EXOS 302

Intelligent Ethernet Controller For VMEbus Systems

Reference Manual

Publication No. 4200090-00 Revision A

EXOS 302

Intelligent Ethernet Controller For VMEbus Systems

Reference Manual

MANUAL REVISION HISTORY

MANUAL REVISION	DATE	SUMMARY OF CHANGES
A	3-25-88	Initial Release EXOS 302 Intelligent Ethernet Controller For VMEbus Systems Reference Manual Publication No. 4200090-00

PREFACE

This manual describes the EXOS 302 Intelligent Ethernet Controller board. It covers information necessary to integrate the EXOS 302 board in a VMEbus-based system. The manual is intended to be used only as a reference manual and does not undertake to explain the product's design philosophy.

The Ethernet and VMEbus standards are described in readily available documents; this manual makes no special effort to explain them.

The documents listed below provide related reference and study material for EXOS 302 board users.

> For optimum accuracy, you should use this manual in conjunction with the Release Notes supplied with the EXOS 302 board.

The EXOS 302 board conforms to the following specifications:

1. *The Ethernet: A Local Area Network: Data Link Layer and Physical Layer Specifications*, DEC, Intel, and Xerox Corporations, 1980.

2. *The Ethernet: A Local Area Network: Data Link Layer and Physical Layer Specifications, Version 2.0*, DEC, Intel, and Xerox Corporations, 1982.

3. *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications (Standard 802.3-1985 and International Standard 8802/3)*, The Institute of Electrical and Electronics Engineers, Inc., 1985.

The EXOS 302 board conforms to the VMEbus specifications, which are described in the following document:

4. *VMEbus Specifications Manual Rev. C.1*, Motorola Inc., and VMEbus International Trade Association (VITA), October 1985.

The EXOS 302 board uses the Intel 82586 LAN Coprocessor for implementation of Ethernet Data Link protocol. This coprocessor is described in the following documents:

5. *LAN Components User's Manual,* Document No. 230814-001, Intel Corp., 1984.

6. *Microcommunications Handbook,* Document No. 231658-002, Intel Corp,. 1987.

The EXOS 302 board supports front-end processing of user-written higher-level protocols, which execute on an 80286 CPU. This CPU is described in the following document:

7. *iAPX 286 Programmer's Reference Manual,* Document No. 210498-001, Intel Corp., 1983.

User-written protocol software must use the onboard, EPROM-resident NX 300 Network Executive. The NX 300 Network Executive is identical to the NX 200 Network Executive which is described in the following document:

8. *NX 200 Network Executive Reference Manual,* Publication No. 4300036-00, Excelan, Inc., 1986.

The following reference describes the C language, which is used for procedural specifications in this manual:

9. Kernighan, B.W. and Ritchie, D.M, *The C Programming Language,* Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

The following reference describes the ISO Open Systems Model:

10. *Reference Model of Open Systems Interconnection*, Document No. ISO/TC97/SC16 N227, International Organization for Standardization (ISO), 1979.

DOCUMENTATION CONVENTIONS

This manual uses the documentation conventions described below.

Numerical values are specified in decimal or hexadecimal bases. The decimal numbers have no prefixes or suffixes. The hexadecimal numbers are postfixed with the letter H. For example, A024H represents a hexadecimal number.

Notes of special importance are indicated by a pointing hand and double lines, as shown to the left of this paragraph.

TABLE OF CONTENTS

Chapter **Page**

Chapter	Page

Appendix

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure

INTRODUCTION

1

INTRODUCTION

1

The EXOS 302 Intelligent Ethernet Controller is a high-performance, front-end communications processor board that connects a VMEbus system to an Ethernet or an IEEE 802.3 local area network.[1] It implements the complete Ethernet Data Link Layer interface, with significant functional extensions, on a single VMEbus double-height board.

The EXOS 302 board is designed to facilitate the implementation of higher level communications protocols on its own processor. The NX 300 kernel provides a real-time, multitasking environment for the implementation of higher level protocols. The NX 300 kernel implements consistent and portable access methods for the Ethernet and host interfaces.

The EXOS 302 board can be used as an intelligent front-end processor or as a link-level controller. When used as a front-end processor, it executes the high-level network protocols on the board, thereby offloading this burden from the host CPU. When used as a link-level controller, it extends the standard Ethernet Data Link interface to the host system. These two modes are described briefly later in this chapter.

Figure 1-1 shows a block diagram of the EXOS 302 board. Architecturally, the EXOS 302 board consists of two loosely coupled elements: a microprocessor-based, protocol-processing engine and an Ethernet Data Link Layer controller. The engine part includes the host interface, an 80286 CPU, RAM, and ROM; the Data Link Controller part includes an 85286 LAN controller and an 8023A serial interface chip. These elements communicate with each other through the onboard RAM.

The microprocessor executes the protocol software, which is downloaded to the EXOS 302 board either from the host or over the network, and the NX 300 firmware. The 82586 LAN Coprocessor implements part of the Data Link layer. The RAM

1. In this manual, the term *Ethernet* refers to both Ethernet and IEEE 802.3.

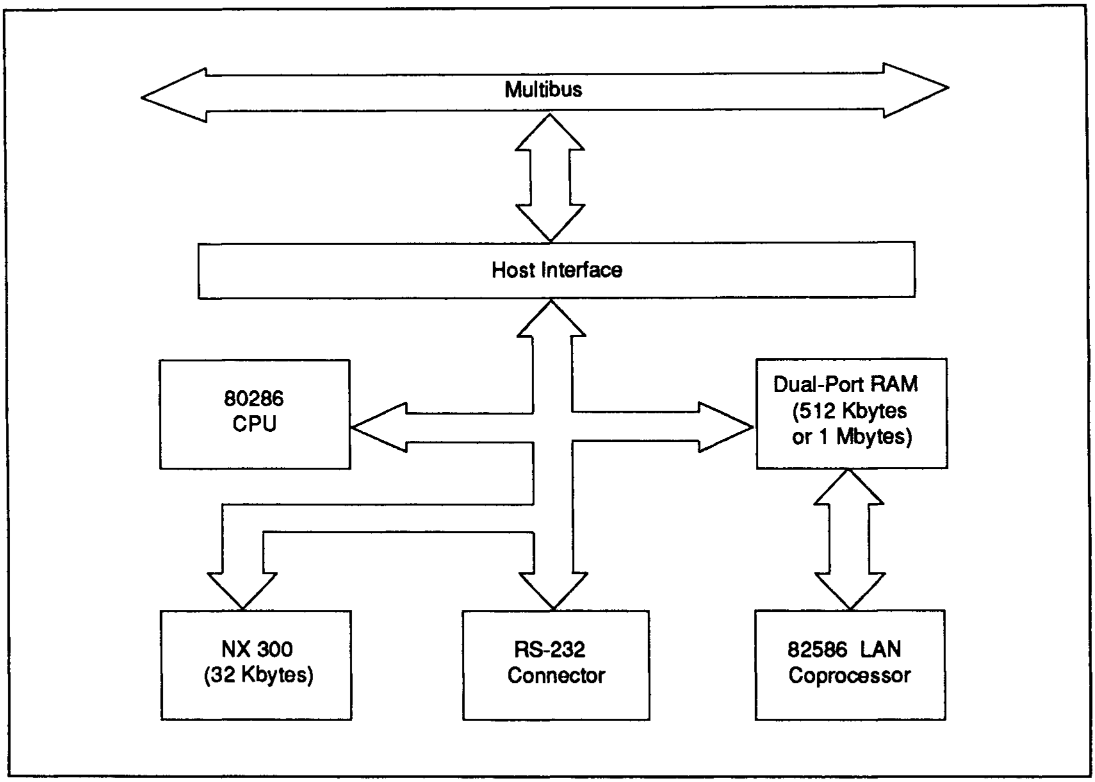


Figure 1-1: EXOS 302 Block Diagram

provides storage for the protocol software and for buffering packets. The NX 300 firmware provides diagnostics, interfaces to host memory and the LAN coprocessor, and operating system environments for execution of the protocol software. The NX 300 firmware also provides link-level controller functions that allow the protocols to reside in the host.

The host system and the EXOS 302 board communicate primarily through command and reply messages located in host memory, which is accessible from the VMEbus. The NX 300 firmware interprets the command messages and generates the replies.

The EXOS 302 board can be configured to be a software-transparent replacement for the EXOS 202 Ethernet Controller. The common interface of NX 300 and NX 200 allows previously written software to be used without modification.

MANUAL ORGANIZATION

This manual is organized as follows:

Chapter 1, Introduction, provides an overview of the EXOS 302 board's features and functions.

Chapter 2, Hardware Reference, describes the architecture of the EXOS 302 board and the functions of various onboard components.

Chapter 3, Installation, discusses how to install the EXOS 302 board in a generic VMEbus system. The chapter also details procedures for installing the EXOS 302 board in an Integrated Solutions VMEbus-based system and in a Motorola VMEbus-based system.

Appendix A, Component Location, provides component and jumper layout for the EXOS 302 board.

Appendix B, Schematic, provides the schematic for the EXOS 302 board.

Appendix C, Service Information, provides guidance on how to obtain technical support and, if necessary, return the EXOS 302 Intelligent Ethernet Controller to Excelan for service.

NX 300 FIRMWARE: THE NETWORK EXECUTIVE

The NX 300 Network Executive, an EPROM-resident set of modules, is an integral part of the EXOS 302 board. It contains board diagnostics, an operating system kernel, interfaces to the host and the Ethernet, and network bootstrap code. When the EXOS 302 board is used in front-end mode, the NX 300 firmware provides the operating system environment for the downloaded protocol software that executes on the board. When the EXOS 302 board is used in link-level mode, the NX 300 firmware provides the Data Link Controller functions for the protocol software that executes on the host system.

The NX 300 firmware resides in EPROM, which appears at the high end of the 1 Mbyte address space of the 80286 microprocessor. The NX 300 firmware data structures use 4 Kbytes of the RAM; the rest is available for higher level software and packet buffering.

Figure 1-2 provides a graphic representation of the NX 300 software architecture.

For users of prepackaged software, such as the Excelan EXOS Series 8000 TCP/IP Network Software, the existence of the NX 300 firmware is transparent. Consequently, they need not concern themselves with the internals of the NX 300 firmware.

For users who plan to write their own protocol software or link-level drivers, it is necessary to understand the NX 300 internals. A summary of NX 300 features is given in the following section. Refer to the *NX 200 Network Executive Reference Manual* for further details.

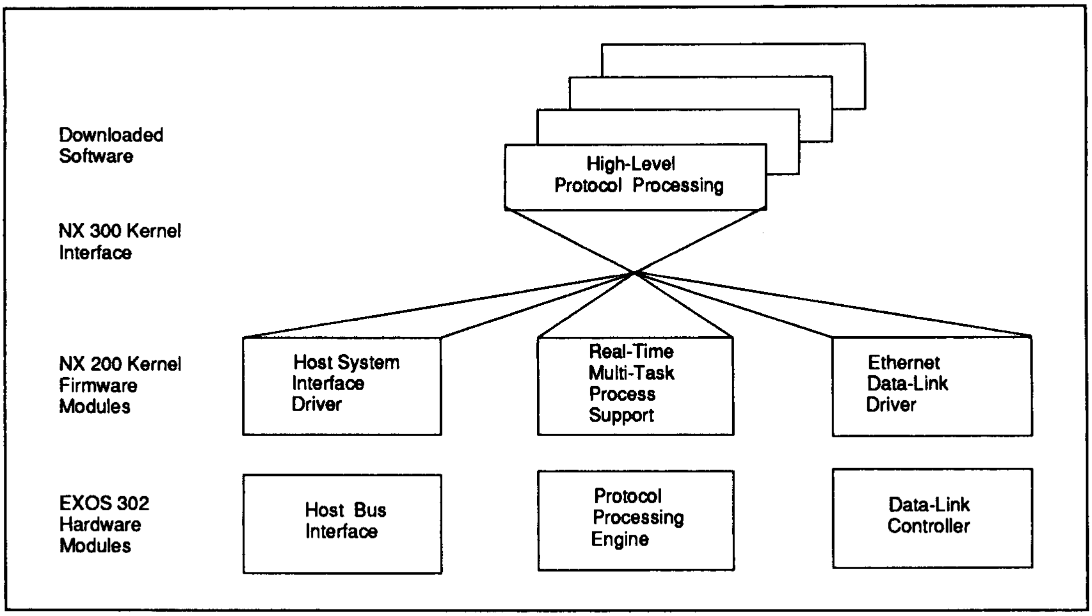


Figure 1-2: NX 300 Software Architecture

Principal Features of NX 300

The following is a list of the main features of the NX 300 firmware:

- Self-diagnostics for testing the integrity of the EXOS 302 hardware.
- Booting process that allows higher level software to be downloaded from either the host or the network.
- A real-time kernel that provides a multitasking environment, enabling the protocol software to be constructed in a structured manner as a set of cooperating processes.
- Device drivers for the Ethernet controller and host computer interface. Access through message queues simplifies pipelined communications.
- Supports network management functions by collecting network statistics.
- In link-level mode, allows the EXOS 302 board to be used as a Data Link controller, giving access to the network at the Data Link layer, without downloading any software.

The EXOS 302 Board in Front-End Mode

In front-end processor mode, the host system downloads protocol software to the EXOS 302 board when the host is initialized (or the EXOS 302 bootstraps itself from the Ethernet network). This software then uses the NX 300 real-time, multitasking process management services and I/O drivers to control the Ethernet interface and to manage communications with the host system.

Standard protocol modules for the EXOS 302 board, such as the DARPA TCP/IP protocols, are available from Excelan. Figure 1-3 shows the relationship between these modules and the Open Systems Interconnection (OSI) reference model of the International Standards Organization (ISO).

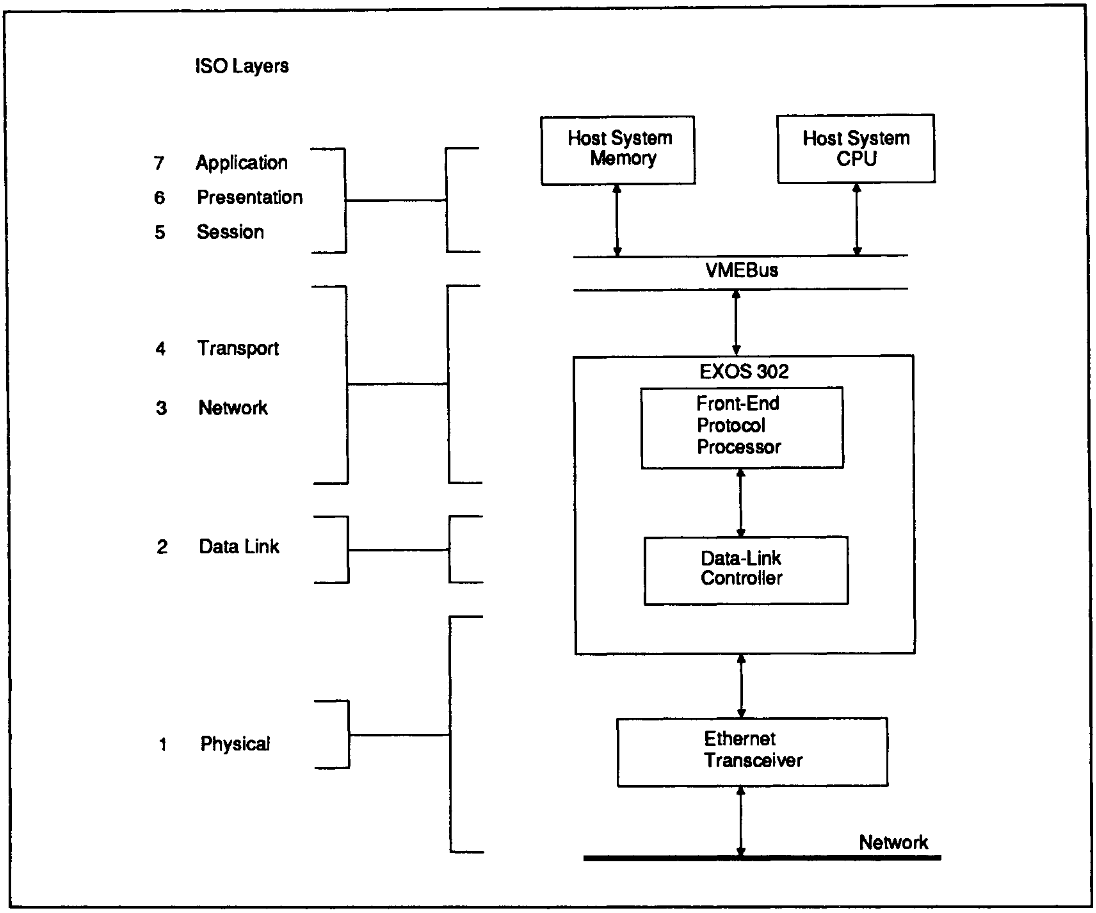


Figure 1-3: EXOS 302 Board In Front-End Mode

The EXOS 302 Board in Link-Level Mode

In link-level mode, the NX 300 firmware extends the EXOS 302 board's Data Link controller functions out to the host interface. The host system obtains Data Link services through standard request and reply messages. In this mode, the EXOS 302 RAM is entirely available for buffering packets. Link-level controller mode is useful for applications in which host-resident protocol software has already been developed, or in which it is otherwise not feasible to download high-level protocols to run on the EXOS 302 board.

Programming information required to write I/O drivers to interface host-resident protocol software to the NX 300 Data Link functions is detailed in the *NX 200 Network Executive Reference Manual.*

Initialization and Self-Test

On reset, the NX 300 firmware performs a series of self-tests that confirm the integrity of the hardware. In case of failure, the firmware communicates diagnostic codes through an LED display and a host status port. After successful completion of the tests, the EXOS 302 board either boots itself from the Ethernet network or awaits initialization by the host system, depending on the jumper option selected on the board.

If the jumper selects initialization by a host system, the host then sends a configuration message that selects the NX 300 mode of operation and specifies several other parameters. The host can request the NX 300 firmware to download software from the host, to download software from the network, or to configure the board in link-level mode. If initialization includes downloading software, then the NX 300 firmware enters the front-end processor mode of operation.

EXOS 302 HARDWARE

The EXOS 302 Intelligent Ethernet Controller is designed around three major onboard hardware components and one firmware component. The hardware components are an Intel 80286 microprocessor, an Intel 82586 LAN Coprocessor, and dual-port RAM (512 or 1024 Kbytes). The firmware component is the EPROM-resident NX 300 Network Executive, also referred to as the NX 300 firmware.

The EXOS 302 board implements the Ethernet Data Link protocol using the Intel 82586 LAN Coprocessor. Functions such as address recognition, CRC check, and buffer chaining are managed by the coprocessor, leaving the onboard 80286 microprocessor available for protocol-processing applications. The protocol processing engine is supported by 512 Kbytes of RAM. (A 1 Mbyte option is not currently supported by the NX 300 firmware.) Two 16-Kbyte EPROMs contain the Excelan NX 300 firmware, which includes self-diagnostic tests, an operating system kernel, host and network interfaces, and network bootstrap code.

Principal Hardware Features

The following are the salient features of the EXOS 302 board:

- Single double-height (6.3" by 9.2") VMEbus board, which is available with either a panel or ejectors for use in different card cage types.
- Onboard Intel 80286 microprocessor and 512 Kbytes of RAM support high-level network protocols on board.
- Dual-port memory allows concurrent, full-speed access by the onboard microprocessor and LAN coprocessor.
- Successive frames can be received or transmitted with minimum interframe spacing (9.6 microseconds). Frames can be received immediately after transmitting, or vice versa, with minimum interframe spacing and without losing data.
- Uses hardware logic for recognition of physical, broadcast, and multiple multicast addresses, in addition to promiscuous mode.
- Hardware-supported buffer chaining allows buffering of an arbitrary number of received frames without any CPU intervention. Allocation of buffers, both location and size, is completely under software control.

VMEbus Compatibility

The EXOS 302 board conforms with VMEbus specifications by Motorola and the VME International Trade Association (VITA), Revision C.1, for electrical, mechanical and timing parameters. IEEE P1024 compliance is 8-bit and/or 16-bit transfers, 24-bit or 32-bit addressing, and 8-bit bus-vectored interrupts.

VMEbus conformity, depending on jumper configuration, can also be stated as follows:

A24 MASTER, D(EO), D16; A24 SLAVE, D(O)
or A32 MASTER, D(EO), D16; A24 SLAVE, D(O)
or A32 MASTER, D(EO), D16; A32 SLAVE, D(O)

VMEbus Interface

The EXOS 302 board acts as a bus master to transfer data between its memory and the system memory over the host bus. It provides 32 address lines (or 24, depending upon mode) to the host, for a total addressable memory space of 4 gigabytes. The host has no access to the EXOS 302 memory.

The EXOS 302 board can be jumpered to operate as an A24 MASTER, which makes it compatible with the EXOS 202 host driver. The difference between A24 and A32 mode, besides the number of address bits, is the source of the six address-modifier bits. In A24 mode the address modifiers are derived from the low-order 6 bits of the upper byte of the mapping register (see Figure 2-1). This value is passed to the board as the upper byte of a 32-bit address. However, in A32 mode, all bits are used as the address. A separate register is provided for the address modifier, which must be written by the host.

A 1-byte communication path is provided from the VMEbus to the EXOS 302 board via a slave port. This is used, in conjunction with interrupts, during initialization to transmit the address of a communication area in the shared VMEbus memory.

The host and the EXOS 302 board can interrupt each other. The 302 generates a vectored interrupt to interrupt the host. Interrupt priority can be set from level 1 to level 7.

Ethernet Compatibility

The EXOS 302 board is fully compliant with the IEEE Standard 802.3, as well as with Ethernet specifications Versions 1.0 and 2.0. Used with a IEEE Standard 802.3 or Ethernet transceiver, the EXOS 302 board provides all Data Link and Physical Layer services.

INSTALLATION

Installing the EXOS 302 board is a relatively simple task. In a typical VMEbus system, it consists of turning off the power, accessing the card cage, inserting the board, and establishing the required connection through appropriate cables and connectors. In general, the complexity of the installation procedure depends on the physical design of the host machine rather than on the design of the EXOS 302 board.

Chapter 3, Installation, provides the step-by-step procedure for installing the EXOS 302 board.

HARDWARE REFERENCE

2

HARDWARE REFERENCE 2

Most of the hardware-dependent aspects of the EXOS 302 board's implementation are hidden by the NX 300 firmware. Therefore, this chapter discusses only those aspects that are "visible" to and are of concern to most users. These include issues related to VMEbus, Ethernet, and jumper-selectable options. Refer to the schematic included in Appendix B for further understanding of the board functions. The schematic is included for reference only.

VMEbus-related issues discussed in this chapter include bus interface, compliance, memory access, slave access, interrupt configuration, and bus priority resolution.

Ethernet-related issues discussed include Ethernet interface, compliance, functions, address recognition, operation timing, packet buffering, error handling, and network connections.

GENERAL DESCRIPTION

The EXOS 302 board is a microprocessor-controlled adapter that provides an Ethernet connection for computers having a VMEbus. Architecturally, it is compatible with the NX 300 Network Executive.

The EXOS 302 board operates as a bus master, capable of reading and writing the host memory. Coordination between the host and the 302 is accomplished by the establishment of control blocks in host memory, the passing of values via special registers, and interrupts. The EXOS 302 board operates as a slave to a bus master reading or writing its control registers.

The following are the major features of the EXOS 302 board:

- Intel 80286 microprocessor, 8 MHz
- Intel 82586 Ethernet controller
- IEEE 802.3 compliance 10Base5
- Ethernet Version 1.0 and Version 2.0 compatibility
- D-type 15-pin transceiver connector
- 512 KBytes or 1 MByte local memory
- 32 KByte or 64 KByte local PROM for 80286 firmware
- Intel 8259A interrupt controller
- Dual UART for RS232 port and timer
- Host bus interface:
  - 8-bit or 16-bit data
  - 24-bit or 32-bit address
  - Daisy-chained bus request/grant
  - Daisy-chain interrupt structure
- EXOS 202 compatibility mode (24-bit address)

VMEbus INTERFACE

The EXOS 302 Intelligent Ethernet Controller is built on a single double-height (6.3" by 9.2") VMEbus board, with standard VMEbus connectors. The board installs into any standard VMEbus system card cage. The board is available with a front panel or ejectors for installation in different card cage types. Refer to Chapter 3 for information about the installation of the EXOS 302 board.

VMEbus Compliance

The EXOS 302 board conforms with VMEbus specifications by Motorola and the VMEbus International Trade Association (VITA), Revision C.1. IEEE P1024 compliance is 8-bit and/or 16-bit transfers, 24-bit or 32-bit addressing, and 8-bit bus-vectored interrupts. EXOS 302 VMEbus conformity is as follows:

A24 MASTER, D(EO), D16; A24 SLAVE, D(O)
or A32 MASTER, D(EO), D16; A24 SLAVE, D(O)
or A32 MASTER, D(EO), D16; A32 SLAVE, D(O)

The bus arbitration interrupts are defined as follows:

4-level Bus Arbitration Requester
7-level Bus Interrupt Requester
Bus-vectored interrupts (software programmable vector)

VMEbus Host Memory Access

The EXOS 302 board acts as a bus master to transfer data between its memory and the system memory over the host bus. The 302 provides 32 address lines to the host, for a total addressable memory space of 4 gigabytes. However, only 128 Kbytes are addressable at a time. The lower 16 bits from the EXOS 302 microprocessor are concatenated with 16 bits from a mapping register to form the full 32-bit address. The value for the mapping register is passed to the NX 300 firmware by the host as the upper half of a 32-bit address. Address line 16 from the 80286 selects which of two 16-bit registers is used (see Figure 2-1). This provides two 64-Kbyte windows, each of which can be placed on any 64-Kbyte boundary in the host's address space.

In the A32 MASTER mode the six VMEbus address modifier bits are derived from a dedicated address modifier register, which is written directly by the host.

The EXOS 302 board can be jumpered to operate as an A24 MASTER, which makes it compatible with the EXOS 202 host driver and address-modifier scheme. The difference betwwen A24 and A32 mode, besides the number of address bits, is the source of the six address-modifier bits. In A24 mode the bits are derived from the low-order 6 bits of the upper byte of the mapping register, the address modifier having been passed by the host as the high byte of the map register value.

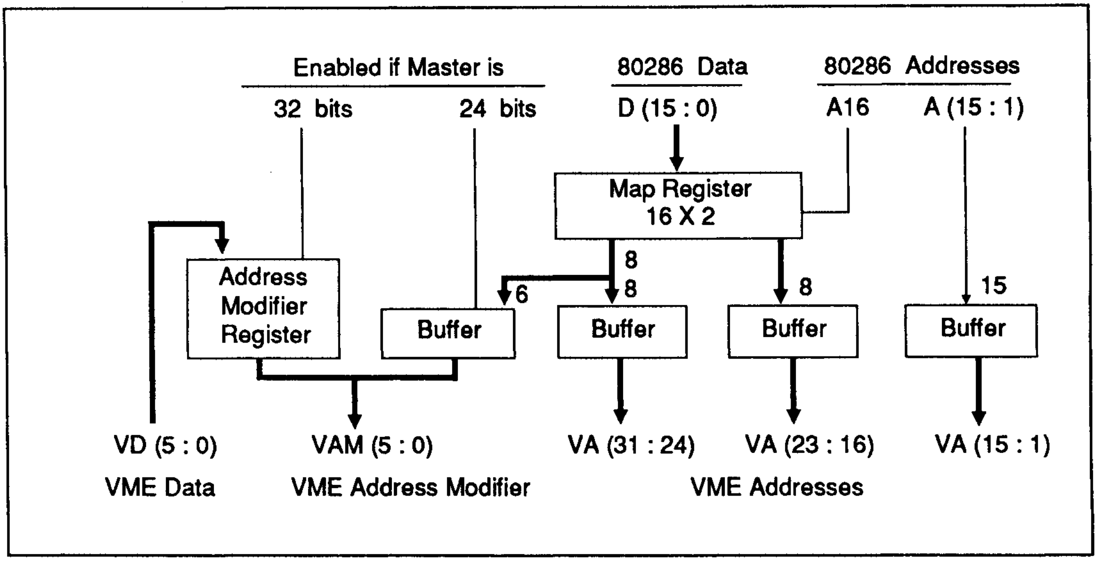


Figure 2-1: EXOS 302 Bus Master Address Generation

The EXOS 302 board is hardware-configured as either an A32 or A24 MASTER. If configured as A32, the address modifier is driven by the addesss modifier register. A24 memories can still be addressed by an A32 MASTER by changing the address modifier register to one valid for 24-bit devices. The upper byte of the map register is not the address modifier source in this case, since the map register is driving 16 bits of address.

In A24 MASTER mode the address modifier from the map register is also driven onto the upper address byte (bits 31-24). According to the VME specification, if the address modifier indicates a 24-bit address, devices must not decode the upper byte.

The EXOS 302 board can do 8-bit or 16-bit data operations. Bytes are automatically swapped by the hardware, since it is assumed that any memory device will present data to the bus in 68000 format. This does not solve byte-order problems between dissimilar hosts on the network. Byte-order problems must be solved at the application level between the two hosts.

The EXOS 302 memory is not accessible from the VMEbus. All data transfers between the EXOS 302 local memory and the host memory are done by the EXOS 302 board, with only a pointer to a control block in the host memory supplied by the host via the 1-byte slave data port.

VMEbus Bus Request Priority Resolution

The EXOS 302 board is normally an Option RWD (Release-When-Done) Requestor. As a bus master, the 302 requests the bus, then releases it at the completion of each bus cycle.

Jumpers J57 through J68 determine the VMEbus Bus Request Priority Level of the EXOS 302 board. Two jumpers connect the grant-in and grant-out signals for the selected level. The unused grants-in are jumpered to their respective grants-out, thereby propagating the other daisy chains. Table 2-3 shows the bus priority resolution configuration.

The 302 hardware monitors the ACFAIL and SYSFAIL signals on the VMEbus. While either of these signals is asserted, the EXOS 302 board is inhibited from making requests on the bus, thereby allowing the system to operate unhindered for shutdown or testing. If the 80286 attempts to access the host during this time, it is held in a wait state.

VMEbus Bus Burst Mode

The EXOS 302 board has a burst mode, which is a modified ROR (Release-On-Request) Requester. It is designed to increase the transfer rate of long string move operations. The bus interface holds the bus as long as the EXOS 302 microprocessor continues requesting the bus, to a limit of 15 microseconds. Then the bus is released, and a new arbitration must occur for the EXOS 302 board to reacquire the bus. Because the bus is released regularly, giving other masters an opportunity for the bus, the bus clear and the other bus request signals are not monitored. This function is enabled by jumper J25.

Normally, burst mode is not effective. However, under certain conditions burst mode may increase performance. These conditions, either singly or in combination, may be the following:

- The bus arbiter has a request-to-grant latency greater than 200 nanoseconds.
- The EXOS 302 board is not at the beginning of the bus grant daisy-chain.

- The EXOS 302 board is sharing a grant level with several other devices.

Burst mode is not a VMEbus block transfer. The 302 provides the address for each data cycle.

VMEbus Interrupts

The EXOS 302 board can assert bus-vectored interrupts on the VMEbus. It is an ROA (Release-On-Acknowledge) INTERRUPTER. Jumpers J47 through J56 select the interrupt priority in the range 1 to 7. Only one interrupt request level should be selected. One jumper connects the interrupt request to the bus. Three jumpers encode the interrupt level for comparison during the interrupt acknowledge cycle. The interrupt acknowledge level must match the interrupt request level as shown in Table 2-4. The factory setting is for interrupt level 5.

The NX 300 firmware can be initialized to generate interrupts to a host in three other ways: polling of the message queues in the host's memory, writing to a dedicated memory-mapped location in a host's space, and writing to a dedicated I/O location in a host's space (not applicable to the VMEbus). These methods do not use the hardware interrupt mechanism. Refer to the chapter Initialization and Host Interface, in the *NX 200 Network Executive Reference Manual* for further details.

VMEbus Slave (I/O) Access

Three ports exist on the EXOS 302 board in the host memory-mapped I/O space (see Table 2-1). Two are for status and control information to and from the EXOS 302 board. The third is for the host to write the 6-bit address modifier register. They are the host's only direct access to the EXOS 302 board. These ports must be accessed as byte values; if not, a system bus timeout, if enabled, occurs.

The base address of these registers is selected by jumpers. There are sixteen compare jumpers (J26-J41) for individual address bits 31-16, and two jumpers (J42 and J43) to define one of four

Table 2-1: Slave Ports

Port	Port Number	Type	Function
A	1	Read	Read Status Byte 1 (extended status)
A	1	Write	EXOS 302 Reset
B	3	Read	Read Status Byte 0
B	3	Write	Host Interrupt/Data Write
C	5	Write	Write Address Modifier

combinations for bits 15-7. These four combinations are programmed into a PAL and currently are compatible with Revision F of the EXOS 202 board. (See Table 2-2.)

The slave address configuration is shown below, in a program representation of the address. Bit 0 does not appear on the VMEbus. Bits 31-24 are not decoded when in A24 SLAVE mode. Bits 6-3 are not decoded, occupying a 128-byte block above the base address. Complete jumper configurations are shown in Table 2-2. The base address is set to FF0000H (A24 SLAVE) at the factory.

```
31      24     23      16     15      8      7      0
bbbb    bbbb | bbbb    bbbb | bbbb    bbbb | b---   -ppp
```

"b" signifies "base." "p" signifies "port number." Ports A, B, and C respond to address modifiers 39H, 3AH, 3DH, and 3EH in A24 SLAVE mode and to 09H, 0AH, 0DH, and 0EH in A32 SLAVE mode. These modifiers are programmed into a PAL.

Note that the port numbers (ppp) may need to be changed if the host CPU does not use Motorola data format. For the EXOS 302 I/O ports, DS0 is the operative data strobe, and data is connected to VME data bits 0-7.

The effects of reading and writing ports A, B, and C are summarized below:

Read A (ppp=001):

> Read the EXOS 302 Status Byte 1 (extended status). All bits are registered bits and can be set by the EXOS 302 firmware. Contents of this register are indeterminate after reset.

Write A (ppp=001):

> Reset the EXOS 302 board. The NX 300 firmware does a self-test, then waits for the host to pass the configuration address.

Read B (ppp=011):

Read EXOS 302 Status Byte 0. All bits are registered bits, and can be set by the EXOS 302 firmware, except bits 1 and 3, which are interrupt status bits. All bits are cleared by reset.

Bit 0: (Error Bit) When 0, indicates a fatal error in the EXOS 302 board. After the board is reset, this bit is 0 but is set to 1 if the self-test completes successfully. If this bit is not set within 10 seconds, the board has failed the self-test.

Bit 1: (Host Interrupt Status) This bit is set by the hardware when the EXOS 302 board issues an interrupt to the host. It is cleared by the hardware during the interrupt acknowledge sequence.

Bit 3: (Ready Bit) When 0, indicates that the EXOS 302 board is ready to accept a byte written to port B. When 1, the EXOS 302 board has not yet read the byte last written to port B.

Bit 6: (Loopback Status) When set, indicates a failure of the network loopback test. This is a warning only and does not indicate that the network is not functional.

Bits 2, 4, 5, 7:
Undefined.

Write B (ppp=011):

Interrupt the EXOS 302 board's CPU and communicate a 1-byte value. It also sets the Host Interrupt Status Bit in port B. This bit is cleared when the 80286 reads the value from the Host Data Register. This is the only way for the host to communicate a value to the EXOS 302 board other than through shared memory.

Write C (ppp=101):

A byte is written into the address modifier register in the EXOS 302 board. Only the low 6 bits are used. This register drives the host address modifier bits during a bus cycle if the 302 is configured as an A32 master.

ETHERNET INTERFACE

The EXOS 302 board, when connected to a network via a standard Ethernet transceiver, performs all standard Ethernet Physical and Link layer functions.

Ethernet Compliance

The EXOS 302 board conforms fully to the IEEE 802.3 standard as well as to Ethernet specifications, Versions 1.0 and 2.0.

As shipped from the factory, the EXOS 302 board is configured for use with Ethernet Version 1.0. To configure for use with IEEE 802.3 and Version 2.0, jumper J3 must be removed.

Ethernet Functions

The EXOS 302 board performs all Physical and Link Layer Ethernet functions, except for transceiver functions. These include the following:

- Serial-to-parallel and parallel-to-serial conversions
- Address recognition
- Framing and unframing of messages
- Manchester encoding and decoding
- Preamble generation and removal
- Carrier sense and deference
- Collision detection and enforcement, including jamming, backoff timing, and retry
- FCS (CRC) generation and verification
- Serial-to-parallel and parallel-to-serial conversion
- Physical and multicast address recognition
- Alignment and length error detection and handling

For a complete description of the Ethernet functions implemented by the 82586 LAN Coprocessor, refer to the Intel Microcommunications Handbook listed in the preface of this manual.

Ethernet Address Recognition

Each EXOS 302 board has a unique 48-bit Ethernet address, which is stored in an address PROM at component location U50. The Ethernet address for each EXOS 302 board is provided on a white sticker placed on top of the address PROM. This is the board's default physical address. However, the effective physical address resides in RAM and can be modified by user software, so that the host software can override the address stored in the PROM when it begins running.

Recognition of physical, broadcast, and multicast addresses is fully supported, without user software intervention. An efficient multicast address filter, implemented in hardware, greatly reduces the overhead of multicast address recognition. As many as 252 multicast addresses can be assigned to a station. The LAN coprocessor also provides a promiscuous mode, in which the multicast address filter can be disabled, so that it accepts all multicast addresses.

Ethernet Frame Formatting

Link-level frames are formatted according to the following Ethernet specification:

- Preamble (64 bits of synchronizing sequence)
- Destination address (48 bits)
- Source address (48 bits)
- Message type (16 bits)
- Data (46 to 1500 bytes)
- FCS (32 bits)

The LAN Coprocessor generates and removes the preamble. It also generates and checks the Frame Check Sequence (FCS).

Ethernet Interframe Spacing

The EXOS 302 board can receive successive frames with the minimum interframe spacing (9.6 microseconds) without losing data. It can also receive immediately after transmitting, and vice versa, with the minimum interframe spacing, without losing data.

Ethernet Packet Buffering

In front-end mode and under NX 300 firmware control, the EXOS 302 board can buffer an arbitrary number of both receive and transmit packets. The actual number of available buffers depends on the application.

In link-level mode, the EXOS 302 board can chain up to 32 receive packet buffers, and can receive as many packets, without host intervention. Transmit packets are chained by the NX 300 firmware and transmitted with minimal delay.

Ethernet Error Handling

The EXOS 302 board handles all Ethernet error conditions, including CRC, alignment, and length errors.

The EXOS 302 board can be enabled, at the host's direction, to receive packets that would normally be rejected due to CRC and alignment errors.

INDICATORS (LED)

The EXOS 302 board has three Light Emitting Diodes (LEDs) that communicate status information. The LEDs are located in adjacent positions along the outer edge of the board; they can easily be seen while the board is installed (if unobstructed). Figure 2-2 shows their relative locations and functions.

The following LEDs are located on the EXOS 302 board:

- **NX 300 Status LED.** The NX 300 Status LED, at position DS1, flashes operational status and error codes. Refer to the section Errors, later in this chapter, for a description of how to interpret these codes. The NX 300 firmware normally leaves this LED off after initialization.

- **Ethernet Transmit Status LED.** The EXOS 302 board lights the Ethernet Transmit Status LED, at position DS2, while transmitting on the Ethernet network. It is derived from the Request to Send output of the 82586. Short or infrequent packets may not light the LED enough to be easily seen.

- **VMEbus Cycle Status LED.** The VMEbus Cycle Status LED, in position DS3 on the EXOS 302 board, indicates that a VMEbus cycle is in progress. It goes on when the bus is requested, and stays on while the cycle is in progress. It goes

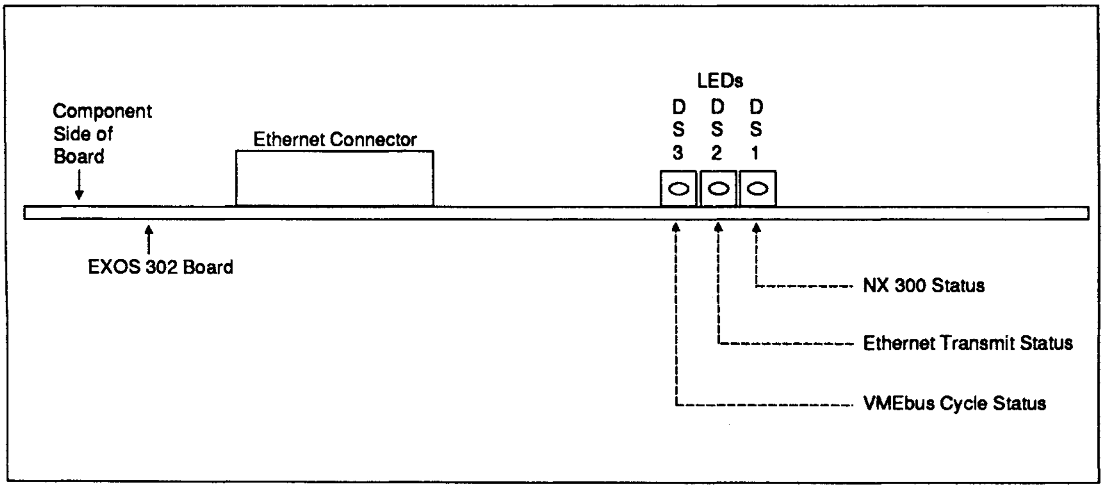


Figure 2-2: EXOS 302 Board (Side View) Showing Status LEDs

off when no cycle is in progress, even if the 302 is holding the bus (burst mode). If the LED is lit steadily and the system appears to have hung, the two most likely causes are the following:

- The board has a request pending, but has not been granted the bus.
- The board has attempted to access a non-existent memory address on the VMEbus.

The first condition points toward a bus request/grant configuration problem with the board or the VMEbus backplane. The second condition points toward a user software bug. (The second condition can occur only if the host's bus timeout feature is not enabled.)

TRANSCEIVER FUSE

A 1-amp fuse is inserted in the line which provides +12 VDC to the transceiver connector P3 and the (jumpered) backplane Ethernet connection via P2. The NX 300 firmware can determine if this fuse is blown; if so, it sends a message to the host. This fuse is user-replaceable. Refer to Appendix A for the fuse location (FS1).

JUMPER SETTING OPTIONS

Jumpers J2-J16 select NX 300 firmware options.

It may be necessary to install jumper J2 to disable the Carrier Sense function when a broadband transceiver is used. This requirement is indicated by the return code 20H, which is described in the *NX 200 Network Executive Reference Manual.*

Jumper J3, when installed (default), disables the SQE check that otherwise is performed after each transmission. This provides an Ethernet Version 1.0 compatible transceiver connection. If an Ethernet Version 2.0 or IEEE 802.3 transceiver is to be used, jumper J3 should be removed.

If J4 is installed, the EXOS 302 board attempts to download software from the Ethernet after self-testing is complete. If J4 is not installed, the EXOS 302 board awaits initialization from the host after self-testing is complete.

Table 2-2 lists the function of the jumpers on the EXOS 302 board. Appendix A shows the location of components and jumpers on the EXOS 302 board.

Table 2-2: EXOS 302 Jumpers

Jumper	Function	If Absent	If Installed	Factory Setting
J1	Uncut (factory use only)			Uncut
J2	CRS Check Flag	Enable	Disable	Absent
J3	Version 1.0 (SQE)	Enable	Disable	Installed
J4	Non standard console	Enable	Disable	Absent
J5	Reserved			Absent
J6	Reserved			Absent
J7	Reserved			Absent
J8	Reserved			Absent
J9	Debug option			Absent
J10	Memory Size	512KB	1MB	Absent
J11	Reserved			Absent
J12	Reserved			Absent
J13	Reserved			Absent
J14	EPROM Wait state	Disable	Enable	Absent
J15	EPROM size jumper 1			Installed 1-2
J16	EPROM size jumper 2			Installed 1-2

J15 and J16 enable EPROMS as follows:

TYPE	J16	J15
27128	1-2	1-2
27256	1-2	2-3
27512	2-3	2-3

Jumper	Function	If Absent	If Installed	Factory Setting
J17	E+12V (P2-C16) *	Open	Connect	Absent
J18	TRMT+ (P2-A13) *	Open	Connect	Absent
J19	TRMT- (P2-A14) *	Open	Connect	Absent
J20	RECV+ (P2-A15) *	Open	Connect	Absent
J21	RECV- (P2-A16) *	Open	Connect	Absent
J22	CLSN+ (P2-C13) *	Open	Connect	Absent
J23	CLSN- (P2-C14) *	Open	Connect	Absent
J24	GND (P2-C15) *	Open	Connect	Absent
J25	VME master burst transfer			Installed 2-3

J25 selects burst mode as follows:

Enable	Disable
1-2	2-3

Jumper	Function	If Absent	If Installed	Factory Setting
J26	VME slave address bit 31 comparator	= 1	= 0	Absent
J27	VME slave address bit 30 comparator	= 1	= 0	Absent
J28	VME slave address bit 29 comparator	= 1	= 0	Absent
J29	VME slave address bit 28 comparator	= 1	= 0	Absent
J30	VME slave address bit 27 comparator	= 1	= 0	Absent
J31	VME slave address bit 26 comparator	= 1	= 0	Absent
J32	VME slave address bit 25 comparator	= 1	= 0	Absent
J33	VME slave address bit 24 comparator	= 1	= 0	Absent
J34	VME slave address bit 23 comparator	= 1	= 0	Absent
J35	VME slave address bit 22 comparator	= 1	= 0	Absent
J36	VME slave address bit 21 comparator	= 1	= 0	Absent
J37	VME slave address bit 20 comparator	= 1	= 0	Absent

* These jumpers connect the associated signals on the indicated pins of connector P2.

Table 2-2: EXOS 302 Jumpers (Continued)

Jumper	Function	If Absent	If Installed	Factory Setting
J38	VME slave address bit 19 comparator	= 1	= 0	Absent
J39	VME slave address bit 18 comparator	= 1	= 0	Absent
J40	VME slave address bit 17 comparator	= 1	= 0	Absent
J41	VME slave address bit 16 comparator	= 1	= 0	Absent
J42	VME slave address bits 15-0 selector 1			Absent
J43	VME slave address bits 15-0 selector 0			Absent

J42 and J43 select the low 16 bits of slave address as follows:

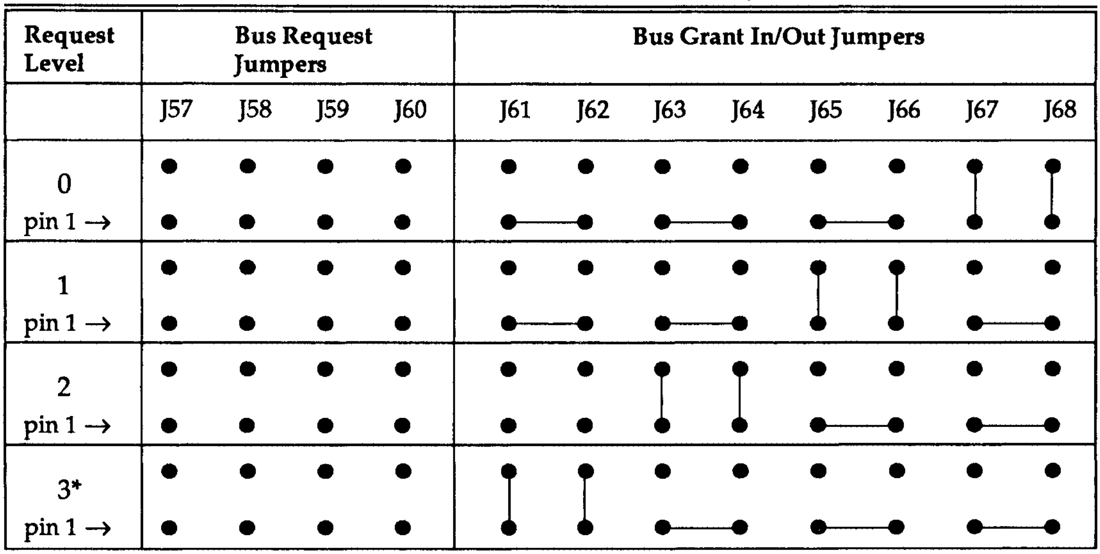
Base	J42	J43
0000	Absent	Absent
7F80	Absent	Installed
8000	Installed	Absent
FF80	Installed	Installed

Jumper	Function	If Absent	If Installed	Factory Setting
J44	VME master address width	32 bits	Follows J46	Installed
J45	Factory test only			Absent
J46	VME slave address width	32 bits	24 bits	Installed
J47	VME interrupt request 1 **	Disable	Enable	Absent
J48	VME interrupt request 2 **	Disable	Enable	Absent
J49	VME interrupt request 3 **	Disable	Enable	Absent
J50	VME interrupt request 4 **	Disable	Enable	Absent
J51	VME interrupt request 5 **	Disable	Enable	Installed
J52	VME interrupt request 6 **	Disable	Enable	Absent
J53	VME interrupt request 7 **	Disable	Enable	Absent
J54	VME interrupt acknowledge bit 1 **			Installed
J55	VME interrupt acknowledge bit 2 **			Absent
J56	VME interrupt acknowledge bit 3 **			Installed
J57	VME Bus request level 3 ***			Installed
J58	VME Bus request level 2 ***			Absent
J59	VME Bus request level 1 ***			Absent
J60	VME Bus request level 0 ***			Absent
J61	VME Bus grant 3 out ***			Installed
J62	VME Bus grant 3 in ***			Installed
J63	VME Bus grant 2 out ***			Installed J63-1 to J64-1
J64	VME Bus grant 2 in ***			Installed J64-1 to J63-1
J65	VME Bus grant 1 out ***			Installed J65-1 to J66-1
J66	VME Bus grant 1 in ***			Installed J66-1 to J65-1
J67	VME Bus grant 0 out ***			Installed J67-1 to J68-1
J68	VME Bus grant 0 in ***			Installed J68-1 to J67-1

** Interrupt request and acknowledge jumpers are interrelated, as shown in Table 2-4.

*** Allowable configurations for the bus request jumpers J57-J60 and the bus grant jumpers J61-J68 are interrelated, as shown in Table 2-3.

Table 2-3: Bus Priority Resolution Jumpers



*The factory setting is level 3.

Table 2-4: Interrupt Priority Configuration

Request Level	Interrupt Request Jumpers							Acknowledge Jumpers		
	J47	J48	J49	J50	J51	J52	J53	J54	J55	J56
1	●–●	● ●	● ●	● ●	● ●	● ●	● ●	● ●	● ●	●–●
2	● ●	●–●	● ●	● ●	● ●	● ●	● ●	● ●	●–●	● ●
3	● ●	● ●	●–●	● ●	● ●	● ●	● ●	● ●	●–●	●–●
4	● ●	● ●	● ●	●–●	● ●	● ●	● ●	●–●	● ●	● ●
5*	● ●	● ●	● ●	● ●	●–●	● ●	● ●	●–●	● ●	●–●
6	● ●	● ●	● ●	● ●	● ●	●–●	● ●	●–●	●–●	● ●
7	● ●	● ●	● ●	● ●	● ●	● ●	●–●	●–●	●–●	●–●

*The factory setting is level 5.

CONNECTORS

This section describes the EXOS 302 board's connectors. These include the following:

- VMEbus connectors
- Transceiver connector
- Ethernet backplane connector
- RS-232-C serial connector

VMEbus Connectors

The following table shows the EXOS 302 signal assignments for the VME host bus connectors P1 and P2. Signals not used by the EXOS 302 board are designated n/u. Pins 13-16 of P2A and P2C can be jumpered.

Table 2-5: VMEbus Pin Assignments

PIN	P1A	P1B	P1C	P2A	P2B	P2C
1	VD00	VBSY-	VD08		+5V	
2	VD01	BCLR- n/u	VD09		GND	
3	VD02	ACFAIL-	VD10			
4	VD03	VBG0IN-	VD11		VA24	
5	VD04	VBG0OUT-	VD12		VA25	
6	VD05	VBG1IN-	VD13		VA26	
7	VD06	VBG1OUT-	VD14		VA27	
8	VD07	VBG2IN-	VD15		VA28	
9	GND	VBG2OUT-	GND		VA29	
10	SYSCLK n/u	VBG3IN-	SYSFAIL-		VA30	
11	GND	VBG3OUT-	VBERR-		VA31	
12	VDS1-	VBR0-	SYSRESET-	GND		
13	VDS0-	VBR1-	LWORD-	VTRMT+	+5V	VCLSN+
14	VWRITE-	VBR2-	VAM5	VTRMT-	D16 n/u	VCLSN-
15	GND	VBR3-	VA23	VRECV+	D17 n/u	VGND
16	VDTACK-	VAM0	VA22	VRECV-	D18 n/u	VE+12V
17	GND	VAM1	VA21		D19 n/u	
18	VAS-	VAM2	VA20		D20 n/u	
19	GND	VAM3	VA19		D21 n/u	
20	IACK-	GND	VA18		D22 n/u	
21	IACKIN-	SERCLK n/u	VA17		D23 n/u	
22	IACKOUT-	SERDAT n/u	VA16		GND	
23	VAM5	GND	VA15		D24 n/u	
24	VA07	IRQ7-	VA14		D25 n/u	
25	VA06	IRQ6-	VA13		D26 n/u	
26	VA05	IRQ5-	VA12		D27 n/u	
27	VA04	IRQ4-	VA11		D28 n/u	
28	VA03	IRQ3-	VA10		D29 n/u	
29	VA02	IRQ2-	VA09		D30 n/u	
30	VA01	IRQ1-	VA08		D31 n/u	
31	–12 V	+5 V SB n/u	+12 V	GND		
32	+5 V	+5 V	+5 V	+5 V		

Transceiver Connector

The EXOS 302 board's Ethernet connector is a 15-pin D-subminiature connector. Pin-outs, as shown in Table 2-6, are defined according to Ethernet specifications for the transceiver cable.

Table 2-6: Ethernet Connector (P3) Pin Definition

Pin #	Function
1	GND
2	CLSN (+)
3	TRMT (+)
4	GND
5	RECV (+)
6	GND
7	Not used
8	GND
9	CLSN (-)
10	TRMT (-)
11	GND
12	RECV (-)
13	Power + 12V
14	GND
15	Not used

Ethernet Backplane Connector

The EXOS 302 board allows routing the Ethernet transceiver signals to the VMEbus backplane via connector P2. Normally these signals are not connected to P2. Jumpers J17-J24 must be installed as shown in Table 2-7 to connect the indicated signal lines. If this type of connection is not desired, it is advisable to remove these jumpers, to prevent radiation from the unterminated signals, and noise from feeding into the transceiver signals.

RS-232-C Serial Connector

The EXOS 302 board provides a 9600 baud serial interface that can be used for attaching a terminal. Your terminal must have the following settings:

- 9600 baud
- No parity
- 7 bit
- 1 stop bit

For software development purposes, a debugger program that uses this interface is available from Excelan. The RS-232-C Serial Connector (designated as P4) is a dual-row, 26-pin, header-wired connector. This configuration of P4 allows making of flat cable with a 25-pin D-subminiature connector that can be connected directly to a standard terminal. Note that the pin numbering scheme for a D-Subminiature connector is different from that of a flat-cable connector.

Table 2-7: Configuring Ethernet Connector P2

P2 Pin	Signal	Jumper
A13	TRMT (+)	J18
A14	TRMT (-)	J19
A15	RECV (+)	J20
A16	RECV (-)	J21
C13	CLSN (+)	J22
C14	CLSN (-)	J22
C15	GND	J24
C16	Power (+12 V)	J17

Table 2-8 shows the pin definition for the P4 connector. Figure 2-3 shows the pin-out for the P4 connector, using the "flat cable numbering scheme."

Table 2-8: RS-232-C P4 Connector Pin Definition

Pin #	Function
3	Receive data from terminal
5	Transmit data to terminal
7	CTS from terminal
9	RTS to terminal
11	DTR to terminal
13	Signal ground

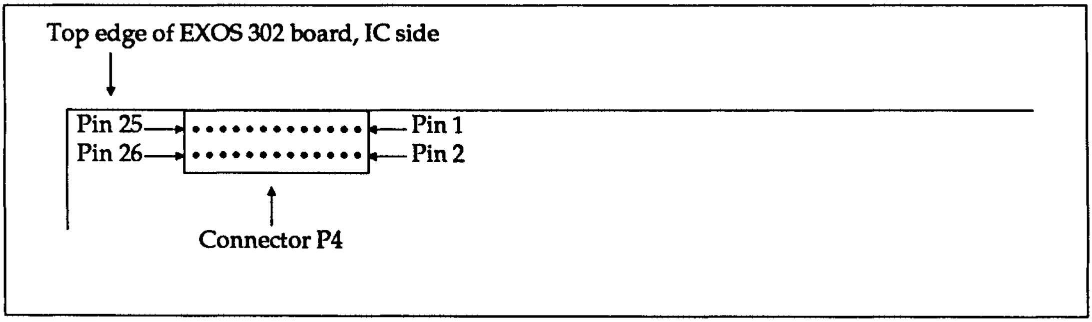


Figure 2-3: RS-232-C P4 Connector

SELF-TEST OPERATION

The NX 300 firmware performs a series of tests that exercise the hardware and software components of the EXOS 302 Intelligent Ethernet Controller. In addition to ensuring that the EXOS 302 board is functioning properly, the tests can identify specific software or hardware problems associated with configuration or operation. Errors are reported via an LED, making it possible for them to be identified and corrected (refer to the section Errors, later in this Chapter).

When the EXOS 302 board is reset by the VMEbus system reset line or by host software, the NX 300 firmware runs comprehensive diagnostic tests on the EXOS 302 board's components. The tests complete in about 3 seconds, after which the board is ready to be configured by the host. If the tests fail, this is reported to the host via port A and B status bytes. For additional information, refer to the chapter Initialization and Host Interface, in the *NX 200 Network Executive Reference Manual.*

Test progress and status are also reported via an LED at position DS1. When the EXOS 302 board is reset, this LED is lit, and it remains lit while the self-tests are in progress. When the self-tests complete, the LED flashes evenly until the EXOS 302 board is configured by the host or from the Ethernet network. After configuration, LED DS1 turns off.

If the diagnostics indicate a hardware problem, the LED is lit constantly, or it communicates an error code by flashing long and short pulses. Refer to the following section, Errors, for information about flashing LED error codes.

ERRORS

The NX 300 firmware handles all Ethernet error conditions, including user software configuration and native hardware errors. Additionally, the NX 300 firmware monitors for fatal hardware and software errors that may occur during general network operation.

Error codes are 8-bit numbers and are presented bit by bit, starting with the most significant bit. A long pulse is a "1" bit and a short pulse is a "0" bit. The error code is continuously repeated, with a pause in between to demarcate the starting point. Table 2-7 specifies all defined error codes for the EXOS 302 board.

If the LED at location DS1 flashes an error code, you can display its textual form by attaching a terminal to the RS-232-C serial connector on the EXOS 302 board and then typing a "?" character. Your terminal must have the following settings:

- 9600 baud
- No parity
- 7 Bit
- 1 Stop bit

Errors are divided into the following four groups:

- Warnings (70H-72H)
- Fatal software configuration errors (A0H-AFH)
- Fatal hardware errors (88H and B0H-BFH)
- Fatal run-time errors (91H, C0H-CFH,and F0H)

Warnings

Warnings are nonfatal errors. When a warning is encountered, the self-test continues to execute while the error code is displayed by the LED at position DS1. The following warning error codes can occur:

70H: Transceiver fuse failed or not present.

This error occurs if the transceiver fuse is blown out or is not present. Check the fuse, and install or replace as necessary.

71H: Loopback test failed.

This error occurs when the transceiver is faulty or its connection to the network is faulty. This error does not prevent the NX 300 firmware from attempting to use the network. However, possible causes should be investigated. Check the transceiver and the transceiver cable and then take necessary corrective action.

72H: Negative volts (–12 V) supply failed.

This error is of consequence only if you plan to use the EXOS debugger on the board, in which case you should return the board to Excelan as detailed in Appendix C.

Fatal Software Errors

Software configuration errors generally occur from entering inappropriate values for system configuration. On rare occasions, they can result from a bad bit in a memory chip that is interpreted by NX 300 as an invalid value. Software-generated error codes are useful for debugging user-written network protocol software.

Fatal Hardware Errors

Fatal hardware errors generally occur when the NX 300 firmware encounters specific EXOS hardware failure. In general, if you encounter any of these errors, you should return the board to Excelan as detailed in Appendix C.

Error BAH is a system-bus error (read-parity or sytem-bus timeout). It can be caused by an incorrect hardware configuration, system memory error, or incorrect memory address (a software error).

Run-time Errors

The run-time errors are encountered, as the name suggests, during normal board operation; they do not occur on reset or during self-testing. These errors can be produced by either software, hardware, or a combination of the two.

The errors listed below are generally associated with the physical interface between the host and the EXOS 302 board.

- C1H – Host memory read-write test failed
- C9H – NMI interrupt for bus error failed
- CAH – Host interrupt test failed
- CCH – Divide error exception
- CDH – Undefined interrupt type
- F0H – 80286 segmentation fault

The error "CBH - Command unit test failed" can occur between the host and the network if the transceiver malfunctions or is physically not connected to the network. In such a case, reconnect the host to the network and restart the network hardware. If the error occurs again, it usually indicates a transceiver malfunction.

If the error F0H occurs, the following additional information is displayed on the LED:

F0H followed by the four bytes of CS:IP, high byte first.

If any of the run-time errors (except the error CBH) are encountered, you should contact Excelan if you are using Excelan software. Otherwise, you should attempt to isolate and fix the problem indicated by the error message.

Table 2-9: Self-Diagnostic/Configuration Error Codes

Hex Code	Pulse Code	Explanation of Error Code
Warnings		
70H	.---....	Transceiver fuse failed or not present.
71H	.--- ...-	Loopback test failed.
72H	.---..-.	Minus 12 volts (–12 V) supply failed.
Software-generated Errors		
A0H	-.-.	Invalid address for configuration message.
A4H	-.-. .-..	Invalid operation mode parameter.
A5H	-.-. .-.-	Invalid host data format test pattern.
A7H	-.-. .---	Invalid configuration message format.
A8H	-.-. -...	Invalid movable data block parameter.
A9H	-.-. -..-	Invalid number of processes parameter.
AAH	-.-. -.-.	Invalid number of mailboxes parameter.
ABH	-.-. -.--	Invalid number of address slots parameter.
ACH	-.-. --..	Invalid number of hosts parameter.
ADH	-.-. --.-	Invalid host queue parameter.
AEH	-.-. ---.	Improper objects allocation.
Hardware-generated Errors		
B1H	-.-- ...-	Memory test failed for 0-128K.
B2H	-.-- ..-.	Memory test failed for 128K to top of memory.
B3H	-.-- ..--	Counter test failed.
B4H	-.-- .-..	Interrupts test failed.
B5H	-.-- .-.-	Transmission test failed.
B6H	-.-- .--.	Receive test failed.
B7H	-.-- .---	Local loopback data path test failed.
B8H	-.-- -...	CRC test failed.
B9H	-.-- -..-	Checksum on physical address PROM failed.
BAH	-.-- -.-.	Bus error (parity or timeout).
BBH	-.-- -.--	Ethernet chip initialization failed.
BCH	-.-- --..	Ethernet chip self-test failed.
BDH	-.-- --.-	Ethernet chip resource counter failed.
BEH	-.-- ---.	External loopback test alignment error.
Run-time Errors		
C0H	--..	Specified time exhausted.
C1H	--.. ...-	Host memory read/write test failed.
C8H	--.. -...	Parity hardware logic failed.
C9H	--.. -..-	NMI interrupt for bus timeout failed.
CAH	--.. -.-.	Host interrupt test failed.
CBH	--.. -.--	Command unit test failed.
CCH	--.. --..	Divide error exception.
CDH	--.. --.-	Undefined interrupt type.
CEH	--.. ---.	Command not executed by the CU of the 82586.
CFH	--.. ----	Command block sync failed between h/w and s/w.
F0H	----....	80286 segmentation fault.

GENERAL SPECIFICATIONS

Table 2-10 lists the general specifications for the EXOS 302 Intelligent Ethernet Controller.

Table 2-10: EXOS 302 Specifications

Characteristic	Description
Power Requirement	+5 VDC @ 6.0 Amp max +12 VDC @ 0.5 Amp max (for transceiver and serial port) -12 VDC @ 0.05 Amp max (for serial port only)
Operating Environment	Temperature: 5°C to 50°C Humidity: 0% to 90% without condensation
Slave Port Addresses	Jumper-selectable from 0000001 to FFFFFF01 in increments of 64 Kbytes, with three intervening boundaries Factory setting (24-bit address): Port A : FF0001H; Port B : FF0003H; Port C : FF0005H
Interrupt Vector Address	Software programmable
Interrupt Priority Level	Jumper selectable, one of seven levels (INT1-INT7)
VMEbus Timeout	None
Physical Size	One 6-layer, double-height, printed circuit board

3

INSTALLATION

3 INSTALLATION

Installing the EXOS 302 board is a relatively simple task. In a typical VMEbus system, it consists of turning off the power, configuring the EXOS 302 board and system backplane (if required), inserting the board in the card cage, and establishing the required connection through appropriate cables and connectors. In general, the complexity of the installation procedure depends on the physical design of the host machine rather than on the design of the EXOS 302 board.

The EXOS 302 Intelligent Ethernet Controller board is designed for use in VMEbus-based computer systems with different types of card cage and enclosure designs. It is produced in the following two configurations suitable for installation in the indicated VMEbus-based system types.

- **EXOS 302-04.** This configuration has the Ethernet connector and two ejectors on the top side of the board. The EXOS 302 board in this configuration is suitable for use in Integrated Solution's VME-based system and other systems that use conventional pull-out or backplane-access type card cages.

- **EXOS 302-05.** This configuration has a front-panel plate attached to the top side of the board with the Ethernet connector accessible through a cut-out in the plate. EXOS 302 boards in this configuration are suitable for installation in Motorola VMEbus-based systems or systems that have front-installed card cages.

Actual installation of the EXOS 302 Intelligent Ethernet Controller depends upon the board's physical configuration and the VMEbus system card cage and enclosure design. The following sections describe the installation of an EXOS 302 board in a generic VMEbus-based system, and provide examples of installing the board in an Integrated Solutions system and in a Motorola Delta system.

As shipped from the factory, the EXOS 302 Intelligent Ethernet Controllers (all configurations) have the same default values set for various jumper-selectable options. You should make sure that these defaults are suitable for your system's hardware and software configuration. If any options conflict with your system configuration, you can change those options by reconfiguring the associated jumpers, as described in Chapter 2.

In particular, you should check the default values for the following four options:

- VMEbus Address Width. The EXOS 302 board is shipped configured as A24 MASTER and SLAVE. It can be configured as A32 MASTER and SLAVE, or A32 MASTER and A24 SLAVE. Jumpers J44 and J46 define the two widths (refer to Appendix A). Make sure you understand how the host system is operating. Some systems limit nonhost masters and/or slave I/O devices to A24 operation.
- Slave Port Address — Ports A, B, and C. The default base address for these ports is FF0000, in A24 SLAVE mode. You can select another address by reconfiguring jumpers J34-J41 (for address bits 23 through 16) and J42-J43 (for address bits 15 through 7). Address bits 6 through 3 are not decoded. If using A32 SLAVE mode, jumpers J26-J33 configure address bits 24 through 31.
- Bus Priority Resolution Level. The default setting is for level 3. You can change this setting by reconfiguring jumpers J57-J68.
- Bus Interrupt Level. The default setting is for level 5. You can change this setting by reconfiguring jumpers J47-J56.

A TYPICAL INSTALLATION PROCEDURE

The procedure for installing an EXOS 302 board varies from system to system. In general, however, you must perform the following steps:

1. Make sure that the default Slave Port (Communication Register) base address, Interrupt level, and Bus Priority level do not conflict with the current configuration of your system. If necessary, reconfigure the board to suit your system.

2. Shut down the system by following the procedure specified for your system.

3. Consult your system manual for placement of the EXOS 302 board. The general rules are as follows:

    - The CPU boards must be at one end of the card cage.
    - The memory boards must be next to the CPU boards.
    - The disk controller must be next to the memory boards.
    - The I/O boards, including the EXOS 302 board, must be next to the disk controller.
    - There must be no empty slots between CPU board(s) and all other boards that require interrupts or VMEbus ownership. Empty slots usually require backplane jumpers or special continuity boards, which maintain the daisy chain signals over the empty slots. The EXOS 302 board can be plugged into any slot except slot 1. (Slot 1 is reserved for the system controller, containing the arbiter, etc.) Some jumper configurations of the system might have to be changed to accommodate the EXOS 302 board; consult your system manual for jumper locations and settings. You must be careful when changing jumper configurations, because changing jumper options can have profound effects on system operation.

4. Plug the male end of the Ethernet transceiver cable (or internal system cable) into the Ethernet connector on the EXOS 302 board. Make sure that the latches on the connector are engaged properly so the cable cannot be pulled out accidentally.

Step 4 completes the installation of the EXOS 302 board. You can now turn on the power in the sequence described in the system manual, proceed with the normal initialization process, and install the appropriate software package(s).

INSTALLING IN AN INTEGRATED SOLUTIONS SYSTEM

The following is an example of a step-by-step procedure to install an EXOS 302 board in a VMEbus-based Integrated Solutions system. It should be noted that this procedure is included here only as an example. For the actual installation procedure, you should consult the Integrated Solutions system manual.

☞ The Integrated Solutions system uses the EXOS 302-04 configuration board and a special internal cable (Excelan Part No. 1213-36). The male end of that cable connects to the EXOS 302 board and the female end screws onto the external connector panel of the system. The transceiver cable then connects to the external panel.

1. Make sure that the Slave Ports (Communication Register) base address, Interrupt level, and Bus Priority level do not conflict with the current configuration of your system. If necessary, reconfigure the board to suit your system. Note that many ISI systems cannot accomodate an A32 MASTER due to variations from the VMEbus standard of the upper address byte. Consult an ISI or Excelan representative if this is a requirement.
2. Follow the shutdown procedure to turn off the power.
3. Wait one minute for the peripherals to settle down.
4. Locate the 24-slot card cage rack.
5. Holding either side of the front plate of the card cage, carefully pull the card cage forward until the slide locks.
6. Loosen all the screws that secure the top cover of the card cage and remove the cover to expose the inside of the card cage.
7. Install the EXOS 302 board according to the general rules for board placement listed in Step 3 of the preceding section.
8. Attach the male end of the special internal cable to the 15-pin Ethernet connector (P3) on the EXOS 302 board. Secure the latch to make sure that the cable will not be disconnected accidentally.
9. Route the cable to the rear left corner and secure the female end of the cable to the back panel with two screws.

10. Put the top cover back on and secure all the screws.
11. Slide the VME-bus card cage back in.
12. Connect the system to the network by attaching one end of the transceiver cable to the female connector on the backplane and the other to the transceiver already attached to the network.
13. Turn on the system power by following the procedure specified for your system.

Step 13 completes the installation of the EXOS 302 hardware. You can now proceed with the usual system initialization process.

INSTALLING IN A MOTOROLA DELTA SYSTEM

The following is an example of a step-by-step procedure to install an EXOS 302 Intelligent Ethernet controller in a VMEbus-based MOTOROLA system. This procedure is included here only as an example.

☞ The MOTOROLA system uses the EXOS 302-05 configuration.

1. Make sure that the default Slave Port (Communication Register) base address, Interrupt level, and Bus Priority level do not conflict with the current configuration of your system. If necessary, reconfigure the board to suit your system.
2. Follow the shutdown procedure to turn off the power.
3. Wait one minute for the peripherals to settle down.
4. Install the EXOS 302 board according to general board placement rules listed in Step 3 of the section A Typical Installation Procedure.
5. Attach the male end of the transceiver cable to the 15-pin Ethernet connector (P3) on the EXOS 302 board. Secure the latch to make sure that the cable will not be disconnected accidentally.

6. Plug the female end to the transceiver, and secure its latch.

Step 6 completes the EXOS 302 hardware installation. You can now turn on the power according to the prescribed sequence and proceed with the normal system initialization process.

APPENDICES

COMPONENT LOCATION

A

Figure A-1 shows the location of the main components and various jumpers on the EXOS 302 board. Refer to Table 2-2 for a description of various jumpers.

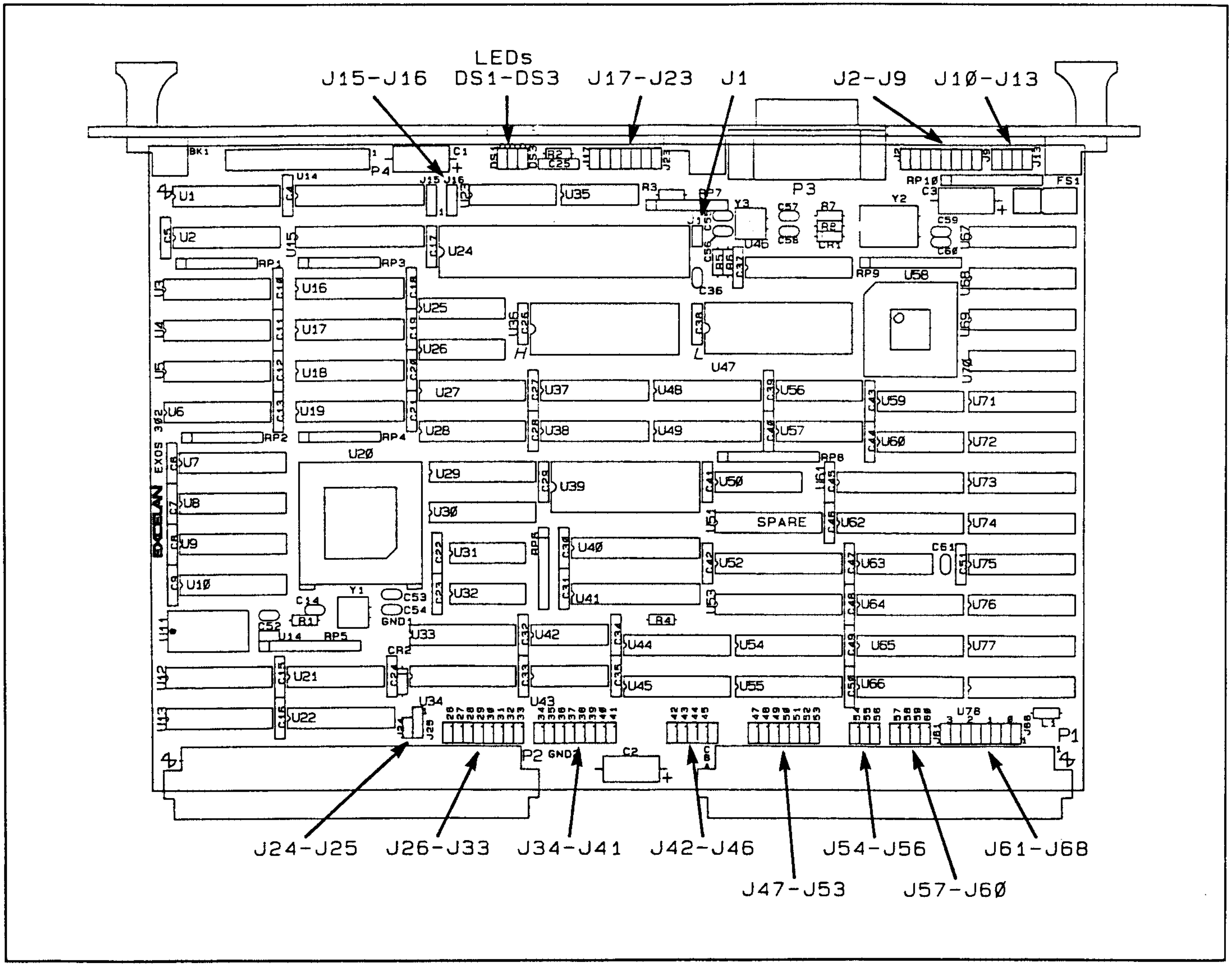


Figure A-1: EXOS 302 Component Location

SCHEMATIC

B

The schematic for the EXOS 302 Intelligent Ethernet Controller is shown on the following pages. The schematic is furnished for reference only.

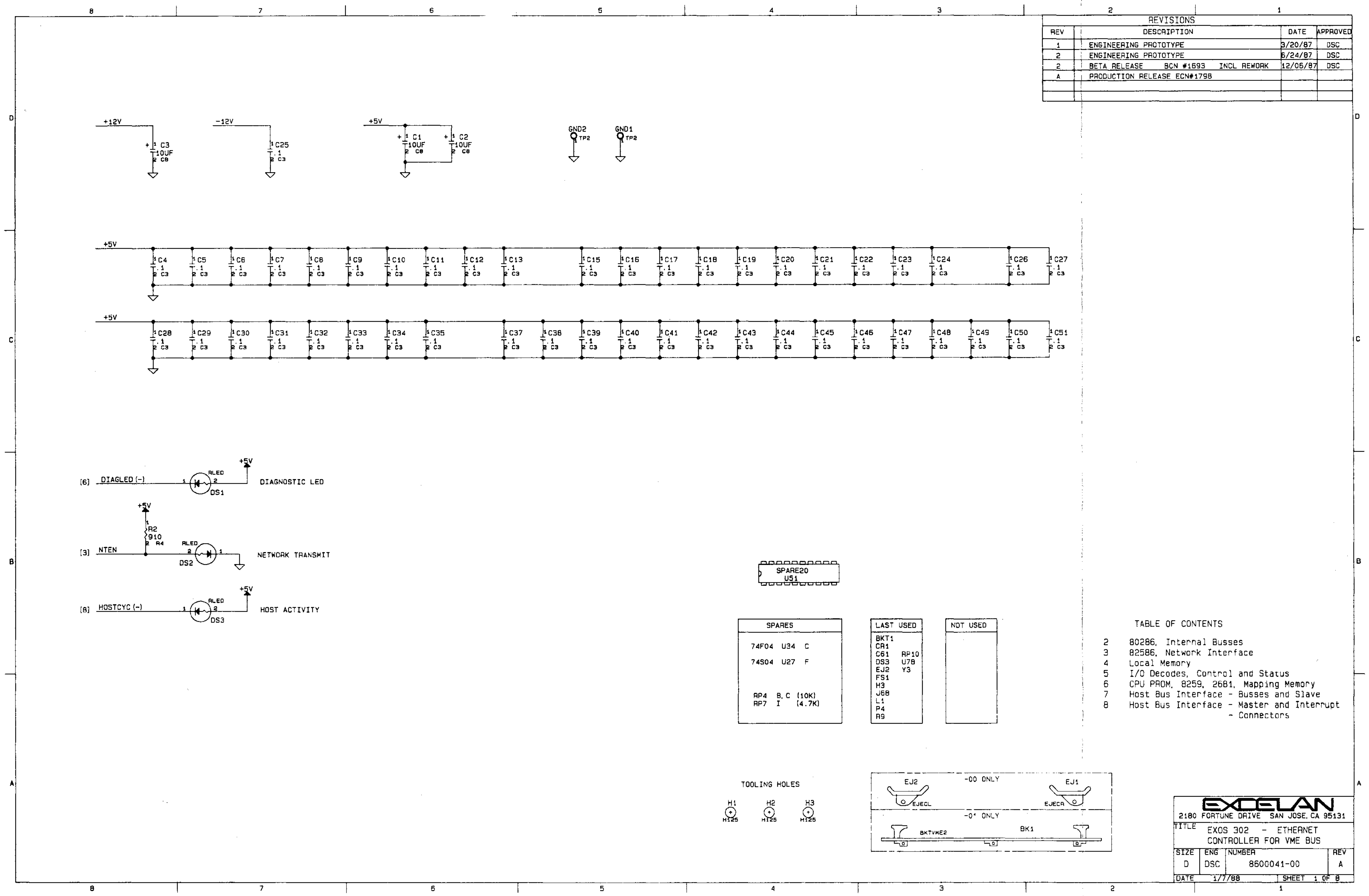


Figure B-1: EXOS 302 Schematic

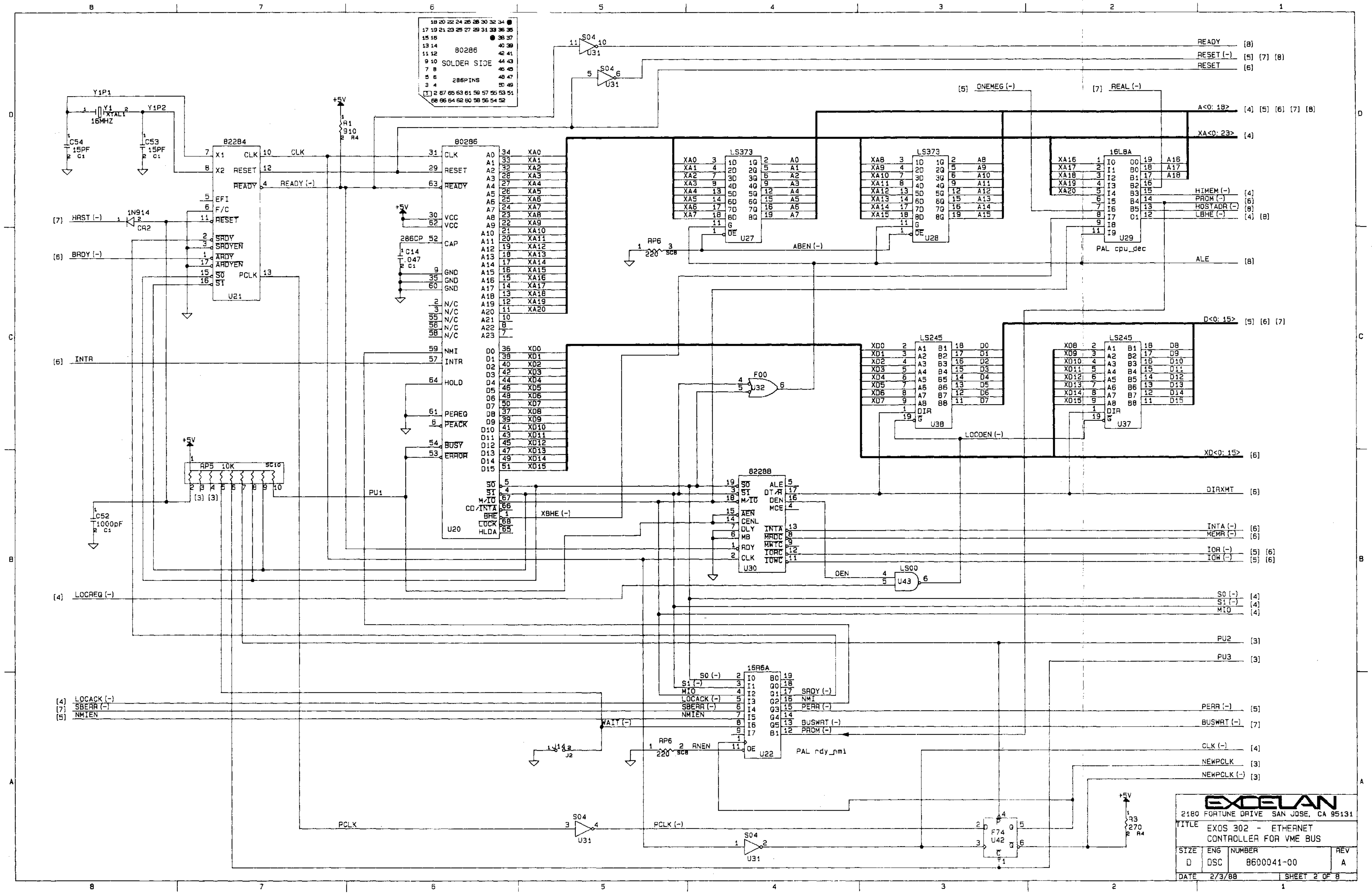


Figure B-1: EXOS 302 Schematic (continued)

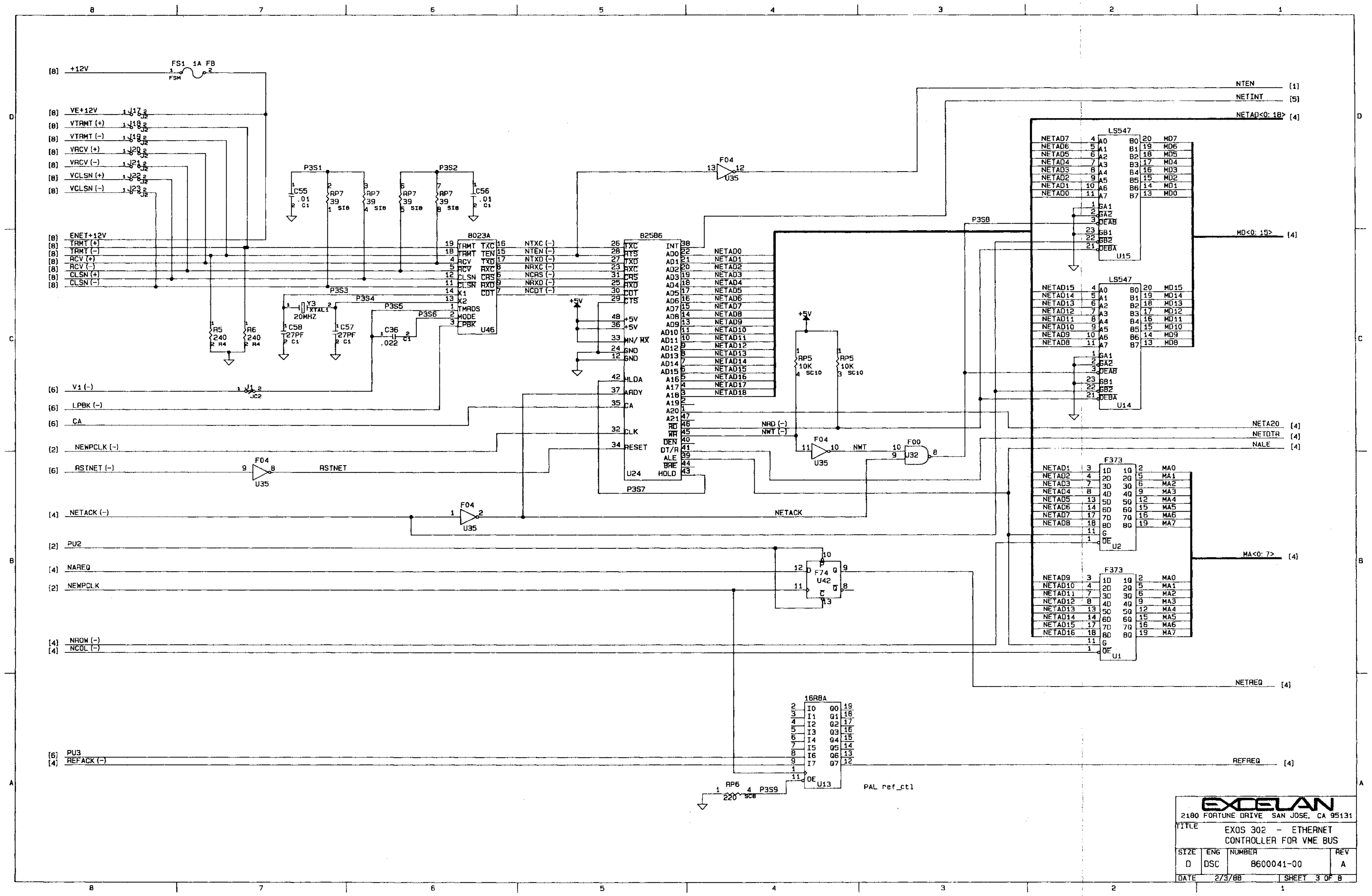


Figure B-1: EXOS 302 Schematic (continued)

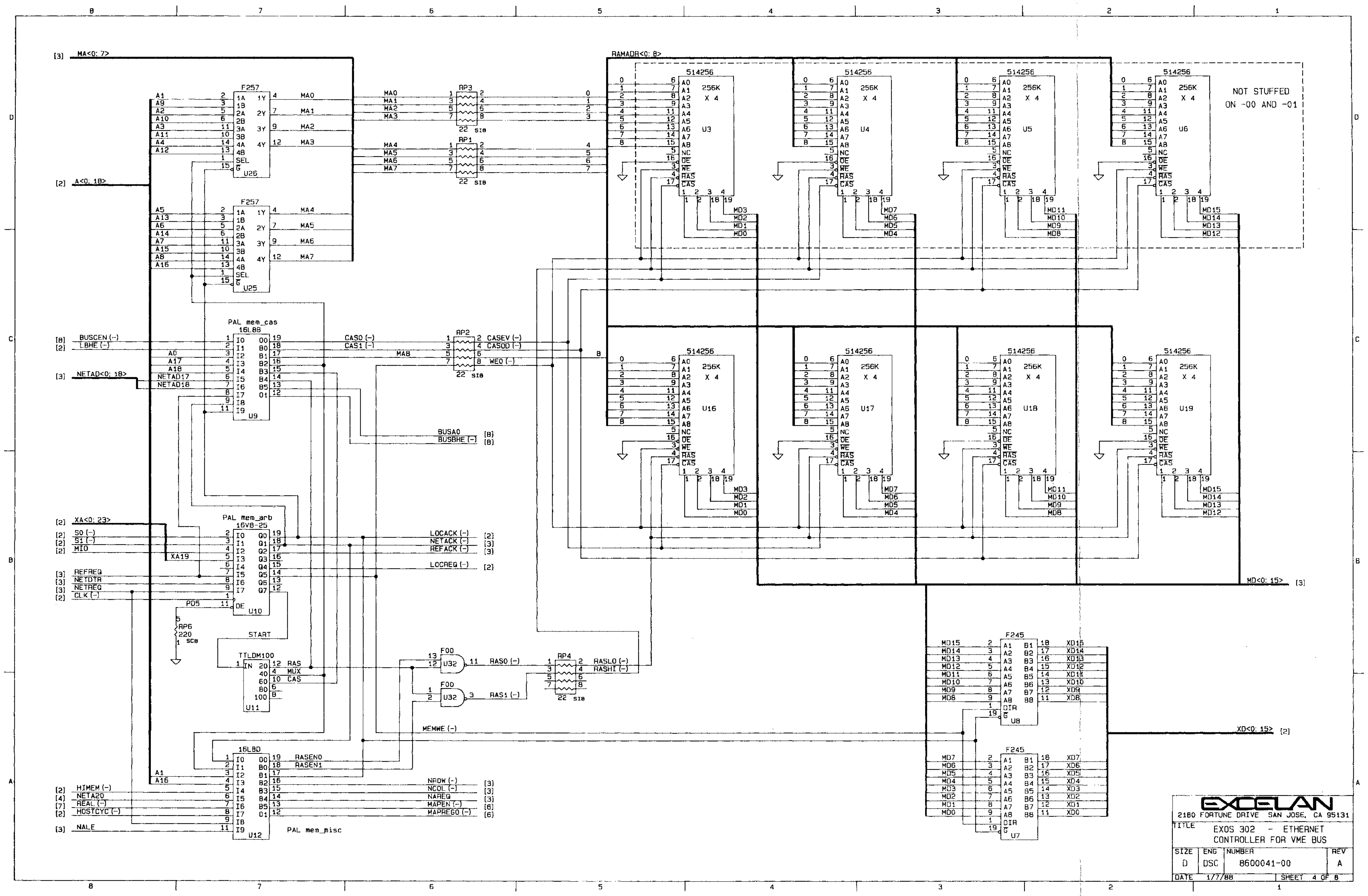


Figure B-1: EXOS 302 Schematic (continued)

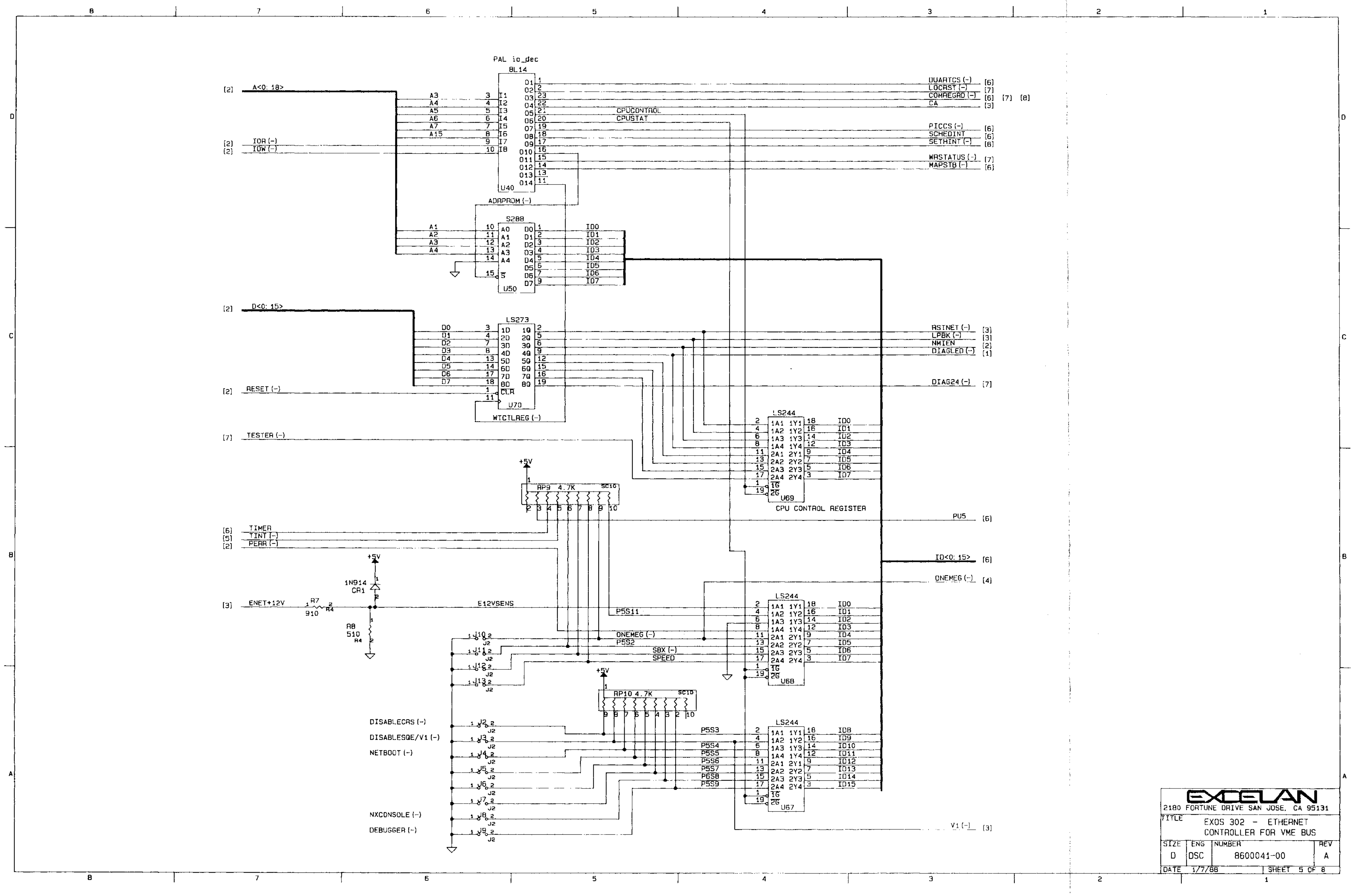


Figure B-1: EXOS 302 Schematic (continued)

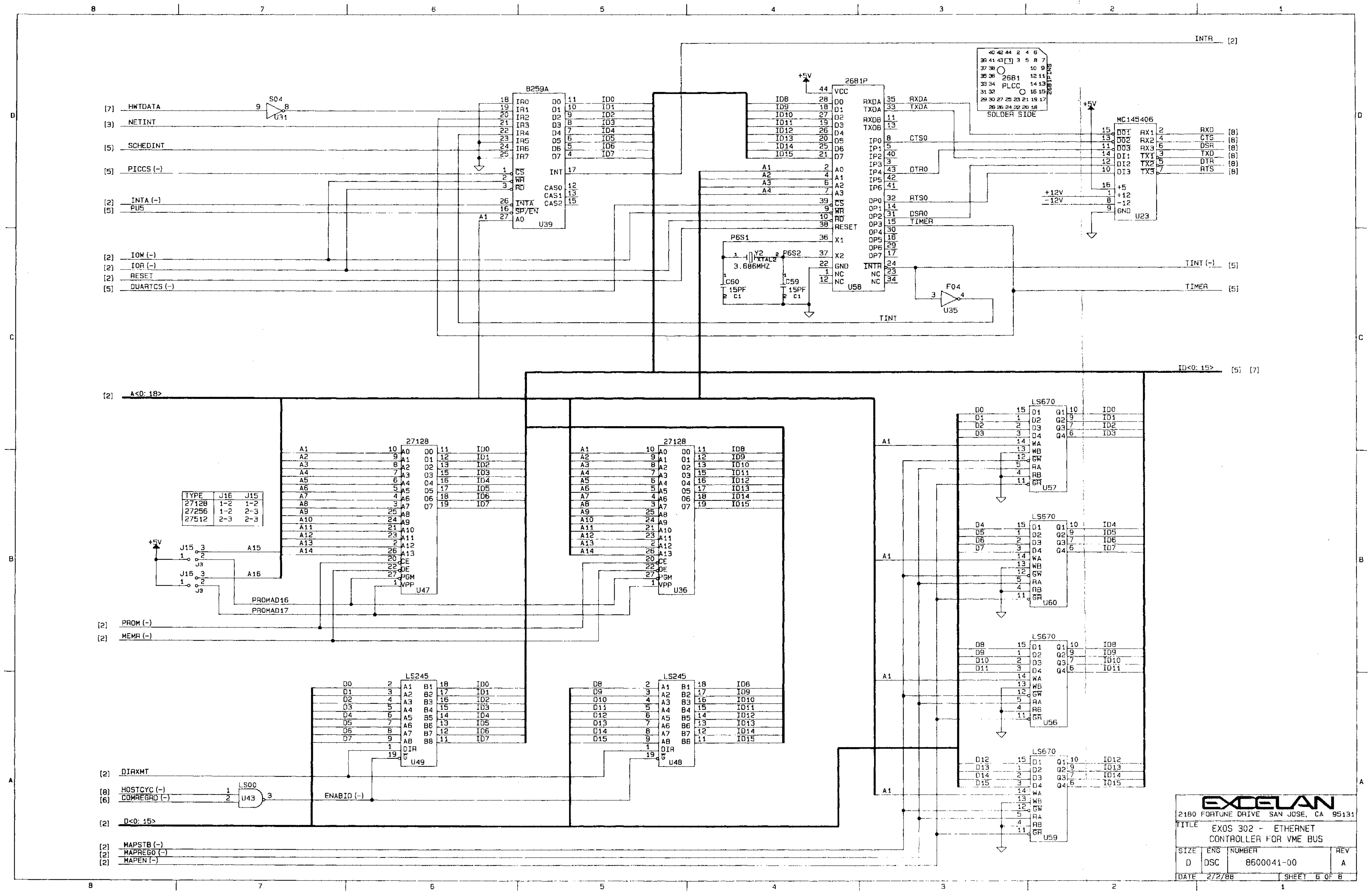


Figure B-1: EXOS 302 Schematic (continued)

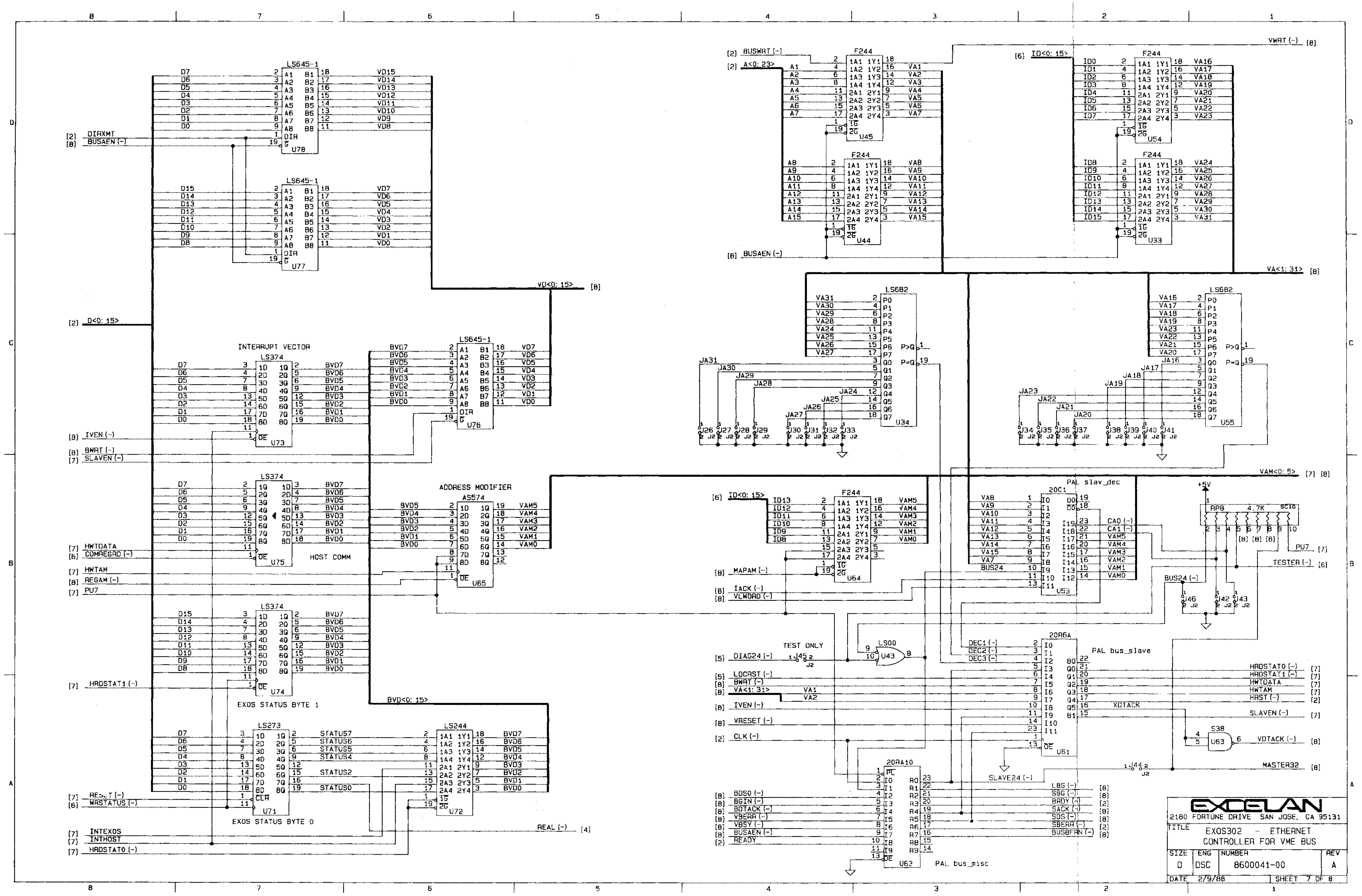


Figure B-1: EXOS 302 Schematic (continued)

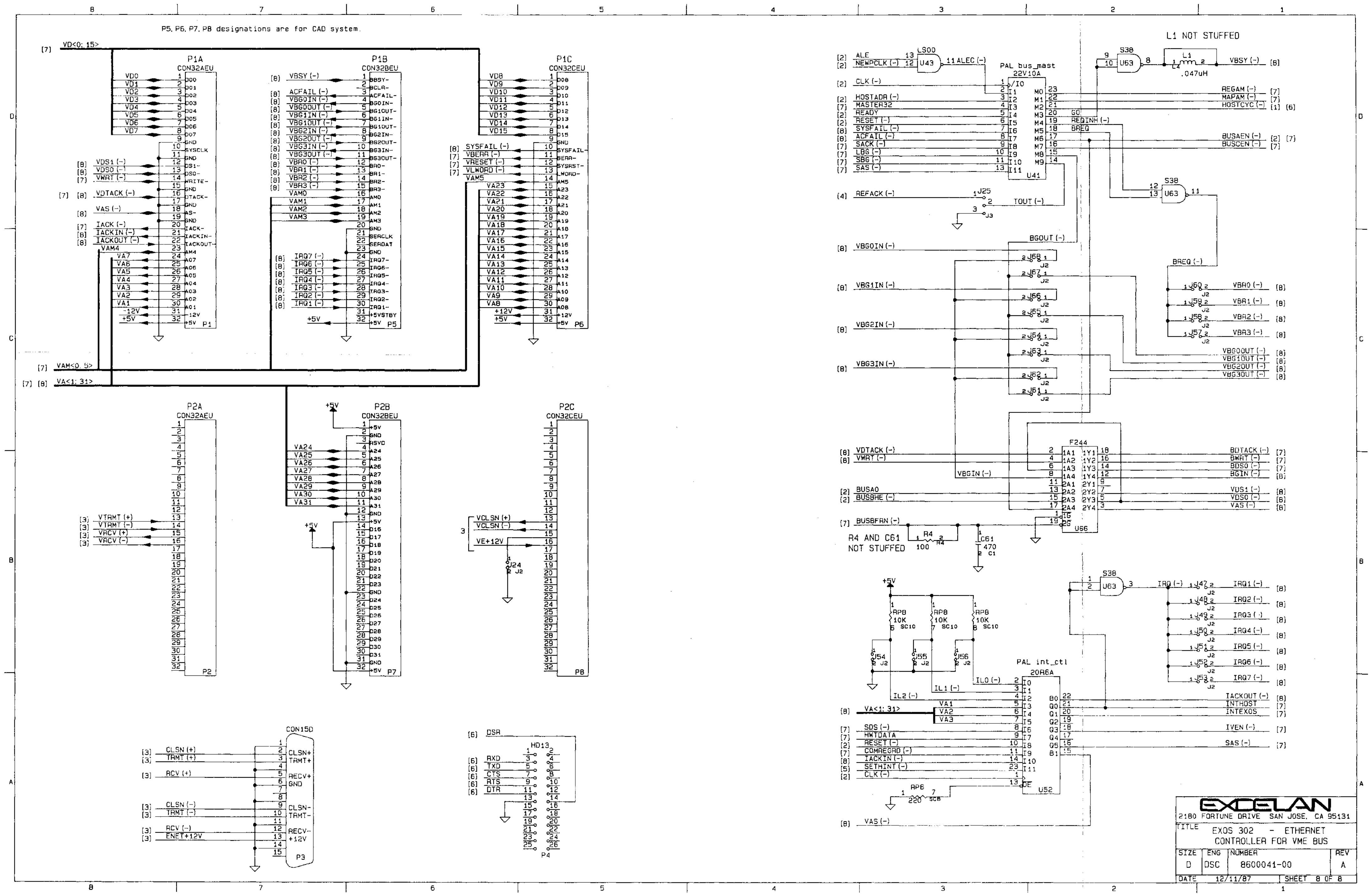


Figure B-1: EXOS 302 Schematic (continued)

C

SERVICE INFORMATION

Prior to shipping, each EXOS 302 Ethernet Intelligent Ethernet Controller board is thoroughly tested and exercised, both at the component level and at the system level. However, since it is not possible to simulate every possible situation that might exist on the network, occasionally you might encounter problems with the EXOS 302 board. For such rare cases, Excelan provides prompt technical service assistance. After hearing from you, our technical support personnel will discuss the problem with you over the telephone. Quite often, the problem is a simple one and can be resolved during such discussions. At other times, it may be necessary for you to ship the board to Excelan for repair or replacement.

While communicating with the Excelan Service Center, you might need to provide or refer to information included in Appendices A and B of this manual.

Appendix A presents an assembly diagram of the EXOS 302 board that shows the location of the EXOS 302 serial number, assembly or part number, board revision level, the various components, and jumpers.

Appendix B provides a set of EXOS 302 schematic diagrams.

If you encounter any performance problem with your EXOS 302 board and you or your system administrator cannot resolve it, please contact Excelan at the following address:

Excelan Service Center
2180 Fortune Drive
San Jose, CA 95131
(408) 434-2285

For prompt assistance, the Service Center requires the following information. Please have it ready when you call.

- Company name
- Technical contact
- Company address
- Telephone number
- Product name, part number, and serial number
- Your purchase order number or service contract number

The Customer Service staff will then discuss the problem you are experiencing with the EXOS 302 board. They will help you to isolate the fault and fix it, if possible.

If the problem cannot be resolved through this discussion, the service staff will give you a Return Material Authorization (RMA) number. At this point, depending on whether or not the unit is under warranty or extended warranty, the service staff will be able to advise you of any applicable charges for the sevice.

You should then securely pack the unit in its original or similar packing and return it to Excelan, Inc. at the address given above.

For detailed service terms, please read the "Warranty/Service Information" document shipped with your EXOS 302 board.