Protocol Conversion Systems: Market Overview

In this report:

Synopsis

Editor's Note
This report focuses on the protocol conversion systems market. It describes the industry's origins, the market leaders, and market trends. For information on the technology of protocol conversion, see "Protocol Conversion Systems: Technology Overview" (Report C23-010-201). Comparison columns listing detailed characteristics of more than 120 conversion products from 33 different vendors can be found in "Protocol Conversion Systems: Comparison Columns" (Report C23-010-301).

Highlights
Protocol conversion technology provides a way to link incompatible host computers and devices. A major portion of this market addresses incompatibilities between IBM (synchronous) and non-IBM (asynchronous) hosts, displays, and printers. Conversion is also necessary for device and host access to packet-switching networks; communications between PCs or LANs and host computers; and connection of devices using different physical interfaces, data codes, and communications speeds.

Until IBM entered the market in 1982, other vendors of protocol conversion products flourished. Another setback to the industry has been the shift away from host-controlled display terminals in favor of personal computers configured for terminal emulation.

The traditional protocol converter has largely given way to communications controllers capable of linking multiple environments and devices. Niche markets, such as Macintosh-to-IBM connectivity, also provide the most inventive vendors with fresh avenues for business.

—By Martin Dintzis
Assistant Editor

© 1991 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Research Group. Delran NJ 08075 USA

Analysis

Market Overview

The market for protocol conversion systems developed as a solution to the incompatibility problems between IBM and non-IBM display terminals, printers, and hosts. IBM made its part of the world synchronous, while other vendors made theirs asynchronous. Connecting peripheral equipment from other vendors to IBM hosts spawned a new industry dedicated to smoothing out the differences between the two worlds. Since asynchronous displays were generally less expensive than IBM products, protocol conversion also became a popular means to inexpensively connect large numbers of displays to an IBM system.

After recognizing the need for asynchronous-to-synchronous transmission solutions, KMW Systems of Austin, TX (now known as Andrew/KMW) set out to fill the void, thereby establishing itself in 1971 as the pioneer of the protocol conversion market. Thereafter, other companies, such as Local Data (now known as Andrew Corp.), Micom Communications, and Netlink, entered the market, each bringing its own expertise to that field.

These protocol conversion manufacturers flourished until 1982, when they received a setback initiated by IBM. Presumably acting under the dictum, "If you can't beat them, join them," IBM released its own line of protocol converters.

The proliferation of private and public packet-switching networks in the latter half of the 1980s increased the need for conversion between the CCITT X.25 packet data mode and IBM BSC, IBM SNA/SDLC, and asynchronous transmission modes. As a result, some vendors of asynchronous-to-IBM protocol conversion products, including Memotec Data, Micom, and Plantronics Futurecomms, also offer X.25-to-IBM and X.25-to async connectivity.

The increasing need to link multiple incompatible computers and devices has spawned the development of other conversion products, including software for front-end processors, emulation cards, interface adapters, multifunction communications controllers, and gateways.

Market Leaders

Andrew Corp. acquired Local Data, a leading protocol conversion manufacturer, in 1987. Local Data had developed the DataLynx, InterLynx, and VersaLynx product lines, which provide conversion between asynchronous and IBM BSC or SNA/SDLC environments for displays, printers, and PCs emulating displays. These devices are still marketed under Andrew's name.

Within the past two years, Andrew has released a steady stream of conversion products for both IBM mainframe and midrange environments, including the InterLynx/400 Protocol Converter and the Newport/Coax and Newport/Twinax synchronous adapters for Hewlett-Packard LaserJet printers. InterLynx/400 allows up to seven asynchronous display terminals, printers, or personal computers emulating displays to access an IBM AS/400 or System/3X host.

Andrew's protocol converters and display terminal adapters provide concurrent user access to both synchronous and asynchronous computers. The vendor's printer adapters allow a display- or PC-attached printer to be shared by both a host computer and the workstation user.

Andrew/KMW (formerly KMW Systems, which was acquired by Andrew Corp. in 1990) continues to blaze trails in the protocol conversion market by offering Macintosh connectivity products. Last year, the vendor introduced NetAxcess, the first adapter board that transforms a Macintosh II personal computer into a gateway capable of linking an entire AppleTalk network with an IBM midrange host. Macintosh workstations appear as IBM 52XX or 31XX displays, while Apple printers emulate IBM 52XX printers. Each Macintosh user has access to up to seven concurrent IBM midrange host applications and any number of Macintosh-resident applications.

Andrew/KMW also supports Macintosh access to IBM midrange environments through its TwinAxcess Series II (multiport) and TwinAxcess Series III (single port) protocol converters. Series II (multiport) and Series III (single port) products for

© 1991 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Research Group. Delran NJ 08075 USA

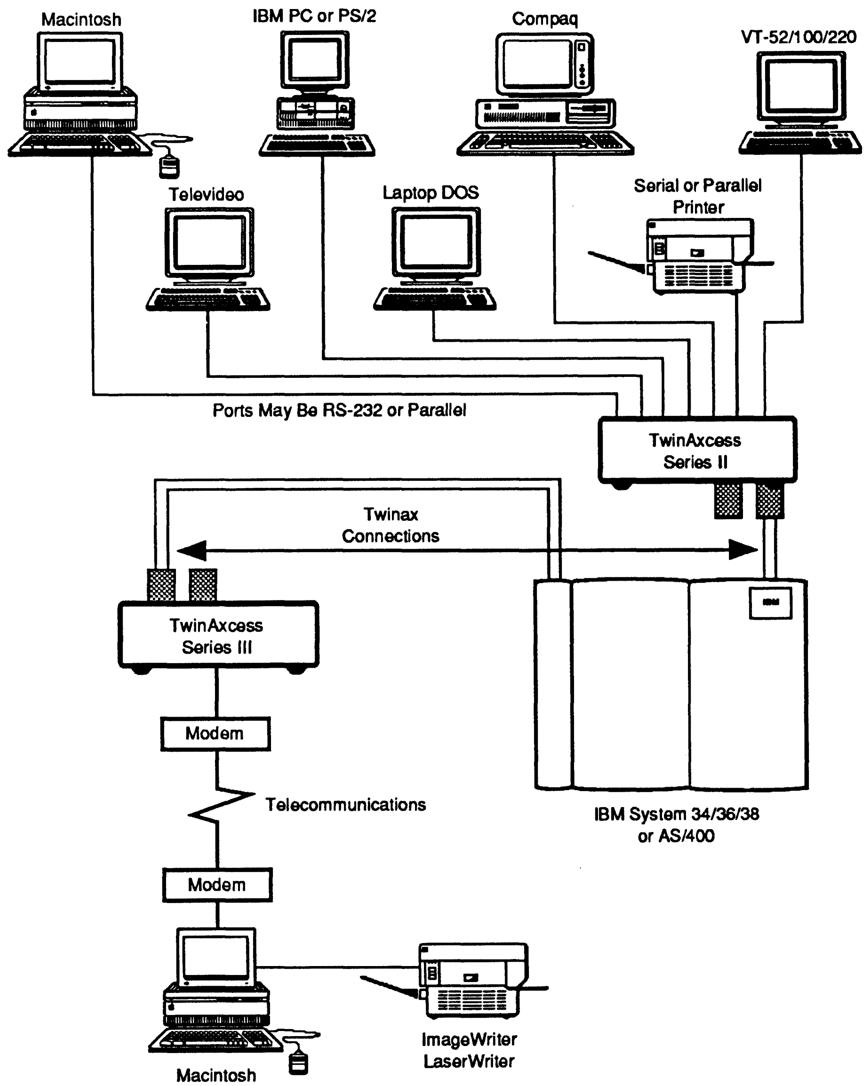


Figure 1.
Andrew/KMW's TwinAxcess Protocol Converters

TwinAxcess Series II accommodates up to seven local or remote asynchronous devices, including IBM-compatible and Macintosh personal computers, display terminals, and serial or parallel printers. TwinAxcess Series III is a one-port version of the TwinAxcess Series II unit.

3270 (IBM BSC, SNA/SDLC, and RJE) connectivity form another part of the vendor's product line.

IBM provides bidirectional conversion for both synchronous and asynchronous devices through the 3174 Establishment Controller, which also provides token-ring gateway functionality. IBM also continues to market the 3708 Network Conversion Unit and the 7171 Protocol Converter. The 3708 converts a 3270 datastream to and from ASCII code, allowing asynchronous devices to appear as 3270 displays and printers to an IBM SNA host. The 7171 can support from 16 to 64 asynchronous ASCII devices via an RS-232-C interface to the block multiplexer channel of an IBM host.

Micom Communications markets the Micom Box Type 3 unit, a network processor that can be configured, through a selection of software cartridges, for operation as an async-to-SNA/SDLC or async-to-BSC protocol converter; an async, SNA/SDLC, BSC, or multiprotocol (async/SNA or async/BSC) packet assembler/disassembler (PAD); or an X.25 packet switch or switching PAD.

Netlink offers SNA_Gate, a versatile product that can function as a protocol converter, a cluster controller, a line concentrator, and a remote job entry station facility. Connecting to an IBM 37XX communications controller, SNA_Gate provides async-to BSC, async-to-SNA/SDLC, or BSC-to-SNA/SDLC conversion, accommodating up to 250 devices over multidrop lines.

Future Directions

Replacing older display terminals with microcomputers configured for terminal emulation has become a common practice. Users want access to

© 1991 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Research Group. Delran NJ 08075 USA

more than one computer system but do not want two terminals taking up space on their desks. By the early 1980s, organizations confirmed their preferences for micros over display terminals, installing them at a rapid rate and benefiting from their programmability. The shift from host-based systems to local area networks has heightened this trend, thereby weakening both the display terminal and protocol conversion industries.

The need for protocol conversion remains strong, however, because of the increasing need to link multiple dissimilar environments. Microcomputers have encouraged the development of new terminal emulation hardware and software products, including LAN gateway solutions. Products that link Macintoshes to IBM host environments, for example, are in demand, as evidenced by the product introductions of Apple Computer, Andrew/KMW, and other vendors.

While the sale of traditional protocol converters is on the decline, vendors throughout the IBM display system market, including AT&T, Apertus Technologies (formerly Lee Data), IBM, IDEA Courier, and Memorex Telex, have been successful in marketing large communications controllers capable of transparently linking multiple IBM hosts with large numbers of devices distributed across IBM 3270/5250, asynchronous, and token-ring environments. Some of these systems also provide enhanced functionality, such as multiple sessions with windowing for attached display terminals.

As businesses continue to expand and merge, the use of packet-switching networks to link multiple remote IBM and non-IBM environments remains a widespread practice. The sale of multiprotocol PADs, therefore, will continue to be a major source of revenue to many vendors of protocol conversion products. ■

© 1991 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Research Group. Delran NJ 08075 USA

Protocol Conversion Systems: Market Overview

Synopsis

Editor's Note
This report focuses on the market for protocol conversion systems. For information on the technology of protocol conversion, see "Protocol Conversion Systems: Technology Overview," Report C23-010-201. For detailed comparison column listings of protocol conversion products, see Report C23-010-301.

Highlights
The market for protocol conversion systems developed as a solution to the asynchronous/synchronous problem reflected by the incompatibility of non-IBM terminals with IBM hosts. IBM made its significant part of the world synchronous while other vendors made theirs asynchronous. Connecting peripheral equipment from other vendors to IBM hosts spawned a new industry dedicated to smoothing out the differences between the two worlds.

After recognizing the need for IBM-compatible synchronous transmission solutions, KMW Corporation of Austin, Texas set out to fill the void, thereby establishing itself in 1971 as the pioneer of the protocol conversion market. Since that time, other companies, notably Renex, Andrew Corporation, and Netlink, have entered the market, each bringing its own expertise to the industry.

Protocol conversion manufacturers flourished until 1982 when they received a setback initiated by IBM. Presumably acting under the dictum, "If you can't beat them, join them," IBM released its own line of protocol converters.

At present, IBM offers the 3708 Network Conversion Unit and the 7171 Protocol Converter. The 3708 converts a 3270 datastream to and from ASCII code, allowing asynchronous devices to appear as 3270 displays and printers to an IBM SNA host. The 7171 can support from 16 to 64 asynchronous ASCII devices via an RS-232-C interface to the block multiplexer channel of an IBM host.

Although protocol conversion vendors do not enjoy the thriving market they once did, they continue to enhance their product lines.

© 1990 McGraw-Hill, Incorporated. Reproduction Prohibited. Datapro Research.
Delran NJ 08075 USA

Analysis

In addition to IBM's entrance into the protocol conversion market, the practice of businesses replacing older terminals with microcomputers strongly hampered the growth of the standalone protocol conversion industry. By the early eighties, organizations confirmed their preferences for micros over terminals, installing them at a rapid rate and benefiting from their internal conversion devices.

Contributions to softening the market have also come from other areas. Some vendors are offering data switches that incorporate protocol conversion capabilities. PBX vendors are marketing products that perform protocol conversion at the board level to link ASCII devices in back of the switch into synchronous networks.

These developments challenged the ingenuity of vendors, some of whom took a page from the books of plug-compatible mainframe manufacturers and produced a product that acted like an IBM controller, but cost considerably less. Some vendors market devices that emulate an IBM cluster controller, but deliver greater functionality to the user at a lower price. Going the cluster controller route enabled vendors to design products that perform both conversion and emulation.

Although the market has dwindled, the need for protocol conversion has not. Aware that the need still exists, vendors have added new capabilities to their products and increased their capacities.

The Cx-81 from Commtex, Inc. provides asynchronous ASCII displays, printers, and PCs with access to two IBM mainframes.

Activity in 1989

Andrew Network Products

Andrew Corporation acquired Local Data, a leading protocol conversion manufacturer, in 1987. Local Data, now part of Andrew Network Products, offered the DataLynx, InterLynx, and VersaLynx lines of conversion products. These devices are now marketed under Andrew's name.

In 1989, Andrew expanded the product family with the introduction of TruLynx/400. When used with the InterLynx 5251 protocol converter, TruLynx/400 provides a PC interface that is similar in many aspects to the IBM twinax card. The product enables users to access PC programs on an AS/400 or System/3X through a number of asynchronous devices.

Andrew also produces the MALIBU/Coax protocol converter, which uses National Semiconductor's DP8344 Biphase Communications Processor. The MALIBU/Coax device allows PC and mainframe users to share an ASCII printer and dispenses with manual intervention.

Apple Computer

In 1989, Apple entered the protocol conversion business with a serial card, the Apple Serial NB Card, which is a board-level controller within the Macintosh. The card enables the Macintosh to link to remote systems via RS-232-C, X.21, and V.35 protocols. An on-board Motorola 68000 microprocessor handles protocol conversion between the Macintosh and IBM computers.

An expansion card enables Macintosh IIs to connect to an IBM SNA network as if they were 3270s. When operating with MacDFT, the card allows users to access mainframe applications just as they would from terminals. The card features a twinax connector for 5250 emulation.

Commtex Inc.

In June 1989, Commtex Inc. announced the release of Cx-81, a five-port ASCII-to-3270 protocol converter. The product enables asynchronous ASCII displays, printers, and PCs to access two IBM mainframes. Each device can conduct up to five concurrent sessions with one or both hosts. Each host communication line can independently support BSC or SNA/SDLC protocols.

Emulating 3174 control units, the Cx-81 allows asynchronous devices to be directly connected

© 1990 McGraw-Hill, Incorporated. Reproduction Prohibited. Datapro Research.
Delran NJ 08075 USA

via nine-pin "D" connectors at speeds up to 38.4K bps. Users can also remotely connect the devices via dial-up or leased lines. The Cx-81 supports over 250 terminal emulations. Bundled software enables the user to add support for new terminals or customize those already supported.

Recently, Commtex has increased the density of its large controller to 50 ports, but Donald Parker, president of Commtex, observed, "We didn't want to forsake the small remote office so we distilled Cx-80 essentials into an inexpensive five-port unit."

KMW Systems

KMW continues to blaze trails in the protocol conversion market. In September 1989, the company announced a new capability for its TwinAxcess line of protocol converters. Both models, the Series II and Series III, allow the new Apple Macintosh Portable to communicate locally or remotely with an IBM AS/400 or IBM System/34, /36, or /38.

When equipped with an optional internal modem, the Macintosh Portable can access files on the IBM midrange computers from any location over a telephone line. A modem attached to the TwinAxcess protocol converter at the host site completes the connection. An alternative approach with an external modem attached to the Macintosh computer provides the same functionality.

TwinAxcess Series II (multiport) and TwinAxcess Series III (single port) perform terminal emulation and file transfer between the Portable and the IBM midrange. The Macintosh can emulate either an IBM 5251 or 5291 terminal when connected to an AS/400 or System/3X. File transfer can occur when KMW's TwinAxcess LINK software runs on the Macintosh Portable in conjunction with a TwinAxcess protocol converter and file transfer software on the host.

Renex Corporation

An active participant in protocol conversion since 1980, Renex originally produced converters that replaced controllers. The company's newest offering, the Protocol-Converting Multiplexer (PCM), performs SNA protocol conversion through existing controllers, without making use of PC converter cards. The product enables ASCII devices to dial up or directly access an IBM mainframe by means of IBM 3X74 controllers.

The latest Renex release pares down operating costs by eliminating the need to buy PC cards to support new network users. This innovative approach of plugging directly into the controller instead of replacing it infuses new life into the market. An additional bonus for users is PCM's support of dial-up access, a feature not found on a PC card. ■

© 1990 McGraw-Hill, Incorporated. Reproduction Prohibited. Datapro Research.
Delran NJ 08075 USA

Protocol Conversion Systems

Protocols, whether employed in a military chain of command or in data communications, define procedures necessary to assure mutual understanding. Protocol conversion systems include protocol converters, terminal controllers and emulators, packet assembler/disassemblers (PADs), gateways, and network processors. Using the Open Systems Interconnection (OSI) seven-layer model for data communications as a reference point, we present various conversions that can occur in a network.

A vendor list and two sets of comparison columns follow this report. The comparison columns present specifications for protocol conversion systems and code, speed, and interface converters.

REPORT HIGHLIGHTS: PAGE

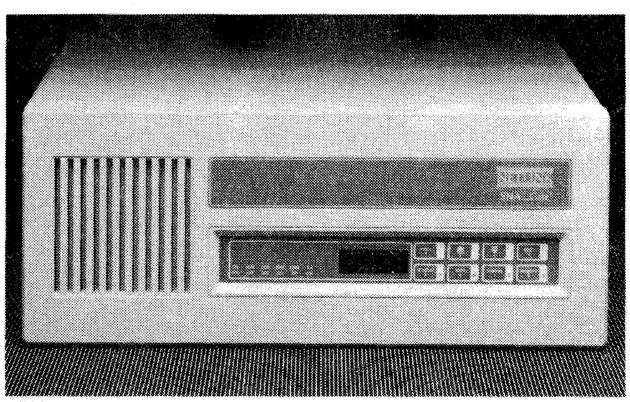
The term *protocol conversion* means far more than simply translation from one protocol to another. The process may encompass numerous products, including protocol converters, emulation devices, gateways, and packet assemblers/disassemblers (PADs), that allow compatibility among communications devices, local area networks, packet switched networks, or computer operating systems. Available products range from microprocessor-based circuit boards to front-end processors (FEPs) capable of performing conversion functions through software applications. Some devices perform only code or interface conversions, while others handle protocol conversion, device emulation, and/or code and interface translations in the same unit.

This report focuses on standalone hardware products that perform conversions to allow equipment from one manufacturer to communicate with another manufacturer's equipment. The largest market segment for these products addresses incompatibilities between IBM's synchronous mainframes and asynchronous ASCII terminals.

For information on software packages performing conversion and emulation, consult the *Datapro Directory of Software* and the *Datapro Directory of Microcomputer Software*. For coverage of micro-to-mainframe conversion products, see Report C22-010-301, "Microcomputer-to-Host Communications Products," in this volume. We now cover PADs in a separate report, C20-010-301.

PROTOCOLS

Protocols govern the format of a data exchange, recognition of a remote connection, identification of the transmitting and receiving locations, the transmission sequence, the handling of interruptions, error-checking methods and control, methods of blocking data, and security procedures. They range from single character-by-character communications with no error checking to complex algorithms moving data among many devices.



In general, protocols specify three major areas:

- The method in which data is to be represented or encoded—the code set. Most data processing systems today use either the American Standard Code for Information Interchange (ASCII) or IBM's Extended Binary Coded Decimal Interchange Code (EBCDIC).
- The method in which the codes are transmitted and received—asynchronous or synchronous. In asynchronous transmission, data is sent with start and stop bits between individual characters; data is sent at random intervals and does not require specific timing. In synchronous transmission, characters or bits are sent at a fixed rate; transmitting and receiving devices are synchronized, eliminating the need for start/stop bits.
- The nondata exchanges of information by which the two devices establish control, detect failures or errors, and initiate corrective action.

COPYRIGHT © 1988 McGRAW-HILL, INCORPORATED. REPRODUCTION PROHIBITED
DATAPRO RESEARCH, DELRAN NJ 08075 USA

Through hardware or software, the sending device automatically formats the data and adds the required bits before transmitting each block. The receiving device automatically checks each of the appended bits before acknowledging receipt of data. Upon detecting any failures, the protocol initiates error-control procedures.

Byte-oriented protocols require transmission of data in eight-bit blocks; an acknowledgment is required after each transmitted block before the next block can be sent. Bit-oriented protocols allow data to be transmitted in blocks of any length up to a specified maximum; an acknowledgment may take place after one or several blocks have been sent, depending on the protocol. Some of the most common protocols are as follows:

- ASCII or TTY—an asynchronous protocol that uses the ASCII code set. It provides very little error checking. Transmission is in the form of a start bit, a number of data bits (usually five to eight), and one or more stop bits. Data in ASCII protocol enters the communications line at any time; the end of the link is synchronized through the specifications of a common line speed and detection of the start bits and the beginning of the character transmission. ASCII requires an acknowledgment after each block is sent. ASCII protocol, often referred to as Teletype (TTY) protocol, traditionally relates to teletypewriter equipment and services.

- IBM's Synchronous Data Link Control (SDLC)—a bit-oriented synchronous protocol that uses a synchronized series of frames. Each frame has a synchronization flag, followed by an address field, a control field that tells the purpose of the transmission, the data itself, then a frame-check field, and finally a trailing flag. The flag character marks synchronization. SDLC permits up to 127 frames to be outstanding before an acknowledgment is required. Because SDLC supports full-duplex transmission, users can send multiple blocks of data on one acknowledgment. SDLC is used in private-line networks.

- IBM Binary Synchronous Communications (BSC)—a character-oriented synchronous protocol, also referred to as *bisync*. Binary synchronous data and control characters consist of eight-bit bytes. A transmission in BSC consists of a number of synchronizing (SYN) characters that ensure synchronization at both ends of the communications link. These are followed by a start-of-text (STX) character, an eight-bit block of text, an end-of-text (ETX) character, and a block error-checking character (BCC). BSC does not support full-duplex transmission, nor is it supported by IBM's Systems Network Architecture (SNA). An acknowledgment must follow each block of data. The BSC protocol works in multipoint applications over private lines.

Other communications protocols include High-Level Data Link Control (HDLC), a CCITT-specified, bit-oriented protocol upon which most other bit-oriented protocols are based; Univac U200, CDC UT200, and Burroughs Multipoint Poll Select, which are similar to IBM BSC but can run on both synchronous and asynchronous links; and Digital Equipment Corporation's Digital Data Communications Message Protocol (DDCMP), a byte-oriented protocol that can handle up to 255 unacknowledged transmissions.

ISO SEVEN-LAYER MODEL FOR DATA COMMUNICATIONS

(7) Application—provides communications services
(6) Presentation—defines syntax of data
(5) Session—controls data exchange
(4) Transport—handles data flow, error control
(3) Network—handles data routing
(2) Data Link—ensures data transfer via protocols
(1) Physical—provides mechanical/electrical interface

Figure 1. Layers One through Three define the interface between the host computer and the network. Layers Four through Seven provide compatibility to data format and exchange.

THE OSI MODEL

The International Standards Organization (ISO) Open Systems Interconnection reference model provides a framework for understanding the differences in conversion products. Each of the model's layers defines a particular aspect of the entire data communications process. Figure 1 illustrates the seven-layer hierarchy.

- Layer 1—*Physical Connection* provides mechanical and electrical specifications and procedures to establish, maintain, and end physical connections. This layer defines interface, code, speed, and synchronization functions. Interface, code, and asynchronous-to-synchronous converters fall into this category.

- Layer 2—*Data Link Control* ensures that the data passes without error from one computer to another. This process involves protocols that specify the format for data transmission. Protocol converters are the devices that handle conversions in this layer. Parameters such as modem control, ring signaling, and dedicated connections fall into this category.

- Layer 3—*Network Layer* lets two systems exchange data. This layer defines packet addressing and data routing to final destination. Units that handle conversion in this layer include gateway devices, such as packet assemblers/disassemblers that provide access to X.25 networks or between local area networks. Front-end pro-

COPYRIGHT © 1988 McGRAW-HILL, INCORPORATED. REPRODUCTION PROHIBITED
DATAPRO RESEARCH, DELRAN NJ 08075 USA

Protocol Conversion Systems

cessors (FEPs) that include protocol conversion in their functions also fall into this classification.

- Layer 4—*Transport Layer* handles end-to-end error and flow control to ensure that the communications exchange is orderly and reliable. PAD devices, a type of gateway product, are the major products in this layer. Note that PADs affect both the Network and Transport layers.

- Layer 5—*Session Layer* provides the structure for a data exchange by managing connections between application processes, establishing and terminating connections, and sending end-to-end messages and controller dialogs. There are currently few conversion products in this category.

- Layer 6—*Presentation Layer* both defines the way in which data is put together and provides a systematic arrangement for the communications exchange to occur. This layer defines functions to translate coded data and convert it into display formats for terminal or microcomputer screens, printers, and other peripherals. In this layer, data is expanded or compressed and structured for file transfer or command translation. Devices called emulators, which allow one type of terminal to appear as another type of terminal, fall into the Presentation Layer category. Products in this category include ASCII-to-3270 emulators, interfaces that let personal computers act as 3270-type devices or access public networks, and word processor interfaces that handle conversions between dissimilar word processors.

- Layer 7—*Applications Layer* supports user and application tasks and provides the communications services that are available to specific computer applications. In essence, this layer provides the meaning to the message. Conversion devices that we discuss in this report do not provide conversions on this layer.

Converters must often provide translations on more than one level in the model. Conversion at one layer generally implies a need for compatibility in lower layers. For example, a protocol converter working on Level 2 functions also assumes responsibility for compatibility in the interface, code, and synchronization functions.

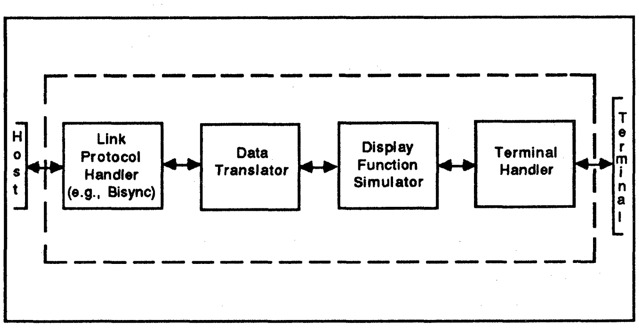


Figure 2. The protocol conversion process.

The Mechanics of Protocol Conversion

Protocol converters perform translations for dissimilar devices simulating the appropriate protocol for each. As Figure 2 shows, this gives protocol converters a distinctive, double-ended structure. For each end of the conversion process, a local protocol handler uses the protocol required by the attached device. Connecting these handlers is a gateway task that provides for the movement of user data between the handlers. If all communication protocols were structured in accordance with the OSI Reference Model, the converter would implement a set of seven-layer OSI protocols joined by the gateway task. Because the central task of a fully structured OSI protocol is to isolate users from the communication environment, a protocol converter dealing exclusively with full OSI-model protocols would be fairly simple to develop and would operate with few restrictions. With non-OSI protocols, such as those commonly in use in today's networks, the following issues complicate the conversion process:

- *The format of the user data.* If the data is easily separated from communication and device control protocols, it is more easily transferred to another environment. Special features like data compression, complicate protocol conversion if they do not exist in the other protocol.

- *The degree of layering in the protocols.* Even though full compliance with the OSI model is unlikely in the protocols being considered for conversion today, any amount of OSI-like layering in the protocols will aid in the separation of useful data from control information that must not be introduced into the other environment.

- *The availability of common functions in the protocols involved.* Data exchange between the users requires a degree of synchronization between the two foreign protocols. For example, most older protocols operate in half-duplex mode—only one station at a time may send information. It is necessary for converters operating between half-duplex protocols to insure that both stations are not given permission to send at the same moment, since neither could receive under those circumstances.

When protocol converters allow devices to simulate other devices, device control protocol translation may be needed. IBM's popular 3270 series of terminals is often emulated using lower cost asynchronous devices, but the 3270 has special features, such as the ability to return only modified fields to the host computer. This ability must be emulated within the protocol converter. Figure 3 shows the structure of a terminal emulator protocol converter.

COPYRIGHT © 1988 McGRAW-HILL, INCORPORATED. REPRODUCTION PROHIBITED
DATAPRO RESEARCH, DELRAN NJ 08075 USA

Interface, Code, and Asynchronous-to-Synchronous Converters

An interface provides the physical connection between two devices. Interface conversion offers the lowest level of established compatibility. Data and control lines from devices terminate at a connector that handles assigned signal functions. For example, the RS-232-C interface connector has 25 pins—1 pin per function. The interface also prescribes voltage levels for electrical signals passing over the data and control lines.

Interface converters serve as adapters for differing interfaces, accept the connectors of two different interfaces, and/or translate signals and voltage levels of one interface to another. Interface conversions commonly occur between RS-232-C and MIL-STD-188 or between RS-232-C and V.35.

Code converters translate one communications code to another. The most common codes are ASCII, EBCDIC, and Baudot. Conversion from one code to another may be simple, involving only the addition or deletion of control bits or the alteration of parity. A more complex code conversion might require changing the data character's bit pattern.

Basic code conversion hardware consists of two universal synchronous/asynchronous receiver/transmitters (USARTs), a translation table contained in ROM, and control circuitry. Characters received by the USART in one code are mapped in the ROM table into a corresponding character in the destination device's code. Converted data goes to the other USART, which transmits it to the destination device.

Asynchronous-to-synchronous converters perform conversion of data from asynchronous terminals for use on synchronous facilities.

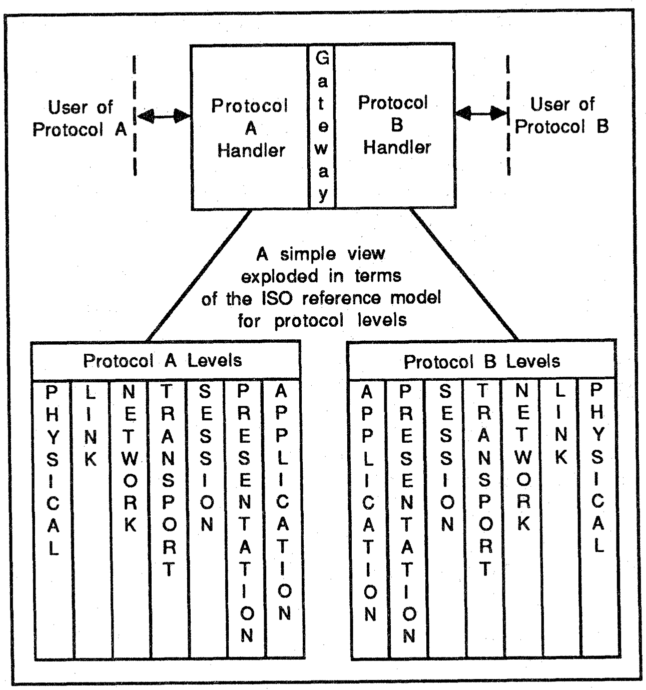


Figure 3. Inside a terminal emulator.

PROTOCOL CONVERTERS

Protocol converters, one of the largest categories of conversion devices, perform changes at the Data Link Layer to ensure device compatibility. Protocol converters connect incompatible peripheral devices to hosts utilizing microprocessors. A protocol converter actually changes one protocol to another by separating control characters from data and assembling the new datastream according to new specifications.

During the conversion sequence, the converter accepts blocks of data, adds or deletes the necessary control characters, reformats the block, and calculates the required check characters so the receiving device receives characters formatted according to its requirements. For example, in an ASCII-to-SDLC conversion, the converter accepts a character string, eliminates start and stop bits, assembles characters into a block, and adds headers and trailers to create complete frames. In a BSC-to-SDLC conversion, the converter changes the first four SYN bits of the bisync algorithm to the first flag bit of the SDLC algorithm.

Since protocol converters must stop, store, process, and retransmit data, they usually increase response time. The device generally accepts low-speed input in the buffer; works with the data; and then transmits it out in short, high-speed bursts.

Gateways and PADs

Gateways and PADs perform conversions on OSI Layers Three and Four (the Network and Transport Layers) and also perform lower layer functions. Gateway devices allow access to incompatible networks, for example, between SNA and DECnet, or between SNA and Ethernet, or between a data communications device and an X.25 public data network. Gateways also provide compatibility between network architectures' inherent protocols, codes, and interfaces. We cover these products in Report C11-010-301, "Local Area Network Products."

By far the largest subset of gateway products are packet assembler/disassemblers (PADs). We now cover these products in a separate report, C20-010-301.

COPYRIGHT © 1988 McGRAW-HILL, INCORPORATED. REPRODUCTION PROHIBITED
DATAPRO RESEARCH, DELRAN NJ 08075 USA

Protocol Conversion Systems

Emulation Devices

While protocol converters resolve incompatibility problems between devices, an emulator resolves incompatibilities including differences in protocol, code, interface, device characteristics, and link characteristics. To the emulator, protocol conversion is secondary.

Many—but not all—protocol converters today provide both protocol conversion and emulation, whereas all emulation devices provide protocol conversion. Commonly, devices performing protocol and emulation translations are called value-added terminal controllers, remote cluster controllers, or terminal emulators.

An IBM 3271 serves up to 32 IBM 3277-type terminals on a multipoint line. Data moving in this configuration is blocked out in 1,920-character screen images (blocks of data). If a user wants to replace IBM 3277 terminals with asynchronous ASCII devices, the ASCII units must appear as IBM 3277s to the IBM host. A terminal controller/emulator solves the problem by accumulating an asynchronous datastream in its buffer until a 1,920-character screen image is filled, or until the emulator receives an end-of-record, end-of-block control character. The terminal controller converts the ASCII terminal protocol to the host protocol (i.e., BSC); rearranges the data format to appear as if it comes from an IBM 3271; and then transfers the screen image to the host, which recognizes the data of an IBM 3277—not an asynchronous ASCII terminal. The terminal controller performs all functions of the device it replaces, including data concentration; poll/select; flow control; buffering; error detection and correction; and interfacing of multiple, attached terminals.

Sometimes the emulating device connects to an IBM cluster controller rather than replacing it. It then, in effect, performs the conversion between the terminal and the IBM controller instead of between the controller and the host. The purpose of these emulators is to allow the user to integrate incompatible equipment into an existing terminal cluster.

During an emulation/conversion/transfer sequence, the emulator interprets control sequences from a terminal to simulate the emulated terminal's operations. The equivalent control sequence for one terminal and another differs widely. For example, no asynchronous ASCII keyboard provides all of the special 3270 function keys.

Many users install terminal controllers to allow non-IBM devices in remote locations to access IBM mainframes. Many remote controllers have one synchronous line for 3270 access and two or more minicomputer interfaces. Local users can switch between hosts depending upon application.

Although most protocol conversion systems perform ASCII-to-IBM conversions, other products provide conversion between IBM BSC protocols and IBM SDLC protocols. Users of older IBM BSC equipment planning to migrate to an SNA/SDLC environment benefit from these products without replacing their old equipment. BSC-to-SDLC conversions generally occur between BSC 2780/3780 RJE or 3270 BSC protocols and SDLC protocols.

THE MARKET

The market for standalone protocol conversion systems emerged in the mid-1970s and grew fairly rapidly until the beginning of the 1980s. At this time, businesses began replacing older terminals with microcomputers, which supported internal conversion devices. This dampened the need for standalone conversion products considerably.

During the late seventies and early eighties, IBM introduced its own protocol converters, seriously affecting the market positions of a number of small companies that had carved out lucrative market niches in the field. IBM dealt a final blow to the market when it introduced the 3174 cluster controller, which supports direct connection of ASCII terminals.

The future of the protocol conversion systems market is not bright. In response to IBM's 3174 announcement, protocol conversion vendors lowered prices on their products, sometimes more than 25 percent. Vendors in the conversion market must now adapt their products for special applications or align themselves with vendors selling complete systems, which complement conversion units. In this year's comparison columns, we note that more vendors are supporting Burroughs and Honeywell applications and moving away from the ASCII-to-IBM product line.

COMPARISON COLUMNS

The columns that accompany this report present listings of the key characteristics of approximately 73 protocol conversion systems and 50 code, speed, interface, and async/sync converters.

In July and August 1988, Datapro surveyed 108 firms known or believed to manufacture some type of hardware conversion device. *The absence of any company from the columns means that the company either failed to respond to our request by the survey deadline, did not produce a hardware conversion product, or chose not to be listed.*

COPYRIGHT © 1988 McGRAW-HILL, INCORPORATED. REPRODUCTION PROHIBITED
DATAPRO RESEARCH, DELRAN NJ 08075 USA

Protocol Conversion System Vendors

Arkansas Systems
8901 Kanis Road
Little Rock, AK 72205 (501) 227-8471

Commtex Inc.
1655 Crofton Boulevard, Suite 300
Crofton, MD 21114 (301) 721-3666

Computer Communications, Inc.
2610 Columbia Street
Torrance, CA 90503 (213) 320-9101

Datagraf, Inc.
8305 Highway 71 West
Austin, TX 78735 (512) 288-0453

Digital Controls Corp.
3495 Newmark Drive
Miamisburg, OH 45342 (513) 435-5455

Gandalf Data, Inc.
1020 S. Noel Avenue
Wheeling, IL 60090 (312) 541-6060

INCAA Datacom b.v.
Amerfoortseweg 15
7313 AB Apeldoorn, Holland

Instrumentation Services Inc.
957 Winnetka Avenue North
Minneapolis, MN 55427 (612) 544-8916

JBM Electronics Co.
6020 N. Lindbergh Boulevard
Hazelwood, MO 63042 (314) 731-7781

Jupiter Technology, Inc.
78 Fourth Avenue
Waltham, MA 02154 (617) 890-4555

KMW Systems Corp.
100 Shepherd Mountain Plaza
Austin, TX 78730-5014 (512) 338-3000

Lemcom Systems, Inc.
2104 W. Peoria Avenue
Phoenix, AZ 85029 (602) 944-1543

Local Data, Inc.
2771 Plaza Del Amo
Torrance, CA 94086 (213) 320-7126

Memotec Data Inc.
600 McCaffrey
Montreal, PQ, Canada H42 1N1 (514) 738-4781

Micom Systems, Inc.
4100 Los Angeles Avenue
Simi Valley, CA 93062 (805) 583-8600

Netlink, Inc.
3214 Spring Forest Road
Raleigh, NC 27604 (919) 878-8612

Renex Corp.
1513 Davis Ford Road
Woodbridge, VA 22192 (703) 494-2200

Shaffstall Corp.
7901 E. 88th Street
Indianapolis, IN 46256 (317) 842-2077

Software Results Corp.
2887 Silver Drive
Columbus, OH 43211 (614) 267-2203

Trax Softworks, Inc.
10801 National Boulevard
Los Angeles, CA 90064 (213) 475-8729

Wall Data Inc.
17769 NE 78th Place
Redmond, WA 98052 (206) 883-4777

Western DataCom
5083 Market Street
Youngstown, OH 44512 (216) 788-6583

Unisync Inc.
508 N. First Avenue
Upland, CA 91786 (714) 985-5088

VENDORS OF CODE, SPEED, AND INTERFACE CONVERTERS

Com/Tech Systems
10 Halyard Road
Valley Stream, NY 11581 (516) 791-1175

DCC Corp.
7300 N. Crescent Boulevard
Pennsauken, NJ 08110 (609) 662-7272

COPYRIGHT © 1988 McGRAW-HILL, INCORPORATED. REPRODUCTION PROHIBITED
DATAPRO RESEARCH, DELRAN NJ 08075 USA

Protocol Conversion System Vendors

Digital Controls Corp.
3495 Newmark Drive
Miamisburg, OH 45342 (513) 435-5455

Gandalf Data, Inc.
1020 S. Noel Avenue
Wheeling, IL 60090 (312) 541-6060

General DataComm Industries, Inc.
Route 63
Middlebury, CT 06762 (203) 574-1118

INCAA Datacom b.v.
Amerfoortseweg 15
7313 AB Apeldoorn, Holland

Method Systems, Inc.
3511 Lost Nation Road, 202
Willoughby, OH 44094 (216) 942-2100

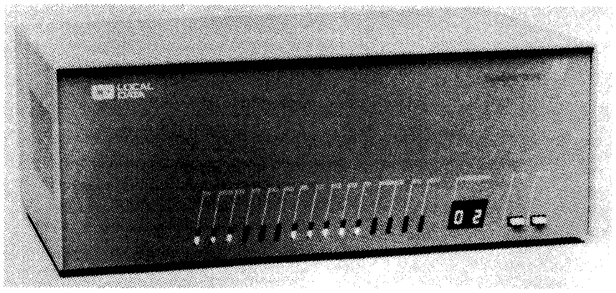
Shaffstall Corp.
7901 E. 88th Street
Indianapolis, IN 46256 (317) 842-2077

Teleprocessing Products Inc.
4565 E. Industrial Street, Suite 7-K
Simi Valley, CA 93063 (803) 522-8147

Wall Data Inc.
17769 NE 78th Place
Redmond, WA 98052 (206) 883-4777 □

COPYRIGHT © 1988 McGRAW-HILL, INCORPORATED. REPRODUCTION PROHIBITED
DATAPRO RESEARCH, DELRAN NJ 08075 USA

All About Protocol Conversion Systems



The Local Data Datalynx/3174, first available in January 1987, is an asynchronous ASCII to SNA/SDLC protocol converter that transmits data up to 19.2K bps.

Just as a group of people conversing have to agree to sets of rules or limits in language and flow of speech to assure a level of mutual understanding, so do devices within networks or between networks need a common set of rules. In the data communications world, a rule that sets procedures for establishing and controlling transmissions is called a protocol.

In this report, we discuss protocol conversion systems, which include a wide variety of devices, such as code and interface converters, protocol converters, terminal controllers and emulators, packet assembler/disassemblers (PADs), gateways, and network processors. Using the Open Systems Interconnection (OSI) seven-layer model for data communications as a reference point, we discuss the various types of conversions that can take place in a network. A discussion of the mechanics of protocol conversion in the "real world" follows. Also included are a discussion of IBM's importance in the protocol conversion marketplace, a review of current trends in the conversion market, and recommendations for selecting conversion products.

Following the textual portion of this report are three groups of comparison columns, listing device specifications for protocol conversion systems, X.25 packet assembler/disassemblers (PADs), and a sampling of interface, code, speed, and async/sync converters. For your convenience, we have listed the names and addresses of vendors whose equipment is represented in the comparison columns at the end of this report.

Today's data communications networks generally fall into two broad categories: older networks and newer, technologically advanced ones; but each device in these two types of networks has its own communications protocols. Many organizations having networks with older terminals are reluctant to move towards newer network architectures because of the high cost of replacing their installed base of terminals. A significant barrier in the use of older devices with newer devices is incompatibility between communications protocols, an issue with little relevance to the user's applications, but a potentially insurmountable technical barrier. Protocol conversion is often the solution to the problem.

Simply stated, protocol conversion is the process of translating a protocol native to an end-user device, such as a terminal, into a different protocol to allow the end-user to communicate with another device that it normally would be incompatible with, such as a computer. Protocol conversion can be performed by a dedicated device, a software package loaded onto an existing system, or by a value-added network, like Telenet.

The complexity of a protocol is related to the complexity of the communications environment. Only very simple, character-by-character protocols are needed to effect a text exchange between human terminal operators, because human judgement can be applied at both the sending and receiving end of the communications link. However, when one of the users is a computer, or when speed and volume of information exchange makes character-by-character review impractical, more complex protocols are necessary.

In data communications, the solution to the problem of incompatibility lies in special hardware and software products that perform some type of conversion that translates the communications system of one device into that of another. Today, there are growing numbers and varieties of these products to handle many types of incompatibilities in the data communications network. These products range from microprocessor-based circuit boards to front-end processors with the ability to handle conversion functions through software applications programs. Available conversion devices may handle only one, or more than one, type of conversion. For example, some devices handle only code or interface conversions, while others handle protocol conversion, device emulation and code and interface translations.

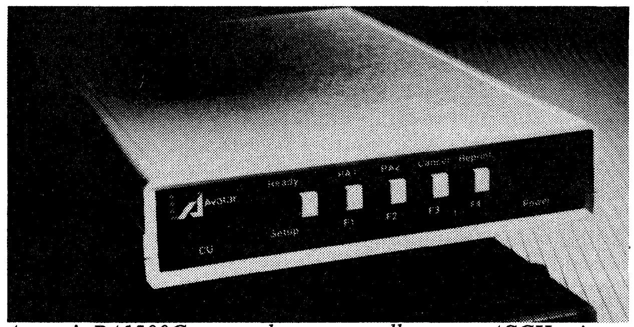
This report concentrates on hardware conversion devices, particularly "black box" protocol converters/emulators and terminal controllers that perform some type of conversion. We are aware that software packages for conversion and emulation are an extremely important part of the market; however, this reference service is primarily concerned with hardware. Readers interested in software conversion products should consult the *Datapro Directory of Software* and the *Datapro Directory of Microcomputer Software.*

For coverage of the micro-to-mainframe segment of the market, see Report C22-010-101, *An Overview of Microcomputer Communications*, in this volume.

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

In this report, we focus attention on the ways in which devices must be compatible in order to communicate. Using the Open Systems Interconnection (OSI) seven-layer model for data communications as a guide, we explain the various kinds of conversions that can take place between devices. We then discuss the mechanics of protocol conversion, the various products that handle particular conversions, and the ways in which conversions occur. The report also contains a discussion about conversion in the IBM 3270 environment, since solving problems of incompatibility between ASCII devices and IBM hosts is of particularly high interest to many readers. Also included are discussions about current trends in the conversion marketplace, including SNA to X.25 PADs, and some tips for selecting conversion products. At the end of the report is a list of vendors that provide various kinds of conversion products; their addresses and phone numbers are included.



Avatar's PA1500G protocol converter allows any ASCII printer to be used in standard IBM 3270 applications. It emulates an IBM 3287 or 3262 system printer when attached directly to any IBM control unit via Type A coax.

PROTOCOLS

The exchange of information is vital to users of data communications equipment, and the process of exchange is similar in many ways to human conversation. As stated previously, individuals in conversation must agree to sets of rules—protocols—in language and flow of speech in order to understand one another.

A protocol is a fixed set of rules that specifies the format of a data exchange. The rules govern the recognition of a connection with a remote point, the identification of the transmitting and receiving location, the transmission sequence, the handling of interruptions, methods of error checking and control, methods of blocking data, and security procedures.

Communications protocols cover a wide spectrum: they range from single character-by-character communications with no error checking to complex rules for moving large amounts of data among many devices.

In general, three major areas comprise a communications protocol:

- The method in which data is to be represented or encoded—the code set. Most data communications today use either the American Standard Code for Information Interchange (ASCII) or IBM's Extended Binary Coded Decimal Interchange Code (EBCDIC).

- The method in which the codes are transmitted and received—asynchronous or synchronous. In asynchronous transmission, data is sent with start and stop bits that encapsulate individual characters; data is sent at random intervals and does not require specific timing. In synchronous transmission, characters or bits are sent at a fixed rate, with the transmitting and receiving devices synchronized. Synchronous transmission eliminates the need for start/stop bits.

- The nondata exchanges of information by which the two devices establish control, detect failures or errors, and initiate corrective action. These sequences establish the context in which data can be exchanged.

The physical manifestation of the protocol is a series of characters in bit combinations that are appended to each block or frame of transmitted data. Through hardware or software, the sending device automatically formats the data and adds the required bits before transmitting each block. The receiving device automatically checks each of the appended bits before signalling an acknowledgement that data has been received. If any established condition is not met, the protocol initiates error control procedures.

Data communications protocols are either bit-oriented or byte-oriented. Byte-oriented protocols require that data be transmitted in eight-bit blocks; an acknowledgement is required after each transmitted block before the next block can be sent. Bit-oriented protocols allow data to be transmitted in blocks of any length up to a specified maximum; an acknowledgement may take place after one or several blocks have been sent, depending on the protocol. Some of the most common protocols are as follows:

- ASCII or TTY—an asynchronous protocol that uses the ASCII code set. Provides very little error checking. Transmission is in the form of a start bit, a number of data bits (usually five to eight), and one or more stop bits. Data in ASCII protocol can enter the communications line at any time; the end of the link is synchronized through the specifications of a common line speed and detection of the start bits and the beginning of the character transmission. ASCII requires an acknowledgement after each block is sent. ASCII protocol is often referred to as Teletype (TTY) protocol, since it is traditionally associated with teletypewriter equipment and services.

- IBM's SDLC (Synchronous Data Link Control)—a bit-oriented synchronous protocol that uses a synchronized series of frames. Each frame has a synchronization flag, followed by an address field, a control field that tells the purpose of the transmission, the data itself, then a frame-check field, and finally a trailing flag. The flag character is used to achieve the synchronization. SDLC permits up to 127 frames to be outstanding before an acknowledgement is required. Because SDLC works in full-duplex mode, a

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

user can send multiple blocks of data on one acknowledgement. SDLC is used in private-line networks.

- IBM BSC (Binary Synchronous Communications)—a character-oriented synchronous protocol that is also referred to as *bisync*. Binary synchronous data and control characters consist of eight-bit bytes. A transmission in BSC consists of a number of synchronizing (SYN) characters that ensure synchronization at both ends of the communications link. These are followed by a start-of-text (STX) character, an eight-bit block of text, an end-of-text (ETX) character, and a block error-checking character (BCC). BSC lacks the capability to handle full-duplex data, and does not comply with IBM's System Network Architecture (SNA) concept. Blocks of data can only be sent one at a time because each block must be acknowledged before the next can be sent. The BSC protocol works in multipoint applications over private lines.

Other communications protocols include HDLC (High-Level Data Link Control), a CCITT-specified, bit-oriented protocol upon which most other bit-oriented protocols are based; Univac U200, CDC UT200, and Burroughs Multipoint Poll Select, which are similar to IBM BSC but can run on both synchronous and asynchronous links; and Digital Equipment Corporation's DDCMP (Digital Data Communications Message Protocol), a byte-oriented protocol that can handle up to 255 unacknowledged transmissions.

Protocols are often application-dependent. This dependence, combined with the increasing importance of the computer and the increasing use of intelligent workstations, has resulted in a trend toward more complex protocols.

Typically, equipment manufacturers have viewed protocols in much the same way as they have viewed other products—they have introduced their own protocol rather than adopt that of a competitor. Many terminals in operation today use a vendor-established protocol; no industry-wide standard exists. Because of this, many terminals that perform the same functions cannot be used on the same system because they do not use the same protocols. For example, minicomputer users who have purchased asynchronous terminals from different vendors have discovered that even though the code set, speed, and transmission method are the same, communication with different terminals from the same computer port may not be possible. This is because each type of terminal has a set of commands or sequences of special characters that it recognizes and uses to perform functions such as cursor positioning and screen editing. Terminals of different manufacturers do not typically execute the same commands.

**ISO SEVEN-LAYER MODEL
FOR DATA COMMUNICATIONS**

(7) Application—provides communications services
(6) Presentation—defines syntax of data
(5) Session—controls data exchange
(4) Transport—handles data flow, error control
(3) Network—handles data routing
(2) Data Link—ensures data transfer via protocols
(1) Physical—provides mechanical/electrical interface

Figure 1. Layers One through Three define the interface between the host computer and the network. Layers Four through Seven provide compatibility to data format and exchange.

INCOMPATIBILITY IN DATA COMMUNICATIONS

We have said that data communications devices can be incompatible with one another in several ways. The International Standards Organization (ISO) Open Systems Interconnection reference model—a seven-layer hierarchy that defines the electrical characteristics, communications standards, and software applications for computer systems—provides a framework for understanding the ways in which devices differ. Each layer of the model defines a particular aspect of the entire data communications process. Refer to Figure 1 for a representation of the seven-layer hierarchy.

- Layer 1 is the *Physical Connection*, which provides mechanical and electrical specifications and procedures to establish, maintain, and end physical connections. This layer defines interface, code, speed, and synchronization functions. Interface, code, and asynchronous-to-synchronous converters fall into this category.

- Layer 2 is the *Data Link Control*, which insures that the data passes without error from one computer to another. This process involves protocols that specify the format for data transmission. Protocol converters are the devices that handle conversions in this layer. Parameters such as modem control, ring signalling, and dedicated connections fall into this category.

- Layer 3 is the *Network Layer*, which lets two systems exchange data. This layer defines packet addressing and data routing to final destination. Units that handle conversion in this layer include gateway devices, such as packet assemblers/disassemblers that provide access to X.25 networks or between local area networks. Front-end processors (FEPs) that include protocol conversion in their functions also fall into this classification.

- Layer 4 is the *Transport Layer*, which handles end-to-end error and flow control to ensure that the communications exchange is orderly and reliable. PAD devices, a type of gateway product, are the major products in this layer. Note that we classify PADs in both the Network and Transport layers.

- Layer 5 is the *Session Layer*, which provides the structure for a data exchange by managing connections between application processes, establishing and terminating connections, and sending end-to-end messages and controller dialogues. There are currently few conversion products in this category; Protocom's P2500 PAD device

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

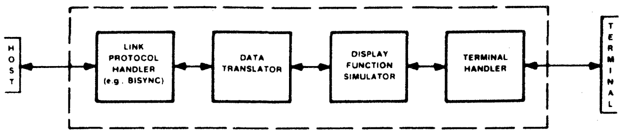


Figure 2. The protocol conversion process

handles conversion on the Session Layer, but is one of the few products that does so.

- Layer 6 is the *Presentation Layer*, which both defines the way in which data is put together and provides a systematic arrangement for the communications exchange to occur. This layer defines functions to translate coded data and convert it into display formats for terminal or microcomputer screens, printers, and other peripherals. In this layer, data is expanded or compressed and structured for file transfer or command translation. Devices called emulators, which allow one type of terminal to appear as another type of terminal, fall into the Presentation Layer category. Products in this category include ASCII-to-3270 emulators, interfaces that let personal computers act as 3270-type devices or access public networks, and word processor interfaces that handle conversions between dissimilar word processors.

- Layer 7 is the *Applications Layer*, which supports user and application tasks and provides the communications services that are available to specific computer applications. In essence, this layer provides the meaning to the message. Conversion devices that we discuss in this report do not provide conversions on this layer.

For devices to communicate with one another, they must be compatible on the interface, code, and protocol levels and must be alike according to link characteristics, device type, and device characteristics. Therefore, to connect incompatible equipment, converters must often provide translations on more than one of the levels in the network model. Conversion at one layer generally implies that compatibility in the layers below it in the model must also be accomplished. For example, a protocol converter working on Level 2 functions also assumes responsibility for compatibility in the interface, code, and synchronization functions.

Later in this report, we discuss the various products that handle conversion functions. These include interface, code, and asynchronous-to-synchronous converters; protocol converters; gateway devices, including PADs; protocol conversion in front-end processors; and terminal emulation/controllers and remote cluster controllers that let one device appear as another.

The Mechanics of Protocol Conversion

The earlier comparison of data communication and human conversation is useful in understanding the structure of protocol conversion. If two people speaking different languages wish to communicate, they may use a translator. The translator talks to each in the correct language and internally repackages the ideas for presentation in the correct form to the other party. A protocol converter performs a similar task; it sits on the communication path between the communicating devices and simulates the appropriate protocol for each. As Figure 2 shows, this gives protocol converters a distinctive double-ended structure. For each end of the conversion process, there is a local protocol handler that acts as a communicator and uses the protocol required by the attached device. Connecting these handlers is a gateway task that provides for the movement of user data between the handlers. If all communication protocols were structured in accordance with the OSI Reference Model, the converter would implement a set of seven-layer OSI protocols joined by the gateway task. Because the central task of a fully structured OSI protocol is the isolation of the user from the communication environment, a protocol converter dealing exclusively with full OSI-model protocols would be fairly simple to develop and would operate with few restrictions. With non-OSI protocols, such as those commonly in use in today's networks, the task of conversion may be complicated by the following issues:

- *The format of the user data.* If the data is easily separated from communication and device control protocols, it is more easily transferred to another environment. Use of special features, such as data compression, will normally complicate protocol conversion because such facilities do not necessarily exist in the other protocol.

- *The degree of layering in the protocols.* Even though full compliance with the OSI model is unlikely in the protocols being considered for conversion today, any amount of OSI-like layering in the protocols will aid in the separation of useful data from control information that must not be introduced into the other environment.

- *The availability of common functions in the protocols involved.* Data exchange between the users requires a degree of synchronization between the two foreign protocols. For example, most older protocols operate in half-duplex mode—only one station at a time may send information. It is necessary for converters operating between half-duplex protocols to insure that both stations are not given permission to send at the same moment, since neither could receive under those circumstances.

Where a protocol converter is used to allow a terminal of one type to simulate the operation of another type device, some form of device control protocol translation may be needed. IBM's popular 3270 series of terminals is often emulated using lower cost asynchronous devices, but the 3270 has special features, such as the ability to return only modified fields to the host computer. This ability must be emulated within the protocol converter, making converters of this type look almost like a small computer system. Figure 3 shows the structure of a terminal emulator protocol converter.

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

Interface, Code, and Asynchronous-to-Synchronous Converters

An interface is the physical connection between two devices. Interface conversion is the lowest level of established compatibility. Data and control lines from a device terminate in a connector that has pins that handle assigned signal functions. For example, the industry standard RS-232-C interface connector has 25 pins—one pin per function. The interface also prescribes voltage levels for electrical signals passing over the data and control lines.

Interface converters handle incompatibility between two interfaces. The devices link incompatible plugs, accept the connectors of two different interfaces, and/or translate the signals and voltage levels of one interface to that of another. Interface conversions commonly occur between RS-232-C and MIL-STD-188 or between RS-232-C and V.35. Several vendors, including Avanti Communications Corporation, Gandalf, and Datatel, offer products that handle many different types of conversions at the interface level.

Code converters handle the transformation of one communications code to another. A communications code is a bit pattern for each text, graphics, or control character. The most common data communications codes are ASCII, EBCDIC, and Baudot. An end-user device that operates using one of these codes cannot accept data in another code. In addition, all error-checking codes (e.g., parity) must be compatible. The conversion from one code to another may be simple, involving only the addition or deletion of control bits or the alteration of parity. A more complex code conversion might require transforming the data character's bit pattern.

Basic code conversion hardware consists of two universal synchronous/asynchronous receiver/transmitters (USARTs), a translation table contained in ROM, and some control circuitry. Characters received by the USART in one code are mapped in the ROM table into a corresponding character in the destination device's code. Converted data goes to the other USART, which transmits it to the destination device.

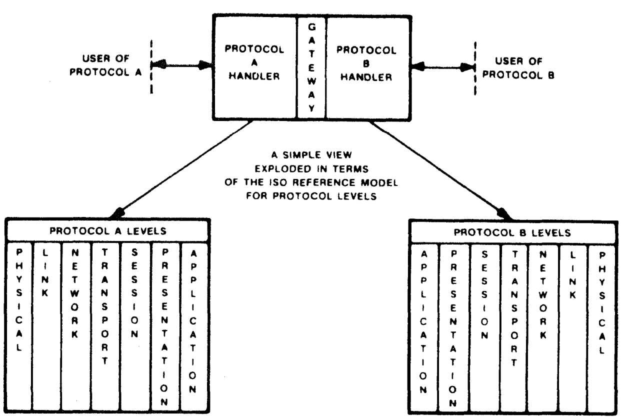


Figure 3. Inside a terminal emulator

One of the biggest problems of code conversion today is that of integrating word processors into data processing networks. Word processors typically have large character sets and control characters that are not used by data communications equipment. In some cases, the data communications device uses a word processor character for a different function. To integrate word processors into a data communications network, users must first convert the code of the word processor to a code that data communications equipment understands.

Placing word processors in data communications networks is difficult for other reasons. In many cases, the word processor manufacturer has developed a complete communications protocol for the equipment. Changing that protocol requires a higher level of conversion.

Asynchronous-to-synchronous converters are an older type of equipment, used mostly in applications that require conversion of asynchronous terminals for use on synchronous lines. In most newer conversion units, asynchronous-to-synchronous conversion is included along with other translation functions.

Protocol Converters

Protocol converters, one of the largest categories of conversion devices, handle changes that must occur on the Data Link Layer to ensure device compatibility. Protocol converters, or protocol conversion processors, as they are sometimes called, typically connect some type of incompatible peripheral device to a host. Protocol converters are microprocessor-based machines that usually communicate with the peripheral in a simple protocol and with the host in a more complex protocol that incorporates error-checking and retransmission capabilities. The converter communicates in the language of the peripheral and transforms and reformats data received from that peripheral before relaying it to the host, or the reverse, thus acting as an intermediary between the host and the peripheral. The peripheral to which the protocol converter attaches can be a terminal, a plotter, a microcomputer, a minicomputer, or another host.

A protocol converter actually changes one protocol to another by stripping the data down and rewrapping it according to the rules of a new set of specifications. Although hardware specifications differ from vendor to vendor, protocol converters usually contain a microprocessor, a realtime clock, two serial ports, associated data-rate generators, and the necessary firmware and RAM buffer.

During the conversion sequence, the protocol converter accepts blocks of data in one protocol, adds or deletes the necessary control characters, reformats the block, and calculates the required check characters so that the receiving device receives characters formatted according to its requirements. For example, in an ASCII-to-SDLC conversion, the converter will accept a string of characters, eliminate the start and stop bits, assemble the characters into a block, and add appropriate headers and trailers to create

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

complete frames. In a BSC-to-SDLC conversion, the converter must change the first four SYN bits of the Bisync algorithm to the first flag bit of the SDLC algorithm.

All protocol converters have some level of intermediate storage area to hold characters for conversion. Because of this buffering, a converter will always extend response time in the communications exchange. The device generally accepts low-speed input in the buffer, works with the data, and then transmits it out in short, high-speed bursts.

The data transmission method in the protocol converter differs from device to device. There are, however, some basic converter techniques. One of these techniques, called virtual protocol conversion, is used by a protocol converter that supports data transmissions up to 9600 bps. In virtual conversions, a central processor in the converter transforms each incoming datastream to its own protocol (the virtual protocol) and then reconverts the datastream to the protocol desired by the receiving device (the desired protocol).

An alternative technique uses a separate microprocessor to perform the conversion for each line interface that the device handles. The interface has approximately 12K of PROM in which a conversion program resides. Additional RAM (usually about 2K) holds the data from each line. A common memory module serves as a shared RAM buffer area, where input/output queuing takes place. Converted data goes to the shared area where it is transferred to the host in queue.

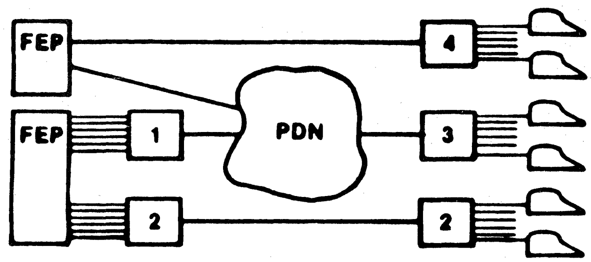
Besides pure protocol conversion, protocol converters often resolve related incompatibilities. For example, the converter might also translate ASCII code to EBCDIC or make several point-to-point links appear to the host as one multipoint link.

A special type of protocol converter is the Satellite Delay Compensation Unit (SDCU), which cuts propagation delay during satellite transmissions. Propagation delay is the amount of time between signal transmission over a circuit and acknowledgement of the transmission from the receiving end. Since the propagation delay during a satellite transmission is about a quarter of a second, this send-and-wait procedure can be quite time-consuming when every block requires acknowledgement before the next can be sent—as required by certain protocols, such as IBM BSC. The SDCU, which connects between the terminal and the modem, converts BSC into a specially conditioned form of SDLC that does not require an acknowledgement after each block. The end result is nearly 100 percent efficiency when transmitting in batch or message mode.

Gateways and PADs

These products handle conversions on OSI Layers Three and Four (the Network and Transport layers, respectively) and also perform lower layer functions.

Gateway devices are products that provide access between incompatible networks, for example, between SNA and DECnet, or between SNA and Ethernet, or between a data communications device and an X.25 public data network. Gateway products provide compatibility between network architectures' inherent protocols, codes, and interfaces. Gateway converters may link specific devices with one another like protocol converters do, or they may link two complete, but mutually exclusive systems, such as a minicomputer and an IBM mainframe, each with its own complement of peripherals. Since gateway devices are a logical subset of local area networks, we have included coverage of many of these products in Tab C11, Networks and Architectures, although readers will find some gateway products represented in the comparison charts that follow this report.



1 = Multi-PAD as host computer concentrator to network.
2 = Pair of PADs in statistical TDM configuration.
3 = Multi-PAD as terminal concentrator to network.
4 = Multi-PAD as terminal concentrator to host or front-end processor.

Figure 4. Typical Configurations for Dynatech's Multi-PAD.25.

By far the largest subset of gateway products are packet assembler/disassemblers (PADs). These devices permit host computers and peripheral equipment that use a communications protocol other than X.25 to be interconnected via a public data network. On the terminal side, most PADs support the connection of several devices, which can be terminals, CPU ports, printers, and so forth. On the network side, a high-speed port usually provides a link to the X.25 network. PADs usually perform concentrating and multiplexing functions as well as protocol conversion.

Most PAD products actually adapt a protocol rather than change it completely. The adaptation allows data in one protocol to pass through a network that uses another protocol. The transmitting PAD receives messages from the host or peripheral in the protocol of the sending device, converts and packetizes the information according to X.25 standards, and sends the packet through the network. At the receiving end of the X.25 link, another PAD performs error checking, disassembles the packets, and converts messages back to the native protocol. Some PADs can also perform true protocol conversion between the sending device and the destination device, when necessary.

In normal operations, the use of the PAD and the X.25 network are transparent to both the sending and the receiving devices. However, for test purposes, the PAD can be made to poll and to present status information to the host. Some PADs also have a supervisory port so that users can

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

configure the PAD's operating parameters and even diagnose network problems through the PAD.

In Figure 4, we see a typical set of configurations for Dynatech's Multi-PAD.25. As the diagram shows, users can configure PADs to work as concentrators for the host computer, as statistical time division multiplexers, as terminal concentrators for the public data network, and as terminal concentrators to a host or FEP.

One of the trends in gateway conversion is interconnection between incompatible systems and peripherals through a PBX. For example, InteCom's IBX provides conversion between ASCII and 3270 protocols, and Rolm's CBX provides a gateway to IBM networks. Interfacing to X.25 networks and compatibility with specified local area networks (e.g., Ethernet) are also sometimes supported. Other PBX vendors are now including gateway conversion functions in their products.

Conversion can also occur in host-independent network processors. These devices usually rely on a microprocessor-based architecture to perform multiple functions. They can often work as an X.25 packet processor to allow ASCII terminals to communicate with a host through X.25 networks, and many allow different hosts and workstations to communicate with one another in a network. The protocol translation capabilities of these devices let users configure networks that typically include products from various vendors, including IBM, Burroughs, and Digital Equipment Corporation.

Communications processors cannot be specifically classified as converters because they handle several other high-level functions in a data communications network. These products do not exist primarily to provide conversion functions. For more information on these devices, consult Tab C13 of Volume 1 of *Datapro Reports on Data Communications*

Some vendors include protocol conversion functions on their minicomputers. Data General, for example, provides an architecture for its Eclipse systems to handle extensive protocol conversions. Other vendors provide conversion software packages for their minicomputers.

At present, very few vendors offer products that handle conversion on the Session Layer. Protocom Devices does provide a P2500 PAD that supports Session Layer conversions to provide network security, simultaneous dual sessions, operation in Data Streaming/Turbo Mode, and error handling. The P2500 protects an organization from unauthorized network access via random password generation and permits only authorized terminal-connected PADs to access preassigned host-connected PADs. P2500 also permits some connected terminals to engage more than one host at a time. Turbo Mode operation on the P2500 decreases queuing delays that occur during transmission of large messages. The P2500 uses an inter-PAD block-check sequence, local end-to-end acknowledgements, and data retransmission to provide efficient error-handling functions.

Emulation Devices

Devices that handle conversions on the Presentation Layer provide the capability for one device to appear as another device. While protocol converters handle incompatibility problems between the sets of rules that particular devices use to communicate information, an emulator must handle incompatibilities in all specification differences between sending and receiving units—including differences in protocol, code, interface, device characteristics, and link characteristics. To the emulator, protocol conversion is secondary; the protocol converter actually strips down data and rewraps it according to a new set of rules, the emulator reads the text in a whole message and emulates that text to the specifications of a different device.

A great many protocol converters on the market today provide both protocol conversion and emulation. Often vendors call protocol/emulation products protocol converters, although this nomenclature is somewhat inaccurate. All emulation devices provide protocol conversion, but not all protocol converters provide emulation. Most often, however, devices that handle protocol and emulation translations are called value-added terminal controllers, remote cluster controllers, or terminal emulators.

To use information in a transmission, a receiving device—whether a host or a terminal—must interpret data in the

Method System's PCT-100 (Programmable Communications Translator) is a user-configurable in-line RS-232-C protocol and data translator. It allows software/hardware interfaces to be made compatible and can provide system-level enhancements.

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

context of the device that it supports. Device specifications impose many constraints on the data communications protocol that the device handles. This means that although a host and a terminal might operate in the same protocol, they might not be compatible with one another.

The unit that connects device-incompatible equipment must reformat data to offset restrictions imposed by an emulated device. Restrictions can include differences in record size and blocking characteristics, or they might relate to functional differences between equipment types. Most terminal emulators are not general-purpose units: they convert only between specific types of devices.

The way a terminal emulator handles conversions depends upon the specific characteristics of the emulated and emulating devices. Thus, describing a general emulation technique is difficult. But an example of how a terminal emulator takes an asynchronous datastream and converts it to the protocol and format used by an IBM 3271 terminal controller illustrates a basic conversion sequence.

An IBM 3271 serves up to 32 IBM 3277-type terminals on a multipoint line. Data moving in this type of configuration is blocked out in 1920-character screen images (blocks of data). If a user wants to replace IBM 3277 terminals with asynchronous ASCII devices, the ASCII units must appear as IBM 3277s to the IBM host. A terminal controller/emulator, or terminal controller as it is often called, can handle this problem by taking an asynchronous datastream into its buffer and keeping it there until a 1920-character screen image is filled or until the emulator receives an end-of-record, end-of-block control character. The terminal controller converts the protocol of the ASCII terminal to the protocol of the host (i.e., BSC), rearranges the data format to appear as if it comes from an IBM 3271, and then transfers the screen image to the host, which recognizes the data as that of an IBM 3277—not an asynchronous ASCII terminal. The terminal controller performs all functions of the device it replaces, including data concentration, poll/select, flow control, buffering, error detection and correction, and interfacing of multiple attached terminals. For example, Icot's Virtual Terminal controllers emulate an IBM 3271 or 3274 controller and provide ASCII terminal-to-IBM 3277/3278/3279 terminal emulation and IBM 3284 printer emulation.

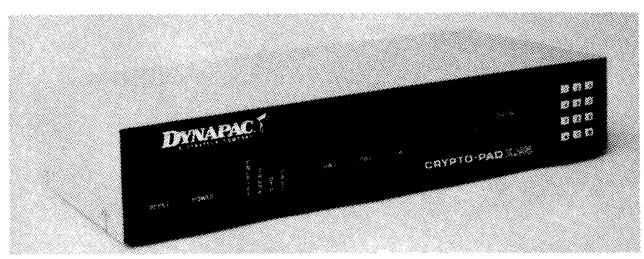
Sometimes the emulating device connects to an IBM cluster controller rather than replacing it, in effect, performing the conversion between the terminal and the IBM controller instead of between the controller and the host. The purpose of these emulators is to allow the user to integrate incompatible equipment into an existing terminal cluster. Local Data's Interlynx, for example, attaches to the IBM 3274 or 3276 controller to provide protocol and emulation translations that allow ASCII terminals to replace IBM 3278 or 3279 terminals.

During an emulation/conversion/transfer sequence, the emulator must interpret control sequences from an attached terminal to simulate the operations of the emulated terminal. The equivalents for a specified control sequence between one terminal model and another model vary widely. For example, no asychronous ASCII keyboard provides all of the special 3270 function keys, and those that are provided are generally encoded differently by different devices. Functions like erasing a screen, setting cursor address, and so forth are also encoded differently. As commands arrive, the emulator must translate the sequence and operate upon it according to the equivalent function of the emulated device. The emulator unit then updates its internal buffers and the display screen of the attached terminal according to the control sequence it receives and translates.

One of the biggest problems users face when using terminal emulation products concerns the special keystrokes an operator must learn to produce capabilities not normally supported on a particular terminal. Terminal operators accustomed to the keystrokes of a particular terminal must learn a new set of keystrokes to effect the functions of the emulated terminal. This operation can be compared to typing in Arabic on a typewriter with an English keyboard and an Arabic font. (Type a "g" and another symbol appears on the paper.) Because this kind of operation can cause confusion, vendors usually provide key maps that show keystroke equivalents between the emulated terminal and the various emulating devices. Some vendors also provide stick-on decals for emulating keyboards.

Many users are purchasing these terminal controllers to allow non-IBM devices in remote locations to access IBM mainframes. Remote cluster controllers eliminate the need to dedicate one terminal (e.g., a 3270) to one application, and another terminal at the same site to a different application. Many remote controllers have one synchronous line for 3270 access and two or more minicomputer interfaces. Terminals attached to the controller can switch between a remote host mainframe and the remote and local minicomputers in this type of configuration.

Users can configure most terminal controllers for dial-up access, allowing ASCII terminals in a remote location to dial into the local controller, which then makes the connection with a CPU that is located at the same or a third site. The controller eliminates the need for an IBM controller and additional synchronous lines to access the mainframe. A prominent cluster controller vendor, Lee Data/Data-



The DYNAPAC Crypto-PAD X.25 is a Packet Assembler/Disassembler with worldwide network certification. The device supports CCITT Recommendations X.25 (all levels), X.3, and X.29.

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

stream Networking Division, offers several models, including the 774 and the 776. The Model 776 operates in a point-to-point, multipoint, or switched BSC network and acts as, and replaces, an IBM 3271/3276 cluster controller.

Units that handle conversions to make microcomputers and personal computers compatible with IBM mainframes represent a large and growing area in the conversion/emulation marketplace. Organizations are using more and more microcomputers for decentralized applications, but in many instances microcomputer users must have access to a centralized database, which generally resides on an IBM mainframe. Users can establish a micro-to-mainframe link through an emulation package that typically includes a diskette containing the emulation logic and a communications circuit board that is installed inside the microcomputer. An example of this type of product is DCA's Irma, one of the most popular micro-to-mainframe interfaces. The Irma is an IBM PC board with a coaxial interface that connects the PC to an IBM 3270 terminal controller that accesses a mainframe. With Irma installed and running on the PC, users can download data from the mainframe to the microcomputer, where it is viewed on the microcomputer screen. Like other forms of emulation, micro-to-mainframe links usually specify the microcomputers supported and the host ports and/or peripherals to which they can be connected. The Irma, for example, must attach to an IBM Personal Computer or compatible microcomputer and will attach only to an IBM or compatible 3274/3276 terminal controller. Other emulators provide IBM 2780/3780 batch terminal emulation for specified micro- and minicomputers.

As networks have grown in complexity, incorporating both synchronous and asynchronous hosts and terminals, it has become necessary to provide several different types of conversions in one environment. While asynchronous ASCII-to-SNA SDLC is a popular conversion, other types of emulation products offer conversions that are the reverse. These units let an IBM 3270-type device talk to an ASCII host. Embracing a new philosophy concerning asynchronous equipment in an SNA world, IBM has recognized the importance of letting terminals in the network access both synchronous and asynchronous hosts. An ASCII passthrough feature on IBM's 3710 and 3708 controllers allows asynchronous devices to communicate with asynchronous hosts, while protocol conversion capabilities on both units provide ASCII-to-SNA SDLC conversion.

Conversion and emulation in a data communications network can occur in many different devices and at many points in a network. Converters can be separate hardware units placed between a terminal and a modem; shared hardware units that handle other functions (e.g., front-end processors); devices that replace cluster controllers; interface cards in a personal computer; or applications programs, specialized emulation software packages or software/hardware resident on minicomputers, mainframes, and PBX systems. Many network services, such as Tymnet and Telenet, also offer conversion as part of their value-added products.

The Amdahl 4460 X.25 Network Concentrator can be loaded with a single protocol or with mixes of protocols, including 3270 bisync or asynchronous.

Protocol conversion and emulation products address problems of incompatibility among many types of data communications devices. But as you might have surmised from our discussion above, the majority of conversion units are designed specifically to incorporate incompatible devices in an IBM environment. In the next section of this report, we will discuss that environment in relation to conversion and emulation products.

The IBM 327X Environment

Tremendous growth in the minicomputer, microcomputer, and personal computer markets has led to a rapid increase in the number of installed ASCII asynchronous terminals that access these computers. However, ASCII devices cannot access information that resides on IBM mainframes. IBM's series of products that provide interactive communications in an IBM network is the IBM 3270 Information Display System. This series includes controllers, terminals, and printers that are dedicated to a single host and usually to a single application.

Components in the current 3270 system include the 3278, 3279, 3178, 3179, 3180, and 3290 display terminals; the 3262, 3268, 3287, 4250, and 5210 printers; the 3274 and 3276 cluster controllers; and the 3270 Personal Computer. Each component comes in various models. For example, the 3278 is a monochromatic display available in five models that essentially differ only in their screen capacities. The 3279 is a color display version of the 3278. The 3274 controller comes in various models that handle up to 32 attached displays or printers, local or remote host connection, and BSC or SDLC protocol. The 3276 is a smaller controller designed for clusters of up to eight displays or printers.

Because of the 3270's huge installed base, many models that are no longer actively marketed by IBM continue to play a significant role in the IBM-compatible markets, particularly the 3277 display terminal, and the 3271 and 3272 controllers. The 3271 is a remote cluster controller that handles up to 32 displays or printers and comes in BSC and SDLC versions. The 3272 is a local channel-attached version of the 3271.

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

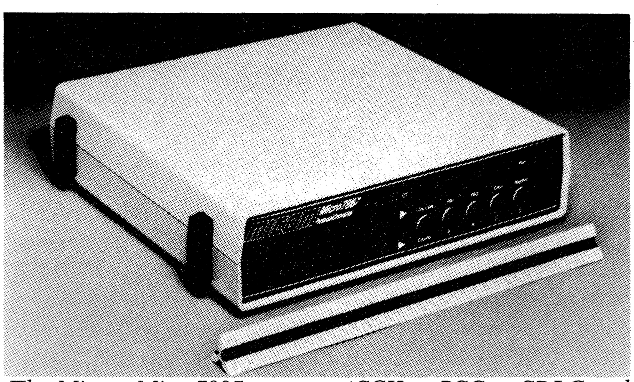
All About Protocol Conversion Systems

There are some shortcomings to using products in the 3270 family. First, they are more expensive than ASCII terminals. Second, many of the older IBM components are physically larger and take up more space than the ASCII terminals and the emulators that can be used in their place. (IBM has reduced both the price and size of its newer 3270 components, effectively eliminating these shortcomings.)

In 1979, IBM introduced the Model 3101 terminal that can attach directly to a 3705 communications controller and participate in ASCII applications resident in the host. With the introduction of the Model 3101, IBM acknowledged the need for asynchronous communication. The company has since introduced its second generation of asynchronous terminals, the 316X family.

IBM's first protocol converter, the 7426, introduced in October 1982, allowed the company's ASCII 3101 terminal to communicate with 8100 and 43XX computers. Although it was designed primarily for the 3101, the unit also enabled other asynchronous ASCII devices to connect to an SNA host. At the time the 7426 was introduced, interest in and sales of protocol conversion products had begun to increase dramatically, and several companies announced new converters that would allow asynchronous ASCII devices to emulate IBM 3270 equipment. From 1982 to 1984, revenues from protocol converter sales were strong, and IBM began making statements of direction concerning its intention to introduce more conversion products of its own.

In September 1984, IBM announced the 7171 protocol converter and the 3710 Network Controller. The 7171 allows the direct attachment of from 16 to 64 asynchronous ASCII devices to the block multiplexer channel of a 43XX or 308X host. Devices attached to the converter appear to the IBM host as 3270-type equipment. The 3710 offers the ability to manage mixed protocols (start/stop, BSC, and SDLC) in the network, as well as to multiplex and concentrate lines from attached devices to a 37X5 communications controller. One of the chief advantages of the 3710 is its ability to off-load a variety of SNA network management functions from the communications controller, thus freeing that device for other tasks. The 3710's line concentration function also allows users to save port space on the controller.



The Micom Micro7887 converts ASCII to BSC or SDLC and emulates IBM 3287/3289 printers.

IBM introduced the 3708 Network Conversion unit in 1985. The 3708 provides line concentration, protocol conversion, protocol enveloping, and ASCII passthrough support for asynchronous devices. The ten-port unit, designed for customer installation and maintenance, allows the attachment of one or two IBM hosts, asynchronous hosts, and asynchronous ASCII devices, which when attached to the unit emulate IBM 3270 equipment. The 3708 operates with IBM's System/370, 303X, 308X, 3090, 43XX, and 4700 processors; 8100 Systems; System 36 and 38 units; the 9370, the 3710 controller; and Rolm's CBX II voice/data PABX.

Competitors were quick to recognize the threat this new product posed. For a number of years, IBM had no protocol converters in its product line, and many manufacturers reaped the rewards of a strong market for devices that allowed asynchronous equipment to communicate with IBM hosts. Of course, a large part of this market included IBM mainframe customers. Now IBM has converters of its own to sell to its huge installed base, and the competition is forced to react to this formidable challenger.

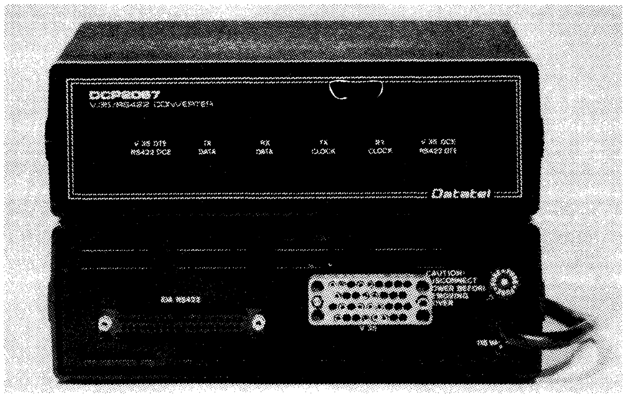
Most importantly, however, IBM introduced 3270 emulation support for most of its mini- and microprocessor-based products including the IBM PC, the System 34/36/38, the 8100, and the 43XX. IBM also introduced the 3270-PC, a version of the PC that is designed specifically for use in a 3270 system. In doing so, IBM, in effect, changed the 3270 from a single-host, dedicated terminal system to a system that can accommodate many different devices.

Although the majority of protocol converters and terminal controllers on the market today handle some type of conversion between ASCII devices and IBM units, other products handle conversion between IBM BSC protocols and IBM SDLC protocols. This conversion is particularly useful to users of older IBM BSC equipment who wish to migrate to an SNA/SDLC environment without replacing all of their old equipment. BSC-to-SDLC conversions generally occur between BSC 2780/3780 RJE or 3270 BSC protocols and SDLC protocols.

As IBM PCs have become prevalent in organizations, products to provide micro-to-mainframe compatibility are essential. Units that make ASCII terminals and personal computers compatible with SNA/SDLC networks are still in demand, despite a general downturn in this market, due in part to the introduction of IBM's protocol conversion products. ASCII devices provide flexible and inexpensive solutions to network problems, but IBM's mainframes are still the de facto standard for centralized computer facilities that must handle large databases and many applications. It seems unlikely that this situation will change soon, and vendors offering conversion products to handle ASCII-to-IBM conversions should continue to enjoy an adequate market share for their products, although it will be necessary for them to diversify their product lines in order to remain in the market.

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems



The DCP 2067 is part of Datatel's DCP family of interface converters that provide connection and conversion from modems to terminals and vice versa.

CURRENT TRENDS

Many different products handle some type of conversion to provide compatibility between communications devices. A number of large data communications equipment vendors, including IBM, have incorporated protocol converters and terminal controllers into their general line of products.

From an historical perspective, we can benchmark interest in protocol conversion at IBM's introduction of its 7426 converter in October 1982. With this announcement, IBM not only sanctioned conversion technology as a viable solution to network problems, but also focused industry attention on the technology.

Formerly the venue of small companies like Protocol Computers, Inc. and Innovative Electronics, which specialize in standalone protocol converter products, protocol conversion has become incorporated into existing data communications products, such as modems. We now find conversion is an integral capability in digital data switches, PBXs, personal computers, and word processors, as well.

The day of the protocol conversion chip is already here. Currently still virgin territory, this market has tremendous growth potential. Observers of the market predict that the concept of an integral X.25 PAD will eventually replace the current, separate device. For example, a modem may have a serial port, a parallel port, and an X.25 port.

Despite the new, integral technology, the standalone protocol converter, ideal for upgrading older networks, is alive and well and will continue to be during the next few years.

It is true, however, that growth in the entire protocol converter market is not as hearty as it was at the end of the last decade. As large-scale integration and personal computers began entering the market, the demand for a unit dedicated solely to protocol conversion waned. IBM's 1986 release of the 3710 controller, 3708 processor, and 3174 cluster controller, a device that connects ASCII terminals directly to the controller, also hurt the black box market. Users, however, have benefited from the subsequent price reductions on protocol converters.

Conversion products that facilitate LAN-to-LAN compatibility and access X.25 public data networks are also expected to have a large market. We have seen a growing interest in PAD devices that affect X.25 access, and we can anticipate greater PBX conversion capabilities in the months ahead. Conversion offerings from value-added carriers, such as Tymnet and Telenet, and from the Bell Operating Companies (BOCs), will also grow as data communications moves farther into the home markets where personal computer users are becoming more interested in linking into public networks and databases.

The market for SNA-to-X.25 conversion is growing. Users of 3270s want to communicate with packet networks for several reasons: to gain dial-up access to multiple resources, to save money through sharing bandwidth, and to have one IBM network management system for IBM as well as non-IBM equipment. Presently, SNA-to-X.25 conversion is available from two major sources: leading public data network providers and equipment vendors that offer 3270 SNA/SDLC PADs. These include Dynatech, Amdahl, and BBN Communications. The two prevailing approaches to SNA-to-X.25 conversion are the "pipe" method (presently the most common) and the value-added method. In the pipe technique, a transmission medium for SDLC information (the pipe) passes higher level SNA protocols transparently through the network. (Other protocols are being transported through the data link layer.) In the value-added approach, emulation is used to perform SNA-to-X.25 conversion, and PADs at the terminal and host are required. All polling takes place between the terminal PAD and the cluster controller, as it does in the pipe approach. But in addition to emulating SNA session activation protocols to establish the SDLC link the terminal PAD in a value-added set up also emulates System Service Control Point (SSCP) functions that are normally performed in the IBM host. Therefore, the terminal PAD also emulates protocols that activate IBM Physical Units (PUs) and Logical Units (LUs). This allows additional features, such as the ability to handle both BSC and SNA; to access by individual LUs to multiple hosts and applications (eliminating cross-domain SNA session establishment); network expansion through preconfiguration of devices; the ability to restrict access to a host or application; and the ability to pass call requests from a busy or inactive port to another available port.

Although less commonly used, the value-added approach provides far greater functionality, and it is certain that more vendors will adopt this technique as users demand it.

A problem with the SNA-to-X.25 technology is that synchronous and bisynchronous 3270 terminals transmit data in blocks instead of characters. As a result, delays are inevitable during the conversion process. The acuteness of the delays is dependent on the application.

Until the data communications industry adapts and uses worldwide protocol standards to link equipment, protocol

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

converters and emulators will remain an important part of the general market. It is unlikely that such standardization will occur in the very near future. What is certain is the integration of protocol conversion into a wide variety of equipment. This development will have a negative effect on the black box converter market, which will steadily diminish in the years ahead.

CHOOSING CONVERSION DEVICES

Before choosing a conversion unit, users should consider some of the negative characteristics of the devices. First, protocol converters will cause delays in response time on the network because data must flow into the converter's buffer before transmission. If data backs up in the buffer, overruns occur; if the buffer is small, the converter can lose data.

Terminal operators dealing with devices that emulate other products may have problems learning the new key sequences and key functions necessary to the emulation process. Thus, organizations can expect some decreased productivity during the initial months of a conversion installation.

In addition, protocol converters usually do not offer the security provided by, for example, the IBM 3270-type devices. Organizations must deal with the problem of protecting sensitive data, particularly in dial-up applications.

When an organization decides to install conversion products in a data communications network, it should determine exactly what kinds of conversions are needed to solve particular incompatibilities; for example, a new mainframe is installed and the organization wants to use existing terminals, or the organization has purchased microcomputers and micro-to-mainframe connections are now required. Once the application is established, users should determine which kind of products can handle the conversion most effectively in a particular application. This can be an extremely confusing task because there are so many conversion products available. To narrow choices, it is wise to contact many vendors and ask for product specifications and documentation that explain how a product operates. When studying specifications and operating procedures, users must note exactly what types of terminals, controllers, or hosts are supported by the device because most converters and controllers support specific products rather than a general range of devices. For example, a protocol converter specifically designed for IBM 3277 emulation might not work with a 3278 application.

Also important is finding out what added features and functions the converter handles. Does it support more than one host? Does it replace an IBM controller, or is it used in conjunction with a controller? Does the device incorporate any multiplexing or concentrating? Is the device user-reconfigurable (e.g., transmission speed, parity)? Can the network manager monitor the network via the converter? If additional features are available, are they standard or optional? What cost savings will it represent in your overall networking scheme?

Other important considerations include availabilty, reliability, and service. What is the delivery lead time? Can the customer install the device? What training is required? Is the device available for lease or purchase? Are quantity discounts available? What is the average mean-time-between failures? Is the device serviced on- or off-site? What is the replacement policy?

After narrowing the choices to the products of several vendors, users should ask the company to provide an in-house demonstration of the product. A prospective buyer should also request a list of current users who will discuss their experiences with the product. These individuals can provide information about the advantages and disadvantages of the product, hardware reliability, and the type and quality of support provided by the vendor.

IBM mainframe users in particular should find out whether conversion equipment can be upgraded as IBM upgrades and changes its SNA architecture.

After further narrowing the selection to two or three vendors, users should request a free trial of the product. By using a converter in a particular application, prospective buyers can soon find out whether a product provides the desired compatibility in an efficient manner.

COMPARISON CHARTS

The charts that accompany this report present listings of the key characteristics of approximately 150 protocol converters or terminal emulators, 70 X.25 PADs, and 50 code, speed, interface, and async/sync converters. This information was supplied by the vendors during March of 1987. Datapro wishes to thank the participating vendors for their cooperation.

Datapro sent repeated requests for information to 104 firms known or believed to manufacture some type of hardware conversion device. *The absence of any company from the charts means that the company either failed to respond to our request by the survey deadline, was unknown to us, did not make a hardware conversion product, or chose not to be listed.*

Many companies presently manufacture conversion devices for microcomputer-to-mainframe communication. These products consist of a circuit board that is inserted into the microcomputer and accompanying software. These devices, predominanty designed to effect IBM PC-to-mainframe connections, comprise a large and growing segment of the conversion marketplace and are covered in a separate report in *Datapro Reports on Data Communications* behind the Microcomputer Communications (C22) tab in Volume 2.

For the reader's convenience, we have organized the comparison charts into three broad categories. **Conversion Sys-**

All About Protocol Conversion Systems

tems/Terminal Controllers can be a variety of devices, including protocol converters, terminal emulators, remote cluster controllers, and terminal controllers. Basically, devices in this section provide conversion from one protocol to another and/or allow one device, e.g., an asynchronous ASCII terminal, to act as another type of device, e.g., an IBM 3270 terminal, in a network. **X.25 Packet Assemblers/ Disassemblers (PADS)** convert equipment using a non-X.25 protocol to the X.25 protocol for transmission over public and private networks. PADs also perform other functions, including concentrating or multiplexing. **Code**, **Speed**, **Interface**, and **Async/Sync Converters** include a number of miscellaneous devices that handle conversions from one code, interface, speed, or synchronization to another. These are generally less sophisticated devices than those represented in the other two categories.

The following text briefly describes the chart entries in the order in which they appear in the charts. Not all of the characteristics listed appear in all of the charts because some entries do not apply to all categories. For example, "link-level framing support" is relevant only to X.25 PADs.

General Characteristics

Manufacturer and model. We list here the manufacturer and exact model number or name of each device.

Device type. In the Conversion Systems/Terminal Controllers section, we have asked the manufacturer to specify whether the device is a protocol converter, terminal emulator, code or interface converter, and so forth.

Conversion performed. All converters perform some type of translation from one code, speed, or protocol to another. The most common conversion is asynchronous ASCII to IBM SNA/SDLC or BSC, but a number of other translations are available on the units represented in the tables.

Specific device emulated. In many cases, conversion devices provide the means to convert the text format of one type of device into the characteristics and format of another device. This translation, called emulation, is indicated, if available. Note that most protocol converters also provide device emulation.

Specific functionality provided. Most converters allow one device to be used as another type of device in the network. For example, a number of units allow asynchronous ASCII CRTs to be used as IBM 3277 Models 1 and 2. It is very important for prospective users to know exactly what type of device can be emulated by a given converter because many converters are designed for a specific application.

Virtual screen sizes supported. For a device to provide emulation, it must support the screen size, in characters, of the emulated device. For example, a device emulating an IBM 3270 terminal must support a 1,920-character screen.

Command port support. Some converters support a port through which users can select operating parameters and monitor, diagnose, and control the network. A "yes" answer indicates that the device does have a command port.

Network certification. X.25 PADs allow devices to interface with public and private networks. These networks must certify use of the device on their facilities. Prominent network providers include GTE Telenet, Tymnet, and Uninet in the U.S., and Datapac and Infoswitch in Canada.

CCITT recommendation supported. CCITT has specified certain protocols for devices operating on an X.25 network. Most X.25 PADs support all CCITT recommendations, including X.25 Levels 1, 2, 3; X.3; X.28; and X.29. X.25 defines the operating protocol of the packet network. X.3 defines a set of 18 parameters that govern operation of the PAD and each asynchronous terminal port. X.28 defines the interface between the asynchronous terminal and the PAD, and X.29 defines the format of the packets that carry control information from the PAD to the remote destination.

Buffer memory capacity. PAD devices contain a buffer memory that holds packets before transmission. Software is generally held in ROM.

Additional RAM available. Many PADs can accommodate the additional RAM necessary to expand the capacity of the device.

Device configuration. Most X.25 PADs can be configured to act as other types of devices in the network. Typical configurations include host concentration, time division multiplexing, or front-end processing.

Host Side/Network Channel Specifications

Specific hosts supported. Conversion devices generally support IBM or compatible hosts or asynchronous hosts such as Digital's VAX or PDP 11. In this entry, we have listed the type of mainframe with which the converter operates.

Number and type of host lines supported. Most converters support one line to a host; that line can be one IBM SNA/ SDLC or BSC line, or one line to an asynchronous host. Some devices do support more than one host connection. This connection may be a dual IBM BSC or SDLC link, or one IBM link and one link to an asynchronous host.

Number of host sessions supported concurrently. If a converter supports more than one host line, the device may support both connected hosts concurrently, or separately through a switch selection.

Connections supported. Conversion devices support direct connections, multipoint, and/or point-to-point connections. Most converters support more than one type of connection, and most support all three.

Connection to host via controller. While some conversion devices emulate a controller, others must connect to a controller in the network. We have asked the vendors to

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

specify the type of controller to which the converter interfaces, if applicable.

Number of X.25 channels supported. Here the vendor has specified the number of channels a PAD supports for connection to the public or private data network.

Number of virtual circuits supported. A virtual circuit is the logical connection between the input port and the destination port. When a terminal connects to a PAD, the PAD automatically or manually establishes a circuit to the destination by selecting an unused logical channel number and transmitting a Call Request packet that uses that logical channel number. This request packet identifies the input port and the destination port and carries other information used to set up the logical connection. In this entry, vendors have specified the maximum number of virtual circuits supported by the PAD.

Maximum window size, in frames. Window size describes the maximum number of unacknowledged frames (packets) that can be handled by the PAD at one time. Generally, PADs support up to seven frames in the window. When the PAD's transmitter reaches the maximum window size, it blocks the window. In effect, window framing is a form of flow control.

Link-level framing supported. At the link level, blocks of data are assembled according to certain framing protocols. These include character- or bit-oriented framing (BSC or HDLC, respectively). This is an EBCDIC or ASCII option on character-oriented (BSC) framing.

Terminal Side/Input Specifications

Number of types of ports (or devices) supported. In general, a conversion device supports asynchronous ports that accommodate a large variety of asynchronous ASCII printers, terminals, and personal computers. Many converters also support a dynamic printer port. Although more uncommon, conversion devices and PADs do support synchronous ports, or asynchronous and synchronous ports in combination. Devices represented in the charts support from one to many input devices; a number of units accommodate input-port expansion in specified increments.

Specific devices supported. It is important to know whether the unit supports a particular type of terminal. In today's market, most conversion devices designed for asynchronous ASCII to IBM SDLC or BSC conversion support virtually any asynchronous ASCII device. Some converters, however, are designed for operation only with a specific terminal. In this entry, the vendors have noted the manufacturer and model number of devices supported. An answer of "virtually any device" means that the vendor's list of supported terminals was too long to fit into the assigned space, but the converter did support all major asynchronous ASCII terminals and/or personal computers available in today's market.

Autospeed/autoparity available. Many X.25 PADs automatically adjust to the transmission speed and parity of the inputting DTE. A "yes" answer indicates that the PAD supports this feature.

Channel configuration data downline loadable. X.25 PADs may support this feature, which allows terminal operators to configure channel parameters from the terminal and download those configurations to the PAD.

Ports configurable for permanent or switched circuits. Some PADs will allow users to configure an input port for permanent or switched virtual circuit connection through the network. In cases where the circuit is switched, the termination of a logical connection signals that the connection is free and can be used by another port. When the virtual circuit is permanent, the connection is dedicated to one port only. Many PADs support both permanent and switched virtual circuits on a selectable basis.

Transmission Specifications

Maximum transmission, in bps. This entry indicates the maximum speed of operation or data rate supported by the device stated in bits per second.

Maximum aggregate input rate, in bps. Conversion devices generally support many input ports, each operating at several different speeds, e.g., from 50 to 9600 bps. Aggregate input refers to the maximum data rate accepted from all channels simultaneously. For example, if there are four channels operating at a maximum 9600 bps rate per channel, the aggregate input rate could be four times 9600, or 38.4K bps.

Synchronization. This refers to the time relationship among the bits that make up the characters, which make up the messages. Conversion devices handle data in spurts (asynchronous) or continuous streams (synchronous).

Transmission mode. Most converters operate in either half- or full-duplex mode, or both. Half-duplex mode permits data transmission in either direction, but not simultaneously. Full-duplex operation implies that the data is simultaneously transmitted and received over a common communications facility. Simplex mode permits unidirectional data transmission, whereby data is either transmitted or received.

Protocols supported. Protocols are a set of rules that establish and control transmission. There are two basic types of protocols: byte-oriented (IBM's BSC or Digital's DDCMP) or bit-oriented (IBM SNA/SDLC or ISO HDLC). Converters usually translate one protocol to another and thus support different protocols on the terminal and host sides.

Codes supported. Codes consist of specific sets of characters that can be alphanumeric, graphic, and control characters. Control characters initiate, modify, or halt an action that effects data transmission. The most common data communications codes are ASCII, used in the asynchro-

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

nous protocol, and EBCDIC, the usual code generated by synchronous devices.

Interface. Interface is the electrical connection between components. Most communications devices provide an electrical interface (RS-232-C) in accordance with the standards established by the Electronics Industries Association (EIA). Several other interface standards exist, notably CCITT Recommendation V.24 and V.28.

Clocking. Clocking refers to the repetitive, regularly timed signals used to control synchronous transmission. Clocking may be established internally by the device itself, externally by another device (for example, a modem), or be derived from the datastream.

Diagnostics. Many conversion devices perform tests that check the device and the line connections. Most converters conduct a self-test of internal circuitry upon power-up and provide front-panel LEDs to monitor system status.

Features (on X.25 PADs)

Channel priority assignment. Some PADs allow users to assign priority to incoming channels. Whenever the priority channel requests a connection to the network, that channel receives immediate access to the PAD and "bumps" channels with less priority.

Password protection. On many devices, users must enter a password to gain access to the PAD. This feature prevents unauthorized access to the network.

Supervisory port. Through this port, users can monitor and control the network and send messages throughout the system.

Echoplex. This feature refers to the printing of keyboarded characters on return of the signal from the other end of the line, using full-duplex transmission, to assure that the data was received correctly at the other end.

Autodialer support. Autodialers allow users to set dialing parameters, such as delay specifications for dial tone and call answering and switch-to-switch delay pauses, in memory. When this feature is present, dialing and disconnecting calls occur automatically.

Pricing and Availability

Purchase price. This is the basic price of the unit, excluding any options, except where noted in the Comments.

Rental. The monthly charge for leasing the unit from the vendor or a third party is shown in this entry.

Installation. When vendors charge a fee for installing the unit, we have included the onetime charge. Note that most conversion devices are designed for customer installation.

Maintenance. Many vendors charge an annual fee to service the unit and provide ongoing maintenance.

Serviced by. In this entry, we list the provider of service on the unit. Usually, the vendor offers service on an on-site or factory repair/return basis. In some cases, a third party provides service.

Availability. Here we list the current lead time on orders, given in days after receipt of order (ARO).

Date of first commercial delivery. Here we provide the date when the vendor first delivered the product to the marketplace.

Number installed to date. Some vendors list the approximate number of installed units as of March 1987. Note that in some cases, the vendor combines this figure for all models installed.

Comments. In this section, we have listed various special characteristics pertaining to a particular device. These might include additional capabilities, features, software, as well as information regarding related products offered by the vendor.

VENDORS

Listed below are the names, headquarters addresses, and telephone numbers of vendors who manufacture conversion devices. We have provided this list so that readers can contact the vendors for more information about the products they offer.

Advanced Computer Communications, 720 Santa Barbara Street, Santa Barbara, CA 93101. Telephone (805) 963-8801.

Agile, 825 Alfred Nobel Drive, Hercules, CA 94547. Telephone (415) 825-9220.

Altertext, 210 Lincoln Street, Boston, MA 02111. Telephone (617) 426-0009.

Amdahl Communications, 2200 North Greenville, Richardson, TX 75081. Telephone (214) 699-9500.

Arkansas Systems, Incorporated, 8901 Kanis Road, Little Rock, AK 72205. Telephone (501) 227-8471.

Astrocom Corporation, 120 West Plato Boulevard, St. Paul, MN 55107. Telephone (612) 227-8651.

Atlantic Research Corporation, 7401 Boston Boulevard, Springfield, VA 22153. Telephone (703) 644-9190.

Avanti Communications Corporation, Aquidneck Industrial Park, Newport, RI 02840. Telephone (401) 849-4660.

Avatar Technologies Incorporated, 99 South Street, Hopkinton, MA 01748. Telephone (617) 435-6872.

BBN Communications Corporation, 70 Fawcett Street, Cambridge, MA 02238. Telephone (617) 497-2800.

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

Black Box Corporation, Mayview Road at Park Drive, Box 12800, Pittsburgh, PA 15241. Telephone (412) 746-5500.

Bridge Communications, 2081 Stierlin Road, Mountain View, CA 94043. Telephone (415) 969-4400.

Cableshare Incorporated, P.O. Box 5880, 20 Enterprise Drive, London, Ontario, Canada N6A 4L6. Telephone (519) 686-2900.

Codex Corporation, 7 Blue Hill River Road, Maresfield Farm, Canton, MA 02021-1097. Telephone (617) 364-2000.

ComData Corporation, 7900 North Nagle, Morton Grove, IL 60053. Telephone (312) 470-9600.

Com/Tech Systems, 505 Eighth Avenue, New York, NY 10018. Telephone (212) 594-5377.

ComDesign, Incorporated, 751 South Kellogg Avenue, Santa Barbara, CA 93117. Telephone (805) 964-9852.

Commercial Data Processing, Incorporated, 2241 South Grand Boulevard, St. Louis, MO 63104. Telephone (314) 776-1130.

Commtex Incorporated, 1655 Crofton Boulevard, Crofton, MD 21114. Telephone (301) 721-3666.

Computer Communications, Incorporated, 2610 Columbia Street, Torrance, CA 90503. Telephone (213) 320-9101.

Computer Peripheral Systems, Incorporated, Box 98282, Atlanta, GA 30359. Telephone (404) 292-9565.

Comstat Datacomm Corporation, 1351 Oakbrook Drive, Suite 165, Norcross, GA 30093. Telephone (404) 446-9496.

CTi Data Corporation, 5275 North Boulevard, Raleigh, NC 27604. Telephone (919) 876-8731.

Datagraf, Incorporated, 8305 Highway 71 West, Austin, TX 78735. Telephone (512) 288-0453.

Datagram Corporation, 11 Main Street, East Greenwich, RI 02818. Telephone (401) 885-4840 or (800) 235-5030.

Datamaxx USA Corporation, P.O. Box 6477, 1815 South Gadsden Street, Tallahasse, FL 32314. Telephone (904) 224-8213.

Dataprobe, 110 West Palisades Boulevard, Palisades Park, NJ 07650. Telephone (201) 947-9500.

Datatel, Incorporated, Pin Oak and Springdale Road, Cherry Hill, NJ 08003. Telephone (609) 424-4451.

DCC Corporation, 7300 North Crescent Boulevard, Pennsauken, NJ 08110. Telephone (609) 662-7272.

Digital Communications Associates, 1000 Alderman Drive, Alpharetta, GA 30201. Telephone (404) 442-4000.

Digital Controls Corporation, 3495 Newmark Drive, Miamisburg, OH 45342. Telephone (513) 435-5455.

Diversified Data Resources, Incorporated, 25 Mitchell Boulevard, Suite 7, San Rafael, CA 94903. Telephone (415) 499-8870.

Dynatech Packet Technology, 6464 General Green Way, Alexandria, VA 22312. Telephone (703) 642-9391.

EDA Instruments Incorporated, 4 Thorncliffe Park Drive, Toronto, Ontario, Canada M4H 1H1. Telephone (416) 425-7800.

Forest Computer, 1749 Hamilton Road, P.O. Box 509, Okemos, MI 48864. Telephone (517) 349-4700.

Gandalf Data, Incorporated, 1019 Noel Avenue, Wheeling, IL 60090. Telephone (312) 541-6060.

General Datacomm Industries, Route 63, Middlebury, CT 06762-1299. Telephone (203) 574-1118.

GTE Telenet Communications Corp., 12490 Sunrise Valley Drive, Reston, VA 22096. Telephone (703) 689-6000.

INCAA Datacom B.V., Amerfoortseweg 15, 7313 AB Apeldoorn-Holland. Telephone (31) (0)55-551262.

Infinet, Incorporated, 40 High Street, North Andover, MA 01845. Telephone (617) 681-0600.

Infotron Systems Corporation, 9 North Olney Avenue, Cherry Hill, NJ 08003-1688. Telephone (609) 424-9400 or (800) 257-8352.

Innovative Electronics, Incorporated, 4714 Northwest 165 Street, Miami, FL 33014. Telephone (305) 624-1644

Instrumentation Services Incorporated (ISI), 957 Winnetka Avenue North, Minneapolis, MN 55427. Telephone (612) 544-8916.

International Business Machines Corporation, Old Orchard Road, Armonk, NY 10504. Contact your local IBM representative

JBM Electronics, 6020 North Lindbergh Boulevard, Hazelwood, MO 63042. Telephone (314) 731-7781.

KMW Systems Corporation, 100 Shepherd Mountain Plaza, Austin, TX 78730-5014. Telephone (512) 338-3000.

Lee Data/Datastream Networking Division, 2520 Mission College Boulevard, Santa Clara, CA 95050. Telephone (408) 986-8022.

Local Data, 2771 Toledo Street, Torrance, CA 90503. Telephone (213) 320-7126.

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

▷ **May-Craft Information Systems**, 4312 Beltwood Parkway South, Dallas, TX 75244. Telephone (214) 392-3766.

Memotec Data, Incorporated, 600 McCaffrey, Montreal, Quebec, Canada H4T 1N1. Telephone (514) 738-4781.

Method Systems Incorporated, 3511 Lost Nation Road, #202, Willoughby, OH 44094. Telephone (800) 533-6116.

Micom Systems, Incorporated, 4100 Los Angeles Avenue, Simi Valley, CA 93062. Telephone (805) 583-8600.

Modemsplus Incorporated, 217 East Trinity Place, P.O. Box 1727, Decatur, GA 30031. Telephone (404) 378-5276.

NCR Comten, 2700 Snelling Avenue North, St. Paul, MN 55113. Telephone (612) 638-7944.

Netlink Incorporated, 3214 Spring Forest Road, Raleigh, NC 27604. Telephone (919) 878-8612.

Nu Data Incorporated, 32 Fairview Avenue, Little Silver, NJ 07739. Telephone (201) 842-5757.

PCI, 26630 Agoura Road, Calabasas, CA 91302-1988. Telephone (818) 880-4900.

Perle Systems Incorporated, 1980 Springer Drive, Lombard, IL 60148. Telephone (312) 932-4171.

Protocom Devices, Incorporated, 1660 Bathgate Avenue, Bronx, NY 10457-8102. Telephone (212) 716-5400.

Quasitronics, Incorporated, 211 Vandale Drive, Houston, PA 15342. Telephone (412) 745-2663.

Renex Corporation, 1513 Davis Ford Road, Woodbridge, VA 22192. Telephone (703) 494-2200.

Shaffstall Corporation, 7901 East 88th Street, Indianapolis, IN 46256. Telephone (317) 842-2077.

Tekelec, 26540 Agoura Road, Calabasas, CA 91302. Telephone (818) 880-5656.

Telebyte Technology Incorporated, 270 East Pulaski Road, Greenlawn, NY 11740. Telephone (516) 423-3232.

Teleprocessing Products Incorporated, 4565 East Industrial Street, Suite 7-K, Simi Valley, CA 93063. Telephone (805) 522-8147.

Thomas Engineering, 2440 Stanwell Drive, Concord, CA 94520. Telephone (415) 680-8640.

Trax Softworks, Incorporated, 10801 National Boulevard, Los Angeles, CA 90064. Telephone (213) 475-8729.

Tri-Data Corporation, 505 East Middlefield Road, Mountain View, CA 94043. Telephone (415) 969-3700.

Tymnet/McDonnell Douglas, 2710 Orchard Parkway, San Jose, CA 95134. Telephone (408) 435-7744.

VIR Incorporated, 105 James Way, Southampton, PA 18966. Telephone (215) 364-8866.

Wall Data Incorporated, 17769 N.E. 78th Place, Redmond, WA 98052-4992. Telephone (206) 883-4777.

Western DataComm Company, 5083 Market Street, Youngstown, OH 44512. Telephone (216) 788-6583.

Western Digital Corporation, 2445 McCabe Way, Irvine, CA 92714. Telephone (714) 863-0102. □

© 1987 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED—FOR REPRINTS, CALL 1-800-328-2776

All About Protocol Conversion Systems

The 1980s have seen a dramatic acceleration in the rate of technological change, coupled with an economic climate that requires careful analysis of capital investment. Many organizations with older terminals have hesitated to move to newer network architectures because of the high cost of replacing their terminal populations. A significant barrier in the use of older devices with newer networks is incompatibility between communications protocols, an issue with little relevance to the user's applications, but a potentially insurmountable technical barrier. Protocol conversion is an obvious solution to the problem.

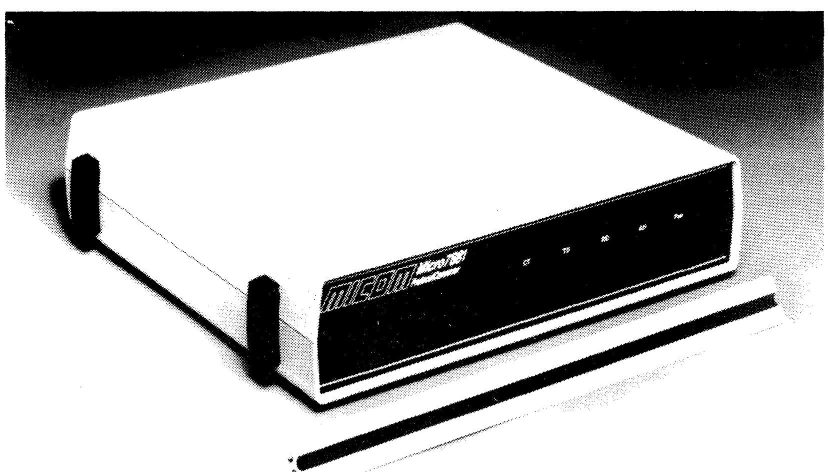
This report discusses protocol conversion systems, which include a wide variety of devices, such as code and interface converters, protocol converters, terminal controllers and emulators, packet assembler/disassemblers (PADs), gateways, and network processors. Using the OSI seven-layer model for data communications as a reference point, we discuss the various types of conversions that can take place in a network. A discussion of the mechanics of protocol conversion "in the real world" follows. Also included are a discussion of IBM's importance in the protocol conversion marketplace, a review of current trends in the conversion market, and recommendations for selecting conversion products. We also present the 1986 LAN/Terminal Users Survey results that pertain to protocol conversion systems.

Following the textual portion of this report are three groups of comparison columns, listing device specifications for protocol conversion systems, X.25 packet assembler/disassemblers, and a sampling of interface, code, speed, and async/sync converters. For your convenience, we have listed the names and addresses of vendors whose equipment is represented in the comparison columns at the end of this report.

Parties in a conversation must agree to a set of rules or limits in language and flow of speech to assure some level of mutual understanding. The collection of these rules is called a protocol. The complexity of a protocol is related to the complexity of the communications environment. Only very simple, character-by-character protocols are needed to effect a text exchange between human terminal operators because human judgement can be applied at both the sending and receiving end of the communications link. However, when one of the users is a computer, or when speed and volume of information exchange makes character-by-character review impractical, more complex protocols are necessary.

Data communications network designers can achieve flexibility and economic rewards by using products from more than one vendor. However, hardware and software from different vendors are seldom compatible with one another. They rarely speak the exact same language. If a user wants to design a network of diverse hardware and software products, he or she must deal with the problem of incompatibility. In doing so, the user must first understand exactly how various devices are incompatible with one another and then determine the most effective way to deal with the differences.

In data communications, the solution to the problem of incompatibility lies in special hardware and software products that perform some type of conversion that translates the communications system of one device into that of another. Today, there are growing numbers and varieties of these products to handle many types of incompatibilities in the data communications network. These products range from microprocessor-based circuit boards to front-end processors with the ability to handle conversion functions through software applications programs. Available conversion devices may handle only one, or more than one, type of conversion. For example, some devices handle only code or interface conversions, while others handle protocol conversion, device emulation, as well as code and interface translations.



The Micom Micro7881 is a compact display converter that allows asynchronous display terminals or personal computers to be used with IBM 3270 Information Display Systems.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

ISO SEVEN-LAYER MODEL FOR DATA COMMUNICATIONS

(7) Application—provides communications services
(6) Presentation—defines syntax of data
(5) Session—controls data exchange
(4) Transport—handles data flow, error control
(3) Network—handles data routing
(2) Data Link—ensures data transfer via protocols
(1) Physical—provides mechanical/electrical interface

Figure 1. Layers One through Three define the interface between the host computer and the network. Layers Four through Seven provide compatibility to data format and exchange.

This report concentrates on hardware conversion devices, particularly "black box" protocol converters/emulators and terminal controllers that perform some type of conversion. We are aware that software packages for conversion and emulation are an extremely important part of the market; however, this reference service is primarily concerned with hardware. Readers interested in software conversion products should consult the *Datapro Directory of Software* and the *Datapro Directory of Microcomputer Software.*

For coverage of the micro-to-mainframe segment of the market, see Report C22-010-101, *An Overview of Microcomputer Communications,* in this volume.

In this report, we focus attention on the ways in which devices must be compatible in order to communicate. Using the Open Systems Interconnection (OSI) seven-layer model for data communications as a guide, we explain the various kinds of conversions that can take place between devices. We then discuss the mechanics of protocol conversion, the various products that handle particular conversions, and the ways in which conversions occur. The report also contains a discussion about conversion in the IBM 3270 environment, since solving problems of incompatibility between ASCII devices and IBM hosts is of particularly high interest to many readers. Also included are discussions about current trends in the conversion marketplace, some tips for selecting conversion products, and the results of a section of our 1986 LAN/Terminal User's Survey concerning protocol conversion. At the end of the report is a list of vendors that provide various kinds of conversion products; their addresses and phone numbers are included.

PROTOCOLS

The exchange of information is vital to users of data communications equipment, and the process of exchange is similar in many ways to human conversation. As stated previously, individuals in conversation must agree to sets of rules—protocols—in language and flow of speech in order to understand one another.

A protocol is a fixed set of rules that specifies the format of a data exchange. The rules govern the recognition of a connection with a remote point, the identification of the transmitting and receiving location, the transmission sequence, the handling of interruptions, methods of error checking and control, methods of blocking data, and security procedures.

Communications protocols cover a wide spectrum: they range from single character-by-character communications with no error checking to complex rules for moving large amounts of data among many devices.

In general, three major areas comprise a communications protocol:

- The method in which data is to be represented or encoded—the code set. Most data communications today use either the American Standard Code for Information Interchange (ASCII) or IBM's Extended Binary Coded Decimal Interchange Code (EBCDIC).
- The method in which the codes are transmitted and received—asynchronous or synchronous. In asynchronous transmission, data is sent with start and stop bits that encapsulate individual characters; data is sent at random intervals and does not require specific timing. In synchronous transmission, characters or bits are sent at a fixed rate, with the transmitting and receiving devices synchronized. Synchronous transmission eliminates the need for start/stop bits.
- The nondata exchanges of information by which the two devices establish control, detect failures or errors, and initiate corrective action. These sequences establish the context in which data can be exchanged.

The physical manifestation of the protocol is a series of characters in bit combinations that are appended to each block or frame of transmitted data. Through hardware or software, the sending device automatically formats the data and adds the required bits before transmitting each block. The receiving device automatically checks each of the appended bits before signalling an acknowledgement that data has been received. If any established condition is not met, the protocol initiates error control procedures.

Data communications protocols are either bit-oriented or byte-oriented. Byte-oriented protocols require that data be transmitted in eight-bit blocks; an acknowledgement is required after each transmitted block before the next block can be sent. Bit-oriented protocols allow data to be transmitted in blocks of any length up to a specified maximum; an acknowledgement may take place after one or several blocks have been sent, depending on the protocol. Some of the most common protocols are as follows:

- ASCII—an asynchronous protocol that uses the ASCII code set. Provides very little error checking. Transmis-

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

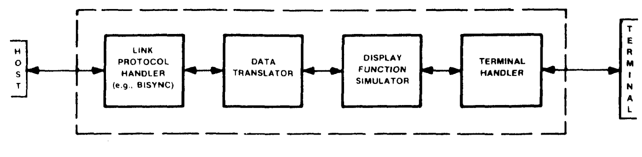


Figure 2. The protocol conversion process

sion is in the form of a start bit, a number of data bits (usually five to eight), and one or more stop bits. Data in ASCII protocol can enter the communications line at any time; the end of the link is synchronized through the specifications of a common line speed and detection of the start bits and the beginning of the character transmission. ASCII requires an acknowledgement after each block is sent. ASCII protocol is often referred to as Teletype (TTY) protocol, since it is traditionally associated with teletypewriter equipment and services.

- IBM's SDLC—a bit-oriented protocol that uses a synchronized series of frames. Each frame has a synchronization flag, followed by an address field, a control field that tells the purpose of the transmission, the data itself, then a frame-check field, and finally a trailing flag. The flag character is used to achieve the synchronization. SDLC permits up to 127 frames to be outstanding before an acknowledgement is required.

- IBM BSC—a character-oriented protocol. Binary synchronous data and control characters consist of eight-bit bytes. A transmission in BSC consists of a number of synchronizing (SYN) characters that ensure synchronization at both ends of the communications link. These are followed by a start-of-text (STX) character, an eight-bit block of text, an end-of-text (ETX) character, and a block error-checking character (BCC). BSC lacks the capability to handle full-duplex data, and does not comply with IBM's System Network Architecture (SNA) concept. Each block must be acknowledged before the next can be sent.

Other communications protocols include HDLC (High-Level Data Link Control), a bit-oriented protocol; Univac U200, CDC UT200, and Burroughs Multipoint Poll Select, which are similar to IBM BSC but can run on both synchronous and asynchronous links; and Digital Equipment Corporation's DDCMP (Digital Data Communications Message Protocol), a byte-oriented protocol that can handle up to 255 unacknowledged transmissions.

Protocols are often application-dependent. This dependence, combined with the increasing importance of the computer and the increasing use of intelligent workstations, has resulted in a trend toward more complex protocols.

Typically, equipment manufacturers have viewed protocols in much the same way as they have viewed other products—they have introduced their own protocol rather than adopt that of a competitor. Many terminals in operation today use a vendor-established protocol; no industry-wide standard exists. Because of this, many terminals that perform the same functions cannot be used on the same system because they do not use the same protocols. For example, minicomputer users who have purchased asynchronous terminals from different vendors have discovered that even though the code set, speed, and transmission method are the same, communication with different terminals from the same computer port may not be possible. This is because each type of terminal has a set of commands or sequences of special characters that it recognizes and uses to perform functions such as cursor positioning and screen editing. Terminals of different manufacturers do not typically execute the same commands.

INCOMPATIBILITY IN DATA COMMUNICATIONS

We have said that data communications devices can be incompatible with one another in several ways. The International Standards Organization (ISO) Open Systems Interconnection reference model—a seven-layer hierarchy that defines the electrical characteristics, communications standards, and software applications for computer systems—provides a framework for understanding the ways in which devices differ. Each layer of the model defines a particular aspect of the entire data communications process. Refer to Figure 1 for a representation of the seven-layer hierarchy.

- Layer 1 is the *Physical Layer*, which provides mechanical and electrical specifications and procedures to establish, maintain, and end physical connections. This layer defines interface, code, speed, and synchronization functions. Interface, code, and asynchronous-to-synchronous converters fall into this category.

- Layer 2 is the *Data Link Layer*, which insures that the data passes without error from one computer to another. This process involves protocols that specify the format for data transmission. Protocol converters are the devices that handle conversions in this layer.

- Layer 3 is the *Network Layer*, which lets two systems exchange data. This layer defines packet addressing and data routing to final destination. Units that handle con-

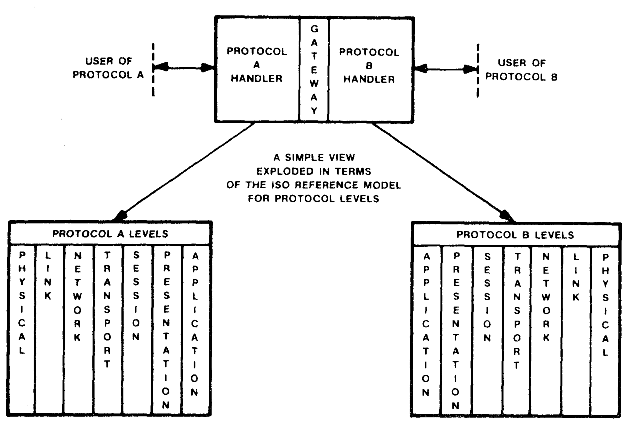


Figure 3. Inside a terminal emulator

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

version in this layer include gateway devices, such as packet assemblers/disassemblers that provide access to X.25 networks or between local area networks. Front-end processors (FEPs) that include protocol conversion in their functions also fall into this classification.

- Layer 4 is the *Transport Layer*, which handles end-to-end error and flow control to ensure that the communications exchange is orderly and reliable. PAD devices, a type of gateway product, are the major products in this layer. Note that we classify PADs in both the Network and Transport layers.

- Layer 5 is the *Session Layer*, which provides the structure for a data exchange by managing connections between application processes, establishing and terminating connections, and sending end-to-end messages and controller dialogues. There are currently few conversion products in this category; ProtoComm's P2500 PAD device handles conversion on the Session Layer, but is one of the few products that does so.

- Layer 6 is the *Presentation Layer*, which both defines the way in which data is put together and provides a systematic arrangement for the communications exchange to occur. This layer defines functions to translate coded data and convert it into display formats for terminal or microcomputer screens, printers, and other peripherals. In this layer, data is expanded or compressed and structured for file transfer or command translation. Devices called emulators, which allow one type of terminal to appear as another type of terminal, fall into the Presentation Layer category. Products in this category include ASCII-to-3270 emulators, interfaces that let personal computers act as 3270-type devices or access public networks, and word processor interfaces that handle conversions between dissimilar word processors.

- Layer 7 is the *Applications Layer*, which supports user and application tasks and provides the communications services that are available to specific computer applications. In essence, this layer provides the meaning to the message. Conversion devices that we discuss in this report do not provide conversions on this layer.

For devices to communicate with one another, they must be compatible on the interface, code, and protocol levels and must be alike according to link characteristics, device type, and device characteristics. Therefore, to connect incompatible equipment, converters must often provide translations on more than one of the levels in the network model. Conversion at one layer generally implies that compatibility in the layers below it in the model must also be accomplished. For example, a protocol converter working on Level 2 functions also assumes responsibility for compatibility in the interface, code, and synchronization functions.

Later in this report, we discuss the various products that handle conversion functions. These include interface, code, and asynchronous-to-synchronous converters; protocol converters; gateway devices, including PADs; protocol conversion in front-end processors; and terminal emulation/controllers and remote cluster controllers that let one device appear as another.

The Mechanics of Protocol Conversion

The earlier comparison of data communication and human conversation is useful in understanding the structure of protocol conversion. If two people speaking different languages wish to communicate, they may use a translator. The translator talks to each in the correct language and internally repackages the ideas for presentation in the correct form to the other party. A protocol converter performs a similar task; it sits on the communication path between the communicating devices and simulates the appropriate protocol for each. As Figure 2 shows, this gives protocol converters a distinctive double-ended structure. For each end of the conversion process there is a local protocol handler that acts as a communicator and uses the protocol required by the attached device. Connecting these handlers is a gateway task that provides for the movement of user data between the handlers. If all communication protocols were structured in accordance with the OSI Reference Model, the converter would implement a set of seven-layer OSI protocols joined by the gateway task. Because the central task of a fully structured OSI protocol is the isolation of the user from the communication environment, a protocol converter dealing exclusively with full OSI-model protocols would be fairly simple to develop and would operate with few restrictions. With non-OSI protocols, such as those commonly in use in today's networks, the task of conversion may be complicated by the following issues:

- The format of the user data. If the data is easily separated from communication and device control protocols, it is more easily transferred to another environment. Use of special features such as data compression will normally complicate protocol conversion because such facilities do not necessarily exist in the other protocol.

- The degree of layering in the protocols. Even though full compliance with the OSI model is unlikely in the protocols being considered for conversion today, any amount of OSI-like layering in the protocols will aid in the separation of useful data from control information that must not be introduced into the other environment.

- The availability of common functions in the protocols involved. Data exchange between the users requires a degree of synchronization between the two foreign protocols. For example, most older protocols operate half-duplex—only one station at a time may send. It is necessary for converters operating between half-duplex protocols to insure that both stations are not given permission to send at the same moment, since neither could receive under those circumstances.

Where a protocol converter is used to allow a terminal of one type to simulate the operation of another type device, some form of device control protocol translation may be needed. IBM's popular 3270 series of terminals is often

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

emulated using lower cost asynchronous devices, but the 3270 has special features such as the ability to return only modified fields to the host computer. This ability must be emulated within the protocol converter, making converters of this type look almost like a small computer system. Figure 3 shows the structure of a terminal emulator protocol converter.

Interface, Code, and Asynchronous-to-Synchronous Converters

An interface is the physical connection between two devices; interface conversion is the lowest level of established compatibility. Data and control lines from a device terminate in a connector that has pins that handle assigned signal functions. For example, the industry standard RS-232-C interface connector has 25 pins—one pin per function. The interface also prescribes voltage levels for electrical signals passing over the data and control lines.

Interface converters handle incompatibility between two interfaces. The devices link incompatible plugs, accept the connectors of two different interfaces, and/or translate the signals and voltage levels of one interface to that of another. Interface conversions commonly occur between RS-232-C and MIL-STD-188 or between RS-232-C and V.35. Several vendors, including Avanti Communications Corporation, Gandalf, and Datatel, offer products that handle many different types of conversions at the interface level.

Code converters handle the transformation of one communications code to another. A communications code is a bit pattern for each text, graphics, or control character. The most common data communications codes are ASCII, EBCDIC, and Baudot. An end-user device that operates using one of these codes cannot accept data in another code. In addition, all error-checking codes (e.g., parity) must be compatible. The conversion from one code to another may be simple, involving only the addition or deletion of control bits or the alteration of parity. A more complex code conversion might require transforming the data character's bit pattern.

Basic code conversion hardware consists of two universal synchronous/asynchronous receiver/transmitters (USARTs), a translation table contained in ROM, and some control circuitry. Characters received by the USART in one code are mapped in the ROM table into a corresponding character in the destination device's code. Converted data goes to the other USART, which transmits it to the destination device.

One of the biggest problems of code conversion today is that of integrating word processors into data processing networks. Word processors typically have large character sets and control characters that are not used by data communications equipment. In some cases, the data communications device uses a word processor character for a different function. To integrate word processors into a data communications network, users must first convert the code of the word processor to a code that data communications equipment understands.

Placing word processors in data communications networks is difficult for other reasons. In many cases, the word processor manufacturer has developed a complete communications protocol for the equipment. Changing that protocol requires a higher level of conversion.

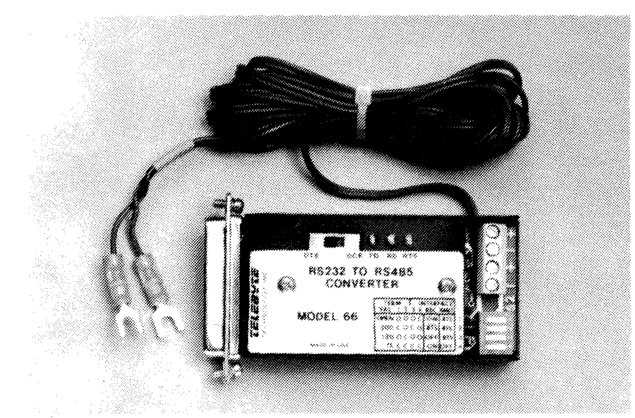
Asynchronous-to-synchronous converters are an older type of equipment, used mostly in applications that require conversion of asynchronous terminals for use on synchronous lines. In most newer conversion units, asynchronous-to-synchronous conversion is included along with other translation functions.

Protocol Converters

Protocol converters, one of the largest categories of conversion devices, handle changes that must occur on the Data Link Layer to ensure device compatibility. Protocol converters, or protocol conversion processors, as they are sometimes called, typically connect some type of incompatible peripheral device to a host. Protocol converters are microprocessor-based machines that usually communicate with the peripheral in a simple protocol and with the host in a more complex protocol that incorporates error-checking and retransmission capabilities. The converter communicates in the language of the peripheral and transforms and reformats data received from that peripheral before relaying it to the host, or the reverse, thus acting as an intermediary between the host and the peripheral. The peripheral to which the protocol converter attaches can be a terminal, a plotter, a microcomputer, a minicomputer, or another host.

A protocol converter actually changes one protocol to another by stripping the data down and rewrapping it according to the rules of a new set of specifications. Although hardware specifications differ from vendor to vendor, protocol converters usually contain a microprocessor, a realtime clock, two serial ports, associated data-rate generators, and the necessary firmware and RAM buffer.

During the conversion sequence, the protocol converter accepts blocks of data in one protocol, adds or deletes the necessary control characters, reformats the block, and calculates the required check characters so that the receiving



Telebyte's Model 66 interface converter provides RS-232-C to RS-485 conversion.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

device receives characters formatted according to its requirements. For example, in an ASCII-to-SDLC conversion, the converter will accept a string of characters, eliminate the start and stop bits, assemble the characters into a block, and add appropriate headers and trailers to create complete frames. In a BSC-to-SDLC conversion, the converter must change the first four SYN bits of the Bisync algorithm to the first flag bit of the SDLC algorithm.

All protocol converters have some level of intermediate storage area to hold characters for conversion. Because of this buffering, a converter will always extend response time in the communications exchange. The device generally accepts low-speed input in the buffer, works with the data, and then transmits it out in short, high-speed bursts.

The data transmission method in the protocol converter differs from device to device. There are, however, some basic converter techniques. One of these techniques, called virtual protocol conversion, is used by a protocol converter that supports data transmissions up to 9600 bps. In virtual conversions, a central processor in the converter transforms each incoming datastream to its own protocol (the virtual protocol) and then reconverts the datastream to the protocol desired by the receiving device (the desired protocol).

An alternative technique uses a separate microprocessor to perform the conversion for each line interface that the device handles. The interface has approximately 12K of PROM in which a conversion program resides. Additional RAM (usually about 2K) holds the data from each line. A common memory module serves as a shared RAM buffer area, where input/output queuing takes place. Converted data goes to the shared area where it is transferred to the host in queue.

Besides pure protocol conversion, protocol converters often resolve related incompatibilities. For example, the converter might also translate ASCII code to EBCDIC or make several point-to-point links appear to the host as one multipoint link.

A special type of protocol converter is the Satellite Delay Compensation Unit (SDCU), which cuts propagation delay during satellite transmissions. Propagation delay is the amount of time between signal transmission over a circuit and acknowledgement of the transmission from the receiving end. Since the propagation delay during a satellite transmission is about a quarter of a second, this send-and-wait procedure can be quite time-consuming when every block requires acknowledgement before the next can be sent—as required by certain protocols, such as IBM BSC. The SDCU, which connects between the terminal and the modem, converts BSC into a specially conditioned form of SDLC that does not require an acknowledgement after each block. The end result is nearly 100 percent efficiency when transmitting in batch or message mode.

Gateways and PADs

These products handle conversions on OSI Layers Three and Four (the Network and Transport layers, respectively) and perform lower layer functions.

Gateway devices are products that provide access between incompatible networks, for example, between SNA and DECnet, or between SNA and Ethernet, or between a data communications device and an X.25 public data network. Gateway products provide compatibility between network architectures' inherent protocols, codes, and interfaces. Gateway converters may link specific devices with one another like protocol converters do or they may link two complete, but mutually exclusive systems, such as a minicomputer and an IBM mainframe, each with its own complement of peripherals. Since gateway devices are a logical subset of local area networks, we have included coverage of many of these products in Tab C11, Networks and Architectures, although readers will find some gateway products represented in the comparison charts that follow this report.

By far the largest subset of gateway products are packet assembler/disassemblers (PADs). These devices permit host computers and peripheral equipment that use a communications protocol other than X.25 to be interconnected via a public data network. On the terminal side, most PADs support the connection of several devices, which can be terminals, CPU ports, printers, and so forth. On the network side, a high-speed port usually provides a link to the X.25 network. PADs usually perform concentrating and multiplexing functions as well as protocol conversion.

Most PAD products actually adapt a protocol rather than change it completely. The adaptation allows data in one protocol to pass through a network that uses another protocol. The transmitting PAD receives messages from the host or peripheral in the protocol of the sending device, converts and packetizes the information according to X.25 standards, and sends the packet through the network. At the receiving end of the X.25 link, another PAD performs error checking, disassembles the packets, and converts messages back to the native protocol. Some PADs can also perform true protocol conversion between the sending device and the destination device, when necessary.

In normal operations, the use of the PAD and the X.25 network are transparent to both the sending and the receiving devices. However, for test purposes, the PAD can be made to poll and to present status information to the host. Some PADs also have a supervisory port so that users can configure the PAD's operating parameters and even diagnose network problems through the PAD.

In Figure 4, we see a typical set of configurations for Dynatech's Multi-PAD.25. As the diagram shows, users can configure PADs to work as concentrators for the host computer, as statistical time division multiplexers, as terminal concentrators for the public data network, and as terminal concentrators to a host or FEP.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

One of the trends in gateway conversion is interconnection between incompatible systems and peripherals through a PBX. For example, InteCom's IBX provides conversion between ASCII and 3270 protocols, and Rolm's CBX provides a gateway to IBM networks. Interfacing to X.25 networks and compatibility with specified local area networks (e.g., Ethernet) are also sometimes supported. Other PBX vendors are now including gateway conversion functions in their products.

Conversion can also occur in host-independent network processors. These devices usually rely on a microprocessor-based architecture to perform multiple functions. They can often work as an X.25 packet processor to allow ASCII terminals to communicate with a host through X.25 networks, and many allow different hosts and workstations to communicate with one another in a network. The protocol translation capabilities of these devices let users configure networks that typically include products from various vendors, including IBM, Burroughs, and Digital Equipment Corporation.

These communications processors cannot be specifically classified as converters because they handle several other high-level functions in a data communications network. These products do not exist primarily to provide conversion functions. For more information on these devices, consult Tab C13 of Volume 1 of *Datapro Reports on Data Communications.*

Some vendors include protocol conversion functions on their minicomputers. Data General, for example, provides an architecture for its Eclipse systems to handle extensive protocol conversions. Other vendors provide conversion software packages for their minicomputers.

At present, very few vendors offer products that handle conversion on the Session Layer. Protocom Devices does provide a P2500 PAD that supports Session Layer conversions to provide network security, simultaneous dual sessions, operation in Data Streaming/Turbo Mode, and error handling. The P2500 protects an organization from unauthorized network access via random password generation and permits only authorized terminal-connected PADs to access preassigned host-connected PADs. P2500 also permits some connected terminals to engage more than one host at a time. Turbo Mode operation on the P2500 decreases queuing delays that occur during transmission of large messages. The P2500 uses an inter-PAD block-check sequence, local end-to-end acknowledgements, and data retransmission to provide efficient error-handling functions.

Emulation Devices

Devices that handle conversions on the Presentation Layer provide the capability for one device to appear as another device. While protocol converters handle incompatibility problems between the sets of rules that particular devices use to communicate information, an emulator must handle incompatibilities in all specification differences between sending and receiving units—including differences in protocol, code, interface, device characteristics, and link characteristics. To the emulator, protocol conversion is secondary; the protocol converter actually strips down data and rewraps it according to a new set of rules, the emulator reads the text in a whole message and emulates that text to the specifications of a different device.

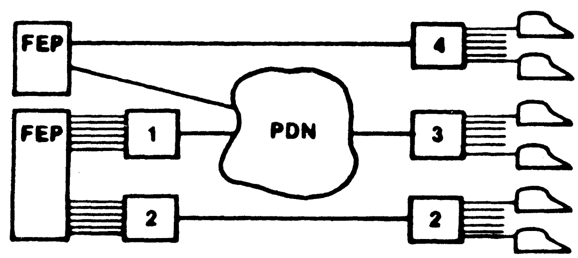
A great many protocol converters on the market today provide both protocol conversion and emulation. Often vendors call protocol/emulation products protocol converters, although this nomenclature is somewhat inaccurate. All emulation devices provide protocol conversion, but not all protocol converters provide emulation. Most often, however, devices that handle protocol and emulation translations are called value-added terminal controllers, remote cluster controllers, or terminal emulators.

To use information in a transmission, a receiving device—whether a host or a terminal—must interpret data in the context of the device that it supports. Device specifications impose many constraints on the data communications protocol that the device handles. This means that although a host and a terminal might operate in the same protocol, they might not be compatible with one another.

The unit that connects device-incompatible equipment must reformat data to offset restrictions imposed by an emulated device. Restrictions can include differences in record size and blocking characteristics, or they might relate to functional differences between equipment types. Most terminal emulators are not general-purpose units: they convert only between specific types of devices.

The way a terminal emulator handles conversions depends upon the specific characteristics of the emulated and emulating devices. Thus, describing a general emulation technique is difficult. But an example of how a terminal emulator takes an asynchronous datastream and converts it to the protocol and format used by an IBM 3271 terminal controller illustrates a basic conversion sequence.

An IBM 3271 serves up to 32 IBM 3277-type terminals on a multipoint line. Data moving in this type of configuration is blocked out in 1920-character screen images (blocks of data). If a user wants to replace IBM 3277 terminals with asynchronous ASCII devices, the ASCII units must appear



1 = Multi-PAD as host computer concentrator to network.
2 = Pair of PADs in statistical TDM configuration.
3 = Multi-PAD as terminal concentrator to network.
4 = Multi-PAD as terminal concentrator to host or front-end processor.

Figure 4. Typical Configurations for Dynatech's Multi-PAD.25.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

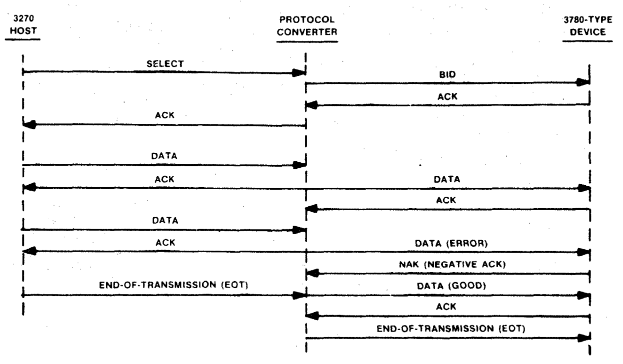


Figure 5. A complete sequence in protocol conversion

as IBM 3277s to the IBM host. A terminal controller/emulator, or terminal controller as it is often called, can handle this problem by taking an asynchronous datastream into its buffer and keeping it there until a 1920-character screen image is filled or until the emulator receives an end-of-record, end-of-block control character. The terminal controller converts the protocol of the ASCII terminal to the protocol of the host (i.e., BSC), rearranges the data format to appear as if it comes from an IBM 3271, and then transfers the screen image to the host, which recognizes the data as that of an IBM 3277—not an asynchronous ASCII terminal. The terminal controller performs all functions of the device it replaces, including data concentration, poll/select, flow control, buffering, error detection and correction, and interfacing of multiple attached terminals. For example, Icot's Virtual Terminal controllers emulate an IBM 3271 or 3274 controller and provide ASCII terminal-to-IBM 3277/3278/3279 terminal emulation and IBM 3284 printer emulation.

Sometimes the emulating device connects to an IBM cluster controller rather than replacing it, in effect, performing the conversion between the terminal and the IBM controller instead of between the controller and the host. The purpose of these emulators is to allow the user to integrate incompatible equipment into an existing terminal cluster. Local Data's Interlynx, for example, attaches to the IBM 3274 or 3276 controller to provide protocol and emulation translations that allow ASCII terminals to replace IBM 3278 or 3279 terminals.

During an emulation/conversion/transfer sequence, the emulator must interpret control sequences from an attached terminal to simulate the operations of the emulated terminal. The equivalents for a specified control sequence between one terminal model and another model vary widely. For example, no asychronous ASCII keyboard provides all of the special 3270 function keys, and those that are provided are generally encoded differently by different devices. Functions like erasing a screen, setting cursor address, and so forth are also encoded differently. As commands arrive, the emulator must translate the sequence and operate upon it according to the equivalent function of the emulated device. The emulator unit then updates its internal buffers and the display screen of the attached terminal according to the control sequence it receives and translates.

One of the biggest problems users face when using terminal emulation products concerns the special keystrokes an operator must learn to produce capabilities not normally supported on a particular terminal. Terminal operators accustomed to the keystrokes of a particular terminal must learn a new set of keystrokes to effect the functions of the emulated terminal. This operation can be compared to typing in Arabic on a typewriter with an English keyboard and an Arabic font. (Type a "g" and another symbol appears on the paper.) Because this kind of operation can cause confusion, vendors usually provide key maps that show keystroke equivalents between the emulated terminal and the various emulating devices. Some vendors also provide stick-on decals for emulating keyboards.

Many users are purchasing these terminal controllers to allow non-IBM devices in remote locations to access IBM mainframes. Remote cluster controllers eliminate the need to dedicate one terminal (e.g., a 3270) to one application, and another terminal at the same site to a different application. Many remote controllers have one synchronous line for 3270 access and two or more minicomputer interfaces. Terminals attached to the controller can switch between a remote host mainframe and the remote and local minicomputers in this type of configuration.

Users can configure most terminal controllers for dial-up access, allowing ASCII terminals in a remote location to dial into the local controller, which then makes the connection with a CPU that is located at the same or a third site. The controller eliminates the need for an IBM controller and additional synchronous lines to access the mainframe. A prominent cluster controller vendor, Datastream Communications, Inc., offers several models, including the 774 and the 776. The Model 776 operates in a point-to-point, multipoint, or switched BSC network and acts as, and replaces, an IBM 3271/3276 cluster controller.

Units that handle conversions to make microcomputers and personal computers compatible with IBM mainframes represent a large and growing area in the conversion/emulation marketplace. Organizations are using more and more microcomputers for decentralized applications, but in many instances microcomputer users must have access to a centralized database, which generally resides on an IBM mainframe. Users can establish a micro-to-mainframe link through an emulation package that typically includes a diskette containing the emulation logic and a communications circuit board that is installed inside the microcomputer. An example of this type of product is DCA's Irma, one of the most popular micro-to-mainframe interfaces. The Irma is an IBM PC board with a coaxial interface that connects the PC to an IBM 3270 terminal controller that accesses a mainframe. With Irma installed and running on the PC, users can download data from the mainframe to the microcomputer, where it is viewed on the microcomputer screen. Like other forms of emulation, micro-to-mainframe links usually specify the microcom-

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

puters supported and the host ports and/or peripherals to which they can be connected. The Irma, for example, must attach to an IBM Personal Computer or compatible microcomputer and will attach only to an IBM or compatible 3274/3276 terminal controller. Other emulators provide IBM 2780/3780 batch terminal emulation for specified micro- and minicomputers.

Other types of emulation products offer conversions that are the reverse of the popular ASCII-to-3270 conversion. For example, Protocol Computers' 74D unit lets an IBM 3270-type device talk to an ASCII host. The 74D interfaces with an IBM communications controller, an IBM 3270 cluster controller, and up to six ASCII hosts, which can be Digital hosts, public networks, and personal computers. Supported by the 74D, the 3270 can switch between SDLC and Digital hosts.

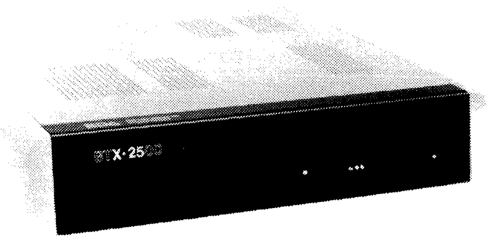
Conversion and emulation in a data communications network can occur in many different devices and at many points in a network. Converters can be separate hardware units placed between a terminal and a modem; shared hardware units that handle other functions (e.g., front-end processors); devices that replace cluster controllers; interface cards in a personal computer; or applications programs, specialized emulation software packages or software/hardware resident on minicomputers, mainframes, and PBX systems. Many network services, such as Tymnet and Telenet, also offer conversion as part of their value-added products.

Protocol conversion and emulation products address problems of incompatibility among many types of data communications devices. But as you might have surmised from our discussion above, the majority of conversion units are designed specifically to incorporate incompatible devices in an IBM environment. In the next section of this report, we will discuss that environment in relation to conversion and emulation products.

The IBM 327X Environment

Tremendous growth in the minicomputer, microcomputer, and personal computer markets has led to a rapid increase in the number of installed ASCII asynchronous terminals that access these computers. However, ASCII devices cannot access information that resides on IBM mainframes. IBM's series of products that provide interactive communications in an IBM network is the IBM 3270 Information Display System. This series includes controllers, terminals, and printers that are dedicated to a single host and usually to a single application.

Components in the current 3270 system include the 3278, 3279, 3178, 3179, 3180, and 3290 display terminals; the 3262, 3268, 3287, 4250, and 5210 printers; the 3274 and 3276 cluster controllers; and the 3270 Personal Computer. Each component comes in various models. For example, the 3278 is a monochromatic display available in five models that essentially differ only in their screen capacities. The 3279 is a color display version of the 3278. The 3274 controller comes in various models that handle up to 32 attached displays or printers, local or remote host connection, and BSC or SDLC protocol. The 3276 is a smaller controller designed for clusters of up to eight displays or printers.



EDA Instrument's BTX.2500 is a microprocessor-based packet assembler/disassembler that enables users of Burroughs Poll Select Protocol to use CCITT X.25-based data networks.

Because of the 3270's huge installed base, many models that are no longer actively marketed by IBM continue to play a significant role in the IBM-compatible markets, particularly the 3277 display terminal, and the 3271 and 3272 controllers. The 3271 is a remote cluster controller that handles up to 32 displays or printers and comes in BSC and SDLC versions. The 3272 is a local channel-attached version of the 3271.

There are some shortcomings to using products in the 3270 family. First, they are more expensive than ASCII terminals. Second, many of the IBM components are physically larger and take up more space than the ASCII terminals and the emulators that can be used in their place. (IBM has reduced both the price and size of its newer 3270 components, effectively eliminating these shortcomings.)

In 1979, IBM introduced the Model 3101 terminal that can attach directly to a 3705 communications controller and participate in ASCII applications resident in the host. With the introduction of the Model 3101, IBM acknowledged the need for asynchronous communication. The company has since introduced its second generation of asynchronous terminals, the 316X family.

IBM's first protocol converter, the 7426, introduced in October 1982, allowed the company's ASCII 3101 terminal to communicate with 8100 and 43XX computers. Although it was designed primarily for the 3101, the unit also enabled other asynchronous ASCII devices to connect to an SNA host. At the time the 7426 was introduced, interest in and sales of protocol conversion products had begun to increase dramatically, and several companies announced new converters that would allow asynchronous ASCII devices to emulate IBM 3270 equipment. From 1982 to 1984, revenues from protocol converter sales were strong, and IBM began making statements of direction concerning its intention to introduce more conversion products of its own.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

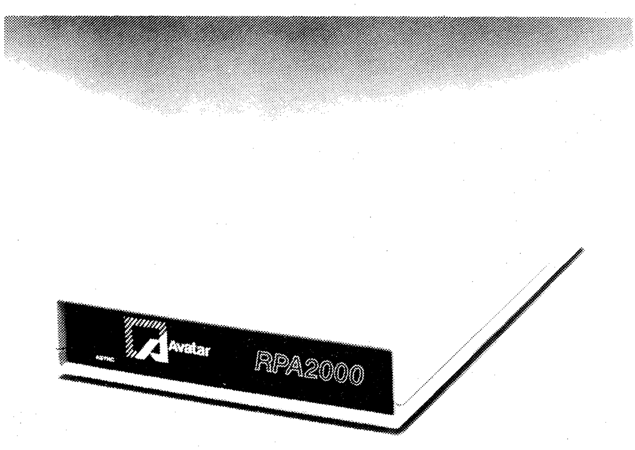
All About Protocol Conversion Systems

In September 1984, IBM announced the 7171 protocol converter and the 3710 Network Controller. The 7171 allows the direct attachment of from 16 to 64 asynchronous ASCII devices to the block multiplexer channel of a 43XX or 308X host. Devices attached to the converter appear to the IBM host as 3270-type equipment. The 3710 offers the ability to manage mixed protocols (start/stop, BSC, and SDLC) in the network, as well as to multiplex and concentrate lines from attached devices to a 37X5 communications controller. One of the chief advantages of the 3710 is its ability to off-load a variety of SNA network management functions from the communications controller, thus freeing that device for other tasks. The 3710's line concentration function also allows users to save port space on the controller.

After months of "statements of direction" concerning its plans regarding protocol conversion products, IBM introduced the 3708 Network Conversion Unit in 1985. The 3708 provides line concentration, protocol conversion, protocol enveloping, and ASCII passthrough support for asynchronous devices. The ten-port unit, designed for customer installation and maintenance, allows the attachment of one or two IBM hosts, asynchronous hosts, and asynchronous ASCII devices, which when attached to the unit emulate IBM 3270 equipment. The 3708 operates with IBM's System/370, 303X, 308X, 3090, and 43XX processors; 8100 Systems; System/38 units; the 3710 controller; and Rolm's CBX II voice/data PABX.

Competitors were quick to recognize the threat this new product posed. For a number of years, IBM had no protocol converters in its product line, and many manufacturers reaped the rewards of a strong market for devices that allowed asynchronous equipment to communicate with IBM hosts. Of course, a large part of this market included IBM mainframe customers. Now IBM has converters of its own to sell to its huge installed base, and the competition is forced to react to this formidable challenger.

Most importantly, however, IBM introduced 3270 emulation support for most of its mini- and microprocessor-based products including the IBM PC, the System 34/36/38, the 8100, and the 43XX. IBM also introduced the 3270-PC, a version of the PC that is designed specifically for use in a 3270 system. In doing so, IBM, in effect, changed the 3270 from a single-host, dedicated terminal system to a system that can accommodate many different devices.



The Avatar RPA2000 Protocol Converter provides 3270 terminal users with transparent access to ASCII systems.

Although the majority of protocol converters and terminal controllers on the market today handle some type of conversion between ASCII devices and IBM units, other products handle conversion between IBM BSC protocols and IBM SDLC protocols. This conversion is particularly useful to users of older IBM BSC equipment who wish to migrate to an SNA/SDLC environment without replacing all of their old equipment. BSC-to-SDLC conversions generally occur between BSC 2780/3780 RJE or 3270 BSC protocols and SDLC protocols.

As IBM PCs become increasingly prevalent in organizations, products to provide micro-to-mainframe compatibility will become more and more important. The entire protocol/emulation market is exploding today because units that make ASCII terminals and personal computers compatible with SNA/SDLC networks are in tremendous demand. ASCII devices provide flexible and inexpensive solutions to network problems, but IBM's mainframes are still the de facto standard for centralized computer facilities that must handle large databases and many applications. It seems unlikely that this situation will change soon, and vendors that offer conversion products to handle ASCII-to-IBM conversions should continue to enjoy a healthy market share for their products.

CURRENT TRENDS

Many different products handle some type of conversion to provide compatibility between communications devices. Presently, a number of large data communications equipment vendors are incorporating protocol converters and terminal controllers into their general line of products. Micom Systems, for example, acquired Industrial Computer Controls, Inc., one of the oldest specialized protocol converter manufacturers in the marketplace. At the same time, Micom introduced the Micro7400 protocol conversion, a replacement of ICCI's CA20 converter. The Micro7400 handles ASCII-to-3270 conversion and provides network monitoring and control functions as well.

Formerly the venue of small companies like Protocol Computers, Inc. and Innovative Electronics, which specialize in standalone protocol converter products, protocol conversion has become incorporated into existing data communications products. We now find conversion as an integral capability in digital data switches, PBXs, personal computers, and word processors.

From an historical perspective, we can benchmark interest in protocol conversion at IBM's introduction of its 7426 converter in October 1982. With this announcement, IBM not only sanctioned conversion technology as a viable solution to network problems, but also focused industry attention on the technology.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

Conversion products that facilitate LAN-to-LAN compatibility and access X.25 public data networks are also expected to have a large market. We can expect to see a growing interest in PAD devices that effect X.25 access, and we can anticipate greater PBX conversion capabilities in the months ahead. Conversion offerings from value-added carriers, such as Tymnet and Telenet, and from the Bell Operating Companies (BOCs), will also grow as data communications moves further into the home markets where personal computer users are becoming more interested in linking into public networks and databases.

Until the data communications industry adapts and uses worldwide protocol standards to link equipment, protocol converters and emulators will remain an important part of the general market. It is unlikely that such standardization will occur in the very near future.

CHOOSING CONVERSION DEVICES

Before choosing a conversion unit, users should consider some of the negative characteristics of the devices. First, protocol converters will cause delays in response time on the network because data must flow into the converter's buffer before transmission. If data backs up in the buffer, overruns occur; if the buffer is small, the converter can lose data.

Terminal operators dealing with devices that emulate other products may have problems learning the new key sequences and key functions necessary to the emulation process. Thus, organizations can expect some decreased productivity during the initial months of a conversion installation.

In addition, protocol converters usually do not offer the security provided by, for example, the IBM 3270-type devices. Organizations must deal with the problem of protecting sensitive data, particularly in dial-up applications.

When an organization decides to install conversion products in a data communications network, it should determine exactly what kinds of conversions are needed to solve particular incompatibilities; for example, a new mainframe is installed and the organization wants to use existing terminals, or the organization has purchased microcomputers and micro-to-mainframe connections are now required. Once the application is established, users should determine which kind of products can handle the conversion most effectively in a particular application. This can be an extremely confusing task because there are so many conversion products available. To narrow choices, it is wise to contact many vendors and ask for product specifications and documentation that explains how a product operates. When studying specifications and operating procedures, users must note exactly what types of terminals, controllers, or hosts are supported by the device because most converters and controllers support specific products rather than a general range of devices. For example, a protocol converter specifically designed for IBM 3277 emulation might not work with a 3278 application.

Also important is finding out what added features and functions the converter handles. Does it support more than one host? Does it replace an IBM controller, or is it used in conjunction with a controller? Does the device incorporate any multiplexing or concentrating? Is the device user-reconfigurable (e.g., transmission speed, parity)? Can the network manager monitor the network via the converter? If additional features are available, are they standard or optional? What cost savings will it represent in your overall networking scheme?

Other important considerations include availabilty, reliability, and service. What is the delivery lead time? Can the customer install the device? What training is required? Is the device available for lease or purchase? Are quantity discounts available? What is the average mean-time-between failures? Is the device serviced on- or off-site? What is the replacement policy?

After narrowing the choices to the products of several vendors, users should ask the company to provide an in-house demonstration of the product. A prospective buyer should also request a list of current users who will discuss their experiences with the product. These individuals can provide information about the advantages and disadvantages of the product, hardware reliability, and the type and quality of support provided by the vendor.

IBM mainframe users in particular should find out whether conversion equipment can be upgraded as IBM upgrades and changes its SNA architecture.

After further narrowing the selection to two or three vendors, users should request a free trial of the product. By using a converter in a particular application, prospective buyers can soon find out whether a product provides the desired compatibility in an efficient manner.

USER EXPERIENCE

In January and February 1986, Datapro conducted a LAN/Terminal Users Survey, which was based on results received from questionnaires mailed to a cross section of *Data Communications* magazine subscribers. Several questions in the survey pertained to the use of protocol conversion devices in a network. Below we show the results of these questions.

METHODOLOGY

A questionnaire was designed and produced by Datapro's senior data communications editors, and mailed in January to a selected group of subscribers to *Data Communications* magazine. These subscribers were identified as domestic end users of data communications equipment. The subscribers were asked to fill out the forms, providing ratings and other information for any conversion systems or conversion devices that they were using. They were then asked to return the completed form, in a postage paid envelope, to Datapro. By the cutoff date for returns, February 27, approximately 500 completed forms had been received by Datapro.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

When Datapro received the returns, they were edited by senior-level editors. All forms were examined for validity before being sent for tabulation. Subscriber names and addresses were used for initial validation and identification. In addition, responses to the survey were disqualified whenever a vendor/model identity was omitted, user ratings were not assigned, an obvious vested interest on the part of the respondent was judged to exist, or incomprehensible or unreasonable answers were given.

When the invalid forms had been eliminated, a total of 427 valid forms were identified. After these forms were processed by Datapro personnel, they were shipped to *Mathematica Policy Research, Inc. (MPR)*, of Princeton, NJ, for key entry and computer tabulation. Summary information was prepared in the form of totals, percentages, or weighted averages as appropriate for each question.

Weighted averages were used to determine the ratings given by the users of the equipment. These were computed in the following manner: "Excellent" was weighted as 4, "Good" was weighted as 3, "Fair" was weighted as 2, and "Poor" was weighted as 1. The tallied numbers for each value were then multiplied by the corresponding weight, and the average taken by dividing the sum of the products by the total number of responses for that category.

Users were asked to provide information on *all* conversion systems or devices that they were using. This resulted in multiple responses from the users. As a result, a total of 249 responses were received on conversion systems/devices. The responses were identified first by manufacturer, and then by model. A minimum of three responses was required to break out the ratings for a specific manufacturer. Once the manufacturer was identified with the proper number of responses, a minimum of two responses was required to break out the ratings for a specific model. The ratings are summarized in Table 1.

We asked the users to identify the *type* of conversion device they were using. An overwhelming majority, 167 (67 percent of the total responses), stated that they were using protocol converters. X.25 PADs received the second highest response, 26 (10 percent), while terminal controllers received 24 responses (10 percent). Terminal emulators (16 responses, or 6 percent) and interface/code converters (12 responses, or 5 percent) received the fewest responses.

We then asked the users to identify any types of protocol conversion that were being performed for their applications. Below are the responses of the users who answered this question:

	Number of Responses	Percent of Total Responses
ASCII-to-BSC	90	51
ASCII-to-SDLC	100	57
BSC-to-SDLC	13	7
Other	48	27

... and by what means this protocol conversion is performed:

	Number of Responses	Percent of Total Responses
Software loaded onto an existing system such as a general-purpose computer, front-end processor, terminal controller, or PBX system	50	28
Standalone conversion device, such as a protocol converter, interface/code converter, terminal emulator, X.25 PAD, or terminal controller	154	88
Value-added network service (e.g., Telenet)	21	12
Other	11	6

The strength of the protocol conversion market that has emerged in the past three years is certainly confirmed by these users' responses. The heavy use of ASCII-to-BSC and ASCII-to-SDLC conversion seems to bear out the conclusions concerning the increased use of asynchronous terminals we noted earlier in this report. Most likely, a great many of the ASCII-to-BSC conversions involve ASCII terminals emulating 3270-type devices.

Datapro strongly suggests that the reader use the information presented with discretion. The ratings are not intended as a statistically accurate indicator of the capabilities of a device. Rather, the ratings and other information should be used as guides to potential strengths and weaknesses of that device. The responses also may be examined to provide an indication of a manufacturer's share of the market. Any equipment acquisition decision should be made only after further investigation on the part of the buyer.

COMPARISON CHARTS

The charts that accompany this report present listings of the key characteristics of approximately 114 protocol converters or terminal emulators, 50 X.25 PADS, and 31 code, speed, interface, and async/sync converters. This information was supplied by the vendors during February and March of 1986. Datapro wishes to thank the participating vendors for their cooperation.

Datapro sent repeated requests for information to 70 firms known or believed to manufacture some type of hardware conversion device. *The absence of any company from the charts means that the company either failed to respond to our request by the survey deadline, was unknown to us, did not make a hardware conversion product, or chose not to be listed.*

Many companies presently manufacture conversion devices for microcomputer-to-mainframe communication. These products consist of a circuit board that is inserted into the microcomputer and accompanying software. These devices, predominanty designed to effect IBM PC-to-mainframe connections, comprise a large and growing segment of the conversion marketplace and are covered in a separate report in *Datapro Reports on Data Communications* behind the Microcomputer Communications (C22) tab in Volume 2.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

TABLE 1. USER RATINGS FOR CONVERSION SYSTEMS AND EMULATION DEVICES

MANUFACTURER/ MODEL	No. of User Responses	No. of Devices Installed	Ease of Installation					Ease of Operation					Device Reliability					Manufacturer's Maintenance Service/ Technical Support					Overall Performance					Would you Recommend This device?		
			WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	Yes	No	Undecided
Avatar—																														
All models	3	45	4.0	3	0	0	0	4.0	3	0	0	0	3.0	1	1	1	0	3.0	2	0	0	1	3.7	2	1	0	0	3	0	0
Black Box—																														
A/S Series	5	51	2.8	1	2	2	0	3.6	3	2	0	0	3.2	2	2	1	0	3.0	2	1	2	0	3.2	1	4	0	0	5	0	0
Others & unspecified	4	9	3.3	1	3	0	0	3.8	3	1	0	0	3.5	2	2	0	0	3.0	1	2	1	0	3.3	1	3	0	0	4	0	0
Subtotal	9	60	3.0	2	5	2	0	3.7	6	3	0	0	3.3	4	4	1	0	3.0	3	3	3	0	3.2	2	7	0	0	9	0	0
Codex—																														
4250	3	17	2.0	0	1	1	1	2.7	1	1	0	1	3.0	2	0	0	1	2.7	1	1	0	1	3.0	2	0	0	1	2	1	0
Commtex—																														
CX-80	4	5	3.0	1	2	1	0	2.8	1	1	2	0	3.3	1	3	0	0	2.5	0	2	2	0	3.0	1	2	1	0	1	1	1
Others & unspecified	6	38	3.0	0	6	0	0	3.0	1	4	1	0	3.5	3	3	0	0	3.0	1	4	1	0	3.0	0	6	0	0	5	0	1
Subtotal	10	43	3.0	1	8	1	0	2.9	2	5	3	0	3.4	4	6	0	0	2.8	1	6	3	0	3.0	1	8	1	0	6	1	2
Comstat Datacomm—																														
All models	3	13	3.0	0	3	0	0	3.3	1	2	0	0	4.0	3	0	0	0	3.0	0	3	0	0	3.7	2	1	0	0	3	0	0
Datastream—																														
All models	15	31	2.9	2	9	4	0	3.0	4	7	4	0	3.5	7	8	0	0	2.5	1	7	4	2	2.9	3	8	4	0	10	2	2
Digital Communications Associates (DCA)—																														
Irma	3	40	3.3	1	2	0	0	3.3	1	2	0	0	3.3	1	2	0	0	3.0	1	1	1	0	3.3	1	2	0	0	2	0	1
Others & unspecified	5	164	2.8	1	2	2	0	3.0	1	3	1	0	2.6	1	1	3	0	1.8	0	1	1	2	2.8	1	2	2	0	2	2	1
Subtotal	8	204	3.0	2	4	2	0	3.1	2	5	1	0	2.9	2	3	3	0	2.3	1	2	2	2	3.0	2	4	2	0	4	2	2
Digital Equipment Corporation—																														
SNA Gateway	5	9	3.2	1	4	0	0	3.2	1	4	0	0	3.3	2	1	1	0	3.0	1	2	1	0	3.2	1	4	0	0	5	0	0
Others & unspecified	2	6	3.5	1	1	0	0	3.0	1	0	1	0	2.0	0	0	1	0	3.0	0	1	0	0	3.5	1	1	0	0	2	0	0
Subtotal	7	15	3.3	2	5	0	0	3.1	2	4	1	0	3.0	2	1	2	0	3.0	1	3	1	0	3.3	2	5	0	0	7	0	0
Diversified Data Resources—																														
Hydra II	6	7	3.0	2	2	2	0	3.2	2	3	1	0	3.3	3	2	1	0	2.8	2	2	1	1	3.2	2	3	1	0	5	0	1
Dynapac—																														
All models	5	45	3.5	2	2	0	0	3.5	2	2	0	0	3.5	2	2	0	0	3.5	2	2	0	0	3.4	2	3	0	0	5	0	0
GTE Telenet—																														
All models	5	61	3.2	1	4	0	0	3.2	1	4	0	0	3.0	1	3	1	0	2.4	1	1	2	1	3.0	1	3	1	0	4	0	1
Hewlett-Packard—																														
HP 2334A	4	32	3.3	2	1	1	0	3.5	2	2	0	0	3.5	2	2	0	0	3.0	1	2	1	0	3.5	2	2	0	0	2	0	2
IBM—																														
7171	15	22	3.1	4	9	1	1	2.9	2	10	2	1	3.5	10	4	0	1	3.3	8	5	1	1	3.4	8	6	0	1	14	1	0
Others & unspecified	10	22	2.5	1	5	2	2	2.7	2	4	3	1	3.0	4	3	2	1	2.8	4	3	0	3	2.9	2	6	1	1	5	2	3
Subtotal	25	44	2.8	5	14	3	3	2.8	4	14	5	2	3.3	14	7	2	2	3.1	12	8	1	4	3.2	10	12	1	2	19	3	3
Icot—																														
All models	3	54	2.7	0	2	1	0	2.3	0	1	2	0	2.7	0	2	1	0	2.0	0	1	1	1	2.3	0	1	2	0	3	0	0
ITT—																														
All models	5	18	3.2	3	0	2	0	3.2	3	0	2	0	3.0	2	1	2	0	2.8	2	1	1	1	3.2	3	0	2	0	4	1	0
KMW—																														
All models	4	27	2.8	0	3	1	0	3.3	1	3	0	0	2.8	0	3	1	0	3.3	2	1	1	0	2.8	0	3	1	0	3	0	1
Local Data—																														
DataLynx/3274	4	6	3.0	1	2	1	0	3.3	1	3	0	0	3.0	0	4	0	0	2.8	0	3	1	0	3.0	0	4	0	0	4	0	0
Others & unspecified	8	30	2.9	2	3	3	0	2.9	0	7	1	0	2.9	3	2	2	1	2.5	0	4	4	0	2.8	0	6	2	0	6	1	1
Subtotal	12	36	2.9	3	5	4	0	3.0	1	10	1	0	2.9	3	6	2	1	2.6	0	7	5	0	2.8	0	10	2	0	10	1	1
Micom—																														
Micro 7400	6	25	3.0	2	2	2	0	3.3	3	2	1	0	3.3	3	2	1	0	2.2	1	1	2	2	3.0	2	2	2	0	4	0	2
Others & unspecified	2	9	3.0	0	2	0	0	3.0	1	0	1	0	3.5	1	1	0	0	2.0	0	0	2	0	3.0	0	2	0	0	1	0	1
Subtotal	8	34	3.0	2	4	2	0	3.3	4	2	2	0	3.4	4	3	1	0	2.1	1	1	4	2	3.0	2	4	2	0	5	0	3

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

TABLE 1. USER RATINGS FOR CONVERSION SYSTEMS AND EMULATION DEVICES (Continued)

MANUFACTURER/ MODEL	No. of User Re-sponses	No. of De-vices In-stalled	Ease of Installation					Ease of Operation					Device Reliability					Manufacturer's Maintenance Service/ Technical Support					Overall Performance					Would you Recommend This device?		
			WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	Yes	No	Un-decided
Protocol Computers (PCI)—																														
1067	4	6	2.8	1	2	0	1	3.0	1	2	1	0	3.0	2	1	0	1	2.3	1	1	0	2	2.8	1	2	0	1	2	1	1
1071	3	28	3.0	1	1	1	0	3.0	1	1	1	0	2.7	0	2	1	0	2.0	0	1	1	1	2.7	0	2	1	0	2	1	0
1076	8	14	2.9	0	7	1	0	3.4	3	5	0	0	3.3	3	4	1	0	3.1	2	4	1	0	3.1	2	5	1	0	7	0	1
Others & unspecified	15	33	3.1	5	8	1	1	3.1	6	7	0	2	2.9	2	9	1	1	2.3	2	5	0	5	2.9	4	5	4	1	10	3	2
Subtotal	30	81	3.0	7	18	3	2	3.2	11	15	2	2	3.0	7	16	3	2	2.5	5	11	2	8	2.9	7	14	6	2	21	5	4
Renex—																														
All models	18	343	3.1	6	6	5	0	3.1	4	10	3	0	3.1	5	9	3	0	2.4	3	4	7	3	3.1	4	11	3	0	14	1	3
Rolm—																														
Gateway	3	4	3.0	1	1	1	0	3.3	1	2	0	0	3.3	1	2	0	0	2.7	1	1	0	1	3.3	1	2	0	0	3	0	0
Wall Data—																														
All models	3	3	3.0	1	1	1	0	3.3	1	2	0	0	3.7	2	1	0	0	2.7	1	1	0	1	3.3	1	2	0	0	3	0	0
All others	60	649	3.0	14	34	10	2	2.9	14	30	11	3	3.0	17	26	13	2	2.8	18	17	15	8	2.9	14	28	15	2	39	12	7
GRAND TOTAL	249	1,866	3.0	61	132	46	8	3.1	72	127	38	8	3.1	88	108	37	8	2.7	61	85	54	37	3.0	65	132	43	7	184	29	32

Weighted Average (WA) is based on an assigning a weight of 4 to each user rating of Excellent (E), 3 to Good (G), 2 to Fair (F), and 1 to Poor (P).

KEY TO CONVERSION EQUIPMENT TABLE ENTRIES

For the reader's convenience, we have organized the comparison charts into three broad categories. **Conversion Systems/ Terminal Controllers** can be a variety of devices, including protocol converters, terminal emulators, remote cluster controllers, and terminal controllers. Basically, devices in this section provide conversion from one protocol to another and/ or allow one device, e.g., an asynchronous ASCII terminal, to act as another type of device, e.g., an IBM 3270 terminal, in a network. **X.25 Packet Assemblers/Disassemblers (PADS)** convert equipment using a non-X.25 protocol to the X.25 protocol for transmission over public and private networks. PADs also perform other functions, including concentrating or multiplexing. **Code, Speed, Interface, and Async/Sync Converters** include a number of miscellaneous devices that handle conversions from one code, interface, speed, or synchronization to another. These are generally less sophisticated devices than those represented in the other two categories.

The following text briefly describes the chart entries in the order in which they appear in the charts. Not all of the characteristics listed appear in all of the charts because some entries do not apply to all categories. For example, "link-level framing support" is relevant only to X.25 PADs.

General Characteristics

Manufacturer and model. We list here the manufacturer and exact model number or name of each device.

Device type. In the Conversion Systems/Terminal Controllers section, we have asked the manufacturer to specify whether the device is a protocol converter, terminal emulator, code or interface converter, and so forth.

Conversion performed. All converters perform some type of translation from one code, speed, or protocol to another. The most common conversion is asynchronous ASCII to IBM SNA/SDLC or BSC, but a number of other translations are available on the units represented in the tables.

Specific device emulated. In many cases, conversion devices provide the means to convert the text format of one type of device into the characteristics and format of another device. This translation, called emulation, is indicated, if available. Note that most protocol converters also provide device emulation.

Specific functionality provided. Most converters allow one device to be used as another type of device in the network. For example, a number of units allow asynchronous ASCII CRTs to be used as IBM 3277 Models 1 and 2. It is very important for prospective users to know exactly what type of device can be emulated by a given converter because many converters are designed for a specific application.

Virtual screen sizes supported. For a device to provide emulation, it must support the screen size, in characters, of the emulated device. For example, a device emulating an IBM 3270 terminal must support a 1,920-character screen.

Command port support. Some converters support a port through which users can select operating parameters and monitor, diagnose, and control the network. A "yes" answer indicates that the device does have a command port.

Network certification. X.25 PADs allow devices to interface with public and private networks. These networks must certify use of the device on their facilities. Prominent network providers include GTE Telenet, Tymnet, and Uninet in the U.S., and Datapac and Infoswitch in Canada.

CCITT recommendation supported. CCITT has specified certain protocols for devices operating on an X.25 network. Most X.25 PADs support all CCITT recommendations, including X.25 Levels 1, 2, 3; X.3; X.28; and X.29. X.25 defines the operating protocol of the packet network. X.3 defines a set of 18 parameters that govern operation of the PAD and each asynchronous terminal port. X.28 defines the interface between the asynchronous terminal and the PAD, and X.29 defines the format of the packets that carry control information from the PAD to the remote destination.

Buffer memory capacity. PAD devices contain a buffer memory that holds packets before transmission. Software is generally held in ROM.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

KEY TO CONVERSION EQUIPMENT TABLE ENTRIES (Continued)

Additional RAM available. Many PADs can accommodate the additional RAM necessary to expand the capacity of the device.

Device configuration. Most X.25 PADs can be configured to act as other types of devices in the network. Typical configurations include host concentration, time division multiplexing, or front-end processing.

Host Side/Network Channel Specifications

Specific hosts supported. Conversion devices generally support IBM or compatible hosts or asynchronous hosts such as Digital's VAX or PDP 11. In this entry, we have listed the type of mainframe with which the converter operates.

Number and type of host lines supported. Most converters support one line to a host; that line can be one IBM SNA/SDLC or BSC line, or one line to an asynchronous host. Some devices do support more than one host connection. This connection may be a dual IBM BSC or SDLC link, or one IBM link and one link to an asynchronous host.

Number of host sessions supported concurrently. If a converter supports more than one host line, the device may support both connected hosts concurrently, or separately through a switch selection.

Connections supported. Conversion devices support direct connections, multipoint, and/or point-to-point connections. Most converters support more than one type of connection, and most support all three.

Connection to host via controller. While some conversion devices emulate a controller, others must connect to a controller in the network. We have asked the vendors to specify the type of controller to which the converter interfaces, if applicable.

Number of X.25 channels supported. Here the vendor has specified the number of channels a PAD supports for connection to the public or private data network.

Number of virtual circuits supported. A virtual circuit is the logical connection between the input port and the destination port. When a terminal connects to a PAD, the PAD automatically or manually establishes a circuit to the destination by selecting an unused logical channel number and transmitting a Call Request packet that uses that logical channel number. This request packet identifies the input port and the destination port and carries other information used to set up the logical connection. In this entry, vendors have specified the maximum number of virtual circuits supported by the PAD.

Maximum window size, in frames. Window size describes the maximum number of unacknowledged frames (packets) that can be handled by the PAD at one time. Generally, PADs support up to seven frames in the window. When the PAD's transmitter reaches the maximum window size, it blocks the window. In effect, window framing is a form of flow control.

Link-level framing supported. At the link level, blocks of data are assembled according to certain framing protocols. These include character- or bit-oriented framing (BSC or HDLC, respectively). This is an EBCDIC or ASCII option on character-oriented (BSC) framing.

Terminal Side/Input Specifications

Number of types of ports (or devices) supported. In general, a conversion device supports asynchronous ports that accommodate a large variety of asynchronous ASCII printers, terminals, and personal computers. Many converters also support a dynamic printer port. Although more uncommon, conversion devices and PADs do support synchronous ports, or asynchronous and synchronous ports in combination. Devices represented in the charts support from one to many input devices; a number of units accommodate input-port expansion in specified increments.

Specific devices supported. It is important to know whether the unit supports a particular type of terminal. In today's market, most conversion devices designed for asynchronous ASCII to IBM SDLC or BSC conversion support virtually any asynchronous ASCII device. Some converters, however, are designed for operation only with a specific terminal. In this entry, the vendors have noted the manufacturer and model number of devices supported. An answer of "virtually any device" means that the vendor's list of supported terminals was too long to fit into the assigned space, but the converter did support all major asynchronous ASCII terminals and/or personal computers available in today's market.

Autospeed/autoparity available. Many X.25 PADs automatically adjust to the transmission speed and parity of the inputting DTE. A "yes" answer indicates that the PAD supports this feature.

Channel configuration data downline loadable. X.25 PADs may support this feature, which allows terminal operators to configure channel parameters from the terminal and download those configurations to the PAD.

Ports configurable for permanent or switched circuits. Some PADs will allow users to configure an input port for permanent or switched virtual circuit connection through the network. In cases where the circuit is switched, the termination of a logical connection signals that the connection is free and can be used by another port. When the virtual circuit is permanent, the connection is dedicated to one port only. Many PADs support both permanent and switched virtual circuits on a selectable basis.

Transmission Specifications

Maximum transmission, in bps. This entry indicates the maximum speed of operation or data rate supported by the device stated in bits per second.

Maximum aggregate input rate, in bps. Conversion devices generally support many input ports, each operating at several different speeds, e.g., from 50 to 9600 bps. Aggregate input refers to the maximum data rate accepted from all channels simultaneously. For example, if there are four channels operating at a maximum 9600 bps rate per channel, the aggregate input rate could be four times 9600, or 38.4K bps.

Synchronization. This refers to the time relationship among the bits that make up the characters, which make up the messages. Conversion devices handle data in spurts (asynchronous) or continuous streams (synchronous).

Transmission mode. Most converters operate in either half- or full-duplex mode, or both. Half-duplex mode permits data transmission in either direction, but not simultaneously. Full-duplex operation implies that the data is simultaneously transmitted and received over a common communications facility. Simplex mode permits unidirectional data transmission, whereby data is either transmitted or received.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

KEY TO CONVERSION EQUIPMENT TABLE ENTRIES (Continued)

Protocols supported. Protocols are a set of rules that establish and control transmission. There are two basic types of protocols: byte-oriented (IBM's BSC or Digital's DDCMP) or bit-oriented (IBM SNA/SDLC or ISO HDLC). Converters usually translate one protocol to another and thus support different protocols on the terminal and host sides.

Codes supported. Codes consist of specific sets of characters that can be alphanumeric, graphic, and control characters. Control characters initiate, modify, or halt an action that effects data transmission. The most common data communications codes are ASCII, used in the asynchronous protocol, and EBCDIC, the usual code generated by synchronous devices.

Interface. Interface is the electrical connection between components. Most communications devices provide an electrical interface (RS-232-C) in accordance with the standards established by the Electronics Industries Association (EIA). Several other interface standards exist, notably CCITT Recommendation V.24 and V.28.

Clocking. Clocking refers to the repetitive, regularly timed signals used to control synchronous transmission. Clocking may be established internally by the device itself, externally by another device (for example, a modem), or be derived from the datastream.

Diagnostics. Many conversion devices perform tests that check the device and the line connections. Most converters conduct a self-test of internal circuitry upon power-up and provide front-panel LEDs to monitor system status.

Features (on X.25 PADs)

Channel priority assignment. Some PADs allow users to assign priority to incoming channels. Whenever the priority channel requests a connection to the network, that channel receives immediate access to the PAD and "bumps" channels with less priority.

Password protection. On many devices, users must enter a password to gain access to the PAD. This feature prevents unauthorized access to the network.

Supervisory port. Through this port, users can monitor and control the network and send messages throughout the system.

Echoplex. This feature refers to the printing of keyboarded characters on return of the signal from the other end of the line, using full-duplex transmission, to assure that the data was received correctly at the other end.

Autodialer support. Autodialers allow users to set dialing parameters, such as delay specifications for dial tone and call answering and switch-to-switch delay pauses, in memory. When this feature is present, dialing and disconnecting calls occur automatically.

Pricing and Availability

Purchase price. This is the basic price of the unit, excluding any options, except where noted in the Comments.

Rental. The monthly charge for leasing the unit from the vendor or a third party is shown in this entry.

Installation. When vendors charge a fee for installing the unit, we have included the onetime charge. Note that most conversion devices are designed for customer installation.

Maintenance. Many vendors charge an annual fee to service the unit and provide ongoing maintenance.

Serviced by. In this entry, we list the provider of service on the unit. Usually, the vendor offers service on an on-site or factory repair/return basis. In some cases, a third party provides service.

Availability. Here we list the current lead time on orders, given in days after receipt of order (ARO).

Date of first commercial delivery. Here we provide the date when the vendor first delivered the product to the marketplace.

Number installed to date. Some vendors list the approximate number of installed units as of March 1984. Note that in some cases, the vendor combines this figure for all models installed.

Comments. In this section, we have listed various special characteristics pertaining to a particular device. These might include additional capabilities, features, software, as well as information regarding related products offered by the vendor.

VENDORS

Listed below are the names, headquarters addresses, and telephone numbers of vendors who manufacture conversion devices. We have provided this list so that readers can contact the vendors for more information about the products they offer.

Advanced Computer Communications, 720 Santa Barbara Street, Santa Barbara, CA 93101. Telephone (805) 963-8801.

Agile Corporation, 4041 Pike Lane, Concord, CA 94520. Telephone (415) 825-9220.

Air Land Systems Corporation, 2710 Prosperity Avenue, Fairfax, VA 22031. Telephone (703) 573-1100.

Altertext, 210 Lincoln Street, Boston, MA 02111. Telephone (617) 426-0009.

Amdahl CSD, 2200 North Greenfield Avenue, Richardson, TX 75081. Telephone (214) 699-9500.

Ark Electronic Products, Inc., 1500 West Nasa Boulevard, Melbourne, FL 32901. Telephone (305) 724-5260.

Avanti Communications Corporation, Aquidneck Industrial Park, Newport, RI 02840. Telephone (401) 849-4660.

Avatar Technologies Inc., 99 South Street, Hopkinton, MA 01748. Telephone (617) 435-6872.

BBN Communications Corporation, 70 Fawcett Street, Cambridge, MA 02238. Telephone (617) 497-2800.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

Black Box Corporation, P.O. Box 12800, Pittsburgh, PA 15241. Telephone (412) 746-5500.

Cableshare Inc., P.O. Box 5880, 20 Enterprise Drive, London, Ontario, Canada N6A 4L6. Telephone (519) 686-2900.

CASE Communications, 2120 Industrial Parkway, Silver Spring, MD 20904-1999. Telephone (301) 622-2121.

Com Design, 751 South Kellogg Avenue, Goleta, CA 93117. Telephone (910) 334-1189.

Com/Tech Systems, Inc., 505 Eighth Avenue, New York, NY 10018. Telephone (212) 594-5377.

Commercial Data Processing, Inc., 2241 S. Grand Avenue, St. Louis, MO 63104. Telephone (314) 776-1130.

Commtex, 2411 Crofton Lane, Crofton, MD 21114. Telephone (301) 721-3666.

Computer Communications, Inc., 2610 Columbia Street, Torrance, CA 90503. Telephone (213) 320-9101.

Computer Peripheral Systems, Inc., Box 98282, Atlanta, GA 30059. Telephone (404) 292-9565.

Comstat Datacomm Corporation, 1351 Oakbrook Drive, Suite 165, Norcross, GA 30093. Telephone (404) 446-9496.

Control Concepts, P.O. Box 2367, Mannassas, VA 22110. Telephone (703) 361-5545.

CTiData Corp., 5275 North Boulevard, Raleigh, NC 27604. Telephone (919) 876-8731.

Datagraf, Inc., 6626 Silvermine Drive, Suite 100, Austin, TX 78736. Telephone (512) 288-0453.

Datagram Corporation, 11 Main Street, East Greenwich, RI 02818. Telephone (800) 235-5030.

Datamaxx USA Corporation, P.O. Box 6477, 1815 South Gadsden Street, Tallahassee, FL 32314. Telephone (904) 224-8213.

Dataprobe, 110 West Palisades Boulevard, Palisades Park, NJ 07650. Telephone (201) 947-9500.

Datastream Communications, Inc., 2520 Mission College Boulevard, Santa Clara, CA 95050. Telephone (408) 986-8022.

Datatel, Inc., Cherry Hill Industrial Center, Cherry Hill, NJ 08003. Telephone (609) 424-4451.

Davox Communications Corp., 4 Federal Street, Billerica, MA 01821. Telephone (617) 667-4455.

DCC/Duracom Corporation, 7300 North Crescent Boulevard, Pennsauken, NJ 08110. Telephone (609) 642-7272.

Digital Communications Associates, 1000 Alderman Drive, Alpharetta, GA 30201. Telephone (404) 442-4000.

Diversified Data Resources, Inc., 25 Mitchell Boulevard, Suite 7, San Rafael, CA 94903. Telephone (415) 499-8870.

Dynapac, A Dynatech Company, 6464 General Green Way, Alexandria, VA 22312. Telephone (703) 642-9391.

EDA Instruments, Inc., 4 Thorncliff Park Drive, Toronto, Ontario, Canada M4H 1H1. Telephone (416) 425-7800.

Fibronics International, 325 Stevens Street, Hyannis, MA 02601. Telephone (617) 778-0700.

Forest Computer, 1749 Hamilton Road, P.O. Box 509, Okemos, MI 48864. Telephone (517) 349-4700.

Gandalf Data, Inc., 350 East Dundee, Suite 201, Wheeling, IL 60090. Telephone (312) 541-6060.

General Datacomm Industries, Middlebury, CT 06762-1299. Telephone (203) 574-1118.

GTE Telenet Communications Corp., 12490 Sunrise Valley Drive, Reston, VA 22096. Telephone (703) 689-6000.

Hewlett-Packard Grenoble, 5 Avenue Raymond Chanas 38320 Eybens, R.C.S. Grenoble, France B7098 05030. Telephone (76) 25 81 41.

Icot Corporation, P.O. Box 5143, San Jose, CA 95150. Telephone (408) 433-3300.

Infotron Systems, 9 North Olney Avenue, Cherry Hill, NJ 08003. Telephone (609) 424-9400.

Innovative Electronics, Inc., 4714 NW 165th Street, Miami Lakes, FL 33014. Telephone (305) 624-1644.

Instrumentation Services Incorporated (ISI), 957 Winnetka Avenue North, Minneapolis, MN 55427. Telephone (612) 544-8916.

International Business Machines Corporation, Old Orchard Road, Armonk, NY 10504. Contact your local IBM representative.

KMW Systems, 8307 Highway 71 West, Austin, TX 78735. Telephone (512) 288-1453.

Local Data, Inc., 2771 Toledo Street, Torrance, CA 90503. Telephone (213) 320-7126.

M/A Com Alanthus Data, Inc., 5515 Security Lane, Suite 1100, Rockville, MD 20852. Telephone (301) 984-3636.

May-Craft Information Systems, Inc., 3412 Beltwood Parkway, S., Dallas, TX 75244. Telephone (214) 392-3766.

Memotec Data Inc., 600 McCaffrey, Montreal, Quebec, Canada H4T 1N1. Telephone (514) 738-4781.

Micom Systems, Inc., P.O Box 8100, Simi Valley, CA 93062-8100. Telephone (805) 583-8600.

Modemsplus, Inc., 217 East Trinity Place, Decatur, GA 30030. Telephone (404) 378-5276.

NCR Comten, Inc., 2700 Snelling Avenue North, St. Paul, MN 55113. Telephone (612) 638-8592.

Netlink Technology, 2920 Highwoods Boulevard, Suite 110, Raleigh, NC 27625. Telephone (919) 878-8612.

Perle Systems Ltd., 360 Tapscott Road, Scarborough, Ontario, Canada M1B 3C4. Telephone (416) 299-4999.

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Protocol Conversion Systems

Protocol Computers, Inc., 6150 Canoga Avenue, Woodland Hills, CA 91367-3773. Telephone (818) 716-5500.

ProtoCom Devices, 190 Willow Avenue, Bronx, NY 10454. Telephone (212) 993-0077.

Quasitronics, Inc., 211 Vandale Drive, Houston, PA 15342. Telephone (800) 245-4192; (412) 745-2663.

Telebyte Technology Inc., 270 E. Pulaski Road, Greenlawn, NY 11740. Telephone (516) 423-3232.

Renex Corporaton, 1513 Davis Ford Road, Woodbridge, VA 22192. Telephone (703) 494-2200.

Shaffstall Corporation, 7901 East 88th Street, Indianapolis, IN 46256. Telephone (317) 842-2077.

Telematics, Crown Center, 1415 N. 62nd Street, Fort Lauderdale, FL 33309. Telephone (305) 772-3070.

Teleprocessing Products, 4565 East Industrial Street, Building 7K, Simi Valley, CA 93063. Telephone (805) 522-8149.

Thomas Engineering, 2440 Stanwell Drive, Concord, CA 94520. Telephone (415) 680-8640.

Timeplex, Inc., 400 Chestnut Ridge Road, Woodcliff Lake, NJ 07675. Telephone (201) 391-1111.

Tri-Data Corporation, 505 East Middlefield Road, Mountain View, CA 94039-7505. Telephone (415) 969-3700.

Tymnet, 2450 North First Street, San Jose, CA 95131. Telephone (408) 435-7756.

Universal Data Systems, 5000 Bradford Drive, Huntsville, AL 35805. Telephone (810) 726-2100.

Wall Data, Inc., 17769 N.E. 78th Place, Redmond, WA 98052. Telephone (206) 883-4777.

Wolfdata, Inc., 187 Billerica Road, Chelmsford, MA 01824. Telephone (617) 250-1500. □

© 1986 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

This report discusses protocol conversion and emulation in the data communications environment. The various types of conversions that can take place in a network are explained using the OSI seven-layer model for data communications as a reference. We report on various types of conversion products, including code and interface converters, protocol converters, terminal controllers and emulators, PAD devices, and communications interface cards for microcomputers. Also included are a discussion of the IBM 3270 environment, a review of current trends in the conversion market, and recommendations for selecting conversion products. We also present the 1984 Terminal Users Survey results that pertain to protocol conversion.

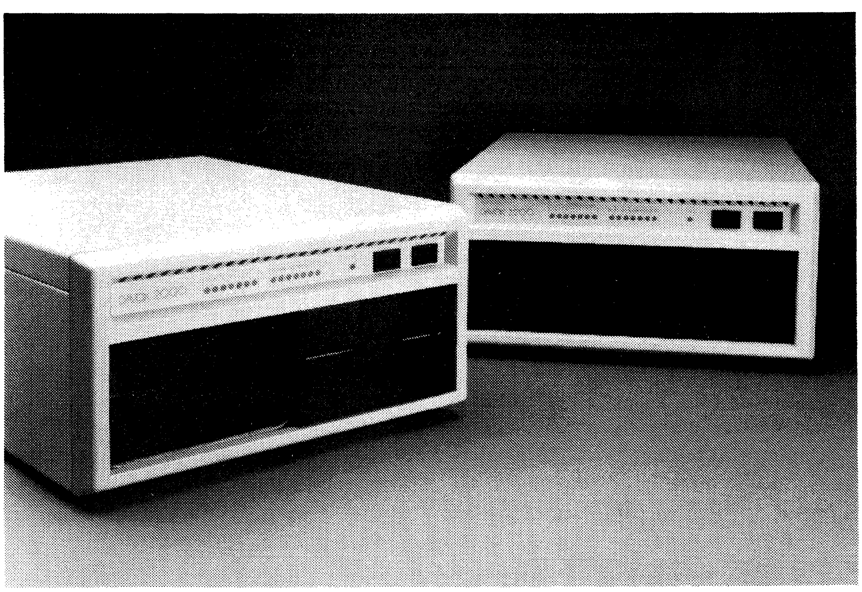
For your convenience, we have listed the names and addresses of vendors that offer conversion products listed in the comparison charts that follow this report.

Intrinsic to any communications system is the problem of translation. Translators of written and spoken languages have always faced the complexities of converting one language to another. The technological advancements of the twentieth century have created the means to produce machines with the ability to communicate with one another. The science of such communication, which we call data communications, has become one of the most important developments of the Information Age. But like other communications systems, translation problems are an inherent part of data communications.

Data communications network designers can achieve flexibility and economic rewards by using products from more than one vendor. However, hardware and software from different vendors are seldom compatible with one another. They rarely speak the exact same language. If a user wants to design a network of diverse hardware and software products, he or she must deal with the problem of incompatibility. In doing so, the user must first understand exactly how various devices are incompatible with one another and then determine the most effective way to deal with the differences.

In data communications, the solution to the problem of incompatibility lies in special hardware and software products that perform some type of conversion that translates the communications system of one device into that of another. Today, there are growing numbers and varieties of these products to handle many types of incompatibilities in the data communications network. These products range from microprocessor-based circuit boards to front-end processors with the ability to handle conversion functions through software applications programs. Available conversion devices may handle only one, or more than one, type of conversion. For example, some devices handle only code or interface conversions, while others handle protocol conversion, device emulation, as well as code and interface translations.

This report concentrates on hardware conversion devices, particularly "black box" protocol converters and terminal controllers. We are aware that software packages for conversion and emulation are an extremely important part of the market. However, this reference service is primarily concerned with hardware. Readers interested in software



Davox Controller Systems provide multiprotocol communications (3270 and async), IBM PC resource sharing, and networking for Davox integrated voice/data workstations. The Series 5000 Dual-Host Controller System, pictured here, allows a user to switch communications paths between two host mainframes at speeds up to 56K bps.

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

conversion products should consult the *Datapro Directory of Software* and the *Datapro Directory of Microcomputer Software.*

For coverage of the large and expanding micro-to-mainframe segment of the market see Report C22-010-101, Communications Capabilities of Microcomputers in this Volume.

In this report, we focus attention on the ways in which devices must be compatible in order to communicate. Using the Open Systems Interconnection (OSI) seven-layer model for data communications as a guide, we explain the various kinds of conversions that can take place between devices. We then discuss the various products that handle particular conversions and the ways in which conversions occur. The report also contains a discussion about conversion in the IBM 3270 environment, since solving problems of incompatibility between ASCII devices and IBM hosts is of particularly high interest to many readers. Also included are discussions about current trends in the conversion marketplace, some tips for selecting conversion products, and the results of a section of our 1983 Terminal User's Survey concerning protocol conversion. At the end of the report is a list of vendors that provide various kinds of conversion products; their addresses and phone numbers are included.

INCOMPATIBILITY IN DATA COMMUNICATIONS

We have said that data communications devices can be incompatible with one another in several ways. The International Standards Organization (ISO) Open Systems Interconnection reference model—a seven-layer hierarchy that defines the electrical characteristics, communications standards, and software applications for computer systems—provides a framework for understanding the ways in which devices differ. Each layer of the model defines a particular aspect of the entire data communications process. Refer to Figure 1 for a representation of the seven-layer hierarchy.

- Layer 1 is the *Physical Layer,* which provides mechanical and electrical specifications and procedures to establish, maintain, and end physical connections. This layer defines interface, code, speed, and synchronization functions. Interface, code, and asynchronous-to-synchronous converters fall into this category.

- Layer 2 is the *Data Link Layer,* which insures that the data passes without error from one computer to another. This process involves protocols, a set of rules that specify the format for data transmission. Protocol converters are the devices that handle conversions in this layer.

- Layer 3 is the *Network Layer,* which lets two systems exchange data. This layer defines packet addressing and data routing to final destination. Units that handle conversion in this layer include gateway devices, such as packet assemblers/disassemblers that provide access to X.25 networks or between local area networks. Front-end processors (FEPs) that include protocol conversion in their functions also fall into this classification.

ISO SEVEN-LAYER MODEL FOR DATA COMMUNICATIONS

(7) Application—provides communications services
(6) Presentation—defines syntax of data
(5) Session— controls data exchange
(4) Transport—handles data flow, error control
(3) Network—handles data routing
(2) Data Link—ensures data transfer via protocols
(1) Physical—provides mechanical/electrical interface

Figure 1. Layers One through Three define the interface between the host computer and the network. Layers Four through Seven provide compatibility to data format and exchange.

- Layer 4 is the *Transport Layer,* which handles end-to-end error and flow control to ensure that the communications exchange is orderly and reliable. PAD devices, a type of gateway product, are the major products in this layer. Note that we classify PADs in both the Network and Transport layers.

- Layer 5 is the *Session Layer,* which provides the structure for a data exchange by managing connections between application processes, establishing and terminating connections, and sending end-to-end messages and controller dialogues. There are currently few conversion products in this category; ProtoComm's P2500 PAD device does handle conversion on the Session Layer, but is one of the few products that does so.

- Layer 6 is the *Presentation Layer,* which both defines the way in which data is put together and provides a systematic arrangement for the communications exchange to occur. This layer defines functions to translate coded data and convert it into display formats for terminal or microcomputer screens, printers, and other peripherals. In this layer, data is expanded or compressed and structured for file transfer or command translation. Devices called emulators that allow one type of terminal to appear as another type of terminal fall into the Presentation Layer category. Products in this category include ASCII-to-3270 emulators, interfaces that let personal computers act as 3270-type devices or access public networks, and word processor interfaces that handle conversions between dissimilar word processors.

- Layer 7 is the *Applications Layer,* which supports user and application tasks and provides the communications services that are available to specific computer applications. In essence, this layer provides the meaning to the message. Conversion devices that we discuss in this report do not provide conversions on this layer.

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

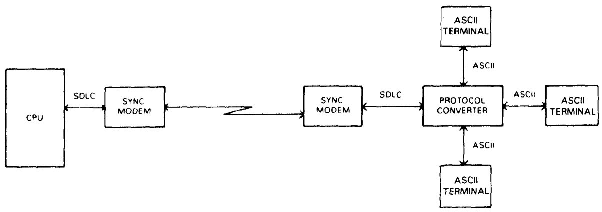


Figure 2. A typical configuration of a protocol converter at the terminal end of the line.

For devices to communicate with one another, they must be compatible on the interface, code, and protocol levels and must be alike according to link characteristics, device type, and device characteristics. Therefore, to connect incompatible equipment, converters must often provide translations on more than one of the levels in the network model. Conversion at one layer generally implies that compatibility in the layers below it in the model must also be accomplished. For example, a protocol converter working on Level 2 functions also assumes responsibility for compatibility in the interface, code, and synchronization functions.

Below, we discuss the various products that handle conversion functions. These include interface, code, and asynchronous-to-synchronous converters; protocol converters; gateway devices, including PADs; protocol conversion in front-end processors; and terminal emulation/controllers, and remote cluster controllers that let one device appear as another.

Interface, Code, and Asynchronous-to-Synchronous Converters

An interface is the physical connection between two devices; interface conversion is the lowest level of established compatibility. Data and control lines from a device terminate in a connector that has pins that handle assigned signal functions. For example, the industry standard RS-232-C interface connector has 25 pins—one pin per function. The interface also prescribes voltage levels for electrical signals passing over the data and control lines.

Interface converters handle incompatibility between two interfaces. The devices link incompatible plugs, accept the connectors of two different interfaces, and/or translate the signals and voltage levels of one interface to that of another. Interfaces conversions commonly occur between RS-232-C and MIL-STD-188 or between RS-232-C and V.35. Several vendors, including Avanti Communications Corporation, Gandalf, and Remark Datacom, offer products that handle many different types of conversions at the interface level.

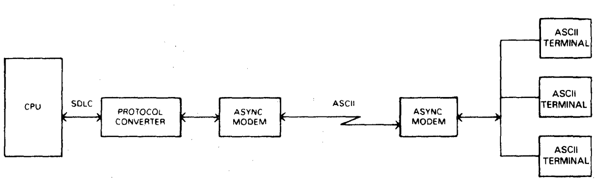


Figure 3. A typical configuration of a protocol converter at the host end of the line.

Code converters handle the transformation of one communications code to another. A communications code is a bit pattern for each text, graphics, or control character. The most common data communications codes are ASCII, EBCDIC, and Baudot. An end-user device that operates using one of these codes cannot accept data in another code. In addition, all error-checking codes (e.g., parity) must be compatible. The conversion from one code to another may be simple, involving only the addition or deletion of control bits or the alteration of parity. A more complex code conversion might require transforming the data character's bit pattern.

Basic code conversion hardware consists of two universal asynchronous/synchronous receiver/transmitters (USARTs), a translation table contained in ROM, and some control circuitry. Characters received by the USART in one code are mapped in the ROM table into a corresponding character in the destination device's code. Converted data goes to the other USART, which transmits it to the destination device.

One of the biggest problems of code conversion today is that of integrating word processors into data processing networks. Word processors typically have large character sets and control characters that are not used by data communications equipment. In some cases, the data communications device uses a word processor character for a different function. To integrate word processors into a data communications network, users must first convert the code of the word processor to a code that data communications equipment understands.

Placing word processors in data communications networks is difficult for other reasons. In many cases, the word processor manufacturer has developed a complete communications protocol for the equipment. Changing that protocol requires a higher level of conversion.

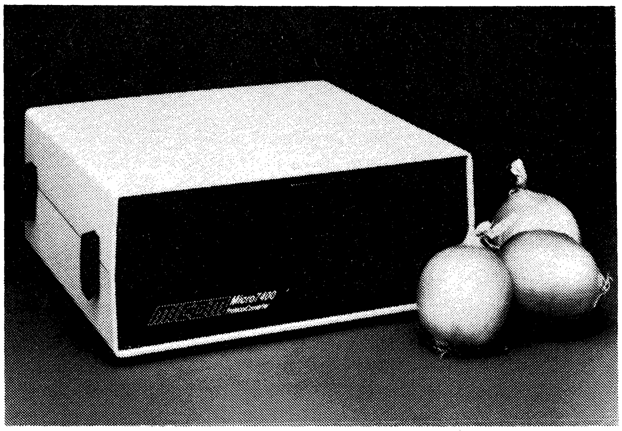
Asynchronous-to-synchronous converters are an older type of equipment, used mostly in applications that require conversion of asynchronous terminals for use on synchronous lines. In most newer conversion units, asynchronous-to-synchronous conversion is included along with other translation functions.

Protocol Converters

Protocol converters, one of the largest categories of conversion devices, handle changes that must occur on the Data Link Layer to ensure device compatibility. Protocol converters, or protocol conversion processors, as they are sometimes called, typically connect some type of incompatible peripheral device to a host. Protocol converters are microprocessor-based machines that usually communicate with the peripheral in a simple protocol and with the host in a more complex protocol that incorporates error-checking and retransmission capabilities. The converter communicates in the language of the peripheral and transforms

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices



Micom's Micro7400 protocol converter allows asynchronous display terminals, printer, teleprinter, and PCs to be used as IBM 3270-type terminals. It is a low-cost substitute for IBM's 3274 Model 61C Terminal Controller.

and reformats data received from that peripheral before relaying it to the host, or the reverse, thus acting as an intermediary between the host and the peripheral. The peripheral to which the protocol converter attaches can be a terminal, a plotter or display, a microcomputer, a minicomputer, or another host.

A protocol is a fixed set of rules that specifies the format of a data exchange. The rules govern the recognition of a connection with a remote point, the identification of the transmitting and receiving location, the transmission sequence, the handling of interruptions, methods of error checking and control, methods of blocking data, and security procedures.

Communications protocols cover a wide spectrum: they range from single character-by-character communications with no error checking to complex rules for moving large amounts of data among many devices.

The physical manifestation of the protocol is a series of characters in bit combinations that are appended to each block or frame of transmitted data. Through hardware or software, the sending device automatically formats the data and adds the required bits before transmitting each block. The receiving device automatically checks each of the appended bits before signalling an acknowledgement that data has been received. If any established condition is not met, the protocol initiates error control procedures.

Data communications protocols are either bit-oriented or byte-oriented. Byte-oriented protocols require that data be transmitted in eight-bit blocks; an acknowledgement is required after each transmitted block before the next can be sent. Bit-oriented protocols allow data to be transmitted in blocks of any length up to a specified maximum; an acknowledgement may take place after one or several blocks have been sent, depending on the protocol. Some of the most common protocols are as follows:

- ASCII—an asynchronous protocol with very little error checking. Transmission is in the form of a start bit, a number of data bits (usually five to eight), and one or more stop bits. Data in ASCII protocol can enter the communications line at any time; the end of the link is synchronized through the specifications of a common line speed and detection of the start bits and the beginning of the character transmission. ASCII requires an acknowledgement after each block is sent. ASCII protocol is often referred to as Teletype (TTY) protocol, since it is traditionally associated with teletypewriter equipment and services.

- IBM's SDLC—a bit-oriented protocol that uses a synchronized series of frames. Each frame has a synchronization flag, followed by an address field, a control field that tells the purpose of the transmission, the data itself, then a frame-check field, and finally a trailing flag. The flag character is used to achieve the synchronization. SDLC permits up to 127 frames to be outstanding before an acknowledgement is required.

- IBM BSC—a character-oriented protocol. Binary synchronous data and control characters consist of eight-bit bytes. A transmission in BSC consists of a number of synchronizing (SYN) characters that ensure synchronizing at both ends of the communications link. These are followed by a start-of-text (STX) character, an eight-bit block of text, an end-of-text (ETX) character, and a block error-checking character (BCC). BSC lacks the capability to handle full-duplex data, and does not comply with IBM's System Network Architecture (SNA) concept. Each block must be acknowledged before the next can be sent.

Other communications protocols include HDLC (High Level Data Link Control), a bit-oriented protocol; Univac U200, CDC 200UT and Burroughs Multipoint Poll Select, which are similar to IBM BSC but can run on both synchronous and asynchronous links; and DEC's DDCMP (Digital Data Communications Message Protocol), a byte-oriented protocol that can handle up to 255 unacknowledged transmissions.

A protocol converter actually changes one protocol to another by stripping the data down and rewrapping it according to the rules of a new set of specifications. Although hardware specifications differ from vendor to vendor, protocol converters usually contain a microprocessor, a realtime clock, two serial ports, associated data-rate generators, and the necessary firmware and RAM buffer.

During the conversion sequence, the protocol converter accepts blocks of data in one protocol, adds or deletes the necessary control characters, reformats the block, and calculates the required check characters so that the receiving device receives characters formatted according to its requirements. For example, in an ASCII-to-SDLC conversion, the converter will accept a string of characters, eliminate the start and stop bits, assemble the characters into a block, and add appropriate headers and trailers to create

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

complete frames. In a BSC-to-SDLC conversion, the converter must change the first four SYN bits of the Bisync algorithm to the first flag bit of the SDLC algorithm.

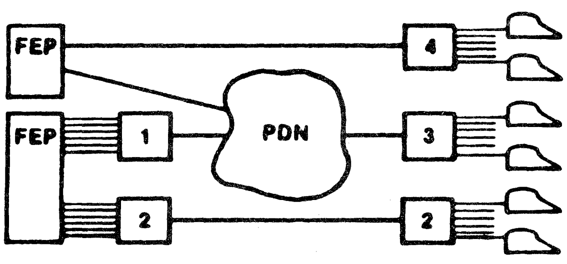
All protocol converters have some level of intermediate storage area to hold characters for conversion. Because of this buffering, a converter will always extend response time in the communications exchange. The device generally accepts low-speed input in the buffer, works with the data, and then transmits it out in short, high-speed bursts.

The data transmission method in the protocol converter differs from device to device. There are, however, some basic converter techniques. One of these techniques, called virtual protocol conversion, is used by a protocol converter that supports data transmissions up to 9600 bps. In virtual conversions, a central processor in the converter transforms each incoming datastream to its own protocol (the virtual protocol) and then reconverts the datastream to the protocol desired by the receiving device (the desired protocol).

An alternative technique uses a separate microprocessor to perform the conversion for each line interface that the device handles. The interface has approximately 12K of PROM in which a conversion program resides. Additional RAM (usually about 2K) holds the data from each line. A common memory module serves as a shared RAM buffer area, where input/output queuing takes place. Converted data goes to the shared area where it is transferred to the host in queue.

Besides pure protocol conversion, protocol converters often resolve related incompatibilities. For example, the converter might also translate ASCII code to EBCDIC or make several point-to-point links appear to the host as one multipoint link.

A special type of protocol converter is the Satellite Delay Compensation Unit (SDCU), which cuts propagation delay during satellite transmissions. Propagation delay is the amount of time between signal transmission over a circuit and acknowledgement of the transmission from the receiving end. Since the propagation delay during a satellite transmission is about a quarter of a second, this send-and-wait procedure can be quite time-consuming when every



1 = Multi-PAD as host computer concentrator to network.
2 = Pair of PADs in statistical TDM configuration.
3 = Multi-PAD as terminal concentrator to network.
4 = Multi-PAD as terminal concentrator to host or front-end processor.

Figure 4. Typical Configurations for Dynatech's Multi-PAD.25.

block requires acknowledgement before the next can be sent—as required by certain protocols, such as IBM BSC. The SDCU, which connects between the terminal and the modem, converts BSC into a specially conditioned form of SDLC that does not require an acknowledgement after each block. The end result is nearly 100 percent efficiency when transmitting in batch or message mode.

Gateway Devices, PADs, and FEPs

These products handle conversions on OSI Layers Three and Four (the Network and Transport layers, respectively) as well as performing lower layer functions. Very few conversion units handle translations on the Session Layer, although one vendor, ProtoComm, offers a PAD that does perform conversions at this level.

Gateway devices are products that provide access between incompatible networks, for example, between SNA and DECnet, or between SNA and Ethernet, or between a data communications device and an X.25 public data network. Gateway products provide compatibility between network architectures, with their inherent protocols, codes, and interfaces. Gateway converters may link specific devices with one another like protocol converters do or they may link two complete, but mutually exclusive systems, such as a minicomputer and an IBM mainframe, each with its own complement of peripherals.

By far the largest subset of gateway products are packet assembler/disassemblers (PADs). These devices permit host computers and peripheral equipment that use a communications protocol other than X.25 to be interconnected via a public data network. On the terminal side, most PADs support the connection of several devices, which can be terminals, CPU ports, printers, and so forth. On the network side, a high-speed port usually provides a link to the X.25 network. PADs usually perform concentrating and multiplexing functions as well as protocol conversion.

Most PAD products actually adapt a protocol rather than change it completely. The adaptation allows data in one protocol to pass through a network that uses another protocol. The transmitting PAD receives messages from the host or peripheral in the protocol of the sending device, converts and packetizes the information according to X.25 standards, and sends the packet through the network. At the receiving end of the X.25 link, another PAD performs error checking, disassembles the packets and converts messages back to the native protocol. Some PADs can also perform true protocol conversion between the sending device and the destination device, when necessary.

In normal operations, the use of the PAD and the X.25 network are transparent to both the sending and the receiving devices. However, for test purposes, the PAD can be made to poll and to present status information to the host. Some PADs also have a supervisory port so that users can configure the PAD's operating parameters and even diagnose network problems through the PAD.

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

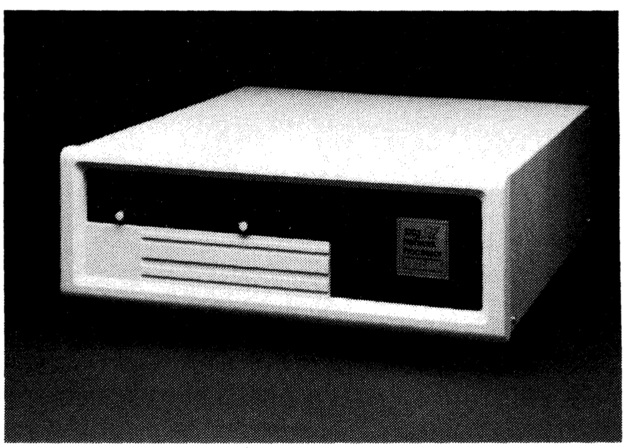
All About Conversion Systems and Emulation Devices

In Figure 3, we see a typical set of configurations for Dynatech's Multi-PAD.25. As the diagram shows, users can configure PADs to work as concentrators for the host computer, as statistical time division multiplexers, as terminal concentrators for the public data network, and as terminal concentrators to a host or FEP.

One of the trends in gateway conversion is interconnection between incompatible systems and peripherals through a PBX. For example, InteCom's IBX provides conversion between ASCII and 3270 protocols, and Rolm's CBX provides a gateway to IBM networks. Interfacing to X.25 networks and compatibility with specified local area networks (e.g., Ethernet) are also sometimes supported. Other PBX vendors are now including gateway conversion functions in their products.

Conversion can also occur in front-end processors (FEPs). These units achieve protocol conversion through software and hardware combinations. The FEP handles conversions for ASCII, BSC, SDLC, and X.25. How FEPs handle conversions varies widely, and most processors provide a variety of conversion functions.

Conversion can also occur in host-independent network processors. These devices usually rely on a microprocessor-based architecture to perform multiple functions. For example, Tri-data's Netway processor handles protocol conversion between dissimilar devices, serves as a concentrator for attached workstations, and manages communications among dissimilar hosts. The product also works as an X.25 packet processor to allow ASCII terminals to communicate with a host through X.25 networks. Netway processors allow many different hosts and workstations to communicate with one another in a network. The protocol translation capabilities of the device let users configure networks that include products from various vendors, including IBM, Burroughs, and DEC.



Netlink's 3703 Network Processor is a multifunction controller that concentrates lines using various protocols at full duplex transmission speeds up to 9600 bps and provides single-ended concentration of non-SNA terminals. Up to 12 non-SNA devices can be connected to the 3703.

These communications processors cannot be specifically classified as converters because they handle several other high-level functions in a data communications network. These products do not exist primarily to provide conversion functions. For more information on these devices, consult Tab C13 of Volume 1 of *Datapro Reports on Data Communications.*

Some vendors include protocol conversion functions on their minicomputers. Data General, for example, provides an architecture for its Eclipse units to handle extensive protocol conversions. Other vendors provide conversion software packages for their minicomputers.

At present, very few vendors offer products that handle conversion on the Session Layer. Protocom Devices does provide a P2500 PAD that supports Session Layer conversions to provide network security, simultaneous dual sessions, operation in Data Streaming/Turbo Mode, and error handling. The P2500 protects an organization from unauthorized network access via random password generation and permits only authorized terminal-connected PADs to access preassigned host-connected PADs. P2500 also permits some connected terminals to engage more than one host at a time. Turbo Mode operation on the P2500 decreases queuing delays that occur during transmission of large messages. The P2500 uses an inter-PAD block-check sequence, local end-to-end acknowledgements, and data retransmission to provide efficient error-handling functions.

Emulation Devices

Devices that handle conversions on the Presentation Layer provide the capability for one device to appear as another device. While protocol converters handle incompatibility problems between sets of rules particular devices use to communicate information, an emulator must handle incompatibilities in all specification differences between sending and receiving units—including differences in protocol, code, interface, device characteristics, and link characteristics. To the emulator, protocol conversion is secondary: while the protocol converter actually strips down data and rewraps it according to a new set of rules, the emulator reads the text in a whole message and emulates that text to the specifications of a different device.

A great many protocol converters on the market today provide both protocol conversion and emulation. Often vendors call protocol/emulation products protocol converters, although this nomenclature is somewhat inaccurate. All emulation devices provide protocol conversion, but not all protocol converters provide emulation. Most often, however, devices that handle protocol and emulation translations are called value-added terminal controllers, remote cluster controllers, or terminal emulators.

To use information in a transmission, a receiving device—whether a host or a terminal—must interpret data in the context of the device that it supports. Device specifications impose many constraints on the data communications

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

protocol that the device handles. This means that although a host and a terminal might operate in the same protocol, they might not be compatible with one another.

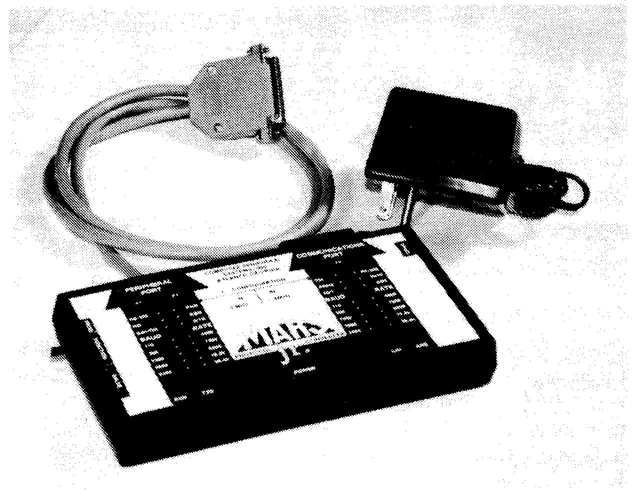
The unit that connects device-incompatible equipment must reformat data to offset restrictions imposed by an emulated device. Restrictions can include differences in record size and blocking characteristics, or they might relate to functional differences between equipment types. Most terminal emulators are not general-purpose units: they convert only between specific types of devices.

The way a terminal emulator handles conversions depends upon the specific characteristics of the emulated and emulating devices. Thus, describing a general emulation technique is difficult. But an example of how a terminal emulator takes an asynchronous datastream and converts it to the protocol and format used by an IBM 3271 terminal controller illustrates a basic conversion sequence.

An IBM 3271 serves up to 32 IBM 3277-type terminals on a multipoint line. Data moving in this type of configuration is blocked out in 1920-character screen images (blocks of data). If a user wants to replace IBM 3277 terminals with asynchronous ASCII devices, the ASCII units must appear as IBM 3277s to the IBM host. A terminal controller/emulator, or terminal controller as it is often called, can handle this problem by taking an asynchronous datastream into its buffer and keeping it there until a 1920-character screen image is filled or until the emulator receives an end-of-record, end-of-block control character. The terminal controller converts the protocol of the ASCII terminal to the protocol of the host (i.e., BSC), rearranges the data format to appear as if it comes from an IBM 3271, and then transfers the screen image to the host, which recognizes the data as that of an IBM 3277—not an asynchronous ASCII terminal. The terminal controller performs all functions of the device it replaces, including data concentration, poll/select, flow control, buffering, error detection and correction, and interfacing of multiple attached terminals. For example, Icot's Virtual Terminal controllers emulate an IBM 3271 or 3274 controller and provide ASCII terminal-to-IBM 3277/3278/3279 terminal emulation and IBM 3284 printer emulation.

Sometimes the emulating device connects to an IBM cluster controller rather than replacing it, in effect, performing the conversion between the terminal and the IBM controller instead of between the controller and the host. The purpose of these emulators is to allow the user to integrate incompatible equipment into an existing terminal cluster. Local Data's Interlynx, for example, attaches to the IBM 3274 or 3276 controller to provide protocol and emulation translations that allow ASCII terminals to replace IBM 3278 or 3279 terminals.

During an emulation/conversion/transfer sequence, the emulator must interpret control sequences from an attached terminal to simulate the operations of the emulated terminal. The equivalents for a specified control sequence between one terminal model and another model vary widely. For example, no asychronous ASCII keyboard provides all of the special 3270 function keys, and those that are provided are generally encoded differently by different devices. Functions like erasing a screen, setting cursor address, and so forth are also encoded differently. As commands arrive, the emulator must translate the sequence and operate upon it according to the equivalent function of the emulated device. The emulator unit then updates its internal buffers and the display screen of the attached terminal according to the control sequence it receives and translates.



The MARS, jr. from Computer Peripheral Systems, Inc. converts any RS-232-C or Parallel device to Burroughs Poll/Select.

One of the biggest problems users face when using terminal emulation products concerns the special keystrokes an operator must learn to produce capabilities not normally supported on a particular terminal. Terminal operators accustomed to the keystrokes of a particular terminal must learn a new set of keystrokes to effect the functions of the emulated terminal. This operation can be compared to typing in Arabic on a typewriter with an English keyboard and an Arabic font. (Type a "g" and another symbol appears on the paper.) Because this kind of operation can cause confusion, vendors usually provide key maps that show keystroke equivalents between the emulated terminal and the various emulating devices. Some vendors also provide stick-on decals for emulating keyboards.

Many users are purchasing these terminal controllers to allow non-IBM devices in remote locations to access IBM mainframes. Remote cluster controllers eliminate the need to dedicate one terminal (e.g., a 3270) to one application, and another terminal at the same site to a local minicomputer. Many remote controllers have one synchronous line for 3270 access and two or more minicomputer interfaces. Terminals attached to the controller can switch between a remote host mainframe and the remote and local minicomputers in this type of configuration.

Users can configure most terminal controllers for dial-up access, allowing ASCII terminals in a remote location to dial into the local controller, which then makes the connec-

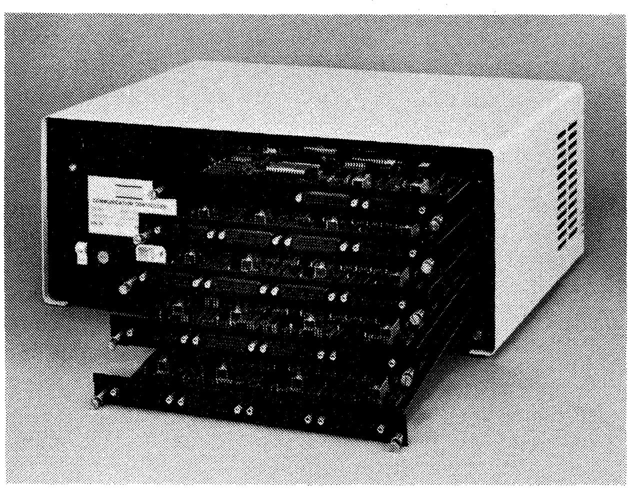
© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

tion with a CPU that is located at the same or a third site. The controller eliminates the need for an IBM controller and additional synchronous lines to access the mainframe. A prominent cluster controller vendor, Datastream Communications, Inc., offers several models, including the 774 and the 776. The Model 776 operates in a point-to-point, multipoint or switched BSC network and acts as, and replaces, an IBM 3271/3276 cluster controller.

Units that handle conversions to make microcomputers and personal computers compatible with IBM mainframes represent a large and growing area in the conversion/emulation marketplace. Organizations are using more and more microcomputers for decentralized applications, but in many instances microcomputer users must have access to a centralized database, which generally resides on an IBM mainframe. Users can establish a micro-to-mainframe link through an emulation package that typically includes a diskette containing the emulation logic and a communications circuit board that is installed inside the microcomputer. An example of this type of product is DCA's Irma, one of the most popular micro-to-mainframe interfaces. The Irma is an IBM PC board with a coaxial interface that connects the PC to an IBM 3270 terminal controller that accesses a mainframe. With Irma installed and running on the PC, users can download data from the mainframe to the microcomputer, where it is viewed on the microcomputer screen. Like other forms of emulation, micro-to-mainframe links usually specify the microcomputers supported and the host ports and/or peripherals to which they can be connected. The Irma, for example, must attach to an IBM Personal Computer or compatible microcomputer and will attach only to an IBM or compatible 3274/3276 terminal controller. Other emulators provide IBM 2780/3780 batch terminal emulation for specified micro- and minicomputers.

Other types of emulation products offer conversions that are the reverse of the popular ASCII-to-3270 conversion.



The Perle PDS 350/525 protocol converter is a diskette-based unit that permits revisions to be easily installed in the field. It supports up to 16 asynchronous ports.

For example, Protocol Computers' 74D unit lets an IBM 3270-type device talk to an ASCII host. The 74D interfaces with an IBM communications controller, an IBM 3270 cluster controller, and up to six ASCII hosts, which can be DEC hosts, public networks, and personal computers. Supported by the 74D, the 3270 can switch between SDLC and DEC hosts.

Conversion and emulation in a data communications network can occur in many different devices and at many points in a network. Converters can be separate hardware units placed between a terminal and a modem; shared hardware units that handle other functions (e.g., front-end processors); devices that replace cluster controllers; interface cards in a personal computer; or applications programs, specialized emulation software packages or software/hardware resident on minicomputers, mainframes, and PBX systems. Many network services, such as Tymnet and GTE Telenet, also offer conversion as part of their value-added products.

Protocol conversion and emulation products address problems of incompatibility among many types of data communications devices. But as you might have surmised from our discussion above, the majority of conversion units are designed specifically to incorporate incompatible devices in an IBM environment. In the next section of this report, we will discuss that environment in relation to conversion and emulation products.

The IBM 327X Environment

Tremendous growth in the minicomputer, microcomputer, and personal computer markets has led to a rapid increase in the number of installed ASCII asynchronous terminals that access these computers. However, ASCII devices cannot access information that resides on IBM mainframes. IBM's series of products that provide interactive communications in an IBM network is the IBM 3270 Information Display System. This series includes controllers, terminals, and printers that are dedicated to a single host and usually to a single application.

Components in the current 3270 system include: the 3278, 3279, 3178, 3179, 3180, and 3290 display terminals, 3287, and 3289 printers, and the 3274 and 3276 cluster controllers. Each component comes in various models. For example, the 3278 is a monochromatic display available in five models that essentially differ only in their screen capacities. The 3279 is a color display version of the 3278. The 3274 controller comes in various models that handle up to 12, 16, or 32 attached terminals, local or remote host connection, and BSC or SDLC protocol. The 3276 is a smaller controller designed for clusters of up to eight terminals.

Because of the 3270's huge installed base, many models that are no longer actively marketed by IBM continue to play a significant role in the IBM-compatible markets, particularly the 3277 display terminal, and the 3271 and 3272 controllers. The 3271 is a remote cluster controller that handles up to 32 terminals and comes in BSC and SDLC versions. The 3272 is a local channel-attached version of the 3271.

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

There are some shortcomings to using products in the 3270 family. First, they are more expensive than ASCII terminals. Second, many of the IBM components are physically larger and take up more space than the ASCII terminals and the emulators that can be used in their place. (IBM has reduced both the price and size of its newer 3270 components, effectively eliminating these shortcomings.)

To acknowledge the need for asynchronous communication, in 1979 IBM introduced a Model 3101 terminal, which can attach directly to a 3705 communications controller and participate in ASCII applications resident in the host. The company also introduced a four-port protocol converter, the 7426, to allow the 3101 to appear as a 3278 to the 8100 and 4300 Series computers. A Yale ASCII Communications System software package lets almost any ASCII device access 3270 applications and appear as a 3270 terminal. Most importantly, however, IBM introduced 3270 emulation support for most of its mini- and microprocessor-based products including the IBM PC, the System/36, and the Displaywriter. IBM also introduced the 3270-PC, a version of the PC that is designed specifically for use in a 3270 system. In doing so, IBM, in effect, changed the 3270 from a single-host, dedicated terminal system to a system that can accommodate many different devices.

Although the majority of protocol converters and terminal controllers on the market today handle some type of conversion between ASCII devices and IBM units, other products handle conversion between IBM BSC protocols and IBM SDLC protocols. This conversion is particularly useful to users of older IBM BSC equipment who wish to migrate to an SNA/SDLC environment without replacing all of their old equipment. BSC-to-SDLC conversions generally occur between BSC 2780/3780 RJE or 3270 BSC protocols and SDLC protocols.

As IBM PCs become increasingly prevalent in organizations, products to provide micro-to-mainframe compatibility will become more and more important. The entire protocol/emulation market is exploding today because units that make ASCII terminals and personal computers compatible with SNA/SDLC networks are in tremendous demand. ASCII devices provide flexible and inexpensive solutions to network problems, but IBM's mainframes are still the de facto standard for centralized computer facilities that must handle large databases and many applications. It seems unlikely that this situation will change soon, and vendors that offer conversion products to handle ASCII-to-IBM conversions should continue to enjoy a healthy market share for their products.

CURRENT TRENDS

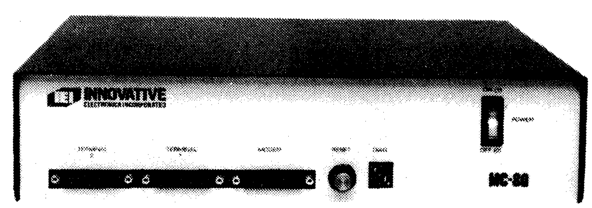
Many different products handle some type of conversion to provide compatibility between communications devices. Presently, a number of large data communications equipment vendors are incorporating protocol converters and terminal controllers into their general line of products. Micom Systems, for example, acquired Industrial Computer Controls, Inc., one of the oldest specialized protocol converter manufacturers in the marketplace. At the same time, Micom introduced the Micro7400 protocol conversion, a replacement of ICCI's CA20 converter. The Micro7400 handles ASCII-to-3270 conversion and provides network monitoring and control functions as well.

Formerly, the venue of small companies like Protocol Computers, Inc. and Innovative Electronics, which specialize in standalone protocol converter products, protocol conversion has become incorporated into existing data communications products. We now find conversion as an integral capability in digital data switches, PBXs, personal computers, and word processors. And a market that in 1980 earned gross yearly revenues of about $5 million mushroomed into a $100 million a year business in 1983. Overnight, protocol converters have become the "hot" product in the ever-changing data communications environment.

From a historical perspective, we can benchmark interest in protocol conversion at IBM's introduction of its 7426 converter in October 1982. With this announcement, IBM not only sanctioned conversion technology as a viable solution to network problems, but also focused industry attention on the technology.

Conversion products that facilitate LAN-to-LAN compatibility and access X.25 public networks are also expected to have a large market. We can expect to see a growing interest in PAD devices that effect X.25 access, and we can anticipate greater PBX conversion capabilities in the months ahead. Conversion offerings from value-added carriers, such as Tymnet and GTE Telenet, and from the BOCs will also grow as data communications moves further into the home markets where personal computer users are becoming more interested in linking into public networks and databases.

Until the data communications industry adapts and uses worldwide protocol standards to link equipment, protocol converters and emulators will remain an important part of the general market. It is unlikely that such standardization will occur in the very near future.



Innovative Electronics' MC-80 Series protocol converters include many models that provide various conversion: IBM 2780/3780 to ASCII, Burroughs Poll/Select to ASCII, IBM 3270 to Burroughs Poll/Select, and others.

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

CHOOSING CONVERSION DEVICES

Before choosing a conversion unit, users should consider some of the negative characteristics of the devices. First, protocol converters will cause delays in response time on the network because data must flow into the converter's buffer before transmission. If data backs up in the buffer, overruns occur; if the buffer is small, the converter can lose data.

Terminal operators dealing with devices emulating other products may have problems learning the new key sequences and key functions necessary to the emulation process. Thus, organizations can expect some decreased productivity during the initial months of a conversion installation.

In addition, protocol converters usually do not offer the security provided by, for example, the IBM 3270-type devices. Organizations must deal with the problem of protecting sensitive data, particularly in dial-up applications.

When an organization decides to install conversion products in a data communications network, it should determine exactly what kinds of conversions are needed to solve particular incompatibilities. Once this is established, users should determine which kind of products can handle the conversion most effectively in a particular application. This can be an extremely confusing task because there are so many conversion products available. To narrow choices, it is wise to contact many vendors and ask for product specifications and documentation that explains how a product operates. When studying specifications and operating procedures, users must note exactly what types of terminals, controllers, or hosts are supported by the device because most converters and controllers support specific products rather than a general range of devices. For example, a protocol converter specifically designed for IBM 3277 emulation might not work with a 3278 application.

Also important is finding out what added features and functions the converter handles. Does it support more than one host? Does it replace an IBM controller, or is it used in conjunction with a controller? Does the device incorporate any multiplexing or concentrating? Can the network manager monitor the network via the converter? If additional features are available, are they standard or optional? What cost savings will it represent in your overall networking scheme?

After narrowing their choices to the products of several vendors, users should ask the company to provide an in-house demonstration of the product. A prospective buyer should also request a list of current users who will discuss their experiences with the product. These individuals can provide information about the advantages and disadvantages of the product, hardware reliability, and the type and quality of support provided by the vendor.

IBM mainframe users in particular should find out whether conversion equipment can be upgraded as IBM upgrades and changes its SNA architecture.

After further narrowing the selection to two or three vendors, users should request a free trial of the product. By using a converter in a particular application, prospective buyers can soon find out whether a product provides the desired compatibility in an efficient manner.

USER EXPERIENCE

In November 1984, Datapro conducted a Terminal Users Survey, which was based on results received from questionnaires mailed to a cross section of *Data Communications* magazine subscribers. Several questions in the survey pertained to the use of protocol conversion devices in a network. Below we show the results of these questions.

METHODOLOGY

The questionnaire was designed and produced by Datapro's senior data communications editors, and mailed to 15,000 subscribers of Data Communications magazine. These subscribers were identified as domestic end users of data communications equipment. The subscribers were asked to fill out the forms to provide ratings and other information for conversion systems and devices. They were then asked to return the completed form by the cut-off date of December 10, 1984; approximately 1,000 completed forms were received by Datapro.

When Datapro received the returns, they were edited by senior level editors. All forms were examined for validity before being sent for tabulation. Subscriber names and addresses were used for initial validation and identification. In addition, responses to the survey were disqualified whenever a vendor/model identity was omitted, user ratings were not assigned, an obvious vested interest on the part of the respondent was judged to exist, or incomprehensible or unreasonable answers were given.

After the forms were processed by Datapro personnel, they were shipped to *Mathematica Policy Research, Inc. (MPR)*, of Princeton, NJ, for key entry and computer tabulation. Summary information was prepared in the form of totals, percentages, or weighted averages as appropriate for each question.

Weighted averages were used to determine the ratings given by the users of the equipment. These were computed in the following manner: "Excellent" was weighted as 4, "Good" was weighted as 3, "Fair" was weighted as 2, and "Poor" was weighted as 1. The tallied numbers for each value were then multiplied by the corresponding weight, and the average taken by dividing the sum of the products by the total number of responses for that category.

Users were asked to provide information on *all* conversion systems and devices that they were using. This resulted in multiple responses from the users. The responses were identified first by manufacturer, and then by model. A

All About Conversion Systems and Emulation Devices

Table 1. User Ratings for Conversion Systems and Emulation Devices

Manufacturer/ Model	No. of User Responses	No. of Units Installed	Overall Performance					Ease of Installation					Ease of Operation					Device Reliability					Maintenance Service/Technical Support					Would You Recommend This System?		
			WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	Yes	No	Undecided
Alanthus—																														
CX-80	5	13	3.8	4	1	0	0	3.0	2	1	2	0	3.4	2	3	0	0	3.8	4	1	0	0	3.2	2	2	1	0	4	0	0
Other/ Unspecified	2	4	—	2	0	0	0	—	2	0	0	0	—	1	0	1	0	—	2	0	0	0	—	1	1	0	0	2	0	0
Subtotal	7	17	3.9	6	1	0	0	3.3	4	1	2	0	3.3	3	3	1	0	3.9	6	1	0	0	3.3	3	3	1	0	6	0	0
Amdahl—																														
Unspecified	5	1453	3.4	2	3	0	0	3.4	2	3	0	0	3.4	2	3	0	0	3.4	2	3	0	0	3.2	1	4	0	0	4	0	0
Subtotal	5	1453	3.4	2	3	0	0	3.4	2	3	0	0	3.4	2	3	0	0	3.4	2	3	0	0	3.2	1	4	0	0	4	0	0
Ark—																														
Unspecified	4	925	3.5	2	2	0	0	3.5	2	2	0	0	3.5	2	2	0	0	3.5	2	2	0	0	3.5	2	2	0	0	3	0	1
Subtotal	4	925	3.5	2	2	0	0	3.5	2	2	0	0	3.5	2	2	0	0	3.5	2	2	0	0	3.5	2	2	0	0	3	0	1
Avatar Technologies—																														
PA1000	7	60	3.6	4	3	0	0	3.0	2	3	2	0	2.9	2	2	3	0	3.4	3	4	0	0	3.3	2	5	0	0	7	0	0
Subtotal	7	60	3.6	4	3	0	0	3.0	2	3	2	0	2.9	2	2	3	0	3.4	3	4	0	0	3.3	2	5	0	0	7	0	0
Black Box Catalog—																														
Unspecified	8	38	3.6	5	3	0	0	3.3	3	4	1	0	3.8	4	4	0	0	3.8	6	2	0	0	3.5	5	2	1	0	8	0	0
Subtotal	8	38	3.6	5	3	0	0	3.3	3	4	1	0	3.8	4	4	0	0	3.8	6	2	0	0	3.5	5	2	1	0	8	0	0
Commtex—																														
CX-80	5	72	3.4	2	3	0	0	3.4	2	3	0	0	3.6	3	2	0	0	3.6	3	2	0	0	3.0	2	1	2	0	5	0	0
Subtotal	5	72	3.4	2	3	0	0	3.4	2	3	0	0	3.6	3	2	0	0	3.6	3	2	0	0	3.0	2	1	2	0	5	0	0
Data General—																														
Unspecified	3	9	3.7	2	1	0	0	3.3	1	2	0	0	4.0	3	0	0	0	3.0	0	3	0	0	2.0	0	0	3	0	0	0	3
Subtotal	3	9	3.7	2	1	0	0	3.3	1	2	0	0	4.0	3	0	0	0	3.0	0	3	0	0	2.0	0	0	3	0	0	0	3
Datastream—																														
T7	9	15	2.8	0	7	2	0	2.4	0	4	5	0	2.8	0	7	2	0	3.1	2	6	1	0	2.3	1	2	5	1	6	2	0
T8	3	7	3.3	1	2	0	0	2.7	0	2	1	0	3.0	0	3	0	0	3.0	1	1	1	0	3.0	0	3	0	0	2	0	0
774	5	9	2.8	0	4	1	0	2.6	0	3	2	0	2.8	0	4	1	0	3.0	1	3	1	0	2.8	1	3	0	1	4	1	0
874	6	47	3.8	5	1	0	0	3.2	2	2	1	0	3.6	3	2	0	0	3.6	3	2	0	0	3.4	3	1	1	0	5	0	1
Unspecified	6	10	2.8	2	2	1	1	2.8	1	3	2	0	2.7	0	4	2	0	2.8	2	2	1	1	2.2	0	3	1	2	5	1	0
Subtotal	29	88	3.1	8	16	4	1	2.7	3	14	11	0	2.9	3	20	5	0	3.1	9	14	4	1	2.6	5	12	7	4	22	4	1
DEC—																														
Unspecified	3	3	3.0	0	3	0	0	2.7	0	2	1	0	2.7	0	2	1	0	2.7	0	2	1	0	2.3	0	2	0	1	1	0	2
Subtotal	3	3	3.0	0	3	0	0	2.7	0	2	1	0	2.7	0	2	1	0	2.7	0	2	1	0	2.3	0	2	0	1	1	0	2
Digital Communications Associates (DCA)—																														
IRMA	5	6	3.2	2	2	1	0	3.2	2	2	1	0	3.0	2	1	2	0	3.6	3	2	0	0	3.0	2	0	2	0	4	0	1
Other/ Unspecified	3	15	3.7	2	1	0	0	3.3	1	2	0	0	3.3	1	2	0	0	3.7	2	1	0	0	3.7	2	1	0	0	3	0	0
Subtotal	8	21	3.4	4	3	1	0	3.3	3	4	1	0	3.1	3	3	2	0	3.6	5	3	0	0	3.3	4	1	2	0	7	0	1
Diversified Data Resources—																														
Hydra II	6	7	3.7	4	2	0	0	3.3	4	0	2	0	3.5	3	3	0	0	3.7	4	2	0	0	3.4	2	3	0	0	5	0	0
Subtotal	6	7	3.7	4	2	0	0	3.3	4	0	2	0	3.5	3	3	0	0	3.7	4	2	0	0	3.4	2	3	0	0	5	0	0
Dynapac—																														
Unspecified	4	135	3.3	2	1	1	0	4.0	4	0	0	0	4.0	4	0	0	0	2.8	1	1	2	0	3.5	3	0	1	0	3	0	0
Subtotal	4	135	3.3	2	1	1	0	4.0	4	0	0	0	4.0	4	0	0	0	2.8	1	1	2	0	3.5	3	0	1	0	3	0	0
Dynatech—																														
Unspecified	3	202	4.0	3	0	0	0	3.3	2	0	1	0	3.7	2	1	0	0	4.0	3	0	0	0	3.7	2	1	0	0	3	0	0
Subtotal	3	202	4.0	3	0	0	0	3.3	2	0	1	0	3.7	2	1	0	0	4.0	3	0	0	0	3.7	2	1	0	0	3	0	0
Gandalf Data—																														
Unspecified	3	3	3.5	1	1	0	0	2.3	0	2	0	1	3.5	1	1	0	0	4.0	2	0	0	0	4.0	2	0	0	0	2	1	0
Subtotal	3	3	3.5	1	1	0	0	2.3	0	2	0	1	3.5	1	1	0	0	4.0	2	0	0	0	4.0	2	0	0	0	2	1	0

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

Table 1. User Ratings for Conversion Systems and Emulation Devices (Continued)

Manufacturer/ Model	No. of User Responses	No. of Units Installed	Overall Performance					Ease of Installation					Ease of Operation					Device Reliability					Maintenance Service/Technical Support					Would You Recommend This System?		
			WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	WA	E	G	F	P	Yes	No	Undecided
GTE Telenet—																														
TP3010	4	214	3.3	1	3	0	0	3.3	1	3	0	0	3.0	1	2	1	0	3.0	0	4	0	0	3.0	1	2	1	0	4	0	0
TP4000	4	27	3.0	1	2	1	0	2.3	0	2	1	1	3.0	1	2	1	0	3.3	2	1	1	0	2.8	1	1	2	0	3	1	0
Subtotal	8	241	3.1	2	5	1	0	2.8	1	5	1	1	3.0	2	4	2	0	3.1	2	5	1	0	2.9	2	3	3	0	7	1	0
IBM—																														
Series 1	10	25	3.5	5	5	0	0	2.4	3	1	3	3	2.7	3	3	2	2	3.8	8	2	0	0	3.7	7	3	0	0	6	2	2
3274	5	18	3.2	1	4	0	0	3.0	1	3	1	0	3.2	1	4	0	0	3.0	1	3	1	0	2.8	1	3	0	1	5	0	0
Other/ Unspecified	8	31	2.9	4	1	1	2	2.8	2	2	4	0	3.1	4	2	1	1	2.8	2	4	0	2	2.6	2	3	1	2	4	2	2
Subtotal	23	74	3.2	10	10	1	2	2.7	6	6	8	3	3.0	8	9	3	3	3.3	11	9	1	3	3.1	10	9	1	3	15	4	4
ICOT Corporation—																														
352/362	4	4	3.0	0	4	0	0	2.5	0	2	2	0	2.5	0	2	2	0	2.5	0	2	2	0	1.8	0	1	1	2	2	0	2
Subtotal	4	4	3.0	0	4	0	0	2.5	0	2	2	0	2.5	0	2	2	0	2.5	0	2	2	0	1.8	0	1	1	2	2	0	2
Kaufman Data—																														
Unspecified	4	37	3.8	3	1	0	0	3.3	2	1	1	0	3.5	3	0	1	0	3.8	3	1	0	0	3.3	2	1	1	0	3	0	1
Subtotal	4	37	3.8	3	1	0	0	3.3	2	1	1	0	3.5	3	0	1	0	3.8	3	1	0	0	3.3	2	1	1	0	3	0	1
KMW Systems—																														
Unspecified	5	7	3.0	1	3	1	0	3.0	1	3	1	0	3.2	1	4	0	0	2.6	1	1	3	0	2.6	1	1	3	0	2	0	3
Subtotal	5	7	3.0	1	3	1	0	3.0	1	3	1	0	3.2	1	4	0	0	2.6	1	1	3	0	2.6	1	1	3	0	2	0	3
Local Data—																														
3274	10	33	3.0	1	8	1	0	2.9	2	4	3	0	3.0	2	5	2	0	2.8	0	7	2	0	2.6	0	6	2	1	6	1	1
3278	2	7	—	1	1	0	0	—	1	0	1	0	—	1	0	1	0	—	1	1	0	0	—	2	0	0	0	2	0	0
3287	2	22	—	1	1	0	0	—	1	1	0	0	—	1	1	0	0	—	1	1	0	0	—	0	2	0	0	0	0	1
Other/ Unspecified	4	4	2.8	0	3	1	0	2.8	1	1	2	0	2.8	0	3	1	0	2.8	0	3	1	0	2.3	0	1	3	0	3	1	0
Subtotal	18	66	3.1	3	13	2	0	2.9	5	6	6	0	3.0	4	9	4	0	2.9	2	12	3	0	2.7	2	9	5	1	11	2	2
Micom Systems—																														
Unspecified	13	72	3.5	7	5	1	0	3.1	7	3	3	0	3.2	6	4	3	0	3.3	5	7	1	0	2.9	4	4	5	0	10	1	2
Subtotal	13	72	3.5	7	5	1	0	3.1	7	3	3	0	3.2	6	4	3	0	3.3	5	7	1	0	2.9	4	4	5	0	10	1	2
Protocol Computers—																														
1051/151	4	4	3.0	1	1	1	0	3.0	1	2	1	0	3.3	1	3	0	0	3.3	2	0	1	0	3.0	1	1	1	0	2	0	2
1067/167	13	57	3.2	5	6	1	1	3.1	2	10	1	0	3.0	3	7	3	0	2.8	4	3	5	1	2.4	2	3	5	2	8	4	1
1071/171	9	32	2.6	0	6	2	1	2.9	0	8	1	0	2.8	0	7	2	0	2.4	0	5	1	2	2.1	0	4	0	3	6	2	1
1076/176	44	101	3.4	21	19	4	0	3.1	11	25	8	0	3.3	19	20	4	1	3.2	19	17	8	0	2.8	8	19	15	2	35	2	5
Other/ Unspecified	13	199	2.9	4	5	2	2	2.8	4	5	1	3	3.0	4	7	0	2	2.9	3	7	2	1	2.5	2	5	3	3	8	2	3
Subtotal	83	393	3.2	31	37	10	4	3.0	18	50	12	3	3.1	27	44	9	3	3.0	28	32	17	4	2.6	13	32	24	10	59	10	12
Renex Corporation—																														
Unspecified	10	24	3.0	4	3	2	1	3.1	4	3	1	1	2.9	2	5	1	1	3.1	3	5	0	1	2.7	2	2	2	1	7	0	2
Subtotal	10	24	3.0	4	3	2	1	3.1	4	3	1	1	2.9	2	5	1	1	3.1	3	5	0	1	2.7	2	2	2	1	7	0	2
Thomas Engineering—																														
Unspecified	6	119	2.8	1	3	2	0	2.2	1	1	2	2	2.7	2	2	0	2	3.0	3	1	1	1	2.2	1	4	0	1	4	1	0
Subtotal	6	119	2.8	1	3	2	0	2.2	1	1	2	2	2.7	2	2	0	2	3.0	3	1	1	1	2.2	1	4	0	1	4	1	0
All Others—	69	713	3.2	22	29	17	4	2.9	24	22	18	5	3.3	20	34	11	5	3.4	28	26	11	4	3.2	18	26	16	6	42	15	13
Subtotal	69	713	3.2	22	29	17	4	2.9	24	22	18	5	3.3	20	34	11	5	3.4	28	26	11	4	3.2	18	26	16	6	42	15	13
Grand Total	341	4783	3.2	129	155	43	12	3.0	101	141	77	16	3.1	110	163	48	14	3.2	132	140	47	13	2.8	88	124	82	29	238	39	49

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

minimum of three responses was required to break out the ratings for a specific manufacturer. Once the manufacturer was identified with the proper number of responses, a minimum of three responses was again required to break out the ratings for a specific model.

We asked the users to identify any types of protocol conversion that were being performed for their applications. Below are the responses of the users who answered this question:

	Number of Responses	Percent of Total Responses
ASCII-to-BSC	118	36
ASCII-to-SDLC	123	38
BSC-to-SDLC	26	8
Other	57	18

... and by what means this protocol conversion is performed:

	Number of Responses	Percent of Total Responses
Software loaded onto an existing system such as a general-purpose computer, front-end processor, terminal controller, or PBX system	51	17
Dedicated protocol converter	213	72
Value-added network service (e.g., GTE Telenet)	25	8
Other	10	3

The strength of the protocol conversion market that has emerged in the past three years is certainly confirmed by these users' responses. The heavy use of ASCII-to-BSC and ASCII-to-SDLC conversion seems to bear out the conclusions concerning the increased use of asynchronous terminals we noted earlier in this report. Most likely, a great many of the ASCII-to-BSC conversions involve ASCII terminals emulating 3270-type devices.

Datapro strongly suggests that the reader use the information presented with discretion. The ratings are not intended as a statistically accurate indicator of the capabilities of a device. Rather, the ratings and other information should be used as guides to potential strengths and weaknesses of that device. The responses also may be examined to provide an indication of a manufacturer's share of the market. Any equipment acquisition decision should be made only after further investigation on the part of the buyer.

COMPARISON CHARTS

The charts that accompany this report present listings of the key characteristics of approximately 93 protocol converters or terminal emulators, 39 X.25 PADS, and 20 code, speed, interface, and async/sync converters. This information was supplied by the vendors during February and March of 1985. Datapro wishes to thank the participating vendors for their cooperation.

Datapro sent repeated requests for information to 70 firms known or believed to manufacture some type of hardware conversion device. *The absence of any company from the charts means that the company either failed to respond to our request by the survey deadline, was unknown to us, did not make a hardware conversion product, or chose not to be listed.*

Many companies presently manufacture conversion devices for microcomputer-to-mainframe communication. These products consist of a circuit board that is inserted into the microcomputer and accompanying software. These devices, predominanty designed to effect IBM PC-to-mainframe connections, comprise a large and growing segment of the conversion marketplace and are covered in a separate report in *Datapro Reports on Data Communications* behind the Microcomputer Communications (C22) tab in Volume 2.

KEY TO CONVERSION EQUIPMENT TABLE ENTRIES

For the reader's convenience, we have organized the comparison charts into three broad categories. **Conversion Systems/ Terminal Controllers** can be a variety of devices, including protocol converters, terminal emulators, remote cluster controllers, and terminal controllers. Basically, devices in this section provide conversion from one protocol to another and/ or allow one device, e.g., an asynchronous ASCII terminal, to act as another type of device, e.g., an IBM 3270 terminal, in a network. **X.25 Packet Assemblers/Disassemblers (PADS)** convert equipment using a non-X.25 protocol to the X.25 protocol for transmission over public and private networks. PADs also perform other functions, including concentrating or multiplexing. **Code, Speed, Interface, and Async/Sync Converters** include a number of miscellaneous devices that handle conversions from one code, interface, speed, or synchronization to another. These are generally less sophisticated devices than those represented in the other two categories.

The following text briefly describes the chart entries in the order in which they appear in the charts. Not all of the characteristics listed appear in all of the charts because some entries do not apply to all categories. For example, "link-level framing support" is relevant only to X.25 PADs.

General Characteristics

Manufacturer and model. We list here the manufacturer and exact model number or name of each device.

Device type. In the Conversion Systems/Terminal Controllers section, we have asked the manufacturer to specify whether the device is a protocol converter, terminal emulator, code or interface converter, and so forth.

Conversion performed. All converters perform some type of translation from one code, speed, or protocol to another. The most common conversion is asynchronous ASCII to IBM SNA/SDLC or BSC, but a number of other translations are available on the units represented in the tables.

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

KEY TO CONVERSION EQUIPMENT TABLE ENTRIES (Continued)

Specific device emulated. In many cases, conversion devices provide the means to convert the text format of one type of device into the characteristics and format of another device. This translation, called emulation, is indicated, if available. Note that most protocol converters also provide device emulation.

Specific functionality provided. Most converters allow one device to be used as another type of device in the network. For example, a number of units allow asynchronous ASCII CRTs to be used as IBM 3277 Models 1 and 2. It is very important for prospective users to know exactly what type of device can be emulated by a given converter because many converters are designed for a specific application.

Virtual screen sizes supported. For a device to provide emulation, it must support the screen size, in characters, of the emulated device. For example, a device emulating an IBM 3270 terminal must support a 1920-character screen.

Command port support. Some converters support a port through which users can select operating parameters and monitor, diagnose, and control the network. A "yes" answer indicates that the device does have a command port.

Network certification. X.25 PADs allow devices to interface with public and private networks. These networks must certify use of the device on their facilities. Prominent network providers include GTE Telenet, Tymnet, and Uninet in the U.S., and Datapac and Infoswitch in Canada.

CCITT recommendation supported. CCITT has specified certain protocols for devices operating on an X.25 network. Most X.25 PADs support all CCITT recommendations, including X.25 Levels 1, 2, 3; X.3; X.28; and X.29. X.25 defines the operating protocol of the packet network. X.3 defines a set of 18 parameters that govern operation of the PAD and each asynchronous terminal port. X.28 defines the interface between the asynchronous terminal and the PAD, and X.29 defines the format of the packets that carry control information from the PAD to the remote destination.

Buffer memory capacity. PAD devices contain a buffer memory that holds packets before transmission. Software is generally held in ROM.

Additional RAM available. Many PADs can accommodate the additional RAM necessary to expand the capacity of the device.

Device configuration. Most X.25 PADs can be configured to act as other types of devices in the network. Typical configurations include host concentration, time division multiplexing, or front-end processing.

Host Side/Network Channel Specifications

Specific hosts supported. Conversion devices generally support IBM or compatible hosts or asynchronous hosts such as DEC VAX or PDP 11. In this entry, we have listed the type of mainframe with which the converter operates.

Number and type of host lines supported. Most converters support one line to a host; that line can be one IBM SNA/SDLC or BSC line, or one line to an asynchronous host. Some devices do support more than one host connection. This connection may be a dual IBM BSC or SDLC link, or one IBM link and one link to an asynchronous host.

Number of host sessions supported concurrently. If a converter supports more than one host line, the device may support both connected hosts concurrently, or separately through a switch selection.

Connections supported. Conversion devices support direct connections, multipoint, and/or point-to-point connections. Most converters support more than one type of connection, and most support all three.

Connection to host via controller. While some conversion devices emulate a controller, others must connect to a controller in the network. We have asked the vendors to specify the type of controller to which the converter interfaces, if applicable.

Number of X.25 channels supported. Here the vendor has specified the number of channels a PAD supports for connection to the public or private data network.

Number of virtual circuits supported. A virtual circuit is the logical connection between the input port and the destination port. When a terminal connects to a PAD, the PAD automatically or manually establishes a circuit to the destination by selecting an unused logical channel number and transmitting a Call Request packet that uses that logical channel number. This request packet identifies the input port and the destination port and carries other information used to set up the logical connection. In this entry, vendors have specified the maximum number of virtual circuits supported by the PAD.

Maximum window size, in frames. Window size describes the maximum number of unacknowledged frames (packets) that can be handled by the PAD at one time. Generally, PADs support up to seven frames in the window. When the PAD's transmitter reaches the maximum window size, it blocks the window. In effect, window framing is a form of flow control.

Link-level framing supported. At the link level, blocks of data are assembled according to certain framing protocols. These include character- or bit-oriented framing (BSC or HDLC, respectively). This is an EBCDIC or ASCII option on character-oriented (BSC) framing.

Terminal Side/Input Specifications

Number of types of ports (or devices) supported. In general, a conversion device supports asynchronous ports that accommodate a large variety of asynchronous ASCII printers, terminals, and personal computers. Many converters also support a dynamic printer port. Although more uncommon, conversion devices and PADs do support synchronous ports, or asynchronous and synchronous ports in combination. Devices represented in the charts support from one to many input devices; a number of units accommodate input-port expansion in specified increments.

Specific devices supported. It is important to know whether the unit supports a particular type of terminal. In today's market, most conversion devices designed for asynchronous ASCII to IBM SDLC or BSC conversion support virtually any asynchronous ASCII device. Some converters, however, are designed for operation only with a specific terminal. In this entry, the vendors have noted the manufacturer and model number of devices supported. An answer of "virtually any device" means that the vendor's list of supported terminals was too long to fit into the assigned space, but the converter did support all major asynchronous ASCII terminals and/or personal computers available in today's market.

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

KEY TO CONVERSION EQUIPMENT TABLE ENTRIES (Continued)

Autospeed/autoparity available. Many X.25 PADs automatically adjust to the transmission speed and parity of the inputting DTE. A "yes" answer indicates that the PAD supports this feature.

Channel configuration data downline loadable. X.25 PADs may support this feature, which allows terminal operators to configure channel parameters from the terminal and download those configurations to the PAD.

Ports configurable for permanent or switched circuits. Some PADs will allow users to configure an input port for permanent or switched virtual circuit connection through the network. In cases where the circuit is switched, the termination of a logical connection signals that the connection is free and can be used by another port. When the virtual circuit is permanent, the connection is dedicated to one port only. Many PADs support both permanent and switched virtual circuits on a selectable basis.

Transmission Specifications

Maximum transmission, in bps. This entry indicates the maximum speed of operation or data rate supported by the device stated in bits per second.

Maximum aggregate input rate, in bps. Conversion devices generally support many input ports, each operating at several different speeds, e.g., from 50 to 9600 bps. Aggregate input refers to the maximum data rate accepted from all channels simultaneously. For example, if there are four channels operating at a maximum 9600 bps rate per channel, the aggregate input rate could be four times 9600, or 38.4K bps.

Synchronization. This refers to the time relationship among the bits that make up the characters, which make up the messages. Conversion devices handle data in spurts (asynchronous) or continuous streams (synchronous).

Transmission mode. Most converters operate in either half- or full-duplex mode, or both. Half-duplex mode permits data transmission in either direction, but not simultaneously. Full-duplex operation implies that the data is simultaneously transmitted and received over a common communications facility. Simplex mode permits unidirectional data transmission, whereby data is either transmitted or received.

Protocols supported. Protocols are a set of rules that establish and control transmission. There are two basic types of protocols: byte-oriented (IBM's BSC or DEC's DDCMP) or bit-oriented (IBM SNA/SDLC or ISO HDLC). Converters usually translate one protocol to another and thus support different protocols on the terminal and host sides.

Codes supported. Codes consist of specific sets of characters that can be alphanumeric, graphic, and control characters. Control characters initiate, modify, or halt an action that effects data transmission. The most common data communications codes are ASCII, used in the asynchronous protocol, and EBCDIC, the usual code generated by synchronous devices.

Interface. Interface is the electrical connection between components. Most communications devices provide an electrical interface (RS-232-C) in accordance with the standards established by the Electronics Industries Association (EIA). Several other interface standards exist, notably CCITT Recommendation V.24 and V.28.

Clocking. Clocking refers to the repetitive, regularly timed signals used to control synchronous transmission. Clocking may be established internally by the device itself, externally by another device (for example, a modem), or be derived from the datastream.

Diagnostics. Many conversion devices perform tests that check the device and the line connections. Most converters conduct a self-test of internal circuitry upon power-up and provide front-panel LEDs to monitor system status.

Features (on X.25 PADs)

Channel priority assignment. Some PADs allow users to assign priority to incoming channels. Whenever the priority channel requests a connection to the network, that channel receives immediate access to the PAD and "bumps" channels with less priority.

Password protection. On many devices, users must enter a password to gain access to the PAD. This feature prevents unauthorized access to the network.

Supervisory port. Through this port, users can monitor and control the network and send messages throughout the system.

Echoplex. This feature refers to the printing of keyboarded characters on return of the signal from the other end of the line, using full-duplex transmission, to assure that the data was received correctly at the other end.

Autodialer support. Autodialers allow users to set dialing parameters, such as delay specifications for dial tone and call answering and switch-to-switch delay pauses, in memory. When this feature is present, dialing and disconnecting calls occur automatically.

Pricing and Availability

Purchase price. This is the basic price of the unit, excluding any options, except where noted in the Comments.

Rental. The monthly charge for leasing the unit from the vendor or a third party is shown in this entry.

Installation. When vendors charge a fee for installing the unit, we have included the one-time charge. Note that most conversion devices are designed for customer installation.

Maintenance. Many vendors charge an annual fee to service the unit and provide ongoing maintenance.

Serviced by. In this entry, we list the provider of service on the unit. Usually, the vendor offers service on an on-site or factory repair/return basis. In some cases, a third party provides service.

Availability. Here we list the current lead time on orders, given in days after receipt of order (ARO).

Date of first commercial delivery. Here we provide the date when the vendor first delivered the product to the marketplace.

Number installed to date. Some vendors list the approximate number of installed units as of March 1984. Note that in some cases, the vendor combines this figure for all models installed.

Comments. In this section, we have listed various special characteristics pertaining to a particular device. These might include additional capabilities, features, software, as well as information regarding related products offered by the vendor.

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

VENDORS

Listed below are the names, headquarters addresses, and telephone numbers of vendors who manufacture conversion devices. We have provided this list so that readers can contact the vendors for more information about the products they offer.

Advanced Computer Communications, 720 Santa Barbara Street, Santa Barbara, CA 93101. Telephone (805) 963-8801.

Agile Corporation, 4041 Pike Lane, Concord, CA 94520. Telephone (415) 825-9220.

Air Land Systems Corporation, 2710 Prosperity Avenue, Fairfax, VA 22031. Telephone (703) 573-1100.

Alphamatrix Inc., 1021 Millcreek Drive, Feasterville, PA 19047. Telephone (215) 355-3297.

Amdahl CSD, 2500 Walnut Avenue, Marina Del Rey, CA 90291. Telephone (213) 822-3202.

Avanti Communications Corporation, Aquidneck Industrial Park, Newport, RI 02840. Telephone (401) 849-4660.

Avatar Technologies Inc., 99 South Street, Hopkinton, MA 01748. Telephone (617) 435-6872.

BBN Communications Corporation, 70 Fawcett Street, Cambridge, MA 02238. Telephone (617) 497-2800.

Black Box Corporation, P.O. Box 12800, Pittsburgh, PA 15241. Telephone (412) 746-5500.

Cableshare Inc., P.O. Box 5880, 20 Enterprise Drive, London, Ontario, Canada N6A 4L6. Telephone (519) 686-2900.

Com/Tech Systems, Inc., 505 Eighth Avenue, New York, NY 10018. Telephone (212) 594-5377.

Commercial Data Processing, Inc., 2241 Grand Avenue, St. Louis, MO 63104. Telephone (314) 776-1130.

Commtex, 2411 Crofton Lane, Crofton, MD 21114. Telephone (301) 721-3666.

Computer Communications, Inc., 2610 Columbia Street, Torrance, CA 90503. Telephone (213) 320-9101.

Computer Peripheral Systems, Inc., Box 98282, Stone Mountain, GA 30059. Telephone (404) 292-9565.

Control Concepts, P.O. Box 2367, Mannassas, VA 22110. Telephone (703) 361-5545.

CTiData Corp., 5275 North Boulevard, Raleigh, NC 27604. Telephone (919) 876-8731.

Datagram Corporation, 11 Main Street, East Greenwich, RI 02818. Telephone (800) 235-5030.

Dataprobe, 110 West Palisades Boulevard, Palisades Park, NJ 07650. Telephone (201) 947-9500.

Datastream Communications, Inc., 2520 Mission College Boulevard, Santa Clara, CA 95050. Telephone (408) 986-8022.

Davox Communications Corp., 4 Federal Street, Billerica, MA 01821. Telephone (617) 667-4455.

Digital Communications Associates, 303 Technology Park, Norcross, GA 30092. Telephone (404) 448-1400.

Diversified Data Resources, Inc., 25 Mitchell Boulevard, Suite 7, San Rafael, CA 94903. Telephone (415) 499-8870.

DCC/Duracom Corporation, 7300 North Crescent Boulevard, Pennsauken, NJ 08110. Telephone (609) 662-7272.

Dynatech Packet Technology, Inc. (Dynapac), 6464 General Green Way, Alexandria, VA 22312. Telephone (703) 642-9391.

EDA Instruments, Inc. 4 Thorncliff Park Drive, Toronto, Ontario, Canada M4H 1H1. Telephone (416) 425-7800.

Gandalf Data, Inc., 350 East Dundee, Suite 201, Wheeling, IL 60090. Telephone (312) 541-6060.

General Datacomm Industries, , Middlebury, CT 06762-1299. Telephone (203) 574-1118.

GTE Telenet Communications Corp., 12490 Sunrise Valley Drive, Reston, VA 22096. Telephone (703) 689-6000.

Hewlett-Packard Grenoble, 5 Avenue Raymond Chanas 38320 Eybens, R.C.S. Grenoble, France B7098 05030. Telephone (76) 25 81 41.

Icot Corporation, 830 Maude Avenue, Mountain View, CA 94043. Telephone (415) 964-4635.

Incaa Computers, P.O. Box 211, 7300 AE Apeldoorn, Holland 51262. Telephone 055-5.

Infotron Systems, 9 North Olney Avenue, Cherry Hill, NJ 08003. Telephone (609) 424-9400.

Innovative Electronics, Inc., 4714 NW 165th Street, Miami Lakes, FL 33014. Telephone (305) 624-1644.

Instrumentation Services Incorporated (ISI), 957 Winnetka Avenue North, Minneapolis, MN 55427. Telephone (612) 544-8916.

International Business Machines Corporation, Old Orchard Road, Armonk, NY 10504. Contact your local IBM representative.

Kaufman Data Communications, Inc., 145 East Dana Street, Mountain View, CA 94041. Telephone (415) 962-8811.

KMW Systems, 8307 Highway 71 West, Austin, TX 78735. Telephone (512) 288-1453.

Local Data, Inc., 2771 Toledo Street, Torrance, CA 90503. Telephone (213) 320-7126.

Micom Systems, Inc., 4100 Los Angeles Avenue, P.O Box 8100, Simi Valley, CA 93062-8100. Telephone (805) 583-8600.

Modemsplus, Inc., 217 East Trinity Place, Decatur, GA 30030. Telephone (404) 378-5276.

Netlink Technology, 2920 Highwoods Boulevard, Suite 110, Raleigh, NC 27625. Telephone (919) 878-8612.

Perle Systems Ltd., 360 Tapscott Road, Scarborough, Ontario, Canada M1B 3C4. Telephone (416) 299-4999.

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

All About Conversion Systems and Emulation Devices

▷ **Protocol Computers, Inc.,** 6150 Canoga Avenue, Woodland Hills, CA 91367-3773. Telephone (818) 716-5500.

ProtoCom Devices, 190 Willow Avenue, Bronx, NY 10454. Telephone (212) 993-0077.

Remark Datacom, Division of Telebyte Technology Inc., 270 E. Pulaski Road, Greenlawn, NY 11740. Telephone (516) 423-3232.

Renex Corporaton, 6901 Old Keene Mill Road, Suite 500, Springfield, VA 22150. Telephone (703) 451-2200.

Shaffstall Corporation, 7901 East 88th Street, Indianapolis, IN 46250. Telephone (317) 842-2077.

Teleprocessing Products, 4565 East Industrial Street, Building 7K, Simi Valley, CA 93063. Telephone (805) 522-8149.

Thomas Engineering, 2440 Stanwell Drive, Concord, CA 94520. Telephone (415) 680-8640.

Tri-Data Corporation, 505 East Middlefield Road, Mountain View, CA 94039-7505. Telephone (415) 969-3700.

Tymnet, 2710 Orchard Parkway, San Jose, CA 95134. Telephone (408) 946-4900. □

© 1985 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

In this report, we discuss protocol conversion and emulation in the data communications environment. Using the OSI seven-layer model for data communications, we explain the various types of conversions that can take place in a network. We report on various types of conversion products, including code and interface converters, protocol converters, terminal controllers and emulators, PAD devices, and communications interface cards for microcomputers. Also included are a discussion of the IBM 3270 environment, a review of current trends in the conversion market, and recommendations for selecting conversion products. We have also presented 1983 Terminal Users Survey results that pertain to protocol conversion.

At the end of the report, we list the names and addresses of vendors that offer conversion products listed in the comparison charts that follow this report.

The problem of translation is an integral part of any communications system. Translators of written and spoken languages have always faced the complexities of converting one language to another. In music, the same tune played on one instrument sounds quite different when played on another. And one artist's interpretation of a scene is never exactly the same as that of another artist viewing the same scene.

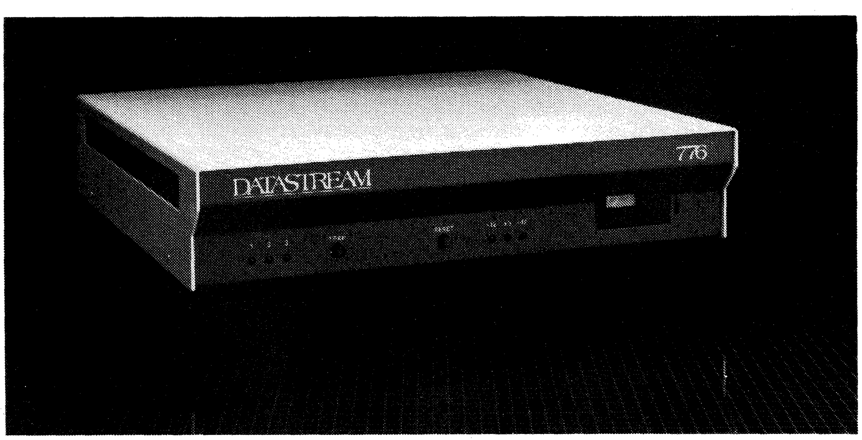
The technological advancements of the twentieth century have created the means to produce machines with the ability to communicate with one another. The science of such communication, which we call data communications, has become one of the most important developments of the Information Age. But like other communications systems, data communications is not immune to the problems of translation.

Data communications network designers can achieve flexibility and economic rewards by using products from more than one vendor. However, hardware and software from different vendors are rarely compatible with one another. They rarely speak the exact same language. If a user wants to design a network of diverse hardware and software products, he or she must deal with the problem of incompatibility. In doing so, the user must first understand exactly how various devices are incompatible with one another and then determine a way to deal with the differences.

In data communications, the solution to the problem of incompatibility lies in special hardware and software products that perform some type of conversion that translates the communications system of one device into that of another. Today there are growing numbers and varieties of these products to handle many types of incompatibilities in the data communications network. These products range from microprocessor-based circuit boards to front-end processors with the ability to handle conversion functions through software applications programs. Available conversion devices may handle only one, or more than one, type of conversion. For example, some devices handle only code or interface conversions, while others handle protocol conversion, device emulation, as well as code and interface translations.

This report concentrates on hardware conversion devices, particularly "black box" protocol converters and terminal controllers. We are aware that software packages for conversion and emulation are an extremely important part of the market. However, this reference service is primarily concerned with hardware. Readers interested in software conversion products should consult the *Datapro Directory of Software* and the *Datapro Directory of Microcomputer Software.*

We will cover the large and expanding micro-to-mainframe segment of the market in a separate report behind a new Tab (C22), Microcomputer Communications. Look for this section in Volume 2 of the August 1984 supplement.



Datastream's 776 Remote Cluster Controller operates in point-to-point, multipoint, and switched BSC networks as an IBM 3271/3276-compatible cluster controller. It supports IBM 3278 emulation on ASCII displays.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

ISO SEVEN-LAYER MODEL FOR DATA COMMUNICATIONS

(7) Application—provides communications services
(6) Presentation—defines syntax of data
(5) Session— controls data exchange
(4) Transport—handles data flow, error control
(3) Network—handles data routing
(2) Data Link—ensures data transfer via protocols
(1) Physical—provides mechanical/electrical interface

Figure 1. Layers One through Three define the interface between the host computer and the network. Layers Four through Seven provide compatibility to data format and exchange.

In this report, we focus attention on the ways in which devices must be compatible in order to communicate. Using the Open Systems Interconnection (OSI) seven-layer model for data communications as a guide, we explain the various kinds of conversions that can take place between devices. We then discuss the various products that handle particular conversions and the ways in which conversions occur. The report also contains a discussion about conversion in the IBM 3270 environment, since solving problems of incompatibility between ASCII devices and IBM hosts is of particularly high interest to many readers. Also included are discussions about current trends in the conversion marketplace, some tips for selecting conversion products, and the results of a section of our 1983 Terminal User's Survey concerning protocol conversion. At the end of the report is a list of vendors that provide various kinds of conversion products; their addresses and phone numbers are included.

INCOMPATIBILITY IN DATA COMMUNICATIONS

We have said that data communications devices can be incompatible with one another in several ways. The International Standards Organization (ISO) Open Systems Interconnection reference model—a seven-layer hierarchy that defines the electrical characteristics, communications standards, and software applications for computer systems—provides a framework for understanding the ways in which devices differ. Each layer of the model defines a particular aspect of the entire data communications process. Refer to Figure 1 for a representation of the seven-layer hierarchy.

- Layer 1 is the *Physical Layer,* which provides mechanical and electrical specifications and procedures to establish, maintain, and end physical connections. This layer defines interface, code, speed, and synchronization functions. Interface, code, and asynchronous-to-synchronous converters fall into this category.

- Layer 2 is the *Data Link Layer,* which insures that the data passes without error from one computer to another. This process involves protocols, a set of rules that specify the format for data transmission. Protocol converters are the devices that handle conversions in this layer.

- Layer 3 is the *Network Layer,* which lets two systems exchange data. This layer defines packet addressing and data routing to final destination. Units that handle conversion in this layer include gateway devices, such as packet assemblers/disassemblers that provide access to X.25 networks or between local area networks. Front-end processors (FEPs) that include protocol conversion in their functions also fall into this classification.

- Layer 4 is the *Transport Layer,* which handles end-to-end error and flow control to ensure that the communications exchange is orderly and reliable. PAD devices, a type of gateway product, are the major products in this layer. Note that we classify PADs in both the Network and Transport layers.

- Layer 5 is the *Session Layer,* which provides the structure for a data exchange by managing connections between application processes, establishing and terminating connections, and sending end-to-end messages and controller dialogues. There are currently few conversion products in this category; ProtoComm's new P2500 PAD device does handle conversion on the Session Layer, but is one of the few products that does so.

- Layer 6 is the *Presentation Layer,* which both defines the way in which data is put together and provides a systematic arrangement for the communications exchange to occur. This layer defines functions to translate coded data and convert it into display formats for terminal or microcomputer screens, printers, and other peripherals. In this layer, data is expanded or compressed and structured for file transfer or command translation. Devices called emulators that allow one type of terminal to appear as another type of terminal fall into the Presentation Layer category. Products in this category include ASCII-to-3270 emulators, interfaces that let personal computers act as 3270-type devices or access public networks, and word processor interfaces that handle conversions between dissimilar word processors.

- Layer 7 is the *Applications Layer,* which supports user and application tasks and provides the communications services that are available to specific computer applications. In essence, this layer provides the meaning to the message. Conversion devices that we discuss in this report do not provide conversions on this layer.

For devices to communicate with one another, they must be compatible on the interface, code, and protocol levels and must be alike according to link characteristics, device type, and device characteristics. Therefore, to connect incompatible equipment, converters must often provide translations on more than one of the levels in the network model. Conversion at one layer generally implies that compatibility in the layers below it in the model must also be accomplished. For example, a protocol converter work-

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

ing on Level 2 functions also assumes responsibility for compatibility in the interface, code, and synchronization functions.

Below we discuss the various products that handle conversion functions. These include interface, code, and asynchronous-to-synchronous converters; protocol converters; gateway devices, including PADs; protocol conversion in front-end processors; and terminal emulation/controllers, and remote cluster controllers that let one device appear as another.

Interface, Code, and Asynchronous-to-Synchronous Converters

An interface is the physical connection between two devices; interface conversion is the lowest level of established compatibility. Data and control lines from a device terminate in a connector that has pins that handle assigned signal functions. For example, the industry standard RS-232-C interface connector has 25 pins—one pin per function. The interface also prescribes voltage levels for electrical signals passing over the data and control lines.

Interface converters handle incompatibility between two interfaces. The devices link incompatible plugs, accept the connectors of two different interfaces, and/or translate the signals and voltage levels of one interface to that of another. Interfaces conversions commonly occur between RS-232-C and MIL STD 188 or between RS-232-C and V.35. Several vendors, including Avanti Communications Corporation, Gandalf, and Versitron, offer products that handle many different types of conversions at the interface level.

Code converters handle the transformation of one communications code to another. A communications code is a bit pattern for each text, graphic, or control character. The most common data communications codes are ASCII, EBCDIC, and Baudot. An end-user device that operates using one of these codes cannot accept data in another code. In addition, all error-checking codes (e.g., parity) must be compatible. The conversion from one code to another may be simple, involving only the addition or deletion of control bits or the alteration of parity. A more complex code conversion might require transforming the data character's bit pattern.

Basic code conversion hardware consists of two universal asynchronous/synchronous receiver/transmitters (USARTs), a translation table contained in ROM, and some control circuitry. Characters received by the USART in one code are mapped in the ROM table into a corresponding character in the destination device's code. Converted data goes to the other USART, which transmits it to the destination device.

One of the biggest problems of code conversion today is that of integrating word processors into data processing networks. Word processors typically have large character sets and control characters that are not used by data communications equipment. In some cases, the data com-

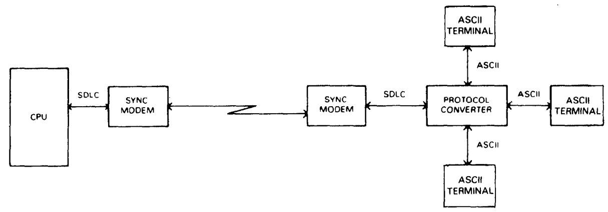


Figure 2. A typical configuration of a protocol converter at the terminal end of the line.

munications device uses a word processor character for a different function. To integrate word processors into a data communications network, users must first convert the code of the word processor to a code that data communications equipment understands.

Placing word processors in data communications networks is difficult for other reasons. In many cases, the word processor manufacturer has developed a complete communications protocol for the equipment. Changing that protocol requires a higher level of conversion.

Asynchronous-to-synchronous converters are an older type of equipment, used mostly in applications that require conversion of asynchronous terminals for use on synchronous lines. In most newer conversion units, asynchronous-to-synchronous conversion is included along with other translation functions.

Protocol Converters

Protocol converters, one of the largest categories of conversion devices, handle changes that must occur on the Data Link Layer to ensure device compatibility. Protocol converters, or protocol conversion processors, as they are sometimes called, typically connect some type of incompatible peripheral device to a host. Protocol converters are microprocessor-based machines that usually communicate with the peripheral in a simple protocol and with the host in a more complex protocol that incorporates error checking and retransmission capabilities. The converter communicates in the language of the peripheral and transforms and reformats data received from that peripheral before relaying it to the host, or the reverse, thus acting as an intermediary between the host and the peripheral. The peripheral to which the protocol converter attaches can be a terminal, a plotter or display, a microcomputer, a minicomputer, or another host.

A protocol is a fixed set of rules that specifies the format of a data exchange. The rules govern the recognition of a

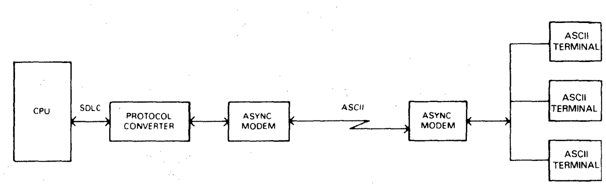


Figure 3. A typical configuration of a protocol converter at the host end of the line.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

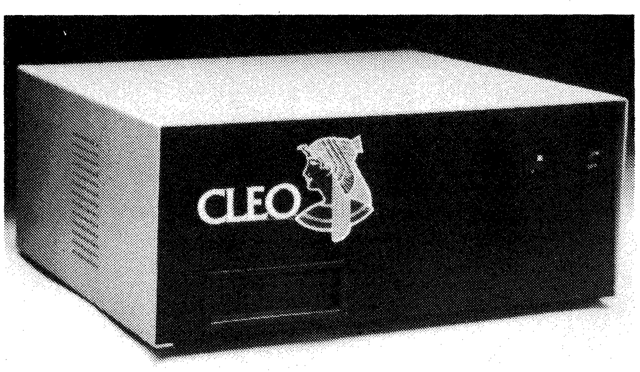
connection with a remote point, the identification of the transmitting and receiving location, the transmission sequence, the handling of interruptions, methods of error checking and control, methods of blocking data, and security procedures.

Communications protocols cover a wide spectrum: they range from single character-by-character communications with no error checking to complex rules for moving large amounts of data among many devices.

The physical manifestation of the protocol is a series of characters in bit combinations that are appended to each block or frame of transmitted data. Through hardware or software, the sending device automatically formats the data and adds the required bits before transmitting each block. The receiving device automatically checks each of the appended bits before signalling an acknowledgement that data has been received. If any established condition is not met, the protocol initiates error control procedures.

Data communications protocols are either bit oriented or byte oriented. Byte-oriented protocols require that data be transmitted in eight-bit blocks; an acknowledgement is required after each transmitted block before the next can be sent. Bit-oriented protocols allow data to be transmitted in blocks of any length up to a specified maximum; an acknowledgement may take place after one or several blocks have been sent, depending on the protocol. Some of the most common protocols are as follows:

- ASCII—an asynchronous protocol with very little error checking. Transmission is in the form of a start bit, a number of data bits (usually five to eight), and one or more stop bits. Data in ASCII protocol can enter the communications line at any time; the end of the link is synchronized through the specifications of a common line speed and detection of the start bits and the beginning of the character transmission. ASCII requires an acknowledgement after each block is sent. ASCII protocol is often referred to as Teletype (TTY) protocol, since it is traditionally associated with teletypewriter equipment and services.



Phone 1's Cleo protocol converter allows ASCII terminals to emulate IBM 3270 functions. Supported terminals include DEC, IBM, Televideo, Zenith, ADDS, Wicat, and Soroc.

- IBM's SDLC—a bit-oriented protocol that uses a synchronized series of frames. Each frame has a synchronization flag, followed by an address field, a control field that tells the purpose of the transmission, the data itself, then a frame-check field, and finally a trailing flag. The flag character is used to achieve the synchronization. SDLC permits up to 127 frames to be outstanding before an acknowledgement is received.

- IBM BSC—a character-oriented protocol. Binary synchronous data and control characters consist of eight-bit bytes. A transmission in BSC consists of a number of synchronizing (SYN) characters that ensure synchronizing at both ends of the communications link. These are followed by a start-of-text (STX) character, an eight-bit block of text, an end-of-text (ETX) character, and a block error-checking character (BCC). BSC lacks the capability to handle full-duplex data, and does not comply with IBM's System Network Architecture (SNA) concept. Each block must be acknowledged before the next can be sent.

Other communications protocols include HDLC (High Level Data Link Control), a bit-oriented protocol; Univac U200, CDC 200UT and Burroughs Multipoint Poll Select, which are similar to IBM BSC but can run on both synchronous and asynchronous links; and DEC's DDCMP (Digital Data Communications Message Protocol), a byte-oriented protocol that can handle up to 255 unacknowledged transmissions.

A protocol converter actually changes one protocol to another by stripping the data down and rewrapping it according to the rules of a new set of specifications. Although hardware specifications differ from vendor to vendor, protocol converters usually contain a microprocessor, a realtime clock, two serial ports, associated data-rate generators, and the necessary firmware and RAM buffer.

During the conversion sequence, the protocol converter accepts blocks of data in one protocol, adds or deletes the necessary control characters, reformats the block, and calculates the required check characters so that the receiving device receives characters formatted according to its requirements. For example, in an ASCII-to-SDLC conversion, the converter will accept a string of characters, eliminate the start and stop bits, assemble the characters into a block, and add appropriate headers and trailers to create complete frames. In a BSC-to-SDLC conversion, the converter must change the first four SYN bits of the Bisync algorithm to the first flag bit of the SDLC algorithm.

All protocol converters have some level of intermediate storage area to hold characters for conversion. Because of this buffering, a converter will always extend response time in the communications exchange. The device generally accepts low-speed input in the buffer, works with the data, and then transmits it out in short, high-speed bursts.

The data transmission method in the protocol converter differs from device to device. There are, however, some basic converter techniques. One of these techniques, called virtual protocol conversion, is used by a protocol converter that supports data transmissions up to 9600 bps. In virtual conversions, a central processor in the converter transforms each incoming data stream to its own protocol (the virtual protocol) and then reconverts the data stream to the protocol desired by the receiving device (the desired protocol).

An alternative technique uses a separate microprocessor to perform the conversion for each line interface the device handles. The interface has approximately 12K of PROM in which a conversion program resides. Additional RAM (usually about 2K) holds the data from each line. A common memory module serves as a shared RAM buffer area, where input/output queuing takes place. Converted data goes to the shared area where it is transferred to the host in queue.

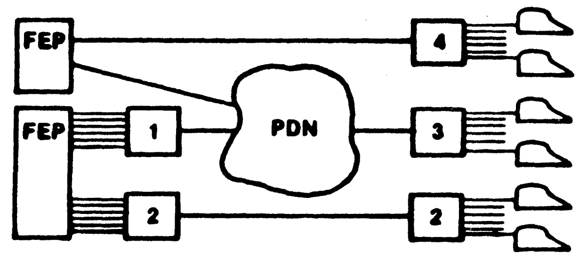
Besides pure protocol conversion, protocol converters often resolve related incompatibilities. For example, the converter might also translate ASCII code to EBCDIC or make several point-to-point links appear to the host as one multipoint link.

A special type of protocol converter is the Satellite Delay Compensation Unit (SDCU), which cuts propagation delay during satellite transmissions. Propagation delay is the amount of time between signal transmission over a circuit and acknowledgement of the transmission from the receiving end. Since the propagation delay during a satellite transmission is about a quarter of a second, this send-and-wait procedure can be quite time-consuming when every block requires acknowledgement before the next can be sent—as required by certain protocols, such as IBM BSC. The SDCU, which connects between the terminal and the modem, converts BSC into a specially conditioned form of SDLC that does not require an acknowledgement after each block. The end result is nearly 100 percent efficiency when transmitting in batch or message mode.

Gateway Devices, PADs, and FEPs

These products handle conversions on OSI Layers Three and Four (the Network and Transport layers, respectively) as well as performing lower layer functions. Very few conversion units handle translations on the Session Layer, although one vendor, ProtoComm, now offers a PAD that does perform conversions at this level.

Gateway devices are products that provide access between incompatible networks, for example, between SNA and DECnet, or between SNA and Ethernet, or between a data communications device and an X.25 public data network. Gateway products provide compatibility between network architectures, with their inherent protocols, codes, and interfaces. Gateway converters may link specific devices with one another like protocol converters do or they may link two complete, but mutually exclusive systems, such as a minicomputer and an IBM mainframe, each with its own complement of peripherals.



1 = Multi-PAD as host computer concentrator to network.
2 = Pair of PADs in statistical TDM configuration.
3 = Multi-PAD as terminal concentrator to network.
4 = Multi-PAD as terminal concentrator to host or front-end processor.

Figure 4. Typical Configurations for Dynatech's Multi-PAD.25.

By far the largest subset of gateway products are packet assembler/disassemblers (PADs). These devices permit host computers and peripheral equipment that use a communications protocol other than X.25 to be interconnected via a public data network. On the terminal side, most PADs support the connection of several devices, which can be terminals, CPU ports, printers, and so forth. On the network side, a high-speed port usually provides a link to the X.25 network. PADs usually perform concentrating and multiplexing functions as well as protocol conversion.

Most PAD products actually adapt a protocol rather than change it completely. The adaptation allows data in one protocol to pass through a network that uses another protocol. The transmitting PAD receives messages from the host or peripheral in the protocol of the sending device, converts and packetizes the information according to X.25 standards, and sends the packet through the network. At the receiving end of the X.25 link, another PAD performs error checking, disassembles the packets and converts messages back to the native protocol. Some PADs can also perform a true protocol conversion between the sending device and the destination device, when necessary.

In normal operations, the use of the PAD and the X.25 network are transparent to both the sending and the receiving devices. However, for test purposes, the PAD can be made to poll and to present status information to the host. Some PADs also have a supervisory port so that users can configure the PAD's operating parameters and even diagnose network problems through the PAD.

In Figure 3 we see a typical set of configurations for Dynatech's Multi-PAD.25. As the diagram shows, users can configure PADs to work as concentrators for the host computer, as statistical time division multiplexers, as terminal concentrators for the public data network, and as terminal concentrators to a host or FEP.

One of the trends in gateway conversion is interconnection between incompatible systems and peripherals through a PBX. For example, Intecom's IBX provides conversion between ASCII and 3270 protocols, and Rolm's CBX provides a gateway to IBM networks. Interfacing to X.25 networks and compatibility with specified local area net-

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

works (e.g., Ethernet) are also sometimes supported. Other PBX vendors are now including gateway conversion functions into their products.

Conversion can also occur in front-end processors (FEPs). These units achieve protocol conversion through software and hardware combinations. The FEP handles conversions for ASCII, BSC, SDLC, and X.25. How FEPs handle conversions varies widely, and most processors provide a variety of conversion functions.

Conversion can also occur in host-independent network processors. These devices usually rely on a microprocessor-based architecture to perform multiple functions. For example, Tri-data's Netway processor handles protocol conversion between dissimilar devices, serves as a concentrator for attached workstations, and manages communications among dissimilar hosts. The product also works as an X.25 packet processor to allow ASCII terminals to communicate with a host through X.25 networks. Netway processors allow many different hosts and workstations to communicate with one another in a network. The protocol translation capabilities of the device let users configure networks that include products from various vendors, including IBM, Burroughs, and DEC.

These communications processors cannot be specifically classified as converters because they handle several other high-level functions in a data communications network. These products do not exist primarily to provide conversion functions. For more information on these devices, consult Tab C13 of Volume 1 of *Datapro Reports on Data Communications.*

Some vendors include protocol conversion functions on their minicomputers. Data General, for example, provides an architecture for its Eclipse units to handle extensive protocol conversions. Other vendors provide conversion software packages for their minicomputers.

At present, very few vendors offer products that handle conversion on the Session Layer. One new company, Protocom Devices, does provide a P2500 PAD that supports Session Layer conversions to provide network security, simultaneous dual sessions, operation in Data Streaming/ Turbo Mode, and error handling. The P2500 protects an organization from unauthorized network access via random password generation and permits only authorized terminal-connected PADs to access preassigned host-connected PADs. P2500 also permits some connected terminals to engage more than one host at a time. Turbo Mode operation on the P2500 decreases queuing delays that occur during transmission of large messages. The P2500 uses an inter-PAD block-check sequence, local end-to-end acknowledgements, and data retransmission to provide efficient error-handling functions.

Emulation Devices

Devices that handle conversions on the Presentation Layer provide the capability for one device to appear as another device. While protocol converters handle incompatibility problems between sets of rules particular devices use to communicate information, an emulator must handle incompatibilities in all specification differences between sending and receiving units—including differences in protocol, code, interface, device characteristics, and link characteristics. To the emulator, protocol conversion is secondary: while the protocol converter actually strips down data and rewraps it according to a new set of rules, the emulator reads the text in a whole message and emulates that text to the specifications of a different device.

A great many protocol converters on the market today provide both protocol conversion and emulation. Often vendors call protocol/emulation products protocol converters, although this nomenclature is somewhat inaccurate. All emulation devices provide protocol conversion, but all protocol converters do not provide emulation. Most often, however, devices that handle protocol and emulation translations are called value-added terminal controllers, remote cluster controllers, or terminal emulators.

To use information in a transmission, a receiving device—whether a host or a terminal—must interpret data in the context of the device that it supports. Device specifications impose many constraints on the data communications protocol that the device handles. This means that although a host and a terminal might operate in the same protocol, they may not be compatible with one another.

The unit that connects device-incompatible equipment must reformat data to offset restrictions imposed by an emulated device. Restrictions can include differences in record size and blocking characteristics, or they may relate to functional differences between equipment types. Most terminal emulators are not general-purpose units: they convert only between specific types of devices.

The way a terminal emulator handles conversions depends upon the specific characteristics of the emulated and emulating devices. Thus, describing a general emulation technique is difficult. But an example of how a terminal emulator takes an asynchronous data stream and converts it to the protocol and format used by an IBM 3271 terminal controller illustrates a basic conversion sequence.

An IBM 3271 serves up to 32 IBM 3277-type terminals on a multipoint line. Data moving in this type of configuration is blocked out in 1920-character screen images (blocks of data). If a user wants to replace IBM 3277 terminals with asynchronous ASCII devices, the ASCII units must appear as IBM 3277s to the IBM host. A terminal controller/ emulator, or terminal controller as it is often called, can handle this problem by taking an asynchronous data stream into its buffer and keeping it there until a 1920-character screen image is filled or until the emulator receives an end-of-record, end-of-block control character. The terminal controller converts the protocol of the ASCII terminal to the protocol of the host (i.e., BSC), rearranges the data format to appear as if it comes from an IBM 3271, and then transfers the screen image to the host, which recognizes the data as that of an IBM 3277—not an asynchronous ASCII terminal. The terminal controller per-

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

forms all functions of the device it replaces, including data concentration, poll/select, flow control, buffering, error detection and correction, and interfacing of multiple attached terminals. For example, Icot's Virtual Terminal controllers emulate an IBM 3271 or 3274 controller and provide ASCII terminal-to-IBM 3277/3278/3279 terminal emulation and IBM 3284 printer emulation.

Sometimes the emulating device connects to an IBM cluster controller rather than replacing it, in effect, performing the conversion between the terminal and the IBM controller instead of between the controller and the host. The purpose of these emulators is to allow the user to integrate incompatible equipment into an existing terminal cluster. Local Data's Interlynx, for example, attaches to the IBM 3274 or 3276 controller to provide protocol and emulation translations that allow ASCII terminals to replace IBM 3278 or 3279 terminals.

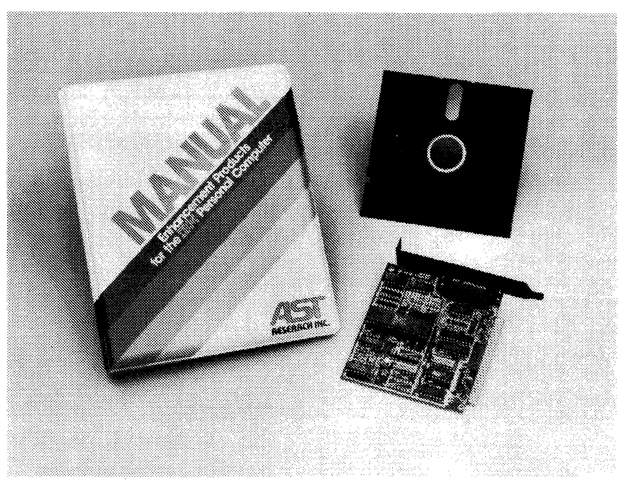
During an emulation/conversion/transfer sequence, the emulator must interpret control sequences from an attached terminal to simulate the operations of the emulated terminal. The equivalents for a specified control sequence between one terminal model and another model varies widely. For example, no asychronous ASCII keyboard provides all of the special 3270 function keys, and those that are provided are generally encoded differently by different devices. Functions like erasing a screen, setting cursor address, and so forth are also encoded differently. As commands arrive, the emulator must translate the sequence and operates upon it according to the equivalent function of the emulated device. The emulator unit then updates its internal buffers and the display screen of the attached terminal according to the control sequence it receives and translates.

One of the biggest problems users face when using terminal emulation products concerns the special keystrokes an operator must learn to produce capabilities not normally supported on a particular terminal. Terminal operators accustomed to the keystrokes of a particular terminal must learn a new set of keystrokes to effect the functions of the emulated terminal. This operation can be compared to typing in Arabic on a typewriter with an English keyboard and an Arabic font. (Type a "g" and another symbol appears on the paper.) Because this kind of operation can cause confusion, vendors usually provide key maps that show keystroke equivalents between the emulated terminal and the various emulating devices. Some vendors also provide stick-on decals for emulating keyboards.

Many users are purchasing these terminal controllers to allow non-IBM devices in remote locations to access IBM mainframes. Remote cluster controllers eliminate the need for dedicating one terminal (e.g., a 3270) to one application, and another terminal at the same site to a local minicomputer. Many remote controllers have one synchronous line for 3270 access and two or more minicomputer interfaces. Terminals attached to the controller can switch between a remote host mainframe and the remote and local minicomputers in this type of configuration.

Users can configure most terminal controllers for dial-up access, allowing ASCII terminals in a remote location to dial into the local controller, which then makes the connection with a CPU loca ed at the same or a third site. The controller eliminates the need for an IBM controller and additional synchronous lines to access the mainframe. A prominent cluster controller vendor, Datastream Communications, Inc., offers several models, including the 774 and the 776. The Model 776 operates in a point-to-point, multipoint or switched BSC network and acts as, and replaces, an IBM 3271/3276 cluster controller.

Units that handle conversions to make microcomputers and personal computers compatible with IBM mainframes represent a large and growing area in the conversion/emulation marketplace. Organizations are using more and more microcomputers for decentralized applications, but in many instances microcomputer users must have access to a centralized database, which generally resides on an IBM mainframe. Users can establish a micro-to-mainframe link through an emulation package that typically includes a diskette containing the emulation logic and a communications circuit board that is installed inside the microcomputer. An example of this type of product is DCA's Irma, one of the most popular micro-to-mainframe interfaces. The Irma is an IBM PC board with a coaxial interface that connects the PC to an IBM 3270 terminal controller that accesses a mainframe. With Irma installed and running on the PC, users can download data from the mainframe to the microcomputer, where it is viewed on the microcomputer screen. Like other forms of emulation, micro-to-mainframe links usually specify the microcomputers supported and the host ports and/or peripherals to which they can be connected. The Irma, for example, must attach to an IBM Personal Computer or compatible microcomputer and will attach only to an IBM or compatible 3274/3276 terminal controller. Other emulators provide IBM 2780/3780 batch terminal emulation for specified micro- and minicomputers.



AST Research Inc. supplies a number of communications packages that allow IBM PCs to emulate 3270 or 2770 terminals. The communications packages consist of diskette-resident software and a circuit board that fits into the PC.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

Other types of emulation products offer conversions that are the reverse of the popular ASCII-to-3270 conversion. For example, Protocol Computers' new 74D unit lets an IBM 3270-type device talk to an ASCII host. The 74D interfaces with an IBM communications controller, an IBM 3270 cluster controller, and up to six ASCII hosts, which can be DEC hosts, public networks, and personal computers. Supported by the 74D, the 3270 can switch between SDLC and DEC hosts.

Conversion and emulation in a data communications network can occur in many different devices and at many points in a network. Converters can be separate hardware units placed between a terminal and a modem; shared hardware units that handle other functions (e.g., front-end processors); devices that replace cluster controllers; interface cards in a personal computer; or applications programs, specialized emulation software packages or software/hardware resident on minicomputers, mainframes, and PBX systems. Many network services, such as Tymnet and Telenet, also offer conversion as part of their value-added products. Even phone companies offer conversion as part of their network services. One recent controversy between GTE Telenet and Southern Bell concerns the BOC's inclusion of protocol conversion in its LADT service. Value-added carriers would like to restrict such offerings by the BOCs, but the most recent Computer Inquiry rulings do not specifically prohibit the telephone companies from offering the service.

Protocol conversion and emulation products address problems of incompatibility among many types of data communications devices. But as you might have surmised from our discussion above, the majority of conversion units are designed specifically to incorporate incompatible devices in an IBM environment. In the next section of this report, we will discuss that environment in relation to conversion and emulation products.

The IBM 327X Environment

Tremendous growth in the minicomputer, microcomputer, and personal computer markets has led to a rapid increase in the number of installed ASCII asynchronous terminals that access these computers. However, ASCII devices cannot access information that resides on IBM mainframes. IBM's series of products that provide interactive communications in an IBM network is the IBM 3270 Information Display System. This series includes controllers, terminals, and printers that are dedicated to a single host and usually to a single application.

Components in the current 3270 system include: the 3277, 3278, 3279, 3178, and 3290 display terminals, 3284, 3286, 3287, 3288, and 3289 printers, and the 3274 and 3276 cluster controllers. Each component comes in various models. For example, the 3278 is a monochromatic display available in five models that essentially differ only in their screen capacities. The 3279 is a color display version of the 3278. The 3274 controller comes in various models that handle up to 12, 16, or 32 attached terminals, local or remote host connection, BSC or SDLC protocol. The 3276 is a smaller controller designed for clusters of up to eight terminals.

Because of the 3270's huge installed base, many models not longer actively marketed by IBM continue to play a significant role in the IBM-compatible markets, particularly the 3271 and 3272 controllers. The 3271 is a remote cluster controller that handles up to 32 terminals and comes in BSC and SDLC versions. The 3272 is a local channel-attached version of the 3271.

There are some shortcomings to using products in the 3270 family. First, they are expensive. Second, many of the IBM components are physically larger and take up more space than the ASCII terminals and the emulators that can be used in their place.

To acknowledge the need for asynchronous communication, in 1979 IBM introduced a Model 3101 terminal, which can attach directly to a 3705 communications controller and participate in ASCII applications resident in the host. The company also introduced a four-port protocol converter, the 7426, to allow the 3101 to appear as a 3278 to the 8100 and 4300 Series computers. A new Yale ASCII Communications System software package lets almost any ASCII device access 3270 applications and appear as a 3270 terminal. Most importantly, however, IBM introduced 3270 emulation support for most of its mini- and microprocessor-based products including the IBM PC, the System/36, and the Displaywriter. In doing so, IBM, in effect, changed the 3270 from a single-host, dedicated terminal system to a protocol that can accommodate many different devices.

Although the majority of protocol converters and terminal controllers on the market today handle some type of conversion between ASCII devices and IBM units, other products handle conversion between IBM BSC protocols to IBM SDLC Protocols. This conversion is particularly useful to users of older IBM BSC equipment who wish to migrate to an SNA/SDLC environment without replacing all of their old equipment. BSC-to-SDLC conversions generally occur between BSC 2780/3780 RJE or 3270 BSC protocols and SDLC protocols.

As IBM PCs become increasingly prevalent in organizations, products to provide micro-to-mainframe compatibility will become more and more important. The entire protocol/emulation market is exploding today because units that make ASCII terminals and personal computers compatible with SNA/SDLC networks are in tremendous demand. ASCII devices provide flexible and inexpensive solutions to network problems, but IBM's mainframes are still the de facto standard for centralized computer facilities that must handle large databases and many applications. It seems unlikely that this situation will change soon, and vendors that offer conversion products to handle ASCII-to-IBM conversions should enjoy a huge market for their products. In fact, current trends in the protocol conversion and terminal emulation marketplace reflect an increasingly

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

heated competition among data communications vendors for a share of the potentially large and lucrative conversion market.

CURRENT TRENDS

Many different products handle some type of conversion to provide compatibility between communications devices. Presently, a number of large data communications equipment vendors are incorporating protocol converters and terminal controllers into their general line of products. Micom Systems, for example, recently acquired Industrial Computer Controls, Inc., one of the oldest specialized protocol converter manufacturers in the marketplace. At the same time, Micom introduced the Micro7400 protocol conversion, a replacement of ICCI's CA20 converter. The Micro7400 handles ASCII-to-3270 conversion and provides network monitoring and control functions as well.

Formerly the venue of small companies like Protocol Computers, Inc. and Innovative Electronics, which specialize in standalone protocol converter products, protocol conversion has become incorporated into existing data communications products. We now find conversion as an integral capability in digital data switches, PBXs, personal computers, and word processors. And a market that in 1980 earned gross yearly revenues of about $5 million has in 1983 mushroomed into a $100 million a year business. Overnight, protocol converters have become the "hot" product in the ever-changing data communications environment.

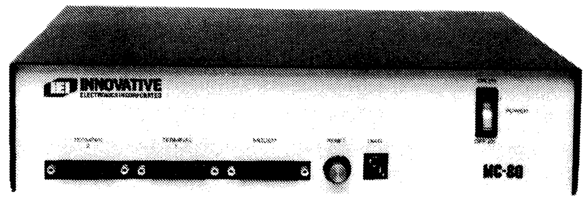
From a historical perspective, we can benchmark interest in protocol conversion at IBM's introduction of its 7426 converter in October 1982. With this announcement, IBM not only sanctioned conversion technology as a viable solution to network problems, but also focused industry attention on the technology.

Conversion products that facilitate LAN-to-LAN compatibility and access X.25 public networks are also expected to have a large market. We can expect to see a growing interest in PAD devices that effect X.25 access, and we can anticipate greater PBX conversion capabilities in the months ahead. Conversion offerings from value-added carriers, such as Tymnet and Telenet, and from the BOCs will also grow as data communications moves further into the home markets where personal computer users are becoming more interested in linking into public networks and databases.

Until the data communications industry adapts and uses worldwide protocol standards to link equipment, protocol converters and emulators will remain an important part of the general market. It is unlikely that such standardization will occur in the very near future.

CHOOSING CONVERSION DEVICES

Before choosing a conversion unit, users should consider some of the negative characteristics of the devices. First, protocol converters will cause delays in response time on the network because data must flow into the converter's buffer before transmission. If data backs up in the buffer, overruns occur; if the buffer is small, the converter can lose data.



Innovative Electronics' MC-80 Series protocol converters include many models that provide various conversions: IBM 2780/3780 to ASCII, Burroughs Poll/Select to ASCII, IBM 3270 to Burroughs Poll/Select, and others.

Terminal operators dealing with devices emulating other products may have problems learning the new key sequences and key functions necessary to the emulation process. Thus, organizations can expect some decreased productivity during the initial months of a conversion installation.

In addition, protocol converters usually do not offer the security provided by, for example, the IBM 3270-type devices. Organizations must deal with the problem of protecting sensitive data, particularly in dial-up applications.

When an organization decides to install conversion products in a data communications network, it should determine exactly what kinds of conversions are needed to solve particular incompatibilities. Once this is established, users should determine which kind of products can handle the conversion most effectively in a particular application. This can be an extremely confusing task because there are so many conversion products available. To narrow choices, it is wise to contact many vendors and ask for product specifications and documentation that explains how a product operates. When studying specifications and operating procedures, users must note exactly what types of terminals, controllers, or hosts are supported by the device because most converters and controllers support specific products rather than a general range of devices. For example, a protocol converter specifically designed for IBM 3277 emulation might not work with a 3278 application.

Also important is finding out what added features and functions the converter handles. Does it support more than one host? Does it replace an IBM controller, or is it used in conjunction with a controller? Does the device incorporate any multiplexing or concentrating? Can the network manager monitor the network via the converter? If additional features are available, are they standard or optional? What cost savings will it represent in your overall networking scheme?

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

After narrowing their choices to the products of several vendors, users should ask the company to provide an in-house demonstration of the product. A prospective buyer should also request a list of current users who will discuss their experiences with the product. These individuals can provide information about the advantages and disadvantages of the product, hardware reliability, and the type and quality of support provided by the vendor.

IBM mainframe users in particular should find out whether conversion equipment can be upgraded as IBM upgrades and changes its SNA architecture.

After further narrowing the selection to two or three vendors, users should request a free trial of the product. By using a converter in a particular application, prospective buyers can soon find out whether a product provides the desired compatibility in an efficient manner.

USER EXPERIENCE

In July 1983, Datapro conducted a Terminal Users Survey, which was based on results received from questionnaires mailed to a cross-section of *Data Communications* magazine subscribers. Several questions in the survey pertained to the use of protocol conversion devices in a network. Below we show the results of these questions and draw some conclusions from the information.

We asked the users to indicate the primary protocols supported by their terminals:

	Number of Responses	Percent of Total Responses
Asynchronous	274	70
IBM BSC	187	47
IBM SDLC	130	33
Other bit-oriented synchronous protocol (e.g., ANSI ADCCP, ISO HDLC, Sperry UDLC, or Burroughs BDLC)	44	11
X.25 packet-level	34	9
Other byte-oriented synchronous protocol (e.g., DEC DDCMP)	35	9
Other	22	6

Although we are not in a position to draw any formal conclusions, since this year's user sample consists of different respondents from last year's, some interesting observations can be made when the two years' responses are compared. (The size of the respondent group is approximately the same: 447 respondents in 1982 versus 404 respondents for 1983.) Asynchronous protocol users have increased from 62 percent in 1982 to 70 percent in 1983, indicating an increased use of ASCII terminals, while users of IBM BSC and other byte-oriented protocols have decreased from 65 to 56 percent. One possible explanation may be the increasing use of protocol conversion in IBM environments. The use of IBM SDLC has remained nearly steady—34 percent for 1982 compared to 33 percent for 1983, and the high number of users who indicated using multiple protocols in their networks suggests that many of these users are still in various stages of migration to SNA and in no hurry to complete it. While the number of X.25 packet-level users remains small, the percentage has more than doubled since last year (four percent in 1982 versus nine percent in 1983), indicating a steadily growing use of packet switching networks.

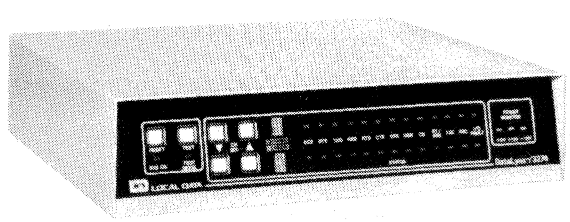
We also asked these users to identify any types of protocol conversion that was being performed for their applications:

	Number of Responses	Percent of Total Responses
ASCII-to-BSC	107	27
ASCII-to-SDLC	75	19
BSC-to-SDLC	33	8
Other	27	6

. . . and by what means this protocol conversion is performed:

	Number of Responses	Percent of Total Responses
Software loaded onto an existing system such as a general-purpose computer, front-end processor, terminal controller, or PBX system	107	27
Dedicated protocol converter	96	24
Value-added network service (e.g., Telenet)	22	6
Other	2	1

The strength of the protocol conversion market that has emerged in the past two years is certainly confirmed by these users' responses. Although these two questions were new on the 1983 questionnaire, and thus we have no comparative data from last year, the heavy use of ASCII-to-BSC conversion seems to bear out the conclusions concerning the increased use of asynchronous terminals we noted earlier in this report. Most likely, a great many of the ASCII-to-BSC conversions involve ASCII terminals emulating 3270-type devices.



Local Data's DataLynx/3274 provides asynchronous ASCII-to-BSC or SDLC conversion. The unit emulates an IBM 3274 or 3276 controller and allows ASCII CRTs to emulate IBM 3278 terminal displays.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

COMPARISON CHARTS

The charts that accompany this report present listings of the key characteristics of approximately 82 protocol converters or terminal emulators, 15 X.25 PADS, and 22 code, speed, interface, and async/sync converters. This information was supplied by the vendors during February and March of 1984. Datapro wishes to thank the participating vendors for their cooperation.

Datapro sent repeated requests for information to 60 firms known or believed to manufacture some type of hardware conversion device. *The absence of any company from the charts means that the company either failed to respond to our request by the survey deadline, was unknown to us, did not make a hardware conversion product, or chose not to be listed.*

Many companies presently manufacture conversion devices for microcomputer-to-mainframe communication. These products consist of a circuit board that is inserted into the microcomputer and accompanying software. These devices, predominanty designed to effect IBM PC-to-mainframe connections, comprise a large and growing segment of the conversion marketplace and will be covered in a separate report in the August supplement of *Datapro Reports on Data Communications.* Look for this report and the accompanying comparison charts behind the Microcomputer Communications (C22) tab in Volume Two.

KEY TO CONVERSION EQUIPMENT TABLE ENTRIES

For the reader's convenience, we have organized the comparison charts into three broad categories. **Conversion Systems/Terminal Controllers** may be a variety of devices, including protocol converters, terminal emulators, remote cluster controllers, and terminal controllers. Basically, devices in this section provide conversion from one protocol to another and/or allow one device, e.g., an asynchronous ASCII terminal, to act as another type of device, e.g., an IBM 3270 terminal, in a network. **X.25 Packet Assemblers/Disassemblers (PADS)** convert equipment using a non-X.25 protocol to the X.25 protocol for transmission over public and private networks. PADs also perform other functions, including concentrating or multiplexing. **Code, Speed, Interface, and Async/Sync Converters** include a number of miscellaneous devices that handle conversions from one code, interface, speed, or synchronization to another. These are generally less sophisticated devices than those represented in the other two categories.

The following text briefly describes the chart entries in the order in which they appear in the charts. Not all of the characteristics listed appear in all of the charts because some entries do not apply to all categories. For example, "link-level framing support" is relevant only to X.25 PADs.

General Characteristics

Manufacturer and model. We list here the manufacturer and exact model number or name of each device.

Device type. In the Conversion Systems/Terminal Controllers section, we have asked the manufacturer to specify whether the device is a protocol converter, terminal emulator, code or interface converter, and so forth.

Conversion performed. All converters perform some type of translation from one code, speed, or protocol to another. The most common conversion is asynchronous ASCII to IBM SNA/SDLC or BSC, but a number of other translations are available on the units represented in the tables.

Specific device emulated. In many cases, conversion devices provide the means to convert the text format of one type of device into the characteristics and format of another device. This translation, called emulation, is indicated, if available. Note that most protocol converters also provide device emulation.

Specific functionality provided. Most converters allow one device to be used as another type of device in the network. For example, a number of units allow asynchronous ASCII CRTs to be used as IBM 3277 models 1 and 2. It is very important for prospective users to know exactly what type of device can be emulated by a given converter because many converters are designed for a specific application.

Virtual screen sizes supported. For a device to provide emulation, it must support the screen size, in characters, of the emulated device. For example, a device emulating an IBM 3270 terminal must support a 1920 character screen.

Command port support. Some converters support a port through which users can select operating parameters and monitor, diagnose, and control the network. A "yes" answer indicates that the device does have a command port.

Network certification. X.25 PADs allow devices to interface with public and private networks. These networks must certify use of the device on their facilities. Prominent network providers include Tymnet, Telenet, Datapac, and Uninet.

CCITT recommendation supported. CCITT has specified certain protocols for devices operating on an X.25 network. Most X.25 PADs support all CCITT recommendations, including X.25 Levels 1, 2, 3; X.3; X.28; and X.29. X.25 defines the operating protocol of the packet netwok. X.3 defines a set of 18 parameters that govern operation of the PAD and each asynchronous terminal port. X.28 defines the interface between the asynchronous terminal and the PAD, and X.29 defines the format of the packets that carry control information from the PAD to the remote destination.

Buffer memory capacity. PAD devices contain a buffer memory that holds packets before transmission. Software is generally held in ROM.

Additional RAM available. Many PADs can accommodate the additional RAM necessary to expand the capacity of the device.

Device configuration. Most X.25 PADs can be configured to act as other types of devices in the network. Typical configurations include host concentration, time division multiplexing, or front-end processing.

(continued on next page)

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

KEY TO CONVERSION EQUIPMENT TABLE ENTRIES (Continued)

Host Side/Network Channel Specifications

Specific hosts supported. Conversion devices generally support IBM or compatible hosts or asynchronous hosts such as DEC VAX or PDP 11. In this entry, we have listed the type of mainframe with which the converter operates.

Number and type of host lines supported. Most converters support one line to a host; that line can be one IBM SNA/SDLC or BSC line or one line to an asynchronous host. Some devices do support more than one host connection. This connection may be a dual IBM BSC or SDLC link or one IBM link and one link to an asynchronous host.

Number host sessions supported concurrently. If a converter supports more than one host line, the device may support both connected hosts concurrently, or separately through a switch-selection.

Connections supported. Conversion devices support direct connections, multipoint, and/or point-to-point connections. Most converters support more than one type of connection, and most support all three.

Connection to host via controller. While some conversion devices emulate a controller, others must connect to a controller in the network. We have asked the vendors to specify the type of controller to which the converter interfaces, if applicable.

Number X.25 channels supported. Here the vendor has specified the number of channels a PAD supports for connection to the public or private data network.

Number virtual circuits supported. A virtual circuit is the logical connection between the input port and the destination port. When a terminal connects to a PAD, the PAD automatically or manually establishes a circuit to the destination by selecting an unused logical channel number and transmitting a Call Request packet that uses that logical channel number. This request packet identifies the input port and the destination port and carries other information used to set up the logical connection. In this entry, vendors have specified the maximum number of virtual circuits supported by the PAD.

Maximum window size, in frames. Window size describes the maximum number of unacknowledged frames (packets) that may be handled by the PAD at one time. Generally, PADs support up to seven frames in the window. When the PAD's transmitter reaches the maximum window size, it blocks the window. In effect, window framing is a form of flow control.

Link-level framing supported. At the link level, blocks of data are assembled according to certain framing protocols. These include character- or bit-oriented framing (BSC or HDLC, respectively). This is an EBCDIC or ASCII option on character-oriented (BSC) framing.

Terminal Side/Input Specifications

Number of type of ports (or devices) supported. In general, a conversion device supports asynchronous ports that accommodate a large variety of asynchronous ASCII printers, terminals, and personal computers. Many converters also support a dynamic printer port. Although more uncommon, conversion devices and PADs do support synchronous ports, or asynchronous and synchronous ports in combination. Devices represented in the charts support from one to many input devices; a number of units accommodate input-port expansion in specified increments.

Specific devices supported. It is important to know whether the unit supports a particular type of terminal. In today's market, most conversion devices designed for asynchronous ASCII to IBM SDLC or BSC conversion support virtually any asynchronous ASCII device. Some converters, however, are designed for operation only with a specific terminal. In this entry, the vendors have noted the manufacturer and model number of devices supported. An answer of "virtually any device" means that the vendor's list of supported terminals was too long to fit into the assigned space, but the converter did support all major asynchronous ASCII terminals and/or personal computers available in today's market.

Autospeed/autoparity available. Many X.25 PADs automatically adjust to the transmission speed and parity of the inputting DTE. A "yes" answer indicates that the PAD supports this feature.

Channel configuration data downline loadable. X.25 PADs may support this feature, which allows terminal operators to configure channel parameters from the terminal and down load those configurations to the PAD.

Ports configurable for permanent or switched circuits. Some PADs will allow users to configure an input port for permanent or switched virtual circuit connection through the network. In cases where the circuit is switched, the termination of a logical connection signals that the connection is free and can be used by another port. When the virtual circuit is permanent, the connection is dedicated to one port only. Many PADs support both permanent and switched virtual circuits on a selectable basis.

Transmission Specifications

Maximum transmission, in bps. This entry indicates the maximum speed of operation or data rate supported by the device stated in bits per second.

Maximum aggregate input rate, in bps. Conversion devices generally support many input ports, each operating at several different speeds, e.g., from 50 to 9600 bps. Aggregate input refers to the maximum data rate accepted from all channels simultaneously. For example, if there are four channels operating at a maximum 9600 bps rate per channel, the aggregate input rate could be four times 9600, or 38.4K bps.

Synchronization. This refers to the time relationship among the bits that make up the characters, which make up the messages. Conversion devices handle data in spurts (asynchonrous) or continuous streams (synchronous).

Transmission mode. Most converters operate in either half- or full-duplex mode, or both. Half-duplex mode permits data transmission in either direction, but not simultaneously. Full-duplex operation imples that the data is simultaneously transmitted and received over a common communications facility. Simplex mode permits unidirectional data transmission, whereby data is either transmitted or received.

Protocols supported. Protocols are a set of rules that establish and control transmission. There are two basic types of proto-

(continued on next page)

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

KEY TO CONVERSION EQUIPMENT TABLE ENTRIES (Continued)

cols: byte-oriented (IBM's BSC or DEC's DDCMP) or bit-oriented (IBM SNA/SDLC or ISO HDLC). Converters usually translate one protocol to another and thus support different protocols on the terminal and host sides.

Codes supported. Codes consist of specific sets of characters that can be alphanumeric, graphic, and control characters. Control characters initiate, modify, or halt an action that effects data transmission. The most common data communications codes are ASCII, used in the asynchronous protocol, and EBCDIC, the usual code generated by synchronous devices.

Interface. Interface is the electrical connection between components. Most communications devices provide an electrical interface (RS-232-C) in accordance with the standards established by the Electronics Industries Association (EIA). Several other interface standards exist, notably CCITT Recommendation V.24 and V.28.

Clocking. Clocking refers to the repetitive, regularly timed signals used to control synchronous transmission. Clocking may be established internally by the device itself, externally by another device (for example, a modem), or be derived from the data stream.

Diagnostics. Many conversion devices perform tests that check the device and the line connections. Most converters conduct a self-test of internal circuitry upon power-up and provide front-panel LEDs to monitor system status.

Features (on X.25 PADs)

Channel priority assignment. Some PADs allow users to assign priority to incoming channels. Whenever the priority channel requests a connection to the network, that channel receives immediate access to the PAD and "bumps" channels with less priority.

Password protection. On many devices, users must enter a password to gain access to the PAD. This feature prevents unauthorized access to the network.

Supervisory port. Through this port, users can monitor and control the network and send messages throughout the system.

Echoplex. This feature refers to the printing of keyboarded characters on return of the signal from the other end of the line, using full-duplex transmission, to assure that the data was received correctly at the other end.

Autodialer support. Autodialers allow users to set dialing parameters, such as delay specifications for dial tone and call answering and switch-to-switch delay pauses, in memory. When this feature is present, dialing and disconnecting calls occur automatically.

Pricing and Availability

Purchase price. This is the basic price of the unit, excluding any options, except where noted in the Comments.

Rental. The monthly charge for leasing the unit from the vendor or a third party is shown in this entry.

Installation. When vendors charge a fee for installing the unit, we have included the one-tme charge. Note that most conversion devices are designed for customer installation.

Maintenance. Many vendors charge an annual fee to service the unit and provide on-going maintenance.

Serviced by. In this entry, we list the provider of service on the unit. Usually, the vendor offers service on an on-site or factory repair/return basis. In some cases, a third party provides service.

Availability. Here we list the current lead time on orders, given in days after receipt of order (ARO).

Date of first commercial delivery. Here we provide the date when the vendor first delivered the product to the marketplace.

Number installed to date. Some vendors list the approximate number of installed units as of March 1984. Note that in some cases, the vendor combines this figure for all models installed.

Comments. In this section, we have listed various special characteristics pertaining to a particular device. These might include additional capabilities, features, software, as well as information regarding related products offered by the vendor.

▷ VENDORS

Listed below are the names, headquarters addresses, and telephone numbers of vendors who manufacture conversion devices. We have provided this list so that readers can contact the vendors for more information about the products they offer.

Advanced Computer Communications, 720 Santa Barbara Street, Santa Barbara, CA 93101. Telephone (805) 963-8801.

Agile Corporation, 4041 Pike Road, Concord, CA 94520. Telephone (415) 825-9220.

Air Land Systems Corporation, 2710 Prosperity Avenue, Fairfax, VA 22031. Telephone (703) 573-1100.

Alphamatrix Inc., 1021 Millcreek Drive, Feasterville, PA 19047. Telephone (215) 355-3297.

Avanti Communications Corporation, Aquidneck Industrial Park, Newport, RI 02840. Telephone (401) 849-4660.

Avatar Technologies Inc., 99 South Street, Hopkinton, MA 01748. Telephone (617) 435-6872.

BBN Communications Corporation, 70 Fawcett Street, Cambridge, MA 02238. Telephone (617) 497-2800.

Black Box Catalog, P.O. Box 12800, Pittsburgh, PA 15241. Telephone (412) 746-2910.

Cableshare Inc., P.O. Box 5880, 20 Enterprise Drive, London, Ontario N6A 4L6. Telephone (519) 686-2900. ▷

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

▷ **Commercial Data Processing, Inc.,** 2241 Grand Avenue, St. Louis, MO 63104. Telephone (314) 776-1130.

Commtex, 2411 Crofton Lane, Crofton, MD 21114. Telephone (301) 721-3666.

Computer Communications, Inc., 2610 Columbia Street, Torrance, CA 90503. Telephone (800) 421-1178; in California (213) 320-9101.

Computer Peripheral Systems, Inc., 4765 Banner Elk Drive, Stone Mountain, GA 30083. Telephone (404) 292-9565.

Com/Tech Systems, Inc., 505 Eighth Avenue, New York, NY 10018. Telephone (212) 594-5377.

CTiData Corp., 5275 North Boulevard, Raleigh, NC 27604. Telephone (919) 876-8731.

Datagram Corporation, 11 Main Street, East Greenwich, RI 02818. Telephone (401) 885-4840.

Dataprobe, 110 West Palisades Boulevard, Palisades Park, NJ 07650. Telephone (201) 947-9550.

Datastream Communications, Inc., 2520 Mission College Boulevard, Santa Clara, CA 95050. Telephone (408) 986-8022.

Davox, 6 Continental Boulevard, Merrimack, NH 03054. Telephone (603) 424-4500.

Digital Communications Associates, 303 Technology Park, Norcross, GA 30092. Telephone (404) 448-1400.

Diversified Data Resources, Inc., 25 Mitchell Boulevard, Suite 7, San Rafael, CA 94903. Telephone (415) 499-8870.

Duracom Corporation, 7300 North Crescent Boulevard, Pennsauken, NJ 08110. Telephone (609) 662-7277.

Dynapac, 6464-G General Green Way, Alexandria, VA 22312. Telephone (703) 642-9391.

Gandalf Data, Inc., 1019 Noel Avenue, Wheeling, IL 60090. Telephone (312) 541-6060.

General Datacomm Industries, One Kennedy Avenue, Danbury, CT 06810. Telephone (203) 797-0711.

Hewlett-Packard Grenoble, 5 Avenue Raymond Chanas 38320 Eybens, R.C.S. Grenoble, France B709805030. Telephone (76) 25 81 41.

Icot Corporation, 830 Maude Avenue, Mountain View, CA 94043. Telephone (415) 964-4635.

Incaa Computers, P.O. Box 211, 7300 AE Apeldoorn, Holland. Telephone 055-551262.

Infotron Systems, 9 North Olney Avenue, Cherry Hill, NJ 08003. Telephone (609) 424-9400.

Innovative Electronics, Inc., 4714 NW 165th Street, Miami, FL 33014. Telephone (305) 624-1644.

Instrument Services Incorporated, 957 Winnetka Avenue North, Minneapolis, MN 55427. Telephone (612) 544-8916.

International Business Machines Corporation, Old Orchard Road, Armonk, NY 10504. Contact your local IBM representative.

Kaufman Data Communications, Inc., 145 East Dana Street, Mountain View, CA 94041. Telephone (415) 962-8811.

KMW Systems, 8307 Highway 71 West, Austin, TX 78735. Telephone (512) 288-1453.

Local Data, Inc., 2701 Toledo Street, Suite 706, Torrance, CA 90503. Telephone (213) 320-7126.

Micom Systems, Inc., 20151 Nordhoff Street, Chatsworth, CA 91311. Telephone (213) 998-8844.

Modemsplus, Inc., 217 East Trinity Place, P.O. Box 1727, Decatur, GA 30031. Telephone (404) 378-5276.

Netlink Technology, Inc., 1340 Saratoga Sunnyvale Road, San Jose, CA 95129. Telephone (408) 973-9411.

NuData Corporation, 32 Fairview Avenue, P.O. Box 125, Little Silver, NJ 07739. Telephone (201) 842-5757.

Peripheral Technology, Inc., 14784 N.E. 95th Street, Redmond, VA 98052. Telephone (800) 822-2208; in Virginia (206) 881-6691.

Perle Systems Ltd., 360 Tapscott Road, Scarborough, Ontario M1B 3C4. Telephone (416) 299-4999.

Protocol Computers, Inc., 6150 Canoga Avenue, Woodland Hills, CA 91367-3773. Telephone (800) 423-5904; in California (213) 716-5500.

ProtoCom Devices, 207 Atlantic Street, Stamford, CT 06901-3504. Telephone (203) 327-6893.

Racal-Telesystems, 401 North Michigan Avenue, Chicago, IL 60611. Telephone (312) 329-0700.

Remark Datacom, (Division of Telebyte Technology), 148 New York Avenue, Halesite, NY 11743. Telephone (516) 423-3237.

Renex Corporaton, 6901 Old Keene Mill Road, Suite 500, Springfield, VA 22150. Telephone (703) 451-2200.

Shaffstall Corporation, P.O. Box 50990, Indianapolis, IN 46250. Telephone (317) 842-2077.

SR Systems, 8150 Trans-Canada Highway, St. Laurent, Quebec H45 1M5. Telephone (514) 335-1210.

Teleprocessing Products, 4565 East Industrial Street, Building 7K, Simi Valley, CA 93063. Telephone (805) 522-8149.

Thomas Engineering, 1040 Oak Grove Road, Suite 106, Concord, CA 94518. Telephone (415) 680-8640.

Tri-Data Corporation, 505 East Middlefield Road, Mountain View, CA 94043. Telephone (415) 696-3700.

Tymnet, 2710 Orchard Parkway, San Jose, CA 95134. Telephone (408) 946-4900.

Versitron Inc., 6310 Chillum Place, N.W., Washington, DC 20011. Telephone (202) 882-8464. □

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

In this report, we discuss protocol conversion and emulation in the data communications environment. Using the OSI seven-layer model for data communications, we explain the various types of conversions that can take place in a network. We report on various types of conversion products, including code and interface converters, protocol converters, terminal controllers and emulators, PAD devices, and communications interface cards for microcomputers. Also included are a discussion of the IBM 3270 environment, a review of current trends in the conversion market, and recommendations for selecting conversion products. We have also presented 1983 Terminal Users Survey results that pertain to protocol conversion.

At the end of the report there is a comprehensive list of the names and headquarters addresses of vendors that offer conversion and emulation products.

The problem of translation is an integral part of any communications system. Translators of written and spoken languages have always faced the complexities of converting one language to another. In music, the same tune played on one instrument sounds quite different when played on another. And one artist's interpretation of a scene is never exactly the same as that of another artist viewing the same scene.

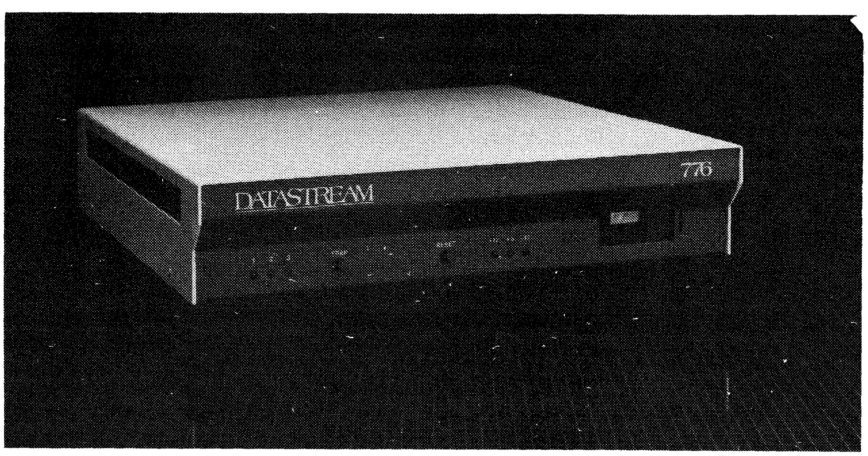
The technological advancements of the twentieth century have created the means to produce machines with the ability to communicate with one another. The science of such communication, which we call data communications, has become one of the most important developments of the Information Age. But like other communications systems, data communications is not immune to the problems of translation.

Data communications network designers can achieve flexibility and economic rewards by using products from more than one vendor. However, hardware and software from different vendors are rarely compatible with one another. They rarely speak the exact same language. If a user wants to design a network of diverse hardware and software products, he or she must deal with the problem of incompatibility. In doing so, the user must first understand exactly how various devices are incompatible with one another and then determine a way to deal with the differences.

In data communications, the solution to the problem of incompatibility lies in special hardware and software products that perform some type of conversion that translates the communications system of one device into that of another. Today there are growing numbers and varieties of these products to handle many types of incompatibilities in the data communications network. These products range from microprocessor-based circuit boards to front-end processors with the ability to handle conversion functions through software applications programs. Available conversion devices may handle only one, or more than one, type of conversion. For example, some devices handle only code or interface conversions, while others handle protocol conversion, device emulation, as well as code and interface translations.

This report concentrates on hardware conversion devices, particularly "black box" protocol converters and terminal controllers. We are aware that software packages for conversion and emulation are an extremely important part of the market. However, this reference service is primarily concerned with hardware. Readers interested in software conversion products should consult the *Datapro Directory of Software* and the *Datapro Directory of Microcomputer Software.*

In this report, we focus attention on the ways in which devices must be compatible in order to communicate. Using the Open Systems Interconnection (OSI) seven-layer model for data communications as a guide, we explain the various kinds of conversions that can take place between devices. We then discuss the various products that handle ▷



Datastream's 776 Remote Cluster Controller operates in point-to-point, multipoint, and switched BSC networks as an IBM 3271/3276-compatible cluster controller. It supports IBM 3278 emulation on ASCII displays.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

ISO SEVEN-LAYER MODEL FOR DATA COMMUNICATIONS

(7) Application—provides communications services
(6) Presentation—defines syntax of data
(5) Session— controls data exchange
(4) Transfer—handles data flow, error control
(3) Network—handles data routing
(2) Data Link—ensures data transfer via protocols
(1) Physical—provides mechanical/electrical interface

Figure 1. Layers One through Three define the interface between the host computer and the network. Layers Four through Seven provide compatibility to data format and exchange.

particular conversions and the ways in which conversions occur. The report also contains a discussion about conversion in the IBM 3270 environment, since solving problems of incompatibility between ASCII devices and IBM hosts is of particularly high interest to many readers. Also included are discussions about current trends in the conversion marketplace, some tips for selecting conversion products, and the results of a section of our 1983 Terminal User's Survey concerning protocol conversion. At the end of the report is a list of vendors that provide various kinds of conversion products; their addresses and phone numbers are included.

INCOMPATIBILITY IN DATA COMMUNICATIONS

We have said that data communications devices can be incompatible with one another in several ways. The International Standards Organization (ISO) Open Systems Interconnection reference model—a seven-layer hierarchy that defines the electrical characteristics, communications standards, and software applications for computer systems—provides a framework for understanding the ways in which devices differ. Each layer of the model defines a particular aspect of the entire data communications process. Refer to Figure 1 for a representation of the seven-layer hierarchy.

- Layer 1 is the *Physical Layer,* which provides mechanical and electrical specifications and procedures to establish, maintain, and end physical connections. This layer defines interface, code, speed, and synchronization functions. Interface, code, and asynchronous-to-synchronous converters fall into this category.

- Layer 2 is the *Data Link Layer,* which insures that the data passes without error from one computer to another. This process involves protocols, a set of rules that specify the format for data transmission. Protocol converters are the devices that handle conversions in this layer.

- Layer 3 is the *Network Layer,* which lets two systems exchange data. This layer defines packet addressing and data routing to final destination. Units that handle conversion in this layer include gateway devices, such as packet assemblers/disassemblers that provide access to X.25 networks or between local area networks. Front-end processors (FEPs) that include protocol conversion in their functions also fall into this classification.

- Layer 4 is the *Transfer Layer,* which handles end-to-end error and flow control to ensure that the communications exchange is orderly and reliable. PAD devices, a type of gateway product, are the major products in this layer. Note that we classify PADs in both the Network and Transfer layers.

- Layer 5 is the *Session Layer,* which provides the structure for a data exchange by managing connections between application processes, establishing and terminating connections, and sending end-to-end messages and controller dialogues. There are currently few conversion products in this category; ProtoComm's new P2500 PAD device does handle conversion on the Session Layer, but is one of the few products that does so.

- Layer 6 is the *Presentation Layer,* which both defines the way in which data is put together and provides a systematic arrangement for the communications exchange to occur. This layer defines functions to translate coded data and convert it into display formats for terminal or microcomputer screens, printers, and other peripherals. In this layer, data is expanded or compressed and structured for file transfer or command translation. Devices called emulators that allow one type of terminal to appear as another type of terminal fall into the Presentation Layer category. Products in this category include ASCII-to-3270 emulators, interfaces that let personal computers act as 3270-type devices or access public networks, and word processor interfaces that handle conversions between dissimilar word processors.

- Layer 7 is the *Applications Layer,* which supports user and application tasks and provides the communications services that are available to specific computer applications. In essence, this layer provides the meaning to the message. Conversion devices that we discuss in this report do not provide conversions on this layer.

For devices to communicate with one another, they must be compatible on the interface, code, and protocol levels and must be alike according to link characteristics, device type, and device characteristics. Therefore, to connect incompatible equipment, converters must often provide translations on more than one of the levels in the network model. Conversion at one layer generally implies that compatibility in the layers below it in the model must also be accomplished. For example, a protocol converter working on Level 2 functions also assumes responsibility for compatibility in the interface, code and synchronization functions.

Below we discuss the various products that handle conversion functions. These include interface, code, and asynchronous-to-synchronous converters; protocol converters; gateway devices, including PADs; protocol conversion in front-end processors, and terminal emulation/controllers

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

and remote cluster controllers that let one device appear as another.

Interface, Code, and Asynchronous-to-Synchronous Converters

An interface is the physical connection between two devices; interface conversion is the lowest level of established compatibility. Data and control lines from a device terminate in a connector that has pins that handle assigned signal functions. For example, the industry standard RS-232-C interface connector has 25 pins—one pin per function. The interface also prescribes voltage levels for electrical signals passing over the data and control lines.

Interface converters handle incompatibility between two interfaces. The devices link incompatible plugs, accept the connectors of two different interfaces, and/or translate the signals and voltage levels of one interface to that of another. Interface conversions commonly occur between RS–232–C and MIL STD 188 or between RS–232–C and V.35. Several vendors, including Avanti Communications Corporation, Gandalf, and Versitron, offer products that handle many different types of conversions at the interface level.

Code converters handle the transformation of one communications code to another. A communications code is a bit pattern for each text, graphic, or control character. The most common data communications codes are ASCII, EBCDIC, and Baudot. An end-user device that operates using one of these codes cannot accept data in another code. In addition, all error-checking codes (e.g., parity) must be compatible. The conversion from one code to another may be simple, involving only the addition or deletion of control bits or the alteration of parity. A more complex code conversion might require transforming the data character's bit pattern.

Basic code conversion hardware consists of two universal asynchronous/synchronous receiver/transmitters (USARTs), a translation table contained in ROM, and some control circuitry. Characters received by the USART in one code are mapped in the ROM table into a corresponding character in the destination device's code. Converted data goes to the other USART, which transmits it to the destination device.

One of the biggest problems of code conversion today is that of integrating word processors into data processing networks. Word processors typically have large character sets and control characters that are not used by data communications equipment. In some cases, the data communications device uses a word processor character for a different function. To integrate word processors into a data communications network, users must first convert the code of the word processor to a code that data communications equipment understands.

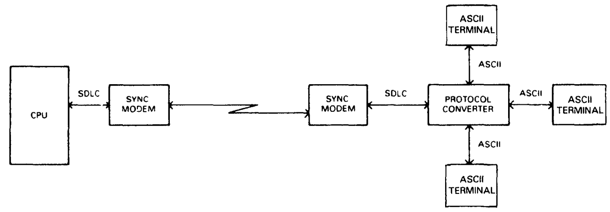


Figure 2. A typical configuration of a protocol converter at the terminal end of the line.

Placing word processors in data communications networks is difficult for other reasons. In many cases, the word processor manufacturer has developed a complete communications protocol for the equipment. Changing that protocol requires a higher level of conversion.

Asynchronous-to-synchronous converters are an older type of equipment, used mostly in applications that require conversion of asynchronous terminals for use on synchronous lines. In most newer conversion units, asynchronous-to-synchronous conversion is included along with other translation functions.

Protocol Converters

Protocol converters, one of the largest categories of conversion devices, handle changes that must occur on the Data Link Layer to ensure device compatibility. Protocol converters, or protocol conversion processors as they are sometimes called, typically connect some type of incompatible peripheral device to a host. Protocol converters are microprocessor-based machines that usually communicate with the peripheral in a simple protocol and with the host in a more complex protocol that incorporates error checking and retransmission capabilities. The converter communicates in the language of the peripheral and transforms and reformats data received from that peripheral before relaying it to the host, or the reverse, thus acting as an intermediary between the host and the peripheral. The peripheral to which the protocol converter attaches can be a terminal, a plotter or display, a microcomputer, a minicomputer, or another host.

A protocol is a fixed set of rules that specifies the format of a data exchange. The rules govern the recognition of a connection with a remote point, the identification of the transmitting and receiving location, the transmission sequence, the handling of interruptions, methods of error

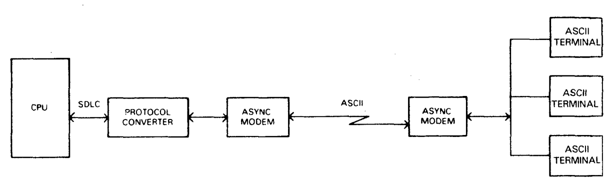


Figure 3. A typical configuration of a protocol converter at the host end of the line.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

checking and control, methods of blocking data, and security procedures.

Communications protocols cover a wide spectrum: they range from single character-by-character communications with no error checking to complex rules for moving large amounts of data among many devices.

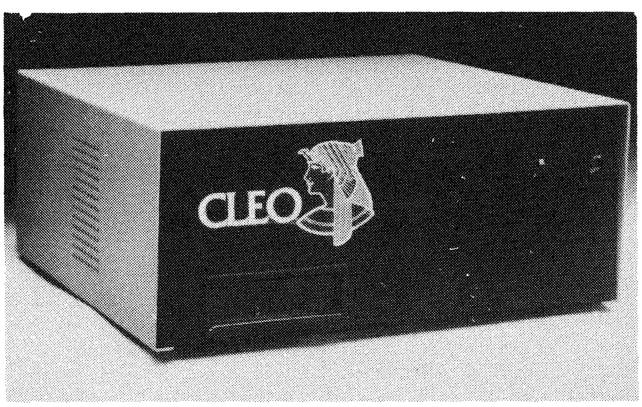
The physical manifestation of the protocol is a series of characters in bit combinations that are appended to each block or frame of transmitted data. Through hardware or software, the sending device automatically formats the data and adds the required bits before transmitting each block. The receiving device automatically checks each of the appended bits before signalling an acknowledgement that data has been received. If any established condition is not met, the protocol initiates error control procedures.

Data communications protocols are either bit oriented or byte-oriented. Byte-oriented protocols require that data be transmitted in eight-bit blocks; an acknowledgement is required after each transmitted block before the next can be sent. Bit-oriented protocols allow data to be transmitted in blocks of any length up to a specified maximum; an acknowledgement may take place after one or several blocks have been sent, depending on the protocol. Some of the most common protocols are as follows:

- ASCII—a bit-oriented asynchronous protocol with very little error checking. Transmission is in the form of a start bit, a number of data bits (usually 5 to 8), and one or more stop bits. Data in ASCII protocol can enter the communications line at any time; the end of the link is synchronized through the specifications of a common line speed and detection of the start bits and the beginning of the character transmission. ASCII requires an acknowledgement after each block is sent. ASCII protocol is often referred to as Teletype (TTY) protocol, since it is traditionally associated with teletypewriter equipment and services.

- IBM's SDLC—a bit-oriented protocol that uses a synchronized series of frames. Each frame has a synchronization flag, followed by an address field, a control field that tells the purpose of the transmission, the data itself, then a frame-check field, and finally a trailing flag. The flag character is used to achieve the synchronization. SDLC permits up to 127 frames to be outstanding before an acknowledgement is received.

- IBM BSC—a byte-oriented protocol. Binary synchronous data and control characters consist of eight-bit bytes. A transmission in BSC consists of a number of synchronizing (SYN) characters that ensure synchronizing at both ends of the communications link. These are followed by a start-of-text (STX) character, an 8-bit block of text, an end-of-text (ETX) character, and a block error-checking character (BCC). BSC lacks the capability to handle full-duplex data, and does not comply with IBM's System Network Architecture (SNA) concept. Each block must be acknowledged before the next can be sent.



Phone 1's Cleo protocol converter allows ASCII terminals to emulate IBM 3270 functions. Supported terminals include DEC, IBM, Televideo, Zenith, ADDS, Wicat, and Soroc.

Other communications protocols include HDLC (High Level Data Link Control), a bit-oriented protocol; Univac U200, CDC 200UT and Burroughs Multipoint Poll Select, which are similar to IBM BSC but can run on both synchronous and asynchronous links; and DEC's DDCMP (Digital Data Communications Message Protocol), a byte-oriented protocol that can handle up to 255 unacknowledged transmissions.

A protocol converter actually changes one protocol to another by stripping the data down and re-wrapping it according to the rules of a new set of specifications. Although hardware specifications differ from vendor to vendor, protocol converters usually contain a microprocessor, a real-time clock, two serial ports, associated data-rate generators, and the necessary firmware and RAM buffer.

During the conversion sequence, the protocol converter accepts blocks of data in one protocol, adds or deletes the necessary control characters, reformats the block, and calculates the required check characters so that the receiving device receives characters formatted according to its requirements. For example, in an ASCII-to-SDLC conversion, the converter will accept a string of characters, eliminate the start and stop bits, assemble the characters into a block, and add appropriate headers and trailers to create complete frames. In a BSC-to-SDLC conversion, the convert must change the first four SYN bits of the Bisync algorithm to the first flag bit of the SDLC algorithm.

All protocol converters have some level of intermediate storage area to hold characters for conversion. Because of this buffering, a converter will always extend response time in the communications exchange. The device generally accepts low-speed input in the buffer, works with the data, and then transmits it out in short, high-speed bursts.

The data transmission method in the protocol converter differs from device to device. There are, however, some basic conversion techniques. One of these techniques, called virtual protocol conversion, is used by a protocol converter that supports data transmissions up to 9600 bps. In virtual conversions, a central processor in the converter

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

transforms each incoming data stream to its own protocol (the virtual protocol) and then reconverts the data stream to the protocol desired by the receiving device (the desired protocol).

An alternative technique uses a separate microprocessor to perform the conversion for each line interface the device handles. The interface has approximately 12K of PROM in which a conversion program resides. Additional RAM (usually about 2K) holds the data from each line. A common memory module serves as a shared RAM buffer area, where input/output queuing takes place. Converted data goes to the shared area where it is transferred to the host in queue.

Besides pure protocol conversion, protocol converters often resolve related incompatibilities. For example, the converter might also translate ASCII code to EBCDIC or make several point-to-point links appear to the host as one multipoint link.

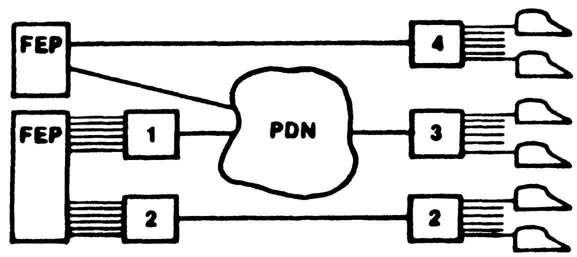
A special type of protocol conversion is the Satellite Delay Compensation Unit (SDCU), which cuts propagation delay during satellite transmissions. Propagation delay is the amount of time between signal transmission over a circuit and acknowledgement of the transmission from the receiving end. Since the propagation delay during a satellite transmission is about a quarter of a second, this send-and-wait procedure can be quite time-consuming when every block requires acknowledgement before the next can be sent—as required by certain protocols, such as IBM BSC. The SDCU, which connects between the terminal and the modem, converts BSC into a specially conditioned form of SDLC that does not require an acknowledgement after each block. The end result is nearly 100 percent efficiency when transmitting in batch or message mode.

Gateway Devices, PADs, and FEPs

These products handle conversions on OSI Layers Three and Four (the Network and Transport layers, respectively) as well as performing lower layer functions. Very few conversion units handle translations on the Session Layer, although one vendor, ProtoComm, now offers a PAD that does perform conversions at this level.

Gateway devices are products that provide access between incompatible networks, for example, between SNA and DECnet, or between SNA and Ethernet, or between a data communications device and an X.25 public data network. Gateway products provide compatibility between network architectures, with their inherent protocols, codes, and interfaces. Gateway converters may link specific devices with one another like protocol converters do or they may link two complete, but mutually exclusive systems, such as a minicomputer and an IBM mainframe, each with its own complement of peripherals.

By far the largest subset of gateway products are packet assembler/disassemblers (PADs). These devices permit host computers and peripheral equipment that use a communications protocol other than X.25 to be interconnected via a public data network. On the terminal side, most PADs support the connection of several devices, which can be terminals, CPU ports, printers, and so forth. On the network side, a high-speed port usually provides a link to the X.25 network. PADs usually perform concentrating and multiplexing functions as well as protocol conversion.



1 = Multi-PAD as host computer concentrator to network.
2 = Pair of PADs in statistical TDM configuration.
3 = Multi-PAD as terminal concentrator to network.
4 = Multi-PAD as terminal concentrator to host or front-end processor.

Figure 4. Typical Configurations for Dynatech's Multi-PAD.25.

Most PAD products actually adapt a protocol rather than change it completely. The adaptation allows data in one protocol to pass through a network that uses another protocol. The transmitting PAD receives messages from the host or peripheral in the protocol of the sending device, converts and packetizes the information according to X.25 standards, and sends the packet through the network. At the receiving end of the X.25 link, another PAD performs error checking, disassembles the packets and converts messages back to the native protocol. Some PADs can also perform a true protocol conversion between the sending device and the destination device, when necessary.

In normal operations, the use of the PAD and the X.25 network are transparent to both the sending and the receiving devices. However, for test purposes, the PAD can be made to poll and to present status information to the host. Some PADs also have a supervisory port so that users can configure the PAD's operating parameters and even diagnose network problems through the PAD.

In Figure 3 we see a typical set of configurations for Dynatech's Multi-PAD.25. As the diagram shows, users can configure PADs to work as concentrators for the host computer, as statistical time division multiplexers, as terminal concentrators for the public data network, and as terminal concentrators to a host or FEP.

One of the trends in gateway conversion is interconnection between incompatible systems and peripherals through a PBX. For example, Intecom's IBX provides conversion between ASCII and 3270 protocols, and Rolm's CBX provides a gateway to IBM networks. Interfacing to X.25 networks and compatibility with specified local area networks (e.g., Ethernet) are also sometimes supported. Other PBX vendors are now including gateway conversion functions into their products.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

Conversion can also occur in front-end processors (FEPs). These units achieve protocol conversion through software and hardware combinations. The FEP handles conversions for ASCII, BSC, SDLC, and X.25. How FEPs handle conversions varies widely, and most processors provide a variety of conversion functions.

Conversion can also occur in host-independent network processors. These devices usually rely on a microprocessor-based architecture to perform multiple functions. For example, Tri-data's Netway processor handles protocol conversion between dissimilar devices, serves as a concentrator for attached workstations, and manages communications among dissimilar hosts. The product also works as an X.25 packet processor to allow ASCII terminals to communicate with a host through X.25 networks. Netway processors allow many different hosts and workstations to communicate with one another in a network. The protocol translation capabilities of the device let users configure networks that include products from various vendors, including IBM, Burroughs, and DEC.

These communications processors cannot be specifically classified as converters because they handle several other high-level functions in a data communications network. These products do not exist primarily to provide conversion functions. For more information on these devices, consult Tab C11 of Volume 1 of *Datapro Reports on Data Communications.*

Some vendors include protocol conversion functions on their minicomputers. Data General, for example, provides an architecture for its Eclipse units to handle extensive protocol conversions. Other vendors provide conversion software packages for their minicomputers.

At present, very few vendors offer products that handle conversion on the Session Layer. One new company, Protocomm Devices, does provide a P2500 PAD that supports Session Layer conversions to provide network security, simultaneous dual sessions, operation in Data Streaming/ Turbo Mode, and error handling. The P2500 protects an organization from unauthorized network access via random password generation and permits only authorized terminal-connected PADs to access pre-assigned host-connected PADs. P2500 also permits some connected terminals to engage more than one host at a time. Turbo Mode operation on the P2500 decreases queuing delays that occur during transmission of large messages. The P2500 uses an inter-PAD block-check sequence, local end-to-end acknowledgements, and data retransmission to provide efficient error-handling functions.

Emulation Devices

Devices that handle conversions on the Presentation Layer provide the capability for one device to appear as another device. While protocol converters handle incompatibility problems between sets of rules particular devices use to communicate information, an emulator must handle incompatibilities in all specification differences between sending and receiving units—including differences in protocol, code, interface, device characteristics, and link characteristics. To the emulator, protocol conversion is secondary: while the protocol converter actually strips down data and re-wraps it according to a new set of rules, the emulator reads the text in a whole message and emulates that text to the specifications of a different device.

A great many protocol converters on the market today provide both protocol conversion and emulation. Often vendors call protocol/emulation products protocol converters, although this nomenclature is somewhat inaccurate. All emulation devices provide protocol conversion, but all protocol converters do not provide emulation. Most often, however, devices that handle protocol and emulation translations are called value-added terminal controllers, remote cluster controllers, or terminal emulators.

To use information in a transmission, a receiving device—whether a host or terminal—must interpret data in the context of a device that it supports. Device specifications impose many constraints on the data communications protocol that the device handles. This means that although hosts or terminals might operate in the same protocol, they may not be compatible with one another.

The unit that connects device-incompatible equipment must reformat data to offset restrictions imposed by an emulated device. Restrictions can include differences in record size and blocking characteristics, or they may relate to functional differences between equipment types. Most terminal emulators are not general-purpose units: they convert only between specific types of devices.

The way a terminal emulator handles conversions depends upon the specific characteristics of the emulated and emulating devices. Thus, describing a general emulation technique is difficult. But an example of how a terminal emulator takes an asynchronous data stream and converts it to the protocol and format used by an IBM 3271 terminal controller illustrates a basic conversion sequence.

An IBM 3271 serves up to 32 IBM 3277-type terminals on a multipoint line. Data moving in this type of configuration is blocked out in 1920-character screen images (blocks of data). If a user wants to replace IBM 3277 terminals with asynchronous ASCII devices, the ASCII units must appear as IBM 3277s to the IBM host. A terminal controller/ emulator, or terminal controller as it is often called, can handle this problem by taking an asynchronous data stream into its buffer and keeping it there until a 1920-character screen image is filled or until the emulator receives an end-of-record, end-of-block control character. The terminal controller converts the protocol of the ASCII terminal to the protocol of the host (i.e., BSC), rearranges the data format to appear as if it comes from an IBM 3271, and then transfers the screen image to the host, which recognizes the data as that of an IBM 3277—not an asynchronous ASCII terminal. The terminal controller performs all functions of the device it replaces, including data concentration, poll/select, flow control, buffering, error

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

detection and correction, and interfacing of multiple attached terminals. For example, Icot's Virtual Terminal controllers emulate an IBM 3271 or 3274 controller and provide ASCII terminal-to-IBM 3277/3278/3279 terminal emulation and IBM 3284 printer emulation.

Sometimes the emulating device connects to an IBM cluster controller rather than replacing it, in effect, performing the conversion between the terminal and the IBM controller instead of between the controller and the host. The purpose of these emulators is to allow the user to integrate incompatible equipment into an existing terminal cluster. Local Data's Interlynx, for example, attaches to the IBM 3274 or 3276 controller to provide protocol and emulation translations that allow ASCII terminals to replace IBM 3278 or 3279 terminals.

During an emulation/conversion/transfer sequence, the emulator must interpret control sequences from an attached terminal to simulate the operations of the emulated terminal. The equivalents for a specified control sequence between one terminal model and another model varies widely. For example, no asychronous ASCII keyboard provides all of the special 3270 function keys, and those that are provided are generally encoded differently by different devices. Functions like erasing a screen, setting cursor address, and so forth are also encoded differently. As commands arrive, the emulator must translate the sequence and operates upon it according to the equivalent function of the emulated device. The emulator unit then updates its internal buffers and the display screen of the attached terminal according to the control sequence it receives and translates.

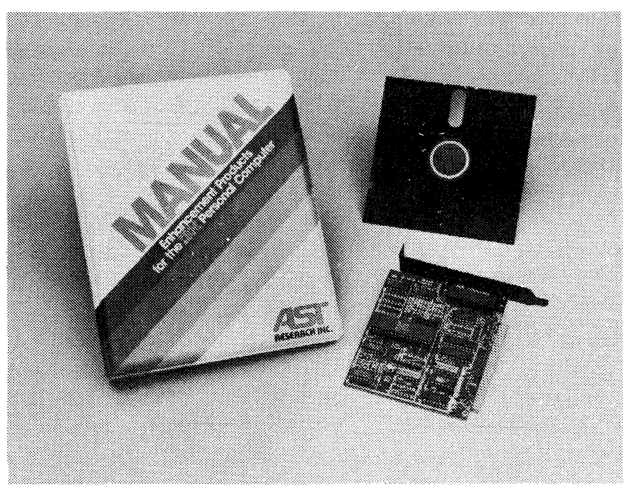
One of the biggest problems users face when using terminal emulation products concerns the special keystrokes an operator must learn to produce capabilities not normally supported on a particular terminal. Terminal operators accustomed to the keystrokes of a particular terminal must learn a new set of keystrokes to effect the functions of the emulated terminal. This operation can be compared to typing in Arabic on a typewriter with an English keyboard and an Arabic font. (Type a "g" and another symbol appears on the paper.) Because this kind of operation can cause confusion, vendors usually provide key maps that show keystroke equivalents between the emulated terminal and the various emulating devices. Some vendors also provide stick-on decals for emulating keyboards.

Many users are purchasing these terminal controllers to allow non-IBM devices in remote locations to access IBM mainframes. Remote cluster controllers eliminate the need for dedicating one terminal (e.g., a 3270) to one application, and another terminal at the same site to a local minicomputer. Many remote controllers have one synchronous line for 3270 access and two or more minicomputer interfaces. Terminals attached to the controller can switch between a remote host mainframe and the remote and local minicomputers in this type of configuration.

Users can configure most terminal controllers for dial-up access, allowing ASCII terminals in a remote location to dial into the local controller, which then makes the connection with a CPU located at the same or a third site. The controller eliminates the need for an IBM controller and additional synchronous lines to access the mainframe. A prominent cluster controller vendor, Datastream Communications, Inc., offers several models, including the 774 and the 776. The Model 776 operates in a point-to-point, multipoint or switched BSC network and acts as, and replaces, an IBM 3271/3276 cluster controller.

Units that handle conversions to make microcomputers and personal computers compatible with IBM mainframes represent a large and growing area in the conversion/emulation marketplace. Organizations are using more and more microcomputers for decentralized applications, but in many instances microcomputer users must have access to a centralized database, which generally resides on an IBM mainframe. Users can establish a micro-to-mainframe link through an emulation package that typically includes a diskette containing the emulation logic and a communications circuit board that is installed inside the microcomputer. An example of this type of product is DCA's Irma, one of the most popular micro-to-mainframe interfaces. The Irma is an IBM PC board with a coaxial interface that connects the PC to an IBM 3270 terminal controller that accesses a mainframe. With Irma installed and running on the PC, users can download data from the mainframe to the microcomputer, where it is viewed on the microcomputer screen. Like other forms of emulation, micro-to-mainframe links usually specify the microcomputers supported and the host ports and/or peripherals to which they can be connected. The Irma, for example, must attach to an IBM Personal Computer or compatible microcomputer and will attached only to an IBM or compatible 3274/3276 terminal controller. Other emulators provide IBM 2780/3780 batch terminal emulation for specified micros and minicomputers.

Other types of emulation products offer conversions that are the reverse of the popular ASCII-to-3270 conversion. For example, Protocol Computers' new 74D unit lets an



AST Research Inc. supplies a number of communications packages that allow IBM PCs to emulate 3270 or 2770 terminals. The communications packages consist of diskette-resident software and a circuit board that fits into the PC.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

IBM 3270-type device talk to an ASCII host. The 74D interfaces with an IBM communications controller, an IBM 3270 cluster controller, and up to six ASCII hosts, which can be DEC hosts, public networks, and personal computers. Supported by the 74D, the 3270 can switch between SDLC and DEC hosts.

Conversion and emulation in a data communications network can occur in many different devices and at many points in a network. Converters can be separate hardware units placed between a terminal and a modem; shared hardware units that handle other functions (e.g., front-end processors); devices that replace cluster controllers; interface cards in a personal computer; or applications programs, specialized emulation software packages or software/hardware resident on minicomputers, mainframes, and PBX systems. Many network services, such as Tymnet and Telenet, also offer conversion as part of their value-added products. Even phone companies offer conversion as part of their network services. One recent controversy between GTE Telenet and Southern Bell concerns the BOC's inclusion of protocol conversion in its LADT service. Value-added carriers would like to restrict such offerings by the BOCs, but the most recent Computer Inquiry rulings do not specifically prohibit the telephone companies from offering the service.

Protocol conversion and emulation products address problems of incompatibility among many types of data communications devices. But as you might have surmised from our discussion above, the majority of conversion units are designed specifically to incorporate incompatible devices in an IBM environment. In the next section of this report, we will discuss that environment in relation to conversion and emulation products.

The IBM 327X Environment

Tremendous growth in the minicomputer, microcomputer, and personal computer markets has led to a rapid increase in the number of installed ASCII asynchronous terminals that access these computers. However, ASCII devices cannot access information that resides on IBM mainframes. IBM's series of products that provide interactive communications in an IBM network is the IBM 3270 Information Display System. This series includes controllers, terminals, and printers that are dedicated to a single host and usually to a single application.

Components in the current 3270 system include: the 3277, 3278, 3279, 3178, and 3290 display terminals, 3284, 3286, 3287, 3288, and 3289 printers, and the 3274 and 3276 cluster controllers. Each component comes in various models. For example, the 3278 is a monochromatic display available in five models that essentially differ only in their screen capacities. The 3279 is a color display version of the 3278. The 3274 controller comes in various models that handle up to 12, 16, or 32 attached terminals, local or remote host connection, BSC or SDLC protocol. The 3276 is a smaller controller designed for clusters of up to eight terminals.

Because of the 3270's huge installed base, many models not longer actively marketed by IBM continue to play a significant role in the IBM-compatible markets, particularly the 3271 and 3272 controllers. The 3271 is a remote cluster controller that handles up to 32 terminals and comes in BSC and SDLC versions. The 3272 is a local channel-attached version of the 3271.

There are some shortcomings to using products in the 3270 family. First, they are expensive. Second, many of the IBM components are physically larger and take up more space than the ASCII terminals and the emulators that can be used in their place.

To acknowledge the need for asynchronous communication, in 1979 IBM introduced a Model 3101 terminal, which can attach directly to a 3705 communications controller and participate in ASCII applications resident in the host. The company also introduced a four-port protocol converter, the 7426, to allow the 3101 to appear as a 3278 to the 8100 and 4300 Series computers. A new Yale ASCII Communications System software package lets almost any ASCII device access 3270 applications and appear as a 3270 terminal. Most importantly, however, IBM introduced 3270 emulation support for most of its mini- and microprocessor-based products including the IBM PC, the System/36, and the Displaywriter. In doing so, IBM, in effect, changed the 3270 from a single-host, dedicated terminal system to a protocol that can accommodate many different devices.

Although the majority of protocol converters and terminal controllers on the market today handle some type of conversion between ASCII devices and IBM units, other products handle conversion between IBM BSC protocols to IBM SDLC Protocols. This conversion is particularly useful to users of older IBM BSC equipment who wish to migrate to an SNA/SDLC environment without replacing all of their old equipment. BSC-to-SDLC conversions generally occur between BSC 2780/3780 RJE or 3270 BSC protocols and SDLC protocols.

As IBM PCs become increasingly prevalent in organizations, products to provide micro-to-mainframe compatibility will become more and more important. The entire protocol/emulation market is exploding today because units that make ASCII terminals and personal computers compatible with SNA/SDLC networks are in tremendous demand. ASCII devices provide flexible and inexpensive solutions to network problems, but IBM's mainframes are still the de facto standard for centralized computer facilities that must handle large databases and many applications. It seems unlikely that this situation will change soon, and vendors that offer conversion products to handle ASCII-to-IBM conversions should enjoy a huge market for their products. In fact, current trends in the protocol conversion and terminal emulation marketplace reflect an increasingly heated competition among data communications vendors for a share of the potentially large and lucrative conversion market.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

CURRENT TRENDS

Many different products handle some type of conversion to provide compatibility between communications devices. Presently, a number of large data communications equipment vendors are incorporating protocol converters and terminal controllers into their general line of products. Micom Systems, for example, recently acquired Industrial Computer Controls, Inc., one of the oldest specialized protocol converter manufacturers in the marketplace. At the same time, Micom introduced the Micro7400 protocol conversion, a replacement of ICCI's CA20 converter. The Micro7400 handles ASCII-to-3270 conversion and provides network monitoring and control functions as well.

Formerly the venue of small companies like Protocol Computers, Inc. and Innovative Electronics, which specialize in standalone protocol converter products, protocol conversion has become incorporated into existing data communications products. We now find conversion as an integral capability in digital data switches, PBXs, personal computers, and word processors. And a market that in 1980 earned gross yearly revenues of about $5 million has in 1983 mushroomed into a $100 million a year business. Overnight, protocol converters have become the "hot" product in the ever-changing data communications environment.

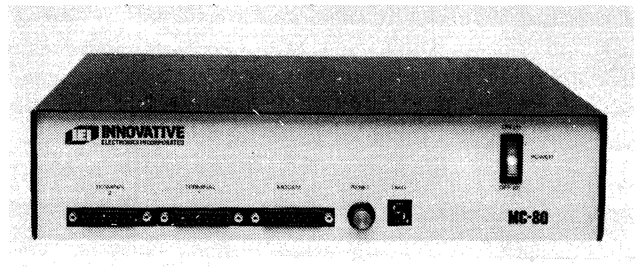
From a historical perspective, we can benchmark interest in protocol conversion at IBM's introduction of its 7426 converter in October 1982. With this announcement, IBM not only sanctioned conversion technology as a viable solution to network problems, but also focused industry attention on the technology.

Conversion products that facilitate LAN-to-LAN compatibility and access X.25 public networks are also expected to have a large market. We can expect to see a growing interest in PAD devices that effect X.25 access, and we can anticipate greater PBX conversion capabilities in the months ahead. Conversion offerings from value-added carriers, such as Tymnet and Telenet, and from the BOCs will also grow as data communications moves further into the home markets where personal computer users are becoming more interested in linking into public networks and databases.

Until the data communications industry adapts and uses worldwide protocol standards to link equipment, protocol converters and emulators will remain an important part of the general market. It is unlikely that such standardization will occur in the very near future.

CHOOSING CONVERSION DEVICES

Before choosing a conversion unit, users should consider some of the negative characteristics of the devices. First, protocol converters will cause delays in response time on the network because data must flow into the converter's buffer before transmission. If data backs up in the buffer, overruns occur; if the buffer is small, the converter can lose data.



Innovative Electronics' MC-80 Series protocol converters include many models that provide various conversions: IBM 2780/3780 to ASCII, Burroughs Poll/Select to ASCII, IBM 3270 to Burroughs Poll/Select, and others.

Terminal operators dealing with devices emulating other products may have problems learning the new key sequences and key functions necessary to the emulation process. Thus, organizations can expect some decreased productivity during the initial months of a conversion installation.

In addition, protocol converters usually do not offer the security provided by, for example, the IBM 3270-type devices. Organizations must deal with the problem of protecting sensitive data, particularly in dial-up applications.

When an organization decides to install conversion products in a data communications network, it should determine exactly what kinds of conversions are needed to solve particular incompatibilities. Once this is established, users should determine which kind of products can handle the conversion most effectively in a particular application. This can be an extremely confusing task because there are so many conversion products available. To narrow choices, it is wise to contact many vendors and ask for product specifications and documentation that explains how a product operates. When studying specifications and operating procedures, users must note exactly what types of terminals, controllers, or hosts are supported by the device because most converters and controllers support specific products rather than a general range of devices. For example, a protocol converter specifically designed for IBM 3277 emulation might not work with a 3278 application.

Also important is finding out what added features and functions the converter handles. Does it support more than one host? Does it replace an IBM controller, or is it used in conjunction with a controller? Does the device incorporate any multiplexing or concentrating? Can the network manager monitor the network via the converter? If additional features are available, are they standard or optional? What cost savings will it represent in your overall networking scheme?

After narrowing their choices to the products of several vendors, users should ask the company to provide an in-house demonstration of the product. A prospective buyer

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

should also request a list of current users who will discuss their experiences with the product. These individuals can provide information about the advantages and disadvantages of the product, hardware reliability, and the type and quality of support provided by the vendor.

IBM mainframe users in particular should find out whether conversion equipment can be upgraded as IBM upgrades and changes its SNA architecture.

After further narrowing the selection to two or three vendors, users should request a free trial of the product. By using a converter in a particular application, prospective buyers can soon find out whether a product provides the desired compatibility in an efficient manner.

USER EXPERIENCE

In July 1983, Datapro conducted a Terminal Users Survey, which was based on results received from questionnaires mailed to a cross-section of *Data Communications* magazine subscribers. Several questions in the survey pertained to the use of protocol conversion devices in a network. Below we show the results of these questions and draw some conclusions from the information.

We asked the users to indicate the primary protocols supported by their terminals:

	Number of Responses	Percent of Total Responses
Asynchronous	274	70
IBM BSC	187	47
IBM SDLC	130	33
Other bit-oriented synchronous protocol (e.g., ANSI ADCCP, ISO HDLC, Sperry UDLC, or Burroughs BDLC)	44	11
X.25 packet-level	34	9
Other byte-oriented synchronous protocol (e.g., DEC DDCMP)	35	9
Other	22	6

Although we are not in a position to draw any formal conclusions, since this year's user sample consists of different respondents from last year's, some interesting observations can be made when the two years' responses are compared. (The size of the respondent group is approximately the same: 447 respondents in 1982 versus 404 respondents for 1983.) Asynchronous protocol users have increased from 62 percent in 1982 to 70 percent in 1983, indicating an increased use of ASCII terminals, while users of IBM BSC and other byte-oriented protocols have decreased from 65 to 56 percent. One possible explanation may be the increasing use of protocol conversion in IBM environments. The use of IBM SDLC has remained nearly steady—34 percent for 1982 compared to 33 percent for 1983, and the high number of users who indicated using multiple protocols in their networks suggests that many of these users are still in various stages of migration to SNA and in no hurry to complete it. While the number of X.25 packet-level users remains small, the percentage has more than doubled since last year (4 percent in 1982 versus 9 percent in 1983), indicating a steadily growing use of packet switching networks.

We also asked these users to identify any types of protocol conversion that was being performed for their applications:

	Number of Responses	Percent of Total Responses
ASCII-to-BSC	107	27
ASCII-to-SDLC	75	19
BSC-to-SDLC	33	8
Other	27	6

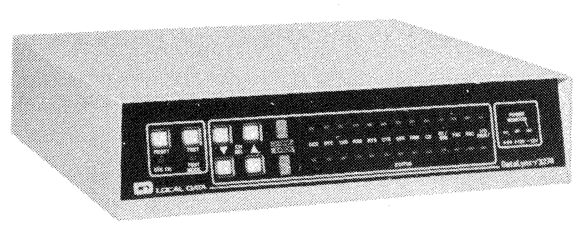
. . . and by what means this protocol conversion is performed:

	Number of Responses	Percent of Total Responses
Software loaded onto an existing system such as a general-purpose computer, front-end processor, terminal controller, or PBX system	107	27
Dedicated protocol converter	96	24
Value-added network service (e.g., Telenet)	22	6
Other	2	1

The strength of the protocol conversion market that has emerged in the past two years is certainly confirmed by these users' responses. Although these two questions were new on the 1983 questionnaire, and thus we have no comparative data from last year, the heavy use of ASCII-to-BSC conversion seems to bear out the conclusions concerning the increased use of asynchronous terminals we noted earlier in this report. Most likely, a great many of the ASCII-to-BSC conversions involve ASCII terminals emulating 3270-type devices.

VENDORS

Listed below are the names, headquarters addresses, and telephone numbers of vendors who manufacture conversion devices and terminal controllers. We have provided this list so that readers can contact the vendors for more information about the products they offer.



Local Data's DataLynx/3274 provides asynchronous ASCII-to-BSC or SDLC conversion. The unit emulates an IBM 3274 or 3276 controller and allows ASCII CRTs to emulate IBM 3278 terminal displays.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

Agile Corporation, 1050 Stewart Drive, Sunnyvale, CA 94086. Telephone (408) 735-9904.

Air Land Systems Corporation, 2710 Prosperity Avenue, Fairfax, VA 22031. Telephone (703) 573-1100.

Alphamatrix Inc., 1021 Millcreek Drive, Feasterville, PA 19047. Telephone (215) 355-3297.

American Satellite Corp., 20301 Century Boulevard, Germantown, MD 20767. Telephone (301) 428-6041.

Ark Electronic Products, 325 West Hibiscus Boulevard, Melbourne, FL 32901. Telephone (305) 573-1100.

Associated Computer Consultants, 228 East Cota Street, Santa Barbara, CA 93101. Telephone (805) 963-8801.

AST Research, 2121 Alton Avenue, Irvine, CA 92714. Telephone (714) 540-1333.

Atlantic Research Corporation, 5390 Cherokee Avenue, Alexandria, VA 22314. Telephone (703) 642-4000.

Avanti Communications Corporation, Aquidneck Industrial Park, Newport, RI 02840. Telephone (401) 849-4660.

Backus Data Systems, Inc., 1440 Koll Circle, San Jose, CA 95112. Telephone (408) 279-8711.

Black Box Catalog, P.O. Box 12800, Pittsburgh, PA 15241. Telephone (412) 746-2910.

Carterfone Communications Corp., 1111 West Mockingbird Lane, Suite 1400, Dallas, TX 75247. Telephone (214) 630-9700.

CDI, P.O. Box 2043, 290 Huyler Street, South Hackensack, NJ 07606. Telephone (201) 489-8172.

Commercial Data Processing, Inc., 2241 Grand Avenue, St. Louis, MO 63104. Telephone (314) 776-1130.

CompreComm, Inc., 3200 North Farber Drive, Box 357, Champaign, IL 61821. Telephone (217) 352-4277.

Computer Peripheral Systems, Inc., 3870 North Peachtree Road, Atlanta, GA 30341. Telephone (404) 451-4005.

Computerwise, Inc., 4006 East 137th Terrace, Grandview, MO 64030. Telephone (816) 765-3330.

Computrol, Inc., 10820 Sunset Office Drive, St. Louis, MO 63127. Telephone (314) 965-2206.

Com/Tech Systems, Inc., 505 Eighth Avenue, New York, NY 10018. Telephone (212) 594-5377.

Control Concepts, 12004-B, Balls Ford Road, Manassas, VA 22110. Telephone (703) 361-5545.

Convergent Technologies, 3055 Patrick Henry Drive, Santa Clara, CA 95050. Telephone (408) 980-0850.

CTiData Corp., 5275 North Boulevard, Raleigh, NC 27604. Telephone (919) 876-8731.

Data General Corporation, 4400 Computer Drive, Westboro, MA 01580. Telephone (617) 366-8911.

Datagram Corporation, 11 Main Street, East Greenwich, RI 02818. Telephone (401) 885-4840.

Dataprobe, 110 West Palisades Boulevard, Palisades Park, NJ 07650. Telephone (201) 489-5588.

Datastream Communications, Inc., 1115 Space Park Drive, Santa Clara, CA 95050. Telephone (408) 727-2980.

Datatel, Inc., 1008 Astoria Boulevard, Cherry Hill, NJ 08034. Telephone (609) 424-4451.

Davox, 6 Continental Boulevard, Merrimack, NH 03054. Telephone (603) 424-4500.

Digital Communications Associates, 303 Technology Park, Norcross, GA 30092. Telephone (404) 448-1400.

Diversified Data Resources, Inc., 25 Mitchell Boulevard, Suite 7, San Rafael, CA 94903. Telephone (415) 499-8870.

Duracom Corporation, 7300 North Crescent Boulevard, Pennsauken, NJ 08110. Telephone (609) 662-7277.

Dynatech Packet Technology, Inc., 6464-G General Green Way, Alexandria, VA 22312. Telephone (703) 642-9391.

Four-Phase Systems, 10700 North DeAnza Boulevard, Cupertino, CA 95014. Telephone (408) 255-0900.

Gandalf Data, Inc., 1019 Noel Avenue, Wheeling, IL 60090. Telephone (312) 541-6060.

GTE Telenet Communications Corporation, 8229 Boone Boulevard, Vienna, VA 22180. Telephone (703) 442-1000.

Halcyon Communications, 2121 Zanker Road, San Jose, CA 95131. Telephone (408) 293-9970.

Harris Corporation, Data Communications Division, 16001 Dallas Parkway, P.O. Box 400010, Dallas, TX 75240. Telephone (214) 386-2550.

Icot Corporation, 830 Maude Avenue, Mountain View, CA 94043. Telephone (415) 964-4635.

Innovative Electronics, Inc., 4714 NW 165th Street, Miami, FL 33014. Telephone (305) 624-1644.

Instrument Services Incorporated, 957 Winnetka Avenue North, Minneapolis, MN 55427. Telephone (612) 544-8916.

InteCom, Inc., 601 Intecom Drive, Allen, TX 75002. Telephone (214) 422-5450.

International Business Machines Corporation, Old Orchard Road, Armonk, NY 10504. Contact your local IBM representative.

International Entry Systems, Inc., 408 N.E. 72nd Street, Seattle, WA 98115. Telephone (800) 426-7740; in Washington (206) 525-6800.

Intersil Systems, Inc., 1275 Hammerwood Avenue, Sunnyvale, CA 94086. Telephone (408) 743-4300.

Kaufman Research Manufacturing, Inc., 145 East Dana Street, Mountain View, CA 94041. Telephone (415) 962-8811.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

KMW Systems, 8307 Highway 71 West, Austin, TX 78735. Telephone (512) 288–1453.

Lee Data Corporation, 7075 Flying Cloud Drive, Eden Prairie, MN 55344. Telephone (612) 828–0300.

Level Two Systems, 14 Dudley Street, Reading, MA 01867. Telephone (617) 942–0075.

Local Data, Inc., 2701 Toledo Street, Suite 706, Torrance, CA 90503. Telephone (213) 320–7126.

May-Craft Information Systems, 3412 Beltwood Parkway South, Dallas, TX 75234. Telephone (214) 392–3766.

Memorex Corporation, San Tomas at Central Expressway, Santa Clara, CA 95052. Telephone (408) 987–1000.

Memotec Data, Inc., 4940 Fisher, St. Laurent, Quebec. Telephone (514) 738–4781.

Micom Systems, Inc., 20151 Nordhoff Street, Chatsworth, CA 91311. Telephone (213) 998–8844.

Microcom, Inc., 1400A Providence Highway, Norwood, MA 02062. Telephone (617) 762–9310.

Modemsplus, Inc., 217 East Trinity Place, P.O. Box 1727, Decatur, GA 30031. Telephone (404) 378–5276.

Northstar Computers, Inc., 14440 Catalina Street, San Leandro, CA 94577. Telephone (415) 357–8500.

NuData Corporation, 32 Fairview Avenue, P.O. Box 125, Little Silver, NJ 07739. Telephone (201) 842–5757.

Perle Systems Ltd., 360 Tapscott Road, Scarborough, Ontario M1B 3C4. Telephone (416) 299–4999.

Peripheral Technology, Inc., 14784 N.E. 95th Street, Redmond, VA 98052. Telephone (800) 822–2208; in Virginia (206) 881–6691.

Phone 1, Inc., 461 North Mulford Road, Rockford, IL 61107. Telephone (815) 397–8110.

Protocol Computers, Inc., 6150 Canoga Avenue, Woodland Hills, CA 91367–3773. Telephone (800) 423–5904; in California (213) 716–5500.

ProtoComm Devices, 207 Atlantic Street, Stamford, CT 06901–3504. Telephone (203) 327–6893.

Pulsecom Division, Harvey Hubbell Incorporated, 2900 Towerview Road, Herndon, VA 22071. Telephone (703) 471–2900.

Racal-Milgo, 8600 N.W. 41st Street, Miami, FL 33166. Telephone (305) 592–8600.

Racal-Telesystems, 401 North Michigan Avenue, Chicago, IL 60611. Telephone (312) 329–0700.

Radio Shack, A Division of Tandy Corporation, 300 One Tandy Center, Fort Worth, TX 76102. Telephone (817) 390–2140.

Remark Datacomm, Four Sycamore Drive, Woodbury, CT 11797. Telephone (516) 367–3806.

Renex Corporation, 6641 Backlick Road, Suite 210, Springfield, VA 22150. Telephone (703) 569–0607.

Rixon Inc., 2120 Industrial Parkway, Silver Spring, MD 20904. Telephone (301) 622–2121.

Shaffstall Corporation, P.O. Box 50990, Indianapolis, IN 46250. Telephone (317) 842–2077.

Sytek, Inc., 1225 Charleston Road, Mountain View, CA 94043. Telephone (414) 966–7300.

Techland Systems, Inc., 39 Carwall Avenue, Mount Vernon, NY 10552. Telephone (914) 699–8467.

Tektronix, Inc., Mailing Station 63–635, P.O. Box 1700, Beaverton, OR 97077. Telephone (910) 467–8708.

Teleprocessing Products, 4565 East Industrial Street, Building 7K, Simi Valley, CA 93063. Telephone (805) 522–8149.

Teltone Corporation, P.O. Box 657, 10801 120th Avenue Northeast, Kirkland, WA 98033–0657. Telephone (206) 827–9629.

Thomas Engineering, 1040 Oak Grove Road, Suite 106, Concord, CA 94518. Telephone (415) 680–8640.

Timeplex, Inc., 400 Chestnut Ridge Road, Woodcliff Lake, NJ 07675. Telephone (201) 391–1111.

3R Computers, 18 Lyman Street, Westboro, MA 01581. Telephone (617) 366–5300.

Tri-Communications Industries, 25 Van Zant Street, East Norwalk, CT 06855. Telephone (203) 866–1154.

Tri-Data Corporation, 505 East Middlefield Road, Mountain View, CA 94043. Telephone (415) 696–3700.

Versitron Inc., 6310 Chillum Place, N.W., Washington, DC 20011. Telephone (202) 882–8464.

Winterhalter Incorporated, P.O. Box 2180, Ann Arbor, MI 48106. Telephone (313) 662–2002.

ZBX Telecomputers, 8110 Trans Canada Highway, St. Laurent, Quebec H4S 1M5. Telephone (514) 331–9102.

Asynchronous/Synchronous Conversion Devices

The following manufacturers offer standalone devices dedicated to asynchronous-to-synchronous conversion.

Avanti Communications Corporation, Aquidneck Industrial Park, Newport, RI 02840. Telephone (401) 849–4660.

Com/Tech Systems, Inc., 505 Eighth Avenue, New York, NY 10018. Telephone (212) 594–5377.

Digital Controls Corp., 2779 Orchard Road, Dayton, OH 45449. Telephone (513) 435–5455.

General Datacomm Industries, One Kennedy Avenue, Danbury, CT 06810. Telephone (203) 797–0711.

Intertel, Inc., Six Vine Brook Park, Burlington, MA 01803. Telephone (617) 681–0600.

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED

Conversion and Emulation in Data Communications

▷ **Modemsplus, Inc.,** 217 East Trinity Place, Box 1727, Decatur, GA 30031. Telephone (404) 378-5276.

Paradyne Corporation, 8550 Ulmerton Road, Largo, FL 33541. Telephone (813) 530-2000.

Teleprocessing Products, Inc., 4565 East Industrial Street, Building 7K, Simi Valley, CA 93063. Telephone (805) 522-8149.

Universal Data Systems, 500 Bradford Drive, Huntsville, AL 35805. Telephone (205) 837-8100. □

© 1984 DATAPRO RESEARCH CORPORATION, DELRAN, NJ 08075 USA
REPRODUCTION PROHIBITED