# 32位

**立即数**

是一块数据存储作为指令本身，就是一个值，类似C的 int num = 1030;

这个数放入内存(寄存器)之前叫立即数，放入之后就不是了

**逻辑数**

用二进制表示逻辑中的是与非，

**逻辑运算**

与或非，异或同或

**算数运算**

加减乘除

**数据操作指令**

**//使用方法**

**cmd Rd, Rn, Oper**

**操作码 目标寄存器 第一操作寄存器 第二操作数(也可以是寄存器)**

ADC 带进位加法

ADD 加法

ADDW 宽加法（可以加 12 位立即数）

AND 按位与

ASR 算术右移

BIC 位清零（把一个数按位取反后，与另一个数逻辑与）

BFC 位段清零

BFI 位段插入

CMN 负向比较（把一个数和另一个数的二进制补码比较，并更新标志位）

CMP 比较两个数并更新标志位

CLZ 计算前导零的数目

EOR 按位异或

LSL 逻辑左移

LSR 逻辑右移

MLA 乘加

MLS 乘减

MOVW 把 16 位立即数放到寄存器的底16位，高16位清0

MOV 加载16位立即数到寄存器（其实汇编器会产生MOVW——译注）

MOVT 把 16 位立即数放到寄存器的高16位，低 16位不影响

MVN 移动一个数的补码

MUL 乘法

ORR 按位或

ORN 把源操作数按位取反后，再执行按位或

RBIT 位反转（把一个 32 位整数先用2 进制表达，再旋转180度——译注）

REV 对一个32 位整数做按字节反转

REVH/REV16 对一个32 位整数的高低半字都执行字节反转

REVSH 对一个32 位整数的低半字执行字节反转，再带符号扩展成32位数

ROR 圆圈右移

RRX 带进位的逻辑右移一格（最高位用C 填充，且不影响C的值——译注）

SFBX 从一个32 位整数中提取任意的位段，并且带符号扩展成 32 位整数

SDIV 带符号除法

SMLAL 带符号长乘加（两个带符号的 32 位整数相乘得到 64 位的带符号积，再把积 加到另一个带符号 64位整数中）

SMULL 带符号长乘法（两个带符号的 32 位整数相乘得到 64位的带符号积）

SSAT 带符号的饱和运算

SBC 带借位的减法

SUB 减法

SUBW 宽减法，可以减 12 位立即数

SXTB 字节带符号扩展到32位数

TEQ 测试是否相等（对两个数执行异或，更新标志但不存储结果）

TST 测试（对两个数执行按位与，更新标志但不存储结果）

UBFX 无符号位段提取

UDIV 无符号除法

UMLAL 无符号长乘加（两个无符号的 32 位整数相乘得到 64 位的无符号积，再把积加到 另一个无符号 64位整数中）

UMULL 无符号长乘法（两个无符号的 32 位整数相乘得到 64位的无符号积）

USAT 无符号饱和操作（但是源操作数是带符号的——译注）

UXTB 字节被无符号扩展到32 位（高24位清0——译注）

UXTH 半字被无符号扩展到32 位（高16位清0——译注）

**存储器数据传送指令**

LDR 加载字到寄存器

LDRB 加载字节到寄存器

LDRH 加载半字到寄存器

LDRSH 加载半字到寄存器，再带符号扩展到 32位

LDM 从一片连续的地址空间中加载多个字到若干寄存器

LDRD 从连续的地址空间加载双字（64 位整数）到2 个寄存器

STR 存储寄存器中的字

STRB 存储寄存器中的低字节

STRH 存储寄存器中的低半字

STM 存储若干寄存器中的字到一片连续的地址空间中

STRD 存储2 个寄存器组成的双字到连续的地址空间中

PUSH 把若干寄存器的值压入堆栈中

POP 从堆栈中弹出若干的寄存器的值

**转移指令**

B 无条件转移

BL 转移并连接（呼叫子程序）

TBB 以字节为单位的查表转移。从一个字节数组中选一个8位前向跳转地址并转移

TBH 以半字为单位的查表转移。从一个半字数组中选一个16 位前向跳转的地址并 转移

**其他**

LDREX 加载字到寄存器，并且在内核中标明一段地址进入了互斥访问状态

LDREXH 加载半字到寄存器，并且在内核中标明一段地址进入了互斥访问状态

LDREXB 加载字节到寄存器，并且在内核中标明一段地址进入了互斥访问状态

STREX 检查将要写入的地址是否已进入了互斥访问状态，如果是则存储寄存器的字

STREXH 检查将要写入的地址是否已进入了互斥访问状态，如果是则存储寄存器的半字

STREXB 检查将要写入的地址是否已进入了互斥访问状态，如果是则存储寄存器的字节

CLREX 在本地的处理上清除互斥访问状态的标记（先前由 LDREX/LDREXH/LDREXB做 的标记）

MRS 加载特殊功能寄存器的值到通用寄存器

MRS R0，CPSR //传送CPSR的内容到R0

MRS R0，SPSR //传送 SPSR的内容到R0

MSR 存储通用寄存器的值到特殊功能寄存器

MSR CPSR，R0 //传送R0的内容到CPSR

MSR SPSR，R0 //传送R0的内容到SPSR

MSR CPSR\_c，R0 //传送R0的内容到SPSR，但仅仅修改CPSR中的控制位域

NOP 无操作

SEV 发送事件

WFE 休眠并且在发生事件时被唤醒

WFI 休眠并且在发生中断时被唤醒

ISB 指令同步隔离（与流水线和 MPU等有关——译注）

DSB 数据同步隔离（与流水线、MPU 和cache等有关——译注）

DMB 数据存储隔离（与流水线、MPU 和cache等有关——译注）

DMB 数据存储器隔离。DMB 指令保证： 仅当所有在它前面的存储器访问操作都执 行完毕后，才提交(commit)在它后面的存储器访问操作。

DSB 数据同步隔离。比 DMB 严格： 仅当所有在它前面的存储器访问操作都执行 完毕后，才执行在它后面的指令（亦即任何指令都要等待存储器访 问操作)

ISB 指令同步隔离。最严格：它会清洗流水线，以保证所有它前面的指令都执行完 毕之后，才执行它后面的指令。