

版权声明

本用户手册的版权归西安唐都科教仪器开发有限责任公司所有，保留一切权利。
非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本用户手册的部分或全部，并以任何形式传播。

西安唐都科教仪器开发有限责任公司，1999-2008(C)，All right reserved.

TD-CMA 实验系统用户手册

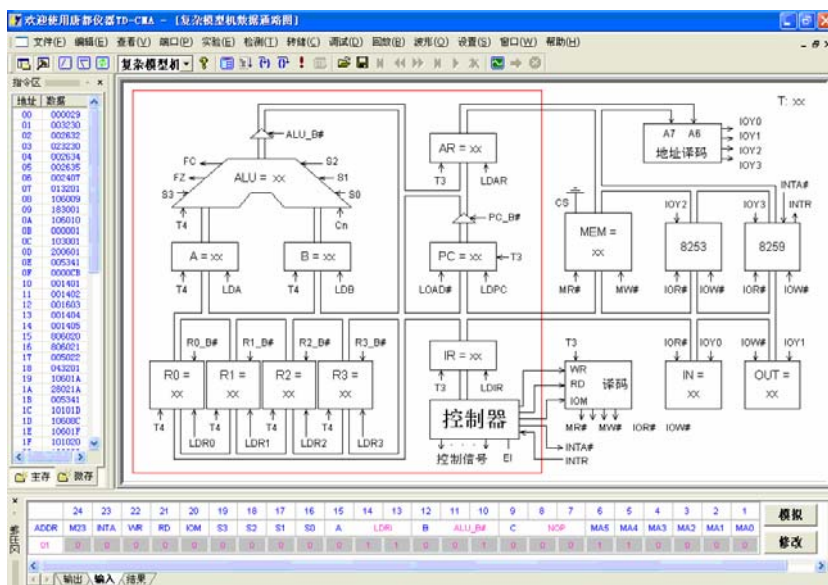
©版权所有 非经许可 严禁复制

技术支持邮箱：tangdukejiao@126.com

唐都公司网址：<http://www.tangdu.com/>

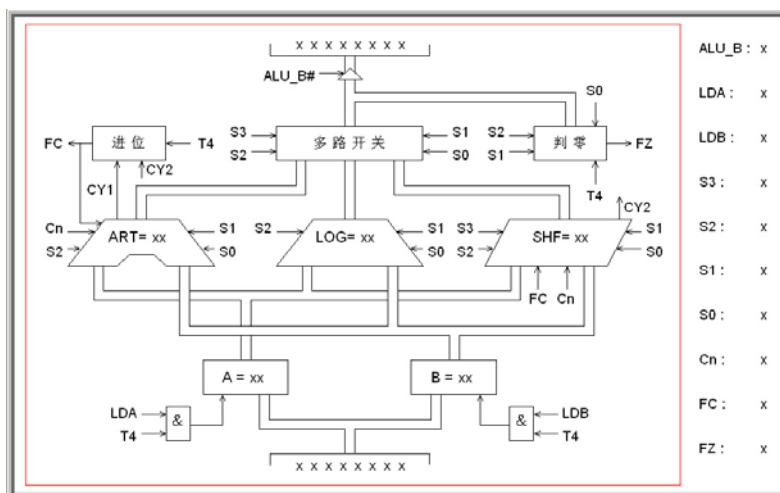
目 录

第 1 章	TD-CMA系统概述	1
1.1	系统功能及特点	1
1.2	系统构成	3
1.3	系统主要实验项目	4
第 2 章	TD-CMA系统的配置与安装	5
2.1	系统配置	5
2.2	系统的安装	6
第 3 章	TD-CMA系统硬件环境	7
3.1	系统硬件布局图	7
3.2	系统电源	7
3.3	系统实验单元电路	8
3.4	注意事项	21
第 4 章	TD-CMA系统集成操作软件	22
4.1	与PC联机说明	22
4.2	软件操作说明	22
第 5 章	TD-CMA系统检测功能说明	32
第 6 章	TD-CMA系统常见故障的分析及处理	34
附录 1	微程序流程图编程方法	35



4. 先进的运算器部件

运算器部件由一片 CPLD 来实现，内含算术、逻辑和移位三个运算部件，其中移位运算采用桶形移位器，各部件独立并行工作，体现了主流运算器设计思想。



5. Cache 控制器部件设计

提供 Cache 高速缓存控制器设计实验，可深刻理解高速缓存的基本原理和设计思想。

6. 开放的控制部件设计

微程序控制器部件由微程序存储器，微命令寄存器，微地址寄存器，微命令译码器，编程电路等构成，其微指令格式和微指令定义可由用户自行设计确定。也可以使用 CPLD 构造组合逻辑控制器，实现计算机硬布线控制器的设计。

7. 先进的系统总线和总线接口设计

系统提供了先进的系统总线结构，与主流的 X86 微机具有相似的系统总线和总线接口设计。实验构建的模型计算机总线接口信号，除数据总线、地址总线外，其控制总线是需要根据计算机功能的要求来设计的，由此便可以开展关于计算机总线接口的设计实验。如：基本输入输出

功能的总线接口设计实验、具有中断控制功能和 DMA 控制功能的总线接口设计实验等。

8. 更为灵活、更为实用的时序发生电路和操作台设计

系统提供的时序发生器其机器周期可以在 2 节拍和 4 节拍之间选择, 这为实验教学提供了更大的灵活性; 系统的本地操作控制台也是全新的设计, 使得系统在独立使用时, 操作起来更为合理、更为方便和实用。

9. 系统电路的保护性设计保证了系统的安全性

系统除了采用了具有抗短路、抗过流的高性能稳压电源来保证产品的安全性外, 还增加了总线竞争报警等多处保护性电路设计, 可进一步保证系统的安全运行。

10. 系统电路检测功能和实验电路查错功能

系统提供了系统电路检测功能和实验电路查错功能, 既可对系统电路进行维护性检测, 还可对实验电路连线的正确与否进行检查, 能够精确检查到用户的每一根实验电路连线。

1.2 系统构成

TD-CMA 实验系统硬件内容如表 1-1 所示。

表 1-1 TD-CMA 系统硬件内容

MC 单元	微程序存储器, 微命令寄存器, 微地址寄存器, 微命令译码器等
ALU® 单元	算术逻辑移位运算部件, A、B 显示灯, 4 个通用寄存器
PC&AR 单元	程序计数器, 地址寄存器
IR 单元	指令寄存器, 指令译码逻辑, 寄存器译码逻辑
CPU 内总线	CPU 内部数据排线座
控制总线	读写译码逻辑, CPU 中断使能寄存器, DMA 控制逻辑
数据总线	LED 显示灯, 数据排线座
地址总线	LED 显示灯, 地址译码电路, 数据排线座
扩展总线	LED 显示灯, 扩展总线排线座
IN 单元	8 位开关, LED 显示灯
OUT 单元	数码管, 数码管显示译码电路
MEM 单元	SRAM6116
8259 单元	8259 一片
8237 单元	8237 一片
8253 单元	8253 一片
CON 单元	3 组 8 位开关, 系统清零按钮
时序与操作台单元	时序发生电路, 555 多谐振荡电路, 单脉冲电路 本地主/控存编程、校验电路, 本地机器调试及运行操作控制电路
SYS 单元	系统监视电路, 总线竞争报警电路
逻辑测量单元	4 路逻辑示波器
扩展单元	LED 显示灯, 扩展接线座
CPLD 扩展板	ALTERA MAX II EPM1270T144C5, 下载电路, LED 显示灯

1.3 系统主要实验项目

1. 运算器及设计实验

- (1) 基本运算器实验
- (2) 超前进位加法器设计实验
- (3) 阵列乘法器设计实验

2. 存储系统及设计实验

- (1) 静态随机存储器实验
- (2) Cache 控制器设计实验

3. 控制器及设计实验

- (1) 时序发生器设计实验
- (2) 微程序控制器实验

4. 系统总线与总线接口实验

- (1) 系统总线和具有基本输入输出功能的总线接口实验
- (2) 具有中断控制功能的总线接口设计
- (3) 具有 DMA 控制功能的总线接口设计

5. 模型计算机的设计实验

- (1) CPU 与简单模型机设计实验
- (2) 硬布线控制器模型机设计实验
- (3) 复杂模型机设计实验

6. 输入、输出系统实验

- (1) 具有中断处理功能的模型计算机设计实验
- (2) 具有 DMA 处理功能的模型计算机设计实验
- (3) 典型 I/O 接口 8253 扩展设计实验

7. 先进计算机结构的设计及研究实验

- (1) 基于 RISC 处理器的模型计算机设计实验
- (2) 基于重叠技术的模型计算机设计实验
- (3) 基于流水技术的模型计算机设计实验

第 2 章 TD-CMA 系统的配置与安装

2.1 系统配置

TD-CMA 实验系统出厂时已全部安装完好，其中的元件配置情况如表 2-1 所示。

表 2-1 TD-CMA 系统的主要配置

项目	内容	数量	项目	内容	数量
微程序控制器	2816	3	程序地址计数器	CPLD	1
	74LS245	5		74LS245	1
	74LS04	1	控制总线	74LS74	3
	74LS74	3		GAL16V8	1
	74LS273	2	IN 单元	拨动开关	8
	74LS175	1		74LS245	2
	74LS138	2	地址总线	74LS139	1
	GAL16V8	1	CON 单元	74LS245	1
	三档开关	1		拨动开关	24
运算器	ALU	1		清零按钮	1
	74LS245	4	CPLD 扩展板	EPM1270T144C5	1
SYS 单元	单片机	1		LED 灯	16
	MAX232	1	指令译码器	GAL20V8	1
	74LS245	5	寄存器译码器	GAL16V8	1
	74LS374	1	8259 单元	8259	1
	74LS138	2	8237 单元	8237	1
	74LS00	1	8253 单元	8253	1
	74LS04	1	扩展单元	LED 灯	8
	74LS32	1	数据总线	74LS245	1
时序单元	三档开关	5	通讯电缆	RS-232C	1
	555	1	下载电缆	ByteBlaster	1
	74LS00	1	机内电源	5V, $\pm 12V$	1
	微动按钮	2	实验用排线		若干
OUT 单元	7 段数码管	2	程序存储器	74LS245	3
	74LS273	1		6116	1
	GAL16V8	2		74LS374	1
集成操作软件		1			

2.2 系统的安装

1. TD-CMA 系统与 PC 微机相连

用 RS-232C 通讯电缆一根，按图 2-1 所示，将 PC 微机串口和 TD-CMA 系统中的串口连接在一起。

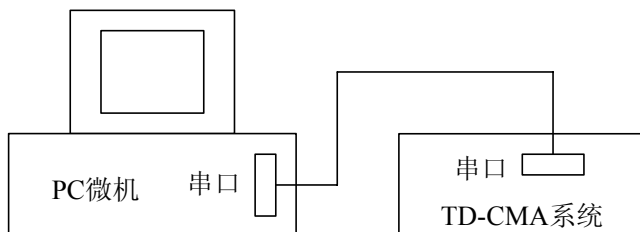


图 2-1 TD-CMA 系统与 PC 微机联机示意图

串行通讯电缆的接线情况如图 2-2 所示：

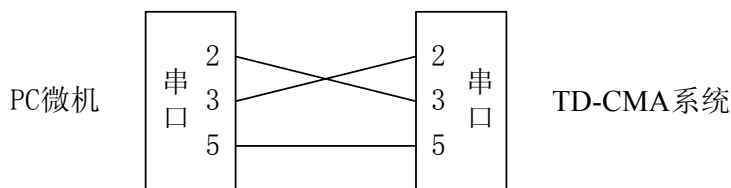


图 2-2 串口连接示意图

2. TD-CMA 系统联机软件的安装

(1) 软件运行环境。

操作系统：Windows 98/NT/2000/XP

最低配置：

CPU：奔腾 300MHz

内存：64MB

显示卡：标准 VGA，256 色显示模式以上

硬盘：20MB 以上

光驱：标准 CD-ROM

(2) 安装软件与运行。

可以通过“资源管理器”，找到光盘驱动器本软件安装目录下的‘安装 CMA.EXE’，双击执行它，按屏幕提示进行安装操作。“TD-CMA”软件安装成功后，在“开始”的“程序”里将出现“CMA”程序组，点击“CMA”即可执行程序。

(3) 卸载软件。

联机软件提供了自卸载功能，使您可以方便地删除“TD-CMA”的所有文件、程序组或快捷方式。单击【开始】/【程序】打开“CMA”的程序组，然后运行“卸载”项，就可执行卸载功能，按照屏幕提示操作即可以安全、快速地删除“TD-CMA”。

第3章 TD-CMA 系统硬件环境

3.1 系统硬件布局图

系统硬件的电路布局是按照计算机组成结构来设计的,如图 3-1 所示,最上面一部分是 SYS 单元,这个单元是非操作区,其余单元均为操作区,在 SYS 单元之上架有 CPLD 扩展板,逻辑测量单元位于 SYS 单元的左侧,时序与操作台单元位于 SYS 单元的右侧。所有构成 CPU 的单元放在中间区域的左边,并标注有‘CPU’,CPU 对外表现的是三总线:控制总线、数据总线和地址总线,三总线并排位于 CPU 右侧。与三总线挂接的主存和各种 IO 设备,都集中放在系统总线的右侧。在实验箱中上部对 CPU、系统总线、主存及外设分别有清晰的丝印标注,通过这三部分的模块可以方便地构造各种不同复杂程度的模型计算机。

系统独立运行时,为了对微控器或是主存进行读写操作,在实验箱下方的 CON 单元中安排了一个开关组 SD07~SD00,专门用来给出主/控存的地址。在进行部件实验时,有很多的控制信号需要用二进制开关模拟给出,所以在实验箱的最下方安排的是控制开关单元 CON 单元。

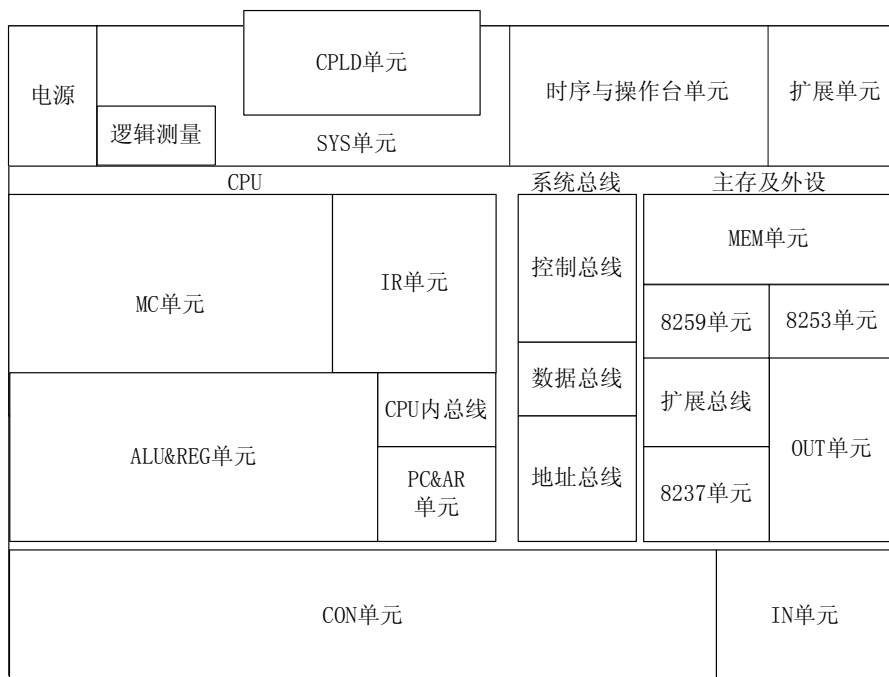


图 3-1 TD-CMA 系统布局图

3.2 系统电源

TD-CMA 系统采用本公司生产的 SP-15 型三路高效开关电源作为系统工作和实验电源,其主要技术指标为:

- (1) 输入电压：AC165—260V
- (2) 输出电压／电流：5V/2A、12V/0.2A、-12V/0.2A
- (3) 输出功率：15W
- (4) 效率： $\geq 75\%$
- (5) 稳压性能：电压调整率 $\leq 0.2\%$
负载调整率 $\leq 0.5\%$
纹波系数 $\leq 0.5\%$
- (6) 工作环境温度： -5°C — 40°C

系统电源已置于电路板下方机箱内，电源开关在电路板的左上角。当关闭电源后，不要立即重新开启，关闭到重新开启之间需要至少 30 秒间隔。

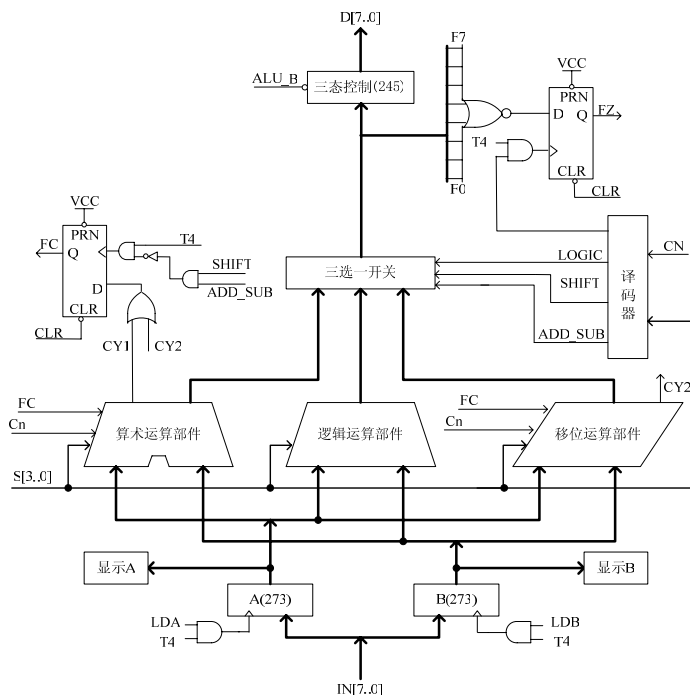


图 3-2 ALU 内部原理图

3.3 系统实验单元电路

1. ALU® 单元

ALU® 单元由以下部分构成：一片 CPLD 实现的 ALU，五片 74LS245 构成的保护电路。ALU 的输出以排针形式引出 D7...D0，ALU 与 REG 的输入以排针形式引出 IN7...IN0，REG 的输出以排针形式引出 OUT7...OUT0，运算器的控制信号（LDA、LDB、S0、S1、S2、S3、ALU_B、CN）分别以排针形式引出，寄存器堆的输入控制信号（LDR0、LDR1、LDR2、LDR3）、寄存器堆的输出控制信号（R0_B、R1_B、R2_B、R3_B）也分别以排针形式引出，另外还有进位标志 FC 和零标志 FZ 指示灯。其 ALU 内部构成如图 3-2 所示，但图中有三部分不在 CPLD 中实现，而是在外围电路中实现，这三部分为图中的‘显示 A’、‘显示 B’和 ALU 的输出控制‘三

态控制 245’，请注意：实验箱上凡丝印标注有马蹄形标记 ‘┌┐’，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，实验箱中所有部件单元的 T1、T2、T3、T4 都已连接至系统总线单元的 T1、T2、T3、T4，CLR 都连接至 CON 单元的 CLR 按钮。

其中暂存器 A 和暂存器 B 中的数据能在 LED 灯上实时显示，本实验箱上所有的 LED 灯均为正逻辑，即 ‘1’ 时亮，‘0’ 时灭，原理如图 3-3 所示（以 A0 为例，其它相同）。本单元中的进位标志 FC 和零标志 FZ 显示原理也是如此。

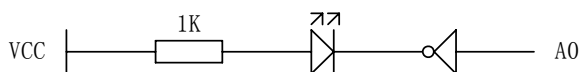


图 3-3 A0 显示原理图

ALU 和 REG 的连接如图 3-4 所示，由于 ALU 的工作电压为 3.3V，所以在所有用户操作的 IO 脚都加上 74LS245 加以隔离保护，以防误操作烧坏 ALU 芯片。

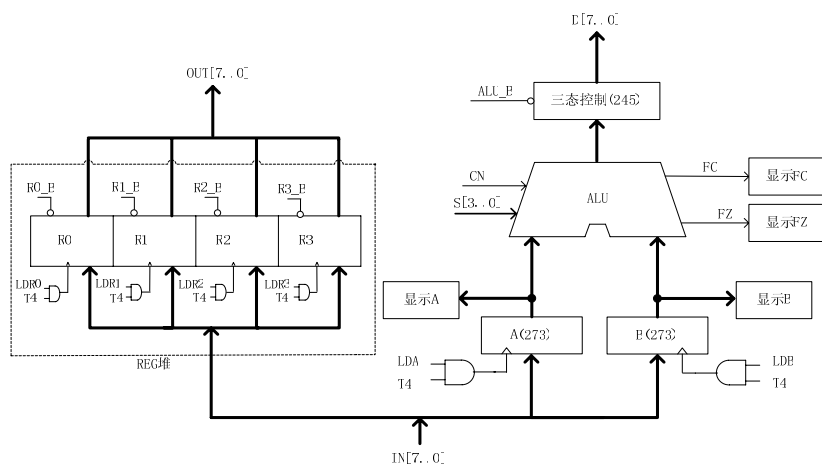


图 3-4 ALU 和 REG 连接原理图

ALU® 单元由运算器和寄存器堆构成，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。寄存器堆由 R0、R1、R2、R3 组成，它们用来保存操作数及中间运算结果等，其中 R2 还兼做变址寄存器，R3 兼做堆栈指针。

2. 程序计数器与地址寄存器单元 (PC&AR 单元)

此单元由地址寄存器 AR、程序计数器 PC 构成。地址寄存器的输出以排针形式引出 A7...A0，其电路原理如图 3-5 所示。

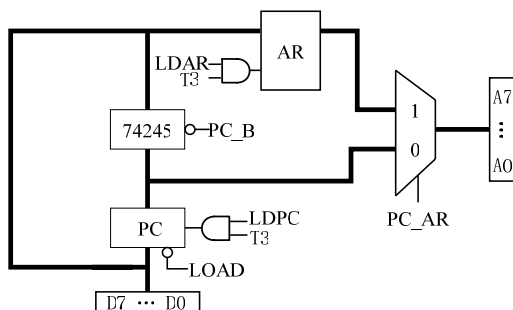


图 3-5 PC&AR 单元原理图

3. 指令寄存器单元 (IR 单元)

IR 单元包括三大部分：指令寄存器、指令译码电路 INS_DEC 和寄存器译码电路 REG_DEC，指令寄存器单元中指令寄存器的输入和输出都以排针形式引出，构成模型机时实现程序的跳转控制和对通用寄存器的选择控制，其电路构成如图 3-6 所示，其中 REG_DEC 由一片 GAL16V8 实现，内部原理如图 3-7 所示，INS_DEC 由一片 GAL20V8 实现，内部原理如图 3-8 所示。

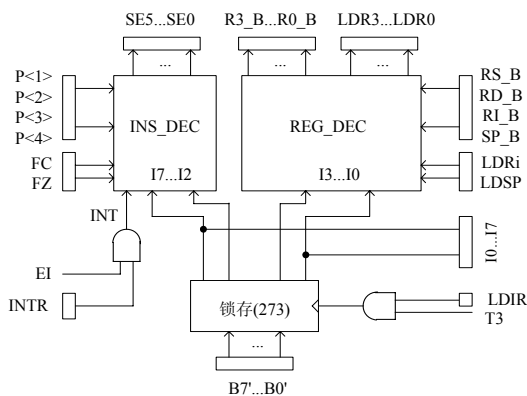


图 3-6 IR 单元原理图

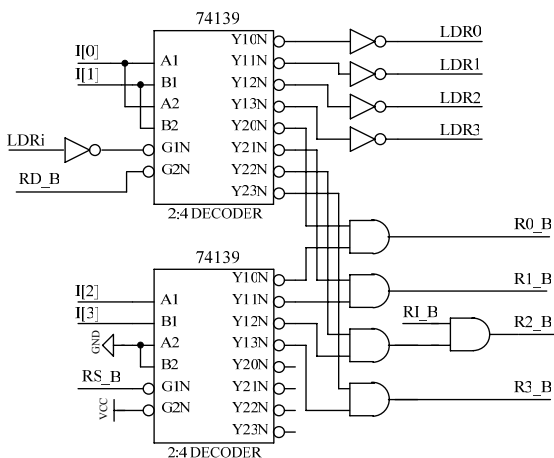


图 3-7 寄存译码原理图

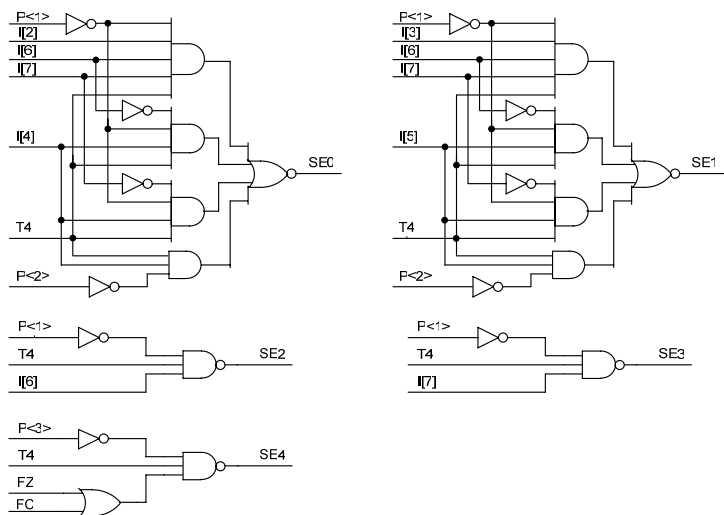


图 3-8 指令译码原理图

4. 微程序控制器电路单元 (MC 单元)

本系统的微控器单元主要由编程部分和核心微控器部分组成，其电路构成如图 3-9 所示。

编程部分是通过编程开关的相应状态选择及由 T2 引入的节拍脉冲来完成将预先定义好的微代码程序写入到 2816 控制存储器中，并可以对控制存储器中的程序进行校验。该系统具有本地直接编程和校验功能，且由于选用 2816 E²PROM 芯片为控制存储器，所以具备掉电保护功能。

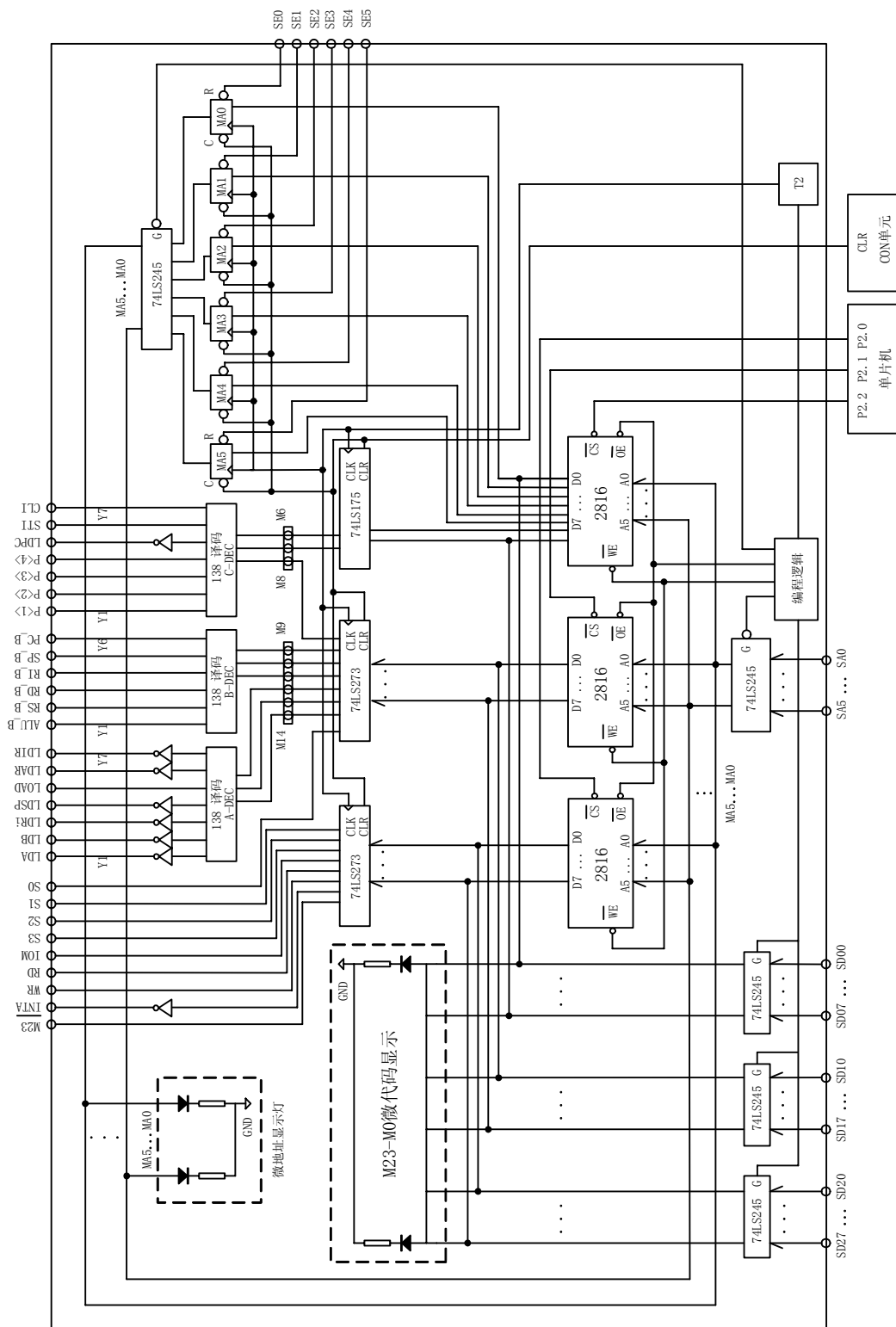
核心微控器主要完成接收机器指令译码器送来的代码，使控制转向相应机器指令对应的首条微代码程序，对该条机器指令的功能进行解释或执行的工作。更具体讲，就是通过接收 CPU 指令译码器发来的信号，找到本条机器指令对应的首条微代码的微地址入口，再通过由 T2 引入的时序节拍脉冲的控制，逐条读出微代码。实验板上的微控器单元中的 24 位显示灯 (M23—M0) 显示的状态即为读出的微指令。然后，其中几位再经过译码，一并产生实验板所需的控制信号，将它们加到数据通路中相应的控制位，可对该条机器指令的功能进行解释和执行。指令解释到最后，再继续接收下一条机器指令代码并使控制转到对应的微地址入口，这样周而复始，即可实现机器指令程序的运行。

核心微控器同样是根据 24 位显示灯所显示的相应控制位，再经部分译码产生的二进制信号来实现机器指令程序顺序、分支、循环运行的，所以，有效地定义 24 位微代码对系统的设计至关重要。

(1) 核心微控器单元

通过编程开关的不同状态，可进行微代码的编程、校验、运行。在本单元电路原理图中：

- 微地址显示灯显示的是后续微地址，而 24 位显示灯显示的是后续地址的二进制控制位。
- T2 为微地址锁存器 (74LS74) 的时钟信号。
- 2816 的片选信号 (CS) 在手动状态下一直为 '0'，而在和 PC 联机状态下，受 89S51 控制。



CLR 为清零信号的引出端，实验板中已接至 CON 单元中最右边的 CLR 按钮上，此二进制按钮为 CLR 专用。SE5-SE0 端挂接到 CPU 的指令译码器的输出端，通过译码器确定相应机器指令的微代码入口，也可人为手动模拟 CPU 的指令译码器的输出，达到同一目的。

5. CPU 内总线单元

此单元由五排 8 线排针组成，它们之间相应位是相互连通的，CPU 内总线是 CPU 内部数据集散地，每个部件的输入数据来自于 CPU 内总线，输出的数据也要通过 CPU 内总线到达目的地。

6. 时序与操作台单元

时序单元可以提供单脉冲或连续的时钟信号：KK 和 Φ 。其中的 Q 为 555 构成的多谐振荡器的输出，其原理如附图 3-10 所示，经分频器分频后输出频率大约为 30Hz、300Hz、占空比为 50%的 Φ 信号。

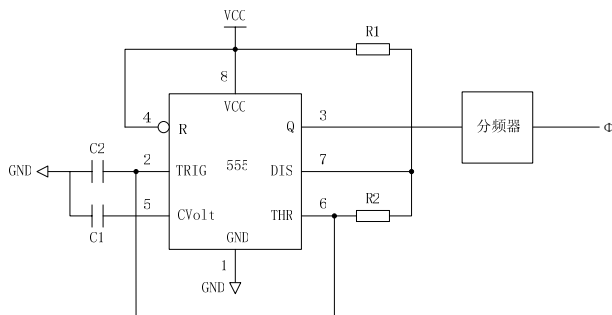


图 3-10 555 多谐振荡器原理图

时序与操作台单元的“MODE”短路块短路，系统工作在四节拍模式；“MODE”短路块拨开，系统工作在两节拍模式。

时序与操作台单元的“SPK”短路块短路，系统具有总线竞争报警功能；“SPK”短路块拨开，系统无报警功能。

每按动一次 KK 按钮，在 KK+ 和 KK- 端将分别输出一个上升沿和下降沿单脉冲。其原理如图 3-11 所示：

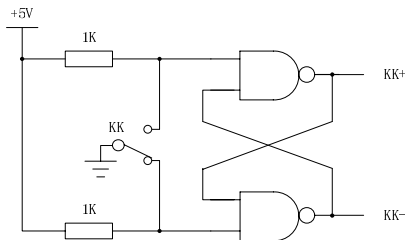


图 3-11 KK 单脉冲电路原理图

每按动一次 ST 按钮，根据时序开关档位的不同，在 TS1、TS2、TS3、TS4 端输出不同的波形。当开关处于‘连续’档时，TS1、TS2、TS3、TS4 输出的是图 3-12 所示的连续时序。开关处于‘单步’档时，TS1、TS2、TS3、TS4 只输出一个 CPU 周期的波形，如图 3-13。开关处于‘单拍’档时，TS1、TS2、TS3、TS4 交替出现，如图 3-14 所示。

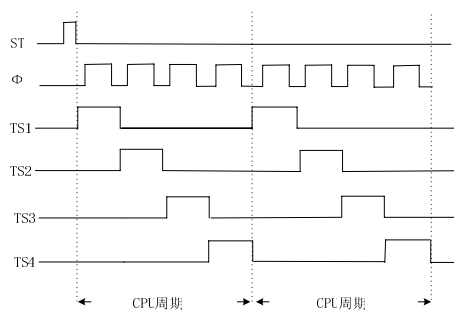


图 3-12 连续时序

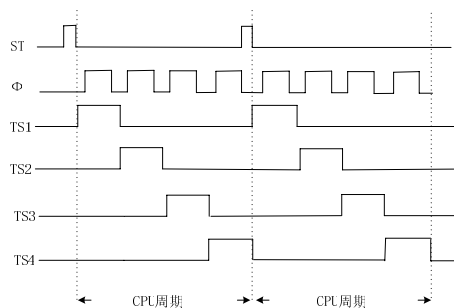


图 3-13 单步时序

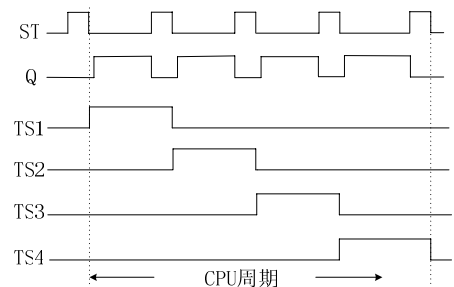
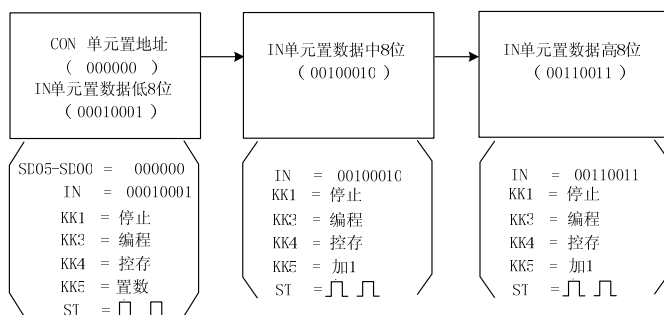


图 3-14 单拍时序

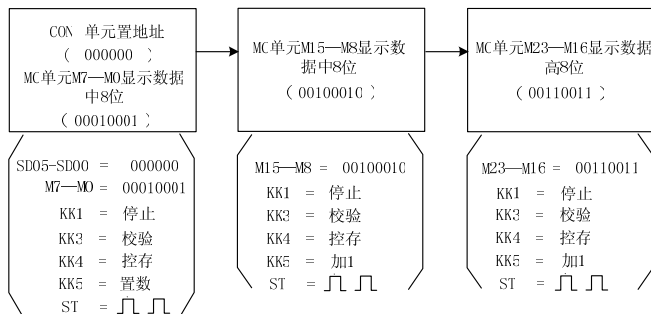
当 TS1、TS2、TS3、TS4 输出连续波形时，有四种方法可以停止输出：将时序状态开关 KK1 拨至停止挡、将 KK2 打到‘单拍’或‘单步’档、按动 CON 单元的 CLR 按钮或是系统单元的复位按钮。CON 单元的 CLR 按钮和 SYS 单元的复位按钮的区别是，CLR 按钮完成对各实验单元清零，复位按钮完成对系统及时序发生器复位。

在实验平台中设有一组编程控制开关 KK1、KK2、KK3、KK4、KK5（位于时序与操作台单元），可实现对存储器（包括程序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（包括程序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。



同样的，操作控制开关 KK1、KK2、KK3、KK4、KK5 (KK4 拨至‘主存’档)，可实现对存储器的操作，手动操作存储器时，将 PC&AR 单元的 D0——D7 用排线接到 CPU 内总线的 D0——D7，这样可以在地址总线的地址灯上看到操作的地址。

7. 输入设备单元 (IN 单元)

此单元使用 8 个拨动开关作为输入设备, 其电路原理如图 3-15 所示, 左边表示的是 IN 单元的整体连接原理, 右边表示的是一个拨动开关的连接原理, 拨动开关采用的是双刀双掷开关, 一刀用来输出数据, 一刀用来在 LED 灯上显示开关状态。

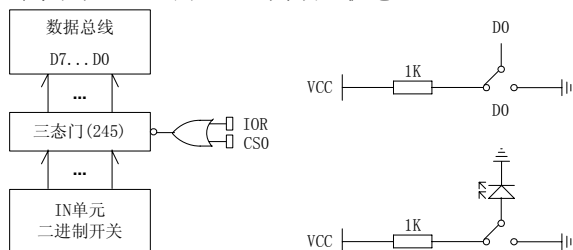


图 3-15 IN 单元原理图

8. 输出设备单元 (OUT 单元)

在 OUT 单元, 数据由锁存器 74LS273 进行锁存, 并通过两片 GAL16V8 进行显示译码, 形成数码管显示的驱动信号, 具体电路原理如图 3-16 所示。

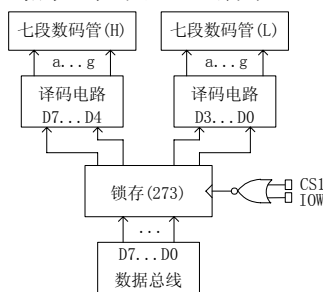


图 3-16 OUT 单元原理图

9. 控制总线单元

此单元包含有 CPU 对存储器和 IO 进行读写时的读写译码电路 (这一电路在 GAL16V8 中实现, 如图 3-17 所示)、CPU 中断使能寄存器 (如图 3-18 所示)、外部中断请求指示灯 INTR、CPU 中断使能指示灯 EI。

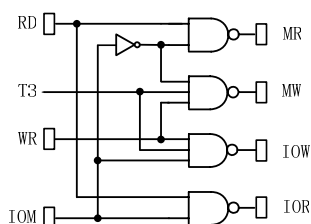


图 3-17 读写译码原理

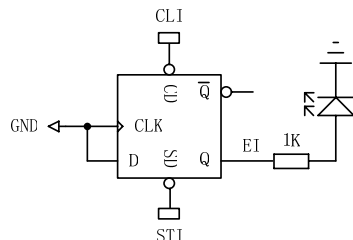


图 3-18 CPU 中断使能寄存器

10. 数据总线单元

数据总线是 CPU 和主存以及外设之间数据交换的通道，其包含五排 8 线排针，排针的相应位已和 CPU 内总线连通。

11. 地址总线单元

此单元由两排 8 线排针，I/O 地址译码芯片 74LS139，地址指示灯组成。两排 8 线排针已连通，为了选择 I/O，产生 I/O 片选信号，还需要进行 I/O 地址译码，系统的 I/O 地址译码原理见图 3-19 所示。

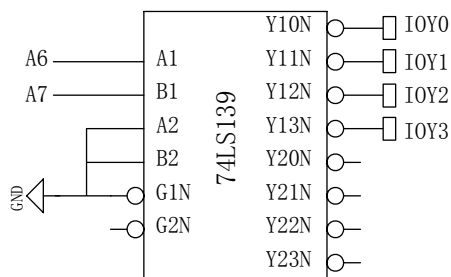


图 3-19 I/O 地址译码原理图

由于用的是地址总线的高两位进行译码，I/O 地址空间被分为四个区，如表 3-1 所示，地址指示显示原理同图 3-18 中 EI 的显示。

表 3-1 I/O 地址空间分配

A7	A6	选定	地址空间
00		IOY0	00-3F
01		IOY1	40-7F
10		IOY2	80-BF
11		IOY3	C0-FF

12. 存储器单元 (MEM 单元)

存储器单元包括一片 SRAM 6116（静态随机存储器）和一套编程电路，如图 3-20 所示，因为要对存储器手动进行读写，所以设计了对存储器的读写电路。

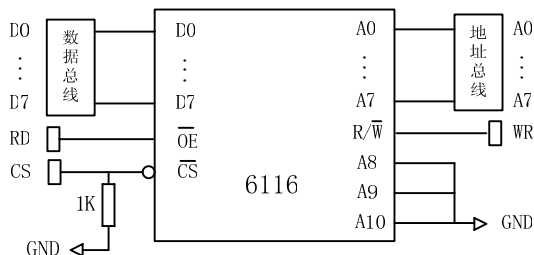


图 3-20 存储器原理图

13. 8253 单元

此单元由一片 8253 构成，数据线、地址线、信号线均以排线引出，8253 的三个通道均开放出来，其中 GATE0 接到高电平，如图 3-21 示。

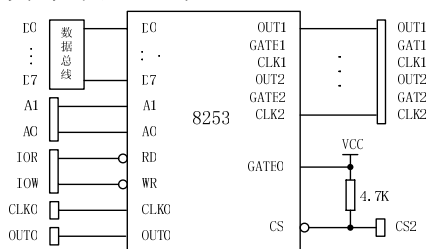


图 3-21 8253 连接图

14. 8259 单元

此单元由一片 8259 构成，数据线、地址线、信号线均以排线引出，如图 3-22 示。

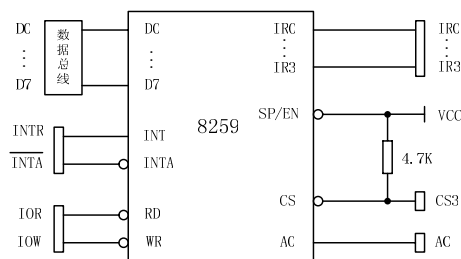


图 3-22 8259 连接图

15. 8237 单元

此单元由一片 8237 构成，数据线、地址线、信号线均以排线引出，如图 3-23 示。

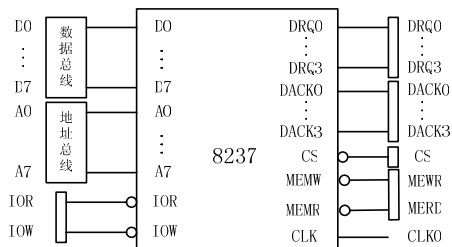


图 3-23 8237 连接图

16. 控制台开关单元 (CON 单元)

此单元包含一个系统总清按钮 CLR 和 24 个双刀双掷开关，开关分成三组，分别为：SD27-SD20、SD17-SD10 和 SD07-SD00。有部分开关有双重丝印，为的是方便接线，一个开关可能对应两个排针，根据丝印就能找到开关和排针的对应关系。开关为双刀双掷，一刀用来提供数据，一刀用来显示开关值，其原理如图 3-24 示（以 SD20 为例，其它相同）。

CLR 按钮连接如图 3-25 示，平时为高，按下后 CLR 输出变为低，为系统部件提供清零信

号，按下 CLR 按钮后会清零的部件有：程序计数器 PC、地址寄存器 AR、暂存器 A、暂存器 B、指令寄存器 IR、微地址寄存器 MAR。

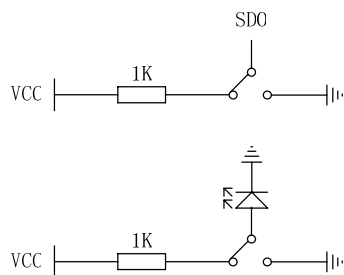


图 3-24 双刀双掷开关原理图

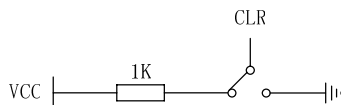


图 3-25 CLR 按钮原理图

17. 扩展单元

此单元由 8 个 LED 显示灯，电源（+5V）和地排针，以及 3 排 8 线排针组成。8 线排针相应位已连通，主要是为了电路转接而设计。LED 灯电路如图 3-26（以 E0 为例，其余相同）。

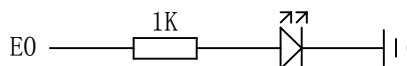


图 3-26 ED 灯显示原理图

18. CPLD 扩展板

TD-CMA 的部分实验在 CPLD 扩展板上进行，CPLD 扩展板由两大部分组成，一是 LED 显示灯，两组 16 只，供调试时观测数据，LED 灯连接原理同图 3-26 中 LED 显示，另外就是一片 MAXII EPM1270T144 及其外围电路。

EPM1270T144 有 144 个引脚，分成四个块，即 BANK1...BANK4，将每个块的通用 IO 脚加以编号，就形成 A01...A24、B01...B30 等 IO 号，见图 3-27 所示。扩展板上排针的丝印分为两部分，一是 IO 号，以 A、B、C、D 打头，如 A15，一是芯片引脚号，是纯数字，如 21。在 Quartus II 软件中分配 IO 时用的是引脚号，而在实验接线图中，都以 IO 号来描述。

EPM1270T144 共有 116 个 IO 脚，本扩展板引出 110 个，其中 60 个装有排针，其余 50 个以预留形式给出，在扩展板上丝印标为 JP，JP 座的 IO 分配见图 3-28。

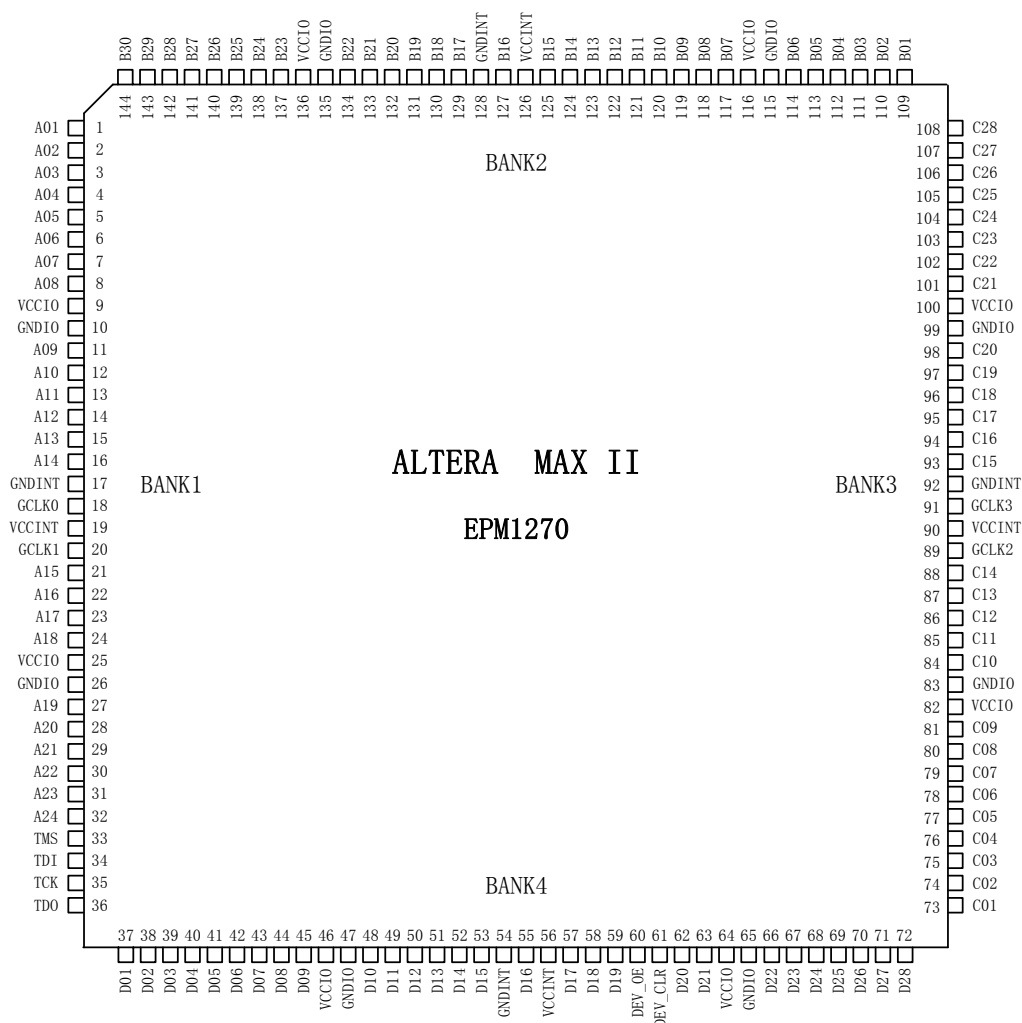


图 3-27 EMP1270 引脚分配图



图 3-28 JP 座 IO 分配图

19. 逻辑测量单元

此单元包含四路逻辑示波器 CH3-CH0，四路示波器的电路一样，如图 3-29（以 CH0 为例）。通过四路探笔，可以测得被测点逻辑波形，在软件界面中显示出来。

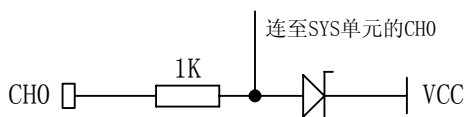


图 3-29 CH0 采样电路

20. 系统单元 (SYS 单元)

此单元是为了和 PC 联机而设计，其原理是通过单片机的串口和 PC 机的串口相连，PC 以命令形式和单片机进行交互，当单片机接收到某命令后，产生相应的时序，实现指定操作。SYS 单元还安排了一个检测电路，当总线上数据发生竞争时，蜂鸣器会发出‘嘀’警报声。SYS 单元还有一个重要职责：当 ST 按钮按下时会对单片机的 INT1 产生一个中断请求，此时单片机根据时序单元状态开关的档位，产生相应的时序。逻辑示波器启动后，单片机会定期采样 CH3-CH0，图 3-33 中的‘连至 SYS 单元的 CH0’线就是单片机采样通道，并将采样所得数据通过串口发送到 PC 机，PC 机再根据收到的数据，在屏幕上绘制波形。

3.4 注意事项

- (1) 使用前后均应仔细检查主机板，防止导线、元件等物品落入装置内导致线路短路、元件损坏。
- (2) 电源线应放置在机内专用线盒中。
- (3) 注意系统的日常维护，经常清理灰尘和杂物。
- (4) 电源关闭后，不能立即重新开启，关闭与重新开启之间至少应有 30 秒间隔。

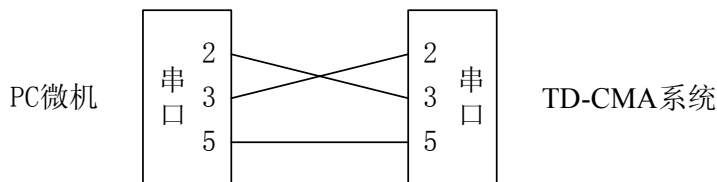
第4章 TD-CMA 系统集成操作软件

4.1 与 PC 联机说明

TD-CMA 计算机实验系统安装有一个标准的 D 型 9 针 RS-232C 串口插座, 使用随机配套的串行通讯电缆分别插在 TD-CMA 及 PC 微机的串口, 即可实现系统与 PC 的联机操作。

本系统软件是通过 PC 机串行口向 TD-CMA 上的单片机控制单元发送指令, 从而使用单片机直接对程序存储器、微程序控制器进行读写, 并可实现单步微程序, 单步机器指令和程序连续运行等操作。

系统与 PC 微机采用的通讯协议规定如下: 57600 波特, 8 位数据位, 1 位停止位, 无校验位, 通讯电缆连接方式如下:



4.2 软件操作说明

(一) 界面窗口介绍

主界面如图 4-1 所示, 由指令区、输出区和图形区三部分组成。

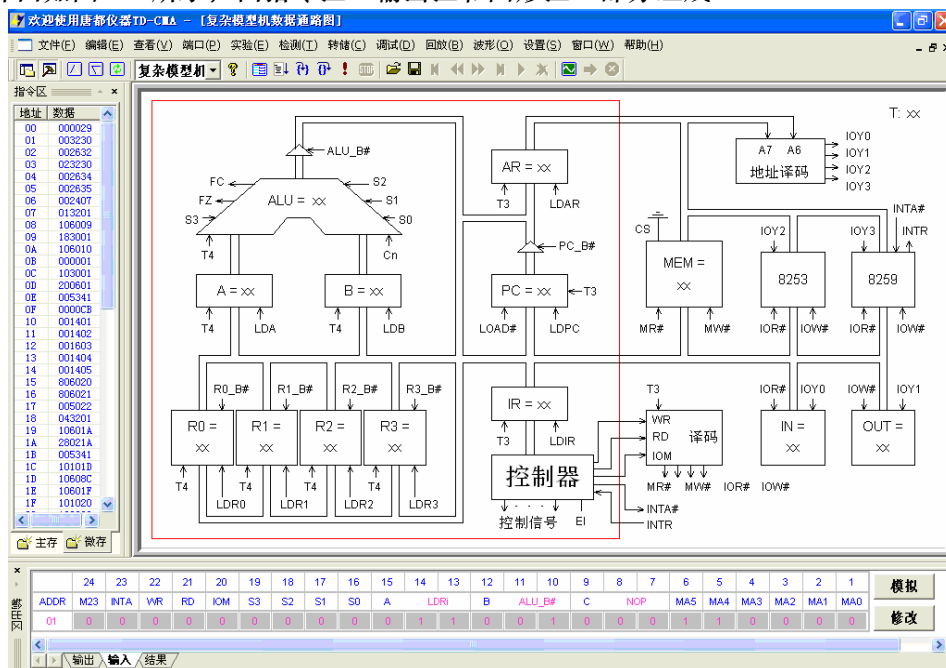


图 4-1 软件主界面

指令区：

分为机器指令区和微指令区，指令区下方有两个 Tab 按钮，可通过按钮在两者之间切换。

机器指令区：分为两列，第一列为主存地址（00—FF，共 256 个单元），第二列为每个地址所对应的数值。串口通讯正常且串口无其它操作，可以直接修改指定单元的内容，用鼠标单击要修改单元的数据，此时单元格会变成一个编辑框，即可输入数据，编辑框只接收两位合法的 16 进制数，按回车键确认，或用鼠标点击别的区域，即可完成修改工作。按下 ESC 键可取消修改，编辑框会自动消失，恢复显示原来的值，也可以通过上下方向键移动编辑框。

微指令区：分为两列，第一列为微控器地址（00—3F，共 64 个单元），第二列为每个地址所对应的微指令，共 6 字节。修改微指令操作和修改机器指令一样，只不过微指令是 6 位，而机器指令是 2 位。

输出区：

输出区由输出页、输入页和结果页组成。

输出页：在数据通路图打开，且该通路中用到微程序控制器，运行程序时，输出区用来实时显示当前正在执行的微指令和下条将要执行的微指令的 24 位微码及其微地址。当前正在执行微指令的显示可通过菜单命令“【设置】—【当前微指令】”进行开关。

输入页：可以对微指令进行按位输入及模拟，鼠标左键单击 ADDR 值，此时单元格会变成一个编辑框，即可输入微地址，输入完毕后回车，编辑框消失，后面的 24 位代表当前地址的 24 位微码，微码值用红色显示，鼠标左键单击微码值可使该值在 0 和 1 之间切换。在数据通路图打开时，按动‘模拟’按钮，可以在数据通路中模拟该微指令的功能，按动‘修改’按钮则可以将当前显示的微码值下载到下位机。

结果页：用来显示一些提示信息或错误信息，保存和装载程序时会在这区域显示一些提示信息。在系统检测时，也会在这区域显示检测状态和检测结果。

图形区：

可以在此区域编辑指令，显示各个实验的数据通路图、示波器界面等。

(二) 菜单功能介绍**1. 文件菜单项**

文件菜单提供了以下命令：

① 新建 (N)：

在 CMA 中建立一个新文档。在文件新建对话框中选择您要建立的新文件的类型。

② 打开 (O)：

在一个新的窗口中打开一个现存的文档。您可同时打开多个文档。您可用窗口菜单在多个打开的文档中切换。

③ 关闭 (C)：

关闭包含活动文档的所有窗口。CMA 会建议您在关闭文档之前保存对您的文档所做的改动。如果您没有保存而关闭了一个文档，您将会失去自从您最后一次保存以来所做的所有改动。在关

新建 (N)	Ctrl+N
打开 (O)...	Ctrl+O
关闭 (C)	
保存 (S)	Ctrl+S
另存为 (A)...	
打印 (P)...	Ctrl+P
打印预览 (V)	
打印设置 (E)...	
最近文件	
退出 (X)	

闭一无标题的文档之前，CMA 会显示另存为对话框，建议您命名和保存文档。

④ 保存 (S)

将活动文档保存到它的当前的文件名和目录下。当您第一次保存文档时，CMA 显示另存为对话框以便您命名您的文档。如果在保存之前，您想改变当前文档的文件名和目录，您可选用另存为命令。

⑤ 另存为 (A) ...

保存并命名活动文档。CMA 会显示另存为对话框以便您命名您的文档。

⑥ 打印 (P) ...

打印一个文档。在此命令提供的打印对话框中，您可以指明要打印的页数范围、副本数、目标打印机，以及其它打印机设置选项。

⑦ 打印预览 (V)

按要打印的格式显示活动文档。当您选择此命令时，主窗口就会被一个打印预览窗口所取代。这个窗口可以按它们被打印时的格式显示一页或两页。打印预览工具栏提供选项使您可选择一次查看一页或两页，在文档中前后移动，放大和缩小页面，以及开始一个打印作业。

⑧ 打印设置 (R) ...

选择一台打印机和一个打印机连接。在此命令提供的打印设置对话框中，您可以指定打印机及其连接。

⑨ 最近使用文件

您可以通过此列表，直接打开最近打开过的文件，共四个。

⑩ 退出 (X)

结束 CMA 的运行阶段。您也可使用在应用程序控制菜单上的关闭命令。

2. 编辑菜单项

① 撤销 (U)

撤销上一步编辑操作。

② 剪切 (T)

将当前被选取的数据从文档中删除并放置于剪贴板上。如当前没有数据被选取时，此命令则不可用。

③ 复制 (C)

将被选取的数据复制到剪切板上。如当前无数据被选取时，此命令则不可用。

④ 粘贴 (P)

将剪贴板上内容的一个副本插入到插入点处。如剪贴板是空的，此命令则不可用。

撤销 (U)	Ctrl+Z
剪切 (T)	Ctrl+X
复制 (C)	Ctrl+C
粘贴 (P)	Ctrl+V

3. 查看菜单项

查看菜单提供了以下命令：

① 工具栏 (I)

显示和隐藏工具栏，工具栏包括了 CMA 中一些最普通命令的按钮。当工具栏被显示时，在菜单项目的旁边会出现一个打勾记号。

② 指令区 (W)

显示和隐藏指令区，当指令区被显示时，在菜单项目的旁边会出现一个打勾记号。

<input checked="" type="checkbox"/>	工具栏 (I)
<input checked="" type="checkbox"/>	指令区 (W)
<input checked="" type="checkbox"/>	输出区 (O)
<input checked="" type="checkbox"/>	状态栏 (S)

③ 输出区 (O)

显示和隐藏输出区，当输出区被显示时，在菜单项目的旁边会出现一个打勾记号。

④ 状态栏 (S)

显示和隐藏状态栏。状态栏描述了被选取的菜单项目或被按下的工具栏按钮，以及键盘的锁定状态将要执行的操作。当状态栏被显示时，在菜单项目的旁边会出现一个打勾记号。

4. 端口菜单项

端口菜单提供了以下命令：

① 端口选择...

选择通讯端口，选择该命令时会弹出图 4-2 所示对话框。该命令会自动检测当前系统可用的串口号，并列于组合框中，选择某一串口后，按确定键，对选定串口进行初始化操作，并进行联机测试，报告测试结果，如果联机成功，则会将指令区初始化。

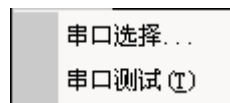


图 4-2 串口选择对话框

② 端口测试 (T)

对当前选择的串口进行联机通讯测试，并报告测试结果，只测一次，如果联机成功，则会将指令区初始化。如串不能正常初始化，此命令则不可用。

5. 实验菜单项



实验菜单提供了以下命令：

① 运算器实验

打开运算器实验数据通路图，如果该通路图已经打开，则把通路激活并置于最前面显示。

② 存储器实验

打开存储器实验数据通路图，如果该通路图已经打开，则把通路激活并置于最前面显示。

③ 微控器实验

打开微控器实验数据通路图，如果该通路图已经打开，则把通路激活并置于最前面显示。

④ 简单模型机

打开简单模型机数据通路图，如果该通路图已经打开，则把通路激活并置于最前面显示。

⑤ 复杂模型机

打开复杂模型机数据通路图，如果该通路图已经打开，则把通路激活并置于最前面显示。

⑥ RISC 模型机

打开 RISC 模型机数据通路图，如果该通路图已经打开，则把通路激活并置于最前面显示。

⑦ 重叠模型机

打开重叠模型机数据通路图，如果该通路图已经打开，则把通路激活并置于最前面显示。

⑧ 流水模型机

打开流水模型机数据通路图，如果该通路图已经打开，则把通路激活并置于最前面显示。

6. 检测菜单项

检测菜单提供了以下命令：

① 连线检测 (C)

(a) 简单模型机

对简单模型机的连线进行检测，并在‘输出区’的‘结果页’显示相关信息。

(b) 复杂模型机

对复杂模型机的连线进行检测，并在‘输出区’的‘结果页’显示相关信息。

② 系统检测 (T) ...

启动系统检测，可以进行部件或是整机检测。

③ 停止检测 (S)

停止系统检测。



7. 转储菜单项

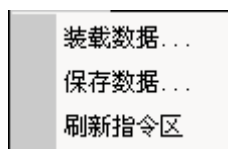
转储菜单提供了以下命令：

① 装载数据...

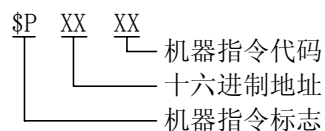
将上位机指定文件中的数据装载到下位机中，您选择该命令会弹出打开文件对话框。

可以打开任意路径下的*.TXT 文件，如果指令文件合法，系统将把这些指令装载到下位机中，装载指令时，系统提供了一定的检错功能，如果指令文件中有错误的指令，将会导致系统退出装载，并提示错误的指令行。

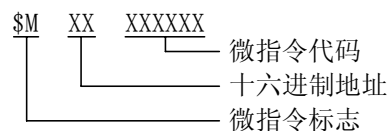
指令文件中指令书写格式如下：



机器指令格式说明：



微指令格式说明：



例如机器指令\$P00FF, “\$”为标记号, “P”代表机器指令, “00”为机器指令的地址, “FF”为该地址中的数据。微指令\$M00AA77FF, “\$”为标记号, “M”代表微指令, “00”为机器指令的地址, “AA77FF”为该地址中的数据。

② 保存数据…

将下位机中（主存，微控器）的数据保存到上位机中，选择该命令会弹出一个保存对话框，如图 4-3 所示：

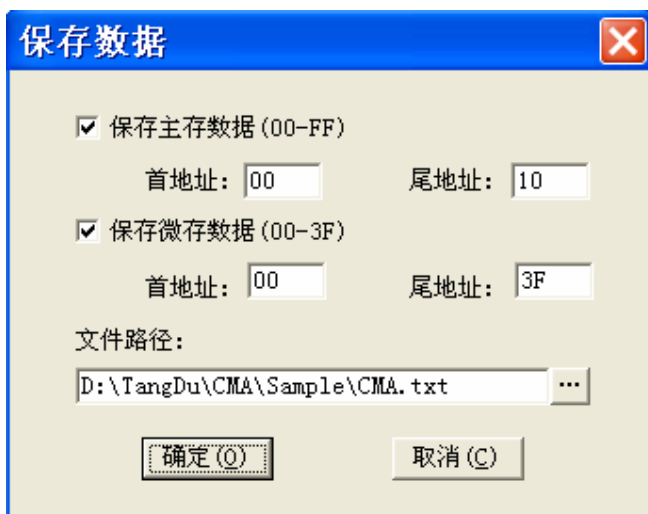


图 4-3 保存数据对话框

可以选择保存机器指令，此时首尾地址输入框将会变亮，否则首尾地址输入框将会变灰，在允许输入的情况下您可以指定需要保存的首尾地址，微指令也是如此，数据到保存指定路径的*.TXT 格式文件中。

③ 刷新指令区

从下位读取所有机器指令和微指令，并在指令区显示。

8. 调试菜单项

调试菜单提供了以下命令：

① 微程序流程图…

当微控器实验、简单模型机和综合性实验中任一数据通路图打开时，可用此命令来打开指定的微程序流程图，选择该命令会弹出打开文件对话框。

② 单节拍

向下位机发送单节拍命令，下位机完成一个节拍的工作。



③ 单周期

向下位机发送单周期命令，下位机完成一个机器周期的工作。

④ 单步机器指令

向下位机发送单步机器指令命令，下位机运行一条机器指令。

⑤ 连续运行

向下位机发送连续运行命令，下位机将会进入连续运行状态。

⑥ 停止运行

如果下位机处于连续运行状态，此命令可以使得下位机停止运行。

9. 回放菜单项

回放菜单提供了以下命令：

① 打开…

打开现存的数据文件。

② 保存…

保存当前的数据到数据文件。

③ 首端

跳转到首页。

④ 向前

向前翻一页。

⑤ 向后

向后翻一页。

⑥ 末端

跳转到末页。

⑦ 播放

连续向后翻页。

⑧ 停止播放

停止连续向后翻页。

10. 波形菜单项

波形菜单提供了以下命令：

① 打开 (O)

打开示波器窗口。

② 运行 (R)

启动示波器，如果下位机正运行程序则不启动。

③ 停止 (S)

停止处于启动状态的示波器。

11. 设置菜单项

设置菜单提供了以下命令：

① 流动速度 (L) …

打开…

保存…

首端

向前

向后

末端

播放

停止播放

打开 (O)

运行 (R)

停止 (S)

流动速度 (L) …

系统颜色 (C) …



当前微指令

设置数据通路图中数据的流动速度，选择该命令会弹出一个流动速度设置对话框，如图 4-4 所示。拖动滑动块至适当位置，点击‘确定’按钮即完成设置。

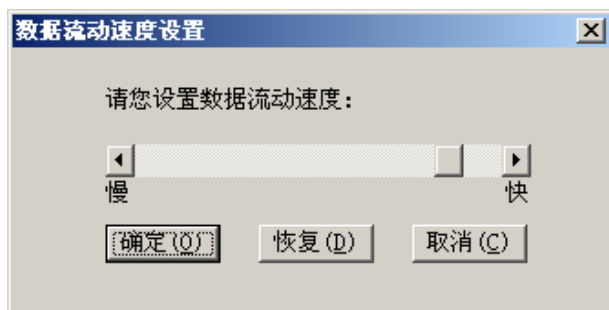


图 4-4 流动速度设置对话框

② 系统颜色 (C) ...

设置数据通路图、微程序流程图和示波器的显示颜色，选择该命令会弹出一个设置对话框，如图 4-5 所示。



图 4-5 系统颜色设置对话框

分为三页，分别为通路图、微流图和示波器，按动每页的 TAB 按钮，可在三页之间切换。选择某项要设置的对象，然后按下‘更改’按钮，或直接用鼠标左键点击要设置对象的颜色框，可弹出颜色选择对话框，选定好颜色后，点击‘应用’按钮相应对象的颜色就会被修改掉。

③ 当前微指令

设置‘输出区’的‘输出页’是否显示当前微指令，当前微指令用灰色显示，并在地址栏标记为‘C’，下条将要执行的微指令标记为‘N’。

12. 窗口菜单项

窗口菜单提供了以下命令。这些命令使您能在应用程序窗口中安排多个文档的多个视图：

① 新建窗口 (N)

打开一个具有与活动的窗口相同内容的新窗口。您可同时打开数个文档窗口以显示文档的不同部分或视图。如果您对一个窗口的内容做了改动，所有其它包含同一文档的窗口也会反映出这些改动。当您打开一个新的窗口，这个新窗口就成了活动的窗口并显示于所有其它打开窗口之上。

② 层叠 (C)

按相互重叠形式来安排多个打开的窗口。

③ 平铺 (T)

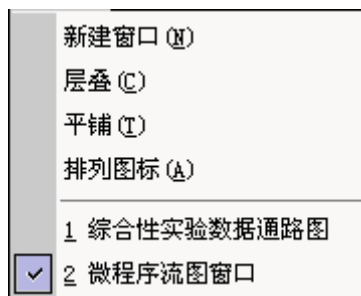
按互不重叠形式来安排多个打开的窗口。

④ 排列图标 (A)

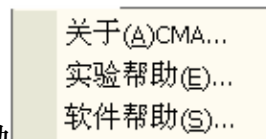
在主窗口的底部安排被最小化的窗口的图标。如果在主窗口的底部有一个打开的窗口，则有可能看不见某些或全部图标，因为它们在这个文档窗口的下面。

⑤ 窗口选择

CMA 在窗口菜单的底部显示出当前打开的文档窗口的清单。有一个打勾记号出现在活动的窗口的文档名前。从该清单中挑选一个文档可使其窗口成为活动窗口。



13. 帮助菜单项



帮助菜单提供以下的命令，为您提供使用这个应用程序的帮助

① 关于 (A) CMA...

显示您的 CMA 版本的版权通告和版本号码。

② 实验帮助 (E) ...

显示实验帮助的开场屏幕。从此开场屏幕，您可跳到关于 CMA 所提供实验的参考资料。

③ 软件帮助 (S) ...

显示软件帮助的开场屏幕。从此开场屏幕，您可跳到关于使用 CMA 设备的参考资料。

(三) 工具栏命令按钮介绍:



显示或隐藏指令区。



显示或隐藏输出区。



保存下位机数据。




















向下位机装载数据。



刷新指令区数据。



打开实验帮助。

-  打开微程序流程图。
-  单节拍运行。
-  单周期运行。
-  单机器指令运行。
-  连续运行。
-  停止运行。
-  打开实验数据文件。
-  保存实验数据。
-  跳转到首页。
-  向前翻页。
-  向后翻页。
-  跳转到末页。
-  连续向后翻页。
-  停止向后翻页。
-  打开示波器窗口。
-  启动示波器。
-  停止示波器。

第 5 章 TD-CMA 系统检测功能说明

为了更好地维护设备，TD-CMA 提供了两种系统检测功能，一种针对实验接线的检测，一种是针对维护的系统检测。对接线相对比较复杂的两个实验——简单模型机和复杂模型机实验——提供了实验接线检测，其余实验要么接线较少，要么是在复杂模型机实验接线基础上增加少量线，所以没有安排连线检测，连线检测结果会在软件‘输出区’的‘结果页’显示。

对于系统检测，提供了两种检测功能，包括基本检测和整机检测。基本检测需要按照图 5-1 接线后进行，可以实现对实验平台的 ALU® 单元、PC&AR 单元、MC 单元、MEM 单元的初级检测。基本检测通过后，可基本上保证上述单元电路的正确性，可主要用于设备维护性检测；整机检测需如图 5-2 所示连接线路后进行，一般在系统检测发现问题后使用，以进一步确定故障所在。

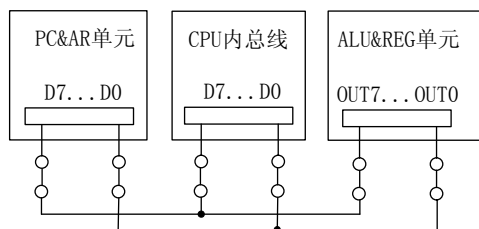


图 5-1 系统检测接线图

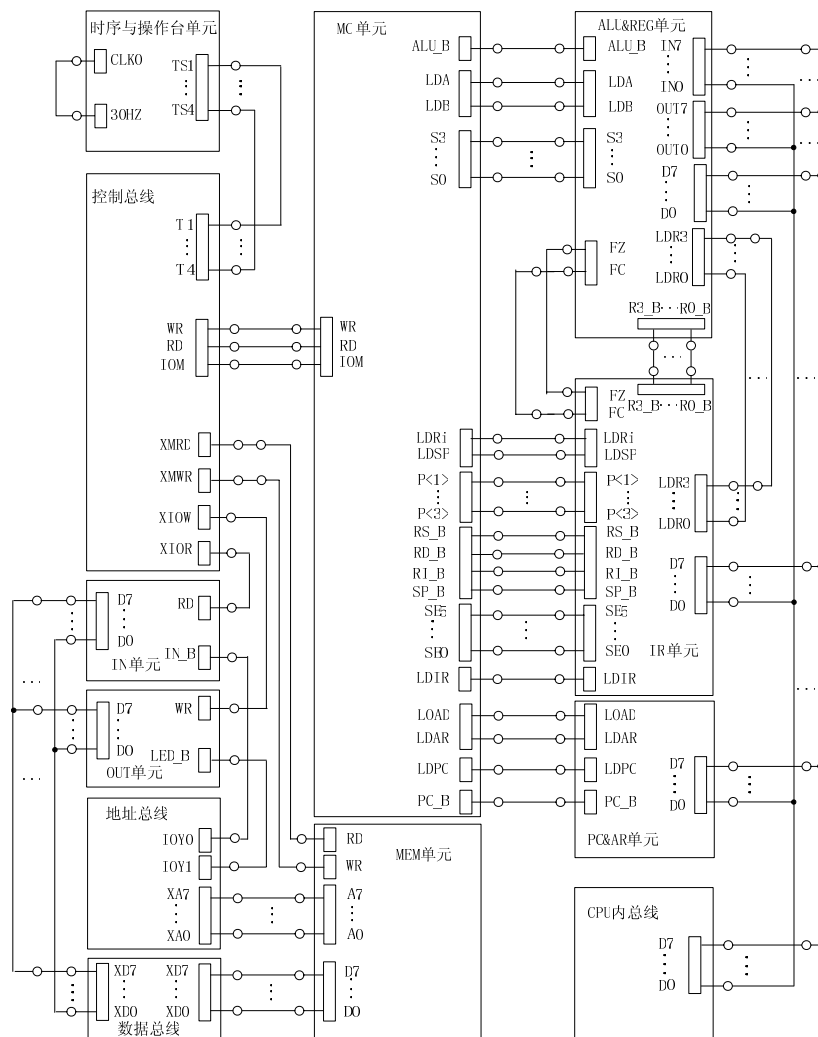


图 5-2 整机检测接线图

第 6 章 TD-CMA 系统常见故障的分析及处理

1. 与 PC 联机失败

- (1) 首先按复位键，检查单片机复位是否正确。
- (2) 检查串口是否接触良好，（包括 PC 机的串口是否完好，通讯电缆是否完全接触）。
- (3) 在前述(1)、(2)均正常的情况下，请检查电源的+5V 输出电压是否正常，若输出正常，请检查实验接线或查看线路板上有无散落的导线或元器件，导致短路。
- (4) 若以上均正常，请更换 SYS 单元的 MAX232 芯片。
- (5) 若以上均正常，请更换 SYS 单元的 89S51 芯片。
- (6) 若以上均正常，请更换晶振。

2. 微程序无法写入

- (1) 若手动写入程序时，检查时序与操作台单元的 KK3 开关应处于‘编程’状态。
- (2) 若以上均正常，请更换 2816 即可。
- (3) 若联机写入程序时，检查检查单片机复位是否正确，单片机和 MC 单元可能坏。

3. 实验无法通过

这种情况一般出现在一次或几次实验之后，其原因是：

- (1) 实验时，芯片损坏，只需更换相应单元的芯片即可。
- (2) 实验接线错误，应仔细检查接线。

如果您在实验中遇到无法解决的问题，请与我公司售后服务部联系。

附录 1 微程序流程图编程方法

程序以行为单位，每一行代表一条指令，并允许有注释，凡“;”后的所有内容都认为是被注释掉的内容。程序中可以有任意多的空格或是空行，即使是同一个数字之间有空格也认为是合法的，如“45”或是“4 5”，认为是等价的，因为在生成图形文件的过程中，这些注释和空格或是空行都将被忽略掉。但是有一点，凡是双引号中的空格将不会被除掉，因为这些可能是您有意添加的。

1. 指令种类说明

(1) “T 指令”。

格式为：T：“列号：字符串”。如：T：“9:复杂模型机微程序流程图”。

说明：“T:”为指令标志，指明该行为标题行，字母 T 必须为大写，双引号中的内容为标题，其中内容又分为两部分，冒号前的数字为标题显示的列号，可以由‘0’打头，如写成‘09’也是合法的。冒号后面的内容即为用来显示的字符串，因而指令行：“T:“9:复杂模型机微程序流程图””的作用就是以第 9 列的中心，在窗口客户区顶部显示标题：复杂模型机微程序流程图。

如果不指明列号，则以窗口总宽度的一半为中心显示标题，如果标题指令中出现多个冒号则只认第一个，如果字符串以冒号打头则认为列号为 0。该指令行必须位于逻辑上的第 1 行，所谓逻辑上的行是指去除注释行和空行以后的行号。

(2) “R 指令”。

格式：R：行数，如：R：020。

说明：“R:”为指令标志，指明该指令行为总行数，字母 R 也必须为大写，其后的数字为总行数，可以以‘0’打头。该指令行必须位于逻辑上的第 2 行，且行数最大不能超过 500。

(3) “C 指令”。

格式：C：列数，如：C：017。

说明：“C:”为指令标志，指明该指令行为总列数，字母 C 也必须为大写，其后的数字为总列数，可以以‘0’打头。该指令行必须位于逻辑上的第 3 行，且列数最大不能超过 500。

(4) “\$指令”。

格式：\$：行号，列号，微地址，图片类型，测试字，指令说明，如：\$：04, 09, 02, 1, 80, "RAM->BUS:BUS->IR"。

2. 指令使用说明

其中微地址、测试字和指令说明字段可以为空。

(1) “\$:”为指令标志，指明该指令为一般性指令。

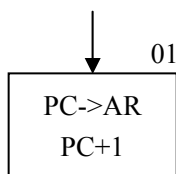
(2) 行号、列号指定了该图形节点所在的位置，不论是哪种图形，其所占有的空间大小是固定的，这样便于一般性处理。

(3) 微地址指明了该节点所对应的微地址，如果该节点图片类型为 1，也就是普通意义上的一条微指令（下位机每执行一次所对应的一个节点），其值是 16 进制的，因而取值范围是 00-3F，共 64 个单位，而如果该节点图片类型为其他类型的话，其取值最好不要在 00-3F 范围之内，以

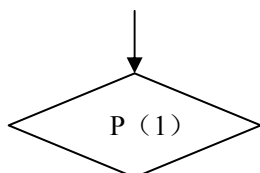
免发生冲突。每个结点的微地址应该是唯一的，如果有重复的情况将会导致错误。如果该字段为空则将其视为-1。

(4) 图片类型指明该节点以什么方式绘图，而且不同的图形类型，其后的微程序说明格式也不一样，因而每种图形的微程序说明格式也在此表述，系统共提供了 9 种图形类型，在您的程序中可以包含其中的任意种。这 9 种图形分别为：

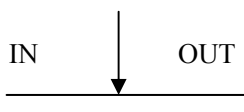
I、图形号为 1，整个图形高度所占用的象素为逻辑上的一行，而整个图形宽度所占用的象素为逻辑上的一列，以后的图形中的行列都以此为标准。图形中箭头所占用的高度为图形总高度的 1/3，矩形中的文字为微程序说明，居中显示，最多可显示 3 行，且图形高度能自动调整一级，但超过 3 行将不显示，微程序说明的书写方法是以“:”来换行，如果一行长度过长将会导致错误，右上角的数字为微地址，这就是用得最多的一种图形，其每个图形和下位机一条微指令相对应，而其它图形都只是做一些辅助性的工作。



II、这是测试字图形，图形号为 2，其行列号同图形 1，菱形中的文字为测试字说明，无论您输入多少行，其只会标出第一行，书写方式也同图形 1，其‘微地址’不加以显示，但要求用户在编程序时加以关联，否则运行时可能达不到预期的效果。

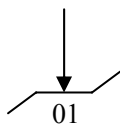


III、这是跟在测试字后的助记符说明，图形号为 3，其行号同图形 1，但其列号是有别的意义的，列号指的是箭头所在列的列号，因为一般的助记符显示位置为该列的正中间，而对于列号所在的列在靠左的位置加以显示，而且在绘画运行轨迹时，也是从列号所在的列绘到当前的列。助记符的书写方法是：在微指令说明字段加以描述，如：8:10:KWE(01):KRD(00):RP(11)，第一个冒号前的数字代表图形的起始列号，第二个冒号前的数字代表图形的结束列号，且应大于起始列号，这两部分必须加以描述清楚。后面用冒号隔开的每个字段代表一个助记符，助记符应按从左到右的次序依次书写，如果中间某项为空，也得用冒号隔开，终止列号和起始列号之差必须和助记符的个数相等。



IV、这是转公共的图形，图形号为 4，行列号和图形 1 相同，微地址一般为空，微指令说

明字段中的内容中为转公共的微地址，如下图中显示的 01 即为转公共的微地址。



V、这是每个微程序流图分支的标题，图形号为 5，其行号同图形 1，而列号是指该标题所显示的中心所在的列号，显示颜色同主标题，微地址和测试字一般为空，微指令说明字段中的内容即为要显示的标题。

运行微程序

VI、这是转移图形，图形号为 6，如果遇到有循环之类的图形，可以用转移图形来表示，该图形和图形 4 很相似，指令也类似，下图中的文字描述指的是要转移的地址，其内容来自微指令说明字段中。



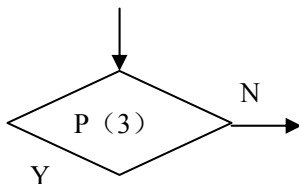
VII、该图形在平时用得不多，图形号为 7，只有在复杂的微程序流图中才可能用到，主要用于填充空白，使整幅图在视觉上变得美观，如复杂模型机微程序流图。其一般用在图形 1 之后，利用图形 1 或其它图形的测试字字段和其建立起关联。



VIII、图形号 8，该图形和上面的一样，只是在较为复杂的微程序流图中才会用到，具体可见复杂模型机微程序流图文件中的范例。



IX、带分支的测试字图形，图形号为 9，该图形和图形 2 很类似，编程方式也是一样的，具体见示例。



您可以在任意的纯文本编辑器中编写微程序流图文件，但一定要以*.MFP 的格式存储，您

也可以在系统中带有的编辑器进行编辑，当您编辑好并存盘后，就可以生成微程序流程图了。方法是选择菜单：[运行\微程序流程图]命令来解释您编写的微程序流程图文件，如果您的程序没有错误，系统就会自动为您生成一个微程序流程图，这一过程相当于预览功能，您可以边编写边预览，但是有一点，您必须打开复杂模型机或是重叠模型机数据通路图后才可以启用上述功能。

3. 系统检错类型

如果您的微程序流程图文件有错误，不用紧张，系统会为您找出错误，并提供错误的一些信息，如：程序的第 10 行是这样一行代码：`$: 16, 05, v82, 9, , "P(3)"`，那么系统在解释程序时会提供下面的错误：

Error At:第 10 行，错误的微地址号！

`$: 16, 05, v82, 9, , "P(3)"`

您可以根据系统提供的错误信息很容易改掉程序中的错误，将 82 前的字母 v 去掉就可以了。

系统提供的检错类型有：

(1) 双引号不匹配！

要求微指令说明必须用成对的双引号括起来，而且只能出现一对，不能出现多对双引号或是成单的双引号。

(2) 错误的标题！

第一行程序即为标题行，如果不以“T:”打头，或者双引号外还有别的字母或数字等都可能该错误。

(3) 错误的总行数或总行数溢出！

第二行程序即为总行数，如果不以“R:”打头，或者总行数值小于 1，或大于 500，或总行数字中含有非数字的字母等都可能该错误。

(4) 错误的总列数或总列数溢出！

第三行程序即为总列数，如果不以“C:”打头，或者总列数值小于 1，或大于 500，或总列数字中含有非数字的字母等都可能该错误。

(5) 错误的指令,参数不全！

从第四行开始，就是普通的指令了，如果指令不是以“\$:”打头，或是指令中的“,”不为 5 个就会导致该错误。

(6) 错误的行号！

指令中行号含有非数字字母等，会导致该错误。

(7) 行号溢出或为空！

行号值小于 0，或行号数大于 500 会导致该错误。

(8) 错误的列号！

指令中列号含有非数字字母等，会导致该错误。

(9) 列号溢出或为空！

列号值小于 0，或列号数大于 500 会导致该错误。

(10) 错误的微地址号！

指令中微地址含有非数字字母等，会导致该错误。

(11) 微地址号溢出！

微地址值小于 0，或微地址数大于 2 位数会导致该错误。

(12) 错误的图形号!

指令中图形号含有非数字字母等，会导致该错误。

(13) 图形号溢出或为空!

图形号值小于 0，或图形号大于 9 会导致该错误。

(14) 错误的测试字号!

指令中测试字号含有非数字字母等，会导致该错误。

(15) 测试字号溢出!

测试字号值小于 0，或测试字号大于 2 位数会导致该错误。

(16) 微指令说明错误!

微指令说明没有被引号括住，或者引号外有其它内容会导致该错误。

(17) 微指令说明中出现非法双引号!

微指令说明中出现多余的双引号会导致该错误。

(18) 微指令说明中字符数超标(大于 16)!

微指令说明中每行的字符是有限制的，不能大于 16 个字节（一个汉字为两个字节），如果过了就会导致该错误。

(19) 指令中行号大于总行数

一旦总行数指定后，指令中所有的行号是不能超过总行数的，如果超过就会导致该错误。

(20) 指令中列号大于总列数

一旦总列数指定后，指令中所有的列号是不能超过总列数的，如果超过就会导致该错误。

(21) 助记符说明错误!

如果助记符说明时指定的总列数（最大列和最小列之差）和实际的助记符个数不相等就会导致该错误。

(22) 找不到测试字中对应的结点

因为测试字字段的用途就是将当前执行的节点和其紧密关联的下个节点关联起来，所以如果指定的测试字号在所有结点中找不到与之对应的结点就会导致该错误。

(23) 第 X 行和第 Y 行和微地址相冲突

我们必须保证每个节点微地址的唯一性，这样才会使得将上位机中的节点和下位机中相应的微地址单元对应起来，如果有两个结点的微地址一样就会导致该错误。

4. 例程说明

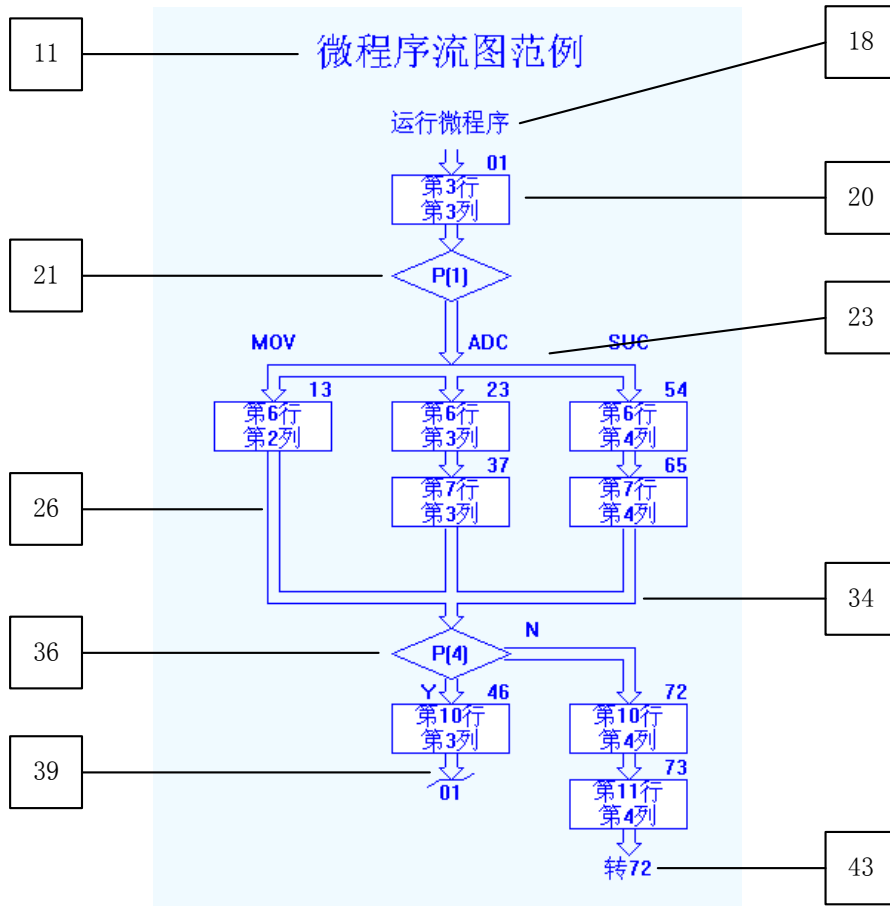
现在假设要生成如附图 1-1 所示的微程序流程图，看一看其生成过程。

您可以在安装目录下的 Sample 子目录下找到一个叫微程序流程图范例.MFP 的文件，就是该范例，微程序流程图文件的内容如下（行号为后来人为加上的）：

```

01  ;//*****
02  ;//
03  ;//      微程序流图范例文件      ;//
04  ;//
05  ;// FileName: Sample.MFP By TangDu Co.,LTD ;//
06  ;//*****

```



附图 1-1 微程序流图示例

```

07
08  ;//*****
09  ;//      标题及总行、列数
10
11  T : "3:微程序流图范例"
12  R : 13
13  C : 05
14
15  ;//*****
16  ;//      运行微程序
17
18  $ : 02, 03, , 5, , "运行微程序"
19

```

```

20 $ : 03, 03, 01, 1, 80, "第3行:第3列"
21 $ : 04, 03, 80, 2, , "P(1):Poilk:oifu"
22
23 $ : 05, 03, 81, 3, , "2:4:MOV:ADC:SUC"
24
25 $ : 06, 02, 13, 1, 82, "第6行:第2列"
26 $ : 07, 02, 82, 7, 83, ""
27
28 $ : 06, 03, 23, 1, , "第6行:第3列"
29 $ : 07, 03, 37, 1, 83, "第7行:第3列"
30
31 $ : 06, 04, 54, 1, , "第6行:第4列"
32 $ : 07, 04, 65, 1, 83, "第7行:第4列"
33
34 $ : 08, 03, 83, 8, 91, "2:4"
35
36 $ : 09, 03, 91, 9, , "P(4)"
37
38 $ : 10, 03, 46, 1, , "第10行:第3列"
39 $ : 11, 03, , 4, , "01"
40
41 $ : 10, 04, 72, 1, , "第10行:第4列"
42 $ : 11, 04, 73, 1, , "第11行:第4列"
43 $ : 12, 04, , 6, , "72"
44
45 ;//*****
46 ;// End of All

```

对于每种图形类型在上图中都有体现，而每种图形和实际程序行号的对应关系也已在图中标注出来。第 01-06 行为注释行，第 07 行为空行，第 11 行为标题行，第 12、13 分别对总的行列数。

图形号 1 对应程序中的第 20、25 行等，对应图中的 20；

图形号 2 对应程序中和图中的 21；

图形号 3 对应程序中和图中的 23；

图形号 4 对应程序中和图中的 39；

图形号 5 对应程序中和图中的 18；

图形号 6 对应程序中和图中的 43；

图形号 7 对应程序中和图中的 26；

图形号 8 对应程序中和图中的 34；

图形号 9 对应程序中和图中的 36；

注意：在微程序流图文件的微指令说明字段中不能含有：逗号、引号、分号以及星号 ‘*’。每个微程序流图文件至少要有 4 行，前三行分别为标题、总行数和总列数，再就是一行 \$ 指令。