



第八讲

微程序控制方式下模型机的设计实例 (一)

—— 模型机的系统结构及指令系统的设计





一、学习内容：

1. 指令格式的设计方法；
2. 微指令格式的设计方法；
4. 微机系统结的组成；
4. 微程序控制器的组成
5. 指令译码工作原理





学习重点

1. 掌握机器指令格式的设计方法
2. 掌握微指令设计方法
3. 了解微程序控制器的硬件组成
4. 掌握在模型机下微程序的设计





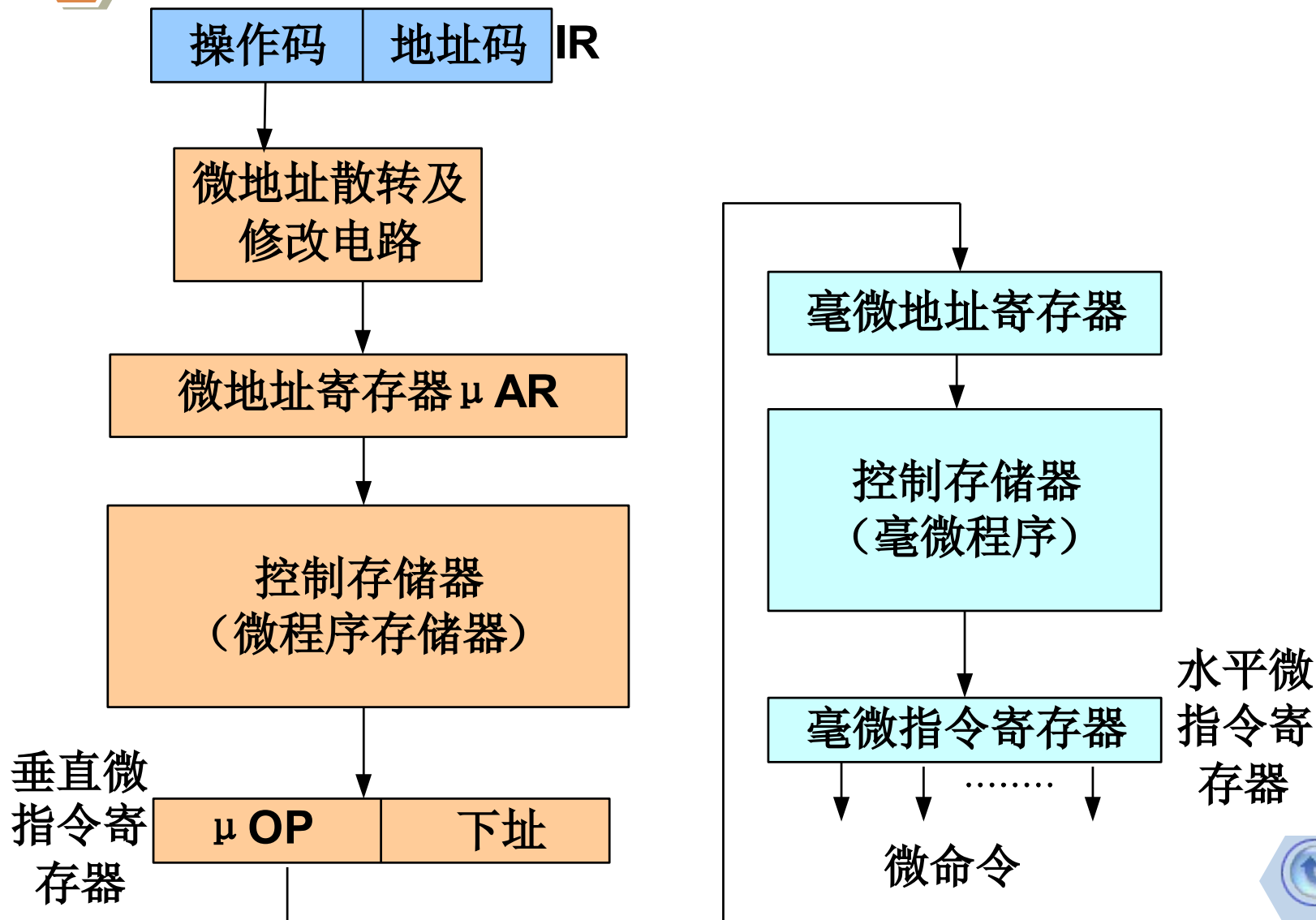
4、微程序控存和动态微程序设计

- 在一台微程序控制的计算机中，假如能根据用户的要求改变微程序，那么这台机器就具有动态微程序设计功能。
- 动态微程序设计的目的是使计算机能更灵活、更有效地适应于各种不同的应用场合。
- 动态微程序设计要求用户对计算机硬件组成结构非常熟悉，因此真正由用户自行编写微程序是很困难的，所以尽管设想很好，事实上是难以推广，一般要由机器设计人员才能设计实现。





5、毫微程序设计





四、微程序控制方式下模型机的设计实

1

指令格式

2

微型机系统结构

3

微指令格式

4

微程序控制器

5

指令译码器





1、指令格式

- 指令格式设计

指令格式的决定了硬件结构的组成

- 指令格式的设计要规整，容易理解，易于硬件实现控制





1、指令格式

- 格式 1：一般指令格式

I7 I6 I5 I4 I3 I2 I1 I0

OP	SR	DR
DATA/ADDR/DISP/X		

- 格式 2：带寻址方式码的指令格式

I7 I6 I5 I4 I3 I2 I1 I0

OP1	MOD	OP2	DR
ADDR/DISP/X			





1、指令格式

■ 格式 3：三字节指令

17 16 15 14 13 12 11 10

OP	SR	DR
ADDR/DISP/X1		
DATA/ADDR/DISP/X2		

17 16 15 14 13 12 11 10

OP1	MOD	OP2	DR
ADDR/DISP/X1			
DATA/ADDR/DISP/X2			

■ 格式 4：操作码扩展指令格式

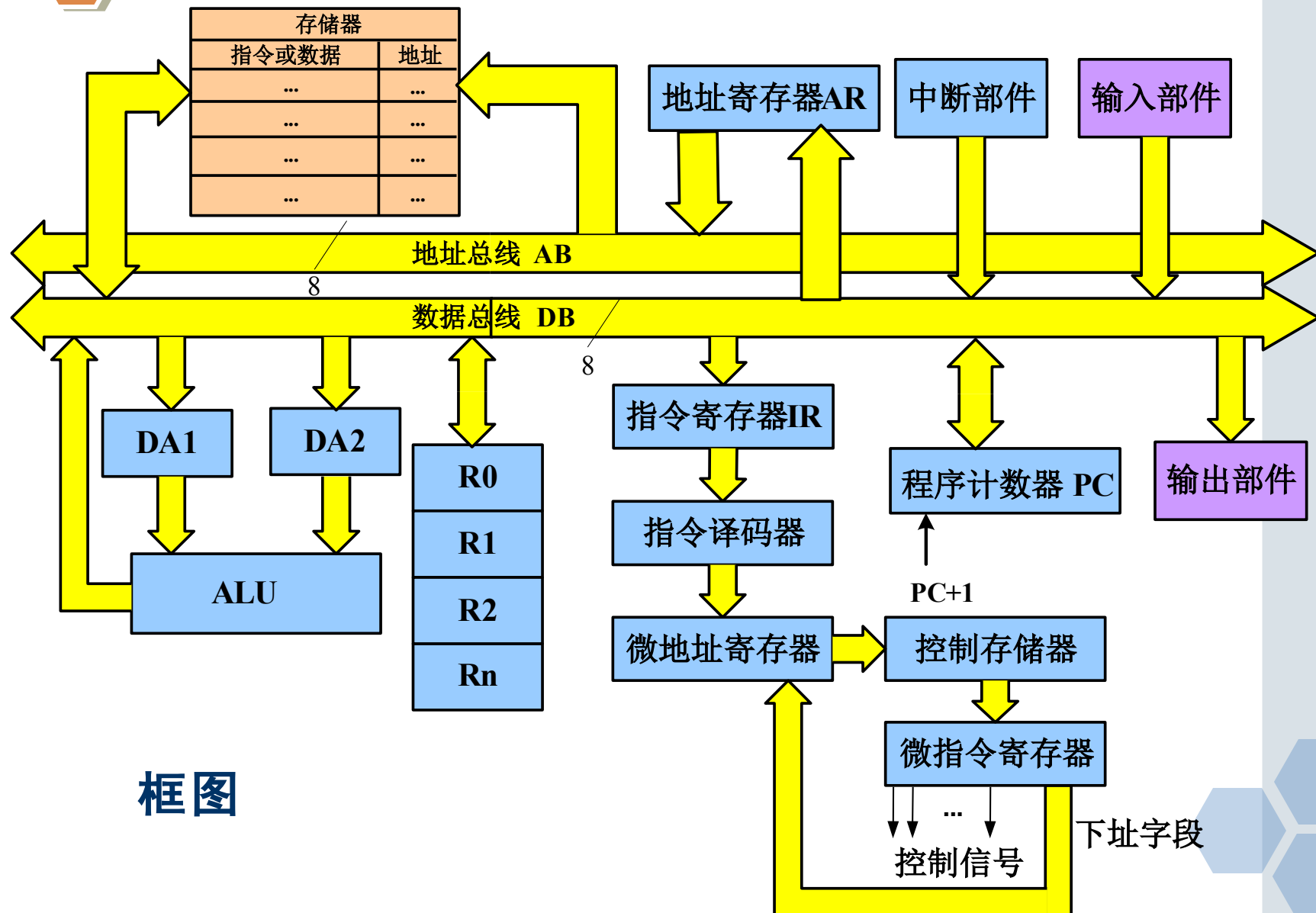
17 16 15 14 13 12 11 10

11	OP	SR/ DR
DATA/ADDR/DISP/X		



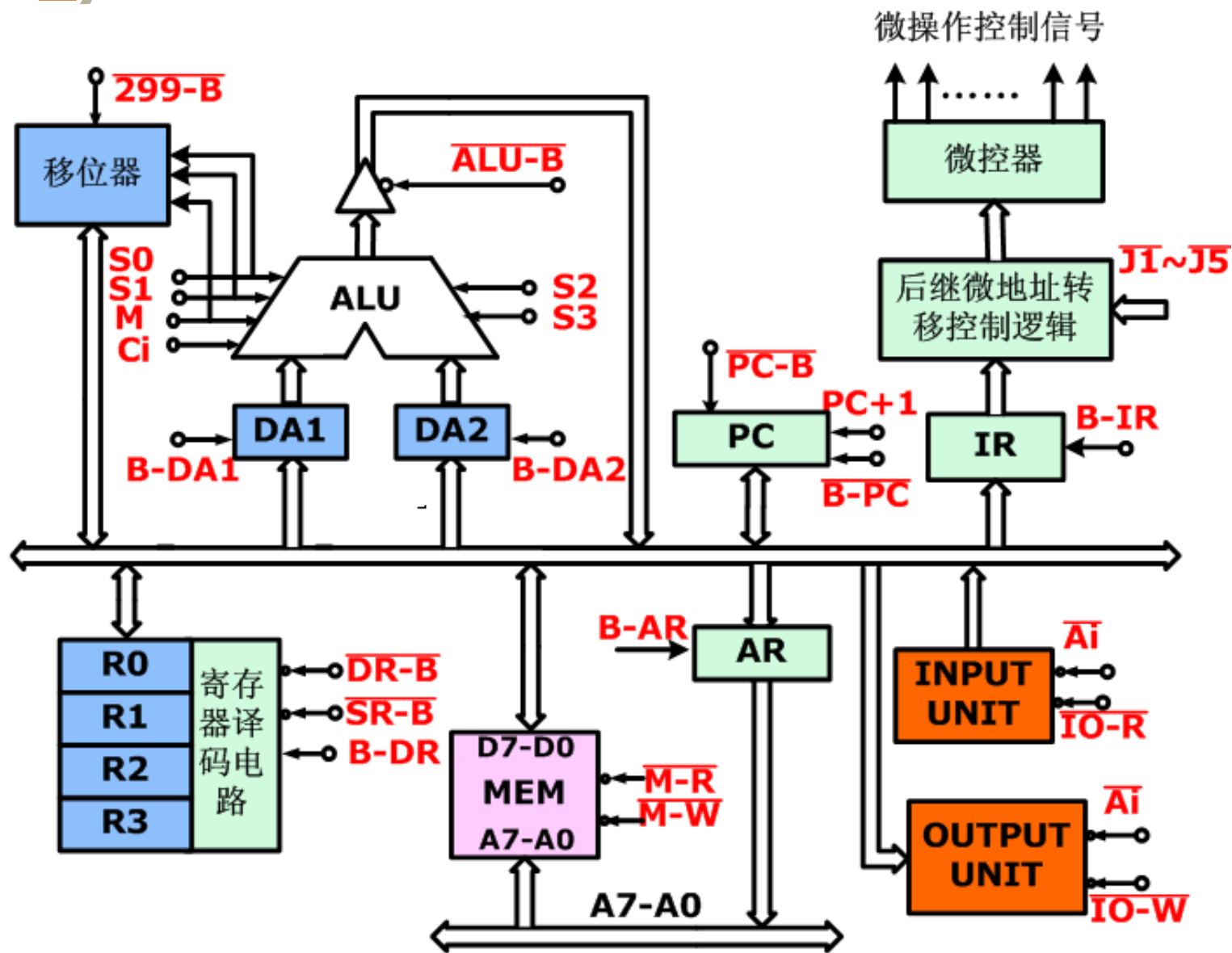


2、模型机系统结构





2、模型机系统结构



具体控制信号



2、模型机系统结构

1. (1) 运算器

- ① **ALU**：两片 74LS181 串连构成，16 种算术运算和 16 种逻辑运算。
- ② **暂存器 DA1 和 DA2**：各由一片 74LS273 构成，暂存送到 ALU 运算的数据。
- ③ **ALU 输出缓冲器**：一片 74LS245 构成，控制 ALU 运算结果是否送总线。
- ④ **状态寄存器**：指示 ALU 运算结果的状态 FC 和 FZ。
- ⑤ **移位器**：一片 74299 构成，有四种移位功能和置数功能。





寄存器译码电路

1. 对于控制器来说，指令 MOV R0, R1 和 MOV

B-R0、B-R1、B-R2、B-R3、R0-B#、R1-B#、R2-B#、R3-B#) 指令，还是 2 条不同的指

2. 微指令发送的是同一段微程序，

微指令发送一样的控制信号。但如何

区分不同的寄存器号？

3. 方法是通过寄存器译码电路，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

寄存器控制信号。

B-DR、DR-B#、SR-B#、SI-B#、SP-B#

I1 I0

I3 I2



寄存器译码电路

1. 输入信号有：

① **B-DR**、**DR-B#**、**SR-B#**、**SI-B#**、**SP-B#**：
来自微指令译码器。

② 指令码 I3-I2：来自指令寄存器的 **SR** 字段

③ 指令码 I1-I0：来自指令寄存器的 **DR** 字段

① 输出信号为：（送至寄存器单元 REG UNIT）

① 寄存器打入脉冲：**B-R0**、**B-R1**、**B-R2**、**B-SP**（**B-R3**）。

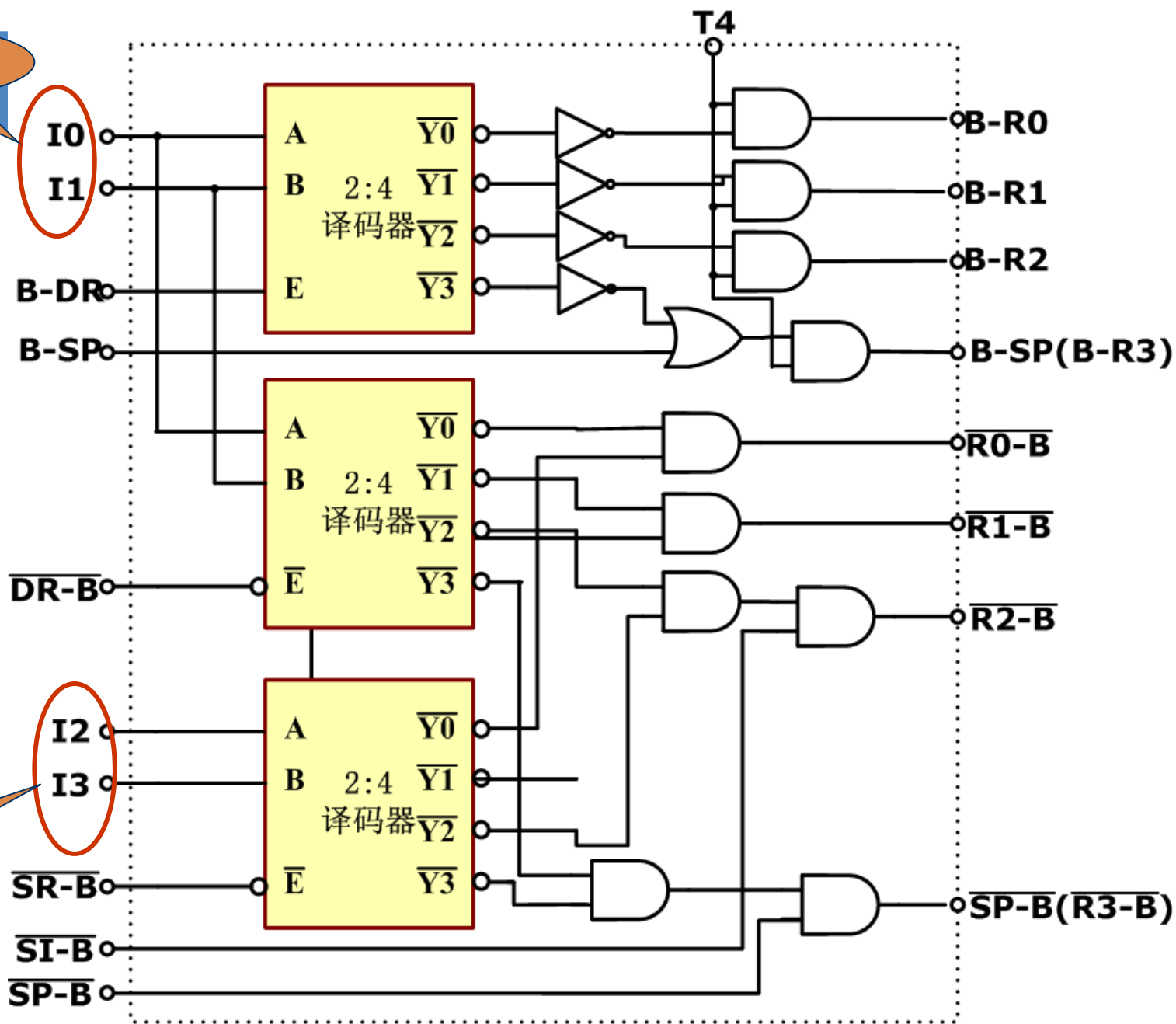
② 寄存器输出控制：**R0-B#**、**R1-B#**、**R2-B#**、**SR-B#**（**R3-B#**）



DR 字
段

寄存器译码电路原理图

SR 字
段





寄存器控制信号的定义

信号	输入 / 输出	解释
I1 I0	输入	指令码中目的寄存器 DR 的编码字段
I3 I2	输入	指令码中源寄存器 SR 的编码字段
B-DR	输入	目的寄存器 DR 的装载控制信号，与指令码 I1I0 一起译码产生 B-R[0:3] 四个信号
DR-B#	输入	目的寄存器 DR 内容送总线的控制信号，与指令码 I1I0 一起译码产生 R[0:3]-B# 四个信号
SR-B#	输入	源寄存器 SR 内容送总线的控制信号，与指令码 I3I2 一起译码产生 R[0:3]-B# 四个信号
RI-B#	输入	变址寄存器 SI 内容送总线的控制信号，即 R2-B#
B-SP	输入	堆栈指针寄存器 SP 的装载控制信号，即 B-R3
SP-B#	输入	SP 内容送总线的控制信号，即 R3-B#



寄存器控制信号的产生逻辑

输入	输出	输入	输出
B-DR=1， 且 I1I0=00	B-R0=1	SR-B#=0， 且 I3I2=00	R0-B#=0
B-DR=1， 且 I1I0=01	B-R1=1	SR-B#=0， 且 I3I2=01	R1-B#=0
B-DR=1， 且 I1I0=10	B-R2=1	SR-B#=0， 且 I3I2=10	R2-B#=0
B-DR=1， 且 I1I0=11	B-R3=1	SR-B#=0， 且 I3I2=11	R3-B#=0
DR-B#=0， 且 I1I0=00	R0-B# =0	SI-B#=0	R2-B#=0
DR-B#=0， 且 I1I0=01	R1-B# =0	B-SP=1	B-R3=1
DR-B#=0， 且 I1I0=10	R2-B#=0	SP-B#=0	R3-B#=0
DR-B#=0， 且 I1I0=11	R3-B#=0		



2、模型机系统结构

1. (2) 存储器

一片 6116 芯片构成，容量 256×8 位。

2. (3) 输入设备

8 位逻辑开关：用于输入二进制程序和数据。

3. (4) 输出设备

8 位 LED 显示灯：用于显示数据。

4. (5) 控制器

① 采用微程序控制器。

② 微指令 24 位，采用下址字段法，下址 7 位。

③ 控存容量： 128×24 位。





3、微指令格式

M23:21 (3)	M20:18 (3)	M17:15 (3)	M14 (1)	M13:8 (6)	M7	M6:0 (7)
BTO	OTB	FUNC	FS	CyCn#	INT_E#	INT_R#
编码 + 译码	BTO	OTB	FUNC			
			FS=1	FS=0		
000	空	空	PC+1	空		
001	B-DA1(t4)	ALU-B# (t2)	J 1# (t2)	M-W# (t3)		
010	B-DA2(t4)	299-B# (t2)	J 2# (t2)	M_R# (t2)		
011	B-IR(t3)	SR-B# (t2)	J 3# (t2)	I/O-W# (t3)		
100	B-DR(t4)	DR-B# (t2)	J 4# (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)			



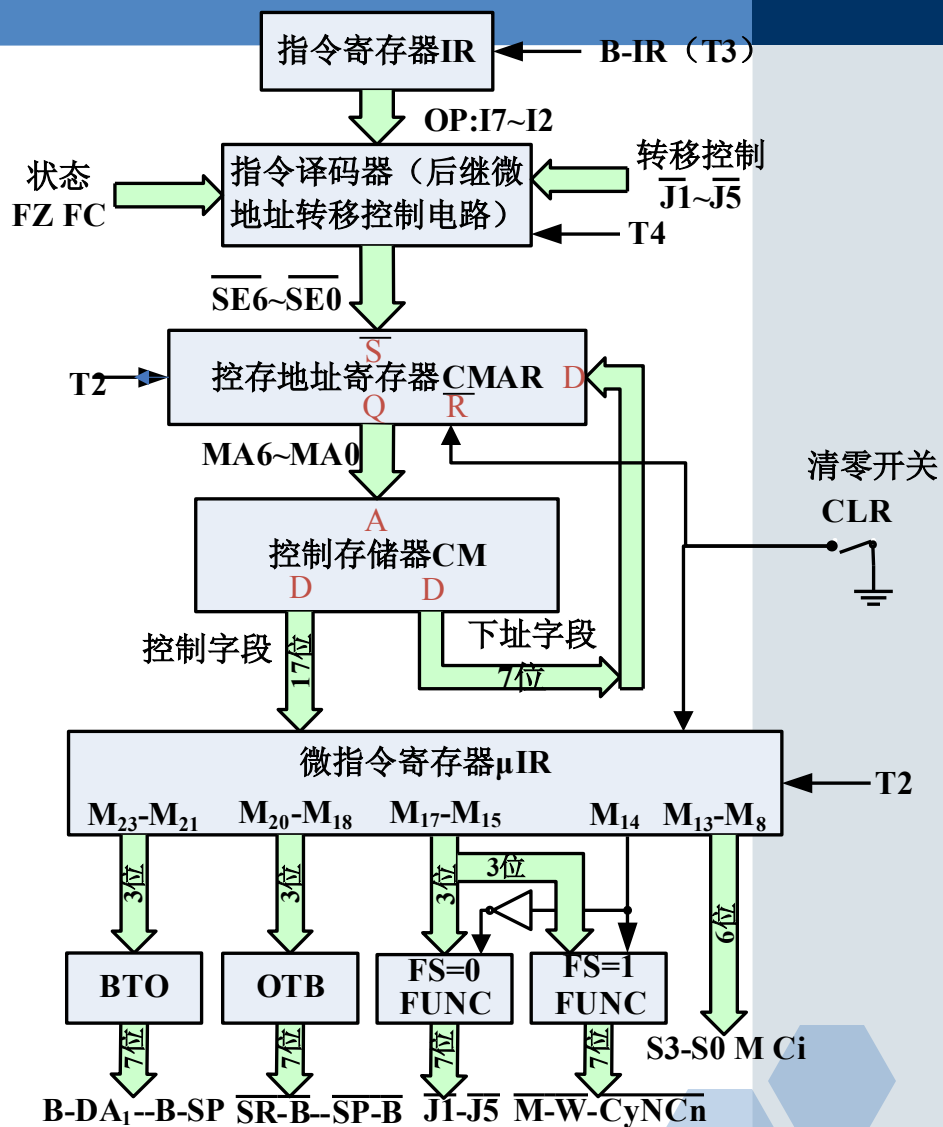
4、微程序控制器

1. 控制存储器 CM：读操作

128×24 位

2. 指令寄存器 IR：T3 打入

读出的 24 位微指令，在每个微周期的 **T2** 节拍：
高 17 位送 μ IR 保存并译码，低 7 位送 CMAR 保存。





4、微程序控制器

3. 指令译码（后继微地址转移控制电路）
J1#--J5#，译码控制
4. 微指令寄存器 μIR 和微指令译码器

