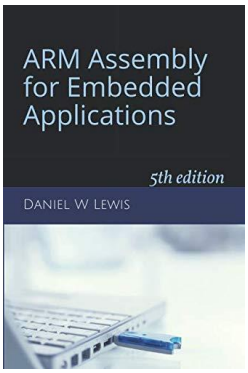
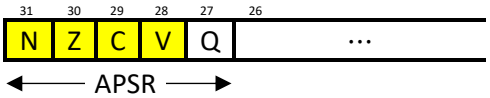


ARM Instructions Worksheet #4

Addition and Subtraction

And their effect on the NZCV Flags in the CPSR register:



Prerequisite Reading: Chapter 5
Revised: March 26, 2020

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

1. That the flags are not affected unless the letter '`S`' is appended to the instruction.
2. How the oVerflow flag (`V`) is affected as the result of a signed addition or a signed subtraction.
3. How the Carry flag (`C`) is affected as the result of an unsigned addition or an unsigned subtraction.
4. How to Zero flag (`Z`) and Negative flag (`N`) are affected as the result of an arithmetic operation.

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

```
                    .syntax     unified
                    .global     _start

00000000  _start:   MSR         APSR_nzcvq,0        // *** EXECUTION STARTS HERE ***
00000004            LDR         R0,=0x40000001
00000008            ADD         R1,R0,R0
0000000C            ADDS        R1,R0,R0
00000010            SUBS        R1,R0,R0
00000014            ADCS        R1,R1,0
00000018            RSBS        R1,R0,1

0000001C  done:     B           done                // infinite loop

                    .end
```

	R0 (as unsigned decimal)	R0 (as signed decimal)	N	Z	C	V
What is left in `R0` by the `LDR` at 00000004_{16}?			0	0	0	0

	R1 (as unsigned decimal)	R1 (as signed decimal)	N	Z	C	V
What is in `R1` and flags after `ADD` at 00000008_{16}? Unsigned Overflow? ☐ Signed Overflow? ☐						
What is in `R1` and flags after `ADDS` at $0000000C_{16}$? Unsigned Overflow? ☐ Signed Overflow? ☐						
What is in `R1` and flags after `SUBS` at 00000010_{16}? Unsigned Overflow? ☐ Signed Overflow? ☐						
What is in `R1` and flags after `ADCS` at 00000014_{16}? Unsigned Overflow? ☐ Signed Overflow? ☐						
What is in `R1` and flags after `RSBS` at 00000018_{16}? Unsigned Overflow? ☐ Signed Overflow? ☐						

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.

   Note: You can change the number format in the "Settings" window between hex, unsigned decimal and signed decimal as needed. It's probably easiest to go through the instructions three times – once to get the unsigned decimal values, once to get the signed decimal values, and once to get the flag values (using hex format). Before each pass, press Ctrl-R to restart the simulation.

***Step 1:** Press F2 exactly **2** times to execute the `MSR, LDR` pseudo-instruction (`MOV`) sequence*

What is left in `R0` by the `LDR` at 00000004_{16}?	R0 (as unsigned decimal)	R0 (as signed decimal)	N	Z	C	V
			0	0	0	0

***Step 2:** Press F2 exactly **once** to execute the `ADD R1,R0,R0` instruction.*

What is in `R1` and flags after `ADD` at 00000008_{16}?	R1 (as unsigned decimal)	R1 (as signed decimal)	N	Z	C	V

Unsigned Overflow? ☐ Signed Overflow? ☐

***Step 3:** Press F2 exactly **once** to execute the `ADDS R1,R0,R0` instruction.*

What is in `R1` and flags after `ADDS` at $0000000C_{16}$?	R1 (as unsigned decimal)	R1 (as signed decimal)	N	Z	C	V

Unsigned Overflow? ☐ Signed Overflow? ☐

***Step 4:** Press F2 exactly **once** to execute the `SUBS R1,R0,R0` instruction.*

What is in `R1` and flags after `SUBS` at 00000010_{16}?	R1 (as unsigned decimal)	R1 (as signed decimal)	N	Z	C	V

Unsigned Overflow? ☐ Signed Overflow? ☐

***Step 5:** Press F2 exactly **once** to execute the `ADCS R1,R1,0` instruction.*

What is in `R1` and flags after `ADCS` at 00000014_{16}?	R1 (as unsigned decimal)	R1 (as signed decimal)	N	Z	C	V

Unsigned Overflow? ☐ Signed Overflow? ☐

***Step 6:** Press F2 exactly **once** to execute the `RSBS R1,R1,1` instruction.*

What is in `R1` and flags after `RSBS` at 00000018_{16}?	R1 (as unsigned decimal)	R1 (as signed decimal)	N	Z	C	V

Unsigned Overflow? ☐ Signed Overflow? ☐