Net186™ Demonstration Board User's Manual

Net186™ Demonstration Board User's Manual, Release 1.0

1.0

Advanced Micro Devices, Inc.
5204 E. Ben White Blvd.
Austin, TX 78741-7399

Contents

About the Net186™ Demonstration Board

Chapter 1

Quick Start

Chapter 2

Demonstration Board Functional Description

1.0

Chapter 3

Product Support

Appendix A

Schematics and Board Bill of Materials

Appendix B

PCnet Family History

Appendix C

References

Appendix D

PAL Source File Listing

Appendix E

EEPROM Contents

Appendix F

TCP/IP Primer

Index

1.0

List of Figures

List of Tables

1.0

About the Net186™ Demonstration Board

The AMD® Net186™ Demonstration Board is a small (3.5" x 3.5") demonstration board combining AMD's Am79C961A PCnet™-ISA II Ethernet controller, Am186ES microcontroller, PALCE22V10, and Am29F400 Flash memory. It demonstrates how simple, low-cost Ethernet capability can be added to a wide variety of embedded networking applications. See Figure 2-1 on page 2-2 for a block diagram of the demonstration board.

Typical applications of the demonstration board design include low-cost, managed Ethernet hubs, "smart house" components, industrial control, point-of-sale terminals, and software development tools such as ROM emulators. An entire new class of applications known as "net appliances", ranging from electric meters to coffee pots, could also use a design similar to that of the Net186 demonstration board.

The Net186 demonstration board uses the Am186™ES microcontroller. The Am186ES microcontroller integrates peripherals such as twelve 16-bit memory chip-select controllers, two asynchronous serial controllers, three timers, 32 programmable I/Os, an interrupt controller, and a watchdog timer to increase system functionality while reducing the overall cost. The memory controller also supports a glueless connection to SRAM, Flash memory, and EEPROM. The Am186ES microcontroller also features an innovative bus design that allows the processor to run at nearly twice the speed of standard 80C186 processors while using commodity memory devices.

The Net186 demonstration board is designed with a 104-pin Am186 expansion interface that provides access to the Am186ES microcontroller signals. For more information about the Am186 expansion interface, see “Am186 Expansion Interface” on page 2-12.

Demonstration Board Features

The Net186 demonstration board provides the following features:

- Am186ES microcontroller
- PCnet-ISA II single-chip Ethernet controller
- E86™ family boot monitor (E86MON™) board-resident utility

  Information on invoking and using the E86MON software is provided in the *E86MON™ Software User's Manual* provided in your kit.
- 512 Kbyte SRAM
- 512 Kbyte Am29F400-70 Flash memory
- Am186 104-pin expansion interface
- Two RS-232 serial ports with DB-9 connectors
- One 10-Mbit/s 10Base-T port for twisted-pair Ethernet connection with an RJ-45 connector
- Activity LED indicators for PIO signals and the Ethernet controller
- Reset circuitry

1.0

Software

The Net186 demonstration board is supplied with three software packages.

E86MON — A general-purpose interactive monitor program that allows you to load, run, and debug programs from an attached PC. See the *E86MON™ Software User's Manual* included in your kit for more information.

USNET — A TCP/IP stack with associated applications from US Software that have been ported to the Net186 demonstration board for evaluation purposes. This stack allows you to run common applications like FTP, Telnet, and E86Web on live networks for demonstrations. TCP/IP stacks are discussed in more detail in Appendix F, "TCP/IP Primer".

E86Web — An embedded web-server application using US Software's Internet Access Package (IAP). This application allows the Net186 demonstration board to return simple web pages to a web browser running on another machine.

Information on how to configure and run the sample applications is provided in the README.TXT file on the 3½" disk named Example Applications that is included in your kit.

1.0

Documentation

The *Net186™ Demonstration Board User's Manual* provides information on the design and function of the Net186 demonstration board. Detailed instructions for using the E86MON software are provided in the *E86MON™ Software User's Manual* included in your kit. The demonstration board is shipped with the E86MON software installed in the on-board Flash memory.

About This Manual

Chapter 1, "Quick Start" provides implementation and installation information for the demonstration board and instructions for invoking the E86MON software. Detailed information on using the E86MON software is provided in the *E86MON™ Software User's Manual* included in your kit.

Chapter 2, “Demonstration Board Functional Description” contains descriptions of the basic sections of the demonstration board including: Flash memory, serial ports, clock and reset logic, expansion interface, Ethernet controller, LED indicators, and power-supply circuitry.

Chapter 3, “Product Support” provides information on reaching and using the AMD Corporate Applications technical support services, product information available through AMD’s WWW and FTP sites, and support tools for the embedded E86 and PCnet families.

Appendix A, “Schematics and Board Bill of Materials” contains the schematics and Bill of Materials (BOM) for the Net186 demonstration board.

Appendix B, “PCnet Family History” contains information about AMD’s PCnet family of networking products.

Appendix C, “References” contains names of various publications and web site addresses that provide more information about Ethernet and general networking.

Appendix D, “PAL Source File Listing” contains the contents of the PAL source file included in your kit.

Appendix E, “EEPROM Contents” contains the text file that shows the contents of the EEPROM used on the Net186 demonstration board.

Appendix F, “TCP/IP Primer” contains a brief explanation of protocol stacks and information about protocol stack vendors.

Suggested Reference Material

For information on ordering the literature listed below, see Chapter 3, “Product Support”.

- *Am186™ES and Am188™ES Microcontrollers Data Sheet*
  Advanced Micro Devices, order #20002
- *Am186™ES and Am188™ES Microcontrollers User’s Manual*
  Advanced Micro Devices, order #21096
- *Am186™ and Am188™ Family Instruction Set Manual*
  Advanced Micro Devices, order #21267
- *Am79C961A PCnet™-ISA II Jumperless, Full Duplex Single-Chip Ethernet Controller for ISA Data Sheet*
  Advanced Micro Devices, order #19364
- *FusionE86℠ Catalog*
  Advanced Micro Devices, order #19255
- *FusionE86℠ Development Tools Reference CD*
  Advanced Micro Devices, order #20158

For current application notes and technical bulletins, see our WWW page at **http://www.amd.com**.

1.0

Documentation Conventions

The *Net186™ Demonstration Board User's Manual* uses the conventions shown in Table 0-1 (unless otherwise noted).

Table 0-1. Notational Conventions

Symbol	Usage
Boldface	Indicates that characters must be entered exactly as shown, except that the alphabetic case is only significant when indicated.
`Typewriter face`	Indicates computer text input or output in an example or listing.

1.0

Chapter 1

Quick Start

This chapter provides information that will help you quickly set up and start using the Net186 demonstration board.

The Net186 demonstration board is supported by the E86MON software. The E86MON software enables you to load, run, and debug programs on the Net186 demonstration board. For more information on using the E86MON software, refer to the *E86MON™ Software User's Manual* included in your kit.

For information on how to:

1.0

Connecting to a PC

The procedure in this section describes how to connect the Net186 demonstration board to a PC for use as a standard Am186 microcontroller evaluation board using the E86MON software over an RS-232 port.

For information on attaching the Net186 demonstration board using the PCnet-ISA II Ethernet controller, see the README.TXT file on the 3½" disk named Example Applications that is included in your kit.

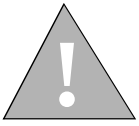
Follow the steps below to connect the Net186 demonstration board to your PC.

Installation Requirements

The items listed below are necessary to install and run the Net186 demonstration board:

- PC with an available COM port
- Terminal emulation software (such as Microsoft Windows® Terminal or ProComm Plus) that supports ASCII file transfers, software flow control (Xon/Xoff), and *send break* capability
- Power source for universal power supply

Board Installation

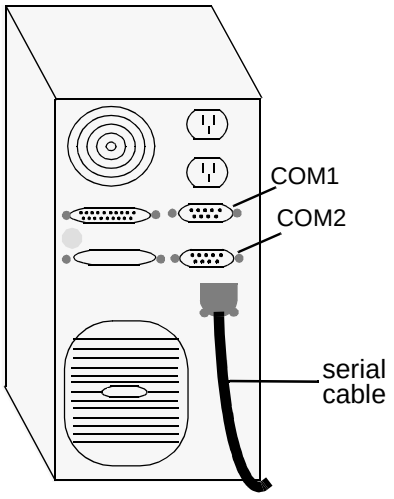


CAUTION: As with all computer equipment, the Net186 demonstration board may be damaged by electrostatic discharge (ESD). Please take proper ESD precautions when handling any board.

1. Remove the board from the shipping carton. Visually inspect the board to verify that it was not damaged during shipment.

1.0

2. Connect the Net186 demonstration board's DB-9 serial port (J3) to an available COM port. Use the serial cable included in the Net186 demonstration board kit and note that a DB-9 to DB-25 serial connector adapter is provided if your host system requires it. The pinout of the demonstration board's serial connector is shown in Figure 2-4 on page 2-9.



DANGER: Make sure the power supply is *not* plugged into an electrical outlet before connecting it to the Net186 demonstration board.

3. Connect the power supply to the barrel connector on the Net186 demonstration board.
4. Apply power to the demonstration board by connecting the power supply to an electrical outlet. When the demonstration board is powered up, the LEDs (CR5–CR12) should flash in an oscillating pattern for a three-second interval.

CAUTION: If using your own power supply, ensure that it is a 5-V supply. Using a 9-V supply will permanently damage the board.

5. Invoke the terminal emulation program at 19200 baud or higher, no parity, 8 data bits, and 1 stop bit; enable the software flow control (Xon/Xoff), if supported.

   Note that the Net186 demonstration board can reliably autobaud at rates up to 115 Kb/s. You can use these higher baud rates if your PC supports them.

6. Reset the demonstration board by depressing and releasing the RESET switch located in the upper right corner of the demonstration board. The LEDs on the board (CR5–CR12) will flash in an oscillating pattern for three seconds, as they did upon power up.

   During the three-second period while the LEDs are flashing, type an **a** in the terminal window to ensure that the E86MON software uses the correct baud rate. When the E86MON software receives an **a**, it adjusts its baud rate (if necessary) and displays the welcome message and prompt.

   ```
   Welcome to AMD's EMon 186!      (? <Enter> for help)

   es86mon:
   ```

   If an **a** key is not received, the E86MON software jumps to the user-application bootstrap vector, located at F7FF0h (see the *E86MON™ Software User's Manual* included in your kit for more information). Depressing and releasing the RESET switch gives you another opportunity to type an **a**.

7. To display the version of the E86MON software and the commands available, type **?** and press Enter.
8. You can now specify one of the commands from the E86MON software command menu or run one of the sample applications included in your kit.

   For detailed information about using the E86MON software, refer to the *E86MON™ Software User's Manual* included in your kit.

   For information about the Net186 sample applications, see "Net186 Sample Applications" on page 1-6.

Table 1-1. Installation Troubleshooting

Problem	Solution
Nothing happens when pushing the RESET button.	Sometimes it is difficult to make a good connection when pushing the small RESET button. Try removing the power supply from the AC electrical outlet and disconnecting and reconnecting the power supply. You should also verify that the Flash chip select switch (SW2) is pointing in the correct direction. See Figure 2-8 on page 2-15 for more information. The LEDs will flash in an oscillating pattern when the reset is successful.
The computer does not respond with the E86MON software prompt.	Reset the board by pressing the RESET switch and typing an **a** while the LEDs are flashing in an oscillating pattern. If this does not work, verify the power, check the cables, etc.
After typing **a** during reset, the terminal emulation software displays unreadable characters.	Check the baud rate setting for the terminal emulation software. It should be set to 19200. Also check the word length (8), stop bits (1), parity (N), and turn off any hardware flow control.
After a processor reset, the LEDs do not flash in the expected pattern.	Check that the power LED is on and the correct voltage is supplied to the board. Ensure that the polarity of the power connector is correct.
The terminal emulation program locks up the software or PC.	Check the COM port connection with the target board. Make sure that the same COM port is selected in the terminal emulation software. In some PCs if the correct COM port is not specified, the software will fail to function—it will lock in a continuous loop waiting for an answer from the incorrect serial port.
The power LED does not turn on with power.	Immediately disconnect the power supply. Ensure that the polarity of the power connector is correct. This is a very serious failure of the hardware. If the power source is connected incorrectly, the board will be permanently damaged.
There is a problem you cannot resolve.	Contact the AMD Corporate Applications technical support services (see Chapter 3, “Product Support” for phone numbers and more information.)

Net186 Sample Applications

Two sources of sample applications are available for the Net186 demonstration board and are available in your kit:

- The 3½" disk containing the E86MON demonstration board utility provides sample applications in the /OUT and /SAMPLES directories.

  For more information about loading and running the E86MON sample code, see the *E86MON™ Software User's Manual* included in your kit.

- The 3½" disk containing the Net186 demonstration board applications provides a demonstration version of US Software's TCP/IP stack USNET, as well as a sample web server application.

  To load the demonstration library and application, refer to the README.TXT file on the disk.

1.0

For More Information...

If you need more information about:

- Net186 demonstration board hardware,
  see Chapter 2 of this manual
- The E86MON software,
  see the *E86MON™ Software User's Manual* included in your kit
- The Am186ES microcontroller,
  see the *Am186™ES/ESLV and Am188™ES/ESLV Microcontrollers Data Sheet*
  and the *Am186™ES and Am188™ES Microcontrollers User's Manual*
- The PCnet-ISA II Ethernet controller,
  see the *Am79C961A PCnet™-ISA II Jumperless, Full Duplex Single-Chip Ethernet Controller for ISA Data Sheet*
- The Am29F400 Flash,
  see the *Am29F400 Data Sheet*
- The PALCE22V10 device,
  see the *PALCE22V10 PAL Devices Book and Design Guide*
- Network products,
  see the *Networking Products Literature and Software CD*

1.0

Chapter 2

Demonstration Board Functional Description

The Net186 demonstration board shows how easy it is to create a low-cost, embedded Ethernet solution using the Am186ES microcontroller and the Am79C961A PCnet-ISA II single-chip Ethernet controller. In addition to the Am186ES microcontroller and PCnet-ISA II Ethernet controller, the Net186 demonstration board contains a single Am29F400 4-Mbit Flash memory and a single 4-Mbit SRAM. The Flash memory is shipped with the E86MON software and demonstration application software and can be loaded with user-application programs. The SRAM will typically contain temporary user data in addition to the Ethernet packet data.

Read the following sections to learn more about the Net186 demonstration board hardware:

Hardware Block Diagram and Memory Map

One of the key features of the Net186 demonstration board is the near glueless interface between the Am186ES microcontroller and the PCnet-ISA II Ethernet controller. Figure 2-1 shows a block diagram of the Net186 demonstration board. Some elements, such as the RS232 serial ports and the Ethernet connection, were left off for clarity.

Note that a single PAL22V10 device is used to connect the processor to the Ethernet controller. In fact, the logic inside the PAL22V10 will fit easily into a PAL16V8, or could be implemented with SSI logic gates. The PAL22V10 was used on the Net186 demonstration board for user-expansion purposes.

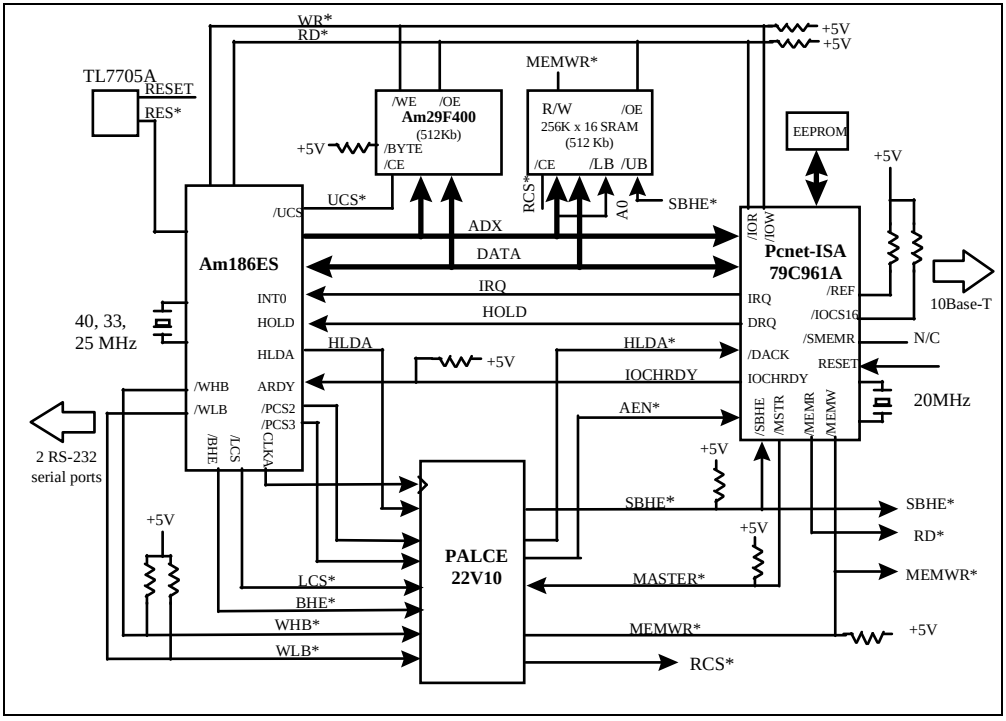


Figure 2-1. Block Diagram of the Net186 Demonstration Board

1.0

Table 2-1 and Table 2-2 show where Flash memory, SRAM, and the PCnet-ISA II Ethernet controller are configured in Am186ES microcontroller memory and I/O space. Note that the PCnet-ISA II Ethernet controller I/O base address can be modified by editing the setup EEPROM.

Please refer to the PCnet-ISA II Ethernet controller documentation for I/O register mapping for that device. Also, please refer to the *Am186™ES and Am188™ES Microcontrollers Data Sheet* for locations and use of built-in peripherals, registers, and logic.

Table 2-1. Memory Space Configuration

CPU Address	Memory Space
0x80000–0xFFFFF	Flash memory (512 KByte)
0x00000–0x7FFFF	SRAM (512 KByte)

Table 2-2. I/O Space Configuration

CPU Address	I/O Space
0x200–0x21F	PCnet-ISA II Ethernet controller registers
0xFF00–0xFFFF	Am186ES peripheral control block

The Am186ES microcontroller boots at the very top of addressable memory. The first code fetch is done at address FFFF0h, and asserts the Upper Chip Select ($\overline{\text{UCS}}$) line. On the Net186 demonstration board, this line is configured to enable the Flash memory. The Flash memory is available exclusively to the Am186ES microcontroller, while the SRAM can be accessed by both the Am186ES microcontroller and the PCnet-ISA II Ethernet controller.

The PCnet-ISA II Ethernet controller actually takes over the processor local bus, and with DMA control, transfers Ethernet packet data directly to and from the SRAM. Note that the Net186 demonstration board comes equipped with 512 Kbyte of Flash memory and 512 Kbyte of SRAM. Most applications will not need this much memory.

Net186 Demonstration Board Parts List

Table 2-3 lists the Net186 demonstration board part numbers, parts, and where to find more information about the individual parts.

Table 2-3. Net186 Demonstration Board Parts List

Part Number	Description	For more information, see
CR1–CR12	Signal LEDs	Page 2-16
J1	Power connector	Page 2-18
J2, J3	Serial ports	Page 2-9
J5	RJ-45 Ethernet connector	Page 2-10
P1, P2	Am186 Expansion interface	Page 2-12
SW2	UCS chip select switch	Page 2-15
U10	Am186ES 40-MHz microcontroller	Page 2-5
U11	Flash memory	Page 2-8
U2	PAL22V10	Page 2-19
U4	Reset controller (TI TL7705ACD)	Page 2-12
U5, U6	RS-232 driver/receiver devices (MAX232)	Page 2-9
U7	PCnet-ISA II Ethernet controller	Page 2-10
U9	SRAM	Page 2-8
Y1	40-MHz fundamental mode crystal	Page 2-12
Y2	20-MHz fundamental mode crystal	Page 2-12

1.0

Am186ES Microcontroller Implementation

The Net186 demonstration board is shipped with an Am186ES microcontroller that operates at 40 MHz. Although not supported by the on-board crystal, the Am186ES microcontrollers are also available in 20-, 25-, and 33-MHz operating frequencies. The Net186 demonstration board also includes a PCnet-ISA II Ethernet controller. See Figure 2-3 on page 2-7 for a block diagram detailing the PCnet-ISA II Ethernet controller's functionality.

The Am186ES microcontrollers are designed to meet the most common requirements of embedded products developed for the communications, office automation, mass storage, and general embedded markets. Specific applications include feature phones, cellular phones, PBXs, multiplexers, modems, disk drive controllers, hand-held and desktop terminals, fax machines, line cards, managed hubs, and industrial control. Refer to the *Am186™ES/ESLV and Am188™ES/ESLV Microcontrollers Data Sheet* for more information on the specific features of the Am186ES microcontrollers.

See Figure 2-2 on page 2-6 for a block diagram detailing the Am186ES microcontroller's functionality.

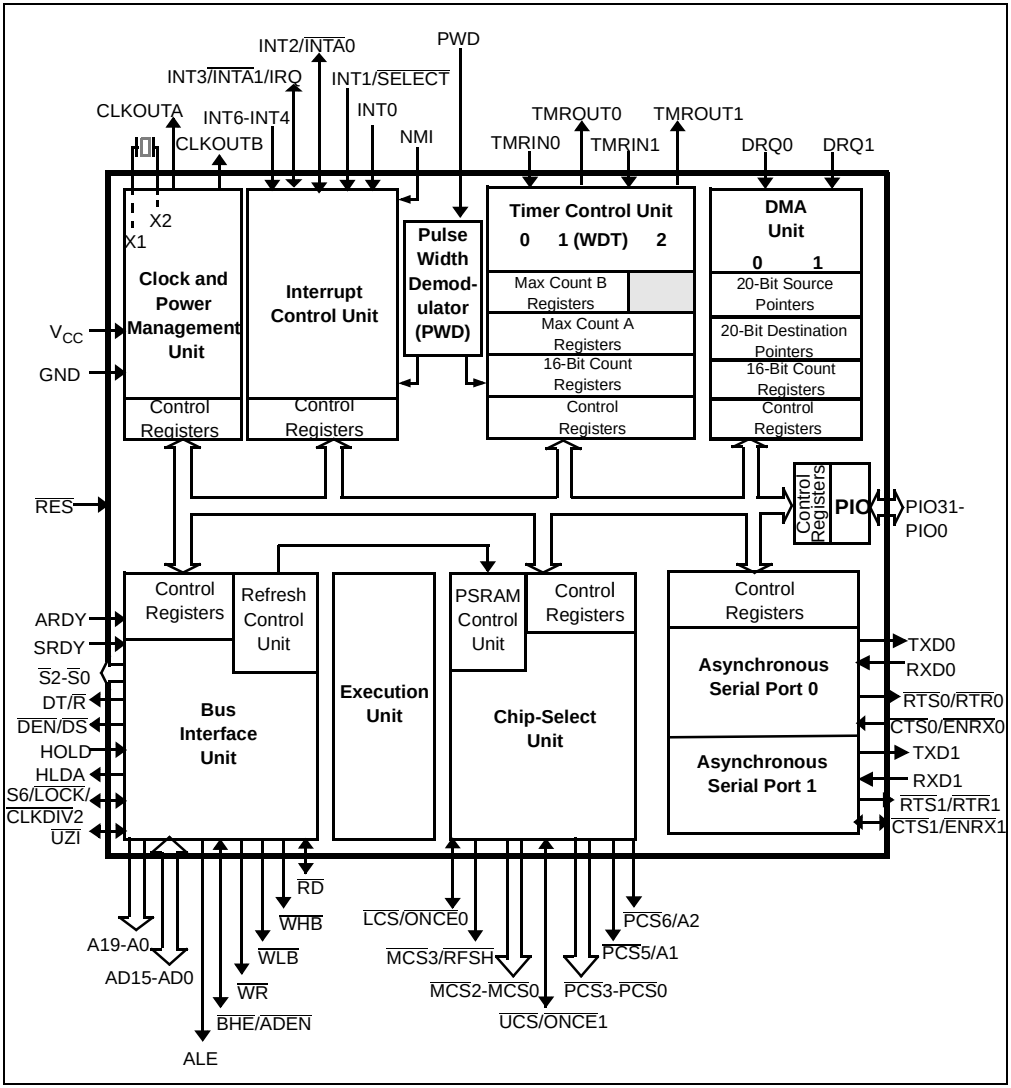
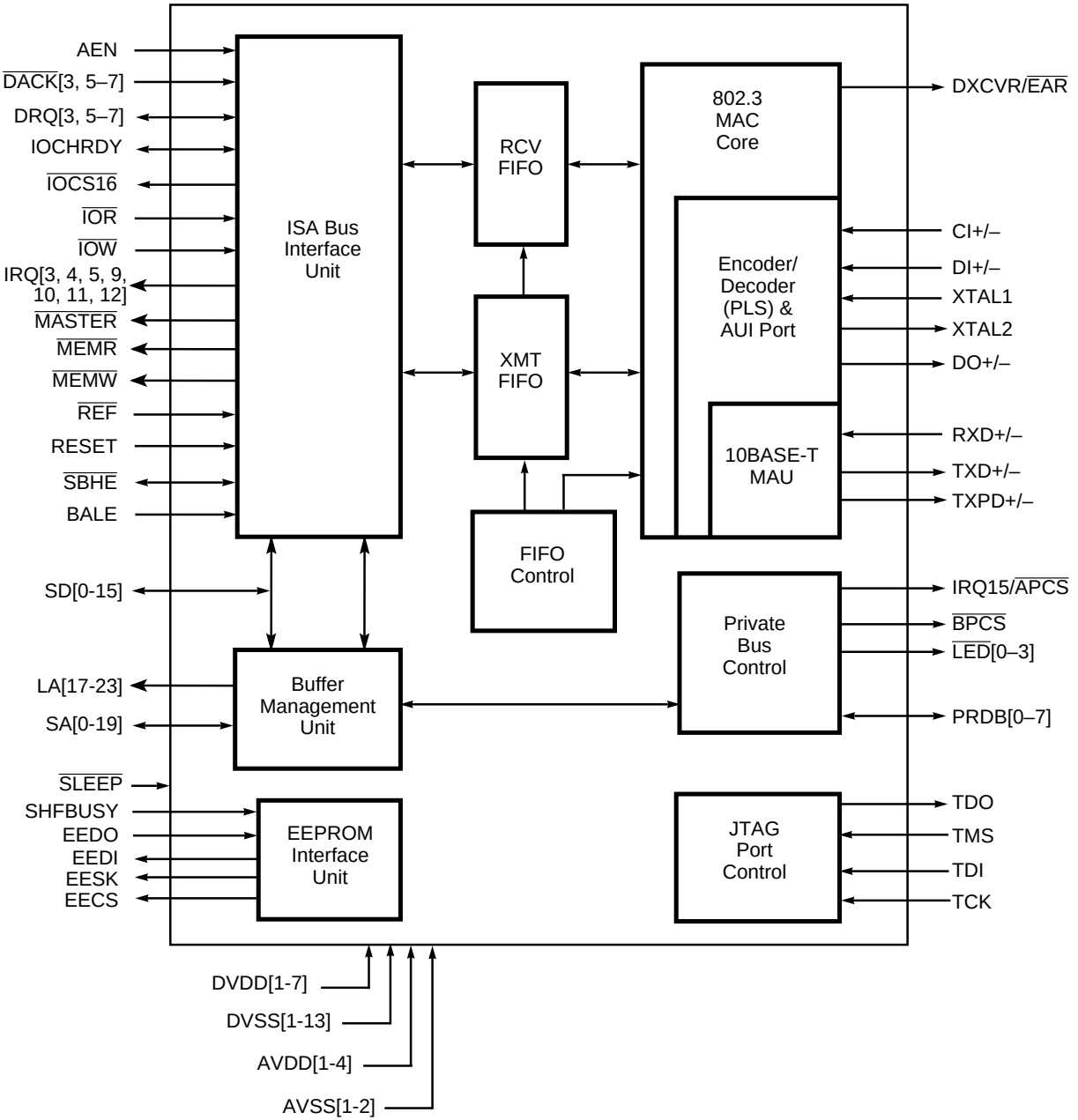


Figure 2-2. Am186ES Microcontroller Block Diagram

1.0

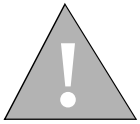

AEN
DACK[3, 5–7]
DRQ[3, 5–7]
IOCHRDY
IOCS16
IOR
IOW
IRQ[3, 4, 5, 9, 10, 11, 12]
MASTER
MEMR
MEMW
REF
RESET
SBHE
BALE
SD[0-15]
LA[17-23]
SA[0-19]
SLEEP
SHFBUSY
EEDO
EEDI
EESK
EECS
DVDD[1-7]
DVSS[1-13]
AVDD[1-4]
AVSS[1-2]
ISA Bus Interface Unit
RCV FIFO
XMT FIFO
FIFO Control
Buffer Management Unit
EEPROM Interface Unit
802.3 MAC Core
Encoder/ Decoder (PLS) & AUI Port
10BASE-T MAU
Private Bus Control
JTAG Port Control
DXCVR/EAR
CI+/–
DI+/–
XTAL1
XTAL2
DO+/–
RXD+/–
TXD+/–
TXPD+/–
IRQ15/APCS
BPCS
LED[0–3]
PRDB[0–7]
TDO
TMS
TDI
TCK

ROM Space

The Net186 demonstration board contains on-board ROM space for use by the E86MON software and application code. This ROM space is implemented as an Am29F400 70-ns Flash memory device.

The Flash memory device is mapped to the upper region of addressable memory at 80000h to FFFFFh. The Flash memory device is organized as 256K x 16 bits and is connected to the $\overline{\text{UCS}}$ (Upper Memory Chip Select) signal of the microcontroller. After a valid reset, the Am186ES microcontroller fetches the first instruction from the Flash memory device by asserting $\overline{\text{UCS}}$ and driving the address bus with the value FFFF0h.

The E86MON software enables you to program the Flash memory device with specific types of hex files. Intel hex and Intel extended hex format files are supported. This software functionality is provided to eliminate the need to remove the Flash memory device.



CAUTION: Do not attempt to remove the TSOP Flash memory (U11) or SRAM device (U9) because doing so may cause damage to the board.

1.0

SRAM

The Net186 demonstration board utilizes SRAM for its read/write storage. The board provides 512 Kbyte of SRAM using a 70-ns device that is mapped from 00000h to 7FFFFh. The SRAM device is organized as 256K x 16 bits and is attached to the $\overline{\text{LCS}}$ (Lower Memory Chip Select) signal of the microcontroller. For every access to the above address range, the Am186ES microcontroller will assert $\overline{\text{LCS}}$.

RS-232 Serial Ports

The Net186 demonstration board provides two on-board RS-232 serial ports (J2 and J3) that are directly driven by the Am186ES microcontroller. The serial ports are equipped with DB-9 DCE connectors. The pin assignment for the DB-9 connectors is shown in Figure 2-4.

Traditionally, PCs have Data Terminal Equipment (DTE) ports which connect directly to the Data Communication Equipment (DCE) port on the Net186 demonstration board. A null modem cable is not required to connect a DTE port with a DCE port.

The RS-232 specification calls for signals that are driven at non-TTL levels. Single-chip RS-232 driver/receiver devices (MAX232, U5 and U6) are used to convert to and from the required voltages.

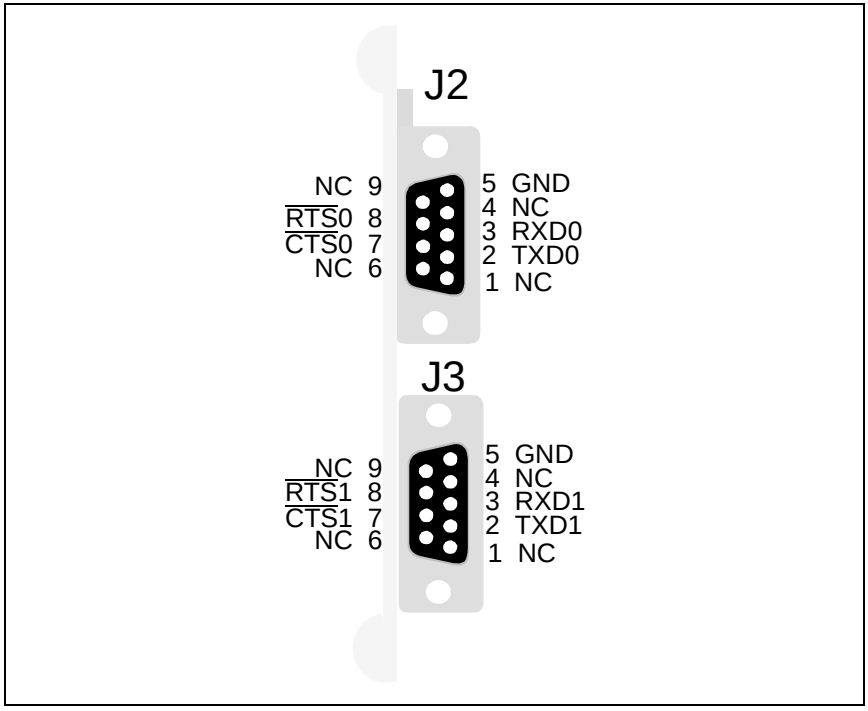


Figure 2-4. DB-9 Serial Connector Pinouts

PCnet-ISA II Ethernet Controller

The Net186 demonstration board provides one on-board 10BASE-T port (J5) that is directly driven by the PCnet-ISA II Ethernet controller. The 10BASE-T port is equipped with an RJ-45 connector. Figure 2-5 and Table 2-4 show the pin assignment and pin functions for the RJ-45 connector.

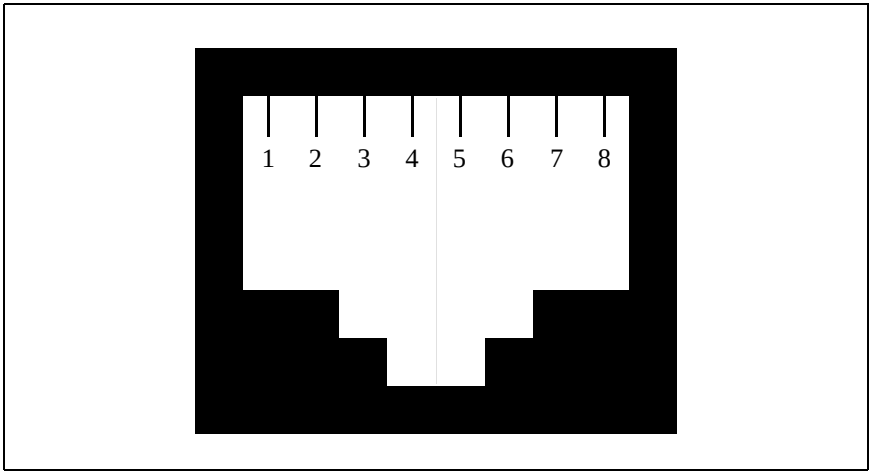


Figure 2-5. Front View of the RJ-45 Connector

Table 2-4. RJ-45 Connector Pin Functions

Pin Number	Function
1	TX+
2	TX-
3	RX+
4	Not used
5	Not used
6	RX–
7	Not used
8	Not used

10BASE-T Port

The PCnet-ISA II Ethernet controller provides an on-chip physical layer function so the only other device required to interface to the RJ-45 connector is a 10BASE-T filter with transformer.

The 10BASE-T interface supports the maximum cable length (100 meters) between a node and a hub. To link two stations through a 10BASE-T hub, simply use two straight-through cables: one cable connects the twisted-pair port on the Net186 demonstration board to the hub; the other cable connects the hub to the second station. Each end of the twisted-pair cable has a mating, RJ-45 type, eight-pin modular plug that connects to the twisted-pair jack of each station. Figure 2-5 and Table 2-4 on page 2-10 show the pin assignment and pin functions for the RJ-45 connector.

The *PCnet™-ISA II Ethernet Controller Hardware User's Manual*, order #19425, contains detailed information about the 10BASE-T interface.

Access Unit Interface (AUI) Port

The PCnet-ISA II Ethernet controller provides an Access Unit Interface (AUI) port. This AUI port, with the appropriate transceiver, allows connections to different Ethernet medias; for example, coaxial shielded cables for 10BASE2 and 10BASE5 connections and filter cables for 10BASE-F connections. Connection to the AUI port effectively bypasses the on-chip 10BASE-T transceiver.

NOTE: The AUI port is not supported by the Net186 demonstration board.

During initialization, the PCnet-ISA II Ethernet controller first checks for activity on its 10BASE-T port. If the controller determines there is activity on that port, a link will be established with the network. The AUI port will be ignored, even if there is a physical connection on the AUI port via a transceiver device.

If the PCnet-ISA II Ethernet controller determines there is no activity on the 10BASE-T port, the controller checks for activity on the AUI port. When the controller determines there is activity on the AUI port, a link is established with the network. The 10BASE-T port is ignored until the next PCnet-ISA II Ethernet controller initialization sequence.

When the PCnet-ISA II Ethernet controller determines there is no activity on either the 10BASE-T or the AUI port, the controller typically issues an error condition.

Clock and Reset Logic

The Am186ES microcontroller can be configured for either 1x or ½x clock mode. As configured on the Net186 demonstration board, the microcontroller is in 1x clock mode. The input is generated by a 40-MHz fundamental mode crystal (Y1) that is connected to the X1/X2 inputs of the microcontrollers, resulting in a 40-MHz system clock.

The PCnet-ISA II Ethernet controller is driven by a 20-MHz fundamental mode crystal (Y2). Y2 is configured in ½x clock mode, resulting in a 10-MHz controller clock frequency.

System reset is controlled by a voltage supply supervisor (TI TL7705ACD, U4). This device generates the processor's reset input, asserting the Am186ES microcontroller's $\overline{\text{RES}}$ pin for 13 ms when the RESET switch is depressed. The voltage supply supervisor also holds reset active when the power falls below 4.75 V.

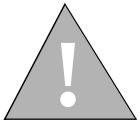
Am186 Expansion Interface

1.0

The Am186 expansion interface facilitates prototyping with external devices by using the Net186 demonstration board as the Ethernet and processor elements of an embedded design.

The Net186 demonstration board supports the PC/104 form-factor expansion-type connector for additional prototyping and testing. The traditional PC/104 signals are not present on the board; however, the Am186 expansion interface enables you to attach wirewrap or prototype boards that have the same standard physical interface.

The pinout of the expansion interface is shown in Figure 2-6 on page 2-13 and Figure 2-7 on page 2-14.



WARNING: The Am186 expansion connector is mechanically identical to the PC-104 standard. However, the Am186 expansion is not electrically compliant with the PC-104 standard and should not be used with PC-104 plug-on cards.

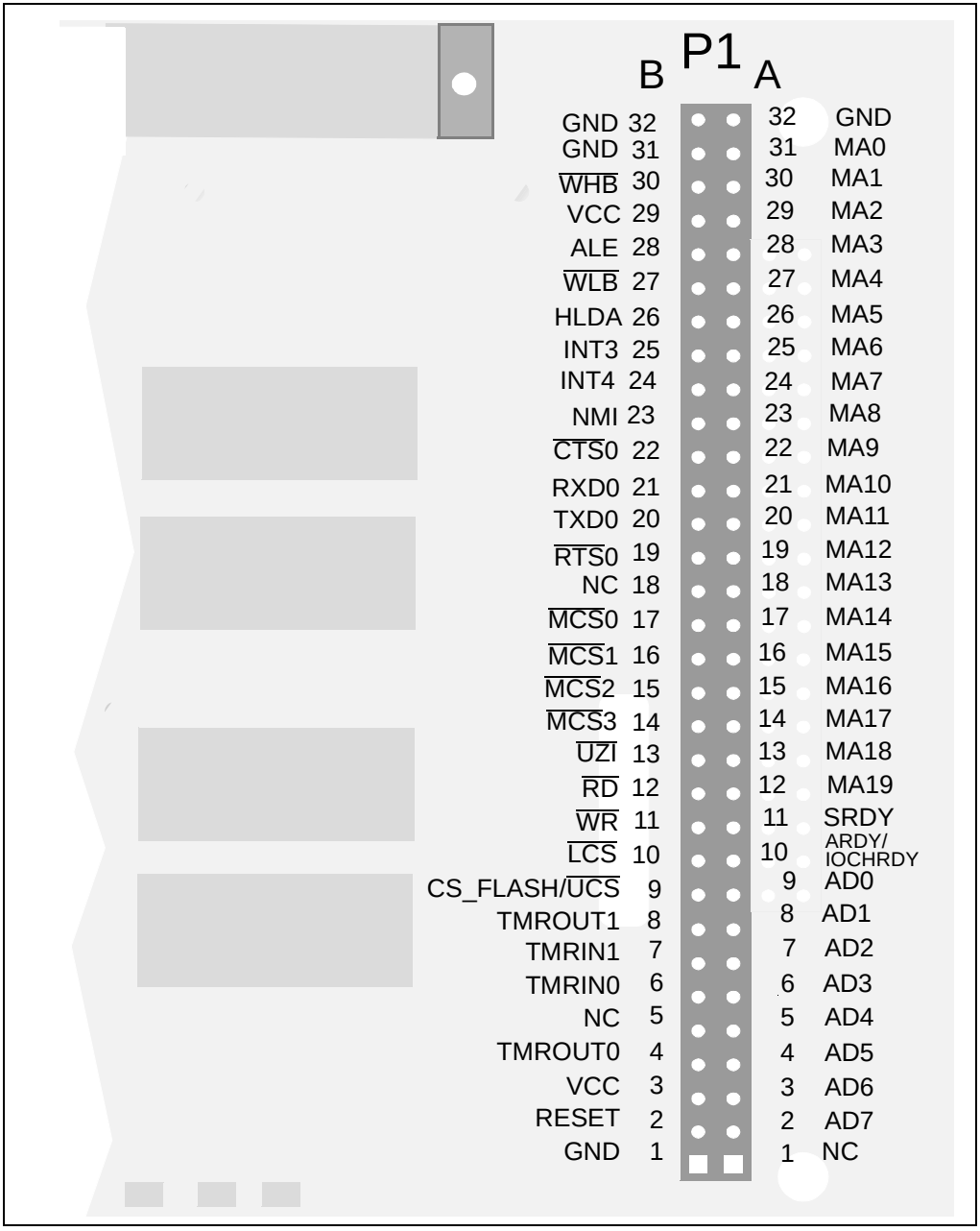


Figure 2-6. Am186 Expansion Interface Pinout (P1)

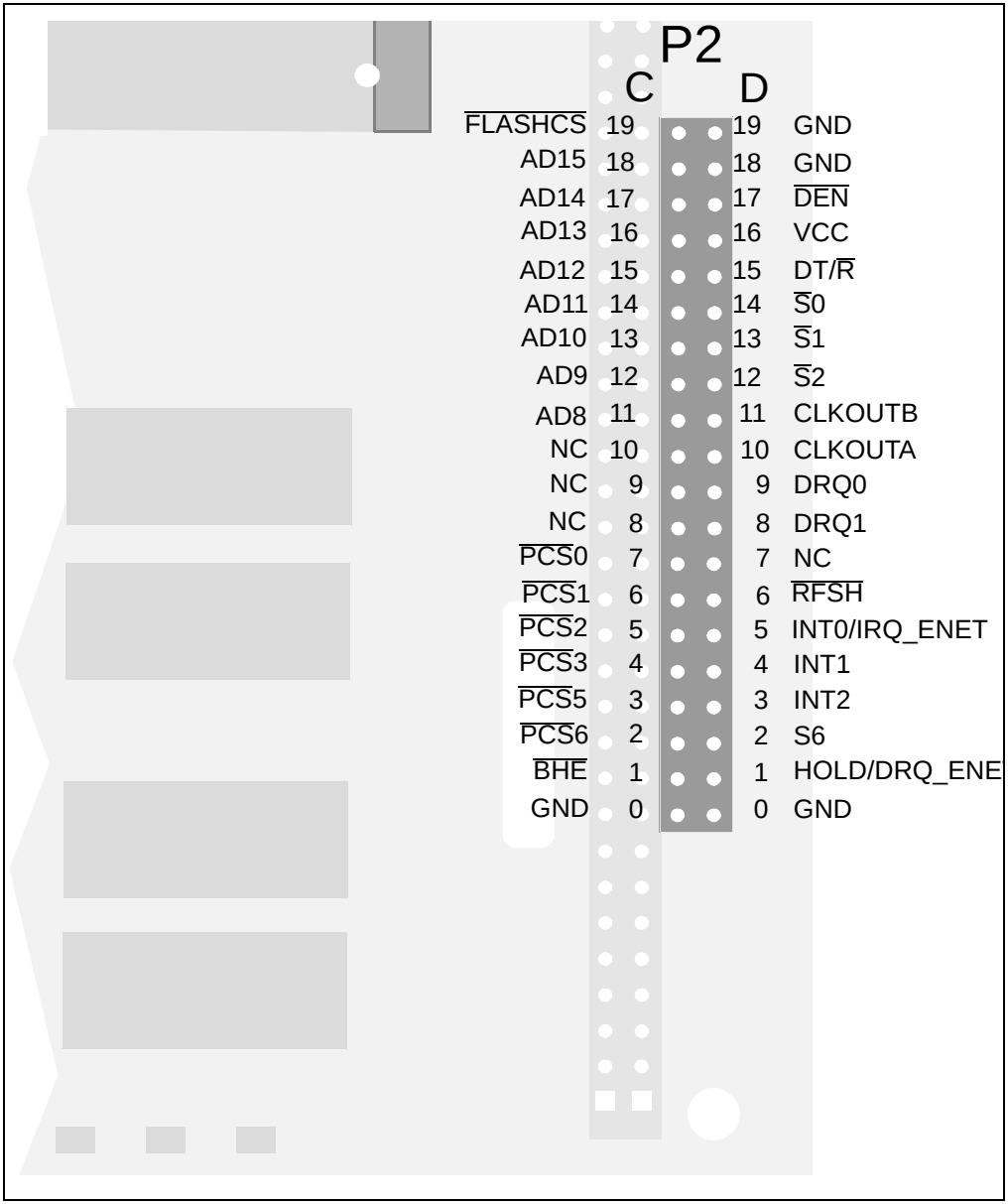


Figure 2-7. Am186 Expansion Interface Pinout (P2)

Flash Chip Select Switch

The E86MON software utility enables you to program the on-board Flash memory with your own application code; however, there may be situations when you want to test your application from reset.

To enable you to test your own application code, the Net186 demonstration board provides a switch that routes the on-board Flash memory chip select from the Am186 expansion interface connector, instead of from the Am186ES $\overline{\text{UCS}}$ signal. (The chip select for the on-board Flash memory has a weak pull-up to prevent the $\overline{\text{UCS}}$ signal, available on the Am186 expansion interface connector, as its chip select.)

For normal operation, the Flash chip select switch (SW2) should be set as shown in Figure 2-8. The Am186ES microcontroller's $\overline{\text{UCS}}$ signal will be connected to the on-board Flash memory's $\overline{\text{CE}}$ signal.

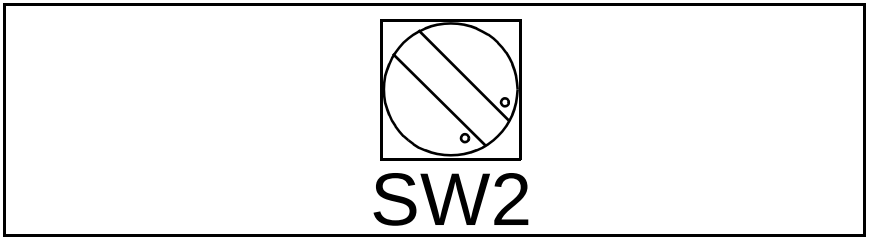


Figure 2-8. Flash Chip Select Switch Set for Normal Operation

To disconnect the Am186ES microcontroller from the on-board Flash memory, set the Flash chip select switch as shown in Figure 2-9. This allows external equipment to access the Flash memory via the $\overline{\text{FLASHCS}}$ signal on the expansion connector. The $\overline{\text{FLASHCS}}$ signal is connected to the Flash memory $\overline{\text{CE}}$ signal. This also allows the Am186ES microcontroller to execute code from an external device with the on-board Flash memory disabled.

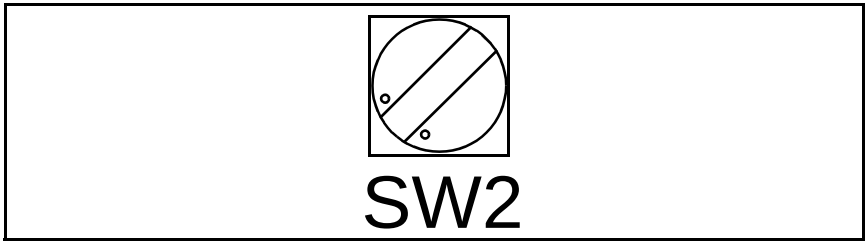


Figure 2-9. Flash Chip Select Switch Set to Disconnect the Am186 Microcontroller from On-Board Flash Memory

LED Indicators

The Net186 demonstration board uses on-board LED indicators to show activity on a subset of the programmable I/O (PIO) signals from the Am186ES microcontroller and on the PCnet-ISA II Ethernet controller. The following sections describe these LED indicators.

PIO Activity LED Indicators

Eight LEDs (CR5–CR12) are used to indicate activity on a subset of the Am186 microcontroller's PIO signals. Table 2-5 shows which PIO signal is represented by each LED.

Table 2-5. PIO LED Indicator Interface

LED	PIO Pin Name	PIO Register Bit Number
CR5	TMROUT0	15
CR6	SRDY	14
CR7	$\overline{\text{DEN}}$	5
CR8	DT/$\overline{\text{R}}$	4
CR9	$\overline{\text{PCS}}$5	3
CR10	$\overline{\text{PCS}}$6	2
CR11	TMROUT1	1
CR12	TMRIN1	0

1.0

PCnet-ISA II Ethernet Controller Status LEDs

Four LEDs (CR1–CR4) are used to indicate the status of the PCnet-ISA II Ethernet controller interface. These LEDS are connected in reverse order to LED outputs 0–3 of the Ethernet controller.

The function of these LEDs can be controlled by the configuration of the ISA bus configuration registers on the PCnet-ISA II Ethernet controller. The configuration registers can be configured both by EEPROM and software. The Net186 demonstration board EEPROM and software leave the LEDs in their default configuration (CR1/LED3 is inverted from its default function by the EEPROM). XXX shows the LEDs, their corresponding Ethernet controller signal, and the function of each one.

Table 2-6. PCnet-ISA II Ethernet Controller LED Indicator Interface

LED	Ethernet Controller Signal Name	LED Function
CR4	LED0	10BASE-T link status. When on, this LED indicates a good 10BASE-T connection.
CR3	LED1	Indicates receive activity from the network.
CR2	LED2	Indicates transmit activity from the Ethernet controller.
CR1	LED3	Indicates incorrect receive polarity on the 10BASE-T connection.

For more information about the use of these LEDs, see the *Am79C961A PCnet™-ISA II Jumperless, Full Duplex Single-Chip Ethernet Controller for ISA Data Sheet*, order #19364.

Power Supply

When used as a stand-alone board, the Net186 demonstration board requires an input power supply of 5.0 V DC, ±5%, 250 mA.

When adding components to the Net186 demonstration board via the Am186 expansion interface, additional power may be necessary.

The power supply connector is a 5.5-mm barrel connector where the center post is V_{CC} and the outer ring is GND, as shown in Figure 2-10.

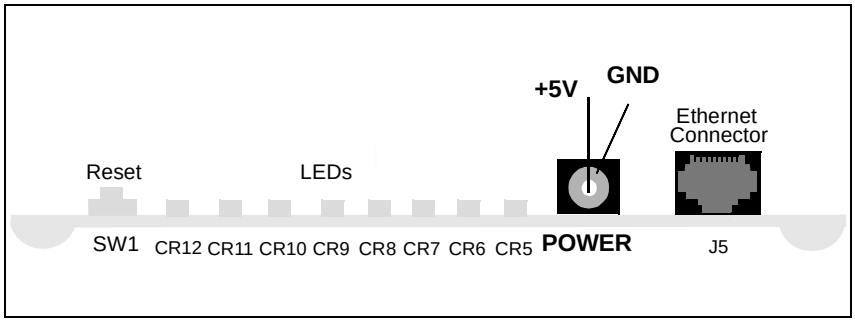


Figure 2-10. Power Supply Polarity

CAUTION: Use the 5-V universal power supply included with the kit. Using a 9-V supply will permanently damage the board.

1.0

PAL Equations

The glue logic required for the Am186ES microcontroller-to-PCnet-ISA II Ethernet controller interface is minimal. A total of eight inputs, six outputs, and a CLK are required to complete the design and will actually fit in a smaller PAL16V8 device. Refer to Appendix D, “PAL Source File Listing” for the PAL22V10 source file in PALASM format. Note that the input signal $\overline{\text{MCS}}$0 is not required and is connected only to allow you to "memory map" the PCnet-ISA II Ethernet controller (versus mapping in I/O space).

The most complicated task is converting the signals available on the Am186ES microcontroller to the ISA bus read/write logic required by the PCnet-ISA II Ethernet controller. To allow 8-bit I/O cycles, $\overline{\text{SBHE}}$ (System Byte High Enable) must be driven appropriately to the PCnet-ISA II Ethernet controller. In the case of read operations, $\overline{\text{SBHE}}$ is asserted for both 8-bit and 16-bit operations. In the case of write operations, $\overline{\text{SBHE}}$ is asserted only when the upper byte needs to be written. This is the case for all word writes to even addresses and byte writes to odd addresses. The ISA bus specification requires a 5-ns hold time from $\overline{\text{IOR}}$/$\overline{\text{IOW}}$ inactive to $\overline{\text{SBHE}}$ inactive. A digital one-shot implemented in the PAL device extends $\overline{\text{SBHE}}$ from the Am186ES microcontroller to satisfy this requirement.

Unfortunately, the Am186ES microcontroller’s $\overline{\text{LCS}}$ signal, which would typically drive the SRAM memory directly, does not three-state during a bus hold. Because the PCnet-ISA II Ethernet controller must master the bus and DMA to and from memory, RCS is created in the PAL device to drive the chip select on the RAM. This signal is the logical OR of $\overline{\text{LCS}}$ and $\overline{\text{MASTER}}$. When the PCnet-ISA II Ethernet controller controls the bus, $\overline{\text{MASTER}}$ will assert, and the RAM chip select will be continuously active.

The SRAM selected (Toshiba 256K x 16 or equivalent) requires a minimum write pulse of 50 ns. The Am186ES microcontroller’s $\overline{\text{WR}}$ signal *cannot* be used directly because it has a minimum pulse width of only 40 ns at 40 MHz. The PAL device is used to combine $\overline{\text{WHB}}$ and $\overline{\text{WLB}}$, generating the proper timing.

Net186 Initialization Overview

When the Net186 demonstration board comes out of reset, both the Am186ES microcontroller and the PCnet-ISA II Ethernet controller will go through an initialization process. The Am186ES microcontroller will execute a program resident in the Flash memory that initializes the various internal registers (Peripheral Control Block) required to talk to peripherals. Specifically, the $\overline{\text{UCS}}$, $\overline{\text{LCS}}$, $\overline{\text{PCS}}$2, and $\overline{\text{PCS}}$3 chip selects need to be programmed for the Flash memory, SRAM, and PCnet-ISA II Ethernet controller, respectively. The following portion of code demonstrates how the chip selects are set-up for the Net186 demonstration board:

```
/* set /UCS to 512 KBytes beginning at 80000h, no wait states */
OutPortWord(UMCS, 0x803C) ;

/* set /LCS to 512 KBytes ending at 7FFFFh, no wait states */
OutPortWord(LMCS, 0x7F3C) ;

/* set PIOs 18 and 19 (PCS2,3)to normal mode as Peripheral Chip Selects
*/
OutPortWord(PIO1_MODE,InPortWord(PIO1_MODE) & ~0x000C) ;

/* set PIOs 18 and 19 (PCS2,3)to normal mode as Peripheral Chip Selects
*/
OutPortWord(PIO1_DIR,InPortWord(PIO1_DIR) & ~0x000C) ;

/* Assert PCS in I/O space (vs Memory Mapped) */
OutPortWord(MPCS, 0x81B8) ;

/* Set base for PCS to 0x0000, external ready required, 3 wait states
minimum */
OutPortWord(PACS, 0x0073) ;
```

When this setup is complete, the Am186ES microcontroller can communicate with the PCnet-ISA II Ethernet controller register set at any I/O address between 200h and 3FFh. Note that where the PCnet-ISA II Ethernet controller is mapped depends upon how it has been initialized.

On the Net186 demonstration board, the PCnet-ISA II Ethernet controller initialization is executed by the preprogrammed EEPROM. Immediately following reset, the contents of the EEPROM are automatically read into the PCnet-ISA II Ethernet controller register set allowing the Am186ES microcontroller to communicate to it. Refer to the *Am79C961A PCnet™-ISA II Jumperless, Full Duplex Single-Chip Ethernet Controller for ISA Data Sheet*, order #19364, for details regarding the contents and arrangement of the EEPROM.

PCnet-ISA II Ethernet Controller Legacy Mode

The Am79C961A PCnet-ISA II Ethernet controller is designed so that it always responds to Plug-and-Play (PnP) configuration software. Most embedded networking applications, including the Net186 demonstration board, do not require PnP. In fact, PnP can complicate software initialization. Fortunately, the PCnet-ISA II Ethernet controller features a Legacy mode that allows you to hard code the resources (I/O port, DMA, IRQ, etc.) in the EEPROM. This allows the Am186ES microcontroller to communicate to the PCnet-ISA II Ethernet controller immediately following reset.

In Legacy mode, the Ethernet controller ignores the PnP software's special initiation key sequence (6A) and is visible in the I/O space. Only special setup programs are able to reconfigure the Ethernet controller while in Legacy mode. If the EEPROM is missing, empty, or corrupted, the Ethernet controller will still recognize AMD's special initiation key sequence (6B).

To enable Legacy mode, write a 1 to the LGCY_EN bit (bit 6) of PnP Register 0xF0. The preferred method for this is to set the LGCY_EN bit in the Vendor Byte (PnP 0xF0) field of the EEPROM located in word offset 0x1A.

When written with a 1, the Ethernet controller will not respond to the PnP initiation sequence (6A), but will respond to the AMD key sequence (6B); therefore, the Ethernet controller cannot be reconfigured as PnP software.

When set to 0, the Ethernet controller will respond to the 6A key sequence if the EEPROM read was successful; otherwise, it will respond to the 6B key sequence.

See the *Am79C961A PCnet™-ISA II Jumperless, Full Duplex Single-Chip Ethernet Controller for ISA Data Sheet*, order #19364, for a detailed description of Legacy mode and the LGCY_EN bit.

Things to Remember

- Under normal operation, the PCnet-ISA II Ethernet controller has to reside somewhere between address 200h and 3FFh for it to respond to I/O cycles. The PCnet-ISA II Ethernet controller only decodes 10 address bits for I/O cycles and 12 for PnP commands.
- The important PnP I/O ports are 279h, A79h, and 203h–3FFh. If the Net186 demonstration board is ever run without an EEPROM, these addresses must be available to bring the device out of PnP isolation.
- $\overline{\text{PCS}}2$ and $\overline{\text{PCS}}3$ assert at address 200h and 3FFh respectively. These pins are used as CTS and RTS for serial port 1, thus preventing the use of hardware flow control if the PCnet-ISA II Ethernet controller is mapped to I/O space. CTS and RTS are routed to the driver for serial port 1, but are disabled on the production Net186 demonstration boards by not stuffing the 0-ohm resistors at R3 and R4. If the PCnet-ISA II Ethernet controller is memory mapped, thus not requiring $\overline{\text{PCS}}2$ and $\overline{\text{PCS}}3$, then R3 and R4 could be stuffed.
- With 512 Kbyte of Flash memory and 512 Kbyte of SRAM, the entire memory addressing capability of the Net186 demonstration board is utilized. The PCnet-ISA II Ethernet controller *must* be mapped to I/O space.

  The logical place to map the PCnet-ISA II Ethernet controller is in I/O space using $\overline{\text{PCS}}2$ and $\overline{\text{PCS}}3$. This provides the flexibility to map the device anywhere between 200h and 3FFh and to run without an EEPROM if desired. The tradeoff is giving up CTS and RTS flow control on the second serial port.

  If an application required less Flash memory or SRAM, then the PCnet-ISA II Ethernet controller could be memory mapped using one of the MCS signals. This would allow you to recover CTS and RTS. This application is provided for on the Net186 demonstration board by including $\overline{\text{MCS}}0$ as an input to the PAL device. By simply reprogramming the PAL device, the PCnet-ISA II Ethernet controller could be memory mapped.

1.0

Chapter 3

Product Support

This chapter provides information on:

Note that AMD does not support source code changes to the E86MON software or other demonstration software, and AMD does not support the running of the E86MON software on demonstration boards other than the AMD SD186/SD188 family and the Net186 demonstration boards. The E86MON software and other demonstration board source code is provided to customers "as is".

10

AMD Corporate Applications Technical Support Services

Technical support for the E86 family of microcontrollers, corresponding support products, and the PCnet family of products is available via e-mail, online (BBS and WWW), and through telephone or fax.

E-Mail Support

Please include your name, company, telephone and fax numbers, AMD product requiring support, and question or problem in all e-mail correspondence.

In the USA and Canada, send mail to:

`lpd.support@amd.com`

In Europe and the UK, send mail to:

`euro.tech@amd.com`

1.0

Online Support

AMD offers technical support on our WWW site and through our bulletin board services. See “Product Support” on page 3-4 for more on what our WWW and FTP sites have to offer.

WWW Technical Support

Go to AMD’s home page at **http://www.amd.com** and click on “Service” for the latest AMD technical support phone numbers, software, and Frequently Asked Questions.

Bulletin Board Support

Country	Number
USA and Canada	(408) 749-4659
UK and Europe	44-(0) 1276-803-211

Telephone and Fax Support

Telephone assistance is available in the U.S. from 8:00 A.M. to 5:00 P.M. Pacific time, Monday through Friday (except major holidays). In Europe, assistance is available during U.K. business hours. Contact the hotlines at one of the following telephone or fax numbers.

Direct Dial Numbers

Country	Number
USA and Canada	Tel.: (408) 749-5703 Fax: (408) 749-4753
Japan	Tel.: (03) 3346-7550 Fax: (03) 3346-9828
Far East Asia	Fax: (852) 2956-0599
Germany	Tel.: 089 450 53199
UK and Europe	Tel.: 44-(0) 1276-803-299 Fax: 44-(0) 1276-803-298

Toll-Free Numbers

Country	Number
USA and Canada	(800) 222-9323
France	0590-8621
Italy	1678-77224
Japan	0031-11-1163

Product Support

AMD's WWW and FTP sites are described below. Questions, requests, and input concerning these sites can be sent via e-mail to **webmaster@amd.com**.

WWW Site

A subset of the AMD WWW pages, the embedded processor and networking product pages are frequently updated and include general product information, technical documentation, and support and tool information. To access these pages, go to the AMD home page at **http://www.amd.com** and click on "Embedded Processors" or "Networking". You can also access the pages directly at **http://www.amd-embedded.com** or **www.amd.com/products/npd/npd.html**.

The "Embedded Processors" and "Networking" home pages are divided into four sections:

- "What's New" announces new E86 or PCnet family products, and highlights new applications using our products.
- "Product Overviews" briefly describes all the products in the E86 and PCnet families, and describes how these parts are ideal in specific focus markets.
- "Support and Tools" provides information about the tools that support our products, and offers online benchmarking tools.
- "Technical Documentation" provides the *Available Literature List* of datasheets, application notes, user's manuals, and promotional literature, and describes how to order these documents. Many are also available online in PDF form. (To access the Literature Ordering Center via telephone, call one of the numbers listed on the back cover of your manual.)

The "Embedded Processors" page also includes a link under "Support and Tools" called "Demo Board Updates" which provides access to the AMD FTP site where the latest E86MON software releases are available.

1.0

FTP Site

In addition to the documentation on our WWW pages, AMD provides software through an anonymous FTP site. To download the software, ftp to **ftp.amd.com** and log on as "anonymous" using your e-mail address as a password. Or via your web browser, go to **ftp://ftp.amd.com**. Software relating to the embedded processor and networking products can be found in the **/pub/epd/e86/** or **/pub/npd/software/**directories.

Third-Party Development Support Products

The FusionE86SM Program of Partnerships for Application Solutions provides the customer with an array of products designed to meet critical time-to-market needs. Products and solutions available from the AMD FusionE86 partners include emulators, hardware and software debuggers, board-level products, and software development tools, among others. The *FusionE86SM Catalog*, order #19255, and the *FusionE86SM CD*, order #21058 describe these solutions.

In addition, mature development tools and applications for the x86 platform are widely available in the general marketplace.

Appendix A

Schematics and Board Bill of Materials

This appendix contains schematics for the Net186 demonstration board components (see page A-2) and the bill of materials (see page A-12).

1.0

Schematics

The Net186 demonstration board schematics have been blocked out to isolate functionality of the design onto separate pages as follows:

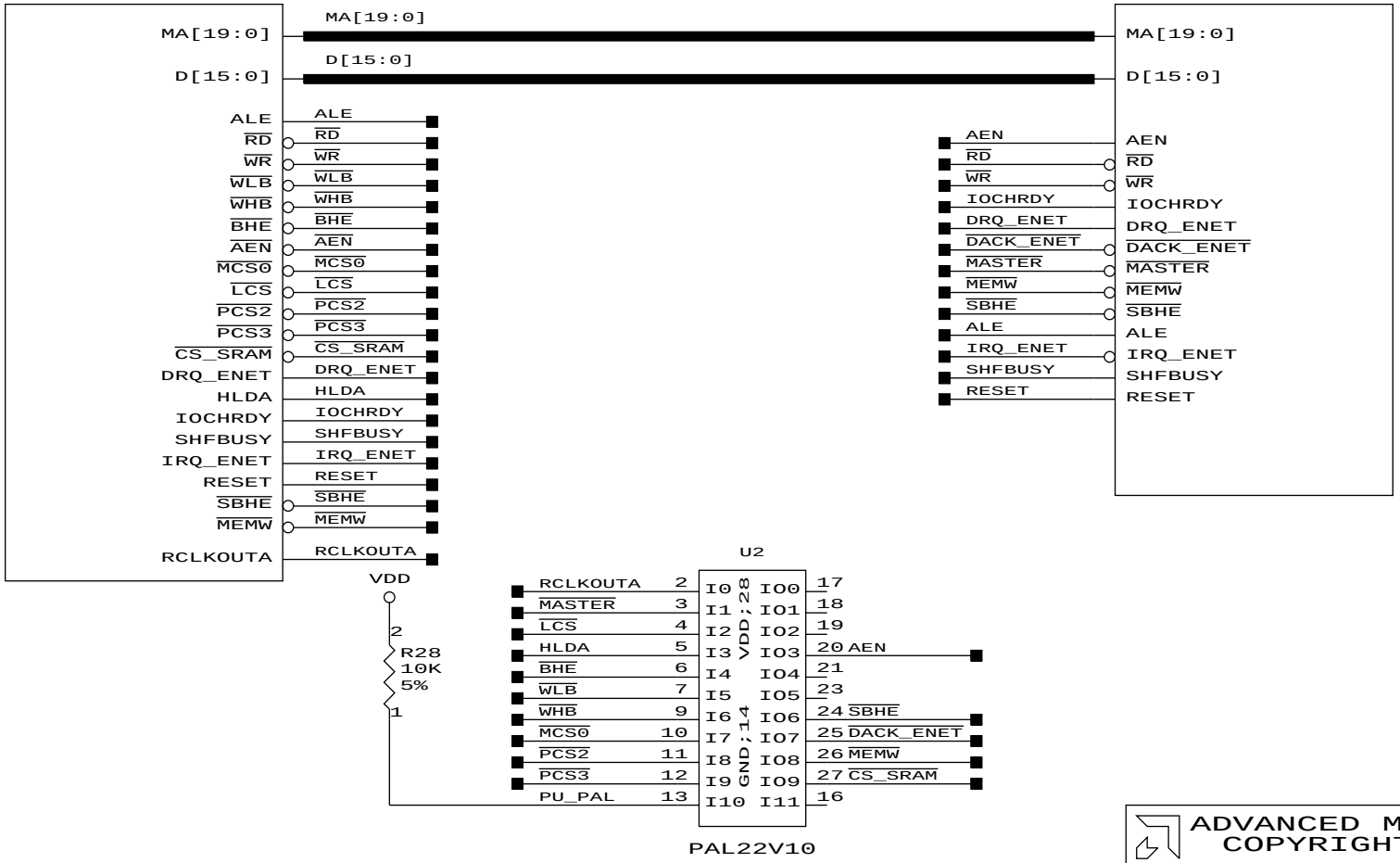
- Page A-3 contains the top level interconnect signals and the PAL22V10.
- Page A-4 contains the PCnet-ISA II Ethernet controller.
- Page A-5 contains the 10BASE-T interface and EEPROM.
- Page A-6 contains various signal pull-ups and pull-downs, including the reset board ID.
- Page A-7 contains the Am186ES microcontroller.
- Page A-8 contains the RS-232 interfaces.
- Page A-9 contains the processor LEDS (CR5-CR12), the SRAM, and the Flash memory.
- Page A-10 contains the Am186 expansion interface.
- Page A-11 contains the reset circuit and switch and the power supply decoupling caps.

These schematics and design are subject to change.

1.0

CPU/Memory Subsection

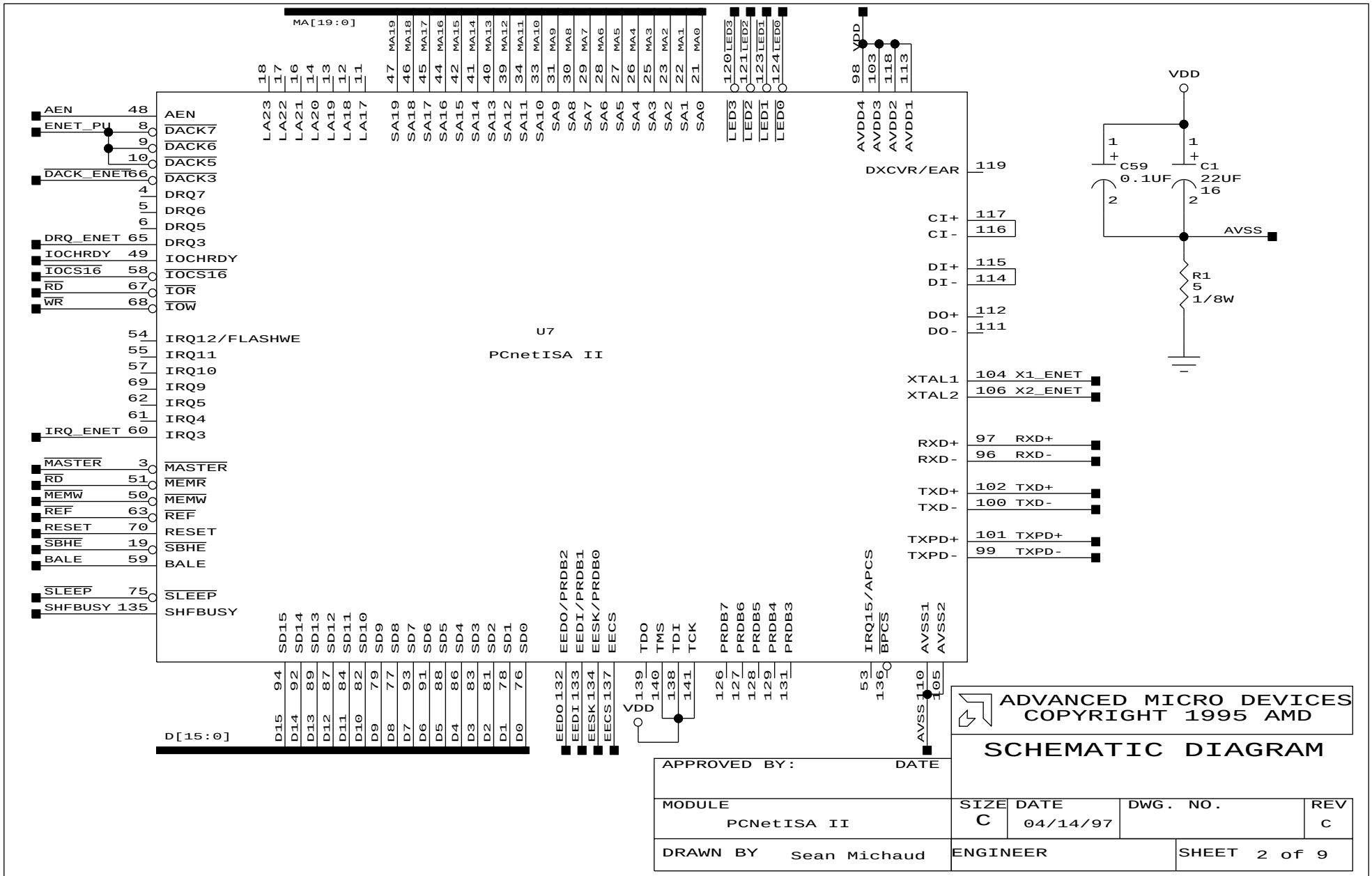
Ethernet Subsection



SCHEMATIC DIAGRAM

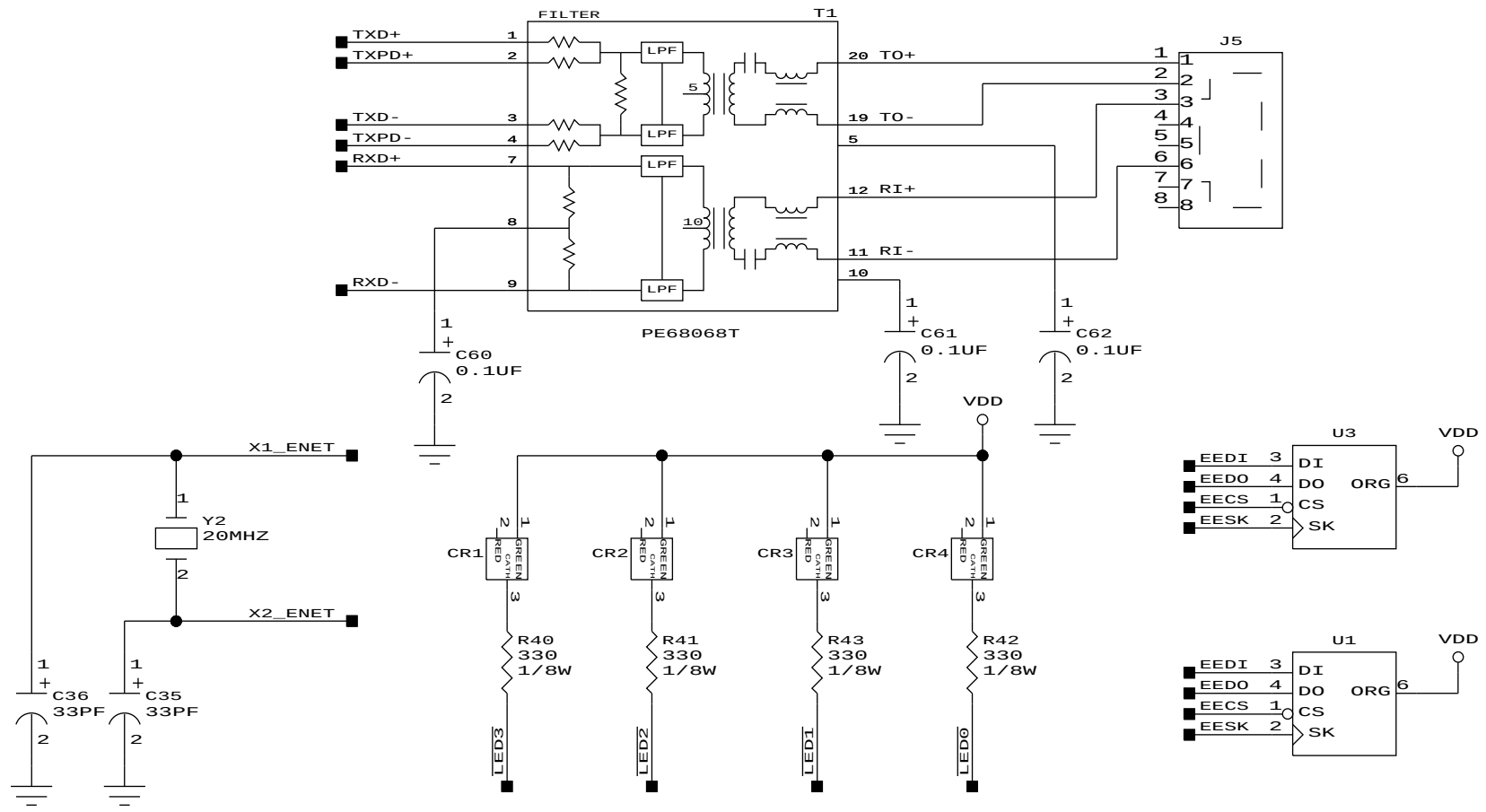
APPROVED BY: DATE				
MODULE Net186 Top Level	SIZE C	DATE 04/14/97	DWG. NO.	REV C
DRAWN BY Sean Michaud	ENGINEER Mark Bowers, Rick Purvis		SHEET 1 OF 9	

$1I173



SCHEMATIC DIAGRAM

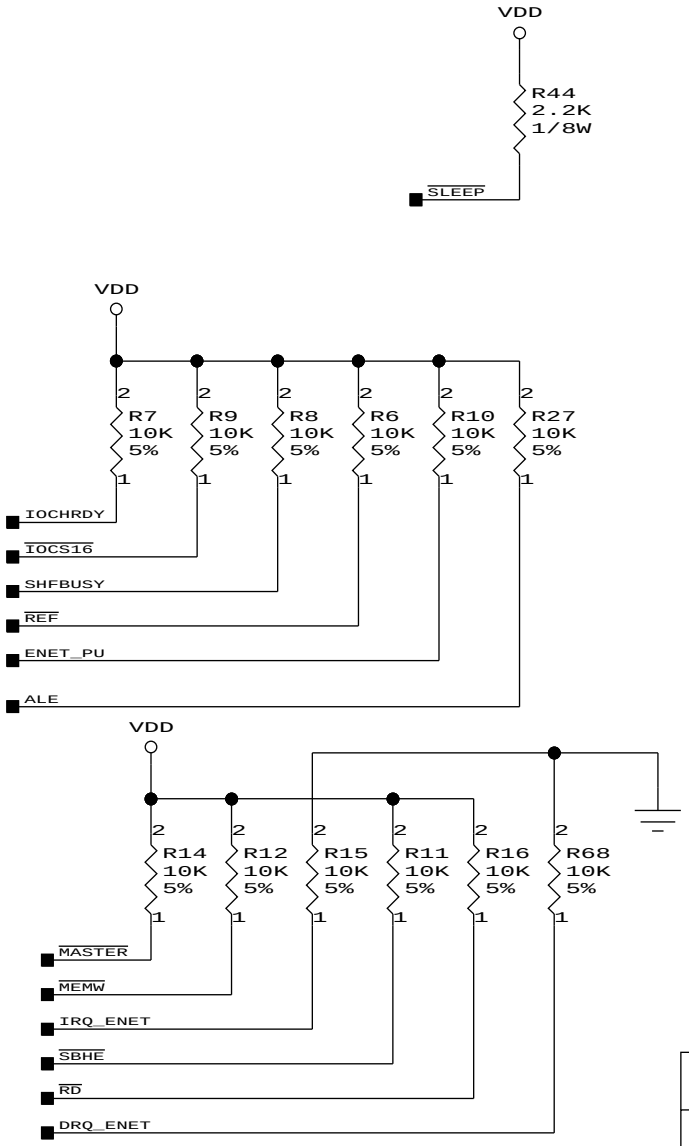
APPROVED BY:	DATE		
MODULE: PCNetISA II	SIZE: C	DATE: 04/14/97	DWG. NO. / REV: C
DRAWN BY: Sean Michaud	ENGINEER	SHEET 2 of 9	

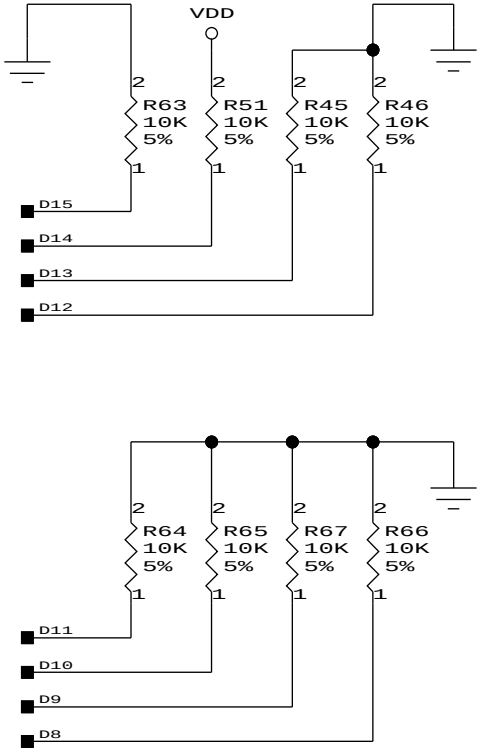


ADVANCED MICRO DEVICES

SCHEMATIC DIAGRAM

APPROVED BY: DATE				
MODULE: Ethernet Support	SIZE: C	DATE: 04/14/97	DWG. NO.	REV: C
DRAWN BY: Sean Michaud	ENGINEER		SHEET: 3 OF 9	

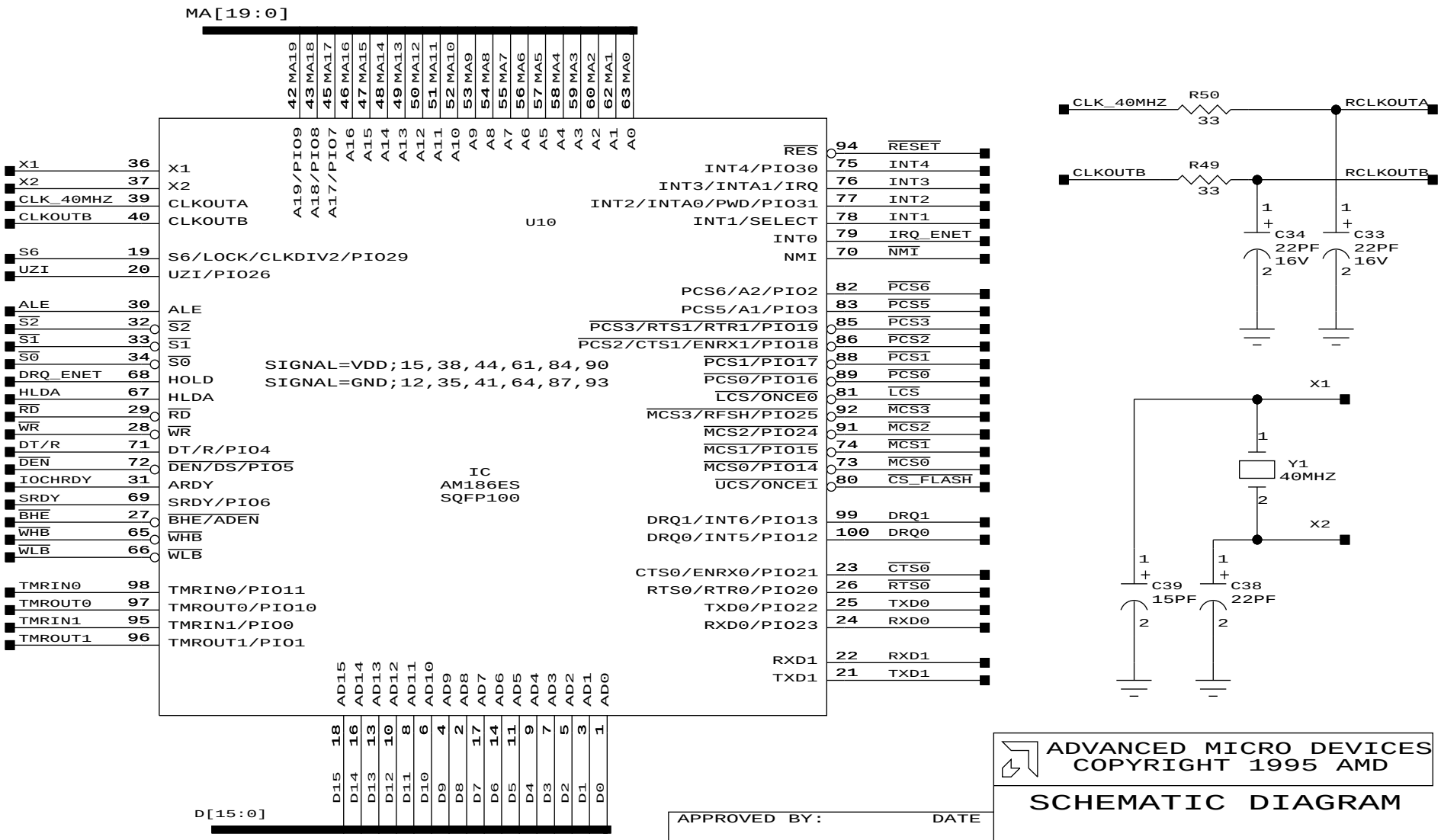




ADVANCED MICRO DEVICES

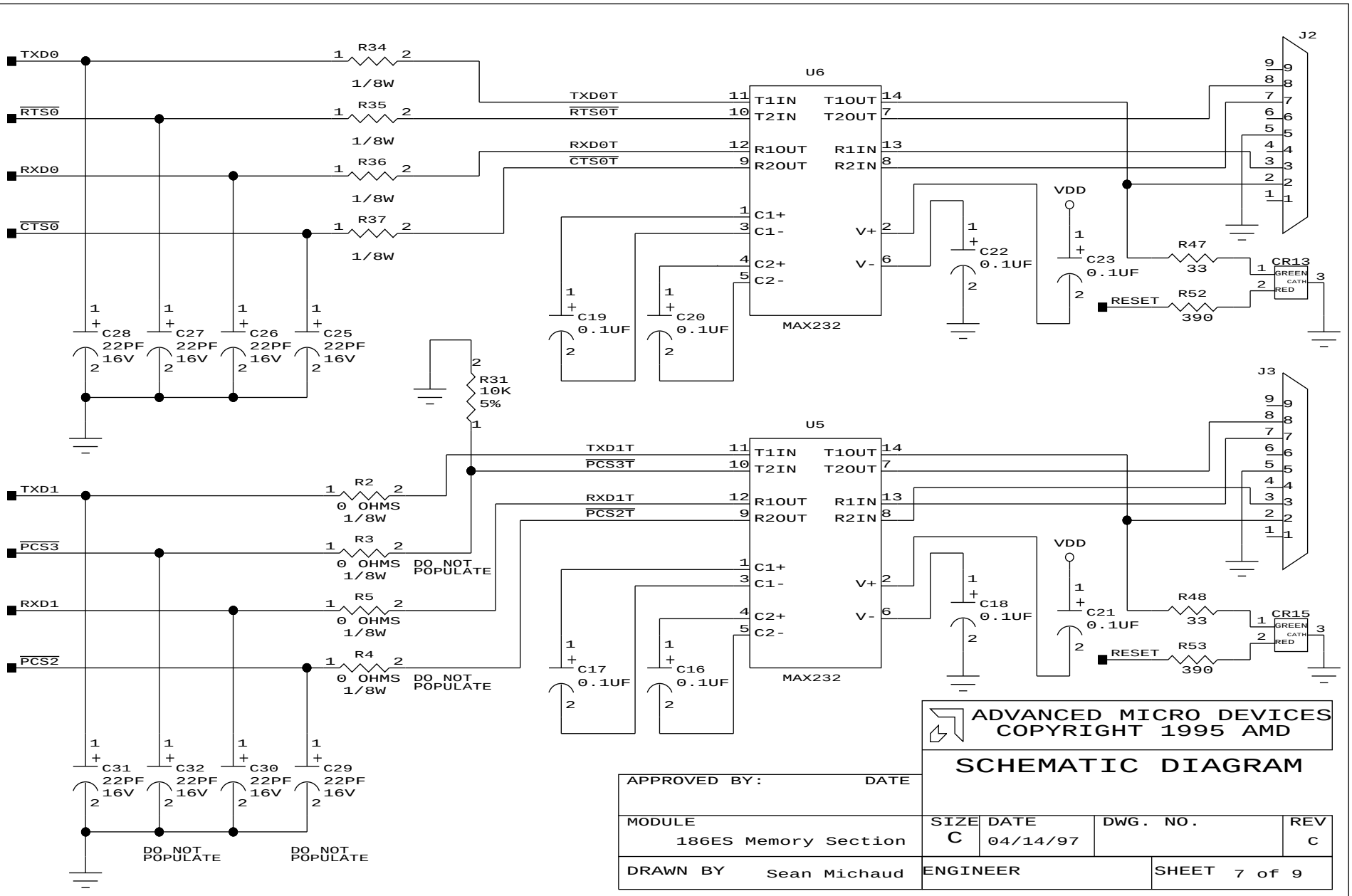
SCHEMATIC DIAGRAM

APPROVED BY:	DATE			
MODULE: Ethernet Pullup Resistors	SIZE: C	DATE: 04/14/97	DWG. NO.	REV: C
DRAWN BY: Sean Michaud	ENGINEER		SHEET: 4 OF 9	



SCHEMATIC DIAGRAM

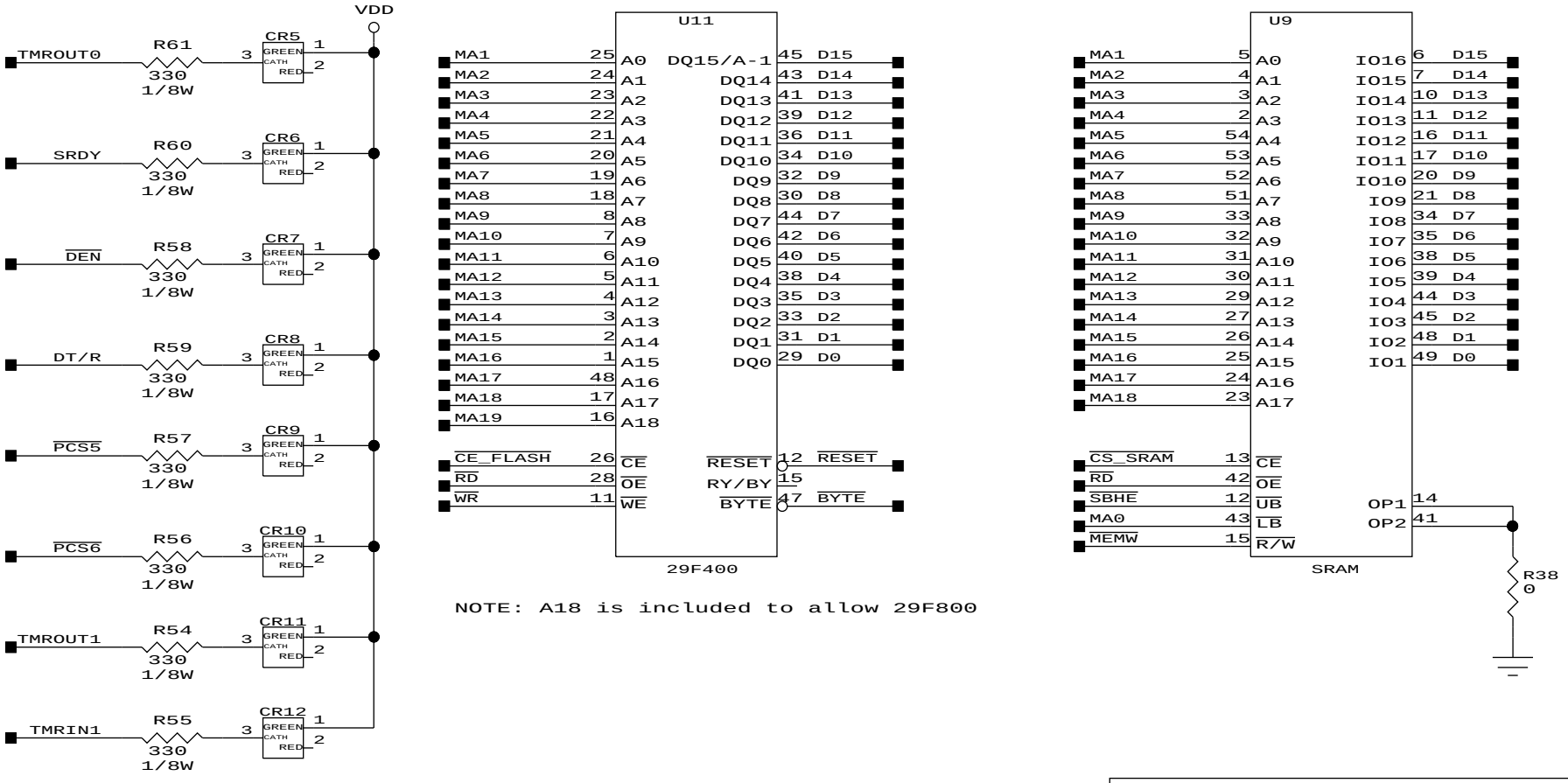
APPROVED BY: DATE			
MODULE 186ES CPU Subsystem	SIZE C	DATE 04/14/97	DWG. NO. / REV C
DRAWN BY Sean Michaud	ENGINEER	SHEET 5 of 9	



ADVANCED MICRO DEVICES
COPYRIGHT 1995 AMD

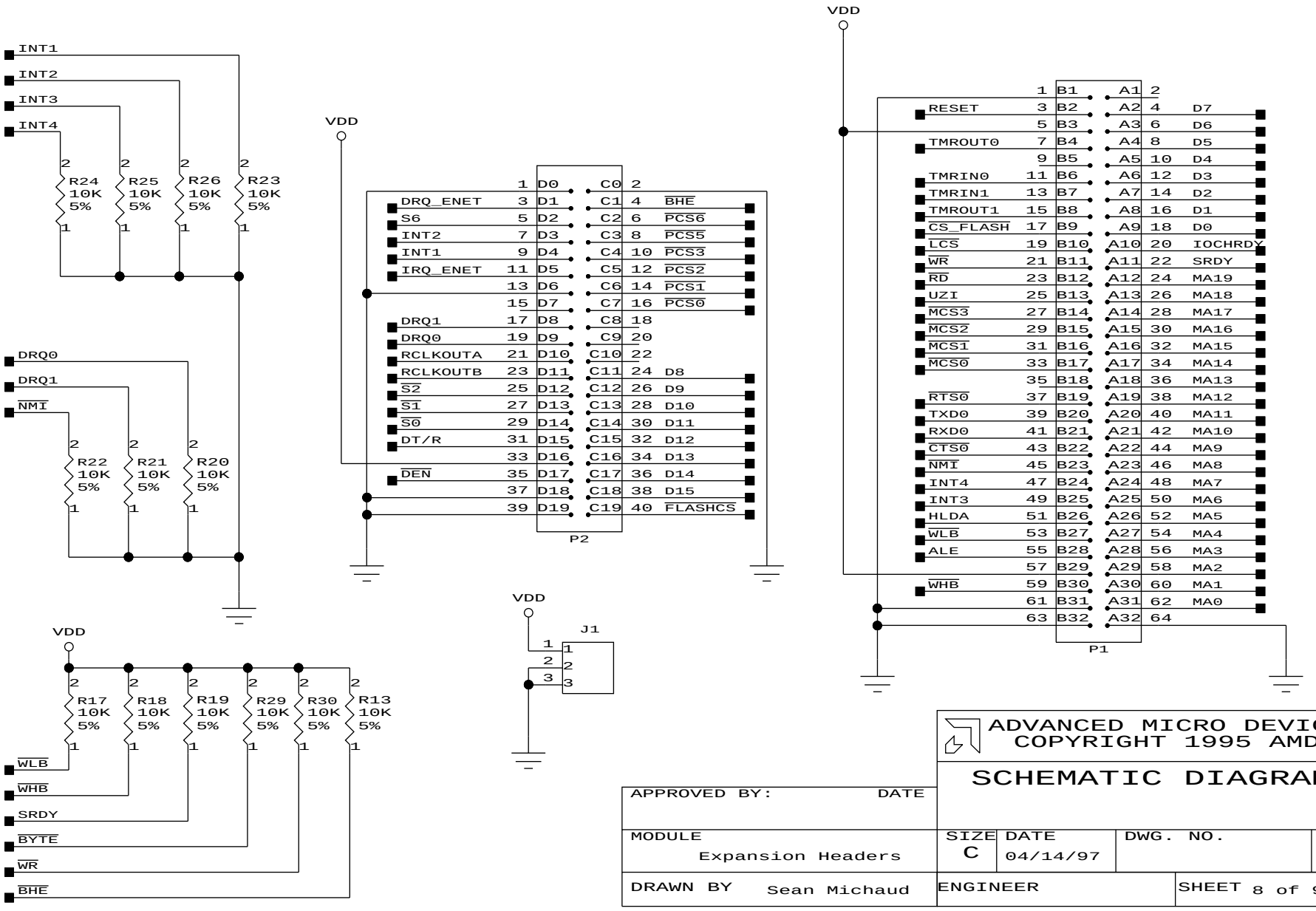
SCHEMATIC DIAGRAM

APPROVED BY: DATE				
MODULE 186ES Memory Section	SIZE C	DATE 04/14/97	DWG. NO.	REV C
DRAWN BY Sean Michaud	ENGINEER		SHEET 7 of 9	



SCHEMATIC DIAGRAM

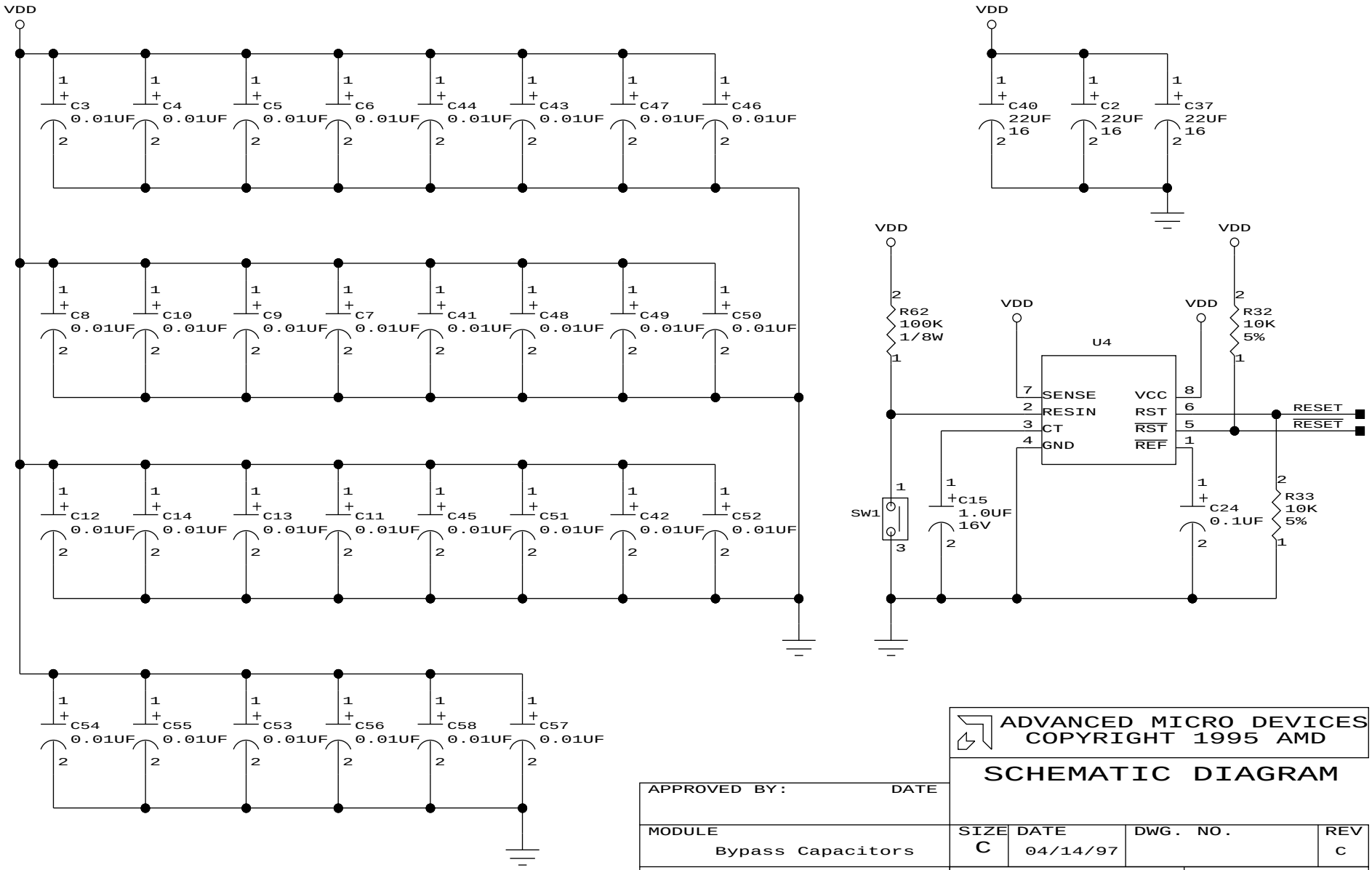
APPROVED BY: DATE				
MODULE 186ES Memory Section	SIZE C	DATE 04/14/97	DWG. NO.	REV C
DRAWN BY Sean Michaud	ENGINEER		SHEET 6 of 9	



ADVANCED MICRO DEVICES

SCHEMATIC DIAGRAM

APPROVED BY: DATE			
MODULE: Expansion Headers	SIZE: C	DATE: 04/14/97	DWG. NO. / REV: C
DRAWN BY: Sean Michaud	ENGINEER	SHEET: 8 of 9	



ADVANCED MICRO DEVICES

SCHEMATIC DIAGRAM

APPROVED BY: DATE				
MODULE Bypass Capacitors	SIZE C	DATE 04/14/97	DWG. NO.	REV C
DRAWN BY Sean Michaud	ENGINEER		SHEET 9 of 9	

Board Bill of Materials (BOM)

Table A-1. Net186 Demonstration Board BOM

Qty	Ref	Description	MFG1	Part No 1
4	C1, C2, C37, C40	22 MFd, SMT, C case, 20 V	Any	
30	C3–C14, C41–C58	0.01 MFd, SMT, 16 V	Any	
1	C15	1.0 MFd, SMT, 16 V	Any	
13	C16–C24, C59–C62	0.1 MFd, SMT, 20 V	Any	
10	C25–C34	22 PFd, SMT, 16 V	Any	
2	C35, C361	33 PFd, SMT, 16 V	Any	
2	C38, C39	15 PFd, SMT, 16 V	Any	
14	CR1–CR14	LED, 3-pin, SMT	LUMEX	SSL-LX15IGC-RP-TR
1	J1	Power jack, 5.5 mm, RA	Switchcraft	RAPC-712
2	J2, J3	DB9 right-angle connector, front metal-shell	AMP	787844-1
1	J5	RJ-45 connector		
1	P1	Am186 Conn, 64-pin header	Any	
1	P2	Am186 Conn, 40-pin header	Any	
1	R1	5 Ω, 5%	Any	
4	R2–R5	0 Ω, 5%	Any	
28	R6–R33, R45, R46, R51, R63–R67	10 kΩ, 5%	Any	
5	R34–R38	0 Ω, 5%	Any	
4	R40–R43	330 Ω, 5%	Any	
1	R44	2.2 kΩ, 5%	Any	
4	R47–R50	33 Ω, 5%	Any	
2	R52, R53	390 Ω, 5%	Any	
8	R54–R61	330 Ω, 5%	Any	
1	R62	100 kΩ, 5%	Any	
1	SW1	Reset switch, (SMT-J lead)	C&K	KT11P3JM
1	T1	10BASE-T filter and transformer	Pulse Engineering	PE68068T

1.0

Qty	Ref	Description	MFG1	Part No 1
1	U1	128 x 16 EEPROM	National	NM93C56N
1	U2	22V10 PAL	AMD	PALCE22V10H-7JC/5
1	U3	128 x 16 EEPROM	National	NM93C56N
1	U4	Reset controller, SMT	TI	TL7705ACD
2	U5, U6	RS-232 driver, narrow SMT	Maxim	MAX232ACSE
1	U7	PCnet-ISA II Ethernet Controller	AMD	Am79C961AVC\W
1	U9	256K x 16 SRAM (SMT)	Toshiba	TC514161
1	U10	Am186ES microcontroller	AMD	Am186ES-40VC\W
1	U11	512K x 16 Flash memory (SMT)	AMD	Am29F400AT-70EC
1	Y1	40.0-MHz quartz crystal, 16 pF	Epson	MA-306
1	Y2	20.0-MHz quartz crystal, 16 pF	Epson	MA-306

1.0

Appendix B

PCnet Family History

The following is a brief discussion/history of the LANCE (Local Area Network Controller for Ethernet) and the subsequent family of integrated single-chip Ethernet controllers that followed based on the LANCE/CLANCE (CMOS Local Area Network Controller for Ethernet) architecture called the PCnet Family.

The PCnet Family of single-chip Ethernet controllers is based on the original Am7990 LANCE bus mastering architecture. The original LANCE controller was introduced in 1985. Its architecture is the basis for AMD's PCnet Family of highly integrated single-chip Ethernet controllers. The one Ethernet controller that is the exception is the Am79C940 MACE (Media Access Controller for Ethernet). The original LANCE, the PCnet family, and MACE are described below.

For more information about members of AMD's family of Ethernet controllers, refer to the AMD Networking Products Guide CD, order #21244, included in your kit.

10

LANCE / CLANCE

The LANCE (Am7990) was a standalone Media Access Controller (MAC). It had no PHY layer logic. This means it had no integrated Manchester Encoder/Decoder (ENDEC) nor did it have an integrated 10BASE-T transceiver (TMAU). In 1992, AMD refabricated the NMOS LANCE to produce a CMOS LANCE called the C-LANCE (Am79C90). Though the internal architecture of the CLANCE is a little different than the internal architecture of the LANCE, to the user/programmer, the architecture of the CLANCE is exactly the same as the LANCE. This means that whatever software suites were developed for the LANCE can also be used with the Am79C90 CLANCE without changing any of the original Am7990 LANCE software code. Thus, whenever the name CLANCE is used, it is really the original LANCE Ethernet media access controller fabricated using CMOS technology. The CLANCE is pin compatible to the LANCE and like the LANCE, CLANCE is also a bus-mastering Ethernet controller.

AMD still offers the CLANCE (Am79C90) Ethernet controller. The original LANCE architecture is as popular today as it was when it was first introduced over a decade ago.

PCnet Family of Ethernet Controllers

During the last half of the 1980s, AMD embarked on a program to produce highly integrated single-chip Ethernet controllers based exclusively on the original LANCE/CLANCE architecture; that is, an Ethernet controller chip with integrated MAC/ENDEC/TMAU (Media Access Control/Encoder-Decoder/10BASE-T Media Access Unit). In addition, it was decided to integrate the bus interface logic on board the chip. The idea of maintaining architecture compatibility with the original LANCE/CLANCE Ethernet controllers was done to ensure software compatibility. AMD realized that there was a very large base of LANCE/CLANCE users who did not want to rewrite their LANCE-based software code. The inclusion of a bus interface logic on board the chip was done to provide easy, painless direct connections, without additional hardware glue logic, to most popular standard buses. These buses included ISA, EISA, VL and PCI buses. AMD now has a series of single-chip Ethernet controllers, collectively called the PCnet family of Ethernet controllers.

There is a series of four basic sets of PCnet family Ethernet controllers. You can tell which bus environment a particular PCnet Ethernet controller was designed for by its name, as described in the following sections.

1.0

PCnet-ISA Series (16-Bit) for ISA/EISA-Based Environments

PCnet-ISA (Am79C960) was AMD's first PCnet Ethernet controller introduced in 1992. This particular product won the 1992 Award from *PC Magazine* and was described as "The year's most technically innovative connectivity product brings networking to the masses."

PCnet-ISA+ (Am79C961) was the next PCnet-ISA member introduced in 1993, and was not pin compatible to the PCnet-ISA. Compared to the PCnet-ISA, the PCnet-ISA+ was jumperless and was compliant to Microsoft's Plug and Play specifications for ISA.

PCnet-ISA II (Am79C961A) is the last member of the PCnet-ISA series introduced in 1994. This device is pin compatible to the PCnet-ISA+. Like the PCnet-ISA+, this device is jumperless and is compliant to Microsoft's Plug and Play specifications for ISA. Additionally, PCnet-ISA II is full-duplex capable. This means it can transmit and receive data at the same time over its 10BASE-T twisted pair port. Full-duplex gives PCnet-ISA II the capability of bidirectional data transfer rate of up to 20 Mbit/s combined. PCnet-ISA II also has a feature unique to AMD's Ethernet controllers, Magic Packet™ Technology. Created in collaboration with Hewlett-Packard, Magic Packet Technology allows you to remotely wake up a sleeping PC or any piece of equipment on a network.

Today, only the PCnet-ISA II (Am79C961A) is available and is featured on the Net186 standalone demonstration board. The PCnet-ISA (Am79C960) and the PCnet-ISA+ (Am79C961) are no longer offered.

PCnet-32 (32-Bit) for VL or General 32-Bit Local-Bus Based Environments

PCnet-32 (Am79C965) was introduced in 1993. While it appears to be made specifically for the VL bus, PCnet-32 is really a general 32-bit Ethernet controller. However, this device does not support full-duplex or Magic Packet Technology. It is well suited for general 32-bit embedded networking applications, and in particular, high-end routers. The PCnet-32 also interfaces very well to an 80486.

PCnet-PCI II (32-Bit) for PCI-Based Environments

PCnet-PCI II (Am79C970A) was introduced in 1995 and is a 32-bit Ethernet controller designed for use in PCI bus based environments. The PCnet-PCI II has a General Purpose Serial Interface (GPSI) enabling you to bypass the ENDEC & TMAU and connect directly to the MAC, External Address Detection Interface (EADI), bigger FIFOs, and full-duplex capability. PCnet-PCI was introduced in 1993. While AMD no longer offers the PCnet-PCI (Am79C970), software drivers written for the PCnet-PCI can also be used on the PCnet-PCI II. Like the PCnet-ISA II, PCnet-PCI II also has Magic Packet Technology to remotely wake up a sleeping PC or any piece of equipment on a network.

PCnet-FAST (32-Bit) for PCI-Based Environments

PCnet-FAST (Am79C971) was introduced in June 1996 and is AMD's latest PCnet family member introduced to address the growing need for 100 Mbit/s fast Ethernet capability. PCnet-FAST is capable of operating as a 10-Mbit/s Ethernet controller as well. Like PCnet-ISA II and PCnet-PCI II, PCnet-FAST is full-duplex capable and also has Magic Packet Technology.

Software Compatibility

The PCnet family of Ethernet controllers (PCnet-ISA II, PCnet-32, PCnet-PCI II and PCnet-FAST) is LANCE/CLANCE software compatible. This means you can use the original 16-bit LANCE/CLANCE software on the above members of the PCnet family of single-chip Ethernet controllers. It is this software compatibility that is the PCnet family's biggest value proposition. You have an upgrade path from the popular standalone LANCE/CLANCE MAC to a highly integrated, single-chip solution whose architecture is compatible to the LANCE/CLANCE.

1.0

MACE (16-Bit) for General 16-Bit Environments

AMD offers another single-chip Ethernet controller device that is not a member of the PCnet Family. This Ethernet controller is called the Media Access Controller for Ethernet (MACE Am79C940). The name implies a standalone MAC, which it is not. Think of the PCnet-ISA II Ethernet controller without the ISA bus interface logic, LANCE DMA controller, and buffer memory management unit: a 16-bit, general purpose, busless, single-chip Ethernet controller. Unlike the members of the PCnet family, which are bus-mastering devices, the MACE is a bus-slave Ethernet controller device.

Because the MACE does not have the same architecture as the LANCE/CLANCE, software originally developed for the LANCE/CLANCE will not run on the MACE.

Appendix C

References

This appendix contains a short list of reference material for those who would like to learn more about Ethernet or networking in general.

Books and Literature

TCP/IP Illustrated by W. Richard Stevens, Addison-Wesley Publishing (**http://www.aw.com**), ISBN 0-201-63346-9.

Internetworking by Mark A. Miller, M&T Books (1-800-533-4372), ISBN 1-55851-143-1. A good general overview, including wide area communications.

Embedded Networking Applications Design Guide Kit, AMD (1-800-222-9323), order #20397.

Periodicals

Looking in the classifieds of *Embedded Systems Programming* magazine (Miller Freeman, 1-800-829-5537, **http://embedded.com**), you will find numerous advertisements for protocol stacks, RTOSs, emulators, etc. This magazine also has a number of general embedded articles that are worthwhile reading.

Another noteworthy publication is *Electronic Engineering Times* (CMP Publications, Inc., **http://techweb.cmp.com/**). This is a weekly publication providing an abundance of information on the current state of the electronics industry, including occasional in-depth articles on networking and embedded applications as well as advertisements for all types of electronics services and products.

World Wide Web

The WWW is an excellent resource for finding out more about networking, especially embedded networking applications. Just searching on “TCP tutorial” will bring up plenty to keep you busy.

A few web sites produced by such searches are:

- **http://www-cne.gmu.edu/modules/network/index.html**
  This is a very informative “subway map” of a wide range of networking tutorials and resources, including a link to the original TCP/IP tutorial RFC (request for comment).
- **http://www.softaid.net/emulate/articles/article.html**
  Another informative source, this provides an extensive collection of articles written by Jack Ganssle of Softaid (one of our Fusion E86 partners). There are dozens of articles of relevance to the embedded developer.

Most networking companies also have web sites which offer a wide variety of useful information. Also, several newsgroups frequently discuss Ethernet issues, including embedded networking application issues. Try reading the **comp.arch.embedded** or **comp.protocols.tcp-ip** newsgroups.

1.0

Appendix D

PAL Source File Listing

This appendix contains the contents of the PAL source file included in your kit.

1.0

PAL Source File Contents

```
;PALASM Design Description
;------------------- Declaration Segment ------------
TITLE    186ES PCnet-ISA Glue Logic
PATTERN
AUTHOR   Mark Bowers, FAE Atlanta
COMPANY  Advanced Micro Devices, inc.
DATE     3/19/96

REVISION HISTRY
; 3/19/96       MB - CREATED
; 3/28/96       MB - add 85C30 support, use 186ES
; 4/8/96        MB - included digital one-shot for sbhe_
; 5/7/96        MB - fixed mistake w/ I/O slave reads (added PCS3,
;                    removed rd_)
; 5/7/96       MB - added support for PCnet byte writes (added ADX0)
; 5/7/96        MB - fixed mistake on sbhe tristate
; 7/25/96       MB - took out 85C30, converted from 16V8 to 22V10
; 11/7/96       MB - went back to I/O mapped
; 11/18/96    MB - changed to support 16 bit SRAM, swapped pins to match
;                    schematic
; 11/27/96      MB - swapped pins (again) to match schematic
; 1/24/97       MB - swapped pins (yet again) to match schematic

CHIP  es_pcnet  PALCE22V10

;Note DIP pinout for SSOP

;---------------------------------- PIN Declarations ---------------
;    DIP                 PLCC
PIN  1          clka    ;2 - clk from 186ES
PIN  2          master_ ;3 - from PCnet, indicates PCnet has bus
PIN  3       lcs_   ;4 - lower chip select from 186ES, addresses memory
PIN  4          hlda    ;5 - from 186ES, inverted to DACK* on PCnet
PIN  5          bhe_    ;6 - Byte High Enable from 186ES
PIN  6          wlb_    ;7 - Write Low Byte from 186ES
PIN  7          whb_    ;9 - Write High Byte from 186ES
PIN  8       mcs0_  ;10 - /MCS0, not used in this design, could be used
;                           to memory map PCnet-ISA
PIN  9          pcs2_   ;11 - peripheral CS from 186ES
PIN  10         pcs3_   ;12 - peripheral CS from 186ES

PIN  16         dbhe_   ;19 - Delayed bhe_, external N/C
PIN  17         aen_    ;20 - Address Enable to PCnet
PIN  20        sbhe_   ;24 - System Byte High Enable on PCnet and SRAM
PIN  21         hlda_   ;25 - connected to DACK* on PCnet
PIN  22      memw_  ;26 - Memory Write drives SRAM when ES has bus
PIN  23         rcs_    ;27 - ram chip select
```

1.0

```
;---------------------------------- Boolean Equation Segment ------
EQUATIONS

hlda_ = /hlda

/rcs_ = /lcs_ + /master_        ;lcs_ doesn't tristate on a bus hold

;PCS2 and PCS3 assert when the 186ES address is x2xxh and x3xx
respectively.
;This is the I/O space that we want to talk to the PCnet-ISA in.
/aen_ = /pcs2_ + /pcs3_

;The signal memw_ drives the R/W line on the SRAM.  When the 186ES has the
;bus, this is simply a logical AND of whb_ and wlb_.  The 186ES signal /WR
;can NOT be used because the pulse width is too short at 40MHz.
/memw_ = /whb_ + /wlb_
memw_.TRST = master_

;To support 8 bit I/O cycles to the PCnet-ISA, SBHE must be asserted
before
;IOR or IOW. To conform to the ISA spec, SBHE is extended so the trailing
;edge occurs after IOR or IOW goes high.  The signal sbhe is tristated
when
;PCnet has the bus.
dbhe_ := bhe_
/sbhe_ = /bhe_ + /dbhe_
sbhe_.TRST = master_
```

10

Appendix E

EEPROM Contents

This appendix contains the text file named NET186.DAT that shows the contents of the EEPROM used on the Net186 demonstration board. The utility EESETUP can be used to program the EEPROM based on the data in this file.

Note that the data in this file is byte reversed from what ends up in the PCnet registers. There are several differences from a typical PC application:

1. The I/O resources are hardcoded to:

   DMA = 3
   IRQ = 3
   I/O ADX = 200h

   The DMA and IRQ must be assigned as shown because that is how the PCnet-ISA II is connected to the Am186ES microcontroller. The I/O ADX could be changed (200h - 3FFh in increments of 20h), but the contents of the EEPROM have to match where the PCnet is mapped in the 186 software.

2. ISACSR0 and ISACSR1 are both set to 0002h. This sets the Master Mode Read/Write Active time to 100 ns, which maximizes DMA performance.
3. ISACSR2 is set to 0212h, which minimizes the Read/Write Inactive time for DMA transfers, and maximizes performance.

When the Am186ES microcontroller and PCnet-ISA II are "alive", software drivers can be loaded and network traffic can be processed. Please refer to “Software” on page x.

NET186.DAT File Listing

```
// File: net186.dat
// Revision: 1.0
// PCnet Family controller: Am79C961A PCnet-ISA II
//
// ANSI identifier string:  AMD PCnet-ISA II Ethernet Network Adapter
//
// Card state at power-on/Reset: active/visible in I/O space
// Plug and Play BIOS or Configuration Manager: not required to wake
card
// Plug and Play Boot Device: No
// Plug and Play Boot ROM: Not specified
/
********.*********.*********+******************.*********.********/
/*                          EEPROM Byte Map                          */
/*        PCnet-ISA+ Data Sheet - PID# 18183  Rev.B - Apr 1994       */
/*       PCnet-ISA II Data Sheet - PID# 19364A  Rev.A - OCT 1994      */
/
********.*********.**********+******************.*********.********
/
// 64 bytes of PCnet-ISA+ Configuration Information follows:

// The #ieee_addr keyword here stimulates a call to Address Manager
which
// dispenses the next available IEEE address from ADDR_MGR.DAT data
base.
//      OR
// The #force_addr keyword here gives you the ability to force a
specific
// IEEE address by providing your own six bytes of information.
//      OR
// The #reuse_ieee_addr keyword here causes EESETUP to (blindly) reuse
the
// IEEE address currently programmed in the EEPROM.
//
// Choose and enable exactly one of the following three examples...
//#ieee_addr
//#force_ieee_addr 0x00 0x11 0x22 0x33 0x44 0x55
#reuse_ieee_addr

#reserved 0x00 0x00 0x00

// Used by AMD device drivers.
#hwid 0x01

// User defined bytes.
#user1 0x00 0x00
```

NET186.DAT File Listing (continued)

```
// 16-bit checksum #1 - automatically computed by EESETUP.
#chksum1 0xC1 0xC2

// Used by AMD device drivers.
#driver_ww 0x57 0x57

#eisa_cfg 0x00 0x01 0x02 0x03

// Master Mode Read Active time set for 100ns for maximum performance
#isacsr0 0x02 0x00

// Master Mode Write Active time set for 100ns for maximum performance
#isacsr1 0x02 0x00

// #isacsr2: Bit 1 of the second byte is the P&P_ACT bit:
//      bit 1 = 0 = inactive/invisible, board needs Plug and Play to
wake it.
//    bit 1 = 1 = active/visible, board powers up visible on the ISA bus.
//
// bit 4 determines the ISA bus inactive time.  Setting this bit
// to a 1 will minimize this time for the Net186
//
// Choose and enable exactly one of the following two examples...
// If you want the adapter active/visible (Legacy):
#isacsr2 0x12 0x02
// If you want the adapter inactive/invisible (Plug and Play):
//#isacsr2 0x12 0x00

#isacsr5 0x84 0x00

#isacsr6 0x08 0x40

#isacsr7 0x90 0x00

// PCnet-ISA II (Full Duplex): Enable Full Duplex Register
// If you're using a PCnet-ISA II controller,
//    These two bytes are automatically included in the EEPROM image.
//   You must select the #pnp_unused field with 8 bytes to compensate.
// If you're using a PCnet-ISA+ controller,
//    These two bytes are NOT included in the EEPROM image.
//   You must select the #pnp_unused field with 10 bytes to compensate.
#isacsr9 0x00 0x00

// Plug and Play Registers 60/61. Base I/O address: 200
//  The address you specify here should also be specified on the
//  #io (Programmable I/O Port Descriptor) command line.
```

NET186.DAT File Listing (continued)

```
#pnp_i/o 0x02 0x00
// Plug and Play Registers 70/71. Interrupt Request: 3
//  The interrupt you specify here should also be specified on the
//  #irq (Interrupt Request Descriptor) command line.
#pnp_irq 0x03 0x00

// Plug and Play Register 74. DMA Channel: 3
//  The DMA channel you specify here should also be specified on the
//  #dma (DMA Channel Descriptor) command line.
#pnp_dma 0x03

#reserved 0x00

// Plug and Play Registers 40/41/42/43/44. ROM Address Descriptor.
//      The five byte descriptor syntax is as follows:
//          #pnp_rom 0x0M 0xN0 0xWW 0xS1 0xS2
//      M represents address lines 19-16 and N represents address
lines lines
//     15-12 of the ROM starting address. To get 8-bit ROM accesses you
//     substitute 00 for WW and to get 16-bit accesses you substitute
02 for
//    WW. For the ROM size you substitute FF for S1 and substitute one of
//      the following for S1:
//         if 8K use E0,  if 16K use C0,  if 32K use 80,  if 64K use 00
//
//      For boot ROM set all five bytes to 0x00.
//
//     Example: To specify a boot ROM at address 0xC8000 with 16-bit
//     accesses and a size of 32K...you substitute C for M and 8 for N.
//    To get 16-bit accesses you substitute 02 for WW. You substitute FF
//      for S1 and 80 for S2 to specify a 32K size.
//          #pnp_rom 0x0C 0x80 0x02 0xFF 0x80
//
//    Specific address alignment limitations for each size of boot ROM
//     are explained in the PCnet-ISA+ and PCnet-ISA II data sheets.
//
// Choose and enable exactly one of the following examples...
// No ROM.
#pnp_rom 0x00 0x00 0x00 0x00 0x00
// ROM @ 0C8000, 8-bit data, 8K in size.
//#pnp_rom 0x0C 0x80 0x00 0xFF 0xE0
// ROM @ 0C8000, 8-bit data, 16K in size.
//#pnp_rom 0x0C 0x80 0x00 0xFF 0xC0
// ROM @ 0C8000, 8-bit data, 32K in size.
//#pnp_rom 0x0C 0x80 0x00 0xFF 0x80
// ROM @ 0D0000, 8-bit data, 64K in size.
```

1.0

NET186.DAT File Listing (continued)

```
//#pnp_rom 0x0D 0x00 0x00 0xFF 0x00
#reserved 0x00

// Plug and Play Registers 48/49/4A/4B/4C. Shared SRAM Address
Descriptor.
// The Shared SRAM descriptor is disabled for AMD Bus Master boards.
#pnp_ram 0x00 0x00 0x00 0x00 0x00

#reserved 0x00

// Plug and Play Register F0. Vendor Defined Configuration Register.
// MUST BE 0x00 for AMD PCnet-ISA+ and PCnet-ISA II boards!
#pnp_vendid 0x41

// 8-bit checksum #2 - automatically computed by EESETUP.
#chksum2 0xC2

#ext_shft_chn 0x00 0x00

// If you're using a PCnet-ISA II controller,
//   The #isacsr9 bytes are automatically included in the EEPROM image.
//   You must select the #pnp_unused field with 8 bytes to compensate.
// If you're using a PCnet-ISA+ controller,
//    The #isacsr9 bytes are NOT included in the EEPROM image.
//   You must select the #pnp_unused field with 10 bytes to compensate.
//
// Choose and enable exactly one of the following two examples...
// If you're using the PCnet-ISA+:
//#pnp_unused 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
// If you're using the PCnet-ISA II:
#pnp_unused 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07

/
********.*********.******************+*****************.********/
/*              Plug and Play ISA Configuration Record             */
/*                 Plug and Play ISA Specification                 */
/*                           Version 1.0a                          */
/*                           May 5, 1994                           */
/********.*********.******************+************************/
/* The following Plug and Play fixed and variable length data
structures,   */
/* which are loaded into the EEPROM beginning at offset 64, are
absolutely, */
/* positively, required by EESETUP to be present and in the order
shown.    */
```

NET186.DAT File Listing (continued)

```
/*                                                                    */
/* It's possible to have data structures present that don't actually
*/
/* consume system resources. For example, a #mem_rom_desc structure
could   */
/* have a range length field of zero which disables the memory
requirement. */
/********.*********.************************.*********.********/

// Serial Identifier (8 bytes with 1 byte computed checksum)
// 32-bit EISA Vendor ID (AMD, PCnet-ISA+), 32-bit Vendor Serial/
Unique Number
#ser_id 0x04 0x96 0x55 0xAA 0x00 0x00 0x00 0x00 0xC3

// Plug and Play Version Number (small resource, 3 bytes)
#pnp_ver 0x0A 0x10 0x00

// ANSI Identifier String (large resource, 3 bytes+string)
//
// Choose and enable exactly one of the following examples...
//#ansi_id 0x82 0x27 0x00 AMD PCnet-ISA+ Ethernet Network Adapter
#ansi_id 0x82 0x29 0x00 AMD PCnet-ISA II Ethernet Network Adapter

// Logical Device ID (small resource, 5 bytes)
// 32-bit logical device ID (if single device same as #ser_id), flag
byte
// Bit 0/last byte indicates if device required for boot: 0=No, 1=Yes.
#log_dev_id 0x15 0x04 0x96 0x55 0xAA 0x02

// Compatible Device ID (small resource, 5 bytes)
// ID of other device with which this device is compatible.
#comp_dev_id 0x1C 0x41 0xD0 0x82 0x8C

// Memory Range Descriptor 0 (large resource, 12 bytes)
//     If memory is enabled for a boot ROM, make sure the #mem_rom_desc
//    field reflects the same configuration as the #pnp_rom field above.
//      If boot ROM is not enabled, you must edit the length field to
specify
//      a zero length.
//      This descriptor does not directly affect hardware settings.
//      This descriptor is supplied as a "courtesy" to Plug and Play
//    so that it may accurately determine the board's hardware settings.
//
//      Generic example for 16K boot ROM @ address 0xMN000 where M
represents
```

1.0

NET186.DAT File Listing (continued)

```
//      address lines 19-16 and N represents address lines 15-12.
//         0x81 0x09 0x00 0x62 0xN0 0x0M 0xN0 0x0M 0x00 0x40 0x40 0x00
//     For 8K ROM at address 0xC8000, you would substitute C for M and
//      substitute 8 for N and get the following.
//         0x81 0x09 0x00 0x62 0x80 0x0C 0x80 0x0C 0x00 0x20 0x20 0x00
//      Address alignment limitations for each size of boot ROM are
explained
//      in the PCnet-ISA+ and PCnet-ISA II data sheets.
//
// Choose and enable exactly one of the following examples...
// 8K boot ROM @ 0C8000
//#mem_rom_desc 0x81 0x09 0x00 0x62 0x80 0x0C 0x80 0x0C 0x00 0x20 0x20
0x00
// 16K boot ROM @ 0C8000
//#mem_rom_desc 0x81 0x09 0x00 0x62 0x80 0x0C 0x80 0x0C 0x00 0x40 0x40
0x00
// 32K boot ROM @ 0C8000
//#mem_rom_desc 0x81 0x09 0x00 0x62 0x80 0x0C 0x80 0x0C 0x00 0x80 0x80
0x00
// 64K boot ROM @ 0C8000
//#mem_rom_desc 0x81 0x09 0x00 0x62 0x80 0x0C 0x80 0x0C 0x00 0x00 0x00
0x01
// No boot ROM (length = 0)
//#mem_rom_desc 0x81 0x09 0x00 0x62 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00

// Memory Range Descriptor 1 (large resource, 12 bytes)
//  The Shared SRAM is disabled for AMD Bus Master boards.
// No Shared Static RAM (length = 0)
//#mem_ram_desc 0x81 0x09 0x00 0x33 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00

// Programmable I/O Port Descriptor (small resource, 8 bytes)
//  Min I/O Address, Max I/O Address, alignment, contiguous ports.
//  This descriptor does not directly affect hardware settings. It is
//  supplied as a "courtesy" to Plug and Play configuration software so
//   the board's hardware settings may be accurately determined.
//   The #io command should be in agreement with the #pnp_io command.
//
// Choose and enable exactly one of the following examples...
// I/O base 200h
#io 0x47 0x00 0x00 0x02 0x00 0x02 0x20 0x18
// I/O base 220h
//#io 0x47 0x00 0x20 0x02 0x20 0x02 0x20 0x18
// I/O base 240h
//#io 0x47 0x00 0x40 0x02 0x40 0x02 0x20 0x18
```

NET186.DAT File Listing (continued)

```
// I/O base 260h
//#io 0x47 0x00 0x60 0x02 0x60 0x02 0x20 0x18
// I/O base 280h
//#io 0x47 0x00 0x80 0x02 0x80 0x02 0x20 0x18
// I/O base 2A0h
//#io 0x47 0x00 0xA0 0x02 0xA0 0x02 0x20 0x18
// I/O base 2C0h
//#io 0x47 0x00 0xC0 0x02 0xC0 0x02 0x20 0x18
// I/O base 2E0h
//#io 0x47 0x00 0xE0 0x02 0xE0 0x02 0x20 0x18
// I/O base 300h
#io 0x47 0x00 0x00 0x03 0x00 0x03 0x20 0x18
// I/O base 320h
//#io 0x47 0x00 0x20 0x03 0x20 0x03 0x20 0x18
// I/O base 340h
//#io 0x47 0x00 0x40 0x03 0x40 0x03 0x20 0x18
// I/O base 360h
//#io 0x47 0x00 0x60 0x03 0x60 0x03 0x20 0x18
// I/O base 380h
//#io 0x47 0x00 0x80 0x03 0x80 0x03 0x20 0x18
// I/O base 3A0h
//#io 0x47 0x00 0xA0 0x03 0xA0 0x03 0x20 0x18
// I/O base 3C0h
//#io 0x47 0x00 0xC0 0x03 0xC0 0x03 0x20 0x18
// I/O base 3E0h
//#io 0x47 0x00 0xE0 0x03 0xE0 0x03 0x20 0x18

// DMA (small resource, 3 bytes)
//  DMA supported bit mask, Transfer Type Info (8/16-bit, Master).
//  This descriptor does not directly affect hardware settings. It is
//  supplied as a "courtesy" to Plug and Play configuration software so
//  the board's hardware settings may be accurately determined.
//  The #dma command should be in agreement with the #pnp_dma command.
//
// Choose and enable exactly one of the following examples...
// DMA3
#dma 0x2A 0x04 0x05
// DMA5
//#dma 0x2A 0x20 0x05
// DMA6
//#dma 0x2A 0x40 0x05
// DMA7
//#dma 0x2A 0x80 0x05

// IRQ (small resource, 4 bytes)
//  IRQ supported bit mask, High true edge sensitive, Low true Level
```

NET186.DAT File Listing (continued)

```
sensitive
//  This descriptor does not directly affect hardware settings. It is
//  supplied as a "courtesy" to Plug and Play configuration software so
//   the board's hardware settings may be accurately determined.
//  The #irq command should be in agreement with the #pnp_irq command.
//
// Choose and enable exactly one of the following examples...
// IRQ3
#irq 0x23 0x08 0x00 0x09
// IRQ4
//#irq 0x23 0x10 0x00 0x09
// IRQ5
//#irq 0x23 0x20 0x00 0x09
// IRQ9
//#irq 0x23 0x00 0x02 0x09
// IRQ10
//#irq 0x23 0x00 0x04 0x09
// IRQ11
//#irq 0x23 0x00 0x08 0x09
// IRQ12
//#irq 0x23 0x08 0x10 0x09
// IRQ15
//#irq 0x23 0x08 0x80 0x09

// END TAG (small resource, 1 byte with 1 byte computed checksum)
#end_tag 0x79 0xC4

// end of net186.dat
```

Appendix F

TCP/IP Primer

Protocol stacks are one of the most difficult areas to understand about networking. Figure F-1 below shows a client and server communicating using FTP on top of a TCP/IP stack, using drivers for their particular Ethernet hardware. If you understood that sentence, you don't need to read further!

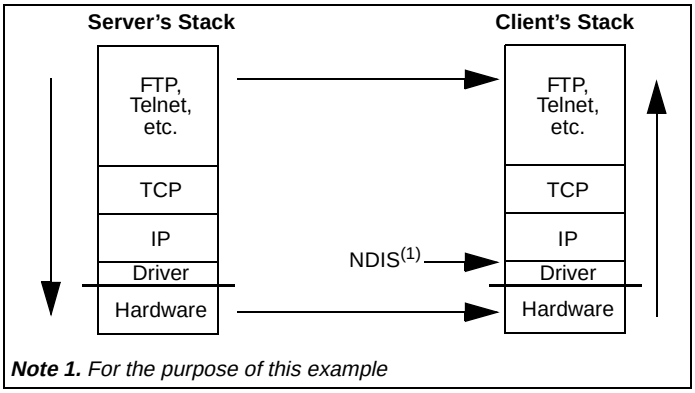


Figure F-1. Client and Server Protocol Stacks

Overview

TCP/IP and its associated applications like FTP are arguably the most used set of protocols in the embedded world. The basic idea is that applications do useful work, while everything below them is infrastructure. A key stack concept is that levels communicate with their peers by making requests of the level below while satisfying requests from above. The client FTP, for example, converses with the server FTP by sending data down through its stack and up through the server stack. Additionally, each successive layer has a clearly defined task:

- TCP is basically responsible for end-to-end reliability, meaning that it guarantees that each packet arrives at the other end without error and in sequence.
- IP on the other hand, doesn't care about reliability, but instead enables global connectivity with its IP addressing scheme (that's those dotted decimal addresses like 122.1.2.3 that you've probably seen).
- The driver is nothing more than a software shim that allows any vendor's Ethernet solution to work with standard IP software interfaces like NDIS.

The net result is that Ethernet hardware exchanges packets containing application data, a TCP header, and an IP header, all wrapped up with an 802.3 (Ethernet) header and CRC. Please see any of the widely available TCP/IP references for more detail.

1.0

Drivers, Protocol Stacks, and RTOS Support

Choosing a protocol stack is a matter of matching design requirements and budget to the available sources. Freeware stacks that reduce initial costs are available, but can take a lot of time to get working and may result in terrible performance. At the other end of the spectrum, vendors such as US Software offer a wide range of high-level protocols (such as SNMP), as well as porting and integration services. High-end stacks may cost more, but are worthwhile if performance and time-to-market are critical.

All commercial protocol stacks have a few target-dependent modules that provide independence from hardware drivers and real-time operating system APIs. To interface to a new hardware driver or RTOS, only these interface modules need to be changed so that generic calls like send_packet or task_wait are replaced by the calls specific to the given driver and RTOS. This means that you can use nearly any protocol stack with any combination of hardware drivers and real-time operating system.

Most stacks also come with drivers for the most common Ethernet chips and interface modules for the most popular RTOS'. Also some stacks come with a very simple pseudo-RTOS, which implements rudimentary task switching. If your needs are simple, these nano-kernels may work just fine. And of course you can always integrate a home grown RTOS. Table F-1 lists some protocol stack vendors and the protocols they support.

Table F-1. Protocol Stack Vendors

Vendor	Product	Protocols Supported
US Software 800-356-7097 www.ussw.com	USNET	TCP/IP, FTP, TELNET, PING, SNMP, etc.
EBS, Inc. 508-448-9340 www.etcbin.com	RT-IP	TCP/IP, FTP, TELNET, PPP, etc.
Accelerated Technology 800-468-6853 www.atinucleus.com	Nucleus NET	TCP/IP
Pacific Softworks 800-541-9508 www.pscificsw.com	Fusion TCP/IP	TCP/IP, SNMP, PPP, SMTP, etc.
Epilogue 505-271-9933 www.epilogue.com	Attache +	IP, UDP, TCP, RMON

1.0

Most of these stack vendors have separate prices for various protocol extensions. Typically, a base set includes the basic protocols like IP, UDP, PING, and TELNET. You then pay extra for extensions like TCP, FTP, PPP, SNMP, RMON, and other higher level protocols and applications. Different vendors have different combinations of up-front charges and per-copy royalties. You will have to get quotes from each vendor, and compare them for your anticipated needs.

The Net186 demonstration board comes with a TCP/IP stack and an example web server application from US Software. For more information about these, see the US Software literature include in your kit.

For freeware protocol stacks, download Packet Driver from **http://www.crynwr.com/crynwr/**. Packet Driver is not a TCP/IP stack itself. If you unzip the file and look at SOFTWARE.DOC, you will find a list of various protocol stacks (including several TCP/IP packages) and other applications that support Packet Driver. Many of these applications should be suitable for embedded applications.

Table F-2 lists some RTOS vendors that advertise in *Embedded Systems Programming* or *Electronic Engineering Times*. They support the 80186 16-bit processors.

Table F-2. RTOS Vendors

Vendor	Product	Web Site
US Software 800-356-7097	SuperTask!	www.ussw.com
Kadak, Inc. 604-734-2796	AMX	www.kadak.com
Accelerated Technology 800-468-6853	Nucleus	www.atinucleus.com
Embedded Systems Product 800-525-4302	RTXC	www.esphou.com
CMX 508-872-7675	CMX	www.cmx.com

10

Also, see the March 1997 issue of *Embedded Systems Programming.* This issue contains a special report on Real-Time Operating Systems written by Nicholas Cravotta. This article references a web page which summarizes the results. The URL is: **http://www.embedded.com/97/SR9703**.

For a list of RTOS vendors that specifically support the PCnet-ISA II Ethernet controller, see the *Fusion E86^{SM} Catalog*, the *Embedded Systems Programing* article mentioned above, and *Third Party RTOS Software Vendors Committed to Support PCnet Family,* order #20430.

For a list of shareware/freeware RTOS', check out **http://www.eg3.com/realxrto.htm**.

Index

Numerics

A

B

C

D

E

F

H

I

L

M

N

1.0

P

R

S

T

U

W

1.0