UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT PAPER AND COPYRIGHT

Author's full name : AIMI RUZAINI BINTI AHMAD

Date of birth : 8TH JUNE 1986

Title : LAB MANUAL FOR THE LOGIC ANALYSER : 68000 MICROPROCESSOR AS THE TEST HARDWARE

Academic Session: 2008/2009

I declare that this thesis is classified as :

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1972)*
☐	RESTRICTED	(Contains restricted information as specified by the organisation where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (full text)

I acknowledged that Universiti Teknologi Malaysia reserves the right as follows :

1. The thesis is the property of Universiti Teknologi Malaysia.
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by :

aimi
SIGNATURE

860608-29-6074
(NEW IC NO. /PASSPORT NO.)

Date :

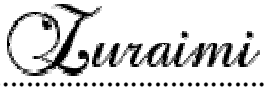
Zuraimi
SIGNATURE OF SUPERVISOR

EN. ZURAIMI BIN YAHYA
NAME OF SUPERVISOR

Date :

NOTES : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organisation with period and reasons for confidentiality or restriction.

"I hereby declare that I have read this report and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of
Bachelor of Engineering (Computer)"

Signature :  .....

Name of Supervisor : En. Zuraimi bin Yahya

Date : 15 May 2009

LAB MANUAL FOR THE LOGIC ANALYSER : 68000 MICROPROCESSOR AS THE TEST HARDWARE

AIMI RUZAINI BINTI AHMAD

A report submitted partial fulfilment of the
requirement for the award of the degree of
Bachelor of Engineering (Computer)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

MAY, 2009

I declare that this report entitled *Lab Manual for the Logic Analyser : 68000 Microprocessor is the Test Hardware* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : aimi.....

Name : Aimi Ruzaini binti Ahmad

Date : 15 May2009

Dedicated to my lovely family

My dad, Ahmad bin Din,

My lovely mum, Rahimah binti Mat Yassim,

My inspiration elder brother, Aizuddin Akmal bin Ahmad,
My spirit brother, Anwar Ashraf bin Ahmad,
My dearest sister, Anis Ruwaida binti Ahmad,
My pamper sister, Aini Razanah binti Ahmad

To all my friends

Unforgettable, those who are assisting and taught me in developing this project, En Zuraimi bin Yahya, my supervisor. Truly, your assistance, En. Zainul Abidin bin Halim is valuable and I will never forget it.

This Project Report is dedicated to all those who have supported my educational growth over the years. Without you all, I would not have had the inner strength to persevere through the fact that to be implemented in future.

Thank you.

ACKNOWLEGEMENTS

Thanks to Allah, with His strength and bless I could complete my Final year Project (FYP) successfully. I would like to thank my parents for providing me with the opportunity to gain this tremendous education and the necessary tools to succeed in life.

My supervisor, En. Zuraimi bin Yahya that assisting, teaching, and encouraging me to make sure the project done according to the schedule and successfully. All the sacrifice, effort, time, and ideas contributed by all of you, may Allah SWT rewarding it. I would also like to thank to En. Zainul Abidin, technition from Microprocessor Lab which allow me to use the PC and instruments there.

Finally, I would like to express my sincere thanks to all the lecturer and staffs in Faculty of Electrical Engineering, UTM, my fellow course mates, friends and those who involved directly and indirectly in this project for all their helps, encourage and advice.

Thank you very much

ABSTRACT

The aim of this project is to build a lab manual on how to use the Logic Analyser by using a 68000 microprocessor board as test hardware. The Logic Analyser can be used to monitor real time hardware operation of 68000 microprocessor chip. The lab manual for the Logic Analyser is intended to assist student to learn how to use the equipment and understand the operation of the software loaded on the 68000 microprocessor. Student also will be able to view the real operation of the software and hardware through the multiple digital signals output. The logic of listing file created by assembling the source program can be visualized on the display of the Logic Analyser. Trigger function can be used as a breakpoint for Logic Analyser to start capturing data on the 68000 board and helps student to debug and understand the bus cycle operation of a 68000 microprocessor.

ABSTRAK

Tujuan projek dijalankan adalah untuk menyediakan satu manual penggunaan makmal mengenai bagaimana menggunakan Logic Analyser dan litar mikroprosessor 68000 adalah sebagai litar uji. Logic Analyser boleh digunakan untuk meneliti secara langsung operasi pada cip mikroprosessor 68000. Dengan tersedianya manual penggunaan makmal mengenai Logic Analyser ini, ia dapat membantu pelajar untuk belajar bagaimana menggunakan peralatan dan memahami operasi yang terdapat di dalam perisian mikroprosessor 68000. Pelajar juga akan dapat melihat operasi sebenar dalam perisian dan perkakasan melalui keluaran pelbagai isyarat digital. Logik yang terdapat pada 'listing file' yang diwujudkan daripada pembinaan perisian sumber boleh dilihat di paparan Logic Analyser. Fungsi 'trigger' boleh digunakan sebagai 'breakpoint' pada Logic Analyser untuk memulakan menangkapan data pada kotak mikroprosessor 68000 dan ia dapat membantu pelajar untuk 'debug' dan faham tentang putaran bus melalui operasi mikroprosessor 68000.

TABLE OF CONTENTS

CHAPTER	TITLE		PAGE
	DECLARATION		**ii**
	DEDICATION		**iii**
	ACKNOWLEGEMENTS		**iv**
	ABSTRACT		**v**
	ABSTRAK		**vi**
	TABLE OF CONTENTS		**vii**
	LIST OF TABLES		**xi**
	LIST OF FIGURES		**xii**
	LIST OF ABBREVIATIONS		**xv**
	LIST OF APPENDICES		**xvi**
1	**INTRODUCTION**		
	1.1	Problems Statement	2
	1.2	Objectives of the Project	4
	1.3	Scope of Work	4
	1.4	Project Planning	5
2	**LITERATURE REVIEW**		
	2.1	Logic Analyser	8
	2.2	Disassembler POD for 68000	9
	2.3	Software 68000 microprocessor	10

2.3.1 EASy68K Software 10
2.3.2 Command Prompt 13
2.4 Test Hardware 15
2.4.1 68000 Microprocessor Board 16
2.4.2 Existing Application Board 16
2.4.3 EPROM 27C256 17
2.5 Chip Max Programmer 18
2.6 EPROM Eraser 19

3 METHODOLOGY

3.1 Part 1 of the Laboratory Manual 21
3.2 Part 2 of the Laboratory Manual 21
3.3 Part 3 of the Laboratory Manual 22
3.3.1 Software for Part 3 of the Laboratory Manual 22
3.3.2 Hardware for Part 3 of the Laboratory Manual 23
3.3.3 Analysis for Part 3 of the Laboratory Manual 28
3.4 Part 4 of the Laboratory Manual 28
3.4.1 Software for Part 4 of the Laboratory Manual 29
3.4.2 Hardware for Part 4 of the Laboratory Manual 29
3.4.3 Analysis for Part 4 of the Laboratory Manual 30

4 RESULT ANALYSIS

4.1 Part 1 of the Laboratory Manual 32
4.1.1 Instructions from the Laboratory Manual for Part 32
4.2 Part 2 of the Laboratory Manual 34
4.2.1 Instructions from the Laboratory Manual for Part 2 34

4.2.2 Question and Answer for Part 2 of Laboratory Manual (SECTION A) 36

4.2.3 Data from the Listing File for Part 2 of Laboratory Manual 37

4.3 Part 3 of the Laboratory Manual 37

4.3.1 Instructions from the Laboratory Manual for Part 3 38

4.3.2 Question and Answer for Part 3 of Laboratory Manual (SECTION B) 39

4.3.3 Data from the Listing File for Part 3 of Laboratory Manual 41

4.4 Part 4 of the Laboratory Manual 42

4.4.1 Instructions from the Laboratory Manual for Part 4 42

4.4.2 Question and Answer for Part 4 of Laboratory Manual (SECTION C) 44

4.4.3 Aquired Data from the Logic Analyser 45

4.4.4 The Program for Part 4 of the Laboratory Manual 46

5 DISCUSSION

5.1 Part 1 of the Laboratory Manual 47

5.2 Part 2 of the Laboratory Manual 48

5.3 Part 3 of the Laboratory Manual 48

5.4 Part 4 of the Laboratory Manual 49

6 SUGGESTIONS AND CONCLUSION

6.1 Recommendation on Future Works 50

6.2 Conclusion 51

REFERENCES **52**

Appendix A 53

Appendix B 61

Appendix C 72

Appendix D 73

LIST OF TABLES

TABLE NO.	TITLE	PAGE
1.1	Gantt chart for FYP 1	5
1.2	Gantt chart for FYP 2	6

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
Figure 1.1	The Test Hardware (Application Board and 68000 Microprocessor Board)	3
Figure 2.1	Logic Analyser	8
Figure 2.2	Disassembler POD for 68000	9
Figure 2.3	Windows of EASy68K Software	10
Figure 2.4	Editor of EASy68K Software	11
Figure 2.5	Write the Program using EASy68K Software	11
Figure 2.6	Window of EASy68K Software	12
Figure 2.7	Listing File from EASy68K Software	12
Figure 2.8	Write the Program Using Command Prompt	13
Figure 2.9	Assemble the Program Using Command Prompt	14
Figure 2.10	Create the Listing File and .BIN file	14
Figure 2.11	Split the .BIN file to .EVN and .ODD files Using Command Prompt	15
Figure 2.12	Application Board (Left Side) and 68000 Microprocessor Board (Right Side)	15
Figure 2.13	Connection between 68000 microprocessor and EPROM 27C64 (Example)	16
Figure 2.14	EPROM 27C256	17

Figure 2.15	Chip Max Programmer	18
Figure 2.16	EPROM eraser	19
Figure 3.1	Flow Chart for the Part 3	22
Figure 3.2	Hyperterminal (Connection Description)	23
Figure 3.3	Hyperterminal (Connect To)	24
Figure 3.4	COM1 Properties	24
Figure 3.5	After press ENTER (3 times)	25
Figure 3.6	Type ‘lt’	25
Figure 3.7	Load data .BIN to chip using Hyperterminal	26
Figure 3.8	Load Completed	26
Figure 3.9	Type Address for Microprocessor	27
Figure 3.10	Connections between 68000 Disassembler POD and 68000 microprocessor chip	27
Figure 3.11	Flow Chart for the Part 4	28
Figure 3.12	Replace the EPROM from the 68000 microprocessor board	30
Figure 4.1 (a)	Instructions for the Part 1 of the Laboratory Manual	32
Figure 4.1 (b)	Instructions for the Part 1 of the Laboratory Manual	33
Figure 4.2 (a)	Instructions for the Part 2 of the Laboratory Manual	34
Figure 4.2 (b)	Instructions for the Part 2 of the Laboratory Manual	35
Figure 4.3 (a)	Question and Answers for the Part 2 of the Laboratory Manual	36
Figure 4.3 (b)	Question and Answers for the Part 2 of the Laboratory Manual	37
Figure 4.4 (a)	Instructions for the Part 3 of the Laboratory Manual	38
Figure 4.4 (b)	Instructions for the Part 3 of the Laboratory Manual	39
Figure 4.5 (a)	Question and Answers for the Part 3 of the Laboratory Manual	39

Figure 4.5 (b)	Question and Answers for the Part 3 of the Laboratory Manual	40
Figure 4.5 (c)	Question and Answers for the Part 3 of the Laboratory Manual	41
Figure 4.6	Listing File for part 3 of the Program	42
Figure 4.7	Instructions for the Part 4 of the Laboratory Manual	43
Figure 4.8 (a)	Question and Answers for the Part 4 of the Laboratory Manual	44
Figure 4.8 (b)	Question and Answers for the Part 4 of the Laboratory Manual	45
Figure 4.9	The Program That Shown on a Single Screen	45
Figure 4.10	The Program for Part 4 of the Laboratory Manual	46
Figure 5.1	The program that be created before programmed the EPROM	49

LIST OF ABBREVIATIONS

EEPROM	Electrically Erasable Programmable Read-Only Memory
EPROM	Erasable Programmable Read-Only Memory
FKE	Faculty of Electrical Engineering
FYP	Final Year Project
IDE	Integrated Drive Electronics
LED	Light Emitted Diode
UTM	Universiti Teknologi Malaysia
UV	Ultra-Violet

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Program for Part 3 of the Laboratory Manual	53
B	Listing File for Part 3 of the Laboratory Manual	61
C	Program for Part 4 of the Laboratory Manual	72
D	Lab Manual for the Logic Analyser : 68000 Microprocessor Board as the Test Hardware	73

CHAPTER 1

INTRODUCTION

First chapter will explain about the problems statement, objectives, scope of work and project planning for Final Year Project 1 (FYP 1) and Final Year Project 2 (FYP 2). This project executed based on the problem happened directly or indirectly. Objective of the project is needed to ensure the target in this project will be achieved. To make sure the project is on the track towards achievable objective, scope of work is important. Gantt chart is used as project planning to get the overall view of project completion milestone beside as a guideline of the process.

As an undergraduate student, researches could help most of student daily schedule. From the research, all of the theories applied could easily remember. Before run the research, the laboratory manual is needed as a guide line. If without the laboratory manual, the research will take quite long time to search all the other references. In assisting and helping students to be easier, the laboratory manual for the Logic Analyser will be provides.

1.1 Problems Statement

The Logic Analyser is the instrument that never being used before in the Faculty of Electrical Engineering (FKE), University Teknologi Malaysia (UTM). This instrument is important to know because today it is been used in most industries. Also, it is an advantage if we know to use it in proper. Among of few advantages of this instrument are could trace the data that have on the chip by using reverse engineering method and also it could display the data that cannot done by some software. The data that will display is very detail.

68000 microprocessor board is the test hardware for this project. The application devices that use on this project are LEDs and switches. It is normal applications are connected to a microprocessor board as an output. The application board for LEDs and switches is already existed on the laboratory. The application board should connect by IDE cable to the 68000 microprocessor board for LED and switch to be functional. Figure 1.1 shows that the test hardware that will be use on this project. The test hardware will be connected to the Logic Analyser. Using Logic Analyser, the real time hardware operation of microprocessor chip can be monitor.

Most students couldn't understand the data that was displayed from the listing file of the software. And sometimes, the student misreading the data that given in the listing file. The Logic Analyser can be solving that problem. Student will be able to view the real operation of the software and hardware through the multiple digital signals output. The data that will be display is one by one on the memory location. So, it is easier to the student to read the data that existed in the program compared with listing file.

Figure 1.1 The Test Hardware
(Application Board and 68000 Microprocessor Board)

1.2 Objectives of the Project

By this project, it will solve the problem that faced by most of the students. Therefore, the objectives to overcome all of these are:

1. To assist student to learn how to use the equipment and understand the operation of the software loaded on the 68000 microprocessor
2. To develop laboratory manual for the Logic Analyser using the 68000 microprocessor board as the test hardware.

1.3 Scope of Work

This project done base on the scope of guide line that been fixed as follows:

1. Explore function of the Logic Analyser and refer the information about Logic Analyser on the internet.
2. Compare the data of the program between the listing file and the data on Logic Analyser.
3. Trace the data on the 68000 microprocessor chip without software (reverse engineering method).
4. Build a laboratory manual which allow students to understand the function of the Logic Analyser.

1.4 Project Planning

Table 1.1 Gantt chart for FYP 1

TASK SCHEDULE	PSM 1												
	W 3	**W 4**	**W 5**	**W 6**	**W 7**	**W 8**	**W 9**	**W 10**	**W 11**	**W 12**	**W 13**	**W 14**	**W 15**
Meeting with Coordinator													
Meet The Supervisor													
Get & Confirm the Title													
Analyze the title													
Check the manual of the Logic Analyser													
Study the software													
Study the hardware													
Preparation for presentation													
Presentation													
Report Writing													

Table 1.2 Gantt chart for FYP 2

TASK SCHEDULE	PSM 2																
	W1	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**	**W8**	**W9**	**W10**	**W11**	**W12**	**W13**	**W14**	**W15**	**W16**	**W17**
Built a program																	
Built the hardware																	
Determine the connection of the HC11 system to the Logic Analyser																	
Built a laboratory manual																	
Submit project summary																	
Preparation for Presentation and Demo																	
Presentation and Demo																	
Writing Thesis																	

CHAPTER 2

LITERATURE REVIEW

As we know in statement problem has been explained that the Logic Analyser is never being used in the FKE's laboratory, UTM. For solving this problem, the laboratory manual is provided to the students as their guide and reference on how to use the Logic Analyser easily. The analysis is using the Logic Analyser will be running after the program was uploaded on the 68000 microprocessor chip. The existing board for the application device will be used to test the program is running properly or not. The connection between application board and 68000 microprocessor board will be connecting using IDE cable. This chapter will explained in details about the instruments and devices that will be used in this project.

2.1 Logic Analyser

The Logic analyser is an electronic instrument that could display multiple digital signals on a single screen. Students can easily analyse the operation of a digital signal, which cannot be done using an Oscilloscope. The Logic Analyzer can trigger on a complicated sequence of digital events, and then capturing a large amount of digital data. These probes provide a durable, reliable mechanical and electrical connection between the probe and the circuit board with less than 0.5pF to 0.7 pF loading per signal. Once the probes are connected, the student programs of the analyzer with the names of each signal, and can grouping several signals into groups for easier manipulation.

Figure 2.1 Logic Analyser

2.2 Disassembler POD for 68000

The most common method of data capture for logic analyzers is through a probe. The Logic Analyzer can measure anything electronic if it has the proper probe connected. Mostly the Logic Analyzer measure data buses from the ports. The probes try to tap into the electronic signals being passed through a data bus or wire.

Figure 2.2 Disassembler POD for 68000

2.3 Software 68000 microprocessor

Now a day, so many software that can be used for 68000 microprocessor such as EASy68K, ide68K and so on. Another way for make the program is use Command Prompt. For this project, EASy68K software and Command Prompt will be used for make and assemble the program.

2.3.1 EASy68K Software

Figure 2.3 shows the windows of EASy68K software. The program of EASy68K is simple and easy to understand. To write the program in EASy68K, we use the editor as shows in Figure 2.4. As shown in Figure 2.4, that is the template to write program. This software can be assembling the program to know the error of the program as shown in Figure 2.6 and the full program is in Appendix A. After assemble the program, the listing file will be create. The example of listing file was shown in Figure 2.7 and the full listing file is in Appendix B.

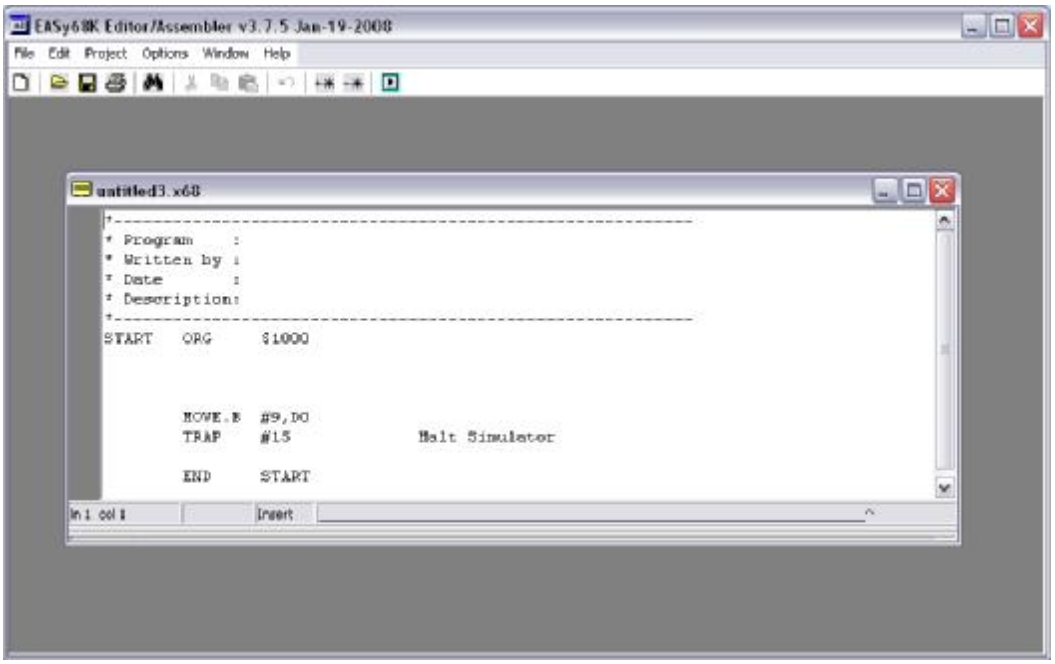


Figure 2.3 Windows of EASy68K software

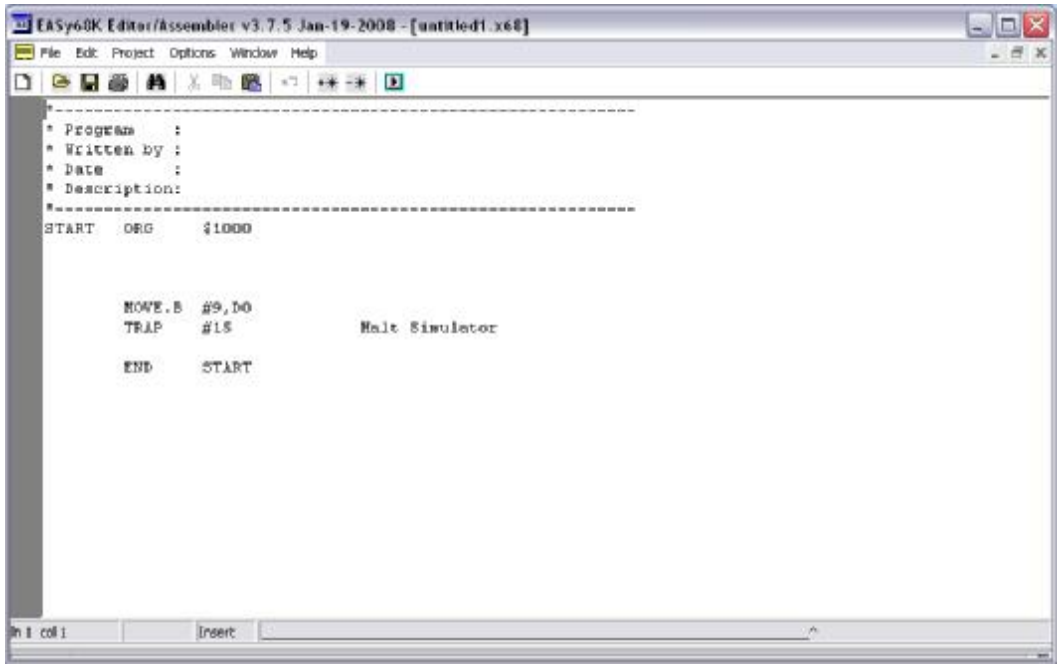


Figure 2.4 Editor of EASy68K software

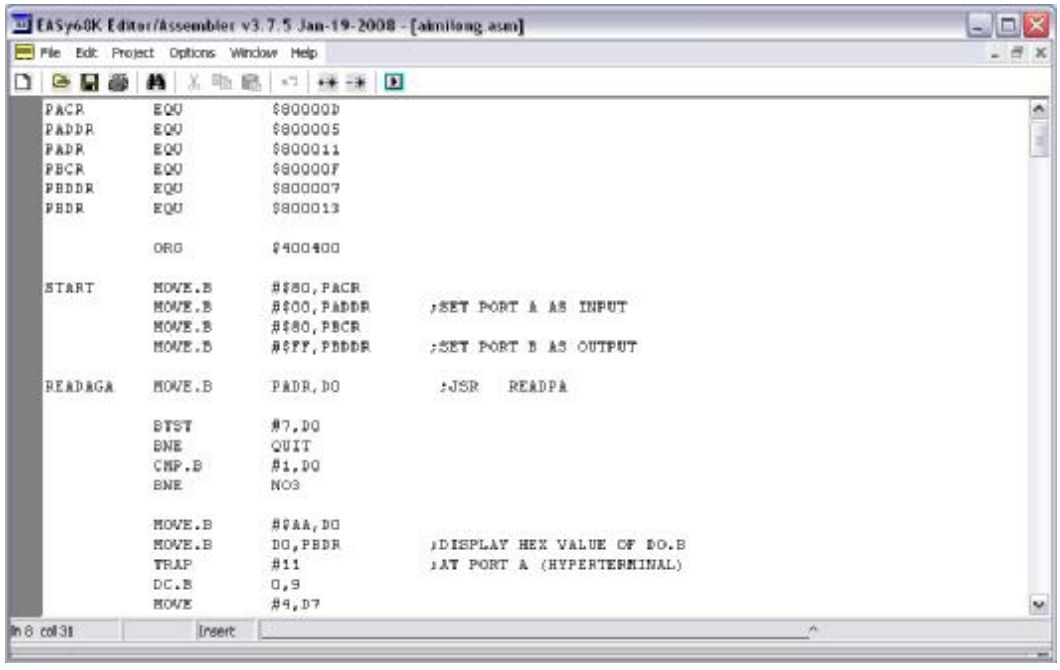


Figure 2.5 Write the Program using EASy68K software

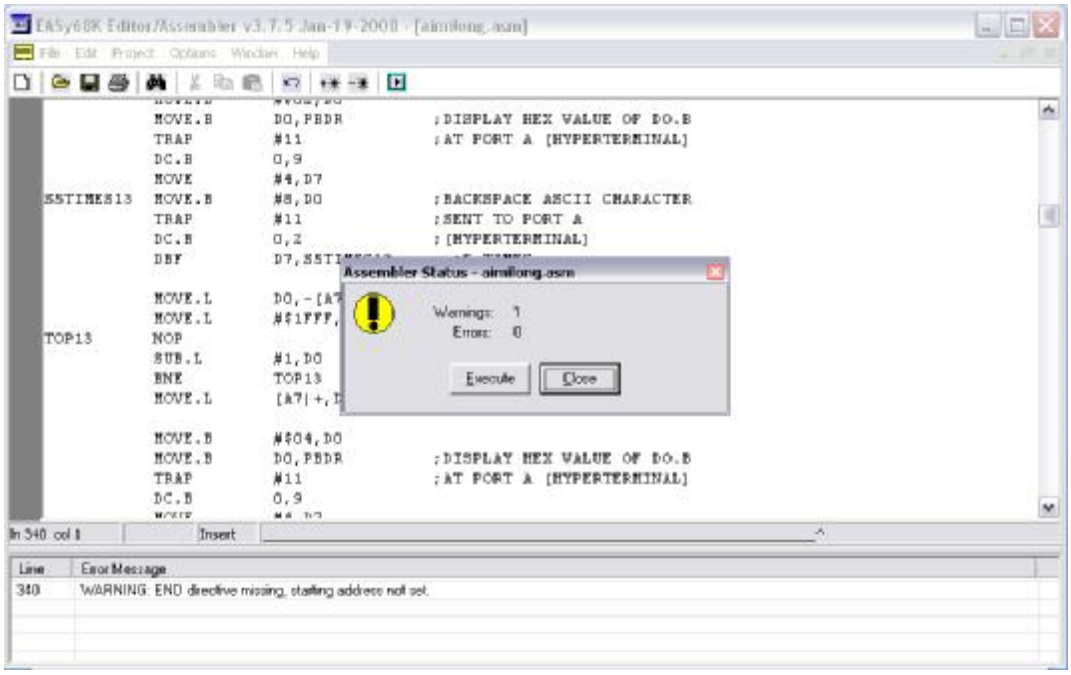


Figure 2.6 Window of EASy68K software

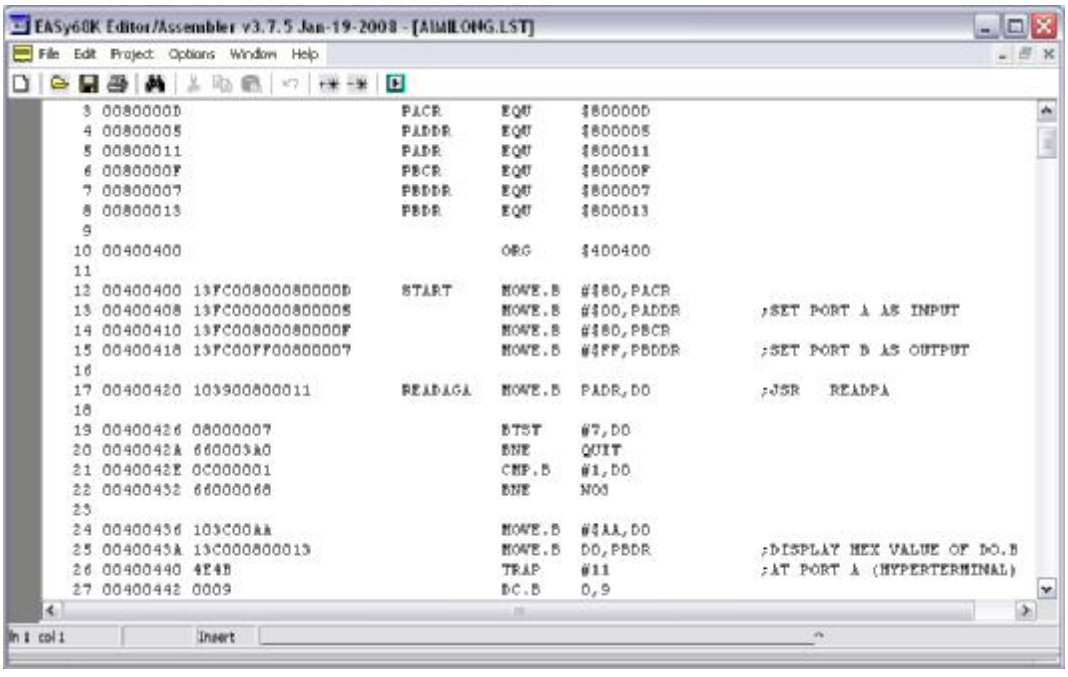


Figure 2.7 Listing File from EASy68K software

2.3.2 Command Prompt

Command Prompt can be use as the editor and assembler to the make a program. Figure 2.8 shows the editor to write the program using command prompt. After finish writing the program, assemble the program as shown in Figure 2.9 using XASM.exe. The error and warning can show after assemble. Re-edit the program if it has the error. Ignore the warning because it is not effect to the program. The listing file and .BIN file will be create after assemble.

For the second part of the project, the program will be burn to the EPROM. So, the program should be split to two. For split the program, SPLIT2.exe software can be use. Figure 2.11 shows the instruction to split the program to even and odd. Then, the program can burn to the EPROM using Chip Max.

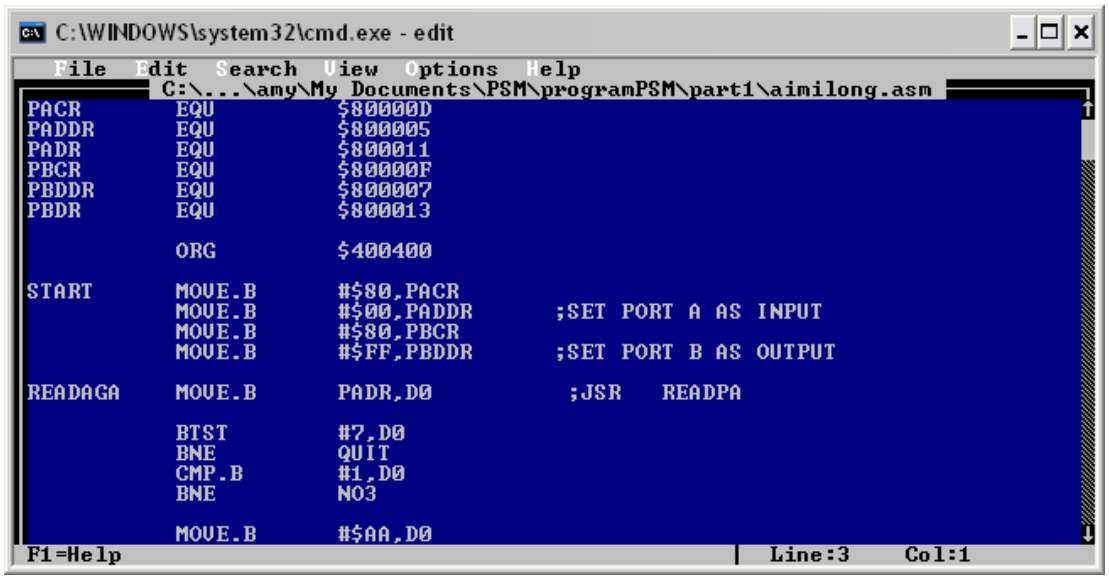


Figure 2.8 Write the Program Using Command Prompt

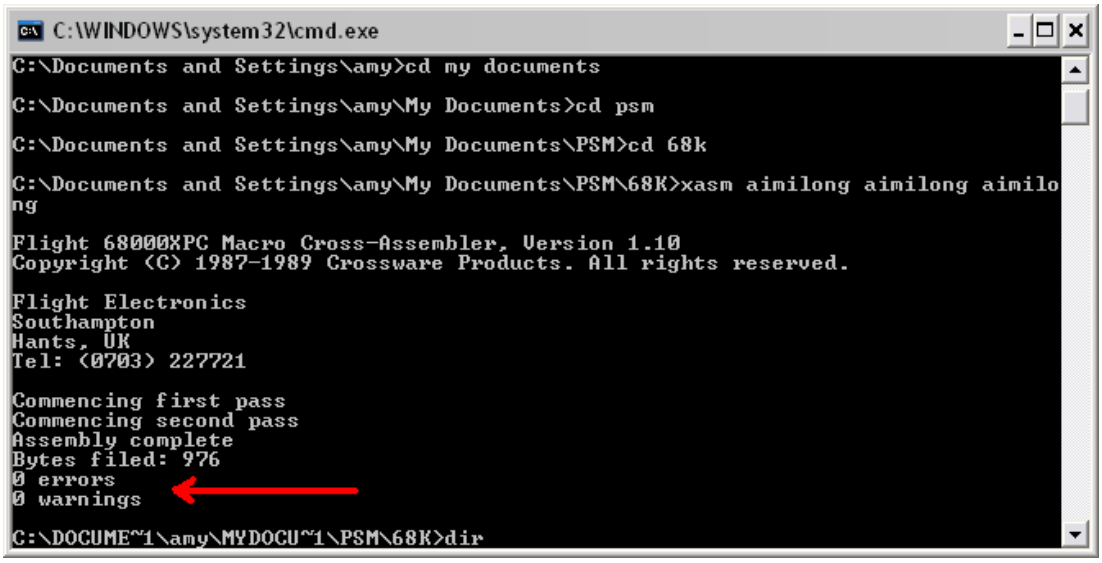


Figure 2.9 Assemble the Program Using Command Prompt

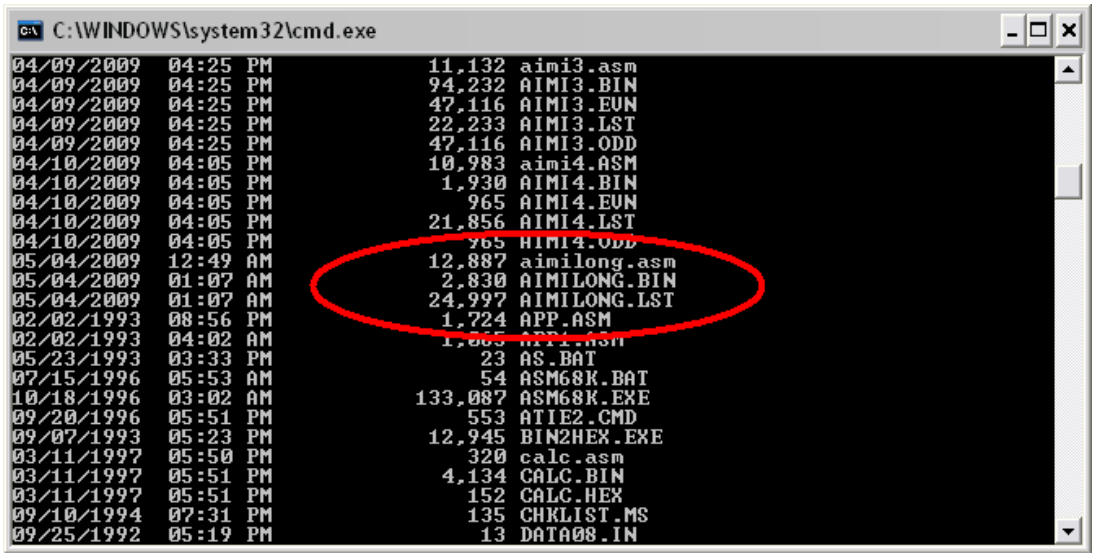


Figure 2.10 Create the Listing File and .BIN file

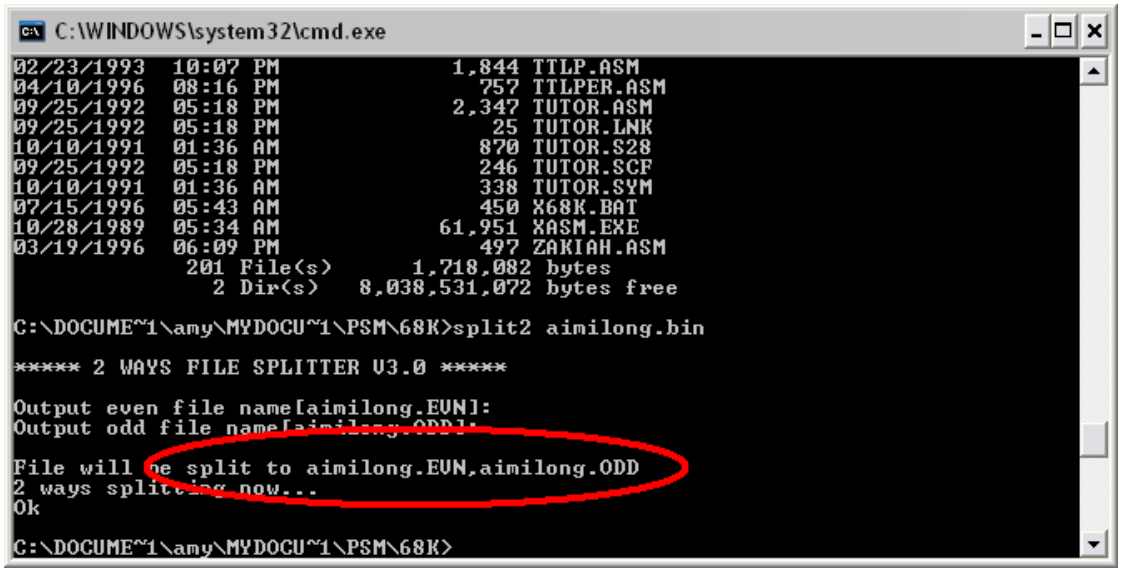


Figure 2.11 Split the .BIN file to .EVN and .ODD files Using Command Prompt

2.4 Test Hardware

In this project, the 68000 microprocessor board and application board will be use as the test hardware. EPROM 27C256 was used on the 68000 microprocessor board.

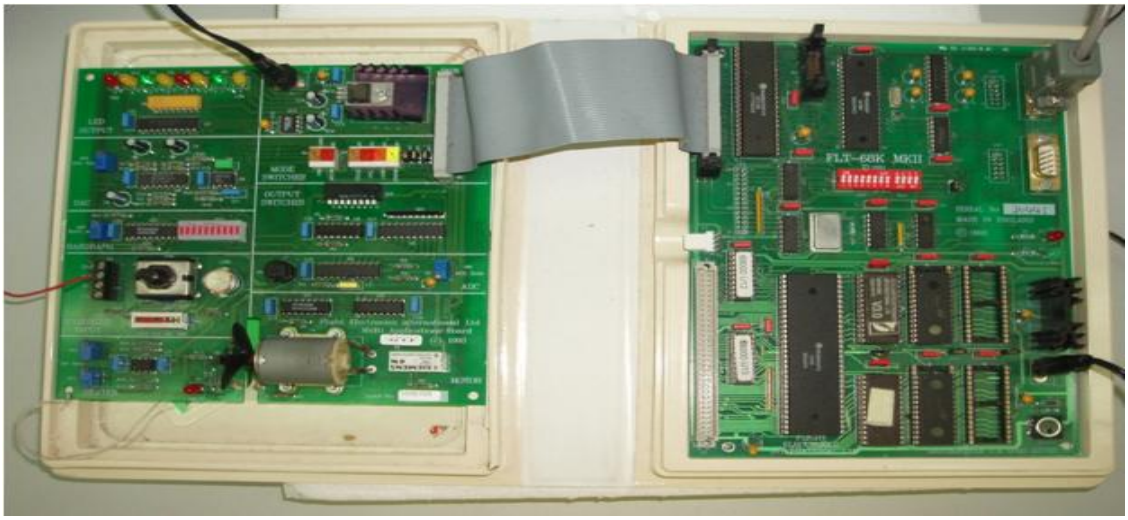


Figure 2.12 Application Board (Left Side) and 68000 Microprocessor Board (Right Side)

2.4.1 68000 Microprocessor Board

From Figure 2.12 was shown the architecture of 68000 Microprocessor Board. The important thing on this board is 68000 microprocessor and EPROM 27C256 (Even and Odd). Figure 2.13 shows the example of connection between 68000 microprocessor and EPROM 27C256.

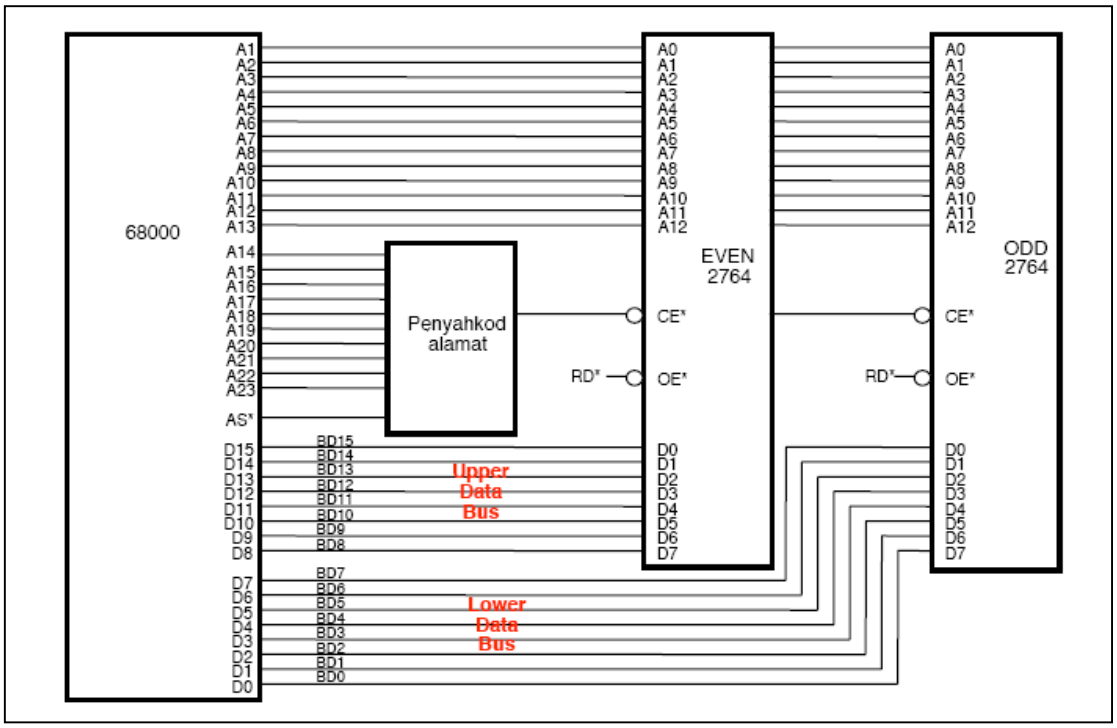


Figure 2.13 Connection between 68000 microprocessor and EPROM 27C64 (Example)

2.4.2 Existing Application Board

The Application Board was shown in Figure 2.12. The device that was used on the existing application board is Light Emitted Diode (LED) and switch. After the program was load on the 68000 microprocessor or burn on the EPROM, that devices can be function like the program planning before. The application board will be connected to the 68000 microprocessor board using the IDE cable.

2.4.3 EPROM 27C256

EPROM 27C256 is a non-volatile device that the data retained without power. The Figure 2.13 shows that the two EPROM was use on the 68000 microprocessor board because it has two parts of memory which is even and odd. EPROM must be erased using ultra-violet (UV) light before reprogramming. EPROM uses the expensive ceramic package and has a quartz window to allow UV to pass through. After the EPROM memory is clear, burn the new program on the EPROM using the programmer.

Figure 2.14 EPROM 27C256

2.5 Chip Max Programmer

Chip Max is one of the programmer that can be use for burn the program to the chip. EPROM 27C256 is available to use for program the chip. Actually, not all devices can be programmed using Chip Max. In Figure 2.15 shows the Chip Max instrument.

Figure 2.15 Chip Max Programmer

2.6 EPROM Eraser

EPROM must be erased using UV light. Before reprogramming the EPROM, ensure that the EPROM memory is clear without the data. Then, after erase the data, the new program can be programmed on the chip.

Figure 2.16 EPROM eraser

CHAPTER 3

METHODOLOGY

In this chapter, the methodology of this project will be achieving the objective of the project. Overall, in this project, the laboratory manual is dividing by four parts. For the first part is beginning on how to connect the Logic Analyser with 68000 Disassembler POD and the correct way of the connection of 68000 Disassembler POD to the 68000 microprocessor chip. The second part is the introduction about the Logic Analyser on how to set up the Logic analyser and trigger the data using reset vector. After the student has known the step how to use the Logic Analyser, the program will be uploaded to the 68000 microprocessor board by using Hyperterminal on the third part. The last part is triggering the data directly from the EPROM.

3.1 Part 1 of the Laboratory Manual

The beginning on how to connect the Logic Analyser with 68000 Disassembler POD had been explained on part 1 of the laboratory manual. Other items that also in part 1 is how to connect between 68000 Disassembler POD to the 68000 microprocessor chip. It is very important because if the 68000 Disassembler POD wrongly attach, the data couldn't appeared correctly from Logic Analyser. Refer to Appendix D to know how to make the connection correctly.

3.2 Part 2 of the Laboratory Manual

In part 2 of the laboratory manual was explained on how to set up the Logic Analyser and test the trigger function by using the reset vector. The set up is very important because the result will be effect if using the wrong set up. The laboratory manual on Appendix D will help the students to set up the Logic Analyser. The reset vector is the starting point of 68000 microprocessor to start the program. The reset vector for 68000 microprocessor is $000000. In part 2, the set up of trigger function was provided. The laboratory manual must be followed so that the students can use it perfectly.

3.3 Part 3 of the Laboratory Manual

Figure 3.1 is showing the planning for the Part 3 of this project that separate into three sessions which is software, hardware and analysis. This part is different with part 4. Part 3 is comparing between the data from the Logic Analyser and the listing file.

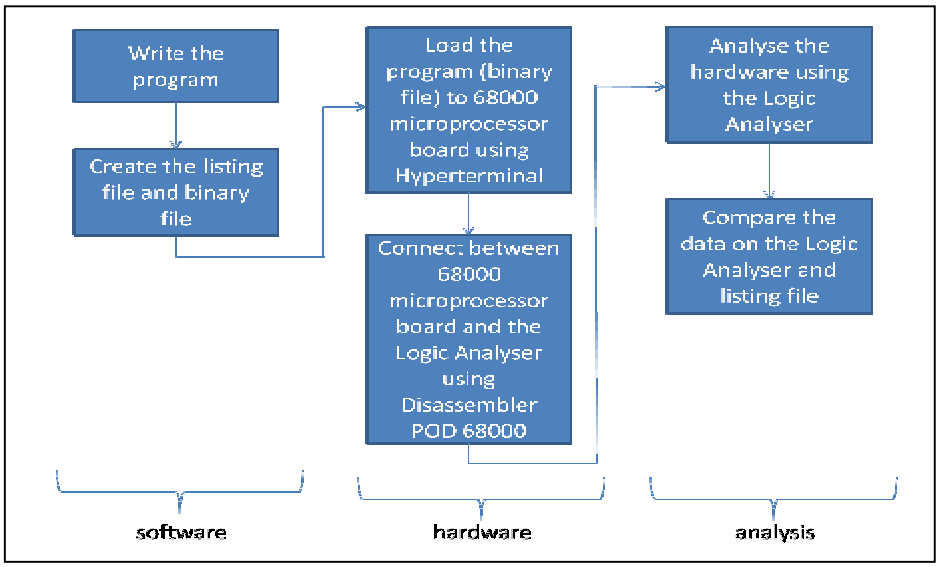


Figure 3.1 Flow Chart for the Part 3

3.3.1 Software for Part 3 of the Laboratory Manual

In this project, the software part was written on EASy68K editor. The software that will be used in this project has been explained in literature review. After completing writing the program in a source file (.asm), the file is assembled to create the S-format which has the machine codes of the program if there is no error. Then, to load the program to the 68000 microprocessor board, the S-format file will be created. Data in the S-format file will be loaded into the user memory of the 68000 board. The step load the program into the user memory of the 68000 board is shown in Figure 2.9 in literature review. The listing file created together will be used to analyse the hardware operation of the 68000 board by comparing with the Logic analuser display. The program for this part is in Appendix A and the listing file is in Appendix B.

3.3.2 Hardware for Part 3 of the Laboratory Manual

The hardware is a 68000 microprocessor board and application board. The detail about the existing board was explained in literature review.

Hyperterminal is the terminal emulator software which can be use for upload the program to the 68000 microprocessor board. The 68000 microprocessor board is connected to the computer using the RS-232 interface. The program can be uploaded to the 68000 microprocessor board by taking the step shown on Figure 3.2 until Figure 3.9.

Figure 3.2 Hyperterminal (Connection Description)

Figure 3.3 Hyperterminal (Connect To)

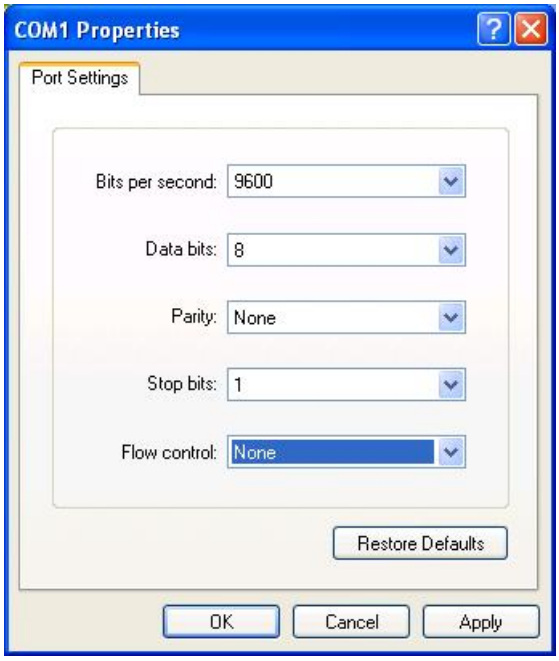


Figure 3.4 COM1 Properties

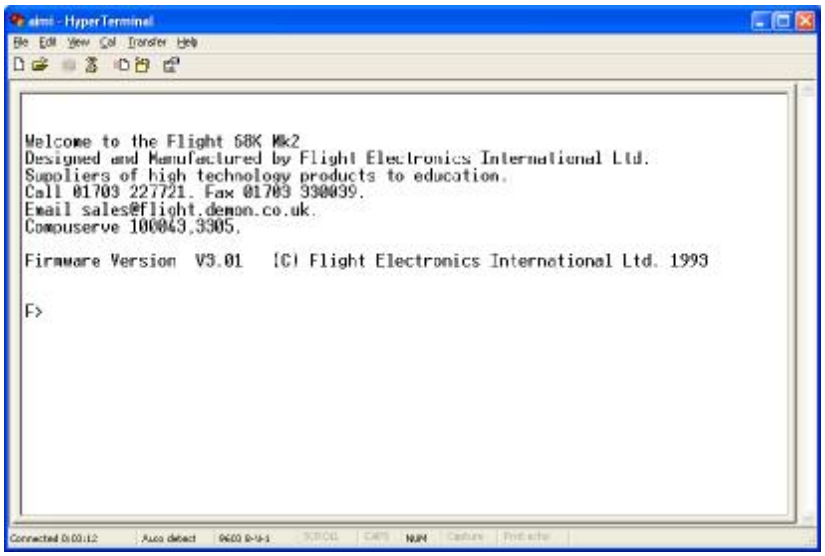


Figure 3.5 After press ENTER (3 times)

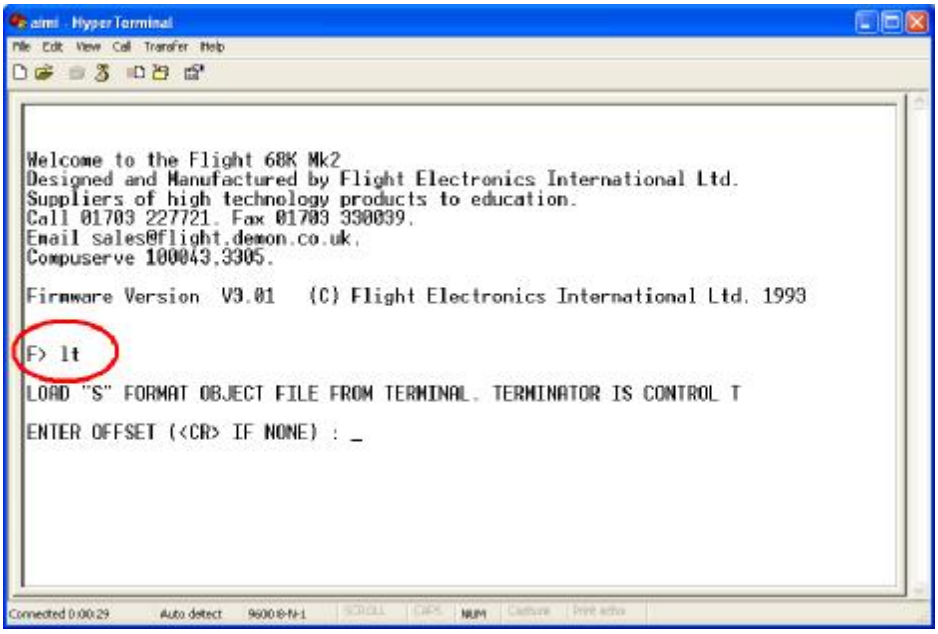


Figure 3.6 Type ‘lt’

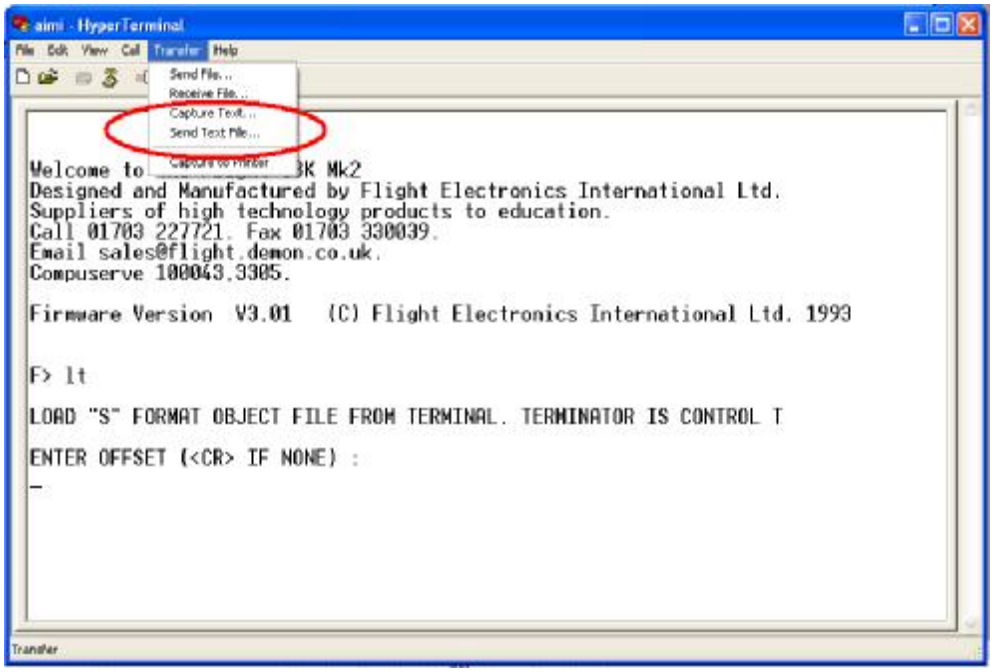


Figure 3.7 Load data .BIN to chip using Hyperterminal

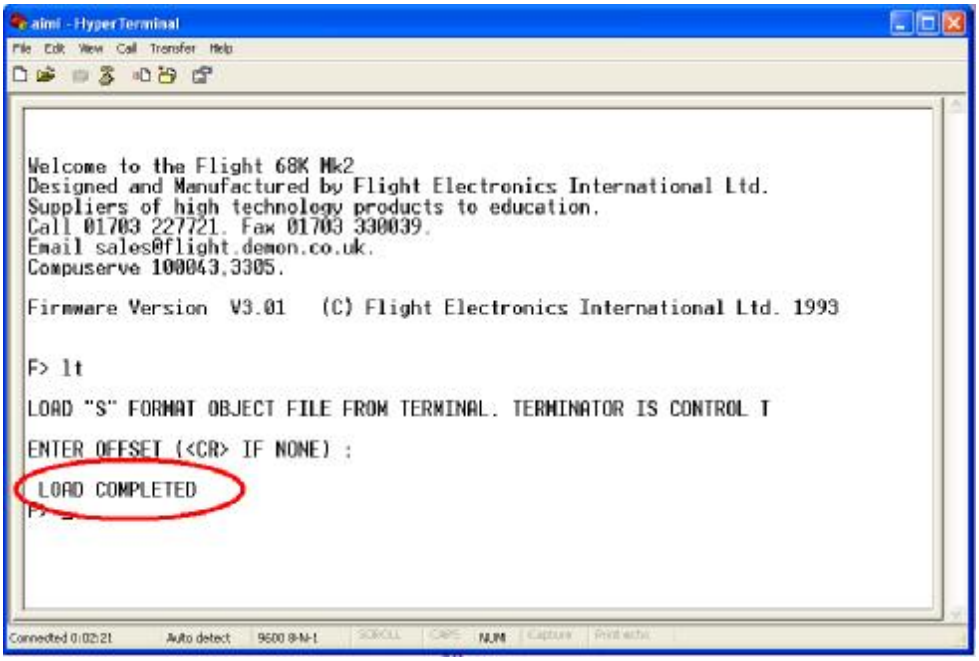


Figure 3.8 Load Completed

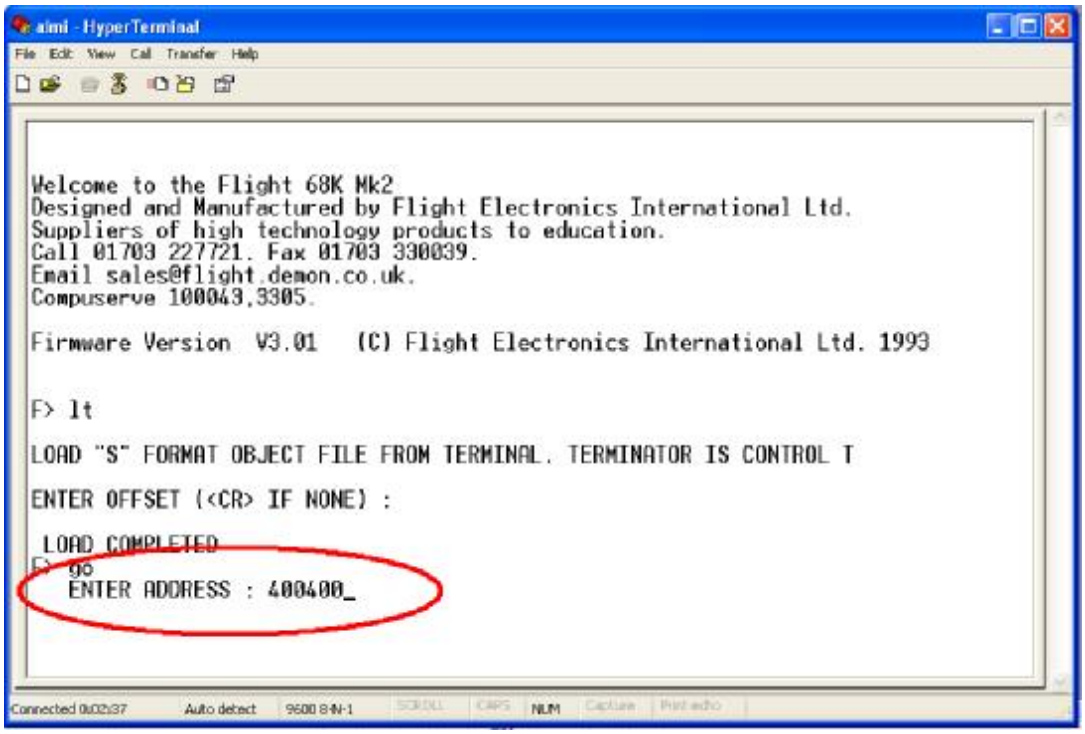


Figure 3.9 Type Address for Microprocessor

After the program is uploaded on the 68000 microprocessor board, the 68000 Disassembler POD is connected to the 68000 microprocessor chip to observe the operation of the 68000 microprocessor board. The connection between 68000 Disassembler POD and 68000 microprocessor chip was shown on Figure 3.10.

Figure 3.10 Connections between 68000 Disassembler POD and 68000 microprocessor chip

3.3.3 Analysis for Part 3 of the Laboratory Manual

The last step for the Part 3 is the analysis the data read from the display of the logic analyser. The trigger function on the Logic Analyser can be used to initiate data to be collected from 68000 Disassembler POD. We can compare the data with listing file that was created before. The data should be get from the Logic Analyser supposedly same with listing file. The detail operation of the 68000 microprocessor can be triggered by the Logic Analyser. The detail step for analyse the data is in the laboratory manual on Appendix D.

3.4 Part 4 of the Laboratory Manual

Figure 3.11 shows the steps taken to implement the Part 4 laboratory. The step of this part is quite similar with part 3. It is separate into three sessions which is software, hardware and analysis. This part is like a reverse engineering method. Students do not know the program that was burn on the EPROM. Then, student can use the Logic Analyser to analyse the program on the EPROM.

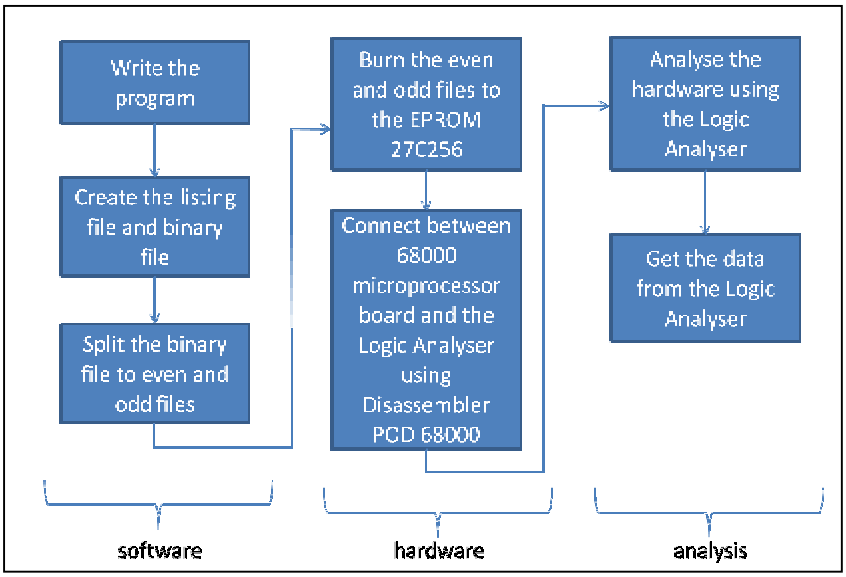


Figure 3.11 Flow Chart for the Part 4

3.4.1 Software for Part 4 of the Laboratory Manual

EASy68K editor is the software that will be use in this part of the laboratory manual which is same in part 3. The different in this part is that the program should be written for ROMable code for RESET entry so it must be located in the address of the ROM area. The program for this part is shown in Appendix C.

After completing writing the program in a source file (.asm), the file is assembled to create the S-format which has the machine codes of the program if there is no error. The binary file created will be used as the data to program to the EEPROM as the 68000 board simple boot firmware. Then the even address and odd address data of program has to be divided to separate file so that the original binary codes can be burned into two EPROM (high (even) byte and low (odd) byte) because on the 68000 microprocessor architecture used two used using command prompt. The command for split the binary file was shown in Figure 2.11.

3.4.2 Hardware for Part 4 of the Laboratory Manual

Labeling the EPROM as even and odd to make sure the ROMs will not be inter change. Each EPROM label are program with the data from their respective splitted binary file using the Chip Max programmer. Before burning the program, the memory of EPROM was erased. If the EPROM is still with the data, erase the memory of the EPROM using EPROM eraser. The details about EPROM eraser was explained in literature review.

Then the EPROM 27C256's was replace on their respective socket on the 68000 Microprocessor Board.

Figure 3.12 Replace the EPROM from the 68000 microprocessor board

3.4.3 Analysis for Part 4 of the Laboratory Manual

Analysis the data that can be triggered from the Logic Analyser is the last step for the Part 4. Actually, for this part, students do not know the program that was burned on the EPROM. So, in this step, student will be triggered the data to know the program that was operated on the 68000 microprocessor board. The detail step for analyse the data is in laboratory manual on Appendix D.

CHAPTER 4

RESULT ANALYSIS

To ensure the program execute properly, the program need to test on the existing tested hardware. In this chapter, the observation from the Logic Analyser's data is the main item. Four parts from the laboratory manual are the steps on how to use the Logic Analyser properly. The result can be analyse on the experiment session.

The trigger function will be used to initiate aquiring the data from the Logic Analyser. Laboratory manual stated that the trigger function of the Logic Analyser is on part 2, part 3 and part 4. Part 2 is the basic of trigger function while part 3 is comparing the data between the Logic Analyser and listing file. This chapter will clarify the research results. The result for this project is the procedure and the question created on the laboratory manual.

4.1 Part 1 of the Laboratory Manual

4.1.1 Instructions from the Laboratory Manual for Part 1

In this part, student will learn how to connect the Logic Analyser with 68000 Disassembler POD and attach the 68000 Disassembler POD to 68000 microprocessor chip. No question is given on this part by laboratory manual to be answer by the student. Figure 4.1 (a) and (b) shows the instruction of Part 1 of the laboratory manual.

Part 1 : Beginning

1. In this experiment we will use Logic Analyser to analyse the working of a 68000 Microprocessor.
2. Connect the Disassembler POD 68000 to the Logic Analyser. Make sure that the Disassembler POD 68000 is connecting correctly on MC68000 chip.

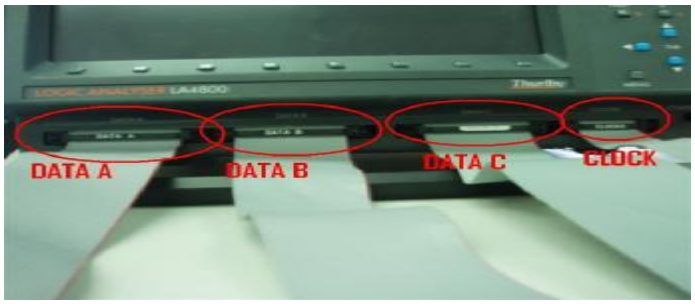


Figure 3: Connection to the Logic Analyser

3. Switch on the Logic Analyser then check the 68000 Disassembler POD was assembled or not.

Figure 4.1 (a) Instructions for the Part 1 of the Laboratory Manual

4. Clip the 68000 Disassembler POD to the chip MC68000 on 68000 Microprocessor Board. Make sure the connection is correct (Pin 1 on DP 68000 to Pin 1 on chip).

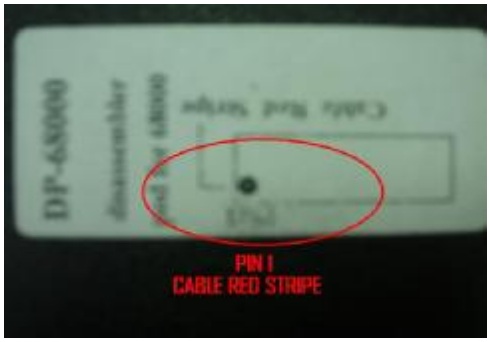


Figure 4: 68000 Disassembler POD

Figure 5: Connection between 68000 Disassembler POD to 68000 microprocessor

5. Press Confirm.

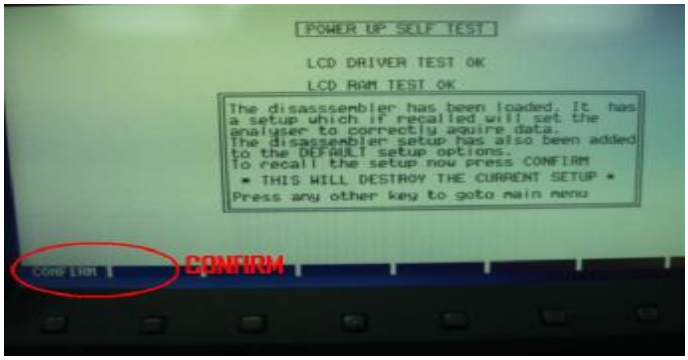


Figure 6: The disassembler has been loaded

Figure 4.1 (b) Instructions for the Part 1 of the Laboratory Manual

4.2 Part 2 of the Laboratory Manual

Part 2 is the basic set up of the Logic Analyser and how to use the trigger function. The data that will be trigger on this part, started from reset vector ($000000). But the result only showed certain part of the data because a lot of data line on the Logic Analyser been registered.

4.2.1 Instructions from the Laboratory Manual for Part 2

Figure 4.2 (a) and (b) shows the instruction of Part 2 of the laboratory manual.

Part 2 : Introduction for the Logic Analyser Using Reset Vector

1. Press 1 for CONFIGURATION.
   i. Set clock selected as EXTERNAL.
   ii. Press MENU.

2. Press 1 for CONFIGURATION.
   i. Set clock selected as EXTERNAL.
   ii. Press MENU.

ADDR00 → A00
ADDR01 → A01
ADDR02 → A02
ADDR03 → A03
⋮
ADDR15 → A15

Figure 4.2 (a) Instructions for the Part 2 of the Laboratory Manual

iii. Press the button on right side of Logic Analyser to rename the address.
iv. Press tab cursor to continue change the label. Repeat step Part 2(2(ii) and 2(iii)).
v. After finish labeling, press EXIT.
vi. Press MENU.

4. Press 3 for STATE LISTING.
iv. Confirm that, no data on the Logic Analyser.
v. Press PAGE DEC (page decreases) or GOTO TRG (go to trigger) until it shows the '0000' POS on the upper left of Logic Analyser display (Starting point it will be triggering). **POS is position.
vi. Press MENU.

5. Press 4 for TRIGGER SETUP.
i. On Address HEHEX, Trig Wrd, set as 000000 (Reset Vector).
ii. Press MENU.

6. Press 7 for 68000 DISASSEMBLER VER (1.01)
i. Press RUN button on right side of Logic Analyser.
ii. Press SINGLE on the list menu on the bottom.

7. Switch on the Flight 68000 Board.
i. Press reset button on 68000 Microprocessor Board.

8. Logic Analyser will be triggered the data and displayed to the screen.

Figure 8: Trigger Function

9. Answer the question in SECTION A.

Figure 4.2 (b) Instructions for the Part 2 of the Laboratory Manual

4.2.2 Question and Answer for Part 2 of Laboratory Manual (SECTION A)

Figure 4.3 (a) and (b) shows the question and answer for part 2 of the laboratory manual. The questions will be given to the student to ensure that they are understood of some necessary instruction on how to use Logic Analyser.

Section A

1. What the operation on cursor 0027?
   ANSWER: MOVEA.L 00000080,D0

2. On cursor 0028, write the Address, Data, Operation, Bus Transfer, UDS/LDS, R/W and FC2-FC0.
   ANSWER:

Cursor	Address	Data	Operation	Bus Transfer	UDS/LDS	R/W	FC2-FC0
0028	**A0000A**	**0000**	**sp data wr**	**-low byte**	**10**	**0**	**101**

Table 1: DISASSEMBLER DISPLAY

   a) What does the result of R/W signal signify?
      ANSWER: Write

   b) What does the result of UDS/LDS signal signify on cursor 0028.
      ANSWER: The data is on low byte. So, UDS is 1 and LDS is 0.

3. We can see the operation on cursor 0021 until 0023;

Cursor	Address	Data	Operation	Bus Transfer	UDS/ LDS	R/ W	FC2-FC0
0021	00223A	2C7C	MOVEA.L 00A00001,A6		00	1	110
0022	00223C	00A0	sp prog rd		00	1	110
0023	00223E	0001	sp prog rd		00	1	110

Table 2: DISASSEMBLER DISPLAY

Figure 4.3 (a) Question and Answers for the Part 2 of the Laboratory Manual

a) Refer to operation on cursor 0021, which one the destination?
ANSWER: A6

b) What is the operand?
ANSWER: 00A00001

c) Explain the about the data on cursor 0022 and 0023.
ANSWER: For the first word from the operand (00A0), it will put on address 00223C and second word (0001), it will put on address 00223E.

d) How many bus cycles on that table?
ANSWER: 2 bus cycle

Figure 4.3 (b) Question and Answers for the Part 2 of the Laboratory Manual

4.2.3 Data from the Listing File for Part 2 of Laboratory Manual

The listing file is not included in this part because all the data was taken directly from 68000 microprocessor board. Therefore, the data are only getting from the Logic Analyser. The trigger function had been used to simplify Logic Analyser to make the breakpoint.

4.3 Part 3 of the Laboratory Manual

In part 3, the comparison between data from the Logic Analyser and listing file could be seen. All the data from Logic Analyser are more detail compared with listing file. By using Logic Analyser, memory will be seen easily by student that stored in it address.

4.3.1 Instructions from the Laboratory Manual for Part 3

In this part, student will be exposed with the other application of trigger function such as $400400. That address is a starting point for program that already uploaded to the 68000 microprocessor board. Figure 4.4 (a) and (b) shows the instruction of Part 3 of the laboratory manual.

Part 3 : Using Hyperterminal Trigger the Another Address

1. Type the program on the APPENDIX.

2. Assemble the program using command prompt to get .BIN file (refer to the laboratory manual Flight 68000).

3. Press MENU on the Logic Analyser.

4. Press 4 for TRIGGER SETUP.
   i. Change the Address HEHEX (should be HEX but this is the display in the Logic Analyser), Trig Wrd (means Word), to 400400 (Starting address for microprocessor).
   ii. Press MENU.

5. On 68000 Microprocessor Board.
   i. Switch on the 68000 Microprocessor Board.
   ii. Open Hyperterminal on the PC.
   iii. Press reset on 68000 Microprocessor Board.
   iv. Press ENTER (3 times) on keyboard. This figure will be entering.
   v. Type ‘lt’.
   vi. Press ENTER after Hyperterminal show ENTER OFFSET (<CR IF NONE>):
   vii. Click TRANSFER on the MENU, and then SEND TEXT FILE.
   viii. Search .BIN file that you create before.
   ix. After LOAD COMPLETED, type ‘go’ and enter address 400400
   x. Press ENTER.

Figure 4.4 (a) Instructions for the Part 3 of the Laboratory Manual

6. (Assuming you have press Menu as instructed above in Part 3(1)(ii)) Press 7 for 68000 DISASSEMBLER VER (1.01)
   i. Press RUN button on right side of Logic Analyser.
   ii. Press SINGLE on the list menu on the bottom.
   iii. Repeat procedure Part 3 (2(iii), 2(iv), 2(ix))

7. Logic Analyser will be triggered on the screen and it will get the data.

8. Answer the question SECTION B.

Figure 4.4 (b) Instructions for the Part 3 of the Laboratory Manual

4.3.2 Question and Answer for Part 3 of Laboratory Manual (SECTION B)

To ensure that the student understood of some necessary instruction on how to use Logic Analyser, the questions will be given. Figure 4.5 (a), (b) and (c) shows the question and answer for part 3 of the laboratory manual.

Section B

1. After Logic Analyser was triggered, what the DISASSEMBLER DISPLAY showed.

ANSWER:

Address	Data	Operation
400400	**13FC**	**MOVE.B #80,0080000D.L**
400402	**0080**	**sp prog rd**
400404	**0080**	**sp prog rd**
400406	**000D**	**sp prog rd**
400408	**13FC**	**MOVE.B #00,00800005.L**
****80000C**	**8080**	**sp prog wr**
40040A	**0000**	**sp prog rd**
40040C	**0080**	**sp prog rd**
40040E	**0005**	**sp prog rd**

Table 3: DISASSEMBLER DISPLAY **(**GLITCH)**

Figure 4.5 (a) Question and Answers for the Part 3 of the Laboratory Manual

2. After Logic Analyser was triggered, what the STATE LISTING showed.
   ANSWER:

Address	BERR	BGACK	UDS/LDS	R/W	FC2-FC0
400400	**1**	**1**	**00**	**1**	**110**
400402	**1**	**1**	**00**	**1**	**110**
400404	**1**	**1**	**00**	**1**	**110**
400406	**1**	**1**	**00**	**1**	**110**
400408	**1**	**1**	**00**	**1**	**110**
****80000C**	**1**	**1**	**10**	**0**	**101**
40040A	**1**	**1**	**00**	**1**	**110**
40040C	**1**	**1**	**00**	**1**	**110**
40040E	**1**	**1**	**00**	**1**	**110**

Table 4: STATE LISTING **(**GLITCH)**

3. Compared the listing file in Part 3 with DISASSEMBLER DISPLAY in Table 1. It is same? Why?
   ANSWER:
   The data on the Logic Analyser is the same when we compared with listing file because the data that was transferred on 68000 microprocessor chip is the same program.

4. Open the timing diagram and draw the FC2-FC0 (press group if not display).
   ANSWER:the timing diagram must be related each other**

5. What the relationship between timing diagram and Table 4. Explain.
   ANSWER:the timing diagram must be related each other**

Figure 4.5 (b) Question and Answers for the Part 3 of the Laboratory Manual

6. Repeat the procedure in Part 3 (no need to begin the step 5, proceed to 6). Change Address HEHEX, Trig Wrd to 400408. After Logic Analyser was triggered, what the STATE LISTING showed.

ANSWER:

Address	BERR	BGACK	UDS/LDS	R/W	FC2-FC0
400408	**13FC**	**MOVE.B #00,00800005.L**	**00**	**1**	**110**
****80000C**	**8080**	**sp prog wr**	**10**	**0**	**101**
40040A	**0000**	**sp prog rd**	**00**	**1**	**110**
40040C	**0080**	**sp prog rd**	**00**	**1**	**110**
40040E	**0005**	**sp prog rd**	**00**	**1**	**110**

Table 5: STATE LISTING

7. Compare the Table 5 and Table 4. Explain.

ANSWER:It is the same thing because we run the same program.

Figure 4.5 (c) Question and Answers for the Part 3 of the Laboratory Manual

4.3.3 Data from the Listing File for Part 3 of Laboratory Manual

Figure 4.6 was shows the listing file of the program. When the result from table 3 (in Figure 4.5 (a)) comparing with Figure 4.6, the data is similar, but the data from the Logic Analyser is more detail compared with data in the listing file.

If look at the address $400400, the operation from logic analyser is ‘MOVE.B #80,0080000D.L’ but from the listing file is ‘MOVE.B #$80, PACR’. Actually this is the same. PACR was declared before start the program as $0080000D. The Logic Analyser will analyse the PACR is 0080000D as the hexadecimal number. So, for the conclusion between the Logic Analyser data and listing file are same.

```
 3 0080000D                          PACR    EQU     $80000D
 4 00800005                          PADDR   EQU     $800005
 5 00800011                          PADR    EQU     $800011
 6 0080000F                          PBCR    EQU     $80000F
 7 00800007                          PBDDR   EQU     $800007
 8 00800013                          PBDR    EQU     $800013
 9
10 00400400                                  ORG     $400400
11
12 00400400 13FC00800080000D  START   MOVE.B  #$80,PACR
13 00400408 13FC000000800005          MOVE.B  #$00,PADDR
14 00400410 13FC00800080000F          MOVE.B  #$80,PBCR
15 00400418 13FC00FF00800007          MOVE.B  #$FF,PBDDR
```

Figure 4.6 Listing File for part 3 of the Program

4.4 Part 4 of the Laboratory Manual

Part 4 is like a reverse engineering method. The students do not know the program that have on the chip. To get the data from the chip, the student can use the Logic Analyser to capture the data. The programmed EPROM will be replace from the original EPROM for observe the data.

4.4.1 Instructions from the Laboratory Manual for Part 4

In this part, student will be exposed with the other application of trigger function such as $000000. That address is a starting point for program that already uploaded to the 68000 microprocessor board. Figure 4.7 shows the instruction of Part 3 of the laboratory manual.

Part 4 : Trigger the data directly from the EPROM

1. Switch off the Microprocessor Board. Replace the EPROM 27C256 on the 68000 Microprocessor Board to the another EPROM that was programmed. Below is the listing file of the program.

2. Press 4 for TRIGGER SETUP.
   i. Change the Address HEHEX, Trig Wrd, to 000000.
   ii. Press MENU.

3. Press 7 for 68000 DISASSEMBLER VER (1.01)
   i. Press RUN button on right side of Logic Analyser.
   ii. Press SINGLE on the list menu on the bottom.

4. On 68000 Microprocessor Board.
   i. Switch on the 68000 Microprocessor Board.
   ii. Press reset on 68000 Microprocessor Board.

5. Logic Analyser will be triggered the data and displayed to the screen.

Figure 19: Trigger Function

6. Check the data on Logic Analyser and answer the question on SECTION C.

Figure 4.7 Instructions for the Part 4 of the Laboratory Manual

4.4.2 Question and Answer for Part 4 of Laboratory Manual (SECTION C)

The student will understood how to use Logic Analyser when the answer the questions that will be given in Section C of the laboratory manual. Figure 4.8 (a) and (b) shows the question and answer for part 4 of the laboratory manual.

Section C

1. Write the program that you can get from the Logic Analyser (from address 000400 until 000436).

ANSWER:

```
MOVEA.L   #4003F0,A7
MOVE.B    #80,0080000D.L
MOVE.B    #00, 00800005.L
MOVE.B    #80, 0080000F.L
MOVE.B    #FF, 00800007.L
MOVE.B    00800011.L,D0
EORI.W    #00FF,D0
MOVE.B    D0, 00800013.L
JMP       0426.W
```

2. What is the benefit if we used the ROM that was programmed?

ANSWER:

- **No needs to use the PC for upload the program.**

3. Press the SW1 port 0, what happened to LED?

ANSWER:

- **LED 0 light-off**

4. Press the SW1 port 0 until port 7, what happened to LED?

ANSWER:

- **All LED light-off**

Figure 4.8 (a) Question and Answers for the Part 4 of the Laboratory Manual

5. Rewrite the program will make LED to light-up when the switch is pressed.
   ANSWER:

```
MOVEA.L    #4003F0,A7
MOVE.B     #80,0080000D.L
MOVE.B     #00, 00800005.L
MOVE.B     #80, 0080000F.L
MOVE.B     #FF, 00800007.L
MOVE.B     00800011.L,D0
MOVE.B     D0, 00800013.L
JMP        0426.W
```

Figure 4.8 (b) Question and Answers for the Part 4 of the Laboratory Manual

4.4.3 Aquired Data from the Logic Analyser

To trigger the data from the chip, set the trigger function as $000000. It will trigger when the address was found on the chip. After the Logic Analyser was triggered, the data will be display on multiple digital signals on a single screen. The triggered result by using the Logic Analyser was shown on Figure 4.2.

```
MOVEA.L    #4003F0,A7
MOVE.B     #80,0080000D.L
MOVE.B     #00, 00800005.L
MOVE.B     #80, 0080000F.L
MOVE.B     #FF, 00800007.L
MOVE.B     00800011.L,D0
EORI.W     #00FF,D0
MOVE.B     D0, 00800013.L
JMP        0426.W
```

Figure 4.9 The Program That Shown on a Single Screen

4.4.4 The Program for Part 4 of the Laboratory Manual

The program below is cross-assembled, and programmed to the EPROM for the 68000 board compared with the data aquired from the Logic Analyser. Figure 4.3 is the actual program of this part. When comparing the program, it is same.

```
PACR        EQU         $80000D
PADDR       EQU         $800005
PADR        EQU         $800011
PBCR        EQU         $80000F
PBDDR       EQU         $800007
PBDR        EQU         $800013
            bin
            ORG         0
            DC.L        $4003F0
            DC.L        START
            DS.B        $400-8
            ORG         $400
START       MOVEA.L     #$4003F0,A7
            MOVE.B      #$80,PACR
            MOVE.B      #$00,PADDR
            MOVE.B      #$80,PBCR
            MOVE.B      #$FF,PBDDR
READAGA     MOVE.B      PADR,D0
            EOR         #%11111111,D0
            MOVE.B      D0,PBDR
            JMP         READAGA
here        DS.B        $8000-*
            END
```

Figure 4.10 The Program for Part 4 of the Laboratory Manual

CHAPTER 5

DISCUSSION

The steps on how to use the Logic Analyser properly was include on the laboratory manual. The result was shows in chapter 4. After completing this project, the result of the laboratory manual will be discuss in details in this chapter. The trigger function will be used to initiate aquiring the data from the Logic Analyser. Laboratory manual stated that the trigger function of the Logic Analyser is on part 2, part 3 and part 4. Part 2 is the basic of trigger function while part 3 is comparing the data between the Logic Analyser and listing file. This chapter will clarify the research results.

5.1 Part 1 of the Laboratory Manual

In this part, student will learn how to connect the Logic Analyser with 68000 Disassembler POD and attach the 68000 Disassembler POD to 68000 microprocessor chip. The respond from the person who had done the testing said that this part is easy to follow. Question is not given on this part because it just the basic of the connection of the Logic Analyser.

5.2 Part 2 of the Laboratory Manual

On the result only showed certain part of the data because a lot of data line on the Logic Analyser been registered. The student will be understood of some necessary instruction on how to use Logic Analyser after answer the question.

From Figure 4.3 (a) (Table 1), the data from the Logic Analyser was shown that the operation on address $A0000A on cursor 0028. From the listing file, this part not available in analysis because we not create the source code of the program. But by using Logic Analyser, the data can be observed. Figure 4.3 (a) (Table 2), shown the instructions from the 68000 microprocessor board on address $00223A. This program already embedded on the board for reset vector function.

5.3 Part 3 of the Laboratory Manual

The comparison between data from the Logic Analyser and listing file could be seen. $400400 is a starting point for program that already uploaded to 68000 microprocessor board. Compared with the listing file, the data from Logic Analyser are more details.

From Figure 4.5 (a) (Table 3), the data from the Logic Analyser was shown. That is the operation on address $400400 until $40040E. The data between the Logic Analyser and the listing file in Figure 4.6 is same but in the data on Logic Analyser dispaly are more details. Figure 4.5 (a) (Table 4), is the details from the state listing on the Logic Analyser. And the operations which had done by every line in 68000 microprocessor chip. Here also shown active or inactive of each line in the chip.

5.4 Part 4 of the Laboratory Manual

Reverse engineering method was used in this part. Figure 4.7 was shows the step how to capture the data from the 68000 microprocessor. The program below is cross-assembled, and programmed to the EPROM for the 68000 board compared with the acquired data from the Logic Analyser. Figure 4.10 is the program that be created before programmed the EPROM. When comparing the program, the data is same with the acquired data from the Logic Analyser.

```
PACR        EQU       $80000D
PADDR       EQU       $800005
PADR        EQU       $800011
PBCR        EQU       $80000F
PBDDR       EQU       $800007
PBDR        EQU       $800013
            bin
            ORG       0
            DC.L      $4003F0
            DC.L      START
            DS.B      $400-8
            ORG       $400
START       MOVEA.L   #$4003F0,A7
            MOVE.B    #$80,PACR
            MOVE.B    #$00,PADDR
            MOVE.B    #$80,PBCR
            MOVE.B    #$FF,PBDDR
READAGA     MOVE.B    PADR,D0
            EOR       #%11111111,D0
            MOVE.B    D0,PBDR
            JMP       READAGA
here        DS.B      $8000-*
            END
```

Figure 5.1 The program that be created before programmed the EPROM

CHAPTER 6

CONCLUSION AND SUGGESTIONS

6.1 Recommendation on Future Works

This project is only to apply certain part of the function of the logic analyser because of the time constraint. Most of the time is spent on exploring the functionality of the Logic Analyser because this project runs without a specific usage manual. This project could be further expand so that all the other functions in the Logic Analyser.

Among of the suggestion to continue this project is by utilising the print function that is available on the Logic Analyser. To print the data from the Logic Analyser, print function is very important because the triggered data is very long and taking long time to write by hand. The advantage of this function is very useful to the students.

Other than that, for part 4 in the provided lab manual should try to use the EEPROM because it will save a lot of time. By the EPROM, it will take quite long time erase and reprogrammed, so if there is error after the program is burned to erase the EPROM and reprogrammed will take very long time. EEPROM can be erased by overwriting during reprogramming and take no extra time for erasing.

6.2 Conclusion

As a conclusion, this project has been completed successfully and fulfilling the objective and scope specified. By practicing the laboratory manual for this project, student will be expose on method of writing a program, cross-assembling the program, writing a ROMable program and splitting the binary file of the program so the data can be stored in two ROM mapped as even and odd byte. After replacing the ROM on its respective socket on the 68000 board, student will be exposed on how to use the Logic Analyser to analyse the 68000 hardware line based on the program executed on the 68000 processor. The laboratory manual is in Appendix D.

REFERENCES

1. Donald Krantz, James Stanley. *68000 assembly language: techniques for building programs*. 2nd edition. University of Michigan. 2007

2. William D. Cramer, Gerry Kane. *68000 microprocessor handbook*. McGraw-Hill. 2nd edition. 2006

3. Joseph J. Carr. 68000 User's Manual. 1987

4. Contributor Motorola Staff . *MC 68000 16-bit Microprocessor: User's Manual*. Motorola : Prentice-Hall. 3rd edition. 1982

5. Walter A. Triebel, Avtar Singh. *The 68000 and 68020 microprocessors: hardware, software, and interfacing techniques.* Prentice Hall, University of Michigan. 1991

6. Walter A. Triebel, Avtar Singh. *The 68000 microprocessor: architecture, software, and interfacing techniques.* Prentice Hall, University of Michigan. 1986

7. *User manual for the Logic Analyser.*1989

APPENDIX A

Program for Part 3 of the Laboratory Manual

*A 68000 program that control the 8 bit LED Display Pattern Control By using Switch SW1, on *the Application board.

```
PACR        EQU         $80000D
PADDR       EQU         $800005
PADR        EQU         $800011
PBCR        EQU         $80000F
PBDDR       EQU         $800007
PBDR        EQU         $800013

            ORG         $400400

START       MOVE.B      #$80,PACR
            MOVE.B      #$00,PADDR      ;SET PORT A AS INPUT
            MOVE.B      #$80,PBCR
            MOVE.B      #$FF,PBDDR      ;SET PORT B AS OUTPUT

READAGA     MOVE.B      PADR,D0         ;JSR   READPA

            BTST        #7,D0
            BNE         QUIT
            CMP.B       #1,D0
            BNE         NO3

            MOVE.B      #$AA,D0
            MOVE.B      D0,PBDR         ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11             ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES16   MOVE.B      #8,D0           ;BACKSPACE ASCII CHARACTER
            TRAP        #11             ;SENT TO PORT A
            DC.B        0,2             ;(HYPERTERMINAL)
            DBF         D7,S5TIMES16    ;5 TIMES

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP16       NOP
            SUB.L       #1,D0
```

```
            BNE         TOP16
            MOVE.L      (A7)+,D0
            MOVE.B      #$55,D0

            MOVE.B      D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11           ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES15   MOVE.B      #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP        #11           ;SENT TO PORT A
            DC.B        0,2           ;(HYPERTERMINAL)
            DBF         D7,S5TIMES15     ;5 TIMES

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP15       NOP
            SUB.L       #1,D0
            BNE         TOP15
            MOVE.L      (A7)+,D0
            BRA         READAGA

NO3         MOVE.B      #$01,D0
            MOVE.B      D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11           ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES14   MOVE.B      #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP        #11           ;SENT TO PORT A
            DC.B        0,2           ;(HYPERTERMINAL)
            DBF         D7,S5TIMES14     ;5 TIMES

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP14       NOP
            SUB.L       #1,D0
            BNE         TOP14
            MOVE.L      (A7)+,D0

            MOVE.B      #$02,D0
            MOVE.B      D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11           ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES13   MOVE.B      #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP        #11           ;SENT TO PORT A
```

```
            DC.B    0,2         ;(HYPERTERMINAL)
            DBF     D7,S5TIMES13    ;5 TIMES

            MOVE.L  D0,-(A7)
            MOVE.L  #$1FFF,D0
TOP13       NOP
            SUB.L   #1,D0
            BNE     TOP13
            MOVE.L  (A7)+,D0

            MOVE.B  #$04,D0
            MOVE.B  D0,PBDR     ;DISPLAY HEX VALUE OF DO.B
            TRAP    #11         ;AT PORT A (HYPERTERMINAL)
            DC.B    0,9
            MOVE    #4,D7
S5TIMES12   MOVE.B  #8,D0       ;BACKSPACE ASCII CHARACTER
            TRAP    #11         ;SENT TO PORT A
            DC.B    0,2         ;(HYPERTERMINAL)
            DBF     D7,S5TIMES12    ;5 TIMES

            MOVE.L  D0,-(A7)
            MOVE.L  #$1FFF,D0
TOP12       NOP
            SUB.L   #1,D0
            BNE     TOP12
            MOVE.L  (A7)+,D0

            MOVE.B  #$08,D0
            MOVE.B  D0,PBDR     ;DISPLAY HEX VALUE OF DO.B
            TRAP    #11         ;AT PORT A (HYPERTERMINAL)
            DC.B    0,9
            MOVE    #4,D7
S5TIMES11   MOVE.B  #8,D0       ;BACKSPACE ASCII CHARACTER
            TRAP    #11         ;SENT TO PORT A
            DC.B    0,2         ;(HYPERTERMINAL)
            DBF     D7,S5TIMES11    ;5 TIMES

            MOVE.L  D0,-(A7)
            MOVE.L  #$1FFF,D0
TOP11       NOP
            SUB.L   #1,D0
            BNE     TOP11
            MOVE.L  (A7)+,D0

            MOVE.B  #$10,D0
```

```
            MOVE.B    D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP      #11           ;AT PORT A (HYPERTERMINAL)
            DC.B      0,9
            MOVE      #4,D7
S5TIMES10   MOVE.B    #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP      #11           ;SENT TO PORT A
            DC.B      0,2           ;(HYPERTERMINAL)
            DBF       D7,S5TIMES10     ;5 TIMES

            MOVE.L    D0,-(A7)
            MOVE.L    #$1FFF,D0
TOP10       NOP
            SUB.L     #1,D0
            BNE       TOP10
            MOVE.L    (A7)+,D0

            MOVE.B    #$20,D0
            MOVE.B    D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP      #11           ;AT PORT A (HYPERTERMINAL)
            DC.B      0,9
            MOVE      #4,D7
S5TIMES9    MOVE.B    #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP      #11           ;SENT TO PORT A
            DC.B      0,2           ;(HYPERTERMINAL)
            DBF       D7,S5TIMES9     ;5 TIMES

            MOVE.L    D0,-(A7)
            MOVE.L    #$1FFF,D0
TOP9        NOP
            SUB.L     #1,D0
            BNE       TOP9
            MOVE.L    (A7)+,D0

            MOVE.B    #$80,D0
            MOVE.B    D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP      #11           ;AT PORT A (HYPERTERMINAL)
            DC.B      0,9
            MOVE      #4,D7
S5TIMES8    MOVE.B    #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP      #11           ;SENT TO PORT A
            DC.B      0,2           ;(HYPERTERMINAL)
            DBF       D7,S5TIMES8     ;5 TIMES

            MOVE.L    D0,-(A7)
            MOVE.L    #$1FFF,D0
```

```
TOP8        NOP
            SUB.L       #1,D0
            BNE         TOP8
            MOVE.L      (A7)+,D0

            MOVE.B      #$40,D0
            MOVE.B      D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11           ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES7    MOVE.B      #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP        #11           ;SENT TO PORT A
            DC.B        0,2           ;(HYPERTERMINAL)
            DBF         D7,S5TIMES7     ;5 TIMES

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP7        NOP
            SUB.L       #1,D0
            BNE         TOP7
            MOVE.L      (A7)+,D0

            MOVE.B      #$20,D0
            MOVE.B      D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11           ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES6    MOVE.B      #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP        #11           ;SENT TO PORT A
            DC.B        0,2           ;(HYPERTERMINAL)
            DBF         D7,S5TIMES6     ;5 TIMES

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP6        NOP
            SUB.L       #1,D0
            BNE         TOP6
            MOVE.L      (A7)+,D0

            MOVE.B      #$10,D0
            MOVE.B      D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11           ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES5    MOVE.B      #8,D0          ;BACKSPACE ASCII CHARACTER
```

```
            TRAP        #11         ;SENT TO PORT A
            DC.B        0,2         ;(HYPERTERMINAL)
            DBF         D7,S5TIMES5     ;5 TIMES

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP5        NOP
            SUB.L       #1,D0
            BNE         TOP5
            MOVE.L      (A7)+,D0

            MOVE.B      #$08,D0
            MOVE.B      D0,PBDR         ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11         ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES4    MOVE.B      #8,D0         ;BACKSPACE ASCII CHARACTER
            TRAP        #11         ;SENT TO PORT A
            DC.B        0,2         ;(HYPERTERMINAL)
            DBF         D7,S5TIMES4     ;5 TIMES

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP4        NOP
            SUB.L       #1,D0
            BNE         TOP4
            MOVE.L      (A7)+,D0

            MOVE.B      #$04,D0
            MOVE.B      D0,PBDR         ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11         ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES3    MOVE.B      #8,D0         ;BACKSPACE ASCII CHARACTER
            TRAP        #11         ;SENT TO PORT A
            DC.B        0,2         ;(HYPERTERMINAL)
            DBF         D7,S5TIMES3     ;5 TIMES

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP3        NOP
            SUB.L       #1,D0
            BNE         TOP3
            MOVE.L      (A7)+,D0
```

```
            MOVE.B    #$02,D0
            MOVE.B    D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP      #11           ;AT PORT A (HYPERTERMINAL)
            DC.B      0,9
            MOVE      #4,D7
S5TIMES2    MOVE.B    #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP      #11           ;SENT TO PORT A
            DC.B      0,2           ;(HYPERTERMINAL)
            DBF       D7,S5TIMES2   ;5 TIMES

            MOVE.L    D0,-(A7)
            MOVE.L    #$1FFF,D0
TOP2        NOP
            SUB.L     #1,D0
            BNE       TOP2
            MOVE.L    (A7)+,D0

            MOVE.B    #$01,D0
            MOVE.B    D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP      #11           ;AT PORT A (HYPERTERMINAL)
            DC.B      0,9
            MOVE      #4,D7
S5TIMES1    MOVE.B    #8,D0          ;BACKSPACE ASCII CHARACTER
            TRAP      #11           ;SENT TO PORT A
            DC.B      0,2           ;(HYPERTERMINAL)
            DBF       D7,S5TIMES1   ;5 TIMES

            MOVE.L    D0,-(A7)
            MOVE.L    #$1FFF,D0
TOP1        NOP
            SUB.L     #1,D0
            BNE       TOP1
            MOVE.L    (A7)+,D0

            BRA       READAGA

NO7         CMP.B     #7,D0
            BNE       READAGA

            MOVE.B    #$FF,D0
            MOVE.B    D0,PBDR       ;DISPLAY HEX VALUE OF DO.B
            TRAP      #11           ;AT PORT A (HYPERTERMINAL)
            DC.B      0,9
            MOVE      #4,D7
S5TIMES18   MOVE.B    #8,D0          ;BACKSPACE ASCII CHARACTER
```

```
            TRAP        #11         ;SENT TO PORT A
            DC.B        0,2         ;(HYPERTERMINAL)
            DBF         D7,S5TIMES18    ;5 TIMES

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP18       NOP
            SUB.L       #1,D0
            BNE         TOP18
            MOVE.L      (A7)+,D0

            MOVE.B      #$00,D0
            MOVE.B      D0,PBDR     ;DISPLAY HEX VALUE OF DO.B
            TRAP        #11         ;AT PORT A (HYPERTERMINAL)
            DC.B        0,9
            MOVE        #4,D7
S5TIMES17   MOVE.B      #8,D0       ;BACKSPACE ASCII CHARACTER
            TRAP        #11         ;SENT TO PORT A
            DC.B        0,2         ;(HYPERTERMINAL)
            DBF         D7,S5TIMES17    ;5 TIMESJSR     WRITEPB

            MOVE.L      D0,-(A7)
            MOVE.L      #$1FFF,D0
TOP17       NOP
            SUB.L       #1,D0
            BNE         TOP17
            MOVE.L      (A7)+,D0

            BRA         READAGA
QUIT        TRAP        #11
            DC.B        0,0


*~Font name~Courier New~
*~Font size~10~
*~Tab type~1~
*~Tab size~8~
```

APPENDIX B

Listing File for Part 3 of the Laboratory Manual

PAGE 001 aimilong.asm Mon Apr 13 10:40:39 2009

```
 3 0080000D                          PACR      EQU       $80000D
 4 00800005                          PADDR     EQU       $800005
 5 00800011                          PADR      EQU       $800011
 6 0080000F                          PBCR      EQU       $80000F
 7 00800007                          PBDDR     EQU       $800007
 8 00800013                          PBDR      EQU       $800013
 9
10 00400400                                    ORG       $400400
11
12 00400400 13FC00800080000D  START   MOVE.B    #$80,PACR
13 00400408 13FC000000800005          MOVE.B    #$00,PADDR
14 00400410 13FC00800080000F          MOVE.B    #$80,PBCR
15 00400418 13FC00FF00800007          MOVE.B    #$FF,PBDDR
16
17 00400420 103900800011      READAGA MOVE.B    PADR,
18
19 00400426 08000007                  BTST      #7,D0
20 0040042A 660003A0                  BNE       QUIT
21 0040042E 0C000001                  CMP.B     #1,D0
22 00400432 66000068                  BNE       NO3
23
24 00400436 103C00AA                  MOVE.B    #$AA,D0
25 0040043A 13C000800013              MOVE.B    D0,PBDR
26 00400440 4E4B                      TRAP      #11
27 00400442 0009                      DC.B      0,9
28 00400444 3E3C0004                  MOVE      #4,D7
29 00400448 103C0008          S5TIMES16 MOVE.B  #8,D0
30 0040044C 4E4B                      TRAP      #11
31 0040044E 0002                      DC.B      0,2
32 00400450 51CFFFF6                  DBF       D7,S5TIMES16
33
34 00400454 2F00                      MOVE.L    D0,-(A7)
35 00400456 203C00001FFF              MOVE.L    #$1FFF,D0
36 0040045C 4E71              TOP16   NOP
37 0040045E 048000000001              SUB.L     #1,D0
38 00400464 66F6                      BNE       TOP16
39 00400466 201F                      MOVE.L    (A7)+,D0
```

```
40 00400468 103C0055                MOVE.B    #$55,D0
41
42 0040046C 13C000800013            MOVE.B    D0,PBDR
43 00400472 4E4B                    TRAP      #11
44 00400474 0009                    DC.B      0,9
45 00400476 3E3C0004                MOVE      #4,D7
46 0040047A 103C0008      S5TIMES15 MOVE.B    #8,D0
47 0040047E 4E4B                    TRAP      #11
48 00400480 0002                    DC.B      0,2
49 00400482 51CFFFF6                DBF       D7,S5TIMES15
50
51 00400486 2F00                    MOVE.L    D0,-(A7)
52 00400488 203C00001FFF            MOVE.L    #$1FFF,D0
53 0040048E 4E71          TOP15     NOP
54 00400490 048000000001            SUB.L     #1,D0
55 00400496 66F6                    BNE       TOP15
56 00400498 201F                    MOVE.L    (A7)+,D0
57 0040049A 6084                    BRA       READAGA
58
59 0040049C 103C0001      NO3       MOVE.B    #$01,D0
60 004004A0 13C000800013            MOVE.B    D0,PBDR
61 004004A6 4E4B                    TRAP      #11
62 004004A8 0009                    DC.B      0,9
63 004004AA 3E3C0004                MOVE      #4,D7
```

```
 64 004004AE 103C0008          S5TIMES14  MOVE.B   #8,D0
 65 004004B2 4E4B                         TRAP     #11
 66 004004B4 0002                         DC.B     0,2
 67 004004B6 51CFFFF6                     DBF      D7,S5TIMES14
 68
 69 004004BA 2F00                         MOVE.L   D0,-(A7)
 70 004004BC 203C00001FFF                 MOVE.L   #$1FFF,D0
 71 004004C2 4E71               TOP14     NOP
 72 004004C4 048000000001                 SUB.L    #1,D0
 73 004004CA 66F6                         BNE      TOP14
 74 004004CC 201F                         MOVE.L   (A7)+,D0
 75
 76 004004CE 103C0002                     MOVE.B   #$02,D0
 77 004004D2 13C000800013                 MOVE.B   D0,PBDR
 78 004004D8 4E4B                         TRAP     #11
 79 004004DA 0009                         DC.B     0,9
 80 004004DC 3E3C0004                     MOVE     #4,D7
 81 004004E0 103C0008          S5TIMES13  MOVE.B   #8,D0
 82 004004E4 4E4B                         TRAP     #11
 83 004004E6 0002                         DC.B     0,2
 84 004004E8 51CFFFF6                     DBF      D7,S5TIMES13
 85
 86 004004EC 2F00                         MOVE.L   D0,-(A7)
 87 004004EE 203C00001FFF                 MOVE.L   #$1FFF,D0
 88 004004F4 4E71               TOP13     NOP
 89 004004F6 048000000001                 SUB.L    #1,D0
 90 004004FC 66F6                         BNE      TOP13
 91 004004FE 201F                         MOVE.L   (A7)+,D0
 92
 93 00400500 103C0004                     MOVE.B   #$04,D0
 94 00400504 13C000800013                 MOVE.B   D0,PBDR
 95 0040050A 4E4B                         TRAP     #11
 96 0040050C 0009                         DC.B     0,9
 97 0040050E 3E3C0004                     MOVE     #4,D7
 98 00400512 103C0008          S5TIMES12  MOVE.B   #8,D0
 99 00400516 4E4B                         TRAP     #11
100 00400518 0002                         DC.B     0,2
101 0040051A 51CFFFF6                     DBF      D7,S5TIMES12
102
103 0040051E 2F00                         MOVE.L   D0,-(A7)
104 00400520 203C00001FFF                 MOVE.L   #$1FFF,D0
105 00400526 4E71               TOP12     NOP
```

```
106 00400528 048000000001                SUB.L     #1,D0
107 0040052E 66F6                        BNE       TOP12
108 00400530 201F                        MOVE.L    (A7)+,D0
109
110 00400532 103C0008                    MOVE.B    #$08,D0
111 00400536 13C000800013                MOVE.B    D0,PBDR
112 0040053C 4E4B                        TRAP      #11
113 0040053E 0009                        DC.B      0,9
114 00400540 3E3C0004                    MOVE      #4,D7
115 00400544 103C0008          S5TIMES11 MOVE.B    #8,D0
116 00400548 4E4B                        TRAP      #11
117 0040054A 0002                        DC.B      0,2
118 0040054C 51CFFFF6                    DBF       D7,S5TIMES11
119
120 00400550 2F00                        MOVE.L    D0,-(A7)
121 00400552 203C00001FFF                MOVE.L    #$1FFF,D0
122 00400558 4E71              TOP11     NOP
123 0040055A 048000000001                SUB.L     #1,D0
124 00400560 66F6                        BNE       TOP11
125 00400562 201F                        MOVE.L    (A7)+,D0
126
```

```
127 00400564 103C0010                MOVE.B    #$10,D0
128 00400568 13C000800013            MOVE.B    D0,PBDR
129 0040056E 4E4B                    TRAP      #11
130 00400570 0009                    DC.B      0,9
131 00400572 3E3C0004                MOVE      #4,D7
132 00400576 103C0008      S5TIMES10 MOVE.B    #8,D0
133 0040057A 4E4B                    TRAP      #11
134 0040057C 0002                    DC.B      0,2
135 0040057E 51CFFFF6                DBF       D7,S5TIMES10
136
137 00400582 2F00                    MOVE.L    D0,-(A7)
138 00400584 203C00001FFF            MOVE.L    #$1FFF,D0
139 0040058A 4E71          TOP10     NOP
140 0040058C 048000000001            SUB.L     #1,D0
141 00400592 66F6                    BNE       TOP10
142 00400594 201F                    MOVE.L    (A7)+,D0
143
144 00400596 103C0020                MOVE.B    #$20,D0
145 0040059A 13C000800013            MOVE.B    D0,PBDR
146 004005A0 4E4B                    TRAP      #11
147 004005A2 0009                    DC.B      0,9
148 004005A4 3E3C0004                MOVE      #4,D7
149 004005A8 103C0008      S5TIMES9  MOVE.B    #8,D0
150 004005AC 4E4B                    TRAP      #11
151 004005AE 0002                    DC.B      0,2
152 004005B0 51CFFFF6                DBF       D7,S5TIMES9
153
154 004005B4 2F00                    MOVE.L    D0,-(A7)
155 004005B6 203C00001FFF            MOVE.L    #$1FFF,D0
156 004005BC 4E71          TOP9      NOP
157 004005BE 048000000001            SUB.L     #1,D0
158 004005C4 66F6                    BNE       TOP9
159 004005C6 201F                    MOVE.L    (A7)+,D0
160
161 004005C8 103C0080                MOVE.B    #$80,D0
162 004005CC 13C000800013            MOVE.B    D0,PBDR
163 004005D2 4E4B                    TRAP      #11
164 004005D4 0009                    DC.B      0,9
165 004005D6 3E3C0004                MOVE      #4,D7
166 004005DA 103C0008      S5TIMES8  MOVE.B    #8,D0
167 004005DE 4E4B                    TRAP      #11
168 004005E0 0002                    DC.B      0,2
```

```
169 004005E2 51CFFFF6                DBF       D7,S5TIMES8
170
171 004005E6 2F00                    MOVE.L    D0,-(A7)
172 004005E8 203C00001FFF            MOVE.L    #$1FFF,D0
173 004005EE 4E71          TOP8      NOP
174 004005F0 048000000001            SUB.L     #1,D0
175 004005F6 66F6                    BNE       TOP8
176 004005F8 201F                    MOVE.L    (A7)+,D0
177
178 004005FA 103C0040                MOVE.B    #$40,D0
179 004005FE 13C000800013            MOVE.B    D0,PBDR
180 00400604 4E4B                    TRAP      #11
181 00400606 0009                    DC.B      0,9
182 00400608 3E3C0004                MOVE      #4,D7
183 0040060C 103C0008      S5TIMES7  MOVE.B    #8,D0
184 00400610 4E4B                    TRAP      #11
185 00400612 0002                    DC.B      0,2
186 00400614 51CFFFF6                DBF       D7,S5TIMES7
187
188 00400618 2F00                    MOVE.L    D0,-(A7)
189 0040061A 203C00001FFF            MOVE.L    #$1FFF,D0
```

```
190 00400620 4E71            TOP7       NOP
191 00400622 048000000001               SUB.L     #1,D0
192 00400628 66F6                       BNE       TOP7
193 0040062A 201F                       MOVE.L    (A7)+,D0
194
195 0040062C 103C0020                   MOVE.B    #$20,D0
196 00400630 13C000800013               MOVE.B    D0,PBDR
197 00400636 4E4B                       TRAP      #11
198 00400638 0009                       DC.B      0,9
199 0040063A 3E3C0004                   MOVE      #4,D7
200 0040063E 103C0008        S5TIMES6   MOVE.B    #8,D0
201 00400642 4E4B                       TRAP      #11
202 00400644 0002                       DC.B      0,2
203 00400646 51CFFFF6                   DBF       D7,S5TIMES6
204
205 0040064A 2F00                       MOVE.L    D0,-(A7)
206 0040064C 203C00001FFF               MOVE.L    #$1FFF,D0
207 00400652 4E71            TOP6       NOP
208 00400654 048000000001               SUB.L     #1,D0
209 0040065A 66F6                       BNE       TOP6
210 0040065C 201F                       MOVE.L    (A7)+,D0
211
212 0040065E 103C0010                   MOVE.B    #$10,D0
213 00400662 13C000800013               MOVE.B    D0,PBDR
214 00400668 4E4B                       TRAP      #11
215 0040066A 0009                       DC.B      0,9
216 0040066C 3E3C0004                   MOVE      #4,D7
217 00400670 103C0008        S5TIMES5   MOVE.B    #8,D0
218 00400674 4E4B                       TRAP      #11
219 00400676 0002                       DC.B      0,2
220 00400678 51CFFFF6                   DBF       D7,S5TIMES5
221
222 0040067C 2F00                       MOVE.L    D0,-(A7)
223 0040067E 203C00001FFF               MOVE.L    #$1FFF,D0
224 00400684 4E71            TOP5       NOP
225 00400686 048000000001               SUB.L     #1,D0
226 0040068C 66F6                       BNE       TOP5
227 0040068E 201F                       MOVE.L    (A7)+,D0
228
229 00400690 103C0008                   MOVE.B    #$08,D0
230 00400694 13C000800013               MOVE.B    D0,PBDR
231 0040069A 4E4B                       TRAP      #11
```

```
232 0040069C 0009                       DC.B    0,9
233 0040069E 3E3C0004                   MOVE    #4,D7
234 004006A2 103C0008       S5TIMES4    MOVE.B  #8,D0
235 004006A6 4E4B                       TRAP    #11
236 004006A8 0002                       DC.B    0,2
237 004006AA 51CFFFF6                   DBF     D7,S5TIMES4
238
239 004006AE 2F00                       MOVE.L  D0,-(A7)
240 004006B0 203C00001FFF               MOVE.L  #$1FFF,D0
241 004006B6 4E71           TOP4        NOP
242 004006B8 048000000001               SUB.L   #1,D0
243 004006BE 66F6                       BNE     TOP4
244 004006C0 201F                       MOVE.L  (A7)+,D0
245
246 004006C2 103C0004                   MOVE.B  #$04,D0
247 004006C6 13C000800013               MOVE.B  D0,PBDR
248 004006CC 4E4B                       TRAP    #11
249 004006CE 0009                       DC.B    0,9
250 004006D0 3E3C0004                   MOVE    #4,D7
251 004006D4 103C0008       S5TIMES3    MOVE.B  #8,D0
252 004006D8 4E4B                       TRAP    #11
```

```
253 004006DA 0002                       DC.B    0,2
254 004006DC 51CFFFF6                   DBF     D7,S5TIMES3
255
256 004006E0 2F00                       MOVE.L  D0,-(A7)
257 004006E2 203C00001FFF               MOVE.L  #$1FFF,D0
258 004006E8 4E71           TOP3        NOP
259 004006EA 048000000001               SUB.L   #1,D0
260 004006F0 66F6                       BNE     TOP3
261 004006F2 201F                       MOVE.L  (A7)+,D0
262
263 004006F4 103C0002                   MOVE.B  #$02,D0
264 004006F8 13C000800013               MOVE.B  D0,PBDR
265 004006FE 4E4B                       TRAP    #11
266 00400700 0009                       DC.B    0,9
267 00400702 3E3C0004                   MOVE    #4,D7
268 00400706 103C0008       S5TIMES2    MOVE.B  #8,D0
269 0040070A 4E4B                       TRAP    #11
270 0040070C 0002                       DC.B    0,2
271 0040070E 51CFFFF6                   DBF     D7,S5TIMES2
272
273 00400712 2F00                       MOVE.L  D0,-(A7)
274 00400714 203C00001FFF               MOVE.L  #$1FFF,D0
275 0040071A 4E71           TOP2        NOP
276 0040071C 048000000001               SUB.L   #1,D0
277 00400722 66F6                       BNE     TOP2
278 00400724 201F                       MOVE.L  (A7)+,D0
279
280 00400726 103C0001                   MOVE.B  #$01,D0
281 0040072A 13C000800013               MOVE.B  D0,PBDR
282 00400730 4E4B                       TRAP    #11
283 00400732 0009                       DC.B    0,9
284 00400734 3E3C0004                   MOVE    #4,D7
285 00400738 103C0008       S5TIMES1    MOVE.B  #8,D0
286 0040073C 4E4B                       TRAP    #11
287 0040073E 0002                       DC.B    0,2
288 00400740 51CFFFF6                   DBF     D7,S5TIMES1
289
290 00400744 2F00                       MOVE.L  D0,-(A7)
291 00400746 203C00001FFF               MOVE.L  #$1FFF,D0
292 0040074C 4E71           TOP1        NOP
293 0040074E 048000000001               SUB.L   #1,D0
294 00400754 66F6                       BNE     TOP1
```

```
295 00400756 201F                       MOVE.L      (A7)+,D0
296
297 00400758 6000FCC6                   BRA         READAGA
298
299 0040075C 0C000007       NO7         CMP.B       #7,D0
300 00400760 6600FCBE                   BNE         READAGA
301
302 00400764 103C00FF                   MOVE.B      #$FF,D0
303 00400768 13C000800013               MOVE.B      D0,PBDR
304 0040076E 4E4B                       TRAP        #11
305 00400770 0009                       DC.B        0,9
306 00400772 3E3C0004                   MOVE        #4,D7
307 00400776 103C0008       S5TIMES18   MOVE.B      #8,D0
308 0040077A 4E4B                       TRAP        #11
309 0040077C 0002                       DC.B        0,2
310 0040077E 51CFFFF6                   DBF         D7,S5TIMES18
311
312 00400782 2F00                       MOVE.L      D0,-(A7)
313 00400784 203C00001FFF               MOVE.L      #$1FFF,D0
314 0040078A 4E71           TOP18       NOP
315 0040078C 048000000001               SUB.L       #1,D0
```

```
 316 00400792 66F6                          BNE         TOP18
 317 00400794 201F                          MOVE.L      (A7)+,D0
 318
 319 00400796 103C0000                      MOVE.B      #$00,D0
 320 0040079A 13C000800013                  MOVE.B      D0,PBDR
 321 004007A0 4E4B                          TRAP        #11
 322 004007A2 0009                          DC.B        0,9
 323 004007A4 3E3C0004                      MOVE        #4,D7
 324 004007A8 103C0008          S5TIMES17   MOVE.B      #8,D0
 325 004007AC 4E4B                          TRAP        #11
 326 004007AE 0002                          DC.B        0,2
 327 004007B0 51CFFFF6                      DBF         D7,S5TIMES17
 328
 329 004007B4 2F00                          MOVE.L      D0,-(A7)
 330 004007B6 203C00001FFF                  MOVE.L      #$1FFF,D0
 331 004007BC 4E71                          TOP17       NOP
 332 004007BE 048000000001                  SUB.L       #1,D0
 333 004007C4 66F6                          BNE         TOP17
 334 004007C6 201F                          MOVE.L      (A7)+,D0
 335
 336 004007C8 6000FC56                      BRA         READAGA
 337 004007CC 4E4B              QUIT        TRAP        #11
 338 004007CE 0000                          DC.B        0,0
 339
 340
 341
 342                    *~Font name~Courier New~
 343                    *~Font size~10~
 344                    *~Tab type~1~
 345                    *~Tab size~8~
Assembly complete
Bytes filed: 976
0 errors
0 warnings
```

APPENDIX C

Program for Part 4 of the Laboratory Manual

```
*A 68000 program that control the 8 bit LED Display

PACR      EQU       $80000D
PADDR     EQU       $800005
PADR      EQU       $800011
PBCR      EQU       $80000F
PBDDR     EQU       $800007
PBDR      EQU       $800013

          bin
          ORG       0
          DC.L      $4003F0
          DC.L      START
          DS.B      $400-8

          ORG       $400

START     MOVEA.L   #$4003F0,A7
          MOVE.B    #$80,PACR
          MOVE.B    #$00,PADDR     ;SET PORT A AS INPUT
          MOVE.B    #$80,PBCR
          MOVE.B    #$FF,PBDDR     ;SET PORT B AS OUTPUT

READAGA   MOVE.B    PADR,D0  ;PORT A (INPUT-SWITCH) TO 'D0'
          EOR       #%11111111,D0
          MOVE.B    D0,PBDR       ;'D0' TO PORT B (OUTPUT-LED)

          JMP       READAGA

here      DS.B      $8000-*
          END
```

APPENDIX D

Lab Manual for the Logic Analyser : 68000 Microprocessor Board as the Test Hardware

OBJECTIVE:

1. To learn how to use the Logic Analyser.
2. To study the operation of the 68000 Microprocessor Chip.

EQUIPMENT:

1. Logic Analyser.
2. Disassembler POD for 68000.
3. 68000 Microprocessor board.
4. Flight 68000 application board.
5. EPROM 27c256

THEORY:

1. Logic Analyser

- A Logic Analyser is an electronic instrument that could display multiple digital signals on a single screen. They are typically used for capturing data in digital systems that have too many channels to be examined with an oscilloscope. Software running on the Logic Analyser can convert the captured data into timing diagrams, protocol decodes, state machine traces, assembly language, or correlate assembly with source-level software.

Figure 1 : Logic Analyser

2. Disassembler POD for 68000

- The most common method of data capture for logic analyzers is through a probe. A Logic Analyzer can measure anything electronic if it has the proper probe connected. Mostly Logic Analyzer measure data buses. The probes try to tap into the electronic signals being passed through a data bus or wire.

Figure 2 : Disassembler POD for 68000

INSTRUCTION:

Part 1 : Beginning

1. In this experiment we will use Logic Analyser to analyse the working of a 68000 Microprocessor.
2. Connect the 68000 Disassembler POD to the Logic Analyser. Make sure that the 68000 Disassembler POD is connecting correctly on MC68000 chip.

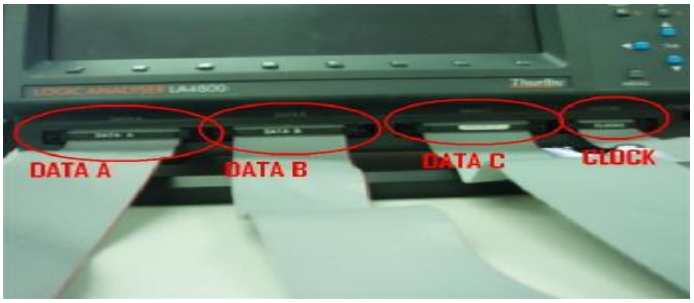


Figure 3: Connection to the Logic Analyser

3. Switch on the Logic Analyser then check the 68000 Disassembler POD was assembled or not.
4. Clip the 68000 Disassembler POD to the chip MC68000 on 68000 Microprocessor Board. Make sure the connection is correct (Pin 1 on DP 68000 to Pin 1 on chip).

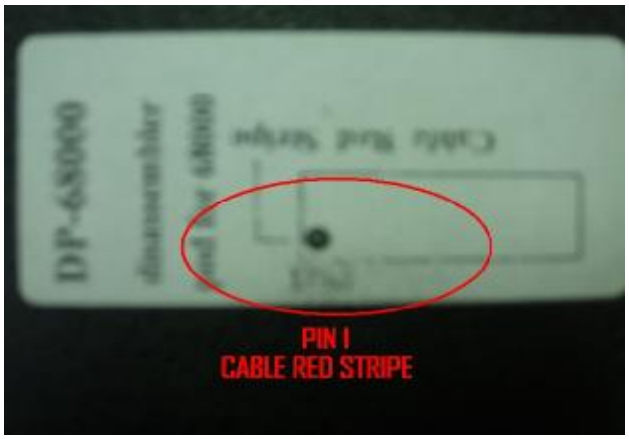


Figure 4: 68000 Disassembler POD

Figure 5: Connection between 68000 Disassembler POD to 68000 microprocessor

5. Press Confirm.

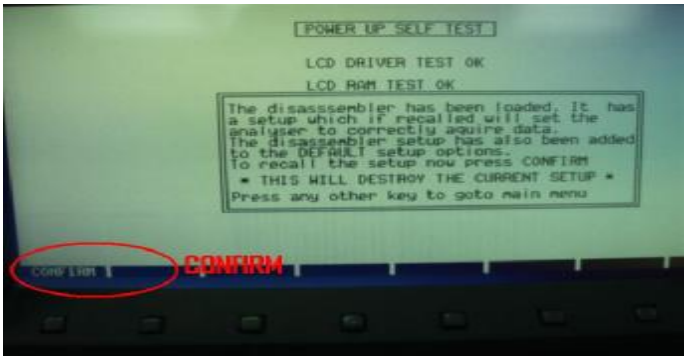


Figure 6: The disassembler has been loaded

Part 2 : Introduction for the Logic Analyser Using Reset Vector

Figure 7: Main Menu

1. Press 1 for CONFIGURATION.
   i. Set clock selected as EXTERNAL.
   ii. Press MENU.

2. Press 2 for TIMING DIAGRAM.
   i. Press FORMAT for change the label.
   ii. Press CLEAR to delete ADDR00 and rename to A00 (refer below).

ADDR00 **à** A00
ADDR01 **à** A01
ADDR02 **à** A02
ADDR03 **à** A03
⋮
ADDR15 **à** A15

   iii. Press the button on right side of Logic Analyser to rename the address.
   iv. Press tab cursor to continue change the label. Repeat step Part 2(2(ii) and 2(iii)).
   v. After finish labeling, press EXIT.
   vi. Press MENU.

3. Press 3 for STATE LISTING.
    i. Confirm that, no data on the Logic Analyser.
    ii. Press PAGE DEC (page decreases) or GOTO TRG (go to trigger) until it shows the ‘0000’ POS on the upper left of Logic Analyser display (Starting point it will be triggering). **POS is position.
    iii. Press MENU.

4. Press 4 for TRIGGER SETUP.
    i. On Address HEHEX, Trig Wrd, set as 000000 (Reset Vector).
    ii. Press MENU.

5. Press 7 for 68000 DISASSEMBLER VER (1.01)
    i. Press RUN button on right side of Logic Analyser.
    ii. Press SINGLE on the list menu on the bottom.

6. Switch on the Flight 68000 Board.
    i. Press reset button on 68000 Microprocessor Board.

7. Logic Analyser will be triggered the data and displayed to the screen.

Figure 8: Trigger Function

8. Answer the question in SECTION A.

Part 3 : Using Hyperterminal Trigger the Another Address (Program is in APPENDIX)

1. Type the program on the APPENDIX.

2. Assemble the program using command prompt to get .BIN file (refer to the lab manual Flight 68000).

3. Press MENU on the Logic Analyser.

4. Press 4 for TRIGGER SETUP.
    i. Change the Address HEHEX (should be HEX but this is the display in the Logic Analyser), Trig Wrd (means Word), to 400400 (Starting address for microprocessor).
    ii. Press MENU.

5. On 68000 Microprocessor Board.
    i. Switch on the 68000 Microprocessor Board.
    ii. Open Hyperterminal on the PC.

Figure 9: Hyperterminal (Connection Description Connect To)

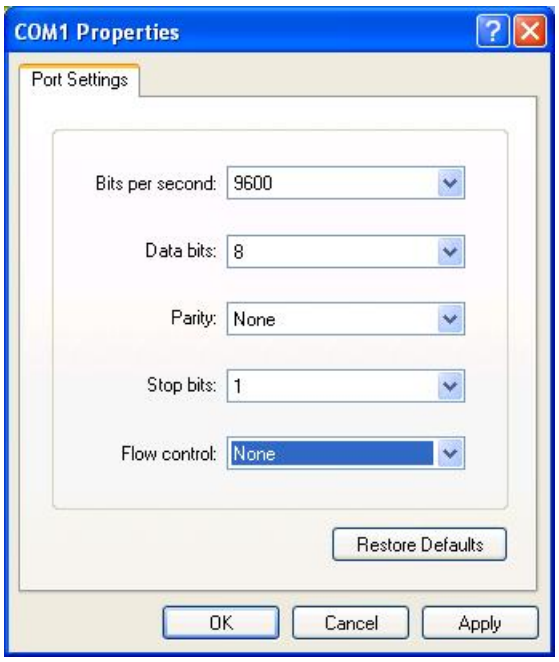


Figure 10: COM1 Properties

iii. Press reset on 68000 Microprocessor Board.
iv. Press ENTER (3 times) on keyboard. This figure will be entering (Figure 11).
v. Type ‘lt’ (Figure 12).

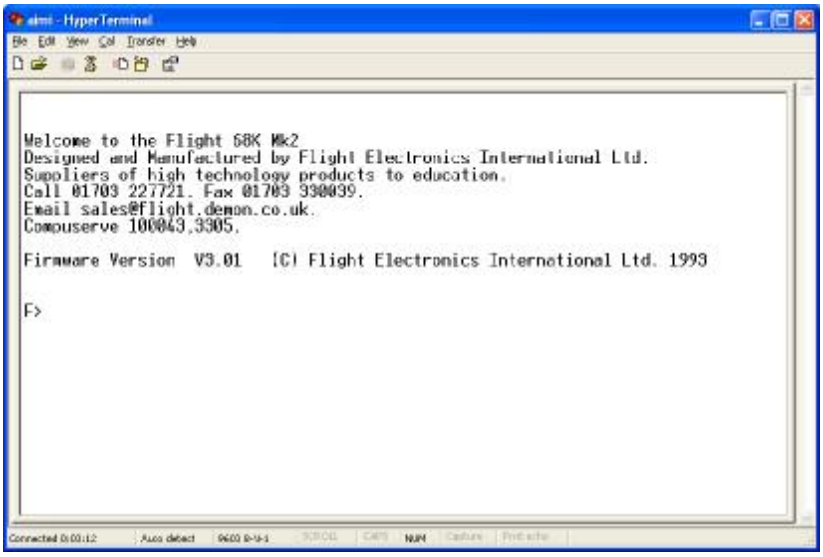


Figure 11: After press ENTER (3 times)

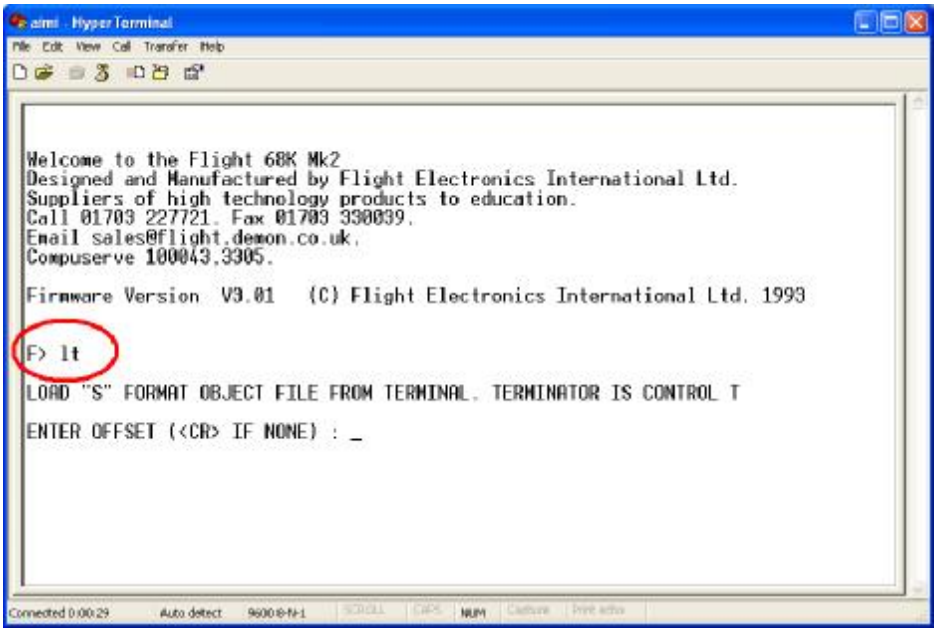


Figure 12: Type 'lt'

vi. Press ENTER after Hyperterminal show ENTER OFFSET (<CR IF NONE>):

vii. Click TRANSFER on the MENU, and then SEND TEXT FILE.

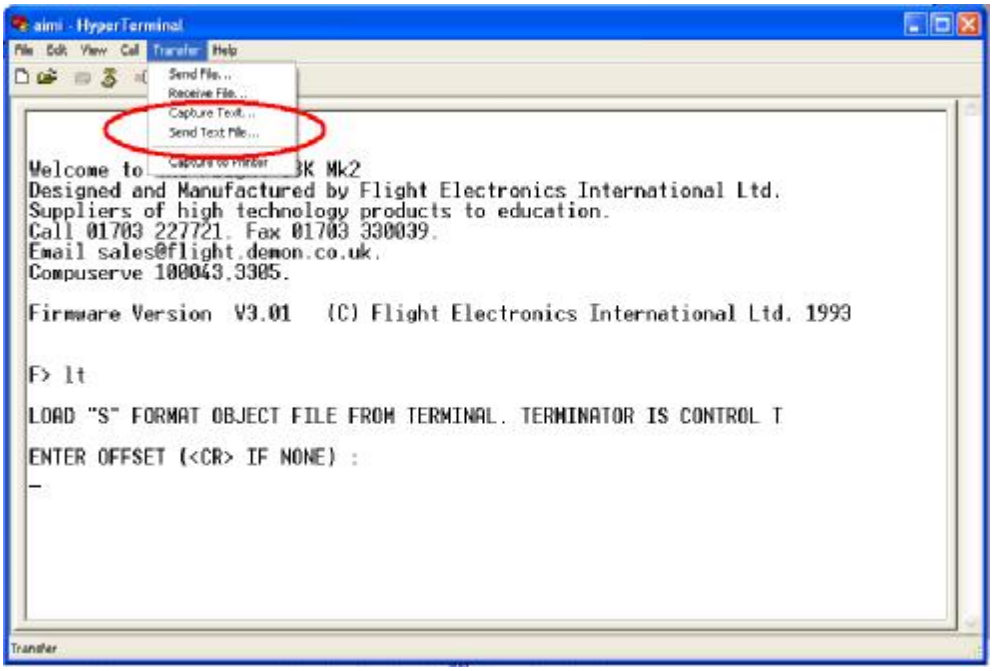


Figure 13: Load data .BIN to chip using Hyperterminal

viii. Search .BIN file that you create before.

ix. After LOAD COMPLETED, type ‘go’ and enter address 400400 (Figure 14 and 15).
x. Press ENTER.

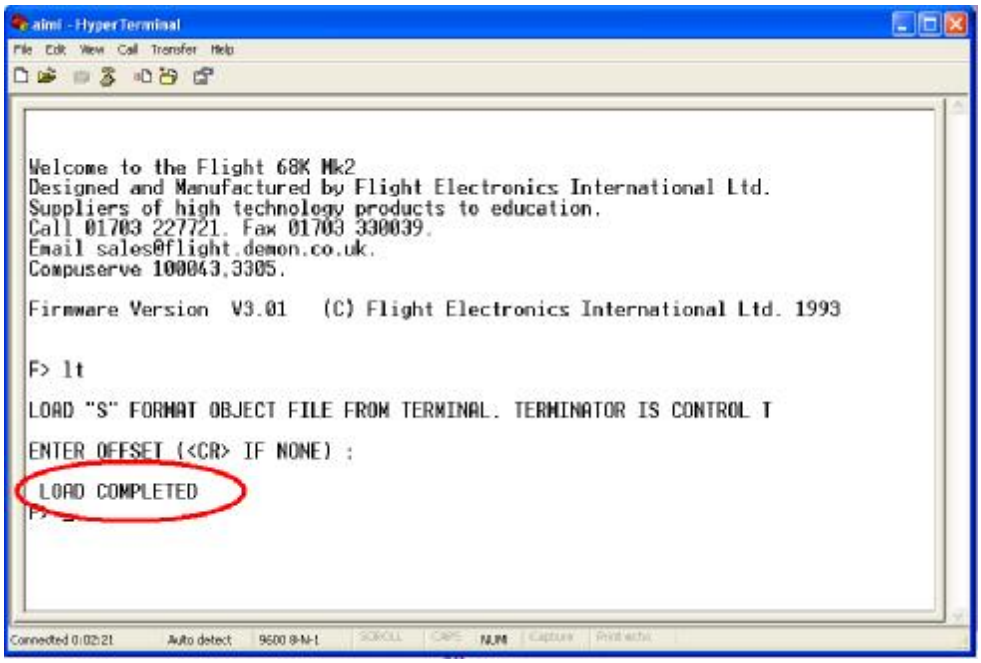


Figure 14: Load Completed

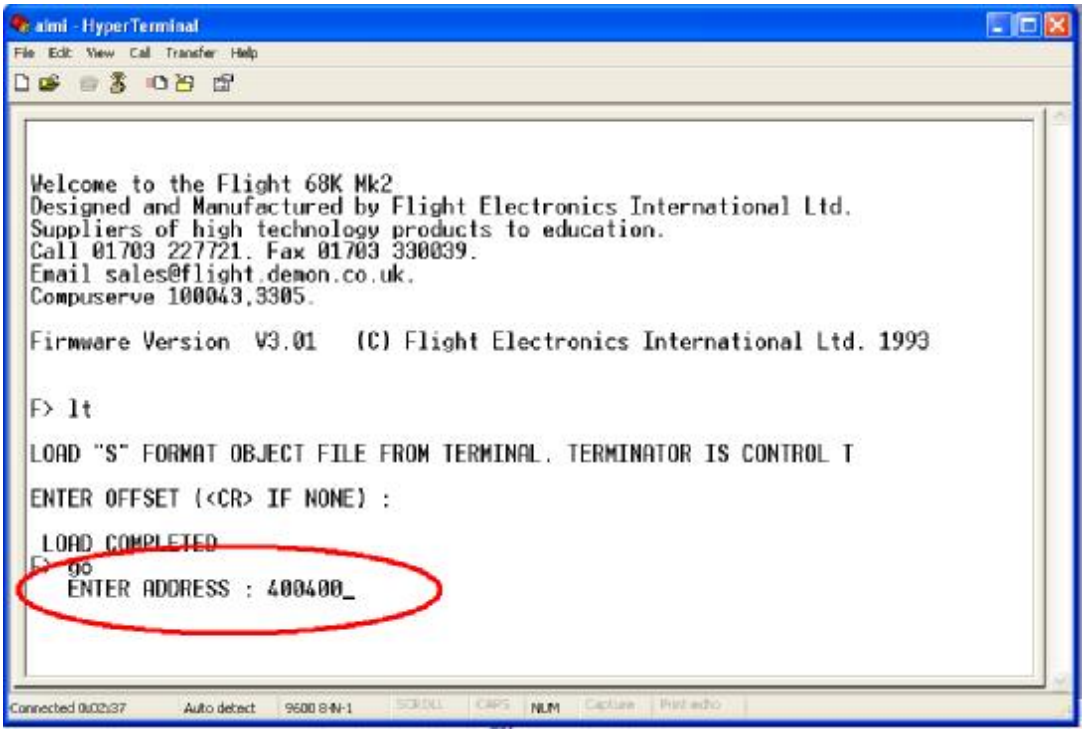


Figure 15: Type Address for Microprocesser

6. (Assuming you have press Menu as instructed above in Part 3(1)(ii)) Press 7 for 68000 DISASSEMBLER VER (1.01)
   i. Press RUN button on right side of Logic Analyser.
   ii. Press SINGLE on the list menu on the bottom.
   iii. Repeat procedure Part 3 (2(iii), 2(iv), 2(ix))

7. Logic Analyser will be triggered on the screen and it will get the data.

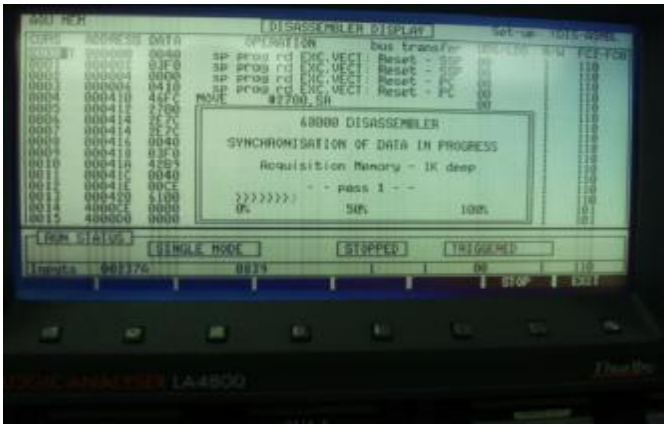


Figure 16: Trigger Function

8. Answer the question SECTION B.

Part 4 : Trigger the data directly from the EPROM

1. Switch off the Microprocessor Board. Replace the EPROM 27C256 on the 68000 Microprocessor Board to the another EPROM that was programmed. Below is the listing file of the program.

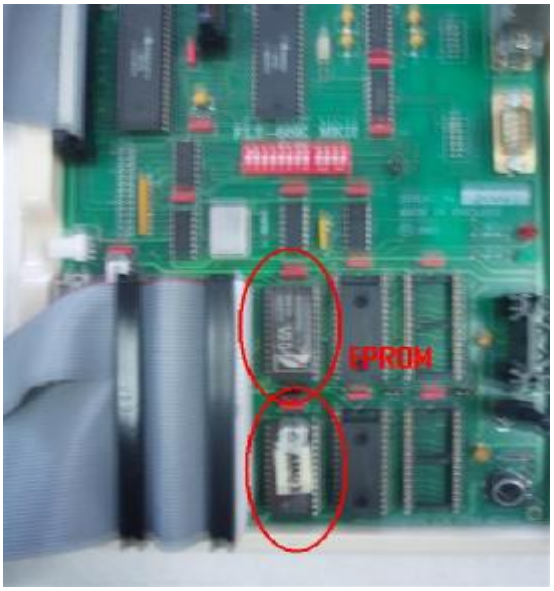


Figure 17: Replace the EPROM from the 68000 microprocessor board

```
PAGE  001  aimi2.asm                              Thu Apr 09 15:55:06 2009

    1                              *A 68000 program that control the 8 bit LED Display
    2
    3 0080000D                     PACR      EQU     $80000D
    4 00800005                     PADDR     EQU     $800005
    5 00800011                     PADR      EQU     $800011
    6 0080000F                     PBCR      EQU     $80000F
    7 00800007                     PBDDR     EQU     $800007
    8 00800013                     PBDR      EQU     $800013
    9
   10                                        bin
   11 00000000                               ORG     0
   12 00000000 004003F0                      DC.L    $4003F0
   13 00000004 00000400                      DC.L    START
   14 00000008                               DS.B    $400-8
   15
   16 00000400                               ORG     $400
   17
   18 00000400 2E7C004003F0       START     MOVEA.L #$4003F0,A7
   19 00000406 13FC008000800000D             MOVE.B  #$80,PACR
   20 0000040E 13FC000000800005              MOVE.B  #$00,PADDR        ;SET PORT A AS INPUT
   21 00000416 13FC008000800000F             MOVE.B  #$80,PBCR
   22 0000041E 13FC00FF00800007              MOVE.B  #$FF,PBDDR        ;SET PORT B AS OUTPUT
   23
   24 00000426 103900800011       READAGA   MOVE.B  PADR,D0           ;PORT A (INPUT-SWITCH) TO 'D0'
   25 0000042C 0A4000FF                      EOR     #%11111111,D0     ;COMPLIMENT THE SWITCH FUNCTION
   26 00000430 13C000800013                  MOVE.B  D0,PBDR           ;'D0' TO PORT B (OUTPUT-LED)
   27
   28 00000436 4EF80426                      JMP     READAGA
   29
   30 0000043A                    here      DS.B    $8000-*
   31                                        END
Assembly complete
Bytes filed: 32768
0 errors
0 warnings
```

Figure 18: Listing file

2. Press 4 for TRIGGER SETUP.
    i. Change the Address HEHEX, Trig Wrd, to 000000.
    ii. Press MENU.

3. Press 7 for 68000 DISASSEMBLER VER (1.01)
    i. Press RUN button on right side of Logic Analyser.
    ii. Press SINGLE on the list menu on the bottom.

4. On 68000 Microprocessor Board.
    i. Switch on the 68000 Microprocessor Board.
    ii. Press reset on 68000 Microprocessor Board.

5. Logic Analyser will be triggered the data and displayed to the screen.

Figure 19: Trigger Function

6. Check the data on Logic Analyser and answer the question on SECTION C.

QUESTIONS:

Section A

1. What the operation on cursor 0027?
   ANSWER: MOVEA.L 00000080,D0

2. On cursor 0028, write the Address, Data, Operation, Bus Transfer, UDS/LDS, R/W and FC2-FC0.
   ANSWER:

Cursor	Address	Data	Operation	Bus Transfer	UDS/LDS	R/W	FC2-FC0
0028	**A0000A**	**0000**	**sp data wr**	**-low byte**	**10**	**0**	**101**

Table 1: DISASSEMBLER DISPLAY

   a) What does the result of R/W signal signify?
      ANSWER: Write

   b) What does the result of UDS/LDS signal signify on cursor 0028.
      ANSWER: The data is on low byte. So, UDS is 1 and LDS is 0.

3. We can see the operation on cursor 0021 until 0023;

Cursor	Address	Data	Operation	Bus Transfer	UDS/LDS	R/W	FC2-FC0
0021	00223A	2C7C	MOVEA.L 00A00001,A6		00	1	110
0022	00223C	00A0	sp prog rd		00	1	110
0023	00223E	0001	sp prog rd		00	1	110

Table 2: DISASSEMBLER DISPLAY

   a) Refer to operation on cursor 0021, which one the destination?
      ANSWER: A6

   b) What is the operand?
      ANSWER: 00A00001

c) Explain the about the data on cursor 0022 and 0023.
ANSWER: For the first word from the operand (00A0), it will put on address 00223C and second word (0001), it will put on address 00223E.

d) How many bus cycles on that table?
ANSWER: 2 bus cycle

Section B

1. After Logic Analyser was triggered, what the DISASSEMBLER DISPLAY showed.
ANSWER:

Address	Data	Operation
400400	**13FC**	**MOVE.B #80,0080000D.L**
400402	**0080**	**sp prog rd**
400404	**0080**	**sp prog rd**
400406	**000D**	**sp prog rd**
400408	**13FC**	**MOVE.B #00,00800005.L**
****80000C**	**8080**	**sp prog wr**
40040A	**0000**	**sp prog rd**
40040C	**0080**	**sp prog rd**
40040E	**0005**	**sp prog rd**

Table 3: DISASSEMBLER DISPLAY **(**GLITCH)**

2. After Logic Analyser was triggered, what the STATE LISTING showed.
ANSWER:

Address	BERR	BGACK	UDS/LDS	R/W	FC2-FC0
400400	**1**	**1**	**00**	**1**	**110**
400402	**1**	**1**	**00**	**1**	**110**
400404	**1**	**1**	**00**	**1**	**110**
400406	**1**	**1**	**00**	**1**	**110**
400408	**1**	**1**	**00**	**1**	**110**
****80000C**	**1**	**1**	**10**	**0**	**101**
40040A	**1**	**1**	**00**	**1**	**110**
40040C	**1**	**1**	**00**	**1**	**110**
40040E	**1**	**1**	**00**	**1**	**110**

Table 4: STATE LISTING **(**GLITCH)**

3. Compared the listing file in Part 3 with DISASSEMBLER DISPLAY in Table 1. It is same? Why?

   ANSWER:

   The data on the Logic Analyser is the same when we compared with listing file because the data that was transferred on 68000 microprocessor chip is the same program.

4. Open the timing diagram and draw the FC2-FC0 (press group if not display).

   ANSWER:the timing diagram must be related each other**

5. What the relationship between timing diagram and Table 4. Explain.

   ANSWER:the timing diagram must be related each other**

6. Repeat the procedure in Part 3 (no need to begin the step 5, proceed to 6). Change Address HEHEX, Trig Wrd to 400408. After Logic Analyser was triggered, what the STATE LISTING showed.

   ANSWER:

Address	BERR	BGACK	UDS/LDS	R/W	FC2-FC0
400408	**13FC**	**MOVE.B #00,00800005.L**	**00**	**1**	**110**
****80000C**	**8080**	**sp prog wr**	**10**	**0**	**101**
40040A	**0000**	**sp prog rd**	**00**	**1**	**110**
40040C	**0080**	**sp prog rd**	**00**	**1**	**110**
40040E	**0005**	**sp prog rd**	**00**	**1**	**110**

Table 5: STATE LISTING

7. Compare the Table 5 and Table 4. Explain.

   ANSWER:It is the same thing because we run the same program.

Section C

1. Write the program that you can get from the Logic Analyser (from address 000400 until 000436).
   ANSWER:

```
MOVEA.L   #4003F0,A7
MOVE.B    #80,0080000D.L
MOVE.B    #00, 00800005.L
MOVE.B    #80, 0080000F.L
MOVE.B    #FF, 00800007.L
MOVE.B    00800011.L,D0
EORI.W    #00FF,D0
MOVE.B    D0, 00800013.L
JMP       0426.W
```

2. What is the benefit if we used the ROM that was programmed?
   ANSWER:
   - **No needs to use the PC for upload the program.**

3. Press the SW1 port 0, what happened to LED?
   ANSWER:
   - **LED 0 light-off**

4. Press the SW1 port 0 until port 7, what happened to LED?
   ANSWER:
   - **All LED light-off**

5. Rewrite the program will make LED to light-up when the switch is pressed.
   ANSWER:

```
MOVEA.L   #4003F0,A7
MOVE.B    #80,0080000D.L
MOVE.B    #00, 00800005.L
MOVE.B    #80, 0080000F.L
MOVE.B    #FF, 00800007.L
MOVE.B    00800011.L,D0
MOVE.B    D0, 00800013.L
JMP       0426.W
```