**Mnemonics**

compiler把高级程序转成汇编或者是机器码，assembler把汇编码转成机器码

数据溢出时取低32位

算术指令

SUB rz, ry, rx ：表示z= y-x

RSB rz, ry, rx ：逆减，表示z=x-y

MLA rz, ry, rx, rw:  Multiply and accumulate  // z = y\*x+w

BIC rz, ry, rx  :  bit clear , ry  AND  NOT(rx) ，x为1的地方都得为0

AND rz,ry,rx 逻辑与

ORR rz,ry,rx 逻辑或

EOR rz,ry,rx 逻辑异或

立即数要用在指令的最后一位

寻址 Addressing Mode

Register Addressing

0xE1A06002: MOV r6 r2: 将r2寄存器的内容存到r6

Immediate Addressing

0xE3A0C0A0: MOV r12, #0xA0

为了表示大的立即数，一条指令只有32位，所以不能够直接表示

MOV rZ, #N×2^2(16-M) is 0xE3A0ZMNN （M：0-15，NN:0-255）

Indirect Addressing，间接寻址

LDR r6,[r11] : 到r11的内容所表示的地址处取值放到r6.

从寄存器取值到memory

STR r6,[r11] ：从r6中的数据存到r11内容表示的地址中

Little endian

就是least significant bit放在地址较低的位置上

LDRH、LDRB、STRH、STRB

分别只存储加载半个word、byte

Base plus offset addressing

LDR r6, [r11, #12]: 到r11 value + 12 所表示的地址处取值到r6

自动更新

LDR r6, [r11, #12]！: 到r11 value + 12 所表示的地址处取值到r6，同时更新r11的值为+12

Pre-indexed and post-indexed

LDR r6, [r11, #12]! （Pre-indexed）:r11在被用作取值地址前更新

LDR r6, [r11], #12 （post-indexed）:r11在被用作取值地址之后更新

RISC的指令和高级语言相距甚远（high semantic gap），所以它的compiler也更复杂，Code density（同等体积的代码表达的意思量）低