

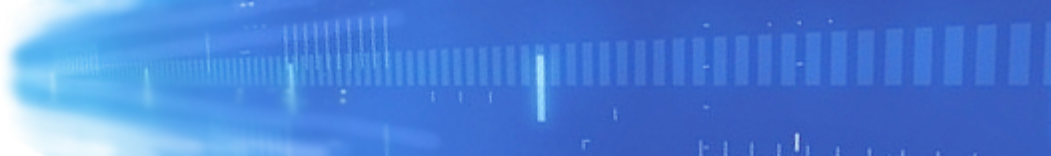


单片机原理与应用

北京航空航天大学
电子信息工程学院

王 俊

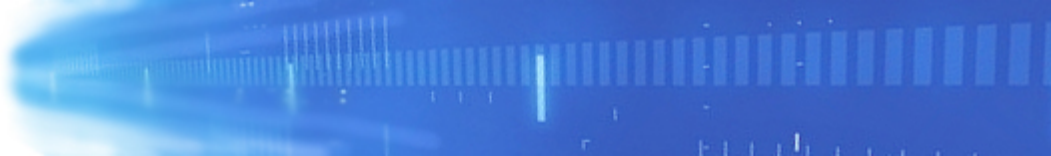
wangj203@buaa.edu.cn



第二章 单片机的基本结构(8051)

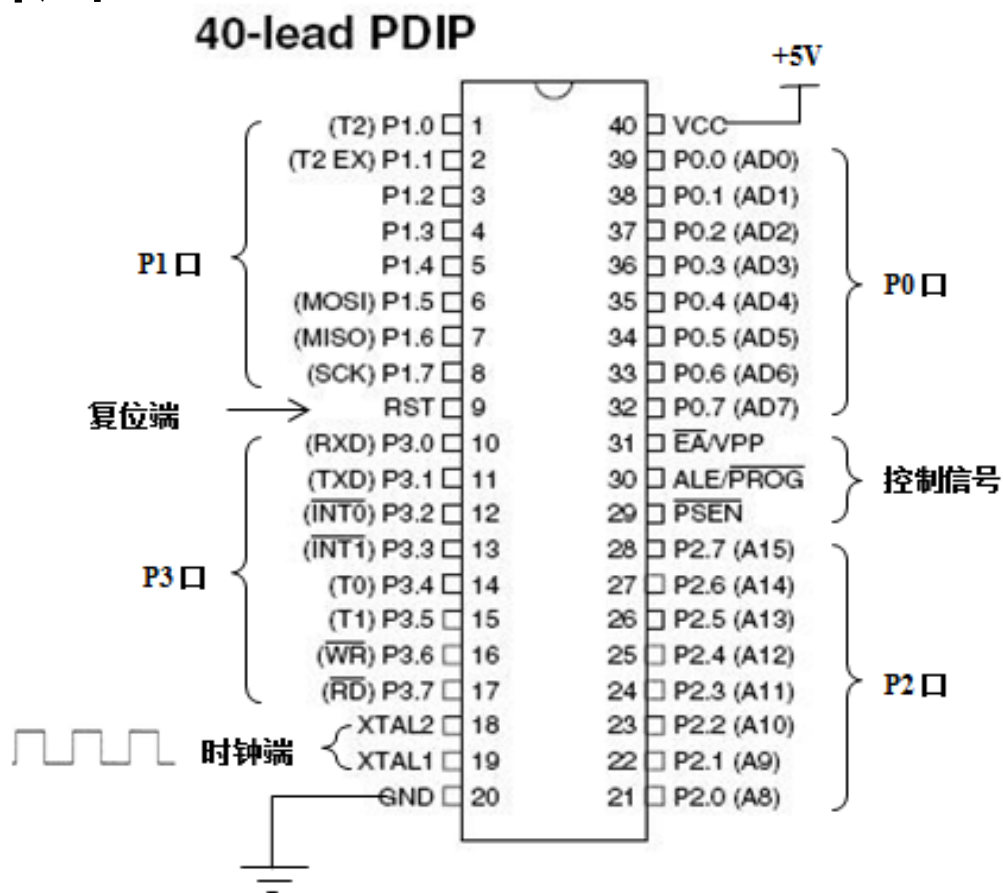
- 引脚及最小系统
- 内部结构
- 基本参数
- CPU
- 存储器
- 工作过程
- 实例分析





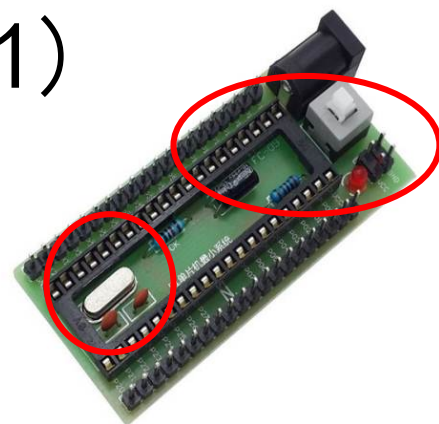
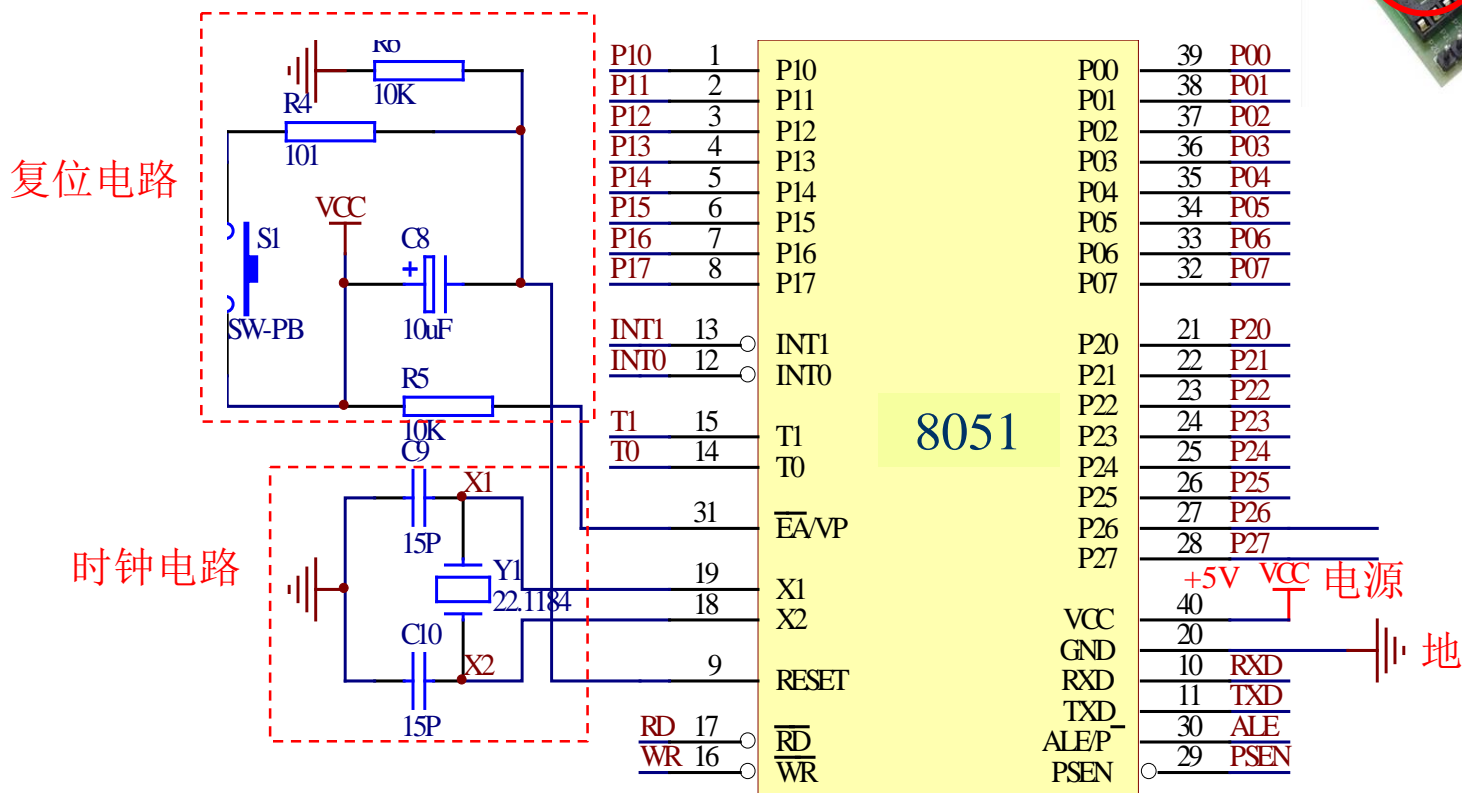
第二章 单片机的基本结构(8051)

■ 引脚



第二章 单片机的基本结构(8051)

■ 最小系统连接图





第二章 单片机的基本结构(8051)

■ 两位巨人





第二章 单片机的基本结构(8051)



■ 图灵 (Alan Mathison Turing)

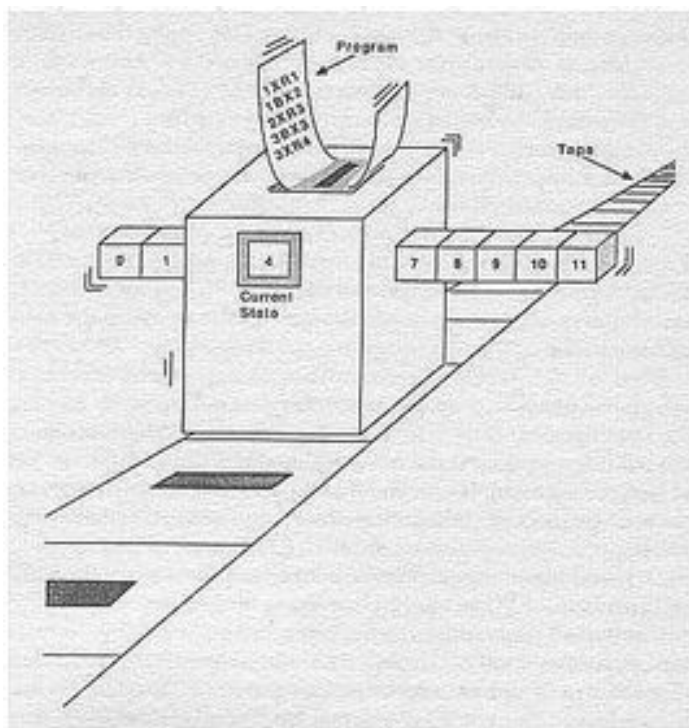
- 英国著名的数学家、逻辑学家，计算机之父
- 著名图灵机模型为现代计算机的逻辑工作方式奠定了基础。人们为纪念其在计算机领域的卓越贡献而设立“图灵奖”。
- 一位世界级长跑运动员。





第二章 单片机的基本结构(8051)

➤ 图灵机模型

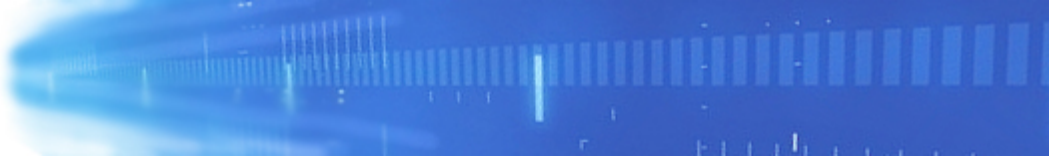


第二章 单片机的基本结构(8051)

■ 冯·诺依曼 (John von Neumann)

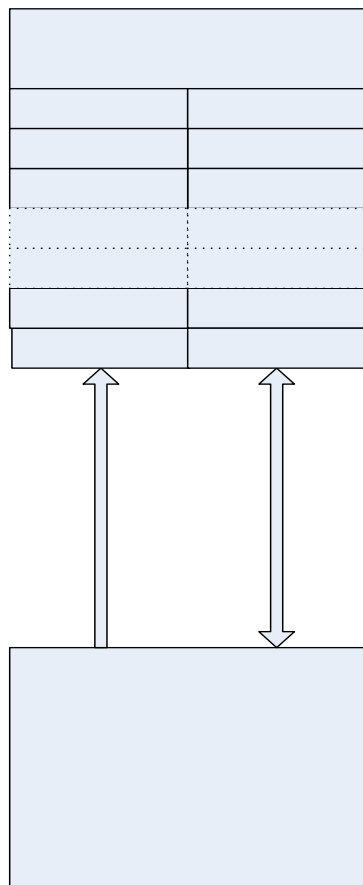
- 出生于匈牙利的美国籍犹太人数学家
- 现代电子计算机创始人之一。他在计算机科学、经济、物理学中的量子力学及几乎所有数学领域都作过重大贡献。





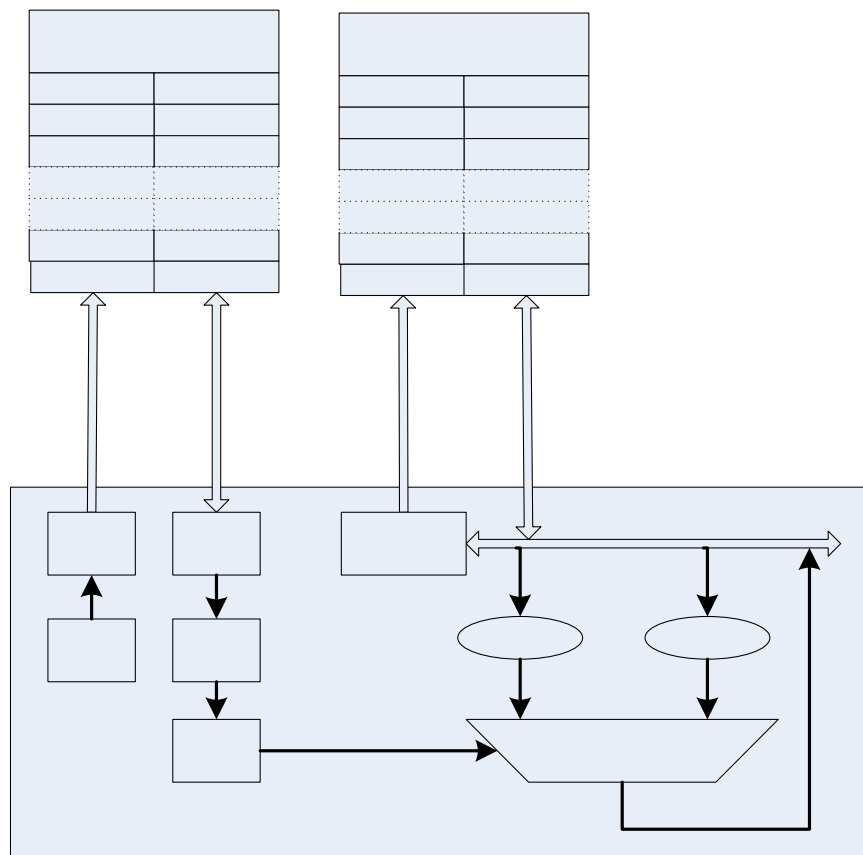
第二章 单片机的基本结构(8051)

➤ 冯·诺依曼结构



第二章 单片机的基本结构(8051)

➤ “改进的” 冯•诺依曼结构

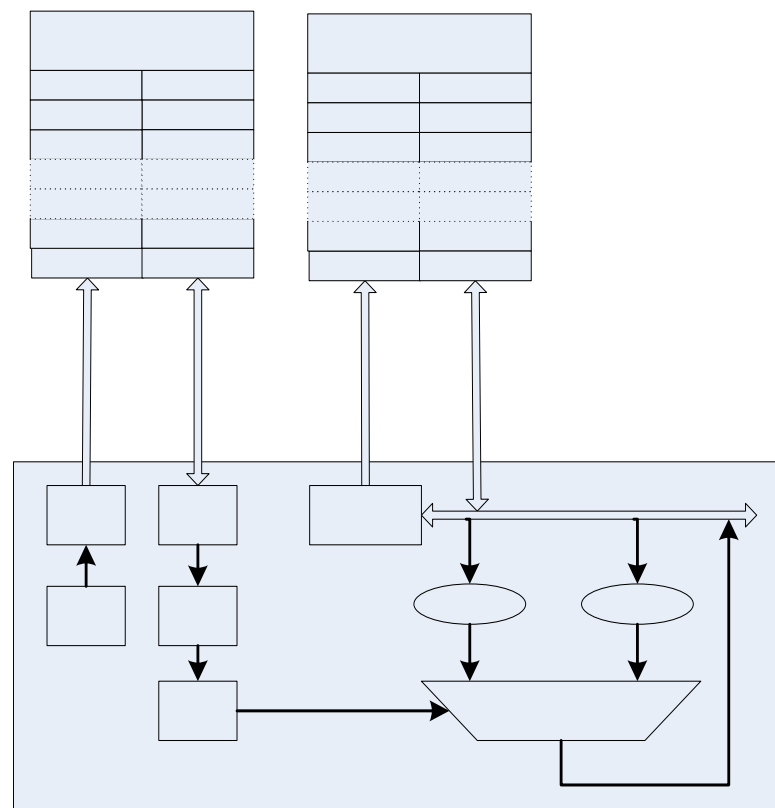


第二章 单片机的基本结构(8051)



■ 基本参数

- 8位CPU
- 4kbytes 程序存储器
- 256bytes的数据存储器
- 32条I/O口线
- 21个专用寄存器
- 2个可编程定时/计数器
- 5个中断源，2个优先级
- 一个全双工串行通信口



第二章 单片机的基本结构(8051)

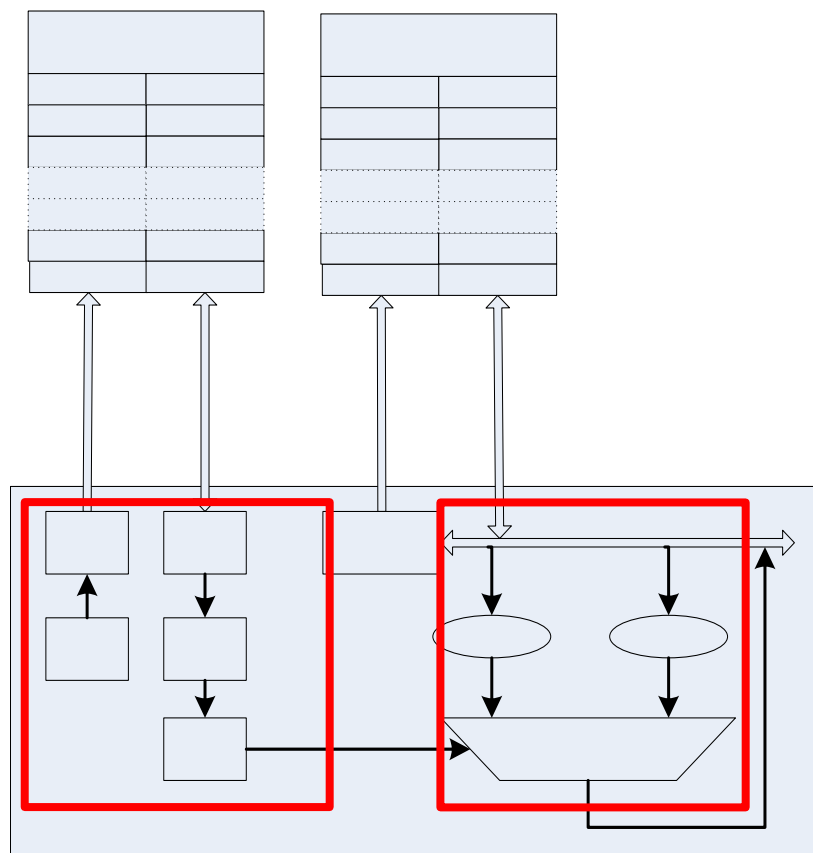
■ CPU

➤ ALU

- 加减运算
- 逻辑运算

➤ 控制器

- 指令寄存、译码
- 程序跳转控制





第二章 单片机的基本结构(8051)

■ 指令集与指令周期

➤ 指令集

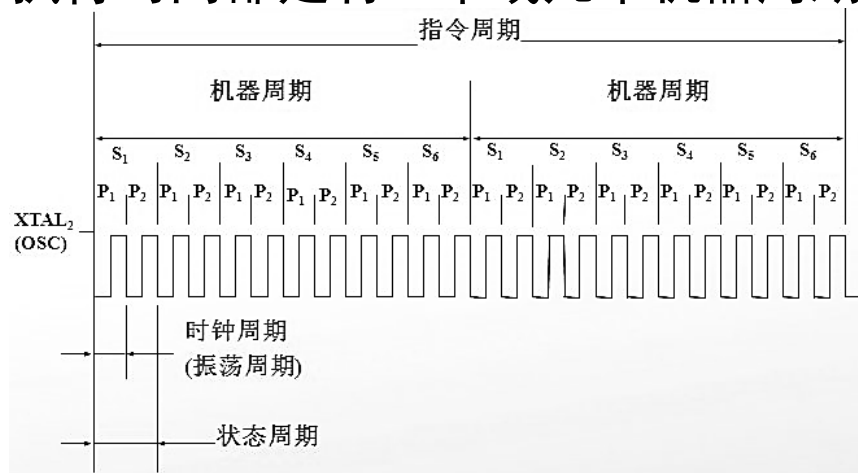
- 数据传送类
- 算术操作类
- 逻辑操作类
- 控制转移类
- 布尔变量操作类

助记符	功能说明	字节数	12 时钟/机器周期所需时钟	1 时钟/机器周期所需时钟	效率提升 (倍)
MOV A, Rn	寄存器内容送入累加器	1	12	1	12
MOV A, direct	直接地址单元中的数据送入累加器	2	12	2	6
MOV A, @Ri	间接 RAM 中的数据送入累加器	1	12	2	6
MOV A, #data	立即送入累加器	2	12	2	6
MOV Rn, A	累加器内容送入寄存器	1	12	2	6
MOV Rn, direct	直接地址单元中的数据送入寄存器	2	24	4	6
MOV Rn, #data	立即数送入寄存器	2	12	2	6
MOV direct, A	累加器内容送入直接地址单元	2	12	3	4
MOV direct, Rn	寄存器内容送入直接地址单元	2	24	3	8
MOV direct, direct	直接地址单元中的数据送入另一个直接地址单元	3	24	4	6
MOV direct, @Ri	间接 RAM 中的数据送入直接地址单元	2	24	4	6
MOV direct, #data	立即数送入直接地址单元	3	24	3	8
MOV @Ri, A	累加器内容送入间接 RAM 单元	1	12	3	4
MOV @Ri, direct	直接地址单元数据送入间接 RAM 单元	2	24	3	8
MOV @Ri, #data	立即数送入间接 RAM 单元	2	12	3	4
MOV DPTR, #data16	16 位立即数送入地址寄存器	3	24	3	8
MOVC A, @A+DPTR	以 DPTR 为基址变址寻址单元中的数据送入累加器	1	24	4	6
MOVC A, @A+PC	以 PC 为基址变址寻址单元中的数据送入累加器	1	24	4	6
MOVX A, @Ri	外部 RAM (8 位地址) 送入累加器	1	24	3	8
MOVX A, @DPTR	外部 RAM (16 位地址) 送入累加器	1	24	3	8
MOVX @Ri, A	累加器送外部 RAM (8 位地址)	1	24	3	8
MOVX @DPTR, A	累加器送外部 RAM (16 位地址)	1	24	3	8
PUSH direct	直接地址单元中的数据压入堆栈	2	24	4	6
POP direct	出栈送直接地址单元	2	24	3	8
XCH A, Rn	寄存器与累加器交换	1	12	3	4
XCH A, direct	直接地址单元与累加器交换	2	12	4	3
XCH A, @Ri	间接 RAM 与累加器交换	1	12	4	3
XCHD A, @Ri	间接 RAM 的低半字节与累加器交换	1	12	4	3

第二章 单片机的基本结构(8051)

➤ 指令周期

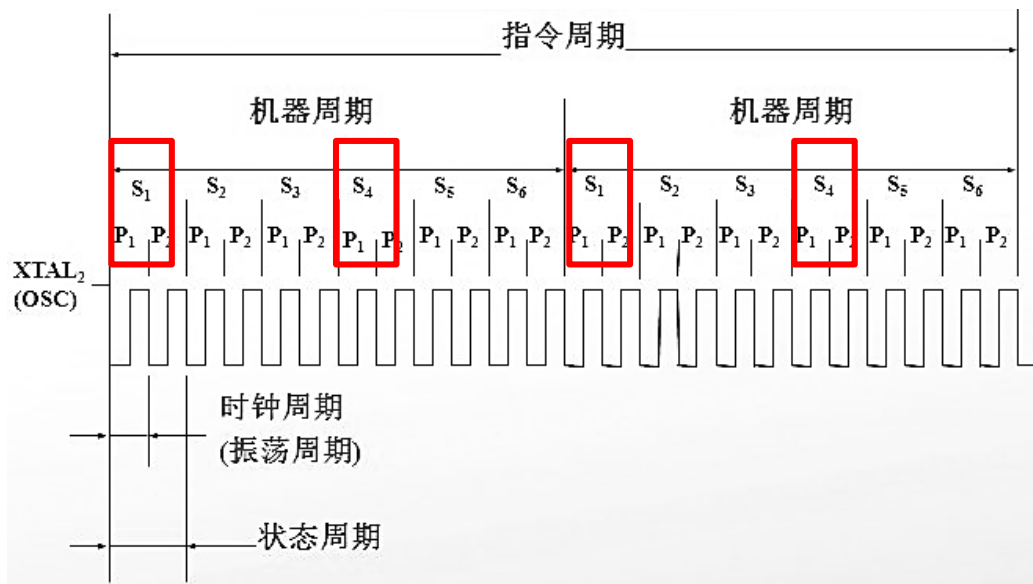
- **状态周期**：每个状态周期为时钟周期的2倍，是时钟周期经二分频后得到的。
- **机器周期**：一个机器周期包含6个状态周期 $S_1 \sim S_6$ ，也就是12个时钟周期，即6节（s）12拍（p）。是指令执行的时间单位。
- **指令周期**：它是指CPU完成一条操作所需的全部时间。每条指令执行时间都是有一个或几个机器周期组成。



第二章 单片机的基本结构(8051)

➤ 指令周期

- 执行过程分为：**取指令、译码、执行**三步。
- 在s1p1和s4p1下降沿时，分别读一次指令（每次读取1字节）

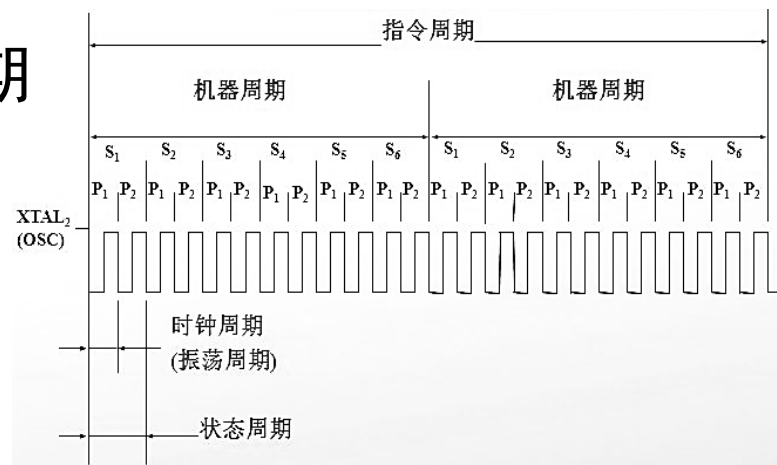


第二章 单片机的基本结构(8051)

➤ 指令周期

- 不同指令有不同的指令周期

- 单字节单周期
- 双字节单周期
- 单字节双周期
- 双字节双周期
- 双字节三周期...



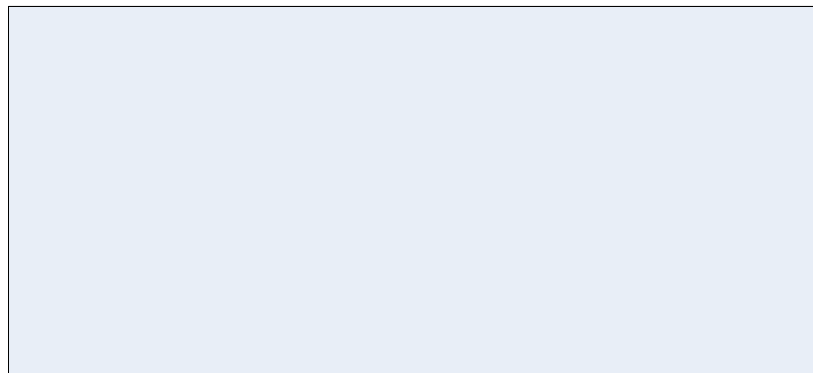
例	指令	字节数	指令周期
单字节单周期	INC A	1	1
双字节单周期	MOV A,#data	2	1
单字节双周期	INC DPTR	1	2
双字节双周期	MOV Rn,direct	2	2
三字节双周期	LJMP addr16	3	2

第二章 单片机的基本结构(8051)

➤ 指令周期

- 不同寻址方式有不同的指令周期
 - **寄存器寻址**，ALU能直接访问这些内部寄存器，不需要读写存储器，执行速度快
 - **立即数寻址**，操作数直接从指令中取得并送给ALU运算，不执行指令周期，执行速度快
 - **直接寻址**和**间接寻址**需要访问数据存储器，速度慢

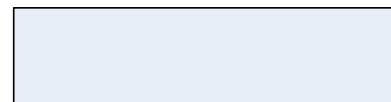
指令	寻址方式	字节数	12时钟/周期所需时钟	1时钟/周期所需时钟
ADD A, Rn	寄存器寻址	1	12	2
ADD A, #data	立即数寻址	2	12	2
ADD A, direct	直接寻址	2	12	3
ADD A, @Ri	间接寻址	1	12	4



第二章 单片机的基本结构(8051)

■ 存储器

➤ 数据存储器 (00H~FFH)



- 工作寄存器区
- 位寻址区
- 用户RAM区
- 特殊寄存器区

第二章 单片机的基本结构(8051)

■ 工作寄存器

- 4组（每组8个， $R0 \sim R7$ ）
- 某一时刻，ALU直接操作其中一组，执行速度快
- 程序状态字PSW的RS1和RS0切换当前工作寄存器组



RS1 RS0	指令
00	第0组
01	第1组
10	第2组
11	第3组

第二章 单片机的基本结构(8051)

■ 工作寄存器

➤ 举例：使用第一组寄存器

CLR RS1; --RS1置0

SETB RS0; --RS0置1

RS1 RS0	指令
00	第0组
01	第1组
10	第2组
11	第3组

➤ 作用：暂存数据，且各组寄存器的值相互屏蔽，完成快速现场保护任务



第二章 单片机的基本结构(8051)

■ 位寻址区

➤ 计算机中大量使用布尔运算，因此位寻址区配合位操作指令，可以实现更直接简单的运算

➤ 例：

- CLR 25H. 5,
表示25H单元的第5位置零

位地址								单元地址
07H	06H	05H	04H	03H	02H	01H	00H	20H
0FH	0EH	0DH	0CH	0BH	0AH	09H	08H	21H
17H	16H	15H	14H	13H	12H	11H	10H	22H
1FH	1EH	1DH	1CH	1BH	1AH	19H	18H	23H
27H	26H	25H	24H	23H	22H	21H	20H	24H
2FH	2EH	2DH	2CH	2BH	2AH	29H	28H	25H
37H	36H	35H	34H	33H	32H	31H	30H	26H
3FH	3EH	3DH	3CH	3BH	3AH	39H	38H	27H
47H	46H	45H	44H	43H	42H	41H	40H	28H
4FH	4EH	4DH	4CH	4BH	4AH	49H	48H	29H
57H	56H	55H	54H	53H	52H	51H	50H	2AH
5FH	5EH	5DH	5CH	5BH	5AH	59H	58H	2BH
67H	66H	65H	64H	63H	62H	61H	60H	2CH
6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	2DH
77H	76H	75H	74H	73H	72H	71H	70H	2EH
7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	2FH

第二章 单片机的基本结构(8051)

■ 用户RAM

➤ 存储程序中的变量

- 变量：程序执行过程中值可以变化的量，例如

```
int x = 0;
```

```
x = 1;
```

➤ 堆栈

- 暂存数据、地址，配合中断保护断点和现场
- 栈顶起始位置7FH，向低地址生长，避免与变量冲突
- 先进后出原则



第二章 单片机的基本结构(8051)

■ 先进后出原则

➤ 入栈:

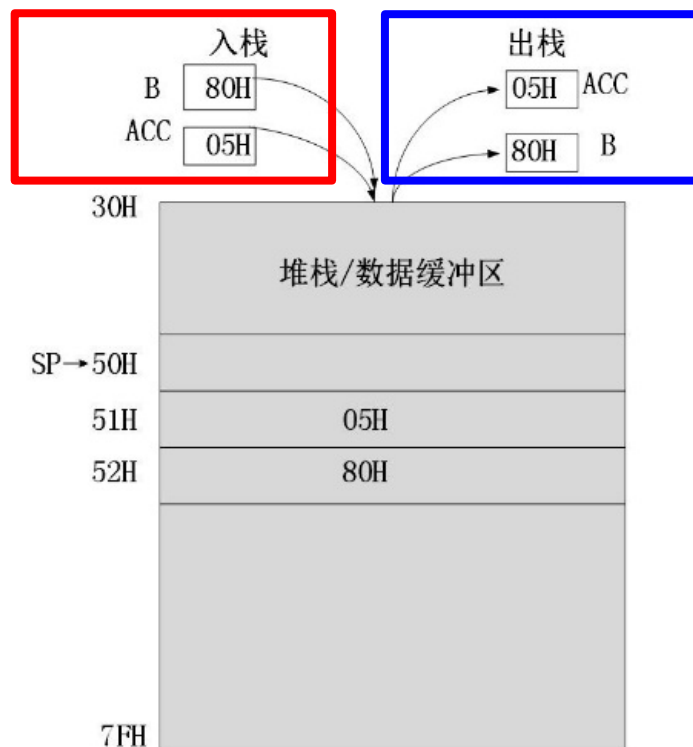
PUSH B

PUSH ACC

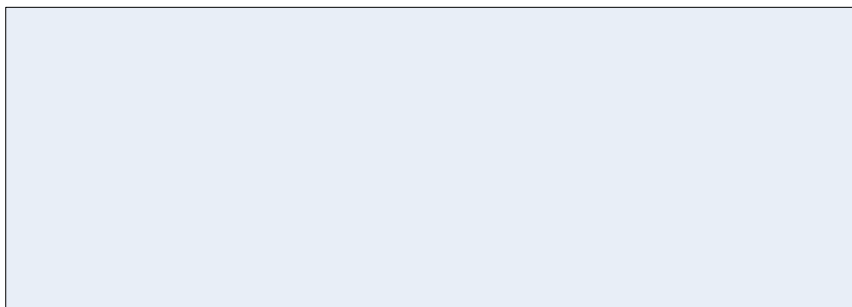
➤ 出栈:

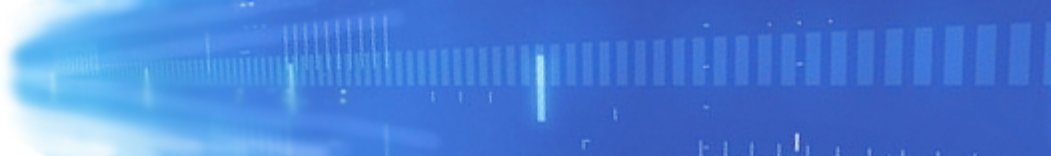
POP ACC

POP B



出栈入栈顺序相反！

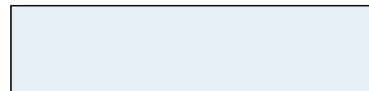




第二章 单片机的基本结构(8051)

■ 存储器

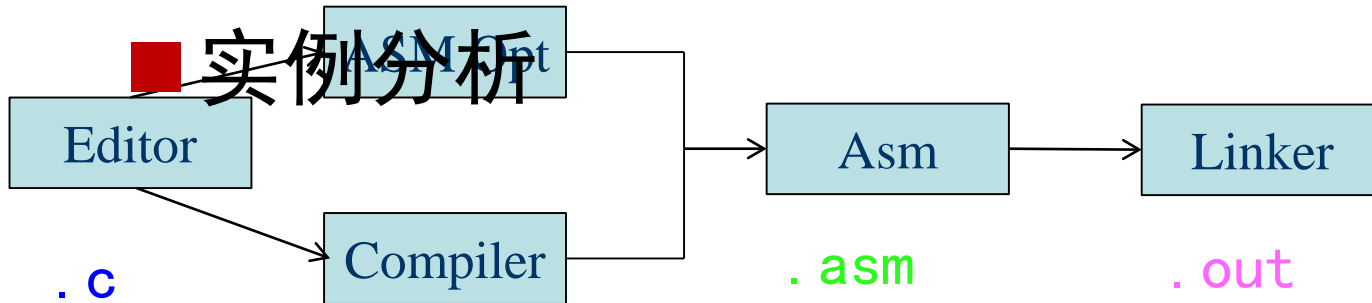
➤ 程序存储器



- 用于存储用户程序、数据和表格等信息
- 64KB统一寻址空间
- 提供外部扩展接口——单片机EA引脚
 - EA=1, 执行片内和片外ROM中的程序
 - EA=0, 只能寻址外部程序存储器址

第二章 单片机的基本结构(8051)

实例分析



```

void main()
{
    int x, y, z;
    x = 1;
    y = 2;
    z = x+y;
}
    
```

```

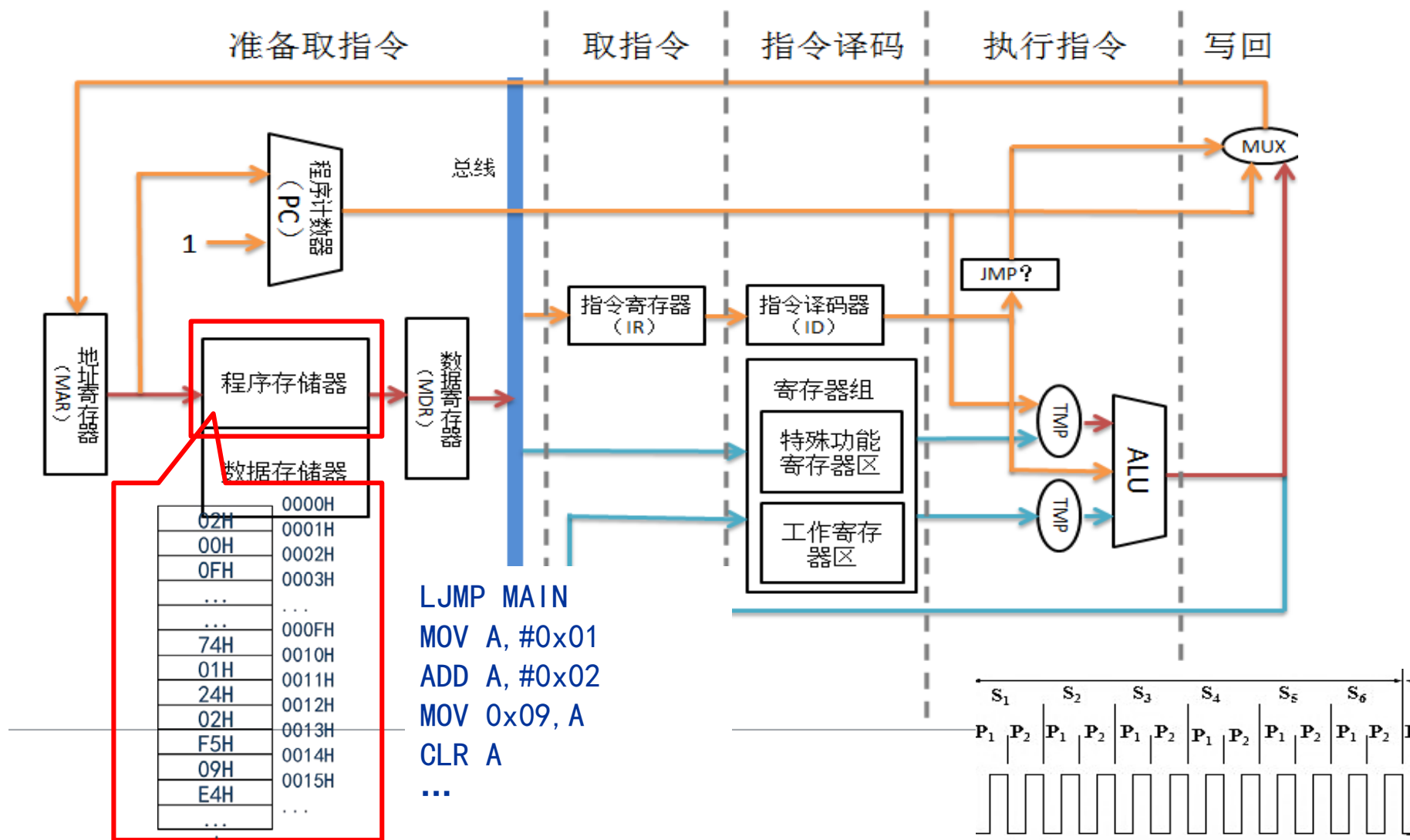
ORG 0000H; 表示下一条指令存放地址
LJMP MAIN ..... 02000FH
ORG 000FH
MAIN: MOV A, #0x01 ..... 7401H
      ADD A, #0x02 ..... 2402H
      MOV 0x09, A ..... F509H
      CLR A ..... E4H
      ...
    
```

ROM中的位置

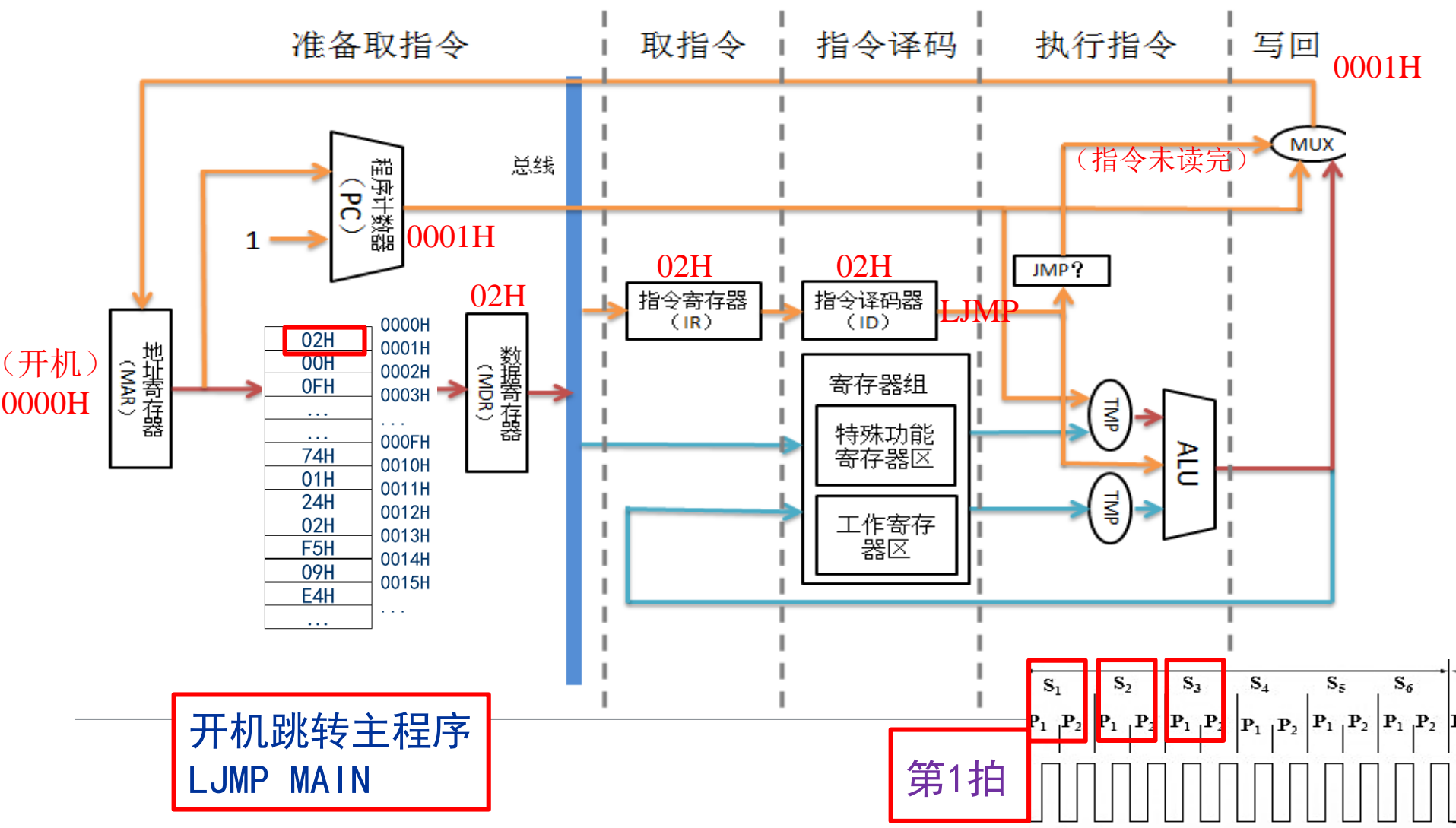
02H	0000H
00H	0001H
0FH	0002H
...	0003H
...	...
...	000FH
74H	0010H
01H	0011H
24H	0012H
02H	0013H
F5H	0014H
09H	0015H
E4H	...
...	...

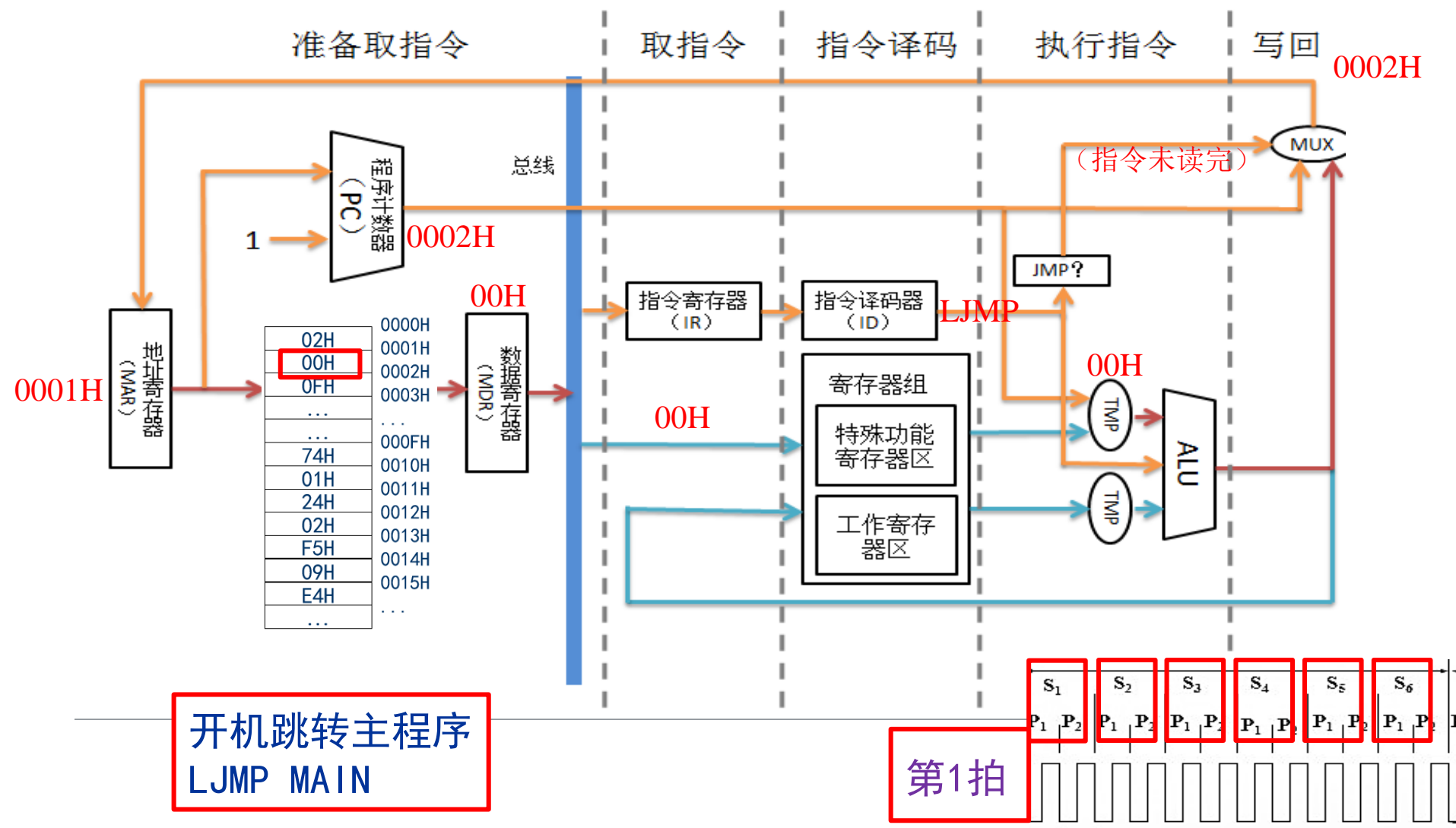


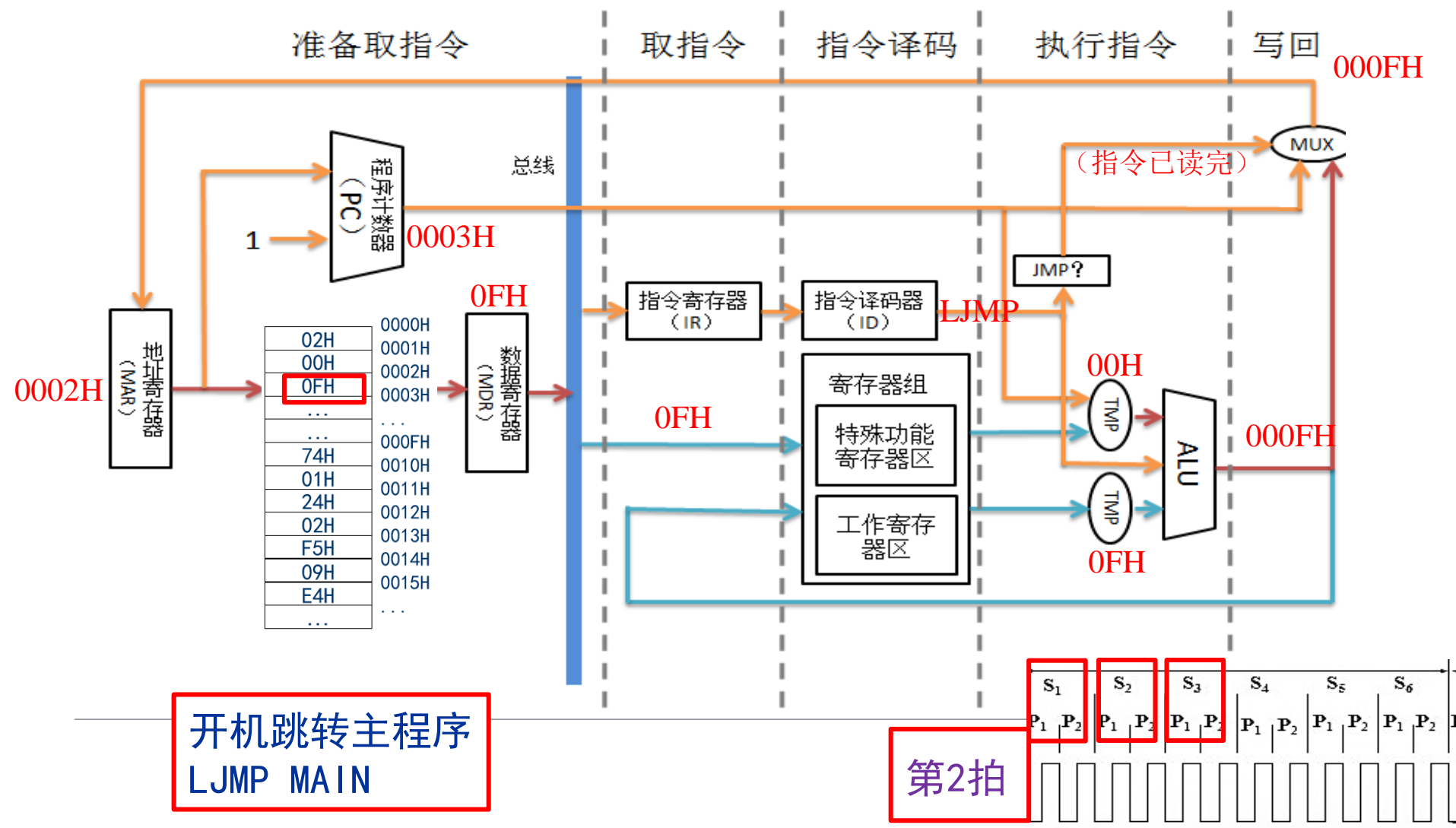
第二章 单片机的基本结构(8051)



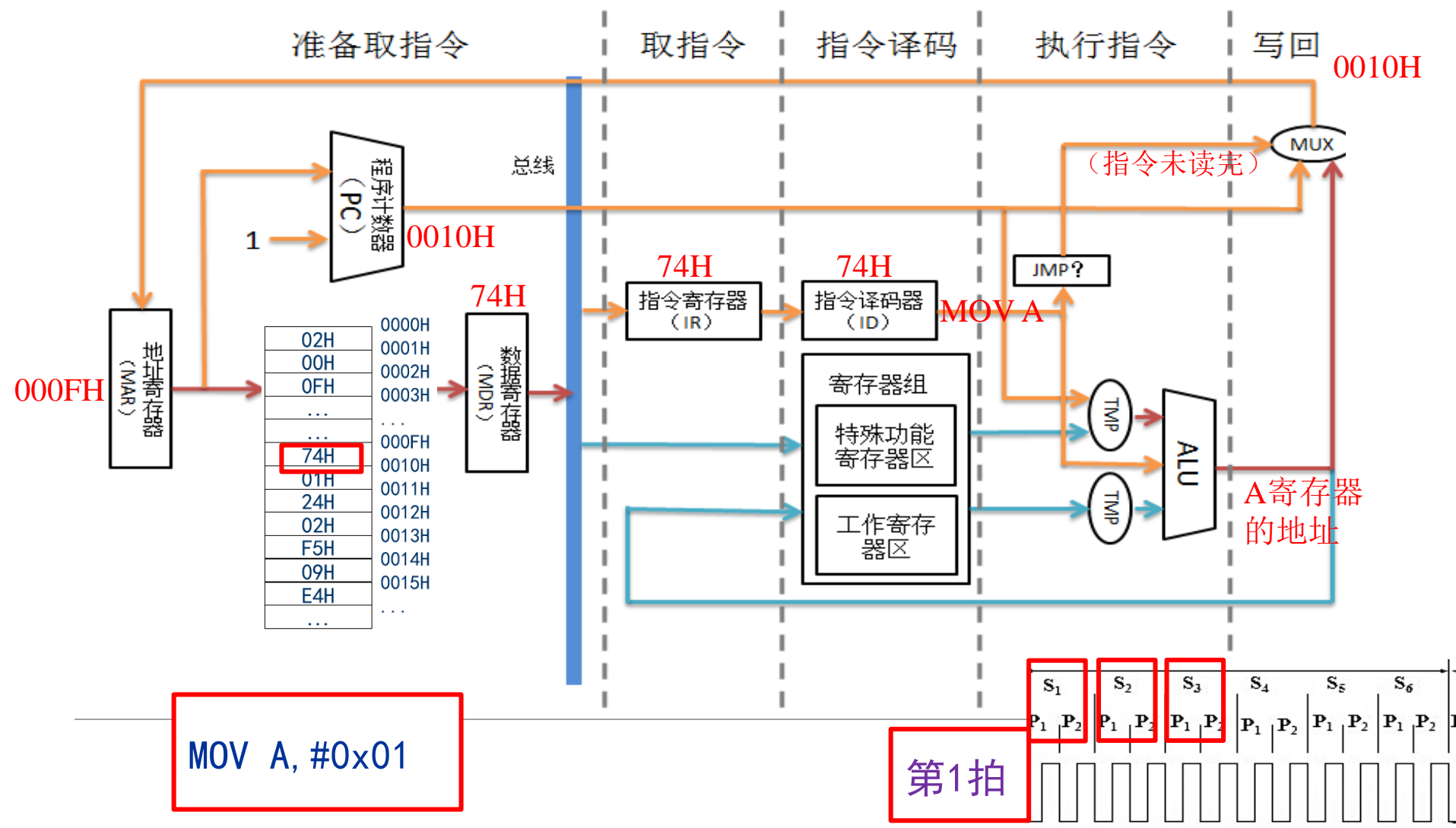
第二章 单片机的基本结构(8051)



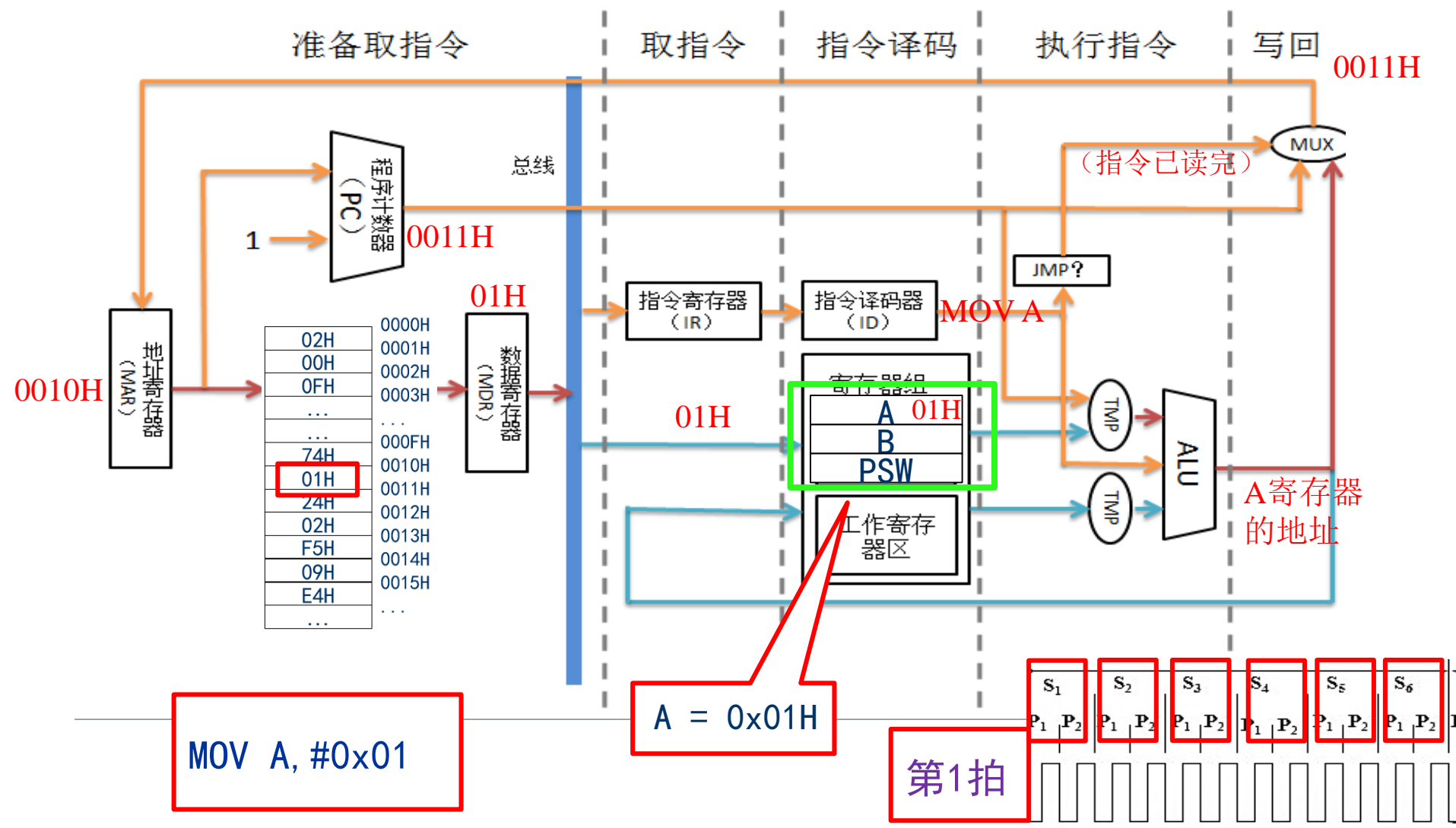




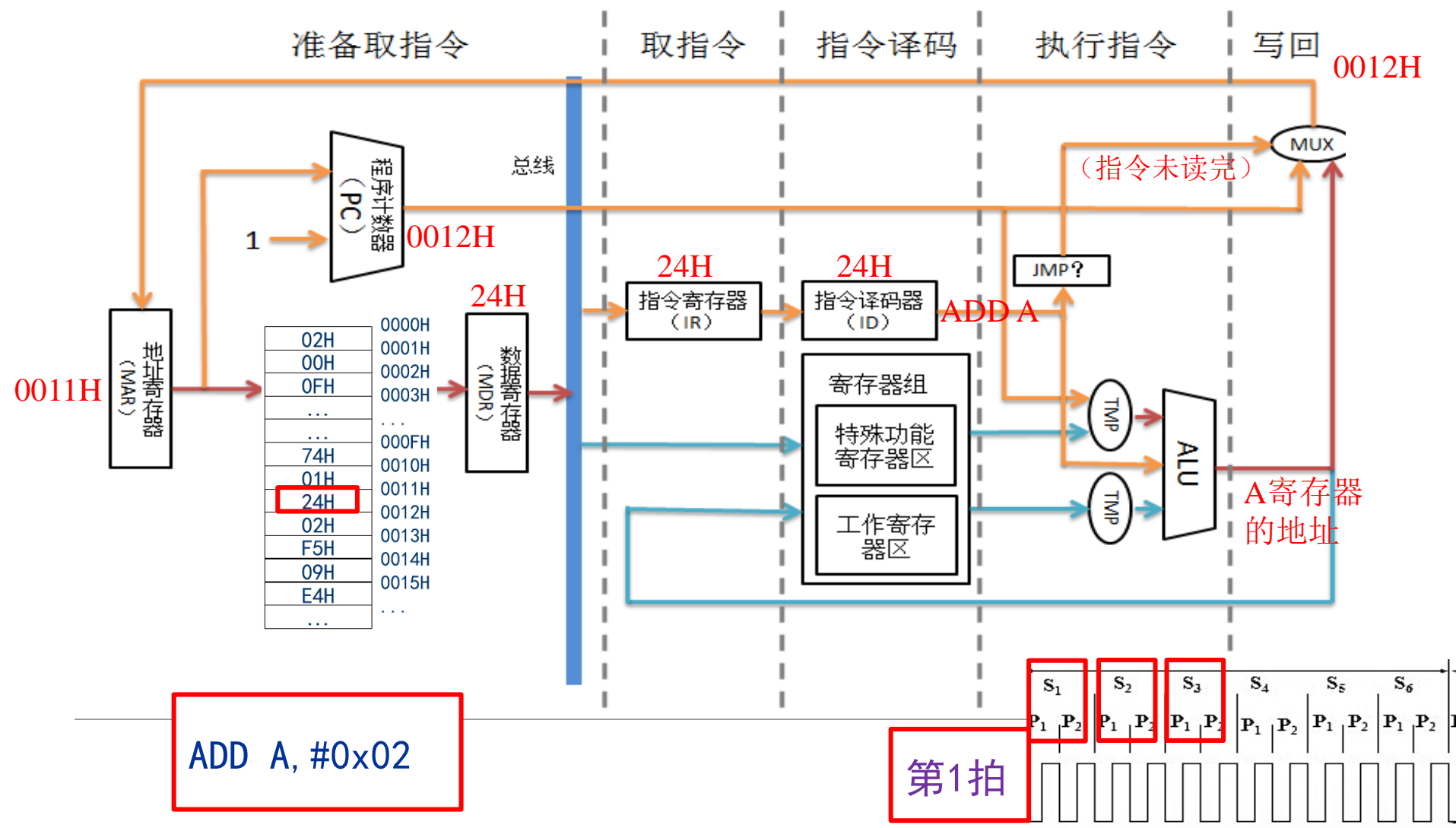
第二章 单片机的基本结构(8051)



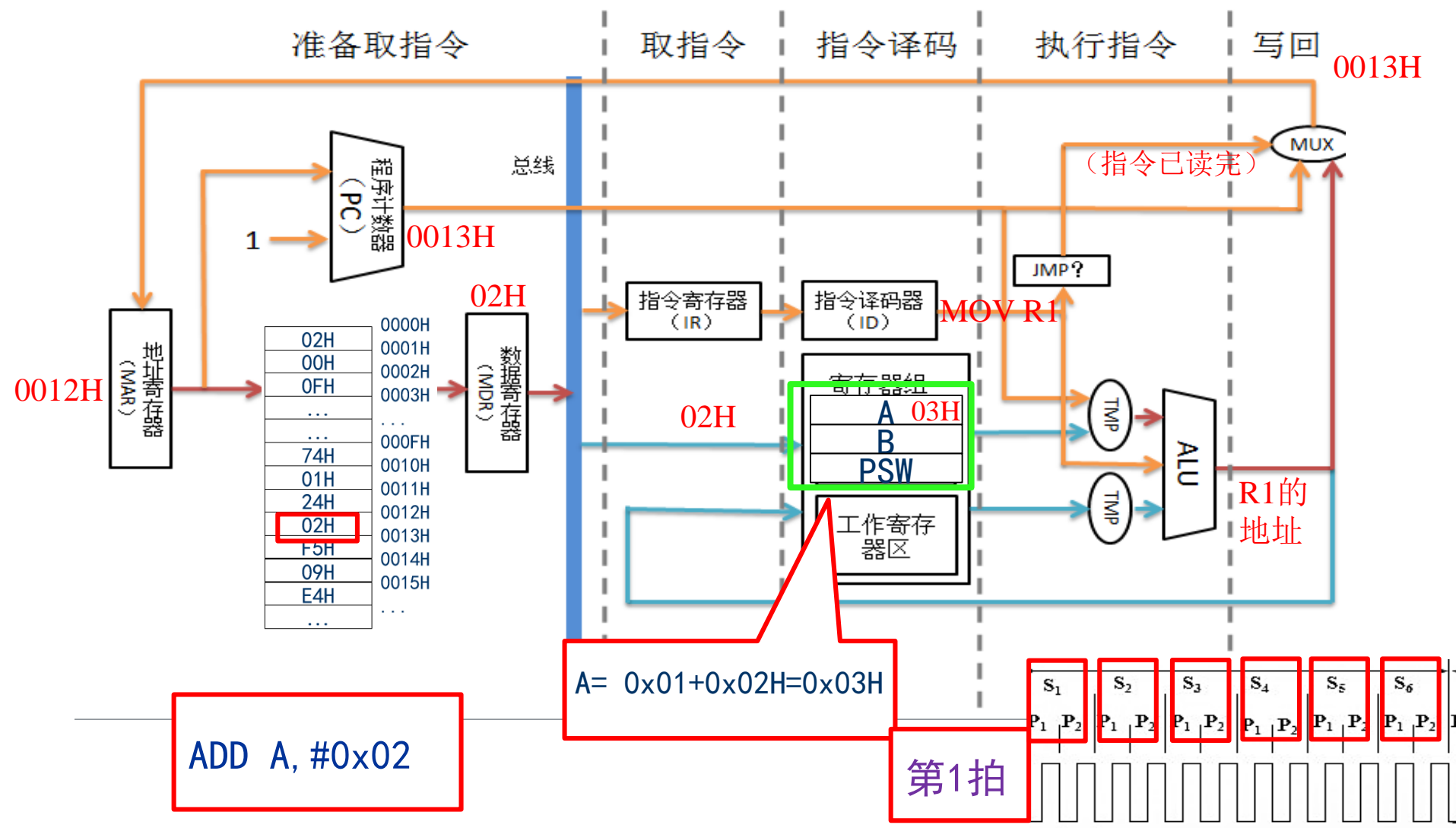
第二章 单片机的基本结构(8051)



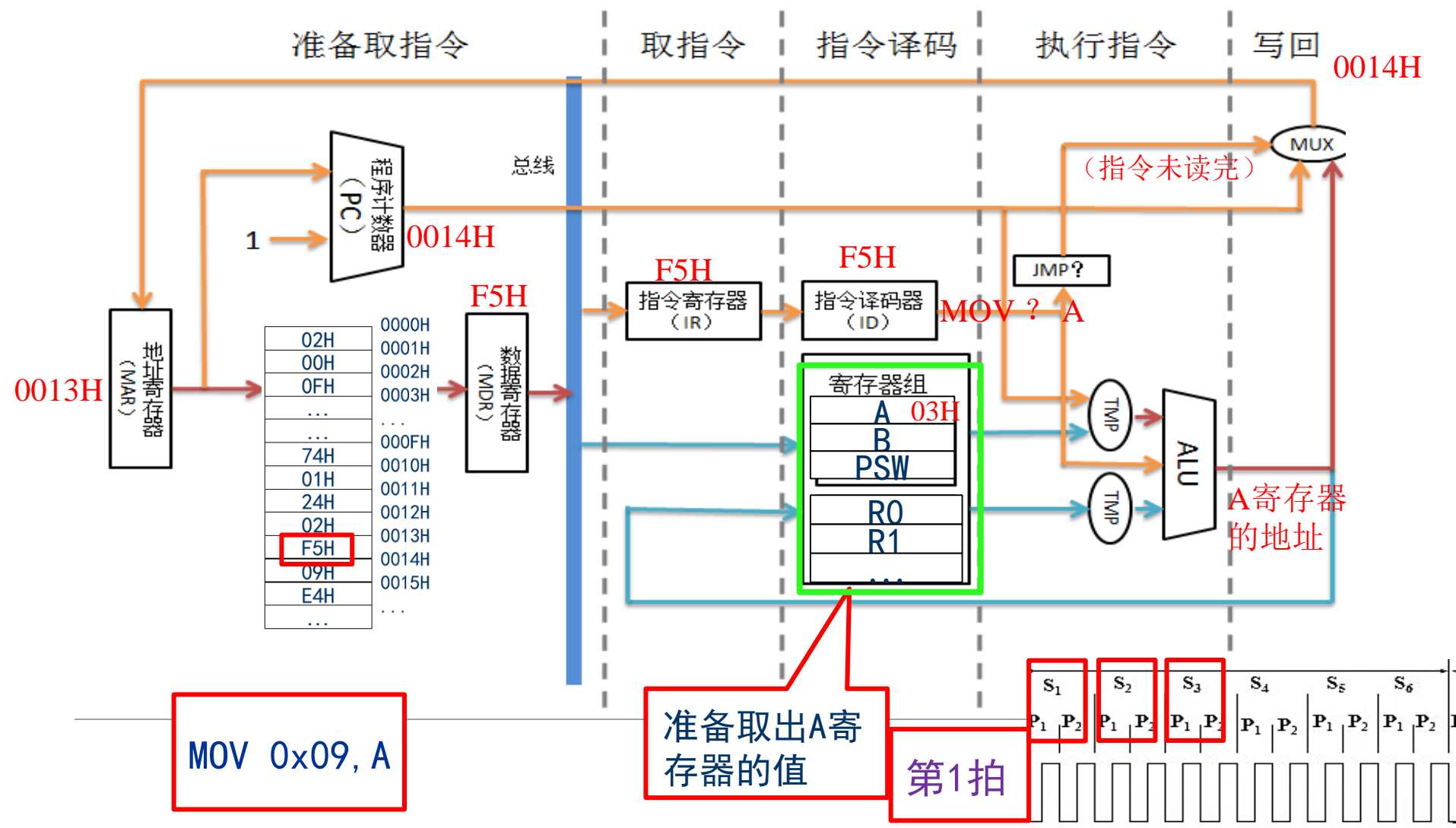
第二章 单片机的基本结构(8051)



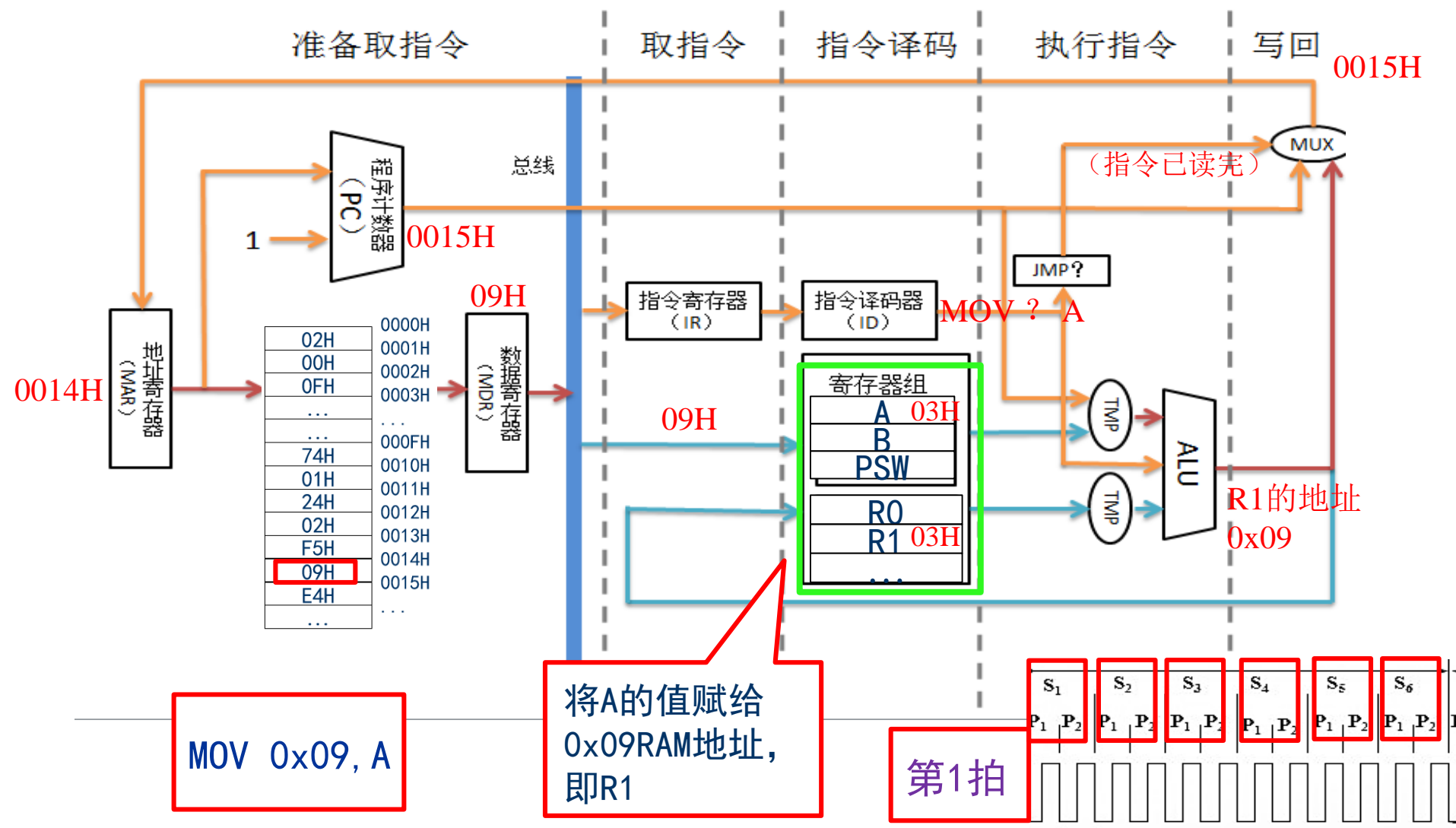
第二章 单片机的基本结构(8051)



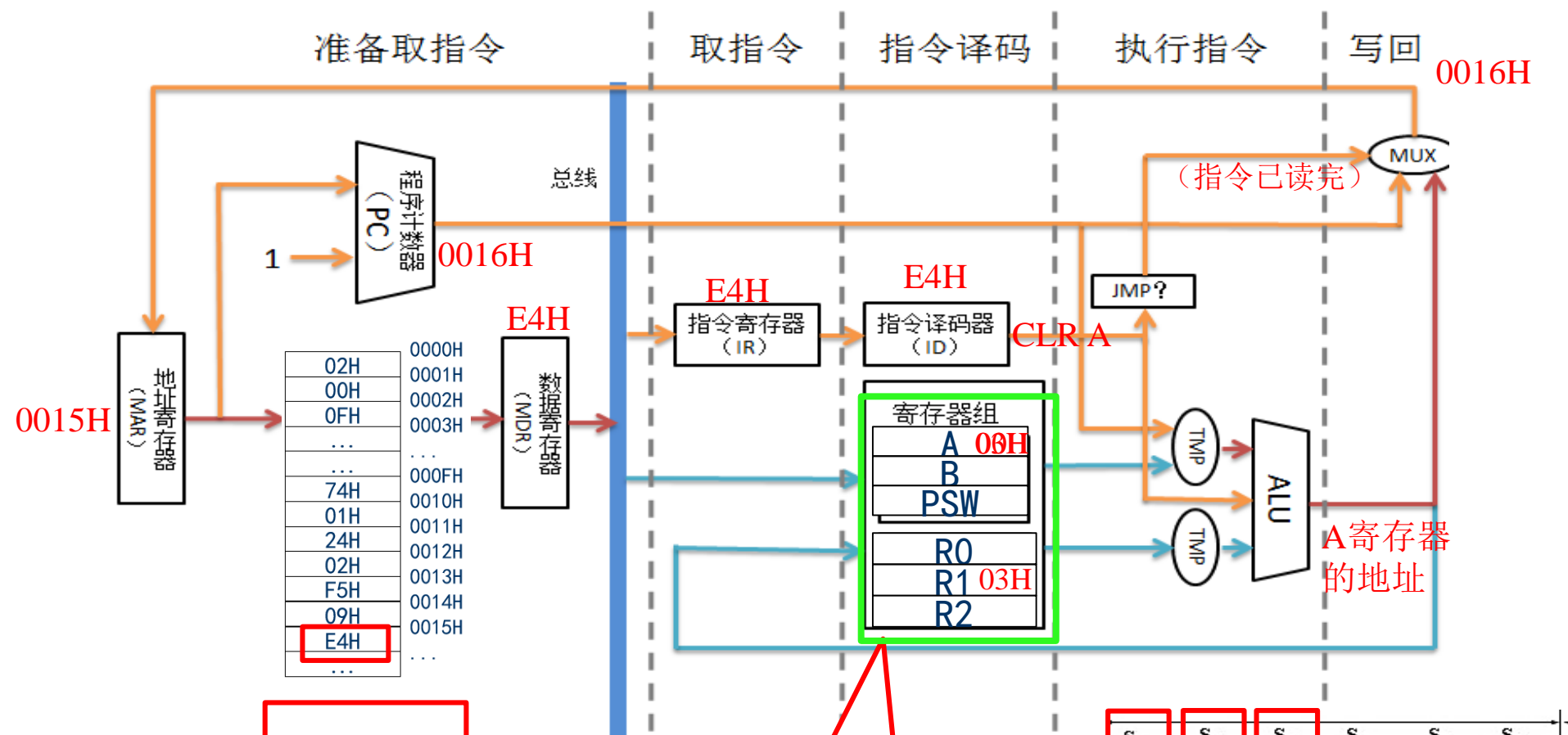
第二章 单片机的基本结构(8051)



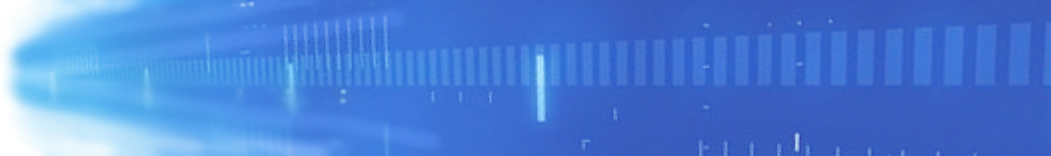
第二章 单片机的基本结构(8051)



第二章 单片机的基本结构(8051)



12个时钟周期为1个指令周期
1-2个指令周期执行1条指令



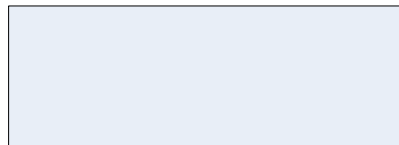
小结

■ 单片机的内部结构

- 冯·诺依曼结构，由CPU和存储器构成
- 存储器分为数据存储器 and 程序存储器
- 寄存器可控制所有资源

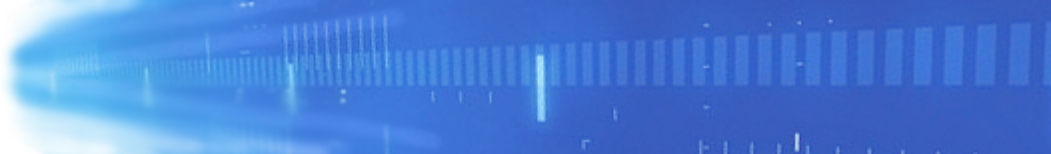
■ 工作过程

- 分为取指令、译码、执行三步
- 12个时钟周期为1个指令周期





北京航空航天大学
BEIHANG UNIVERSITY



再见