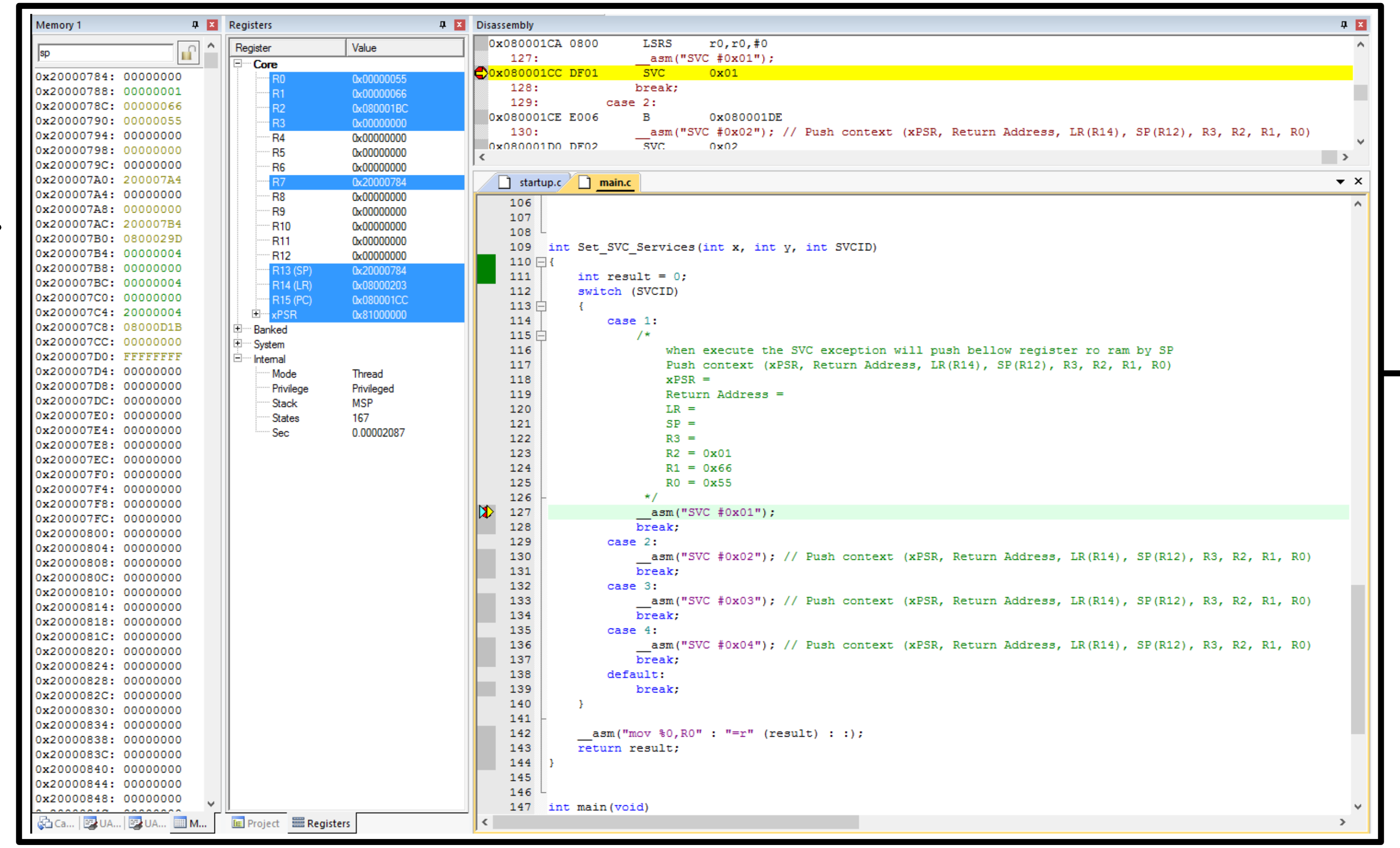
We Call the function Set_SVC_Services and pass three parameters to it.
These parameters are saved in registers as shown in the image.

```c
int main(void)
{
    int ret = 0;
    ret = Set_SVC_Services(0x55,0x66,1); // R0 = 0x55, R1 = 0x66, R2 = 0x01
    ret = Set_SVC_Services(0x22,0x11,2); // R0 = 0x22, R1 = 0x11, R2 = 0x02
    ret = Set_SVC_Services(0x53,0x64,3); // R0 = 0x53, R1 = 0x64, R2 = 0x03
    ret = Set_SVC_Services(0x57,0x69,4); // R0 = 0x57, R1 = 0x69, R2 = 0x04
    while (1)
    {
        /* code */
    }


    return 0;
}
```

when execute the SVC exception will push bellow register ro ram by SP Push context (xPSR, Return Address, LR(R14), SP(R12), R3, R2, R1, R0)

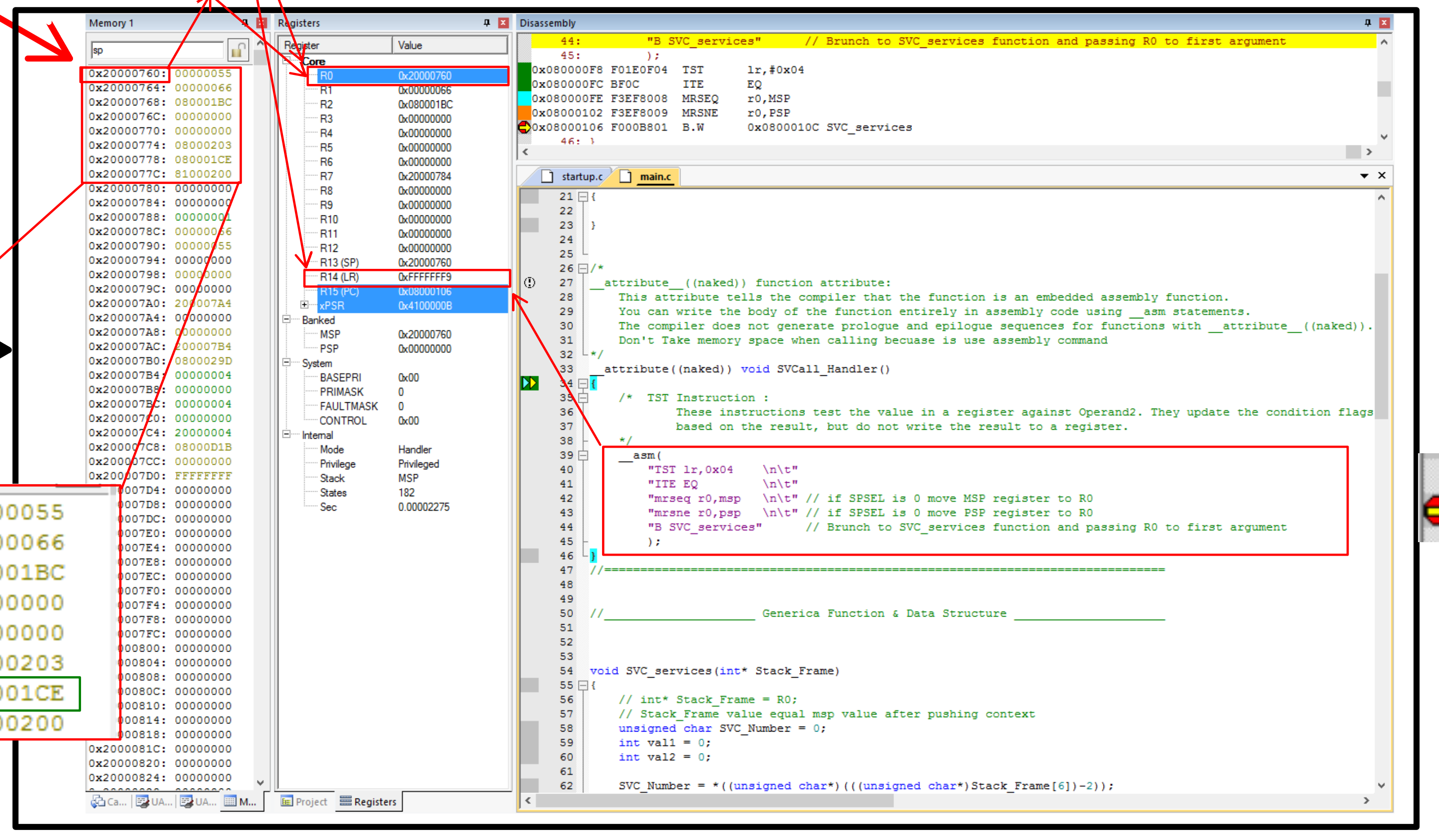


```
0x20000760: 00000055
0x20000764: 00000066
0x20000768: 080001BC
0x2000076C: 00000000
0x20000770: 00000000
0x20000774: 08000203
0x20000778: 080001CE
0x2000077C: 81000200
```

```
127:              __asm("SVC #0x01");
0x080001CC DF01        SVC      0x01
128:              break;
129:          case 2:
0x080001CE E006        B        0x080001DE
```

The address of the instruction after the SVC Call

Here, we check the used stack before executing the SVC_exception, and then assign the value of the stack to R0.



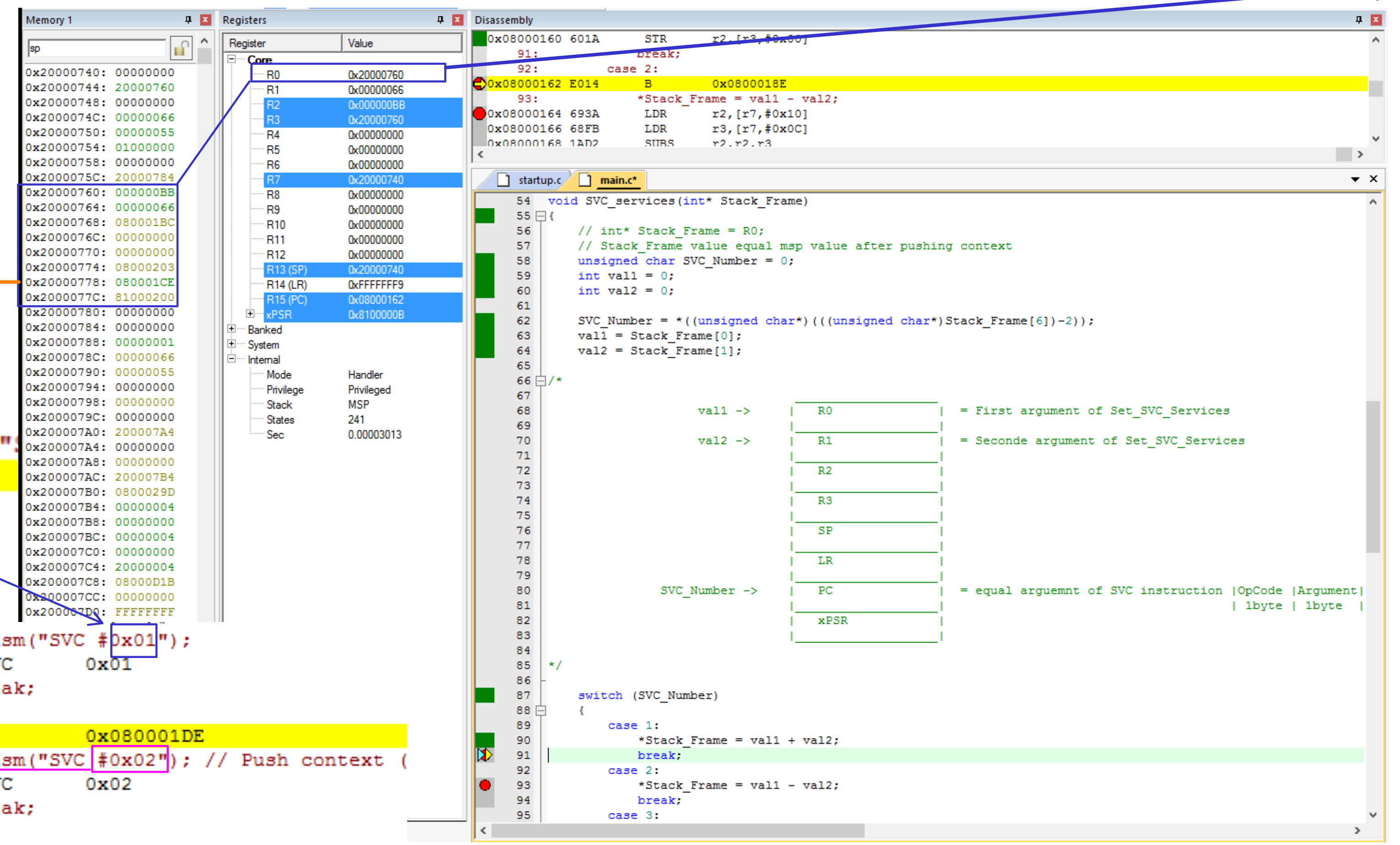
Note that the stack is not used during the invocation of the SVCall_Handler function due to the use of attribute((naked)).
Please read the clarification regarding the SVCall_Handler function.

0x080001CE - 2 =
0x080001CC

```
127:              __asm("S
0x080001CC  DF01       SVC
```

Argument

This value can be accessed through r0 or the Stack_Frame, which contains the location of the context pushing operation.



```
127:              __asm("SVC #0x01");
0x080001CC DF01        SVC      0x01
128:              break;
129:          case 2:
0x080001CE E006        B        0x080001DE
130:              __asm("SVC #0x02"); // Push context (
0x080001D0 DF02        SVC      0x02
131:              break;
132:          case 3:
0x080001D2 E004        B        0x080001DE
133:              __asm("SVC #0x03"); // Push context (
0x080001D4 DF03        SVC      0x03
134:              break;
135:          case 4:
0x080001D6 E002        B        0x080001DE
136:              __asm("SVC #0x04"); // Push context (
0x080001D8 DF04        SVC      0x04
137:              break;
```

Argument
Argument
Argument