THE 16-BIT 8096

PROGRAMMING, INTERFACING, APPLICATIONS.

122 HANDS-ON EXPERIMENTS WITH NTEL'S iSBE-96 EMULATOR

RON KATZ
and
HOWARD BOYET

MTI PUBLICATIONS

MICROPROCESSOR TRAINING INC.

THE 16-BIT

8096

PROGRAMMING, INTERFACING, APPLICATIONS.

122 HANDS-ON EXPERIMENTS WITH INTEL'S iSBE-96 EMULATOR

by

RON KATZ

[AT&T BELL LABORATORIES]

and

HOWARD BOYET

[MICROPROCESSOR TRAINING INC.]

PUBLICATIONS

MICROPROCESSOR TRAINING INC.

14 East 8th Street, NYC 10003
(212) 473-4947

DEDICATIONS:

To that special SFR, my daughter Gigi Rebecca Boyet,

To Linda and Vanessa Katz—the sparkles of my life,

and

To the Intel Troika: Andy Grove, Gordon Moore, and Bob Noyce.

Printing 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.

Library of Congress Catalog Card Number: 85-61954

Printed in the United States of America

FOREWORD

Since the introduction of 8048 (MCS®-48) Microcontroller in 1976, Intel has continued to drive the evolution of single chip microcontrollers. In 1980, Intel introduced the 8051 (MCS-51) the second generation of microcontrollers integrating more functions, features and a powerful CPU with ability to address larger memory spaces and improved overall performance. The 8048 and 8051 have since become the industry standard 8 bit microcontrollers.

In recent years, parallel to the advances in circuit design the process technologies have continued to evolve allowing more complex circuits to be fabricated on a single piece of silicon. Microcontroller designers at Intel have exploited the most advanced process technologies and forged a third generation of single chip microcontrollers called the 8096. (MCS-96 family of microcontrollers). The 8096 offers the highest level of system integration every achieved on a single chip microcontroller. It uses over 120,000 transistors to implement a high performance, powerful 16 bit CPU, 8K bytes of program memory, 232 bytes of RAM and analog, digital and serial I/O subsystems. The 8096 executes most instructions in 1 to 1.5us with the 16 x 16 multiply and 32 x 16 divide taking 6.25 usec.

The members of the MCS-96 microcontroller family (8396 ROM version, 8096 ROMless version and 8796 EPROM version) contain an oscillator and clock circuitry, full duplex serial port, watchdog timer, pulse width modulated output, 16 bit timer, 16 bit counter, high speed I/O (HSIO) subsystem, and 20 interrupt sources. In addition, versions are available with three or five 8-bit ports and 10 bit A/D converter. ROM options include 8K bytes of ROM or 8K bytes of EPROM. The package offerings include a 48 pin DIP, 68 pin grid array and a 68 pin plastic lead chip carrier.

This book provides a hands-on-experience with the MCS-96 system. Howard and Ron have made enormous efforts to provide experiments on using every aspect of this powerful single chip system. Included are experiments using the comprehensive instruction set, external and internal interrupt structure, the special function registers, PWM, A/D converters, Serial port, the high speed I/O units, arithmetic applications, the timer/counters, WDT, and signal processing.

There is an extensive range of development support tools available for the 8096 including macro assembler, PL/M, SBE-96 (Single Board Emulator) and VLSiCE-96. The authors have provided extensive hands-on working knowledge with these tools especially the SBE-96 which is a low cost, stand-alone emulator board.

I believe that this book will provide invaluable assistance to engineers, scientists, students and all others needing to learn the 8096 and to design 8096-based applications. Intel highly recommends this book.

Manmohan S. Passi
Product Marketing Manager
Microcontroller Operation
INTEL Corporation, Chandler, AZ

As part of Intel's new open development environment, the iSBE-96 Single Board Emulator can now be used on a wide variety of Intel development systems and PC-DOS compatible machines.

This low cost, easy to use tool may be run, along with the experiments in this book, on Intel's Intellec Series III and Series IV development systems. The SBE96SKIT includes an iSBE-96 emulator and Intel's assembler ASM-96 (Series III/Series IV version).

For those engineers who have an IBM PC AT, IBM PC XT, or compatible system running DOS Version 3.0 or greater, Intel offers the SBE96DKIT containing Intel's DOS version of ASM-96, all host communcation software, and cables needed to run on these systems. Intel's ASM-96 high quality assembler (both the DOS and Series III/IV versions) runs with RL-96, (linker and relocater) makes this a complete symbolic relocatable assembler supporting full macro capabilities. Intel also offers the PL/M-96 high-level language which runs on all of the above systems and complements Intel's ASM-96.

For those engineers who have an Intel iPDS system or Series II development system, iSBE-96 host communication software (ATOP-96) and cross assembler may be purchased from U.S. Software Corporation. This software can be used with either of the two kits SBE96SKIT or SBE96DKIT available from Intel.

Intel highly recommends this book on the 8096. All of the experiments may be run using any of the above host systems. This excellent hands-on applications book will quickly propel engineers, scientists, designers, technicians and all those with a need to know, into the technologically advanced world of the 8096 microcontroller. This book makes an excellent supplement to the Intel Microcontroller Handbook and any other Intel literature.

Stephen Isacoff
Product Manager
Development Systems
INTEL CORP. Hillsboro, Oregon

PREFACE

In a word, a powerhouse of a single-chip microcomputer/microcontroller and a milestone in I/O controls!

The 16-bit 8096 is a complex chip with enormous power, functionality, versatility, and far-ranging, industry-wide applications capability. In the opinion of the authors, no other chip, as of this writing, can match it in the area of microcontroller applications.

And yet, it is also a powerful general purpose microcomputer which excels in areas aside from the purely microcontroller domain. It rivals, and even surpasses in a number of areas, other well-known 16-bit microprocessors.

At this point it is desirable to present a preview of some of the advanced features of the 8096, so that the reader may gain insight and perspective as to what this book is about, why it is needed, and what it will accomplish for the user. In what follows, we occasionally refer to the 8096 by its family designation MCS-96 (Intel trademark), as there are presently a number of versions of the chip (68 pin, 48 pin, ROMLESS, ROM, EPROM, Digital I/O only, analog and digital I/O).

The 8096, 68-pin versions with analog and digital I/O on-chip, has a 16-bit CPU (register to register architecture) with 8K bytes of on-chip ROM (8397) or 8K bytes of on-chip EPROM (8797), 232 bytes of RAM (registers), 38 Special Function Registers (SFRs), 8 A/D input channels (10-bit conversion), pulse width modulation capability (PWM) with programmable duty cycle for motor control and D/A conversion, and five 8-pin I/O ports, one with alternate function capability. It has on-chip a full duplex serial communications port that is also capable of implementing I/O port expansion. This serial port also allows efficient realization of interprocessor (master-slave) communications. The versatile A/D inputs (22 usec. conversion time, sample and hold) and D/A output (PWM) allow for efficient on-chip signal processing capability, motor and closed loop control. 232 registers mean less Push/Pop, more pointers, etc.

A "watchdog Timer" (WDT) on-chip allows the programmer to take precautions against possible software/hardware malfunctions. Available to the programmer is the option of selecting at run-time the external bus width, 8-bits or 16-bits, to offer the user a choice that best adapts to his/her existing peripherals.

A 16-bit, free-running, real-time timer (2 usec. intervals) and a 16-bit event counter (with the unusual feature of two possible event sources and four software/hardware reset sources), each with overflow interrupt capability, are on-chip and can be read as Special Function Registers. The event counter can record 0 to 1 and 1 to 0 transitions that are separated in time by 2 usec. or more.

The interrupt system is extremely versatile and highly suited to the multitasking requirements of a microcontroller environment. There are 8 interrupt vectors (2 external and 6 internal) with each vector having 1 or more sources (for a total of 24 different interrupt sources). All activities (related or independent) are tied together and kept humming, in a manner transparent to the user, by the 8096's powerful interrupt system. The main program and its background/foreground activities perform in a "well-oiled" manner.

A truly remarkable and advanced feature of the MCS-96 family are the High Speed Output (HSO) and High Speed Input (HSI) units. The HSO unit on the chip can be programmed to create events (transitions) at any one or more of 6 HSO output lines at any specified times (or to generate interrupts at any programmed times from four independent "Software Timers", or to reset the event counter at any event value, or to start an A/D conversion at any desired time). The times referred to can be any multiple of 2 usec. Delay loops are thus circumvented. The HSI unit can be programmed to record the times of incoming events (and the nature of those transitions) that may appear at any one or more of 4 HSI input lines. The recorded times will be some multiple of 2 usec. The HSO and HSI pins are in addition to the 40 pins at the 5 I/O ports mentioned earlier. The resulting vastly increased I/O through-put and extremely accurate I/O control and monitoring is impressive. The HSI/O units form an independent subsystem that can be programmed to generate interrupts when their current activity is over. Thus, they may be commanded to engage in a sequence of independent I/O control and monitoring activities that significantly helps alleviate software overhead and speeds up execution time.

The HSO unit, as mentioned, contains four "Software Timers". They, too, can be programmed to provide up to four independent and, importantly, concurrent time delays of specified amount without the need to write software loops. When the specified time(s) expire(s), the software timer(s) can generate interrupts(s) to vector the program flow from "other activity" to the Software Timer Interrupt Routine where action appropriate to that expiration time is programmed. Again, the effect on I/O through-put is considerable. Times can be programmed to 2 usec.

resolution (12 MHz clock) ensuring unusual accuracy.

The many aspects and novel features of the powerful 8096 instruction set will be brought out in the experiments where they are used and explained. Some of those unusual features follow. 6 addressing modes (with two variants), allows the programmer to deal with both internal and external memory in a flexible and general manner. The ease this affords in operating on any register or memory location overcomes the bottleneck often associated with accumulator-based processors. Another powerful attribute is the ease of establishing pointers with the indirect addressing mode (with or without auto-increment). This allows the use of any on-chip word register location (116 of them) as a pointer to anywhere in memory, providing an extremely powerful tool for manipulation of data. Other features are the availability of 3-operand instructions (usually associated only with minicomputers or main frames) which make the programmer's task easier; inclusion of 16 x 16 (or 8 x 8) multiply and 32 x 16 (or 16 x 8) divide instructions (both signed and unsigned); a very versatile and complete set of multiple shift instructions; a normalization instruction (valuable in floating point arithmetic); the existence of the "sticky bit" and "overflow trap" flags (useful in rounding, truncation, and precision arithmetic); "sign extend" and "load extend" instructions (for ease in matching 8-bit to 16-bit quantities); the flexibility and accuracy to be had with the availabilty of byte, word, and double-word operand instructions -- these are just some of the features that allow the 8096's instruction set to go beyond some of the more powerful 16-bit microprocessors, let alone microcontrollers. Its arithmetic processing power is formidable, accurate and fast. Average execution time of the multiply and divide instructions is about 6.5 usec. Most of the 8096's instructions execute in 1 to 1.5 usec. (12 MHz clock).

These are just some of the advanced and powerful features and attractions of the MCS-96 family of microcontrollers that over 120,000 transistors on-chip have implemented. And that makes a lot to learn and master.

All these features and functionality in the 8096 make the task of learning and mastering the 8096 by oneself a formidable one, even for the experienced microprocessor designer or user.

The purpose of this book is to get you through all aspects of the 8096 in the quickest and, yet, most thorough manner. We do this by a logical step-by-step, chapter-by-chapter approach from ground up. 122 hands-on experiments, programmed in ASM-96 (and assembled in 8096 HEX code) are presented, using Intel's iSBE-96 emulator as the learning and applications vehicle. All

the fundamentals behind the 8096 are stressed and investigated, and a large number and variety of applications experiments are presented, either in conjunction with the presentation of the fundamentals, or separately (but always based on them).

The 8096 and this book are not for beginners. While no prior 8096 background is assumed, it is assumed that the reader has a solid classical microprocessor background (architecture, assembly language, a thorough working facility with HEX and binary, interfacing/digital logic, and programming). While not necessary, previous exposure to microcontrollers (e.g. the 8051) might be helpful (see the book "The 8051: Programming, Interfacing, Applications. 81 Hands-On Experiments with Intel's SDK-51" by the authors of this book). Except for their serial ports, the 8051 and 8096 are completely different microcontrollers.

The keys to understanding and mastering the 8096 are: 1) its Special Function Registers, through which efficient control of all 8096 I/O (or its monitoring) is accomplised, 2) its versatile interrupt system which makes possible the transparent execution of a large number of related or unrelated tasks, 3) its ultra-fast independent HSI/O input/output units, and 4) the unique and powerful features of its instruction set. Keep that in mind as you go through this book and as you read the descriptions of the various chapters which follow.

The experienced user may find that chapters 2 and 3 can be skipped or given brief attention. In that case the reader can go directly to chapter 4. All the experiments in this book were entered in ASM-96 assembly language, with the program print-outs showing both ASM-96 mnemonics and HEX object code (chapter 10 gives examples using the high level language PL/M-96).

Chapter 1 presents the relevant 8096 information, gathered in one place for the convenience of the user. It is to be referred to, as necessary, when the experiments are studied. A list of Intel references, from which the information was taken, and others, are presented, should the reader need additional 8096 information.

Chapter 2 is concerned with the iSBE-96 emulator. Relevant and necessary information on it is presented. We then discuss our approach in downloading programs to the iSBE-96 in HEX format object code from ASM-96 assembly language input to an IBM PC. A listing of other host systems and PCs on which the experiments in this book can be performed is given (see also the FOREWORD to the book). After the interfacing requirements needed to do

the experiments are given, 9 experiments are performed demonstrating the use of the iSBE-96, its relevant commands, and the important features of the iSBE-96 monitor's stack that the user should be alerted to. Debugging techniques (single-step, breakpoints, etc.) using the SBE's commands are covered. The first I/O experiment involving ports P0 and P1 is presented, as well as one on the Watchdog Timer.

Chapter 3 focuses on the 8096 instruction set and its capabilities. 18 experiments demonstrate all the addressing modes, the flags, the shift instructions, the "sticky bit", the "overflow trap", the load/extend, the normalization, and the multiply and divide instructions. Also covered are the indirect branch and generic branch instructions, the conditional jumps, the PUSHF, POPF, PUSH, and POP instructions, the System Reset instruction, the CALL and JUMP instructions (of various kinds), use of 3-operand instructions, and more.

Chapter 4 on the 8096's interrupt system is a key chapter in coming to grips and working with the 8096. 10 experiments are presented covering every facet of the 8096's interrupt structure. While the experiments center on the 8096's two external interrupts (to more easily and conveniently demonstrate the interrupt facilities), their understanding will lead to a direct understanding of the 8096's six internal interrupt vectors (and their sources). The fundamentals are identical. (The internal interrupts are introduced in Chapter 5 and used in all their aspects thereafter through-out the book). Conditions for interrupt servicing (immediate or pending) are established. The interplay among the pending interrupt, interrupt mask, and global interrupt bits in the relevant SFRs is thoroughly investigated. Both hardware and software-generated interrupts are explored. Nesting of interrupts with the 8096 is thoroughly covered. A powerful user option available with the 8096 is the possibility of establishing one's own priority interrupt structure among the 8 interrupt vectors. This is covered in two experiments.

Chapter 5 is extremely important to understanding and applying the 8096. It involves a study of all the 8096's SFRs (except the serial I/O SFRs that are deferred until chapter 8). To this end, 25 experiments are presented in which all SFR bits are examined as they relate to I/O control and status or data monitoring. A good number of these experiments already project a strong applications orientation. It should be kept in mind that the 8096 on-chip devices (A/D, PWM, WDT, HSI/O, serial I/O, timers, counters, digital I/O ports) and interrupt facilities are controlled and monitored by means of on-chip (RAM) register-mapped I/O through the SFRs. This reduces software overhead and

results in extremely efficient I/O processing. Half the experiments in this chapter deal with the HSI/O SFRs.

Chapter 6 covers 26 applications experiments centered around the SFRs. Half are devoted to the HSI/O units. In several examples the applications are purposely implemented first the "classical way", to point out the major software and hardware I/O advantages the HSI/O units afford. A glance at the table of contents will reveal the broad scope of this important chapter. A mastery of this chapter is essential to working successfully with, and applying, the 8096's SFRs and HSI/O units.

Chapter 7 is devoted to a discussion of the important arithmetic features of the 8096 that give it its extraordinary arithmetic processing capability where speed, accuracy, versatility, and reductions in software overhead are essential. In particular, the normalization instruction (very useful in floating point arithmetic), the sign-extend and load-extend instructions, the "sticky bit", "overflow trap", and "negative" flags are all discussed and used to illustrate the 8096's efficiency and power in the arithmetic realm. Three experiments involving sign extension, the sticky bit (in rounding operations), and the negative flag are presented to illustrate some of the points brought out in the chapter. More extended arithmetic is cound in ch. 9.

Chapter 8 deals with the 8096's serial port. Serial communications is the medium employed in many real world applications. A thorough discussion and explanation of the relevant SFRs and how to program them to achieve desired transmission/reception characteristics in the various serial modes are first presented. Six applications experiments are then studied demonstrating serial communciations via the TXD and RXD pins on the 8096. The first two are concerned with I/O port expansion (most important for a microcontroller), and are implemented in serial mode 0. Reception from, and transmission back (echo) to a CRT terminal involving mode 1 is next covered in three experiments. Two involve polling of the serial port's status SFR and the other is based on the serial port's interrupt vector. All three illustrate either a simple echo or the processing of the received data before re-transmission (in the latter case processed data buffer tables are created). Finally, an experiment on interprocessor communciations between master and slave(s) is presented (here an iSBE-96 master and another iSBE-96 slave). Serial modes 2 and 3 and serial interrupts are involved.

Chapter 9, in a sense, is the culmination point of the book. It presents applications with the 8096 in a broad range of areas beyond those covered in previous chapters. The vehicle is again the iSBE-96. As always, we attempt to emphasize the unique

features of the 8096 that make it the microcontroller of choice in the applications covered (or suggested in the problems). It is not our intent (nor within our capability) to focus on everyone's areas of interest, or to implement the reader's specific problem(s) and needs with the 8096. In this chapter we endeavor to present a series of broad applications in various areas with the purpose of suggesting ideas and methods of implementation, and of illustrating the far-ranging capabilities, power, versatility, and efficiency of the 8096. The chapter puts to use everything learned in the previous chapters. The applications presented may themselves be of direct interest and use to the reader. They span a wide range from areas involving frequency measurement and control, to serial (remote) control of I/Os, to high speed manufacturing operations, to materials handling, to closed loop control, to digital filters, to smoothing, and more-much more (consult the table of contents under chapter 9). We could go on and on with a microcomputer like the 8096.

Chapter 10, the final one of the book, attempts to make the reader aware, albeit in a small way, of some of the Intel software/hardware development tools available for the 8096 besides the iSBE-96 emulator board that we have been working with. Thus, 2 applications using the high level language PL/M-96 are offered (one of which is a PL/M-96 version of a previous experiment from chaper 6 that used ASM-96 assembly language). FPAL-96 routines are used. Also presented is some information on the Intel VLSiCE-96P (Intel trademark): In-Circuit Emulator for the MCS-96 Family of Microcontrollers. These software/hardware development tools are often indispensable in achieving quicker and more economical solutions to specific applications.

We should like to bring to your attention that a large number of problems have been added in the book. They appear at the end of various experiments and suggest either other ways of accomplishing the same application with the 8096 and/or new or further applications to explore based on that experiment. We believe these problems will add significant value to the book.

Also to be kept in mind is the fact that in many experiments we have ended the main program part of the application in a self loop, simulating "other tasks" rather than a mere "halt". The various interrupt sources employed in the particular experiment then achieve multitasking over and above the main activity, as various interrupt routines are alternately entered and serviced. The multitasking capability of the 8096 is impressive.

The various applications and programs represented in the numerous experiments can themselves possibly serve as a library of modules in building more encompassing programs for still larger applications.

ACKHOWLEDGEMENTS

The undertaking of this book was originally suggested by Alex Toth and Martin Pawloski of the Intel Microcontroller Operation, Chandler, Az. The final decision rested with Mohan Passi, product marketing manager of 16-bit operations in the Microcontroller group. We are indebted for his confidence and continued solid support, encouragement, and total cooperation. He made the book possible and did everything to ease our task.

The authors would like to single out two individuals who played key roles in helping to expedite the book's progress both from a technical and time stand-point: Ira Horden, Applications Engineer in the Microcontroller Operation was outstanding for his prompt assistance on a number of technical questions relating to the 8096 raised by the authors. Some of the mysteries of the 8096 began to unravel as Ira showed the way. This led to important break-throughs. We greatly appreciate all the time he took, on top of a very busy schedule, to go through the manuscript from beginning to end. His comments and suggestions were invaluable. Errors of any sort are the responsibility of the authors. Steve Isacoff, Product manager, Development Systems, Intel (Hillsboro, Oregon) was indispensable. His prompt, concerned, and enthusiastic cooperation in contributing several iSBE-96 emulator boards at various stages of iSBE-96 hardware and monitor updating (and in expediting a prompt repair at a critical time in the book's undertaking) was instrumental to the smooth, uninterrupted flow of progress that would otherwise not have been possible. The master-slave and master-remote controller experiments would not be in this book were it not for Steve's help in supplying boards. He was also responsible for supplying latest information on VLSiCE-96 that appears in chapter 10 and on iSBE-96 emulators that appears in chapter 2.

We would also like to express our thanks to Randy Wilhelm, Dennis Regimbal, Dave Ryan, and Lionel Smith of the Intel Microcontroller Operation for their help in many different and important ways. Dennis was instrumental in getting updated 8096 information to us at a number of critical times. Mohan, Denis, and Dave also made valuable suggestions regarding the manuscript. Tom Geletka of Intel's Customer Training division in Schaumburg, IL was responsible for duplicating one of our experiments, and for contributing another, both in PL/M-96, which then became part of chapter 10. Tom was another example of cooperation with personal effort over and above a busy work load. We thank him for his timely contribution.

The Intel Sales Office in Hauppauge, NY helped in providing up-

dated Intel manuals on a number of crucial occasions. (With the 8096 you've got to have manuals and you've got to read them like a hawk. We think you've also got to have this book).

One of us (R.K.) wishes to express thanks to his management at AT&T Bell Laboratories, Whippany, NJ for their encouragement and support of his independent effort in this important project.

The word-processing of the manuscript was a tedious, long and difficult undertaking. "A Personal Touch Word Processing Service Bureau" of 550 Durie Avenue, Closter, NJ was outstanding in its service and promptness. President (Pauline Carroll) and Office Manager (Linda Jackle) came through during many a crunch. They're real pros. (Mike Carroll also helped out).

Another real pro is Frank Moore who has done many covers, and title/copyright pages for MTI Publications. His work on the mechanicals has always been outstanding.

The manuscript for this book was written in many near and far-off places: airlines, parks, restaurants, busses, subways, vans, taxis, trains, pubs, libraries, in bed, dance and acting studios (daughter Gigi) - wherever a table, desk, or prop of any kind was available. One of us (H.B.) wishes to thank MaryLou's and Viva Pancho's restaurants in Greenwich Village, NY for providing a conducive atmosphere where ideas and experiments fermented through various stages of the book.

The 8096 was a BIG challenge from beginning (January 1985) to end (January 1986) - more so than any of the other microprocessors and microcomputers on which the authors have written books. We hope we have met that challenge and that this book will succeed in quickly helping others conquer this complex chip of great versatility and awesome, but beautiful, power. We are indebted to Migdalia and Linda for their unswerving patience and understanding during this period of almost total immersion.

THE AUTHORS ARE GRATEFUL TO THE INTEL CORPORATION FOR THEIR PERMISSION REGARDING RE-PRINTING REFERENCE MATERIAL FROM THE VARIOUS 8096 INFORMATION SOURCES LISTED ON P. 6 OF CHAPTER 1. THE PHOTOS ON PP. V AND VI ARE PRESENTED COURTESY INTEL CORP. ALL MNEMONICS USED IN THIS BOOK ARE COPYRIGHT INTEL CORPORATION, 1983. COPYRIGHT NOTICES ARE LISTED FOR EACH REFERENCE ON P. 6, CHAPTER 1.

January, 1986
New York, NY

Ron Katz
(AT&T Bell Laboratories)

Howard Boyet
(Microprocessor Training Inc.)

CONTENTS