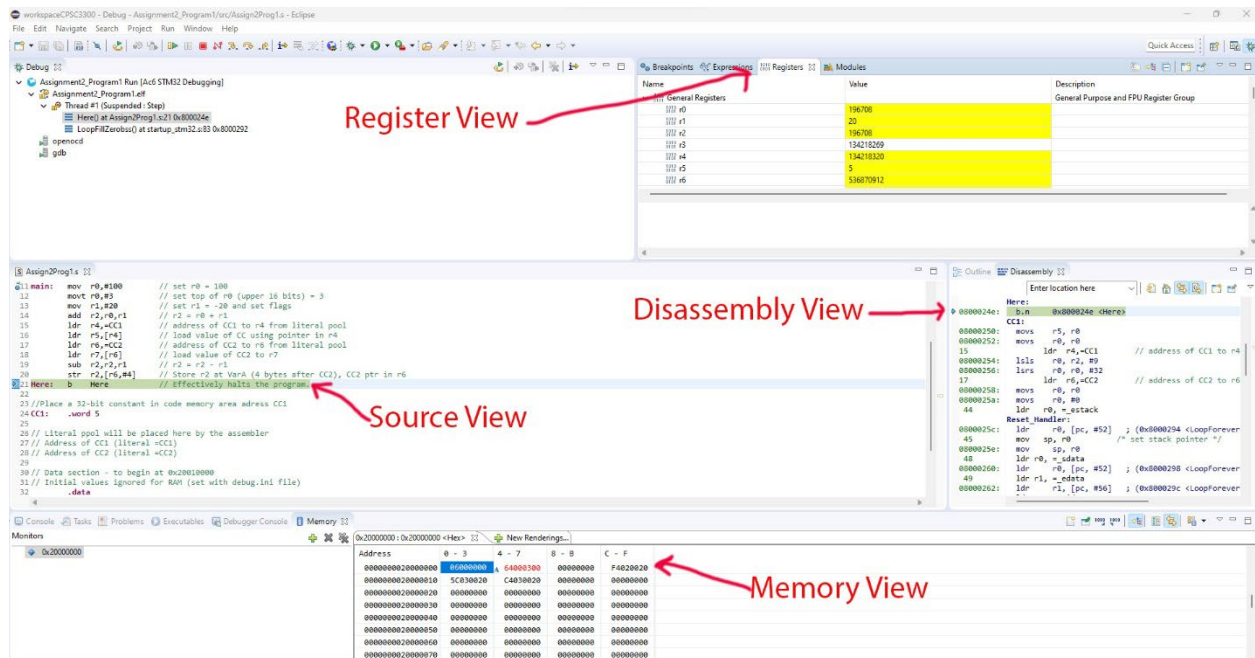


Programming Exercise 1



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

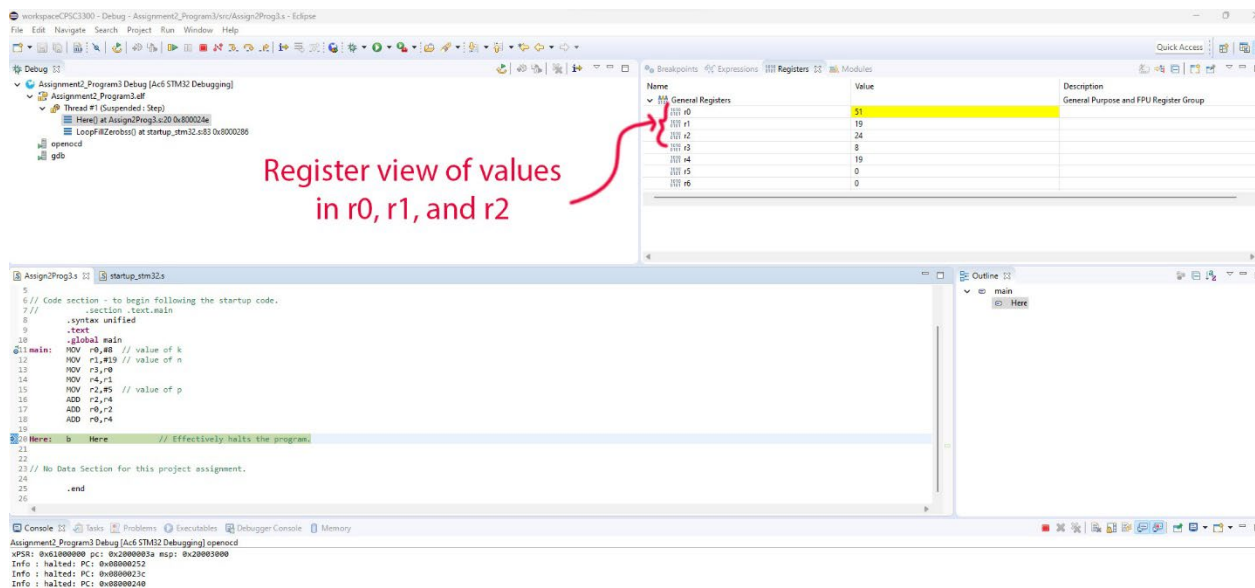
```
/* Jonathan Elder CPSC 3300
   Program 3 of Assignment 2.
   3/24/2023
*/

// 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
```



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.