CMPEN 472, The Pennsylvania State University

Homework 12

Due: April 25, 2025 11:30pm

Objective

To learn how to write a simple serial port based MONITOR program, a system program.

Instruction

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

   S: Show the content of memory location in word
   W: Write a data word (not byte) to memory location
   MD: Display the contents of continuous memory locations
   LD: Load a block of data to continuous memory locations
   GO: Run the program at the specified 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, and a user types in the first line below, ending with an Enter/Return key. Then the following should be displayed on the HyperTerminal connected to the HCS12 board:

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

   The data $126A is printed in **binary**, in **hexadecimal**, and in **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 a decimal number. For example, if one wants to store the data $126A in memory location $3003, a user types the following line ending with an Enter/Return key. Then the subsequent line should be displayed on the HyperTerminal 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 displayed with the 'S$3003' 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 16bit number and each memory location hold an 8bit data. And we call a 16 bit data "word" and it is two byte data.

5. **Command MD:** This command displays the contents of continuous memory locations. The **MD** command is followed by the 16bit address and 16bit data size, both numbers are in hexadecimal numbers separated by a space character. Below shows an example of this command. A user types in the first line ending with an Enter/Return key. The following should be displayed on the HyperTerminal connected to the HCS12 board:

   >MD$3000 $0120

   ```
   3000 FF 00 01 02 03 04 C1 D0 6A BC DE 1F 00 00 00 00
   3010 CF 31 00 86 FF 5A 02 5A 00 5A 03 5A 01 86 0C 5A
   3020 CB CC 00 01 5C C8 CE 30 05 16 31 79 16 31 98 CE
   3030 02 5A 00 5A 03 5A 01 86 0C 5A CB CC 00 01 5C C8
     *
     *
   3100 8E 00 28 25 11 CE 00 00 7E 30 03 96 01 88 01 5A
   3110 01 86 2A 16 31 88 30 32 3D 36 34 A6 30 81 00 27
   ```

   >

   An address and the 16 BYTE data per line is printed, all in **hexadecimal** number format. The character ' > ' is the prompt for this program.
   The data display always starts at the address boundary of the 16 byte data, such as $3010, $49F0, $FF00, etc. and the data display always ends at the address boundary of the 16 byte data, such as $300F, $456F, $FFFF, etc. (1) In case of the specified starting address that are not the boundary of the 16 byte data, display the memory data starting at the address that is the boundary of the 16 byte data which occurs before the specified starting address. (2) In case of the specified ending address that are not the boundary of the 16 byte data, display the memory data ending at the address that is the boundary of the 16 byte data which occurs after the specified ending address.

6. **Command LD:** This command loads a block of data to the continuous memory locations. The **LD** command is followed by (1) the 16bit memory block starting address, (2) the 16bit memory size (total number of bytes), and (3) the data that needs to be placed in the memory. They are separated by a space or an enter/return key character. Below shows an example of this command. A user types in the first line ending with an Enter/Return key. Subsequently a user types the data (or characters may be read from a file using 'Input File ...' option of the CodeWarrior Debugger Terminal app).

   >LD$3000 $0120

   ```
   FF0001020304C1D06ABCDE1F00000000
   CF310086FF5A025A005A035A01860C5A
   CBCC00015CC8CE3005163179163198CE
   025A005A035A01860C5ACBCC00015CC8
     *
     *
   8E00282511CE00007E3003960188015A
   01862A16318830323D3634A630810027
   ```

   >

   Each line of the data contains 16 bytes, all in **hexadecimal** number format. Each line is separated by an enter/return key character. You may assume that the data always starts at the address boundary of the 16 byte data, such as $3010, $49F0, $FF00, etc. and the data always ends at the address boundary of the 16 byte data, such as $300F, $456F, $FFFF, etc.

7. **Command GO:** This command run/execute the program at the specified memory location. The

GO command is followed by a 16bit address in hexadecimal number format. Below shows an example of this command. A user types in the first line ending with Enter/Return key.

```
>GO$3000
>
```

8. To test the **LD**, **MD**, and **GO** commands, do the following:

```
>LD$3E00 $0030

  43686F69CF3E00CE3E00A6300713A630
  070FA630070BA6300707A7A7A7A7A720
  F94FCC80FC5ACF3D0000000000000000

>
>MD$3E00 $0030

    *
    *
    *

>
>GO$3E04
>
```

   You will see 'Choi' printed on the terminal window.

9. Design your Homework 12 program to start at $3100 and data to start at $3000.

10. 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 HyperTerminal.

11. 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.

12. The HyperTerminal display should look like the following:

```
Welcome to the Simple MONITOR 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 => %0000000000000101    $0005    5
>
```

13. You may want to first draw the Flow Chart of the above algorithm.

14. Be sure to put many comments so that grader and others can clearly and quickly understand your program. Comments are very important in assembly language programs.

15. You may want to see and check the Sample Grading Sheet for this homework.

16. Copy your 'main.asm' file to 'cmpen472hw12_YourLastName.asm'. For example, mine will be 'cmpen472hw12_choi.asm' Then turn-in your .asm file (do NOT ZIP your file).

17. 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 the correct file name.

Congratulations on your twelveth CMPEN 472 homework completion!
