Name: _____ Student ID: _____ Date: _____
Name: _____ Student ID: _____

Worksheet #6
Multiplication and Memory Access

Objectives

- To understand how to do multiplication operations in ARM processor
- To understand read and write memory instruction in ARM processor

1. Create a new project, then type and add the following code to the project.

```
        AREA  PROG6_1,  CODE,  READONLY
        ENTRY
start
        LDR   R0, =1024
        MOV   R1,  R0,  LSL  #X
loop    B       loop
        END
```

Program 6.1

2. Replace X in the program above with the value shown in Table 6.1.
3. Build and Run the program. Then observe and record the results in Table 6.1.

The values of	**The results from LSL instruction**	
X	**$R1_{16}$ (hex format)**	**$R1_{10}$ (decimal format)**
0		
1		
2		
3		
4		

Table 6.1

The results shown in Table 6.1 are equivalent to a mathematical expression

_____

_____

4. From Program 6.1, replace the LSL instruction with LSR instruction as follows:

MOV R1, R0, LSL #X ➔ **MOV R1, R0, LSR #X**

5. Repeat the step 2 and 3 again. Then observe and record the results in Table 6.2.

The values of	**The results from LSR instruction**	
X	**$R1_{16}$**	**$R2_{10}$**
0		
1		
2		
3		
4		

Table 6.2

The results shown in Table 6.2 are equivalent to mathematical expression

_____

_____

From Table 6.1 and 6.2, was the value in R0 changed after executing the program? What value is stored in R0?

_____

_____

6. Create a new project, then type and add the following code to the project.

```
        AREA  PROG6_2,  CODE,  READONLY
        ENTRY
start
        MOV   R0, #7
        MOV   R1,  R0,  LSL  #X
        ADD   R2,  R1,  R0,  LSL  #Y
loop    B         loop
        END
```

Program 6.2

7. Replace X and Y in the program above with the values as shown in Table 6.3.

8. Build and Run the program. Then observe and record the results in Table 6.3.

The values of		**The results from ADD instruction**		
X	**Y**	**R0**	**R1**	**R2**
1	0			
1	1			
1	2			
2	0			
2	2			
2	4			

Table 6.3

9. Write an assembly language program that will multiply the value in R0 by 25 and store the result in R2.
   Hint R0 x 25 = R0 x (16 + 8 + 1) = (R0 x 16) + (R0 x 8) + (R0 x 1)
   = (R0 x 2^4) + (R0 x 2^3) + (R0 x 2^0)

```
        AREA  R0x25,  CODE,  READONLY
        ENTRY
start




loop    B         loop
        END
```

10. Create a new project, then type and add the following code to the project.

```
        AREA  PROG6_2,  CODE,  READONLY
        ENTRY
start
        MOV   R0, #7
        MOV   R1,  R0,  LSL  #X
        RSB    R2,  R1,  R0,  LSL  #5
loop    B       loop
        END
```

Program 6.3

11. Replace X in the program above with the values as shown in Table 6.4.

12. Build and Run the program. Then observe and record the results in Table 6.4.

The values of	**The results from RSB instruction**		
X	**R0**	**R1**	**R2**
0			
1			
2			
3			
4			
5			

Table 6.4

13. Write an assembly language program that will multiply the value in R0 by 15 and store the result in R2.
    Hint using the **RSB** instruction. See also the hint above

```
        AREA  R0x15,  CODE,  READONLY
        ENTRY
start
loop    B       loop
        END
```

14. Create a new project, then type and add the following code to the project.

```
        AREA  PROG6_4,  CODE,  READONLY
        ENTRY
start
        MOV   R0,  #X
        MOV   R1,  #Y
        MULS   R2,  R0,  R1
loop    B       loop
        END
```

Program 6.4

15. Replace X and Y in the program above with the value shown in Table 6.5.

16. Build and Run the program. Then observe and record the results in Table 6.5.

The values of		**The results from MULS instruction**				
X	**Y**	**R2**	**N**	**Z**	**V**	**C**
120	0					
9	8					
-9	8					
-9	-8					
12345678	-98765432					
0x2BCDEF90	0x67531000					

Table 6.5

From Table 6.5, what do the negative condition code flags imply?

_____

_____

Try to change the MULS instruction to the MUL instruction. Examine the results in R2 and condition code flags.

Are the results in R2 produced by the last two values of X and Y in Table 6.5 correct? Why? Explain.

_____

_____

_____

_____

17. From Program 6.4, replace the MULS with SMULLS instruction as follows:

MULS R2, R0, R1 ➔ **UMULLS R3, R2, R0, R1**

18. Replace X and Y in the program above with the value shown in Table 6.6.

19. Build and Run the program. Then observe and record the results in Table 6.6.

The values of		The results from UMULLS instruction			
X	**Y**	**R2**	**R3**	**N**	**Z**
120	0				
9	8				
-9	8				
-9	-8				
12345678	-98765432				
0x2BCDEF90	0x67531000				

Table 6.6

20. From Program 6.4, replace the MULS with SMULLS instruction as follows:

MULS R2, R0, R1 ➔ SMULLS R3, R2, R0, R1

21. Replace X and Y in the program above with the value shown in Table 6.7.

22. Build and Run the program. Then observe and record the results in Table 6.7.

The values of		The results from SMULLS instruction			
X	**Y**	**R2**	**R3**	**N**	**Z**
120	0				
9	8				
-9	8				
-9	-8				
12345678	-98765432				
0x2BCDEF90	0x67531000				

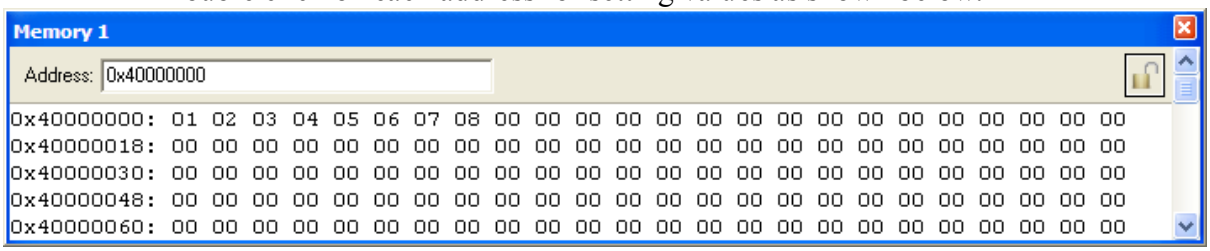
Table 6.7

23. Create a new project, then type and add the following code to the project.

```
        AREA  PROG6_5,  CODE,  READONLY
        ENTRY
start
        LDR   R0,   =0x40000000
        LDR   R1,   =0x40000030
        MOV  R2,   #4
        LDR   R3,  [R0,  R2]                -----④
        STR    R3,  [R1,  R2]               -----⑤
loop    B        loop
        END
```

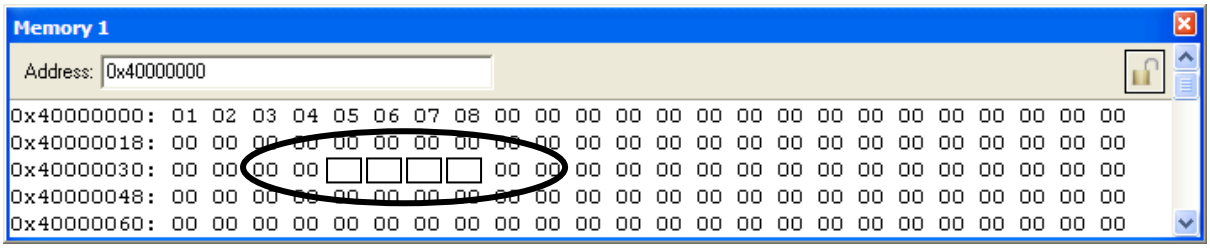
Program 6.5

24. Build the program. **Before running the program**, initialize the memory at 0x40000000 – 0x40000007as follows:
- Select the menu View -> Memory Windows -> Memory1.
- In the Address's field, type the memory address: 0x40000000.
- Double click on each address for setting values as shown below.



25. Run the program. Then observe and record the results

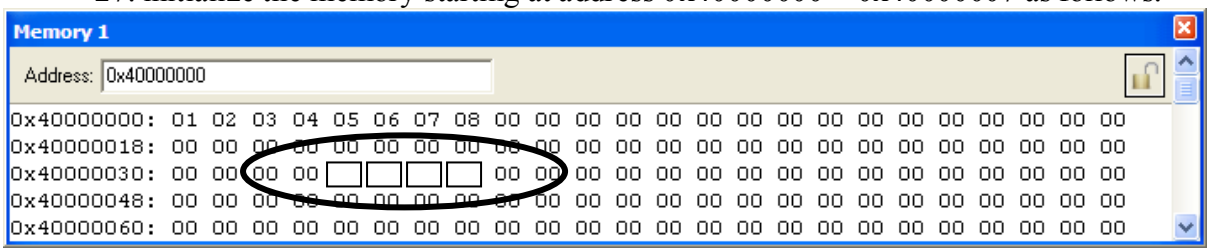
R0 = _____ R1 = _____
R2 = _____ R3 = _____



26. From Program 6.5, replace the instructions ④ and ⑤ with the instructions as follows:

LDR R3, [R0, R2] ➔ **LDR R3, [R0, R2]!** -----④
STR R3, [R1, R2] ➔ **STR R3, [R1, R2]!** -----⑤

27. initialize the memory starting at address 0x40000000 – 0x40000007 as follows:



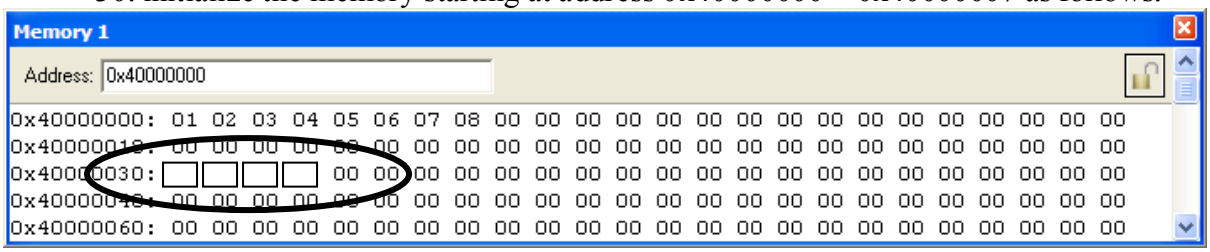
28. Run the program. Then observe and record the results

R0 = _____ R1 = _____
R2 = _____ R3 = _____

29. From Program 6.5, replace the instructions ④ and ⑤ with the instructions as follows:

LDR R3, [R0, R2] ➔ **LDR R3, [R0], R2** -----④
STR R3, [R1, R2] ➔ **STR R3, [R1], R2** -----⑤

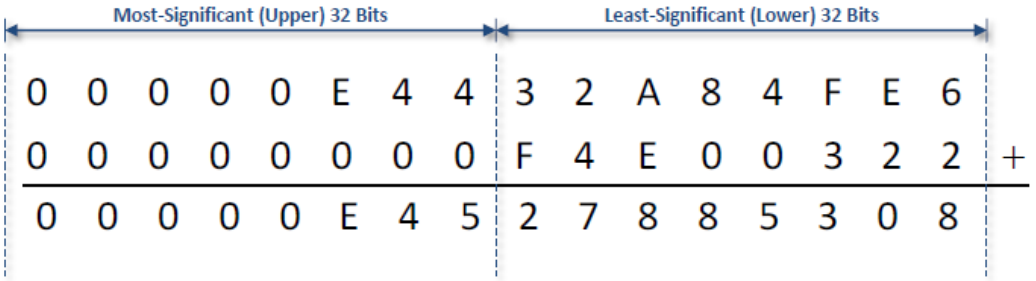
30. initialize the memory starting at address 0x40000000 – 0x40000007 as follows:



31. Run the program. Then observe and record the results

R0 = _____ R1 = _____
R2 = _____ R3 = _____

32. Write an assembly program that adds two 64-bit values together as the detail shown below:



Given that the first operand is stored in memory at 0x40000000 to 0x40000007 and the second operand is stored in memory at 0x40000018 to 0x4000001F. The result of addition will also be stored in memory starting at 0x40000030.