EtherLink II Adapter Technical Reference Manual

A Member of the EtherLink Product Family

3165 Kifer Road
Santa Clara, California 95052-8145
Printed in the U.S.A.

Manual Part No. 4185-01
Published January, 1989

Copyright Statement

Disclaimer

Contents

Chapter 11: Power Requirements

Chapter 12: Operational Specifications

Chapter 13: Physical / Mechanical Specifications

Chapter 14: Agency Approvals

Appendix A: Recommended Guidelines for Operation

Appendix B: Programming Sequences

Chapter 1: Introduction

This document provides information about the programming requirements of the gate array on the EtherLink II network adapter. The EtherLink II adapter is a high-performance, multi-packet buffering, low-cost Ethernet adapter, designed to operate in IBM PC, PC AT, and compatible personal computers. The gate array is the central source for the operation of the adapter by connecting directly to the PC bus interface and the adapter's buses. The gate array, composed of 16 registers, is subdivided into control registers (used for data transfers), configuration registers (used for establishing address assignments), and status registers (used for providing information to the software).

The software, by programming the gate array registers, determines the best suited operating parameters for the adapter operating environment. The operating parameters determine the connectivity for data transfers (memory mapped, DMA transfers) and transceiver connections (external, on-board). The data transfer parameters are programmable to support a variety of PC data transfer methods (DMA, programmed I/O, either byte or word, and memory-to-memory transfers). The DMA data transfer supports single byte and demand mode with the ability to personalize the number of bytes transferred to the personal computer.

The gate array also provides software programmable options that play a key role in reducing the time required to configure the EtherLink II adapter. The software programmable options are the DMA channels, interrupt channel, and transceiver type.

Chapter 2: Applicable Documents

EtherLink II Installation Guide
The *EtherLink II Installation Guide* provides detailed information about the operation of the EtherLink II adapter. Topics that are explained are installation, diagnostics, software installation, cabling, adapter configurations, and support.

IEEE 802.3 Specification

Technical Reference
Advanced Peripherals IEEE 802.3 Local Area Network Guide, published by National Semiconductor Corp., National part number 550083-001).

Technical Reference Personal Computer, published by IBM Corp., IBM part number 6322507.

Technical Reference Personal Computer AT, published by IBM Corp., IBM part number 6230070.

Technical Reference Personal System/2, Model 30, published by IBM Corp.

Chapter 3: General Description

This section provides an overview of the component blocks that comprise the EtherLink II adapter. The major functions of each block are described.

Gate Array

Decoding

All I/O addresses used by component blocks (LAN controller registers, Ethernet address PROM, gate array registers, interrupt sharing) on the adapter are decoded by the gate array. The I/O decoding logic uses the full 16-bit I/O address bus. This guarantees that the EtherLink II does not conflict with other adapters that do a partial (< 16 bits) decode.

Memory address decoding for the EPROM and for the static RAM during memory-mapped data transfers are decoded by a circuit in the gate array. The memory decoding logic uses the full 20-bit memory address bus. This full 20-bit decode provides a unique address for adapter memory components.

Packet Transfers

Transmit packets are "downloaded" (data transferred from the PC to the adapter) to the board-resident RAM by using either the DMA controller in the gate array or by programming the adapter for memory map mode. The gate array DMA controller supports handshaking with a motherboard-resident DMA controller (8237), using either single-byte or demand-mode transfers. The gate array DMA controller also interfaces to program transfer instructions (byte or word). Instructions supported are **outb, outw, rep outsb, rep outsw.**

Receive packets are "uploaded" (data transferred to the PC from the adapter) from the board-resident RAM by using either the DMA controller in the gate array or by programming the adapter for memory-map mode. The gate array DMA controller supports handshaking with a motherboard-resident DMA controller (8237), using either single-byte or demand-mode transfers. The DMA controller also interfaces to program transfer instructions (byte or word). Instructions supported are **inb, inw, rep insb, rep insw.**

Programmable Options

Interrupts

There are four (4) programmable interrupt channel options (2, 3, 4, 5). Following a power up condition, the output drivers (**irq2, irq3, irq4, irq5**) remain in the "off state" until programmed to the "on state." The output driver "on state" is defined as sinking current (driving the line to a logical zero < 0.4V) during interrupt request activity. The output driver does not have an "on state" that sources current. The source current for the line is provided by a discrete pull-up resistor.

The output drivers are returned to the "off state" during a software reset. The "off state" is defined as not sinking current (Ima). The line is driven to a logical one by the pull-up resistor. This DOES cause a low to high transition on the assigned interrupt line to the interrupt controller (8259) located on the motherboard. It is RECOMMENDED that you disable the interrupt channel assigned to the adapter in the interrupt controller during software reset.

The programmable interrupt lines support either dedicated interrupt operation or interrupt sharing. Dedicated interrupts are assigned to only one adapter at a time. Interrupt sharing allows multiple adapters to "drive" the same interrupt line.

DMA Channels

There are three (3) programmable direct memory access (DMA) channel options (1, 2, 3). Following a power up condition, the output drivers (**drq1, drq2, drq3**) remain in the "off state" until programmed to the "on state." The output driver "on state" is defined as either sinking current (driving the line to a logical zero < 0.4V) during non-DMA activity or sourcing current (driving the line to a logical one > 2.4V) during DMA activity. There are no conditions during the "on state" that cause the output driver not to drive the line.

The output drivers are returned to the "off state" during a software reset. The "off state" is defined as not sinking or sourcing current. The line is "floating" (unless the motherboard provides line termination resistors). This "floating" condition may cause a DMA request to the DMA controller (8237). It is advisable to disable the DMA controller during software reset.

DIX/BNC

The selection between using the onboard transceiver (BNC) or using an external transceiver (DIX, 15-pin connector) is programmable. The power-up condition causes the onboard transceiver to be selected (default). Bit 1 in the Control Register determines the transceiver mode.

The adapter is capable of delivering +12V @ 0.5A to either the BNC OR DIX port. It is recommended that the following conditions never exist:

- Attaching an external transceiver with the onboard transceiver selected (this may overload the +12V fuse.)
- Selecting the onboard transceiver while having an external transceiver attached (another overload condition for the fuse).

Arbitration

Requests for access to the board resident RAM (local packet buffer) are arbitrated by logic in the gate array. The LAN controller requests the bus to store packets it receives from the network in the receive packet segment of the local packet buffer, and retrieve packets from the transmit segment of the local packet buffer for transmission to the network.

Requests are prioritized with the LAN controller given the highest priority. LAN controller requests cause the current cycle to the local packet buffer to complete, then control is given to the LAN controller.

FIFO Logic

To compensate for the bus latency due to the LAN controller operation, a first in, first out (FIFO) circuit is implemented in the gate array. This FIFO allows the simultaneous transfer of data to the host processor in addition to data transfers to the local packet buffer from the LAN controller. The FIFO can be configured into a two 8-byte parallel operation or a single 16-byte serial configuration.

LAN Chips

LAN Controller

Data Conversion

The LAN controller reads byte parallel data from the local packet buffer during packet transmission, and converts that data to bit serial information for the encoder. For receive packets, the LAN controller receives bit serial data from the decoder and generates byte parallel data to the local packet buffer.

Packets

Transmit packets that are assembled in the local packet buffer (transmit segment) are retrieved by the LAN controller and passed to the encoder block. The data is passed to the encoder in the following sequence: 8 bytes of preamble (inserted by the LAN controller), destination address (6 bytes), source address (6 bytes), packet data (up to 1500 bytes) read from the local packet buffer, and 4 CRC bytes (generated by the LAN controller).

On receive packets, the LAN controller strips the preamble from the packet and checks the CRC bytes. The data written into the local packet buffer by the LAN controller is header information (4 bytes), packet data field (1500 max), and CRC bytes (4).

CRC Generate/Check

Cyclic redundancy check (CRC) is used to determine the validity of the transmitted data. The CRC bytes are appended to the packet by the transmitting LAN controller and checked by the receiving LAN controller. These CRC bytes are used in conjunction with the data in the packet to determine an error.

It is not correct to assume that a CRC error indicates a transmitter problem. A method to determine whether the transmitter has sent an error packet or the receiver CRC checker is defective is to accept error packets, then read the packet from the local packet buffer and perform a software CRC check, and then compare the results against the CRC bytes in the packet buffer.

Encoder/Decoder

The link between the transceiver (the component block that is attached to the coax) and the LAN controller (the component block that interfaces to the gate array) is the encoder/decoder component block. The encoder/decoder supplies all the timing clocks used for passing data between the LAN controller and the transceiver.

The packet information on the coax is encoded using the Manchester encoding scheme. Encoded packets, combining data and clocks, are derived from this component block, and received packets are separated into data and clock information by the decoder. Data passed to/from the transceiver is in the form of differential signal levels, with data passed to/from the LAN controller in the form of the signal levels.

Transceiver

An onboard transceiver capable of driving "thin Ethernet" coax is provided as a standard configuration on the EtherLink II adapter. Functions of the transceiver include transmitting packets onto the coax, receiving packets from the coax and detecting collisions when multiple transceivers are transmitting simultaneously.

RAM

The adapter uses static RAM for the board-resident packet buffer. The standard configuration is a single 8K x 8 device with an option to increase to a single 32K x 8 device. Software has direct access to the RAM through the host's memory address space (when configured in memory mapped mode). Indirect software access to the RAM data is through the FIFO logic. There are four address options for locating the RAM in host memory addressable space. The selected address is shared between the RAM and the EPROM.

NOTE: Selection of RAM vs. EPROM is controlled by bit 3 of the GA Configuration register.

Segments

The RAM is divided into transmit buffer space and receive buffer space. Using the standard RAM configuration of 8K, the transmit space is 1.5K (one maximum size packet) and the receive space is 6.5K.

EPROM

The adapter provides a socket for an 8K x 8 EPROM. A jumper-selectable memory address defines the EPROM segment within the ROM address space in the personal computer. The memory address is shared with the RAM during memory-mapped data transfer mode.

NOTE: Selection of RAM vs. EPROM is controlled by bit 3 of the GA Configuration register.

Ethernet Address

The Ethernet address block consists of a 32 x 8 PROM. The PROM contains the Ethernet address of the adapter in the first six locations. All 32 locations are accessible, with only the first six locations having a definition. Information in the PROM is accessed through the I/O address space.

Chapter 4: Configuration Data

This section defines the configuration options that EtherLink II offers and the method used for setting these options.

I/O Base Address

The adapter requires a contiguous block of 16 (xx0 -xxF) I/O locations where xx equals the I/O base address. The address block assigned to the adapter is jumper settable (1 of 2 jumpers on the adapter). There are eight possible options (350, 330, 310, 300, 2E0, 2A0, 280, 250), with only one jumpered at any given time.

LAN Controller

The I/O registers in the LAN controller reside at the jumpered I/O base address. The registers share the I/O addresses with the Ethernet Address PROM. The information is "windowed" into the address space controlled by the Control Register bits 2, 3. Bits 2, 3 of Control Register are initialized to window the high-order bytes of the Ethernet Address PROM.

Ethernet Address PROM Bytes 31-16

The high-order bytes of the PROM share the I/O address space with the LAN controller registers and the low-order bytes of the Ethernet Address PROM. The information is "windowed" into the address space controlled by bits 2, 3 of the control register. Bits 2, 3 of Control Register are initialized to window the high-order bytes of the Ethernet address PROM.

Ethernet Address PROM Bytes 15-0

The low-order bytes of the PROM shares the I/O address space with the LAN Controller registers and the high-order bytes of the Ethernet Address PROM. The information is "windowed" into the address space controlled by bits 2, 3 of Control Register 2. Bits 2, 3 of Control Register 2 are initialized to window the high-order bytes of the Ethernet Address PROM.

Gate Array

The registers in the gate array are located at the base I/O address + 400h (if the base address is jumpered to 300h, then the gate array is at 700h). This requires the personal computer in which the adapter is installed to have I/O addressing capability to 70Fh.

Memory Address Mapping

The adapter memory can be mapped into the host's address space. When mapping is selected, the adapter requires a contiguous block of 8K (1FFFh) locations. The address assigned to the adapter is jumper settable. There are four possible options (DC000, D8000, CC000, C8000) with only one jumpered at any given time.

EPROM

The adapter provides support for an 8Kbyte EPROM. The EPROM is located in the personal computer's ROM memory address space. The exact memory address of the EPROM is determined by the memory address jumper setting. The memory address selected for the EPROM is shared with the RAM, with selection controlled by the GA Configuration register.

RAM

The adapter provides support for up to 32Kbytes of static RAM. The RAM is located in the personal computer's ROM memory address space. The exact memory address of the RAM is determined by the memory address jumper setting. The memory address selected for the RAM is shared with the EPROM, with selection controlled by the GA Configuration register.

Interrupt Channels

There are four interrupt channel options available (irq2-irq5). The interrupt options are software programmable via the Interrupt/DMA Configuration Register bits 4-7. The IDC is part of the gate array register set.

DMA Channels

There are three DMA channel options available (drq1-drq3). The DMA channel options are software programmable via the Interrupt/DMA Configuration Register bits 0-3. The IDC is part of the gate array register set.

DIX/BNC

The adapter supports either "thin Ethernet" via an onboard transceiver or "thick Ethernet" via an AUI/ external transceiver. The DIX/BNC option is software programmable via Control Register 2 bit 1.

Chapter 5: Data Transfer

This section defines the data transfer methods that the EtherLink II adapter supports.

Control Blocks

DMA Controller

A high-performance DMA controller is built into the EtherLink II gate array. The DMA controller is used to deliver received packet data from the board resident packet buffer to the PC bus interface in an upload operation. It also controls the transfer of information from the PC bus interface to the local packet buffer in a download operation.

Page Start Register

The function of the Page STart Register (PSTR) is to define the starting address of the board resident packet buffer designated for received packets from the network.

The page start register (PSTR) in the gate array MUST be programmed with the exact value used for the LAN controller page start register (PSTART).

Page Stop Register

The purpose of the Page StoP Register (PSPR) is to define the end address within the board-resident packet buffer for receive packets from the network.

The Page StoP Register (PSPR) in the gate array MUST be programmed with the exact value used for the LAN controller page stop register (PSTOP).

Drq Timer Register

This register is used to control the number of bytes to be transferred between the system memory and the gate array DMA controller during demand mode DMA transfers. The gate array DMA controller will de-assert the DMA request signal on the PC bus interface when the count has been decremented to zero. The DMA request signal will be reasserted (provided terminal count has not been reached) AFTER the appropriate DMA acknowledge has been de-asserted by the system DMA controller. This allows other peripherals a chance to gain DMA access. It is recommended that the value loaded into the drq timer register does not cause the PC bus DMA transfer to exceed the refresh rate (15us).

GA (Gate Array) Configuration Register

With the standard RAM configuration of 8K, bit 0 MUST be programmed to a logical one. With the 32K RAM configuration, bit 0 MUST be programmed to a logical zero. The rationale for different values based on the RAM configuration is due to different pin assignments between the 8K device and the 32K device. Pin 26 is chip select (positive true) on the 8K device and address line 13 on the 32K device.

Control Register

Three bits in this register are related to the DMA transfer. These bits control the starting/stopping of the DMA controller, the direction of the DMA transfer, and the FIFO configuration.

Start DMA

Programming the start DMA bit to a logical one (positive true logic) causes the DMA controller to move data between the PC bus interface and the board resident packet buffer. The start DMA bit may be asserted coincident with the DMA direction bit. Prior to asserting the start DMA bit, the DMA address registers should be programmed with the address of the data in the packet buffer to be transferred by the DMA controller.

Programming the start DMA bit to zero (off) should only be done AFTER the transfer is complete. During a download operation (transfer ROM the system memory to the packet buffer) the high to low transition of start DMA bit causes the FIFO to be flushed of any residual data (residual data is less than 8 bytes). The DMA in progress bit in the status register is set "on" (a logical one) indicating a flush operation. It is ILLEGAL to change the value of the DMA address register during the time of a flush.

DMA Direction

Controls the direction of the DMA transfer. Setting the bit to a logical one moves data to the board resident packet buffer (a download operation). Setting the bit to a zero moves data from the board resident packet buffer to the system memory buffer (an upload operation). The DMA direction bit may be asserted simultaneously with the start DMA bit. It is ILLEGAL, however, to change the DMA direction bit after the start DMA bit is asserted.

Follow this programming sequence to initiate a change in the DMA transfer direction:

1. De-assert the start DMA bit.

2. Poll the DMA in-progress bit in the status register for a zero (loop until zero).

3. Assert start DMA bit with the new DMA direction value.

16-Byte Select (Double Buffer Select)

This bit controls the configuration of the FIFOs. The default configuration is two 8-byte FIFOs operating in parallel. Programming the bit to a one concatenates the two 8-byte FIFOs to form a FIFO depth of 16 bytes.

Status Register

Status information on the current operation is provided to the software via bits 3, 4, 5, 6, 7 in the status register. Status information is available to the software as long as the start DMA bit in the control register is asserted (logical one). Clearing the status register is done by EITHER an I/O write (no specific data value is needed on the PC data bus since the gate array decodes only the address) or by programming the start DMA bit in the control register to zero (off).

Data Port Ready

The function of this bit is to indicate when the FIFO(s) is ready to receive data from the PC bus or deliver data to the PC bus. For a download transfer (PC to adapter), the FIFO(s) is immediately ready for receiving data (when the start DMA bit is asserted). For an upload operation (adapter to PC), the data port ready bit is asserted after 8 bytes are loaded into the FIFO (if 16 bytes equal zero) or after 16 bytes are loaded into FIFOs (if 16 byte select equals one).

Data Port Underflow

This status bit indicates that the gate array detected that an I/O read was issued to the register files address, but the DMA controller had not yet loaded the register file(s) with data from the board-resident packet buffer.

Data Port Overflow

This status bit indicates that the gate array detected that an I/O write was issued to the register files address with the register files full. This overflow condition can occur only when the PC bus I/O access exceeds the DMA controller access speed.

DMA Terminal Count (T/C)

Asserted by the gate array to indicate that a the PC bus signal (terminal count T/C) pulsed. The DMA controller will continue to transfer data until it receives a terminal count pulse OR the start DMA bit in Control register is de-asserted.

DMA In Progress

The DMA in progress status bit is set to a one by the DMA controller logic in the gate array.The primary purpose of this status bit is to allow the software to determine the completion of the flush operation during a download operation. During the flush operation it is ILLEGAL for the software to change the value of the DMA address registers.

DMA Address Registers

The DMA address registers consist of two (2) 8-bit registers (DMA address MSB, DMA address LSB) capable of addressing 64Kbytes of board-resident packet buffer. The DMA address registers are used only by the gate array DMA controller to store (download) or retrieve (upload) data to/from the packet buffer. The DMA address registers MUST be loaded by software prior to setting the start DMA bit in the Control register. The value in the DMA address registers CANNOT be altered after the start DMA bit is set. You MUST initiate a DMA "shutdown" to change the value in the DMA address registers. A DMA shutdown is done by setting the start DMA bit in the Control register to zero and verifying that the DMA in progress bit in the status register is also zero.

Using the standard RAM configuration of 8Kbytes, the DMA address MSB and the LAN controller page start register MUST be loaded by software with a value of 20h.

Register File Access

Two consecutive I/O addresses are assigned to the register files. Byte instructions (inb, outb, ...) can be issued to either address, but word instructions (inw, outw, etc.) MUST be issued to the lower address.

Programmed I/O

Data can be exchanged between the register files and the PC data bus under software programmed I/O control. The number of bytes to be transferred for each burst is limited to 8 (if Control register bit 5 is zero) or 16 bytes (if Control Register bit 5 is a one). In either case, bit 7 (Data Port Ready) in the Status register MUST be checked after EACH burst transfer. During program I/O data transfers, the number of bytes transferred SHOULD be divisible by 8 to allow the data to be burst aligned. This alignment allows a demand mode DMA transfer(s) following a programmed I/O transfer, without having to reprogram the gate array (the gate array will continue where the program I/O ended). However, additional programming may be required to start the DMA controller on the motherboard.

DMA Transfer

The EtherLink II supports a variety of DMA data transfer methods along with the handshake protocols. Described below are some of the DMA transfer methods that the EtherLink II adapter supports.

Single Byte Mode

A DMA data transfer mode supported by the EtherLink II adapter. Programming the DMA controller on the motherboard for single-byte mode causes it to transfer ONLY one byte per DMA session. The DMA controller on the adapter is armed with multiple bytes available but will deliver a byte of data for each I/O read or write signal that it receives. It should be noted that each byte transferred in this mode has the overhead of re-arbitrating for the bus (lots of time loss).

Demand Mode

A method of data transfer supported by the EtherLink II Ethernet adapter. Demand mode is a DMA data transfer between the DMA controller on the motherboard and the DMA controller on the EtherLink II Ethernet adapter. Data transfer is started by the DMA controller on the adapter by rising the drq (DMA request) signal on the PC bus. The DMA controller on the motherboard responds by asserting the dack (DMA acknowledge) signal on the PC bus. The DMA controller on the motherboard proceeds with asserting the I/O read OR I/O write PC bus signal. The DMA controller on the adapter responds with a new byte of data for each I/O read OR I/O write signal it receives. The transferring of data continues as long as the drq signal is active.

Drq Timer

Controls the number of bytes to be transferred during a demand mode DMA transfer. The drq x (x equals the DMA channel assigned to the adapter) PC bus signal is asserted and data is transferred until the value loaded into the drq timer register has been decremented to zero. The drq signal is de-asserted once the drq timer register reaches zero and reasserted after the appropriate dack (DMA acknowledge) signal is de-asserted, indicating this DMA data transfer session has gracefully terminated. This sequence allows other adapters/devices the opportunity for DMA service.

Memory Mapped

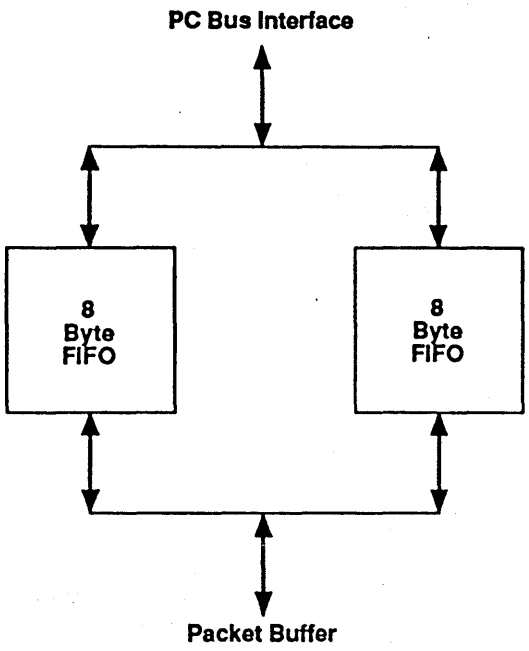
The EtherLink II adapter has the capability of "mapping" the board resident packet buffer into the personal computer's memory address space. The memory address is user determined by the position of the J1 jumper. The J1 jumper offers four possible addresses (dc000, d8000, cc000, c8000). The address selected by the J1 jumper also determines the address for the bootable EPROM socket. Memory-mapped mode is selected by setting the GA Configuration register to a value of 49h. Resetting memory-mapped mode is done by either resetting the system (power up reset) or executing an instruction at an address that matches the value in the vector pointer registers.

Word Instructions

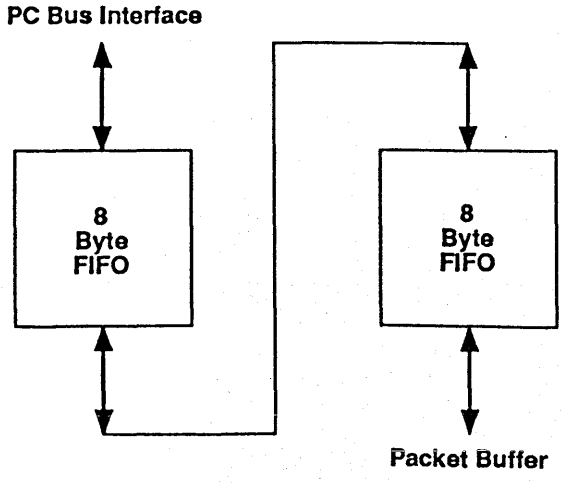
The EtherLink II adapter supports word I/O accesses to the register files (FIFO). The register files are the only register that can be accessed with a word instruction. The word access should be addressed to the lower address of the register files (base address + 40Eh). The Data Port Ready bit in the status register MUST be checked after four words (8 bytes) in parallel register file configuration or after eight words (16 bytes) in serial register file configuration.

FIFO Configurations - (Selected in GA Control Register)

Parallel Configuration



Serial Configuration



Chapter 6: Initialization

Power Up Reset

I/O Address	Gate Array Register	Data Value
base + 400h	Page Start	00
base + 401h	Page Stop	00
base + 402h	Drq Timer	00
base + 403h	Base Configuration	ww
base + 404h	PROM Configuration	xx
base + 405h	GA Configuration	00
base + 406h	Control Register	0A
base + 407h	Status Register	yy
base + 408h	Int/DMA Register	00
base + 409h	DMA Address MSB	00
base + 40Ah	DMA Address LSB	00
base + 40Bh	Vector Pointer 2	00
base + 40Ch	Vector Pointer 1	00
base + 40Dh	Vector Pointer 0	00
base + 40Eh	Register File LSB	zz
base + 40Fh	Register File MSB	zz

ww = dependent on position of J2 jumper
xx = dependent on position of J1 jumper
yy = gate array revision code
zz = unknown data until DMA transfer started

Software Reset

A software reset is initiated by setting bit 0 in the Control register (base address + 406h) to a logical one. A software reset emulates all properties of a hard reset (power on initialization) except that the configuration registers (base, PROM) are not reloaded. The configuration registers retain their previously loaded values. A reset condition continues to exist until the software reset bit is de-asserted (a logical zero). It is the responsibility of the driver software to de-assert the software reset bit after a time duration which provides reasonable assurance that the condition(s) that caused the reset to be issued has been cleared. The gate array (after the de-assertion of the software reset bit) registers are reset to the values in the following table.

I/O Address	Gate Array Register	Data Value
base + 400h	Page Start	00
base + 401h	Page Stop	00
base + 402h	Drq Timer	00
base + 403h	Base Configuration	++
base + 404h	PROM Configuration	++
base + 405h	GA Configuration	00
base + 406h	Control Register	0A *
base + 407h	Status Register	yy
base + 408h	Int/DMA Register	00
base + 409h	DMA Address MSB	00
base + 40Ah	DMA Address LSB	00
base + 40Bh	Vector Pointer 2	00
base + 40Ch	Vector Pointer 1	00
base + 40Dh	Vector Pointer 0	00
base + 40Eh	Register File LSB	zz
base + 40Fh	Register File MSB	zz

++ = retains data value
yy = gate array revision code
zz = register file pointer is reset to byte 0 of the register file. Residual data will exist in the register file.
* = contains data value 0B during software reset bit asserted.

Software Reset Programming Sequence

The software reset bit takes precedence and will abruptly end any operation in progress. The RECOMMENDED value to activate the assertion of the software reset is 03 (the on-board transceiver select bit and the software reset bit).

Asserting bits in the Control register and de-asserting the software reset bit REQUIRES two sequential program instructions. You issue the first instruction to de-assert the software reset bit and the second instruction to assert the desired bits. A simultaneous assertion of bits along with the de-assertion of the software reset bit results in a Control register of 0A.

Chapter 7: I/O Address Map

Address Range	Control Register Bits 3, 2	Assignments
Base Address	0, 0	LAN Controller
Base Address	0, 1	Ethernet PROM Bytes 0 - 15
Base Address*	1, 0	Ethernet PROM Bytes 16 - 31
Base Address + 400h	x, x	Gate Array Registers
	1, 1	Unassigned

* = default power-on value
x = don't care

Ethernet Address PROM Bytes 31-16

Address Decode	Control Register Bits 3, 2	Assignments
base address	1, 0	byte 16
base address + 1h	1, 0	byte 17
base address + 2h	1, 0	byte 18
base address + 3h	1, 0	byte 19
base address + 4h	1, 0	byte 20
base address + 5h	1, 0	byte 21
base address + 6h	1, 0	byte 22
base address + 7h	1, 0	byte 23
base address + 8h	1, 0	byte 24
base address + 9h	1, 0	byte 25
base address + Ah	1, 0	byte 26
base address + Bh	1, 0	byte 27
base address + Ch	1, 0	byte 28
base address + Dh	1, 0	byte 29
base address + Eh	1, 0	byte 30
base address + Fh	1, 0	byte 31

Ethernet Address PROM Bytes 15-0

Address Decode	Control Register Bits 3, 2	Assignments
base address	0, 1	Ethernet Station Address byte 0
base address + 1h	0, 1	byte 1
base address + 2h	0, 1	byte 2
base address + 3h	0, 1	byte 3
base address + 4h	0, 1	byte 4
base address + 5h	0, 1	byte 5
base address + 6h	0, 1	byte 6
base address + 7h	0, 1	byte 7
base address + 8h	0, 1	byte 8
base address + 9h	0, 1	byte 9
base address + Ah	0, 1	byte 10
base address + Bh	0, 1	byte 11
base address + Ch	0, 1	byte 12
base address + Dh	0, 1	byte 13
base address + Eh	0, 1	byte 14
base address + Fh	0, 1	byte 15

LAN Controller

Address Decode	Control Register Bits 3, 2	Assignments
base address	0, 0	CR
base address + 1h	0, 0	PSTART
base address + 2h	0, 0	PSTOP
base address + 3h	0, 0	BNDY
base address + 4h	0, 0	TPSR
base address + 5h	0, 0	TBCR0
base address + 6h	0, 0	TBCR1
base address + 7h	0, 0	ISR
base address + 8h	0, 0	RSAR0
base address + 9h	0, 0	RSAR1
base address + Ah	0, 0	RBCR0
base address + Bh	0, 0	RBCR1
base address + Ch	0, 0	RCR
base address + Dh	0, 0	TCR
base address + Eh	0, 0	DCR
base address + Fh	0, 0	IMR

Gate Array

Address Decode	Control Register Bits 3, 2	Assignments
base address	x, x	PSTR
base address + 401h	x, x	PSPR
base address + 402h	x, x	DQTR
base address + 403h	x, x	BCFR
base address + 404h	x, x	PCFR
base address + 405h	x, x	GACFR
base address + 406h	x, x	CTRL
base address + 407h	x, x	STREG
base address + 408h	x, x	IDCFR
base address + 409h	x, x	DAMSB
base address + 40Ah	x, x	DALSB
base address + 40Bh	x, x	VPTR2
base address + 40Ch	x, x	VPTR1
base address + 40Dh	x, x	VPTR0
base address + 40Eh	x, x	RFMSB
base address + 40Fh	x, x	RFLSB

x = don't care

Chapter 8: Memory Address Map

EPROM

Address	Type	Description
DC000 - DDFFF*	R	Estart Address
D8000 - D9FFF*	R	Estart Address
CC000 - CDFFF*	R	Estart Address
C8000 - C9FFF*	R	Estart Address
x1FFE - x1FFF**	R	Base Configuration Enable

* Address decoded in conjunction with PROM configuration jumpers and GA Config register bit 3.
** Reading the last 2 bytes of the PROOM address space returns the value of the GA Base Configuration register. Allows Startup PROM code to determine the adapters' Base Address.

x = selected address range (only values D or C are valid)

RAM

Address	Type	Description
DC000 - DDFFF	R	RAM Address
D8000 - D9FFF	R	RAM Address
CC000 - CDFFF	R	RAM Address
C8000 - C9FFF	R	RAM Address

One of these address ranges is selected in conjunction with PROM configuration jumpers and GA Configuration register bit 3.

Vector Pointer

The value loaded into these registers are compared against PC bus address bits 19-0. If a compare occurs DURING a memory read (-Smemr asserted), the memory-mapped mode is reset (GA Configuration bit 3). The suggested value for the vector pointer registers is the interrupt vector address issued during the "soft boot sequence" (Alt-Ctrl-Del).

Chapter 9: Jumper Position Equates

J1 Jumper Position	Memory Addresses
1	Disabled (Factory Setting)
2	DC000 - DDFFF
3	D8000 - D9FFF
4	CC000 - CDFFF
5	C8000 - C9FFF

J2 Jumper Position	Gate Array Addresses	I/O Base Addresses
1	300 - 30F (Factory Setting)	700 - 70F
2	310 - 31F	710 - 71F
3	330 - 33F	730 - 73F
4	350 - 35F	750 - 75F
5	250 - 25F	650 - 65F
6	280 - 28F	680 - 68F
7	2A0 - 2AF	6A0 - 6AF
8	2E0 - 2EF	6E0 - 6EF

Chapter 10: Gate Array Descriptions

Gate Array Register Definitions

Page Start Register

Base + 400h (read/write access)

The Page STart Register (PSTR) is an 8-bit read/write register. The value written into PSR determines the most significant byte of the starting address for receive packets in the local RAM. The data value loaded into PSTR MUST match the data value loaded into the Edlc Page Start register. The recommended value to load in the page start registers is 26h (h = hex). A page start register value of 26h segments the packet buffer into six pages (256 bytes/page) of transmit space for one maximum size Ethernet packet.

PSTR Bit	7	6	5	4	3	2	1	0
AdrMSB Bit	A15	A14	A13	A12	A11	A10	A9	A8

The value loaded in this register sets the beginning of the receive segment of the board resident packet buffer. Receive packets are placed in the packet buffer specified starting at the address specified by the value loaded in this register. The most significant address bit is A15 and the least significant is A8. See Recommended Adapter Memory Configuration on next page.

Page Stop Register

Base + 401h (read/write access)

The Page StoP Register (PSPR) is an 8-bit read/write register. The value written into PSPR determines the most significant byte of the ending address for receive packets in the local RAM. The data value loaded into PSPR MUST match the data value loaded into the Edlc Page Stop register. The recommended value to load in the page stop registers is 40h (h = hex). A page stop register value of 40h segments the packet buffer into 26 pages (256 bytes/page) of receive space for a variety of Ethernet packet sizes. The most significant address bit is A15 and the least significant is A8.

PSPR Bit	7	6	5	4	3	2	1	0
AdrMSB Bit	A15	A14	A13	A12	A11	A10	A9	A8

The value loaded in this register sets the ending address for the receive segment of the board resident packet buffer. Receive packets are placed in the packet buffer up to the address specified by the PSPR register. See Recommended Adapter Memory Configuration on next page.

Recommended Adapter Memory Configuration

With the standard 8K memory size, the adapter sees the memory starting at address 2000h so that an 8K page is 2000h-3FFFh. The recommended register values are:

NIC TPSR = 20h
NIC PSTART = 26h
NIC PSTOP = 40h
GA PageStart = 26h
GA PageStop = 40h

This yields the following partitioning of the buffer space:

Transmit buffer becomes: 2000h - 2600h = 256 x 6 = 1536 bytes
Receive buffer becomes: 2600h - 4000h = 256 x 26 = 6656 bytes

This allows transmit space for one full size 802.3 packet and receive space for four full size packets.

Drq Timer Register

Base + 402h (read/write access)

The DrQ Timer Register (DQTR) is an 8-bit read/write register. The value (binary weighted) written into DQTR determines the number of bytes to be transferred during a DMA burst. Transfer Time must be less than system refresh time. The recommended value to load into the DQTR is:

# of bytes transferred + transfer overhead = < 15uS (which is the system memory refresh rate)

The value loaded into the Drq Timer register MUST meet the following conditions:

1. Be a multiple of 4 (i.e.. 4, 8, 24, 32, etc.)

2. If the value is greater than or equal to 12, then bit 5 (16 byte select/double buffer select) in the Control register must be set.

DQTR Bit	7	6	5	4	3	2	1	0
Timer Bit	—	—	—	tb4	tb3	tb2	tb1	tb0

Bit	Symbol	Description
7	—	not used
6	—	not used
5	—	not used
4	tb4	timer bit 4 (MSB)
3	tb3	timer bit 3
2	tb2	timer bit 2
1	tb1	timer bit 1
0	tb0	timer bit 0 (LSB)

Base Configuration Register

Base + 403h (read only)

The Base ConFiguration Register (BCFR) is an 8-bit read-only register. The value written into BCFR during initialization (power up) determines the I/O base address of the adapter. To change the I/O base address loaded into the base configuration register requires you to power down the computer, change the position of the jumper (J2), and power up the computer.

BCFR Bit	7	6	5	4	3	2	1	0
I/O Base Address	300	310	330	350	250	280	2A0	2E0

Bit	Symbol	Description
7	300	i/o base address option 7 (default)
6	310	i/o base address option 6
5	330	i/o base address option 5
4	350	i/o base address option 4
3	250	i/o base address option 3
2	280	i/o base address option 2
1	2A0	i/o base address option 1
0	2E0	i/o base address option 0

EPROM Configuration Register

Base + 404h (read only)

The PROM ConFiguration Register (PCFR) is an 8-bit read-only register. The value written into PCFR during initialization (power up) determines the memory base address of the adapter. To change the memory base address loaded into the PROM configuration register requires you to power down of computer, change the position of the jumper (J1), and power up the computer.

PCFR Bit	7	6	5	4	3	2	1	0
Memory address	DCXXX	D8XXX	CCXXX	C8XXX	0	0	0	0

Bit	Symbol	Description
7	DCXXX	memory address option
6	D8XXX	memory address option
5	CCXXX	memory address option
4	C8XXX	memory address option
3	0	zero
2	0	zero
1	0	zero
0	0	zero

GA Configuration Register

Base + 405h (read/write)

The GA ConFiguration Register (GACFR) is an 8-bit read/write register. The majority of the bits in the register are initialized during the execution of a software initialization routine.

GACFR Bit	7	6	5	4	3	2	1	0
Configuration	nim	tcm	ows	test	rsel	mbs2	mbs1	mbs0

Bit	Symbol	Description
7	nim	(Nic Int Mask). A positive true signal used to block the propagation of interrupts from the LAN Controller to the interrupt controller (8259). LAN Controller generates interrupts for receive packets, transmit packets. 0 = pass interrupts (power up state) 1 = block interrupts
6	tcm	(Terminal Count Mask). A positive true signal used to block the setting of a DMA complete interrupt generated by the DMA controller in the gate array. The gate array generates an interrupt upon receiving a terminal count pulse from the PC bus interface during a DMA transfer. 0 = pass interrupts (power up state) 1 = block interrupts
5	ows	(0 Wait State). Used to allow the assertion of the zero wait state signal during I/O accesses to only registers in the gate array.
4	test	(TEST). For use ONLY by the gate array vendor. Setting this bit causes the counters in the gate array to count at a rate of 10Mbs.
3	rsel	(RAM Select). Used to map the packet buffer into system memory address space. The position of the J1 (memory) MUST be positioned in one of the four pre-determined address options (DC000, D8000, CC000, C8000) in conjunction with setting this bit to determine the memory address of the packet buffer.
2	mbs2	(Memory Bank Select 2).*
1	mbs1	(Memory Bank Select 1).*
0	mbs0	(Memory Bank Select 0).*

* See Memory Bank Select on next page.

The mbs bits are applicable ONLY during memory-mapped mode. The software is required to set the appropriate bits to select 8K segments of the packet buffer. Using the standard packet buffer configuration of 8K, the mbs0 bit is set. Using the 32K option, the 8K, 16K, 24K, 32K configurations are software selectable. Segments above 32K are not a selectable option.

Memory Bank Select

The mbs bits are applicable ONLY during memory-mapped mode. These three bits determine which 8K window of adapter memory appears in the host address space at the location selected by jumper1 on the card. The three bits, mbs 2-0 are equivalent to adapter address bits A15-13.

An adapter with a 32K memory* has four 8K windows that can be selected. A standard adapter with an 8K memory has only one possible window and, due to hardware considerations, this must be the second 8K starting at 2000h in the adapter memory space.

8K Configuration

Memory Access	msb2	msb1	msb0	Adapter Start Address
1st 8K	0	0	1	2000h

32K Configuration*

Memory Access	msb2	msb1	msb0	Adapter Start Address
1st 8K	0	0	0	0000h
2nd 16K	0	0	1	2000h
3rd 24K	0	1	0	4000h
4th 32K	0	1	1	6000h

* Adapters with 32K memory are not currently available from 3Com (December 1988.)

Control Register

Base + 406h (read/write access)

CR Bit	7	6	5	4	3	2	1	0
Control	start	ddir	bsel	share	eahi	ealo	xsel	rst

Bit	Symbol	Description
7	start	(START). Used to start the DMA controller in the gate array. Prior to setting this bit, the DMA address registers should point to the starting address of the packet buffer from which a data transfer will begin.
6	ddir	(DMA DIRection). Used to set the direction of the data transfer between the gate array and the PC bus interface. The setting of this bit may occur simultaneously with the Start bit. It is ILLEGAL to change the DMA Direction bit after the Start bit is programmed to the active state (a logical one). 0 = upload (packet buffer to system) 1 = download (system to packet buffer)
5	dbsel	(Double Buffer SELect). Used to connect the two 8-byte FIFOs in a serial configuration.
4	share	(interrupt SHARE). Used to select the interrupt sharing capability of the adapter. Interrupt sharing allows multiple adapters that have implemented interrupt sharing hardware to share a common interrupt channel to generate interrupts. Adapter configured for interrupt sharing CANNOT exist on the same channel with a non-sharing adapter. 0 = non-sharing interrupt channel 1 = sharing interrupt channel
3	eahi	(Ethernet Address HIgh) used to "window" the Ethernet Address PROM bytes 31-16 into the I/O base address. This bit is set active following a power-up condition or a software reset.
2	ealo	(Ethernet Address LOw). Used to "window" the Ethernet Address PROM bytes 15-0 into the I/O base address. Bytes 5-0 of the Ethernet address PROM contain the station address of the adapter.
1	xsel	(Xcvr Select). Used to select the transceiver type on the adapter. 1 = on-board transceiver (BNC), default 0 = external transceiver (DIX)
0	rst	(Software ReSeT). Used to emulate a power up reset. The reset initializes the gate array and the LAN Controller registers. 0 = software reset inactive 1 = software reset active

Status Register

Base + 407h (read only)

The Status Register (STREG) is an 8-bit, read-only register. The bits in this register provide information on the progress or completion status of the present operation.

STREG Bit	7	6	5	4	3	2	1	0
Operaton Status	dprdy	uflw	oflw	dtc	dip	rev2	rev1	rev0

Bit	Symbol	Description
7	dprdy	(Data Port ReaDY). This status bit indicates that the register files are ready for data transfer. During a download operation (system to adapter), the register files are "flushed" when the Start bit in the Control register is de-asserted by the software. During an upload operation, the register files end the data transfer by either receiving a terminal count (during DMA transfers) or when the Start bit in the Control register is de-asserted by the software. 0 = data NOT available 1 = data available
6	uflw	(UnderFLoW). Indicates that a read operation was issued to the register files when data was not available.
5	oflw	(OverFLoW). Indicates that a write operation was issued to the register files when all locations in the register files were full.
4	dtc	(DMA Terminal Count). Indicates that a terminal count (last byte transferred) was received from the PC bus interface during a DMA transfer.
3	dip	(DMA In Progress). Indicates that the DMA controller in the gate array is active. During a download operation, the register files are "flushed" (data moved from the register files to the packet buffer) when the Start bit in the Control register is programmed to a zero. The DMA In Progress bit remains active (a logical one) until the transfer is complete. During the flush operation, it is ILLEGAL to change the value in the DMA address registers.
2	rev2	(Gate array REVision bit 2)
1	rev1	(Gate array REVision bit 1)
0	rev0	(Gate array REVision bit 0) These bits indicate the revision level of the gate array. The revision level is implemented to track changes to the gate array.

Interrupt/DMA Configuration Register

Base + 408h (read/write)

The Interrupt/DMA Configuration (IDCFR) register is an 8-bit read/write register. The value loaded into IDCFR enables the gate array output driver(s) connected to the selected channel to be activated. The adapter's output driver will source/sink current from the assigned channel(s). The proper adapter configuration is to have only one interrupt channel and one DMA channel assigned to the adapter. The interrupt and/or the DMA channels should NOT be programmed if the functions are not expected to be utilized by the software. The bits in the register are powered up to the off state (no channels assigned to the adapter).

IDCFR Bit	7	6	5	4	3	2	1	0
assigned channel	irq5	irq4	irq3	irq2	—	drq3	drq2	drq1

Bit	Symbol	Description
7	irq5	(Interrupt ReQuest 5). Programming this bit to a logical one enables the gate array output driver to the IRQ5 channel on the PC bus. The IRQ5 bit on the PC bus is driven high, > 2.4V, to indicate an interrupt condition is present on the adapter or driven low, < .8V, indicating no interrupt condition on the adapter.
6	irq4	(Interrupt ReQuest 4). Programming this bit to a logical one enables the gate array output driver to the IRQ4 channel on the PC bus. The IRQ4 bit on the PC bus is driven high, > 2.4V, to indicate an interrupt condition is present on the adapter or driven low, < .8V, indicating no interrupt condition on the adapter.
5	irq3	(Interrupt ReQuest 3). Programming this bit to a logical one enables the gate array output driver to the IRQ3 channel on the PC bus. The IRQ3 bit on the PC bus is driven high, > 2.4V, to indicate an interrupt condition is present on the adapter or driven low, < .8V, indicating no interrupt condition on the adapter.
4	irq2	(Interrupt ReQuest 2). Programming this bit to a logical one enables the gate array output driver to the IRQ3 channel on the PC bus. The IRQ2 bit on the PC bus is driven high, > 2.4V, to indicate an interrupt condition is present on the adapter or driven low, < .8V, indicating no interrupt condition on the adapter.
3	——	Not used
2	drq3	(DMA ReQuest 3). Programming this bit to a logical one enables the gate array output driver to the DRQ3 channel on the PC bus. The DRQ3 bit on the PC bus is driven high, > 2.4V, to indicate a DMA request condition is present on the adapter or driven low, < .8V, indicating no DMA service is required on the adapter.

Interrupt/DMA Configuration Register Bits

Bit	Symbol	Description
1	drq2	(DMA ReQuest 2). Programming this bit to a logical one enables the gate array output driver to the DRQ2 channel on the PC bus. The DRQ2 bit on the PC bus is driven high, > 2.4V, to indicate a DMA request condition is present on the adapter or driven low, < .8V, indicating no DMA service is required on the adapter.
0	drq1	(DMA ReQuest 1). Programming this bit to a logical one enables the gate array output driver to the DRQ1 channel on the PC bus. The DRQ1 bit on the PC bus is driven high, > 2.4V, to indicate a DMA request condition is present on the adapter or driven low, < .8V, indicating no DMA service is required on the adapter.

DMA Address Register MSB

Base + 409h (read/write)

The DMA Address Register (DAMSB) is an 8-bit read/write register. The value written into DAMSB is used by the gate array DMA controller to address the packet buffer. The DAMSB register provides an address range from locations 256 - 65281. Using the standard packet buffer configuration of 8K, the DAMSB register MUST be loaded with an address that contains bit 15 inactive (a logical zero) and bit 13 active (a logical one). The minimum value loaded into DAMSB for an 8K packet buffer is 20h and the maximum value is 3Fh.

DAMSB Bit	7	6	5	4	3	2	1	0
Packet Buffer, MSB Bit	A15	A14	A13	A12	A11	A10	A9	A8

The value loaded into this register is used by the gate array DMA controller during data transfers to/ from the packet buffer.

DMA Address Register LSB

Base + 40Ah (read/write)

The DMA Address Register (DALSB) is an 8-bit read/write register. The value written into DALSB is used by the gate array DMA controller to address the packet buffer. The DALSB register provides an address range from locations 0 - 255. Using the standard packet buffer configuration of 8K, the DALSB register is loaded with a value of 00h - FFh.

DALSB Bit	7	6	5	4	3	2	1	0
Packet Buffer, LSB Bit	A7	A6	A5	A4	A3	A2	A1	A0

The value loaded into this register is used by the gate array DMA controller during data transfers to/ from the packet buffer.

Vector Pointer Register 2

Base + 40Bh (read/write)

The Vector Pointer Register 2 (VPTR2) is an 8-bit read/write register. The value written into VPTR2 is combined with the values loaded into Vector Pointer Registers 1 and 0 to form a 20-bit address comparator. The accumulated value in the vector pointer registers is compared against the address on the PC address bus signals SA19-0. Memory-mapped mode is reset on an address match between the vector pointer registers and the PC address bus signals SA19-0. The recommended address to load into the vector pointer registers is an address that is executed ONLY during a "soft boot" sequence (Alt-Ctrl-Del key function).

VPTR2 Bit	7	6	5	4	3	2	1	0
Vector Address Bit	A19	A18	A17	A16	A15	A14	A13	A12

The value loaded into this register combined the values loaded into Vector Pointer Registers 1 and 0 is used to reset the RAM Select (memory-mapped mode). RAM Select is bit mode). RAM Select is bit 3 in the GA Configuration register.

Vector Pointer Register 1

Base + 40Ch (read/write)

The Vector Pointer Register 1 (VPTR1) is an 8-bit read/write register. The value written into VPTR1 is combined with the values loaded into Vector Pointer Registers 2 and 0 to form a 20-bit address comparator. The accumulated value in the vector pointer registers is compared against the address on the PC address bus signals SA19-0. Memory-mapped mode is reset on an address match between the vector pointer registers and the PC address bus signals SA19-0. The recommended address to load into the vector pointer registers is an address that is executed ONLY during a "soft boot" sequence (Alt-Ctrl-Del key function).

VPTR1 Bit	7	6	5	4	3	2	1	0
Vector Address Bit	A11	A10	A9	A8	A7	A6	A5	A4

The value loaded into this register combined the values loaded into Vector Pointer Registers 2 & 0 are used to reset the RAM Select (memory-mapped mode). RAM Select is bit 3 in the GA Configuration register.

Vector Pointer Register 0

Base + 40Dh (read/write)

The Vector Pointer Register 0 (VPTR0) is an 4-bit read/write register. The value written into VPTR0 is combined with the values loaded into Vector Pointer Registers 2 and 1 to form a 20-bit address comparator. The accumulated value in the vector pointer registers is compared against the address on the PC address bus signals SA19-0. Memory-mapped mode is reset on an address match between the vector pointer registers and the PC address bus signals SA19-0. The recommended address to load into the vector pointer registers is an address that is executed ONLY during a "soft boot" sequence (Alt-Ctrl-Del key function).

VPTR1 Bit	7	6	5	4	3	2	1	0
Vector Address Bit	A3	A2	A1	A0	—	—	—	—

The value loaded into this register combined the values loaded into Vector Pointer Registers 2 and 1 is used to reset the RAM Select (memory-mapped mode). RAM Select is bit 3 in the GA Configuration register.

Register File Access MSB

Base + 40Eh (read/write)

The Register File (alias FIFO) Access MSB (RFMSB) is an 8-bit read/write register. Access to RFMSB during a download operation (system-to-adapter transfers) can be either a byte I/O write (outb) or a word I/O write (outw). Access to RFMSB during an upload operation (adapter-to-system transfers) can be either a byte I/O read (inb) or a word I/O read (inw). The register file access port is designed to temporarily store data from the PC data bus until the gate array DMA controller moves the stored data to the transmit segment of packet buffer during a download operation. The register file access port is used by the gate array DMA controller to temporarily store data from the receive segment of the packet buffer for transferring data to the PC data bus during an upload operation. Data written or read from RFMSB is invalid until the DMA controller is activated.

RFMSB Bit	7	6	5	4	3	2	1	0
Packet MSB Bit	D7	D6	D5	D4	D3	D2	D1	D0

This I/O port address is used to store MSB data for transmit packets and used to retrieve data from received packets.

Register File Access LSB

Base + 40F (read/write)

The Register File (alias FIFO) Access LSB (RFMSB) is an 8-bit read/write register. Access to RFLSB during a download operation (system-to-adapter transfers) can be ONLY a byte I/O write (outb). Access to RFLSB during an upload operation (adapter-to-system transfers) can be ONLY a byte I/O read (inb). The register file access port is designed to temporarily store data from the PC data bus until the gate array DMA controller moves the stored data to the transmit segment of packet buffer during a download operation. The register file access port is used by the gate array DMA controller to temporarily store data from the receive segment of the packet buffer for transferring data to the PC data bus during an upload operation. Data written or read from RFMSB is invalid until the DMA controller is activated.

RFMSB Bit	7	6	5	4	3	2	1	0
Packet LSB Bit	D7	D6	D5	D4	D3	D2	D1	D0

This I/O port address is used to store LSB data into for transmit packets and to retrieve data from received packets.

I/O Signal Names

I/O Pin Number	Signal Name
A12	PC Addr 19
A13	PC Addr 18
A14	PC Addr 17
A15	PC Addr 16
A16	PC Addr 15
A17	PC Addr 14
A18	PC Addr 13
A19	PC Addr 12
A20	PC Addr 11
A21	PC Addr 10
A22	PC Addr 9
A23	PC Addr 8
A24	PC Addr 7
A25	PC Addr 6
A26	PC Addr 5
A27	PC Addr 4
A28	PC Addr 3
A29	PC Addr 2
A30	PC Addr 1
A31	PC Addr 0
A02	PC Data 7
A03	PC Data 6
A04	PC Data 5
A05	PC Data 4
A06	PC Data 3
A07	PC Data 2
A08	PC Data 1

I/O Pin Number	Signal Name
A09	PC Data 0
B27	T/C
B12	Smemr
B11	Smemw
B08	OWS
A10	I/O ch rdy
A11	Aen
B02	Rst Drv
B23	Irq5
B24	Irq4
B25	Irq3
B04	Irq2
B16	Drq3
B06	Drq2
B18	Drq1
B15	Dack3
B26	Dack2
B17	Dack1
B14	Iord
B13	Iowr
B01	Gnd
B10	Gnd
B31	Gnd
B03	+5V
B29	+5V
B09	+12V

I/O Signal DC Parameters

PC Address bus

Signal Name	Pin Type	Iol	Ioh	Iil	Iih
PC Addr 19	In	-	-	10µa	1µa
PC Addr 18	In	-	-	10µa	1µa
PC Addr 17	In	-	-	10µa	1µa
PC Addr 16	In	-	-	10µa	1µa
PC Addr 15	In	-	-	10µa	1µa
PC Addr 14	In	-	-	10µa	1µa
PC Addr 13	In	-	-	10µa	1µa
PC Addr 12	In	-	-	10µa	1µa
PC Addr 11	In	-	-	10µa	1µa
PC Addr 10	In	-	-	10µa	1µa
PC Addr 9	In	-	-	10µa	1µa
PC Addr 8	In	-	-	10µa	1µa
PC Addr 7	In	-	-	10µa	1µa
PC Addr 6	In	-	-	10µa	1µa
PC Addr 5	In	-	-	10µa	1µa
PC Addr 4	In	-	-	10µa	1µa
PC Addr 3	In	-	-	10µa	1µa
PC Addr 2	In	-	-	10µa	1µa
PC Addr 1	In	-	-	10µa	1µa
PC Addr 0	In	-	-	10µa	1µa

PC Data bus

Signal Name	Pin Type	Iol	Ioh	Iil	Iih
PC Data 7	In/Out	16ma	16ma	10µa	1µa
PC Data 6	In/Out	16ma	16ma	10µa	1µa
PC Data 5	In/Out	16ma	16ma	10µa	1µa
PC Data 4	In/Out	16ma	16ma	10µa	1µa
PC Data 3	In/out	16ma	16ma	10µa	1µa
PC Data 2	In/Out	16ma	16ma	10µa	1µa
PC Data 1	In/Out	16ma	16ma	10µa	1µa
PC Data 0	In/Out	16ma	16ma	10µa	1µa

PC Control signals

Signal Name	Pin Type	Iol	Ioh	Iil	Iih
T/C	In	-	-	10µa	1µa
Smemr	In	-	-	10µa	1µa
Smemw	In	-	-	10µa	1µa
OWS	Out	24ma	24ma	-	-
I/O ch rdy	Out	8ma	8ma	-	-
Aen	In	-	-	10µa	1µa
Rst Drv	In	-	-	10µa	1µa
Irq5	In/Out	8ma	8ma	10µa	1µa
Irq4	In/Out	8ma	8ma	10µa	1µa
Irq3	In/Out	8ma	8ma	10µa	1µa
Irq2	In/Out	8ma	8ma	10µa	1µa
Drq3	Out	8ma	8ma	-	-
Drq2	Out	8ma	8ma	-	-
Drq1	Out	8ma	8ma	-	-
Dack3	In	-	-	10µa	1µa
Dack2	In	-	-	10µa	1µa
Dack1	In	-	-	10µa	1µa
Iord	In	-	-	10µa	1µa
Iowr	In	-	-	10µa	1µa

Chapter 11: Power Requirements

+ 5 Volts Current Requirements

The current consumption from the +5 volt power supply are given for the three active modes of the EtherLink II. The numbers are nominal values at ambient temperature.

Quiescent Mode

This mode is achieved after either a power-up reset or a software reset. In this mode, neither the DIX (15-pin D connector) or the BNC has an attachment cable connected.

Quiescent Mode Current Measurement

The current consumption from the +5 volt power supply during quiescent mode is 348.4 milliamps.

Receive Mode

This mode is defined as the operating state after an initialization and receive packet sequence has been executed. In this mode, the packet(s) is received through the BNC via the on-board transceiver with no attachment cable on the DIX (15-pin D connector). The BNC attachment is predominantly RG58 c/u coax.

Receive Mode Current Measurement

The current consumption from the +5 volt power supply during receive mode is 350.1 milliamps.

Transmit Mode

This mode is defined as the operating state after an initialization and transmit packet sequence has been executed. In this mode, the packet(s) are transmitted through the BNC via the on-board transceiver with no attachment cable on the DIX. The BNC attachment is predominantly RG58 c/u coax.

Transmit Mode Current Measurement

The current consumption from the +5 volt power supply during transmit mode is 352.3 milliamps.

+12 Vollts Current Requirements

The current consumption from the +12 volt power supply is given for the three active modes of the EtherLink II. The numbers are nominal values at ambient temperature.

Quiescent Mode 1

This mode is achieved after a power-up reset condition and the selection of an external transceiver mode. In this mode, neither the DIX (15-pin D connector) or the BNC has an attachment cable connected.

Quiescent Mode Current Measurement

The current consumption from the +12 volt power supply during quiescent mode is 27.4 milliamps.

Quiescent Mode 2

This mode is achieved after either a power-up reset or a software reset. In this mode, neither the DIX (15-pin D connector) or the BNC has an attachment cable connected.

Quiescent Mode Current Measurement

The current consumption from the +12 volt power supply during quiescent mode is 228.2 milliamps.

Receive Mode

This mode is defined as the operating state after an initialization and receive packet sequence has been executed. In this mode, the packet(s) is received through the BNC via the on-board transceiver with no attachment cable on the DIX (15-pin D connector). The attachment cable for the BNC is predominantly "thin net" coax (RG58 c/u).

Receive Mode Current Measurement

The current consumption from the +12 volt power supply during quiescent mode is 223.5 milliamps.

Transmit Mode

This mode is defined as the operating state after an initialization and transmit packet sequence has been executed. In this mode, the packet(s) are transmitted through the BNC via the on-board transceiver with no attachment cable on the DIX. The attachment cable for BNC is predominantly (RG58 c/u) coax.

Transmit Mode Current Measurement

The current consumption from the +12 volt power supply during quiescent mode is 251.6 milliamps.

Power Dissipation

The EtherLink II has achieved a total power dissipation that is the lowest in its adapter class. The total power dissipation is the summation of the power from both the +5 volts and the +12 volts.

+5 Volts Power Dissipation

The +5 volts power dissipation is the product of the voltage times the current:

voltage x current = Pd (in watts)
+5 volts x 352.3 ma = 1.7615 watts

+12 Volts Power Dissipation

The +12 volts power dissipation is the product of the voltage times the current:

voltage x current = Pd (in watts)
+12 volts x 251.6 ma = 3.0192 watts

Total Power Dissipation

The total power dissipation is the summation of the +5 volts and the +12 volts.

Summation:
+5 volts Pd + +12 volts Pd = total
1.7615 watts + 3.0192 watts = **4.7807 watts**

Chapter 12: Operational Specifications

The EtherLink II adapter was subjected to rigorous testing during design verification testing (dvt) to identify any anomalies or deficiencies within the design. The battery of tests included, but were not limited to, temperature cycling, electrostatic discharge, voltage variations, shock/vibration, RF emissions, safety, and drop testing. The results from the dvt testing showed that the EtherLink II adapter has no anomalies or deficiencies.

DC Voltage Margin

A portion of the dvt emulated the voltage variation that can occur during operation in a personal computer. The IBM power supply specification for the +5 volts is +5% and -4% which results in an output voltage swing of +4.8 to +5.25 volts. The specification also states the variation for the +12 volts is +5% and -4% netting an output variation of +11.40 to +12.60 volts.

+5 Voltage Margin

The EtherLink II adapter exhibited no abnormal operation during power supply variations of +/- 5%.

+12 Voltage Margin

The EtherLink II adapter exhibited no abnormal operation during power supply variations of +/- 5%.

Environment Characteristics

The EtherLink II adapter was subjected to numerous lab created environments to determine if any weakness existed in the construction of the product (mechanical assembly, solder joints, plating) or the materials used in the product. The following paragraphs provide information about the tests.

Shock & Vibration

Vibration testing conducted on EtherLink II adapters was performed on all three axis (x, y, z). The vibration consisted of three 12 minutes/cycles with 7 cycles applied, netting a total vibration time of 84 minutes. The vibration cycles used were:

- 5Hz - 7.6Hz at 0.4 inch double amplitude
- 7.6Hz - 500Hz - 7.6Hz at 1.5g
- 7.6Hz - 5Hz at 0.4 inch, double amplitude.

There was no external physical damage to the EtherLink II during this test.

Drop Tests

Packaged Drop Test:
Two fully functional EtherLink II adapters were packaged and subjected to free fall drop tests on all six faces and eight corners of the packaging. Each drop test was conducted a minimum of five times, at a height of 30 inches above a concrete surface. After the drop test, the adapters were retested and it was confirmed that the units remained fully functional.

Bench Drop Test:
Drop tests conducted on a bench top were performed on two unpackaged EtherLink II adapters. Two of the four edges of the adapter were elevated above the bench surface to an approximate height of 5 inches and released while maintaining contact of the opposite edge on the bench surface. The test was conducted 10 times on both the component and solder side.

Humidity

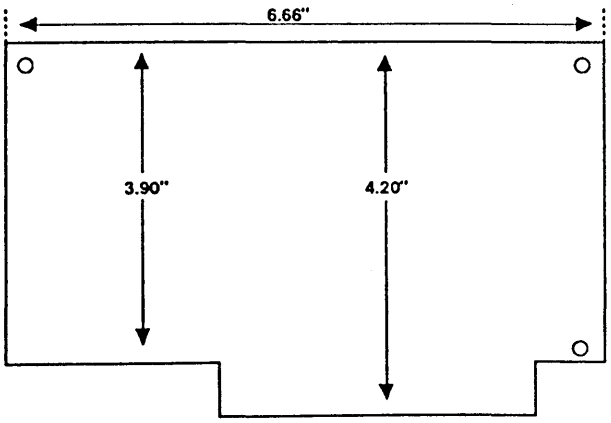
The temperature and humidity were varied during this testing. The temperature varied from -10 degrees Celsius to 65 degrees Celsius. The humidity varied up to 90%.

Salt Spray

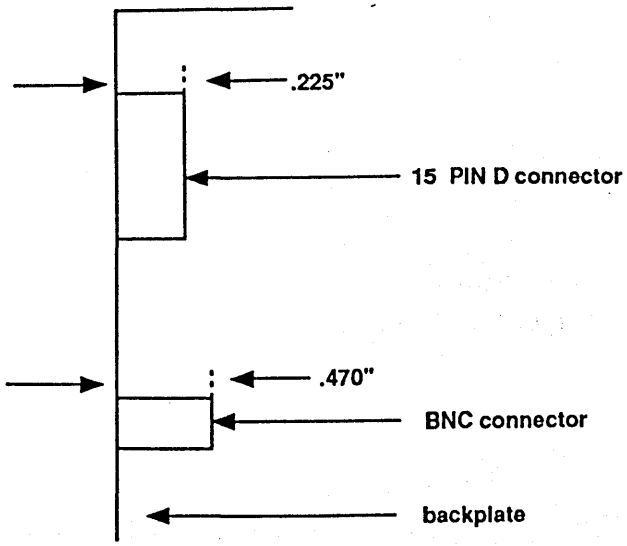
This test consisted of subjecting several EtherLink II adapters to a saline solution. The adapters were exposed to the solution for a period of time and then allowed to dry.

Chapter 13: Physical / Mechanical Specifications

Mechanical Specification



Backplate Mechanical Dimensions



Chapter 14: Agency Approvals

FCC Certification
Classification: Class B Computing Device Peripheral
Grant Date: July 10, 1987
FCC ID: DF67CC3C503

UL
Listed accessory. Complies with UL 478, Information Processing and Business Equipment.

VDE
Complies with RFI suppression requirements of Vfg 1046/1984.

Appendix A: Recommended Guidelines for Operation

OEM customers who want to develop software for the EtherLink II PC/AT bus Ethernet adapter are encouraged to make use of the Link Level Library (3L) sub-routines to accomplish all direct software interfacing to the proprietary ASIC, the programmable National Semiconductor Network Interface Controller (NIC), and the 8Kbyte packet buffer RAM which reside on the adapter board. For developers who want to write code that directly programs the adapter, the following guidelines are strongly recommended based on lab-based testing of several NIC-based Ethernet implementations.

The NIC anomalies have been identified by factory and lab system testing over an 18-month period. The NIC has been tested in a variety of networks with various traffic loading, network lengths, and network sizes. Due to this testing, a number of anomalies in the behavior of the NIC have been observed. Most of these are corrected in the current levels of NIC silicon or in surrounding hardware, both ASIC and discrete, provided on the EtherLink II. The several anomalies which remain are documented here, with software solutions described as implemented in the 3L implementation for the EtherLink II. The problems identified in this section were generated using artificially created environments and would not normally occur. Nonetheless, implementation of these software 'workarounds' is recommended for any application directly addressing the EtherLink II programming interface.

The following is provided as a supplement to the EtherLink II External Reference Specification and to the several National Semiconductor data sheets and addenda relating to the DP8390/8391/8393 chipset.

Three areas of anomalous behavior are discussed:

- Transmitter Deadlock
- Receiver Anomalies
- Operational Constraints

Transmitter Deadlock

Description

Upon occurrence of a particular timed asynchronous series of COLL, CRS, and CSNIC signals to the NIC, while the NIC is deferring transmission of a packet, the NIC transmitter may deadlock. The transmission will not be retried, and the NIC will remain in this transmission deferred state until stopped and restarted.

Solution

Transmissions should be executed with a watchdog timer in software which times out after a period sufficient to assure that the transmission, including the up to 15 retries specified by IEEE 802.3, has had time for normal completion. If the timer expires, stop the NIC and restart prior to retrying the same transmission. See below for the stop/restart procedure.

Receiver Anomalies

Receive status byte check for invalid data in packet

Description

Upon enabling the NIC or upon certain configuration changes of the NIC, a packet may be stored to RAM erroneously; all bytes have bits shifted from their original values. Such packets can always be identified by examination of the receive status byte in the packet header within the receive ring. The status byte will have one or both of the high order 2 bits set: DFR (deferring) and/or DIS (receiver disabled).

Solution

Status for received packets should always be checked, and packets received for which either the DFR or DIS bit is set should be discarded. The remaining header information (next page and size) will be valid and may be used to check for the next packet.

Bytes shifted within packet header

Description

In an extremely heavily loaded network environment, a packet may be received with invalidly formatted data. The receive status is lost, the next page byte occurs as the first byte, and the size is presented in the second and third bytes. The data in such packets may not be valid.

Solution

For each received packet, check whether the header has shifted. To determine if a shifted header has occurred, it is necessary and sufficient to verify the value of the next page byte, based on the word size in the packet header. The packet may be expected to consume a number of 256-byte pages of packet buffer space exactly equal either to the high-order size byte plus 1 or to the high-order size byte plus 2. The next page byte must therefore equal the starting page of the packet adjusted forward by one of these values, and adjusted for wraparound the receive ring. If either of these two adjusted next page values is equal to the next page byte in the second byte position in the packet header, the packet header is not shifted.

If the next page byte is shifted, the packet and any following packets already received must be discarded, as positioning of such packets in the ring can not reliably be determined, and data shifting may have occurred. The NIC should be stopped, the receive ring reset (NIC BNRY and CURR registers as well as program memory variables reinitialized), and the NIC restarted, as per the stop/restart procedures.

Bytes shifted within packet data

Description

If a packet is received while the NIC's internal FIFO is partway through prefetching data from a packet to be transmitted, it is possible for the received packet to be stored incorrectly in the packet buffer memory. Two types of mispositioning may occur, both involving the odd byte positioned bytes (1st, 3rd, 5th, ...) being shifted. In one case, the bytes are shifted backward 2-byte positions (data stored, in byte order is: 3rd, 2nd, 5th, 4th, 7th, 6th, 9th, 8th, ... from the actual packet). In the other case, the bytes are shifted forward 2-byte positions (data stored, in byte order is: xx, 2nd, 1st, 4th, 3rd, 6th, 5th, 8th, from the actual packet, where xx is a byte from the transmit prefetch). Packet header information is valid.

Solution

If a packet of either type is received, the packet should be discarded and processing may continue with the next packet received, whose position is accurately indicated by the packet header's next page byte. Discovery of such packets may be accomplished by protocol testing for specific fields' validity. We describe the solution used for MINDS/XNS protocols, which uses only the 802.3 source and destination fields for link-level checks. If receiving only packets with matching station address, the address is checked in software, and the packet discarded if it does not match. If broadcast addresses are also being accepted, a check follows for a valid broadcast address (ffh, ffh, ffh, ffh, ffh, ffh). Additional range checks for valid protocol control fields (IDP sockets and checksum, etc.) reduce the probability of "handing up" a corrupted packet from the protocols well below the probability of a undetected CRC error on the network.

Operational Constraints

Capability to use NIC Remote DMA facility not implemented

Description

Problems in early internal implementations of the remote DMA facility of the NIC, both in reliability and performance, indicated the inadvisability of its use. Reliability was improved in recent NIC implementations, but not proven in 3Com factory testing.

Solution

The EtherLink II provides this capability of adapter/host data transfer in three ways:

1. Interface to host based DMA channels 1, 2, and 3 under ASIC control.

2. Programmed I/O transfers through two ASIC ports allowing word-wide transfers.

3. Shared memory approach, mapping the packet buffer in high address systemmemory.

Required NIC programming sequences

Description

Documentation in the National Semiconductor data sheet for the DP8390 (NIC) does not take into account certain anomalous behaviors when starting up the NIC on active networks, when the receive buffer reaches overflow condition, when changing configuration, and when operating in loopback mode. The following procedures for programming the NIC in these situations are as implemented in the 3L for the EtherLink II, and correspond to recommendations from NSC included in several addenda to the NIC data sheet.

Solution

Starting the NIC:

a. NIC CR = 21h (stopped, addressing page 0 registers).
b. NIC DCR = 48h.
c. NIC RBCR0 = 0.
d. NIC RBCR1 = 0.
e. NIC RCR = {selected receive configuration}.
f. NIC TCR = 2 (loopback mode).
g. NIC BNRY = 3fh.
h. NIC PSTART = 26h.
i. NIC PSTOP = 40h.
j. NIC ISR = FFh (clearing ISR).
k. NIC IMR = {selected interrupts}.
l. NIC CR = 61h. (page 1 registers addressed).
m. NIC PAR0 - PAR5 = station address to be received.

n. NIC MAR0 - MAR7 = multicast bits desired.
o. NIC CURR = 26h.
p. NIC CR = 21h (return to page 0 registers).
q. NIC CR = 22h (start the NIC).
r. NIC TCR = 00h (exit loopback mode).

Stopping the NIC:
a. NIC CR = 21h.
b. poll NIC ISR until RST bit is set. NIC will not stop until completion of the current process (e.g., a reception or transmission of a packet).

Restart the NIC:
a. Execute the procedure to stop the NIC; proceed only after completion.
b. Execute the procedure to start the NIC. *

Receive buffer ring overflow:
a. Execute the procedure to stop the NIC; proceed only after completion.
b. Remove at least one packet from the receive ring.
c. Execute the procedure to start the NIC. *

* configuration registers and address registers other than the TCR do not require resetting.

Ninth byte corruption of transmitted packets

Description

A specifically timed sequence of signals CRS, COLL, and CSNIC to the NIC while the NIC is deferring a transmission may cause the NIC to corrupt the ninth byte in the transmit buffer. If a later attempt at retransmission of the packet is successful, the NIC will send the packet with the corrupted byte.

Solution

Protocol software must not depend on the validity of the source address field of the Ethernet layer packet encapsulation, or must check it against some higher level protocol field. Alternatively, software recheck of the Ethernet CRC will result in discard of such packets. In 3Com's MINDS/ XNS protocols, the field is not used; higher level protocols (e.g., IDP) have their own source address fields.

Reminder

The problems identified in this section were generated using artificially created environments with special packet blasters and test programs in laboratory networks. In working network environments, it is expected that these would not occur. Nonetheless, implementation of these software "workarounds" is recommended for any application directly addressing the EtherLink II programming interface. These workarounds are implemented as described in the 3L implementation for the EtherLink II.

Appendix B: Programming Sequences

This section provides an overview of the programming needed to initialize, receive, and transmit packets using an EtherLink II network adapter.

Initialization Programming Sequence

EtherLink II Gate Array Accesses (I/O base address + 400h)

1. Toggle Software Reset Bit (on/off)

   Set the Control Register to 03h (h=hex).

   Set the Control Register to 02h.

4. Set Control Register to 06h (enables Ethernet address prom).

5. Read I/O locations 300 - 305h (station address).

6. Set Control Register to 02h (for on-board transceiver operation) or 00h (for external transceiver operation).

7. Set Page Start Register to 26h. *

8. Set Page Stop Register to 40h. *

9. Set Interrupt/Dma Register

10. Set Drq Timer Register to 08h (8 byte burst transfers).

11. Set Dma Address MSB Register to 20h (allows 1.5K transmit buffer).

12. Set Dma Address LSB Register to 00h.

* Gate array and LAN Controller values must be the same.

EtherLink II LAN Controller (Jumpered I/O base address)

1. Set Command Register to 21h (h = hex)
2. Set Data Configuration Register to 48h
3. Set Transmit Configuration Register to 00h.
4. Set Receive Configuration Register to 20h (Monitor Mode)
5. Set Page Start Register to 26h. *
6. Set Page Stop Register to 40h. *
7. Set Boundary Register to 3Fh
8. Set Command Register to 61h (move to page 1 registers)
9. Set Current Page Register to 26h.
10. Set Six Physical Address Registers to the node's assigned address.
11. Set eight Multicast Address Registers to the assigned multicast address.
12. Set Command Register to 21h (back to page 0)
13. Reset Interrupt Status Register by writing FFh
14. Set Interrupt Mask Register to enable desired interrupts
15. Set Command Register to 22h (start mode)
16. Set Receive Configuration Register to desired packets.
17. Ready Host software to receive interrupts.

* Gate array and LAN Controller values must be the same.

Transmit Sequence

1. Load the Packet into the transmit buffer beginning on a 256-byte page boundary.
2. Set the Transmit Configuration Register to 00h (normal operation).
3. Set the Transmit Page Start Register to point to the beginning of the packet in the transmit buffer.
4. Load the Transmit Byte Count Registers (low and high) to the length of the packet to be transmitted.
5. Set the Interrupt Mask Register transmit interrupt bit(s), if any interrupt routine is to handle completed and/or errored transmissions.
6. Set the Command Register to 22h to start the Nic, if it is not already started. Then set the Command Register to 26h to initiate transmission of the packet.
7. Upon completion, status indications are available in the Interrupt Status Register (ISR) and the Transmit Status Register (TSR) and interrupts as programmed into the Interrupt Mask Register (IMR).

Receive Sequence

You can run the following sequence to receive packets after initialization.

1. If an interrupt routine is to handle completed and/or errored receptions, set the Interrupt Mask Register receive interrupt bit(s).
2. Set the Receive Configuration Register to receive the desired packet types only while the NIC is already started. NIC should be in Monitor mode when started. The NIC begins packet reception when it senses the first packet that has the appropriate destination address.
3. Status indications are available in the Interrupt Status Register (ISR), the Receive Status Register (RCR), and the first byte of the ring packet header, upon receipt of a packet.
4. You can set the Boundary Pointer Register to point to the last page of the finished packet when all data for the packet has been processed and the local memory containing a packet may be freed for additional packets. This keeps the boundary pointer one page behind the next packet.