

DEVELOPMENT OF AN 8085
SOFTWARE DEBUGGING SYSTEM

BY

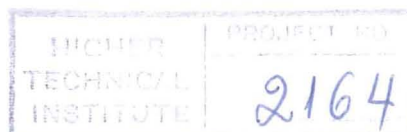
MARIA PAVLOU

PROJECT REPORT

Submitted to the Department of Electrical Engineering
of the Higher Technical Institute
Nicosia Cyprus
in partial of the fulfilment of the requirements
for the award the diploma of
TECHNICIAN ENGINEER in ELECTRICAL ENGINEERING

Project Supervisor:
Mr. Ch. Theopemptou

JUNE 1993



SUMMARY

PROJECT TITLE: DEVELOPMENT OF AN 8085 SOFTWARE REBUGGING SYSTEM

STUDENT: PAVLOU MARIA STAVROU

SUPERVISOR: CH. THEOPEMPTOU

The aims of this project are:

- To investigate the requirements of a PC based 8085 debugging system, in terms of 8085 software and interfacing signals and PC software.
- To investigate the kind of commands and user interface that should be provided and the language to be used for software development.
- To develop and test the required 8085 monitor software that would be suitable in providing all the required facilities to the PC based software
- To design and test the IBM PC interrupt driven interface software for serial communications.
- To develop and test the software interface that will communicate with the hardware.
- To provide the typical commands normally found in software debuggers.

Required Facilities

1. RS-232C Communication should be done reliably
2. The debugging commands should offer the following functions:
 - a. Perform standard debugging (normally found in software debuggers) with the PC Memory and/or the 8085 board Memory.
 - b. These debugging facilities are the following:

- Compare the contents of memory locations
- Copy memory locations from Board Memory into PC memory
- Display (dump) memory locations
- Enter byte values into memory locations
- Fill a range of addresses with one or more bytes
- Execute the program
- Perform hex addition and subtraction
- Load file into memory (PC memory or board memory)
- Move addresses
- Quit the program
- Display change the content of registers
- Search for certain bytes
- Save the file on a disk file (a file being debugged in either PC or board memory)

CONTENTS

	<u>PAGE</u>
ACKNOWLEDGEMENTS	i
SUMMARY	ii
INTRODUCTION	iv
<u>CHAPTER 1</u> MICROTARGET BOARD	1
1.1 Applications	1
1.2 Construction	2
1.3 Specifications	3
1.4 Memory Map and I/O decoding	3
1.5 Connections	3
1.6 Target expansion	5
1.7 Circuit diagram of MTB85	5a
<u>CHAPTER 2</u> CONVERTER CIRCUIT FOR RS232C SERIAL PORT ..	7
<u>CHAPTER 3</u> THE MAGIC WORLD OF SERIAL COMMUNICATION .	8
3.1 Basic Concepts in serial I/O	8
3.2 Interfacing Requirements	8
3.3 Alphanumeric Codes	8
3.4 Transmission Format	9
3.4.1 Comparing Synchronous and Asynchronous Transmission	9
3.4.2 Rate of Transmission (Baud Rate)	10
3.4.3 Error Checks in data Communication	12

a. Parity check	
b. Checksum	
c. Cyclic Redundancy check	
3.5 Standards in serial I/O	12
3.5.1 RS232C	13
3.5.2 Establishment of serial communication	14
3.6 8085 pins used for serial communication	14
a. Serial output data (SOD)	
b. Serial input data (SID)	
3.7 Selected values of baud, start, data and stop bits	15
3.8 Time Delays	15

<u>CHAPTER 4</u> THE TEMPTING ENVIRONMENT AND FACILITIES OF TURBO PASCAL VER 6.0	17
4.1 Why choose Pascal for the PC control program	17
4.2 Interrupt Driven Serial Communication	18
4.2.1 What is an Interrupt	18
4.2.2 Interrupt from PC's side	19
- Hardware and Software Interrupts	20
- Interrupt Vector Table	20
- Replacing Interrupt	21
- Writing the Interrupt Handler	24
- Restoring the data segment	25
- Turbo Pascal Interrupt Support	25
4.2.3 The Program	26
4.2.4 What does it actually do?	33

<u>CHAPTER 5</u> DEBUGGING FACILITIES	38
5.1 What is DEBUG?	38
5.2 What does DEBUG offer	38
<u>CHAPTER 6</u> THE MONITOR PROGRAM	40
6.1 Introduction	40
6.2 Programming Hints	40
6.2.1 Stack Pointer	40
6.2.2 RAM I/O Command Status Register (Control Register)	40
6.3 Reference to the basic routines	41
<u>CHAPTER 7</u> CONCLUSIONS & POSSIBLE IMPROVEMENTS ...	43

APPENDICES

- Appendix A: Listing of the ASCII codes
- Appendix B: Monitor Program listing
- Appendix C: Control Program listing
- Appendix D: User's Manual