EMBSYS310-AU20ASSIGNMENT 02

1.Inject 0x7FFFFFFF for the “counter” value in the variable window, then step thru the program only once to increment “counter”.

a) What is the value of the “counter” from the “Locals” window? Answer -2147483648

b) What is the value of the “counter” in the “Registers” window? Answer 0x80000000

c) Please note down if the N and/or V flags are set in the APSR register. And explain why.

Answer: The N and V flags are set, because the variable counter is of type integer and signed by default (unless it is declared to be unsigned) where the MSB is for sign of the number, i.e. to indicate whether the number is positive or negative. The remaining number indicates the magnitude. The given number, 0x7FFFFFFF, is the last positive number for signed integer and stepping to add 1 would change MSB from 0 to 1 (0x7FFFFFFF to 0x80000000) and this set N to show the number has negative and sets V to 1 to show there is overflow due range for signed int type data.

2.If your write all Fs (0XFFFFFFFF) in the Register value for “counter” then step thru the program once to increment “counter”

a) What happens to the value of “counter” in the “Locals” window? Answer 0

b) Please note down if the N and/or V flags are set in the APSR register. And explain why.

Answer the N and V flags are both 0, because 0XFFFFFFFF plus 1 is 0, which is non negative and within the range of the integer type data.

3.Change the “counter” variable type in your code to “unsigned int”. Inject the values “0x7FFFFFFF” then step thru the program to increment the “counter” once:

a) What is the value of “counter” in the “Locals” window after incrementing for each value? Answer 2147483648

b) Please note down if the N and/or V flags are set in the APSR register. And explain why. Answer the N and V flags are both 0, because 2147483648 is non negative and within the range for the unsigned integer type data => . So, no stack over flow and no negative flag is set in this case.

4.Change the “counter” variable type in your code to “unsigned”. Inject the values “0xFFFFFFFF” then step thru the program to increment the “counter” once:

a) What is the value of “counter” in the “Locals” window after incrementing for each value? Answer 0

b) Please note down if the N and/or V flags are set in the APSR register. And explain why. Answer the N and V flags are both 0, because 0xFFFFFFFF plus 1 is 0x00000000 with carry over 1 for 32 register. So, 0 is non negative and within the range for unsigned int type data and so both N and V are 0.

5.Move the “counter’variable outside of main (at the top of the file):

a) What is the scope of the variable “counter”? Answer - The scope of the variable “counter” is Global

b) Is it still visible in the “Locals” view? Answer - No

c)In which window view can we track “counter” now? Answer – Watch window

d)What is the address of the “counter” variablein memory? Answer 0x20000000

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

int counter = 0x0;

int main() {

int \*p\_int = (int \*)0x20000000;

++(\*p\_int);

++(\*p\_int);

++(\*p\_int);

counter ++;

return 0;

}

a) What is the value of “counter” at the end of the program (halting at the return 0 statement) Answer 4

b) Explain why the counter value has changed? Answer: The counter is defined as a global variable and takes the location of 0x20000000. The pointer p\_int is defined to point to the same location, which means the pointer is pointing to the counter. When we are manipulating the pointer by adding 1 (++(\*p\_int)), we are actually adding 1 to the “counter” variable.

7.Change the setting of IARto run the same program on the evaluation board:

a) What is the address where “counter” is stored? Answer R1 = 0x20000000

b) Is the “counter” variable stored in RAM or ROM? Answer RAM

c) What is the value of “counter” at the end of the program (halting at the return 0 statement) Answer 4