)
				1	F=(B2) FC=(1) FZ=(0)
	65		0 1 1 1	0	F=(CA) FC=(1) FZ=(0)
				1	F=(CA) FC=(0) FZ=(0)
算术运算	65	A7	1 0 0 0	X	F=(65) FC=(0) FZ=(0)
	65	A7	1 0 0 1	X	F=(C) FC=(1) FZ=(0)
	65	A7	1 0 1 0 (FC=0)	X	F=(C) FC=(0) FZ=(0)
			1 0 1 0 (FC=1)	X	F=(D) FC=(1) FZ=(0)
	65	A7	1 0 1 1	X	F=(BE) FC=(1) FZ=(0)
	65	A7	1 1 0 0	X	F=(64) FC=(0) FZ=(0)
	65	A7	1 1 0 1	X	F=(66) FC=(0) FZ=(0)

五、实验体会

这是计算机组成原理的第一个实验，了该实验装置按功能分成几大区，何时操作各种开关、按键。通过实验掌握了运算器工作原理，熟悉算术和逻辑的运算过程以及控制这种运算的方法，了解了进位对算术与逻辑运算结果的影响。

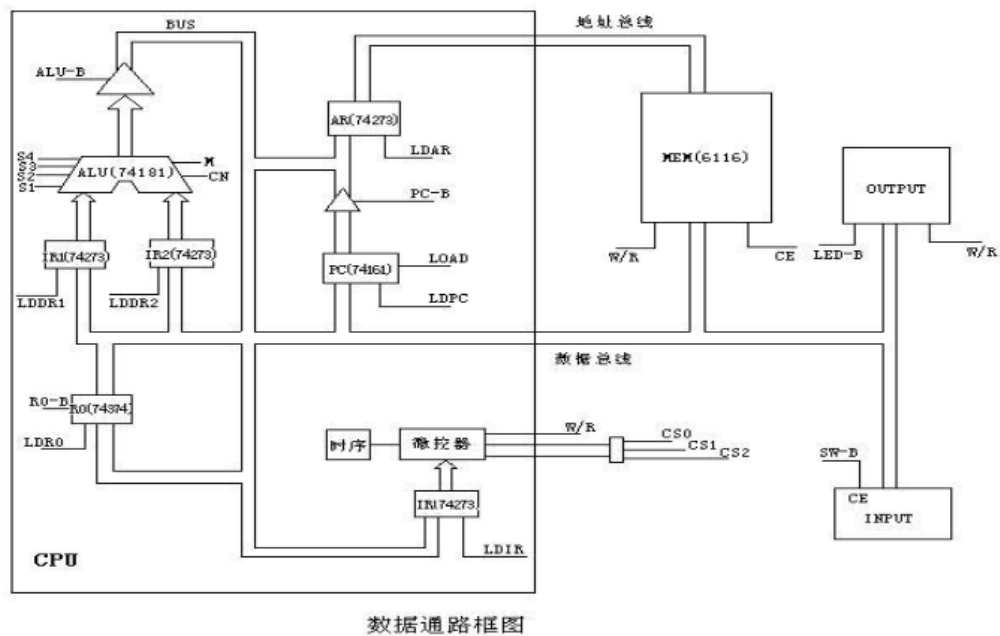
六、思考题

1、运算器的组成

答：运算器由算术逻辑单元（ALU）、累加器、状态寄存器、通用寄存器组等组成。

2、运算器的数据通路

答：



3、运算器的运算功能的选择

答：

运算类型	S3 S2 S1 S0	CN	功 能
逻辑运算	0000	X	$F=A$ (直通)
	0001	X	$F=B$ (直通)
	0010	X	$F=AB$ (FZ)
	0011	X	$F=A+B$ (FZ)
	0100	X	$F=\neg A$ (FZ)
移位运算	0101	X	$F=A$ 不带进位循环右移 B (取低 3 位) 位 (FZ)
	0110	0	$F=A$ 逻辑右移一位 (FZ)
		1	$F=A$ 带进位循环右移一位 (FC, FZ)
	0111	0	$F=A$ 逻辑左移一位 (FZ)
		1	$F=A$ 带进位循环左移一位 (FC, FZ)
算术运算	1000	X	置 $FC=CN$ (FC)
	1001	X	$F=A$ 加 B (FC, FZ)
	1010	X	$F=A$ 加 B 加 FC (FC, FZ)
	1011	X	$F=A$ 减 B (FC, FZ)
	1100	X	$F=A$ 减 1 (FC, FZ)
	1101	X	$F=A$ 加 1 (FC, FZ)
	1110	X	(保留)
	1111	X	(保留)

4、运算器的各种成部件的控制信号的作用

答：ALU 主要完成对二进制数据的定点算术运算（加减乘除）、逻辑运算（与或非异或）以及移位操作。在某些 CPU 中还有专门用于处理移位操作的移位器。

LDA、LDB 控制是否存入数据及存入数据的状态。

SD27-SD20 通过开关的逻辑状态控制输入数据的大小。

S0、S1、S2、S3 和 CN 是控制运算器做何种运算

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: 实验二 微程序控制器实验

实验时间: 2020 年 12 月 27 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: _____

实验二 微程序控制器实验

一、实验目的

- (1) 掌握微程序控制器的组成原理。
- (2) 掌握微程序的编制、写入，观察微程序的运行过程。

二、实验设备

PC 机一台，TD-CMA 实验系统一套。

三、微程序控制器的组成原理

1、微程序控制器的组成及原理（描述）

微程序控制器的基本任务是完成当前指令的翻译和执行，即将当前指令的功能转换成可以控制的硬件逻辑部件工作的微命令序列，完成数据传送和各种处理操作。它的执行方法就是将控制各部件动作的微命令的集合进行编码，即将微命令的集合仿照机器指令一样，用数字代码的形式表示，这种表示称为微指令。这样就可以用一个微指令序列表示一条机器指令，这种微指令序列称为微程序。微程序存储在一种专用的存储器中，称为控制存储器，微程序控制器原理框图如图 3-2-1 所示。

微程序控制器的组成见图 3-2-2，其中控制存储器采用 3 片 2816 的 E²PROM，具有掉电保护功能，微命令寄存器 18 位，用两片 8D 触发器（273）和一片 4D（175）触发器组成。微地址寄存器 6 位，用三片正沿触发的双 D 触发器（74）组成，它们带有清“0”端和预置端。在不判别测试的情况下，T2 时刻打入微地址寄存器的内容即为下一条微指令地址。当 T4 时刻进行测试判别时，转移逻辑满足条件后输出的负脉冲通过强置端将某一触发器置为“1”状态，完成地址修改。

2、微程序控制器的电路原理或通路图

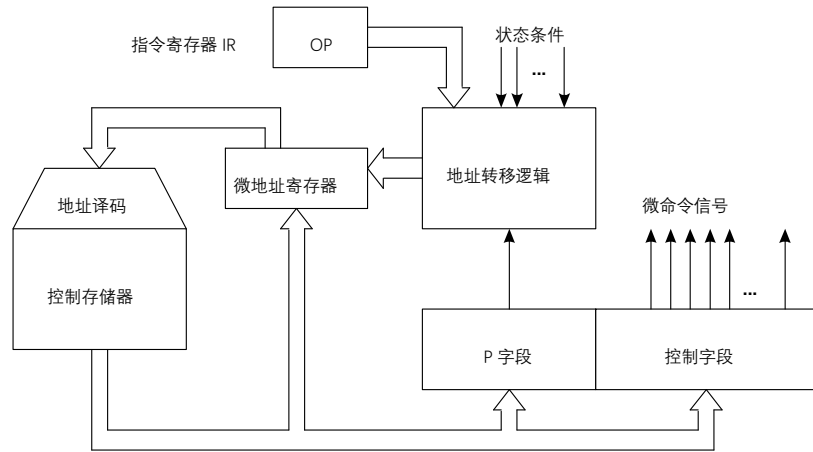


图 3-2-1 微程序控制器组成原理框图

按图 3-2-10 所示连接实验线路

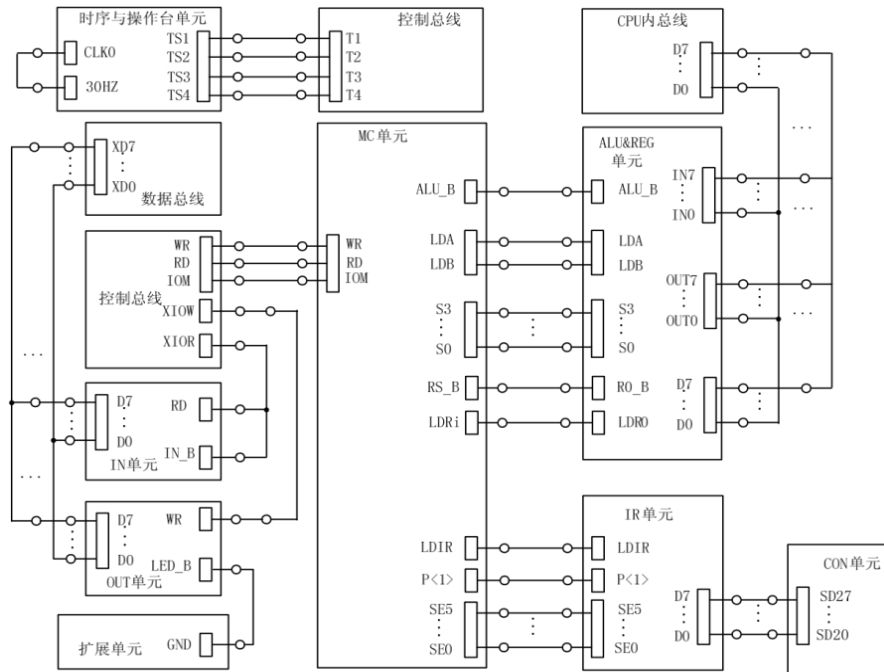


图 3-2-10 实验接线图

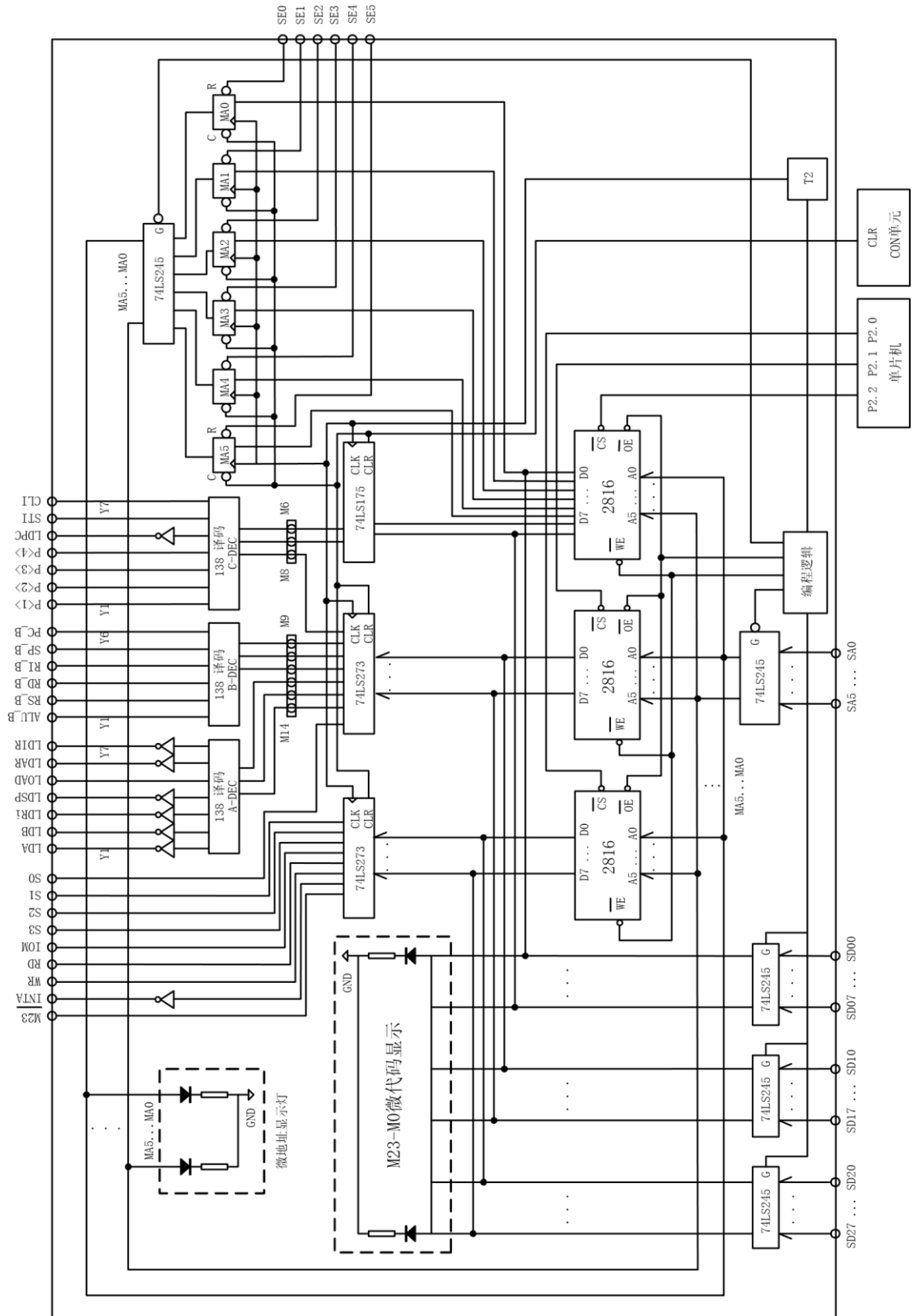
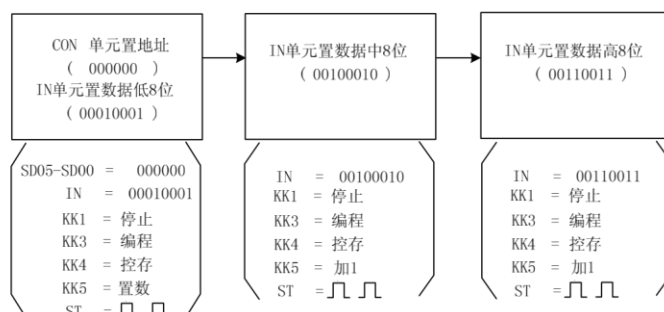


图 3-2-2 微程序控制器原理图

3、解释涉及到的控制信号

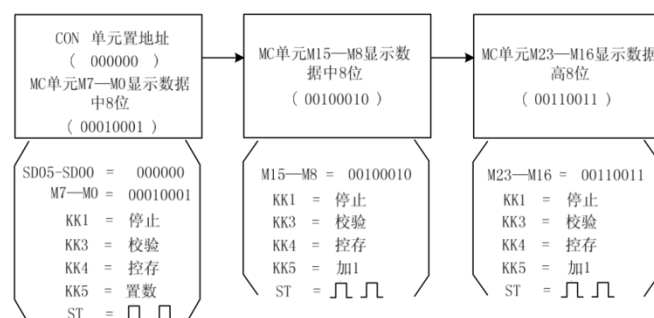
在实验平台中设有一组编程控制开关 KK3、KK4、KK5（位于时序与操作台单元），可实现对存储器（包括存储器和控制存储器）的三种操作：编程、校验、运行。考虑到对于存储器（包括存储器和控制存储器）的操作大多集中在一个地址连续的存储空间中，实验平台提供了便利的手动操作方式。以向 00H 单元中写入 332211 为例，对于控制存储器进行编辑的具体操作步骤如下：首先将 KK1 拨至‘停止’档、KK3 拨至‘编程’档、KK4 拨至‘控存’档、KK5 拨至‘置数’档，由 CON 单元的 SD05——SD00 开关给出需要编辑的控存单元首地址(000000)，IN 单元开关给出该控存单元数据的低 8 位 (00010001)，连续两次按动时序与操作台单元的开关 ST（第一次按动后 MC 单元低 8 位显示该单元以前存储的数据，第二次按动后显示当前改动的数据），此时 MC 单元的指示灯 MA5——MA0 显示当前地址（000000），M7——M0 显示当前数据（00010001）。

然后将 KK5 拨至‘加 1’档，IN 单元开关给出该控存单元数据的中 8 位（00100010），连续两次按动开关 ST，完成对该控存单元中 8 位数据的修改，此时 MC 单元的指示灯 MA5——MA0 显示当前地址（000000），M15——M8 显示当前数据（00100010）；再由 IN 单元开关给出该控存单元数据的高 8 位（00110011），连续两次按动开关 ST，完成对该控存单元高 8 位数据的修改此时 MC 单元的指示灯 MA5——MA0 显示当前地址（000000），M23——M16 显示当前数据（00110011）。此时被编辑的控存单元地址会自动加 1（01H），由 IN 单元开关依次给出该控存单元数据的低 8 位、中 8 位和高 8 位配合每次开关 ST 的两次按动，即可完成对后续单元的编辑。



编辑完成后需进行校验，以确保编辑的正确。以校验 00H 单元为例，对于控制存储器进行校验的具体操作步骤如下：首先将 KK1 拨至‘停止’档、KK3 拨至‘校验’档、KK4 拨至‘控存’档、KK5 拨至‘置数’档。由 CON 单元的 SD05——SD00 开关

给出需要校验的控存单元地址（000000），连续两次按动开关 ST，MC 单元指示灯 M7——M0 显示该单元低 8 位数据（00010001）；KK5 拨至‘加 1’档，再连续两次按动开关 ST，MC 单元指示灯 M15——M8 显示该单元中 8 位数据（00100010）；再连续两次按动开关 ST，MC 单元指示灯 M23——M16 显示该单元高 8 位数据（00110011）。再连续两次按动开关 ST，地址加 1，MC 单元指示灯 M7——M0 显示 01H 单元低 8 位数据。如校验的微指令出错，则返回输入操作，修改该单元的数据后再进行校验，直至确认输入的微代码全部准确无误为止，完成对微指令的输入。



位于实验平台 MC 单元左上角一列三个指示灯 MC2、MC1、MC0 用来指示当前操作的微程序字段，分别对应 M23——M16、M15——M8、M7——M0。实验平台提供了比较灵活的手动操作方式，比如在上述操作中在对地址置数后将开关 KK4 拨至‘减 1’档，则每次随着开关 ST 的两次拨动操作，字节数依次从高 8 位到低 8 位递减，减至低 8 位后，再按动两次开关 ST，微地址会自动减一，继续对下一个单元的操作。

本实验除了用到指令寄存器（IR）和通用寄存器 R0 外，还要用到 IN 和 OUT 单元，从微控器出来的信号中只有 IOM、WR 和 RD 三个信号，所以对这两个单元的读写信号还应先经过译码，其译码原理如图 3-2-4 所示。IR 单元的原理图如图 3-2-5 所示，R0 单元原理如图 3-2-7 所示，

IN 单元的原理图见图 2-1-3 所示，OUT 单元的原理图见图 3-2-6 所示。

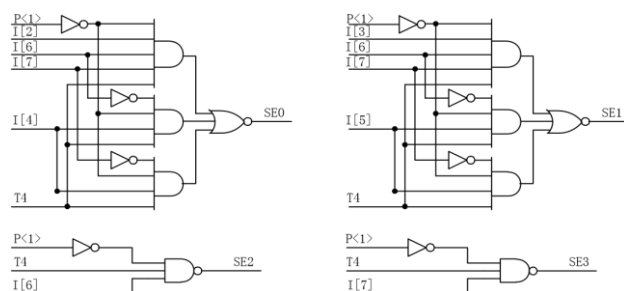


图 3-2-3 指令译码原理图

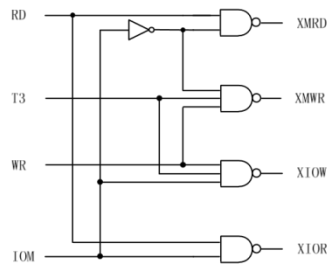


图 3-2-4 读写控制逻辑

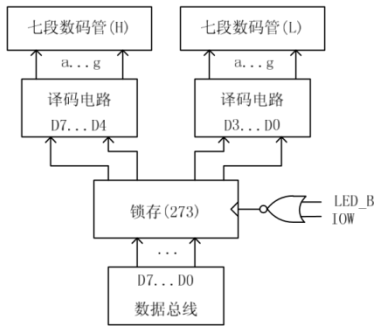


图 3-2-5 IR 单元原理图

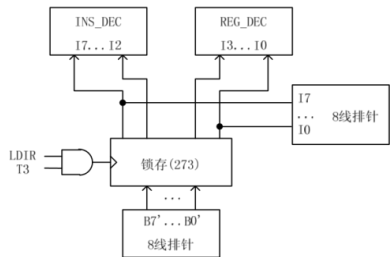


图 3-2-6 OUT 单元原理图

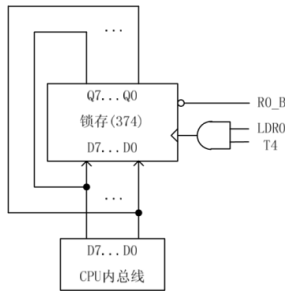


图 3-2-7 R0 原理图

4、微指令的格式及编码方法

微指令字长共 24 位，控制位顺序如表 3-2-1：

表 3-2-1 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	RO_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	保留	1	0	1	保留	1	0	1	保留
1	1	0	保留	1	1	0	保留	1	1	0	保留
1	1	1	LDIR	1	1	1	保留	1	1	1	保留

其中 MA5...MA0 为 6 位的后续微地址, A、B、C 为三个译码字段, 分别由三个控制位译码出多位。C 字段中的 P<1>为测试字位。其功能是根据机器指令及相应微代码进行译码, 使微程序转入相应的微地址入口, 从而实现完成对指令的识别, 并实现微程序的分支, 本系统上的指令译码原理如图 3-2-3 所示, 图中 I7...I2 为指令寄存器的第 7...2 位输出, SE5...SE0 为微控器单元微地址锁存器的强置端输出, 指令译码逻辑在 IR 单元的 INS DEC (GAL20V8) 中实现。

5、画出相关指令得周期流程图

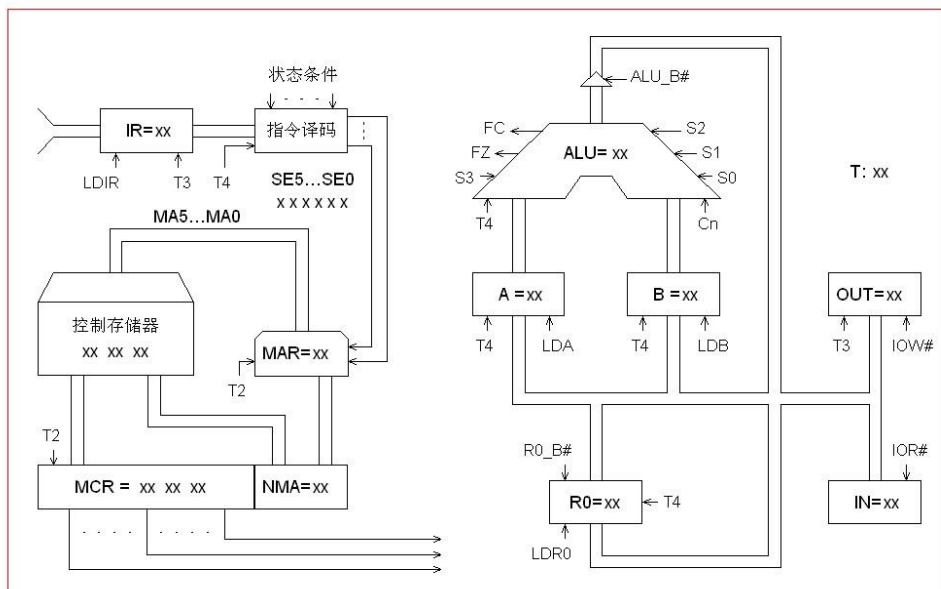


图 3-2-8 数据通路图

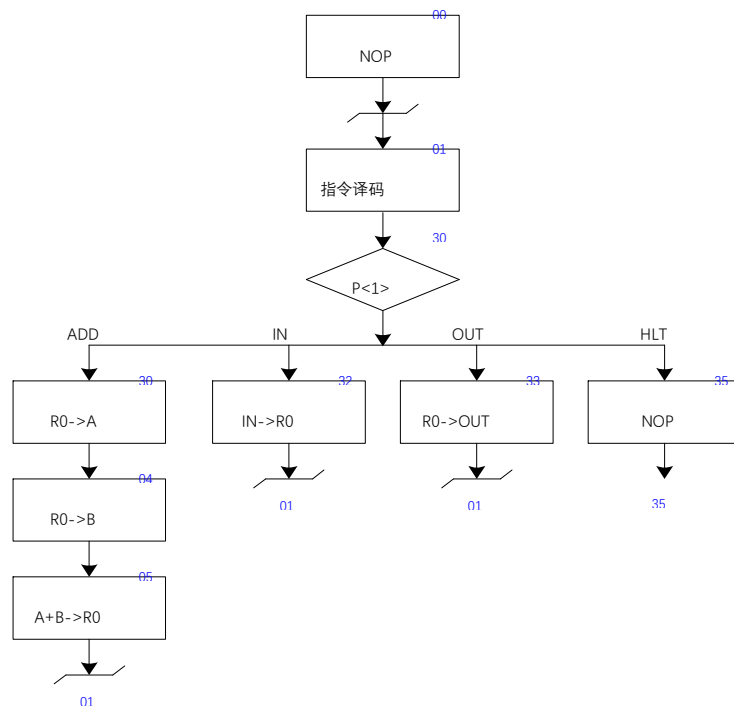


图 3-2-9 微程序流程图

四、分析与比较与实验一的不同

实验一是 CPU 内部变换不同的数值后进行不同的运算。实验二是在微程序的控制下进行运算，通过机器指令及微指令来控制的。

五、实验结果与问题分析

由实验可得按要求完成实验步骤后，数据会按照数据通路进行流程验算，最终的到运算数二倍结果。

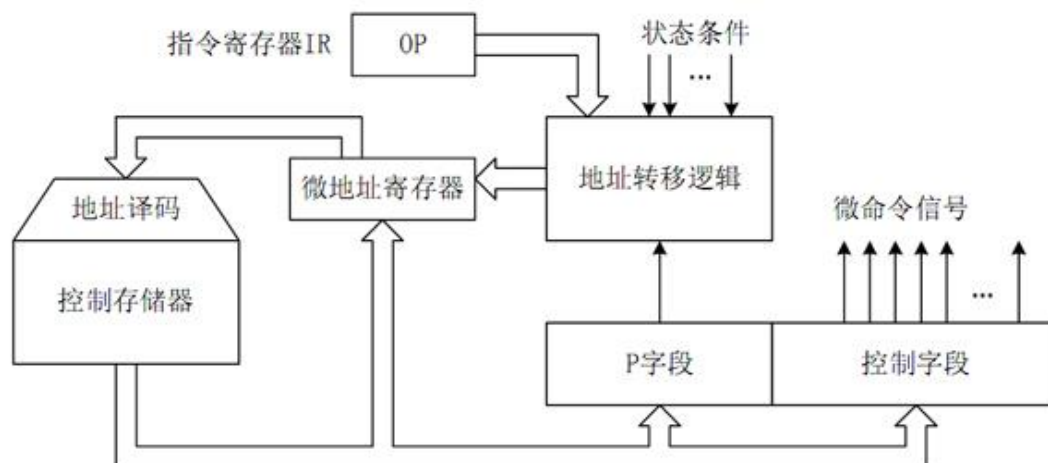
六、实验体会

通过此次的微程序控制器的实验，我们通过实际操作基本掌握微程序控制器的功能、组成知识。以及程序的编制、写入、观察微程序的运行等功能。了解到了理论知识与实际操作的区别。

七、思考题

1、画出微程序控制器原理框图，解释每部分作用。

答：



- (1) 控制存储器：存储微指令。
- (2) 指令寄存器：存储机器指令。
- (3) 微命令寄存器：记录当前正在执行的微命令以及判别字段。
- (4) 微地址寄存器：根据判别字段的不同微地址寄存器存储不同的内容，当判别字段无效时存储的是下一条微指令的地址。

2、说明微指令和机器指令的关系

答：一条机器指令对应一个微程序，这个微程序是由若干条微指令构成的。因此，一条机器指令的功能是若干条微指令组成的序列来实现的。

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。

二、实验设备

PC 机一台, TD-CMA 实验系统一套。

三、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

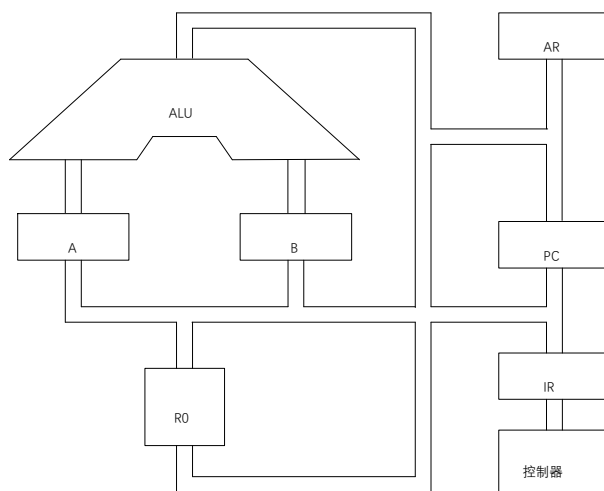


图 5-1-1 基本 CPU 构成原理图

按图 5-1-5 连接实验线路。

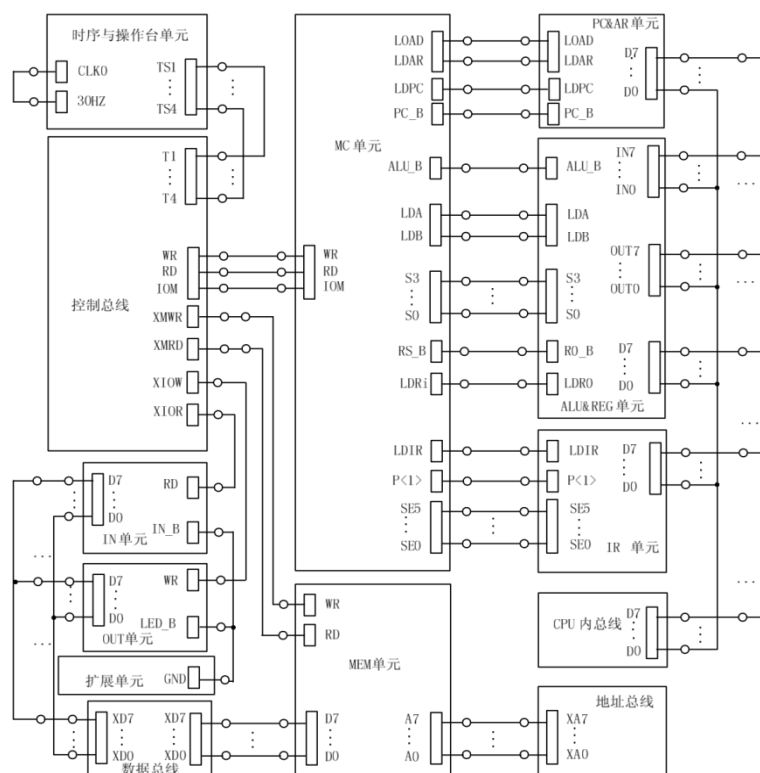


图 5-1-5 实验接线图

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。

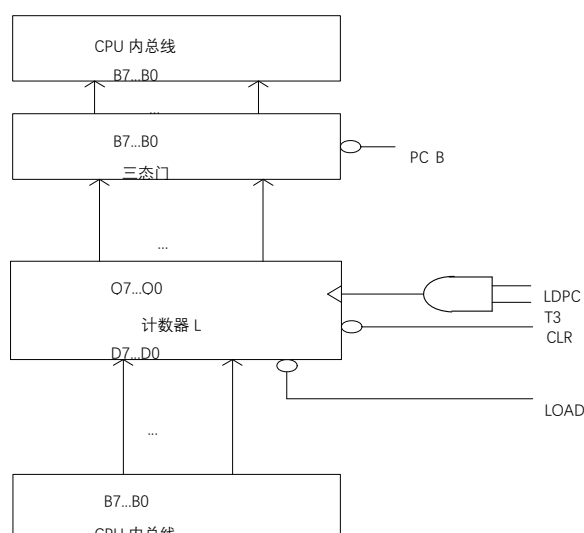


图 5-1-2 程序计数器(PC)原理图

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 \rightarrow R0
ADD	0000 0000	R0 + R0 \rightarrow R0
OUT	0011 0000	R0 \rightarrow OUT
JMP addr	1110 0000 *****	addr \rightarrow 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。

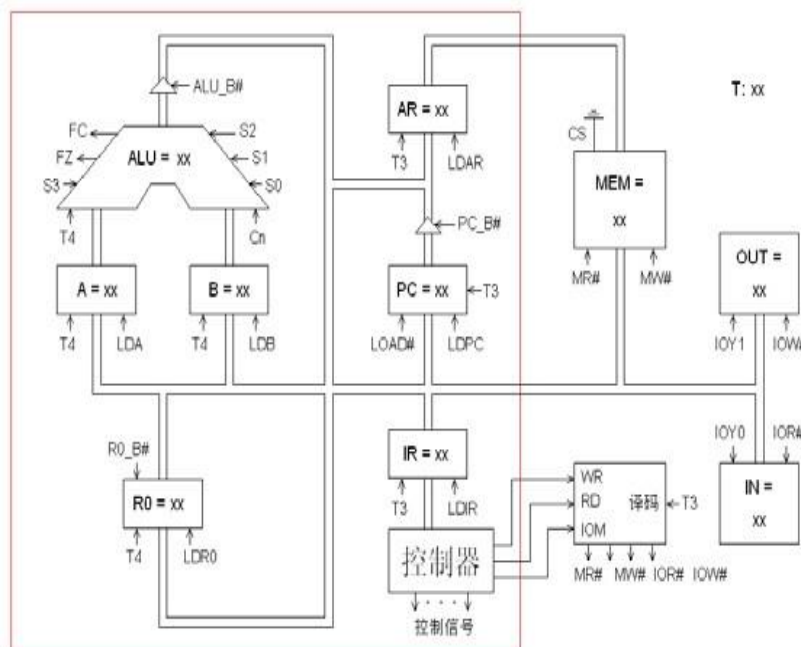


图 5-1-3 数据通路图

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 \rightarrow R0
ADD	0000 0000	R0 + R0 \rightarrow R0
OUT	0011 0000	R0 \rightarrow OUT
JMP addr	1110 0000 *****	addr \rightarrow 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						
0	0	0	NOP						
0	0	1	P<1>						
0	1	0	保留						
0	1	1	保留						
1	0	0	保留						
1	0	1	LDPC						
1	1	0	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。

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。

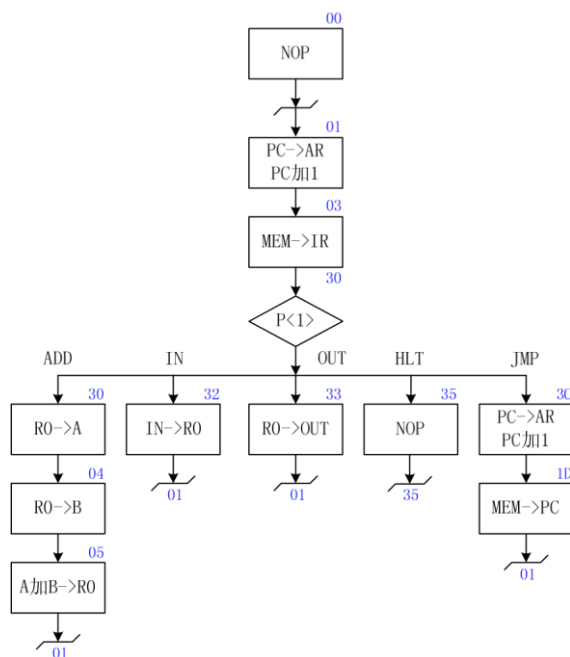


图 5-1-4 简单模型机微程序流程图

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

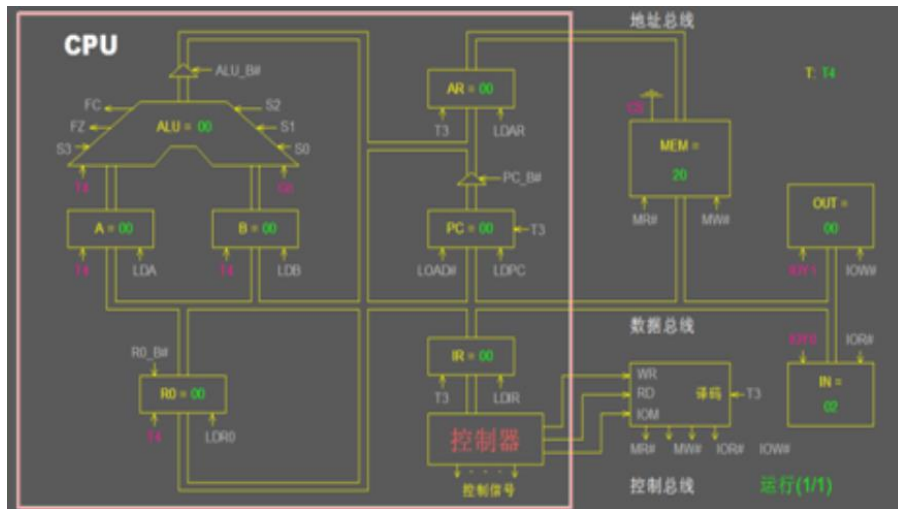
本实验有如何构建一个简单 CPU 的全部过程，并且在此 CPU 的基础上，继续构建一个简单的模型计算机。CPU 由运算器、微程序控制器、通用寄存器，指令寄存器、程序计数器和地址寄存器组成。在 CPU 的基础上增加一个主存和基本的输入输出部件，就构成一个简单的模型计算机。在掌握部件单元电路的基础上，进一步将其构造一台基本模型计算机。

七、思考题

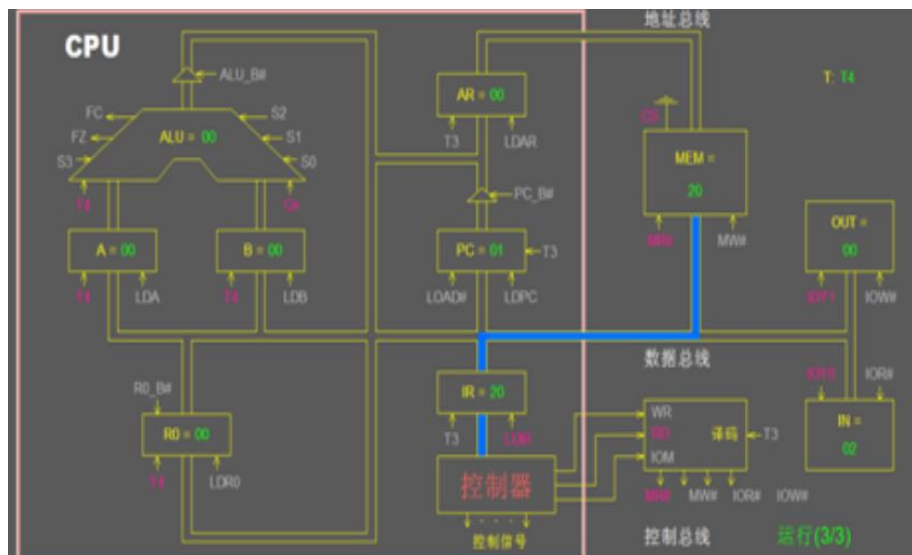
根据实验模型画出数据通路图，并解释每个部分作用和工作流程。

答：

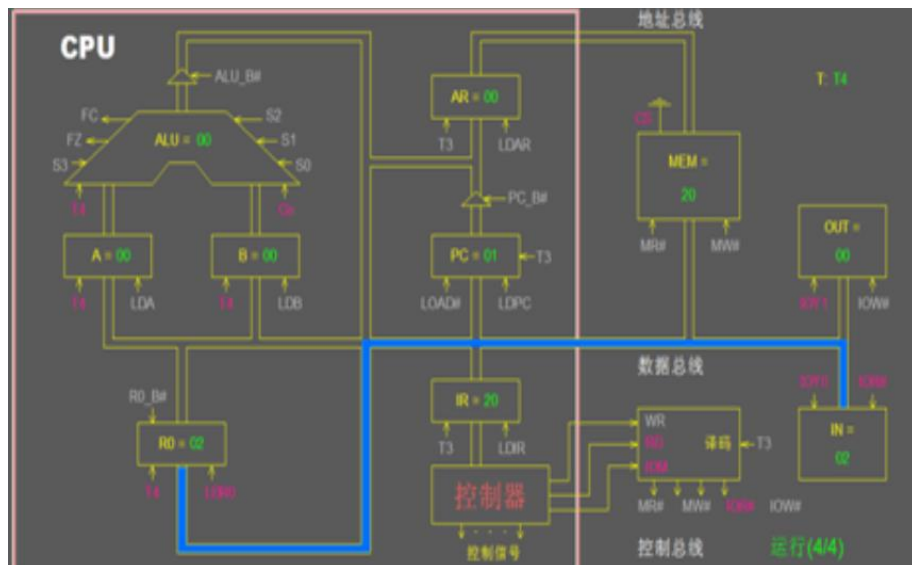
（1）从 IN 单元读入数据送至 AR。首先，AR 地址为 00，计数器 PC 加 1，机器指令代码送至 MEM 单元，MEM=20。



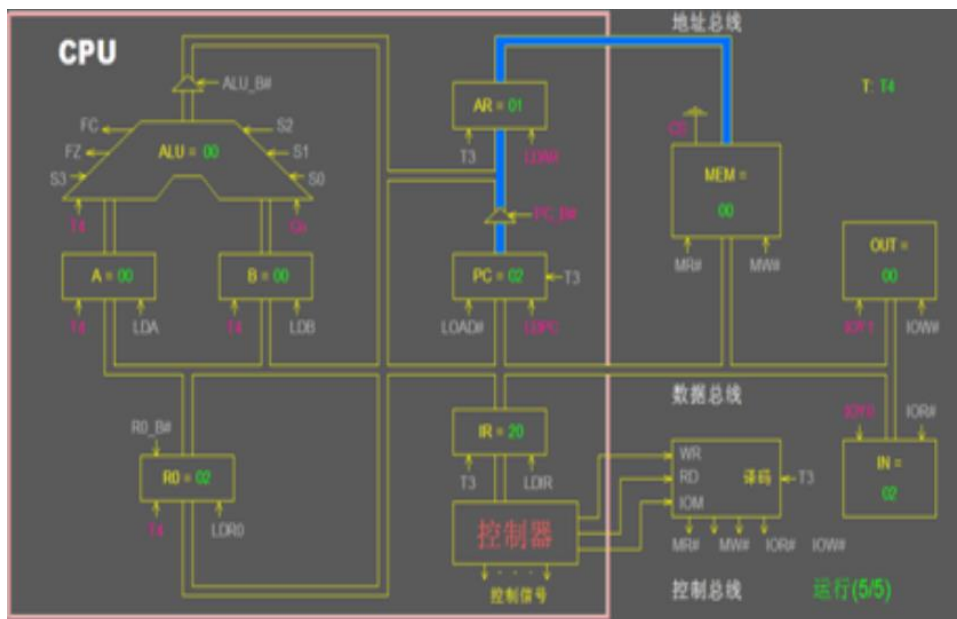
(2) MEM 中的机器指令代码送至 IR 单元，再到控制器，进行译码操作。



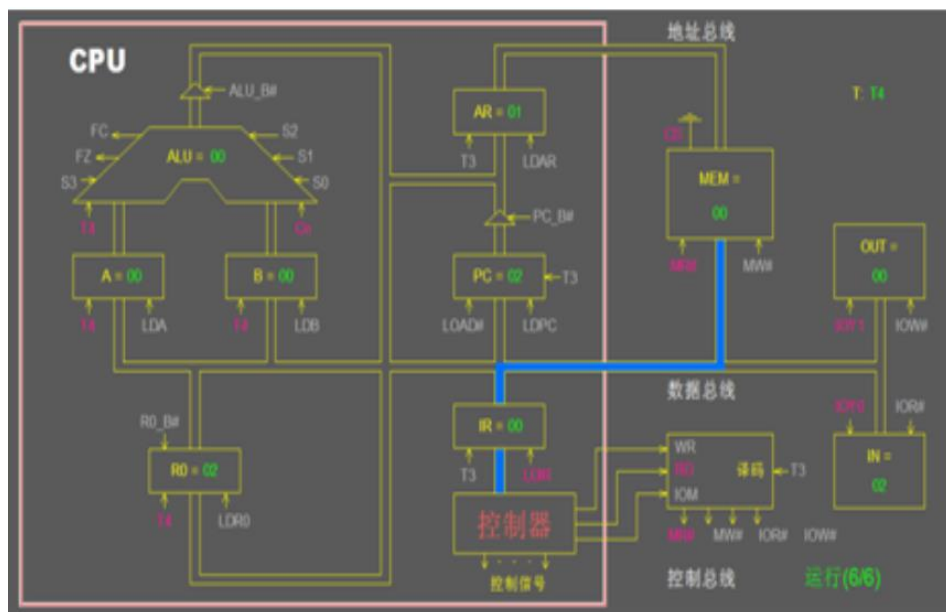
(3) R0 寄存器读入 IN 单元输入的数据。



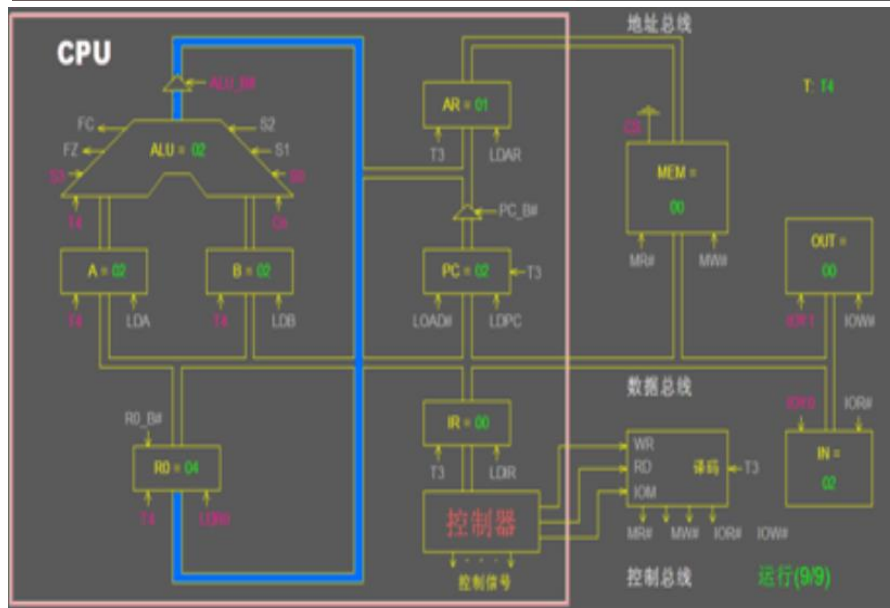
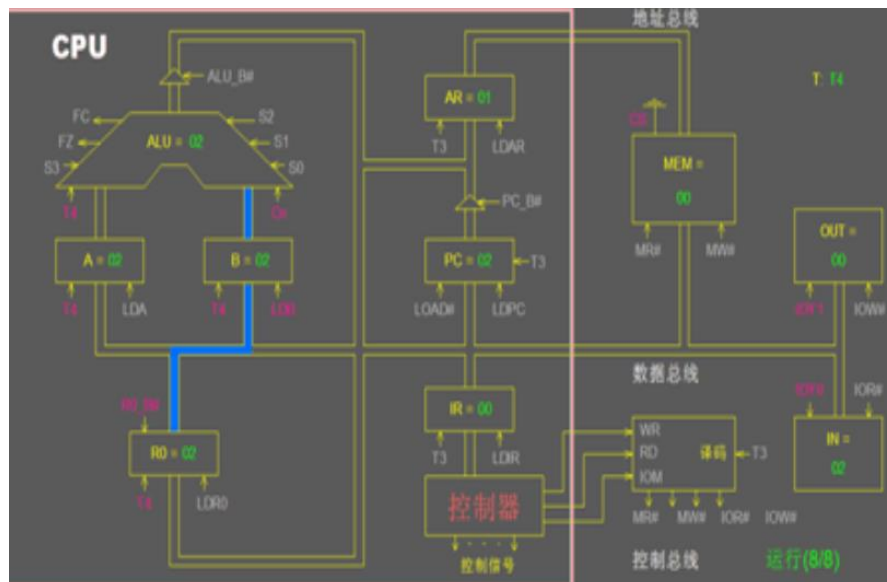
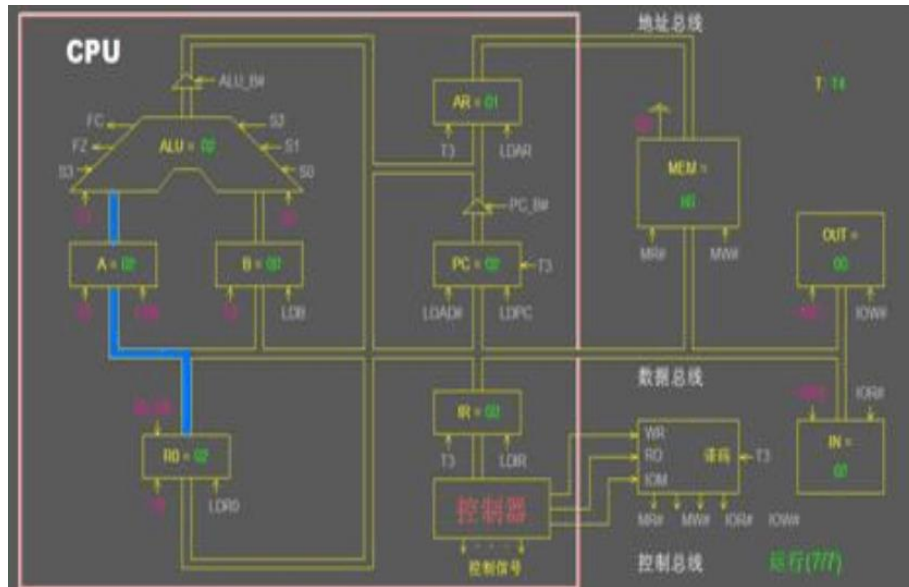
(4) AR 地址为 01，计数器 PC 加 1，机器指令代码送至 MEM 单元，MEM=00



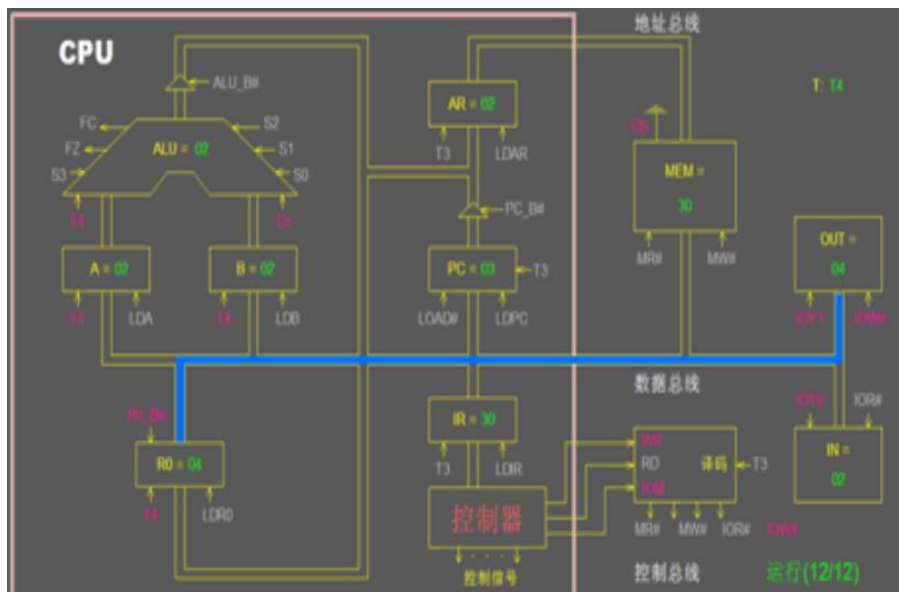
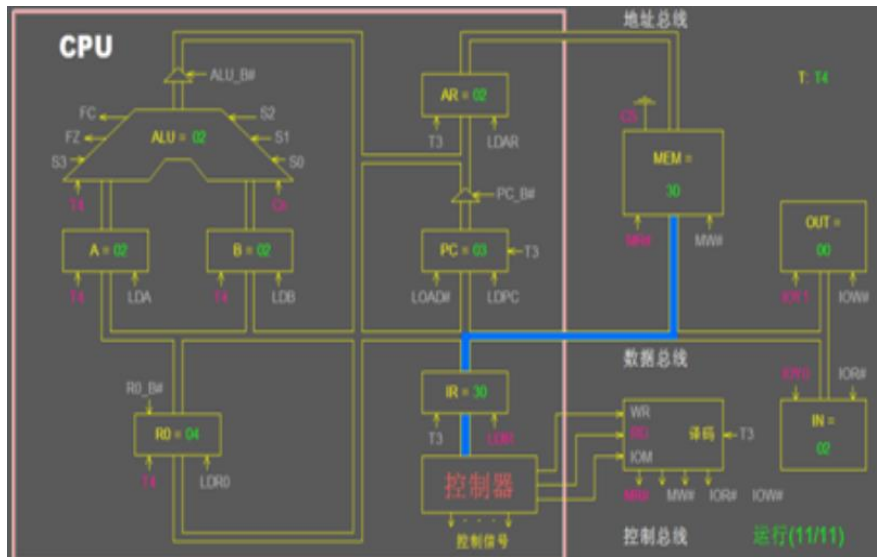
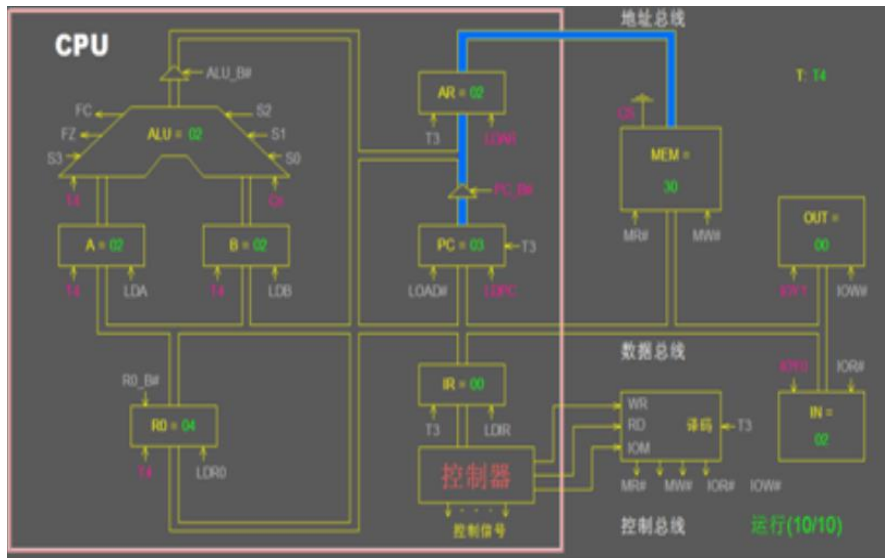
(5) MEM 中的机器指令代码送至 IR 单元，再到控制器，进行译码操作。



(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

