

实验六 简单微程序控制器设计实验

一、 实验目的

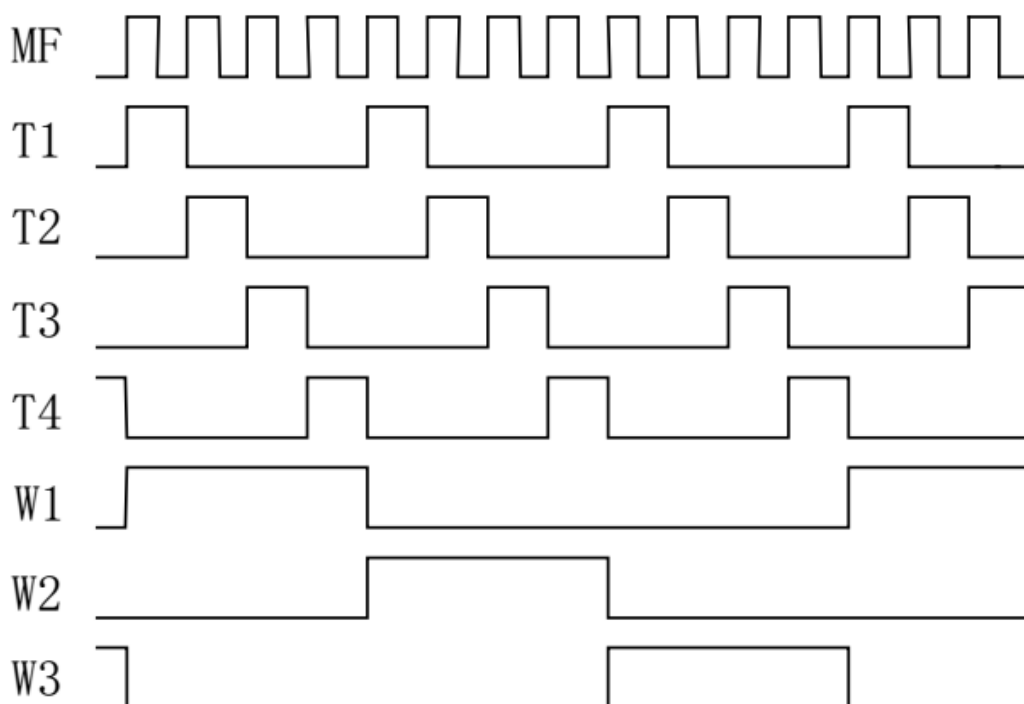
- (1) 掌握微程序控制器的组成原理；
- (2) 将微程序控制器同运算器、存储器、通用寄存器堆、程序计数器、指令寄存器联机，组成一台模型计算机；
- (3) 通过CPU运行九条机器指令组成的简单程序，掌握机器指令与微指令的关系，牢固建立计算机的整机概念。

二、 实验电路与原理

本次实验用到前面实验中的所有电路，包括运算器、存储器、通用寄存器堆、程序计数器、指令寄存器、微程序控制器等，将几个模块组合成为一台简单计算机。在本次实验中，数据通路的控制将由微程序控制器来完成。CPU 从内存取出一条机器指令到执行指令结束的一个机器指令周期，是由微指令组成的序列来完成的，即一条机器指令对应一个微程序。

1. 时序发生器

时序发生器产生计算机模型所需的时序和数字逻辑实验所需的时钟。时序电路由一个 500KHz 晶振、2 片 GAL22V10 组成。根据本机设计，执行一条微指令需要 4 个节拍脉冲 T1、T2、T3、T4，执行一条机器指令需要三个节拍电位 W1、W2、W3，因此本机的基本时序如下：



图中，MF 是晶振产生的 500KHz 基本时钟，T1、T2、T3、T4 是数据通路和控制中各寄存器的节拍脉冲信号，印制板上已将它们和相关的寄存器相连。T1、T2、T3、T4 既供微过程控制器使用，也供硬连线控制器使用。W1、W2、W3，只供硬连线控制器做节拍电位信号使用。

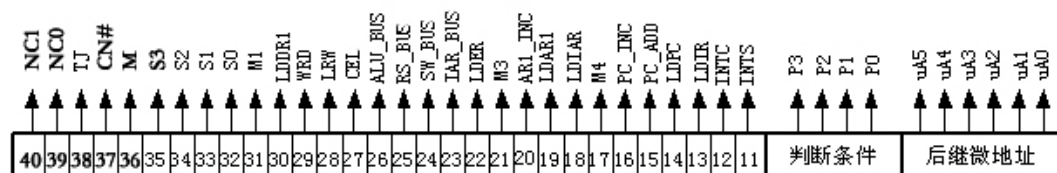
2. 机器指令

本实验仪使用 11 条机器指令，均为单字长（8 位）指令。指令功能及格式如下表所示。指令的高 4 位提供给微程序控制器，低 4 位提供给数据通路。

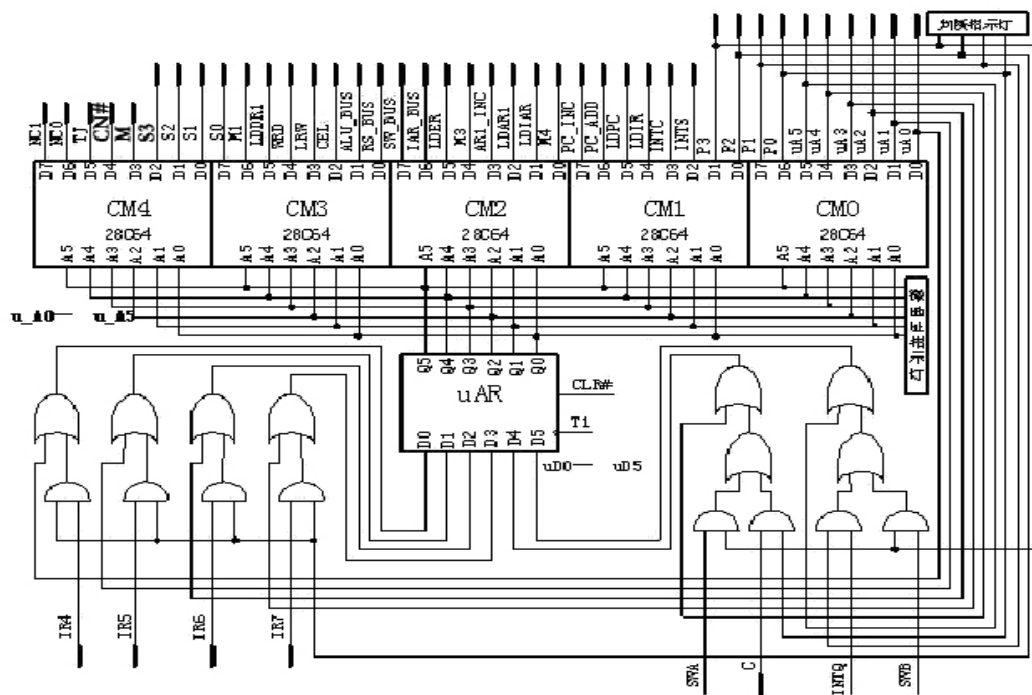
名称	助记符	功能	指令格式							
			R7	R6	R5	R4	R3	R2	R1	R0
加法	ADD Rd, Rs	$Rd+Rs \rightarrow Rd$	0	0	0	0	RS1	RS0	RD1	RD0
减法	SUB Rd, Rs	$Rd-Rs \rightarrow Rd$	0	0	0	1	RS1	RS0	RD1	RD0
逻辑与	AND Rd, Rs	$Rd \& Rs \rightarrow Rd$	0	0	1	1	RS1	RS0	RD1	RD0
存数	STA Rd, [Rs]	$Rd \rightarrow [Rs]$	0	1	0	0	RS1	RS0	RD1	RD0
取数	LDA Rd, [Rs]	$[Rs] \rightarrow Rd$	0	1	0	1	RS1	RS0	RD1	RD0
无条件转移	JMP [Rs]	$[Rs] \rightarrow PC$	1	0	0	0	RS1	RS0	X	X
条件转移	JC D	若 $C=1$ 则 $PC+D \rightarrow PC$	1	0	0	1	D3	D2	D	1 0
停机	STP	暂停运行	0	1	1	0	X	X	X	X
中断返回	IRET	返回断点	1	0	1	0	X	X	X	X
开中断	INTS	允许中断	1	0	1	1	X	X	X	X
关中断	INTC	禁止中断	1	1	0	0	X	X	X	X

3. 微指令格式与微程序控制器电路

采用的微指令格式见下图。微指令字长共 38 位。其中后继微地址 6 位，判别字段 4 位，操作控制字段 28 位，各位进行直接控制。微指令格式中，信号名为高有效信号。

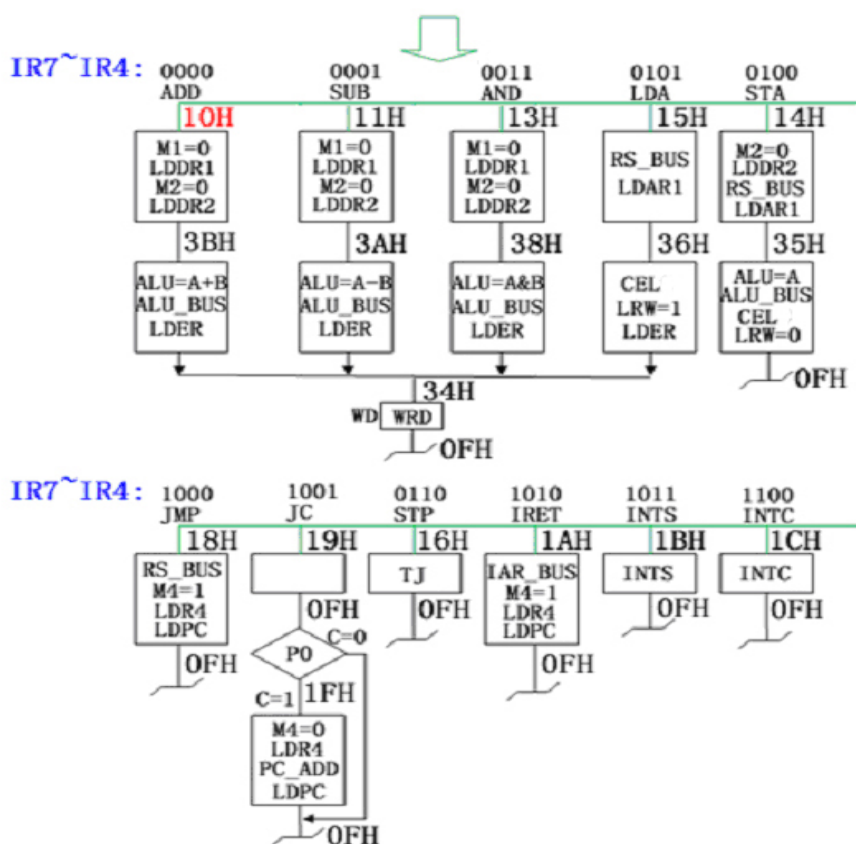


微程序控制器电路如下图所示。控制存储器采用 5 片 EEPROM 28C64。28C64 的输出是 D0—D7。CM0 是最低字节，CM4 是最高字节。微地址寄存器 6 位，用一片 8D 触发器 74LS273 组成，带有清零端。两级与门、或门构成微地址转移逻辑，用于产生下一微指令的地址。在每个 T1 上升沿时刻，新的微指令地址会打入微地址寄存器中，控制存储器随即输出相应的微命令代码。微地址转移逻辑生成下一地址，等下一个 T1 上升沿时打入微地址寄存器。5 片 EEPROM 的地址 A6 直接与控制台开关 SWC 连接，当 SWC = 1 时，微地址大于或者等于 40H，当 SWC = 0 时，微地址的范围为 00H—3FH。SWC 主要用于实现读寄存器堆的功能。



微程序控制器的组成

上述 11 条指令的微程序流程设计如下图所示。每条微指令可按前述的微指令格式转换成二进制代码，然后写入 5 个 28C64 中。



三、 实验设备

TEC-9 计算机组成原理实验系统一台。

四、 实验任务

1. 对机器指令系统组成的简单程序进行译码。

地址	指令	机器代码
00H	LDA R0, [R2]	58H
01H	LDA R1, [R3]	5DH
02H	ADD R0, R1	04H
03H	JC +4	94H
04H	AND R2, R3	3EH
05H	SUB R3, R2	1BH
06H	STA R3, [R2]	4BH
07H	STP	60H
08H	JMP [R1]	84H

2. 根据微程序流程图编写微程序控制器信号表。

微指	ADD		SUB		AND		LDA		STA		JMP	JC		STP	IRET	INTS	INTC		
微址	10	3B	11	3A	13	38	15	36	14	35	18	19	1F	16	1A	1B	1C	34	0F
下址	3B	34	3A	34	38	34	36	34	35	0F	0F	1F/0F	0F	0F	0F	0F	0F	0F	5
P0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
P2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
INTS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
INTC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
LDIR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LDPC	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0
PCADD	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
PCINC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
LDIAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LDAR1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1
AR1INC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LDER	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
IARBUS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
SWBUS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RSBUS	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
ALUBUS	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0
CEL	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
LRW	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
WRD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
LDDR1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
M1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
S1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
S2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S3	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
CN#	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
TJ	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

五、 实验步骤

1. 接线

微程序控制器与数据通路之间的线通过选择开关直接选择，设置控制器选择模式开关拨=“微程序”。

连接数据通路部分的线：

- a. 数据通路的 LDIR 接 CER、LDPC 接 LDR4、LDDR1 接 LDDR2、M1 接 M2、LDAR1 接 LDAR2。
- b. 指令寄存器 IR 的输出 IR0 接寄存器堆的 RD0、WR0，IR1 接 RD1、WR1，IR2 接 RS0，IR3 接 RS1。

信号 1	LDIR	LDPC	LDDR1	M1	LDAR1	IR0	IR0	IR1	IR1	IR2	IR3
信号 2	CER	LDR4	LDDR2	M2	LDAR2	RD0	WR0	RD1	WR1	RS0	RS1

2. 使用 PC 端软件设置通用寄存器 R2=60H, R3=61H。
3. 使用 PC 端软件从内存 00 地址开始依次存 10 个机器代码：58H, 5DH, 04H, 94H, 3EH, 1BH, 4BH, 60H, 84H。在 60H 存入 24H，用于给 R0 置初值；在 61H 存入 83H，用于给 R1 置初值。
4. 用单拍（DP）方式执行一遍程序。

在单拍执行过程中，首先要随时监测 AR2 的值和 IR 的值，以判定程序执行到何处，正在执行哪条指令。监测微地址指示灯和判断字段指示灯，对照微程序流程图，可以判断出微指令的地址和正在进行的微操作。

程序执行的结果如下：

初值：R0 未定，R1 未定，R2 = 60H，R3 = 61H。存储器 60H 单元的内容是 24H，61H 单元的内容是 83H。

1) LDA R0, [R2]

执行结果 R0 = 24H、R1=XXH、R2=60H、R3=61H

2) LDA R1, [R3]

执行结果 R0 = 24H、R1=83H、R2=60H、R3=61H

3) ADD R0, R1

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=61H

C = 0。

4) JC +4

因为 C = 0, 顺序执行。

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=61H

5) AND R2, R3

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=61H

6) SUB R3, R2

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=01H

7) STA R3, [R2]

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=01H

存储器 60 单元的内容为 01H。

8) STP

执行结果：无变化

9) JMP [R1]

执行结果 转移到 83H。（察看 AR2 即 PC 地址）

5. 用单指（**DZ=1**）方式执行程序。

程序执行的结果如下：

初值：R0 未定，R1 未定，R2 = 60H，R3 = 61H。存储器 60H 单元的内容是 24H，61H 单元的内容是 83H。

1) LDA R0, [R2]

执行结果 R0 = 24H、R1=XXH、R2=60H、R3=61H

2) LDA R1, [R3]

执行结果 R0 = 24H、R1=83H、R2=60H、R3=61H

3) ADD R0, R1

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=61H
C = 0。

4) JC +4

因为 C = 0, 顺序执行。

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=61H

5) AND R2, R3

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=61H

6) SUB R3, R2

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=01H

7) STA R3, [R2]

执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=01H
存储器 60 单元的内容为 01H。

8) STP

执行结果：无变化

9) JMP [R1]

执行结果 转移到 83H。（察看 AR2 即 PC 地址）

6. 用连续方式执行程序。

执行结果为：

PC 地址停在 07H。

R0 = 0A7H、 R1=83H、 R2=60H、 R3=01H

存储器 60 单元的内容为 01H。