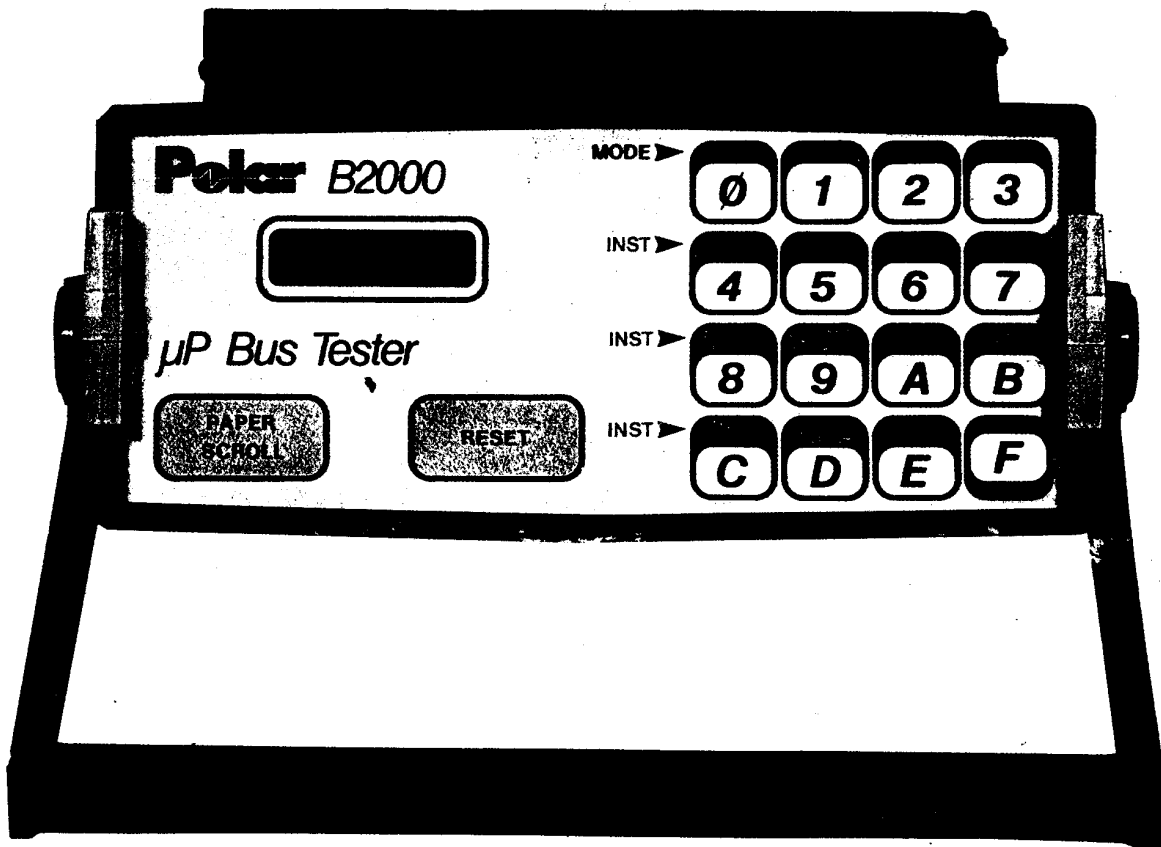

B2000

MICROPROCESSOR SYSTEM TESTER



POLAR ELECTRONICS LIMITED

CONTENTS

Introduction	2
How is a System Tested?	2
Hex Digit Entry	3
Introduction	2
How is a System Tested?	2
Tests (Instructions) Available	4
Operating Modes	7
Programs	9
Constructing a Simple Program	9
Entering the Simple Program	9
Running the Simple Program	11
Editing/Changing a Program or Key Entry	12
Testing Strategy	14
Memory Map Program	16
Incomplete Decoding	16
Soldered in Microprocessors	17
Dynamic RAM	17
Printer Paper	18
Reconfiguration for a different Processor	18
Change of Mains Range	18
Input/Output Details	13
Timing Waveforms	22
Z80 and 8085 Connection Details	23
6800 and 6502 Connection Details	24
ASCII Code Printout	25
Schematic Diagrams	(fold out at rear)

POLAR ELECTRONICS LIMITED
P.O. Box 97, Lowlands Industrial Estate, St. Sampson's, Guernsey, Channel Islands.
Telephone Guernsey (0481) 48129 Telex No. 4191416 Seals G. (Polar)

Polar Electronics Limited pursues a policy of continuous development and product improvement. The specification in this document may therefore be changed without notice.

INTRODUCTION

Fault location on microprocessor based systems cannot be done effectively using conventional pieces of test equipment. Units employing techniques specifically designed for microprocessor troubleshooting need to be used.

The technician or engineer is likely to have easy access to an oscilloscope, D.V.M., logic probe etc. In some facilities there may even be logic analysers or signature analysers available.

These all have their respective uses and in particular the latter items are oriented towards microprocessor circuitry. Unfortunately logic analysers are often complex to use and it can be difficult to interpret their results. They are more suited to the design/development laboratory. Signature analysis is highly useful but suffers from two main drawbacks:-

- 1) It needs to be designed into the product from day one.
- 2) To obtain signatures, the basic system must be operating.

The B2000 has been specifically designed to aid rapid troubleshooting on microprocessor boards. It does not suffer from the disadvantages stated previously and in addition it features:-

- (i) Low cost.
- (ii) Ease of use.
- (iii) Storage of programs dedicated to a particular board

It is primarily of use in manufacturing and production areas, where small to medium (or high) volumes of products are being tested. The other main user will be involved in field service repair, where equipment has been returned by the customer. It has been designed to support the Z80, 8085, 6800 or 6502 microprocessor.

The unit is not a design or development tool although it may occasionally be used to verify a board or to exercise some part of a circuit in the late design stages. It has deliberately been given simple pre-programmed instructions that are aimed at the technician or engineer who has to repair production boards that contain microprocessors and their associated devices (ROM, RAM etc).

HOW IS A SYSTEM TESTED?

The underlying concept is to test the board by setting up the B2000 so that it simulates many of the microprocessor operations. The microprocessor IC itself is not tested. It is usually removed for the duration of the tests (eg by unplugging it*) so that the B2000 can take over control.

The microprocessor has three busses connected to it:-

- (i) DATA BUS.
- (ii) ADDRESS BUS.
- (iii) CONTROL BUS.

There will be a variety of devices connected to these three busses and the microprocessor, during normal operation, reads data from and/or writes data to these devices. This, in turn, can cause some further action to occur (eg a light comes on, a motor runs etc).

With the microprocessor removed from the circuit and the B2000 plugged in its place, the B2000 has control of the devices. The engineer can write simple programs into the B2000 which will cause the various devices to be individually tested so that the specific fault can be isolated. For example the B2000 can be programmed to perform a checksum on a ROM IC, which effectively checks that it has the correct data (software) stored in it.

*For soldered in microprocessors, see page 17

Programming of the B2000 does NOT require any knowledge of computer or microprocessor languages. There are thirteen pre-programmed instructions which are all selected by single key stroke entry.

The programmer does (usually) need to know the memory map of the system (ie how the I.C.'s have been connected within the system). This is almost always available to a production or field service area from the service manual, test procedures, designer data etc. In the event of it not being available, the B2000 contains a memory MAP feature that will, in most cases, provide the information.

In addition, the unit features a non volatile store which allows an engineer to devise and enter a sequence of instructions to test a particular board. These will then be stored (even if power is removed) so that they may be subsequently re-used by a production operator or technician at a later date.

HEX DIGIT ENTRY

Most of the B2000 instructions are concerned with writing or reading data into or from particular memory address locations.

The data bus is 8 bits wide, each bit can be a 1 or 0 and an eight bit pattern is termed a byte.

Data bytes can be expressed in binary (a pattern of 1's and 0's) or in hex notation using digits 0 to F. Two hex digits are needed to specify a 8 bit pattern.

The address bus is 16 bits wide. Hence it consists of two bytes and requires four hex digits to specify it.

Hex notation is used when programming the B2000 – data as 2 digits, addresses as 4 digits.

TESTS (INSTRUCTIONS) AVAILABLE

Before describing how programs (which consist of a sequence of instructions) are entered and run on a faulty pcb, this section is concerned with details of the tests available from the B2000.

Instruction titles are shown on the front panel in the red area above keys 4 to F. When the alphanumeric display is prompting INSTR> then depression of any key between 4 to F will be read as the instruction printed in the red block of that key.

1. Checksum (Key 4)

An efficient way of testing R.O.M. devices is to perform an addition (summation) of all the data bytes stored in it and compare the result with that from a known good ROM.

The B2000 will perform a checksum between a start address and end address which are both entered as two 4 digit locations after depression of the CHECKSUM key.

Shown below is the printout for a CHECKSUM between 0000 and 0FFF with a result of 008E3B. (The digit 1 preceding CHECKSUM shows that it is step 1 of a program).

```
1 CHECKSUM          0000 0FFF
SUM=008E3B
```

The operator would require to know what the sum from a known good ROM was, so that he could determine if the ROM was faulty.

2. Ramtest (Key 5)

The function of RAM is that data can be written into an address location, stored and then read back at a later stage. In this test the B2000 writes a series of ones into a location, ensures that it can read them back, then repeats the process with a series of zeros.

This is done to all addresses between a start and stop address entered as for checksum.

Shown below is the printout for a good RAM between address locations 4000 and 43FF (inclusive).

```
1 RAMTEST          4000 43FF
RAM OK
```

When a fault is found (ie one bit is stuck high or low) the printout shows this. Eg data bit 6 is stuck high at location 419C is shown as:-

```
7 RAMTEST          4000 43FF
419C .*1.....
```

Data bit 3 stuck low at location 416C is shown as:

```
8 RAMTEST          4000 43FF
416C .***0***
```

The printer will give specific details of all bits stuck high or low (• means OK). Hence if the RAMTEST is inadvertently done on a ROM, all bits will be reported as stuck and you will receive a binary dump of the ROM (and a long printout!)

Pressing any key for several seconds whilst a Ramtest is in progress (eg if many bytes are being reported as stuck) will cause the B2000 to finish printing the current line and then go on to the next instruction.

3. Disassemble (Key 6)

Stored within part of the ROM will be the faulty pcb's operating instructions. The DISASSEMBLE function allows the user to have this printed out in assembly language which can allow inspection of the details of the stored program.

The following shows a Z80 program disassembled between locations 00B0 and 00D0.

9	DISASSEMBLE	00B0	00D0
00B0	LD BC,0030		013000
00B3	CALL C390		CD0000
00B6	LD A,FE		3EFE
00B8	OUT (07),A		D307
00BA	IN A,(06)		DB06
00BC	CP FF		FEFF
00BE	JP Z,00D3		CA0300
00C1	CP EF		FEFF
00C3	JP NZ,00CB		020B00
00C6	LD A,10		3E10
00C8	JP 012C		C32C01
00CB	LD B,00		0600
00CD	CALL 012D		CD2D01
00D0	JP 012C		C32C01

The start and end addresses are entered as for checksum.

If any key is pressed for several seconds whilst the B2000 is disassembling, then it will complete the next line and then go on to the next instruction.

4. Delay (Key 7)

This instruction allows the user to insert a time delay between two instructions of a program. The delay time is controlled by entering two digits between 00 and FF. FF gives approximately 30 seconds delay, 80 about 15 seconds. The B2000 sounder will tick to advise that delay is occurring.

The delay feature can be used to give time for an operator to observe the effect of a previous instruction (eg to check that a light comes on).

The printout is as below for a delay of 30 seconds.

1	WAIT	FF
---	------	----

5. Memory Read (Key 8)

This allows inspection of the data stored at any entered memory location. The printout is in hex, binary and where applicable, ASCII. An example below shows the printout at address 00D0 with the hex, binary and decoded ASCII.

5	READ	00D0
50	00110000	P

6. Memory Write (Key 9)

A data byte may be written to any memory location which is entered when the alphanumeric display prompts ADDR. After entry of the address the display then prompts for DATA.

Below is the printout for byte 0E written to memory location 47AB.

B	WRITE	47AB 0E
---	-------	---------

7. Input/Output Read (Key A)

An input port is read for data in the same way as a Memory Read reads data from memory.

The printout also has the same format as for Memory Read.

8. Input/Output Write (Key B)

Data (two digits) is written to an output port in the same way as a Memory Write.

The printout format is also the same as for a Memory Write.

9. ASCII display (Key C) see table on page 25

After this instruction is selected, a start and end address are entered as for Checksum. The B2000 will read the data stored between the two locations and will print it out in decoded ASCII at 16 characters per line. This instruction is useful for locating text areas within a Computer's ROM.

A printout below shows all the characters between locations 0020 and 007F. Where the stored data is not in ASCII code, then the printer leaves a blank (eg at locations 0020 and 007F).

```
2 ASCII          0020 007F
0020          !"#$%&'()*+,-.:/
0030          0 1 2 3 4 5 6 7 8 9 <=>?
0040          @ ABCDEFGHIJKLMNOP
0050          PQRSTUVWXYZ[\]^_`
0060          abcdefghijklmnop
0070          qrstuvwxyz{|}~
```

Pressing any key for several seconds whilst an ASCII dump is in progress will cause the B2000 to finish that line and then go on to the next instruction.

10. Hex display (Key D)

This instruction operates in the same way as for ASCII display only it gives a dump of the data between two locations in hex at 8 two digit bytes per line

The printout below shows a Hex display between addresses 00D3 and 0123.

```
3 HEX          00D3 0123
00D3 3E FD D3 07 DB 06 FE FF
00DB CA F0 00 FE EF C2 E8 00
00E3 3E 20 C3 2C 01 06 04 CD
00EB 2D 01 C3 2C 01 3E FB D3
00F3 07 DB 06 FE FF CA 0D 01
00FB FE EF C2 05 01 3E 40 C3
0103 2C 01 06 08 CD 2D 01 C3
010B 2C 01 3E F7 D3 07 DB 06
0113 FE FF CA 2A 01 FE EF C2
011B 22 01 3E 80 C3 2C 01 06
0123 0C
```

If any key is pressed for several seconds whilst a Hex dump is being performed, the B2000 will finish that line and then move on to the next instruction.

11. Search (Key E)

After selection of the Search instruction, a four digit starting address is entered. The display then prompts for four digits and the B2000 will search from the starting address through the whole of memory for every pair of contiguous memory addresses that store the four digits.

This is extremely useful in locating the positions of calls, jumps etc. to a specific address location, the uses of an absolute address etc.

A printout below shows the addresses of the **first** memory location when searching for CD60 from 0000 (ie 0180 contains CD, 0181 contains 60)

```
5 SEARCH          0000 CD60
0180
018A
0191
0198
0203
1180
118A
1191
119B
```

For the Z80, 8085 and 6502 call and jump addresses are encoded backwards i.e. the low order address byte **precedes** the high order address in the instruction. In the above example the subroutine is located at address 60CD not CD60.

The 6800 encodes them high followed by low order address byte eg Jump to address F4C5 is encoded as JSR F4C5.

The B2000 will search for the address bytes as entered by the user who must determine in which order they will occur in the faulty systems ROM.

In the event of no occurrence being found, the printout states this.

```

4 SEARCH          0000 0100
NONE FOUND
  
```

12. Shorts (Key F)

Selection of this instruction will cause the B2000 to check for shorts between each data line to:-

- (i) Vcc or GND.
- (ii) Other data lines.
- (iii) Each address line.

Printout results are interpreted as below:-

- (a) No faults found.

```

7 SHORTS
NONE TO RAILS
NONE TO DATA
NONE TO ADDRESS
  
```

- (b) Data line 3 to ground, data line 6 to Vcc.

```

8 SHORTS
DATA-RAILS *1** 0***
  
```

- (c) Data lines 4 and 5 shorted together.

```

9 SHORTS
NONE TO RAILS
DATA-DATA **11****
  
```

- (d) Data line 7 shorted to address line A (A₁₀)

```

3 SHORTS
NONE TO RAILS
NONE TO DATA
ADDR A 1*****
  
```

The B2000 stops as soon as the first short is found and prints this out. This must be removed before future tests have any validity.

OPERATING MODES

The B2000 tests a faulty board by running through a series of instructions that have been programmed into its memory via the front panel keys.

Up to 15 programs each containing 12 steps (instructions) can be entered and stored in its non volatile memory (this is a CMOS RAM with a rechargeable NiCad battery back up).

When the display is reading MODE> it is prompting the user to select one of four operating modes.

The blue areas above keys 0, 1, 2 and 3 contain the four mode titles and are used to select it.

1. **Program**

If key 0 is pressed, the programming only mode is chosen. This is solely used to enter a program, where the user has made a note of the desired instruction sequence that he wishes to enter, and no other operation is required.

2. **Direct**

If key 1 is chosen, a program may be entered using the front panel keys as in the program mode. In addition however, the unit will **perform the instructions as they are being entered**. The program will be stored in exactly the same way as for the program mode, the only difference is that the instructions are directly carried out (and the results will appear on the printer)

3. **Run**

Key 2 is used when it is desired to only run a program and no instructions are to be entered. For example, if program 6 had been previously entered, then its sequence of instructions could be run through on the faulty board by selecting RUN and then 6 (when the display prompts PROG NO) and then 1 (when the display prompts LINE NO).

4. **List**

Depression of key 3 followed by a program number and then a line number will cause the printer to print out a list of the selected program, starting with the line number chosen.

PROGRAMS

The B2000 can store up to 15 programs, each one being given a reference digit between 0 and E.

After selecting the MODE, the display will prompt for a PROG NO and it is then that the digit is entered.

If a production area has two types of board to be tested (X and Y) programs 3 and 4 could be assigned to board X and programs 7, 8 and 9 to board Y.

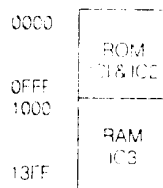
These references will be retained by the B2000 so that each time board X is to be tested, program 3 or 4 should be selected by the user.

CONSTRUCTING A SIMPLE PROGRAM

Devices within a microprocessor based board will be connected to buses and each is enabled when the microprocessor outputs a specific range of addresses.

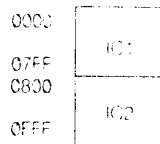
To test the board, the addresses of all IC's must be known, this is termed the Memory Map. In addition, input and output ports, if not memory mapped, will also have a location address that must be known.

Consider a very simple system where the memory comprises of two ROMS and one RAM.



The ROM consists of two ICs and it is better to test these **individually** so that any fault is identified down to a specific device.

The map for the ROMs.



The program will consist of --

- (i) Shorts test – there is little point in doing further tests if any shorts are present on the data bus. Hence this is done first.
- (ii) Checksum for IC1 between addresses 0000 and 07FF.
- (iii) Checksum for IC2 between addresses 0800 and 0FFF.
- (iv) Ramtest for IC3 between addresses 1000 and 13FF.
- (v) End.

ENTERING THE SIMPLE PROGRAM

We will assume that the above program is to test a specific board and that it will be assigned program number 4 (This will be stored by the B2000 so that program 4 can be recalled at a later date for subsequent use).

When power is first applied to the unit, it performs a self check routine and after this is completed, shows briefly on the display which of the processors it is configured for (Z80, 6800, 6502 or 8085). Check that the processor type agrees with your board under test!!

Pressing the front panel RESET mimics the start up routine.

Once the self check is completed and OK, the display reads MODE> and is waiting for one of the top four keys to be pressed to select the mode. If any of the other twelve keys is pressed, the display will read INVALID and reject the entry.

In this case, we wish to enter a program (as opposed to listing or running one already stored) so the choice is either PROGRAM or DIRECT.

Assume that no faulty board is connected to the B2000, hence there is no point in running the tests as they are being entered (DIRECT) so we will choose PROGRAM mode.

Once this is pressed the display will now prompt for a PROG NO. and this must be a digit between 0 and E. (Key F will be rejected).

Since we have decided to call this program number 4, this key is now pressed. Note that if there was a previous program already stored under number 4, then your new instructions will **destroy** (overwrite) those of the old program!

The display is now prompting for a LINE NO which must be between 1 and C (ie up to twelve instructions can be included in one program).

Programs start at line 1, so key 1 is pressed. (Selection of line number allows for editing/changing as described later).

The display is now prompting INSTR> (an instruction).

Referring to page 9, the first instruction is a shorts test, hence key F (SHORTS) is pressed.

The display confirms this key entry and then requests a new instruction which will automatically be assigned to the next line number (ie 2).

This is to be CHECKSUM so key 4 is pressed. The display confirms CHECKSUM and then prompts for the start address (FROM — — —). This is to be 0000 so the 0 key is pressed four times and the display registers these four entries.

It now prompts for an end address TO — — — so 07FF is entered. After these four digits, the instruction is complete and the unit moves on to prompt for the next instruction.

This is another checksum between 0800 and 0FFF. The reason for entering two separate checksum tests is so that if a fault is present with IC1, it will be diagnosed by line 2, whereas line 3 will identify a fault with IC2.

The key entry for line 3 will be --

Display	Key entry
INSTR	CHECKSUM
FROM	0800 (four entries)
TO	0FFF (four entries)

The display now prompts for the next instruction which is a RAMTEST between 1000 and 13FF.

The sequence will be --

Display	Key entry
INSTR	RAMTEST
FROM	1000
TO	13FF

After this instruction, the unit is ready to receive the next one, but our program requires no further steps.

To END a program (ie to insert an END instruction) any one of the four blue keys (0, 1, 2 or 3) is pressed when the prompt INSTR is present in the display.

If one of these keys is pressed, the display confirms END and then reverts to MODE.

Whilst the program was being entered, the printer was operating and giving a printout of each line so that a hard copy exists for checking.

If you wish to have a further copy of this (or any other program) the LIST mode may be used.

eg.

Display	Key entry
MODE	LIST
PROG NO	4
LINE NO	1

The printer will now printout program 4 commencing from line 1 and at the end the display will return to MODE.

RUNNING THE SIMPLE PROGRAM

Although it is most unlikely that you will have a board suitable for the previous program, it is worthwhile describing the exact procedure for testing such a board.

It is usual for the operator to have a checksheet that would state "Program 4 used to test board number PCB123" and then to give certain instructions as below, reference the tests to a particular IC and state the correct checksums for lines 2 and 3 in the program.

The test sequence would be as follows: -

1. Power up the faulty pcb with its microprocessor connected in the circuit. (The B2000 is NOT connected to the board at this stage).
2. Using a DVM, check all the supply rails on the board. (Remember to include rails that may only be present on the RAM's).
3. Using an oscilloscope, check that the clock is present on the relevant microprocessor pins.

The reason for performing steps 2 and 3 **prior** to using the B2000 is that if there is a fault with either the clock or supply rails, the B2000 will not diagnose them and valuable time could be spent in trying to interpret the B2000 results when the fault is a defective crystal, voltage regulator etc.

Assuming that 2 and 3 are OK,

4. Remove power from the faulty board and remove its microprocessor noting the position of pin 1. (See page 17 for a soldered in device).

It is possible to damage microprocessors by static, so use anti-static handling procedures.

5. Plug in the 40 pin plug into the microprocessor socket, ensuring that pin 1 is in the same position as the processor's pin 1. Also check that none of the pins bend under the plug and fail to go into the socket.

Connect the other end to the B2000 rear panel.

6. Apply power to the B2000 and the faulty pcb. Check that the microprocessor number (Z80, 6800, 8085, 6502) that appears on the display, matches that of the removed part!

7. The display will prompt MODE. Press RUN.
8. Select 4 in response to PROG NO.
9. Select 1 in response to LINE NO. ie you are choosing to start at line 1. (If you wished to only test the RAM, you could commence at line 4).
10. Program 4 will run, the two checksums and the RAM result are printed out as below:—

```

RUN 4 FROM 1
1 SHORTS
NONE TO RAILS
NONE TO DATA
NONE TO ADDRESS

2 CHECKSUM          0000 07FF
SUM=008E3B

3 CHECKSUM          0800 0FFF
SUM=009100

4 RAMTEST           1000 13FF
RAM OK

```

11. To see if the ROM's are OK, the operator will need to know the correct checksums. These will have been obtained by running program 4 on a board with known good ROM's and keeping a record of the results (e.g. by using the printout and marking it "reference").

EDITING/CHANGING A PROGRAM OR KEY ENTRY

Doubtless there will be times when you wish to change part of a program or find that you have entered the wrong data etc.

1. If you press a key and as you are pressing it, realise that it is the wrong key – keep it depressed until NULL or X appears in the display. This will show that the unit registers that it is an incorrect key entry and has rejected it.

If you have removed your finger from the key then this will not work. The display and printer will show you what has been or is about to be entered into the B2000 memory.

2. Let us assume that you wish to modify an address in line 4 of program 5 – but leave the rest of the program unaltered. eg change line 4 from:—

RAMTEST from 0000 to 07FF

to

RAMTEST from 0000 to 0FFF

The key sequence is as follows:—

Display—	Key Entry
(i) MODE—	PROGRAM
(ii) PROG NO—	5
(iii) LINE NO—	4
(iv) INSTR—	RAMTEST
(v) FROM—	0000 (4 entries)
(vi) TO—	0FFF (4 entries)

This has now overwritten the old instruction and the display is prompting for a new instruction (which could be written into line 5).

Since no other changes are required and you do not want to end after line 4, do NOT press one of the four blue keys which would insert an END instruction in line 5.

Instead you press the RESET key which causes the B2000 to exit from the program without altering any subsequent lines and returns it to the MODE prompt.

3. If an incorrect key is pressed during program entry eg as the address of a memory write is being entered, 8421 instead of A421 is entered, the same procedure as in 2 above has to be performed – ie the whole line must be re-written. The procedure is:–

- (a) Exit from the program using RESET.
- (b) Re-enter using PROGRAM (or DIRECT).
- (c) Select the relevant PROG NO.
- (d) Select the relevant LINE NO which has the error (consult the printout if unsure)
- (e) Re-enter the line with the correct details.
- (f) Proceed with the rest of the program.

TESTING STRATEGY

You will require a memory map of the board to be tested before a program can be entered into the B2000. The map contains the address locations of RAM, ROM and, if present, input/output devices.

The map is usually available from the board's handbook and/or inspection of the schematic diagrams.

Where this is not available, the B2000 has a MAP program which is described on page 16.

The basic testing method is to use one or more instructions to test each I.C. If one instruction tests two devices, then a fault on that test will not tell the operator which IC is faulty.

Each ROM is tested by performing a checksum between its start and end addresses. The operator will require a running sheet which gives the correct checksum and this can be obtained by running the test with a known good ROM.

RAM's are again individually tested between their start and end addresses using the ramtest instruction. The printer will either report the RAM OK or give a printout of locations that have bits stuck high or low. There are certain RAMs that can suffer from a fault known as "pattern sensitivity" and the B2000 is unlikely to detect this subtle condition.

Note that if in error a ramtest is performed on a ROM, the printer will print out every location as being stuck!

The first instruction in most programs will be the shorts test – this will check for databus shorts and give a printout of any found. These need to be cleared before any tests on RAM or ROM since a short will invalidate the results.

Input/output devices (these may be memory mapped or I/O mapped) can be tested by addressing the specific location and writing or reading data. They may require a number of bytes to configure them.

Often the delay instruction is useful when testing I/O since it allows the operator time to either:–

- (i) Observe an output effect (eg a light coming on)
- or
- (ii) Alter the position of input data (eg coded switches).

Where I/O is connected to devices external to the pcb, it may be convenient in a production environment to build a test fixture that can manually test the functions (switches for input, LED's for output). In some cases they can be interconnected so that the data is written out and then read back.

The usual case will be that the bulk of simple faults will be located using programs based on:–

- Shorts
- Checksum
- Ramtest
- Delay
- Memory Read – Memory Write
- I/O Read – I/O Write

If the fault cannot be located using the above, then more detailed information about the system can be obtained using:–

- Disassemble
- Ascii display
- Hex display
- Search

As previously discussed, the operator will require some form of test sheet when using the program (to state which program is required for a particular board, to give the correct checksums etc). A typical example is shown below.

B2000 – Set for Z80
Program 4 to test PCB 127

Test No.	Instruction	Printer	Notes
1	Shorts	-	
2	Checksum 0000 to 0FFF	002124	IC4
3	Ramtest 1000 to 1FFF	OK	IC5
4	I/O Write 8888 - FF	--	LED off
5	I/O Write 8888 - 00	--	LED comes on
6	Delay 80	--	Set switch to F
7	Mem. Read 2000	FF	IC2
8	Delay 80	--	Set switch to 0
9	Mem. Read 2000	0F	IC2

MEMORY MAP PROGRAM

To be able to test RAM, ROM, etc. the memory map of the system must be consulted. If you are confronted with equipment for which there is little or no documentation, then the B2000 contains a MAP program which will, in many instances, give a basic map of the system.

The program is run by the following key sequence:-

Display	Key
MODE	RUI.
PROG NO	MAP (Key F)
FROM	00 (2 entries)

The B2000 will then attempt to determine what is in the addressing space between 0000 and FFFF by writing to and reading from all locations if necessary.

For this test, memory is divided into pages of 256 bytes and hence only the high order address bytes are entered to define a starting address (this is usually 00).

The unit will read the contents of each location, complement it, attempt to write it back to that location, and then re-read the data.

Four different conditions are recognised:-

- (i) The data does not change which is taken to mean that the device is ROM.
- (ii) The complemented data is read back which implies that the device is RAM.
- (iii) The least significant byte of the **address** is read back showing that the data lines are high impedance and nothing is connected, this is EMPTY.

The standard software writes the least significant byte of the address on the data bus and applies a positive address latch enable for use with multiplexed bus processors. If no device is selected, there is nothing to discharge the data bus line capacitance and hence the low order address will be read back as data. This does not work if any non-MOS devices are connected to the bus (eg bus devices, pull up resistors etc).

(iv) The B2000 does not report ROM, RAM or EMPTY unless 256 contiguous bytes have behaved exactly as ROM, RAM or EMPTY. If when performing a test on a page, some behaves like ROM and some behaves like RAM, then it prints I/O? This is a default condition where something is connected to the lines but 256 contiguous bytes do not behave similarly. For example if one location of RAM was faulty with a stuck bit, the whole 256 bytes would be reported as I/O?

As soon as the I/O? condition is reached in a page, then this is reported and the B2000 immediately begins to test the next page, hence I/O is reported quickly whereas ROM, RAM and EMPTY testing is more lengthy because it requires checking every address location.

It is important that the way in which the test works is understood, since the results do require interpretation by the user. Note that since non-MOS devices will be connected to the data bus for an 8085 system, it cannot report any EMPTY space.

Pressing the RESET key whilst a Memory Map is in progress will cause the B2000 to exit from the program and return to prompting MODE in the display.

INCOMPLETE DECODING

Since the address bus is 16 bits wide it is theoretically capable of addressing up to 65,536 (64k) locations. Many systems do not use this amount of memory and this gives the designer flexibility in -

- (i) Where devices are put in the memory map and
- (ii) How the address decoding to select an IC is performed

Number two is important from the viewpoint of testing since the implication is usually that incomplete decoding is used which means that the **same** IC can appear at **different** address locations in memory.

When performing a MAP, the B2000 cannot know this and will purely print out what it finds.

If the user has a specific map relating to the board (eg from the manual or designer etc) then this is unlikely to cause a problem since the addressing will usually be tested within the decoded areas.

Caution however must be used when interpreting the MAP program results, since it can show more memory than actually exists.

SOLDERED IN MICROPROCESSORS

Boards with the 6502 soldered in cannot be tested without unsoldering the microprocessor. (The 3 busses are not all tri-state).

The other three processors supported have tri state busses and may be able to be tested by putting the busses into the high impedance state.

This is achieved by connecting the processor so that it is in a reset, hold etc. condition **BUT** care must be taken to consult schematics and analyse what other effects this may have on external circuitry.

For example if the relevant pin is grounded (directly) and it is strapped by a lead to Vcc to disable the busses, the Vcc line will be shorted out! Hence there are no simple rules to this other than inspection of the circuits is necessary!

Pins that affect the tri state condition and that should be investigated are as below:-

Z80 – pin 25 (BUSRQ) active low.

8085 – pin 39 (HOLD) active high.

6800 – pin 2 (HALT) active low.

Assuming that the busses are put into their high impedance state, then testing can be done in the normal way using a special lead.*

DYNAMIC RAM

If the dynamic RAM contains its own refresh, is not dependent on the system clock and does not put the processor into a WAIT state, then RAMTEST and the other memory tests will operate as on static ram.

If the processor is involved in the refresh then there may be timing problems. This is because the data out is only valid during Column Address strobe whereas the B2000 could attempt to read it at any time.

There is no point in performing Memory Write or Memory Read to dynamic ram if it is directly using the microprocessors' own refresh.

As with the soldered in processors, some schematic interpretation and trial and error will show if the dynamic ram can be tested.

*This can be ordered from your distributor by quoting "B2000 40 pin lead for soldered processor"

PRINTER PAPER

On the front panel there is a SCROLL button which causes the paper to scroll upwards for the duration of pressure on the button.

To change the paper roll:-

- (i) Loosen the four black screws in the lid retaining the metal cover, and slide it off.
- (ii) Put the new roll in the holder and feed the end into the back of the printer. Ensure that the thermal (sensitive) side of the paper is facing downwards.
- (iii) Press the SCROLL button as you feed the paper in until the printer eventually pulls the paper through.
- (iv) Position the black cover so that the paper exits smoothly from the slot and retighten the screws. Do **not** overtighten.

Spare paper rolls can be ordered from your distributor.

RECONFIGURATION FOR A DIFFERENT PROCESSOR

The B2000 has essentially been designed to support four processors and the supported device is selected by an internal personality card plugged into the main pcb.

The unit is not designed for regular alteration of the board but in the event of this being required, the method is simple.

Remove the case lid as described in the section on changing the mains range. **(Remove power first!)**

Remove the fitted card and plug in the new one obtained from your distributor. The personality card has some resistors (one or more) fitted on it used to select the correct software. The side of the board with the resistors faces into the unit, the side with no resistors is adjacent to the rear panel.

Reassemble and check that at switch on, the display shows the correct new processor type.

CHANGE OF MAINS RANGE

There are two primary mains ranges:-

- (i) 200 – 250V
- (ii) 100 – 127V

These are selected by altering the connections to the primary of the mains transformer

Note! This should only be performed by a qualified technician or engineer.

1. Remove the power source from the instrument (ie unplug it).
2. Unscrew the four screws through the feet and remove them.
3. Remove the top of the enclosure. There is a lead connecting the main pcb to the printer assembly in the case top. Take care when lifting off the lid and do not damage the delicate printer flat cable. The lead to the main pcb can be unplugged from the main board.
4. Undo the screws holding the main pcb to the enclosure so that the rear panel can be lifted out of its slot.
5. The transformer primary has two 120V Windings labelled 0 – 120V, 0 – 120V. For 200 – 250V primary these should be connected in series.
For 100 – 127V primary these should be connected in parallel (0 to 0, 120 to 120). Alter the connections as desired. For information the live-neutral resistance should be approximately:-
200 – 250V — 100Ω
100 – 127V — 50Ω
6. Reassemble the unit taking care to plug the lead to the printer the correct way. (Leads exit towards the rear panel).
7. Change the rear panel fuse in accordance with rating on panel.

INPUT/OUTPUT DETAILS

The B2000 has five octal latches ('373's) through which it connects to the 40 pin microprocessor socket. This is shown on the main schematic and on the individual connection details for each processor.

U1 and U2 are for address lines (A₁ - A₁₅), U3 and U4 deal with data in and data out respectively and U5 outputs the control signals.

The internal software routines are universal ie the personality card does not alter the internal software for this part of the program.

The function of the personality card is:-

- (i) Configure the five octal latches to the correct microprocessor socket pin.
- (ii) Select the required outputs from U5 so that correct control bus signals are produced.
- (iii) Provide coding for the B2000 to select the correct disassemble program.

The 8085 has a multiplexed bus structure and to cope with this, the low order address byte appears on the data bus prior to a read or write output. Since the software is universal, this occurs for **all** processor types. Only the 8085 card outputs an Address Latch Enable (ALE) so that the system under test can latch this address, leaving the data bus free to then input or output data.

The MEMORY MAP PROGRAM makes use of the data bus having the low order address byte as described previously.

Page 22 shows details of the relative timing for all outputs of U5 and which pins they are connected to by the four personality cards.

There may be instances where a detailed appreciation of both these waveforms and the relevant program are required by the user, eg if the B2000 fails to test part of a system, there may be some incompatibility which can be traced and rectified.

The other key reason for including these details is that the user may wish to modify the B2000 to test devices not supported by the standard personality cards.

There are two basic requirements:-

- (i) Alter the personality card configuration so that address, data and control bus signals appear on the correct pins.
- (ii) Possibly alter the B2000 software to adjust the control signals

A commented software listing of the input and output routines is provided for this purpose. Neither Polar Electronics, nor any of its agents or distributors are able to offer further information or guidance to the customer if he should undertake to alter the software. It should also be noted that warranty is invalidated if alterations are performed to the unit.

Changes can be performed successfully although one point to remember is that the disassemble function is microprocessor specific and will therefore give incorrect results

B2000 INPUT/OUTPUT ROUTINES

```
A000          CONTROL EQU 0A000H
4000          DATAIN EQU 4000H
6000          DATAOUT EQU 6000H
DFFF          ADDR EQU 0DFFFH
2003          DPORT EQU 2003H
              ; UNIVERSAL INPUT ROUTINE.
              ; LEFT: C,D,E.
0C1F          INPUTCONT:CALL INOUT2:ALL FOR H
```

0C22	E5	PUSH HL
0C23	7E D9	LD A,0D9H;MREQ ONLY ON
0C25	32 A000	LD (CONTROL),A
0C28	3E B8	LD A,0B8H;RD+MREQ ON
0C2A	2E D9	LD L,0D9H
0C2C	32 A000	INI0IN:LD (CONTROL),A
0C2F	3A 4000	LD A,(DATAIN);READS THE BYTE ; AT THIS POINT L CONTAINS A VALUE ; WHICH WILL BE PUT ONTO THE ; CONTROL LATCH.
0C32	26 5D	ENDINOUTS:LD H,5DH;EVERYTHING OFF
0C34	22 A000	LD (CONTROL),HL;L FIRST THEN H
0C37	E1	POP HL
0C38	C9	RET ; PUTS OUT THE ADDRESS TO THE ; CORRECT LATCHES.
0C39	E5	INOUT1:PUSH HL; FOR IO
0C3A	3A 0000+	LD A,(CPU)
0C3D	3D	DEC A
0C3E	F6 02	OR 2
0C40	32 2003	LD (DFURT),A
0C43	22 DFFF	LD (ADDR),HL;ADDRESS ONTO LATCHES
0C46	7D	LD A,L
0C47	32 6000	LD (DATAOUT),A;FOR MULTIPLEXED CPUS ; N.B. THE ABOVE 2 LINES MUST BE ; INCLUDED WHETHER THE SYSTEM YOU ; ARE WRITING IN/OUT ROUTINES FOR ; IS MULTIPLEXED OR NOT. THIS IS ; BECAUSE THE MAP NEEDS THEM TO ; NOTICE EMPTY SPACE.
0C4A	3E 5F	LD A,5FH;ALE ON
0C4C	32 A000	LD (CONTROL),A
0C4F	2E 5D	LD L,5DH;ALE OFF
0C51	18 DF	JR ENDINOUTS
0C53	E5	INOUT2:PUSH HL; FOR M
0C54	3A 0000+	LD A,(CPU)
0C57	3D	DEC A
0C58	F6 02	OR 2
0C5A	32 2003	LD (DFORT),A
0C5D	22 DFFF	LD (ADDR),HL;ADDRESS ONTO LATCHES
0C60	7D	LD A,L
0C61	32 6000	LD (DATAOUT),A;FOR MULTIPLEXED CPUS ; N.B. THE ABOVE 2 LINES MUST BE ; INCLUDED WHETHER THE SYSTEM YOU ; ARE WRITING IN/OUT ROUTINES FOR ; IS MULTIPLEXED OR NOT. THIS IS ; BECAUSE THE MAP NEEDS THEM TO ; NOTICE EMPTY SPACE.
0C64	3E 5B	LD A,5BH;ALE ON
0C66	32 A000	LD (CONTROL),A
0C69	2E 59	LD L,59H;ALE OFF ; WRITES A TO (HL) IN EXTERNAL RAM. ; KEPT:HL,DE,B,C AND A.
0C6B	18 C5	JR ENDINOUTS
0C6D	F5	OUTPUT:PUSH AF
0C6E	CD 0C53	CALL INOUT2; ALE FOR M
0C71	E5	PUSH HL
0C72	C5	PUSH BC
0C73	21 DDD9	LD HL,0DDD9H
0C76	06 08	LD B,B

0C78	22	A000	WAKEUP:LD(CONTROL),HL
0C7B	10	FB	DJNZ WAKEUP
0C7D	C1		POP BC
0C7E	E1		POP HL
0C7F	F1		POP AF
0C80	32	6000	LD(DATAOUT),A:BYTE TO BE OUTPUT
0C83	E5		PUSH HL
0C84	F5		PUSH AF
0C85	3E	59	LD A,59H;W+MREQ ON
0C87	32	A000	LD(CONTROL),A
0C8A	3E	68	LD A,68H;W+WR+MREQ ON
0C8C	2E	59	LD L,59H
0C8E	32	A000	OUTIOOUT:LD(CONTROL),A
0C91	F1		POP AF
0C92	18	9E	JR ENDINGOUTS
0C94	EF		IOREAD:RST FROMLOC
0C95	CD	0C39	CALL INOUT1; ALE FOR IO
0C98	3E	F5	LD A,0F5H;R+IOREQ ON
0C9A	32	A000	LD(CONTROL),A
0C9D	3E	05	LD A,0B5H;R+RD+IOREQ ON
0C9F	01	0000*	LD BC,BYTEBITS
0CA2	C5		PUSH BC;SO IT JUMPS TO ICI.
0CA3	E5		PUSH HL
0CA4	2E	D5	LD L,0D5H
0CA6	18	84	JR INIOIN
0CAB	EF		IOWRITE:RST FROMLOC
0CA9	CD	0C39	CALL INOUT1; ALE FOR IO
0CAC	FD	7E 03	LD A,(1Y+3)
0CAF	32	6000	LD(DATAOUT),A
0CB2	01	00C3	LD BC,HELP
0CB5	C5		PUSH BC
0CB6	E5		PUSH HL
0CB7	F5		PUSH AF
0CB8	3E	75	LD A,75H;W+IOREQ ON
0CBA	32	A000	LD(CONTROL),A
0CBD	3E	65	LD A,65H;W+IOREQ+WR ON
0CBF	2E	55	LD L,55H
0CC1	18	0B	JR OUTIOOUT
0CC3	CF		HELP:RST PRINTER
0CC4	C9		RET

END

Symbols:

DFFF	ADDR	00B1*	BYTEBITS	A000	CONTROL
0036*	CPU	4000	DATAIN	6000	DATAOUT
2003	DPORIT	0032	ENDINGOUTS	0028	FROMLOC
00C3	HELP	002C	INIOIN	0C39	INOUT1
0C53	INOUT2	0C1F I	INOUTCONT	0C94 I	IOREAD
0CAB I	IOWRITE	0C8E	OUTIOOUT	0C6D I	OUTPUT
000B	PRINTER	0C78	WAKEUP		

TIMING WAVEFORMS PRESENT ON U5

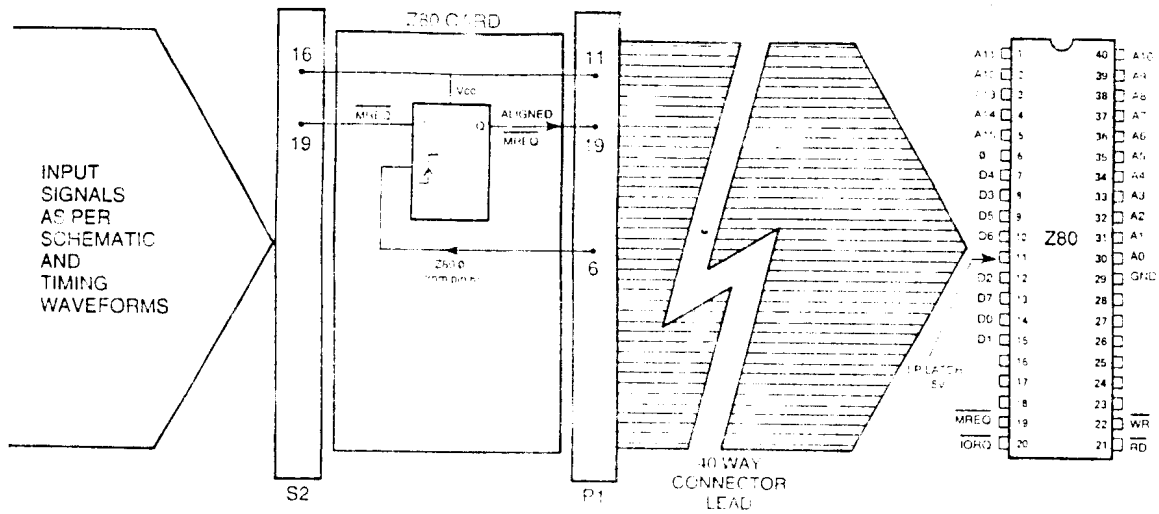
Outputs to processor socket are selected by personality card and are as below except that Z80 card aligns MREQ with system under test clock.

U5 PIN No	6502	6800	8085	Z80	INPUT (READ) ROUTINE	OUTPUT (WRITE) ROUTINE	I/O READ ROUTINE	I/O WRITE ROUTINE
9	ϕ_1	/	/	/			Not Applicable	Not Applicable
12	/	/	ALE	/				
15	/	/	$\overline{IO/\overline{M}}$	\overline{MREQ}				
2	/	/	/	\overline{IORQ}				
22	/	/	\overline{WR}	\overline{WR}				
19	ϕ_2	VMA	/	/			Not Applicable	Not Applicable
5	/	/	\overline{RD}	\overline{RD}				
16	\overline{RW}	\overline{RW}	/	/				

Where a / is shown, the waveform is unused for that processor.

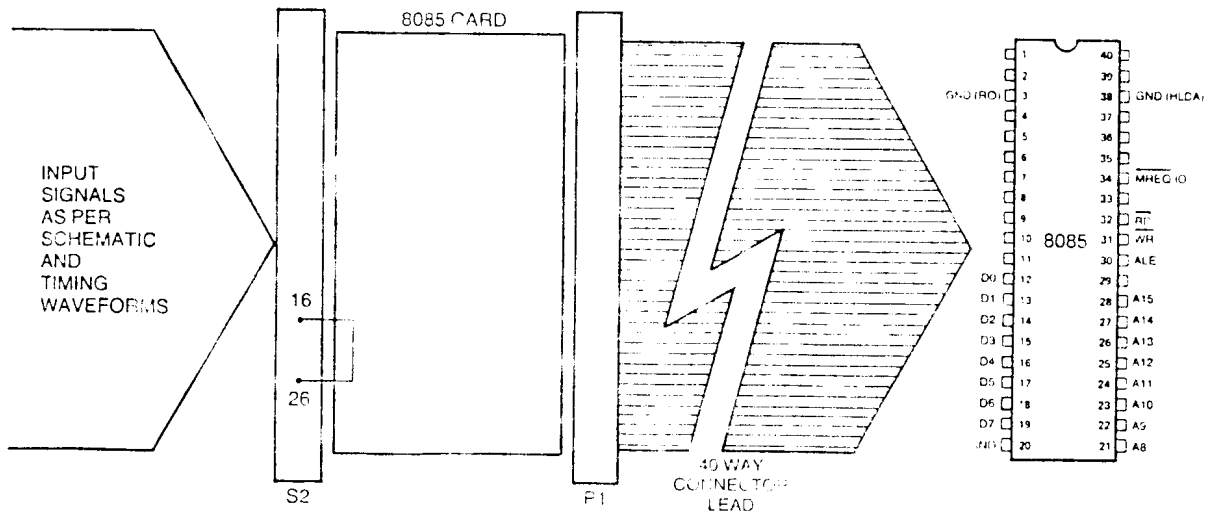
* 8 cycles preceding \overline{WR}

Z80 CONNECTION DETAILS



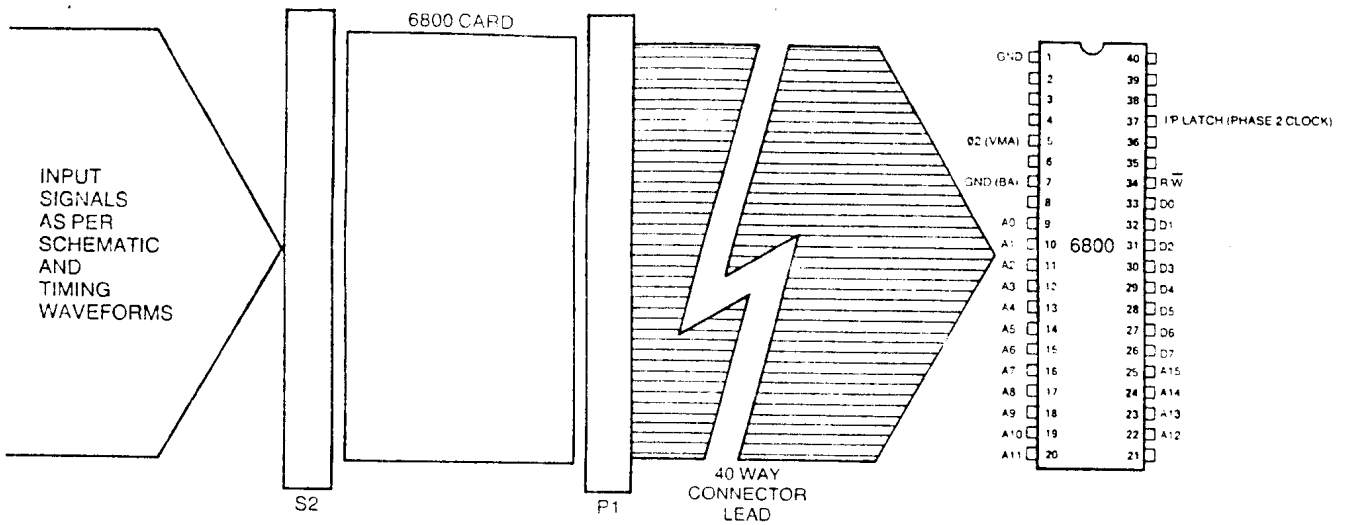
1. Flipflop (74LS74) on personality card aligns $\overline{\text{MREQ}}$ to system clock
2. +5V from Z80 socket powers 74LS74 and enables input latch U3.
3. No connections are made to Z80 pins 16, 17, 18, 23, 24, 25, 26, 27 and 28.
4. S2 pins 25, 26 and 28 LOW to select Z80 software.

8085 CONNECTION DETAILS



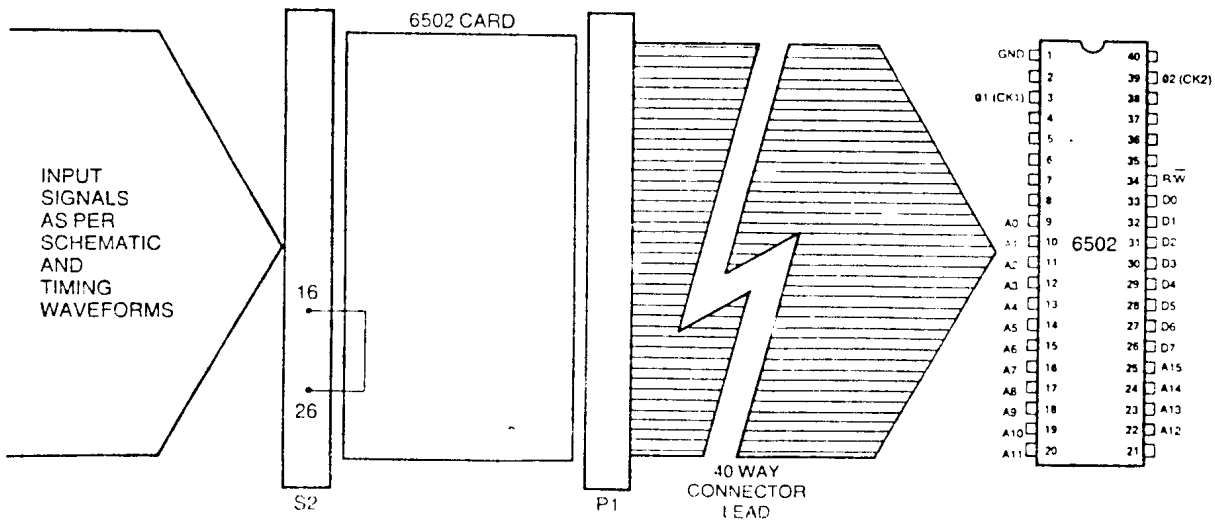
1. S2 pins 16 and 26 linked to enable input latch U3.
2. No connections are made to 8085 pins 1, 2, 4-11 (incl), 29, 33, 35, 36, 37, 39 and 40.
3. S2 pin 25 LOW, pins 26 and 28 HIGH to select 8085 software.

6800 CONNECTION DETAILS



1. No connections made to 6800 pins 2, 3, 4, 6, 8, 21, 35, 36, 38, 39 and 40.
2. S2 pins 25 and 26 LOW, pin 28 HIGH to select 6800 software.

6502 CONNECTION DETAILS



1. S2 pins 16 and 26 linked to enable input latch U3.
2. No connections made to 6502 pins 2, 4, 5, 6, 7, 8, 35, 36, 37, 38 and 40.
3. S2 pins 25 and 28 LOW, pin 26 HIGH to select 6502 software.

ASCII CODE PRINTOUT

HEX	ASCII	HEX	ASCII	HEX	ASCII
21	!	41	A	61	a
22	"	42	B	62	b
23	#	43	C	63	c
24	\$	44	D	64	d
25	%	45	E	65	e
26	&	46	F	66	f
27	'	47	G	67	g
28	(48	H	68	h
29)	49	I	69	i
2A	*	4A	J	6A	j
2B	+	4B	K	6B	k
2C	,	4C	L	6C	l
2D	-	4D	M	6D	m
2E	.	4E	N	6E	n
2F	/	4F	O	6F	o
30	0	50	P	70	p
31	1	51	Q	71	q
32	2	52	R	72	r
33	3	53	S	73	s
34	4	54	T	74	t
35	5	55	U	75	u
36	6	56	V	76	v
37	7	57	W	77	w
38	8	58	X	78	x
39	9	59	Y	79	y
3A	:	5A	Z	7A	z
3B	;	5B	[7B	{
3C	<	5C	¥	7C	.
3D	=	5D]	7D	}
3E	>	5E	^	7E	~
3F	?	5F	=		
40	@	60	`		

LIMITED WARRANTY

For a period of one year from the date of its purchase new and undamaged from Polar Electronics Ltd., POLAR ELECTRONICS LTD. or its authorised distributors will, without charge, repair or replace at its option, this product if found by it to be defective in materials or workmanship, and if returned to POLAR ELECTRONICS LTD. or its authorised distributors transportation prepaid. This limited warranty is expressly conditioned upon the product having been used only in normal usage and service in accordance with instructions of POLAR ELECTRONICS LTD. and not having been altered in any way or subject to misuse, negligence or damage, and not having been repaired or attempted to be repaired by any one other than POLAR ELECTRONICS LTD. or its authorised distributors. EXCEPT FOR THE FOREGOING EXPRESS WARRANTY OF REPAIR OR REPLACEMENT POLAR ELECTRONICS LTD. MAKES NO WARRANTY OF ANY KIND, INCLUDING BUT NOT LIMITED TO, ANY EXPRESS OR IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE AND POLAR ELECTRONICS LTD. SHALL NOT BE LIABLE FOR ANY DAMAGES, WHETHER DIRECT OR NOT OR OTHERWISE, BEYOND REPAIR OR REPLACING THIS PRODUCT

B2000

MICROPROCESSOR

The B2000 is a low cost production or field service troubleshooting instrument that will diagnose and locate faults in microprocessor systems.

It does this by simulating the microprocessor action i.e. the μ P is removed and the B2000 plugged in which then takes command of the address, data and control busses.

Using the functions pre-programmed into the unit, it can then generate the required data, address and control signals to test and activate ROM, RAM, I/O devices etc. Dynamic RAM can often be tested depending on configuration.

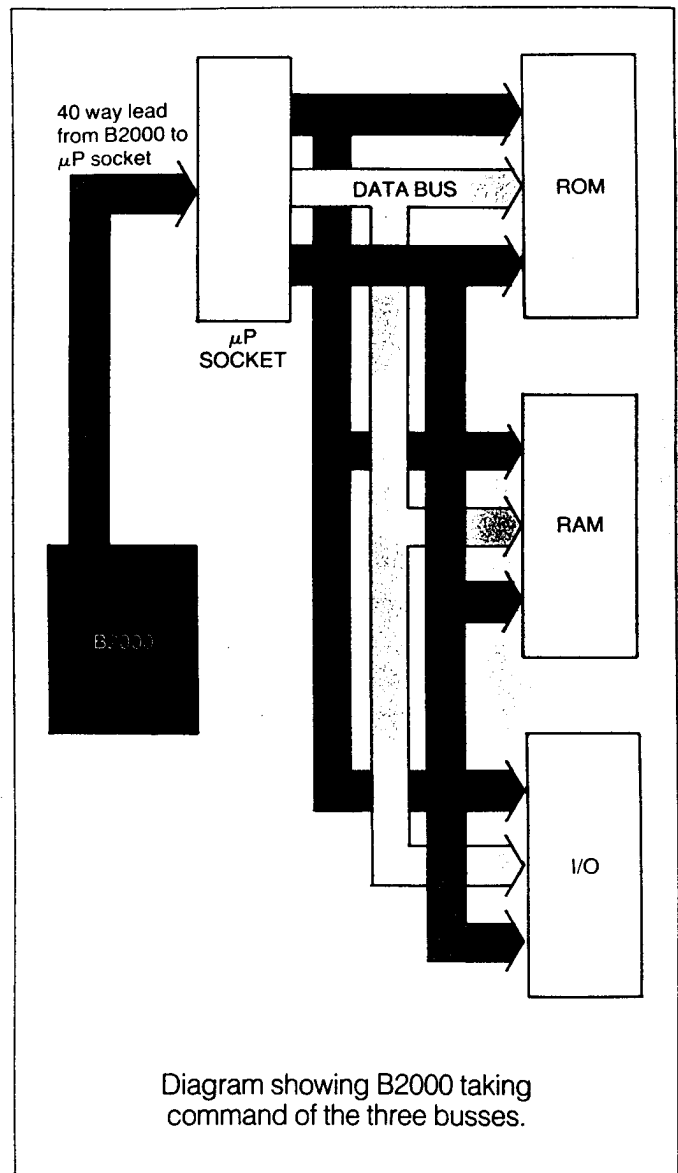
Test sequences are stored in a non volatile memory for subsequent use. Results are reported on an integral printer.

FEATURES

1. Internal personality cards will support Z80, 8085, 6800 or 6502.
2. Alphanumeric display to ease programming and general use.
3. Results shown on an integral printer.
4. Will test RAM, ROM and I/O.
5. Non volatile memory will store up to 15 test sequences each containing up to 12 steps.

TESTS

1. ROM checksum.
2. Tests ability to read and write to RAM.
3. Will printout disassembled ROM program.
4. Will read a programmed memory location.
5. Will write to a programmed memory location.
6. Will read a programmed I/O location.
7. Will write to a programmed I/O location.
8. Will decode a block of ROM and print out result in HEX and/or ASCII.
9. Checks for data bus shorts to V_{CC}, GND, address lines or other data lines.
10. Will printout a memory map (RAM, ROM, I/O Empty) of system.



Polar

SYSTEM TESTER

ROM Checksum
between
0000 and 0FFF

RAM tested
between
4000 and 43FF

Data AB written
to location 4321
in memory

Data bus
shorts test

System ROM
disassembled between
0090 and 0095

Memory location
4321 read (in HEX
and binary)

Hex dump of data
between locations
0105 and 0122

