**汇编语言实现冒泡排序**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

存储器传输指令 MOVS MOV MRS

存储器加载指令 LDR LDRH LDRB LDRSH LDRSB

存储器保存指令 STR STRH STRB

多数据加载指令 LDM LDMIA LDMFD

多数据保存指令 STM STMIA STMEA

堆栈访问指令 PUSH POP

算数运算指令 ADDS ADD ADCS ADR SUBS SBCS RSBS SUB MULS CMP CMN

逻辑操作指令 ANDS ORRS EORS MVNS BICS TST

移位操作指令 ASRS LSRS RORS LSLS

反序操作指令 REV REV16 REVSH

扩展操作指令 SXTB SXTH UXTB UXTH

程序流控制指令 B B+\_ BL BX BLX

存储器屏障指令 DMB DSB ISB

异常相关指令 SVC CPS

休眠相关指令 WFI WFE SEV

其他： 空操作指令 NOP

断点指令BKPT

提示指令YIELD

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

多数据保存指令STM R0！, {R1, R2 – R7}，该指令的机器码格式为(C0FE)16 。该指令实现下面的功能：

（1）将寄存器R1的内容复制到寄存器R0所指向存储器地址的单元中。

（2）将寄存器R2的内容复制到寄存器R0+4所指向存储器地址的单元中。

（3）将寄存器R3的内容复制到寄存器R0+8所指向存储器地址的单元中。

……

（7）将寄存器R7的内容复制到寄存器R0+24所指向存储器地址的单元中。

（8）**用R0+7\*4的值更新R0寄存器的值。**【这就是感叹号的作用】

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

多数据加载指令LDM R0! , {R1, R2 – R7}，该指令的机器码格式为(C8FE)16 。该指令实现下面的功能：

（1）将寄存器R0所指向存储器地址单元的内容复制到寄存器R1中。

（2）将寄存器R0+4所指向存储器地址单元的内容复制到寄存器R2中。

（3）将寄存器R0+8所指向存储器地址单元的内容复制到寄存器R3中。

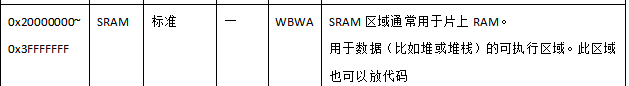
……

（7）将寄存器R0+24所指向存储器地址单元的内容复制到寄存器R7中。

（8）**用R0+4\*7的值更新寄存器R0的内容**。【这就是感叹号的作用】

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

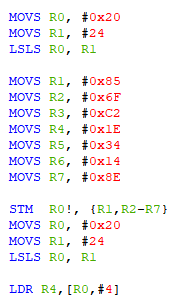
**第一步**：注释掉启动引导代码跳转到主函数的命令

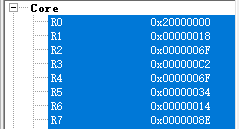
**第二步**：找一个位置存储数据，如下图

SRAM区域用于存放需要排序的数据。

**第三步**：让R0指向此地址，在寄存器R1-R7放入待排序数据

**第四步**：用多数据保存指令将寄存器R1-R7数据导入存储器，R0指向地址复位0x2000 0000



****

此处用R4寄存器读取0x2000 0004位置数据验证

结果R4=R2正确，可进行下一步工作

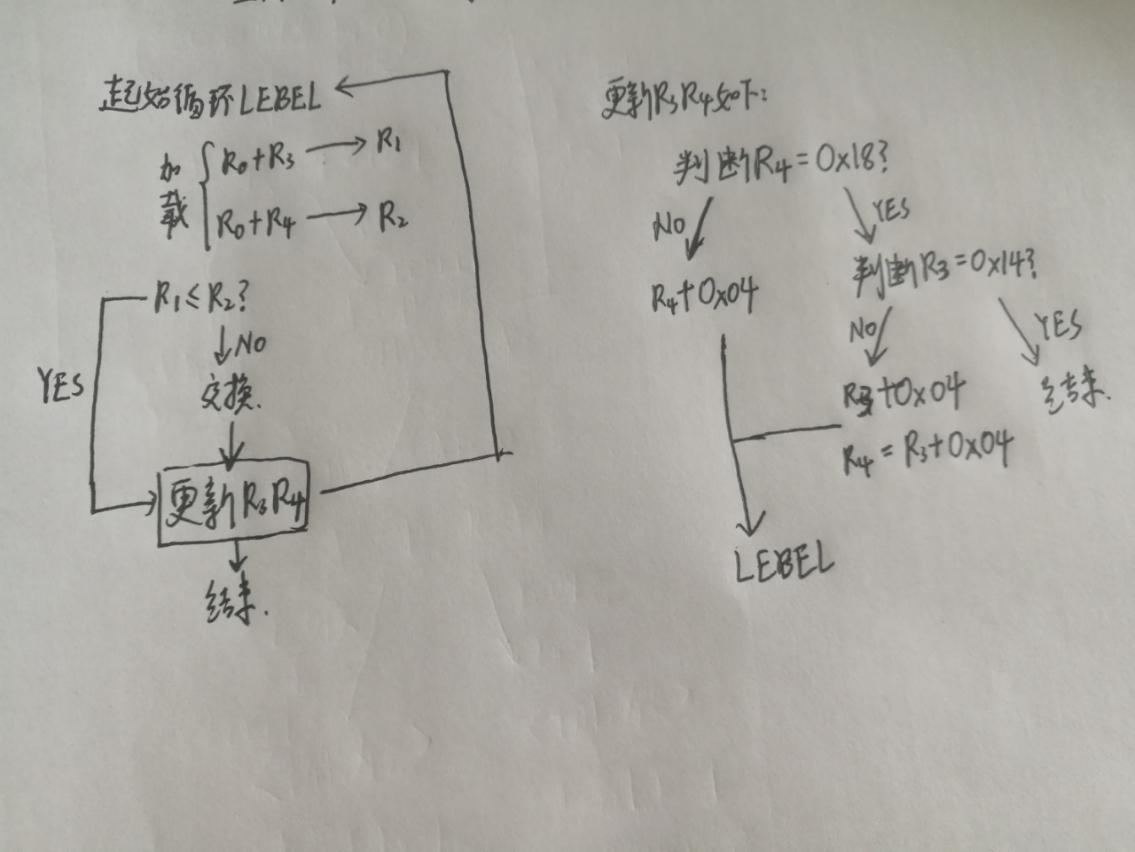
**第五步**：排序部分的设计思路

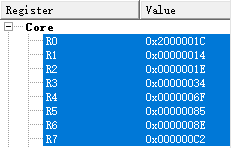
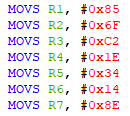
初始化：R0寄存器存储基址0x2000 0000

R3\R4寄存器存储两个比较值的偏移地址

同时用R3和R4控制冒泡排序的循环

循环排序部分如下图：



**第六步**：验证数据在存储器中的排序

**第七步**：设计代码

MOVS R0, #0x20

MOVS R1, #24

LSLS R0, R1

MOVS R1, #0x85

MOVS R2, #0x6F

MOVS R3, #0xC2

MOVS R4, #0x1E

MOVS R5, #0x34

MOVS R6, #0x14

MOVS R7, #0x8E

STM R0!, {R1,R2-R7}

MOVS R0, #0x20 ;R0 back

MOVS R1, #24

LSLS R0, R1

MOVS R3, #0x00 ;register initial

MOVS R4, #0x04

LEBEL ;begin cycle

LDR R1, [R0,R3]

LDR R2, [R0,R4]

CMP R1, R2

BLS LEBEL1

STR R1, [R0,R4] ;memory exchange

STR R2, [R0,R3]

LEBEL1

CMP R4, #0x18

BEQ LEBEL2

ADDS R4, #0x04

B LEBEL

LEBEL2

CMP R3, #0x14

BEQ LEBEL3

ADDS R3, #0x04

MOVS R4, R3

ADDS R4, #0x04

B LEBEL

LEBEL3

LDM R0!, {R1, R2-R7} ;read date