RESEARCH MACHINES

380Z

DISC SYSTEM

INFORMATION FILE

PN 10930

380Z DISC SYSTEM INFORMATION FILE

Version 1, November 1981

The policy of Research Machines Limited is one of continuous development and improvement of its products and services, and the right is therefore reserved to revise this document or to make changes in the computer software it describes without notice. Research Machines Limited make every endeavour to ensure the accuracy of the contents of this document but do not accept liability for any error or omission.

Additional copies of this publication may be ordered from Research Machines Limited at the address above. Please ask for '380Z Information File'.

Research Machines Limited is committed to the policy of continuous improvement of its products and services. To assist the Company to improve the quality of its documentation and computer software it would like to receive reports from users on their experience with using them.

Reports referring to documentation should include:

- a description of any errors you have found;
- comments on problems you have encountered with understanding and using the manual, and suggestions for improvement;
- your name and address and a brief description of your experience with using microcomputer systems.

Queries on documentation should be sent to:

The Documentation Department,
Research Machines Limited,
PO Box 75,
Oxford.
OX2 0BW

Reports referring to software errors should include:

- a written description of each step taken and each key or command you type.
- details of the hardware configuration, and the version number of the software in use.

Queries on software errors should be sent to:

Technical Support Department,
Research Machines Limited,
PO Box 75,
Oxford.
OX2 0BW

CONTENTS

1. INTRODUCTION

1.1 380Z DISC SYSTEM DOCUMENTATION

The following items of documentation are provided by Research Machines to help you set up and use a 380Z Disc System:

- **380Z Disc System User Guide:** this is the basic item of documentation which explains how to set up your system and how to use the facilities provided by the disc operating system.

- **Applications program documentation:** each item of software that you purchase is supported by documentation which explains how to set up and use the software provided.

- **Firmware Reference Manual (e.g. COS 3.4):** you will need to refer to this document if you intend to program the 380Z in machine or assembly language or if you want to take full advantage of the many features of the 380Z display.

- **380Z Disc System Information File:** you will need to refer to this document if you want to extend your basic system, for example by adding a printer.

Research Machines can also provide copies of the following documents, the last three of which are produced by Digital Research Inc., the originators of the CP/M Disc Operating System.

- **Monitor Listing:** you will only need to refer to this document if you want to modify the action of the COS monitor, for example by patching in a non-standard peripheral driver. You should specify the version of COS you require (e.g. 3.4, 3.4B, 3.4C, 4.0A etc).

- **Introduction to CP/M Features and Facilities:** You will need to refer to this manual if you want to use any of the CP/M facilities which are not described in the Research Machines documentation (for example the SUBMIT command).

- **CP/M Interface Guide:** You will need to refer to this manual if you want to write assembly language programs using the CP/M input/output facilities.

- **CP/M Systems Alteration Guide:** You will need to refer to this manual if you want to generate a version of CP/M which operates with a non-standard disc drive controller and/or peripheral devices.

CP/M is a trademark of Digital Research Inc.

1.2 THE 380Z DISC SYSTEM INFORMATION FILE

This file contains three sorts of information:

- descriptions of how to extend the hardware of your basic 380Z Disc System, for example by adding a printer.

- explanations of how to use various features of the COS monitor, the CP/M operating system and Research Machines utility programs which are not covered in the User Guide, for example: the utility program for diagnosing disc drive faults.

- technical specifications of various features of the 380Z hardware, for example: the interfaces that are supported by the 380Z.

The information is presented in three sections. The first explains how to attach a cassette tape recorder to your system and how to transfer ASCII files between tape and floppy disc (Reminder: cassette recorders cannot be attached to 80-character machines). The second section describes the features of various elements of the 380Z hardware and explains how to attach additional hardware to the system. The last section describes various aspects of the 380Z Disc System software and explains how to use them.

2. USING A CASSETTE RECORDER WITH A 380Z DISC SYSTEM

Note that this section does not apply to 380Z Varitext (80/40 character) systems as they do not have a cassette interface.

This Section explains how to attach a cassette tape recorder to a 380Z Disc System and describes how to transfer ASCII files between cassette tape and floppy discs. The tape files must be in Cassette File System (CFS) format. Files produced by TXED and data files produced by BASIC are in this format as are program files SAVED from extended BASIC version 5. Program files produced by the BASIC FSAVE command are not in this format.

2.1 SETTING UP

In order to use a cassette recorder with a 380Z Disc System to transfer files between cassette tape and floppy discs you will need the following:

- a cassette recorder with a mains lead and plug
- a muting plug, or a loose jack plug, which fits the MIC socket on the recorder
- a dual cassette controller unit (optional)

Place the cassette recorder and the cassette controller in your work area so that you can easily reach the switches on the recorder and controller when seated at the keyboard, then

- **switch off the mains supply to the 380Z Disc System**
- **connect the cassette recorder to the multi-way mains adaptor which supplies the system with power**
- **push the blanking plug into the 'MIC' socket on the recorder**
- if a controller unit is being used then:
  - **push the 7-way DIN plug on the controller into the cassette recorder socket at the rear of the processor (the socket below the keyboard socket)**
  - **push the 5-way DIN plug on the controller lead with the YELLOW BAND on it into the socket on the recorder and the jack plug ON THE SAME LEAD into the REMOTE socket on the recorder**
- otherwise, when a controller unit is not being used:
  - **push the 7-way DIN plug on the cassette lead into the cassette recorder socket at the rear of the processor (the socket below the keyboard socket)**
  - **push the 5-way DIN plug on the cassette lead into the socket on the recorder.**

Now when you turn your system on the cassette recorder will be ready for use.

****

NOTES

1. It is essential to connect all the components of your system to a single multi-way mains adaptor otherwise 'earthing interference' may adversely affect the performance of the system.

2. It is essential to insert a muting plug into the MIC socket on the recorder otherwise background sound will be mixed with computer data when recording. It may not be possible to load such a tape back into the computer.

3. When only one recorder is attached to the controller (if used) it is important to connect it to the correct lead on the controller (the one colour coded yellow). Remember you must also switch the controller to 'SINGLE'.

4. Before using a cassette recorder for the first time it is necessary to determine the correct setting for the volume control. The procedure for carrying out this check is described in Section 2.5.

5. It is essential to carry out operations involving the cassette recorder and the controller in the correct order. For example, if you forget to push down the REC and PLAY keys before initiating the transfer of a file to tape then you may lose some or all of the file. The system only controls the recorder motor. It cannot detect whether the recorder is operating in record or playback mode, or even whether the recorder is turned on or not.

6. It is essential to record a blank leader at the start of the tape in order to ensure that the recording of the file will start on the magnetic medium rather than on the plastic leader on the tape. If you are not using a dual cassette controller you should make sure that the tape is running before making a recording.

****

2.2 CASSETTE FILE SYSTEM

2.2.1 General Principles

The Cassette File System (CFS) supplied with cassette-based 380Z systems, for example as part of cassette BASIC, provides the user with a set of facilities to control input from, and output to, files recorded on cassettes. The system takes care of all data transfers to or from cassette, handles all the necessary packing or unpacking of blocks of data, and detects and reports errors.

This Section describes those features of CFS which are relevant to anyone who is planning to use the FILEX file transfer programs to transfer CFS files on tape to CP/M files on disc.

CFS File Structure

A CFS file consists of a number of blocks. Each block holds the name of the file, the block number, and the data. A checksum is also included.

The block layout is:

Byte	0:	SYN
Bytes	1-10:	Name of file
Byte	11:	BLKH
Byte	12:	BLKL
Byte	13:	Length of Data Block = n
Bytes	14-(n+13):	Data (0-255 bytes)
Byte	n+14:	Checksum

The SYN character (16 hex, 22 decimal) allows accurate synchronization when reading the tape.

The name of a file comprises 1-6 characters for the primary name, optionally followed by a full stop and 1-3 further characters for the extension. The characters should be drawn from the sets A-Z, 0-9, and $. It is recommended that programs should convert any lower case letters to upper case before writing the block. If the primary name and extension are shorter than 6 or 3 characters respectively, they must be padded with spaces.

BLKH and BLKL are the high and low bytes of the block number. Note the order of these two bytes. The first block of a file is numbered zero.

The Length of Data Block field is self-explanatory. A zero-length block represents the end of a file.

The checksum is calculated by adding together all of the bytes in the record apart from the SYN character (and the checksum!), negating, and reducing modulo 256. This means that, on reading, all the bytes apart from the SYN character should be added together, and the result should be zero (modulo 256) if the block is not corrupt.

When a file is recorded, it is usually safest to record more than one copy of each block. Copies of a block should be separated by at least 200 ms. Different blocks should be separated by at least 2 seconds of '1' tone if the tape is to be read by general-purpose programs. Specialist applications may avoid reducing this gap. Standard practice is to use two copies of each block.

Normal practice is to turn on the appropriate cassette relay bit just before reading/writing a block, and to turn it off immediately after completing the operation. If multiple copies of a block are being written, the bit should be reset only after writing the last copy. These measures allow the use of the Dual Cassette Controller.

There is no restriction on the data which may be recorded in a cassette file. However, most programs regard CTRL Z (1A hex, 26 decimal) as an end of file marker, so this character should be avoided if possible.

2.2.2 Device and file specification

When CFS is requested to open a device or file for reading or writing, it requires a file specification of the following form:

```
dev:filename.ext
```

where dev represents three characters specifying the device name. CFS recognises 5 devices:

Display and keyboard	CON
Tape out or paper tape punch	PUN
Tape in or paper tape reader	RDR
Line printer	LST
Cassette file	CAS

If there is no colon present then a cassette file, CAS:, is assumed by default.

filename — represents the name, up to six alphanumeric characters long, of the file on the given device which is to be opened. A filename is required ONLY if the device name is CAS:

ext — represents an extension, or file type, of up to 3 alphanumeric characters. If this is omitted, a blank extension will be assumed by default. If the extension is omitted, the period may be omitted also.

2.2.3 File Format

When outputting a file CFS normally writes two copies of each block of the file to tape.

When CFS is inputting a file it starts the cassette recorder and looks for an interblock gap and a block header. The file name and block number of any block header found are usually displayed on the console usually on the bottom line of the screen. Automatic paging is disabled. CFS keeps reading blocks until it finds a block header which matches the block it is looking for. CFS then reads the block into its input buffer. If a checksum error is detected, then CFS indicates a soft error by showing an arrow (→) in front of the filename displayed on the screen, and then tries to read a later copy of the same block from the tape. If CFS exhausts the copies of the block, without having read an error free copy, it returns to the user's program with a hard error indication. It is then the responsibility of the user's program to decide what action to take.

2.3 THE FILE EXCHANGE PROGRAM:FILEX

FILEX is a program which allows the user to transfer Cassette File System (CFS) files to disc and vice versa. It will only work on text files. These include files produced with TXED or ZAS. FILEX will also transfer BASIC programs produced with Extended BASIC Version 5 or later, provided that the program was saved in the usual ASCII format, rather than in internal format using FSAVE.

BASIC programs for the earlier cassette BASICs should be transferred using the special programs BASLOAD and BASDUMP provided on the BASIC distribution disc. Internal format files produced using the FSAVE command in Extended BASIC can only be read by the particular BASIC which produced them.

2.3.1 Using FILEX

The following example illustrates a typical transfer, of a file called CIRCLE.BAS from disc to cassette. It is shown without comment to enable the casual user to get started without needing to read the full instructions. In case of difficulty, consult the rest of this section.

Most lines typed by the user must be terminated by the RETURN key. The only exceptions are typing CTRL C to leave FILEX, and typing a space to end a pause. User input is underlined.

```
A>FILEX

Cassette/Disc exchange Version 4.0 A
Copyright (c) 1981 by Research Machines

Read or write to cassette (R/W): W
Cassette file name: CIRCLE.BAS
Disc file name: CIRCLE.BAS
Switches (press RETURN for normal)
:                                          press RETURN here

Transferring from disc to cassette
Cassette specification: CIRCLE.BAS
Disc specification: CIRCLE.BAS
Switches in force: None

Is this correct (Y/N):Y
Press RECORD on recorder, hit space bar    type a space
Switch off recorder
Transfer complete

Read or write to cassette (R/W): ↑C         type CTRL C

A>
```

Typing mistakes can be corrected in two ways. Typing DELT will delete the last character typed excluding RETURN and CTRL C. If the error is near the start of a line, it may be quicker to type CTRL U. This character erases the entire line, and you can then try again.

Attempts are made to keep the line as it appears on the screen in step with

what the computer has in its buffer. However, it may get out of step. Typing CTRL R will cause a new copy of the line to be printed.

Typing CTRL C causes FILEX to terminate and control is returned to CP/M.

FILEX is loaded from the system disc in the normal way by typing its name from the CP/M command line.

A>FILEX

2.3.2 Direction of Transfers

FILEX will load and print a sign on message. It will then print

Read or write to cassette (R/W):

The response should be either R, to transfer a file from cassette to disc, or W to transfer a file from disc to cassette. Any other response will cause the prompt to be repeated until a valid letter is given. E.g.

Read or write to cassette (R/W):R

Note that the response must be terminated by the RETURN key.

2.3.3 Cassette File Names

After getting the transfer direction the program will prompt for a cassette file name. This should be a valid Cassette File System file name, which consists of primary name and an optional secondary name or extension. If given, the extension must be separated from the primary name by a dot (.). The primary name consists of one to six characters drawn from the sets 0-9, A-Z, a-z, and $. Lower case letters are changed into the corresponding upper case letters. The extension consists of one to three characters drawn from the same sets.

Instead of a file name, it is possible to specify an unstructured device. The available devices are:

CON:	Console
TTY:	Console
RDR:	Unstructured cassette (input only)
PUN:	Unstructured cassette (output only)
LST:	Printer (output only)
LP:	Printer (output only)

In addition, a file name may be preceded by CAS:. This is the default.

For example:

Cassette file name: CIRCLE.TXT

If the file name is incorrect, FILEX will print an appropriate error message and prompt again for a file name.

2.3.4 Disc File Names

The program will then prompt for a disc file name. A valid disc file name comprises a device name, a primary file name, and a secondary name or extension. The device name is optional, as is the extension.

A device name can be one of the unstructured device names described above for cassette file names. More commonly, it can be a drive name. For users running under CP/M version 1.4, valid drive names are A:, B:, C: and D:. If the drive is omitted, the logged on drive is assumed.

Note that the extension part defaults to no extension. Many languages, amongst them BASIC, assume an extension, for example .BAS. If a file has an extension FILEX must be told explicitly.

As with the cassette file name, badly formed file names are reported and the prompt repeated. An example of a correctly formed name is:

Disc file name: CIRCLES.BAS

2.3.5 Switches

The next prompt asks

Switches (press RETURN for normal)
:

The switches allow unusual file transfers. The available switches are:

Bn The B switch sets the block number at which to start reading the cassette file. The block number n should lie in the range 0 to 255, with 0 the default. This switch is mainly intended to allow retrieval of cassette files which have no block 0. Such files are sometimes created accidentally by recording the start of the file on leader tape or by overwriting one file by another. This switch has no effect when transferring a file from disc to cassette.

F This switch sets the cassette speed to fast (1200 baud). This is the default, and the switch is included to maintain compatibility with earlier versions. Note that the cassette speed selected via the COS or Front Panel O command under COS 3.0 has no effect on transfers within FILEX.

In This switch sets the interrecord gap to be n tenths of a second. The default is 20 (2 seconds). The interrecord gap can sometimes be reduced, but it can lead to difficulties when attempting to read the file. This switch has no effect when transferring a file from cassette to disc. FILEX will not allow an attempt to reduce the interrecord gap below 0.2 seconds.

Nn This switch sets the copy count, which defaults to 2. On writing to cassette, the specified number of copies of each block are written. On reading, FILEX will read copies of each block until either it reads one which is not corrupt, or until the specified number of blocks has been read. If all copies of a block are found to be corrupt, FILEX will report a hard error, as described below.

FILEX will only allow copy counts in the range 1 to 9. In practice, the default of 2 is adequate in almost all circumstances, and the extra security offered by increasing the copy count is not worth the extra time taken to read a file. Reducing the copy count to 1 is often worthwhile.

Q The Q switch disables the recorder control messages, such as

Press PLAY on recorder

which FILEX prints out as appropriate. This might often be done by users with Dual Cassette Controllers.

R Normally, FILEX does a disc reset before writing to the disc. This reduces the possibility of a BDOS Read Only error message, although the determined user could still get himself into difficulties with this. Sometimes, however, a disc reset is undesirable, and setting the R switch disables this safety check.

S The S switch sets the cassette speed to be slow (300 baud). This speed is much more reliable than the fast (1200 baud) speed normally used, especially when attempting to read on one machine a cassette prepared on another. However, it does take about four times as long to read or write a slow tape, and, as with the copy count, the extra security may not always be judged worthwhile. Note that the cassette speed selected with the COS or Front Panel O command (under COS 3.0) does not affect the cassette speed used within FILEX. Also note that, in contrast to previous versions of FILEX, the effect of an S switch lasts only for the transfer for which it was specified. In previous versions, the effect of an S switch lasted until an F switch.

Multiple switches may be separated by spaces, or they may be contiguous. It is legal, although pointless, to specify any switch more than once. In the case of the F and S switches, if both are given, the last one is the one actually used.

2.3.6 Transferring the File

After verifying that the file names, switches, and direction of transfer have been set up correctly, FILEX will transfer the file. It will read the source file until either it has read the entire file or until all available memory has been filled. It will then write the file to the destination. This process will be repeated until the file has been transferred. The message

Transfer complete

signifies that the transfer has finished.

If a disc file is being transferred to cassette, FILEX checks that the disc file exists. If it does not, the error message

Disc file not found

is printed. If the transfer is from cassette to disc, FILEX checks that the disc file does not already exist. If it does, the message

File exists -- replace (Y/N):

is printed and FILEX waits for a response. If the answer Y or YES is entered, the existing file of that name is erased and the transfer proceeds. If the response is N or NO, the transfer is aborted. Otherwise, the 'File exists' prompt is repeated.

As an aid to users without Dual Cassette Controllers, FILEX will print out messages instructing the user to manipulate the controls of the cassette recorder (but see the Q switch above). These are mostly self-explanatory, but there are two which might cause confusion.

When FILEX is about to write a buffer-full of text, it will print the message

Press RECORD on recorder, hit space bar

FILEX will then pause until a key is typed at the keyboard. Note that there is no need to type RETURN .

When FILEX stops reading a cassette file because its buffer is full, it will print

Switch off recorder, rewind a little

This rewind is necessary because, unless the user is very alert, or a Dual Cassette Controller is in use, the recorder will have run on past the correct part of the tape. The tape should be positioned just before the point where FILEX started printing the message. It does not matter if the cassette is rewound too far. However if it is not rewound far enough the blocks will be missed. In practice, this message will hardly ever occur because of the rarity of such large files on systems without Dual Cassette Controllers.

At any time during the reading or writing of a cassette file, the operation may be aborted by typing any control key on the keyboard. Note that such keys as RETURN and LINE FEED generate suitable codes. Once a transfer has

been aborted, it cannot be resumed, so the entire operation must be restarted.

If, when reading a cassette file, FILEX cannot find an uncorrupted copy of a block, it will prompt

```
Rd Err--A=Accept, B=Abort, R=Retry?
```

Accepting the block signifies that the user wants the block, errors included. The tape will probably have to be rewound a little if a Dual Cassette Controller is not in use. Aborting completely aborts the entire transfer. The retry option involves rewinding the tape and attempting to read a valid copy of the block. Note that the response to this question must not be terminated by RETURN .

2.3.7 Help

When FILEX is expecting a response to a question, the word HELP may be typed. This will cause FILEX to print some explanatory information, and then go back to the prompt. The only problem this causes is that transfers involving a file called HELP become somewhat tricky. The simplest way to transfer such a file is to specify a file name of HELP., i.e. append a dot to the name.

2.4 CASSETTE SUPPORT IN COS 3.4

A full description of the cassette interface and the associated COS facilities is given in the FIRMWARE REFERENCE MANUAL . Disc versions of COS 3.4 do not support slow cassette loading and saving (300 baud rate). A utility program entitled SCASS is provided to change the cassette I/O transfer vectors in COS workspace in order to provide the slow cassette transfer rate.

- **turn the system on, and load CP/M**
- **type SCASS RETURN**

SCASS will be executed and CP/M re-entered. Note that FILEX contains its own slow handler for the cassette system and that using SCASS with FILEX will bring no benefit.

After SCASS has been run the SLOW HANDLER remains effective until the RESET button is pressed.

The COS commands

L Load a machine code program from cassette

D Dump memory to cassette

which are described in the FIRMWARE REFERENCE MANUAL can be used with a disc system provided that, when necessary, the utility SCASS is used to select slow mode.

2.5 TESTING A CASSETTE RECORDER

A utility program TSTSYS is provided to check the volume and phase of a cassette recorder.

2.5.1 Setting the replay volume

When a cassette recorder is first interfaced to a system the correct replay volume should be ascertained. The procedure is as follows:

- **connect up the recorder, turn the system on and set the controller to MANUAL**
- **press down the REC and PLAY keys on the recorder to record a minute or two of the 2400 Hz tone on tape** (this is present on the cassette interface when COS is quiescent (i.e. waiting at the COS prompt - see FIRMWARE REFERENCE MANUAL)
- **if your recorder does not have automatic level control on record, set record level to near full modulation**
- **stop the recorder, rewind the tape and set volume control on its lowest setting**
- **load CP/M**
- **type TSTSYS RETURN**

TSTSYS will be run and the message

Test MEMORY, VOLUME, or PHASE(M/V/P):

- **type V**

will appear

You should now see the display screen fill with dots.

- **Press the PLAY key on the recorder.**

This plays back the previously recorded section of 2400 Hz tone.

- **Slowly increase the volume control on the recorder until some of the characters entering the screen are asterisks. Then reduce the volume very slowly until no asterisks are seen.**

This is the correct setting for the volume control - make a note of it for future use. On recorders with a tone control, set this control for minimum high frequency cut (i.e. maximum hiss).

2.5.2 Checking the recording quality

COS demodulates data on tape in slow format by measuring the duration of each cycle of tone. When a bit is recorded (8 cycles at a nominal 2400 Hz for a '1', 4 cycles at 1200 Hz for a '0') or in fast format (2 cycles at 0 2400 Hz for a '1', 1 cycle at 1200 Hz for a '0') and if a change in frequency is required, the transition takes place as the wave form crosses the baseline. If the phase inversions between record and replay on a recorder are such that COS reads cycles between the 'wrong' baseline crossings, the result is that cycles around a frequency transition are measured incorrectly, since the 'cycle' includes half a cycle at 1200 Hz and half at 2400 Hz. If you are having difficulties loading programs from a tape then the program TSTSYS can be used to test the quality of the recording. The procedure is as follows:

- **connect up the recorder**
- **turn the system on, insert a system disc including the file TSTSYS. COM and load CP/M**
- **put the tape to be checked in the recorder and set the controller on MANUAL**
- **type TSTSYS RETURN**
- **type P**
- **press the PLAY key on the recorder.**

After a few moments, when the 2400 Hz tone starts to replay, you should see a vertical line of dots appearing in the centre of the screen, with a broader broken line to the left of it. The numbers on the right of the screen give the duration (in hex) of each measured cycle. The line of dots indicates the value used to discriminate between '0' and '1' cycles (47 hex), and the broad line on the left shows cycles shorter than this ('1' cycles, 2400 Hz). You can halt the program by stopping the recorder.

When the tape reaches the data area, a second broad line should appear to the right of the dotted line, representing '0' cycles, 1200 Hz. Once you have got used to looking at the display, you may notice some 'cycles' in each broad line which are slightly closer to the dotted line. These are transitional cycles (a cycle immediately adjacent to another of the opposite frequency) and indicate 'correct' phasing. If, on the other hand, there are frequent rogue cycles near the dotted line ('half-and-half' cycles) the replay quality of the tape is not satisfactory.

TSTSYS will also give an unsatisfactory quality indication if the recording has been made in the 'wrong phase' - this will not happen if you use recorders provided by Research Machines.

3. HARDWARE NOTES

3.1 MEMORY TESTING

The operation of the COS monitor relies on the integrity of a small area of RAM so a gross memory fault will result in failure of the system to work at all. Less catastrophic memory faults may be picked up by use of the 'Front Panel' fill and test memory command, P, or by the failure of the monitor to set location 'HIMEM' correctly after a system reset. HIMEM normally contains the address of the highest available location plus one. (See Firmware Reference Manual.)

A program TSTSYS which carries out a more thorough memory test is provided on your CP/M Software Issue Disc. This program fills memory with a pseudo-random bit pattern and then compares the current content of a location with the value that was loaded. In each 'pass' of the test each byte is tested with 256 bit patterns. During each fill and compare cycle a test pattern for each byte is constructed from the exclusive-OR of a constant, the low byte and the high byte of its address. Thus each byte within a memory 'page' of 256 locations receives a different pattern and the order of test patterns between pages varies. In this way errors in which bits are 'stuck' at either 0 or 1, and interactions within a byte, are readily detected. All memory is written before being tested to detect address interaction.

The completion of each cycle is indicated by the printing of a slash (/). At the end of a successful test of the whole memory TSTSYS prints the pass number and then begins another test. Testing continues until either a memory fault is detected or CTRL C is typed.

If a memory fault is detected TSTSYS halts with entry to the Front Panel via a breakpoint (See Section 4.1.1). Register pair HL points to the byte in which the fault has been detected (with its content displayed further along the HL row of the register section of the Front Panel), while Register A contains the expected pattern. Register pair DE contains the pass number.

To test for memory faults:

- **turn the system on, insert a system disc and load CP/M**
- **type TSTSYS** [**RETURN**]
  This will result in the message:

  Test MEMORY, VOLUME or PHASE (M/V/P):

- **type M**
  this will result in the output of a / at the SUCCESSFUL completion of the testing of each page of memory and a pass number at the SUCCESSFUL completion of the testing of the whole memory. Allow the program to go through several passes and then
- **type** [**CTRL C**]
  to return to COS.

****

NOTES

1. The memory size tested will be that known to the version of CP/M under which the test is run.

2. If you locate a memory fault then you can either

   - have the faulty board repaired, or

   - temporarily isolate the fault by re-configuring CP/M to operate in a smaller memory size (See MOVCPM, Chapter 4 of 380Z User Guide)

****

3.2 DISCS

3.2.1 Disc Formats

Research Machines' disc systems use soft-sectored floppy discs. Storage on the discs is organised into a number of concentric tracks, each of which is divided into a number of sectors. A sector consists of a header block containing the track and sector number, which is recorded on the disc at the time it is formatted, followed by a data block of 128 bytes. The header block also provides timing information to the disc controller.

Mini Disc Systems

Single-density discs used by RML MDSs have 40 tracks of 16 sectors. The first three tracks are reserved for the operating system and the fourth for the directory, leaving 72K bytes available for storage of programs and data.

Full Disc Systems

Single-density discs used by RML FDSs have 77 tracks of 26 sectors. The first two tracks are reserved for the operating system and part of the third for the directory, leaving 241K bytes available for storage of programs and data.

3.2.2 Disc Verification

On your CP/M Software Issue Disc you should find a command file called VERIFY.COM. The program VERIFY can be used to check every sector of a disc to see if it is readable. If a sector cannot be read, the program will invite you to retry reading it. Occasional "soft" errors which disappear on retrying are to be expected. Running the program VERIFY does not alter a disc in any way.

The VERIFY program tests whether all sectors of a disc are readable without error. VERIFY is run in the usual way by typing its name followed by RETURN and after displaying the title and version issues the prompt

```
Verify which disc (A,B,C,D):
```

Enter the disc unit you wish to test or CTRL C to return to CP/M. The message

```
Insert disc in d, then type RETURN:
```

appears, where "d" is the unit selected in the previous step. Type RETURN to proceed or any other character to return to the previous step. All sectors of the disc are now read and, if no errors occur, the message

```
Verification complete, no errors
```

is output, followed by the initial prompt.

If an error occurs, it is reported in the form

```
Read error on B at T=37 S=11 Err=C
```

where "T" and "S" are the track and sector and "Err" is the disc error type described in Section 3.2.3. A standard retry procedure is attempted, and if successful a "soft error" is reported, else a "hard error". At the end of a pass in which errors occurred a summary is displayed similar to the following:

```
Errors detected
03 soft errors, no hard errors
```

Occasional soft errors are acceptable but if hard errors occur the disc should be reformatted. If errors persist after formatting the disc should be discarded.

VERIFY can be interrupted by typing any key; the initial prompt appears.

3.2.3 Diagnosing and patching discs

On a CP/M Software Issue Disc is a program called MDOC for Mini Disc Systems or FDOC for Full Disc Systems. It is a disc diagnostic program controlled from the console keyboard. It allows any sector on the disc to be read into a memory buffer or written back to disc, but since the buffer can be changed by using the 'Front Panel' (See Section 4.1) it is quite a powerful disc patching tool. Most users will need it only occasionally to recover from disc read errors but it is as well to be prepared by running the program to familiarise yourself with its commands. Information about the program is built in to the program itself and can be displayed by typing H. The following notes are intended only to supplement this display.

The program communicates with the user by means of a status display which is displayed in response to each command and looks like:

```
U=01 T=12 S=02 D=E5 ST=00000000
```

This particular display means that you are logged on to unit 1 (where 0 is disc A, 1 is disc B etc.) and that track 12 and sector 2 are selected. D is the last data byte read or written. ST is the disc controller status register.

The meaning of the status register bits depends upon the last controller operation, as follows:

Bit	Seek	Read address	Read	Write
7	Not ready	Not ready	Not ready	Not ready
6	Write protect		Rec type	Write protect
5	Head engaged		Rec type	Write fault
4	Seek error	ID not found	Rec not found	Rec not found
3	CRC error	CRC error	CRC error	CRC error
2	Track 0	Lost data	Lost data	Lost data
1	Index	DRQ	DRQ	DRQ
0	Busy	Busy	Busy	Busy

Before attempting to read or write a sector you must select the disc unit and initialise it (e.g. for drive B, issue the command 1U, followed by Z), then set the track and sector (12T, 2S for the above example). Now the R command can be given to try to read the chosen sector. If the read is

successful the status register will contain all zeroes; if not you may repeat the R command any number of times to try to overcome a "soft error". The W command writes back the memory buffer to disc. Thus if you wish to modify the contents of a sector, you can read it into the buffer and then type CTRL F to enter the Front Panel. The buffer is at 1000 to 107F Hex, so typing **M>1000** will set the memory pointer to the first data byte. After modifying the buffer, type K to return to MDOC then W to write the buffer back to disc.

If a read error occurs, the sector is still read into memory. It can of course be rewritten without modification and will usually then reread correctly. However it is fairly probable that there will be some incorrect data, so this procedure is usually only worthwhile on a text file which can later be corrected with the Text Editor. If you are very skilful you could attempt to use the Front Panel to correct the errors.

To use the diagnostic program

- **turn the system on, insert a system disc and load CP/M**
- **type MDOC (or FDOC) RETURN**
  This will result in the message
- **insert disc to be checked in drive A**
- **type H**
  Make an appropriate range of tests using the H(help) command to guide you
- **type CTRL C**
  to return to COS and reload CP/M.

****

NOTES

1. All numbers are in HEX.
2. Valid track numbers are 0-27 (MDOC) and 0-4C (FDOC)
3. Valid sector numbers are 1-10 (MDOC) and 1-1A (FDOC)
4. All numbers in the status display refer to the current contents of the memory buffer. Thus, for example, in the above example, if 1T,3S was to be typed the display would remain unchanged until a read or write command is issued.

****

3.2.4 CP/M Disc System Error Messages

There are three types of CP/M Disc error that can arise - the SELECT, READ ONLY and BAD SECTOR errors.

A SELECT error will occur if you attempt to access a disc unit outside the range A to D. Cause a select error by typing DIR E:. Note that typing any key reloads CP/M. Try to log in to disc E by simply typing E:. Again this results in a select error but now you are in an endless loop. Typing a key causes CP/M to access E and it fails again. Escape by RESET and reload CP/M. Some programs (e.g. ASM, ZASM) rely on the select error mechanism to detect an out-of-range disc unit.

A READ ONLY (R/O) error results from an attempt to write to a disc which has been set to be read only, either by the STAT program or because the disc in the relevant drive has been changed. You can cause this error by loading CP/M, then changing the disc in A to any other disc (except a direct copy) and then attempting to write a file onto the disc (using PIP for example). After the error occurs, typing any key reloads CP/M and allows writing. This error mechanism is designed to prevent inadvertent writing, but it can sometimes prove more of a nuisance than a benefit. A rule which prevents it occurring is to remember to type CTRL C whenever you change a disc. Some programs (e.g. TXED, BASIC) have a special command to reinitialise the disc system; this allows you to change a disc whilst running a program. Note that this type of 'write protection' differs from the method of physically protecting a disc by a notch. The latter method gives a type P BAD SECTOR error if writing is attempted (see below).

The BAD SECTOR error message is potentially the most serious, for it may indicate an actual error in disc transfer. This message is preceded by a line giving information about how the error occurred. Simulate a bad sector error by removing the disc in A, then typing DIR. You will see a display something like

```
O       00 03 01 0080
BDOS ERR ON A: BAD SECTOR
```

The upper line indicates, from the left, that the error is type O and occurred while addressing disc 0 (where 0 to 3 corresponds with A to D), track 3 sector 1, and attempting to transfer data to or from address 0080 Hex. Possible error types are O, P, C, A, D and V.

O means that the disc is off line. It indicates that there is no disc in the drive, that it has been improperly inserted, that you are addressing a non-existent drive, or attempting to access the second side of a single-sided disc.

P means that the disc is physically write protected and that writing is being attempted. It is due to a hardware interlock triggered by covering the write enable notch for MDS systems or uncovering the write disable notch for FDS systems. It also occurs when READING a file greater than 16K long on a protected disc (e.g. BASICSG2) because CP/M attempts to write to the directory. Pressing RETURN continues reading the file (ignoring the error message).

C indicates that a checksum error occurred whilst reading the sector. This is the only serious error that is at all common and even so should occur rarely. It probably arises because a speck of dirt has intervened between the head and the disc at the moment of writing. The sector becomes marginal and later cannot be read correctly. (Normally, if a sector cannot be reread correctly at the time of writing, it is rewritten automatically.) In almost all cases a

sector which produces a C error can be recovered by merely writing data to it, but since it is the data which it contained that is important and since this is almost certainly corrupt this fact is of little comfort! As mentioned above, MDOC or FDOC can be used to recover the data and it may be possible to correct it.

A indicates that an error occurred whilst trying to read the sector header and may occur during reading or writing. Usually the cause is a worn or corrupted disc since the disc controller never alters the header block, and it implies that the disc should be reformatted or discarded. Occasionally it can occur as a result of the controller becoming 'confused' by a checksum error so it is worthwhile proceeding as for a C error before adopting a more drastic solution.

D indicates that the 'data lost' bit of the disc controller status register is set. It implies a hardware malfunction.

V can occur only during writing and indicates an error during the read after write check. Like D, it implies a hardware malfunction.

3.3 INTERFACES FOR THE 380Z

The 380Z is fitted with one parallel interface and can be fitted with a variety of different serial interfaces.

3.3.1 User I/O Cable Parallel Interface

The 25-way parallel interface socket on the rear panel of the 380Z is labelled P1. On the earliest 380Zs a parallel interface was optional, so there may not be a P1 socket or it may not be labelled. To identify a suspected parallel interface socket:

- **remove the lid of the case and find the CPU board**
- **locate the 20-way edge connector at the top left hand edge of the PCB** (when the board is viewed from the component side) and if there is a cable plugged into this connector trace it down to the back panel
- the 25-way socket attached to this cable is the parallel interface socket.

If there is no cable plugged into the 20-way socket then in order to create a parallel interface for your system you must

- **obtain from Research Machines a USER I/O CABLE**
- **follow the procedure below for connecting up this cable.**

The User I/O cable has been designed to connect the 20-way user port socket on the CPU board to a 25-way sub-min 'D' type connector which can be mounted through the rear panel of the 380Z case.

These instructions will help you install the cable correctly into your 380Z.

1. **Remove the lid of the 380Z's case.**
2. **Remove a suitable blanking panel from the rear of the case and attach the 25-way 'D' socket using the screws provided.**
3. **Now plug the 20 way connector into the user I/O port on the left-hand end of the CPU board.**

   The edge of the flat cable with the brown or coloured line on must point to the edge of the CPU board. Failure to do this will result in damage.

4. **Replace the lid of the case, switch on the computer and attach a suitable printer (CENTRONICS compatible), via a suitable cable, to the 25-way user I/O connector on the rear of the case.**

Testing

To test the parallel port, first select it as lineprinter option 3 using the Front Panel. Now type CTRL T on the keyboard. The computer will enter a small test routine that causes everything you type to be printed ONCE on the screen and ONCE to the lineprinter (in this case the Parallel Port). The computer will not 'understand' the characters you type, only print them. To exit test mode type CTRL F to get into the Front Panel.

Specification of 380Z USER I/O CONNECTIONS

CPU BOARD 20-Way		REAR PANEL 380Z 25-Way 'D' Sub. Min. Socket	
PIN	SIGNAL	PIN	SIGNAL*
1	D0 IN	1	Busy
2	D1 IN	14	Fault Input (N)'
3	D2 IN	2	
4	D3 IN	15	
5	D4 IN	3	
6	D5 IN	16	
7	D6 IN	4	
8	D7 IN	17	
9	D0 OUT	5	Data Bit 1 (LSB)
10	D1 OUT	18	Data Bit 2
11	D2 OUT	6	Data Bit 3
12	D3 OUT	19	Data Bit 4
13	D4 OUT	7	Data Bit 5
14	D5 OUT	20	Data Bit 6
15	D6 OUT	8	Data Bit 7 (MSB)
16	D7 OUT	21	Data Strobe (N)'
17	+ 5V	9	
18	-12V	22	
19	+12V	10	
20	V	23	Circuit Common (Ground)

* When used to drive a parallel interface printer with Centronics compatible parallel interface.

' Indicates signal that is active low

LSB = Least Significant Bit MSB = Most Significant Bit

Input is to a 74LS244 which requires standard TTL Input voltages.

High Level Input Voltage	Min. 2V
Low Level Input Voltage	Max. 0.8V
High Level Input Current	20uA at 2.7V
Low Level Input Current	-200uA at 0.4V

Output device is 74LS374:

High Level Output Voltage	Min. 2.4V at Ioh = -3mA
Low Level Output Voltage	Max. 0.4V at Iol = 12mA

3.3.2 Serial Interfaces - Identification

Before using a serial interface on your 380Z, you must first establish which serial interface you have.

There are several types of interface that you can use on a 380Z. The most common are SIO-2 and SIO-4. The SIO-4 is a high speed interface supplied in various configurations (SIO-4C, SIO-4F, SIO-4M and also the SIO-5 and 6 which are really versions of the SIO-4 configured to run at different addresses.) A type of interface supplied on earlier 380Zs is the SIO-1, another high speed interface.

The suffix B added to an interface name denotes a 20 mA current loop interface, the suffix C denotes a cassette version of an interface.

General Information

The SIO-1 was available in three versions, the SIO-1A which was RS232/V24 standard, the SIO-1B which was 20 mA current loop standard and the SIO-1 option 4 which was really an SIO-4. These are interfaces in which the serial to parallel and parallel to serial conversion is done by the electronics of the interface. Because of this, receiving or transmitting a byte of data only takes a few microseconds of the processor's time, and therefore the system can be receiving and transmitting data simultaneously. An SIO-1A or B was used in cases which required simultaneous input and output, such as operating the 380Z as a terminal. The SIO-1A is now superseded by the SIO-4.

The SIO-2 and SIO-2B (formerly SIO-3) are level conversion interfaces which convert the TTL level signals of the computer to the standard serial interface levels and vice versa. The SIO-2 is an RS232 interface while the SIO-2B is a 20 mA current loop interface. The computer communicates with these interfaces through the user I/O port, and the parallel to serial conversion and timing is carried by software resident in the monitor. The SIO-2/2B interfaces are suitable for input or output, but these interfaces are not capable of transmitting and receiving simultaneously. If you want to do this use an SIO-4, 5, or 6.

The SIO-4 is an RS232 (V24) interface which is fitted as standard on all disc-based systems. The SIO-4C is the standard serial interface for cassette-based 380Zs.

The SIO-5 and 6 are the same as the SIO-4 except that they are located at different addresses. These interfaces may thus be used together in the same machine.

How to Locate and Indentify Interfaces

The SIO-1A and SIO-1B

The SIO-1 is a standard sized 380Z board occupying one 380Z board slot.

It can be located in any of the vacant 380Z board slots. It will have a 3-way or 4-way cable leading from an on-board 14-pin plug at the top edge of the board to a socket on the 380Z rear panel.

WARNING

Do not mistake an SIO-1A for an SIO-1B. Internally these two interfaces are almost identical in appearance. They can be identified by the type of outlet socket on the 380Z rear panel. The SIO-1A has a 25-way outlet socket which is an elongated multi-pin socket. The SIO-1B has a ROUND 6-way socket.

THE SIO-2 and SIO-2B/SIO-3

The SIO-2 interface is a small non-standard board and is about 4" x 3". It stands on three red pillars and has a multi-coloured cable leading from one end and a 3-way or 4-way cable leading from the other end of the board, to an outlet socket on the 380Z rear panel.

WARNING

Do not mistake an SIO-2 for an SIO-2B (formerly called the SIO-3). Internally these two interfaces are almost identical in appearance. They can be identified by the type of outlet socket on the 380Z rear panel. The SIO-2 has a 25-way outlet socket which is an elongated multi-pin socket. The SIO-2B has a ROUND 6-way socket.

The SIO-4C (Also refers to former SIO-1A lineprinter Option 4), SIO-5 or SIO-6.

The SIO-4C is a standard size 380Z board occupying one 380Z board slot. It is a standard serial interface for cassette-based 380Zs. It may be located in any of the 380Z board slots and will have a 10-way multi-coloured flat cable leading from the top edge of the board to a 25-way outlet socket on the 380Z rear panel.

SIO-4 and SIO-4C (also applies to SIO-5 and SIO-6)

Installation

The SIO-4 comes installed as part of the 380Z disc system. If your SIO-4C (cassette-based version of the SIO-4) is not installed in your system then follow the normal procedure for plugging an extra board into the 50-way bus, and mount the external 25-way connector in a suitable cut-out in the rear panel, after removing a blanking plate. The ten-way connector plugs into the rear top edge of the board. The plug should be inserted with the brown wire towards the rear of the board when assembled into the machine.

Connection

The SIO-4 I/O connector is a 25-way 'D Submin.' female socket mounted on the rear panel of the 380Z.

Technical details

Technical details (e.g. pin connections) for serial interfaces are given in Section 3.6.5.

3.3.3 Serial Interfaces - use of SIO-4

Selection

Before using an SIO for output or input it is necessary to select it, as described in full in Section 3.5.1. This ensures that the correct handler in COS is used. For example, the SIO-4 may be selected as the system lineprinter option either on power on, after RESET, or from the Front Panel. After power on or RESET type the letter O (for Options), 4 (for output device handler 4) and then a number between 0 and 6 for the required Baud rate, as given by the following table:

0 = 110 Baud
1 = 300 Baud
2 = 600 Baud
3 = 1200 Baud
4 = 2400 Baud
5 = 4800 Baud
6 = 9600 Baud

If you require a non-standard baud rate you will have to write your own handler and link it to COS - for guidance refer to the 'SIO-4 handler' in the COS listing (available separately).

Reset

will always cancel any existing lineprinter option selection, so repeat this procedure after reset. If you did not select your options immediately after power on or RESET, you can select them at any time by going into 'Front Panel' by typing Control-F, and then typing O, 4, etc., as above. You can then return from Front Panel to the program you were in by typing K. Note however that K will often appear to have little effect as the mode you were in will usually be 'waiting for an input of some kind' and you will simply return to this wait.

Outputting

Once the option has been selected, high level software lineprinter commands will be directed to the SIO-4 device. For details of these commands refer to the relevant manual. Outputting to the SIO-4 from machine language, once the option has been selected, can be achieved by using an EMT LPOUT call, with the ASCII code in register A.

Inputting

Once the option has been selected, an EMT S4KIN call can be used as an SIO-4 reader routine, and EMT S4KTL can be used to test the status. For more information refer to the 'Firmware Reference Manual'.

3.3.4 Other Interfaces

A number of other interfaces can be fitted to the 380Z and appropriate software is available to support their use.

IEEE-488 Interface

The IEEE-488 standard is a bus structure for interconnecting a number of devices in order for them to exchange data. Software support for the IEEE-488 bus and interface is provided through a number of special statements added to the standard Research Machines Disc Extended BASIC. For full information please refer to:

Research Machines IEEE-488 Interface BASIC USER MANUAL

PIO/RTC Interface Development Board

This is a general purpose board giving full access to the facilities provided by the Z80 PIO and Z80 RTC features. Software support is provided through a number of BASIC programs and ZASM assembler code routines. For full information please refer to:

Z80 PIO and RTC Manuals
Research Machines PIO Support Routines.

Analogue I/O Board

This board provides 16 channels of analogue input and 2 channels of analogue output, all to 10 bit resolution. Software support for this board must be through user written assembler code routines, which can be linked to Research Machines high level languages (BASIC, FORTRAN or ALGOL). For full information please refer to:

Research Machines 236-222 380Z Analogue
Input/Output Board Operating Manual.

3.4 ADDING INPUT DEVICES

It is quite easy using the 380Z Disc System to read in programs and data from additional peripheral devices. This Section describes how to transfer ASCII data (such as programs) from paper tape or cards to a disc file. It also contains some suggestions for implementing a simple link between a 380Z and another computer. It is assumed that the peripheral device (e.g. Teletype, mark sense card reader) has a serial interface (RS232 or 20 mA current loop) and that a suitable serial interface (SIO-1, SIO-2 or SIO-4) is available (See Section 3.3).

3.4.1 Modifying PIP.COM

The first step is to make a modification to the PIP program such that the device name 'INP:' is associated with the new peripheral. The patch files S1INP.HEX, S2INP.HEX and S4INP.HEX included on the CP/M disc are for this purpose. The patched PIP can then be used to transfer data from the new device to disc.

Select the patch file corresponding to your serial interface from the following table:

INTERFACE	PATCH FILE
SIO-1	S1INP.HEX
SIO-2 or SIO-2B	S2INP.HEX
SIO-4 (any type)	S4INP.HEX

Make sure you have a copy of the patch file, DDT.COM and PIP.COM on your current system disc. Before proceeding check that you have a backup copy of PIP.COM on another disc in case of accidents.

Now combine PIP.COM and the selected patch by typing the following commands:

```
A>DDT PIP.COM [RETURN]          load PIP into memory with DDT
NEXT PC
1D00 0100
-IS4INP.HEX [RETURN]            or the name of whichever patch
                                you have chosen

-R [RETURN]                     read in patch

NEXT PC
1D00 0000

CTRL C [RETURN]

A>SAVE 28 PPIP.COM [RETURN]     save patched PIP with name
                                PPIP.COM
```

You should now have a patched copy of PIP called PPIP.COM on your system disc. Use the DIR command to check that this is so. Note that you need only carry out the patch procedure once. When you are satisfied that PPIP is working properly, you may wish to delete PIP.COM and rename PPIP.COM to PIP.COM to conserve disc space. This last step is, of course, optional.

3.4.2 Preparing for a Transfer

Before starting a transfer, it is necessary to initialise the interface you will be using. This is most easily done by selecting it for the printer output stream by means of the COS O command. This can be done from COS command level before bootstrapping CP/M, or afterwards from the Front Panel.

```
A>                          type CTRL F
!O                          type O for Option

PRINTER TYPE (0-4):1        or as desired
!K                          return to CP/M
```

When using the SIO-2 or SIO-4 interfaces, you will also need to enter the appropriate baud rate. Printer selection is dealt with in more detail in Section 3.5.1.

3.4.3 Reading In Paper Tape or Cards

Essentially all that is needed is to run PPIP, taking input from the new device INP:. Thus you type

```
A>PPIP RETURN

*PROG1.BAS=INP: RETURN      or any file name you choose
```

and start the paper tape or card reader. When the transfer is complete, PIP will reprompt with an asterisk. Type RETURN or CTRL C to restart CP/M. There are two constraints:

a. The last character of the input must be CTRL Z (1A hex). You must make sure that a CTRL Z character is included after the end of the input file. With paper tape this can be punched just like any other character, and with cards a special end-of-file card can be prepared for placing at the bottom of the deck. If you are using a Teletype to read in paper tape and have forgotten the CTRL Z, you can stop the tape when it gets to the end and enter CTRL Z from the keyboard.

b. The input data must fit into PIP's buffer in one chunk. This is because PIP pauses when memory fills up to write to disc; during the pause some characters passing through the tape or card reader will be lost. (The only exception to this is if you are using a Teletype fitted with automatic reader control and an SIO-2 interface with the reader control connected in which case the reader is stopped during the disc transfer.) You can gain a measure of control over the buffering by using block mode transfer. Please refer to the manual "An Introduction to CP/M Features and Facilities" - Digital Research for full details.

3.4.4 Transfer To and From Another Computer

For transferring ASCII data between two 380Z computers, all of the above description applies at the receiving end; at the sending end a serial interface must also be available, with its output pin connected to the receiving interface's input. At the sending end, the serial interface is again selected for printer output, but now PIP is used to output to the

LST: device (See Section 3.6), with the terminating CTRL Z sent from 'device' EOF:. Thus a typical command at the sending end might be:

***LST:=PROG1.BAS,EOF:** **RETURN**

with the serial stream received in the same way as described in Section 3 above. Note that the receiving end should be started before the sending one. When transferring data between the 380Z and another type of computer, it will be necessary to arrange that CTRL Z is the last character sent on output and is recognised as end-of-file on input. This type of simple link is of course restricted to ASCII data and will not work with binary (.COM) files in particular.

3.5 ADDING OUTPUT DEVICES

It is reasonably easy to use a wide variety of different printers with a 380Z disc system. The most important step in attaching a printer to the 380Z is to choose the interface (see Section 3.3) which matches the characteristics of your printer. You should obtain the characteristics of the printer which you intend to use from the supplier of the printer and match these against the characteristics of the 380Z interfaces. If you are uncertain as to which interface to use then ask your printer supplier to advise you. He will need a description of the 380Z interfaces given in Section 3.3 if he is to help you.

****

NOTES

1. The Serial interface printers which Research Machines provide can all be plugged into the SIO-4 interface fitted as standard on current disc systems, or the SIO-4C supplied as an option on cassette systems.

2. The Parallel interface printers that Research Machines provide can all be plugged into the User Input/Output port supplied as standard on all current disc systems, and as an option on current cassette or very early disc systems.

3. In order to support some printers, some interfaces supplied before 1st October 1980 will need to be rewired to match the pin connections of the current interfaces.

****

3.5.1 Testing a Printer

Plug in your printer and test it as follows: (Please note that these instructions apply to COS 3.4 et seq.)

Switch on the 380Z and the printer and make sure that the printer is ready for use (if in doubt, consult the instructions provided with the printer). Check that all connectors are plugged in properly.

In the bottom left-hand corner of the 380Z screen the COS prompt should have appeared. To initialise the printer type

O the Letter O for Options

Then type a digit from the following table corresponding to the type of interface that you wish to use.

1	SIO-1
2	SIO-2/2B formerly SIO-3
3	Parallel User I/O interface
4	SIO-4 or SIO-1 lineprinter option 4
5	SIO-5
6	SIO-6

For appropriate interfaces you will then be prompted for the baud rate. Choose a digit from the following table corresponding to the baud rate at which your printer will operate.

0	110 baud
1	300 baud
2	600 baud
3	1200 baud
4	2400 baud
5	4800 baud
6	9600 baud

If you are using a parallel printer the message

Attend to Printer!

may appear on the screen. It means the printer is not responding properly to the 380Z. Check that all appropriate switches are on, that the printer is 'on-line', that the paper is in, and that all plugs are firmly in place. If none of these helps, please refer to the instructions provided with the printer.

This process is called 'Selecting the Printer Option'. The purpose of this is to 'tell' the 380Z what kind of printer you are using.

Now type CTRL T. The computer will enter a small test routine that causes everything you type to be printed ONCE on the screen and ONCE to the lineprinter. The computer will not act on the characters you type, except to print them. Type a few characters and then press RETURN. The printer should print what you have typed. If it does not, try again a number of times. If that does not work, press the RESET switch on the 380Z, select the printer again, and try once more. If that does not work, check the matching of the printer to the interface, and the wiring of the interface cable. Remember that if you want to go onto another line on the printer you may have to follow the RETURN you type with the LINE FEED key.

To exit from test mode type CTRL F which will get you into the Front Panel.

****

NOTES

You can use the utility program CONFIG, which is supplied on the CP/M Disc Operating System issue disc, to create a version of CP/M which will automatically select the printer attached to the system each time you load CP/M (see Section 4.5).

A common cause of printers failing to work when connected is mis-wiring of the lead between the computer and the printer. If the lead you have is not one supplied by Research Machines specifically for your printer, please check all the connections carefully. If the lead was supplied by the printer supplier, please contact the printer supplier for advice.

****

3.5.2 Outputting Files

The PIP command can be used to output files to a printer by using the device name LST:. The procedure to print out a file NEWFILE.TXT is as follows:

- select the printer and restart CP/M

in response to the CP/M prompt

- type PIP RETURN

in response to the PIP prompt

- type LST:=NEWFILE.TXT RETURN

To print a small group of files, for example the files CHAP1.TXT, CHAP2.TXT and CHAP3.TXT, you can use the PIP file concatenation operation; just

- type **LST:=CHAP1.TXT, CHAP2.TXT, CHAP3.TXT RETURN**

3.5.3 Using a Printer with a Ready Line

Some serial interface printers can receive data faster than they can actually print it. For instance, a Qume Sprint 5 printer can receive data at 1200 baud, the equivalent of 120 characters per second. However, the Qume can only print at an average of 45 characters per second, so some way of stopping the computer from sending data is required. This signal, which originates at the printer, should go low (logic 0) when the printer is BUSY, i.e. when it wants the computer to suspend transmission.

The Research Machines interfaces types SIO-2, SIO-4 and 4C, SIO-5 and SIO-6 can all handle a READY line. The connection details are given in Section 3.6.7.

3.6 DETAILED HARDWARE INFORMATION

This section contains detailed information on various aspects of the following items of hardware:

- keyboard
- cassette recorder (40-character systems only)
- cables and connectors
- serial interfaces
- printed circuit boards

****

NOTE

The policy of Research Machines Limited is one of continuous development and improvement of its products. Consequently the hardware in the 380Z system which you have may differ in a number of minor ways from that described below.

****

3.6.1 Keyboard

Summary: 7 bit ASCII parallel output on D0-D6 optionally with 8th bit on D7 (COS 4.0 Only)
Parity not used
Positive logic

Data is clocked in to the 380Z on the positive going transition of the keyboard's strobe line. Thus it does not matter whether the keyboard produces a 'level strobe', a positive going pulse strobe or a negative going pulse strobe, so long as a low to high transition occurs (and only occurs once) at a time when the data is valid. The data lines go to the inputs of 2 74173 devices and must, therefore, be capable of driving at least one TTL load.

The strobe should be a low impedance TTL output i.e. should be able to source 400 uA and sink 16 mA.

Inside the computer there is a 1nF capacitor between the strobe line and ground to filter out high frequency noise. Therefore a 47 ohm resistor (or other value determined by experiment) should be placed in series with the strobe line at the keyboard end. Depending on the type of keyboard a reflection would cause each character to appear twice, or each character to appear and be immediately deleted, or the desired character and a spurious character to be generated.

Strobe Pulse Width: Subject to the above, minimum strobe pulse width should be 10 microseconds.

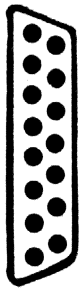
D7 is read by the computer, but is automatically set to zero by the keyboard input routine in the Monitor except in COS 4.0.

Connector: 15-way 'D Submin' male D plug.

3.6.2 Keyboard Cable Designations

15 way Male 'D' submin male D plug

Pin No.	Function	Colour
1	D0	Brown
9	D1	Red
2	D2	Orange
10	D3	Yellow
3	D4	Green
11	D5	Blue
4	D6	Violet
12	D7	Grey
5	Strobe	White
13	N.C.	
6	-12V	Pink
14	+5V	Turquoise
7	0V	Black
15	Beeper*	Yellow/Red
8	Ground	Screen
	Top Right **	Red/Blue
	Bottom Right **	Red/Green



The -12V and +5V power rails are supplied by the 380Z. Signal Ground and Chassis Ground should be kept separate.

* The 80-character VDU can drive a Beeper supported by COS 4. This is not normally provided, but a spare wire is taken down the keyboard cable and left unconnected within the keyboard unit for use with a small high impedance loudspeaker.

** The two blank keys at the top and bottom right of the keyboard produce two logic level outputs that are available separately. Two wires carrying these signals are left unconnected within the keyboard cable plug shell.

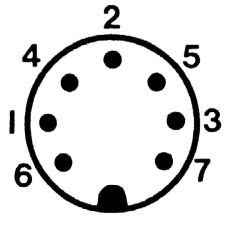
VDU Board 26-way Connector

Pin No.	Wire Colour	40 char Function	80 char Function
1	Brown	D0	D0
2	Red	D1	D1
3	Orange	D2	D2
4	Yellow	D3	D3
5	Green	D4	D4
6	Blue	D5	D5
7	Violet	D6	D6
8	Grey	D7	D7
9	White	Strobe	Strobe
10	Black	N.C.	N.C.
11	Brown	-12V	-12V
12	Red	+5V	+5V
13	Orange	0V	0V
14	Yellow (cut)	+12V	+12V
15	Green (cut)	N.C.	N.C.
16	Blue (cut)	N.C.	N.C.
17	Violet (cut)	+12V	+12V
18	Grey	Reset button	Reset button
19	White	0V	0V
20	Brown	Relay 2	0V
21	Brown	+5V	+5V
22	Red	+5V	Frame Sync
23	Orange	Relay 1	Line Sync
24	Yellow	0V	Video Earth
25	Green	Cassette Out	Video Signal
26	Blue	Cassette In	Beeper

3.6.3 Cassette Interface

On 40-character systems a cassette recorder or a cassette motor control unit can be connected to the 380Z computer via the 7-pin DIN socket at the rear of the cabinet, near the keyboard socket. 80-character systems do not have a cassette interface.

Socket connections (Viewed from rear of plug)



Pin 6	Signal to computer
Pin 1	Signal from computer
Pin 4	ground
Pin 2	recorder 1 motor control
Pin 5	no connection
Pin 3	+5V
Pin 7	recorder 2 motor control

Description of Signals

1. The replay signal from the recorder (pin 6) should be taken from the recorder DIN connector or external loudspeaker socket. A signal of about 1.5V RMS is required but this is not very critical. The input lead from the recorder to the computer should be screened.

2. The record signal from the computer (pin 1) is at low level, suitable for the cassette recorder microphone input (although it will normally go to the DIN connector along with the replay signal). Connect a 47 nF capacitor between the microphone input and ground at the cassette recorder end to reduce digital noise. The output lead connecting the computer to the recorder should be screened.

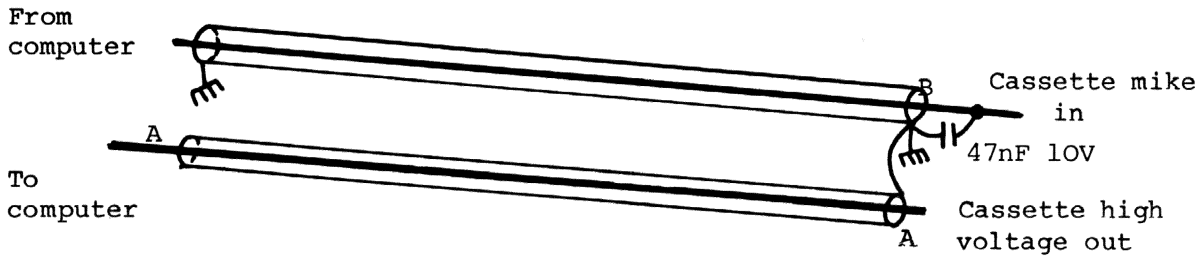
3. The motor control signals (pins 2 and 7) are TTL outputs which are logic 0 for motor on, logic 1 for motor off. In a two recorder set-up, pin 2 controls the recorder normally used for reading.

4. Only a few milliamps (e.g. to drive a LED indicator) should be drawn from the +5V outlet (pin 3).

3.6.4 Cable Connections

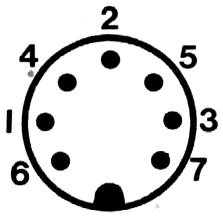
Making a connecting cable for Hitachi TRQ-299R Cassette Recorders

Materials: 7 way DIN plug (e.g. RS part no. 478-037), 5 way 180° DIN plug (e.g. RS part no. 477-876), twin screened cable (e.g. RS part no. 367-189, 25m.), 47nF 10V ceramic disc capacitor (or 47nF 12V or 25V; 47nF; e.g. RS part no. 124-162, pack of 5)



AT COMPUTER END

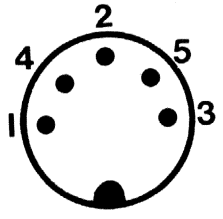
7 way DIN plug (viewed from rear of plug)



1. Connect screens together
2. Connect wire A (signal to computer) to pin 6
3. Connect wire B (signal from computer) to pin 1
4. Connect screen of both wires to pin 4; this connection must be isolated, i.e. insulated so that the screens are not connected to the case of the DIN plug.

AT CASSETTE RECORDER END

5 way 180° DIN plug (viewed from rear of plug)



1. Connect wire A (cassette high voltage out) to pin 3.
2. Connect wire B (cassette input) to pin 1
3. Connect a 47nF capacitor across pins 2 and 4
4. Connect screens of both wires to pin 2.

Pins 1 and 4 of the TRQ299R 5 way DIN plug are conncted together so the positioning of the 47nF is consistent with the specification provided. Leads for connecting other tape recorders to the 380Z should be constructed in a similar manner.

3.6.5 50 way CONNECTOR and Z50 BUS

Function	Wire and Pin no.	Wire and Pin no.	Function	50 way BUS Cable	
BRESET	1	2	Reserved	1	BRESET
				2	Reserved
BDO	3	4	BD1	3	BDO
				4	BD1
BD2	5	6	BD3	5	BD2
				6	BD3
BD4	7	8	BD5	7	BD4
				8	BD5
BD6	9	10	BD7	9	BD6
				10	BD7
BA0	11	12	BA1	11	BA0
				12	BA1
BA2	12	14	BA3	13	BA2
				14	BA3
BA4	15	16	BA5	15	BA4
				16	BA5
BA6	17	18	BA7	17	BA6
				18	BA7
BA8	19	20	BA9	19	BA8
				20	BA9
BA10	21	22	BA11	21	BA10
				22	BA11
BA12	23	24	BA13	23	BA12
				24	BA13
BA14	25	26	BA15	25	BA14
				26	BA15
BUSRQ -	27	28	BWAIT -	27	BUSRQ -
				28	BWAIT -
BUSAK -	29	30	BINT -	29	BUSAK -
				30	BINT -
BNMI -	31	32	BHALT -	31	BNMI -
				32	BHALT -
BIORQ -	33	34	BWR -	33	BIORQ -
				34	BWR -
BRD -	35	36	BMREQ -	35	BRD -
				36	BMREQ -
BRFSH -	37	38	BM1 -	37	BRFSH -
				38	BM1 -
OV	39	40	BCLK	39	OV
				40	BCLK
+5V	41	42	+5V	41	+5V
				42	+5V
+5V	43	44	+12V	43	+5V
				44	+12V
+5V	45	46	-12V	45	+5V
				46	-12V
OV	47	48	OV	47	OV
				48	OV
OV	49	50	OV	49	OV
				50	OV

The bus is TTL buffered.

3.6.6 Serial Interfaces - connection information

Connecting to a Device with a Ready Line

Buffered printers and some other devices have an output which must be used to tell the 380Z that the receiving device is ready to receive data. For these devices connection must be made to wire 5 on the 10-way cable which connects the SIO-4 to the rear panel sockets. This wire connects to pin 4 of the 25-way 'D' plug marked 'S4', sited on the rear panel of the 380Z.

It is likely that the peripheral will not use pin 4 for its busy signal as there are a number of ways of interpreting RS232 functions. If this is the case, you will have to make a suitable adaptor, or rewire the plug with reference to your printer manual.

Advanced Use of the SIO-4:

The SIO-4 uses an 8251 universal asynchronous receiver/transmitter. There are several inputs and outputs to the 8251 which are buffered and brought out to the 10-way cable. Here is the wiring list for the 10-way cable. Wire one (pin one on the 10 way connector) is the wire nearest the outer edge of the interface board. Function names refer to the relevant 8251 pin.

		Colours
Wire 1:	TxD' Transmitted data out	Brown
Wire 2:	signal ground	Red
Wire 3:	RxD' received data in	Orange
Wire 4:	direct digital ground	Yellow
Wire 5:	DSR data set ready input	Green
Wire 6:	spare	Blue
Wire 7:	DTR data terminal ready output	Violet
Wire 8:	RTS ready to send ouput	Grey
Wire 9:	minus twelve volts	White
Wire 10:	CTS Clear to send output	Black

' Indicates signal that is active low

The inputs to the 8251 integrated circuit (IC) are buffered and inverted by a 75189 IC, the outputs from the 8251 are inverted and buffered by an 75188 IC.

Handler Programming Notes

The device is accessed via Z80 input/output instructions. The base address of the set of ports used varies. On an SIO-4 supplied in conjunction with an FDS system the base address "F SIO" is E8, on an MDS system it is C8. On an SIO-4C supplied in a cassette system it is C8. On a disc machine the base address will be whichever of the above two has not been used by the SIO-4. A machine may be equipped with three serial interfaces of this general type, named SIO-4 (standard), SIO-5 and SIO-6 - each having a different port base address. The port base address of an SIO-6 is 48H.

SIO-4 Connections

'D' submin 25-Way Female Pin No.	Colour
1	NC
2	ORANGE
3	BROWN
4	GREEN
5	NC
6	NC
7	RED
8	NC
9-25	NOT CONNECTED

NC - No Connection

The SIO-4 has been wired in three ways during its life. Pins 2,3 and 7 have never altered, these are Data In, Data Out and Ground respectively.

Early SIO-4 leads had only pins 2,3 and 7 wired. The next wiring method included GREEN to pin 4 in addition to the data and ground, this was to provide a Busy line, and matches that wired on the QUME sprint 5 daisywheel printer. For a brief period pins 2,3,4,5,6,7,8 and 20 were wired. The standard now is for only one wire in addition to pins 2,3 and 7 being connected, this is pin 4 - the BUSY line. This connection method matches that used on the Microline 80 printer currently sold by Research Machines.

3.6.7 Serial Interfaces

SIO-1B

This interface is a 20 mA current loop type, separate documentation accompanies this interface.

SIO-2 and SIO-2B (formerly SIO-3)

The SIO-2 is an RS232 (V24) standard interface. It communicates with the processor internally via the user I/O port and is controlled by software built into the COS monitor ROM. The maximum baud rate at which these interfaces will operate is 2400 baud. The interface is fitted with a 25 way 'D' submin connector. The wiring is as follows:

Signal into computer	Pin 2
Signal out from computer	Pin 3
Signal ground	Pin 7
Busy signal input	Pin 4
(may be left unconnected)	

There are two unused RS232 V24 level input signals which can be read by the processor and three unused RS232 V24 level output signals which can be controlled by the processor. These are available to users who require extra input and output control signals.

The SIO-2B (formerly SIO-3) is the 20 mA current loop version of the SIO-2 type interface. The SIO-2B is similar in most respects to the SIO-2 except that the signal levels and connector wiring are different.

SIO-2B. Input/Output level: 20 mA current loop.

Data output from computer current loop positive:	Pin 1
Data output from computer current loop negative:	Pin 2
Data input to computer current loop positive:	Pin 3
Data input to computer current loop negative:	Pin 4
Reader enable current loop positive:	Pin 5
Reader enable current loop negative:	Pin 6

SIO-4 and SIO-4C

The SIO-4 interface is now fitted as standard on all current disc-based systems. The SIO-4 interface is connected to a 25-way sub-min 'D' type connector. The wiring to this connector are:

Signal into computer	Pin 2
Signal out from computer	Pin 3
Signal ground	Pin 7
Busy signal	Pin 4

The interface uses an 8251 universal asynchronous receiver/transmitter (UART) and there are several inputs and outputs to this device which are buffered and brought out to a 10-way cable. The following control signals can be supported:

DSR	Data set ready.	(input)
DTR	Data terminal ready.	(input)
RTS	Ready to send.	(output)
CTS	Clear to send.	(output)

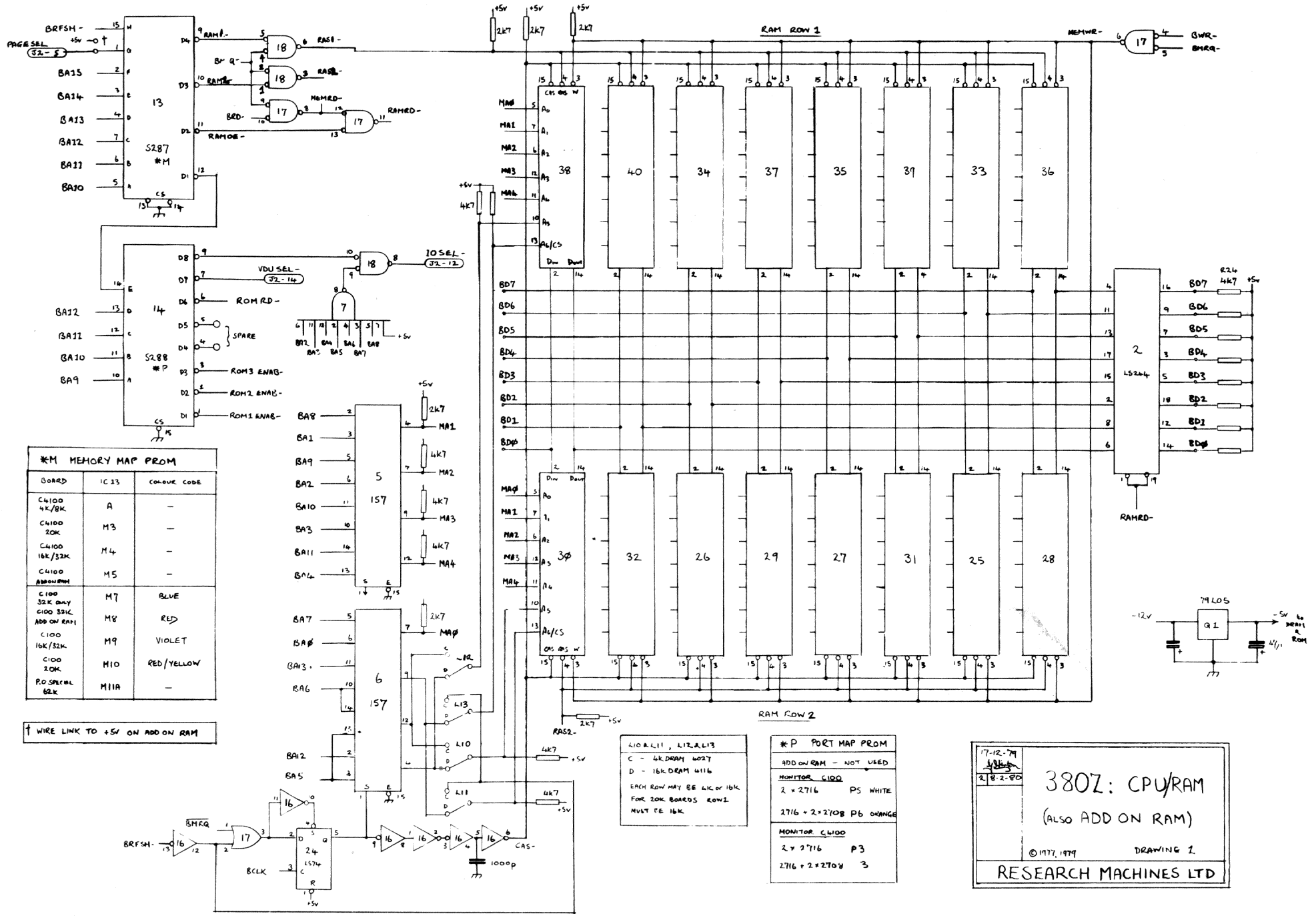
This interface operates at rates of 110 to 9600 baud.

The SIO-4C is the cassette version of this interface.

3.6.8 Printed Circuit Boards

The following data sheets provide the component layout and the overall circuit diagrams for

- the CPU board
- the 40 character VDU board
- the 80 character VDU board
- the FDC (floppy disc controller) board
- the HRG (high resolution graphics) board.



*M MEMORY MAP PROM		
BOARD	IC 13	COLOUR CODE
C4100 4K/8K	A	–
C4100 20K	M3	–
C4100 16K/32K	M4	–
C4100 ADD ON RAM	M5	–
C100 32K ONLY	M7	BLUE
C100 32K ADD ON RAM	M8	RED
C100 16K/32K	M9	VIOLET
C100 20K	M10	RED/YELLOW
P.O SPECIAL 62K	M11A	–

† WIRE LINK TO +5V ON ADD ON RAM

L10 & L11 , L12 & L13
C - 4k DRAM 4027
D - 16k DRAM 4116
EACH ROW MAY BE 4K or 16K
FOR 20K BOARDS ROW1
MUST BE 16K

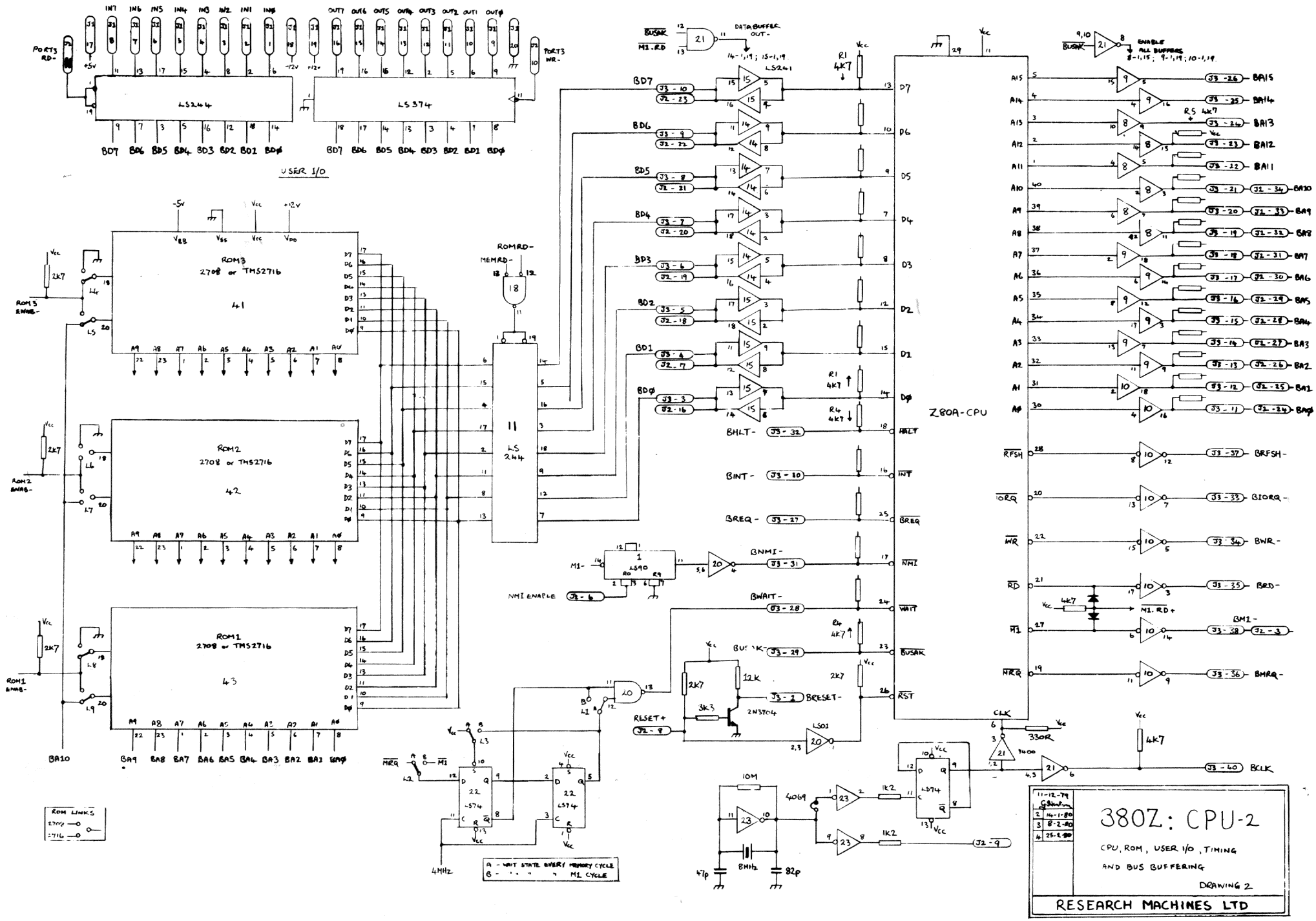
*P PORT MAP PROM	
ADD ON RAM – NOT USED	
MONITOR C100	
2 × 2716	P5 WHITE
2716 + 2 × 2708	P6 ORANGE
MONITOR C4100	
2 × 2716	P3
2716 + 2 × 2708	3

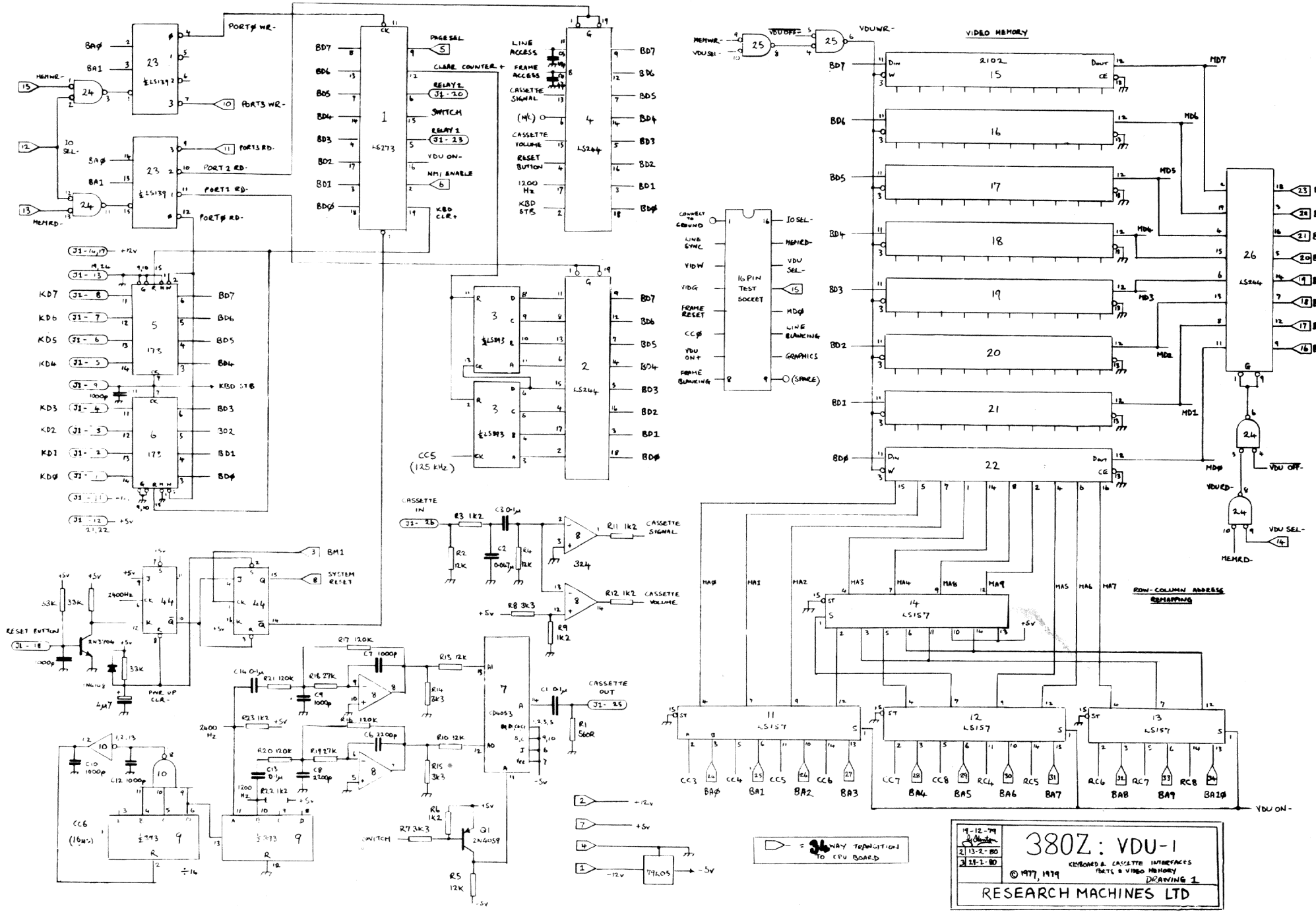
17-12-79
2.8-2-80

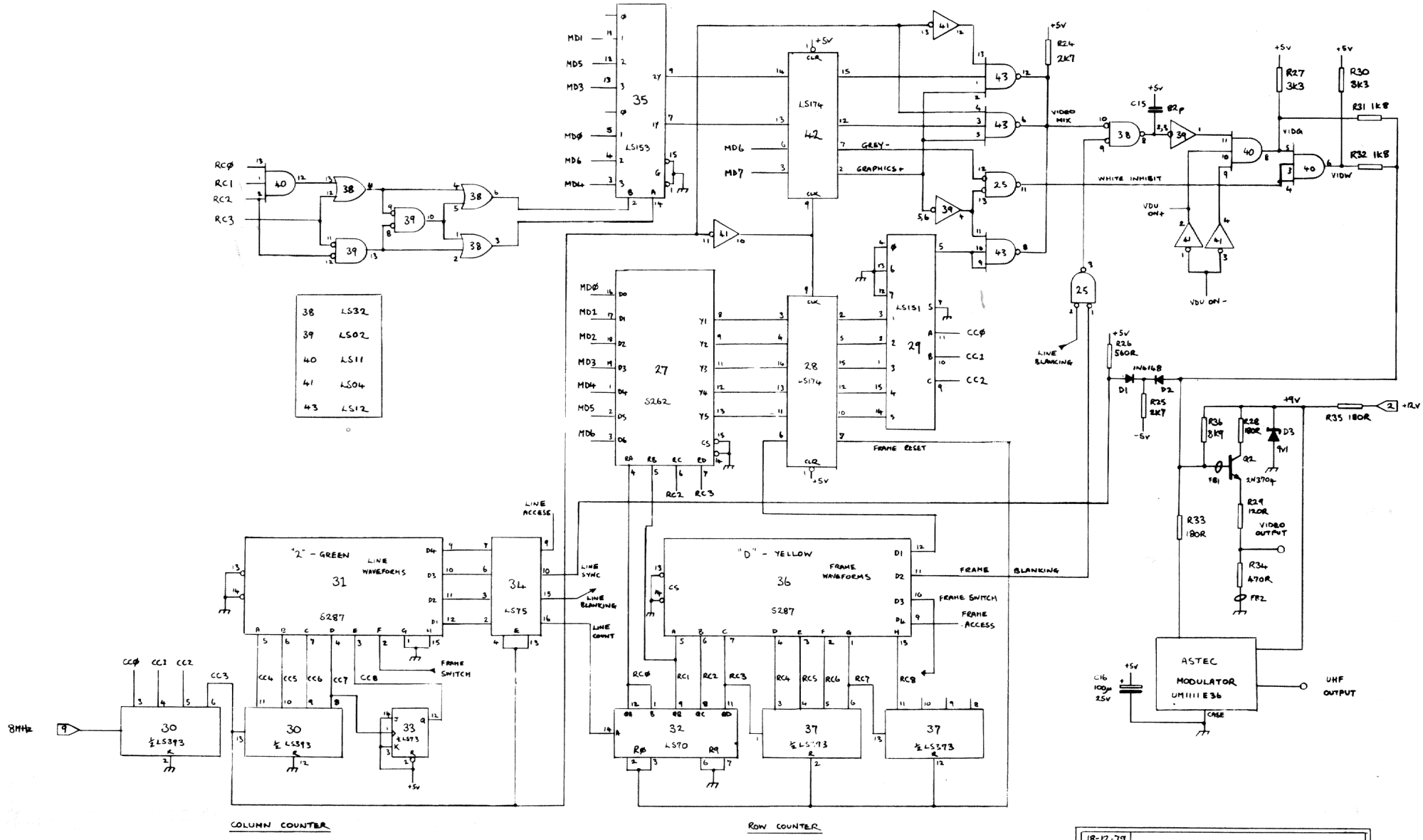
380Z: CPU/RAM
(ALSO ADD ON RAM)

DRAWING 1

RESEARCH MACHINES LTD

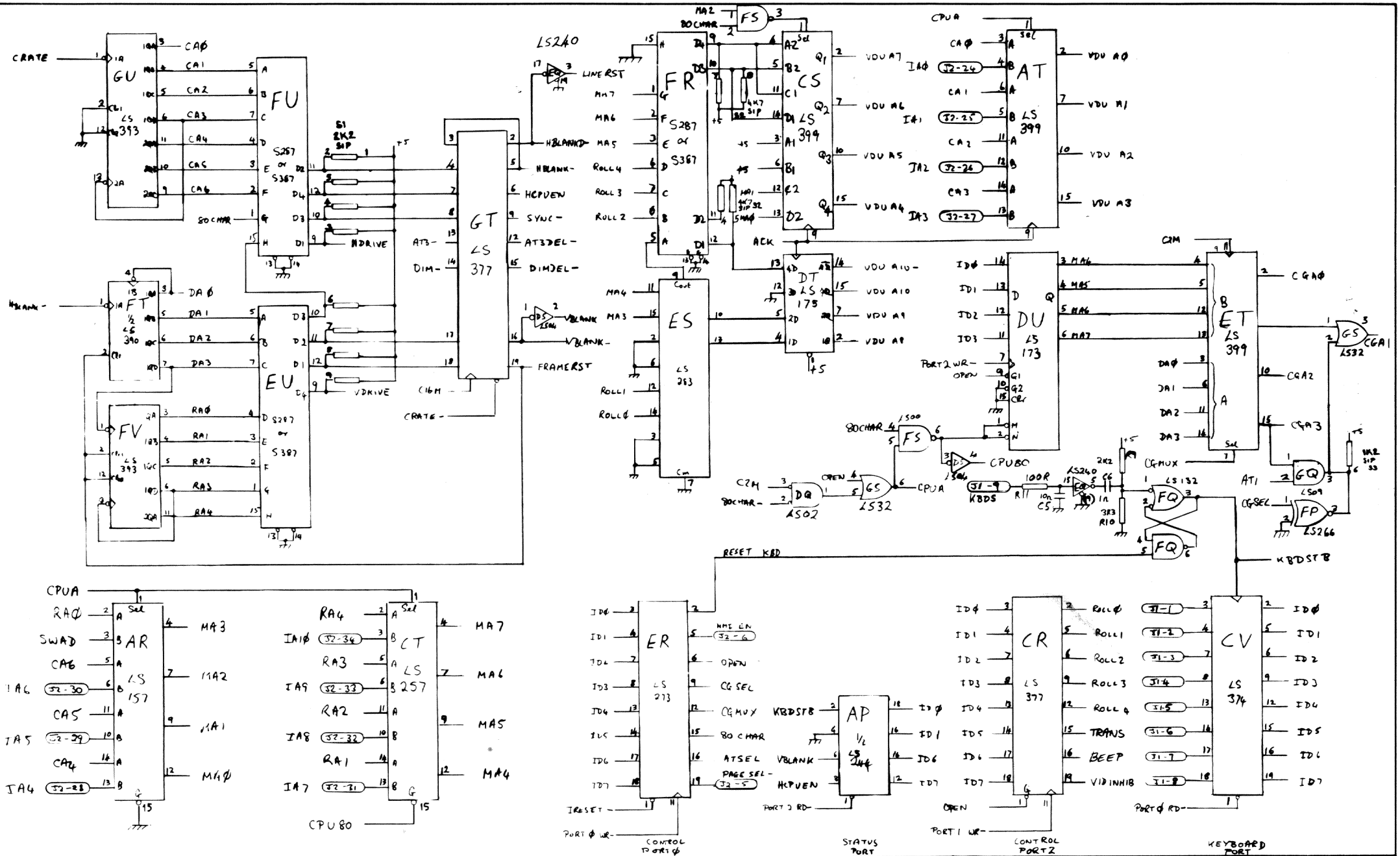

USER I/O
PORT3 RD-
PORT3 WR-
LS244
LS374
BD7 BD6 BD5 BD4 BD3 BD2 BD1 BD0
ROM3 2708 or TMS2716 41
ROM2 2708 or TMS2716 42
ROM1 2708 or TMS2716 43
ROMRD-
MEMRD-
LS 244
Z80A-CPU
DATA BUFFER OUT-
BUSAK-
M1.RD
BUSRQ-
BINT-
BNMI-
BWAIT-
BRESET-
RESET+
NMI ENABLE
BHALT-
BREQ-
BRFSH-
BIORQ-
BWR-
BRD-
BM1-
BMRQ-
BCLK
ENABLE ALL BUFFERS
4MHz
A - WAIT STATE EVERY MEMORY CYCLE
B - " " " M1 CYCLE
ROM LINKS
380Z: CPU-2
CPU, ROM, USER I/O, TIMING AND BUS BUFFERING
DRAWING 2
RESEARCH MACHINES LTD


VIDEO MEMORY
2102
15
16
17
18
19
20
21
22
26
LS244
LS157
11
12
13
14
PORT0 WR-
PORT3 WR-
PORT2 RD-
PORT1 RD-
PORT0 RD-
PAGESEL
CLEAR COUNTER+
RELAY2
SWITCH
RELAY1
VDU ON-
NMI ENABLE
KBD CLR+
LINE ACCESS
FRAME ACCESS
CASSETTE SIGNAL
CASSETTE VOLUME
RESET BUTTON
12.00 Hz
KBD STB
16 PIN TEST SOCKET
CONNECT TO GROUND
LINE SYNC
VIDW
VIDG
FRAME RESET
CCØ
VDU ONT
FRAME BLANKING
IOSEL-
MEMRD-
VDU SEL-
MDØ
LINE BLANKING
GRAPHICS
(SPACE)
CC5 (125 kHz)
CASSETTE IN
CASSETTE SIGNAL
CASSETTE VOLUME
CASSETTE OUT
SYSTEM RESET
POWER UP CLR-
RESET BUTTON
SWITCH
2400 Hz
1200 Hz
CC6 (16 Hz)
÷16
VDUWR-
VDUOFF-
VDUSEL-
MEMWR-
VDU OFF-
VDURD-
ROW-COLUMN ADDRESS REMAPPING
= 36 WAY TRANSITION TO CPU BOARD
380Z : VDU-1
KEYBOARD & CASSETTE INTERFACES
© 1977, 1979
DRAWING 1
RESEARCH MACHINES LTD



3.32

18-12-79	380Z: VDU-2	
2 13-2-80	VIDEO GENERATION	
	© 1977, 1979	DRAWING 2
	RESEARCH MACHINES LTD	



MODIFICATION			
A	22/2/80		
B	23/7/80		
C	11/8/80	12/8/80	
D	11/8/81	31/7/81	AW
G	11/8/81	AW	
H	24/10/81	AW	

Research Machines Ltd

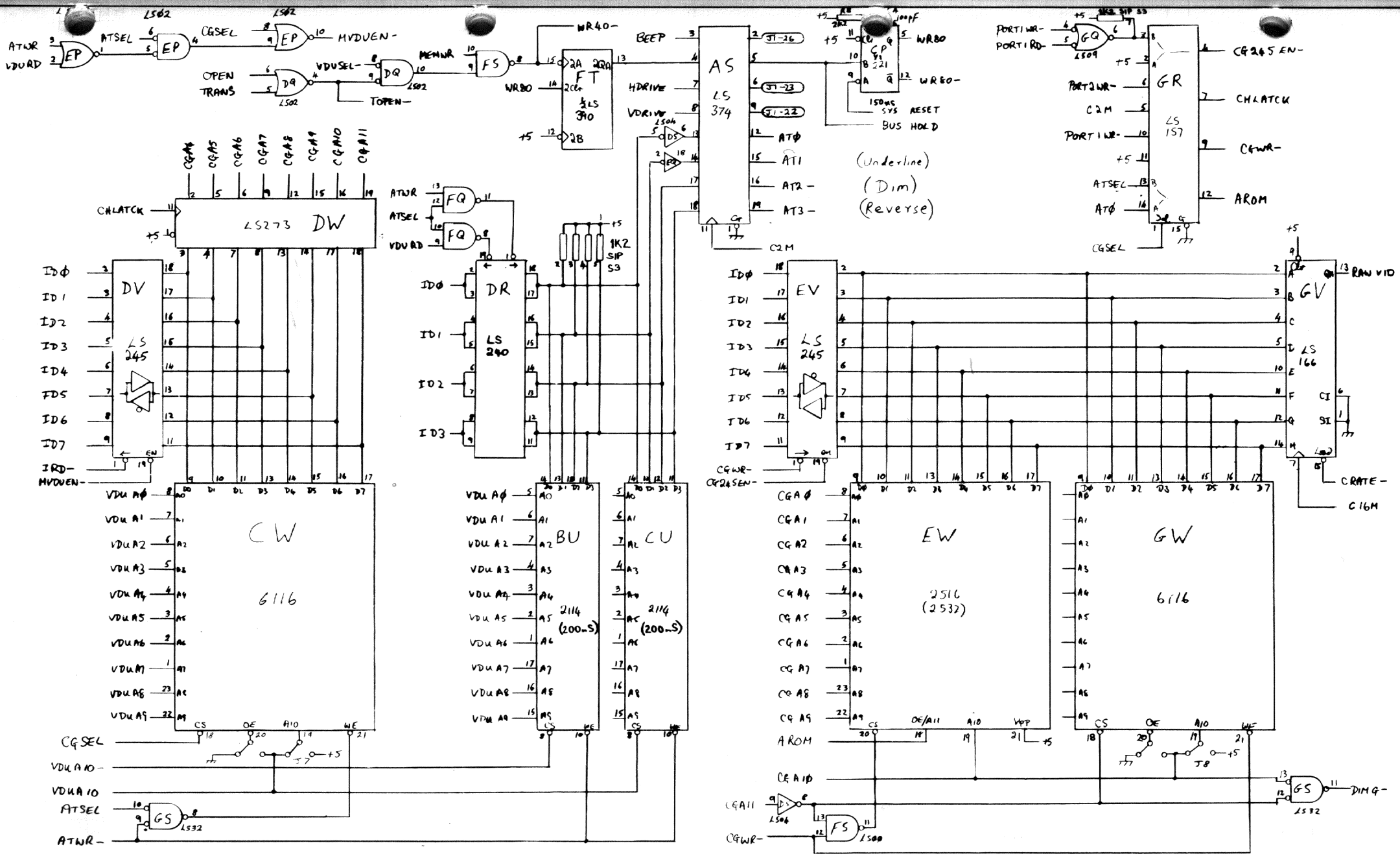
DATE	DRAWN BY	CHECKED BY
22-2-80	TBrisbourne	

TITLE: 80/40 VDU TIMING / PORTS

DRAWING NO. VDU 8[illegible]

3.33

WP 1001



MODIFICATION		
A	23/7/80	
B	11/8/80	
C	12/8/80	
D	14/5/81	
E	31/7/81	AW
F	11/8/81	AW
H	26/10/81	AW

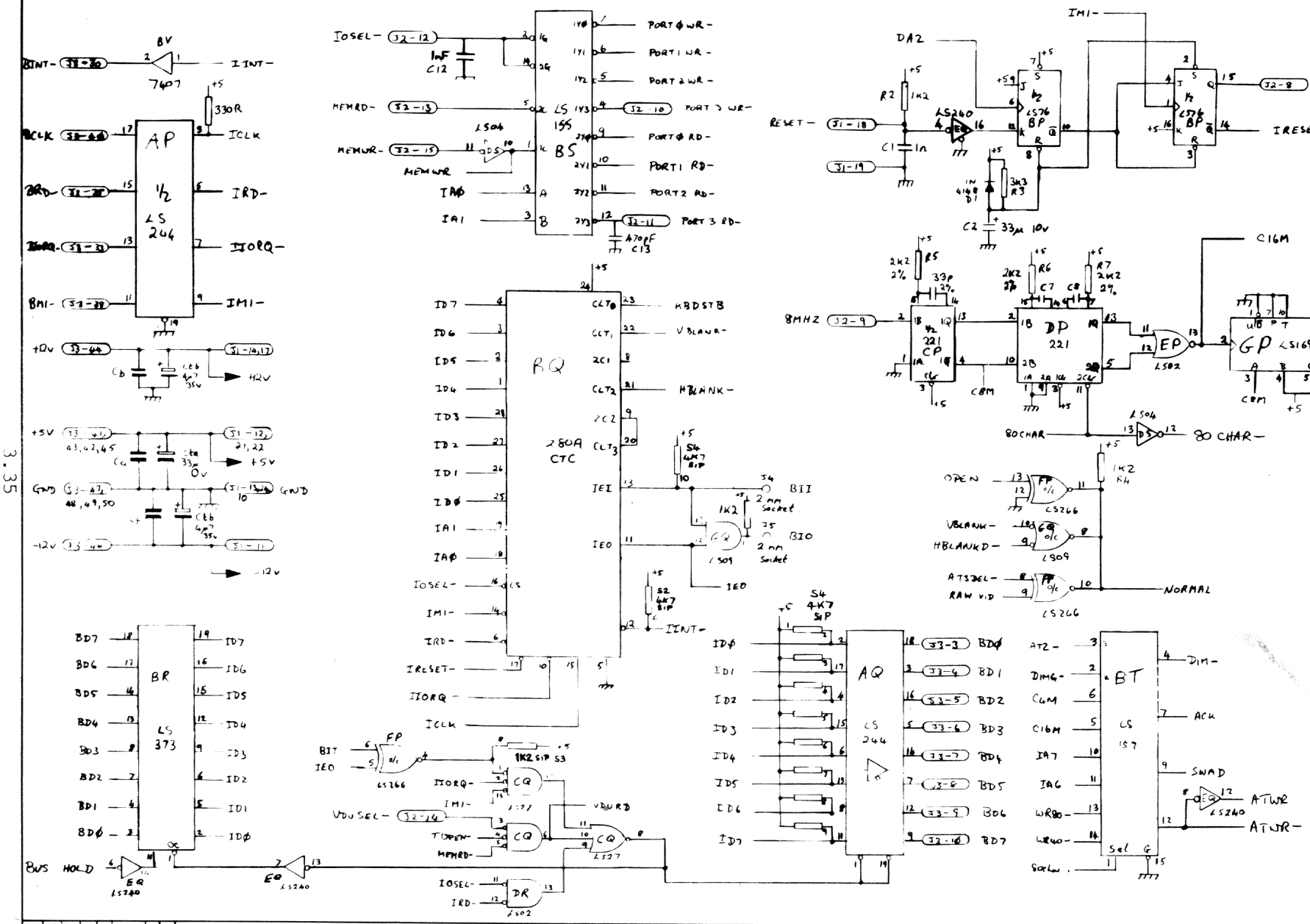
Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
23/7/80	T.B.	

TITLE: 80/40 VDU MEMORY

DRAWING NO.: VDU80-2

3.34



3.35

MODIFICATION			
A	14/2/80		
B	27/5/80		
C	1/8/80		
D	12/8/80	31/5/81	AW
E		31/7/81	AW
G		11/8/81	AW
H		21/8/81	AW

Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
14-2-80	T.B.R.L.-Jm	

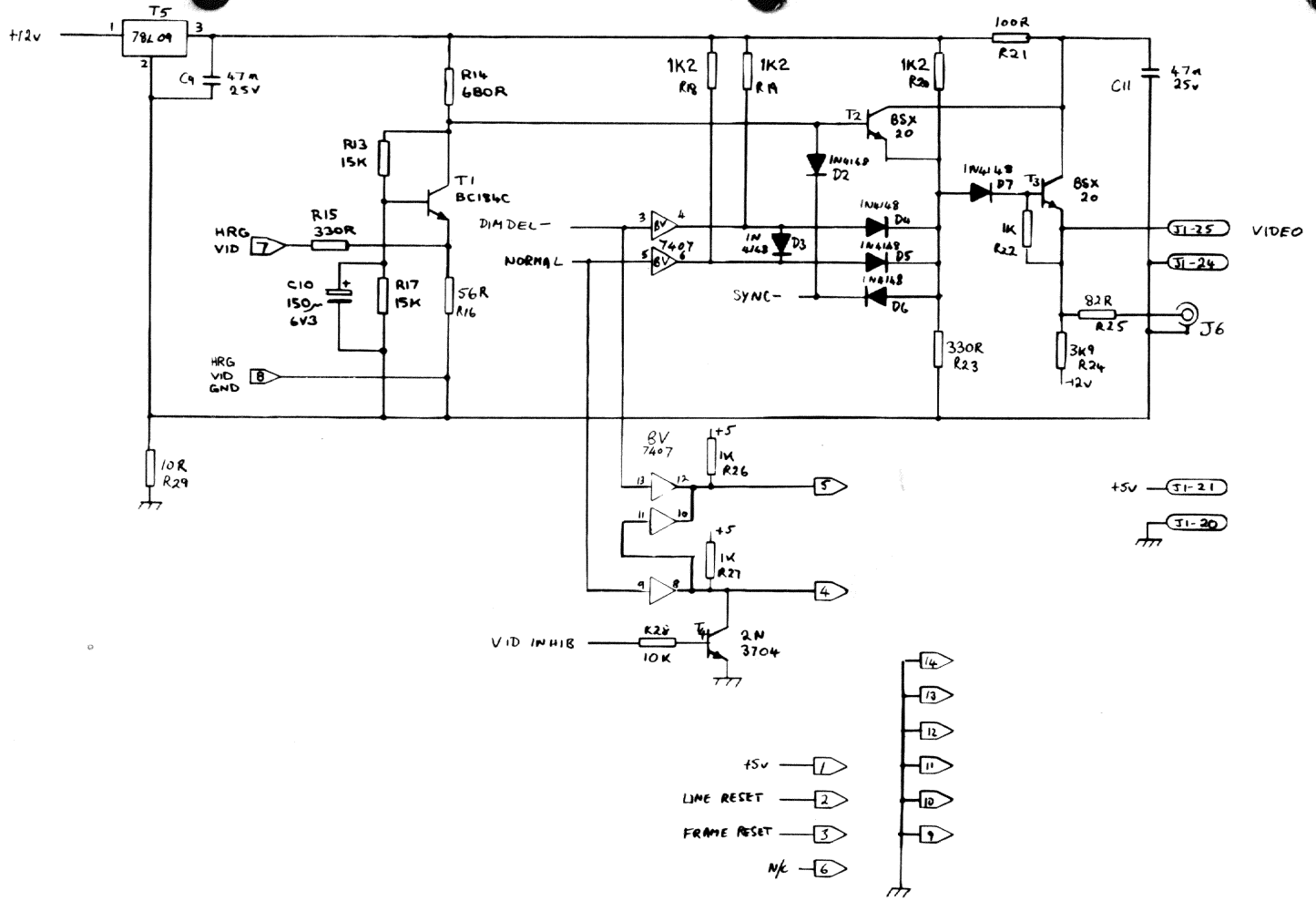
TITLE

80/40 VDU BUS BUFFERS

DRAWING NO.

VDU8

WP 1001



NOTES

1. ⇨ INDICATES CONNECTION TO 16-WAY DIL SOCKET AY/J7
2. T1 BC184C MIN. GAIN 500

MODIFICATION			
A	20/6/80		
C	31/7/81	AW	
D	31/7/81	AW	
E	20/8/81	AW	
H	26/10/81	AW	

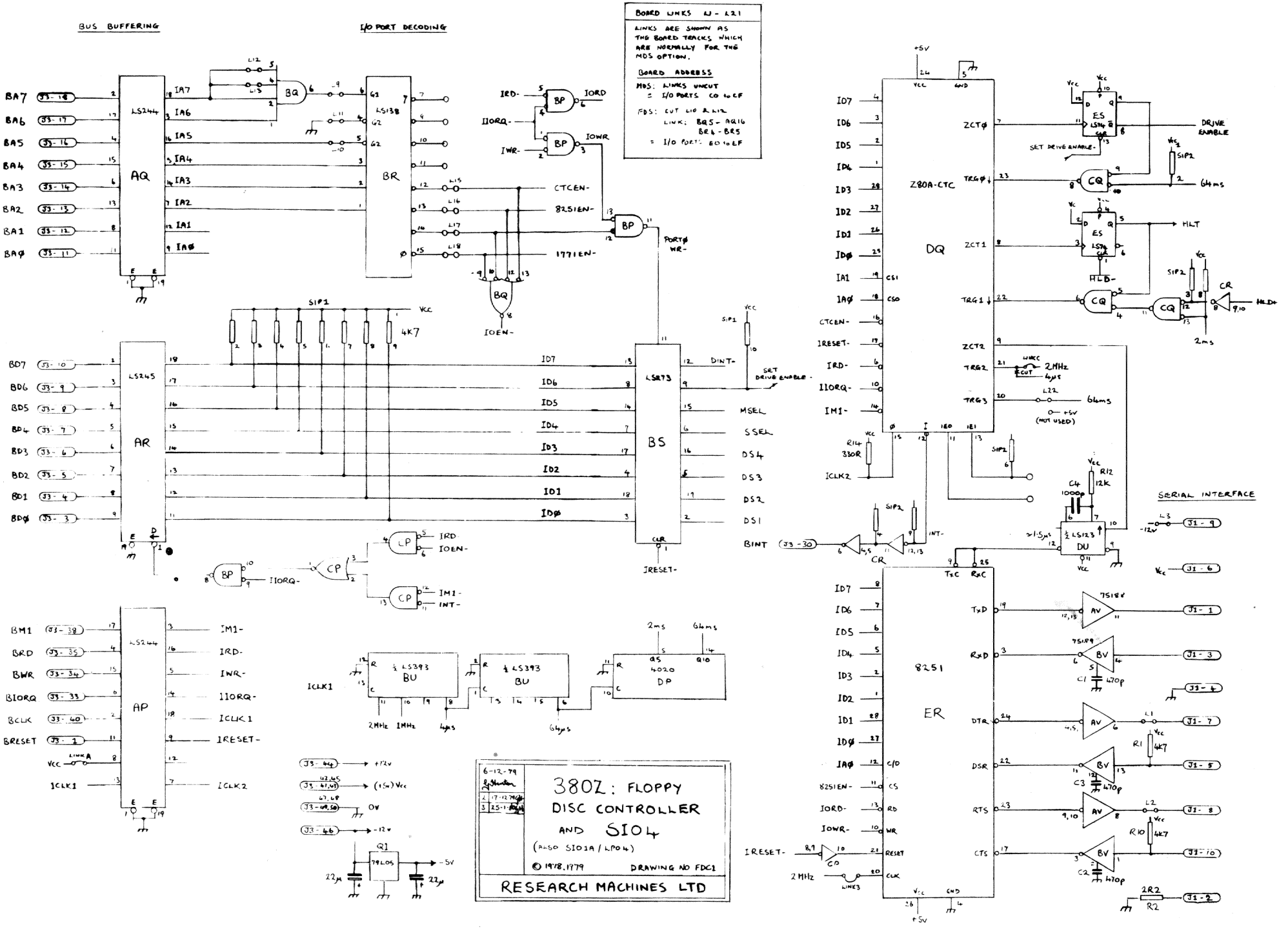
Research Machines Ltd

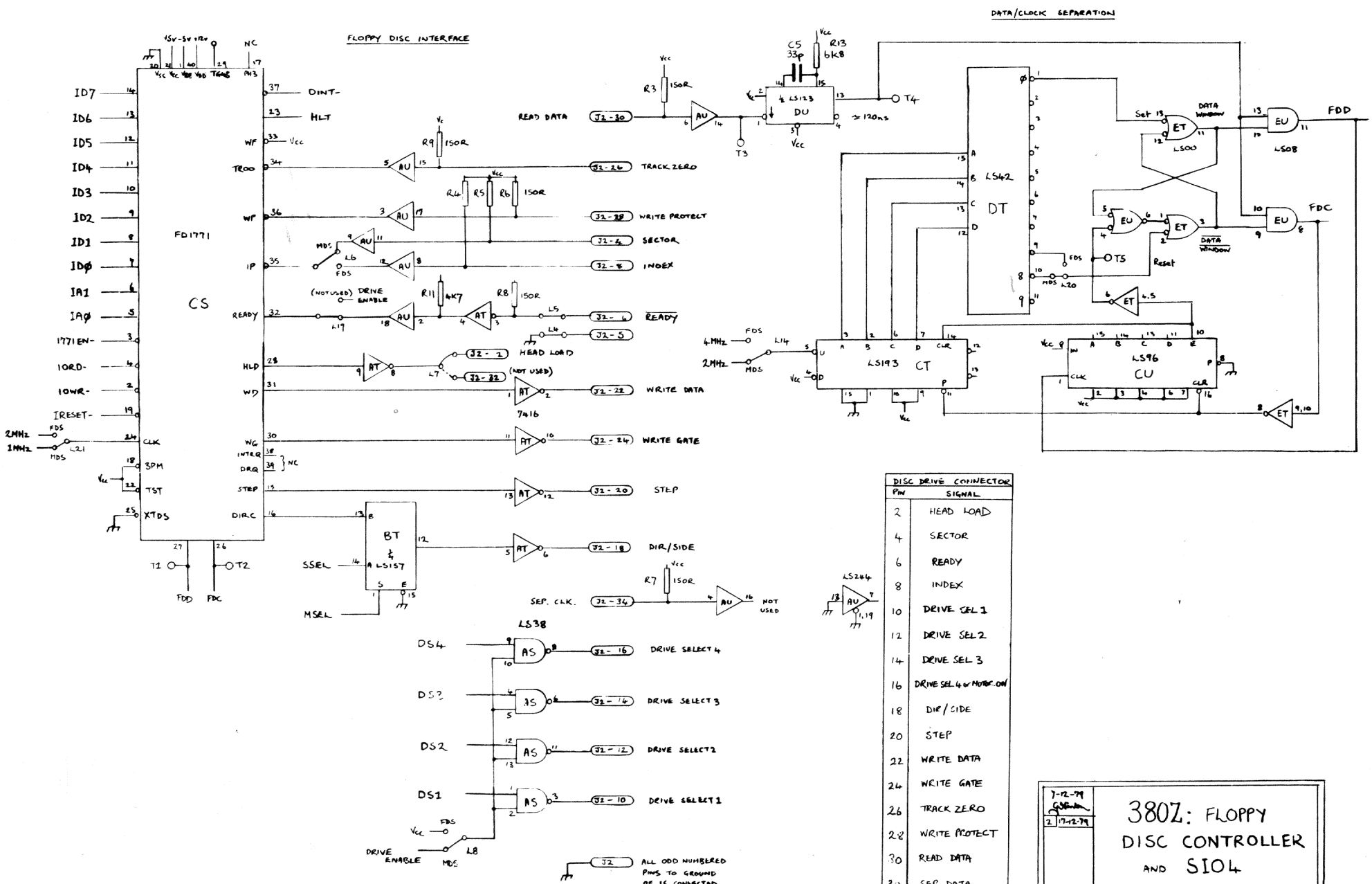
DATE	DRAWN BY	CHECKED BY
20/6/80	T.B Rhodes	

TITLE: 80/40 VDU OUTPUT

DRAWING NO. VDU80 -4

WP 1001


BUS BUFFERING
I/O PORT DECODING
BOARD LINKS L1 - L21
LINKS ARE SHOWN AS THE BOARD TRACKS WHICH ARE NORMALLY FOR THE MDS OPTION.
BOARD ADDRESS
MDS: LINKS UNCUT = I/O PORTS C0 to CF
FDS: CUT L10 & L12
LINK: BQ5 - AQ16
BR4 - BR5
= I/O PORTS E0 to EF
BA7 BA6 BA5 BA4 BA3 BA2 BA1 BA0
LS244 AQ
LS138 BR
BQ
IA7 IA6 IA5 IA4 IA3 IA2 IA1 IA0
IRD- IIORQ- IWR-
BP
IORD IOWR
CTCEN- 8251EN- 1771EN-
PORT0 WR-
IOEN-
SIP1 4K7 Vcc
BD7 BD6 BD5 BD4 BD3 BD2 BD1 BD0
LS245 AR
ID7 ID6 ID5 ID4 ID3 ID2 ID1 ID0
LS373 BS
DINT-
SET DRIVE ENABLE-
MSEL SSEL DS4 DS3 DS2 DS1
BINT
IRESET-
CP
IRD IOEN- IM1- INT-
BM1 BRD BWR BIORQ BCLK BRESET
LS244 AP
IM1- IRD- IWR- IIORQ- ICLK1 IRESET-
LINK A
Vcc
ICLK1
ICLK2
ICLK
2MHz 1MHz 4μs 64μs
½ LS393 BU
4020 DP
2ms 64ms
Q5 Q10
J3-44 +12v
J3-42,45 J3-41,43 (+5v) Vcc
J3-47,48 J3-49,50 0V
J3-46 -12v
Q1 79L05 -5v
22μ 22μ
6-12-79
3 25-1-79
380Z: FLOPPY DISC CONTROLLER AND SIO4
(ALSO SIO1A / LP04)
© 1978,1979 DRAWING NO FDC1
RESEARCH MACHINES LTD
+5V
Z80A-CTC DQ
ID7 ID6 ID5 ID4 ID3 ID2 ID1 ID0 IA1 IA0 CTCEN- IRESET- IRD- IIORQ- IM1-
CS1 CS0
ZCT0 ZCT1 ZCT2
TRG0 TRG1 TRG2 TRG3
ZC/TO0 ZC/TO1 ZC/TO2
2MHz 4μs
64ms
+5V (NOT USED)
ES LS74
DRIVE ENABLE
SET DRIVE ENABLE-
64ms
MLT
HLD-
HLD+
2ms
R14 330R
ICLK2
C4 1000p
R12 12k
½ LS123 DU
≈1.5μs
SERIAL INTERFACE
-12v
8251 ER
TxC RxC
TxD RxD DTR DSR RTS CTS
75188
75189
AV BV
C1 470p
C3 470p
C2 470p
R1 4K7
R10 4K7
2R2 R2
J1-1 J1-3 J1-4 J1-5 J1-6 J1-7 J1-8 J1-10 J1-2
C/D CS RD WR RESET CLK
8251EN- IORD- IOWR- IRESET-
2MHz
LINK3
C0



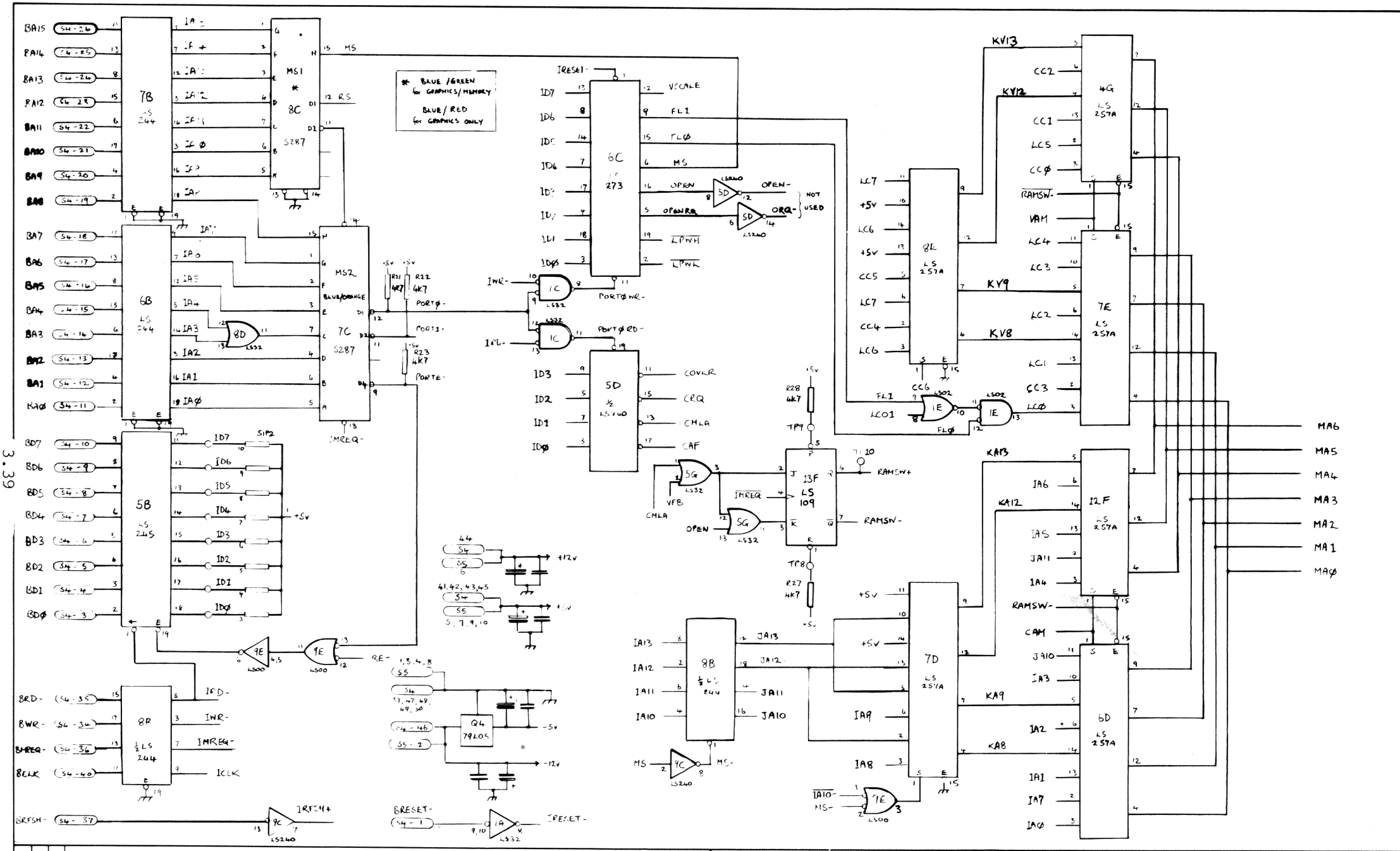
DISC DRIVE CONNECTOR	
PIN	SIGNAL
2	HEAD LOAD
4	SECTOR
6	READY
8	INDEX
10	DRIVE SEL 1
12	DRIVE SEL 2
14	DRIVE SEL 3
16	DRIVE SEL 4 or MOTOR ON
18	DIR/SIDE
20	STEP
22	WRITE DATA
24	WRITE GATE
26	TRACK ZERO
28	WRITE PROTECT
30	READ DATA
32	SEP. DATA
34	SEP CLOCK

7-12-79
2 17-12-79

380Z: FLOPPY DISC CONTROLLER AND SIO4

 DRAWING No FDC2

RESEARCH MACHINES LTD



3.39

MODIFICATION		
A	5-2-80	
B	27-3-80	

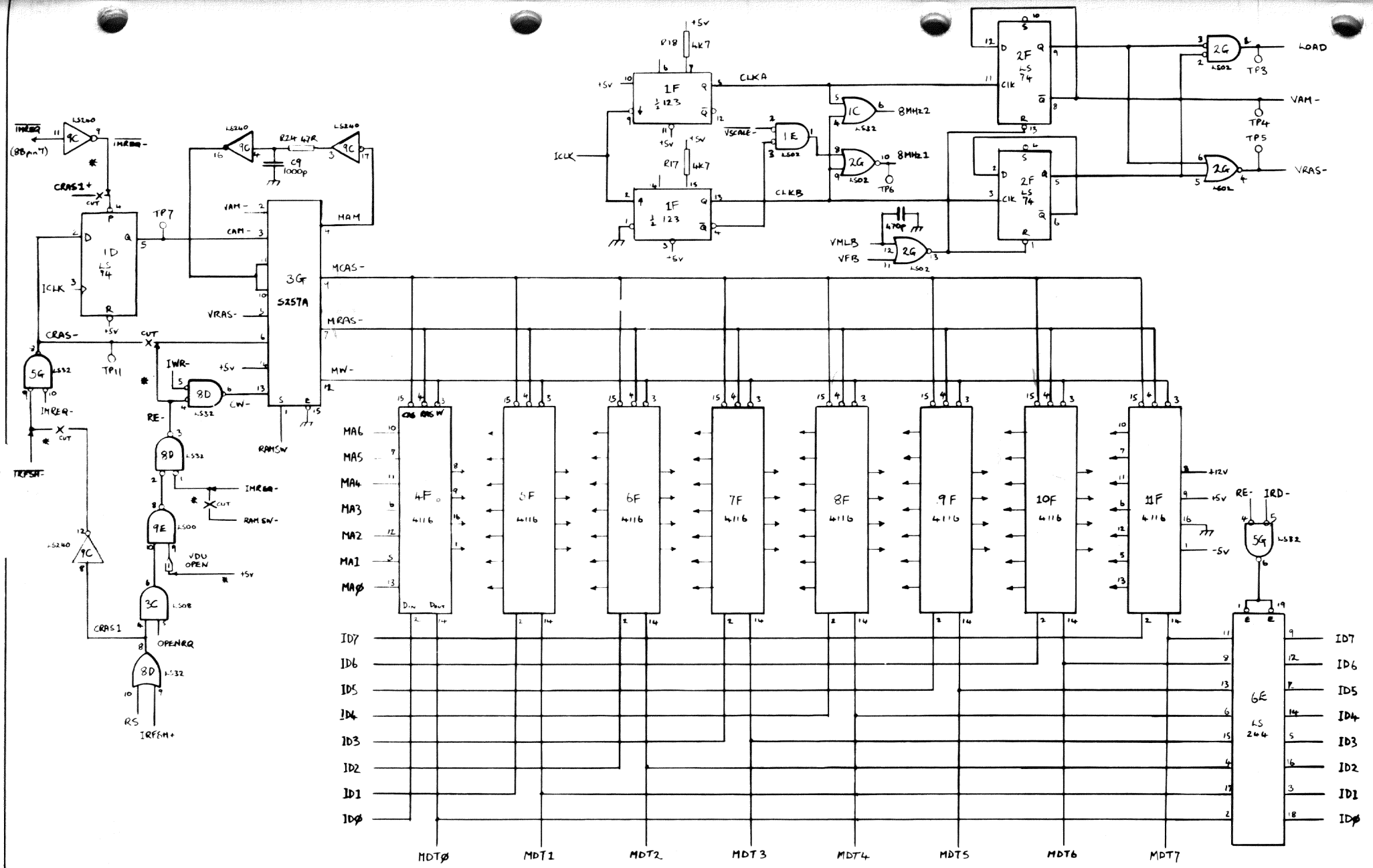
Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
5-2-80	G Winton	

TITLE: HIGH RESOLUTION GRAPHICS BOARD FOR LETTERING AND MEMORY MAPPING

DRAWING NO. 3807 RG-1

WP 1001



* PCB MODS

MODIFICATION		
A	6-2-80	
B	27-3-80	
C	8-6-80	
D	6-1-81	

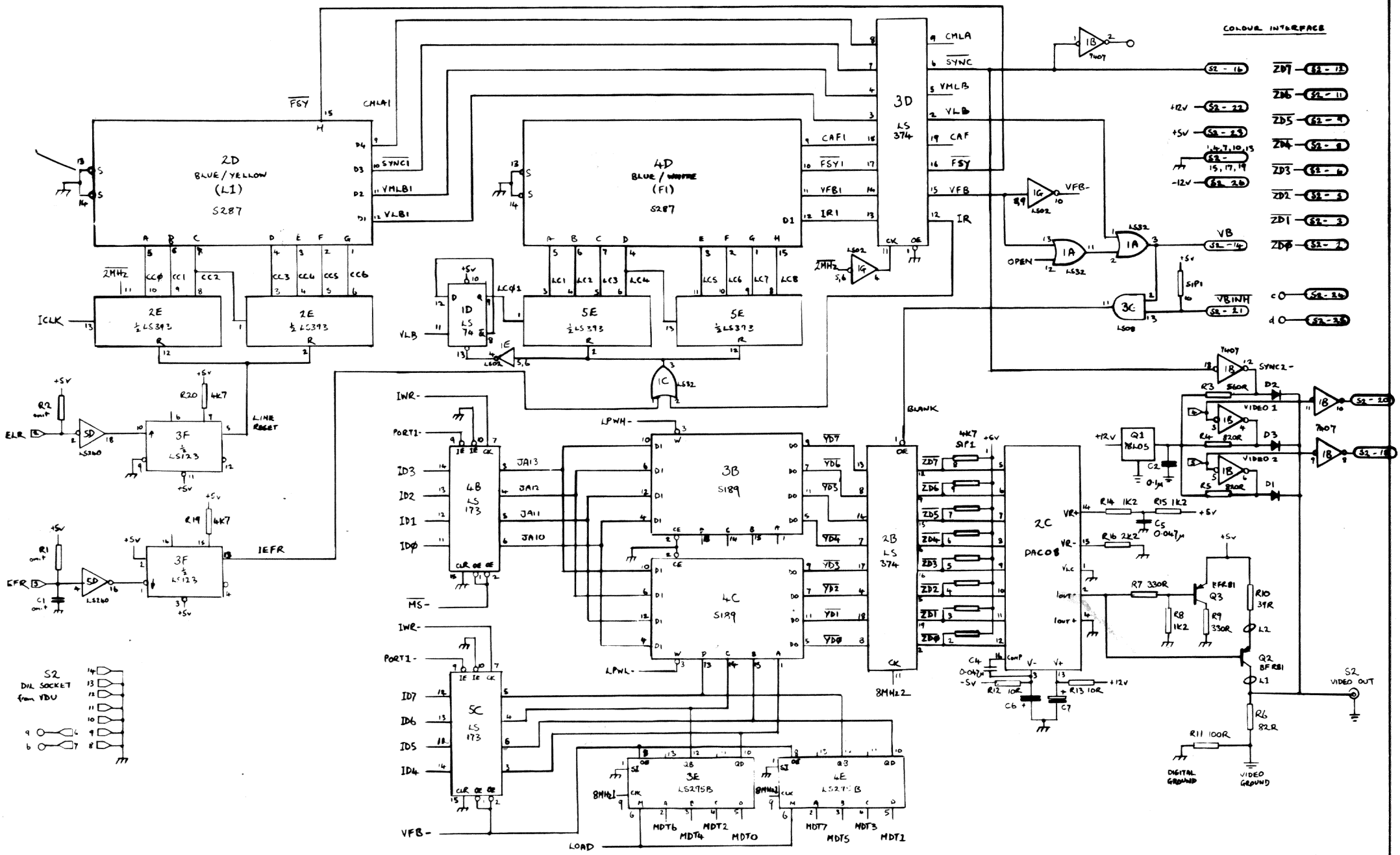
Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
6-2-80		

TITLE: HIGH RESOLUTION GRAPHICS BOARD
16K RAM & TIMING

DRAWING NO. 380Z-HRG-2

WP 1001



MODIFICATION		
A	6-2-80	
B	27-3-80	AJM

Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
6-2-80	G. Hinton	

TITLE: HIGH RESOLUTION GRAPHICS BOARD
VIDEO GENERATION

DRAWING NO. 3807 RG-3

4. SOFTWARE NOTES

4.1. SOME COS FEATURES

A complete specification of the Research Machines resident monitor program used with the current disc systems is given in Research Machines' Firmware Reference Manual.

4.1.1 A Restart Procedure

If, when you are running a program, you encounter a situation in which, for example, the keyboard will not respond to any input, then you can use the procedure described below to recover and often to restart your program. Of course if the error arose from a defect in your program then restarting may well lead to the same situation being repeated. However, even in this case restarting may enable you to gain some useful information about the cause of the problem or to save some of the results of your previous work.

When the situation arises

- press RESET

which clears the screen and the keyboard, re-initializes several COS work variables and pointers (but does not change the content of user main memory) and displays the COS message and prompt.

- Type B

This reloads CP/M and displays the system prompt. This step may not always be necessary but it is a wise precaution to carry it out because it ensures CP/M is properly initialized. It does not affect the contents of the memory space occupied by normal user programs. However, some programs, e.g., TXED, use space occupied by part of CP/M. Using this procedure with such programs may result in corruption of the program or its data. In such cases it is advisable to omit the above step.

- Type CTRL F

This enters the Front Panel (see below).

- Type J

In response to the prompt >

- type the restart address of the program to be restarted.

In some Research Machines software, for example, Extended BASIC Version 5, TXED etc., this restart is normally 103.

A BREAK point (identified by the message BREAK at the bottom of the screen) would not normally occur while using Research Machines software. If one does occur it often implies a hardware malfunction (often temporary). In these circumstances it is usually best to follow the restart procedure described above.

4.1.2 Front Panel

Usually when the keyboard is being polled for input, you can type CTRL F to enter the Front Panel. An automatic entry to the Front Panel occurs whenever the processor executes a break point instruction (code FF hex).

Front Panel is useful because it provides a wide range of features to help you debug a machine code program (or to discover why your program has failed), plus some machine - control commands.

An example of the use of the second of these features is the procedure for selecting and testing a printer (see Section 3.5). In this case once you have set up the printer you can exit from the Front Panel and continue executing your program where you left off by means of the Front Panel K command.

The debugging features provided by the Front Panel include the display of the contents of some of the Z80 registers, a set of commands for displaying and altering the contents of memory, and a command for stepping through a program one instruction at a time. A full description of all these features is given in the Firmware Reference Manual and an example of their use is given in the next Section.

4.2. PATCHING USING THE FRONT PANEL

If you want to create a "personal version" of some standard item of software, for example by altering the default value of some of the parameters, then this can be done by "patching" the program by using the Front Panel. Examples of how to do this can often be found in the documentation for the program concerned.

****

NOTES

1. Before patching an item of software make sure you have a back-up copy of the original program.

2. When you have patched an item of software give it a NEW name so you can distinguish it from the original program.

3. In order to patch an item of software successfully you will need up-to-date documentation for the program concerned.

****

As an example, to change the formatter's initial line length to 72, the procedure would be:

1.	**TXED**	Load TXED.
2.	**CTRL F**	Enter front panel (type Y in reponse to query).
3.	**M> 106 RETURN**	
4.	**I**	Point to formatter default value table. (first byte is initial line length)
5.	**48 RETURN**	72 in hex.
6.	**K**	Return to TXED.
7.	**EX ESC ESC**	Return to CP/M.
8.	**SAVE n EDIT.COM RETURN**	

Please refer to the TXED Release Note for the exact value of n in step 8.

4.3 PATCHING USING THE CP/M DDT UTILITY.

If you want to make substantial additions to an existing program then you can use the CP/M DDT Utility Program to "patch" the program. An example of this technique is given in Section 3.5 where DDT was used to add a patch to the PIP utility in order to create a program which can input files from a paper tape reader or a card reader. The material below describes some of the facilities provided by the CP/M DDT command.

4.3.1 Using DDT on the 380Z

The following Section does not describe all of the DDT commands since many of these duplicate features of the 380Z Front Panel, which is generally more powerful. If you mistype one of the commands described it may produce a bewildering response if it is one of those recognised by DDT but not described below. DDT is fully described in the CP/M DDT User's Guide which is available separately.

It should be noted that if a breakpoint instruction (code 0FFH) is executed while under the control of DDT it will cause a DDT breakpoint to be executed and the Front Panel will NOT be entered.

The following procedure is recommended:

- Load the module(s) to be tested into memory using the DDT I and R commands (described below).
- Type CTRL C to exit from DDT.
- Type CTRL F to enter the Front Panel if debugging is necessary
- Save the program with the SAVE command if required.

4.3.2 An Introduction to DDT

Initiating DDT

The DDT program allows dynamic interactive testing and debugging of programs generated in the CP/M environment it is however suggested that DDT is not used for debugging purposes as the Front Panel is, in many ways, superior. The debugger program is initiated by typing one of the following commands at the CP/M Console Command level.

```
DDT
DDT filename.HEX
DDT filename.COM
```

where "filename" is the name of the program to be loaded. In these cases, the DDT program is brought into main memory in the place of the normal CP/M Console Command Processor, and thus it resides directly below the Basic Disc Operating System portion of CP/M. The BDOS starting address, which is located in the address field of the JP instruction at location 5H, is altered to reflect the reduced Transient Program Area size.

The second and third forms of the DDT command shown above perform the same actions as the first, except there is a subsequent automatic load of the specified HEX or COM file. The action is similar to the sequence of

commands

```
DDT
Ifilename.HEX or Ifilename.COM
R
```

where the I and R commands set up and read the specified program to test (see the explanation of the I and R commands below for exact details).

Operating DDT

Following the sign on message, DDT prompts the operator with the character "-" and waits for input commands from the console. The operator can type any of several single character commands, terminated by a carriage RETURN to execute the command. Each command can be up to 32 characters in length (an automatic carriage RETURN is inserted as the 33rd character), where the first character determines the command type. The only two commands normally used with an 380Z are:-

```
I  set up a standard input file control block
R  read program for subsequent testing
```

Execution of DDT may be terminated at any point by typing CTRL C.

You can save the current contents of memory in a file by using a SAVE command of the form

```
SAVE n filename.COM
```

where n is the number of pages (256 byte blocks) to be saved on disc (see below)

****

NOTE

When SAVE is used to save a memory image it does not save the machine state, and thus the program must be restarted from the beginning when you want to use it or to make further tests.

****

4.3.3 DDT Commands

Two individual commands are described in some detail. In each case, the operator must wait for the prompt character (-) before entering the command.

The I (Input) Command

The I command allows the user to insert a file name into the CP/M default file control block created by CP/M for the use of transient programs. The default FCB can be used by the program under test as if it had been passed by the CP/M Console Processor. This file name is also used by DDT for reading additional files.

The form of the I command is

```
Ifilename
```

or

```
Ifilename.filetype
```

then Subsequent R commands can be used to read the file (see the R command for further details).

The R (Read) Command

The R command is used in conjunction with the I command to read files from the disc into the transient program area in preparation for the debug run. The forms are

```
R
Rb
```

where b is an optional bias address which is added to each program or data address as it is loaded. The load operation must not overwrite any of the system parameters from 000H through 0FFH (i.e., the first page of memory), nor should it overwrite CP/M, or the relocated DDT, or COS workspace. If b is omitted, then b = 0000 is assumed. The R command requires a previous I command, specifying the name of a file. The load address for each record is obtained from each individual record of a .HEX file, while an assumed load address of 100H is taken for files of any other extension.

Any number of R commands can be issued following the I command to re-read the program assuming the default file control block has not been destroyed. Further, any file specified with the filetype "HEX" is assumed to contain machine code in Intel hex format (created, for example, by the ZASM assembler), and all others are assumed to contain machine code in pure binary form (produced, for example, by the SAVE command).

The command

```
DDT filename.filetype
```

which can be used to initiate the DDT program is similar to the commands

```
DDT
-Ifilename.filetype
-R
```

Whenever the R command is issued, DDT responds with either the error indicator "?" which means that the file cannot be opened, or a checksum error has occurred in a .HEX file, or with a load message taking the form

```
NEXT PC
nnnn pppp
```

where nnnn is the next address following the loaded program, and pppp is the assumed value of the program counter (100H for COM files, or taken from the last record if a HEX file is specified).

****

NOTES

1. All numerical values input to DDT must be expressed as hexadecimal numbers.

2. All numerical values output by DDT are expressed in hexadecimal.

3. The DDT loading operation can be used to determine the length of a program for use in a Save command. For example

```
A>DDT TEST. COM  RETURN

NEXT  PC
1D00  0100
```

1DOO hex is one greater than the last address in the program - to convert this to blocks use the following algorithm:

a) Subtract 1.

b) Now take the first two digits and convert them to decimal - the result is the block size of the program.

In this case we have

```
1DOO-1=1CFF
```

1C Converts to 28 so we can save a new copy of TEST by

```
CTRL C

SAVE 28 NEW.COM
```

****

4.4 ADDITIONAL CP/M COMMANDS: LOAD, DUMP and CONFIG

This Section provides a description of three CP/M transient commands which are not covered in the Mini Disc System User Guide and which may be useful if you are concerned with assembly language programming or with tailoring CP/M to match characteristics of your system.

4.4.1 LOAD

If you have a file called, for example, OBJECT.HEX on the disc in drive A containing machine code in standard INTEL hexadecimal format you can convert it into a CP/M command file, called OBJECT.COM and stored on the disc in drive A by the command

A>LOAD OBJECT [RETURN]

Typically a .HEX file is produced by assemblers and would commonly be converted to a .COM file so that it can easily be executed.

You can carry out the load operation on files from any of the disc drives simply by specifying the drive name in the LOAD command, the command file is always written onto the specified disc. For example the command

B>LOAD D:OBJECT [RETURN]

reads a file named OBJECT.HEX from disc drive D and outputs a file named OBJECT.COM to disc drive D (the specified drive).

****

NOTES

1. The "HEX" file should contain valid INTEL hexadecimal code - for example code produced by the Research Machines ZASM Assembler.

2. The "HEX" file must contain records whose block addresses are in ascending order (LOAD will fill any "gaps" left by the memory address with zeros). Files produced by the Research Machines ZASM Assembler are in this format.

3. LOAD can only be used to load programs which fit into the CP/M Transient Program Area, any other programs must be loaded using the CP/M DDT Utility program (see Section 4.3).

****

4.4.2 DUMP

The DUMP command can be used to display the contents of any file in hexadecimal form. Any file can be dumped, since all data on the disc is in binary form. However, DUMP will normally only be used with COM files, other files being displayed by the TYPE command. The argument of the dump command must be the name of a specific file and may not, for example, contain * or ?.

A>DUMP OBJECT.COM [RETURN]

displays in hexadecimal notation the contents of the file OBJECT.COM. The file is displayed in lines, 16 bytes to a line, with the absolute address of the first byte in the line at the left-hand end of each line.

4.4.3 CONFIG

The CONFIG command can be used to create a version of CP/M to match the characteristics of the hardware configuration of your system. To configure a version of CP/M proceed as follows:

- **turn the system on, insert a system disc containing CONFIG in drive A and load CP/M**

- **type CONFIG RETURN**

CONFIG is executed and the message

```
RML CONFIG V 4.0
Copyright (C) 1980 by Research Machines

Configure which disc (A,B):
```

will appear

- **type B**

The message

```
Insert disc in B then type RETURN
```

will appear.

- **insert the disc containing CP/M which is to be configured in drive B**

- **type RETURN**

A message like

```
LINE PRINTER - unselected
HRG - BOARD - Graphics
G(raphics, L(ineprinter,
X(ecute, Q(uit, R(estart, S(how, H(elp:
```

will appear. Choose the appropriate option and type the command letter. For example to configure a lineprinter

- **type L**

The message

```
Leave unselected (Y/N):
```

will appear

- **type N**

The message

Printer type

0 - console screen
1 - SIO-1
2 - SIO-2/3
3 - parallel interface
4 - SIO-4
5 - SIO-5
6 - SIO-6

Select number:

will appear. To select the SIO-4 interface

- type 4

The message,

Baud code

0 - 100
1 - 300
2 - 600
3 - 1200
4 - 2400
5 - 4800
6 - 9600

Select number:

will appear. To select 1200 baud

- type 3

The system has now been set up to configure a printer and CONFIG now represents the choice of options.

- Type X

to create a configured version of CP/M and copy it onto disc B. When the process is complete the message

Configuration complete

will appear and CP/M is restarted.

The G option allows you to select the graphics board as memory, the R option allows you to restart CONFIG - for example if you notice you have selected the wrong disc, the Q option terminates CONFIG without reconfiguring CP/M and the H option provides a page of text to describe the options available.

****

NOTES

1. If you attempt to configure a disc other than A or B the message

```
Invalid disc, try again
Configure which disc (A,B):
```

will appear.

2. If there is no disc inserted in the drive which you select to configure, or the disc is not inserted properly then the message

```
Error in reading system -
Configurations aborted
```

will appear and CP/M is restarted.

3. If you specify an invalid option or an invalid selection then the invitation will simply be repeated.

4. You can over-ride the printer option configured into CP/M by using the Front Panel O command.

****

4.5 TERMINAL MODE SOFTWARE

The Research Machines 380Z may be used as a terminal to another computer. To do this two things are needed, a Serial Interface and RML 'Terminal Mode Software' (the program TERM4). The interfaces suitable for use with Terminal Mode Software are the SIO-4 (now supplied as standard with disc machines) and its equivalents (See Section 3.3). The program TERM4 is loaded under CP/M in the normal way. When it is run it causes the 380Z to emulate a 'Digilog' terminal. Characters from the 380Z keyboard are sent through the interface to the central computer, and all characters received from the interface are displayed on the 380Z screen. Certain control characters cause cursor movement on the screen.

The Terminal Mode Software and the appropriate documentation can be ordered from the Sales Office in the normal way - please quote your COS version number when ordering.

4.6 UTILITY SUBROUTINES

This Section contains assembler subroutines to perform useful functions:-

1. INPUT FROM TTY WITH SIO-2 AND SIO-3

2. INPUT FROM TTY WITH SIO-1

3. DISPLAY MEMORY ON SCREEN

4. DUMP MEMORY TO PRINTER

5. DUMP ON CASSETTE IN INTEL FORMAT

6. DISPLAY MEMORY AS ASCII CHARACTERS ON SCREEN

4.6.1 Input from TTY with SIO-2 and SIO-3

This subroutine will read a byte from the SIO-2 or SIO-3 interface. These interfaces differ only in the signal levels that they use, the driver software is the same for both. Please note that this subroutine will only work on COS 3.4 and earlier.

```
0001 ;
0002 ;SERIAL INPUT FROM SIO-2 AND SIO-3
0003 ;
0004 ;This utility subroutine reads
0005 ;the next byte from a serial
0006 ;device (such as an ASR33
0007 ;TELETYPE) each time it is
0008 ;called.
0009 ;The device is interfaced
0010 ;to the 380Z via an SIO-2
0011 ;or SIO-3 interface.
0012 ;
0013 ;Users can place the code between
0014 ;the lines of percent signs
0015 ;anywhere in a program. Each
0016 ;time address 'PR:' is called,
0017 ;the next byte from the device
0018 ;will be returned in register A.
0019 ;No other registers are altered.
0020 ;This area of code will run
0021 ;at any address since it is
0022 ;position independent.
0023 ;
0024 ;Optionally the whole routine
0025 ;can be typed into memory
0026 ;as it stands, starting at
0027 ;100H. Then when started at
0028 ;100H, the first section will
0029 ;shift the position independent
0030 ;section to the top of memory
0031 ;(just below the address contained
0032 ;in 'HIMEM') and link the
0033 ;'TIV' vector to 'PR:' such
0034 ;that programs can access
0035 ;the routine via EMT GETBYT
0036 ;RML programs containing
0037 ;the cassette file system
0038 ;can also access the routine
0039 ;after it has been loaded
0040 ;in this way by specifying
0041 ;that input should be from
0042 ;device 'RDR:'.
```

```
                0043 ;
                0044 ;The subroutine contains provision
                0045 ;for automatic reader control
                0046 ;of an ASR33 fitted with the
                0047 ;appropriate hardware and
                0048 ;interfaced via the SIO-2 or SIO-3
                0049 ;
                0050 ;If this program is being used
                0051 ;on a disc system, CP/M will have
                0052 ;to be reloaded after all the
                0053 ;input from the serial device
                0054 ;has been gathered.
                0055 ;
                0056 ;This program was assembled
                0057 ;with the disc version of
                0058 ;the RML Z80 Assembler
                0059 ;
FF21 =          0060 TIV     EQU     0FF21H
0006 =          0061 HIMEM   EQU     0006H
FF3C =          0062 ADDR    EQU     0FF3CH
FF3E =          0063 UMASK   EQU     0FF3EH
FF3F =          0064 DLYTIM  EQU     0FF3FH
FBFF =          0065 UPORT   EQU     0FBFFH
                0066
0100            0067         ORG     100H
                0068
                0069 ;The first section moves the
                0070 ;position independent section
                0071 ;to the top of memory, then
                0072 ;resets 'HIMEM' to point just
                0073 ;below the entry point:
                0074
0100 ED5B0600   0075         LD      DE,(HIMEM)
0104 1B         0076         DEC     DE
0105 217D01     0077         LD      HL,PEND
0108 015000     0078         LD      BC,PLEN
010B EDB8       0079         LDDR
010D 13         0080         INC     DE
010E ED530600   0081         LD      (HIMEM),DE
                0082
                0083 ;The device handler is then linked
                0084 ; to the 'TIV' transfer vector:
                0085
0112 ED5322FF   0086         LD      (TIV+1),DE
                0087
                0088 ;The user I/O port, mask and baud
                0089 ;rate are initialised.
                0090
0116 213EFF     0091         LD      HL,UMASK
0119 7E         0092         LD      A,(HL)  ;Clear RDR
011A E6FC       0093         AND     0FCH
011C 77         0094         LD      (HL),A
011D CD7701     0095         CALL    UPDUSR
0120 218903     0096         LD      HL,389H ;For 110 baud
0123 223FFF     0097         LD      (DLYTIM),HL
                0098
```

```
                0099 ;Finally the start address is
                0100 ;cleared and control is returned
                0101 ;to the COS monitor:
                0102
0126 210000     0103            LD      HL,0
0129 223CFF     0104            LD      (ADDR),HL
012C F700       0105            EMT     0
                0106
                0107 ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                0108
                0109 ;Reader input subroutine
                0110
                0111 ;The RDR relay (if any) is pulsed
                0112 ;for each byte
                0113
012E E5         0114 PR:        PUSH    HL
012F D5         0115            PUSH    DE
0130 C5         0116            PUSH    BC
0131 213EFF     0117            LD      HL,UMASK
0134 ED5B3FFF   0118            LD      DE,(DLYTIM)
0138 D5         0119            PUSH    DE
0139 CB3A       0120            SRL     D       ;DE <- DE/2
013B CB1B       0121            RR      E
013D 0608       0122            LD      B,8     ;8 data bits
                0123
013F CBCE       0124            SET     1,(HL)  ;Step RDR
0141 EF34       0125            CALR    UPDUSR
                0126
0143 3AFFFB     0127 PR1:       LD      A,(UPORT)  ;Get start bit
0146 CB47       0128            BIT     0,A
0148 20F9       0129            JR      NZ,PR1
                0130
014A EF1F       0131            CALR    DELAY   ;Wait 1/2 bit
                0132
014C 3AFFFB     0133            LD      A,(UPORT)  ;Still there?
014F CB47       0134            BIT     0,A
0151 20F0       0135            JR      NZ,PR1  ;If not assume glitch
                0136
0153 CB8E       0137            RES     1,(HL)  ;Stop RDR
0155 EF20       0138            CALR    UPDUSR
                0139
0157 D1         0140            POP     DE
0158 EF11       0141 PR2:       CALR    DELAY   ;Wait 1 bit
015A 3AFFFB     0142            LD      A,(UPORT)  ;Read data
015D 1F         0143            RRA             ;Build 8 bits
015E CB19       0144            RR      C       ;In C reg
0160 10F6       0145            DJNZ    PR2
                0146
0162 EF07       0147            CALR    DELAY   ;Wait for last
                0148
0164 79         0149            LD      A,C     ;Get byte
0165 E67F       0150            AND     7FH     ;Strip parity
0167 C1         0151            POP     BC
0168 D1         0152            POP     DE
0169 E1         0153            POP     HL
016A C9         0154            RET
```

```
                0155
016B D5         0156 DELAY:  PUSH    DE
016C F5         0157         PUSH    AF
016D 7A         0158 D1:     LD      A,D
016E B3         0159         OR      E
016F 2803       0160         JR      Z,D2
0171 1B         0161         DEC     DE
0172 18F9       0162         JR      D1
0174 F1         0163 D2:     POP     AF
0175 D1         0164         POP     DE
0176 C9         0165         RET
                0166
0177 F5         0167 UPDUSR: PUSH    AF
0178 7E         0168         LD      A,(HL)  ;From mask
0179 32FFFB     0169         LD      (UPORT),A
017C F1         0170         POP     AF
017D C9         0171         RET
                0172
                0173 ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                0174
017D =          0175 PEND    EQU     $-1
0050 =          0176 PLEN    EQU     $-PR
                0177
0000            0178         END
```

Symbols:

```
ADDR    FF3C  D1      016D  D2      0174  DELAY   016B  DLYTIM FF3F
HIMEM   0006  PEND    017D  PLEN    0050  PR      012E  PR1    0143
PR2     0158  TIV     FF21  UMASK   FF3E  UPDUSR  0177  UPORT  FBFF
```

No errors

4.6.2 Input from TTY with SIO-1

The following code can be used as a subroutine to read an incoming character from the SIO-1:

```
DDE5      S1KIN: PUSH    IX
DD21E0FB         LD      IX,0FBE0H
DDCB0166 TEST:   BIT     4,(IX+1)  ;Test data ready flag
28FA             JR      Z,TEST    ;Loop if no char.
DD7E00           LD      A,(IX+0)  ;Read char.
DD360302         LD      (IX+3),2  ;Reset data ready flag
DDE1             POP     IX
C9               RET
```

Before using the SIO-1 for input, it must be initialised. This is done if the SIO-1 line printer option is selected. The code for this can be found in the monitor listing.

4.6.3 Display Memory as ASCII Characters on Screen

MDISP	First	100H
	Last	127H
	Start	100H

This program provides a rapid display of the memory. Each byte of memory is displayed as the corresponding ASCII or graphics character, unless the code is less than 20H (32 decimal), in which case a 'C' (ASCII 43H) is displayed. This modification is used because, in the interests of brevity of the program, a monitor routine (EMT OUTC, see monitor manual) is used to output the characters.

On start at 100H, the program will prompt for a start address with 'START>'. This requires a hexadecimal number, and will be the address of the first byte to be output. The screen is first cleared and then filled with characters from memory.

At any stage, CONTROL-C will send control back to the monitor.

MDISP is written entirely in position independent code, and can therefore run at any address. It makes use of no memory locations other than those which contain the program itself, although it calls various monitor routines via EMTs.

```
                0001 ;
                0002 ;**********************************
                0003 ;       MEMORY DISPLAY
                0004 ;**********************************
                0005 ;
                0006 ;
0100            0007            ORG     100H
0100 EF00       0008            CALR    $+2         ;put PC on top of stack
0102 111F00     0009            LD      DE,MSGS-$
0105 E1         0010            POP     HL
0106 19         0011            ADD     HL,DE       ;HL<- Address of message
0107 3E0D       0012            LD      A,0DH       ;<CR>
0109 F701       0013            EMT     OUTC
010B F717       0014            EMT     MSG
010D F713       0015            EMT     GETHEX      ;HL<-Start Addr.
010F 3E0C       0016            LD      A,0CH       ;<FF>
0111 F701       0017            EMT     OUTC        ;Clear screen
0113 7E         0018 NEXTCH:    LD      A,(HL)      ;A<-Memory Byte
0114 FE20       0019            CP      20H         ;If it's <' ',
0116 3002       0020            JR      NC,$+4      ;Set it
0118 3E43       0021            LD      A,'C'       ;To 'C'
011A F701       0022            EMT     OUTC        ;Output A
011C F702       0023            EMT     KBDC        ;Check for   C
011E 23         0024            INC     HL          ;Point to next byte
011F 18F2       0025            JR      NEXTCH      ;and loop
```

```
                0026
0121 53544152   0027 MSGS:   DEFM    'START>'
0127 FF         0028        DEFB    0FFH
0002 =          0029 KBDC   EQU     2
0001 =          0030 OUTC   EQU     1
0017 =          0031 MSG    EQU     17H
0013 =          0032 GETHEX EQU     13H
0100            0033        END     100H
```

Symbols:

```
GETHEX 0013  KBDC  0002  MSG    0017  MSGS   0121  NEXTCH 0113
OUTC   0001
```

No errors

4.6.4 Dump Memory to Printer

Program MOUT outputs selected areas of memory to the printer with one byte per line, each byte in the format of two hexadecimal digits. This output could be used, for example, as an aid to the manual disassembly of machine code programs.

On starting at 100H, the program prompts in turn for 'FIRST' and 'LAST'. These are the initial and final addresses of bytes to be output. If FIRST is less than LAST the program will return immediately to the monitor. WITH cos 4.0 the instruction JP C, 0 must be inserted after EMT FSTLST below. This action is also taken if an attempt is made to list from location 0 to location 0FFFFH.

After satisfactory values for FIRST> and LAST> have been input, the area of memory between these values is sent to the printer. Control is returned to the monitor either when the program has finished or by pressing CTRL C.

MOUT is written entirely in position independent code, and can be shifted to any area of RAM and run from there without modification.

To modify the program to direct output to a device other than the printer, modify the subroutine CHOUT at 0132H to include your own peripheral handler.

```
                0001 ;
                0002 ;---------------
                0003 ;    MOUT
                0004 ;---------------
                0005 ;This program dumps memory to the
                0006 ;line printer, one byte per line,
                0007 ;each byte being represented by two
                0008 ;hexadecimal digits.
                0009 ;The program is written entirely
                0010 ;in position independent code.
                0011 ;
                0012 ;
0100            0013          ORG     100H
0100 EF00       0014          CALR    $+2       ;put PC on top of stack
0102 113300     0015          LD      DE,BUF-$
0105 DDE1       0016          POP     IX
0107 DD19       0017          ADD     IX,DE
0109 F725       0018          EMT     FSTLST
010B 23         0019          INC     HL
010C F702       0020 NEXT:    EMT     KBDC      ;Check for  C
010E 3E0D       0021          LD      A,0DH
0110 EF20       0022          CALR    CHOUT     ;New line
0112 3E0A       0023          LD      A ,0AH    ;<LF>
0114 EF1C       0024          CALR    CHOUT
0116 1A         0025          LD      A,(DE)    ;Get byte
0117 EF08       0026          CALR    OUTP      ;& output it
0119 2B         0027          DEC     HL        ;Decrement Byte count
011A 7C         0028          LD      A,H
011B B5         0029          OR      L
011C 13         0030          INC     DE        ;Increment pointer
011D 20ED       0031          JR      NZ,NEXT   ;Enough?
011F F700       0032          EMT     0
```



```
0121 E5       0033 OUTP:   PUSH    HL
0122 DDE5     0034         PUSH    IX
0124 E1       0035         POP     HL
0125 F715     0036         EMT     BYTEO    ;Convert A to hex
0127 2B       0037         DEC     HL
0128 2B       0038         DEC     HL
0129 7E       0039         LD      A,(HL)
012A EF06     0040         CALR    CHOUT    ;First digit
012C 23       0041         INC     HL
012D 7E       0042         LD      A,(HL)
012E EF02     0043         CALR    CHOUT    ;& second
0130 E1       0044         POP     HL
0131 C9       0045         RET
              0046 ;
              0047 ;Subroutine CHOUT
              0048 ;The output channels through this sub.
              0049 ;to simplify patching to send output
              0050 ;whither required.
              0051 ;
0132 F705     0052 CHOUT:  EMT     LPOUT    ;Char to line printer
0134 C9       0053         RET
              0054
0135          0055 BUF:    DEFS    2        ;Workspace for BYTEO
0002 =        0056 KBDC    EQU     2
0015 =        0057 BYTEO   EQU     15H
0001 =        0058 OUTC    EQU     1
0005 =        0059 LPOUT   EQU     5
0025 =        0060 FSTLST  EQU     37
0100          0061         END     100H
```

Symbols:

```
BUF     0135  BYTEO   0015  CHOUT   0132  FSTLST 0025  KBDC   0002
LPOUT   0005  NEXT    010C  OUTC    0001  OUTP   0121
```

No errors

4.6.5 Dump on Cassette in Intel Hex Format

THIS PROGRAM IS NOT APPROPRIATE FOR 380Z VARITEXT (80/40 CHARACTER) SYSTEMS.

The Program CDUMP dumps selected areas of memory onto cassette tape in 'Intel' hex format, suitable for reading by CLOAD. The output is not file structured, so reading via the cassette file system must be done on the 'RDR' channel.

On starting at 100H, the program prompts 'FIRST>' and 'LAST>'. The response to these are the addresses (in hexadecimal) of the first and last bytes to be dumped. If 'FIRST' is greater than 'LAST' control returns immediately to the monitor. Attempts to dump from address 0000 to 0FFFFH will also cause an immediate return to the monitor.

The program then prompts for 'START'. This requires a hexadecimal address which will be the start address of the dumped program. However, CLOAD does not implement this feature, so program dumps intended for use with CLOAD should be done by typing Carriage Return in reply to 'START'. However, before terminating the start address, start the cassette recorder in 'RECORD'.

Nothing will appear to happen until the dump has finished. Control will return to the monitor.

CDUMP is written entirely in position independent code. It may be shifted to any portion of RAM and run from there.

```
                0001 ;
                0002 ;-----------------
                0003 ;    CDUMP
                0004 ;-----------------
                0005 ;This program dumps areas of memory
                0006 ;onto cassette in 'INTEL' format
                0007 ;it is written entirely in PIC
                0008 ;
                0009 ;
                0010 ;
0100            0011        ORG     100H
0100 EF00       0012        CALR    $+2     ;put PC on top of stack
0102 117200     0013        LD      DE,BUF-$
0105 DDE1       0014        POP     IX
0107 DD19       0015        ADD     IX,DE
```



```
                0016
0109 F725       0017            EMT     FSTLST
010B EF1C       0018 NREC:      CALR    HEAD    ;Output start of record
010D 0618       0019            LD      B,18H   ;For 18H bytes
010F 1A         0020 NBYTE:     LD      A,(DE)  ;Output the byte
0110 EF48       0021            CALR    OUTP
0112 2B         0022            DEC     HL      ;Decrement the byte count
0113 7C         0023            LD      A,H     ;& test for zero
0114 B5         0024            OR      L
0115 2809       0025            JR      Z,LREC
0117 13         0026            INC     DE      ;Increment byte pointer
0118 F702       0027            EMT     KBDC    ;Check for  C
011A 10F3       0028            DJNZ    NBYTE   ;Repeat for next byte
011C EF35       0029            CALR    CHKSM   ;Output Checksum
011E 18EB       0030            JR      NREC    ;and goto next record
0120 EF31       0031 LREC:      CALR    CHKSM   ;Output Checksum
0122 D1         0032            POP     DE      ;GET START ADDR.
0123 EF04       0033            CALR    HEAD    ;Output head of end record
0125 EF2C       0034            CALR    CHKSM   ;And the checksum
0127 F700       0035            EMT     0
                0036 ;
                0037 ;Subroutine Head
                0038 ;Outputs <CR>,<LF>:
                0039 ;Zeroes the checksum (stored in A')
                0040 ;Outputs the No. of data bytes
                0041 ;Outputs the data addr.
                0042 ;Outputs 00
                0043 ;
                0044 ;First <CR>,<LF>
0129 3E0D       0045 HEAD:      LD      A,CR
012B EF44       0046            CALR    CHOUT
012D 3E0A       0047            LD      A,LF
012F EF40       0048            CALR    CHOUT
0131 3E3A       0049            LD      A,':'
0133 EF3C       0050            CALR    CHOUT
                0051 ;
                0052 ;Now zero the checksum
0135 08         0053            EX      AF,AF'
0136 AF         0054            XOR     A
0137 08         0055            EX      AF,AF'
                0056 ;
                0057 ;Next the number of data bytes.
                0058 ;This will be the smaller of 18H and HL
0138 E5         0059            PUSH    HL
0139 011800     0060            LD      BC,18H
013C AF         0061            XOR     A
013D ED42       0062            SBC     HL,BC
013F E1         0063            POP     HL
0140 3804       0064            JR      C,SREC  ;Jump if <18H bytes left
0142 3E18       0065            LD      A,18H
0144 1801       0066            JR      HEAD1
                0067
0146 7D         0068 SREC:      LD      A,L
0147 EF11       0069 HEAD1:     CALR    OUTP
                0070 ;
```



```
                    0071 ;Next the block address, stored in DE
0149 7A             0072          LD      A,D
014A EF0E           0073          CALR    OUTP
014C 7B             0074          LD      A,E
014D EF0B           0075          CALR    OUTP
                    0076
                    0077 ;And lastly the zeroes
014F AF             0078          XOR     A
0150 EF08           0079          CALR    OUTP
0152 C9             0080          RET
                    0081
                    0082 ;Subroutine CHKSM
                    0083 ;Outputs the checksum, held in A'
                    0084
0153 08             0085 CHKSM:   EX      AF,AF'
0154 ED44           0086          NEG
0156 EF02           0087          CALR    OUTP
0158 08             0088          EX      AF,AF'
0159 C9             0089          RET
                    0090
                    0091 ;Subroutine OUTP
                    0092 ;outputs the accumulator as 2 hex digits
                    0093 ;It also updates the checksum
015A E5             0094 OUTP:    PUSH    HL
015B 6F             0095          LD      L,A
015C 08             0096          EX      AF,AF'
015D 85             0097          ADD     A,L
015E 08             0098          EX      AF,AF'
015F DDE5           0099          PUSH    IX      ;IX points to 2 spare bytes
                    0100
0161 E1             0101          POP     HL
0162 F715           0102          EMT     BYTEO   ;BYTEO does the conversion
                    0103                          ;from binary to ascii
0164 2B             0104          DEC     HL      ;Restore HL
0165 2B             0105          DEC     HL
0166 7E             0106          LD      A,(HL)
0167 EF08           0107          CALR    CHOUT   ;First digit
0169 00             0108          NOP
016A 23             0109          INC     HL
016B 7E             0110          LD      A,(HL)
016C EF03           0111          CALR    CHOUT   ;Second digit
016E 00             0112          NOP
016F E1             0113          POP     HL
0170 C9             0114          RET
                    0115
                    0116
0171 F703           0117 CHOUT:   EMT     PUTBYT
                    0118 ;
                    0119 ;All the output is directed through this
                    0120 ;subroutine, so that patches are easily made.
                    0121 ;
0173 C9             0122          RET
                    0123 ;
                    0124 ;
                    0125 ;
                    0126 ;
0174                0127 BUF:     DEFS    2
0002 =              0128 KBDC     EQU     2
```

```
0015 =          0129 BYTEO    EQU     15H
0001 =          0130 OUTC     EQU     1
0013 =          0131 GETHEX   EQU     13H
0003 =          0132 PUTBYT   EQU     3
0025 =          0133 FSTLST   EQU     37
                0134
000D =          0135 CR       EQU     13
000A =          0136 LF       EQU     10
                0137
0100            0138          END     100H
```

Symbols:

```
BUF     0174  BYTEO   0015  CHKSM   0153  CHOUT   0171  CR      000D
FSTLST  0025  GETHEX  0013  HEAD    0129  HEAD1   0147  KBDC    0002
LF      000A  LREC    0120  NBYTE   010F  NREC    010B  OUTC    0001
OUTP    015A  PUTBYT  0003  SREC    0146
```

No errors

APPENDICES

APPENDIX.1: HEXADECIMAL NUMBERS

Hexadecimal Notation

Hexadecimal notation is widely used in computer generated output (e.g. assembler listings and the Front Panel display) to represent addresses and the contents of store locations. Where the decimal number system uses a base of 10, and the symbols 0-9, the hexadecimal system uses a base of 16, and the symbols 0-9, A, B, C, D, E, F. The relation between counting in decimal and in hexadecimal goes like this

One	1	Eight	8	Fifteen	F
Two	2	Nine	9	Sixteen	10
Three	3	Ten	A	Seventeen	11
Four	4	Eleven	B	Eighteen	12
Five	5	Twelve	C	etc.	
Six	6	Thirteen	D		
Seven	7	Fourteen	E		

Four bit binary numbers

There are sixteen different four-bit binary numbers each of which can be used to represent a different hex digit.

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Conversely a single hex digit is a simple way of defining a four bit number.

Eight bit binary numbers

To cope with eight bits, the convention is to divide the eight bits into two groups of four and to write a pair of hex digits, the first corresponding to the left-hand four bits (most significant), and the second to the right-hand four bits (least significant). So a single byte eight bit number containing 10101111 can be written in hex as AF (1010=A, 1111=F).

Sixteen bit binary numbers

In the same way that an eight bit number can be represented by two hex digits, a sixteen bit number can be represented by four hex digits.

Negative numbers within the 380Z are often held in twos complement notation. In this notation, a number is negated by complementing each bit and adding one. The advantage of this representation is that two such quantities can be added without treating negative numbers specially.

	A	E
Write number in binary	1010	1110
Invert each bit	0101	0001
Add 1	0101	0010
Convert back to hex and insert sign	- 5	2
Convert to decimal	- 8	2

So in twos complement notation the hexadecimal number AE is equivalent to the decimal number - 82

Alternatively, the same values could have been reached using hexadecimal arithmetic:

Subtract value from 100 hex	100 - AE	52
Insert sign		—
Convert to decimal		- 52 hex
		- 82

Hexadecimal to Decimal Conversion

Hexadecimal to decimal conversion can be carried out with the aid of the table below:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	00	000
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	0
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	256	4096
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	512	8192
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	768	12288
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	1024	16384
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	1280	20480
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	1536	24576
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	1792	28672
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	2048	32768
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	2304	36864
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	2560	40960
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	2816	45056
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	3072	49152
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	3328	53248
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	3584	57344
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	3840	61440

For example,
1A80 hexadecimal

column 000	entry	1	=	4096
column 00	entry	A	=	2560
column 0	entry	8	=	128
				6784

APPENDIX 2: CHARACTER SETS

ASCII CODE TABLE

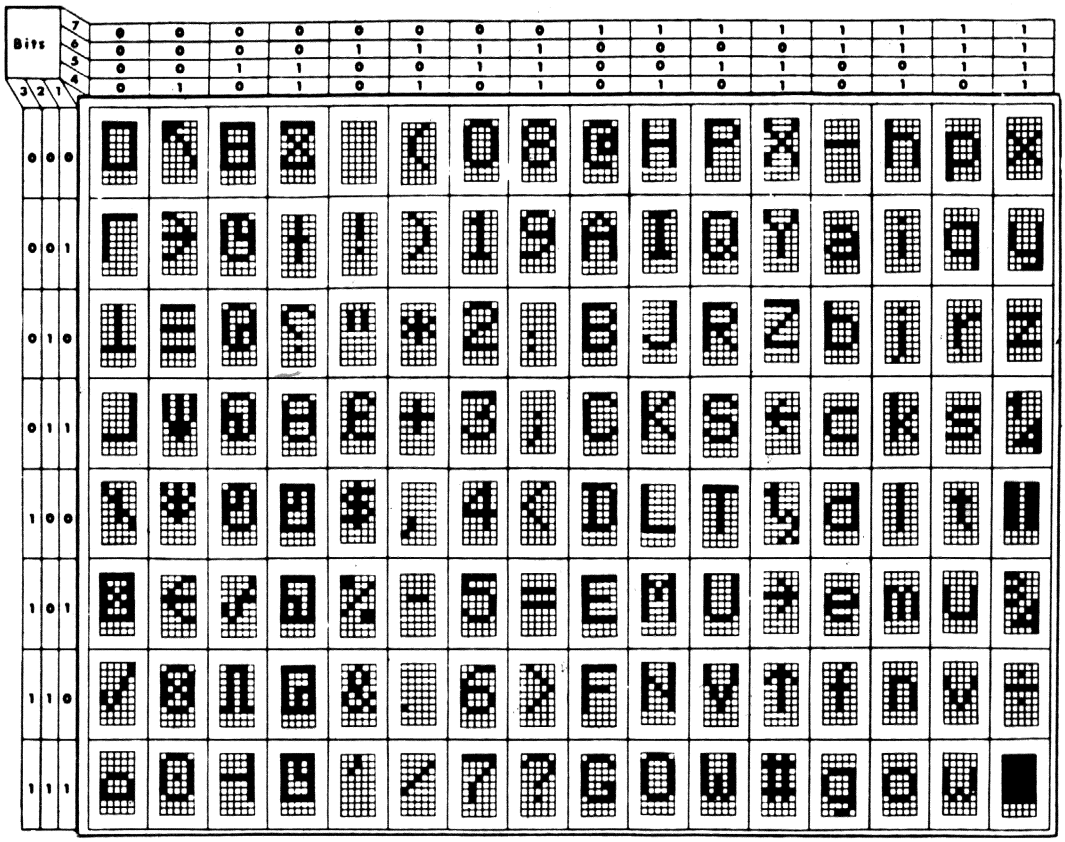
Column → / Row ↓	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Reading the column number and the row number for an entry in the Table will yield a Hexadecimal code for the character concerned, for example the character G has code 47 hex (71 decimal).

The control codes in columns 0 and 1 all have standard ASCII Mnemonic names such as NUL, DLE, FF. Research Machines refers to these characters as CTRL F etc. If column 0 is transposed with column 4 and column 1 with column 5, the corresponding letter code can be found, e.g. CTRL F is the same as ACK, CTRL Z is the same as SUB, CTRL L is the same as FF and so on.

Control codes, when sent to display devices such as printers or VDU screens often result in some special action and are not normally printed. Different devices will interpret the same control codes in different ways. For instance, CTRL O sent to the 380Z screen will cause it to discard output, but when sent to some printer may cause it to revert to single width characters. For example

Character Set-40 character screen



SN74S262N ROM CHARACTER FORMAT

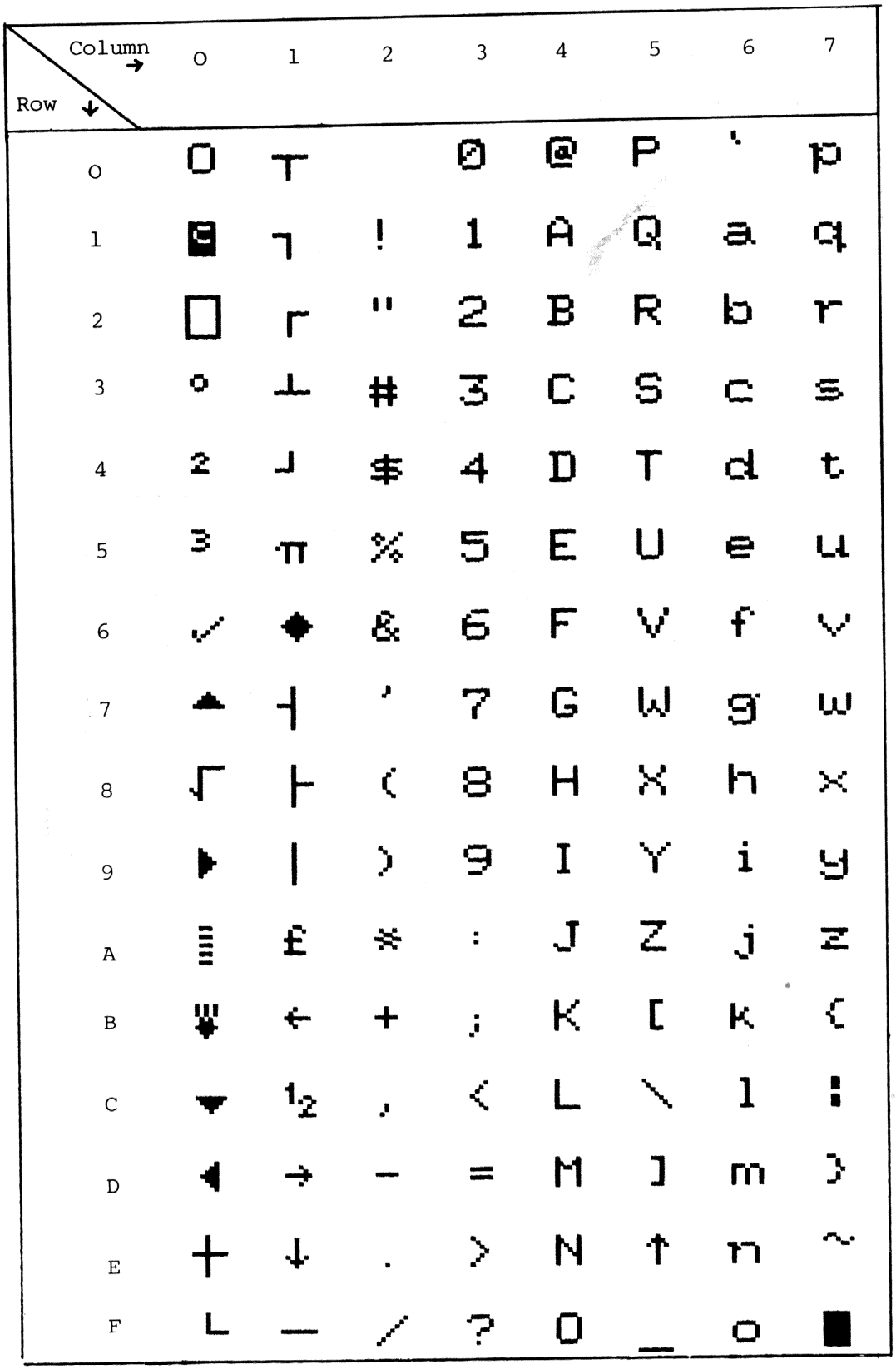
The character set output on the 40-character 380Z screen is shown above.

The following hexadecimal codes have different representations on the screen from the ASCII standard:-

23, 5A-60, 7A-7F.

In addition COS distinguishes between zero and the letter 'O' and does not display any character for codes 00 to 20 hex and code 7F hex (DEL). The COS mapping affects only the shape of the characters on the screen and not their internal representation. If sent to a printer they will normally produce the correct shapes (depending on how the printer character set maps onto the ASCII standard).

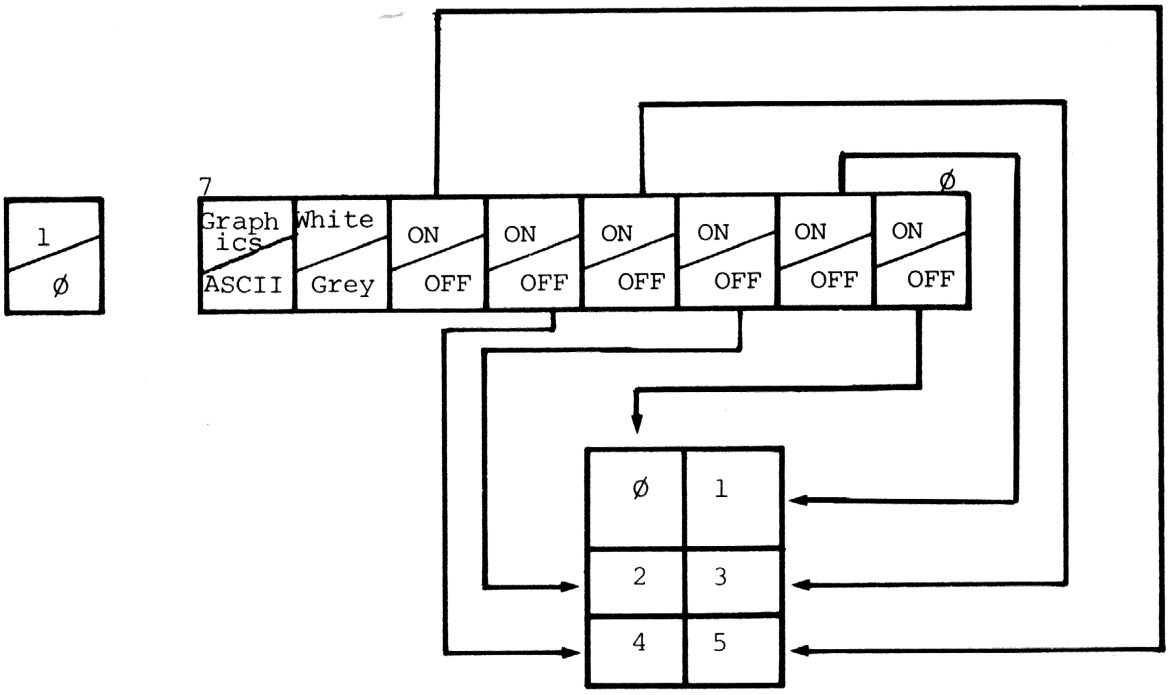
Character Set - 80 character screen



The characters in Columns 0 and 1 will only be printed on the screen by sending a suitable ESCape sequence since they will normally be interpreted by the screen as meaning certain cursor movements.

Graphics Character Set

Characters 80 to FF hex are the standard Teletext graphics - codes 80 to BF hex are displayed in grey and C0 to FF hex are displayed in white. The graphics character set displayed is tabulated in the User Guide (at the end of Chapter 4). Note that on COS 4.0 the character displayed for 80H-FFH are user definable.



INDEX

RESEARCH MACHINES

MICROCOMPUTER SYSTEMS