

# 实 验 报 告

课程名称\_\_\_\_\_

院 系 部:

专业班级:

学生姓名:

学 号:

同 组 人:

实验台号:

指导教师:

成 绩:

实验日期:

华北电力大学

## 实验名称：超前进位加法器设计实验

### 一、 实验目的：

- (1) 掌握超前进位加法器的原理及其设计方法。
- (2) 熟悉 FPGA 应用设计及 EDA 软件的使用。

### 二、 实验设备：实验中所用到的仪器和设备。

PC 机一台，TD-CMX 实验系统一套。

### 三、 实验线路：

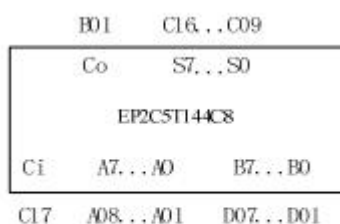


图 1-2-7 引脚分配图

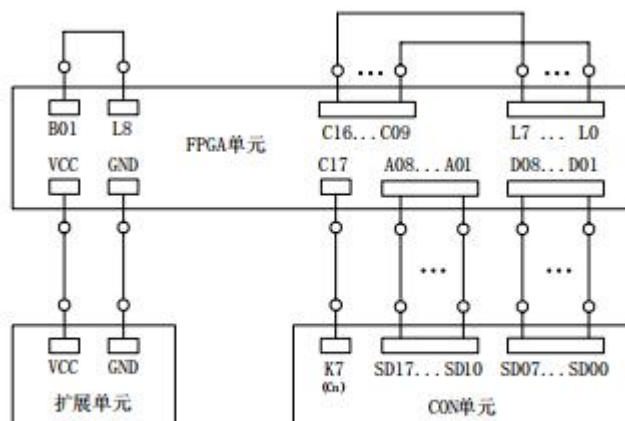


图 1-2-8 实验接线图

### 四、 实验步骤及实验结果：

(1) 根据上述加法器的逻辑原理使用 Quartus II 软件编辑相应的电路原理图并进行编译，其在 EP2C5T144C8 芯片中对应的引脚如图 1-2-7 所示，框外文字表示 I/O 号，框内文字表示该引脚的含义（本实验例程见‘安装路径\FPGA\Adder\Adder.qpf’工程）。

(2) 关闭实验系统电源, 按图 1-2-8 连接实验电路, 图中将用户需要连接的信号用圆圈标明。

(3) 打开实验系统电源, 将下载电缆插入 FPGA 单元的 JTAG 口, 把生成的 SOF 文件下载到 FPGA 单元中去。

(4) 以 CON 单元中的 SD17…SD10 八个二进制开关为被加数 A, SD07…SD00 八个二进制开关为加数 B, K7 用来模拟来自低位的进位信号, 相加的结果在 FPGA 单元的 L7…L0 八个 LED 灯显示, 相加后向高位的进位用 FPGA 单元的 L8 灯显示。给 A 和 B 置不同的数, 观察相加的结果。

## 五、 结果分析和讨论:

六、 **实验心得:** 实验成功或失败的原因, 实验后的心得体会、建议等。

# 实验名称: 阵列乘法器设计实验

## 一、 实验目的:

- (1) 掌握乘法器的原理及其设计方法。
- (2) 熟悉 FPGA 应用设计及 EDA 软件的使用。

## 二、 实验内容:

设计一个阵列乘法器

## 三、 实验设备: 实验中所用到的仪器和设备。

PC 机一台, TD-CMX 实验系统一套。

## 四、 实验线路:

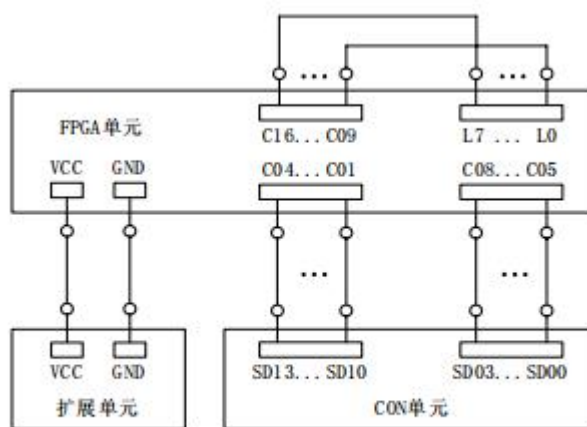


图 1-3-3 阵列乘法器实验接线图

## 五、 实验步骤及实验结果：

(1) 根据上述阵列乘法器的原理，使用 Quartus II 软件编辑相应的电路原理图并进行编译，其在 EP2C5T144C8 芯片中对应的引脚如图 1-3-2 所示，框外文字表示 I/O 号，框内文字表示该引脚的含义（本实验例程见‘安装路径\FPGA\Multiply\Multiply.qpf’工程）。

(2) 关闭实验系统电源，按图 1-3-3 连接实验电路，图中将用户需要连接的信号用圆圈标明。

(3) 打开实验系统电源，将下载电缆插入 FPGA 单元的 JTAG 口，把生成的 SOF 文件下载到 FPGA 单元中去，FPGA 单元介绍见实验 1.2。

(4) 以 CON 单元中的 SD10…SD13 四个二进制开关为乘数 A，SD14…SD17 四个二进制开关为被乘数 B，而相乘的结果在 FPGA 单元的 L7…L0 八个 LED 灯显示。给 A 和 B 置不同的数，观察相乘的结果。

## 六、 结果分析和讨论：

七、 实验心得：实验成功或失败的原因，实验后的心得体会、建议等。

**实验名称： Cache 控制器设计实验**

## 一、 实验目的:

- (1) 掌握 Cache 控制器的原理及其设计方法。
- (2) 熟悉 FPGA 应用设计及 EDA 软件的使用。

## 二、 实验内容:

设计 Cache 控制器

## 三、 实验设备:

PC 机一台, TD-CMX 实验系统一套。

## 四、 实验线路:

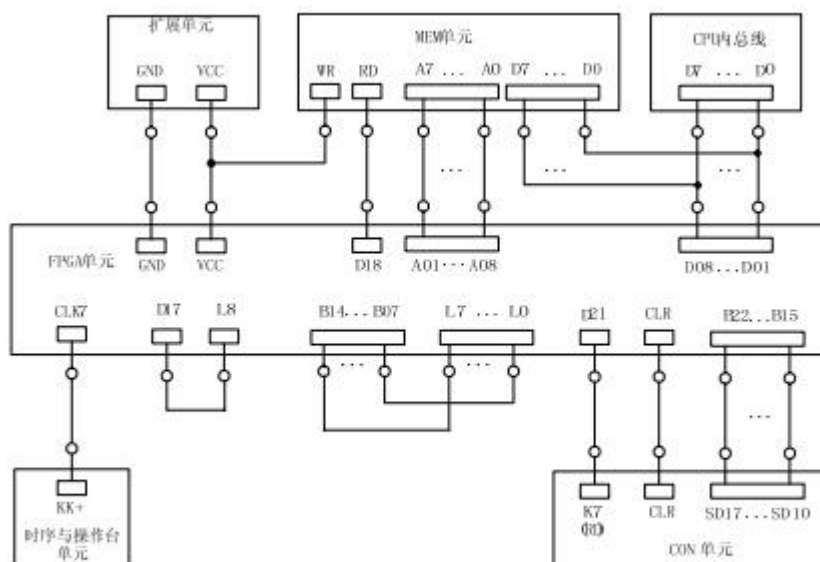


图 2-2-6 实验接线图

## 五、 程序设计 (流程图及程序代码)

```
; /******* //  
; // //  
; // Cache 控制器实验指令文件 //  
; // //  
; // By TangDu CO.,LTD //  
; // //  
; /******* //
```

```

; //***** Start Of Main Memory Data ***** //
$P 00 11 ; 数据
$P 01 22
$P 02 33
$P 03 44
$P 04 55
$P 05 66
$P 06 77
$P 07 88
$P 08 99
$P 09 AA
$P 0A BB
$P 0B CC
$P 0C DD
$P 0D EE
$P 0E FF
$P 0F 00
; //***** End Of Main Memory Data ***** //

```

## 六、 实验步骤及实验结果：

(1) 使用 Quartus II 软件编辑实现相应的逻辑并进行编译，直到编译通过，Cache 控制器在 EP2C5T144C8 芯片中对应的引脚如图 2-2-5 所示，框外文字表示 I/O 号，框内文字表示该引脚的含义（本实验例程见‘安装路径\FPGA \CacheCtrl\CacheCtrl.qpf’工程）。

(2) 关闭实验系统电源，按图 2-2-6 连接实验电路，并检查无误，图中将用户需要连接的信号用圆圈标明。

(3) 打开实验系统电源，将下载电缆插入 FPGA 单元的 JTAG 口，把生成的 SOF 文件下载到 FPGA 单元中去，FPGA 单元介绍见实验 1.2。

(4) 将时序与操作台单元的开关 KK3 置为‘运行’档，CLR 信号由 CON 单元的 CLR 模拟给出，按动 CON 单元的 CLR 按钮，清空区表。

(5) 预先往主存写入数据：联机软件提供了机器程序下载功能，以代替手动读写主存，机器程序以指定的格式写入到以 TXT 为后缀的文件中，

(6) 联机软件在启动时会读取所有机器指令和微指令，在指令区显示，软件启动后，也可以选择菜单命令“【转储】—【刷新指令区】”读取下位机指令，并在指令区显示。点击指令区的‘主存’TAB 按钮，两列数据中显示了主存的所有数据，第一列为主存地址，第二列为该地

址中的数据。对上面文件检查机器程序是否正确，如果不正确，则说明写入操作失败，应重新写入，可以通过联机软件单独修改某个单元的指令，单击需修改单元的数据，此时该单元变为编辑框，输入 2 位数据并回车，编辑框消失，写入数据以红色显示。

(7) CPU 访问主存地址由 CON 单元的 SD17…SD10 模拟给出，如 00000001。CPU 访问主存的读信号由 CON 单元的 K7 模拟给出，置 K7 为低，可以观察到 FPGA 单元上的 L8 指示灯亮，L0…L7 指示灯灭，表示 Cache 失效。此时按动 KK 按钮四次，注意 CPU 内总线上指示灯的变化情况，地址会依次加一，数据总线上显示的是当前主存数据，按动四次 KK 按钮后，L8 指示灯变灭，L0…L7 上显示的值即为 Cache 送往 CPU 的数据。

(8) 重新给出主存访问地址，如 00000011，L8 指示灯变灭，表示 Cache 命中，说明第 0 块数据已写入 Cache。

(9) 记住 01H 单元的数据，然后通过联机软件，修改 01H 单元的数据，重新给出主存访问地址 00000001，再次观察 L0-L7 指示灯表示的值是 01H 单元修改前的值，说明送往 CPU 的数据是由 Cache 给出的。

(10) 重新给出大于 03H 地址，体会 Cache 控制器的工作过程。

## 七、 结果分析和讨论：

八、 **实验心得：** 实验成功或失败的原因，实验后的心得体会、建议等。

# 实验名称： 时序发生器设计实验

## 一、 实验目的：

- (1) 掌握时序发生器的原理及其设计方法。
- (2) 熟悉 FPGA 应用设计及 EDA 软件的使用。

## 二、 实验内容：

设计一个时序发生器

### 三、 实验设备：

PC 机一台，TD-CMX 实验系统一套。

### 四、 实验线路：

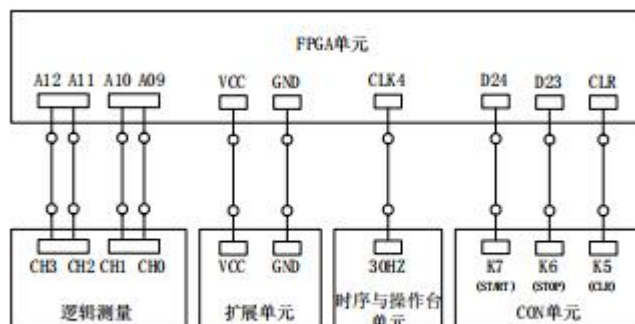


图 3-1-4 实验接线图

### 五、 实验步骤及实验结果：

(1)参照上面的实验原理，用 VHDL 语言来具体设计一个时序发生器。使用 Quartus II 软件编辑 VHDL 文件并进行编译，时序发生器在 EP2C5T144C8 芯片中对应的引脚如图 3-1-3 所示，框外文字表示 I/O 号，框内文字表示该引脚的含义（本实验例程见‘安装路径\FPGA\Timer\Timer.qpf’工程）。

(2) 关闭实验系统电源，按图 3-1-4 连接实验电路，并检查无误，图中将用户需要连接的信号用圆圈标明。

(3) 打开实验系统电源，将下载电缆插入 FPGA 单元的 JTAG 口，把生成的 SOF 文件下载到 FPGA 单元中去，FPGA 单元介绍见实验 1.2。

(4)将 CON 单元的 K7 (START)、K6 (STOP) 开关置‘1’，按下 CON 单元的总清按钮 CLR，使 T1...T4 输出低。运行联机软件，选择“【波形】—【打开】”打开逻辑示波器窗口，然后选择“【波形】—【运行】”启动逻辑示波器，逻辑示波器窗口显示 T1...T4 四路时序信号波形。

(5) 将 CON 单元的 K7 (START) 开关置‘1-0-1’，启动 T1...T4 时序，示波器窗口



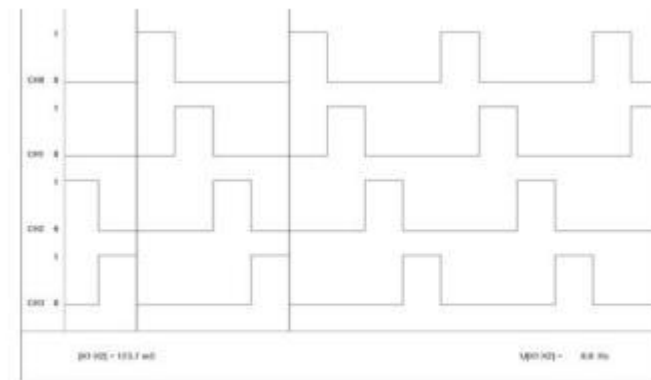


图 3-1-5 时序波形图

显示 T1...T4 波形，如图 3-1-5 所示。

(6) 将 CON 单元的 K6 (STOP) 开关置'0'，停止 T1...T4 时序，示波器窗口显示 T1...T4 波形均变为低。

## 六、 结果分析和讨论：

七、 实验心得：实验成功或失败的原因，实验后的心得体会、建议等

# 实验名称： 具有中断控制功能的总线接口实验

## 一、 实验目的：

1. 掌握中断控制信号线的功能和应用。
2. 掌握在系统总线上设计中断控制信号线的方法

## 二、 实验内容：

### 三、 实验设备：

PC 机一台，TD-CMX 实验系统一套，电压表一台。

### 四、 实验线路：

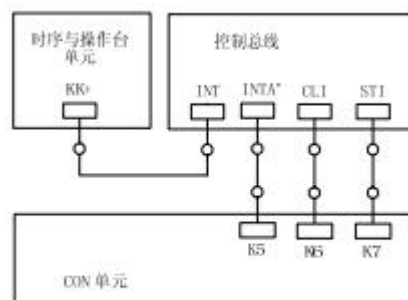


图 4.2-3 实验接线图

### 五、 实验步骤及实验结果：

(1) 按照图 4-2-3 实验接线图进行连线。

(2) 具体操作步骤图示如下：

① 对总线进行置中断操作 ( $K6=1, K7=0$ )，观察控制总线部分的中断允许指示灯 EI，此时 EI 亮,表示允许响应外部中断。按动时序与操作台单元的开关 KK，观察控制总线单元的指示灯 INTR，发现当开关 KK 按下时 INTR 变亮，表示总线将外部的中断请求送到 CPU。

② 对总线进行清中断操作 ( $K6=0, K7=1$ )，观察控制总线部分的中断允许指示灯 EI，此时 EI 灭,表示禁止响应外部中断。按动时序与操作台单元的开关 KK，观察控制总线单元的指示灯 INTR，发现当开关 KK 按下时 INTR 不变，仍然为灭，表示总线锁死了外部的中断请求。

③ 对总线进行置中断操作( $K6=1, K7=0$ )，当 CPU 给出的中断应答信号  $INTA'$  ( $K5=0$ ) 有效时，使用电压表测量数据缓冲 74LS245 的 DIR (第 1 脚)，显示为低，表示 CPU 允许外部送中断向量号

### 六、 结果分析和讨论：

### 七、 实验心得：实验成功或失败的原因，实验后的心得体会、建议等

# 实验名称： 具有 DMA 控制功能的总线接口实验

## 一、 实验目的：

1. 掌握 DMA 控制信号线的功能和应用。
2. 掌握在系统总线上设计 DMA 控制信号线的方法。

## 二、 实验内容：

## 三、 实验设备：

PC 机一台，TD-CMX 实验系统一套，电压表一台。

## 四、 实验线路：

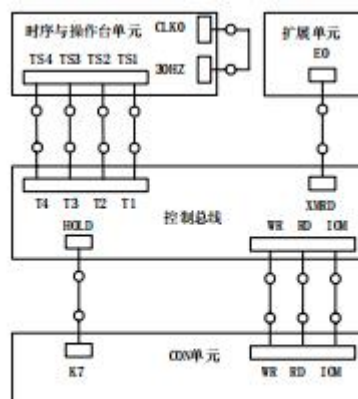


图 4-3-2 实验接线图

## 五、 实验步骤及实验结果：

- (1) 按照图 4-3-2 实验接线图进行连线
- (2) 具体操作步骤如下：

① 将时序与操作台单元的开关 KK1、KK3 置为‘运行’档，开关 KK2 置为‘单拍’档，按动 CON 单元的总清按钮 CLR，将 CON 单元的 WR、RD、IOM 分别置为“0”、“1”、“0”，此时 XMRD 为低，相应的指示灯 E0 灭。使用电压表测量数据总线和地址总线左侧的芯片 74LS245 的使能控制信号（第

19 脚),发现电压为低,说明数据总线和地址总线与 CPU 连通。

② 然后将 CON 单元的 K7 置为 1,连续按动时序与操作台单元的开关 ST,T4 时刻控制总线的指示灯 HALD 为亮,继续按动开关 ST,发现控制总线单元的时钟信号指示灯 T1——T4 保持不变,说明 CPU 的时钟被锁死。此时 XMRD 为高阻态,相应的指示灯 E0 亮。使用万用表测量数据总线和地址总线左侧的芯片 74LS245 的使能控制信号(第 19 脚),发现电压为高,说明总线和 CPU 的连接被阻断。

③将 CON 单元的 K7 置为 0,按动时序与操作台单元的开关 ST,当时序信号走到 T4 时刻时,控制总线的指示灯 HALD 为灭,继续按动开关 ST,发现控制总线单元的时钟信号指示灯 T1——T4 开始变化,说明 CPU 的时钟被接通。此时 XMRD 受 CPU 控制,恢复有效为低,相应的指示灯 E0 灭。使用万用表测量数据总线和地址总线左侧的芯片 74LS245 的使能控制信号(第 19 脚),发现电压为低,说明总线和 CPU 恢复连通。

## 六、 结果分析和讨论:

七、 实验心得: 实验成功或失败的原因,实验后的心得体会、建议等。

## 实验名称: 硬布线控制器模型机设计实验

### 一、 实验目的:

- (1) 掌握硬布线控制器的组成原理、设计方法。
- (2) 了解硬布线控制器和微程序控制器的各自优缺点。

### 二、 实验内容:

### 三、 实验设备：

PC 机一台，TD-CMX 实验系统一套。

### 四、 实验线路：

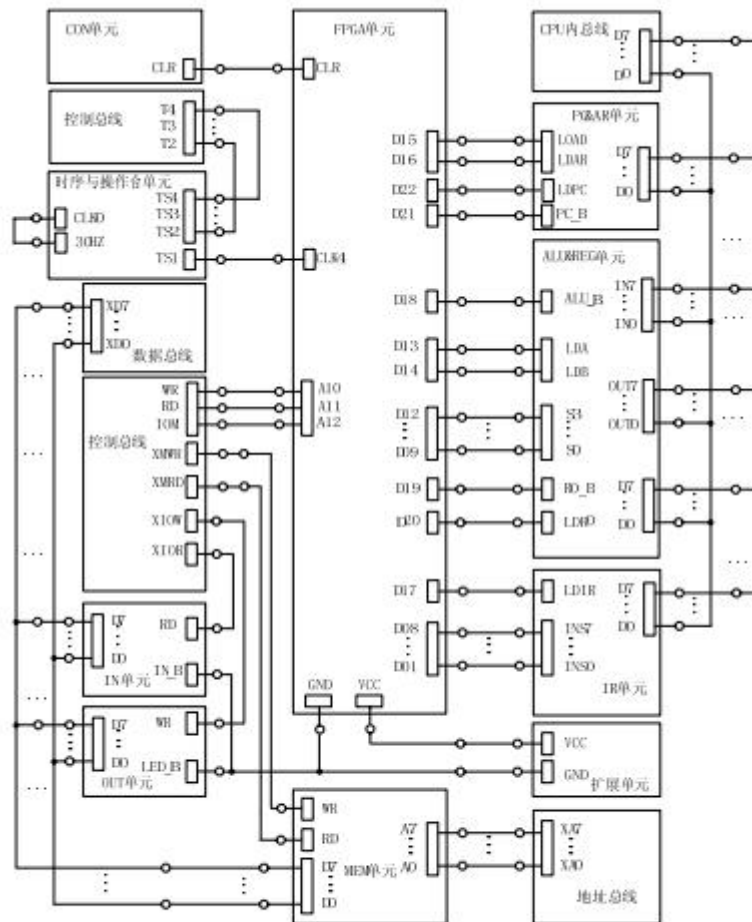


图 5-2-3 实验接线图

### 五、 程序设计（流程图及程序代码）

```
;//***** //  
;// //  
;// 硬布线控制器模型机实验指令文件 //  
;// //  
;// By TangDu CO.,LTD //  
;// //
```

```

; //*****
; //**** Start Of Main Memory Data **** //
$P 00 20 ; START: IN R0 从 IN 单元读入数据送 R0
$P 01 00 ; ADD R0,R0 R0 和自身相加, 结果送 R0
$P 02 30 ; OUT R0 R0 的值送 OUT 单元显示
$P 03 E0 ; JMP START 跳转至 START
$P 04 00 ;
$P 05 50 ; HLT 停机
; //***** End Of Main Memory Data *****/

```

## 六、 实验步骤及实验结果：

(1) 分析每个状态所需的控制信号，并汇总成表，如表 5-2-1 所示。

表 5-2-1 控制信号表

状态号	控制信号
S0	0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
S1	0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 1
S2	0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0
S3	0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0
S4	0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0
S5	0 0 0 1 0 0 1 0 0 1 0 0 0 1 1 1 0
S6	0 1 1 0 0 0 0 0 0 1 0 0 1 1 1 1 0
S7	1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0
S8	0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
S9	0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 1
S10	0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1

控制信号由左至右，依次为：WR，RD，IOM，S3，S2，S1，S0，LDA，LDB，LOAD，LDAR，LDIR，ALU\_B，R0\_B，LDR0，PC\_B，LDPC。

(2) 用 VHDL 语言来设计本实验的状态机，使用 Quartus II 软件编辑 VHDL 文件并进行编译，硬布线控制器在 EP2C5T144C8 芯片中对应的引脚如图 5-2-2 所示（本实验例程见‘安装路径\FPGA\Controllor\Controllor.qpf’工程）

(3) 关闭实验系统电源，按图 5-2-3 连接实验电路。注意：不要将 FPGA 扩展板上的“A09”引脚接至控制总线的“INTA”否则可能导至实验失败。

(4) 打开实验系统电源，将下载电缆插入 FPGA 单元的 JTAG 口，把生成的 SOF 文件下载到 FPGA 单元中去。

(5) 用本实验定义的机器指令系统，可具体编写多种应用程序，下面给出的是本次实验的例程，其程序的

文件名以.TXT 为后缀。程序中分号';'为注释符,分号后面的内容在下载时将被忽略掉

(6) 进入软件界面,装载机器指令,选择菜单命令“【实验】—【简单模型机】”,打开简单模型机数据通路图,按动 CON 单元的总清按钮 CLR,使程序计数器 PC 地址清零,控制器状态机回到 S0,程序从头开始运行,选择相应的功能命令,即可联机运行、监控、调试程序。

(7) 当模型机执行完 JMP 指令后,检查 OUT 单元显示的数是否为 IN 单元值的 2 倍,按下 CON 单元的总清按钮 CLR,改变 IN 单元的值,再次执行机器程序,从 OUT 单元显示的数判别程序执行是否正确。

## 七、 结果分析和讨论:

八、 实验心得: 实验成功或失败的原因,实验后的心得体会、建议等。

# 实验名称: 复杂模型机设计实验

## 一、 实验目的:

综合运用所学计算机组成原理知识,设计并实现较为完整的计算机。

## 二、 实验内容:

## 三、 实验设备:

PC 机一台,TD-CMX 实验系统一套

#### 四、实验线路：

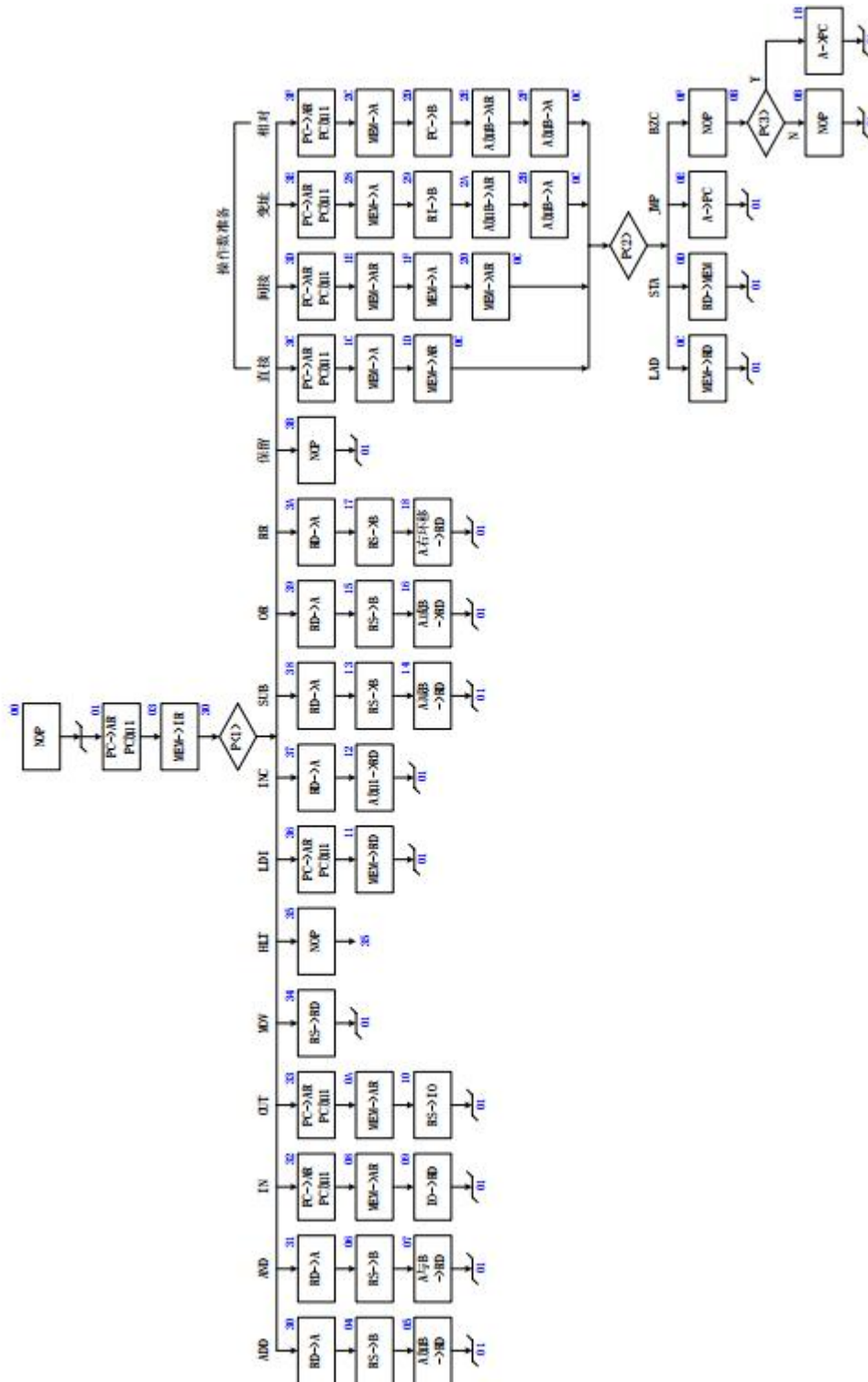


图 5-3-5 微程序流程图



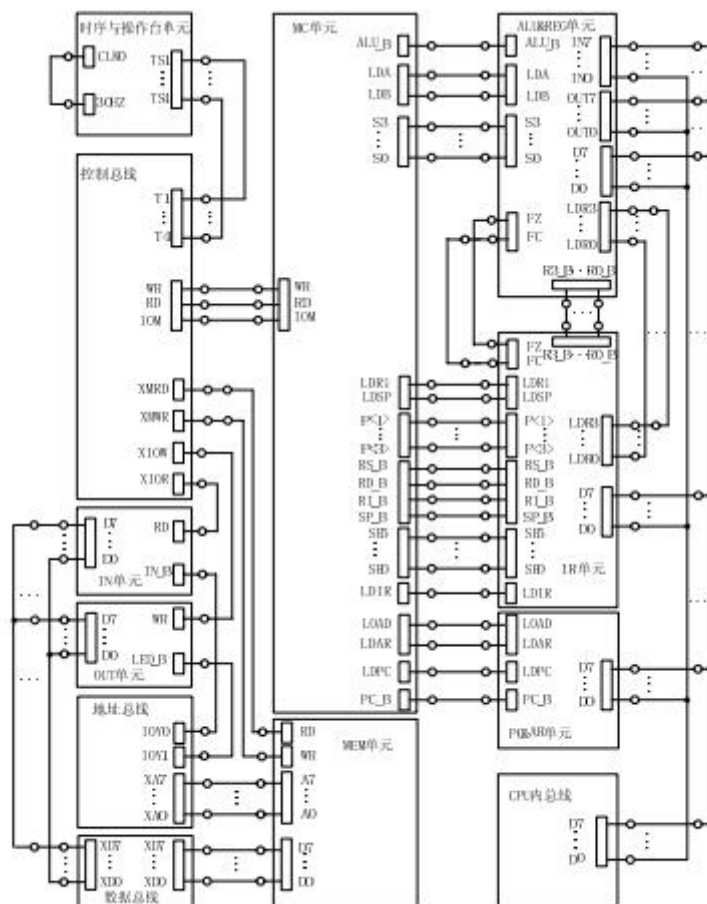


图 5-3-6 实验接线图

## 五、 程序设计（流程图及程序代码）

地 址	内 容	助记符	说 明
00000000	00100000	; START: IN R0, 00H	从 IN 单元读入计数初值
00000001	00000000		
00000010	01100001	; LDI R1, 0FH	立即数 0FH 送 R1
00000011	00001111		
00000100	00010100	; AND R0, R1	得到 R0 低四位
00000101	01100001	; LDI R1, 00H	装入和初值 00H
00000110	00000000		
00000111	11110000	; BZC RESULT	计数值为 0 则跳转
00001000	00010110		
00001001	01100010	; LDI R2, 60H	读入数据始地址
00001010	01100000		
00001011	11001011	; LOOP: LAD R3, [R1], 00H	从 MEM 读入数据送 R3, 变址寻址, 偏移量为 00H
00001100	00000000		
00001101	00001101	; ADD R1, R3	累加求和
00001110	01110010	; INC R1	变址寄存加 1, 指向下一数据
00001111	01100011	; LDI R3, 01H	装入比较值
00010000	00000001		
00010001	10001100	; SUB R0, R3	

00010010	11110000	; BZC RESULT	相减为 0, 表示求和完毕
00010011	00010110		
00010100	11100000	; JMP LOOP	未完则继续
00010101	00001011		
00010110	11010001	; RESULT: STA 70H, R1	和存于 MEM 的 70H 单元
00010111	01110000		
00011000	00110100	; OUT 40H, R1	和在 OUT 单元显示
00011001	01000000		
00011010	11100000	; JMP START	跳转至 START
00011011	00000000		
00011100	01010000	; HLT	停机
01100000	00000001	; 数据	
01100001	00000010		
01100010	00000011		
01100011	00000100		
01100100	00000101		
01100101	00000110		
01100110	00000111		
01100111	00001000		
01101000	00001001		
01101001	00001010		
01101010	00001011		
01101011	00001100		
01101100	00001101		
01101101	00001110		
01101110	00001111		

## 六、 实验步骤及实验结果：

1. 按图 5-3-6 连接实验线路，仔细检查接线后打开实验箱电源。

2. 写入实验程序，并进行校验，分两种方式，手动写入和联机写入。

1) 手动写入和校验

(1) 手动写入微程序

① 将时序与操作台单元的开关 KK1 置为‘停止’档，KK3 置为‘编程’档，KK4 置为‘控存’档，KK5 置为‘置数’档。

② 使用 CON 单元的 SD05——SD00 给出微地址，IN 单元给出低 8 位应写入的数据，连续两次按动时序与操作台的开关 ST，将 IN 单元的数据写到该单元的低 8 位。

③ 将时序与操作台单元的开关 KK5 置为‘加 1’档。

④ IN 单元给出中 8 位应写入的数据，连续两次按动时序与操作台的开关 ST，将 IN 单元的数据写到该单元的中 8 位。IN 单元给出高 8 位应写入的数据，连续两次按动时序与操作台的开关 ST，将 IN 单元的数据写到该单元的高 8 位。

⑤ 重复①、②、③、④四步，将表 5-3-5 的微代码写入 2816 芯片中。

(2) 手动校验微程序

① 将时序与操作台单元的开关 KK1 置为‘停止’档，KK3 置为‘校验’档，KK4 置为‘控存’档，KK5 置为‘置数’档。

② 使用 CON 单元的 SD05——SD00 给出微地址，连续两次按动时序与操作台的开关 ST，MC 单元的指数数据指示灯 M7——M0 显示该单元的低 8 位。

③ 将时序与操作台单元的开关 KK5 置为‘加 1’档。

④ 连续两次按动时序与操作台的开关 ST，MC 单元的指数数据指示灯 M15——M8 显示该单元的中 8 位，MC 单元的指数数据指示灯 M23——M16 显示该单元的高 8 位。

⑤ 重复①、②、③、④四步，完成对微代码的校验。如果校验出微代码写入错误，重新写入、校验，直至确认微指令的输入无误为止。

#### (5) 手动写入机器程序

① 将时序与操作台单元的开关 KK1 置为‘停止’档，KK3 置为‘编程’档，KK4 置为‘主存’档，KK5 置为‘置数’档。

② 使用 CON 单元的 SD7——SD0 给出地址，IN 单元给出该单元应写入的数据，连续两次按动时序与操作台的开关 ST，将 IN 单元的数据写到该存储器单元。

③ 将时序与操作台单元的开关 KK5 置为‘加 1’档。

④ IN 单元给出下一地址（地址自动加 1）应写入的数据，连续两次按动时序与操作台的开关 ST，将 IN 单元的数据写到该单元中。然后地址会又自加 1，只需在 IN 单元输入后续地址的数据，连续两次按动时序与操作台的开关 ST，即可完成对该单元的写入。

⑤ 亦可重复①、②两步，将所有机器指令写入主存芯片中。

#### (6) 手动校验机器程序

① 将时序与操作台单元的开关 KK1 置为‘停止’档，KK3 置为‘校验’档，KK4 置为‘主存’档，KK5 置为‘置数’档。

② 使用 CON 单元的 SD7——SD0 给出地址，连续两次按动时序与操作台的开关 ST，CPU 内总线的指数数据指示灯 D7——D0 显示该单元的数据。

③ 将时序与操作台单元的开关 KK5 置为‘加 1’档。

④ 连续两次按动时序与操作台的开关 ST，地址自动加 1，CPU 内总线的指数数据指示灯 D7——D0 显示该单元的数据。此后每两次按动时序与操作台的开关 ST，地址自动加 1，CPU 内总线的指数数据指示灯 D7——D0 显示该单元的数据，继续进行该操作，直至完成校验，如发现错误，则返回写入，然后校验，直至确认输入的所有指令准确无误。

⑤ 亦可重复①、②两步，完成对指令码的校验。如果校验出指令码写入错误，重新写入、校验，直至确认指令的输入无误为止。

#### 2) 联机写入和校验

联机软件提供了微程序和机器程序下载功能，以代替手动读写微程序和机器程序，但是微程序和机器程序得以指定的格式写入到以 TXT 为后缀的文件中，本次实验程序如下，程序中分号‘；’为注释符，分号后面的内容在下载时将被忽略掉。

## 七、 结果分析和讨论：

**八、 实验心得：**实验成功或失败的原因，实验后的心得体会、建议等。

## **实验要求**

- 一、 实验前按照实验内容认真选好实验设备。
- 二、 实验中遵守实验守则，爱护设备。如遇设备故障，及时向指导老师报告，  
如操作不当造成设备损坏，应承担赔偿责任。
- 三、 实验结束后，将实验设备按编号摆放整齐，经指导老师验收后方可离开。