

# 19CSE303 – EMBEDDED SYSTEMS

Lab Eval 2 – 13.09.2021

## Practice Lab (Arrays)

---

- R.Abhinav  
- CB.EN.U4CSE19453

1. Write an ALP using ISA of ARM7 TDMI to add two 64 bit numbers and show the result in R6 ( LSB 32 bits) and R7 ( MSB 32 bits).

**Write an ALP using ISA of ARM7 TDMI to do the following**

- A in R0, B in R1, X in R2, Y in R3 ( A, B, X, Y are unsigned 32 bit numbers)
- Find the difference of A and B store in R4 (  $R4 = A - B$  )
- Find the difference of Y and X store in R5 (  $R5 = Y - X$  )
- R6 should contain the lowest of (A-B) and (X-Y)
- R7 should contain the highest of ( A-B) and (X-Y)

**Implement the following conditions:**

- If  $(A-B) = (X-Y) \Rightarrow R8 = R0 \text{ (XOR) } R1$
- If  $(A-B) > (X-Y) \Rightarrow R8 = R0 \text{ (OR) } R1$

**Code :**

```
                AREA  add64, CODE, READONLY

ENTRY

MAIN

    LDR  R0, =A
    LDR  R1, [R0]
    LDR  R2, [R0, #4]
    LDR  R0, =Bs
    LDR  R3, [R0]
    LDR  R4, [R0, #4]
```

SUB R6, R2, R4

SBC R5, R1, R3

LDR R0, =Result

STR R5, [R0]

STR R6, [R0, #4]

SWI &11

A DCD &122AE640, &2F100123

Bs DCD &001019BF, &40023F51

Result DCD 0

END

```

1      AREA      add64, CODE, READONLY
2 ENTRY
3 MAIN
4      LDR        R0, =Value1
5      LDR        R1, [R0]
6      LDR        R2, [R0, #4]
7      LDR        R0, =Value2
8      LDR        R3, [R0]
9      LDR        R4, [R0, #4]
10
11     SUB        R6, R2, R4
12     SBC        R5, R1, R3
13
14     LDR        R0, =Result
15
16     STR        R5, [R0]
17     STR        R6, [R0, #4]
18
19     SWI        &11
20
21 Value1      DCD      &12A2E640, &F2100123
22 Value2      DCD      &001019BF, &40023F51
23 Result      DCD      0
24
25     END
26
27

```

assembling ADD\_64BIT.asm...

linking...

Program Size: Code=32 RO-data=0 RW-data=0 ZI-data=0

".\alp.axf" - 0 Error(s), 0 Warning(s).

The screenshot displays the Keil uVision IDE interface during the assembly and linking process. The **Registers** window on the left shows the current state of the processor registers, with R0 through R15 and CPSR all set to 0x00000000. The **Disassembly** window in the center shows the assembly code for the 'add64' section, which includes instructions for loading values, performing subtraction and subtraction with carry, and storing the results. The **Command** window at the bottom shows the linker output, indicating that the program size is 32 bytes and there are no errors or warnings.