S-100-4
COMPUTER ASSEMBLY
AND
OPERATING INSTRUCTIONS

DATA SCIENCE, INC.

DEALER:
U.S. MICRO SALES
15392 Assembly Lane
Huntington Beach, CA 92649
(714) 898-1492
CUSTOMER SERVICE:
(714) 898-5525

TABLE OF CONTENTS

TABLE OF CONTENTS (Con't)

TABLE OF CONTENTS (Con't)

FORWARD

This manual is divided into four chapters and a section called MICRO NOTES for your convenience.

Chaper One, INTRODUCTION, is a overview of computers, their basic operation and what you can expect it to do for you and what you can expect it not to do for you.

Chaper Two, INSPECTION, contains the initial unpacking, inspection and testing instructions. The contents of this chapter is intended to help get you up and running.

Chapter Three, GENERAL OPERATION PROCEDURES, consists of technical information on systems hardware and software.

Chapter Four, contains TROUBLE SHOOTING AND RETURN MATERIAL AUTHORIZTION (RMA) procedures for repairs.

MICRO NOTES is the section on configuring various printers, terminals, and disk drives to run with our system.

A CP/M Manual is included, separately.

Included with each system are two disks "A" and "B" which contain the CP/M operating system and disk utilities.

PLEASE READ THIS MANUAL BEFORE USING YOUR COMPUTER. YOU SHOULD BE FAMILIAR WITH THE CONTENTS OF THIS MANUAL BEFORE TRYING TO USE YOUR COMPUTER SYSTEM. THE SECTIONS WITH AN "*" SHOULD BE COMPLETELY UNDERSTOOD BEFORE ATTEMPTING ANY USE OF YOUR COMPUTER.

Some of the utility programs will only read upper case characters so please use upper case, only, to avoid confusion.

Quotation marks ,"", as used in this manual are intended only to draw attention to a particular piece of information. They don't have to be typed as part of a command line or drive designation.

Whenever you change a switch setting, change a baud rate, or anything else that may affect operation of your computer, terminal, or printer, turn them off and back on again. These devices read this information when turned on.

BACKUP PROCEDURES *** IMPORTANT TO DO THIS FIRST THING ***

Place your system disk "A" in A drive, insert a blank disk in "B" drive; type "backup". This is an automatic program which will give you an exact duplicate of your system disk, ie, copy "A:" to "B:". After the first copy is completed, you may replace the disk in "A" with the "B" master, and place a blank disk in "B" drive, type return and an exact copy of "B" will be made.

After these backup disks are made, remove your master A and B disks to a safe place. At present the Backup submit data files are for 8 in drives. Mini's can be backed up by using the DFOCO DCOPY program sequence or the PIP program.

US MICRO SALES

CHAPTER ONE

INTRODUCTION

INTRODUCTION

The difference between a computer user and a computer professional is similar to the difference between a first year automobile driver and a stock car racer: although both operate similar machines, one does so with style and efficiency, knowing the strengths and weaknesses of the device and maximizing its efficiency while the other simply drives from one place to another. Since you have invested in a U.S. Micro Sales computer system, we want to help you derive the maximum power from you machine by discussing its potential and its drawbacks and advising you on proper operating procedures.

Regardless of your experience with computers, you'll find that the value of your computer system will be greatly enhanced if you take the time to study this overview.

WHAT'S IN THIS BOX?

FUNCTIONAL DEFINITION

A computer is a tool with which you impose order on a set of data through the implementation of preinstalled intelligence. Now that we've defined the computer, let's define the definition.

A computer is a machine that reads to and writes from memory, nothing more. A computer operator can put information into or take information out of that memory, nothing more. The key to successful operation of a computer lies in the nature of the information you put in and take out of the computer. For example, you may put into memory a system of filing names and addresses that say: If a name begins with "A", put it at the top of the list; if a name begins with "B", put it after the names that begin with "A" and so on. Once the file system is installed in memory, you may enter a list of names and instruct the computer to order that list of names according to the system you previously installed. Your file is then stored in memory alphabetically. When you want to extract those names, they will be given to you in alphabetical order.

Although the system of entering commands and data is a little more complex than just described, all you have actually done is write to and read from the computer's memory. However, order, (alphabetical) has been imposed on a set of data (names) by implementing preinstalled intelligence (the file system).

An important feature of current computer technology is the fact that it is not necessary for the computer user to create the intelligence system he needs to order his data. Prepackaged application programs that are the fruit of many years of other people's labor are widely available, and can save the user many hours of work while allowing him the ability to keep his data in a form that is efficient and universally legible. Why is it necessary for data to be understandable to those other than the collector of that data?

Consider the dilemma of Izzy Tyke, sole proprietor of His Owne Bicycle Shoppe. Izzy has been doing quite well in business for the past twelve years and would like to buy the storefront next door in order to expand his business. He hires an accountant to go over his financial records to see if he has enough money to buy the extra space. The accountant comes to the store and Izzy shows him the financial records--a huge cardboard box stuffed full of invoices, check stubs, receipts, and assorted other material, including a lunch pail that Izzy misplaced in 1964. The cardboard box is a very convenient method of recordkeeping for Izzy, but the accountant refuses the assignment of looking over the records.

Izzy then decides it would be easier to borrow the money than to sort out the material in the box, so he goes to the local bank. The bank refuses Izzy the loan on the grounds that the condition of his financial records do not inspire confidence in Izzy's character or his ability of repay the loan. So Izzy loses the opportunity to buy the storefront.

Izzy is convinced that he must keep his records in a more formal manner, and considers several methods of doing so. He could go to night school and take a course in bookkeeping, but the late nights and homework assignments may affect his salesmanship during the day; besides, there's no guarantee that Izzy will be a good bookkeeper even if he passes the course. He could hire a bookkeeper--but can he find a good one and does he have enough money for that?

If Izzy buys a computer and some accounting software, he has at his disposal all the procedures and formality necessary to keep his books in perfect order. The computer will guide him through the steps involved in maintaining records--all Izzy has to do is plug in the numbers the computer asks him for.

Through the use of the computer, the bicycle shop owner's book are in perfect order without the necessity of months of study or the expense of an accountant's time. The computer has imposed order on Izzy's finances using the preinstalled intelligence of the accounting software package.

The useful applications of a computer's ability to read and write from memory are wide and varied. Any task that is made up of step by step, orderly processes and that is of a receptive nature will be completed more quickly and accurately through the use of a computer than it would be by a human being. However, there are projects which are actually impeded when a computer is applied to the job.

MISUSES OF A COMPUTER

Any task that is not done more than once, and any task which is subjective rather than objective would not benefit from the use of a computer.

A personal computer enthusiast, trying to justify the expense of a personal computer to his wife, often waxes eloquent on the ease a computer will bring to the kitchen. "You can keep all your recipes on computer" he'll explain. "So convenient."

"They're perfectly convenient in the recipe file, and I can prop an index card up on the back of the counter, which I couldn't do with a computer," she'll reply.

"But you could print out the recipe when you wanted to use it, and besides, what about converting the amounts of each ingredient--the computer will do that for you when you want to increase the recipe for a party."

"We never have parties," will be the retort. "Besides, if I have a recipe that calls for one egg for four servings and I want the dish to serve ten people, won't the computer just tell me to add two and a half eggs? Where does that get me?"

The woman will win the argument with three salient points. The purchase of a computer or the development of a software program to do a task that must be done only rarely is a waste of effort and money, since the time involved in amassing the equipment will frequently exceed the time involved in doing the task itself. Secondly, the portabilty and convenience of her present file system exceeds the portability and convenience the computer can give her. Finally, the act of cooking is a subjective experience; the proportions of ingredients depend upon one's personal taste, the freshness of the ingredients and so many factors that a computer's value in accurately estimating the proportions is almost null.

A computer should also not be used as a crutch or scapegoat. Your machine is a finely tuned instrument and will occasionally go out of tune. Be prepared: Back up your data, have alternate methods of dealing with emergencies, be patient when your computer needs repairs. Remember that A) you are accomplishing much more with the help of this tool than you ever could before and, B) that you only get out of a computer what you put into it. A computer only very rarely makes mistakes, a computer operator errs much more frequently.

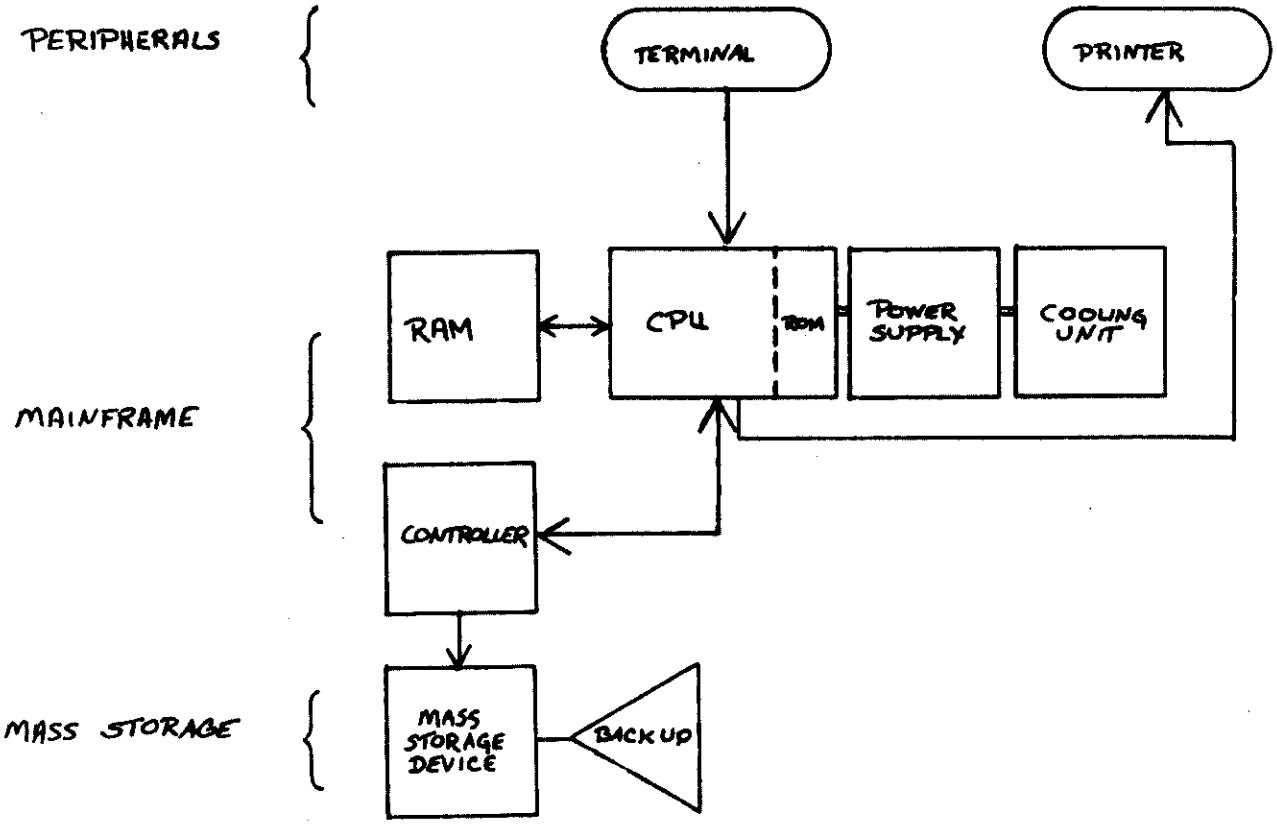
PHYSICAL DEFINITION

THE HARDWARE (Illustrated on page 4)

As we mentioned earlier, a computer is a machine that reads and writes from memory. The component that makes this possible is the CENTRAL PROCESSING UNIT (CPU). The CPU is the most active element in your system, directing and pacing the flow of information to and from various other parts of the system and participating in the functions of all parts of the system.

The information that the CPU handles is located in SYSTEM MEMORY, which can be designed in a number of way: READ ONLY MEMORY (ROM), PROGRAMMABLE READ ONLY MEMORY (PROM), which can be written to initially, and only read thereafter, ERASABLE PROGRAMMABLE READ ONLY MEMORY (EPROM), which can be erased and written to under special conditions, and RANDOM ACCESS MEMORY (RAM), which can be written to and read from at will.

BASIC COMPUTER HARDWARE



Information stored in ROM is the most easily accessible to the CPU, PROM and EPROM are less quickly accessible, while RAM is least accessible. The cost of system memory is directly correlated to the speed of access. The speed and expense of all memory is relative, however; RAM can be accessed by the CPU in a time span measured in nanoseconds (1/1,000,000,000 second), and can cost several hundred dollars for enough space to store a small file. Most computer systems, ours included, use ROM for a small portion of information (see software section) and RAM for the bulk of the information needed by the CPU.

If all the information in a computer was stored in ROM or RAM, the cost of the system would far outweigh any efficiency considerations, making the computer virtually unusable. Therefore, only the information currently in use by the CPU is stored in RAM. Other information is located in MASS STORAGE.

The most common types of mass storage in use today are FLOPPY DISK, HARD DISK, and MAGNETIC TAPE STORAGE. Floppy disks are small, inexpensive circles of magnetic material that can hold the equivalent of a filing cabinet of information. Hard disks are larger, inflexible units that can hold several libraries' worth of information. The size and capacity of magnetic tape is variable.

Each system of mass storage incorporates a mechanism to locate the required piece of information and make it available to the CPU. The ease and speed with which the information in mass storage is accessible is a function of the locating mechanism and the type of mass storage. Information stored on magnetic tape is least accessible because the tape must wind to the appropriate space in which the information is stored. Information on floppy disk is quickly available because of the size of the disk and the mechanism which rotates the disk. Hard disk storage is the optimal medium between the accessibility of information on floppy disk and the flexible capacity of tape.

The mechanical devices, called DRIVES, that wind the tape or revolve the disk, locate the information and make it available to the processor are extremely slow in comparison to the speed at which the CPU operates. When the CPU requires information that is located in mass storage, it is ready to process that information within nanoseconds. It may take several seconds, however, for the mechanical device that controls the storage unit to locate the information. Therefore, an intermediary between the mass storage device and the processor is needed: this intermediary is the CONTROLLER.

The controller is activated by the CPU whenever information from mass storage is needed. It in turn notifies the storage mechanism to locate the information, receives it from storage and notifies the CPU that the information is available for processing. Under such conditions the CPU is free to function on other tasks while the information is being retrieved, and the likelihood of the CPU picking up inaccurate information in its haste to process is eliminated.

Information can be transmitted from an outside source, such as a computer operator or another computer through the use of an INPUT DEVICE. A MODEM is a device that allows input from one electronic device to another.

A TERMINAL allows communication between a person and a computer. A computer terminal looks like a typewriter with a video screen attached, but the similarity between them ends with appearance. A COMPUTER TERMINAL IS NOT A TYPEWRITER. A computer terminal is more flexible than a typewriter, allowing you to correct mistakes, insert words, and change the locations of entire paragraphs before you send the information to memory. However, a missed or inaccurate stroke on a typewriter keyboard results in a misspelling on a sheet of paper; an error on a terminal, if transmitted to memory, can result in a command to the CPU that will seriously affect your data. The terminal is an integral part of the entire computer system, and must be treated with care and respect.

Occasionally it will be necessary for you to make the information in your computer available to people who do not have access to your system. Since it is inadvisable to carry your computer around with you in the trunk of your car or to entrust it to the United States Postal Service, a means of liberating the required data from the system is desirable. The most common method of doing so is through a printer. A printer takes information from the computer and prints it on paper. The paper can then be taken off the machine and carried around in a convenient briefcase or sent through the mail.

PERIPHERALS such as terminal and printers are hooked up to the central processing unit through COMMUNICATIONS or INTERFACE PORTS. These ports act as translators between the CPU ad the peripheral, making sure that the signals given out by the CPU are understandable to the peripheral and vise versa.

Since the computer is run by means of electricity, the smooth and even transmission of electrical power through the system is important. The POWER SUPPLY located in your computer takes the electrical impulses from your wall outlet and monitors their even transmission through the system. The COOLING UNIT in your system maintains a steady temperature inside the system for efficient operation.

In order of importance, the essential parts of your computer system are: The CPU, the system memory, the controller, mass storage, power supplies and cooling units, and input/output peripherals. The order in which these elements are likely to give you trouble is exactly the reverse; the more mechanical devices fail long before the electronic devices will.

THE SOFTWARE

You can think of your computer hardware as different parts of a mechanical brain and of the software as the processes that utlilize the physical elements. Just as there are several layers of human intelligence, there are layers--intelligence levels--of software in your computer system. These levels are identified as the system monitor, the operating system, the high level language

and the application package.

The most elementary level of software is the SYSTEM MONITOR. Like the subconscious portion of your intelligence that is constantly monitoring your physical condition and surroundings, the computer's system monitor, located in ROM, surveys the different elements of the system and verifies that everything is in working order. It also serves as a link between the hardware and the next layer of software: the operating system.

Just as common experience allows people to communicate with one another by giving them a universal method of understanding, the OPERATING SYSTEM establishes standardized protocols of operation for computer systems. Having a common means of operation enables your system to operate with software that has been developed on other computers. U. S. Micro Sales supplies CP/M, operating systems for use with its computers. These systems, developed by Digital Research in Pacific Grove CA, are in common use; most application packages are accessible to these operating systems.

The next layer of intelligence is the HIGH LEVEL LANGUAGE, which utilizes the methods of operation set up by the operating system to form a direct method of communication between the computer and the user. It applies definition and language to the processes of the computer system. When the operator communicates to a computer in high level language, he or she can direct the processes of the system.

The translation between computer language and English, French, or another human language occurs in the APPLICATION PACKAGE. The application package contains instructions to the computer that will enable it to complete a specific task and guides the user through his part in transferring information to the system.

SYSTEM STRUCTURE

At U. S. Micro Sales we use different combinations of hardware and software to configure a system that will meet the specific needs of the user without waste or limitation.

A SINGLE-USER SYSTEM generally consists of one CPU, two to four K(kilobytes) of ROM that contain a system monitor, 64K RAM for system memory, two to four interface ports enabling you to connect peripherals to your system, and a CP/M operating system.

The mass storage requirements of a single-user system are usually met through the use of floppy disks. While these elements suffice for the needs of the average user, systems frequently provide extra space to accommodate additional memory and interface capability.

Since U. S. Micro Sales' CPU can process the commands of a single-user with time left over, it is possible for more than one operator to utilize a single CPU. In a MULTI-USER SYSTEM, several input terminals share the processing time of the CPU, system memory, mass storage, and any peripherals available.

Sharing is made possible through the use of an MP/M operating system, which divides the system memory and processing time between work stations. In order to make the distribution as efficient as possible, a multi-user system is broken into divisions of ports, users, and tasks. There may be four ports of entry on the computer system, meaning there are four terminals hooked up to the system. Although only four people can operate with the system at any given time, there may be sixteen users who maintain files on the system. When an operator sits down at a terminal, he or she will gain access to a specific file by asking for it by user number. For example, a company's purchasing records may be filed under user number 12. A user can sit down at any terminal and ask for the user 12 files when he needs information from the purchasing records.

In a common multi-user system the number of tasks that can be performed by the computer at any given time corresponds to the divisions of memory (memory banks) in use at the time. If a printer is printing a file while an operator is entering another file, the computer is processing two tasks.

Under certain conditions a standard multi-user system is not sufficient to meet the needs of the users. Some application packages, for example, take up more than a reasonable share of CPU time. Unequal CPU requirements slow down the processor in its round from task to task; and short processes may be lengthened by waiting for the CPU. Occasionally, too, it may be efficient for the system to be multi-tasking--not to share processing time but to process tasks simultaneously. In such environments a NETWORK system is suitable.

A networking system differs from standard multi-user configuration in that each work station has its own processor, system memory, and interface ports. These individual computer systems work independently of one another but are linked to a central station (host) so that, although each station may have its own files and storage capacity, they may share data and expensive options such as printers and mass storage devices. These communication links are made through both hardware (cables) and software.

A primary benefit of a networked system of hardware and software is the fact that an existing system can be expanded to include more work stations or peripheral devices without major revisions to either current hardware or software. The computer system is as flexible as possible.

USING YOUR COMPUTER

Before you even take the computer out of the packing material, look over the manual provided for your system. The manual will instruct you in setting up your system for operation. Make sure that the trip from our plant to yours did not jar loose any connections in the system. TURN ON THE COMPUTER.

Once power is supplied to all parts of your computer system, your system monitor will immediately begin diagnostic checks of the computer system. Your terminal will display the sign on message and the symbol F800, which indicates that your system

memory is in good order. Any message other than F800 indicates that a memory error has occured.

Your next contribution to the successful operation of your computer is to insert the disk containing the system into the drive. On your system disk are the instructions necessary for your computer to function.

The disk's label should be on the right hand side of the disk as it enters the drive when using Tandon drives, to the left hand side of the disk as it enters the drive when using other drives, and facing up when using mini drives. The notches in the disk always go to the back.

When using 8" disks you must use a write protect tab in order to write information onto the disk. When using 5" mini disks the reverse is true. You must remove the write protect tab in order to write onto the disk. Close the drive enclosure latch. Press the RETURN (or NEW LINE) key on your terminal.

Press down and hold the CONTROL key on the terminal and strike the letter C in order to bring up (boot) the operating system from the floppy disk. If you wish to boot the operating system from a location other than the floppy disk, a prompt other than CONTROL C is necessary; consult the operating manual for that location.

Watch the video screen for messages from the operating system. If an error message is displayed on the screen, your system is not ready for operation; STOP. It may simply be that you inserted the disk in backwards. Correct the problem and retype CONTROL C (or whatever brings up your particular medium).

When the operating system is in good working order, you will receive a prompt message, meaning "go ahead". Consult the manual for your particular operating system to determine the correct prompt and further instructions.

USAGE CONSIDERATIONS

Since you have invested a considerable amount of time and money in this computer system, you will want to protect that investment from inadvertent depreciation. The most effective way to do this is by ensuring that your data is not forever destroyed by an interruption in power or through an inconsiderate user on a multi-user system resetting the central system. There are obviously a limited number of ways in which you can prevent city wide blackouts or discourtesy; we suggest you BACK UP YOUR DATA i.e., make copies of your computer files.

When and how should you back up your files? It depends upon your circumstances and the value of your data: in general, you should make a copy of your file whenever it would take longer to recreate the file than to back it up or whenever the monetary value of your information (either intrinsic or in time spent) reaches or is increased by $100 or more. If your environment is such that you are likely to be interruped often, however, you may wish to back you data more frequently.

There are two basic methods of backing up data, the streaming file and the file managed methods. In a streaming file backup, all the information in any given system is automatically copied at a scheduled time. The file managed method gives each user the resonsibility of making back up copies of his or her own data. Each has its advantages and disadvantages; your choice will depend on your requirements.

We have attempted to give you some brief tips on the most effective use of your computer system, but true proficiency will come with careful study of your manual and "hands on" experience with your system. So, go to it, admire the efficiency of your system and enjoy the freedom computerization gives you.

Application Package: A unit of software that instructs the computer in the processes needed for a particular task.

Central Processing Unit(CPU): The element of the computer system that processes information between the other system components.

Communication Port: See interface port.

Controller: The component which mediates between the CPU and the mass storage device.

Cooling Unit: A device, usually a fan, which maintains a steady temperature inside the computer.

Drives: Mechanical devices that locate information in mass storage units.

EPROM: Eraseable Programmable Read Only Memory.

Floppy Disk: An inexpensive type of mass storage.

Hard Disk: A type of mass storage that has great capacity.

High Level Language: A method of communication between the user and the computer system.

Host: The central work station or main system of a multi-user computer system.

Input: As a verb, the act of transmitting information to the computer system; as a noun, the information placed in the computer system.

Input Device: The component of the system used for transmitting information to the computer.

Interface Ports: The elements of a system which allow the user to attach peripheral devices to the system.

Kilobyte: A unit of measurement (1000 bytes) for computer memory.

Magnetic Tape: A type of mass storage.

Mass Storage: A method, or device, for storing large amounts of information in a system.

Memory Bank: In a multi-user system, the part of system memory reserved for a specific user.

Modem: An input device for transmitting information from one electronic or mechanical device to another.

Nanosecond: A unit of time measurement: 1/1,000,000,000 second.

Operating System: A software subsystem.

Output: As a verb, the act of transmitting information from the computer system to a peripheral; as a noun, the information extracted from the computer.

Output Device: The component of the system that receives the information from the computer.

Peripheral: Element of the computer system that is not specifically necessary to the operation of the CPU.

Power Supply: Device which transmits electrical power to the computer system.

Preinstalled Intelligence: A set of instructions for the completion of a task that is filed in the computer's memory.

PROM: Programmable Read Only Memory

RAM: Random Access Memory

ROM Read Only Memory

System Monitor: The most basic software subsystem.

Terminal: See Input Device.

CHAPTER TWO

INSPECTION

UNPACKING

We have shipped your S100-4 in a special carton with what, we feel, is adequate foam packing surrounding all 4 sides. If after you receive your computer you should note any physical damage to the carton or any internal damage to the computer itself please notify the carrier, as US MICRO SALES is not responsible for shipping damage. In the case of United Parcel Service, you may call the local UPS office to file a damage claim. UPS will send a representative to survey the damage. UPS will then settle the claim. Please save the carton and all packing materials in case you ever need to return the computer.

INSPECTION

You may inspect the insides of the computer by laying it down on either side and removing the six screws on the bottom which hold the case to the main frame. Gently spread the sides of the cover and slip it off the base plate. You may now inspect for any internal shipping damage, or loose cables or boards that may have jarred loose in shipping.

DISKETTES

SOFT SECTORED IDENTIFICATION

SINGLE SIDED

The S100-4 uses standard 8 in. SOFT SECTORED diskettes. If diskettes are not labeled to tell you what type it is the following will help you indentify them.

The small hole near the center is an access hole for drive read electronics it is located at 1 O'clock on the disk (single sided disks). If the diskette is rotated in the jacket by gently grasping the film between two fingers and turning 360 degrees a smaller hole can be seen on the diskette. If there are many small holes, you have a HARD sectored diskette and it WILL NOT OPERATE with the S100-4.

DOUBLE SIDED

The second type of diskette is double sided. If you have a double sided system you must have a double sided diskette in order to read and write to both sides. A single sided diskette will work but only one side will be utilized.

The access hole will be at approximately 2 O'clock, (down to the right more with the label facing you). A single sided drive cannot read a double sided diskette since data is store alternately on the sides. One side is not filled and then the other.

DISK DRIVE

Your drives have been shipped with internal cardboard head protectors, you must remove these inserts before applying power. If your drives are made by Tandon, (slim line with the closing lever at the top) the disks are to be inserted with the labels facing to the right and the notches to the back. This is different from other models used in XOR systems in the past. If your drives are made by Mitsubishi, the disks are inserted with the labels to the left.

If your S100-4 has the Mini 5 in drives, A drive is on top, and the disks are inserted with the labels up. All other functions are the same as 8 in.

DRIVE NOTES (DISK CAPACITY WITH DIFFERENT FORMATS USING "DFOCO" TO FORMAT DISKS)

STANDARD 8" DRIVES

FORMAT B:
Standard 3740 IBM single density/single side 128 bytes 26 sectors per track.
Capacity is 241K.

DFORMAT B:
128 bytes per sector 51 sectors per track, double density, single sided.
Capacity is 474K.

DFORMAT B: SIZE 512
512 bytes per sector 16 sectors per track, double density, single sided.
Capacity is 596K.

DFORMAT B: SIZE 512 DBL
512 bytes per sector 16 sectors per track, double density, double sided.
Capacity is 1212K

MINI DRIVES

DFORMAT C: MINI 48

128 bytes per sector, 30 sectors per track, double density, single sided.
Capacity is 169K.

DFORMAT C: MINI 48 SIZE 512

512 bytes per sector, 10 sectors per track, double density, single sided.
Capacity is 181K.

DFORMAT C: MINI 48 SIZE 512 DBL

512 bytes per sector, 10 sectors per track, double density, double sided.
Capacity is 362K.

DFORMAT C: MINI 96 SIZE 512

512 bytes per sector, 10 sectors per track, double density, single sided.
Capacity is 366K.

DFORMAT C: MINI 96 SIZE 512 DBL

512 bytes per sector, 10 sectors per track, double density, double sided.
Capacity is 750K.

TERMINAL

The CPU is shipped with the "B" terminal port set to 9600 baud. You must remove the cover of your system in order to change the baud rate unless you have software to do it, or your terminal has the capability to set different baud rates. See your terminal manual for this information. The baud rate for this port is the lower 4 switches of the 8 position dip switch located in the lower right corner of the CPU board, (switches 5,6,7,and 8.) Note, for 9600 SW 5 is on 6,7,8, are off. The CPU will support operation at 19,200 baud, if your terminal will, you may set the CPU to 19200 by setting all 4 switches to the off or left position. (See the section on baud rates for complete selection.)

The terminal is connected to the "B" port, which is the lower of the two serial ports on the back of the computer. For most terminals The only pins that need to be connected in the terminal cable are 2, 3, and 7. If you have a standard RS-232 cable with pins 1-8 and 20 connected you may have to disconnect pins. We have noticed some difficulty with different terminals when more than 2,3,and 7 are used.

INITIAL POWER ON

The on/off switch is located at the rear of the computer on the right hand side. Upon initial power on, a sign on message should appear on the terminal screen and a HEX number should appear, F800, this denotes that all power on and memory checks are functioning normally. If any other number appears, please refer to CHAPTER 5, for trouble shooting procedures. At this time you may also perform some monitor routines to verify the operation of the monitor. The drive light on "A" will light up (the drive on the left) and the drive motor will start up. If you are not familiar with computers and do not understand the section MONITOR COMMANDS you may skip it at this time.

MONITOR COMMANDS

DUMP

D (DUMP) - The dump command will accept 4 hex bytes, jump to the next field, accept 4 more hex bytes and then display all of the memory locations between those two addresses. If you should wish to enter the numbers without leading zeros, you may do so by hitting a carriage return. For example, you could dump location 2 to location 8 in the computer by typing D2<cr> followed by 8<cr>.

LOAD

L (LOAD) - The load command will accept one 4 byte address and then display the contents of that memory location on the screen. The prompt character will allow you to replace what is in that location of memory with the data that you type onto the screen. If you merely want to look at system memory, you can enter a carriage return or a series of carriage returns. The memory locations will not be modified, but will only be displayed on the screen in sequential manner.

FILL

F (FILL) - The "FILL" command accepts a 4 byte (starting) address, moves to the next field, accepts a second 4 byte (ending) address, moves to the next field and accepts a 2 byte set of data. At this point, the "FILL" command will automatically be executed, filling all system memory between the starting address and the ending address with the data character set.

MOVE

M (MOVE) - The "MOVE" command operates in the same manner as the "FILL" command with the exception that the 4 byte addresses are, respectively, the address of the source data and the address of the memory location to which the source data is to be moved. The 2 bytes which are entered next indicate the size (number of bytes in hex) of the block of memory to be moved.

VIEW

V (VIEW) - The "VIEW" command upon receiving a 4 byte starting memory location will display sixteen lines of ASCII data (640 bytes) on the CRT screen. Typing carriage returns following the initial display causes the next sequential 640 byte blocks to appear.

G (GO)

G (GO) Enter destination address.

H (HEX)

H (HEX) Hex String Locate, enter starting address, ending address string to locate.

X (EXAMINE)

X (EXAMINE BANK) For selecting a certain bank to write to in multi user systems.

DISK DRIVE BOOT UP

FLOPPY SYSTEMS

Insert system disk "A" in drive A, the drive on the left, with the label facing to the right if Tandon drive, to the left if Mitsubishi, close the door, reset and type a control "C", (when typing a "control C" the control key and the C key must both be held down). The system should boot up to an A>. At this time type "D" the screen should display a directory of the utilities on disk "A".

Insert disk "B" in drive B and type a "B:", then "D" a directory of disk B should display on the screen. If either disk does not boot or display these preliminary directories, please try the following: Reset the computer, open the door and type a control C. You should hear the drive trying to load the head 5 times, and then the error message "drive not ready" should appear on your terminal screen, this denotes that the drive would probably load if the door were closed. Close the door and try to boot, if the error message "no system" comes on the screen this means that there is no valid system on the system tracks and your computer will not boot.

Other errors that may appear are "10" which means read ID not found, check your disk format or try to validate the disk, it may not be compatible with the system or be a defective disk. 08, which is not a fatal error, and denotes not enough time to load the head, try to reboot. CRC not found generally means a totally bad disk. For further trouble shooting refer to Chapter Four. If your S100-4 has 5 1/4 in mini drives they will be booted up in the same manner, except that the label is face up. Drive A is normally the upper drive.

HARD DISK SYSTEMS

When using a hard disk system you boot with a Control "A" "B" or "C" to boot from the hard disk, (depending on the size of your system), or a control "F" to boot from the floppy that is part of the hard disk system.

DISK UTILITIES

The "A" disk contains 32 utility programs, the programs on this disk are the tools and utilities of CP/M (e.g. PIP, DDT, ED, ASM, SUBMIT). The 42 programs on the "B" disk are the subroutines, assembly files, and the equate tables with which the operating systems is generated and/or modified. Also contained on the "B" disk are the Help files, a useful program for the first time computer user. A more detailed description of the files on the disks can be found in this manual under "EXPLANATION OF DISK FILES". Please note that if you plan to be doing any custom modification of the BIOS, you will require a MACRO assembler, since our BIOS has been assembled with MACRO.COM. There are MACRO assemblers available from a number of sources.

FORMATTING WITH DFOCO

The program Dfoco is the XOR fast copy, validate, and formatting program. Any command you wish to function double density must be preceded by a "D", ie, dformat, dcopy, dvalidate. The default is to single density IBM 128 format. There is a set of instructions on the screen after you type dfoco. The most common use of DFOCO is for formatting new disks. We will review that command line here.

NOTE: MAKE SURE THAT YOU PUT A COLON ":" AFTER THE DRIVE NAME EXACTLY AS THE EXAMPLES BELOW INDICATE OR YOU WILL FORMAT YOUR MASTER DISK IN "A" DRIVE AND ERASE ALL FILES ON THE DISK, OR YOU WILL GET AN ERROR MESSAGE OR ? FROM CP/M IF INSTRUCTIONS ARE NOT FOLLOWED EXACTLY IN DCOPY OR DVALID.

As an extra precaution while formatting a disk it is wise to remove your original disk from A drive after giving the formatting command line below. The program is loaded into memory so there is no need for the original disk to be in the drive.

Type DFORM B: SIZE 512. This will format the disk in drive "B" 512 byte sectors, single sided, double density.
DFORM B: SIZE 512 DBL will format "B" double sided.

COPYING WITH DFOCO

Dcopy is the copy program, and will copy track-for-track "A" to "B", including the system. To make a copy of "A" to "B",

Type DCOPY A: TO B: SIZE 512 for single sided disks
DCOPY A: TO B: SIZE 512 DBL, for double sided copy.

This will copy all information from "A" to "B".

For mini drives the format command line might be

DFORM B: SIZE 512 MINI48, OR DFORM B: SIZE 512 MINI96
or
DFORM B: SIZE 512 MINI96 DBL for double sided.

If you get the error message "FORMATTING ERROR DRIVE B", check to see if your disk is write protected, inserted incorrectly, or the wrong type of disk.

VALIDATING WITH DFOCO

This portion of DFOCO checks to see that the recording surface of a disk is completely intact before using the disk. Validating a disk consists of reading each sector on a disk and verifying that the crc is correct.

The command lines follow the same format as the lines used in copy and format.

CHAPTER THREE

GENERAL OPERATING PROCEDURES

S100 POWER SUPPLY

The S100 power supply consists of a power transformer, two bridge rectifiers, one 52,000 ufd capacitor and two 10,000 ufd capacitors. The specifictions of the bus voltages are plus 11 volts, plus 16, minus 16. The power is supplied to the bus via black, red, white, and yellow wires, please refer to internal view for wire locations. The bus sub assembly AC is connected to the main AC at connector located the right rear of the bus sub assembly, refer to diagram for voltage check point.

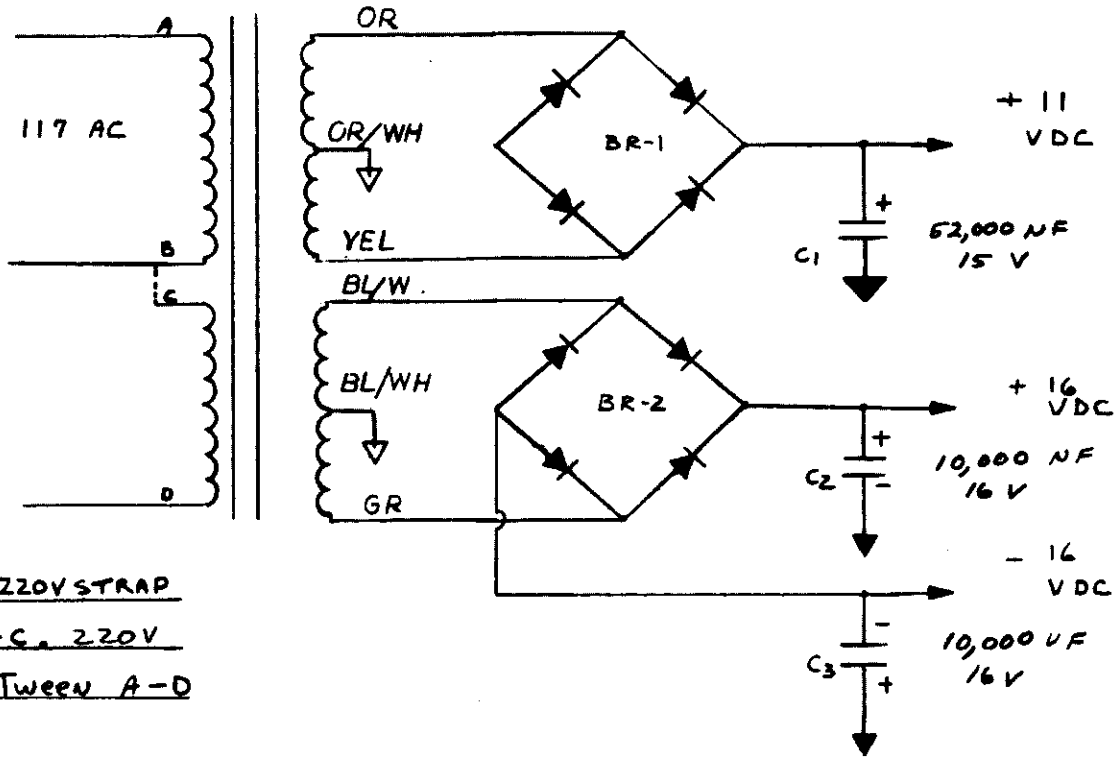
S-100-4

PARTS LIST BUS SUPPLY

T-1	TRANSFORMER	I-8000-35
BR-1,BR-2	DIODE BRIDGE	I-3001-06
C-1	CAP LYTIC 52,000 ufd 15V	I-3001-42
C-2.C-3	CAP LYTIC 10,000 ufd 16V	I-3001-10
PCB	PRINTED BOARD	I-3001-31

10-1028-B

T-1



FOR 220V STRAP

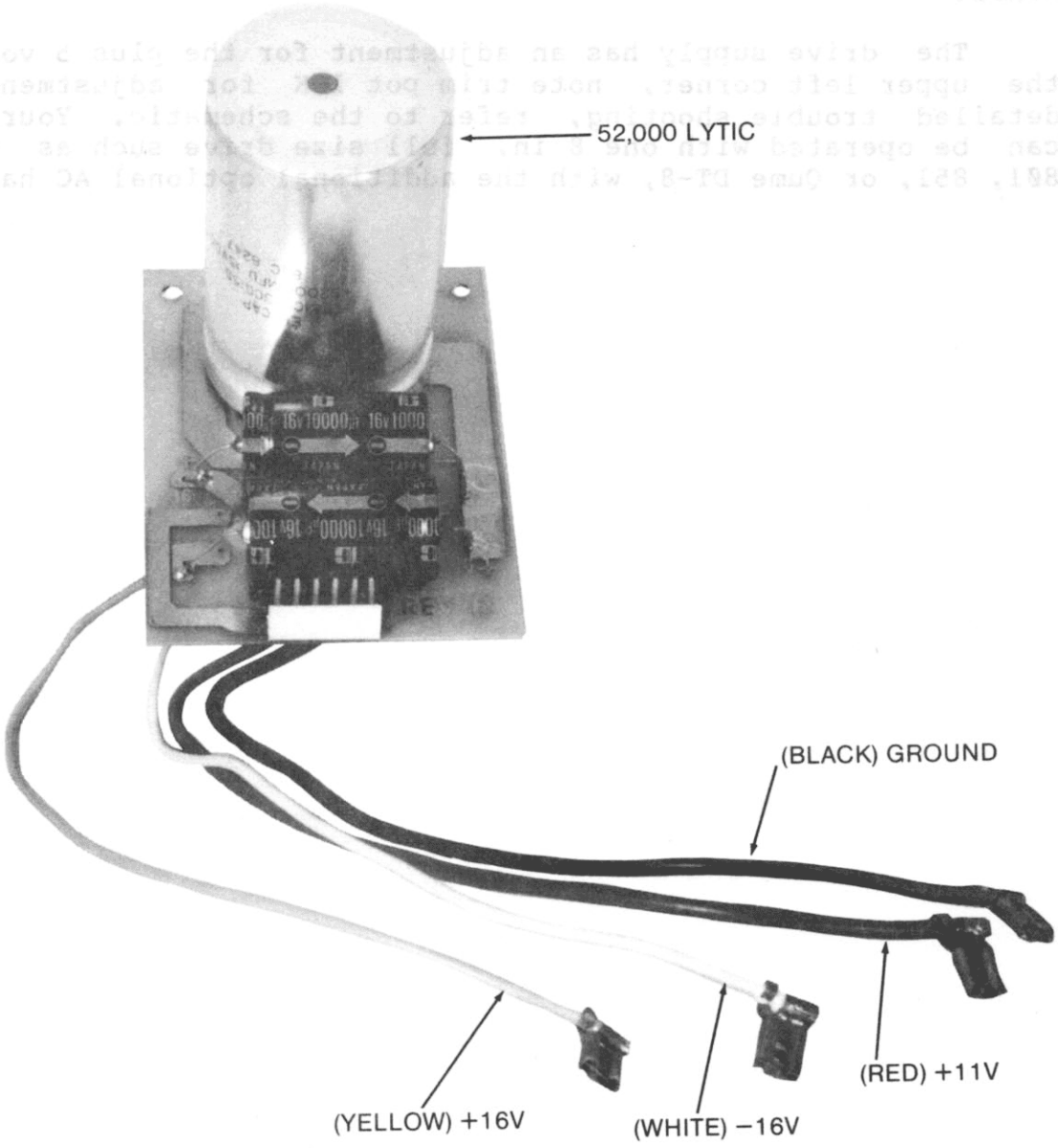
B-C, 220V

BETWEEN A-D

SCALE:	APPROVED BY:	DRAWN BY PWS
DATE: 2/25/83		REVISED

S-100-4 BUS SUPPLY

DRAWING NUMBER



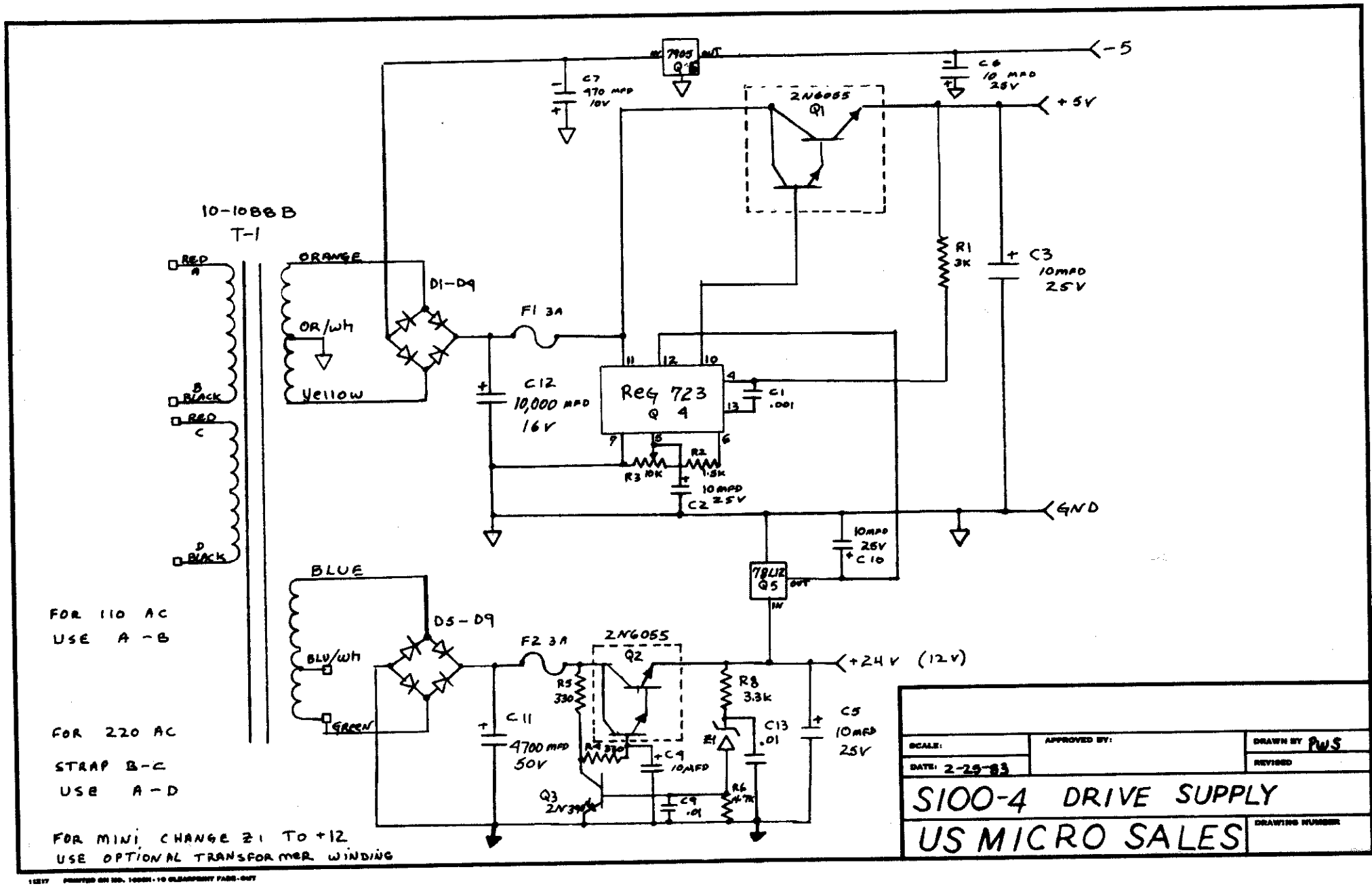
S-100-4 BUS POWER SUPPLY

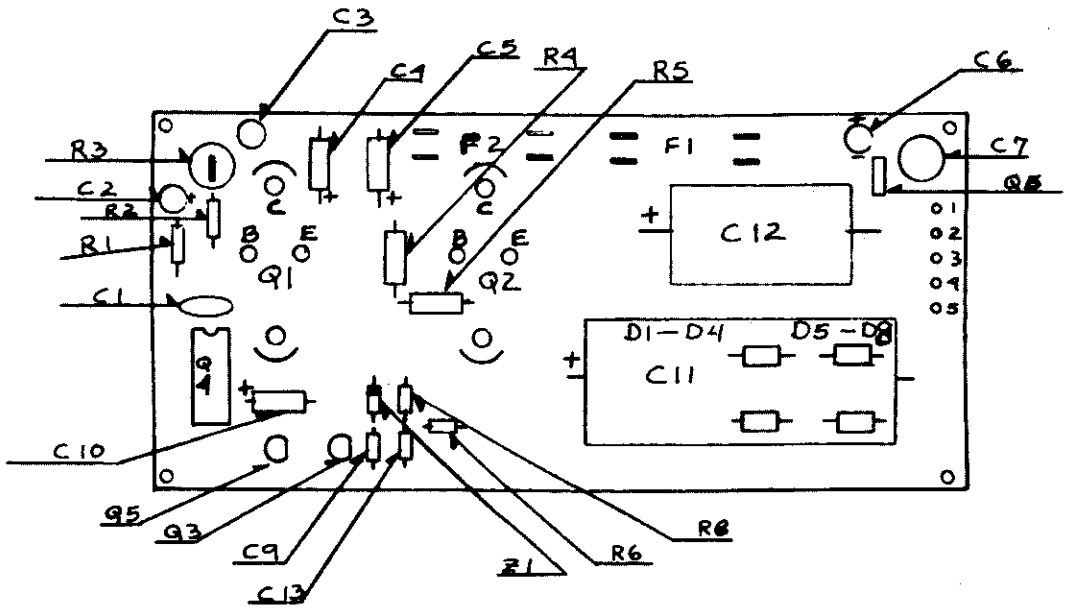
DRIVE POWER SUPPLY

Drive power is supplied by the circuit board mounted on the back panel, refer to the internal view. The power supply supplies plus 24 at 4 amps, and plus 5 at 3 amps. There is also an optional minus 5 at one amp, which is not required for the Tandon slim line drives.

If your S100-4 has the MINI drive option, the power supply will be jumpered to provide plus 12 volts at 4 amps, for the Mini drives. There will also be a different drive DC cable for the Minis.

The drive supply has an adjustment for the plus 5 volts in the upper left corner, note trim pot 10K for adjustment. For detailed trouble shooting, refer to the schematic. Your S100-4 can be operated with one 8 in. full size drive such as Shugart 801, 851, or Qume DT-8, with the additional optional AC harness.


10-1088 B
T-1
RED A
BLACK B
RED C
BLACK D
ORANGE
OR/wh
Yellow
D1-D9
F1 3A
C12
10,000 MFD
16V
Reg 723
Q 4
C1
.001
R1
3K
C3
10MFD
25V
2N6055
Q1
C7
470 MFD
10V
C6
10 MFD
25V
+5V
-5
R3 10K
R2
1.5K
C2 10MFD 25V
GND
10MFD
25V
C10
78L12
Q5
BLUE
BLU/wh
GREEN
D5-D9
F2 3A
2N6055
Q2
+24V (12V)
R5
330
R8
3.3k
C11
4700 MFD
50V
C13
.01
C5
10MFD
25V
C4
10MFD
Q3
2N3904
C9
.01
R6
4.7K
FOR 110 AC
USE A-B
FOR 220 AC
STRAP B-C
USE A-D
FOR MINI CHANGE Z1 TO +12
USE OPTIONAL TRANSFORMER WINDING
SCALE:
APPROVED BY:
DRAWN BY PWS
DATE: 2-25-83
REVISED
S100-4 DRIVE SUPPLY
US MICRO SALES
DRAWING NUMBER



SCALE: 1:1	APPROVED BY:	DRAWN BY PWS
DATE: 2-24-83		REVISED

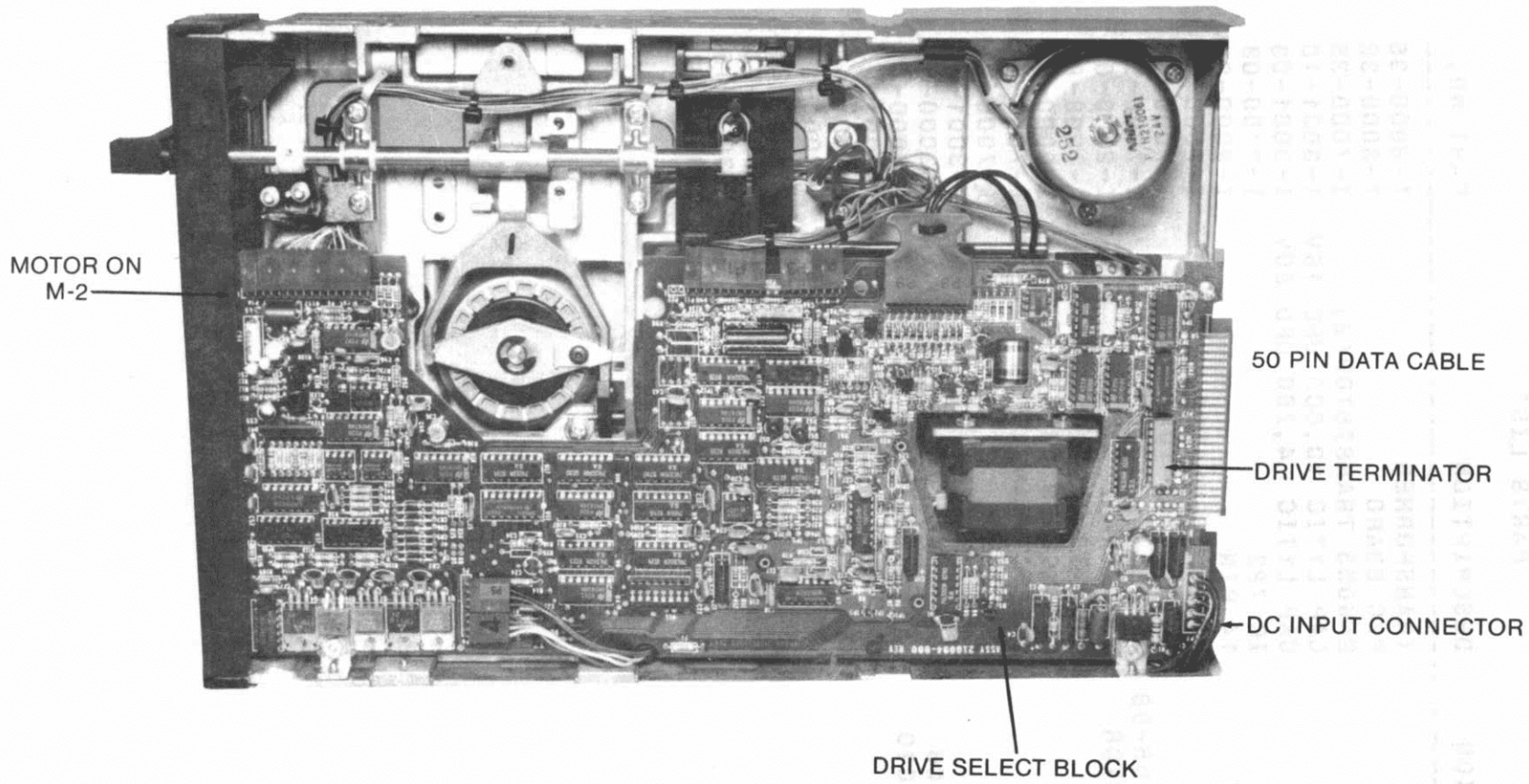
DRIVESUPPLY PARTS LAYOUT

US MICRO SALES

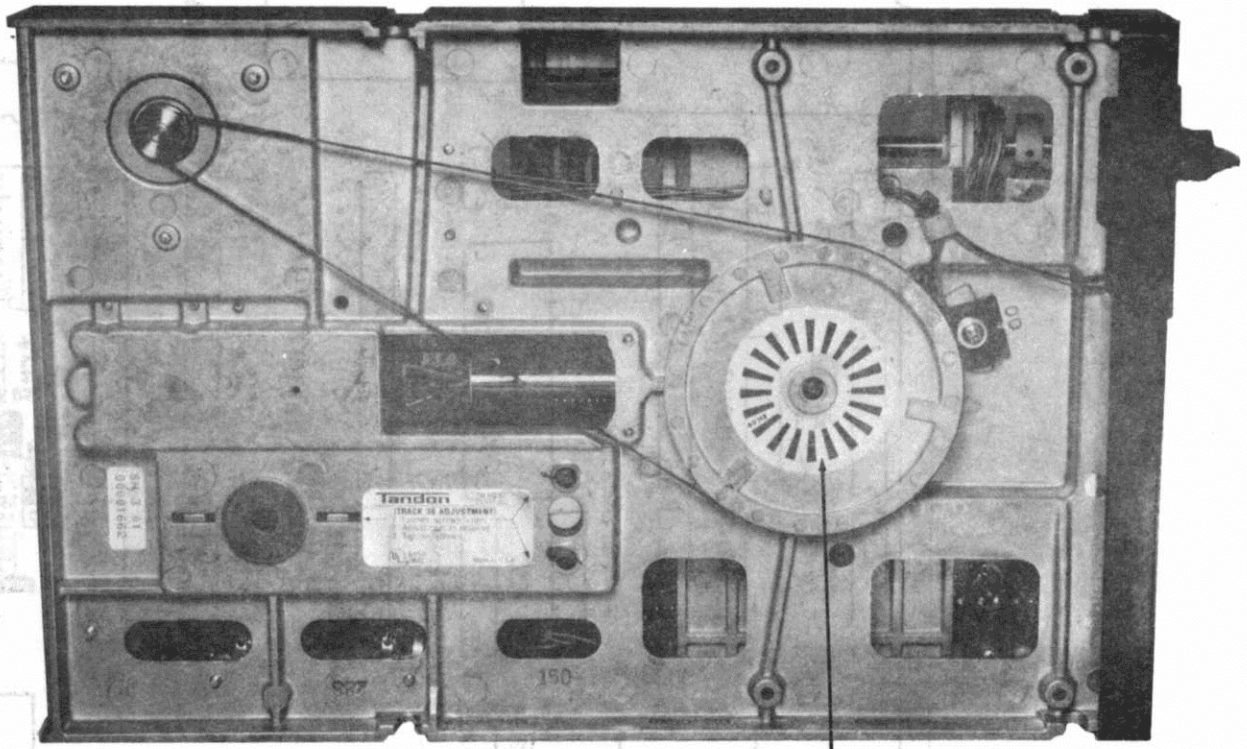
DRAWING NUMBER

PARTS LIST

LOCATION	DESCRIPTION	PART NO.
T1	TRANSFORMER	I-8000-35
PCB	P/C BOARD	I-3000-32
Q1,Q2	2N6055 TRANSISTOR(2)	I-7000-35
C12	CAP LYTIC 10,000 UFD 16V	I-3001-10
C11	CAP LYTIC 4,700 UFD 50V	I-3001-03
Q4	IC 723	I-7400-03
SOCKET	14 PIN	I-2000-05
Q5	78L12 REGULATOR	I-7000-13
Q6	7905 REGULATOR	I-7000-04
D1-D4,D5-D8	DIODE	I-7000-50
CONNECTOR	5 PIN MOLEX	I-2000-83
F1,F2	FUSE 3 AMP (2)	I-3000-80
	FUSE CLIPS (4)	I-0000-04
Q3	2N3904 TRANSISTOR	I-7000-27
Z1	24V ZENER	I-7000-21
R3	TRIM POT 10K	I-3001-47
C2,C3,C6	CAP LYTIC 10MF 25V R	I-3000-63
C4,C5,C10	CAP LYTIC 10UF 25V A	I-3000-41
C1	CAP DISK .001UF	I-3001-24
C9,C13	CAP DISK .01	I-3001-19
C7	CAP 470 MF 10V R	I-3000-87
R4,R5	RESISTOR 330 OHM 1 WATT	I-3001-29
R2	RESISTOR 1.4K OHM 1/4 W	I-3001-14
R1,R8	RESISTOR 3K OHM	I-3001-15
R6	RESISTOR 4.7	I-3001-01

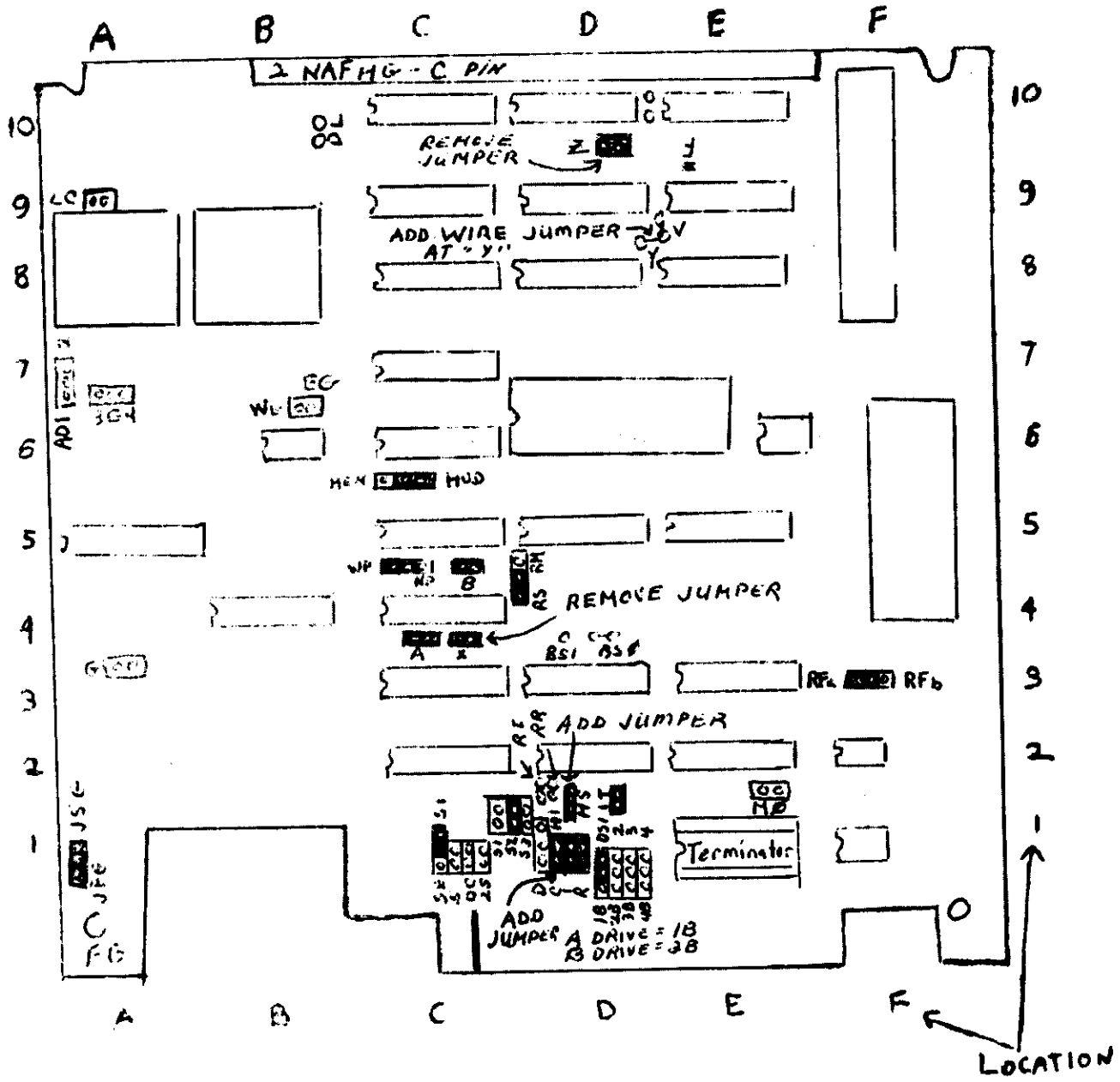


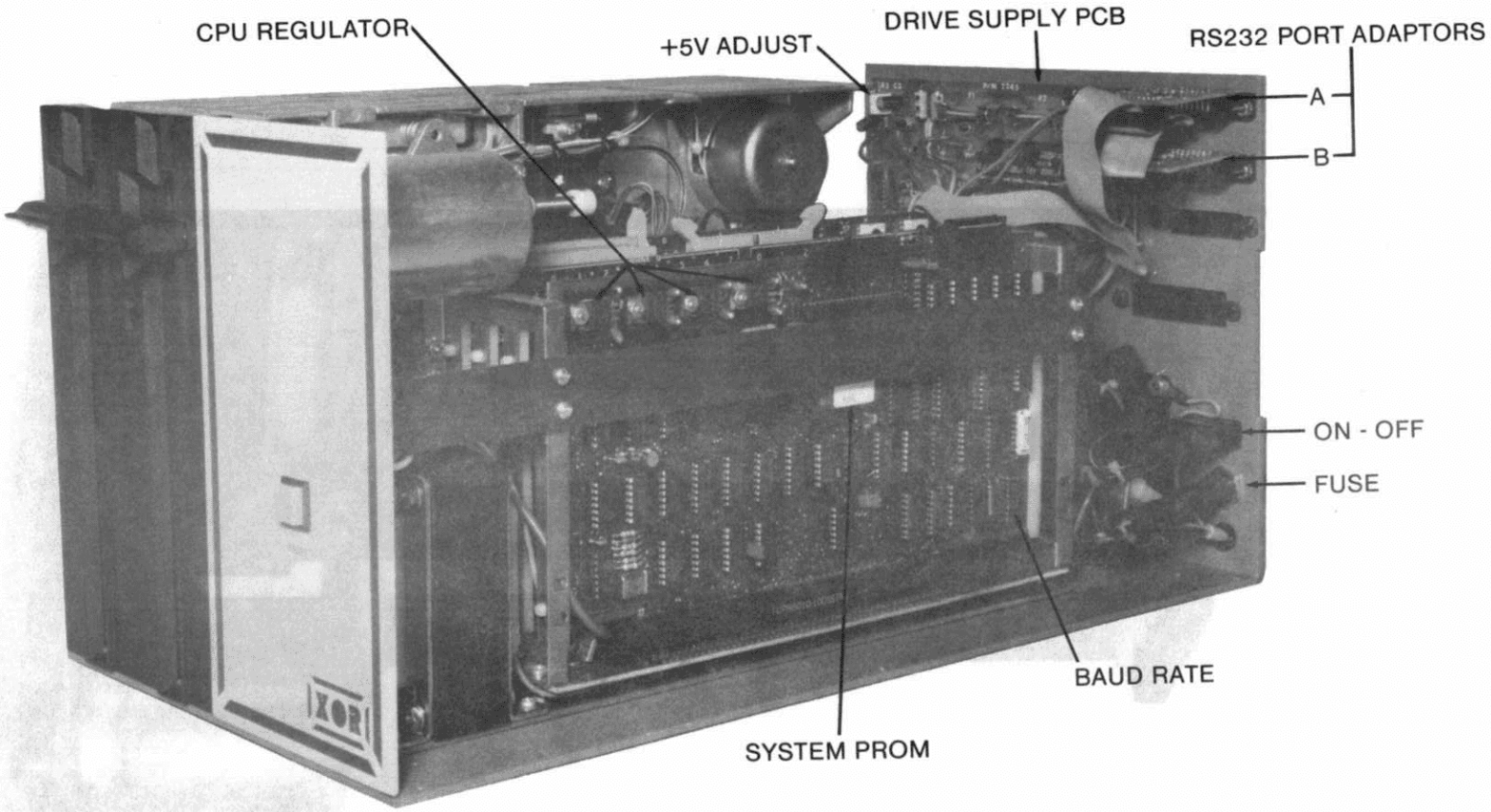
TANDON DRIVE SIDE VIEW



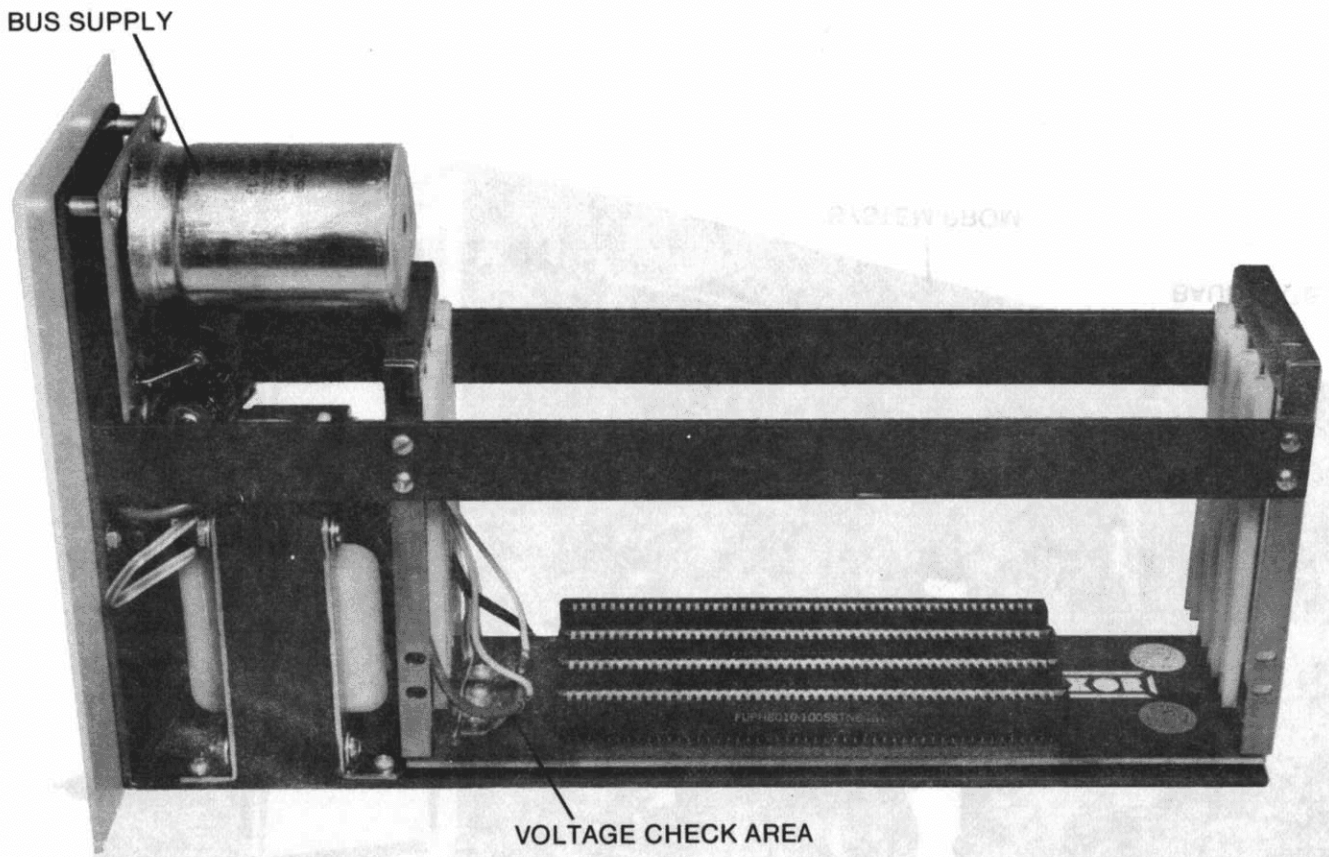
TANDON DRIVE SIDE VIEW

MITSUBISHI JUMPERING





S-100-4 INTERNAL VIEW



S-100-4 4 SLOT SUB-ASSY

BOARD REMOVAL

Refer to internal view for location of the three computer circuit boards. Boards may be removed by gently pulling up on them evenly, be sure do turn off power before removing any of the boards. When installing boards, make sure they are evenly seated in the bus connectors.

INTERNAL CABLES

Refer to internal view for location of the internal cables. Note the location of the optional internal parallel cable. It can be installed in location noted. The "A" paddle card is located to the rear of the computer, top card, to enable optional printer jumpers to be installed without removing the card.

DRIVE JUMPERS

If your S100-4 has the Tandon thin line drives, refer to Tandon Drive circuit board picture for jumper locations. You will note the location for drive select, optional m-2 jumper motor on from drive select, and motor speed adjust. No other jumpers are required.

MITSUBISHI JUMPERING (8" THIN LINE)

See drawing in this chapter on a previous page for jumper locations.

BAUD RATE SELECT

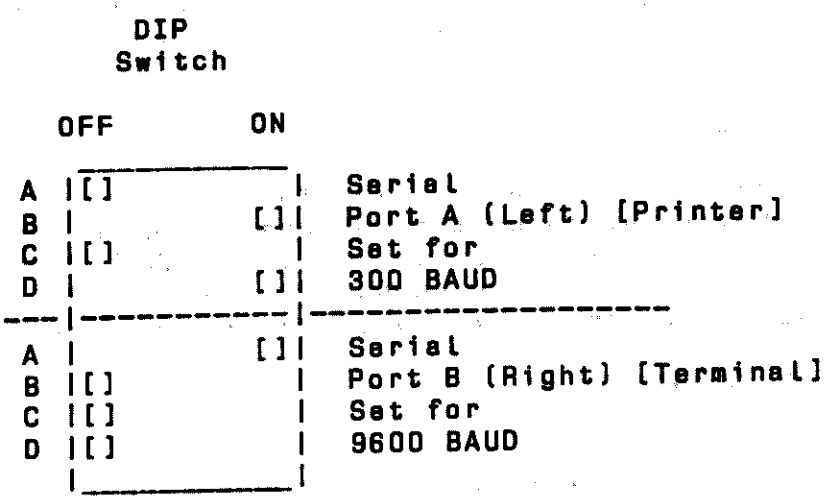
Refer to internal view for the location of the BAUD Rate switch on the CPU board. As shipped from the factory both ports are set at 9600 Baud.

SERIAL I/O BAUD RATE SELECT

The 8 position DIP switch at the lower right hand corner of the board is broken into two 4 bit sections. The upper 4 bits select the IO baud for the left 8251 and the lower 4 bits select the baud for the right. We connect the left 8251 to the "A" paddle card and the right one to the "B" paddle card. The paddle cards are mounted to the rear panel and make strapping adjustments easier. The "A" port is committed in the CP/M bios software to the printer, and the "B" port to the CRT Terminal.

BAUD RATE SWITCH SETTINGS CPU BOARD

The switches are used in a binary pattern to set the rates as follows:



X = O F F O = O N

Baud	D	C	B	A	
50	0	0	0	0	(00H)
75	0	0	0	X	(01H)
110	0	0	X	0	(02H)
134.5	0	0	X	X	(03H)
150	0	X	0	0	(04H)
300	0	X	0	X	(05H)
600	0	X	X	0	(06H)

Baud	D	C	B	A	
1200	0	X	X	X	(07H)
1800	X	0	0	0	(08H)
2000	X	0	0	X	(09H)
2400	X	0	X	0	(0AH)
3600	X	0	X	X	(0BH)
4800	X	X	0	0	(0CH)
7200	X	X	0	X	(0DH)
9600	X	X	X	0	(0EH)
19200	X	X	X	X	(0FH)

N O T E :
Make sure when using the above chart that the positions we show correspond to the type of switch you have on the board. Some models of switches have the "ON" to the opposite side.

XOR DISK UTILITIES "A"

ASM.COM	DI.DAT	M.COM	STAT.COM
BACKUP.COM	DIAGNOSE.COM	M1.COM	SUBMIT.COM
BK.DAT	DSKTYP.COM	M2.COM	SURVEY.COM
CPM61.COM	DSYSGEN.COM	MOVCPM.COM	SWORM.COM
D.COM	ED.COM	PASSCHK.COM	WORM.COM
DDT.COM	ERASE.COM	PIP.COM	XSUB.COM
DDUMP.COM	PRINT.COM	WORM48.COM	
DFOCO.COM	LOAD.COM	PRINTER.COM	

XOR DISK UTILITIES "B"

5MSAFMT.COM	CBASIC2.HLP	DSYSGEN.COM	
ALGOLM.HLP	CBOOT.ASM	EXAMPLE.HLP	PASCAL.HLP
ASM.HLP	CPM.HLP	FLOP.ASM	PASSCHK.COM
BACKUP.COM	CXLST.ASM	FORTRAN.HLP	PIP.COM
BK.DAT	CXLST.DOC	HELP.COM	PRINTER.COM
BLK.ASM	D.COM	HELP.HLP	PRINTER.ASM
BLKEQU61.ASM	DATA.ASM	IO.ASM	PRM185.ASM
BLKSYS61.HEX	DDUMP.COM	M.COM	SURVEY.COM
BLKSYS61.SUB	DFOCO.COM	MAC.HLP	Z80.LIB
C.HLP	DI.DAT	MACRO.LIB	
CBASIC.HLP	DIAGNOSE.COM	MBASIC.HLP	

FILE EXTENTIONS

Files with the extention ".COM" are command files. They are programs that are ready to run by just typing the file name. Example: DFOCO.COM will run on your system by just typing "DFOCO", it is not necessary to type DFOCO.COM.

Files with the extention ".ASM" are files that are the original source files that the programmer writes in assembly language code.

Files with the the extention ".HLP" are explained below.

Files with the extention ".LIB" are library files used with a Macro Assembler.

Files with the extention ".SUB" are submit files.

Files with the extention ".DOC" are document files and contain information to be dumped to a printer or using the TYPE command in CP/M. Example: "TYPE CXLST.DOC"<CR>

NOTE: Files with WD or W prefix are for Western Digital Hard disk Systems.

Some of the files on our system disks are for experienced computer users. If you don't understand a file skip it for now. However, file explanations with an "*" after the name need to be understood by even inexperienced users. If you are an inexperienced user please read these file explanations thoroughly. If after reading them you still can't understand them please call Customer Service at (714) 898-5525, or perhaps have someone with more experience help you.

EXPLANATION OF DISK FILES

BACKUP:COM *

Will help first time users make that initial important backup of their CP/M source disks. BACKUP should be the FIRST program that you run after receiving your computer. The SOURCE disk (the disk you want to copy from) goes in drive "A" and a BLANK destination disk (BLANK disk you want to copy to) goes in drive "B "; Type "BACKUP" BACKUP will format the destination disk and copy A to B. Backup is intended for use with our operating system only. It can be used to back up disks that are single sided, double density, 512 byte sectors only. It can't be used to backup single sided disks to double sided disks as it will not format both sides of the double sided disk. To format and copy double sided, double density, or single sided single density disks use DFOCO and PIP.

DIAGNOSE.COM

An automatic, NON-DESTRUCTIVE system test utility that will test system memory and both drives. It should be run before drive "B" has been accessed. Put a disk in drive B to run this test. The program runs a memory test and then goes into DFOCO and validates the disk in drive B.

The test takes about 5 minutes to run and assures everything is in good shape to run your software. Data is output to a system printer so you need not watch the screen. (Got to keep up with IBM you know!) If you don't have a printer hooked up at this time you can still run this utility but you must watch the screen to get the output. If any error messages are generated refer to the appropriate section of this manual.

HELP.COM

Will aid first users of CP/M, BASIC, and many other run time programs. Simply type HELP CPM <CR> or HELP MBASIC or whatever ???.HLP program libraries appear on your current distribution disks. To see what they might be, type D *.HLP <CR>.

SURVEY.

This program looks over all I.O. ports and reports what is being used. Interesting to use when you want to add something else to the system and want to avoid clashes. (why not ?)

DFOCO:COM *

DFOCO is a fast format and copy routine which allows the user to copy and format in both single and double density and allows different sector sizes to be utilized. (128, or 512 bytes per sector). See Chapter Two for specific instructions.

M, M2

M and M2 are non destructive memory tests which can be left to run for long periods of time on their own. M will not restart from the beginning once it completes a pass, but will continuously run random number test. (See Chapter four in this manual for more detailed instructions.)

WORM, WORM 48

WORM and WORM48 are effective memory tests which runs freely within RAM. They are DESTRUCTIVE memory tests. (See Chapter four in this manual for more detailed instructions.)

DDUMP:

DDUMP is the disk read utility The user can dump a specific track and sector of the diskette as well as perform edit operations.

This program is an improved DUMP utility for CP/M. Any CP/M file can be DUMPED to the screen in a format similiar to that used by the DDT DUMP command. In addition, any sector or group of sectors may be DUMPED in the same format. DDump should be used for single and double sided disks.

OPERATION

The program may be run either by typing DDUMP or DDUMP followed by the file name or track and sector. Note: The format is quite free. Spaces are usually ignored. They are only required after the words track and sector or T and S, and after the word DDUMP>

If DDUMP is typed the program responds with a heading followed by '*' and waits for more input. Operation in this mode is similiar to other utilities like PIP or DDT.

The operation desired may then be typed in as follows:

```
DDUMP FILE.NAM
DDUMP A:FIL.NAM
DDUMPB:FIL.NAM
```

or DDUMP may be typed separately as:

```
DDUMP
*FILE.NAM
*B:FIL.NAM        (The * is a program prompt)
```

The program may also be used to DUMP disk sectors directly, DUMP any CP/M eight sector group.

FILL

Fill will allow you to fill a certain area of a disk with any character you wish. Within DDUMP *, type FILL. The program will prompt you with Drive to fill?, Track?, Starting sector?, ending sector?, and character to fill.

EDIT

A limited editing feature is included in DDUMP, which allows you to change data on the disk. The edit feature works as follows:

Any single sector on either drive may be edited by requesting display of the sector followed by EDIT.

```
DUMP B: TRACK 4 SECTOR 2 EDIT
```

The requested sector will be displayed followed by an edit prompt

```
EDIT-
```

Enter the address of the first BYTE to be changed. The program will respond by typing back the address entered and the present contents of that address. To change the contents of the address enter the new BYTE followed by a carriage return. The program will display the next address and its contents. To stop entering data type a period. The program will redisplay the sector showing the changes made. The edit feature works almost exactly like the S entry in DDT> typing only a carriage return omits entry.

Typing a period merely redisplays the sector from memory; it does not cause it to be written back on the disk. When editing is complete, redisplay the sector by typing a period and type either

```
WRITE          (writes the sector back on disk)
STOP           (stop editing without writing on disk)
```

All edit entries must be made in HEX. Entering non HEX characters results in an error message. The permissable address range is 0000 to 007F. Larger addresses give an error message.

The edit feature should be used with caution since it is possible to edit CP/M to "death" by changing a single byte. One occasional valuable use is to restore files that have been accidently erased. Erasing a file using the ERA command does not erase the data from the disk, but only enteres an E5 into the first Byte of he directory. To restore a file, display the directory, i.e., T 2 S 5. Find the sector containing the name of the file, and change the E5 to 00 and write the sector back onto the disk. This will restore the file if none of the sectors in the file were changed after the file was erased

DSYSGEN: *

DSYSGEN is used to copy a developed system onto the system tracks of a disk (hard disks as well as floppys).

(More information concerning DSYSGEN can be found in the section entitled DSYSGEN in this manual.)

D

D is a directory utility which will print to the console all files on disk in alphabetical order and their file size. At the end of the directory the total number of files and their total size is also printed to the console. Other drives can be referenced for example:

"D"<CR>, shows a directory of currently logged drive.

"D B:"<CR>, shows directory of files on drive B.

"D B:*.ASM"<CR> shows all the .ASM files on drive B.

DSKTYP: *

DSKTYP is a disk identification program. The program asks the user which drive to check, once answered the program will print to the console what type of diskette is in the specified drive. The program asks for the drive to be tested and then replies with the type of diskette in the drive. This is used to see if your disks are compatible with each other for functions such as DCOPY in which both disk formats must be the same.

DDT:

DDT is a utility which has many features one of which allows the user to read .COM and .HEX files into RAM for execution. Under DDT the user may edit, examine, or link .COM or .HEX files or make small routines for test or other purposes. If under DDT you wish to save an edited program, use the CP/M SAVE NN FILENAME.TYPE command. (NN being the number of records to be saved)

(More information concerning DDT can be found in DIGITAL RESEARCH'S "CP/M MANUAL in the DYNAMIC DEBUGGING TOOL (DDT)" Chapter.)

ED:

ED is a source editor for all source files.

(More information concerning ED can be found in DIGITAL RESEARCH'S CP/M MANUAL in the chapter "ED: A CONTEXT EDITOR FOR CP/M DISK SYSTEM".)

ASM:

ASM is a 8080 source code assembler. This utility makes a .SYM .PRN .HEX file from .ASM files.

(More information concerning ASM can be found in DIGITAL RESEARCH'S "CP/M MANUAL in the chapter entitled ASM.)

LOAD:

LOAD takes the .HEX files developed from a source file and makes a .COM for execution of the file.

STAT:

STAT is a utility which will print to the console the size of any file or the amount of storage remaining on the disk. Example:

"STAT"<CR> "STAT B:"<CR>
"STAT *.*"<CR> "STAT B: STAT.COM"<CR>

(More information concerning STAT can be found in DIGITAL RESEARCH'S CP/M MANUAL in the chapter USER'S GUIDE FOR CP/M OWNERS)

PIP: *

PIP is a copy utility which will copy disk to disk from one to all files on the source disk.

Example:

"PIP"<CR>
* (This means you are in the PIP program and is a response from the program.)
"PIP B:=A:PIP.COM"<CR> Copies the file PIP.COM from the disk in drive A to the disk in drive B.

"PIP B:=A:*.*"<CR> Copies all the files from drive A to drive B>

"PIP A:=B:*.*"<CR> Copies all the files from drive B to drive A.

(More information concerning PIP can be found in DIGITAL RESEARCH'S CP/M MANUAL in the chapter entitled USER'S GUIDE FOR CP/M OWNERS.)

SUBMIT:

SUBMIT is a special utility which allows console input simulation so that long tedious steps which have to be repeated quite often can be done by typing a one line command. Example submit file.

```
DDT
IEXAMPLE.HEX
R
GO
SAVE 1 EXAMPLE.COM
```

XSUB:

XSUB extends the power of SUBMIT to include line input to programs as well as the console commands, the XSUB when included must be in the first line of the sub file.

(More information concerning XSUB can be found in DIGITAL RESEARCH'S "CP/M MANUAL in the chapter entitled USER'S GUIDE FOR CP/M OWNERS".)

MACRO.LIB & Z80.LIB

MACRO.LIB & Z80.LIB are the library files for most standard Z80 Macro assemblers.

BLKSYS61.SUB:
     BLKSYS61 is the file used to develop 8" & 5" disk systems
     with BLOCKING/DEBLOCKING MULTI SECTORS.

BLKEQU61.ASM:
     Following is an example of an equate table.

```
          ;EQU
          TITLE 'XOR STANDARD BIOS VS 2.22 02/20/83'

;
;************************ SYSTEM SIZE ***************************
;
;       MSIZE IS THE EFFECTIVE SIZE OF YOUR SYSTEM AFTER ACCOUNTING
;       FOR CBIOS AND THE HOST BUFFERS. THE VALUE COMPUTED IS USED
;       AS THE NUMERIC VALUE 'NN' IN YOUR MOVCPM SEQUENCE:
;
;               E.G.    MOVCPM  NN *
;
          MSIZE   EQU     60
;
;****************************************************************
;****************************************************************
;       SYSTEM GENERATION EQUATES
;
TRUE      EQU     OFFFFH                  ;DEFINE VALUE OF TRUE.
FALSE     EQU     NOT TRUE                ;DEFINE VALUE OF FALSE.
;
;******** FLOPPY RELATIVE EQUATES ********************************
;
ONEFLP    EQU     TRUE                    ;IF ONE FLOPPY
TWOFLP    EQU     FALSE                   ;IF TWO FLOPPY
TREFLP    EQU     FALSE                   ;IF THREE FLOPPY
FORFLP    EQU     FALSE                   ;IF FOUR FLOPPY
;
MAXI8     EQU     TRUE                    ;TRUE IF STANDARD 8" FLOPPY
;
MINI48    EQU     FALSE                   ;TRUE IF MINI DISK DRIVE 48TPI
MINI96    EQU     FALSE                   ;TRUE IF MINI DISK DRIVE 96TPI
;
SEKRATE   EQU     00                      ;00 = 3MS,|01 = 6MS,|02 = 10MS,|
                                           QUME,850 |801       |03 =15MS   |
;                                          MINIS    |         |           |
;
SLIMLINE  EQU     FALSE                   ;TRUE IF HALF HI DRIVES
;
;******** HARD DISK RELATIVE EQUATES *****************************
;
ONEHD     EQU     FALSE                   ;TRUE IF 5,10 MEG
TWOHD     EQU     TRUE                    ;TRUE IF 16,20 MEG
TREHD     EQU     FALSE                   ;TRUE IF 24,30 MEG
FORHD     EQU     FALSE                   ;TRUE IF 40+ MEG
;
HD        EQU     TRUE                    ;TRUE IF WESTERN DIGITAL
;
;******** SYSTEM CONFIGUIRATION EQUATES **************************
;
MMPM      EQU     FALSE                   ;IF MP/M SET TRUE
;
MLDR      EQU     FALSE                   ;IF M/PM LOADER SET TRUE
XMLDR     EQU     FALSE                   ;IF M/PM LOADER W/O BOOT
;
MDISK     EQU     FALSE                   ;TRUE IF MDISK
```

```
;
DUAL      EQU     FALSE                   ;TRUE IF DUAL SPINDLE DRIVE (PERSCI)
;
DBLOCK    EQU     TRUE                    ;TRUE FOR MULTISECTOR OPERATION
F256      EQU     FALSE                   ;TRUE FOR 256 SECTOR FLOP
F512      EQU     TRUE                    ;TRUE FOR 512 SECTOR FLOP
MAXHST    EQU     512                     ;SIZE OF BLOCKING BUFFER
;
SIDE2BIT  EQU     FALSE                   ;SIDE TWO COMPARE BIT (NORMALLY OFF)
;
```

IO.ASM:
IO is the basic INPUT/OUTPUT routine used within the BIOS.

FLOP.ASM:
FLOP is the driver used for all the floppy drives.

BLK.ASM:
BLK is the BLOCKING/DEBLOCKING MULI SECTOR routine.

DATA.ASM:
DATA is the common data file which contains all of the necessary tables for the operation of the BIOS.

CBOOT.ASM:
CBOOT is the cold boot loader for all systems.

MDISK

MDISK is now with us. - WHAT IS MDISK you ask? It is a kind of cache memory system that is CP/M oriented. It takes the place of the "D" drive and is as large as your pocket book. For all intents and purposes it is a disk drive. Only it stores it's data in RAM. It can be expanded in 48K increments to about 500K. If you do a lot of assembler or compiler work and can think of several hundred better things to do than sit waiting for the processor to complete its task, this may be useful to you.

Typical performance increases over our floppy times will be in excess of 100 per cent. (That means you will only have to sit half as long.) Those of you who are familiar with our system will realize that as we are already the fastest around, this is some real improvment. Comparing it to an Apple or an Orange is a joke.

MDISK will assemble a 20K assembly source program producing a 30K .PRN file and associated .HEX and .SYM files, also reading in two 20K MACRO LIB files in 21 seconds ! Our Apple took 3 1/2 minutes and our Plastic Shack 2 min and 14 secs to do similar tasks. Unfortunetly the Orange was in peels by the time we got to it so no testing could be done. Loading a large .COM file like MBASIC (28k) takes .68 sec. Actually hard to measure. It's fun impressing your friends with it. See MDISK in Appendix of this manual for modfication information. Call your nearest U. S. Micro Sales office for more information on this option.

One of the more common functions you will be performing is the transfer of the operating system from one disk to another. This can be easily accomplished with the Dsysgen program. All XOR systems are organized in the following manner: Tracks 0-1 are reserved for the operating system. Track 2 is the directory. Tracks 3-77 are for data.

The formatting process must be performed first. This process completely blanks a disk and then places sector marks on the disk where the data is later to be written. The operating system is written next using DSYSGEN. REMEMBER, first format a disk with DFOCO, then type DSYSGEN, Answer (n) and (n) to the first two questions that come on the screen. A table will then appear with a sub-line asking "CODE BYTES FOR SOURCE DISK?". Usually you will type 1A, (the system disks we distribute with our systems are 1A) which signifies a 512, single sided, double density source disk. The other common code byte is 9A for a 512 double sided, double density, source disk.

The next question will be enter source drive, which will normally be A, hit a <cr>. A message will appear at bottom of the screen, at this point hit a "return".

The same table will re-appear and you will enter the code byte for the destination disk, usually either 1A, or 9A, and then enter the destination drive, usually B, hit <cr>, and wait for a short period, a complete message should appear, type a control C and you will be back in CP/M.

The preceeding procedure has transferred the operating system from A to B. You should now be able to remove the disk from drive B, do a reset, insert it in drive A and again boot up.

If you should ever see the error message, "type of source disk and destination disk do not match, do you wish to continue?" type "N", (this is a fatal error, you will destroy your operating system if you continue). You are probably trying to copy a single sided system to a double sided formatted disk. The program will restart, re-check the formats and try again .

Another item to be noted, all XOR systems are compatible with standard 128 single density, however there is not enough room on the system tracks of a single density disk to place the system. You must always boot up on a Double density disk in "A" drive. Single density files may then be transferred from "B" to "A" with the PIP program.

MODIFYING AND GENERATING CP/M SYSTEMS

The CP/M operating system is composed of several parts: the BDOS (Basic Disk Operating System), the BIOS (Basic Input Output System), the CCP (Command Control Processor), and the BOOT (Cold Boot Loader).

In the US Micros Sales system, the programs BOOT and BIOS are broken into several parts that are concatenated at the time that the system is built into one large module. The reason this is done is that fewer base modules need to be maintained to support all of the varied disk systems. Also, most larger multiuser modules would not fit in the system under the editor.

This BIOS system is intended to be used with SUBMIT modules that bear the name of the object BIOS produced. For example, BLKSYS61.SUB contains the following:

```
PIP
BLKSYS61.ASM=BLKEQU61.ASM,   IO.ASM,   CXLST.ASM   FLOP.ASM,
BLK.ASM, DATA.ASM, XBOOT.ASM
```

To concatenate this module, type: SUBMIT BLKSYS61 <CR>.

The PIP program picks out the necessary modules from the disk you have reserved for systems and creates the correct BIOS. The MACRO assembler will then assemble it for you.

Maintaining operating systems is a problem in that often minor "fixes" need to be added to them and having a dozen or so "versions" around means that not all of them will get the latest updates.

One of the greatest advantages of this system is that you only maintain one set of basic modules. In fact, we recommend that you keep only one backup copy of the disk containing these modules. In that way, you can be sure that the software you are building is the latest revision.

Modifications you wish to make should be added by putting conditional assembly flags around the code that is optional and establishing a flag in the main xxxEQU61.ASM header file. This keeps confusion and multiple copies to a minimum. Also, if you need to see if a certain group of files are present you can type: "DIR *.SUB" and just the system building files will appear. This is a real time saver when dealing with a hard disk that can display 4 or 5 screens of directory!

Programs you will need on the system disk to assemble a system:

1. ALL NECESSARY ... .ASM
2. CUSTOM BUILT ...SYS.SUB
3. SUBMIT.COM
4. MACRO.COM
5. MACRO.LIB
6. Z80.LIB
7. MOVCPM.COM
8. ED OR POLYVIEW (Polyview is HIGHLY recommended over ED)
9. DSYSGEN.COM

STEP BY STEP

A - MODIFICATION ***

The first thing to do is to edit the appropriate xxxEQU61.ASM and set all the equates to suit the configuration of the system you desire.

B - EXAMINATION ***

Next, type the ...SYS61.SUB file and look at the subparts needed to concatenate the final BIOS. Then Type DIR *.ASM and see if all the correct files are present.

C - CONCATENATION AND ASSEMBLY ***

If they are, type SUBMIT ...SYS61 The final product will be a ...SYS61.HEX file.

D - GENERATING CP/M ***

Type "MOVCPM 61 *"

The program MOVCPM.COM will create the BDOS for you. Usually, this needs to be done only once. Spend some time and make several of the common system sizes (e.g. 63K, 62K, 48K, etc.). When the MOVCPM has completed its job, type: "SAVE 34 CPMXX.COM" where XX is the system size.

E - PUTTING SYSTEM ON SYSTEM TRACKS ***

This is the job of DSYSGEN.COM. When you load DSYSGEN, it will ask you if the system has a file on disk. Answer 'N'. Then it will ask you if you want to build a system. This time, answer 'Y'. DSYSGEN will then ask you for the name of the CP/M file and then the name of the BIOS file. It will fetch the data that you have prepared and put it in the correct places in preparation for installation on the disk of your choice.

The next thing that will appear on the screen is a large table which has at the left of it some code bytes that will be written to the last byte of the first sector of the disk chosen.

At this time, the system you have created can be put on any of the disk formats specified in the table. Of course, the hardware must be compatible with the type of media you have picked and the disk must have been formatted with a format compatible with your choice (i.e. you could not put a 512 SKEW system on a 128 byte, double density disk.)

It should be noted that you need not put the system on a disk at this time. In fact, the computer system on which the operating system was created need not even be capable of writing to the drive type that the program you have built is designed for. You can type a control 'C' in answer to the "DESTINATION DRIVE" question and then type: "SAVE xx xxxCPMxx.COM" (the x's are unique system identifiers). The prepared operating system can be transported to the desired system via the floppy disk and read in using the first option on DSYSGEN (The one which asks if the system is a file on disk). For more information on how to use the DSYSGEN utility, refer to that section in this manual.

A third method for transferring operating systems is achieved by answering 'N' to both of the first questions DSYSGEN asks. DSYSGEN will then ask for the codebytes of the source disk you are going to take the operating system from. Then it will ask for the drive. From that point on you can proceed just as if you had constructed a system or read it as a file from disk.

The programmer must make sure that the operating system put on the system tracks will function properly. Very little error checking can be done in this process. It is extremely easy to make a "fatal" error. One of the most common is assembling a system that is too large to fit on the system tracks.

The best we can suggest for checking the load size is to go ahead and try to put the system on the disk. The DSYSGEN program will report to the screen the system size, load address, ending address and BIOS jump table adress for the current system. Also it will see if the system will fit on the system tracks of the media you have selected.

Use the following as a guide for how large systems may get:

PROGRAM	SIZE	TRACKS REQUIRED	IMAGE LOAD ADR
COLD BOOT	80H	1	900H - 97FH
CCP	800H	16	980H - 117FH
BDOS	E00H	28	1180H - 1F7FH
BIOS	F80H	31	1F80H - 2F00H
	2600H	76	900H - 2F00H

F - SYSTEM TRACKS AVAILABLE ON VARIOUS MEDIA ***

MEDIA TYPE	TRK-0	TRK-1	TRK-2	TOTAL
5" 128 DD	18	30	30	78
5" 512 DD	18	35	35	88
8" 128 SD	26	26	--	52
8" 128 DD	26	51	--	77
8" 512 DD	26	64	--	90

Note that these values are relative to 128 byte blocks. That is to say, if you multiply the total on the right by 128, the result is what DSYSGEN will record on the system tracks.

If your system exceeds the 76 track guideline used in the example for using DDT to check your system size, you will have to adjust the upper boundary accordingly. Simply add the system size to the image base address and look with DDT at the resulting RAM location.

You also can dump the system tracks into RAM after writing them by re-entering DSYSGEN and letting it get the system by specifying the drive at the prompt "SOURCE DISK?". Exit and type: "SAVE 40 (40 may not be enough) xxCPMxx.COM". Remember, the 40 is PAGES so a SAVE 40 = 40 X 256 = 10,240 or 2800H. If your system is very large, you will have to do some math.

G - DESTINATION DISK PARAMETERS ***

After you specify a destination drive, a large menu will fill the screen. You will be prompted to choose the type of media that you wish to write the system to. The DSYSGEN program is quite universal in that it can service all the system types U S Micro supports. Be sure the computer system and media you are writing to are capable of supporting the operating system you are putting on them. No error checking will be done.

Occasionally, a message will appear that says "CODE BYTE FOR OPERATING SYSTEM AND MEDIA DO NOT AGREE, CONTINUE?" This is caused because the formatting program (DFOCO) has installed what it thinks is the correct code byte for this system in the last byte of the boot sector (7FH). DFOCO is usually correct. If you get this message, go over what you are doing once again, you probably will find something wrong. You may override the warning and write the operating system to the disk.

DETAILED DSYSGEN OPERATING INSTRUCTIONS

DSYSGEN is a program designed to put the CP/M, operating systems on the system tracks of a floppy or hard disk mass storage unit.

The C/PM operating system consists of a cold boot loader, the CCP (Command Control Processor), BDOS (Basic Disk Operating System) and the BIOS, (Basic Input - Output System).

On a floppy disk, the operating system resides on the first two tracks of the disk. The monitor PROM that signs on immediately after a reset contains a short boot routine that reads the first sector into memory at 0000H and jumps to 0000. The cold boot loader, as it is called, then proceeds to read in all of C/PM and put it where it belongs in the system memory. The cold boot loader then jumps to the cold boot entry point in the BIOS and initializes the I/O and completes the sign on.

To prepare for putting the system on the disk, an image of the final operating system must be assembled in RAM. Two separate programs must be prepared.

The first is the C/PM BDOS and CCP. It is prepared by using the MOVCPM utility on your C/PM distribution disk. When you are sure of the size of the system simply type: "MOVCPM <size> *". There will be a short delay and then a message will appear saying "Type SAVE 40 C/PMxx.COM" where xx will be the size.

The other program is a little more difficult in that you must correctly set some "equates" at the beginning to configure it to operate the hardware you are running. In the appendix, you will find some examples of single user, multi user, network and hard disk configurations that should be of help in determining the right way to set up the BIOS. You will notice that the BIOS is now broken into several subparts. The parts are explained as follows:

I/O = Cold boot loader, console input and output, common drive select routines, logical unit map, miscellaneous routines and messages needed by all configurations. I/O must be included in all system configuations.

CXLST = Printer driver, (There are several versions for commonly-used printers such as Centronics, Diablo, etc.)

FLOP = Floppy disk drivers.

BLK = Blocking and De-blocking drivers for use with sector sizes greater than 128.

DATA = Disk parameter tables, skew tables, data storage and disk drive buffers.

XBOOT = Cold boot sign on code that gets overwritten by the directory buffer and de-blocking buffer.

The parts of the BIOS are put together using the concatenation feature of PIP. To make the system simply choose the appropriate xxSYS.SUB file that assembles the configuration you need and type "SUBMIT xxSYS . The MACRO assembler is required along with MACRO.LIB and Z80.LIB to construct most modules.

The equates in IO should be modified using a text editor like the one supplied on your distribution disk called ED.COM. ED is difficult to use. We recommend, for assembly language work, that you purchase Polyview by MICRO CONCEPTS in Fullerton, Ca. Word Star by Micro Pro would be an acceptable substitute, but is not as easy to use as Polyview on assembly formats.

The size of the system can be adjusted by changing the MSIZE equate in the BIOS. This number should always be the same as the size you used to create the C/PM BDOS.

Once you have the CP/M image saved on disk and have modified the equates correctly to configure the BIOS to your hardware, run the SUBMIT program on the xxSYS.SUB which links together the SUB modules to create the final BIOS and assemble the whole thing.

At this point, we will assume you have both CPMxx.COM and xxSYS.HEX on the same disk with DSYSGEN.COM. Notice that

there is no longer a separate BOOT.HEX file. It is included in the IO section of the BIOS. There is also no need to find the 'BIAS' values to tack onto the BIOS, the DSYSGEN program figures all of that out for you.

When DSYSGEN signs on it will ask if the system you wish to install on the drive exists as a xxxxx.SYS file on the disk already. If you are sure to use the current configuration again, it would be wise to save it after building it with DSYSGEN as a file. Then you need only answer 'Y' to "IS THE SYSTEM ON THE DISK". This saves a few steps. If you are supporting several configurations of mini and 8" drives, various printers, etc., you could keep one diskette with DSYSGEN and all the system files on it. Then putting them on the drives is a snap.

For the moment, we will assume the system has to be built. Answer 'Y' to the query: "DO YOU WANT TO CONSTRUCT A SYSTEM". DSYSGEN will ask for the drive and file name for both C/PM and what is called BOOT + BIOS. DSYSGEN will ask for a destination drive, confirm it, then display a rather large menu.

This menu is a guide for the code bytes stored in the last byte of the first sector of all drives. The appendix explains the meaning of each bit in the code byte, if you are interested. Other details are included for your information. This data was taken from a listing of DSYSGEN, so it is in rather raw form.

Whenever possible, it is advisable to choose the "SKEW" option. Many companies are currently raving about their 'four times faster' C/PM. Optimizing the skew factor is all that is required to multiply disk performance (You should try the double-sided 512 byte 8" configuration if you would like to see a floppy perform like a hard disk).

Using the DFOCO (for Double density FOrmat and COpy) utility, you should have already prepared your disk with the correct format. After answering with the correct code byte, DSYSGEN will write the system onto the disk. If the code byte on the disk and the code byte you have chosen do not agree, the following message will appear: "The new system type and the destination disk format do not match. Do you want to continue (Y or N) ?". This may or may not be ok. DFOCO does not know what you will use for a code byte and has some fixed ideas about what it puts on the disks. Situations such as putting a code byte for an 8" floppy disk on a hard disk drive are incorrect, but choosing a skew code when DFOCO has not put one there (or vice versa) is ok. Presently, skew codes are automatically put on 512 byte sectored disks by DFOCO because we assume you are after the speed and the extra space which that format provides. On other formats such as 128 byte, double density, skew is not assumed because you may have compatability problems with existing disks. In any case, DSYSGEN will overwrite whatever code byte is there if you tell it to.

After writing the system out, DSYSGEN will state that the function is complete and ask if you want to do another disk or hit 'return' and re-boot. Upon being re-booted, it will display a "SAVE" message with a size that has been calculated for the

number of bytes in your current system size. If you use this option, you can avoid going through the whole system building process again by simply using the image off disk.

Some attempts are made in the DSYSGEN program to aid you in building a system that will not overwrap FFFF and destroy the boot program that is loading the system. If DSYSGEN detects that your system is too large for the memory size you have chosen, it will tell you so. If the system will not fit on the system tracks of the disk you are using it will notify you of this, as well. In the latter case, DSYSGEN will trim the system to fit (not doing so would hang the disk controller).

PRINTER PORTS (PARALLEL AND SERIAL)

CXLST.ASM

A parallel printer driver has been installed in the bios CXLST.ASM. This driver can be used for Centronics type cable interfaces. (See section entitled ENGINEERING NOTES at the back of this manual for parallel cable wiring.) Three methods may be used to "turn on" this driver. As shipped the bios drives the serial port "A" on the CPU with all CP/M list outputs. To switch all output to the parallel port on a temporary basis you can set a byte in memory at EE33H to a 05H. This can be done with DDT or BASIC with "POKE" instruction. This memory can be calculated for different system sizes by picking up the the bios jump table and can be calculated for different system sizes by picking up the warm boot entry address at memory location 1 and 2, and adding 30H to that address. (30H = 48 decimal)

USING DDUMP TO CHANGE BYTE OF SYSTEM DISK

A second method, which is permanent for the disk that the change was made on and all copies of that disk is to use the system utility DDUMP.COM to modify the byte on the system tracks of the floppy. You will find the bios jump table on track 1 sector 20.

```
DDUMP <CR>              (OPERATOR INPUT)
*                       (PROGRAM RESPONSE)
T 1 S 20 EDIT <CR>      (OPERATOR INPUT)
- EDIT                  (PROGRAM RESPONSE)
33 <CR>                 (OPERATOR INPUT)
0033 00                 (PROGRAM RESPONSE)
05 <CR>                 (OPERATOR INPUT)
0034 00                 (PROGRAM RESPONSE)
  . <CR>                (OPERATOR INPUT) TERMINATES EDIT
WRITE <CR>              (OPERATOR INPUT) WRITES YOUR EDIT TO DISK
^C                      (OPERATOR INPUT) RETURNS TO CP/M
```

Obviously if you should want to switch back to seriel output, the same byte should be set to a "0".

USING MACRO ASSEMBLER TO CHANGE BIOS JUMP TABLE

A third method is to use a text editor to modify the bios source code module IO.ASM which contains the bios jump table. You will need the Digital Research MAC assembler to do this. It may be purchased from your closest U S MICRO SALES OFFICE.

PRINTER.COM

PRINTER.COM will select the printers by simply typing PRINTER 3 (for example). PRINTER.COM can also change the baud rate that is supplied to the respective printer. The default is to printer 0, and outputs to the serial driver connected to the "A" paddle card.

PRINTER.COM also will send either single characters or a continuous string so that system handshaking can be tested. BEFORE you try to send data to your printer, run PRINTER.COM in the test mode to see if the status bits are set correctly. This will save an enormous amount of time in getting things to run.

Just typing "PRINTER" will tell you what printer is currently selected.

PRINTER T (TEST MODE)

PRINTER.COM has a "TEST" mode which has several useful routines which can save many hours of frustration when trying to get your printer running.

Type "PRINTER T<CR>

A menu will then come on the screen.
(A spacebar will return you to menu in most instances. A Ctrl C will return you to CP/M.)

"P" = Select new Printer
"B" = Select new Baud Rate
"T" = Manual Output Mode (Characters sent from keybord, ESC terminates)
"A" = Auto output mode (String sent until space bar hit)
"S" = Display printer status byte

"P", "B", "T", "A" OR "S" ?

The "P" option will respond with "Printer" ?
You must enter the number of the printer you want to use, i.e "1", which could be a printer hooked up to the parallel port.

THE "B" Option will respond with

1 = 300 baud
2 = 600 baud
3 = 1200 baud
4 = 1800 baud
5 = 2400 baud
6 = 3600 baud
7 = 4800 baud
8 = 9600 baud
9 = 19200 baud

Select baud rate ->

Respond with the number, i.e., 1, 3, 7, of the baud rate that you want to select for the printer selected. In order to check that the baud rate option works properly you may set the baud rate to a slow baud rate of 300 and use the "A" option explained below. The characters are automatically generated at a very slow speed. Reset the baud rate to the original number your printer uses, and select the "A" option again and see the difference.

The "M" Option will respond with a blank line. You must manually input characters which will be output to the printer and echoed onto the screen. Hit ESC to terminate this test and return to the menu.

The "A" option will send a continuous string of ASCII characters to the printer and the terminal until the space bar is hit. If the printer is not ready a "printer not ready" message will appear on the screen after a short pause.

The "S" option will respond with a screen showing what good status bits for parallel, 8251, SIO, and the printer currently selected should look like.

```
PARALLEL Status Bits
1 X 0 1 X X X X = Good Status
W   B 0
A   U N
I   S L
T   Y N

8251 Status Bits
1 X X X X 1 X 1 = Good Status
D S F O P T R T
S Y E E E X X X
R         E R R

SIO Status Bits
X X 1 X 1 1 X X = Good Status
B T C S D T I R
R X T Y T X N X
K U S N R R T R

1 0 0 0 0 1 0 1 = Status Byte for printer 0
```

In the above example the bottom line, the Status Byte for the printer selected is printer 0 and the port being used is the 8251. Read down the columns from the good status line under 8251 Status Bits and compare with the Status Byte for printer 0. 1's and 0's must match down, X's are "I don't care" and a 1 or 0 in that column doesn't matter.

If you can't go through these programs without the console output freezing or the Status bytes matching up, check to see that the resistors are installed properly on the A paddlecard. If this doesn't cure the problem check your cables to make sure that the pins are in the proper places for your particular printer and that the dip switches are set properly on the printer.

When using the parallel printer port, baud rate is not a factor. The status bytes will not match until the printer is on line. If the status bytes do not match, check your cable wiring, check your switch settings on your printer, make sure that you have the right printer number selected, and that the bit in the sofware at track 1 sector 20 line 33 is set to 05.

CUSTOM PRINTER.COM PROGRAMS

Custom PRINTER.COM type programs can be written and renamed DIABLO.COM or MX80.COM or TI810.COM (for example). These can be very simple programs that merely modify one or two cells in system memory.

The memory cell 30H bytes past the warm boot entry in the jump table is where the currently logged printer is kept. Bytes at 31H,32H and 33H offset from warm boot are the baud rate bytes for the cpu baud rate generator, serial board 1 and 2 respectively.

The system can handle 10 serial ports and 1 Centronics type port with parts available from your USM SALES OFFICE. A0 at offset 30H from warm boot will select channel "A" serial port on the cpu. This is printer # 0. CXLST always initializes to this printer.

Printers 1, 2, 3 and 4 are expected to be on an optional serial expansion board that is available from your U.S. MICRO SALES OFFICE. Printer 5 is expected to be the parallel port on the CPU. An example of a program that would select printer #3 and set the baud rate to 1200 baud is as follows:

```
ORG 100H

START:
        MVI  A,03H          ; printer select
        LHLD 1              ; get warm boot address
        LXI  D,30H          ; offset
        DAD  D              ;
        MOV  A,M            ; set printer
```

The baud rate port on the CPU is at I.O. port location 0BH. A slight complication arises in that you probably would not wish to change the baud rate at which the system terminal is running. In this case you would have to read the CPU baud rate switch and ORI the upper 4 data bits on to your new lower 4 bits before re-programming the baud rate generator. Look at how things are done in PRINTER.ASM as an example if you have any problems.

An easy way to see if you are sending information to the printer is to type a "CTRL P", then type "D" or "DIR" and see if your printer responds by typing the directory of the disk. If there is no response from the printer read through the different sections on printer hookup and see if you can find the problem.

SOME TIPS ON HOOKING UP PRINTERS

Almost all serial printers use four basic lines. Our computers "look" like modems in terms of RS-232 specifications. That is because originally terminals attached themselves to modems that in turn talked over phone lines to mainframes. When cheap Microcomputers came on the scene it was logical to make them plug directly into the terminals.

The minimum lines needed to maintain communication are Transmit, Receive, Logic Ground, and Frame Ground. These are on a DB-25 connector on pins 3,2,7 and 1 respectively.

These terms only hold true if you are a modem. DB-25 connector pin assignments should always be referred to as if the modem was the subject. Therefore if pin 3 is "Transmit", if you are a terminal, you should receive on pin 3.

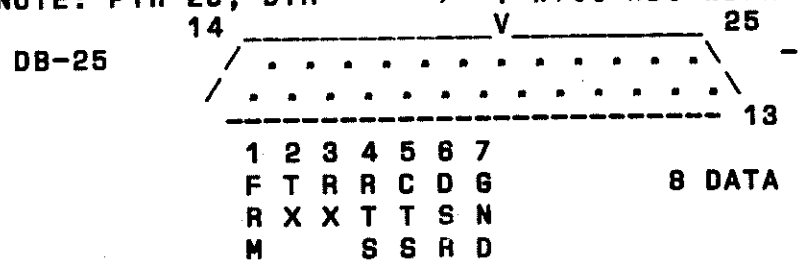
There are other signals on a DB-25 RS-232 serial connector that are standard. These signals were originally intended for modem use and their names reflect these intentions. They are; CTS (Clear to send), RTS (Request to send), DSR (Data set ready), DTR (Data terminal ready).

These "secondary signals" as they are called are used to inform the terminal user when the computer and transmission equipment can send data.

When attaching a printer, we use the principal signals to convey data in both directions and the secondary signals to "handshake". To handshake means to inform each other when you are prepared to communicate. It's rather like picking up the phone after hearing it ring and saying "Hello". Then the other person identifies himself, and so on. Humans "handshake" in many ways that are not readily apparent. Facing someone and nodding your head to start a conversation is "handshaking".

On our Peddle cards at the back of the computer are two resistors. They are called "pull up" resistors and must be installed if the serial port handshaking lines DSR and CTS are not to be used. These resistors can be anything from 1K to 10K. If buffer full handshaking is to be used, the correct line must be located which indicates that status on the printer you are using. The resistor "pulling up" the status line you wish to use must be removed (either can be used), and a jumper soldered between the RS-232 line and place where the resistor was. The other available handshaking line can be used for ON LINE/ OFF LINE status or ??. But for any transmission to take place both serial units must be at the same baud rate and both handshake lines must be acive (high).

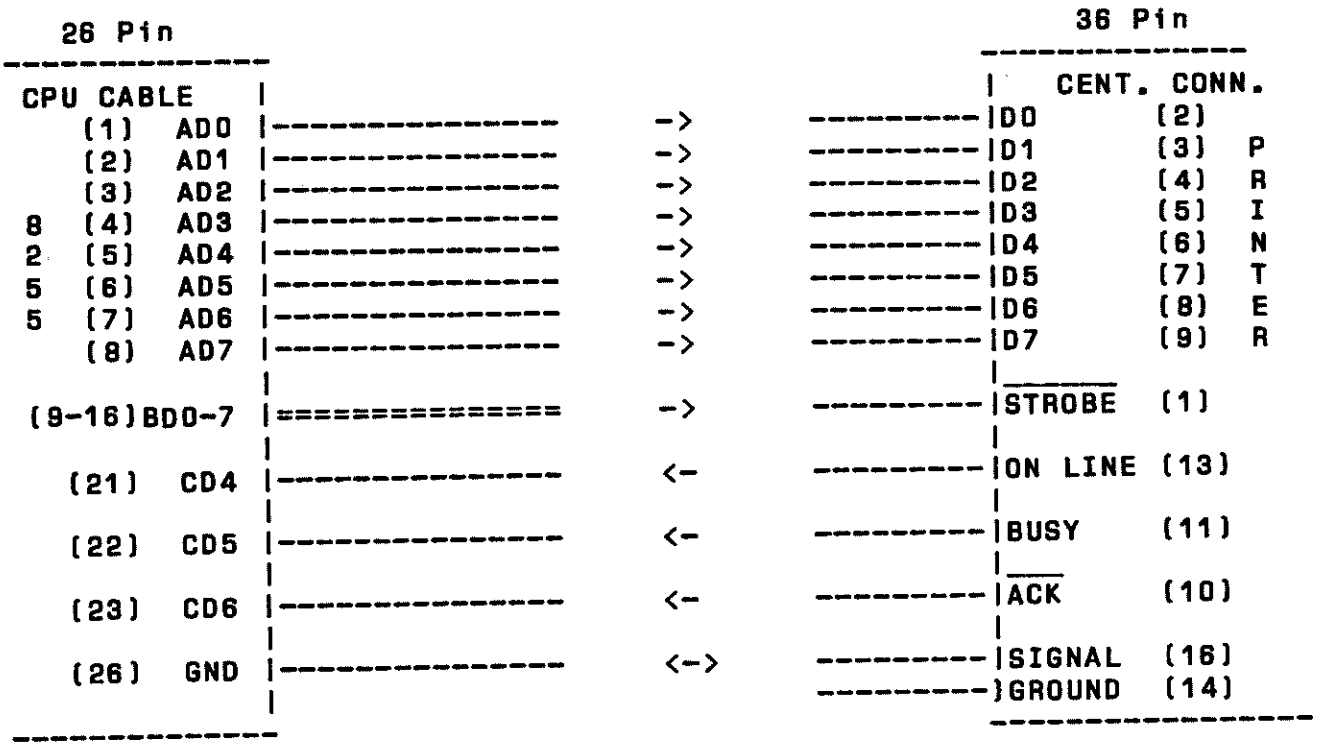
NOTE: Pin 20, DTR -----> | Will not work on 8251, use CTS, pin 5



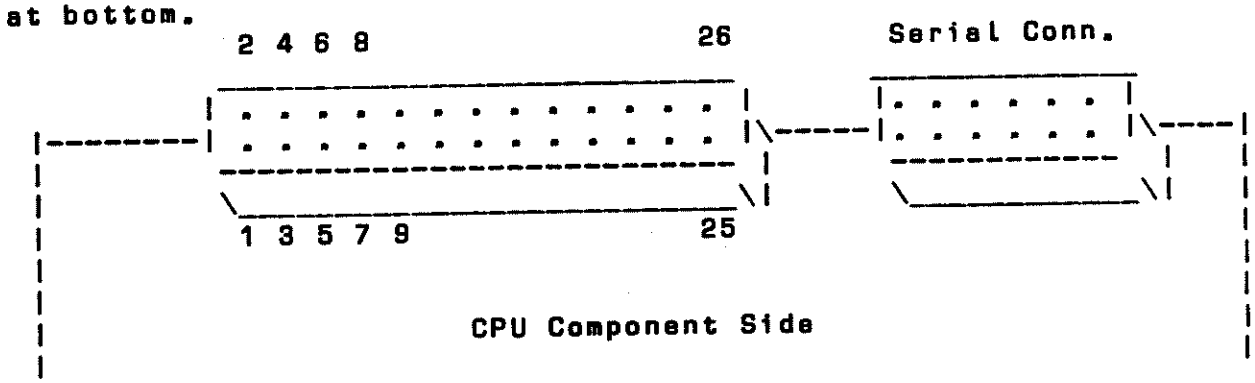
PARALLEL CABLE WIRING

This cable is used for Centronics interface printers. It uses a 8255, PA0-7 are used for data transfer, PB0-7 are tied together for data strobe PC6 for ACK from printer.

NOTE: This cable is available from your U S MICRO SALES SALES OFFICE.



CPU 8255 connector pinout looking from front with edge connector at bottom.



CHAPTER FOUR

TROUBLE SHOOTING AND SERVICE POLICY

MONITOR AND SIGN ON

If upon turn on there is no sign on message, check the following, (1) Is the proper RS-232 cable installed on Port (B (the lower port)? (2) Is the Baud rate set correctly on both CPU and the terminal? (3) Are all 3 of the computer boards seated properly in their sockets?. (4) Is the internal serial cable connected to the serial port on the CPU board?. If your computer still does not sign on you can shut off the power, remove the CPU board and check it for any obvious defect such as loose ic's or optional jumpers placed wrong.

MEMORY CHECKS

Each time the computer signs on, a check of all available free memory is performed. The correct ending address for a successful menory check is F800. If any other address is displayed this indicates a problem and further checks should be made before attempting to boot up. Reset more then once and notice if the same address is displayed every time. This may indicate which block of memory is malfunctioning.

Typical memory failures are caused by loose memory ic,s dirty contacts on the memory board, or possibly intermittent contacts on a particular slot of the 12 slot mother board. Try the memory board in another slot, and also reseat all ics on the memory board.

MEMORY TESTS

M AND M2

M AND M2 ARE EFFECTIVE QUICK CP/M ORIENTATED MEMORY TEST PROGRAMS THAT SHOULD BE READILY AVAILABLE DURING THE NORMAL COURSE OF USE OF ANY MICRO SYSTEM.

BIT SHORTED:

FILLS TEST AREA WITH FF'S AND CHECKS FOR FF'S
FILLS TEST AREA WITH 00'S AND CHECKS FOR 00'S

BIT SHORTED:

ROTATES A BIT FROM LSB TO MSB FILLING TESTED MEMORY EACH TIME, CHECKING ONE BIT AT A TIME

ADDRESS SHORTED:

FILLS ALL MEMORY WITH 55 HEX THEN WRITES AN AA HEX AT 0000 (OR THE LOWEST ADDRESS TESTED). IT THEN TESTS THE REST OF MEMORY FOR 55'S. THEN IT CLEARS LOCATION OF THE AA AND WRITES IT INTO 0001. THEN 0002, 0004, 0008, ETC. SETTING A NEW ADDRESS BIT HIGH EACH TIME AND TESTING ALL OF MEMORY. IF ANY ADDRESS BIT IS SHORTED TO ANOTHER, THE TEST WILL FIND AN "AA" IN A LOCATION OTHER THAN WHERE IT WROTE ONE. THIS TEST TAKES THE MOST TIME AND IS RUN LAST.

RANDOM NUMBERS:

A RANDOM NUMBER ROUTINE GENERATES AN EIGHT BIT NUMBER PATTERN AND WRITES IT THROUGH ALL TEST MEMORY. IT THEN RE-INSERTS THE SAME SEED TO THE ROUTINE AND TEST READS THE MEMORY. A NEW SEED IS GENERATED AND THE EXERCISE IS REPEATED WITH A NEW PATTERN. THIS TEST CONTINUES, REPORTING EACH PASS UNTIL ABORTED WITH THE SPACE BAR.

The first test (M) when run in its fundamental mode (executed under CP/M by typing M (cr) then (cr) (cr) in response to the start and stop questions,) will calculate the system size and begin a typical pattern of tests to uncover the most common problems first and the most uncommon ones last.

Typing a 'P' along with the address in the address field will PAUSE the test program after the memory fill passes. After the operator has waited an appropriate amount of time he may hit (cr) to continue the test. This feature will test for the existence of Phantom-bits. These are memory cells that change after being left un-accessed for a period of time. Initiating the test with M<sp>R<cr> will exclude the address and data tests and drop immediately into the random numbers test.

M AND M2 (which take quite some time) may be skipped and the random numbers test entered directly by typing M R at the time of program invocation. Phantom bits may be located by typing a 'P'in the first address field. After filling memory with a test pattern, the program will halt until you press a key. You may leave the computer for several hours, if you wish, the program goes on. If memory has changed, it will report the error. For CPM users who want just a quick 'security blanket' test, CR,CR in response to the address requests will automatically test the TPA without destroying BIOS. Upon hitting 'SPACE' warm boot will get you back into CPM.

WORM AND WORM48

WORM and WORM48 are effective memory tests which run freely within RAM. These memory tests are destructive to the operating system. WORM is generally used in single user systems with 64K of RAM. WORM48 is generally used in multi-user systems with more than one 48K memory board. This program starts in bank 0 and runs until it reaches the top of the first bank, then switches to the next bank of memory until it reaches the last bank and then starts again with bank 0. WORM48 is faster than WORM and has a screen report which tells the user the address, the bank, and the pass count the program is on. When using WORM48 the program will ask the user for number of memory banks to be tested, the user should answer with the total number of banks within the unit, minus one (the program counts the memory banks starting with 0). The number the user inputs can be less but never greater than the total number of banks in the unit.

WORM is designed to find memory cells that cannot stand rapid cycle time demands placed on them by some of the Z80 block move and math routines. While these chips will pass all standard memory tests for hours on end, they randomly fail during program execution time for seemingly unexplained reasons. The reason is that there is a difference between a chips ACCESS time and its

CYCLE time. We generally only rate a memory board by its access time, (i.e. 450ns or 250ns.)

Worm starts itself at the load address and relocates itself repeatedly, reporting its current location periodically. It 'tests' by simply moving itself and constantly doing instruction fetches. A RAM failure is evidenced by the program crashing. The last reported address on the CRT should be interpreted to be the bank in which the errant chip resides. Binary replacing of chips, one half at a time, watching to see when the problem moves to the new bank, will uncover the bad device.

CPU CHECKS

Very little trouble is experienced with the CPU. Improper jumpering of the 2/4 Mhz can cause non double density operation. If you are having trouble with one serial channel, you can exchange the 8251 and/or the 1488/1489 chips to either serial channel for troubleshooting purposes. For a completely inoperative CPU, a check of the onboard regulators, and a look at the CPU clock signals and the Baud rate clock would be the first items to check. All IC's on XOR boards are socketed for ease of troubleshooting.

DISK CONTROLLER CHECKS AND ERROR MESSAGES

The XOR ver 1.85 prom has internal fault messages which will be displayed on the screen if the following occurs:

For no system " NO SYSTEM" will be displayed, for a Read ID or CRC error, DISK ERROR will be displayed. Try another disk with a known good system .

If you attempt to boot up with the door open the message "DRIVE NOT READY" will be displayed.

The error code 10 means that READ ID NOT FOUND, either the controller cannot read a valid sector on the disk or it is not there in the first place.

The error message 08 means that sufficient time has not been given for the head to load and to read a valid track, generally you can reboot sucessfully.

CRC error means that the proper Cyclical Redundancy Checks have not been made, this can be caused by bad diskette, malfunction in the controller, or incorrect speed of the drive.

In XOR systems with the Tandon thin line drives, the speed can be checked by looking at the large belt spool on the side of the drive under a flouresecent light. The strobe lines should be stationery. Refer to fig for the location of the motor speed adjustment.

An error code of DRIVE NOT READY can be caused by not enough time for the motor to get up to speed. This is sometimes caused by belt slippage. Belt dressing from your auto parts store that is usually used on slipping car fan belts, will cure this problem. Remove the belt and place on a piece of newspaper. Spray the inside of the belt and allow to dry before replacing.

POWER SUPPLY CHECKS

Your S100-4 is has two internal power supplys. One supply provides unregulated +11 VDC and +/- 16 VDC for the bus. Refer to fig for the voltage measuring points. The second supply mounted on the rear panel provides regulated +24VDC and +/-5 VDC for the drives. Refer to Fig for the voltage measuring points. Note that the +5 is adjustable. These voltages should remain steady under load, and any deviation would indicate some malfunction of the supply itself. Check for any broken leads on the large filter caps, or loose screws on the regulator transistors, note there are two fuses on the board.

SERVICE POLICY

We would hope that your XOR would never require service, however, we feel that with a thorough reading of the manual most problems will be resolved. Should you feel the need to consult with our Customer Service personnel, they will be happy to answer your questions. Please call the number (714-898-5525) during normal business hours.

If you should require service on your XOR it will be performed at the plant facilities in Huntington Beach, or at an authorized XOR Service Center.

Equipment will be accepted for service only after you have been assigned a "Return Material Authorization Number". RMA. This number may be obtained by calling the Customer Service Hot Line 714-898-5525. The following information will be required:

The purchase date
The serial number
The problem

All shipments received without an RMA will be returned to the sender.

Warranty (6 months parts/labor, from the date of purchase) repairs will be made at no charge for parts and/or labor for XOR hardware only. Merchandise such as drives, terminals, and printers sold by US MICRO SALES as part of their systems will be sent back to the manufacturer for repair. Repair and freight charges for non-XOR merchandise will be billed to you by US MICRO SALES. All warranties are void if any portion of your computer system is altered by anyone other than factory authorized personnel.

Incoming freight MUST BE PREPAID. US MICRO SALES will not accept incoming freight which is marked COLLECT or COD. Shipping charges for the return of repaired equipment to you that is under warranty will be prepaid by US MICRO SALES except in cases where you specify method of shipment other than the one chosen by US MICRO SALES. All return shipping charges for equipment that is no longer under warranty are the responsibility of the party who sent the equipment for repairs.

SUGGESTIONS

If you should have any suggestions regarding the use of this manual, or have any corrections or additions, we would be happy to hear from you. Please address your correspondence to:

U S MICRO SALES
Tech/Man Dept
15392 Assembly lane
Huntington Beach, CA 92649

rev 030883

!!! ANOUNCEMENT !!!

Several new features are contained in the 10/15/82 release of your XOR CP/M.

1.) DIAGNOSE.COM is an automatic, non-destructive system test utility that will test system memory and both drives. It should be run before drive "B:" has been accessed. The tests take about 5 minutes to run and assure everything is in good shape to run your software. Data is output to system printer so you need not wait and watch the screen. (Got to keep up with IBM you know!)

2.) BACKUP.COM will help first time users make that initial important backup of their CP/M source disks. BACKUP should be the FIRST program that you run after receiving your computer. The source disk goes in drive "A" and a blank destination disk in drive "B". BACKUP will format the destination disk and copy A to B. (In all XOR systems drive "A" is on the left and drive "B" is on the right.)

3.) HELP.COM will aid first users of CP/M, BASIC, and many other run time programs. Simply type HELP CPM <CR> or HELP MBASIC or whatever ???.HLP program libraries appear on your current distribution disks. To see what they might be, type D *.HLP <CR>.

4.) Maybe this is not a feature but it does have to be mentioned. If your drives are made by TANDON, (a slimline with closing lever at the upper right) the disks are to be inserted with the label (top) to the RIGHT. This is different than the other models used in XOR systems in the past.

5.) The disk drivers for 48 and 96 tpi, single and double sided mini 5" drives have been turned on in the distribution version of the CP/M "A" and "B" disks. We strongly recommend (in fact insist?) that you use the 512 byte per sector mode for all the mini types. To run minis with 8" drives, simply plug in the drives and run DFOCO to format the minis. The command line to correctly format a 96 tpi double sided drive in DFOCO would be;

* DFORMAT C: SIZE 512 MINI96 DBL <CR>

We are assuming drives "A" and "B" are 8" and drives "C" and "D" are strapped as minis.

After formatting and PIPing data to the mini, the operating system may be transferred to the mini useing DSYSGEN. See the manual for specifics on using DSYSGEN. Transfer method #3 where you answer <CR>, <CR> to the first two questions and then take the OS directly from the 8" floppy and put it on the mini would be the easiest to do. After successfully accomplishing the above two feats, re-strapping the floppy controller so that the minis are "A" and "B" will allow you to boot from them after a reset.

Some problems have been noticed with version 1.83 proms booting

mini floppies by certain manufacturers. It will work but sometime not always on the first try. It is caused by a drive ready signal not being available as is on 8" systems. A new version 1.83A is supposed to help the problem.

8 INCH JUMPER STRAPS AS SHIPPED FROM FACTORY A,B = 8" C,D = MINI

5 INCH AS A and B

6.) A parallel printer driver has been installed in the bios STDLST.ASM. This driver can be used for centronics type cable interfaces. (See the appendix for parallel cable wiring.) Three methods may be used to "turn on" this driver. As shipped the bios drives the serial port "A" on the CPU with all CP/M list output. To switch all output to the parallel port on a temporary basis you can set a byte in memory at EE33H to a 01H. This can be done with DDT or BASIC with "POKE" instruction. This memory can be calculated for different system sizes by picking up the the bios jump table and can be calculated for different system sizes by picking up the warm boot entry address at memory location 1 and 2, and adding 30H to that address. (30H = 48 decimal)

A second method, which is permanent for the disk that the change was made on and all copies of that disk is to use the system utility DDUMP.COM to modify the byte on the system tracks of the floppy. You will find the bios jump table on track 1 sector 20.

```
DDUMP <CR>
T 1 S 20 EDIT <CR>
33 <CR>
01 <CR>
WRITE <CR>
^C
```

A third method is to use a text editor to modify the bios source code module IO.ASM which contains the bios jump table. You will need the Digital Research MAC assembler to do this. It may be purchased from your closest U S MICRO SALES distributor.

Obviously if you should want to switch back to serial output, the same byte should be set to a "0".

Note that another complex printer driver option has been supplied in the form of CXLST.ASM and its associated PRINTER.COM. To install CXLST requires modifying the bios source module by editing the concatenation submit command file BLKSYS61.SUB. In this command file several modules are concatenated by PIP.COM to assemble the system bios. The distribution version contains a simple driver called STDLST.ASM. By replacing STDLST.ASM with CXLST.ASM and re-assembling the bios, up to six printer drivers can be resident in the bios at one time.

PRINTER.COM will select the printers by simply typing PRINTER 3 (for example). PRINTER.COM can also change the baud rate that is supplied to the respective printer. Source code to PRINTER is supplied on your "B" system disks.

Custom PRINTER.COM type programs can be written and renamed DIABLO.COM or MX80.COM or TI810.COM (for example). These can be very simple programs that merely modify one or two cells in system memory. The cell at 003DH is used to indicate the currently logged printer and the one at 003CH to set the baud rate. A "0" at 003DH means the printer attached to the CPU channel "A" serial port is selected. This is printer # 1. CXLST always initializes to this printer. Printers 2,3,4 and 5 are expected to be on an optional serial expansion board that is available from your U.S. MICRO SALES distributor. Printer 6 is expected to be the parallel port on the CPU. An example of a program that would select printer number 3 and set the baud rate to 1200 baud is as follows:

```
ORG 100H

START:
        MVI  A,03H          ; printer select
        STA  003DH          ; cxlst will look here
        MVI  A,07H          ; see CPU manual for binary bauds
        STA  003CH          ; upper nib is left brg, lower is
        RET                 ; right, here we pgm the right.
```

The baud rate port on the CPU is at I.O. port location 0BH. A slight complication arises in that you probably would not wish to change the baud that the system terminal is running at. In this case you would have to read the CPU baud rate switch and ORI the upper 4 data bits on to your new lower 4 bits before re-programming the brg. Look at how things are done in PRINTER.ASM as an example if you have any problems.

Another common problem we get calls on bears mentioning. It is explained in the General Operating proceedures section of your manuel how to put the operating system from one disket to another, but for some reason many customers don't understand from

the reading. If you want to transfer the operating system from one disk to another, simply load DSYSGEN and answer (N) and (N) to the first two questions that come on the screen. A table will then appear with a sub-line asking "CODE BYTES FOR SOURCE DISK?". Usually you will need to enter 1A, which signifies 512 byte single sided. The other common code byte is 9A, which is for 512 byte sectors, but double sided. After determining the type of source disk and drive name, hit a <CR> to "CONTINUE". The next question asked will be for the code byte for the destination disk. After telling dsysgen the type and drive, it will do the transfer. It's really very simply and should take no more than 15 seconds to do. If you find yourself well into the first hour, READ THE MANUEL. There is a step by step example in there. Don't forget that the destination disk must have a suitabl format on it. If your not sure on this point, use DFOCO to format the disk.

CPU INFO -

Port assignments:

```
00   Channel "A" status (See status flag below)
01   Channel "A" data
02   Channel "B" status (See status flag below)
03   Channel "B" data
04   Parallel "A" data
05   Parallel "B" data
06   Parallel "C" data
07   Parallel command port
08   Memory management
09   Prom Toggle, 00 = on    01 = off
0A   Boot prom disable
0B   Read/ Baud rate switch    Write/ Baud rate generator
0C   CTC chan 1
0D   CTC chan 2
0E   CTC chan 3
0F   CTC CMD
```

The 8251 can be programmed under software control to do a number of things. The following code can be used to initialize what might be a "normal" mode for the 8251.
[The chip must be initialized or it will do nothing.]

```
MVI A, 0AAH    ; Load A
OUT 03         ; Initialize Port B
OUT 01         ; Initialize Port A
MVI A, 40H     ; Load A
OUT 03         ; With Internal Reset
OUT 01         ; Write to Both Ports
```

This initialization will set up 8 data bits, one stop bit and no parity. Your printers and terminals have to be set up to receive the same pattern.

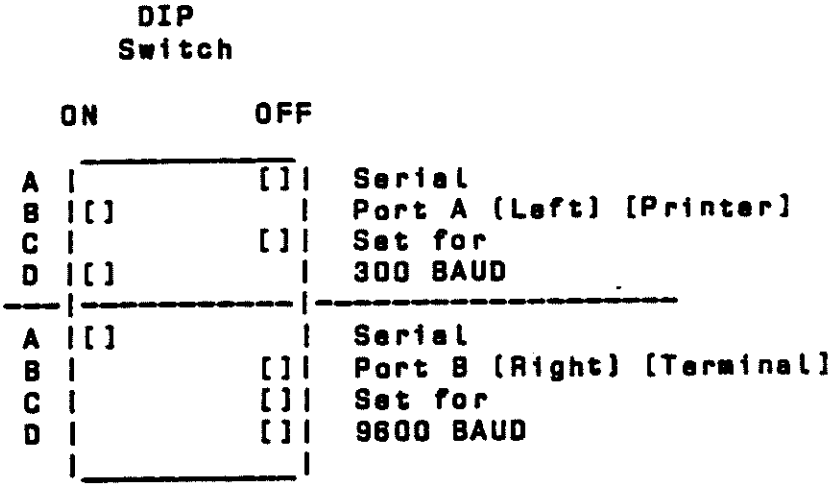
8251 Status Flags (when you input status port, this is what byte will mean):

```
Bit:  07  06  05  04  03  02  01  00
      ---------------------------------
      DSR  SY  FE  OE  PE  TXE RXR TXR <-Output bit (RDY when Hi)
                                 |
                                 |--------Input bit (RDY when Hi)
```

Ports for the Serial IO are: A Status = 1 A Data = 0 (on left)
B Status = 3 B Data = 2 (on right)

The 8 position DIP switch at the lower right hand corner of the board is broken into two 4 bit sections. The upper 4 bits select the IO baud for the left 8251 and the lower 4 bits select the baud for the right. We connect the left 8251 to the "A" paddle card and the right one to the "B" paddle card. The paddle cards are mounted to the rear panel and make strapping adjustments easier. The "A" port is committed in the CP/M bios software to the printer, and the "B" port to the CRT Terminal.

The switches are used in a binary pattern to set the rates as follows:



X = O F F O = O N

Baud	D	C	B	A	
50	0	0	0	0	(00H)
75	0	0	0	X	(01H)
110	0	0	X	0	(02H)
134.5	0	0	X	X	(03H)
150	0	X	0	0	(04H)
300	0	X	0	X	(05H)
600	0	X	X	0	(06H)

Baud	D	C	B	A	
1200	0	X	X	X	(07H)
1800	X	0	0	0	(08H)
2000	X	0	0	X	(09H)
2400	X	0	X	0	(0AH)
3600	X	0	X	X	(0BH)
4800	X	X	0	0	(0CH)
7200	X	X	0	X	(0DH)
9600	X	X	X	0	(0EH)
19200	X	X	X	X	(0FH)

N O T E :
Make sure when using the above chart that the positions we show correspond to the type of switch you have on the board. Some models of switches have the "ON" to the opposite side.

SOME TIPS ON HOOKING UP PRINTERS

Almost all serial printers use four basic lines. Our computers "look" like modems in terms of RS-232 specifications. That is because originally terminals attached themselves to modems that in turn talked over phone lines to mainframes. When cheap Microcomputers came on the scene it was logical to make them plug directly into the terminals.

The minimum lines needed to maintain communication are Transmit, Receive, Logic Ground, and Frame Ground. These are on a DB-25 connector on pins 3,2,7 and 1 respectively.

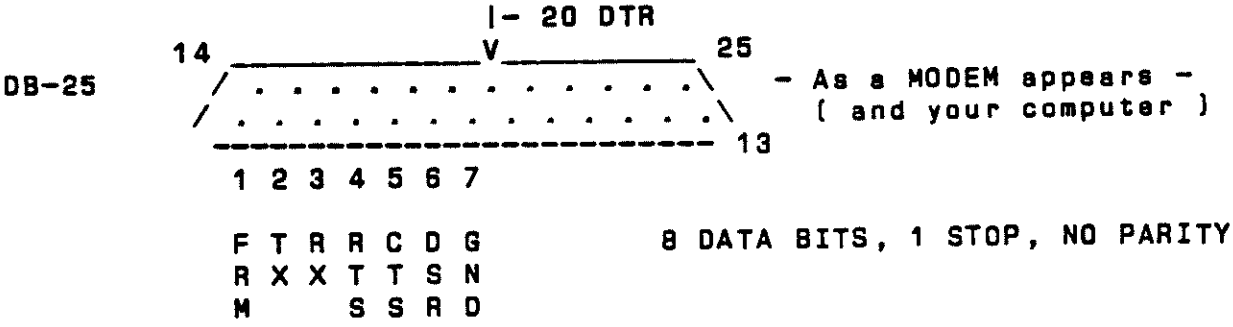
These terms only hold true if you are a modem. DB-25 connector pin assignments should always be referred to as if the modem was the subject. Therefore if pin 3 is "Transmit", if you are a terminal, you should receive on pin 3.

There are other signals on a DB-25 RS-232 serial connector that are standard. These signals were originally intended for modem use and their names reflect these intentions. They are; CTS (Clear to send), RTS (Request to send), DSR (Data set ready), DTR (Data terminal ready).

These "secondary signals" as they are called are used to inform the terminal user when the computer and transmission equipment can send data.

When attaching a printer, we use the principal signals to convey data in both directions and the secondary signals to "handshake". To handshake means to inform each other when you are prepared to communicate. It's rather like picking up the phone after hearing it ring and saying "Hello". Then the other person identifies himself, and so on. Humans "handshake" in many ways that are not readily apparent. Facing someone and nodding your head to start a conversation is "handshaking".

On our Paddle cards at the back of the computer are places for two resistors. They are called "pull up" resistors and must be installed if the serial port handshaking lines DSR and CTS are not to be used. These resistors can be anything from 1K to 10K. If buffer full handshaking is to be used, the correct line must be located which indicates that status on the printer you are using. The resistor "pulling up" the status line you wish to use must be removed (either can be used), and a jumper soldered between the RS-232 line and place where the resistor was. The other available handshaking line can be used for ON LINE/ OFF LINE status or ??. But for any transmission to take place both serial units must be at the same baud rate and both handshake lines must be acive (high).



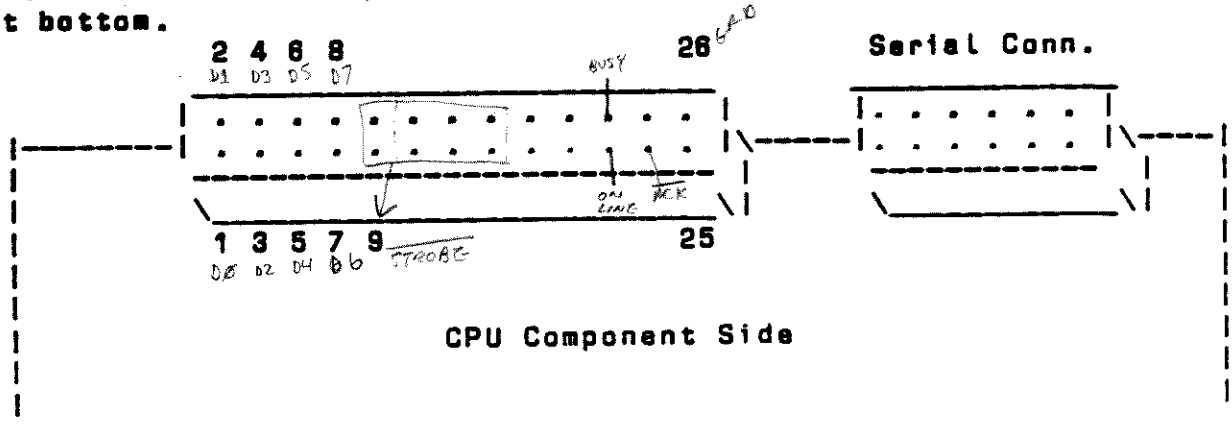
PARALLEL CABLE WIRING

This cable is used for Centronics interface printers. It uses a 8255, PA0-7 are used for data transfer, PB0-7 are tied together for data strobe PC6 for ACK from printer.

NOTE: This cable is available from your U S MICRO SALES distributor.

26 Pin CPU CABLE				36 Pin CENT. CONN.	
(1)	AD0	->		D0	(2)
(2)	AD1	->		D1	(3) P
(3)	AD2	->		D2	(4) R
8 (4)	AD3	->		D3	(5) I
2 (5)	AD4	->		D4	(6) N
5 (6)	AD5	->		D5	(7) T
5 (7)	AD6	->		D6	(8) E
(8)	AD7	->		D7	(9) R
(9-16)	BD0-7	->		STROBE	(1)
(21)	CD4	<-		ON LINE	(13)
(22)	CD5	<-		BUSY	(11)
(23)	CD6	<-		ACK	(10)
(26)	GND	<->		GND	(17)

CPU 8255 connector pinout looking from front with edge connector at bottom.



****
**
** DOUBLE DENSITY OPERATIONS
**
****

WITH A FEW EXCEPTIONS THE SAME OPERATIONS ARE AVAILABLE DOUBLE DENSITY AS ARE AVAILABLE SINGLE DENSITY. DOUBLE DENSITY MAY BE SELECTED EITHER BY ADDING A "D" TO THE DESIRED COMMAND OR BY PLACING THE CODE "DD" ANYWHERE ON THE COMMAND LINE. SOME EXAMPLES:

```
DVALID B:
VALID A: SIZE 512  DD

DCOPY A: TO B:  SIZE 256
DCOPY
COPY B: TO D: DD SIZE 512

DFORMAT
DFORMAT B: SIZE 512
FORMAT C: SIZE 256  DD
```

THE FOLLOWING EXCEPTIONS SHOULD BE NOTED. THE MAPPING FUNCTION IS NOT AVAILABLE DOUBLE DENSITY. IT IS DIFFICULT TO DO AN ACCURATE TRACK READ OF THE FORMAT ON A DOUBLE DENSITY DISK DUE TO THE FORMAT USED. EVEN WHERE NOT SPECIFICALLY PROHIBITED, IT IS RECOMMENDED THAT OPERATIONS REQUIRING TRACK READ SUCH AS COPY FORMAT OR COPY USING NOT BE ATTEMPED. PROBLEMS IN READING THE FORMAT FROM DISKS MAY CAUSE UNPREDICTABLE RESULTS.

THE SPECIAL FORMAT FUNCTION IS NOT AVAILABLE DOUBLE DENSITY. THE STANDARD FORMATS PROVIDED BY THE PROGRAM DOUBLE DENSITY ARE AS FOLLOWS:

128 BYTE SECTORS -

TRACK 0 STANDARD 3740 FORMAT SINGLE DENSITY. TRACKS 1-76 HAVE 51 SECTORS WRITTEN IN A 6 TO 1 INTERLACE PATTERN FOR USE WITH CP/M VERS 1.4. THE SECTOR ORDER IS AS FOLLOWS:

```
 1,18,35,10,27,44,2,19,36,11,28,45,3,20,37,12,29,46,4,21,38,13
,30,47,5,22,39,14,31,48,6,23,40,15,32,49,7,24,41,16,33,50,8,25,42
,17,34,51,9,26,43
```

ONE ADDITIONAL FEATURE IS PROVIDED FOR USE WITH CP/M. THE LAST BYTE OF DATA ON TRACK ZERO SECTOR ONE IS WRITTEN WITH A SPECIAL FLAG BYTE TO INDICATE THE FORMAT OF THE REST OF THE DISK. THE CODES ARE:

```
E5      SINGLE DENSITY  (E5 IS THE STANDARD FILL CHAR)
DD      DOUBLE DENSITY 51 SECTORS PER TRACK
4D      "QUAD" DOUBLE DENSITY DOUBLE SIDED DISK
```

TO INSERT THE "QUAD" CODE INTO SECTOR ONE OF A 128 BYTE SECTOR DISK THE FOLLOWING EXAMPLES MAY BE USED:

```
DFORMAT C:  QUAD
DFORMAT A:  QUAD
```

256 BYTE SECTORS -

DOUBLE DENSITY DISKS FORMATTED WITH 256 BYTE SECTORS ARE IN STANDARD IBM FORMAT. 26 SECTORS OF 256 BYTES. TRACK ZERO IS SINGLE DENSITY 3740 FORMAT.
THERE IS NO FLAG BYTE IN SECTOR 1 OF TRACK ZERO

512 BYTE SECTORS -

DOUBLE DENSITY DISKS FORMATTED WITH 512 BYTE SECTORS HAVE 16 SECTORS PER TRACK. TRACK ZERO IS STANDARD 3740 FORMAT. NO FLAG BYTE ON TRACK ZERO

THE OFFSET AND SKEW FUNCTIONS WORK AS BEFORE ALTHOUGH THE EFFECT OF FURTHER SKEWING THE ALREADY INTERLACED 128 BYTE PATTERN MAY BE CONFUSING.

COPYING DISKS DOUBLE DENSITY IS ALMOST EXACTLY THE SAME AS COPYING THEM SINGLE DENSITY. TWO SIMPLE RULES MUST BE OBSERVED.

1. BOTH DISKS MUST BE OF THE SAME DENSITY.
2. BOTH DISKS MUST HAVE THE SAME SECTOR SIZE.

A VIOLATION OF THES RULES MAY CAUSE THE CONTROLLER TO "HANG" REQUIRING THE COMPUTER TO BE RESET TO RECOVER.

IT WILL BE NOTED THAT COPY OPERATIONS ON DOUBLE DENSITY DISKS HAVING 51 SECTORS IS MUCH SLOWER THAN OTHER COPY OPERATIONS. THIS IS BECAUSE THE SECTORS ON THESE DISKS ARE WRITTEN SO CLOSE TOGETHER THAT THE 1791 CONTROLLER DOES NOT HAVE TIME TO WRITE CONSECUTIVE SECTORS ALTHOUGH IT IS ABLE TO READ CONSECUTIVE SECTORS IN THIS FORMAT. COPY OPERATIONS ON DISKS IN THIS FORMAT READ AND WRITE EVERY OTHER SECTOR THUS REQUIRING TWICE AS MANY DISK REVOLUTIONS AND TWICE THE TIME.

DOUBLE SIDED OR "QUAD" DISKS ARE NOT AUTOMATICALLY COPIED BY DFOCO IN A SINGLE OPERATION BUT REQUIRE TWO COPY OPERATIONS. TO COPY A "QUAD" DISK YOU MIGHT TYPE:

```
DCOPY A: TO C:          (COPY THE FRONT SIDE)
DCOPY B: TO D:          (COPY THE BACK SIDE)
```

****
**
** DOUBLE SIDED OPERATION **
**
****

WITH PRESENT DISK CONTROLLERS, A NUMBER OF DISK FORMATS ARE POSSIBLE. WHILE THE ORIGINAL IBM FLOPPY DISK FORMAT SPECIFIED DISKS FORMATTED SINGLE DENSITY, SINGLE SIDED WITH 26 SECTORS OF 128 BYTES PER TRACK, A LARGE NUMBER OF VARIATIONS ARE POSSIBLE. DISKS MAY BE FORMATTED DOUBLE DENSITY (WHICH DOUBLES THE AMOUNT OF DATA RECORDED ON A DISK), BOTH SIDES MAY BE USED, WHICH AGAIN DOUBLES THE AMOUNT OF DATA STORAGE, AND THE SECTOR SIZE MAY BE 128, 256, 512 OR 1024 BYTES. BECAUSE OF THE WAY CP/M USUALLY OPERATES, INCREASING THE SECTOR SIZE MARKEDLY INCREASES THE DATA TRANSFER RATE BOTH FOR THE SYSTEM AND MOST PROGRAMS. IN ORDER TO DISTINGUISH BETWEEN THE MANY DIFFERENT POSSIBLE FORMATS THE FOLLOWING CONVENTION HAS BEEN ADOPTED:

FORMAT CODES

REGARDLESS OF THE DISK FORMAT, TRACK ZERO SIDE 0 IS ALWAYS WRITTEN SINGLE DENSITY. A FORMAT CODE IS WRITTEN IN BYTE 7FH SECTOR 1 OF TRACK ZERO. THE FORMAT CODE SPECIFIES THREE PARAMETERS, SECTOR SIZE, DENSITY, AND WHETHER OR NOT THE DISK IS DOUBLE SIDED. THE BITS OF THE FORMAT CODE ARE ALLOCATED AS FOLLOWS:

BITS 0 - 1	SECTOR SIZE
BITS 4 - 5	DENSITY
BIT 7	DOUBLE SIDED

SINGLE SIDED CODES

	SINGLE DENSITY	DOUBLE DENSITY
128 BYTE SECTORS	20H	10H
256 BYTE SECTORS	21H	11H
512 BYTE SECTORS	22H	12H
1024 BYTE SECTORS	23H	13H

DOUBLE SIDED CODES

128 BYTE SECTORS	A0H	90H
256 BYTE SECTORS	A1H	91H
512 BYTE SECTORS	A2H	92H
1024 BYTE SECTORS	A3H	93H

****
**
**
**
**
****

DFOCO OPERATIONS

DFOCO SUPPORTS 3 BASIC OPERATIONS; 1.) FORMAT, 2.) VALIDATE, 3.) COPY. THESE OPERATIONS ARE BASICALLY THE SAME AS IN PREVIOUS VERSIONS. FOR EACH OPERATION A NUMBER OF MODIFIERS IS POSSIBLE.

THE KEY WORDS CONTROLLING THESE OPERATIONS ARE AS FOLLOWS:

NOTE: IN ALL EXAMPLES BELOW A:, B:, ETC. DESIGNATE THE DRIVE.

FORMAT ***
SIZE.....(SECTOR SIZE I.E., 128, 256, 512, ETC.)
OFFSET.....(A NUMBER YOU SPECIFY I.E., 3, 13, 5)
DD.....(DOUBLE DENSITY, ALSO MAY USE DFORMAT)
DBL.....(DOUBLE SIDED DISK)
TRACK 2-5.....(FOR EXAMPLE, ONLY TRACKS 2 THRU 5 ARE TO BE FORMATTED)

EXAMPLE: { FORMAT D: DD OFFSET 13 SIZE 512 DBL }

VALIDATE *** (CHECK TO SEE IF DISK RECORDING SURFACE INTACT)
RETRY.....(A NUMBER YOU SPECIFY OR DEFAULT IS 10)

EXAMPLE: { VALID C: RETRY 25 }

COPY ***
SIZE.....(SECTOR SIZE I.E., 128, 256, 512, ETC.)
DD.....(DOUBLE DENSITY, ALSO MAY USE DCOPY)
DBL.....(DOUBLE SIDED DISK)
TRACK.....(COPY JUST CERTAIN TRACKS I.E., 2-5)
RETRY.....(DEFAULT IS 10)
NOFILL.....(ON ERROR, DON'T FILL SECTOR WITH E5H)

EXAMPLE: { COPY A: TO D: RETRY 50 SIZE 1024 DD }

MAP*** (DISPLAY DISK FORMAT)

EXAMPLE: { MAP A: }

CODE*** (ENABLES WRITING FORMAT CODE ON DISK)

EXAMPLE: { CODE B: }

COMPARE*** (COMPARES 2 DISKS AND DISPLAYS VARYING SECTORS)

EXAMPLE: { COMPARE A: WITH B: }

****
**
** SINGLE DENSITY OPERATION **
**
****

VALIDATING DISKS

IT IS OFTEN USEFUL TO BE ABLE TO TELL WHETHER OR NOT THE RECORDING SURFACE OF A DISK IS COMPLETELY INTACT BEFORE USING THE DISK. THIS IS ESPECIALLY TRUE SINCE WHILE CP/M ALWAYS CHECKS FOR READ ERRORS THERE IS NO CHECK DONE FOR WRITE ERRORS. YOU WILL NOT KNOW YOU HAVE A BAD COPY OF A FILE UNTIL THE FIRST TIME YOU TRY TO READ IT AGAIN WHICH MAY BE DAYS OR EVEN MONTHS LATER.

VALIDATING A DISK CONSISTS SIMPLY OF READING EACH SECTOR ON THE DISK AND VERIFYING THAT THE CRC IS CORRECT. THIS SIMPLE PROCESS NOT ONLY DETECTS PHYSICAL DAMAGE TO THE MEDIA SUCH AS SCRATCHES BUT ALSO FORMATTING ERRORS SOMETIMES CAUSED BY NOISE TRANSIENTS, ERRORS WHICH WILL DISAPPEAR UPON REFORMATTING THE DISK. THE FOLLOWING ARE EXAMPLES OF VALIDATION COMMANDS:

```
VALID                          (DEFAULT IS DRIVE A:)
VALID C:
VALID D: RETRY 20
```

THE PROGRAM WILL READ THE ENTIRE DISK AND REPORT ANY READ ERRORS. IF A CRC ERROR OCCURS THE SECTOR WILL BE REREAD A MAXIMUM OF EITHER THE NUMBER OF RETRYS SPECIFIED OR 10 TIMES IF NO RETRY COUNT IS ENTERED. BOTH THE TOTAL NUMBER OF PERMANENT OR HARD ERROR AND RETRYS ARE REPORTED AT THE END OF THE VALIDATION PROCESS.

```
SUCCESSFUL VALIDATION DRIVE B:

OR

PERMANENT READ ERROR TRACK 17 SECTOR 5 DRIVE B
PERMANENT READ ERROR TRACK 42 SECTOR 19 DRIVE B
THERE WERE 27 RETRYS AND 2 HARD ERRORS
```

COPYING DISKS

ALL OR PART OF A DISK MAY BE COPIED BETWEEN ANY SPECIFIED DRIVES. THE FOLLOWING ARE EXAMPLES OF TYPICAL COPY COMMANDS:

```
COPY                           (DEFAULT IS A: TO B:)
COPY A: TO C:
COPY D: TO B: RETRY 20
COPY A: TO C: TRACK 0-1
COPY B: TO D: TRACK 3-20
COPY A: TO A:                  (SINGLE DISK TRANSFER)
```

DFOCO WILL RESPOND WITH AND ACKNOWLEDGMENT AND INSTRUCTIONS. FOR EXAMPLE:

```
COPYING DISK A TO DISK B
TYPE RETURN TO START
```

THE PROGRAM WILL HALT AND WAIT FOR DISKS TO BE CHANGED ETC. AND WILL BEGIN THE ACTUAL COPY OPERATION AFTER RETURN IS TYPED. IF YOU HAVE MADE AN ERROR AND WISH TO REENTER THE COMMAND TYPE CONTROL C. THE COPY OPERATION IS OPTIMIZED FOR SPEED. THE PROGRAM WILL DETERMINE THE MAXIMUM AMOUNT OF MEMORY AVAILABLE AND READ WHOLE TRACKS FROM THE SOURCE DISK UNTIL THE MEMORY IS FILLED. THEN THE ENTIRE MEMORY BUFFER IS WRITTEN ONTO THE DESTINATION DISK. ALL TRANSFERS ARE VERIFIED. AS EACH TRACK IS WRITTEN ONTO THE DESTINATION DISK IT IS READ BACK AND COMPARED BYTE BY BYTE WITH THE CONTENTS OF THE MEMORY BUFFER. ANY ERRORS ARE REPORTED.

IF THE SOURCE AND DESTINATION ARE THE SAME DFOCO WILL FIRST RESPOND WITH A REQUEST THAT YOU VERIFY THAT YOU REALLY WANT TO PERFORM A SINGLE DRIVE COPY. IF THE REQUEST IS ACKNOWLEDGED THE PROGRAM WILL INDICATE WHICH DISK TO MOUNT IN THE SPECIFIED DRIVE. AS WITH TWO DRIVE COPIES THE PROGRAM WILL FILL THE ENTIRE MEMORY BUFFER WITH DATA. THE PROGRAM WILL THEN HALT AND NOTIFY YOU TO CHANGE DISKS. THE NUMBER OF DISK CHANGES REQUIRED WILL, OF COURSE, DEPEND UPON THE AMOUNT OF MEMORY AVAILABLE. WITH A 48K SYSTEM 8 SWAPS ARE REQUIRED, WITH A 64K SYSTEM ONLY 6 ARE NEEDED.

COPYING DISKS WITH DIFFERENT SIZE SECTORS

DFOCO WILL COPY DISKS WITH SECTORS OF 128, 256 OR 512 BYTES. THE PRIMARY LIMITATION OF THE PROGRAM IS THAT BOTH DISKS MUST HAVE THE SAME SIZE SECTORS. THE SIZE SHOULD BE SPECIFIED IN THE COPY COMMAND.

```
COPY A: TO C:  SIZE 256
COPY B: TO A:  SIZE 512
COPY C:  TO D:  SIZE 128
```

SPECIFYING A SECTOR SIZE OF OTHER THAN 128, 256 OR 512 BYTES WILL GENERATE AN ERROR MESSAGE.

ERROR HANDLING DURING COPY OPERATIONS

ERROR HANDLING AND RECOVERY IS AN EXTREMELY INPORTANT PART OF ANY DISK FILE MANAGEMENT SYSTEM. DFOCO IS DESIGNED TO PERMIT THE MAXIMUM RECOVERY OF DATA FROM DAMAGED OR "CRASHED" DISKS. TWO TYPES OF ERRORS OCCUR DURING COPY OPERATIONS, ERRORS IN READING FROM THE SOURCE DISK AND ERRORS WRITING TO THE DESTINATION DISK. READ ERRORS ARE BY FAR THE MOST COMMON.

IF A READ ERROR OCCURS DFOCO WILL ATTEMPT TO REREAD THE SECTOR 10 (OR RETRY) TIMES. IF THE READ ERROR PERSISTS ONE OF TWO ACTIONS MAY BE CHOSEN. THE DEFAULT IS TO FILL THE SECTOR WITH E5H AND WRITE IT ON THE DESTINATION DISK. THE OTHER CHOICE IS SIMPLY TO ACCEPT THE DATA AS READ IN AND WRITE IT ON THE DESTINATION DISK. THIS IS THE NOFILL OPTION. NOFILL PERMITS THE MAXIMUM DATA RECOVERY BUT REQUIRES CAREFUL INSPECTION OF THE SECTOR CAUSING THE ERROR SINCE THE ERRORS MAY NOT BE OBVIOUS.

WRITE ERRORS ARE HANDLED IN A SOMEWHAT DIFFERENT FASHION. SINCE THERE IS USUALLY NO VALUABLE DATA ON THE DESTINATION DISK, THE NORMAL CHOICE IS TO NOT PERMIT COPY OPERATIONS IF WRITE ERROR OCCUR. DFOCO WILL ATTEMPT TO WRITE A SECTOR 10 (OR RETRY) TIMES. IF THE ERROR PERSISTS, DFOCO WILL STOP THE COPY OPERATION AND VALIDATE THE DESTINATION DISK THUS REPORTING ALL BAD SECTORS. IT IS POSSIBLE TO COPY IN THE PRESENCE OF WRITE ERRORS BY TURNING OFF THE WRITE VERIFICATION, THE NOVERIFY OPTION. HOWEVER SINCE DISKS HAVE BECOME RELATIVELY INEXPENSIVE THIS OPTION IS PROBABLY UNWISE EXCEPT IN VERY SPECIAL CASES. SOME EXAMPLES ARE AS FOLLOWS:

```
COPY B: TO C: NOFILL   RETRY 25
COPY A: TO B: TRACK 0-5 NOVERIFY
COPY D: TO A: NOVERIFY NOFILL
```

DISK FORMATTING AND MAPPING

THE STANDARD IBM FORMAT FOR 8 INCH FLOPPY DISKS IS GIVEN IN THE WESTERN DIGITAL DOCUMENTATION FOR THE 1791 DISK CONROLLER CHIP AND IN VARIOUS IBM DOCUMENTS. THE USUAL FORMAT IS "SOFT SECTORED". THIS MEANS ESSENTIALLY THAT THE TRACK AND SECTOR NUMBERS ARE ACTUALLY WRITTEN ON THE DISK AS DATA RATHER THAN BEING DETERMINED BY THE PRESENCE OF PHYSICAL INDICATORS SUCH AS HOLES IN THE DISK. FORMATTING A DISK CONSISTS OF WRITING BOTH THE TRACK AND SECTOR NUMBERS AS WELL AS CLOCKING INFORMATION FOR THE CONTROLER ON THE DISK. IT IS IMPORTANT TO REMEMBER THAT A "BLANK" DISK IS BY NO MEANS REALLY BLANK. RATHER, THE DISK CONTAINS A GREAT DEAL OF FORMATTING INFORMATION WITHOUT WHICH THE CONTROLLER IS UNABLE TO READ IT.

BECAUSE OF THIS "SOFT SECTORING" IT IS SOMETIMES POSSIBLE AND OFTEN USEFUL TO CHANGE THE FORMAT TO ALLOW INCREASED AMOUNTS OF DATA TO BE WRITTEN ON THE DISK OR HIGHER SPEED OF OPERATION. THERE ARE TWO POSSIBLE CHANGES THAT ARE USEFUL, CHANGING THE SIZE OF THE SECTORS OR CHANGING THEIR ORDERING.

THIS VERSION OF DFOCO SUPPORTS THREE DIFFERENT SECTOR SIZES USING THE "IBM" SOFT SECTOR FORMAT. SECTORS MAY BE FORMATED WITH THE NORMAL 26 SECTORS OF 128 BYTES OR WITH 16 SECTORS OF 256 BYTES OR 8 SECTORS OF 512 BYTES EACH. USING 256 OR 512 BYTE SECTORS ALLOWS APPROXIMATELY 20% MORE DATA TO BE WRITTEN ON A SINGLE DISK, HOWEVER THERE ARE AT PRESENT FEW PROGRAMS WHICH WILL SUPPORT THE USE OF LARGER SECTORS. DFOCO ALSO SUPPORTS A WIDE VARIETY OF SECTOR ORDERINGS.

THE DESIGN OF CP/M DATES FROM A TIME WHEN DISK CONTROLLERS WERE QUITE SLOW AND COMPUTER MEMORIES SMALL AND VERY COSTLY. SMALL MEMORIES DICTATED SMALL SECTOR SIZES ON DISKS SINCE THE LARGER THE SECTORS THE LARGER THE MEMORY BUFFERS REQUIRED. SLOW CONTROLERS MEANT THAT HAVING READ A SECTOR FROM THE DISK IT WAS NECESSARY TO WAIT BEFORE ANOTHER SECTOR COULD BE READ. DELAYS OF 5 SECTOR TIMES (ABOUT 25 MSEC) WERE COMMON. THUS CP/M IS SET UP TO READ EVERY 6TH SECTOR AROUND THE DISK. THIS STRATEGY UNFORTUNATELY IS FAR FROM OPTIMAL FOR PRESENT DAY CONTROLLERS WHICH CAN READ CONSECUTIVE SECTORS FROM A DISK WITH EASE. NOTE THAT DFOCO WHICH READS AN ENTIRE TRACK IN A SINGLE DISK REVOLUTION IS OVER 5 TIMES AS FAST AS PIP. UNFORTUNATELY CP/M STANDARD SYSTEM PROGRAMS SUCH AS PIP AND THE ASSEMBLER CAN ONLY BE SPEEDED UP A SMALL AMOUNT, ABOUT 20 PER CENT, BY CHANGING DISK FORMATS ALONE. HOWEVER, NEW PROGRAMS WRITTEN TO TAKE ADVANTAGE OF FASTER CONTROLLERS CAN BE SPEEDED UP A GREAT DEAL MORE.

MAPPING DISK FORMATS

WHEN EXPERIMENTING WITH NONSTANDARD FORMATS IT IS OFTEN VERY USEFUL TO BE ABLE TO READ AND DISPLAY THE ACTUAL DISK FORMAT. YOU CAN'T NECESSARILY TELL WHAT'S ON A DISK BY LOOKING AT THE LABEL. THE MAP COMMAND READS THE FORMAT FROM A SINGLE TRACK ON A SPECIFIED DISK AND DISPLAYS IT. THERE ARE 26 SECTORS PER TRACK IN THE STANDARD IBM FORMAT WHICH ARE NUMBERED IN SEQUENTIAL ORDER. TO DISPLAY THE SECTOR ORDERING TYPE:

```
MAP                          (DEFAULT IS TRACK 0 DRIVE A:)
MAP C: TRACK 76
```

THE PROGRAM WILL READ THE SPECIFIED TRACK AND DISPLAY THE PHYSICAL TO LOGICAL SECTOR MAPPING. THE PHYSICAL SECTORS STARTING FROM THE SINGLE INDEX HOLE IN THE DISK ARE SIMPLY NUMBERED 1 THRU 26. THE CORRESPONDING LOGICAL SECTORS ACTUALLY WRITTEN ON THE DISK ARE DISPLAYED BESIDE THE PHYSICAL SECTOR NUMBER. FOR DISKS FORMATTED WITH LESS THAN 26 SECTORS, THE UNUSED LOGICAL SECTOR NUMBERS DISPLAY AS A '-'.

OCCASIONALLY DFOCO WILL DISPLAY OBVIOUSLY INCORRECT MAPPING DATA, FOR EXAMPLE TRACK 404 SECTOR NUMBER 201. THIS MEANS THE FORMAT ON THE DISK IS INCORRECT. THE DISK CONTROLLER WILL OFTEN READ THESE DISKS CORRECTLY BUT IT IS USUALLY A GOOD IDEA TO COPY THE DATA TO A CORRECTLY FORMATTED DISK. FORMATTING PROBLEMS OF THIS TYPE OFTEN SHOW UP WHEN YOU ATTEMPT TO READ DISKS PRODUCED ON ANOTHER COMPUTER SYSTEM. IF THE HEAD ALLIGNMENT IF ONLY SLIGHTLY DIFFERENT FROM YOURS YOU MAY GET MAPPING ERRORS EVEN THOUGH YOU ARE USUALLY ABLE TO READ THE DATA CORRECTLY.

****
** **
** FORMATTING DISKS **
** **
****

DFOCO PERMITS FORMATTING DISKS ON ANY DRIVE SUPPORTED BY CP/M. EITHER THE ENTIRE DISK OR SPECIFIED TRACKS MAY BE FORMATTED. IT IS EVEN POSSIBLE TO WRITE DIFFERENT FORMATS ON DIFFERENT TRACKS OF THE SAME DISK.

STANDARD FORMAT

TO WRITE THE STANDARD IBM FORMAT ON A DISK TYPE

```
FORMAT                              (DEFAULT IS DRIVE A)
FORMAT B:
FORMAT C:  TRACK  20-40
```

THE PROGRAM WILL HALT AND THEN RESPOND WITH:

```
        STANDARD IBM 3740 FORMAT

        INSERT DISK TO BE FORMATTED IN DRIVE A
        TYPE CARRIAGE RETURN
```

NON STANDARD FORMATS MAY HAVE ALMOST ANY FORM YOU SPECIFY. THE FIRST VARIATION IS TO OFFSET THE SAME FORMAT FROM TRACK TO TRACK. THIS IS USEFUL TO COMPENSATE FOR THE TIME IT TAKES TO STEP THE HEAD FROM ONE TRACK TO ANOTHER AND IS ONE OF THE TECHNIQUES USED IN DFOCO TO INCREASE THE COPY SPEED. THE FOLLOWING IS AN EXAMPLE OF TRACK OFFSETTING:

```
        FORMAT B: OFFSET 5
```

THIS RESULTS IN THE FOLLOWING FORMAT ON THE DISK

	TRACK 0	TRACK 1	TRACK 2	ETC.
SECTOR	1	6	11	
SECTOR	2	7	12	
SECTOR	3	8	13	
SECTOR	4	9	14	

A SECOND FORMATTING VARIATION IS TO SKEW THE SECTORS BY A CONSTANT AMOUNT. THIS CAN BE SPECIFIED AS FOLLOWS.

FORMAT B: SKEW 3

THIS WILL RESULT IN A DISPLAY OF THE PHYSICAL TO LOGICAL SECTOR MAPPING AND PERMIT CHANGING THE SPECIFICATIONS BEFORE WRITING THE FORMAT ON THE DISK.

PHYSICAL SECTOR	LOGICAL SECTOR	PHYSICAL SECTOR	LOGICAL SECTOR
1	1	14	14
2	4	15	17
3	7	16	20
4	10	17	23
5	13	18	26
6	16	19	3
7	19	20	6
8	22	21	9
9	25	22	12
10	2	23	15
11	5	24	18
12	8	25	21
13	11	26	24

TYPE RETURN TO FORMAT, SECTOR NO TO CORRECT

TYPING A SECTOR NUMBER ALLOWS THE LOGICAL SECTOR NUMBER TO BE CHANGED. BEFORE USING A SECTOR NUMBER IT MUST BE FIRST SET TO ZERO SINCE THE PROGRAM CHECKS AND DOES NOT PERMIT TWO SECTORS WITH THE SAME NUMBER. THE SECTOR MAPPING IS REDISPLAYED FOR VERIFICATION AFTER EACH CHANGE.

THE FINAL FORMATTING OPTION IS SIMPLY TO TYPE IN THE PHYSICAL TO LOGICAL SECTOR MAPPING FOR EACH SECTOR. TO SELECT THIS OPTION TYPE:

SPECIAL FORMAT A:

THE PROGRAM WILL RESPOND BY DISPLAYING EACH PHYSICAL SECTOR NUMBER AND WAITING FOR THE CORRESPONDING LOGICAL SECTOR NUMBER TO BE ENTERED. AGAIN THE PROGRAM CHECKS THE SECTOR NUMBERS AS ENTERED AND WILL NOT ALLOW THE SAME SECTOR NUMBER TO BE USED TWICE.

NOTE THAT THE VARIOUS OPTIONS MAY BE COMBINED IF DESIRED.

SPECIAL FORMAT B: TRACK 0-1
FORMAT C: OFFSET 6 SKEW 3 TRACK 10-76

IT IS EVEN POSSIBLE TO COPY THE FORMAT FROM ONE DISK TO ANOTHER.

COPY FORMAT A: TO B:
COPY FORMAT B: TO D: TRACK 10

FORMATTING WITH DIFFERENT SIZE SECTORS

THE DEFAULT SECTOR SIZE GENERATED BY DFOCO IS 128 BYTES, HOWEVER THE PROGRAM WILL ALSO FORMAT TRACKS WITH 16 SECTORS OF 256 BYTES OR 8 SECTORS OF 512 BYTES. THE SECTOR SIZE IS SPECIFIED BY THE SIZE PARAMETER.

```
FORMAT B: SIZE 512
SPECIAL FORMAT A:  SIZE 256
FORMAT C: OFFSET 2 SIZE 512
```

CAUTION MUST BE USED WITH THE SPECIAL FORMAT OPTION SINCE DFOCO WILL ALLOW SECTOR NUMBERS GREATER THAN THE NUMBER OF SECTORS ON A TRACK. THE 1791 WILL ACTUALLY READ SECTORS NUMBERED IN THIS FASHION. FOR EXAMPLE A TRACK MAY BE FORMATTED WITH 8 512 BYTE SECTORS NUMBERED 11 THRU 18, HOWEVER DISKS WRITTEN IN THIS FASHION MAY NOT VALIDATE CORRECTLY. IT IS ALSO POSSIBLE TO FORMAT A DISK WITH DIFFERENT SIZE SECTORS ON DIFFERENT TRACKS. AGAIN, DISKS WRITTEN IN THIS FASHION MAY NOT VALIDATE CORRECTLY.

REGARDLESS OF THE SECTOR SIZE AND MAPPING CHOSEN, TRACK 0 OF THE DISK IS ALWAYS WRITTEN IN THE STANDARD IBM 3740 FORMAT. THIS IS DONE TO FACILITATE IDENTIFICATION OF THE DISK FORMAT BY A PROGRAM. READ AND WRITE OPERATIONS TO A DISK MAY ACTUALLY BE IMPOSSIBLE AND "HANG" THE CONTROLLER IF THE PROGRAM EXPECTS A FORMAT THAT IS NOT PRESENT. THIS IS ESPECIALLY TRUE IF THE INCORRECT DENSITY IS SELECTED.

PROGRAM TIMING

THE FOLLOWING TIMING FIGURES ARE TYPICAL OF A 64K SYSTEM AND WILL BE SLIGHTLY HIGHER FOR SMALLER SYSTEMS. THE COPY TIMINGS VARY WITH DISK FORMATS. IF THE FORMAT IS NON STANDARD BUT THE SAME ON BOTH DISKS THE TIMES ARE THE SAME AS FOR STANDARD FORMATS BUT IF THE SECTOR FORMATS ARE DIFFERENT ON THE TWO DISKS THE COPY TIMES WILL BE INCREASED. THE FOLLOWING TIMINGS ARE FOR SINGLE DENSITY ONLY.

VALIDATING	17 SEC	
FORMATTING	43 SEC	(INCLUDES VALIDATION)
COPY SAME FORMAT	46 SEC	
COPY DIFFERENT FORMAT	90 SEC	AVERAGE
	300 SEC	WORST CASE

IF THE FORMAT IS DIFFERENT ON DIFFERENT TRACKS OF A DISK IT MAY BE POSSIBLE TO INCREASE COPY SPEED WITH THE USING OPTION. SINCE A WRITE OPERATION TAKES TWICE AS LONG AS A READ OPERATION THE PROGRAM CAN OPTIMIZE THE COPY BY READING THE TRACK FORMAT FROM THE DESTINATION DISK AND USING IT TO CONTROL READING AND WRITING. A SAMPLE COPY WITH THIS OPTION IS

```
COPY A: TO B: USING 3
```

THIS CAUSES TRACK 3 TO BE READ FROM THE DESTINATION DISK AND USED TO CONTROL THE COPY OPERATION. THIS CAN OFTEN DOUBLE THE COPY SPEED IF THE FORMATS ARE DIFFERENT ON THE TWO DISKS.

NEC
NEC Electronics U.S.A. Inc.
Microcomputer Division

PROGRAMMABLE COMMUNICATION INTERFACES

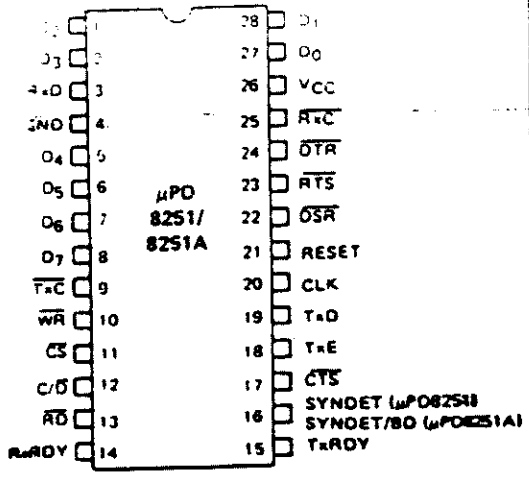
DESCRIPTION

The μPD8251 and μPD8251A Universal Synchronous/Asynchronous Receiver/Transmitters (USARTs) are designed for microcomputer systems data communications. The USART is used as a peripheral and is programmed by the 8080A or other processor to communicate in commonly used serial data transmission techniques including IBM Bi-Sync. The USART receives serial data streams and converts them into parallel data characters for the processor. While receiving serial data, the USART will also accept data characters from the processor in parallel format, convert them to serial format and transmit. The USART will signal the processor when it has completely received or transmitted a character and requires service. Complete USART status including data format errors and control signals such as TxE and SYNDET, is available to the processor at any time.

FEATURES

- Asynchronous or Synchronous Operation
  - Asynchronous:
    - Five 8-Bit Characters
    - Clock Rate – 1, 16 or 64 x Baud Rate
    - Break Character Generation
    - Select 1, 1-1/2, or 2 Stop Bits
    - False Start Bit Detector
    - Automatic Break Detect and Handling (μPD8251A)
  - Synchronous:
    - Five 8-Bit Characters
    - Internal or External Character Synchronization
    - Automatic Sync Insertion
    - Single or Double Sync Characters
- Baud Rate (1X Mode) – DC to 56K Baud (μPD8251)
  – DC to 64K Baud (μPD8251A)
- Full Duplex, Double Buffered Transmitter and Receiver
- Parity, Overrun and Framing Flags
- Fully Compatible with 8080A/8085/μPD780 (Z80™)
- All Inputs and Outputs are TTL Compatible
- Single ±5 Volt Supply, ±10% (8251A), ±5% (8251)
- Separate Device Receive and Transmit TTL Clocks
- 28 Pin Plastic DIP Package
- N-Channel MOS Technology

PIN CONFIGURATION



PIN NAMES

D_7-D_0	Data Bus (8 bits)
C/D	Control or Data is to be Written or Read
RD	Read Data Command
WR	Write Data or Control Command
CS	Chip Enable
CLK	Clock Pulse (TTL)
RESET	Reset
TxC	Transmitter Clock (TTL)
TxD	Transmitter Data
RxC	Receiver Clock (TTL)
RxD	Receiver Data
RxRDY	Receiver Ready (has character for 8080A)
TxRDY	Transmitter Ready (ready for char. from 8080A)
DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDET	Sync Detect
SYNDET/BD	Sync Detect/Break Detect
RTS	Request to Send Data
CTS	Clear to Send Data
TxE	Transmitter Empty
V_{CC}	+5 Volt Supply
GND	Ground

TM: Z80 is a registered trademark of Zilog, Inc.

FUNCTIONAL DESCRIPTION

The μPD8251 and μPD8251A Universal Synchronous/Asynchronous Receiver/Transmitters are designed specifically for 8080 microcomputer systems but work with most 8-bit processors. Operation of the μPD8251 and μPD8251A, like other I/O devices in the 8080 family, are programmed by system software for maximum flexibility.

In the receive mode, the μPD8251 or μPD8251A converts incoming serial format data into parallel data and makes certain format checks. In the transmit mode, it formats parallel data into serial form. The device also supplies or removes characters or bits that are unique to the communication format in use. By performing conversion and formatting services automatically, the USART appears to the processor as a simple or "transparent" input or output of byte-oriented parallel data.

μPD8251A FEATURES AND ENHANCEMENTS

The μPD8251A is an advanced design of the industry standard 8251 USART. It operates with a wide range of microprocessors, including the 8080, 8085, and μPD780 (Z80™). The additional features and enhancements of the μPD8251A over the μPD8251 are listed below.

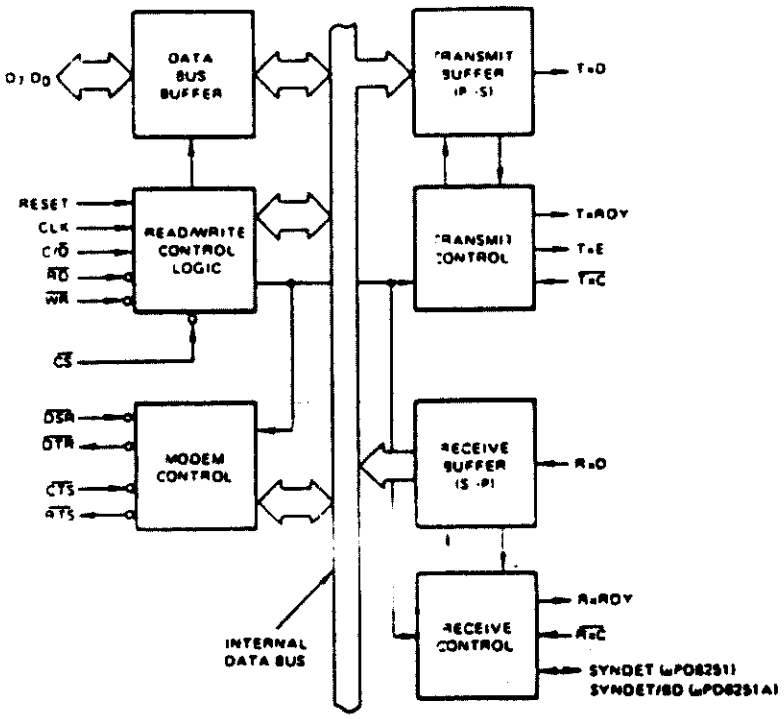
1. The data paths are double-buffered with separate I/O registers for control, status, Data In and Data Out. This feature simplifies control programming and minimizes processor overhead.
2. The Receiver detects and handles "break" automatically in asynchronous operations, which relieves the processor of this task.
3. The Receiver is prevented from starting when in "break" state by a refined Rx initialization. This also prevents a disconnected USART from causing unwanted interrupts.
4. When a transmission is concluded the TxD line will always return to the marking state unless SBRK is programmed.
5. The Tx Disable command is prevented from halting transmission by the Tx Enable Logic enhancement, until all data previously written has been transmitted. The same logic also prevents the transmitter from turning off in the middle of a word.
6. Internal Sync Detect is disabled when External Sync Detect is programmed. An External Sync Detect Status is provided through a flip-flop which clears itself upon a status read.
7. The possibility of a false sync detect is minimized by:
   - ensuring that if a double sync character is programmed, the characters be contiguously detected.
   - clearing the Rx register to all Logic 1s (V_{OH}) whenever the Enter Hunt command is issued in Sync mode.
8. The $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device as long as the μPD8251A is not selected.
9. The μPD8251A Status can be read at any time, however, the status update will be inhibited during status read.
10. The μPD8251A has enhanced AC and DC characteristics and is free from extraneous glitches, providing higher speed and improved operating margins.
11. Baud rate from DC to 64K.

BASIC OPERATION

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	μPD8251/μPD8251A → Data Bus
0	1	0	0	Data Bus → μPD8251/μPD8251A
1	0	1	0	Status → Data Bus
1	1	0	0	Data Bus → Control
X	X	X	1	Data Bus → 3-State
X	1	1	0	

™: Z80 is a registered trademark of Zilog, Inc.

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Operating Temperature	0°C to +70°C
Storage Temperature	-65°C to +150°C
All Output Voltages	-0.5 to +7 Volts
All Input Voltages	-0.5 to +7 Volts
Supply Voltages	-0.5 to +7 Volts

$T_a = 25°C$

*COMMENT: Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

$T_a = 0°C$ to $70°C$; $V_{CC} = 5.0V \pm 10\%$ for 8251A and $\pm 5\%$ for 8251; GND = 0V

PARAMETER	SYMBOL	LIMITS μPD8251 MIN	TYP	MAX	LIMITS μPD8251A MIN	MAX	UNIT	TEST CONDITIONS
Input Low Voltage	V_{IL}	-0.5		0.8	0.5	0.8	V	
Input High Voltage	V_{IH}	2.0		V_{CC}	2.2	V_{CC}	V	
Output Low Voltage	V_{OL}			0.45		0.45	V	μPD8251 I_{OL} = 1.7 mA μPD8251A I_{OL} = 2.2 mA
Output High Voltage	V_{OH}	2.4			2.4		V	μPD8251: I_{OH} = -100 μA μPD8251A: I_{OH} = -400 μA
Data Bus Leakage	I_{DL}			-50		-10	μA	V_{OUT} = 0.45V
				10		10		$V_{OUT} = V_{CC}$
Input Load Current	I_{IL}			10		10	μA	At 5.5V
Power Supply Current	I_{CC}		45	90		100	mA	μPD8251A: All Outputs = Logic 1

CAPACITANCE

$T_a = 25°C$; $V_{CC} = GND = 0V$

PARAMETER	SYMBOL	LIMITS MIN	TYP	MAX	UNIT	TEST CONDITIONS
Input Capacitance	C_{IN}			10	pF	fc = 1 MHz
I/O Capacitance	$C_{I/O}$			20	pF	Unmeasured pins returned to GND

T_A = 0°C to 70°C; V_{CC} = 5.0V ± 10% for 8251A; GND = 0V; V_{CC} = 5.0V ± 5% for 8251

PARAMETER	SYMBOL	LIMITS μPD8251 MIN	LIMITS μPD8251 MAX	LIMITS μPD8251A MIN	LIMITS μPD8251A MAX	UNIT	TEST CONDITIONS
READ							
Address Stable before READ (CS, C/D)	t_{AR}	50		50		ns	
Address Hold Time for READ (CS, C/D)	t_{RA}	5		50		ns	
READ Pulse Width	t_{RR}	430		250		ns	
Data Delay from READ	t_{RD}		350		250	ns	μPD8251: C_L = 100 pF μPD8251A: C_L = 150 pF
READ to Data Floating	t_{DF}	25	200	10	100	ns	μPD8251 C_L = 100 pF C_L = 15 pF
WRITE							
Address Stable before WRITE	t_{AW}	20		50		ns	
Address Hold Time for WRITE	t_{WA}	20		50		ns	
WRITE Pulse Width	t_{WW}	400		250		ns	
Data Set-Up Time for WRITE	t_{DW}	200		150		ns	
Data Hold Time for WRITE	t_{WD}	40		30		ns	
Recovery Time Between WRITES (2)	t_{RV}	6		6		t_{CY}	
OTHER TIMING							
Clock Period (3)	t_{CY}	0.422	1.35	0.32	1.35	μs	
Clock Pulse Width High	t_{pW}	220	0.7t_{CY}	140	t_{CY}-90	ns	
Clock Pulse Width Low	t_{pW}			90		ns	
Clock Rise and Fall Time	$t_{R,F}$	0	50	5	20	ns	
TxD Delay from Falling Edge of TxC	t_{DT_x}		1		1	μs	
R_x Data Set-Up Time to Sampling Pulse	t_{SR_x}	2		2		μs	μPD8251 C_L = 100 pF
R_x Data Hold Time to Sampling Pulse	t_{HR_x}	2		2		μs	
Transmitter Input Clock Frequency 1X Baud Rate 16X Baud Rate 64X Baud Rate	f_{T_x}	DC DC DC	56 520 520		64 310 615	kHz kHz kHz	
Transmitter Input Clock Pulse Width 1X Baud Rate 16X and 64X Baud Rate	t_{TPW}	12 1		12 1		t_{CY} t_{CY}	
Transmitter Input Clock Pulse Delay 1X Baud Rate 16X and 64X Baud Rate	t_{TPD}	15 3		15 3		t_{CY} t_{CY}	
Receiver Input Clock Frequency 1X Baud Rate 16X Baud Rate 64X Baud Rate	f_{R_x}	DC DC DC	56 520 520		64 310 615	kHz kHz kHz	
Receiver Input Clock Pulse Width 1X Baud Rate 16X and 64X Baud Rate	t_{RPW}	12 1		12 1		t_{CY} t_{CY}	
Receiver Input Clock Pulse Delay 1X Baud Rate 16X and 64X Baud Rate	t_{RPD}	15 3		15 3		t_{CY} t_{CY}	
TxRDY Delay from Center of Data Bit	t_{TR}		16		8	t_{CY}	μPD8251 C_L = 50 pF
RxRDY Delay from Center of Data Bit	t_{RR}		20		24	t_{CY}	
Internal SYNDET Delay from Center of Data Bit	t_{IS}		25		24	t_{CY}	
External SYNDET Set-Up Time before Falling Edge of RxC	t_{ES}	16		16		t_{CY}	
TxEMPTY Delay from Center of Data Bit	t_{TxE}		16		20	t_{CY}	μPD8251: C_L = 50 pF
Control Delay from Rising Edge of WRITE (TxE, DTR, RTS)	t_{WC}		16		8	t_{CY}	
Control to READ Set-Up Time (DSR, CTS)	t_{CR}	16		20		t_{CY}	

Notes:
① AC timings measured at V_{OH} = 2.0, V_{OL} = 0.8, and with load circuit of Figure 1
② This recovery time is for initialization only; when MODE, SYNC1, SYNC2, COMMAND and first DATA BYTES are written into the USART. Subsequent writing of both COMMAND and DATA are only allowed when TxRDY = 1
③ The TxC and RxC frequencies have the following limitations with respect to CLK:
For 1X Baud Rate, f_{T_x} or $f_{R_x} \leq 1/(30\ t_{CY})$
For 16X and 64X Baud Rate, f_{T_x} or $f_{R_x} \leq 1/(4.5\ t_{CY})$
④ Reset Pulse Width = 6 t_{CY} minimum.
⑤ $T_{TxRDYCLR} = 2t_{CY} + T_D + T_R + 200ns$
⑥ $T_{RxRDYCLR} = 2t_{CY} + T_D + T_R + 170ns$

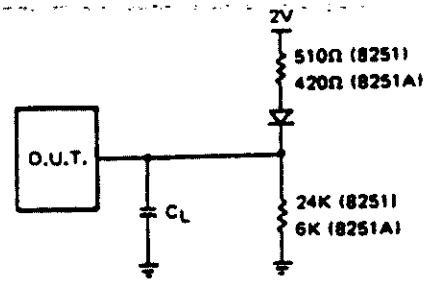
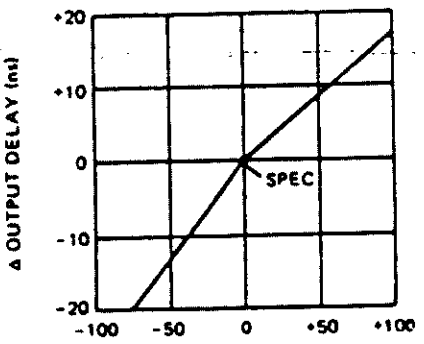


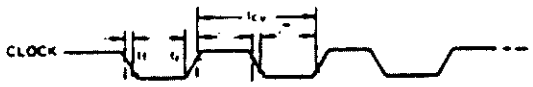
Figure 1.



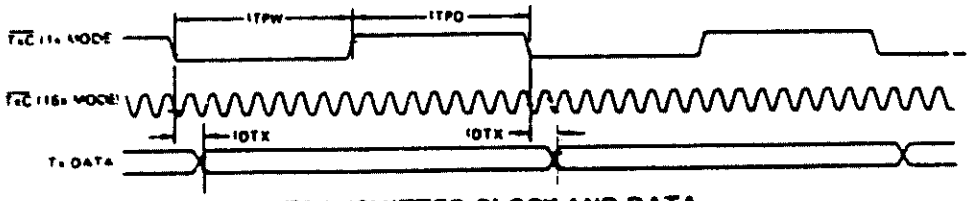
Typical Δ Output Delay Versus Δ Capacitance (pF)

TEST LOAD CIRCUIT

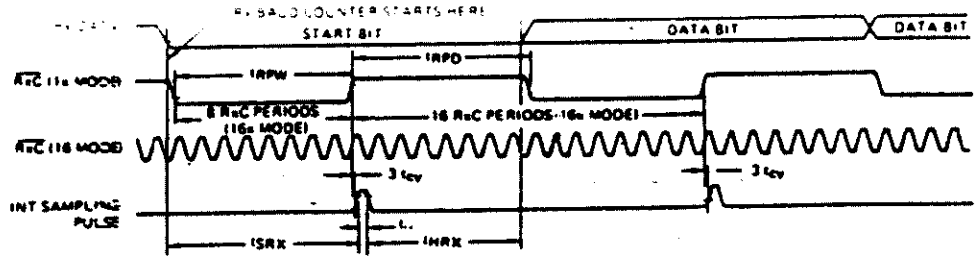
TIMING WAVEFORMS



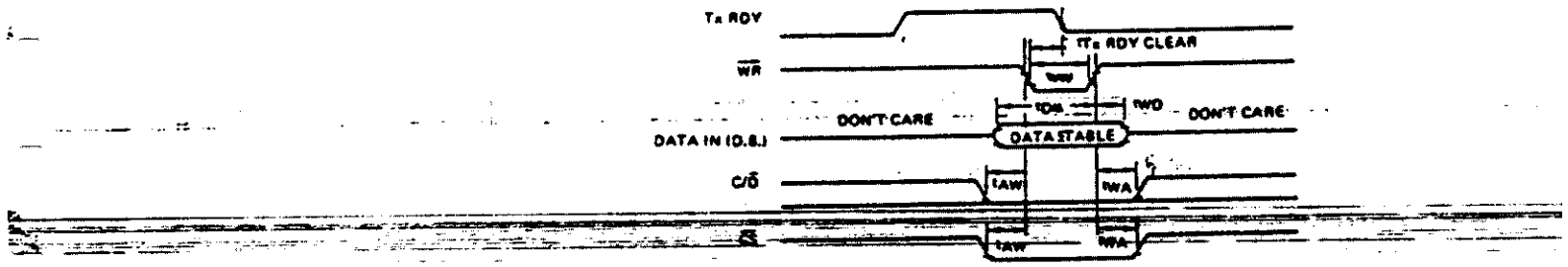
SYSTEM CLOCK INPUT



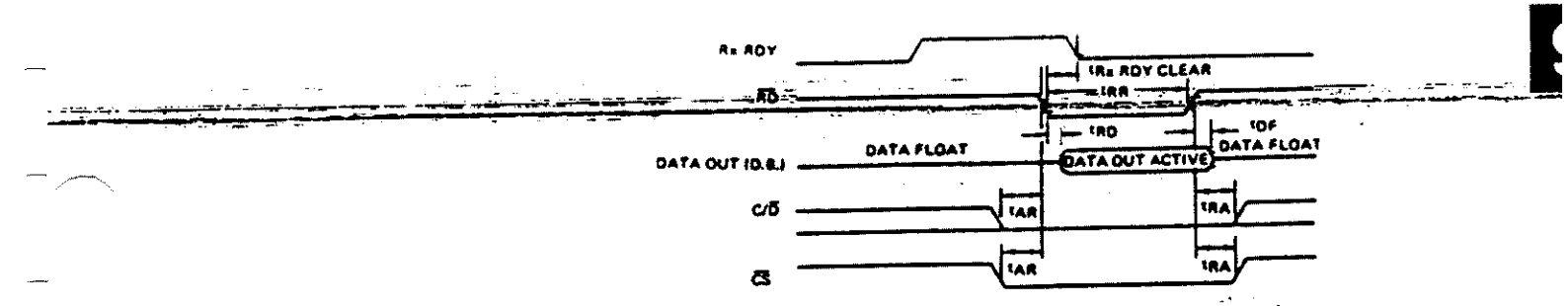
TRANSMITTER CLOCK AND DATA



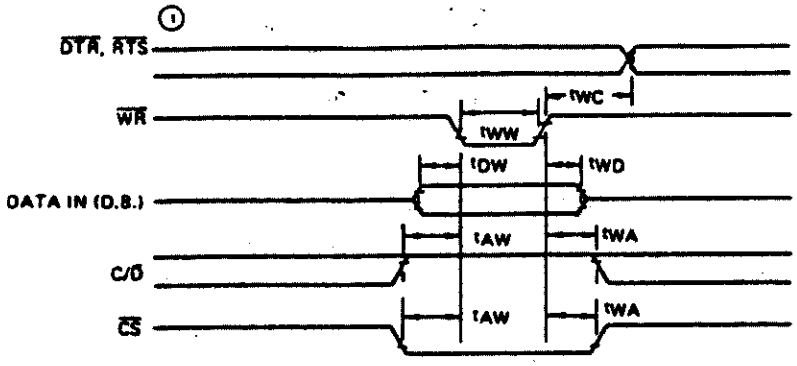
RECEIVER CLOCK AND DATA



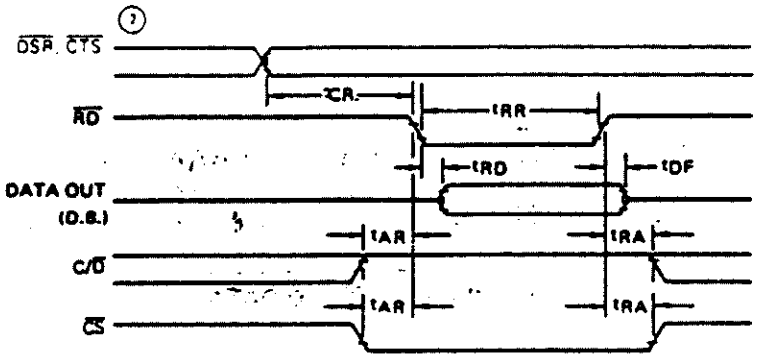
WRITE DATA CYCLE (PROCESSOR → USART)



READ DATA CYCLE (PROCESSOR ← USART)



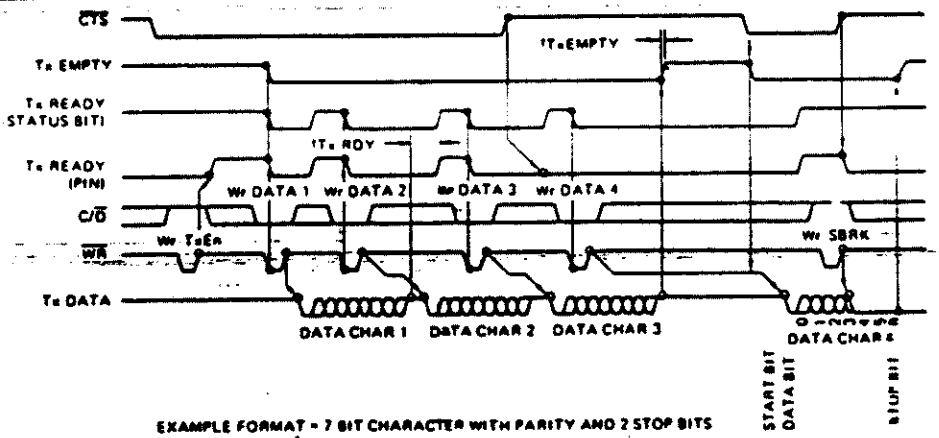
WRITE CONTROL OR OUTPUT PORT CYCLE
(PROCESSOR → USART)



READ CONTROL OR INPUT PORT CYCLE
(PROCESSOR ← USART)

NOTES: ① T_{WC} includes the response timing of a control byte.

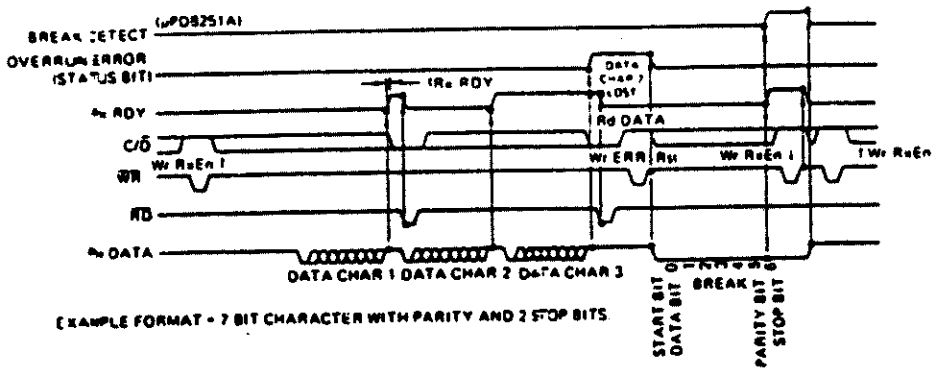
② T_{CR} includes the effect of CTS on the TxENBL circuitry



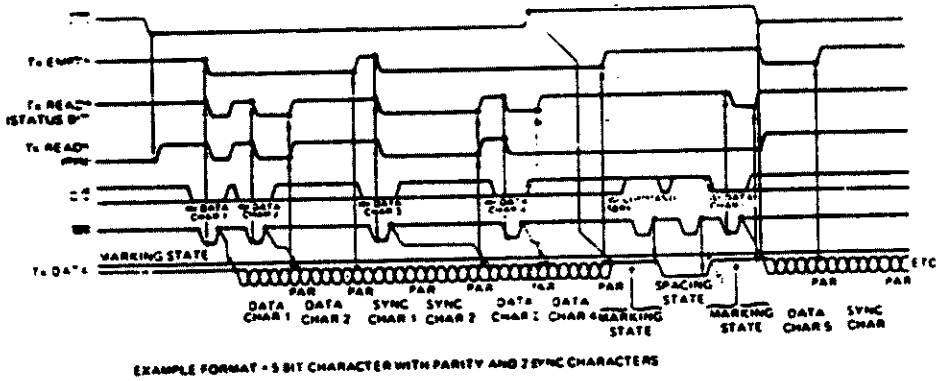
EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY AND 2 STOP BITS

TRANSMITTER CONTROL AND FLAG TIMING
(ASYNC MODE)

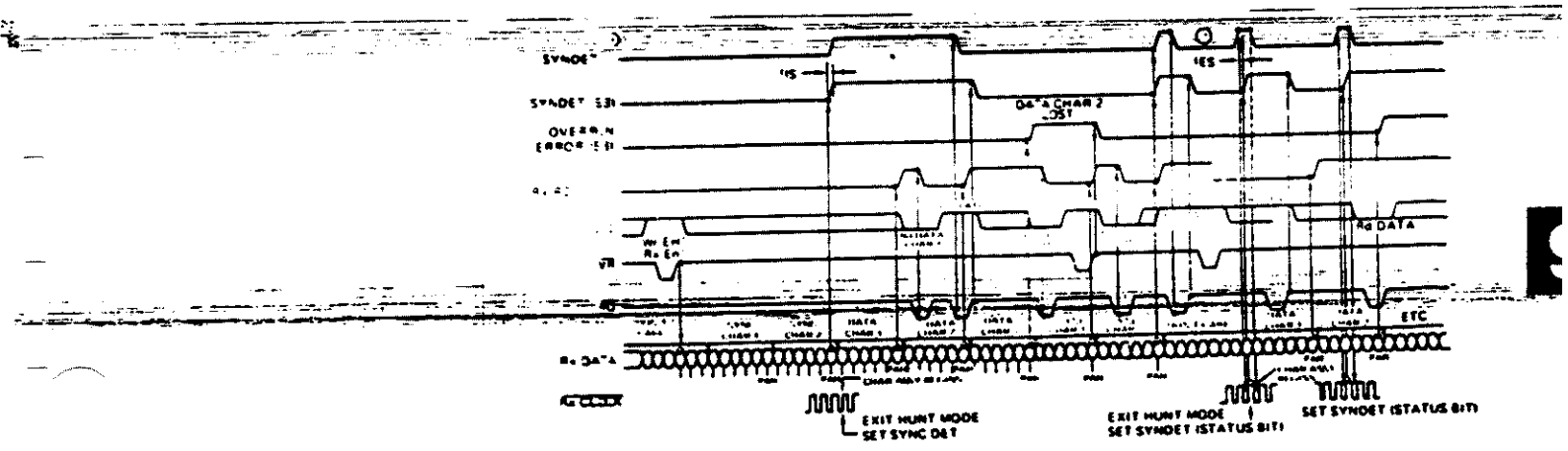
TIMING WAVEFORMS (CONT.)



RECEIVER CONTROL AND FLAG TIMING (ASYNC MODE)



TRANSMITTER CONTROL AND FLAG TIMING (SYNC MODE)



RECEIVER CONTROL AND FLAG TIMING (SYNC MODE)

Notes: ① Internal sync, 2 sync characters, 5 bits, with parity.
② External sync, 5 bits, with parity.

PIN IDENTIFICATION

PIN NO.	PIN SYMBOL	PIN NAME	FUNCTION
1, 2, 27, 28 5 – 8	$D_7 - D_0$	Data Bus Buffer	An 8-bit, 3-state bi-directional buffer used to interface the USART to the processor data bus. Data is transmitted or received by the buffer in response to input/output or Read/ Write instructions from the processor. The Data Bus Buffer also transfers Control words, Command words, and Status.
26	V_{CC}	V_{CC} Supply Voltage	+5 volt supply
4	GND	Ground	Ground
Read/Write Control Logic			This logic block accepts inputs from the processor Control Bus and generates control signals for overall USART operation. The Mode Instruction and Command Instruction registers that store the control formats for device functional definition are located in the Read/ Write Control Logic.
21	RESET	Reset	A "one" on this input forces the USART into the "idle" mode where it will remain until reinitialized with a new set of control words. Minimum RESET pulse width is 6 t_{CY}.
20	CLK	Clock Pulse	The CLK input provides for internal device timing and is usually connected to the Phase 2 (TTL) output of the μPB8224 Clock Generator. External inputs and outputs are not referenced to CLK, but the CLK frequency must be at least 30 times the Receiver or Transmitter clocks in the synchronous mode and 4.5 times for the asynchronous mode.
10	$\overline{WR}$	Write Data	A "zero" on this input instructs the USART to accept the data or control word which the processor is writing out on the data bus.
13	$\overline{RD}$	Read Data	A "zero" on this input instructs the USART to place the data or status information onto the Data Bus for the processor to read.
12	C/$\overline{D}$	Control/Data	The Control/Data input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs the USART to accept or provide either a data character, control word or status information via the Data Bus. 0 = Data; 1 = Control.
11	$\overline{CS}$	Chip Select	A "zero" on this input enables the USART to read from or write to the processor.
Modem Control			The μPD8251 and μPD8251A have a set of control inputs and outputs which may be used to simplify the interface to a Modem.
22	$\overline{DSR}$	Data Set Ready	The Data Set Ready input can be tested by the processor via Status information. The $\overline{DSR}$ input is normally used to test Modem Data Set Ready condition.
24	$\overline{DTR}$	Data Terminal Ready	The Data Terminal Ready output can be controlled via the Command word. The $\overline{DTR}$ output is normally used to drive Modem Data Terminal Ready or Rate Select lines.
23	$\overline{RTS}$	Request to Send	The Request to Send output can be controlled via the Command word. The $\overline{RTS}$ output is normally used to drive the Modem Request to Send line.
17	$\overline{CTS}$	Clear to Send	A "zero" on the Clear to Send input enables the USART to transmit serial data if the TxEN bit in the Command Instruction register is enabled (one).

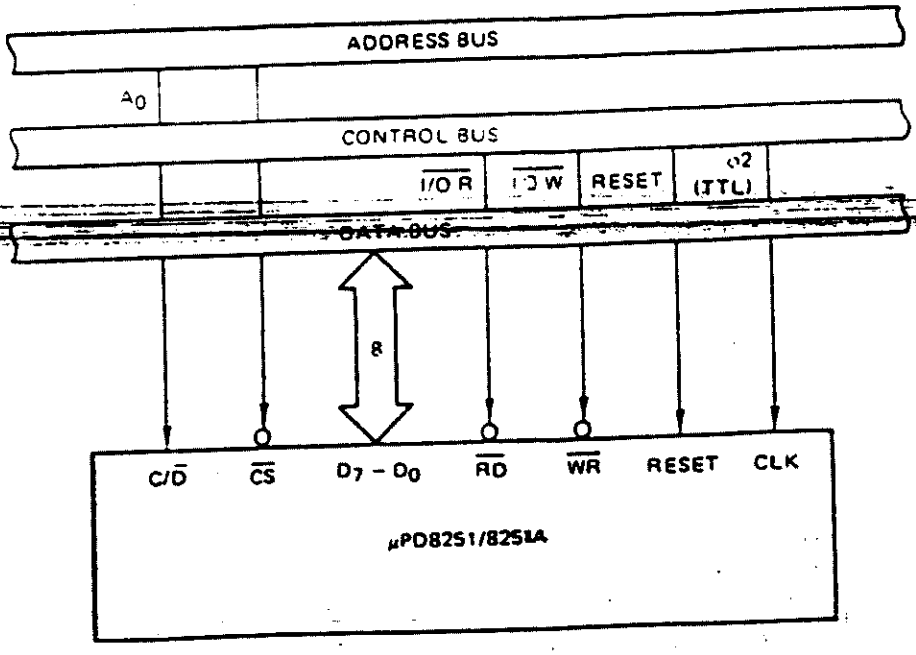
TRANSMIT BUFFER

The Transmit Buffer receives parallel data from the Data Bus Buffer via the internal data bus, converts parallel to serial data, inserts the necessary characters or bits needed for the programmed communication format and outputs composite serial data on the TxD pin.

PIN IDENTIFICATION (CONT.)

PIN NO.	PIN SYMBOL	PIN NAME	FUNCTION
Transmit Control Logic			The Transmit Control Logic accepts and outputs all external and internal signals necessary for serial data transmission.
15	TxRDY	Transmitter Ready	Transmitter Ready signals the processor that the transmitter is ready to accept a data character. TxRDY can be used as an interrupt or may be tested through the Status information for polled operation. Loading a character from the processor automatically resets TxRDY, on the leading edge.
10	TxE	Transmitter Empty	The Transmitter Empty output signals the processor that the USART has no further characters to transmit. TxE is automatically reset upon receiving a data character from the processor. In half-duplex, TxE can be used to signal end of a transmission and request the processor to "turn the line around." The TxEn bit in the command instruction does not effect TxE. In the Synchronous mode, a "one" on this output indicates that a Sync character or characters are about to be automatically transmitted as "fillers" because the next data character has not been loaded.
9	$\overline{TxC}$	Transmitter Clock	The Transmitter Clock controls the serial character transmission rate. In the Asynchronous mode, the $\overline{TxC}$ frequency is a multiple of the actual Baud Rate. Two bits of the Mode Instruction select the multiple to be 1x, 16x, or 64x the Baud Rate. In the Synchronous mode, the $\overline{TxC}$ frequency is automatically selected to equal the actual Baud Rate. Note that for both Synchronous and Asynchronous modes, serial data is shifted out of the USART by the falling edge of $\overline{TxC}$.
19	TxD	Transmitter Data	The Transmit Control Logic outputs the composite serial data stream on this pin.

μPD8251 AND μPD8251A INTERFACE TO 8080 STANDARD SYSTEM BUS



RECEIVE BUFFER

The Receive Buffer accepts serial data input at the $\overline{RxD}$ pin and converts the data from serial to parallel format. Bits or characters required for the specific communication technique in use are checked and then an eight-bit "assembled" character is readied for the processor. For communication techniques which require less than eight bits, the μPD8251 and μPD8251A set the extra bits to "zero."

PIN IDENTIFICATION (CONT.)

PIN NO.	PIN SYMBOL	PIN NAME	FUNCTION
Receiver Control Logic			This block manages all activities related to incoming data.
14	RxRDY	Receiver Ready	The Receiver Ready output indicates that the Receiver Buffer is ready with an "assembled" character for input to the processor. For Polled operation, the processor can check RxRDY using a Status Read or RxRDY can be connected to the processor interrupt structure. Note that reading the character to the processor automatically resets RxRDY.
25	$\overline{RxC}$	Receiver Clock	The Receiver Clock determines the rate at which the incoming character is received. In the Asynchronous mode, the $\overline{RxC}$ frequency may be 1, 16 or 64 times the actual Baud Rate but in the Synchronous mode the $\overline{RxC}$ frequency must equal the Baud Rate. Two bits in the mode instruction select Asynchronous at 1x, 16x or 64x or Synchronous operation at 1x the Baud Rate. Unlike $\overline{TxC}$, data is sampled by the μPD8251 and μPD8251A on the rising edge of $\overline{RxC}$. ①
3	RxD	Receiver Data	A composite serial data stream is received by the Receiver Control Logic on this pin.
16	SYNDET (μPD8251)	Sync Detect	The SYNC Detect pin is only used in the Synchronous mode. The μPD8251 may be programmed through the Mode Instruction to operate in either the internal or external Sync mode and SYNDET then functions as an output or input respectively. In the internal Sync mode, the SYNDET output will go to a "one" when the μPD8251 has located the SYNC character in the Receive mode. If double SYNC character (bi-sync) operation has been programmed, SYNDET will go to "one" in the middle of the last bit of the second SYNC character. SYNDET is automatically reset to "zero" upon a Status Read or RESET. In the external SYNC mode, a "zero"-to-"one" transition on the SYNDET input will cause the μPD8251 to start assembling data characters on the next falling edge of $\overline{RxC}$. The length of the SYNDET input should be at least one $\overline{RxC}$ period, but may be removed once the μPD8251 is in SYNC.
16	SYNDET/BD (μPD8251A)	Sync Detect/ Break Detect	The SYNDET/BD pin is used in both Synchronous and Asynchronous modes. When in SYNC mode the features for the SYNDET pin described above apply. When in Asynchronous mode, the Break Detect output will go high when all zero word of the programmed length is received. This word consists of: start bit, data bit, parity bit and one stop bit. Reset only occurs when Rx data returns to a logic one state or upon chip reset. The state of Break Detect can be read as a status bit.

Note: ① Since the μPD8251 and μPD8251A will frequently be handling both the reception and transmission for a given link, the Receive and Transmit Baud Rates will be same. $\overline{RxC}$ and $\overline{TxC}$ then require the same frequency and may be tied together and connected to a single clock source or Baud Rate Generator.

Examples:

If the Baud Rate equals 110 (Async):
- $\overline{RxC}$ or $\overline{TxC}$ equals 110 Hz (1x)
- $\overline{RxC}$ or $\overline{TxC}$ equals 1.76 KHz (16x)
- $\overline{RxC}$ or $\overline{TxC}$ equals 7.04 KHz (64x)

If the Baud Rate equals 300:
- $\overline{RxC}$ or $\overline{TxC}$ equals 300 Hz (1x) A or S
- $\overline{RxC}$ or $\overline{TxC}$ equals 4800 Hz (16x) A only
- $\overline{RxC}$ or $\overline{TxC}$ equals 19.2 KHz (64x) A only

OPERATIONAL DESCRIPTION

A set of control words must be sent to the μPD8251 and μPD8251A to define the desired mode and communications format. The control words will specify the BAUD rate factor (1x, 16x, 64x), character length (5 to 8), number of STOP bits (1, 1-1/2, 2) Asynchronous or Synchronous mode, SYNDET (IN or OUT), parity, etc.

After receiving the control words, the μPD8251 and μPD8251A are ready to communicate. TxRDY is raised to signal the processor that the USART is ready to receive a character for transmission. When the processor writes a character to the USART, TxRDY is automatically reset.

Concurrently, the μPD8251 and μPD8251A may receive serial data; and after receiving an entire character, the RxRDY output is raised to indicate a completed character is ready for the processor. The processor fetch will automatically reset RxRDY.

Note The μPD8251 and μPD8251A may provide faulty RxRDY for the first read after power-on or for the first read after receive is re-enabled by a command instruction (RxE). A dummy read is recommended to clear faulty RxRDY. But this is not the case for the first read after hardware or software reset after the device operation has once been established.

The μPD8251 and μPD8251A cannot transmit until the TxEN (Transmitter Enable) bit has been set by a Command Instruction and until the $\overline{\text{CTS}}$ (Clear to Send) input is a "zero". TxD is held in the "marking" state after Reset awaiting new control words.

USART PROGRAMMING

The USART must be loaded with a group of two to four control words provided by the processor before data reception and transmission can begin. A RESET (internal or external) must immediately proceed the control words which are used to program the complete operational description of the communications interface. If an external RESET is not available, three successive 00 Hex or two successive 80 Hex command instructions (C/$\overline{\text{D}}$ = 1) followed by a software reset command instruction (40 Hex) can be used to initialize the μPD8251 and μPD8251A.

There are two control word formats:

1. Mode Instruction
2. Command Instruction

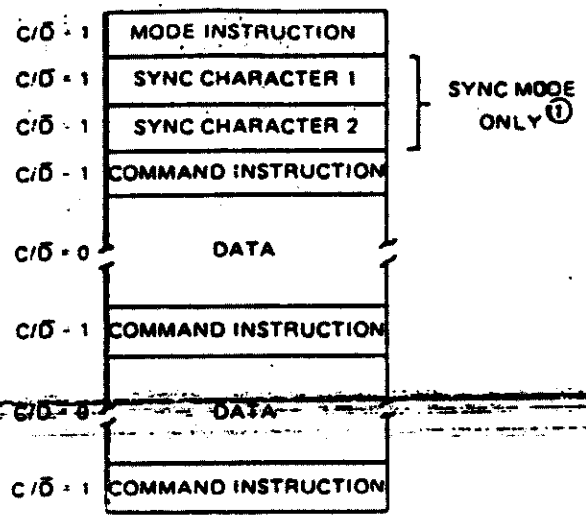
MODE INSTRUCTION

This control word specifies the general characteristics of the interface regarding the Synchronous or Asynchronous mode, BAUD rate factor, character length, parity, and number of stop bits. Once the Mode Instruction has been received, SYNC characters or Command Instructions may be inserted depending on the Mode Instruction content.

COMMAND INSTRUCTION

This control word will be interpreted as a SYNC character definition if immediately preceded by a Mode Instruction which specified a Synchronous format. After the SYNC character(s) are specified or after an Asynchronous Mode Instruction, all subsequent control words will be interpreted as an update to the Command Instruction. Command Instruction updates may occur at any time during the data block. To modify the Mode Instruction, a bit may be set in the Command Instruction which causes an internal Reset which allows a new Mode Instruction to be accepted.

TYPICAL DATA BLOCK



NOTE (1) The second SYNC character is skipped if MODE instruction has programmed the μPD8251 and μPD8251A to single character Internal SYNC Mode. Both SYNC characters are skipped if MODE instruction has programmed the μPD8251 and μPD8251A to ASYNC mode.

MODE INSTRUCTION DEFINITION

The μPD8251 and μPD8251A can operate in either Asynchronous or Synchronous communication modes. Understanding how the Mode Instruction controls the functional operation of the USART is easiest when the device is considered to be two separate components (one asynchronous and the other synchronous) which share the same support circuits and package. Although the format definition can be changed at will or "on the fly," the two modes will be explained separately for clarity.

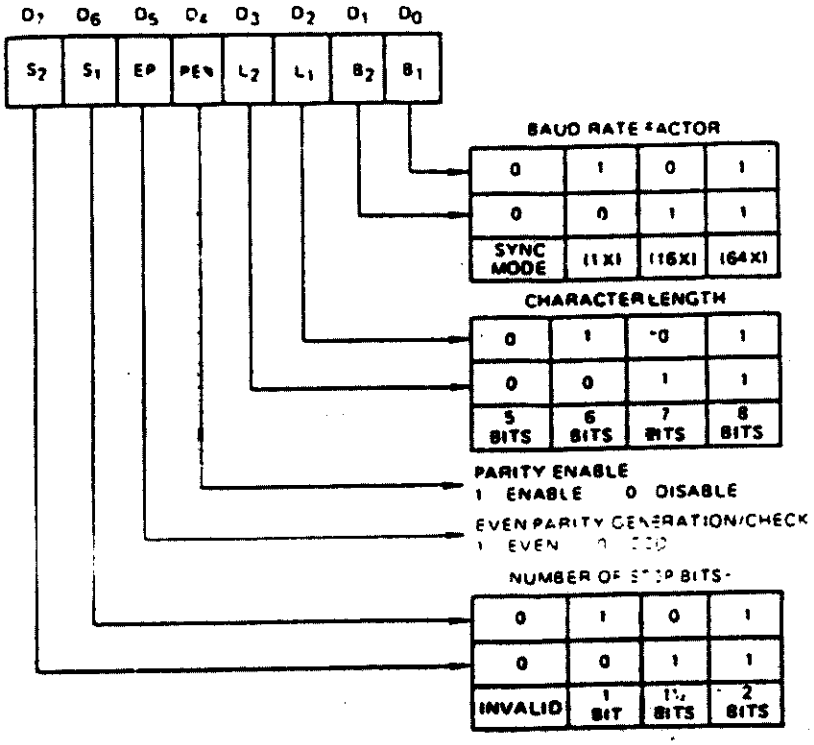
ASYNCHRONOUS TRANSMISSION

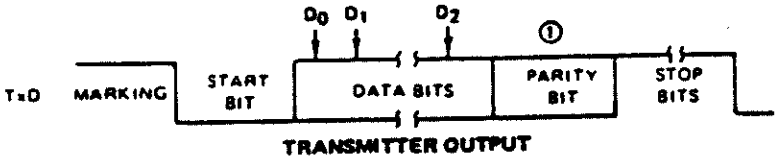
When a data character is written into the μPD8251 and μPD8251A, the USART automatically adds a START bit (low level or "space") and the number of STOP bits (high level or "mark") specified by the Mode Instruction. If Parity has been enabled, an odd or even Parity bit is inserted just before the STOP bit(s), as specified by the Mode Instruction. Then, depending on $\overline{CTS}$ and TxEN, the character may be transmitted as a serial data stream at the TxD output. Data is shifted out by the falling edge of $\overline{TxC}$ at $\overline{TxC}$, $\overline{TxC}/16$ or $\overline{TxC}/64$, as defined by the Mode Instruction.

If no data characters have been loaded into the μPD8251 and μPD8251A, or if all available characters have been transmitted, the TxD output remains "high" (marking) in preparation for sending the START bit of the next character provided by the processor. TxD may be forced to send a BREAK (continuously low) by setting the correct bit in the Command Instruction.

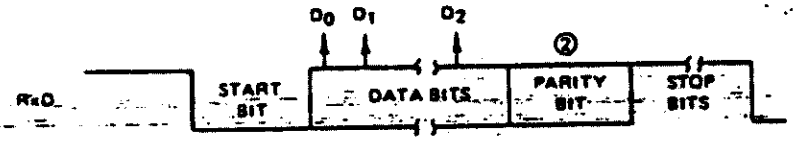
ASYNCHRONOUS RECEIVE

The RxD input line is normally held "high" (marking) by the transmitting device. A falling edge at RxD signals the possible beginning of a START bit and a new character. The START bit is checked by testing for a "low" at its nominal center, as specified by the BAUD RATE. If a "low" is detected again, it is considered valid, and the bit assembling counter starts counting. The bit counter locates the approximate center of the data, parity (if specified), and STOP bits. The parity error flag (PE) is set, if a parity error occurs. Input bits are sampled at the RxD pin with the rising edge of $\overline{RxC}$. If a high is not detected for the STOP bit, which normally signals the end of an input character, a framing error (FE) will be set. After a valid STOP bit, the input character is loaded into the parallel Data Bus Buffer of the μPD8251 and μPD8251A and the RxRDY signal is raised to indicate to the processor that a character is ready to be fetched. If the processor has failed to fetch the previous character, the new character replaces the old and the overrun flag (OE) is set. All the error flags can be reset by setting a bit in the Command Instruction. Error flag conditions will not stop subsequent USART operation.





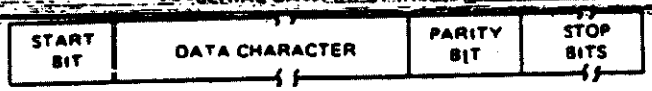
TRANSMITTER OUTPUT

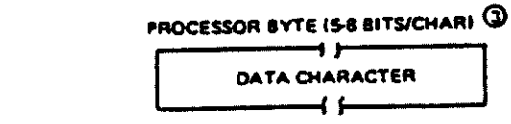


RECEIVER INPUT

ASSEMBLED SERIAL DATA OUTPUT (T×D)

TRANSMISSION FORMAT





RECEIVE FORMAT

Notes: ① Generated by μPD8251/8251A
② Does not appear on the Data Bus.
③ If character length is defined as 5, 6, or 7 bits, the unused bits are set to "zero."

SYNCHRONOUS TRANSMISSION

As in Asynchronous transmission, the TxD output remains "high" (marking) until the μPD8251 and μPD8251A receive the first character (usually a SYNC character) from the processor. After a Command Instruction has set TxEN and after Clear to Send ($\overline{CTS}$) goes low, the first character is serially transmitted. Data is shifted out on the falling edge of $\overline{TxC}$ and the same rate as $\overline{TxC}$.

Once transmission has started, Synchronous Mode format requires that the serial data stream at TxD continue at the $\overline{TxC}$ rate or SYNC will be lost. If a data character is not provided by the processor before the μPD8251 and μPD8251A Transmit Buffer becomes empty, the SYNC character(s) loaded directly following the Mode Instruction will be automatically inserted in the TxD data stream. The SYNC character(s) are inserted to fill the line and maintain synchronization until new data characters are available for transmission. If the μPD8251 and μPD8251A become empty, and must send the SYNC character(s), the TxEMPTY output is raised to signal the processor that the Transmitter Buffer is empty and SYNC characters are being transmitted. TxEMPTY is automatically reset by the next character from the processor.

SYNCHRONOUS RECEIVE

In Synchronous Receive, character synchronization can be either external or internal. If the internal SYNC mode has been selected, and the Enter HUNT (EH) bit has been set by a Command Instruction, the receiver goes into the HUNT mode.

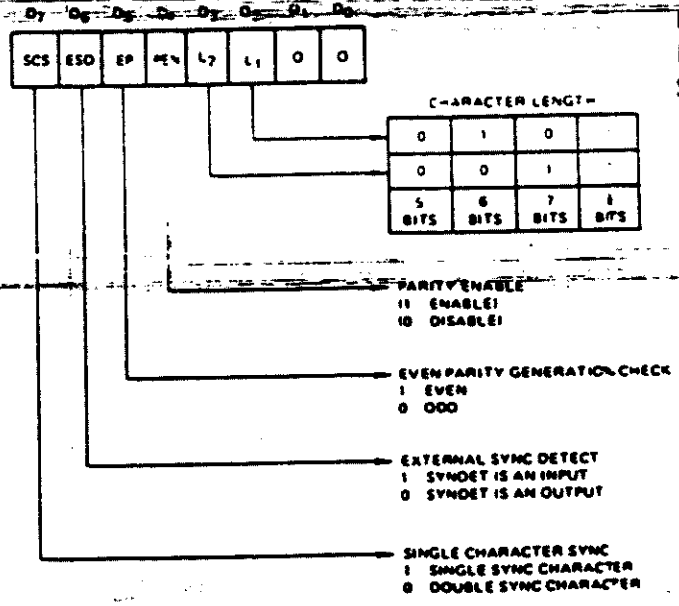
Incoming data on the RxD input is sampled on the rising edge of $\overline{RxC}$, and the Receive Buffer is compared with the first SYNC character after each bit has been loaded until a match is found. If two SYNC characters have been programmed, the next received character is also compared. When the SYNC character(s) programmed have been detected, the μPD8251 and μPD8251A leave the HUNT mode and are in character synchronization. At this time, the SYNDET (output) is set high. SYNDET is automatically reset by a STATUS READ.

If external SYNC has been specified in the Mode Instruction, a "one" applied to the SYNDET (input) for at least one $\overline{RxC}$ cycle will synchronize the USART.

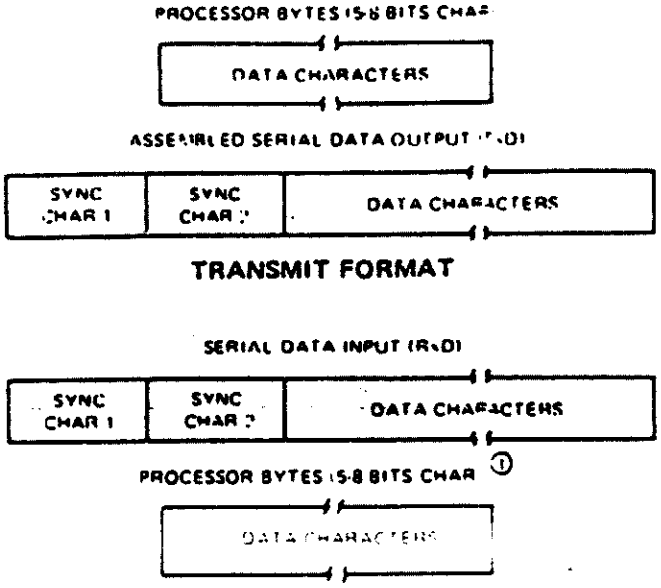
Parity and Overrun Errors are treated the same in the Synchronous as in the Asynchronous Mode. If not in HUNT, parity will continue to be checked even if the receiver is not enabled. Framing errors do not apply in the Synchronous format.

The processor may command the receiver to enter the HUNT mode with a Command Instruction which sets Enter HUNT (EH) if synchronization is lost.

MODE INSTRUCTION FORMAT SYNCHRONOUS MODE



TRANSMIT/RECEIVE FORMAT SYNCHRONOUS MODE



Note ① If character length is defined as 5, 6 or 7 bits the unused bits are set to "zero."

COMMAND INSTRUCTION FORMAT

After the functional definition of the μPD8251 and μPD8251A has been specified by the Mode Instruction and the SYNC character(s) have been entered (if in SYNC mode), the USART is ready to receive Command Instructions and begin communication. A Command Instruction is used to control the specific operation of the format selected by the Mode Instruction. Enable Transmit, Enable Receive, Error Reset and Modem Controls are controlled by the Command Instruction.

After the Mode Instruction and the SYNC character(s) (as needed) are loaded, all subsequent "control writes" ($C/\overline{D}$ = 1) will load or overwrite the Command Instruction register. A Reset operation (internal via CMD IR or external via the RESET input) will cause the μPD8251 and μPD8251A to interpret the next "control write", which must immediately follow the reset, as a Mode Instruction.

STATUS READ FORMAT

It is frequently necessary for the processor to examine the status of an active interface device to determine if errors have occurred or if there are other conditions which require a response from the processor. The μPD8251 and μPD8251A have features which allow the processor to read the device status at any time. A data fetch is issued by the processor while holding the $C/\overline{D}$ input "high" to obtain device Status Information. Many of the bits in the status register are copies of external pins. This dual status arrangement allows the μPD8251 and μPD8251A to be used in both Polled and Interrupt driven environments. Status update can have a maximum delay of 16 clock periods in the μPD8251 and 28 clock periods in the μPD8251A.

PARITY ERROR

When a parity error is detected, the PE flag is set. It is cleared by setting the ER bit in a subsequent Command Instruction. PE being set does not inhibit USART operation.

OVERRUN ERROR

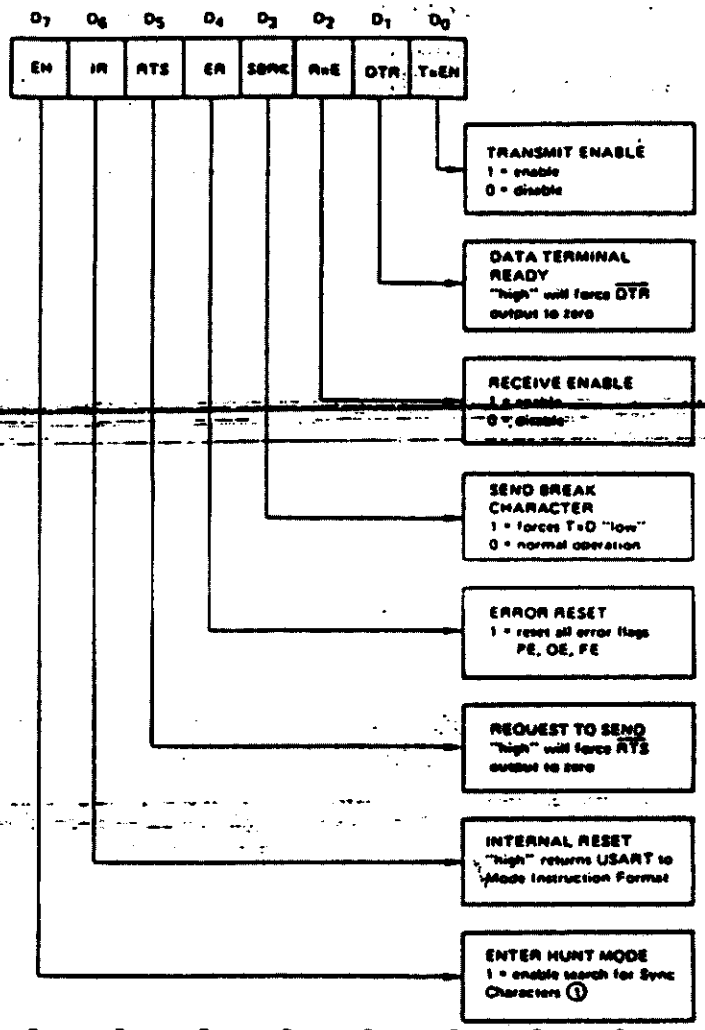
If the processor fails to read a data character before the one following is available, the OE flag is set. It is cleared by setting the ER bit in a subsequent Command Instruction. Although OE being set does not inhibit USART operation, the previously received character is overwritten and lost.

FRAMING ERROR ①

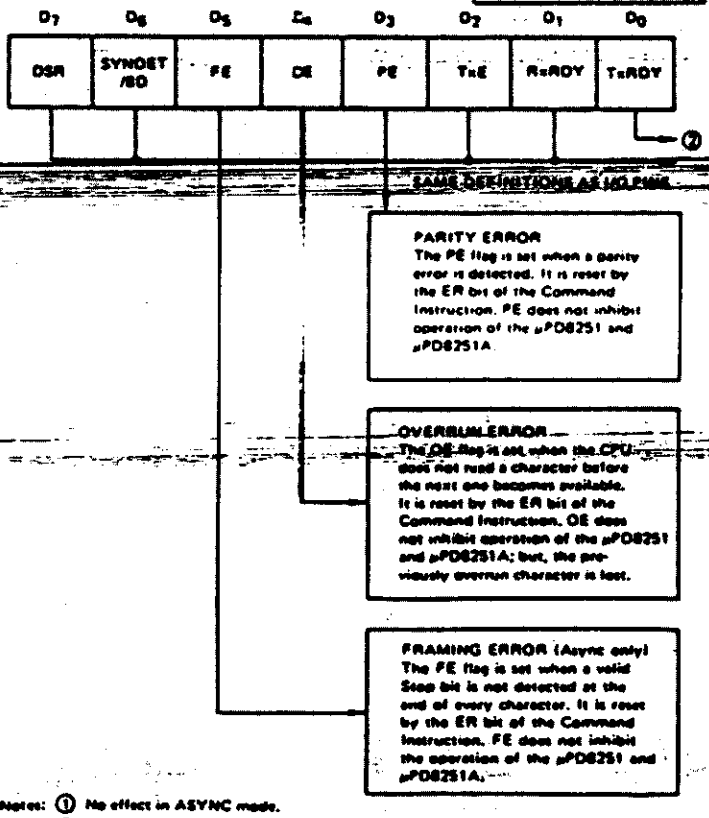
If a valid STOP bit is not detected at the end of a character, the FE flag is set. It is cleared by setting the ER bit in a subsequent Command Instruction. FE being set does not inhibit USART operation.

Note: ① ASYNC mode only.

COMMAND INSTRUCTION FORMAT



STATUS READ FORMAT

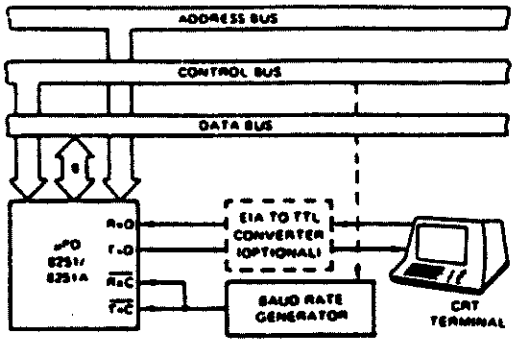


Notes: ① No effect in ASYNC mode.

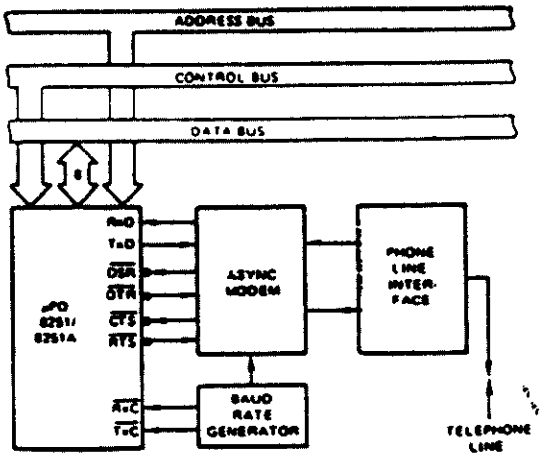
② TxRDY status bit is not totally equivalent to the TxRDY output pin, the relationship is as follows:

TxRDY status bit = DB Buffer Empty

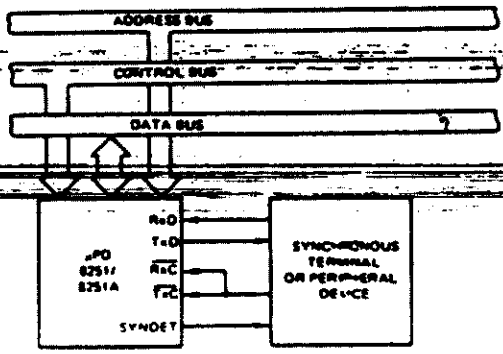
APPLICATION OF THE μPD8251 AND μPD8251A



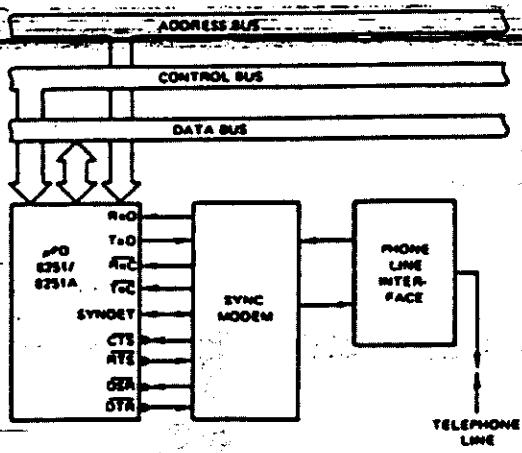
ASYNCHRONOUS SERIAL INTERFACE TO CRT TERMINAL, DC to 9600 BAUD



ASYNCHRONOUS INTERFACE TO TELEPHONE LINES



SYNCHRONOUS INTERFACE TO TERMINAL OR PERIPHERAL DEVICE



SYNCHRONOUS INTERFACE TO TELEPHONE LINES

intel

APPLICATION NOTE

AP-15

8255 Programmable Peripheral Interface Applications

Alan Ebright
Microcomputer Applications

INTRODUCTION

Microprocessor-based system designs are a cost-effective solution to a wide variety of problems. When a system designer is presented with the task of selecting a microprocessor for a design, the capabilities of the microprocessor should not be the only consideration. The microprocessor should be an element of a compatible family of devices. The MCS-80 component family is a group of compatible devices which have been designed to directly address and solve the problems of microprocessor-based system design. One member of the MCS-80 component family is Intel's 8255 programmable peripheral interface chip. This device replaces a significant percentage of the logic required to support a variety of byte oriented Input/Output interfaces. Through the use of the 8255, the I/O interface design task is significantly simplified, the design flexibility is increased, and the number of components required is reduced.

This application note presents detailed design examples from both the hardware and software points of view. Since the 8255 is an extremely flexible device, it is impossible to list all of the applications and configurations of the device. A number of designs are presented which may be modified to fulfill specific user interface requirements.

Detailed design examples are discussed within the context of the 8080 system shown in Figure 1. The basic 8080 system is composed of the CPU module, memory module, and the I/O module. CPU module and memory module design are discussed within other Intel publications. This application note deals exclusively with I/O module design.

It is assumed that the reader is familiar with the "8080 Microcomputer Systems User's Manual", particularly the 8255 device description.

OVERVIEW OF THE 8255

The 8255 block diagram shown in Figure 2 has been divided into three sections: 8080 CPU Module Interface, Peripheral Interface, and the Internal Logic.

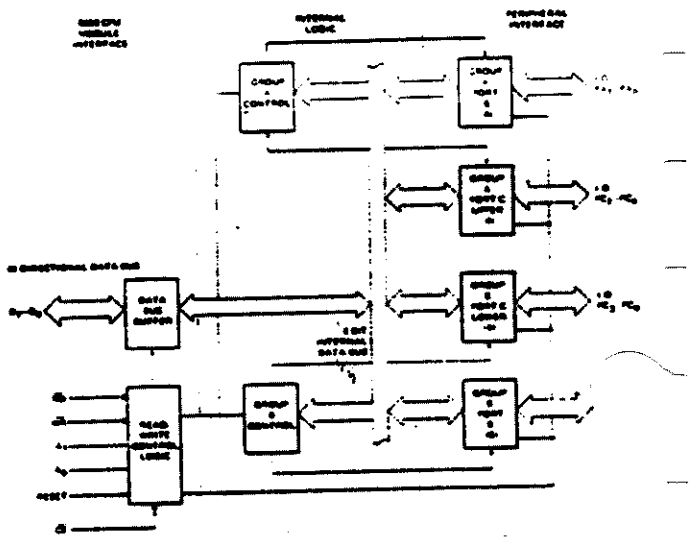


Figure 2. 8255 Block Diagram

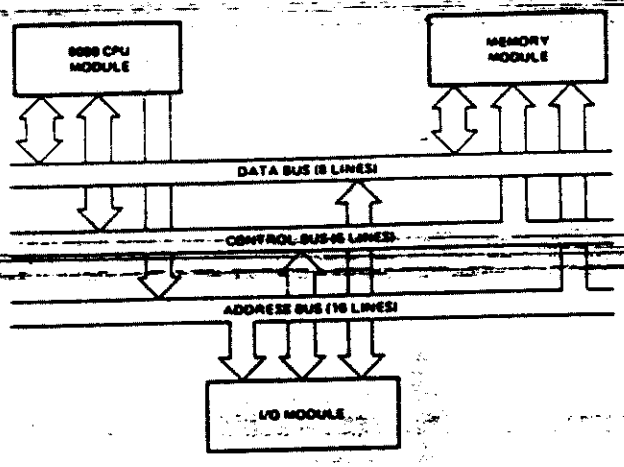


Figure 1. Typical 8080 System

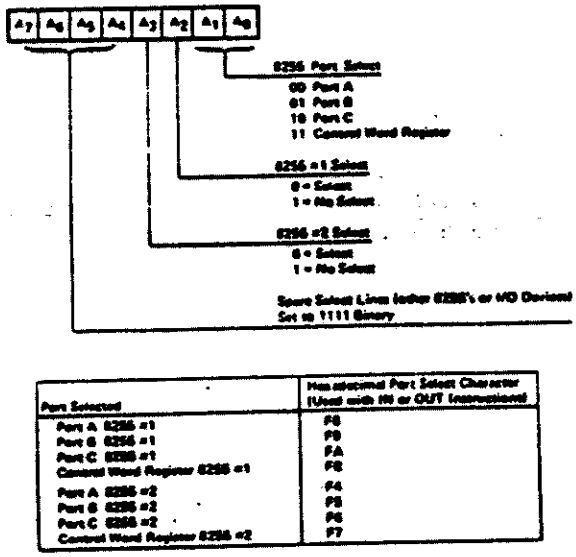
8080 CPU MODULE INTERFACE

The 8255 is a compatible member of the MCS-80 component family and, therefore, may be directly interfaced to the 8080. Figure 3 displays one method of interconnecting the 8255 and an 8080 CPU module. The 8080 CPU module consists of the 8080A CPU, the 8224 Clock Generator, and the 8228 System Controller. The system shown in Figure 3 utilizes a linear select scheme which dedicates an address line as an exclusive enable (chip select) for each specific I/O device. The chip select signal is used to enable communication between the selected 8255 and the 8080 CPU. I/O Ports A, B, C or the Control Word Register are selected by the two port select signals (A_1, A_0). These signals (A_1 and A_0) are driven by the least significant bits of the address bus. The I/O port select characters required by this configuration are shown in Figure 4.

When a system utilizing the linear select scheme is implemented, a maximum of six I/O devices may be selected. If more than six I/O devices must be addressed, the six device select bits must be encoded to generate a maximum of 64 device select lines. Note that when large systems are implemented, bus loading considerations may require that bus drivers be included in the CPU module. The MCS-80 component family contains parts which are designed to perform this function (8216, 8226).

The 8255 I/O read ($\overline{RD}$) and I/O write ($\overline{WR}$) signals may be directly driven by the 8228. This results in an isolated I/O architecture where 8080 Input/Output instructions are used to reference an independent I/O address space. An alternate approach is memory mapped I/O. This architecture treats an area of memory as the I/O address space. The memory mapped I/O architecture utilizes 8080 memory reference instructions to access the I/O address space. Interfacing with the 8080 is outlined in Chapter 3 of the "8080 Microcomputer User's Manual".

The most important feature of the 8255 to 8080 CPU Module Interface is that for small system designs the 8255 may be interfaced directly to the standard MCS-80 component family with no external logic. Minimum external logic is required in large system designs.



Port Selected	Hexadecimal Port Select Character (Used with IN or OUT Instructions)
Port A 8255 #1	F8
Port B 8255 #1	F9
Port C 8255 #1	FA
Control Word Register 8255 #1	FB
Port A 8255 #2	F4
Port B 8255 #2	F5
Port C 8255 #2	F6
Control Word Register 8255 #2	F7

Figure 4. I/O Port Select Characters

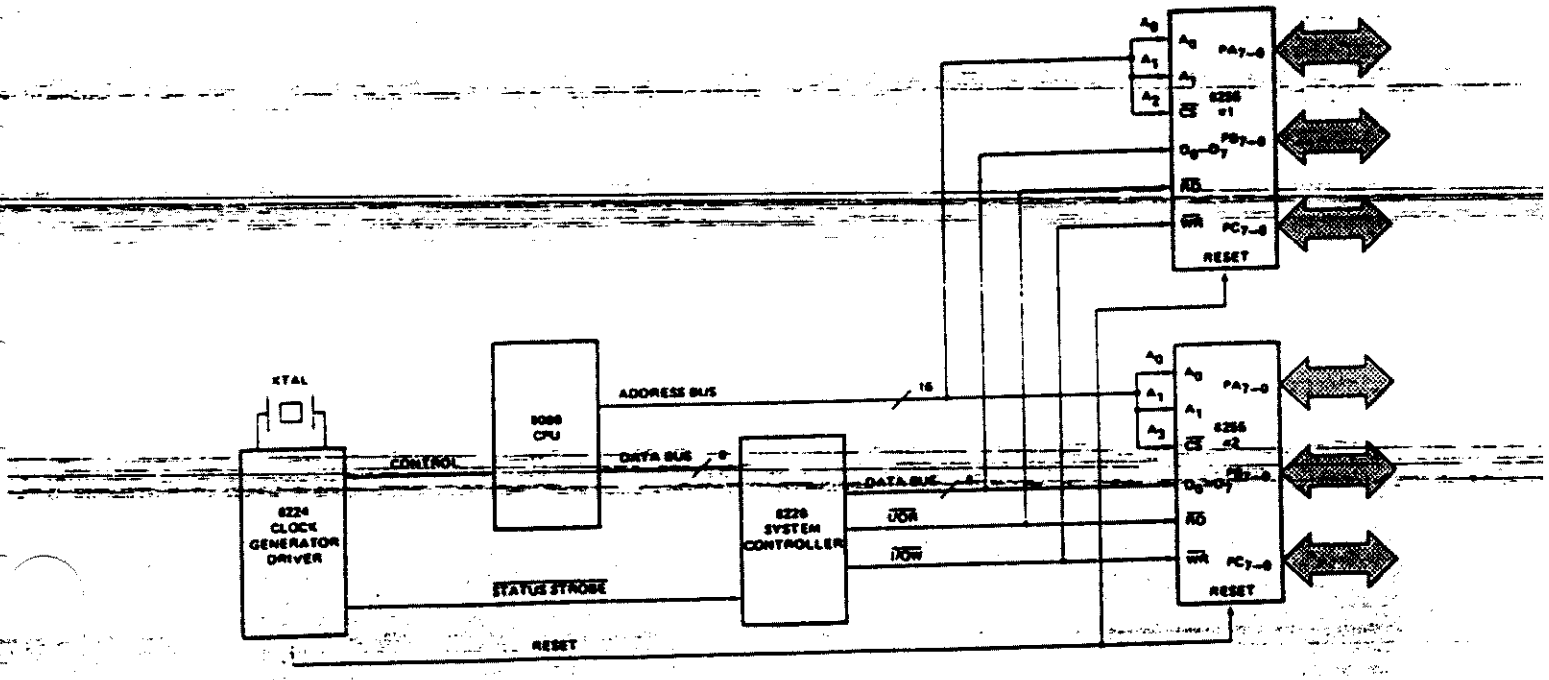


Figure 3. Linear Select 8255 Interconnect

PERIPHERAL INTERFACE SECTION

The peripheral interface section contains 24 peripheral interface lines, buffers, and control logic. The characteristics and functions of the interface lines are determined by the operating mode selected under program control. The flexibility of the 8255 is due to the fact that the device is programmable. Three modes of operation may be selected under program control: Mode 0 – Basic Input/Output, Mode 1 – Strobed Input/Output with interrupt support, and Mode 2 – Bidirectional bus with interrupt support. Through selecting the correct operating mode, the interface lines may be configured to fulfill specific interface requirements. The characteristics of the interface lines within each mode must be understood so that the designer may utilize the 8255 to achieve the most efficient design. Table I lists the basic features of the peripheral interface lines within each mode group. Figure 5 shows the grouping of the peripheral interface lines within each mode.

One feature of Port C is important to note. Each Port C bit may be individually set and reset. Through the use of this feature, device strobes may be easily generated by software without utilizing external logic. The Mode 1 and Mode 2 configurations use a number of the Port C lines for interrupt control lines. Thus, the 8255 contains a large portion of the logic required to implement an interrupt driven I/O interface. This feature simplifies interrupt driven hardware design and saves a significant amount of the external logic that is normally required when less powerful I/O chips are used. In fact, the design examples contained in this application note describe how interrupt driven interfaces may be designed such that the only interrupt control logic required is that contained in the 8255.

Table I. Features of Peripheral Interface Lines

Mode 0 – Basic Input/Output
Two 8-bit ports Two 4-bit ports with bit set/reset capability Outputs are latched Inputs are not latched
Mode 1 – Strobed Input/Output
One or two strobed ports Each Mode 1 port contains: 8-bit data port 3 control lines Interrupt support logic Any port may be input or output If one Mode 1 port is used, the remaining 13 lines may be configured in Mode 0. If two Mode 1 ports are used, the remaining 2 bits may be input or output with bit set/reset capability.
Mode 2 – Strobed Bidirectional Bus
One bidirectional bus which contains: 8-bit bidirectional bus supported by Port A 5 control lines Interrupt support logic Inputs and outputs are latched The remaining 11 lines may be configured in either Mode 0 or Mode 1.

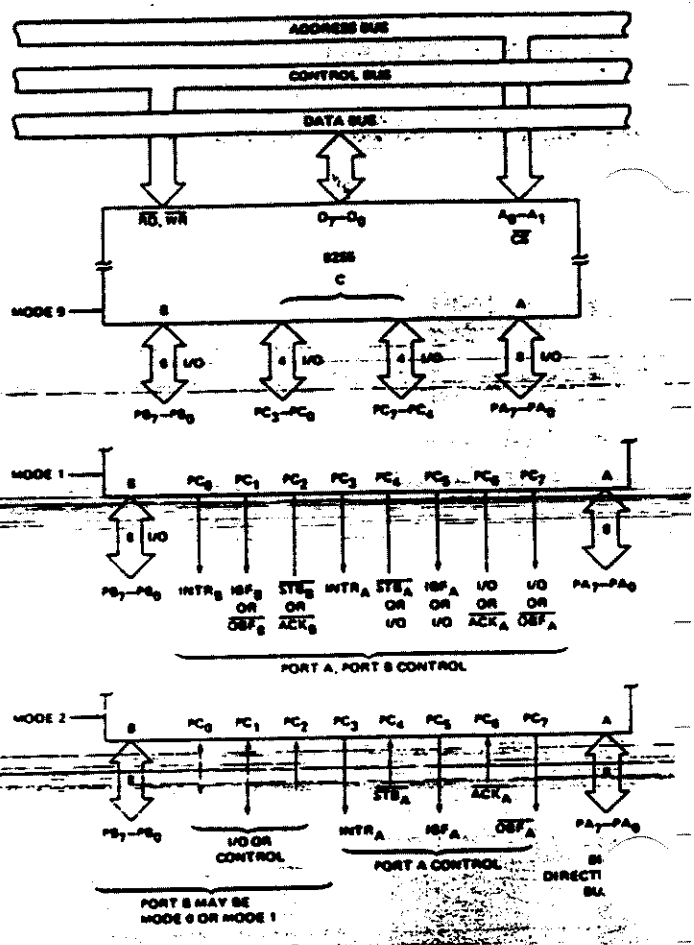


Figure 5. Grouping of Peripheral Interface Lines

INTERNAL LOGIC SECTION

The internal logic section manages the transfer of data and control information on the internal data bus (refer to Figure 2). If the port select lines (A_1 and A_0) specify Ports A, B, or C, the operation is an I/O port data transfer. The internal logic will select the specified I/O port and perform the data transfer between the I/O port and the CPU interface. As was previously mentioned, both the functional configuration of each port and bit set/reset on Port C are controlled by the system's software. When the control word register is selected, the internal logic performs the operation described by the control word. The control word contains an opcode field which defines which of the two functions are to be performed (mode definition or bit set/reset).

Mode Definition

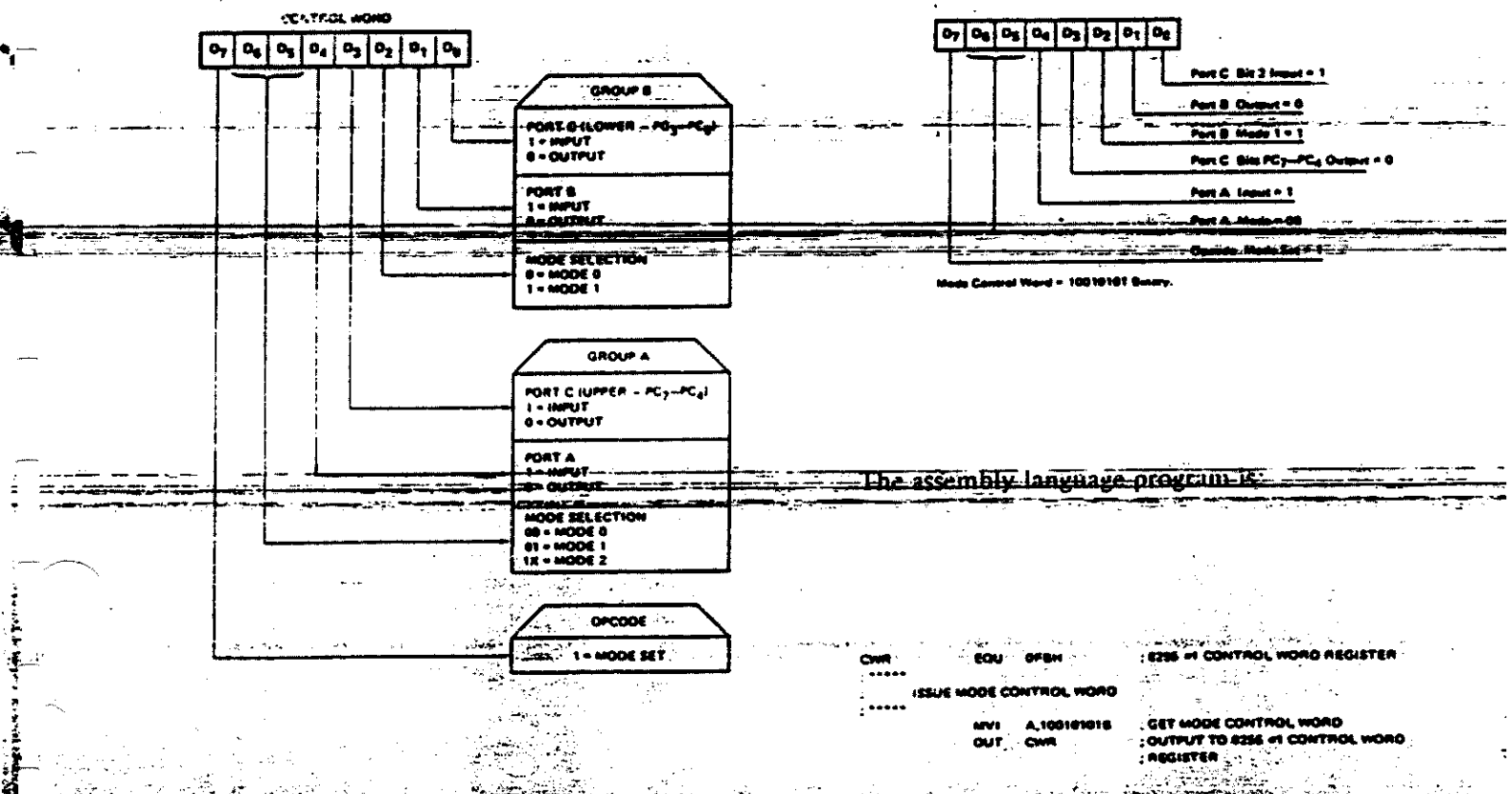
When the opcode field (Bit 7) of the control word is equal to a one, the control word is interpreted by the 8255 as a mode definition control word. The mode definition control word (shown in Figure 6) is used to specify the configuration of the 24 8255 peripheral interface lines. The system's software may specify the modes of Port A and Port B independently. Port C may be treated independently or divided into two portions as required by the Port A and Port B mode definitions.

Example #1: This example demonstrates how a mode control word is constructed and issued to an 8255. The mode control word is passed to the device through the use of an output instruction that references an 8080 I/O port address. The value of the I/O port address is determined by the 8080 CPU interface implemented. This example references the I/O port addresses realized by the simple 8080 to 8255 interface shown in Figure 3.

If an 8255 is to be configured through the use of the mode control word interface as:

Port A	Mode 0 Input
Port B	Mode 1 Output
Port C	Bits PC_7–PC_4 Output
Port C	Bit 3 Input

The following mode control word is used:



The assembly language program is:

```
CWR     EQU   0F8H      ;8255 #1 CONTROL WORD REGISTER
;*****
;ISSUE MODE CONTROL WORD
;*****
        MVI   A,10010101B   ;GET MODE CONTROL WORD
        OUT   CWR           ;OUTPUT TO 8255 #1 CONTROL WORD
                            ;REGISTER
```

Figure 6. Mode Definition Control Word

Bit Set/Reset

When the opcode field (Bit 7) of the control word is equal to a zero, the control word is interpreted by the 8255 as a Port C bit set/reset command word (see Figure 7). Through the use of the bit set/reset command, any of the 8 bits on Port C may be independently set or reset. Note that control word bits 6–4 are not used. Bits 6–4 should be set to zero.

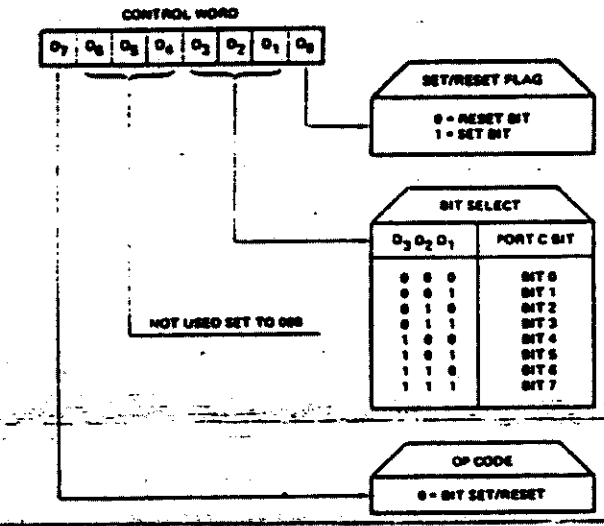
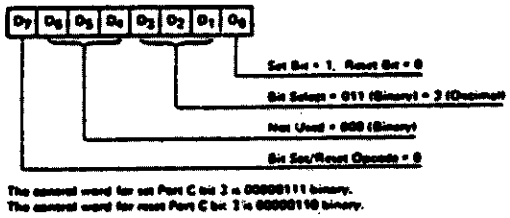


Figure 7. Bit Set/Reset Control Word

Example #2: This example demonstrates how a Port C bit set/reset control word is constructed and issued to an 8255. The bit set/reset control word is passed to the device through the use of an output instruction that references an 8080 I/O port address. The value of the I/O port address is determined by the 8080 CPU interface implemented. This example references the I/O port addresses realized by the simple 8080 to 8255 interface shown in Figure 3.

Control word (see Figure 7).



The assembly language program is:

```
CWR     EQU     0F8H            ;8255 #1 CONTROL WORD REGISTER
;*****
        SET BIT 3
;*****
        MVI     A,00000111B     ;GET SET BIT 3 CONTROL WORD
        OUT     CWR             ;OUTPUT TO 8255 #1 CONTROL WORD REGISTER
;*****
        RESET BIT 3
;*****
        MVI     A,00000110B     ;GET RESET BIT 3 CONTROL WORD
        OUT     CWR             ;OUTPUT TO 8255 #1 CONTROL WORD REGISTER
```

NOTE: An MVI instruction is used to load the reset bit 3 control word into the A register. Since it is known that the set bit control word is already in the A register, a "DCR A" Instruction could be used to generate the correct control word and save one byte of code.

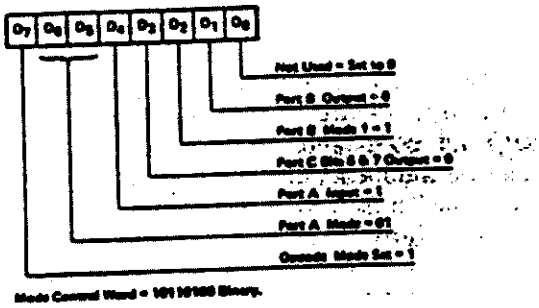
00000111 – 1 = 00000110 (RESET BIT 3 CONTROL WORD)

Example #3: This example demonstrates one simple method of performing a bit set/reset operation on Ports A and B. The state of any output port may be determined by reading the port. The assembly language program which may be used to set/reset Port A or B bits is:

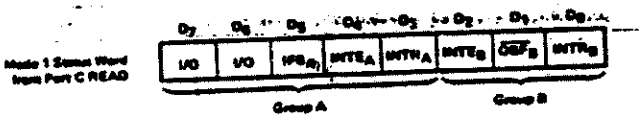
```
PORTA   EQU     0F8H            ;8255 #1 PORT A
;*****
        SET BIT 0
;*****
        IN      PORTA           ;GET STATE OF PORT
        ORI     01H             ;SET BIT 0
        OUT     PORTA           ;OUTPUT TO PORT
;*****
        RESET BIT 0
;*****
        IN      PORTA           ;GET STATE OF PORT
        ANI     0FEH            ;RESET BIT 0
        OUT     PORTA           ;OUTPUT TO PORT
```

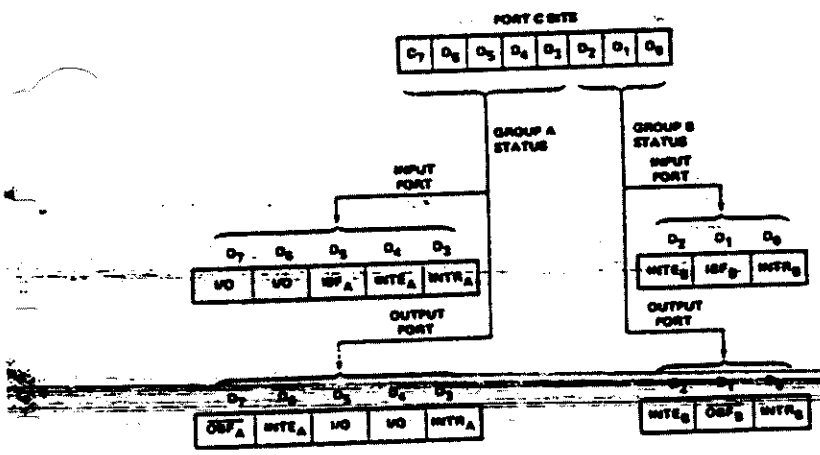
INTERRUPT CONTROL LOGIC STATUS WORDS

As previously mentioned, the 8255 Mode 1 and Mode 2 configurations support interrupt control logic. If a read of Port C is issued when the 8255 is configured in Mode 1, the software will receive the Mode 1 status word shown in Figure 8. The bits in the status word correspond to the state of the associated Port C lines (buffer full, interrupt request, etc.). The INTE bit shown in the status word corresponds to the interrupt enable flip-flop contained in the 8255. This signal is not available externally. The structure of the Mode 1 status word varies as a function of the mode of the 8255. Example #4 shows the status word which results from reading Port C from an 8255 which is configured with Port A Mode 1 input and Port B Mode 1 output.



After the 8255 mode control word has been issued, a READ of Port C will obtain the following Mode 1 status word:





NOTE: The Port C I/O bits D_7 and D_6 should be modified through the use of the Port C bit set/reset command word. If a write to Port C is issued, the $INTE_A$ and $INTE_B$ bits may be inadvertently modified by the user. The IBF_A, $INTR_A$, $\overline{OBF}_B$, and $INTR_B$ bits will not be modified by either a write to Port C or a bit set/reset command. These four bits always reflect the state of the interrupt control logic.

Figure 8. Mode 1 Status Word

Example #4 – MODE 1 STATUS WORD

If an 8255 is to be configured through the use of the mode control word interface as:

Port A – Mode 1 Input
Port B – Mode 1 Output
Port C – Bits 6 & 7 Output

The following mode control word is used:

Note that the Mode 2 status word (shown in Figure 9) differs from the Mode 1 status word. The format of the status word data bits D_2–D_0 are defined by the specification of the Port B configuration. Example #5 shows the structure of the Mode 2 status word when the 8255 is configured with Port A Mode 2 (bidirectional bus) and Port B Mode 1 input.

The Mode 1 and Mode 2 status words reflect the state of the interrupt logic supported by the 8255.

Example #6 demonstrates how the interrupt enable bits are controlled through the use of the Port C bit set/reset feature. The application examples provide a more detailed explanation of the use of the Port C status word in the Mode 1 and Mode 2 configurations.

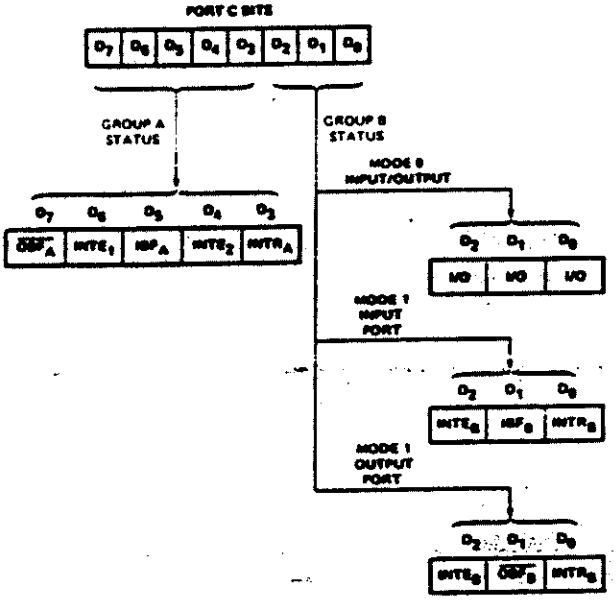


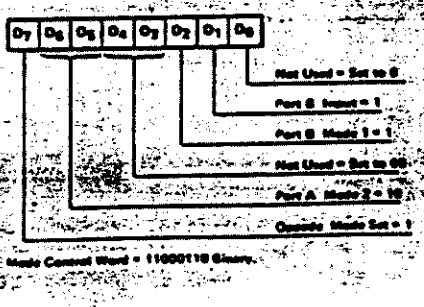
Figure 9. Mode 2 Status Word

Example #5 – MODE 2 STATUS WORD

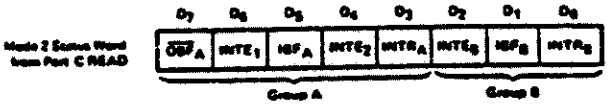
If the 8255 is to be configured as follows:

Port A — Mode 2 Bidirectional Bus
Port B — Mode 1-Input

The following mode control word is used:



After the 8255 mode control word has been issued, a read of Port C will obtain the following Mode 2 status word:



Example #6 – MODE 2 INTERRUPT ENABLE/ DISABLE

The Mode 2 status word shown in Figure 9 contains two interrupt enable bits:

$INTE_1$ – Bit 6 – Enable output interrupts
$INTE_2$ – Bit 4 – Enable input interrupts

Bit set/reset control words may be constructed which may be used to control the INTE bits.

Set Bit 6 (Enable Output Interrupts) = 00001101 Binary

Reset Bit 6 (Disable Output Interrupts) = 00001100 Binary

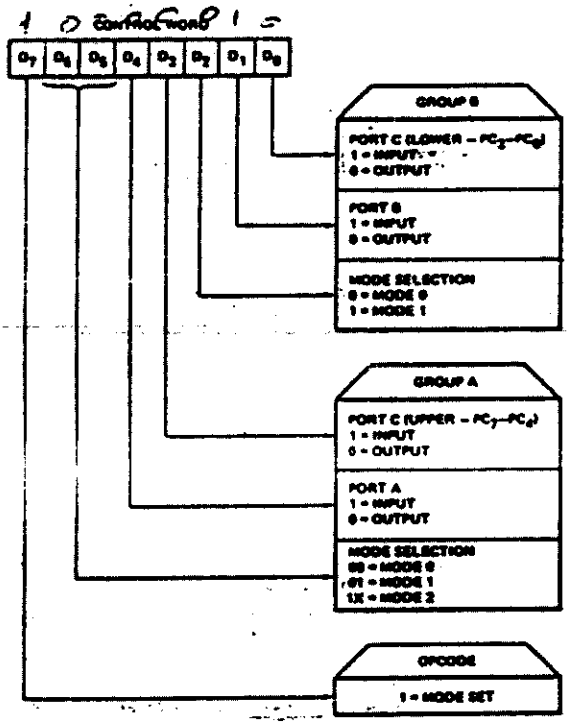
Set Bit 4 (Enable Input Interrupts) = 00001001 Binary

Reset Bit 4 (Disable Input Interrupts) = 00001000 Binary

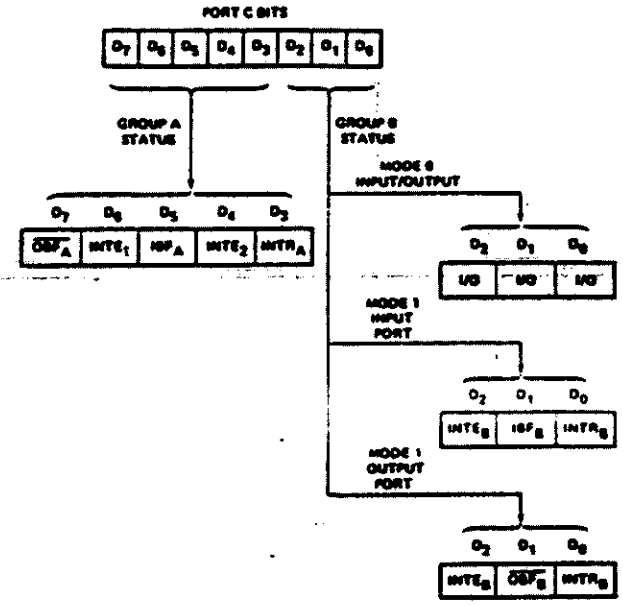
The control words shown were constructed from the standard bit set/reset format shown in Figure 7.

The value of CWR used in the following progr example corresponds to the 8080 configurat shown in Figure 3.

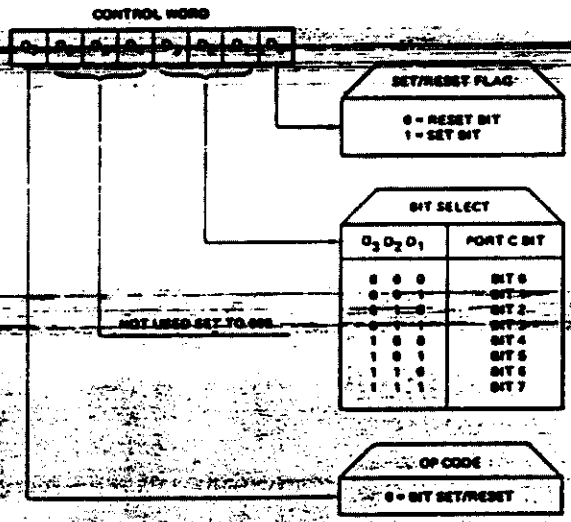
```
CWR     EQU     0F8H            ;8255 #1 CONTROL WORD REGISTER
;*******ENABLE INTERRUPTS FOR MODE 2 OUTPUT (SET PORT C BIT 6)
;*******
        MVI     A,00001101B     ; GET SET BIT 6 CONTROL WORD
        OUT     CWR             ; OUTPUT TO 8255 #1 CONTROL WORD REGISTER
;*******
;*******DISABLE INTERRUPTS FOR MODE 2 OUTPUT (RESET PORT C BIT 6)
;*******
        MVI     A,00001100B     ; GET RESET BIT 6 CONTROL WORD
        OUT     CWR             ; OUTPUT TO 8255 #1 CONTROL WORD REGISTER
```



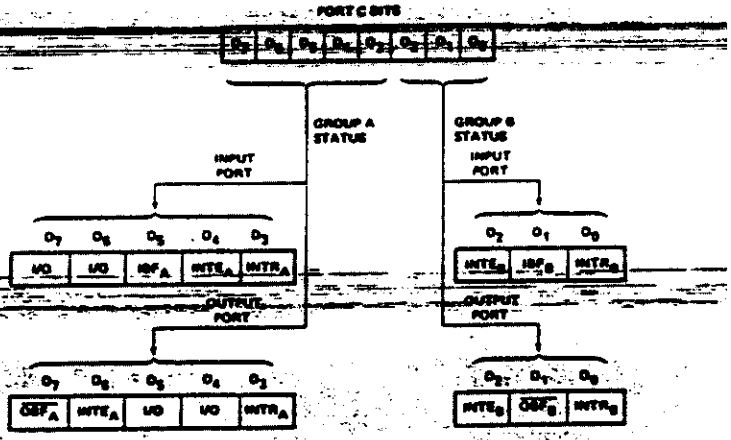
MODE CONTROL WORD



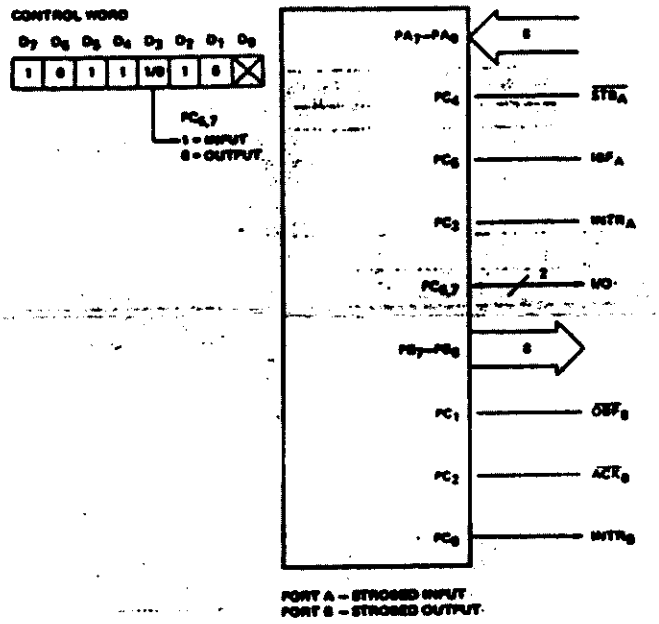
MODE 1 STATUS WORD

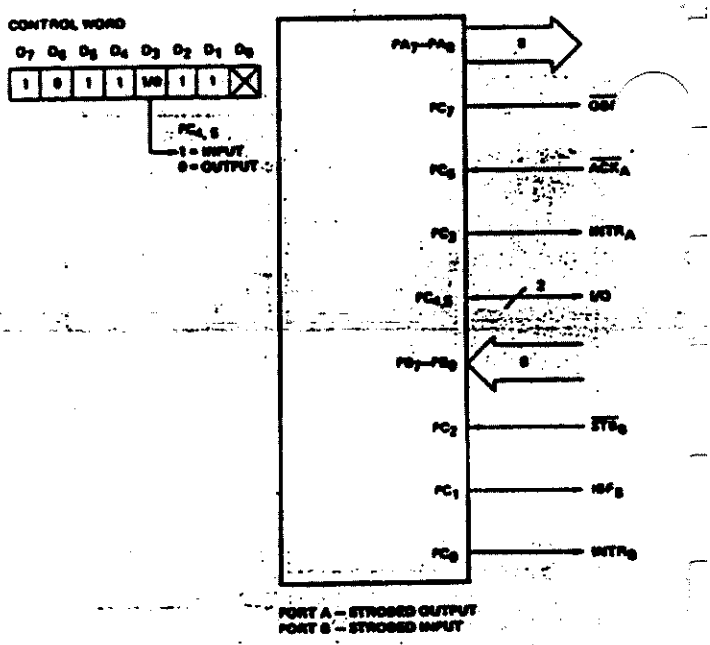


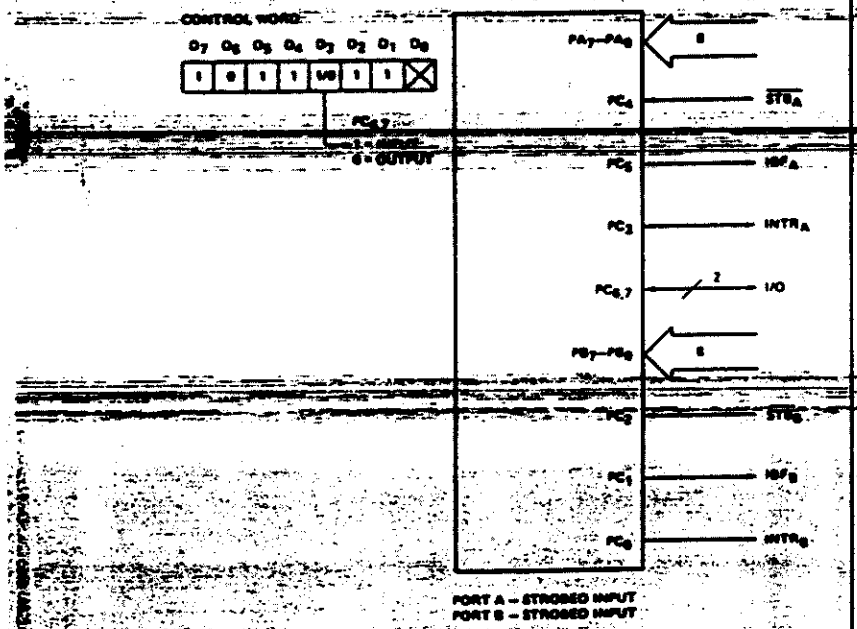
BIT SET/RESET CONTROL WORD

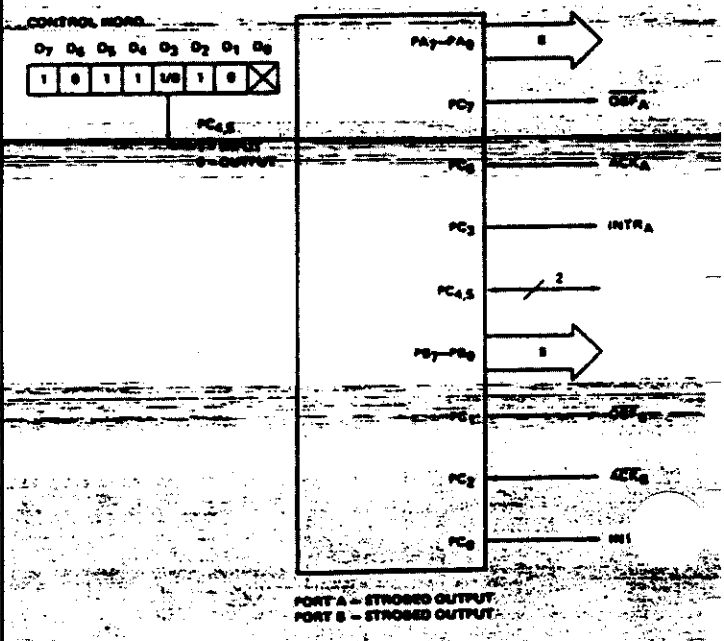


MODE 2 STATUS WORD

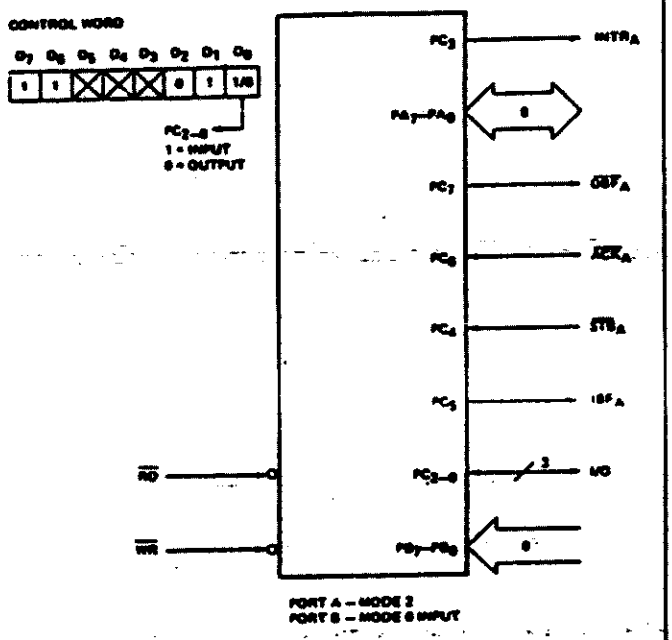

CONTROL WORD
D7 D6 D5 D4 D3 D2 D1 D0
1 0 1 1 I/O 1 0
PC4,7
1 = INPUT
0 = OUTPUT
PA7–PA0
8
PC4
STBA
PC5
IBFA
PC3
INTRA
PC6,7
2
I/O
PB7–PB0
8
PC1
OBFB
PC2
ACKB
PC0
INTRB
PORT A – STROBED INPUT
PORT B – STROBED OUTPUT

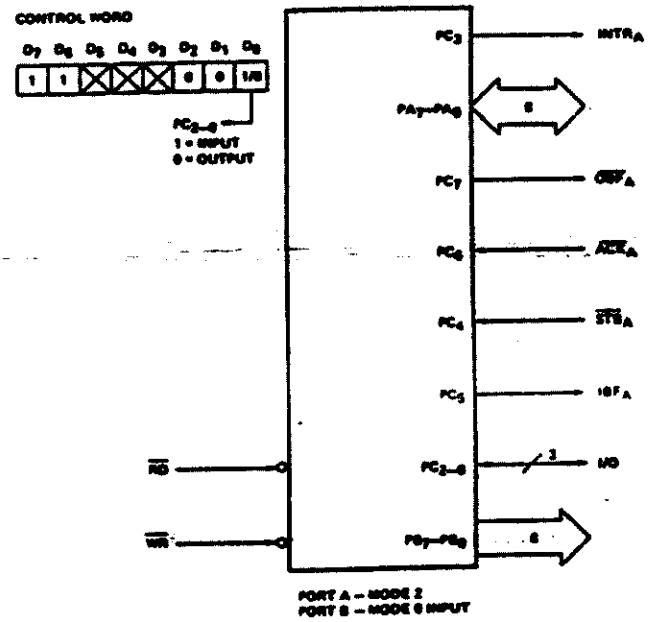

CONTROL WORD
D7 D6 D5 D4 D3 D2 D1 D0
1 0 1 1 I/O 1 1
PC4,5
1 = INPUT
0 = OUTPUT
PA7–PA0
8
PC7
OBFA
PC6
ACKA
PC3
INTRA
PC4,5
2
I/O
PB7–PB0
8
PC2
STBB
PC1
IBFB
PC0
INTRB
PORT A – STROBED OUTPUT
PORT B – STROBED INPUT

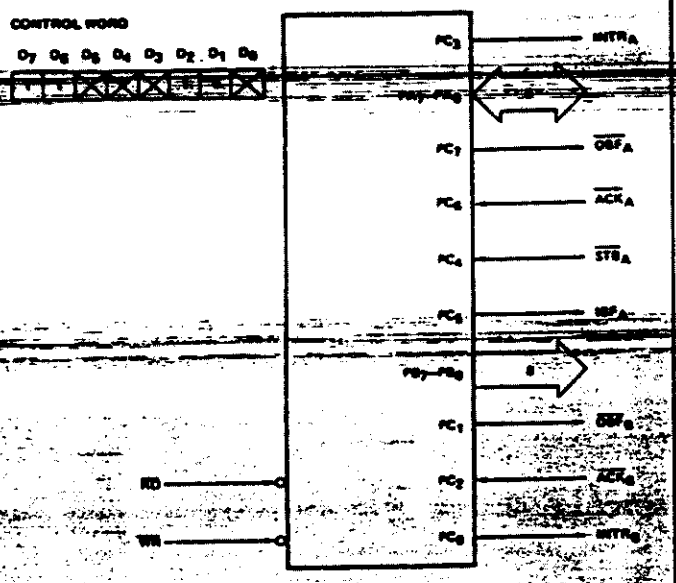

CONTROL WORD
D7 D6 D5 D4 D3 D2 D1 D0
1 0 1 1 I/O 1 1
PC4,7
1 = INPUT
0 = OUTPUT
PA7–PA0
8
PC4
STBA
PC5
IBFA
PC3
INTRA
PC6,7
2
I/O
PB7–PB0
8
PC2
STBB
PC1
IBFB
PC0
INTRB
PORT A – STROBED INPUT
PORT B – STROBED INPUT

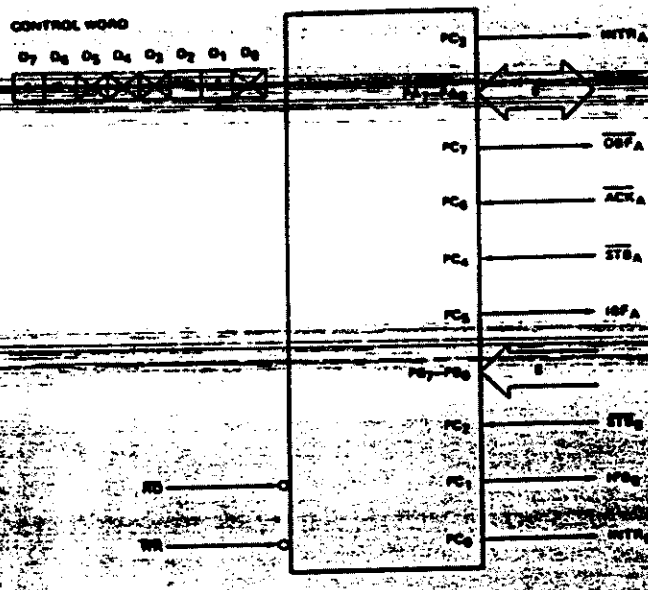

CONTROL WORD
D7 D6 D5 D4 D3 D2 D1 D0
1 0 1 1 I/O 1 0
PC4,5
1 = INPUT
0 = OUTPUT
PA7–PA0
8
PC7
OBFA
PC6
ACKA
PC3
INTRA
PC4,5
2
PB7–PB0
8
PC2
ACKB
PC0
INTR
PORT A – STROBED OUTPUT
PORT B – STROBED OUTPUT

MODE 2 CONFIGURATIONS










XOR