

计算机组成原理 实验报告

xx 大学计算机科学与技术学院

姓 名	<u>Banban</u>
专 业	<u>计 算 机 科 学 与 技 术</u>
班 级	<u></u>
学 号	<u></u>
指导教师	<u></u>

2022 年 6 月 2 日

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，将数据写入累加器 A 和工作寄存器 W，利用开关控制 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，有 8 中运算，通过 S2, S1, S0 来控制，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。

S2	S1	S0	功能说明
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

- **实验步骤：**

(1) 照下表连接线路

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	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	选通 W	低电平有效
7	CyIN	K5	运算器进位输入	
8	CK	已连	ALU 工作脉冲	上升沿打入

(2) 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，进入手动状态（液晶屏幕上有“Hand……”显示）。

(3) 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，设置数据 55H

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

置控制信号为：

K5 (CyIN)	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。

(4) 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，设置数据 55H

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

置控制信号为：

K5 (CyIN)	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。

(5) 按下表设置控制信号，填写实验结果，并检验运算器的运算结果的正确性。

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

注意观察实验过程，回答以下问题：

运算器在加上控制信号及数据 (A,W) 后，结果什么时候产生？需不需要时钟？

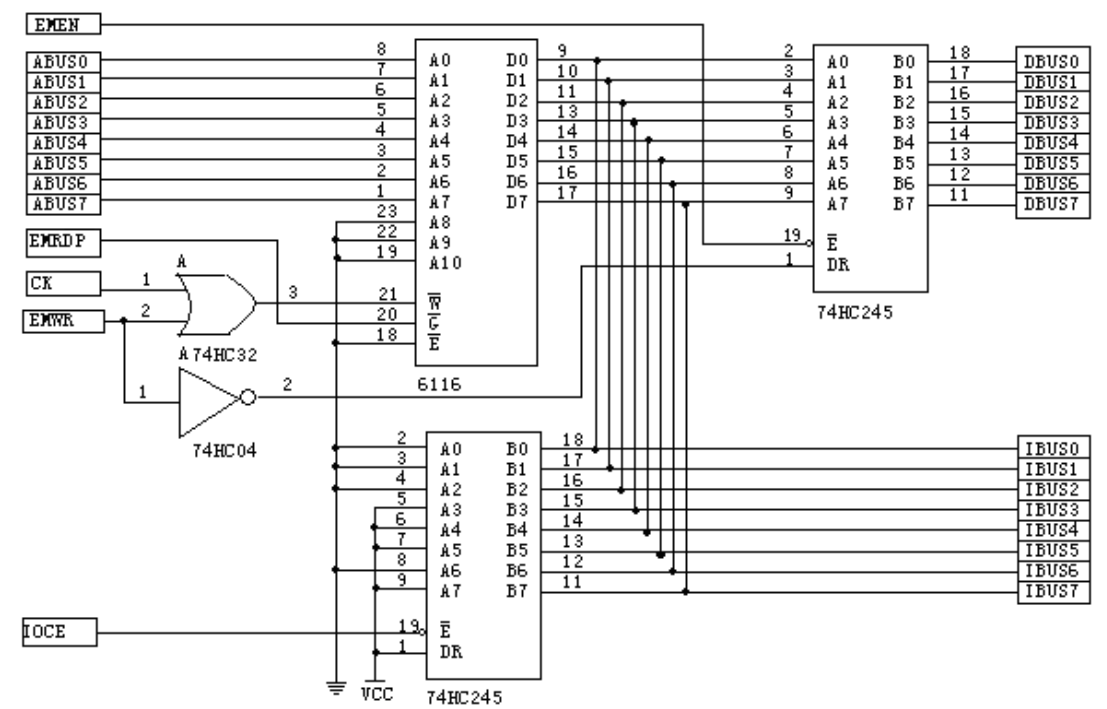
运算器在加上控制信号及数据 (A,W) 后，结果立即产生，不需要时钟。

实验四 存储器 EM 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，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 的数据输出直接接到指令总线 IBUS，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。



EM 原理图

- **实验步骤：**

(1) 照下表连接线路

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

(2) 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, 进入手动状态 (液晶屏幕上有 “Hand……” 显示)。

实验内容 1: PC/MAR 输出地址选择

如下表置控制信号, 观察地址输出指示灯的情况, 并填写下表:

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

实验内容 2: 存储器 EM 写实验

1、向存储单元 EM[0]中写入数据 11H

(1) 将地址 0 写入 MAR

二进制开关 K23~K16 用于 DBUS 【7…0】的数据输入, 设置数据 00H

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

置控制信号为:

K6	K5	K4	K3	K2	K1	K0
1	1	1	0	1	1	1

按 STEP 键, 将地址 0 写入 MAR

(2) 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，设置数据 00H

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

置控制信号为：

K6	K5	K4	K3	K2	K1	K0
1	1	0	1	0	1	0

按 STEP 键，数据 11H 写入 EM[0]

1、向存储单元 EM[1]中写入数据 22H

(1) 将地址 1 写入 MAR

二进制开关 K23~K16 用于 DBUS 【7...0】的数据输入，设置数据 00H

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

置控制信号为：

K6	K5	K4	K3	K2	K1	K0
1	1	1	0	1	1	1

按 STEP 键，将地址 1 写入 MAR

(2) 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，设置数据 00H

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

置控制信号为：

K6	K5	K4	K3	K2	K1	K0
1	1	0	1	0	1	0

按 STEP 键，数据 22H 写入 EM[1]

实验内容 3：存储器 EM 读实验

1、读存储器 EM[0]单元信息

(1) 将地址 0 写入 MAR

二进制开关 K23~K16 用于 DBUS 【7...0】的数据输入，设置数据 00H

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

置控制信号为：

K6	K5	K4	K3	K2	K1	K0
1	1	1	0	1	1	1

按 STEP 键，将地址 0 写入 MAR

(2) 读 EM[0]

置控制信号为：

K6	K5	K4	K3	K2	K1	K0
1	1	0	1	1	0	1

EM[0]被读出，请记录下结果，并比较是否如写入的数据相一致？

EM[0]：11H，与写入数据一致。

2、读存储器 EM[1]单元信息

(1) 将地址 1 写入 MAR

二进制开关 K23~K16 用于 DBUS 【7...0】的数据输入，设置数据 00H

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

置控制信号为：

K6	K5	K4	K3	K2	K1	K0
1	1	1	0	1	1	1

按 STEP 键，将地址 1 写入 MAR

(2) 读 EM[1]

置控制信号为：

K6	K5	K4	K3	K2	K1	K0
1	1	0	1	1	0	1

EM[1]被读出，请记录下结果，并比较是否如写入的数据相一致？

EM[1]：22H，与写入数据一致。

实验五 微程序存储器 uEM 实验

- **实验要求：**利用 CPTH 实验仪的开关作为控制信号，实现微程序存储器 uEM 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 uEM 的控制方法。

- **实验电路：**

存储器 uEM 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。存储器 uEM 始终输出 uPC 指定地址单元的数据。

- **实验步骤：**

照下表连接线路

连接	信号孔	接入孔	作用	状态说明
1	J1 座	J2 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

实验内容 1：使用试验仪小键盘输入 uEM

输入操作步骤如下：

- 1、连接 J1，J2
- 2、打开电源
- 3、K0 置 0
- 4、按 TV/ME 键，选择 uEM
- 5、输入两位地址，00
- 6、按 NEXT，进入微程序修改
- 7、按六位微程序数据
- 8、按 NEXT 选择下个地址/按 LAST 选择上个地址
- 9、重复 6,7 步输入微程序
- 10、按 RST 结束

按上述步骤将下表中的微程序输入 uEM

微地址	微程序	微地址	微程序	微地址	微程序
00	CBFFFF	07	D7BFF7	0D	CBFFFF
01	C7FFF7	08	CBFFFF	0E	FFDF9F

02	CBFFFF	09	C77FFF	0F	CBFFFF
03	FFF7F7	0A	D7BFF7	10	FFFE91
05	CBFFFF	0B	CBFFFF	11	D7BFFF
06	FF77FF	0C	FFFF17	12	CBFFFF

实验内容 2：微程序存储器 uEM 读出

1、照下表连接线路

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

2、置控制信号为：K0 为 1

转换到手动模式，连接 J2，J3

通过 STEP 键调节 UPC 的值可以查看不同微地址下的微程序（通过 uEM 下方的 24 个信号灯显示）

uEM 输出 uEM[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。

uEM 输出 uEM[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。

uEM 输出 uEM[2]的数据。

仔细观察每改变一次微地址 uEM 的输出，并填写下表。

● 实验结果：

微地址	微程序	微地址	微程序	微地址	微程序
00	<u>CBFFFF</u>	07	<u>D7BFF7</u>	0D	<u>CBFFFF</u>
01	<u>C77FFF</u>	08	<u>CBFFFF</u>	0E	<u>FFDF9F</u>
02	<u>CBFFFF</u>	09	<u>C77FFF</u>	0F	<u>CBFFFF</u>
03	<u>FFF7F7</u>	0A	<u>D7BFF7</u>	10	<u>FFFE91</u>
05	<u>CBFFFF</u>	0B	<u>CBFFFF</u>	11	<u>D7BFFF</u>
06	<u>FF77FF</u>	0C	<u>FFFF17</u>	12	<u>CBFFFF</u>

实验六 模型机综合实验

- **实验要求：**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。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。
- **实验目的：**了解模型机如何整体工作。利用模型机的指令系统提供的指令，写一段程序，实现两个数 33H 和 55H 的加、减、与、或四种运算，并观察运算结果是否正确。
- **实验电路：**CPTH 软件、LED 或用显示屏。

实验内容 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 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	70	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,@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,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	FFFF17	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 中做为微程序存储器 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。

MOV A, #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 中做为微程序存储器 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 做为程序存储器 EM 的地址，EMRD 就是从程序存储器 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，参照第步的说明，此微指令从程序存储器 EM 中读取指令。

MOV A, 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 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。

实验内容 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 再存到 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 0C7FFE9H, 表示以

PC 做为地址，从 EM 中读出数据送到 DBUS，再将 DBUS 数据存 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 MAR 中。在 T2 状态，微指令为 0D7BFEFH，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。

实验内容 3：移位/取反实验

1. 在 CPTH 软件中的源程序窗口输入下列程序

```
MOV A, #55H
RR A
RLC A
CPL A
END
```

2. 将程序另存为 EX3.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 55	MOV A, #55	立即数 55H 存入累加器 A
02	D0	RR A	不带进位右移累加器 A
03	DC	RLC A	带进位左移累加器 A
04	E4	CPL A	累加器 A 内容取反

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“EX3.ASM 程序跟踪结果”详细介绍)

EX3.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	移位控制	uPC	PC
	T0	00	CBFFFF		指令寄存器 IR	PC 输出	A 输出		写入	1
00 MOV A, #55	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 RR A	T1	D0	FFFCB7	ALU 右移	寄存器 A 标志位 C,Z		A 输出	右移	1	
	T0	D1	CBFFFF		指令寄存器 IR	PC 输出	A 输出		写入	1
03 RLC A	T1	DC	FFED7	ALU 左移	寄存器 A 标志位 C,Z		A 输出	带进位左移	1	
	T0	DD	CBFFFF		指令寄存器 IR	PC 输出	A 输出		写入	1
04 CPL A	T1	E4	FFFE96	ALU 直通	寄存器 A 标志位 C,Z		A 输出		1	
	T0	E5	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。

MOV A, #55: 将累加器的值设为 055H，以便下面观察。

RR A: 本指令为两个状态周期。在 T1 状态，由上次取指操作取出的指令码为 D0H，访问微程序存储器的 20H 单元，读出微指令值为 0FFFCB7H，有效位为 CN、FEN 及 AEN，表示不带进位移位，运算器控制 S2S1S0=111(二进制)表示运算不运算，输出结果就为 A 的值，X2X1X0=101(二进制)表示，运算器“右移”输出到总线，FEN 将标志位保存，AEN 将 DBUS 内容存入 A 中，uPC 加 1 取出下条微令。在 T0 状态，取出下条将要执行的指令。

RLC A: 本指令有两个状态周期。在 T1 状态微指令为 0FFFD7H, CN=1 表示带进位移位, S2S1S0=111 表示 ALU 不做运算, 直接输出 A 内容, X2X1X0=110(二进制)表示, 运算器“左移”输出到 DBUS, AEN 表示 DBUS 内容存入 A 中, FEN 表示保存标志位。T0 状态为取指操作。取出下条将要执行的指令。

CPL A: 本指令为两个状态周期。在 T1 状态, 微指令为 0FFFE96H, S2S1S0=110 表示 ALU 做“取反”运算, X2X1X0=100(二进制)表示, 运算器结果直通到 DBUS, 再存入 A 中, 并保存标志位。T0 状态为取指操作。取出下条将要执行的指令。

实验内容 4: 转移实验

1. 在 CPTH 软件中的源程序窗口输入下列程序

```

LO: MOV  A, #01
LOOP:
      SUB  A, #01
      JC   LOOP
      JZ   LOOP
      JMP  LO
      END

```

2. 将程序另存为 EX4.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 01	MOV A, #01	立即数 01H 存入累加器 A
02	3C 01	SUB A, #01	累加器 A 减 1
04	A0 02	JC 02	若有进位跳到程序 02 地址
06	A4 02	JZ 02	若 A=0 跳转到程序 02 地址
08	AC 00	JMP 00	无条件跳转到程序开始

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。观察在条件满足和不满足的情况下, 条件跳转是否正确执行。(见“EX4.ASM 程序跟踪结果”详细介绍)

EX4.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	PC
	T0	00	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
00 MOV A, #01	T1	7C	C7FFF7	存贮器值 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 SUB A, #01	T2	3C	C7FFEF	存贮器值 EM	寄存器 W	PC 输出	A 输出	1	1
	T1	3D	FFFE91	ALU 直通	寄存器 A 标志位 C,Z		减运算	1	
	T0	3E	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
04 JC 02	T1	A0	C6FFFF	存贮器值 EM	寄存器 PC	PC 输出	A 输出	1	写入
	T0	A1	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
06 JZ 02	T1	A4	C6FFFF	存贮器值 EM	寄存器 PC	PC 输出	A 输出	1	写入
	T0	A5	CBFFFF		指令寄存器 IR	MAR 输出	A 输出	写入	1
02 SUB A, #01	T2	3C	C7FFEF	存贮器值 EM	寄存器 W	PC 输出	A 输出	1	1
	T1	3D	FFFE91	ALU 直通	寄存器 A 标志位 C,Z		减运算	1	
	T0	3E	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
04 JC 02	T1	A0	C6FFFF	存贮器值 EM	寄存器 PC	PC 输出	A 输出	1	写入
	T0	A1	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
02 SUB A, #01	T2	3C	C7FFEF	存贮器值 EM	寄存器 W	PC 输出	A 输出	1	1
	T1	3D	FFFE91	ALU 直通	寄存器 A 标志位 C,Z		减运算	1	
	T0	3E	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
04 JC 02	T1	A0	C6FFFF	存贮器值 EM	寄存器 PC	PC 输出	A 输出	1	写入
	T0	A1	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
02 SUB A, #01	T2	3C	C7FFEF	存贮器值 EM	寄存器 W	PC 输出	A 输出	1	1
	T1	3D	FFFE91	ALU 直通	寄存器 A 标志位 C,Z		减运算	1	
	T0	3E	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
04 JC 02	T1	A0	C6FFFF	存贮器值 EM	寄存器 PC	PC 输出	A 输出	1	写入
	T0	A1	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
06 JZ 02	T1	A4	C6FFFF	存贮器值 EM	寄存器 PC	PC 输出	A 输出	1	写入
	T0	A5	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
08 JMP 00	T1	AC	C6FFFF	存贮器值 EM	寄存器 PC	PC 输出	A 输出	1	写入
	T0	AD	CBFFFF		指令寄存器 IR	PC 输出	A 输出	写入	1
00 MOV A, #01	T1	7C	C7FFF7	存贮器值 EM	寄存器 A	PC 输出	A 输出	1	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。

MOV A, #01: 将累加器的值设为 01H，用于下面计算来产生进位标志和零标志。

SUB A, #01: A 值原为 1，将 A 值第一次减 1 后，应产生“零标志”位。

JC 02: 由上条取指读出的指令码为 0A0H，存入 IR 寄存器后，IR3、IR2 的值为 00(二进制)，表示判进位跳转功能，指令码存入 uPC 后，从 uM 读出的微指令值为 0C6FFFFH，表示以 PC 为地址从 EM 中读出数据 02H 并送到 DBUS，ELP 为低成有效状态，与 IR3、IR2 组成进位跳转控制，此时若有进位，就会产生一个控制信号，将总线 DBUS 上的值 02H 打入 PC，下条微指令取指时，就会从 EM 新的地址 02 中读指令码；此时若无进位，DBUS 上

的值被忽略，PC 加 1，下条取指操作按新 PC 取出指令码执行。当前无进位标志，顺序执行下条指令。

JZ 02: 由上条取指读出的指令码为 0A4H，存入 IR 寄存器后，IR3、IR2 的值为 01(二进制)，表示判零跳转功能，指令码存入 uPC 后，从 uM 读出的微指令值为 0C6FFFH，表示以 PC 为地址从 EM 中读出数据 02H 并送到 DBUS，ELP 为低成有效状态，与 IR3、IR2 组成零跳转控制，与上条指令相比，尽管微指令相同，由于指令码不同，上一个为判进位跳转，这个为判零跳转。此时若零标志位为 1，即 A=0 时，就会产生一个控制信号，将总线 DBUS 上的值 02H 打入 PC，下条微指令取指时，就会从 EM 新的地址 02 中读指令码；此时若零标志位为 0，DBUS 上的值被忽略，PC 加 1，下条取指操作按新 PC 取出指令码执行。由于 A=0，零标志位为 1，产生 PC 打入信号，将 DBUS 上的值 02H 打入 PC。下一条取指操作，PC=02，以 PC 为地址从 EM 的 02 单元取出指令码执行，程序转到 02 地址。

SUB A, #01: A 值现为 0，再减 1 后，A=0FFH，并产生“进位标志”位。

JC 02: 此为判进位跳转指令，此时由于进位标志为 1，与 ELP、IR3、IR2 组成的电路产生 PC 打入信号，将数据总线上的值存入 PC，程序跳转到 02H 地址执行。

SUB A, #01: A 值现为 0FFH，再减 1 后，A=0FEH，无“零标志”，无“进位标志”位。

JC 02: 此为判进位跳转指令，此时无进位标志，程序顺序执行下条指令。

JZ 02: 此为判零跳转指令，此时无零标志位，程序顺序执行下条指令。

JMP 00: 由上条取指操作读出的指令码为 0ACH，存入 IR 寄存器后，IR3、IR2 的值为 11(二进制)，此为无条件跳转控制，指令码存入 uPC 后，从 uM 读出的微指令为 0C6FFFH，表示以 PC 为地址从 EM 中读出数据并送到数据总线 DBUS 上，因为 ELP 有效，与 IR3、IR2 组合产生 PC 的打入信号，将 DBUS 上的数据存入 PC 中，下一条取指微指令按新的 PC 值读出程序的指令码。MOV A, #01: 程序从开头重新执行。

实验内容 5：调用实验

1. 在 CPTH 软件中的源程序窗口,输入下列程序

```
MOV A, #00H
LOOP:
CALL INCA
JMP LOOP
INCA:
ADD A, #01
RET
END
```

2. 将程序另存为 EX5.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 55	MOV A, #00	立即数00H存入累加器A
02	BC 06	CALL 06	调用子程序
04	AC 02	JMP 02	跳转到02地址,循环执行
06	1C 01	ADD A, #01	累加器A加1
08	CC	RET	子程序返回

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。观察在调用子程序和从子程序

返回时，堆栈的工作情况。(见“EX5.ASM 程序跟踪结果”详细介绍)

EX5.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	PC
	T0	00	CBFFFF		指令寄存器IR	PC输出	A输出	写入	1
00 MOV A,#00	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 CALL 06	T3	BC	EF7F7F	PC值	地址寄存器MAR	PC输出	A输出	1	1
	T2	BD	FFEF7F	PC值	堆栈寄存器ST		A输出	1	
	T1	BE	D6BFFF	存储器值EM	寄存器PC	MAR输出	A输出	1	写入
	T0	BF	CBFFFF		指令寄存器IR	PC输出	A输出	写入	1
06 ADD A,#01	T2	1C	C7F7EF	存储器值EM	寄存器W	PC输出	A输出	1	1
	T1	1D	FFFE90	ALU直通	寄存器A 标志位CZ		加运算	1	
	T0	1E	CBFFFF		指令寄存器IR	PC输出	A输出	写入	1
08 RET	T1	CC	FEFF5F	堆栈寄存器ST	寄存器PC		A输出	1	写入
	T0	CD	CBFFFF		指令寄存器IR	PC输出	A输出	写入	1
04 JMP 02	T1	AC	C6FFFF	存储器值EM	寄存器PC	PC输出	A输出	1	写入
	T0	AD	CBFFFF		指令寄存器IR	PC输出	A输出	写入	1
02 CALL 06	T3	BC	EF7F7F	PC值	地址寄存器MAR	PC输出	A输出	1	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。

MOV A, #00: 将累加器的值设为 00H，以便下面观察 A 加 1 后的结果。

CALL 06: 本指令有四个状态周期。在 T3 状态，根据指令码为 0BCH，读出微指令值 0FF7F7FH，有效位为 PCOE、MAREN，X2X1X0 的值为 011(二进制)，PCOE 有效是将 PC 加 1，以便在下步将 PC 压栈时，存入堆栈的是程序下一条指令的地址，MAREN 有效及 X2X1X0 的值表示从 PC 中读出值并送到 MAR 中。在 T2 状态，读出微指令为 0FFE7F7FH，有效位 STEN，X2X1X0=100(二进制)，表示从 PC 中读数据并存入堆栈寄存器 ST 中。在 T1 状态，微指令值为 0D6BFFFH，表示以 MAR 为地址从 EM 中读出数据，此数据就是子程序的地址，此时堆栈中保存的是调用子程序下条指令的地址。将此数据送到 DBUS，再存入 PC 中，实现程序跳转。在 T0 状态，按新的 PC 值，取出下条将要执行的指令。

ADD A, 01: 本指令将累加器加 1。

RET: 本指令有两个状态周期。在 T1 状态，上条取指操作读出的指令码为 0CCH，存入 IR 后，IR3、IR2 的值为 11(二进制)，取出的微指令的值为 0FEFF5FH，有效位为 ELP，X2X1X0=010(二进制)表示从 ST 中输出数据到总线，ELP 有效与 IR3、IR2=11 表示无条件将数据总线 DBUS 的数据打入 PC，实现子程序返回功能。在 T0 状态，按新 PC 取出指令，准备执行。

JMP 02: 程序无条件跳转到 02 地址，执行程序。

实验内容 6：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。

MOV A, #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，所以预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。

MOV A, R0: 由于预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。

MOV A, @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 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。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。

● 实验步骤:

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

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

● 实验结果：

1、根据所写的程序填写下表

程序地址	机器码	反汇编指令	指令说明
00	7C 33	Mov A,#33	将数 33H 送到累加器 A
02	81	MOV R1,A	将累加器 A 的值送到寄存器 R1
03	7C 55	MOV A,#55	将数 55H 送到累加器 A
05	21	ADDC A,R1	累加器 A 的值加上 R1 加进位
06	3C 33	Sub A,#33	累加器 A 的值减数 33H
08	5C 33	AND A,#33	累加器 A 逻辑与立即数 33H
0A	68 33	OR A,#33	累加器 A 逻辑或立即数 33H

2、在模型机上输入所写的程序，并仔细观察程序的执行过程，记录程序跟踪结果填入下表。

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	PC
	T0	7C	C7FFF7		指令存储器 IR	PC	A 输出	写入	1
Mov A,#33	T0	7D	CBFFFF	存储器值 EM	指令存储器 IR	PC	A 输出	写入	1
MOV R1,A	T0	80	FFFB9F	累加器 A	指令存储器 IR	D	A 输出	写入	1
MOV A,#55	T0	7C	C7FFF7	存储器值 EM	指令存储器 IR	PC	A 输出	写入	1
ADDC A,R1	T0	7D	CBFFFF	寄存器 R1	指令存储器 IR	PC	A 输出	写入	1
SUB A,#33	T2	20	FFF7EF	存储器值 EM	寄存器 W	R1	A 输出	1	
	T1	21	FFFE98		寄存器 A, 标志 C,Z	D	A 输出	1	
	T0	22	CBFFFF		指令寄存器 IR	PC	A 输出	写入	1
	T0	3C	C7FDEF		指令寄存器 IR	PC	A 输出	写入	1
AND A,#33	T1	3D	FFFE91	存储器值 EM	寄存器 A, 标志 C,Z	D	A 输出	1	1
	T0	3E	CBFFFF		指令寄存器 IR	PC	A 输出	写入	1
	T0	5C	C7FDEF		指令寄存器 IR	PC	A 输出	写入	1
OR A,#33	T1	5D	FFFE93	存储器值 EM	寄存器 A, 标志 C,Z	D	A 输出	1	1
	T0	5E	CBFFFF		指令寄存器 IR	PC	A 输出	写入	1
	T0	68	C77FFF		指令寄存器 IR	PC	A 输出	写入	1

总结

学习了一个学期的《计算机组成原理》这门课程。在 xx 老师给我们讲《计算机组成原理》这门课程的学期了，我们对于计算机都有了更深的认识 and 了解。计算机技术是世界上发展最快的科学技术之一，产品不断升级换代。当前计算机正朝着巨型化、微型化、智能化、网络化等方向发展，计算机本身的性能越来越优越，应用范围也越来越广泛，从而使计算机成为工作、学习和生活中必不可少的工具。对于计算机我们只是一个小小的探索者，还有更大的知识海洋等待着我们去挖掘，去学习。

计算机组成原理课程设计是在我们上完计算机组成原理课之后的硬件实践课程，是把组成原理课上的理论知识与实践相结合的过程。通过课程设计进一步加深对理论知识的理解和对计算机的工作过程的了解。

在对于今后对计算机接触中，《计算机组成原理》这门课程对我们今后的工作学习也有着不可磨灭的作用。再次感谢同学的帮助，感谢老师的教导！