



# 实验五 微程序控制器 (2)

了解微指令是如何工作的

# 复习

我们想要逐条运行4条机器指令

4条机器指令需要有4段微程序与之相对应

构成这4段微程序需要有8条微指令

这8条微指令需要以位为单位进行确定

- 存放的位置
- 微指令的控制功能
- 微指令的后继地址

助记符

机器指令码

说明

IN

(0010 0000)

IN LDA

ADD

(

OUT

(

HLT

(

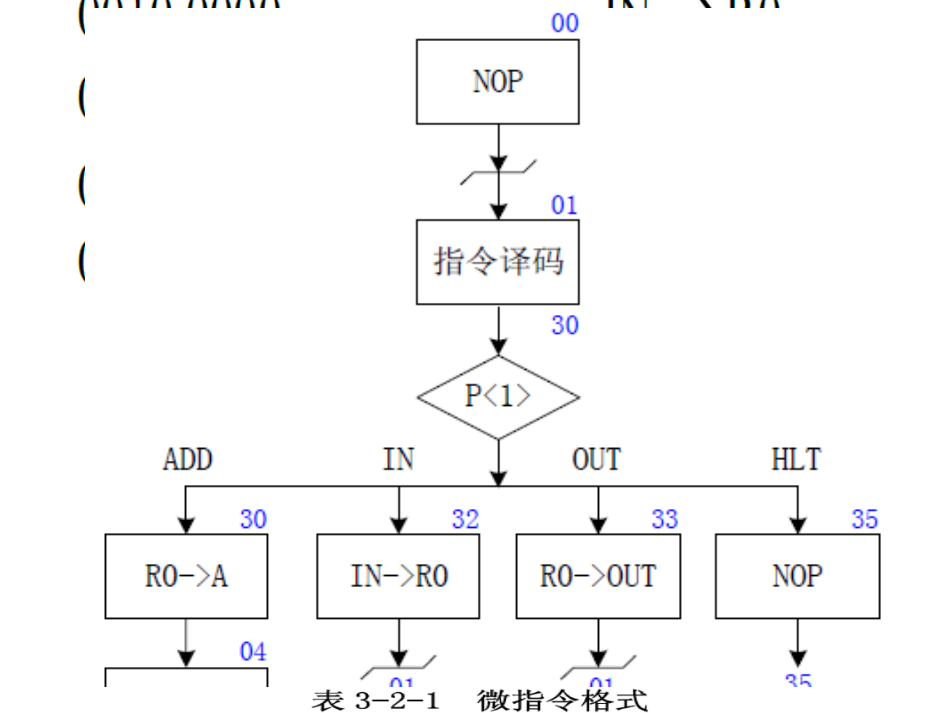


表 3-2-1 微指令格式

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

A字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDRO
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	RO_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

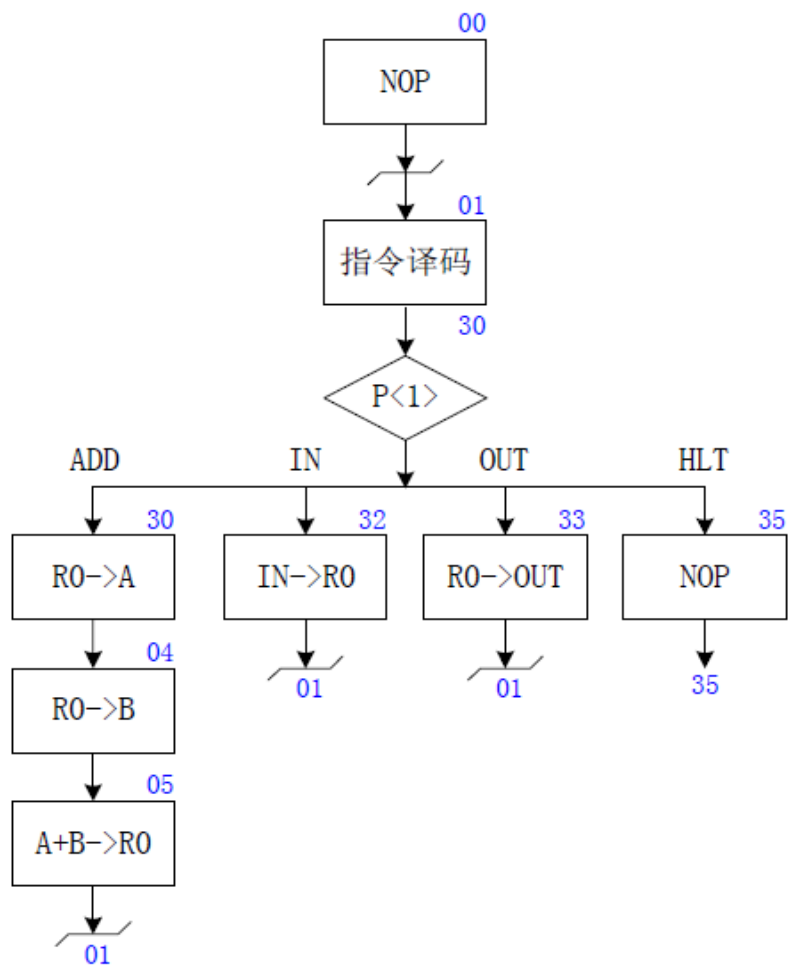


图 3-2-9 微程序流程图

表 3-2-1 微指令格式

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

A字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDRO
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	RO_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

# 微地址以16进制表示

# NOP

00 0000

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

0 0 0 0 0 0000 000 000 000 000001

A字段

B字段

C字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDR0
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

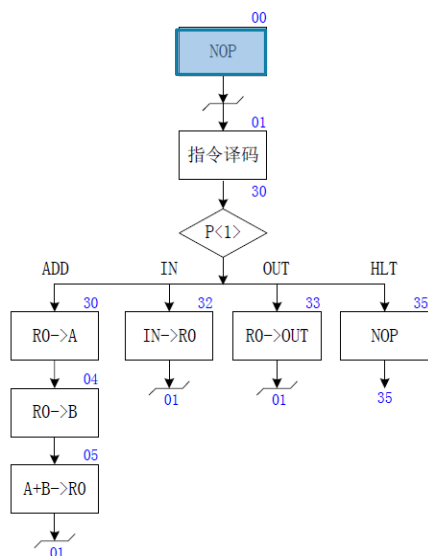


图 3-2-9 微程序流程图

# 指令译码

00 0001

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

0 0 0 0 0 0000 111 000 001 110000

A字段

B字段

C字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDR0
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

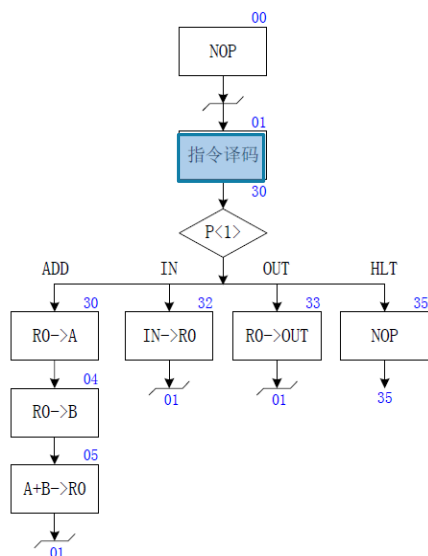


图 3-2-9 微程序流程图

# 指令译码

当 $I_6=I_7=1$ 时， $I_7$ 、 $I_6$ 、 $I_3$ 、 $I_2$ 修改微地址最后四位  
否则， $I_7$ 、 $I_6$ 、 $I_5$ 、 $I_4$ 修改微地址最后四位

P<1>测试前的后继微地址	指令码（助记符）	P<1>测试后的后继微地址
11 0000	00 00 0000 (ADD)	11 0000
11 0000	00 0 0000 (IN)	11 0010
11 0000	00 1 0000 (OUT)	11 0011
11 0000	01 01 0000 (HLT)	11 0101

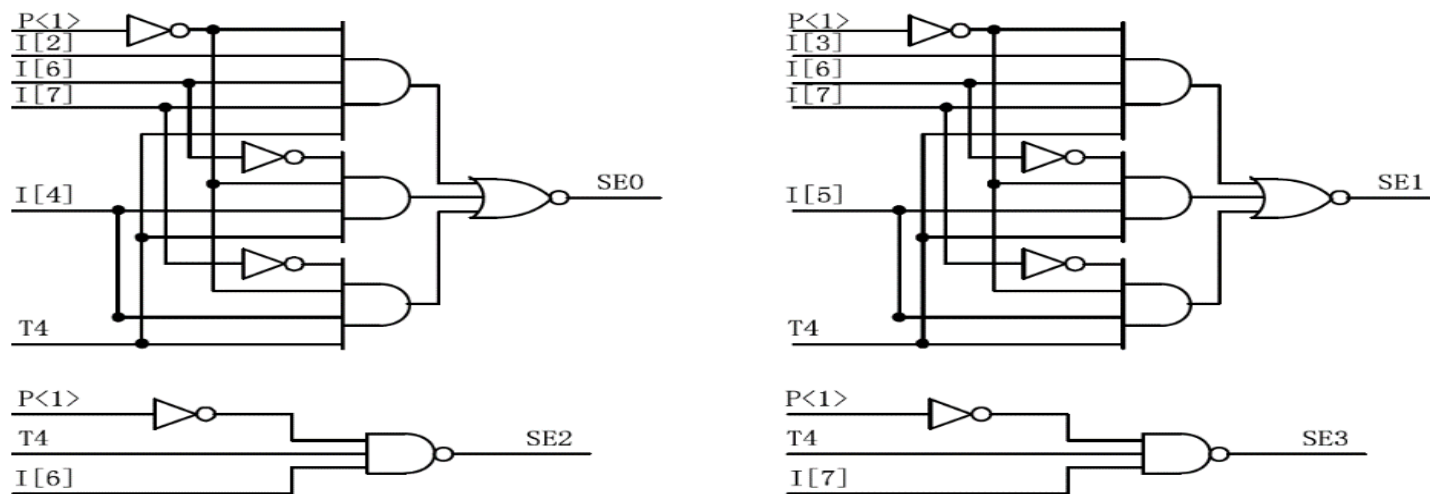


图 3-2-3 指令译码原理图

11 0000

$R0 \rightarrow A$

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

0 0 0 0 0 0000 001 010 000 000100

A字段

B字段

C字段

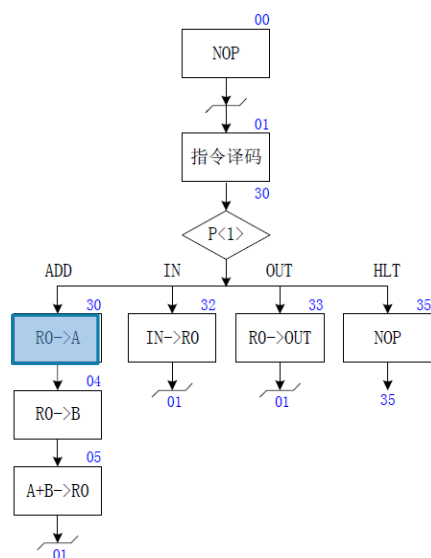


图 3-2-9 微程序流程图

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDR0
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留



**R0 → B**

00 0100

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

0 0 0 0 0 0000 010 010 000 000101

A字段

B字段

C字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDR0
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

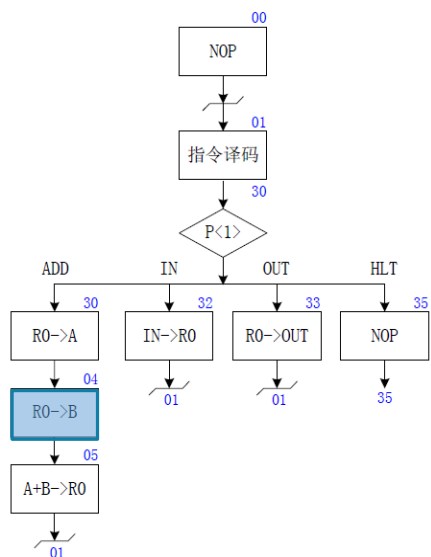


图 3-2-9 微程序流程图



$A + B \rightarrow R0$

00 0101

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

A字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDRO
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

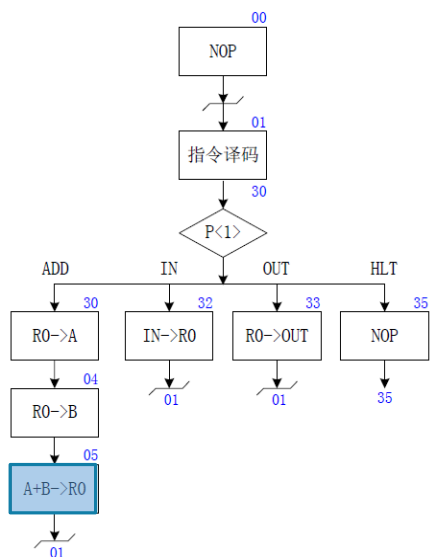


图 3-2-9 微程序流程图

表 1-1-1 运算器逻辑功能表

运算类型	S3 S2 S1 S0	CN	功 能
逻辑运算	0000	X	F=A (直通)
	0001	X	F=B (直通)
	0010	X	F=AB (FZ)
	0011	X	F=A+B (FZ)
	0100	X	F=/A (FZ)
移位运算	0101	X	F=A 不带进位循环右移 B (取低 3 位) 位 (FZ)
	0110	0	F=A 逻辑右移一位 (FZ)
		1	F=A 带进位循环右移一位 (FC, FZ)
	0111	0	F=A 逻辑左移一位 (FZ)
		1	F=A 带进位循环左移一位 (FC, FZ)
算术运算	1000	X	置 FC=CN (FC)
	1001	X	F=A 加 B (FC, FZ)
	1010	X	F=A 加 B 加 FC (FC, FZ)
	1011	X	F=A 减 B (FC, FZ)
	1100	X	F=A 减 1 (FC, FZ)
	1101	X	F=A 加 1 (FC, FZ)
	1110	X	(保留)
	1111	X	(保留)

算术加

11 0010

# IN → R0

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

A字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDRO
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

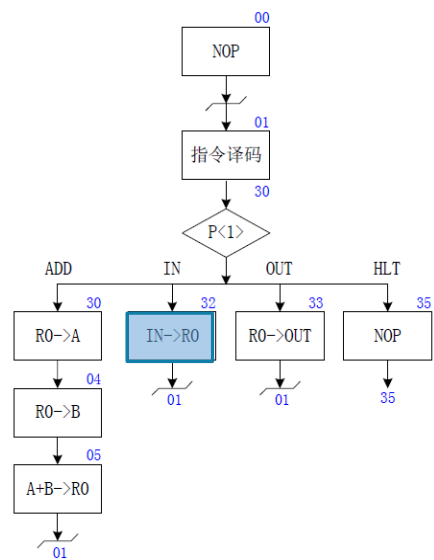


图 3-2-9 微程序流程图

# R0→OUT

11 0011

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

A字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDR0
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

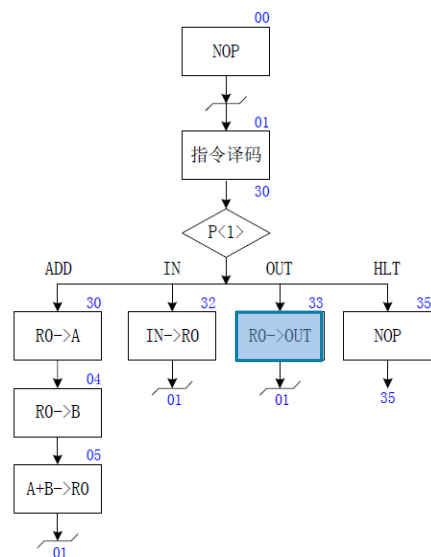


图 3-2-9 微程序流程图

# HLT

11 0101

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

## A字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDR0
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	LDIR

## B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

## C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	保留

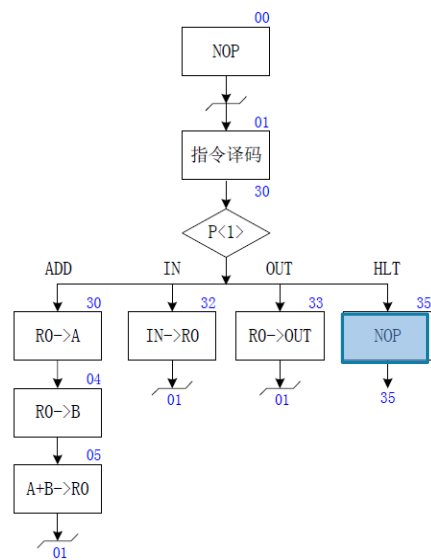


图 3-2-9 微程序流程图

# 目标做一点小小的改动

我们想要运行5条机器指令

助记符	机器指令码	说明
IN	0010 0000	IN→R0
STA	???? ????	R0→00
ADD	0000 0000	R0+[00]→R0 多加几次?
OUT	0011 0000	R0→OUT
HLT	0101 0000	停机

# 相应需要做哪些改变？

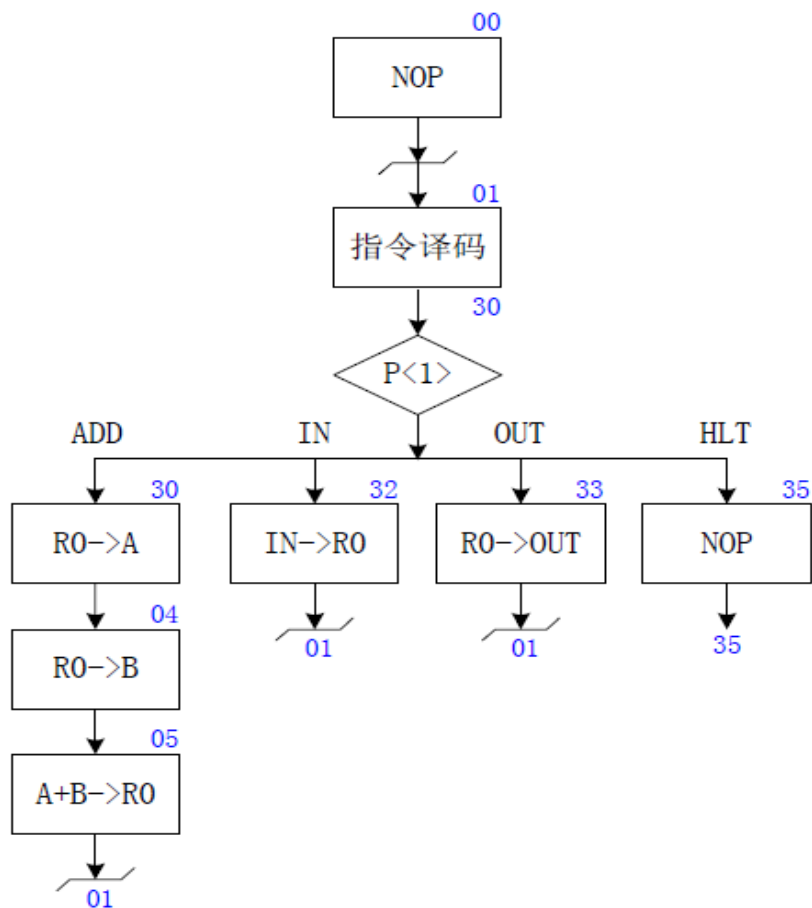
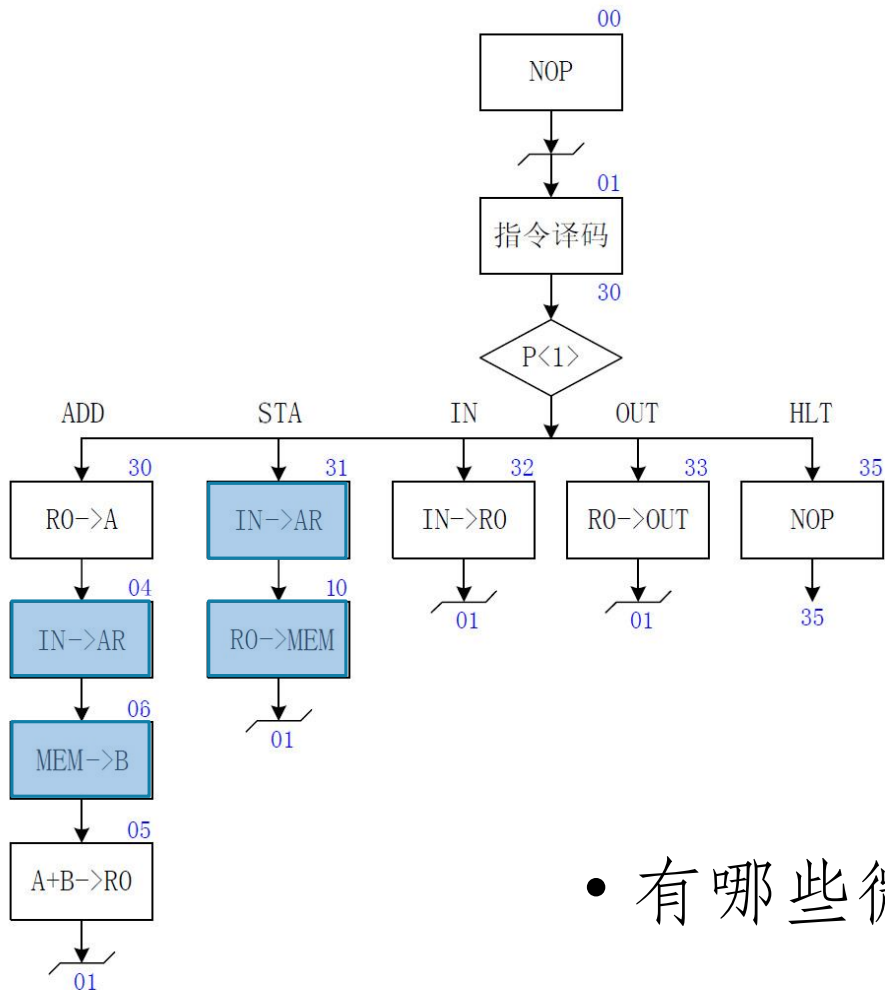


图 3-2-9 微程序流程图

- 如何增加一条机器指令
  - 增加一道微程序
  - 设计相应的机器码
- 如何修改一条机器指令
  - 更改其微程序中的微指令
    - 条数
    - 控制码
    - 地址



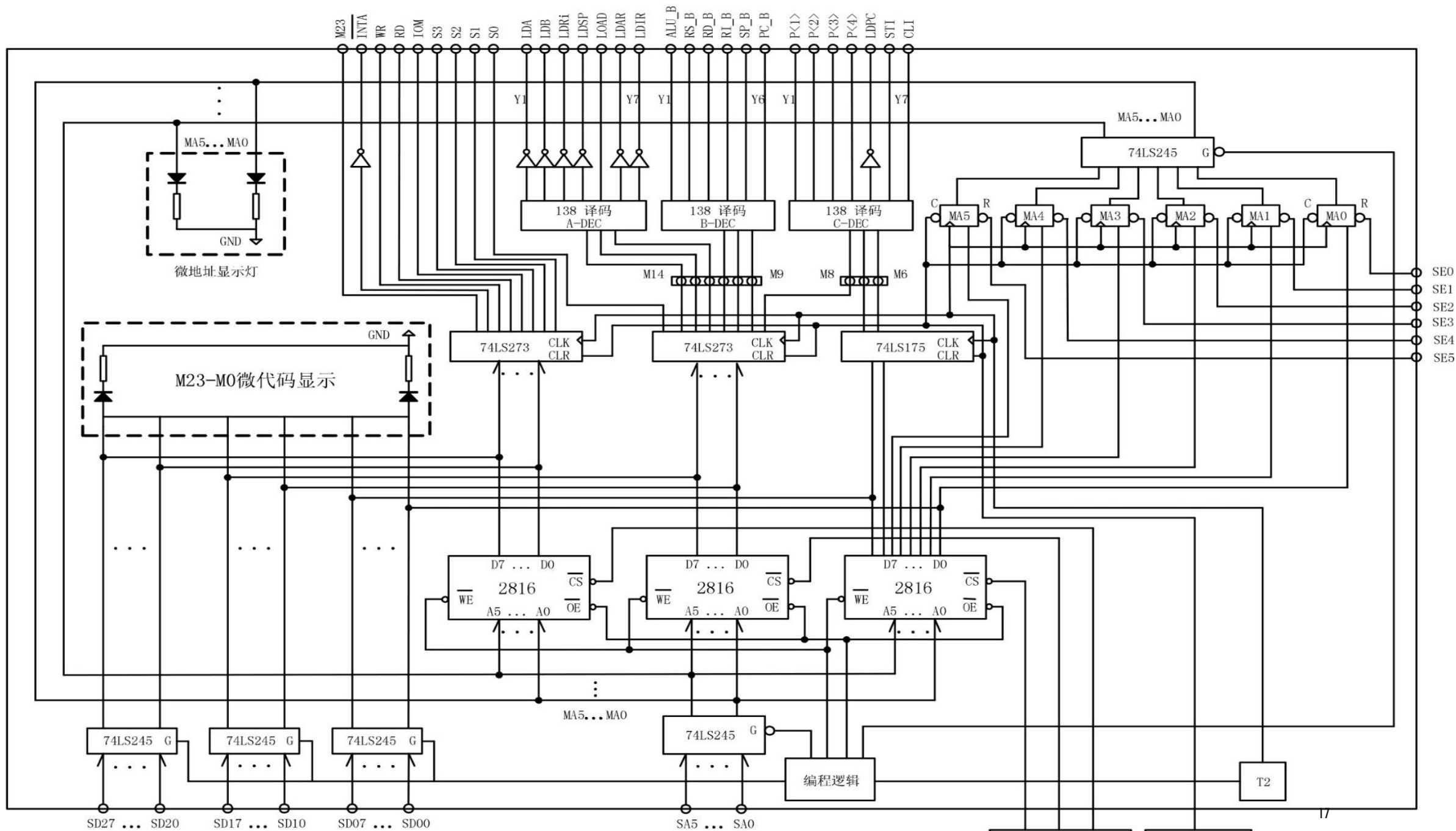
# 修改后的微程序流程图

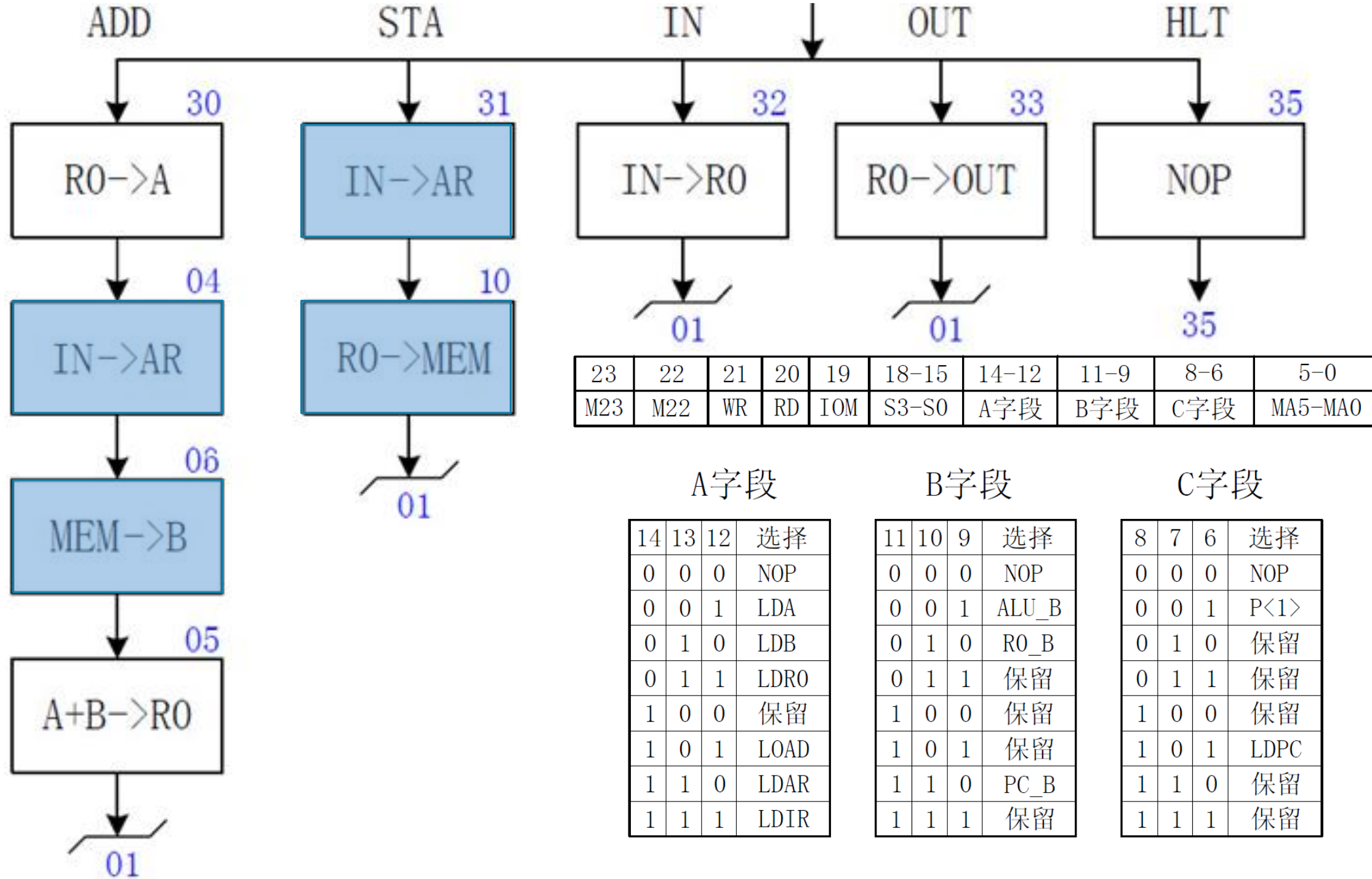


**STA 的机器码确定为 00010000**

- 有哪些微指令是新增或需要修改的呢？

图 3-2-2 微程序控制器原理图





# IN→AR

00 0100

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

## A字段

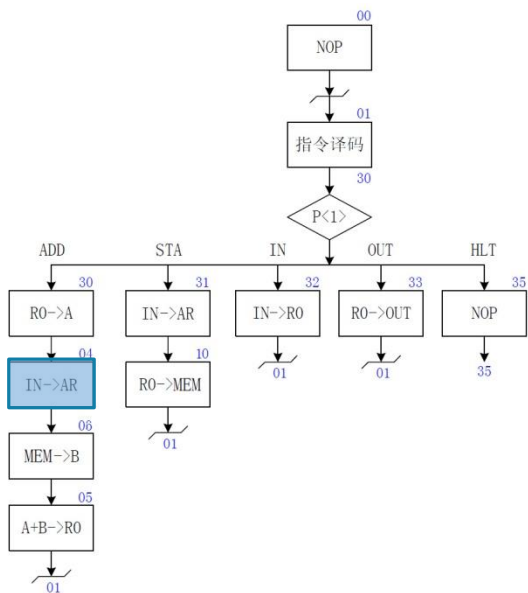
14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDRO
1	0	0	保留
1	0	1	LOAD
1	1	0	LDAR
1	1	1	LDIR

## B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	RO_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	PC_B
1	1	1	保留

## C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	LDPC
1	1	0	保留
1	1	1	保留



# MEM → B

## 00 01 10

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

### A字段

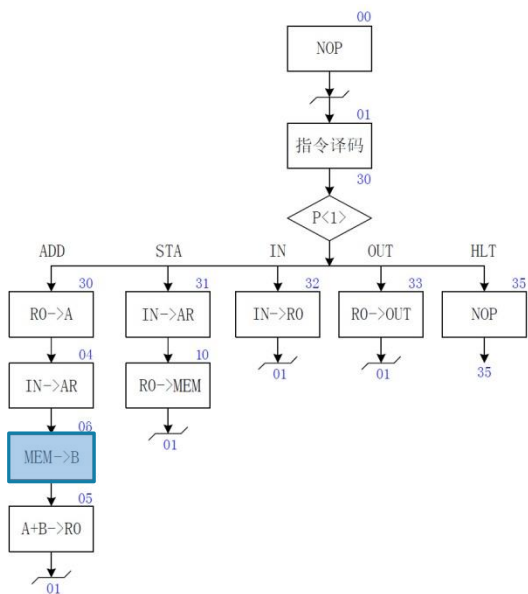
14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDR0
1	0	0	保留
1	0	1	LOAD
1	1	0	LDAR
1	1	1	LDIR

### B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	RO_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	PC_B
1	1	1	保留

### C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	LDPC
1	1	0	保留
1	1	1	保留



# IN→AR

11 0001

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

## A字段

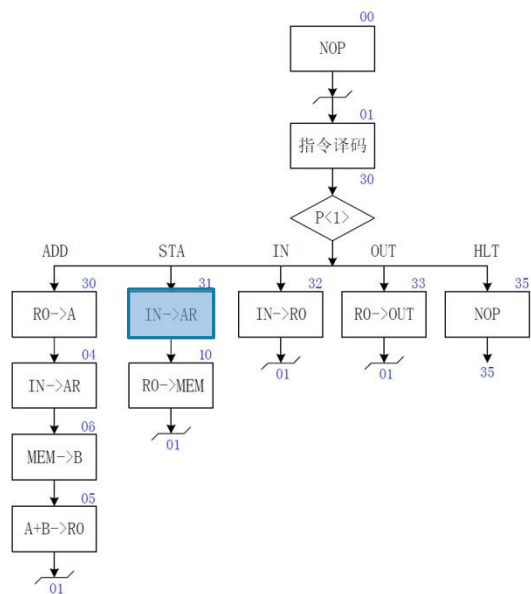
14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDRO
1	0	0	保留
1	0	1	LOAD
1	1	0	LDAR
1	1	1	LDIR

## B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	PC_B
1	1	1	保留

## C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	LDPC
1	1	0	保留
1	1	1	保留



# R0 → MEM

01 0000

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

## A字段

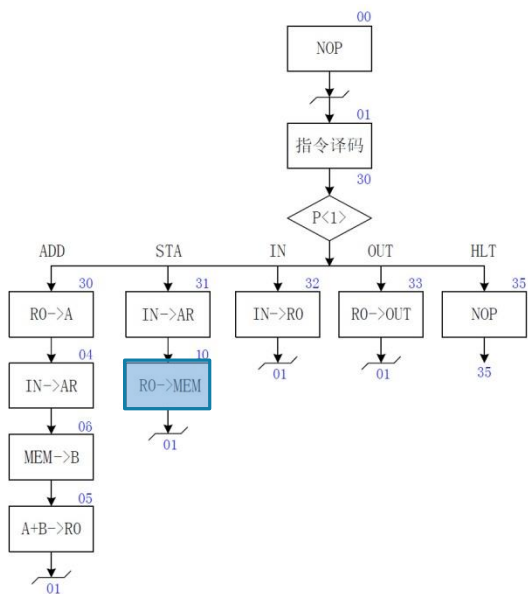
14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDR0
1	0	0	保留
1	0	1	LOAD
1	1	0	LDAR
1	1	1	LDIR

## B字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	R0_B
0	1	1	保留
1	0	0	保留
1	0	1	保留
1	1	0	PC_B
1	1	1	保留

## C字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	保留
0	1	1	保留
1	0	0	保留
1	0	1	LDPC
1	1	0	保留
1	1	1	保留

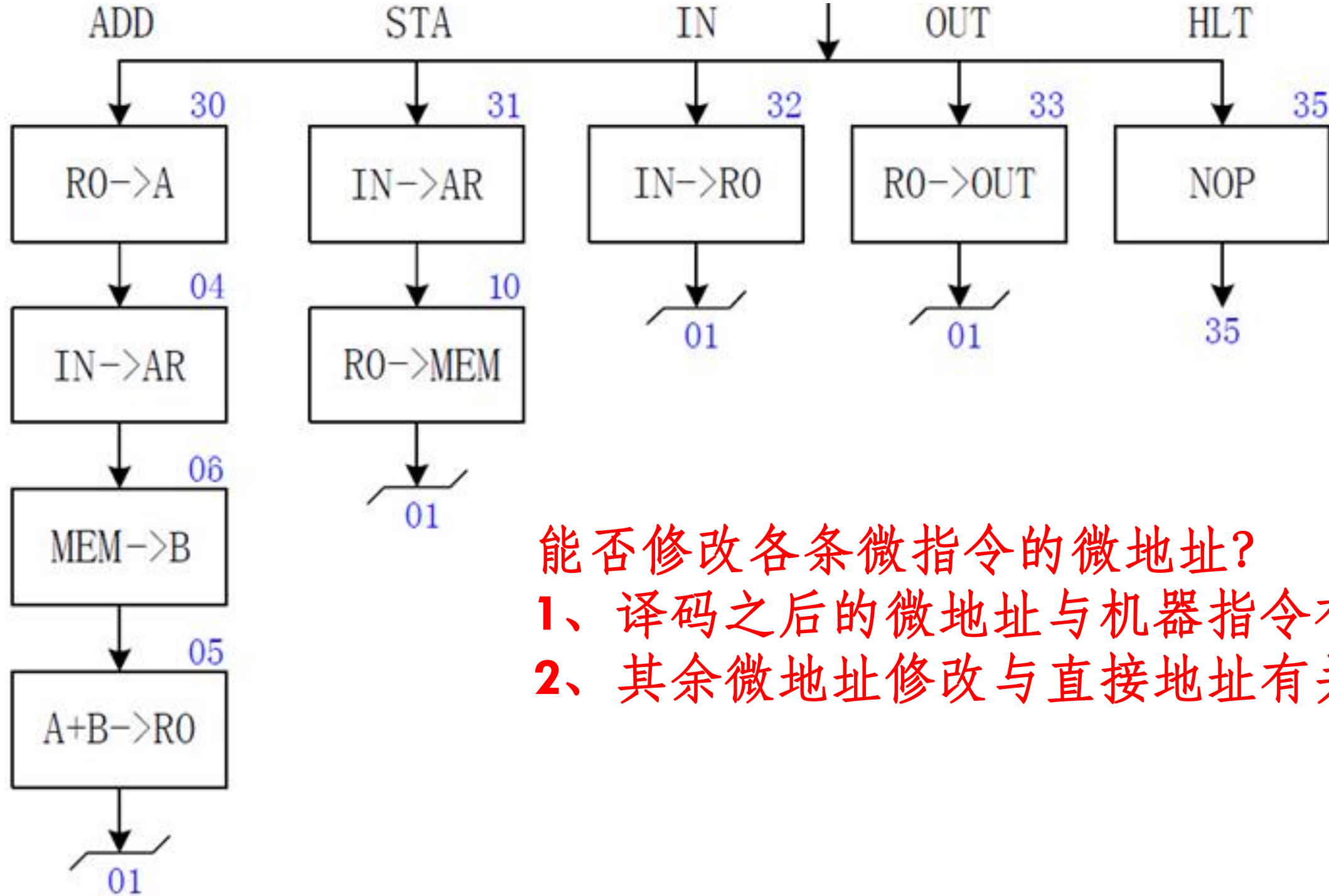




# 思考

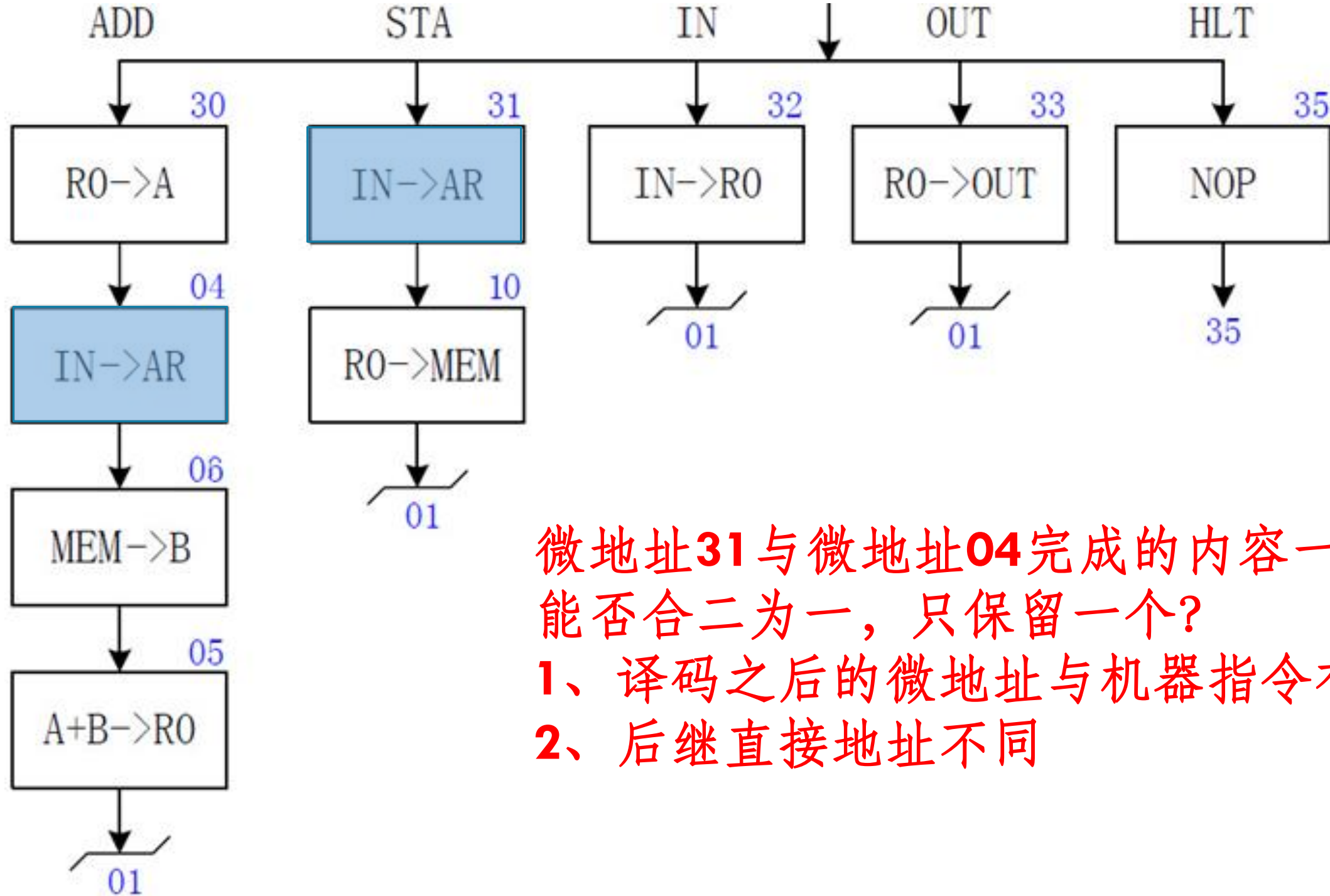
1、**NOP**微指令为空操作，有什么作用？能否删除？

每次开始运行前按**CLR**键，将微地址清零  
从**00**开始，起到一个承上启下的作用



能否修改各条微指令的微地址?

- 1、译码之后的微地址与机器指令有关
- 2、其余微地址修改与直接地址有关



微地址**31**与微地址**04**完成的内容一样，能否合二为一，只保留一个？

- 1、译码之后的微地址与机器指令有关
- 2、后继直接地址不同