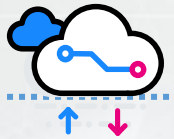


微指令格式与微程序存储

2022.11

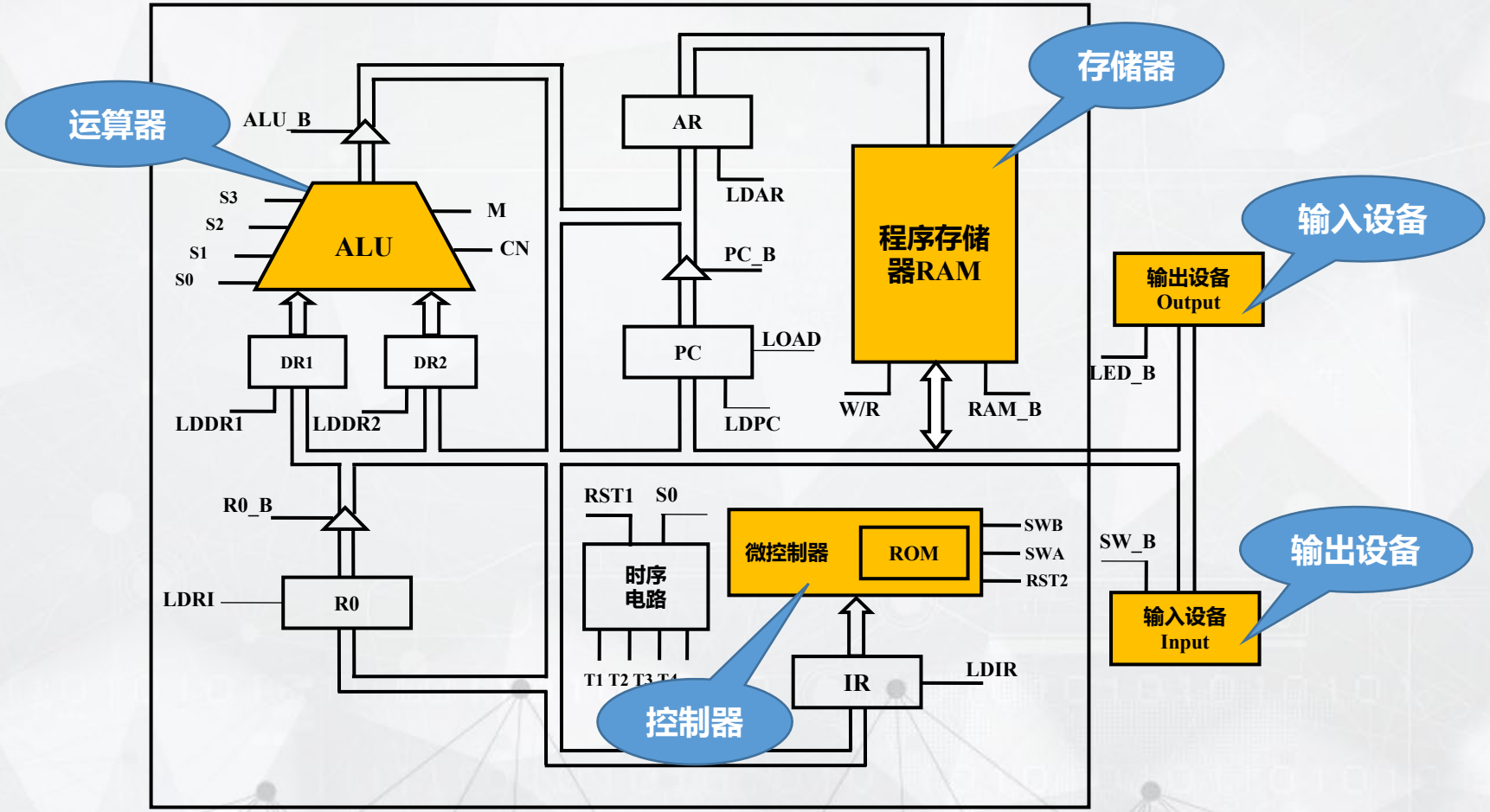
哈尔滨工程大学计算机实验教学中心

基本模型机指令与控制台命令

助记符	机器指令	说明
IN	00000000	“Input Device” →R0
ADD Addr	00010000 XXXXXXXX	R0+[Addr]→R0
STA Addr	00100000 XXXXXXXX	R0→[Addr]
OUT Addr	00110000 XXXXXXXX	[Addr]→ “Output Divice”
JMP Addr	01000000 XXXXXXXX	Addr→PC

SWB	SWA	控制台命令
0	0	读内存（KRD）
0	1	写内存（KWE）
1	1	启动程序（RP）

基本模型机数据通路


运算器
ALU_B
S3
S2
S1
S0
ALU
M
CN
DR1
DR2
LDDR1
LDDR2
R0_B
LDRI
R0
AR
LDAR
PC_B
PC
LOAD
LDPC
存储器
程序存储器RAM
W/R
RAM_B
输入设备
输出设备
Output
LED_B
RST1
S0
时序电路
T1 T2 T3 T4
微控制器
ROM
SWB
SWA
RST2
SW_B
输出设备
输入设备
Input
IR
LDIR
控制器

微命令（控制信号）整理

- ✓ALU：S3、S2、S1、S0、M、Cn
- ✓RAM：WE
- ✓寄存器：
  - 数据寄存器：LDRI
  - 指令寄存器：LDIR
  - 地址寄存器：LDAR
- ✓程序计数器：LDPC、LOAD、PC_B
- ✓总线控制：SW_B、RAM_B、LED_B、ALU_B、R0_B、R1_B、R2_B
  微程序顺序控制：P（1）、P（2）、P（3）、P（4）
- ✓时序电路：S、RST

微命令（控制信号）整理

- ✓ALU：S3、S2、S1、S0、M、Cn
- ✓RAM：WE
- ✓寄存器：
  - 数据寄存器：LDRI
  - 指令寄存器：LDIR
  - 地址寄存器：LDAR
- ✓程序计数器：LDPC、LOAD、PC_B
- ✓总线控制：SW_B、RAM_B、LED_B、ALU_B、R0_B、R1_B、R2_B
  微程序顺序控制：P（1）、P（2）、P（3）、P（4）
- ✓时序电路：S、RST

微命令（控制信号）整理

- ✓ALU：S3、S2、S1、S0、M、Cn
- ✓RAM：WE
- ✓寄存器：
  - 数据寄存器：LDRI
  - 指令寄存器：LDIR
  - 地址寄存器：LDAR
- ✓程序计数器：LDPC、LOAD、PC_B
- ✓总线控制：SW_B、RAM_B、LED_B、ALU_B、R0_B、R1_B、R2_B
  微程序顺序控制：P（1）、P（2）、P（3）、P（4）
- ✓时序电路：S、RST

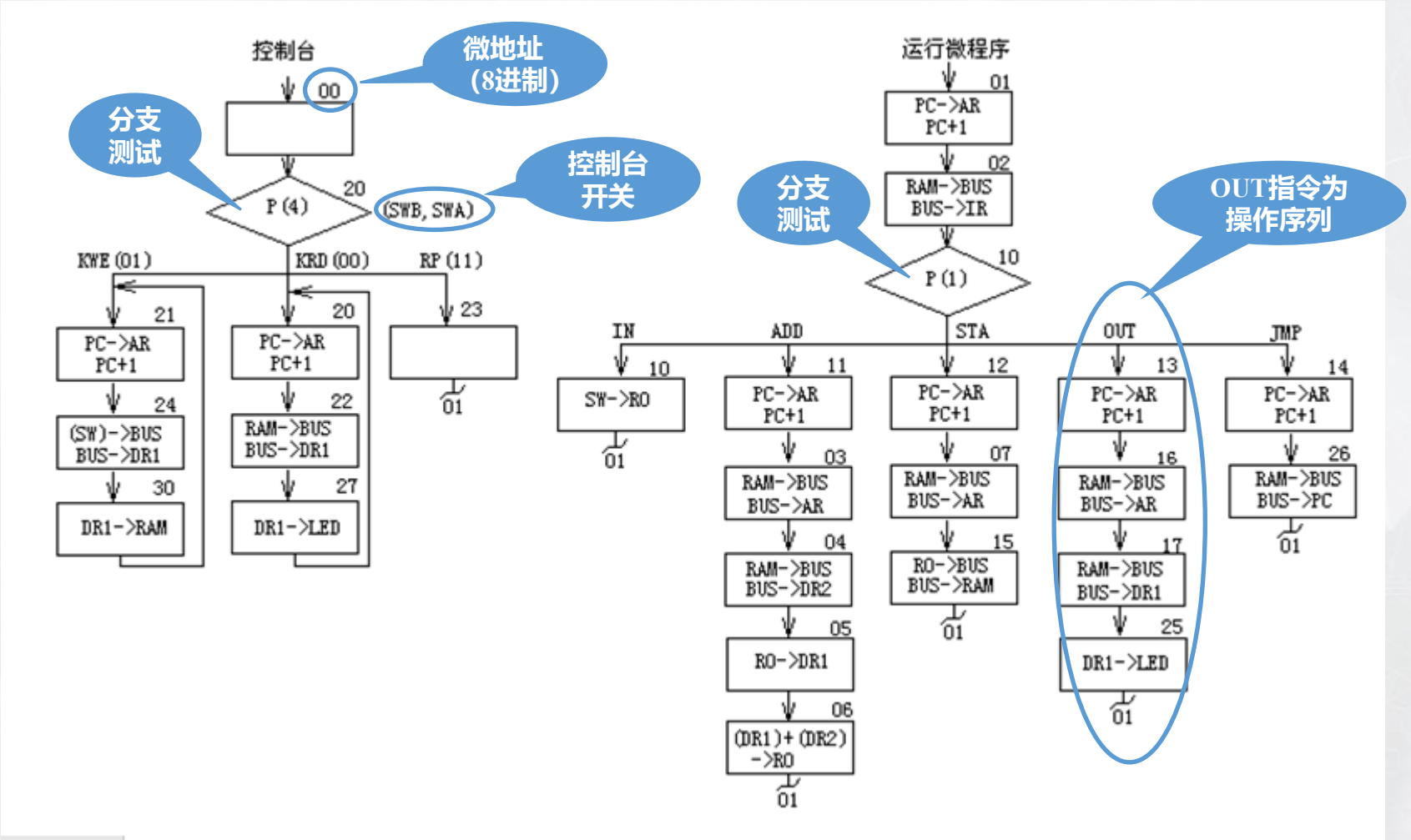
微命令（控制信号）整理

- ✓ALU：S3、S2、S1、S0、M、Cn
- ✓RAM：WE
- ✓寄存器：
  - 数据寄存器：LDRI
  - 指令寄存器：LDIR
  - 地址寄存器：LDAR
- ✓程序计数器：LDPC、LOAD、PC_B
- ✓总线控制：SW_B、RAM_B、LED_B、ALU_B、R0_B、R1_B、R2_B
  微程序顺序控制：P（1）、P（2）、P（3）、P（4）
- ✓时序电路：S、RST

微程序控制器原理

一条机器指令往往分成几步执行，将每一步操作所需的若干微命令（控制信号）以代码形式编写在一条微指令中，若干条微指令组成一段微程序，对应一条机器指令。

基本模型微程序流程


控制台
00
微地址
(8进制)
分支
测试
P(4)
20
(SWB, SWA)
控制台
开关
KWE(01)
KRD(00)
RP(11)
21
PC->AR
PC+1
24
(SW)->BUS
BUS->DR1
30
DR1->RAM
20
PC->AR
PC+1
22
RAM->BUS
BUS->DR1
27
DR1->LED
23
01
运行微程序
01
PC->AR
PC+1
02
RAM->BUS
BUS->IR
分支
测试
P(1)
10
OUT指令为
操作序列
IN
ADD
STA
OUT
JMP
10
SW->RO
01
11
PC->AR
PC+1
03
RAM->BUS
BUS->AR
04
RAM->BUS
BUS->DR2
05
RO->DR1
06
(DR1)+(DR2)
->RO
01
12
PC->AR
PC+1
07
RAM->BUS
BUS->AR
15
RO->BUS
BUS->RAM
01
13
PC->AR
PC+1
16
RAM->BUS
BUS->AR
17
RAM->BUS
BUS->DR1
25
DR1->LED
01
14
PC->AR
PC+1
26
RAM->BUS
BUS->PC
01

基本模型机微代码定义

微命令编码格式定义

24	23	22	21	20	19	18	17	16	15 14 13	12 11 10	9 8 7	6	5	4	3	2	1
S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	UA5	UA4	UA3	UA2	UA1	UA0

微命令控制信号的功能

A9、A8字段			A字段				B字段				C字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	

基本模型机控制信号的存储

微地址 (8 进制)	S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	UA5——UA0	微指令 (16 进制)
0 0	0	0	0	0	0	0	0	1	1	000	000	100	0 1 0 0 0 0	018110
0 1	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 0 1 0	01ED82
0 2	0	0	0	0	0	0	0	0	1	100	000	001	0 0 1 0 0 0	00C048
0 3	0	0	0	0	0	0	0	0	1	110	000	000	0 0 0 1 0 0	00E004
0 4	0	0	0	0	0	0	0	0	1	011	000	000	0 0 0 1 0 1	00B005
0 5	0	0	0	0	0	0	0	1	1	010	001	000	0 0 0 1 1 0	01A206
0 6	1	0	0	1	0	1	0	1	1	001	101	000	0 0 0 0 0 1	959A01
0 7	0	0	0	0	0	0	0	0	1	110	000	000	0 0 1 1 0 1	00E00D
1 0	0	0	0	0	0	0	0	0	0	001	000	000	0 0 0 0 0 1	001001
1 1	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 0 1 1	01ED83
1 2	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 1 1 1	01ED87
1 3	0	0	0	0	0	0	0	1	1	110	110	110	0 0 1 1 1 0	01ED8E
1 4	0	0	0	0	0	0	0	1	1	110	110	110	0 1 0 1 1 0	01ED96
1 5	0	0	0	0	0	0	1	1	1	000	001	000	0 0 0 0 0 1	038201
1 6	0	0	0	0	0	0	0	0	1	110	000	000	0 0 1 1 1 1	00E00F
1 7	0	0	0	0	0	0	0	0	1	010	000	000	0 1 0 1 0 1	00A015
2 0	0	0	0	0	0	0	0	1	1	110	110	110	0 1 0 0 1 0	01ED92
2 1	0	0	0	0	0	0	0	1	1	110	110	110	0 1 0 1 0 0	01ED94
2 2	0	0	0	0	0	0	0	0	1	010	000	000	0 1 0 1 1 1	00A017
2 3	0	0	0	0	0	0	0	1	1	000	000	000	0 0 0 0 0 1	018001
2 4	0	0	0	0	0	0	0	0	0	010	000	000	0 1 1 0 0 0	002018
2 5	0	0	0	0	0	1	0	1	0	000	101	000	0 0 0 0 0 1	050A01
2 6	0	0	0	0	0	0	0	0	1	101	000	110	0 0 0 0 0 1	00D181
2 7	0	0	0	0	0	1	0	1	0	000	101	000	0 1 0 0 0 0	050A10
3 0	0	0	0	0	0	1	1	1	1	000	101	000	0 1 0 0 0 1	068A11

先确定每段微程序的第一条微代码的微地址（ROM单元地址）。然后每段微程序中，当前微指令的低六位指向下一个微指令在ROM中存放的地址，即下一个微地址。

Addr	+0	+1	+2
000	00000001100000010010000	00000001111101101100000010	00000000110000000001001000
003	00000000111000000000100	00000000101100000000000101	00000001101000010000000110
006	10010101100110100000001	00000000111000000000011101	00000000000010000000000001
011	00000001111101101100000011	00000001111101101100000111	00000001111101101100011110
014	00000001111101101100010110	00000011100000100000000001	00000000111000000000011111
017	00000000101000000000010101	00000001111101101100010010	00000001111101101100101000
022	00000000101000000000010111	00000001100000000000000001	00000000001000000000011000
025	00000101000010100000000001	00000000110100011000000001	00000101000010100000010000
030	00000110100010100001000 1	00000000000000000000000000	00000000000000000000000000
033	00000000000000000000000000	00000000000000000000000000	00000000000000000000000000

24位控制信号

微程序控制器

ROM ucode.hex

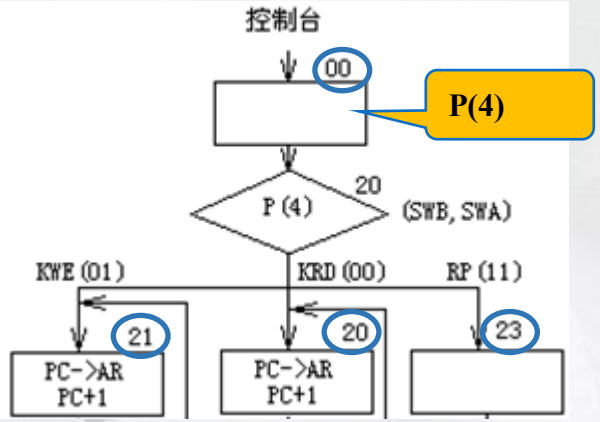
ROM初始化数据文件ucode.hex

控制台命令分支选择微代码存储与分析(P4分支测试)

微地址(8 进制)	S3 S2 S1 S0	M Cn	WE A9 A8	A	B	C	UA5——UA0
0 0	0 0 0 0	0 0	0 1 1	000	000	100	0 1 0 0 0 0
…	…			…	…	…	…
2 0	0 0 0 0	0 0	0 1 1	110	110	110	0 1 0 0 1 0
2 1	0 0 0 0	0 0	0 1 1	110	110	110	0 1 0 1 0 0
2 2	0 0 0 0	0 0	0 0 1	010	000	000	0 1 0 1 1 1
2 3	0 0 0 0	0 0	0 1 1	000	000	000	0 0 0 0 0 1

下一个微地址**不由低六位决定，不一定是20，而需要由P4分支测试，进行跳转，3路分支，由SWB、SWA决定。**

A9、A8 字段			A 字段				B 字段				C 字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	



控制台命令-写内存微代码存储与分析(SWB=0,SWA=1)

微地址(8进制)	S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	UA5——UA0
0 0	0	0	0	0	0	0	0	1	1	000	000	100	0 1 0 0 0 0
0 1	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 0 1 0
…	…									…	…	…	…
2 1	0	0	0	0	0	0	0	1	1	110	110	110	0 1 0 1 0 0
2 2	0	0	0	0	0	0	0	0	1	010	000	000	0 1 0 1 1 1
2 3	0	0	0	0	0	0	0	1	1	000	000	000	0 0 0 0 0 1
2 4	0	0	0	0	0	0	0	0	0	010	000	000	0 1 1 0 0 0
…	…									…	…	…	…
3 0	0	0	0	0	0	1	1	1	1	000	101	000	0 1 0 0 0 1

下一个微地址24

下一个微地址30

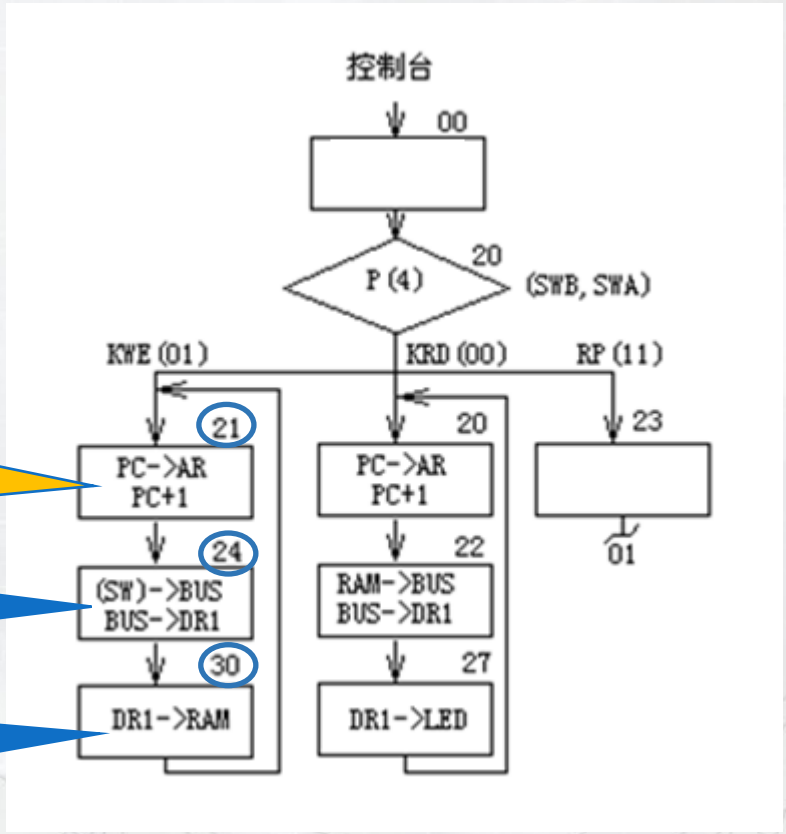
下一个微地址21

A9、A8字段			A字段				B字段				C字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	

LDPC
LDAR
PC_B

SW_B
LDDR1

SM
ALU_B
WE



控制台命令-读内存微代码存储与分析(SWB=0,SWA=0)

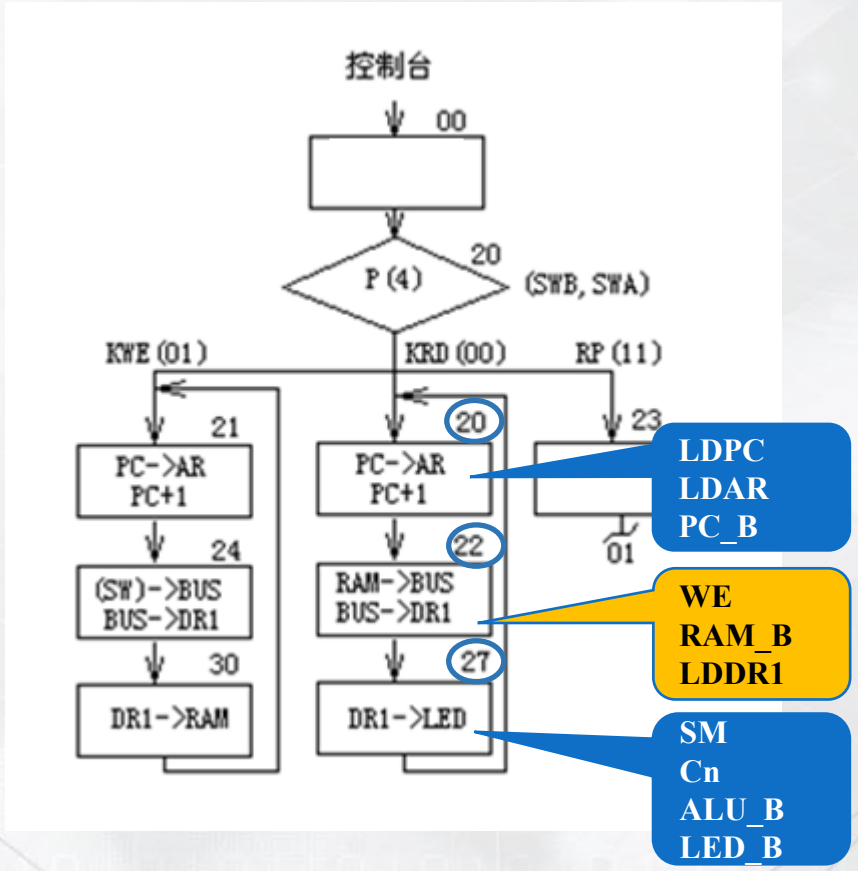
微地址(8进制)	S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	UA5——UA0
0 0	0	0	0	0	0	0	0	1	1	000	000	100	0 1 0 0 0 0
0 1	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 0 1 0
…	…									…	…	…	…
2 0	0	0	0	0	0	0	0	1	1	110	110	110	0 1 0 0 1 0
2 1	0	0	0	0	0	0	0	1	1	110	110	110	0 1 0 1 0 0
2 2	0	0	0	0	0	0	0	0	1	010	000	000	0 1 0 1 1 1
…	…									…	…	…	…
2 7	0	0	0	0	0	1	0	1	0	000	101	000	0 1 0 0 0 0

下一个微地址 22

下一个微地址 27

下一个微地址 20

A9、A8字段			A字段				B字段				C字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	

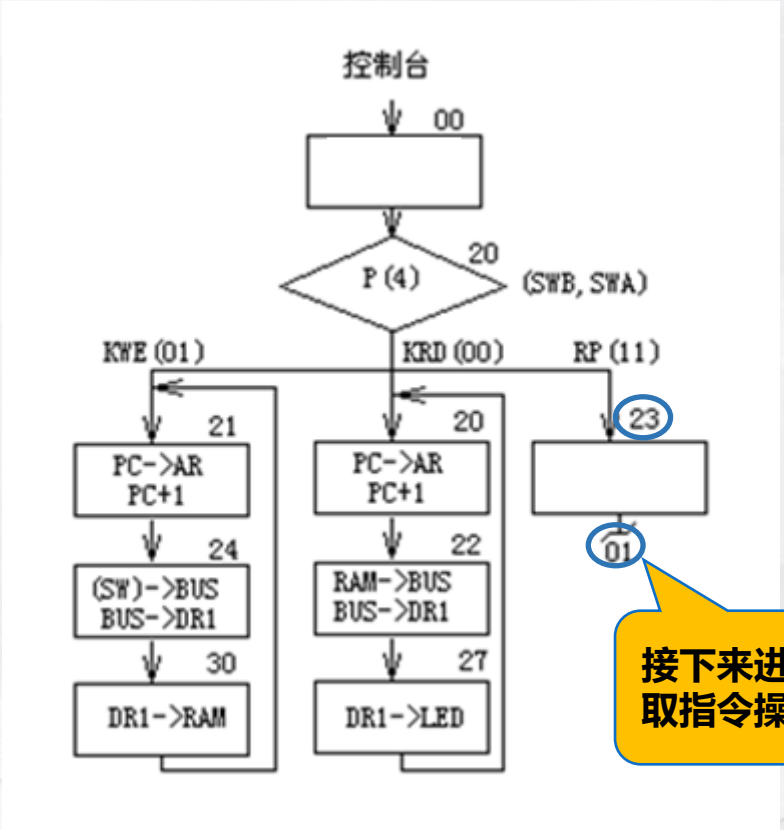


控制台命令-执行程序微代码存储与分析(SWB=1,SWA=1)

微地址(8进制)	S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	UA5——UA0
0 0	0	0	0	0	0	0	0	1	1	000	000	100	0 1 0 0 0 0
0 1	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 0 1 0
…	…									…	…	…	…
2 3	0	0	0	0	0	0	0	1	1	000	000	000	0 0 0 0 0 1
2 4	0	0	0	0	0	0	0	0	0	010	000	000	0 1 1 0 0 0
…	…									…	…	…	…
3 0	0	0	0	0	0	1	1	1	1	000	101	000	0 1 0 0 0 1

下一个微地址01

A9、A8字段			A字段				B字段				C字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	

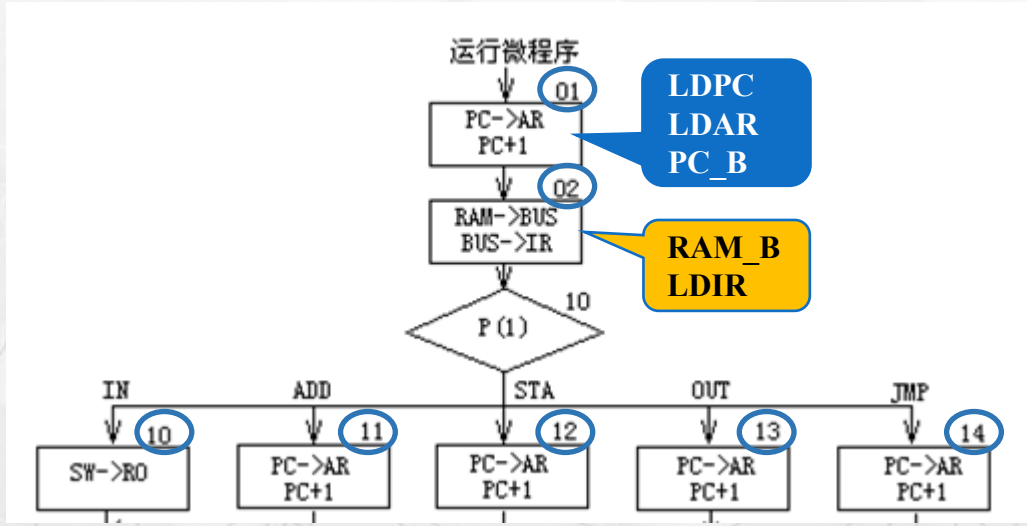


取指令微代码存储与分析(P1分支测试)

微地址(8进制)	S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	UA5——UA0
0 0	0	0	0	0	0	0	0	1	1	000	000	100	0 1 0 0 0 0
0 1	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 0 1 0
0 2	0	0	0	0	0	0	0	0	1	100	000	001	0 0 1 0 0 0
…	…									…	…	…	…
1 0	0	0	0	0	0	0	0	0	0	001	000	000	0 0 0 0 0 1
1 1	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 0 1 1
1 2	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 1 1 1
1 3	0	0	0	0	0	0	0	1	1	110	110	110	0 0 1 1 1 0
1 4	0	0	0	0	0	0	0	1	1	110	110	110	0 1 0 1 1 0

下一个微地址**不由低六位决定**，**不一定是10**，而需要由P1分支测试，进行跳转，**5路分支，由指令的操作码决定。**

A9、A8字段			A字段				B字段				C字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	

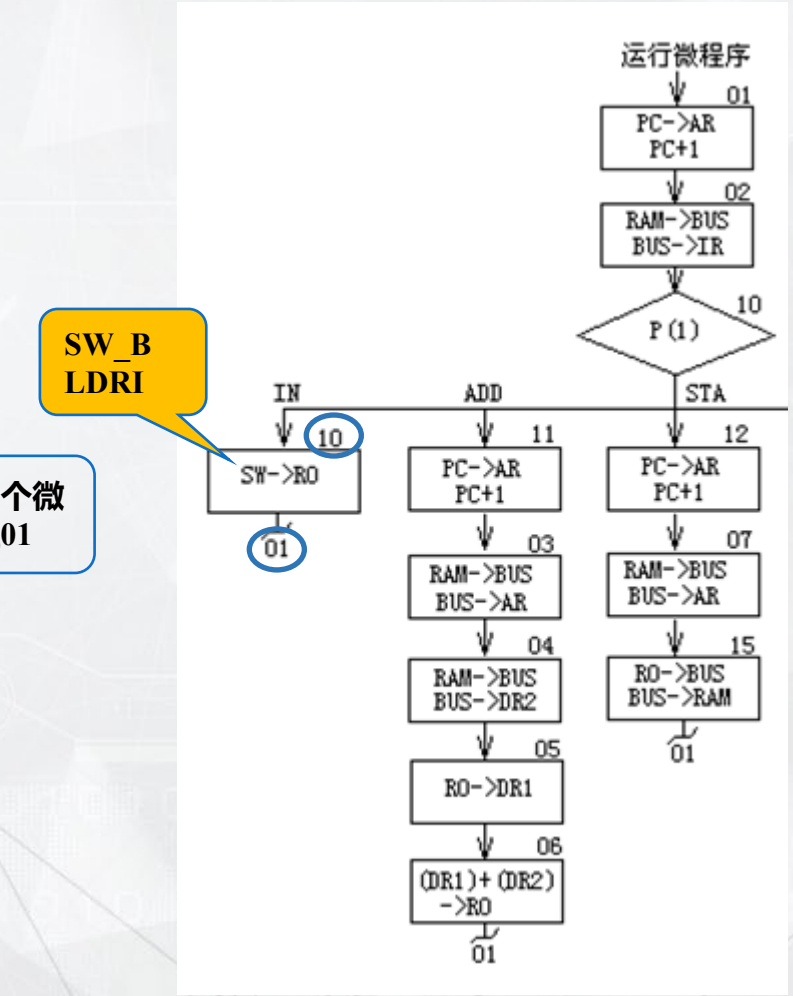


IN指令微代码存储与分析（“Input Device” →R0 ）

微地址 (8 进制)	S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	UA5——UA0
0 0	0	0	0	0	0	0	0	1	1	000	000	100	0 1 0 0 0 0
0 1	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 0 1 0
0 2	0	0	0	0	0	0	0	0	1	100	000	001	0 0 1 0 0 0
0 3	0	0	0	0	0	0	0	0	1	110	000	000	0 0 0 1 0 0
0 4	0	0	0	0	0	0	0	0	1	011	000	000	0 0 0 1 0 1
0 5	0	0	0	0	0	0	0	1	1	010	001	000	0 0 0 1 1 0
0 6	1	0	0	1	0	1	0	1	1	001	101	000	0 0 0 0 0 1
0 7	0	0	0	0	0	0	0	0	1	110	000	000	0 0 1 1 0 1
1 0	0	0	0	0	0	0	0	0	0	001	000	000	0 0 0 0 0 1

下一个微地址01

A9、A8 字段			A 字段				B 字段				C 字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	



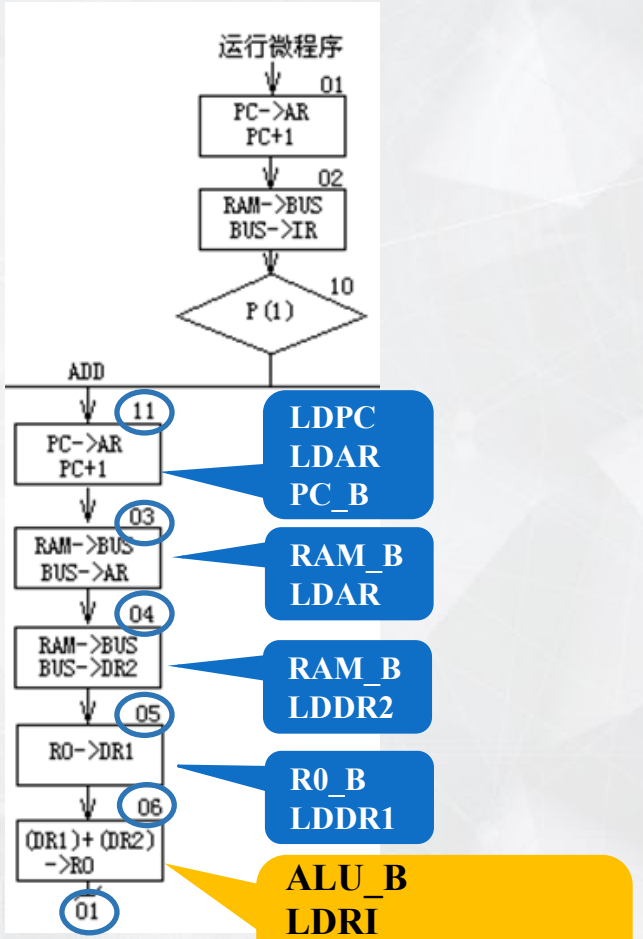
ADD指令微代码存储与分析(R0+[Addr]→R0)

微地址(8 进制)	S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	UA5——UA0
0 0	0	0	0	0	0	0	0	1	1	000	000	100	0 1 0 0 0 0
…	…									…	…	…	…
0 3	0	0	0	0	0	0	0	0	1	110	000	000	0 0 0 1 0 0
0 4	0	0	0	0	0	0	0	0	1	011	000	000	0 0 0 1 0 1
0 5	0	0	0	0	0	0	0	1	1	010	001	000	0 0 0 1 1 0
0 6	1	0	0	1	0	1	0	1	1	001	101	000	0 0 0 0 0 1
…	…									…	…	…	…
1 0	0	0	0	0	0	0	0	0	0	001	000	000	0 0 0 0 0 1
1 1	0	0	0	0	0	0	0	1	1	110	110	110	0 0 0 0 1 1

下一个微地址01

下一个微地址03

A9、A8 字段			A 字段				B 字段				C 字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	



STA指令微代码存储与分析(R0→[Addr])

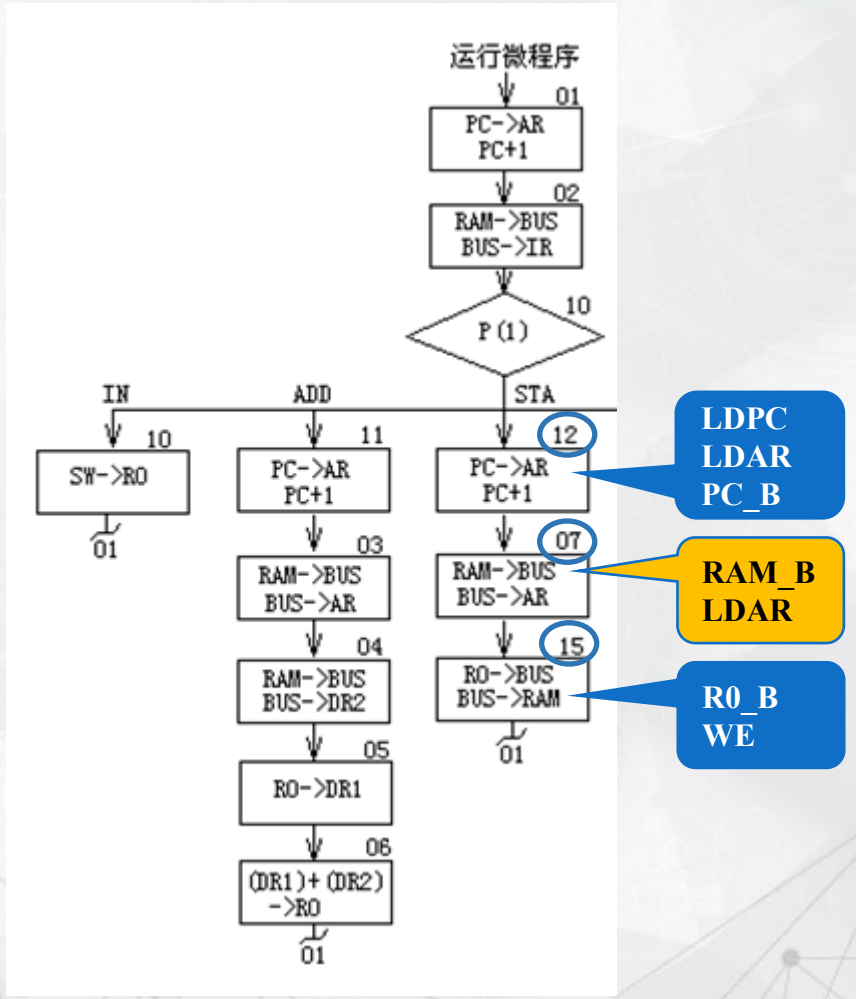
微地址(8进制)	S3 S2 S1 S0 M Cn WE A9 A8	A	B	C	UA5——UA0
0 0	0 0 0 0 0 0 0 1 1	000	000	100	0 1 0 0 0 0
0 1	0 0 0 0 0 0 0 1 1	110	110	110	0 0 0 0 1 0
…	…	…	…	…	…
0 7	0 0 0 0 0 0 0 0 1	110	000	000	0 0 1 1 0 1
1 0	0 0 0 0 0 0 0 0 0	001	000	000	0 0 0 0 0 1
1 2	0 0 0 0 0 0 0 1 1	110	110	110	0 0 0 1 1 1
1 3	0 0 0 0 0 0 0 1 1	110	110	110	0 0 1 1 1 0
1 5	0 0 0 0 0 0 1 1 1	000	001	000	0 0 0 0 0 1

下一个微地址 15

下一个微地址 07

下一个微地址 01

A9、A8字段			A字段				B字段				C字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	



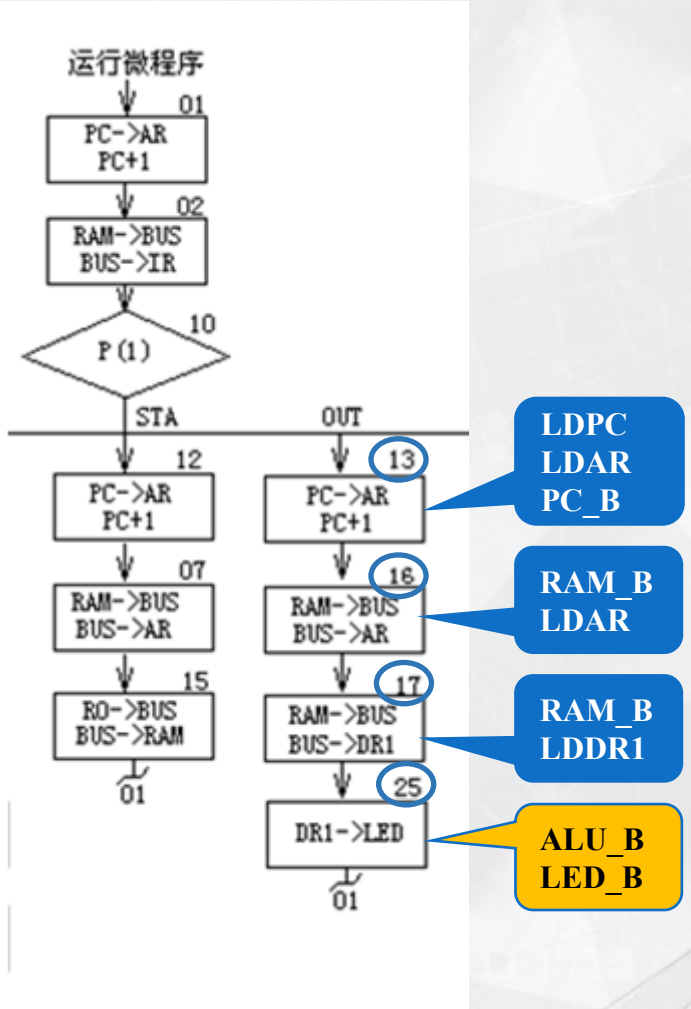
OUT指令微代码存储与分析([Addr] → “Onput Device”)

微地址(8 进制)	S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	UA5——UA0
0 0	0	0	0	0	0	0	0	1	1	000	000	100	0 1 0 0 0 0
…	…									…	…	…	…
1 3	0	0	0	0	0	0	0	1	1	110	110	110	0 0 1 1 1 0
…	…									…	…	…	…
1 6	0	0	0	0	0	0	0	0	1	110	000	000	0 0 1 1 1 1
1 7	0	0	0	0	0	0	0	0	1	010	000	000	0 1 0 1 0 1
…	…									…	…	…	…
2 5	0	0	0	0	0	1	0	1	0	000	101	000	0 0 0 0 0 1

下一个微地址17

下一个微地址01

A9、A8 字段			A 字段				B 字段				C 字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	



JMP指令微代码存储与分析(Addr →PC)

微地址 (8 进制)	S3 S2 S1 S0 M Cn WE A9 A8	A	B	C	UA5——UA0
0 0	0 0 0 0 0 0 0 1 1	000	000	100	0 1 0 0 0 0
0 1	0 0 0 0 0 0 0 1 1	110	110	110	0 0 0 0 1 0
…	…	…	…	…	…
1 4	0 0 0 0 0 0 0 1 1	110	110	110	0 1 0 1 1 0
…	…	…	…	…	…
					0 0 0 1
					0 0 0 0
					0 0 0 1

下一个微地址 26

下一个微地址 01

A9、A8 字段			A 字段				B 字段				C 字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P（1）
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P（2）
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P（3）
			1	0	0	LDIR	1	0	0		1	0	0	P（4）
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1