

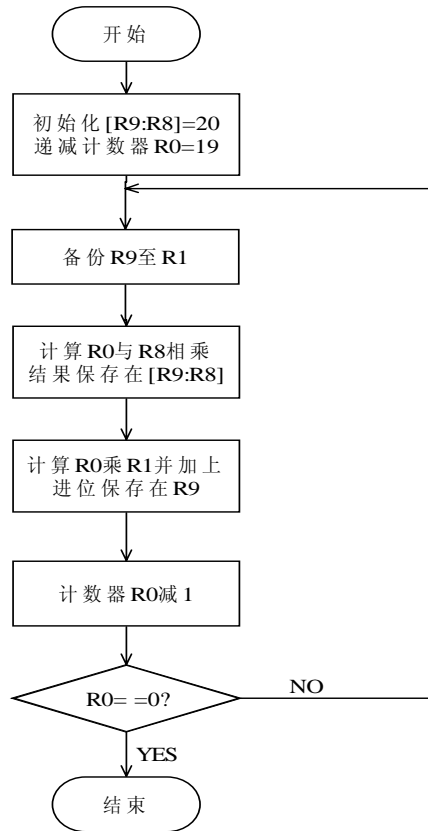
ARM 汇编语言程序设计精讲

阶乘操作实例解析

求一个数的阶乘（64 位结果）

用 ARM 汇编语言设计程序实现求 20!(即 20 的阶乘), 并将其 64 位结果放在[R9: R8] 中。(R9 中存放高 32 位)

解: 程序设计思路: 64 位结果的乘法指令通过两个 32 位的寄存器相乘, 可以得到 64 位的结果, 在每次循环相乘中, 我们可以将存放 64 位结果两个 32 位寄存器分别与递增量相乘, 最后将得到的高 32 位结果相加。程序设计流程如图 7-1 所示:



程序设计流程图

程序代码如下:

在 ARM 集成开发环境下编程 ADS:

AREA Fctrl, CODE, READONLY	; 声明代码段 Fctrl
ENTRY	; 标识程序入口
CODE32	; 声明 32 位 ARM 指令
START	
MOV R8, #20	; 低位初始化
MOV R9, #0	; 高位初始化
SUB R0, R8, #1	; 初始化计数器
Loop	
MOV R1, R9	; 暂存高位值
UMULL R8, R9, R0, R8	; [R9:R8]=R0*R8
MLA R9, R1, R0, R9	; R9=R1*R0+R9
SUBS R0, R0, #1	; 计数器递减
BNE Loop	; 计数器不为 0 继续循环
Stop	
B Stop	
END	; 文件结束

程序执行后输出结果如下:

R8=0x82B40000

R9=0x21C3677C