

课程报告

课程名称： 计算机组成原理

报告题目： 复杂模型机设计实验

所在院系： 计算学部

所在专业： 人工智能

学生姓名：

学生学号：

选课时间： 2023年春季学期

评阅成绩：

# 复杂模型机设计实验

#### 处理器功能及指令系统定义

1. **数据格式定义**

模型机规定采用定点无符号数表示数据，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≤X≤28－1。

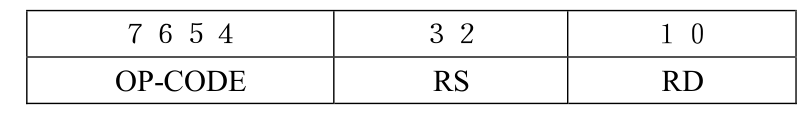
1. **指令设计和指令格式**

**指令设计：**

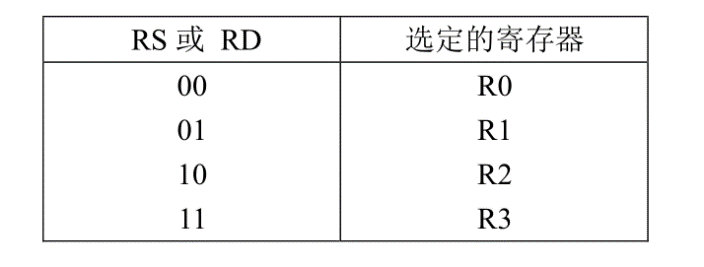
模型机设计三大类指令共十五条，其中包括运算类指令、控制转移类指令，数据传送类指令。运算类指令包含三种运算，算术运算、逻辑运算和移位运算，设计有 6 条运算类指令，分别为：ADD、AND、INC、SUB、OR、RR，所有运算类指令都为单字节，寻址方式采用寄存器直接寻址。控制转移类指令有三条 HLT、JMP、BZC，用以控制程序的分支和转移，其中 HLT 为单字节指令，JMP 和 BZC 为双字节指令。数据传送类指令有 IN、OUT、MOV、LDI、LAD、STA 共 6 条，用以完成寄存器和寄存器、寄存器和 I/O、寄存器和存储器之间的数据交换，除MOV 指令为单字节指令外，其余均为双字节指令。

**指令格式：**

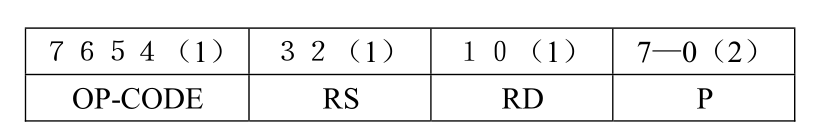
所有单字节指令（ADD、AND、INC、SUB、OR、RR、HLT 和 MOV）格式如下：



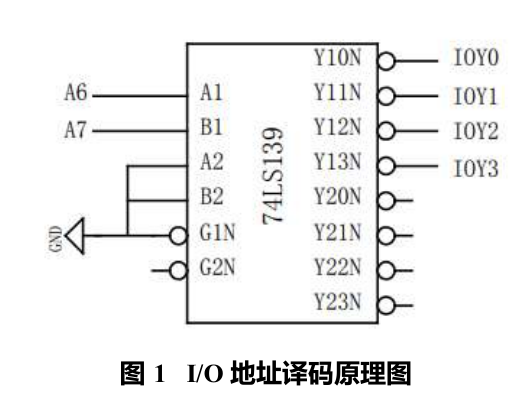
其中，OP-CODE 为操作码，RS为源寄存器，RD为目的寄存器，并规定:



IN和OUT的指令格式为：



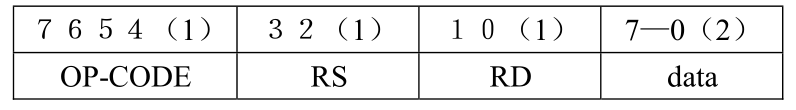
其中括号中的1表示指令的第一字节，2表示指令的第二字节，OP-CODE 为操作码，RS为源寄存器，RD为目的寄存器，P为I/O端口号，占用一个字节，系统的 I/O地址译码原理见图1（在地址总线单元）。



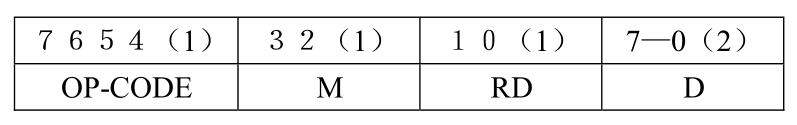
由于用的是地址总线的高两位进行译码，I/O地址空间被分为四个区，如表1所示：



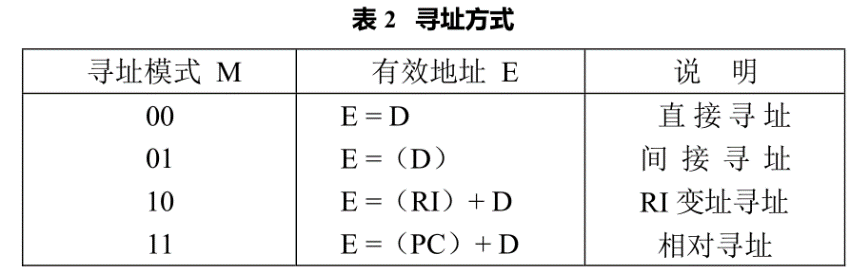
系统设计五种数据寻址方式，即立即、直接、间接、变址和相对寻址，LDI指令为立即寻址，LAD、STA、JMP 和 BZC指令均具备直接、间接、变址和相对寻址能力。

LDI的指令格式如下，第一字节同前一样，第二字节为立即数。 

LAD、STA、JMP和 BZC指令格式如下。

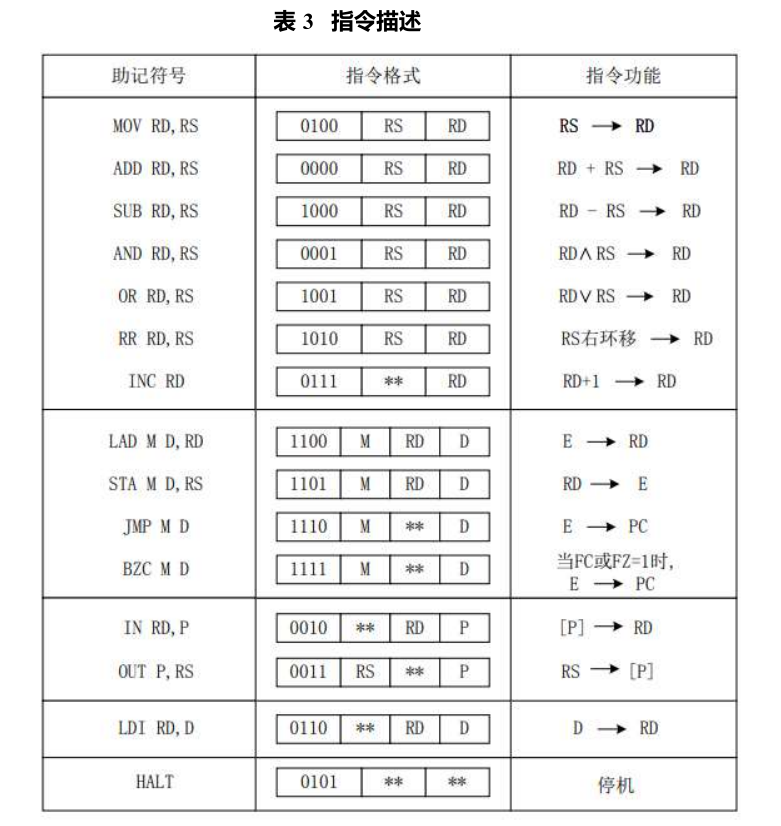


其中M为寻址模式，具体见表2，以 R2作为变址寄存器RI。



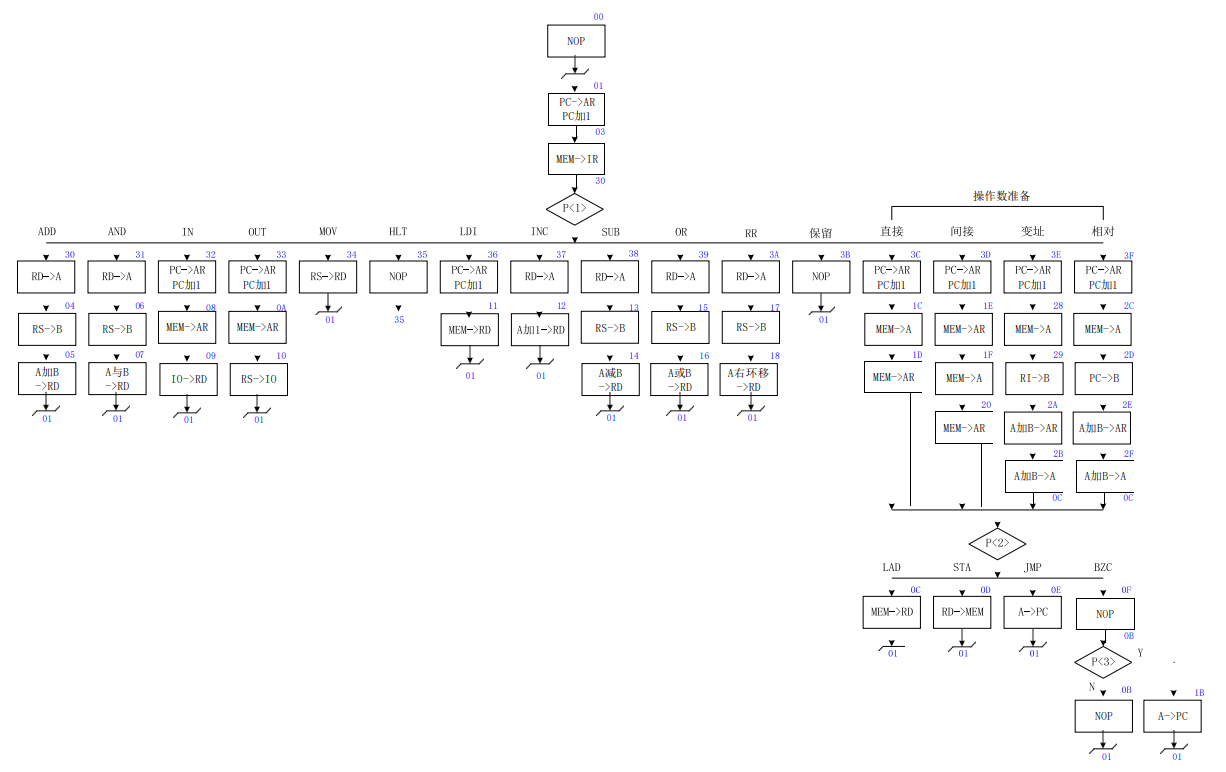
1. **指令系统**

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。

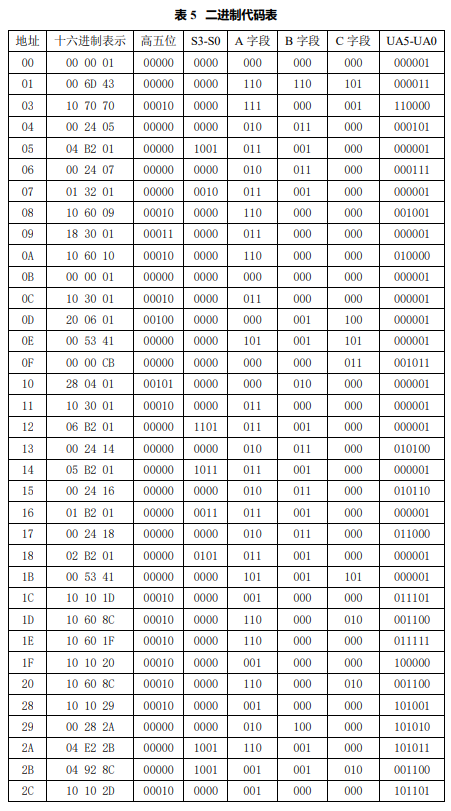


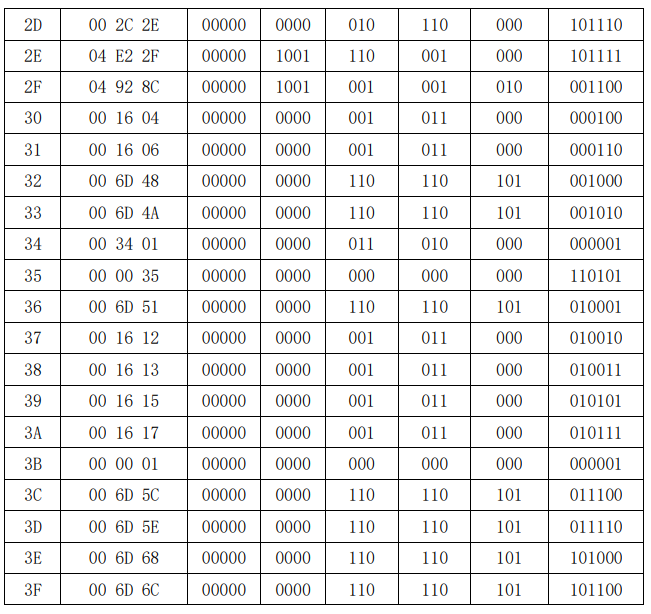
#### 微程序设计

**微程序流程图**：



**二进制代码表:**





#### 机器指令程序执行过程

**3.1 必选部分：**

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。

根据要求可以得到如下程序，地址和内容均为二进制数。

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |  |  |
| 00000010 | 01100001 | ; LDI R1,0FH | 立即数 0FH 送 R1 |
| 00000011 | 00001111 |  |  |
| 00000100 | 00010100 | ; AND R0,R1 | 得到 R0 低四位 |
| 00000101 | 01100001 | ; LDI R1,00H | 装入和初值 00H |
| 00000110 | 00000000 |  |  |
| 00000111 | 11110000 | ; BZC RESULT | 计数值为 0 则跳转 |
| 00001000 | 00010110 |  |  |
| 00001001 | 01100010 | ; LDI R2,60H | 读入数据始地址 |
| 00001010 | 01100000 |  |  |
| 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 |  |  |
| 00001101 | 00001101 | ; ADD R1,R3 | 累加求和 |
| 00001110 | 01110010 | ; INC RI | 变址寄存加 1，指向下一数据 |
| 00001111 | 01100011 | ; LDI R3,01H | 装入比较值 |
| 00010000 | 00000001 |  |  |
| 00010001 | 10001100 | ; SUB R0,R3 |  |
| 00010010 | 11110000 | ; BZC RESULT | 相减为 0，表示求和完毕 |
| 00010011 | 00010110 |  |  |
| 00010100 | 11100000 | ; JMP LOOP | 未完则继续 |
| 00010101 | 00001011 |  |  |
| 00010110 | 11010001 | ; RESULT:  STA 70H,R1 | 和存于 MEM 的 70H 单元 |
| 00010111 | 01110000 |  |  |
| 00011000 | 00110100 | ; OUT 40H,R1 | 和在 OUT 单元显示 |
| 00011001 | 01000000 |  |  |
| 00011010 | 11100000 | ; JMP START | 跳转至 START |
| 00011011 | 00000000 |  |  |
| 00011100 | 01010000 | ; HLT | 停机 |
| 01100000 | 00000001 | ; 数据 |  |
| 01100001 | 00000010 |  | |
| 01100010 | 00000011 |  | |
| 01100011 | 00000100 |  | |
| 01100100 | 00000101 |  | |
| 01100101 | 00000110 |  | |
| 01100110 | 00000111 |  | |
| 01100111 | 00001000 |  | |
| 01101000 | 00001001 |  | |
| 01101001 | 00001010 |  | |
| 01101010 | 00001011 |  | |
| 01101011 | 00001100 |  | |
| 01101100 | 00001101 |  | |
| 01101101 | 00001110 |  | |
| 01101110 | 00001111 |  | |

**微指令执行过程：**

取指阶段各机器指令的微指令执行过程都相同，统一为

PC->AR，(PC)+1->PC，MEM->IR

执行阶段各机器指令的微指令执行过程：

1. IN R0, 00H

PC->AR，(PC)+1->PC

MEM->AR

IO->R0

关键寄存器变化：IO(00H)->R0

1. LDI R1, 0FH

PC->AR，(PC)+1->PC

MEM->R1

关键寄存器变化：0FH->R1

1. AND R0, R1

R0->A

R1->B

A与B->R0

关键寄存器变化：R0与R1->R0

1. LDI R1, 00H

PC->AR，(PC)+1->PC

MEM->R1

关键寄存器变化：00H->R1

1. BZC RESULT

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

(FC+FZ) \*A+(FC+FZ) \*PC->PC

关键寄存器变化：(FC+FZ) \*16H+(FC+FZ) \*PC->PC

1. LDI R2, 60H

PC->AR，(PC)+1->PC

MEM->R2

关键寄存器变化：60H->R2

1. LAD R3, [RI], 00H

PC->AR，(PC)+1->PC

MEM->A，RI->B

A+B->AR，A+B->A

MEM->RD

关键寄存器变化：[(RI)+00H]->R3 （这里RI实际是R2）

1. ADD R1, R3

R1->A

R3->B

A+B->R1

关键寄存器变化：(R1)+(R3)->R1

1. INC RI

RI->A

(A)+1->RI

关键寄存器变化：(RI)+1->RI

1. LDI R3, 01H

PC->AR，(PC)+1->PC

MEM->R3

关键寄存器变化：01H->R3

1. SUB R0, R3

R0->A

R3->B

A-B->R0

关键寄存器变化：(R0)-(R3)->R0

1. BZC RESULT

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

(FC+FZ) \*A+(FC+FZ) \*PC->PC

关键寄存器变化：(FC+FZ) \*A+(FC+FZ) \*PC->PC

1. JMP LOOP

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

A->PC

关键寄存器变化：0BH->PC

1. STA 70H, R1

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

R1->MEM

关键寄存器变化：R1->MEM(70H)

1. OUT 40H, R1

PC->AR，(PC)+1->PC

MEM->AR

R1->IO

关键寄存器变化：R1->IO(40H)

1. JMP START

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

A->PC

关键寄存器变化：00H->PC

1. HLT

NOP

**时序图：**

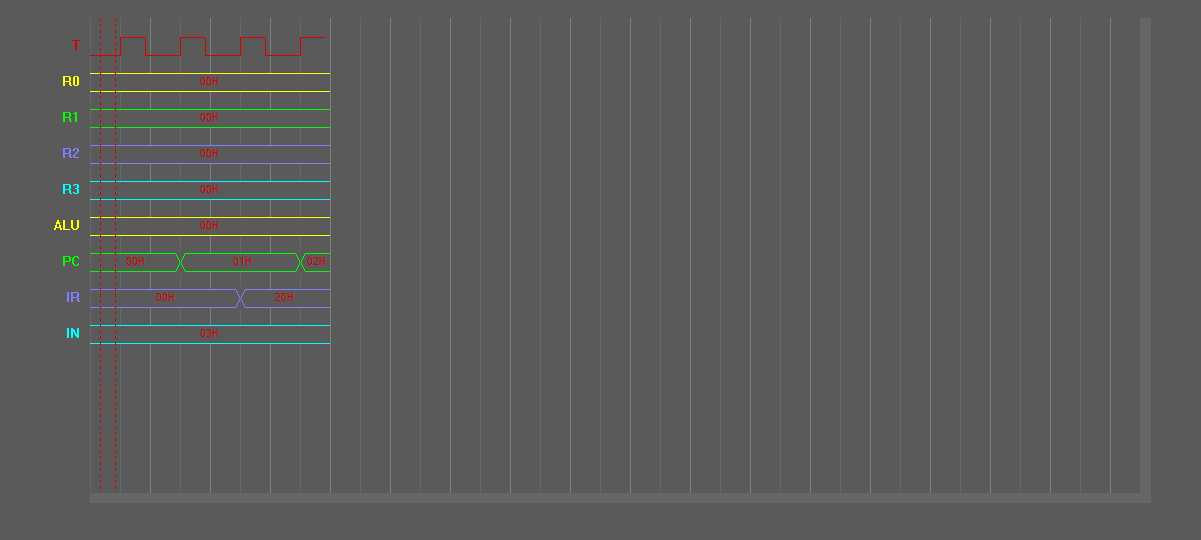


图3.1.1起始寄存器清零

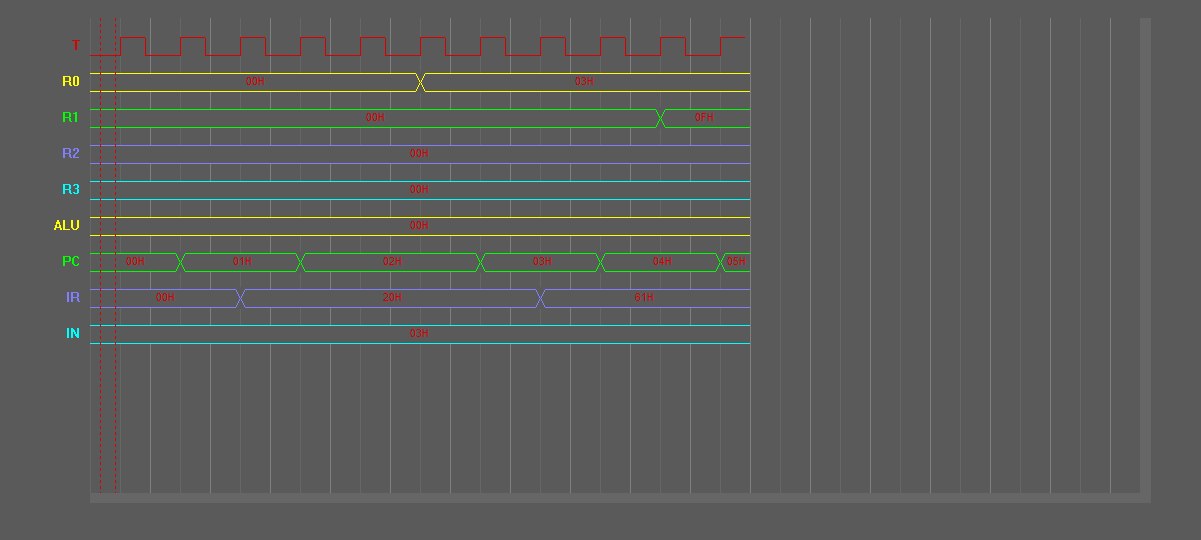


图3.1.2 读入R1

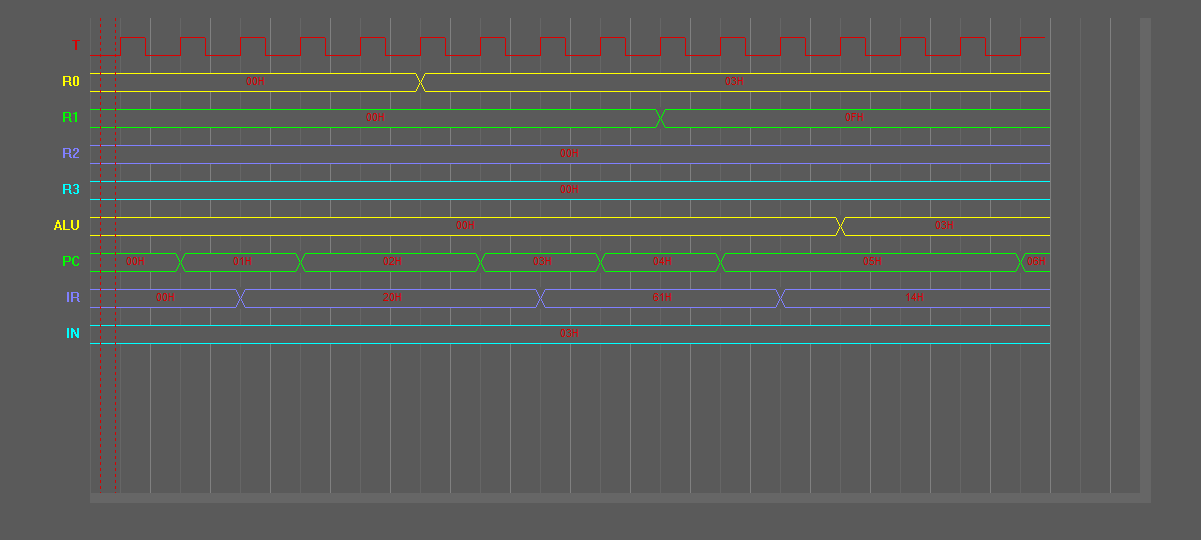


图3.1.3 低位求与

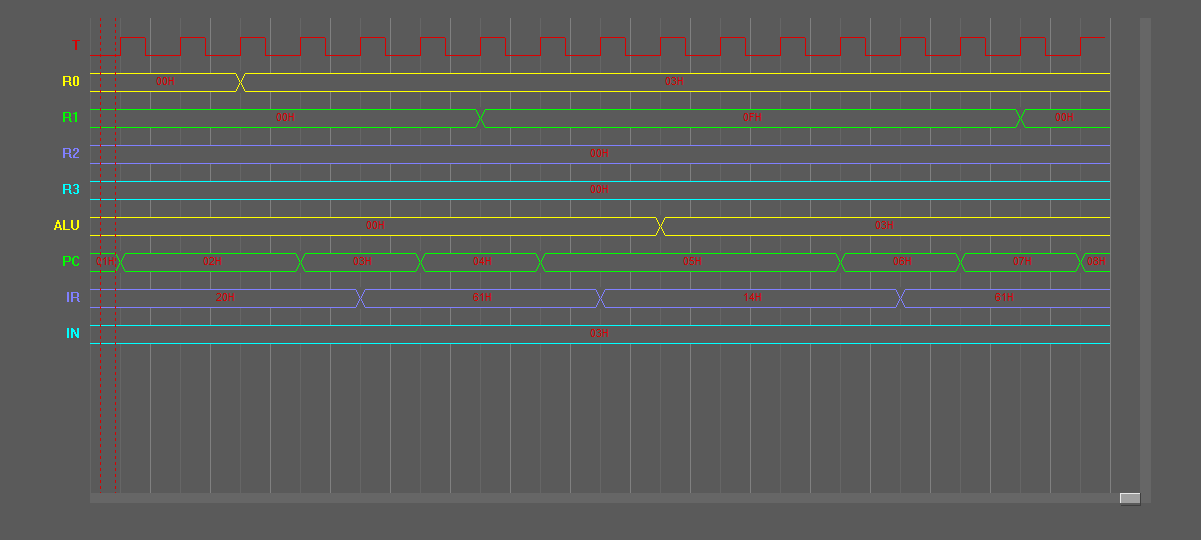


图3.1.4 R1清零

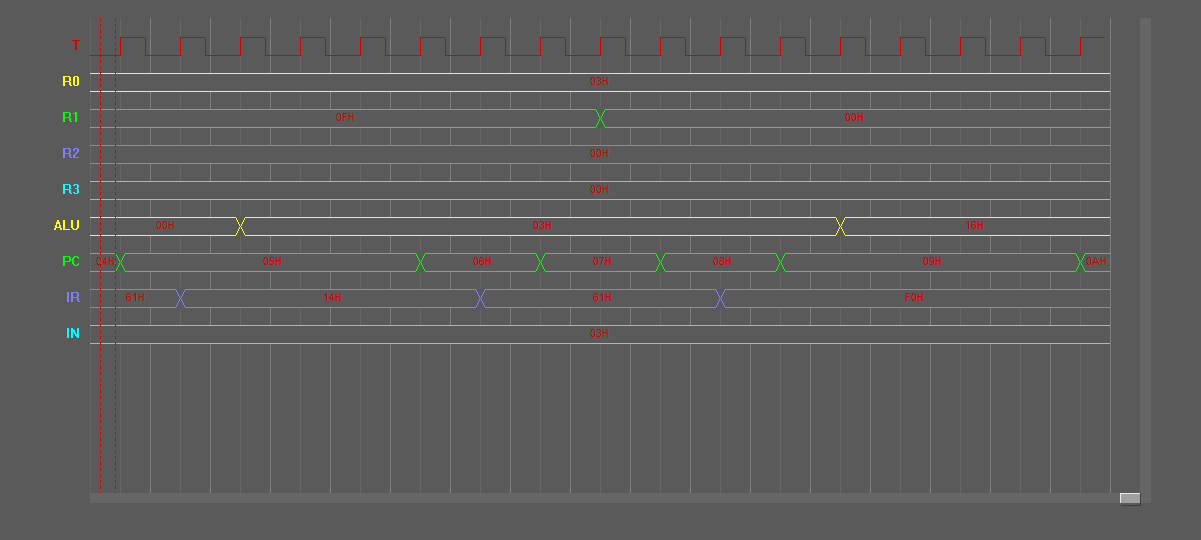


图3.1.5 判断是否为0

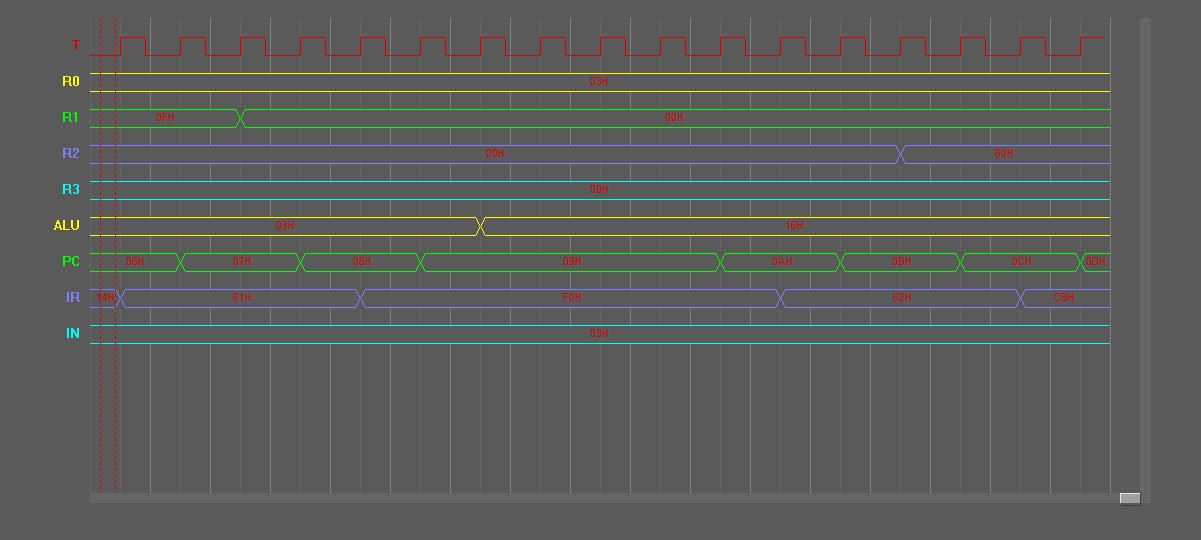


图3.1.6 读入R2

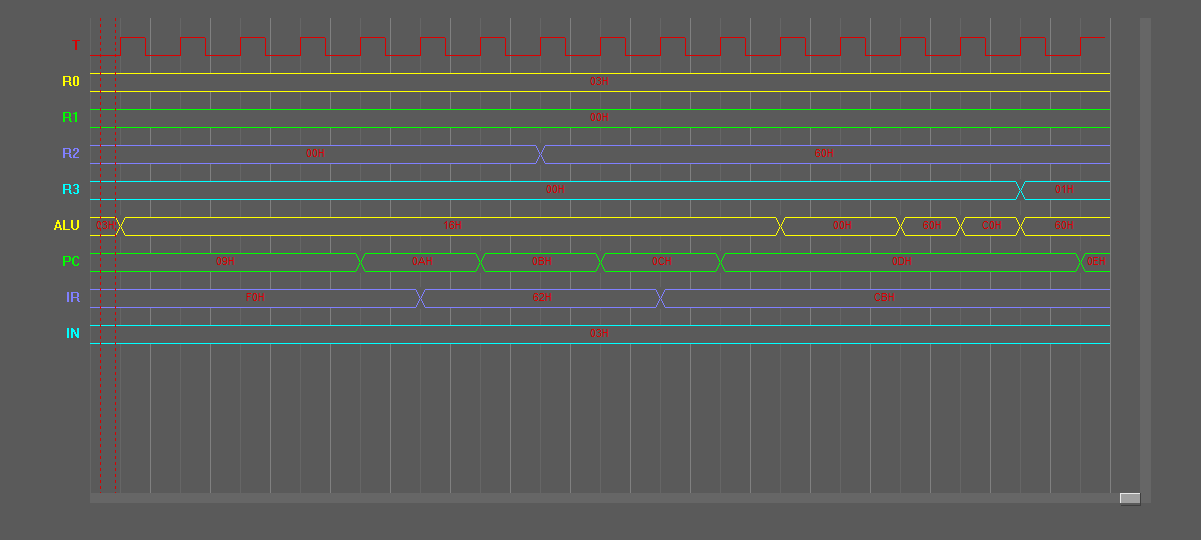


图3.1.7 将MEM数据读入R3

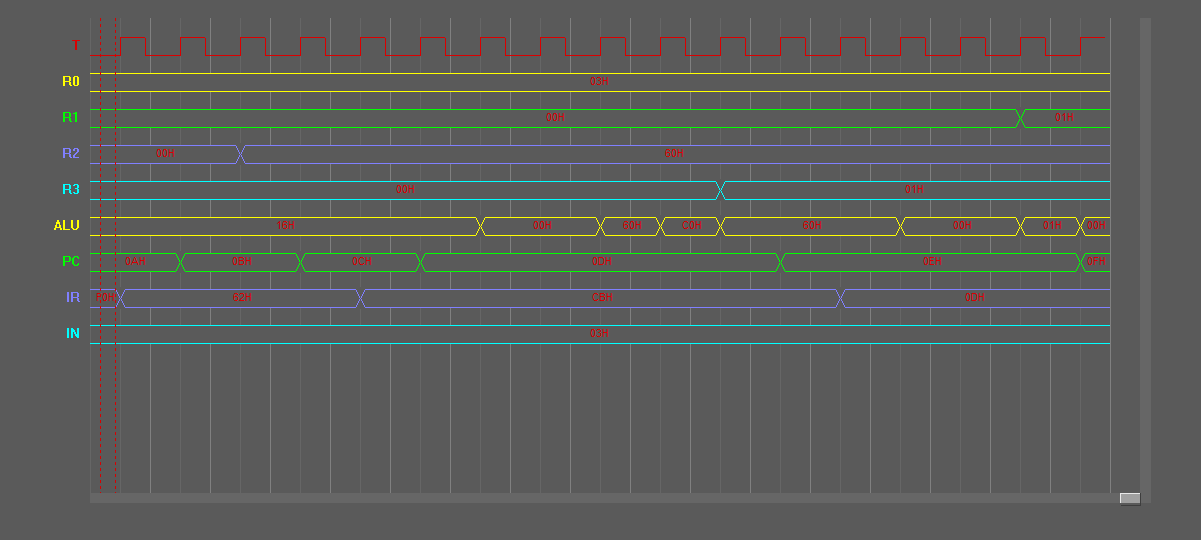


图3.1.8 R1变为和

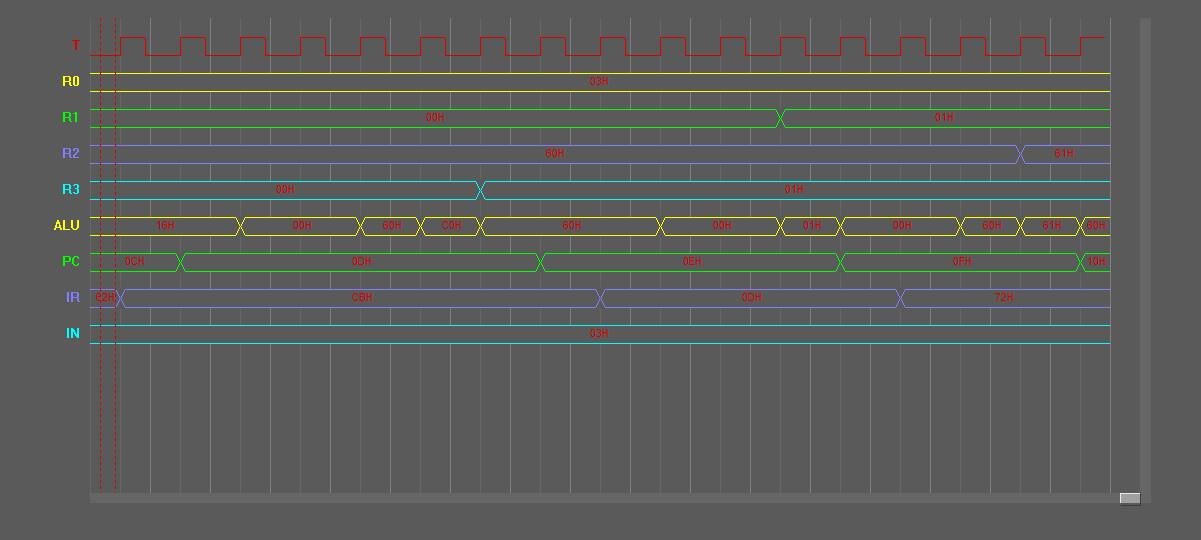


图3.1.9 变址寄存器（R2）自增

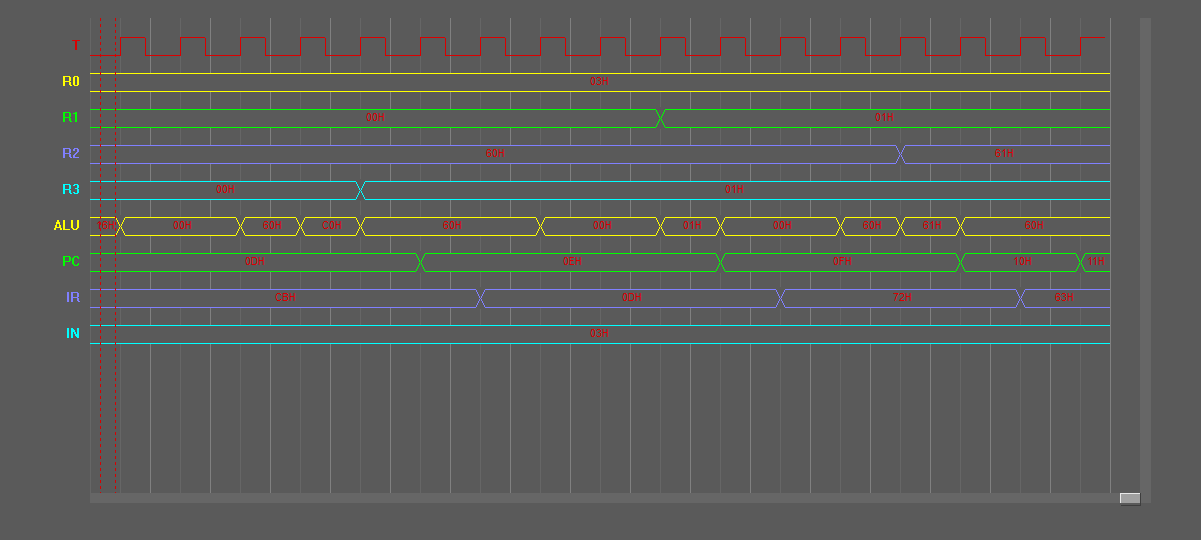


图3.1.10 读入R3

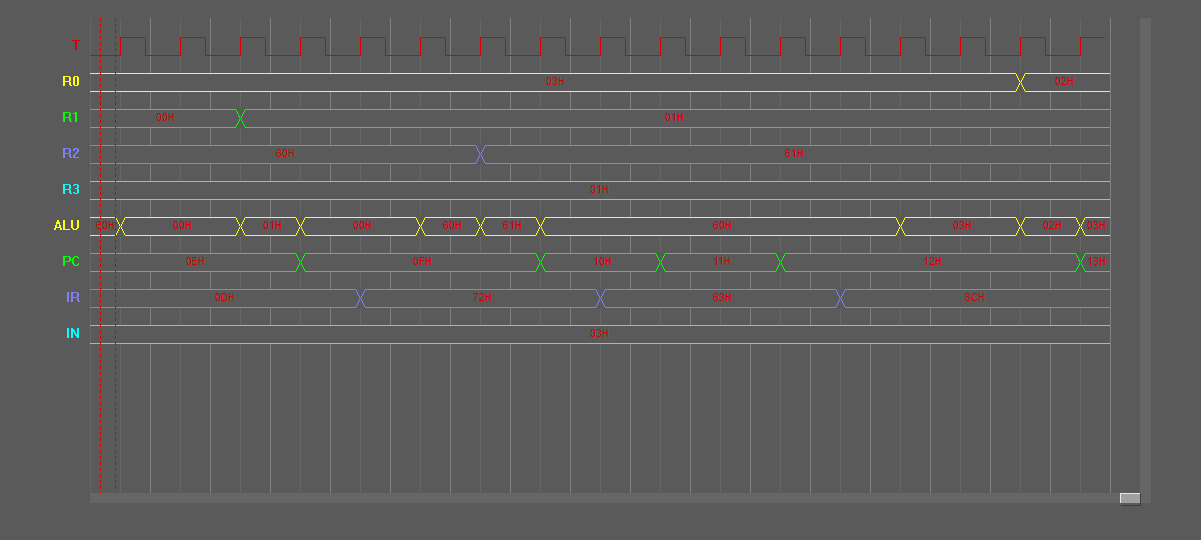


图3.1.11 计数器（R0）减一

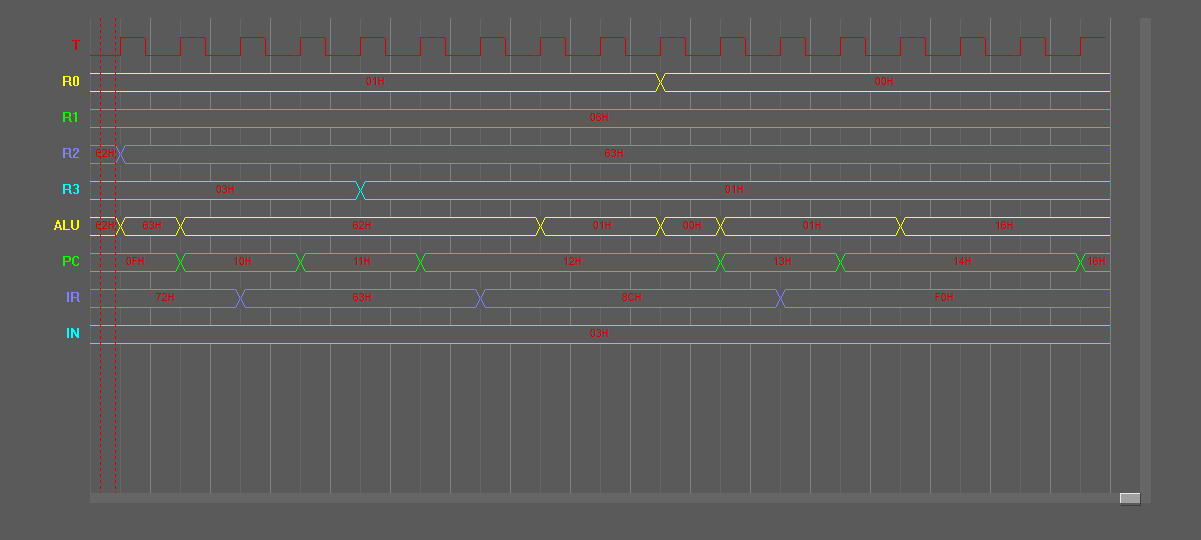


图3.1.12 最终跳出循环（PC信号从14H直接变为16H）

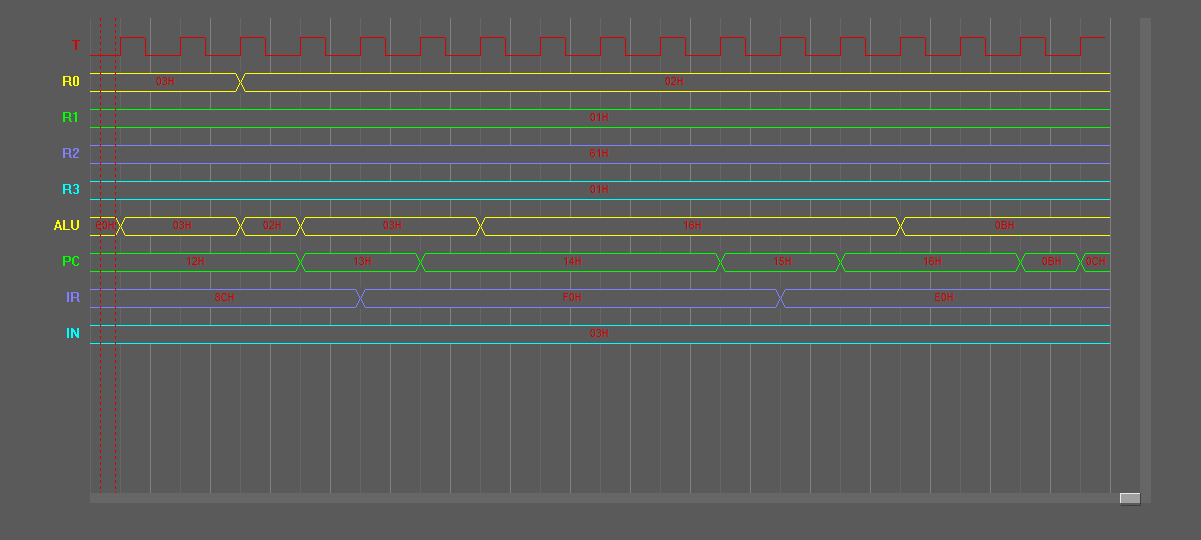


图3.1.13 跳到循环入口（PC从16H变到0BH）

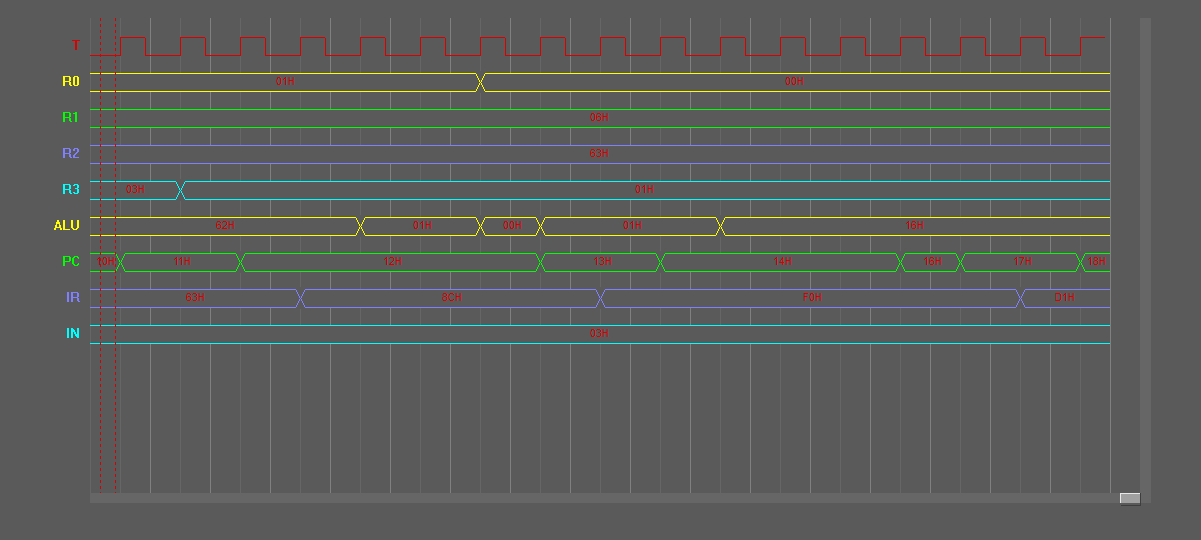


图3.1.14 存结果

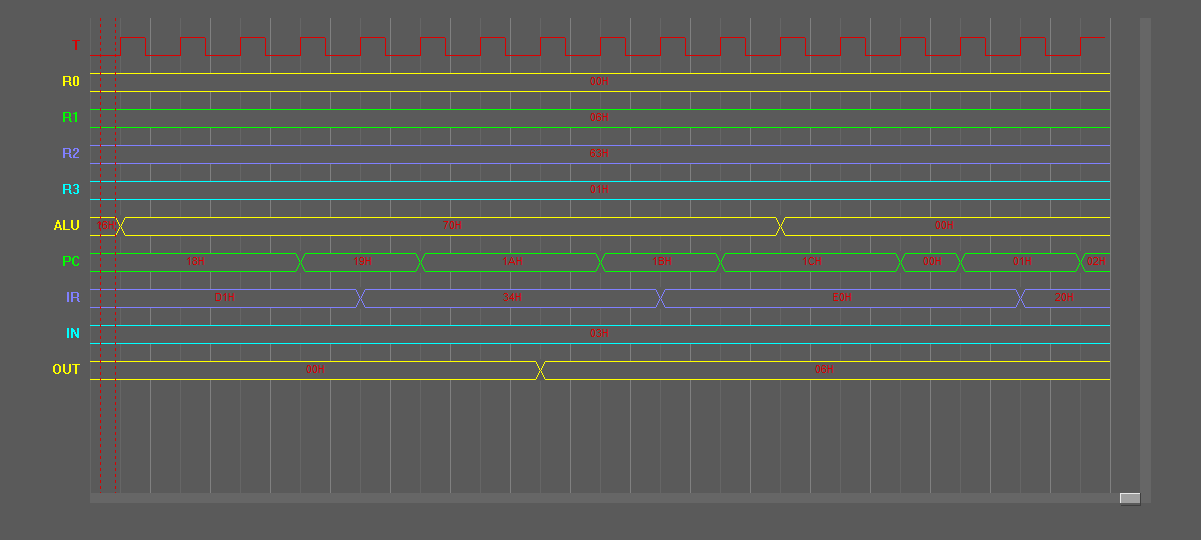


图3.1.15 OUT输出

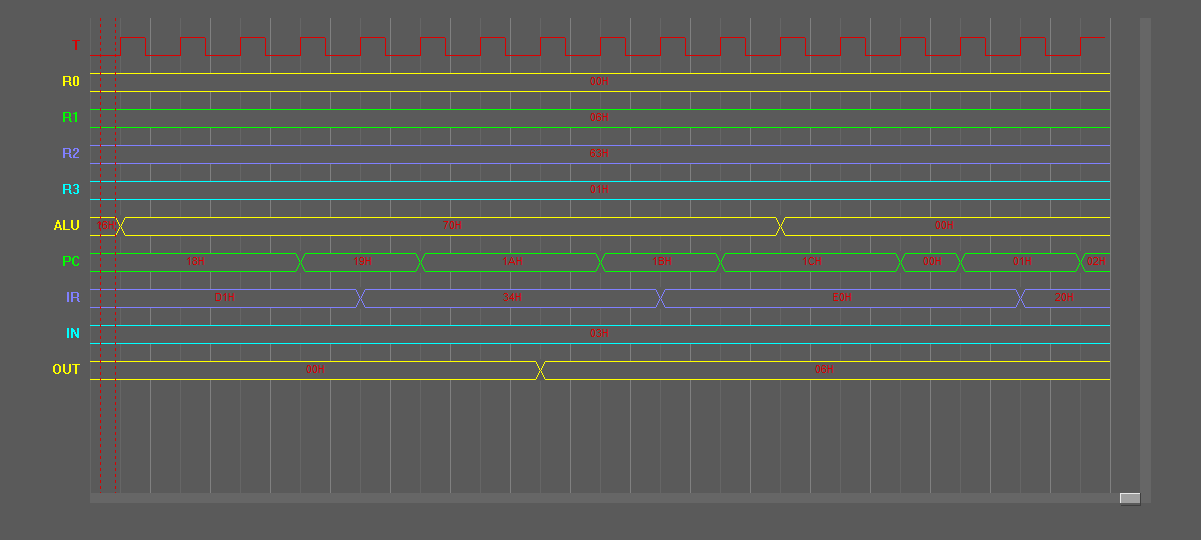


图3.1.16 跳回程序初始地址（PC变为00H）

**3.2 选做部分：**

程序功能：从IN单元读入低四位作为x，实现从(x+1)+(x+2)+...+15的计算。01H 到 0FH 共 15个数据存于 60H 到 6EH 单元。

程序设计：先设定变址寄存器中的值为60H+X,然后做一个减法，得到循环次数存于寄存器R0中，然后进入循环进行累加，最后输出结果。

验证结果：输入13（十进制），得到1D（=14+15），结果符合预期，并且程序运行过程中寄存器变化符合预期，程序成功正常运行。

根据要求可以得到如下程序，地址和内容均为二进制数。

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |  |  |
| 00000010 | 01100001 | ; LDI R1,0FH | 立即数 0FH 送 R1 |
| 00000011 | 00001111 |  |  |
| 00000100 | 00010100 | ; AND R0,R1 | 得到 R0 低四位 |
| 00000101 | 01100001 | ; LDI R1,00H | 装入和初值 00H |
| 00000110 | 00000000 |  |  |
| 00000111 | 01100010 | ; LDI R2,60H | 读入数据始地址 |
| 00001000 | 01100000 |  |  |
| 00001001 | 00000010 | ADD R2,R0 | 加上地址偏移 |
| 00001010 | 01100011 | LDI R3,0FH | 将15装入R3 |
| 00001011 | 00001111 |  |  |
| 00001100 | 10000011 | SUB R3,R0 | 将R3中的15减去R0，得到循环次数 |
| 00001101 | 01001100 | MOV R0,R3 | 循环次数放入R0 |
| 00001110 | 11110000 | ; BZC RESULT | 计数值为 0 则跳转 |
| 00001111 | 00011011 |  |  |
| 00010000 | 11001011 | ; LOOP:  LAD R3,[RI],00H | 从 MEM 读入数据送 R3，变址寻址，偏移量为 00H |
| 00010001 | 00000000 |  |  |
| 00010010 | 00001101 | ; ADD R1,R3 | 累加求和 |
| 00010011 | 01110010 | ; INC RI | 变址寄存加 1，指向下一数据 |
| 00010100 | 01100011 | ; LDI R3,01H | 装入比较值 |
| 00010101 | 00000001 |  |  |
| 00010110 | 10001100 | ; SUB R0,R3 |  |
| 00010111 | 11110000 | ; BZC RESULT | 相减为 0，表示求和完毕 |
| 00011000 | 00011011 |  |  |
| 00011001 | 11100000 | ; JMP LOOP | 未完则继续 |
| 00011010 | 00010000 |  |  |
| 00011011 | 11010001 | ; RESULT:  STA 70H,R1 | 和存于 MEM 的 70H 单元 |
| 00011100 | 01110000 |  |  |
| 00011101 | 00110100 | ; OUT 40H,R1 | 和在 OUT 单元显示 |
| 00011110 | 01000000 |  |  |
| 00011111 | 11100000 | ; JMP START | 跳转至 START |
| 00100000 | 00000000 |  |  |
| 00100001 | 01010000 | ; HLT | 停机 |
| 01100000 | 00000001 | ; 数据 |  |
| 01100001 | 00000010 |  | |
| 01100010 | 00000011 |  | |
| 01100011 | 00000100 |  | |
| 01100100 | 00000101 |  | |
| 01100101 | 00000110 |  | |
| 01100110 | 00000111 |  | |
| 01100111 | 00001000 |  | |
| 01101000 | 00001001 |  | |
| 01101001 | 00001010 |  | |
| 01101010 | 00001011 |  | |
| 01101011 | 00001100 |  | |
| 01101100 | 00001101 |  | |
| 01101101 | 00001110 |  | |
| 01101110 | 00001111 |  | |

**微指令执行过程：**

取指阶段各机器指令的微指令执行过程都相同，统一为

PC->AR，(PC)+1->PC，MEM->IR

执行阶段各机器指令的微指令执行过程如下：

1. IN R0, 00H

PC->AR，(PC)+1->PC

MEM->AR

IO->R0

关键寄存器变化：IO(00H)->R0

1. LDI R1, 0FH

PC->AR，(PC)+1->PC

MEM->R1

关键寄存器变化：0FH->R1

1. AND R0, R1

R0->A

R1->B

A与B->R0

关键寄存器变化：R0与R1->R0

1. LDI R1,00H

PC->AR，(PC)+1->PC

MEM->R1

关键寄存器变化：00H->R1

1. LDI R2,60H

PC->AR，(PC)+1->PC

MEM->R2

关键寄存器变化：60H->R2

1. ADD R2,R0

R2->A

R0->B

A+B->R2

关键寄存器变化：(R2)+(R0)->R2

1. LDI R3,0FH

PC->AR，(PC)+1->PC

MEM->R3

关键寄存器变化：0FH->R3

1. SUB R3,R0

R0->A

R3->B

A-B->R3

关键寄存器变化：(R3)-(R0)->R3

1. MOV R0,R3

R3->R0

1. BZC RESULT

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

(FC+FZ) \*A+(FC+FZ) \*PC->PC

关键寄存器变化：(FC+FZ) \*16H+(FC+FZ) \*PC->PC

1. LAD R3,[RI],00H

PC->AR，(PC)+1->PC

MEM->A，RI->B

A+B->AR，A+B->A

MEM->RD

关键寄存器变化：[(RI)+00H]->R3

1. ADD R1,R3

R1->A

R3->B

A+B->R1

关键寄存器变化：(R1)+(R3)->R1

1. INC RI

RI->A

(A)+1->RI

关键寄存器变化：(RI)+1->RI

1. SUB R0,R3

R0->A

R3->B

A-B->R0

关键寄存器变化：(R0)-(R3)->R0

1. BZC RESULT

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

(FC+FZ) \*A+(FC+FZ) \*PC->PC

关键寄存器变化：(FC+FZ) \*A+(FC+FZ) \*PC->PC

1. JMP LOOP

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

A->PC

关键寄存器变化：10H->PC

1. STA 70H,R1

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

R1->MEM

关键寄存器变化：R1->MEM(70H)

1. OUT 40H,R1

PC->AR，(PC)+1->PC

MEM->AR

R1->IO

关键寄存器变化：R1->IO(40H)

1. JMP START

PC->AR，(PC)+1->PC

MEM->A

MEM->AR

A->PC

关键寄存器变化：00H->PC

1. HLT

NOP

**时序图如下：**

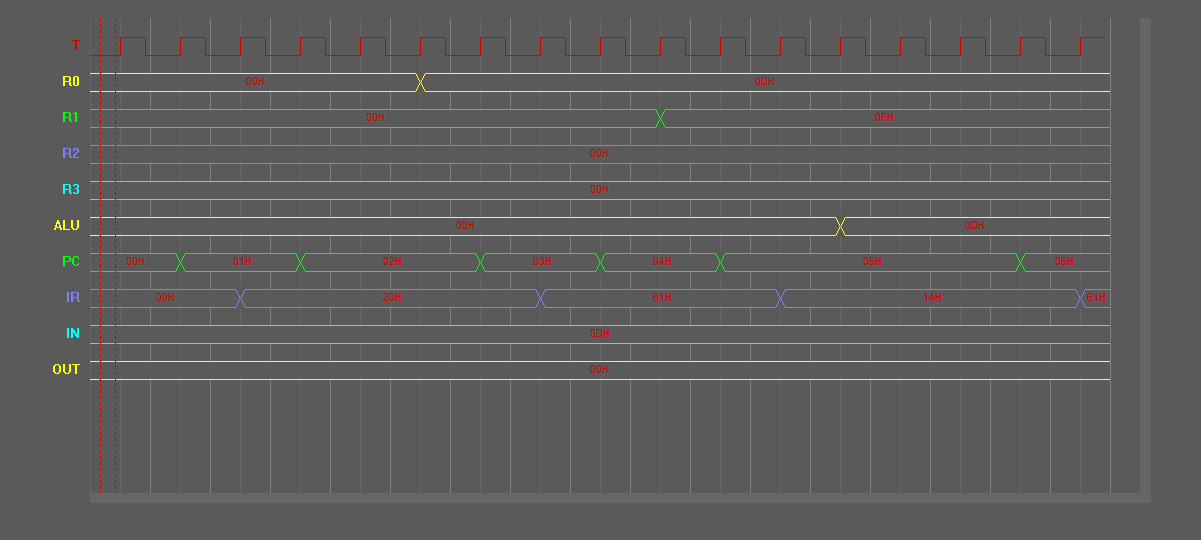


图3.2.1 读入R1

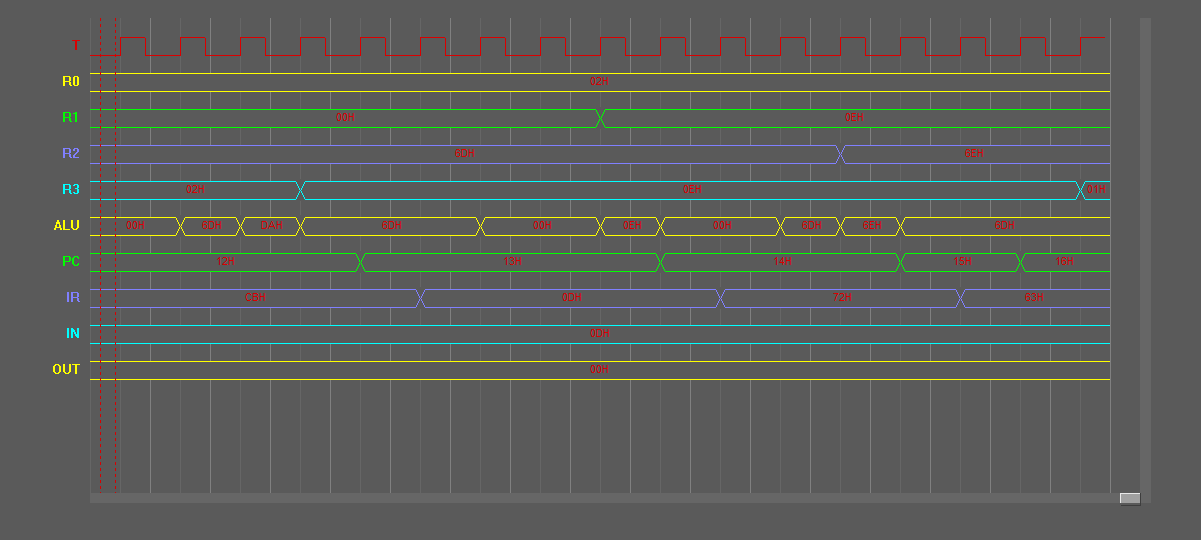


图3.2.2 得到低四位结果

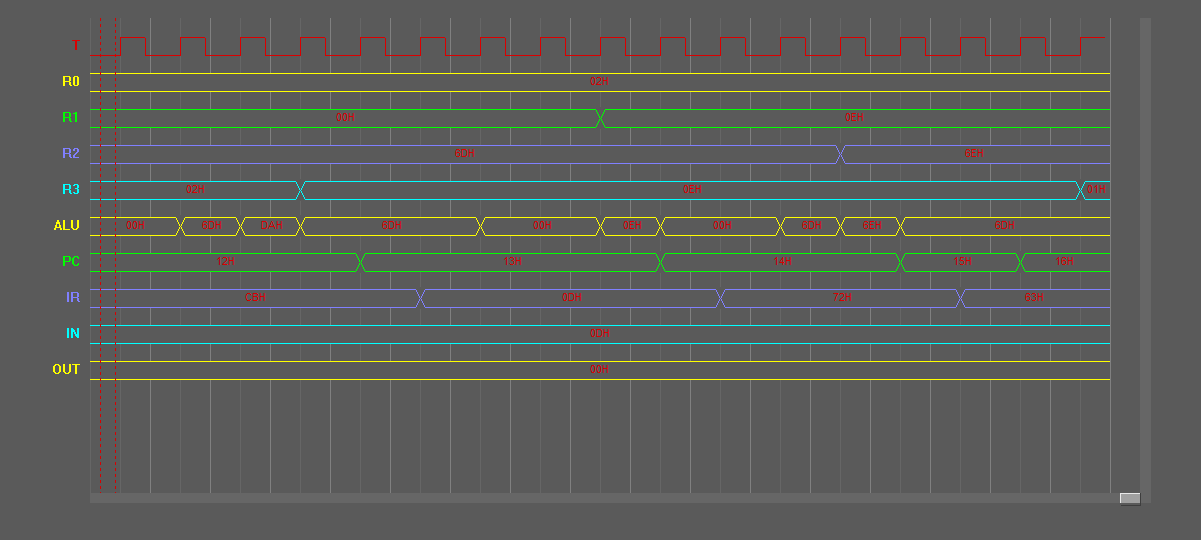


图3.2.3 得到实际循环次数存入R0

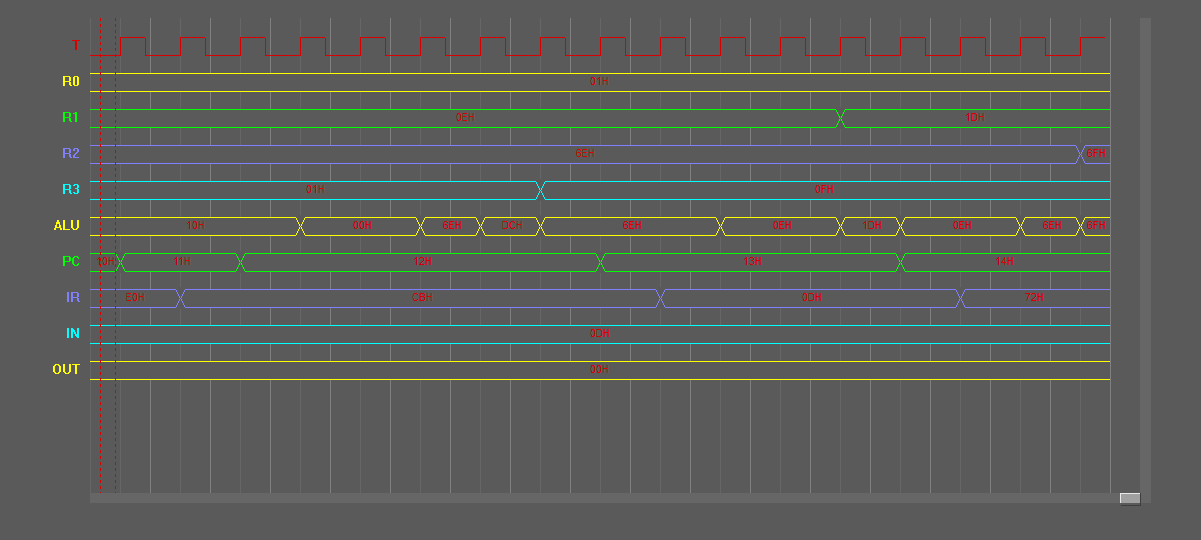


图3.2.4 和送入R1

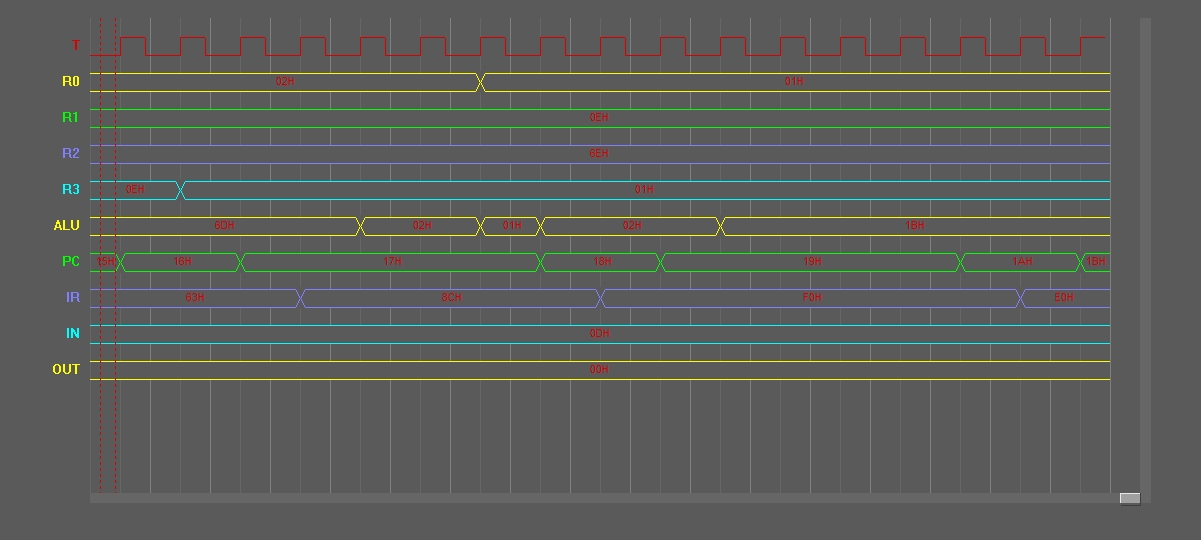


图3.2.5 循环次数（R0）减少1

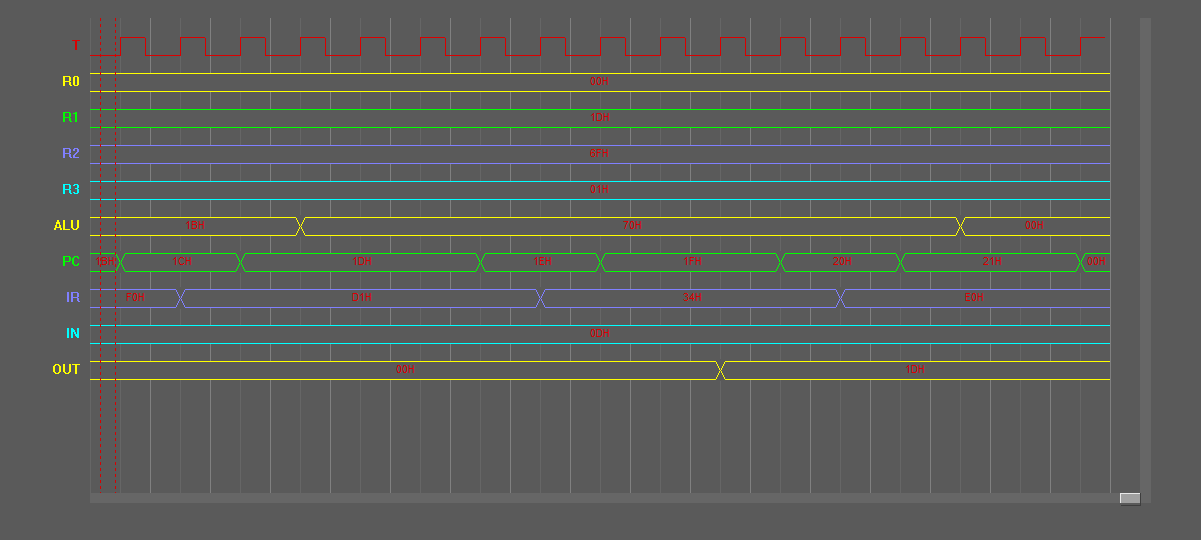


图3.2.6 累加和输出OUT

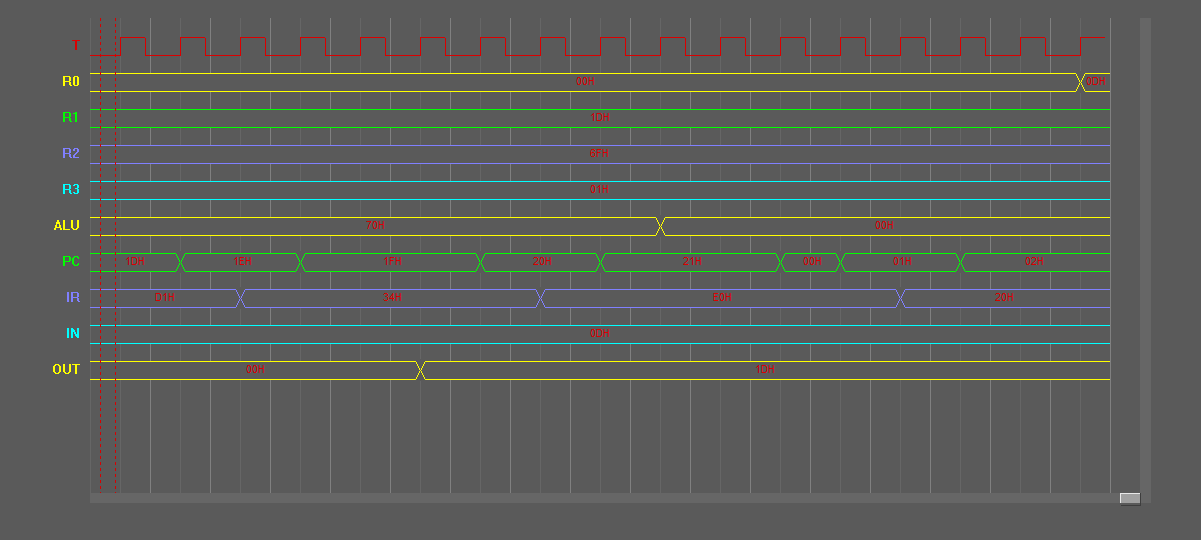


图3.2.7 返回程序开始处