EMBEDDED PROGRAMMING LAB

LAB-1 DATE:7-08-2024

PREETHISH K R

STEPS TO EXECUTE PROGRAM IN KEIL UVISION:

- Open Keil uvision
- Click on new project then new uvision project save the file with name
- Select LPC1768 as the target device
- Double click on startup_LPC17xx.s and make the following changes:
  line 121 IMPORT __main
  Comment the following:
  line 127 ;IMPORT SystemInit
  line 129 ;LDR R0, =SystemInit
  line 130 ;BLX R0
  line 272 ;IMPORT __use_two_region_memory
  line 273 ;EXPORT __user_initial_stackheap
- Right click on source group 1 select add new item to source group 1 further select ASM File and write name of the file it will be suffixed by .s after saving on its own.
- Type the program
- The code must always start and end with following instructions

```
        AREA BLOCK, CODE,READONLY
        ENTRY
        EXPORT __main
__main

        (code)

        NOP
        END
```

- After writing the code save it and click on Build check for errors
- After having no errors go to Debug and select start Debug session

- Open memory window then right click on it and select signed further select int
- Reduce the size of memory window till required column of memory location is obtained and type the Address to start
- Enter the data values to the respective memory locations
- Click on step over to perform the execution of each line of the code and hence shows the result
- Also, the respective values loaded to the registers and flags are also displayed on the left corner
- Stop the Debug session

1) Write a program to perform addition of two numbers available at two consecutive memory locations and store the result to the next location

I. Initialize one register as pointer which points to the address of 1st data
II. Load 1st number to another register R2
III. Load the 2nd number to the register R3
IV. Perform addition of number available at R2 and R3 further store the result to next location

Program:

```
        AREA BLOCK, CODE,READONLY
        ENTRY
        EXPORT __main
__main
        LDR R1,=0X10000000
        LDR R2,[R1]
        LDR R3,[R1,#4]
        ADD R4,R2,R3
        STR R4,[R1,#8]
        NOP
        END
```

Output:

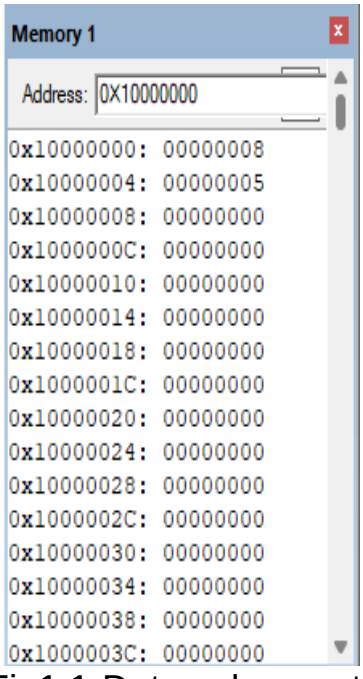

Fig1.1-Data values entered

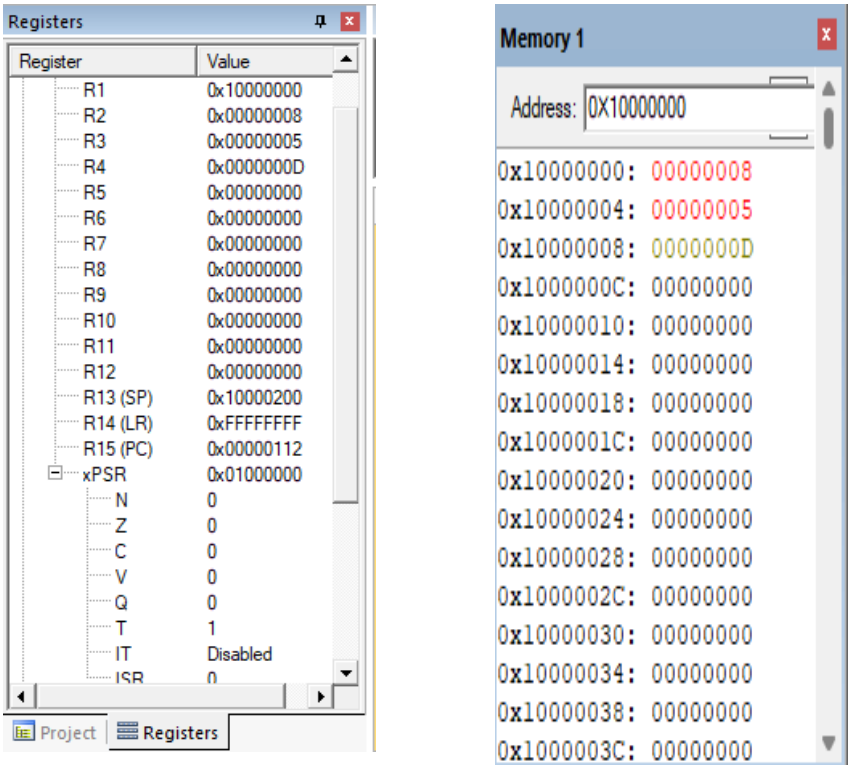

Fig1.2-Result obtained

2)Write a program to perform subtraction of two numbers available at two consecutive memory locations and store the result to the next location

I. Initialize one register as pointer which points to the address of 1st data
II. Load 1st number to another register R2
III. Load the 2nd number to the register R3
IV. Perform subtraction of number available at R2 and R3 further store the result to next location

Program:

```
        AREA BLOCK, CODE, READONLY
        ENTRY
        EXPORT __main
__main
        LDR R1,=0X10000000
        LDR R2,[R1]
        LDR R3,[R1,#4]
        SUB R4,R2,R3
        STR R4,[R1,#8]
        NOP
        END
```

Output:

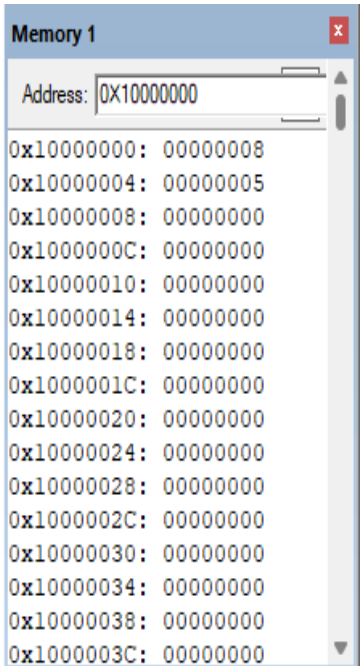

Fig2.1-Data values entered

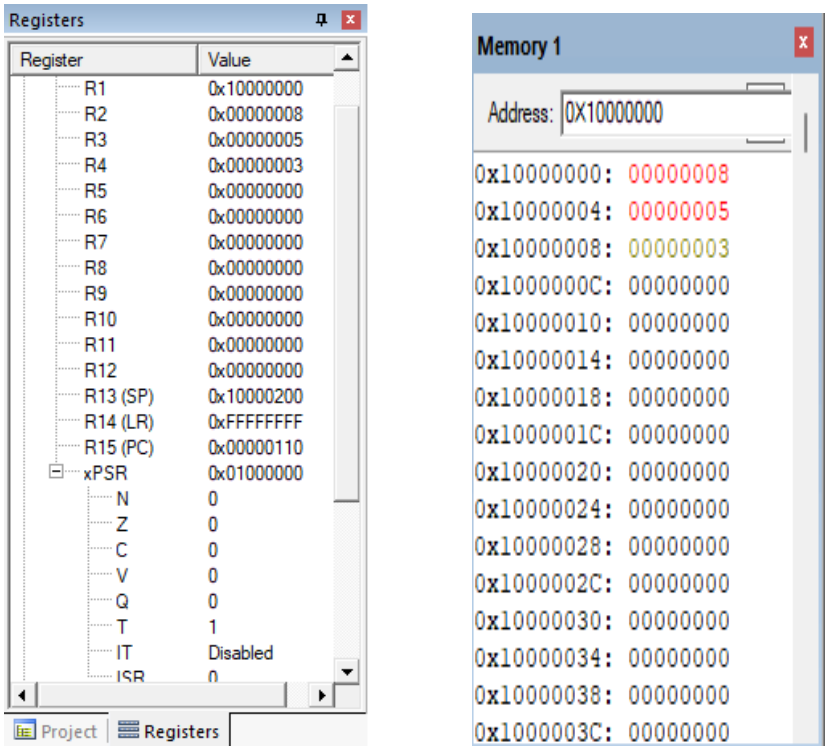

Fig2.2-Result obtained

3)Write a program to perform multiplication of two numbers available at two consecutive memory locations and store the result to the next location

I. Initialize one register as pointer which points to the address of 1st data
II. Load 1st number to another register R2
III. Load the 2nd number to the register R3
IV. Perform multiplication of number available at R2 and R3 further store the result to next location

Program:

```
        AREA BLOCK, CODE, READONLY
        ENTRY
        EXPORT __main
__main
        LDR R1,=0X10000000
        LDR R2,[R1]
        LDR R3,[R1,#4]
        MUL R4,R2,R3
        STR R4,[R1,#8]
        NOP
        END
```

Output:

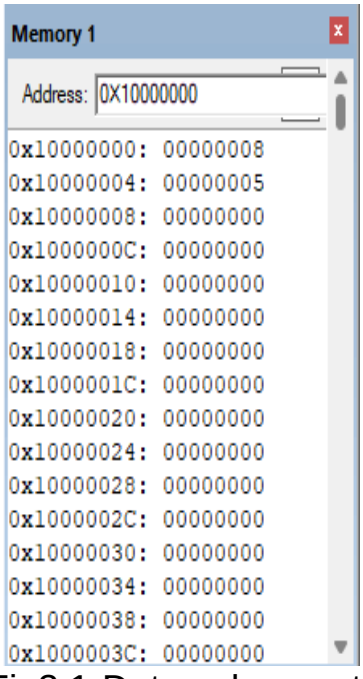

Fig3.1-Data values entered

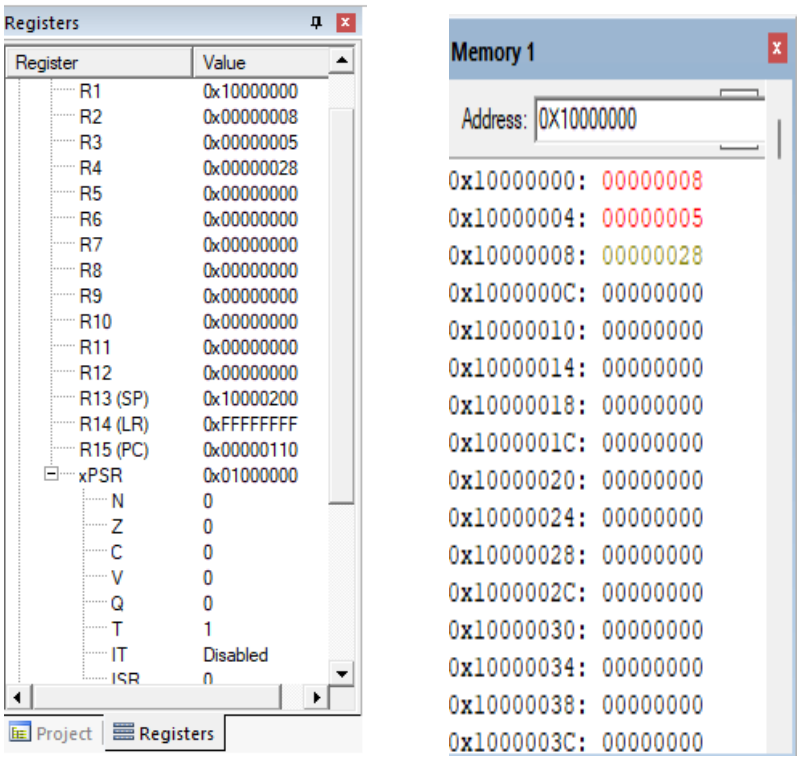

Fig3.2-Result obtained

4)Write a program to perform Division of two numbers available at two consecutive memory locations and store the result to the next location

I. Initialize one register as pointer which points to the address of 1st data
II. Load 1st number to another register R2
III. Load the 2nd number to the register R3
IV. Perform Division of number available at R2 and R3 further store the result to next location

Program:

```
        AREA BLOCK, CODE, READONLY
        ENTRY
        EXPORT __main
__main
        LDR R1,=0X10000000
        LDR R2,[R1]
        LDR R3,[R1,#4]
        UDIV R4,R2,R3
        STR R4,[R1,#8]
        NOP
        END
```

Output:

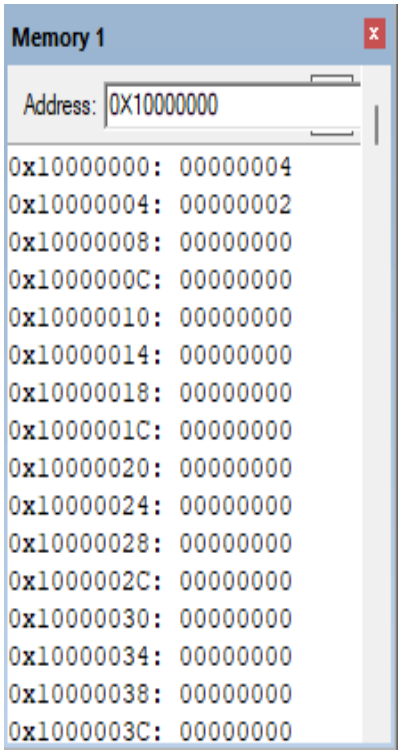

Fig4.1-Data values entered

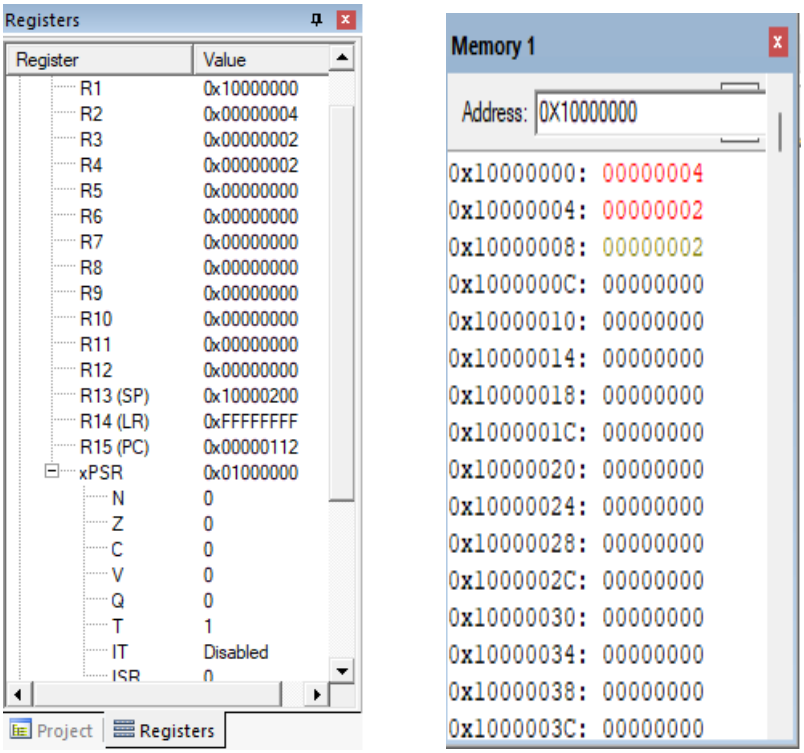

Fig4.2-Result obtained