Rep_Lab_01_B0829002

Assembly Programming:

Introduction:

In this lab, we are going to learning how to write our first 8051 program and implement the double arrays multiplication and summation equation $S = \sum_{i=0}^{N-1} A[i] * B[i]$. The most crucial part of this lab is to know how to know the usage of each type of registers and how to through them to get or store value or address in these registers.

Code and explanation:

```
ORG 0x0000
MOV R0, #20h ;
MOV R1, #28h ;
MOV R4, #8 ;
MOV R5, #0 ; summary

Matrix_LOOP:
        MOV A, @R0;
        MOV B, @R1;
        MUL AB;
        ADD A, R5;
        MOV R5, A;
        INC R0;
        INC R1;
        DJNZ R4, Matrix_LOOP;

HERE:SJMP HERE
END
```

Firstly, we start with the address 0x000 and load 2 addresses from registers in this our program. Following this, the loop limitation was defined in line MOV R4, #8; with value eight, that is use to ensure the loop will run from A[0], B[0] to A[7], B[7] and then R5 is initialize with value zero.

Next, we wrote a loop to operate the multiplication and summation of these two arrays. Load the value from address array R0 and R1 to A and B. MUL AB to multiple A[i] and B[i] and then store at A. After that add the A with the whole summation of each loop and load it back to R5. INC R0 and INC R1 are like the ++ operation in C++ or C to move the pointer address to next address. Next, DJNZ R4, Matrix_LOOP; is use to determination the R4 is zero or not, if yes break out this loop and stay at HERE:SJMP HERE, which is used to stop the program at here.

Difficulties I've encountered and solution:

In this lab, I got some problem,

Problem 1st: I cannot use the @ to get the value of each register, because not all the register can use the immediate address method to get values in every register. From datasheet "@Ri - Data RAM location addressed indirectly through R0 or R1.", we can only use the R0, R1 to get the value with @.

Solution 1st: Use R0, R1 instead of other registers to load the arrays address. Following this, we can use @ to get the values of A[i] and B[i].

Problem 2nd: How to determinate the end of loop?

Solution 2nd: We can use DJNZ to minus one of each loop. After the variable get to zero, we can break out this loop.

What I learned in this lab?

In this lab, I learned how to basically use the start a project and open the file in Silicon IDE as well as the variables and addresses to get the data of registers. Furthermore, I implemented 2 loops, one to operate the multiplication and summation and the other one is used to keep the process in this program.