Guide to This Manual 1

CHAPTER 1
GUIDE TO THIS MANUAL

INTRODUCTION

This manual provides reference information applicable to the 80960SA/SB embedded processor. It is intended for use by both software and hardware designers familiar with the principles of microprocessors and with the 80960SA/SB architecture.

This manual contains:

- Information about the processor's programming environment and kernel (or executive) support facilities.
- Information relating to the Intel i960 architecture.
- Sufficient information to develop 80960SA/SB microprocessor systems at the software and hardware level.

MANUAL STRUCTURE

This manual contains 14 chapters and seven appendices for specific users as listed in Table 1-1.

Table 1-1: Chapters of Interest to Specific Users

User	Chapters
Applications Programmers	Chapters 2, 4, 8, 9, 10 Appendices A, C, E, and F
Compiler Designers	Chapters 2, 4, 5, 6, 8, 9, 10 Appendices A, B, C, E, and F
Kernel Designers	Chapters 2 through 11 Appendices A through F
Hardware Designers	Chapters 2, 12, 13 Appendix G

CHAPTER OVERVIEW

The following is a brief overview of each chapter and appendix:

Chapter 1 — Guide to This Manual. Overview of this manual.

Chapter 2 — Introduction to the i960 architecture. Describes the environment in which instructions are executed and discusses the address space, registers, instruction pointers, and arithmetic controls. In addition, this chapter describes the hardware system design of a 80960SA/SB system and includes a system configuration illustrating the various components of the 80960SA/SB system.

Chapter 3 — Processor Management and Initialization. Describes the processor management facilities and includes a discussion of the system data structures, interrupts, process controls, processor and program states, and memory requirements. This chapter concludes with the software requirements for processor management and processor initialization.

Chapter 4 — Procedure Calls. Describes the various mechanisms available for making procedure calls including the call/return mechanism, local calls, parameter passing, system calls, system-procedure table, user-supervisor protection model, and branch-and-link procedure calls.

Chapter 5 — Interrupts. Describes the 80960SA/SB interrupt mechanism including interrupt priority, interrupt table, interrupt-handling procedures, interrupt stack, software requirements for handling interrupts, and hardware signaling of interrupts.

Chapter 6 — Faults. Describes the processor's fault-handling mechanism, the fault-table structure, fault-handling procedures, software requirements for handling faults, and details about fault-handling mechanisms in hardware. Each fault is described in detail in a reference section at the end of the chapter.

Chapter 7 — Debugging. Describes the debugging, tracing and monitoring support facilities, including the trace control register. Trace modes and the software support required for tracing are also discussed.

Chapter 8 — Data Types and Addressing Modes. Describes the non-floating-point data types and methods of addressing bits and bytes, including the addressing modes provided for accessing data in memory.

Chapter 9 — Instruction Set. Describes the non-floating-point instructions in the 80960SA/SB instruction set, arranged by functional groups, and includes a brief description of the assembly language instruction format. An alphabetical listing of the complete 80960SA/SB instruction set with detailed descriptions of each instruction, assembly-language syntax, examples, and algorithms, is at the end of this chapter.

Chapter 10 — Floating-Point Instructions. Describes the floating-point processing facilities of the 80960SB processor. This chapter includes an overview of floating-point numbers and describes the 80960SB floating-point data types and their relationship to the IEEE floating-point standard. Also described are the floating-point instructions, exceptions, and faults.

Chapter 11 — Intra-Agent Communication. Describes the intra-agent communication (IAC) mechanism, which lets several processors communicate with one another on the bus. The topics covered include the IAC mechanism and software requirements for using internal IACs. Each IAC is described in detail in a reference section at the end of the chapter.

Chapter 12 — 80960SA/SB Bus. Describes the local bus interface to the 80960SA/SB processor and includes a detailed discussion of local-bus signal descriptions, timing generation, arbitration, interrupt handling, initialization, and error signals.

Chapter 13 — Typical System. Describes techniques of designing memory subsystems and interfacing I/O devices to the local bus.

Appendix A — Instruction Quick Reference. A quick reference section for 80960SA/SB instructions and data structures. Includes an alphabetical list of 80960SA/SB instructions and a list of instructions sorted by machine language opcode.

Appendix B — Machine-Level Instruction Formats. Describes the machine-level instruction formats and the four instruction formats used by the 80960SA/SB processors.

Appendix C — Instruction Timing. Describes the 80960SA/SB processor's instruction pipeline, instruction timing, the number of clock cycles required for each instruction, and the internal structure of the 80960SA/SB processor.

Appendix D — Initialization Code. Listing of example code to initialize the 80960SA/SB processor.

Appendix E — Register and Data Structure Summary. Illustrations of system data structures.

Appendix F — Considerations for Writing Portable Software. Discusses various aspects of the 80960SA/SB architecture that affect the portability of software written for the 80960SA/SB processor with respect to future implementations of the i960 architecture.

Appendix G — 80960SA/SB Signal Reference. Describes the pin-outs of the 80960SA/SB processor.

Appendix H — 80960SA/SB-27960KX Burst EPROM Interface. Describes the use of high speed burst EPROMs in an 80960SA/SB system.

NOTATION AND TERMINOLOGY

The following paragraphs describe the notation and terminology used in this manual that have special meaning.

Reserved and Preserved

Certain fields in the processor's system data structures are described as being either *reserved* fields or *preserved* fields. A reserved field is one that other implementations of the 80960SA/SB architecture can use. To help insure that a current software design is compatible with future processors based on the 80960SA/SB architecture, the bits in reserved fields should be set to 0 when the data structure is initially created. Thereafter, software should not access these fields.

Some fields in system data structures are shown as being required to be set to either 1 or 0. These fields should be treated as if they were reserved fields. They should be set to the specified value when the data structure is created and not accessed by software thereafter.

A preserved field is one that the processor does not use. Software may use preserved fields for any function.

Set and Clear

The terms *set* and *clear* are used in this manual to refer to the value of a bit field in a system data structure. If a bit is set, its value is 1; if the bit is clear, its value is 0. Likewise, setting a bit means giving it a value of 1 and clearing a bit means giving it a value of 0.

Instruction Set Extensions

In the complete listing of the 80960SA/SB instruction set contained in Chapter 9, a box around the alphabetic reference heading, such as:

addr, addrl

indicates the instruction or group of instructions are extensions to the i960 architecture instruction set. Most boxed extensions apply only to the 80960SB processor; extensions that apply to the 80960SA and 80960SB processors are boxed with a double-line border.