AREA Fctrl,CODE,READONLY ; declare Fctrl

ENTRY ;

CODE32 ; declare 32 b ARM

START

LDR R6, =0x4000 ; A[] frist address sent to r6

LDR R7, =0x5032 ; B[] frist address sent to r7

MOV R9 , #0 ; init temp

MOV R8 , #8 ; init i

Loop ; for swap

LDR R9 , [R6] ;temp save value

SWP R9, R9, [R7] ; SAVE [R7] IN R9, SAVE [R6] IN R7

STR R9, [R6] ; SAVE [R7] IN [R6]

ADD R6,#4;

ADD R7,#-4;

SUB R8,R8,#1 ; i--

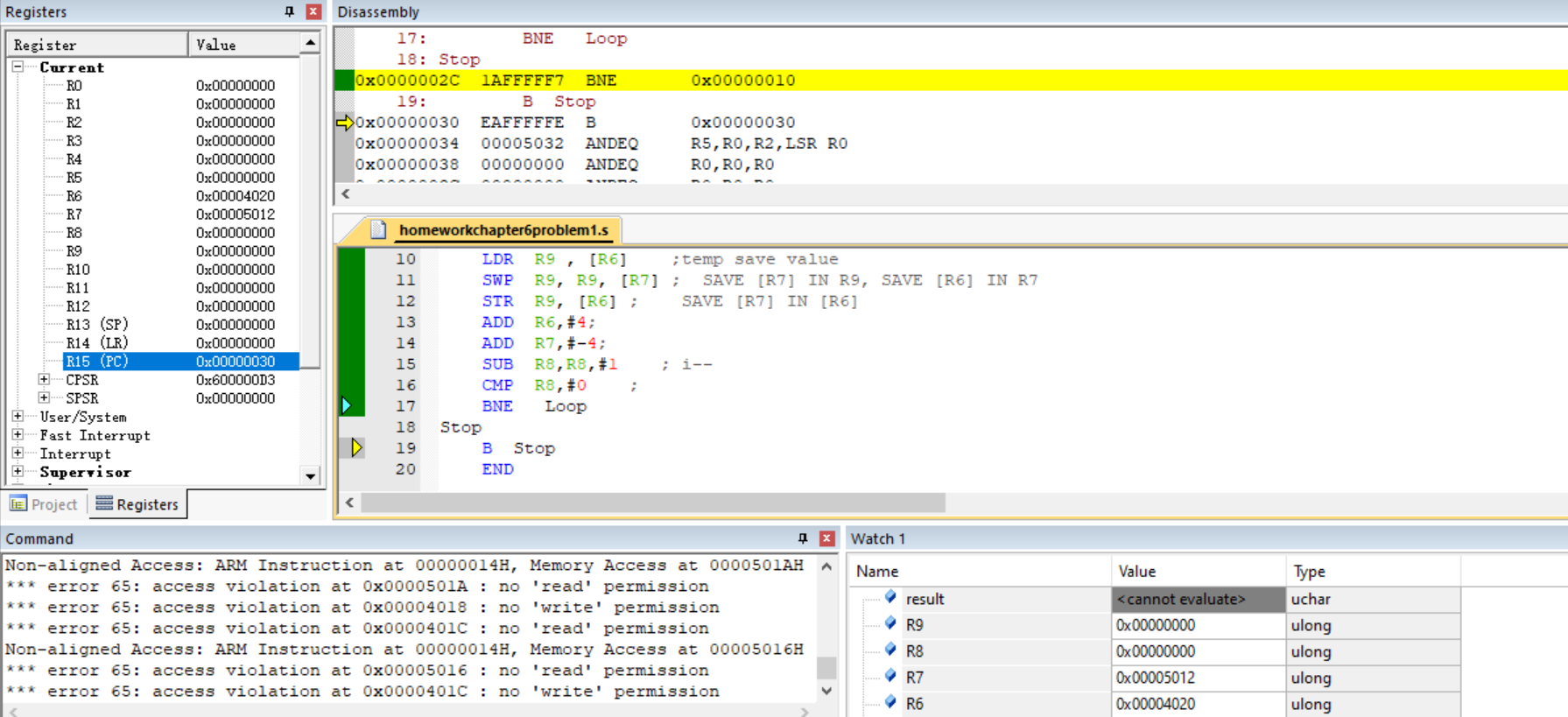
CMP R8,#0 ;

BNE Loop

Stop

B Stop

END



第六题:

代码如下:

AREA Fctrl,CODE,READONLY ; declare Fctrl

ENTRY ;

CODE32 ; declare 32 b ARM

START

MOV R6, #0 ; save sum

MOV R7, #5 ; R7 from 5 begin

loop2 ;this is FOR i--

MOV R8 , R7 ; init low bit , such as calculate 4!, R7 give 4 to R8

MOV R9 , #0 ; init high bit

SUB R0,R8,#1 ; init R0

Loop ; for calculate i!

MOV R1 , R9 ;temp save high bit value

UMULL R8 , R9 , R0 , R8 ;[R9:R8]=R0\*R8

MLA R9 , R1 , R0 , R9 ; R9=R1\*R0+R9

SUBS R0 , R0 , #1 ; R0 from 9 to 0 loop

BNE Loop ;if not 0 continue loop ,dont' need cmp is ok . SUBS S will influence CPSR, compare with zero

ADD R6, R8,R6 ; SUM =SUM + I!

SUBS R7, R7 ,#1 ;

BNE loop2

Stop

B Stop

END

结果保存在R6

