**Programming Exercise 1**

Graphical user interface, application

Description automatically generated

**Source Code View**

The pointer from the source code window shows the highlighted line of code that halts the program.

**Disassembly View**

The pointer from the disassembly window shows the highlighted line of code that halts the program.

**Registers View**

The pointer from the register window shows the highlighted registers affected.

**Memory View**

The red highlighted segment of memory in the memory view shows where the value from the program Is stored. The pointer simply points in general to the memory view.

**Expressions View**

The pointer to the expressions view shows the value of VarA.

**Programming Exercise 2**

*Source Code*

// Test program

.syntax unified

.text

.global main

main: // Register contents after instruction:

// My prediction Actual value

mov r0,#10 // r0 =\_\_\_\_\_\_\_10\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_10\_\_\_\_\_\_

ldr r1,=pp // r1 =\_\_\_0x08000298\_\_ \_\_\_\_\_\_134218336\_\_\_\_\_\_\_

ldr r2,[r1] // r2 =\_\_\_0x12345678\_\_ \_\_\_\_\_\_\_305419896\_\_\_\_\_\_\_

mov r1,r0 // r1 =\_\_\_\_\_\_\_10\_\_\_\_\_\_ \_\_\_\_\_\_\_10\_\_\_\_\_\_\_

add r3,r2,r0 // r3 =\_\_\_\_305419896\_\_\_\_\_\_ \_\_\_\_\_\_\_305419896\_\_\_\_\_\_\_

ldr r1,=fr // r1 =\_\_\_\_0b01000001\_ \_\_\_\_\_\_\_134218336\_\_\_\_\_\_\_

ldr r2,=ir // r2 =\_\_134218336\_\_\_\_ \_\_\_\_\_\_\_134218336\_\_\_\_\_\_\_

ldr r3,[r1] // r3 =\_\_\_-1090518975\_\_\_ \_\_\_\_\_\_-1090518975\_\_\_\_\_\_\_

ldr r4,=tmp // r4 =\_\_\_\_\_0x0800024C\_ \_\_\_\_\_\_\_134218336\_\_\_\_\_\_\_

ldr r5,[r2] // r5 =\_\_\_\_\_0x1234\_\_\_\_ \_\_\_\_\_\_\_460293\_\_\_\_\_\_\_

ldr r1,=512 // r1 =\_\_\_\_\_512\_\_\_\_\_\_ \_\_\_\_\_\_\_512\_\_\_\_\_\_\_

add r2,r1,r5 // r2 =\_\_\_\_460293\_\_\_\_\_ \_\_\_\_\_\_\_460293\_\_\_\_\_\_\_

sub r0,r6,r6 // r0 =\_\_\_\_\_\_\_0\_\_\_\_\_\_\_ \_\_\_\_\_\_\_0\_\_\_\_\_\_\_

b . // Equivalent to Here b Here

// Constant data in the CODE AREA - these follow the last instruction code

.align

pp: .word 0x12345678

ir: .byte 5,6,7

.align

ts: .word 0x0805

fr: .byte 'A'

.align

reg: .word 18,-20,15,-10

mem: .space 8

tmp: .word 0x1234,0x5678

// No DATA AREA for this program – above data is all in the CODE AREA

.end

***The highlighted area shows steps 2 and 5 for this exercise.***

**Programming Exercise 3**

*Source Code*

/\* Jonathan Elder CPSC 3300

Program 3 of Assignment 2.

3/24/2023

\*/

// Code section - to begin following the startup code.

// .section .text.main

.syntax unified

**.text**

**.global** main

**main:** MOV r0,#8 // value of k

MOV r1,#19 // value of n

MOV r3,r0

MOV r4,r1

MOV r2,#5 // value of p

ADD r2,r4

ADD r0,r2

ADD r0,r4

**Here:** b Here // Effectively halts the program.

// No Data Section for this project assignment.

.end

Graphical user interface, application

Description automatically generated

**Shows the values for K, N, and P respectively with respect to r0, r1, and r2. Above the image is the code for this assignment piece.**