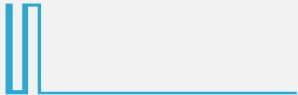




第九章 中央处理器(五)

秦磊华 计算机学院



9.7 微程序控制器设计

9.8 控制器设计总结

1. 微程序控制器的基本思想

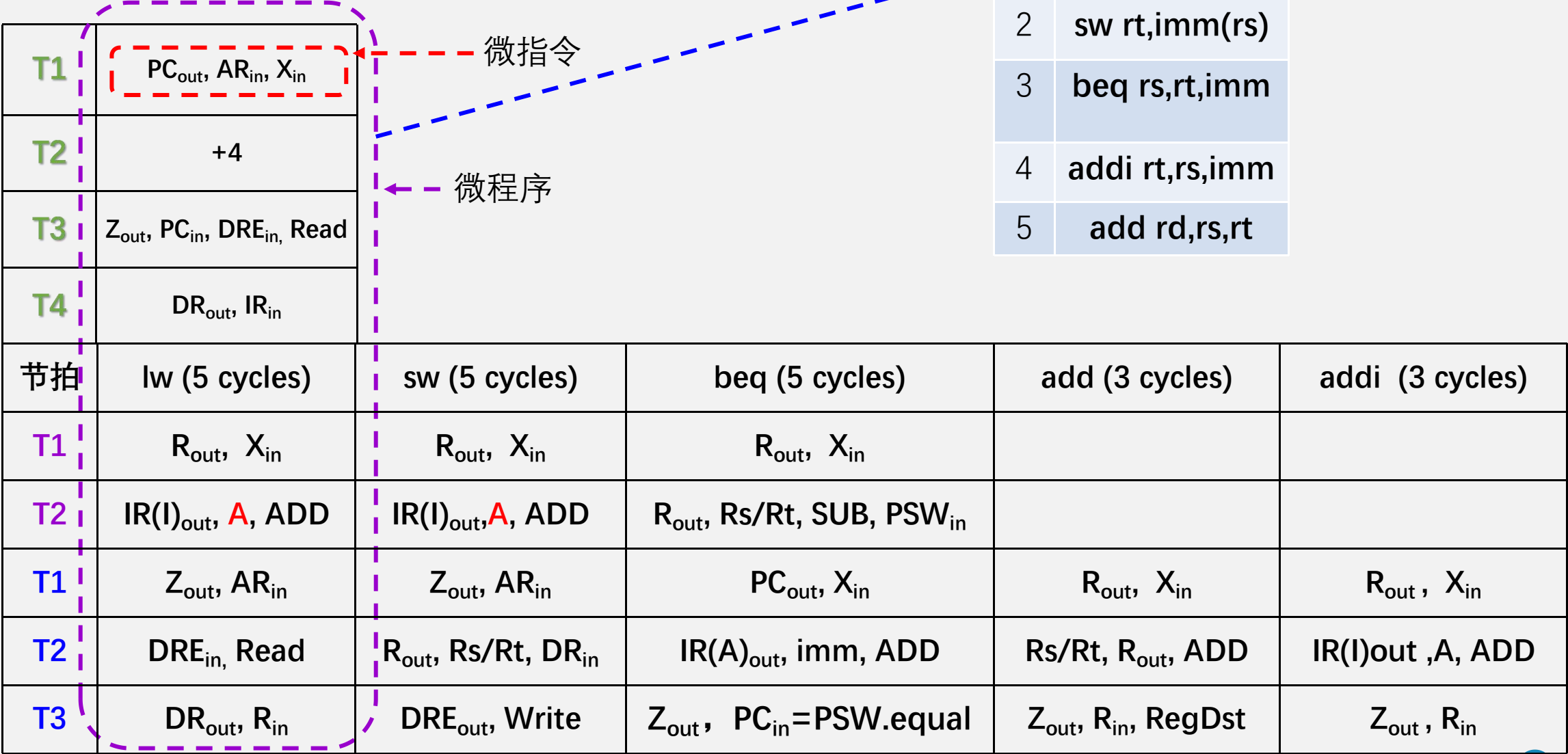
- ◆微程序控制器采用存储逻辑;
- ◆采用软件方法设计硬件的技术;

节拍	控制信号(4 cycles)
T1	PC _{out} , AR _{in} , X _{in}
T2	+4, Read
T3	Z _{out} , PC _{in} , DRE _{in} , Read
T4	DR _{out} , IR _{in}

◆仿照程序设计方法，把每条指令所需的操作控制信号编写成微指令，存放到只读存储器(控存)中。每条机器指令对应一段微程序，机器执行程序时依次读出指令所对应的微指令,执行其中的微操作,从而完成指令的功能。

9.7 微程序控制器设计

1. 微程序控制器的基本思想



2. 指令与微指令

指令格式与作用:

操作码字段	寻址方式	地址码字段
-------	------	-------

微指令作用?

节拍	控制信号(4 cycles)
T1	$PC_{out}, AR_{in}, X_{in}$
T2	$+4, Read$
T3	$Z_{out}, PC_{in}, DRE_{in}, Read$
T4	DR_{out}, IR_{in}

产生指令周期某段时间内的操作控制信号

微指令的格式?

微指令的格式与什么有关?

2. 指令与微指令

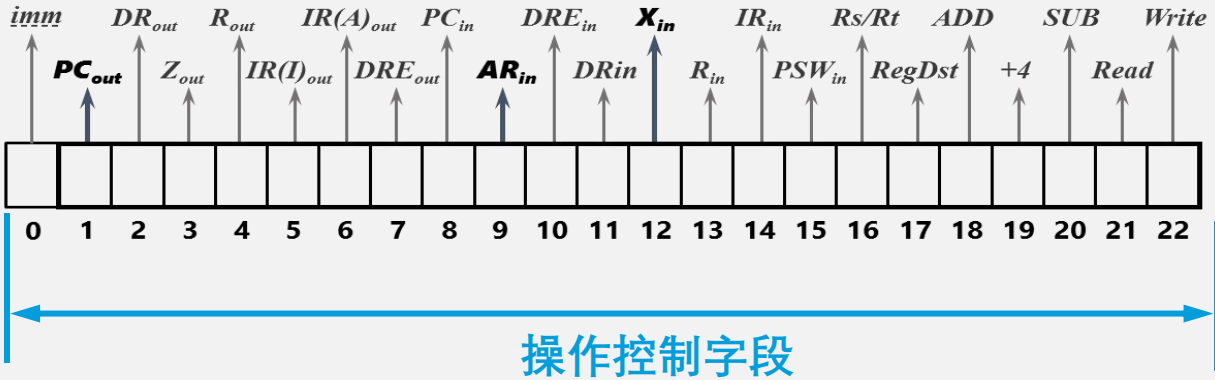
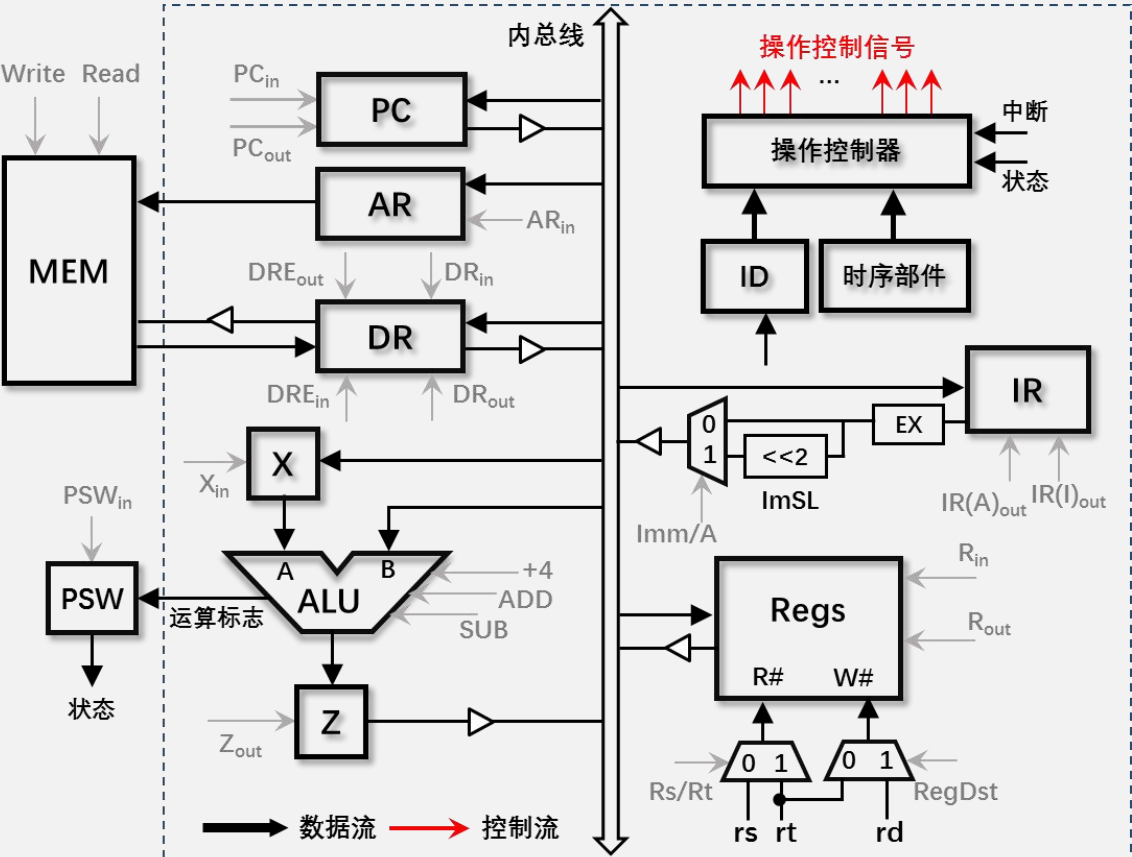
微指令的格式与什么有关？

节拍	lw (5 cycles)	sw (5 cycles)	beq (5 cycles)	add (3 cycles)	addi (3 cycles)
T1	R _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}		
T2	IR(I) _{out} , A, ADD	IR(I) _{out} , A, ADD	R _{out} , Rs/Rt, SUB, PSW _{in}		
T1	Z _{out} , AR _{in}	Z _{out} , AR _{in}	PC _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}
T2	DRE _{in} , Read	R _{out} , Rs/Rt, DR _{in}	IR(A) _{out} , imm, ADD	Rs/Rt, R _{out} , ADD	IR(I) _{out} , A, ADD
T3	DR _{out} , R _{in}	DRE _{out} , Write	Z _{out} , PC _{in} =PSW.equal	Z _{out} , R _{in} , RegDst	Z _{out} , R _{in}



9.7 微程序控制器设计

3. 微指令格式



承载操作控制信号，每一位对应一个控制信号，也称微命令

3. 微指令格式

T1	PC _{out} , AR _{in} , X _{in}
T2	+4
T3	Z _{out} , PC _{in} , DRE _{in} , Read
T4	DR _{out} , IR _{in}

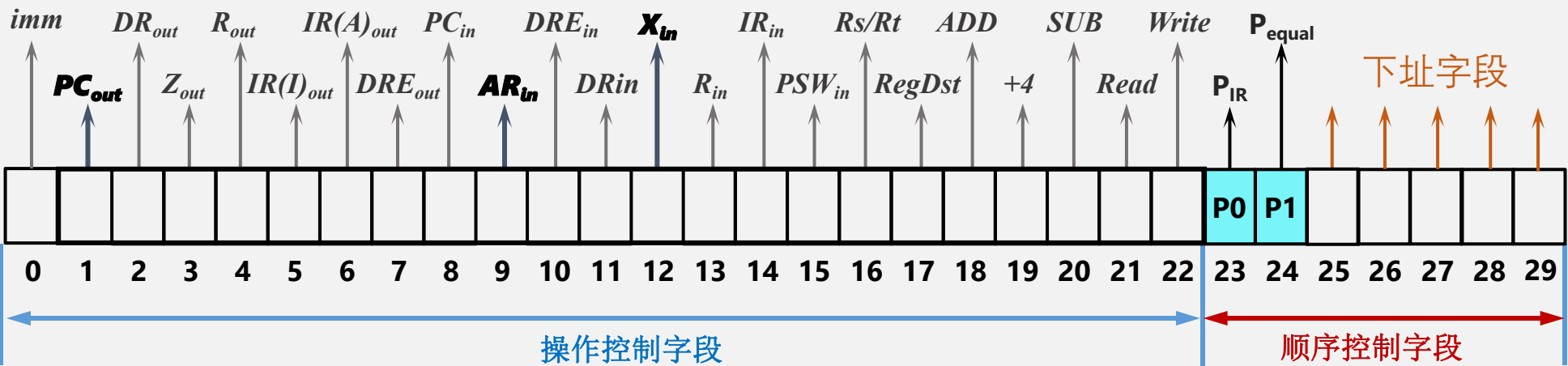
与每条指令执行阶段的微指令一同编制在同一微程序中还是独立为一段微程序？

节拍	lw (5 cycles)	sw (5 cycles)	beq (5 cycles)	add (3 cycles)	addi (3 cycles)
T1	R _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}		
T2	IR(I) _{out} , A, ADD	IR(I) _{out} , A, ADD	R _{out} , Rs/Rt, SUB, PSW _{in}		
T1	Z _{out} , AR _{in}	Z _{out} , AR _{in}	PC _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}
T2	DRE _{in} , Read	R _{out} , Rs/Rt, DR _{in}	IR(A) _{out} , imm, ADD	Rs/Rt, R _{out} , ADD	IR(I) _{out} , A, ADD
T3	DR _{out} , R _{in}	DRE _{out} , Write	Z _{out} , PC _{in} =PSW.equal	Z _{out} , R _{in} , RegDst	Z _{out} , R _{in}



9.7 微程序控制器设计

3. 微指令格式

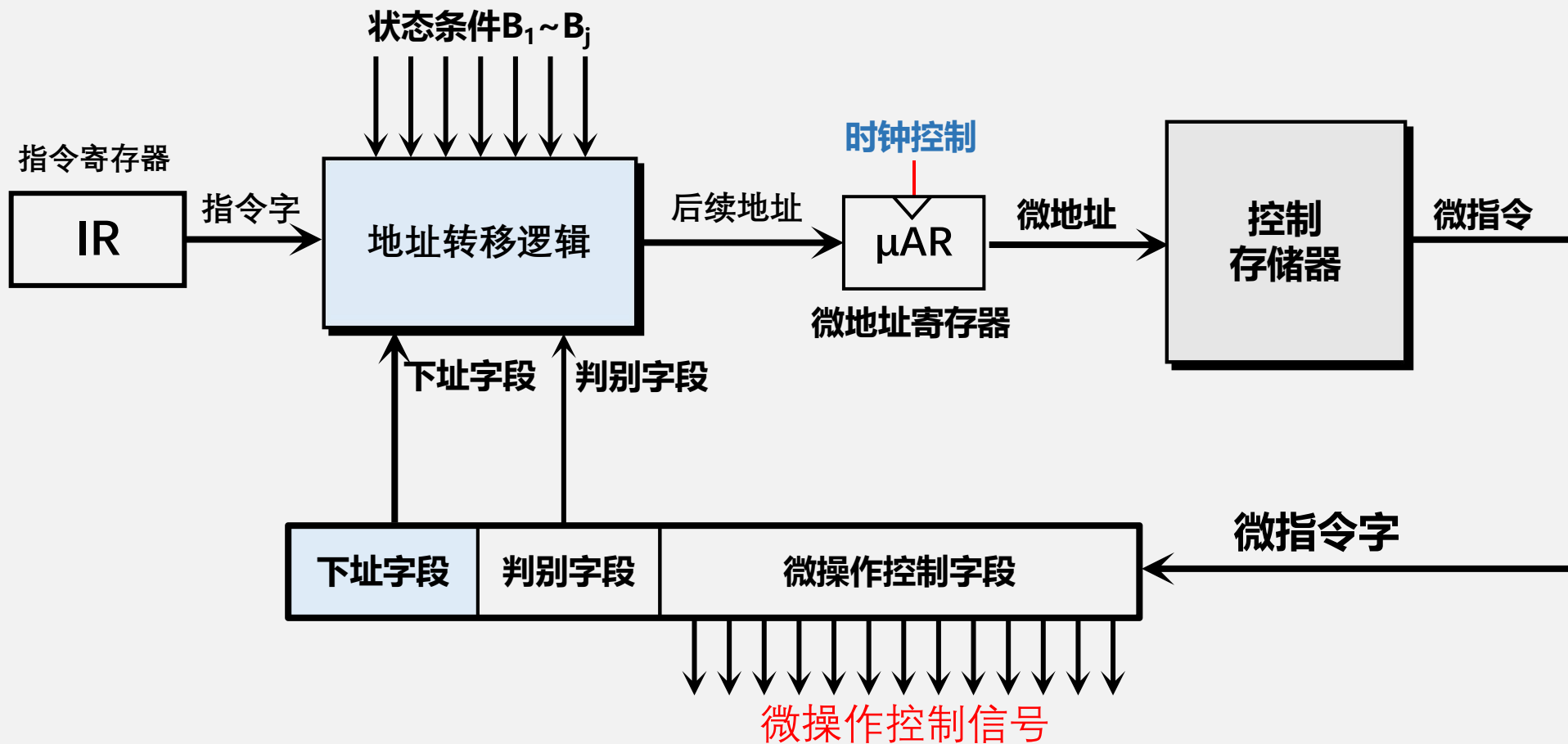


顺序控制字段: 用于控制微程序中微指令的执行顺序

判别测试字段: 取指转移、测试条件状态寄存器中的相关条件

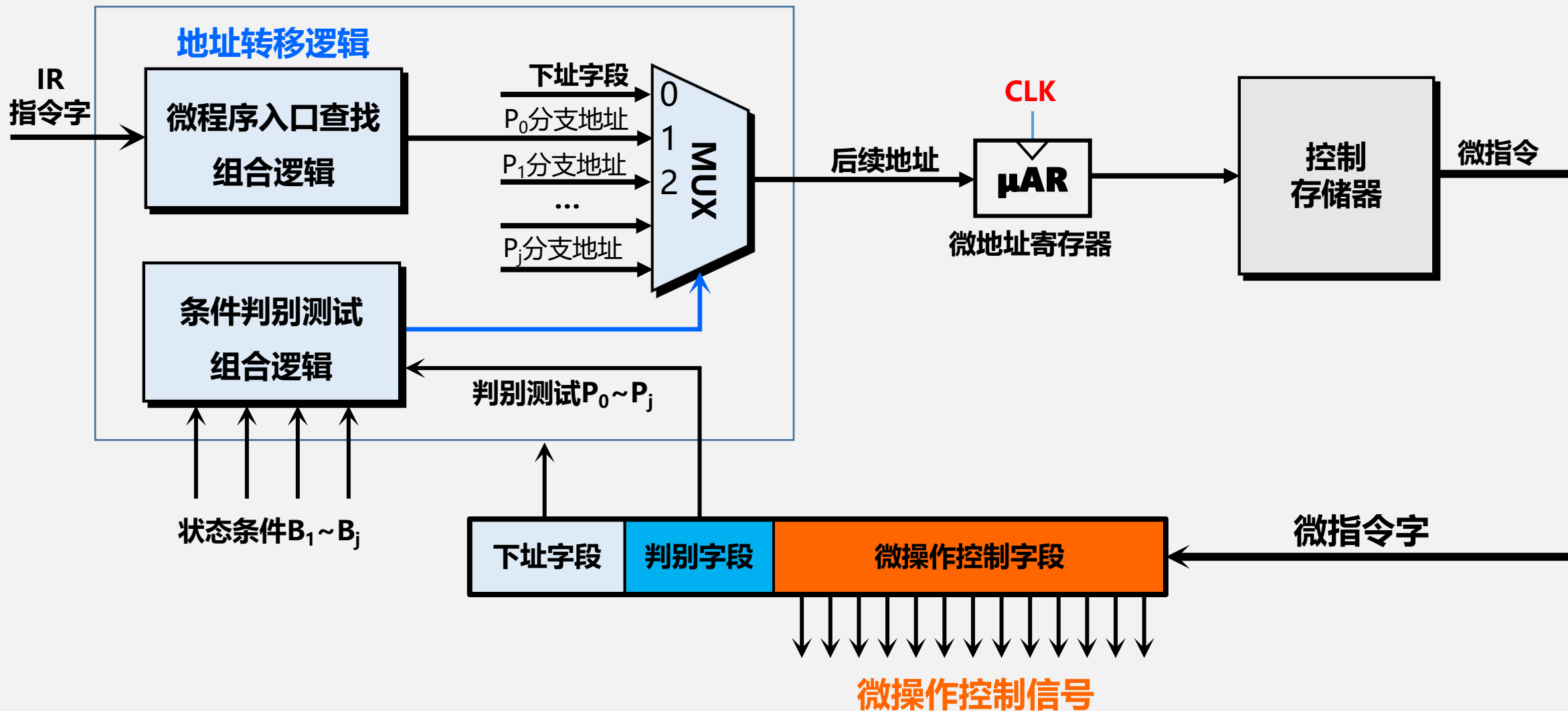
9.7 微程序控制器设计

4. 微程序控制器组成原理框图(下址字段法)



9.7 微程序控制器设计

4. 微程序控制器组成原理框图(下址字段法)



5. 微指令编码

T1	PC _{out} , AR _{in} , X _{in}
T2	+4
T3	Z _{out} , PC _{in} , DRE _{in} , Read
T4	DR _{out} , IR _{in}

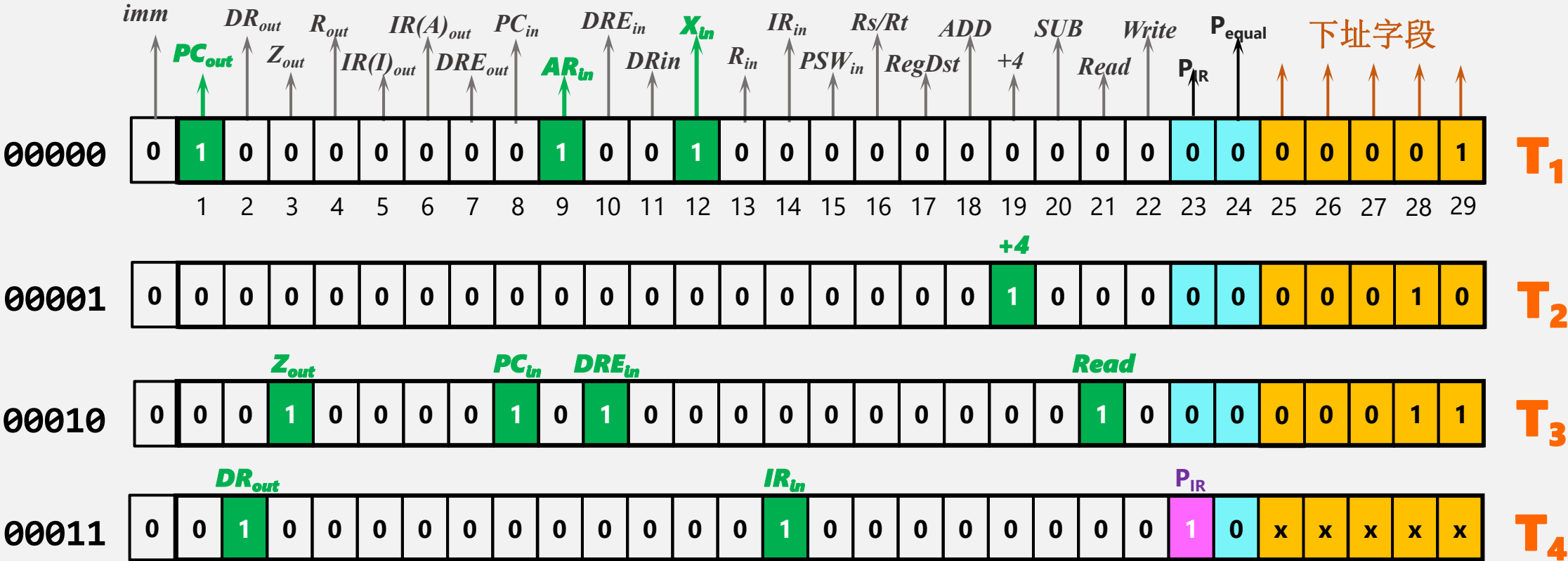
- ◆ 一个时钟周期内的操作安排一条微指令
- ◆ 微指令操作控制字段的值为1表示在该时钟周期内有效
- ◆ 指令需多少时钟周期则对应微程序包括多少条微指令

节拍	lw (5 cycles)	sw (5 cycles)	beq (5 cycles)	add (3 cycles)	addi (3 cycles)
T1	R _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}		
T2	IR(I) _{out} , A, ADD	IR(I) _{out} , A, ADD	R _{out} , Rs/Rt, SUB, PSW _{in}		
T1	Z _{out} , AR _{in}	Z _{out} , AR _{in}	PC _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}
T2	DRE _{in} , Read	R _{out} , Rs/Rt, DR _{in}	IR(A) _{out} , imm, ADD	Rs/Rt, R _{out} , ADD	IR(I) _{out} , A, ADD
T3	DR _{out} , R _{in}	DRE _{out} , Write	Z _{out} , PC _{in} =PSW.equal	Z _{out} , R _{in} , RegDst	Z _{out} , R _{in}

9.7 微程序控制器设计

5. 微指令编码(取指)

节拍	数据通路 (数据流)	控制信号(控制流)
T1	PC→AR, PC→X	PC _{out} , AR _{in} , X _{in}
T2	X+4→Z	+4
T3	Z→PC, M[AR]→DR	Z _{out} , PC _{in} , DRE _{in} , Read
T4	DR→IR	DR _{out} , IR _{in}

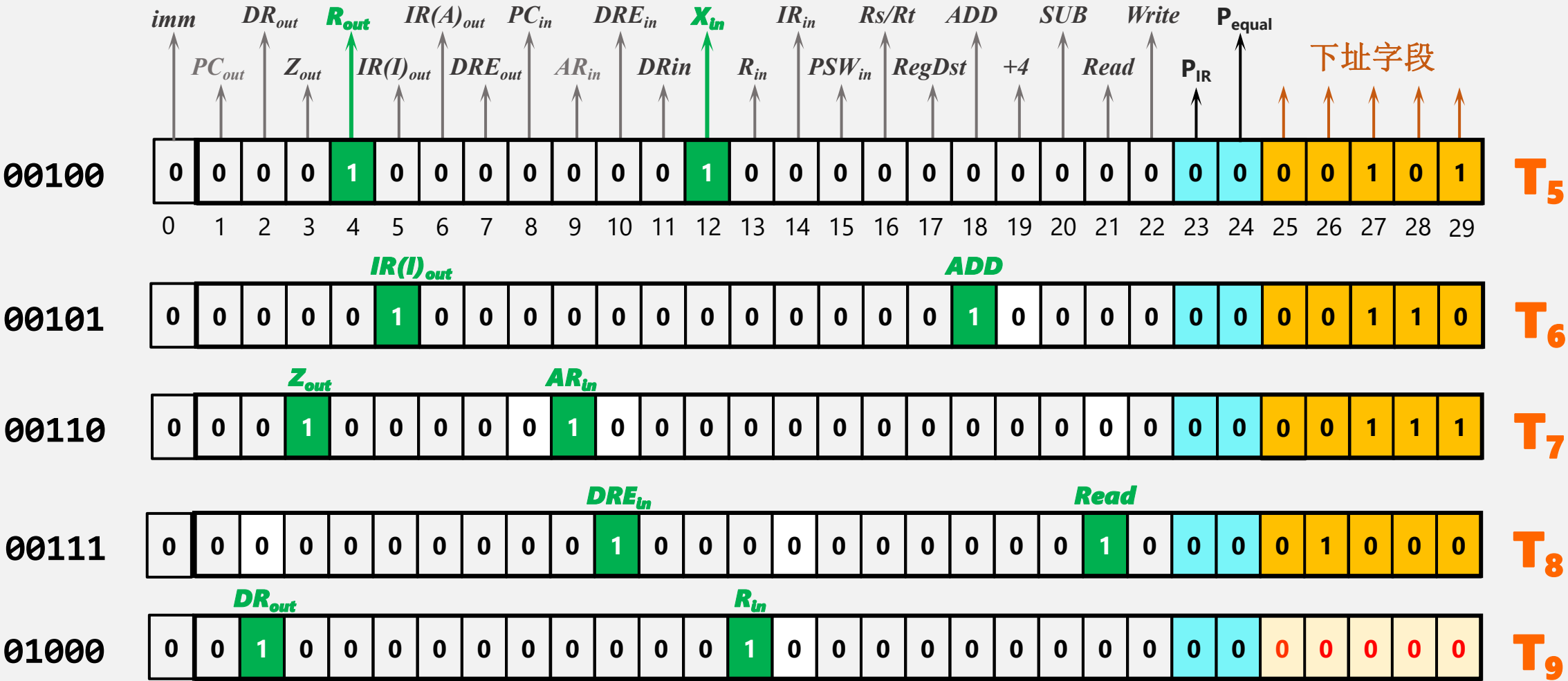




9.7 微程序控制器设计

5. 微指令编码(LW)

R_{out}, X_{in}	$IR(I)_{out}, A, ADD$	Z_{out}, AR_{in}	$DRE_{in}, Read$	DR_{out}, R_{in}
-------------------	-----------------------	--------------------	------------------	--------------------

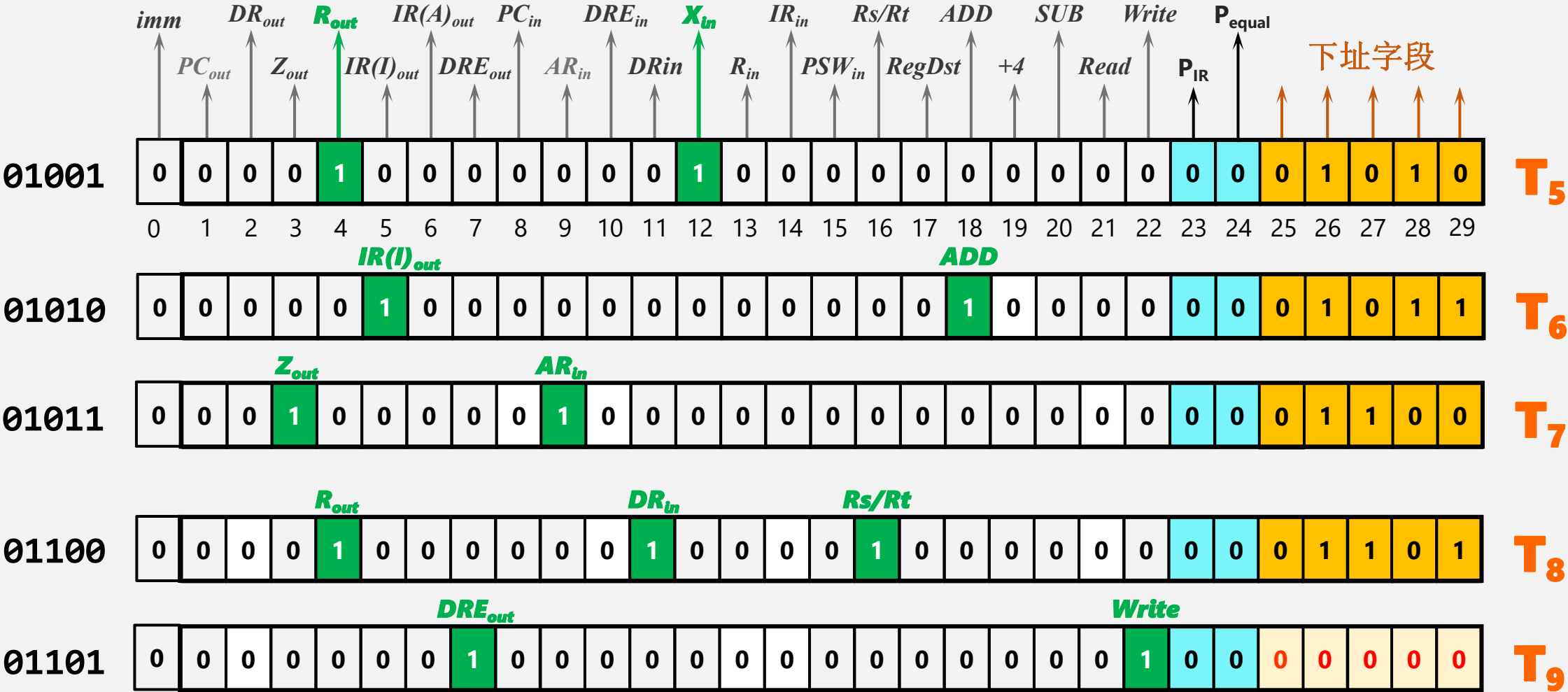




9.7 微程序控制器设计

5. 微指令编码(SW)

R_{out}, X_{in}	$IR(I)_{out}, A, ADD$	Z_{out}, AR_{in}	$R_{out}, Rs/Rt, DR_{in}$	$DRE_{out}, Write$
-------------------	-----------------------	--------------------	---------------------------	--------------------

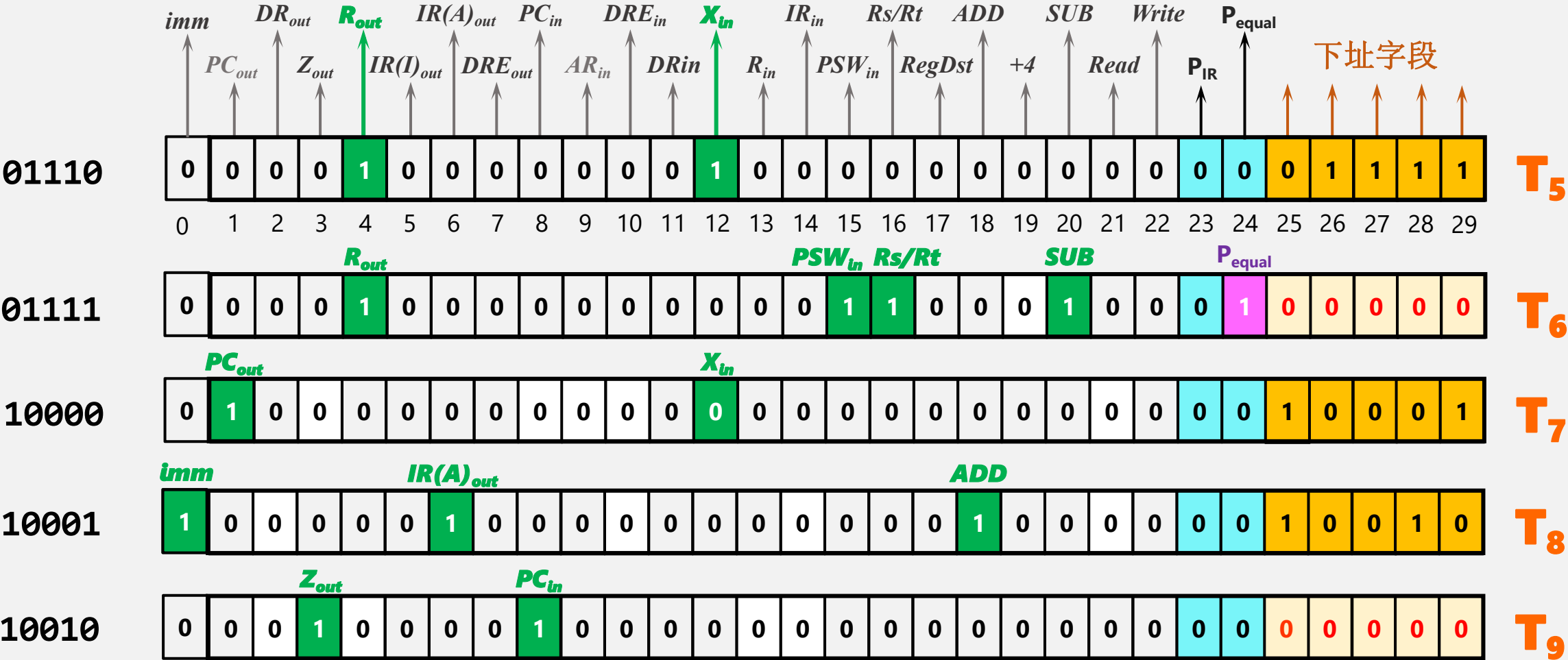




9.7 微程序控制器设计

5. 微指令编码(Beq)

R_{out}, X_{in}	$R_{out}, Rs/Rt, SUB, PSW_{in}$	PC_{out}, X_{in}	$IR(A)_{out}, imm, ADD$	$Z_{out}, PC_{in}=PSW.equal$
-------------------	---------------------------------	--------------------	-------------------------	------------------------------

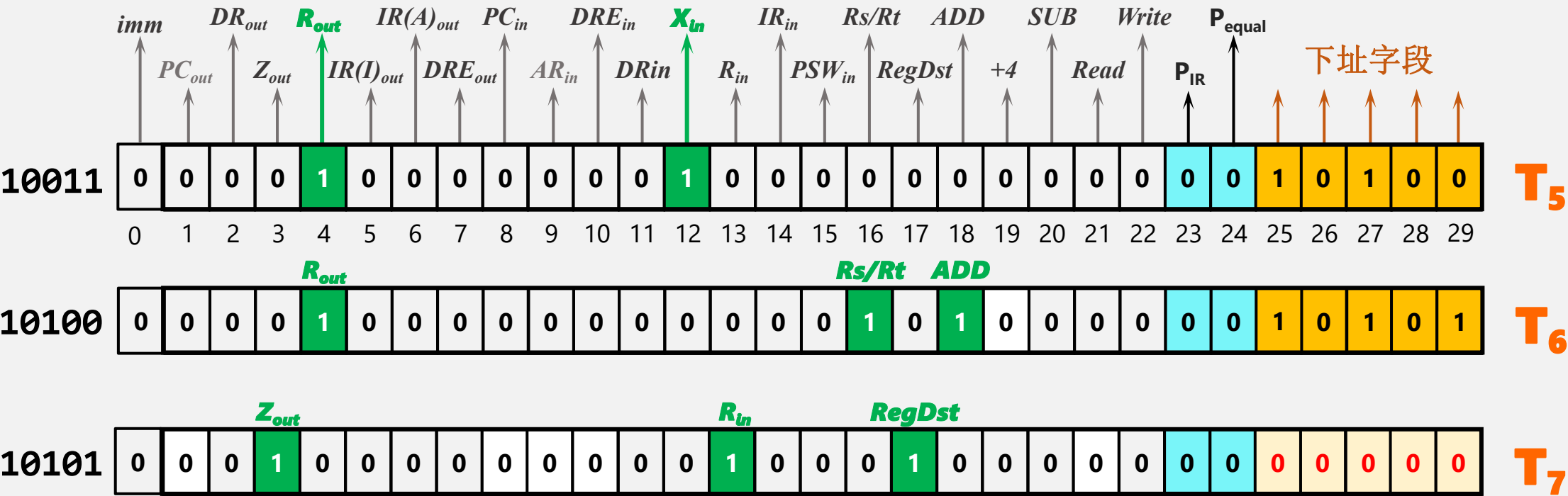




9.7 微程序控制器设计

5. 微指令编码(add)

R_{out}, X_{in}	$Rs/Rt, R_{out}, ADD$	$Z_{out}, R_{in}, RegDst$
-------------------	-----------------------	---------------------------

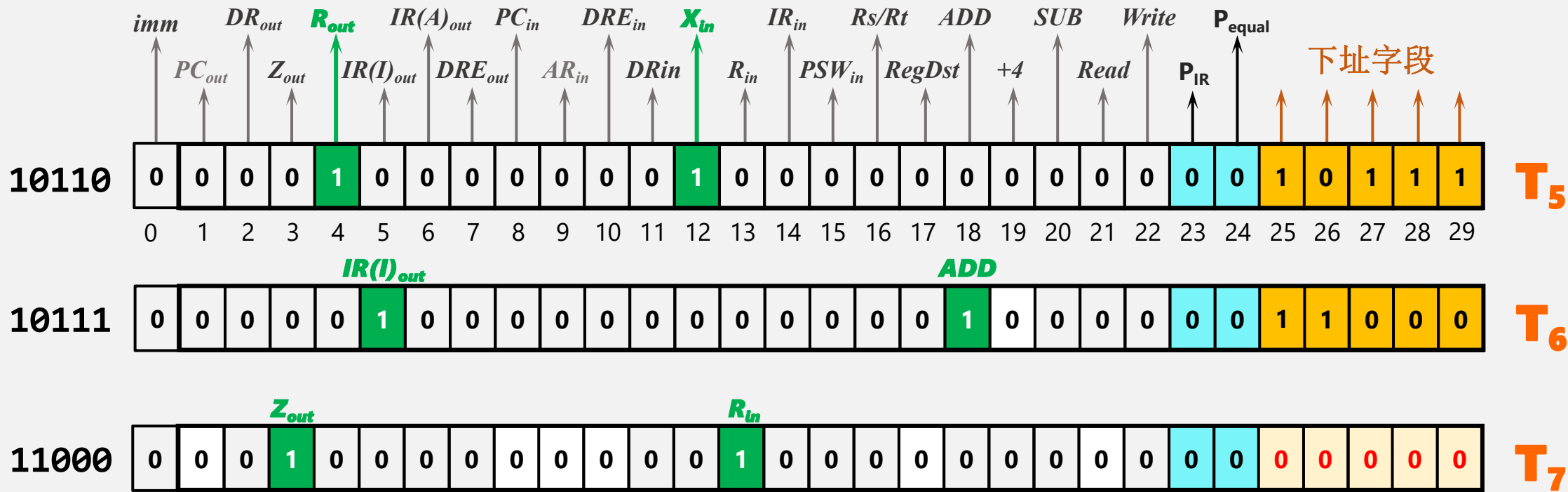




9.7 微程序控制器设计

5. 微指令编码(addi)

R_{out}, X_{in}	$IR(I)_{out}, A, ADD$	Z_{out}, R_{in}
-------------------	-----------------------	-------------------



9.7 微程序控制器设计

微地址与状态等价!

状态	微地址	操作控制字段																顺序控制字段										
0	00000	1							1			1										0	0	0	0	1	取指微程序	
1	00001															1						0	0	0	1	0		
2	00010			1					1		1							1				0	0	0	1	1		
3	00011		1											1					1			1						
4	00100				1								1										0	0	1	0	1	Lw
5	00101					1										1							0	0	1	1	0	
6	00110			1						1													0	0	1	1	1	
7	00111										1									1			0	1	0	0	0	
8	01000		1											1									0	0	0	0	0	SW
9	01001				1								1										0	1	0	1	0	
10	01010					1										1							0	1	0	1	1	
11	01011			1						1													0	1	1	0	0	
12	01100				1							1				1							0	1	1	0	1	beq
13	01101						1											1					0	0	0	0	0	
14	01110				1								1										0	1	1	1	1	
15	01111				1									1	1				1				0	0	0	0	0	
16	10000	1											1										1	0	0	0	1	add
17	10001						1										1						1	0	0	1	0	
18	10010			1					1														0	0	0	0	0	
19	10011				1								1										1	0	1	0	0	
20	10100				1										1		1						1	0	1	0	1	addi
21	10101			1									1				1						0	0	0	0	0	
22	10110				1								1										1	0	1	1	1	
23	10111					1											1						1	1	0	0	0	
24	11000			1									1										0	0	0	0	0	

取指微程序

Lw

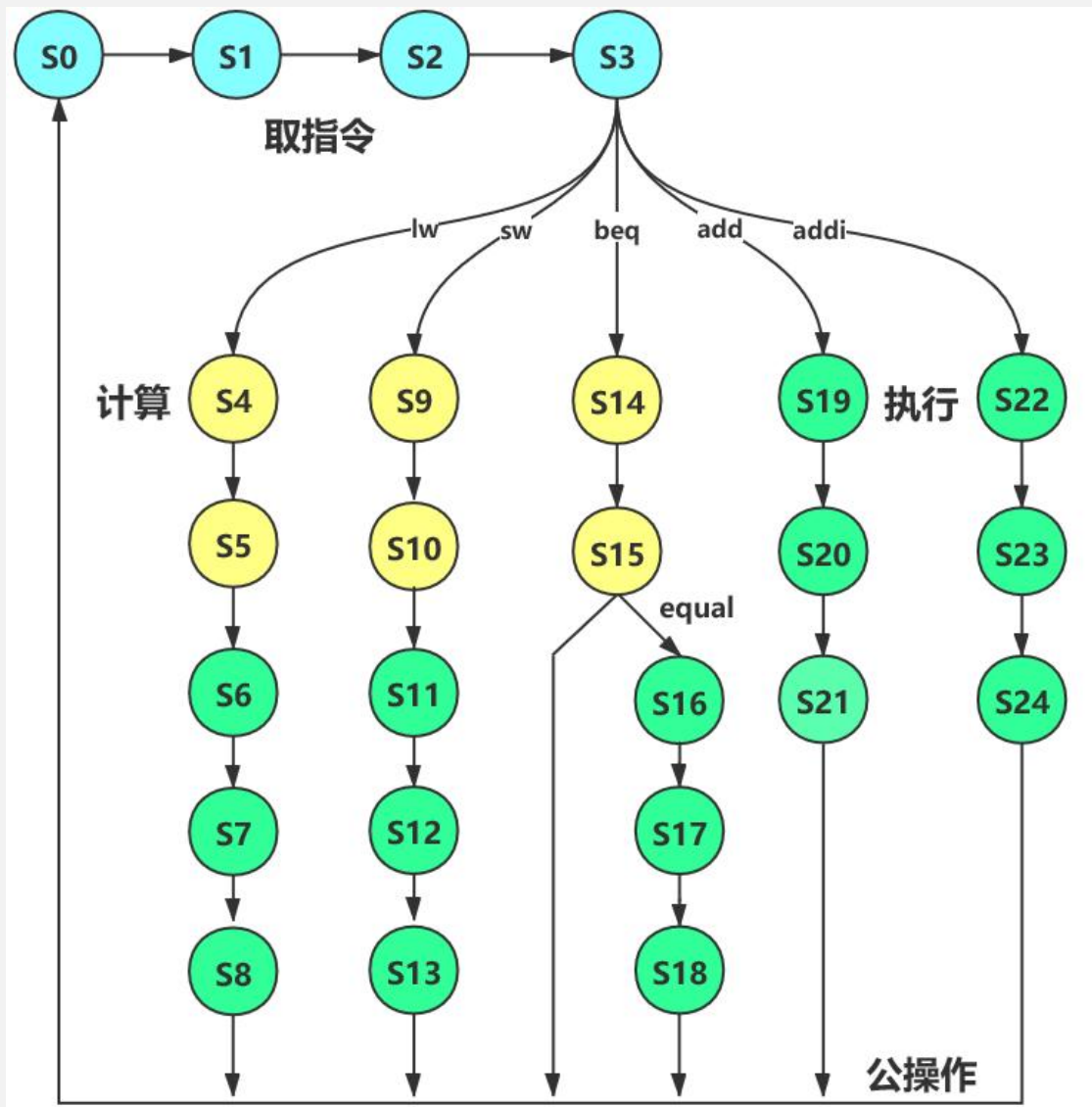
SW

beq

add

addi

9.7 微程序控制器设计

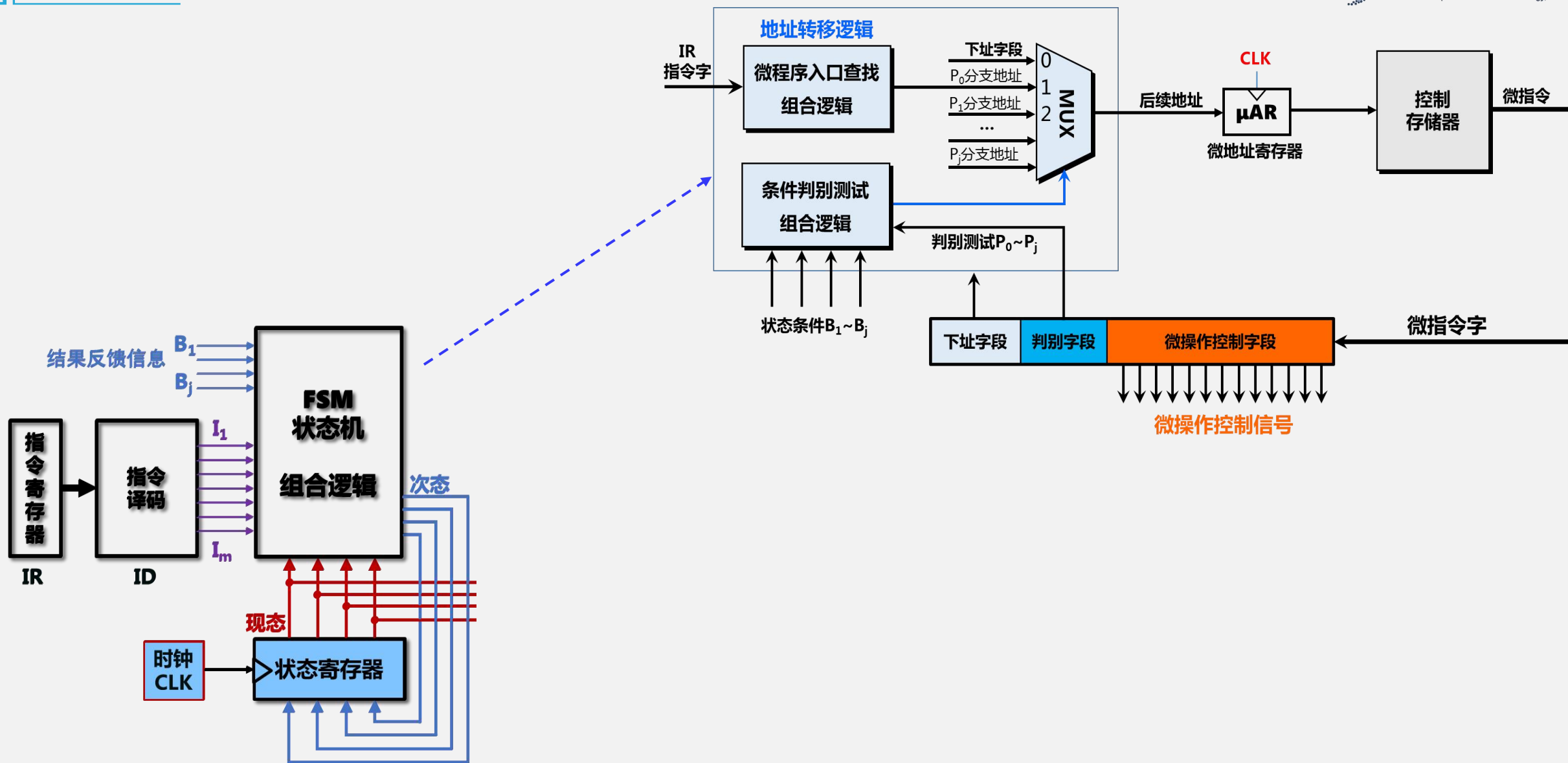


现态	<i>lw</i>	<i>sw</i>	<i>beq</i>	<i>add</i>	<i>addi</i>	<i>equal</i>	次态
S0	x	x	x	x	x		S1
S1	x	x	x	x	x		S2
S2	x	x	x	x	x		S3
S3	1						S4
S3		1					S9
S3			1				S14
S3				1			S19
S3					1		S22

现态	<i>beq</i>	<i>add</i>	<i>addi</i>	<i>equal</i>	次态
S0	x	x	x		S1
S1	x	x	x		S2
S2	x	x	x		S3
S3	1				S14
S3		1			S19
S3			1		S22
s14					s15
s15				1	s16
s15				0	s0
s16					s17
s17					s18
s18					s0
s19					s20
s20					s21
s21					s0
s22					s23
s23					s24
s24					s0

现态	<i>beq</i>	<i>add</i>	<i>addi</i>	<i>equal</i>	次态
0000	x	x	x		0001
0001	x	x	x		0010
0010	x	x	x		0011
0011	1				0100
0011		1			1001
0011			1		1100
0100					0101
0101				1	0110
0101				0	0000
0110					0111
0111					1000
1000					0000
1001					1010
1010					1011
1011					0000
1100					1101
1101					1110
1110					0000

9.7 微程序控制器设计



9.7 微程序控制器设计

6. 程序与微程序的动态执行过程

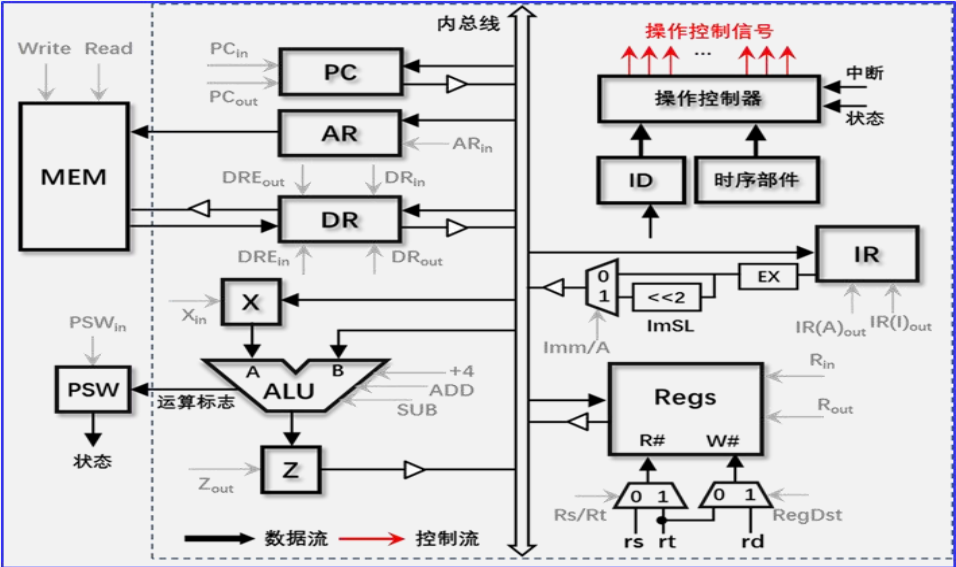
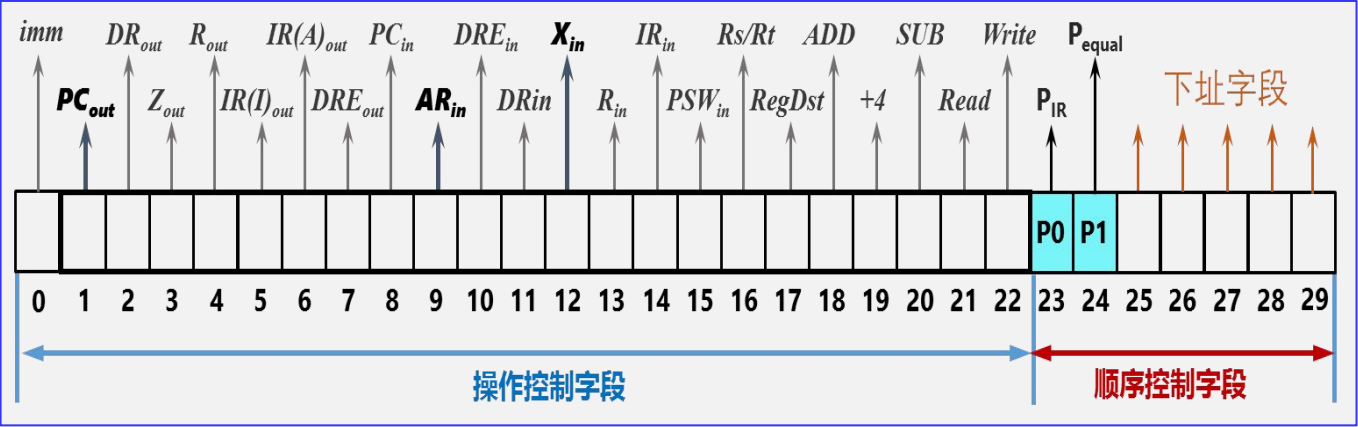
主存

1	lw rt,imm(rs)
2	beq rs,rt,imm
3	add rd,rs,rt
4	sw rt,imm(rs)

控存

状态	微地址	顺序控制字段					
0	00000			0	0	0	0 1
1	00001			0	0	0	1 0
2	00010			0	0	0	1 1
3	00011	1		x	x	x	x x
4	00100			0	0	1	0 1
5	00101			0	0	1	1 0
6	00110			0	0	1	1 1
7	00111			0	1	0	0 0
8	01000			0	0	0	0 0
9	01001			0	1	0	1 0
10	01010			0	1	0	1 1
11	01011			0	1	1	0 0
12	01100			0	1	1	0 1
13	01101			0	0	0	0 0
14	01110			0	1	1	1 1
15	01111	1		0	0	0	0 0
16	10000			1	0	0	0 1
17	10001			1	0	0	1 0
18	10010			0	0	0	0 0
19	10011			1	0	1	0 0
20	10100			1	0	1	0 1
21	10101			0	0	0	0 0

7. 微指令设计原则



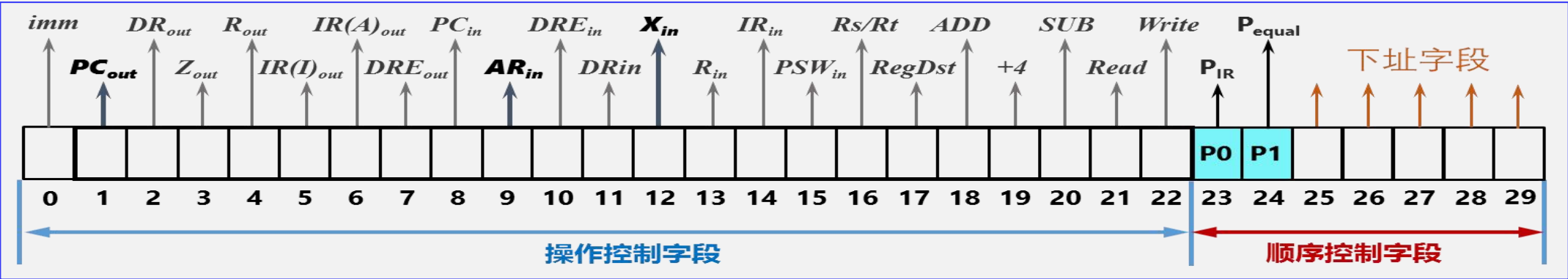
- ◆ 有利于缩短微指令字长度
- ◆ 有利于减少控制存储器容量
- ◆ 有利于提高微程序执行速度
- ◆ 有利于对微指令进行修改
- ◆ 有利于提高微程序设计的灵活性

节拍	lw (5 cycles)	sw (5 cycles)	beq (5 cycles)	add (3 cycles)	addi (3 cycles)
T1	R _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}		
T2	IR(I) _{out} , A, ADD	IR(I) _{out} , A, ADD	R _{out} , Rs/Rt, SUB, PSW _{in}		
T1	Z _{out} , AR _{in}	Z _{out} , AR _{in}	PC _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}
T2	DRE _{in} , Read	R _{out} , Rs/Rt, DR _{in}	IR(A) _{out} , imm, ADD	Rs/Rt, R _{out} , ADD	IR(I) _{out} , A, ADD
T3	DR _{out} , R _{in}	DRE _{out} , Write	Z _{out} , PC _{in} =PSW.equal	Z _{out} , R _{in} , RegDst	Z _{out} , R _{in}



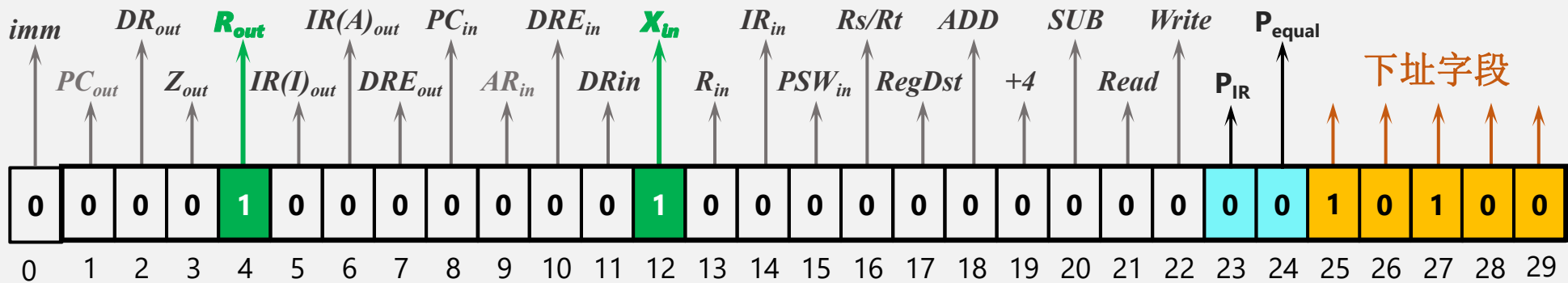
9.7 微程序控制器设计

8. 微命令编码



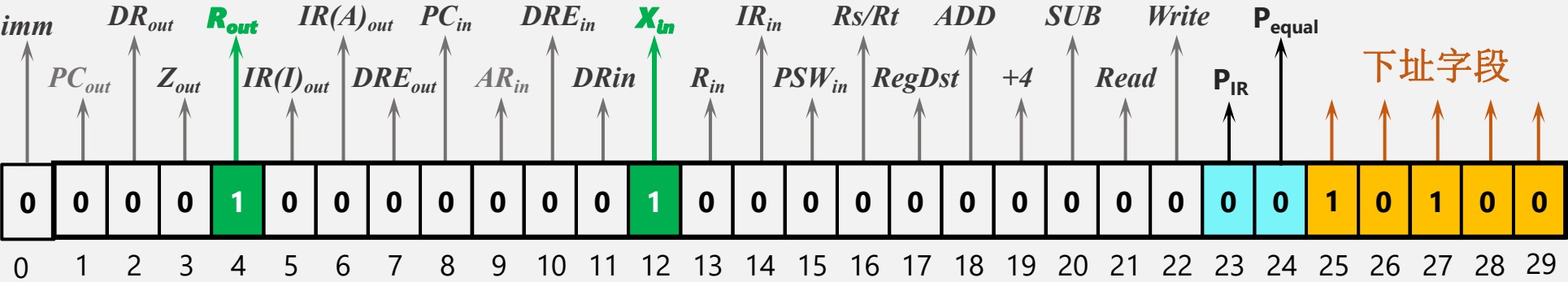
- ◆ 直接表示法
- ◆ 编码表示法
- ◆ 混合表示法

8. 微命令编码 - 直接表示法



- ◆微指令的操作控制字段每位直接表示一个微命令(微操作)
- ◆简单直观，便于输出控制，微指令长度太长，控存容量大
- ◆有减少微指令长度的办法吗？

8. 微命令编码 - 编码表示法



- ◆ 微指令操作控制字段现有23位，若编码则可表示8*1024*1024个微命令
- ◆ 微指令如果全部采用编码表示会带来哪些问题？
- ◆ 对我们有什么启发？ 对微指令的部分进行编码表示

8. 微命令编码 - 编码表示法

◆ 相容性微操作和互斥性微操作

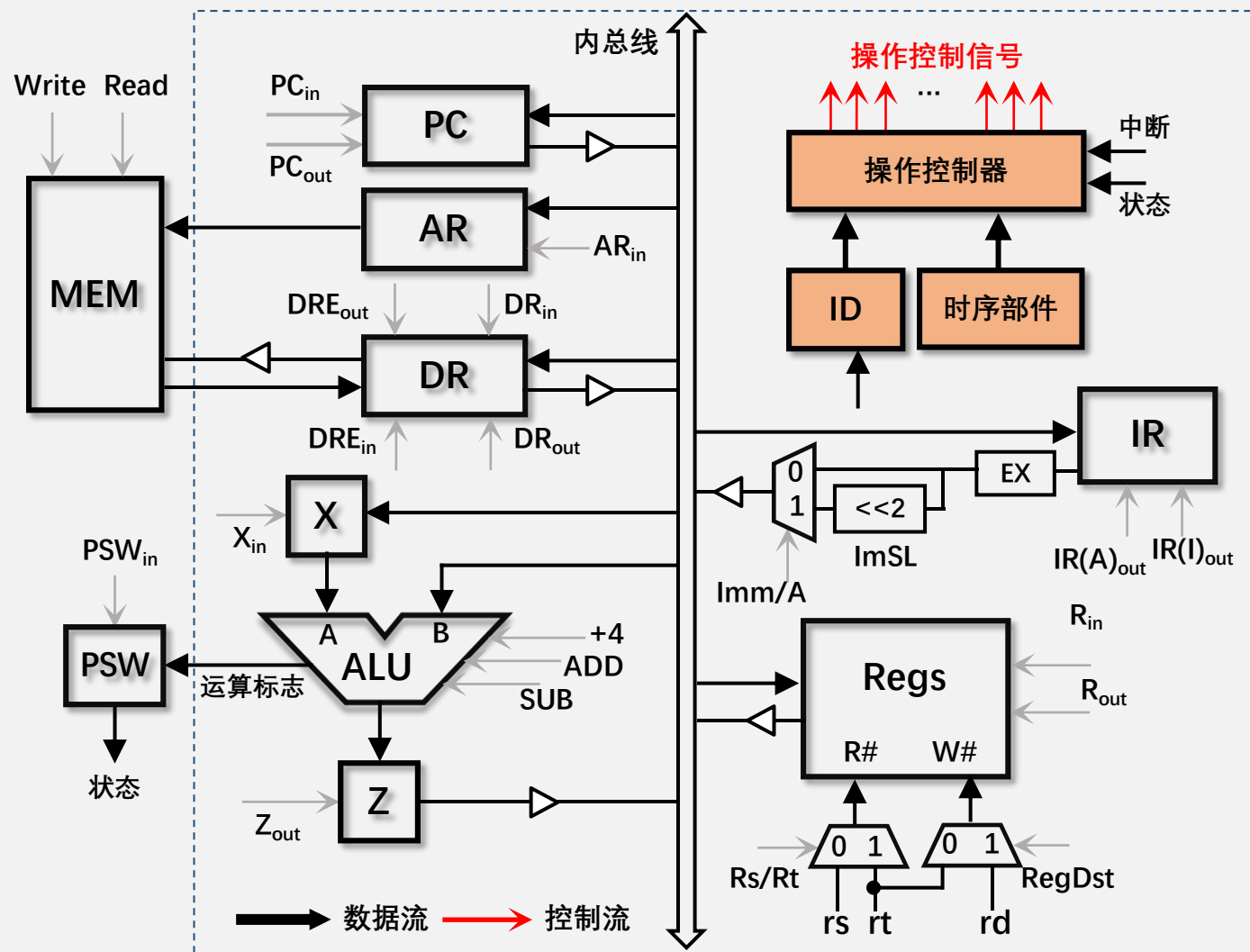
不能同时执行的微操作

能同时执行的微操作

◆ 互斥性微操作

$$PC_{out}, DR_{out}, R_{out}, IR(A)_{out}, Z_{out}, IR(I)_{out}$$

+4、ADD、SUB

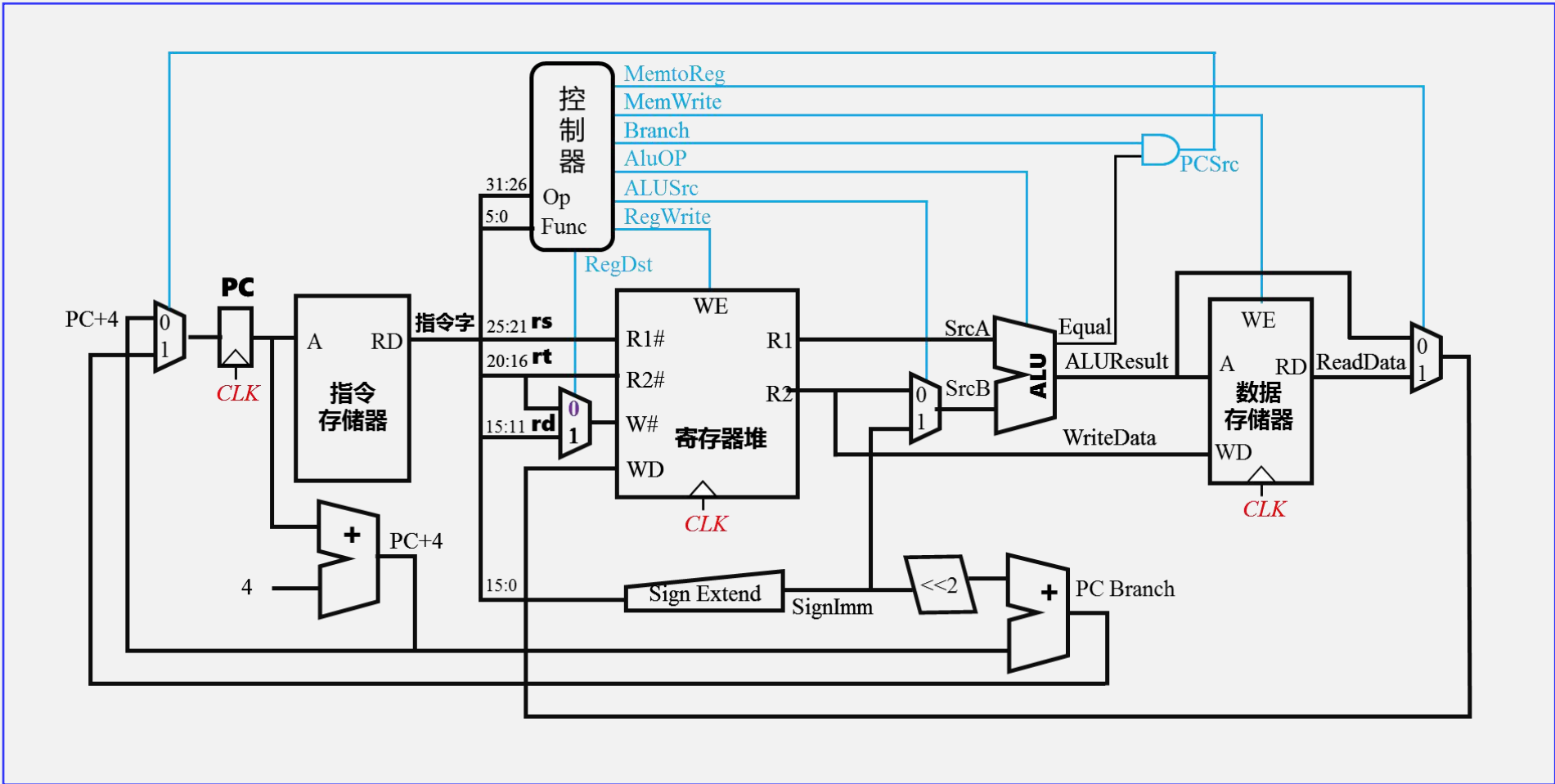




9.7 微程序控制器设计

8. 微命令编码 - 编码表示法

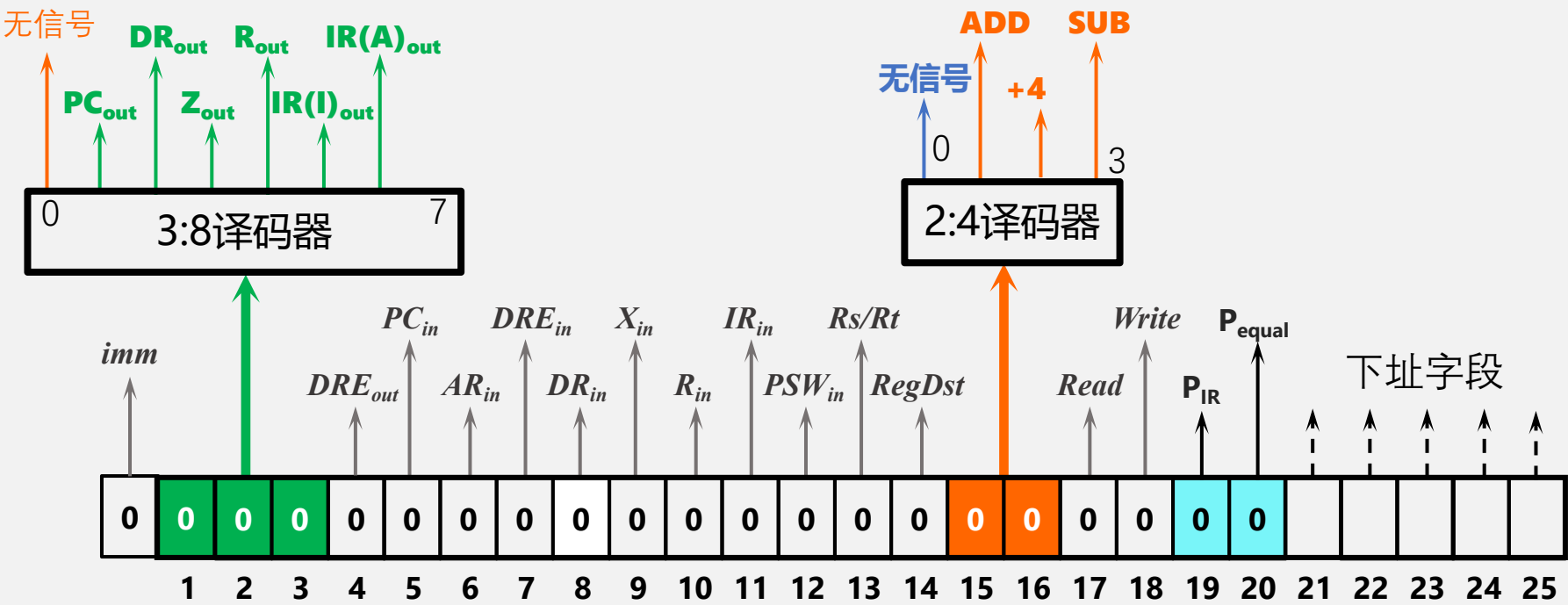
有哪些互斥性微操作？





9.7 微程序控制器设计

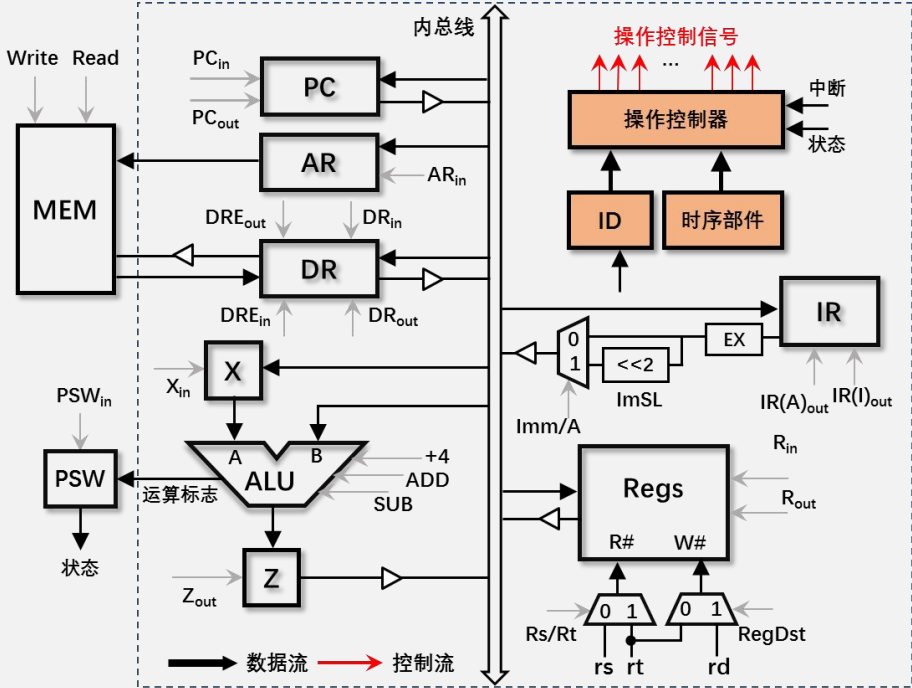
8. 微命令编码 - 编码表示法 - 混合表示法



◆ 有利于缩短微指令长度，减少控存容量，但降低了微指令的执行速度

8. 微命令编码 - 编码表示法 - 混合表示法

如何准确判断互斥性微命?

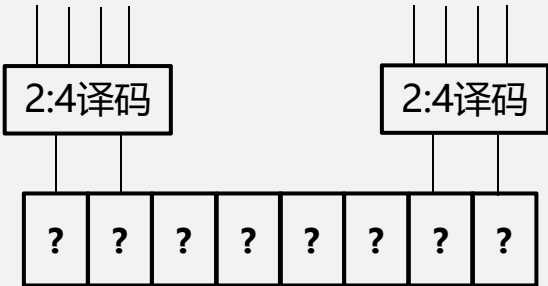
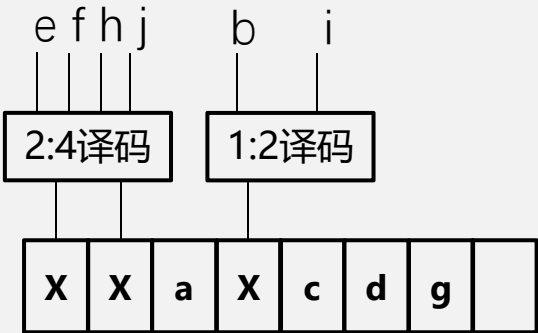


T1	PC _{out} , AR _{in} , X _{in}				
T2	+4				
T3	Z _{out} , PC _{in} , DRE _{in} , Read				
T4	DR _{out} , IR _{in}				
T1	R _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}		
T2	IR(I) _{out} , A, ADD	IR(I) _{out} , A, ADD	R _{out} , Rs/Rt, SUB, PSW _{in}		
T1	Z _{out} , AR _{in}	Z _{out} , AR _{in}	PC _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}
T2	DRE _{in} , Read	R _{out} , Rs/Rt, DR _{in}	IR(A) _{out} , imm, ADD	Rs/Rt, R _{out} , ADD	IR(I) _{out} , A, ADD
T3	DR _{out} , R _{in}	DRE _{out} , Write	Z _{out} , PC _{in} =PSW.equal	Z _{out} , R _{in} , RegDst	Z _{out} , R _{in}

8. 微命令编码 - 编码表示法 - 混合表示法

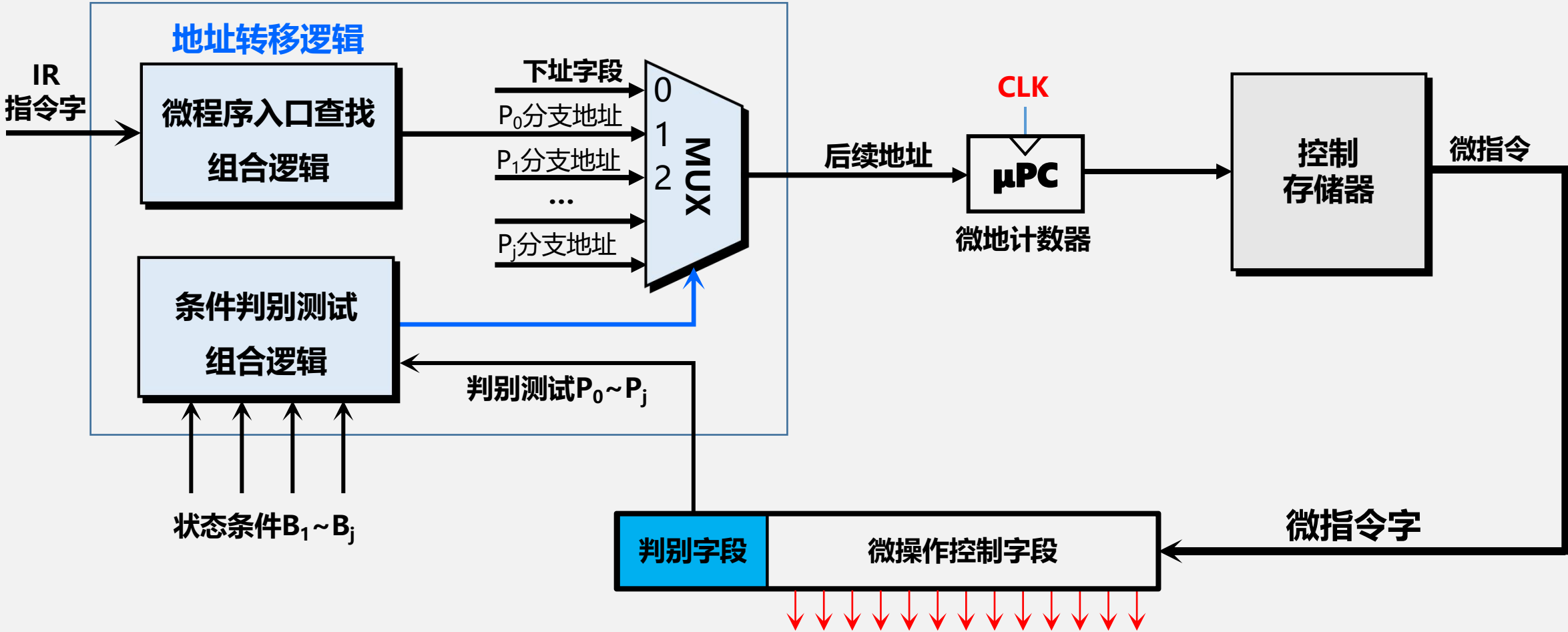
- ◆ 采用如图所示的判定表
- ◆ 你能从中找到哪些互斥性微命令吗?
- ◆ (e,f,h,j)、(b,i)
- ◆ 若微命令控制字段只有8位, 如何表示?

微指令 \ 微命令	a	b	c	d	e	f	g	h	i	j
μl_1	√	√	√	√	√					
μl_2	√			√		√	√			
μl_3		√						√		
μl_4			√							
μl_5			√		√		√		√	
μl_6	√									√
μl_7			√	√				√		
μl_8	√	√						√		



(e, f, h) (b,i, j)

9. 微程序控制器组成原理框图(计数器法)



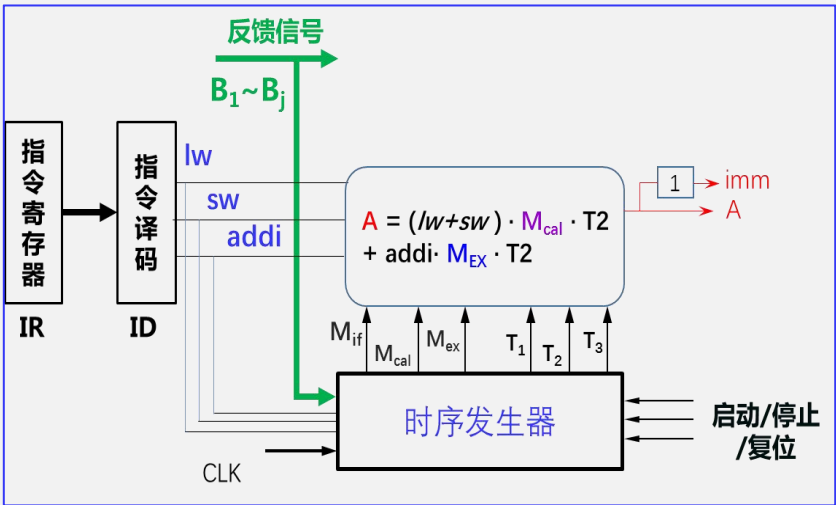
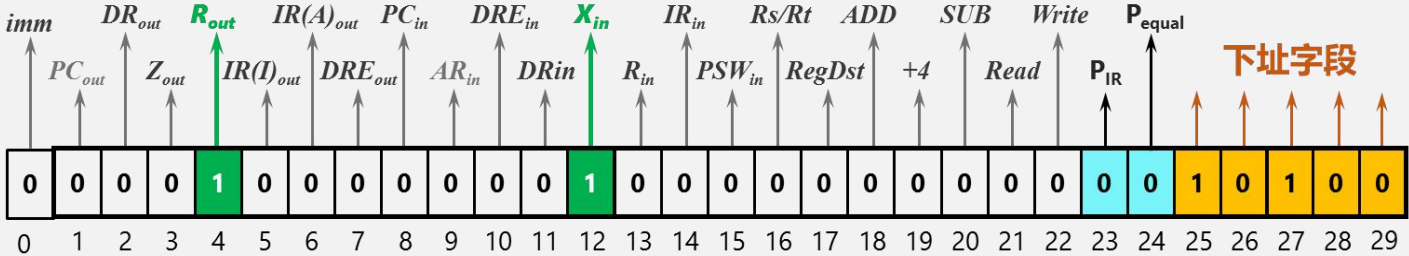
10. 微程序控制器与硬布线控制器对比

◆微程序控制器

速度慢(访控存频繁)、设计规整，简单，易于修改

◆硬布线控制器

执行速度快，设计复杂，不便于修改



1. 设计处理器的步骤

- 1) 选定指令系统，将指令功能用RTL来表示
- 2) 设计部件与通路，根据指令功能设计功能部件，并考虑如何互连
- 3) 设计控点，确定所有功能部件，数据通路所需控制信号
- 4) 分析所有指令的控制信号序列，生成指令与控制信号关系表
- 5) 设计控制器
 - ◆ 设计微指令，微程序，地址转移逻辑，设计微程序控制器
 - ◆ 生成每个控制信号的逻辑表达式，设计硬布线控制器

2. 硬件设计四原则

1) 简单性来自规则性

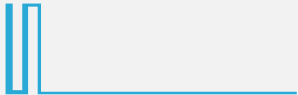
- ◆ Simplicity favors regularity
- ◆ 指令越规整设计越简单

2) 越小越快

- ◆ Smaller is faster
- ◆ 面积小，传播路径小，门延迟少

3) 加快经常性事件 (Make the common case fast)

4) 好的设计需要适度的折衷 (Good design demands good compromises)



第五部分完