**FLAG**

**ARM7的flag**

zero flag (Z)

negative flag(N)

carry flag(C )

overflow flag (V)

所有的ALU指令都有S后缀版：MOVS, ADDS, SUBS, RSBS, MULS, MLAS, ANDS, EORS, ORRS, BICS

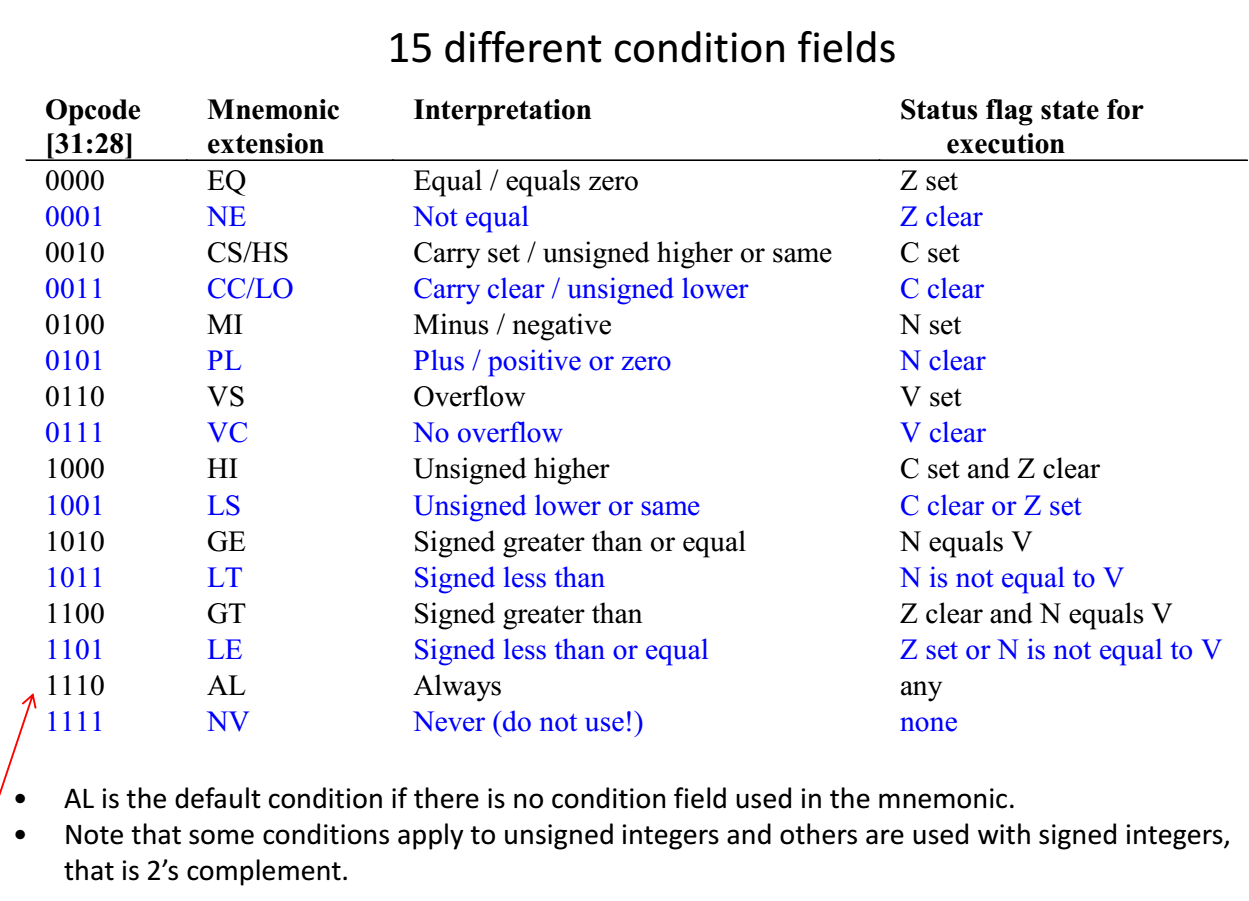
**Conditional Execution**

有些指令通过FLAG的状态来决定要不要执行。

MOVCS r12, #114

means move 114 into r12 if the **carry flag** is set.

Condition Fields



**The unconditional branch.**

Memory address Instruction

0x00000080 EOR r4, r5, r9

0x00000084 B 0x0000008C ; branch

0x00000088 AND r5, r9, r2

0x0000008C MLA r8, r6, r6, r2

0x00000090 B 0x00000088 ; branch

These instructions are executed in the following

order:

EOR,

B 0x8C,

MLA,

B 0x88,

AND,

MLA,

B 0x88,

AND,

MLA,

B 0x88, etc.

**Conditional branches**

分支指令中的内存地址必须包含在指令本身中

指令的32位中，有24位Bits被用来表示目标地址到目前program counter 的 offset

Program to add together 10 values from

memory 0x00000080 onwards:

用 ZERO FLAG 的例子：

MOV r0, #9 ; set loop counter

MOV r1, #0x080 ; set address for data

LDR r2, [r1], #4 ; load 1st value

loop SUBS r0, r0, #1 ; decrement counter

LDR r3, [r1], #4 ; load next value

ADD r2, r2, r3 ; add to running total

BNE loop ; branch back to loop if zero flag is cleared

**Setting the carry flag**

触发CARRY FLAG的指令

MOV r2, 0xF2000000

MOV r7, 0x11000000

ADDS r9, r2, r7

**Use of Carry Flag**

用作指令是否执行的条件

ADDCS r1, r1, #1 will only execute if the carry flag is set

‘BCC label’ will only branch if the carry flag is clear.

The other use of the carry flag is in the addition instruction ADC (and the subtraction instructions SBC and RSC)

ADC：add with carry. 将两个数加起来，如果Carry Flag is set，和再加一。

r0 holds 0xCB417800，r2 holds 0x42770C00，

ADDS r4, r2, r0：

carry flag is set, r4 储存低32位的数值0x0DB88400

**Negative numbers**

表示负数一般有两种方式：

符号位 Sign magnitude

补码 Two’s complement

m.s.b 表示符号，1：负数，0：正数

为了查看一个数是正数还是负数，我们使用：MOVS rx, rx

rx储存的值不会改变，但如果rx存的值是正数，则Negative Flag会被清零，否则会被设为1.

技巧：

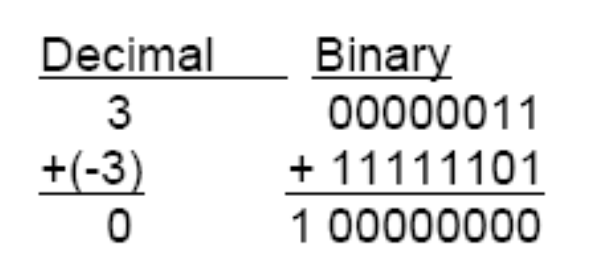
所有十六进制的数，如果以>=8的值开头，就是负数，否则是正数

取绝对值：与0x7FFFFFFF做AND操作

**2’s complement**

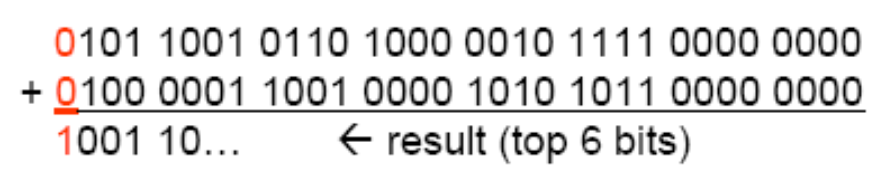
负数取反加一，自动把Negative Flag设置为1

用补码进行计算会比较简单：



**The Overflow flag**

overflow flag可以用来判断两个补码相加会不会越界



如果符号位在运算中被设为1了，overflow flag就会被设置为1

**Flags Summary**

Zero Flag 在结果为0x00000000时被设为1

Negative Flag 在32位的最高位为1时被设为1

Carry Flag 在无符号数的结果大于（2^31-1）时被设为1

Overflow Flag 在补码计算结果大于（2^31 -1）或者小于-2^31时被设为1

**IEEE 754**

5,637,144,57610 in IEEE 754

