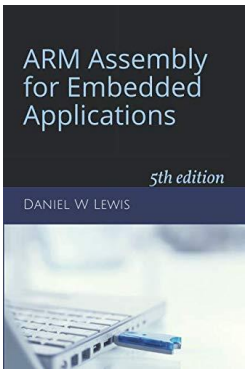

ARM Instructions Worksheet #7

Shift Instructions

LSL, LSR, ASR, ROR, and RRX

Prerequisite Reading: Chapter 7
Revised: March 26, 2020

Objectives: *To use the web-based simulator ("CPULator") to better understand ...*

1. The operation of the basic shift instructions (LSL, LSR, ASR, ROR, and RRX)
2. The relationship between the shift instructions and the carry flag (C)

To do offline: Answer the questions that follow the listing below. (Numbers at far left are memory addresses.)

```
                    .syntax     unified
                    .global     _start

00000000  _start:   LDR         R0,=0x00000001      //*** EXECUTION STARTS HERE ***
00000004            LSLS        R1,R0,1

00000008            LDR         R0,=0x80000001
0000000C            LSLS        R1,R0,1

00000010            LDR         R0,=0x40000000
00000014            LSR         R1,R0,1
00000018            ASR         R1,R0,1

0000001C            LDR         R0,=0x80000000
00000020            LSR         R1,R0,1
00000024            ASR         R1,R0,1

00000028            ROR         R1,R0,1
0000002C            RRX         R1,R0

00000030  done:     B           done

                    .end
```

What is in R1 and R0 after the instructions at 00000000_{16} and 00000004_{16}? R1 (as hexadecimal) ____ R0 (as hexadecimal) ____

What is in the carry flag (CPSR bit 29) by the LSLS instruction at address 00000004_{16}? Carry Flag (C) ☐

What is in R1 and R0 after the instructions at 00000008_{16} and $0000000C_{16}$? R1 (as hexadecimal) ____ R0 (as hexadecimal) ____

What is in the carry flag (CPSR bit 29) by the LSLS instruction at address $0000000C_{16}$? Carry Flag (C) ☐

What is in R1 and R0 after the instructions at 00000010_{16} and 00000014_{16}? R1 (as hexadecimal) ____ R0 (as hexadecimal) ____

What is in `R1` and `R0` after the `ASR` instruction at address 00000018_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

What is in `R1` and `R0` after the instructions at $0000001C_{16}$ and 00000020_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

What is in `R1` and `R0` after the `ASR` instruction at address 00000024_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

What is in `R1` and `R0` after the `ROR` instruction at address 00000028_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

What is in `R1` and `R0` after the `RRX` instruction at address $0000002C_{16}$?

R1 (as hexadecimal) | R0 (as hexadecimal)

Getting ready: Now use the simulator to collect the following information and compare to your earlier answers.

1. Click here to open a browser for the ARM instruction simulator with pre-loaded code.

***Step 1:** Press F2 exactly **2** times to execute the instructions at addresses 00000000_{16} and 00000004_{16}*

What is in `R1` and `R0` after the instructions at 00000000_{16} and 00000004_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

What is in the carry flag (`CPSR` bit 29) by the `LSLS` instruction at address 00000004_{16}?

Carry Flag (C)

***Step 2:** Press F2 exactly **2** times to execute the instructions at addresses 00000008_{16} and $0000000C_{16}$*

What is in `R1` and `R0` after the instructions at 00000008_{16} and $0000000C_{16}$?

R1 (as hexadecimal) | R0 (as hexadecimal)

What is in the carry flag (`CPSR` bit 29) by the `LSLS` instruction at address $0000000C_{16}$?

Carry Flag (C)

***Step 3:** Press F2 exactly **2** times to execute the instructions at addresses 00000010_{16} and 00000014_{16}*

What is in `R1` and `R0` after the instructions at 00000010_{16} and 00000014_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

***Step 4:** Press F2 exactly **once** to execute the `ASR R1,R0,1` instruction at address 00000018_{16}.*

What is in `R1` and `R0` after the `ASR` instruction at address 00000018_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

***Step 5:** Press F2 exactly **2** times to execute the instructions at addresses $0000001C_{16}$ and 00000020_{16}.*

What is in `R1` and `R0` after the instructions at $0000001C_{16}$ and 00000020_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

***Step 6:** Press F2 exactly **once** to execute the second `ASR R1,R0,1` instruction at address 00000024_{16}.*

What is in `R1` and `R0` after the `ASR` instruction at address 00000024_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

***Step 7:** Press F2 exactly **once** to execute the `ROR R1,R0,1` instruction at address 00000028_{16}.*

What is in `R1` and `R0` after the `ROR` instruction at address 00000028_{16}?

R1 (as hexadecimal) | R0 (as hexadecimal)

***Step 8:** Press F2 exactly **once** to execute the `RRX R1,R0` instruction at address $0000002C_{16}$.*

What is in `R1` and `R0` after the `RRX` instruction at address $0000002C_{16}$?

R1 (as hexadecimal) | R0 (as hexadecimal)