National
Semiconductor
Corporation

SC/MP
Technical
Description

Simple-to-use
Cost-effective
MicroProcessor

Publication Number 4200079A

SC/MP Microprocessor

SC/MP Technical Description

January 1976

2900 Semiconductor Drive
Santa Clara, California 95051

PREFACE

This technical description defines SC/MP (Simple Cost-effective Micro-Processor) and its supporting complement of hardware and software items. SC/MP is a full-featured microprocessor designed and manufactured by the National Semiconductor Corporation.

The material contained in this publication is presented at a level-of-detail sufficient for use in preparing a general, preliminary design of an SC/MP-based application. Additional information pertaining to SC/MP may be obtained from the nearest sales office of the National Semiconductor Corporation.

The information presented herein is up-to-date at the time of publication and is subject to change without notice.

CONTENTS

CONTENTS (Continued)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

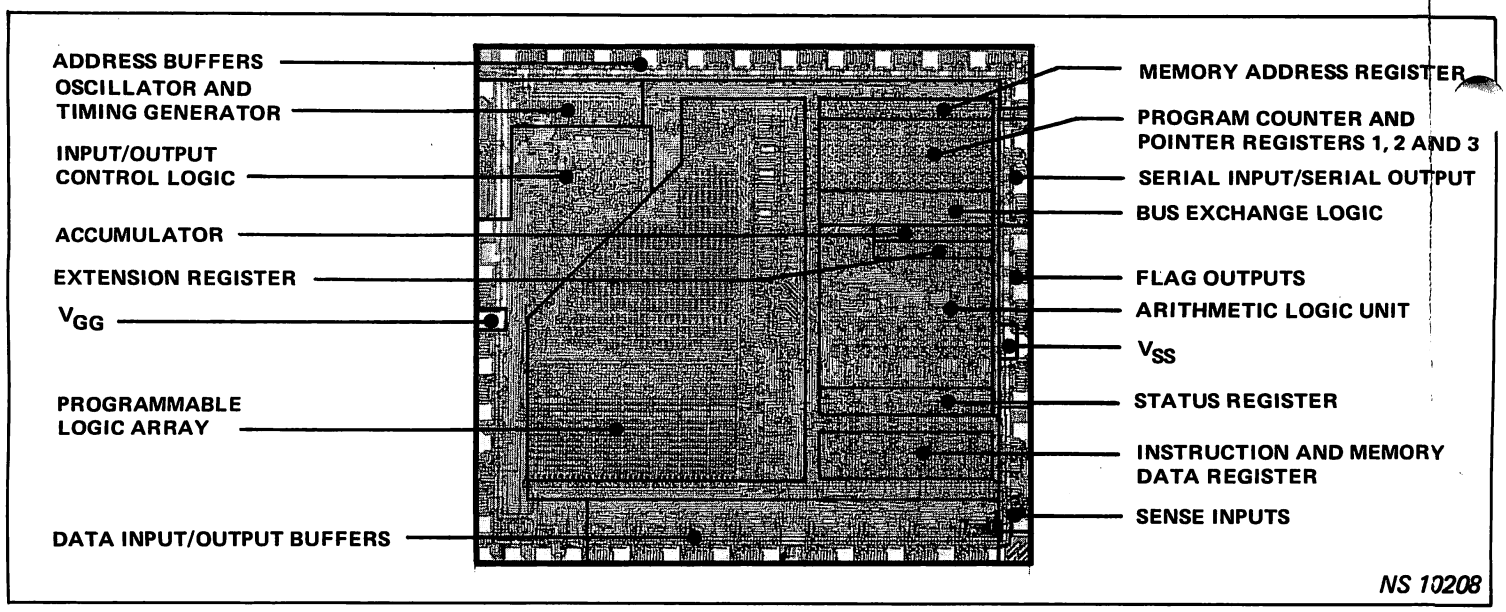


Figure 1-1. Layout of SC/MP Chip

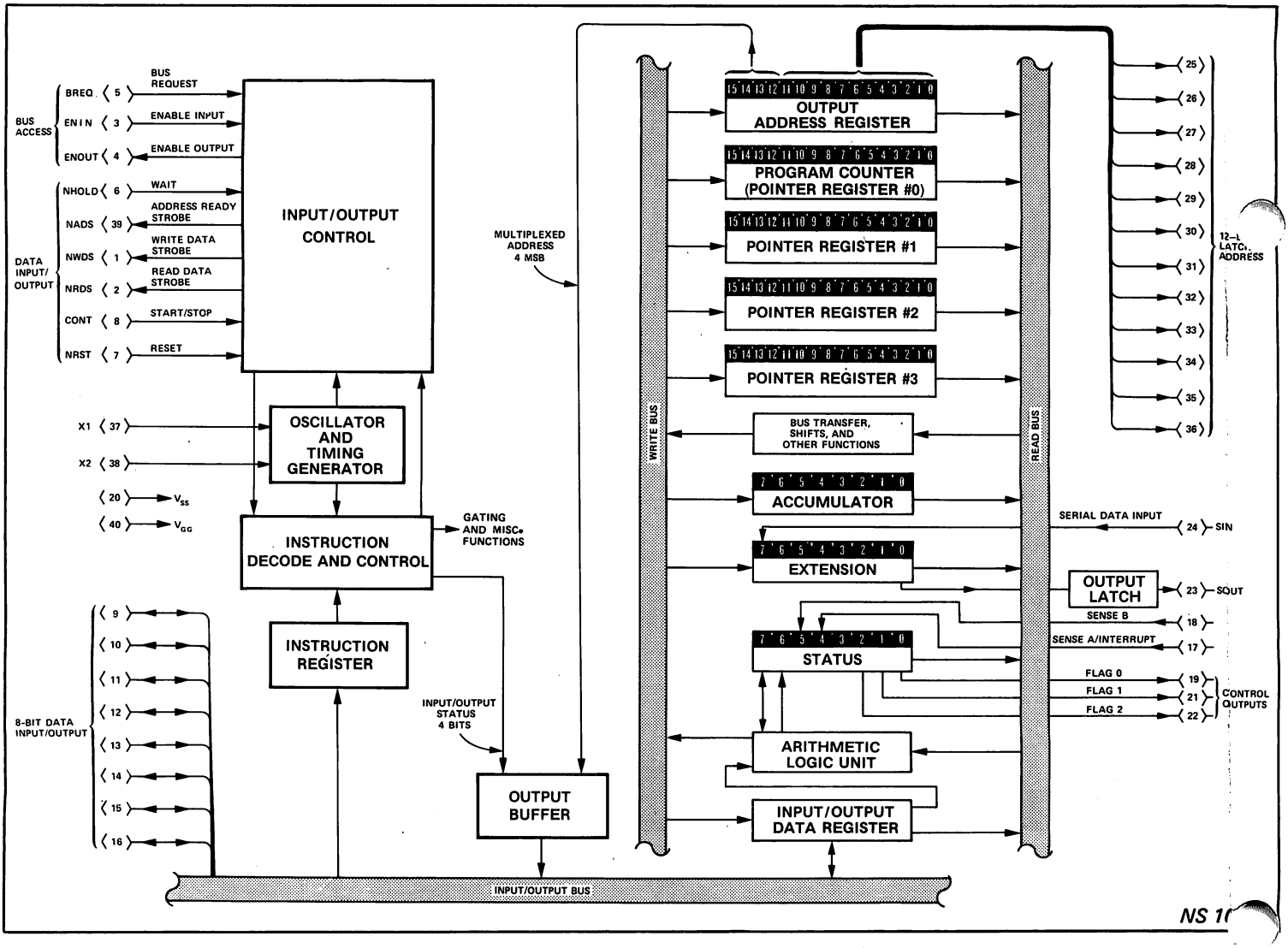


Figure 1-2. CPU Architecture and Pinouts of SC/MP

Chapter 1

INTRODUCTION TO SC/MP

1.1 GENERAL DESCRIPTION

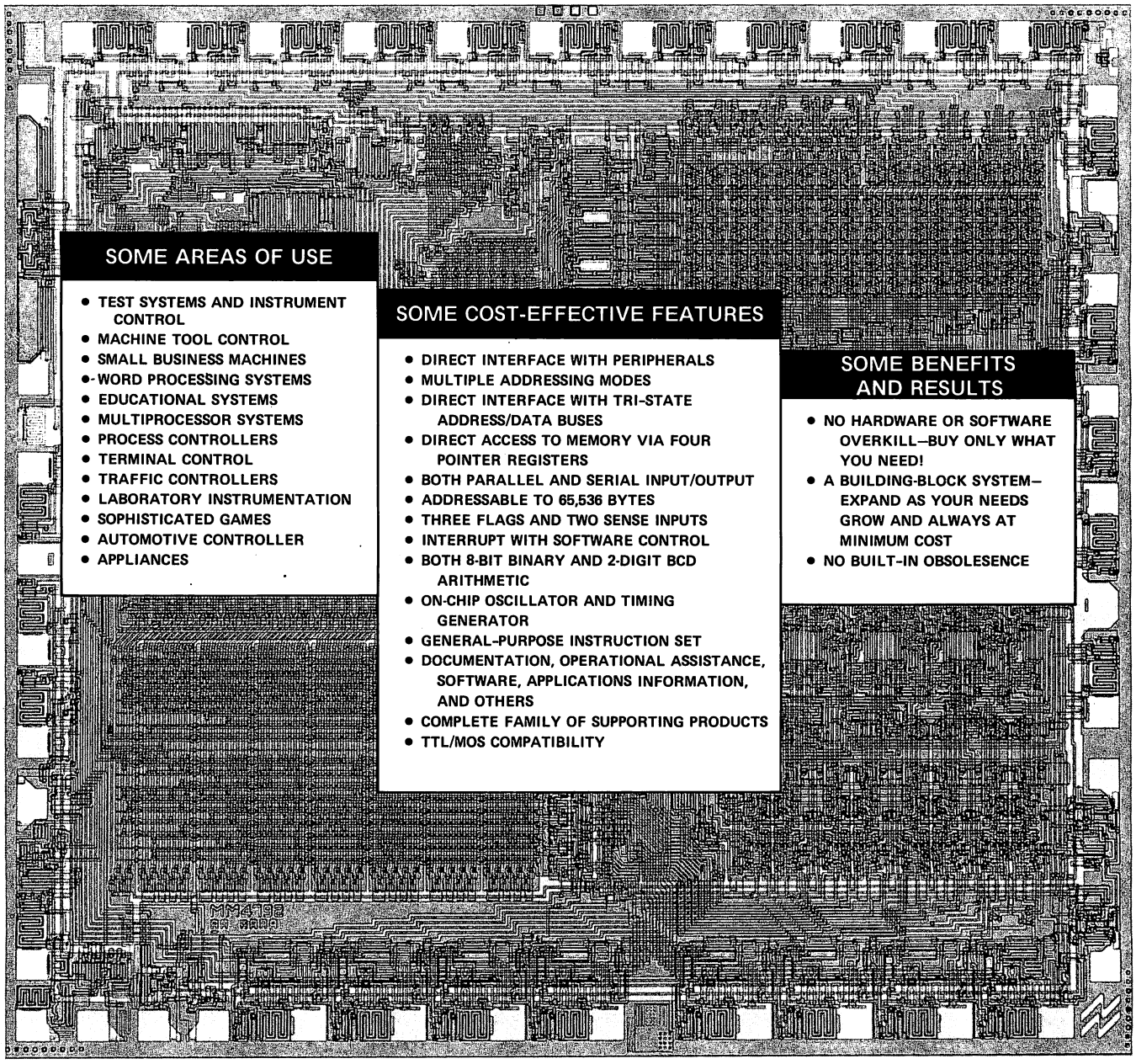
The SC/MP microprocessor is an inexpensive single-chip Central Processing Unit (CPU) with outstanding functional capabilities. SC/MP is housed in a 40-pin, dual in-line package; the chip layout is shown in figure 1-1. Figure 1-2 illustrates CPU architecture and the pinouts of the chip. Some user benefits and matching SC/MP features are listed below.

User Benefits	SC/MP Features
★ **APPLICATIONS PROGRAMS ARE EASY TO WRITE AND ARE MEMORY EFFICIENT**	• Five memory and peripheral addressing modes –Program-counter relative –Immediate –Indexed –Auto-Indexed –Implied • Four 16-bit address pointer registers –Reduces address formation overhead –Allows subroutine nesting • Forty-six instruction types – single-byte and double-byte • Software-controlled interrupt structure • Software-controlled serial input/output
★ **SUPPORTED BY COMPLETE DEVELOPMENT SOFTWARE**	• Editor • Assemblers • Loaders • Debug
★ **END-SYSTEM RELIABILITY**	• Inherent reliability of LSI devices • Low system component count • Low power consumption
★ **FULLY SUPPORTED BY NATIONAL SEMICONDUCTOR**	• SC/MP technical training offered in Miami, Dallas, and Santa Clara • Field applications engineers around the world • Factory applications engineers • Factory warranty services • COMPUTE, National Semiconductor Microprocessor Users Group
★ **LOW END-SYSTEM COST AND COMPONENT COUNT**	• Single-chip microprocessor (40-pin DIP) • On-chip oscillator and timing generator • Uses standard memories • Uses standard peripheral components
★ **SIMPLIFIED END-SYSTEM DESIGN**	• Static operation (no refresh circuits required) • Latched 12-bit TRI-STATE® address port • Direct interfacing with standard memory components, up to 65K bytes • Bidirectional 8-bit TRI-STATE® data/control port • Serial input/output port • TTL/CMOS compatible inputs/outputs • Three program-controlled output flags • Two program-testable sense inputs • Direct interfacing with standard peripheral components • Control signals for Direct Memory Access (DMA) implementation • Control signals for multiprocessor system implementation • Supported by system development aids –Development systems –Prepackaged applications cards –Complete documentation with design examples

1.2 SC/MP APPLICATIONS

SC/MP can be used in almost any application. The chip can be used in a minimal configuration that might include a few switches for control, a read-only memory for implementing instructions, and a few indicators for monitoring purposes. On the other hand, a maximum system might include several input-output peripherals, read/write as well as read-only memory, and a full-featured control panel. Whatever your application, you will find that SC/MP and its supporting products include everything needed to develop, debug, and implement *your* system.

A summary of SC/MP applications is shown below. By no means are the uses, features, and benefits all-inclusive; they are simply indicative of how applications-oriented SC/MP really is and how cost-effective it can be in solving your control problems.



1.3 ARCHITECTURE OF SC/MP

1.3.1 Hardware Summary

SC/MP provides all the basic features of a general-purpose microprocessor. The basic features include an input/output capability, a memory-access capability, a data-processing capability, and a powerful instruction set. These capabilities are implemented on a single chip. The hardware relationships with functional nomenclature are shown in figure 1-2. An overview of these functions and some extensions thereof are described in the following paragraphs.

1.3.2 Input/Output Capabilities

The input/output capabilities of SC/MP are summarized in figure 1-3. For parallel data transfers, the microprocessor communicates with user systems via a 12-bit TRI-STATE address port and an 8-bit TRI-STATE data port; two serial ports – one for input and one for output – are also provided.

Realtime control is accomplished via a sense input, an interrupt, and flags. Other hard-wired control signals provide bus access, access priority, data-flow supervision, and general control of the processor. The bus-access and access-priority control lines can be used to cascade several microprocessors – all with direct interface to the address and data buses. If SC/MP is to be the only bus controller, the bus-access control line can be hard-wired in the active state for uninterrupted access to both buses. Timing parameters affecting each input/output data transfer are described in chapter 2. The input/output control sequence of SC/MP can be summed up as follows:

- Bus request from SC/MP
- Request granted or denied – if denied, bus request remains active until bus request is granted
- Address and address-valid strobe from SC/MP
- Data valid and inputted or outputted, as required

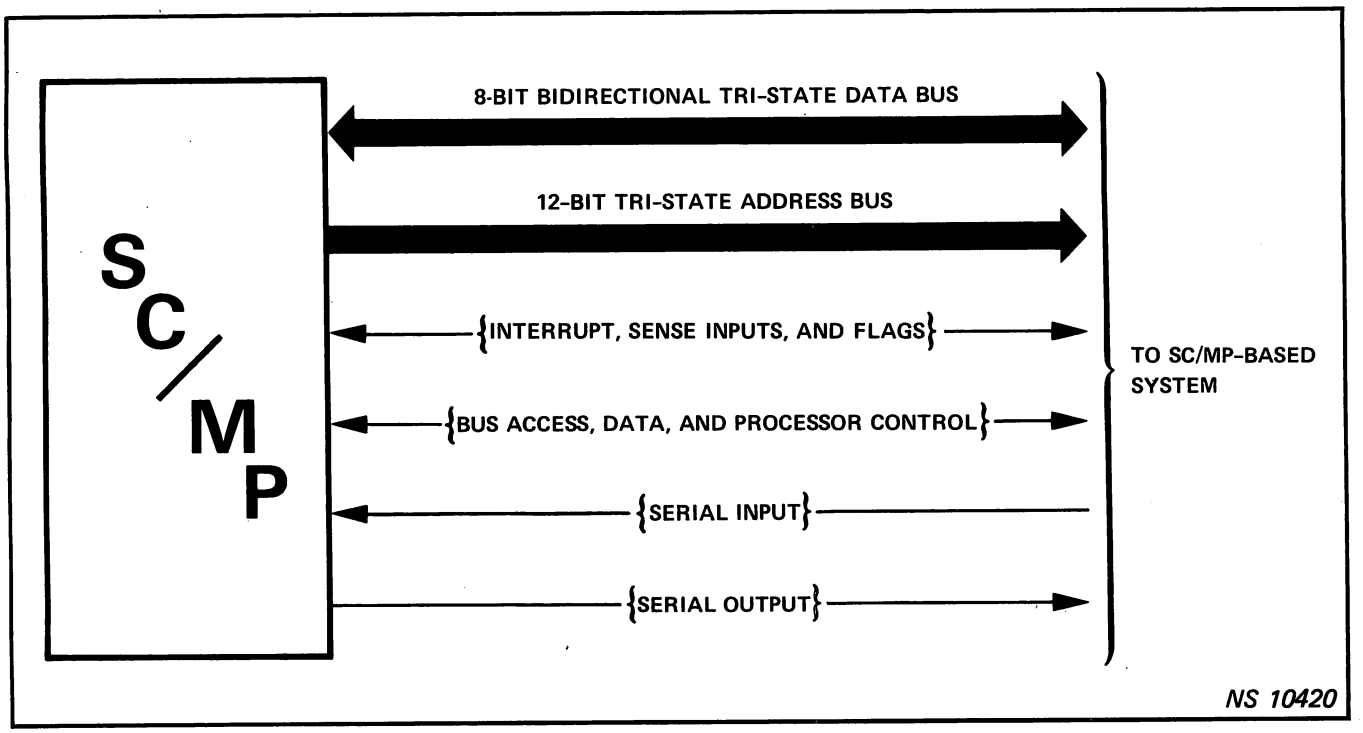


Figure 1-3. Input/Output Capabilities of SC/MP

1.3.3 Memory-Access Capabilities

The SC/MP chip has a 16-bit address capability; thus, any one of 65,536 memory locations can be discretely specified. As shown in figure 1-4, four of the address bits are sent over the data bus to select any of 16 memory "pages," and 12 bits are sent over the address bus to specify the memory location within the page. ROM, PROM, and RAM memories may be intermixed in the address space.

SC/MP uses any one of five addressing modes to generate the address: these are *program-counter (PC) relative, immediate, indexed, auto-indexed,* and *implied.* If the addressing range is not more than 127 bytes above or 128 bytes below the address specified by the program counter, the PC-relative mode of addressing can be used. In this mode, an effective address is formed by combining the contents of the program counter with the displacement field (second byte of the instruction). The immediate mode simply uses the second byte of the instruction as the memory data; this mode is relatively fast because an additional data address is not formed, nor is an additional memory access to fetch the data required. For indexed addressing, the displacement field is combined with the contents of a pointer register to form the effective address. Auto-indexed addressing is similar to indexed addressing except that the contents of the pointer register are replaced by the effective address. Refer to chapter 2 (2.4.4) for details of effective address formation. Both indexed modes are useful for block transfers, inputting and outputting of tabular data, and similar applications where the addressing range does not exceed +127 or -128 bytes relative to the contents of the referenced pointer register. The implied mode of addressing uses the operation code (OPCODE) to specify origin and destination addresses; for example, the CSA command (Copy Status Register to Accumulator – OPCODE = 06) implicitly defines the status register as the originating operand and the accumulator as the destination operand.

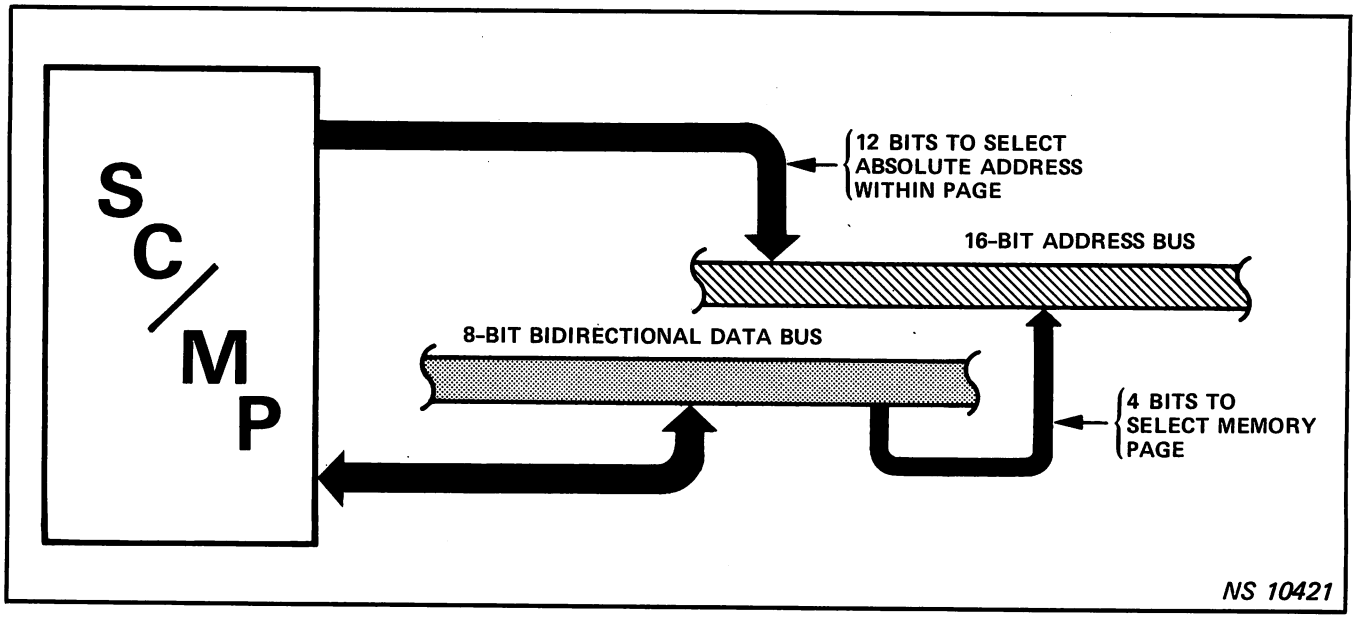


Figure 1-4. Memory-Access Capabilities of SC/MP

1.3.4 Processing (CPU) Capabilities

The CPU capabilities of SC/MP are primarily a function of the instruction set and the implementing hardware; these two functions are summarized in figure 1-5. The instruction set is general-purpose, provides a wide range of programming techniques, and is easy to use. Likewise, the implementation hardware provides input/output flexibility, a full complement of control and data registers, and simple interfaces to user systems.

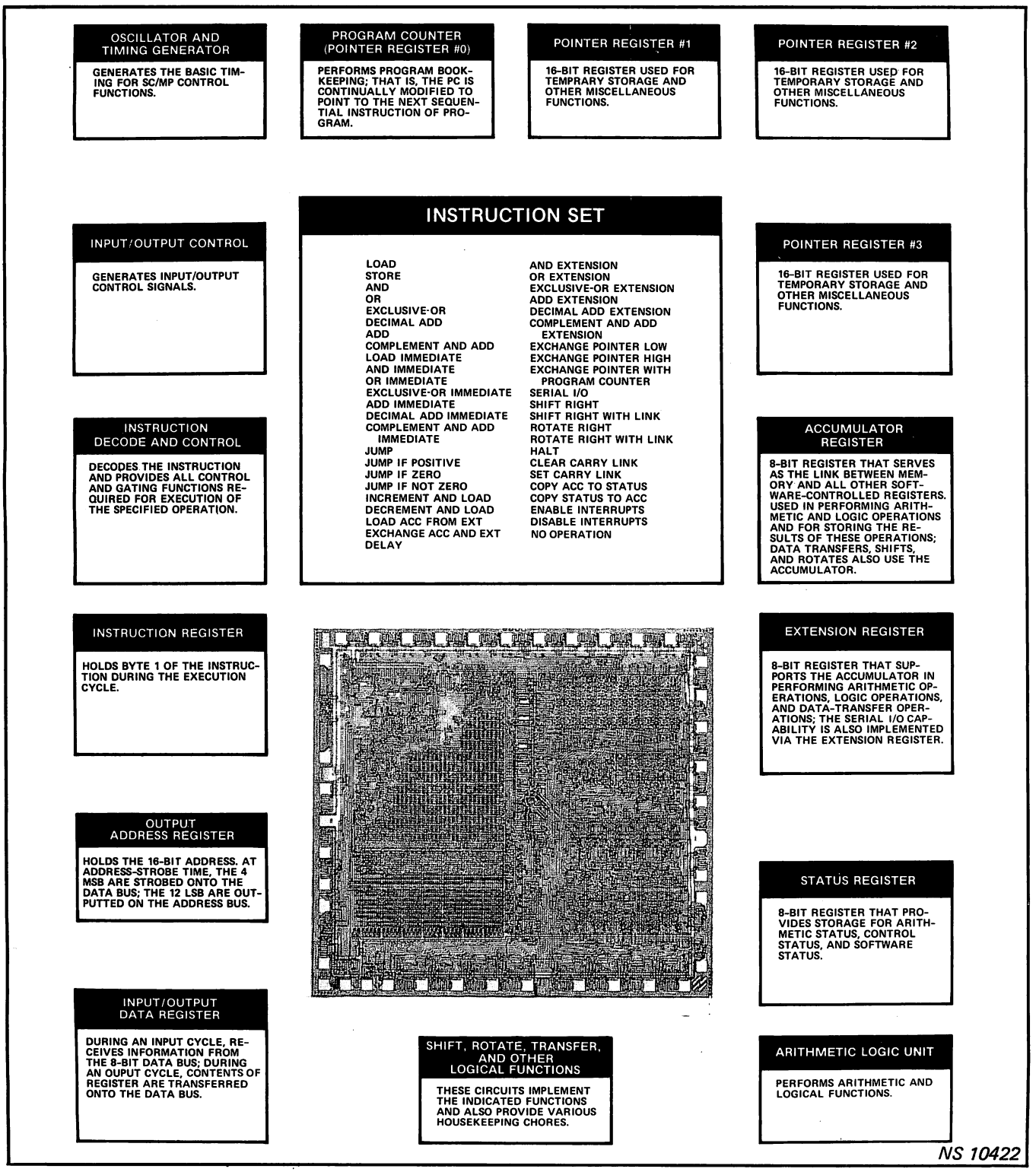


Figure 1-5. CPU Summary of SC/MP

1.4 SC/MP AND SUPPORTING PRODUCTS

1.4.1 SC/MP Development System

The upper part of figure 1-6 shows SC/MP supported by a low-cost, general-purpose, development system. This low-cost configuration is particularly suited to low-volume users for designing, verifying, and debugging hardware/software systems. To facilitate the development process, an operating panel with appropriate controls and indicators allows for the examination and modification of user software. In addition, the low-cost system provides interface peripherals and the firmware required for development of user programs; valid results can then be committed to an appropriate media (paper tape, cards, and so forth) for the generation of custom ROMs/PROMs.

1.4.2 Universal Development System

The lower part of figure 1-6 shows SC/MP supported by a universal development system. The development system provides a full complement of software, a complete line of input/output peripherals, and software/hardware features that permits the user to adapt and finalize design in virtually any microprocessor application. The universal development system is designed such that the user can load, assemble, and debug development programs via the host computer and its associated peripherals. Using the host-generated software, a viable applications-oriented system can be developed and implemented by the target (SC/MP) interface.

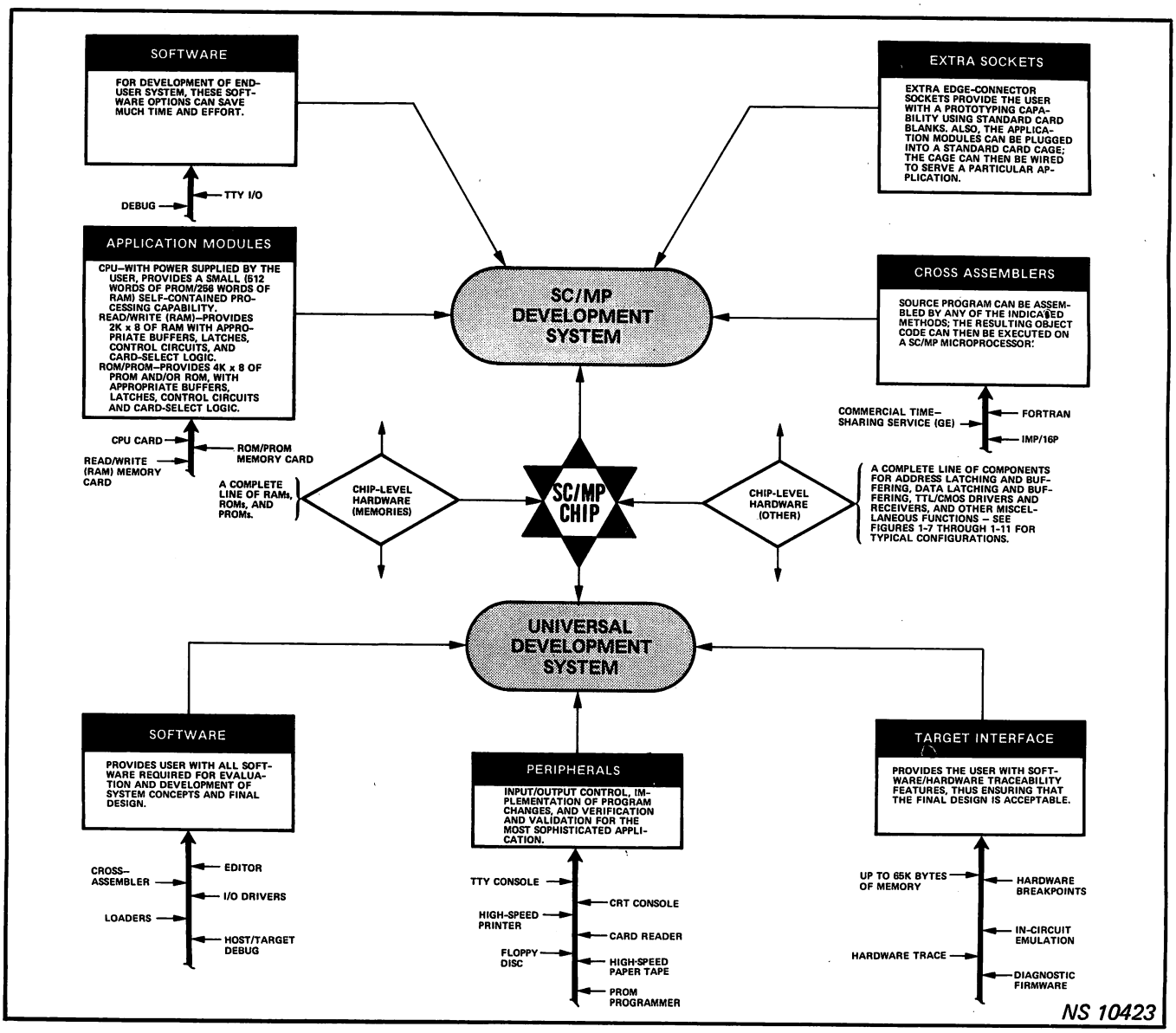


Figure 1-6. SC/MP and Supporting Products

1.4.3 Chip-Level Hardware

To extend the capabilities of SC/MP, a number of components are available; in succeeding paragraphs, these components are described functionally and configured to show their use in typical applications.

1.4.3.1 Buffers

In applications where loading and fanout requirements do not exceed capabilities of the chip, SC/MP can be directly connected to the address and data buses. Where more than one TTL load must be serviced, the address, control, and data lines can be buffered as shown in figure 1-7. The indicated buffering components provide a medium-range fanout capability and the data-bus buffer is bidirectional; thus, these components can be used in both input and output applications.

1.4.3.2 Latches

Latches are useful in expanding addressing and peripheral-interface capabilities; they are also useful in TRI-STATE bus interfacing. An example of how address and control line expansion could be accomplished is shown in figure 1-8. With properly chosen buffers and latches, this configuration can easily accommodate up to 65K bytes of memory with relatively simple timing and control schemes.

An example that uses latches for interfacing is shown in figure 1-9. The interface latch chip can be used as a bidirectional input/output port, a dedicated input port, or as a dedicated output port; a control signal (dynamic for bidirectional mode and static for dedicated modes) is supplied by the user. In the configuration shown, separate control lines enable the input/output ports; thus, the latch may be a high-impedance device (that is, it does not load the system bus unless an enable signal is present).

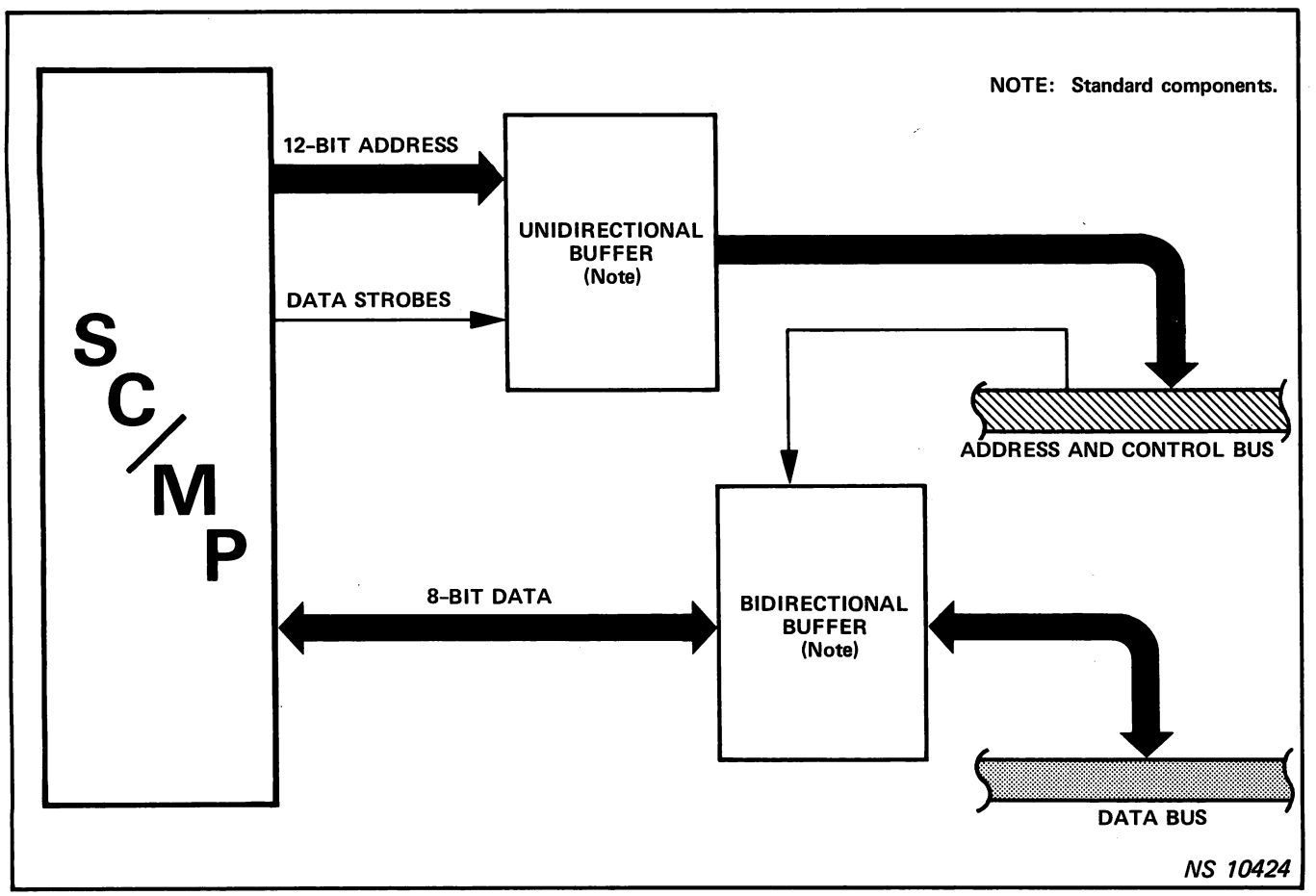


Figure 1-7. Fanout Buffering of SC/MP to System Buses

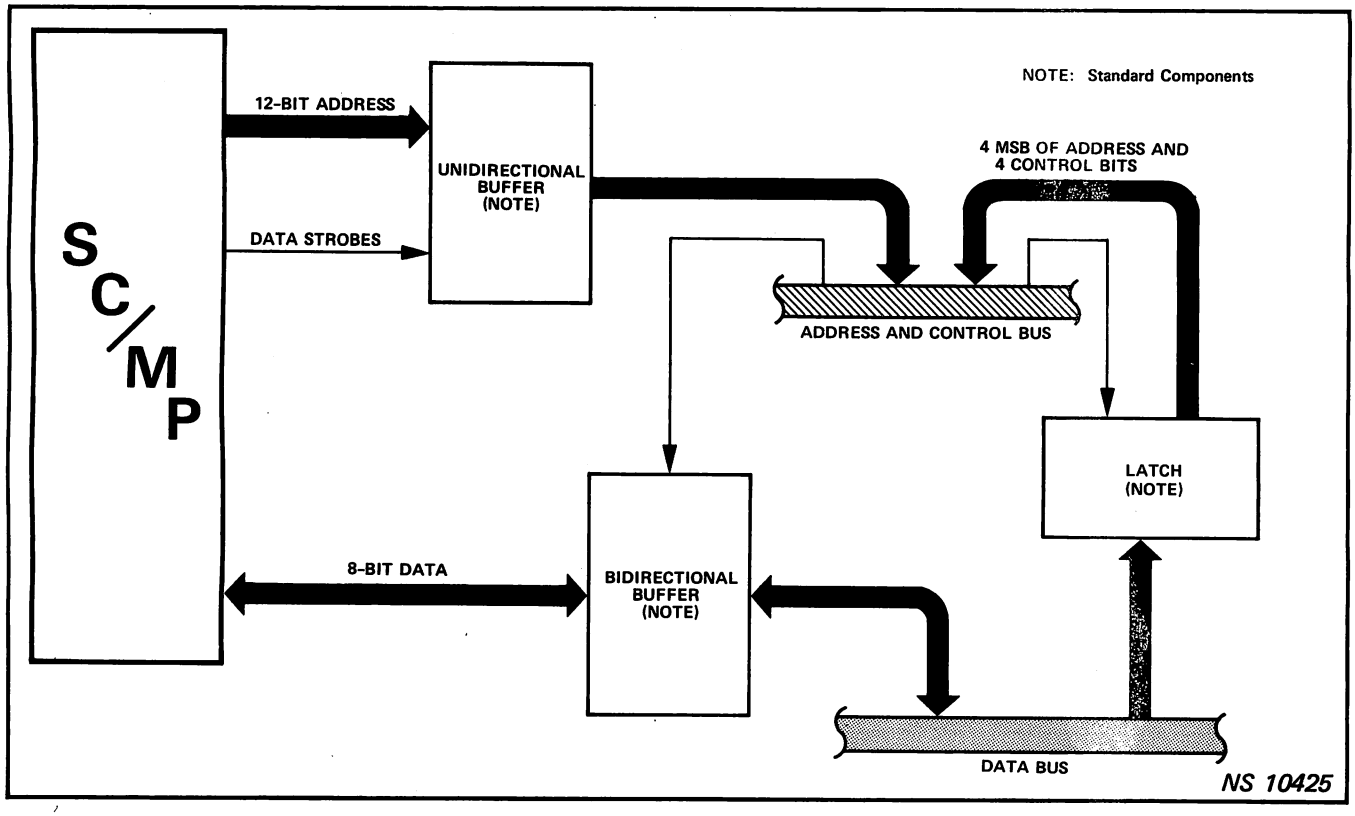


Figure 1-8. SC/MP Using General-Purpose Latch to Expand Address/Control Lines

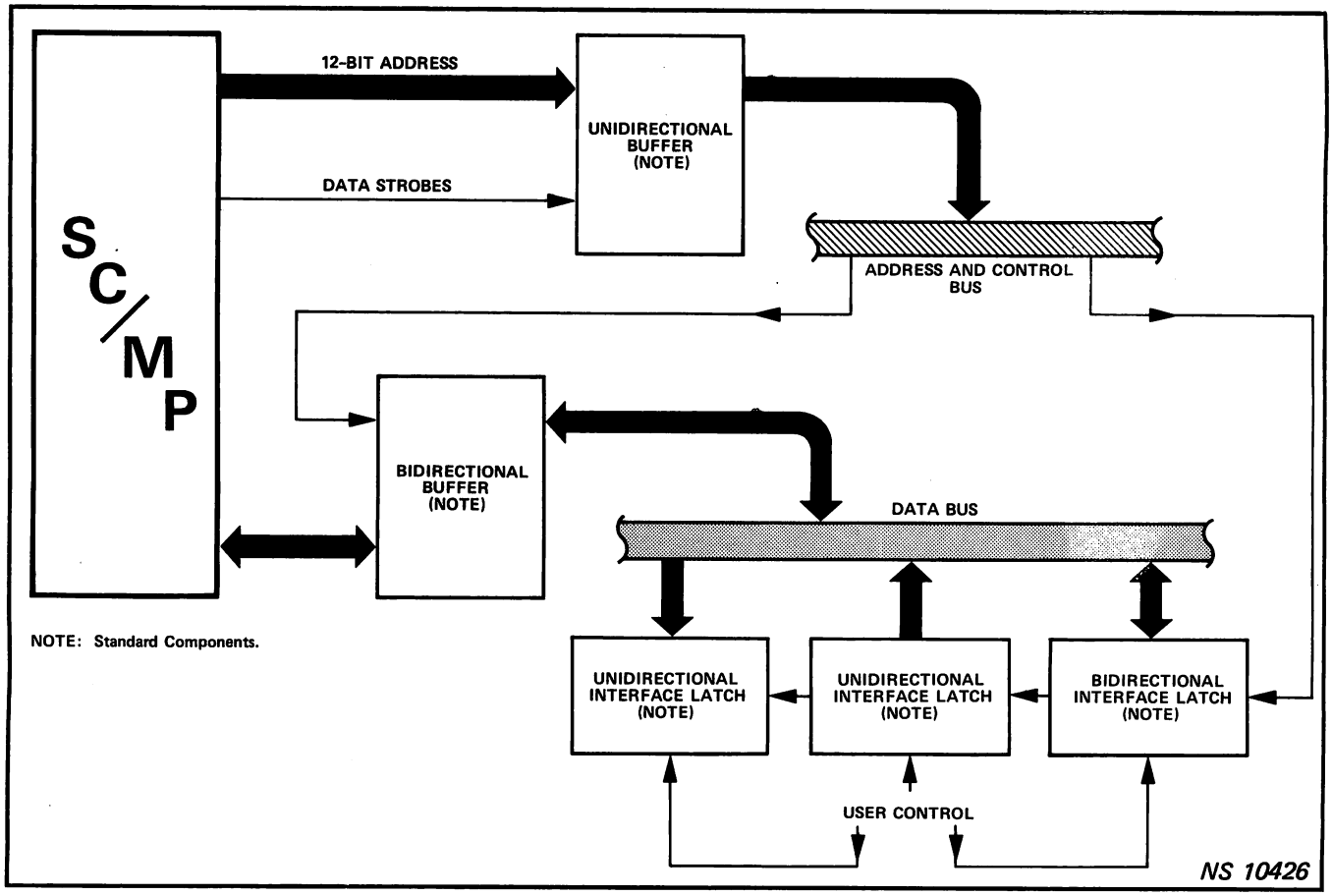


Figure 1-9. SC/MP Using Interface Latch For Input/Output Device

1.4.3.3 Memory Devices

SC/MP is directly compatible with many standard memory components – Random Access Memories (RAMs), Read Only Memories (ROMs), and Programmable Read Only Memories (PROMs). Memory accessing is under control of the SC/MP microprocessor, which provides input/output and other control signals via the address/control bus – see figure 1-10. These control signals permit the memory chips to accept address information or to implement input/output data transfers. Depending on bus loading, buffering circuits may or may not be required.

Some PROM memory chips are pin-for-pin compatible with ROM and, when used, provide a user-programmable memory that is cost-effective for program development and other low-volume applications.

Figure 1-11 shows SC/MP configured with standard component chips. As indicated, SC/MP can be expanded to serve large system needs.

1.5 SC/MP APPLICATION MODULES

The following application modules support system design around the SC/MP microprocessor.

- SC/MP CPU Application Module
- SC/MP RAM Application Module
- SC/MP PROM/ROM Application Module

Each module measures 4.375 by 4.852 inches and can be inserted into a standard card cage – refer to table 3-1 for this and other vendor-supplied accessories. The applications modules can easily be connected to form the basis of a custom-designed system or they can be used in stand-alone end applications. The modules are particularly well suited to portable equipment where physical space is limited. For functional details of the CPU, RAM, and ROM/PROM modules, refer to chapter 3.

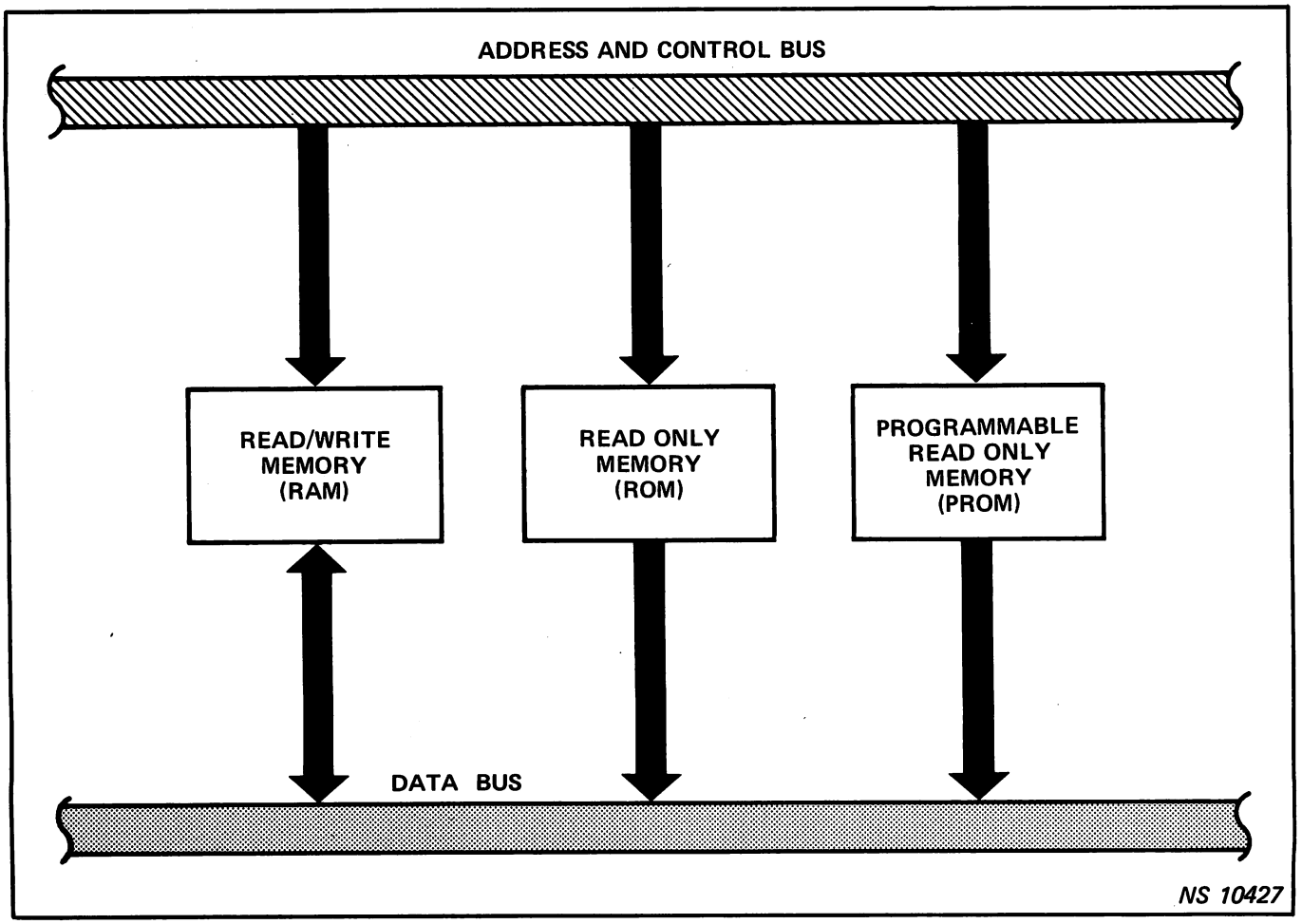


Figure 1-10. SC/MP Memory Chips

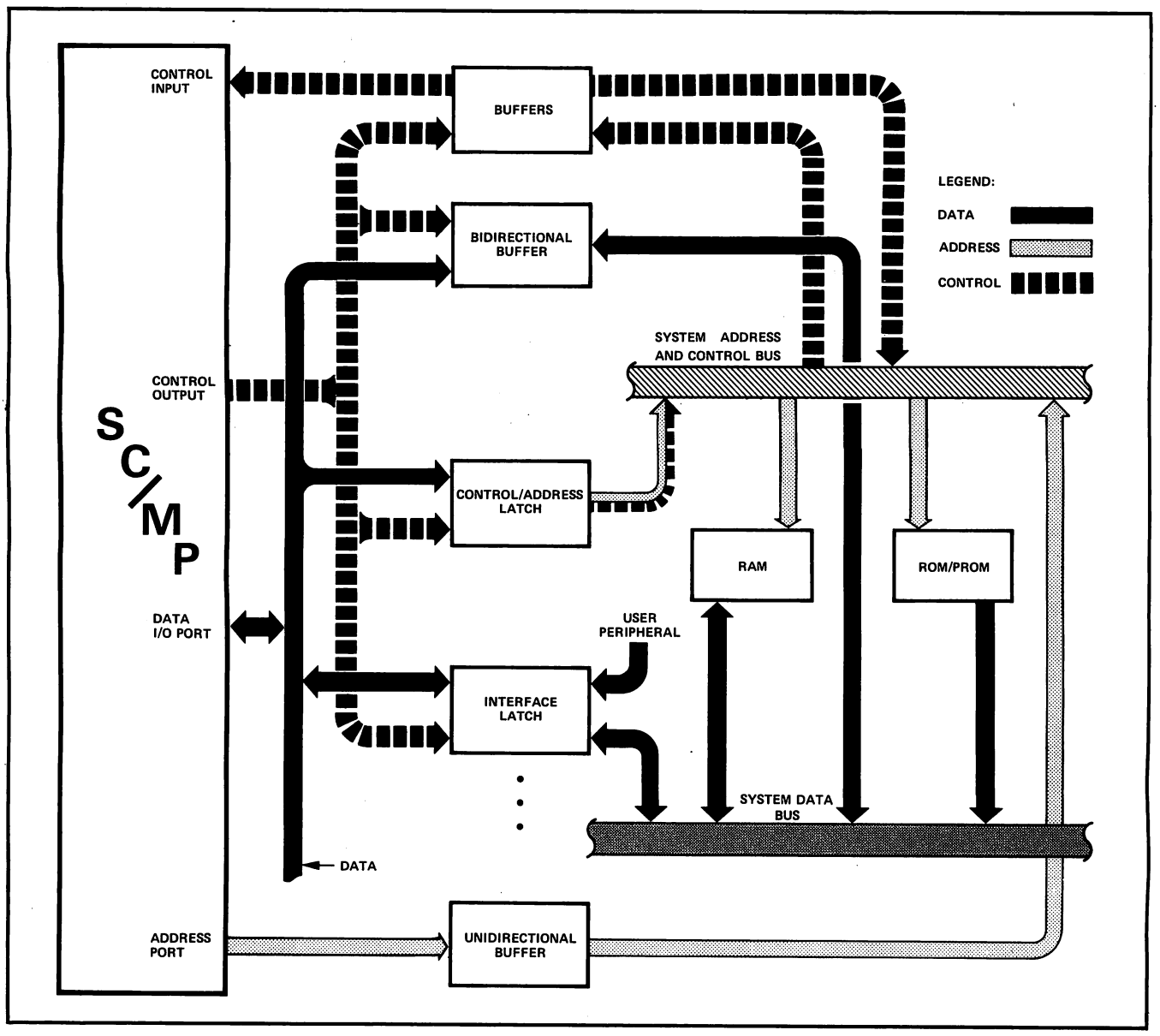


Figure 1-11. Typical Configuration of SC/MP with Supporting Chips

1.6 SC/MP SOFTWARE

The importance of software support cannot be overemphasized. System development to meet a particular application is most efficient when the designer fully appreciates and uses the support software. At present, SC/MP software includes cross assemblers, loader/debug utilities, and input/output routines; supporting software is described in the paragraphs that follow.

1.6.1 (IMP-16) Cross Assembler

The cross assembler accepts free-format statements from either a keyboard, a paper tape, or a card reader; each program produces a load module (LM) on paper tape and a program listing (figure 1-12a). The assembler requires three passes over the source program; however, if either the object listing or the LM is suppressed, only two passes are required. Depending on the system configuration running the cross assembler, this may or may not be apparent to the user.

1.6.2 (FORTRAN) Cross Assembler

This program, written in FORTRAN IV (USA Standard Language Subset), assembles a source program on a host computer for subsequent execution by SC/MP. The FORTRAN cross assembler accepts free-format source statements and, in two passes, produces a load module (object program) and a program listing (figure 1-12b). This cross assembler is installed and available to users of General Electric national time-sharing service under the program name, SAS$$$.

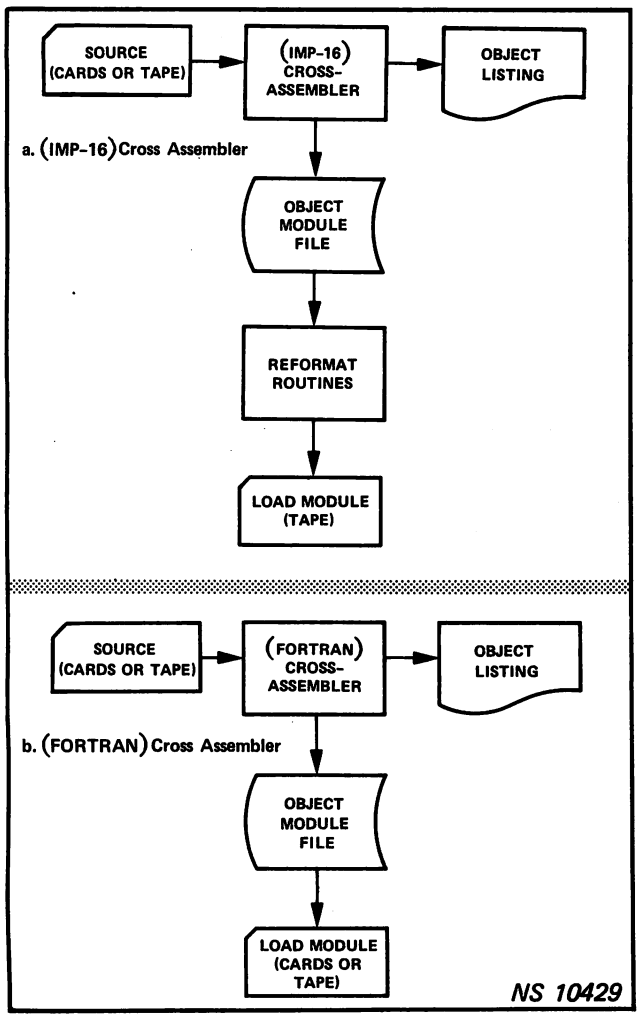


Figure 1-12. SC/MP Cross Assemblers–Operational Flow Diagrams

1.6.3 Absolute Loader

An absolute loader loads one or more programs into pre-allocated, fixed areas of memory. The exact memory areas to be occupied by each user-generated program must be determined by the user before assembly. Also, any linking of one program to another or to common, shared data must be accomplished at assembly time by assignment of common labels to fixed, absolute addresses in memory.

1.6.4 SC/MP Teletype Routines

These routines are used to send and receive information to and from the Teletype (TTY) or to receive data from the Paper Tape Reader/Punch. One routine transfers keyboard inputs to the processor without a character echo, whereas the other routine echos the received character back to the TTY printer.

1.6.5 SC/MP Debug Program

The SC/MP Debug Program supervises the operation of a user program during checkout. This program provides the following facilities for testing computer programs:

- Printing selected areas of memory in hexadecimal format
- Modifying the contents of selected areas in memory
- Displaying and modifying CPU registers
- Inserting instruction breakpoint halts
- Initiating execution at any point in a program

1.7 CUSTOMER SUPPORT

At National Semiconductor, we believe that the *product* and *its support* are an infrangible partnership and that both are equally important. Consequently, our customer-support organization is structured to provide the best possible assistance both *before* and *after* sales. Available services are summarized below; an in-depth look at our customer-support organization is provided in chapter 5.

- FIELD SUPPORT – On-site technical assistance (domestic and international) is provided by engineers that specialize in microprocessors and microprocessing systems.
- FACTORY SUPPORT – Home-based engineers provide the field specialists with hardware and software support and, when necessary, they provide direct support to the user.
- FACTORY SERVICE – Repair of any microprocessor product supplied by National Semiconductor. The factory-repair service is directly applicable to OEM customers; this service is also available for end users if the product is returned through the OEM supplier or through an authorized distributor.
- TRAINING – Elementary and advanced training courses are offered in the West, Midwest, and Eastern regions of the United States; in-depth coverage and hands-on experience are provided.
- SOFTWARE SUPPORT – Loaders, assemblers, debug routines, diagnostics, and other software are available to assist the user in SC/MP design and SC/MP implementation.
- USER GROUP – Provides a vehicle of communication between users of microprocessors and National Semiconductor. A *user-group software library* shared by all members is an important feature of this program.
- DOCUMENTATION – A *technical description* provides in-depth coverage such that benchmarks can be established, programs can be written, and preliminary design of systems can be accomplished. A *users manual* describes the use of SC/MP equipment and software. A *data sheet* provides a functional description of the SC/MP chip and includes parametric specifications.

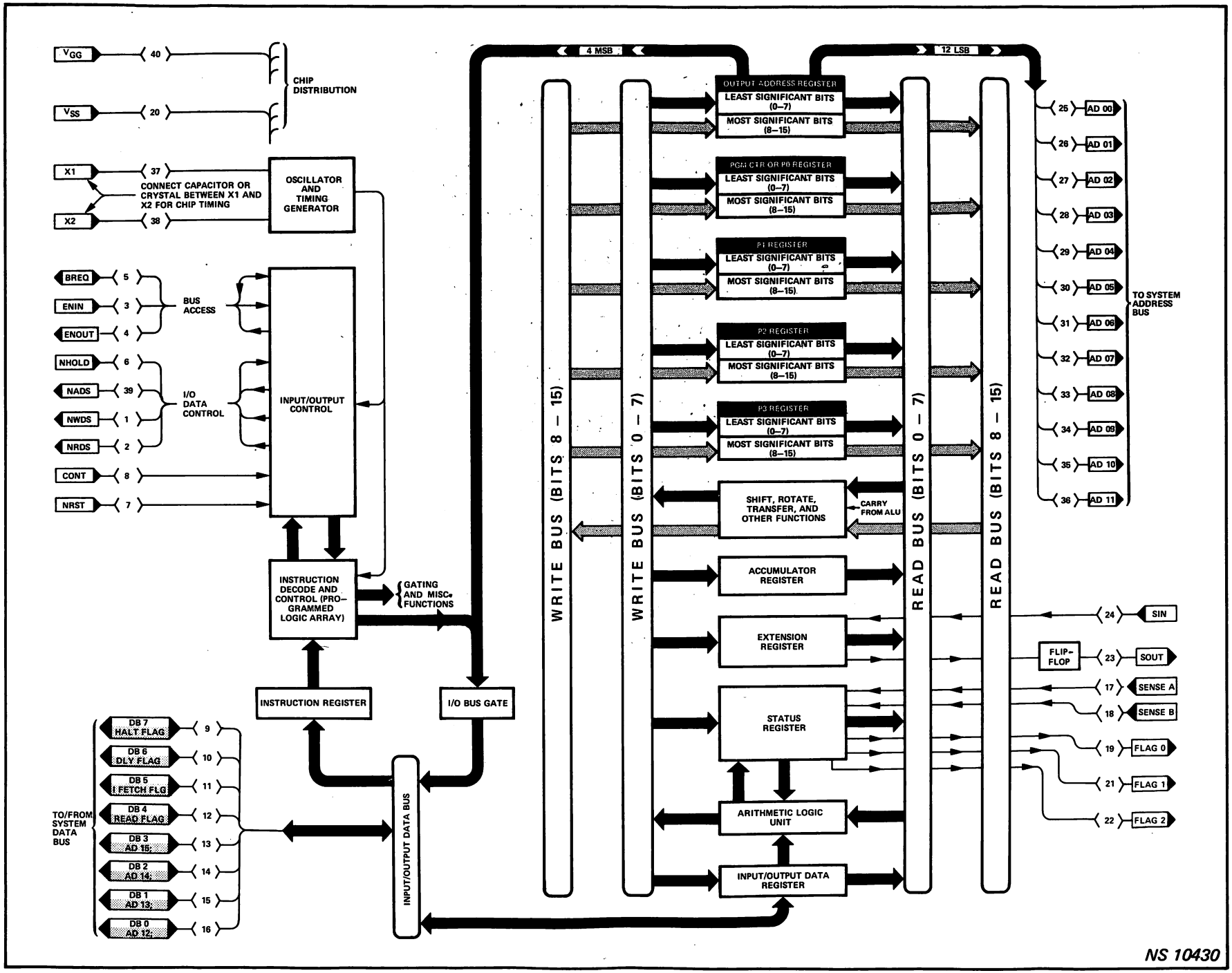


Figure 2–1. SC/MP Functional Block Diagram with Pinouts

Chapter 2

THE SC/MP CHIP

2.1 FUNCTIONAL OVERVIEW

Figure 2-1 is a functional block diagram of the SC/MP chip. Input/output signals shown on figure 2-1 are described in table 2-1. The various functional units shown are described in this chapter under the following major headings: (1) Power and Timing Control, (2) Input/Output Control, and (3) Internal Control and Data Movement. These descriptions are oriented to viewpoints of user operation and application.

The SC/MP instruction set is described in detail in appendix A.

NOTES

1. Positive logic convention is used throughout this manual. A logic '1' or high signal corresponds to a more-positive voltage level. A logic '0' or low signal corresponds to a more-negative voltage level. All signal names beginning with '*N*' or followed by an asterisk (*) denote complemented signals that are asserted or activated by a logic '0'. Otherwise, signals are asserted by a logic '1'.
2. Bits are numbered from 00 to 15, right to left, with bit 00 representing the least significant bit.
3. The *X'* preceding a value denotes the hexadecimal numbering system.

2.2 POWER AND TIMING CONTROL

As shown in figure 2-2, the SC/MP chip can be powered from a single dc voltage: -12 volts with respect to V_{SS}. For direct interface with TTL and 5-volt CMOS peripheral circuits, a power arrangement similar to that shown in figure 2-2a is recommended; for 5-volt devices and 17-volt PROMs, a satisfactory arrangement is shown in figure 2-2b.

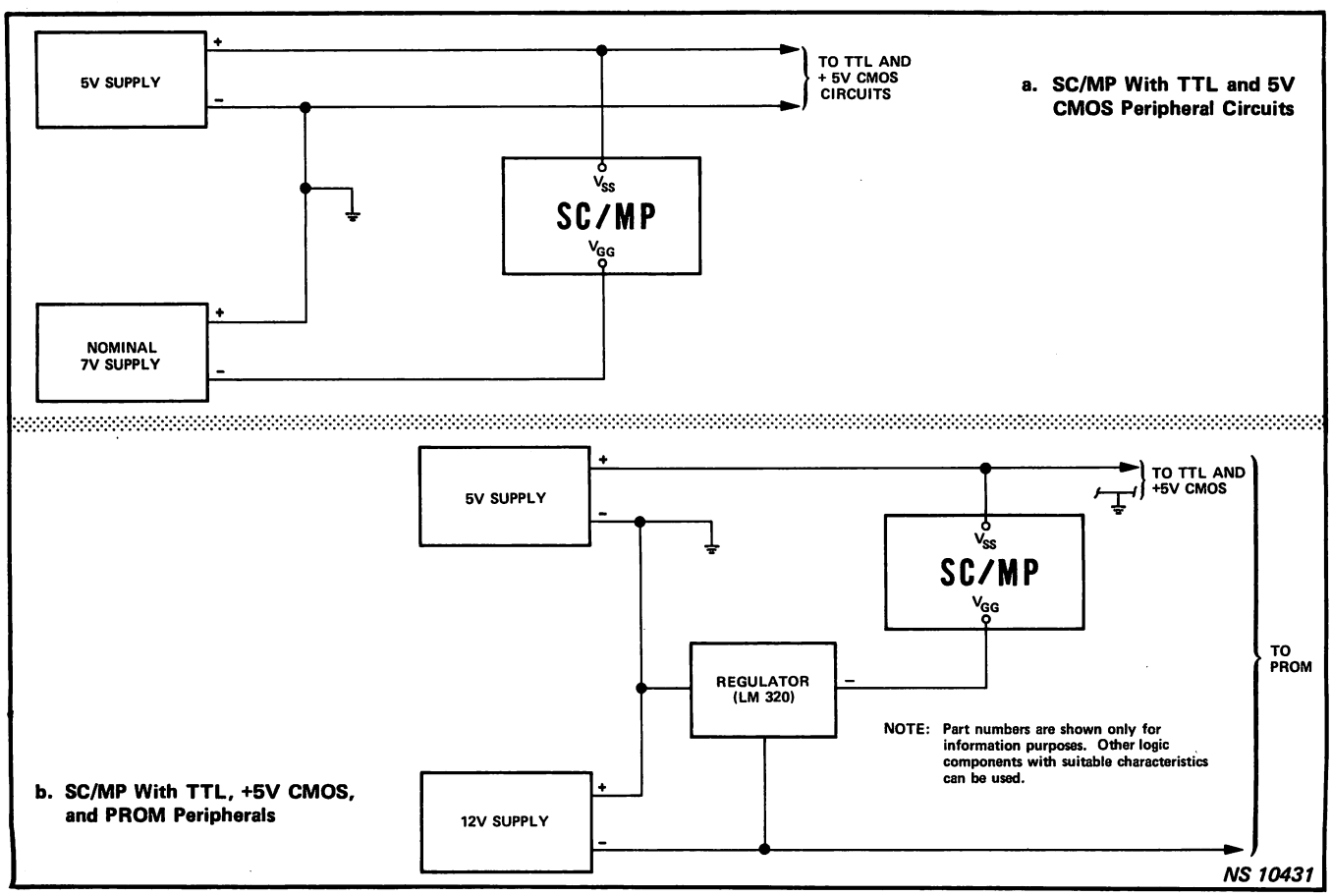


Figure 2-2. SC/MP Power Requirements, Typical Configurations

All necessary timing signals are provided by an on-chip oscillator and timing generator. If precision timing is unimportant, a capacitor is connected between X1 and X2. In applications where better timing accuracies are required, a crystal must be connected between X1 and X2. See note below.

NOTE

Quartz crystals required for SC/MP operation should be hermetically sealed. These crystals usually are in "HC" series holders (industry standard) and can be obtained from many manufacturers. Four such manufacturers are as follows:

1. X-Tron Electronics, Hayward, California
2. M-Tron Industries, Yankton, South Dakota
3. Crystek Crystal Co., Ft. Myers, Florida
4. JAN Crystals, Ft. Myers, Florida

2.3 INPUT/OUTPUT CONTROL

The data and address ports of the SC/MP chip are connected to input/output devices via two system buses – an 8-bit bidirectional data bus and a 12-bit address bus. High-speed data transfers are made in parallel; however, for slow-rate peripherals, a serial input/output capability is also provided. Refer to table 2-1 for descriptions of the SC/MP chip pinouts.

Table 2-1. Input/Output Signal Descriptions

Signal Mnemonic/ Pin Designation	Functional Name	Description
X1/X2		Connect capacitor or crystal between X1 and X2 for chip timing.
V_{SS}		Positive supply voltage—see figure 2-2 for typical power hookups.
V_{GG}		Negative supply voltage—see figure 2-2 for typical power hookups.
NRST (Input)	Reset	When low, aborts in-process operation; when returned from low to high, initializes (zeros) all internal registers and next instruction is fetched from memory location 0001_{16}.
CONT (Input)	Continue	When high, instruction is fetched from address specified in the program counter and executed. When low, processor operation is halted prior to next instruction fetch; accordingly, fetch and execution cycles can be manually implemented in a single-instruction mode.
BREQ (Input/Output)	Bus Request	Part of simplified input/output bus-interface logic. Depending on system configuration, the BREQ line is used as a bus-request and/or as a bus-busy signal. The BREQ "wired-AND" line requires an external load resistor to V_{GG} or ground.
ENIN (Input)	Enable In	When high, the processor is granted access to the input/output bus. When low, access is denied.
ENOUT (Output)	Enable Out	When high, indicates that ENIN is high and that the CPU does not have access to the bus. When low, indicates that CPU has access to the bus or that ENIN is low.
NADS (Output)	Address Strobe	When low, the 4-bit input/output status and the 4 most significant bits of 16-bit address are valid on the system data bus.
NRDS (Output)	Read Strobe	When low, data are strobed from the system data bus into the processor; when high, processor is not reading from the input/output bus. This line is a high-impedance (open-circuit) load when SC/MP does not have access to the input/output bus.
NWDS (Output)	Write Strobe	When low, data are valid from the processor on the system input/output data bus. When the processor does not have access to the input/output bus, the NWDS line is a high-impedance (open-circuit) load.
NHOLD (Input)	Hold or Extend	When low, extends the input/output cycle until signal goes high—in effect, delaying the trailing (rising) edge of the NRDS (or NWDS) pulse to permit interface with peripherals whose input/output characteristics are not as fast as those of SC/MP.

Table 2-1. Input/Output Signal Descriptions (Continued)

Signal Mnemonic/ Pin Designations	Functional Name	Description
SENSE A/SENSE B	Sense Input – Interrupt Request Sense Input	Two input lines that are sampled by testing bits 4 and 5 in the status register; these lines are synchronously tested. Sense A serves as an interrupt request line if interrupts are enabled by the software.
SIN	Serial Input to E-Register	Under software control, data on this line are "right-shifted" into the E-Register by the SIO instruction.
SOUT	Serial Output from E-Register	Under software control, contents of E-Register are "right-shifted" onto the SOUT line by the SIO instruction; the output is latched to maintain valid data between SIO commands.
FLAG 0, FLAG 1 AND FLAG 2	Flag Outputs	Flags 0, 1, and 2 correspond, respectively, to bits 0, 1, and 2 of the status register; these bits are available for user-designated functions.
AD00–AD11	Address Bit 00 through Address Bit 11	Twelve TRI-STATE® output lines; at NADS (address strobe) time, valid address signals appear on these lines. The address remains valid through the trailing edge of the read (NRDS) or the write (NWDS) strobe. [*Note:* The address lines are a high-impedance (open-circuit) load when SC/MP does not have access to the input/output bus.]

Signal Mnemonic/ Pin Designations	Output at NADS Time			Input at NRDS Time	Output at NWDS Time
	Mnemonic	Functional Name	Description		
DB 0	AD 12	Address Bit Number 12	Fourth most significant bit of 16-bit address.	↑	↑
DB 1	AD 13	Address Bit Number 13	Third most significant bit of 16-bit address.		
DB 2	AD 14	Address Bit Number 14	Second most significant bit of 16-bit address.		
DB 3	AD 15	Address Bit Number 15	Most significant bit of 16-bit address.	Input data are expected on the eight (DB 0–DB 7) lines.	Output data are valid on the eight (DB 0–DB 7) lines.
DB 4	RFLG	R-FLAG	When high, data-input cycle is starting; when low, data-output cycle is starting.		
DB 5	IFLG	I-FLAG	When high, first byte of instruction is being fetched.	↓	↓
DB6	DFLG	D-FLAG	When high, indicates delay cycle is starting; that is, second byte of DLY instruction is being fetched.		
DB 7	HFLG	H-FLAG	When high, indicates that HALT instruction has been executed. (In some system configurations, the H-Flag output is latched, and, in conjunction with the CONTinue input, provides a programmed halt.)	Note The DB 0 through DB 7 (AD 12 through HFLG) lines are high-impedance (open-circuit) loads when SC/MP does not have access to the input/output bus.	

2.3.1 Bus Access

Before SC/MP can transfer data to or receive data from memories or other peripherals, it must have access to the system address/control bus and the system data bus. A simple but effective means of controlling the buses is shown in figure 2-3. Bus access is controlled by three signals – bus request (BREQ), enable input (ENIN), and enable output (ENOUT). For simple systems, BREQ and ENOUT need not be used, and ENIN can be permanently enabled; refer to notes on figure 2-3.

With a hookup similar to that shown, bus access is always controlled by SC/MP and data transfers from one peripheral to another must go through the processor.

In larger systems, especially those with peripherals that feature high-speed data transfers, Direct Memory Access (DMA) is a method frequently used to effect data transfers between peripherals and memory. Using this technique, data transfers can be directly implemented without involving SC/MP (other than control functions). Typical DMA configurations and related control-signal timing are shown in figure 2-4. In figure 2-4a, bus access is controlled by logic circuits in the external DMA controller. As shown in the associated timing diagram, SC/MP requests bus-access by making the BREQ line high. This signal alerts the DMA controller, and if the bus is "idle" (no peripheral with bus-access), the enable input (ENIN) line goes high and bus access is granted. With both BREQ and ENIN lines high, SC/MP can use the address and data buses without interruption until the input/output operation is completed; at this time, the BREQ line automatically goes low and bus-access is terminated. In like manner, the DMA peripherals use the "peripheral request" and "peripheral enable" lines.

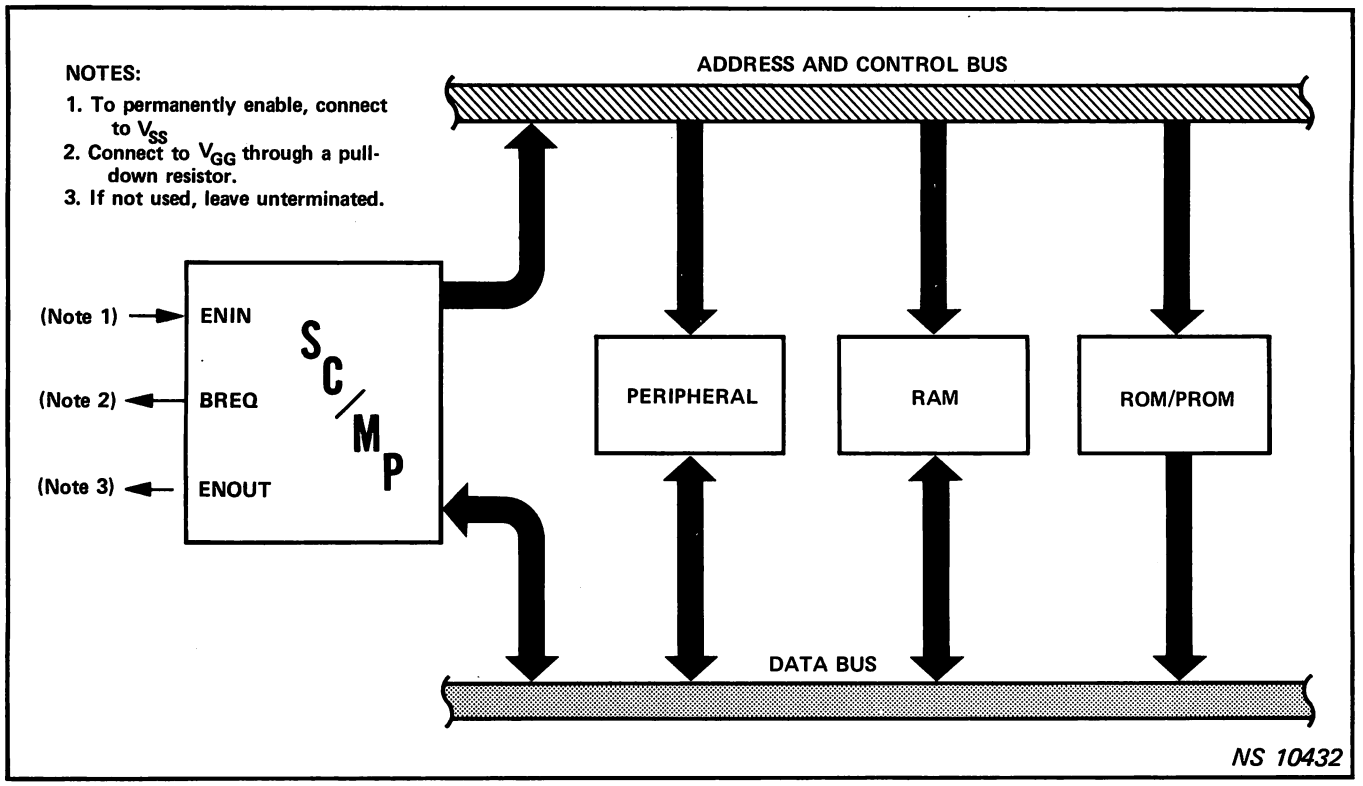


Figure 2-3. SC/MP-Controlled Bus Access

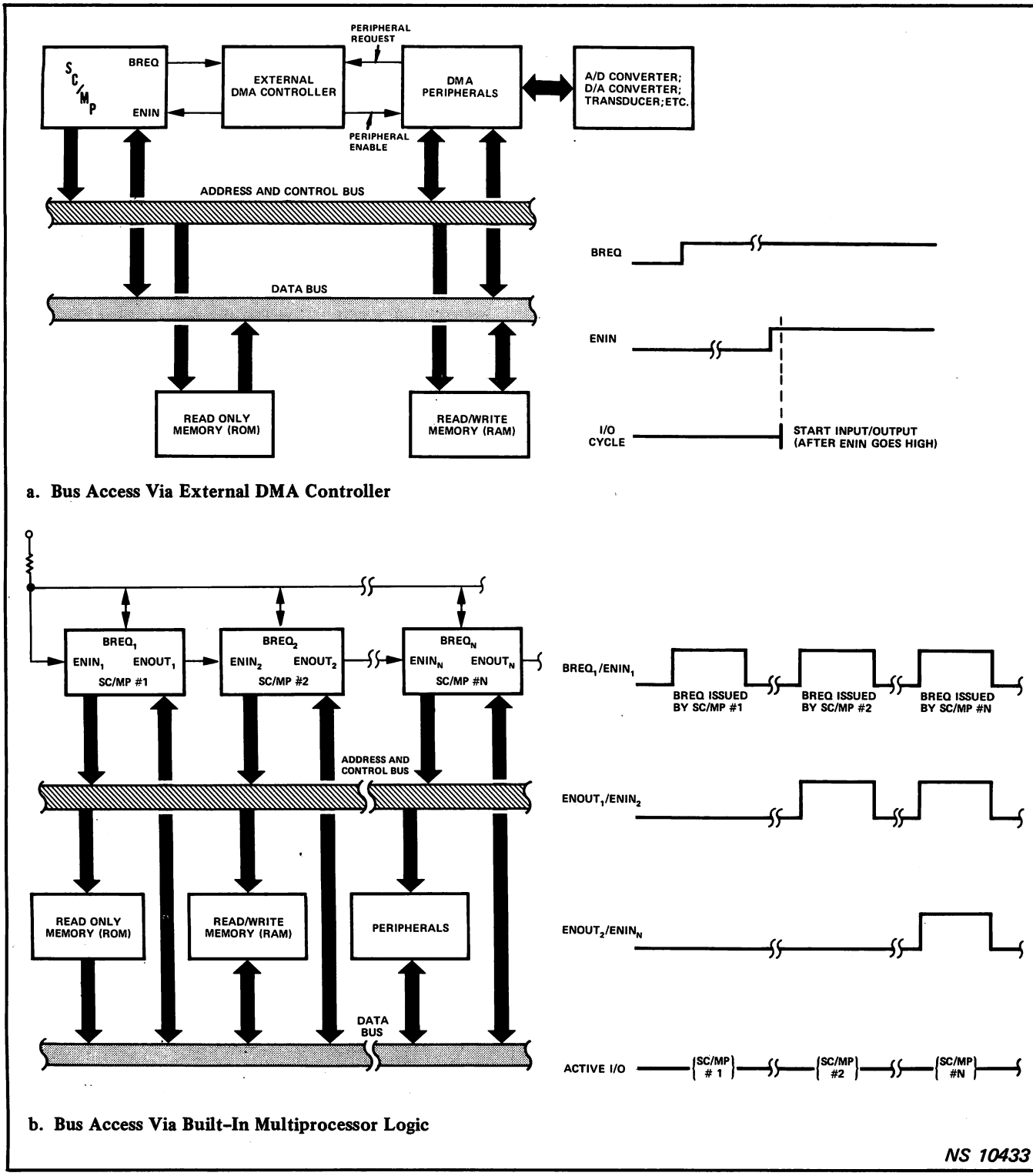


Figure 2-4. Typical DMA Configurations

Figure 2-4b shows a configuration where the external DMA controller in figure 2-4a is replaced by logic built into SC/MP. Three control signals (BREQ, ENIN, and ENOUT) provide both bus-access and priority-select functions. The enable input (ENIN) for SC/MP Number 1 is tied to the wire-ANDed BREQ. Thus, if a bus request is issued by the Number 1 processor, it has priority and controls the bus; that is, the input/output cycle for SC/MP Number 1 is active. With SC/MP Number 1 controlling the bus, the enable out ($ENOUT_1$) signal is low and other processors in the string are locked out. If both SC/MP Number 2 and SC/MP Number "N" initiate a bus request ($BREQ_2$ and $BREQ_N$ set high) while SC/MP Number 1 is controlling the bus, the following operations occur. $ENOUT_1$ is low until SC/MP Number 1 is finished; then, it goes high. At this time, $ENOUT_1$ and $ENIN_2$ go high; thus, SC/MP Number 2 takes control of the bus. In short, if all processors issue a bus request simultaneously, the string is served on a priority-select basis – SC/MP Number 1 first, SC/MP Number 2 second, SC/MP Number 3 third – and so on. Conversely, if SC/MP Number "N" issues a bus request and there are no others awaiting service, $ENIN_N$ is high and the request is granted. When microprocessors are cascaded as shown in figure 2-4b, stray capacitance can be a design consideration. With the processors in close physical proximity and with care in both hardware selection and interconnect design, the capacitance can generally be reduced to an acceptable level.

Any one of the SC/MP microprocessors shown in figure 2-4b can be replaced by a DMA peripheral; the BREQ, ENIN, and ENOUT control signals are used in exactly the same way. Typically, no more than three processors are used in a cascade arrangement without an external priority-logic controller.

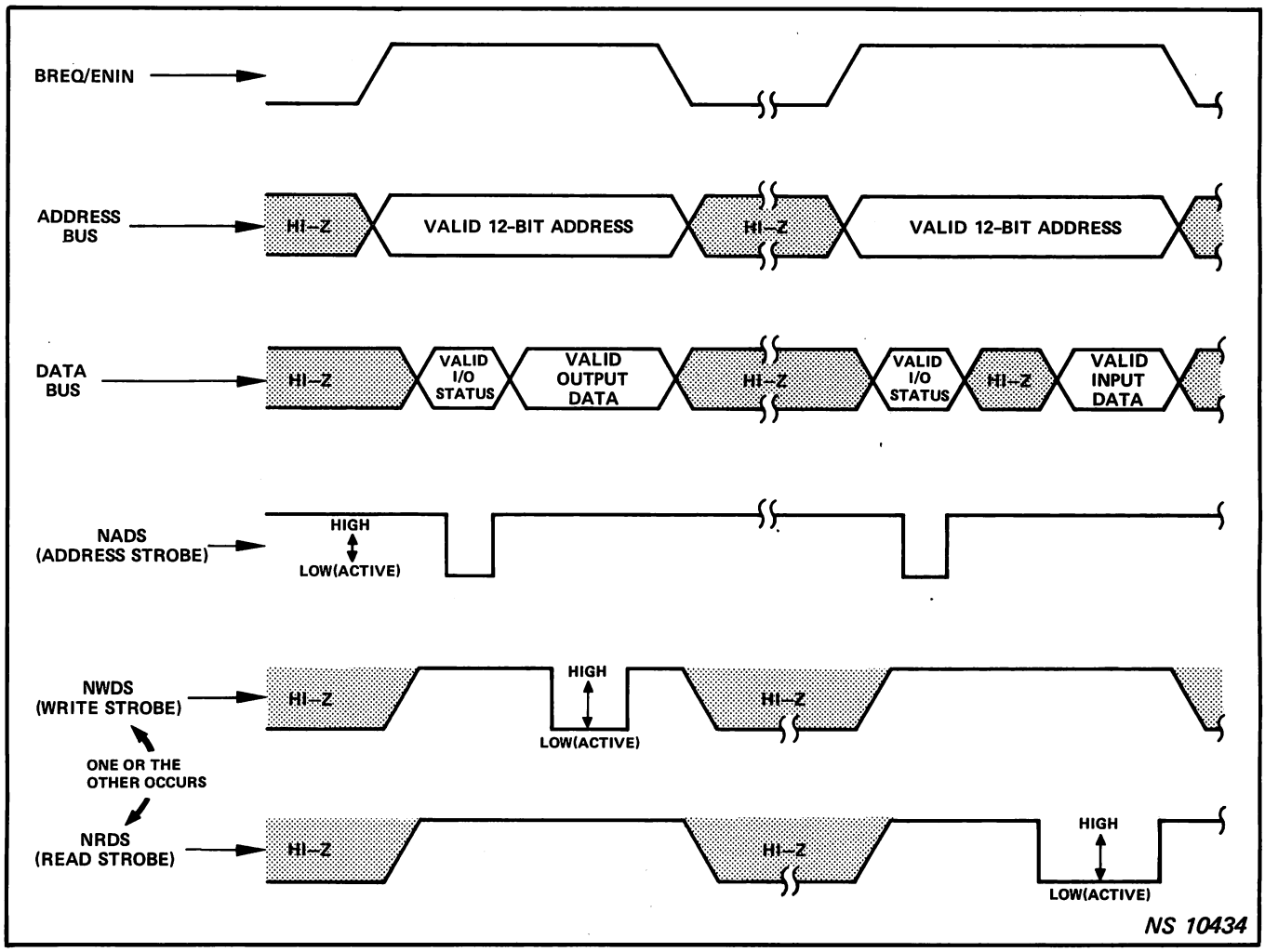


Figure 2-5. Typical Input/Output Sequence Showing Relative Timing

2.3.2 Input/Output Cycle

Once SC/MP has control of the address and data buses, the input/output cycle begins. Basically, the input/output cycle consists of inputting (reading data) from or outputting (writing data) into a specified memory location. Timing is shown in figure 2-5 for memory-access, read-data, and write-data operations. NADS strobes in the address for either a read or a write operation, and an associated NWDS or NRDS thereafter strobes in the write or read data, respectively. As shown, the processor either accepts (reads) input data from the data bus or outputs (writes) data onto the data bus. Since the timing is somewhat different for each function, both a read (NRDS) and write (NWDS) cycle is indicated. Although the timing relationships in figure 2-5 are not precise, they are adequate for a study of the purpose of signals shown and the sequence of operation.

When a bus request is granted, BREQ and ENIN are high; at this time, AD00 through AD11 of figure 2-1 are recognized as a valid address and any one of 4,096 discrete memory locations can be selected. The 12-bit address is latched on the SC/MP chip. That is, it remains on the bus for the duration of the input/output cycle; hence, the address strobe (NADS) may not be needed if only the 12-bit latched address is used.

An expanded view of the data bus at address-strobe time is shown in figure 2-6. When the address strobe (NADS) is low, input/output status information on the data bus is guaranteed to be valid. As shown by the accompanying legend, the 4 high-order address bits (AD12 through AD15) specify one of 16 address "pages," whereas the other 4 bits are flags available for hardware control. The 4-bit page address combined with the 12-bit latched address provides a 65,536-byte (16 by 4,096) address capability; however, to use the page address bits, peripheral latches must be provided.

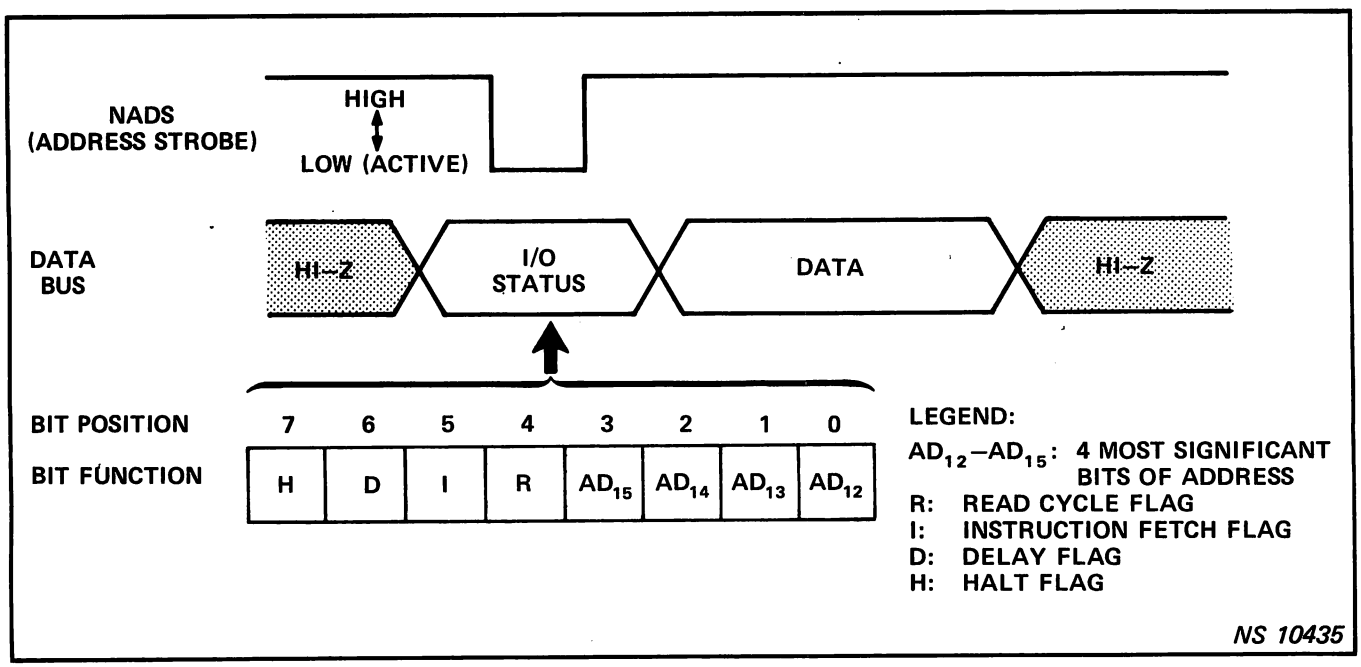


Figure 2-6. Data Bus at Address Strobe Time

For functional control, the status flags are generally latched. The latching arrangement can be anything from a simple flip-flop to an MSI device, such as a 4-bit or an 8-bit latch. Figure 2-7 shows a typical application where the "H" flag (status bit 7) and the CONTinue input are combined to generate a programmed halt. When a start switch S1 is momentarily closed to the NO contacts, the debounce circuit generates a positive-going clock pulse that sets the CONT input high via Q. As long as DB7 is not high (H-flag not set), the CLR input is high and the processor runs. When the H-flag is set high, the CLR input goes low at address strobe (NADS) time; accordingly, the CONT input is driven low by Q and the processor is halted.

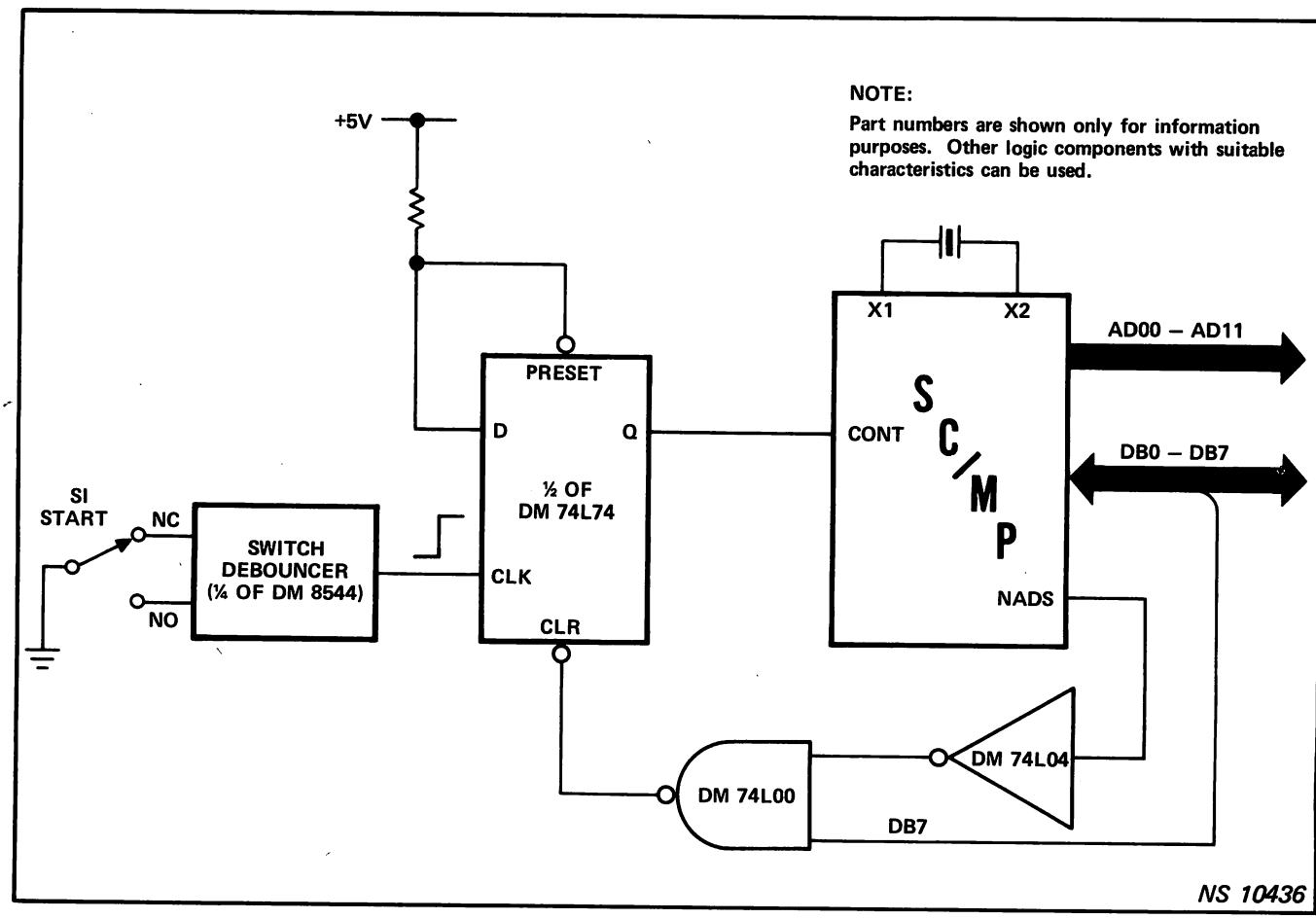


Figure 2-7. Using "H" Flag To Generate a Programmed Halt

Besides the H-flag/halt function shown in figure 2-7, the SC/MP chip is readily adaptable to other control circuits. One of these is shown in figure 2-8; here, single-cycle/single-instruction operation is implemented by two flip-flops, two switches, and some simple logic. Switch S1 is set to the desired operating mode and switch S2 is momentarily closed to the NO contact. These events cause the CONT input to go high and the NHOLD input to go low; thus, an instruction is fetched and executed if S1 is set to SINGLE INSTRUCTION, or a fetch-and-wait operation occurs if S1 is set to SINGLE CYCLE. At address-strobe (NADS) time, the flip-flops are cleared for the beginning of a new operation.

At the conclusion of the address strobe, the processor is ready to begin a data-input (read) cycle or a data-output (write) operation. As shown in figure 2-5, the read and write functions are synchronized by the read (NRDS) and write (NWDS) strobes. When the read strobe is low, data are gated from the data bus into the processor; when the write strobe is low, data transferred from the processor to the data bus are guaranteed to be valid. For a given input/output cycle, either the read or the write strobe is active (not both).

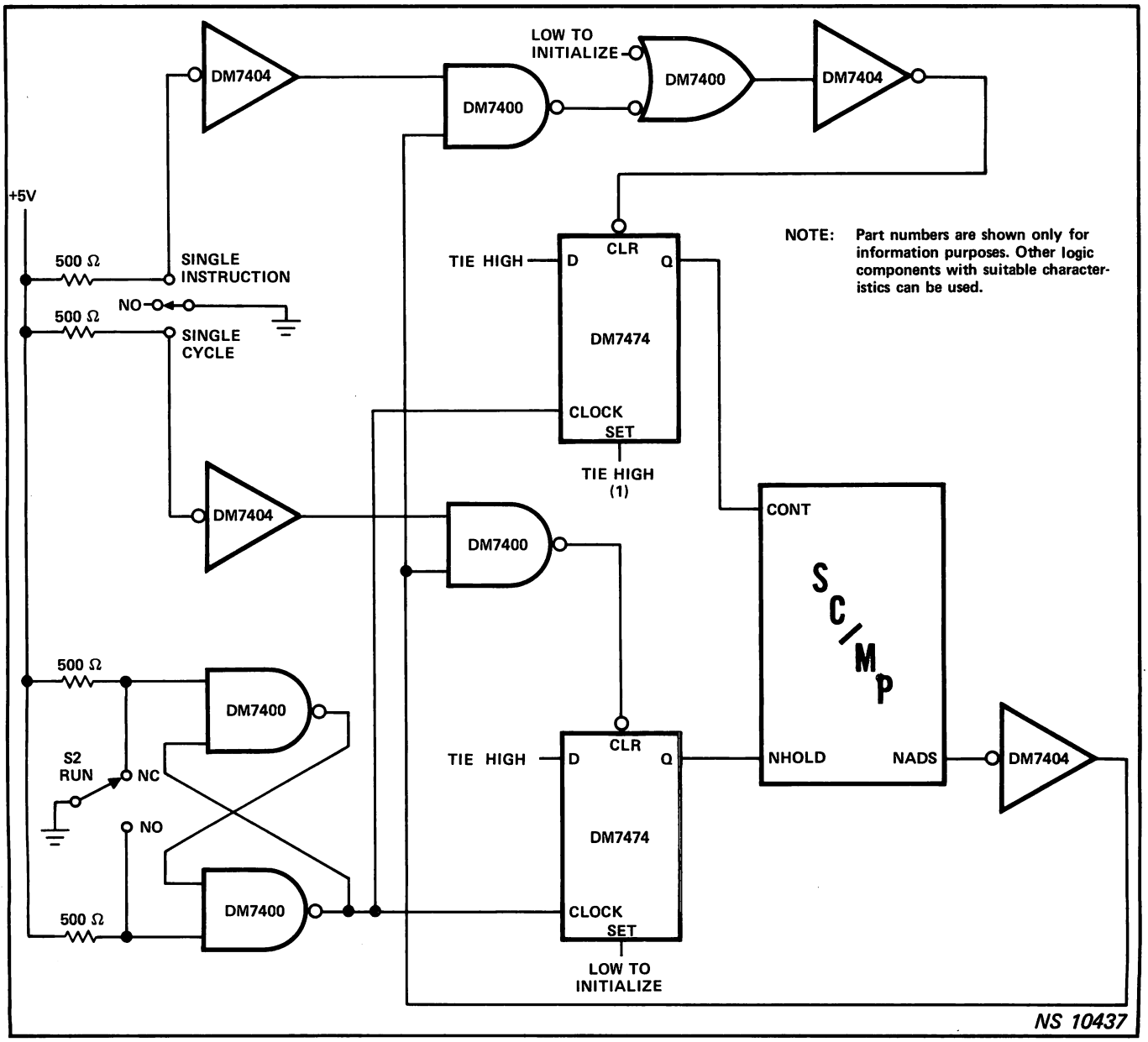
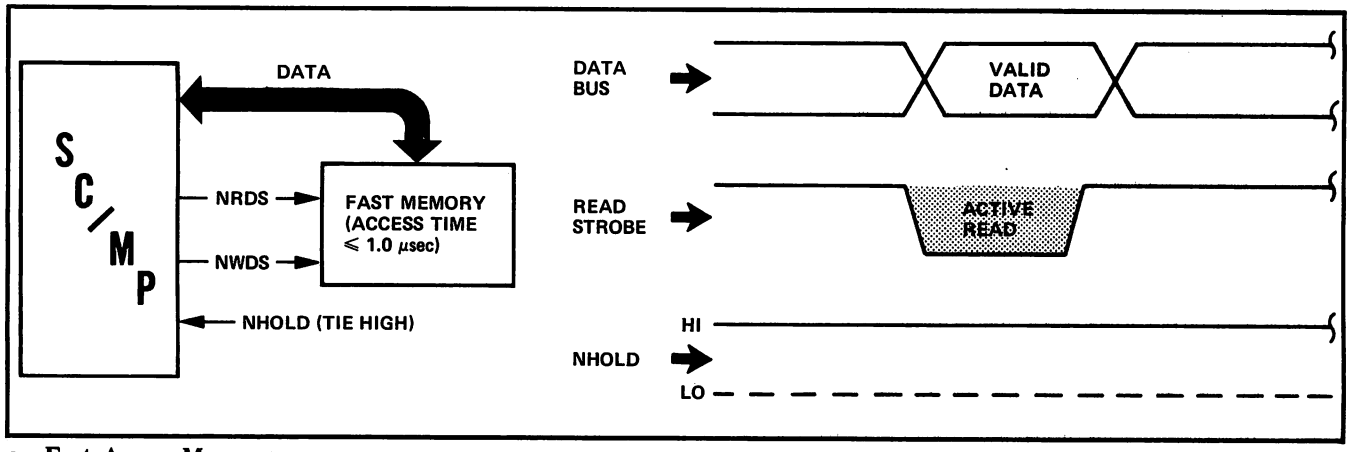


Figure 2-8. Circuit Detail To Implement Single-Cycle/Single-Instruction Control

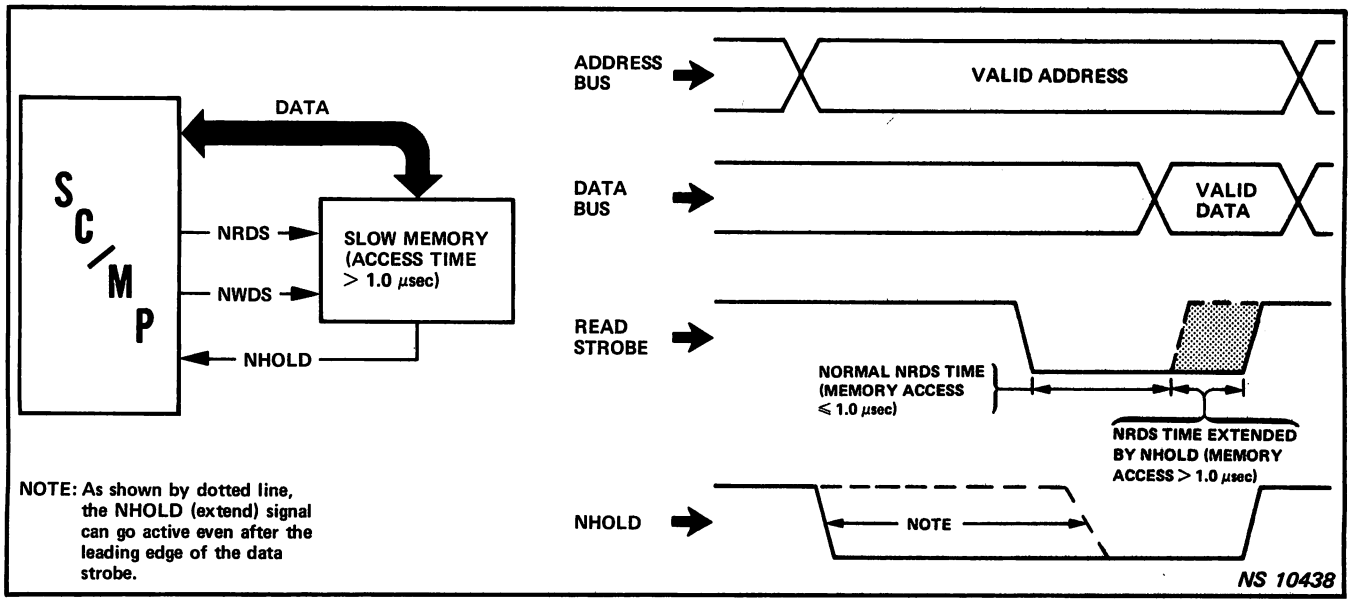
Data transfers to and from SC/MP need not be synchronized, or slaved, to a particular timing sequence; devices with widely different data rates and all using common system buses can be serviced by the processor. As shown in figure 2-9a, a data transfer between SC/MP and memory peripherals (with access times of 1.0 microsecond or less in systems that use a 1MHz crystal) is a simple and straightforward process. The input/output sequence proceeds as follows:

- Bus request from SC/MP
- Request granted by internal or external logic control
- Address valid (and latched if memory exceeds 4K)
- Data valid and either inputted or outputted, as appropriate

Valid data appear on the data bus before the read strobe makes a low-to-high transition. When using a fast memory, valid data are always present before the read strobe goes inactive; thus, there is no need to extend the read-strobe interval. Compare this with the slow-memory read cycle shown in figure 2-9b. The input/output sequence is identical to that just described; however, now the memory-access time is appreciably longer, and the read strobe may terminate before the slow-memory device can put "valid" data on the bus. To prevent such an occurrence, an NHOLD signal is generated by the slow-memory peripheral and is applied to the control logic of SC/MP. As illustrated, the read (or write) strobe is extended by NHOLD to satisfy the particular memory-access requirement, without any sacrifice in processing speed.



a. Fast-Access Memory



b. Slow-Access Memory

Figure 2-9. Extending Input/Output Cycle for Slow-Memory Devices

2.3.3 Buffering SC/MP Buses

In small systems with minimum memory requirements, buffering of the address and data buses may not be required. Such a simple system with hookup details is shown in figure 2-10a. As peripheral and memory requirements increase, buffering is required. Typically, the buses can be buffered as shown in figure 2-10b.

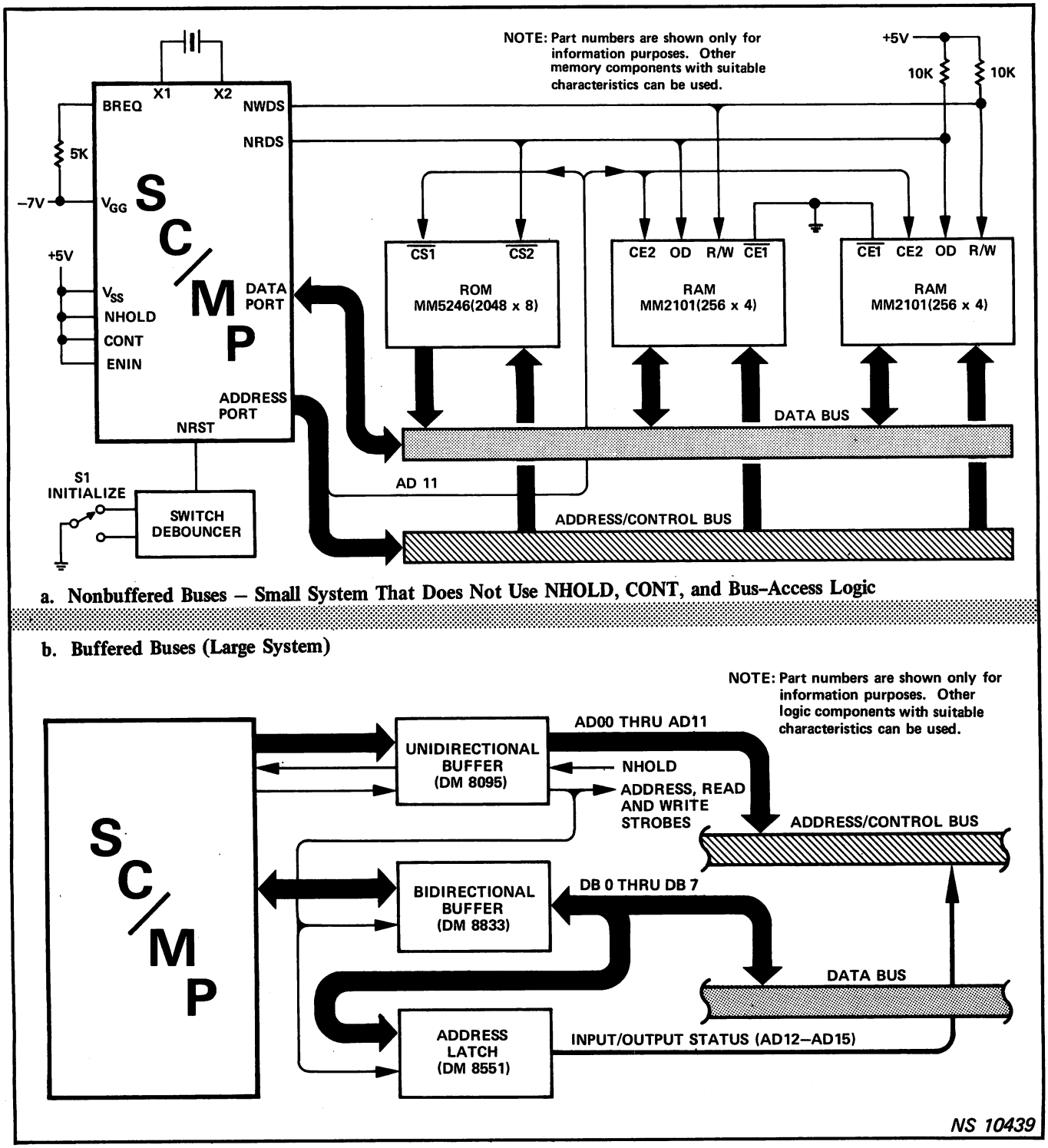


Figure 2-10. Nonbuffered and Buffered Bus Interfaces

2.3.4 Serial Input/Output Data Transfers

For high-data rate peripherals, information is moved to and from SC/MP in parallel; however, input or output of data in serial form is an efficient means of transferring data for slow data-rate peripherals – such as X-Y plotters, teletype-writers, slow-speed printers, and so on. An example of how the serial input/output capability can be used is shown in figure 2-11. Serial data are moved directly into and out of the extension register using the SIN and SOUT ports of SC/MP. In this case, flags 0, 1, and 2 are used to control the shift registers; in other applications, the flags and the two sense inputs might be used to input and output the serial data. In figure 2-11, the SIN and SOUT lines could be expanded to serve eight or more input/output devices by the use of appropriate multiplexer components.

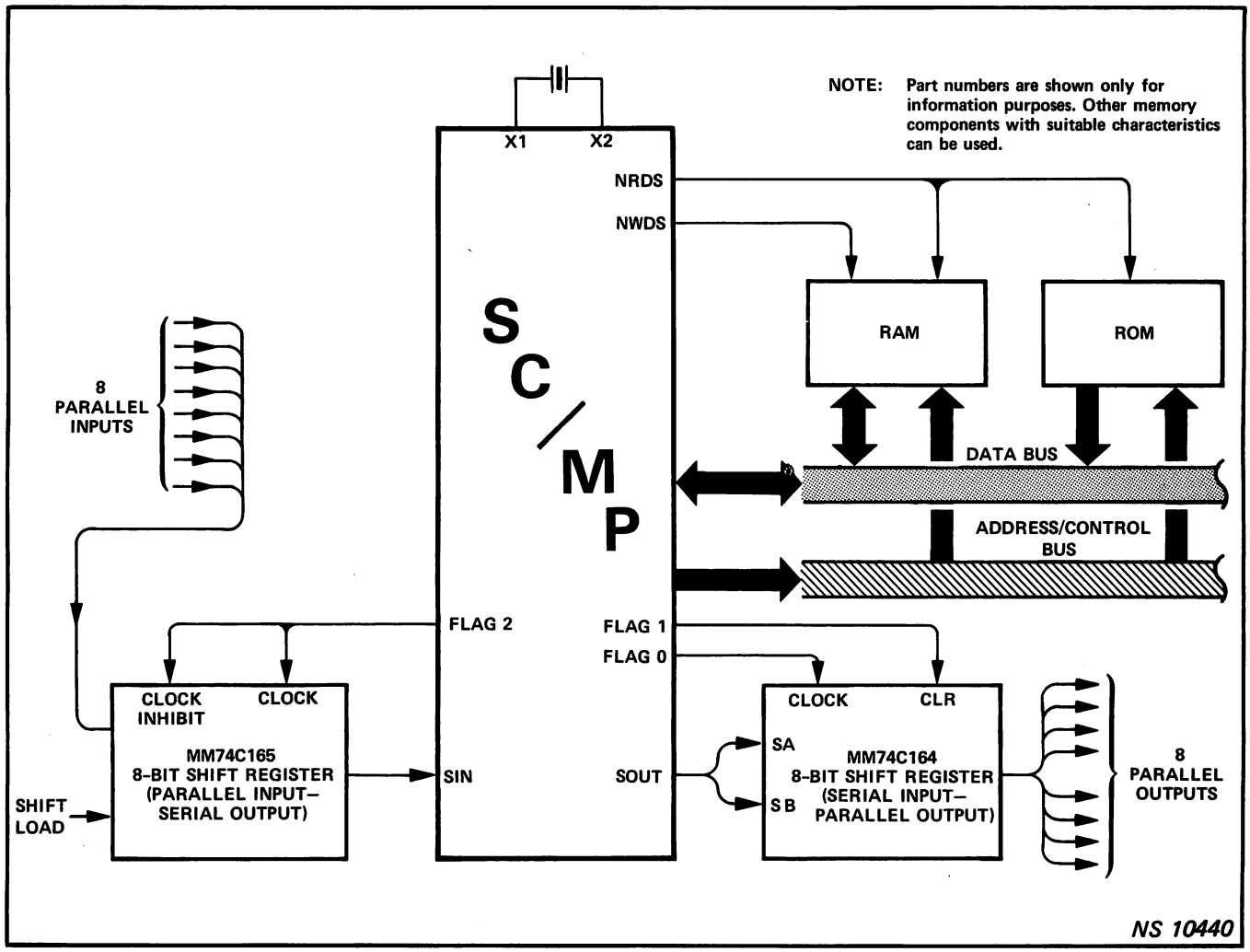


Figure 2-11. Using SC/MP With a Simple Serial Interface

2.3.5 Flags and Sense

Bits 0, 1, 2,4, and 5 of the status register provide three flags and two sense inputs. Bit assignments are as follows:

Status Register Bit	*Description*
0	User Flag 0 (Output)
1	User Flag 1 (Output)
2	User Flag 2 (Output)
4	Sense A (Input)
5	Sense B (Input)

Each of the foregoing functions is available at a pin of the SC/MP chip; thus, it may be monitored easily during program development or a debug procedure. Both the flag outputs and the sense inputs can be used for hardware control; likewise, they can be used for software status or for a combination of hardware/software control.

2.3.6 SC/MP Interrupt

The interrupt system of SC/MP is under software control and is supervised as indicated in figure 2-12. Before an instruction is fetched, bit 3 of the status register is tested. If the bit is not set (interrupt enable flag low) and the CONTinue input is high, the program counter is incremented, and the next instruction is fetched and executed. If bit 3 is set and the enable-interrupt (Sense A) line is high, the interrupt is serviced: bit 3 (enable flag) is reset and the contents of the program counter are exchanged with the contents of pointer register 3 – the pointer contains the address of the subroutine that services the interrupt.

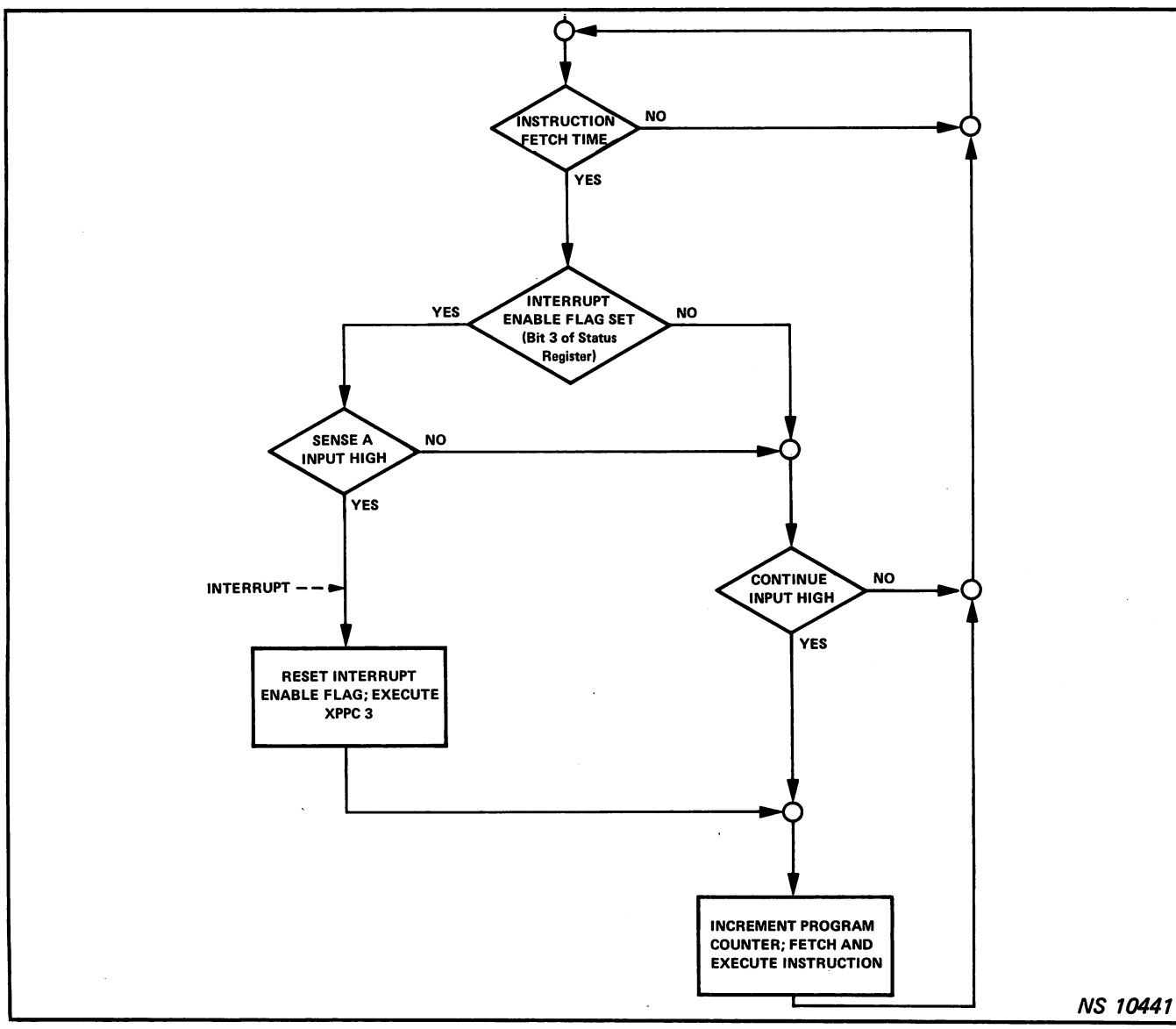


Figure 2-12. SC/MP Interrupt/Instruction-Fetch Process

2.4 INTERNAL CONTROL AND DATA MOVEMENT

2.4.1 General Considerations

Except for input and output serial transfers of data, all data enter and exit the SC/MP chip via an 8-bit bidirectional input/output data bus – pins 9 through 16 in figure 2-1.

Two operator-controlled functions start operation of the CPU. First, the CPU is initialized by pulsing the NRST (reset) signal low for an interval at least twice the time period of the crystal (1/fXtal); NRST must be returned high for the processor to start operation. (Once initialized, NRST must remain high for normal SC/MP operation.) Second, the CONT (continue) signal is set high to start the processor. Operation then proceeds as follows.

When the processor is started, the program counter is incremented and the first byte (an instruction) is fetched from the address specified by the contents of the program counter – memory location 0001_{16}. The first instruction byte enters the CPU via the input/output data bus and then enters the instruction register. The instruction is decoded and then is implemented under control of the instruction decode and control unit. A single-byte instruction specifies an operation that SC/MP can execute without further reference to memory. A single-byte instruction has a '0' in bit position '7' (most significant bit), whereas a double-byte instruction has a '1' in bit position '7' of the first byte. A double-byte instruction in addition to the operational information contained in the first byte also contains a second byte that is either an 8-bit data or an 8-bit displacement field. When the second byte represents data, the data are processed by SC/MP during execution of the instruction. When the second byte represents a displacement value, it is used to calculate an address that will be accessed (written into or read from) during execution of the instruction (this is covered later in this chapter under SC/MP Addressing (2.4.4)).

It is also possible to initialize the CPU with the CONT signal high; in this case, all registers are cleared and program execution resumes at location 0001_{16} when NRST goes high.

Other than the control, the decoding, and the arithmetic logic that effect implementation of the CPU operations, all on-chip data manipulation uses one or more of the seven programmer-accessible registers – shown in figure 2-13. Three of these registers are 8 bits wide: the accumulator, the status register, and the extension register. The other four registers are 16-bit pointer registers 0, 1, 2, and 3; pointer register 0 is dedicated as the program counter. All 16-bit registers are linked internally by two read buses and two write buses – the low-order set of buses dedicated to bits 0 through 7 and the high-order set dedicated to bits 8 through 15. The 8-bit registers connect only to one read bus and one write bus. The capability to exchange high-order and low-order bytes is discussed in the descriptions of appropriate instructions (appendix A).

In contrast to the seven above-described registers, there are three other registers. Two registers, the instruction register and the input/output data register, are 8 bits wide. The instruction register (mentioned above) holds an instruction byte en route to the instruction decode and control logic. The input/output data register is the link between the input/output data bus and the seven programmer-accessible registers. The 16-bit output address register is the link between the programmer-accessible registers and the address buses.

2.4.2 Summary of SC/MP Registers

The formats of the seven SC/MP operator- and program-controllable registers are shown in figure 2-13. The purposes and functions of these registers are described in 2.4.2.1 through 2.4.2.5.

2.4.2.1 Accumulator (AC)

The 8-bit accumulator (AC) is the primary working register of SC/MP. The accumulator is used in performing arithmetic and logic operations and for storing the results of these operations. Data transfers, shifts, and rotates also use the accumulator. In all, 37 of the 46 SC/MP instructions use the accumulator.

2.4.2.2 Status Register (SR)

7	6	5	4	3	2	1	0	Bit Positions
CY/L	OV	SB	SA	IE	F3	F2	F1	Flags

The status register (SR) provides storage for arithmetic, control, and software status flags. The function of each bit in the register is shown below.

Bit	Description
0	*User Flag 0 (F0).* User assigned for control function or for software status. The output of this bit is available at a pin of the SC/MP chip.
1	*User Flag 1 (F1).* Same as F0.
2	*User Flag 2 (F2).* Same as F0.
3	*Interrupt Enable Flag (IE).* The processor recognizes the interrupt input if this flag is set.

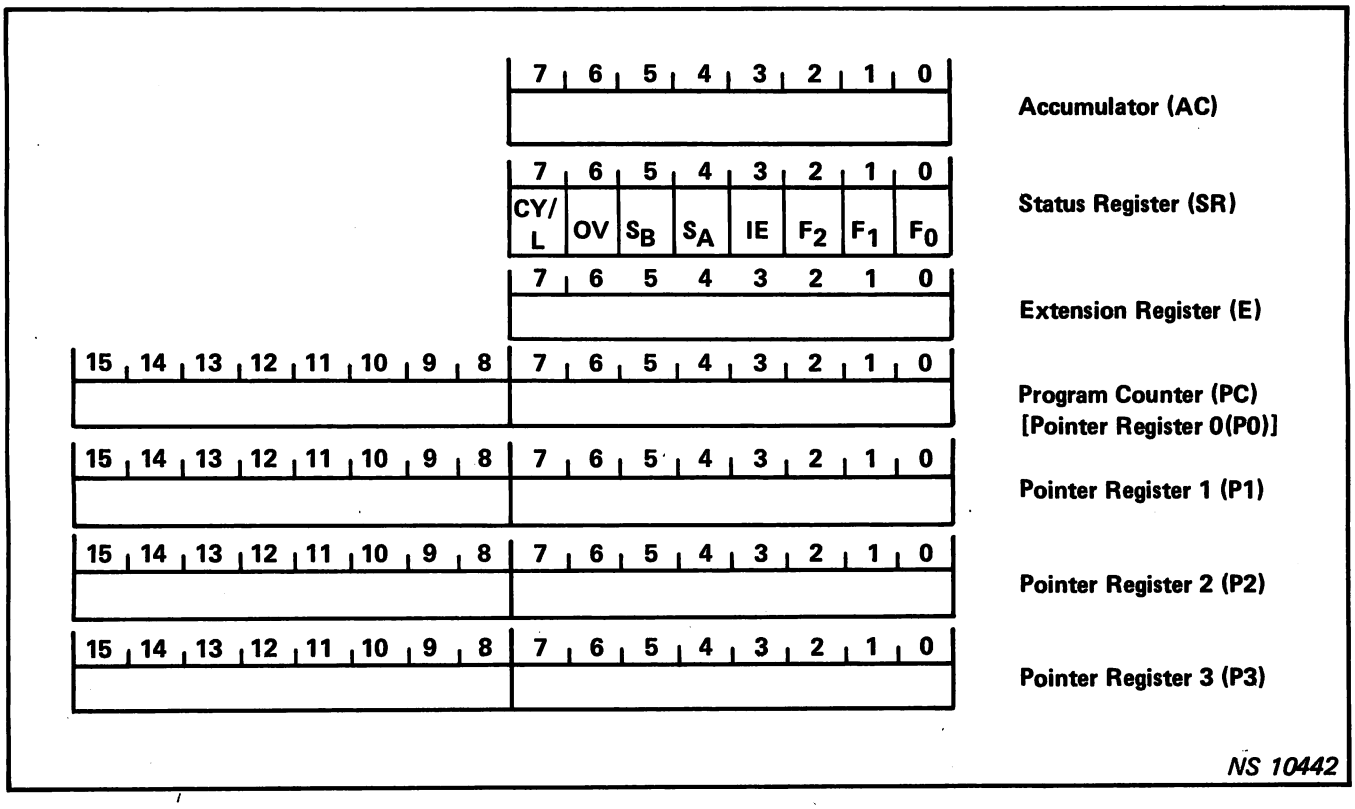


Figure 2-13. Operator-Controlled Registers

Bit Description

4 *Sense Bit A (SA).* This bit is tied to a package pin and may be used to sense external conditions. This bit is "read-only"; thus, the Copy Accumulator to Status Register (CAS) Instruction does not affect this bit. When Interrupt Enable is set, Sense Bit A serves as the interrupt input.

5 *Sense Bit B (SB).* Same as SA, except it is not used as an interrupt input.

6 *Overflow (OV).* This bit is set if an arithmetic overflow occurs during an add (ADD, ADI, or ADE) or a complement-and-add instruction (CAD, CAI, or CAE). Overflow is not affected by the decimal-add instructions (DAD, DAI, or DAE).

7 *Carry/Link (CY/L).* This bit is set if a carry from the most significant bit occurs during an add, a complement-and-add, or a decimal-add instruction. The bit is also included in the Shift Right with Link (SRL) and the Rotate Right with Link (RRL) Instructions. CY/L is input as a carry into the bit 0 position of the add, complement-and-add, and decimal-add instructions.

2.4.2.3 Extension Register (E)

The 8-bit extension register (E) is used primarily with the accumulator to perform arithmetic, logic, and data-transfer operations. If the displacement in an indexed or an auto-indexed memory-reference instruction equals -128_{10}, then the contents of E are substituted for the displacement for the given instruction. Another function of the extension register is serial input/output; the serial input/output function is a simultaneous operation (that is, as the high-order bit is inputted, the low-order bit is outputted).

2.4.2.4 Program Counter (PC)

The program counter (PC) is the dedicated 16-bit pointer register P0. The program counter contains the address of the instruction being executed; it is incremented just before an instruction fetch. Arithmetic affecting the program counter is performed on the low-order 12 bits; the high-order 4 bits are not affected. Thus, the contents of the program counter can "wrap around" in a 4,096-byte page.

2.4.2.5 Pointer Registers (PTR)

There are three 16-bit pointer registers (PTR) available for memory and peripheral device addressing, and for use as page pointers, stack pointers, or index registers. As mentioned previously, P0 is assigned the function of program counter by the design of the hardware.

2.4.3 Inter-Register Data Flow

Data flow relationships between memory and the seven registers of SC/MP are shown and described in figure 2-14.

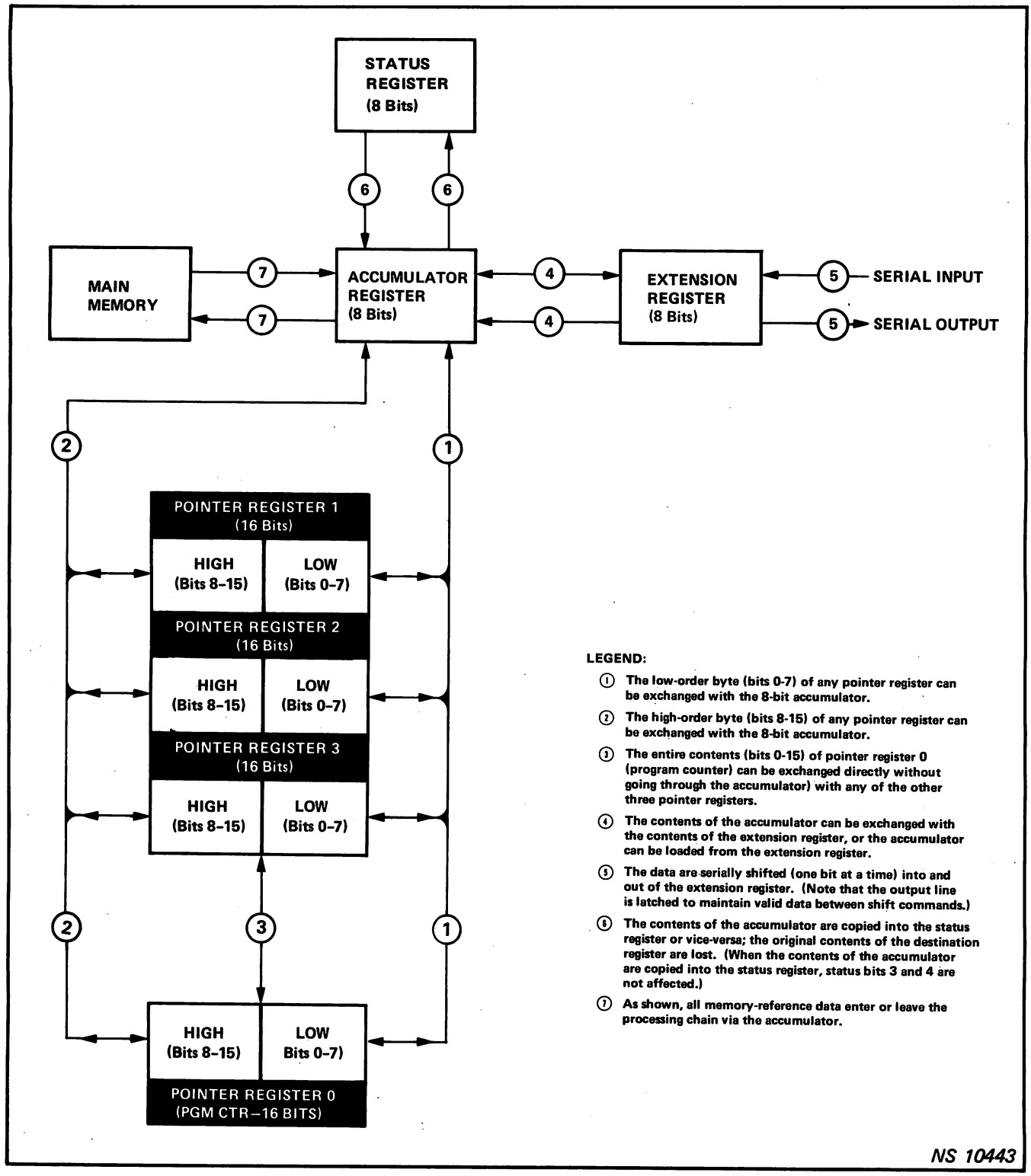


Figure 2-14. Interrelationships of SC/MP Registers

2.4.4 SC/MP Addressing

2.4.4.1 General Capabilities

One of the foregoing 16-bit pointer registers is used in each memory-reference instruction. The program counter (pointer register 0) is always used to specify the addresses of program instructions (or data), whereas pointers 1, 2, and 3 are used for other address/data requirements. Unless otherwise specified, the SC/MP assembler always uses the program counter for memory access; if no other pointer is specified and access cannot be gained via the program counter, an addressing error is indicated.

The basics of addressing can best be understood by a study of the register formats shown in figure 2-15. As shown in *a*, each pointer register can identify absolutely any of 65,536 memory locations – 0000_{16} to $FFFF_{16}$. To conserve pinouts on the chip, the absolute-addressing capability of the pointer register is subdivided – see view *b*. The 4 most significant bits are dedicated to "page" selection (1 of 16) and can only be changed via a program-load command. The 12 low-order bits can be altered arithmetically to address any one of 4,096 locations within the selected page. As shown in *c*, the 8-bit displacement field in the instruction provides an offset within the selected memory page. The displacement is a signed 8-bit value that when algebraically combined with the content of a designated pointer register yields an address variance of 256 words (-128 to +127) within the selected page.

To provide maximum programming flexibility, wraparound addressing is used. That is, when the last address on the selected page is reached, a new page is not automatically selected; instead, the carry into page-select bits are ignored. Refer to following description of addressing architecture.

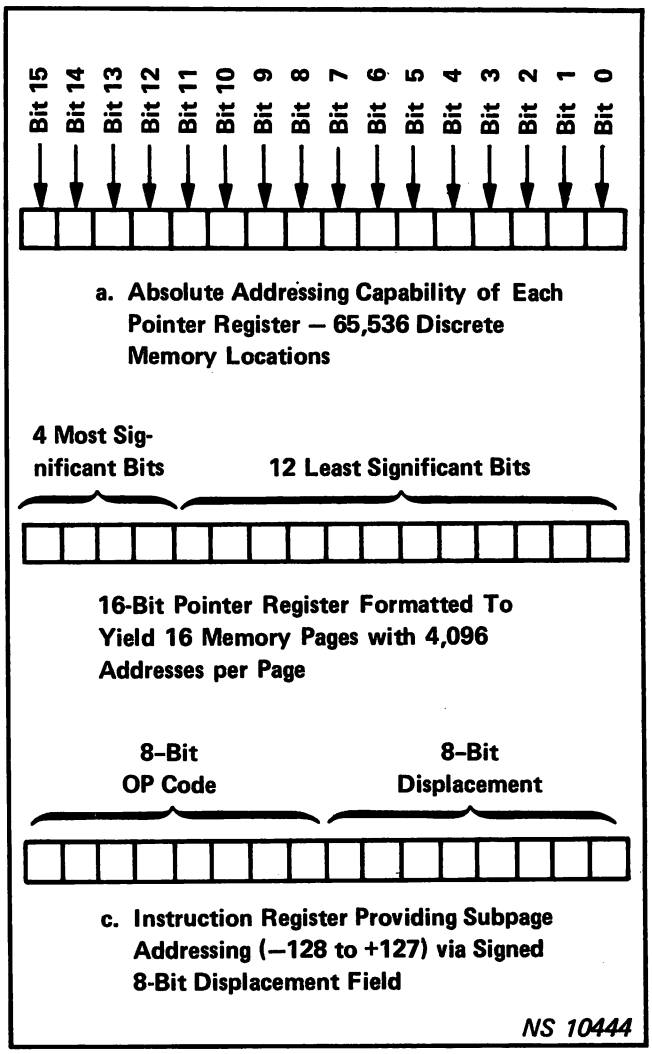


Figure 2-15. Addressing Capabilities of SC/MP

2.4.4.2 Addressing Architecture

Memory is organized as a sequence of 8-bit bytes. Each byte is identified by a 16-bit address that represents its sequential position in memory from 0 to X'FFFF ($65,535_{10}$). As shown in figure 2-16, memory is divided into 16 pages of 4,096 bytes each. Each address consists of a 4-bit page address and a 12-bit page displacement.

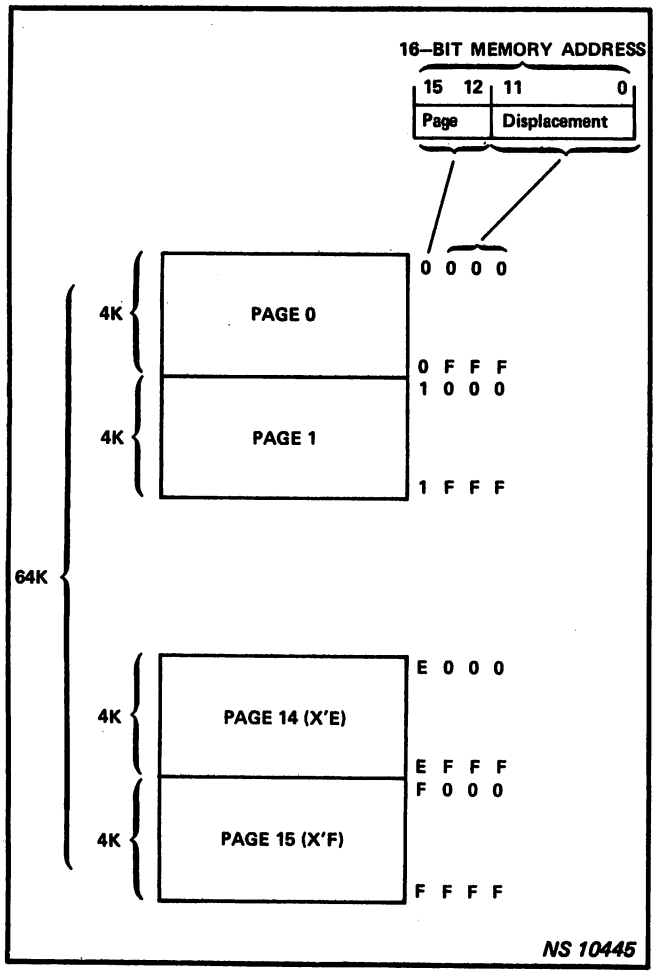


Figure 2-16. Memory Organization of SC/MP

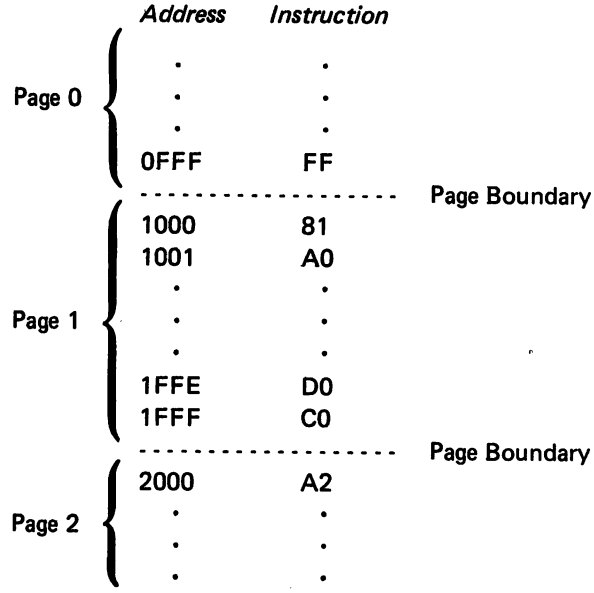
When performing arithmetic to calculate the effective address of an operand, the calculations are performed on the low-order (displacement) portion of the address with no carry into the high-order (page) portion. See the following table for examples.

When the address displacement remains within the current page, no carry is generated because the sum of the displacements did not produce a carry. In the example where the displacement exceeds the page size, a carry is normally generated when the two numbers are summed, but it is not carried into the page address field.

	Address Displacement Remains Within Page		Address Displacement Exceeds Page Size	
	Address of Page	Displacement Within Page	Address of Page	Displacement Within Page
Current Address	0	FB4	0	FB4
Displacement From Instruction		05		4D
New Address	0	FB9	0	001

When incrementing the address to fetch the next instruction, the same page/displacement arithmetic occurs.

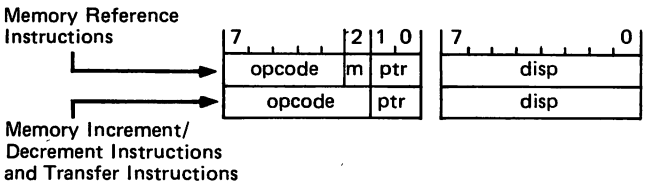
If a 2-byte instruction is inadvertently separated by a page boundary, an error occurs. Consider the following sequence of instructions on pages 0, 1 and 2 – with the first digit of the address designating the page and the next three digits, the location within the page.



The instruction intended, when the PC = 1FFF (last word in page 1), is X'C0A2 (LD 20A2). However, instead of fetching the latter half of the instruction from page 2, a wrap-around is made to the first word of page 1; the instruction that will be executed is X'C081 (LD 1081). The user must organize his programs to provide protection from the situation described above.

2.4.4.3 Addressing Formats

During execution, instructions and data defined in a program are stored into and loaded from specific memory locations, the accumulator, or selected registers. Because the CPU, memory (read/write and read-only), and peripherals are on a common data bus, any instruction used to address memory may also be used to address the peripherals. The formats of the instruction groups that reference memory are shown below.



Memory-reference instructions use the PC-relative, indexed, or auto-indexed methods of addressing memory. The memory increment/decrement instructions and the transfer instructions use the PC-relative or indexed methods of addressing. Immediate addressing is the addressing mode specific to the immediate instruction group.

The various methods of addressing memory and peripherals are shown in table 2-2.

Table 2-2. Addressing Modes

Type of Addressing	Operand Formats		
	m	ptr	disp
PC-relative	0	0	-128 to +127
Indexed	0	1, 2, or 3	-128 to +127
Immediate	1	0	-128 to +127
Auto-indexed	1	1, 2, or 3	-128 to +127

For PC-relative, indexed, and auto-indexed memory-reference instructions, another feature of the addressing architecture is that the contents of the extension register are substituted for the displacement if the instruction displacement equals -128_{10}.

2.4.4.3.1 PC-Relative Addressing

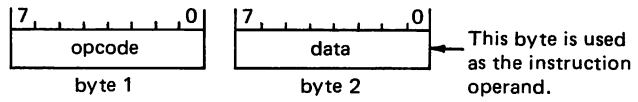
A PC-relative address is formed by adding the displacement value specified in the operand field of the instruction to the current contents of the program counter. The displacement is an 8-bit twos-complement number, so the range of the PC-relative addressing format is -128_{10} to $+127_{10}$ bytes from the current location of the program counter. During execution of an instruction, the program counter contains the address of the last byte of the instruction. The following examples show the use of PC-relative addressing.

```
Location Generated
Counter  Code
0005     C00E    LOOP: LD    TEMP  ;LOAD THE VALUE
                                   ; IN TEMPORARY
                                   ;STORAGE
                       .
                       .
                       .
000E     90F5          JMP   LOOP  ;REPEAT
                       .
                       .
                       .
0014     04      TEMP: .BYTE X'04
```

The assembler assumes PC-relative addressing in the memory-reference and transfer instructions when no pointer-register operand is specified.

2.4.4.3.2 Immediate Addressing

Immediate addressing uses the value in the second byte of a double-byte instruction as the operand for the operation to be performed (see below).



For example, compare a Load Instruction (LD) to a Load Immediate Instruction (LDI). The Load Instruction uses the contents of the second byte of the instruction in computing the effective address of the data to be loaded. The Load Immediate Instruction uses the contents of the second byte as the data to be loaded. Because the operand occurs as the second byte of a 2-byte instruction, page boundary conditions should be observed as mentioned in 2.4.4.2.

2.4.4.3.3 Indexed Addressing

Indexed addressing enables the programmer to address any location in memory through the use of a pointer register and the displacement. When indexed addressing is specified in an instruction, the displacement is added to the contents of the designated pointer register to form the effective address. The contents of the pointer register are not modified by indexed addressing. Indexed addressing is used to access tables or subroutines, to transfer control to another page, or to transfer control to a section of the current page that is outside the range of the PC-relative transfer. The rules for page boundaries still apply, so the user is cautioned about crossing page boundaries when using indexed

addressing to access tables. Such a reference results in a wrap-around from the end to the beginning of the page, or vice-versa (see 2.4.4.2).

2.4.4.3.4 Auto-Indexed Addressing

Auto-indexed addressing provides the same capabilities as indexed addressing along with the ability to increment or decrement the designated pointer register by the value of the displacement. If the displacement is less than zero, the pointer register is decremented by the displacement before the contents of the effective address are fetched or stored.

If the displacement is equal to or greater than zero, the pointer register is used as the effective address, and the pointer register is incremented by the displacement after the contents of the effective address are fetched or stored.

NOTE

The contents of the pointer register are modified by auto-indexed addressing.

An "at sign" (@) before the displacement operand designates an auto-indexed operation. Example:

```
Generated
Code

C601      LD   @1(P2)     ;GET A BYTE FROM THE
                          ;TABLE, AUTO-INDEX
```

Chapter 3

SC/MP APPLICATION MODULES

3.1 INTRODUCTION

The following paragraphs provide information regarding the following three SC/MP application modules.

- SC/MP CPU Application Module (order number ISP-8C/100)
- SC/MP RAM Application Module (order number ISP-8C/002)
- SC/MP PROM/ROM Application Module (order number ISP-8C/004P)

The application modules are intended for end application use or prototyping. The three modules can be interconnected to form a system that can be used to develop software or firmware for the intended end application. Together, the modules can be used to form the basis of a microcomputer system; thus, engineering design time is reduced as compared to a system using integrated circuits as the basic component.

Physically, each module is implemented on a printed-circuit card which is 4.375 inches by 4.862 inches (see figure 3-1). The small card size permits use in physically confined situations such as portable equipment applications. In addition, placing the microprocessor on one card and the two types of memory on two individual cards allows a modular configuration to be used in the system design.

Each circuit card is equipped with a 72-pin edge connector. Mating connectors, a line of compatible card cages, extender cards, and wire-wrap breadboard cards are available from a variety of sources. Table 3-1 lists the sources of accessory equipment compatible with SC/MP application modules.

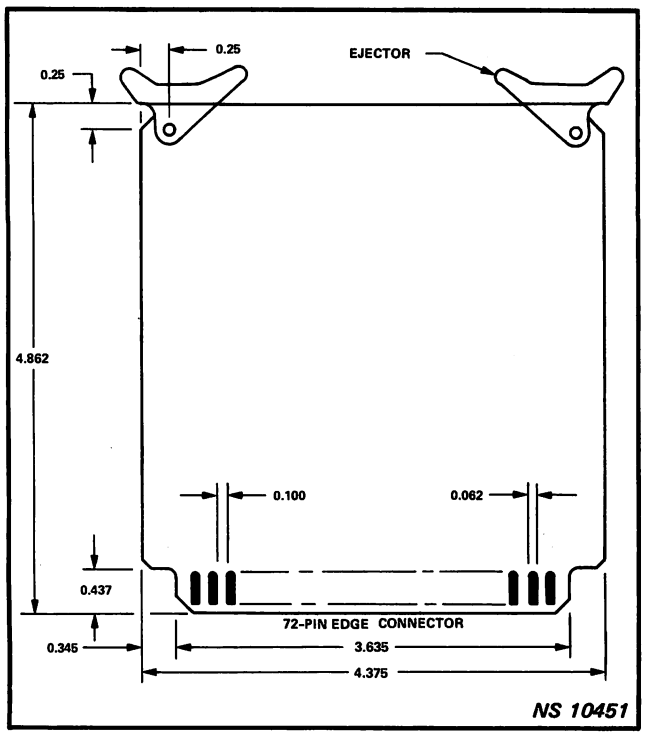


Figure 3-1. SC/MP Application Modules Dimensional Details

Table 3–1. Sources of Accessory Equipment* for SC/MP Application Modules

Equipment	Source	Part Number
72-contact Edge Connector	Augat Robinson-Nugent Stanford Applied Eng. National Connector Cinch Winchester Elco Viking	14005-17P3 EC-721 CDP7000-72 900100-36 50-72C-30 HW36C0111 00-6307-072-309-001 3VH36/1JND5
13-connector Card Cage with Backplane	Augat Robinson-Nugent Scanbe	8170-MG1 MECA-1
9-connector Card Cage with Backplane	Augat	8170-MG10
6-connector Card Cage with Backplane	Augat	8170-MG8
3-connector Card Cage with Backplane	Augat	8170-MG6
Extender Card	Augat Robinson-Nugent	8136-MG13 EB-72
Universal w/w Card with Terminals	Augat Robinson-Nugent	8136-UMG1 UNI-24
High-density w/w Card with Terminals	Augat Robinson-Nugent	8136-MG15
Universal w/w Card without Terminals	Robinson-Nugent	(Special)

*The accessory equipment listed in table 3–1 has not necessarily been evaluated by National Semiconductor.

3.2 CPU APPLICATION MODULE

A *functional* block of the CPU application module is shown in figure 3-2. The CPU module provides both read-only and read-write memories plus all required buffering and latching circuits. Standard versions of the module provide 256 bytes of RAM and 512 bytes of PROM (or ROM); in optional versions, the read-only memory can be expanded to 2,048 bytes. For applications where the foregoing memory capability is sufficient, the CPU module is a self-contained system, requiring only +5-volt and -12-volt power supplies. Power-on initialize and all necessary timing signals are generated by the module.

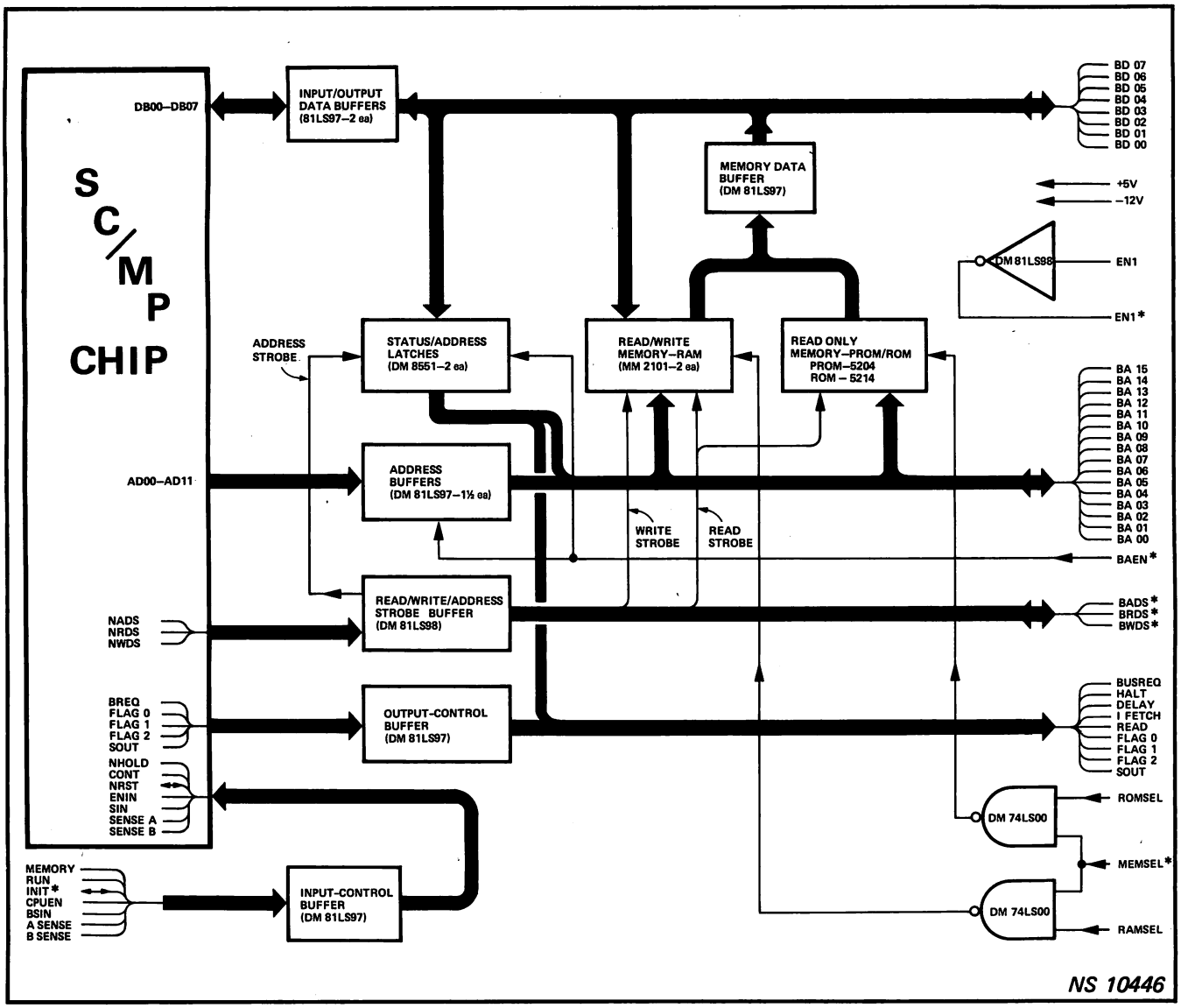


Figure 3-2. SC/MP CPU Application Module, Functional Block Diagram

As shown, the 4 most significant address bits (AD12-AD15) and the input/output status signals (H-Flag, D-Flag, I-Flag and R-Flag) are latched; other input/output lines use LS-series buffers. This particular series of buffer elements requires extremely small input currents; thus, loading is minimal and, if required, the CPU module can easily service a memory capacity of 65K bytes.

If system memory exceeds the on-card capability, three memory-control signals (MEMSEL, RAMSEL, and ROM-SEL) must be supplied by the user; these signals can be derived from the module-select/memory-select logic on the RAM or PROM/ROM applications modules. Unless the RAM module is selected, the MEMSEL* signal is high-impedance (open circuit); with the module selected, the signal is active low. The same set of conditions apply to the MEMSEL* line on the PROM/ROM module. All MEMSEL* lines are wire-ANDed, and, as shown in figure 3-2, the common MEMSEL* line forms one leg to the memory-select gates on the CPU module. When a memory module is selected, MEMSEL* is low and the memories of the CPU module are locked out. When no memory card is selected, the MEMSEL* line is high and the CPU RAM or ROM memory can be selected – RAM if RAMSEL is high and ROM if ROMSEL is high.

If the CPU module is used as a stand-alone system, the MEMSEL* signal can be wired high and address bit 10 can be used, via the ENI/ENI* inverter, to select PROM/ROM or RAM memory. A timing summary of the CPU card is shown in figure 3-3.

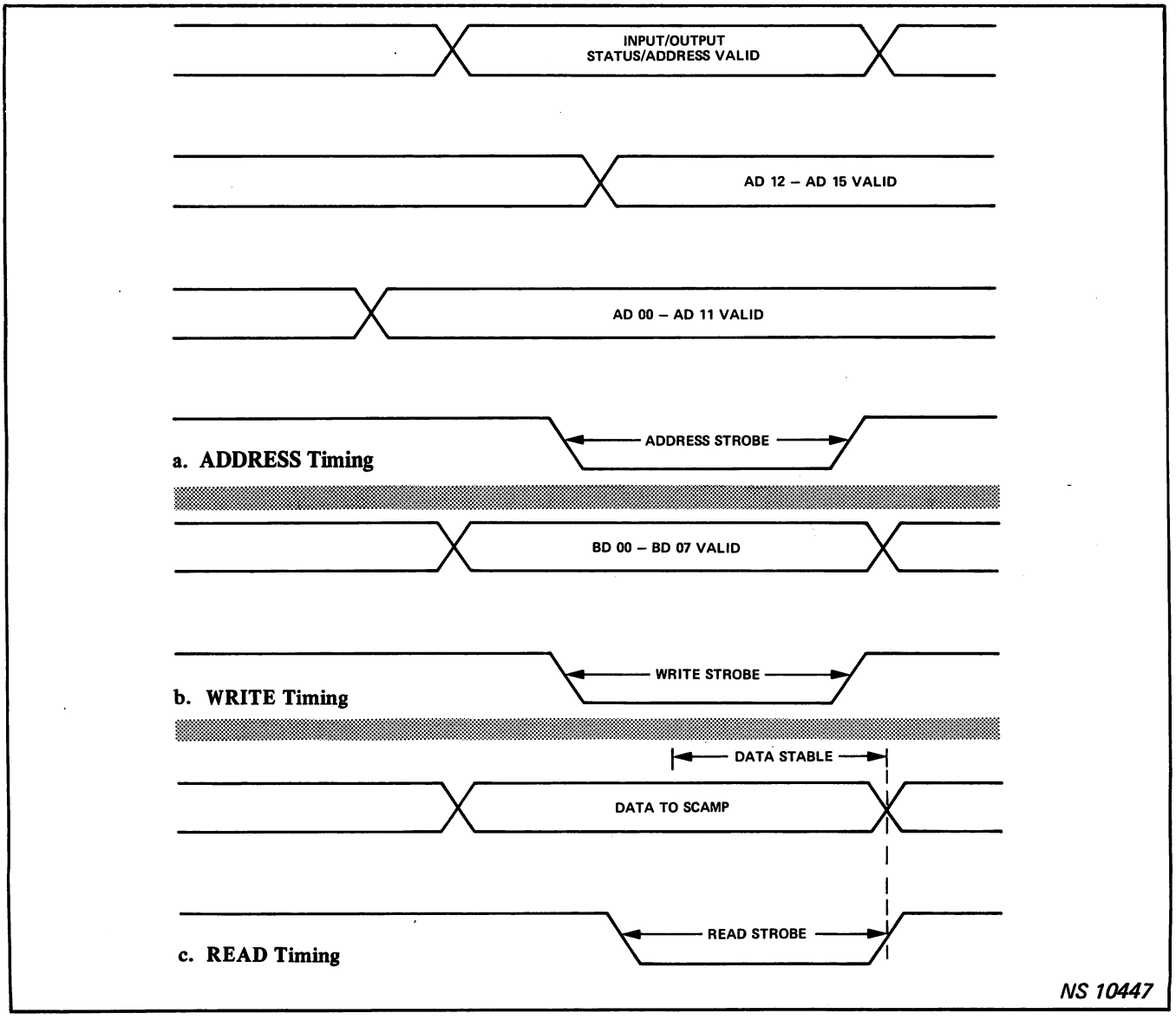


Figure 3-3. Timing Summary of CPU Module

3.3 RAM APPLICATION MODULE

A functional block diagram and timing summary of the RAM application module is shown in figure 3-4. Basically, the module consists of 16 1K-by-1 RAMs set up in a 2K-by-8 array (2,048 bytes), input/output buffers, and the required control circuits and module-select logic. The buffers are LSI devices that draw extremely low input currents; a 65K-byte memory can be configured without excessive loading of the buses. Both the read (BRDS) and write (BWDS) strobes are buffered and remain inactive until selected by the control circuits. The preset logic (PS1 through PS5) allows the RAM module to assume any one of 32 unique designations, thus providing a reasonable memory/peripheral complement for system development.

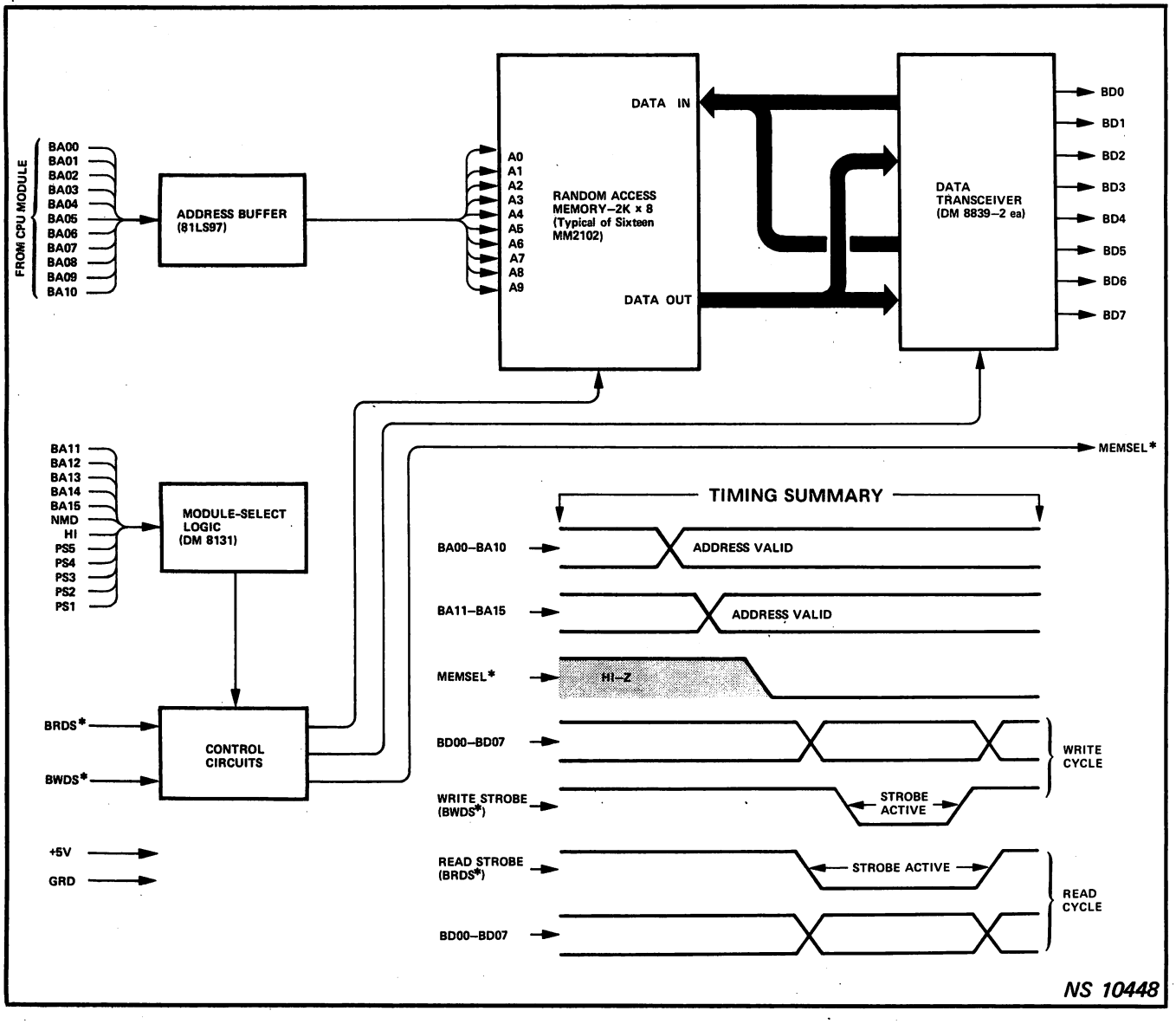


Figure 3-4. SC/MP RAM Application Module, Functional Block Diagram and Timing Summary

The preset logic can be implemented at the card-edge connector or the card itself. When the address code (BA11 through BA15) matches the preset code (PS1 through PS5), the module is selected – unless NMD is set low; as long as NMD is low, the module is disabled (that is, it cannot be selected). Unless the RAM module is selected, the MEMSEL* signal is high-impedance (open-circuit); with the card selected, the signal is active low.

3.4 PROM/ROM APPLICATION MODULE

Except for control differences and memory array, the PROM/ROM module shown in figure 3-5 is functionally equivalent to the RAM module described in the preceding paragraph. Since the PROM/ROM module uses a 4K-by-8 array, 12 bits are required for address and only four presets (PS1 through PS4) are required for module selection. Because memory is read-only, a write (NWDS) strobe is not necessary. PROM or ROM devices can be plugged into the module.

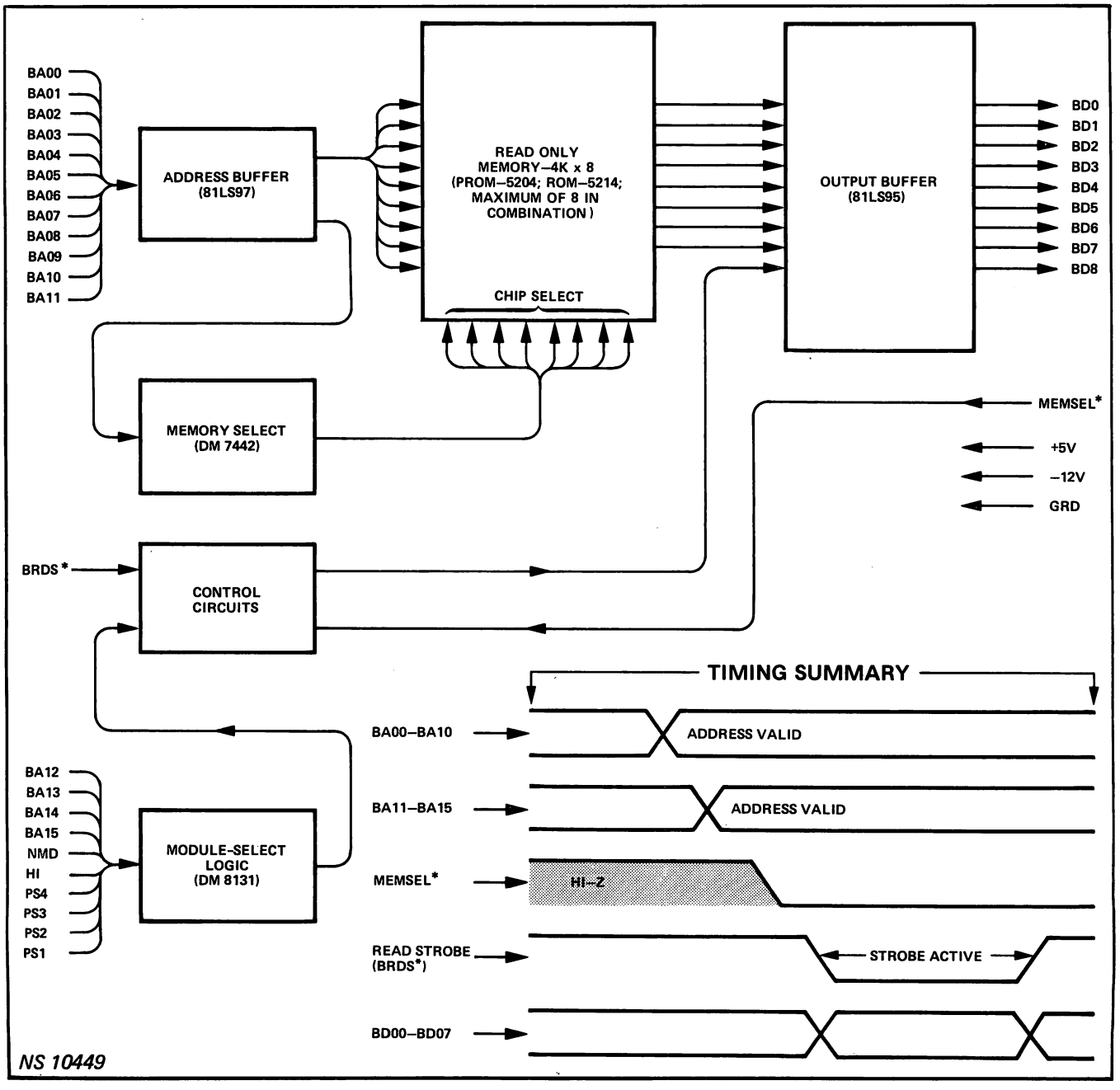


Figure 3-5. SC/MP PROM/ROM Application Module – Functional Block Diagram and Timing Summary

Chapter 4

SC/MP SYSTEMS

4.1 INTRODUCTION

Presently, there are two SC/MP-based systems – (1) the "SC/MP Development System" for those users whose primary concern is "low-cost" development and (2) the "Universal Development System" for those users who require a complete software complement and must interface with a number of peripherals.

4.2 SC/MP DEVELOPMENT SYSTEM

The SC/MP low-cost system is centered around the applications modules described in chapter 3. It provides an excellent, yet inexpensive debugging tool for almost any hardware application. A simple control panel permits the user to display and modify CPU registers, memory contents, and to view step-by-step actions and reactions. Registers and memory locations can be examined, modified, and debugged; thus, the overall development cycle can be shortened by weeks – even months.

4.3 UNIVERSAL DEVELOPMENT SYSTEM

This system is heavily supported by software, a large complement of input/output peripherals, and interface features that provide efficiency in both software and hardware development.

Chapter 5

SC/MP SUPPORT FUNCTIONS

5.1 TECHNICAL CONSULTATION

Once your microprocessing needs are defined, the following questions arise:

a. How can SC/MP solve my problem?
b. How soon can SC/MP solve my problem?
c. Is the SC/MP solution the one I want?

There are many ways SC/MP and its supporting chips can be used to solve a particular problem. The sales representative works continually with our applications engineers and other highly specialized technical personnel to solve your particular problem; that is how SC/MP can best serve you. Your application may require lots of hardware and little software; little hardware and lots of software; or, generally, some in-between compromise. SC/MP with its supporting family of input/output chips, control chips, memory chips, and other peripherals plus the SC/MP development software are adaptable to a large variety of system configurations. Also, you can depend on the counsel of our microprocessor field specialists and in-house applications engineers to assist you in applying SC/MP to solve your particular design problems.

"How long does all of this take?" Well, SC/MP and supporting chips are available now, so it is really a matter of when to start; and, of course, that is up to you. Last, but not least, "Is SC/MP the way to go?" Even our competitors would think twice before saying NO to this question and believe it – we can provide a lot of good reasons for saying YES.

Let us now look at specifics in the overall consultation/product-support chain that links National Semiconductor to you, the user.

5.1.1 Microprocessor Specialist

The area sales representative who responds to your initial inquiry is ably assisted by a microprocessor specialist. This individual is equipped technically to help analyze your application, to translate your needs into a viable hardware/software configuration, and then to follow it through to system delivery. Even though the specialist knows microprocessing systems, in fact, knows them very well, he still must have the best of tools, and these he has in SC/MP and its supporting family of chips – in addition to the broad line of semiconductor products manufactured by National Semiconductor.

Over and above the hardware tools, the field specialist has instant access to a very select assortment of people-tools; these include experts in applications, design, manufacturing and marketing. In short, the microprocessor specialist provides a flexible technical interface that can assist you, the user, in any part of the microprocessing spectrum – from very simple to very complex applications.

To best serve our customers in the U.S.A and nearby territories, each of four general areas are divided into regions with each region being serviced by at least one microprocessor specialist. Likewise, our international customers have access to the same expertise (in their native language) from a number of points on the globe – Germany for European customers, Japan for Asian customers, Scotland for United Kingdom customers, Australia for customers in that part of the world; and our U.S. offices serve our good neighbors South of the border, while our Canadian neighbors are served either from Canadian offices or U.S. offices. As you can see, SC/MP and its supporting family of chips are multilingual and have no geographic barriers.

5.1.2 Applications Support

So far we have met people in the field whose primary concerns are sales, system delivery, and operating integrity of the delivered equipment. Let us now examine the next link in the product-support chain – "how to use microprocessors." It is not practical or economically feasible to have a factory consultant stay with each microprocessor we sell, so we do the next best thing. Our applications engineers have anticipated most of your problems and have generated application notes that, in most cases, will provide a solution. Not only is the application note a problem-solving device, it also is a functional tool that may open up areas of use not previously explored. The applications service is free and we urge you to use it – think of it as free manpower because that is exactly what it is.

5.2 TRAINING

He who trains is he who understands and to this purpose, National Semiconductor operates three training centers. The Eastern center is located in Miami, Florida; the Midwestern center in Dallas, Texas; and the Western center is located near San Francisco, in Santa Clara, California. Each training center is fully equipped and professionally staffed to provide students with a good mix of hardware/software theory and hands-on laboratory experience. Course curricula vary in complexity from "Fundamentals of Microprocessors" for those technical personnel who have never worked with programmable systems to "Advanced Programming" for those who have microprocessor backgrounds.

Currently, the following courses are offered to support SC/MP.

MICROPROCESSOR FUNDAMENTALS – This course is designed for the engineer, technician, or manager who is not familiar with programmable systems. It covers stored program concepts, number systems, logic, input/output control, use of standard software (assemblers, editors, loaders, debug, subroutine packages), simple programming concepts, and an overview of available microprocessors with a guideline of how to select a microprocessor for a specific application. There are no prerequisites for this course, but a knowledge of digital design techniques, binary numbers, hexadecimal numbers, and Boolean algebra would be helpful.

SC/MP APPLICATIONS – This is an in-depth course covering the SC/MP microprocessor. Subjects covered are architecture, instruction set, input/output structure, interface design, applications design, use of development systems, available peripherals, and standard and optional software. Lab time is emphasized using experimental peripheral devices and development systems. Prerequisites: knowledge of basic microprocessor concepts, use of standard software such as assemblers and utilities, some exposure to assembly language programming, and some knowledge of interfacing techniques. Anyone unversed in any of these subjects should attend the Microprocessor Fundamentals Course before attending the SC/MP Application Course.

ADVANCED PROGRAMMING – Many engineers and programmers are finding that programming a microprocessor for a real-time application is considerably different from programming a minicomputer for a data-processing job. This course is designed for the engineer or programmer who must write complex applications software. Some of the subjects covered are real-time concepts, fixed-frequency events, time-of-day events, random external events, interrupt programming, real-time subroutines, program-called subroutines, programming complex math functions, hardware/software trade-offs, and system timing considerations. The IMP-16, PACE, and SC/MP microprocessors are used as the training machines in this course. Numerous examples are given, and the student is required to solve numerous problems in the training center laboratory. Prerequisites: thorough understanding of microprocessor fundamentals and characteristics; an understanding of assemblers, editors, debugs, loaders, and subroutine libraries; experience in programming at the assembly-language level.

NOTE

The Microprocessor Fundamentals Course alone is not adequate preparation for this course. However, anyone who has the experience described but would like some refresher training before attending this course should attend the IMP-16/PACE Applications or SC/MP Applications Course.

All courses include four days of class schedules (Monday through Thursday) and a fifth day where the laboratory is opened to students for additional hands-on experience and for individual consultation with instructors.

Tuition for each course is $395.00, payable when enrollment is accepted. Enrollments should be made at least two weeks in advance of scheduled class start. Before remitting tuition, we suggest you check with the appropriate training center (listed below) to be sure space is available in the desired course. To ensure adequate facilities, class sizes are limited.

Training center addresses and telephone numbers are as follows:

Eastern Microprocessor Training Center
National Semiconductor Corporation
2721 Bayshore Drive South, Suite 121
Miami, Florida 33133
Telephone: (305) 446-8309

Central Microprocessor Training Center
National Semiconductor Corporation
13773 North Central Expressway, Suite 1132
Dallas, Texas 75231
Telephone: (214) 690-4552

Western Microprocessor Training Center/470
National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Telephone: (408) 732-5000, Ext. 7183

5.3 FACTORY SERVICE

It would be nice to say that SC/MP and its supporting chips never fail and are never damaged; however, this would not be a credible statement. Nonetheless, failures are rare and the reasons are worth mentioning. In many companies, quality control occurs at the end of the design and manufacturing cycle; hence, they check overall performance of the chain without much regard for the individual links. At National Semiconductor, we believe in the old adage, no chain is stronger than its weakest link – and we react accordingly. Quality control is an integral part of each link in the chain – from concept through completion. Even after each link meets performance specifications, the end product is again checked for integrity of operation and, after a prolonged burn-in, it is further checked for reliability.

According to "Murphy's Law," some failures will still occur in the field. If it happens to you, here is what you can expect from us. Upon receipt of the failed card or system, we strive for a turn-around time of five working days and, in most cases, we are successful. If the equipment is under warranty, there is no charge for repairs; you pay the freight one way, and we pay it the other way. If equipment is not under warranty, you are charged at the rate of $35 per hour plus parts and applicable tax; the minimum charge is $50.

Questions you may have should be addressed to personnel that will provide the fastest response: may we suggest the sales representative for anything relating to price and delivery and your microprocessor specialist as a first source for technical questions. Feel free to contact applications engineering at National Semiconductor for those technical questions the specialist cannot answer.

5.4 USER GROUP

National Semiconductor sponsors COMPUTE (Club Of Microprocessor Programmers, Users, and Technical Experts). This user group is dedicated to the world-wide distribution of your ideas and techniques relating to the use of microprocessors. Members of COMPUTE communicate on a regular basis by way of *The Bit Bucket*, a newsletter published by National Semiconductor. In *The Bit Bucket*, you will find everything from soup to nuts – even a user-submitted software library. So get involved with SC/MP and COMPUTE; they make an excellent partnership. You can meet the former by calling your nearest sales representative and the latter by writing to the following:

COMPUTE/470
National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Telephone: (408) 732-5000, Ext. 7183

Don't forget the /470 in the address. That's our mail stop, and your letter will be delayed (or worse yet, lost) without it.

NOTES

Appendix A

INSTRUCTION SET SUMMARY AND RELATED INFORMATION

A.1 INTRODUCTION

Tables A-1, A-2, and A-3, respectively, define the SC/MP instruction set in terms of symbols and notations, memory-reference formats, and a descriptive summary of each instruction. Figure A-1 shows the bus utilization for each instruction with reference to the overall input/output period in microcycles. The information in figure A-1 is a valuable design aid when SC/MP processors are used in a cascade arrangement, or in other applications that require extensive time-sharing of the buses.

Table A-1. Symbols and Notations

Symbol and Notation	Meaning
AC	8-bit Accumulator.
CY/L	Carry/Link Flag in the Status Register.
data	Signed, 8-bit immediate data field.
disp	Displacement, represents an operand in a nonmemory reference instruction or an address modifier field in a memory reference instruction. It is a signed twos-complement number.
EA	Effective Address as specified by the instruction.
E	Extension Register; provides for temporary storage, variable displacements, and separate serial input/output port.
i	Unspecified bit of a register.
IE	Interrupt Enable Flag.
m	Mode bit, used in memory reference instructions. Blank parameter sets m = 0, @ sets m = 1.
OV	Overflow Flag in the Status Register.
PC	Program Counter (Pointer Register 0); during address formation, PC points to the last byte of the instruction being executed.
ptr	Pointer Register (ptr = 0 through 3). The register specified in byte 1 of the instruction.
$ptr_{n:m}$	Pointer register bits; n:m = 7 through 0 or 15 through 8.
SIN	Serial Input pin.
SOUT	Serial Output pin.
SR	8-bit Status Register.
( )	Means "contents of." For example, (EA) is contents of Effective Address.
[ ]	Means optional field in the assembler instruction format.
$\sim$	Ones complement of value to right of $\sim$.
$\rightarrow$	Means "replaces."
$\leftarrow$	Means "is replaced by."
$\longleftrightarrow$	Means "exchange."
@	When used in the operand field of instruction, sets the mode bit (m) to 1 for auto-incrementing/auto-decrementing indexing.
10^+	Modulo 10 addition.
$\wedge$	AND operation.
$\vee$	Inclusive-OR operation.
$\bar{\vee}$	Exclusive-OR operation.
$\geqslant$	Greater than or equal to.
=	Equals.
$\neq$	Does not equal.

Table A-2. SC/MP – Memory Reference Formats

Addressing	Operand Formats			
	Object Produced By Assembler			Assembler Source Input
	m	ptr	disp*	
PC-Relative	0	0	–128 to +127	disp
Indexed	0	1, 2, or 3	–128 to +127	disp (ptr)
Auto-indexing	1	1, 2, or 3	–128 to +127	@ disp (ptr)

*Note: If disp = –128, then (E) is substituted for (disp) in calculating EA as well as in performing auto-indexing.

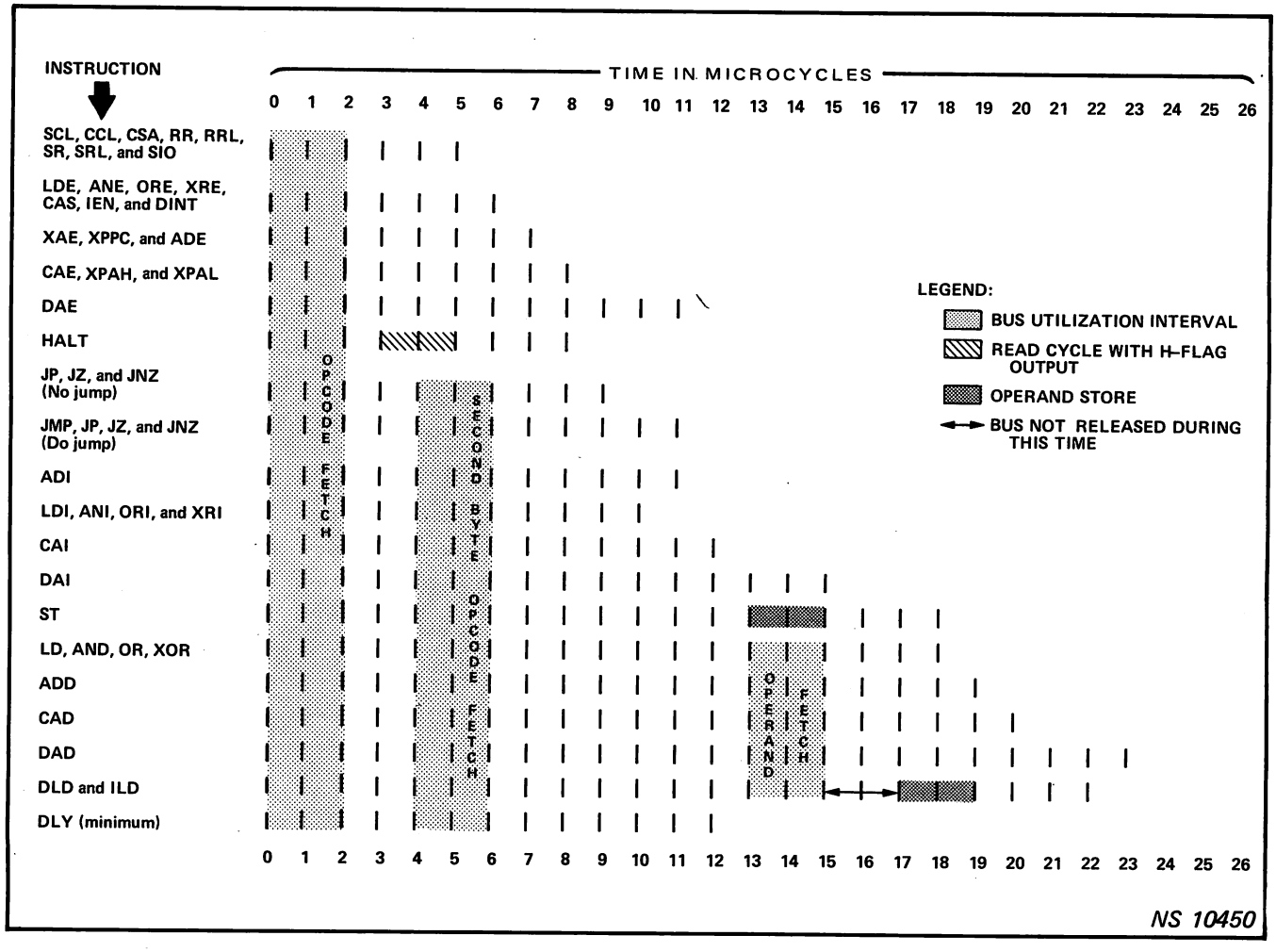


Figure A-1. Bus Utilization of Each Instruction

Table A-3. SC/MP Instruction Summary

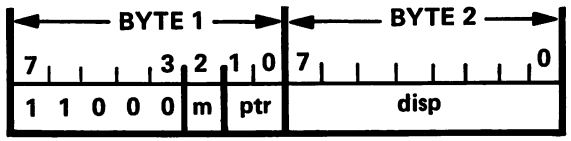
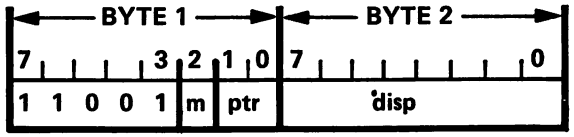
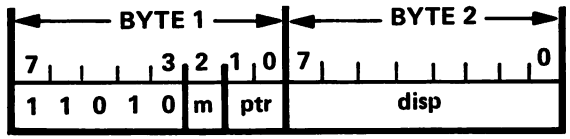
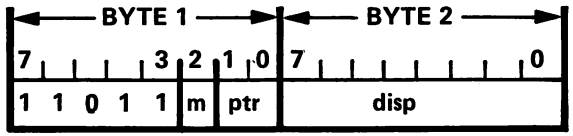
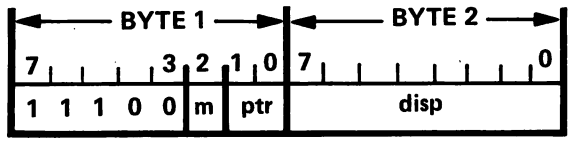
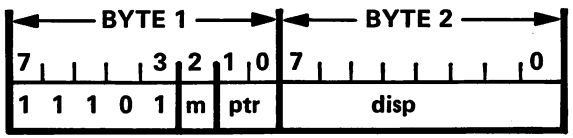
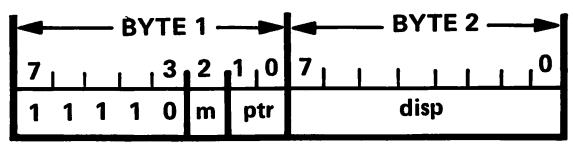
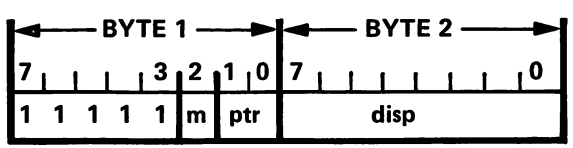
Instruction/Mnemonic	Operation/Description	Assembler Format	Execution Time (Microcycles)
MEMORY REFERENCE INSTRUCTIONS			
LOAD LD(C0)  	(AC) ← (EA) The contents of the Accumulator (AC) are replaced by the contents of the Effective Address (EA). The initial contents of AC are lost; the contents of EA are unaltered.	LD disp (ptr) disp @ (ptr) disp	18
STORE ST(C8)  	(EA) ← (AC) The contents of the Effective Address (EA) are replaced by the contents of the Accumulator (AC). The initial contents of EA are lost; the contents of AC are unaltered.	ST disp (ptr) disp @ (ptr) disp	18
AND AND(D0)  	(AC) ← (AC) ∧ (EA) The contents of the Accumulator (AC) are ANDed with the contents of the Effective Address (EA), and the result is stored in AC. The initial contents of AC are lost; the contents of EA are unaltered.	AND disp (ptr) disp @ (ptr) disp	18
OR OR(D8)  	(AC) ← (AC) ∨ (EA) The contents of the Accumulator (AC) are inclusive-ORed with the contents of the Effective Address (EA), and the result is stored in AC. The initial contents of AC are lost; the contents of EA are unaltered.	OR disp (ptr) disp @ (ptr) disp	18
EXCLUSIVE-OR XOR(E0)  	(AC) ← (AC) ∇ (EA) The contents of the Accumulator (AC) are exclusive-ORed with the contents of the Effective Address (EA), and the result is stored in AC. The initial contents of AC are lost; the contents of EA are unaltered.	XOR disp (ptr) disp @ (ptr) disp	18

Table A-3. SC/MP Instruction Summary (Continued)

Instruction/Mnemonic	Operation/Description	Assembler Format		Execution Time (Microcycles)
DECIMAL ADD **DAD(E8)** 	$(AC) \leftarrow (AC)_{10} + (EA)_{10} + (CY/L)$; CY/L The contents of the Accumulator (AC) and the contents of the Effective Address (EA) are treated as 2-digit binary-coded-decimal numbers greater than or equal to zero, and less than or equal to ninety-nine ($0 \leqslant n \leqslant 99$). The contents of AC and EA and the Carry (CY/L) are added, and the 2-digit binary-coded-decimal sum is stored in AC. The initial contents of AC are lost; the contents of EA are unaltered. The Carry Flag in the Status Register is set if a carry occurs from the most significant decimal digit; otherwise, it is cleared. The Overflow Flag is not affected.	DAD	disp (ptr) disp @ (ptr) disp	23
ADD **ADD(F0)** 	(AC)←(AC) + (EA) + (CY/L); (CY/L), (OV) The contents of the Accumulator (AC) and the contents of the Effective Address (EA) are treated as 8-bit binary twos-complement numbers. The contents of the Accumulator (AC), the Effective Address (EA), and the Carry (CY/L) are added algebraically, and the sum is stored in AC. The Carry Flag in the Status Register is set if a carry from the most significant bit position occurs; otherwise, it is cleared. The Overflow (OV) Flag in the Status Register is set if an overflow occurs (that is, if the sign of the results differs from the sign of both operands); otherwise, the Overflow Flag is cleared.	ADD	disp (ptr) disp @ (ptr) disp	19
COMPLEMENT AND ADD **CAD(F8)** 	(AC)←(AC) + ~(EA) + (CY/L); CY/L, OV The contents of the Accumulator (AC) and the contents of the Effective Address (EA) are treated as 8-bit binary numbers. The contents of the Accumulator (AC), the ones complement of the contents of the Effective Address (EA), and the Carry (CY/L) are added algebraically, and the sum is stored in AC. The initial contents of AC are lost; the contents of EA are unaltered. The Carry Flag (CY/L) in the Status Register is set if a carry from the most significant bit position occurs; otherwise,	CAD	disp (ptr) disp @ (ptr) disp	20

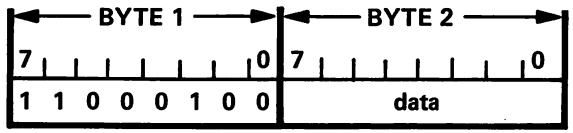
it is cleared. The Overflow Flag (OV) in the Status Register is set if the sign of the result is the same as the sign of (EA) and opposite the sign of (AC); otherwise, it is cleared.

NOTE

If the CY/L Flag is cleared initially, the logical (ones) complement of (EA) is added to the Accumulator. If the CY/L Flag is set, the twos complement of (EA) is added.

IMMEDIATE INSTRUCTIONS

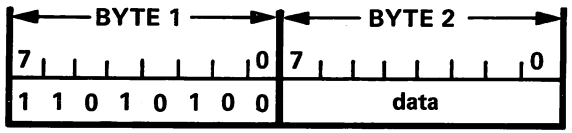
LOAD IMMEDIATE — LDI(C4)



(AC)←(data)
The contents of the Accumulator (AC) are replaced by the data byte. The initial contents of AC are lost; the data byte is unaltered.

LDI	data	10

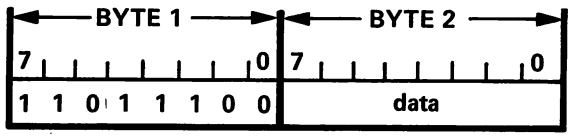
AND IMMEDIATE — ANI(D4)



(AC)←(AC)∧(data)
The contents of the Accumulator (AC) are ANDed with the data byte, and the result is stored in AC. The initial contents of AC are lost; the data byte is unaltered.

ANI	data	10

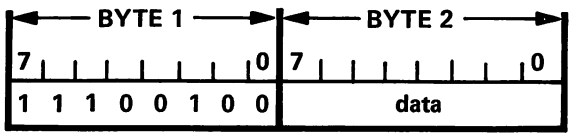
OR IMMEDIATE — ORI(DC)



(AC)←(AC)∨(data)
The contents of the Accumulator (AC are inclusive-ORed with the data byte, and the result is stored in AC. The initial contents of AC are lost; the data byte is unaltered.

ORI	data	10

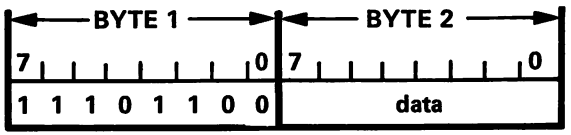
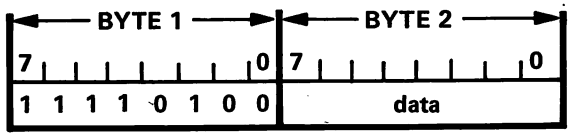
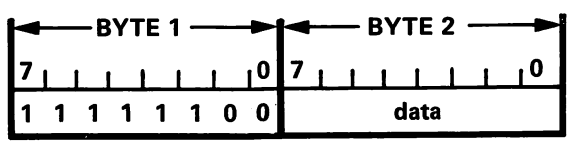
EXCLUSIVE-OR IMMEDIATE — XRI(E4)



(AC)←(AC) ∇ (data)
The contents of the Accumulator (AC) are exclusive-ORed with the data byte, and the result is stored in AC. The initial contents of AC are lost; the data byte is unaltered.

XRI	data	10

Table A-3. SC/MP Instruction Summary (Continued)

Instruction/Mnemonic	Operation/Description	Assembler Format	Execution Time (Microcycles
DECIMAL ADD IMMEDIATE **DAI(EC)**  	$(AC) \leftarrow (AC)_{10} + (data)_{10} + (CY/L), (CY\text{-}L)$ The data byte and the contents of the Accumulator are treated as 2-digit binary-coded-decimal numbers. The contents of the Accumulator (AC), the data byte, and the Carry (CY/L) are added, and the 2-digit binary-coded-decimal sum is stored in AC. The initial contents of AC are lost; the data byte is unaltered. The Carry Flag in the Status Register is set if a carry from the most significant decimal digit occurs; otherwise, it is cleared. The Overflow Flag is not affected.	DAI data	15
ADD IMMEDIATE **ADI(F4)**  	$(AC) \leftarrow (AC) + (data) + (CY/L); (CY/L), (OV)$ Data and the contents of the Accumulator (AC) are treated as 8-bit twos-complement numbers. The contents of the Accumulator (AC), the data byte, and the Carry (CY/L) are added algebraically, and the sum is stored in AC. The initial contents of AC are lost; the data byte is unaltered. The Carry Flag in the Status Register is set if a carry from the most significant bit position occurs; otherwise, it is cleared. The Overflow Flag (OV) in the Status Register is set if the sign of the result differs from the sign of both operands; otherwise, it is cleared.	ADI data	11
COMPLEMENT AND ADD IMMEDIATE **CAI(FC)**  	$(AC) \leftarrow (AC) + \sim(data) + (CY/L); (CY/L), (OV)$ The data byte and the contents of the Accumulator (AC) are treated as 8-bit numbers. The contents of the Accumulator (AC), the ones complement of the data byte, and the Carry (CY/L) are added algebraically and the result is stored in AC. The initial contents of AC are lost; the data byte is unaltered. The Carry Flag in the Status Register is set if a carry from the most significant bit position occurs; otherwise, it is cleared. The Overflow Flag (OV) in the Status Register is set if the sign of the result is the same as the sign of the data byte and opposite the sign of (AC); otherwise, it is cleared.	CAI data	12

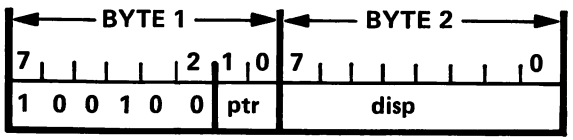
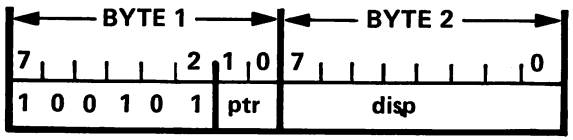
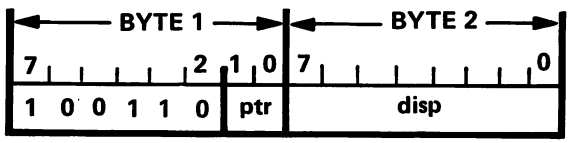
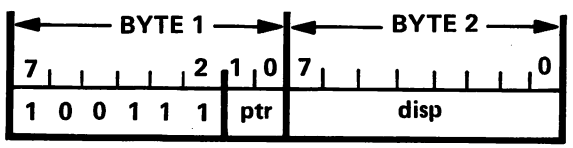
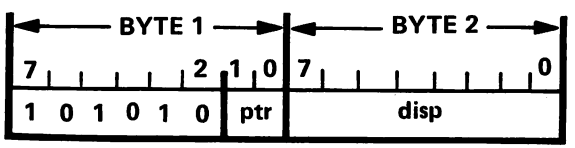
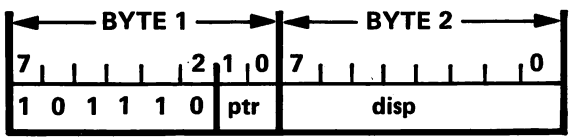
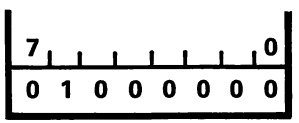
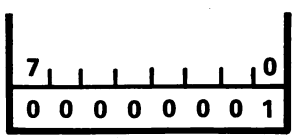
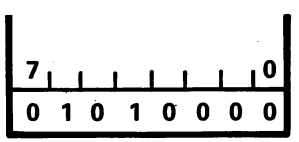
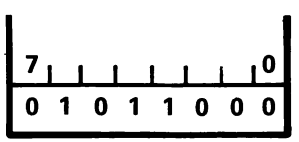
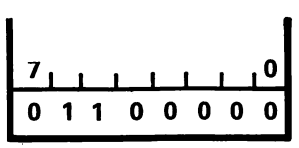
	NOTE If the CY/L Flag is set initially, this operation is equivalent to subtracting the data byte from the Accumulator.			
TRANSFER INSTRUCTIONS				
JUMP JMP(90)  	**(PC) ← EA** The Effective Address (EA) replaces the contents of the Program Counter (PC). The next instruction is fetched from the location designated by the new contents of PC + 1.	**JMP**	**disp (ptr)**	11
JUMP IF POSITIVE JP(94)  	**IF (AC) ⩾ 0, (PC)←EA** If the contents of the Accumulator (AC) are positive or zero, the Effective Address (EA) replaces the contents of the Program Counter (PC). The next instruction is fetched from the location designated by the new contents of PC + 1.	**JP**	**disp (ptr)**	9 (no jump); 11 (jump)
JUMP IF ZERO JZ(98)  	**IF (AC) = 0, (PC)←EA** If the contents of the Accumulator (AC) are zero, the Effective Address (EA) replaces the contents of the Program Counter (PC). The next instruction is fetched from the location designated by the new contents of the PC + 1.	**JZ**	**disp (ptr)**	9 (no jump); 11 (jump)
JUMP IF NOT ZERO JNZ(9C)  	**IF (AC) ≠ 0, (PC)←EA** If the contents of the Accumulator (AC) are not zero, the Effective Address (EA) replaces the contents of the Program Counter (PC). The next instruction is fetched from the location designated by the new contents of the PC + 1.	**JNZ**	**disp (ptr)**	9 (no jump); 11 (jump)
MEMORY INCREMENT/DECREMENT INSTRUCTIONS				
INCREMENT AND LOAD ILD(A8)  	**(AC), (EA)←(EA) + 1** The contents of the Effective Address (EA) are incremented by 1, and the result is stored in the Accumulator (AC) and, also, in EA. The initial contents of AC and EA are lost. The Carry and Overflow Flags are not affected.	**ILD**	**disp** **disp (ptr)**	22

Table A-3. SC/MP Instruction Summary (Continued)

Instruction/Mnemonic	Operation/Description	Assembler Format	Execution Time (Microcycles)
DECREMENT AND LOAD **DLD(B8)** 	(AC), (EA)←(EA) - 1 The contents of the Effective Address (EA) are decremented by 1, and the result is stored in the Accumulator (AC) and, also, in EA. The initial contents of AC and EA are lost. The Carry and Overflow Flags are not affected.	DLD disp disp (ptr)	22
EXTENSION REGISTER INSTRUCTIONS			
LOAD FROM EXTENSION **LDE(40)** 	(AC)←(E) The contents of the Accumulator (AC) are replaced by the contents of the Extension Register (E). The initial contents of AC are lost; the contents of E are unaltered.	LDE	6
EXCHANGE AC AND EXTENSION **XAE(01)** 	(AC)↔(E) The contents of the Accumulator (AC) are exchanged with the contents of the Extension Register (E).	XAE	7
AND EXTENSION **ANE(50)** 	(AC)←(AC)∧(E) The contents of the Accumulator (AC) are ANDed with the contents of the Extension Register (E), and the result is stored in AC. The initial contents of AC are lost; the contents of E are unaltered.	ANE	6
OR EXTENSION **ORE(58)** 	(AC)←(AC)∨(E) The contents of the Accumulator (AC) are inclusive-ORed with the contents of the Extension Register (E), and the result is stored in AC. The initial contents of AC are lost; the contents of E are unaltered.	ORE	6
EXCLUSIVE-OR EXTENSION **XRE(60)** 	(AC)←(AC)∇(E) The contents of the Accumulator (AC) are exclusive-ORed with the contents of the Extension Register (E), and the result is stored in AC. The initial contents of AC are lost; the contents of E are unaltered.	XRE	6

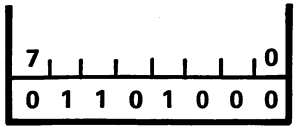
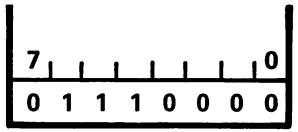
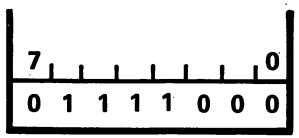
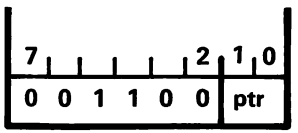
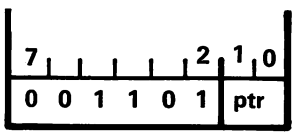
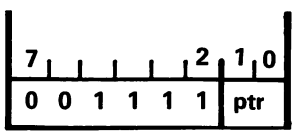
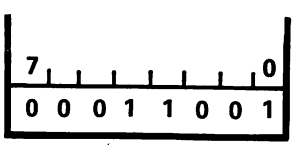
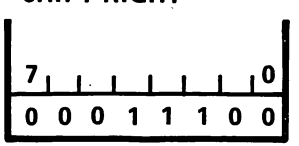
DECIMAL ADD EXTENSION 	**DAE(68)**	$(AC) \leftarrow (AC)_{10} + (E)_{10} + (CY/L)$, (CY/L) The contents of the Accumulator (AC) and the Extension Register (E) are treated as 2-digit binary-coded-decimal numbers, greater than or equal to zero. The contents of the Accumulator (AC), Extension Register (E), and the Carry (CY/L) are added, and the sum is stored in AC. The initial contents of AC are lost; the contents of E are unaltered. The Carry Flag in the Status Register is set if a carry from the most significant decimal digit occurs; otherwise, it is cleared. The Overflow Flag is not affected.	**DAE**	11
ADD EXTENSION 	**ADE(70)**	(AC)←(AC) + (E) + (CY/L); (CY/L), (OV) The contents of the Accumulator (AC) and the Extension Register (E) are treated as 8-bit binary, twos-complement numbers. The contents of the Accumulator (AC), Extension Register (E), and the Carry (CY/L) are added algebraically, and the sum is stored in AC. The initial contents of AC are lost; the contents of E are unaltered. The Carry Flag (CY/L) in the Status Register is set if a carry from the most significant bit position occurs; otherwise, it is cleared. The Overflow Flag (OV) in the Status Register is set if the sign of the result differs from the sign of both operands; otherwise, it is cleared.	**ADE**	7
COMPLEMENT AND ADD EXTENSION 	**CAE(78)**	(AC)←(AC) + ~(E) + (CY/L); (CY/L), (OV) The contents of the Accumulator (AC) and Extension Register (E) are treated as 8-bit binary numbers. The contents of the Accumulator (AC), the ones complement of the contents of the Extension Register (E), and the Carry (CY/L) are added algebraically, and the result is stored in AC. The initial contents of AC are lost; the contents of E are unaltered. The Carry Flag (CY/L) in the Status Register is set if a carry from the most significant bit position occurs; otherwise, it is cleared. The Overflow Flag (OV) in the Status Register is set if the sign of the result is the same as the sign of (E) and opposite the sign of (AC); otherwise, it is cleared. NOTE If the CY/L Flag is set initially, this operation is equivalent to subtracting the contents of E from the contents of AC.	**CAE**	8

Table A-3. SC/MP Instruction Summary (Continued)

Instruction/Mnemonic	Operation/Description	Assembler Format	Execution Time (Microcycles)
POINTER REGISTER MOVE INSTRUCTIONS			
EXCHANGE POINTER LOW XPAL(30)  	$(AC) \leftrightarrow (PTR_{7:0})$ The contents of the Accumulator (AC) are exchanged with the low-order byte (bits 7 through 0) of the designated Pointer Register (PTR).	XPAL ptr	8
EXCHANGE POINTER HIGH XPAH(34)  	$(AC) \leftrightarrow (PTR_{15:8})$ The contents of the Accumulator (AC) are exchanged with the high-order byte (bits 15 through 8) of the designated Pointer Register (PTR).	XPAH ptr	8
EXCHANGE POINTER WITH PC XPPC(3C)  	$(PC) \leftrightarrow (PTR)$ The contents of the Program Counter (PC) are exchanged with the designated Pointer Register (PTR)	XPPC ptr	7
SHIFT, ROTATE, SERIAL INPUT/OUTPUT INSTRUCTIONS			
SERIAL INPUT/OUTPUT SIO(19)  	$(E_i) \rightarrow (E_{i-1})$, SIN $\rightarrow (E_7)$, $(E_0) \rightarrow$ SOUT The contents of the Extension Register (E) are shifted right one bit. The initial content of bit 0 is shifted to the data output pin SOUT. The signal on the data input pin SIN is shifted into bit 7.	SIO	5
SHIFT RIGHT SR(1C)  	$(AC_i) \rightarrow (AC_{i-1}), 0 \rightarrow (AC_7)$ The contents of the Accumulator (AC) are shifted right one bit. The initial content of bit 0 is lost. Zero is shifted into bit 7.	SR	5

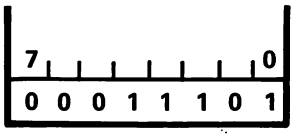
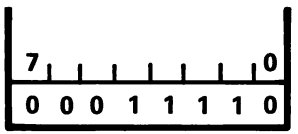
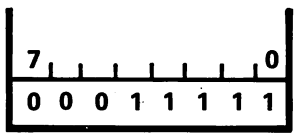
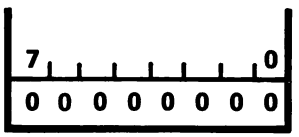
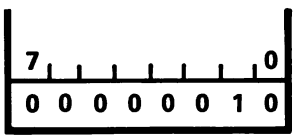
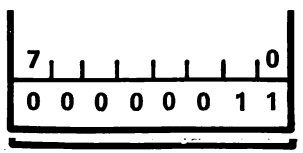
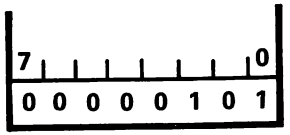
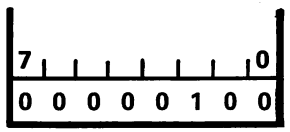
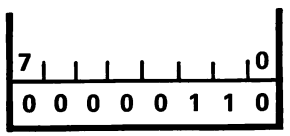
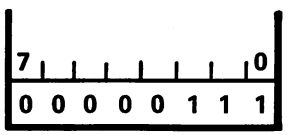
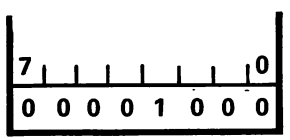
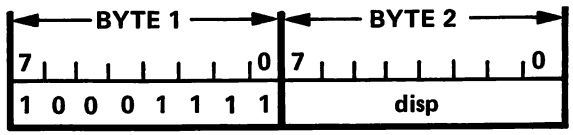
SHIFT RIGHT WITH LINK  	**SRL(1D)**	$(AC_i) \rightarrow (AC_{i-1})$, $(CY/L) \rightarrow (AC_7)$ The contents of the Accumulator are shifted right one bit. The initial content of bit 0 is lost. The Link (CY/L) Flag from the Status Register is shifted into bit 7. The Link Flag is not altered.	**SRL**	5
ROTATE RIGHT  	**RR(1E)**	$(AC_i) \rightarrow (AC_{i-1})$, $(AC_0) \rightarrow (AC_7)$ The contents of the Accumulator (AC) are rotated right one bit. The initial content of bit 0 is shifted into bit 7.	**RR**	5
ROTATE RIGHT WITH LINK  	**RRL(1F)**	$(AC_i) \rightarrow (AC_{i-1})$, $(AC_0) \rightarrow (CY/L) \rightarrow (AC_7)$ The contents of the Accumulator (AC) are rotated right one bit. The initial content of bit 0 is shifted into the Link Flag (CY/L) of the Status Register, and the initial content of the Link Flag is shifted into bit 7 of AC.	**RRL**	5
MISCELLANEOUS INSTRUCTIONS				
HALT  	**HALT(00)**	Pulse H-flag at I/O status time. However, in a particular application system, this instruction may be used for functions other than HALT. For detailed information on the hardware operation of the halt instruction, see the SC/MP Users Manual.	**HALT**	8
CLEAR CARRY LINK  	**CCL(02)**	$(CY/L) \leftarrow 0$ The Carry/Link Flag in the Status Register (SR) is cleared. The remaining bits in SR are not affected.	**CCL**	5
SET CARRY LINK  	**SCL(03)**	$(CY/L) \leftarrow 1$ The Carry/Link Flag in the Status Register (SR) is set. The remaining bits in SR are not affected.	**SCL**	5

Table A-3. SCMP Instruction Summary (Continued)

Instruction/Mnemonic		Operation/Description	Assembler Format	Execution Time (Microcycles)
ENABLE INTERRUPT  	**IEN(05)**	**(IE) ← 1** The Interrupt Enable (IE) Flag in the Status Register (SR) is set; the remaining bits in SR are not affected. The processor interrupt system is enabled. Interrupts will be processed as received *after* the next instruction is fetched and executed.	**IEN**	6
DISABLE INTERRUPTS  	**DINT(04)**	**(IE) ← 0** The Interrupt Enable (IE) Flag in the Status Register (SR) is cleared; the other bits in SR are not affected. The processor interrupt system is disabled. Interrupts which occur while the system is disabled will not be processed.	**DINT**	6
COPY STATUS TO AC  	**CSA(06)**	**(AC) ← (SR)** The contents of the Accumulator (AC) are replaced by the contents of the Status Register (SR). The initial contents of AC are lost; the contents of SR are not altered.	**CSA**	5
COPY AC TO STATUS  	**CAS(07)**	**(SR) ← (AC)** The contents of the Accumulator (AC) replace the contents of the Status Register (SR). SR bits 4 and 5 are external sense bits and are not affected by this instruction. The initial contents of SR (except for bits 4 and 5) are lost; the contents of AC are not altered. If IE is changed from 0 to 1 by this instruction, the interrupt system will be enabled *after* the next instruction is fetched and executed.	**CAS**	6
NO OPERATION  	**NOP(08)**	**(PC) ← (PC) + 1** The Program Counter (PC) is incremented by 1. The NOP instruction takes the minimum 5-microcycle execution time. Undefined opcodes encountered are considered to be one-byte or two-byte NOPs and may take 5 to 10 microcycles to execute, depending on the code.	**NOP**	5 (min) 10 (max)

DELAY **DLY(8F)**



DELAY = 13 + 2 (AC) + 2 disp + 2^9 disp

This instruction delays processing a variable length of time. The contents of the Accumulator (AC) and the Displacement (disp) are considered unsigned binary numbers (maximum value of each is 255). The number computed from the given equation is the execution time in microcycles. The following table gives some typical execution times. Range of delay is from 13 to 131593 microcycles.

DLY

13 (min)
131593 (max)

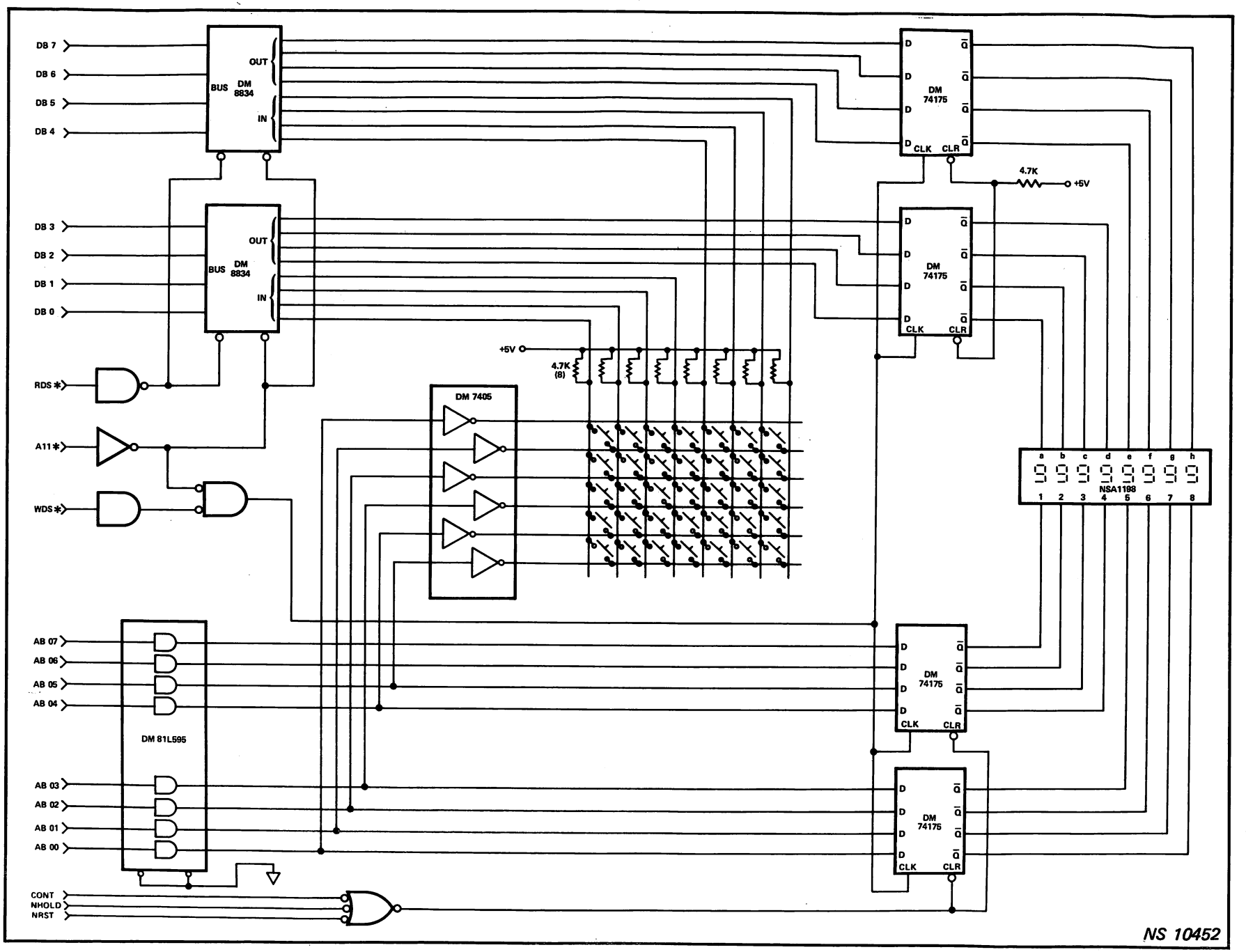
AC

disp	0	25	50	75	100	125	150	175	200	225
0	13	63	113	163	213	263	313	363	413	463
1	527	577	627	677	727	777	827	877	927	977
2	1041	1091	1141	1191	1241	1291	1341	1391	1441	1491
3	1555	1605	1655	1705	1755	1805	1855	1905	1955	2005
4	2069	2119	2169	2219	2269	2319	2369	2419	2469	2519
5	2583	2633	2683	2733	2783	2833	2883	2933	2983	3033
6	3097	3147	3197	3247	3297	3347	3397	3447	3497	3547
7	3611	3661	3711	3761	3811	3861	3911	3961	4011	4061
8	4125	4175	4225	4275	4325	4375	4425	4475	4525	4575
9	4639	4689	4739	4789	4839	4889	4939	4989	5039	5089
10	5153	5203	5253	5303	5353	5403	5453	5503	5553	5603

To determine AC and disp for a specific number of microcycles (m) use the following equations:

$$\text{disp} = \text{truncate } ((m-13)/514)$$
$$AC = ((m-13)-514(\text{disp}))/2$$

Using these equations, the delay time will be either exact or one microcycle less than the required number of microcycles.


DB 7
DB 6
DB 5
DB 4
BUS
DM 8834
OUT
IN
DB 3
DB 2
DB 1
DB 0
RDS *
A11 *
WDS *
+5V
4.7K
4.7K (8)
DM 7405
DM 74175
D
Q
CLK
CLR
NSA1198
AB 07
AB 06
AB 05
AB 04
DM 81L595
AB 03
AB 02
AB 01
AB 00
CONT
NHOLD
NRST
NS 10452

Appendix B

SC/MP INTERFACE WITH KEYBOARD AND DISPLAY

B.1 INTRODUCTION

Figure B-1 schematically shows how SC/MP can be used to interface with a keyboard and a display. The interface logic uses standard components and is capable of full numeric and partial alphabetic display.

B.2 DESCRIPTION

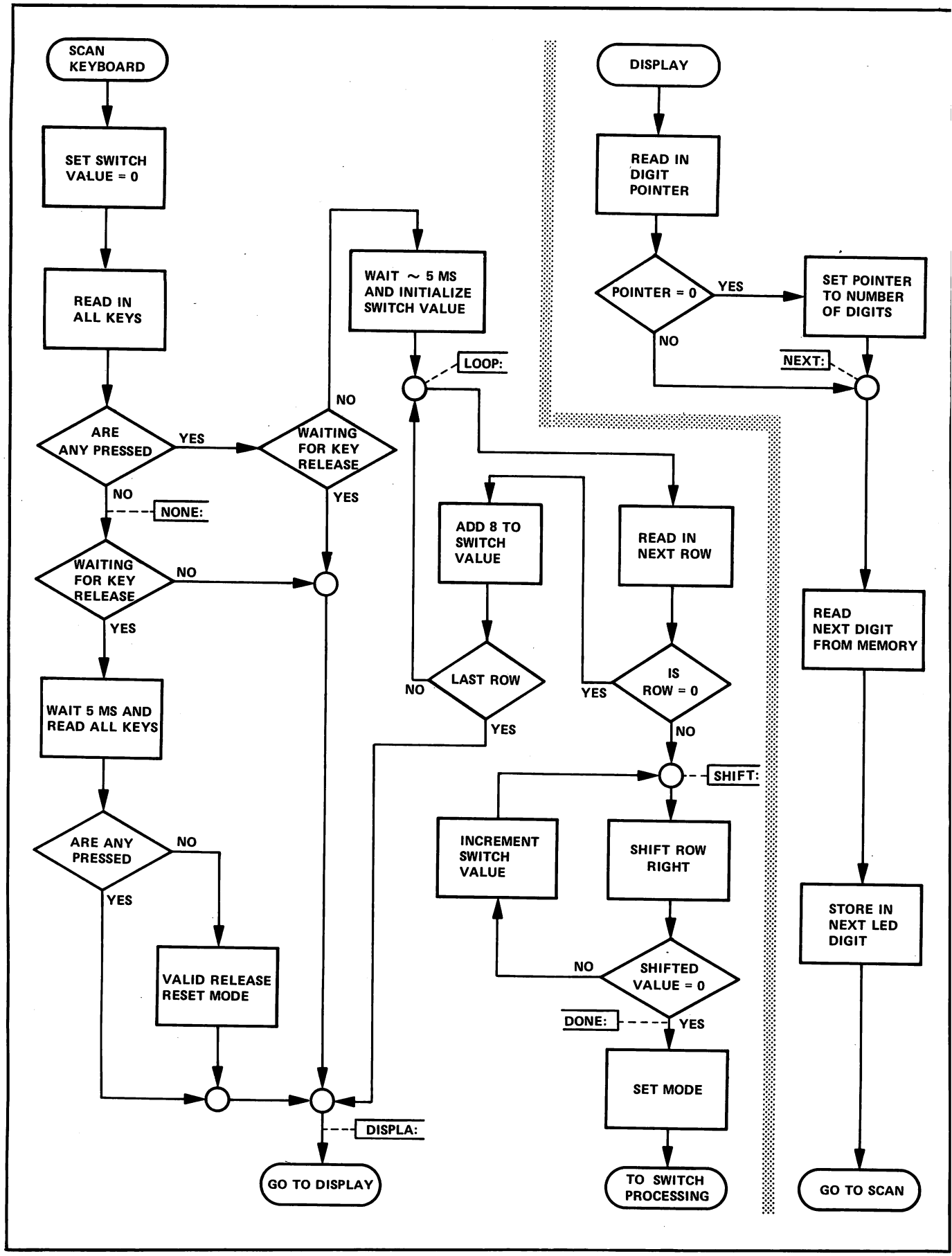
The keyboard is a matrix of single-contact (N.O.) push-button switches arranged in a 6–by–8 array. The input matrix could be a typewriter keyboard, a calculator keyboard, or even a musical-instrument keyboard. Although the example shown uses a 48-key array, the same interface concepts may be used for switching arrays of other sizes.

Information is read from the keyboard by software scanning. A row is driven with an open-collector inverter from the address bus. When a switch closure is made, the bit in the corresponding column is pulled down to a logic '0'. This signal is inverted by the DM8834 and is read onto the data bus when the interface is selected and the read strobe (RDS*) is low. Accordingly, a logic '1' appears in the bit positions corresponding to closed switches.

The keys can be read simultaneously to determine if any switches are closed. Reading of the matrix is accomplished by setting the 6 lower bits of the address high and performing a load from the keyboard-display interface. If there are no switch closures, the contents of the accumulator is zero. If a key is pressed (switch closed), the software performs a keyboard scan to determine the key that is depressed. In the software example that follows, the switch is binary represented in memory location SWITCH.

As shown in figure B-1, the display is a standard calculator-type display (NSA1198). The display is driven directly by the DM74175 latches. Decimal point and segment data are sent out over the data bus. Because segment decoders are not used, a lookup table to decode the decimal-point and segment data must be provided by user software. These data are latched when the keyboard interface is selected and the write strobe (WDS*) is low; at the same time, the display digit is selected by AB00 through AB07. While writing into the display, no more than one address bit should be high. (Note: to avoid digit flicker, the display must be continuously refreshed.) Larger displays may be used if additional segment and digit drivers are used.

To select the keyboard-display interface, an address decoder or one of the most significant address bits can be used. The CONT, NHOLD, and NRST signals are ORed together; so if the processor is halted or cleared, the display is blanked. Refer to the following flowcharts and sample program for one possible software implementation of this system.


SCAN KEYBOARD
SET SWITCH VALUE = 0
READ IN ALL KEYS
ARE ANY PRESSED
YES
NO
NONE:
WAITING FOR KEY RELEASE
NO
YES
WAIT 5 MS AND READ ALL KEYS
ARE ANY PRESSED
NO
YES
VALID RELEASE RESET MODE
WAITING FOR KEY RELEASE
NO
YES
WAIT ~ 5 MS AND INITIALIZE SWITCH VALUE
LOOP:
ADD 8 TO SWITCH VALUE
LAST ROW
NO
YES
READ IN NEXT ROW
IS ROW = 0
YES
NO
SHIFT:
INCREMENT SWITCH VALUE
SHIFT ROW RIGHT
SHIFTED VALUE = 0
NO
DONE:
YES
SET MODE
TO SWITCH PROCESSING
DISPLA:
GO TO DISPLAY
DISPLAY
READ IN DIGIT POINTER
POINTER = 0
YES
NO
SET POINTER TO NUMBER OF DIGITS
NEXT:
READ NEXT DIGIT FROM MEMORY
STORE IN NEXT LED DIGIT
GO TO SCAN

```
;       THE DISPLAY ROUTINE STORES THE NEXT
;       DECODED DIGIT INTO THE APPROPRIATE
;       SEVEN SEGMENT DISPLAY.
;       THE DIGITS HAVE BEEN DECODED BY THE USER'S
;       PROGRAM AND ARE STORED IN MEMORY LOCATIONS
;       POINTED TO BY P2 AND THE CORRESPONDING
;       OFFSET "DIGSEL".
;
DISPLA: LDI     FPLOC           ; FPLOC IS LOCATION OF PANEL
        XPAH    P3              ; P3 POINTS TO PANEL
        LDI     0
        XPAL    P3              ; CLEAR P3L
        DLD     DIGSEL(P2)      ; DIGSEL IS APPROPRIATE OFFSET
        JNZ     NEXT
        LDI     01              ; SET UP DIGDRV
        ST      DIGDRV(P2)
        LDI     8               ; RESET DIGSEL TO NO. OF DIGITS
        ST      DIGSEL(P2)
NEXT:   XAE                     ; DIGSEL TO E
        LD      -128(P2)        ; READ DECODED DIGIT INTO AC
        XAE                     ; SAVE DIGIT IN E
        LD      DIGDRV(P2)      ; DIGDRV TO AC
        RR                      ; ROTATE INTO NEXT LOCATION
        ST      DIGDRV(P2)      ; SAVE NEW VALUE
        XAE                     ; DIGDRV TO E
        ST      -128(P3)        ; STORE AC INTO NEXT DIGIT
```

```
;
;       THE KEYBOARD SCAN ROUTINE SCANS
;       48 KEYS AS A 8 X 6 MATRIX.
;       THE RESULT IS A BINARY NUMBER
;       STORED IN MEMORY LOCATION "SWITCH"
;
        CCL                     ; CLEAR CARRY
        LDI     0
        ST      SWITCH(P2)      ; CLEAR SWITCH WORD
        LD      03F(P3)         ; READ ALL KEYS
        JZ      NONE            ; IF = 0, NO KEYS ARE PRESSED
        LD      KEYMD(P2)       ; READ IN KEY MODE
        JNZ     DISPLA          ; IF NOT 0, WAITING FOR RELEASE
        DLY     5               ; DEBOUNCE ABOUT 5 MS
        LDI     020             ; INITIALIZE ROW DRIVER
LOOP:   XAE                     ; ROW DRIVER TO E
        LD      -128(P3)        ; READ ROW INTO AC
        JNZ     SHIFT           ; IF NOT ZERO, VALID KEY FOUND
        LD      SWITCH(P2)
        ADI     8               ; INCREMENT SWITCH BY 8
        ST      SWITCH(P2)
        XAE                     ; ROW DRIVER TO AC
        RR                      ; ROTATE FOR NEXT DIGIT
        JZ      DISPLA          ;
        JMP     LOOP
SHIFT:  SR
        JZ      DONE            ; IF = 0, SWITCH IS DECODED
        XAE                     ; SAVE IN E
        ILD     SWITCH(P2)      ; INCREMENT SWITCH VALUE
        XAE                     ; RECALL WORD FOR NEXT SHIFT
        JMP     SHIFT           ; REPEAT PROCESS UNTIL = 0
NONE:   LD      KEYMD(P2)       ; READ IN KEY MODE
        JZ      DISPLA          ; IF 0, NOT WAITING RELEASE
        DLY     5               ; DEBOUNCE RELEASE 5 MS
        LD      03F(P3)         ; READ ALL KEYS
        JNZ     DISPLA          ; IF 0, KEY RELEASE IS VALID
        ST      KEYMD(P2)       ; SET KEY MODE = 0 ON RELEASE
        JMP     DISPLA          ; GO TO DISPLAY ROUTINE
DONE:   LDI     1               ; SET KEY MODE = 1 FOR RELEASE
        ST      KEYMD(P2)       ;    SCAN ON NEXT PASS
```

```
        .PAGE
;
;       THIS IS THE LOOKUP TABLE USED TO GENERATE
;       DISPLAY CODES FOR THE MULTIPLEXED SEVEN
;       SEGMENT DISPLAYS
;
DECODE: .BYTE   03F             ; "0"
        .BYTE   006             ; "1"
        .BYTE   05B             ; "2"
        .BYTE   04F             ; "3"
        .BYTE   066             ; "4"
        .BYTE   06D             ; "5"
        .BYTE   07D             ; "6"
        .BYTE   007             ; "7"
        .BYTE   07F             ; "8"
        .BYTE   06F             ; "9"
        .BYTE   077             ; "A"
        .BYTE   07C             ; "B"
        .BYTE   039             ; "C"
        .BYTE   05E             ; "D"
        .BYTE   079             ; "E"
        .BYTE   071             ; "F"
```

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
(408) 732-5000
After March 8, 1976, call (408) 737-5000
TWX: 910-339-9240

NS Electronics SDN BHD
Batu Berendam
Free Trade Zone
Malacca, Malaysia
Telephone: 5171
Telex: NSELECT 519 MALACCA (c/o Kuala Lumpur)

National Semiconductor GmbH
808 Fuerstenfeldbruck
Industriestrasse 10
West Germany
Telephone: (08141) 1371
Telex: 05-27649

National Semiconductor (UK) Ltd.
Larkfield Industrial Estate
Greenock, Scotland
Telephone: GOUROCK 33251
Telex: 778 632

NS Electronics (PTE) Ltd.
No. 1100 Lower Delta Rd.
Singapore 3
Telephone: 630011
Telex: NATSEMI RS 21402

REGIONAL AND DISTRICT SALES OFFICES

ALABAMA
DIXIE REGIONAL OFFICE
3322 Memorial Parkway, S.W. #67
Huntsville, Alabama 35802
(205) 881-0622
TWX: 810-726-2207

ARIZONA
ROCKY MOUNTAIN REGIONAL OFFICE
7353 Sixth Avenue
Scottsdale, Arizona 85251
(602) 945-8473
TWX: 910-950-1195

CALIFORNIA
*NORTH-WEST REGIONAL OFFICE
1333 Lawrence Expressway, Suite 258
Santa Clara, California 95051
(408) 247-6397
TWX: 910-379-6432

*LOS ANGELES REGIONAL OFFICE
Valley Freeway Center Building
15300 Ventura Boulevard, Suite 305
Sherman Oaks, California 91403
(213) 783-8272
TWX: 910-495-1773

*SOUTHERN CALIFORNIA REGIONAL OFFICE
17452 Irvine Boulevard, Suite M
Tustin, California 92680
(714) 832-8113
TWX: 910-595-1523

DISTRICT OFFICE
8333 Clairemont Mesa Blvd., Suite 213
San Diego, California 92111
(714) 565-8411
TWX: 910-335-1566

CONNECTICUT
DISTRICT OFFICE

MID-ATLANTIC REGIONAL SALES OFFICE
Piersall Bldg., Suites 214-215
Wilton Center
Wilton, Connecticut 06897
(203) 762-0378
TWX: 710-479-3512

FLORIDA
*CARIBBEAN REGIONAL OFFICE
2721 South Bayshore Drive, Suite 121
Miami, Florida 33133
(305) 446-8309
TWX: 810-848-9725

ILLINOIS
*WEST-CENTRAL REGIONAL OFFICE
800 E. Northwest Highway, Suite 203
Mt. Prospect, Illinois 60056
(312) 394-8040
TWX: 910-689-3346

INDIANA
*WEST-CENTRAL REGIONAL OFFICE
P.O. Box 40073
Indianapolis, Indiana 46240
(317) 255-5822
TWX: 810-341-3300

MARYLAND
CAPITAL REGIONAL OFFICE
95 Aquahart Rd., Suite 204
Glen Burnie, Maryland 21061
(301) 760-5220
TWX: 710-867-0508

MASSACHUSETTS
*NORTH-EAST REGIONAL OFFICE
#8 Wallis Ct.
Lexington, Massachusetts 02173
(617) 861-6090
TWX: 710-332-0166

MICHIGAN
REGIONAL OFFICE
27650 Farmington Rd.
Farmington Hills, Michigan 48024
(313) 477-0400
TWX: 810-242-2902

MINNESOTA
DISTRICT OFFICE
8053 Bloomington Freeway, Suite 101
Minneapolis, Minnesota 55420
(612) 888-3060
Telex: 290-766

NEW JERSEY
DISTRICT OFFICE
140 Sylvan Avenue
Englewood Cliffs, New Jersey 07632
(201) 461-5959
TWX: 710-991-9734

AREA OFFICE
14 Commerce Drive
Cranford, New Jersey 07016
(201) 272-3344
TWX: 710-996-5803

DISTRICT OFFICE
304 Haddon Avenue
Haddonfield, New Jersey 08033
(609) 629-5704

NEW YORK
CAN-AM REGIONAL OFFICE
104 Pickard Drive
Syracuse, New York 13211
(315) 455-5858

REGIONAL OFFICE (IBM only)
576 South Road, Rm. 128
Poughkeepsie, New York 12601
(914) 462-2380
TWX: 510-248-0043

OHIO
DISTRICT OFFICE
Financial South Building
5335 Far Hills, Suite 214
Dayton, Ohio 45429
(513) 434-0097
TWX: 810-459-1615

TEXAS
*SOUTH-CENTRAL REGIONAL OFFICE
13773 No. Central Expressway, Suite 1132
Dallas, Texas 75231
(214) 690-4552
TWX: 910-867-4741

WASHINGTON
DISTRICT OFFICE
300 120th Avenue N.E.
Building 2, Suite 205
Bellevue, Washington 98005
(206) 454-4600

INTERNATIONAL SALES OFFICES

AUSTRALIA
*NS ELECTRONICS PTY, LTD.
Cnr. Stud Road & Mountain Highway
Bayswater, Victoria 3153
Telephone: 03-729-6333
Telex: 32096

BELGIUM
NATIONAL SEMICONDUCTOR BELGIUM
789 Ave. Houba de Strooper
1020 Bruxelles
Telephone: 02-478-3400
Telex: 61 007 Natsem B

CANADA
NATIONAL SEMICONDUCTOR
DISTRICT OFFICE
268 Wildcat Road
Downview, Ontario M3J 2N5
(416) 630-5751
TWX: 610-492-1337

DENMARK
NATIONAL SEMICONDUCTOR DENMARK
Nyhavn 69
1051 Copenhagen
Telephone: (1) 153110
Telex: 160 39

BRITAIN
*NATIONAL SEMICONDUCTOR (UK) LTD.
19 Goldington Rd.
Bedford
Telephone: 0234-211262
TWX: 826209

FRANCE
NATIONAL SEMICONDUCTOR FRANCE
Expansion 10000
28 rue de la Redoute
92-260 Fountenay Aux Roses
Telephone: 660.81.40
Telex: NSF 25956F+

GERMANY
*NATIONAL SEMICONDUCTOR GmbH
8000 Munchen 81
Cosimastr. 4/1
Telephone: 089/915027
Telex: 05-22772

HONG KONG
NS ELECTRONICS (HONG KONG) Ltd.
8th Floor Cheung Kong Electronic Bldg.
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Telephone: 3-411241-8
Telex: 73866 NSE HK HX
Cable: NATSEMI

ITALY
*NATIONAL SEMICONDUCTOR SRL
Via Alberto Mario 26
20146 Milano
Telephone: (02) 4 69 28 64/4 69 23 41
Telex: 36-540

JAPAN
*NATIONAL SEMICONDUCTOR JAPAN
Nakazawa Building
1-19 Yotsuya, Shinjuku-Ku 160
Tokyo, Japan
Telephone: 03-359-4571
Telex: J 28592

SWEDEN
*NATIONAL SEMICONDUCTOR SWEDEN
Sikvagen 17
13500 Tyreso-Stockholm
Telephone: 08/7 1204 80
Telex: 112 93

TAIWAN
NS ELECTRONICS (HK) LTD.
TAIWAN LIAISON OFFICE
Rm. B, 3rd Fl., Ching Lin Bldg.
#5-7 Ching Tao E. Road
P.O. Box 68-332 Taipei
Telephone: 3917324-6
Telex: 22837 NSTW
Cable: NSTW TAIPEI

*Microprocessor System Specialist Available

P50M16