1. 寄存器

寄存器是CPU中与程序员可以用指令读写的部件，不同的CPU，寄存器的个数、结构是不相同的。

FP帧指针寄存器，代表栈帧，通常指向一个函数的栈帧底部，表示一个函数栈的开始位置。

SP栈指针寄存器，代表函数，在函数调用中，SP指针永远指向栈顶，也就是最后一个被调用函数的栈区。该栈帧包含函数的所有变量，编译器就是使用这个指针分配空间给新变量的。

PC 程序计数器寄存器，代表语句，储存下一条程序指令的地址，如果修改它，就能改变程序的执行流。顺序执行的情况下 PC 存储当前指令地址 + 4 （32 位指令长度为 4）。其实就是指向每条汇编语句的ROM地址(.text代码段)，可以通过改变PC执行执行不同的语句。

IP寄存器，内部程序调用暂存寄存器，arm中ip只是个通用寄存器，用来 在函数间分析和调用时暂存数据，用于寻找临时变量的位置。

LR链接寄存器，存储于程序返回地址，负责记录父函数的调用点程序地址，函数返回时将会使用这个地址进行跳转。Call Stack 记录就是一系列在栈上被保护的 LR 寄存器值组成的。指向ROM中（.text代码段）父函数地址。这个链接寄存器主要用在函数A和函数B，A正在运行时，突然要调用B，那么就引了一个分支了，然后这个函数B去运行，运行完之后还是要返回到最初然后继续往下走，那么这个时候返回值应该要有个记录，这就是链接寄存器。

APSR/CPSR 应用程序状态寄存器/当前程序状态寄存器

SPSR 已存储程序状态寄存器，想象一个程序正在运行，这个程序当前状态正常，这个状态就先把它保存到CPS里面，这个时候突然发生异常，那么当前状态就应该变成异常，就把这个状态存到CPSR上面，但是异常处理完了之后，我们希望还是能够回到之前那个状态，但是这个时候当时的状态已经被清理掉了，这个时候我们就可以用SPSR把原来那个状态保存，这样当状态在发生改变的时候，要还原就可以去SPSR里面读取之前那个状态，这就是它们之间的关系，就类似有一个A的变量，给A这个变量赋了一个值，然后还赋了一个新值，但是又希望原来那个值要保存，所以有个变量B，然后把变量A赋给变量B。

PSP/MSP寄存器，MSP和PSP 的含义是Main\_Stack\_Pointer 和Process\_Stack\_Pointer,在逻辑地址上他们都是R13。PSP和MSP在物理上是有两个寄存器，但是软件只用一个SP也就是R13寄存器去表示。实际用的时候，通过配置control寄存器，去选择R13是用的MSP还是PSP。

Systick寄存器，系统定时器使用起来非常简单。它一共有4个寄存器：SYST\_CSR、SYST\_RVR、SYST\_CVR、SYST\_CALIB。