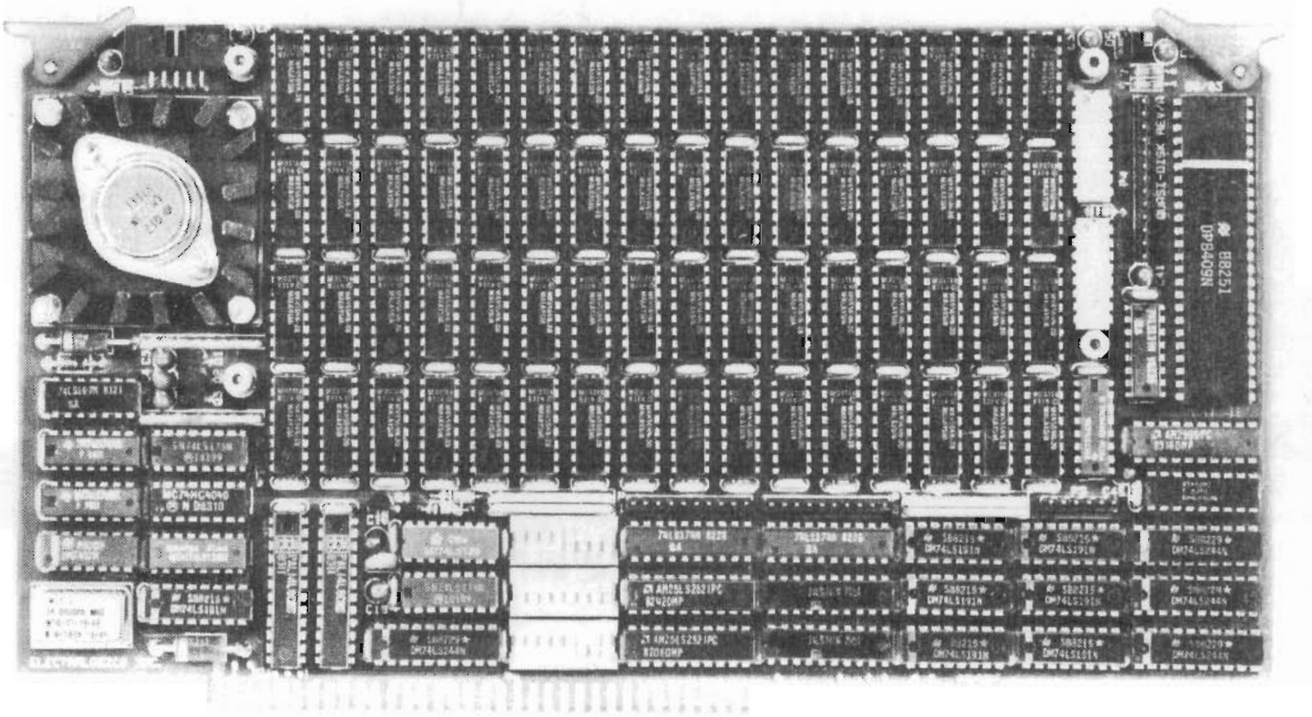
QUASI-DISK

USER'S MANUAL



November 1983.

Revision 1.1

Electra™ logics

Incorporated

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Welcome to the world of QUASI-DISK. This product represents the best engineered enhancement to systems using the traditional floppy disk drive available on the market today.

Every effort has been made to guarantee the quality and reliability of QUASI-DISK from the use of top quality materials, to a 168 hour burn-in before shipping.

This manual has been written to limit the amount of reading required to achieve the task at hand - interfacing QUASI-DISK to your system. We have tried to make the job as straight-forward and simple as possible. If we have not entirely succeeded, please include your comments with the Warranty Registration Form.

Section 1 deals with general information about function, features and operation. We recommend that any first-time user read this section simply to aquaint himself with the ideology behind the product.

Section 2 contains all the information required to get your QUASI-DISK up and running after all the system environmental conditions have been met. This section will prove useful to users who aren't concerned with how, or why, it works, but simply the results.

Armed with the auto-installation software under one arm and the QUASI-DISK under the other, an inexperienced user should be able to initialize the system in a matter of minutes. With a little practice (and periodic reference to this section), using the QUASI-DISK is guaranteed to be a trivial matter in a short time.

Again, we heartily recommend that any first-time users read this section before attempting to install QUASI-DISK on their system. In the long run, it will likely save you a lot of time as well as money spent on long distance calls.

For the user with a technical bent, section 3 deals with the QUASI-DISK board on a register level. If you have plans to incorporate the QUASI-DISK software into your CP/M BIOS, then this section is definitely for you.

Section 4 deals with the software provided by Electralogics for use with QUASI-DISK. Everything from the auto-install routines to the utilities supplied with QUASI-DISK are covered in depth.

Once accustomed to the terrific speed introduced to your system by QUASI-DISK, it stands to reason that an increase in capacity will be on your mind. At that time, section 5 should definitely be read. Use of a small add-on board will facilitate the increase in capacity simply by plugging it in over the existing array of RAMs. QUASI-DISK is capable of supporting up to 128 memory chips in any combination of groups of 8, 64k or 256k, devices. That means up to 4 megabytes of RAM on one card!

One of the major concerns associated with large arrays of semiconductor memory is their susceptibility to the loss of data resulting from power failure. Until now, the only way to protect the data stored in such arrays was to continually back-up the information on a secondary storage medium such as hard, or floppy, disks. Section 6 deals with an optional feature available for QUASI-DISK that virtually eliminates all of these concerns effectively and economically. The battery back-up option is discussed in detail and suggested to any user who would like to be able to turn the system off after use and not worry about losing the data stored on QUASI-DISK.

Following the aforementioned sections come the appendices which are packed with useful information, primarily for the curious. Odds are that you will never need to know how this or that works but it's all there anyway.

We sincerely hope that you enjoy the use of your QUASI-DISK and invite any comments you may wish to make (good or bad, but constructive in nature). If any problems seem to appear, we ask that you reference this manual first, but if, after that effort, there is no light at the end of the tunnel, please give us a call. We will do whatever is necessary (within reason) to ensure that you are pleased with your new purchase.

General Description

The Electralogics QUASI-DISK is a high capacity, semiconductor disk emulator, designed for the S-100 Bus. Addition of this product to computer systems currently using floppy and/or hard disks, results in a dramatic increase in apparent system speed as a result of the elimination of electro-mechanical delays such as head load times, disk rotation times, etc.

Traditionally, small computer systems have relied on the time honoured "floppy disk" for storage of large programs and data bases. The relatively small address space of microprocessors (usually <= 64k for 8 bit systems) forces overlaying of transient programs accompanied by long periods of disk activity to free up memory when possible. The QUASI-DISK may be thought of as an I/O mapped extension to the processor's memory space. Although programs cannot be executed directly while resident in Quasi-Disk, moving them in and out of main RAM is as simple as executing I/O instructions which reference appropriate QUASI-DISK registers. Typically, the CP/M* disk support services are used as a means of organizing and accessing the data while it is passing through QUASI-DISK but "least recently used" algorithms might also be applicable.

Features

- Fully S-100/696 compatible.

- Unique design guarantees that QUASI-DISK will perform as advertized, in standard (as well as non-standard) S-100 systems or your money back.

- 512k bytes of onboard RAM (64k x 1 chips) expandable to 2 Megabytes by plugging in 256k RAMS. (64k and 256k chips may be intermixed in banks of 8 RAM devices.)

- Further expansion is made possible by plugging in an expansion board over the existing board thereby doubling the RAM capacity. If only 64k chips are used, the capacity will increase to 1 megabyte, whereas 256k devices yield 4 megabytes.

- Software error detection is implemented and may be disabled if desired.

- DMA compatible with transfer rates to 2 megabytes/second.

- On board powerfail logic write protects disk during brownouts or power failures to prevent accidental destruction of data.

- Optional battery module to back up QUASI-DISK is available and provides protection for 2 hours. This option includes battery charging logic, circuitry to prevent accidental battery damage which may result from "deep discharge" as well as a power supply to support the data when the host system is turned off.

- Six layer board improves performance and reliability by reducing board noise levels, an important feature for products using dynamic RAMs.

- Completely orthogonal register set - i.e.; all registers are read/write.

- Requires only 6 I/O addresses to access entire board.

- Supports extended I/O addressing when enabled.(ie. A0-A15)

- On board 22 bit address generator may be programmed for auto increment or decrement if desired.

- Any sector size may be implemented.

- Burst transfers of up to 4 megabytes of data (or entire board capacity) are possible because the entire 22 bit address is incremented/decremented as programmed on each data access.

- * Onboard LED's indicate "drive active" and "powerfail status".

- * Includes installation software on 8" SS/SD diskette with all source code supplied.

- * Sample CP/M# BIOS routines are included for integration into any CP/M 2.2 system.

- * Print spooler software also supplied, supporting dynamic allocation of spooler sectors for printer buffering.

- * QUASI-DISK is covered by a 1 year warranty. An extended warranty option is also available. See manual for more details.

CP/M is a registered trademark of Digital Research Incorporated.

INSTALLING YOUR QUASI-DISK

Assuming that you have just received your QUASI-DISK, and that you are anxious to give it a try, you're at the right section of the manual. In this chapter, the requirements that QUASI-DISK places on your system will be examined followed by the steps required to set up the board hardware. Once all this has been completed, you're ready to give the software a try. The final section of this chapter is designed to explain and demonstrate the software installation procedure. Examples are included to show you what you should expect to see on your terminal.

Before proceeding any further, please make sure that you have done the following things:

a) Make a ***BACK-UP*** copy of the software disk supplied with your QUASI-DISK. The supplied routines are on a single-sided/single-density diskette (IBM 3740 format) which is write protected. Remember, this is the only one you've got and it will take weeks to get another (if we use the mails, it could take longer!).

b) Please fill out the warranty registration form supplied at the beginning of this document. If having the product repaired (should it break down), or receiving manual updates are of any interest to you, do this now!

c) Please read the user's manual at least to the end of this section before attempting to actually install your QUASI-DISK. The small amount of time spent now could save you hours later.

***** GOOD LUCK! *****

Required Hardware Environment

QUASI-DISK makes no real demands on any target system but several points do seem worthy of mention.

Firstly, on the QUASI-DISK printed circuit board, edge connector fingers numbered 20, 53 and 70 are all tied to ground. Some older systems (eg. Cromemco, NorthStar) use these lines for special signals (such as "data disable*" on pin 53). Please be sure that your system does not use these lines for anything other than ground. Failure to do so could damage QUASI-DISK and/or your system. If these lines are required by your system, simply cut the exposed traces on the QUASI-DISK printed circuit board with a sharp knife. These traces are additional grounds as specified in the S-100/696 specification, and don't really affect operation of the board.

Measure the voltage present on the +8 Volt rail of the S-100 bus and ensure that it stays above +8 Volts. Onboard powerfail logic may be triggered if this supply fails too far below +8 Volts.

Hardware Setup Procedure

************* Note *************

After carefully unpacking QUASI-DISK, be sure to ground your body and the anti-static bag used to protect the board. Failure to do this may result in a damaged or unreliable product. Remember, static is QUASI-DISK's worst enemy. Some of the damage created by electro-static discharge may not be apparent immediately but may surface later as intermittent malfunction and/or logic failure.

Unpacking and Inspection

Remove QUASI-DISK from its protective bag, being sure to keep both you and the board/bag grounded. Visually examine and inspect the board to ensure that all IC's are properly inserted in their sockets, pins are not bent, parts not broken, etc. On the back of the board, look for shorts caused by IC leads touching each other. Remember, the board was likely shipped many miles to get to you, and probably not in the first class section!

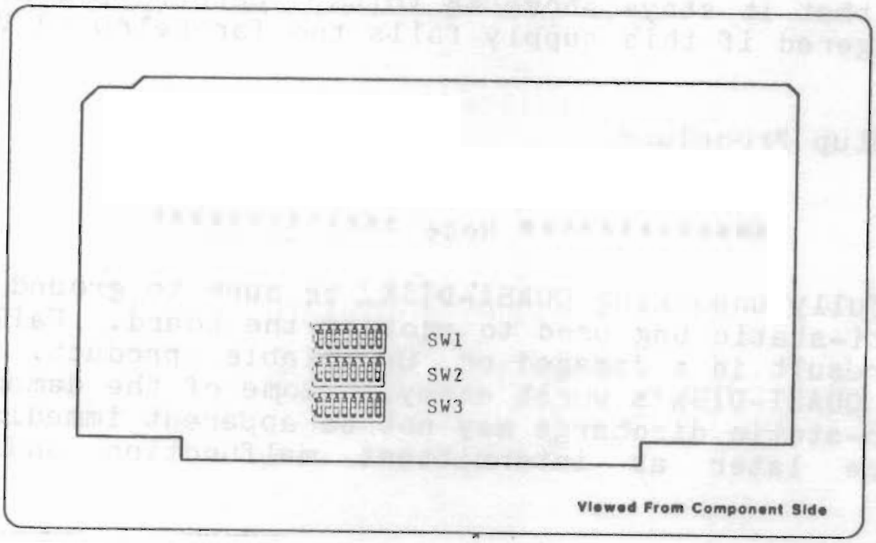
Each QUASI-DISK has been burned-in for 168 hours and then tested before leaving the factory. It is highly unlikely that any malfunction will be the result of component failure but rather will be due to rough handling during shipping or failure to follow instructions regarding protection from electro-static discharge. If you are not satisfied with your QUASI-DISK, please contact the factory.

Switch Locations and Conventions

Locate the three dip switches used to configure the board for your system. Holding the board with the connector down, switch #1 is the top switch, switch #2 is in the middle and switch #3 is at the bottom, closest to the 100 pin connector fingers. The diagram below provides the appropriate details.

Also note that the following convention is applied and assumed when dealing with switches to establish binary values:

"on" = logic 0
"off" = logic 1



Switch Locations

All references to these switches will be made as follows:

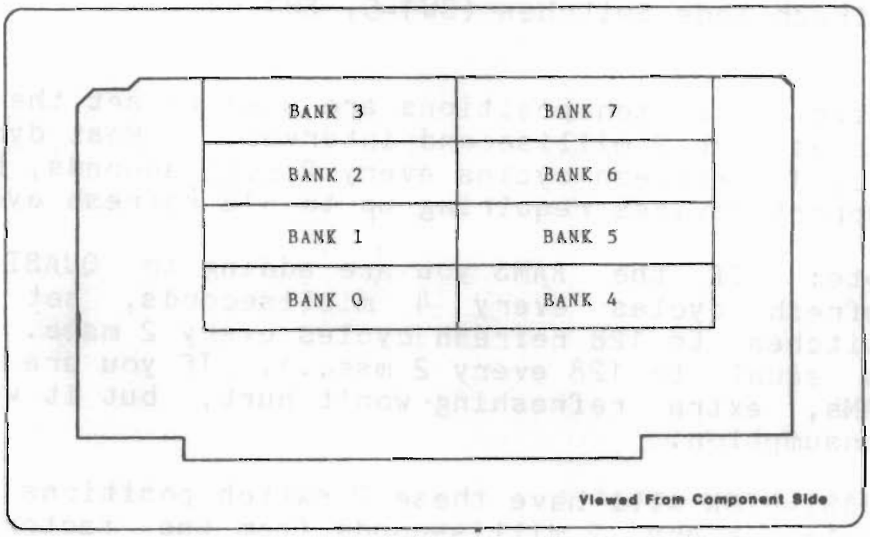
```
                          SWX - Y
                            ^   ^
                            |   |
Switch  #(1,2,3)---------------------Switch  position (1-8)
```

Note that switch position 1 on all three switches is the left-most switch position.

Capacity Switches (SW1-1 to SW1-4)

These four switch positions are used to configure QUASI-DISK for 64k and/or 256k dynamic RAMS. The binary value programmed into these switches modifies on-board logic to accommodate a mixture of these two different RAM devices. Upon installation the capacity of the board is calculated and the new disk parameter block modified to reflect the total QUASI-DISK capacity. To you the user, this means that when you decide to increase the capacity of your board, you need not re-install or re-generate the code supplied; simply change these capacity switches to reflect the number of 256K banks, execute the install program and the system does the rest.

Note - When replacing a bank of 64k devices with 256k devices, it is important to begin in bank 0 and move towards bank 7. (A bank is defined as 8 RAM chips yielding 64k x 8 or 256k x 8). When using only 64K devices (typical configuration) SW1-1 to SW1-4 should all be "on", whether the expansion module is present or not.



Viewed From Component Side

Memory Map

As mentioned previously, the binary value programmed into these four switch positions must indicate the number of 256k banks installed on your board. Thus, if no 256k chips are present, these 4 switch positions should all be set to "0" (ie, all "on") yielding binary 0000 (ie. no 256k banks installed). Note the significance of these 4 switch positions:- "0" = ON and "1" = OFF. This is the factory setting you will find on your board if it was ordered with 64k devices.

If you wish to use 256k devices (8 devices) in one bank, they must be loaded in bank 0 (see above diagram) and the capacity switches set to reflect a binary "0001". In other words, SW1-1 to SW1-3 should all be "on" and SW1-4 should be OFF. Note that you can't damage the board or RAMS by setting the switches incorrectly however, the board will not function properly.

The following table will help determine switch settings for various capacities:

SW1-1	SW1-2	SW1-3	SW1-4	# of 256k Banks	Total Capacity
ON	ON	ON	ON	0	512k
ON	ON	ON	OFF	1	704k
ON	ON	OFF	ON	2	896k
ON	ON	OFF	OFF	3	1088k
ON	OFF	ON	ON	[✓ 4]	1280k
ON	OFF	ON	OFF	5	1472k
[✓ ON]	[✓ OFF]	[✓ OFF]	[✓ ON]	[✓ 6]	1664k
ON	OFF	OFF	OFF	7	1856k
OFF	ON	ON	ON	8	2048k

Capacity Switch Settings

Refresh Mode Switches (SW1-5, SW1-6)

These two switch positions are used to set the number of refresh cycles per 2 millisecond interval. Most dynamic RAMs require only 128 refresh cycles every 2 milliseconds, but QUASI-DISK will support devices requiring up to 512 refresh cycles/2msec.

Note: If the RAMS you are adding to QUASI-DISK require 256 refresh cycles every 4 milliseconds, set the refresh mode switches to 128 refresh cycles every 2 msec. (256 every 4 msec is equal to 128 every 2 msec.). If you are unsure about your RAMs, extra refreshing won't hurt, but it will increase power consumption.

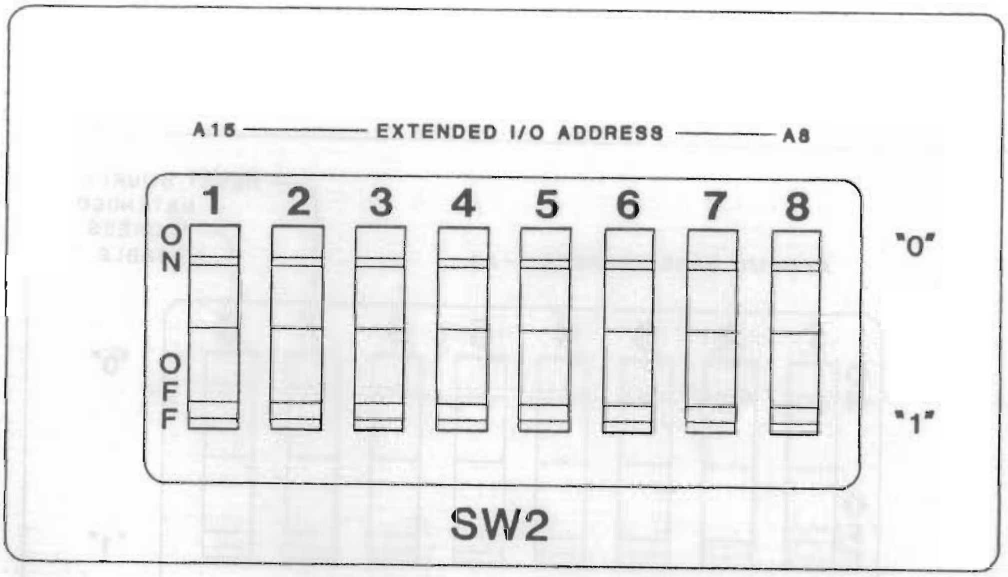
QUASI-DISK will have these 2 switch positions set for 128 refresh cycles every 2 milliseconds from the factory. The following table will help determine the switch settings for various refresh modes:

SW1-5	SW1-6	#Of Refreshes/2 Milliseconds
[✓ ON]	[✓ ON]	128
ON	OFF	512
OFF	ON	256
OFF	OFF	128

Refresh Switch Settings

Extended I/O Address Switch (SW2-1 to SW2-8)

This switch is used to set the extended address to which the board will answer if extended I/O addressing is enabled. (See section 2.2.8.) The extended I/O address must appear on A8-A15 of the S-100 bus during I/O instructions. The following diagram may prove useful in establishing the extended address for your board.



"on" = logic 0
"off" = logic 1

Extended I/O Address Switch (SW2-1 to SW2-8)

Therefore, to set the extended I/O address to 91H, the switches must be set as follows:

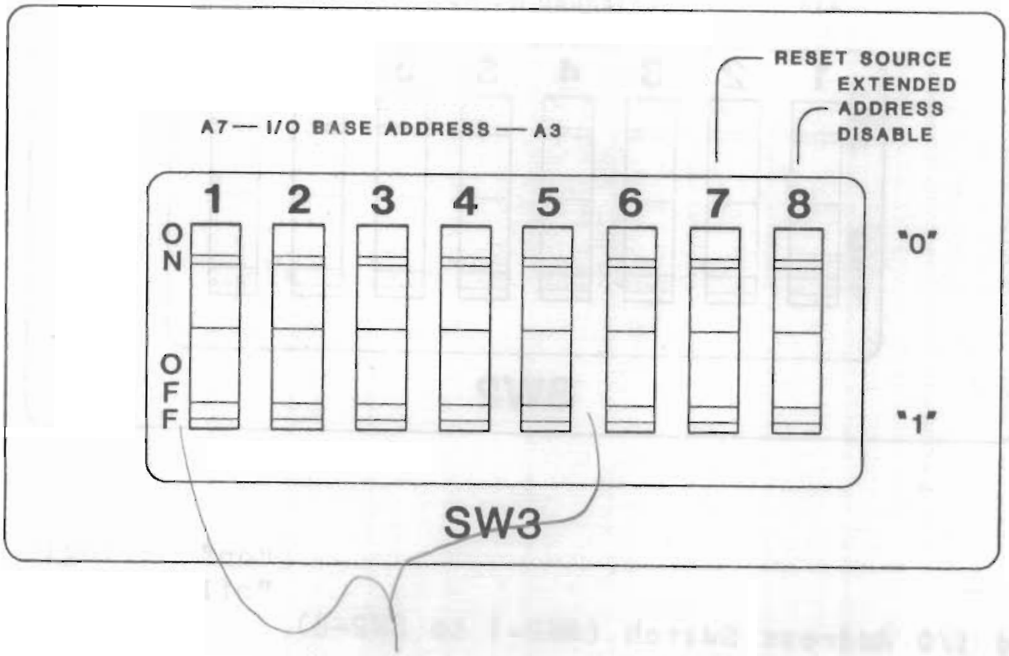
SW2-1	-2	-3	-4	-5	-6	-7	SW2-8
(A15)	(A14)	(A13)	(A12)	(A11)	(A10)	(A9)	(A8)
OFF	ON	ON	OFF	ON	ON	ON	OFF

Sample Extended Address Setting

These switches will always be factory set to "FFH", i.e.; all "OFF".

I/O Base Address (SW3-1 to SW3-5)

These switch positions are programmed to place the QUASI-DISK into any 8 byte I/O address block within the 256 byte I/O page. Note that SW3-1 corresponds to A7 while SW3-5 corresponds to A3. A0-A2 are not compared in this operation because they are used to select 1 of 8 registers within the 8 register I/O block internally. The following diagram will help in setting these switch positions:



"on" = logic 0
"off" = logic 1

Therefore, if QUASI-DISK is to respond to I/O addresses in the range of B0H to B7H the switch positions must be set as follows:

1	0	1	0	1
SW3-1 (A7)	SW3-2 (A6)	SW3-3 (A5)	SW3-4 (A4)	SW3-5 (A3)
=====	=====	=====	=====	=====
OFF	ON	OFF	OFF	ON

Sample I/O Base Address Setting

Reset Source Switch (SW3-7)

Since QUASI-DISK is an I/O slave device, it will always be reset by applying SLAVE CLEAR on the S-100 bus (pin 54). Some systems, however, do not support the SLAVE CLEAR line and it is in these systems that this switch position becomes important. Turning SW3-7 "ON" allows QUASI-DISK to be reset from either the RESET line or the SLAVE CLEAR line of the S-100 bus. Turning SW3-7 "OFF" returns QUASI-DISK to the S-100/696 specification. (ie. only SLAVE CLEAR is able to reset the board). If you are in doubt as to whether or not you system supports SLAVE CLEAR, turn this switch "ON" for the time being. The factory setting for this switch is "OFF".

Extended I/O Address Disable Switch (SW3-8)

In systems supporting the S-100-696 convention for extended I/O addressing, the QUASI-DISK extended I/O address may be set in SW2 (see section 2.2.5.). To enable the extended I/O address option (and thus make SW2 operational), the extended address I/O disable switch must be "OFF". If the disable switch is "ON", QUASI-DISK will ignore SW2 setting and respond only to the address which appears on A0-A8 of the S-100 bus. In this case, the I/O base address should be set on SW3-1 to SW3-5 (see section 2.2.6.). The factory setting for this switch is "ON".

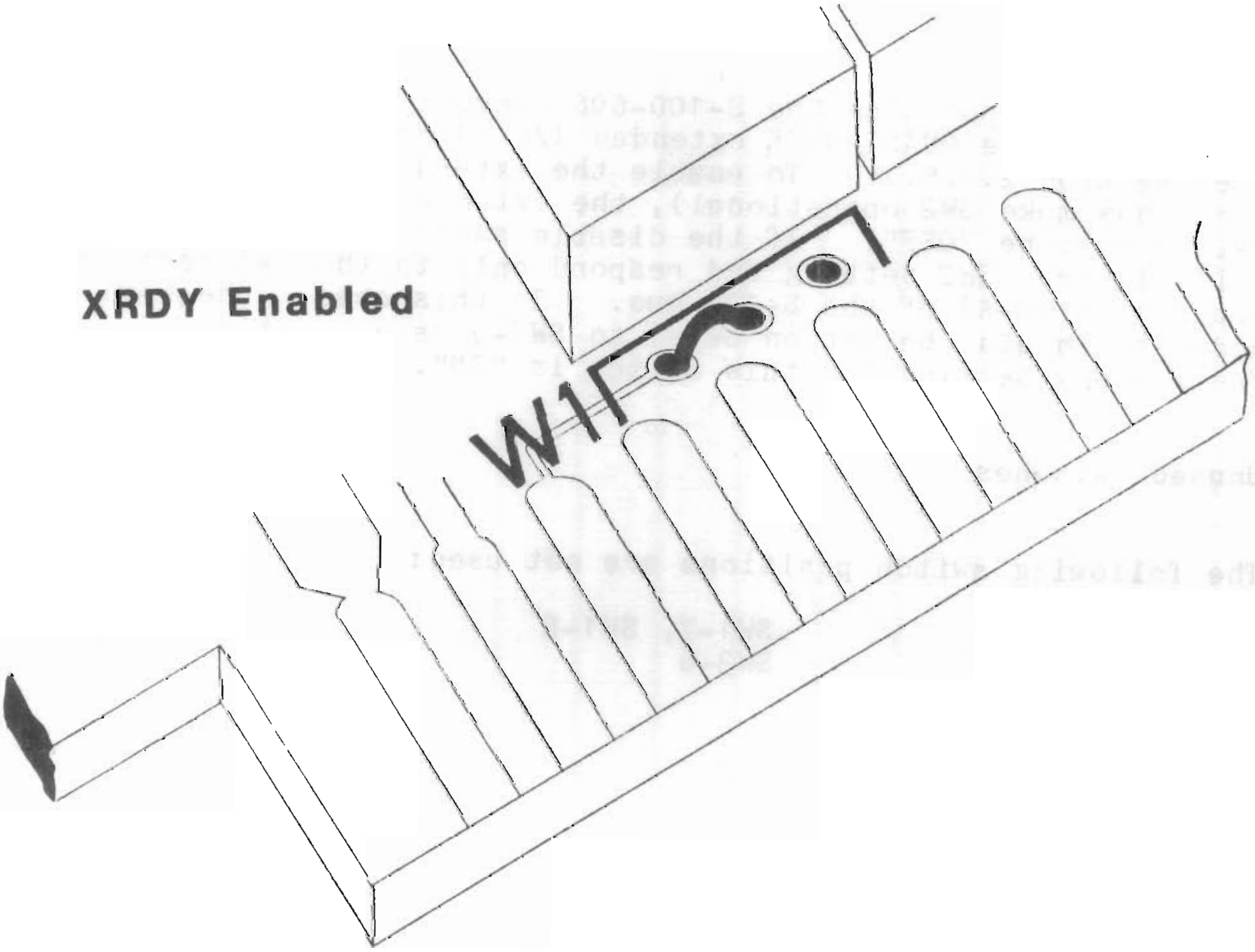
Unused Switches

The following switch positions are not used:

SW1-7, SW1-8
SW3-6

Ready Select Jumper Option (W1)

This jumper option is located just above the edge connector fingers to the right of the Xtal oscillator in the lower left corner of board). All S-100/696 systems expect the READY/NOT READY signal to be sent over the bus signal line called RDY (pin 72). However, some older systems such as Altair, and possibly Imsai, are in conflict when RDY is used for this purpose. They expect the ready signal on Pin 3, XRDY. Should you suspect this problem with your system, you may want to change the setting of W1. As shipped, W1 is set to the S-100/696 standard and uses RDY, pin 72. To enable pin 3, XRDY, cut the fine trace located on the back of the board, then solder a small piece of jumper wire as shown.



Installation of Jumper W1

Generally speaking, this will not be a problem and we recommend extreme care before making the above alterations. Make sure that your system really does have a problem with the RDY line because changing this jumper will sometimes generate all sorts of other problems.

Required Software Environment

The software package included with QUASI-DISK is a tool to allow first time users to test and evaluate the product quickly and easily. Installation is as simple as executing a configuration program and establishing the operational parameters you require. The software provided has been written to support the general user and sooner or later you may want to modify the enclosed routines.

The software supplied with QUASI-DISK is written in 8080 assembler and all modifications to the code must follow that convention. This is required to allow the relocating loader to function correctly.

All QUASI-DISK software is quaranteed to run under CP/M 2.0 or higher. Tests on CP/M 3.0 are still under way. When a release is available supporting CP/M 3.0, you will be automatically copied provided that you have registered your warranty.

Please note that the auto-instalation software provided with QUASI-DISK modifies the BIOS vector table. If your system has these vectors in EPROM (as may be the case in some single board computers) then the installation software may not succeed in loading the Qcode. In that case you would have to modify the BIOS and install the necessary code.

Additionally, if the print spoolers are to be used successfully, please ensure that the host system BIOS implements a list status routine. This is a routine called by the operating system to check for lister busy status without locking up the system if the response is "not ready". Samples of this routine are included in the BIOS extensions provided on the distribution diskette.

Software Installation

Assuming that the QUASI-DISK board has been inserted into the target system and all necessary hardware parameters established, you are now ready to give it a try. The following section of the manual has been designed to "walk" you through your first attempt at getting QUASI-DISK operational. PLEASE, read the manual from the beginning to the end of this chapter before actually installing the device. There is just a little more work to do before actually giving it a try. Be patient, success is just around the corner.

What Do I Do After Hitting System Reset?

The first thing you should do is to make sure that a duplicate copy of the supplied software diskette has been made, and the original filed away for safe keeping. Once this has been done, we are prepared to start.

Place the duplicate diskette into your system drive (A, B or whatever...). Have a look at the directory. This is what you should see:

```
Filename.Typ Size K   Filename.Typ Size K   Filename.Typ Size K
-------- --- -------  -------- --- ------   -------- --- ------
-README  .DOC      7  QCOPY1   .COM      2  QMAKE1   .COM      8
NSWEEP   .COM      7  QDU1     .ASM     23  QS1      .ASM     11
QCOPY1   .ASM     10  QMAKE1   .ASM     35  QS1      .COM      2
```

Directory Listing of Supplied Disk

Following is a brief description of the functions associated with each of the above programs:

QMAKE*.COM - when executed, customizes the QUASI-DISK software and generates a loader program which will be used to bring up the new drive. This routine prompts the user for information and performs tests on QUASI-DISK to ensure that the hardware parameters are correctly established.

QS*.ASM - supports the spooler which is included in your QUASI-DISK package.

QCOPY*.ASM - copies the contents of the print spooler to a user specified file.

QDU*.ASM - is a powerful utility which performs all of CP/M's* CCP comands except loading of a program. Some additional commands allow generation of a memory map, etc.

First, select a vacant I/O address block (8 contiguous I/O bytes) in your system. Once this has been done, the QDU*.ASM utility will have to be re-assembled with the equate labelled "QBASE" set equal to the selected I/O address. This is the only module that requires the change to QBASE since all other utilities are capable of finding the QUASI-DISK on their own.

Once this has been done, there will be no need to repeat the procedure as long as QUASI-DISK's I/O address is not altered.

Edit the file called QDU*.ASM and modify the parameter called "QBASE" so that both the QUASI-DISK I/O Base Address (set up in SW3-1 to SW3-5) and "QBASE" are the same. For now, let's assume that you have chosen 80H as your I/O base address.

Changing the "QBASE" equate to 80H and re-assembling, should provide you with a .HEX file which can be loaded to generate the ASM or MAC. Following the LOAD process, erase all .SYM, .HEX and

Your disk directory should now contain the following files:

```
Filename.Typ Size K  Filename.Typ Size K  Filename.Typ Size K
-------- --- ------  -------- --- ------  -------- --- ------
-README  .DOC     7  QDU1     .ASM    23  QMAKE1   .COM     8
NSWEEP   .COM     7  QDU1     .COM     4  QS1      .ASM    11
QCOPY1   .ASM    10  QMAKE1   .ASM    35  QS1      .COM     2
QCOPY1   .COM     2
```

Directory of Re-Assembled and Loaded Files

Now we are ready to attempt an installation on your system!

Installation Procedure - Sample Run

The following sample run assumes that the QUASI-DISK software diskette is loaded in Drive B.

Comments contained within asterisks are included to explain the significance of the data appearing above the comment lines. All the QMAKE* options are exercised simply to indicate the possible responses to user input. In an actual installation, not all options need be set.

```
                    Beginning of sample run.
------------------------------------------------------------------

B>qmake1

******************************************************************
*    Execute QMAKE*.COM                                          *
******************************************************************

QMAKE Version 1.1  -  (C) 1983 Electralogics Inc.


                     QUASI-DISK CONFIGURATOR
                     -----------------------

        This program will display the present Quasi-Disk
        configuration and allow you change it. The changes
        will have no immediate effect. By selecting the
        'Create QGO1' command, a program called QGO1 will
        be generated on the logged disk and may then be run
        to install and configure QUASI-DISK. Please note
        that this version uses a fixed sector size of 128
        bytes and will only work with CP/M 2.0 to 2.2

Please press RETURN to continue ==>

******************************************************************
*    QMAKE* sign-on message.                                     *
******************************************************************

    MENU AND CURRENT DEFINITIONS
    ---------------------------

        1)  Port address . 00

        2)  Disk name .... D

        3)  Qcode location:0000

        4)  Spooler: is  included

        5)  Create QGO*

        Q)  Quit

        ?)  Help!

Please select a function (1-5,Q or ?) ==> ?

******************************************************************
*    QMAKE* menu - "?" requests help                             *
******************************************************************
```

```
QMAKE Version 1.1  -  (C) 1983 Electralogics Inc.

                        QUASI-DISK CONFIGURATOR
                        -----------------------

        This program will display the present Quasi-Disk
        configuration and allow you change it. The changes
        will have no immediate effect. By selecting the
        'Create QGO1' command, a program called QGO1 will
        be generated on the logged disk and may then be run
        to install and configure QUASI-DISK. Please note
        that this version uses a fixed sector size of 128
        bytes and will only work with CP/M 2.0 to 2.2

Please press RETURN to continue ==>
```

```
*****************************************************************
*    At this point, help returns us to the QMAKE* sign-on.      *
*****************************************************************
```

```
    MENU AND CURRENT DEFINITIONS
    ----------------------------

        1)  Port address . 00

        2)  Disk name .... D

        3)  Qcode location:0000

        4)  Spooler: is  included

        5)  Create QGO*

        Q)  Quit

        ?)  Help!

Please select a function (1-5,Q or ?) ==> 1
```

```
*****************************************************************
*    Back at the menu, entering "1" indicates to QMAKE* that we *
*    want to change the I/O base address currently set to 0.   *
*****************************************************************
```

```
Quasi-disk port address? ==> ?

        The port address should be in hexadecimal and on
        an 8-byte boundary.  Examples of good addresses are:
        0,8,10,. . F0,F8.  The address you select must match
        the switch 1 setting on your QUASI-DISK board.
```

```
*****************************************************************
*    Entering a "?" here supplies information about port       *
*    addressing on QUASI-DISK.                                  *
*****************************************************************
```

```
Quasi-disk port address? ==> 80

*********************************************************************
*    Although the QUASI-DISK is set to address B0h, lets try to *
*    install it at address 80h.                                     *
*********************************************************************

The selected port address can not access Quasi-Disk.
Possible problems:
-  The selected address does not match the address set
   on switch 1 on the Quasi-Disk Board.
-  Another port is sharing the same I/O address.
-  A power-fail condition exists permanently on the bus
   due to a marginal +8V power supply. This condition is
   indicated by the red LED on Quasi-Disk. Consult your
   manual for proper action

Want to retry? Y/N ==> y

*********************************************************************
*    The installation program attempts to verify the existance   *
*    of a QUASI-DISK at the address 80h, but can't find one.      *
*    If the address switches were incorrectly set on the board   *
*    or another I/O device is competing with QUASI-DISK at the    *
*    selected I/O address, this is the message you will see.     *
*    Check your address switches and refer to section 2.2.6 of   *
*    this manual.                                                   *
*********************************************************************

Quasi-disk port address? ==> B0

*********************************************************************
*    The port address is set to B0.                                 *
*********************************************************************

    MENU AND CURRENT DEFINITIONS
    ----------------------------

        1)  Port address . B0

        2)  Disk name .... D

        3)  Qcode location:0000

        4)  Spooler: is  included

        5)  Create QGO*

        Q)  Quit

        ?)  Help!

Please select a function (1-5,Q or ?) ==> 2
```

```
*******************************************************************
*    Now let's try to change the disk name - enter "2" as our     *
*    menu choice.                                                 *
*******************************************************************

Quasi-disk drive identifier? (A to P) ==> ?

        Quasi-Disk may be installed as any drive. (A thru P)
        Care should be taken to ensure that the BIOS implemented
        in your system is capable of supporting the selected
        drive I.D.  Note that if an existing I.D. is chosen
        that drive will be replaced by QUASI-DISK when QGO1.COM
        is executed.

Quasi-disk drive identifier? (A to P) ==> C

*******************************************************************
*    Entering the "?" again supplies the help information, and    *
*    finally we settle on leaving the QUASI-DISK as drive C.      *
*******************************************************************

    MENU AND CURRENT DEFINITIONS
    ----------------------------

        1)  Port address . B0

        2)  Disk name .... C

        3)  Qcode location:0000

        4)  Spooler: is  included

        5)  Create QGO*

        Q)  Quit

        ?)  Help!

Please select a function (1-5,Q or ?) ==> 3

Code size is 0260H

Code location in hex? (0 for below CCP) ==> ?

*******************************************************************
*    Choosing "3" indicates to QMAKE* that we want to modify      *
*    the load address of the BIOS extension which is used to     *
*    support QUASI-DISK. Again, "?" asks for help.                *
*******************************************************************
```

```
               The best location for Qcode is above your BIOS.
               That puts it out of reach of delinquent programs.

               If you specify 0, then QGO1 will locate the Qcode
               on a 100H page boundary below your CCP.

               If you specify a location other than 0, it should be
               above your BIOS and it can be on any boundary.

               Note that if you have changed the location
               of the code while it was already installed, you
               must remove the old code via a cold boot before
               running QGO1 again.

               Code size is the length of the code that will be
               moved by QGO1 to the location you specify.


Code size is 0260H

Code location in hex? (0 for below CCP) ==> 0

************************************************************
*    Lets leave the BIOS extension below the CP/M CCP. In   *
*    other words, enter "0".                                *
************************************************************

    MENU AND CURRENT DEFINITIONS
    ----------------------------

        1)   Port address . B0

        2)   Disk name .... C

        3)   Qcode location:0000

        4)   Spooler: is  included

        5)   Create QGO*

        Q)   Quit

        ?)   Help!

Please select a function (1-5,Q or ?) ==> 4

************************************************************
*    Back at the QMAKE* menu, lets play with the spooler. Enter *
*    a "4".                                                 *
************************************************************
```

```
Do you want the spooler implemented? Y/N ==> ?

        Excluding the print spooler will add an additional
        64K to Quasi-Disk.
        Exclusion will not reduce the Qcode size

Do you want the spooler implemented? Y/N ==> y

*****************************************************************
*    Following the help information, we decide to leave the     *
*    spooler support intact. Enter "Y".                         *
*****************************************************************

    MENU AND CURRENT DEFINITIONS
    ----------------------------

        1)  Port address . B0

        2)  Disk name .... C

        3)  Qcode location:0000

        4)  Spooler: is  included

        5)  Create QGO*

        Q)  Quit

        ?)  Help!

Please select a function (1-5,Q or ?) ==> 5

*****************************************************************
*    Now we are ready to create the QGO*.COM file required to   *
*    load the BIOS extension below the CCP as we requested.     *
*    When we're done with QMAKE*, executing QGO*.COM will bring *
*    up QUASI-DISK as drive C on your system.                   *
*****************************************************************

Create QGO1 and update QMAKE1? Y/N ==> ?

                CREATE QGO ON CURRENT DISK
                --------------------------

        This step will erase the previous copy  (if any)
        of QGO1.COM on the logged drive and relace it with
        a new version containing your current definitions.
        It will also copy the changes to QMAKE1.COM and save you
        re-entering the new configuration the next time you run
        QMAKE1. After this step, installation is completed by
        executing QGO1.COM. If all goes well, you will have added an
        additional drive to your system which behaves very much
        like the existing drives with two exceptions, Quasi-
        Disk will be much faster and quieter!
```

```
Create QGO1 and update QMAKE1? Y/N ==> y

        QGO1.COM was created on the current disk and your
        definitions were saved on QMAKE1.COM.  Good luck!

*******************************************************************
*    After reading the "help" information, we create QGO*.COM    *
*******************************************************************

Press any key to continue

    MENU AND CURRENT DEFINITIONS
    ----------------------------

        1)  Port address . B0

        2)  Disk name .... C

        3)  Qcode location:0000

        4)  Spooler: is  included

        5)  Create QGO*

        Q)  Quit

        ?)  Help!

Please select a function (1-5,Q or ?) ==> Q

*******************************************************************
*    Exit QMAKE* and return to CP/M. We should be ready to       *
*    attempt an installation of QUASI-DISK.                      *
*******************************************************************

B>x

Filename.Typ Size K   Filename.Typ Size K   Filename.Typ Size K
-------- --- ------   -------- --- ------   -------- --- ------
-README  .DOC      7   QDU1     .ASM    23   QMAKE1   .COM      8
NSWEEP   .COM      7   QDU1     .COM     4   QS1      .ASM     11
QCOPY1   .ASM     10   QGO1     .COM     3   QS1      .COM      2
QCOPY1   .COM      2   QMAKE1   .ASM    35

*******************************************************************
*    The directory listing now shows a copy of QGO*.COM which    *
*    must be executed to load the BIOS extension.                *
*******************************************************************
```

```
B>QGO1

QGO Version 1.1   -  (C) 1983 Electralogics Inc.

1024 K bytes
CD06 CD99 formatting.....Disk C active

******************************************************************
*    Well, here it is. After executing QGO*.COM the system       *
*    responds by formatting QUASI-DISK and after a short delay  *
*    you should see the "Disk C Active" message appear. QUASI-  *
*    DISK is now operational. Lets perform a few tests.          *
******************************************************************

B>c:

******************************************************************
*    Log on to drive C.                                          *
******************************************************************

C>x

Directory Empty

******************************************************************
*    Examining the directory shows no files as we would expect. *
******************************************************************

C>a:pipv c:=a:qma:=b:qmake1.asm

C>x

Filename.Typ Size K   Filename.Typ Size K   Filename.Typ Size K
-------- --- ------   -------- --- ------   -------- --- ------
QMAKE1  .ASM     36

******************************************************************
*    Transfer a file (in this case QMAKE*.COM) to drive C. The  *
*    directory indicates a successful transfer.                  *
******************************************************************

C>stat a:stat dsk:

    C: Drive Characteristics
 7600: 128 Byte Record Capacity
  950: Kilobyte Drive  Capacity
  128: 32  Byte Directory Entries
    0: Checked  Directory Entries
  128: Records/ Extent
   16: Records/ Block
  255: Sectors/ Track
    0: Reserved Tracks
```

```
C>a:stat
A: R/W, Space: 820k
C: R/W, Space: 910k

*****************************************************************
*    Finally, STAT shows us that we have 910K left to play with.*
*    Remember, we used part of the drive capacity when we       *
*    transfered QMAKE*.COM to QUASI-DISK as well as 64K for the *
*    print spooler.                                             *
*****************************************************************
```

```
-----------------------------------------------------------------
                      End of sample run.
```

HOW TO USE THE QUASI-DISK UTILITIES

This section of the manual contains functional descriptions and guides for the use of the QUASI-DISK Spooler and the following utilities:

a) QS* .COM

b) QDU* .COM

c) QCOPY*.COM

New utilities are being periodically written and distributed to registered QUASI-DISK owners. Provided that you have returned your Warranty Registration to Electralogics, we will make every effort to keep you up to date.

Note:- Before QDU* can be used, it must be re-assembled with the "QBASE" equate set to the same value as the I/O base address programmed on the QUASI-DISK board. If you don't remember where you set the address, refer to the QMAKE* program and make note of parameter #1. If you are still uncertain, refer to the following sections of this manual:

a) Section 2.2.2. - Switch Locations and Conventions

b) Section 2.2.6. - I/O Base Address

c) Section 2.4.1. - What To Do After Hitting System Reset?

A Description of The Print Spooler

Print spooling is transparent to the user. When using the system with the spooler enabled, the only noticeable difference in the operation of your system will be the ability to run other applications while the printer is printing.

With the spooler enabled, every character sent to the printer is routed to the print spooler buffer. Whenever your system is waiting for keyboard activity, the printer checks the spool buffer and prints any data found there. The spool buffer is set up as a 64k first-in-first-out (FIFO) buffer. As long as the capacity of the buffer is not exceeded, your system will run normally while printing is in progress. If there is more print data than the spool buffer can hold, your system will wait until the printer has printed enough data to clear the "buffer full" condition.

Words of Wisdom - Some Precautions

At this point, it's safe to say that QUASI-DISK is fully operational on your system. Don't be afraid to treat it like any other drive.

Use QUASI-DISK to increase the speed of assemblies, word processing or any other disk-intensive application but remember, always transfer completed work to a floppy, or hard, disk for archival storage. Turning the system off will erase the contents of QUASI-DISK unless you have installed the battery back-up option as described in section 7.

If you should hit RESET on your system, don't worry about the data on QUASI-DISK. Simply execute QGO*.COM again. This time you will notice that the drive is not reformatted, but that the message "Drive C Active" will appear as before. QGO*.COM first checks to see whether or not the drive is correctly formatted before attempting a load of the BIOS extension. If the disk appears correct, the formatting step is skipped and the QUASI-DISK drivers are simply re-loaded. Executing DIR at this time will reveal all the files that were present before the RESET was hit.

Should you wish to change the QUASI-DISK setup at some time, re-execute QMAKE*.COM and change the parameters of interest. By creating and executing a new QGO*.COM file, QUASI-DISK will appear under the new format. Changing the capacity of QUASI-DISK or altering the spooler status is not recommended once data has been transfered to the drive. Doing so could result in loss of data. Save all your files first and then perform desired alterations.

Section 3, the next section of the manual, contains instructions for the use of the utilities supplied with QUASI-DISK. This might be a good time to review the function of these utiluties and to give them a try.

To avoid the "buffer full" condition, break the job into chunks that will fit into the spool buffer and print the chunks separately.

Printer dispatching is done through the CONST routine only. This allows your system to run at full speed, but the printer will be idle whenever the system is not in the keyboard status loop (as is the case during disk transfers).

Example of Spool Buffer Operation

```
     (1) -------  Inptr Outptr       (2) -------  Outptr
        |       |                       |       |
        |       |                       | job 1 |
        |       |                       |       |
        |       |                       |-------| Inptr
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
        |       |                       |       |
         -------                         -------

Initial condition                 One job sent to be printed
All of the buffer is free         Printer has just started
```

```
(3) -------
   |       |
   |(job 1)|
   |       |
   |-------| Inptr Outptr
   |       |
   |       |
   |       |
   |       |
   |       |
   |       |
   |       |
   |       |
   |       |
   |       |
   |       |
    -------
```

Printer has just finished.
Job 1 is still in the buffer,
but its area is now free
because it has printed.

```
(4) -------
   |       |
   |(job 1)|
   |       |
   |-------|
   |       |
   |       | Outptr
   | job 2 |
   |       |
   |-------|
   | job 3 |
   |-------| Inptr
   |       |
   |       |
   |       |
   |       |
    -------
```

Two more jobs sent to the printer
Printer is part way through job 2.
As the printer advances, the area
above Outptr becomes free space.

```
(5) -------
   | job 4 |
   | last  |
   | part  |
   |       |
   |-------| Inptr
   |       |
   | job 2 |
   |       | Outptr
   |       |
   |-------|
   | job 3 |
   |-------|
   |       |
   | First |
   | part  |
   |  of   |
   | job 4 |
   |       |
    -------
```

Job 4 sent to be printed.
Note that it overlaid the
area previously occupied by
job 1. Free space is from
Inptr to Outptr.

```
(6) -------
   | job 4 |
   | last  |
   | part  |
   |       |
   |-------|
   |       |
   | job 5 |
   |~~~~~~~| Inptr Outptr
   | job 2 |
   |-------|
   | job 3 |
   |-------|
   |       |
   | First |
   | part  |
   |  of   |
   | job 4 |
   |       |
    -------
```

Job 5 sent to be printed.
This job is larger than the
free space. The system will
now be busy (it will not let
you run anything else).
When the printer has printed
enough data to make space for
the remainder of job 5, the
system will free up again.

Print Spooler Utility Program "QS*.COM" (Query Spool)

The print spooler may be used without this utility, but QS*.COM makes the spooler more useful by enhancing certain operations and providing additional control over the spooling/printing process.

Typically, it is used to:

- Display number of k bytes queued to the spooler.
- Temporarily, suspend printer output.
- Display data about to be printed.
- Permits spool pointer "set back" to re-print the lost page(s) after printer errors (such as paper jams).

QS*.COM Commands

QS* will run in one of two modes:

- QS* followed by a space and a command letter. QS* will execute the command and then pass control to your system.
- QS* only, the program will prompt for a command.

QS*.COM Single Command Mode

QS* S	Give status. If print output is suspended, the response 'printer in hold' will be made.
QS* H	Hold output to printer
QS* R	Release output to printer
QS* D	Display the first 23 lines of data to be printed
QS* K	Kill (erase) spool buffer. Will ask for confirmation
QS* ?	Display Command list

Responses to a QS* Prompt (==>)

? Display command list

S Give status of spool buffer

D Display 23 lines of data from the 'current display ptr'

K Kill (erase) spool buffer. Will request confirmation

Commands that move the display pointer

L Move the 'current display pointer' by lines

Examples:

L	move forward 1 line
-L	move backward 1 line
-60L	move backward 60 lines

P Move the 'current display pointer' by pages. You can use this command if your print data uses form- feed (hex 0C) characters to control paging. This command works just like the 'L' command

F Find forward
Will prompt for the search argument (16 chars max).

-F Find backward
Same as find forward

Commands to set spool buffer pointers

^B Set beginning of print area. Sets temporary 'Outptr' equal to the display pointer

^E Set end of print area. Sets temporary 'Inptr' equal to the display pointer.

^U Update spool buffer pointers. Copies the temporary pointers to Qcode.

^C Quit

Space bar Repeats the previous command

Modifying Spool Buffer Pointers

By modifying the two spool buffer pointers, you can control which part of the buffer will be printed. The ^B (control-B) sets Outptr which determines where printing will begin. The ^E (control-E) sets Inptr which determines where printing will end.

For example, to repeat a page, back up Outptr by one page. To do this, move the display window with the L P or F commands until the first line to be re-printed appears at the top of the display, then, press ^B.

^B and ^E set temporary pointers in QS*. To actually set the pointers in Qcode, do a ^U (update). Now the change is final.

Since the spool buffer is quite large, it is possible that after a series of commands you may be confused as to where you are. If you have not yet given ^U, exit QS* with a ^C. Then start QS* again and you will be at the starting point again.

The spool buffer may contain several files, some of which may have been already printed. To tell you where you are within the buffer, a number appears to the left of the prompt line. A negative number indicates that you are in the section that has already printed. A number without a prefix shows that you are in the data to be printed, a number with a plus-sign, that you are past the area that will be printed.

Examples:

-367 ==> You have backed 367 bytes past the current print position

0 ==> You are at the current print position

955 ==> You are 955 bytes into the data to be printed

+123 ==> You are 123 bytes past the last byte to be printed

Example: - How to Re-print One Page

Assume that the printer has run out of paper partway through a page and you want to reprint this page.

QS* The printer will automatically go into a temporary hold state while QS* is active. The display will show 23 lines, the top line showing the line to be printed next.

-20L Go back 20 lines.

This is not back far enough....,

-L Goes back one more line. Keep pressing space until the first line to be printed is the top line on the display.

^B Set 'Begin here'

^U Update the spool buffer pointers in Qcode

Fix the paper at the printer

^C Exit QS*. The printer will start now.

If you wish to reprint the whole file, it would probably be too tedious to skip back by doing multiple -L operations. Try to remember some unique word near the beginning of the file and use the -F command (Find backward).

Example: - How to Print a Section of a File

QS* Enter QS*

F Find a word on the first line of the section

-L Go to the beginning of the line

^B Mark 'Begin print'

F Find a word on the last line of the section

L Go to the beginning of the next line

^E Mark 'End print'

^U Update spool buffer pointers

^C Exit QS*

Using the Spool as a Trace Buffer

QUASI-DISK spool buffer routines do not, in any way,affect data sent to the buffer. Binary data may be collected in the buffer via the LIST vector. By directing output from a program to LIST, the output may be traced.

Example:

> You are trying to get a new editor running, and it makes a mess of your terminal. Now you must determine what control codes the editor is sending to the terminal. Here is how to record and display the data:

QS* H	Hold the printer
QS* K	Clear the spool buffer (not needed if nothing in it)
^P	Let BDOS copy console output data to LIST device

>> At this point, invoke the editor and try a few things <<

^P	Stop copy to LIST. Not needed if your system has crashed. See 3.4.9
QCOPY* TRACE.1	Copy the buffer to TRACE.1. Now you can examine the trace with DUMP, DDT or whatever.
QS* K	Get rid of the trace, we don't want the printer to print it.
QS* R	Release the printer

System Crash With Data in the Spool Buffer

As long as power has not dropped, the data will still be in the buffer. Since a cold boot resets the spool buffer pointers, use QS* to set the pointers to include the trace data in the buffer before you run QCOPY*.

Using the F (Find) Command

When doing a backward find, the word found will appear at the top left of the screen. To see the whole line do a -L. When doing a forward find, the word found will not be visible. You must do a -L to see it. If the search was unsuccessful, the display will not change and the message 'Not found' will be displayed. The find command uses the BDOS editor so that you can correct any typing mistakes. CP/M line editor control characters, LF, CR, cannot be used as a search argument. To search for any other control chars (such as ^L form-feed), enter the control character and then press return. Pressing space after a find will repeat the operation.

The QS* Display Routine

The display routine will output a decimal for all non-printing codes except for OC and FF. OC (form-feed) will display an up-arrow and FF stops the display. The spool buffer is filled with hex FF when the system is powered up and is used to prevent the L and P commands from accessing the unused part of the spool buffer. Access to this area would result in attempting to display it on the system console which could result in random hex (screen control) characters being sent to the console.

You may find the the spool buffer pointer control procedures somewhat tedious. If so, use QCOPY* to copy the spool buffer to a file of your choice. You can then use your favourite editor to edit and spool the print data.

Procedure:

QS* H Hold the printer....

QCOPY* YOUR.FIL Will copy the active part of the spool buffer to the file YOUR.FIL. All of the print data is now saved.

QS* K Kill all data in the spool buffer

QS* R Release the printer

Now, edit the newly created file to meet your needs and spool it to the printer.

QUASI-DISK Utility and Diagnostic Package, "QDU*.COM"

QDU* will perform all of CP/M's CCP commands except the loading of a program. Be careful. Some QDU* commands use a different syntax. It is presently fixed for a QUASI-DISK port number of 0. An explanation of the commands follows:

DIR

standard directory display

ERA fn.ft

EX: ERA QDU*.* — erase file(s) on current disk. Will display the files to be erased. Will give a warning when question marks in file spec.

REN old.fil new.fil

EX: REN QDU*.COM QDME.COM — Rename 'old.fil' to 'new.fil'

GET adrs fn.ft

EX: GET 100 QDU*.COM — Loads 'fn.ft' to 'adrs' in hex. Displays ending address.

SAVE from to fn.ft

EX: SAVE 100 DFF QDU*.COM — Saves memory image from hex location 'from' to location 'to' with a name of 'fn.ft'. Will give a warning if the file exists.

`COPY source.fil dest.fil`

Copies 'source.fil' with a new name of 'dest.fil'. The syntax is similar to PIP, except for the destination being last and no equal sign. Does not verify.

Examples:

`COPY A:QDU*.DOC B:` — Copy QDU*.DOC from A to B

`COPY *.DOC E:` — Copy all .DOC files from current disk to E.

`COPY E:QDU*.DOC` — Copy QDU*.DOC from E to current disk

`TYPE fn.ft`

`EX: TYPE QDU*.DOC` — Display 'fn.ft' at the console. Will stop at end of every page. Press space to continue.

`LIST fn.ft`

`EX: LIST QDU*.DOC` — Print 'fn.ft'

`DUMP adrs`

`EX: DUMP 0` — Dump memory to the console from 'adrs'. Press space for the next line of dump. Press return to end the dump.

`GOTO adrs`

`EX: GOTO E000` — Call 'adrs'

`IN port`

`EX: IN 40` — Input from 'port' and display at the console.

`OUT port data`

`EX: OUT C0 41` — Output 'data' to 'port'

QERA

Writes 256k bytes of E5 to QUASI-DISK with verify

QFMT

Formats QUASI-DISK for 76 tracks of 26 records. Every record is 128 bytes of E5, followed by a 1-byte CRC (check sum).

QDMP hi lolo

EX: QDMP 1 2FF0

Dump QUASI-DISK from address 'hi lolo'. Press space for next line, or return to end.

QS*T hi lolo

EX: QS*T 1 2FF0

Store bytes on QUASI-DISK from address 'hi lolo'. Press space to advance to next byte without update. Press return to end.

CS fn.ft

EX: CS PRINT.SAV

Copy spool to a file. To use, disable the printer so that the data remains in the spool, then send the data to be printed. After that, copy the spool to a file using the CS command. This command is handy for editing print data.

QUASI-DISK Utility "QCOPY*.COM"

Used when data has been spooled for printing, but must be edited before the printing is to take place. The sample routines presented in Section 2.4.2. were generated using this utility as follows:

Procedure:

QS* H - Hold the printer and disable output.

^P - CP/M command to cause all console data to be sent to the printer as well.

- At this point, the sample commands were executed and the results displayed at the console and sent to the print spooler simultaneously.

QCOPY* YOUR.FIL- All of the data saved on the spooler is copied to a file called "YOUR.FIL".

QS* K - Kill the data in the spool buffer.

QS* R - Release the printer.

TECHNICAL HARDWARE DETAILS

This section of the manual is provided for those users wishing to modify or incorporate the QUASI-DISK software into their BIOS routines. The main goal is to acquaint the reader with the system interface and hopefully enable him to generate his own code if required. No attempt is made to explain how the board works on an IC level. If this information is required please refer to the service manual available from Electralogics Incorporated.

Accessing the QUASI-DISK Registers

Access to QUASI-DISK is through 6 onboard registers as defined below:

```
           ------------------------------------
Base +0    ¦                                  ¦  Data Register
           ------------------------------------

           ------------------------------------
Base +1    ¦                                  ¦  Command/Status
           ------------------------------------  Register

           ------------------------------------  Powerfail
Base +2    ¦                                  ¦  Acknowledge
           ------------------------------------  Register

           ------------------------------------
Base +4    ¦                                  ¦  Address 0-7
           ------------------------------------  Register

           ------------------------------------
Base +5    ¦                                  ¦  Address 8-15
           ------------------------------------  Register

           ------------------------------------
Base +6    ¦                                  ¦  Address 16-21
           ------------------------------------  Register
```

General Register Overview

Note: "Base" refers to the location of the board in the I/O address space.

QUASI-DISK occupies 8 contiguous I/O addresses, and may be located on any 8 byte boundary within the 256 byte I/O address space. If extended I/O addressing is enabled, QUASI-DISK may be located on any 8 byte boundary within the 64K I/O address space.

Following is more specific QUASI-DISK register information.

Register 0 - Data Register

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Register 0 Bit Assignments

This read/write register is used to send/receive data to/from QUASI-DISK.

Accessing QUASI-DISK RAM is a two step process. When the host system writes to the QUASI-DISK data register, the data is stored in a temporary holding register. Control logic then transfers the data byte to the dynamic RAM as soon as possible. Maximum delay from the time that data is written to QUASI-DISK by the host to the time that the data is stored in RAM is 500 nsec. Additional logic prevents accidental overwriting of data in the holding register. The maximum transfer rate supported is 2 Megabytes/sec. (ie. 500 nsec/byte.)

The reading process is similar. When the host system reads the data register, it actually draws the data byte from a holding register. Once QUASI-DISK notices that the read holding register is empty, a read cycle is begun to replace the data. If the host attempts to read an empty holding register, wait states are generated until the data byte is available (again, a 500 nsec. delay max.).

No wait states will be inserted until a 1 megabyte/sec transfer rate is exceeded, and even then only on the first byte transfer. Once the host and QUASI-DISK are synchronized, rates up to 2 megabytes/sec will not generate wait states. A 6 mHz. Z80 can only transfer data at 272k bytes/sec. No wait states should occur until DMA is used. The holding register concept allows QUASI-DISK to work with any S-100 system, regardless of the processor used. Since the host system does not directly access QUASI-DISK RAM, timing problems associated with such interfaces are eliminated. Furthermore, since all accessible registers are of the LSTTL type, access time is in the 50 nsec. range, faster than most, if not all, microprocessors.

```
Register 1 - Command/Status Register    0      1      0      0
                                         4      2      1
  7                                                          0
-----------------------------------------------------------------
| C3  | C2  | C1  | C0  |  PF  |  WP  |  ENA  |  DIR  |
-----------------------------------------------------------------
  ^                 ^      ^      ^       ^        ^
  |                 |      |      |       |        |
  +--------+--------+      |      |       |        |
           |               |      |       |        |
System     |               |      |       |        |
Capacity --+               |      |       |        |
                           |      |       |        |
Power Fail Status----------+      |       |        |
                                  |       |        |
0 = power failed since last acknowledge   |        |
1 = okay                          |       |        |
                                  |       |        |
Write Protect --------------------+       |        |
                                          |        |
0 = protected                             |        |
1 = unprotected                           |        |
                                          |        |
Enable -----------------------------------+        |
                                                   |
0 = auto inc/dec enabled                           |
1 = auto inc/dec disabled                          |
                                                   |
Direction -----------------------------------------+

0 = auto increment select
1 = auto decrement select                  00      100=4
                                                   101=5
                                                   110=6
```

Register 1 Bit Assignments

Bit 0 - (DIR) - A read/write bit which defines the direction in which the 22 bit address will be stepped after the next access to the data register, provided that the enable bit is set to "0". After a reset or powerfail this bit is reset to "0".

Bit 1 - (ENA) - A read/write bit which controls the auto step feature. If ENA = 0 after the next access to the data register, the 22 bit address on QUASI-DISK will be stepped in the direction specified by DIR. If ENA = 1, the address will remain unchanged. After a reset or power failure, this bit is reset to "0".

protected

Bit 2 - (WP) - A read/write bit used to modify and monitor the write protect feature on QUASI-DISK. Setting this bit to 1 or 0 results in protecting/unprotecting QUASI-DISK RAM. A power failure or reset will clear this bit to "0" and protect the memory contents. When a power failure causes the WP bit to be cleared, the power failure must be acknowledged before the WP bit can be set.

Bit 3 - Power Fail Status - This read only bit reflects the output of the power fail detect logic. When PF = 1, a power failure has been detected since the last power fail acknowledge. After a system reset, an acknowledge should be performed by writing to register 2 to ensure that this bit is reset. After any disk read or write, the status of this bit should be examined.

Bits 4-7 These read only bits are used to define the number of 256k banks which are installed on QUASI-DISK. Reading these 4 bits reflects the setting on SW1-1 to SW1-3. The following table defines the relationship between these switch settings and capacity.

C3 SW1-1	C2 SW1-2	C1 SW1-3	C0 SW1-4	# Of 64k Banks	# of 256k Banks	Total Capacity
0	0	0	0	8	0	512k
0	0	0	1	7	1	704k
0	0	1	0	6	2	896k
0	0	1	1	5	3	1088k
0	1	0	0	4	4	1280k
0	1	0	1	3	5	1472k
0	1	1	0	2	6	1664k
0	1	1	1	1	7	1856k
1	0	0	0	0	8	2048k
				WITH EXPANSION BOARD		
1	0	0	1	7	9	2752k
1	0	1	0	6	10	2944k
1	0	1	1	5	11	3136k
1	1	0	0	4	12	3328k
1	1	0	1	3	13	3520k
1	1	1	0	2	14	3712k
1	1	1	1	0	16	4096k

Capacity Switch Settings

Register 2 - Power Failure Acknowledge Register

This write only register is used to reset the powerfail logic once a failure has been detected. The data written to the register is immaterial. If an acknowledge does not reset the PF status bit, the voltage on the S-100 +8 Volt rail is still below bus specification and should be checked. The red LED on the QUASI-DISK board will be on when a power fail has been detected, and will be turned off if the acknowledge command is successfull. If this LED cannot be turned off after repeated acknowledge commands then the board's I/O base address should be checked and the +8 Volt supply checked.

Register 4 - Address 0-7 Register

```
    7                                                         0
-----------------------------------------------------------------
¦  A7   ¦       ¦       ¦       ¦       ¦       ¦       ¦  A0   ¦
-----------------------------------------------------------------
```

Register 4 Bit Assignments

This read/write register is used to examine and set the lower 8 bits of the QUASI-DISK address.

Register 5 - Address 8-15 Register

```
    7                                                         0
-----------------------------------------------------------------
¦  A15  ¦       ¦       ¦       ¦       ¦       ¦       ¦  A8   ¦
-----------------------------------------------------------------
```

Register 5 Bit Assignments

This read/write register is used to examine and set the middle 8 bits of QUASI-DISK address.

Register 6 - Address 16-21 Register

7							0
0	0	A21	A20	A19	A18	A17	A16

Register 6 Bit Assignments

This read/write register is used to examine and set the 6 highest bits of QUASI-DISK address.

Note: Registers 3 and 7 are not implemented - reading these registers will return undefined data.

TECHNICAL SOFTWARE DETAILS

This section of the manual provides interested users with some detail of the installation and diagnostic package.

All programs are written in 8080 code and will run under CP/M version 2.0 or higher. Tests have not yet been made for QUASI-DISK under CP/M 3.0.

An assembler is not required to customize QUASI-DISK parameters.

Source code is provided in the appendices. It may be altered and re-assembled using CP/M's ASM or other compatible assembler.

A Description of the QGO* Program

The code for QGO*.COM is located in the first section of QMAKE*.COM. When QMAKE* is executed, QGO* may be customized by selecting the appropriate functions. QGO*.COM is then created by selecting the 'Create QGO*' function.

The source code for QMAKE* is provided. Those who feel ambitious (and are capable of doing so) can modify the source and then re-assemble QMAKE* using ASM or another compatible assembler. Please, be very, very careful. This code contains many instructions that alter pointers and relocate code and it is very easy to get into trouble.

QGO* first tests to see if if QUASI-DISK can be accessed. If not, the message 'Cannot access QUASI-DISK' is displayed and QGO* aborts. If the above test is successful, another test is performed to see if the Qcode (the part that interacts with your BIOS) is already installed. It does this by examining the READ vector in your BIOS. If the vector points into QCODE, it is assumed to be already installed. In that event, QGO* resets BDOS and exits after the 'Disk x active' message.

If the code is not installed, QGO* does the following:

* installs your port definition into I/O instructions

* sets up the disk parameter header and disk parameter block for QUASI-DISK

* moves the data section into position

* moves the code section into position while adding the required offset

* modifies the following vectors in your BIOS to point into Qcode:

```
WARMBOOT only if the Qcode is below the CCP
CONST    only if the spooler is present
CONIN    only if the spooler is present
LIST     only if the spooler is present
HOME,  SELDSK, SETTRK, SETSEC, SETDMA, READ, WRITE
and SECTRAN
```

After the above modifications have taken place, all disk calls must pass through Qcode. If QUASI-DISK is selected, Qcode executes the call on QUASI-DISK. Otherwise, control is passed to your BIOS. The warmboot vector is needed by Qcode to restore the BDOS vector at 0006H after DDT and some other programs terminate. These programs alter the data at 0006H and 0007H and then leave it that way.

* displays code starting and ending addresses at the console as well as disk capacity.

* tests to see whether or not the QUASI-DISK directory and spooler need formatting. I.D. bytes located at the top of the QUASI-DISK RAM are checked and compared with those of the current revision loader. If the I.D. bytes match, it is assumed that the QUASI-DISK contains valid data and the formatting process is skipped. Otherwise, the directory and conditionally the spooler are formatted and the message 'formatting . .' is displayed at the console.

* resets the BDOS and exits with the 'Disk x active' message.

Notes Concerning Disk Format

The Disk Parameter Header

Do not change the routines that set up the DPH unless you have a very good reason to do so.

XLT Should point to a sector translate table or be 0000 if no translate table is used.

DIRBUFQGO* Copies this from your BIOS.

DPB QGO* sets this to point at the DPB in Qcode

CSV Not used because CKS in the DPB is set to 0

ALV QGO* sets this to point at ALV in Qcode (points at the end of Qcode). The area between the end of the actual Qcode and your CCP is used for ALV, the allocation bit map. BDOS manipulates this area, so make sure that it is large enough (requires 1 byte for every 16k bytes of space on QUASI-DISK).

The Disk Parameter Block

SPT SECTORS PER TRACK

This value tells BDOS when to seek to the next track. You can make it anything from 1 to 255. A higher number will result in faster operation (fewer seeks).

BSH BLM EXM NUMBER OF SECTORS PER BLOCK

These three parameters are interrelated and must be specified as described in CP/M documentation. A block (some people call it group) is the number of sectors represented by one bit in the allocation bit map. Don't set BSH=3 BLM=7 EXM=0 and DSM >255 This will totally confuse BDOS.

DSM TOTAL NUMBER OF BLOCKS ON DISK

This parameter is used by BDOS to determine the end of your disk. Since each increase of BLM doubles the disk space that can be accessed, and your disk may not be double the size, the DSM value tells BDOS where the end is. If you want to reserve some space at the top of your disk, set

DSM lower than the actual disk size and BDOS will leave your space alone.

DRM AL0 AL1 NUMBER OF DIRECTORY ENTRIES

DRM can be any desired value. Just be sure to set AL0 and AL1 according to the CP/M manual.

CKS DIRECTORY CHECK SIZE

This is forced to 0 by the initializer.

OFF NUMBER OF RESERVED TRACKS BELOW DIRECTORY

We suggest 2. If you select any value less than 2, be sure never to let SYSGEN write onto QUASI-DISK because it will wipe out the directory and part of your first file.

Disabling Error Detection

CRC checking can be disabled by replacing the XRA C at label CRCSW in the QGO* module with XRA A. This is not recommended since any error in the QUASI-DISK data would go unnoticed and could propagate throughout the entire data base.

EXPANSION MODULE OPTION

QUASI-DISK is shipped with 512k of 64k RAMs (64k chips) allowing expansion to 2 Megabytes by plugging in 256k devices. However, until 256k devices become price competitive, a QUASI-DISK may be expanded to 1 Megabyte by plugging in an expansion option above the existing array of RAM. This expansion board doubles the number of chips which are supported to 128, resulting in 1 Megabyte if 64k devices are used, and 4 Megabytes when 256k devices become available.

To increase the board's capacity, simply order the expansion module and plug it onto QUASI-DISK. (Pins are already provided on QUASI-DISK to accept this option.) The installation routine will configure itself for the new RAM. If 256K RAMs are to be added, set SW1 to reflect the increased capacity.

BATTERY BACKUP OPTION

The battery backup option consists of a sealed gel cell and controller board in one module. Each board has a 10 pin option connector located on the upper left corner of the board. A 10 wire cable assembly runs from the QUASI-DISK board, terminating at the battery module.

Four LEDs are located on the battery module. When the module is plugged into QUASI-DISK and/or the power supply is plugged into the module, the yellow LED should be "ON" indicating that the batteries are charging. During power failures, this LED will be "OFF". As long as the host system's power is active, or the power supply is active, the green LED will not be "ON". When QUASI-DISK requires battery energy to remain operational, the green LED will turn "ON". The red LED labelled "SYSTEM FAILURE" is provided to indicate that a power failure lasted longer than 2 hours, and the board had to be shut down to keep from damaging the battery module. This LED will only turn "ON" when power is restored. Statistically, it should never turn on, however, if it does turn "ON", the error reset switch must be used to turn it "OFF".

The red LED labelled "FUSE BLOWN" is provided as an indication to problems inside the module. Consult your manual, or the factory if this LED is "ON".

When the QUASI-DISK board determines that power is below specification, a signal is sent to the battery module. This switches on backup power. Logic within the module will keep QUASI-DISK powered up until one of two things occurs:

1) System power returns to normal.

2) Battery is discharged to critical level.

In either case, backup power is disconnected and in case (1) the battery begins to charge. In case (2), the LED labelled "SYSTEM FAILURE" will turn "ON".

Battery charging is controlled and is a function of the discharged level of the cell. The +16V rail is used as an energy source if and only if the power supply is not used.

S-100 BUS LINE USAGE

Pin	Function	Comments
1	+8 Volts	
2	+16 Volts	Used if battery option is installed.
3	XRDY	Jumper W1 used to select between XRDY and RDY.
20	GND	May be cut for systems using pin 20.
29	A5	
30	A4	
31	A3	
32	A15	
33	A12	
34	A9	
35	DO1	
36	DO0	
37	A10	
38	DO4	
39	DO5	
40	DO6	
41	DI2	
42	DI3	
43	DI7	
45	sOUT	
46	sINP	
50	GND	
51	+8 Volts	
53	GND	May be cut for systems using pin 53.
54	SLAVE CLR*	
70	GND	May be cut for systems using pin 70.
72	RDY	See pin 3.
75	RESET*	
77	pWR*	
78	pDBIN	
79	A0	
80	A1	
81	A2	
82	A6	
83	A7	
84	A8	
85	A13	
86	A14	
87	A11	
88	DO2	
89	DO3	
90	DO7	
91	DI4	
92	DI5	
93	DI6	
94	DI1	
95	DI0	
100	GND	

Sample Bios Routines

Also included on this disk are sample BIOS routines that may prove useful in installing the Quasi-Disk code directly into your system. Please see the file called SAMPLE.DOC for more details. We hope you find the enclosed information useful and wish you much success in its implementation.

Regards,

Rick Harnell

Rick Harnell

3. The CCP does a RESET call to BDOS.

4. BDOS calls the BIOS HOME function. This call is intercepted by the QCODE, wich then replaces the BDOS vector at 0006h to point to itself.

Note:

Steps 1 to 4 are needed because a warm boot causes the BIOS to overlay the BDOS vector at 0006h. We must make this vector point to QCODE to indiate the reduced size of the TPA to application programs.

The CCP must keep location 0004h updated with the current disk and user number and the BIOS warm boot routine must pass this byte to the CCP. IF this is not done, Quasi-Disk will not be re-selected after a warm boot.

A New Diagnostic Routine

We have finally completed the first stage of a diagnostic routine for Quasi-Disk called DIAG.ASM. This routine allows any bank (or banks) of memory to be tested and the results displayed in a convenient form. A short mention as to nomenclature may prove useful here.

Two terms are used to refer to a "chunk" of ram on Quasi-Disk. The term (bank) refers to 8 memory devices, either 64K chips or 256K chips. Therefore, it becomes obvious that a Quasi-Disk contains 8 banks and each bank may be populated with either 64K X 1 or 256K X 1 rams. (The same applies to the expansion module.)

The term (block) is used to describe 64K bytes of storage. In the case of a bank populated with 64K rams, bank and block are interchangeable. However, a 256K bank consists of 4 64K blocks. You will find this information useful when using the diagnostics routine.

The diagnostics routine has been designed to test the Quasi-Disk and display the results in two ways. When the test is actually executed, any errors located will be displayed on the console with all sorts of pertinent data. This will help to locate individual errors or bugs relating to blown chips. Statistics are collected relative to these errors and saved in special counters. Each time a pass is made through all the tests and no errors are detected, the pass counter is incremented. (maximum value=9999) However, if an error is detected a counter is incremented indicating that chip XX had one or more errors this pass. Note that the counter can only be incremented once per pass. (maximum value=99 errored passes) This data may then be displayed with the "Display Error Bit Map" command and a map of the Quasi-Disk is sent to the console showing the actual problem chips. Such a test routine will make finding the bad chip much easier.

Now, What's In This New Package?

Verison 2.3 is a completely new software system for your Quasi-Disk. The major difference between 1.1 and 2.3 is in the spooler function. In version 1.1, the spooler was fixed at 0K or 64K as set by you using QMAKE. Version 2.3 has a dynamic print spooler which occupies no disk space, (actually only 2K) when there is no data to be printed. As data is sent to the list device, the amount of disk space allocated to the spooler grows. But don't worry, it won't overwrite your files. The spooler will continue to grow until there is only 14-16K left on the Quasi-Disk and will then stop. This leaves room for other disk activity. When the printer has completed its printing, the allocated disk space is returned to the disk pool.

The other utilities are basically the same in function. A modification has been made to the way the utilities access the QCODE that is loaded to drive Quasi-Disk. A vector (default location 0015h) is used to point to the location where the Quasi-Disk drivers exist. This greatly simplifies the utilities while making the software easier to support. Here's a point form list of some other enhancements and features:

-The new disk parameter block allows 256K directory entries.

-The warmstart routine passes control to the BIOS warm boot.

-The home routine resets the BDOS vector at 0005 if preceded by a warm boot.

-The user can control the directory formatting as follows:

Example: QG023 f

will cause the Quasi-Disk to be formatted before installation into CP/M. The "f" forces a formatting operation and will prove useful when and if a directory is destroyed but the disk I.D. is still intact.

Any other invocation of QG023 examines the disk I.D. string to see if the Quasi-Disk contains valid data. If so, the Quasi-Disk is installed and not formatted. If the software detects a bad I.D. string, a prompt will be issued before formatting takes place to give you the option of trying to logon as is.

Handling of Control-C and Warm Boot

1. Jump to 0 causes QCODE to set the warm-start flag and to pass control to the BIOS warm boot routine.

2. The BIOS does it's thing and then passes control to the CCP.

NEW QUASI-DISK SOFTWARE RELEASE VERSION 2.3

Dear Quasi-Disk Customer:

Enclosed you will find the latest release of the Quasi-Disk software. Several enhancements have been added to make Quasi-Disk a more functional and user friendly product.

We must also take this opportunity to mention that this release of the latest update will mark the last time we will be able to distribute such updates free of charge. The expanding base of customers means that the expense in terms of time and dollars is more than we can handle.

Instead, we are offering our customers two options. Please note that both these options are dependant upon having a warranty registered with us. (If you haven't done so, see the enclosed registration form.)

Option 1 consists of basically the same service we have now, in other words, the disk with a new software release will be sent to customers who have requested this service. (In order to qualify, you must indicate your wishes on the enclosed form.) The cost for option 1 is $10.00 per release and will cover our expenses relative to copying disks, mailing and administration. A notice will be sent by mail to all registered customers indicating the contents of the update. Those interested in that release should simply return the notice with a check or money order.

Option 2 is perhaps more interesting to anyone having a modem. Customers who register for option 2 will be granted an account on a bulletin board system. Upon receipt of the notification of a new release, the customer may call and retrieve the appropriate software package. Each time a customer signs on to transfer programs his/her Visa account will be billed 5 dollars. As much software as required may be transfered in one call with the only limitation being a 30 minute per day maximum for logged on time.

We sincerely hope you understand the problem we are experiencing, and hope you will find the aforementioned terms agreeable. Please fill out the enclosed form and list your comments, opinions etc. We really are interested in what you think. Remember, you must be registered under one of the above plans if you wish to receive updates.

MANUAL UPDATES

BATTERY BACK-UP OPTION

The battery backup option consists of a battery and controller board in one module. Each board has a 10 pin option connector located on the upper left corner of the board. A 10 wire cable assembly runs from the QUASI-DISK board, terminating at the battery module.

Four LEDs are located on the battery module and are used to indicate the condition of the backup system. When the module is plugged into QUASI-DISK and/or the power supply is plugged into the module, the yellow LED should be "ON" indicating that the batteries are charging. During power failures, this LED will be "OFF". The green LED labelled "Backup on" is used to indicate when the batteries are being discharged. As long as the host system's power is active, or the power supply is active, the green LED will not be "ON". When QUASI-DISK requires battery energy to remain operational, the green LED will turn "ON".

The red LED labelled "Failsafe" is provided to indicate that a power failure lasted longer than 2 hours, and the board had to be shut down to keep from damaging the battery module. This LED will only turn "ON" when power is restored and is reset automatically. Statistically, it should never turn on, however, if it does turn "ON", the data on Quasi-Disk will no longer be valid.

The red LED labelled "FUSE BLOWN" is provided as an indication to problems inside the module. Sample fuses are provided to allow replacement in such an event. Realize however that a blown fuse may mean more serious problems so proceed with caution and common sense.

When the QUASI-DISK board determines that power is below specification, a signal is sent to the battery module. This switches on backup power. Logic within the module will keep QUASI-DISK powered up until one of two things occurs:

1) System power returns to normal.

2) Battery is discharged to critical level.

In either case, backup power is disconnected and in case (1) the battery begins to charge. In case (2), the LED labelled "Failsafe" will flash once when power is restored.

Battery charging is controlled and is a function of the discharged level of the cell. The +16V power supply may be used as an energy source if and only if the wall mount power supply is not used. Logic inside the module determines where to draw the charging current from.

CONNECTING THE BATTERY MODULE TO THE QUASI-DISK

Extreme care must be exercised when connecting the battery module to the Quasi-Disk. Try to follow the steps as outlined below.

1) Plug the wall mount transformer into the battery module.

2) Plug the flat cable supplied into the Quasi-Disk external power connector located in the upper left hand corner of the board.

3) Plug the transformer into a 115V standard unswitched wall outlet. Remember that the transformer must always be on even if the system is off. The yellow LED should turn on indicating that the battery is charging.

4) Turn your system on. Plug the remaining end of the flat cable into the battery module.

**** Note: - The Quasi-Disk should be connected or disconnected from the battery module only when the system power is "ON"! This ensures that the Quasi-Disk will not draw any power from the module while it is being connected. Please be careful.

To test the installation, bring up your Quasi-Disk as usual and transfer several files over. Turning off the system power will not change the status of the LED's since the transformer is still charging the batteries and now powering the Quasi-Disk.

Unplugging the transformer should result in the yellow LED going out and the green "Backup On" LED turning on. The internal batteries are now supplying the power to Quasi-Disk.

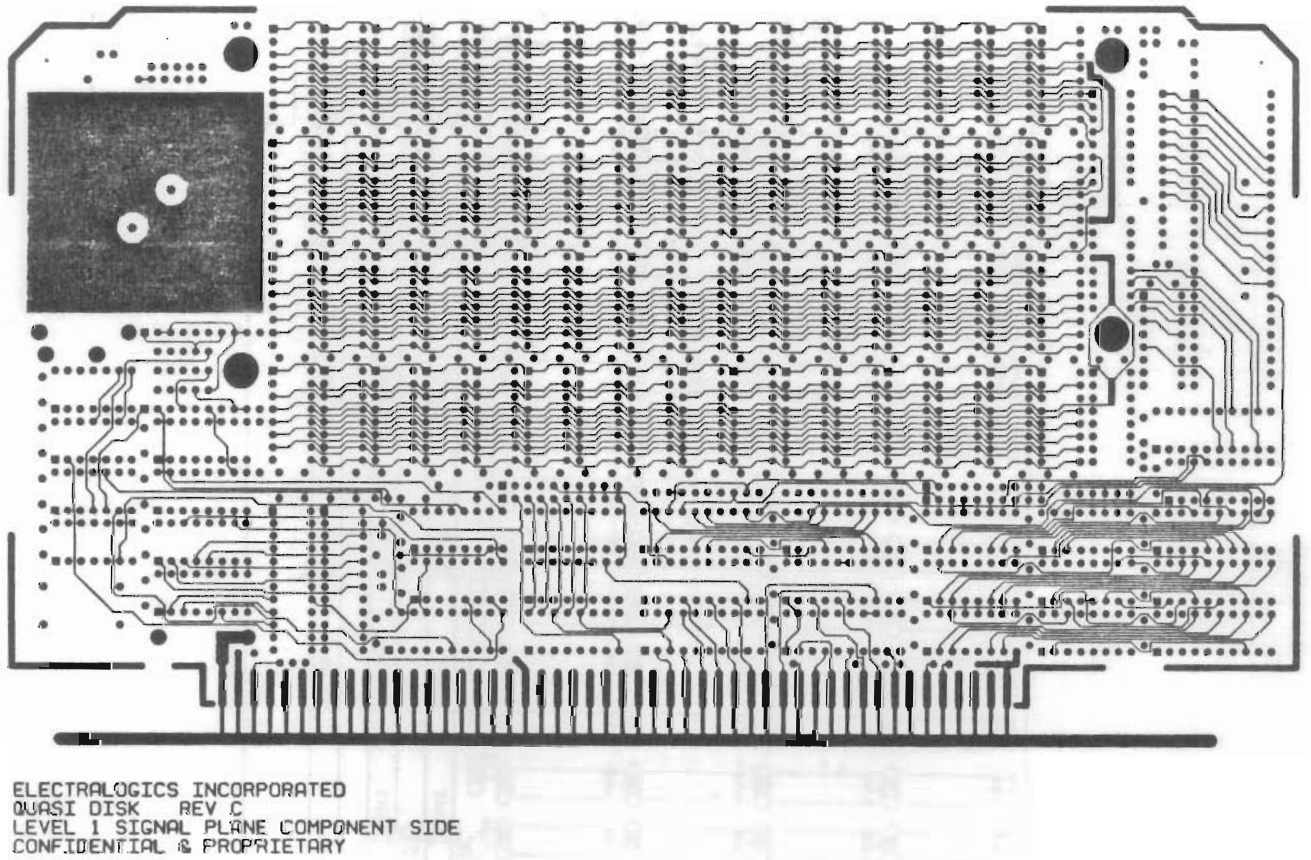
Plug the transformer back in and turn on the system power. The yellow LED will be on and all others will be off. Again, bring up Quasi-Disk and you will notice that the system did not format. Examining the directory should reveal that all files are still intact.

PART #	DESCRIPTION	QTY/UNIT	IDENTIFICATION
161007	IC; TTL 7407	1	U4
167012	IC; MOS MK4564	64	U35-U98
168100	IC; PROM 82S137	1	U31
168200	IC; BIPOLAR DP84244	1	U30
168201	IC; BIPOLAR DP8409	1	U29
171001	OSC; 16 MEGAHERTZ	1	U5
180000	SWITCH; 8 POSN DIP	3	SW1-SW3
190000	SOCKET; IC 8 PIN	2	
190001	SOCKET; IC 14 PIN	4	
190002	SOCKET; IC 16 PIN	80	
190003	SOCKET; IC 18 PIN	1	
190004	SOCKET; IC 20 PIN	11	
190006	SOCKET; IC 24 PIN	2	
191005	PLUG; HEADER 10 PIN	1	P5
200007	PCB; QUASI-DISK	1	
230000	SCREW; 4-40X5/16 PAN HEAD	8	
230001	SCREW; 4-40X1/2 PAN HEAD	2	
235000	STANDOFF; 1/2X1/4-4/40 CLINCH	4	
239001	MISC; CARD EDJECTOR	2	
240002	HEATSINK; #6351B	1	
700002	MANUAL; QUASI-DISK	1	
890003	SUB; BUS INTERFACE CONTROLLER	1	U11
890004	SUB; RAM CYCLE CONTROLLER	1	U10
892006	SUB; POST 10 PIN X 1	2	P1,P2
892007	SUB; POST 13 PIN X 1	1	P4
892008	SUB; POST 6 PIN X 1	1	P3

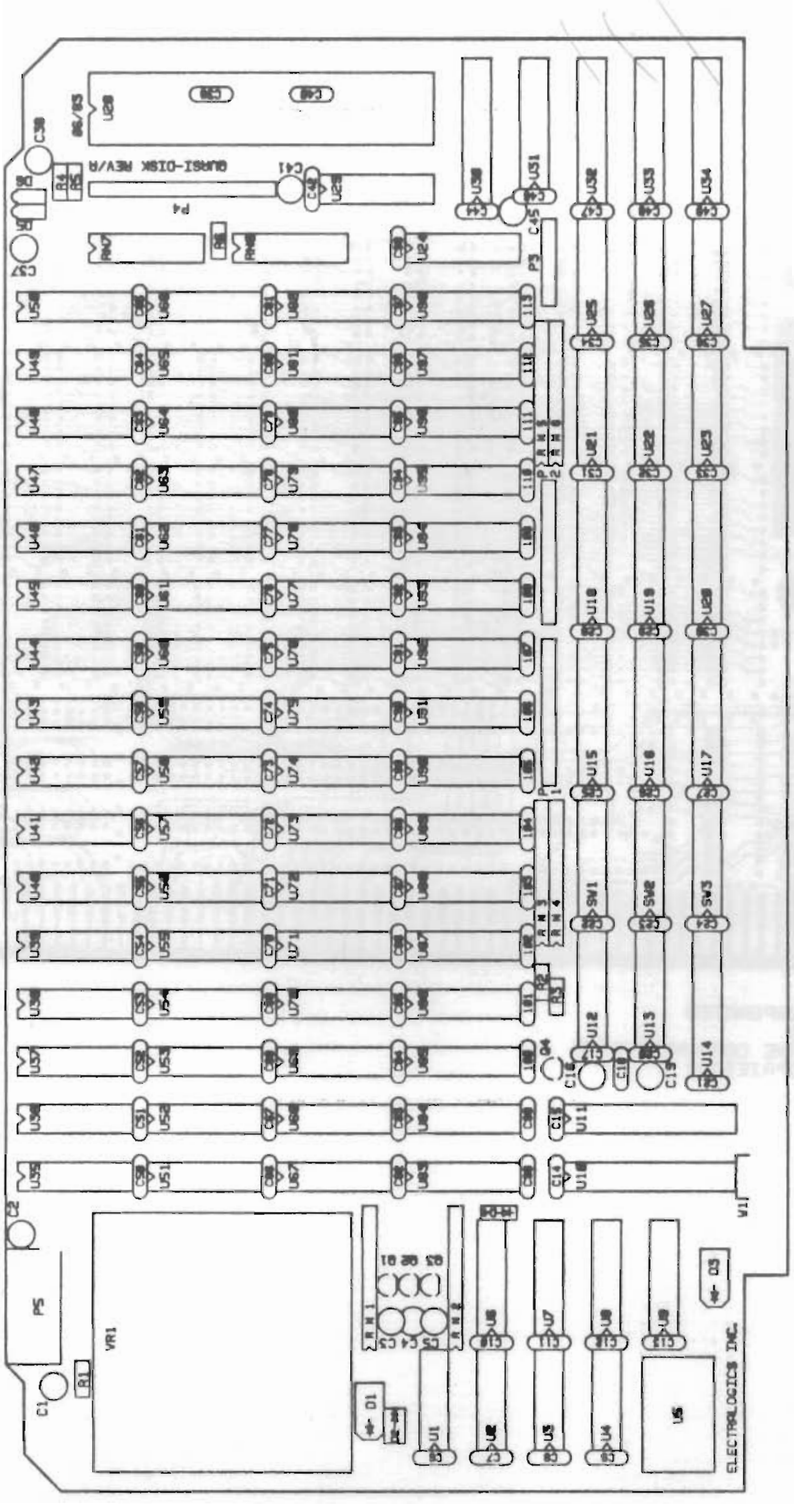
BILL OF MATERIALS

PART #	DESCRIPTION	QTY/UNIT	IDENTIFICATION
100103	RESISTOR; 10k OHM 5% 1/4W	1	R2
100104	RESISTOR; 100k OHM 5% 1/4W	1	R3
100330	RESISTOR; 33 OHM 5% 1/4W	1	R6
100470	RESISTOR; 47 OHM 5% 1/4W	1	R1
100471	RESISTOR; 470 OHM 5% 1/4W	2	R4,R5
102002	RESISTOR NETWORK; 3.3k X 9	4	RN3-RN6
102003	RESISTOR NETWORK; 1k X 5	2	RN1,RN2
102005	RESISTOR NETWORK; 33 OHM X 8	2	RN7,RN8
110002	CAPACITOR; 100 PF/50V CER	1	C18
111000	CAPACITOR; 1 UF/35V TANT	4	C3-C5,C16
111001	CAPACITOR; 10 UF/10V TANT	7	C1,C2,C19,C31,C32 C35,C39
112000	CAPACITOR; 0.1 UF/50V MONO	94	C6-C15,C17,C20-C30 C33,C36-C38 C40-C43,C44-C107
112001	CAPACITOR; 1 UF/50V MONO	1	C34
131001	ZENER; 6.2V 400 MW	1	D4
132000	RECTIFIER; 1A 50PIV	1	D2
132001	RECTIFIER; 3A 50PIV	2	D1,D3
133000	TRANSISTOR; NPN SIGNAL	4	Q1-Q4
140002	LED; RED PCB MOUNTING	1	D5
140003	LED; YELLOW PCB MOUNTING	1	D6
150004	REGULATOR; 5V 3A TO3	1	VR1
160074	IC; LSTTL 74LS74	2	U2,U3
160107	IC; LSTTL 74LS107	1	U1
160123	IC; LSTTL 74LS123	1	U12
160138	IC; LSTTL 74LS138	2	U8,U24
160139	IC; LSTTL 74LS139	1	U29
160175	IC; LSTTL 74LS175	2	U6,U13
160191	IC; LSTTL 74LS191	7	U9,U21-U23,U25-U27
160241	IC; LSTTL 74LS241	2	U19,U20
160244	IC; LSTTL 74LS244	4	U14,U32-U34
160374	IC; LSTTL 74LS374	2	U15,U18
160688	IC; LSTTL 74LS688	2	U16,U17
162940	IC; HCMOS 74HC4040	1	U7

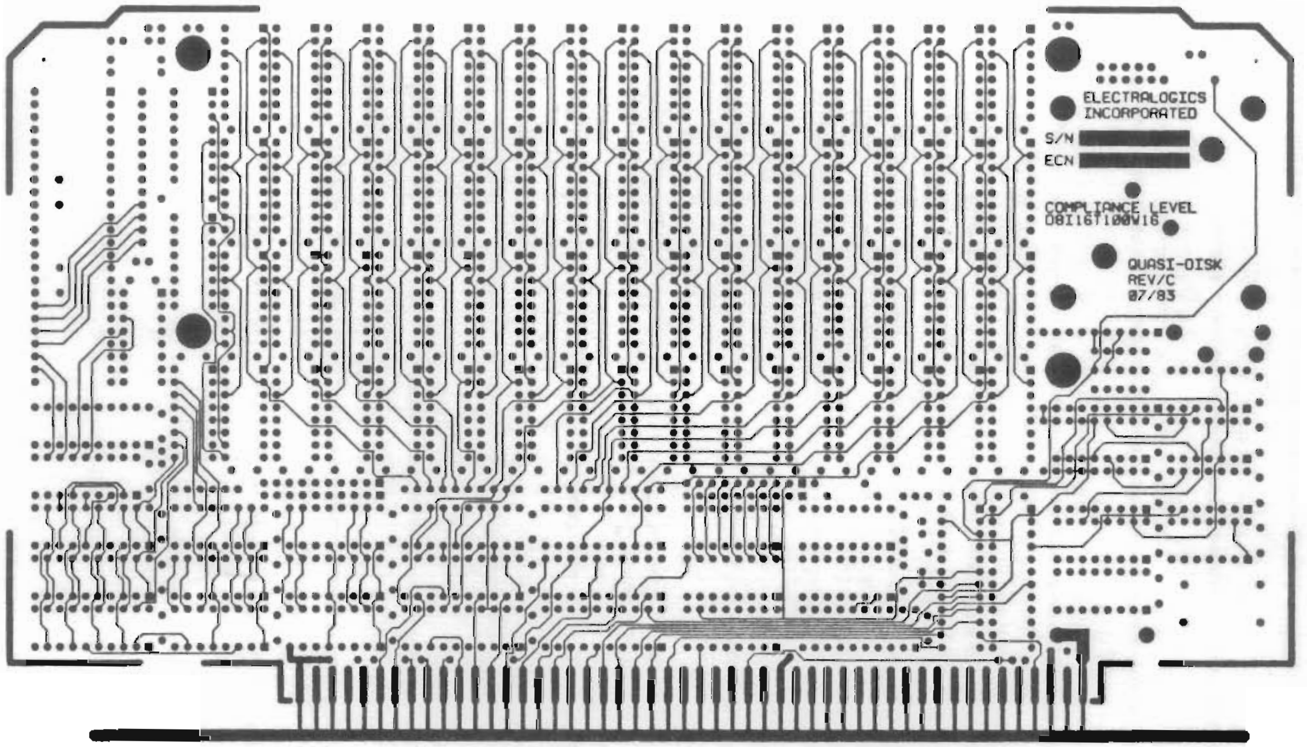
PCB TRACES - Component Side


ELECTRALOGICS INCORPORATED
QUASI DISK REV C
LEVEL 1 SIGNAL PLANE COMPONENT SIDE
CONFIDENTIAL & PROPRIETARY

D. COMPONENTS PLACEMENT



PCB TRACES - Solder Side



ELECTRALOGICS INCORPORATED
QUASI DISK   REV C
LEVEL 6 SIGNAL PLANE SOLDER SIDE
CONFIDENTIAL & PROPRIETARY