

BIRZEIT UNIVERSITY

Computer Systems Engineering

Computer Organization and Microprocessor

ENCS2380

Assembly Project

Student's Name: Tariq Odeh

Student's No: **1190699**

Sec: **1**

Instructor: Dr. Abualseoud Hanani

Date: 9-6-2021

Table of content

Summary	3
Code	4
ASCII code	6
Test & Output	7

Summary

In project, I used Keil Version 4 to implement a encryption and decryption algorithm in ARM assembly. The encryption method that I have used in project is Rotate left ASCII code of each character two bits, then its key is rotate right two bits (decryption), based on [699 mod 3] +1 = 1.

The text message that I used "Don't let your fear destroy you!", I stored it in memory address (0x4000000) to (0x40000083), then I applied the encryption method on every single character and then stored the encrypted message in memory address (0x400000EC) to (0x4000016F), then I take the encrypted message from address (0x400000EC) and applied the reverse of the encryption process (decryption) and then stored the encrypted message in memory address (0x400001D8) to (0x4000025B).

In code I use two procedures: the first one to applied encryption and the second to applied reverse of the encryption (decryption).

Code

```
AREA Tariq_Odeh,CODE,READONLY
        ENTRY
        LDR R0,=0x40000000
STRING DCB
                 "Don't let your fear destroy you!",0
        ADR R1,STRING
STORE
        LDRB R2,[R1]
        CMP R2,#0
        BEQ FINISH
        STR R2,[R0]
        ADD R1,R1,#1
        ADD R0,R0,#4
        B STORE
FINISH
        MOV R7,#0
        STR R7,[R0]
        BL encryption
        BL decryption
HERE BAL HERE
encryption
        LDR R8,=0x40000000
        LDR R9,=0x400000EC
READ
        LDR R2,[R8]
        CMP R2,#0
        BEQ DONE
        MOV R3,R2, ROR #30
```

	STR R3,[R9]
	ADD R8,R8,#4
	ADD R9,R9,#4
	B READ
DONE	
	MOV R7,#0
	STR R7,[R9]
	BX LR
decryptio	on .
	LDR R5,=0x400000EC
	LDR R6,=0x400001D8
AGAIN	
	LDR R2,[R5]
	CMP R2,#0
	BEQ EXIT
	MOV R3,R2, ROR #2
	STR R3,[R6]
	ADD R5,R5,#4
	ADD R6,R6,#4
	B AGAIN
EXIT	
	MOV R7,#0
	STR R7,[R6]
	BX LR
	END
	decryptic

ASCII Code

In the table below there is ASCII code for each character that were used in the code (text), an operation was performed encryption and decryption for each character.

Char	Hex	Hex after Rotate left two bits	_
_		(encryption)	(decryption)
D	44	110	44
0	6F	1BC	6F
n	6E	1B8	6E
1	27	9C	27
t	74	1D0	74
space	20	80	20
I	6C	1B0	6C
е	65	194	65
t	74	1D0	74
space	20	80	20
У	79	1E4	79
0	6F	1BC	6F
u	75	1D4	75
r	72	1C8	72
space	20	80	20
f	66	198	66
е	65	194	65
а	61	184	61
r	72	1C8	72
space	20	80	20
d	64	190	64
е	65	194	65
S	73	1CC	73
t	74	1D0	74
r	72	1C8	72
0	6F	1BC	6F
У	79	1E4	79
space	20	80	20
У	79	1E4	79
0	6F	1BC	6F
u	75	1D4	75
!	21	84	21
zero	00	00	00

Table 1: ASCII code for each character that were used in the code

Test & Output

To check, I do the following:

- 1- Translate the file.
- 2- Build the target files.
- 3- Rebuild all target files.
- 4- Start Debug Session.
- 5- Show the output (result).

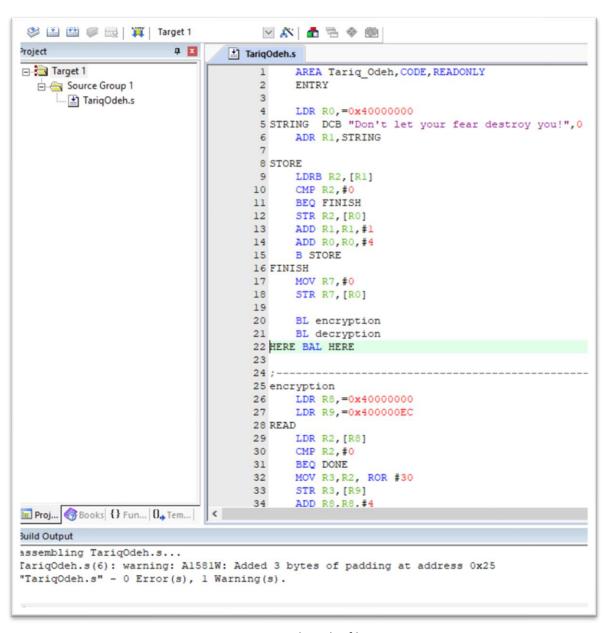


Figure 1: Translate the file.

```
roject

★ TariqOdeh.s

□ Target 1
                                   1
                                        AREA Tariq Odeh, CODE, READONLY
                                   2
                                        ENTRY
  3
     TariqOdeh.s
                                   4
                                        LDR R0,=0x40000000
                                   5 STRING DCB "Don't let your fear destre
                                   6
                                        ADR R1, STRING
                                   8 STORE
                                  9
                                       LDRB R2, [R1]
                                  10
                                      CMP R2,#0
                                  11
                                       BEQ FINISH
                                  12
                                       STR R2, [R0]
                                       ADD R1, R1, #1
                                  13
                                        ADD RO, RO, #4
                                  14
                                  15
                                        B STORE
                                  16 FINISH
                                       MOV R7, #0
                                  17
                                  18
                                        STR R7, [R0]
                                  19
                                  20
                                        BL encryption
                                        BL decryption
                                  21
                                  22 HERE BAL HERE
                                  23
                                  24 ; -----
                                  25 encryption
                                  26 LDR R8,=0x40000000
                                  27
                                       LDR R9,=0x400000EC
                                  28 READ
                                  29
                                        LDR R2, [R8]
                                       CMP R2,#0
                                  30
                                       BEQ DONE
                                  31
                                  32
                                       MOV R3, R2, ROR #30
                                  33
                                       STR R3, [R9]
                                  34
                                        ADD R8.R8.#4
E Proj... ♦ Books {} Fun... | D→ Tem...
Build Output
Build target 'Target 1'
linking...
?rogram Size: Code=204 RO-data=0 RW-data=0 ZI-data=0
'.\projl.axf" - 0 Error(s), 0 Warning(s).
```

Figure 2: Build the target files.

```
roject
                              * TariqOdeh.s
□ Target 1
                                       AREA Tariq Odeh, CODE, READONLY
                                  2
                                       ENTRY
  □ Gource Group 1
                                  3
     TariqOdeh.s
                                  4
                                       LDR R0,=0x40000000
                                  5 STRING DCB "Don't let your fear destro
                                       ADR R1, STRING
                                  6
                                  8 STORE
                                  9
                                       LDRB R2, [R1]
                                 10
                                       CMP R2,#0
                                 11
                                       BEQ FINISH
                                 12
                                      STR R2, [R0]
                                 13
                                      ADD R1, R1, #1
                                 14
                                      ADD RO, RO, #4
                                 15
                                      B STORE
                                 16 FINISH
                                 17
                                       MOV R7, #0
                                 18
                                       STR R7, [R0]
                                 19
                                 20
                                       BL encryption
                                 21
                                       BL decryption
                                 22 HERE BAL HERE
                                 23
                                 24 ;-----
                                 25 encryption
                                      LDR R8,=0x40000000
                                 26
                                 27
                                      LDR R9,=0x400000EC
                                 28 READ
                                 29
                                       LDR R2, [R8]
                                 30
                                       CMP R2,#0
                                 31
                                      BEQ DONE
                                 32
                                      MOV R3, R2, ROR #30
                                 33
                                       STR R3, [R9]
                                 34
                                       ADD R8.R8.#4
Build Output
rariqOdeh.s(6): warning: A1581W: Added 3 bytes of padding at address 0x25
linking ...
Program Size: Code=204 RO-data=0 RW-data=0 ZI-data=0
".\projl.axf" - 0 Error(s), 1 Warning(s).
```

Figure 3: Rebuild the target files.

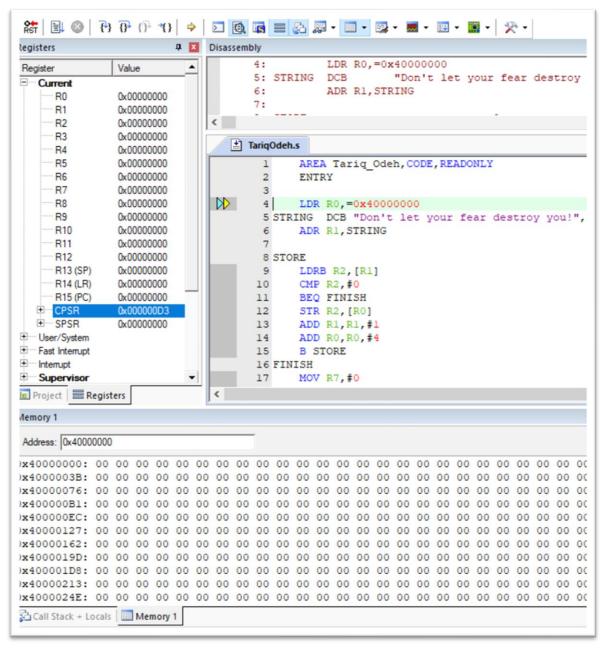


Figure 4: Start Debug Session.

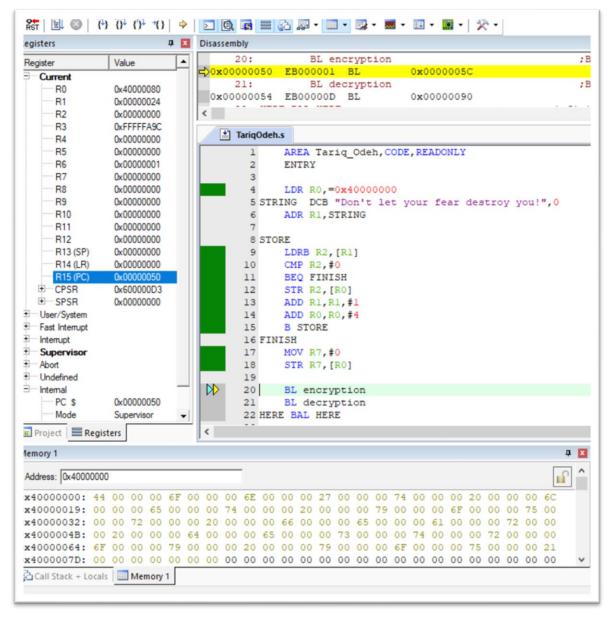


Figure 5: Store String in memory address (0x40000000) to (0x40000083).

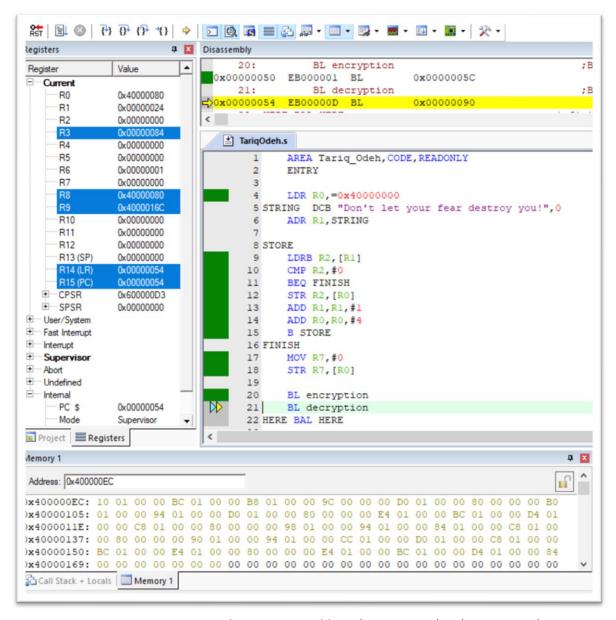


Figure 6: Store encryption result in memory address (0x400000EC) to (0x4000016F).

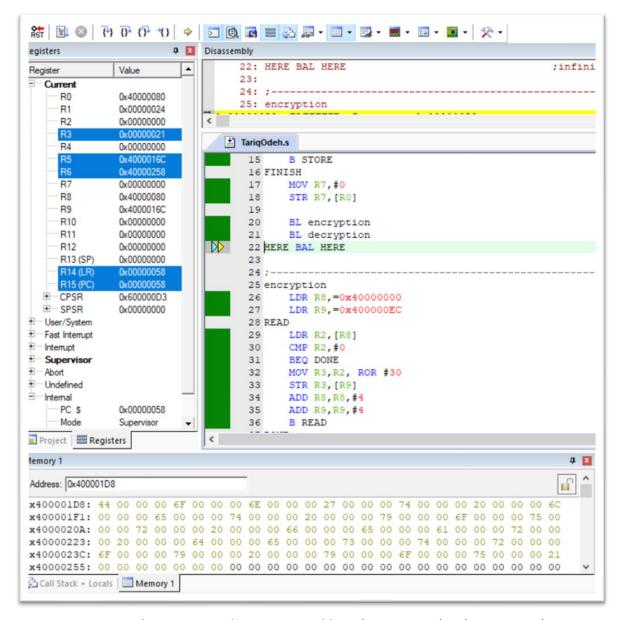


Figure 7: Store decryption result in memory address (0x400001D8) to (0x4000025B).

