

北 京 邮 电 大 学

实 验 报 告

课程名称____计算机组成原理实验____

实验名称_____实验 4-6_____

_计算机_学院_305_班 姓名__张晨阳__

教师__李晶__ 成绩_____

_2024_年_5_月_29_日

实验四、微程序控制器实验

一、 实验任务及目的

实验目的：

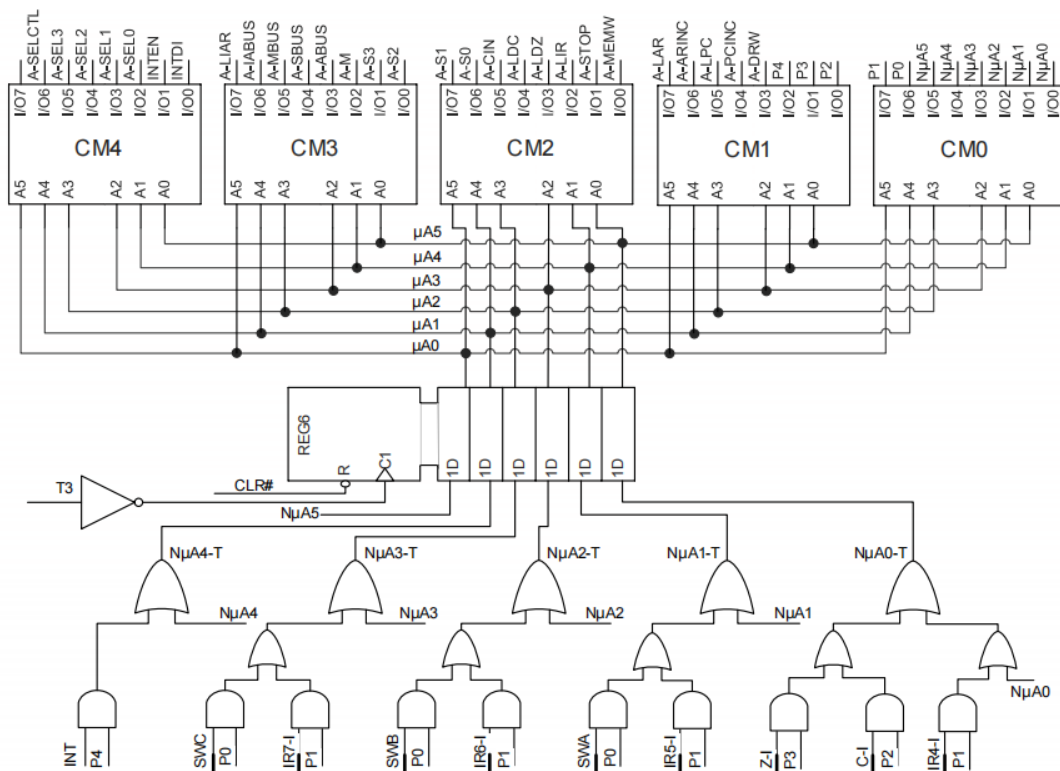
1. 掌握微程序控制器的原理；
2. 掌握 TEC-8 模型计算机中微程序控制器的实现方法，尤其是微地址转移逻辑的实现方法；
3. 理解条件转移对计算机的重要性；

实验任务：

1. 熟悉微程序流程图和微程序指令系统
 - 1.1 跟踪控制台操作写寄存器、写存储器、读存储器、读寄存器、的执行过程；
 - 1.2 跟踪指令的执行过程：执行 ADD、LD、ST 指令

二、 实验电路分析

实验四微程序控制电路如下图：



图中上边部分的 CM4、CM3、CM2、CM1、CM0 组成控制存储器，对应着 40 位的微指令，其中控制字段 29 位，顺序字段 11 位。其中，CM0 用于存储微指令最低的 8 位代码，CM4 用于存储微指令最高的 8 位代码。

中间部分的 REG6 组成微地址寄存器。在一条微指令结束时，用 T3 的下降沿将微地址转移逻辑产生的下一条微指令地址 $N\mu A5$ ， $N\mu A4-T \sim N\mu A0-T$ 写入微地址寄存器。

下面的若干与、或门电路组成微地址转移逻辑，负责决定下一条微指令的地址。其输入信号包括当前微指令的下址和控制信号等；T3 的下降沿触发微地址寄存器的更新。新的微地址用于选择控制存储器中的下一条微指令，从而实现微指令的顺序执行。举例说明，假设现在 P1 为 1，其余判别位为 0，则下一条微指令的地址为：

$$N\mu A5-T = N\mu A5$$

$$N\mu A4-T = N\mu A4$$

$$N\mu A3-T = N\mu A3 \text{ 或 } (P1 \text{ 与 } IR7)$$

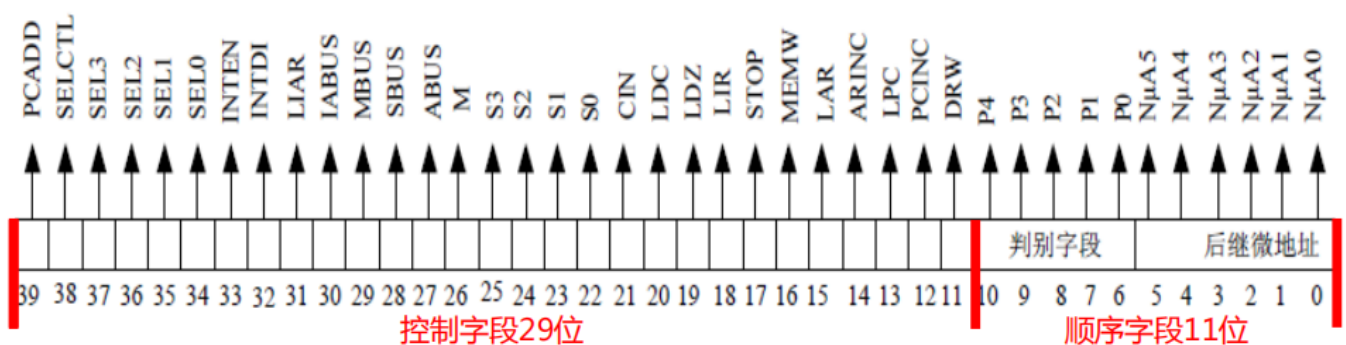
$$N\mu A2-T = N\mu A2 \text{ 或 } (P1 \text{ 与 } IR6)$$

$$N\mu A1-T = N\mu A1 \text{ 或 } (P1 \text{ 与 } IR5)$$

$$N\mu A0-T = N\mu A3 \text{ 或 } (P1 \text{ 与 } IR4)$$

这新的微地址将会在 T3 的下降沿更新至微地址寄存器。

微指令格式如下图：

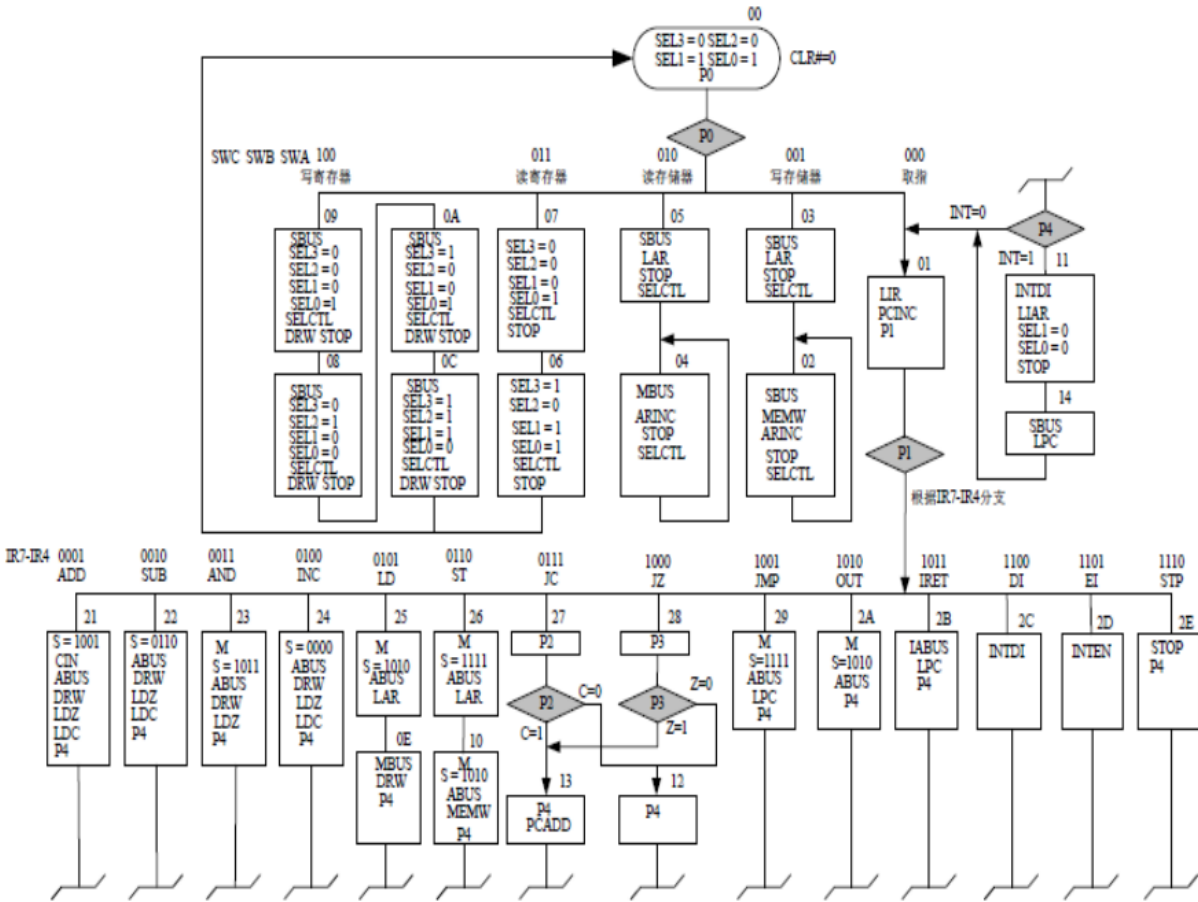


实验中涉及的后继地址、判别字段、微命令的含义如下表:

字段	解释
N μ A5~N μ A0	下址，在微指令顺序执行的情况下，它是下一条微指令的地址
P0	=1 时，根据后继微地址 N μ A5~N μ A0 和模式开关 SWC、SWB、SWA 确定下一条微指令的地址。
P1	=1 时，根据后继微地址 N μ A5~N μ A0 和指令操作码 IR7~IR4 确定下一条微指令的地址。
P2	=1 时，根据后继微地址 N μ A5~N μ A0 和进位 C 确定下一条微指令的地址
P3	=1 时，根据后继微地址 N μ A5~N μ A0 和结果为 0 标志 Z 确定下一条微指令的地址
P4	=1 时，根据后继微地址 N μ A5~N μ A0 和中断信号 INT 确定下一条微指令的地址。模型计算机中，中断信号 INT 由时序发生器在接到中断请求信号后产生。
STOP	=1 时，在 T3 结束后时序发生器停止输出节拍脉冲 T1、T2、T3。
IABUS	=1 时，将中断地址寄存器中的地址送数据总线 DBUS。
LIAR	=1 时，在 T3 的上升沿，将 PC7~PC0 写入中断地址寄存器 IAR
INTDI	=1 时，置允许中断标志(在时序发生器中)为 0，禁止 TEC-8 模型计算机响应中断请求。
INTEN	=1 时，置允许中断标志(在时序发生器中)为 1，允许 TEC-8 模型计算机响应中断请求。
PCADD	=1 时，将当前的 PC 值加上相对转移量，生成新的 PC。

三、微程序流程图分析

实验的微程序流程如下图:



如果操作模式为 000，即取指模式，还会涉及指令的编码。该编码由 IR7~IR4 给出。执行完相应的指令后，对 P4 进行条件判断，继续执行相应的指令。

[illegible]

读寄存器（从R0-R3中分别读出它们存储的数据）										
序号	操作	操作的数据	操作目的	操作结果						
				控制信号（有效的信号）	μA	NμA	D7-D0	A7-A0	B7-B0	
1	CLR	无	复位	SEL3-SELO=0011 PQ=1	00H	01H				
2	SW=011 QD	无	选择操作模式为读寄存器 读取寄存器R0、R1的值	RS1 ~ RS0=01， RD1~RD0=00， STOP=1	07H	06H	00H	00H	03H	
3	QD	无	读取寄存器R2、R3的值	RS1 ~ RS0=11， RD1~RD0=10， STOP=1	06H	00H	00H	04H	44H	
读写存储器（从存储器的000H,010H,020H,03地址分别读出其中存储的指令或数据）										
序号	操作	操作的数据	操作目的	操作结果						
				控制信号（有效的信号）	μA	NμA	D7-D0	A7-A0	B7-B0	AR7-AR0
1	CLR	无	复位	SEL3-SELO=0011 PQ=1	00H	01H				
2	SW=010 QD	00H	选择操作模式为读存储器	SBSU=1， LAR=1， STOP=1	05H	04H	03H	00H	00H	00H
3	QD	无	读取存储器地址00H中的数据	MBSU=1， ARINC=1， STOP=1	04H	04H	00H	00H	00H	00H
4	QD	无	读取存储器地址01H中的数据	MBSU=1， ARINC=1， STOP=1	04H	04H	01H	00H	00H	01H
5	QD	无	读取存储器地址02H中的数据	MBSU=1， ARINC=1， STOP=1	04H	04H	02H	00H	00H	02H
6	QD	无	读取存储器地址03H中的数据	MBSU=1， ARINC=1， STOP=1	04H	04H	03H	00H	00H	03H

实验五、CPU 组成与机器指令的执行

一、 实验任务及目的

实验目的：

- 1. 用微程序控制器控制数据通路，将相应的信号线连接，构成一台能够运行测试程序的 CPU；
- 2. 执行一个简单的程序，掌握机器指令与微指令的关系；
- 3. 理解计算机如何取出指令、如何执行指令、如何在一条指令执行结束之后自动取出下一条指令并执行，从而牢固建立计算机整机概念。

实验任务：

- 1. 预习任务：完成对给定程序的手工汇编。
- 2. 通过简单的连线构成能够运行程序的 TEC-8 模型计算机。
- 3. 将程序写入存储器，给寄存器 R2、R3 赋初值。
- 4. 跟踪执行程序，用单拍方式运行一遍，用连续方式运行一遍，详细记录实验过程及结果。
- 5. 用实验台操作检查程序运行结果。

二、 程序的手工汇编结果

地址	指令	二进制机器代码	地址	指令	二进制机器代码
00H	LD R0, [R3]	01010011	0AH	INC R2	01001000
01H	INC R3	01001100	0BH	ST R2, [R2]	01101010
02H	LD R1, [R3]	01010111	0CH	AND R0, R1	00110001
03H	SUB R0, R1	00100001	0DH	OUT R2	10100010
04H	JZ 0BH	10000110	0EH	STP	11100000
05H	ST R0, [R2]	01101000	0FH	85H	10000101
06H	INC R3	01001100	10H	23H	00100011
07H	LD R0, [R3]	01010011	11H	EFH	11101111
08H	ADD R0, R1	00010001	12H	00H	00000000
09H	JC 0CH	01110010	13H	00H	

三、 实验过程及结果

单拍方式

指令	μA	N μA	P	INS	PC	AR	IR	A	B	D
	000001	100000	00010	01010011	00H	00H	00000000			
LD R0,[R3]	100101	001110	00000	01010011	01H	00H	01010011			
	001110	000001	10000	01001100	01H	0FH	01010011		0FH	85H
	000001	100000	00010	01001100	01H	0FH	01010011	85H	0FH	
INC R3	100100	000001	10000	01010111	02H	0FH	01001100	0FH	85H	10H
	000001	100000	00000	01010111	02H	0FH	01001100	10H	85H	
LD R1,[R3]	100101	001110	00000	00100001	03H	0FH	01010111		10H	10H
	001110	000001	10000	00100001	03H	10H	01010111		10H	23H
	000001	100000	00000	00100001	03H	10H	01010111	23H	10H	
SUB R0,R1	100010	000001	10000	10000110	04H	10H	00100001	85H	23H	62H
	000001	100000	00000	10000110	04H	10H	00100001	62H	23H	
JZ 0BH	101000	010010	00000	01101000	05H	10H	10000110	23H	12H	
	010010	000001	10000	01101000	05H	10H	10000110	23H	12H	
	000001	100000	00000	01101000	05H	10H	10000110	23H	12H	
ST R0,[R2]	100110	010000	00000	01001100	06H	10H	01101000	12H	62H	12H
	010000	000001	10000	01001100	06H	12H	01101000	12H	62H	62H
	000001	100000	00000	01001100	06H	12H	01101000	12H	62H	
INC R3	100100	000001	10000	01010011	07H	12H	01001100	10H	62H	11H
	000001	100000	00000	01010011	07H	12H	01001100	11H	62H	
LD R0,[R3]	100101	001110	00000	00010001	08H	12H	01010011	62H	11H	11H
	001110	000001	10000	00010001	08H	11H	01010011	62H	11H	EFH
	000001	100000	00000	00010001	08H	11H	01010011	EFH	11H	
ADD R0,R1	100001	000001	10000	01110010	09H	11H	00010001	EFH	23H	12H
	000001	100000	00000	01110010	09H	11H	00010001	12H	23H	
JC 0CH	100111	010010	00000	01001000	0AH	11H	01110010	12H	12H	
	010011	000001	10000	01001000	0AH	11H	01110010	12H	12H	

	000001	100000	00000	00110001	0CH	11H	01110010	12H	12H	
AND R0,R1	100001	000001	10000	10100010	0DH	11H	00110001	12H	23H	02H
	000001	100000	00000	10100010	0DH	11H	00110001	02H	23H	
OUT R2	101010	000001	10000	11100000	0EH	11H	10100010	02H	12H	12H
	000001	100000	00000	11100000	0EH	11H	10100010	02H	12H	
STP	101110	000001	10000	10000101	0FH	11H	11100000	02H	02H	
	000001	100000	00000	10000101	0FH	11H	11100000	02H	02H	

连续方式

寄存器/地址	R0	R1	R2	R3	12H
执行操作前的值	00H	00H	12H	0FH	00H
寄存器/地址	R0	R1	R2	R3	12H
执行操作后的值	02H	23H	12H	11H	62H

四、 实验收获及体会

通过实验五，我更加理解了 CPU 的组成结构以及对于存储器和寄存器的读写操作也更加熟悉。也掌握了计算机如何取指令、执行指令、以及执行完当前指令取下一条指令，对计算机组成原理理论的学习也很有帮助。

具体实验时，因为还未完全明白原理，以至于担心数据记不全，在单拍方式时，执行到 STP 后，依然 QD 了很多次，一直到了 15H，记录了很多没用的数据。后来也是经过同学的提醒，才意识到已经做完单拍方式的实验。这也让我意识到，做实验之前，应该先学习理论，掌握原理，实践是用来巩固学习的，而不是通过实验学习陌生的知识。

实验六、中断原理实验

一、 实验任务及目的

实验目的：

1. 从硬件、软件结合的角度，模拟中断的过程；
2. 通过简单的中断系统掌握中断的相关概念；
3. 了解微程序控制器与中断控制器协调的基本原理；
4. 掌握中断子程序和一般子程序的本质区别, 掌握中断的突发性和随机性。

实验任务：

1. 理解中断相关指令，以及每个信号的意义和变化条件
2. 将主程序和中断服务程序手工汇编成二进制机器代码
3. 通过简单的连线构成能够运行程序的 TEC-8 模型计算机。
4. 将主程序和中断服务程序装入存储器，给寄存器 R1 赋初值 01H，R0 赋初值 0。
5. 执行三遍主程序和中断服务程序，详细记录中断有关信号变化情况，特别记录好断点和 R0 的值。
6. 将主程序中地址为 00H 的 EI 指令改为 DI，重新运行程序，记录现象。

二、 程序的手工汇编结果（包括主程序和中断服务程序）

主程序机器代码			中断服务程序机器代码		
地址	指令	二进制机器代码	地址	指令	二进制机器代码
00H	EI\DI	11010000	45H	ADD R0, R0	00010000
01H	INC R0	01000000	46H	EI	11010000
02H	INC R0	01000000	47H	IRET	10110000
03H	INC R0	01000000			
04H	INC R0	01000000			
05H	INC R0	01000000			
06H	INC R0	01000000			
07H	INC R0	01000000			
08H	INC R0	01000000			
09H	JMP [R1]	10010100			

三、 实验过程及结果

执行程序顺序	PC 断点值	中断时的 R0
第 1 遍	05H	15H
第 2 遍	06H	84H
第 3 遍	09H	C0H

第一遍执行到断点后，再一步步执行中断服务程序，PC 的变化如下：

45H → 46H → 47H → 05H → 06H ...

第二遍执行到断点后，再一步步执行中断服务程序，PC 的变化如下：

45H → 46H → 47H → 06H → 07H ...

第三遍执行到断点后，再一步步执行中断服务程序，PC 的变化如下：

45H → 46H → 47H → 09H → 10H ...

四、 实验收获及体会

通过实验六，我熟悉了中断控制器和微程序控制器在中断过程中的作用，掌握了中断机制，加深了对中断的理解。同时，还巩固了实验四和实验五的硬件操作。

在实验过程中，我发现编写结束后正常执行一遍很容易回到断点，继续执行原程序，但继续设置断点，执行时会在 47H 和 48H 两个地址间跳跃而不回到断点处。除此之外，在关中断的情况下，不会跳转到中断程序，而是继续执行原程序。