Experiment No. 01:

Introduction to the SANPER-1 Educational Lab Unit

Instructor: Dr. Jafar Saniie

ECE 441-001

Lab Date: 01-22-2020

Due Date: 01-29-2020

Acknowledgment: I acknowledge all of the work (including figures and codes) belongs to me and/or persons who are referenced.

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**I. Introduction**

**A. Purpose**

The purpose of this experiment is to introduce students to the following items: the SANPER-1 Educational Lab Unit, the TUTOR command set, the TUTOR TRAP 14 Handler, and the MC68000 instruction set. The students will also learn the basic functions of TUTOR.

**B. Background**

The SANPER-1 Educational Lab unit is a Motorola 6800 based computer designed and developed by Dr. J. Saniie and Alumnus Perich for educational purposes. This computer provides different ways of getting outputs. For example, the computer can be set in single-step mode where the user steps through each cycle of a program manually. The address line and raw data are outputted on the computer hardware itself. Here are the specifications of this computer: MC68000 microprocessor, 208K bytes of static RAM, 32K bytes of EPROM, three serial ports, a D/A converter, an A/D converter, a speech synthesizer, and two high power output drivers.

The SANPER-1 ELU contains the TUTOR Monitor program. This program allows users to display and change memory, modify internal registers, execute programs, debug programs, and interface to other types of equipment. It is also equipped with a 1-line assembler/disassembler, which allows the user to enter his or her source code directly from a terminal. Another useful feature of TUTOR is the TRAP 14 Handler. This allows the user programs to invoke system utility programs. Lastly, TUTOR has a feature called Trace mode which allows users to see the program run instruction by instruction.

In a nutshell, the TUTOR resident monitor is an educational tool which allows users to read and write in memory and on registers, assembling and disassembling instructions, and executing programs.

**II. Lab Procedure and Equipment List**

**A. Equipment**

*Equipment*

* SANPER-1 system
* PC with TUTOR software

**B. Procedure**

PART A

1. Invoke the HELP Command using “HE”.
2. Invoke the Display Formatted Registers Command using “DF”.
3. Display and Modify the Status Register.
4. Examine the status register using “DF”.
5. Set the status register to $FFFF.
6. Examine the status register using “DF”.
7. Modify address register A1 to $1234 using “.A1 1234”.
8. Examine the A1 register using the command “.A1”.
9. Display all address registers using the command “.A”.
10. Display all data registers using the command “.D”.

PART B

1. Use the “MM” command, enter in the unassembled source code of Table 1.1 into address $1000.
2. Using the “MS” command, store the string ‘IT WORKS !!’ into location $900.
3. Type in the command “G” to run the program.
4. Exit the program by pressing the ABORT button on the front panel of the ELU.
5. Record the output on the terminal that resulted from step 4.
6. Set the program counter to $1000 using “.PC $1000”. Then, use the trace command by typing in “T”.
7. Enter in the unassembled source code of Table 1.2 into TUTOR.
8. Using the SANPER-1 ELU hardware single step mode, step through a full iteration of the program. Record the contents of the LED displays of the SANPER unit.
9. Reset the SANPER unit and repeat the previous two steps for the source code of Tables 1.3 and 1.4.
10. With the SANPER in single step mode, reset the unit using the RESET button on the front panel.
11. Make sure the ELU front panel display reads $000000.
12. Step through 9 bus cycles, recording the contents of the LED displays after each cycle.

**III. Results and Analysis**

Sample Program #2

|  |  |
| --- | --- |
| **Figure 1** | **Figure 2** |
| **Figure 3** | **Figure 4** |
| **Figure 5** | **Figure 6** |
| **Figure 7** | **Figure 8** |

**Table 1: Single-Step Mode Results of Sample Program No. 2**

Sample Program #3

|  |  |
| --- | --- |
| **Figure 1** | **Figure 2** |
| **Figure 3** | **Figure 4** |
| **Figure 5** | **Figure 6** |
| **Figure 7** | **Figure 8** |

**Table 2: Single-Step Mode Results of Sample Program No. 3**Sample Program #4

|  |  |
| --- | --- |
| **Figure 1** | **Figure 2** |
| **Figure 3** | **Figure 4** |
| **Figure 5** | **Figure 6** |
| **Figure 7** | **Figure 8** |
| **Figure 9** |  |

**Table 3: Single-Step Mode Results of Sample Program No. 4**

Single Step Pulse

|  |  |
| --- | --- |
| **Figure 1** | **Figure 2** |
| **Figure 3** | **Figure 4** |
| **Figure 5** | **Figure 6** |
| **Figure 7** | **Figure 8** |

**Table 4: Results of Single Step Pulse**

1. **Discussion**

Programs

MM $1000;DI ; Program starts at $1000

LEA.L $2000, A7 ; Initialize A7 to value stored at 0x2000

MOVE.L #$900,A5 ; A5 points to start of string

MOVE.L #$90B, A6 ; A6 points to end of string

MOVE.B #243, D7 ; System call 243 - output

TRAP #14 ; Output the string from Port 1

MOVE.B #241, D7 ; Reads input into location pointed to by A6

TRAP #14 ; Input the string from Port 1

MOVE.B #227, D7 ; System call 243- output where a carriage return happens

TRAP #14 ; TRAP 14 Handler

BRA $1004 ; Branch to memory address $1004

. ; End of program

Figure 1: Sample Program No. 1

MM $900; DI ; Program starts at $900

MOVE.B D0, D1 ; Copy byte from D0 and put it into D1

MOVE.B #$AA, $1000 ; Enter the value 0xAAto location $1000

BRA $900 ; Branch to memory address $900

. ; End of program

Figure 2: Sample Program No. 2

MM $900; DI ; Program starts at $900

MOVE.B D0, D1 ; Copy byte from D0 and put it into D1

MOVE.B #$AA, $1001 ; Enter the value 0xAAto location $1001

BRA $900 ; Branch to memory address $900

. ; End of program

Figure 3: Sample Program No. 3

MM $900; DI ; Program starts at $900

MOVE.B D0, D1 ; Copy byte from D0 and put it into D1

MOVE.B $1000, $1001 ; Copy value from location $1000 and put it into $1001

BRA $900 ; Branch to memory address $900

. ; End of program

Figure 4: Sample Program No. 4

Questions

1. **Completed listings of all the program segments. Your listings must include meaningful global and local comments.**
2. **What is the address range of the available memory within the SANPER-1 ELU that may be used for your program? Describe your answer.**

The address range of user memory is 0x900 to 0x3FFF.

1. **For Procedures #7, #8 and #9 list the value of the address, data and control lines, and discuss any unusual events that may have occurred.**

Procedure #7

1. Address 0x900: Data 0x1200, AS, UDS, LDS, FC2, and FC1 lit.
2. Address 0x902: Data 0x11FC, AS, UDS, LDS, FC2, and FC1 lit.
3. Address 0x904: Data 0x00AA, AS, UDS, LDS, FC2, and FC1 lit.
4. Address 0x906: Data 0x1000, AS, UDS, LDS, FC2, and FC1 lit.
5. Address 0x908: Data 0x60F6, AS, UDS, LDS, FC2, and FC1 lit.
6. Address 0x1000: Data 0xAAF6, AS, UDS, FC2, and FC0 lit.
7. Address 0x90A: Data 0xFF2E, AS, UDS, LDS, FC2, and FC1 lit.
8. Loop

Procedure #8

1. Address 0x900: Data 0x1200, AS, UDS, LDS, FC2, and FC1 lit.
2. Address 0x902: Data 0x11FC, AS, UDS, LDS, FC2, and FC1 lit.
3. Address 0x904: Data 0x00AA, AS, UDS, LDS, FC2, and FC1 lit.
4. Address 0x906: Data 0x1001, AS, UDS, LDS, FC2, and FC1 lit.
5. Address 0x908: Data 0x60F6, AS, UDS, LDS, FC2, and FC1 lit.
6. Address 0x1001: Data 0x60AA, AS, UDS, FC2, and FC0 lit.
7. Address 0x90A: Data 0xFFFF, AS, UDS, LDS, FC2, and FC1 lit.
8. Loop

Procedure #9

1. Address 0x900: Data 0x1200, AS, UDS, LDS, FC2, and FC1 lit.
2. Address 0x902: Data 0x11F8, AS, UDS, LDS, FC2, and FC1 lit.
3. Address 0x904: Data 0x1000, AS, UDS, LDS, FC2, and FC1 lit.
4. Address 0x906: Data 0x1001, AS, UDS, LDS, FC2, and FC1 lit.
5. Address 0x1000: Data 0xAA01, AS, UDS, FC2, and FC0 lit.
6. Address 0x908: Data 0x60F6, AS, UDS, LDS, FC2, and FC1 lit.
7. Address 0x1001: Data 0x60AA, AS, LDS, FC2, and FC0 lit.
8. Address 0x90A: Data 0xFFFF, AS, UDS, LDS, FC2, and FC1 lit.
9. Loop
10. **How many serial ports are available on a SANPER-1 ELU? What are the specific memory address associated with these ports?**

There are 3 ACIAs that drive the three serial ports:

The first serial port is located at memory addresses 0x10040 and 0x10042. It communicates with a terminal.

The second serial port is located at memory addresses 0x10041 and 0x10043. It communicates with a host computer.

The third serial port is located at memory addresses 0x10044 and 0x10046. It does not have a specific use.

1. **List all the possible ways to interface your hardware design to the SANPER-1 ELU.**

External hardware can interface directly to the system bus using the buffered I/O. Also via the PIA for parallel interfaces, the ACIA for serial interfaces, the ADC for analog inputs and the DAC for analog outputs.

1. **When the MC68000 control signals “\*AS, \*UDS”, or “\*LDS” are asserted, what is the status of the respective LED (on or off), and why?**

According to the SANPER-1 manual, the LEDs are on when the signal is in the active or asserted state. Therefore, they will be on.

**IV. Conclusions**

This lab helped the users to be more familiar with the operation of the SANPER-1 ELU and the TUTOR firmware by means of successfully inputting, assembling and debugging several small programs. The users also learned the basic functions of TUTOR such as displaying and modifying registers. Lastly, this lab demonstrated how to use the hardware single step mode. This mode allows to watch the program execution on a per-bus-cycle basis with the state of the processor's address, data and control signals output to the SANPER-1's LED displays.

**References**

[1] Experiment 1 Lab Manual

[2] Educational Computer Board manual appendix

**Attachments**