**Kettering University**

Microcomputers I

**Class Exercise Packet II**

Spring 2022

Name: Dylan Lozon

**Notes**

* Answer the questions provided in this handout during normal class times when the instructor asks you to do so, and then upload it to Bb (to get extra credit) using one of the following methods when you are notified:
* You may directly answer the questions in the electronic copy of the handout posted on Bb, and then convert it to .pdf format to submit the packet.
* You may answer the questions in the paper copy, and then scan and convert it to .pdf format to submit the packet.
* (Preferred) You may answer the questions in the paper copy, then transfer your answers to the electronic copy, and convert it to .pdf format to submit the packet.
* In multiple-choice questions, highlight the correct choice(s).
* Exercises are **open** book/notes unless specified otherwise.
* If you know the answers, please **teach** other students; otherwise, **learn** from other students, but do NOT copy from them!
* If you **run out of time** on a question, answer that question later but **before** you walk into the classroom next time.
* **Do NOT hesitate to stop by my office if you need help!**
* This handout (and not anything else!) in .pdf may be collected for grading purposes electronically **anytime**. So, keep it updated!

1. Write a program to perform the following task:

$1010 ← ($1000) + ($1002) – ($1005)

ldaa $1000

adda $1002

suba $1005

staa $1010

1. Assemble the branch instruction:

loop: aba

ldx $2000

decb

decb

bne loop

done: staa $2100

Offset for bne = -7

Machine code for bne loop: $26 F7

1. Determine (in your head as much as you can) whether or not the 10 conditional branch instructions would be taken (successful) if they were placed right after the compare instruction. For each branch instruction type either a Yes or No in the space provided in the table. What happens to the condition flags?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code | Difference | N | Z | V | C | beq | bne | bhi | blo | bhs | bls | bgt | blt | bge | ble |
| ldd #$840D  cpd #$7A10 | 09FD | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 1 |
|  | 8 | 4 | 0 | D |
|  | 8 | 5 | E | F |
|  | 0 | 9 | **F** | **D** |

Table 1

|  |  |
| --- | --- |
|  | Yes or No |
| Borrow | No |
| Overflow | Yes |

Table 2

1. As you know, when subtraction X – Y is performed or X and Y are compared, all the four NZVC flags are affected meaningfully. You obtained the following two tables in HW Assignment 1, which show the necessary and sufficient conditions for the 5 relationships between X and Y, namely =, <, ≤, >, and ≥.

|  |  |
| --- | --- |
| **Unsigned Comparison** | |
| Z = 1 | X = Y |
| B = 1 | X < Y |
| B + Z = 1 | X ≤ Y |
| B + Z = 0 | X > Y |
| B = 0 | X ≥ Y |

|  |  |
| --- | --- |
| **Signed Comparison** | |
| Z = 1 | X = Y |
| N XOR V = 1 | X < Y |
| N XOR V + Z = 1 | X ≤ Y |
| N XOR V + Z = 0 | X > Y |
| N XOR V = 0 | X ≥ Y |

Right after a “Compare X, Y” instruction (not in the HCS12 format!), let us assume that flag N = 1, and then a bgt instruction executes and happens to be successful. What can you tell about the V flag value right after the compare instruction?

The V-flag is 0.

1. Go over the following Data Segment:

; Data Segment

org $3000

pntr: dc.w arey

size: dc.b 5 ; array size

sum: ds.b 1 ; reserved for sum

flag: ds.b 1 ; reserved for error flag

org $3100

arey: dc.b 12, 45, 96, 20, 52, 86, 120, 4, 37

**Then fill in the blanks:**

; Code Segment:

clra ; A = 0

tfr A, Y ; Transfer “sign extended A” to Y, Y = 0

ldab size ; B = 5

aby ; (Y ← Y + B) Y = 5

ldx #arey ; X = 3100

ldy #pntr ; Y = 3000

ldx arey ; X = C2D

ldy pntr ; Y = 3100

clr flag ; which memory location is reset to 0? 3004

ldaa 4, Y ; A = 34

**Note:** Pay close attention to the *indentation* used in this code. Indentation along with *explanatory* and *short* comments make your code much more readable. You are expected to follow these rules in our Microcomputers I class.

\*\*\*

1. Use Boolean instructions to toggle the even bits of location $3000: (Note: The index of the LSb is 0, so even.)

ldaa $3000

eora #%01010101

staa $3000

1. Use bit manipulate instructions to reset the upper 4 bits of location $3500

bclr $3500, #$F0

1. Use bit manipulate instructions to set odd bits of the byte at address $3A00

bset $3A00, #%10101010

1. What is the largest value in location $30C0 that makes the following branch *successful*:

brclr $30C0, $24, loop

DB

Reason:

24 = 00100100

Largest successful branch = 11011011 = DB

1. What is the smallest value in location $30C0 that makes the following branch *unsuccessful*:

brclr $30C0, $24, loop

04

Reason:

24 = 00100100

Smallest unsuccessful branch = 00000100 = 4

1. Determine the contents of register X after the following instruction: (SP = $3010)

leax 8, -SP

3008

\*\*\*