LINK 480Z INFORMATION FILE

PN 10939, Rev.1

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 products software it describes without notice. Research Machines Limited endeavour to ensure the accuracy of this document and to ensure that the products described perform correctly according to their descriptions. However, Research Machines Limited do not accept liability for the consequences of any error or omission.

Additional copies of this publication may be ordered from Research Machines Limited at the address above. Please ask for the 'LINK 480Z Information File, PN 10939'.

Reader's Comments

To assist Research Machines to improve the quality of its technical publications and its computer and software products we would like to hear from users on their experience with using them.

Comments referring to this manual 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 about this manual should be sent to:

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

Reports referring to difficulties in using any of the products described in this manual should include:

- a written description of each step taken and each key or command you have used
- details of the hardware configuration, and the version numbers of the software in use
- the nature of the error

and should be sent to:

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

CONTENTS

CHAPTER 1: INTRODUCTION 1.1

LINK 480Z Documentation 1.1
The LINK 480Z Information File 1.1

CHAPTER 2: HARDWARE SPECIFICATION 2.1

CHAPTER 3: PERIPHERAL INTERFACES 3.1

Cassette Interface 3.1
Use of Cassette Interface 3.1
Cassette Filing System 3.1
General Principles 3.1
File Structure 3.2
Device and File Specification 3.3
File Format 3.3
Cassette Hardware 3.4
Output to Cassette 3.4
Input from Cassette 3.4
Cassette Recorder Motor Control 3.4

Serial Interfaces 3.5
Driving a Serial Interface (SIO-4) 3.6

Parallel Interface 3.7
Comparison with 380Z User I/O Port 3.8
Example of use 3.8

Beeper 3.8
Analogue Output 3.9
Accessories 3.9
Switches 3.9
Connection Information 3.11

CHAPTER 4: HIGH RESOLUTION GRAPHICS 4.1

General Information 4.1
Resolutions 4.1
Look-up Table 4.2
Colour 4.2
Video Outputs 4.2

Detailed Information 4.3
Selection of Operating Mode 4.4
Accessing the Screen Memory 4.4
Calculation of the Byte Address 4.5
Mask Generation 4.5
Modifying Screen RAM 4.5
Accessing the Look-up Table 4.6

Controlling the Video Outputs 4.8
Colour Graphics 4.8
Opening the Screen 4.9

CHAPTER 5: HARDWARE NOTES 5.1

Memory Maps 5.1
64K Byte System 5.2

Ports 5.3
Control/Status Ports 5.4
HRG Control Ports 5.6

Interrupt Priorities 5.7
CTC Usage - Main Board 5.8
General 5.8
Channel 0 5.8
Channel 1 5.8
Channel 2 5.9
Channel 3 5.9
Limitations Imposed by the CTC Usage 5.9

CTC Usage - Option Board 5.9
General 5.9
Channel 0 5.10
Channel 1 5.10
Channel 2 5.10
Channel 3 5.10

Keyboard Interface 5.11
Video Display 5.11

APPENDIX A: Character Sets A.1

APPENDIX B: Circuit Diagrams B.1

INDEX I.1

CHAPTER 1

INTRODUCTION

LINK 480Z MANUALS

The following manuals are provided by Research Machines to help you set up and use a LINK 480Z:

- LINK 480Z User Guide: this manual explains how to set up and use your system and its facilities.
- Programming Manuals: each item of software that you purchase is supported by one or more manuals which explain how to set up and use the software provided.

THE LINK 480Z INFORMATION FILE

This manual contains technical specifications of various features of the 480Z hardware including for example, the various interfaces that are supported by the 480Z.

Chapter 2 gives an overall description of the 480Z hardware. Chapter 3 describes the various interfaces which are available on the 480Z. Chapter 4 provides a description of the hardware aspects of the high resolution graphics facilities, and Chapter 5 describes the other hardware in greater detail.

CHAPTER 2

HARDWARE SPECIFICATION

LINK 480Z - SPECIFICATION

Processor:	Z80A 4MHz
Memory:	64K (58K user) RAM. 8K ROM monitor (ROS). 20K BASIC in ROM. 4K Network firmware.
Keyboard:	65 Key with n-key rollover. Includes 4 cursor control keys and 4 special function keys (all 8 of which are user-definable), upper/lower case, caps lock and repeat key.
Display:	40-character by 24-row, or optional 80-column mode, software selectable down to 40-column mode. 128-character set plus 80 x 72 or 160 x 72 low-resolution graphics in two shades of grey. Video and UHF outputs are provided.
External storage:	1200 bps and 300 bps cassette interface, optional external dual cassette controller.
Tone generation:	Miniature internal loudspeaker driven by an optional digital to analogue converter. Also simple 'beep' facility. Internal volume control/disable.
Reset button:	On rear panel - can be disabled internally.

Input/Output

Parallel I/O:	One 8-bit parallel input/output port with 4 line strobe and acknowledge lines. Also suitable for driving Centronics standard parallel printers.
Serial I/O:	Three serial interfaces fitted as standard. 1. Low speed 110-2400 baud with busy line. 2. High speed 110-9600 baud with full 4-wire handshake. 3. Synchronous 800k bps serial interface for network operation. (Needs network transceiver for connection to the network cable.)

Paddle and pushbutton input: DIN socket which can be used to determine the position of two paddles (or a joystick) and two pushbuttons.

Analogue output: (Optional) User can link output internally to produce beep only. Digital to analogue converter is then available externally on rear panel socket.

ROM: ROM expansion sockets are available which will allow for expansion.

Option Board

Graphics:

- Three major modes:

  640 x 192 monochrome
  320 x 192, 4 colours per point
  160 x 96, 8 colours or 16 shades of grey per point (2 pages)

- Software definable colour look-up table
- Video output, UHF output and TTL RGB output
- Graphics and text output may be displayed separately or be superimposed under software control
- The Graphics memory is separate from CPU RAM, so the full address range of the CPU may be used

Plug-in options

CTC: Real-time clock and interrupt controller

Hardware Floating point (Maths Chip) (Under development): Plug-in hardware floating point option.

Further Expansion of System

- 256K bytes of RAM: Use of 64K RAMs enables the user to have 256K of RAM. The user-visible RAM is 58K bytes (out of 64K). The remaining 192K is available as backing data store.
- Network Operation: To connect to a Research Machines network, a transceiver unit is needed to connect to the network cable. To fit this unit, the 480Z must be returned to Research Machines Ltd.

Rear Panel Connections and Controls

i) Basic system row:

Type	Function
Phono	UHF out
Push button	Reset
6-way DIN	Video out
7-way DIN	Accessory socket, paddles, push buttons, D/A analogue output
7-way DIN	Cassette I/O
25-way D type	Parallel I/O (User I/O)
8-way DIN	RS232 interface (SIO-2)
8-way DIN	RS232 interface (SIO-4)
10-way DIL switch	Network address, reset disable, loudspeaker disable
BNC*	Internal network transceiver

*only active if option installed

ii) Option Board row*:

8-way DIN*	Colour TTL Red, Green, Blue output

*only active if option installed

General:

Power consumption	30 Watts, approximately.
Dimensions	Width 532mm Depth 331mm Height 95mm

CHAPTER 3

PERIPHERAL INTERFACES

CASSETTE INTERFACE

Use of Cassette Interface

The cassette recorder interface circuits comprise a tone generator with a band pass filtered output which is capable of generating approximately sinusoidal waveforms at frequencies in the range ca. 1200Hz to 2400Hz and also some input shaping and filtering circuits which allow the frequency and phase of the incoming signal to be determined. There are also two output lines which are used to control a dual cassette controller, thus allowing the computer to control the cassette recorder motors.

The cassette system is capable of operating in two different modes. In slow mode the data transfer rate is nominally 300 baud while in fast mode the transfer rate is nominally 1200 baud. All cassette tapes supplied by Research Machines are now recorded in fast mode to increase the speed of loading and saving programs.

In slow mode a logic '1' is recorded as 8 cycles of 2400Hz while a '0' is recorded as 4 cycles of 1200Hz. In fast mode a '1' is 2 cycles of 2400Hz while a '0' is 1 cycle of 1200Hz.

Each byte is recorded on the tape with one start bit (a '0') followed by 8 data bits (least-significant bit first) and two stop bits (recorded as 1's). Any gap between bytes will be recorded at 2400Hz and so will behave as a continuation of the stop bits of the previous byte.

Note that although the frequencies used are nominally 1200Hz and 2400Hz, they are actually about 1115Hz and 2230Hz. This is in order to provide compatibility with the 380Z.

Cassette Filing System

General Principles

The Cassette Filing System (CFS) supplied with the cassette-based 480Z 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.

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-purposeprograms. Specialist applications may get away with 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 or 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.

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 the 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 file name 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.

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 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.

Cassette Hardware

Output to Cassette

The output signal is generated by one channel of the Z80 CTC on the 480Z main board. The output from the CTC is actually in the form of a train of very short pulses. This passes through a stage which converts it to a square wave of half the frequency. This is then passed through an analogue band pass filter in order to produce a closer approximation to a sine wave output. This signal is then fed to the output socket at a low level, suitable for feeding into the microphone socket of a tape recorder.

Because the output frequency is generated by a CTC any changes of the output frequency will be neatly synchronised by the CTC itself and hence there is no necessity for the processor to synchronise the transitions itself. However because the output of the CTC is connected to a divide-by-two stage the phase of the output signal will not be known. This problem is overcome by allowing the processor to read the output of the divide-by-two stage.

Input from Cassette

The signal from the cassette recorder on play-back should be at a high level (ca. 1.5V rms) and will normally be taken either from the recorder DIN connector or from an external loudspeaker or earphone socket. After shaping and filtering this signal can then be read by the processor as two bits (0 and 1) in status port 2. Bit 1 of this port is the volume bit and is used for setting the playback volume of the recorder. The volume is correct when there are occasional transitions of this bit from high (its normal state) to low. Bit 0 of the port is the signal from which information is actually recovered. This line is also connected to one of the inputs of the CTC so that an interrupt can be generated on transitions.

The input data is decoded by measuring the time between successive edges of the same polarity and hence determining whether the intervening cycle was 1200Hz or 2400Hz. If the tape being read was in slow mode then there are several such cycles for each bit and a degree of signal averaging is performed on the cycles to provide extra noise immunity and protection from drop-outs etc. In fast mode these protections are not possible and this is the major cause of the lower reliability of fast mode.

It is important that the polarity of the input signal is correct since otherwise, whenever a transition occurs, a cycle consisting of 1/2 cycle at 1200Hz and 1/2 cycle at 2400Hz will be measured. If this occurs in slow mode then signal processing will cope with the problem and the only effect will be to lower reliability. However in fast mode the system will cease to work at all. Thus it is important that the cassette recorder used should have no phase inversion between its input and its output. All cassette recorders provided by Research Machines Limited will satisfy this requirement.

Cassette Recorder Motor Control

Bit 1 and 2 of control port 2 are used to control the cassette recorder

motors. By convention bit 1 controls recorder 1 which is connected for reading (replay) while bit 2 is used for recorder 2 which is used for writing (record). Separating the read and write functions in this way simplifies the actions required of the user (and the software). If no dual cassette controller is being used then one cassette recorder provides both read and write functions with no motor control. If, however, a dual cassette controller is being used but is switched to 'single' mode then only one cassette recorder is necessary but the computer still has motor control.

SERIAL INTERFACES

The 480Z has two RS232 serial interface ports, the SIO-2 and the SIO-4, the names being chosen for compatibility with the 380Z.

The SIO-4 is a high speed bi-directional interface (up to 9600 baud) with full modem handshake and is implemented using channel B of the Z80 SIO chip. ROS supplies sufficient software to allow for its use as an input, output or bi-directional port, the only limitation being that send and receive baud rates must be the same.

The SIO-2, on the other hand, is implemented in software and so is rather slower than the SIO-4, maximum 2400 baud. ROS supports the use of the SIO-2 as an output device (e.g. for driving a printer) but not as an input device. The SIO-2 has a single input handshake line which is intended to be connected to a printer "READY" line. It is equivalent to the CTS line on the SIO-4 (see below).

The baud rates available for these SIOs are:

110 baud	- baud code 0
300 baud	- baud code 1
600 baud	- baud code 2
1200 baud	- baud code 3
2400 baud	- baud code 4
4800 baud	- baud code 5 (SIO-4 only)
9600 baud	- baud code 6 (SIO-4 only)

The baud rates may be set up by means of the BASIC PRINTER command, ROS O command or by a call to EMT SETLST with the printer code (2 for the SIO-2, 4 for the SIO-4) in A and the baud code, defined above, in register E. This call also has the effect of setting EMT LPOUT's transfer vector to the appropriate driver routine. Thus if, for example, it is desired to use the SIO-2 for a printer and the SIO-4 for some other purpose then the SIO-4 baud rate should be set by a call to SETLST prior to setting up the SIO-2 by a second call to SETLST.

The SIO-4 supports the following handshake lines: CTS, DCD (also known as DSR), RTS, and DTR. The function of each of these lines is described briefly overleaf.

CTS (Clear To Send) is an input to the 480Z. It functions as a transmitter enable. Thus the SIO-4 will not transmit any data unless this line is active. It thus serves as a printer "READY" line since no characters will be sent until the printer is ready.

DCD (Data Carrier Detect) or DSR (Data Set Ready) is an input to the 480Z. It functions as a receiver enable line. Thus the SIO-4 will ignore all input data unless this line is active.

RTS (Request To Send) is an output from the 480Z. This line is intended to indicate to the peripheral that the 480Z has a character or characters ready for transmission. Thus if two 480Zs were to be connected via their SIO-4s this line would be connected to the DCD line on the other machine.

DTR (Data Terminal Ready) is an output from the 480Z. The line goes active when the 480Z is ready to receive a byte and goes inactive when the SIO-4's receive buffer becomes full. Thus this line functions in the same way as a printer's "READY" line.

It should be noted that the CTS and DCD pins have pull-up resistors connected to them so that if nothing is connected to them they float into the active state and thus allow the transmitter and receiver to function correctly.

Driving a Serial Interface (SIO-4)

Terminal mode software for the 480Z (fully position independent) is coded in ROS 1.1 and later versions and is activated using the ROS command T:

```
T480:   EMT   KBDTL        ; is there a character on the keyboard?
        JR    Z, T480A     ; jump if not
        EMT   KBDW         ; else get the character...
        EMT   LPOUT        ; ... and output it

T480A:  EMT   S4KTL        ; is there a character on the input line?
        JR    Z, T480      ; jump if not
        EMT   S4KIN        ; else get the character...
        EMT   OUTC         ; ... display it ...
        JR    T480         ; ... and loop.

KBDTL   EQU   31
KBDW    EQU   33
LPOUT   EQU    5
S4KTL   EQU   47
S4KIN   EQU   48
OUTC    EQU    1
```

PARALLEL INTERFACE

The basic 480Z has an 8-bit parallel interface known as the User I/O port. This port is mapped as a Z80 input/output port at I/O address 1DH. Reading a byte from this address, for example with the instruction 'IN A,(1DH)', causes the input data to be read. Outputting a data byte using the instruction 'OUT (1DH),A' results in that byte being latched on the User I/O output lines.

The port will, typically, be used for driving a parallel printer, possibly of the 'Centronics' variety. However the port may also be used to drive simple home-built interfaces. For this purpose a +5V power supply is available on the connector. Only low currents should be drawn from this pin however.

In normal use any data written to port 1DH appears latched on the output lines and an input from port 1DH results in a read of the current state of the input lines. However in order to allow for interface to devices having handshake lines there are also some extra handshake pins on the connector.

There is an input strobe line on the connector. While this line is at logic '0' (it will float to logic '0' if left disconnected) the data read by an input instruction will be the data currently on the input pins. However if the strobe line is taken high then the data on the input lines at that time will be latched and all subsequent reads of the input port (until the strobe is again taken low) will read the latched data. Taking the strobe line high also causes the user I/O ready line to go to a logic '0' thus indicating to the peripheral that the computer is no longer ready to accept input data. This line goes back to a logic '1' (i.e. ready) when the computer reads the user I/O port. There is also a ready line which can be read by the computer. This, however, is of the opposite sense. Thus if the bit is a logic '1' then there is a data byte latched into the user I/O input port ready to be read.

In addition to the above 'hardware' handshake lines there are also three spare uncommitted output pins and one uncommitted input pin which can be used for handshaking purposes (or as extra data bits etc.).

All of the user I/O handshake lines are in port 19H. If the uncommitted output lines are used it is important to set the remaining bits of the port according to the value in the mask which ROS maintains at location 0FF01H. Furthermore, this mask should be updated with the new value output. Failure to utilise and maintain this mask will result in unpredictable operation and almost certainly will 'crash' the machine.

All input lines on the user I/O connector are Schmitt-triggered low-power Schottky devices. The strobe input is also connected to a 1k resistor whose other end is connected to ground. All output lines on the user I/O connector are driven by low-power Schottky buffer chips (74LS374 and 74LS244). For more information on these chips see the appropriate data sheets.

Comparison with 380Z User I/O Port

The 480Z User I/O port is similar in external appearance to the User I/O port on the 380Z. However, it is an I/O mapped port rather than a memory mapped port. It is mapped as I/O port 1D hexadecimal for both input and output. Like the memory mapped port a mask must be kept for the output byte. This is now kept at FF00 hexadecimal.

The pin-out of the User I/O port remains the same as the 380Z for the 8 bits in, the 8 bits out, the +5V line, and the 0V line. The -12V and +12V lines are not available.

It should be noted that software serial interface (serial I/O 2) is not connected to the User I/O port, unlike the 380Z SIO-2.

Example of use

To set bit 3 of the output of the User port high the following code is appropriate:

```
UMASK     EQU       0FF00H
UPORT     EQU       1DH

LD        A,(UMASK)
SET       3,A
LD        (UMASK),A
OUT       (UPORT),A
```

The equivalent BASIC code is:

```
10   UM = &FF00
20   UP = &1D
30   A  = PEEK (UM)
40   A  = (A OR 8)
50   POKE UM,A
60   OUT UP,A
```

BEEPER

The 480Z is capable of producing noises in two ways. Firstly there is a single bit in a port which is connected to the audio amplifier and can be used for producing single tones. This is the method used by ROS to generate beeps. There is an escape sequence that permits user programs to set the tone and duration of this beep. This beep cannot be disabled externally.

The second way is by using optional analogue output, as described overleaf.

ANALOGUE OUTPUT

To produce an analogue output, an 8-bit value, N, should be written to control port 3 (1BH). The output produced is approximately N*10 millivolts. Thus the range of the output is from 0 to 2.55 volts, approximately.

The digital to analogue converter is also connected to the audio amplifier and can be used to produce reasonable approximations to a sine wave, or indeed any waveform. However in order that the digital to analogue converter may also be used to produce an analogue output (available on the accessories socket) without making a noise in the process this output may be disconnected from the audio amplifier by setting the speaker disable switch to the up position. It should be noted that there is no software in ROS to support the use of the digital to analogue converter.

ACCESSORIES

Provision has been made on the 480Z for the connection of an external unit containing a joystick (or two paddles) and two push buttons. Initially, Research Machines are not providing such an accessories unit, or any software to support its use.

Changing bit 4 of control port 2 (1AH) from 0 to 1 triggers two monostables. The joystick or paddles contain potentiometers which affect the width of the pulse generated. This is approximately 1 millisecond + r*0.2 millisecond where r is the resistance in kOhms. The outputs of the monostables can be read from bits 4 and 5 of status port 2 (1AH).

The two push-buttons are assumed to provide low outputs when pressed. These can be sensed via bits 6 and 7 of status port 2 (1AH).

SWITCHES

The basic 480Z has externally accessible switches for the following purposes:-

Reset disable (R)
Speaker disable (S)
Network address (NET ADDR)

They are accessible through a hole in the rear panel, and their position may be changed using a small screwdriver, for example:

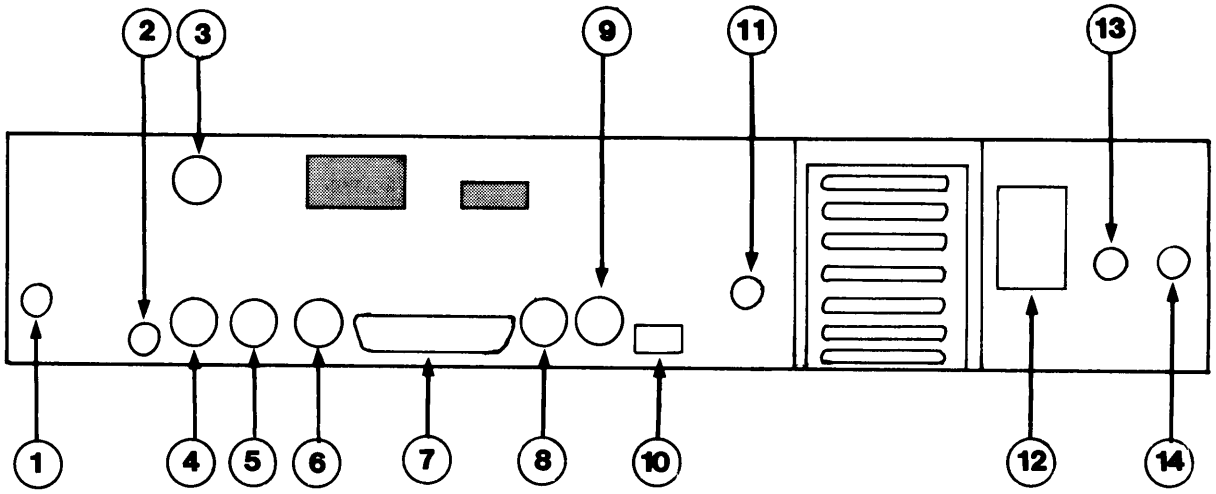
R: Reset Disable Switch: when this switch is in the up position the rear panel reset button of the 480Z is disabled. Under these conditions there is no way of resetting the machine without re-enabling the reset disable switch.

S: Speaker Disable Switch: when this switch is in the up position the output of the digital-to-analogue converter is disconnected from the audio amplifier. It does not disable the 'beep' which can be produced by ROS.

NET ADDR: Network Address Switches: these 8 switches are to be used to set the station address for a 480Z on a network. Switches are arranged with the least significant address bit at the right-hand end when viewed through the rear panel. A switch in the up position corresponds to a 1. Details of setting up network addresses will be found in the Network Manager's Guide.

CONNECTION INFORMATION

The layout of the sockets, switches and mains cables on the rear panel of the LINK 480Z is shown in the diagram below.

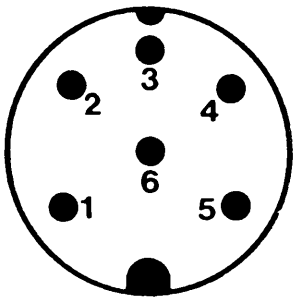


Connection Information

1.	Modulator UHF (TV)	
2.	RESET button	
3.	TTL RGB	
4.	Video	(A)
5.	Accessories	(B)
6.	Cassette	(C)
7.	Parallel I/O	(D)
8.	Serial I/O 4	(E)
9.	Serial I/O 2	(F)
10.	Network Address, Speaker, and Reset Disable Switches	3.10
11.	Network Interface Coaxial	
12.	Mains ON/OFF	
13.	Fuse	
14.	Mains Cable	

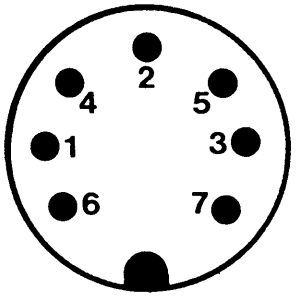
(A) VIDEO SOCKET — 6-PIN DIN SOCKET POSITION 4*

(1)	VIDEO SIGNAL	
(2)	GROUND	
(3)	FRAME SYNC	
(4)	LINE SYNC	WIRE FRAME
(5)	AUDIO OUT	VIDEO MONITORS ONLY
(6)	WIREFRAME VIDEO	



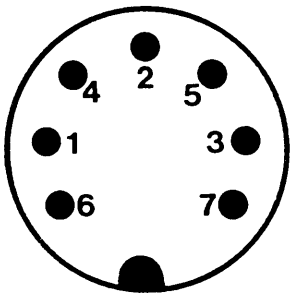
(B) ACCESSORIES SOCKET 7-PIN DIN SOCKET POSITION 5*

(1) ANALOGUE OUTPUT
(2) JOYSTICK 1
(3) +5V
(4) GROUND
(5) BUTTON 1
(6) BUTTON 2
(7) JOYSTICK 2



(C) CASSETTE PORT 7-PIN DIN SOCKET POSITION 6*

(1) SIGNAL FROM COMPUTER TO CASSETTE RECORDER
(2) RECORDER 1 MOTOR CONTROL
(3) +5V
(4) GROUND
(5) N.C.
(6) SIGNAL FROM CASSETTE RECORDER TO COMPUTER
(7) RECORDER 2 MOTOR CONTROL



*Positions refer to rear panel diagram, above.

(D) PARALLEL I/O SOCKET 25-WAY D TYPE POSITION 7*

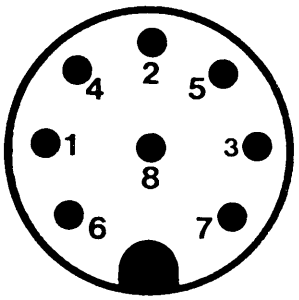
(1) DATA BIT 0 IN
(2) DATA BIT 2 IN
(3) DATA BIT 4 IN
(4) DATA BIT 6 IN
(5) DATA BIT 0 OUT
(6) DATA BIT 2 OUT
(7) DATA BIT 4 OUT
(8) DATA BIT 6 OUT
(9) +5V
(10) INPUT STROBE
(11) HANDSHAKE OUTPUT 1
(12) HANDSHAKE OUTPUT 2
(13) HANDSHAKE OUTPUT 3
(14) DATA BIT 1 IN
(15) DATA BIT 3 IN
(16) DATA BIT 5 IN
(17) DATA BIT 7 IN
(18) DATA BIT 1 OUT
(19) DATA BIT 3 OUT
(20) DATA BIT 5 OUT
(21) DATA BIT 7 OUT
(22) GROUND
(23) GROUND
(24) INPUT READY
(25) HANDSHAKE INPUT 1

(E) SERIAL I/O 4

8-PIN DIN SOCKET POSITION 8*

(1) REQUEST TO SEND
(2) DATA TERMINAL READY
(3) TRANSMITTED DATA
(4) GROUND
(5) CLEAR TO SEND
(6) N.C.
(7) DATA CARRIER DETECT
(8) RECEIVED DATA

PRINTER BUSY LINE

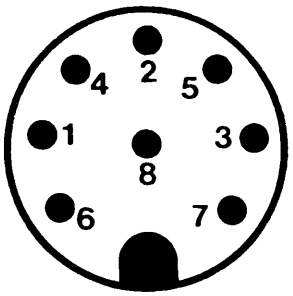


(F) SERIAL I/O 2

8-PIN DIN SOCKET POSITION 9*

(1) N.C.
(2) N.C.
(3) TRANSMITTED DATA
(4) GROUND
(5) CLEAR TO SEND
(6) N.C.
(7) N.C.
(8) RECEIVED DATA

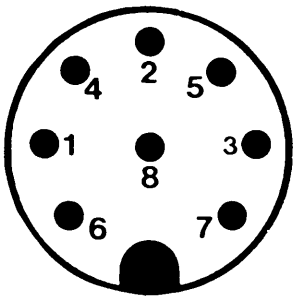
PRINTER BUSY LINE



(G) TTL RGB

8-PIN DIN SOCKET POSITION 3

(1) AUDIO
(2) MIXED SYNC -
(3) RED
(4) GROUND
(5) BLUE
(6) GREEN
(7) LINE SYNC -
(8) FRAME SYNC -



CHAPTER 4

HIGH RESOLUTION GRAPHICS

GENERAL INTRODUCTION

The option board for the 480Z gives the machine a high resolution graphics capability. These graphics are of the 'pixel' type. Thus the screen is divided into a regular rectangular array of dots (pixels) each of which can be controlled independently. In this respect the high resolution graphics is similar to the low resolution ('Teletext' style) graphics which are a feature of the basic 480Z. With the high resolution graphics, however, an ordinary text display can be superimposed on the graphics such that, for example, a text character actually overlaps a graphics line. This is not possible with the low resolution graphics where, within a character cell, graphics and text are mutually exclusive. The high resolution graphics display is limited to approximately the top 80% of the screen (actually the top 19.2 character rows). This allows the use of a four line scroller for the text display which will not overlap the graphics area and hence will not interfere with any text in the graphics area, e.g. labels on graph axes.

Resolutions

The high resolution graphics can operate in one of three resolutions. These are (in terms of pixels):

Medium resolution	160	(horizontal) by 96 (vertical)
High resolution	320	(horizontal) by 192 (vertical)
Extra high resolution	640	(horizontal) by 192 (vertical)

The simplest of these three modes is the extra high resolution mode in which only one bit of the high resolution graphics screen memory is available to store each pixel on the screen. Thus each pixel can be either on (white) or off (black).

In high resolution mode there are two bits of the memory per pixel. This allows the possibility of four logical intensities (0,1,2,3) for each of the pixels. Each pixel can be set to any of these four intensities independently of any other pixel.

In medium resolution mode there are four bits available per pixel and hence 16 logical intensities are possible and, as with high resolution mode, each pixel may be set independently of the others.

In fact only half of the available graphics memory is required to store a medium resolution 'picture' at four bits per pixel. Thus there is sufficient memory for two such pictures. These two pictures may be displayed and updated independently of each other and, further, a picture

may be updated while the other picture is being displayed. This introduces the possibility of 'drawing' a picture while the other is being displayed and then, when it is complete, swapping the two pictures. Thus the display changes instantaneously even though the picture may have taken considerable amounts of time and computing to generate and draw.

Look-up Table

In medium and high resolution modes (but NOT extra high resolution mode) the 16 or 4 logical intensities available for each pixel do not correspond directly to the physical intensity displayed on the screen. Instead there is a small (16 byte) look-up table interposed. This has the effect that each of the available logical intensities may be set to any of the available physical intensities. In fact there are 256 intensities available ranging from 0 (black) to 255 (white).

Since the look-up table can be updated very rapidly this introduces the possibility of changing the colours of large areas of the screen simultaneously.

Colour

The above descriptions have assumed that the graphics output is displayed on a conventional monochrome monitor or television. However the LINK 480Z option board is also capable of driving a colour monitor of the TTL RGB variety. These monitors allow the generation of 8 colours, these being black, blue, red, green, magenta, cyan, yellow and white.

In extra high resolution mode all graphics display (and all text display) will be in black and white only. However medium and high resolution graphics will be displayed in colour. Although there are only 8 physical colours there are still 16 logical colours available in medium resolution and 4 available in high resolution. The colours are still controlled via the look-up table in the same way as for a monochrome display.

Video Outputs

A 480Z with an option board can be connected to a number of different types of screen. The UHF (phono) socket on the main board can be connected to a standard 625 line television and is tuned to approximately channel 36. Although either black and white or colour televisions will work when connected to this socket, the results, in terms of clarity of picture and readability of textual information, will probably be superior with a black and white set. Only monochrome output is given on the UHF output so that pictures will appear as black and white even on a colour TV set. It is not recommended that a television be used with the 480Z switched into 80-character display mode (if the 80-character option is installed) as, although no damage will be done, the display clarity is unlikely to be acceptable.

The video (DIN) socket on the main board can be connected to a monochrome video monitor of either the composite video variety (this includes all monitors currently supplied by Research Machines Limited) or the 'wire-frame' variety. All of the video outputs on the main board carry the same information.

The TTL RGB socket on the option board can interface to a TTL RGB colour monitor as described above. Note however that, owing to a certain amount of non-standardisation among the manufacturers of such monitors, it may be necessary to make a minor modification to the computer. However any monitor supplied by Research Machines Limited will work with an un-modified 480Z.

The display produced by the colour monitor will not necessarily contain the same information as that displayed by any monitor (or television) plugged into the main board.

The monitor connected to the main board will always show the text (and low resolution graphics) display and will normally display the high resolution graphics superimposed upon this. However it is possible, under software control, to inhibit the display of the high resolution graphics (although it will still be possible to update the high resolution screen while it is disabled so that when re-enabled it will show different information).

Similarly the colour monitor will always show the high resolution graphics but may have the text and low resolution graphics display disabled under software control. Again it is possible to update the text display while it is disabled.

DETAILED INFORMATION

This section contains information for those who wish to access the 480Z high resolution graphics from assembly language programs.

The 480Z high resolution graphics system is mapped as a group of four I/O ports, HRG ports 0 - 3, which completely control its function. A description of the bit assignments within these ports can be found on page 5.6.

Manipulation of these ports allows the following functions:

1) The operating mode of the graphics can be set to be either extra high, high or medium resolution and, in the last case, the page to be displayed can be selected.

2) The 16K bytes of screen memory can be accessed both for reading and writing so that pictures can be drawn to the screen, read back from the screen, or modified without having to maintain a copy of the screen contents in user RAM.

3) The look-up table which controls the mapping of logical intensities onto physical intensities (or colours) can be written to. It is not, however, possible to read the contents of this RAM and so it is necessary for the user to maintain a copy of its contents.

4) It is possible to control the video outputs so that the graphics display on the monochrome monitor may be blanked at will as may the graphics display on the colour monitor.

Selection of Operating Mode

The resolution of the display is controlled by two of the bits in HRG port 2. These two bits are high resolution (L) and extra high resolution (L).

Extra high resolution mode is selected by setting both of these bits to '0'.

High resolution mode is obtained by setting the high resolution bit to '0' and the extra high resolution bit to a '1'.

Medium resolution mode requires that both of these bits be set to 1. In medium resolution it is also necessary to select the page which is to be displayed. This function is performed by the Page number bit in HRG port 2. Setting this bit to a 0 results in the display of page 0 while page 1 is obtained by setting it to a 1. This bit is only relevant in medium resolution.

An invalid operating mode is obtained if the extra high resolution is set to a 0 and the high resolution bit is set to a 1 simultaneously. Although no damage results from selection of this mode it is recommended that it should not be used.

Accessing the Screen Memory

There are 16K bytes of screen memory of which about 15K bytes are actually used to store the screen information. This memory is arranged as 192 rows of 80 bytes per row. Each of these bytes of memory is then responsible for the generation of 2, 4, or 8 pixels (medium, high and extra high resolutions respectively).

In order to plot a single pixel on the screen at a known position it is first necessary to calculate the X and Y values of the address of the relevant byte, then to decide which bits of that byte to modify, and finally to read the byte, modify it accordingly and restore it to the screen. Each of these steps is described overleaf:

Calculation of the Byte Address

The Y Byte Address

In high and extra high resolutions the Y byte address is simply the Y co-ordinate of the point (in the range 0-191, 0 corresponding to the bottom row of the graphics screen) subtracted from 191.

For medium resolution the Y co-ordinate of the point (in the range 0-95) is doubled and subtracted from 190. The page number which is currently selected for update (0 or 1) is then added to this number.

The X Byte Address

The X byte address is calculated by dividing the X co-ordinate of the point by 2, 4 or 8 (medium, high or extra high resolutions respectively). The remainder from this division (X remainder) should also be noted as this gives the position of the pixel within the selected byte.

Mask Generation

In general, the user will want to read a byte, modify it, and write it back to the screen since this allows a single pixel to be modified.

Research Machines' software uses a set of two masks for this purpose. The byte is first 'ANDed' with one mask (the AND mask) and then 'XORed' with the second (the XOR mask). This allows bits within the byte to be unchanged, set, cleared or toggled in any combination. The masks are best chosen using look-up table techniques rather than calculation as this is both quicker and easier.

Suitable masks to allow a point to be plotted at position X rem(ainder) within a byte and of intensity dcba (medium), ba (high) or a (extra high) are given in the table.

X	Medium resolution		High resolution		Extra resolution	
rem	AND mask	XOR mask	AND mask	XOR mask	AND mask	XOR mask
0	11001100	00dc00ba	11111100	000000ba	01111111	a0000000
1	00110011	dc00ba00	11110011	0000ba00	10111111	0a000000
2	-	-	11001111	00ba0000	11011111	00a00000
3	-	-	00111111	ba000000	11101111	000a0000
4	-	-	-	-	11110111	0000a000
5	-	-	-	-	11111011	00000a00
6	-	-	-	-	11111101	000000a0
7	-	-	-	-	11111110	0000000a

Modifying screen RAM

In order to modify screen RAM it is necessary first to set up the byte address of the byte to be modified. This is done by writing the X byte address to HRG port 1 and Y byte address to HRG port 0. There are no

timing restrictions on these operations.

For general purpose use it is necessary to observe a single timing restriction when accessing screen RAM. This is that the RAM may be accessed ONLY during the line blanking period. The best way to meet this restriction is to wait until the line access bit in HRG port 2 goes active. There will then be sufficient time to read the byte, modify it with the previously calculated masks, and write it back to the screen. Suitable code to perform a complete read-modify-write cycle would be:

```
;   On entry H contains Y byte address
;            L contains X byte address
;            D contains AND mask
;            E contains XOR mask

;   Set up X and Y addresses

          LD        C,HRG0          ; Address of HRG port 0 (Y address)
          OUT       (C), H          ; Set Y address
          INC       C               ; Now points to X address port
          OUT       (C), L          ; Set X address

;   Now wait for line access to be inactive (low)
;   Note that interrupts must be disabled from here on

          DI                        ; Disable interrupts

LOOP1:    IN        A, (HRG2)       ; Status
          BIT       LINACC, A       ; Test line access
          JR        NZ,LOOP1        ; Jump if active

;   Now wait for line access bit to go active (high)

LOOP2:    IN        A, (HRG2)       ; Status
          BIT       LINACC, A       ; Test line access
          JR        Z, LOOP2        ; Jump if inactive

;   Now at start of line access so do Read Modify Write

          IN        A, (HRG3)       ; Read
          AND       D               ; AND in AND mask
          XOR       E               ; XOR in XOR mask
          OUT       (HRG3), A       ; Write byte back

;   Finished, re-enable interrupts

          EI
```

Accessing the Look-up Table

The look-up table has one byte of memory for each of the available logical intensities. Thus the logical intensity is used as the look-up RAM address

and the corresponding physical intensity is the data byte, 0 corresponding to black and 255 to white.

The look-up table can only be accessed during frame blanking. It is written to by writing the data byte to HRG port 0 and then the 4 bit address to the 4 least-significant bits of HRG port 2. The address is first written with the look-up write bit inactive, then again with it active and finally a third time with it inactive. It is important that after modifying HRG port 2 in this way the port should be restored to its initial value, and a copy should be kept in RAM, before the end of the frame blanking period.

Since it is not possible to read the look-up RAM it is advisable to maintain a copy of its contents in a 16-byte block of RAM. The entire contents of this copy can readily be copied into the look-up RAM during one frame blanking period. Suitable code for such a block update is given below.

```
;   On entry HL points to a 16-byte table.
;   The  first byte of this table should be the value of logical  intensity
;   0, the second logical intensity 1 etc.
;   Note that interrupts should be disabled during this routine

;   First wait for the start of frame blank

;   Wait for frame blank bit inactive (low)
          DI                          ; Disable interrupts

LOOP1:    IN          A, (HRG2)       ; Status
          BIT         FBLANK, A       ; Test frame blank bit
          JR          NZ, LOOP1       ; Jump if active

;   and now wait for it to become active (high)
LOOP2:    IN          A, (HRG2)       ; Status
          BIT         FBLANK, A       ; Test frame blank
          JR          Z, LOOP2        ; Jump if not active

;   Now in frame access so can output data

          LD          D, 0            ; D will be logical intensity
          LD          B, 16           ; Count of bytes to output

WRLOOP:   LD          A, (HL)         ; Value of physical intensity
          OUT         (HRG0), A       ; To data port
          LD          A, D            ; Logical intensity, write bit inactive
          OUT         (HRG2), A       ; to HRG port 2
          SET         LWRITE, A       ; Set write bit
          OUT         (HRG2), A       ; Output
          LD          A, D            ; Clear write bit
          OUT         (HRG2), A       ; Output it
```

```
        INC     D           ; Increment logical intensity
        INC     HL          ; Point to next physical intensity
        DJNZ    WRLOOP      ; And write the next byte (unless
                            ; finished)

;   Finished, re-enable interrupts

        EI
```

NOTE: All 16 entries of the colour look-up table should be set up even if less than 16 logical intensities are in use. For example, if high resolution mode is in use, 4 logical intensities are available so the colour look-up table should be loaded with 4 consecutive copies of the 4 logical intensities.

Controlling the Video Outputs

As mentioned elsewhere there is a feature of the 480Z which allows the high resolution graphics display on the monochrome video outputs from the main board to be suppressed. This is accomplished by setting the HRG inhibit bit in HRG port 2 to a '0'. This does not, however, inhibit in any way the modification of the graphics memory.

Similarly the output of textual information (and low resolution graphics) to the RGB TTL colour monitor socket on the option board may be suppressed by clearing the Video Inhibit bit of HRG port 2. This in no way inhibits the ability to update the VDU screen memory however.

Note that the Video inhibit and HRG inhibit bits are entirely independent of each other and any of the four possible combinations is meaningful.

Colour Graphics

Most of the above has assumed that a monochrome monitor is being used so that the physical intensities stored in the look-up table are actually the position of the desired colour in a grey scale. If however a colour monitor is being used then only 8 colours are available. There is one bit available to control each of the red, green and blue guns of the monitor. Setting one of these bits causes the appropriate gun to become active. The bits in question are:

Bit 4	BLUE
Bit 6	RED
Bit 7	GREEN

These bits have been chosen so that if the colour graphics are displayed on the monochrome monitor a reasonably sensible looking grey scale is obtained. The colours available are thus:

Physical Intensity	Colour
0	Black
16	Blue
64	Red
80	Magenta
128	Green
144	Cyan
192	Yellow
208	White

It should be noted that the physical intensities listed above are not the only ones which will produce these colours and that, for the purposes of the colour display, only bits 4, 6 and 7 have any significance and that all others are ignored.

Opening the Screen

It is possible to open the HRG screen memory for asynchronous access so that the timing restrictions normally imposed on memory accesses are lifted. This is done by clearing the open bit in HRG port 2. However before using this feature the following points should be noted:

1) The HRG screen memory is dynamic and must therefore be refreshed at least once every 2ms if its contents are to be retained. When the screen is opened the only refresh which occurs is that due to accessing the memory. In order to maintain refresh it is necessary to access at least the first 64 bytes of two rows whose Y addresses differ in bit 1 every 2ms.

2) The screen memory may be corrupted if the open bit is set or reset at any time other than during the line blanking period. Thus before setting or resetting this bit the program should wait for the line access bit in HRG port 2 to become active.

3) While the screen is open the HRG display is blanked.

In general it is probably not worthwile using the open bit for purposes other than clearing the screen (as in calls to "RESOLUTION" in BASIC).

CHAPTER 5

HARDWARE NOTES

MEMORY MAPS

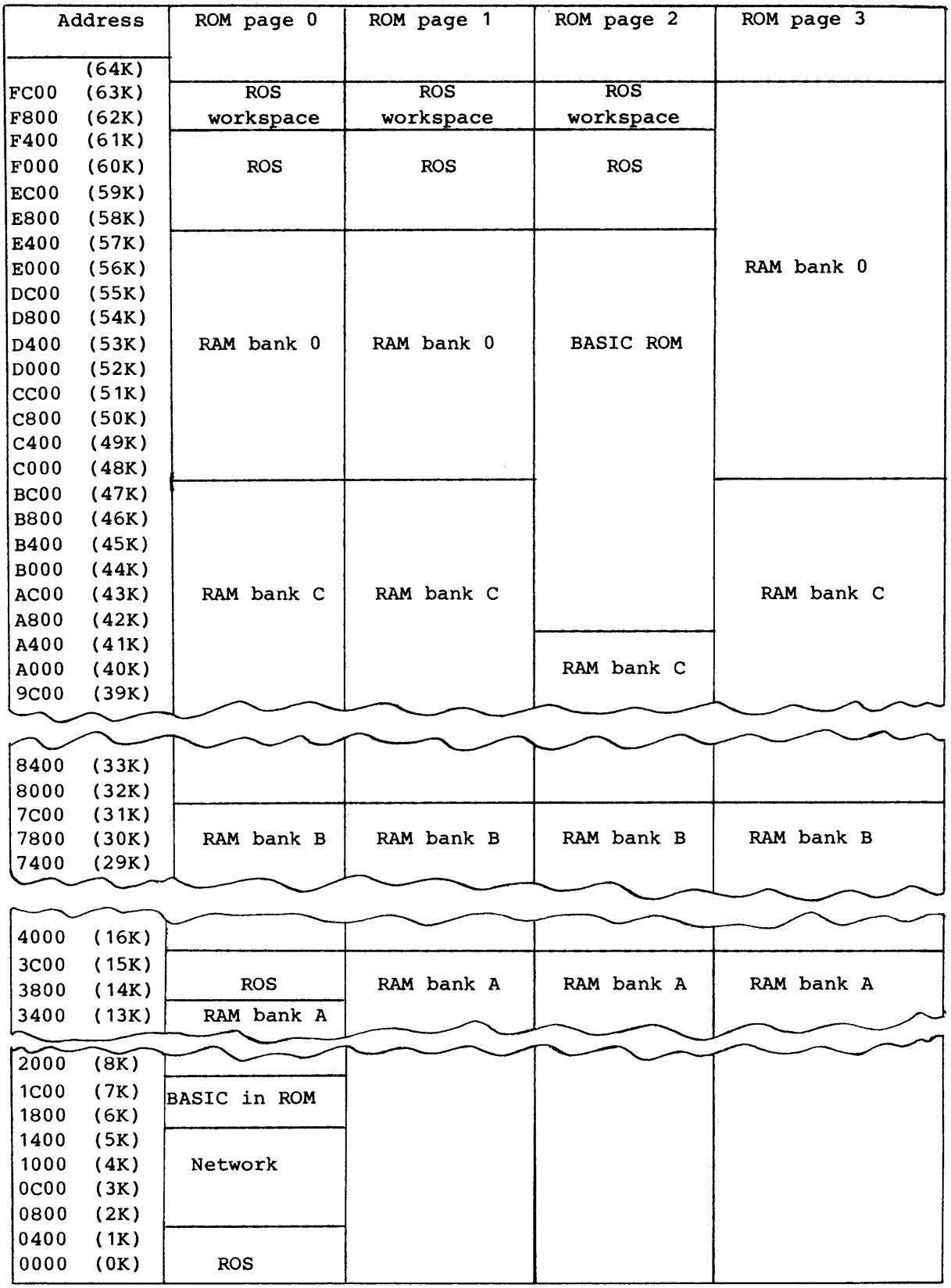
The detailed memory map of the LINK 480Z is liable to change between different versions of the firmware. The map given here is for a 64K machine operating under ROS 1.1 (and later versions) and which contains BASIC in ROM and network firmware.

There are 4 available mappings of ROM into the address space, called ROM pages 0 to 3, and controlled by a PROM. The mapping of RAM into the address space is controlled by the firmware.

The usage of the ROM pages is as follows:-

Page 0	:	initialisation, some ROS functions
Page 1	:	normal operation
Page 2	:	BASIC in ROM
Page 3	:	Not used

64K byte system



NOTES

If the 64K of RAM consists physically of four rows of 16K RAM chips then:-

RAM bank A is RAM bank 4
RAM bank B is RAM bank 8
RAM bank C is RAM bank 12

If, however, there is one row of 64K RAM chips then:-

RAM bank A is RAM bank 1
RAM bank B is RAM bank 2
RAM bank C is RAM bank 3

PORTS

Port Add.		Function
0-17H		VDU Port 0 corresponds to the top line of the screen Port 17H corresponds to the bottom line of the screen
18H		Control / Status Port 0
19H		Control / Status Port 1
1AH		Control / Status Port 2
1BH	(e)	Control / Status Port 3
1DH		Control / Status Port 5 (USERIO Port)
20H		Main Board CTC Channel 0 - SIO-4 and Cassette Input
21H		Main Board CTC Channel 1 - Cassette I/O and SIO-2
22H		Main Board CTC Channel 2 - Keyboard Interrupts
23H		Main Board CTC Channel 3 - 50Hz Interrupts (for repeat key)
24H		SIO Channel A (Network) Data Port
25H		SIO Channel B (SIO-4) Data Port
26H		SIO Channel A (Network) Control / Status Port
27H		SIO Channel B (SIO-4) Control / Status Port
28H	(b)	Maths Chip Data Port
29H	(b)	Maths Chip Control / Status Port
2AH	(b)	(Maths Chip Data Port)
2BH	(b)	(Maths Chip Control / Status Port)
2CH	(c)	Option Board CTC Channel 0 - IEEE Interrupts
2DH	(b)	Option Board CTC Channel 1 - Maths Chip Interrupts
2EH	(d)	Option Board CTC Channel 2 - Real-Time Clock
2FH	(d)	Option Board CTC Channel 3 - Real-Time Clock

30H	(c)	IEEE 0 Interrupt Status 0	Interrupt Mask 0
31H	(c)	IEEE 1 Interrupt Status 1	Interrupt Mask 1
32H	(c)	IEEE 2 Address Status	
33H	(c)	IEEE 3 Bus Status	Auxiliary Command
34H	(c)	IEEE 4	Address Register
35H	(c)	IEEE 5	Serial Poll Register
36H	(c)	IEEE 6 Command Pass Thru	Parallel Poll Register
37H	(c)	IEEE 7 Data Input	Data Output
38H	(a)	HRGPORT 0 DIL-Switch	Y Address Look-up Data
39H	(a)	HRGPORT 1	X Address
3AH	(a)	HRGPORT 2 HRG Status	Control Look-up Address
3BH	(a)	HRGPORT 3 Data Input	Data Output

(a) Only active if Option Board installed
(b) Only active if Option Board and Maths chip option installed
(c) Only active if Option Board and IEEE option installed
(d) Only active if CTC Option installed
(e) Only active if DAC Option installed

Control/Status Ports

Control / Status Port 0 (18H , 24 decimal)

Read = Main Board DIL-Switch
Write = RAM mapping Look-up table

ROS does not maintain a mask for Control Port 0

Read	Bit	Write
DIL-switch 3	7	-
DIL-switch 4	6	-
DIL-switch 5	5	-
DIL-switch 6	4	-
DIL-switch 7	3	Data bit 3 (MA17)
DIL-switch 8	2	Data bit 2 (MA16)
DIL-switch 9	1	Data bit 1 (MA15)
DIL-switch 10	0	Data bit 0 (MA14)

On writing to port 0 the two least-significant bits of the B register contain the address within the look-up RAM to which the data is written.

Control / Status Port 1 (19H , 25 decimal)

Read = Status Port 1
Write = Control Port 1

ROS maintains a mask for Control Port 1 at 0FF01H

Read	Bit	Write
Not used	7	Not used
Not used	6	Keyboard reset (L)
Cassette output bit	5	S'ware USER I/O handshake 3
H'ware USER I/O ready	4	S'ware USER I/O handshake 2
S'ware USER I/O handshake	3	S'ware USER I/O handshake 1
Keyboard ready (L)	2	Single-step enable (H)
Frame blank (H)	1	Page bit 1 (ROM mapping)
Line blank (L)	0	Page bit 0 (ROM mapping)

Control / Status Port 2 (1AH , 26 decimal)

Read = Status Port 2
Write = Control Port 2

ROS maintains a mask for Control Port 2 at 0FF02H

Read	Bit	Write
JB2 (L = Pressed)	7	80 character mode select (L=40 , H=80)
JB1 (L = Pressed)	6	Alternate character set (L = normal, H = alternate)
JT2 (L = Timed-out)	5	Beep (Connected directly to Loud speaker)
JT1 (L = Timed-out)	4	Joystick trigger (+ve edge triggered) (trigger joystick timers)
SIO-2 received data	3	SIO-2 Transmitted data
SIO-2 handshake	2	Cassette motor 2 control (H)
Cassette vol. (L=loud)	1	Cassette motor 1 control (H)
Cassette input bit	0	Cassette write enable (L)

Control / Status Port 3 (1BH , 27 decimal)

Read = Keyboard Data
Write = D/A Converter (may be connected to the loudspeaker)

ROS maintains a mask for Control Port 3 at 0FF03H

Control / Status Port 5 (1DH , 29 decimal)

Read = USER I/O Input Port
Write = USER I/O Output Port

ROS maintains a mask for the USER output port at 0FF00H

HRG Control Ports

These ports are active only if the option board is installed.

HRG Port 0 (38H , 56 decimal)

Read = Option Board DIL-Switch
Write = Y byte Address and Look-up Data

Read	Bit	Write
Option DIL-switch 3	7	Y Addr. 7 / Look-up Data 7
Option DIL-switch 4	6	Y Addr. 6 / Look-up Data 6
Option DIL-switch 5	5	Y Addr. 5 / Look-up Data 5
Option DIL-switch 6	4	Y Addr. 4 / Look-up Data 4
Option DIL-switch 7	3	Y Addr. 3 / Look-up Data 3
Option DIL-switch 8	2	Y Addr. 2 / Look-up Data 2
Option DIL-switch 9	1	Y Addr. 1 / Look-up Data 1
Option DIL-switch 10	0	Y Addr. 0 / Look-up Data 0

HRG Port 1 (39H , 57 decimal)

Write = X byte address
Read = Not used

HRG Port 2 (3AH , 58 decimal)

Write = Control Port
Read = Status Port

Read	Bit	Write
Not used	7	Look-up write (H)
Open (L)	6	Open (L)
Not used	5	High resolution (L)
Not used	4	Extra high resolution (L)
Not used	3	Page number / Look-up address 3
Not used	2	Video inhibit (L)/Look-up address 2
Frame blank	1	HRG inhibit (L)/ Look-up address 1
Line access	0	Look-up address 0

HRG Port 3 (3BH , 59 decimal)

Write = Write data to screen
Read = Read data from screen

INTERRUPT PRIORITIES

The interrupt priorities of the various devices capable of generating interrupts in a LINK 480Z are given in the table below in which high priorities are represented by low numbers. Thus the highest priority interrupt is the Z80 SIO channel A special conditions, and the lowest is the real-time clock 1-second interrupts.

Note that the interrupts generated by the option board CTC will only be active if both the option board and any relevant options (i.e. IEEE or Hardware floating-point) are installed.

Priority	Vector	Device	Channel/reason for interrupt
		Z80 SIO	Channel A - Network interface
0	0FC1EH		Special Conditions
1	0FC1CH		Receive Character Available
2	0FC18H		Transmit Buffer Empty
3	0FC1AH		External Status Change
			Channel B - SIO-4 interface
4	0FC16H		Special Conditions
5	0FC14H		Receive Character Available
6	0FC10H		Transmit Buffer Empty
7	0FC12H		External Status Change
8	0FC00H	Z80 CTC (Main Board)	Channel 0 SIO-4 / Cassette Input
9	0FC02H		Channel 1 SIO-2 / Cassette I/O
10	0FC04H		Channel 2 Keyboard Interrupt
11	0FC06H		Channel 3 20ms Interrupts for Repeat Key
12	0FC08H	Z80 CTC (Option Board)	Channel 0 IEEE Interrupts
13	0FC0AH		Channel 1 Floating Point Interrupts
14	0FC0CH		Channel 2 8ms Interrupts for Real-Time Clock
15	0FC0EH		Channel 3 Second interrupts for Real-Time Clock

CTC USAGE - MAIN BOARD

General

The CTC chip on the main board of the 480Z is used extensively by the Resident Operating System (ROS) for a variety of purposes. These uses are detailed below. The user is advised NOT to write any application software which uses this CTC as results are likely to be, at best, unpredictable.

Channel 0

Clock: 2MHz
Zero count: SIO-4 clock

Usage: This channel has two district uses which are mutually incompatible:

1) The channel is used to generate the clock for the SIO-4. By using a combination of timer and counter modes it is possible to generate clocks suitable for all preferred baud rates in the range 110-9600 baud. Note that since the receive and transmit clocks are permanently connected together it is not possible to transmit and receive at two different baud rates.

2) Channel 0 of the CTC is also used by the cassette system in order to determine the frequency of the input signal. For this purpose the CTC is used in timer mode with a divide-by-16 prescaler and hence counts in 4ms increments.

Channel 1

Clock: Cassette input signal
Zero count: Drives cassette output via a frequency divider (divide by 2)

Usage: This channel has several functions some of which are mutually incompatible:

1) On cassette input the channel is used to detect edges on the cassette input signal. Interrupts are generated by these edges and the time delay between successive edges is measured by means of channel 0 of the CTC.

2) On cassette output the channel is used to generate the output frequency. Note that since the output signal passes through a frequency divider before being fed to the output stage the CTC must be programmed to generate twice the desired frequencies. The frequencies generated are nominally 4800Hz and 2400Hz although for reasons of compatibility with the 380Z the actual frequencies used are approximately 4460Hz and 2230Hz. This allows easy interchange of programs and data between 380Zs and 480Zs.

3) A third use for this channel is the timing of output on the SIO-2. It is set up so that it generates an interrupt when it is time to transmit the next bit of data.

Channel 2

Clock: Keyboard strobe
Zero count: Not used

Usage: This channel has only one function - to generate interrupts on keyboard transitions. As a result of the method used for keyboardencoding, interrupts are actually generated both when a key is pressed and when it is released. This is necessary as keys such as shift, control and repeat are treated by the keyboard as ordinary keys, hence the ROS firmware has to keep track of the current state of these keys.

Channel 3

Clock: 50Hz (nominal)

Usage: The purpose of this channel is to generate interrupts at 20ms intervals. These interrupts are enabled only while the repeat key is depressed and are counted by ROS to generate the repeat frequency.

Limitations Imposed by the CTC Usage

As a result of the multiple functions performed by channels 0 and 1 of the CTC, certain combinations of the features of the 480Z may not be used concurrently. Thus cassette I/O precludes the use of the SIOs. It is also not possible to use cassette input and cassette output concurrently. Also as a result of the sharing of the CTC by several functions, problems can arise when cassette I/O is alternated with usage of the SIOs or when cassette input and cassette output are used alternately. In order to overcome these problems a new EMT (CASCTL) has been included in ROS. It is necessary to call this EMT before cassette I/O can take place and, if cassette I/O has previously been used, before using the SIOs. It should also be called when changing between cassette input and cassette output. In addition to setting up the CTC correctly for the desired function this EMT also initializes the motor control bits for the cassette recorders if a dual cassette controller happens to be in operation.

CTC USAGE - OPTION BOARD

General

The CTC chip on the option board is used partly as a real-time clock and partly as an interrupt controller. These functions are, of course, only

active if the option board (and the relevant options within the option board) are installed.

Channel 0

Clock: IEEE interrupt
Zero count: Not used

Usage: Channel 0 is used as an interrupt controller for the IEEE chip (TMS9914). This enables interrupts to be generated on, e.g., 'byte received' and 'byte transmission completed'. This function is only available if the IEEE option is installed.

Channel 1

Clock: Maths chip interrupt
Zero count: Not used

Usage: Channel 1 of the option board CTC is used as an interrupt controller for the AMD9511 (or AMD9512) hardware floating point maths chip. It allows interrupts to be generated on the completion of a task, thus allowing the use of the maths chip asynchronously with the main program. This feature is only active if the floating-point maths chip option is installed.

Channel 2

Clock: Not used
Zero count: 8ms pulses

Usage: This channel together with channel 3 is used to provide a real-time clock function for the 480Z. This channel divides the 4 MHz clock down to 1 pulse every 8ms. It may also be used to generate interrupts at 8ms intervals to allow for, say, update of the time displayed on the screen in order to minimise display flicker during scrolling.

Channel 3

Clock: 8ms pulses from channel 2 output

Usage: This channel, together with channel 2, provides the real-time clock function of the 480Z. Channel 3 counts the 8ms pulses from channel 2 and generates an interrupt every second to allow the processor to update the ' time counter' which is stored in RAM. It should be noted that the time thus produced will be correct only if the 'clock' has been initialised and set since the machine was powered up.

KEYBOARD INTERFACE

The 480Z keyboard is interfaced to the processor by means of an 8-bit, latched input port at address 1BH. Data is strobed into this by the keyboard strobe line whenever a key on the keyboard is either pressed or released. When this occurs a ready line (Bit 2 of input port 19H) is cleared to indicate to the processor that there is some data ready. This line is also connected to the keyboard to prevent further characters from being output. This line is set back to a logic 1 as soon as the processor reads the keyboard data port (1BH). The keyboard strobe line is also connected to the trigger input of channel 2 of the Z80 CTC on the 480Z main board, and this enables interrupts to be generated on every key transition.

The most significant bit of the keyboard data byte indicates whether the particular transition involved was a key depression or a key release. If it was a depression then the bit is a 0, otherwise it is a 1. The remaining 7 bits form a number unique to the actual key depressed or released (although the two shift keys are identical). In fact there are only 64 different keys and so the most significant of these 7 bits is always a 0.

It is important to note that, so far as the hardware is concerned, the shift, control, caps lock, repeat, function keys and cursor control (arrow) keys are all the same as any other key; any number generated by pressing, for example, 'A' is the same regardless of the state of the shift and control keys. It is the responsibility of the software to maintain a record of the current state of such keys and to behave accordingly. Such software is built into ROS so that, provided the keyboard is only accessed via EMTs, this mechanism is transparent to the user.

One advantage of the interrupt driven nature of the keyboard is that even although a program may not 'look at' the keyboard for some length of time the keyboard will always be serviced when a key is pressed. The software in ROS then puts the character thus obtained into a 'type-ahead buffer' so that the next time the program requests a keyboard character it is there ready.

THE VIDEO DISPLAY

The 480Z video display is a direct display of a block of static RAM onto an unmodified domestic television set and/or a video monitor of the 'wire-frame' variety. The television set, if used, is connected, via its aerial socket, to the output of a UHF modulator internal to the 480Z. The output is on channel 36 approximately. In addition to these video outputs there is also an output on the option board (if this is fitted) which allows the use of a colour monitor of the TTL RGB type.

The 480Z display is one of 24 lines of characters, each containing either 40 characters or 80 characters. Note that even with the 80 character option fitted it is still possible to produce a 40 character per line

display and indeed the machine will 'power-up' in 40 character mode. This is to allow for the use of a television set which would provide a very unclear display in 80 character mode.

The display memory is mapped as a block of 24 Z80 input/output ports and so is accessed via IN and OUT instructions. Although this memory is being used continuously to refresh the display on the screen, this fact is totally transparent to the processor which can read from or write to the memory at any time with no timing restrictions (although an extra 'wait-state' is inserted into all screen memory accesses). The port at I/O address 0 corresponds to the top line of the display while that at 17H (23 decimal) corresponds to the bottom line. In order to determine the character within the row which is to be accessed, when an IN or OUT instruction is executed the Z80 places the I/O address on the low-order 8 bits of the address bus. These top 8 bits are used to determine the character position within the line which is to be accessed. Character position 0 is at the extreme left of the screen while the right-hand edge of the screen corresponds to position 27H (39 decimal) or 4FH (79 decimal) in 40 and 80 character modes respectively.

The most suitable instructions with which to access the screen are:

```
IN r, (C)

OUT (C), r
```

These instructions place the contents of register C on the low-order byte of the address bus (thus selecting row C of the display) and the contents of register B on the high-order byte (thus selecting character position B within that line). Thus, suitable code to write the letter 'A' to the lower left corner of the screen is:

```
LD      C, 23           ; Bottom line
LD      B, 0            ; Left-hand edge
LD      A, 'A'          ; Character
OUT     (C), A          ; Output it
```

Note in particular that there is no need to open or close the screen before or after access. However the line blanking and frame blanking signals are accessible as bits in a port so that, if desired, these can be used for timing purposes by any software which requires it. Thus for example if it was desired to do something every 1/50 second then the 1/50 second could be timed by using the frame blanking signal.

APPENDIX A

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

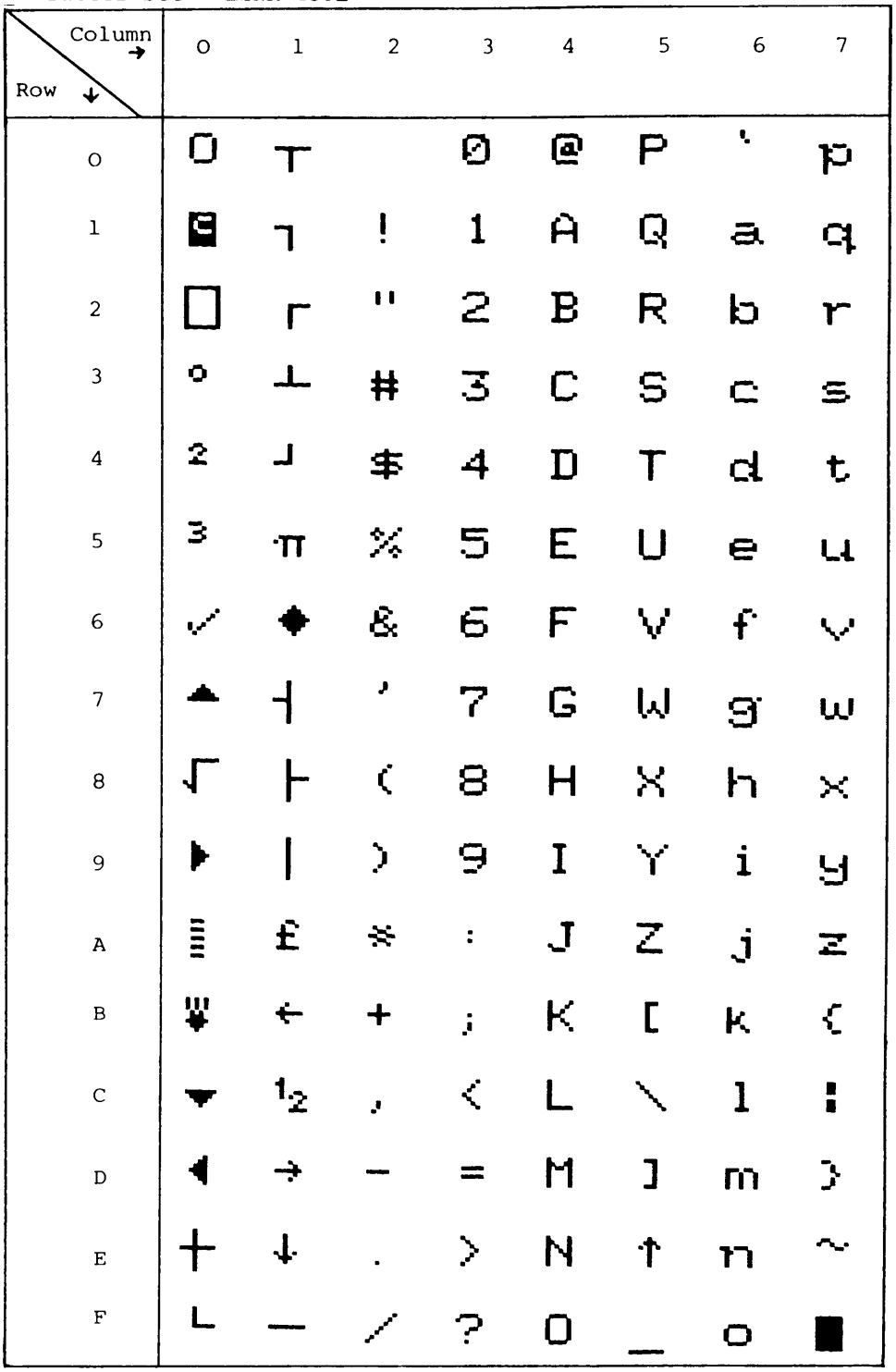
Reading the column number and the row number for an entry in the table will yield a hexadecimal code for the character concerned. For example, the character G has code 47 hex (71 decimal).

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 480Z screen will cause it to discard output, but when sent to some printer may cause it to revert to single width characters.

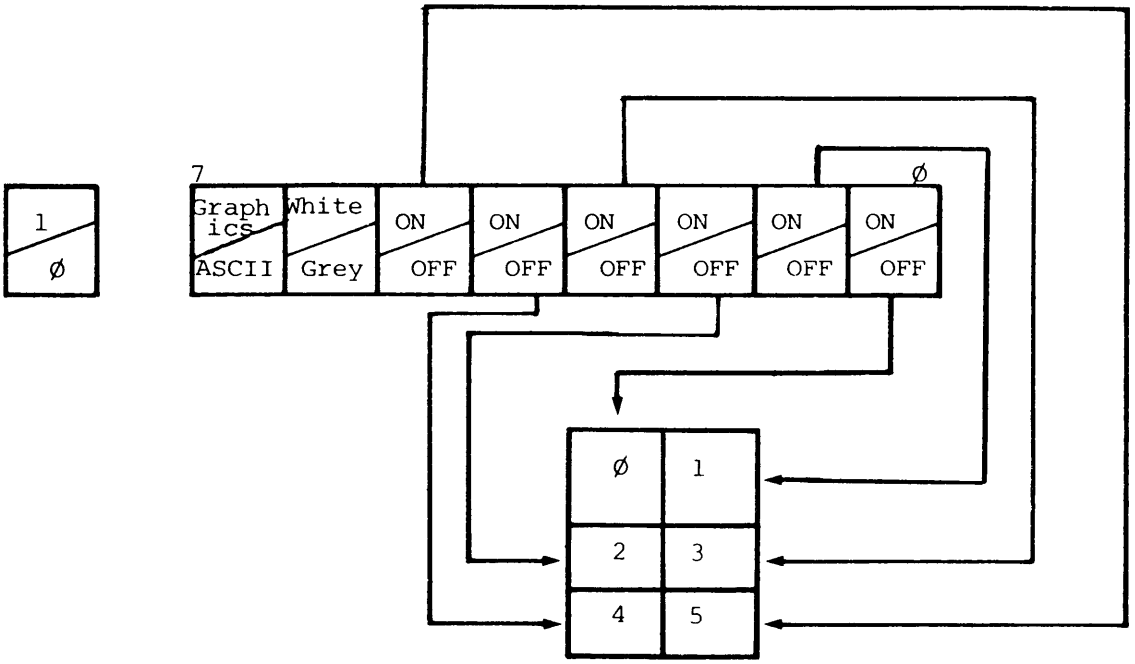
Character Set - LINK 480Z



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

Low Resolution Graphics Character Set

A graphics character is made up of six separate screen units or 'pixels'. The character is formed by a pattern of such pixels and the pattern is determined by switching individual pixels 'on' or 'off'. For each character there is a corresponding byte (8 bits) of screen memory; this byte can be set to a value which decides what state each pixel will take to form that character. The following diagram illustrates the relationship between a byte of data and the six pixels the byte represents (the bits are numbered from 0 to 7):



The six pixels are represented by the lower box numbered 0 to 5. The arrows indicate which bits affect a particular pixel. If a bit is set to 1 then that pixel is switched on; if set to 0 it is switched off.

Bit 6 determines the colour of the graphics character (1 = white, 0 = grey)

Bit 7 is always set to 1 for graphics mode (0 = ASCII mode)

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.

APPENDIX B

CIRCUIT DIAGRAMS

This appendix contains a set of circuit diagrams for the 480Z printed circuit boards. the areas on these boards are illustrated as follows:

480Z Main Board

Diagram

B.1 Processor Support & Port Decoding

B.2 Memory System - Decoding/Timing

B.3 RAM & ROM Arrays, Character Generator & Power

B.4 VDU Circuitry

B.5 TTL I/O Ports - Keyboard, User I/O, Accessories, SIO-2

B.6 SIO, CTC & Analogue: SIO-4, Cassette & Audio

480Z Option Board

Diagram

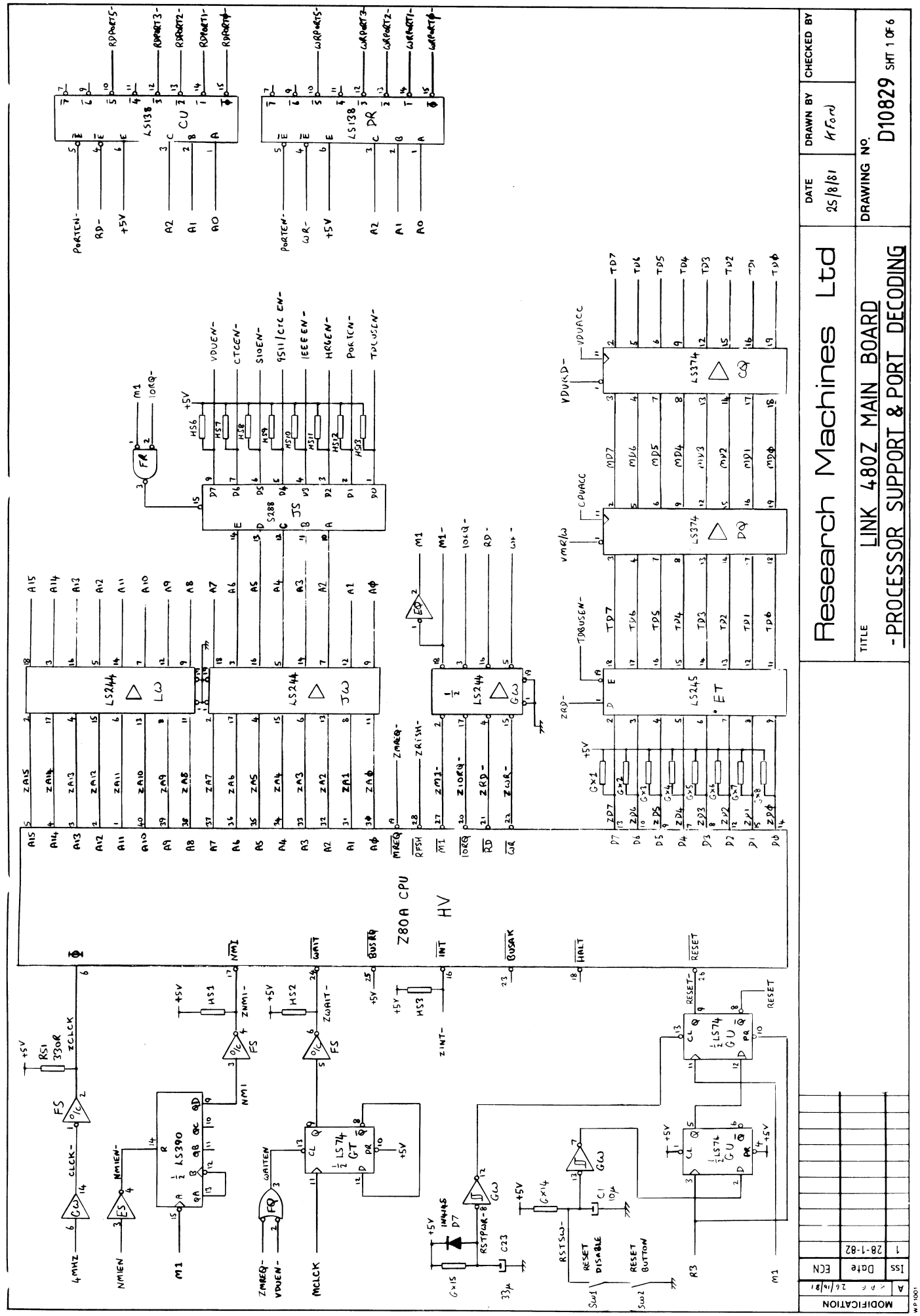
B.7 Add On RAM

B.8 HRG 1: Bus Interface & Memory

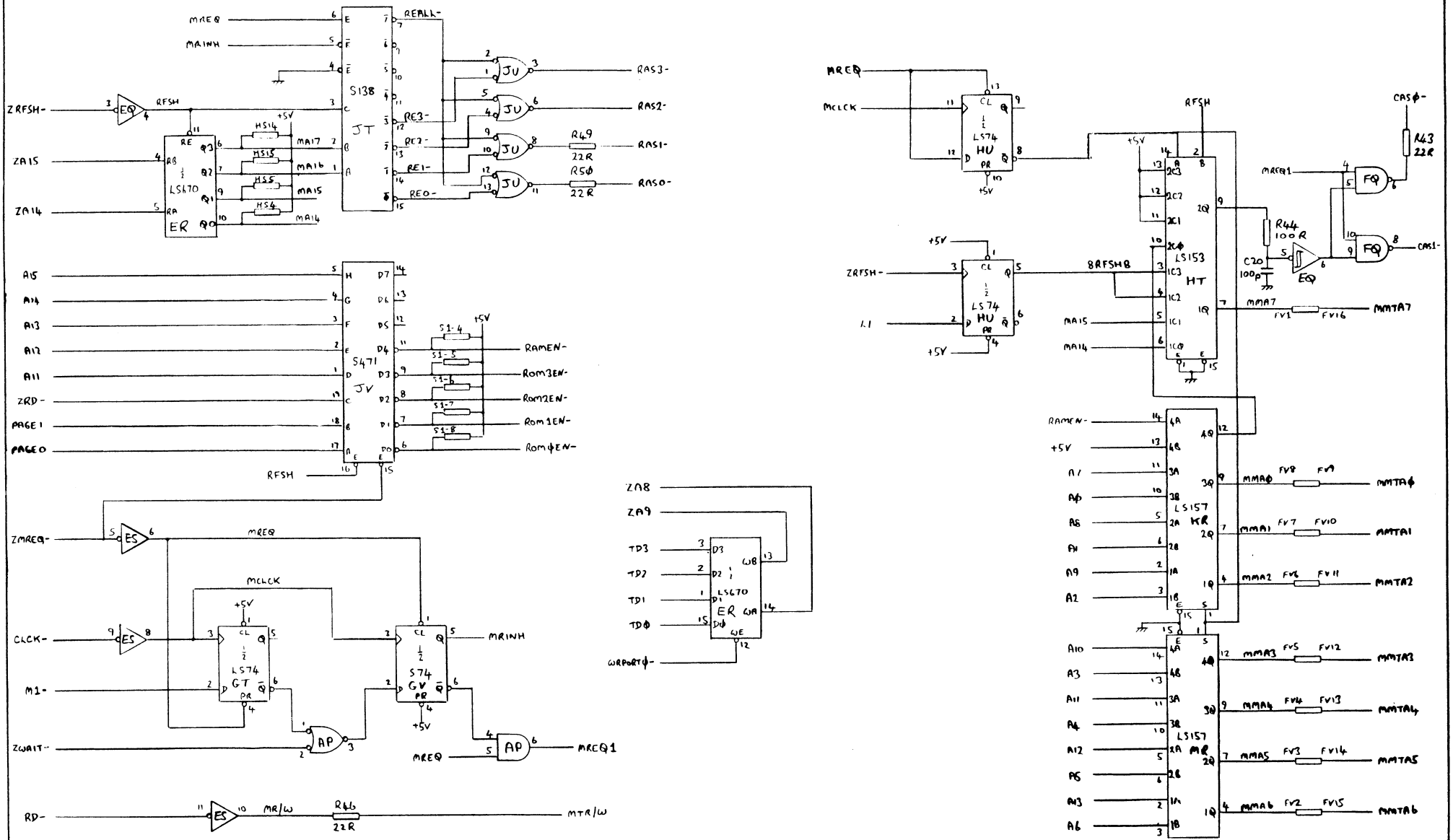
B.9 HRG 2: Timing & Video Generation

B.10 HRG 3: Output Stages

B.11 IEEE Interface, Maths Chip & CTC



B.1 Processor Support & Port Decoding



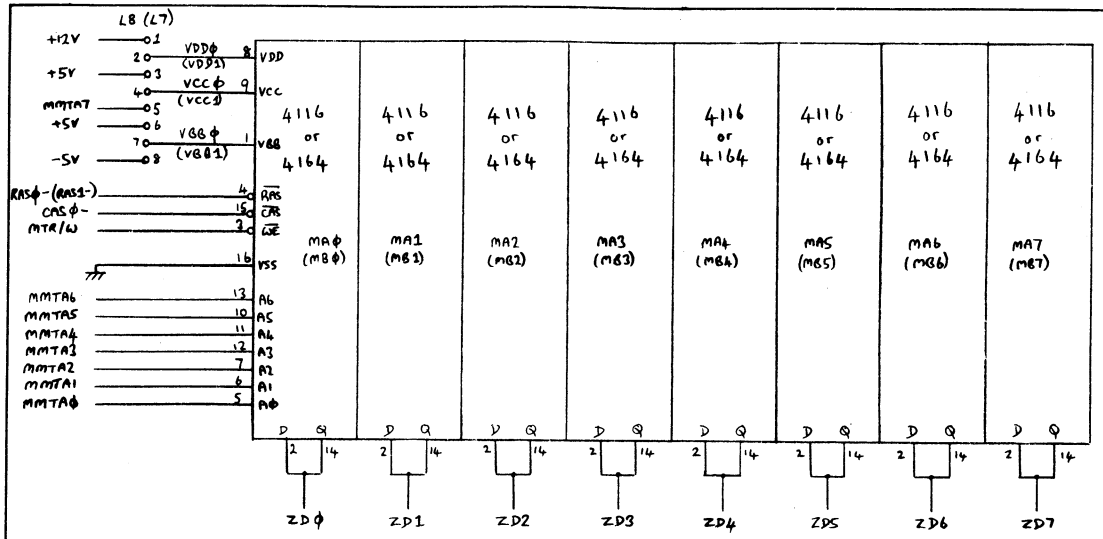
MODIFICATION	ECN	Date	Iss
A		28-1-82	1

Research Machines Ltd	DATE	DRAWN BY	CHECKED BY
	25/8/81	K Fenn	

TITLE: LINK 480Z MAIN BOARD
-MEMORY SYSTEM - DECODING/TIMING.

DRAWING NO. D10829 SHT 2 OF 6

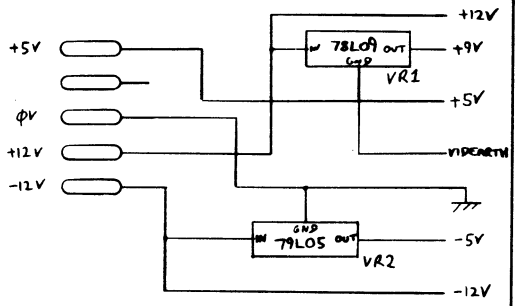
LM2/0



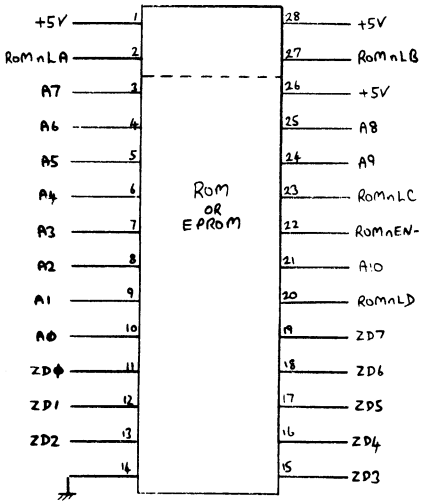
TWO BANKS OF DYNAMIC RAM MAφ-7 AND MBφ-7. EACH BANK CAN CONTAIN EITHER 8 × 4116 OR 8 × 4164. THE TYPES OF CHIP INSTALLED IN A PARTICULAR RAM BANK IS SELECTED WITH THE ROW OF LINK PINS ASSOCIATED WITH THAT BANK - L8 FOR BANK φ AND L7 FOR BANK 1.

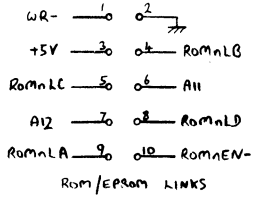
FOR 4116's LINK PINS:-
1-2 , 3-4 , 7-8

FOR 4164's LINK PINS:-
2-3 , 4-5 , 6-7



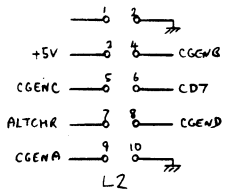
ALL POWER SUPPLY LINES (+12V, +9V, +5V, -5V, -12V) ARE DECOUPLED WITH A MIXTURE OF CERAMIC CAPS (Cc) AND TANTALUM BEADS (Ct FOR 5V, Cu FOR OTHERS) CONNECTED FROM THE SUPPLY TO GROUND (EXCEPT FOR +9V WHICH IS DECOUPLED TO VIDEARTH)

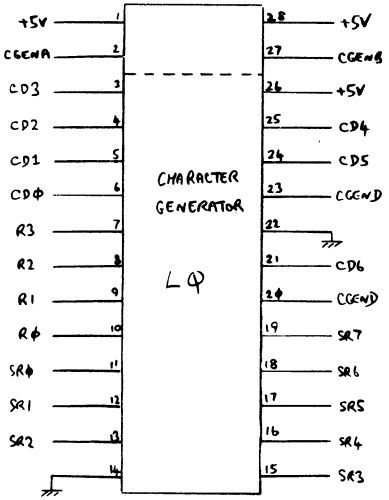




ROM/EPROM LINKS

Rom no.	CHIP	LINKS	
φ	L5	L3	φ
1	L7	L4	1
2	L9	L5	2
3	LV	L6	3





THERE ARE 4 SOCKETS FOR ROM: OR EPROM: CONTAINING SYSTEM OR APPLICATION SOFTWARE. EACH OF THESE SOCKETS HAS, ASSOCIATED WITH IT, A SET OF LINK PINS BY SUITABLE CONNECTION OF THESE A LARGE VARIETY OF ROM: OR EPROM: OF THE 5V ONLY VARIETY MAY BE USED, RANGING IN SIZE FROM 2KByte to 8K Bytes

N.B. WHEN INSTALLING 24 PIN DEVICES PINS 3-26 OF THE SOCKET SHOULD BE USED.

AS WITH THE ROM SOCKETS THE CHARACTER GENERATOR SOCKET CAN SUPPORT MOST TYPES OF ROM/EPROM

CHIP	CONNECT LINKS:-
2516/2716	3-5 , 8-10
2532	3-5 , 6-8
2732	5-6 , 8-10
2564	2-4 , 5-7 , 6-8 , 9-10
2764	3-4 , 5-6 , 7-9 , 8-10
68764	5-7 , 6-8

MODIFICATION			
A	ECN		
	Date	29-1-82	
	Iss	1	

WP 100

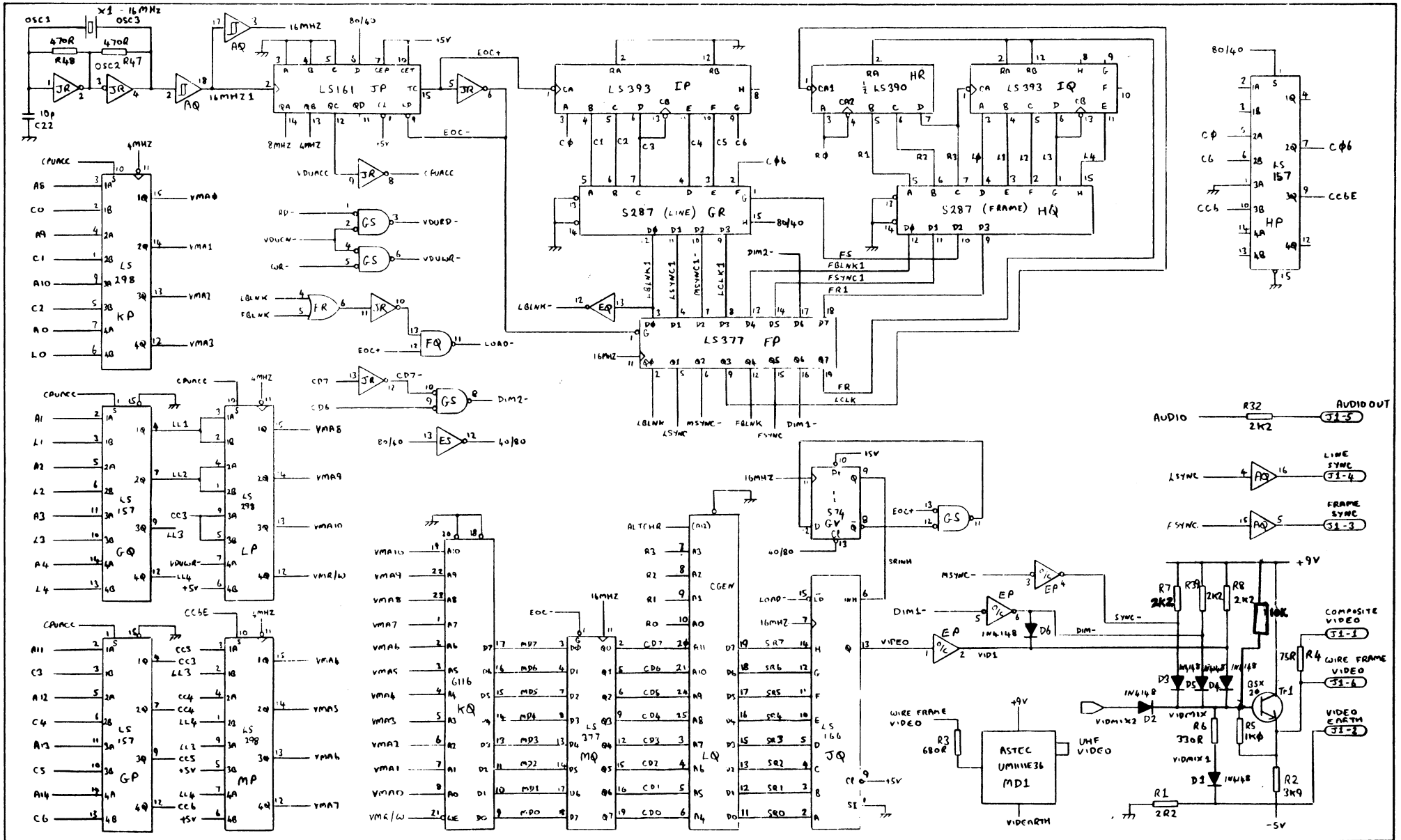
Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
27/8/81	K Ford	

TITLE: LINK 480Z MAIN BOARD
RAM & ROM ARRAYS, CHARACTER GENERATOR & POWER

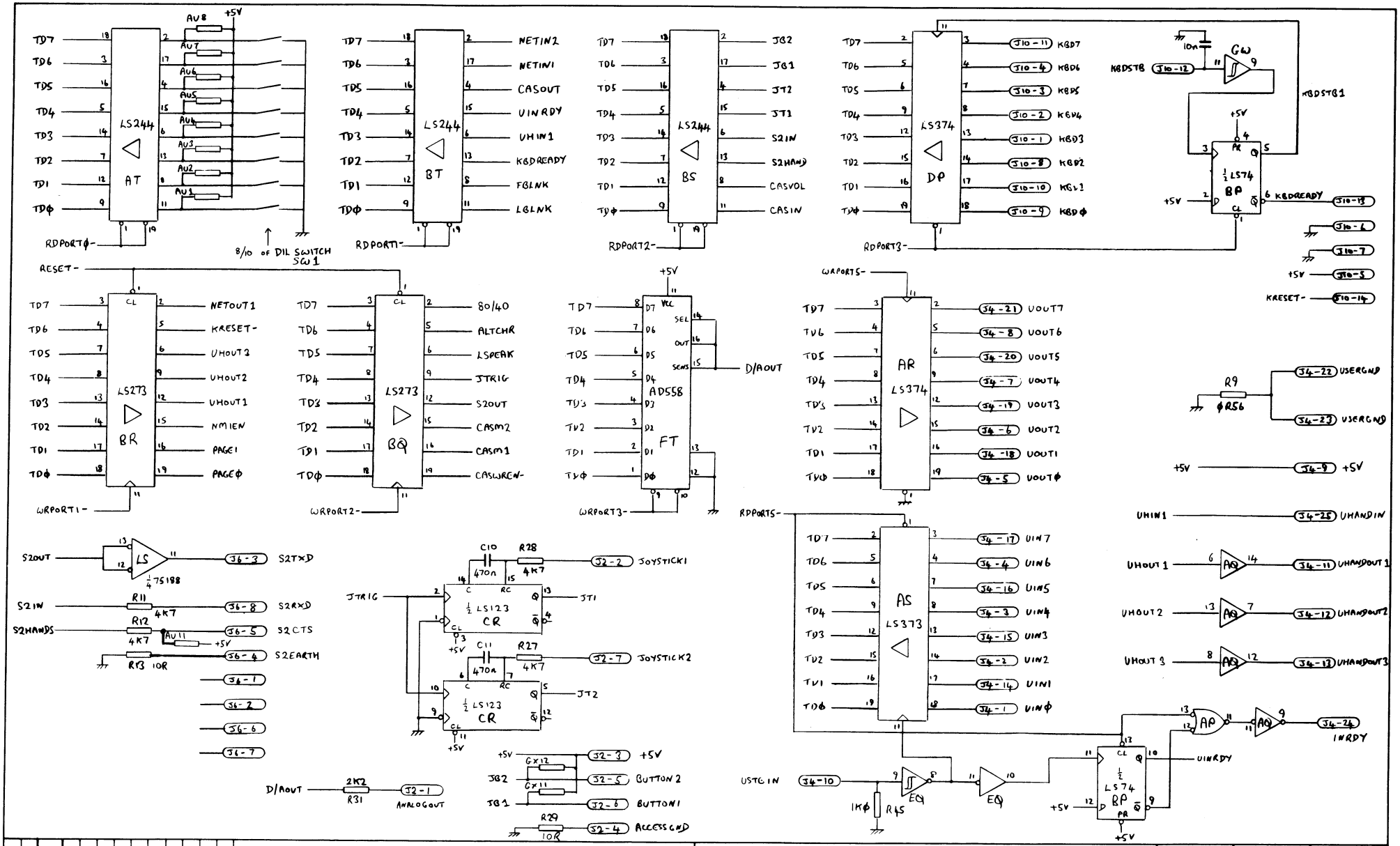
DRAWING NO. D10829 SHT 3 OF 6

LM 4/4



MODIFICATION			
ECN			
Date	29-1-82		
Iss	1		

Research Machines Ltd	DATE	DRAWN BY	CHECKED BY
	21/8/81	K. Ford	
TITLE: LINK 480Z MAIN BOARD - VDU CIRCUITRY	DRAWING NO. D10829 SHT 4 OF 6		



MODIFICATION			ECN
A	KAF	26/10/81	
Iss	Date	29-1-82	
1			

Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
26/8/81	K. Ford	

TITLE

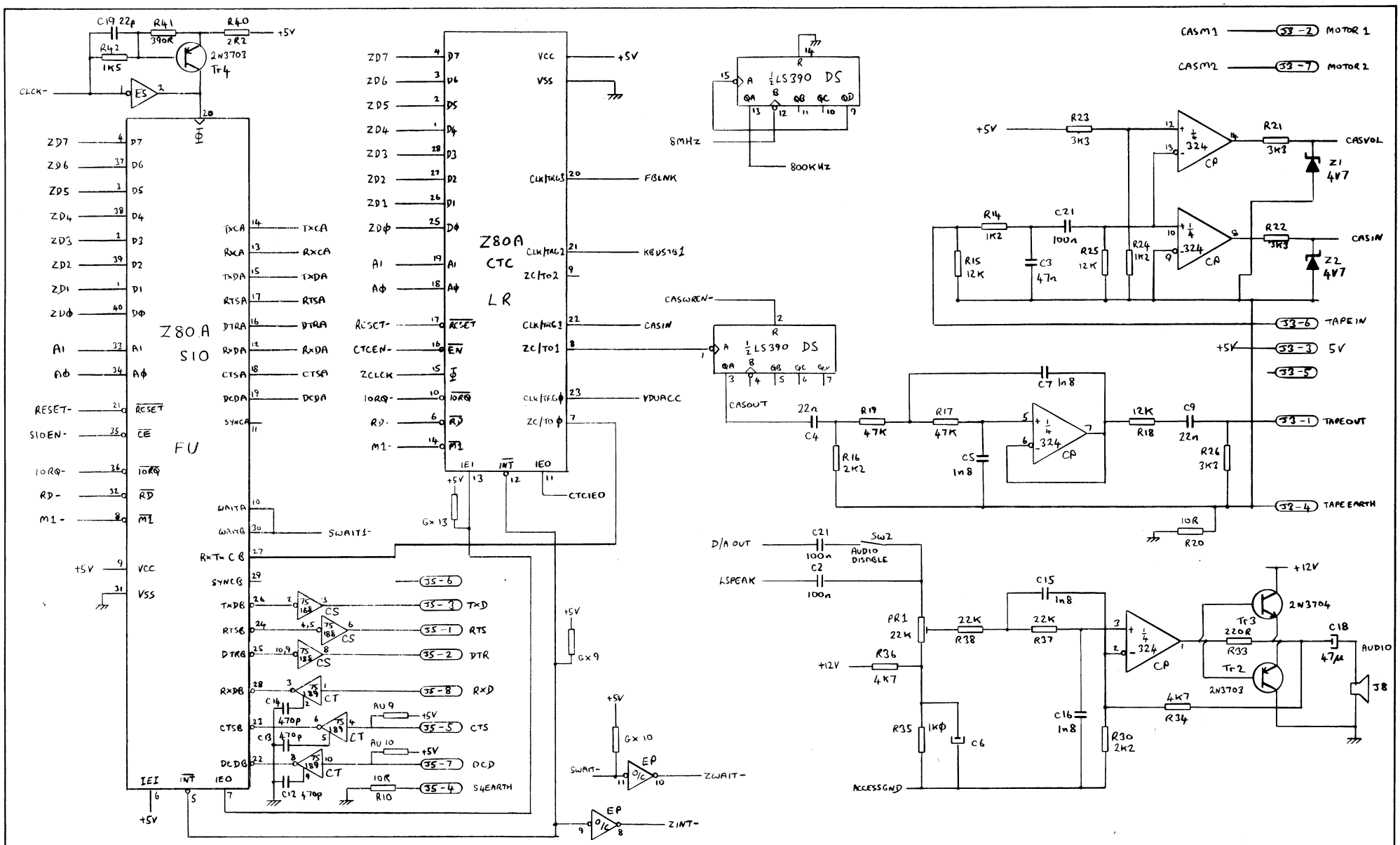
LINK 480Z MAIN BOARD

TTL I/O PORTS-KEYBOARD, USER I/O, ACCESSORIES, SIO-2

DRAWING NO. D10829 SHT 5 OF 6

WP 1001

LMS/0



MODIFICATION			
A	KAF 26/10/81	ECN	
		Date	29-1-82
		Iss	1

WP 1001

Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
27/8/81	K. Ford	

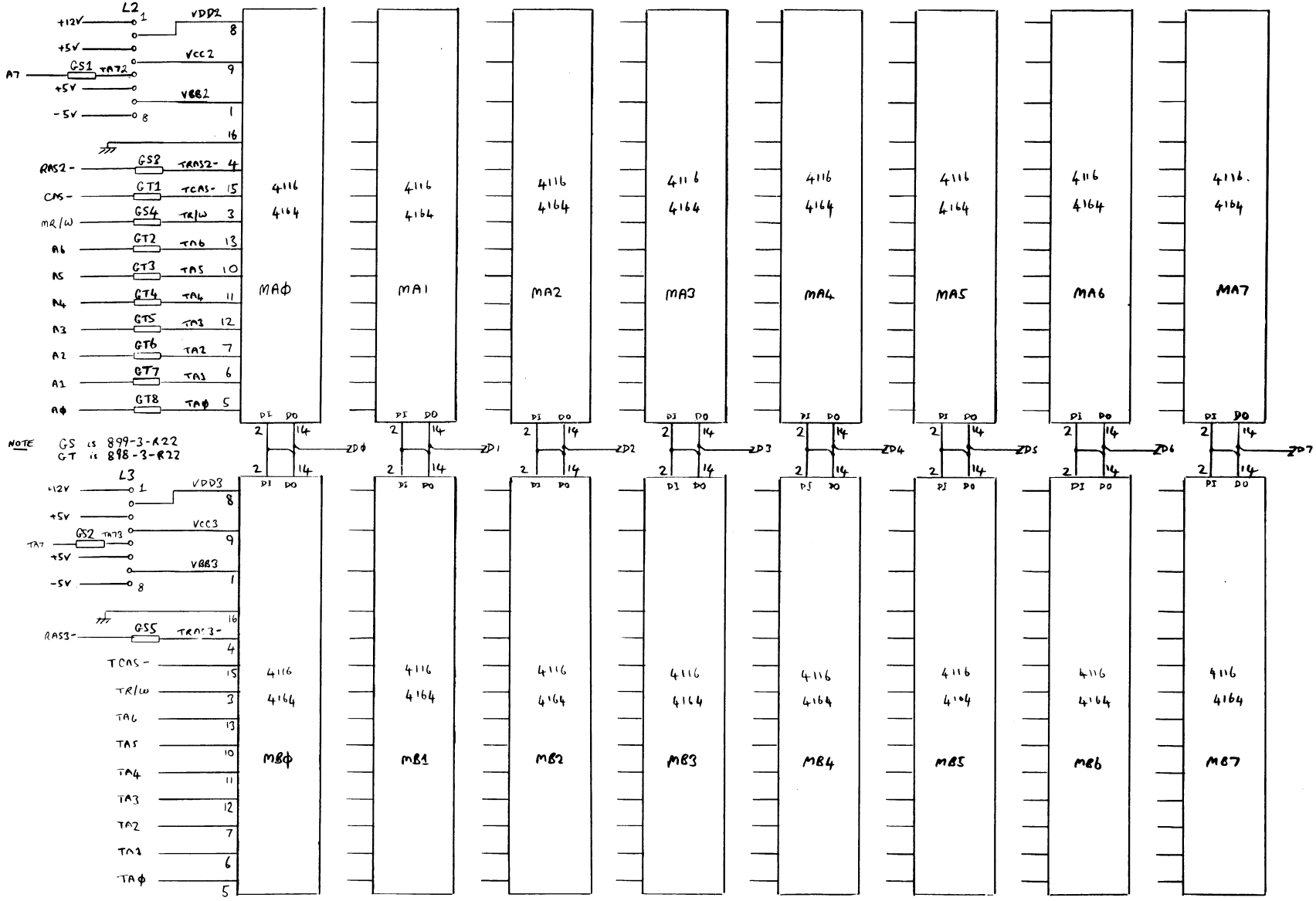
TITLE

LINK 480Z MAIN BOARD

SIO, CTC & ANALOGUE-SIO/4, CASSETTE & AUDIO.

DRAWING NO. D10829 SHT 6 OF 6

LM 6/4



MODIFICATION		
ECN		
Date	29-1-82	
Iss	1	

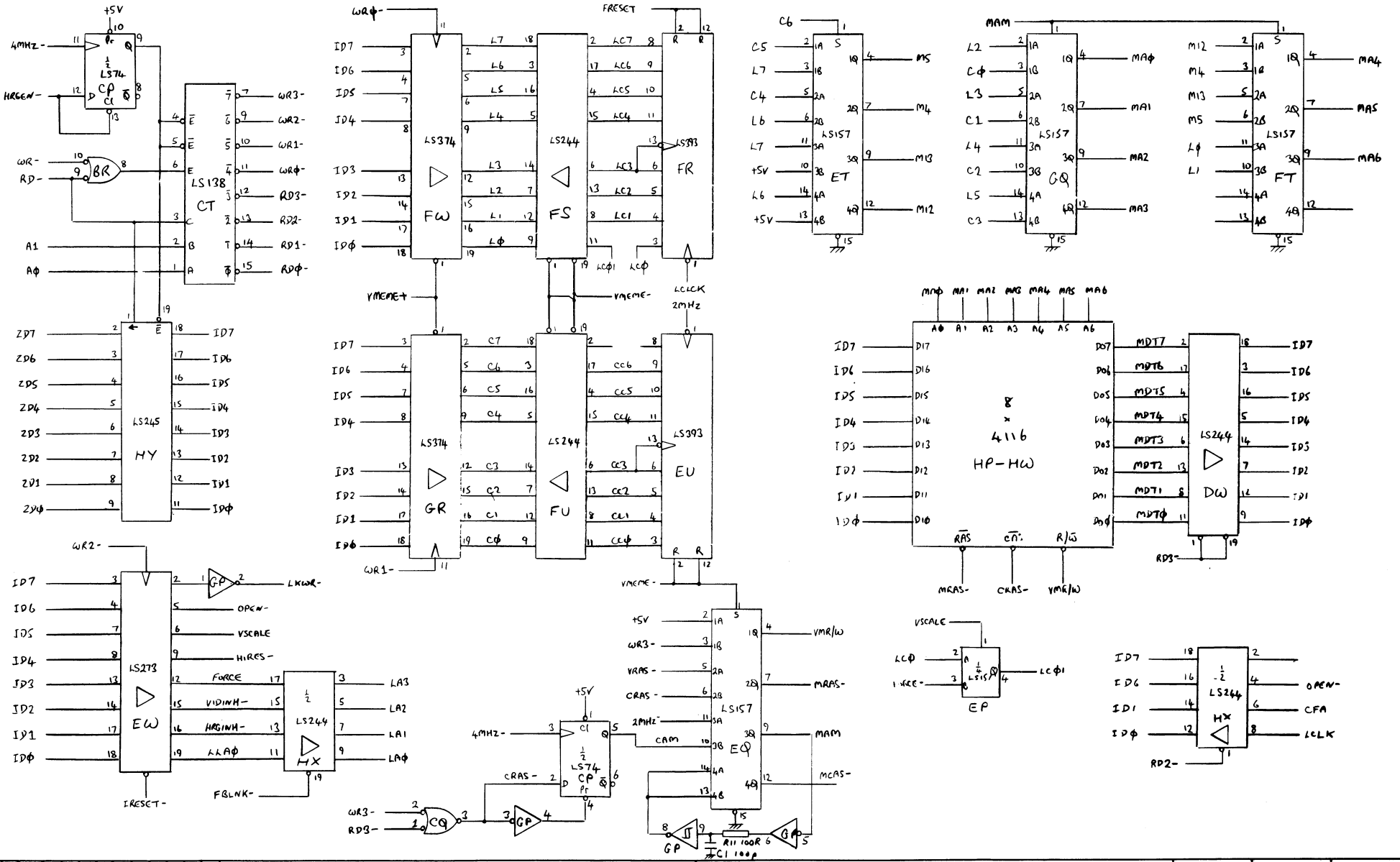
Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
8/12/81	K.Fow	

TITLE: LINK 480Z OPTION BOARD - ADD ON RAM.

DRAWING NO. D10830 SHT 1 OF 5

WP 1001



MODIFICATION			
ECN			
Date	29-1-82		
Iss	1		
A			

Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
8-10-81	K FORD.	

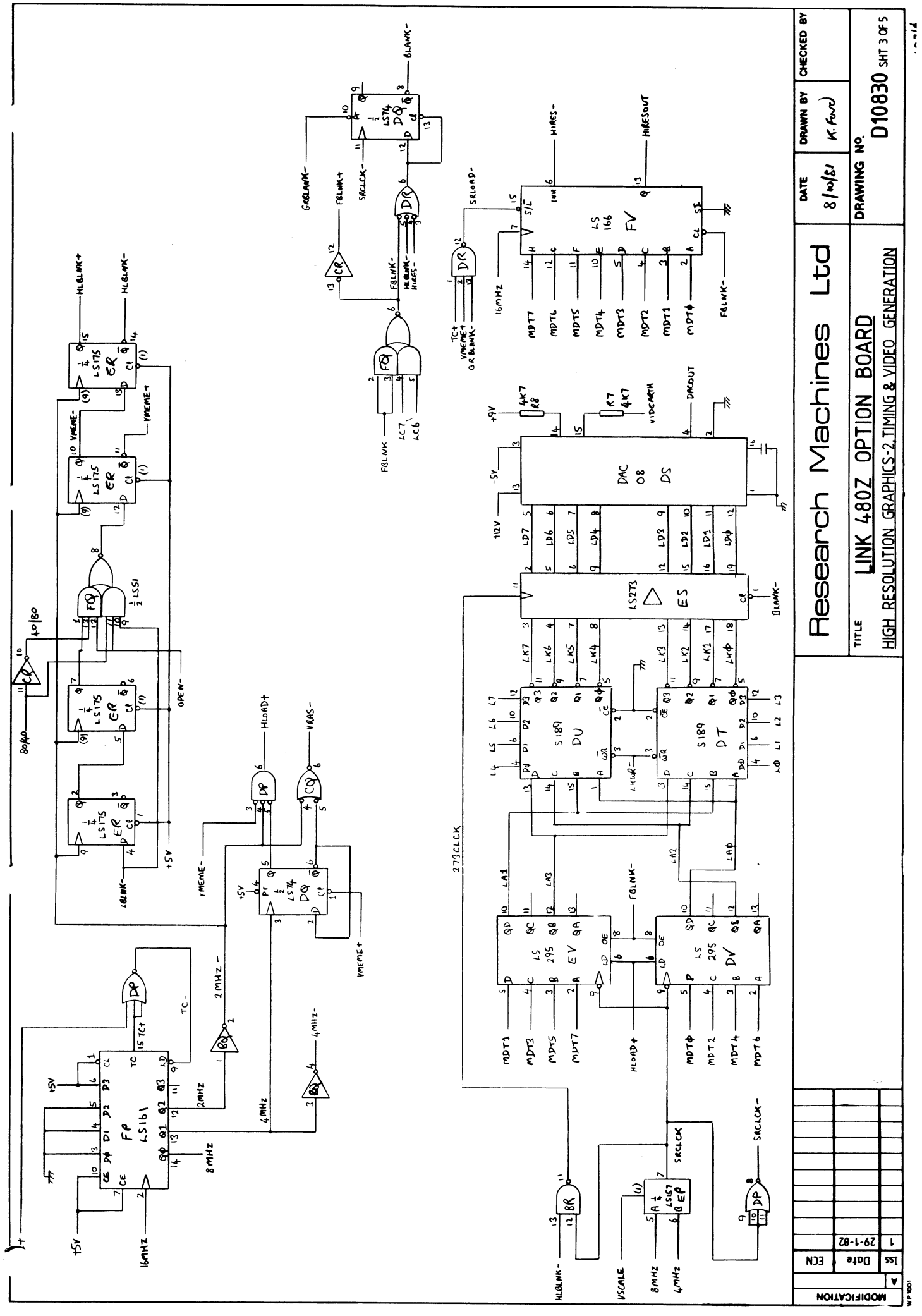
TITLE

LINK 480Z OPTION BOARD

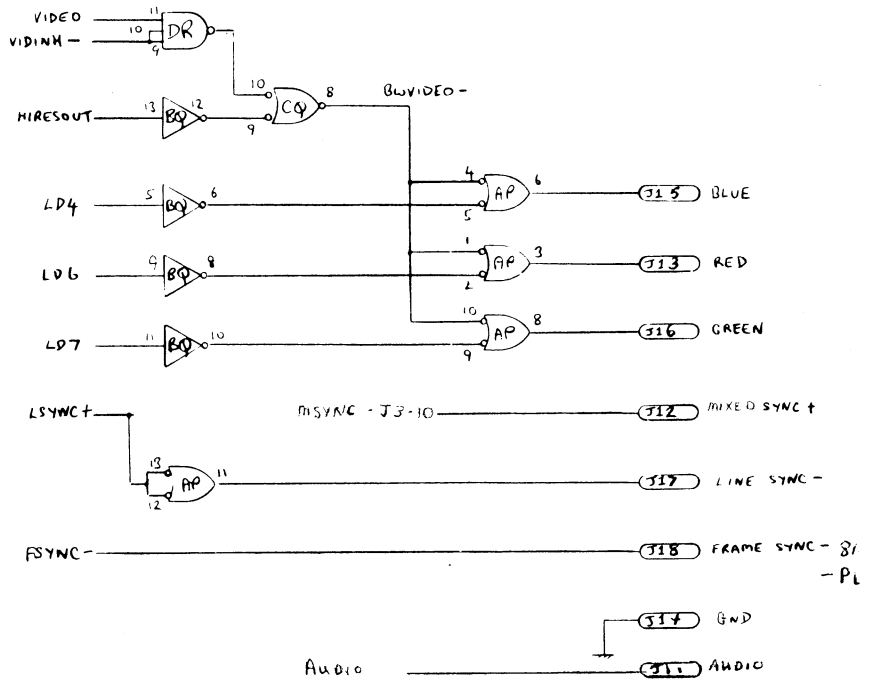
HIGH RESOLUTION GRAPHICS-1, BUS INTERFACE & MEMORY

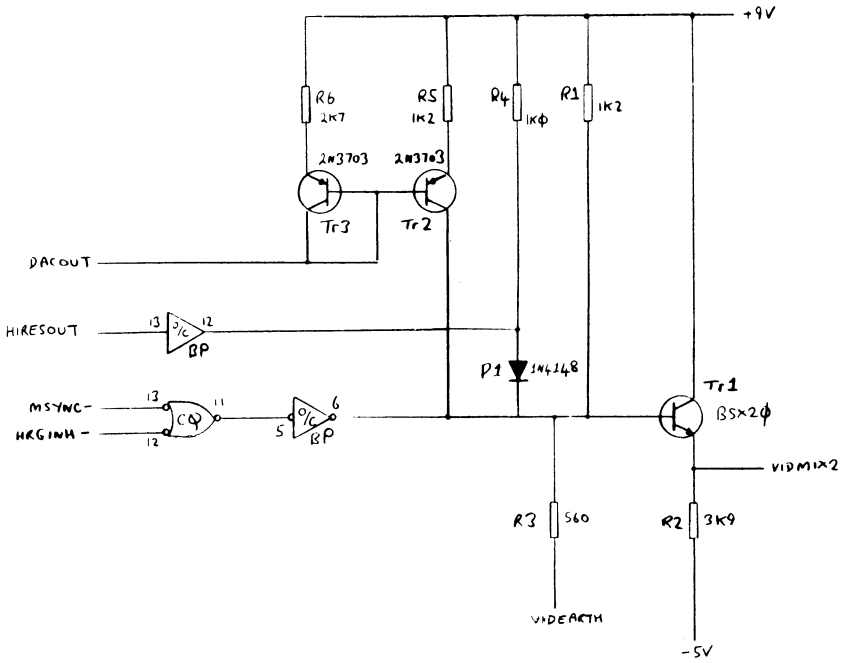
DRAWING NO. D10830 SHT 2 OF 5

WP1001



B.9 HRG 2: Timing & Video Generation





MODIFICATION	ECN	Date	Iss
A		29-1-82	1

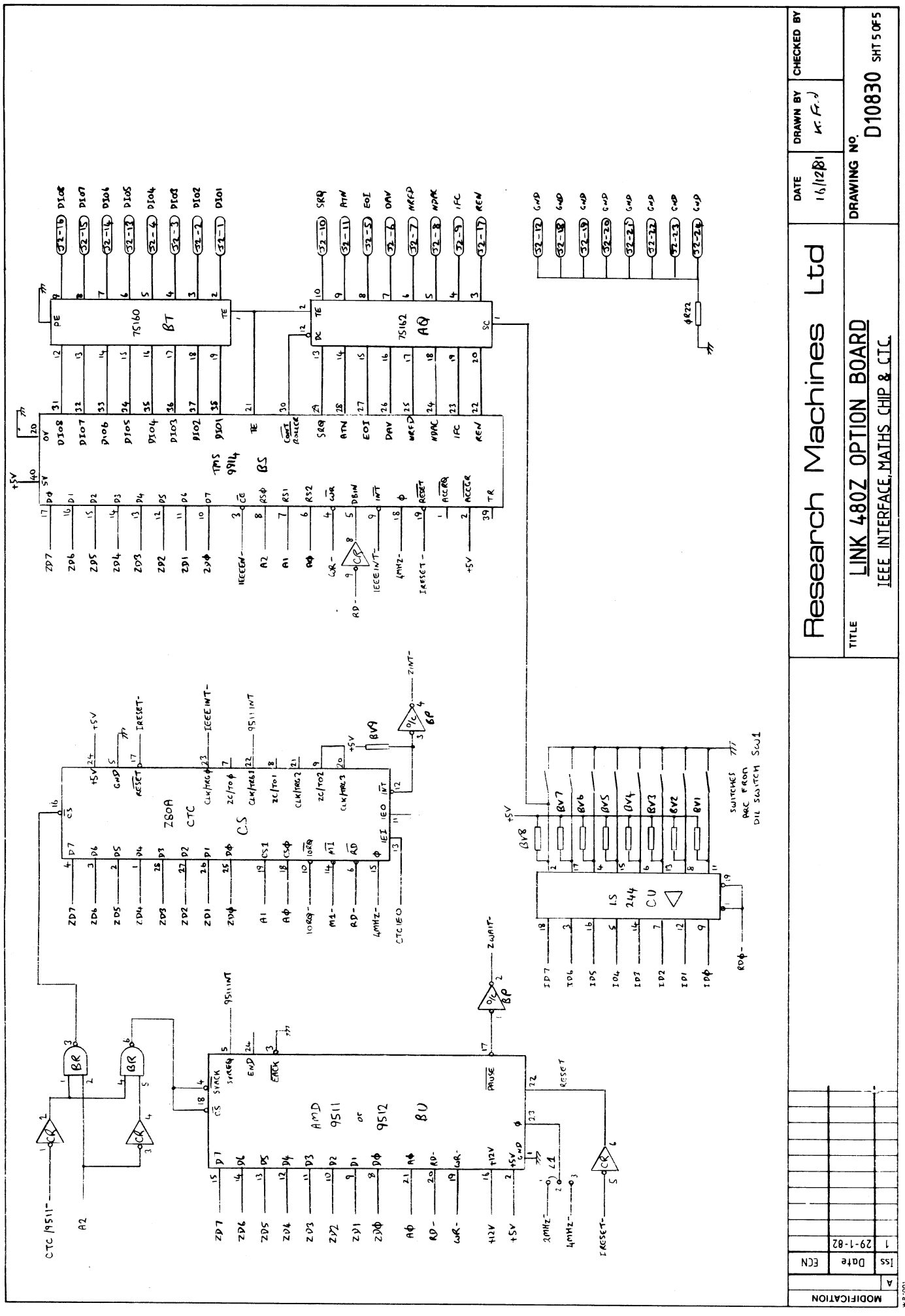
Research Machines Ltd

DATE	DRAWN BY	CHECKED BY
17/12/81	K Ford	

TITLE: LINK 480Z OPTION BOARD
HIGH RESOLUTION GRAPHICS-3, OUTPUT STAGES.

DRAWING NO. D10830 SHT 4 OF 5

WPT



B.11 IEEE Interface, Maths Chip & CTC

INDEX

Accessories Socket	3.9, 3.12
Analogue output	2.2, 3.9
Baud rate	3.5
'Beep'	2.1, 3.8
CASCTL	5.9
Cassette	2.1, 5.3, 5.5, 5.9
Cassette Interface	3.1
Cassette filing system	3.1
Colour	4.1, 4.8
Connection Information	3.11
CTC	5.7-5.10
Display	2.1
Documentation	1.1
Dual cassette controller	3.1
Extra high resolution	4.1
Frame blanking	4.7, 5.5, 5.11
Graphics	2.1
Hardware Floating Point	2.2
High Resolution Graphics	4.1
Interrupt priorities	5.7
Joystick	3.9, 5.5
Keyboard	2.1, 5.3-5.5, 5.9, 5.11
Line blanking	4.6, 5.5, 5.11
Loudspeaker	2.1, 5.5
Low resolution graphics character set	A.4
Maths chip	5.3, 5.10
Medium resolution	4.1
Memory maps	5.1
Network	2.1, 5.3
Network Address Switches	3.10
Option Board	2.2, 2.3
Parallel Interface	2.1, 3.8, 3.15
'Pixel'	4.1, A.4
Ports	5.3
Printer	3.5, A.3
Push buttons	3.9

"READY" line 3.5
Rear panel 3.11
Real-time clock 5.10
Reset button 2.1, 3.11
Reset Disable Switch 3.9
ROM 2.1

Serial interfaces 2.1, 3.5
SETLST 3.5
SIO-2 3.5, 3.13, 5.7
SIO-4 3.5, 3.13, 5.7
Speaker disable switch 3.10, 3.11
Switches 3.9

Television 5.11
Terminal mode software 3.6

User I/O port 3.7, 3.8, 5.3, 5.5

Video Socket 3.11
Video Outputs 4.2, 4.8