

计算机组成原理

第六章 中央处理器

6.12 微指令格式

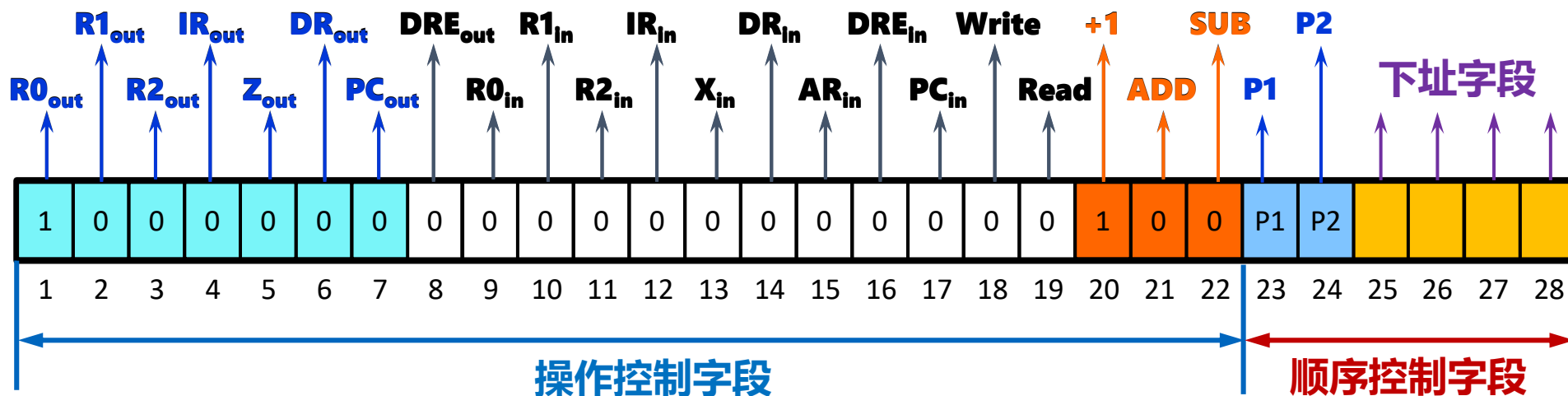
1

微指令设计原则

- 有利于缩短微指令字长度
- 有利于减少控制存储器容量
- 有利于提高微程序执行速度
- 有利于对微指令进行修改
- 有利于提高微程序设计的灵活性

2

微指令格式（直接表示法）



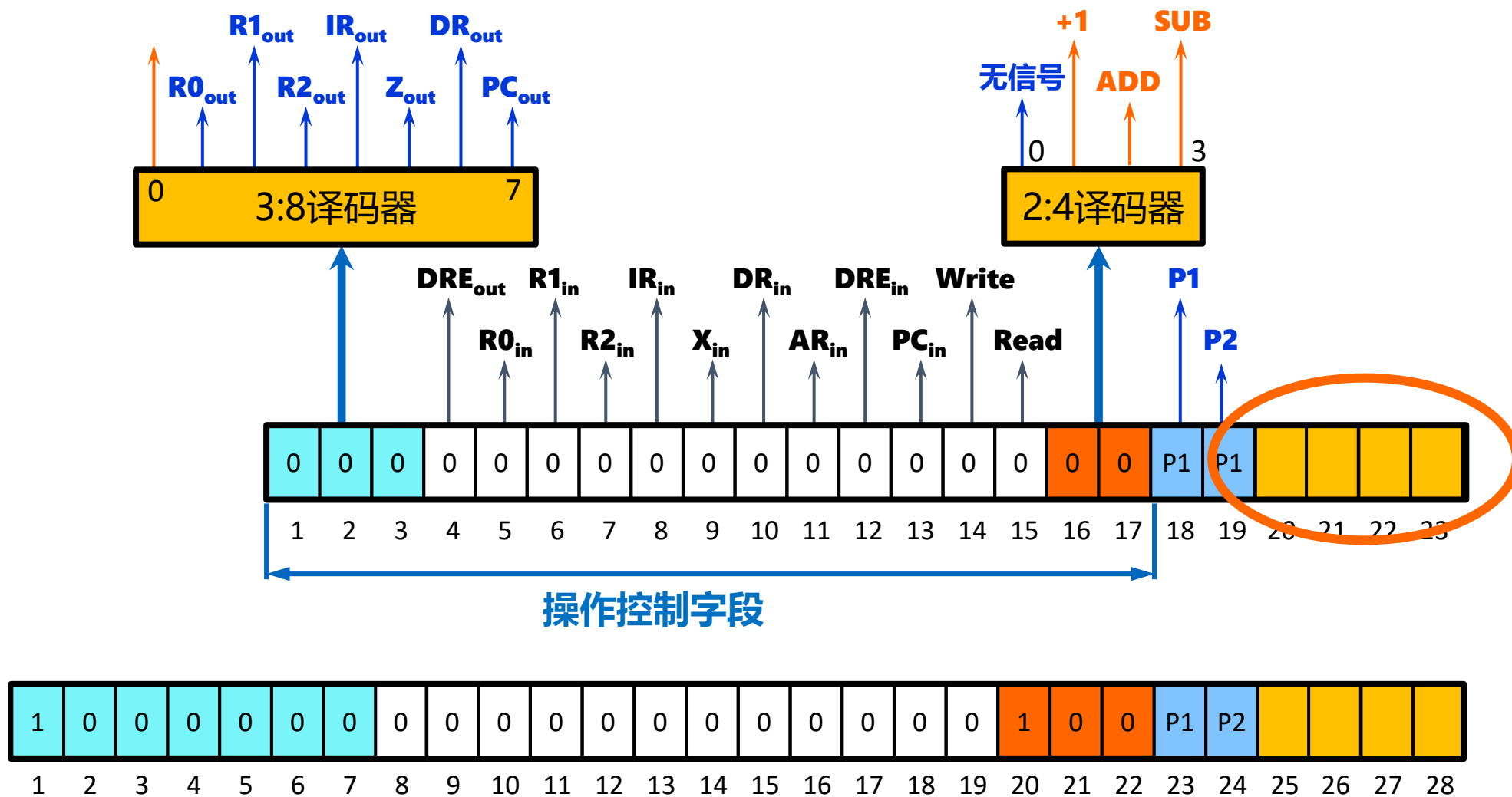
■ 简单直观，便于输出控制，微指令长度太长，控存容量大，如何压缩微指令长度？

◆ 改直接表示为编码表示 （压缩互斥性微指令）

◆ 去掉下址字段，采用 $\mu PC = \mu PC + 1$ 的方式生成微指令地址

◆ 改水平型微指令为垂直型微指令 （牺牲并行性）

- 3 微指令格式（编码表示法） ■ 如果互斥的输出控制信号8个，编码后长度多少？



4

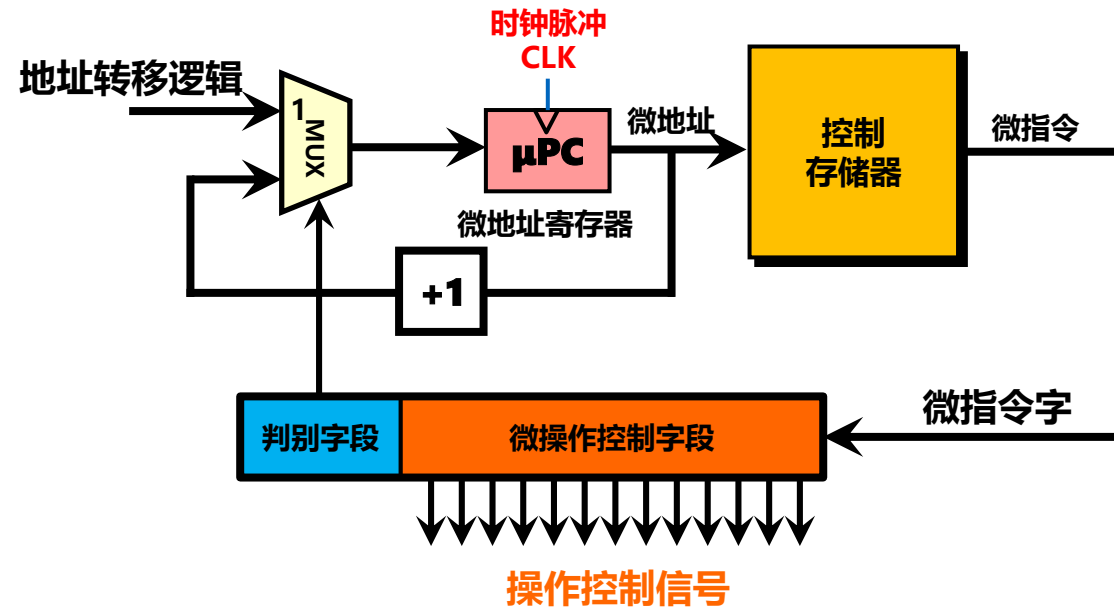
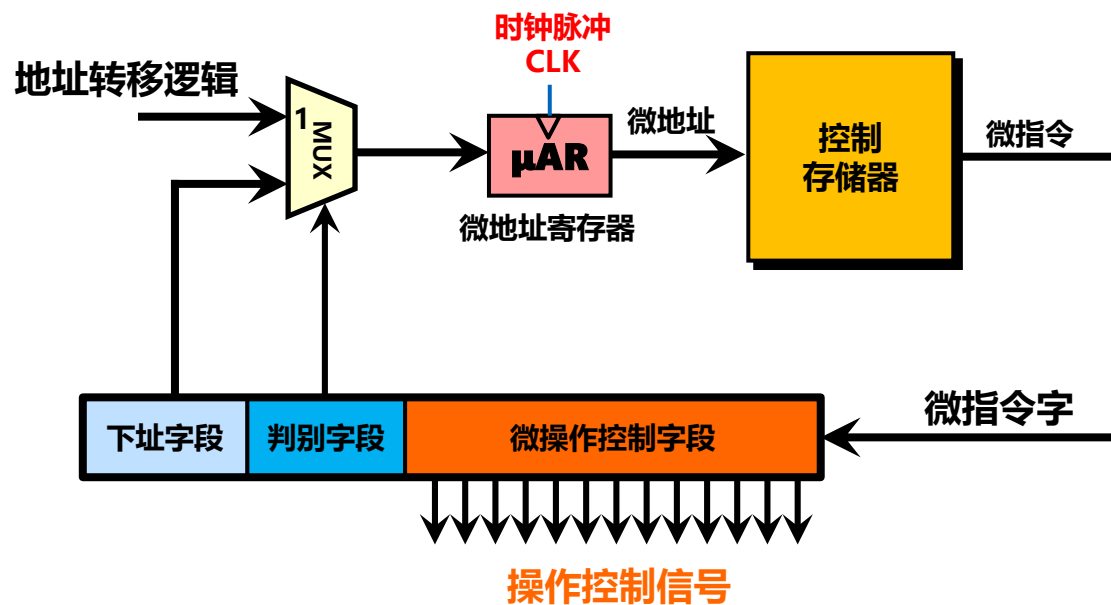
微指令地址形成方法

■ 下地址字段法

微指令长，控存容量大

■ 计数器法 μPC

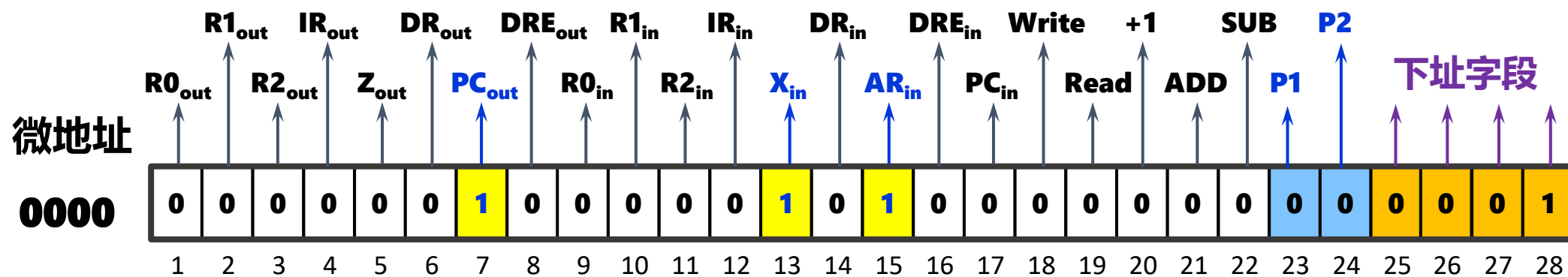
微指令短，需要加法器



5

水平型微指令编码效率

微地址	操作控制字段																顺序控制字段							
0000	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1
0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
0010	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1
0011	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
0100	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0
0110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1
0111	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1000	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1001	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1
1010	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
1011	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1100	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0
1101	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1
1110	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1111	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0



6

垂直型微指令

■ 数据通路分类

- ◆ 寄存器传输

MOV reg1, reg2

MOV AR, PC
- ◆ 运算类型

ALU_OP reg

INC、ADD R1
- ◆ 访存类型

Mem

LOAD、STORE
- ◆ 顺序控制

Branch

Branch P1

如采用扩展操作码，需8位字长

MOV	源寄存器(3)	目的寄存器(3)
INC	无操作数	
ADD	源寄存器编号 (3位)	
Mem	Read/Write	
Branch	判断条件	

节拍	取指令	LOAD	MOVE	ADD	STORE	JMP
T1	(PC)→AR, (PC)→X	(IR _A)→AR, (PC)→X	(IR _A) → R[0]	(R0)→X	(R2)→AR	(IR _A)→PC
T2	(X)+1→Z Read Mem	Read Mem		(X)+(R1)→Z	(R0)→DR	
T3	(Z)→PC, Mem[AR]→DR	Mem[AR]→DR		(Z)→R0	(DR)→Mem[AR]	
T4	(DR)→IR	(DR)→R0				

7

垂直型微指令实例

LOAD指令微程序容量:

■ 直接表示法 + 下址字段

◆ 28位*8=224位

■ 编码表示法 + μ PC

◆ 19位*8=152位

■ 垂直微指令

◆ 8位*14=112位

◆ 容量小，速度慢，指令执行需译码电路

节拍	LOAD指令	#	LOAD	信号
T1	(PC)→AR, (PC)→X	C1	MOV AR,PC	PC _{out} , AR _{in}
		C2	MOV X,PC	PC _{out} , X _{in}
T2	(X)+1→Z Read Mem	C3	LOAD	DRE _{in} ,Read
		C4	LOAD	DRE _{in} ,Read
		C5	INC	+1
T3	(Z)→PC, Mem[AR]→DR	C6	MOV PC,Z	Z _{out} , PC _{in}
T4	(DR)→IR	C7	MOV IR,DR	DR _{out} , IR _{in}
		C8	Branch P1	P1
T5	(IR _A)→AR, (PC)→X	C9	MOV AR,IR	IR _{out} ,AR _{in}
		C10	MOV X, PC	PC _{out} , X _{in}
T6	Read Mem	C11	LOAD	DRE _{in} ,Read
T7	Mem[AR]→DR	C12	LOAD	DRE _{in} ,Read
T8	(DR)→R0	C13	MOV R0, DR	DR _{out} , R0 _{in}
		C14	Branch	

8

水平型与垂直型微指令

■ 水平型微指令

- ◆ 并行操作能力强，效率高，灵活性强，
- ◆ 微指令字较长，微程序短，控存容量大，性能佳

■ 垂直型微指令

- ◆ 字长短，微程序长，控存容量小，性能差
- ◆ 垂直型与指令相似，易于掌握
- ◆ 基本被淘汰

谢谢!

