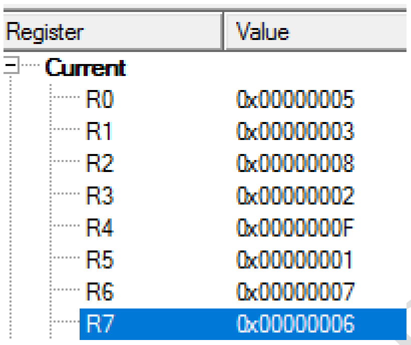
6. To write and simulate *ARM assembly language* programs for *data transfer, arithmetic and logical operations* (Demonstrate with the help of a suitable program).

```
      AREA PRG6,CODE,READONLY    ; defining logical area named prg6 and the
                                    code which is readonly

ENTRY                            ; the entry point where the code starts

      LDR R0,=5                 ; data transfer - R0=5
      LDR R1,=3                 ;                 R1=3

      ADD R2,R0,R1              ; arithmetic   ADD   R2=8 (5+3)
      SUB R3,R0,R1              ;              SUB   R3=2 (5-3)
      MUL R4,R0,R1              ;              MUL   R4=F (5*3 = 15 = F)

      AND R5,R0,R1              ; logical      AND   R5=1 (5&&3)
      ORR R6,R0,R1              ;              OR    R6=7 (5||3)
      EOR R7,R0,R1              ;              XOR   R7=6 (5^3)

      END                        ; end of the program
```

****

OUTPUT:

Press F7, then press Ctrl + F5 (start debug session) and keep pressing F11. You'll notice the following.



```
Register      Value
⊟ Current
   R0         0x00000005
   R1         0x00000003
   R2         0x00000008
   R3         0x00000002
   R4         0x0000000F
   R5         0x00000001
   R6         0x00000007
   R7         0x00000006
```

NOTE:

1. There should be a space before AREA
2. There should be a space before LDR,ADD,SUB etc instructions
3. There should NOT be a space before ENTRY.
4. Please DON'T have startup.s for this program. We require startup.s only for C programs not for ASM files.

TO HELP YOU UNDERSTAND ABOVE 1,2,3 POINTS:

Guys, ARM Assembly Program has some structure which you need to adhere. Let me explain you like this.

Consider the following table.

Column 1	Column 2	Column 3
You should write only ENTRY and loop variables	You should write only AREA, Instructions and END directives.	Comments