

简单CPU设计

智能1601-201608010312-张宁



相对简单CPU的规格说明

1. 64K字节的存储器，每个存储单元8位宽。

地址引脚A[15..0]

数据引脚D[7..0]

2. CPU的三个内部寄存器

◆ **8位累加器AC**：接受任何算术或者逻辑运算的结果，并为使用两个操作数的算术或者逻辑操作指令提供一个操作数。



◆ **寄存器R**：一个**8**位通用寄存器。它为所有的双操作数算术和逻辑运算指令提供第二个操作数。它也可以用来暂时存放累加器马上要用到的数据。

（减少存储器访问次数提高**CPU**的性能）

◆ **零标志位Z**：每次执行算术运算或者逻辑运算的时候，它都将被置位。

3. 指令集结构



指令	指令码	操作
NOP	0000 0000	无
LDAC	0000 0001 Γ	$AC \leftarrow M[\Gamma]$
STAC	0000 0010 Γ	$M[\Gamma] \leftarrow AC$
MVAC	0000 0011	$R \leftarrow AC$
MOVR	0000 0100	$AC \leftarrow R$
JUMP	0000 0101 Γ	GOTO Γ
JMPZ	0000 0110 Γ	IF ($Z=1$) THEN GOTO Γ
JPNZ	0000 0111 Γ	IF ($Z=0$) THEN GOTO Γ



ADD	0000 1000	$AC \leftarrow AC + R$, IF $(AC + R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
SUB	0000 1001	$AC \leftarrow AC - R$, IF $(AC - R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
INAC	0000 1010	$AC \leftarrow AC + 1$, IF $(AC + 1 = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
CLAC	0000 1011	$AC \leftarrow 0_a$, $Z \leftarrow 1$
AND	0000 1100	$AC \leftarrow AC \wedge R$, IF $(AC \wedge R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
OR	0000 1101	$AC \leftarrow AC \vee R$, IF $(AC \vee R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
XOR	0000 1110	$AC \leftarrow AC \oplus R$, IF $(AC \oplus R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
NOT	0000 1111	$AC \leftarrow AC'$, IF $(AC' = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$



4. 一些寄存器

- ◆ **16位的地址寄存器AR**：通过引脚A[15..0]向存储器提供地址。
- ◆ **16位的程序计数器PC**：存放的是将要执行的下一条指令的地址，或者指令需要的下一个操作数的地址。
- ◆ **8位的数据寄存器DR**：通过D[7..0]从存储器中接收指令和数据并且向存储器传送数据。
- ◆ **8位的指令寄存器IR**：存放的是从存储器中取出来的操作码。
- ◆ **8位的临时寄存器TR**：在指令执行过程中，临时存储数据。（程序员不能访问）

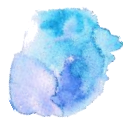


取指令

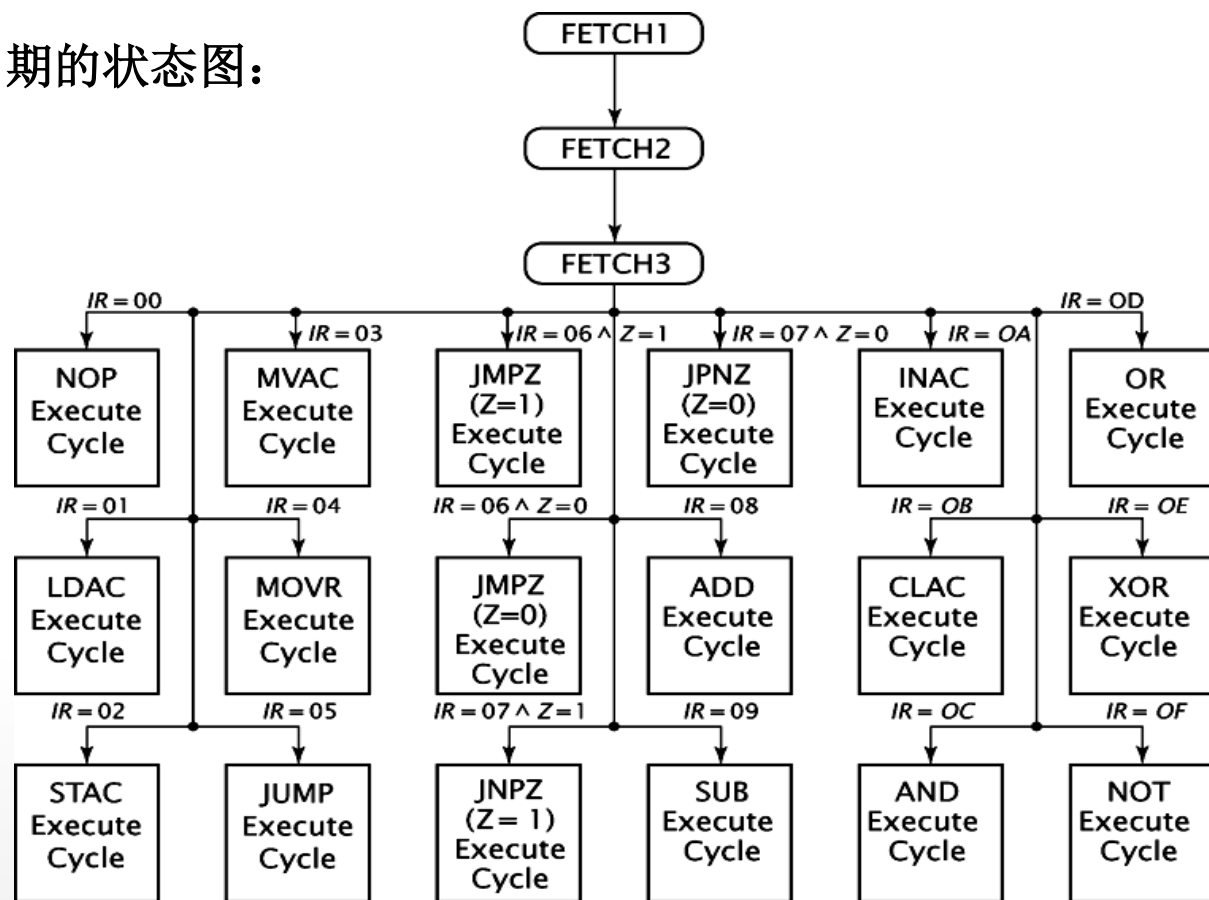
FETCH1: $AR \leftarrow PC$

FETCH2: $DR \leftarrow M, PC \leftarrow PC + 1$

FETCH3: $IR \leftarrow DR, AR \leftarrow PC$



取指令和译码周期的状态图：





执行指令

LDAC指令

- LDAC是一条多字指令。
- 它包含三个字：
- 操作码 地址的低半部分 地址的高半部分
- 功能：从存储器中获得地址，然后从存储器中获得数据，并把数据装载到累加器中。

F1: $AR \leftarrow PC$

F2: $DR \leftarrow M, PC \leftarrow PC + 1$

F3: $IR \leftarrow DR, AR \leftarrow PC$

L1: $DR \leftarrow M, PC \leftarrow PC + 1, AR \leftarrow AR + 1$

L2: $TR \leftarrow DR, DR \leftarrow M, PC \leftarrow PC + 1$

L3: $AR \leftarrow DR, TR$

L4: $DR \leftarrow M$

L5: $AC \leftarrow DR$



STAC指令

STAC指令执行的是与LDAC完全相反的操作。

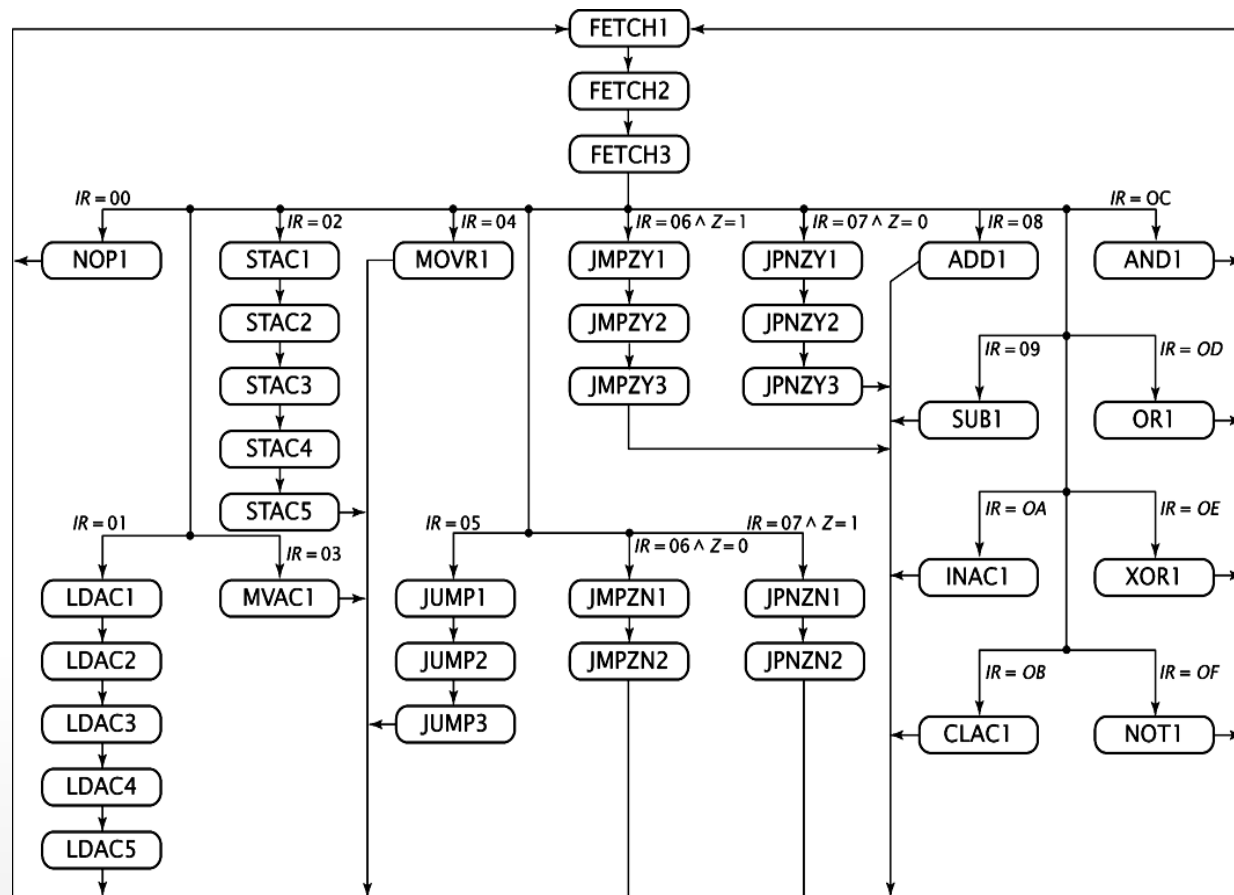
STAC1: $DR \leftarrow M, PC \leftarrow PC + 1, AR \leftarrow AR + 1$

STAC2: $TR \leftarrow DR, DR \leftarrow M, PC \leftarrow PC + 1$

STAC3: $AR \leftarrow DR, TR$

STAC4: $DR \leftarrow AC$

STAC5: $M \leftarrow DR$





相对简单ALU的设计

设计两个运算单元

- 算术运算单元
- 逻辑运算单元

使用一个多路选择器从正确的输出中选择数据送入**AC**。



- 1、明确CPU的功能、目的和基本规格
- 2、设计指令集结构
- 3、取指令、译码（画出状态图）
- 4、执行指令（明确指令的状态）
- 5、创建数据通路
- 6、设计ALU等
- 7、设计控制单元
- 8、产生CPU的状态



THANKS!

