2020.06．20

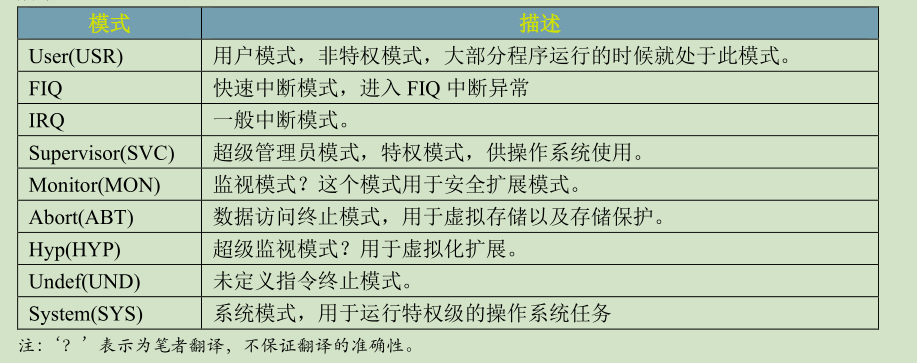
**1. usb host/usb otg 区别**

USB是一种数据通信方式，也是一种数据总线，而且是最复杂的总线之一。

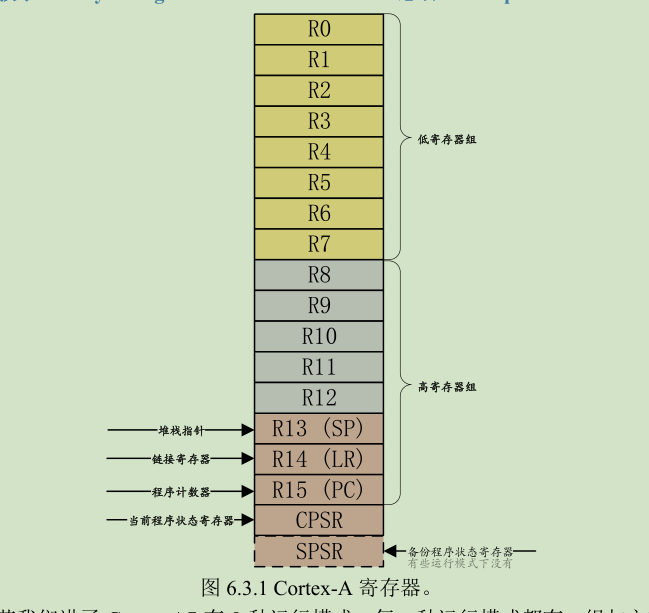
Host: 是由Host端控制整个总线的数据传输的。单个USB总线上，只能有一个Host。

Otg: On The Go，这是在USB2.0引入的一种mode，提出了一个新的概念叫主机协商协议（Host Negotiation Protocol），允许两个设备间商量谁去当Host。

2.cortex-A有9种运行模式



3.ARM构架提供16个32位通用寄存器(R0-R15)供软件使用，前15个（R0-R14）可以用作通用数据存储，R15是程序计数器PC，用来保存将要执行的指令。ARM 还提供了一个当前程序状态寄存器 CPSR 和一个备份程序状态寄存器 SPSR，SPSR 寄存器就是 CPSR 寄存器的备份。这18个寄存器如图所示：



4.程序计数器R15

程序计数器 R15 也叫做 PC，R15 保存着当前执行的指令地址值加 8 个字节，这是因为 ARM的流水线机制导致的。ARM 处理器 3 级流水线：取指->译码->执行，这三级流水线循环执行，比如当前正在执行第一条指令的同时也对第二条指令进行译码，第三条指令也同时被取出存放在 R15(PC)中。

5.GNU汇编语法

**label：instruction @ comment**

其中：

label是标号；

instruction 是指令；

@表示后面的内容是注释；

comment 是注释内容。

6.汇编系统预定义的一些段名（用.section伪操作来定义）

.text 表示代码段

.data 表示初始化的数据段

.bss 表示未初始化的数据段

.rodata 表示只读数据段

7.汇编默认的程序入口标号为\_start,不过也可以在链接脚本中使用ENTRY来指明其他的入口，下面的代码就是使用\_start作为入口：

.global \_start

\_start:

ldr r0,=0x12 @r0=0x12

上面代码中.global是伪操作，表示\_start是全局标号，类似C语言中的全局变量。常见的伪操作如下：

.byte 定义单字节数据，比如.byte 0x12

.short 定义双字节数据

.long 定义4字节数据

.equ 赋值语句，格式为:.equ 变量名，表达式 比如.equ num,0x12

.align 数据字节对齐，比如： .align 4 表示4字节对齐。

.end 表示源文件结束

.global 定义一个全局符号。

8.汇编语言函数格式

函数名：

函数体

返回语句

如：

/\* 未定义中断 \*/

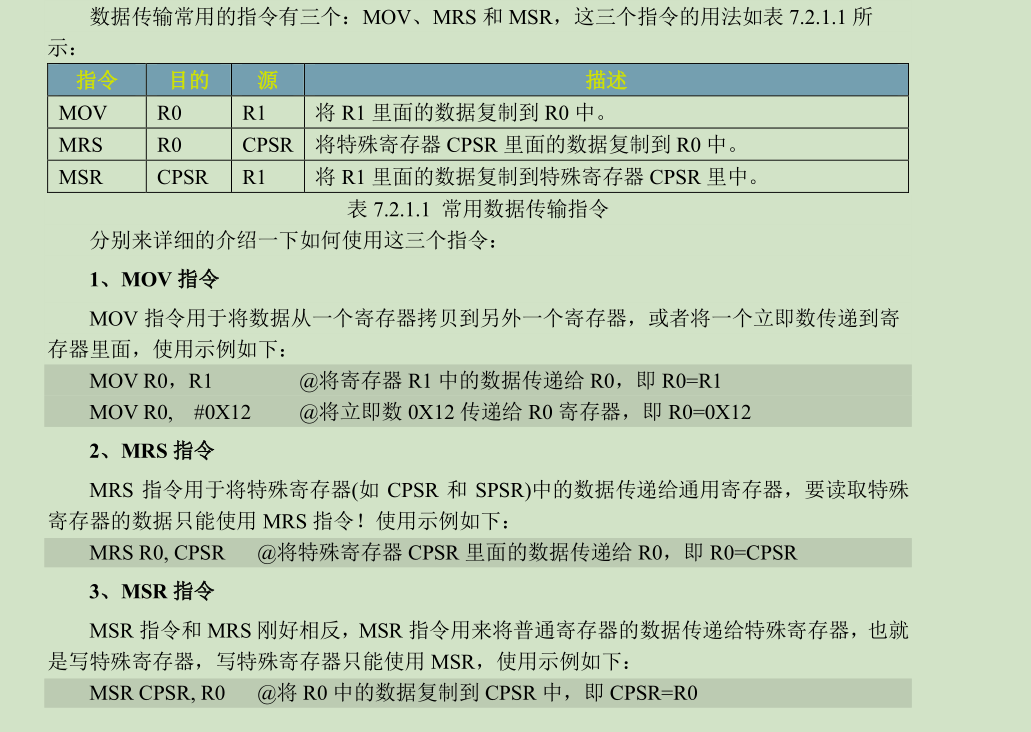
Undefined\_Handler: @函数名

ldr r0, =Undefined\_Handler @函数体

bx r0 @返回值

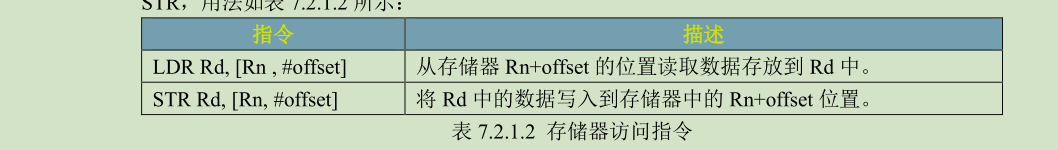
9.cortex-A7常用的汇编指令

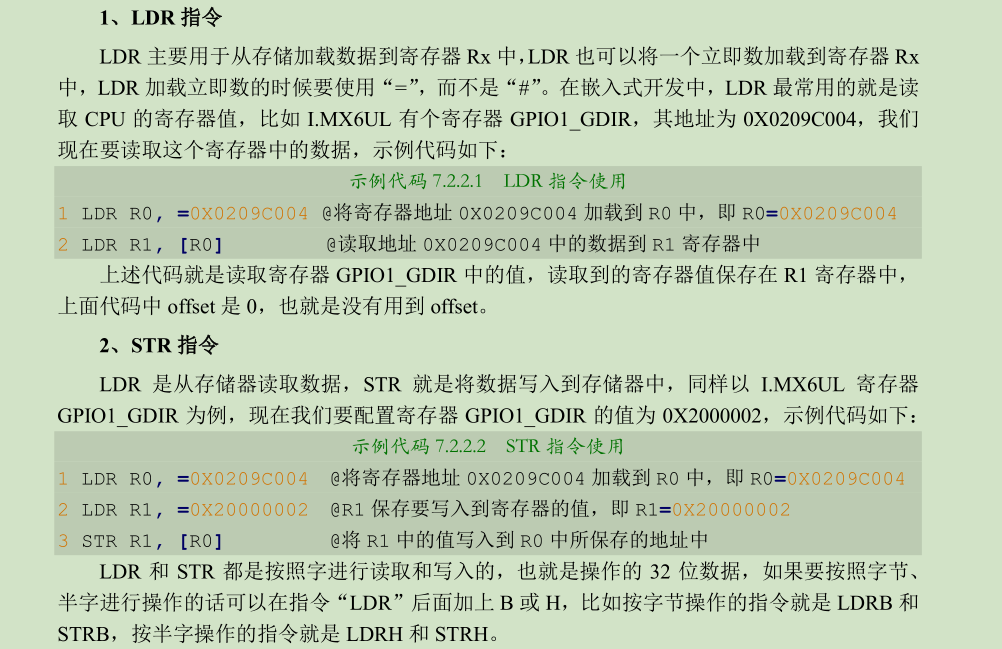
（1）处理器内部数据传输指令



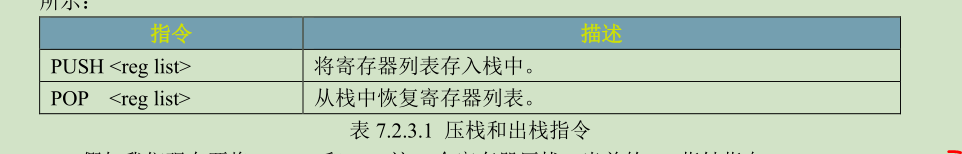
（2）存储器访问指令

ARM不能直接访问存储器。



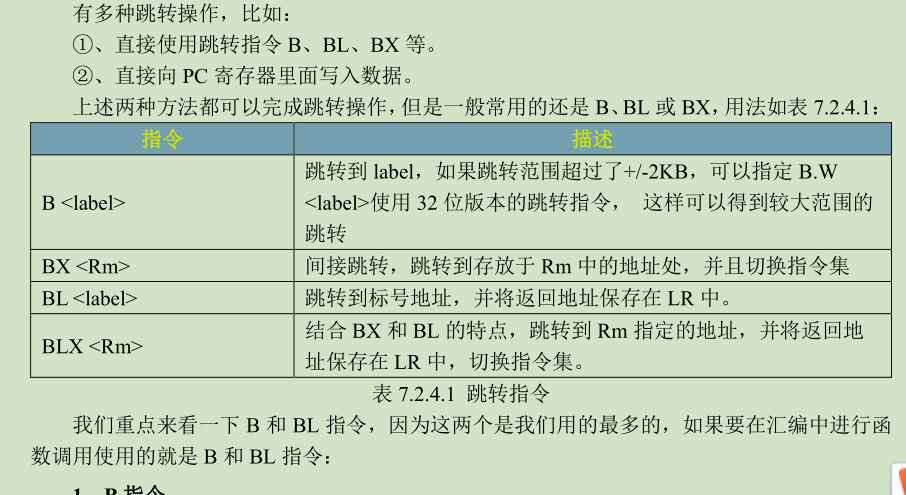


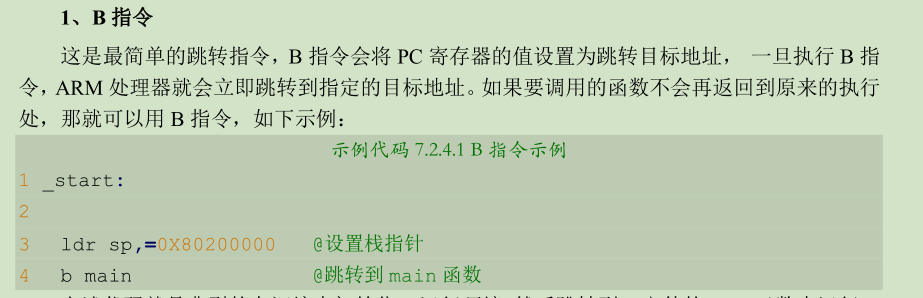
（3）压栈和出栈指令

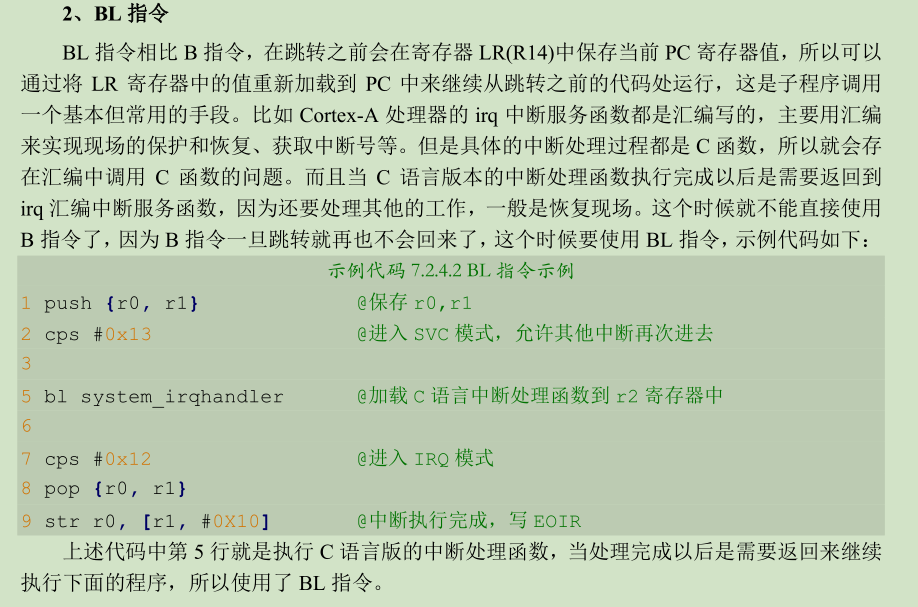


压栈和出栈可以同时操作多个寄存器，地址向下增长。

（4）跳转指令







（5）算术运算



（6）逻辑运算指令

