

E T H E R L I N K P L U S

3 C 5 0 5

D E V E L O P E R ' S G U I D E

Revision 3.0

3Com Corporation

May 27, 1986

1569-02

PREFACE

This document is intended for use by sophisticated software engineers who will either be writing application software that will talk to the 3C505, or software that will actually reside on the card. The user is expected to have a strong background in microcomputer systems. It is recommended that the user browse through the Intel 80186 Data Sheet and the Intel Lan Components User's Manual before beginning (they are available through Intel).

The manual is divided into the following chapters:

- CHAPTER 1** **HARDWARE EXTERNAL REFERENCE SPECIFICATION (ERS)**
Provides a description of the 3C505 architecture, system resources and functional operation.

- CHAPTER 2** **HARDWARE INTERFACE SPECIFICATION**
Describes the programmable registers used to control, configure, and communicate with the 3C505.

- CHAPTER 3** **COMMAND INTERFACE SPECIFICATION**
Describes the function and use of the command level interface software supplied with the card.

- APPENDIX A** **80186 PERIPHERAL CONTROL BLOCK PROGRAMMING**
Provides the values used in the 3C505 firmware to configure the 80186 internal resources.

- APPENDIX B** **82586 CONFIGURATION**
Provides the values used by the 3C505 firmware to configure the 82586.

- APPENDIX C** **3C505 DIAGNOSTIC**
Describes the operation of the 3C505 diagnostic utility program.

- APPENDIX D** **3D DEBUGGER**
Describes a host program that uses a special debug mode of the 3C505 to assist in debugging programs running on the card.

- APPENDIX E** **3C505 DEVELOPER'S SOFTWARE DISKETTE**
Describes the contents of the diskette that accompanies the developer's kit.

- APPENDIX F** **REVISION 2.0 ROM**
Describes changes made in Revision 2.0 ROM code.

- APPENDIX G** **REVISION 3.0 ROM**
Describes changes made in Revision 3.0 ROM code.

TABLE OF CONTENTS

Page #

CHAPTER 1 - HARDWARE EXTERNAL REFERENCE SPECIFICATION

| | | |
|--------|-----------------------------|----|
| 1.0 | Introduction | 1 |
| 1.1 | Resources | 1 |
| 1.2 | Architecture | 1 |
| 1.3 | Address Maps | 3 |
| 1.3.1 | Adapter I/O map | 3 |
| 1.3.2 | Adapter memory map | 3 |
| 1.3.3 | Host I/O map | 3 |
| 1.4 | 80186 Microprocessor | 4 |
| 1.5 | 82586 Ethernet Coprocessor | 4 |
| 1.6 | Network Interface | 5 |
| 1.6.1 | 82586 Serial Interface | 5 |
| 1.6.2 | 8023 Manchester Converter | 5 |
| 1.6.3 | Transceiver | 5 |
| 1.7 | Adapter Firmware ROM | 6 |
| 1.8 | Adapter RAM | 6 |
| 1.9 | Host-Adapter Interface | 9 |
| 1.9.1 | Command Register | 9 |
| 1.9.2 | Data Register | 9 |
| 1.9.3 | Data Register Configuration | 10 |
| 1.9.4 | DMA Transfers | 11 |
| 1.9.5 | Status Flags | 12 |
| 1.10 | Adapter Interrupts | 13 |
| 1.10.1 | Internal Interrupts | 13 |
| 1.10.2 | External Interrupts | 13 |
| 1.11 | Host Interrupts | 14 |
| 1.12 | Resetting the Adapter | 15 |
| 1.13 | Ethernet Address | 15 |
| 1.14 | LED Indicators | 15 |
| 1.15 | Host ROM | 16 |

CHAPTER 2 - HARDWARE INTERFACE SPECIFICATION

| | | |
|-----|--------------------------|----|
| 2.0 | Introduction | 17 |
| 2.1 | Command Register | 18 |
| 2.2 | Data Register | 18 |
| 2.3 | Host Control Register | 19 |
| 2.4 | Host Status Register | 21 |
| 2.5 | Host Aux DMA Register | 23 |
| 2.6 | Adapter Control Register | 23 |
| 2.7 | Adapter Status Register | 25 |

CHAPTER 3 - COMMAND INTERFACE SPECIFICATION

| | | |
|-------|-------------------------------------|----|
| 3.0 | Introduction | 28 |
| 3.1 | Primary Command Block Structure | 28 |
| 3.1.1 | Status Flag Usage for PCB Transfer | 30 |
| 3.1.2 | Host to Adapter Request | 30 |
| 3.1.3 | Adapter to Host Request or Response | 31 |
| 3.2 | PCB Commands | 32 |
| 3.2.1 | Host to 3C505 PCB Formats | 32 |
| 3.2.2 | 3C505 to Host PCB Formats | 37 |
| 3.3 | System ROM Utilities | 41 |
| 3.3.1 | Host I/O Support: INT 80H | 41 |
| 3.3.2 | Network I/O Support: INT 81H | 44 |
| 3.3.3 | Configuration/Status: INT 82H | 45 |
| 3.3.4 | Timer Support: INT 83H | 47 |
| 3.3.5 | Download Program Support: INT 84H | 48 |
| 3.3.6 | PCB Command Processor: INT 85H | 49 |
| 3.3.7 | Packet Processor Vector: INT 86H | 50 |
| 3.3.8 | Idle Vector: INT 87H | 51 |
| 3.3.9 | PCB Enqueue Vector: INT 88H | 51 |

LIST OF FIGURES

| | | |
|-------------|---------------------|---|
| Figure 1-1: | 3C505 Block Diagram | 2 |
| Figure 1-2: | DRAM Refresh | 8 |

| | |
|---|----|
| APPENDIX A - 80186 PERIPHERAL CONTROL BLOCK PROGRAMMING | 52 |
| APPENDIX B - 82586 CONFIGURATION | 54 |
| APPENDIX C - 3C505 DIAGNOSTIC | 55 |
| APPENDIX D - 3D DEBUGGER | 60 |
| APPENDIX E - 3C505 DEVELOPER'S SOFTWARE DISKETTE | 65 |
| APPENDIX F - REVISION 2.0 ROM | 68 |
| APPENDIX G - REVISION 3.0 ROM | 72 |

CHAPTER 1

HARDWARE EXTERNAL REFERENCE SPECIFICATION

1.0 INTRODUCTION

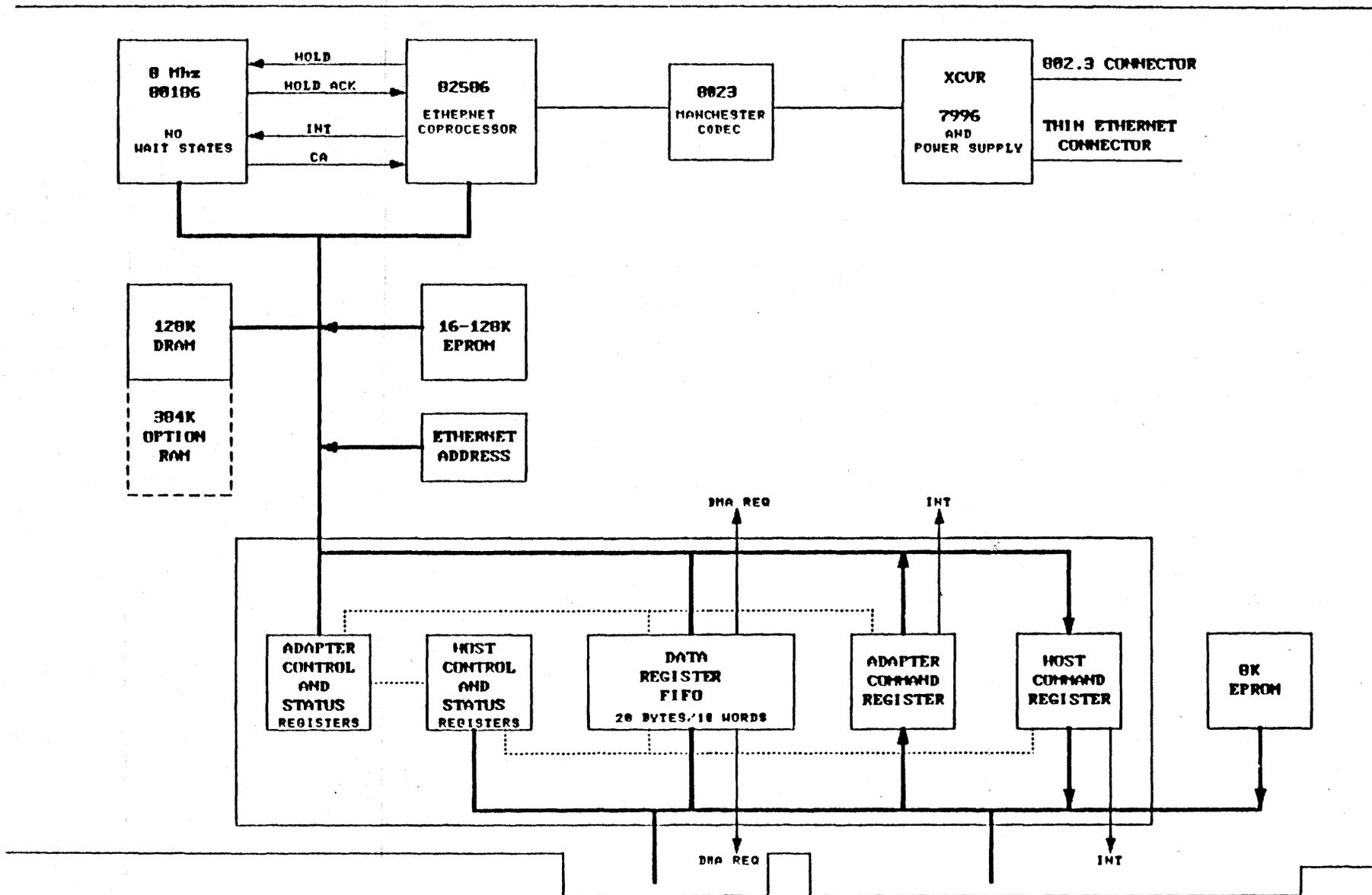
The EtherLink Plus, 3C505, is a high performance Ethernet adapter for IBM PCs and compatibles. It consists of an 80186 16 bit microprocessor, an 82586 Ethernet coprocessor, up to 512K bytes of user RAM, a high speed 16 bit host interface, and a highly integrated on-board transceiver. The 3C505 is particularly well-suited for server and high performance workstation applications.

1.1 RESOURCES

- * 8 Mhz 80186 16 bit microprocessor - no wait states
- * 82586 multi-packet buffer Ethernet coprocessor
- * 16K to 128K bytes EPROM
- * 128K to 512K bytes packet buffer/program memory
- * 8/16 bit host interface - PIO or DMA
- * 20 byte FIFO to maximize host/adaptor data transfer
- * On-board "Thin Ethernet" transceiver/802.3 connector
- * 8K bytes host EPROM

1.2 ARCHITECTURE

The 3C505 is a 16 bit microcomputer with a high performance Ethernet I/O channel and an IBM PC AT interface. The 16K bytes of on-board firmware contain software that supports initialization, program download, and diagnostic software. The 128K RAM, expandable to 512K, allows for protocol processing as well as offloading of application programs from the host PC. The 82586 performs all Data Link functions, as well as powerful network diagnostics. It performs all packet buffer management functions and, in a typical environment, will not "drop" packets. The host interface supports high speed, 8 or 16 bit, DMA transfers as well as programmed I/O. The interface is very flexible, yet simple, allowing for easy programming. The 8 Mhz allows ample processing power, even on a heavily loaded network.



2

FIGURE 1-1
3C505 BLOCK DIAGRAM

8 OR 16 BIT DATA PATH
HOST INTERFACE

1.3 ADDRESS MAPS

1.3.1 ADAPTER I/O MAP

| <u>HEX Address</u> | <u>Byte/Word</u> | <u>Description</u> |
|--------------------|------------------|----------------------------|
| 0 | NA (see text) | 82586 CA |
| 100 | Low Byte | Adapter Command Register |
| 102 | Low Byte Read | Adapter Control Register |
| 102 | High Byte Read | Adapter Status Register |
| 102 | High Byte Write | Adapter Control Register |
| 104 | Word | Data Register |
| 180-18F | Low Byte | Ethernet address (6 Bytes) |
| FF00-FFFF | Word | Peripheral Control Block |

1.3.2 ADAPTER MEMORY MAP

| <u>HEX Address</u> | <u>Description</u> |
|--------------------|--|
| 00000-1FFFF | 128K Bytes system RAM: Bank 1 |
| 20000-3FFFF | 128K Bytes option RAM: Bank 2 |
| 40000-5FFFF | 128K Bytes option RAM: Bank 3 |
| 60000-7FFFF | 128K Bytes option RAM: Bank 4 |
| FC000-FFFFF | 16K Bytes system ROM * |
| E0000-FFFFF | 128K Bytes system ROM (If 27512s are installed) |

* Address lines A20-A23 of the 82586 are ignored and the Initialization Root is located at FFFF6 in system ROM.

1.3.3 HOST I/O MAP

| <u>HEX Address (factory set) **</u> | <u>Description</u> |
|-------------------------------------|-----------------------|
| Base address + 0 (300) | Host Command Register |
| Base address + 2 (302) Read | Host Status Register |
| Base Address + 2 (302) Write | Host Control Register |
| Base address + 4 (304) | Data Register *** |
| Base address + 6 (306) Read | Host Control Register |

** The address is given as an offset from the I/O base address which is set using the I/O address jumpers on the card. The factory set base address is 300H.

*** The Data Register is a byte wide register in an 8 bit slot (PC, XT, or AT) and word wide in a 16 bit slot (AT).

1.4 80186 MICROPROCESSOR

The 3C505 uses the Intel 80186 Microprocessor. This is a highly integrated 16 bit processor with 3 timers, 2 DMA channels, and an interrupt controller on chip. The 80186 is software compatible with the 8086.

The 80186 timing is generated by a 16Mhz crystal. An internal divider generates an 8Mhz clock output which is used for system timing. All 80186 bus cycles are 4 clock cycles long, or 500 nanoseconds, with a system memory bandwidth of 2 Megawords per second. All DMA transfers require 2 bus cycles, or 1 microsecond.

1.5 82586 ETHERNET COPROCESSOR

The 82586 is a high performance, intelligent communications processor responsible for all network related tasks, including frame reception and transmission, error logging, and diagnostics.

The 82586 has two interfaces: a parallel system bus interface to communicate with the 80186 and to retrieve and store packet data in system RAM; and a serial interface to transmit and receive data from the network. The serial interface is described in section 1.6, Network Interface. The 82586 bus interface operates from the 8 Mhz system clock and all bus cycles are 500 nanoseconds.

The 80186 and the 82586 operate in a shared bus configuration using the HOLD/HOLDA protocol. This configuration is described in detail in the Intel Lan Components User's Manual. In this mode, only one of the processors can use the system bus at a time. All interprocessor communications are via the system RAM. The 80186 can initiate a transaction by asserting the CA (Channel Attention) input to the 82586. A read or write to I/O location 00 will cause an active transition on the CA input. The 82586 initiates a transaction by asserting the 80186 INT1 input.

The 82586 can require the bus to access system RAM in three instances:

1. To read or update the SCB (System Control Block).
2. To transmit a packet.
3. To receive a packet.

When receiving or transmitting, the 82586 uses approximately 35% of the system bandwidth, or 715 KW/second. Thus program execution and DMA transfers, although slowed, do not halt.

The Adapter (80186) can reset the 82586 by asserting the R586 bit in the Adapter Control Register. The 82586 remains in the reset state until this bit is cleared.

1.6 NETWORK INTERFACE

The 3C505 network interface consists of the serial interface on the Intel 82586 LAN controller, the SEEQ 8023 Manchester Code Converter, and an on-board transceiver using the AMD 7996 Transceiver IC.

1.6.1 82586 SERIAL INTERFACE

The 82586 performs all parallel to serial and serial to parallel conversion during packet transmission and reception. During transmit, parallel data is retrieved from the Adapter RAM through the 82586 bus interface. The 82586 serializes the data, inserts the preamble, source and destination fields, appends a CRC field to the "packet", and outputs the bit stream. The 82586 also performs the CSMA/CD link management algorithm according to the IEEE 802.3 standard. During reception, the 82586 strips off the preamble and compares the destination address field with the station address to see if the frame should be received. If so, the serial bit stream is converted into bytes and stored in the Adapter RAM.

1.6.2 8023 MANCHESTER CONVERTER

The 8023 is responsible for the Manchester encoding and decoding of the serial bit stream between the 82586 and the transceiver. It also supplies the transmit and receive clocks to the 82586 serial interface. A watchdog timer on the IC prevents continuous transmission of more than 25 milliseconds, thus limiting the maximum packet size to approximately 31k bytes.

For diagnostic purposes, the 8023 can be placed in "loopback mode" whereby the transmitted data is internally routed to the receive section. This is useful for isolating transceiver problems. Enable loopback by clearing the Loopback bit in the Adapter Control Register. Refer to the 3C505 Hardware Interface Specification, Chapter 2.

1.6.3 TRANSCEIVER

The 3C505 onboard transceiver physically connects the 3C505 to the "Thin Ethernet" coax cable. It performs the necessary signal conditioning as well as collision detection.

The user can also connect the 3C505 to a thick Ethernet network through an external transceiver such as the 3Com 3C102. If so, the user must disable the onboard transceiver and enable the 15 pin connector on the backplate of the card. To do so, the transceiver select plug in the card must be moved from the BNC position to the DIX position. The EtherLink Plus Installation Guide, included with the 3C505, illustrates this procedure.

1.7 ADAPTER FIRMWARE ROM

The 3C505 contains 16K bytes of firmware contained in two 8Kx8, 2764 type ROMs. These ROMs can be replaced by 27128, 27256, or 27512 ROMs for up to 128k bytes of firmware. The ROMs must have a maximum address access time of 250 nanoseconds or less.

The 3C505 ROM firmware performs self-test, initialization and configuration, and DRAM refresh. It also provides, through a command block interface, a set of functions which support Host/Adapter I/O, network interfacing and execution of downloaded programs. Refer to the 3C505 Command Interface Specification, Chapter 3, for more details.

The system ROM is mapped to address space FC000H-FFFFFH (E0000H-FFFFFH if 128k bytes are used) and is accessible to both the 80186 and 82586. The 82586 only accesses ROM following an 82586 reset to fetch the initialization root.

1.8 ADAPTER RAM

The 3C505 contains 128K bytes of dynamic memory organized in a 64K x 16 configuration. Three additional 128K banks can be installed for 256K, 384K, or 512K bytes of RAM memory. Each bank consists of four 64K x 4 DRAMs. The first additional bank must be installed in socketed locations U31, U33, U35, and U37. The second additional bank must be soldered into locations U40, U42, U44, and U46. The third additional bank must be soldered into locations U41, U43, U45, and U47. These devices must have a maximum RAS access time of 150 nanoseconds and maximum CAS access time of 75 nanoseconds. In addition, these RAMs must support "CAS before RAS refresh", described below. These parts are currently available from Hitachi, NEC, Fujitsu, Intel, Texas Instruments, and Mitsubishi.

The system RAM is accessible to the 80186 and 82586 and is used for both packet buffering and program storage. No physical partitioning or protection mechanism is used. The RAM is mapped into the Adapter memory space 0-7FFFFH, with bank 1 occupying 0-1FFFFH, bank 2 occupying 20000-3FFFFH, bank 3 occupying 40000-5FFFFH, and bank 4 occupying 60000-7FFFFH.

Software must perform two functions for proper RAM operation: initialization and refresh. To refresh the RAM, 256 consecutive locations in each bank must be accessed every 4 milliseconds. Data loss will occur if refresh is not performed. The initialization procedure depends on the refresh technique used.

To facilitate refresh, the 3C505 contains hardware which utilizes the "CAS before RAS" refresh feature of the DRAMs. In this mode, the RAMs generate the refresh address internally after each CAS before RAS cycle, and the internal address counter increments so that the next CAS before RAS cycle will refresh the next address. A read or write to I/O location 80H will produce a CAS before RAS cycle in all banks simultaneously. The 80186 PCS1 Peripheral Chip Select output is programmed for this range. NOTE: A CAS before RAS cycle, read or write, does not corrupt RAM data.

To increase reliability and to free the 80186 from involvement in RAM refresh, the 3C505 firmware uses 80186 Timer 2 and DMA Channel 0 to automatically generate refresh cycles. The timer causes a DMA cycle to occur every 30 microseconds. Each DMA cycle performs an I/O read and write to location 80H. Thus each DMA cycle refreshes two memory locations. The DMA controller is not programmed to "stop on terminal count" so that refresh, once initialized, will continue without any CPU involvement. Using this technique, refresh consumes 3.3% of the memory bandwidth. The timer generated DMA will only produce one DMA cycle so that burst mode refresh cannot be used.

Upon power-up, the 80186 must wait 200 microseconds and then perform 8 RAM "initialization" cycles. If CAS before RAS refresh is to be used, then 8 refresh cycles (a read or write to I/O location 80H) will initialize all RAM. If CAS before RAS refresh is not used, then 8 reads or writes to any location in each of the installed banks of memory will initialize the RAM.

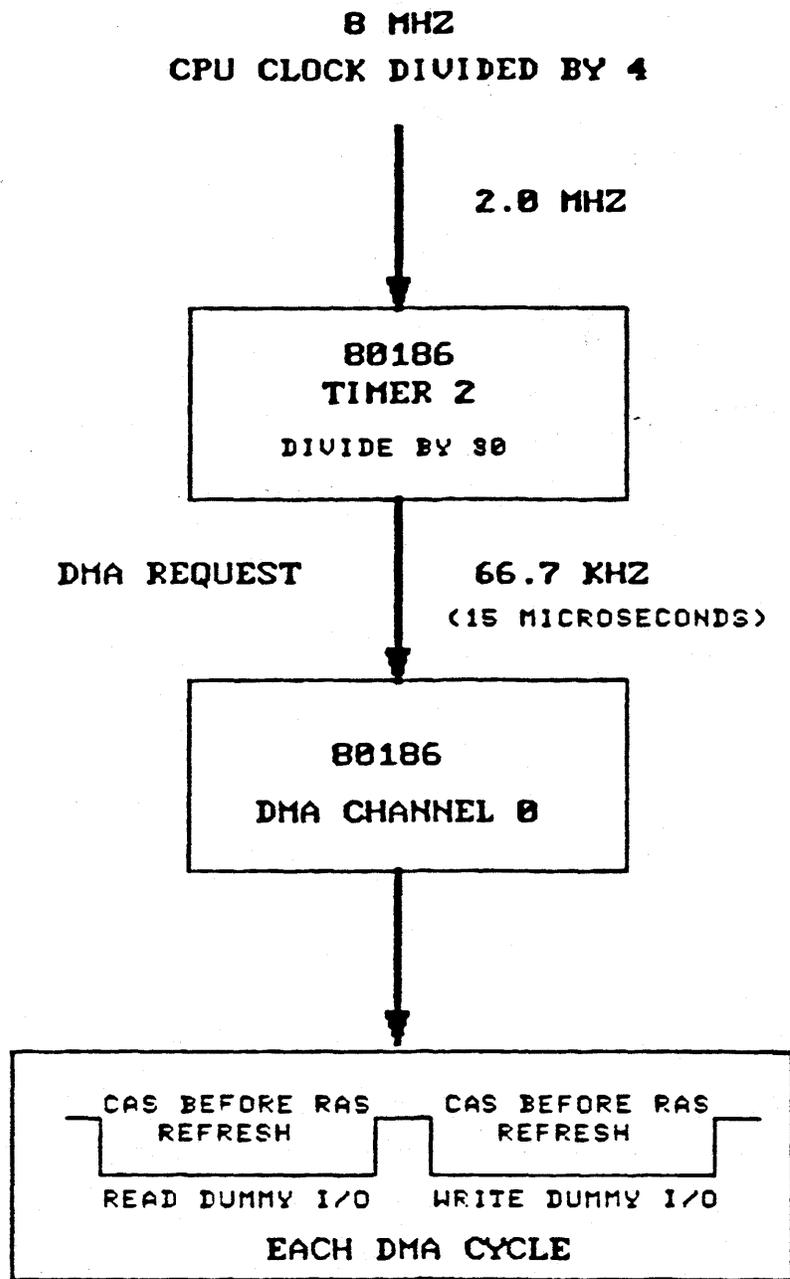


FIGURE 1-2
3C505 DRAM REFRESH

1.9 HOST-ADAPTER INTERFACE

The Host and the Adapter communicate through two I/O mapped registers: the Command Register and the Data Register. In addition, each side has a Control Register and a Status Register where are used for transfer handshaking and interface configuring. A detailed bit level description of these registers is found in the 3C505 Hardware Interface Specification, Chapter 2. The interface requires 16 locations in the Host I/O address space. Jumpers are used to position the base address.

1.9.1 COMMAND REGISTER

The Command Register is a full duplex byte-wide register used to transfer commands and small amounts of data between the Host and the Adapter. The register can be polled using the Command Register Empty (ACRE and HCRE) and Command Register Full (ACRF and HCRF) bits in the Host and Adapter Status Registers. Alternately, the Command Register can be interrupt driven, so that an interrupt is generated to the Host or Adapter when the opposing side has loaded a byte into the Command Register. Refer to Section 1.10 on interrupts for more information.

1.9.2 DATA REGISTER

The Data Register is a half duplex 20 byte FIFO designed for high speed bulk data transfers between the Host and the Adapter. The direction of the data transfer is controlled by the DIR bit in the Host Control Register. If the DIR bit is cleared (0), data transfer is from the Host to the Adapter, which is referred to as a data download. If the DIR bit is set (1), data transfer is from the Adapter to the Host and referred to an upload. The state of the DIR bit can be read in both the Host and Adapter Status Registers.

The Data Register supports both polled I/O and DMA data transfers. In polled operation, the state of the Data Register can be determined by reading the Data Register Ready bit (HRDY and ARDY) in the Host and Adapter Status registers. The meaning of the Ready Bit is determined by its state and the state of the DIR Bit.

| TRANSFER | DIR | HRDY | ARDY | DESCRIPTION |
|-----------------|-----|--------|--------|---|
| PIO DOWNLOAD | 0 | 1 X | X 1 | REGISTER NOT FULL REGISTER NOT EMPTY |
| PIO UPLOAD | 1 | X 1 | 1 X | REGISTER NOT FULL REGISTER NOT EMPTY |

REGISTER NOT FULL ---> WRITE DATA
REGISTER NOT EMPTY ---> READ DATA

To clear a stuck byte from the Data Register, or to ensure that the register is in a known empty state, the FLSH (Flush) bit in the Host and Adapter Control Register is used. By setting and resetting the FLSH Bit, the Data Register Ready Flag is forced to the empty state (the data in the FIFO is not actually cleared). Either the Host or the Adapter can use this bit, regardless of the state of the DIR Bit.

Careful attention should be paid in the use of the DIR bit. Incorrect and confusing results occur if the bit is not set correctly. The DIR must be in its correct state prior to enabling DMA transfers. When changing the state of the DIR bit from download to upload, the Host must make sure that the Adapter has actually completed the download, i.e., the FIFO is empty. One solution is to change the DIR bit only as part of the command block sequence. The 3C505 firmware changes the DIR bit after the Adapter has accepted the first word of a command block. This indicates that the Adapter has completed execution of the last command block.

1.9.3 DATA REGISTER CONFIGURATION

To the Adapter, the Data Register is always a 16 bit wide FIFO, 10 words deep. Only 16 bit data transfers are permitted (A0 and BHE are ignored). However, to the Host, the Data Register is configured as either an 8 bit FIFO, 20 bytes deep, or a 16 bit FIFO, 10 words deep, depending on where it is installed. The register is automatically configured and no jumpers need be set. Also, the Adapter does not need to know whether it is installed in an 8 or a 16 bit slot.

The Data Register is configured as a 16 bit register when installed in a 16 bit I/O slot of an AT. Only word transfers are permitted (A0 and BHE are ignored) and only 16 bit AT DMA channels (5,6,7) can be used.

In a PC, XT, or an 8 bit slot of an AT, the Data Register is configured as a 20 byte FIFO to the Host. The register performs byte to word conversion so that the 80186 always performs word I/O to the Data Register and Adapter performance is not reduced in 8 bit systems. The Host must always transfer an even number of bytes to the register; the Adapter Data Register Ready flag (ARDY) indicates the presence of words, not bytes. An odd byte will get "stuck" in the register because the Adapter will not know of its presence. In Adapter to Host transfers, word to byte conversion is performed. A byte cannot get stuck in this direction because the Host Data Register Ready flag (HRDY) indicates the presence of bytes.

1.9.4 DMA TRANSFER

DMA transfers by the Host to and from the Data Register are enabled using the DMAE bit in the Host Control Register. Since the DMA channel floats when this bit is cleared, caution should be taken to ensure that this channel in the PC DMA controller is not enabled until the DMAE bit is set. When the DMAE bit is cleared, another I/O card may use the same DMA channel.

| TRANSFER | DIR | HRDY | ARDY | DESCRIPTION |
|-----------------|-----|------|------|--------------------------|
| DMA DOWNLOAD | 0 | 1 | X | WRITE REQUEST TO HOST |
| | | X | 1 | READ REQUEST TO ADAPTER |
| DMA UPLOAD | 1 | X | 1 | WRITE REQUEST TO ADAPTER |
| | | 1 | X | READ REQUEST TO HOST |

The 3C505 can be programmed to generate an interrupt to the Host after the last cycle of a Host DMA transfer using the TCEN bit in the Host Control Register. Refer to the section on interrupts for more information.

The Adapter performs DMA transfers to and from the Data Register on 80186 DMA Channel 1. Both the DMA Channel Enable and the DMA DONE interrupt are controlled by programming registers internal to the 80186. The DMA Channel 1 input to the 80186 is never in a floating state.

Note: the Adapter and the Host may perform DMA transfers independent of one another. That is, one may use polled I/O while the other performs DMA. There is little reason for the Adapter to use polled I/O and the 3C505 firmware always uses DMA.

The 3C505 contains hardware to support Host "demand mode" DMA transfers in PCs where this mode is supported. If the Burst (BRST) bit in the Aux DMA Register is not set, the 3C505 will transfer 9 bytes/words and then relinquish the DMA channel for one host CPU cycle. This will allow the Host to refresh its own system DRAM if necessary. The 3C505 will then transfer another 9 bytes/words, and so on. If the Burst bit is set, this pause will not occur. The Burst bit has no effect if single cycle DMA is used. Thus, if the DMAE bit is set, the DMA request input to the host PC will go inactive under the following conditions:

1. The entire Host DMA transfer is completed
2. The Data Register FIFO is temporarily full/empty depending on the transfer direction.
3. The Burst bit is not set and 9 DMA transfers have occurred since the last DMA pause.

1.9.5 STATUS FLAGS

The Host and Adapter also communicate using general purpose Status Flags. The Adapter has three flags, ASF1, ASF2, and ASF3, which are programmed by the Adapter Control Register and directly observable by the Host Status Register. The Host has two Status Flags, HSF1 and HSF2, which are programmed by the Host Control Register and observable through the Adapter Status Register. The Status Flags are used for synchronization, command execution and completion codes, and other assorted tasks. They are not decoded by the hardware in any way.

1.10 ADAPTER (80186) INTERRUPTS

The 80186 microprocessor in the 3C505 may be interrupted by both internal and external sources.

1.10.1 INTERNAL INTERRUPTS

A brief description of how the 3C505 firmware uses internal interrupts follows. These include Timer, DMA, and software generated interrupts. Refer to the 80186 Data Sheet for programming details.

DMA Channel 1 Done Interrupt

This is used to generate an interrupt after the last cycle of a DMA transfer to or from the Data Register.

Timer Interrupt

An interrupt is generated every 10 millisecond from Timer 0. The interrupt is used for general purpose counting and timeouts.

1.10.2 EXTERNAL INTERRUPTS

There are three sources of external interrupts to the 80186: Command Register Full (INT 0), 82586 Int (INT1), and Attention (NMI). Since each interrupt has a unique channel, there is no need for a corresponding status bit to indicate the cause of the interrupt. Except for NMI, which cannot be disabled, the interrupts are enabled or disabled by setting the appropriate bit in the Interrupt Mask Register in the 80186. These inputs never "float" so that these channels can be enabled at any time. All channels are programmed positive edge triggered.

Command Register Full (INT0)

If enabled, an interrupt will be generated to the 80186 when the Host loads a byte in the command register. This condition is also reflected by the HCRF (Host Command Register Full) bit in the adapter status register. The Command Register Full interrupt and status bit are cleared when the 80186 reads the byte from the Command Register

82586 INT (INT1)

This input is tied directly to the INT output on the 82586. If enabled, the 80186 will be interrupted by the 82586 after the SCB has been modified by the 82586. Refer to the 82586 data sheet for more information.

Attention (NMI)

When the ATTN bit is set in the Host Control Register, an NMI is generated in the 80186. This NMI is used as a "soft" reset to bring the adapter back to a known state after an interface error occurs. The NMI is positive edge triggered and the ATTN bit must be brought from low to high to force the NMI.

1.11 HOST INTERRUPTS

The 3C505 can be programmed to interrupt the Host in two situations: DMA complete and Command Register Full. Only one PC interrupt channel is used.

Host DMA Done

By setting the TCEN (Terminal Count Enable) bit in the Host Control Register, an interrupt will be generated to the Host after the last cycle of a DMA transfer to or from the Data Register. If the Command Register Full interrupt is also enabled, the Done bit in the host status register should be used to determine if a DMA Done was the source of the interrupt. The DMA Done interrupt and Done status bit are cleared by disabling the DMA channel using the DMAE (DMA enable) bit in the Host Control Register.

Command Register Full

By setting the CMDE (Command Enable) bit in the Host Control Register, an interrupt will be generated to the Host when the Adapter writes a byte in the Command Register. If the DMA Done interrupt is also enabled, the ACRF (Adapter Command Register Full) bit in the Host Status Register should be used to determine the source of the interrupt. The Command Register Full interrupt and ACRF bit are cleared when the Host reads the byte from the Command Register.

Care must be taken when enabling and disabling the 3C505 interrupts. If both interrupt sources are disabled, the interrupt channel is floated and can cause spurious interrupts if the PC PIC channel is not also disabled. To prevent this, always turn the PIC channel off before disabling 3C505 interrupts, and enable 3C505 interrupts before enabling the PIC channel. When both 3C505 interrupts are disabled, the interrupt channel can be used by another I/O card.

When installed in a PC, XT, or 8-bit AT slot, interrupt channels 3,4,5,6,7 or 9 should be used. In this situation, channel 9 is equivalent to channel 2. In an a 16-bit AT slot, any interrupt channel can be used.

1.12 RESETTING THE ADAPTER

Power On Reset

Upon power up, the 3C505 is put in reset state. Both the 80186 and 82586 are reset, the Command and Data Registers status indicate empty, and both the Host and Adapter Control Registers are cleared.

Adapter Reset

The Host can reset the adapter by simultaneously setting both the ATTN bit and the FLSH bit in the Host Control Register. This reset is similar to the power on reset except that the Host Control Register is not affected. The Adapter will remain reset until the ATTN and FLSH bits are reset.

Note: After either of the above "hard" resets, the adapter firmware performs configuration and self-test routines which last several seconds. The completion of these tasks is indicated by a transition in the Host Status Flags from state 3 to state 0. Visually, this is indicated by LED #1 turning off.

1.13 ETHERNET ADDRESS

The Adapter Ethernet Station address resides in a PROM in the Adapter I/O space. The twelve digits are contained in the low byte of the six consecutive words starting at location 180H.

1.14 LED INDICATORS

The Adapter contains two LEDs which are enabled by the LED1 and LED2 bits in the Adapter Control Register. The LEDs are active high so that setting the bit turns the LED on and clearing the bit turns the LED off.

LED #1

The 3C505 firmware turns this LED on during the self test and initialization following a hard reset. The LED is turned off at the conclusion of these routines. Application software may call 3C505 firmware routines to use the LED for debug and status indications.

LED #2

The 3C505 firmware turns this LED on or off at approximately a 1 HZ rate. It serves as a "heartbeat" signal and is a visual indicator that the card is alive. If the LED should stop blinking, a software or hardware error has occurred. It is not recommended that application software use this LED.

CAUTION! When using 3C505 firmware, downloaded software must control the LEDs by calls to the firmware routines provided in ROM. Otherwise incorrect operation will result.

1.15 HOST ROM

A socket is provided on the card for an 8K x 8 (2764) ROM which resides in the Host memory space. This ROM can be used for applications such as BIOS extensions. The maximum address access time for these devices must not be greater than 250 nanoseconds. The ROM is only accessible to the Host. Note - The PC AT will execute from 8 bit ROMs on I/O cards.

The ROM can be mapped on any 8K boundary in the Host address space. The base address for the ROM is programmed using the memory address jumpers on the card. The 3C505 does not support DMA access to this ROM. Incorrect data will be read.

To enable the ROM, set the Enable jumper on the card must to the ON (0) position. If no ROM is present, or to disable a ROM which is present, place the jumper should in the off (1) position.

CHAPTER 2

HARDWARE INTERFACE SPECIFICATION

2.0 INTRODUCTION

The 3C505 Hardware Interface Specification describes in detail the 3C505 interface registers accessible by the PC host and the 3C505 processor.

Briefly, the Host and 3C505 communicate using four registers: Command, Data, Status, and Control. The Command Register is full duplex and used for command block transfers. The Data Register is a half duplex, 16-bit wide FIFO, and can be used with a DMA channel for efficient bulk data transfer. The Control Register allows programmed configuration of the interface. The Status Register contains interface state flags and programmable flags. The Host and Adapter access these registers in their I/O space relative to a base I/O address:

| | Base Offsets | | Access |
|---------|--------------|---------|-------------|
| | Host | Adapter | |
| Command | 0 | 0 | Read/Write |
| Data | 4 | 4 | Read/Write |
| Status | 2 | 3 | Read only |
| Control | 6 | 3 | Write only |
| Control | 6 | 2 | Read only * |
| AUX DMA | 2 | X | Write only |

The Host base I/O address can be modified with jumpers, while the Adapter base address is fixed at 100 hex. Refer to Section 1.3, Address Maps, for more detail.

Refer to the 3C505 Hardware External Reference Specification in Chapter 1 for a more detailed explanation of the hardware architecture. And also refer to the 3C505 Command Interface Specification in Chapter 3 for a description of how this interface can be programmed.

* The Host and Adapter will read the contents of their own Control Registers.

2.1 COMMAND REGISTER

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| CMD7 | CMD6 | CMD5 | CMD4 | CMD3 | CMD2 | CMD1 | CMD0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

The Command Register (CMDR) is a bidirectional 8-bit data register used for passage of primary command blocks between the Host and the 3C505. Programmed and interrupt driven I/O can be used to read/write this register; DMA is not supported.

2.2 DATA REGISTER

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| DR15 | DR14 | DR13 | ....  .... | DR2 | DR1 | DR0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

The Data Register (DR) is a half duplex, 20 byte FIFO used for high speed data transfers. DMA or programmed I/O methods can be used to read/write this register; interrupt driven I/O is not supported. From an 8-bit Host, the Data Register appears as an 8-bit wide register. Only an even number of bytes can be transferred. To the Adapter or to a 16-bit Host, the register appears as a 16-bit wide register (10 words deep) and only word transfers are supported.

2.3 HOST CONTROL REGISTER

The Host Control Register (HCR) is an 8-bit register used by the Host to cause 3C505 hard or soft resets, to control interrupt and DMA requests to the Host, and to provide synchronization control signals between the PC Host and 3C505 processors. The contents of this register can be read back by the Host. This register is cleared upon power-up.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| ATTN | FLSH | DMAE | DIR  | TCEN | CMDE | HSF2 | HSF1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

ATTN **Attention**

When the Host sets ATTN, a non-maskable interrupt (NMI) is generated to the Adapter's 80186 processor. The Host Control and Status Registers on the Adapter are not affected. The interpretation of NMI is intended to be "soft reset", where the Adapter resets itself into an idle state ready to accept commands.

FLSH **Flush Data Register**

Setting the FLSH bit flushes all data words from the Data Register regardless of the state of the DIR (direction) bit. The FIFO assumes an empty condition, although the actual data in the FIFO is unchanged. The Data Register remains in this state until the FLSH bit is cleared.

******** **Reset adapter**

When the Host simultaneously sets both ATTN and FLSH, the adapter hardware decodes it as a "hard reset". The Data Register, Adapter Status and Control Registers, and the Host Status Register are reset. A reset signal to the 80186 processor is generated which resets all 80186 internal registers and transfers control to the powerup reset location. The 82586 is also reset. The Adapter will stay in this reset state until the ATTN and FLSH bits are cleared.

- DMAE** **DMA enable**
Used in conjunction with the DIR bit, DMAE enables DMA transfers to or from the Data Register. DMA requests to the Host can occur only if this bit is set. With the DMAE bit cleared, the DMA request output to the Host "floats" and another I/O card may use the channel.
- DIR** **Direction flag**
The Host has exclusive control of the direction of the half-duplex Data Register. If DIR is clear, data transfers are to the Adapter (download). If DIR is set, data transfers are to the Host (upload).
- CAUTION!** After completing a download, the Host must make sure that the Adapter has completed its transfer (FIFO empty) before changing the DIR bit to the upload state. This can take 1 to 30 microseconds, depending on the network activity occurring on the Adapter.
- TCEN** **Terminal Count interrupt enable**
TCEN enables an interrupt to the Host at the completion of a DMA transfer to or from the Data Register.
- CMDE** **Command Register interrupt enable**
The CMDE control bit allows the Host to be interrupted when the Adapter has written the Command Register. When neither TCEN nor CMDE are set, the Host should disable the interrupt channel because the interrupt request line will float.
- HSF1** **Host Status Flags 1 and 2**
HSF2 The HSF1 and HSF2 status bits are routed directly to the Adapter Status Register. They are general purpose in nature and can be used by Host and Adapter interface drivers to synchronize data transfer or pass command completion status.

2.4 HOST STATUS REGISTER

The Host Status Register (HSR) is an 8-bit register used by the Host to determine causes of interrupts, check status of both Data and Command Register programmed I/O, and provide a way to synchronize the Host and 3C505 processors.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| HRDY | HCRE | ACRF | DIR  | DONE | ASF3 | ASF2 | ASF1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

HRDY Data Register ready

The HRDY bit indicates whether the Data Register is not full or not empty, depending on the Direction Flag. When the Host is downloading data to the Adapter, HRDY set means that the Data Register is not full, i.e., ready for more data. When the Host is uploading data from the Adapter, HRDY set means that the Data Register is not empty, i.e., input data is available.

HCRE Host Command Register empty

The HCRE flag is used to handshake data transfer through the Command Register from the Host to the Adapter. When the Host writes the Command Register, HCRE is cleared indicating the register is not empty. When the Adapter has read the Command Register, HCRE is set, indicating that the register is empty.

ACRF Adapter Command Register full

The ACRF flag is used to handshake data transfer through the Command Register from the Adapter to the Host. When the Adapter writes the Command Register, ACRF is set, indicating the register is full. When the Host reads the Command Register, ACRF is cleared, indicating that the register is not full.

DIR **Direction flag**
The DIR status bit is the current value of the DIR control bit in the Host Control Register. It specifies in which direction data is allowed to pass through the Data Register. When DIR is clear, transfers are from the Host to the Adapter (download). When DIR is set, transfers are from the the Adapter to the Host (upload). The DIR bit also determines how HRDY should be interpreted.

DONE **DMA done**
The DONE flag is set when a DMA transfer between the Host and the Data Register is complete. An interrupt to the Host will also be generated if the TCEN bit in the Host Control Register is set. The DONE bit is cleared by clearing the DMAE bit in the Host Control Register.

ASF1 **Adapter Status Flags 1, 2 & 3**
ASF2 The ASF1, 2 and 3 status bits are routed directly to the
ASF3 Host Status Register from the Adapter Control Register. They are general purpose in nature and can be used by Host and Adapter interface drivers to synchronize data transfer or pass command completion status.

CAUTION! These bits are set asynchronously with respect to the Host processor and it is possible to read these bits while they are in transition. This is only a problem if the state of more than one flag is tested simultaneously. For example, if the present state is ASF1 = ASF2 = 0 and you are testing for state ASF1=1 and ASF2=0, you could actually read this state during a state transition to ASF1 = ASF2 =1, if the ASF2 flag changed state slower than the ASF1 flag. The solution is to read the Adapter Status Register twice when checking the state of more than one flag to insure that you have not read a flag in transition.

2.5 HOST AUX DMA REGISTER

The Host Aux DMA Register is used to support demand mode DMA transfers. This register is cleared upon power-up.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0   | 0   | 0   | 0   | 0   | 0   | 0   | BRST |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

BRST DMA Burst

If the Burst bit is not set, demand mode DMA transfers by the Host will pause every 9 transfers to allow the PC to refresh its dynamic RAMs. If the Burst bit is set, no such pause will occur. This bit has no effect during single cycle DMA transfers.

CAUTION! Do not use demand mode DMA uploads in PC's or XT type PC's. Data errors will occur!

2.6 ADAPTER CONTROL REGISTER

The Adapter Control Register (ACR) is an 8-bit register used by the Adapter to reset the 82586, flush the Data Register, blink the LEDs, and set the state of synchronization flags between the PC Host and 3C505 processor. The contents of this register can be read back by the Adapter. This register is cleared upon power-up.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| LPBK | FLSH | R586 | LED2 | LED1 | ASF3 | ASF2 | ASF1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

LPBK Loopback control

LPBK specifies a diagnostic mode in which transmitted data is not placed on the network, but is looped back into the Adapter. This controls loopback at the 8023 Manchester Code Converter. If CLEAR, loopback mode is enabled.

FLSH **Flush Data Register**
Setting the FLSH bit flushes all data words from the Data Register regardless of the state of the DIR (direction) bit. The FIFO assumes an empty condition, although the actual data in the FIFO is unchanged. The Data Register remains in this state until the FLSH bit is cleared.

R586 **Reset 82586**
When the Adapter sets R586, a hardware reset is applied to the 82586 coprocessor chip. All major 82586 hardware components are reset to an inactive state and remain reset until R586 is cleared. The 82586 then waits for the Channel Attention signal before completing initialization.

LED2 **LED control bit 2**
LED2 determines the state of LED 2. Setting the bit turns the LED on, and clearing the bit turns the LED off.

LED1 **LED control bit 1**
LED1 determines the state of LED 1. Setting the bit turns the LED on, and clearing the bit turns the LED off.

ASF1 **Adapter Status Flags 1, 2 and 3**
ASF2 The ASF1, 2 and 3 status bits are routed directly to the
ASF3 Host Status Register. They are general purpose in nature
and can be used by Host and Adapter interface drivers to
synchronize data transfer or pass command completion
status.

2.7 ADAPTER STATUS REGISTER

The Adapter Status Register (ASR) is an 8-bit register used by the Adapter to determine causes of interrupts, check status of both Data and Command Register programmed I/O, and provide a way to synchronize the Host and 3C505 processors.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| ARDY | ACRE | HCRF | DIR  | 8/16 | SWTC | HSF2 | HSF1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

ARDY Data Register ready

The ARDY bit indicates whether the Data Register is not full or not empty, depending on the Direction Flag. When the Host is downloading data to the Adapter, ARDY set means that the Data Register is not empty, i.e., input data is available. When the Adapter is uploading data to the Host, ARDY set means that the Data Register is not full, i.e., ready to accept more data.

ACRE Adapter Command Register empty

The ACRE flag is used to handshake data transfer through the Command Register from the Adapter to the Host. When the Adapter writes the Command Register, ACRE is cleared, indicating that the register is not empty. When the Host reads the Command Register, ACRE is set, indicating that the register is empty.

HCRF Host Command Register full

The HCRF flag is used to handshake data transfer through the Command Register from the Host to the Adapter. When the Host writes the Command Register, HCRF is set, indicating the register is full. When the Adapter reads the Command Register, HCRF is cleared, indicating that the register is not full.

DIR **Direction flag**
The DIR status bit specifies in which direction data is allowed to pass through the Data Register. The direction is settable only by the Host using the DIR bit in the Host Control Register. When DIR is clear, transfers are from the Host to the Adapter. When DIR is set, transfers are from the Adapter to the Host.

8/16 **8/16 bit**
The 8/16 bit flag indicates whether the Adapter is installed in an 8 or 16 bit expansion slot. If the 8/16 bit is set, the Adapter is in a sixteen bit slot, i.e., an IBM AT or AT-compatible.

SWTC **External switch**
The SWTC flag in the Adapter Status Register represents the state of the TEST jumper on the Adapter. When the TEST jumper is set to one, the Revision 3.0 ROM code will:

1. Ignore powerup memory test error. Memory errors detected during powerup normally prevent the Adapter from entering the main ROM idle loop. Ignoring errors is useful when using ICE systems that need to modify the NMI vector location in order to operate.
2. Ignore ROM checksum error. During ROM development, it is convenient not to checksum since the code is changing frequently.
3. Install 3D interrupt vectors. The interrupt vectors known as "exceptions" (basically INT 0 to 7) and all unused interrupt vectors are made to point to the 3D slave in the Revision 2.0 ROM. When an exception occurs, 3D becomes active and attempts to communicate with the 3D Debugger program.

HSF1 Host Status Flags 1 and 2

HSF2 The HSF1 and HSF2 status bits are routed directly from the Host Control Register. They are general purpose in nature and can be used by Host and Adapter interface drivers to synchronize data transfer or pass command completion status.

CAUTION! These bits are set asynchronously with respect to the 80186 and it is possible to read these bits while they are in transition. This is only a problem if the state of both flags is tested simultaneously. For example, if the present state is HSF1 = HSF2 = 0 and we are testing for state HSF1=1 and HSF2=0, we could actually read this state during a state transition to HSF1 = HSF2 =1, if the HSF2 flag changed state slower than the HSF1 flag. The solution is to read the Host Status Register twice when checking the state of both flags to insure that you have not read a flag in transition.

CHAPTER 3

COMMAND INTERFACE SPECIFICATION

3.0 INTRODUCTION

The 16K bytes of EPROM on the 3C505 Adapter contain firmware that support the following:

- * Bootup initialization and diagnostics
- * Software memory refresh
- * Network I/O
- * Packet buffer control
- * Host I/O
- * System timer
- * Host PC primary command interface

After Adapter bootup initialization, host-based application programs or drivers can access the network or resources of the Adapter through the primary command block interface described in Sections 3.1 and 3.2. Programs downloaded into the Adapter can access the Adapter resources directly or through a set of utilities described in Section 3.3.

3.1 PRIMARY COMMAND BLOCK STRUCTURE

The 3C505 firmware idles waiting for a Primary Command Block (PCB) from the PC Host. The PCB structure is expected during command/response sequences. The format of a PCB is:

PCB command code (byte)
PCB data length (byte)
PCB data (variable length)

The PCB is passed using programmed I/O through the Command Register. An example PCB might contain an 82586 configuration command, a length field that counts the number of bytes in the data field, and a data field that has configuration data needed to set up the 82586 coprocessor.

The maximum PCB size the Adapter can accept in this version ROM is 64 bytes. The PCB data length field does not include the PCB command code or the length field itself. The maximum data field is 62 bytes long. The valid PCB command codes are summarized in Table 1 and are explained in detail in Section 3.2.

TABLE 1 : PCB COMMAND CODE SUMMARY

HOST -> 3C505 COMMANDS

| | | |
|-----|---------------------------|-----------------------------------|
| 00: | | |
| 01: | configure adapter memory | (set adapter buffer requirements) |
| 02: | configure 82586 | (set 82586 receive mode) |
| 03: | Ethernet address | (get adapter Ethernet address) |
| 04: | download data to 3C505 | (download using 3C505 DMA) |
| 05: | upload data to Host | (upload to Host using 3C505 DMA) |
| 06: | download data to 3C505 | (download using 3C505 PIO) |
| 07: | upload data to Host | (upload to Host using 3C505 PIO) |
| 08: | receive packet | (receive a packet) |
| 09: | transmit packet | (transmit a packet) |
| 0a: | network statistics | (includes 82586 error counts) |
| 0b: | load multicast list | (perform 82586 MC-setup command) |
| 0c: | clear downloaded programs | (release download program memory) |
| 0d: | download program | (download program to 3C505) |
| 0e: | execute program | (execute program in 3C505) |
| 0f: | self-test | (perform 3C505 self-test) |
| 10: | set Ethernet address | (set Ethernet address in 82586) |
| 11: | adapter info | (get adapter information) |
| 12: | reserved | |
| : | | |
| 2f: | reserved | |

3C505 -> HOST

| | | |
|-----|-----------------------------|------------------------------------|
| 30: | | |
| 31: | configure adapter memory | (returns success or failure) |
| 32: | configure 82586 response | (returns success or failure) |
| 33: | address response | (returns Ethernet address) |
| 34: | download data request | (request DMA download to 3C505) |
| 35: | upload data request | (request DMA upload to Host) |
| 36: | n/a | |
| 37: | n/a | |
| 38: | receive packet complete | (receive packet request complete) |
| 39: | transmit packet complete | (transmit packet request complete) |
| 3a: | network statistics response | (returns network statistics) |
| 3b: | load multicast response | (returns success or failure) |
| 3c: | clear program response | (returns success or failure) |
| 3d: | download program response | (returns program id) |
| 3e: | execute response | (returns variable length data) |
| 3f: | self-test response | (returns self-test results) |
| 40: | set address response | (returns success or failure) |
| 41: | adapter info response | (returns adapter information) |
| 42: | reserved | |
| : | | |
| 5f: | reserved | |

3.1.1 STATUS FLAG USAGE FOR PCB TRANSFER

The Adapter uses a 64-byte circular buffer to store the host byte stream sent through the Command Register. For protection against stray bytes (from Host aborted PCB transfers), the Adapter does not consider a PCB transfer complete until the Host Status Flags (HSF2 and HSF1) go to state 11. Simultaneously, the TOTAL length of the PCB should be in the Command Register so the true beginning of PCB can be calculated. (This last total length is NOT included in the PCB data length field.) The Adapter uses its status flags (ASF2 and ASF1) similarly to signal "end of PCB" when sending a PCB to the Host.

The Adapter is always ready to read a PCB but it might not always be able to accept it. To indicate the acceptance of the PCB, the Adapter uses status flag state 01 after the Host signals end-of-PCB. To indicate rejection, the Adapter uses status flag state 10. When the Adapter sends a PCB to the Host, it expects the Host to set its status flags similarly to signal acceptance or rejection.

In summary, the Adapter uses and expects the Host to use the following conventions:

Adapter or Host Status Flag

| SF2 | SF1 | |
|-----|-----|--------------|
| 0 | 0 | Undefined |
| 0 | 1 | PCB accepted |
| 1 | 0 | PCB rejected |
| 1 | 1 | End of PCB |

The state 11 is accompanied by the total length of the PCB just transmitted. After a PCB is received, the state 01 or 10 is used to signal acceptance or rejection of the PCB.

3.1.2 HOST TO ADAPTER REQUEST

The following method is suggested to transmit a Host PCB to the 3C505 Adapter:

- Load the PCB command byte into the Command Register; this will interrupt the 3C505 Adapter, synchronizing it to the PC Host for the data transfer.

- Poll the Command Register Empty (HCRE) flag in the Host Status Register. Abort the I/O if it does not go empty within 40 ms.

- Output the remainder of the PCB similarly, reducing the timeout period to 500us. The Adapter remains in interrupt context to read PCB data.

- After the last actual PCB data byte is transferred, the Host must send one last byte signifying the TOTAL length of the PCB (excluding this byte). Set the Host status flags to state 11 before writing the length.

- Wait for adapter state 01 (accept) or 10 (reject). Assume a reject if a 50ms timeout occurs.

3.1.3 ADAPTER TO HOST REQUEST OR RESPONSE

The 3C505 Adapter to PC Host request is made when the Adapter needs to read or write a block of host memory. The Adapter usually sends a response PCB after it has executed a host request.

The following method is used to transmit an adapter PCB to the Host:

- Load the PCB command byte into the Command Register; this interrupts the PC Host, synchronizing it to the Adapter for the data transfer.

- Poll the Command Register Empty (ACRE) flag in the Adapter Status Register. Abort the I/O if it does not go empty within 20 ms.

- Output the remainder of the PCB similarly, reducing the timeout to 500us. The Host should remain in interrupt context to read PCB data.

- After the last actual PCB data byte is transferred, the Adapter sends one last byte signifying the TOTAL length of the PCB. The Adapter Status Flags are set to state 11 before writing the length.

- The Adapter waits for Host Status Flag state 01 (accept) or 10 (reject).

3.2 PCB COMMANDS

3.2.1 HOST TO 3C505 PCB FORMATS

01H: Configure Adapter Memory. The Adapter allocates memory for the PCB command queue, receive command queue, multicast address list, 82586 frame descriptors, receive buffers, and download program data structures. Each PCB and receive command queue entry is large enough to buffer the maximum size PCB of 64 bytes. A multicast list is kept in adapter memory to be loaded into the 82586 LAN coprocessor when in multicast mode. Receive and transmit buffers of 1.6kb are always used to decrease buffer management and DMA overhead. The number of transmit buffers is fixed at one and is not configurable. If this command is not issued, the Adapter uses the default values shown in parentheses below. The Host should expect the adapter response PCB 31H to confirm execution.

```
db      01      ;command code
db      0C      ;length of data portion of PCB
dw      ?      ;# command queue entries (10)
dw      ?      ;# receive queue entries (20)
dw      ?      ;# multicast addresses (0)
dw      ?      ;# frame descriptors (20)
dw      ?      ;# receive buffers (20)
dw      ?      ;# download programs (10)
```

02H: Configure 82586. Instructs the Adapter to configure the 82586 LAN coprocessor into the given receive mode. If this configure command is not issued, the Adapter will use the default values shown in parentheses below. The Host should expect the adapter response PCB 32H to confirm execution.

```
db      02      ;command code
db      02H     ;length of data portion of PCB
dw      ?      ;receive mode
; bit 2,1,0: receive mode (000)
;      000 = station only
;      001 = plus broadcast
;      010 = multicast
;      100 = promiscuous
; bit 4,3 : loopback mode (00)
;      00 = none (default)
;      01 = 82586 internal loopback
;      10 = 82586 external loopback
```

03H: Ethernet Address. Requests Adapter to return the Ethernet address stored in its address PROM. The Adapter sends the PROM address in PCB 33H.

```
db      03          ;command code
db      00          ;length of data portion of PCB
```

04H: Download Data To 3C505. Requests the Adapter to DMA download data through the data register. The direction bit must be set to the download direction before issuing the command. If the command is accepted, the Adapter sets up the DMA transfer and expects the Host to supply the required number of bytes. There is no Adapter response PCB for this command.

```
db      04          ;command code
db      06          ;length of data portion of PCB
dw      ?          ;data block byte length (must be even)
dw      ?          ;Adapter destination offset
dw      ?          ; " " " segment
```

05H: Upload Data To Host. Requests the Adapter to use its DMA channel to upload data through the data register. The direction bit must be set to the upload direction before issuing this command. If the command is accepted, the Adapter sets up the DMA and expects the Host to read the given number of bytes. There is no adapter response PCB for this command.

```
db      05          ;command code
db      06          ;length of data portion of PCB
dw      ?          ;data block byte length (must be even)
dw      ?          ;Adapter source offset
dw      ?          ; " " " segment
```

06H: Download Data To 3C505. Operates as command code 04H, except that the Adapter uses programmed input/output (PIO) instead of DMA. The direction bit must be set to the download direction before issuing this command. There is no adapter response PCB for this command.

```
db      06          ;command code
db      06          ;length of data portion of PCB
dw      ?          ;data block byte length (must be even)
dw      ?          ;Adapter destination offset
dw      ?          ; " " " segment
```

07H: Upload Data To Host. Operates as command code 05H, except that the Adapter uses PIO instead of DMA. The direction bit must be set to the upload direction before issuing this command. There is no adapter response PCB for this command.

```
db      07          ;command code
db      06          ;length of data portion of PCB
dw      ?          ;data block byte length (must be even)
dw      ?          ;Adapter source offset
dw      ?          ; " " segment
```

08H: Receive Packet. Requests the Adapter to receive an Ethernet packet. The packet type of interest is defined previously by the Configure PCB 02H. When the receive is complete, the Adapter responds with PCB 38H. The Host should set up to input the packet data and then accept the response. The Adapter will DMA upload the packet through the Data Register.

```
db      08          ;command code
db      08          ;length of PCB data portion
dw      ?          ;offset of Host receive buffer
dw      ?          ;segment of Host receive buffer
dw      ?          ;Host receive buffer length in bytes
dw      ?          ;timeout in 10ms increments (zero is
                    ; no timeout; maximum is 32767 ticks)
```

09H: Transmit Packet. Requests the Adapter to transmit a packet. If the PCB is accepted, the Host should DMA download the packet data through the Data Register. When the transmit is complete, the Adapter responds with PCB 39H.

```
db      09H        ;command code
db      06          ;length of PCB data portion
dw      ?          ;offset of Host transmit buffer
dw      ?          ;segment of Host transmit buffer
dw      ?          ;packet length in bytes (must be even)
```

0AH: Network Statistics. This command requests the Adapter to send the cumulative 82586 error statistics and the packet counters kept by the Adapter. The values are returned through the command register in the Adapter response PCB with command code 3AH. The Adapter clears all statistics after sending the response.

```
db      0AH        ;command code
db      00          ;length of PCB data portion
```

OBH: Load Multicast List. The Adapter will add the given list of multicast addresses to the 82586 multicast list. A zero length list will cause the Adapter to clear all multicast addresses and multiple PCB's can be issued to create lists greater than ten entries. The maximum number of addresses in the PCB is ten. The response PCB 3BH will contain command completion status.

```
db      OBH          ;command code
db      6*n          ;length of PCB data portion
db      6 dup(?)     ;Multicast address 1

db      6 dup(?)     ;Multicast address n (10 maximum)
```

OCH: Clear Downloaded Programs. This command releases all adapter memory previously allocated to downloaded programs. The adapter response PCB 3CH will contain the number of paragraphs of program memory available.

```
db      OCH          ;command code
db      0             ;length of PCB data portion
```

ODH: Download Program. Downloaded programs occupy adapter memory not used for packet buffers or system overhead. If this request is accepted, the Adapter will DMA download the program. When done, the Adapter responds with PCB 3DH containing a "program id". The Adapter always provides paragraph alignment to each downloaded program. It is suggested that the Adapter be hard reset before a Download Program sequence.

```
db      ODH          ;command code
db      02           ;length of PCB data portion
dw      ?            ;program length in bytes
```

OEH: Execute Program. The Adapter will pass control to the program defined by the program id. The first executable code is assumed at offset zero relative to the beginning of the downloaded program. The calling sequence to the downloaded program is described in Section 3.3.5, function 2. The Adapter, when done, responds with PCB 3EH.

```
db      OEH          ;command code
db      02+n         ;length of PCB data portion
dw      ?            ;program id
db      n dup(?)     ;variable length parameter list
```

0FH: **Self-Test.** The Adapter will execute its self-test. The Adapter, when done, responds with PCB 3FH.

```
db      0FH          ;command code
db      0            ;length of PCB data portion
```

10H: **Set Ethernet Address.** The Adapter will issue an IA-setup command to the 82586 coprocessor specifying an Ethernet station address. If this command is not used, the address from the Adapter's Ethernet address PROM is used to configure the 82586.

```
db      10H          ;command code
db      6            ;length of PCB data portion
db      6 dup(?)     ;Ethernet address
```

11H: **Adapter Info.** Requests the 3C505 Adapter to send general information that describes the adapter configuration. The Adapter, when done, responds with PCB 41H.

```
db      11H          ;command code
db      0            ;length of PCB data portion
```

3.2.2 3C505 TO HOST PCB FORMATS

31H: **Configure Adapter Response.** After the Adapter has initialized the PCB command queue and the multicast address storage area, it responds with status in this PCB.

```
db      31H      ;command
db      02      ;length of PCB data portion
dw      ?       ;status 0 = successful
```

32H: **Configure 82586 Response.** After the Adapter has initialized the 82586 coprocessor using parameters in the PCB 02H command, it responds with status in this PCB.

```
db      32H      ;command code
db      02      ;length of PCB data portion
dw      ?       ;status 0 = successful
```

33H: **Ethernet Address Response.** The Adapter returns the 6-byte Ethernet address in this response PCB. The address has previously been read from the Ethernet address PROM and stored into memory.

```
db      33H      ;command code
db      06      ;length of PCB data portion
db      6 dup(?) ;Ethernet address
```

34H: **Download Data To 3C505.** In this PCB, the Adapter requests that the Host download a block of host memory to the 3C505. If the command is accepted by the Host, the Adapter will use DMA to transfer the data through the data register.

```
db      34H      ;command code
db      06      ;length of data portion of PCB
dw      ?       ;data block byte length (must be even)
dw      ?       ;Host data block source offset
dw      ?       ;Host " segment
```

35H: **Upload Data To Host.** In this PCB, the Adapter requests that the Host upload a block of data into Host memory. If the command is accepted, the Adapter will set up a DMA to transfer the appropriate data through the data register.

```
db      35H      ;command code
db      06      ;length of data portion of PCB
dw      ?       ;data block byte length (must be even)
dw      ?       ;Host data block destination offset
dw      ?       ;Host " segment
```

38H: Packet Received Response. When the Adapter receives a packet and there is an outstanding Host request to receive a packet, the Adapter sends this response PCB and follows it with a DMA upload. The number of bytes DMA'ed will not exceed the buffer length specified in the receive packet command PCB 8H; extra packet data is discarded.

```

db      38H      ;command code
db      10H      ;length of PCB data portion
dw      ?       ;offset of Host receive buffer
dw      ?       ;segment of Host receive buffer
dw      ?       ;number of bytes to be DMA'ed
dw      ?       ;actual packet length
dw      ?       ;completion status  0 = successful
                        ;                               -1 = timeout
dw      ?       ;82586 receive status
dd      ?       ;double word time tag in 15us ticks

```

39H: Transmit Packet Complete. The status of a packet transmission is returned to the Host in this response PCB.

```

db      39H      ;command code
db      08H      ;length of PCB data portion
dw      ?       ;offset of Host transmit buffer
dw      ?       ;segment of Host transmit buffer
dw      ?       ;completion status  0 = successful
dw      ?       ;82586 transmit status

```

3AH: Network Statistics Response. The Adapter returns the total packet counters and the 82586 error counters in this response PCB.

```

db      3AH      ;command code
db      0CH      ;length of PCB data portion
dd      ?       ;total receive packets
dd      ?       ;total transmit packets
dw      ?       ;CRC error counter
dw      ?       ;alignment error counter
dw      ?       ;no resources error counter
dw      ?       ;overrun error counter

```

3BH: Load Multicast Complete. After the multicast list is loaded into the 82586, the Adapter responds with this PCB.

```

db      3BH      ;command code
db      02       ;length of PCB data portion
dw      ?       ;status  0 = successful

```

3CH: Clear Downloaded Program Response. To clear the downloadable program memory, the Adapter reinitializes the structures and variables describing each downloaded program. The Adapter sends the amount of program memory available in paragraphs in this response PCB.

```
db      3CH          ;command code
db      02          ;length of PCB data portion
dw      ?           ;amount of downloadable program
                        ;memory in paragraphs
```

3DH: Download Program Response. A downloaded program is assigned a "program id" by the Adapter. The id is used by the Host and Adapter when specifying which downloaded program to execute or has executed.

```
db      3DH          ;command code
db      08          ;length of PCB data portion
dw      ?           ;program id: > 0 , if allocated
dw      ?           ;program offset in adapter memory
dw      ?           ;program segment in adapter memory
dw      ?           ;remaining memory in paragraphs
```

3EH: Execute Program Response. After a downloaded program has executed, it sends this response PCB to the Host. The return status and parameters are program dependent.

```
db      3EH          ;command code
db      02+n        ;length of PCB data portion
dw      ?           ;program id: -1 if bad id in request
db      n dup(?)    ;return status and parameters
```

3FH: Self-Test Response. The adapter self-test consists of a ROM checksum, non-destructive RAM test, and internal loopback test on the 82586. Status of the test is returned in this PCB.

```
db      3FH          ;command code
db      2+2n        ;length of PCB data portion
dw      ?           ;self-test status
dw      n dup(?)    ; and optional failure data
```

The self-test status codes and failure data for each are:

| # | STATUS | FAILURE DATA |
|---|--------------|------------------------------------|
| 0 | no errors | none |
| 1 | ROM checksum | computed checksum value |
| 2 | RAM test | failed memory offset:segment |
| 3 | 82586 test | status word: |
| | | bit 14 = internal loopback failure |
| | | 13 = external loopback failure |
| | | 12 = configure error |

40H: Set Address Response. After the Adapter sets the Ethernet address in the 82586, this response is sent to the Host.

```
db      40H      ;command code
db      2        ;length of PCB data portion
dw      ?        ;status 0=successful
```

41H: Adapter Info Response. The Adapter formats a response containing the ROM revision code, ROM checksum value, total amount of memory in kilobytes, and the segment/offset pointer to free memory.

```
db      41H      ;command code
db      10       ;length of PCB data portion
dw      ?        ;ROM revision level (0x0300 = rev 3.0)
dw      ?        ;checksum value in rom
dw      ?        ;amount of memory in kbyte
dw      ?        ;free memory offset
dw      ?        ; and segment
```

3.3 SYSTEM ROM UTILITIES

Programs downloaded into the 3C505 Adapter can access the adapter resources directly or through a set of utilities available in ROM code. To simplify and standardize usage, a set of soft interrupt vectors have been allocated to support the following:

- * Host I/O
- * Network I/O
- * Configuration/Status
- * System Timer
- * Download Program Support
- * PCB Command Processing
- * Receive Packet Processing
- * Timed execution
- * PCB Enqueueing

These vectors may be replaced or chained to by user downloaded programs. To chain to a vector, the downloaded program should first save the current vector before replacing it with a pointer to itself. A program then gains control when the vector is invoked. As appropriate, the program should pass control to the next program in the chain.

The 3C505 ROM utilities always save and restore the calling program's SS, DS, ES and SP. It is suggested that downloaded programs also maintain these registers.

3.3.1 HOST I/O SUPPORT: INT 80H

This group of function allows upload and download of data and command blocks between the Adapter and PC Host. Both DMA and PIO methods of IO are supported.

01H: Download Request. The Adapter formats a download request PCB, sends it to the Host, and waits for Host acknowledgement (HSF2 and HSF1 state 01 or 10). If the Host accepts the request (state 01), the caller should proceed with the data transfer.

ax = 1 function code
es:bx Host source buffer address
cx buffer length bytes (must be even; maximum 64kb)
dx initial timeout in 10ms increments
 (maximum is 32767 ticks)

Return: carry clear if successful
 carry set if error, ax = error code

02H: DMA Upload Request. The Adapter formats a upload request PCB and sends it to the Host. If the request is acknowledged by the Host (HSF2 and HSF1 state 01), the Adapter should perform the data transfer.

ax = 2 function code
es:bx Host destination buffer address
cx buffer length bytes (must be even; maximum 64kb)
dx initial timeout in 10ms increments
(maximum is 32767 ticks)

Return: carry clear if successful
carry set if error, ax = error code

03H: Get Primary Command Block From Host. This function reads a PCB through the Command Register and stores it into the destination buffer.

ax = 3 function code
es:bx buffer address
cx buffer length bytes
dx initial timeout in 10ms increments
(maximum is 32767 ticks)

Return: carry clear if successful
carry set if error, ax = error code

04H: Send Primary Command Block To Host. This function sends the given PCB buffer to the Host and waits for host acknowledgement either accept or reject.

ax = 4 function code
es:bx buffer address
cx total buffer length bytes
dx initial timeout in 10ms increments
(maximum is 32767 ticks)

Return: carry clear if successful and accepted
carry set if error, ax = error code

05H: 06H: Host Data Input. Assuming that the Host is already configured to perform a download, this function transfers host data into the passed buffer using DMA or PIO. When DMA is used, the timeout value passed in register DX is the amount of time the Adapter waits for the DMA semaphore to signal "available". If a timeout occurs, the Adapter assumes some sort of error has occurred and takes over use of the DMA channel. Also, the function initiates DMA transfer but does not wait for completion. Use int 80H function 0BH to poll for DMA completion.

ax = 5 DMA download or
ax = 6 PIO download
es:bx buffer address
cx buffer length in bytes to transfer (must be even)
dx timeout in 10ms increments
(maximum is 32767 ticks)

Return: carry clear if successful
carry set if error, ax = error code

07H: 08H: Host Data Output. Assuming that the Host is already configured to perform an upload, this function transfers adapter data (using DMA or PIO) from the adapter buffer to the Host. The comments describing DMA in int 80H function 05H also apply here.

ax = 7 DMA upload or
ax = 8 PIO upload
es:bx buffer address
cx buffer length in bytes to transfer (must be even)
dx initial timeout in 10ms increments
(maximum is 32767 ticks)

Return: carry clear if successful
carry set if error, ax = error code

09H: 0AH: Accept or Reject PCB. When a PCB is read using int 80H function 3, the protocol described in section 3.1.1 requires that the PCB be accepted or rejected. The following function codes provide this ability to downloaded programs:

ax = 09H Accept Host PCB
ax = 0AH Reject Host PCB

Function 9 to accept a PCB also includes a Data Register flush operation to prepare for a DMA or PIO data transfer.

0BH: Check DMA Complete. When a DMA transfer has been initiated, this function can be called to check if the DMA has completed.

ax = 0BH Check DMA Complete
dx timeout in 10ms ticks (maximum is 32767 ticks)

Return: carry clear if DMA done
carry set if timeout

3.3.2 NETWORK I/O SUPPORT: INT 81H

1H: Transmit Packet. To transmit a packet, this function links the given packet buffer to the first 82586 transmit buffer descriptor, links the descriptor to the one and only transmit command block, links the command block to the system control block, and then signals channel attention to the 82856. The transmit packet function will poll for transmit complete before returning to the calling routine.

ax = 1 function code
es:bx buffer address
cx buffer length bytes
dx timeout in 10ms increments

Return: carry clear if successful
 carry set if error, ax = 82586 transmit status

2H: Receive Packet. During execution of the Adapter and 82586 configure commands (section 3.2), the 82586 Receive Frame Area is initialized and the Receive Unit is commanded to start frame reception. A receive packet is first detected by the 82586 interrupt service routine, which time tags it and then updates global pointers to an "received packet" list. This function checks that list for an entry and gives it to the caller.

ax = 2 function code
dx timeout in 10ms increments
 (maximum is 32767 ticks)

Return: carry clear if successful
 es:bx = packet buffer address
 cx:dx = double word time tag (high:low order)
 di = packet length in bytes
 si = 82586 receive status
 carry set if error, ax = error code

03H: Return Buffer. After a packet buffer is processed, it must be returned to the system so that the buffer can be relinked to a Receive Buffer Descriptor which is in turn relinked to the free RBD list.

ax = 3 function code
es:bx Buffer address

Return: carry clear if successful
 carry set if error, ax = error code

3.3.3 CONFIGURATION/STATUS: INT 82H

01H: Configure Adapter Memory. The Adapter allocates memory for the PCB command queue, receive command queue, multicast address list, frame descriptors, receive buffers, and download program structures. Each PCB or receive command queue entry is large enough to buffer a maximum size PCB of 64 bytes. Each multicast address occupies 6 bytes. Receive and transmit buffers are fixed at 1.6kb in order to decrease management overhead. The number of transmit buffers is fixed at one.

ax = 1 function code
es:bx pointer to configuration control block

num_PCB_entries DW ? (10)
num_receive_Q_entries DW ? (20)
num_multicast_entries DW ? (0)
num_frame_descriptors DW ? (20)
num_receive_buffers DW ? (20)
num_download_programs DW ? (10)

Return: carry clear if successful
 carry set if error, ax = error code

02H: Configure 82586 Receive Mode. Instructs the Adapter to set the 82586 coprocessor into the given receive mode.

ax = 2 function code
bx receive mode
 bit 2,1,0: receive mode (000)
 000 = station only
 001 = plus broadcast
 010 = multicast
 100 = promiscuous
 bit 4,3 : loopback mode (00)
 00 = none (default)
 01 = internal loopback
 10 = external loopback

Return: carry clear if successful
 carry set if error, ax = error code

03H: Return Ethernet Address. Read the Ethernet address PROM and store the 6-byte address into the caller's buffer.

ax = 3 function code
es:bx buffer address

Return: carry clear if successful, buffer es:bx updated
 carry set if error, ax = error code

04H: Set Ethernet Address. Use the supplied Ethernet address and issue an IA-setup command to the 82586 coprocessor.

ax = 4 function code
es:bx buffer address

Return: carry clear if successful
 carry set if error, ax = error

05H: Set LEDs. This function allows downloaded programs to control the state of the Adapter's two LEDs. LED #2 is periodically flashed by the Adapter to indicate normal operation; this is called the "heartbeat".

ax = 5 function code
bx control word
 bit 1,0 = LED2,LED1 value; 1=ON
 bit 2 = enable/disable heartbeat; 1=enable

Return: ax = current control register value in high byte
 carry clear if successful
 carry set if error

06H: Get Adapter Info. Retrieve general adapter information.

ax = 6 function code
es:bx buffer address

Return: es:bx = revision id
 rom checksum
 memory size in kbytes
 free memory offset
 free memory segment
 cx = data length in bytes
 carry clear if successful
 carry set if error

3.3.4 TIMER SUPPORT: INT 83H

The 3C505 Adapter maintains both a 10ms and 15us double word time tick counter using two 16-bit timers in the 80186 microprocessor. The 15us timer is meant for high resolution timeout or time tag applications and generates a timer interrupt every 1.6 seconds. The 10ms timer generates an interrupt every 10ms and can be used for timeout calculations. Every five 10ms ticks the Adapter also calls the Idle vector (section 3.3.8). This allows downloaded programs that are "chained" through the Idle vector a chance to execute periodically.

01H: Set 10ms Double Word Time. Set the global double word 10ms tick counter to given value.

ax = 1 function code

cx high portion of 10ms count
dx low portion of count

02H: Read 10ms Double Word Time. Retrieve the current double word 10ms tick counter.

ax = 2 function code

Return: cx high portion of 10ms count
dx low portion of count

03H: Set 15us Double Word Time. Set the global double word 15us tick counter to given value.

ax = 3 function code
cx high portion of 15us count
dx low portion of count

04H: Read 15us Double Word Time. Retrieve the current double word 15us tick counter.

ax = 4 function code

Return: cx high portion of 15us count
dx low portion of count

3.3.5 DOWNLOAD PROGRAM SUPPORT: INT 84H

The Adapter uses low memory for data, stack, packet buffers, and PCB command queue. Remaining memory is available for downloaded programs. Programs must have statically allocated memory for global data. Programs can use the one kiloword stack segment setup by the 3C505 ROM. Downloaded programs should not reconfigure adapter memory.

01H: Clear Downloaded Programs. This command releases all adapter memory previously allocated to downloaded programs. Soft interrupt vectors are restored to the reset state.

ax = 1 function code
Return: carry clear if successful, ax = # free paragraphs
 carry set if error, ax = error code

02H: Execute Program. Control is passed to the program defined by the program id. Executing a program is a far subroutine call to the program with the following registers setup:

ax = 2 function code
es:bx address of parameter list
cx length of parameter list in bytes
dx program id

Return: carry clear if successful
 es:bx = address of return buffer
 cx = length of return buffer
 carry set if error, ax = error code

3.3.6 PCB COMMAND PROCESSOR: INT 85H

The adapter Command Register ISR (interrupt service routine) reads host PCB's and places all PCB's except receive commands (see section 3.3.7) into a command queue. The 3C505 ROM idle loop monitors and dequeues each PCB for execution. The PCB is passed to a PCB Command Processor whose address is stored in this interrupt vector. The Command Processor is given an ES:BX pointer to the PCB entry.

It uses the PCB command number to key into a jump table of command processing subroutine addresses. The selected command processing subroutine is also passed the PCB (often containing parameters for the subroutine). The Command Processor ignores PCB's with command numbers not defined in this specification, and immediately returns to the foreground idle.

A downloaded program can chain to this vector; that is, the program saves the current interrupt vector contents and replaces it with a pointer to itself. Then the downloaded program has an opportunity (not necessarily the first) to examine the PCB. If the program does not want to execute the PCB, the program must pass it to the command processor that it replaced. In this case the program must be careful that register values are not modified. This becomes a mechanism for creating new commands or replacing existing ones.

3.3. PACKET PROCESSOR VECTOR: INT 86H

This software interrupt vector defines the address of the Packet Processor. The Packet Processor centralizes handling and queuing of host receive command PCB's and received packets. The management of the 82586 LAN coprocessor is performed outside the Packet Processor. A downloaded program can replace this vector in order to implement a more specialized scheme than the ROM-based functions described below.

As mentioned in section 3.3.6, all PCB's EXCEPT the receive packet command are placed into a PCB queue. The Command Register ISR gives the receive command to the Packet Processor with:

01H: Enqueue Receive Command PCB

ax = 1 command code
es:bx pointer to PCB

A separately managed Receive Command Queue is built.

When packets are received, the 82586 ISR time tags the frame with a double word 15us time and updates global pointers to the frame and exits. The foreground idle loop obtains a packet from the the Network I/O vector INT 81H function 2. The idle loop gives the packet to the Packet Processor with:

02H: Enqueue Receive Packet

ax = 2 command code
es:bx pointer to receive buffer
cx:dx double word time tag (high:low)
di packet length

The firmware builds a queue of receive packets. If there is a pending receive command, a receive response PCB is sent to the Host, packet data is DMA uploaded, and the packet buffer is returned to the system. Otherwise, the packet is enqueued and processing is done. This vector/function can be replaced by downloaded programs enabling them to receive and process all incoming packet.

Since receive commands have timeout values, the foreground idle loop needs to periodically call the Packet Processor with:

03H: No Operation

ax = 3 NOP

so that it has an opportunity to scan the receive command queue to check if any request has timed out. If a request has timed out, a receive response PCB with failure status is formatted and sent to the Host.

3.3.8 IDLE VECTOR: INT 87H

The Idle vector is called every 50ms from the main 3C505 ROM idle loop. Programs chained through this vector will have a chance to execute.

To chain to the vector, a downloaded program should first save the current vector before replacing it with a pointer to itself. It is also important that the program pass control (using a far jump) to the next program in the chain. Remember that SS, SP, DS and ES should always be saved on entry and restored on exit. In this way, each downloaded program in the chain has an opportunity to execute.

The default ROM Idle vector routine clears a global flag that determines whether the Idle vector should be called. In a situation where a download program has chained to the Idle vector but never passes control to the default Idle vector routine, this flag is never cleared. The effect is that the download program will be called on every pass through the main 3C505 ROM idle loop.

3.3.9 PCB ENQUEUE VECTOR: INT 88H

The PCB Enqueue Vector is called to add a PCB entry into the system PCB queue. PCB's are dequeued by the foreground idle loop and given to the PCB Command Processor for execution. The calling sequence for this function is:

es:bx pointer to PCB

Return: carry clear if successful
carry set if error, ax = error code

The Command Register interrupt service routine (ISR) uses this vector after it receives a PCB. If a downloaded program has read a PCB from the Host and does not recognize it, it should use this vector to enqueue it for later execution. Alternatively, a downloaded program can chain to this vector to receive PCB's from the Command Register ISR.

APPENDIX A

80186 PERIPHERAL CONTROL BLOCK PROGRAMMING

```

;
reloc_reg_cont      equ      00ffh      ;initialization constant of
umcs_cont           equ      0fc3ch     ;chip selects
lmcs_cont           equ      3ffch
mmcs_cont           equ      81fch
mpcs_cont           equ      0a0bch
pacs_cont           equ      003ch
;
;
; pic priority assignment:
;
level_int0          equ      4          ;int0 priority, Command Register Full
level_int1          equ      2          ;int1 priority, interrupt form 82586
level_int2          equ      5          ;not used
level_int3          equ      6          ;not used
level_dma0          equ      0          ;dma0 priority, DRAM refresh
level_dmal          equ      3          ;dmal priority, Data Register
level_timer         equ      1          ;timer priority
;
; initialization of pic control registers
;
pic_timer_cont      equ      level_timer
pic_dma0_cont       equ      level_dma0
pic_dmal_cont       equ      level_dmal
pic_int0_cont       equ      level_int0
pic_int1_cont       equ      level_int1
pic_int2_cont       equ      000dh
pic_int3_cont       equ      000eh
;
; initialization of timer 2 registers; 15 microsecond for DRAM refresh
; and Hi RES Timer
;
t2cnt_cont          equ      0
t2maxra_cont        equ      30
t2cntrl_cont        equ      0c001h
;
; initialization of timer 0 registers; 10 millisecond interrupt
;
t0cnt_cont          equ      0
t0maxra_cont        equ      20000/t2maxra_cont
t0maxrb_cont        equ      0
t0cntrl_cont        equ      0e029h
;
;

```

```

;
; initialization of timer 1 registers; Hi res system Timer
;
tlcnt_cont      equ      0
tlmaxra_cont    equ      0ffffh
t0maxrb_cont    equ      0
t0cntrl_cont    equ      0e029h
;
; initialization of dma0 registers; DRAM refresh
;
dma0srclo_cont  equ      0000
dma0srchi_cont  equ      0000h
dma0dstlo_cont  equ      0800h
dma0dsthi_cont  equ      0008h
dma0cnt_cont    equ      0ffffh
dma0cntrl_cont  equ      0157fh
;
; DMA 1 control register values
;
dma1_from_dr_cntrl equ    0a347h ;DMA input from host
dma1_to_dr_cntrl   equ    1787h  ;DMA output to host

```

APPENDIX B

This is an example of the parameters used to configure the Intel 82586 LAN Coprocessor. Please refer to the "Intel LAN Components User's Manual" for the description of the abbreviations used.

| | |
|--------------------|--------------------------------------|
| FIFO LIM | = 6 |
| BYTE CNT | = 0CH |
| EXT LP BCK | = 0 |
| INT LP BCK | = 0 |
| PREAM LEN | = 2 |
| AT LOC | = 1 |
| ADDR LEN | = 6 |
| SAV BF | = 0 |
| SRDY | = 1 |
| INTERFRAME SPACING | = 60H |
| BOF MET | = 0 |
| ACR | = 0 |
| LIN PRIO | = 0 |
| RETRY NUM | = 0FH |
| SLOT TIME | = 200H |
| CDT SRC | = 0 |
| CDTF | = 0 |
| CRS SRC | = 0 |
| CRSF | = 0 |
| PAD | = 0 |
| BT STF | = 0 |
| CRC16 | = 0 |
| NCRC INS | = 0 |
| TONO CRS | = 0 |
| MANCH/NRZ | = 0 |
| BC DIS | = 1 (reconfigurable by host command) |
| PRM | = 0 |
| MIN FRM LEN | = 40H |

APPENDIX C

3C505 DIAGNOSTIC

The 3C505 EtherLink Diagnostic Diskette supplied with your EtherLink Plus card includes a diagnostic program called 3C505.EXE, which can be used to help you identify hardware problems on the 3C505.

This appendix describes the 3C505.EXE program, the equipment and tools required to run the program, and a step-by-step procedure on how to use the diagnostic.

3C505.EXE takes about twelve minutes or less (five minutes in an AT) from start to finish. As it runs, it reports its progress by displaying the name of the group of tests being performed and a pass count. The test stops and displays an error message if a hardware error is detected.

- | | |
|-----------------------|---------------------------|
| 0 - Adapter self test | 4 - Recognizer test |
| 1 - Preliminary test | 5 - Message exchange test |
| 2 - DMA test | 6 - Passive receive test |
| 3 - Packet test | 7 - NS echo server |

When the standard 3C505.EXE program is run, Test 0 through Test 4 are performed in sequence. Test 5 through Test 7 are run individually and must be specified separately. Tests 3 and 4 require the use of a loopback plug to prevent network activity from producing erroneous test results, and to keep test packets from polluting the network. Tests 5 through 7 must be run while the Adapter is connected to the network. A brief description of each test follows:

TEST 0

Test 0 resets the Adapter causing the self-test routines to be executed. These include 80186 and 82586 initialization, memory and internal and external loopback tests. The results are passed back to the Host and displayed. If communication between the Host and Adapter can not be established, an error message is displayed.

TEST 1

Test 1 tests the interface between the Host and the Adapter using Programmed I/O data transfers.

TEST 2

Test 2 tests the interface between the Host and the Adapter using DMA data transfers.

TEST 3

Test 3 performs a transmit test and, if successful, a loopback test. The transmit test checks for the correct status from the 82586 LAN controller following transmits. Loopback test further compares the received packets with those transmitted.

TEST 4

Test 4 tests the 82586 LAN controller address matching functions. The receiver is configured to various modes; station only, multicast, broadcast, and promiscuous. In each mode, packets of differing destination address and size are transmitted and the ability of the Adapter to reject or accept packets properly is tested.

TEST 5

Test 5 performs packet exchange with another PC or server on the network. The PC transmits an "echo request packet" into the network. A responding server or PC will transmit the packet (echo) back to the PC under the test.

TEST 6

Test 6 detects legal packets on the network and counts them. This tests the adapter's receive function and provides a diagnostic tool for locating problems elsewhere on the network. This test is "passive" to the network and can be used to check the transmit capability of another PC on the network.

TEST 7

Designates this PC as an "echo server", which is used to exchange packets with PCs running Test 5. The PC remains in this mode until a key is depressed.

FORMAT

3C505 [-Ix][-Dx][-Bxxx][-#][-E][-T]

PROGRAM PARAMETERS

Ix Test uses interrupt x, default (factory setting) is Interrupt 3.

Dx Test uses DMA channel x, default is DMA 1.

Bxxx Sets the base address of the EtherLink card to xxx (three hexadecimal digits). This option should be used if the I/O address jumpers on the EtherLink card have been changed. The default value is 300 hex.

Enter 5, 6 or 7 to select one of the following test.

- 5 - Message exchange test
- 6 - Passive receive test
- 7 - NS echo server

E Used with test 5 only. Use NS echo protocol to access remote nodes during the message exchange test. This option should be used on a Xerox NS 8000 network that have echo servers or if there is another PC with a 3Com EtherLink 3C505 running 3C505 -7 to reply to the echo request. If this option is not specified, the diagnostic will generate EtherSeries echo requests, and any EtherSeries server on the network will reply using EtherSeries protocol.

T Specifies that the Host computer is a TI Professional which requires special treatment.

Requirements for Testing

For testing, you need:

1. A loopback plug;
2. Another IBM PC on the network OR an EtherSeries network server that is connected to the network.

The second PC will be used as an echo server, which will exchange packets over the network with the computer under test.

RUNNING THE 3C505.EXE PROGRAM

To start the 3C505.EXE program, disconnect your PC from the network, attach a loopback plug to the BNC connector, insert the diskette in drive A: and type:

```
A> 3C505
```

Remember to give the option -I, -D, -B if the factory settings were changed. As the program runs, it prints a message indicating which test is being performed and the progress. Once it detects an error, the test stops and displays a message. Test 0 thru 4 mentioned above will start one after another.

To run Test 5, connect the PC to the network with either:

1. An EtherSeries network server

OR

2. Another PC, with an EtherLink 3C505 as a echo server. Start the server running 3C505.EXE on this diskette by typing:

```
A> 3C505 -7
```

OR

3. Another PC with another type of 3Com EtherLink acting as an echo server. Use the diagnostic program supplied with that specific EtherLink such as the 3C501 and type:

```
A> 3C501 7
```

If the echo server is case 2 mentioned above, then type:

```
A> 3C505 -5 -E
```

If the echo server is case 3 mentioned above, then type:

```
A> 3C505 -5
```

3C501 / 3C505 DIAGNOSE PROGRAM DIFFERENCES

The 3C505.EXE diagnostic is modeled after the 3C501 version, 3C501.EXE (or DIAGNOSE.COM in earlier versions). For those of you familiar with the 3C501 diagnostic, all test types (Preliminary, DMA, Packet, Recognizer, Message exchange, Passive receive, NS Echo server) have been carried forward into the 3C505 diagnostic. The actual interpretation and implementation of these tests for the 3C505 are, however, very different. It is worth noting that in the 3C505 diagnostic,

- o there is an additional test 0 (Adapter self test);
- o command line switches (to set adapter base address, DMA channel, etc. for the test) are always preceded by a dash "-";
- o tests 0 through 4 (Adapter Self test, Preliminary, DMA, Packet, Recognizer) are always executed one after another (i.e., you cannot use "3C505 -3" to run Packet test only);
- o tests 0 through 4 are skipped when test 5, 6, or 7 (Message exchange, Passive receive, NS Echo server) is selected.

After running 3C505.EXE, you might find it necessary reboot your system or reinitialize the 3C505. This is especially true if you use download software or if a particular 3C505 configuration is expected.

APPENDIX D

3D DEBUGGER

3D is a program for loading and debugging programs that run on the 3C505 adapter. 3D runs on an IBM PC (or compatible), under PC-DOS; the 3D Host program communicates through the Command Register with the "slave", a small program in the 3C505 ROM. 3D can start, stop, and single-step the 3C505's 80186 processor; set, and clear breakpoints; download, modify, and examine memory.

3D requires an IBM PC with 256K of memory, one serial port, and a Mouse Systems PC mouse. 3D also runs on IBM compatibles such as the Compaq portable and Zenith Z-150. Attach the mouse to the COM2 port. Recall that COM2 always uses interrupt level 3 so beware of conflicts. The base I/O address of the 3C505 in this version of 3D must be set to 310 hex. The TEST jumper on the 3C505 can also be set to the "1" position; this causes the 3D Slave to install itself at boot time. All "exception" and unused interrupt vectors will point to the 3D Slave. The ability to single step and to use breakpoints will be enabled.

3D divides the display into four regions: from top to bottom these are tile area, menu bar, typein, and message area. The tile area is for display and alteration of 3C505 internal registers, I/O registers, and memory. The menu bar is for control of the 3C505. The typein area displays the accumulated typein from the user. The message area is for the display of status and error messages from 3D.

```

*AX          0000 BP          300C CS          03F0 *IP          6A76
*BX          6892 *SP        3000 SS          3C6D *FLAGS        F046
CX           0000 SI          04C4 DS          03F0
DX           0000 *DI        6806 *ES          03F0

*40:8        0040          *3C6D:3000      6A76
40:0A        0000          *3C6D:3002      03F0
40:0C        0300          *3C6D:3004      F046
40:0E        2E2E          *3C6D:3006      F206
40:10        0000          *3C6D:3008      6A85
40:12        0000          *3C6D:300A      6FD7
40:14        0000          *3C6D:300C      3C6D
40:16        0000          *3C6D:300E      3C6D
              *3C6D:3010      3C6D
40:18        0014          *3C6D:3012      0000
40:1A        0000          *3C6D:3014      04C4
40:1C        0000          *3C6D:3016      303A
40:1E        2E2E          *3C6D:3018      3022
40:20        2EDE          *3C6D:301A      064D
40:22        2F0A          *3C6D:301C      0000
40:24        2F5A          *3C6D:301E      0000

Boot! Load! Go! Continue! Stop! Step! Break! Unbreak! UnbreakAll! Probe!
ss:sp
popf >> Non maskable interrupt at 3F0:6A76

```

Figure 1. The display just after a stop.

TILE AREA

The tile area is divided into a 21 by 4 array of tiles. Each tile displays the name of a register or memory location and its contents. A tile consists of three fields; in reading order these are flag, left, and right. The left field holds the register name and the right field displays its contents. A tilde, "~", in the flag means the data in the right field is invalid for some reason. An asterisk, "*", means the right field changed since 3D last read the register. When the cursor enters the tile area, 3D inverts the tile occupied by the cursor.

MENU BAR

The menu bar provides access to functions for controlling the 3C505. To select a function, move the cursor over the name of the function and click the left mouse button. Functions on the menu bar are Boot, Load, Go, Continue, Stop, Step, Break, Unbreak, UnbreakAll, and Probe.

Boot issues a hard reset to the 3C505 Adapter.

Load downloads a file from the IBM PC to the 3C505 through the Command Register. 3D always uses the last file specified unless the typein ends with a slash, "/". In this case, 3D uses the typein, minus the slash, as the file name; thus, if the same file is loaded many times, the file name need not be supplied every time. The slash is not necessary for the first file loaded. The file is downloaded into Adapter memory at the address that is specified in the typein area when function key F6 is pressed. Please note that this Load operation is NOT THE SAME as the download program PCB described in the Chapter 3.

Go evaluates the typein and starts the 3C505 running at the specified address.

Continue starts the 3C505 running at the current CS:IP. Once Go or Continue starts the 3C505 running, 3D becomes a "dumb" terminal for the 3C505; 3D monitors both the IBM PC keyboard and the Command Register. 3D transmits any character typed on the keyboard through the Command Register. The bottom sixteen lines of the tile area become the display area for any characters transmitted by the 3C505 through the Command Register.

Stop establishes communication between 3D and the Slave through the Command Register. If the Slave does not respond, 3D displays an error in the message area. No other commands will work until 3D and the Slave are communicating. Stop is always appropriate no matter what state the 3C505 is in; however, booting will destroy the state of the 3C505.

Step forces the 3C505 to execute the next instruction; values in the tile area are refreshed. The TEST jumper must be in the "1" position in order to single step.

Break sets a breakpoint at the location indicated by the typein. 3D allows eight breakpoints. All of these breakpoints are sticky, unlike debuggers which forget all breakpoints after hitting a breakpoint. After hitting a breakpoint, 3D removes all breakpoints instructions from memory. 3D restores all breakpoints when it Continues the 3C505. The TEST jumper must be in the "1" position to use breakpoints.

Unbreak removes a breakpoint. Breakpoints are numbered from zero to seven; 3D evaluates the typein and removes the breakpoint with the specified number. Blank typein clears breakpoint zero.

UnbreakAll removes all breakpoints.

Probe refreshes the values in the tile area. Probe is useful when the 3C505 has entered the Slave as a result of a software trap condition and no 3D was present to monitor the Command Register. At a later time, 3D can be executed on the PC with the 3C505; a Probe then can retrieve the state of the 3C505 when it entered the Slave.

TYPEIN AREA

Most tile operations take the typein as an argument. 3D evaluates arithmetic expressions in the typein using customary operator precedence; parentheses and square brackets can be used freely to group subexpressions. All operations use long (32 bit) arithmetic. Legal operators are addition (+), subtraction (-), multiplication (*), division (/), remainder (%), bitwise and (&), bitwise exclusive or (^), bit wise inclusive or (|), bit wise and with complement (\). The colon (:) operator specifies a memory address. The expression to left of the colon is the segment; the value to the right is the offset. When 3D encounters a register name in a colon expression, the contents of the referenced register are used for evaluating the expression; for example SS:SP refers to the top of the stack, SS:SP+2 refers to the word just below the top of the stack.

All constants are hexadecimal; constants beginning with an alphabetic digit must have a leading zero. 3D predefines all of the 80186 processor registers: AX, BX, CX, DX, SP, BP, ES, DS, CS, IP, and FLAGS. The symbol .IO references registers on the I/O bus; for example, .IO+6 references I/O bus address six. Any expression which evaluates to a constant represents an unsegmented absolute memory location; hence 400 references the same locations as 40:0 and 30:100.

USING THE MOUSE TO CONTROL DISPLAY

To display a register, move the cursor to the left field of a tile and click the left mouse button. 3D evaluates the expression in the typein as the name of a register or memory location and displays the name contents in the selected tile. To alter the contents of a register move the cursor over the right field of the tile and click the left mouse button. 3D takes the typein, evaluates it, and puts the result in the register.

The other two mouse buttons are also useful. Clicking the right button picks up the text underneath cursor and appends it to the typein; clicking over an empty tile clears the typein. Clicking the middle button appends a colon to the typein.

Columns of tiles can be cleared or filled by holding down the shift key and clicking the left mouse button. If the cursor is over an empty tile, the tiles below are cleared. If the cursor is over the left field of a filled tile, 3D fills the column below with successive words. If the cursor is over the right field of a filled tile, 3D copies the value under the cursor down the column into the right fields. In all cases, 3D evaluates the typein for the number of tiles to touch; zero or illegal values touch all tiles to the bottom of the column.

FUNCTION KEYS

At times, 3D overlays the bottom sixteen lines of the tile area with other information. An inverted square indicates the cursor location; clicking the right mouse button picks up the text under the cursor and places it in the typein. The function keys control what information and the format of data displayed in this region of the tile area.

F1 evaluates the typein and disassembles at the specified memory location. If the typein is blank or can't be evaluated, 3D displays the last instructions that were disassembled. 3D caches the last eight screenfuls of disassembled instructions; PgUp can therefore step through previously displayed disassembly. PgDn disassembles the next screenful.

F2 through F4 work in a manner similar to F1 except that they display memory as longs (32 bits), shorts (16 bits), or bytes respectively. An additional column to the right displays storage as ASCII text; unprintable characters are printed as a period, ".".

F5 makes 3D monitor the Command Register and keyboard.

| | | | | |
|-----|----------|----------|-------------|------|
| *AX | 0000 BP | 300C CS | 03F0 *IP | 6A76 |
| *BX | 6892 *SP | 3000 SS | 3C6D *FLAGS | F046 |
| CX | 0000 SI | 04C4 DS | 03F0 | |
| DX | 0000 *DI | 6806 *ES | 03F0 | |


```

03F0:6A76 9D          popf
03F0:6A77 0B DB        or bx, bx
03F0:6A79 C3          ret
03F0:6A7A 8B 3E 00 68  mov di, 6800
03F0:6A7E 0B FF        or di, di
03F0:6A80 78 06        bmi 6A88
03F0:6A82 E8 BF FF        call 6A44
03F0:6A85 E8 96 FF        call 6A1E
03F0:6A88 04 00        mov cx, #4
03F0:6A8B BF 0E 68      mov di, #680E
03F0:6A8E 8B 1D        mov bx, (di)
03F0:6A90 0B DB        or bx, bx
03F0:6A92 75 07        bne 6A9E
03F0:6A94 83 EF 04      sub di, #4
03F0:6A97 E2 F5        loop 6A8E
03F0:6A99 FA          cli
Boot! Load! Go! Continue! Stop! Step! Break! Unbreak! UnbreakAll! Probe!
cs:ip
popf >> Non maskable interrupt at 3F0:6A76

```

Figure 2. Disassembly.

F6 evaluates the typein as the load address for the Load command in the menu bar. Be sure to select a memory area that is not used by the 3C505 ROM.

F9 toggles the bottom sixteen lines of the tile area between the serial line display and tiles.

F10 causes the tiles to appear in the bottom sixteen rows of the tile area.

[Alt]-q quits.

APPENDIX E

3C505 USER SOFTWARE DISKETTE

The 3C505 EtherLink Developer's Software Diskette contains the 3C505 diagnostic program, the 3D Debugger, a collection of utility subroutines and definition files, and an example host program that uses the utilities. The diagnostic is discussed in Appendix C and the 3D Debugger in Appendix D. The utility subroutines are all written in assembly language and implement a range of functions that can be incorporated in resident device drivers or standalone programs.

All assembly language subroutines are coded using the Microsoft Macro Assembler (Version 3.00) conventions. In addition, many subroutines adhere to the object code, linkage, and function call conventions required by the Lattice C Compiler (Version 2.14). In the ASM files, you will see references to the Lattice C macro file SM8086.MAC (which is not on the 3C505 EtherLink Developer's Software Diskette). It defines the following macros:

- PSEG - defines start of program segment
- ENDPS - defines end of program segment
- DSEG - defines start of data segment
- ENDDS - defines end of data segment

If you do not use the Lattice C Compiler, define these macros to suit your particular needs. The file SM8086.MAC also defines the constant LPROG. If set, LPROG specifies that long pointers and far calls be used. Again, define it or remove it as needed.

The 3C505 EtherLink Developer's Software Diskette contains the following files:

IO.ASM

This file contains subroutines that perform programmed I/O to the 3C505. The subroutine IO_INIT should be called first with the base I/O address of the 3C505. The subroutines OUTPCB and INPCB implement the Primary Command Block (PCB) protocol described in Chapter 4 of the Software Developer's Manual. There are also subroutines that manage the state of the Host Control Register.

DMA.ASM

This file contains all DMA related subroutines and global variables for an IBM PC or IBM AT host. The subroutine DMA_INIT should be called first with the DMA channel number to use (0-3 on the IBM PC plus 5-7 on the IBM AT). DMA transfers and DMA completion interrupts are handled in this module.

INT.ASM

This file contains subroutines to handle 3C505 interrupts in an IBM PC or IBM AT host. The subroutine INT_INIT should be called first with the interrupt vector number to use (channel 3-7 are allowed on the IBM PC plus 9-15 on the IBM AT). There are also subroutines to enable, disable, and acknowledge interrupts in this file.

CMD.ASM

Command Register interrupts are serviced in this module. The subroutine CMD_INIT must be called first to initialize buffer pointers. When a Command Register interrupt is received by the main interrupt handler, CMD_INT is called to read and locally buffer the PCB.

UTIL.ASM

This contains general purpose utility subroutines callable from Lattice C programs.

HOSTIO.DEF

This is the definition file that equates names to the 3C505 registers and to each bit in each register. Equates are also included for the DMA and interrupt controller registers in the IBM PC and IBM AT.

ADAPTER.DEF

This definition file equates names to the 3C505 soft interrupt vectors 80H-88H. Subfunctions for each vector are also defined.

PCB.DEF

This is the definition file that contains the assembly language structures for each PCB from Chapter 3.

PCB.H

Definition file for C-language programs that contains the structures for each PCB.

APPENDIX F

REVISION 2.0 ROM

1. Configure Adapter Memory

In Revision 1.0, certain minimums must be used in the Configure Adapter Memory command as detailed below. In Revision 2.0, the minimums are enforced and the default values are changed.

| | | Rev 1.0 | Rev 2.0 | | |
|---------------|------|---------|---------|-----|-------|
| | Size | Default | Default | Min | Notes |
| pcb_queue | 64 | 5 | 10 | 2 | e |
| rcv_pcb_queue | 14 | 32 | 20 | 2 | e |
| mc_list | 6 | 0 | 0 | 0 | |
| frame_desc | 22 | 40 | 20 | 2 | a,d |
| rcv_buffer | 1536 | 40 | 20 | 2 | d |
| prog_table | 8 | 10 | 10 | 1 | |
| rcv_buf_desc | 10 | 40 | 20 | 2 | b |

- a. The number of Frame Descriptors should normally equal the number of receive buffers.
- b. The number of Receive Buffer Descriptors allocated always equals the number of Receive Buffers specified.
- c. Revision 2.0 has an additional Receive Buffer Queue (12-bytes per entry). The number of entries in the queue always equals the number of Receive Buffers specified plus 1.
- d. The value is the minimum that the firmware will accept but is really not sufficient for network operation. At least 4 Frame Descriptors and Receive buffers should be allocated.
- e. One entry in this circular queue is wasted to simplify queue-empty and queue-full checking.

2. Timeout Values

The maximum timeout value that can be specified in PCB commands and system ROM service routines is increased in Revision 2.0 to 32767 ticks (10ms per tick) from 127 in Revision 1.0.

3. Timestamp and Timer Resolution

Revision 2.0 corrects a problem where the high word of the doubleword timestamp in the receive packet response PCB is wrong. Also, the timer resolution is changed to 15us per tick instead of 25us.

4. **DMA Downloading Programs**

The Adapter downloads programs using DMA utilities in ROM; i.e., int 80H function 5. This function only initiates the DMA transfer and does not wait for it to complete. In Revision 1.0, a successful download response is sent immediately after DMA initiation without waiting for DMA completion status. Revision 2.0 waits for download complete before sending the response PCB.

5. **Zero Offset Problem for Downloaded Programs**

In Revision 1.0, the first downloaded program is guaranteed to be executed with IP set to zero but subsequent downloaded programs will not. If the IP plus the size of the program's code segment is greater than 0ffffH, then the program will not work. Revision 2.0 solves this problem.

6. **Receive/Return Packet functions**

In Revision 1.0, successive Receive Packet (Int 81, function 2) calls without a Return Buffer (Int 81, function 3) will always yield the same receive buffer. Also, the Return Buffer call will always attempt to return the first receive buffer in the queue. Revision 2.0 allows any combination of Receive and Return packets.

7. **PCB Formats**

The following PCB's are changed in Revision 2.0; please refer to chapter 3 for more detail:

- a. Download Program (PCB 0DH). The offset:segment words have been removed.
- b. Download Program Response (PCB 3DH). The offset:segment of the downloaded program is returned.
- c. Adapter Info (PCB 11H). This is a new PCB command that retrieves general status information from the 3C505.
- d. Adapter Info Response (PCB 41H). This is a new response PCB that contains information such as revision ID, memory size, and ROM checksum value.
- e. Network Statistics Response (3AH). The counters for the number of received and transmitted packets are now double word values.
- f. Self-Test Response (3FH). The self-test command now performs ROM checksum, RAM test, and 82586 loopback tests. The response returns error information when any test fails.

8. Interrupt Vector Services

The following service routines are changed in Revision 2.0; please refer to Chapter 3 for detail:

- a. Int 80H, function 0BH: Check DMA Done. This new function allows a program to poll for DMA completion of a DMA initiated with Int 80H function 5 or 7.
- b. Int 82H, function 01H: Configure Adapter Memory. New defaults and minimums noted in item #1 of this Appendix are in effect.
- c. Int 82H, function 05H: Set LEDs. New function giving downloaded programs ability to turn off/on LEDs without conflicting with ROM usage.
- d. Int 82H, function 06H: Get Adapter Info. New function which returns general adapter information such as revision level, amount of memory, free memory pointers, and ROM checksum value.

9. TEST Jumper Usage

The state of the TEST jumper on the Adapter is represented by the SWTC flag in the Adapter Status Register. When the TEST jumper is set to one, the Revision 2.0 ROM code will:

- a. Ignore powerup memory test error. Memory errors detected during powerup normally prevent the adapter from entering the main ROM idle loop. Ignoring errors is useful when using ICE systems that need to modify the NMI vector location in order to operate.
- b. Ignore ROM checksum error. During ROM development, it is convenient not to checksum since the code is changing frequently.
- c. Install 3D interrupt vectors. The interrupt vectors known as "exceptions" (basically INT 0 to 7) and all unused interrupt vectors are made to point to the 3D slave in the Revision 2.0 ROM. When an exception occurs, 3D will become active and attempt to communicate with the 3D Debugger program.

10. Configure 82586 Receive Mode

The Configure 82586 Receive Mode function 2 of the INT 82H service routines is not correct. In the Revision 1.0 implementation, an ES:BX pointer to a Configure 82586 PCB command is expected instead of the receive mode in register BX. In Revision 2.0, this will be fixed to match the documentation.

11. Receive Packet PCB Timeout

In the Receive Packet PCB there is a timeout quantity which represents the maximum time to wait for a packet before sending a response PCB. In Revision 1.0, the timeout may never occur depending on the timeout value and the Adapter system time at the PCB was processed. To flush the PCB from the receive packet queue, use the soft reset function. This problem will be corrected in Revision 2.0.

12. Loopback Mode

The LPBK bit in the Adapter Control Register (described in section 3.5 of the Developer's Manual) controls loopback mode at the 8023 Manchester Code Converter. LPBK is active low. This means that if LPBK is clear, then loopback mode is enabled. Be sure to set LPBK for normal network operation.

APPENDIX G

REVISION 3.0 ROM

1. Adapter Selftest Command

In Revision 2.0, the Adapter Selftest command (PCB 0FH) hangs when testing an Adapter with 128kbytes of memory. A workaround is to load the Adapter with 256kbytes of memory. The Revision 3.0 firmware eliminates this problem and can test the latest revision 3C505, containing up to 512kbytes of memory.

2. Transmit Packet Command

The Transmit Packet command (PCB 09H) in Revision 2.0, after signaling acceptance of the command, waits up to 30ms for the host to download transmit packet data. Then, the packet is transmitted whether the data download is completed or not. The Revision 3.0 firmware will now allow 50ms instead of 30ms and will not transmit the packet if the download has not completed. If the download is not complete, response PCB 39H is returned to the host with error code -2.

3. Get Adapter Information Command

When an Adapter-resident program uses INT 82 function 6 to get adapter status information, the data is incorrectly returned in the buffer pointed to by register pair DS:BX. The Revision 3.0 firmware returns the data in the buffer pointed to by register pair ES:BX as documented.

4. Packet Processor

One of the functions of the firmware Packet Processor INT 86 is to upload receive packets to the host if there is an outstanding Receive Packet command (PCB 08H). The Revision 2.0 firmware, however, might not upload packets in the same order in which they were received. In Revision 3.0, the order of packets is preserved when being uploaded to the host.

5. Adapter LEDs

In the latest revision 3C505 adapter, the bits in the Adapter Control Register (ACR) that control the LEDs are inverted from earlier revision adapters. Downloaded programs that modify the ACR should keep this in mind. The Revision 3.0 firmware detects whether the LED control bits should be inverted and will do so accordingly.

6. Poweron Selftest (POST)

The first 6kbytes of RAM is "system" memory needed by the firmware and the rest of RAM is "buffer" memory. In Revision 3.0 if buffer memory errors are found during POST, the firmware will loop infinitely flashing both LEDs at 800 hz. If an 82586 initialization or loopback test error occurs during POST, the firmware will loop flashing the LEDs at 700 hz.