PES UNIVERSITY

Department of Computer Science & Engineering
Microprocessor & Computer Architecture Lab

Lab_2 Programs

UE23CS251B

Name of Student: Pranav Hemanth	SRN: PES1UG23CS433

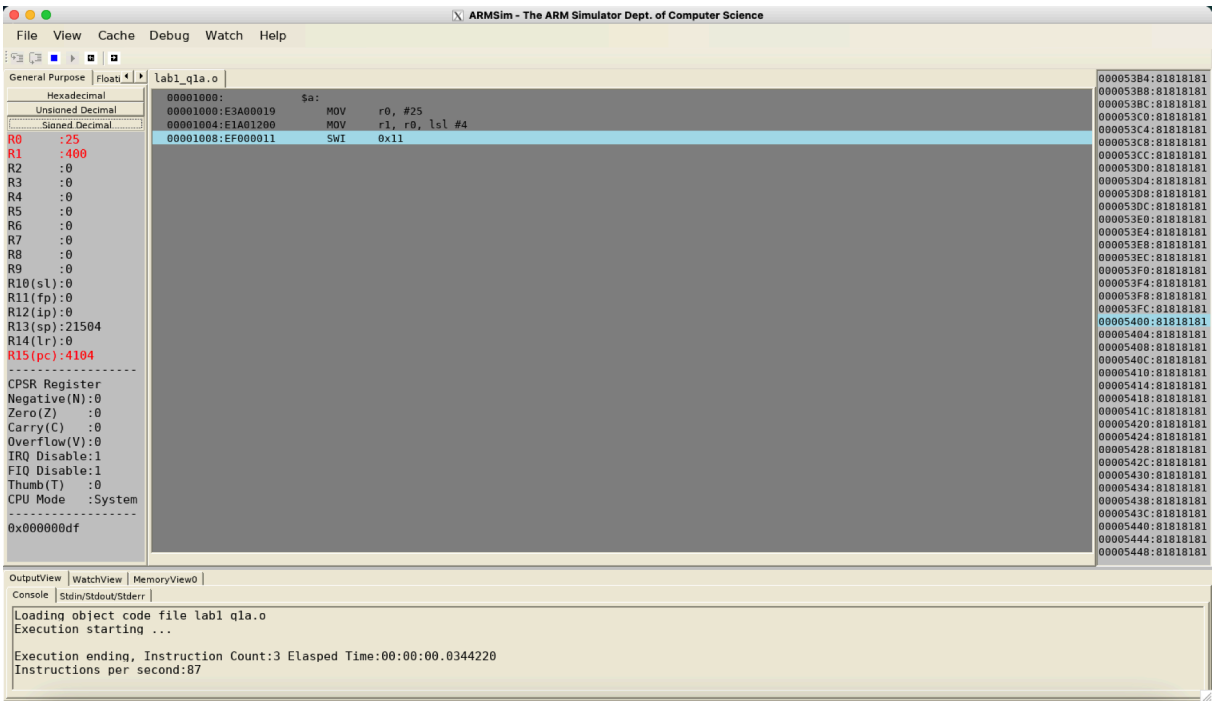
1 Write an ALP using ARM7TDMI to perform the multiplication of 16X25without using MUL instructions.

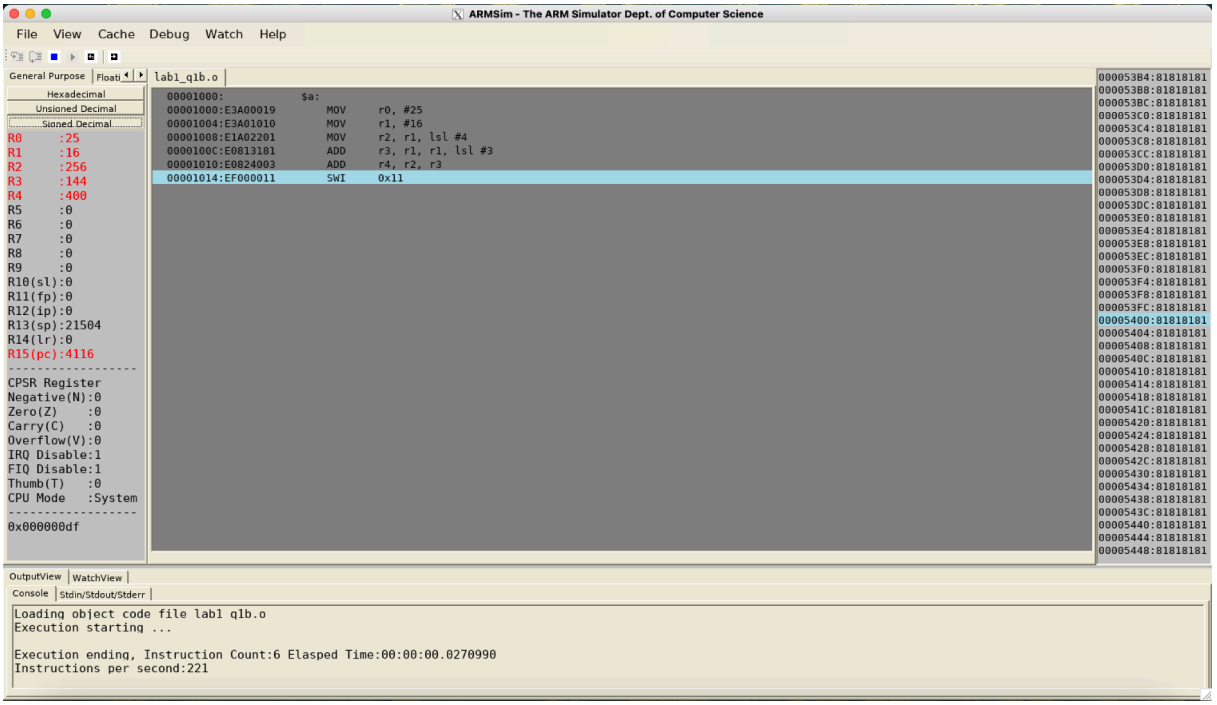
(Hint: barrel shifter instructions.)

(Note : Any number can be considered as multiplier)

ARMSim_files > ASM lab1_q1a.s

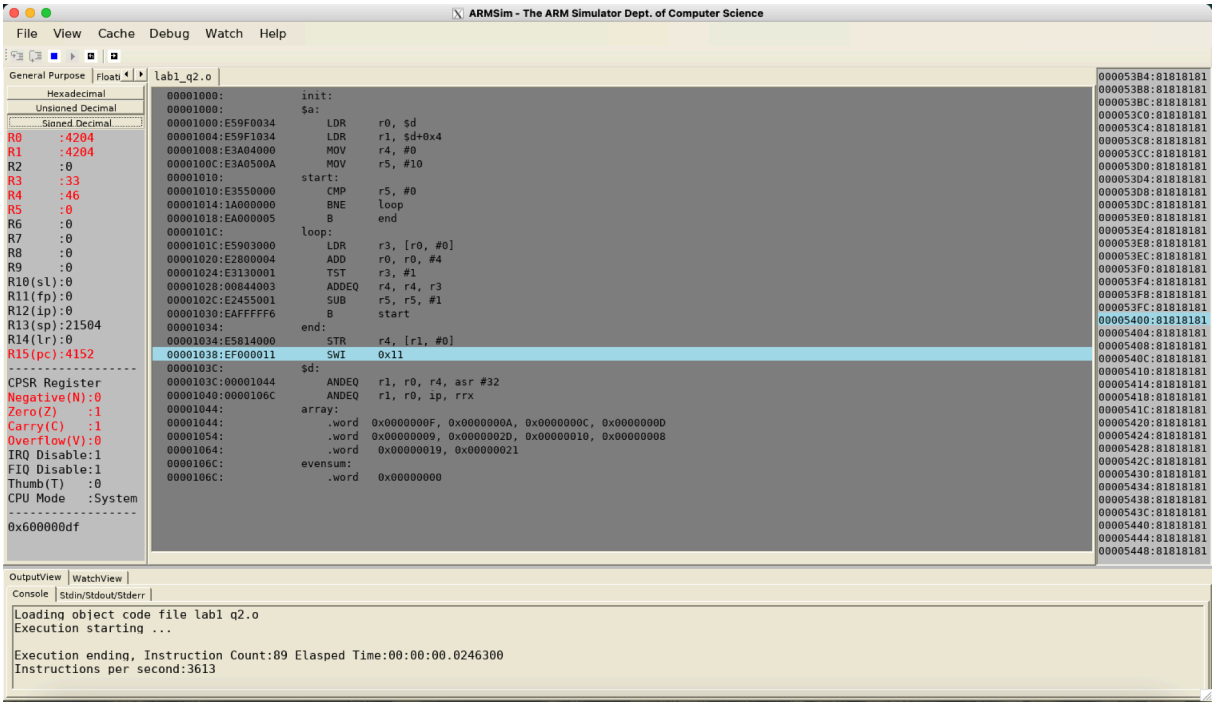
```
1   @ Write an ALP using ARM7TDMI to perform multiplication of 16X25 without using mul instruction
2   @ Hint barrel shifter instructions
3   @ Note: any number
4
5   .text
6   MOV R0, #25
7   MOV R1, R0, LSL#4
8   SWI 0x011
9
```

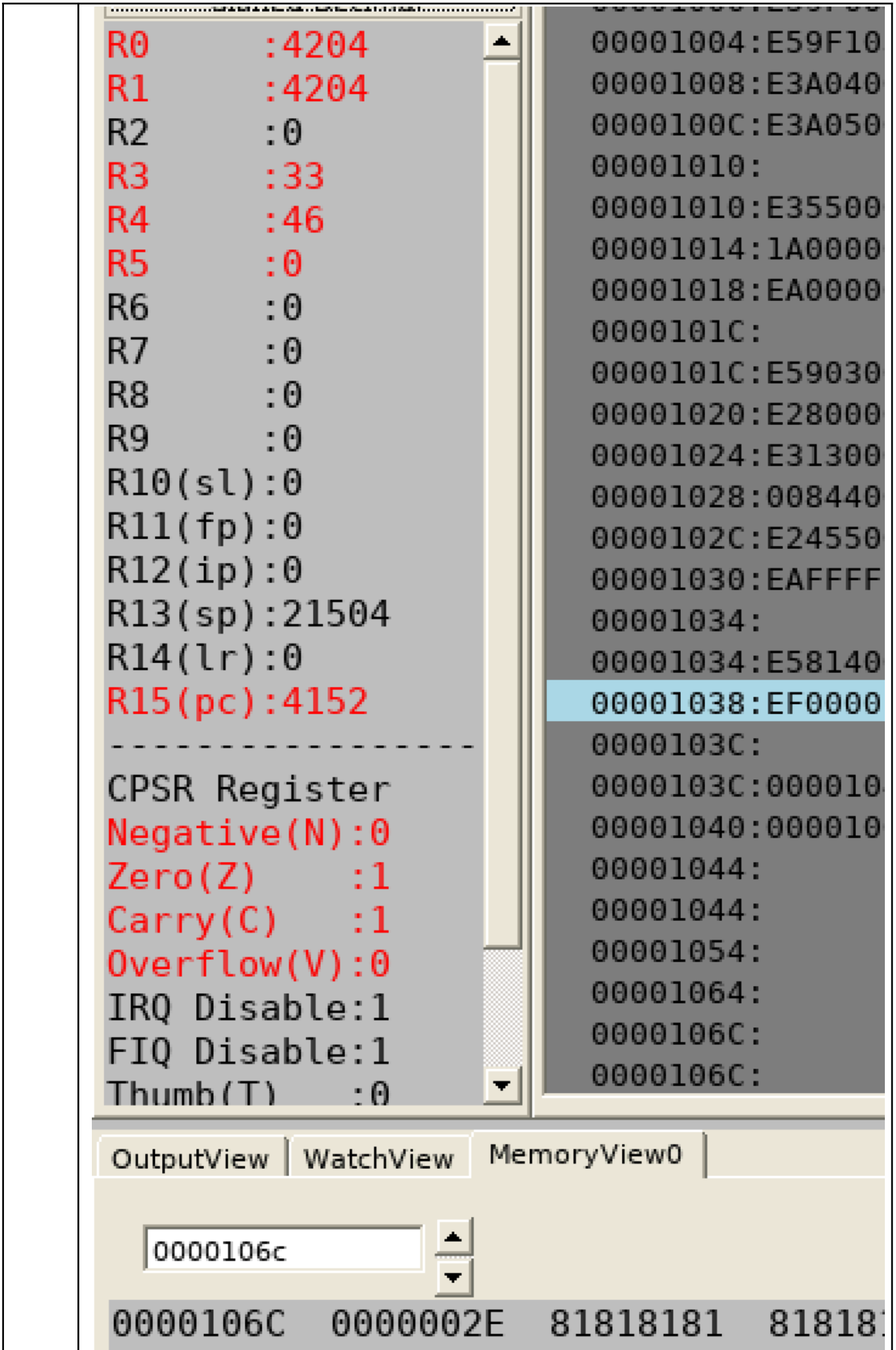


	ARMSim_files > ASM lab1_q1b.s 1 @ Write an ALP using ARM7TDMI to perform multiplication of 16X25 without using mul instruction 2 @ Hint barrel shifter instructions 3 @ Store n in R0 and result in R1 4 @ 16 is considered as multiplier 5 6 .text 7 MOV R0, #25 8 MOV R1, #16 9 MOV R2, R1, LSL#4 10 ADD R3, R1, R1, LSL#3 11 ADD R4, R2, R3 12 SWI 0x011 13 
2	Write an ALP using ARM7TDMI to add only even numbers stored in memory location for a given set of numbers and store the sum in the memory location. Array:. WORD 15,10,12,13,9,45,16,8,25,33 evensum:. WORD

ARMSim_files > ASM lab1_q2.s

```
1  @ Write an ALP to add only even numbers stored in memory location for a given set of numbers and store sum in memory location
2  @ Array: .WORD 15, 10, 12, 13, 9, 45, 16, 8, 25, 33
3  @ evensum: .WORD
4
5  .data
6  array: .word 15, 10, 12, 13, 9, 45, 16, 8, 25, 33
7  evensum: .word 0
8
9  .text
10
11 init:
12     LDR R0, =array
13     LDR R1, =evensum
14     MOV R4, #0
15     MOV R5, #10
16
17 start:
18     CMP R5, #0
19     BNE loop
20     B end
21
22 loop:
23     LDR R3, [R0]
24     ADD R0, R0, #4
25     TST R3, #1
26     ADDEQ R4, R4, R3
27     SUB R5, R5, #1
28     B start
29
30 end:
31     STR R4, [R1]
32     SWI 0x011
33
```

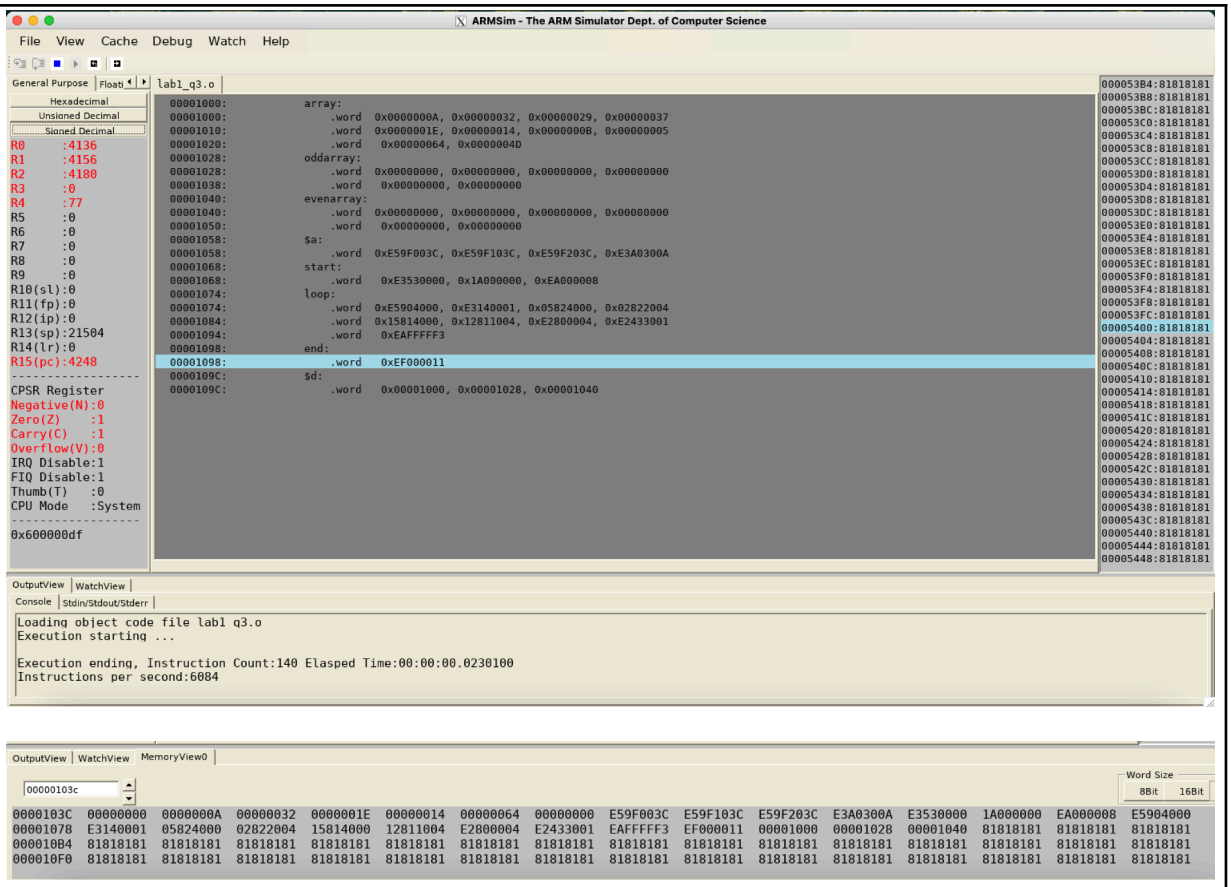



R0 :4204
R1 :4204
R2 :0
R3 :33
R4 :46
R5 :0
R6 :0
R7 :0
R8 :0
R9 :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4152
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
00001004:E59F10
00001008:E3A040
0000100C:E3A050
00001010:
00001010:E35500
00001014:1A0000
00001018:EA0000
0000101C:
0000101C:E59030
00001020:E28000
00001024:E31300
00001028:008440
0000102C:E24550
00001030:EAFFFF
00001034:
00001034:E58140
00001038:EF0000
0000103C:
0000103C:000010
00001040:000010
00001044:
00001044:
00001054:
00001064:
0000106C:
0000106C:
OutputView
WatchView
MemoryView0
0000106c
0000106C 0000002E 81818181

3	Write a ALP using ARMTDMI-ISA to store odd and even numbers in separate memory locations starting from LOCA and LOCB respectively ARRAY: .word 10,50,41,55,30,20,11,5,100,77 LOCA: .word 0,0,0,0,0,0 LOCB: .word 0,0,0,0,0,0

ARMSim_files > ASM lab1_q3.s

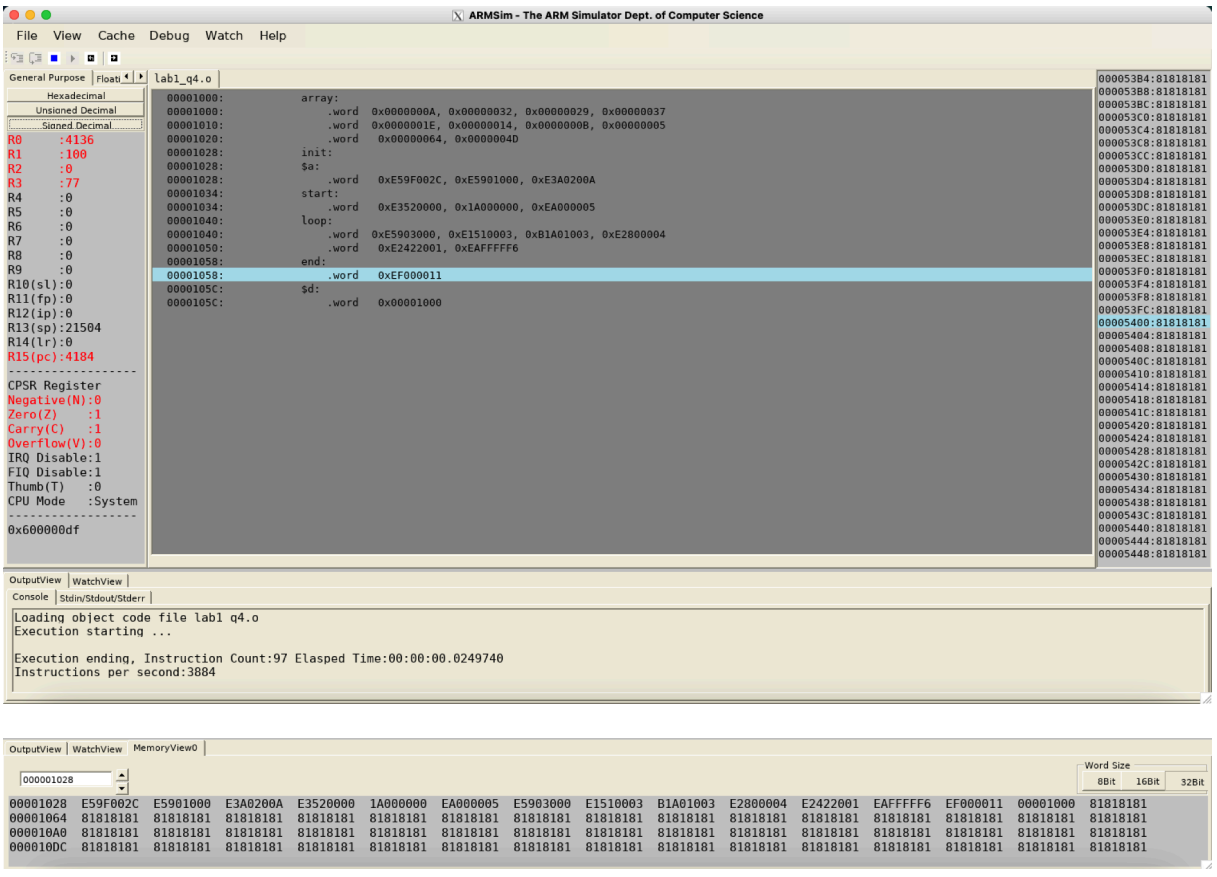
```
1   @ Write a ALP using ARMTDMI-ISA to store odd and even numbers in separate memory locations starting from LOCA and LOCB respectively
2   @ ARRAY: .word 10,50,41,55,30,20,11,5,100,77
3   @ LOCA: .word 0,0,0,0,0,0
4   @ LOCB: .word 0,0,0,0,0,0
5
6   .data
7   array: .word 10, 50, 41, 55, 30, 20, 11, 5, 100, 77
8   oddarray: .word 0, 0, 0, 0, 0, 0
9   evenarray: .word 0, 0, 0, 0, 0, 0
10
11  init:
12      LDR R0, =array
13      LDR R1, =oddarray
14      LDR R2, =evenarray
15      MOV R3, #10
16
17  start:
18      CMP R3, #0
19      BNE loop
20      B end
21
22  loop:
23      LDR R4, [R0]
24      TST R4, #1
25      STREQ R4, [R2]
26      ADDEQ R2, R2, #4
27      STRNE R4, [R1]
28      ADDNE R1, R1, #4
29      ADD R0, R0, #4
30      SUB R3, R3, #1
31      B start
32
33  end:
34      SWI 0x011
35
```



4	Write an ALP using ARM7TDMI to find the largest number from a given set of numbers: A: .word 10,50,41,55,30,20,11,5,100,77

ARMSim_files > ASM lab1_q4.s

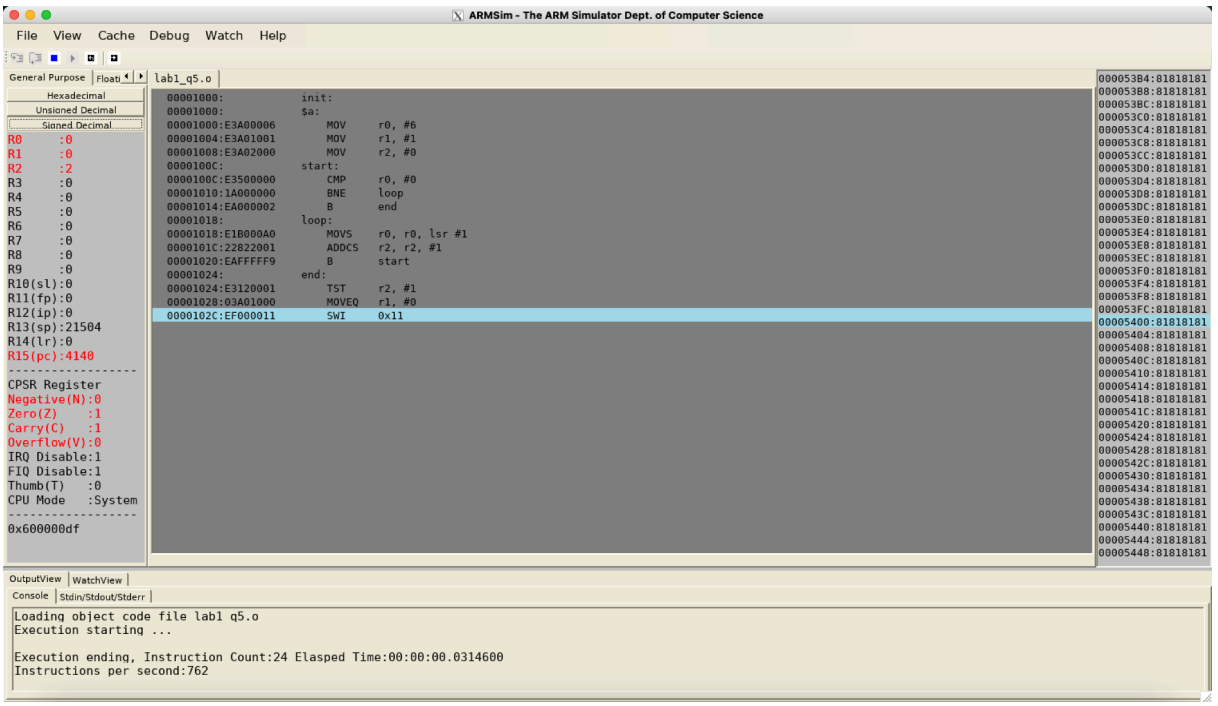
```
1   @ Write an ALP using ARM7TDMI to find the largest number from a given set of numbers:
2   @ A: .word 10,50,41,55,30,20,11,5,100,77
3
4   .data
5   array: .word 10, 50, 41, 55, 30, 20, 11, 5, 100, 77
6
7   init:
8       LDR R0, =array
9       LDR R1, [R0]
10      MOV R2, #10
11
12  start:
13      CMP R2, #0
14      BNE loop
15      B end
16
17  loop:
18      LDR R3, [R0]
19      CMP R1, R3
20      MOVLT R1, R3
21      ADD R0, R0, #4
22      SUB R2, R2, #1
23      B start
24
25  end:
26      SWI 0x011
27
```



	Assignments Questions
5	Write an ALP using ARM7TDMI to find whether the given number is even parity.

ARMSim_files > ASM lab1_q5.s

```
1   @ Write an ALP using ARM7TDMI to find whether the given number is even parity.
2   @ Set 0 on even parity and 1 on odd parity
3
4   .text
5
6   init:
7       MOV R0, #6
8       MOV R1, #1
9       MOV R2, #0
10
11  start:
12      CMP R0, #0
13      BNE loop
14      B end
15
16  loop:
17      MOVS R0, R0, LSR#1
18      ADDCS R2, R2, #1
19      B start
20
21  end:
22      TST R2, #1
23      MOVEQ R1, #0
24      SWI 0x011
25
```



6	Write an ALP using ARM7TDMI to multiplication of 38X72 without using MUL instructions. (Hint: barrel shifter instructions.) (Note :any number can be considered as multiplier)

ARMSim_files > ASM lab1_q6.s

```
1   @ Write an ALP using ARM7TDMI  to multiplication of 38X72  without using MUL instructions.
2   @ (Hint: barrel shifter instructions.)
3   @ (Note :any number can be considered as multiplier)
4
5   @ Logic used: break 38 into 32 + 4 + 2
6
7   .text
8   MOV R0, #38
9   MOV R1, #72
10  MOV R2, R1, LSL#5
11  ADD R3, R2, R1, LSL#2
12  ADD R4, R3, R1, LSL#1
13  SWI 0x011
14
```