EE2016 Experiment-7

Group-3 EE23B027, EE23B033, EE23b039

Task 1

For the given code note the register R0 values.

```
    AREA Program, CODE, READONLY
    ENTRY
        MOV r0,#11
Stop
        B Stop
END
```

Debugging and values stored in each register

Here the R0 will be immediately get loaded with the value 11 and remains the same. The program continues in a loop.

Interpretation

Setting a breakpoint at the B Stop instruction allows you to pause the infinite loop and examine the program's state, making it a useful tool for debugging.

Task 2

Replace 11 in the code in Task 1 by FFFFFFFF

```
    AREA Program, CODE, READONLY
    ENTRY
        MOV r0,#&FFFFFFFF
Stop
        B Stop
END
```

Debugging and value stored in the register

Here the R0 will be immediately get loaded with the hexadecimal number FFFFFFFF and remains the same. The program continues in a loop.

Task 3

Record the values of r1 by single stepping through the code.

```
    AREA Reset, CODE, READONLY
ENTRY

        LDR r0, =7
        MUL r1, r0, r0
        LDR r2, =4
        MUL r1, r2, r1

        LDR r3, =3
        MUL r3, r0, r3
        ADD r1, r1, r3

stop
        B stop

END
```

Debugging and value stored in register

Initially r0 is loaded with number 7. It is then multiplied with itself and is stored in r1 ie r1 = 49. Then it loads r2 with number 4 ans multiplies 4 with number in r1 and stores it in r1, ie r1 =196. It then loads r3 with 3 and then it is multiplied with value in r0 and stored in r3 ie r3 = 21. Finally r1 is added with r3 and is stored in r1 ie r1 = 217.

Task 4

ARM code to obtain 10th number in a fibonacci series.

```
    AREA Program, CODE, READONLY
ENTRY
    MOV R0, #0 ; Initialize the first Fibonacci number (F0 = 0)
    MOV R1, #1 ; Initialize the second Fibonacci number (F1 = 1)
    MOV R2, #2 ; Set the iteration counter to 2 (since F0 and F1 are already known)
    MOV R3, #10 ; We want the 10th Fibonacci number

Loop
    CMP R2, R3 ; Compare the counter with 10
    BEQ End ; If we have calculated the 10th number, exit
    ADD R4, R0, R1 ; Calculate the next Fibonacci number (F(n) = F(n-1) + F(n-2))
    MOV R0, R1 ; Move R1 to R0 (F(n-1) becomes F(n-2) for the next iteration)
    MOV R1, R4 ; Move the newly calculated F(n) to R1 (F(n) becomes F(n-1))
    ADD R2, R2, #1 ; Increment the counter
    B Loop ; Repeat the loop

End
    MOV R5, R1 ; Store the 10th Fibonacci number in R5
    SWI &11 ; Exit
```

Debugging

The 10th Fibonacci term is 34 which is stored in the register R5.

Task 5

ARM assembly language program to divide a 32-bit binary number by a 16-bit binary number and store the quotient as well as the remainder.

```
    AREA Program, CODE, READONLY
ENTRY
       LDR R0, =Num1
       LDR R0, [R0]
       LDR R1, =Num2
       LDRH R1, [R1]
       MOV R2, #0
       MOV R4, R0

Loop
       CMP R1, #0
       BEQ Error1
       CMP R4, R1
       BLT Result
       SUB R4, R4, R1
       ADD R2, R2, #1
       B Loop

Error1
    MOV R3, #0xFFFFFFFF


Result
    LDR R5, =Remainder
    STR R4, [R5]
    LDR R6, =Quotient
        STR R2, [R6]
       B STOP
       SWI &11
```

```
STOP
    B STOP

Num1 DCD 0x00000064
Num2 DCW 0x000A

    ALIGN

    AREA Data2, DATA, Readwrite
Quotient DCD 0
Remainder DCD 0

END
```

Debugging

Here the 32-bit number given is 100 and the 16-bit number given is 10. Therefore the number stored in R6 = 10 and R5 = 0 Similarly for 99 and 10 R6 = 9 and R5 = 9.