



第六讲

微程序设计技术（二）

— 微指令编码及设计





④ 字段间接编译法

I. 编码方法：某字段的编码含意，除了其本身的编码外，还需要由另一字段来加以解释。也就是说，某一字段所产生的微命令，是和另一字段的代码联合定义出来的。

II 优点：进一步缩短微指令字长的一种编译法

直接控制法

举例：用字段间接编译法，重新定义微指令格式

$M_{15} \sim M_{13}$	$M_{12} \sim M_{11}$	M_{10}	M_9	M_8	M_7	M_6	M_5	M_4	M_3	$M_2 \sim M_0$
BTO	OTB	FUNC	FS	S_3	S_2	S_1	S_0	M	C_i	下址

字段直接编译法

字段间接编译法



用字段间接编译法，重新设计微指

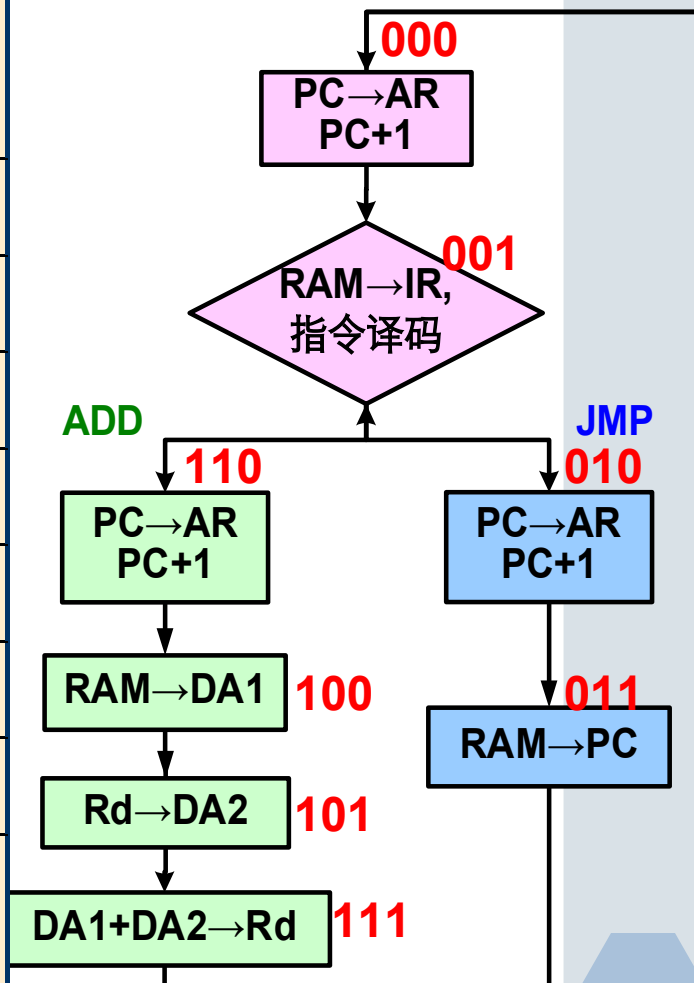
BTO 编码	BTO 信号
000	
001	B-DA1
010	B-DA2
011	B-IR
100	B-AR
101	B-R0
110	M-W#
111	B-PC

OTB 编码	OTB 信号
00	
01	ALU-B#
10	PC-B#
11	M-R#

FUNC 编码	FS=0	FS=1
0	PC+1	
1	J1#	R0-B#

字段间接编译法设计的微程序

微地址	微指令发出的微操作信号	下址字段
000	M0 : PC-B#,B-AR,PC+1	001
001	M1 : M-R# ,B-IR,J1#	xxx
010	JMP•M2 : PC-B#,B-AR,PC+1	011
011	JMP•M3 : M-R#, B-PC#,PC+1	000
100	ADD•M3 : M-R#, B-DA1	101
101	ADD•M4 : R0-B#,B-DA2	111
110	ADD•M2 : PC-B#,B-AR,PC+1	100
111	ADD•M5 : $S_3S_2S_1S_0MC_i=100101$, ALU-B#,B-R0	000





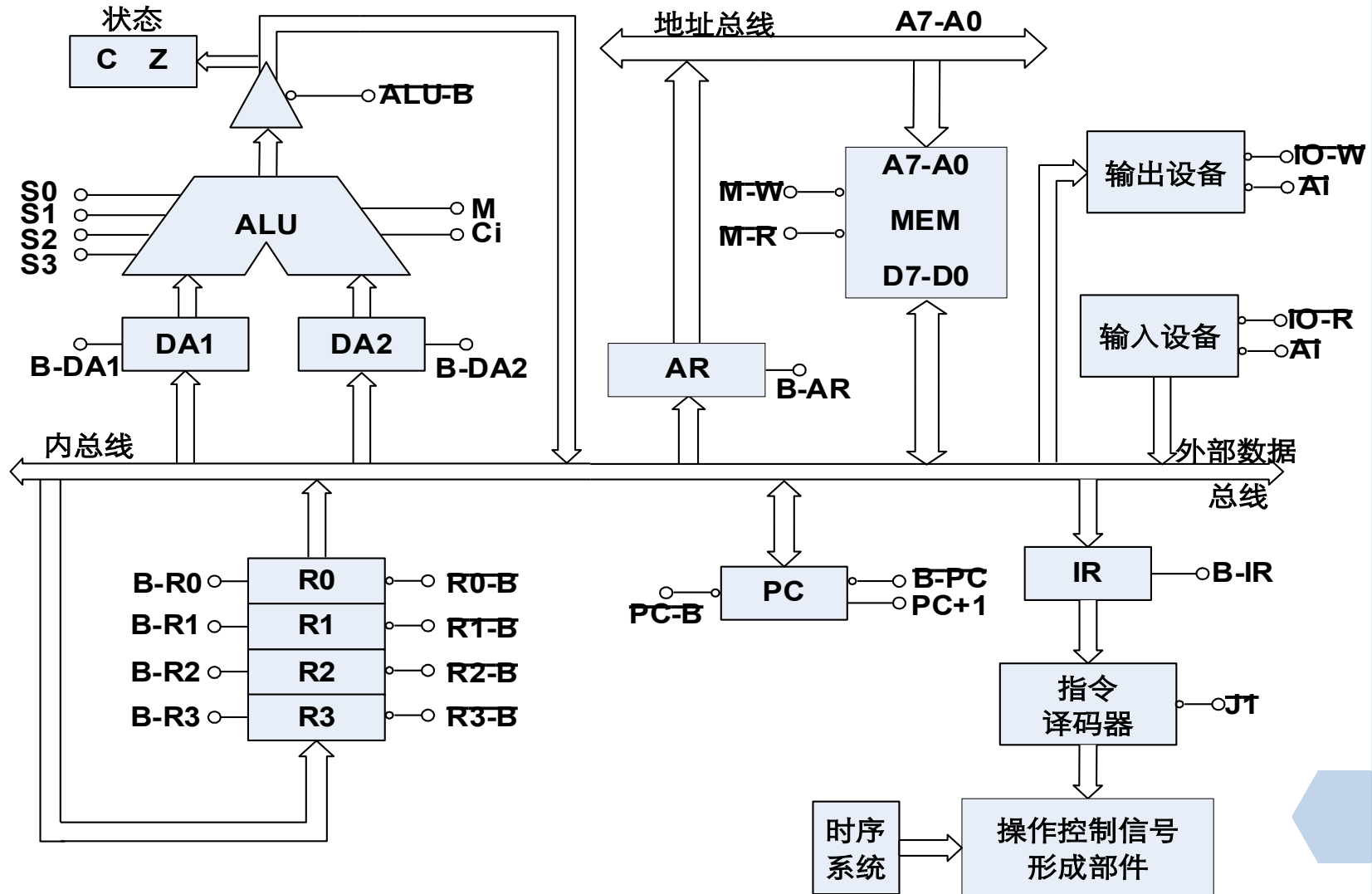
字段间接编译法设计的微程序

微地址	$M_{15} \sim M_{13}$	$M_{12} \sim M_{11}$	M_{10}	M_9	M_8	M_7	M_6	M_5	M_4	M_3	$M_2 \sim M_0$
	BTO	OTB	FUNC	FS	S_3	S_2	S_1	S_0	M	C_i	下址
00H	100	10	0	0	0	0	0	0	0	0	001
01H	011	11	1	0	0	0	0	0	0	0	***
02H	100	10	0	0	0	0	0	0	0	0	011
03H	111	11	0	0	0	0	0	0	0	0	000
04H	001	11	0	1	0	0	0	0	0	0	101
05H	010	00	1	1	0	0	0	0	0	0	111
06H	100	10	0	0	0	0	0	0	0	0	100
07H	101	01	0	1	1	0	0	1	0	1	000



完整的模型机微程序控制器设计

1. 模型机的结构



2. 模型机控制信号（微命令表）

序号	控制信号	功能	序号	控制信号	功能
1	PC-B#	指令地址（程序计数器）送总线	15	ALU-B#	运算器 ALU 内容送总线
2	B-AR	总线内容打入地址寄存器	16	Ci	ALU 进位输入
3	PC+1	程序计数器内容加一	17	B-R0	总线内容打入 R0 寄存器
4	B-PC	总线内容打入程序计数器	18	B-R1	总线内容打入 R1 寄存器
5	B-IR	总线内容打入指令寄存器	19	B-R2	总线内容打入 R2 寄存器
6	M-W#	存储器写	20	B-R3 (B-SP)	总线内容打入 R3 寄存器
7	M-R#	存储器读	21	R0-B#	R0 寄存器内容送总线
8	S ₇	S ₇ - S ₀ 选择 ALU16 种运算之一	22	R1-B#	R1 寄存器内容送总线
9	S ₆	同上	23	R2-B#	R2 寄存器内容送总线
10	S ₅	同上	24	R3-B# (SP-B#)	R4 寄存器内容送总线
11	S ₄	同上	25	I/O-W#	写（输出）I/O 端口
12	M	M 为“1”选择 ALU 做逻辑运算， M 为“0”选择 ALU 做算术运算	26	I/O-R#	读（输入）I/O 端口
13	B-DA1	总线内容打入暂存器 DA1	27	Ai#	端口地址线
14	B-DA2	总线内容打入暂存器 DA2	28	J1#	指令译码器工作



2. 模型机控制信号

❖	17	SR-B#	源寄存器内容送到总线
❖	18	DR→B	目标寄存器的内容送到总线
❖	19	SP-B#	堆栈指示器内容送到总线
❖	20	B_SP	总线上数据打入堆栈指示器
❖	21	SI→B#	变址寄存器内容送到总线
❖	22	B→DR	总线上的数据打入目标寄存器
❖	23	B→SI	总线上的数据打入目标寄存器
❖	24	J2#	译码控制信号
❖	29	J3#	译码控制信号
❖	30	J4#	译码控制信号
❖	31	J5#	译码控制信号
❖	32	CyCn#	进位控制信号
❖	33	CyNCn#	不带进位控制信号
❖	34	INT_R#	中断请求信号
❖	35	INT_E#	中断允许信号



2. 模型机控制信号

SR: R0-R3, 源寄存器

DR: R0-R3, 目标寄存器

SI: (R2) 变址寄存器

J2#: 指令操作码 2 译码控制信号 (低电平有效)

J3#: 进位和零标致译码控制信号 (低电平有效)

J4#: 控制台译码控制信号 (低电平有效)

J5#: 中断控制译码控制信号 (低电平有效)

CyCn#: 进位控制信号 (低电平有效)

CyNCn#: 不带进位控制信号 (低电平有效)

INT R#: 中断请求信号 (低电平有效)



模型机的微指令格式定义 (24)

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

字段间接编
译法





实验模型机微指令字段编码表

字段间接编
译法

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)	

字段直接编译
法

