COMM180 Technical Manual

Release 1.0

The Micromint Inc.
25 Terrace Drive
Vernon, CT 06066

Copyright Notice

Disclaimer

The Micromint Inc. makes no representations or warranties with respect to the contents hereof. Further, changes are periodically made to the information contained herein. The Micromint Inc. reserves the right to incorporate these changes in new editions of this publication without obligation to notify any person of such revisions or changes. Mention in this document of specific product(s) compatible with the COMM180 does not constitute an endorsement of the product(s); rather the information regarding specific product(s) is given for illustrative purposes.

Trademarks

CP/M, CP/M-80 and MP/M are registered trademarks of Digital Research, Inc. MS-DOS is a registered trademark of Microsoft, Inc. Z80 is a registered trademark of Zilog, Inc. IBM, IBM PC, and IBM Personal Computer are trademarks of the International Business Machines Corporation. Z-System is a registered trademark of Eschelon, Inc. ZCPR is a copyright of R.L. Conn. MOSART and Statistical Equalization Algorithm are registered trademarks of Xecom, Inc. SB180 and COMM180 are copyrights of The Micromint, Inc.

Date of this release: November, 1985

* *

WARRANTY

The Micromint, Inc. extends the following warranty:

A factory manufactured circuit board or assembly carries with it a six month warranty covering both parts and labor. Any unit which is found to have a defect in materials or workmanship will be repaired or replaced at the option of The Micromint, Inc.

No credit will be given for units which show damage due to user modification or neglect.

Units returned for repair must have prior authorization from The Micromint, Inc. A return authorization number may be obtained by phone or letter. Please retain a record of the return authorization number as most subsequent correspondence will refer to that number. Under no circumstances is any product to be returned to The Micromint, Inc. without prior authorization. The Micromint, Inc. will assume no responsibility for unauthorized returns.

All returns must be shipped prepaid. Insurance is recommended as losses by a shipping carrier are not the responsibility of The Micromint, Inc. Repaired units will be returned with postage paid.

For repair of units which have expired their warranty, a minimum inspection fee must be prepaid. Contact The Micromint, Inc. for information on current minimum charges.

Return authorization must be obtained prior to any return.

The Micromint, Inc. reserves the right to change any feature or specification at any time.

* *

TABLE OF CONTENTS

LIST OF FIGURES

COMM180 Errata Sheet

COMM180

1) Contrary to what is described in step 4 on page 4, mounting brackets are not supplied with the COMM180 board.

COMM180-S

1) On both the Xebec 1410A and the Xebec Owl, there is a three pin jumper header labeled 'SS' with the option of placing the jumper next to a '2' or a '5'. The sector size ('SS') that the COMM180-S software uses is 512 bytes, so the jumper should be placed next to the '5' (512 byte sectors) instead of the '2' (256 byte sectors).

2) The COMM180-S software expects the hard disk controller board to be set-up for a target ID of zero. Consult your controller board's manual to find out how to jumper the board for an ID of zero.

1.Ø COMM18Ø Overview

The Micromint COMM18Ø expansion board adds two major functions to your SB18Ø:

1) Bell 1Ø3-212A compatible 3ØØ/12ØØ baud modem with Dual Tone Multi-Frequency (DTMF) encode/decode capability and voice synthesis
2) SCSI hard disk controller interface

The COMM18Ø board is available in three versions:

1) Modem, DTMF encode/decode, optional voice synthesis
2) Small Computer System Interface (SCSI) hard disk controller interface alone
3) Both of the above

And either the modem version or the SCSI controller version may be upgraded to the full version at any time.

The COMM18Ø does not use a serial port on the SB18Ø, but rather it adds a third serial port with direct connection to the phone line. It is fully Bell 1Ø3 and Bell 212A compatible (including FCC registered Data Access Arrangement) for 3ØØ and 12ØØ baud use, is 8251A software compatible, features both DTMF and pulse dialing, call progress monitoring, DTMF reception and decoding, and a unique diagnostic capability which automatically compensates for common telephone line deficiencies. In addition the COMM18Ø has optional voice synthesis capabilities which allow it to respond verbally to commands entered via the standard touch tone telephone pad.

The SCSI allows the use of a wide variety of hard disks with the SB18Ø for fast, sophisticated mass storage whether you need just 5 megabytes or 5Ø megabytes. Many manufacturers offer hard disk drives and controller cards which mate with the SCSI interface. If you need more file space than floppies allow, the COMM18Ø board can help meet your storage needs. In addition, many laboratory instruments support the SCSI standard, thus the COMM18Ø can allow the SB18Ø to be more easily used for data logging and data reduction.

* *

* NOTE: This manual describes the operation of the modem, DTMF, and voice synthesis section of the COMM18Ø ONLY. The SCSI section of the COMM18Ø is described in its own manual. *

* *

The COMM180 has the following features:

Modem

- Plugs into the Expansion Bus on the SB180
- Only 4" by 5.25"
- Fully Bell-212A and Bell-103 Compatible
- DTMF or Pulse Dialing
- Jack for External Speaker for Call Progress Monitoring
- DTMF Reception and Decoding
- 8251A Software Compatibility
- Parity Generation/Checking
- Sync Byte Detection/Insertion
- Synchronous 1200 bps, Asynchronous 1200,300,110 bps
- Software Controlled Audio Input and Output Interface (2 Separate Jacks) for Voice Communication or Acoustic Coupling
- Voice Synthesis (LPC coded)
- ASCII Command and Error/Status Codes
- Extensive Built in Diagnostics
- Phone Line Diagnostics
- FCC Registered Direct Connection. Tip and Ring Input
- Operates on +5/+12 Volt Power Supply

SCSI

- Provides a Device Independent Local I/O Bus
- Supports Initiator and Target Roles
- Parity Generation with Optional Checking
- Supports Bus Arbitration
- Provides Direct Control of All Bus Signals
- Supports the two most popular controller cards, the standard Xebec 1410 controller and the high performance Adaptec ACB4000
- Complete source code for the standard 5 MB, 10 MB and high capacity 20 MB 5.25" hard disks

COMM180 Modem Software

The software which comes with the COMM180 consists of TERM III, with Z-MSG optionally available.

TERM III is a sophisticated communications package which offers all the functions of standard modem programs but goes far beyond them when used with COMM180's advanced features and Z-System DOS (required for operation). TERM III was designed to be used as:

1) an originating communications system to allow the user to dial out, communicate with other computers, and perform file transfer functions;
2) a remote access system to allow users to dial into the system, interact with it, and transfer programs into and out of it; and
3) a configuration system to allow the user to configure the attributes of the other two types of systems.

TERM III offers multiple file transfer protocols: Christiansen's MODEM7 (with checksum and with CRC), MODEM7 batch, XMODEM, KERMIT, CompuServe's CIS, and X-ON/X-OFF. Special attention has been paid to making the remote access system completely secure from unauthorized use. You may also call in to TERM III on the COMM180 and use a standard touch tone pad to give special instructions, e.g., run a program using the COMM180's speech synthesis to give a verbal summary of system access.

Z-MSG allows the COMM180, SB180, and TERM III to be custom configured as a "turnkey" Remote Bulletin Board System - either as a public system allowing access to anyone or as a private system restricting access to "members only". Z-MSG allows up to eight user "types" with varying privileges associated with each. Messages may be public or private and may be over 100 lines long. Extensive editing functions are provided and comprehensive on-line help is always available.

SCSI Software

For those wishing to use the SCSI as a hard disk controller, source code is provided which allows many types of hard disk controllers and sizes of Winchester disks to be used as mass storage for the SB180. Two popular controllers are supported directly - the Xebec 1410 controller and the high performance Adaptec ACB4000. Three disk sizes are supported - the standard 5 MB and 10 MB drives as well as the high capacity 20 MB drive. Of course, the SCSI may also be used to link other SB180's or SCSI equipment together as well.

2.0 COMM180 Installation

The COMM180 is connected to the SB180 through the SB180's expansion bus. This expansion bus is accessed through the 40 pin connector J5 on the SB180 board. The COMM180 board is mounted "piggyback" onto J5.

1) The connector on the bottom of the COMM180 board must fit onto the expansion connector J5 in the middle of the SB180. Position the SB180 so that the disk connector is at the top of the board. Hold the COMM180 so that the 50 pin connector is away from you.

2) The modem, terminal, and printer cables may have to be rerouted so the COMM180 sits properly on top of the SB180. The printer cable in particular may have to be folded so it doesn't interfere with the expansion connector. Experiment to find what works best for you.

3) Line up the two connectors and push the COMM180 onto the SB180's expansion header. The sides of the COMM180 board should line up exactly with the sides of the SB180. If they don't, take off the COMM180 and reposition it so the connectors fit together properly.

4) Plug the power cable into the four pin connector at the upper left side of the board. This is the same power connector as on the SB180 and both must be attached to a power source. Plug in any additional cords you may need (phone line, speaker, hard disk controller, or headset, depending on which options you purchased with you COMM180.)

5) Power up the computer and you are ready to try out your COMM180 board.

3.0 Hardware Technical Overview

3.1 What Is a Modem?

The word "modem" is an amalgam of the two words "modulator" and "demodulator". A modem (called the **originate** modem) converts (modulates) digital signals from a computer into analog signals which can be transmitted via a telephone line. At the other end of the phone line another modem (called the **answer** modem) converts (demodulates) the analog signals back into digital signals which are then fed into another computer. Modems which have the ability to communicate in both directions simultaneously operate in the **full-duplex** mode. The same communication in both directions but only one direction at a time is **half-duplex** operation. Communication in only one direction is **simplex** operation. These modes of operation can be likened to a television for simplex, a CB radio which has to be keyed to talk for half-duplex, and a telephone for full-duplex where both parties can talk at once.

Modems are generally characterized by the speed at which they transmit data. These data transmission rates are properly expressed in bits per second (bps), although the term "baud" is often used. Strictly speaking, "baud" measures the number of transitions in the state of the communications link, rather than the amount of data represented by these transmissions. A single change of state in some cases may represent multiple data bits, and therefore the data rate may not equal the baud rate.

Modems are commonly divided into four categories, based on their speed of transmission. The low speed modems are those operating at speeds from 0 to 600 bps. The medium speed modems operate from 1200 to 2400 bps. From about 3600 bps to around 16,000 bps are a group of modems generally called high speed, but still higher in speed are the wide band modems, which work at speeds from 19,200 bps on up. Most modems are generally classified according to which **Bell** (U.S.) or CCITT (European) standard to which they conform. Figure 3.0-1 shows the most popular low and medium speed standards used.

	STANDARD	SPEED	OPERATION	ENCODING TECHNIQUE
A	103	0-300 bps	Full-duplex	FSK
	201	1200 bps	Half-duplex	PSK
	202	1200 bps	Half-duplex	FSK
	212A	0-300 bps	Full-duplex	FSK
		1200 bps	"	PSK
B	V.21	0-300 bps	Full-duplex	FSK
	V.22	1200 bps	Full-duplex	PSK
	V.23	1200 bps	Half-duplex	FSK
		75 bps	"	
	V.26	2400 bps	Half-duplex	PSK

Figure 3.0-1 Popular Bell (A) and CCITT (B) Standards

The higher the data rate, the greater the price of the modem and the higher the quality of the telephone line necessary for reliable operation. Low and medium speed modems generally use voice grade telephone lines, but the higher speed units require dedicated communication grade lines. And as the speed of data communications increases, the techniques required to ensure error-free reception become, by necessity, more sophisticated.

3.2 How Modems Work

Many types of encoding formats are used in modems to modulate the digital information into a form that can be sent through telephone lines. The two most popular are FSK and PSK.

Low speed modems generally employ a technique called frequency-shift keying (FSK) which uses two distinct tones of different frequency to represent logic 1 and Ø. Data is sent by the modem's alternately transmitting the two frequencies (i.e., shifting the frequency of its transmitted carrier tone). With the FSK scheme the higher frequency is known as the mark frequency and the lower is known as the space. In full-duplex systems two mark/space frequency pairs must be used, one for answer mode and another for originate. This is necessary because of the simultaneous two way communication for full-duplex operation.

The amount of information that can be sent using FSK in a given interval of time is limited by the frequency bandwidth of the telephone line: a transmitted data bit must consist of at least the number of cycles of a 1 or Ø tone required for the receiver to recognize it, and the number of cycles of the transmitted tone taking place in the time interval is the same thing as its frequency. The frequency used cannot exceed the capability of the line.

Higher speed modems use more complex and sophisticated transmission techniques, all of which to some extent modulate not only the frequency of the tones but their phase, and possible amplitude, as well. These phase-shift keying (PSK) methods permit more compact data encoding, with more information transmitted in less time, by making a single change in the state of the physical communication link communicate more than one data bit. (In such a technique, the data rate differs from the baud rate.) The most popular variation of PSK is called quadrature amplitude modulation, or QAM. Widely used in 12ØØ bps modems, QAM employs both amplitude and phase modulation to encode 2 bits of data in every state transition.

PSK operates in either synchronous or asynchronous formats. Synchronous systems use a timing signal from the computer to "clock" data out and maintain synchronization. In this format the stream of data has no synchronizing information itself. In asynchronous systems, there is no timing signal from the computer to the modem. In this case synchronization and timing information is derived from start and stop bits placed in the data stream surrounding each character. Timing is maintained by the modem inserting or removing stop bits.

3.3 DTMF Decoding

The keypad on a Touch Tone telephone receiver is a readily available, convenient means of tranmitting data. (Only telephone instruments from the Bell System are properly called Touch Tone; the generic term used by other manufacturers is **dual-tone, multiple-frequency,** or **DTMF,** signaling.) Pressing one of the keys on a DTMF-dialing receiver generates not a single-frequency sine wave but a combination of two frequencies. The 12 keys are arranged in four rows and three columns, as shown in Figure 3.2-1. All the keys in a given row or column have one tone in common. For example, pressing the digit "9" (row 3 and column 3) produces an 852 Hz and a 1477 Hz tone simultaneously. Similarly, pressing "4" (row 2 and column 1) produces 770 Hz and 1209 Hz tones simultaneously.

The full DTMF encoding standard defines four rows and four columns for a total of 16 two-tone combinations. Standard telephones use only 12 of these combinations, but the COMM180 can decode all 16. The eight frequencies associated with the rows and columns are separated into two groups. The low group, containing row information, has a range of 697 Hz to 941 Hz. The high group, containing column information, covers 1209 Hz to 1633 Hz.

		High Group			
		Col 0 1209 Hz	Col 1 1336 Hz	Col 2 1477 Hz	Col 3 1633 Hz
Low Group	Row 0 697 Hz	1	2	3	A
	Row 1 770 Hz	4	5	6	B
	Row 2 852 Hz	7	8	9	C
	Row 3 941 Hz	*	0	#	D

Figure 3.2-1 Dialing matrix of DTMF signaling system

The purpose of a DTMF receiver is to decode tones that indicate which key was pressed on the transmitter. The output from the decoder can be a logic pulse on one of 12 (or 16) output lines, a 4-bit binary code, separate 2-bit row and 2-bit column outputs, or even as the ASCII values of the codes themselves. The COMM180 uses the latter approach.

3.4 LPC Voice Synthesis

There are three techniques in general use to synthesize the human voice: waveform digitization, formant synthesis, and linear-predictive coding. They differ primarily in the number of bits per second (bps) of data required to construct a word.

Waveform digitization is an old technique which produces speech by generating a waveform with the time-domain characteristics of voice. The simplest form of this technique is uncompressed digital data recording, called PCM, for pulse-code modulation. In simple PCM recording, the analog speech waveform is sampled at a rate twice that of the frequency of the highest voice component and converted to digital format through an A/D (analog-to-digital) converter. Once stored, the digitial signal can be played back through a D/A (digital-to-analog) converter and a low-pass filter.

While PCM produces very natural sounding speech (since it is essentially a recorded voice), it requires very high data rates. Rates above 100k bps are not unusual with this method, and even data compression techniques cannot reduce the data rate to that offered by the other two methods.

Formant synthesis is essentially a modeling of the natural resonances of the human vocal tract. The bands of resonant frequencies defined are called formants. In an electronic synthesizer, these frequencies are generated by excitation sources and are then passed through variable-parameter filters.

One variation of the formant technique is called phoneme synthesis. In this, the spectral parameters are derived from basic sound units that make up words. A phoneme generator, in turn, reproduces these sounds. In such a circuit, each phoneme has been assigned a code, and the synthesizer module (or chip) utters the corresponding phoneme sound for each code it receives. Creation of continuous speech, therefore, is simply a matter of stringing the phonemes together.

In most cases, the electronic voice generated is quite intelligible, but it may have a mechanical quality about it. Continuous speech using phoneme synthesis can generally be generated with a data rate of less than 400 bps.

Linear-predictive coding is similar to formant synthesis. Both techniques are based in the frequency domain and use similar hardware to model the vocal tract. Rather than using a simple phoneme code, however, linear-predictive coding stores parameters for filter coefficients, gain, and excitation frequencies. The term "linear-predictive coding" refers to the programmed activities of the multi-stage lattice filters that produce the desired formants. Adequate quality speech can be achieved with data rates of 1200 to 2400 bps.

The COMM180 uses linear predictive coding for its speech synthesis. It comes with a built in vocabulary of 145 "words", including common words, letters, numbers, prefixes and suffixes which may all be combined to give a larger apparent vocabulary.

3.5 COMM18Ø Hardware

The COMM18Ø is based on a hybrid integrated circuit called a MOSART (MOdem Synchronous Asynchronous Receiver Transmitter) which plugs into a 4Ø pin DIP socket. Functionally, the MOSART can be divided into six sections: host interface, modem, analog circuits, DAA (Direct Access Arrangement), speech synthesizer, and microprocessor (see Figure 3.4-1). The MOSART can be viewed as a separate communication interface which looks to the SB18Ø like an 8251A USART chip, including the appearance of registers and interrupt lines. It functions as an 8251A but is not pin compatible with an 8251A (see Figure 3.4-2).

The modem is capable of providing both Bell 1Ø3 and Bell 212A mode synchronous or asynchronous operation. The analog section consists of switching circuits for routing data among the speech synthesizer, modem, audio inputs and outputs, the phone line, and both the modulator and demodulator. The FCC registered DAA on chip provides 1 ØØV isolation from the phone line, protection for the phone company's line, on/off hook control, and ring indication; it requires connection to just Tip and Ring on the phone line. (The Tip and Ring connections to the MOSART are not polarized, and the two signals may be interchanged.) The speech synthesizer is an LPC (Linear Predictive Coding) natural voice synthesizer. Data, in the form of ASCII and binary strings from a word table, is fed to it (as if it were an 8251A) from the SB18Ø. Custom vocabularies can be designed and implemented for special applications but as supplied includes a vocabulary of 145 words, letters, numbers, and phrases.

Interfacing to the MOSART is simple since it looks like an 8251A to the SB18Ø computer and the phone lines merely connect to the tip and ring input on the chip. Plugging the MOSART into an SB18Ø only requires the inclusion of address decoding logic (see Figure 3.4-3). The simplest circuit requires only a 74LSØ2 and a 74LS3Ø. Use of the interrupt lines is optional since the same signals are available from the MOSART's control register.

The phone line is simply wired to the tip and ring pins of the MOSART. Connecting a headset and microphone to allow you to hear or talk directly to the connected party involves the use of 3 resistors and 1 capacitor.

Exchanging data and command/status information with the MOSART is very straight forward. The device is initialized via a hardware reset line on power up, or by writing the initialization bit of the control register. The MOSART initializes similarly to the 8251A device it emulates (See Figure 4.3-1). After writing at least three bytes of hex Ø's to insure the MOSART is not stuck between initialization states, the initialization bit is written to the control register. The mode byte must be written next, followed by 1 or 2 sync bytes if synchronous transmission is being used. Lastly, the control byte is written. Once the MOSART is initialized, a new control byte can be set at any time by simply writing to the control register.

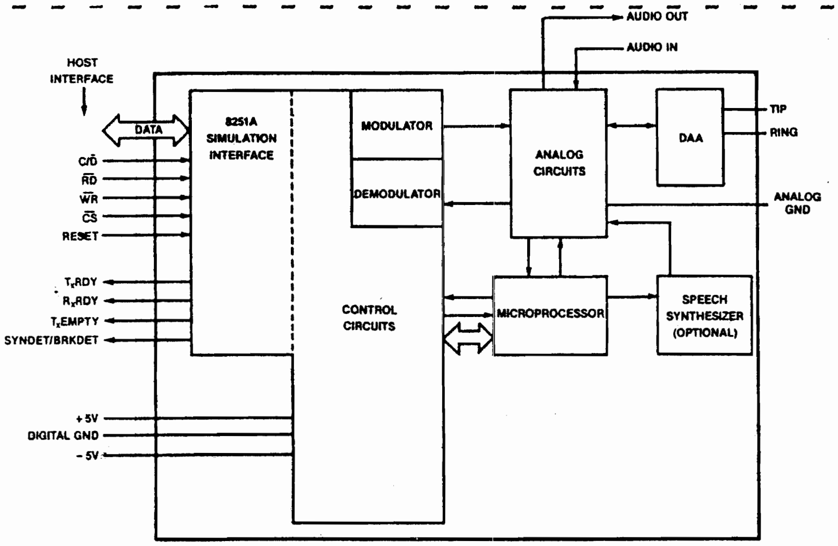


Figure 3.4-1 MOSART Block Diagram

CAUTION
PINS 17-24 HAVE 1500V ISOLATION FROM THE REST OF THE MODEM. THIS ISOLATION SHOULD BE PRESERVED THROUGHOUT THE SYSTEM.

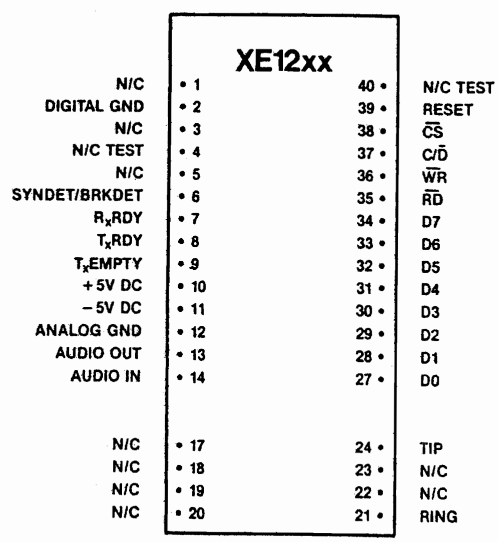


Figure 3.4-2 MOSART Pin Outs

All communication takes place through the MOSART's data port. A bit in the control byte called RTS allows the MOSART to differentiate between command function codes and data. Setting RTS to a logic 1 indicates that the information is to be accepted as data, and sent either to the modem or the speech synthesizer. An RTS bit of logic Ø indicates that the information written to the data port is to be interpreted as a MOSART command function code.

The MOSART outputs both data and command status codes to the host computer over the same port. The host computer interrogates a bit in the control register called DSR to determine if the received information is really data, or command status.

The command function codes the MOSART recognizes are broken up into 4 categories: modem connection, configuration, telephone control, and tests.

Modem connection functions include answering and originating modem connections, as well as monitoring for modem carrier vs voice. The configuration functions allow speed, framing, and parity selection, rotary vs DTMF dialing, and switching control over the synthesizer, external audio, and phone lines.

Telephone control functions handle dialing and DTMF. The commands consist of line hold, DTMF receive, dial digits, and the */# keys. In addition, the letters a-d can be used to send the DTMF codes for the normally unimplemented last row of the keyset. (Tone dialing must have been selected previously with the configuration command.) In addition, a detailed monitor function is included that will watch the phone line, and every .5 seconds return a code that indicates the frequency heard on the line (in tens of hertz). Other standard control functions like pausing, and waiting for dial tone are also included in the telephone control function set.

The test functions are perhaps the most unique to the MOSART. These functions include allowing answer or originate mode loopbacks, a line analysis feature that includes signal/noise, received carrier level, and carrier frequency error statistics, and a special analysis of 12ØØ bps mode phase error information.

4.Ø Using the MOSART

4.1 Host Interface

This tri-state, bi-directional 8 bit bus is the interface between the MOSART and the host. Data is transmitted or received upon execution of instructions from the SB18Ø. Command, Function, and Sync words and status information are also transferred through the Data Bus. The Command, Status, Data-In, and Data-Out registers are separate 8 bit registers communicating with the system bus through the Data Bus.

DØ-D7 DPort These eight pins provide the data bus for control and data values to and from the component. DØ is the low order bit

WR Write A "low" (Ø) on this input pin informs the MOSART that the SB18Ø is writing to the data or control address of the MOSART.

RD Read A "low" (Ø) on this input pin informs the MOSART that the SB18Ø is reading data or status information from the MOSART.

C/D Control/Data This Input pin selects either the control/ status register, C/D "high" (1), or the data register, C/D "low" (Ø).

CS Chip Select A "low" (Ø) on this Input pin selects the MOSART. When CS is "high" (1), the Data Bus is in the float state and RD, C/D and WR have no effect on the component.

	RD	WR	CS	C/D
Host Command Write	1	Ø	Ø	1
Host Status Read	Ø	1	Ø	1
Host Data Write	1	Ø	Ø	Ø
Host Data Read	Ø	1	Ø	Ø
Tri-State	X	X	1	X

Figure 4.1-1 MOSART Register Addressing

TxRDY Transmitter Ready A "high" (1) on this output pin signals the SB18Ø that the modem transmitter is ready to accept a data character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by TxEnable; or, for polled operation, the SB18Ø can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of WR when a data character is loaded from the SB18Ø. Note that when using the polled operation, the TxRDY status is not masked by TxEnable, but will only indicate the status of the transmit data buffer.

TxEMPTY **Transmitter Empty** When the MOSART has no characters to send, the TxEMPTY output pin will go "high" (1). It resets upon receiving a data character from the SB180. TxEMPTY remains "high" (1) when the transmitter is disabled. TxEMPTY can be used to indicate the completion of transmission.

In the synchronous mode, a "high" (1) on this output indicates that a character has not been loaded and that Sync character or characters are being transmitted automatically as fillers. TxEMPTY does not go low when the Sync characters are being transmitted.

RxRDY Receiver Ready This output pin indicates the MOSART contains a character that is ready to be output to the host. RxRDY can be connected to the interrupt structure of the host computer or, for polled operation, the host can check the condition of RxRDY using a Status Read.

SYNDET/BRKDET Sync Detect/Break Detect In Asynchronous Mode this output pin signals a break detect (BRKDET). It is held "high" (1) for the duration of the incoming break signal.

This output is used in Synchronous Mode for SYNDET and will go "high" (1) to indicate that the MOSART has located the Sync character in the Receive Mode. If the MOSART is in double character mode SYNDET is set high (1) after reception of the second sync character. SYNDET is automatically reset upon a Status Read.

RESET Reset A "high" (1) on this input pin forces the MOSART to "idle". The device will remain at "idle" until a new mode is set. Refer to the Mode Byte description below. Minimum reset time is 30 micro-seconds.

DIGITAL GROUND Power ground

TIP, RING Input/output pins are provided for direct connection to a two wire telephone line. Although the pins are named "TIP" and "RING" they are interchangeable. These lines are insulated from the rest of the MOSART system and this isolation must be maintained for safe operation.

AUDIO IN Input to an audio amplifier which can be switched by modem control inputs to perform one of the following functions:

1. Input audio signals from a microphone for voice input over the telephone line.
2. Input data signals from an acoustical coupler for connection to the modem demodulator.

AUDIO OUT Output from a second audio amplifier which can be switched by modem control inputs to perform one of the following functions:

1. Output audio signals to a headset earphone for voice communication over a telephone line.
2. Output data signals from the modem modulator to an acoustical coupler.
3. Output signals from the voice synthesizer to an external speaker, etc.

Note that the DTMF tone transmissions are internally suppressed and will not be heard on the audio output.

ANALOG GROUND Return for the analog signals, normally connected to the DIGITAL GROUND.

+5V (Vdd) Power supply input; +5 +/- 0.5V at 250 ma. maximum.

-5V (Vss) Power supply input; -5 +/- 0.5 at 25 ma. maximum.

	Sym	Unit	Min	Nom	Max
DC supply	Vdd	Volts	4.5	5	5.5
	Vss	Volts	-4.5	-5	-5.5
	Idd	mA		200	250
	Iss	mA		-15	-25
Power Dissipation	Pd	Watts			1.5
Operating Temperature	To	C	0		+50
Storage Temperature	Ts	C	-25		+120
Min. High Level Input Voltage	Vih	Volts	2.2		
Max. Low level Input Voltage	Vil	Volts			0.8
Min. High Level Output Voltage	Voh	Volts	2.4	@Iol =	-1 mA
Max. Low Level Output Voltage	Vol	Volts	@Iol =	5 mA	0.45

Figure 4.1-2 Electrical Specifications

	Sym	Unit	Min	Nom	Max
A.C. Impedance	Z	Ohms		600	
Protection	Conforms to FCC part 68 rules				
Transmit Level	Vt	dBm	-12	-10	-9
Carrier Detect Sensitivity	Vcd	dBm	-43/48	-45/50	-48/53
Audio Receiver Gain	Gar	dB		16	
Output Impedance	X	Ohms			100
Audio Transmit	Gat	dB		22	
Input Impedance	Zai	KOhms		20	
DTMF Level	Vdt	dBm	-4.5	-3	-2
Frequency Accuracy	^Fdt	%		+/-0.5	
Transmit Duty Cycle	Tdt	ms		75/75	
Receive Duty Cycle	Rdt	ms		50/50	
Rotary Dialing Speed	Srd	pps		10	
Duty Cycle	^Trd	%		60/40	
Inter Digit Time	Tri	ms		500	

Figure 4.1-3 Phone Line Interface

Name	Units	Min	Max
tAR, tRA, tAW, tWA	ns	0	-
tRR	ns	250	-
tRD	ns	-	200
tWW	ns	250	-
tWD	ns	20	-
tDW	ns	150	-
tDF	ns	10	100
TxTDY/RxRDY Clear	ns	-	400

Figure 4.1-4 Data Bus Timing

4.2 Interface Description

4.2.1 Status

The current status byte is obtained by reading DPort; DØ - D7 pins with: RD = Ø, WR = 1, CS = Ø and C/D = 1. (On the COMM18Ø DPort is address EØ-E1H.) The bits obtained are:

D7	D6	D5	D4	D3	D2	D1	DØ
DSR	SYNDET/ BRKDET	FE	OE	PE	TxEmpty	RxRDY	TxRDY

DSR Data Set Ready This bit is "high" (1) when a connection is present and data is being received. DSR is cleared when RxRDY is set if the component is presenting progress information.

SYNDET/ BRKDET Sync/Break detect In synchronous mode this bit is used to indicate that the sync character(s) have been detected. In asynchronous mode this bit is used to indicate the reception of a break signal on the line. SYNDET/BRKDET signals an incoming call when the MOSART is 'on-hook'. This bit and its interrupt line (SYNDET/ BRKDET) are activated during the ring cycle.

FE Framing Error This bit set in asynchronous mode when no stop bit is found where expected in the bit stream. Another framing error will not be indicated until the line has returned to mark state for at least one bit time.

OE Over-Run This bit is set if the controlling processor does not fetch the received Data-Byte before the next incoming byte is assembled.

PE Parity Error This bit is set when parity check has been enabled and the received parity bit does not match the calculated parity.

TxEmpty Transmitter Empty This bit is set when the last of the transmit data has been completely clocked out of the component. In asynchronous mode the unit will then start marking the line. In synchronous mode the sync character (or character pairs) will be substituted until TxEN is dropped. This bit is reset when new data is supplied.

RxRDY Receiver Ready This bit is set when the MOSART has assembled an incoming Data-Byte or when a new Information-Byte is ready. The bit is reset when the host fetches the data register.

TxRDY Transmitter Ready This bit is set when the component is prepared to accept another command or Data-Byte. This bit reflects the same information as TxRDY line.

4.2.2 Commands

The control byte is set by writing to the DPort (E1H) with: RD = 1, WR = Ø, CS = Ø, and C/D = 1 after the Mode and Sync bytes are loaded.

The individual bits of the control port are:

D7	D6	D5	D4	D3	D2	D1	DØ
EH	IR	RTS	ER	SBRK	RxE	DTR	TxEN

EH Enter Hunt mode In synchronous mode writing a one to this bit causes the component to start searching the incoming bit-stream for the specified sync character(s).

IR Internal Reset Starts the 8251A compatible initialization sequence. This is an alternative to using the Reset line.

RTS Request to Send This bit is used to distinguish transmit data bytes (RTS = 1) from function bytes (RTS = Ø). Changing this bit in the absence of TxRDY may cause a previous Data-Byte to be treated as a command or an incomplete command to be aborted.

ER Error Reset Writing a one to this bit clears the FE, OE and PE status bits.

SBRK Send Break Setting this bit "high" (1) during asynchronous communication will produce a break on the line. The length of a valid break depends on the characteristics of the receiving unit. A minimum break time is equal to two character times.

The break will not start until the transmitter is empty (TxEmpty = 1).

RxE Receiver Enable Allows the component to start assembling bytes from the incoming bit stream .

DTR Data Terminal Ready DTR "high" (1) indicates the host is ready for a connection. If this bit is turned off the physical connection is terminated.

TxEN Transmitter Enable This bit is set to allow the sending of data.

4.2.3 Mode Byte

The MOSART mode of operation is established by sending a Mode Byte(s) to the Control Port following a Reset. Reset may be triggered either by the RESET line or by the IR Control Bit in the Control Register. The first Mode Byte establishes Sync/Async parameters. A hardware Reset must be followed by three Ø Command bytes and a software Reset command to properly initialize the chip.

	D7	D6	D5	D4	D3	D2	D1	DØ
Asynchronous:	S1	SØ	EP	PEN	L1	LØ	B1	BØ
Synchronous:	SCS	ESD	EP	Ø	1	1	Ø	Ø

S1,SØ These two bits select the number of stop bits to be sent in asynchronous mode. The receive side never requires more than one stop bit.

Stop bits:	2	1.5	1	Reserved
S1:	1	1	Ø	Ø
SØ:	1	Ø	1	Ø

EP Even parity is generated/checked if this bit and PEN are set (1). If off (Ø) odd parity is used.

PEN Parity generation/checking is enabled. Valid parity bits are replaced with a zero in the incoming data. On transmission the generated parity bit replaces any bit the host may have put in that position.

L1,LØ The communication character length is set by these bits. This length does not include the parity bit. If PEN is set then total length is one bit more than indicated.

Bits per character:	8	7	Reserved	
L1:	1	1	Ø	Ø
LØ:	1	Ø	1	Ø

B1,BØ These bits set the default speed and select synchronous or asynchronous mode. (The asynchronous speed may be altered later with the speed selector function.)

	B1	BØ
Synchronous (12ØØ bps only)	Ø	Ø
Async. 12ØØ bps	Ø	1
Async. 3ØØ bps	1	Ø
Async. 11Ø bps	1	1

ESD Reserved and should be zero (Ø).

SCS Single character sync is indicated if this bit is a one. If it is a zero, double character sync is enabled. The sync character (or character pair) must be the next byte (or bytes) written to the control register. If asynch mode is selected, this byte is ignored.

4.3 Functional Description

Data I/O is done in the obvious manner by supplying or taking data from the DPort as dictated by the interrupt lines or their corresponding bits in the status register. Failure to take an incoming Data-Byte before the next data byte is available will turn on the OE status bit.

RESET	
C/D = 1	Mode Instruction
C/D = 1	Sync Character 1
C/D = 1	Sync Character 2
C/D = 1	Command Instruction
C/D = Ø	Data / Function
C/D = 1	Command Instruction
C/D = Ø	Data / Function
C/D = 1	Command Instruction

The second sync character is skipped if mode instruction has programmed the MOSART to single character sync mode. Both sync characters are skipped if mode instruction has programmed the MOSART to async mode. The C/D line is controlled by I/O address bit Ø.

Figure 4.3-1 Initialization Sequence

Function bytes and their information bytes are exchanged in the same manner as data. A function byte is distinguished by clearing the RTS control bit before the DPort is written. The function signals completion by raising the TxRDY bit. If RTS is set before a function signals completion, the function is aborted. Information responses are indicated when DSR status bit is zero and the RxRDY status bit is one.

The functions report successful completion simply by raising TxRDY to indicate that the MOSART is ready to accept another function. If an error should occur, RxRDY will be set and DSR cleared at the same time. (If incoming data is present, RxRDY, DSR and the Information-Byte will be delayed until the input is taken.) The TxRDY bit will not be set until the Information-Byte is presented.

None of the errors cause a change in the status of the phone line. For example the Answer Function A responds with an I (Inappropriate) if a modem connection is already established, but the connection is not disturbed.

The correct way to "hang-up" ("on-hook") the phone is to zero the DTR command bit. Zeroing DTR will always "hang-up" the phone. The host may choose to abort a function by setting RTS before it has completed, but the state of the line is not affected. This procedure gives the host complete control of the current on/off-hook status.

Nominal switch-station timings are enforced by delaying certain functions. For example, a dialing-digit function does not complete until its assertion and pause requirements are met.

Many functions cause the phone to go off-hook. A two second pause, called billing delay, is forced when one of these functions actually takes the MOSART off-hook. When the MOSART goes on-hook from an off-hook state, a two second pause is enforced to assure that the call is cleared. Reset causes the MOSART to go on-hook from an unknown state. A pause is forced at reset time to clear a possible previous call.

4.4 Functions

The functions values were designed to be mnemonic aids when interpreted as ASCII characters. Response information referred to here is in the same form and is tabulated below. Invalid function requests will respond with a ? (3F Hex) Information-Byte. Reserved codes and functions requesting unimplemented options return a ! (21 hex) Information-Byte. To simplify host programming the high bit of all function codes is ignored. To ease modification of application programs, function codes NUL and DEL (ØØ, 8Ø, 7F, and FF hex) execute as no-op's, and code ? (3F or BF hex) always returns a ? (invalid) Information-Byte. In the following table the symbol ^ is used to indicate a shift to the ASCII control character set. E.g. ^A symbolizes a hex value of Ø1 (or 81).

4.4.1 Modem Connection Functions

Modem Connection functions attempt to establish a connection with another modem. A successful connection is indicated if no information is returned. If carrier is lost during a modem connection, the MOSART responds by indicating DET and resetting DSR. If carrier returns, DET is reset and the connection continues. The host may terminate the connection by reqluesting a Hold or "hanging-up" the phone.

A Answer Incoming Call This function causes the component to begin the answer side of a modem handshake. Possible responses are F (Failed); T (Timeout); and I (Inappropriate) a connection is in progress.

^A Controlled Answer This answer function performs like the A function but adds extra monitoring information. The answer sequence may be aborted by the caller in two ways. If the caller presses the DTMF "1", a "1" Information-Byte is returned. If the caller speaks, a "v" (76 hex) Information Byte is returned. In either case the answer tone is terminated.

M Monitor Monitor the line and return the status, or originate a handshake if a modem answer tone is heard. Line monitoring can detect the following: R (Ring-back); D (Dial-tone); B (Busy signal); V (Voice) a voice answered; T (Timeout) the timeout alarm has gone off; F (Failed) an answer tone was heard but the originate handshake failed; I (Inappropriate) there already is a connection. The monitor function should be reissued if the host wishes to continue the attempt (e.g., for the first few rings).

O Originate This function causes the MOSART to attempt the origin side of a modem handshake. This is useful when the physical connection is already established. (Monitor (M) would normally be used following dialing.) Responses: F (Failed); T (Timeout) alarm has gone off; I (Inappropriate) there already is a connection.

4.4.2 Configuration Functions

These functions change the MOSART mode of operation without requiring a reset. The Mode-Byte settings can be over-ridden, and default audio setting can be changed.

R Rotary dial the subsequent digits.

T Use tones for subsequent digits. Default mode on reset.

^O One-Ten BPS. Set "preferred speed" to 110 bps. A response of I (Inappropriate) indicates that a 1200 bps connection is in progress.

^T Three-Hundred BPS. Set "preferred speed" of 300 bps. A response of I (Inappropriate) indicates that a 1200 bps connection is in progress.

^H High-Speed BPS. Set "preferred speed" to 1200 bps. Responds I (Inappropriate) if a low speed connection already exists.

The "preferred speed" will be used when the component originates a connection. When answering, the component is forced to follow the convention of the originator. This may cause a temporary shift to or from 1200 bps. When shifting out of 1200 bps mode the component presumes the low speed to be 300 bps. A host which does not know the speed of an incoming call may switch through the lower speeds to determine it. Synchronous communication is only possible at 1200 bps. When initialized for synchronous mode the component will refuse to shift out of 1200 bps.

^V Enable Voice to Phone Line Connection. The synthesizer and audio input are connected to the phone line. This will give the I (Inappropriate) function if a modem connection exists. Subsequent transmit data is routed to the synthesizer until a v (disable voice) function or a reset is executed. ! (Unimplemented) is returned if the synthesizer is not installed.

V Enable Voice Locally. The synthesizer and audio input are connected to the audio output. Subsequent transmit data is routed to the synthesizer until a Disable Voice (v) function or Reset is executed. ! (Unimplemented) is returned if the synthesizer is not installed.

v Disable Voice. Subsequent transmit data is not routed to the synthesizer.

X,x Enable (X) or Disable (x) the audio output connection to the phone line. This allows the user to listen to the line. An X function followed by a ^V function would allow a normal phone conversation through audio-in/out. When the MOSART is generating DTMF tones or sending a modem carrier this connection is blocked.

Z,z Coupler On (Z) or Coupler Off (z). When the coupler is on, the modulator is connected to the audio-out and the demodulator listens to audio-in. (Acoustic couplers are not recommended for 1200 bps operation.)

= No parity will be generated/checked.

> Even parity will be generated/checked.

< Odd parity will be generated/checked.

^S Seven bit character length.

^E Eight bit character length.

4.4.3 Telephone Control Functions

These functions allow the host to dial through the telephone system, and to monitor the dialing process.

D DTMF Receive Mode When placed in this mode the MOSART will recognize incoming DTMF codes as data input. The code values 0-9, a--d, * and # are presented on the data port as their ACSII values when they are signalled by the other end. Note DTMF may be transmitted by issuing the normal dialing digits. This mode is cancelled by a H, M, O or A function.

H Hold This is the inverse of DTMF (D), answer (A), and Originate (O). It performs a logical disconnect of the modems but leaves the line in hold status. This function is used to quiet the line for voice or other use during a connection.

I,i Identify The Identify function returns a version letter for the MOSART device. Version codes will be assigned consecutively starting with "A" (41 hex) and will signify significant unity upgrades. The process also performs an integrity check and sets the high bit of the version letter if the check fails. The i (lower case) function returns a revision code. The version and revision codes are returned as ASCII data bytes, not information bytes.

^M,m Detailed Monitor This function monitors the phone line and reports the frequency heard every Ø.Ø5 seconds. The frequency is reported as a data-byte ranging in vbalue from Ø to 255. These values may be interpreted as:

Value	Meaning
Ø	Quiet
1-254	Frequency in 1Ø's of Hz
255	Frequency exceeding 254Ø Hz

THIS FUNCTION NEVER TERMINATES. The host may cancel this function by setting RTS or hanging up the phone (Dropping DTR or doing a reset). Using this function with a host program to analyze the data, a user may build specific progress monitor routines. With the newer phone systems both internal and external signals for busy, ring-back, and out-of-order, etc., are becoming highly diversified. Many newer systems provide more information through specific tone sequences. A host program which is aware of the destination system conventions can make more intelligent decisions about call progress.

P,p Pause Pause 5 seconds (P) or 2 seconds (p). These functions provide a simple means for the host to delay dialing or protocol sequences.

W Wait for Dial-Tone This function will complete normally if a dial-tone is sensed. Otherwise, an information byte is returned indicating: B (Busy), I (Inappropriate), M (Modem answer tone), R (Ringing), T (Time-out), V (Voice).

Ø-9 Digits Ø-9 The corresponding number to be rotary dialed or touch-toned depending on the current mode.

a-d Letters a-d These letters represent the four tones for the fourth column of a touch tone key pad. They are I (Inappropriate) in rotary dial mode.

*,# Touch-Tone Keys They will respond I (Inappropriate) if the current mode is rotary dial.

4.4.4 Test Functions

These functions provide for testing of the MOSART hardware and the phone line connection.

L Line Analysis This function acts the same as the Monitor function but returns three data bytes if the connection is successful.

These bytes represent:

BYTE	CONTENTS
1	Carrier Frequency Error.
2	Signal to Noise Ratio.
3	Received Carrier Level.

i The lower case i function may be used during a 12ØØ bps connection to check the phase demodulation statics. I (Inappropriate) is returned when there is no 12ØØ bps connection.

These bytes represent:

BYTE	CONTENTS
1	Average Phase Error.
2	# of phase hits since last request.

^X Analog Loop **Originate** This function initiates a local loop-back in the originate band. The filters for transmit and receive are set for the originate frequencies and are looped to each other. Transmitted data is routed through the full analog-digital path before being received back at the receiver port. Possible responses are: none (loop initiated), T (Time-out), **F** (Failed), or I (Inappropriate), if a connection exists.

^Y Analog Loop **Answer** This function performs the same as ^X but uses the Answer band. Note: both **^X** and **^Y** may be used in either low (1[illegible] or 3ØØ bps) or high (12ØØ bps) modes, to test all modes of modulation and demodulation.

FUNCTION SUMMARY

Hex	Char	Notes	Name	Info-Byte	Timeout
Ø1	^A	1,2	Controlled Answer	1 F I T v	17 sec.
Ø5	^E		Eight bit Characters		
Ø8	^H		12ØØ bps	I	
ØD	^M	1	Detailed Monitor	I	
ØF	^O		11Ø bps	I	
13	^S		Seven Bit Characters		
14	^T		3ØØ bps	I	
16	^V	1,3	Voice to Line	! I	
18	^X		Analog Loop Originate Band	F I T	17 sec.
19	^Y		Analog Loop Answer Band	F I T	17 sec.
21	!		Unimplemented	!	
23	#	1,3	Dual Tone # Key	I	
2A	*	1,3	Dual Tone * Key	I	
3Ø-39	Ø-9	1,3	Dual Digits Ø-9		
3C	<		Set Odd Parity		
3D	=		Set No Parity		
3E	>		Set Even Parity		
3F	?		Invalid function	?	
41	A	1,2	Answer	F I T	17 sec.
44	D	1,3	DTMF Receive	I	
48	H	1,3	Hold		
49	I		Identify		
4C	L	1,2	Line Analysis	B D F 1 R T V	17 sec.
4D	M	1,2	Monitor	B D F I R T V	17 sec.
4F	O	1,2	Originate	F I T	17 sec.
5Ø	P		Pause 5 Seconds		
52	R		Rotary Dialing		
54	T		Touch Tone Dialing		
56	V	3	Voice Data Local		
57	W	1	Wait for Dial Tone	B F I R T V M	5 sec.
58	X		Enable Audio Output		
5A	Z		Coupler Mode On		
61	a	1,3	DTMF Tone A	I	
62	b	1,3	DTMF Tone B	I	
63	c	1,3	DTMF Tone C	I	
64	d	1,3	DTMF Tone D	I	
69	i		Identify Revision		
6C	l		Line Statistics (12ØØ bps)	I	
6D	m	1	Detail Monitor (alias for M)	I	
7Ø	p		Pause 2 Seconds		
76	v		Voice Data Off	!	
78	x		Audio Output Off		
7A	z		Coupler Mode Off		

NOTES:
1. These functions will take the phone-line "off-hook".
2. If successful these functions establish a modem connection.
3. These functions break an existing modem connection if any.

All functions may return the "A" Information-Byte if they are aborted during execution.

Figure 4.4-1 MOSART Function Summary

5.0 Sample Procedures for Using the MOSART

This section presents example procedures for typical uses of the MOSART components. It is assumed that the reader is familiar with the notation and concepts presented in the previous section. Function and information bytes are given in their ASCII forms. (^x represents the "control-shift" of the character x.) Numeric values are given in hexadecimal except as noted.

The MOSART component description and many of these application notes assume a working knowledge of the 8251A USART programming techniques. This note follows the conventions introduced in the 8251A component description by using the words "function" and "information" byte. A function is a byte written to the data port when XR (Transmit Ready) is on and RTS (Request to Send) is off. A function may respond to unusual situations by presenting an information byte at the data port with RR (Receive Ready) on and DSR (Data Set Ready) off.

5.1 Initialization

For those not familiar with 8251A USART operation a few words about "mode-setting" are in order. The MOSART, like the 8251A USART, only loads its mode and sync registers immediately following a reset. This convention requires care when the implementation uses the IR (International Reset) bit of the control register rather than the RESET pin of the package. The IR bit always causes a reset when written in the command register. If the history of the unit is not known, the application may actually be writing to the mode or either of the sync registers. A safe way of assuring the state of the MOSART is to write three zeros to the control port (to complete any possible mode) followed by a hex 40 to start the internal-reset. Unlike the 8251A there is no special timing requirement during the initialization sequence. As soon as the reset is recognized the phone is ploaced on-hook. After the mode and control bytes are set, the unit completes its initialization and presents Transmit Ready a short time later.

5.2 Function Examples

5.2.1 Originate

W5558765O	Wait for dial tone Dial 555-8765 (default DTMF) Set Originate mode

This sequence will initiate a call just as any "smart" modem. Success is indicated for each command by setting XR without first presenting an information byte. At the completion of the O the modems are connected and ready to send and receive data.

The W (Wait for dial-tone) might return a V (Voice), R (Ringing), or B (Busy), if someone else were using the line. If it gives an information byte of T (Time-out) the phone cord is probably unplugged.

Originate (O) is the simplest of the MOSART functions used to establish a connection. A more versatile host program might start with:

W5558760M

The Monitor function will return information about the progress of the call. If an R (Ringing) were returned the program might issue more monitor (M) functions and count the rings to decide when to give-up. Busy (B) information bytes might trigger an alternate number or cause an automatic retry at a later time. Voice (V) responses could cause the host to turn on the audio connection allowing a conversation with the person who answered. Dial-tone (D) detection would indicate readiness to dial through a secondary service. Failure to "break" dial-tone would also respond D. This is the likely case of the attached service not being able to accept touch-tone dialing. A flexible controlling program might attempt rotary (R) dialing to check.

The only difference between the W and M functions is the case that does not return an information byte. W expects a dial-tone and will return nothing if one is heard. M is looking for an answer-tone and attempts to complete a connection if one is heard. Both monitor and wait may return information bytes indicating voice (V), busy (B), ringing (R), failure (F), time-out (T) and European answer tone (E).

The two remaining monitor functions, line analysis (L) and detailed monitor (^M or **m**), are explained further in separate sections: "Processing Line Analysis Data" and "Host Extended Monitoring".

5.2.2 Answering

A (or ^A)

The answer (A) function is the complement of the O function. When receiving a call from another modem this is all that is necessary. ^A is a more flexible version of answer. Under this function the answer tone is cancelled when something other than another modem is heard. A v (voice) response from ^A might signal the host program to enable the audio connections. By starting with an answer-tone, incoming calls from other modems can be handled correctly. If the caller does not have a terminal he need only say something appropriate, like "Shut-up", to get the host program's attention, cancel the answer tone, and shift into the voice mode. Conversely, the caller might remain quiet and instead push the "1" button on the telephone (DTMF]) which would allow the host to shift into the DTMF/Synthesizer mode and receive DTMF commands from the caller.

The answer functions are not the only way to "answer" the phone. The unit is taken "off-hook" by voice, hold, monitor, etc. "Answer" is used here to indicate the answer side of a modem connection (transmit high-band, receive low). It would be quite possible to answer by O (originating) if the caller had programmed his unit to dial and then A (answer). E.g.,

W5558765A

"Auto-answer" is implemented by using the ring-detect status. When the MOSART is completely initialized but still "on-hook", the DET status bit (without the DSR bit) is used to indicate an incoming call. The host program could then issue any of the commands that take the phone off-hook. The detect signal is presented as both a status bit and an interrupt pin. The pin could be used to "bring a system to life" when an incoming call occurs. To block this signal simply reset the MOSART and do not complete the mode-set sequence.

5.2.3 Voice Data and Audio In/Out

V (^V)

These two functions enable the voice synthesizer. This portion of the MOSART can reconstruct speech from a highly compressed data format. The component will treat all subsequent data (until a v function) as food for the synthesizer. Although the average data rate is low (175 bytes/sec) the application needs to feed speech bytes smoothly. Very low speed devices or conflicting tasks might disturb the flow on some systems. This can be avoided by buffering the data at "phrase" boundaries.

V directs the voice to the Audio-Out connection. ^V will switch the voice onto the phone line.

X^V

The audio-in ("mike") connection follows the synthesizer routing. Thus after a ^V function both the synthesizer and the audio-in are heard by the other party. If this function had been preceded by an X function a normal phone conversation would now be possible using audio-in/out. Notice that X and Z do not initiate new modes. The next configuration function that is executed will change the setup to honor these requests.

5.2.4 Dual Tone Multi-Frequency (DTMF) Receiver.

D^V

With the voice synthesizer connected to the phone line (^V) and the DTMF receiver enabled (D), the MOSART can be used to communicate without requiring the caller to use a modem. Numerous order-entry and status-inquiry applications become accessible from any touch-tone phone. Both the received DTMF values and the "transmitted" voice data appear to the host as data bytes. This simplifies the implementation of applications which communicate both with and without terminals on the caller's end. In this mode, incoming information is by DTMF from the telephone buttons and outgoing information is by synthesized voice.

5.3 Calls versus Connections

Unlike "smart" modems the MOSART makes very few assumptions about the relationship between a call and a connection. It is often assumed that one call should produce one modem connection, that the caller should be the originator in the modem hand-shake. These assumptions are reasonable for other units mainly because they are only capable of simple modem connections. With the capabilities of the COMM180 it is simple, and often useful to hold the line (maintain the call) while switching in and out of modem connections. For example:

W5551001X^V

is issued by Pat's host as Pat listens on a headset attached to audio-in/out. At the other end Joe answers and exchanges local weather and personal status information with Pat. After which Pat comes to the point: "Sure could use a copy of those figures we worked up last week.". Joe powers on his machine (connected to the same extension and enters:

X^V

He switches to his head-set, hangs up the other phone, and continues: "Looks like I have two sets here. Do you want the ones from before or after the April accounting...". After they agree each invokes a file-transfer program. By agreement one program issues an originate (O) function while the other issues an answer (A). The COMM180s connect and the programs transfer the required information. During this operation the head-sets are automatically disabled. Upon completion both programs issue the function sequence:

H^V

The H (hold) function disconnects the modems but does not hang up the phone. ^V restores the head-set connection for the two speakers. (X need not be repeated as it is over-ridden, not cancelled, by the connection.) At this point Joe and Pat can use the same phone call to discuss the data just transferred while examining it with application programs on the now idle machines. The procedure can be repeated as needed without re-dialing.

This example demonstrates a simple manner of utilizing multiple connections which requires almost no special programming. Much more interesting possibilities exist. Consider a specially designed data exchange program that automates these sequences. It would be quite simple to enter the "conversation" mode by issuing:

HD^V

thus allowing the two programs to use DTMF signaling during the audio connection. By checking for a particular tone sequence the applications could switch to data transfer mode on request from either operator. This technique can be extended even further by combining the synthesizer and DTMF functions to produce applications that change between audio response and data exchange as appropriate.

5.4 Speeds, Character Lengths, and Parity

To maintain program compatibility with the 8251A the COMM180 provides initialization data bits that set the length, parity and "preferred" speed for asynchronous data. If this were the only means of setting these attributes, the host would be forced to reset the MOSART in order to change any of these.

Functions to change these attributes are provided. Seven (^S) and eight (^E) bit character lengths may be set at any time. The parity may also be set (< Odd, = None, and > Even) regardless of the current MOSART status. These changes will take affect with the next datq byte transmitted or received. Speed change functions are also provided (^O 110, ^T 300 and ^H 1200).

Modem signalling and hand-shake protocols force some restrictions on the use of these speed functions. One restriction is necessitated by the different forms of signalling used during a modem connection.

Low speed data (up to 300 bps) is transferred using frequency shift keying (FSK) while 1200 bps transfers utilize phase modulation. Although the 212A handshake protocol allows a choice of methods, once the connection is established there is no provision for changing. Thus switching to or from 1200 bps is prohibited during a "modem" connection.

The choice allowed by this hand-shake is under control of the originating modem. Thus if a MOSART on the answer side of a handshake does not agree with the originator about the type of modulation to use it must follow the originator to complete the connection. The means an answering unit may shift temporarily to 1200 bps. This is done automatically by the MOSART answer functions and is usually of no concern to the host program.

6.0 Processing Line Analysis Data

This section discusses the two line analysis functions provided by the COMM180's MOSART. Formulas for calculating frequency, phase angle, and decibel figures from the raw data values are provided. For implementations without intrinsic math functions reasonable approximation sequences are provided. A brief discussion of expected error rates and suggested references aid in interpreting these values.

The MOSART component description and many of these application notes assume a working knowledge of 8251A USART programming techniques. This note follows the conventions introduced in the component description by using the words "function" and "information" byte. A function is a byte written to the data port when XR (Transmit Ready) is on and RTS (Request to Send) is off. A function may respond to unusual situations by presenting an information byte at the data port with RR (Receive Ready) on and DSR (Data Set Ready) off.

6.1 L (The Line Analysis Function)

When the COMM180 MOSART is requested to perform the line analysis function L, the following sequence ensues. The MOSART performs full monitor (M) functions. This portion of the function may complete by returning an information byte for the following conditions: (B) Busy signal, (D) Dial-tone, (R) Ring-back signal, (V) Voice, (T) Time-out, (more than 17 seconds of quiet), or (F) Failed, answer-tone disappeared. In the case of (R) Ring-back, the host program would probably reissue the L function to allow for some reasonable number of rings before giving-up.

If an answer-tone is received, none of these information bytes appear. The unit then analyzes the quality of the connection and presents three data bytes to the host. The L function then completes by raising transmitter ready (XR). The data bytes contain values representing:

Byte	Measurement
1	Frequency Error
2	Signal/Noise Ratio
3	Signal Level

6.1.1 Frequency Error

The first data byte indicated the received frequency error. This is a signed value indicating the discrepancy between the expected 2225 Hz answer tone and the actual tone received. On systems which do not have signed eight bit quantities the reading (E) must be sign extended:

$$S = E \quad (\text{if } E < 128)$$
$$S = E - 256 \quad (\text{if } E >= 128)$$

Given the signed quantity S, the deviation from nominal is given by the equation:

$$D = 7372800/(3313.618-S)-2225 \quad \{Hz\}$$

This deviation may also be approximated by:

$$D = 0.68*S \quad \{Hz\}$$

Frequency errors can be introduced both by the originating modem and by switched networks utilizing frequency division multiplexing.

6.1.2 Signal/Noise Ratio

The second data byte represents the signal to noise ratio as measured at the receiver input, using phase jitter evaluation of the received answer tone. It is an integer (L) ranging from 0 to 256. For L greater than 2 aned less than 255 the signal to noise ratio can be calculated as:

$$R = 20*LOG(421/L) \quad \{dB\}$$

For L less than 3, the ratio is greater than 45 dB, representing a very reliable communication connection.

For L equal to 255, the ratio is less than 5 dB, indicating very little chance of useful communication.

6.1.3 Bit Error Rate Evaluation

The signal to noise ratio can be used to evaluate the bit error rate (the probability of an error in the received data due to additive noise). Approximate values of error probability for several signal to noise ratios are given below. For further considerations see **Data Transmission** by W. Bennett and J.Davey, McGraw Hill, 1965, or a similar reference.

Signal/Noise Ratio for Gaussian Noise at 3KHz Bandwidth	Error Rate Order for 300 BPS FSK	Error Rate Order for 1200 BPS PSK
6	10^{-4}	10^{-3}
8	10^{-6}	10^{-5}
10	10^{-8}	10^{-7}

6.1.4 Signal Level

The third data byte reflects the signal level of the answer tone measured after the receiving filter. It is an integer value ranging from 0 to 256. This value (N) is in inverse proportion to the received voltage. The following approximation is valid for values of N greater than 2 and less than 256.

```
VR = 0.483/(N*(1-5.4E-6*N**2))      {Volts RMS}
```

In communication technology, signal levels usually are presented in dBm. For a 600 ohm telephone line the signal level (Z) can be calculated from N using the following approximation:

```
Z = -4.1+N**2/19391-20*LOG(N)       {dBm}
```

This calculation is only meaningful for 2 < N < 256. For N < 3 the level is higher than -10 dBm. This is practically impossible for domestic telephone networks. For N equal to 255 the level is less than -49 dBm. This would indicate a highly unreliable data communication connection. The accuracy of these results depends on the value of N. For greater than 10, the error is within +/-3 dB.

6.2 Approximating 20*LOG(N)

Applications which do not have immediate access to a log base 10 function may find it difficult to implement the above conversions. For values of N between 1 and 255 the following formula gives a reasonable approximation for the decibel conversion function 20*LOG(N):

```
20*LOG(N) = 6*P+23*(M**3-1)/(M+1)**3
```

where P is the largest value such that 2**P is less than or equal to N, and M is N/2**P. The following three-step algorithm sets R equal to the approximation of 20*LOG(N):

```
L1  INITIALIZE:   K=0, T=N

L2  NORMALIZE:    If T < 2 then L3; else K=K+6, T=T/2  repeat L2.

L3  CALCULATE:    X=T+1, T=T*T*T, X=X*X*X, R-K+23*(T-1)/X.
```

These are not integer calculations. Both T and X in algorithm L must be scaled to avoid losing significance.

6.3 l (1200 bps line statistics)

The COMM180 MOSART can provide extra statistics about an ongoing 1200 bps connection. Unlike the line analysis data, these statistics are available from both the originating and the answering units. The l (lower-case L) function returns an I (Inappropriate) information byte if there is not currently a 1200 bps connection. If there is such a connection two data bytes are returned:

Byte	Measurement
1	Average Phase Error
2	Number of Phase Hits since previous measurement

Because these bytes are returned as data, it is important that the host application avoid confusing true received data with the results of the l function. The application can block incoming data by turning off (zeroing) the receiver enable control bit in the control register and then emptying the data port, if necessary, before issuing the l function. Note that data received while this bit is off will be discarded. No over-run error will occur.

6.3.1 Average Phase Error

The phase error is given in units of 45/1289 of a degree. Thus if b is the binary value of the data byte:

$$\text{Average phase Error} = (45*b/128) \text{ Degrees}$$

This value is updated continually and represents the phase errors observed during the last 500 milliseconds.

For reasonably good channels (with signal/noise ratio > 15dB) the main cause of measured phase error is intersymbol interference. In this case, peak phase fluctuation in the center of the baud interval is approximately equal to the value of the data byte. For reasonably good channels, the value is below 20 (decimal) which corresponds to an average phase error of less than 7 degrees.

6.3.2 Number of Phase Hits

This value is a count of the number of times the Statistical Equalization Algorithm discovered different phase readings during a single baud time. Because of the SEA's recovery techniques it is unlikely that these errors have resulted in actual data errors. This number should be considered another measure of the quality of the connection, not a global error counter.

For purposes of judging reliability, only values expressed in Hits/Second or % Bauds Hit are meaningful. To this end, the l functions clear the counter after each reading. To avoid losing information, the counter locks at a count of 256. Thus a value of 255 should be treated as meaning "more than 254 hits". If two l functions issued at T1 and T2 return hit counts of H1 and H2 respectively:

$$\text{Hits/Second} = H2/(T2-T1)$$

$$\% \text{ Bauds Hit} = H2/(6*(T2-T1))$$

6.4 Check Values

As well as presenting these values for the users edification, the host application might make decisions about the connection based on these figures. Various cut-off points could trigger a "redial", the logging of unexpected sun spot activity or a polite suggestion that some other long distance service provider be investigated. These decisions can be based on the "raw" values by using the following table:

Data Byte	Frequency Deviation (Hz)	Signal to Noise Ratio (dB)	Signal Level (Volts)	(dBm)
0	.00	----	-----	----
2	1.34	46.5	0.242	-10.1
4	2.69	40.4	0.121	-16.1
8	6.38	34.4	0.060	-22.2
12	8.09	30.9	0.040	-26.7
16	10.80	28.4	0.030	-28.2
20	13.51	26.5	0.024	-30.1
24	16.23	24.9	0.020	-31.7
28	18.96	23.5	0.017	-33.0
32	21.70	22.4	0.015	-34.2
36	24.44	21.4	0.014	-35.2
40	27.19	20.4	0.012	-36.1
44	29.94	19.6	0.011	-36.9
48	32.70	18.9	0.010	-37.6
52	36.47	18.2	0.009	-38.3
56	38.25	17.5	0.009	-38.9
60	41.03	16.9	0.008	-39.5
64	43.82	16.4	0.008	-40.0
72	49.42	16.3	0.007	-41.0
80	55.05	14.4	0.006	-41.8
88	60.70	13.6	0.006	-42.6
96	66.38	12.8	0.005	-43.3
104	72.10	12.1	0.005	-43.9
112	77.84	11.5	0.005	-44.4
120	83.60	10.9	0.004	-44.9
128	-82.75	10.3	0.004	-45.4
136	-77.76	9.8	0.004	-45.8
144	-72.75	9.3	0.004	-46.2
152	-67.71	8.8	0.004	-46.5
160	-62.65	8.4	0.004	-46.9
168	-57.56	8.0	0.003	-47.2
176	-52.45	7.6	0.003	-47.4
184	-47.32	7.2	0.003	-47.7
192	-42.16	6.8	0.003	-47.9
200	-36.98	6.5	0.003	-48.1
208	-31.77	6.1	0.003	-48.2
212	-29.16	6.0	0.003	-48.3
216	-26.54	5.8	0.003	-48.4
220	-23.91	5.6	0.003	-48.5
224	-21.28	5.5	0.003	-48.5
228	-18.64	5.3	0.003	-48.6
236	-13.35	5.0	0.003	-48.7
240	-10.69	4.9	0.003	-48.7
244	-8.03	4.7	0.003	-48.8
248	-5.36	4.6	0.003	-48.8
252	-2.68	4.5	0.003	-48.9
254	-1.34	4.4	0.003	-48.9

Figure 6.6-1 Selected Data Values

7.0 Host Extended Monitoring

This section discusses the extended line monitoring capabilities of the COMM180 MOSART. The detailed-monitor function and data are described along with the basic concepts of line monitoring. An algorithm and parametrics for processing "standard" progress tone sequences are discussed. Suggestions on how these concepts can be extended in a host application are offered.

7.1 The Need

The COMM180 MOSART provides functions for monitoring the progress of a phone call through a switched network. These functions attempt to decode the tone sequences normally used by networks to indicate status to the caller. The tones used by most systems in the United States are familiar to all of us. Indeed, many of these signals adhere to published standards. (See below.) Newly offered networks and added features on existing ones are quickly, and not always consistantly, extending these standards.

Applications using these networks can make valuable use of these extended signals to evaluate their progress and choose alternative actions. Yet, no static implementation can keep pace with changing standards. The MOSART addresses this problem by making the basic information available to its host application.

7.2 m (Detailed Monitor)

The **m** function of the COMM180 MOSART provides the host with a continuous stream of data bytes. Once initiated this function runs until the MOSART is reset or the function is aborted. (Any function may be abortyed by raising, (setting to 1), the RTS bit in the control register). The data bytes are presented at regular intervals of 0.05 seconds. Each byte has a binary value equal to one tenth of number of Hertz measured on the line during the last 50 milliseconds.

Data	Meaning
0	Quiet
14-254	140 - 2540 Hz
255	Greater Than 2540 Hz

The value returned is an "average" frequency for the period. For pure tones they represent a fairly accuarate measure of the signal. Most of the signalling tones, however, are actually two or more frequencies of unequal amplitude. "Averaging" over 50 milliseconds produces consistant results for all but the most extreme of these mixtures.

7.3 Using Detailed Monitor with other Functions

Unlike the monitor (M) function, detailed monitor does not automatically connect if a modem answer-tone is received. The host progress monitoring can complete the connection by aborting the monitor function and issuing an O (Originate) function request. This is the normal case when a 2225 Hz. answer-tone is recognized. The timing here is not critical, as most modems will not time-out for several seconds. (The normal timeout is about 15 seconds. A few modems wait as little as 5 seconds).

If the host application desires line analysis following detailed monitor, a L (Line-Analysis) function may be issued after aborting monitor. Although L starts by performing a full monitor function, that portion will "fall-through" upon hearing answer-tone.

If the answering modem is another MOSART, the caller can shift to other modes by speaking into the telephone or "dialing" a DTMF "1" when the answering handshake tone is heard. The answering MOSART would then respond to its host with a v or 1 information byte, indicating voice or DTMF respectively.

7.4 Preprocessing

The preprocessing algorithms in the MOSART take considerable care in filtering out the noise and static normally present during network switching. Host processing does not need to provide for the rejection of short term noise. Weak signals are also rejected by the preprocessing. Thus, the values returned are non-zero only for signals of significant energy and duration.

A second important preprocessing function is provided by a sliding temporal filter. This filter bridges missing sections in tone sequences. It also provides leading and trailing edge conditioning. The conditioned signal shows no variation in frequency, even for periods which were only partially filled by the tone. This conditioning is symmetrical. Thus, the conditioned duration matches the actual duration.

7.5 Tones, Periods and Voices

Because of the preprocessing, tone checking becomes a simple matter. A tone produces a series of measurements that are consistant within some delta. For pure tones this delta is very small. The complex tones used to simulate ringing, etc., are observed to vary by as much as +/- 80 Hz.

The periodicity of a tone sequence can be measured by simply counting the number of valued and zero (silent) measurements in the sequence. Care should be taken with the first cycle of the sequence. The tone generators may connect at any point in their cycle. The first tone might not last for a full period. The first silence cannot be destinguished from network switching delay.

The conditioned measurements also make the destinction between voice and tone a simple matter. Tones, after conditioning, will not vary by more than +/- 80 Hz. Thus, a series of measurements that varies by more is probably a voice. While this is the usual distinction, very short utterances may not show any variation in frequency. The timing of these bursts, however, is almost never repeatable.

7.6 Example Information Signals

The following information has been extracted from: Bell System Switched Network Capability and Performance Specifications, Technical Reference PUB 61100, Description of the Analog Voiceband Interface Between the Bell System Local Exchange Lines and Terminal Equipment, January 1983. This list is by no means exhaustive. Indeed the same publication warns:

> "Some existing network systems use non-precise tones with little specific information documented or known about their frequency content or levels, even at the point of application to the transmission path."

Tones (Hz)	On/Off (Sec)	Meaning
350+440	Continuous	Dial Tone
480+620	0.5/0.5	Busy
480+620	0.25/0.25	Reorder (equip. unavailable)
440+480	2.0/4.0	Ringing
480+620	0.5/0.5/0.5/1.5	Vacant (Number unassigned)
1400+2060+ 2450+2600	0.1/0.1	Receiver off hook

The recorded messages used by Bell often begin with standard tone sequences:

Tone 1 Hz/ms	Tone 2 Hz/ms	Tone 3 Hz/ms	Type of Recording
985/380	1429/380	1777/380	No Circuit Available
914/274	1370/274	1777/380	Intercepted
985/380	1370/274	1777/380	Vacant
914/274	1429/380	1777/380	Reorder

7.7 An Example Algorithm

The example given here performs in much the same way as the monitor function of the MOSART. Reorder, busy and off-hook all return a busy indication. No attempt is made to process the recording tone sequences. The busy response is delayed long enough to allow for the atypical ring sequence short-short-long. The following criteria are used to distinguish the tones.

Result	Criteria
T timeout	Silent for more than 20 seconds.
M modem answer	Value between 220 and 225
D dial tone	More than 2.25 secondes of constant tone.
R ring-back	Consistant tone longer than 1.25 sec., or tone > 1.25 and quiet > 3.5 sec. twice.
V voice	Longer than 2.25 sec. and unstable freq., or quiet longer than 4.2 sec. after tone, or length of quiet increased more than .15 sec.
B busy	More than 6 quiet/tone cycles without R or V.

In the Example Algorithm, (shown on the next page), FR is a function which returns the next frequency reading presented by the MOSART. This function implies waiting for the RR (Receiver Ready) status bit and fetching the data port while the MOSART is performing the a function.

This algorithm is intentionally conservative. Experimentation has shown that there exist a few very "noisy" ring and busy signal generators. In older systems these generators appear to start and stop the separate oscillators at slightly different times. This produces frequency shifts at the beginning and end of tones. The approach taken here avoids false indications by using the frequency information only if it is stable. Much faster algorithms are possible when the characteristics of the called system are known.

```
Q := Ø; R := Ø;
repeat                            (--- Initial Silence ___           )
   F := FR;                       ( Get new reading.                 )
   if Q = 17 * 2Ø                 ( Quiet more than 17 sec.,         )
      then goto XT;               ( timeout.                         )
   Q := Q + 1;                    ( Count empty periods.             )
 until F <> O;                    ( Watch for Sound.                 )
                                  ( --- End initial silence ---      )
C := Ø; S := 84;
repeat                            (--- Cycle Loop ---                )
   T := Ø;                        ( Time the Tone.                   )
   B := F; G := Ø;                ( Base frequency and Gap.          )
   repeat                         (--- Tone Loop ---                 )
      if F < B                    ( If new base frequency.           )
         then begin
         G := G + B - F;          ( Gap enlarged.                    )
         B := F;                  ( Set new Base.                    )
        end
         else G := MAX(G,F-B);    ( Save largest gap.                )
      if B < 225 and B > 22Ø      ( Is it an answer tone.            )
         then goto XM;            ( Yes, Indicate Modem              )
      T := T + 1;                 ( Count times.                     )
      F := FR;                    ( Get new reading.                 )
     until T > 44 or F = Ø;       ( Time > 2.25 sec. or Silence.     )
                                  (--- End of Tone ---               )
   if F <> Ø                      ( If timed out,                    )
      then if G > 16              ( If gap is more than 16Ø Hz.,     )
         then goto XV;            ( It was voice.                    )
         else goto XD;            ( Long and steady = Dialtone       )
   if T > 25                      ( If tone longer than 1.1 sec.,    )
      then if G < 16              ( If steady enough,                )
         then goto XR;            ( Then it's a ring.                )
   C := C + 1;                    ( Count cycles.                    )
   if C > 6                       ( More than 6 cycles,              )
      then goto XB;               ( Must be busy.                    )
   Q := Ø;
   repeat                         (--- Quiet Loop ---                )
      Q := Q + 1;                 ( Time the silence.                )
      F := FR;                    ( Get a new reading.               )
    until Q > S or F <> Ø;        ( Until noise or timeout.          )
                                  (--- End of Quiet ---              )
   if F = Ø                       ( If it was a timeout,             )
      then goto XV;               ( it must have been a voice.       )
   S := MIN(S,Q + 3);             ( Next may only be .15 longer.     )
   if T > 25 and Q > 7Ø           ( Tone > 1.8 and quiet > 3.5       )
      then R := R + 1;            ( Vote one more for ring.          )
      else R := Ø;                ( Not a Ring.                      )
   if R > 1                       ( Two votes,                       )
      then goto XR;               ( Probably a noisy ring.           )
   until false;                   (--- Repeat forever ---            )
 XR: ...                          ( Result is Ring.                  )
 XV: ...                          (           Voice.                 )
 XD: ...                          (           DialTone.              )
 XL: ...                          (           Lost Call.             )
 XB: ...                          (           Busy.                  )
```

Figure 7.7-1 Example Algorithm

7.8 Other Potential Uses

The low-level monitor function (**m**) is in general a tool which can be used to measure base frequency of any significant signal present on either the phone-line or the audio-in connection. Analysis of the signal on the audio-in is started by issuing a Z (accoustic coupler on) function followed by an **m** function. Whatever the source, the data gathered by this function has potential beyond simple progress monitoring.

The following ideas are definitely preliminary, but serve as examples of possible useful implementations of the **m** monitor function.

Adaptive Monitoring The host system may have more information about the tone sequences to be expected when calling a particular number. In modern switched networks sequences for "out-of-service", "gone-for-the-day", "camped-on", etc. are becoming common place. A host which is provided with the possible sequences for individual exchanges and networks could use different algorithms for these cases.

Multi-Drop-Devices Either monitor (**m**) or DTMF receive (D) may be used to select among multiple MOSARTs on one side of a connection. This technique involves having all MOSARTs on a single line, answer, at one time. (At the end of the n'th ring.) None of the answering MOSARTs responds until a particular tone (**m**) or DTMF sequence (D) is recognized. The unit which recognizes the sequence responds by establishing a connection (using the **L,M,O, A,D** or **DTMF** digit, function). That unit can exchange information with the caller and later disconnect, logically but not physically, by issuing a hold (H) function. This would allow logical connections with multiple unrelated devices during a single "physical" (phone call) connection.

Audio-Data **Time-Multiplexed** A very viable, "user-friendly", technique of information-transfer is possible between two MOSARTs. Care should be taken, in implementation of low level monitoring algorithms, not to block this function.

> Two MOSARTs may connect the users verbally, (audio-in/out via "X^V") while simultaniously checking for potential data transfers with the monitor (M), DTMF-receive (D) or controlled-monitor (m,^M) finctions.
>
> The simplest technique is to use the high-level-monitor (M) function repeatedly on both sides. The side that first issues an answer function (A versus another M) will complete the connection. The result is that both MOSARTs establish a data connection. (The "M" side completes without an information byte. The "A" side also returns normal completion stats. The logical connection is established.)
>
> If a host monitor program is used, presumptions about when to hang-up etc. should be reviewed carefully.

DTMF Column Because DTMF is a dual-tone the full information content of a DTMF key cannot be recovered from the **m** data. Of course, once DTMF is agreed on as the mode of transmission the MOSART DTMF-Receive function (D) can decode each individual key. Due to the level mix of DTMF frequencies it is possible to determine the column number of an incoming DTMF tone from the m values. This distinction may be used at the initiation of a transaction to determine the abilities of the transmitter.

Whistled-Passwords Many applications start by expecting a data or DTMF sequence to check authority. For applications dealing mainly with voice this can be a restriction. (E.g., a message recorder called from a rotary-dial phone.) The **m** data function can be used to look for a sequence of whistled tones. (Try this with a real phone, the demo-program and a favorite melody.) The consistency is profound even from day to day. (Not counting the exceptionally tone-deaf.)

General Half-Duplex FM Given an arbitrary tone generator for a transmitter, the MOSART can perform as a general purpose FM demodulator. The maximum baud rate and number of levels per baud are beyond the scope of this discussion. A conservative choice might be 64 levels at 5 baud. (40ms tone/40ms silent). While the resulting bit rate is only 50 bits per second, the transmitter could be constructed extremely cheaply. This technique has application both in simple data collection (from the phone line) and in cheap data recording (through audio-in).

8.Ø Audio Input/Output Usage

This application note discusses the proper use of the Audio Input and Output Interface for both voice and data communication using an acoustic coupler or external DAA.

The Audio Input and Output can be used for voice communication over the telephone line using an external headset or similar device, and for data communication in the acoustical coupling mode or "four wire" configuration. The Audio Output may also be used as a synthesized speech output. Each of these modes is described below.

Parameter	Explanation	Nominal Value	Units	Note
Audio transmit gain	from Audio input to the telephone line at 6ØØ ohms termination	22	dB	Frequency 1KHz
Audio receiver gain	from telephone line to audio output	16	dB	Frequency 1KHz
Input impedance	Audio input	> 2Ø	K Ohms	
Output impedance	Audio output	< 1ØØ	Ohms	
Maximum output voltage	Audio output	2.5	Vpeak	Load >= 3ØØ Ohms
Sensitivity in Data Mode	Audio input, on/off	-59/-63	dBm	
Data transmit level	Audio output	-2.5	dBm	Load >= 3ØØ Ohms
Synthesized voice amplitude	Audio output Telephone line	1.7 Ø.7	Vpeak Vpeak	Peak Voltage

Figure 8.Ø-2 Electrical parameters of audio input and output

8.1 Voice Communication

A block diagram of this configuration is shown in Figure 8.1-1. The Audio Input and Output are under host computer control and may be enabled by the following commands:

^V Connects Audio Input and Output (microphone and earphone) to Telephone Line

X Connects Audio Output only (earphone) to Telephone Line upon esecution of subsequent command.

v Disconnects Audio Input from Telephone Line

x Disconnects Audio Output from Telephone Line upon execution of subsequent command.

These commands must be issued by the host computer. Electrical parameters for this configuration are shown in Figure 8.0-1.

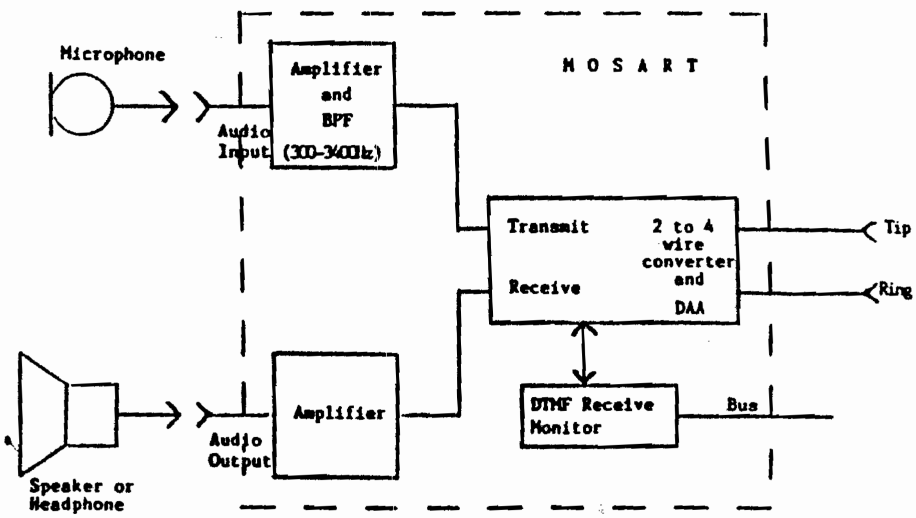


Figure 8.1-1 Voice Communication configuration after X and ^V functions

8.2 Data Communication in Acoustical Coupling Mode

The block diagram of this configuration is shown in Figure 8.2-1. Again, the Audio Input and Output are under host computer software control and may be activated by the following commands:

Z Connect MODEM to Audio Input and Output
z Disconnect MODEM from Audio Input and Output

The electrical parameters for this configuration are shown in Figure 8.0-1.

Additionally, instead of using acoustical coupling this configuration may also be used for 4-wire communication. If two COMM180 modems are used in this configuration, signal attenuation between the Audio Input of one and the Audio Output of the other should be in the range of 25-55 dB.

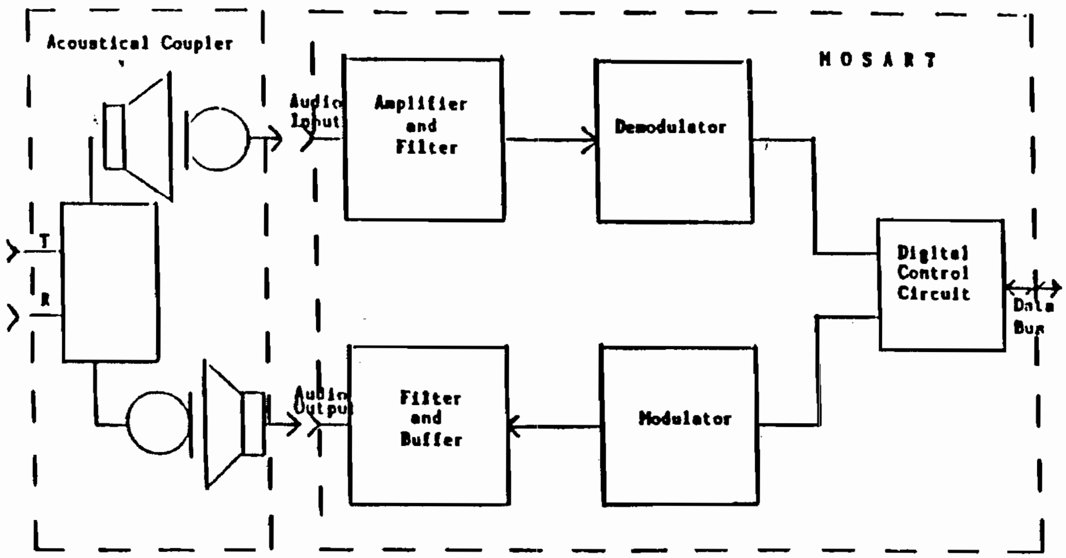


Figure 8.2-1 Data Communication Acoustical Coupling and 4-Wire Mode after Z function

8.3 Synthesized Speech Output

In the COMM18Ø, the voice synthesizer output may be connected to either the Audio Output or the Telephone Line. The commands to enable this function are as follows:

^V Connect Voice Synthesizer to Telephone Line and Audio Output

V Connect Voice Synthesizer to Audio Output only

v Disconnect Voice Synthesizer

Under both the ^V and V commands, all subsequent transmit data is routed to the synthesizer until the v function or reset is executed.

9.0 Word List for the COMM180 Speech Synthesizer

The following words comprise the built-in vocabulary of the COMM180's speech synthesizer.

(Letters of the alphabet)
a b c d e f g h i j k l m n o p q r s t u v w x y z

(These suffixes may be attached to another word as an ending)
_ing _th _ed _s

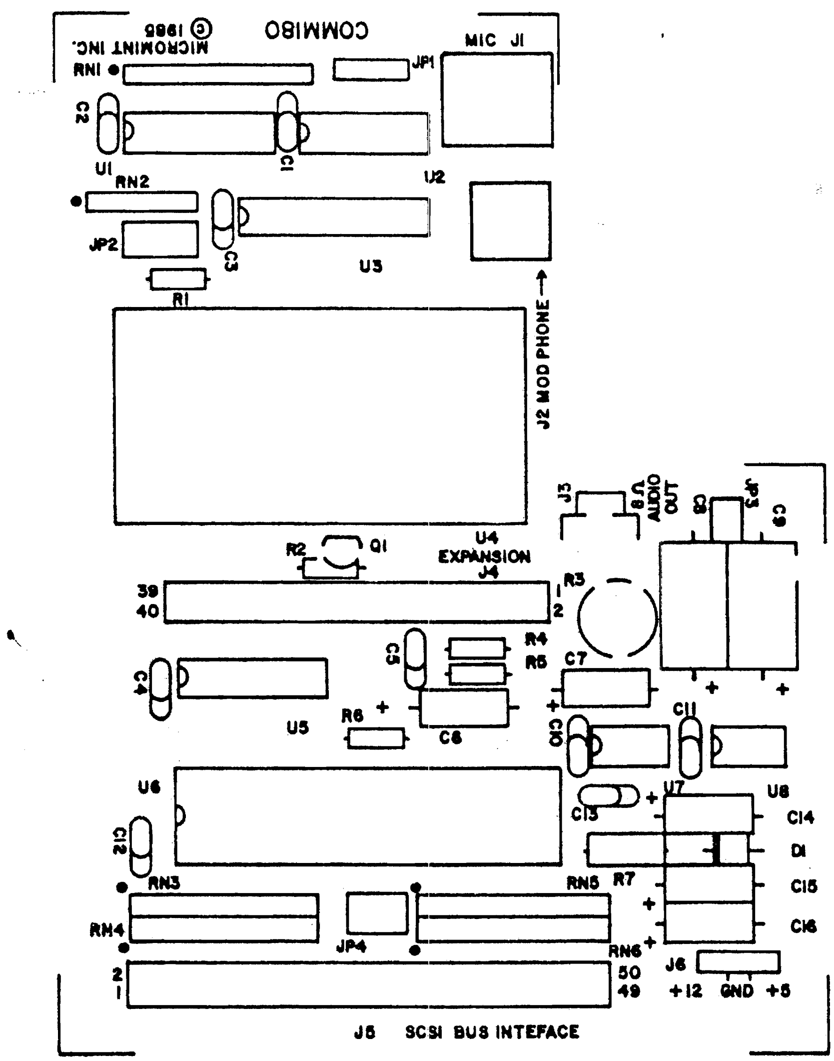
(These words simply cause a pause)
pause40, pause 80, pause100, pause200

tone (makes a high pitched tone)

a.m.
after
after_the_tone
again
and
area_code
at
at_this_number
before
business_hours_are
connected
emergency
error
from
function
goodbye
hello
identification
I'm_sorry
is
later
medical
Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
number
off
oh
on
please_call
please_enter
please_wait
p.m.
police
port
press_the_pound_key
press_the_star_key
status
switch

terminated
thank_you
thank_you_for_calling
the_time_is
this_is_an_automatic_message
thru
to
to_change_your_entry
to_exit
warning
with
you_have_dialed
your
your_call_cannot_be_answered_at_this_time
your_callback_number
your_party
zero, one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty, thirty, forty, fifty, sixty, seventy, eighty, ninety, hundred, thousand, million

10.0 Silkscreen



10.1 Parts List

CAPACITORS

C1-C5	0.1 MFD, MONOLITHIC, 50V, .1 CTR
C6,C7	10 MFD, ELECTROLYTIC, 25V, AXIAL
C8,C9	100 MFD, ELECTROLYTIC, 25V, AXIAL
C10-C12	0.1 MFD, MONOLITHIC, 50V, .1 CTR
C13	.02 MFD, MOLOLITHIC, 50V, .2 CTR
C14-C16	10 MFD, ELECTROLYTIC, 25V, AXIAL

CONNECTORS

J1	MINI PHONE JACK, 3 PIN
J2	MODULAR PHONE JACK, 4 PIN
J3	RCA, AUDIO, RT ANGLE
J4	2x20 HEADER RECEPTACLE, .1 CTR
J5	2x25 PIN HEADER, SHROUDED, .1 CTR
J6	1x4 PIN HEADER, RT ANGLE
JP1	1x4 PIN HEADER, .1 CTR
JP2	2x4 PIN HEADER, .1 CTR
JP3	NO CONNECTOR (WIRE POINT)
JP4	NO CONNECTOR (WIRE JUMPER)

DIODES

D1	1N4735A, 6.8V ZENER

I.C.'S

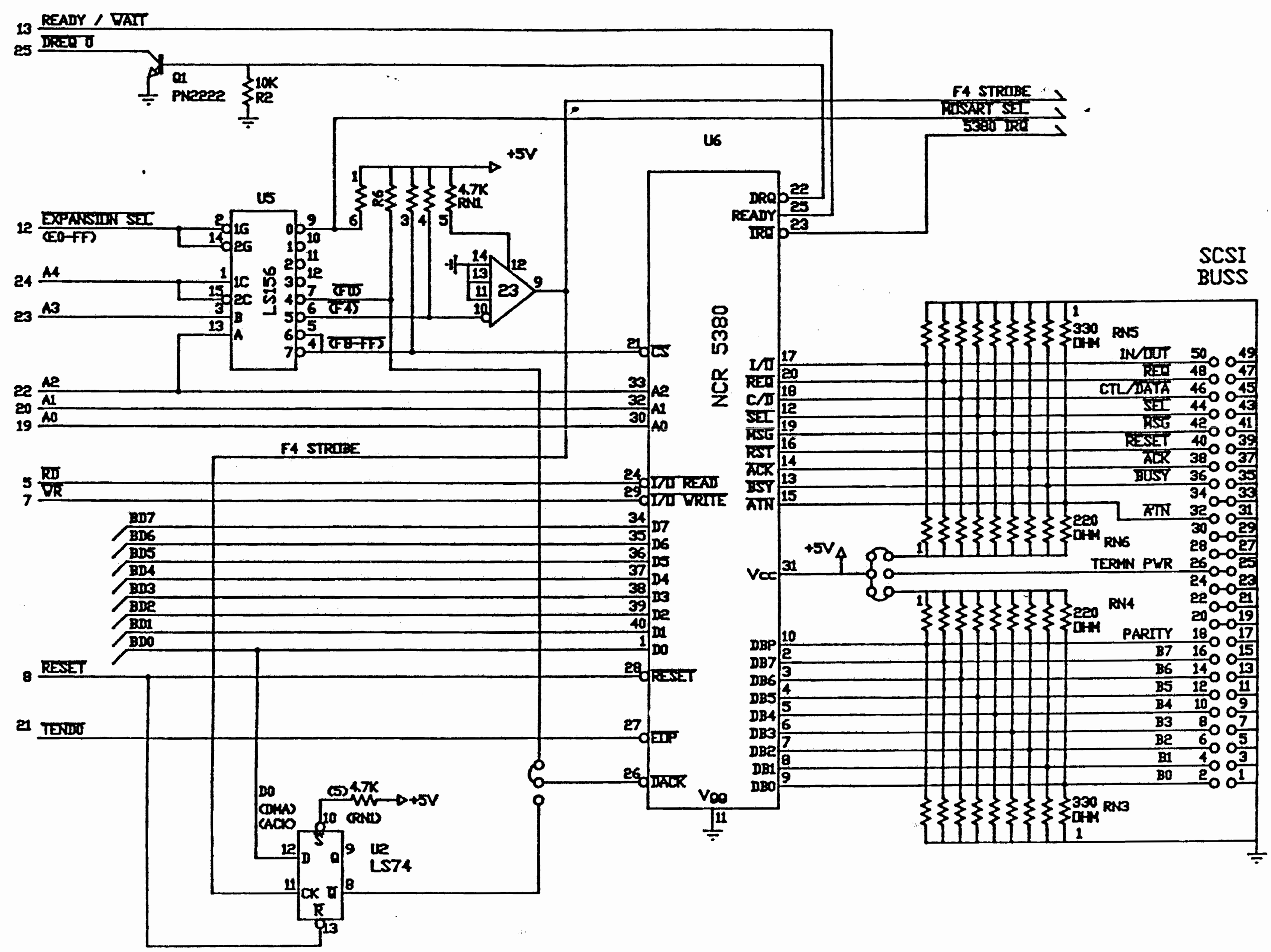
U1	7423, DUAL OR, EXPANDABLE	
U2	74LS74, DUAL D LATCH	
U3	74LS245	
U4	XECOM 1203 MOSART (OR XE1201)	- OPTION 1
U5	74LS156, 1 OF 8 DECODER	
U6	NCR 5380 SCSI CONTROLLER	- OPTION 2
U7	LM386, AUDIO POWER AMP	- OPTION 1
U8	ICL7660, DC-DC INVERTER	- OPTION 1

RESISTORS

R1	560 OHM, 1/4W, 5%
R2	10 K, 1/4W, 5%
R3	10 K, POT
R4	10 OHM, 1/4W, 5%
R5	100 K, 1/4W, 5%
R6	4.7 K, 1/4W, 5%
R7	150 OHM, 1/2W, 5%
R1	560 OHM, 1/4W, 5%
RN1	4.7 K, 9 ELEMENT(10 PIN), SIP
RN2	2.2 K, 5 ELEMENT(6 PIN), SIP
RN3	330 OHM, 9 ELEMENT(10 PIN), SIP
RN4	220 OHM, 9 ELEMENT(10 PIN), SIP
RN5	330 OHM, 9 ELEMENT(10 PIN), SIP
RN6	220 OHM, 9 ELEMENT(10 PIN), SIP

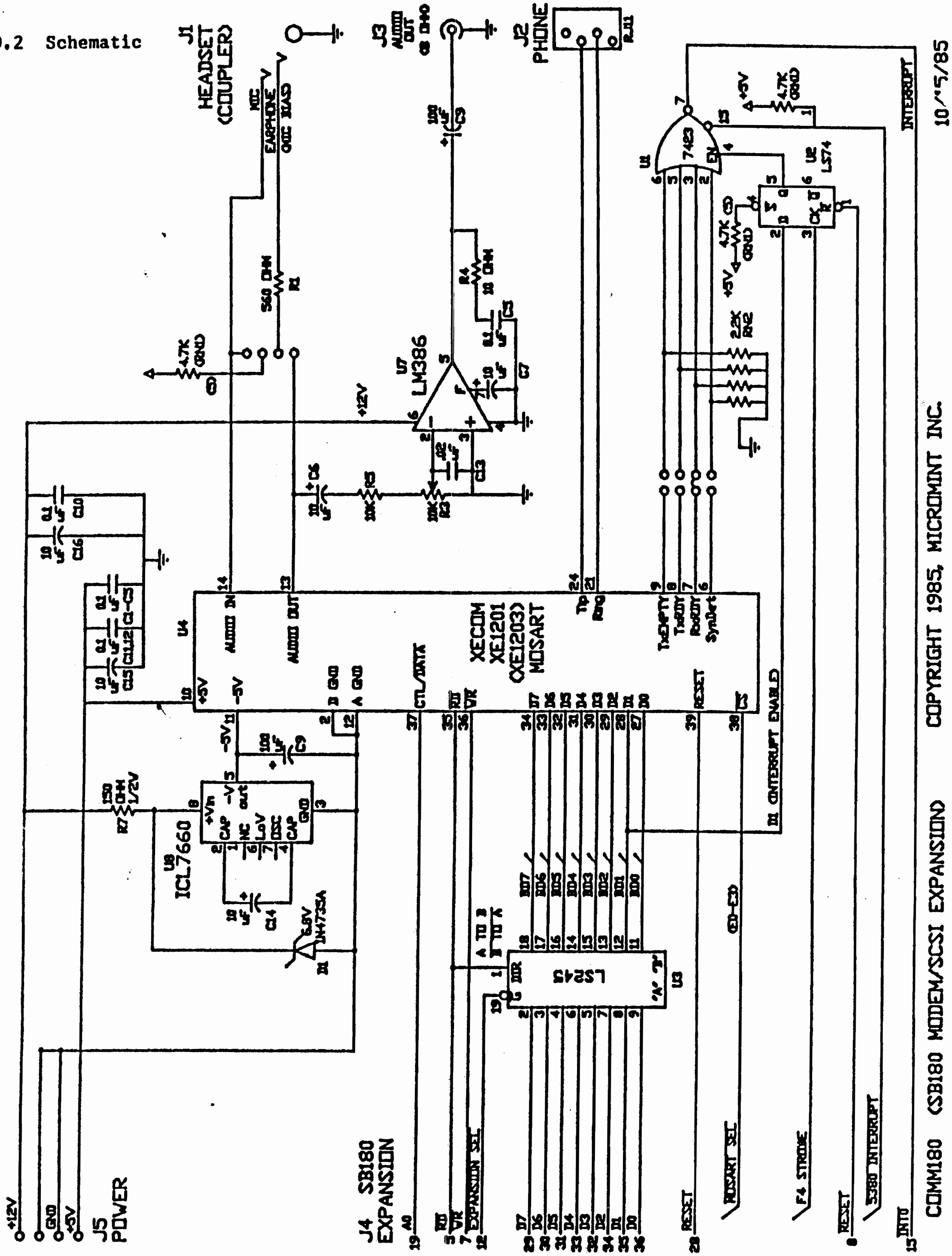
TRANSISTORS

Q1	PN22222, NPN, SILICON



COMM180 (SB180 MODEM/SCSI EXPANSION) SCSI SECTION COPYRIGHT 1985, MICROMINT INC. 11/11/85

10.2 Schematic



MOSART:	Data Register	Port # ØEØH
	Command Register	Port # ØE1H
NCR 538Ø:	Register Ø-7	Port # ØF8H => ØFFH

Interrupts:

- To enable interrupts from both ports, write Ø2H to port ØF4H.
- The 538Ø interrupt is combined with whatever MOSART interrupts are jumpered and produce the INTØ signal for the 6418Ø (if interrupts are enabled).

538Ø DMA Acknowledge:

- To send a latched DMA Ack. to the 538Ø, write a Ø1H to port ØF4H. (To leave interrupts enabled, write a Ø3H.)

- To clear the acknowledge, clear bit Ø.

```
         ---------------------------------
ØF4H    | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Ø |
         ---------------------------------
                                 |   |_____ DMA Acknowledge
                                 |
                                 |_________ Enable interrupts
```

JP2 MOSART interrupts

```
    Tx Empty ___           _____ Rx Ready
                |          |
              - - - - - - - - - - - -
             |  *     *     *     *  |
             |  *     *     *     *  |
              - - - - - - - - - - - -
    Tx Ready _________|          |______ Syn Det
```

JP1 Headset Configuration

```
   - - - - - - - - -          - - - - - - - - -
  | *   *---*   *  |         | *---*   *---*  |
   - - - - - - - - -          - - - - - - - - -

    Microphone only            Mic and earphone
```

A. COMM180-S

The COMM180-S is a general purpose SCSI interface for the SB180 single-board computer. If the COMM180 is to be used as an interface to a hard disk controller, the BIOS upgrade software provided with the board may be used. The upgrade software requires either a Xebec 1410 or an Adaptec ACB4000 hard disk controller. Additionally, any size hard disk drive up to 32 Megabytes must be connected to the controller board. There is also a pre-configured selection designed for a 10 Megabyte Xebec Owl.

Installation

1. Install the COMM180 board as described in section 2 of this manual.

2. Connect a 50 pin cable between your hard disk controller board and the COMM180's 50 pin connector. Pay careful attention to the pin 1 markings on the cable and connectors.

3. On both the Xebec 1410A and the Xebec Owl, there is a three pin jumper header labeled 'SS' with the option of placing the jumper next to a '2' or a '5'. The sector size ('SS') that the COMM180-S software uses is 512 bytes, so the jumper should be placed next to the '5' (512 byte sectors) instead of the '2' (256 byte sectors).

4. The COMM180-S software expects the hard disk controller board to be set-up for a target ID of zero. Consult your controller board's manual to find out how to jumper the board for an ID of zero.

5. Boot up your system as you usually do. If there is no response when power is applied, turn the power off and recheck the seating of the COMM180 board as well as all the cables.

6. Prepare a copy of your present system disk. You will install the new hard disk BIOS on it in step 8.

7. Reboot the system using the copy of your system disk. Place the COMM180-S master disk in drive B:.

8. When the system is done booting, enter 'SYSGEN'. Answer 'B' for source drive and hit return. Answer 'A' for destination drive and press return. (If you only have one floppy drive, you will have to answer 'A' to both questions and do some disk swapping for this operation.)

9. Press reset and reboot the system.

10. Next, you need to initialize the hard disk. Type 'B0:;HDINIT'. The default configuration followed by a menu of commands will be displayed. If you want to use the configuration

that is displayed, simply choose to format the disk (option 2). If you want a different configuration, select option 1. The program will prompt you for all the information it needs. If you are unsure of what to enter for a particular question, just press return and the default value shown in parentheses will be used. You may have to consult the manual that came with the hard disk drive to find answers to some of the questions. When you are satisfied with the configuration that is displayed, format the disk (option 2). (If the program keeps refusing your keystrokes, double check that your terminal is set-up for 8 data bits, 1 stop bit, no parity and that when the return key is pressed, only a CR is being sent by the terminal and not a CRLF combination.)

11. Next, verify the disk (option 3). The program will display the cylinder it is verifying and will tell you if there are any bad sectors on the disk. If the disk checks out OK, go on to the next step. If there is an error, try the verify again. If the error persists, format and verify the drive again. If the error continues to occur, press ESC to leave the program and use the FBAD utility to find and tag the bad sectors. Be sure to run FBAD for each logical drive that is defined. For example, if you selected two partitions, the hard disk would be broken into two logical drives, E: and F:. You would have to issue 'FBAD E:' followed by 'FBAD F:' to check the whole disk. See the description of the FBAD utility for more details.

12. At this point, you should have an error-free formatted hard disk. The next step is to place the system image onto the hard disk's system tracks. This system image is used whenever the computer does a warm boot during normal operation. Type 'PUTSYS' and answer 'Y' to the question. You now have a functional hard disk system.

13. You may now operate your system in either of two disk contexts. If you prefer floppy disks as your regular working media, the system defaults to this mode. Known as the floppy context, the floppies remain drives A: through D: and the hard disk is accessed as drives E: through H:.

However, we recommend that you run the change disk context program with the 'H' option and operate under the hard disk context. The command 'CDC H' swaps drive assignments between the hard disk and the floppies. The floppies are now mapped to drives E: through H: and the hard disk is mapped to drives A: through D:. (The highest drive letter allowable is, of course, dependent on the number of floppies you have and the number of partitions you've defined.)

Cold Booting

The decision was made not to develop a new monitor ROM to allow cold booting from the hard disk. There are a number of reasons for doing this.

First, your system must have at least one floppy disk drive on it anyway. Without the floppy on the system, there would be no way to get files from the outside world onto the hard disk. You also wouldn't be able to back-up the hard drive without the floppy.

Second, most hard disk drives need time after the power is first applied to spin-up to the correct speed for operation. The most efficient way of utilizing the spin-up time is to start the cold boot from the floppy. For example, the computer can be loading the RCP, FCP, NDR, and terminal descriptor from floppy while it waits. After this is done, the start-up alias can transfer control to the hard disk and the boot process can continue.

Finally, many people don't like to deal with changing chips on their system. Even though installation of the new EPROM would be optional, some would feel obliged to install it and risk damaging their system.

The following is an example of a pair of start-up aliases that could be used on a hard disk system. Note how the first alias on the floppy links to the second on the hard disk.

```
START.COM
        (on a floppy disk in the first floppy drive)

     LDR SYS.RCP,SYS.FCP,SYS.NDR,MYTERM.Z3T;
     CDC H;
     STARTHD


STARTHD.COM
          (on the first logical partition of the hard disk)

     MDSK I;
     CP M:=DIR.COM;
     PATH M0 A15;
     VID HELLO
```

In the first alias, the first floppy drive starts out as drive A:. The LDR and CDC commands are loaded off the floppy and are executed. After CDC has finshed, however, drive A: suddenly refers to the hard disk. The computer will find STARTHD on the hard drive and processing continues.

Of course, all the files referenced by the aliases must be available at the time they are run.

Distribution Disk Software

The following section describes each of the programs on the distribution disk.

HDINIT

Parameters: None.

Description:

The Hard Disk INITialize program performs all that's necessary to configure your system and hard disk drive to your specifications. It is fully menu driven and, in most cases, self explanatory. If you ever have a doubt as to how to answer a question, simply press return. The default answer (shown in parentheses in most cases) will be used by the program.

(Note: When verifying the hard disk, a bad sector will cause the verify function to abort. If this occurs, use the public domain program FBAD to find and mark all the bad sectors. See the public domain section for more details on FBAD.)

CDC c

Parameters:

c = H - hard disk context
    F - floppy disk context

Description:

Change Disk Context. It is often useful to be able to map drive A: to the hard disk instead of to a floppy. This allows the system to warm boot from the hard disk, resulting in faster warm boots and freeing you from keeping a boot disk in the first floppy drive.

CDC is used in one of two ways. The command 'CDC H' will set-up the system so the hard disk drive is drives A: through D:, depending on how the drive is partitioned. (If, for example, it is partitioned into three logical drives, they will be known as drives A: through C:.) The floppies are then known as drives E: through H:, depending on how many floppy drives are connected.

Before the CDC H command is issued, however, you must have written a system image to the system tracks of the hard disk drive. If this isn't done, the computer won't be able to do a warm boot after the CDC command has finished, and the system will hang. The procedure for writing to the hard disk's system tracks is described in the installation section.

The other valid usage is 'CDC F'. This simply undoes the CDC H command. The floppies revert back to A: through D: and the hard disk becomes E: through H:.

Another side effect of the CDC command is that some system utilites designed to run on a floppy-only system will continue referring to the floppies as drives A: through D:, regardless of the current disk context. Two examples of this are FVC and

SYSGEN. Suppose you want to format a disk in the first floppy drive, but you have already run CDC so that the first drive is drive E:. FVC won't accept E as a valid drive, so you must enter A. (Don't worry, FVC will never attempt to format your hard disk drive.)

A variation of this is CONFIG. While it correctly recognizes the current disk context, it only accepts A through D as a valid drive. Therefore, if you want to use CONFIG to change the parameters on a disk in, say, the first floppy drive, the system must be in the floppy context so you can refer to that drive as drive A:. Issue the CDC F command to put the system in floppy context.

PUBLIC /ooooo...

Options:

Cc - Clear PUBLIC declarations
 c=A Clear All PUBLIC declarations
 c=D Clear PUBLIC User Area Drive
 c=U Clear PUBLIC User
Dd - Declare Drive=d, PUBLIC
Uu - Declare User=u, PUBLIC

Description:

PUBLIC declares to ZRDOS 1.2 what drive and user areas are to be regarded as public. Any file kept in a public user area can be accessed directly from any other user area on the same drive as long as the fcb specifying the file contains no wild cards. Public user areas can be on drives A-H and user areas 1-8 and can be used to keep the overlays used by programs such as WordStar, dBASE II, and SuperCalc as well as many other commercially available programs. See the PUBLIC.HLP file for more information.

DIR13 dir:afn o...

Options:

A - Display both non-system and system files
S - Display only system files
T - Display files sorted by file type and name

Description:

The latest version of the <u>DIR</u>ectory utility. The main improvement involves recognition of PUBLIC directories. A bug in sizing large files was also fixed. See <u>ZCPR3: The Manual</u> for more details on its usage.

DOSERR n

Parameters:
    n - ZRDOS 1.2 error number.

Description:
In order to add the PUBLIC feature to ZRDOS 1.2, sacrifices had to be made so the code would fit into the same space as the older versions. One of those sacrifices is the elimination of verbose error messages, replacing them with number codes. Running DOSERR with a ZRDOS error number will give a description of the error.

MOVE12 du:afn user:
       du:afn dir:

Parameters:
    user: - destination user area.
    dir: - destination directory name.
    (Both user: and dir: must be on the same drive as du:.)

Description:
This utility will MOVE a file from one user area to another on the same disk. It accomplishes the task very quickly since it doesn't actually copy the file, but changes a byte in the directory.

CONFIG12

Parameters: None.

Description:
This latest version of CONFIG allows you to set the head load parameter of the FDC9266. Drives with a head load solenoid typically need about a 36 ms head load delay, though it may be different on yours.

PUTS ufn.typ

Parameters:
    ufn.typ - name of the file containing a system image.

Description:
This command reads a system image from the given file and writes the image to the system tracks of the hard disk. See the customization section for more details on the usage of this command.

PUTSYS

Parameters: None.

Description:
Similar to PUTS, this command already contains the system image, so doesn't have to read it from disk. Typing 'PUTSYS' places the system image onto the hard disk's system tracks.
An alternate use for this file is to use it with SYSGEN to place the system image onto a floppy's boot tracks. The command 'SYSGEN PUTSYS.COM' will use just the system image portion of the PUTSYS.COM file to write the image to the specified floppy. See the customization section for more details on creating the PUTSYS.COM file after changes to the BIOS or ZCPR3.

Public Domain Software

Very often, when a task needs to be performed, but the tool or utility necessary to perform the task isn't provided, there is a public domain utility which fits the bill. The following public domain software has been found by us to be useful in the everyday operation of a system with a hard disk and is included at no extra cost to the user. However, since it is public domain, we can't guarantee its operation or provide support if bugs are found.

FBAD d:

Parameters:
d: - drive containing disk to be checked. May be a floppy or hard disk.

Description:
Find BAD will find and mark all the bad sectors on a disk, either floppy or hard. The program keeps track of any bad sectors it finds, creates a dummy file called [UNUSED].BAD, and allocates the bad sectors to that dummy file. The dummy file doesn't contain any information, but since it is supposedly using the bad sectors, other files are prevented from using them. You may want to use SFA to make [UNUSED].BAD a system file so it doesn't show up in a normal directory and to make it a read-only file so it can't be accidentally erased. (The command line is 'SFA [UNUSED].BAD /SYS,R/O'.)

BU

Parameters: None.

Description:
This utility will BackUp the files on your hard disk drive to floppy drive. The program prompts for all the information it needs and is very easy to use. (A printer is required, however.) Please see the documentation file provided on the disk for more information on its usage. Note that the source file provided is an .ASM file and must be converted before assembling with ZAS.

MLOAD [outfil=]file1[,file2...] [bias]

Parameters:
outfil - optional output file.
file1, file2, ... - input files.
bias - optional bias address.

Description:
While not truly public domain, MLOAD can be freely copied and distributed. MLOAD will create executable code (.COM file)

from a .HEX file. It will also combine a number of .HEX files into one .COM file, or overlay a preexisting binary file with a converted .HEX file. Type 'MLOAD' with no parameters for more details. Note that the source file provided is an .ASM file and must be converted before assembling with ZAS.

Customization

With all of the compile-time options available in ZCPR3, the user may not be happy with the way his system was set-up by Micromint. To modify any of these options, however, requires the user to change one or more files, reassemble the modified portions, and regenerate the operating system. While not difficult, there are certain files which must be present and steps must be done in a certain order. The process of modification and regeneration will be described here. Further details may be obtained from the book ZCPR3: The Manual.

1. System regeneration is required only when modifications are made to the BIOS or to ZCPR3. Files such as FVC.Z80 or SYSRCP.Z80 aren't integrated into the system, so may be changed independently. To be reassembled, BIOS.Z80 needs the library files PORTS.LIB, Z3BASE.LIB, SYSENV.LIB, and NCRIO.LIB. ZCPR3.Z80 needs Z3BASE.LIB and Z3HDR.LIB. You should never have to make changes to the ZCPR3.Z80 file directly since all the user selectable options are contained in the Z3HDR.LIB file.

2. Copy all the files you will need to a fresh disk. This should include source files, associated library files, your text editor, and a copy of ZAS.

3. Make your changes to the source (or library) files. Assemble them with ZAS to create .REL files (either BIOS.REL, ZCPR3.REL, or both).

4. Now it's time to generate a new system. Rather than copy all the files you will need to a new disk one file at a time, it is easier to copy all of disk four (from your SB180-20 package) with FVC and delete the files you don't need. Make a copy of disk four and delete all the help files (use the command 'ERA *.HLP') to make some work space. Copy your new .REL files to this disk, replacing any old files which may already be there.

5. You must also configure ZDMH before it can be used in the generation process. Run ZDMH and tell it how wide your screen is as well as how many lines it has. Be careful, it expects these values in hexadecimal. A new file, called ZDIH.COM, is then created and the program stops. Rename this file to ZDM.COM so it can be used later.

6. Now you are ready to let the computer do its thing. Simply type 'ZEX MKZSYS'. This batch file links together a new system and creates the file MOVZSYS.COM. Next, type 'MOVZSYS 54' to actually create the system image in a file called ZSYSTEM.MDL. To place the new system on the boot tracks of a floppy disk, put a copy of SYSGEN.COM on this disk, type 'SYSGEN ZSYSTEM.MDL', and follow the instructions that appear on the screen. To put a copy of the new system on the system tracks of the hard disk, put a copy of PUTS.COM on this disk and type 'PUTS ZSYSTEM.MDL'. Boot the new floppy and you are running your new system.

7. It is possible to create a new PUTSYS.COM file so the ZSYSTEM.MDL file isn't needed to put a system image onto the hard disk's system tracks. The command:

```
MLOAD PUTSYS.COM=ZSYSTEM.MDL,PUTS.HEX
```

will create the new file.