ARMv7指令集处理器一Cortex命名，A、R和M系列，A系列争对大型嵌入式设备，比如手机；R系列针对实时性系统；M系列争对单片机市场；

ARMv8-A全新指令集，支持64位指令集，向前兼容ARMv7.

1.小端存储模式

数据的高字节保存在内存的高地址中；

Int checkCPU\_Little(void)

{

Union w

{

Int a;

Char b;

}c;

c.a=1;

return (c.b == 1)

}

2.一条存储读写指令的执行过程

几点五级流水线：取指(IF)、译码(ID)、执行(EX)、数据内存访问(mem)和写回(WB);

现代处理器设计上都采用了超标量体系结构(Superscalar Architecture)和乱序执行技术(Out-of-Order，OOO)。

所有指令通过循序方式通过前端，并采用乱序的方式进行发射，然后乱序执行，最后用循序方式提交结果。

3.内存屏障(memory barrier)产生原因

内存乱序访问的出现是为了提高程序运行时的性能，内存乱序访问主要发生在

a.编译时，编译器优化导致内存乱序访问；

b运行时，多CPU间交互一起的内存乱序访问；

优化可能会违背程序员原始的代码逻辑，到时发生一些错误，编译时的乱序访问可以通过barrier()来规避

#define barrier() \_\_asm\_\_ \_\_volatile\_\_ (“” ::: “memory”)