Preface

We are very happy to witness the launch of this third edition of our work, which we are sure will generate a lot of interest amongst readers. Since the publication of the first edition of this book in 2000, followed by its second edition in 2006, the field of microprocessors and microcontrollers has undergone phenomenal developments. The Intel microprocessor family has evolved from those early days of 4-bit and 8-bit microprocessors through a long evolution of 16-bit, 32-bit microprocessors to Pentium and the advanced I-7 in addition to many variants of Pentium, Celeron and Itanium. The arena of multi-core processors has already begun. On the other hand, 8-bit microprocessors are mostly of academic interest only, and microcontrollers are widely used for small, dedicated hardware systems and embedded applications. In this background, we present a brief overview of microprocessors, starting from the basic concept of a microprocessor followed by the intermediate microprocessors, and concepts of microcontrollers and interfacing with microcontrollers.

A microprocessor is a programmable circuit that supports the execution of a set of instructions called an instruction set. Each instruction in the instruction set is represented by a predefined unique sequence of 1s and 0s called OPCODE (Operation Code). The data required for the execution of operation, i.e. operands are also expressed in 1s and 0s, i.e. binary form, and follows the opcode. Customarily, the opcode and operands are specified in instructions in the form of bytes. Thus, microprocessor instructions consist of opcode and operand bytes. A microprocessor can execute one instruction at a time. Hence, a task to be executed by a microprocessor is expressed in terms of a sequence of instructions called a 'program'. It is to be noted here that the microprocessor is a digital circuit and understands (accepts) the instructions or data expressed only in terms of 1s and 0s. For convenience of entering into a microprocessor-based system, it is customarily entered byte by byte. Thus, a microprocessor system program is a sequence of bytes expressed in hexadecimal system. These programs, expressed in terms of hexadecimal bytes, are called 'machine-language programs'. Thus, the types of circuits, systems and machines which are able to remember the sequence of instructions, the operands related to each instruction, can execute them and finally store the result of the complete execution are called 'programmable circuits or machines'. These are available in the form of crude microprocessor-based systems or advanced personal computers or more savvy laptops, notebooks and palmtops.

Intel Corporation has been one of the pioneers in the microprocessor industry right from its infant stage. Even today, they are one of the major players in the field though many have emerged. It is thus very natural and justified to start the introduction to the field using Intel's framework.

With the advent of the first 4-bit microprocessor 4004 from Intel Corporation in 1971, there has been a silent revolution in the domain of digital system design, which has shaken many facets of the current technological progress. In the last 40 years, the world has seen an evolution of microprocessors, whose impact on today's technological scenario is phenomenal.

This evolution was possible because of tremendous advances in semiconductor process technology. The first microprocessor 4004 contained only around thousand transistors, while the component density increased more than threefold in less than a decade's time. Immediately after the introduction of the 4004, Intel introduced the first 8-bit microprocessor 8008 in 1972; these processors were, however, not successful because of their inherent limitations. In 1974, Intel released the first general-purpose 8-bit microprocessor 8080. This CPU also was not functionally complete and the first 8-bit functionally complete CPU 8085 was introduced in 1977.

The 8085 CPU is still the most popular amongst all the 8-bit CPUs. The 8085 CPU houses an on-chip clock generator and provides good performance utilizing an optimum set of registers and a reasonably powerful ALU. The major limitations of these 8-bit microprocessors are their limited memory addressing capacity, slow speed of execution, limited number of general-purpose registers and non-availability of complex instructions and addressing modes. Another important point to be mentioned here is that 8085 does not support adequate pipelining or parallelism, which is so important for enhancing the speed of computation. For example, the non-availability of any instruction queue in an 8085 CPU leads to a situation where the fetching of opcode and operands along with the execution is compelled in an absolutely sequential manner.

The first 16-bit CPU from Intel was a result of the designers' efforts to produce a more powerful and efficient computing machine. The designers of 8086 CPU had taken note of the major limitations of the previous generations of the 8-bit CPUs. The 8086 contains a set of 16-bit general-purpose registers, supports a 16-bit ALU, a rich instruction set and provides a segmented memory-addressing scheme. The introduction of a set of segment registers for addressing the segmented memory in 8086 was indeed a major step in the process of evolution. All these features made this 16-bit processor a more efficient CPU, and thus it is termed the first member of Intel's advanced microprocessors family.

The development of the IBM PC started in July 1980, and precisely after one year, the first machine based on Intel 8088 CPU (which is functionally equivalent to 8086 but supports only 8-bit external data bus) with one or two floppy disk drives, a keyboard and a monochrome monitor was announced in August 1981. The machine operating system was an early version of the operating system MS-DOS from Microsoft. In March 1983, a new version of IBM PC called PC-XT was introduced with a 10-megabyte hard disk, one double-side double-density floppy disk drive, keyboard, monitor and an asynchronous communication adapter. In fact, the introduction of IBM PCs in 1980s had, to a large extent, produced a profound impact on the evolution of microprocessors. With the introduction of each new generation of microprocessors, the performance of Personal Computers have also been enriched. Rather the computational performance of PC machines and their state of art has improved due to the availability of high-speed advanced microprocessors.

The major limitation in 8086 was that it did not have memory management and protection capabilities, which was considered an extremely important feature, deemed to be an integral part of a CPU of the eighties. The 80286 was the first CPU to possess the ability of memory management, privilege and protection. However, the 80286 CPU also had a limitation on the maximum segment size supported by it (only 64 KB). Another limitation of 80286 was that once it was switched into protected mode, it was difficult to get it back to real mode. The only way of reverting it to the real mode was to reset the system.

In the mid-eighties, more computationally demanding problems necessitated the development of still faster CPUs. Thus appeared 80386, which was the first 32-bit CPU from Intel. The memory management capability of 80286 was enhanced to support huge virtual memory, paging and four levels of protection. The design of 80386 circumvented this problem. Moreover, the maximum segment size in 80386 was enhanced and this could be as large as 4 GB with 80386 supporting as many as 16384 segments. The 80386 along with its math coprocessor 80387, provided a high-speed environment even for graphical applications. Around the early nineties, the graphical displays, and interactive tools started becoming more and more popular. Soon, the operating system Windows 95 with graphical interface became a part of every machine. This lead to design of 80486 as a processor of similar capacity as 80386, but with an integrated math coprocessor. After getting integrated, the speed of execution of mathematical operations required for graphics applications enhanced threefold. In addition, for the first time an 8 KB four-way set associative code and data cache was introduced in 80486. A five-stage instruction pipelining was also introduced.

The earlier generation CPUs supported rather crude instruction sets. It was not expected that the programmers those days would write large machine-code programs. A single high-level instruction might be compiled into ten or even hundred machine-code operations. In the course of evolution, from the early 8-bit CPUs, the trend was to design CPUs that could support more and more complex instructions at the assembly-language level. Designers of Complex Instruction Set Computers (CISC) wanted to reduce this gap. The design and research efforts further led to the first virtually 64-bit processor: Pentium-I. The Pentium used to execute the computationally greedy applications with a 64-bit wide data bus. Thus, a family of operations like Single Instruction Multiple Data (SIMD) and Multiple Instructions Multiple Data (MIMD) emerged and became popular for multimedia applications. Hence, the developing technologies of fabrication and the demand for more and more computational power for advanced applications resulted in launching of many variations of Pentium either with higher clock speed or with special features like superscalar execution, multimedia extension, streaming SIMD extension and RISC features. To name a few Pentium variants, they are P-I, P-II, P-III, P-IV, Pentium PRO, and Pentium MMX, etc. Further, it was observed that whatever the technology of manufacture, a single processor core will always have upper limitation on its processing capability and higher degrees of parallelism may boost the processing power. Thus, multi-core processor architectures with instruction-level massive parallelism have been introduced and are popular in modern PCs.

Since the early days of microprocessor development, designers have tried to make them more powerful by designing more complex instructions. But then, some of these powerful instructions and addressing modes were hardly used by programmers. In fact, some of these instructions' logic took up a large part of the microprocessors' silicon chip. The Reduced Instruction Set Computer (RISC) designers observed that the data-movement type of machine instructions are frequently executed by the CPU. They have optimized CPUs to execute these instructions rapidly. RISC provided a regular set of instructions having the same format with a lot of pipelining. To improve the processor's performance, the possible ways are suggested below.

(a) Increasing the processor and system clock rate
(b) Optimizing and improving the instruction set
(c) Executing multiple instructions in one cycle and incorporating parallelism in the CPU architecture

The first option is applicable both to CISC and RISC processors. The second option is primarily for CISC but is applicable to RISC as well. The third option is more suited to RISC CPUs. Ever since the appearance of commercially available RISC CPUs, there has been a debate over the performance of RISC versus CISC. The RISC architects argue that their instructions may be executed in a single cycle and thus take less time than is taken by a CISC CPU. This is because of pipelining, reduction of instructions to a simple operation and synthesis of complex operations with compiler generated code sequences. When RISC machines first arrived in the market, CISC processors were performing at 6 to 10 cycles per instruction, while RISC CPUs could execute a set of simpler instructions in one cycle and offer better performance. Many CISC processors have subsequently used many features of RISC.

A processor without its memory and peripherals is hardly a useful component. Integrating a microprocessor with its peripherals to form a practical system requires a big circuit board and many soldered connections. Thus, microprocessor systems have disadvantages like big physical size, less reliability, no upgradability, high power consumption and high cost of product and maintenance. To pacify these disadvantages, Intel introduced its first 8-bit embedded microcontroller 8031 with 128 bytes on-chip RAM and capable of addressing program memory of 4 KB with on-chip ports, timers and a serial communication unit. Soon its on-chip EPROM version 8051 was introduced by ATMEL and it became so popular that it replaced most 8-bit microprocessors. It must be noted that microprocessors are implemented as

Von Neumann architecture while microcontrollers are implemented as Harvard architectures. Thus, the microcontroller is nothing but a microprocessor with on-chip integrated peripherals. Microcontroller-based systems have advantages like small size, low power consumption, portability, better upgradability, low cost of manufacturing and maintenance. Due to these advantages of embedded controllers and ARM processors, they have become very popular in dedicated small applications and embedded systems. It should, however, be noted that programming with microcontrollers and ARM using machine language is very tedious and prone to mistakes. The advanced embedded system design and programming tools for microcontrollers and ARM processors facilitate use of sophisticated debuggers, emulators, simulators along with the ease of high-level language programming like C, JAVA and operating systems like VXWORKS, EMBEDDED LINUX, etc.

Target Audience

This book is intended as a textbook on ‘Advanced Microprocessors’ which is a compulsory course at graduate and postgraduate levels in many science and engineering branches of studies, specially in Electronics, Electrical, Instrumentation, Physics and Computer Science disciplines. The book is suitable for a one-semester course on advanced microprocessors, their architectures, programming, hardware interfacing and applications. The book will primarily serve the needs of undergraduate students for CSE, IT, ECE, EE, EEE. It can also be used by polytechnic students doing diploma and DOEACC courses in computer sciences. The subject needs to have a very practical approach and a basic knowledge of C programming is required to understand the practical applications of the subject.

Objective of the Book

The purpose of our book is to provide readers with a good foundation on advanced microprocessors, their principles and practices. We have tried to keep an appropriate balance between the basic concepts and practical applications related to microprocessor technology. Thus, we have aimed at the following:

- To present fundamental concepts of advanced microprocessors and their architectures
- To enable students write efficient programs in assembly-level language of the 8086 family of microprocessors
- To make students aware of the techniques of interfacing between processors and peripheral devices so that they themselves can design and develop a complete microprocessor-based system
- To present the concepts of microcontroller and ARM architectures in a lucid manner
- To present a host of interesting applications involving microprocessors and microcontrollers

Salient Features

Some of the salient features of the book are listed below.

- Simple and easy-to-understand language
- Updated with crucial topics like ARM *Architecture, Serial communication Standard USB*
- New and updated chapters explaining *8051 Microcontrollers, Instruction set and Peripheral Interfacing along with Project(s) Design*
- Improved explanation and presentation of concept like 80286 and 80386 Descriptors, Addressing modes and 8051Stacks
- Explanation on latest real-life applications like *Hard drives, CDs, DVDs, Blue Ray Drives*

Web Supplements

There are a number of supplementary resources available on the Website http://www.mhhe.com//ray/microprocessors 3 and updated from time to time to support this book.

- 700 presentation slides for instructors and students.

Chapter Organisation

1. The book covers a wide range of microprocessors from 16-bit 8086 to Pentium in a lucid manner. The evolution from one processor architecture to another is evident as one goes through the chapters. A detailed description of each microprocessor has been presented in individual chapters. **Chapter 1** covers 8086/8088 architecture in adequate detail. **Chapter 9** covers 80286 along with its coprocessor. **Chapter 10** covers the microprocessor 80386 and its coprocessor 80387. This chapter also covers 80486, the integrated CPU with built-in math coprocessor in sufficient detail. Pentium, the latest in the Intel microprocessor family, has been briefly presented in **Chapter 11**. One of the most advanced microprocessors of Intel, Pentium IV is presented in **Chapter 12**. A few RISC architectures and their features have been presented in **Chapter 13**.

2. An important feature of the book is the inclusion of a number of interesting applications of microprocessors. An adequate account of each one of these applications has been presented in the book. An interesting application of microprocessors for controlling an aluminium smelter has been presented in **Chapter l4. Chapter 15** presents another interesting application in the area of pattern scanner design. Design of a microprocessor-based electronic weighing bridge has been elaborated in **Chapter 16.**

3. One of the major problems encountered by students is difficulty in writing assembly-language programs. In this book, a large number of assembly-language programs have been presented. They will enable students to write efficient codes on 16-bit or 32-bit platforms. **Chapter 2** covers the 8086 family instruction set and the assembler directives with necessary examples. The art of programming in 8086 assembly language has been elaborated with a large number of program examples in **Chapter 3**. A very important spectrum of programs involving stacks, subroutines, interrupts, macros and time delays has been discussed in adequate detail in **Chapter 4.**

4. A good account of a number of general peripheral devices like I/O ports, keyboards, displays, ADCs, DACs, stepper motors, etc., and their interfacing with 8086 has been elaborated in **Chapter 5**.

5. Some special dedicated peripherals like timer, USART, keyboard display interface, interrupt controllers, and DMA controllers, CRT controllers, floppy disk controllers, etc., have been discussed elaborately along with interfacing examples and programs in **chapters 6 and 7**. Detailed knowledge about these peripherals is extremely important for interfacing these devices with advanced CPUs and also for designing standalone microprocessor-based systems. Brief notes on high-capacity memory devices and a high-speed serial communication standard USB are presented at the end of the respective chapters.

6. The importance of multiprocessor-based system design cannot be underestimated in today's world. A full chapter has been devoted to presenting issues related to the multiprocessor-based system design. The co-processors like 8087, 8089, etc., along with their interfacing strategies have been presented in

Chapter 8 of the book. Design of an 8088 based multi-microprocessor system has been described in adequate detail in this chapter with an example.

7. In recent days, the importance of microcontrollers has increased manifold for small, dedicated system designs and embedded applications. Sensing the fact, the instruction set of MCS-51 family has been included in **Chapter 17** followed by a few simple programming examples. A full in-depth chapter on interfacing with MCS-51 microcontroller family has been included with adequate theory , significant number of interfacing, programming and project design examples as **Chapter 18.** Thus, though the core content of this book is weaved around advanced microprocessors and interfacing, this book presents microcontrollers and interfacing in significant details.

Like the earlier editions, this text very much maintains the apt balance between basic concepts and practical/real-life applications related to the subject technology.

Additional projects on microprocessors and microcontrollers will surely help students in grasping better practical applications of the subject!

Feedback

We firmly believe that, on the whole, this complete text will be very useful for the students, designers and general readers for their academic and technical ventures in the field of microprocessors and microcontrollers. Further suggestions for improvement will always be welcome.

KM Bhurchandi
AK Ray

Publisher's Note

Do you have any feedback? We look forward to receive your views and suggestions for improvement. The same can be sent to *tmh.csefeedback@gmail.com*, mentioning the title and authors' name in the subject line.

Pircay related issues, may also be reported.