GTE LENKURT

DEMODULATOR

SEPTEMBER/OCTOBER 1978

MICROPROCESSORS IN TELECOMMUNICATIONS

Over the past decade Large Scale and Very Large Scale Integrated Circuit technology has evolved so rapidly it is revolutionary. The advances equal or surpass the transistor advances of the late 1940's. LSI makes it possible to put a very large number of electronic circuit elements onto a single chip.

Using this technology, the entire central processing unit of a computer can be mounted on a single chip called a microprocessor. The microprocessor is able to perform the same functions as larger CPU's. One such chip, the Intel 8085, is a rectangle 0.222 inches long, and 0.164 inches wide. It has 6,200 transistors and can execute 770,000 instructions per second.

Microprocessors are rapidly finding uses in telecommunications equipment. Their low-cost, high-reliability and ability to process large amounts of information make them ideal for monitoring and control, remote diagnostic and switching applications.

The February, 1977 issue of the Demodulator described microprocessors as a part of a computer. This Sept/Oct 1978 issue discusses the use of microprocessors in telecommunications equipment. These uses may be divided into two categories; One where the microprocessor is considered as a component and the other where it is considered as a system. Table I lists some of the distinctions between the two categories. The categories and distinctions are general statements which have been formulated to aid our discussion. Specific applications may be a mixture of both categories.

From the component viewpoint, the designer sees the microprocessor as a Large Scale Integrated (LSI) circuit. He describes his device as a black box with inputs, outputs and a timing diagram. The end product of his design effort is a hardware component capable of performing the functions for which it is intended. The primary factors he considers in selecting a microprocessor for his application are function, time and cost. The possibility of using his device for other applications is seldom a consideration.

In these component applications the "software" is often referred to as "firmware". It is generally stored in a

read only memory (ROM) component. Once in ROM, software is considered fixed. Changes are held to a minimum.

From the system viewpoint, the designer sees the microprocessor as a data processor. He describes his device in computer and software terms. His end product is largely software. His choice of microprocessors is limited to reduce the requirements for high cost development systems and training.

The software for the system approach is designed to be flexible and easily changed to meet system needs. It is usually stored in Random Access Memory (RAM). This is read/write memory which can be altered by the designer or end user. It is often stored on magnetic tape or discs.

Noise and errors are problems encountered in both products. Systems designers handle these problems by a software "Error Exit" module or a hardware interrupt that periodically forces a return to an executive module. In this type of project, memory is cheap and design time is expensive.

Executive modules are high-level software that provide timing, scheduling, input/output signals, interrupt handling and similar functions. An interrupt is an external event that controls the microprocessor. When an interrupt occurs, the microprocessor interrupts its normal routine to handle the interrupt request.

The component designer must handle noise and errors by careful programming. The memory and external hardware at his disposal are limited. They must be used efficiently. Extra design time is easily justified if it results in fewer parts and lower cost of the end product. As shown in Table 1, the logic component design is generally a one man effort whereas the data processor design is usually accomplished by a team.

	LOGIC COMPONENT VIEWPOINT	DATA PROCESSOR VIEWPOINT
IMPLEMENTATION CONSIDERATIONS	MICROPROCESSOR CONSIDERED AS A COMPONENT	MICROPROCESSOR CONSIDERED AS A SYSTEM
	I/O PINS CONSIDERED SEPARATELY	WORD ORIENTED I/O (BUS ARCHITECTURE)
	COMPONENT SELECTION BASED ON SYSTEM NEEDS	LARGE INVESTMENT IN A FEW COMMON DEVICES
	LARGE VOLUME PRODUCTS	SMALL VOLUME PRODUCTS
	ONE MAN DESIGN	TEAM/GROUP DESIGN
PROGRAMMING CONSIDERATIONS	RESOURCE EFFICIENT PROGRAMMING	STRUCTURED PROGRAMMING
	LOW LEVEL LANGUAGE	HIGH LEVEL LANGUAGE
	LOGIC DESIGN	DATA MANIPULATION
	BIT ORIENTED	WORD ORIENTED
	FIXED	FLEXIBLE
	NOISE & ERRORS HANDLED BY PROGRAMMING METHODS	NOISE & ERRORS HANDLED BY SEPARATE MODULE OR EXTERNAL HARDWARE
HARDWARE/SOFTWARE CONSIDERATIONS	HARDWARE INTENSIVE	SOFTWARE INTENSIVE
	HARDWARE/SOFTWARE INTERACTIVE	HARDWARE/SOFTWARE SEPARATE
	HARDWARE/SOFTWARE TRADE-OFF DECISIONS MADE DURING IMPLEMENTATION	HARDWARE/SOFTWARE TRADE-OFF DECISIONS MADE DURING PLANNING
MAN/MACHINE INTERACTION	MINIMAL	SUBSTANTIAL

Table 1. Comparison Component Vs System

Before proceeding with our discussion, let's define the processor languages which are used. These languages range from the object code for a specific processor to high level languages which may be common to many processors.

Language Definitions

Object Code – Also known as "machine language", Object code language is expressed in binary digits. It makes the most efficient use of computer resources. However, programming in object code is a time consuming process. It is so slow and tedious that it is rarely used for programming today.

Assembly Language – Assembly language is a fundamental programming language devised for programming convenience. It substitutes mnemonic (memory aiding) abbreviations for binary coded computer instructions. An individual abbreviation is assigned to each computer instruction.

Programs written in assembly language must be converted into machine language so the computer can execute the instructions. The conversion is accomplished by a program called an assembler. Generally, a particular assembly language is written for, and only able to run on, one type of microprocessor or computer. Assembly language is computer instruction oriented.

High Level Languages – High Level Languages are the easiest to use. They are closest to ordinary english and, in contrast to assembly language, are written in terms of the problem to be solved instead of in computer instructions. Just as assembly languages require assembler programs to translate the mnemonics into machine language, higher level languages require a compiler or interpreter to translate the high level program into object code. However, high level languages may be used on any computer providing it has a compiler or interpreter for the language used. Examples of high level languages used with microcomputers, are: BASIC, PL/M, Pascal, COBOL and FORTRAN.

Programs written so they can be read by humans are called Source Programs. Programs written in machine language are called Object Programs.

Since higher level languages are easier to use and have broader application than assembly languages, the question arises; Why use assembly language at all? The answer lies in the efficient use of computer resources. The object program resulting from an assembler translating an assembly language source program is much shorter than the object program resulting from a compiler translating a high level language source program.

In fact some machines do not have sufficient memory to use high level languages. Even if the computer has sufficient memory, the object code resulting from high level language may take longer to execute than the object code resulting from assembly language. This can be an important factor in applications where speed is important. However, it should be noted that some of the newer high level language compilers, Pascal for example, do produce efficient object codes. These compilers can be "out programmed" only by experienced programmers.

Returning to our discussion of the two design concepts, when the microprocessor is considered as a component, assembly language will probably be used for programming. High level language will probably be used for programming the microprocessor based system.

Figure 1 shows the steps a designer might use to design a microprocessor

component. Figure 2 shows the steps for designing a microprocessor based system. The following paragraphs discuss each figure in turn and compare the two.

Referring to Figure 1, the logic diagram at the left is a schematic representation of a function to be performed by the microprocessor. The function might be to examine the status of a number of remote alarms and provide an indication if an alarm exists.

The next block represents a series of source statements written in assembly language to form a source program. The mnenomic STRT T means start the timer. The next three steps are used to set a bit. The statement INA, P2 means to get the information at port 2 and store it in the accumulator. The third statement ORL A #02H means OR the contents of the accumulator with bit pattern 02H. This is the Hexadecimal representation of the binary number 00000010.

The 1 in the second bit of this number assures that the output of the OR gate will be a 1 in that position, regardless of the other input. In other words the second bit at the OR gate output is set to 1.

The next instruction tells the microprocessor to jump to the next function. As previously stated, the assembly language source code is translated into a machine language object program to operate the microprocessor. An object program is shown in the next block of Figure 1. There is a direct relationship between each statement in an assembly language program and its resulting object program. Each assembly language line has a corresponding object language line. This bears out the statement that assembly language is computer oriented. The source program is a step by step set of instructions for the computer.

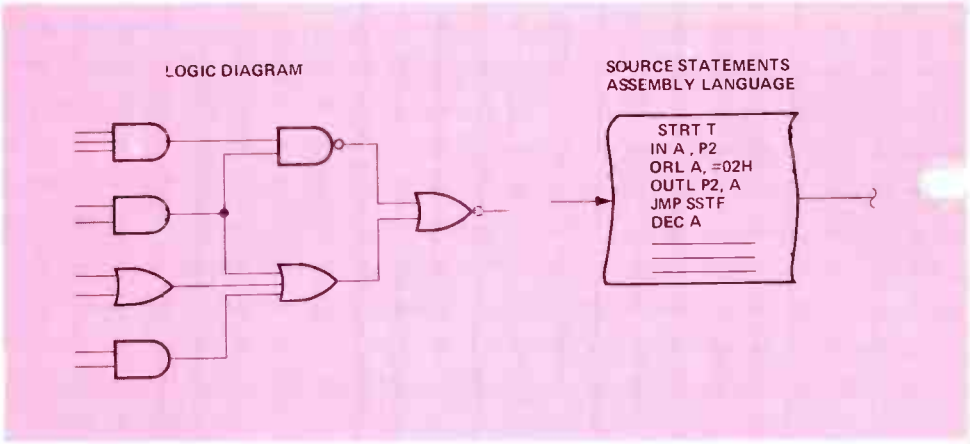
When the designer has produced and tested a satisfactory object program, he records it and sends the record to the manufacturer. Frequently, the recording disc is supplied by the manufacturer.

The manufacturer uses the recording to fabricate a metal mask like the one shown in Figure 1. This mask serves as a kind of template. It is placed over a micro-chip substrate and the program is "burned" into the chip to produce the ROM. This kind of ROM is sometimes called a PROM. The "P" stands for programmable. The PROM is a component part of the microprocessor and is enclosed in the same plug-in package.

It is true that some PROM's are optically-erasable with ultra violet light. However, the erasure is non-selective. The entire program must be erased and a new one inserted if changes are to be made. Given the economics and logistics of the telecommunications industry – this is not a feasible procedure once the units are in the field. Therefore, once a program is in PROM it is fixed for all practical purposes. An erasable PROM is called an EPROM.

Microprocessor Based System

Referring to Figure 2, the first design step is to formulate the system requirements and express them in the form of a hierarchy chart similar to the one at the left of the figure. This type of chart is frequently used in "top down" program design. Each block represents a major subsystem which can be represented by its own hierarchy chart. If the subsystem is fairly elaborate, some of its assemblies may have their own charts. If all these charts are put together to form one large chart, the uppermost blocks will be an abstract representation of the system. As we read down through the



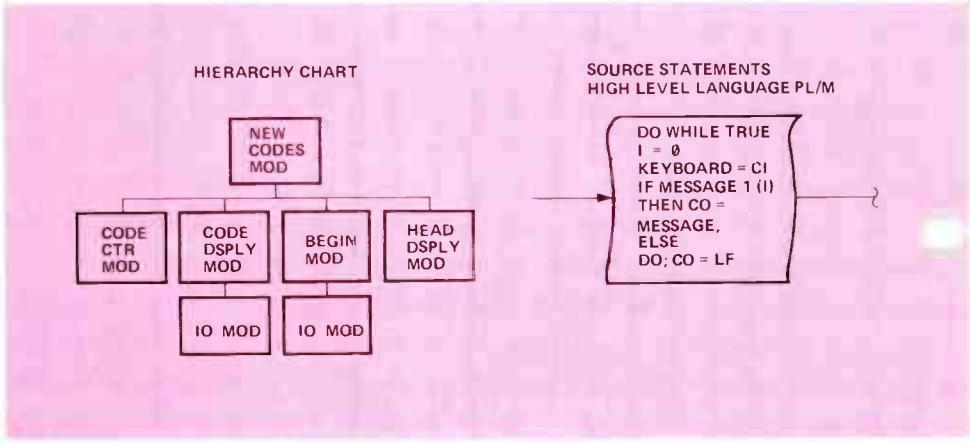


chart the blocks become more specific.

The second block in Figure 2 is a series of statements written in PL/M. This is a high level language devised by Intel Corporation especially for its microcomputers. The language's similarity to ordinary English is evident in the diagram. The IF, THEN, ELSE statements are frequently encountered in high level languages. In our example the statements means: IF there is a message at I THEN process it. ELSE (if a message is not present) process the instruction at address LF.

The next block in the figure is the familiar object program. One fact that is not apparent from the figure is that several lines of object code may result from a single high level language source program instruction, in contrast to assembly language source programs where one object code instruction results from one assembly code instruction.

Once the object program has been established, it is recorded on tape or disc and is read into the microprocessor's read-write, random access memory (RAM) as required. Changes, additions or corrections to the program are relatively easy in contrast to the fixed programs in ROM.

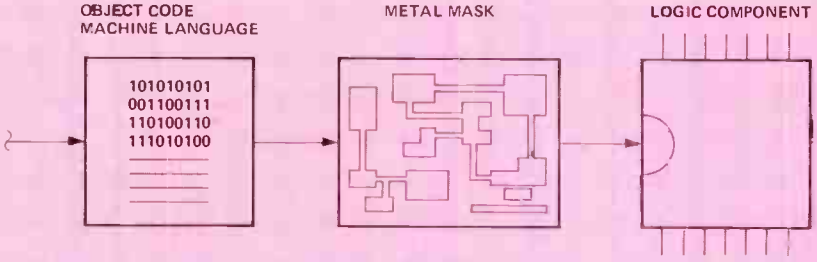


Figure 1. Design Steps — Microprocessor as a Component.

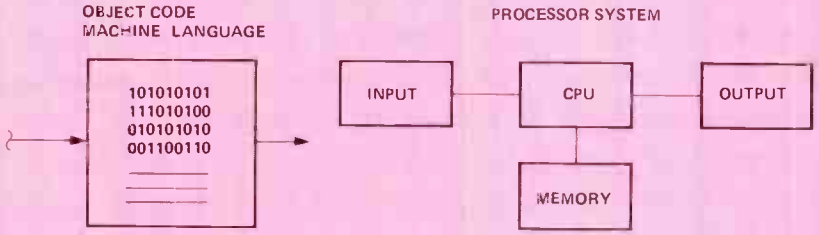


Figure 2. Design Steps — Microprocessor Based System.

Often the design of a particular system is a mixture of both approaches. The master system is designed from the data processor viewpoint while some of the subsystems are designed from the component viewpoint. Sometimes the design approach used is a matter of engineering preference. A hardware oriented engineer will use the component approach while an EDP oriented engineer will prefer the data processor approach. The data processor design method could be used in all cases but the component approach is limited.

Now that we have discussed two design concepts and the differences between them, let's consider an example of each type.

Component Example

The first example is a component type system used to provide alarm and control functions for a PCM channel bank. The microprocessor and associated circuitry are mounted on a card which plugs into a terminal shelf. Figure 3 is a partial block diagram of the system.

The alarm and control unit, in the center of the figure, monitors the operation of the local and remote

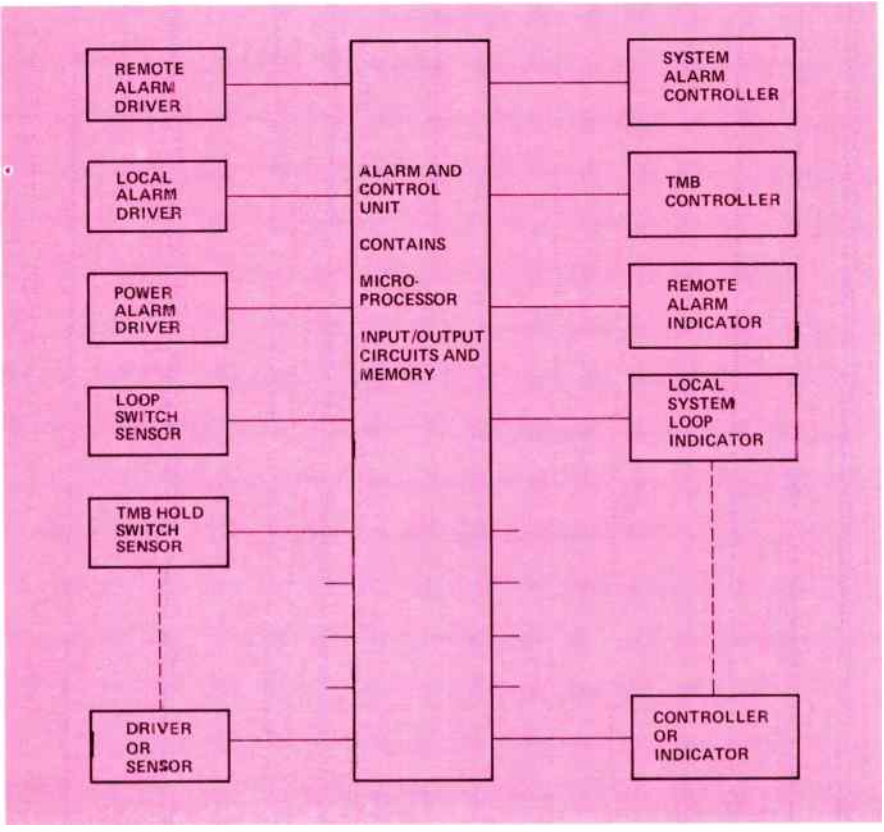


Figure 3. Monitoring, Alarm, and Control Unit Characteristics – Hardware intensive, Devices chosen by system needs, Bit oriented input/output pins considered separately, Maximum number of drivers and sensors that can be used is dependent on input pin count.

common equipment and the local power supply. An alarm condition in any of this equipment will initiate the trunk make busy (TMB) sequence. The TMB busies out all channel units during a system alarm.

The alarm and control unit also actuates electronic exchange, audible and visual alarms. An alarm cutoff (ACO) switch is provided to silence the audible alarm. When the ACO switch is actuated, the system may be held in the TMB state by a TMB hold switch. This is useful for trouble-shooting purposes. Operating the ACO also enables a loop test function which loops the terminal onto itself for alignment and testing. The alarm and control unit also accepts a loop pulse request and, if the terminal is looped, provides a loop pulse for looped line testing.

The easiest way to explain the microprocessor operation is by an example. Let's look at the operation of the power and local alarms. The assembly language program for these alarms is:

Assembly Language	Meaning
INA, PØ	Input to the accumulator the contents of port zero.
ANLA, #ØCØH	AND logic the accumulator with the hexadecimal number ØCØ. This step masks off all but the fifth and sixth bits which are the locations of the local and power alarms respectively.
ORL A, R2	OR logic the contents of the accumulator with the contents of register 2. A register is a place where data is stored.
MOV R2, A	Move the contents of the accumulator to register 2.
	Time interval while computer is looking at other inputs.
MOV A, R2	Move contents of R2 to the accumulator.
ANL A, #3ØH	AND logic the contents of the accumulator with the hexadecimal number 3Ø. This step AND's the contents of the accumulator against a number which provides a binary 1 in the 5th and 6th digits. If the accumulator also has a 1 in either of these positions there will be a 1 output-alarm condition.
JNZ SDLY	Jump if ANL A, #3ØH is not zero to command SDLY.
	Steps if ANL A, #3ØH is zero.
SDLY MOV RØ, #7AH	Move to register Ø the hexadecimal number 7A. (0111 1010 binary) This sets a 10 second delay count and system alarm.

Note that what we have done could be represented in logic circuitry by an OR gate. The local and power alarms are effectively OR'd and if either one is in an alarm (logic 1) state, a 10 second time delay is started and 2.5 seconds later, a system alarm is indicated.

At the end of the 10 second delay, the microprocessor starts a 5 second delay and looks at the alarm status. If the alarm has cleared and does not recur during the 5 second interval, the system is returned to normal. If the alarm has not cleared or recurs during the 5 seconds — the 5 second counter is reset. In other words, the minimum time for an alarm to clear is 15 seconds. Figure 4 is a timing diagram for the circuit.

One important thing to note about the component system is; there is very little interaction between man and the system. Aside from the toggle switches which may be used to silence the audible alarm and initiate certain test sequences, the operation is completely automatic.

This is in marked contrast to the interactive system we will discuss next. In fact, most microprocessor based systems provide a great deal of this

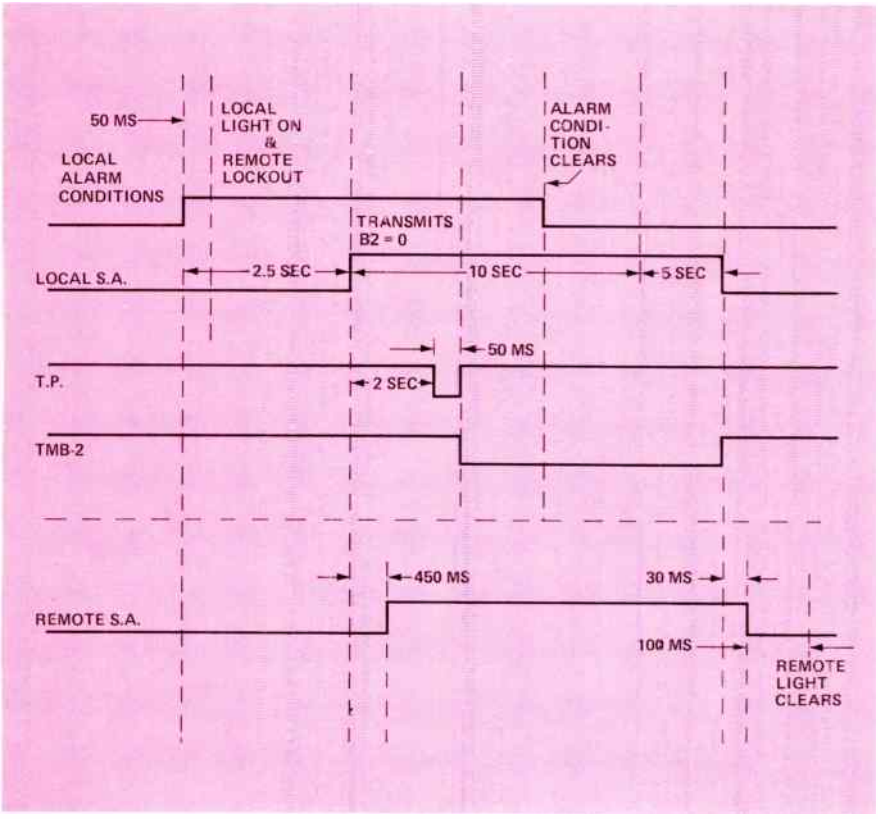


Figure 4. Component Timing Diagram.

interaction even though the man-machine interface is difficult to design and expensive to implement.

Microprocessor Based System

Figure 5 is a block diagram of a monitoring and control system (MACS) used for supervision of a digital radio system. In contrast to the first example where the microcomputer is mounted on a single board, the MACS uses several plug-in units which are housed in an auxiliary shelf. The keyboard and displays are mounted on the shelf's front panel (see Figure 6).

The front panel and plug-in units are interconnected by a Bus. In this case Bus means high speed electrical paths interconnecting the units forming the basic computer. A Bus usually provides parallel paths for transmitting information, whereas the interconnection between the computer and remote peripheral devices, and between the computer and communications equipment, is transmitted serially.

In the United States, the interface connection between data terminal (computer) equipment and data communications equipment is usually governed by EIA Standard RS 232C, although this will eventually be replaced with RS 422 and RS 423. In other United Nations countries, the interconnections are governed by vari-

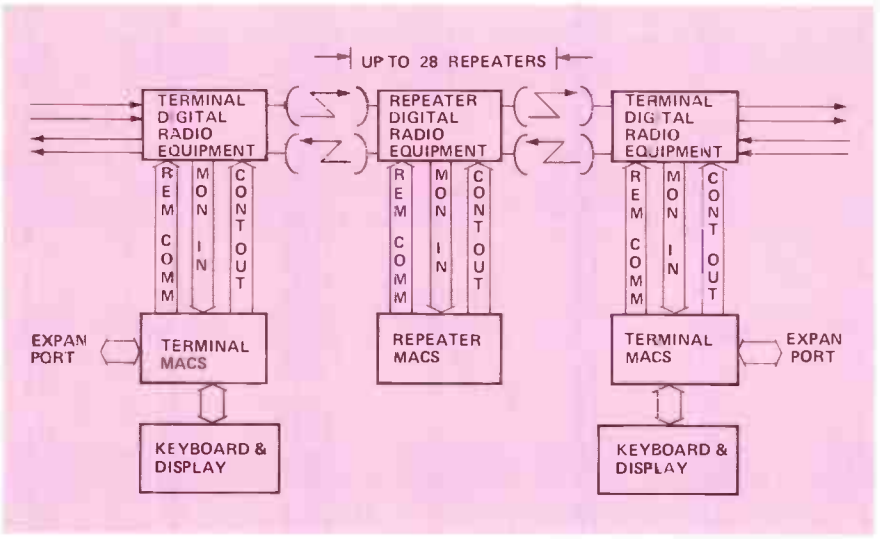


Figure 5. MACS Block Diagram.

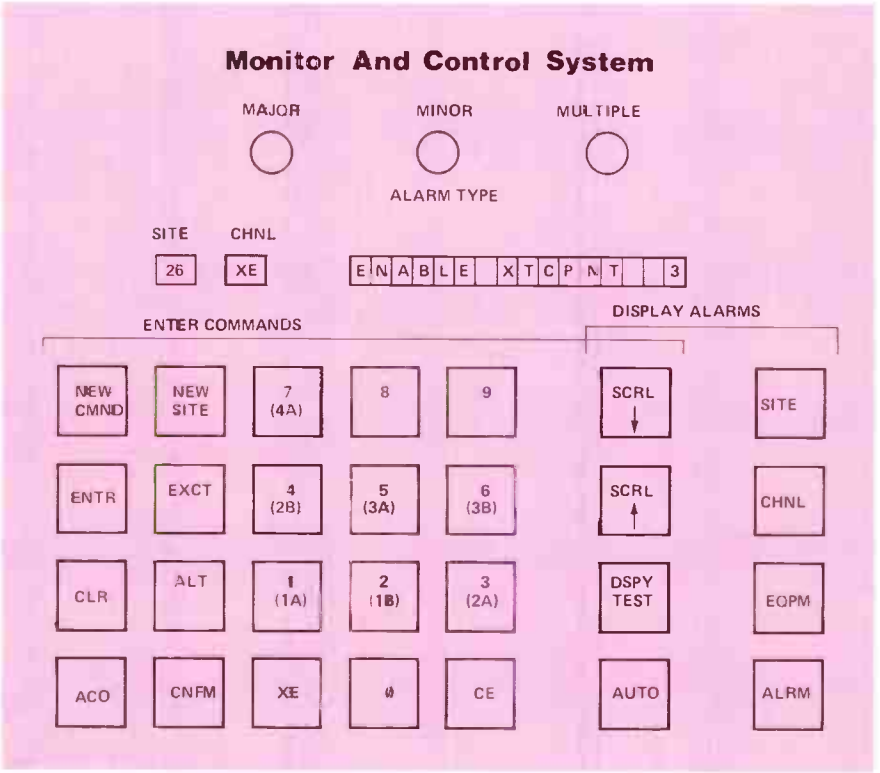


Figure 6. Display Panel.

ous CCITT recommendations including V24, V28 and V35.

Network Limits

Returning to our discussion of MACS, the digital radio network being monitored may have up to 30 stations consisting of terminals, repeater and drop and insert sites. The number of drop and insert "T's" is limited to 7, in this application.

Features at Each Location

The MAC system periodically monitors and stores the state of each individual alarm, summary alarm, front panel switch guard and protection relay in both the digital processing and radio equipment. At each site, 60 alarm points are monitored for each duplex channel. Also, at each site the MACS monitors and stores the state of 32 external station points selected by the user. The choice of major or minor alarm designations, for these points, is also selected by the user at the time of installation.

The system is capable of monitoring all the alarms and relays states of:

a) A single unprotected duplex digital radio channel.
b) Two 1:1 protected duplex digital radio channels with or without space diversity.
c) Four separate but parallel duplex digital radio channels without space diversity receivers.

A four duplex channel system with 30 sites has 60x4+32=272 alarms per site. Multiplying 272 times 30 we find that MACS monitors the status of 8,160 alarm points in the digital system and at the sites.

Remote Communications

The individual stations send their stored information on a polled basis. One terminal MACS is designated as the synchronizing master and generates all network polling signals required to communicate with each site. A polling signal is a request to transmit data.

The total time to complete one full network polling cycle is less than 4 seconds. Provisions are incorporated in the MACS to provide an alarm at all terminal units, if the remote communications system looses synchronization.

Up to six MACS terminal units may be designated as Command Originators. These units are able to originate any of the manual action commands listed in Table 2. All remote MACS units in the network are able to respond individually to the commands. The response time to a command cycle is also less than 4 seconds.

Each MACS site has four relays to control external station equipment. These relays are remotely controlled from the Command Originators.

A software counter is provided at each site for each radio channel, to count the number of major/minor alarms at each site in each digital radio path. The count is reported along with the station status during each polling interval. Manual commands may be used to interrogate or reset a specific counter. Counter values are reported and displayed on terminal MACS front panels. The bit error rate (BER) is also determined for each channel at each site and reported during each polling cycle. The BER can also be displayed, on the front panel, by a keyboard command.

Each MACS unit has four digital outputs which can select any of 16 analog voltages at any site. The voltages are measured and transmitted over a compatible digital orderwire circuit, when the digital radio is equipped with such an orderwire system. The voltage to be measured is selected at any one of the command originating terminals. The measure-

ACTION COMMANDS, EXECUTED BY COMMAND ORIGINATORS

1) Enable External Control Point:
Enables one of the four external relays at the specified site in the external equipment channel. The index entry, 1-4, specifies the relay.

2) Enable Protective Substitution Switching:
Enables protective substitution switching for channels 1A-2B (index=1) or channels 3A-4B (index=2) at the specified site.

3) Enable Analog Voltage Reporting:
Enables one of 15 analog voltage reporting channels at the specified site. The index quantity determines which channel is enabled.

4) Enable Equipment (On Line):
A request to switch the specified piece of equipment on line at the specified site and channel. The index quantity specifies the particular piece of equipment. Valid channel entries are 1A, 2A . . . 4B.

5) Disable External Control Point:
Disables one of the four external relays at the specified site in the external equipment channel. The index entry, 1-4, specifies the relay.

6) Disable Protective Substitution Switching:
Disables protective substitution switching for channels 1A-2B (index=1) or channels 3A-4B (index=2) at the specified site.

7) Disable Analog Voltage Reporting:
Disables analog voltage reporting from the specified site.

8) Reset Alarm Counter:
Resets the value of the alarm counter of the specified site and channel to zero. Valid channel entries are 1A, 2A . . . 4B.

9) Reset MACS:
Causes a reset of the MACS system at the specified site.

INQUIRY COMMANDS EXECUTED AT ANY TERMINAL

1) Display Equipment (Status):
Displays the status of the specified piece of equipment, either on or off line. The command entry must contain an index between 1-8 to specify the piece of equipment and a signal channel number 1A, 2A . . . 4B.

2) Display Status Point:
Displays state of a status point at a specified site and channel, true/false, on/off, open/closed or norm/off norm. A status point is a binary signal that does not cause an alarm regardless of state. An index value between 1-8 must be entered.

3) Display Alarm Counter:
Displays a number representing the current value of the alarm counter for a specified site and channel. The displayed value will be continuously updated until another command is entered or the CLR or AUTO keys are depressed. Channel entries may be: 1A, 1B, 2A . . . 4B.

4) Display Bit Error Rate:
Displays a number representing the current bit error rate at a specified site and channel. Valid channel entries are 1A, 1B, 2A . . . 4B.

Table 2. MACS Keyboard Command List

ment and transmission continues until it is cleared by a new request or a reset. Provision are made to ensure that simultaneous measurements are not requested.

Although manual commands which generate external requests (action commands) must be originated by the Command Originator, every terminal MACS has the ability to execute inquiry types of manual keyboard commands which access the internal storage. This fact emphasizes the man-machine interaction that is a characteristic of data processor type systems. The next several paragraphs provide a detailed description of the interaction between the MACS and a technician. The discussion is referenced to Figure 6 and Table 2.

Front Panel Description

Figure 6 is a sketch of the MACS terminal front panel. The panel consists of a keyboard for entering commands, a 16 character alphanumeric display for displaying messages to the operator, a 4 character numeric display for site and channel information and 3 LED indicators to display the alarm type.

The keyboard is divided into two functional groupings, command entry and alarm display. The sixth column of keys is double purpose since it is used for both functions.

The displays are the primary output devices of the system. The 4 character display is divided into two - 2 character fields: site number and channel. A field is a group of characters which are

read as a unit. When in the command entry mode, this display shows the site and channel to which the command is being directed. When in the alarm display mode this display shows the site and channel of the alarm being examined.

Internally there are three registers that may be outputted, one at a time, to the 16 character alphanumeric display. They are the command entry register, the alarm equipment register and the result register.

The command entry register is a 16 character register that contains the mnemonics for the entered command type, qualifier and index. This register is also used for displaying prompting messages during command entry. The first seven character positions constitute the command field and contain the command type mnemonic, the next seven positions make up the qualifier field and contain the command qualifier mnemonic. The last two positions are the index field and contain the index number.

The alarm equipment register is a 16 character register used to contain the equipment name and individual alarm name of the alarm being displayed (in the alarm display mode). The first eight characters are occupied by the equipment name and the last eight by the alarm name.

The result register is a 16 character register used to display the result of a command execution.

Command Entry Procedure

Commands are entered and executed using the first six columns of keys (left to right) shown in Figure 6. A prompting technique is used to lead the operator through the command entry procedure, step by step.

1) To enter a new command, depress the NEW CMND key. This will clear any previous entries in the command entry register and flash the prompting message "SCROLL" in character positions 1-6 of the 16 character display. Scrolling is accomplished by repetitive depressions of the SCRL↑ or SCRL↓ keys. Repetitive depression of these keys will cause the entire command list to be displayed, one command at a time, in the first seven character positions of the display. The two scrolling keys cause the list of commands to be sequenced in opposite directions.
2) When the desired command is present in the display simply depress the ENTR key to choose this command and terminate entry into the command field. Immediately the prompting message "SCROLL" will be flashed in character positions 9-14 of the display. This prompts the user to enter the qualifier part of the command next. This is done by repetitive depressions of the SCRL keys. Only those qualifiers that are valid with the displayed command will be shown. When the desired qualifier is present the ENTR key is depressed to choose the displayed qualifier and terminate entry into this field. Immediately the prompting message "??" will be flashed in character positions 15-16 of the display.
3) The flashing message "??" indicates that entry into the index field is now required. This is accomplished by a one or two digit numeric entry. When the desired index number is present in the display the operator depresses the ENTR key to choose the index and complete the command entry.
4) If there is no site/CHNL entry displayed in the four digit readout the operator will be prompted to enter a site number by two flashing question marks in the site field.

Entry into this field is by a one or two digit numeric entry. When the site number is correct the operator depresses ENTR.

5) The user is now prompted to make entries into the channel field by two flashing question marks in the CHNL field. Entry is the same as site number and terminated with the ENTR key.
6) After the channel entry is complete both displays will be static, signaling the operator that the command entry is complete and may be executed.
7) Execution is initiated by depressing the EXCT key. Inquiry commands (see Table 2) will be executed immediately and the result displayed in the 16 character display.
8) If the command is an action type, execution is not immediate. The message "CONFIRM COMMAND" appears in the display. The operator may re-examine the command by depressing the ALT DSPY key (toggles display between command entry register and result register). When he is convinced of the correctness of the entered command, he accomplishes final execution by depressing the CNFM key. This will causes command execution and any results to be displayed.

The above describes the normal command entry procedure. Exceptions and extensions are described below.

1) If a mistake is made during command entry, depress the NEW CMND key to clear all previous entries into the command entry register and start over. The message "SCROLL" will be flashing in the display. If a site and channel are displayed they will not be affected.
2) If an error is made in site or channel entry depress NEW SITE and both fields will be cleared. Two flashing question marks will appear in the site field and site and channels can be re-entered. The command entry register is unaffected.
3) If a new command is entered when a site and channel number from a previous entry are displayed, the operator will not be prompted to enter a new site and channel.
4) If a new site and channel are entered when a previous command is displayed, the operator will not be prompted to enter a new command.
5) As described before, if a new command is entered with no previous site/CHNL displayed, the operator will be prompted to enter site and channel information.
6) If a new site and channel number are entered with no previous command displayed the operator will be prompted to enter a new command.
7) The ALT DSPY key simply toggles the display between the result register and the command entry register. Repetitive depressions cause the display to alternate back and forth between the two.
8) Depression of the AUTO key will cause the system to revert back to the automatic alarm reporting mode and will also clear the command entry register and the command site and channel number.
9) The CLR key will clear the command entry register and command site/channel number and result in "SCROLL" flashing in the display.
10) Not all commands require an index entry. In these cases the operator will not be prompted to make this entry.

Displaying Alarms

Alarms are displayed using the last two columns of keys (left to right) in

Figure 6. During normal operation the MACS is left in the automatic alarm reporting mode by depressing the AUTO key. In this mode, the display is blank unless an alarm exists somewhere within the system. If alarms exist, the highest priority alarm will automatically be reported on the front panel, the appropriate alarm type indicator (LED) will be illuminated and the appropriate office alarm actuators will be activated.

If only a single alarm exists it is, be definition, the highest priority alarm, so it will be outputted to the display. If more than one alarm exists simultaneously, then the highest priority is determined by the following rules:

1) All major alarms have higher priority than minor alarms.
2) Within an alarm group, major or minor, lower site numbers have higher priority than higher site numbers.
3) Within a site the common equipment has highest priority followed by the signal channels with lowest numeric channels having highest priority. Finally, the external equipment channel has the lowest priority.
4) Within a channel the priority of each equipment group is determined by its proximity to the receiver. The closest equipment has highest priority.
5) Within an equipment group each individual alarm priority is based upon its proximity to the receive side of that piece of equipment. Alarms closest to the receive side have highest priority.

Whenever multiple alarms exist, the multiple indicator will be illuminated and the leading digits of the field containing multiple alarms will flash on and off. If the user wishes to view the non-displayed alarms, he can use the four field keys labeled SITE, CHNL, EQPM and ALARM in conjunction with the SCRL↑ and SCRL↓ keys to sequence through all active alarms.

1) By depressing SITE followed by either of the SCRL keys, the next highest or next lowest priority site *in alarm* will be displayed. The highest priority alarm within the new site will always be shown with the new site number. The SCRL keys may be used repetitively to sequence through all sites in alarm. When the end of the list is reached the display will "wrap around" to the other end of the list.
2) By depressing CHNL followed by repetitive depressions of the SCRL keys one can sequence through the channels in alarm at a give site (the one displayed in the site field when CHNL was depressed). The order of sequencing is again by priority. The action is essentially equivalent to that discussed in 1 above except that the SCRL keys operate on the channel field instead of the site field.
3) By depressing EQPM followed by repetitive depressions of the SCRL keys one can sequence through the equipment groups in alarm within a given site and channel (those displayed in the site and channel fields when EQPM is depressed).
4) By depressing ALARM followed by repetitive depressions of the SCRL keys one can sequence through the individual alarms at a given site, channel and equipment group.

The foregoing describes the method of sequencing through the alarms that may be present within the monitored system. Any of the four field keys, SITE, CHNL, EQPM, ALRM may be depressed at any time to enable the SCRL keys for sequencing in that field. The field keys need not be depressed in the order shown above.

The AUTO key may be depressed at any time. This will return the system to the automatic reporting mode and the highest priority alarm will again be displayed.

The MACS we have been discussing is a low-level, data-gathering system. The microprocessor collects far more information than can be shown on the small front panel display unit provided at every terminal.

The system is designed to be upgraded by adding a CRT terminal to display the status of alarm points in plain language, when this is necessary. Permanent records then can be made by a high speed printer. A keyboard will be provided to enter software diagnostic routines into the system. Greater storage capacity can then be provided at each site so that long term records of each alarm occurrence would be available. These features will greatly assist in trouble shooting and maintaining large digital radio systems.

The reason most of these features are included only as options is largely economic. Large scale integration has made computational and data processing power available at a very low cost. However, peripheral devices like high speed printers etc. are quite expensive. Also, the cost of software relative to hardware continues to grow.

Definitions of Computer and EDP Terms

Just as in any other speciality, the people involved in designing and manufacturing data processing equipment and software have developed their own terminology. Some of these terms are defined in the following list. The definitions are not rigorous and the list is far from complete.

ALPHANUMERIC – Having characters only. A CRT that is an alphanumeric CRT displays only characters and cannot draw lines etc. for graphical pictures. The characters displayed are usually those specified by ASCII.

ASCII – Acronym for American Standard Code for Information Interchange. Specifies a character set and the binary representation of these characters.

ASSEMBLER – Program that translates assembly language to machine language.

ASSEMBLY LANGUAGE – Symbolic codes representing CPU instructions.

BASIC – Beginners All Purpose Symbolic Instruction Code. The most common high level language for personal computers.

BAUD – A unit of signaling speed derived from the duration of the shortest signaling event. If each signaling event is exactly one bit, then the baud rate is the same as bits per second.

BINARY – A numbering system based on powers of 2 (uses only 0's and 1's).

BIT – One binary digit.

BOOTSTRAP – An instruction set usually in ROM or PROM that is executed automatically when the system is turned on, to initialize the system.

BUS – The very high speed communication path between the components of the computer in the mainframe.

BYTE – A group of 8 bits. Usually one byte is used to represent each character in ASCII code.

COMPILER – A program to translate from higher level language such as FORTRAN or Pascal to machine language.

CORE – A type of primary memory using magnetic cores – also used loosely to mean any primary memory.

CPU – Central Processing Unit. The unit that interprets and executes the instructions.

CRT – Cathode ray tube (TV type tube).

DOS – Disk Operating System – A sophisticated monitor.

DYNAMIC MEMORY – RAM memory that has to be refreshed every few milliseconds to retain its memory. Usually the refresh is done with hardware on the memory board rather than from the CPU.

EDP – Electronic Data Processing.

EPROM – Primary memory that can be erased by special methods.

FIRMWARE – Program built into the system (usually in ROM or PROM).

HARD COPY – A copy of information on paper (as from a printer).

HARDWARE – The physical components of the system.

HEXIDECIMAL – A numbering system based on powers of 16.

HIGH LEVEL LANGUAGE – A programming language that is independent of the type of CPU so that it should work on all computers that can support it. Also known as POL, Procedure Oriented Language.

INTERFACE – The hardware to control the flow of information between a peripheral device (such as a printer) and the CPU.

I/O – Input/output of information to or from the computer.

K – May mean either 1000 or 1024 (2^{10}).

MAINFRAME – The box that holds the CPU, the front panel, the power supply, and the bus.

MONITOR – 1) A CRT that uses direct video input. 2) The lowest level system program for controlling the computer, used for example to load your applications programs.

OCTAL – A numbering system based on powers of 8.

PARALLEL I/O – Transmission of more than one bit (usually 8) simultaneously with multiple transmission lines. This type is frequently used with printers.

PASCAL – A high level language.

PL/M – A high level language which was developed especially for microprocessors.

POLLING – Requesting station to transmit data.

PROM – ROM that can be filled with instructions by special methods, after the device is manufactured.

PROTOCOL – A set of rules for the formatting and relative timing between communicating processes. Also referred to as Control Procedure or Line Discipline.

RAM – Random Access Memory that can be both written on and read from.

RANDOM ACCESS – Access to stored information by jumping to an address determined by a system directory. Random access is faster than sequential access because one does not have to sequentially read all previous data on the unit.

ROM – Read only memory. Memory with permanent instructions that cannot be erased. This type is used for permanent programs, such as bootstraps, to be used with the systems. Programming is done during manufacture.

SELECTING – Requesting another station to receive data.

SEQUENTIAL ACCESS – Access to data by sequentially reading all data from the beginning of the unit to the point of interest.

SERIAL – Sequential transmission of bits over a single line rather than parallel transmission. Most common type for transmission of information to peripherals such as CRT monitors, slow or medium speed printers, and communication sys-

tem modems. The RS-232C standard is the most common serial type.

SOFTWARE – All programs including systems or applications programs.

STATIC MEMORY – Memory that holds its memory as long as power is on and doesn't need refreshing like dynamic memory.

BIBLIOGRAPHY

1. Scientific American, *Microelectronics*, Scientific American September, 1977. (Also available in book form).

2. GTE Lenkurt Demodulator, *Introduction to Microprocessors* February, 1977.

3. Thomas, Owen F., *Software Aid for Firmware Production*, Interface Age March, 1978.

4. Osborne, Adam, *An Introduction to Microcomputers*, Vol. 0, The Beginners Book, Vol. 1, Basic Concepts, Adam Osborne and Associates, Berkeley, CA.

5. McNamara, John E., *Technical Aspects of Data Communication*, Digital Equipment Corporation, Bedford. MA.

6. Karp, Harry R., *Basics of Data Communications*, McGraw-Hill, N.Y., N.Y.

7. Sippl, Charles J., *Microcomputer Dictionary and Guide*, Matrix Publishers Inc., Champagne, IL.

GTE LENKURT
1105 COUNTY ROAD
SAN CARLOS, CALIFORNIA 94070

ADDRESS CORRECTION REQUESTED

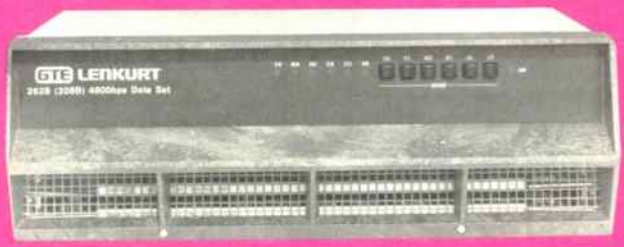
Bulk Rate
U. S. Postage
PAID
San Carlos, CA
Permit No. 37

PIC00319C015
B J PICOU

1

319 CRAWFORD ST
LAFAYETTE LA 70501

GTE Lenkurt's New, Low-Cost 262B (208B) Data Set



- Transmits and receives synchronous 4800 bps data over the switched (DDD) telephone network or privately owned transmission facilities.
- End-to-end compatible and electrically interchangeable with the Western Electric 208B Data Set.
- Registered under FCC part 68 for direct connection to the DDD network, without protective arrangements.
- UL listed under UL-478.

GTE LENKURT

VIDEO, VOICE & DATA
TRANSMISSION SYSTEMS

The GTE Lenkurt Demodulator is circulated bimonthly to selected technicians, engineers and managers employed by companies or government agencies who use and operate communications systems, and to educational institutions. Permission to reprint granted on request.