

A network diagram featuring several blue human silhouettes connected by a web of blue lines. The background is a light beige color with a faint world map. The text '计算机组成原理课程设计' is overlaid in green.

# 计算机组成原理课程设计

## 实验 7

### 简单模型机实验



# 实验 7 简单模型机实验

一、实验目的

二、相关单元

三、模型机结构分析

四、指令格式框架

五、寻址方式

六、微指令格式

七、控制台操作

八、实验原理

九、实验要求

十、思考



# 一、实验目的

1. 在掌握各部件单元电路实验的基础上，进一步将其组成系统，**构造一台基本模型计算机。**
2. 掌握写入、校验、执行程序的方法。
3. 为简单模型计定义 5 条机器指令，编写相应的微程序和测试程序，并上机调试，**掌握整机概念。**





## 二、相关单元

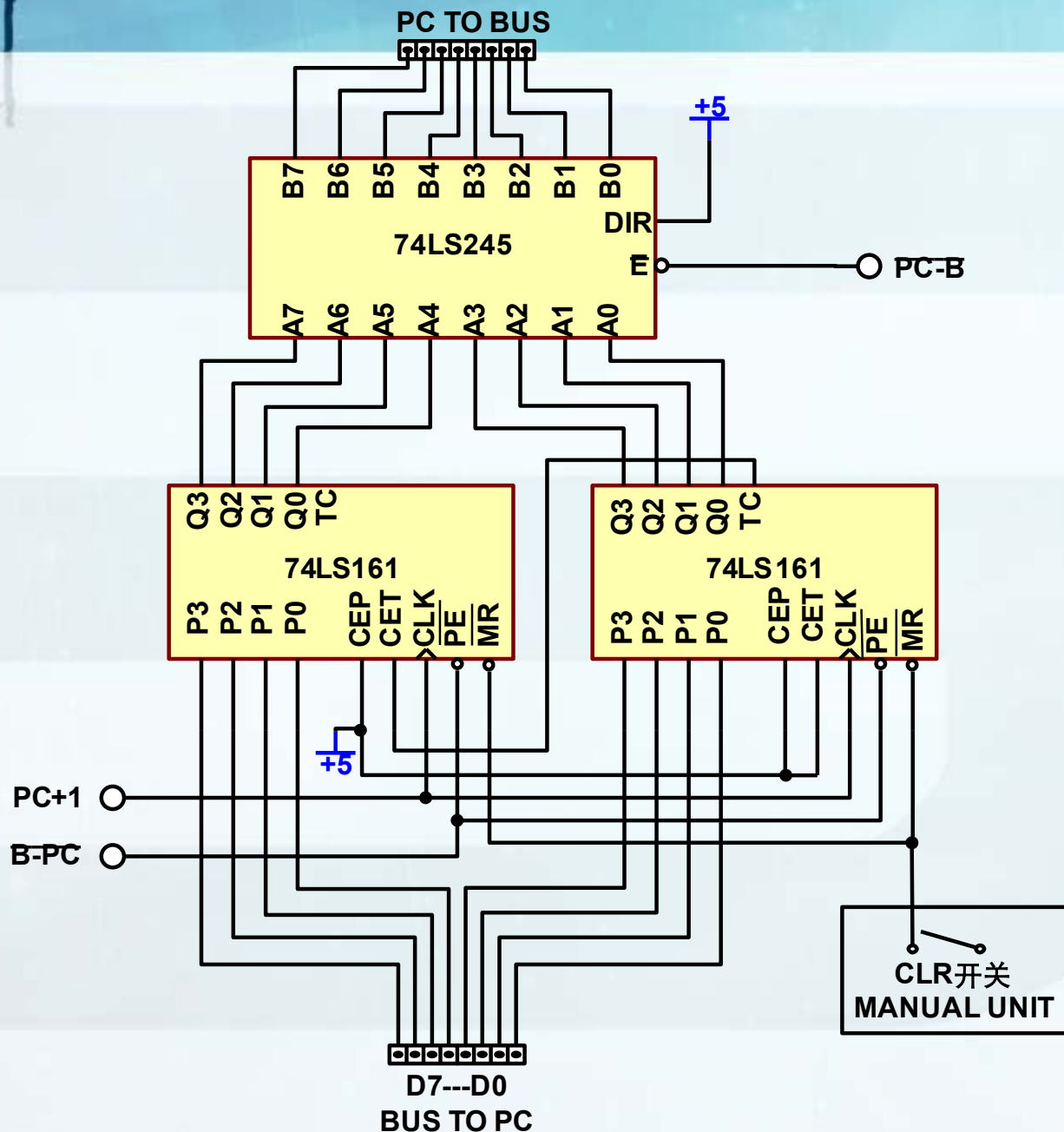
1. 微控器单元、时钟单元、手动开关控制单元、通用寄存器单元、运算器单元、输入 / 输出单元、存储器单元
2. 地址单元（ ADDRESS UNIT ）
3. 指令寄存与译码单元（ INS UNIT ）



## 2、地址单元（程序计数器 PC）

- 地址单元由 **AR**、**PC**、**8 位地址显示灯** 构成
- PC 由两片计数器 74LS161（4 位）构成，输入端接自总线单元（BUS UNIT）的 D7-D0，输出端则通过一片三态缓冲器 74LS245 输出至总线。
- PC 的操作与控制：
  - ① PC 清零：CLR 开关的负脉冲将使 PC 清零；
  - ② PC 置数：则需要控制信号  **$PC+1 = \text{上升沿}$  且  $B-PC \neq 0$** ；
  - ③ PC 加 1 计数：则需要  $PC+1 = \text{上升沿}$ 。
  - ④ PC 送数至总线：则需要  $PC \rightarrow B \neq 0$ 。

# 程序计数器 PC 的电路原理图



# 74LS161 功能表

输入信号					响应操作
MR	PE	CEP	CET	CLK	
0	×	×	×	×	异步清零， $Q_3 \sim Q_0$ 输出低电平
1	0	×	×	↑	预置数， $P_3 \sim P_0$ 装入 $Q_3 \sim Q_0$
1	1	1	1	↑	加1计数，并从 $Q_3 \sim Q_0$ 输出
1	1	×	0	×	保持， $Q_3 \sim Q_0$ 输出保持不变
1	1	0	×	×	保持， $Q_3 \sim Q_0$ 输出保持不变





## 3、指令寄存与译码单元

- 指令寄存器
- 指令译码器
- 寄存器译码电路





# 指令寄存器 IR

■ 由一片

✓ 其输

D7 ~

✓ 输出

NS

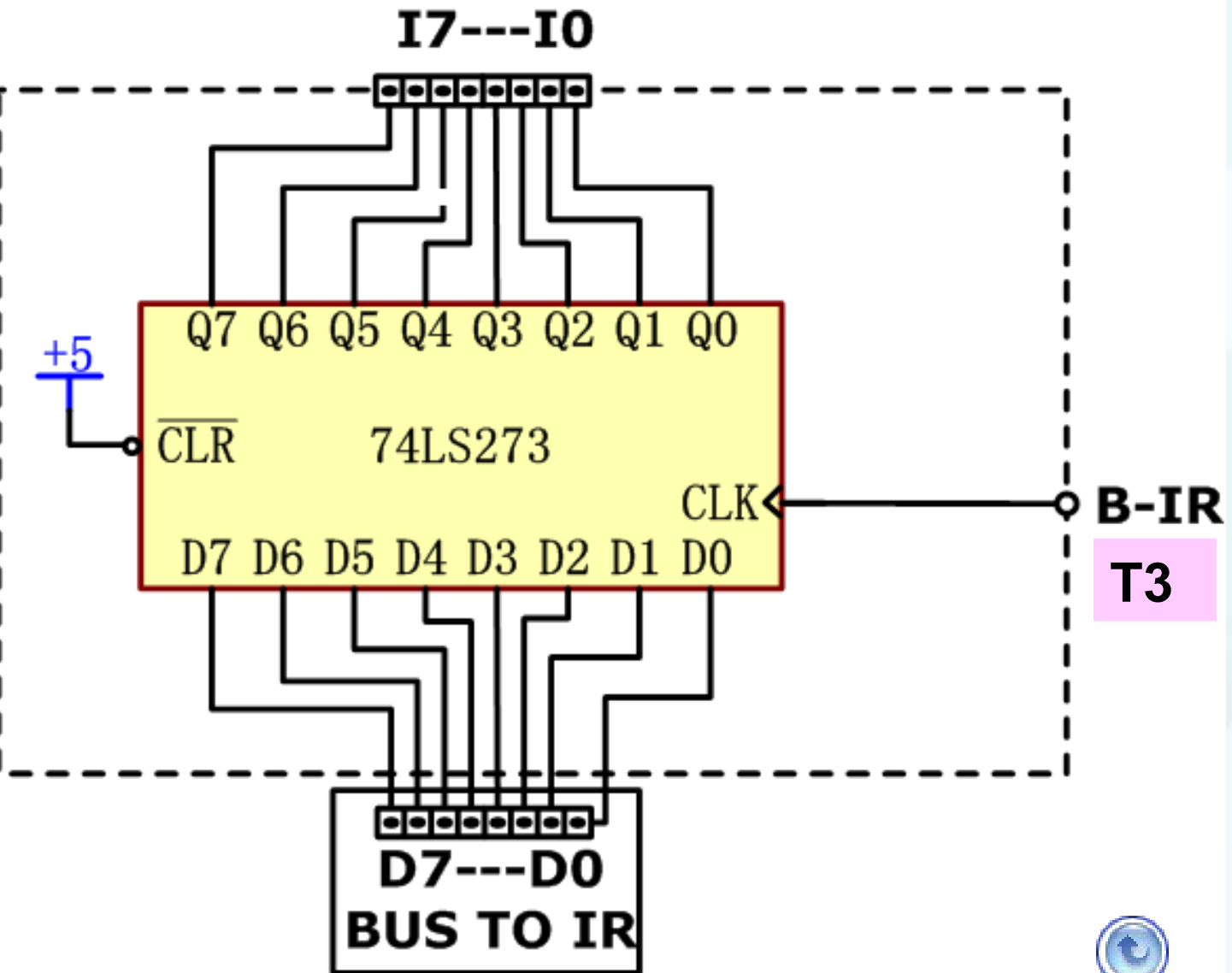
寄存器

路由使

■ 控制信

据总线

■ 电路原



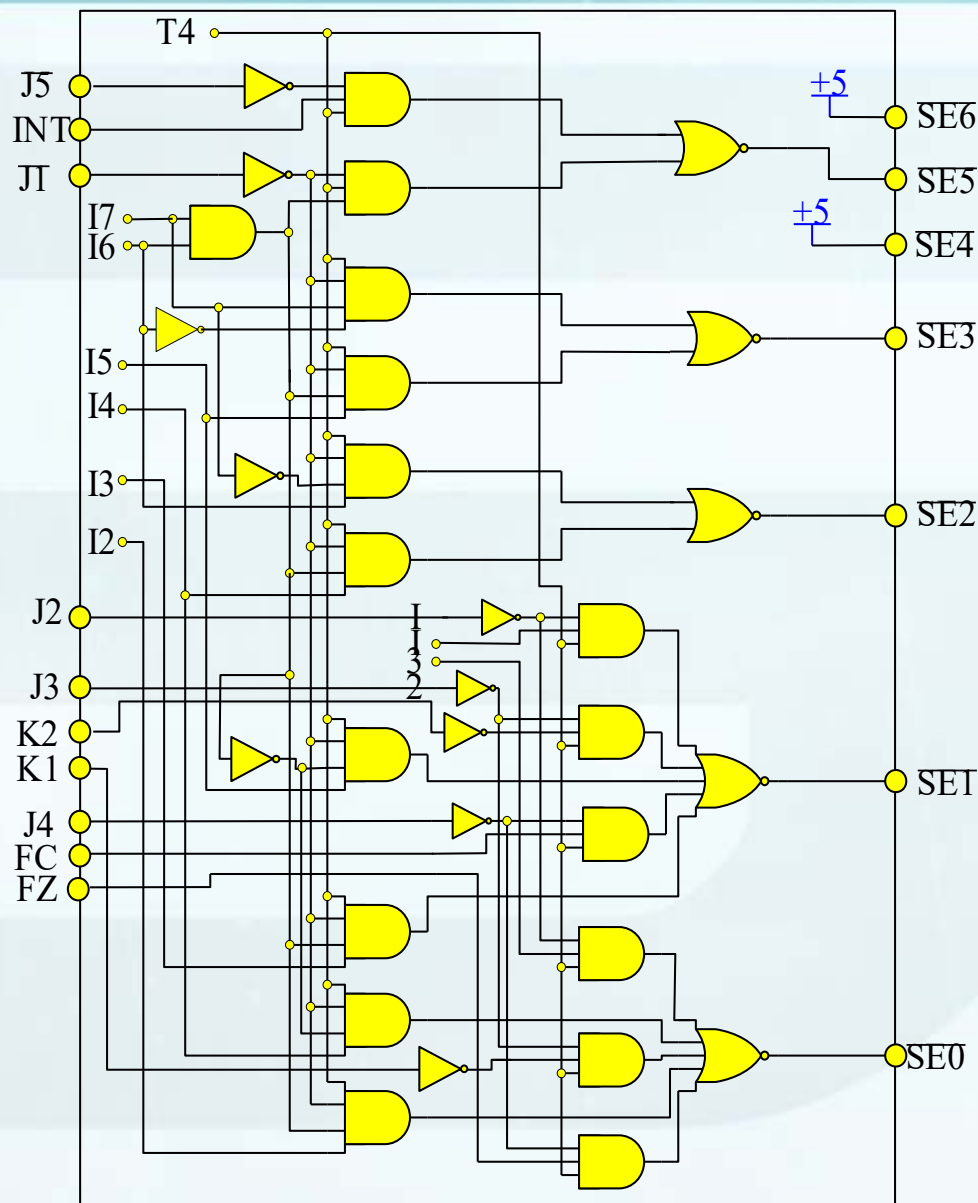
# 指令译码器

- 指令译码器 ID 功能：  
根据指令操作码，译码产生该指令的**微程序入口地址**。

- 模型机后继微地址散转电路：

①**输入**：指令操作码  
I7 ~ I2， INT 信号，状态标志， K2  
K1 开关

②**输出** SE6# ~ SE0#  
：用于**修改微地址**



# 微地址转移控制逻辑：J1 #

❖ J1#=0 时，根据指令的操作码（OP） $I_7 \sim I_2$  进行散转，产生该条指令的微程序入口地址。散转规则：

■ 当  $I_7 I_6 = 11$  时（即指令格式 4 的指令），则

SE5#=0，即  $1 \rightarrow MA_5$ ，并且：

① 若  $I_5=1$ ，则 SE3#=0，即  $1 \rightarrow MA_3$ ；

② 若  $I_4=1$ ，则 SE2#=0，即  $1 \rightarrow MA_2$ ；

③ 若  $I_3=1$ ，则 SE1#=0，即  $1 \rightarrow MA_1$ ；

# 微地址转移控制逻辑：J1 #

❖ J1#=0 时，根据指令的操作码（OP） $I_7 \sim I_2$  进行散转，产生该条指令的微程序入口地址。散转规则：

也即：

■ 当  $I_7 I_6 = 11$  时（即指令格式 4 的指令），

$$MA_6 \sim MA_0 = MA_6 1 MA_0 I_5 I_4 I_3 I_2$$

■ 当  $I_7 I_6 \neq 11$  时（即指令格式 1、2 的指令），  
 $MA_6 \sim MA_0 = MA_6 MA_5 MA_0 I_7 I_6 I_5 I_4$

④ 若  $I_4 = 1$ ，则  $SEO\# = 0$ ，即  $1 \rightarrow MA_0$ ；



## 微地址转移控制逻辑：J2#

❖当 J2#=0 时，根据指令操作码 1312 进行散转，转移至相应指令的微程序段。

- 主要应用于格式 2 的机器指令，即含寻址方式码（MOD）的格式。该指令格式中，1716 为第一操作码，通常是指令格式二的特征码，1514 为寻址方式码（MOD），1312 为指令的第二操作码，用于区分格式二的四条指令。

# 微地址转移控制逻辑：J2#

- 在这些指令的微程序实现中，通常首先根据 I7 ~ I4 进行 J1 # 散转，转移至相应的寻址方式微程序段（主要功能是计算有效地址 EA）；然后进行第二次的 J2 # 散转，转移至不同的机器指令的微程序段（实现指令的功能）。

也即： $MA_6 \sim MA_0 = MA_6 \sim MA_2 \quad I_3 \quad I_2$

✓  $I_3=1$  时，则  $SE1\#=0$ ，即  $1 \rightarrow MA_1$ ；

✓  $I_2=1$  时，则  $SE0\#=0$ ，即  $1 \rightarrow MA_0$ ；

# 微地址转移控制逻辑：J3#、J4#

❖ J3#=0 时，根据开关 K2、K1 状态进行转移；主要应用于控制台操作指令。散转规则：

也即： $MA_6 \sim MA_0 = MA_6 \sim MA_2 \overline{K2} \overline{K1}$

■ K1=0 时，则 SE0#=0，即  $1 \rightarrow MA_0$ ；

❖ J4#=0 时，根据条件 FC 或 FZ 进行转移；主要应用于条件转移指令 JZC 或 JZ 或 JC。散转规则：

也即： $MA_6 \sim MA_0 = MA_6 \sim MA_2 FC FZ$

■ FZ=1 时，则 SE0#=0，即  $1 \rightarrow MA_0$ ；

## 微地址转移控制逻辑：J5#

❖ J5#=0 时，根据中断请求信号 INT 是否有效，确定转移。散转规则：

- INT=1 时，则 SE5#=0，即  $1 \rightarrow MA_5$ ；

也即： $MA_6 \sim MA_0 = MA_6 INT MA_5 \sim MA_0$



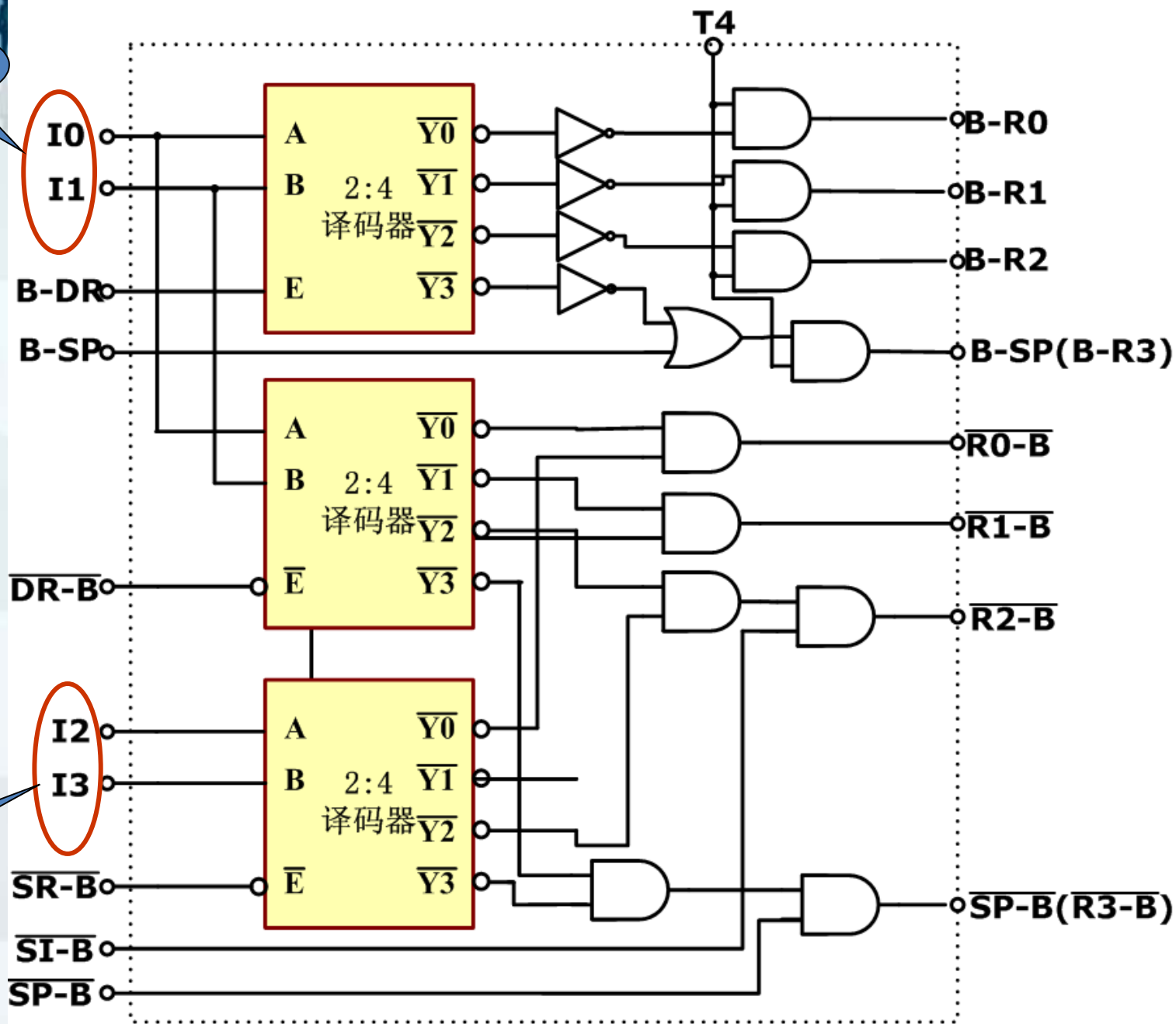




# 寄存器译码电路

- ❖ 功能：依据指令的 **DR** 和 **SR** 字段，将微控器发出的统一的寄存器控制信号，翻译为**具体的、不同的**寄存器控制信号。
- ❖ 输入信号有：
  - **B-DR**、**DR-B#**、**SR-B#**、**SI-B#**、**SP-B#**：  
来自微控器单元 **MAIN CONTROL UNIT**。
  - 指令码 13-10：来自指令寄存器（即 **SR**、**DR** 字段）
- ❖ 输出信号为：（送至寄存器单元 **REG UNIT**）
  - 寄存器打入脉冲：**B-R0**、**B-R1**、**B-R2**、**B-R3**。

DR



SR

存  
器

# 寄存器控制信号译码由路的输入与输出之间的逻辑关系

输入	输出
$B-DR=1, \text{ 且 } I1I0=00$	$B-R0=1$
$B-DR=1, \text{ 且 } I1I0=01$	$B-R1=1$
$B-DR=1, \text{ 且 } I1I0=10$	$B-R2=1$
$B-DR=1, \text{ 且 } I1I0=11$	$B-R3=1$
$\overline{DR} \cdot \overline{B} = 0, \text{ 且 } I1I0=00$	$\overline{R0} \cdot \overline{B} = 0$
$\overline{DR} \cdot \overline{B} = 0, \text{ 且 } I1I0=01$	$\overline{R1} \cdot \overline{B} = 0$
$\overline{DR} \cdot \overline{B} = 0, \text{ 且 } I1I0=10$	$\overline{R2} \cdot \overline{B} = 0$
$\overline{DR} \cdot \overline{B} = 0, \text{ 且 } I1I0=11$	$\overline{R3} \cdot \overline{B} = 0$
$\overline{SR} \cdot \overline{B} = 0, \text{ 且 } I3I2=00$	$\overline{R0} \cdot \overline{B} = 0$
$\overline{SR} \cdot \overline{B} = 0, \text{ 且 } I3I2=01$	$\overline{R1} \cdot \overline{B} = 0$
$\overline{SR} \cdot \overline{B} = 0, \text{ 且 } I3I2=10$	$\overline{R2} \cdot \overline{B} = 0$
$\overline{SR} \cdot \overline{B} = 0, \text{ 且 } I3I2=11$	$\overline{R3} \cdot \overline{B} = 0$
$\overline{SI} \cdot \overline{B} = 0$	$\overline{R2} \cdot \overline{B} = 0$
$B-SP=1$	$B-SP=1$
$\overline{SP} \cdot \overline{B} = 0$	$\overline{SP} \cdot \overline{B} = 0$



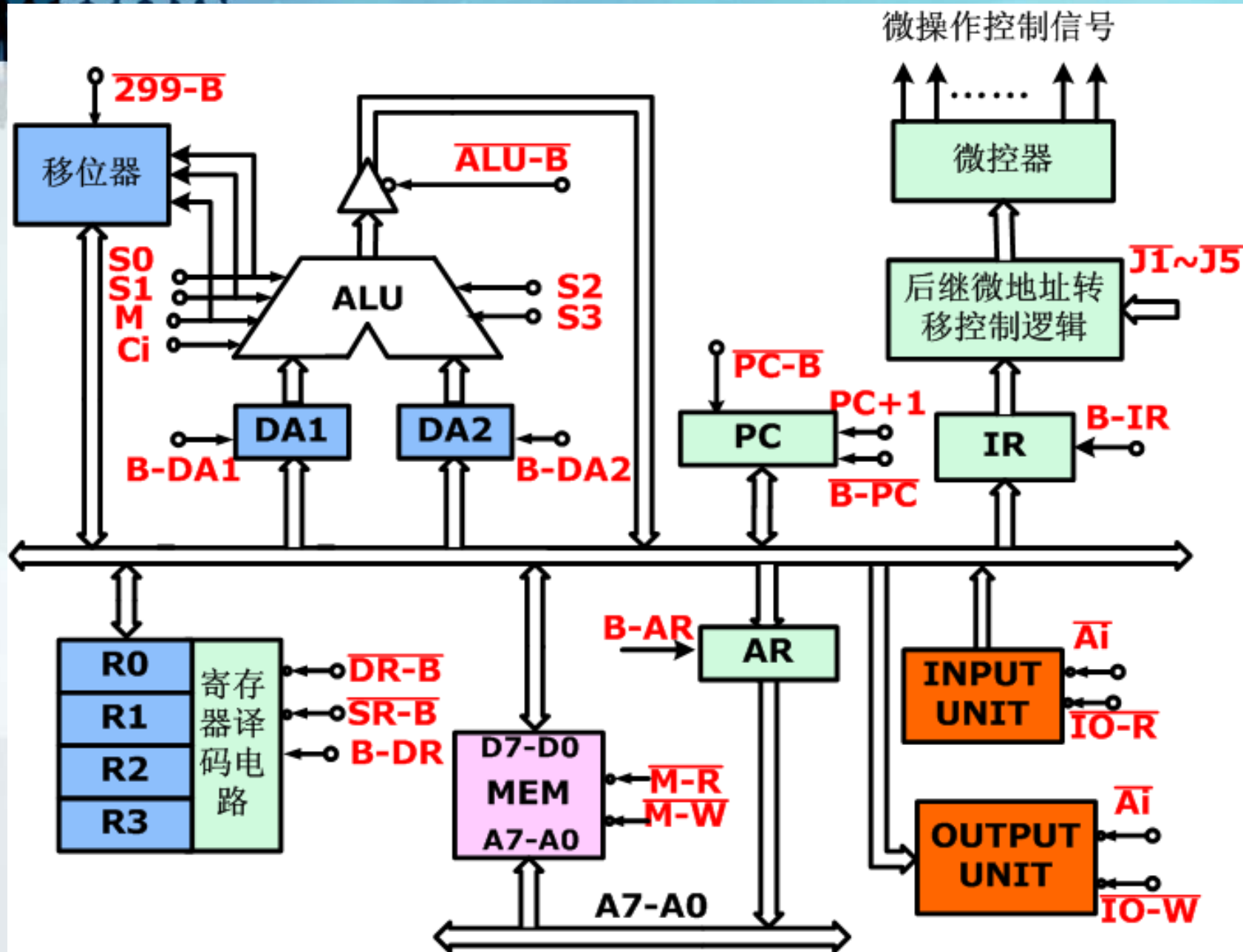


## 三、模型机结构分析

1. 系统结构
2. 模型机的数据通路



# 1. 系统结构





## 2. 模型机的数据通路

### ■ 存储器**读**操作（分两步）：

- ① 将地址送到总线，并打入地址寄存器 AR ；
- ② 发送  $M-R\#=0$  ，启动存储器读操作，并将读出的数据从总线上接收至目的部件（如 R0 、 DA1 、 DA2 等）。

### ■ 存储器**写**操作（分两步）：


- ① 将地址送到总线，并打入地址寄存器 AR ；
- ② 将数据送到总线，并发送  $M-W\#=0$  ，启动存储器

## 2. 模型机的数据通路

### ■ 运算器的运算操作（分三步）：

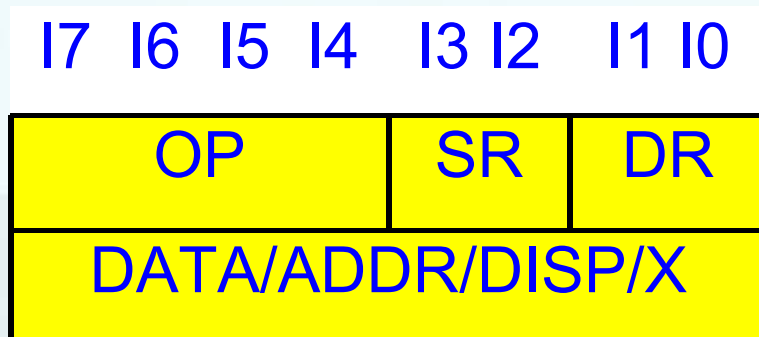
① 将**第一个数据**送到总线，并打入 ALU 暂存器 DA1 /DA2 ；

② 将**第二个数据**送到总线，并打入 ALU 暂存器 DA1 /DA2 ；

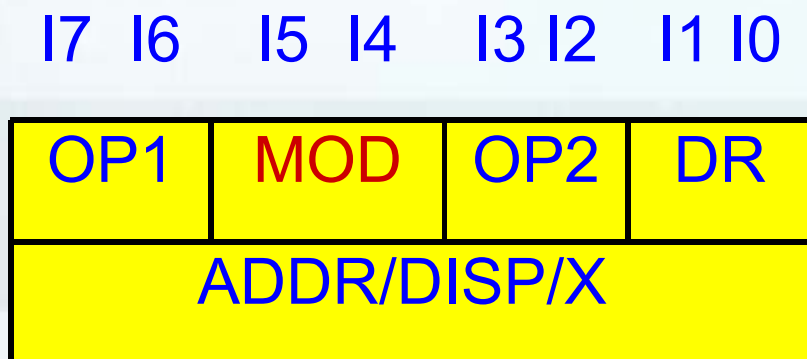
③ 发送运算器功能选择信号  $S_3 \sim S_0$ 、 $M$ 、 $C_i$ ，  
控制 ALU 进行某种运， $ALU-B\#=0$ ，将总线上的  
**运算结果**送目的部件（例如某通用寄存器）。

## 四、指令格式框架

### 格式 1：一般指令格式



### 格式 2：带寻址方式码的指令格式

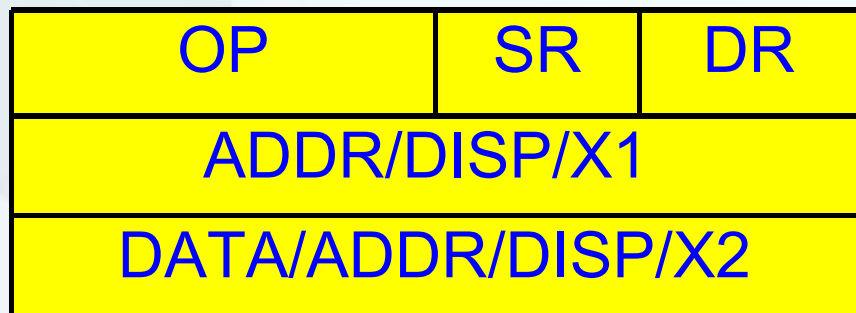




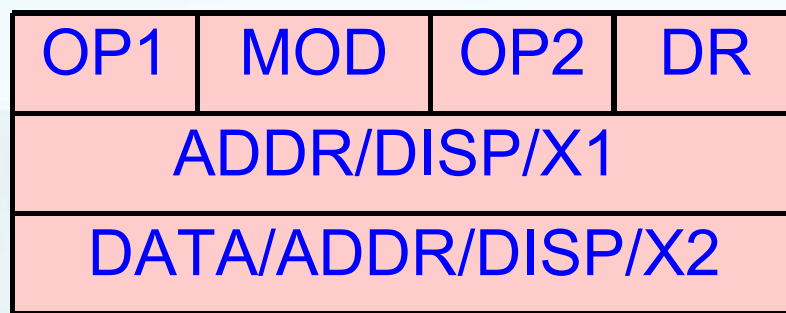
## 四、指令格式框架

### 格式 3：三字节指令

17 16 15 14    13 12    11 10



17 16 15 14    13 12    11 10



### 格式 4：操作码扩展指令格式

17 16    15 14 13 12    11 10



## 五、寻址方式

1. 模型机的指令系统，可实现基本寻址方式

。

2. 对于指令格式 2，即带寻址方式 MOD 的指令格式，可以做以下定义。

- MOD = 00 : 直接寻址；
- MOD = 01 : 间接寻址；
- MOD = 10 : 变址寻址，其中 SI 隐含为 R2；

■ MOD = 11 : 相对寻址

## 六、微指令格式

直接控制  
法

M23:21 ( 3 )	M20:18 ( 3 )	M17:15 ( 3 )	M14 ( 1 )	M13:8 ( 6 )	M7 (1)	M6:0 ( 7 )
BTO	OTB	FUNC	FS	S3:0 M Ci	空	MA6:0

字段直接  
编译法

字段直接  
编译法

字段间接编  
译法

字段间接编  
译法


# 模型机微指令字段编码表

编码 + 译 码	BT0	0TB	FUNC	
			FS=1	FS=0
000	空	空	PC+1	空
001	B-DA1(t4)	ALU-B# (t2)	J1# (t2)	M-W# (t3)
010	B-DA2(t4)	299-B# (t2)	J2# (t2)	M_R# (t2)
011	B-IR(t3)	SR-B# (t2)	J3# (t2)	I/O-W# (t3)
100	B-DR(t4)	DR-B# (t2)	J4# (t2)	I/O_R# (t2)
101	B-SP( t4)	SI-B# (t2)	J5# (t2)	INT_R# (t2)
110	B-AR(t3)	SP-B# (t2)	CyCn# (t2)	INT_E# (t2)
111	B-PC# (t4)	PC-B# (t2)	CyNCn# (t2)	

字段直接编译  
法

# 七、控制台操作

K2 K1	控制台指令	功能
10	读内存	拨动 CLR 开关  后，按 START 微动开关，对内存从 0 号单元开始连续进行读操作
01	写内存	拨动 CLR 开关  后，按 START 微动开关，对内存从 0 号单元开始连续进行写操作
00	启动程序	拨动 CLR 开关 后，按 START 微动开关，则转入取指令微程序段入口开始执行
模型机开机或 CLR 总清后，运行的第一条微指令地址为 0H		

在真实的计算机里，开机或  系统复位后的第一条微指令是 **取指令并译码** 微指令。



# 八、实验原理

1. 指令系统（5 条指令）
2. 微程序流程图
3. 微代码编写
4. 程序代码编写
5. 装入并校验程序代码的方法
6. 执行程序（微程序）方法



# 1. 指令系统

## ■ (1) ADD DR, SR

$(SR) + (DR) \rightarrow DR$

I7-I4 I3 I2 I1 I0

0000	SR	DR
------	----	----

## ■ (2) STA [ADDR], DR

$DR \rightarrow [ADDR]$

I7 I6 I5 - I2 I1 I0

11	0000	DR
ADDR		

## ■ (3) IN DR, PORTAR

从外设端口地址 PORTAR 读取一个数到目的寄存器 D

R

I7 I6 I5 - I2 I1 I0

11	0001	DR
PORTAR		

# 1. 指令系统

■ (4) OUT [PORTAR] ,  
[ADDR]

[ADDR] → LED ;

I7 I6 I5-I2 I1 I0

11	0010	xx
ADDR		
PORTAR		

■ (5) JMP ADDR

■ ADDR → PC

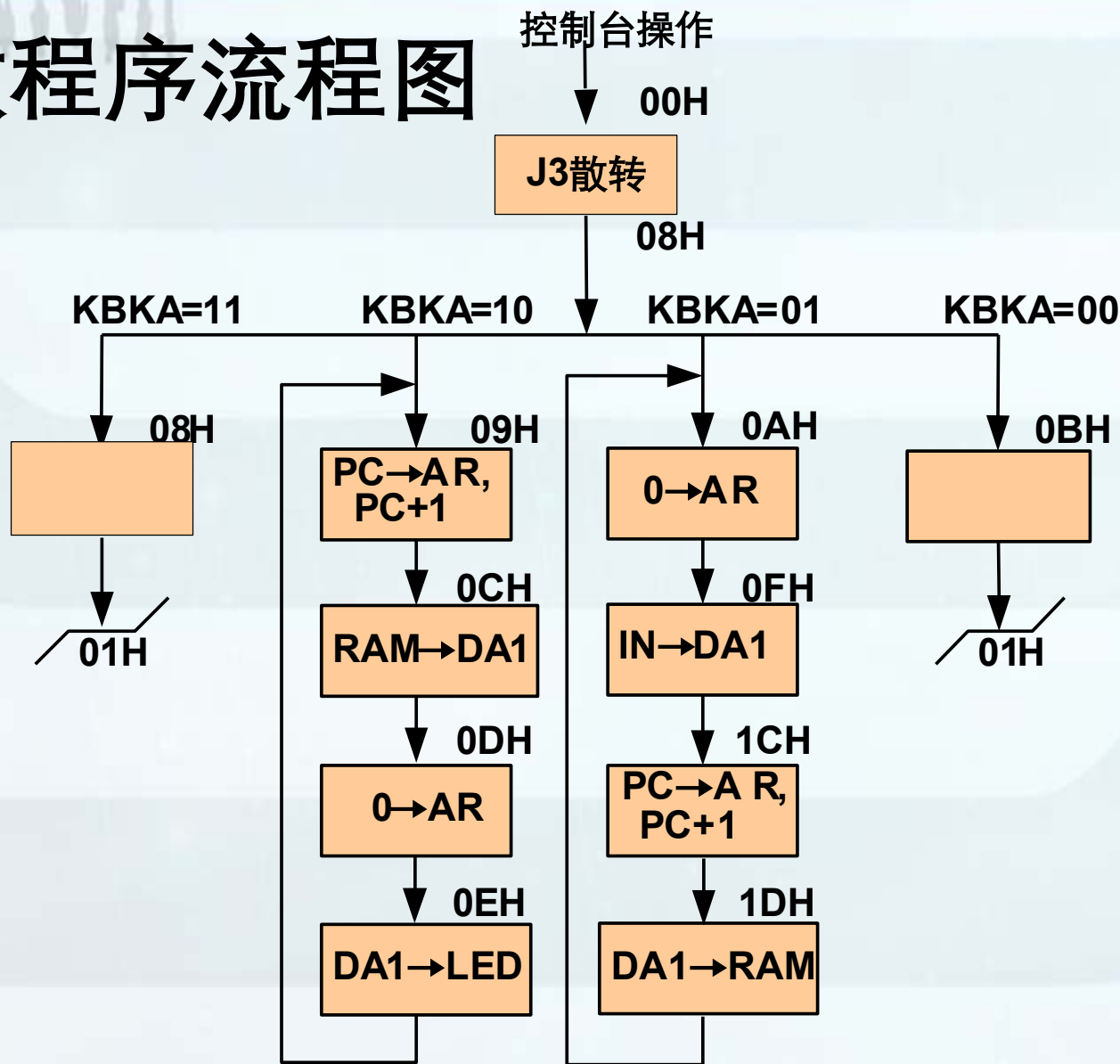
I7 I6 I5 -I2 I1 I0

11	0011	xx
ADDR		



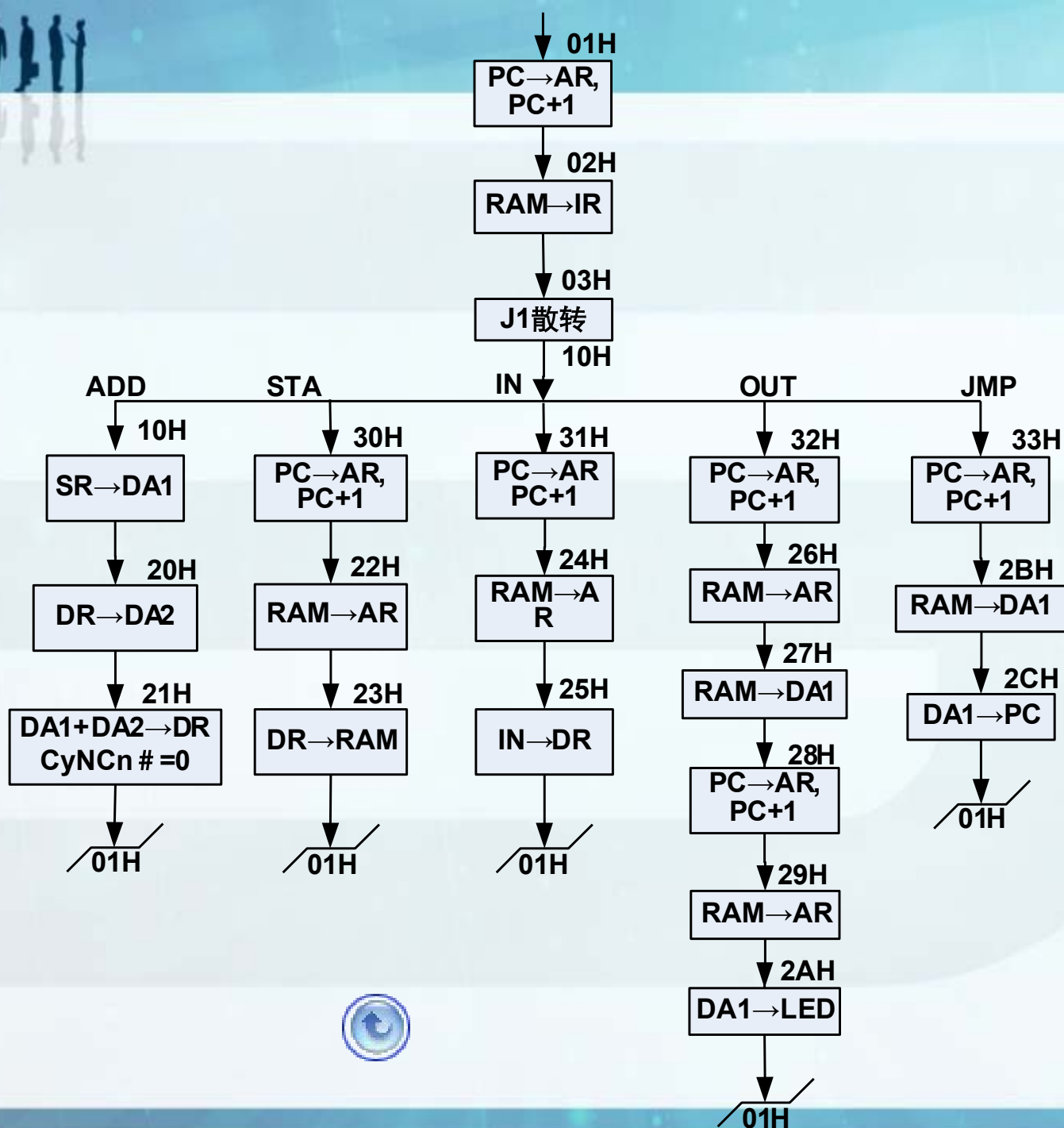


## 2. 微程序流程图



控制台操作  
联机时不需要

# 实验 5、 6 的 微程 序流 程图



# 3. 微代码编写

## ■ p76 表 7-11

微地址	微代码	BTO	OTB	FUNC	FS	S3 S2 S1 S0 M Ci	N	下址	微指令注释
00H	01C008	000	000	011	1	0 0 0 0 0 0	0	0001000	$\overline{J_3}$
01H	DC4002	110	111	000	1	0 0 0 0 0 0	0	0000010	PC→AR, PC=PC+1
02H	610003	011	000	010	0	0 0 0 0 0 0	0	0000011	RAM→IR
03H	00C010	000	000	001	1	0 0 0 0 0 0	0	0010000	$\overline{J_1}$
08H	000001	000	000	000	0	0 0 0 0 0 0	0	0000001	NULL
09H	DC400C	110	111	000	1	0 0 0 0 0 0	0	0001100	PC→AR, PC=PC+1
0AH	C40E0F	110	001	000	0	0 0 1 1 1 0	0	0001111	0→AR
0BH	000001	000	000	000	0	0 0 0 0 0 0	0	0000001	NULL
0CH	21000D	001	000	010	0	0 0 0 0 0 0	0	0001101	RAM→DA <sub>1</sub>
0DH	C40E0E	110	001	000	0	0 0 1 1 1 0	0	0001110	0→AR
0EH	058109	000	001	011	0	0 0 0 0 0 1	0	0001001	DA <sub>1</sub> →OUT
0FH	22001C	001	000	100	0	0 0 0 0 0 0	0	0011100	IN→DA <sub>1</sub>



## 4. 程序代码编写

地 址	内 容	助 记 符	备 注
00H	11000100B	; IN R <sub>0</sub> , [00H]	IN → R <sub>0</sub>
01H	00000000B	; 端口地址 00H	
02H	11000101B	; IN R <sub>1</sub> , [00H];	IN → R <sub>1</sub>
03H	00000000B	; 端口地址 00H	
04H	00000001B	; ADD R <sub>0</sub> , R <sub>1</sub>	R <sub>0</sub> + R <sub>1</sub> → R <sub>0</sub>
05H	11000000B	; STA R <sub>0</sub> , [10H]	R <sub>0</sub> → [10H]
06H	00010000B	; 直接地址 10H	
07H	11001000B	OUT [10H], [PORTAR]	[10H] → LED
08H	00010000B	; 直接地址 10H	
09H	00000000B	; 端口地址 00H	
0AH	11001100B	; JMP 00H	00H → PC
0BH	00000000B	; 直接地址 00H	
.....	.....		
10H		; 和	须检验的结果



## 5. 装入并校验程序代码的方法

- 手动装入：执行控制台“写内存”操作，通过 8 位输入开关输入程序和数据（不做要求）
- 手动校验：执行控制台“读内存”操作来实现校验（不做要求）
- 联机装入：通过上位机（虚拟仿真）软件
- 联机校验：打开上位机软件内存窗口查看程序代码





# 手动装入程序机器码

- 执行控制台“写内存”操作，通过 8 位输入开关输入程序和数据

①将编程开关置于“**RUN**”状态，RUN#/STEP 开关置于“**STEP**”状态。

②操作 CLR 开关，使 CLR 信号“ $1 \rightarrow 0 \rightarrow 1$ ”，程序计数器 PC 清零，微地址清零。

③开关 K2K1 置为“0 1”，即写内存状态。

# 手动装入程序机器码

- ④ 按动启动键 **START** 一次，则从控存 **00H** 单元开始执行微指令，微地址显示灯显示“0001010”，第二次按动 **START** 键，微地址显示灯显示“001111”，此时，将数据开关置为要写入的机器指令代码或数据，再按动 **START** 键两次，即完成该条指令的写入，同时 PC 指向下一个内存单元。继续按动 **START** 键，当且仅当微地址显示灯显示“0001111”时，才从开关上置入指令代码，直至所有的程序代码写入完毕。





# 手动校验程序机器码

■ 执行控制台“读内存”操作，通过输出设备（8 位 LED 显示灯）显示程序和数据代码

① 将编程开关置于“RUN”状态，RUN#/STEP 开关置于“STEP”状态。

② 操作 CLR 开关，使 CLR 信号“ $1 \rightarrow 0 \rightarrow 1$ ”，程序计数器 PC 清零，微地址清零。





# 手动校验程序机器码

④按动启动键 START 一次，则从控存 00H 单元开始执行微指令，微地址显示灯显示“0001001”，第二次按动 START 键，微地址显示灯显示“0001100”，第三次按动 START 键，微地址显示灯显示“0001101”，第四次按动 START 键，微地址显示灯显示“0001110”，此时，输出设备（OUTPUT DEVICE）发光管上将显示内存 00H 号单元的内容，检查是否与写入的数据相同。



# 手动校验程序机器码

⑤继续按动 START 键，当且仅当微地址显示灯显示“0001001”时，发光管上显示的内容才是内存的数据。每个循环 PC 会自动增 1，由此，可检查后续单元的内容是否正确。



Why?





# 联机装入微码、程序代码方法

## ■ 上位机软件工作方式：

- ①**联机主控：**可以通过软件来控制动态显示指令在实验仪上执行过程；上下载程序、微程序；执行程序、微程序、单步程序、单步微程序、连续执行、停止执行
- ②**联机从控：**软件只用来上下载程序、微程序。
- ③**仿真方式：**不使用实验仪，所有部件用软件模拟仿真实现。

# 联机装入微码、程序代码方法

- 编辑微码：控存窗口
- 装入微码：下载按钮、菜单
- 编辑程序代码：主存窗口
- 装入程序代码：下载按钮、菜单
- 按照模型机的硬件机理（Reset 后，PC 清零，CMAR 清零），取指令微程序段应从 0 号控存开始存放，而程序代码应从 0 号内存开始存放。





## 6. 执行程序（微程序）方法

■ **脱机执行**：执行控制台“启动程序”操作指令  
(**不做要求**)

① K2 K1 = 00/11，编程开关（三态）= **RUN**

✓ 单步执行：RUN#/STEP = STEP

✓ 连续执行：RUN#/STEP = RUN #

② CLR 信号“1→0→1”，使**微地址清零**，PC  
**清零**，即**程序首址为 00H**

③ 按动 **START** 键，单步执行（微指令）或者连续执行。



## 6. 执行程序（微程序）方法


### ■ 联机执行：

①联机主控：通过软件控制，文件必须

先下装到实验仪 / 仿真机

②联机从控：同脱机执行的控制方法





## 九、实验要求

1. 根据实验原理和相关单元电路，**画出实验接线图**
2. 使用**联机方式**装入程序和微程序代码，比较和检查结果是否正确





# 十、思考

1. 分析实验的指令系统中，如何实现**隐含**对 R0（目的操作数）的寻址？如果要隐含使用 R1 做目的操作数，如何实现？
2. 在微程序控制流程中，若希望从 **00H** 号单元就开始执行**取指令**的微指令，则如何修改 **00H** 单元的微指令？







**The End !**