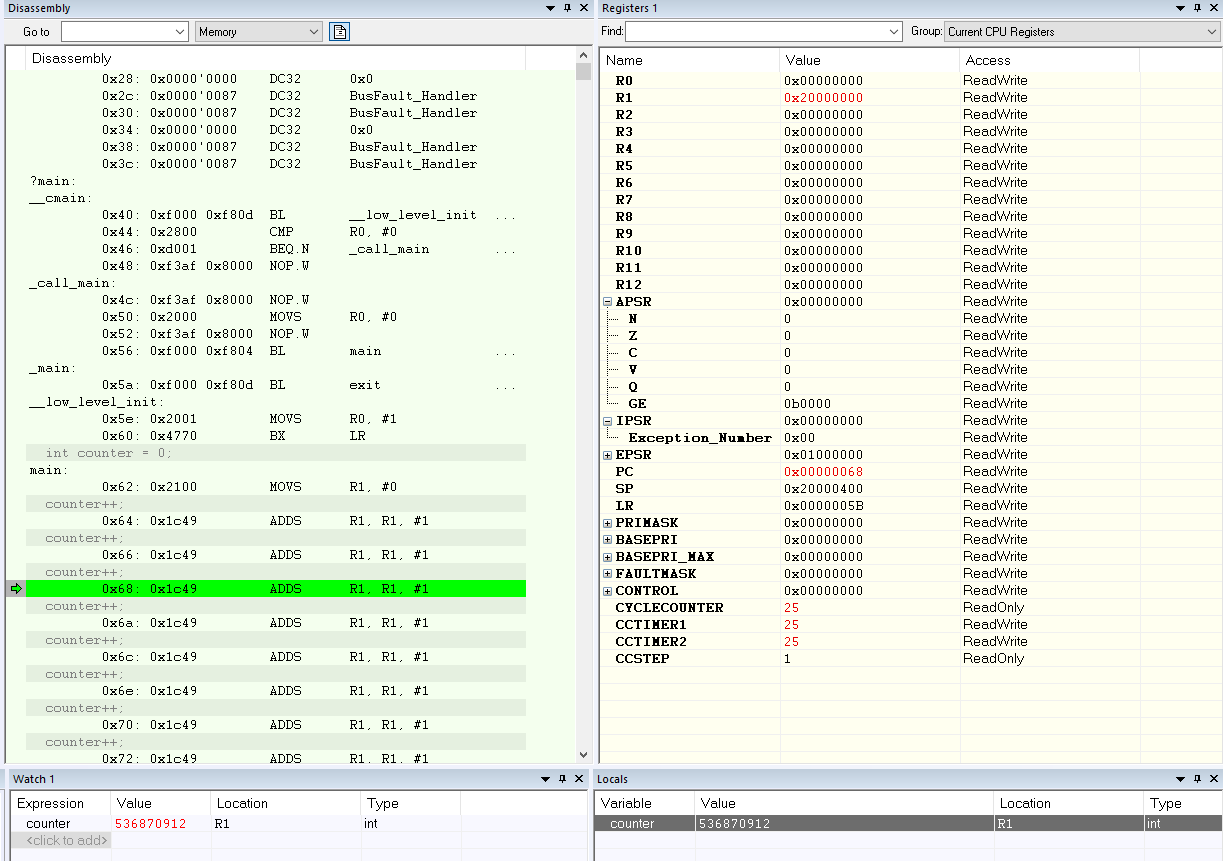
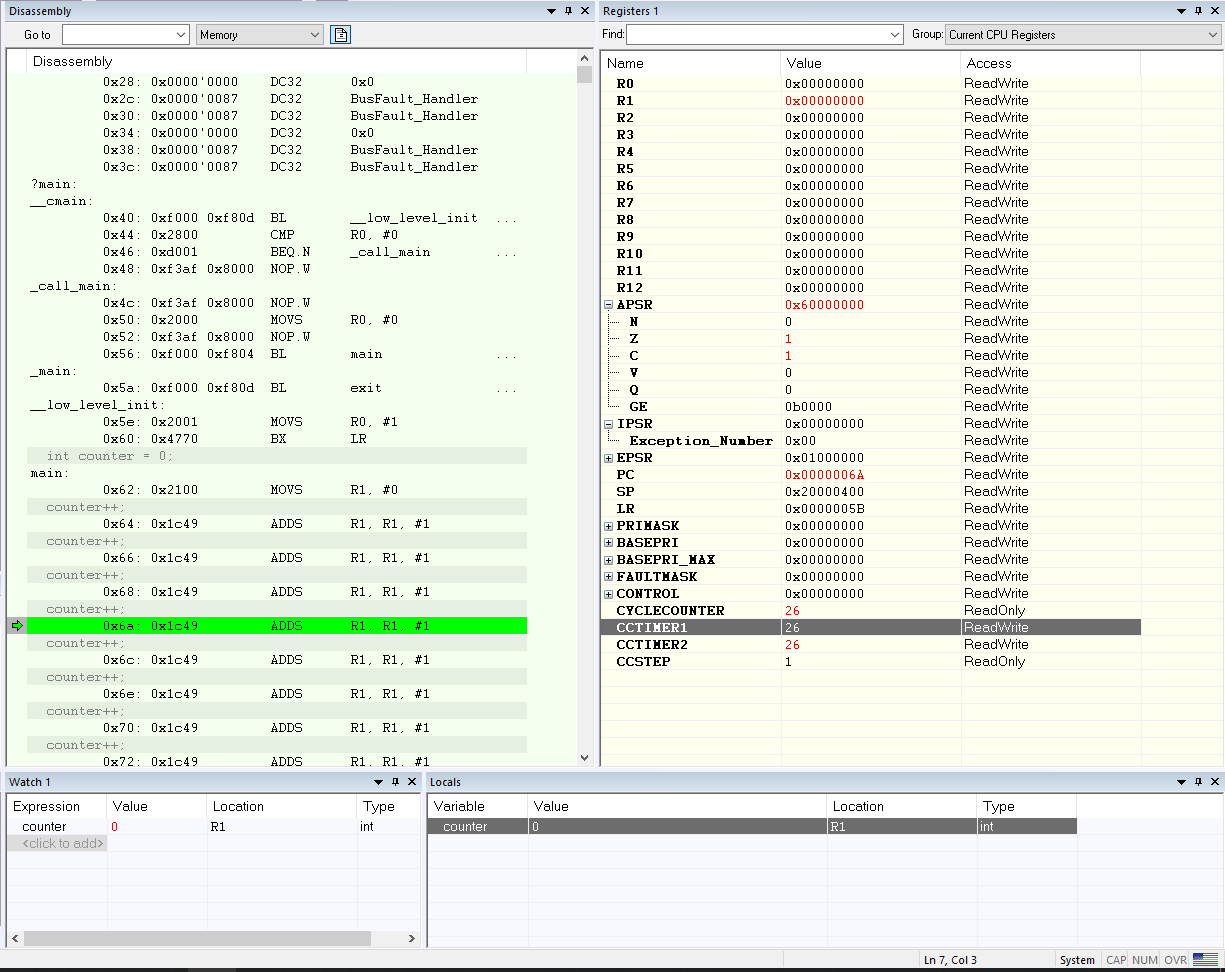
**Observe and answer:**

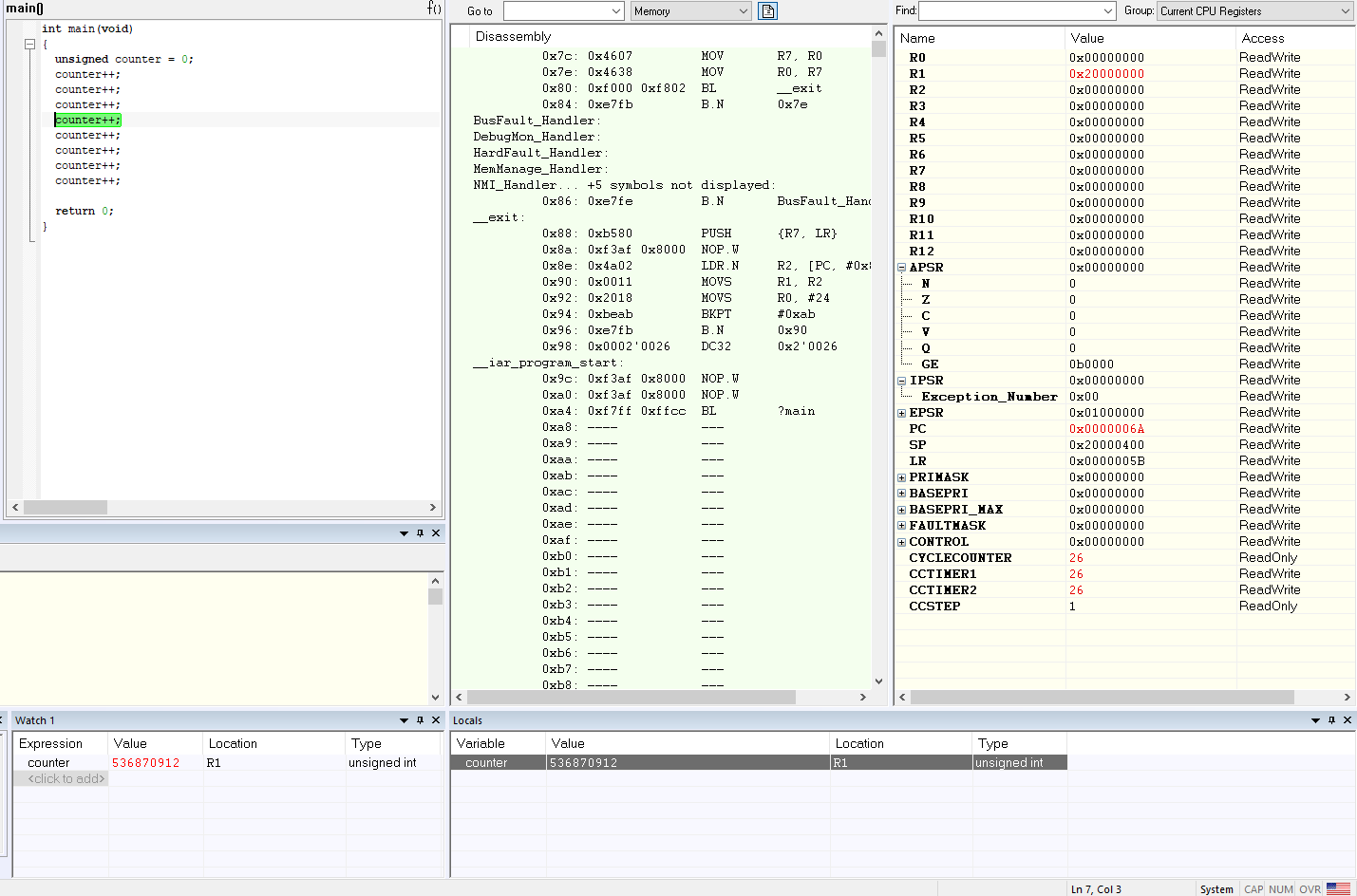
1. Inject 0x1FFFFFFF for the “counter” value in the variable window, then step thru the program only once to increment “counter”.
   1. What is the value of the “counter” from the “Locals” window?   
      *The value of the counter in “locals” window: 536870912 (Dec) and 0x20000000 (Hex)*
   2. What is the value of the “counter” in the “Register” window?  
      *The value of the counter in “Register” windows: R1 = 0x20000000 (Hex)*
   3. Which flags are set in the APSR register? Explain why?  
      *None of flags in APSR are set. The int type in C is singed by default, and with 32 bit system the maximum positive number is 0x7FFF FFFF. Therefore, increasing 1 from 0x1FFFFFFF equal 0x20000000 will not enable any flags in APSR.*



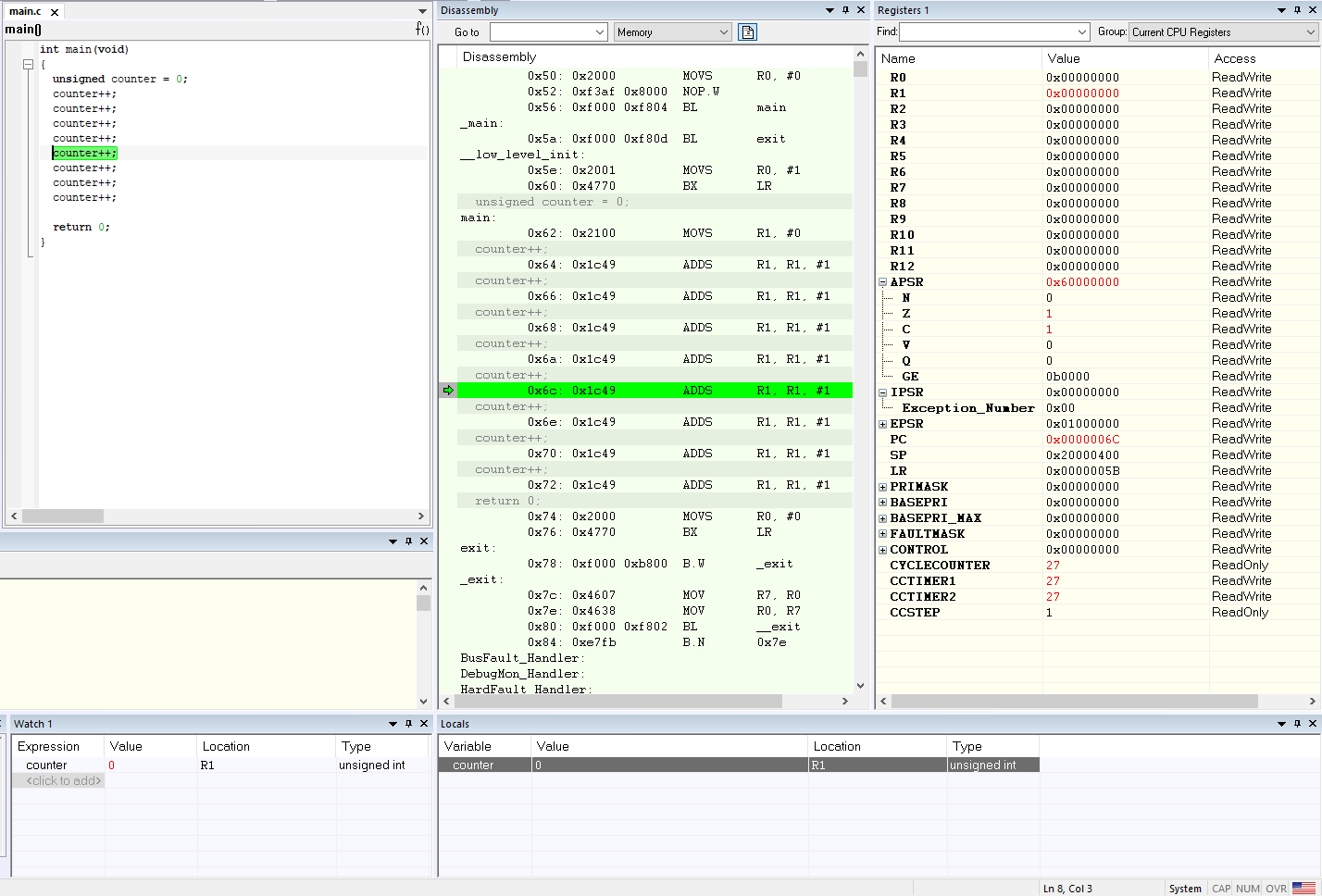
1. If your write all Fs (0XFFFFFFFF) in the Register value for “counter” then step thru the program once to increment “counter”
   1. What happens to the value of “counter” in the “Locals” window?   
      *The value of “counter” in the “locals” window is 0. And register R1 = 0 also.*
   2. What flags, if any, are set in the APSR?   
      *Both APSR.Z = 1 and APSR.C = 1. M4 controller uses 32 bit system, so the maximum hex value is 0XFFFFFFFF. If we increase 1 from there, the counter will back to 0 with 1 carry out. That why the APSR.Z and APSR.C are set in this test. (C bit set when result does not fit in “machine word”. Z bit set when result is 0).*



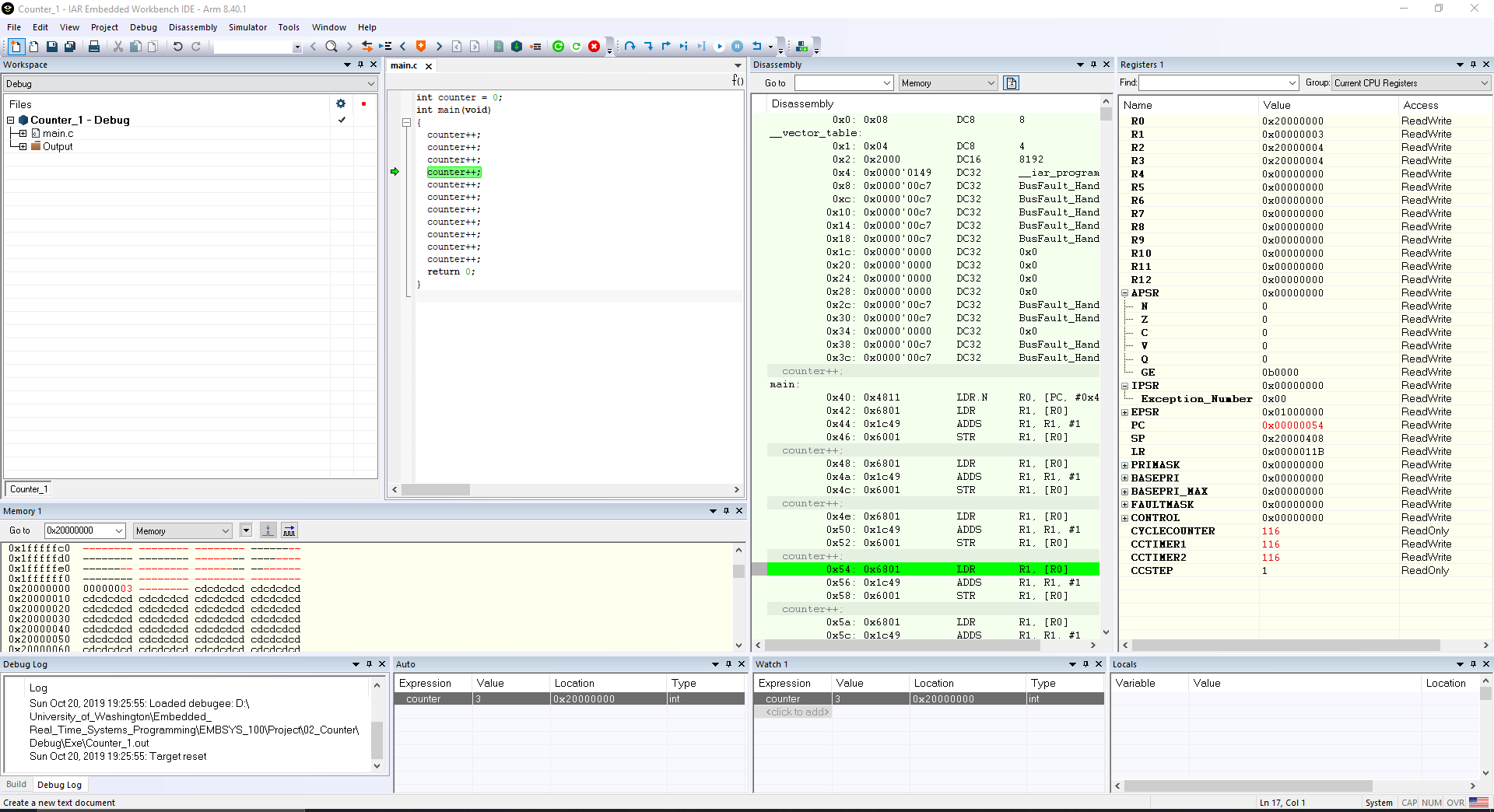
1. Change the “counter” variable type in your code to “unsigned”. Inject the values “**0x1FFFFFFF**” then step thru the program to increment the “counter” once:
   1. What is the value of “counter” in the “Locals” window after incrementing for each value?   
      *The value of the counter in “locals” window: 536870912 (Dec) and 0x20000000 (Hex).*
   2. What flags, if any, are set in the APSR? Explain why?  
      *None of flags in APSR are set. With 32 bit system, the unsinged type will have the range from 0 to 0xFFFFFFFF. Therefore, increasing 1 from 0x1FFFFFFF equal 0x20000000 will not enable any flags in APSR.*



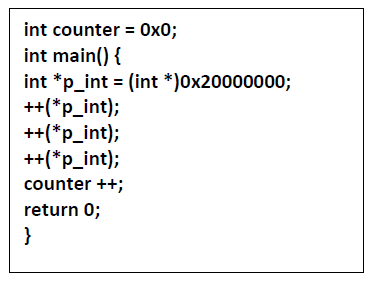
1. Change the “counter” variable type in your code to “unsigned”. Inject the values “**0xFFFFFFFF**” then step thru the program to increment the “counter” once:
   1. What is the value of “counter” in the “Locals” window after incrementing for each value?   
      *The value of “counter” in the “locals” window is 0. And register R1 = 0 also.*
   2. What flags, if any, are set in the APSR? Explain why?   
      *Both APSR.Z = 1 and APSR.C = 1. M4 controller uses 32 bit system, so the maximum hex value is 0XFFFFFFFF event int or unsigned data type. If we increase 1 from there, the counter will back to 0 with 1 carry out. That why the APSR.Z and APSR.C are set in this test. (C bit set when result does not fit in “machine word”. Z bit set when result is 0).*



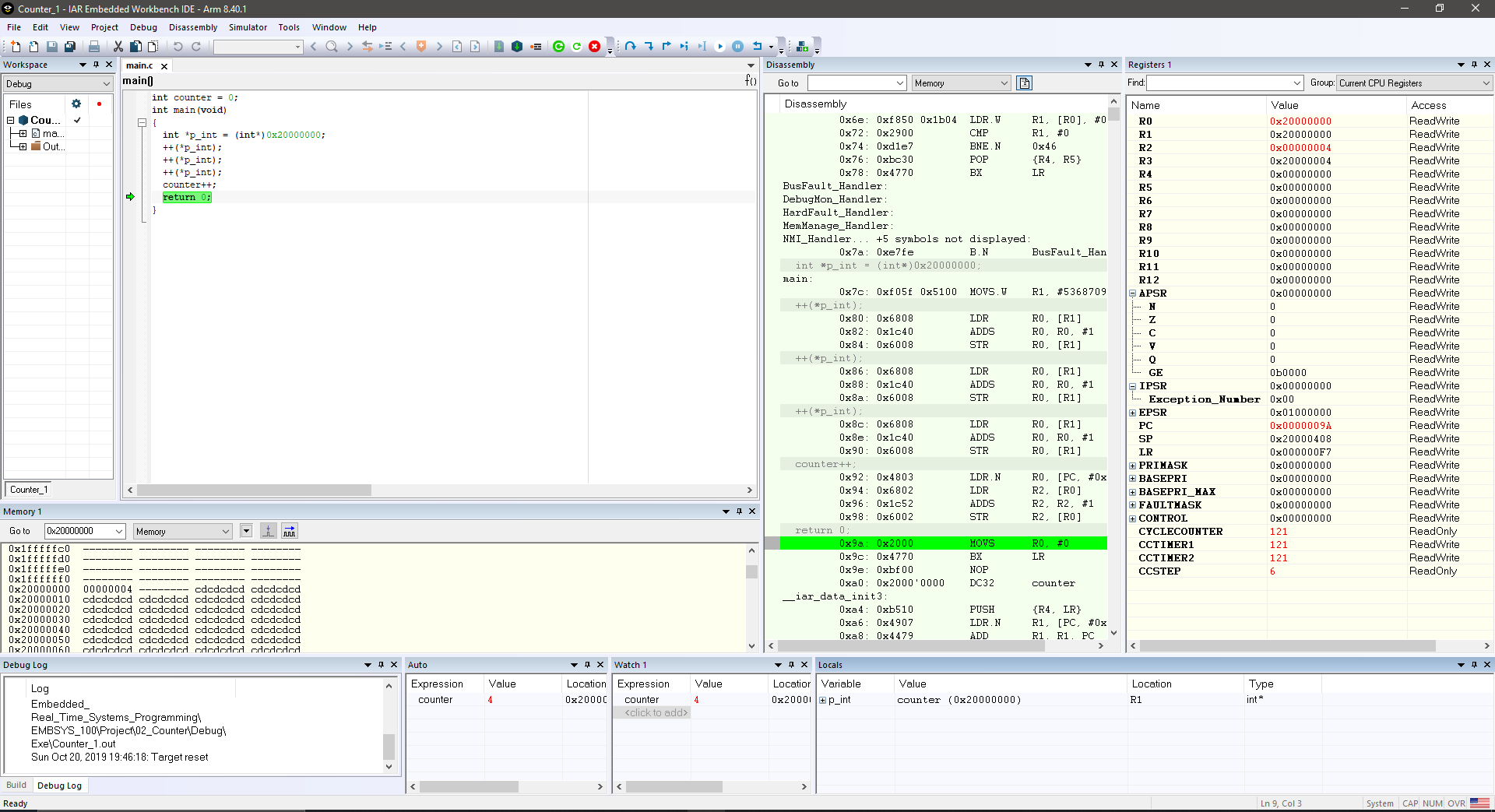
1. Move the “counter’ variable outside of main (at the top of the file):
   1. What is the scope of the variable “counter”?   
      The “counter” change from local variable to global variable.
   2. Is it still visible in the “Locals” view?   
      The “counter” – global variable will not show up in “Locals” view windows.
   3. In which window view can we track “counter” now?   
      To track “counter” now, we can use “watch” or “auto” windows.
   4. What is the address of the “counter” variable in memory?   
      After become global variable, the “counter” has memory at 0x20000000.



1. Change the source code to the following, then run the program still in the simulator:



* 1. What is the value of “counter” at the end of the program (halting at the return 0 statement).  
     *The value of “counter” at the end of the program has value 4 (DEC).*
  2. Explain why the counter value has changed?  
     *The global variable “counter” has value 0 and is stored at 0x2000000 memory location. Then we create a local pointer (\*p\_int) point to “counter address”. Next we increase the value which pointer point to 3 times, then increase the global variable “counter” one more time. Finally, we increase the counter 4 times from 0. Now the counter has value 4 at the end.*



1. Change the setting of IAR to run the same program on the evaluation board:
   1. What is the address where “counter” is stored?  
      *The global variable “counter” is stored at 0x20000000 memory address location.*
   2. Is the “counter” variable stored in RAM or ROM?  
      *The “counter” variable is stored in RAM not ROM.*
   3. What is the value of “counter” at the end of the program (halting at the return 0 statement)  
      *The “counter” at the end of the program has value 4 after 3 times increasing via pointer and 1 time directly increasing.*

