### EEEN 3449

### Microprocessor Systems

# Condition Flags and Rotate Instruction

Tyler Hurson

Spring 2017

**I. INTRODUCTION**

* 1. **Purpose**

The purpose of this experiment is to familiarize oneself the conditional code register and how it is affected by rotate, addition, and subtraction instructions.

* 1. **Problem**

Comparison and branch instructions will be used in order to control program flow and execute a block of code several times. The most common use of looping mechanisms is to iterate over an array of data. An Assembly program that contains comparison and branching instructions that iterates over each element in a 5-byte array will be executed. Six varieties of this program will be executed.

In Program A-1 (Appendix A), the array is located at the end of the program. Two bytes are reserved for the sum and one byte is reserved for the loop counter. The loop counter is initialized to 0 and counts upwards to 5, accessing each element in the array using offset indexed addressing and adding it to the immediate sum.

In Program B-1 (Appendix B), the array is located at the end of the program. The loop counter is initialized to 5 and counts downwards to 0. The array is summed from the first element to the last.

In Program A-2 (Appendix C), the array is located at the end of the program. The loop counter is initialized to 0 and counts upwards to 5. The array is summed from the last element to the first.

In Program B-2 (Appendix D), the array is located at the end of the program. The loop counter is initialized to 5 and counts downwards to 5. The array is summed from the last element to the first.

In Program A-3 (Appendix E), the array is located at the beginning of the program. The loop counter is initialized to 0 and counts upwards to 5. The array is summed from the first element to the last.

In Program B-3 (Appendix F), the array is located at the beginning of the program. The loop counter is initialized to 5 and counts downwards to 0. The array is summed from the first element to the last.

* 1. **Scope**

The scope of this experiment is limited to the HCS12 microcontroller. Only a few basic instructions will be used from the HCS12 instruction set, including storing and loading instructions, comparison instructions, arithmetic instructions, and branching instructions.

**II. TEST AND EVALUATION**

**2.1 Apparatus**

The equipment used in this test includes: Dragon12-Junior development board, USB power cord, and laptop PC with AsmIDE.

**2.2 Procedure**

1. The development board was connected to the computer.
2. The COM port number was determined under Device Manager on PC. AsmIDE was launched. Under View -> Options -> COM Port, the COM port was set to the device’s number. The Terminal Window was enabled. Under Set COM Options, the default values were restored.
3. Program A-1 was opened, and then assembled. After no errors were recorded, program A was downloaded into the development board, by typing load in the Terminal Window in AsmIDE, then downloading the program.
4. br 2007 was typed to set a breakpoint. g 2000 was typed to execute the program. t 100 was typed repeatedly to trace the program, line by line, until the program ended. As the program was traced, the immediate sum (Y) was verified correct for each iteration of the loop.
5. Program B-1 was opened, and then assembled. After no errors were recorded, program B-1 was downloaded into the development board.
6. br 2007 was typed to set a breakpoint. g 2000 was typed to execute the program. t 100 was typed repeatedly to trace the program, line by line, until the program ended. As the program was traced, the immediate sum (Y) was verified correct for each iteration of the loop.
7. Program A-2 was opened, and then assembled. After no errors were recorded, program A-2 was downloaded into the development board.
8. br 2007 was typed to set a breakpoint. g 2000 was typed to execute the program. t 100 was typed repeatedly to trace the program, line by line, until the program ended. As the program was traced, the immediate sum (Y) was verified correct for each iteration of the loop.
9. Program B-2 was opened, and then assembled. After no errors were recorded, program B-2 was downloaded into the development board.
10. br 2007 was typed to set a breakpoint. g 2000 was typed to execute the program. t 100 was typed repeatedly to trace the program, line by line, until the program ended. As the program was traced, the immediate sum (Y) was verified correct for each iteration of the loop.
11. Program A-3 was opened, and then assembled. After no errors were recorded, program A-3 was downloaded into the development board.
12. br 2007 was typed to set a breakpoint. g 2000 was typed to execute the program. t 100 was typed repeatedly to trace the program, line by line, until the program ended. As the program was traced, the immediate sum (Y) was verified correct for each iteration of the loop.
13. Program B-3 was opened, and then assembled. After no errors were recorded, program B-3 was downloaded into the development board.
14. br 2007 was typed to set a breakpoint. g 2000 was typed to execute the program. t 100 was typed repeatedly to trace the program, line by line, until the program ended. As the program was traced, the immediate sum (Y) was verified correct for each iteration of the loop.

**III. RESULTS**

* 1. **Data**

For each program, the immediate sum for each iteration was recorded (Table 1). The full line-by-line output of each program is found under its respective Appendix.

Table 1: Immediate Sums of each Program

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Program/**  **Loop Count** | **A-1** | **B-1** | **A-2** | **B-2** | **A-3** | **B-3** |
| **1** | 0B | 0B | 14 | 14 | 0B | 0B |
| **2** | 19 | 19 | 25 | 25 | 19 | 19 |
| **3** | 28 | 28 | 34 | 34 | 28 | 28 |
| **4** | 39 | 39 | 42 | 42 | 39 | 39 |
| **5** | 4D | 4D | 4D | 4D | 4D | 4D |

1. **CONCLUSION**

**4.1 Assessment**

This experiment served as an introduction to the basic looping mechanisms of the Assembly language. Six varieties of an Assembly program were tested, each involving differences in the way the loop counter was incremented/decremented and the way the array was added.

**APPENDIX A**

**ASSEMBLY PROGRAM A**

org $1500 ; program start

orcc #$05 ; Set zero and carry flag

andcc #$F5 ; Clear overflow and negative flag

ldaa #$50 ; A = $50

suba #40 ; A = A - 40

tsta

adda #$50 ; A = A + $50

lsra ; logical shift right A

rola ; rotate A left

lsla ; logical shift left A

swi

end

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ORCC #$05** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1502 | 3C00 | 0000 | 0000 | 00 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **ANDCC #$F5** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1504 | 3C00 | 0000 | 0000 | 00 | 00 | 1001 0101 |
|  |  |  |  |  |  |  |
| **LDAA #$50** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1506 | 3C00 | 0000 | 0000 | 00 | 00 | 1001 0101 |
|  |  |  |  |  |  |  |
| **SUBA #$28** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1508 | 3C00 | 0000 | 0000 | 50 | 00 | 1001 0001 |
|  |  |  |  |  |  |  |
| **TSTA** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150A | 3C00 | 0000 | 0000 | 28 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **ADDA #$50** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150B | 3C00 | 0000 | 0000 | 28 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **LSRA** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150D | 3C00 | 0000 | 0000 | 78 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **ROLA** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150E | 3C00 | 0000 | 0000 | 3C | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **ASLA** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150F | 3C00 | 0000 | 0000 | 78 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **SWI** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1510 | 3C00 | 0000 | 0000 | F0 | 00 | 1001 1010 |

**APPENDIX B**

**ASSEMBLY PROGRAM B**

org $1500 ; program start

orcc #$04 ; Set zero flag

andcc #$F4 ; Clear overflow, negative, and carry flag

ldaa #$50 ; A = $50

suba #40 ; A = A - 40

tsta

adda #$40 ; A = A + $40

adda #$78 ; A = A + $78

lsra ; logical shift right A

rola ; rotate A left

adda #$CF ; logical shift left A

swi

end

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ORCC #$04** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1502 | 3C00 | 0000 | 0000 | 00 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **ANDCC #$F4** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1504 | 3C00 | 0000 | 0000 | 00 | 00 | 1001 0100 |
|  |  |  |  |  |  |  |
| **LDAA #$50** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1506 | 3C00 | 0000 | 0000 | 00 | 00 | 1001 0100 |
|  |  |  |  |  |  |  |
| **SUBA #$28** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1508 | 3C00 | 0000 | 0000 | 50 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **TSTA** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150A | 3C00 | 0000 | 0000 | 28 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **ADDA #$40** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150B | 3C00 | 0000 | 0000 | 28 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **ADDA #$78** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150D | 3C00 | 0000 | 0000 | 68 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **LSRA** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150F | 3C00 | 0000 | 0000 | E0 | 00 | 1011 1010 |
|  |  |  |  |  |  |  |
| **ROLA** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1510 | 3C00 | 0000 | 0000 | 70 | 00 | 1011 0000 |
|  |  |  |  |  |  |  |
| **ADDA #$CF** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1511 | 3C00 | 0000 | 0000 | E0 | 00 | 1011 1010 |
|  |  |  |  |  |  |  |
| **SWI** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1513 | 3C00 | 0000 | 0000 | AF | 00 | 1001 1001 |

**APPENDIX C**

**ASSEMBLY PROGRAM C**

org $1500 ; program start

ldd #$8287 ; D = $8287

addd #$8998 ; D += 8998

std $1502 ; $1502 = D

ldaa #$95 ; A = $95

adca #$45 ; A += carry + $45

staa $1501 ; $1501 = A

ldaa #$A2 ; A = $A2

adca #$78 ; A += carry + $78

staa $1500 ; $1500 = A

swi

end

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **LDD #$8287** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1502 | 3C00 | 0000 | 0000 | 00 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **ADDD #$8998** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1505 | 3C00 | 0000 | 0000 | 82 | 87 | 1001 1000 |
|  |  |  |  |  |  |  |
| **STD $1502** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1508 | 3C00 | 0000 | 0000 | 0C | 1F | 1001 0011 |
|  |  |  |  |  |  |  |
| **LDAA #$95** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150B | 3C00 | 0000 | 0000 | 0C | 1F | 1001 0001 |
|  |  |  |  |  |  |  |
| **ADCA #$45** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150D | 3C00 | 0000 | 0000 | 95 | 1F | 1001 1001 |
|  |  |  |  |  |  |  |
| **STAA $1501** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150F | 3C00 | 0000 | 0000 | DB | 1F | 1001 1000 |
|  |  |  |  |  |  |  |
| **LDAA #$A2** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1512 | 3C00 | 0000 | 0000 | DB | 1F | 1001 1000 |
|  |  |  |  |  |  |  |
| **ADCA #$78** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1514 | 3C00 | 0000 | 0000 | A2 | 1F | 1001 1000 |
|  |  |  |  |  |  |  |
| **STAA $1500** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1516 | 3C00 | 0000 | 0000 | 1A | 1F | 1001 0001 |
|  |  |  |  |  |  |  |
| **SWI** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1519 | 3C00 | 0000 | 0000 | 1A | 1F | 1001 0001 |

**APPENDIX D**

**ASSEMBLY PROGRAM D**

org $1500 ; program start

nop

nop

ldd #$4321 ; D = $4321

subd #$6789 ; D = D - $6789

std $1502 ; $1502 = D

ldaa #$65 ; A = $65

sbca #$45 ; A = A - carry - $45

staa $1501 ; $1501 = A

ldaa #$87 ; A = $87

sbca #$23 ; A = A - carry - $23

staa $1500 ; $1500 = A

swi

end

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **LDD #$4321** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1502 | 3C00 | 0000 | 0000 | 00 | 00 | 1001 0000 |
|  |  |  |  |  |  |  |
| **SUBD #$6789** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1505 | 3C00 | 0000 | 0000 | 43 | 21 | 1001 0000 |
|  |  |  |  |  |  |  |
| **STD $1502** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1508 | 3C00 | 0000 | 0000 | DB | 98 | 1001 1001 |
|  |  |  |  |  |  |  |
| **LDAA #$65** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150B | 3C00 | 0000 | 0000 | DB | 98 | 1001 1001 |
|  |  |  |  |  |  |  |
| **SBCA #$45** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150D | 3C00 | 0000 | 0000 | 65 | 98 | 1001 0001 |
|  |  |  |  |  |  |  |
| **STAA $1501** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 150F | 3C00 | 0000 | 0000 | 1F | 98 | 1001 0000 |
|  |  |  |  |  |  |  |
| **LDAA #$87** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1512 | 3C00 | 0000 | 0000 | 1F | 98 | 1001 0000 |
|  |  |  |  |  |  |  |
| **SBCA #$23** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1514 | 3C00 | 0000 | 0000 | 87 | 98 | 1001 1000 |
|  |  |  |  |  |  |  |
| **STAA $1500** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1516 | 3C00 | 0000 | 0000 | 64 | 98 | 1001 0010 |
|  |  |  |  |  |  |  |
| **SWI** | |  |  |  |  |  |
| **PC** | **SP** | **X** | **Y** | **A** | **B** | **SXHI NZVC** |
| 1519 | 3C00 | 0000 | 0000 | 64 | 98 | 1001 0000 |