
计算机组成原理 实验指导书

2010年9月

目录

系统概述	2
实验一 寄存器实验	4
实验 1: A, W 寄存器实验	4
实验 2: R0, R1, R2, R3 寄存器实验	6
实验 3: MAR 地址寄存器, ST 堆栈寄存器, OUT 输出寄存器	10
实验二 运算器实验	13
实验三 存储器 EM 实验	15
实验 1: PC/MAR 输出地址选择	16
实验 2: 存储器 EM 写实验	16
实验 3: 存储器 EM 读实验	17
实验 4: 存储器打入 IR 指令寄存器/uPC 实验	18
实验 5: 使用实验仪小键盘输入 EM	19
实验四 PC 实验	21
实验 1: PC +1 实验	23
实验 2: PC 打入实验	23
实验五 微程序存储器 uM 实验	24
实验 1: 微程序存储器 uM 读出	25
实验 2: 使用实验仪小键盘输入 uM	25
实验六 微程序计数器 uPC 实验	26
实验 1: uPC 加一实验	27
实验 2: uPC 打入实验	27
实验七 模型机综合实验 (微程序控制器)	29
实验 1: 数据传送实验/输入输出实验	29
实验 2: 数据运算实验 (加/减/与/或)	31
实验 3: 指令流水实验	33
附录 1: CPTH 集成开发环境使用	36
附录 2: 实验仪键盘使用	41

系统概述

DJ-CPTH 简介

DJ-CPTH 型计算机组成原理实验系统采用单片机管理和 EDA 控制技术, 自带键盘和液晶显示器, 支持脱机和联 PC 机两种工作模式, 运用系统监控和数码管等实时监视, 全面动态管理模型机的运行和内部资源。模型机软硬件配置完整, 支持 8 位字长的多种寻址方式, 指令丰富, 系统支持 RS-232C 串行通讯, 并配有以 win98/2000/XP 为操作平台的动态跟踪集成调试软件, 特别适用于计算机组成原理课程的教学与实验。

DJ-CPTH 特点

1、采用总线结构

总线结构的计算机具有结构清晰, 扩展方便等优点。DJ-CPTH 实验系统使用三组总线即地址总线 ABUS、数据总线 DBUS、指令总线 IBUS 和控制信号, CPU、主存、外设和管理单片机等部件之间通过外部数据总线传输, CPU 内部则通过内部数据总线传输信息。各部件之间, 通过三态缓冲器作接口连接, 这样一方面增强总线驱动能力, 另一方面在模型机停机时, 三态门输出浮空, 能保证不管模型机的 CPU 工作是否正常, 管理单片机总能读/写主存或控存。

2、计算机功能模块化设计

DJ-CPTH 为实验者提供运算器模块 ALU, 众多寄存器模块 (A, W, IA, ST, MAR, R0...R3 等), 程序计数器模块 PC, 指令部件模块 IR, 主存模块 EM, 微程序控制模块 (控存) uM, 微地址计数器模块 UPC, 组合逻辑控制模块及 I/O 等控制模块。各模块间的电源线、地线、地址总线和数据总线等已分别连通, 模块内各芯片间数据通路也已连好, 各模块的控制信号及必要的输出信号已被引出到主板插孔, 供实验者按自己的设计进行连接。

3、智能化控制

系统在单片机监控下, 管理模型机运行和读写, 当模型机停机时, 实验者可通过系统键盘, 读写主存或控存指定单元的内容, 使模型机实现在线开发。模型机运行时, 系统提供单步一条微指令 (微单步)、单步一条机器指令 (程单步), 连续运行程序及无限止暂停等调试手段, 能动态跟踪数据, 流向、捕捉各种控制信息, 实时反映模型机现场, 使实验者及时了解程序和微程序设计的正确性, 便于修改。

4、提供两种实验模式

①手动运行 “Hand……”: 通过拨动开关和发光二极管二进制电平显示, 支持最底层的手动操作方式的输入/输出和机器调试。

②自动运行: 通过系统键盘及液晶显示器或 PC 机, 直接输入或编译装载用户程序 (机器码程序和微程序), 实现微程序控制运行, 运用多种调试手段运行用户程序, 使实验者对计算机组成原理一目了然。

5、开放性设计

运算器采用了 EDA 技术设计, 随机出厂时, 已提供一套已装载的方案, 能进行加、减、与、或、带进位加、带进位减、取反、直通八种运算方式, 若用户不满意该套方案, 可自行

重新设计并通过 JTAG 口下载。逻辑控制器由 CPLD 实现，也可进行重新设计并通过 JTAG 口下载。用户还可以设计自己的指令/微指令系统。系统中已带三套指令/微程序系统，用户可参照来设计新的指令/微程序系统。

系统的数据线、地址线、控制线均在总线接口区引出，并设计了 40 芯锁进插座，供用户进行 RAM、8251、8255、8253、8259 等接口器件的扩展实验。

6、支持中断实验

采用最底层的器件设计，让学生可以从微程序层面上学习中断请求、中断响应、中断处理、中断入口地址的产生、中断服务程序及中断返回（RETI）整个过程。

7、支持两种控制器实验

系统提供两种控制器方式，即微程序控制器和组合逻辑控制器。在微程序控制器中，系统能提供在线编程，实时修改程序，显示程序并进行调试的操作环境。组合逻辑控制器，已下载有一套完整的实验方案，用户也可使用 CPLD 工具在 PC 机上进行自动化设计。

8、支持子程序调用、返回、指令流水线和 RISC 精简指令系统实验。

9、配备以 Win98/2000/XP 为操作平台的集成调试软件包

系统支持 RS-232C 串行通讯，借助 PC 资源形成了强大的在线文档与图形的动态管理系统，自带编译器，支持汇编语言的编辑、编译、调试，一次点击即可完成程序和与其对应微程序的链接装载并自动弹出调试窗口，在主界面中开辟了程序和与其对应微程序的调试、模型机结构示意图（点击各模块即可修改双向模块参数）、微程序等跟踪显示窗口，供用户选择，可动态显示数据流向、实时捕捉数据、地址、控制总线的各种信息，使调试过程极为生动形象。

实验系统组成

CPTH 计算机组成原理实验系统由实验平台、开关电源、软件三大部分组成。

实验平台上有寄存器组 R0-R3、运算单元、累加器 A、暂存器 W、直通/左移/右移单元、地址寄存器、程序计数器、堆栈、中断源、输入/输出单元、存储器单元、微地址寄存器、指令寄存器、微程序控制器、组合逻辑控制器、扩展单元、总线接口区、微动开关/指示灯、逻辑笔、脉冲源、管理单片机、24 个按键、字符式 LCD、RS232。学生在进行实验之前应掌握实验平台上的各组成部分的位置和基本功能，以方便后续使用。

实验一 寄存器实验

实验要求：利用 CPTH 实验仪上的 K16 …… K23 开关做为 DBUS 的数据，其它开关做为控制信号，将数据写入寄存器，这些寄存器包括累加器 A，工作寄存器 W，数据寄存器组 R0..R3，地址寄存器 MAR，堆栈寄存器 ST，输出寄存器 OUT。

实验目的：了解模型机中各种寄存器结构、工作原理及其控制方法。

实验电路：寄存器的作用是用于保存数据的，因为我们的模型机是 8 位的，因此在本模型机中大部分寄存器是 8 位的，标志位寄存器（Cy、Z）是两位的。

CPTH 用 74HC574 来构成寄存器。74HC574 的功能如下：

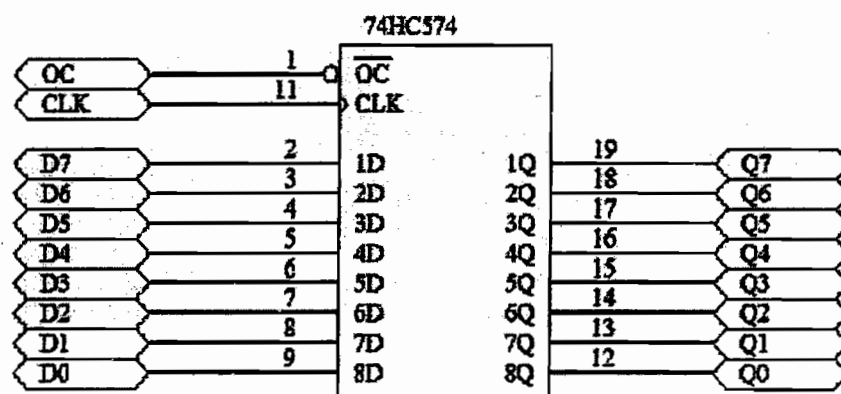


图 1.1 74HC574 寄存器原理图

1. 在 CLK 的上升沿将输入端的数据打入到 8 个触发器中；
2. 当 OC = 1 时触发器的输出被关闭，当 OC=0 时触发器的输出数据如下表所示：

OC	CLK	Q7..Q0	注释
1	X	ZZZZZZZZ	OC为1时触发器的输出被关闭
0	0	Q7..Q0	当OC=0时触发器的数据输出
0	1	Q7..Q0	当时钟为高时，触发器保持数据不变
X	↑	D7..D0	在CLK的上升沿将输入端的数据打入到触发器中

74HC574 工作波形图如下：

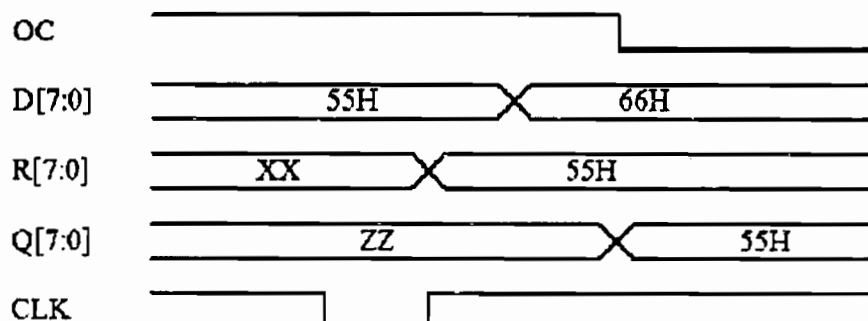


图 1.2 74HC574 工作波形图

实验 1: A, W 寄存器实验

寄存器 A、W 原理图如下：

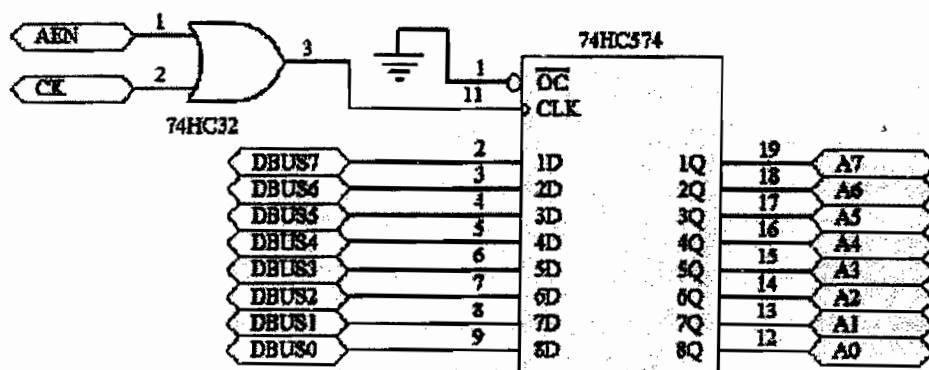


图 1.3 寄存器 A 原理图

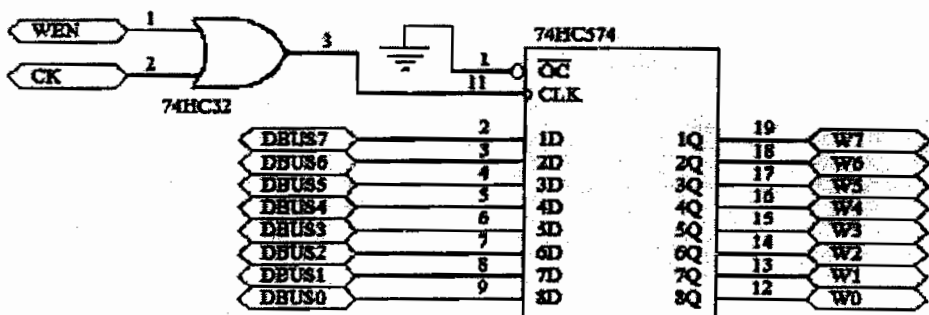


图 1.4 寄存器 W 原理图

寄存器 A、W 写工作波形图如下：

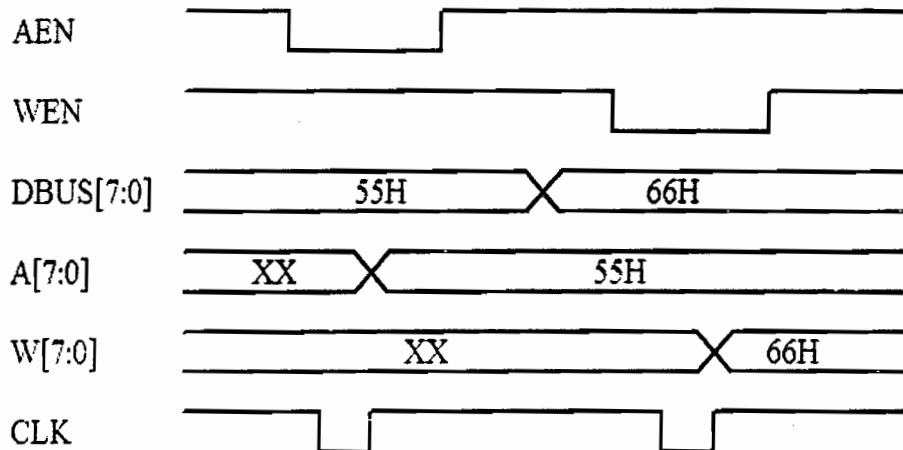


图 1.5 寄存器 A、W 写工作波形图

实验 1 连接线表如下：

连接	信号孔	接入孔	作用	状态说明
1	J1座	J3座	将K23-K16接入DBUS[7:0]	实验模式：手动
2	AEN	K3	选通A	低电平有效
3	WEN	K4	选通W	低电平有效
4	CK	已连	ALU工作脉冲	上升沿打入

系统清零和手动状态设定：K23-K16 开关置零，按[RST]钮，按[TV/ME]键三次，进入"Hand....."手动状态。

在后面实验中实验模式为手动的操作方法不再详述。

● 将 55H 写入 A 寄存器

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据 55H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	1	0	1	0	1	0	1

置控制信号为：

K4(WEN)	K3(AEN)
1	0

按住 STEP 脉冲键，CK 由高变低，这时寄存器 A 的黄色选择指示灯亮，表明选择 A 寄存器。放开 STEP 键，CK 由低变高，产生一个上升沿，数据 55H 被写入 A 寄存器。

● 将 66H 写入 W 寄存器

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据 66H，如下

K23	K22	K21	K20	K19	K18	K17	K16
0	1	1	0	0	1	1	0

置控制信号为：

K4(WEN)	K3(AEN)
0	1

按住 STEP 脉冲键，CK 由高变低，这时寄存器 W 的黄色选择指示灯亮，表明选择 W 寄存器。放开 STEP 键，CK 由低变高，产生一个上升沿，数据 66H 被写入 W 寄存器。

注意观察：

1. 数据是在放开 STEP 键后改变的，也就是 CK 的上升沿数据被打入。
2. WEN, AEN 为高时，即使 CK 有上升沿，寄存器的数据也不会改变。

实验 2：R0, R1, R2, R3 寄存器实验

寄存器 R 原理图如下：

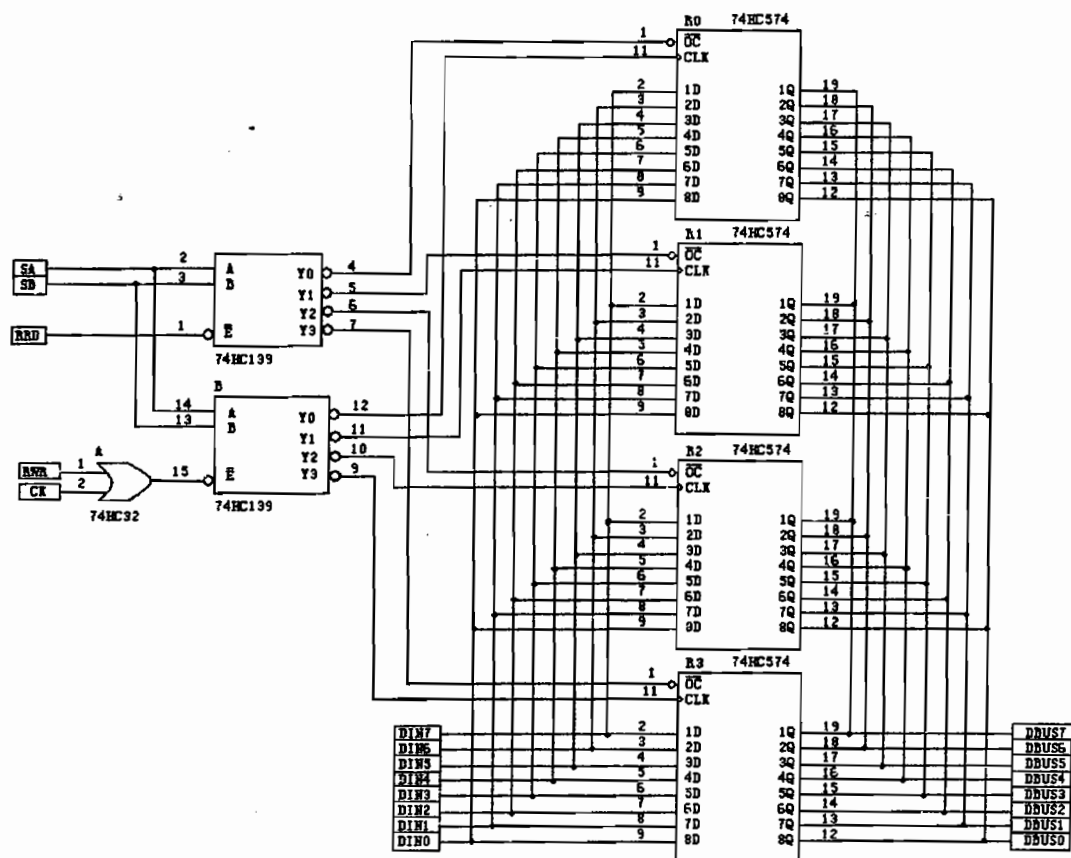


图 1.6 寄存器 R 原理图

寄存器 R 写工作波形图如下:

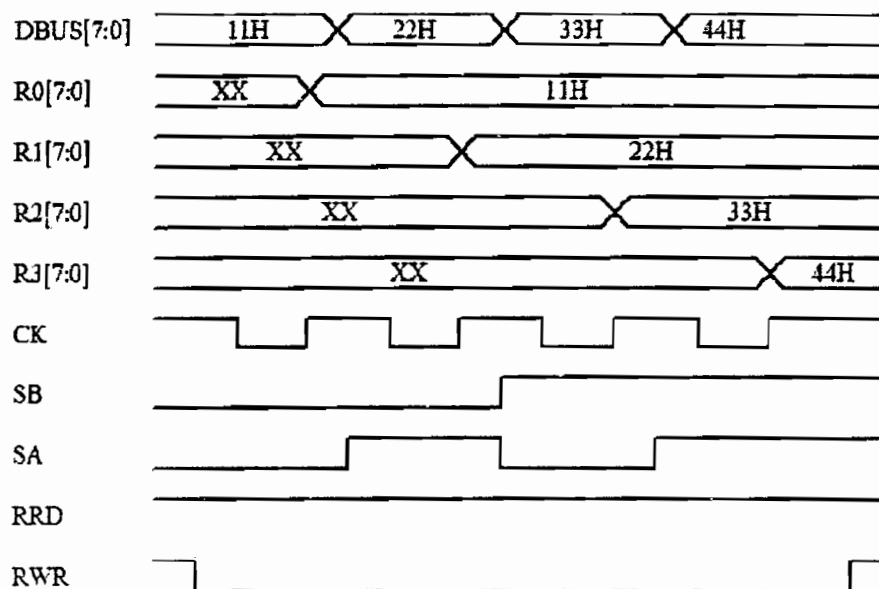


图 1.7 寄存器 R 写工作波形

实验 2 连接线表如下:

连接	信号孔	接入孔	作用	状态说明
1	J1座	J3座	将K23-K16接入DBUS[7:0]	实验模式：手动
2	RRD	K11	寄存器组读使能	低电平有效
3	RWR	K10	寄存器组写使能	低电平有效
4	SB	K1	寄存器选择B	
5	SA	K0	寄存器选择A	
6	CK	已连	寄存器工作脉冲	上升沿打入
7	D7..D0	L7..L0	观察寄存器数据输出	

● 将 11H 写入 R0 寄存器

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据 11H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	1	0	0	0	1

置控制信号为：

K11(RRD)	K10(RWR)	K1(SB)	K0(SA)
1	0	0	0

按住 STEP 脉冲键，CK 由高变低，这时寄存器 R0 的黄色选择指示灯亮，表明选择 R0 寄存器。放开 STEP 键，CK 由低变高，产生一个上升沿，数据 11H 被写入 R0 寄存器。

● 将 33H 写入 R2 寄存器

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据 33H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	1	1	0	0	1	1

置控制信号为：

K11(RRD)	K10(RWR)	K1(SB)	K0(SA)
1	0	1	0

按住 STEP 脉冲键，CK 由高变低，这时寄存器 R2 的黄色选择指示灯亮，表明选择 R2 寄存器。放开 STEP 键，CK 由低变高，产生一个上升沿，数据 33H 被写入 R2 寄存器。

注意观察：

1. 数据是在放开 STEP 键后改变的，也就是 CK 的上升沿，数据被打入。
2. K1(SB)，K0(SA) 用于选择寄存器，组合结果如下表：

K1 (SB)	K0 (SA)	选择
0	0	R0
0	1	R1
1	0	R2
1	1	R3

寄存器 R 读工作波形图如下：

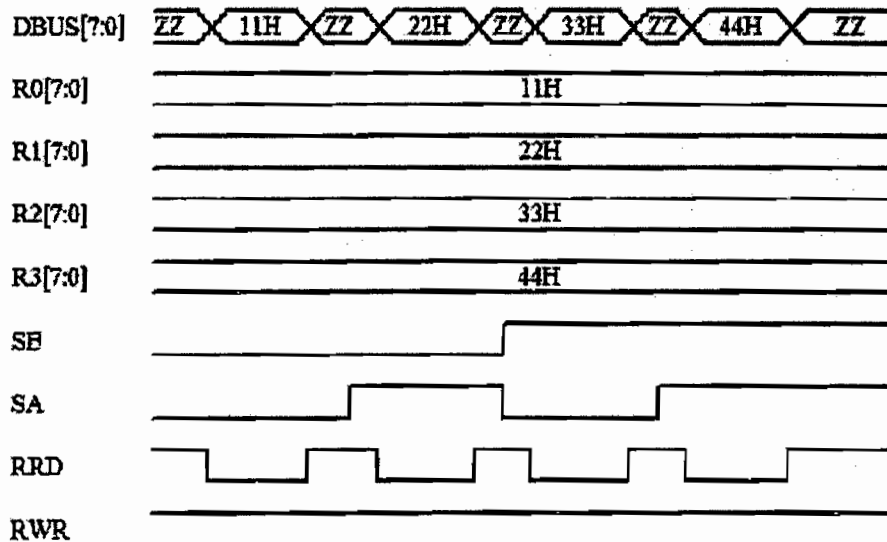


图 1.8 寄存器 R 读工作波形图

● 读 R1 寄存器

置控制信号为：

K11(RRD)	K10(RWR)	K1(SB)	K0(SA)
0	1	0	1

这时寄存器 R1 的红色输出指示灯亮，R1 寄存器的数据送上数据总线。此时数据总线指示灯 L7... L0 为：00100910，将 K11(RRD)置为 1，关闭 R1 寄存器输出。

● 读 R3 寄存器

置控制信号为：

K11(RRD)	K10(RWR)	K1(SB)	K0(SA)
0	1	1	1

这时寄存器 R3 的红色输出指示灯亮，R3 寄存器的数据送上数据总线。此时数据总线指示灯 L7... L0 为：01000100，将 K11(RRD)置为 1，关闭 R3 寄存器输出。

注意观察：

1.数据在 K11(RRD)为 0 时输出，不是沿触发，与数据打入不同。

实验 3: MAR 地址寄存器, ST 堆栈寄存器, OUT 输出寄存器

MAR 地址寄存器, ST 堆栈寄存器及 OUT 输出寄存器的原理图如下所示:

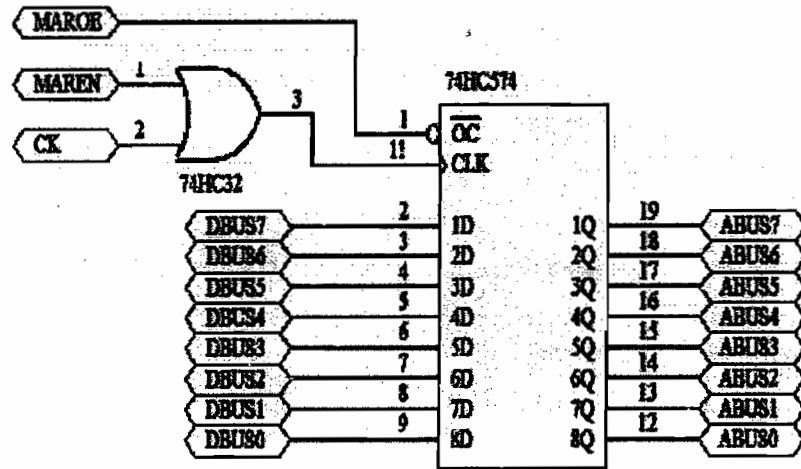


图 1.9 地址寄存器原理图

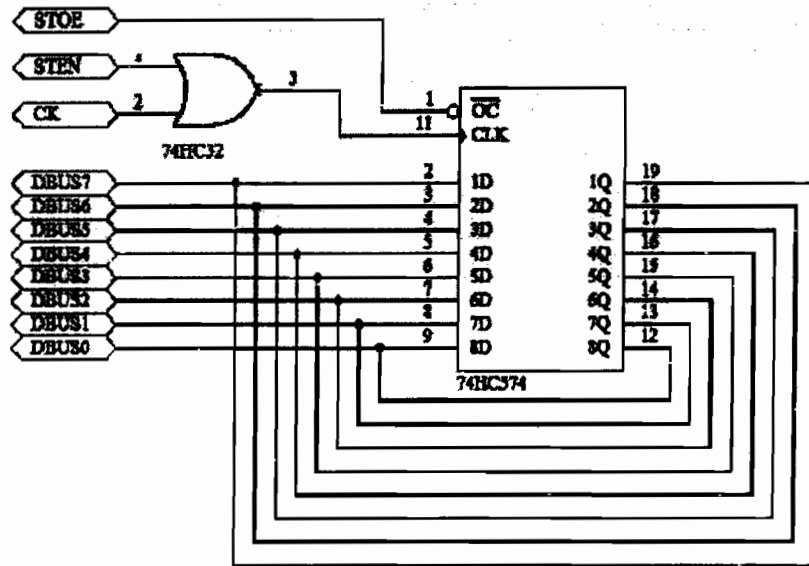


图 1.10 堆栈寄存器原理图

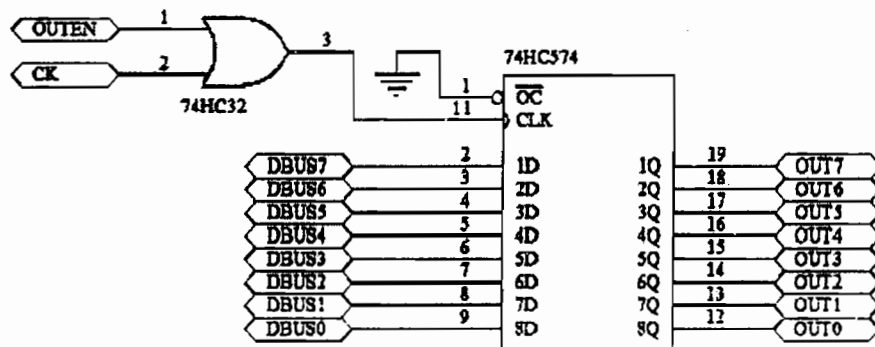


图 1.11 输出寄存器原理图

MAR 地址寄存器, ST 堆栈寄存器及 OUT 输出寄存器的工作波形如下图所示:

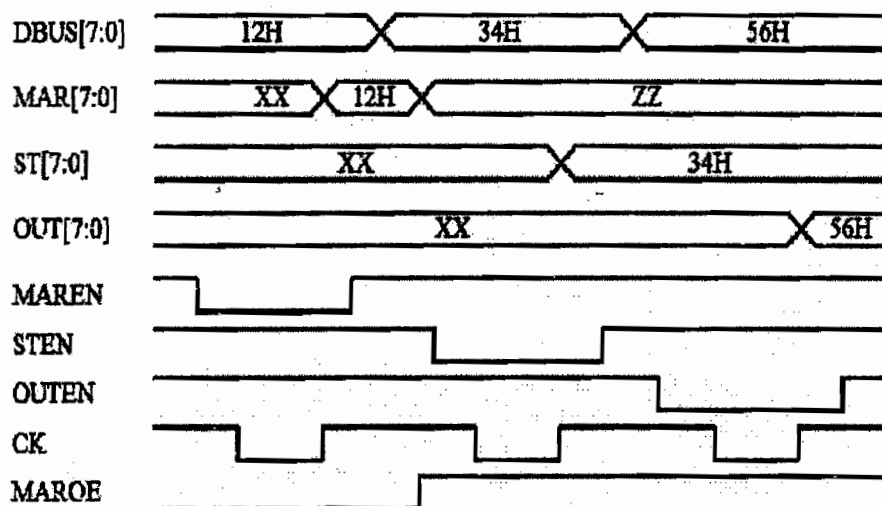


图 1.12 MAR, ST, OUT 寄存器写工作波形图

实验 3 连接线表如下:

连接	信号孔	接入孔	作用	状态说明
1	J2座	J3座	将K23-16接入DBU[7:0]	实验模式: 手动
2	MAROE	K14	MAR地址输出使能	低电平有效
3	MAREN	K15	MAR寄存器写使能	低电平有效
4	STEN	K12	ST寄存器写使能	低电平有效
5	OUTEN	K13	OUT寄存器写使能	低电平有效
6	CK	已连	寄存器工作脉冲	上升沿打入

● 将 12H 写入 MAR 寄存器

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入, 置数据 12H, 如下:

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	1	0	0	1	0

置控制信号为:

K15(MAREN)	K14(MAROE)	K13(OUTEN)	K12(STEN)
0	0	1	1

按住 STEP 脉冲键, CK 由高变低, 这时寄存器 MAR 的黄色选择指示灯亮, 表明选择 MAR 寄存器。放开 STEP 键, CK 由低变高, 产生一个上升沿, 数据 12H 被写入 MAR 寄存器。

K14(MAROE)为 0, MAR 寄存器中的地址输出, MAR 红色输出指示灯亮。

将 K14(MAROE)置为 1, 关闭 MAR 输出。

● 将 34H 写入 ST 寄存器

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据 34H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	1	1	0	1	0	0

置控制信号为：

K15(MAREN)	K14(MAROE)	K13(OUTEN)	K12(STEN)
1	1	1	0

按住 STEP 脉冲键，CK 由高变低，这时寄存器 ST 的黄色选择指示灯亮，表明选择 ST 寄存器。放开 STEP 键，CK 由低变高，产生一个上升沿，数据 34H 被写入 ST 寄存器。

● 将 56H 写入 OUT 寄存器

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据 56H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	1	0	1	0	1	1	0

置控制信号为：

K15(MAREN)	K14(MAROE)	K13(OUTEN)	K12(STEN)
1	1	0	1

按住 STEP 脉冲键，CK 由高变低，这时寄存器 OUT 的黄色选择指示灯亮，表明选择 OUT 寄存器。放开 STEP 键，CK 由低变高，产生一个上升沿，数据 56H 被写入 OUT 寄存器。

思考题：

1. 在这个实验中，为什么按下 STEP 脉冲键时，相应寄存器的数值立即改变，而不是放开 STEP 键时？

实验二 运算器实验

实验要求：利用 CPTH 实验仪的 K16……K23 开关作为 DBUS 数据，其它开关作为控制信号，将数据写累加器 A 和工作寄存器 W，并用开关控制 ALU 的运算方式，实现运算器的功能。

实验目的：了解模型机中算术、逻辑运算单元的控制方法。

实验电路：CPTH 中的运算器由一片 CPLD 实现，共实现 8 种算术与逻辑运算，通过 S2, S1, S0 来选择，运算数据由寄存器 A 及寄存器 W 给出，运算结果输出到直通门 D。运算器功能如下表：

S2	S1	S0	功能
0	0	0	A+W 加
0	0	1	A-W 减
0	1	0	A W 或
0	1	1	A & W 与
1	0	0	A+W+C 带进位加
1	0	1	A-W-C 带进位减
1	1	0	~A A 取反
1	1	1	A 输出 A

实验连接线如下表所示：

连接	信号孔	接入孔	作用	状态说明
1	J1 座	J3	将 K23-K16 接入 DBUS[7:0]	实验模式：手动
2	S0	K0	运算器功能选择	
3	S1	K1	运算器功能选择	
4	S2	K2	运算器功能选择	
5	AEN	K3	选通 A	低电平有效
6	WEN	K4	选 W	低电平有效
7	Cy IN	K5	运算器进位输入	
8	CK	已连	ALU 工作脉冲	上升沿打入

● 将 55H 写入 A 寄存器

二进制开关 K23……K16 用于 DBUS[7:0]的数据输入，置数据为 55H 状态如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	1	0	1	0	1	0	1

置控制信号为：

K5(Cy IN)	K4(WEN)	K3(AEN)	K2(S2)	K1(S1)	K0(S0)
0	1	0	0	0	0

按住 STEP 脉冲键，CK 由高变低，这时寄存器 A 的黄色选择指示灯亮，表明选择 A 寄

寄存器。放开 STEP 键，CK 由低变高，产生一个上升沿，数据 55H 被写入 A 寄存器。

● 将 33H 写入 W 寄存器

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据为 33H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	1	1	0	0	1	1

置控制信号为：

K5(Cy IN)	K4(WEN)	K3(AEN)	K2(S2)	K1(S1)	K0(S0)
0	0	1	0	0	0

按住 STEP 脉冲键，CK 由高变低，这时寄存器 W 的黄色选择指示灯亮，表明选择 W 寄存器。放开 STEP 键，CK 由低变高，产生一个上升沿，数据 33H 被写入 W 寄存器。

最后，置下表的控制信号，检验运算器的运算结果。

K5(Cy IN)	K2(S2)	K1(S1)	K0(S0)	结果(直通门 D)	注释
X	0	0	0	88H	加运算
X	0	0	1	22H	减运算
X	0	1	0	77H	或运算
X	0	1	1	11H	与运算
0	1	0	0	88H	带进位加运算
1	1	0	0	89H	带进位加运算
0	1	0	1	22H	带进位减运算
1	1	0	1	21H	带进位减运算
X	1	1	0	AAH	取反运算
X	1	1	1	55H	输出 A

注意观察：

运算器在加上控制信号及数据(A,W)后，立刻给出结果，不需时钟。

实验三 存储器 EM 实验

实验要求：利用 CPTH 实验仪上的 K16……K23 开关作为 DBUS 的数据，其它开关作为控制信号，实现程序存储器 EM 的读写操作。

实验目的：了解模型机中程序存储器 EM 的工作原理及控制方法。

实验电路：

存储器 EM 由一片 6116RAM 构成，是用户存放程序和数据的地方。存储器 EM 通过一片 74HC245 与数据总线相连。存储器 EM 的地址可选择由 PC 或 MAR 提供。

存储器 EM 的数据输出线直接接到指令总线 IBUS 上，指令总线 IBUS 的数据还可以来自一片 74HC245。当 ICOE 为 0 时，这片 74HC245 输出中断指令 B8。

存储器 EM 的结构原理图如下所示：

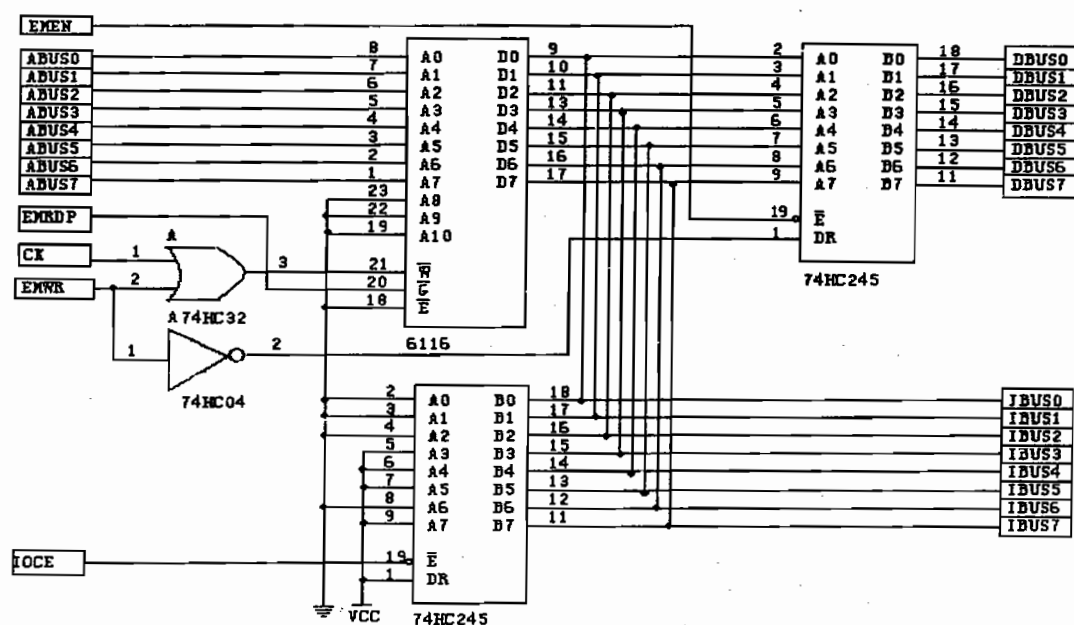


图 3.1 EM 的结构原理图

实验连接线如下表所示：

连接	信号孔	接入孔	作用	状态说明
1	J2座	J3座	将K23-K16接入DBUS[7:0]	实验模式：手动
2	IREN	K6	IR,uPC写允许	低电平有效
3	PCOE	K5	PC输出地址	低电平有效
4	MAROE	K4	MAR输出地址	低电平有效
5	MAREN	K3	MAR写允许	低电平有效
6	EMEN	K2	存储器与数据总线相连	低电平有效
7	EMRD	K1	存储器读允许	低电平有效
8	EMWR	K0	存储器写允许	低电平有效
9	CK	已连	PC工作脉冲	上升沿打入
10	CK	已连	MAR工作脉冲	上升沿打入
11	CK	已连	存储器写脉冲	上升沿打入
12	CK	已连	IR,uPC工作脉冲	上升沿打入

实验 1: PC/MAR 输出地址选择

置控制信号为:

K5 (PCOE)	K4 (MAROE)	地址总线	红色地址输出指示灯
0	1	PC输出地址	PC地址输出指示灯亮
1	0	MAR输出地址	MAR地址输出指示灯亮
1	1	地址总线浮空	
0	0	错误,PC及MAR同时输出	PC及MAR地址输出指示灯亮

注: 以下存储器 EM 实验均由 MAR 提供地址

实验 2: 存储器 EM 写实验

● 将地址 0 写入 MAR

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入, 置数据为 00H, 如下:

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	0	0	0	0	0

置控制信号为:

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	1	0	1	1	1

按 STEP 键, 将地址 0 写入 MAR。

● 将数据 11H 写入 EM[0]

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入, 置数据为 11H, 如下:

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	1	0	0	0	1

置控制信号为:

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	0	1	0	1	0

按 STEP 键, 将数据 11H 写入 EM[0]。

● 将地址 1 写入 MAR

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据为 01H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	0	0	0	0	1

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	1	0	1	1	1

按 STEP 键，将地址 1 写入 MAR。

- 将数据 22H 写入 EM[1]

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据为 22H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	1	0	0	0	1	0

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	0	1	0	1	0

按 STEP 键，将数据 22H 写入 EM[1]。

实验 3：存储器 EM 读实验

- 将地址 0 写入 MAR

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据为 00H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	0	0	0	0	0

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	1	0	1	1	1

按 STEP 键，将地址 0 写入 MAR

- 读 EM[0]

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	0	1	1	0	1

EM[0]被读出：11H。

- 将地址 1 写入 MAR

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据为 01H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	0	0	0	0	1

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	1	0	1	1	1

按 STEP 键，将地址 0 写入 MAR。

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	0	1	1	0	1

EM[1]被读出：22H。

实验 4：存储器打入 IR 指令寄存器/uPC 实验

- 将地址 0 写入 MAR

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据为 00H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	0	0	0	0	0

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	1	0	1	1	1

按 STEP 键，将地址 0 写入 MAR。

● 读 EM[0]，写入 IR 及 uPC

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
0	1	0	1	1	0	1

EM[0]被读出：11H

按 STEP 键，将 EM[0]写入 IR 及 uPC，IR=11H，uPC=10H

● 将地址 1 写入 MAR

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据为 01H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	0	0	0	0	1

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
1	1	1	0	1	1	1

按 STEP 键，将地址 1 写入 MAR。

● 读 EM[1]，写入 IR 及 uPC

置控制信号为：

K6 (IREN)	K5 (PCOE)	K4 (MAROE)	K3 (MAREN)	K2 (EMEN)	K1 (EMRD)	K0 (EMWR)
0	1	0	1	1	0	1

EM[1]被读出：22H。

按 STEP 键，将地址 EM[1]写入 IR 及 uPC，IR=22H，uPC=20H。

实验 5：使用实验仪小键盘输入 EM

1. 连接 J1，J2
2. 打开电源
3. 按 TV/ME 键，选择 EM
4. 输入两位地址，00
5. 按 NEXT，进入程序修改

-
6. 按两位程序数据
 7. 按 NEXT 选择下个地址/按 LAST 选择上个地址
 8. 重复 6, 7 步输入程序
 9. 按 RST 结束

实验四 PC 实验

实验要求：利用 CPTH 实验仪上的 K16..K23 开关作为 DBUS 的数据，其它开关作为控制信号，实现程序计数器 PC 的写入及加 1 功能。

实验目的：1、了解模型机中程序计数器 PC 的工作原理及其控制方法。2、了解程序执行过程中顺序和跳转指令的实现方法。

实验电路：

PC 是由两片 74HC161 构成的八位带预置计数器，预置数据来自数据总线。计数器的输出通过 74HC245 (PCOE) 送到地址总线。PC 值还可以通过 74HC245 (PCOE_D) 送回数据总线。

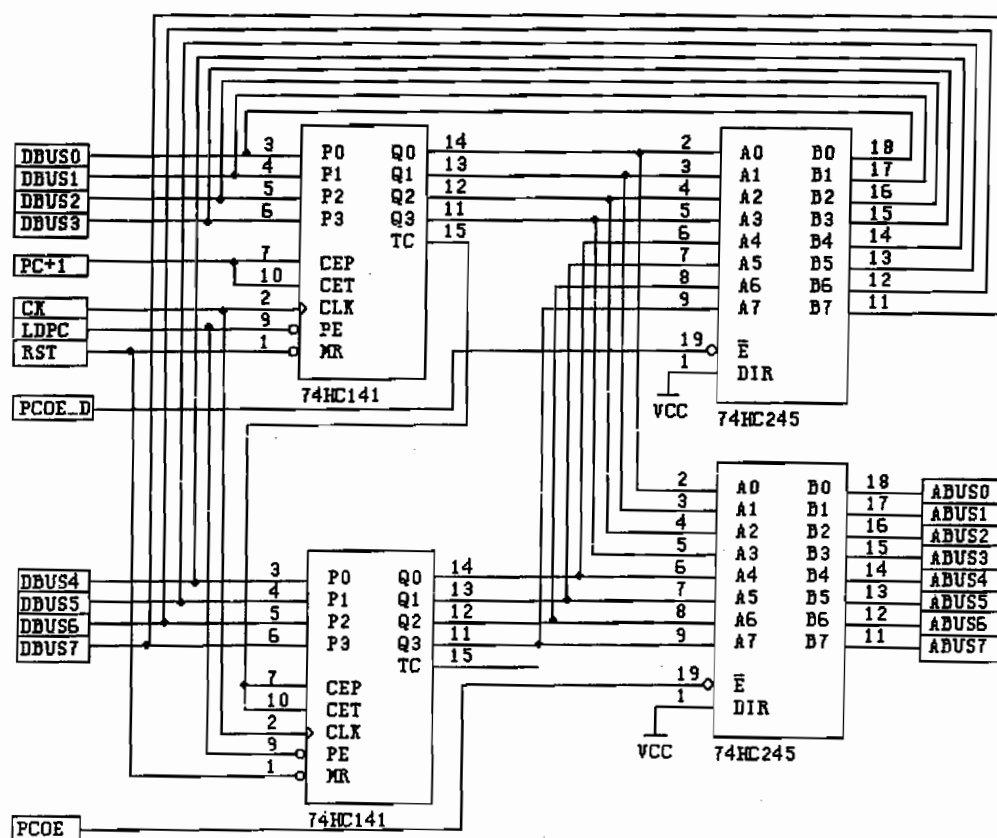


图 4.1 PC 原理图

在 CPTH 中，PC+1 由 PCOE 取反产生。

当 RST=0 时，PC 计数器被清 0

当 LDPC=0 时，在 CK 的上升沿，预置数据被打入 PC 计数器

当 PC+1=1 时，在 CK 的上升沿，PC 计数器加一

当 PCOE=0 时，PC 值送地址总线

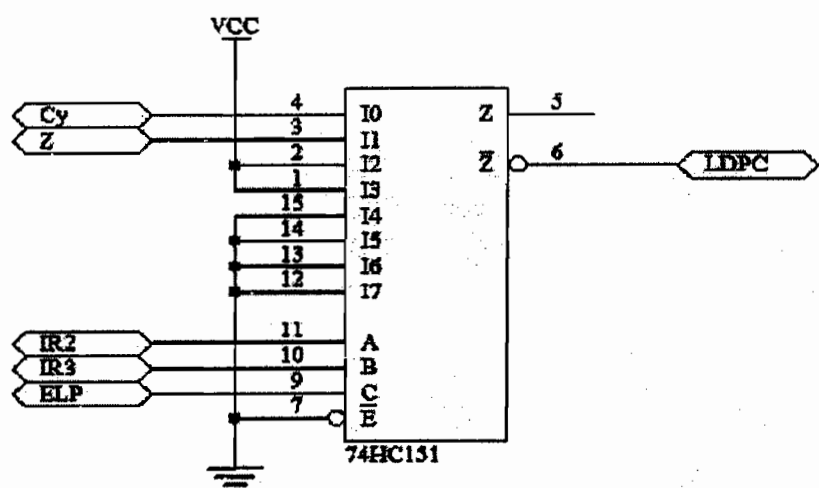


图 4.2 PC 打入控制原理图

PC 打入控制电路由一片 74HC151 八选一构成(isp1016 实现)。

ELP	IR3	IR2	Cy	Z	LDPC
1	X	X	X	X	1
0	0	0	1	X	0
0	0	0	0	X	1
0	0	1	X	1	0
0	0	1	X	0	1
0	1	X	X	X	0

当 ELP=1 时，LDPC=1，不允许 PC 被预置

当 ELP=0 时，LDPC 由 IR3，IR2，Cy，Z 确定

当 IR3 IR2 = 1 X 时，LDPC=0，PC 被预置

当 IR3 IR2 = 0 0 时，LDPC=非 Cy，当 Cy=1 时，PC 被预置

当 IR3 IR2 = 0 1 时，LDPC=非 Z，当 Z=1 时，PC 被预置

连接线表如下：

连接	信号孔	接入孔	作用	状态说明
1	J2座	J3座	将K23-K16接入DBU[7:0]	实验模式：手动
2	PCOE	K5	PC输出到地址总线	低电平有效
3	JIR3	K4	预置选择1	
4	JIR2	K3	预置选择0	
5	JRZ	K2	Z标志输入	
6	JRC	K1	C标志输入	
7	ELP	K0	预置允许	低电平有效
8	CK	已连	PC工作脉冲	上升沿打入

实验 1: PC +1 实验

置控制信号为:

K5 (PCOE)	K0 (ELP)
0	1

按一次 STEP 脉冲键, CK 产生一个上升沿, 数据 PC 被加一。

实验 2: PC 打入实验

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入, 置数据为 12H, 如下:

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	1	0	0	1	0

置控制信号为:

IR3 (K4)	IR2 (K3)	JRZ (K2)	JRC (K1)	ELP (K0)	LDPC	黄色PC预置指示灯
X	X	X	X	1	1	灭
0	0	X	1	0	0	亮
0	0	X	0	0	1	灭
0	1	1	X	0	0	亮
0	1	0	X	0	1	灭
1	X	X	X	0	0	亮

每置控制信号后, 按一下 STEP 键, 观察 PC 的变化。

实验五 微程序存储器 uM 实验

实验要求：利用 CPTH 实验仪上的开关作为控制信号，实现微程序存储器 uM 的输出功能。

实验目的：1、了解微程序控制方式模型机的基本工作原理。

2、了解微程序存储器 uM 的控制方法。

实验电路：

微程序存储器 uM 由三片 6116RAM 构成，共 24 位微指令，采用水平型微指令格式。存储器的地址由 uPC 提供，片选及读信号恒为低，写信号恒为高。微程序存储器 uM 始终输出 uPC，指定地址单元的数据。实验电路原理图如下所示：

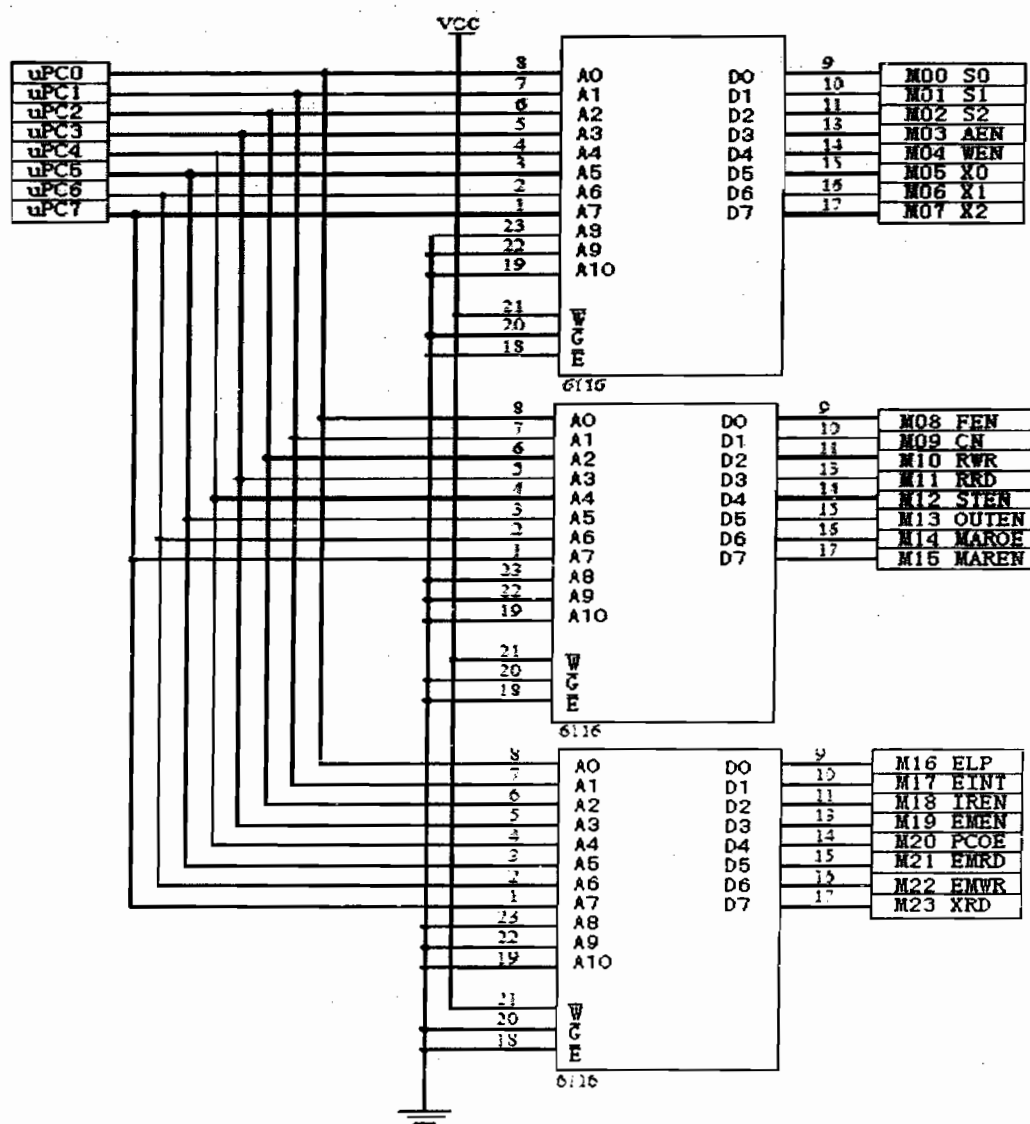


图 5.1 微程序存储器原理图

实验连接线如下表所示：

连接	信号孔	接入孔	作用	状态说明
1	J2座	J3座	将K23-K16接入DBUS[7:0]	实验模式：手动
2	IREN	K0	IR, uPC写使能	低电平有效
6	CK	已连	uPC工作脉冲	上升沿打入

实验 1：微程序存储器 uM 读出

置控制信号为：K0 为 1

uM 输出 uM[0]的数据

按一次 STEP 脉冲键，CK 产生一个上升沿，数据 uPC 被加一。

uM 输出 uM[1]的数据

按一次 STEP 脉冲键，CK 产生一个上升沿，数据 uPC 被加一。

uM 输出 uM[2]的数据

实验 2：使用实验仪小键盘输入 uM

1. 连接 J1, J2
2. 打开电源
3. 按 TV/ME 键，选择 uM
4. 输入两位地址, 00
5. 按 NEXT, 进入微程序修改
6. 按六位微程序数据
7. 按 NEXT 选择下个地址/按 LAST 选择上个地址
8. 重复 6,7 步输入微程序
9. 按 RST 结束

实验六 微程序计数器 uPC 实验

实验要求：利用 CPTH 实验仪上的 K16..K23 开关作为 DBUS 的数据，其它开关作为控制信号，实现微程序计数器 uPC 的写入和加 1 功能。

实验目的：1、了解模型机中微程序的基本概念。

2、了解 uPC 的结构、工作原理及其控制方法。

实验电路：

74HC161 是一片带预置的 4 位二进制计数器。功能如下：

当 RST = 0 时，计数器被清 0

当 IREN = 0 时，在 CK 的上升沿，预置数据被打入计数器

当 IREN = 1 时，在 CK 的上升沿，计数器加一

TC 为进位，当记数到 F (1111) 时，TC=1

CEP, CET 为计数使能，当 CEP, CET=1 时，计数器工作，CEP, CET=0 时，计数器保持原记数值。

微程序计数器 uPC 的结构原理图如下所示：

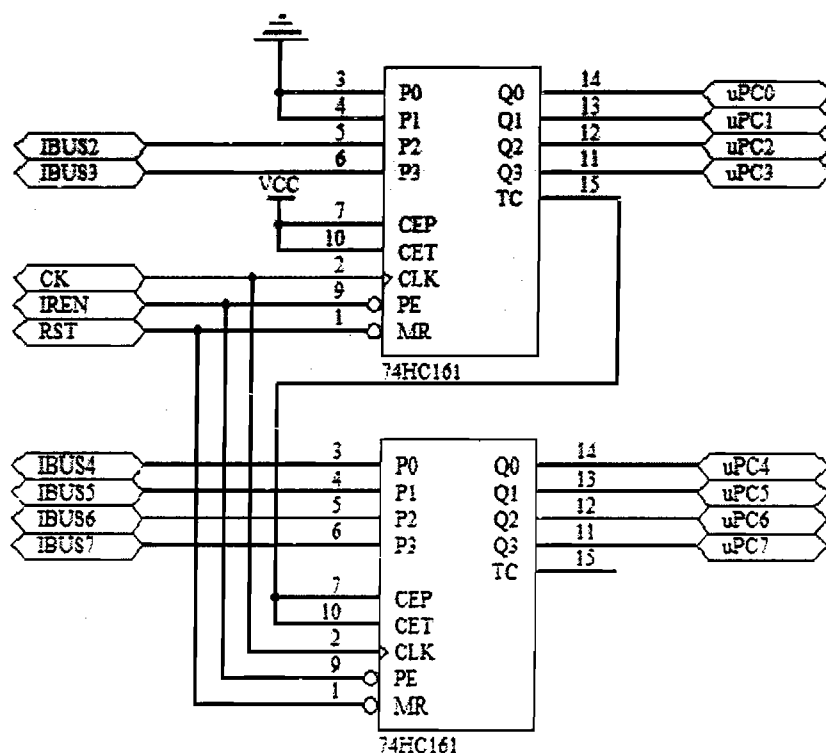


图 6.1 微程序计数器 uPC 原理图

微程序计数器 uPC 工作波形如下图所示：

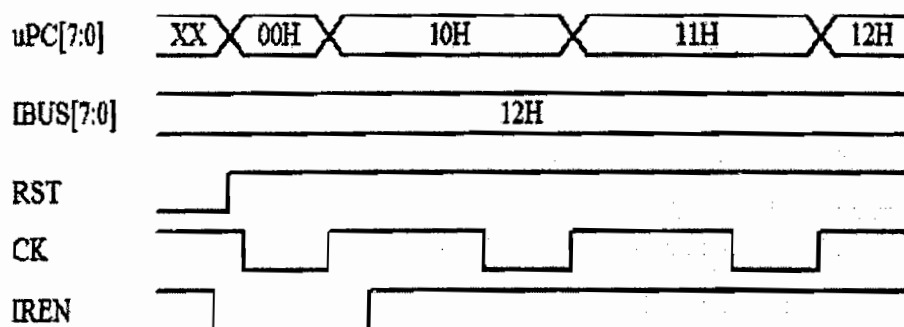


图 6.2 微程序计数器 uPC 工作波形图

在 CPTH 中，指令 IBUS[7:0]的高六位被接到 uPC 预置的高六位，uPC 预置的低两位被置为 0。一条指令最多可有四条微指令。

微程序初始地址为复位地址 00，微程序入口地址由指令码产生，微程序下一地址有计数器产生。

实验连接线如下表所示：

连接	信号孔	接入孔	作用	状态说明
1	J2座	J3座	将K23-K16接入DBU[7:0]	实验模式：手动
2	IREN	K0	预置uPC	低电平有效
3	EMEN	K1	EM存储器工作使能	低电平有效
4	EMWR	K2	EM存储器写使能	低电平有效
5	EMRD	K3	EM存储器读使能	低电平有效
6	CK	已连	uPC工作脉冲	上升沿打入

实验 1：uPC 加一实验

置控制信号为：

K3(EMRD)	K2(EMWR)	K1(EMEN)	K0(IREN)
1	1	1	1

按一次 STEP 脉冲键，CK 产生一个上升沿，数据 uPC 被加一。

实验 2：uPC 打入实验

二进制开关 K23-K16 用于 DBUS[7:0]的数据输入，置数据 12H，如下：

K23	K22	K21	K20	K19	K18	K17	K16
0	0	0	1	0	0	1	0

置控制信号为：

K3(EMRD)	K2(EMWR)	K1(EMEN)	K0(IREN)
1	0	0	0

当 EMWR, EMEN=0 时, 数据总线 (DBUS) 上的数据被送到指令总线 (IBUS) 上。

按住 STEP 脉冲键, CK 由高变低, 这时寄存器 uPC 的黄色预置指示灯亮, 表明 uPC 被预置。放开 STEP 键, CK 由低变高, 产生一个上升沿, 数据 10H 被写入 uPC 寄存器。

实验七 模型机综合实验（微程序控制器）

在综合实验中，模型机作为一个整体来工作的，所有微程序的控制信号由微程序存储器 uM 输出，而不是由开关输出。在做综合实验之前，先用 8 芯电缆连接 J1 和 J2，使系统处于非手动状态，这样实验仪的监控系统会自动打开 uM 的输出允许，微程序的各控制信号就会接到各寄存器、运算器的控制端口。此综合实验(1~7)使用的指令是模型机的缺省指令系统，系统自动默认装入缺省指令系统 / 非流水微指令系统文件：insfile1.mic。

在做综合实验时，可以用 CPTH 计算机组成原理实验软件输入并修改程序，汇编成机器码，然后下载到实验仪上，由软件控制程序实现单指令执行、单微指令执行、全速执行，同时可以在软件上观察指令或微指令执行过程中数据的走向、各控制信号的状态、各寄存器的值。CPTH 软件的使用方法见附录 1 “CPTH 集成开发环境使用”。也可以用实验仪自带的小键盘和显示屏来输入、修改程序，用键盘控制单指令或单微指令执行，用 LED 或用显示屏观察各寄存器的值。实验仪上的键盘使用方法见附录 2 “实验仪键盘使用”。

在用微程序控制方式做综合实验时，在给实验仪通电前，拔掉实验仪上所有的手工连接的接线，再用 8 芯电缆连接 J1 和 J2，控制方式开关 KC 拨到“微程序控制”方向。若想用 CPTH 软件控制组成原理实验仪，就要启动软件，并用快捷图标的“连接通信口”功能打开设置窗口，选择实验仪连接的串行口，然后再点击“OK”按钮接通到实验仪。

实验 1：数据传送实验/输入输出实验

1. 在 CPTH 软件中的源程序窗口输入下列程序

```
MOV    A, #12h
MOV    A, R0
MOV    A, @R0
MOV    A, 01H
IN
OUT
END
```

2. 将程序另存为 EX1.ASM，将程序汇编成机器码，调试窗口会显示出程序地址、机器码、反汇编指令。

程序地址	机器码	反汇编指令	指令说明
00	7C 12	MOV A, #12	立即数12H送到累加器A
02	70	MOV A, R0	寄存器R0送到累加器A
03	74	MOV A, @R0	R0间址的存储器内容送到累加器A
04	78 01	MOV A, 01	存储器01单元内容送到累加器A
06	C0	IN	端口IN内容输入到累加器A
07	C4	OUT	累加器A内容输出到端口OUT

3. 按快捷图标的 F7，执行“单微指令运行”功能，观察执行每条微指令时，寄存器的输入/输出状态，各控制信号的状态，PC 及 uPC 如何工作。（见 EX1.ASM 程序跟踪结果）

EX1.ASM 程序跟踪结果

助记符	状态	微地址	微程序	数据输出	数据打入	地址输出	运算器	uPC	P C
	T0	00	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
00 MOV A,#12	T1	7C	C7FFF7	存储器值 EM	寄存器 A	PC 输出	A 输出	1	1
	T0	7D	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
02 MOV A,R0	T1	7D	FFF7F7	寄存器值 R?	寄存器 A		A 输出	1	
	T0	71	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
03 MOV A,@RD	T2	74	FF77FF	寄存器值 R?	地址寄存器 MAR		A 输出	1	
	T1	75	D7BFF7	存储器值 EM	寄存器 A	MAR 输出	A 输出	1	
	T0	76	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
04 MOV A,D1	T2	78	C77FFF	存储器值 EM	地址寄存器 MAR	PC 输出	A 输出	1	1
	T1	79	D7BFF7	存储器值 EM	寄存器 A	MAR 输出	A 输出	1	
	T0	7A	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
06 IN	T1	C0	FFF1F7	用户 IN	寄存器 A		A 输出	1	
	T0	C1	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
07 OUT	T1	C4	FFDF9F	ALU 直通	用户 OUT		A 输出	1	
	T0	C5	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1

每个程序的一开始的第一条微指令一定是取指令，此微指令的值为 0CBFFFFH，对应到各个控制位就是 EMRD、PCOE、及 IREN 为低，此三位有效，其它所有位都处于无效状态。在程序第一次运行时或复位后，uPC 和 PC 的值都为 0，PCOE 有效，将 PC 值送到 ABUS，做为程序存储器 EM 的地址，EMRD 信号有效就是从程序存储器 EM 中读出程序指令，IREN 将读出的指令送到 IR 寄存器和 uPC，此微指令的作用就是：从程序存储器 EM 的 0 地址读出程序指令机器码 7CH，并存入 uPC 中作为微程序存储器 uM 的地址，从微程序存储器 uM 的 7CH 单元中读出微控制码 0C7FFF7H，同时 PC 加 1 为读下一条指令或数据做准备。

MOVA, #12: 本指令为两个状态周期。在 T1 状态时，上次读出的指令机器码为 7CH，存入 uPC 中作为微程序存储器 uM 的地址，读出微指令的值为 0C7FFF7H，对应到各个控制位就是 EMRDPCOE、EMEN 及 AEN 为低，处于有效状态，其它控制位为无效状态。由于上条微指令（取指操作）已将 PC 加 1，此时 PCOE 是将加 1 后的 PC 输出到 ABUS 作为程序存储器 EM 的地址，EMRD 就是从程序存储 EM 中读出数据，本指令中读出的数据应为 12H，EMEN 将读出的数据送到 DBUS 总线上，AEN 是将 DBUS 总线上的值存入累加器 A 中。同时 uPC 加 1 为执行下条微指令做准备，PC 加 1 为读取下一条指令做准备。每条指令的最后一条微指令一定是取指令操作，本指令的 T0 状态周期即为取指令，执行上一条微指令时 uPC 已经加 1，按照此 uPC 为地址从微程序存储器 uM 读出的微指令的值为 0CBFFFFH，参照第 1 步的说明，此微指令从程序存储器 EM 中读取指令。

MOVA, R0: 本指令为两个状态周期。在 T1 状态时，由上条取指操作取出的指令机器码为 70H，存入 uPC 后作为微程序地址访问微程序存储器 uM 的 70H 单元，读出微指令的值为 0FFF7F7，各控制位的状态为 RRD、AEN 为低电平为有效状态，RRD 有效表示从寄存器组 R0-R3 中读数送到 DBUS 上，在上条取指令操作时，IREN 将取出的指令机器码 70H 送入 IR 寄存器，而 IR 寄存器的最低两位是用来选择寄存器 R?的，此时 IR 寄存器最低两位为 00，被读出的寄存器为 R0。AEN 有效表示将 DBUS 的数据写到累加器 A 中。同时 uPC

加 1, 为执行下条微指令做准备。本指令的 T0 状态也是取指令, 完成的功能是取出下一条要执行的指令机器码, 并存入 uPC 和 IR 寄存器中。

MOV A, @R0: 本指令为三个状态周期。在 T2 状态时, 由上个取指操作读出的指令机器码为 74H, 打入 uPC 后, 从微程序存储器 74H 单元读出的微指令的值为 0FF77FFH, 有效的控制位为 MAREN 和 RRD, RRD 有效表示从寄存器组 R0-R3 中读出数据送到 DBUS, MAREN 有效表示将数据从 DBUS 总线上打入地址寄存器 MAR。uPC 加 1 取出下条微指令执行。在 T1 状态时, 由 uPC 做为微程序存储器址, 从 uM 的 75H 单元中读出微指令的值为 0D7BFF7H, 其中有效的控制位为 EMRD、EMEN、MAROE 和 AEN。MAROE 表示程序存储器 EM 的地址由地址寄存器 MAR 输出, EMRD 表示从程序存储器 EM 中读出数据, EMEN 表示读出的数据送到地址总线 DBUS 上, AEN 有效表示将数据总线 DBUS 上的值存入累加器 A 中。此状态下 uPC 要加 1, 为取下条微指令做准备。本指令的 T0 状态执行的是取指操作。取指操作详细描述见程序开始部分的取指令的说明。

MOV A, 01: 本指令为三个状态周期。在 T2 状态时, 由上条取指操作取出的指令机器码为 78H, 存入 uPC 和 IR 寄存器后做为微程序存储器 uM 的地址, 读出微指令的值为 0C77FFFH, 相应的有效控制位为 EMRD、PCOE、EMEN 和 MAREN, PCOE 有效表示将 PC 值做为程序存储器 EM 的地址, EMRD 表示从程序存储器中读出数据, 在本指令中此数据值为 01H, EMEN 表示将读出的数据送到 DBUS 总线, MAREN 表示将 DBUS 总线上的数据打入地址寄存器 MAR。uPC 同时加 1, 取出下条微指令准备执行。在 T1 状态时, 由 uPC 做为微程序存储器地址, 从 uM 的 79H 单元中读出微指令的值为 0D7BFF7H, 可以参见上条指令的 T1 状态, 此微指令的所完成的功能是, 以 MAR 的值做为程序存储器的地址, 读出数据并送到数据总线 DBUS, 同时将此数据存入累加器 A 中。uPC 加 1 取出下条微指令准备执行。在 T0 状态, 微指令执行取指令操作。

IN: 本指令分两个状态周期。在 T1 状态时, 由上次取指操作取出的指令机器码为 0C0H, 以此做为微地址从 uM 中取出的微指令值为 0FFFF17H, 有效控制位为 AEN、X2X1X0=000, 因为 X2、X1、X0 为低, 被选中的寄存器为输入端口 IN, 也就是说, 输入端口 IN 上的数据被允许送到数据总线 DBUS 上, AEN 有效表示将此数据打入累加器 A 中。同时 uPC 加 1 取出下条微指令准备执行。在 T0 状态, 微指令执行的是取指令操作, 取出下条指令准备执行。

OUT: 本指令分两个状态周期。在 T1 状态, 由上次取出的指令机器码为 0C4H, 以此为微地址从微程序存储器 uM 中读出的微指令为 0FFDF9FH, 有效控制位为 OUTEN、X2X1X0=100(二进制), S2S1S0=111(二进制), S2S1S0=111 表示运算器做“ALU 直通”运算, 也就是累加器不做任何运算, 直接输出结果, 而 X2X1X0=100 表示运算器的结果不移位直接输出到数据总线 DBUS, OUTEN 有效表示将数据总线上的数据打入输出端口寄存器 OUT 内。uPC 加 1, 取出下条微指令准备执行。在 T0 状态, 微指令执行的是取指操作, 取出下条将要执行的指令。

实验 2: 数据运算实验 (加/减/与/或)

1. 在 CPTH 软件中的源程序窗口输入下列程序

ADDC	A, R1
SUB	A, @R1
AND	A, #55H
OR	A, 02H
END	

2. 将程序另存为 EX2.ASM, 将程序汇编成机器码, 调试窗口会显示出程序地址、机器码、反汇编指令。

程序地址	机器码	反汇编指令	指令说明
00	21	ADDC A, R1	累加器 A 的值加上寄存器 R1 加进位
01	35	SUB A, @R1	累加器 A 减去 R1 间址的存储器内容
02	5C 55	AND A, #55	累加器 A 逻辑与立即数 55H
04	68 02	OR A, 02	累加器 A 逻辑或存储器 02 单元的内容

3. 按快捷图标 F7, 执行“单微指令运行”功能, 观察执行每条微指令时, 寄存器的输入/输出状态, 各控制信号的状态, PC 及 uPC 如何工作。(见“EX2.ASM 程序跟踪结果”详细介绍)

4. 在了解数据运算的原理, 可以加上一些数据传输指令给累加器 A 或寄存器 R?赋值, 再运算, 并观察运算结果。

EX2.ASM 程序跟踪结果

助记符	状态	微地址	微程序	数据输出	数据打入	地址输出	运算器	uPC	P C
	T0	00	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
00 ADDC A, R1	T2	20	FFF7EF	寄存器值 R?	寄存器 W		A 输出	1	
	T1	21	FFFE94	ALU 直通	寄存器 A 标志位 C,Z		带进位加运算	1	
	T0	22	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
01 SUB A,@R1	T3	34	FF77FF	寄存器值 R?	地址寄存器 MAR		A 输出	1	
	T2	35	D7BFEF	存储器值 EM	寄存器 W	MAR 输出	A 输出	1	
	T1	36	FFFE91	ALU 直通	寄存器 A 标志位 C,Z		减运算	1	
	T0	37	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
02 AND A,#55	T2	5C	C7FFE9	存储器值 EM	寄存器 W	PC 输出	A 输出	1	1
	T1	5D	FFFE93	ALU 直通	寄存器 A 标志位 C,Z		与运算	1	
	T0	5E	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
04 OR A,02	T3	68	C77FFF	存储器值 EM	地址寄存器 MAR	PC 输出	A 输出	1	1
	T2	69	D7BFEF	存储器值 EM	寄存器 W	MAR 输出	A 输出	1	
	T1	6A	FFFE92	ALU 直通	寄存器 A 标志位 C,Z		或运算	1	
	T0	6B	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1

程序的开始执行一条取指的微指令, 读入程序第一条指令。

ADDC A, R1: 本指令为三个状态周期。在 T2 状态, 由上次取指操作取出的指令码为 21H, 由 IREN 存入指令寄存器 IR, 最低两位为 01(二进制), 选择寄存器 R1, 指令码由于 IREN 打入 uPC 时, 忽略掉指令的最低两位, 而将 uPC 的最低两位置成 00, uPC 的值为 20H, 访问微程序存储器的 20H 单元, 读出微指令值为 0FFF7EFH, 有效位为 RRD 及 WEN, 就是将 R1 内容送到工作寄存器 W, uPC 加 1 取出下条微指令在 T1 状态, 读出的微指令值为 0FFFE94H, 有效位为 FEN 和 AEN, FEN 完成的操作是将标志位存入标志寄存器 F(ALU 内

部), X2X1X0 选择“ALU 直通”到数据总线 DBUS, S2S1S0 选择的运算操作为“带进位的加法运算”, AEN 将 DBUS 上的数据存入累加器 A。在 T0 状态, 取出下条将要执行的指令。

SUB A, @R1: 本指令有四个状态周期。在 T3 状态, 上次取出的指令码为 35H, 最低两位用于寻址 R1 寄存器, uPC 的最低两位位置 0, 来访问 uM 的 34H 单元的微指令, 读出值为 0FF77FFH, 将 R1 的值存入 MAR。在 T2 状态, 微指令为 0D7BFEFH, 表示用 MAR 做为地址从 EM 中读出数据送到 DBUS 再存入 W 中。在 T1 状态微指令为 0FFFE91H, 表示 ALU 做“减运算”, 其结果直通到 DBUS, 再存入 A 中, 同时保存标志位。T0 状态为取指操作。

AND A, #55: 本指令为三个状态周期。在 T2 状态, 微指令值为 0C7FFE FH, 表示以 PC 做为地址, 从 EM 中读出数据送到 DBUS, 再将 DBUS 数据存入 W 中。在 T1 状态, 微指令为 0FFFE93H, 表示 A 和 W 做“逻辑与”运算, 结果直通到 DBUS, 再存入 A 中, 并保存标志位。

OR A, 02: 本指令有四个状态周期。在 T3 状态, 微指令为 0C77FFFH, 表示以 PC 做为地址, 从 EM 中读出数据送到 DBUS, 并存入 MAR 中。在 T2 状态, 微指令为 0D7BFE FH, 表示以 MAR 做为地址, 从 EM 中读出数据送到 DBUS, 并存入 W 中。在 T1 状态微指令为 0FFFE92H, 表示 A 和 W 做“逻辑或”运算, 结果“直通”到 DBUS 并存入 A 中。T0 状态为取指操作。

实验 3: 指令流水实验

指令流水操作, 就是在微指令执行的过程中, 在 T1 状态, 如果 ABUS 和 IBUS 空闲, 则可以利用这个空闲来进行预取指令, 让 ABUS、IBUS 和 DBUS 并行工作, 实现指令的流水工作。我们已经建立了一套可流水操作的指令/微指令系统。用户可调入这个指令/微指令系统进行实验。为了方便比较, 我们仍用实验 1 的程序 EX1.ASM, 其它指令用户可以自己做实验来比较、验证。

1. 在 CPTH 软件中, 用菜单的[文件|调入指令系统/微程序]功能, 打开 CPTH 下的“INSFILE2.MIC”, 这就是流水操作的指令/微指令系统。

2. 在 CPTH 软件中, 用菜单的[文件|打开文件]功能, 打开 CPTH 下的“EX1.ASM”源程序。编译后产生的机器码与实验 1 相同。

3. 按快捷图标 F7, 执行“单微指令运行”功能, 观察执行每条微指令时, 寄存器的输入/输出状态, 各控制信号的状态, PC 及 uPC 如何工作。特别是在每条指令的 T0 状态周期, 取指操作是否和其它指令并行执行。(见“EX1.ASM 程序流水操作跟踪结果”详细介绍)

EX1.ASM 程序流水操作跟踪结果

助记符	状态	微地址	微程序	数据输出	数据输入	地址输出	运算器	uPC	PC
	T0	00	CBFFFF		指令寄存器IR	PC输出	A输出	写入	1
00 MOV A, #12	T1	7C	C7FFF7	存储器值EM	寄存器A	PC输出	A输出	1	1
	T0	7D	CBFFFF		指令寄存器IR	PC输出	A输出	写入	1
02 MOV A, R0	T0	70	CBF7F7	寄存器值R?	寄存器A、指令寄存器IR	PC输出	A输出	写入	1
03 MOV A, @R0	T2	74	FF77FF	寄存器值R?	地址寄存器MAR		A输出	1	
	T1	75	D7BFF7	存储器值EM	寄存器A	MAR输出	A输出	1	
	T0	76	CBFFFF		指令寄存器IR	PC输出	A输出	写入	1
04 MOV A, 01	T2	78	C77FFF	存储器值EM	地址寄存器MAR	PC输出	A输出	1	1
	T1	79	D7BFF7	存储器值EM	寄存器A	MAR输出	A输出	1	
	T0	7A	CBFFFF		指令寄存器IR	PC输出	A输出	写入	1
06 IN	T0	C0	CBFF17	用户IN	寄存器A、指令寄存器IR	PC输出	A输出	写入	1
07 OUT	T0	C4	CBDF9F	ALU直通	指令寄存器IR、OUT	PC输出	A输出	写入	1

每个程序的一开始的第一条微指令一定是取指令，取出下条将要执行的指令。

MOVA, #12: 本指令为两个状态周期。在 T1 状态时，从程序存储器 EM 中读出数据送到累加器 A，ABUS 被占用，所以取指操作不能与数据总线 DBUS 上的操作并行执行。本指令的 T0 状态为正常的取指令操作。

MOVA, R0: 由于取指操作与数据总线可并行工作，本指令只有 1 个状态周期。由上条取指操作取出的指令机器码为 70H，存入 uPC 后做为微程序地址访问微程序存储器 uM 的 70H 单元，读出微指令的值为 0CBF7F7H，有效控制位为 EMRD、PCOE、IREN、RRD、AEN，由于 IR1、IR0 的值为 00，与 RRD 信号组合表示从 R0 中读出数据到 DBUS 总线，AEN 将 DBUS 上的值存入累加器 A，EMRDPCOE 和 IREN 有效表示以 PC 做为地址从 EM 中读出下条指令，并存入 IR 和 uPC 中，PC 加 1。

MOVA, @R0: 本指令为三个状态周期。在 T2 状态时，将 R0 的值存入地址寄存器 MAR。在 T1 状态时，以 MAR 为地址读出数据并送到累加器 A 中。在 T0 状态，取出下条将要执行指令。由于 ABUS 不空闲，所以取指操作不能并行工作。

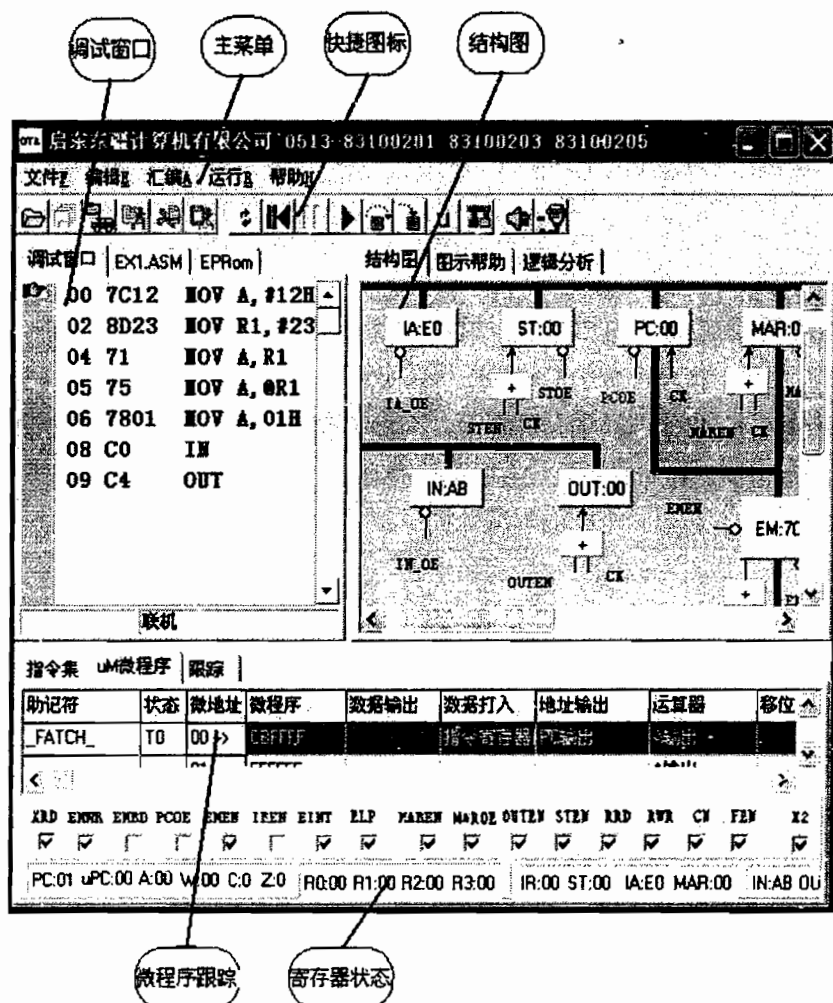
MOV A, 01: 本指令为三个状态周期。在 T2 状态时，以 PC 为地址从 EM 中读出数据存到 MAR 中，在 T1 状态，以 MAR 为地址从 EM 中读出数据存入累加器 A。T0 为取指操作。由于 ABUS 不空闲，取指操作不能并行执行。

IN: 本指令为 1 个状态周期。取指操作和输出操作可并行执行。由上次取指操作取出的指令机器码为 0C0H，以此做为微地址从 uM 中取出的微指令值为 0CBFF17H，有效控制位为 EMRD、PCOE、IREN 和 AEN、X2X1X0=000(二进制)表示从输入寄存器 IN 读数据送到 DBUS，AEN 表示将此数据存入 A，EMRD、PCOE 和 IREN 有效表示以 PC 为地址从 EM 中读出指令存入 IR 和 uPC 中，PC 加 1。

OUT: 本指令有 1 个状态周期。取指操作和输出操作并行完成。由上次取出微指令值为 0CBDF9FH，有效控制位为 EMRD、PCOE、IREN、OUTEN、X2X1X0=100(二进制)，S2S1S0=111(二进制)，S2S1S0=111 表示运算器做“ALU 直通”运算，也就是累加器不做任

何运算，直接输出结果，而 $X_2X_1X_0=100$ 表示运算器的结果不移位直接输出到数据总线 DBUS，OUTEN 有效表示将数据总线上的数据打入输出端口寄存器 OUT 内。与此同时，EMRD、PCOE、IREN 表示以 PC 为地址从 EM 中读出下条指令，存 IR 和 uPC 中，PC 加 1。

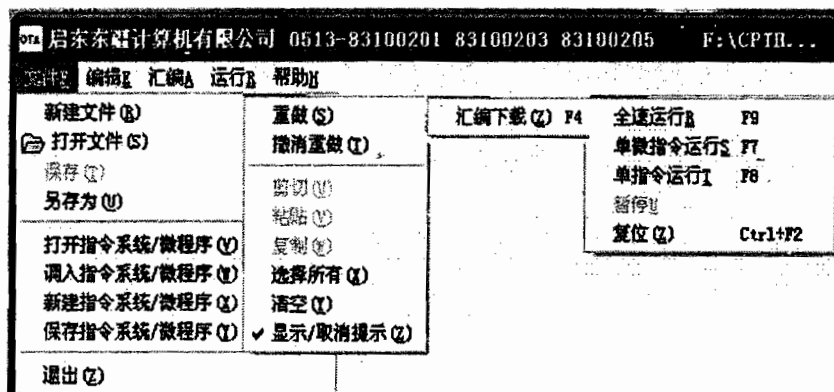
附录 1: CPTH 集成开发环境使用



CPTH 集成调试软件界面分六部分:

- 1) 主菜单区 实现实验仪的各项功能的菜单, 包括[文件][编辑][汇编][运行][帮助]五大项, 各项下面做详细介绍。
- 2) 快捷图标区 快速实现各项功能按键。
- 3) 调试窗口区 在此区域有 调试窗口、源程序窗口、EM 程序代码窗口。调试窗口显示程序编译后的机器码及反汇编的程序; 源程序用于输入、显示、编辑汇编源程序; EM 程序代码窗口用数据方式机器码。
- 4) 结构图区 结构图能结构化显示模型机的各部件, 以及运行时数据走向寄存器值, 图示帮助进行图示化实验指导。逻辑波形图能显示模型机运行时所有信号的时序。
- 5) 微程序/跟踪区 微程序表格用来显示程序运行时微程序的时序, 及每个时钟脉冲各控制位的状态, 跟踪表用来记录显示程序及微程序执行的轨迹, 指令系统显示指令集。
- 6) 寄存器状态区 用来显示程序执行时各内部寄存器的值。

1) 主菜单



主菜单分[文件][编辑][汇编][运行][帮助]五部分

[文件 | 打开文件] 打开已有的汇编程序或文本文件。

[文件 | 保存文件] 将修改过的文件保存。不论是汇编源程序还是其它文本文件，只要被修改过，就会被全部保存。

[文件 | 新建文件] 新建一个空的汇编源程序。

[文件 | 另存为...] 将修改过的程序换名保存。

[文件 | 打开指令系统/微程序] 打开设计好的指令系统和微程序文件，用于修改指令系统和微程序文件。

[文件 | 调入指令系统/微程序] 调入设计好的指令系统和微程序定义，用于编译汇编源程序。

[文件 | 退出] 退出集成开发环境。

[编辑 | 重做] 撤消 / 恢复上次输入的文本。

[编辑 | 剪切] 将选中的文本剪切到剪贴板上。

[编辑 | 复制] 将选中的文本复制到剪贴板上。

[编辑 | 粘贴] 从剪贴板上将文本粘贴到光标处。

[编辑 | 全选] 全部选中文本

[汇编 | 汇编] 将汇编程序汇编成机器码并下载。

[运行 | 全速执行] 全速执行程序。

[运行 | 单指令执行] 每步执行一条汇编程序指令。

[运行 | 单微指令执行] 每步执行一条微程序指令。

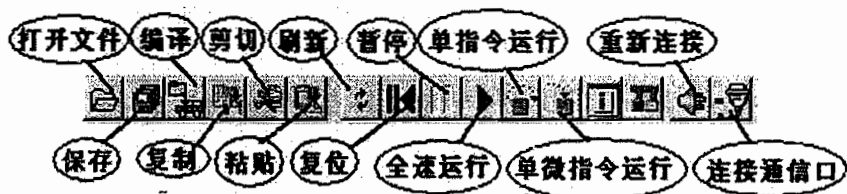
[运行 | 暂停] 暂停程序的全速执行。

[运行 | 复位] 将程序指针复位到程序起始处。

[帮助 | 关于] 有关 CPTH 计算机组成原理实验仪及软件的说明。

[帮助 | 帮助] 软件使用帮助。

2) 快捷键图标

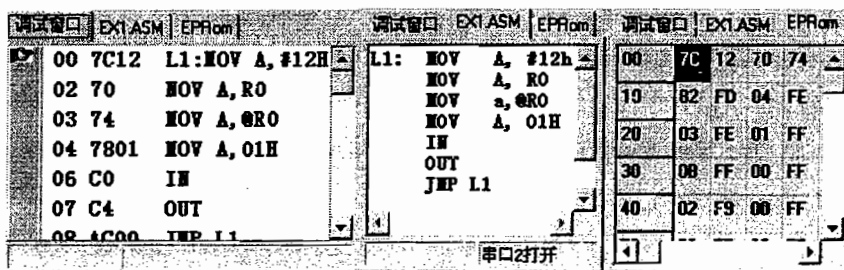


图标的“刷新”功能就是在联机过程中刷新各寄存器，程序存储器，微程序存储器的值。以便观察实验仪的各参数内容。

文件的“打开”、“保存”功能与主菜单的相应功能一样。

文件的编辑功能，执行控制功能，其它快捷命令与主菜单也相同。

3)调试窗口区



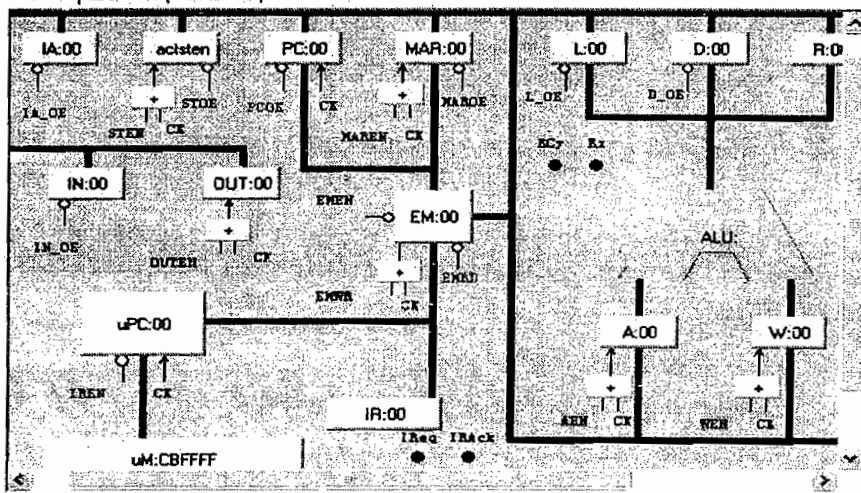
调试窗口区分三个窗口：调试窗口、源程序窗口、EM 程序窗口

源程序窗口用于输入、修改程序。在[文件]菜单中打开一个以“*.ASM”为后缀的文件时，系统认为此文件为源程序，其内容会在源程序窗口显示，并可以修改，然后编译。若再次打开以“*.ASM”后缀的文件，则新文件将旧文件覆盖，在源程序窗口只显示最新打开的汇编源程序。在[文件]菜单中，使用“新建文件”功能，会清除源程序窗口的内容，让用户重新输入新的程序。

调试窗口用于显示程序地址、机器码、反汇编后的程序。

EM 程序窗口以十六进制数据的形式显示程序编译后的机器码。可以直接输入数值来修改机器码。

4) 结构图区

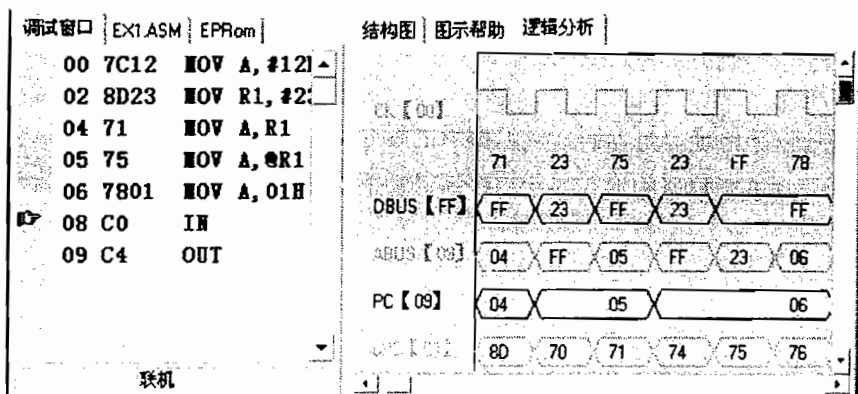


结构图区分三种窗口，结构图窗口、图示帮助窗口、逻辑分析窗口。

结构图窗口显示模型机的内部结构，包括各种寄存器（A、W、R0-R3、MAR、IR、ST、L、D、R）、运算器（ALU）、程序指针（PC）、程序存储器（EM）、微程序指针（uPC）、微程序存储器（uM）及各种状态位（RCy、Rz、IReq、IAck），在程序单步运行时，可以在结构图上看到数据的走向及寄存器的输入输出状态。其中 DBUS 为数据总线、ABUS 为地址总线、IBUS 为指令总线。RT1、RT0 显示的将要执行的指令的第几个时钟周期。本模型机最多有四个时钟周期，用 RT1、RT0 的 11、10、01、00 四个状态表示。见上图。

图示帮助进行图示化实验指导。

逻辑分析窗口显示的是在指令执行时，各种信号的时序波形，包括所有寄存器、所有的控制信号在不同时钟状态下的值，可以直观地看到各种信号彼此之间的先后时序关系。



5) 指令/微程序/跟踪窗口




此区分三页：指令集窗口、微程序窗口、跟踪窗口。

指令集窗口用于显示指令系统和设计用户自己的指令系统。各条指令相应的微程序设计在“uM 微程序”窗口中设计（见 uM 微程序窗口）

助记符	机器码1	机器码2	机器码3	注释
ADD A, R?	000100xx 10-13			将寄存器R?的值加入累加器
ADD A, @R?	000101xx 14-17			将间址存储器的值加入累加
ADD A, MM	000110xx 18-1B	MM		将存储器MM地址的值加入

uM 微程序窗口用于观察每条指令所对应的微程序的执行过程，以及微代码的状态。在此窗口中，可以看到数据是从何寄存器输出的、数据输入到何寄存器、地址是由 PC 输出还是由 MAR 输出、运算器在做何种运算、如何移位、uPC 及 PC 如何工作等等。可以通过改变窗口下方的微代码的各个控制位的方式来重新设计微程序，与“指令系统”窗口的指令修改相结合，可以设计自己的指令。

可以将鼠标移到相应的程序行或微程序行来显示执行该指令或微指令时，各寄存器、控制位的状态。

指令集		uM微程序	跟踪									
助记符	状态	微地址	微程序	数据输出	数据打入	地址输出	运算器	移位控制	uPC	PC		
		01	FFFFFF				A输出		+1			
		02	FFFFFF				A输出		+1			
		03	FFFFFF				A输出		+1			
INC #	T2	04	EF7FFF	指令寄存			PC输出	A输出	写入	+1		
												
XRD ENWR EMRD PCOE EMEN IREN EINT ELP MAREN MAROE OUTEN STEW XRD RWR CY FEY												
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>												
PC:00 uPC:00 A:00 W:00 C:0 Z:0				R0:00 R1:00 R2:00 R3:00			IR:00 ST:00 IA:E0 MAR:00			IN		

跟踪窗口显示程序执行过程的轨迹，包括每条被执行的指令、微指令，以及微指令执行时，各控制位、各个寄存器的状态。

指令集	uM微程序	跟踪										
助记符	状态	微地址	微程序	数据输出	数据打入	地址输出	运算器	移位控制	uPC	PC		
MOVA, #11	T1	7C	C7FFF7	存储器值EM	寄存器A	PC输出	A输出		+1	+1		
	T0	7D	CBFFF7	指令寄存器IR		PC输出	A输出		写入	+1		
MOVA, R?	T1	7D	FFF7F7	寄存器值R?	寄存器A		无输出		+1			

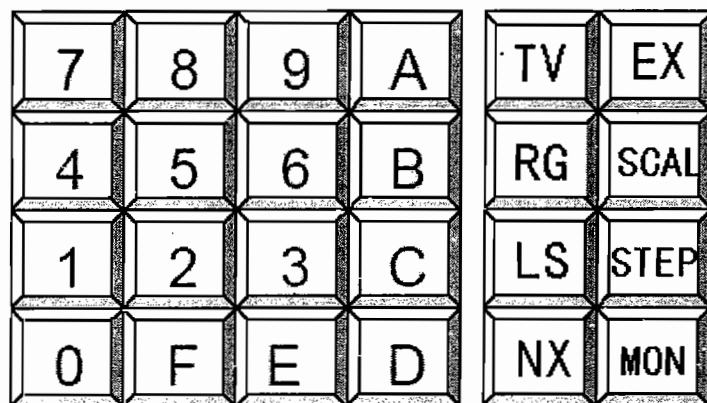
6) 寄存器状态

寄存器状态区显示程序执行时，各内部寄存器的值。

PC:02 uPC:7D A:12 W:00 C:0 Z:1	R0:00 R1:00 R2:00 R3:FC	IR:7C ST:00 IA:E0 MAR
--------------------------------	-------------------------	-----------------------

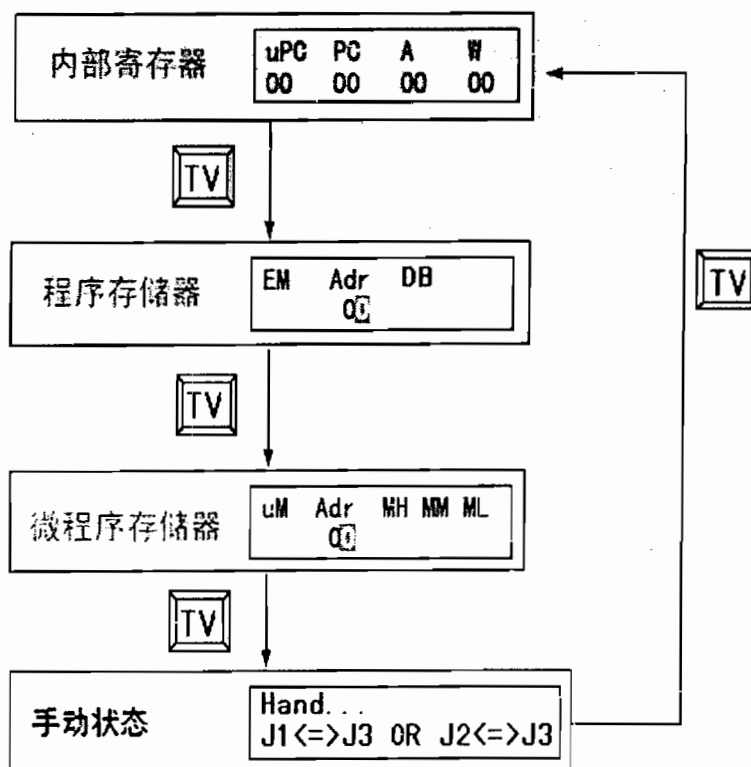
附录 2：实验仪键盘使用

DJ-CPTH 计算机组成原理实验仪除了可以连在 PC 机上调试程序，也可以用实验仪上自带的键盘输入程序及微程序，并可以单步调试程序和微程序，在显示屏上观察、修改各内部寄存器的值，编辑修改程序和微程序存储器。



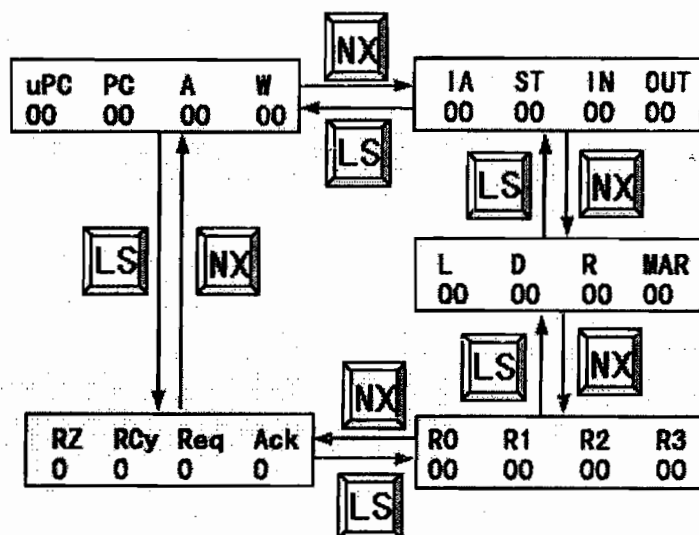
显示屏的显示内容分四个主菜单：

- 1、观察和修改内部寄存器；
- 2、观察和修改程序存储器；
- 3、观察和修改微程序存储器；
- 4、手动状态。四个主菜单用 TV/ME 键切换。如下图：



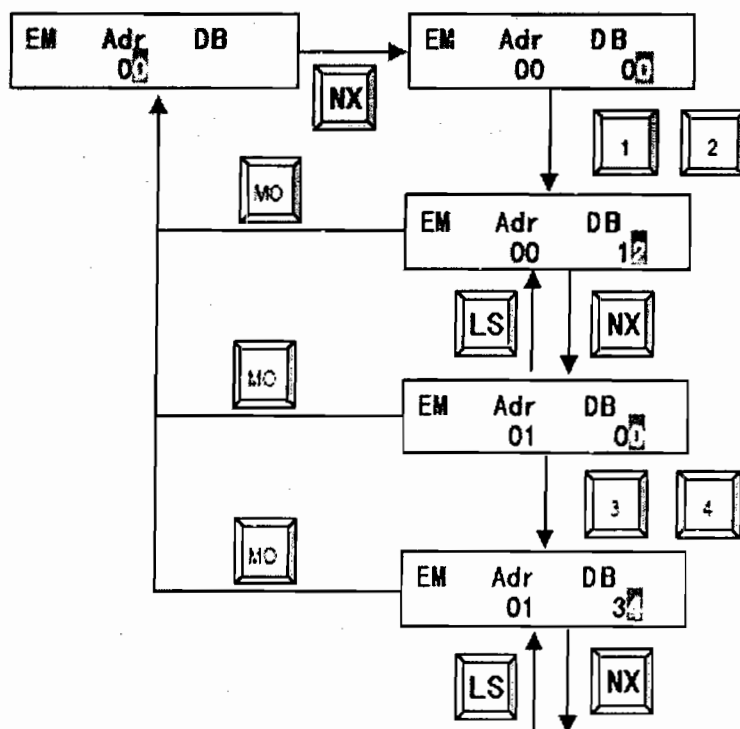
1、观察、修改内部寄存器：

内部寄存器的内容分五页显示，用 LAST 或 NEXT 键向前或向后翻页，可以显示所有内部寄存器值，输入数字可修改非只读寄存器值。见下图：



2、观察、修改程序存储器内容:

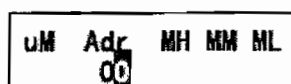
显示屏显示如下图, 其中“Adr”表示程序存储器地址, “DB”表示该地址中数据。光标初始停在“Adr”处, 此时可以用数字键输入想要修改的程序地址, 也可以用 NEXT 和 LAST 键将光标移到“DB”处, 输入或修改此地址中的数据。再次按 NEXT 或 LAST 键可自动将地址+1 或将地址-1, 并可用数字键修改数据。按 MON 键可以回到输入地址 00 的状态。见下图。



3. 观察、修改微程序存储器内容:

微程序存储器数据的观察、修改与上面程序存储器的观察修改方法相似, 不同的是微程序要输入 3 个字节, 而程序存储器的修改只要输入 1 个字节。微程序观察修改的显示屏显

示如下图，其中“Adr”表示微程序地址，“MH”表示微程序的高字节，“MM”表示微程序的中字节，“ML”表示微程序的低字节。



使用实验仪键盘可以用三种方法调试程序，程序单步、微程序单步、全速执行。当用键盘调试程序时，显示屏显示寄存器第一页的内容。

[STEP] 为微程序单步执行键，每次按下此键，就执行一个微程序指令，同时显示屏显示微程序计数器、程序计数器、A 寄存器、W 寄存器的值。可以通过 NX 或 LS 键翻页观察其它寄存器的值，观察各个寄存器的输出和输入灯的状态。

[SCAL]为程序单步执行键，每次按下此键，就执行一条程序指令，同时显示屏显示微程序计数器、程序计数器、A 寄存器、W 寄存器的值。可以通过 NX 或 LS 键翻页观察其它寄存器的值。

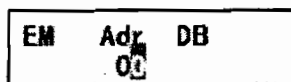
[EX]为全速执行键，按下此键时，程序就会全速执行，按键盘任一键暂停程序执行，并且显示当前寄存器的值。

[RG]为中断请求键，按下此键时，会产生一个中断请求信号 INT。

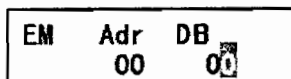
[RST]复位键，按下此键，程序中止运行，所有寄存器清零（IR 除外），程序指针回到 0 地址。

举例：用键盘输入以下程序代码：

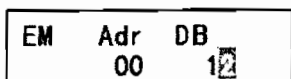
1. 按 TV 键，直到显示屏显示内容为



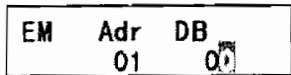
2. 按 NX 键，光标移到“DB”下，显示屏为



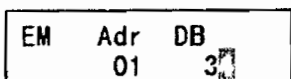
3. 按 1,2 两个数字键，显示屏为



4. 按 NX 键，地址+1，显示屏为



5. 按 3,4 两个数字键，显示屏为



重复 4、5 两步，直到输入所有的程序代码。

在第 1 步时，光标停在“Adr”处，可以按数字键 0...F 输入要修改的程序存储器的地址，然后再按 NEXT 键输入程序代码。如果光标移到“DB”下，而此时又想改变地址，可以按 MON 键，将光标移回到“Adr”处，按数字键输入地址。输入微程序代码的方法与此相似，不同的是程序只需输入两个数字，即一个字节，而微程序要输入 6 个数字，即三

个字节。如果多于 6 个数字会自动从右向左移动光标。如果输入不足 6 个数字就用 NEXT 或 LAST 翻页，则只有被改动的几个数字有效，其它数字不变。

4. 用小键盘调试实验一

程序地址	机器码	反汇编指令	指令说明
00	7C 12	MOV A, #12	立即数12H送到累加器A
02	70	MOV A, R0	寄存器R0送到累加器A
03	74	MOV A, @R0	间址的存储器内容送到累加器A
04	78 01	MOV A, 01	存储器01单元内容送到累加器A
06	C0	IN	端口IN内容输入到累加器A
07	C4	OUT	累加器A内容输出到端口OUT

一：输入机器码

按 TV 键选择 EM

顺序输入机器码：7C 12 70 74 78 01 C0 C4

输完机器码后按 RST 复位

二：单步执行微程序

按 RST 复位键后，PC=0，uPC=0

uM 输出 24 位微程序：CB FF FF 此微指令为取指指令

第一条微指令

按一次 STEP 键，完成一个时钟，此时：

PC 值为 01（时钟上升沿 PC+1）

IR 值为 7C，uPC 值为 7C（指令码）

uM 输出为：C7 FF F7（EM 值送 A）

第二条微指令

按一次 STEP 键，完成一个时钟，此时：

PC 值为 02（时钟上升沿 PC+1）

A 值为 12

uPC 值为 7D（时钟上升沿 uPC+1）

uM 输出为：CB FF FF（取指指令）

第三条微指令

按一次 STEP 键，完成一个时钟，此时：

PC 值为 03（时钟上升沿 PC+1）

IR 值为 70，uPC 值为 70（指令码）

uM 输出为：FF F7 F7（R? 值送 A）

第四条微指令

按一次 STEP 键，完成一个时钟，此时：

PC 值为 03（时钟上升沿 PC+1）

A 值为 00

uPC 值为 71（时钟上升沿 uPC+1）

uM 输出为：CB FF FF（取指指令）

第五条微指令

按一次 STEP 键，完成一个时钟，此时：

PC 值为 04（时钟上升沿 PC+1）

IR 值为 74, uPC 值为 74 (指令码)

uM 输出为: FF 77 FF (R? 值送 MAR)

第六条微指令

按一次 STEP 键, 完成一个时钟, 此时:

MAR 值为 00

uPC 值为 75 (时钟上升沿 uPC+1)

uM 输出为: D7 BF F7 (EM 值送 A)

第七条微指令

按一次 STEP 键, 完成一个时钟, 此时:

A 值为 7C

uPC 值为 76

uM 输出为: CB FF FF (取指)

用同样的方法执行余下的指令。也可以用 SCAL 或 EX 键执行指令。

在做分部模块实验时, 实验仪键盘 0... F, NX, LS 不起作用, 显示屏显示内容为 8 芯电缆的连接方式。例如显示屏显示内容如下: 表示手动方式, J1 通过 8 芯电缆接到 J3, 或 J2 通过 8 芯电缆接到 J3。

Hand... J1<=>J3 OR J2<=>J3
