

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ECE-441

LAB 1

Haomin SHI

September 6, 2017

Date Performed: 31/AUG/2017
Partners: Anh Nguyen
Instructor: Professor Saniie

1 Introduction

The main purpose of this lab is for the student to familiarize the equipment, which is the SANPER Educational Lab Unit and the TUTOR software (Courtesy of MOTOROLA®). This lab will help the student to understand the fundamentals about MC68000 instruction set, especially the functionality of the TRAP #14 instruction.

```
MOVE.B #<Function Number>, D7
TRAP #14
```

1.1 Background

The SANPER ELU: The SANPER ELU (Educational Lab Unit) is based on an MC68000 microprocessor made by MOTOROLA®. The SANPER ELU is developed by Dr. Saniie and Mr. Perich and the unit include multiple peripherals.

MC 68000: The MC 68000 is a 16/32-bit CISC microprocessor, which implements a 32-bit instruction set, with 32-bit registers and 32-bit internal data bus, but with a 16-bit main ALU and a 16-bit external data bus, designed and marketed by MOTOROLA®.

2 Lab Equipment and Procedure

2.1 Equipment

- SANPER ELU
- TUTOR software

2.2 Procedure

2.2.1 Part A

- (a) Connect SANPER unit
- (b) Command testing
 - HE <CR>
Help Command
 - DF <CR>
Display Formatted Registers Command
 - .SR 0000 <CR>
Modify the value of the Register (e.g: set to zero)
 - .A1 1234 <CR>
Changing the contents of A1 register to 1234 (\$00001234) or type in the command without the number to examine A1 register
 - .A <CR>
Display all address registers
 - .D <CR>
Display all data registers

2.2.2 Part B

- (a) Assemble program provided (Table 1.1)
- (b) Start the program from \$1000
- (c) Set the \$900 to output
- (d) Run the program
- (e) Notice problem
- (f) Use trace mode to check register changes
- (g) Repeat for programs in Table 1.2 - 1.4
- (h) Set SANPER-1 ELU to hardware single-step mode and reset it
- (i) Depress the SINGLE STEP PULSE and observe (take pictures)

3 Result and Analysis

The result will be showcased in the appendix section B of this lab report. The screenshot of the tutor terminal input will be included in the appendix A section.

3.1 Discussion

- (1) Terminal inputs(program segments) are in the appendix section of this lab report.
- (2) Based on the “mecb_chapter_7.pdf” document, the range of the available memory for user within the SANPER-1 ELU is \$0900 to \$7FFF.
- (3) The value of address lines will be listed in the appendix section. For this lab section there is no unusual event.
- (4) Based on the “mecb_chapter_7.pdf” document, there are two serial ports:
1 - ACIA1: \$10040 - \$10042 (even)
2 - ACIA2: \$10041 - \$10043 (odd)
- (5) Based on the “overview.pdf” interfaces are:
 - (a) Parallel Interface/Timer
 - (b) Peripheral Interface Adapter
 - (c) Asynchronous Communications Interface Adapter
 - (d) Serial Port
- (6) “*AS”, “*UDS” and “*LDS” means:
 - (a) “*AS” means Address Strobes, it is a three-state signal indicates that the information on the address bus is a valid address.
 - (b) “*UDS” means Upper Data Strobes
 - (c) “*LDS” means Lower Data Strobes

4 Conclusions

The lab section is successful. The student was exposed to the SANPER unit and TUTOR program, meanwhile, the student had the chance to put their hands on the system by using The TRAP #14 call.

5 Appendices

Codes and Terminal Inputs

5.1 Appendix A

5.1.1 Appendix A.1: Original codes with comments

Table 1.1 - Sample Program No.1

```
LEA.L $2000, A7 ;load mem
MOVE.L #$900 ,A5 ;set register with address, this line is at $1004
MOVE.L #$90B ,A6
MOVE.B #243, D7 ;sys call out port 1
TRAP #14
MOVE.B #241, D7
TRAP #14
MOVE.B #227, D7
TRAP #14
BRA $1004 ;back to start
. ;fin
```

Table 1.2 - Sample Program No.2

```
MOVE.B D0, D1 ;d0 to d1 copy, this line is at $900
MOVE.B #$AA, $1000 ;move to a mem address
BRA $900 ;back to start
. ;end
```

Table 1.3 - Sample Program No.3

```
MOVE.B D0, D1 ;copying, this line is at 900
MOVE.B #$AA, $1000 ;move to a mem address
BRA $900 ;loop back to start
. ;end
```

Table 1.4 - Sample Program No.4

```
MOVE.B D0, D1 ;copying, @mem $900
MOVE.B $1000, $1001 ;move mem address
BRA $900 ;loop
. ;end
```

5.1.2 Actuarial Input

```

TUTOR 1.3 > HE
.PC .SR .US .SS
.D0 .D1 .D2 .D3 .D4 .D5 .D6 .D7
.A0 .A1 .A2 .A3 .A4 .A5 .A6 .A7
.R0 .R1 .R2 .R3 .R4 .R5 .R6

BF  BM  BR  NOBR BS  BT  DC  DF
DU  G  GD  GO  GT  HE  LO  M
MD  MM  MS  OF  PA  NOPA PF  T
TM  TR  TT  VE

TUTOR 1.3 > DF
PC=00000000 SR=2700=.S7..... US=FFFF7F5F SS=00000786
D0=FFFF7FFF D1=FFFF3FD7 D2=FFFF7FFF D3=FFFF78FB
D4=FFFF7FFF D5=FFFF7E9B D6=FFFF7FFF D7=FFFF7FCF
A0=FFFFFAF A1=FBFF38FB A2=FFFF7BFF A3=FFFF32FB
A4=FFFF3FFF A5=7FFF7FDD A6=FFFF7FFF A7=00000786
-----000000 0F85 BCLR D7,D5

TUTOR 1.3 > .SR 0000

TUTOR 1.3 > DF
PC=00000000 SR=0000=.0..... US=FFFF7F5F SS=00000786
D0=FFFF7FFF D1=FFFF3FD7 D2=FFFF7FFF D3=FFFF78FB
D4=FFFF7FFF D5=FFFF7E9B D6=FFFF7FFF D7=FFFF7FCF
A0=FFFFFAF A1=FBFF38FB A2=FFFF7BFF A3=FFFF32FB
A4=FFFF3FFF A5=7FFF7FDD A6=FFFF7FFF A7=FFFF7F5F
-----000000 0F85 BCLR D7,D5

TUTOR 1.3 > .SR FFFF

TUTOR 1.3 > DF
PC=00000000 SR=FFFF=TS7XNZVC US=FFFF7F5F SS=00000786
D0=FFFF7FFF D1=FFFF3FD7 D2=FFFF7FFF D3=FFFF78FB
D4=FFFF7FFF D5=FFFF7E9B D6=FFFF7FFF D7=FFFF7FCF
A0=FFFFFAF A1=FBFF38FB A2=FFFF7BFF A3=FFFF32FB
A4=FFFF3FFF A5=7FFF7FDD A6=FFFF7FFF A7=00000786
-----000000 0F85 BCLR D7,D5

TUTOR 1.3 > .A1 1234

TUTOR 1.3 > .AI
WHAT

TUTOR 1.3 > .A1
.A1=00001234

TUTOR 1.3 > .A
A0=FFFFFAF A1=00001234 A2=FFFF7BFF A3=FFFF32FB
A4=FFFF3FFF A5=7FFF7FDD A6=FFFF7FFF A7=00000786

TUTOR 1.3 > .D
D0=FFFF7FFF D1=FFFF3FD7 D2=FFFF7FFF D3=FFFF78FB
D4=FFFF7FFF D5=FFFF7E9B D6=FFFF7FFF D7=FFFF7FCF

TUTOR 1.3 > █

```

Figure 1: Part A.1 input

```

mm $900;DI
WHAT

TUTOR 1.3 > MM $900;DI
000900 3200 MOVE.W D0,D1
000902 5347 SUBQ #1,D7 ?
000904 0F0B08F MOVEP.W $0B8F(A3),D7 ?
000908 .X?
000908 862F0F4C OR.B 3916(A7),D3 ?

TUTOR 1.3 > CLEAR
WHAT

TUTOR 1.3 > CLR
WHAT

TUTOR 1.3 >
he
WHAT

```

Figure 2: Part A.2 input

```

MM $1000;DI
001000 4FF82000 LEA $2000,A7
001004 2A7C0000900 MOVE.L #$900,A5
00100A
00100A 2C7C0000090B MOVE.L #$90B,A6
001010 1E3C00F3 MOVE.B #243,D7
001014 4E4E TRAP #14
001016 1E3C00F1 MOVE.B #241,D7
00101A RAP
00101A 4E4E TRAP #14
00101C
00101C 1E3C00E3 MOVE.B #227,D7
001020 4E4E TRAP #14
001022 60E0 BRA $1004
001024 F269 DC.W $F269 ?

TUTOR 1.3 > MS $900 'IT WORKS !!'

TUTOR 1.3 > G $1000
PHYSICAL ADDRESS=00001000
IT WORKS !!

SOFTWARE ABORT
PC=00009FA8 SR=2710=,S7X... US=FFFF7F5F SS=00001FD4
D0=FFFF7FFF D1=FFFF3F02 D2=FFFF7FFF D3=00000000
D4=FFFF7FFF D5=FFFF7E9B D6=FFFF7FFF D7=00000000
A0=00010040 A1=00001234 A2=FFFF7BFF A3=FFFF32FB
A4=FFFF3FFF A5=0000090B A6=0000090B A7=00001FD4
-----009FA8 02010010 AND.B #16,D1

TUTOR 1.3 > .PC $1000

TUTOR 1.3 > T
PHYSICAL ADDRESS=00001000
PC=00010004 SR=2710=,S7X... US=FFFF7F5F SS=00002000
D0=FFFF7FFF D1=FFFF3F02 D2=FFFF7FFF D3=00000000
D4=FFFF7FFF D5=FFFF7E9B D6=FFFF7FFF D7=00000000
A0=00010040 A1=00001234 A2=FFFF7BFF A3=FFFF32FB
A4=FFFF3FFF A5=0000090B A6=0000090B A7=00002000
-----001004 2A7C00000900 MOVE.L #2304,A5

TUTOR 1.3 :>
PHYSICAL ADDRESS=00001004
PC=0000100A SR=2710=,S7X... US=FFFF7F5F SS=00002000
D0=FFFF7FFF D1=FFFF3F02 D2=FFFF7FFF D3=00000000
D4=FFFF7FFF D5=FFFF7E9B D6=FFFF7FFF D7=00000000
A0=00010040 A1=00001234 A2=FFFF7BFF A3=FFFF32FB
A4=FFFF3FFF A5=00000900 A6=0000090B A7=00002000
-----00100A 2C7C0000090B MOVE.L #2315,A6

TUTOR 1.3 :>
PHYSICAL ADDRESS=0000100A
PC=00001010 SR=2710=,S7X... US=FFFF7F5F SS=00002000
D0=FFFF7FFF D1=FFFF3F02 D2=FFFF7FFF D3=00000000
D4=FFFF7FFF D5=FFFF7E9B D6=FFFF7FFF D7=00000000
A0=00010040 A1=00001234 A2=FFFF7BFF A3=FFFF32FB
A4=FFFF3FFF A5=00000900 A6=0000090B A7=00002000
-----001010 1E3C00F3 MOVE.B #243,D7

TUTOR 1.3 :> █

```

Figure 3: Part B input

```

TUTOR 1.3 > MM $900;DI
WHAT

TUTOR 1.3 > MM $900;DI
000900 1200 MOVE.B D0,D1
000902 MOVE.B X?
000902 11FC00AA1000 MOVE.B #$AA,$1000
000908 X?.

TUTOR 1.3 > MM $7C;L
00007C 0000991E ? 900
000080 00008442 ?

TUTOR 1.3 > █

```

Figure 4: Part 1.2 input

```

WHAT
TUTOR 1.3 > MM $900;DI
000900 1200
000902 11F810001001
000908 60F6
00090A 23CC0F0E2F07
MOVE.B D0,D1
MOVE.B $1000,$1001
BRA $900
MOVE.L A4,$0F0E2F07 ?

TUTOR 1.3 > MM $7C;L
00007C 0000991E ? 900
000080 00008442 ?

TUTOR 1.3 > █

```

Figure 5: Part 1.3 input

```

p
TUTOR 1.3 >
WHAT
TUTOR 1.3 >
WHAT
TUTOR 1.3 > MM $900;DI
000900 1200
000902 11FC00AA1001
000908 60F6
00090A 23CC0F0E2F07
MOVE.B D0,D1
MOVE.B #AA,$1001
BRA $900
MOVE.L A4,$0F0E2F07 ?

TUTOR 1.3 > MM $7C;L
00007C 0000991E ? 900
000080 00008442 ?

TUTOR 1.3 > █

```

Figure 6: Part 1.4 input

5.2 Appendix B