**CompE-271**

* I declare that all material in this assignment is my own work except where there is clear reference to the work of others.
* I have read, understood and agree to the SDSU Policy on Plagiarism and Cheating on the university website at <http://go.sdsu.edu/student_affairs/srr/cheating-plagiarism.aspx> , the syllabus and the student-teacher contract for the consequences of plagiarism, including both academic and punitive sanctions.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Remark\*. By submitting this assignment report electronically, you are deemed to have signed the declaration above.*

12/9/2019

[HW#X2-10]

[#X2-10]

Ckick below to enter/change your Name and RedID

Ethan Nagelvoort, 821234668

**Content**

(\* - Mandatory)

1\*. Description of the problem/method

I first load the bottom portion of x into r2 and load the upper portion of x into r3. I then load the bottom portion of y into r4 and load the upper portion of y into r5. Then, in order to properly switch the doubles, I store r2 into the bottom portion of y and r3 into the upper portion of y. I also store r4 into the bottom of x and r5 into the upper portion of x. This will properly swap the bits between the two doubles, thus swapping the doubles.

2. Pseudocode (if required. Mandatory for the Lab assignments, starting from #5 and Projects)

doubleswap:

save stack pointer into register r12

reserve 32 bytes of space for local variables

r0 = x, r1 = y

load the bottom portion of x into r2

load the top portion of x into r3

load the bottom portion of y into r4

load the top portion of y into r5

store r2 into the bottom portion of y

store r3 into the top portion of y

store r4 into the bottom portion of x

store r5 into the top portion of x

restore the stack pointer -- Please note stack pointer should be equal to the

value it had when you entered the function .

return from the function by copying link register into program counter

3\*. C-code

//main file

void doubleswap(double \*x, double \*y);

#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

void main(void) {

double x, y;

x = 53.299;

y = 66.998;

double \*xPtr = &x;

double \*yPtr = &y;

printf("Before placed into the function, X= %lf and Y= %lf \n", x, y);

doubleswap(xPtr, yPtr);

printf("After placed into the function, X= %lf and Y= %lf \n", x, y);

return(0);

}

//function file

.global doubleswap

.data

// declare any global variables here

.text

doubleswap:

mov r12,r13 // save stack pointer into register r12

sub sp,#32 // reserve 32 bytes of space for local variables

//push {r2} // push link register onto stack -- make sure you pop it out before you return

// r0 = x, r1 = y

ldr r2,[r0] // load the bottom portion of x into r2

ldr r3,[r0,#4] // load the top portion of x into r3

ldr r4,[r1] // load the bottom portion of y into r4

ldr r5,[r1,#4] // load the top portion of y into r5

str r2,[r1] // store r2 into the bottom portion of y

str r3,[r1,#4] // store r3 into the top portion of y

str r4,[r0] // store r4 into the bottom portion of x

str r5,[r0,#4] // store r5 into the top portion of x

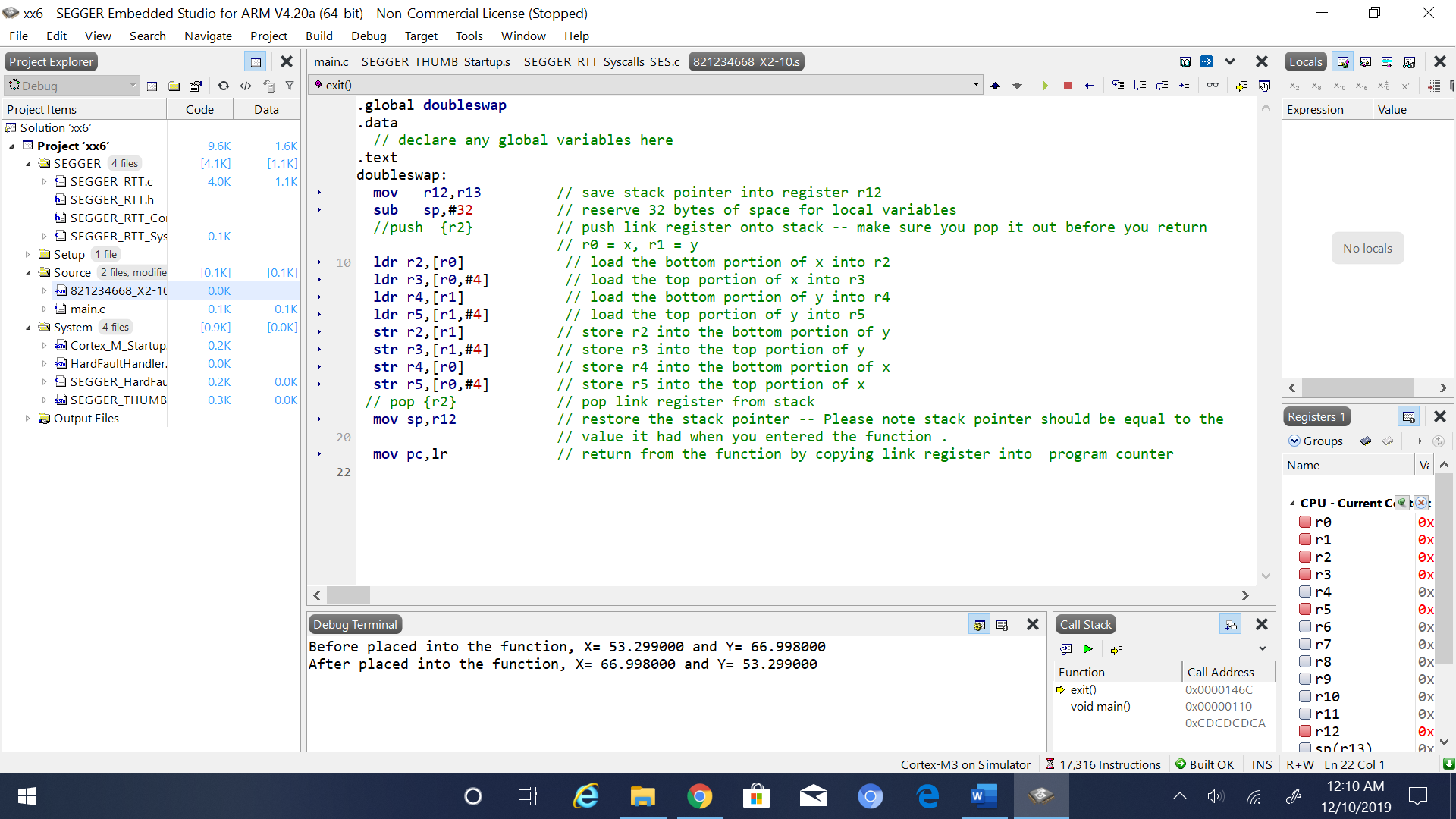
// pop {r2} // pop link register from stack

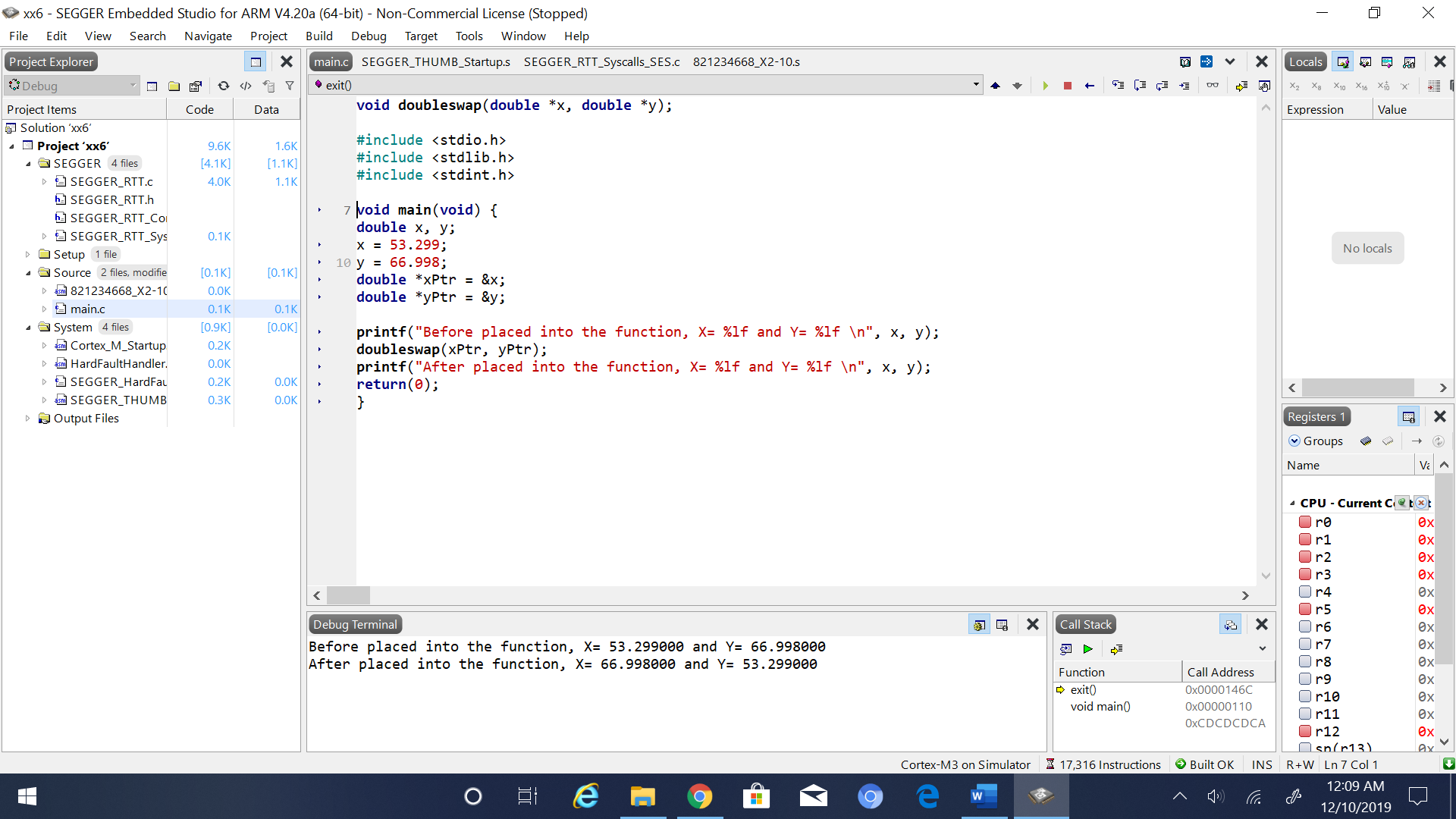
mov sp,r12 // restore the stack pointer -- Please note stack pointer should be equal to the

// value it had when you entered the function .

mov pc,lr // return from the function by copying link register into program counter

4\*. Screen capture of the code and the resulting display(s)





5. Conclusion (if applicable)

6\*. References.

Ken Arnolds slides on arm code