Kettering University

Microcomputers I

Class Exercise Packet I

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!

You *probably* need to change your mindset!
The following could be one big difference between a high-school student and a college student:

1. Suppose that you are given an assignment that is based on instruction XXX. However, you do not know anything about XXX as it has not been covered in class yet.

Question: What will you do then? Highlight your answer:

I will not work on my assignment because XXX has not been covered yet in class!

I will first use the resources that I have to learn about XXX and then work on my assignment. As part of my report, I will let my professor know how I resolved the issue.

2. Look at the following pairs of numbers as *signed* numbers, and then highlight the *greater* one in each pair:

Greater (signed)	674C, 4E96	A7B8, 709D	C2A4, D590	6E24, 8C70

3. Look at the following pairs of numbers as *unsigned* numbers, and then highlight the *higher* one in each pair:

Higher (unsinged)	674C, 4E96	A7B8, 709D	C2A4, D590	6E24, 8C70

4. Look at the subtractions in the following table, and then put either a Yes or No for each subtraction:

	674C - 4E96	A7B8 - 709D	C2A4 - D590	6E24 - 8C70
Borrow	No	No	Yes	Yes

5. Subtract 674C – 4E96. Show your work in Table 1**.** Determine whether

- Borrow is generated. Type/write your answer in Table 2.
- Overflow is generated. Type/write your answer in Table 2.

1	0	0	1	1
	6	7	4	C
	B	1	6	9
1	1	8	**B**	**5**

Table 1

	Yes or No
Borrow	No
Overflow	Yes

Table 2

6. Subtract A7B8 – 709D. Show your work in Table 1**.** Determine whether

- Borrow is generated. Type/write your answer in Table 2.
- Overflow is generated. Type/write your answer in Table 2.

1	1	1	0	1
	A	7	B	8
	8	F	6	2
1	3	7	**1**	**B**

Table 1

	Yes or No
Borrow	No
Overflow	Yes

Table 2

7. Subtract C2A4 – D590. Show your work in Table 1**.** Determine whether

- Borrow is generated. Type/write your answer in Table 2.
- Overflow is generated. Type/write your answer in Table 2.

0	0	1	1	1
	C	2	A	4
	2	A	6	F
	E	D	**1**	**4**

Table 1

	Yes or No
Borrow	Yes
Overflow	No

Table 2

8. Subtract 6E24 – 8C70. Show your work in Table 1**.** Determine whether

- Borrow is generated. Type/write your answer in Table 2.
- Overflow is generated. Type/write your answer in Table 2.

0	1	0	1	1
	6	E	2	4
	7	3	8	F
	E	1	**B**	**4**

Table 1

	Yes or No
Borrow	Yes
Overflow	No

Table 2

9. What are the 3 main attributes of a memory location?

1. Every memory location has a unique address.
2. Every memory location has contents that can be changed and overwritten.
3. A memory location's contents can be interpreted to have several meanings.

10. What are the 3 possible meanings of the contents of a memory location?

1. Instruction
2. Data
3. Address

11. Pointers: What do you mean when you say register X *points* to the memory location at address 6000?

Register X stores a reference to memory location 6000.

12. How does the CPU know which instruction should be fetched and executed next?

The PC keeps track of it.

13. In the following assembly instruction for a hypothetical machine, where does the CPU read data (B and C) from? Where does the CPU store the result, A? List all the possibilities:

```
add   A, B, C       ; A <= B + C
```

A, B, and C could each be either a register or a memory location.

How does the CPU know where to read data (A and B) from? And where to store the result, A?

There are two dedicated registers named A and B.

14. Smith, who is currently taking Mircos I, wants to manually translate the assembly instruction "ldd **82AB**h" into HCS12 machine language. "The machine instruction is the binary equivalent for the associated assembly instruction", Smith says; and therefore he replaces characters l, d, and d (of the assembly operation ldd) with their ASCII equivalents, namely, 6C, 64, and 64, respectively, and gets the following machine code for the above assembly instruction: $6C 64 64 **82 AB**

Liz, who took this class a long time ago, disagrees and believes that (the ASCII codes for) uppercase letters should be used to generate binary equivalent for LDD; so she ends up with the following machine instruction for the above assembly instruction: $4C 44 44 **82 AB**

Question: Who is right? Briefly, clearly and legibly explain your reason:

Smith is right because the ASCII codes for lowercase letters are to be preferred in HCS12 machine code.

15. Execute: ldd #$2A75

A <= 2A
B<= 75
D<= 2A75

16. Assemble the following instructions:

ldab #$C6 ; $C6 C6

ldx #$2A75 ; $CE 2A 75

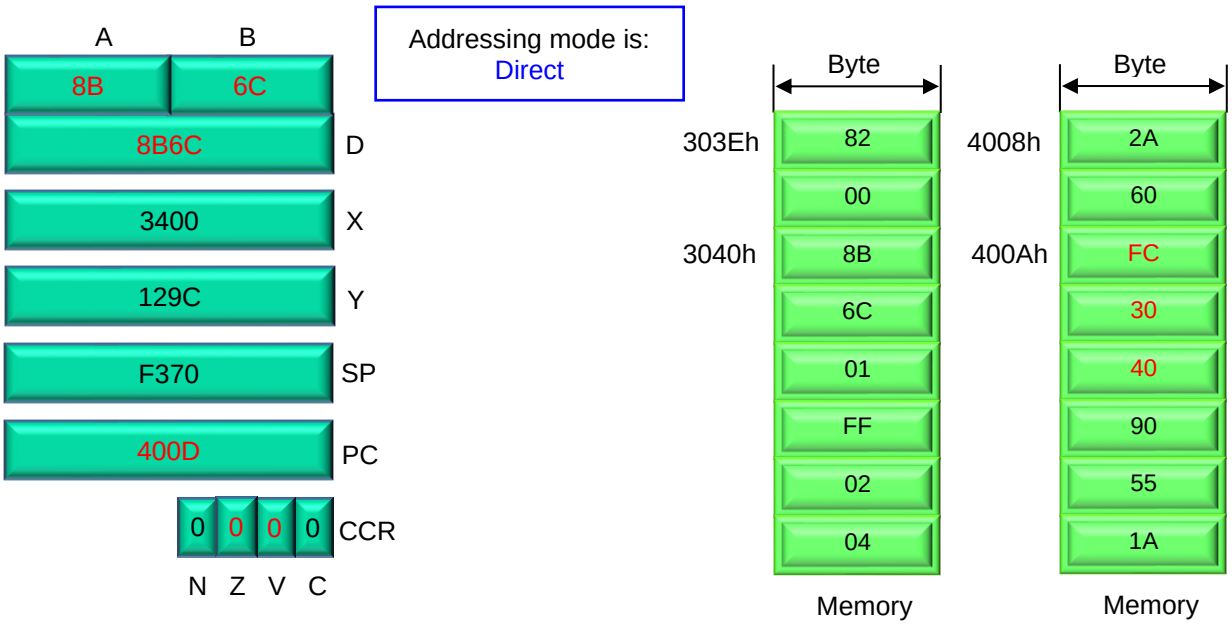
17. If you need to add two numbers, say 1B and D5, you may first use two load immediate instructions, ldaa #1B and ldab #D5, and then use an ADD instruction to add them up. (The ADD instruction adds Register A to Register B and places the sum in Register A. We will learn different types of ADD instruction and more soon.)

Question: Then, what is load EXTENDED for? Why do we need it?

In extended mode, you can load 1B and D5 in one instruction rather than having to use two.

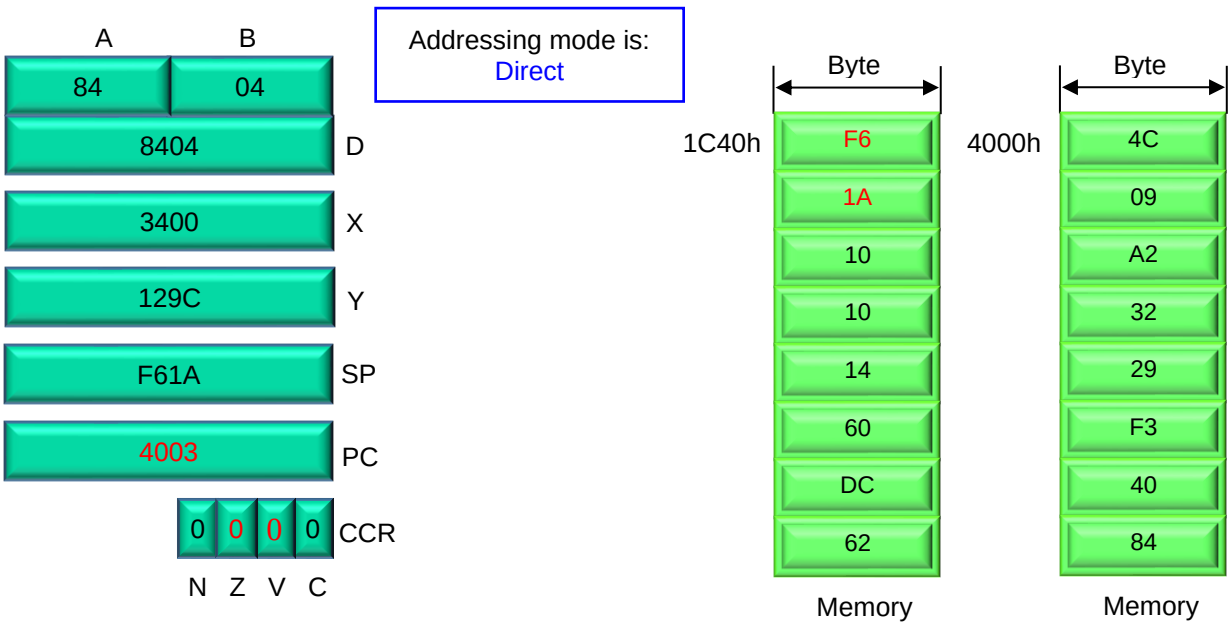
18. Assemble, place in Mem @400A, and execute. Show the changes in red.

```
ldd   $3040  ;
```



19. Assemble, place in Mem @4000, and execute. Show the changes in red.

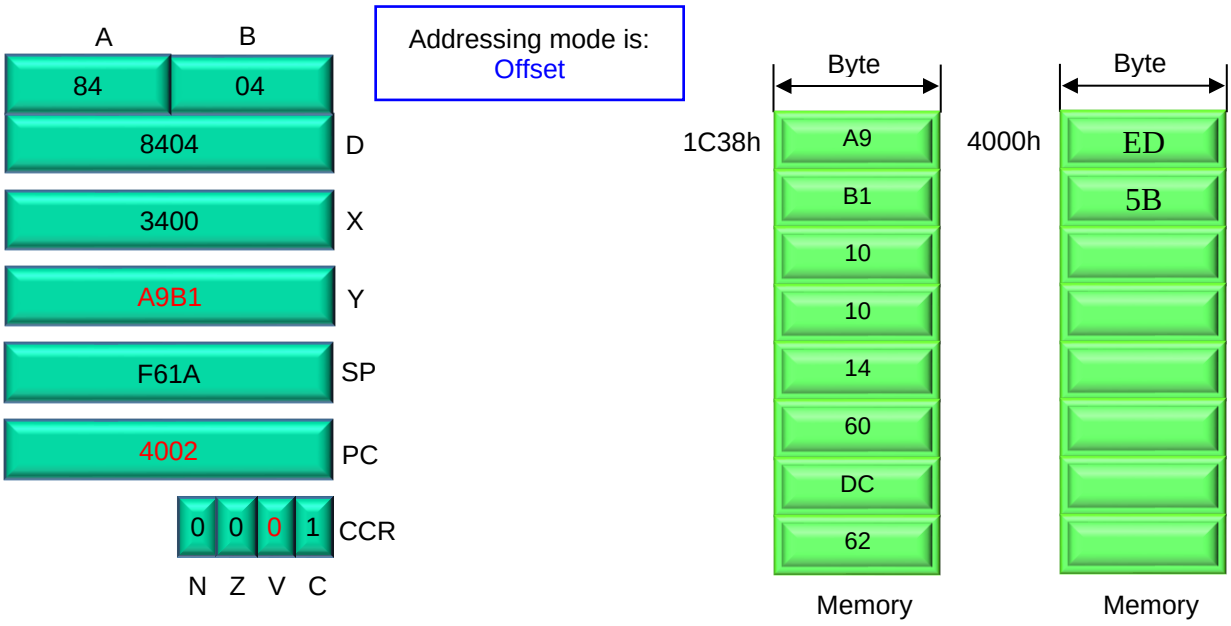
```
sts     $1C40    ;
```



20. Assemble **ldab -54, SP**

$E6 F1 CA

21. Assemble, place in Mem @4000, and execute ldy -5, Y. Show the changes in red



22. Assemble, place in Mem @4000, and execute stx 3, Y+ ;