Lab 5

Date: 04/11/2021 **lab duration: 10:00-13:00**

Plan of the day:

[1] To learn how to use pointers and arrays in a project.
[2] Complete the lab tasks by building an application use Arrays and Pointers.
[3] I plan to spend 15 mins to read the lab 5 guidelines, regarding the knowledge about arrays and pointers. And another 30 mins to complete the introductory task 1. Then with 1 hour to complete task 2 and testing. The final hour is for task 3 and its testing.

Chronological Lab record (24-hour format):

[10:00]:
Read the introductory sessions about arrays and pointers. [1]
[10:15]:
Started task 1
Opened the **es3_hw_lab_5.xpr** file in Vivado and exported hardware platform.
Launched SDK.
Read descriptions of arrays and pointers in page 1, 2 of the lab guide.
Created a new Pointer application and a **pointer_ex.c** file based on the HelloWorld template.
In pointer_ex.c, creatde a Char type pointer variable, created another variable and assigned an initial value of 0 to it in the main. Then assigned a value of 5 to it using its pointer (Please see Figure 1 in the appendix for the code in detail)
Programmed FPGA.
Set up a Debug configuration via inx C/C ++ application -> pointer_db, with Connect STDIO to Console selected to Port COM4 and a BAUD Rate of 9600.
Clicked Apply and Debug.
The task 1 code is tested by running the debug configuration and check the change of value of the variable b in the Expression view
In the Debug window, clicked on Window -> Show View -> Expression, added new expressions “b” and “address” to trace the chages of the two variables.
Added breakpoints at *address =5;
Debug the application using the above pointer_db configuration and step over the break points to see the changes of the variables b, address.
(See Figure 2 in the Appendix for the Test for task 1, including changes of the variables during the debug)

[10:40]: Started task 2
Created a new application project using the Hello World template named Swap_values.
Copied the code/header files in the “Swap_Values” folder inside “Lab5 - Codes folder”.
Followed the guidelines in the lab guide [1],
Created two variables and assigned initial values to them.

I encountered a problem here when trying to call the swap function. I inputed &addressA and &address as argument to the swap() function predefined. But this didn't work out. I tried *addressA and *address on this next, but they didn't work out neither. The reason I wanted to write them like this was because the swap() definition states that it takes two pointer parameters. I solved the problem by consulting one of the Tas, and I finally realise that both addressA and address are already pointers. And simply calling swap(addressA, address) would solve the problem.

```
a = 1;
b = 2;
addressB = &b;
addressA = &a;
swap(addressB, addressA);

while (1){

    displayNumber(b);
    XGpio_DiscreteWrite(&LED_OUT,1,b);
}
```

Another problem I had was I had no idea how to rewrite the swap() method without a third variable c. I had to google it for a solution. Luckily there was one on the internet. [2] So I learned their approach and managed to apply that on my swap() in the end.

```
void swap(u16 *a_ptr, u16 *b_ptr){

    *a_ptr = *a_ptr + *b_ptr;
    *b_ptr = *a_ptr - *b_ptr;
    *a_ptr = *a_ptr - *b_ptr;

}
```

To test my code, a similar method to task 1 testing was used. So I looked at the changes of the values of variables A and B in the Expression view in the debug mode and made sure they are swapped by the end of the program.

[12:00]:

Started task 3

Task 3 was easier for me in comparison to task2, most of my questions are answered through doing task2.

Here a simple while loop can calling the predefined sort algorithm would do the job.

Please see figure 3 for my code.

I tested the task 3 code by using a print() to print the array at the end of the program, and by checking that with the slideswitchIn inputs, I could see that the code indeed does what it's intended to do, that it's sorted based on the values within the array.

Summary:

All three tasks were successfully completed today. I spent 15mins on reading the introduction

of the lab guidance as planned, and task 1 was nice and easy, I managed to solve it in 30 mins as planned. However, I spent a bit longer than expected on task 2 because of the two problems I encountered, I solved them by consulting Tas and with Google. Task 3 was also easy to me because of the basis on task 2, only took me about 30 mins. I completed all three tasks and tested them using the expression view function in less than 3 hours of lab time.

Reference:

[1] Prof Tughrul Arslan. (2021)' Laboratory 5 – Arrays and Pointers Guideline'. *Engineering Software 3*. The University of Edinburgh.
[2] C Program to swap two numbers without third variable at https://www.javatpoint.com/c-program-to-swap-two-numbers-without-using-third-variable

Appendix:

```c
#include <stdio.h>
#include "platform.h"

char b;
char *address;

int main()
{
    init_platform();

    b = 0;
    address = &b;
    *address = 5;
    while(1==1){

    }

    cleanup_platform();
    return 0;
}
```

Figure 1 – Use pointers in an application

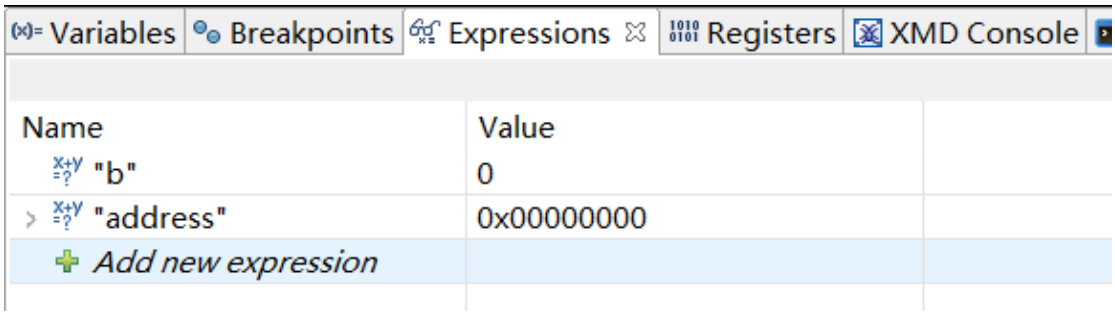

Figure 2a – the initial values of the variables b and address

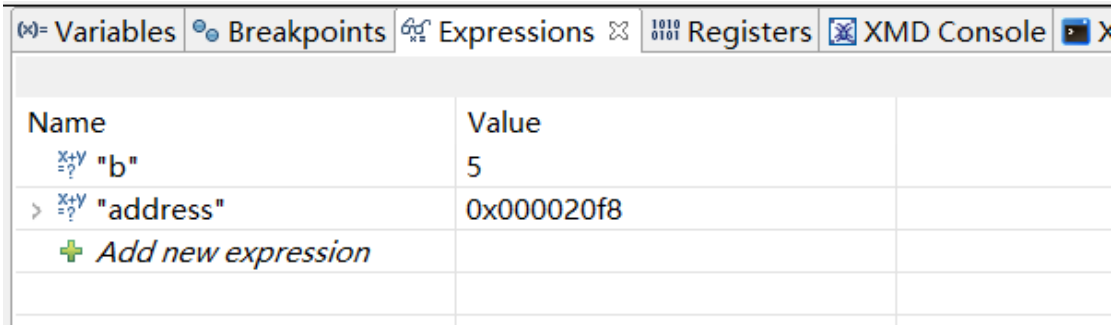

Figure 2b – check the final values of the variables b and address, to test my code

```
int j = 0;
while(j < sizeof(MAX))
{
    slideSwitchIn = XGpio_DiscreteRead(&SLIDE_SWITCHES,1);
    MAX[j] = slideSwitchIn;
    j++;
}
// Call sorting function here
sort(MAX);

int i=0;
while(1)
{

    displayNumber(MAX[i]);
    pushBtnLeftIn = XGpio_DiscreteRead(&P_BTN_LEFT, 1);
    // Check if button has been pressed
    if (pushBtnLeftIn == 1){
    // Wait for the button to be released
        while (pushBtnLeftIn == 1){
            pushBtnLeftIn = XGpio_DiscreteRead(&P_BTN_LEFT, 1);
        }
        if (i<MAX-1){
            i++;
        }else{
            i=0;
        }
    }
}
```

Figure 2b – my code for task 3 with comments