

# 哈爾濱工業大學

## 課程報告

課程名稱： 計算機組成原理

報告題目： 複雜模型機設計實驗

所在院系： 計算學部

所在專業：

學生姓名：

學生學號：

選課時間： 2023 年春季學期

評閱成績：

# 复杂模型机设计实验

## 一、处理器功能及指令系统定义

### 1.1 数据格式

模型机规定采用无符号数表示数据，字长为 8 位，8 位全用来表示数据（最高位不表示符号），数值表示范围是： $0 \leq X \leq 2^8 - 1$ 。

### 1.2 指令设计

模型机设计三大类指令共十五条，其中包括运算类指令、控制转移类指令，数据传送类指令。

运算类指令包含三种运算，算术运算、逻辑运算和移位运算，设计有 6 条运算类指令，分别为：ADD、AND、INC、SUB、OR、RR，所有运算类指令都为单字节，寻址方式采用寄存器直接寻址。

控制转移类指令有三条 HLT、JMP、BZC，用以控制程序的分支和转移，其中 HLT 为单字节指令，JMP 和 BZC 为双字节指令。

数据传送类指令有 IN、OUT、MOV、LDI、LAD、STA 共 6 条，用以完成寄存器和寄存器、寄存器和 I/O、寄存器和存储器之间的数据交换，除 MOV 指令为单字节指令外，其余均为双字节指令。

### 1.3 指令格式

所有单字节指令（ADD、AND、INC、SUB、OR、RR、HLT 和 MOV）格式如下：

7 6 5 4	3 2	1 0
OP-CODE	RS	RD

其中，OP-CODE 为操作码，RS 为源寄存器，RD 为目的寄存器，并规定：

RS 或 RD	选定的寄存器
00	R0
01	R1
10	R2
11	R3

IN 和 OUT 的指令格式为：

7 6 5 4 (1)	3 2 (1)	1 0 (1)	7—0 (2)
OP-CODE	RS	RD	P

其中括号中的 1 表示指令的第一字节，2 表示指令的第二字节，OP-CODE 为操作码，RS 为源寄存器，RD 为目的寄存器，P 为 I/O 端口号，占用一个字节，系统的 I/O 地址译码原理见图 1(在地址总线单元)。

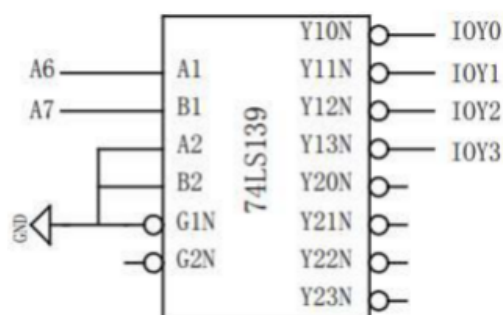


图 1 I/O 地址译码原理图

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 1 所示：

表 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

系统设计五种数据寻址方式，即立即、直接、间接、变址和相对寻址，LDI 指令为立即寻址，LAD、STA、JMP 和 BZC 指令均具备直接、间接、变址和相对寻址能力。LDI 的指令格式如下，第一字节同前一样，第二字节为立即数。

7 6 5 4 (1)	3 2 (1)	1 0 (1)	7—0 (2)
OP-CODE	RS	RD	data

LAD、STA、JMP 和 BZC 指令格式如下。

7 6 5 4 (1)	3 2 (1)	1 0 (1)	7—0 (2)
OP-CODE	M	RD	D

其中 M 为寻址模式，具体见表 2，以 R2 做为变址寄存器 RI。

表 2 寻址方法

寻址模式 M	有效地址 E	说 明
00	$E = D$	直接寻址
01	$E = (D)$	间接寻址
10	$E = (RI) + D$	RI 变址寻址
11	$E = (PC) + D$	相对寻址

## 1.4 指令系统

本模型机共有 15 条基本指令，表 3 列出了各条指令的格式、汇编符号、指令功能。

表 3 指令描述

助记符号	指令格式			指令功能
MOV RD, RS	0100	RS	RD	$RS \rightarrow RD$
ADD RD, RS	0000	RS	RD	$RD + RS \rightarrow RD$
SUB RD, RS	1000	RS	RD	$RD - RS \rightarrow RD$
AND RD, RS	0001	RS	RD	$RD \wedge RS \rightarrow RD$
OR RD, RS	1001	RS	RD	$RD \vee RS \rightarrow RD$
RR RD, RS	1010	RS	RD	RS右环移 $\rightarrow RD$
INC RD	0111	**	RD	$RD+1 \rightarrow RD$
LAD M D, RD	1100	M	RD D	$E \rightarrow RD$
STA M D, RS	1101	M	RD D	$RD \rightarrow E$
JMP M D	1110	M	** D	$E \rightarrow PC$
BZC M D	1111	M	** D	当FC或FZ=1时, $E \rightarrow PC$
IN RD, P	0010	**	RD P	$[P] \rightarrow RD$
OUT P, RS	0011	RS	** P	$RS \rightarrow [P]$
LDI RD, D	0110	**	RD D	$D \rightarrow RD$
HALT	0101	**	**	停机

## 二、微程序设计

### 2.1 微程序流程图

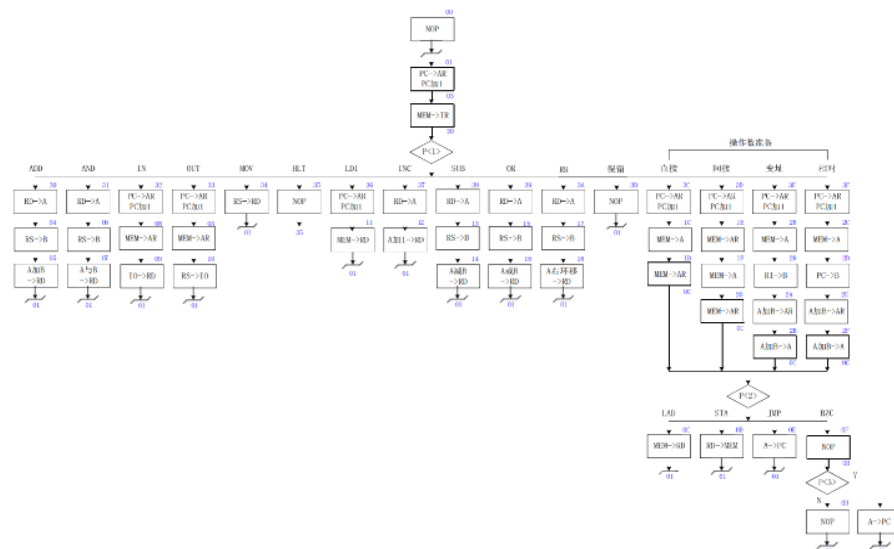


图 2 微程序流程图

## 2.2 二进制代码表

表 5 二进制代码表

地址	十六进制表示	高五位	S3-S0	A 字段	B 字段	C 字段	UA5-UA0
00	00 00 01	00000	0000	000	000	000	000001
01	00 6D 43	00000	0000	110	110	101	000011
03	10 70 70	00010	0000	111	000	001	110000
04	00 24 05	00000	0000	010	011	000	000101
05	04 B2 01	00000	1001	011	001	000	000001
06	00 24 07	00000	0000	010	011	000	000111
07	01 32 01	00000	0010	011	001	000	000001
08	10 60 09	00010	0000	110	000	000	001001
09	18 30 01	00011	0000	011	000	000	000001
0A	10 60 10	00010	0000	110	000	000	010000
0B	00 00 01	00000	0000	000	000	000	000001
0C	10 30 01	00010	0000	011	000	000	000001
0D	20 06 01	00100	0000	000	001	100	000001
0E	00 53 41	00000	0000	101	001	101	000001
0F	00 00 CB	00000	0000	000	000	011	001011
10	28 04 01	00101	0000	000	010	000	000001
11	10 30 01	00010	0000	011	000	000	000001
12	06 B2 01	00000	1101	011	001	000	000001
13	00 24 14	00000	0000	010	011	000	010100
14	05 B2 01	00000	1011	011	001	000	000001
15	00 24 16	00000	0000	010	011	000	010110
16	01 B2 01	00000	0011	011	001	000	000001
17	00 24 18	00000	0000	010	011	000	011000
18	02 B2 01	00000	0101	011	001	000	000001
1B	00 53 41	00000	0000	101	001	101	000001
1C	10 10 1D	00010	0000	001	000	000	011101
1D	10 60 8C	00010	0000	110	000	010	001100
1E	10 60 1F	00010	0000	110	000	000	011111
1F	10 10 20	00010	0000	001	000	000	100000
20	10 60 8C	00010	0000	110	000	010	001100
28	10 10 29	00010	0000	001	000	000	101001
29	00 28 2A	00000	0000	010	100	000	101010
2A	04 E2 2B	00000	1001	110	001	000	101011
2B	04 92 8C	00000	1001	001	001	010	001100
2C	10 10 2D	00010	0000	001	000	000	101101
2D	00 2C 2E	00000	0000	010	110	000	101110
2E	04 E2 2F	00000	1001	110	001	000	101111
2F	04 92 8C	00000	1001	001	001	010	001100
30	00 16 04	00000	0000	001	011	000	000100
31	00 16 06	00000	0000	001	011	000	000110
32	00 6D 48	00000	0000	110	110	101	001000
33	00 6D 4A	00000	0000	110	110	101	001010
34	00 34 01	00000	0000	011	010	000	000001
35	00 00 35	00000	0000	000	000	000	110101
36	00 6D 51	00000	0000	110	110	101	010001
37	00 16 12	00000	0000	001	011	000	010010
38	00 16 13	00000	0000	001	011	000	010011
39	00 16 15	00000	0000	001	011	000	010101
3A	00 16 17	00000	0000	001	011	000	010111
3B	00 00 01	00000	0000	000	000	000	000001
3C	00 6D 5C	00000	0000	110	110	101	011100
3D	00 6D 5E	00000	0000	110	110	101	011110
3E	00 6D 68	00000	0000	110	110	101	101000
3F	00 6D 6C	00000	0000	110	110	101	101100

三、 机器指令程序执行过程——必选部分

3.1 设计思路

根据现有指令，在模型机上实现以下运算：从 IN 单元读入一个数据，根据读入数据的低 4 位值 X，求 1+2+...+X 的累加和，01H 到 0FH 共 15 个数据存于 60H 到 6EH 单元。

3.2 指令分析

1) START: IN R0,00H

地址	内容	助记符	说明
00000000	00100000	; START: IN R0,00H	从 IN 单元读入计数初值
00000001	00000000		

1. 微指令执行过程
- 1) PC->AR;(PC)+1->PC
- 2) MEM->AR
- 3) IO->R0
2. 关键寄存器

表 3-1 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	0	AR	00H->01H->00H
R1	00H	PC	00H->01H->02H
R2	00H	IR	00H->20H
R3	00H	MEM	00H->20H

3. 时序图



图 3-1 时序图

2) LDI R1,0FH

地址	内容	助记符	说明
00000010	01100001	; LDI R1,0FH	立即数 0FH 送 R1
00000011	00001111		

1. 微指令执行过程
- 1) PC->AR;(PC)+1->PC
- 2) MEM->R1
2. 关键寄存器

表 3-2 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	00H->02H->03H
R1	00H->0FH	PC	02H->03H->04H
R2	00H	IR	20H->61H
R3	00H	MEM	20H->61H->0FH

3. 时序图

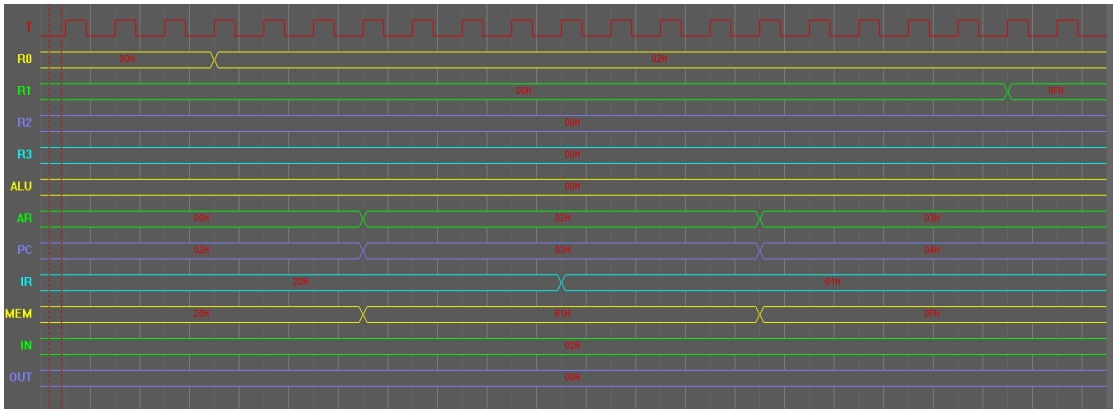


图 3-2 时序图

3) AND R0,R1

地址	内容	助记符	说明
00000100	00010100	; AND R0,R1	得到 R0 低四位

1. 微指令执行过程
- 1) R0->A
- 2) R1->B
- 3) A AND B->R0

2. 关键寄存器

表 3-3 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	03H->04H
R1	0FH	PC	04H->05H
R2	00H	IR	61H->14H
R3	00H	MEM	0FH->14H

3. 时序图

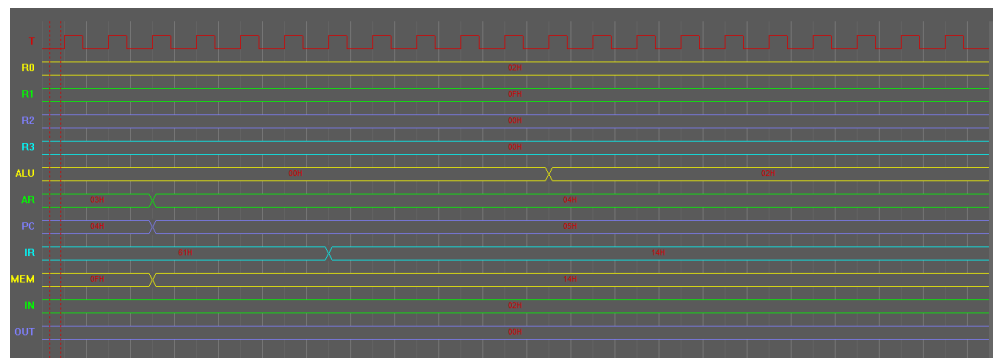


图 3-3 时序图

4) LDI R1,00H

地址	内容	助记符	说明
00000101	01100001	; LDI R1,00H	装入和初值 00H
00000110	00000000		

1. 微指令执行过程

1)PC->AR;(PC)+1->PC

2) MEM->R1

2. 关键寄存器

表 3-4 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	04H->05H->06H
R1	0FH->00H	PC	05H->06H->07H
R2	00H	IR	14H->61H
R3	00H	MEM	14H->61H->00H



3. 时序图

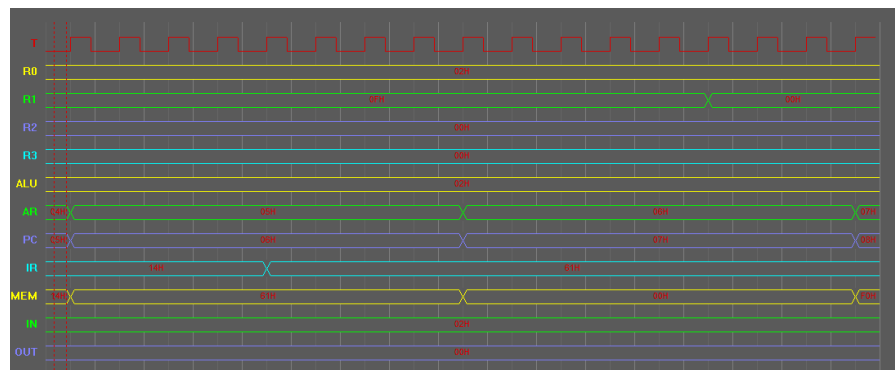


图 3-4 时序图

5) BZC RESULT

地址	内容	助记符	说明
00000111	11110000	; BZC RESULT	计数值为 0 则跳转
00001000	00010110		

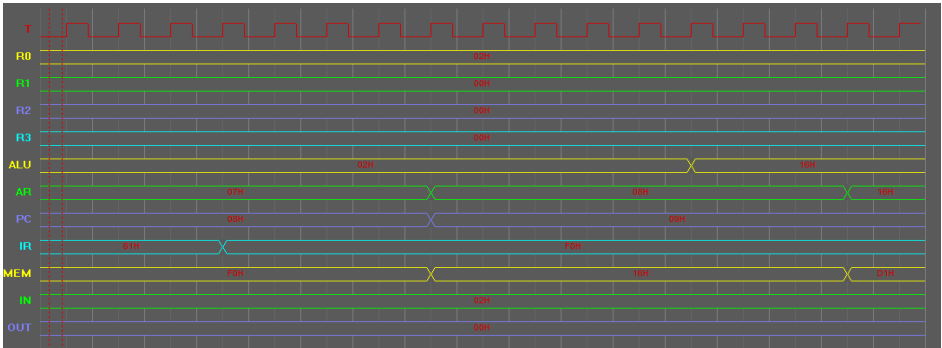
1. 微指令执行过程

- 1) PC->AR;(PC)+1->PC
- 2) MEM->A
- 3) MEM->AR
- 4) NOP,P<3>
- 5) A->PC

2. 关键寄存器

表 3-5 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	06H->07H->08H->16H
R1	00H	PC	07H->08H->09H
R2	00H	IR	61H->F0H
R3	00H	MEM	F0H->16H->D1H



3. 时序图

图 3-5 时序图

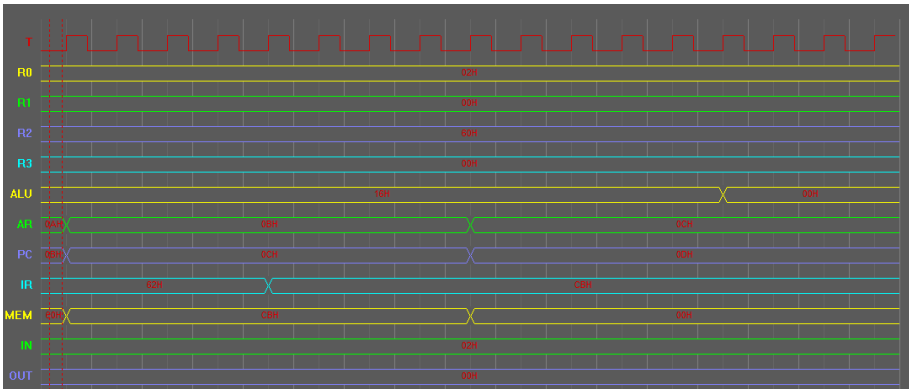
6) LDI R2,60H

地址	内容	助记符	说明
00001001	01100010	; LDI R2,60H	读入数据始地址
00001010	01100000		

1. 微指令执行过程
- 1)PC->AR;(PC)+1->PC
- 2) MEM->R2
2. 关键寄存器

表 3-6 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	16H->09H->0AH
R1	00H	PC	09H->0AH->0BH
R2	00H->60H	IR	F0H->62H
R3	00H	MEM	D1H->62H->60H



3. 时序图

图 3-6 时序图

7) LOOP: LAD R3,[RI],00H(循环开始)

地址	内容	助记符	说明
00001011	11001011	; LOOP: LAD R3,[RI],00H	从 MEM 读入数据送 R3，变址寻址，偏移量为 00H
00001100	00000000		

1. 微指令执行过程
- 1) PC->AR;(PC)+1->PC
- 2) MEM->A
- 3) RI->B
- 4) A+B->AR

- 5) A+B->A
  - 6) MEM->R3
2. 关键寄存器

表 3-7 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	0AH->0BH->0CH->60H
R1	00H	PC	0BH->0CH->0DH
R2	60H	IR	62H->CBH
R3	00H->01H	MEM	60H->CBH->00H->01H

3. 时序图

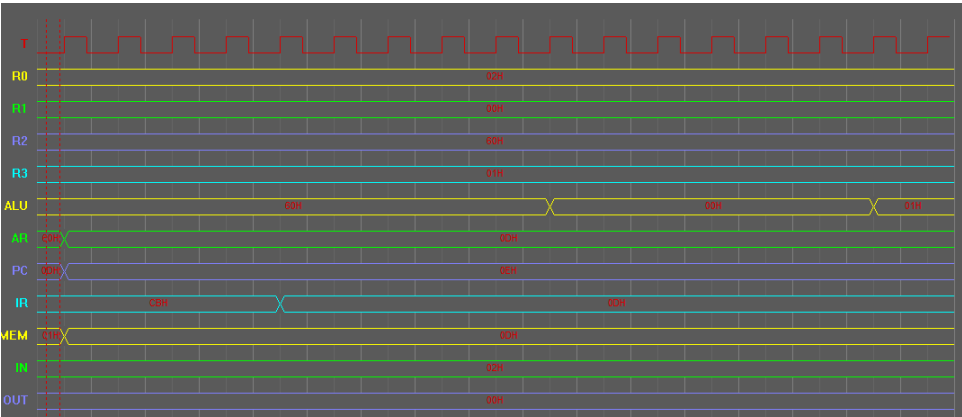


图 3-7 时序图

8) ADD R1,R3

地址	内容	助记符	说明
00001101	00001101	; ADD R1,R3	累加求和

1. 微指令执行过程
- 1) R1->A
  - 2) R3->B
  - 3) A+B->R1
2. 关键寄存器

表 3-8 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	60H->0DH
R1	00H->01H	PC	0DH->0EH
R2	60H	IR	CBH->0DH

R3	01H	MEM	01H->0DH
----	-----	-----	----------

3. 时序图

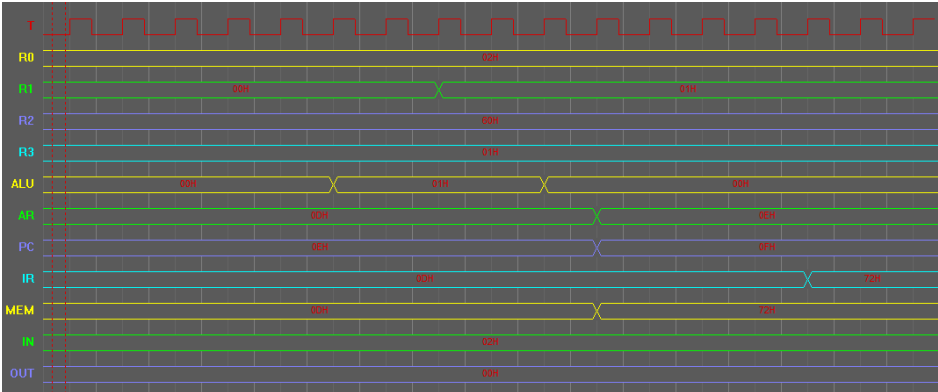


图 3-8 时序图

9) INC RI

地址	内容	助记符	说明
00001110	01110010	; INC RI	变址寄存器加 1,指向下一数据

- 1. 微指令执行过程
  - 1) RI->A
  - 2) (A)+1->RI
  - 3) A+B->R1
- 2. 关键寄存器

表 3-9 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	0DH->0EH
R1	01H	PC	0EH->0FH
R2	60H->61H	IR	0DH->72H
R3	01H	MEM	0DH->72H

3. 时序图

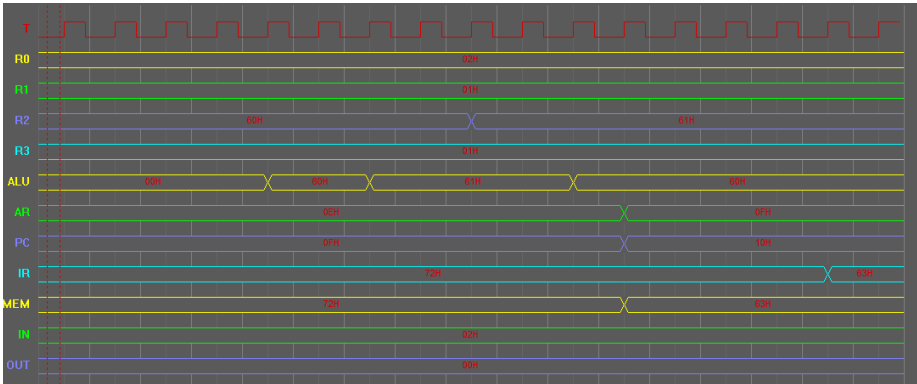


图 3-9 时序图

10) LDI R3,01H

地址	内容	助记符	说明
0000111	0110001	; LDI R3,01H	装入比较值
1	1		
000100	000000		
00	01		

1. 微指令执行过程
- 1) PC->AR;(PC)+1->PC
- 2) MEM->R3
2. 关键寄存器

表 3-10 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	0EH->0FH->10H
R1	01H	PC	0FH->10H->11H
R2	60H	IR	72H->63H
R3	01H	MEM	72H->63H->01H

3. 时序图

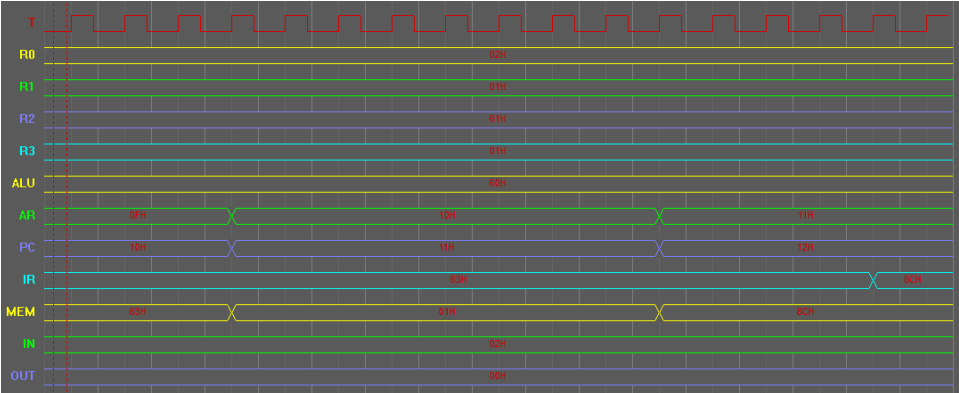


图 3-10 时序图

11) SUB R0,R3

地址	内容	助记符	说明
000100	1000110	; SUB R0,R3	两数相减
01	0		

1. 微指令执行过程
- 1) R0->A
- 2) R3->B
- 3) A-B->R0

2. 关键寄存器

表 3-11 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H->01H	AR	10H->11H
R1	01H	PC	11H->12H
R2	61H	IR	63H
R3	01H	MEM	01H->8CH

3. 时序图

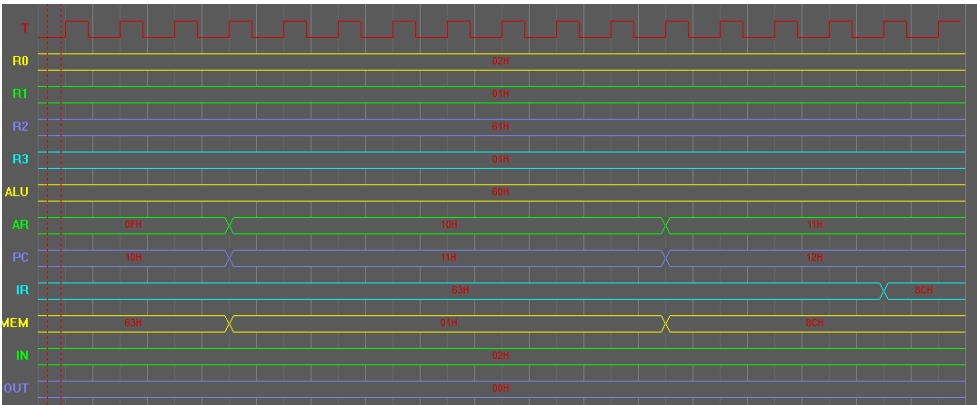


图 3-11 时序图

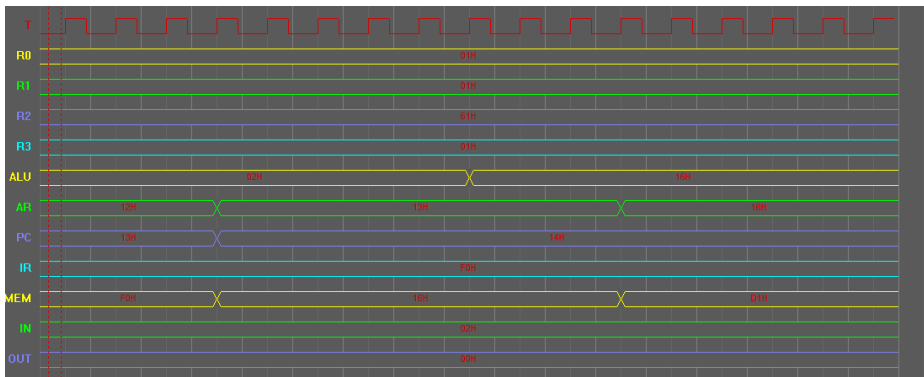
12) BZC RESULT

地址	内容	助记符	说明
0001001	1111000	; BZC RESULT	相减为 0, 表示求和完毕
0	0		
0001001	0001011		
1	0		

- 1. 微指令执行过程
  - 1) PC->AR;(PC)+1->PC
  - 2) MEM->A
  - 3) MEM->AR
  - 4) NOP,P<3>
  - 5) A->PC
- 2. 关键寄存器

表 3-12 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	01H	AR	11H->12H->13H->16H
R1	01H	PC	12H->13H-14H
R2	61H	IR	63H->F0H
R3	01H	MEM	8CH->F0H->16H->D1H



3. 时序图

图 3-12 时序图

13) JMP LOOP (继续循环)

地址	内容	助记符	说明
0001010	1110000	; JMP	未完则继续
0	0	LOOP	
0001010	0000101		
1	1		

- 1. 微指令执行过程
  - 1) PC->AR;(PC)+1->PC
  - 2) MEM->A

- ## 2. 关键寄存器

寄存器	内容变化	寄存器	内容变化
R0	01H	AR	16H->14H->15H->0BH
R1	01H	PC	14H->15H->16H->0BH
R2	61H	IR	F0H->E0H
R3	01H	MEM	D1H->E0H->0BH->CBH

### 3. 时序图



图 3-13 时序图

- 14) 重复执行(7)~(13)步直至循环结束

- 15) RESULT: STA 70H,R1

地址	内容	助记符	说明
0001011 0	1101000 1	; RESULT: STA 70H,R1	和存于 MEM 的 70H 单元
0001011 1	0111000 0		

- ## 2. 关键寄存器



表 3-15 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	01H	AR	16H->17H->70H
R1	03H	PC	16H->17H->18H
R2	62H	IR	F0H->D1H
R3	01H	MEM	D1H->70H->03H

3. 时序图



图 3-15 时序图

16) OUT 40H,R1

地址	内容	助记符	说明
0001100	0011010	; OUT 40H,R1	和在 OUT 单元显示
0	0		
0001100	0100000		
1	00		

1. 微指令执行过程

- 1) PC->AR;(PC)+1->PC
- 2) MEM->AR
- 3) R1->IO

2. 关键寄存器

表 3-16 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	01H	AR	70H->18H->19H->40H

R1	03H	PC	18H->19H->1AH
R2	62H	IR	D1H->34H
R3	01H	MEM	03H->34H->40H->00H

3. 时序图

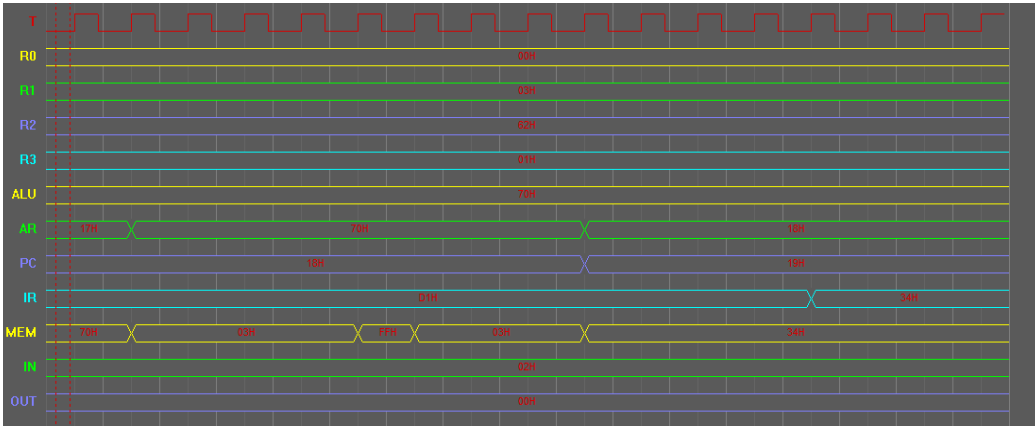


图 3-16 时序图

17) JMP START

地址	内容	助记符	说明
0001101	1110000	; JMP	跳转至 START
0	0	START	
0001101	000000		
1	00		

1. 微指令执行过程

- 1) PC->AR;(PC)+1->PC
- 2) MAM->A
- 3) MEM->AR
- 4) A->PC

2. 关键寄存器

表 3-17 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	01H	AR	40H->1AH->1BH->00H

R1	03H	PC	1AH->1BH->1CH
R2	62H	IR	34H->E0H
R3	01H	MEM	00H->EOH->00H

3. 时序图

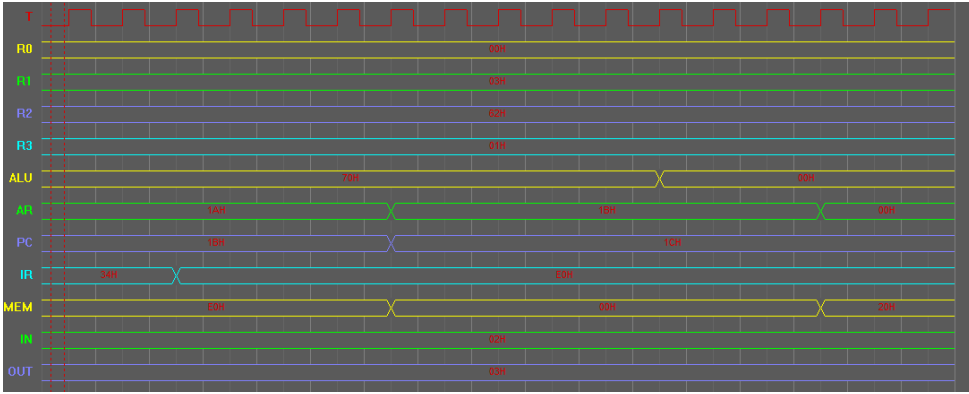


图 3-17 时序图

18) HLT

地址	内容	助记符	说明
0001110	0101000		
0	0	; HLT	停机

1. 微指令执行过程

1) NOP

四、 机器指令程序执行过程——选做部分

4.1 设计思路

根据现有指令，在模型机上实现以下运算：首先判断 x 是否为 0，若是，则直接跳转至输出 x；否则输出 2x-3。

4.2 机器代码

Code: Start Of Main Memory Data			
\$P 00 20	; START: IN R0,00H	读入计数初值	
\$P 01 00			
\$P 02 61	; LDI R1,0FH	立即数 0FH 送 R1	
\$P 03 0F			
\$P 04 14	; AND R0,R1	得到 R0 低四位	
\$P 05 61	; LDI R1,00H	装入和初值 00H	

\$P 06 00		
\$P 07 F0	; BZC RESULT	计数值为 0 则跳转
\$P 08 0E		
\$P 09 46	; MOV R2,R1:0110	令 R2=R1
\$P 0A 09	; ADD R1,R2:1001	令 R1=2*R1
\$P 0B 63	; LDI R3,03H	立即数 03H 送入 R3
\$P 0C 03		
\$P 0D 8D	; SUB R1,R3:1101	令 R1=R1-R3
\$P 0E D1	; RESULT: STA 70H,R1	和存于 MEM 的 70H 单元
\$P 0F 70		
\$P 10 34	; OUT 40H,R1	和在 OUT 单元显示
\$P 11 40		
\$P 12 E0	; JMP START	跳转至 START
\$P 13 00		
\$P 14 50	; HLT	停机

#### 4.2.1 指令分析

##### 1) START: IN R0,00H

地址	内容	助记符	说明
000000	001000	; START: IN	从 IN 单元读入计数初值
00	00	R0,00H	
000000	000000		
01	00		

##### 1. 微指令执行过程

- 4) PC->AR;(PC)+1->PC
- 5) MEM->AR
- 6) IO->R0

##### 2. 关键寄存器

表 4-1 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	0	AR	00H->01H->00H
R1	00H	PC	00H->01H->02H
R2	00H	IR	00H->20H
R3	00H	MEM	00H->20H

3. 时序图



图 4-1 时序图

2) LDI R1,0FH

地址	内容	助记符	说明
000000	0110000	; LDI R1,0FH	立即数 0FH 送 R1
10	1		
000000	0000111		
11	1		

1. 微指令执行过程

- 1) PC->AR;(PC)+1->PC
- 2) MEM->R1

2. 关键寄存器

表 4-2 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	00H->02H->03H
R1	00H->0FH	PC	02H->03H->04H
R2	00H	IR	20H->61H
R3	00H	MEM	20H->61H->0FH

3. 时序图

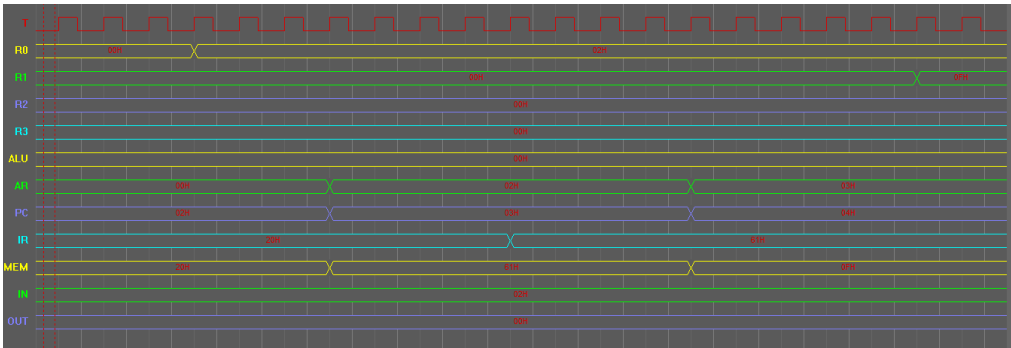


图 4-2 时序图

3) AND R0,R1

地址	内容	助记符	说明
000001	0001010	; AND R0,R1	得到 R0 低四
00	0		位

1. 微指令执行过程
- 1) R0->A
- 2) R1->B
- 3) A AND B->R0
2. 关键寄存器

表 4-3 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	03H->04H
R1	0FH	PC	04H->05H
R2	00H	IR	61H->14H
R3	00H	MEM	0FH->14H

3. 时序图

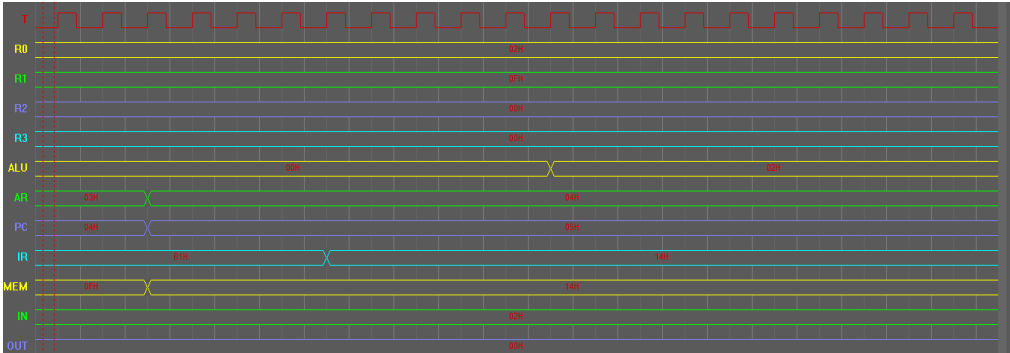


图 4-3 时序图

4) MOV R1,R0

地址	内容	助记符	说明
000001	010000	; MOV R1,R0	令 R1=R0
01	01		

1. 微指令执行过程
- 1) PC->AR;(PC)+1->PC

- 2) MEM->IR
- 3) R0->R1

2. 关键寄存器

表 4-4 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	04H->05H
R1	0FH->02H	PC	05H->06H
R2	00H	IR	14H->41H
R3	00H	MEM	14H->41H

3. 时序图

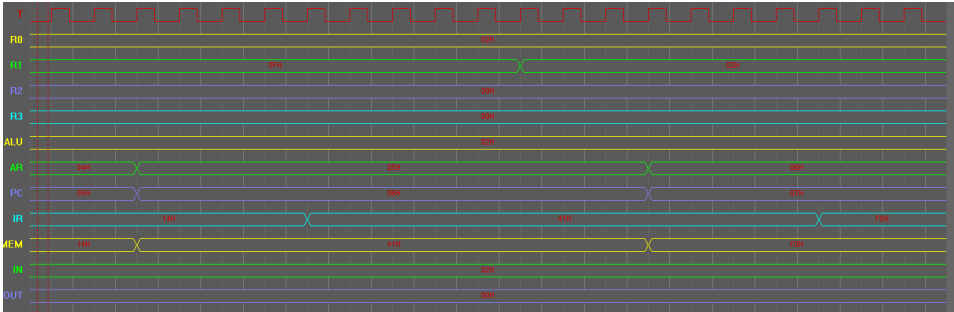


图 4-4 时序图

5) BZC RESULT

地址	内容	助记符	说明
0000011	1111000	; BZC RESULT	计数值为 0 则跳转
0	0		
0000011	0000110		
1	1		

- 1. 微指令执行过程
  - 1) PC->AR;(PC)+1->PC
  - 2) MEM->A
  - 3) MEM->AR
  - 4) NOP,P<3>
  - 5) A->PC

2. 关键寄存器

表 4-5 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	05H->06H->07H->0DH
R1	02H	PC	06H->07H->08H
R2	00H	IR	41H->F0H
R3	00H	MEM	41H>F0H->0DH->D1H

3. 时序图

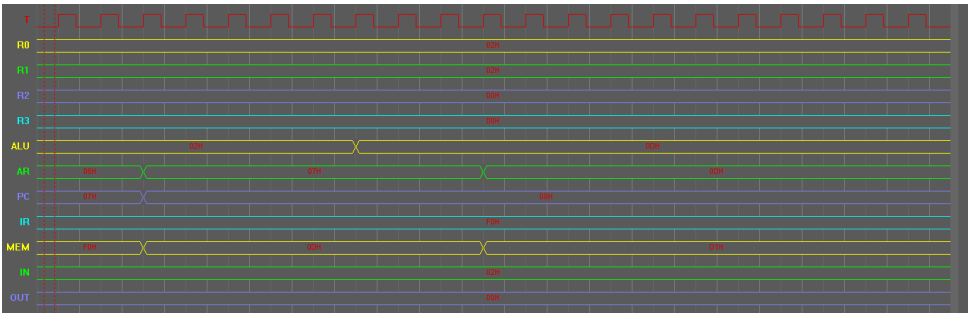


图 4-5 时序图

6) MOV R2, R1

地址	内容	助记符	说明
000010	0100011	; MOV R2, R1	令 R2=R1
00	0		

1. 微指令执行过程

- 1) PC->AR;(PC)+1->PC
- 2) MEM->IR
- 3) R1->R2
- 2) 关键寄存器

表 4-6 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	0DH->08H
R1	02H	PC	08H->09H
R2	00H->02H	IR	F0H->46H
R3	00H	MEM	D1H->46H

3) 时序图



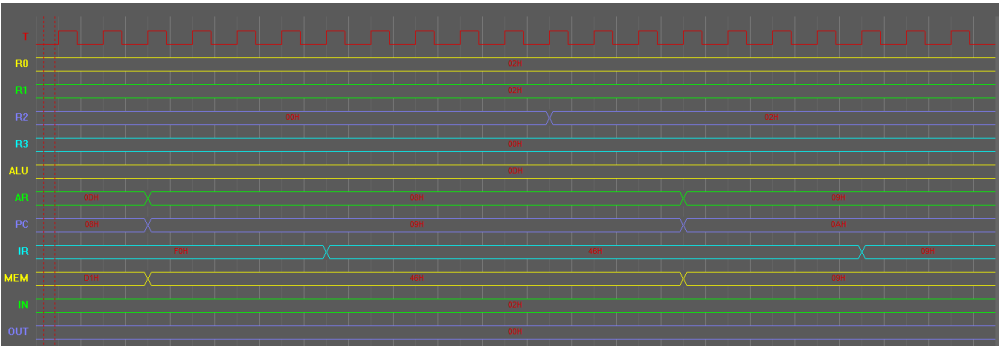


图 4-6 时序图

7) ADD R1, R2

地址	内容	助记符	说明
000010	000010	; ADD R1,R2	令 R1=R1+R2=2*R1
01	01		

1. 微指令执行过程

- 1) R1->A
- 2) R2->B
- 3) A+B->R1

2. 关键寄存器

表 4-7 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	08H->09H
R1	02H->04H	PC	09H->0AH
R2	02H	IR	46H->09H
R3	00H	MEM	46H->09H

3. 时序图

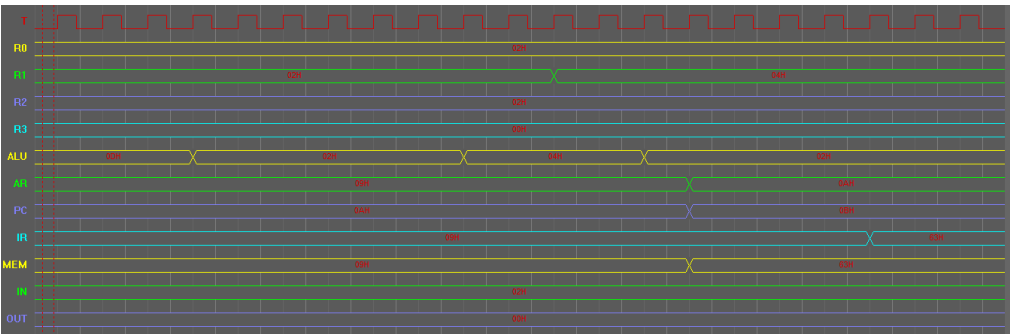


图 4-7 时序图

8) LDI R3,03H

地址	内容	助记符	说明
0000101	0110001	; LDI R3,03H	立即数 03H 送入 R3

0	1
0000101	000000
1	11

- 1. 微指令执行过程
  - 1) PC->AR;(PC)+1->PC
  - 2) MEM->R3
- 2. 关键寄存器

表 4-8 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	09H->0AH->0BH
R1	04H	PC	0AH->0BH->0CH
R2	03H	IR	09H->63H
R3	00H->02H	MEM	09H->63H->03H

3. 时序图

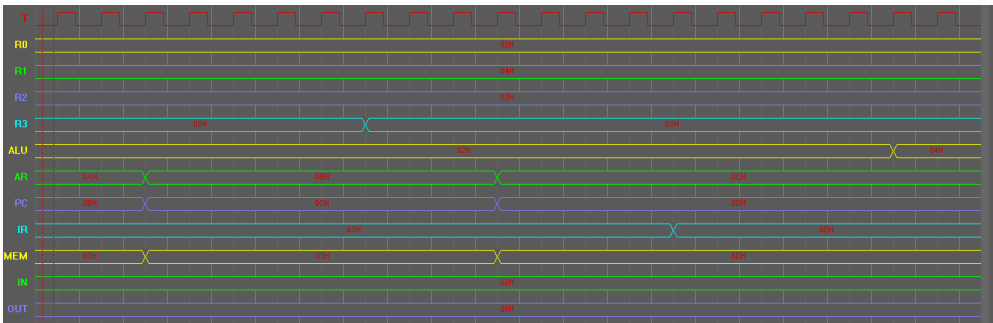


图 4-8 时序图

9) SUB R1, R3

地址	内容	助记符	说明
0000110	1000110		
0	1	; SUB R1, R3	令 R1=R1-R3

- 1. 微指令执行过程
  - 1) R1->A
  - 2) R3->B
  - 3) A-B->R1
- 2. 关键寄存器

表 4-9 寄存器变化

寄存器	内容变化	寄存器	内容变化
-----	------	-----	------

R0	02H	AR	0BH->0CH
R1	04H->01H	PC	0CH->0DH
R2	02H	IR	63H->8DH
R3	00H	MEM	03H->8DH

3. 时序图

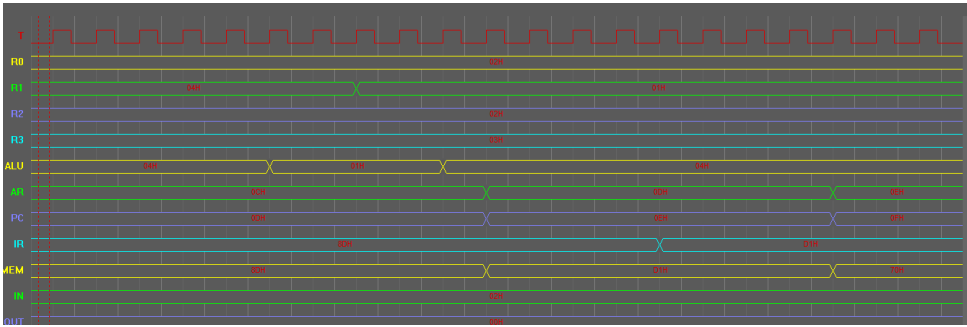


图 4-9 时序图

10) RESULT: STA 70H, R1

地址	内容	助记符	说明
0000110	1101000	; RESULT: STA 70H, R1	和存于 MEM 的 70H 单元
1	1		
0000111	0111000		
0	0		

1. 微指令执行过程

- 1) PC->AR;(PC)+1->PC
- 2) MEM->A
- 3) MEM->AR
- 4) R1->MEM

2. 关键寄存器

表 4-10 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	0CH->0DH->0EH->70H
R1	01H	PC	0DH->0EH->0FH
R2	02H	IR	8DH->D1H
R3	03H	MEM	70H->FDH->FFH->01H

3. 时序图

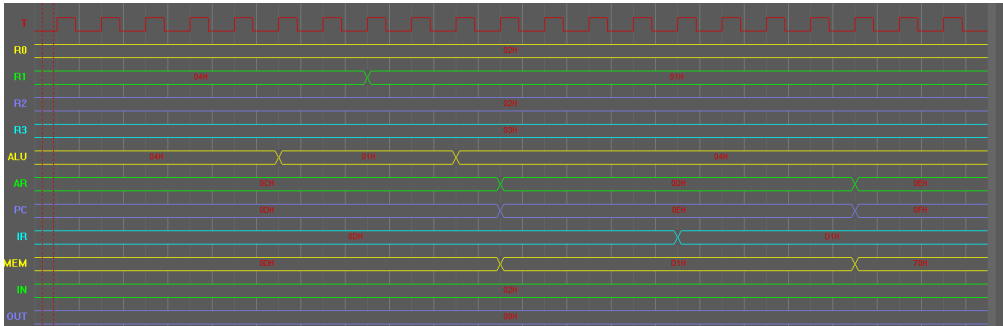


图 4-10 时序图

11) OUT 40H, R1

地址	内容	助记符	说明
0000111	0011010	; OUT 40H, R1	和在 OUT 单元显示
1	0		
000100	010000		
00	00		

1. 微指令执行过程

- 1) PC->AR;(PC)+1->PC
- 2) MEM->AR
- 3) R1->IO

2. 关键寄存器

表 4-11 寄存器变化

寄存器	内容变化	寄存器	内容变化
R0	02H	AR	70H->0FH->10H->40H
R1	01H	PC	0FH->10H->11H
R2	02H	IR	D1H->34H
R3	03H	MEM	01H->34H->40H->00H

3. 时序图

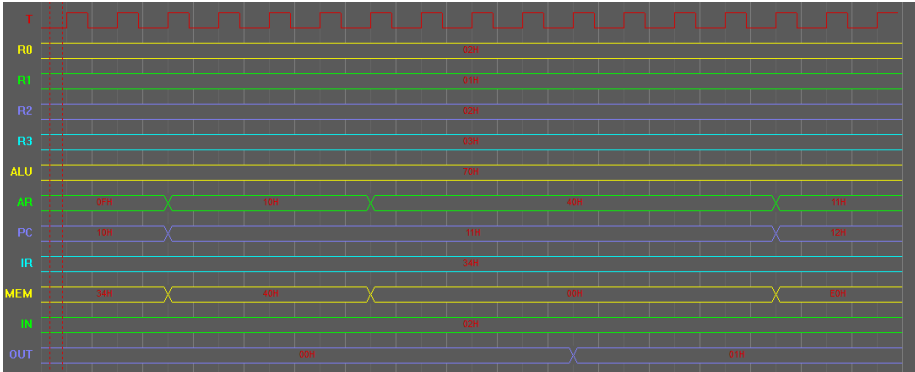


图 4-11 时序图

如上图，OUT 单元显示当  $x=01H$  时  $2x-1$  为  $01H$ 。