

**Servicing the BBC Micro**  
**Additional Circuits**



## The "Econet" Circuitry.

The heart of the "Econet" circuitry is IC89, a Motorola MC6854 Advanced Data Link Controller (ADLC).

Connection to the "Econet" network is by a 5-pin 180 degree DIN socket (SK7) on the rear of the machine. The network clock connects to pins 3 and 5 of this socket, whilst pins 1 and 4 carry the data. Pin 2 is connected to 0 volts. The data line is bi-directional, thus both transmitted and received data is carried along the same pair of wires in half-duplex mode. Unlike the RS423 serial data transmission port, the "Econet" system uses a differential mode of operation to provide high common mode noise immunity and achieve high transfer rates.

Pin 6 (TXD) of the ADLC feeds the TTL data to be transmitted to one half of a differential line driver (IC93). Provided that this line driver is enabled by the output of IC91, one quarter of which is used as an inverting buffer, data will then be transmitted. The input signal to this inverter comes directly from pin 2 (RTS) of the ADLC.

Any "incoming" data on the data lines is converted to a TTL signal by an analogue comparator (one half of IC94). The TTL signal is then fed into the received data input (RXD) of the ADLC. In a similar way, the network clock is also detected by the other half of IC94. The output of this comparator feeds both the receive clock and transmit clock inputs of the ADLC. Thus it is not possible to use different receive and transmit baud rates.

The network clock must always be present in order for the ADLC to synchronise correctly, a monostable (half of IC87) is used to achieve this. The time constant of this monostable is such that as long as the clock signal is present it remains permanently triggered. One output of the monostable provides the data carrier detect signal (DCD) into the ADLC, the other output activates the clear to send signal (CTS) via a dual input NAND schmitt gate. By using this technique, network activity is suspended if the clock signal is interrupted.

Because of the democratic nature of the Econet system it is possible for two or more devices to attempt simultaneously to transmit data on the network. This condition is known as a "collision". A collision on the network will cause the nominal signal level on the data lines to drop because of the increased loading, this condition is detected by a dual comparator

(IC95). The dual comparator arrangement sends a TTL output signal into one quarter of IC91, which is acting as an inverter. The output of this inverter activates the clear to send input of the ADLC via another quarter of IC91, thus transmission is suspended in the event of a collision. Each of the microcomputers involved in the collision will then retry. The condition is detected by both machines and an arbitration algorithm within the "Econet" system software ensures that no two computers will retry simultaneously.

The individual station address is set by the group of links S11, in conjunction with IC96. This is detailed in the chapter 4 which deals with link functions.

The "Econet" network must be terminated at each end with its characteristic impedance in order to prevent signal reflections. On early issue printed circuit boards resistors R26,42,53,54,56, capacitors C19,22 and diode D3 provide this optional termination.

The network clock is usually derived from an external "clock box", however, on some early issue main PCB's, circuitry was provided to enable the network clock to be generated on board. This circuitry consisted of IC90 (4018), IC97 (74LS74) and the normally unused half of IC93. The 6MHz clock signal derived from the output of IC37 is divided by two, using half of IC97 (74LS74). This 3MHz signal is then further subdivided in to a range of frequencies by IC90, link S3 is then used to select the required base clock frequency. Clock base frequencies usually lie in the region 75KHz to 625KHz. The required frequency is dependant on the line length of the network.

The base clock frequency is then fed directly in to the input of the normally unused half of the line driver IC93. The output of this line driver is then connected to the clock output of the DIN"Econet" connector and provides the network clock source.

The link S5 is used to enable the clock line driver and link S6 may be used to further subdivide the clock base frequency by 2 or 4.

### The Voice Synthesiser.

The voice synthesiser used in the BBC Microcomputer is a Texas Instrument two chip set, comprising of a microprocessor-bus compatible Voice Synthesis Processor (VSP) type TMS5220 (IC99) and a Voice Synthesis read-only Memory (VSM) containing speech data, type TMS6100 (IC98). Speech data for the VSP can be taken either from the VSM or from a slow (1MHz) data bus which is provided by port A of the system VIA (IC3). Interrupt (INT) and ready (RDY) output signals from the VSP are fed to lines PB6 and PB7 of the system VIA, respectively. Thus the VSP is under software control.

The speech data inside the VSM is a digitised form of the originally recorded human speech. In order to keep memory requirements to a minimum, this data is a compressed form of the original digitised speech data. The compression technique used is pitch excited Linear Predictive Coding (LPC). The data from the VSM is decoded within the VSP to reconstruct the digital speech information. An 8-bit digital-to-analogue converter inside the VSP provides the analogue speech signal, which is then filtered using an operational amplifier based filter (part of IC17). The filtered signal is then mixed with the analogue signals from the 1MHz bus and sound generator before amplification by the LM386 power amplifier (IC19), details of which are given in the following section.

### The 1MHz Bus Interface.

In addition to the high speed "Tube" interface the BBC Microcomputer also has a 1MHz extension bus. Connection to this interface is through PL11 on the underside of the machine. The 1MHz extension bus provides buffered address lines A0 to A7 (IC71), buffered bi-directional data bus lines D0 to D7 (IC72) and numerous control lines. The interface also has an analogue input line which is fed to the machines internal audio amplifier (IC's 17 and 19) and loudspeaker. The input impedance of this analogue input is 9 kilohms. If the speech and internal sound generator systems are not in use then a signal of + 3 volts RMS will produce full volume output at the loudspeaker. This input level will need to be reduced in order to avoid excessive distortion if the speech and sound generator systems are in use at the same time. There are two other lines on the 1MHz bus connector which will require further explanation, these are the lines NPGFC (Not PaGe &FC ) and NPGFD (Not PaGe &FD) which are often referred to as FRED and JIM respectively.

#### 3.19 FRED.

This signal is decoded from the machines address bus such that when the address bus holds a valid address in the range &FC00 to &FCFF this line will go low. The microcomputer address decoding circuitry around IC20 (74LS139) is used to accomplish this. FRED is intended for use by peripheral devices with small memory requirements.

There are various links within the machine that can be used to enable/disable FRED (S17) or alter its access times (S16).

Acorn Computers Ltd. have standardised on particular addresses in the range &FC00 to &FCFF (FRED) to be used for specific functions, at the present time these are as follows;

```

&FC00 - &FC0F   Test hardware use.
&FC10 - &FC13   Teletext use.
&FC14 - &FC1F   Prestel use.
&FC20 - &FC27   IEEE 488 interface (Control Universal Ltd.).
&FC28 - &FC2F   Future Acorn expansion.
&FC30 - &FC3F   Cambridge Ring interface.
&FC40 - &FC47   Winchester disc interface.
&FC48 - &FC7F   Future Acorn expansion.
&FC80 - &FC8F   Test hardware devices.
&FC90 - &FCBF   Future Acorn expansion.
&FCC0 - &FCFE   User applications.
&FCFF          JIM paging register (see JIM).

```

Normally devices attached to the 1MHz bus should decode the address lines A0 to A7 and also the NPGFC line (FRED). In most cases the signal on the NPGFC line will need to be "cleaned up"

The reason that the NPGFC and NPGFD signals need to be cleaned up is because of the difference in the speed that the 1MHz bus is running at, compared to the speed at which the machines internal 6502A processor is running. The machines internal processor runs at a clock speed of 2MHz unless a slow speed device, such as the 1MHz bus, is being accessed. Circuitry within the BBC Microcomputer is used to effectively "stretch" the 2MHz normal 6502A clock cycle into the 1MHz clock (1MHz E) cycle, required by 1MHz peripheral devices attached to the 1MHz bus. This circuitry is described in some detail above.

The NPGFC signal (FRED) produced by the microcomputer's internal circuitry suffers from two distinct problems, these are "glitches" and "double accessing";

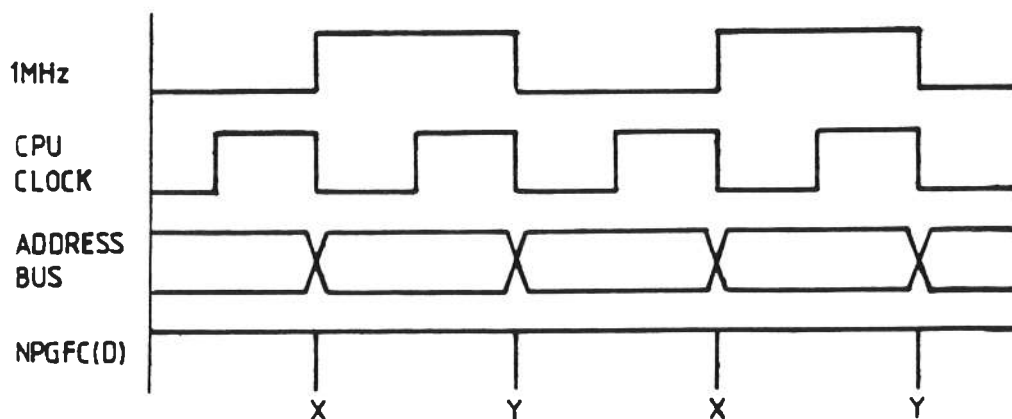


FIGURE 1 TIMING DIAGRAM (1MHz BUS)

glitches are marked X and Y in figure 1. The glitches which occur when the 1MHz E clock is low (marked Y in figure 1) are normally of no consequence but the glitches occurring when 1MHz E is high (marked X in the diagram) may cause the chip select inputs of the various IC's on the 1MHz peripheral device to be spuriously activated. One of the circuits shown in figure 2, below, can be used to eliminate this problem.

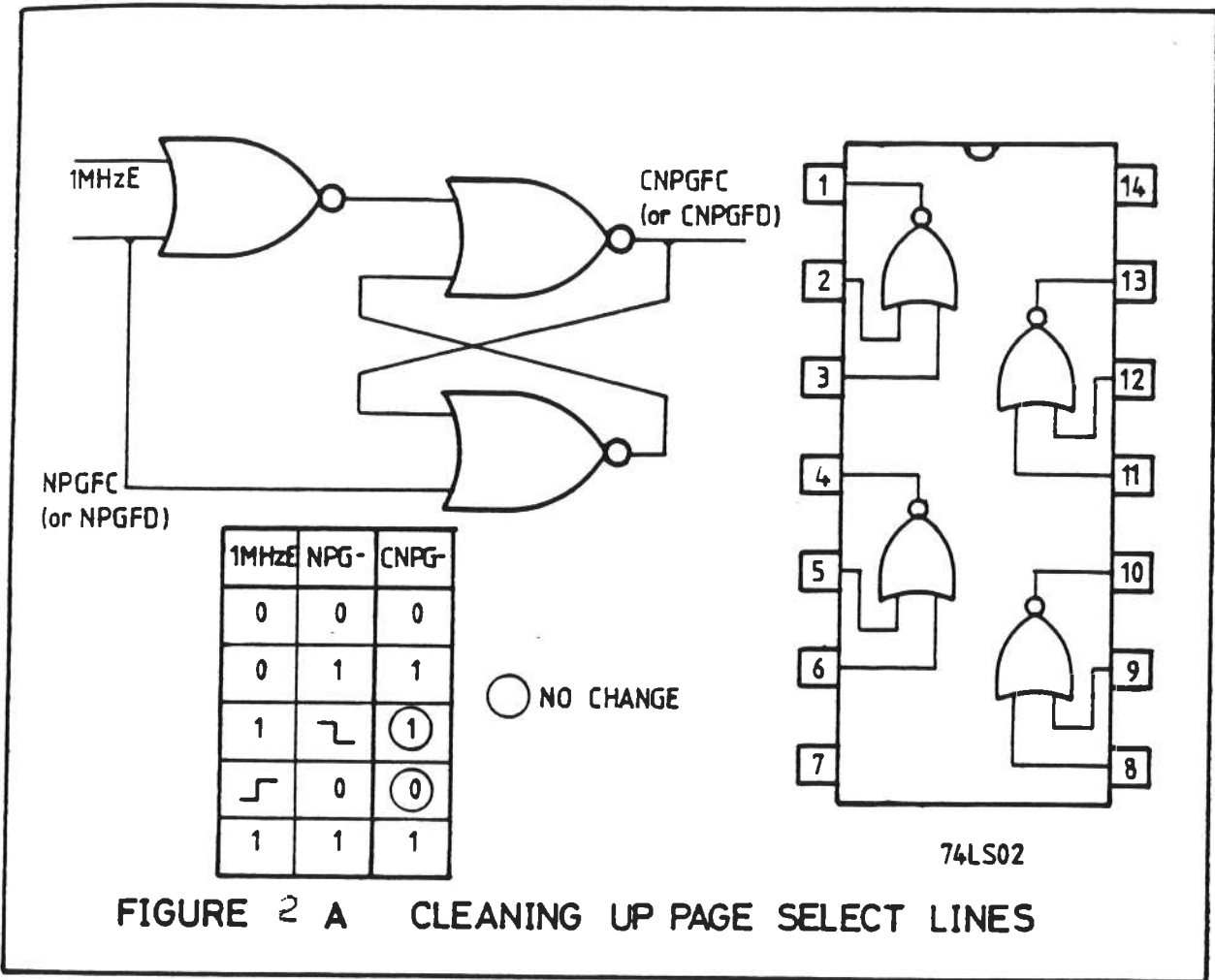


FIGURE 2 A CLEANING UP PAGE SELECT LINES

The circuit shown in figure 2 (a) is basically a gated S-R flip flop which will only allow the clean NPGFC/D signal to be low when 1MHz E is low. This same circuit can be used to clean up the NPGFD (JIM) signal which is also subject to a similar "glitch" problem.

There are also other circuits shown below in figure 2 (b) and (c). Figure 2 (b) uses a single integrated circuit and allows both FRED and JIM to be cleaned up. Figure 2 (c) is an alternative to 2 (a).

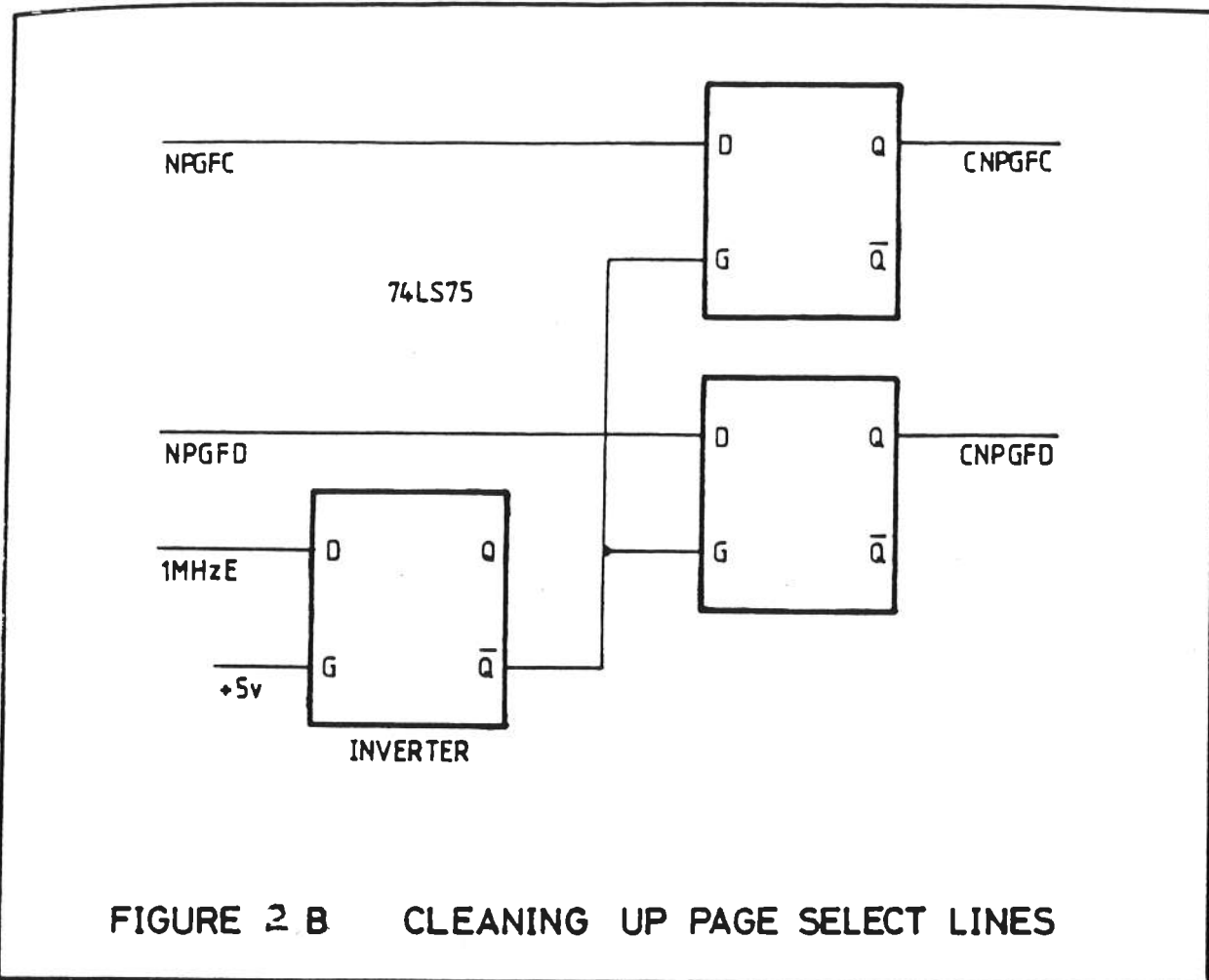


FIGURE 2 B CLEANING UP PAGE SELECT LINES

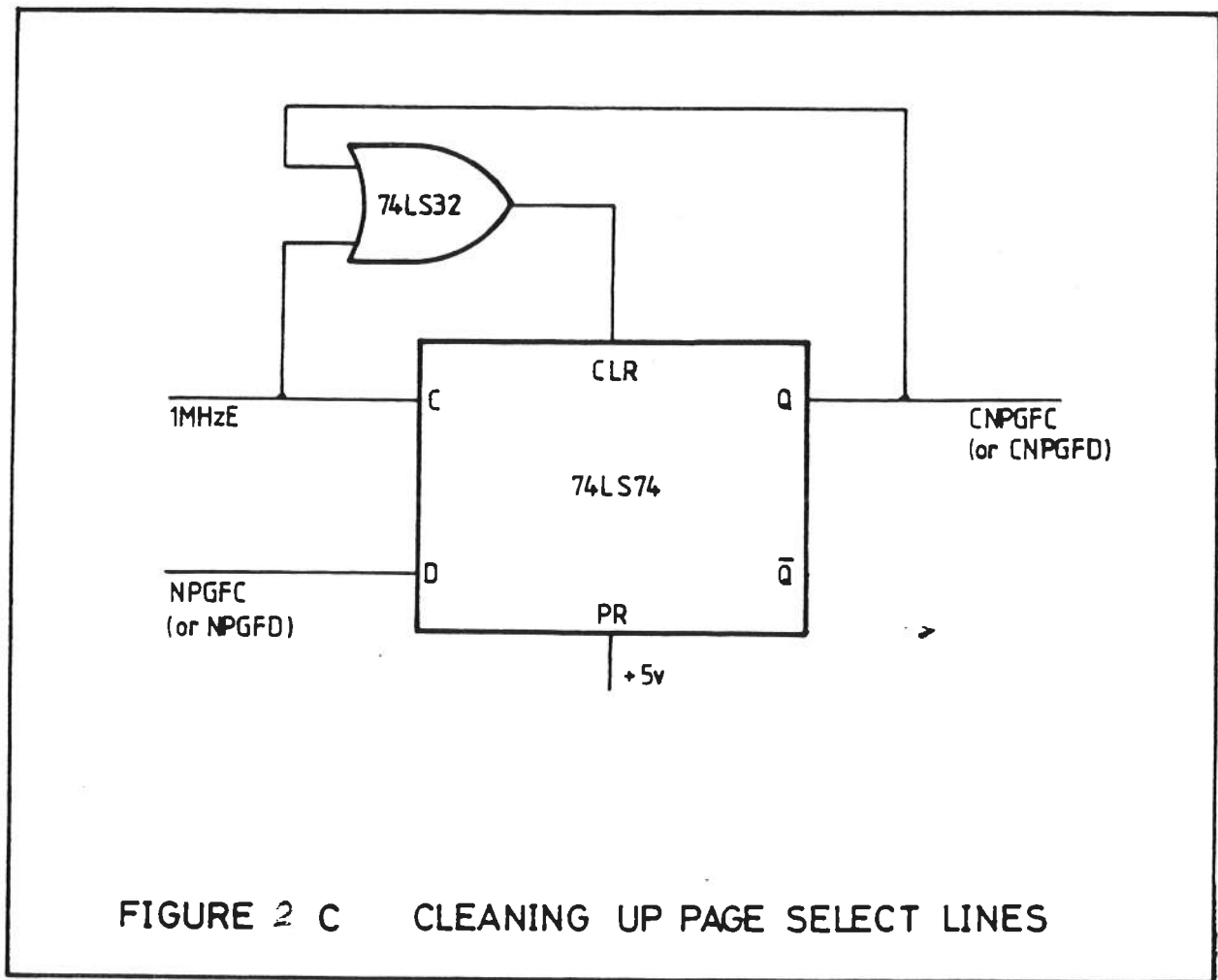


FIGURE 2 C CLEANING UP PAGE SELECT LINES



### Double Accessing.

It is possible for a peripheral device, which is attached to the 1MHz bus, to be accessed twice by the CPU. This happens as a result of the way in which the pulse stretching circuitry within the microcomputer works. Basically, it occurs because the CPU clock is in fact held high until the falling edges of the 1MHz clock and the 2 MHz clock are coincident. If a 1MHz peripheral is accessed when the 1MHz clock is high then this peripheral will be given almost immediate access to the CPU. However, because of the fact that the CPU clock is held high, waiting for the coincident edge, the peripheral device will be accessed for a second time when the 1MHz clock next goes high. This characteristic will not normally be a problem unless the peripheral is reading or writing to a device in order to change the state of an interrupt flag, for instance. This could result in an interrupt going unrecognised.

### NPGFD (JIM).

The NPGFD signal is very similar to the NPGFC line the only difference being that this line is active when the address bus holds a valid address in the range &FD00 to &FDFF. As with FRED this signal also suffers from glitches and must, therefore, be cleaned up using a similar clean up circuit to one of those shown in figure 2 .

The NPGFD line is intended to be used in conjunction with the paging register in FRED (address &FCFF) to allow the machine to address up to 64k of additional memory. This memory would normally be accessed one page (256 bytes) at a time, the paging register in FRED (at address &FCFF) would contain the page number ie; up to 256 pages each with 256 bytes. The value contained in the paging register is referred to as the "Extended Page Number" (EPN).

Acorn Computers Ltd. have adopted a convention for use of the extended pages which presently is as follows;

EPN's.	Allocation.
&00 to &7F	Reserved for use by Acorn.
&80 to &FF	General user applications.

On power up or hard break the contents of the paging register are set to &00.

## General Guidelines For Use Of The 1MHz Bus.

Acorn Computers Ltd. have issued certain guidelines which should be adhered to when designing devices to attach to the 1MHz bus, these are summarised as follows;

i) The maximum permissible loading on any one line of the 1MHz bus is 1 low power Schottky TTL load. In practise this will mean that all lines will have to be buffered by the peripheral device. It may be possible when using an expansion box that the expansion box back-plane will provide the necessary buffering requirements. The Control Universal expansion box is to be recommended in this respect.

ii) Each of the logic lines on the 1MHz bus except NIRQ, NRST and NNMI should have the facility to be terminated by a potential divider composed of two 2K2 resistors placed between 0 volts and +5 volts. Each logic line is then connected to the junction of its potential divider. In this way, bus reflections etc, will be minimised.

iii) All peripheral devices should have their own power supply source and should not draw power from the BBC Microcomputer (via the 1MHz bus).

iv) The timing relationship of the logic signals on the 1MHz bus may alter when fully loaded. This will happen as a result of the increased rise and fall times which occur as the lines become more capacitively and inductively loaded. The set up times for the address bus and for FRED and JIM will have a minimum value of 300ns and a maximum value of 1000ns. Any attached peripheral device should, therefore, be able to cope with this.

v) The recommended method of connecting the peripheral device to the 1MHz bus is via a 600mm length of 34 way ribbon cable terminated at each end with a 34 way IDC header socket ie; RS part no. 467-302. The peripheral device should allow all of the signal lines to "pass through" and fed to an on-board 34 way header plug ie; RS part no. 467-368 or 467-992. This arrangement allows additional peripheral devices to be "daisy chained".





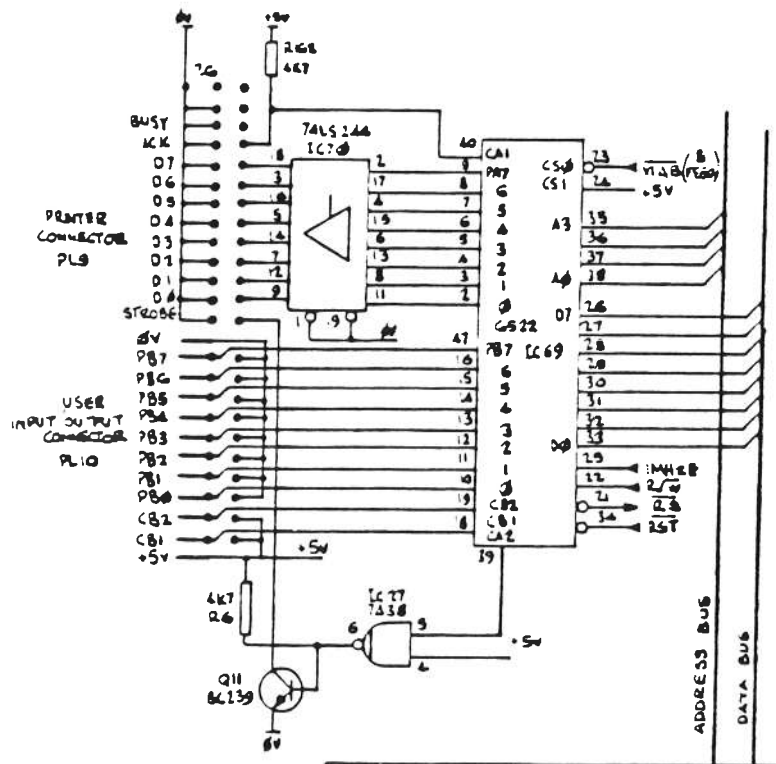
**Servicing the BBC Micro  
Printer and User Port Tests**



## PRINTER AND USER PORT TEST S.

Below is the circuit diagram of the printer and user ports (Fig 1).

Fig 1.



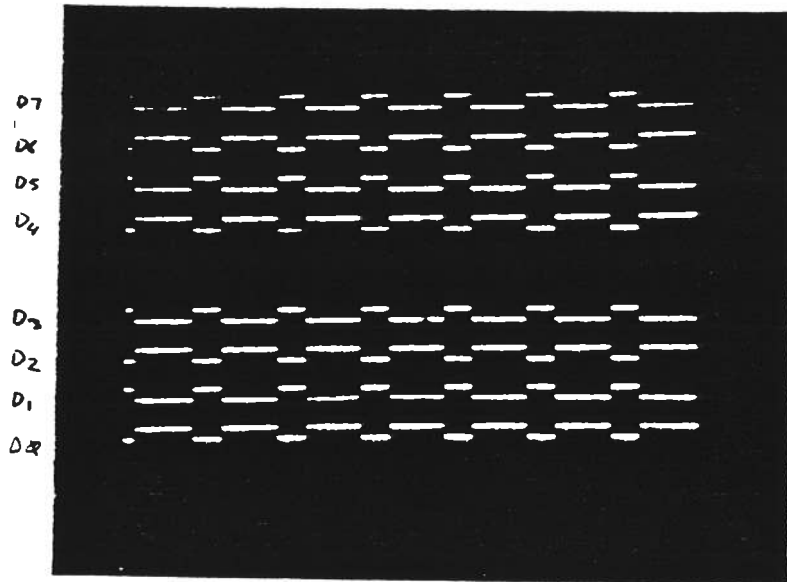
Before starting to test the printer and the user port circuits a number of quick checks should be carried out on the CPU side of IC69.

1. A square wave clock of 1MHz should be found at pin 25
2. Address lines A0 to A3 should be active ie. 0v to approx. +4v pulses on each line.(pins 35 to 38)
3. Data lines D0 thru D7 should also be active again 0v to approx. +4v on each line.(pins 26 to 33)
4. R/W should have 0v to approx +4v pulse and will go high and stay in that condition while the break is being pressed.(pin 22)
5. RST should be high +4v and will go low while the break key is being pressed(pin 34)
6. IRQ will have pulses +5v to 0v and will go to 0v while the break is being pressed.
8. CS0 should be sitting high approx. +4v this will be activated when any of the programs below are run which should cause sharp pulses going from +4v to 0v.
9. CS1 is high +5v
10. Always check supply voltages are present on the IC'S under test.



PRINTER DATA LINES.

FIG. 2.



On FIG. 2. there are eight pulse traces, each trace represents a printer data line when the test program is running. It can be seen that each alterenate trace is identical ie. D0 is the same as D2 and D1 is the same as D3 this makes it very easy using a dual beam oscilloscope to detect a bad data line. It is also worth noting that adjacent data lines are equal but opposite ie. the pulses on line D0 will be high for the same duration as the pulses on line D1 are low and vise-versa. this pattern repeated on all the adjacent printer data lines during this test.

FIG. 3.

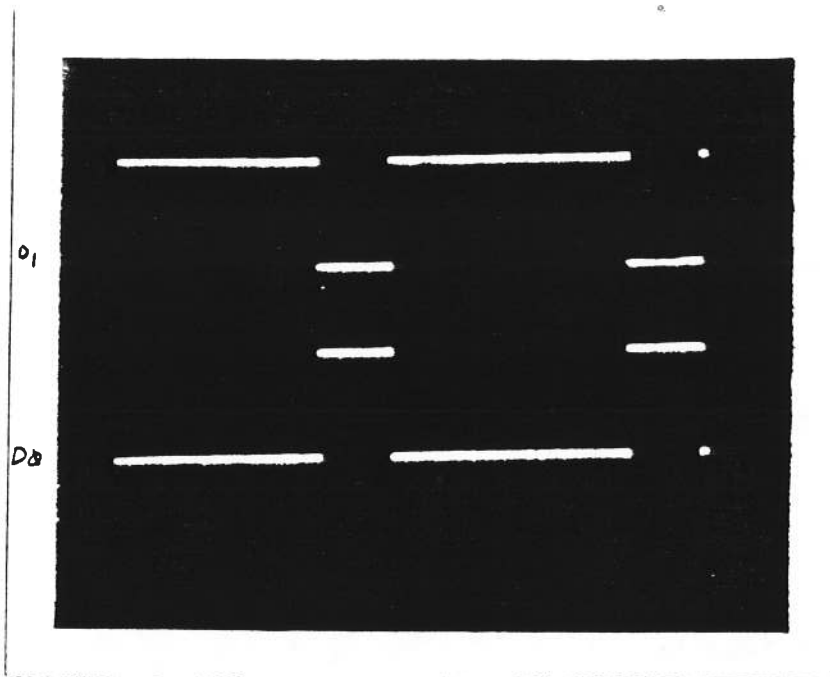


Fig.3 shows the trace of both D0 and D1.it can therefore be seen that a quick check on the printer data lines can be achieved by hooking one channel of the dual beam scope to D0 or any other data line on the printer connector PL9 and use the other channel to compare with the remaining data lines on PL9, remembering that alternate lines are identical and adjacent lines equal but opposite.The traces on Fig.3 were acheived with a time base setting of 10ms/cm.

If the anticipated signal is not found on any of these lines do the same check on the other side of IC70 ie. if no signal was found on line D0 at PL9 or pin 9 of IC70 then check pin 11 of IC70 if the signal is there then replace IC70 and if not check pin 2 of IC69 again if the signal is not found check the continuity of line D0 between IC69 and IC70 if continuity is found to be good then replace IC69.

Below is the program used to test the printer port.

```
10 REM THIS PROGRAM SENDS &55 AND &AA TO THE PRINTER PORT
20 REM IT ALSO CHECKS STATUS LINES CA1,CA2.
30 CLS
40 VDU 23,1,0;0;0;0;
50 PRINT 'TAB(5,6);CHR$(141);CHR$(131);"RUNNING"TAB(5,7);CHR$(141);
CHR$(131);"RUNNING"
55
60 ?&FE63=&FF: REM MAKE ALL PRINTER DATA BITS OUTPUTS.
70 ?&FE6C=&0C: REM PUT A PULSE OUT ON CA2 AND SET
80 ?&FE6C=&0E: REM CA1 TO RECOGNIZE A FALLING EDGE.
90 ?&FE6C=&0C
95
100 Z=?&FE6D: REM READ THE INTERRUPT FLAG REGISTER.
110 IF Z=&02 THEN MES$=STRING$(30," ")
ELSE MES$=CHR$(129)+"CHECK STATUS LINES CA1,CA2."
120 PRINT TAB(5,10);CHR$(141);MES$TAB(5,11);CHR$(141);MES$
130 X=&55
140 Y=&AA
150 ?&FE61=X:FOR N=1 TO 20:NEXT: REM PUT THE VALUES OF X AND Y
160 ?&FE61=Y:FOR N=1 TO 20:NEXT: REM OUT TO THE PRINTER PORT.
170 GOTO 60
```

## USER PORT TEST.

The user port test is almost identical to that of the printer port except that a small program is added which inputs from the port and prints the value on the screen.

This means that the data lines (PB0 to PB7) can be held at a known value and read on screen. This check is necessary to ensure that the CPU can read from the VIA. (IC69).

As before check the port with the output test which also checks status lines CB1, CB2. (short pins 2 and 4 PL10) again choose a data line to hook one channel of your dual beam scope onto, and use the other channel to compare with the remaining data lines remembering alternate lines are equal, and adjacent are equal but opposite as in FIG3.

The good thing about the user port ie. from a testing point of view, is that unlike the printer port there is no data line buffer. This means that if the desired signal pattern on PB0 thru PB7 on IC69 is correct, and the same pattern is not found at PL10.

Then it is fair to presume that you have a continuity problem between IC69 and PL10 (check with multi-meter).

But if the signal was incorrect at IC69 then replace the chip.

Below is the user port output test program.

```
10 REM THIS PROGRAM SENDS &55 AND &AA TO THE USER PORT
20 REM AND ALSO CHECKS STATUS LINES CB1,CB2.
30 CLS
40 VDU 23,1,0;0;0;0;
50 PRINT 'TAB(5,6);CHR$(141);CHR$(131);"RUNNING"TAB(5,7);
CHR$(141);CHR$(131);"RUNNING"
55
60 ?&FE62=&FF: REM MAKE ALL USER PORT DATA BITS OUTPUTS
70 ?&FE6C=&D0: REM PUT A PULSE OUT ON CB2 AND SET CB1
80 ?&FE6C=&F0: REM TO RECOGNISE A RISING EDGE
90 ?&FE6C=&D0
95
100 Z=?&FE6D: REM READ INTERRUPT FLAG REGISTER
110 IF Z=&10 THEN MES$=STRING$(30," ")
ELSE MES$=CHR$(129)+"CHECK STATUS LINES CB1,CB2."
120 PRINT TAB(5,10);CHR$(141);MES$TAB(5,11);CHR$(141);MES$
130 X=&55
140 Y=&AA
150 ?&FE60=X:FOR N=1 TO 20:NEXT:REM PUT THE VALUES OF X AND Y
160 ?&FE60=Y:FOR N=1 TO 20:NEXT:REM OUT TO THE USER PORT
170 GOTO 60
```

Below is the user input port test program.

```
10 REM THIS PROGRAM READS FROM THE USER PORT AND PRINTS
20 REM THE VALUE ON THE SCREEN
30 CLS
40 VDU 23,1,0;0;0;0;
50 PRINT 'TAB(5,6);CHR$(141);CHR$(131);"RUNNING"TAB(5,7);
CHR$(141);CHR$(131);"RUNNING"
55
60 ?&FE62=&00: REM MAKE ALL USER PORT DATA BITS INPUTS
70 N=?&FE60: REM READ THE VALUE OF PORT DATA BITS.
75
80 PRINT 'TAB(5,10);CHR$(141);CHR$(130);"USER PORT = ";~N;" HEX"
'TAB(5,11);CHR$(141);CHR$(130);"USER PORT = ";~N;" HEX"
90 GOTO 70
```









**Servicing the BBC Micro**

**Cassette Interface Circuitry Protection**

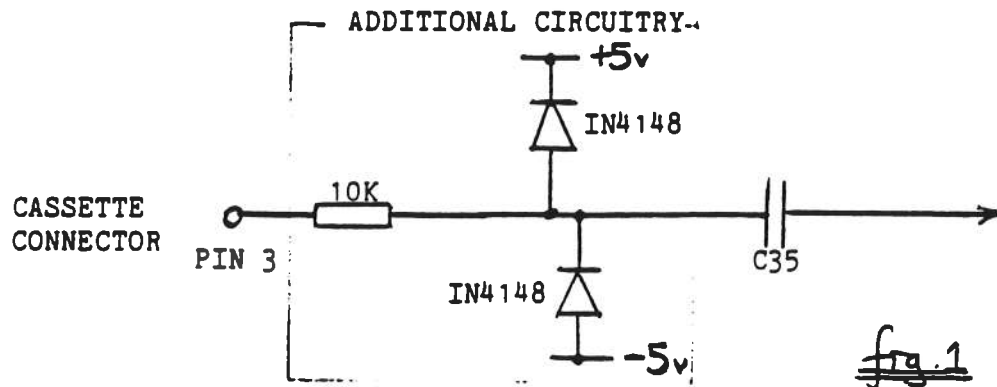


### 3.5 CASSETTE INTERFACE CIRCUITRY PROTECTION.

It has come to our notice that the cassette interface of the BBC Microcomputer can in some cases be damaged by certain tape recorders. The following additional circuitry can be added to any board in order to minimise this problem. (fig. 1)

#### COMPONENTS REQUIRED:

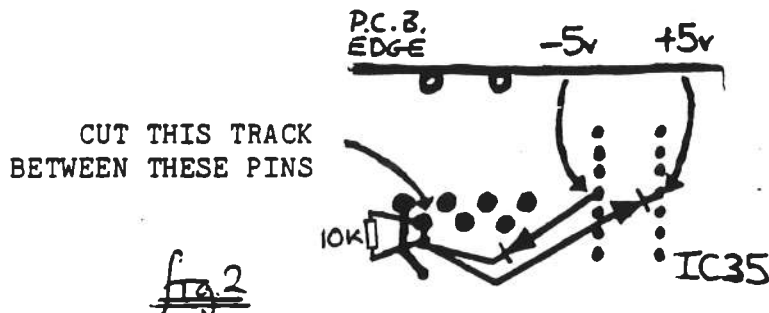
- 2 x 1N4148 Diode
- 1 x 10K  $\frac{1}{4}$  Watt 10% Carbon Film Resistor



#### MODIFICATION:

The modification is performed as follows:

- i) Ensure the computer is disconnected from the mains
- ii) Disconnect the 7 power supply connectors from the main PCB, and disconnect or unsolder (as appropriate) the two connectors to the Video Out socket.
- iii) Unscrew the PCB and remove it from the case. Place it upside-down on a flat surface.



- iv) Identify the cassette socket and IC35 as shown in fig 2 above. Cut the track marked, and fit the 10K resistor and the two diodes as shown. Make sure that the leads from the diodes do not touch

any other of the pins of IC35, preferably by insulating them with a short length of heat-shrink sleeving or wire insulation.

- v) Re-install the main PCB by reversing (ii) and (i) above. This completes the modification.





**Servicing the BBC Micro  
In Circuit Emulation**





## IN CIRCUIT EMULATION

The microprocessor socket usually provides all the address, data and control bus lines needed to gain access to most of a microcomputer system. The technique of IN CIRCUIT EMULATION makes use of this by removing the processor from the unit under test and imitating or emulating its functions using another computer.

The HOST system is connected to the unit under test (TARGET) through a dual in-line plug and ribbon cable. Normally the host computer uses a similar microprocessor to that removed from the target to provide all the signals that are required by the target system. At first glance, this might seem pointless, removing the original microprocessor to have it replaced by another in the host. However, the technique allows the host to 'get between' the target system and its microprocessor enabling it to monitor bus activity or to inject signals into the target system.

## AS A DEVELOPMENT TOOL

The technique was first introduced by INTEL on their microprocessor development system (MDS) to help debug hardware and software problems encountered during the development phase of a project. Early development systems were essentially software orientated providing editors, assemblers, linkers and to a limited extent some facility to execute and debug the final machine code. However, to effectively test hardware and software it was necessary to commit the final code to EPROM before transferring it to the unit under development. Any further debugging that was required was now carried out using a logic analyser.

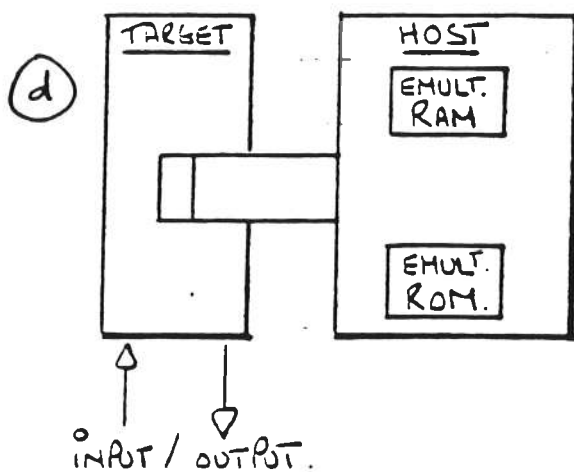
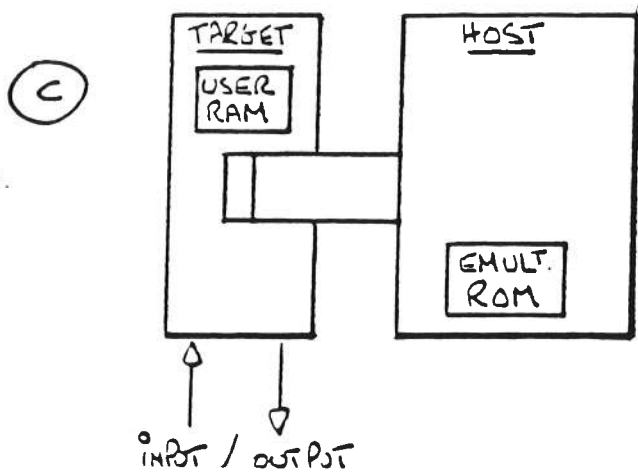
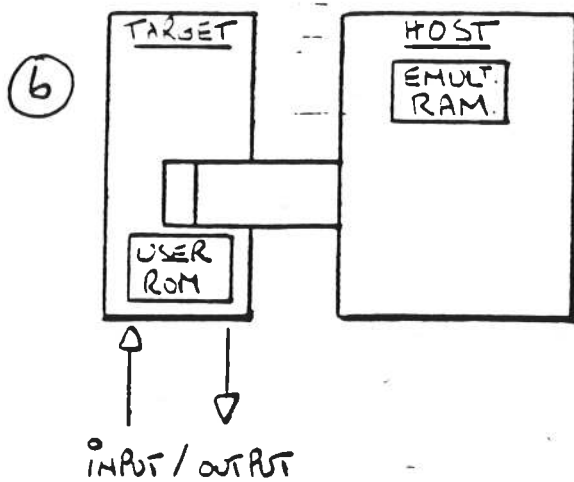
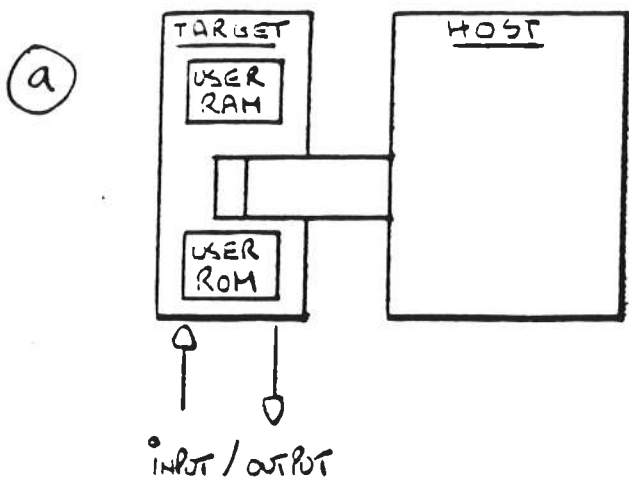
If faults were discovered the engineer returned to the development system to update the software before programming a new EPROM.

The introduction of ICE into microprocessor development systems shortened this time consuming procedure, allowing the design engineer to run the system under development directly from the MDS thus eliminating the need to use EPROMS as a means of transporting software into the target system. Further since the MDS has access to the target systems buses during program execution it can also be used as a debugging tool. To help with this task most microprocessor development systems also provide on-board logic analyser functions.

In circuit emulation is now a standard feature on most development systems with some manufacturers, including Hewlett Packard, Genrad and Tektronix, providing a range of emulators covering a variety of 8 bit and 16 bit microprocessors.

Figure ( | ) gives examples of possible emulation configurations between two systems. In (a) the host system takes on the job of imitating the microprocessor only. All the memory used by the target system lies on the target board (termed USER memory). Example (c) illustrates how the memory map of the target system can be made up of its own hardware, USER RAM, and memory from the host system, EMULATION ROM.

Finally examples (e) and (f) show how the peripheral devices to the host, namely disc, keyboard, printer etc., can be used to simulate input and output signals to the target system.



SIMULATED INPUT / OUTPUT.

INPUT FROM - DISC / KEYBOARD / RS 232.  
OUTPUT TO - DISC / DISPLAY / PRINTER / RS 232.

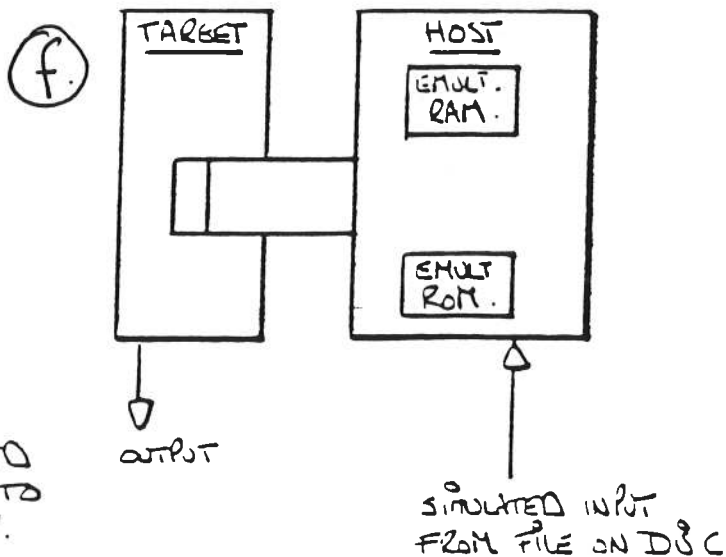
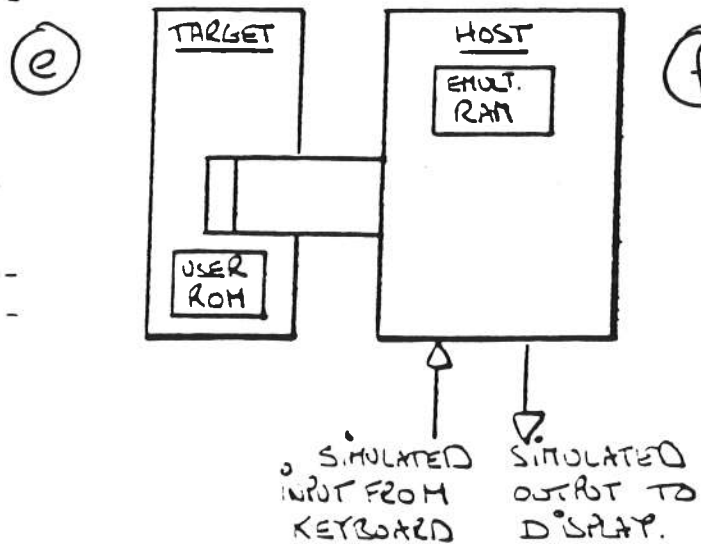


FIGURE ( ) Possible EMULATION CONFIGURATIONS

## AS A FAULT FINDING TOOL

A microprocessor development system offers an extremely powerful fault finding tool but the number of service personnel with access to this type of equipment is limited. Other factors, including physical size, the need for highly skilled operators as well as price, rule out the MDS as a service tool. However, several manufacturers, Applied Microsystems, Fluke, Hewlett Packard, Millenium Solartron etc., market stand-alone emulators, some of which are ideally suited to the servicemans needs,

To the service engineer in circuit emulation provides a method of gaining control of the unit under test, and injecting test or stimulus programs onto its buses. This can prove an essential facility if the unit under test does not respond to normal keyboard operation or perhaps does not contain a keyboard. Once in control the emulator can run routines to check the operation of the targets systems functional components using routines similar to those employed in self-testing. If

faults are discovered the emulator's display reports its location giving the serviceman clues to the possible source of trouble. The Fluke troubleshooter, is an example of this type of emulator. However, not all emulators contain a keyboard and a display. For example the Solartron Micropod is designed to be used with test software developed on another machine and loaded into the emulator in EPROM. The test routines exercise the target system generating data streams that <sup>maybe</sup> ~~are~~ monitored using a signature analyser.





**Servicing the BBC Micro**  
**Tracing Processor Execution**





## EXPERIMENT

Using a Logic Analyser to trace processor address bus activity and data bus activity during read operations.

## EQUIPMENT

BBC micro with monitor, LJ Logic analyser

## PROCEDURE

a) Connect the 16 trigger word inputs of the Logic Analyser as indicated below:-

1/P's 0-7 to the 6502 (IC1), AB0-AB7  
1/P's 8-15 to the 6502, AB8-AB15  
Connect the OV to Pin 21 of IC1  
Connect the clock 1/P to  $\phi$ 2 (P<sup>in</sup> 39 of IC1)  
Leave the two qualifiers unconnected.


Enter and run the following program

```
10P%=&2800
20 [ .LOOP LDA #&88
30     STA &2050
40     LDA #&AA
50     STA &2850
60     LDA &2050
70     LDA &2850
80     JMP LOOP
90 ]
```

This should give the following output

```
2800 A9 88     .LOOP LDA #&88
2802 8D 50 20  STA &2050
2805 A9 AA     LDA #&AA
2807 8D 50 28  STA &2850
280A AD 50 20  LDA &2050
280D AD 50 28  LDA &2850
2810 4C 00 28  JMP LOOP
```

Set the trigger word to 2800 with all four disable/enable switches set to enable.

Set clock edge to rising ie   
clock qualifier to X  
trigger qualifier to X

Arm the analyser.

Start the program by CALL &2800

Write down the first 20 numbers which have been stored by the analyser

0	10
1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19

Explain the sequence in terms of the program.

b) Connect I/Ps 0 to 7 to data lines DB0 to DB7: Do not disconnect  $\psi$ s 8-15  
Connect the clock qualifier to the R/ $\bar{w}$  pin (pin 34 of IC1)

Break the program. Enter OLD and LIST

Check that the program has not been corrupted by transients during the changing of connections. RUN the program.

Set the trigger word to 28A9 with all switches set to enable.  
(First data item read A9, from address 2800.)

Set clock edge to falling ie  $\lrcorner$   
clock qualifier to X  
trigger qualifier to X

Arm the analyser.  
Start the program.

Copy out the first 26 least significant bytes of data which have been clocked in to the analyser.

0	13
1	14
2	15
3	16
4	17
5	18
6	19
7	20
8	21
9	22
10	23
11	24
12	25

Relate the data sequence to the program.

c) Set the ~~trigger~~<sup>clock</sup> qualifier to 1. All other settings and connections are as before

Rearm the analyser. It should trigger again immediately on 28A9. (If

not Break, enter OLD and LIST, check the program, RUN it again and CALL&2800)

Copy out the first 26 least significant bytes which have been clocked in to the analyser

0	13
1	14
2	15
3	16
4	17
5	18
6	19
7	20
8	21
9	22
10	23
11	24
12	25

Explain the differences from the previous data sequence.

d) Set the trigger word to 2088. Set the clock qualifier to 0. All other setting and all connection are as before. Rearm the analyser which should trigger immediately (If not follow the same procedure as before)

Note the first six least significant bytes. Explain these

1  
2  
3  
4  
5  
6

## EXPERIMENT

Using a Logic Analyser to trace data bus write activity.

## EQUIPMENT

BBC Micro with VDU. LJ Logic Analyser

## PROCEDURE

- a) Connect the 16 trigger I/P's to AB0-AB7 and AB8-AB15 on the 6502 (chip IC1).

Connect the 0V to IC1, Pin 21.

Connect the clock to  $\phi$ 2 (IC1, Pin 39.)

Connect the clock qualifier to R/w (IC1 pin 34)

Enter and run the following program.

```
10P%=&2800
20 [.START LDA #&00
30     STA &70
40     LDA #&25
50     STA &71
60     LDA #&FF
70     LDX #&00
80     LDY #&80
90     .LOOP STA (&70),Y
100    STY &72,X
110    STX &A0,Y
120    SEC
130    SBC #&01
140    INX
150    DEY
160    CPX #06
170    BNE LOOP
180    JMP START
190 ]
```

This should result in the output:

```
2800 A9 00     .START LDA #&00
2802 85 70     STA &70
2804 A9 25     LDA #&25
2806 85 71     STA &71
2808 A9 FF     LDA #&FF
280A A2 00     LDX #&00
280C A0 80     LDY #&80
280E 91 70     .LOOP STA (&70),Y
2810 94 72     STY &72,X
2812 96 A0     STX &A0,Y
2814 38        SEC
2815 E9 01     SBC #&01
2817 E8        INX
2818 88        DEY
2819 E0 06     CPX #&06
```

```
281B D0 F1   BNE LOOP
281D 4C 00 28 JMP START
```

Set trigger word to 2800  
clock edge to rising ie  $\lrcorner$   
set qualifiers to X  
Arm the analyser.

Start the program with CALL&2800 and record the first 32 addresses captured by the analyser, checking against the program instructions.

0	8	16	24
1	9	17	25
2	10	18	26
3	11	19	27
4	12	20	28
5	13	21	29
6	14	22	30
7	15	23	31

Now connect I/Ps 0 to 7 to data lines DB0 to DB7. Break the program. Enter OLD and LIST. Check the program has not been corrupted. RUN the program.

Set trigger word to 25FF  
Set clock edge to falling-ie  $\lrcorner$   
Set clock qualifier to 0

Arm analyser

Start the program with CALL&2800

Record the first 30 least significant bytes clocked by the analyser. These are write data. Explain.

0	10	20
1	11	21
2	12	22
3	13	23
4	14	24
5	15	25
6	16	26
7	17	27
8	18	28
9	19	29

## EXPERIMENT

Using the Logic Analyser to capture data read from Read Only memory.

## EQUIPMENT

BBC Micro with Monitor. LJ Logic Analyser

## AIM

A program will be run to read data from the read only memories on the BBC board. The Logic Analyser clock will be qualified to ensure that program information is not clocked into the analyser.

## PROCEDURE

Powerdown to DB0-DB7  
Connect the trigger I/P's to 0-7 and AB8-AB15 of the 6502, and the 0V to Pin 21, IC1.

Connect clock to 02(chip IC1, pin 39)  
clock qualifier to the chip select signal CS (usually Pin20) of ROM  
the trigger qualifier to IC69 Pin2

Power up, enter and RUN the following program. This loads a machine code program which reads out all ROM data from &8000 to &FFFF. It also generates a positive edge on PA0 of the VIA chip IC69 to be used as a trigger. (Note - remarks after ; are not essential).

```
10P%=&2800
20 [   LDA #&01
30     STA &FE61      ;TRIGGER.
40 .STRT LDY #&00     ;SET UP BASE
50     STY &70       ;ADDRESS &8000
60     LDA #&80      ;IN MEMORY LOCATIONS
70     STA &71       ;&70 AND &71.
80 .LOOP LDA (&70),Y ;READ ROM.
90     INY           ;INCREMENT BYTE POINTER.
100    CPY #&00      ;END OF PAGE?
110    BNE LOOP      ;IF NOT LOOP BACK.
120    INC &71       ;INCREMENT PAGE POINTER.
130    LDA #&00
140    CMP &71       ;ALL ROM READ
150    BNE LOOP      ;IF NOT LOOP.
160    JMP STRT      ;DO IT ALL AGAIN.
170]
```

Running the program should give the following output.

```
2800 A9 01   LDA #&01
2802 8D 61 FE STA &FE61      ;TRIGGER
```

```

2805 A0 00 .STRT LDY #&00 ;SET UP BASE
2807 84 70 STY &70 ;ADDRESS &8000
2809 A9 80 LDA #&80 ;IN MEMORY LOCATIONS
280B 85 71 STA &71 ;&70 AND &71.
280D B1 70 .LOOP LDA (&70) ,Y ;READ ROM.
280F C8 INY ;INCREMENT BYTE POINTER.
2810 C0 00 CPY #&00 ;END OF PAGE?
2812 D0 F9 BNE LOOP ;IF NOT LOOP BACK.
2814 E6 71 INC &71 ;INCREMENT PAGE POINTER.
2816 A9 00 LDA #&00
2818 C5 71 CMP &71 ;ALL ROM READ?
281A D0 F1 BNE LOOP ;IF NOT LOOP.
281C 4C 05 28 JMP START ;DO IT ALL AGAIN.

```

Set clock edge to falling.

Set clock qualifier to zero.

Set trigger qualifier to one.

Disable all of the trigger word (the trigger word pattern is immaterial)

Arm the analyser.

Start the routine with CALL&2800

Record the first ten ~~alternate~~<sup>?</sup> least significant bytes.  
 Compare these with the contents of the selected ROM.

0	5
1	6
2	7
3	8
4	9

## EXPERIMENT

Using the Logic Analyser to trace processor activity when the stack is being used.

## EQUIPMENT

BBC Micro with monitor, LJ Logic analyser.

## PROCEDURE

Connect I/Ps 0-15 to AB0-AB15 on the 6502  
Connect the 0V to IC1 pin 21  
Connect clock to  $\phi$ 2 (IC1 pin 39)  
Connect clock qualifier to R/ $\bar{w}$  (IC1 pin 34)

Enter and RUN the following program

```
10 FOR PASS=1 TO 3 STEP 2
20 P%=&2800
30 [ OPT PASS
40         LDX #&20
50         TXS
60         LDA #&FE
70         STA &2100
80 .LOOP   CLC
90         ADC #&01
100        JSR SUB
110        JMP LOOP
120 ]
130 :
140 REM SUBROUTINE
150 :
160 P%=&2800&2850
170 [.SUB   PHA
180        PHP
190        STA &2100
200        FLP
210        FLA
220        RTS
230 ]
240 NEXT PASS
```

This should give the output

```
2800          OPT PASS
2800 A2 20     LDX #&20
2802 9A       TXS
2803 A9 FE     LDA #&FE
2805 8D 00 21 STA &2100
2808 18       .LOOP   CLC
2809 69 01     ADC #&01
280B 20 50 28 JSR SUB
280E 4C 08 28 JMP LOOP
2850 48       .SUB     PHA
```



2851	08	PHP
2852	8D 00 21	STA &2100
2855	28	PLP
2856	68	FLA
2857	60	RTS

- a) Set the trigger word to 2800.  
 Enable all inputs.  
 Set trigger qualifier to X.  
 Set clock qualifier to X.  
 Set clock edge to rising.

Arm analyser.

Start the routing<sup>e</sup> with CALL&2800. Note the first 30 addresses recorded by the analyser. Explain these in terms of the program.

0	10	20
1	11	21
3	13	23
4	14	24
5	15	25
6	16	26
7	17	27
8	18	28
9	19	29

- b) Break the program. Enter OLD.

Transfer the I/Ps 0-7 to DB0-DB7. All other connections are as before.

List the program to ensure it has not been corrupted.  
 RUN the program

Set the trigger word to 21FE. All inputs enabled.  
 Set clock qualifier to zero to capture write data.  
 Set clock edge to falling.  
 Set trigger qualifier to X.  
 Start program with CALL&2800

Note the first 16 bytes of write data recorded.  
 Explain in terms of program execution

0	8
1	9
2	10
3	11
4	12
5	13
6	14
7	15







# **Servicing the BBC Micro**

## **Logic Analysis**



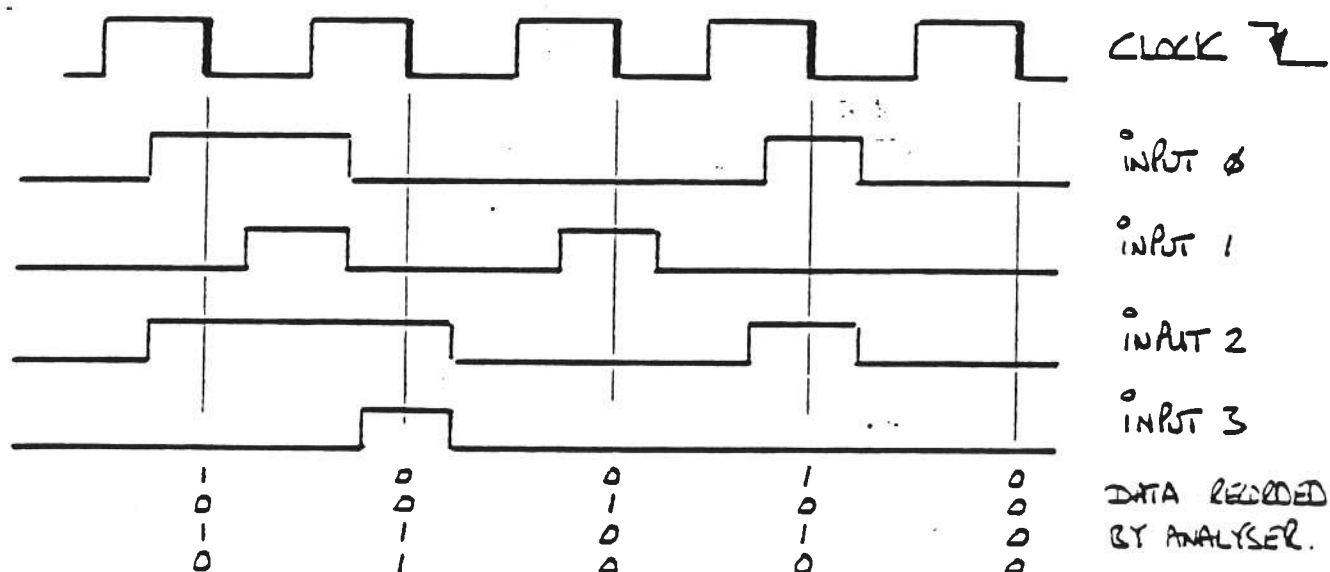
# **Logic Analysis**

## LOGIC ANALYSERS

Logic analysers are designed to record the logic levels at several points (typically 4, 8, 16 or 32) in a computer system. Their input channels can be used to monitor the time behaviour of any of the signals found in a microprocessor system.

e.g.      Data bus  
             Address bus  
             Control bus  
             Input/output ports.

Data is not continuously recorded but is sampled and stored in the analyser's own memory under the control of a clock signal.



Logic STATE analysers normally use the clock of the system under test. Each time the external circuit generates a clock pulse the analyser records the logic levels on its input channels. Some analysers have a feature called TIMING ANALYSIS, where the analyser uses its own internal clock, running much faster than the system clock, to control the capture of data. With this feature it is possible to study in greater detail the timing relationships between channels.

The triggering circuits used by analysers are much more sophisticated than those found in oscilloscopes. When armed, the analyser continuously monitors its input channels and triggers when a previously defined TRIGGER WORD occurs. Depending on the sophistication of the analyser, various 'extra' conditions can be placed on the trigger word.

- e.g. - Trigger after N occurrences of the trigger word  
       - Trigger N clock cycles after the trigger word.

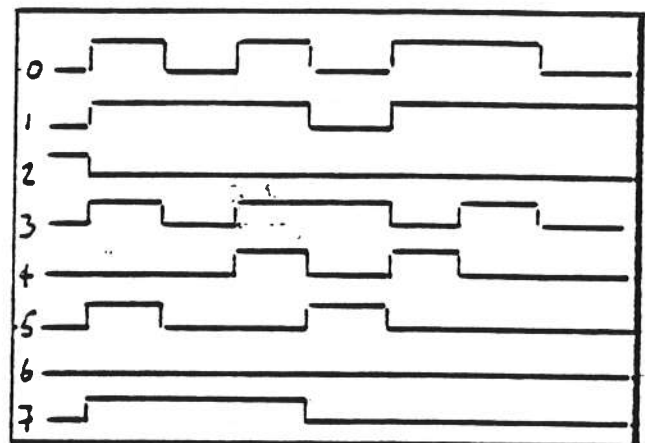


Like oscilloscopes, logic analysers can display the data that occurs after the trigger event. Unlike most scopes, analysers can also display data prior to the trigger event. This capability is called NEGATIVE TIME recording and can be used for trouble shooting by choosing a faulty system operation as the trigger event and then observing the events that led up to it.

Most analysers will give you a choice of display mode for the captured data. The diagrams below show the two most common formats.

CLOCK CYCLE	BINARY	
00	0000	0100
01	1010	1011
02	1000	0010
03	1001	1011
04	0010	1000
05	0001	0011
06	0000	1011
07	0000	0010
08	0000	0000

DATA DISPLAYED IN BINARY



DATA DISPLAYED IN TIMING MODE

### L.J. LOGIC ANALYSER

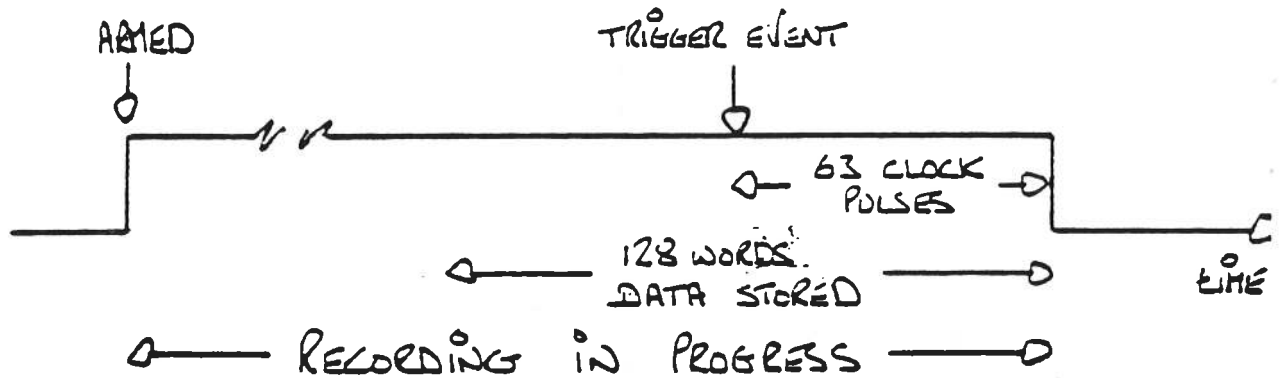
The L.J. Logic Analyser is a low cost 16 channel state analyser with a maximum clock frequency of 4 MHz. Data display is by either a 4 digit LED hexadecimal readout or via co-axial cables to an oscilloscope. Circuit connection is by a 22 way ribbon cable containing the following:

- |                          |          |
|--------------------------|----------|
| (a) 3 ground connections | (black)  |
| (b) 16 inputs            | (red)    |
| (c) Clock                | (yellow) |
| (d) Clock Qualifier      | (green)  |
| (e) Trigger Qualifier    | (blue)   |

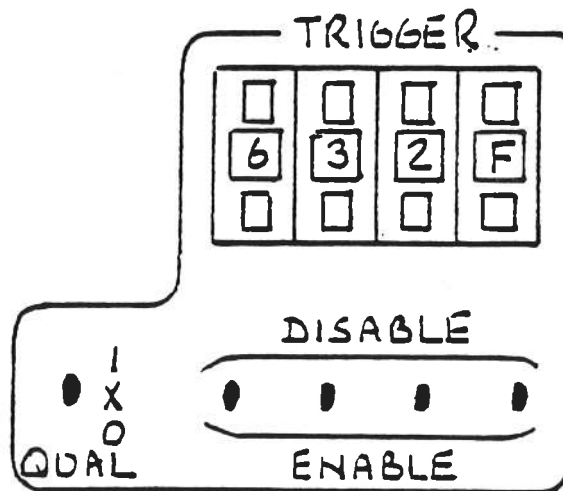
## OPERATION

### Arm/Abort Control

Depressing the ARM control initiates a data recording sequence to capture a new set of data. While recording is in progress the displays are blanked, the ARM LED is illuminated and data is stored in memory using the external clock. This state remains unchanged until the trigger event is encountered, when the ARM LED will be extinguished. Sixty-three clock cycles later the recording mode will be terminated. Alternatively, if the trigger event does not occur, the ABORT control can be operated. This will immediately terminate the recording mode.



### Recording Sequence - Trigger & Trigger Qualifier Controls



### Trigger Controls

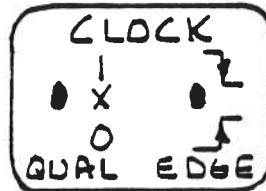
These controls are used to select the particular event that will cause the analyser to trigger and therefore define the section of input data which is stored in memory.

The trigger event is set by the 4 thumbwheel switches and the single line trigger qualifier switch. Each digit can be set in hexadecimal code (0 - F) and enabled or disabled depending upon the required trigger condition.




The trigger qualifier can be used as a 17th bit extension of the trigger word, as a bit mask on one of the trigger digits, or as an external trigger input by completely disabling the 4 digit trigger.

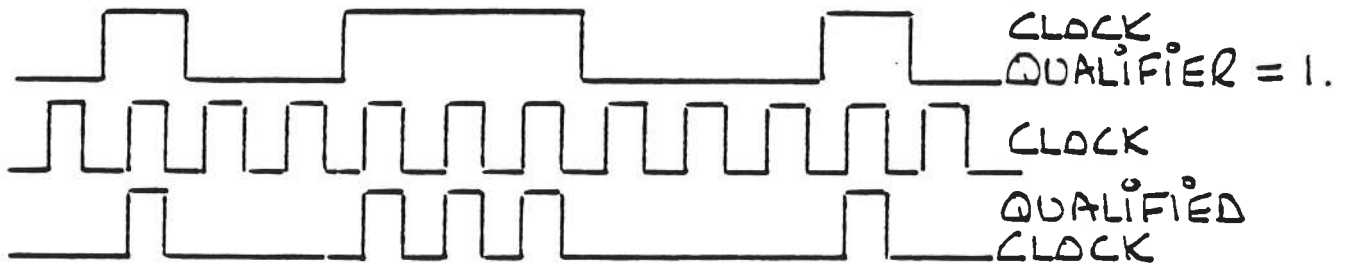
The data stored in memory contains 64 words of pre-trigger data, and 63 words of post-trigger data.

Clock and Clock Qualifier Controls



Clock Controls

The analyser requires a clock to store data into the memory. It accepts an external clock at rates from DC to 4 MHz with an action edge selectable between  and . If the active edge of the users system clock is negative-going, then the analyser should be run on  so that the system data is stable and meets the required set-up and hold times of the instrument. The ANDed clock qualifier enables data to be selectively recorded during valid memory addresses or during read or write operations in microprocessor circuits.



Qualified Clock Generation

Should recording be attempted when there is no clock signal present, or when the clock is present but not qualified, then an LED in the cursor window will flash.

Data Display

The analyser has two separate yet complimentary display media: the 4 digit display with hexadecimal readout; and a timing diagram output for display on an oscilloscope.

The hexadecimal data display shows one word at a time with

its position in memory, relative to the trigger word, indicated by the two digit cursor display. This has a range of -64 to +63. The INC/DEC switch is used to examine the data one word at a time.

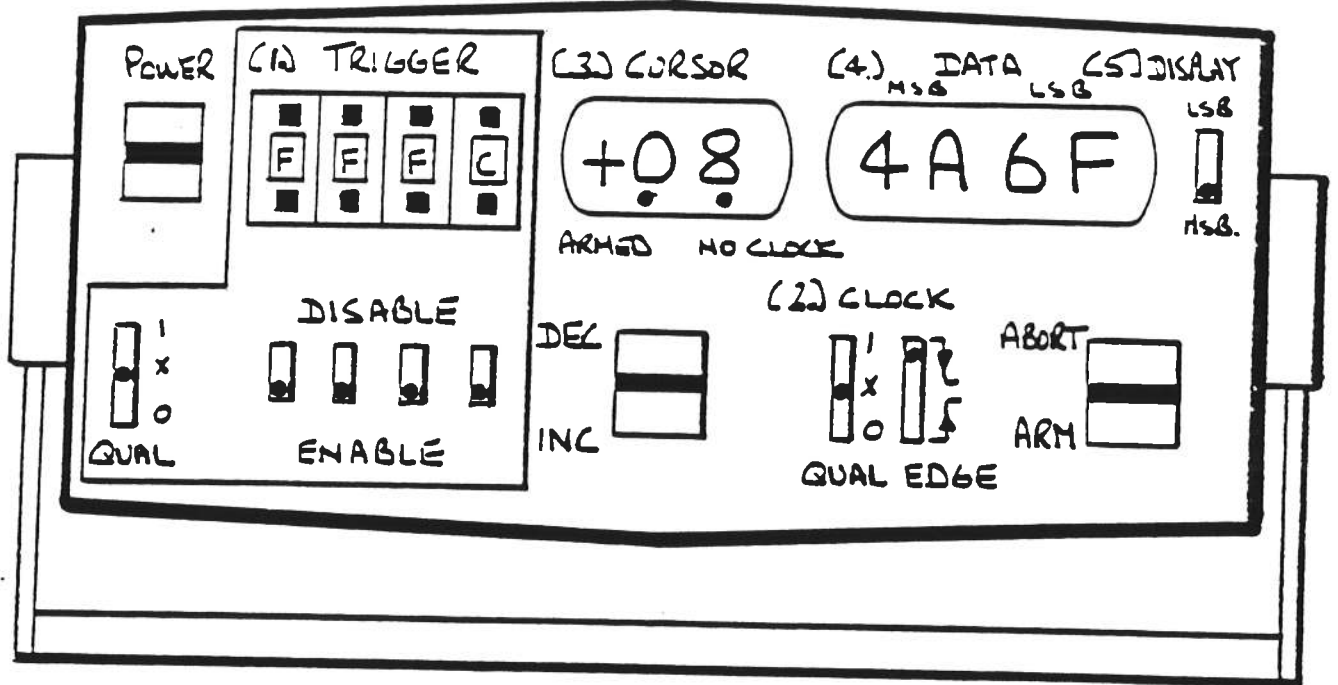
To view the data in timing format an oscilloscope is required. Connect the oscilloscope as in the following section. The timing diagram displays the whole of the memory contents (16 x 128 bits) with the trigger event identified in the centre of the display. The hexadecimal data word displayed on the front panel is indicated on the timing display by a flashing cursor marker.

#### Connection to an Oscilloscope

- (a) Connect the display trigger output to the external trigger input of the oscilloscope. Select the external trigger function.
- (b) Set the internal time base to  $100\mu\text{s}/\text{div}$  and the trigger mode to D.C. coupled.
- (c) Connect the display output (MSB/LSB) to the vertical amplifier input (Y) of the oscilloscope.
- (d) Select 0.5 volts/div, D.C. coupled on the vertical amplifier.
- (f) Use the INC/DEC switch to position the cursor at +63, then using the display output frequency adjust (located on the rear panel), position the flashing cursor at the extreme right hand side of the display.
- (g) If it is required to display all 16 channels of data, connect the display output LSB to the second vertical amplifier input (Y) of the oscilloscope. Set the gain to 1V./div on both amplifiers, select chopped trace and re-position the displays using the Y offset adjusts.



# LOGIC ANALYSER.



- (1) TRIGGER. - SWITCHES TO SELECT TRIGGER WORD IN HEX. INPUTS CAN BE ENABLED / DISABLED IN BANKS OF 4. TRIGGER QUALIFIER CAN BE SET TO +SV, 0V OR DON'T CARE.
- (2) CLOCK. - SELECT CLOCK EDGE RISING OR FALLING. CLOCK QUALIFIER CAN BE +SV, 0V OR DON'T CARE.
- (3) CURSOR - GIVES POSITION OF DISPLAYED DATA RELATIVE TO TRIGGER WORD IN DELTAL.
- (4.) DATA - VOLTAGE PATTERN ON 16 INPUT CHANNELS DISPLAYED AS A 4 DIGIT HEX. NUMBER.
- (5) DISPLAY - ALLOWS YOU TO SELECT LAST SIGNIFICANT OR MOST SIGNIFICANT 8 INPUT CHANNELS FOR DISPLAY ON AN OSCILLOSCOPE.

## Circuit Connections

- (a) GND - BLACK (3 OFF)
- (b) CLOCK - YELLOW.
- (c) CLOCK QUAL - GREEN
- (d) TRIGGER QUAL - BLUE
- (e) INPUTS - RED  
(MARKED 0 to 15)

## CRO Connections

- (a) DISPLAY OUTPUT LSB/MSB to Y INPUT ON CRO.
- (b) DISPLAY TRIGGER. to EXT TRIGGER ON CRO.
- (c) SCOPE 100 NS/CM ; EXT TRIG 0.5V/CM ; DC.





