

Homework 6

Due: March 1, 2025 11:30pm

Objective

To learn how to use arithmetic instructions and write basic system I/O subroutines. Learn user interface through a serial port.

Instruction

1. Write a user-friendly system program for the following commands:

S: Show the contents of memory location in word
W: Write the data word (not byte) to memory location
QUIT: Quit the main program, run 'Type writer' program.

2. **Command S:** This command shows the contents of memory location specified by the address in hexadecimal number followed by the 'S' character. For example, if the data \$126A is stored in memory location \$3000, a user types in the first line ending with Enter/Return key and the following should be displayed on the Terminal connected to the HCS12 board:

```
>S$3000
$3000 => %0001001001101010 $126A 4714
>
```

The data \$126A is printed in **binary**, **hexadecimal** and **decimal** number format. The character '>' is the prompt for this program.

3. **Command W:** this command writes data into the memory location specified by the address in hexadecimal followed by the 'W' character. The data to be written to the memory location is followed by a space and it can be specified by hexadecimal '\$' or just decimal number. For example, if one wants to store the data \$126A in memory location \$3003, a user types in the first line ending with Enter/Return key and the following should be displayed on the Terminal connected to the HCS12 board:

```
>W$3003 4714
$3003 => %0001001001101010 $126A 4714
>
```

Or

```
>W$3003 $126A
$3003 => %0001001001101010 $126A 4714
>
```

As a result, the word data \$126A is stored in the memory location \$3003 and it is shown with the 'S' command. The 'W' command accepts both decimal number '4714' or hexa-decimal number '\$126A' as a 16 bit number.

4. For HCS12 chip, a memory address is a 16 bit number and each memory location hold an 8 bit data. And we call a 16 bit data "word" and it is two byte.

5. Design the program to start at \$3100 and data to start at \$3000.
6. Make your program user-friendly and fool proofed. Print detail guide on the terminal screen so that users will properly use your program. Once your program is running, everything must be self-explanatory to user at the Terminal.
7. For this homework, you must do error checking to see if correct input is entered by a user. And give correct command usage example if invalid command was entered. Your program must NOT crash or hang if a user enters wrong input.
8. The Terminal display should look something like the following:

(after hit "run" button on simulator)

```
Welcome to the Simple Memory Access Program!
Enter one of the following commands (examples shown below)
and hit 'Enter'.
```

```
>S$3000                                ;to see the memory content at $3000 and $3001
> $3000 => %0001001001101010    $126A    4714
>

>W$3003 $126A                        ;to write $126A to memory locations $3003 and $3004
> $3003 => %0001001001101010    $126A    4714
>

>W$3003 4714                        ;to write $126A to memory location $3003 and $3004
> $3003 => %0001001001101010    $126A    4714
>

QUIT                                ;quit the Simple Memory Access Program
Type-writing now, hit any keys:

>S$30G0
> invalid input, address
>

>S$30123
> invalid input, address
>

>S345678
> invalid input, address
>

>W$3003 $126AB
> invalid input, data
>

>W$3003 70000
> invalid input, data
>

>W$3003
> invalid input, data
>

>W$3003 -106
> invalid input, data
>

>W$3 $120F
> $0003 => %0001001000001111    $120F    4623
>
```

```
>W$700 $9
> $0700 => %0000000000001001    $0009    9
>

>W$4000 12
> $4000 => %0000000000001100    $000C    12
>

>W$0001 005
> $0001 => %000000000000101    $0005    5
>
```

9. You may want to first draw the Flow Chart of the above algorithm.
10. Be sure to put much comments so that grader and others can clearly and quickly understand your program. Comments are very important in assembly language programs.
11. You may want to see and check the [Sample Grading Sheet](#) for this homework.
12. Copy your 'main.asm' file to 'cmpen472hw6_YourLastName.asm'. For example, mine will be 'cmpen472hw6_choi.asm' Then turn-in your .asm file (do NOT ZIP your file).
13. Turn-in your project source code file through [Penn State CANVAS](#). Upload your source code file into the CANVAS Assignment's Homework submission. Be sure to select CMPEN 472 class and correct Homework number, and with correct file name.

Congratulations on your sixth CMPEN 472 homework completion!

Epilogue:

Aid for the Homework 6. [Click here.](#)

More Aid for the Homework 6. [Click here.](#)
