User manual

IMS B004 evaluation board

Preface

The IMS B004 IBM Personal Computer add in board enables users to evaluate and demonstrate the use of transputers

The board is a member of a family of compatible evaluation boards. It provides standard buffered INMOS link connections and external control of the transputer's Reset and Analyse functions. This allows it to control a subsystem consisting of other compatible boards, or to be a component of such a subsystem.

This manual details the product specific aspects of the IMS B004, and contains all the data necessary to install. power up, test and program the board.

Other information relevant to all transputer products is contained in the occam programming manual (supplied with INMOS software products and available as a separate publication), and the transputer reference manual (supplied with this board). This board is designed to be used in conjunction with a Transputer Development System (TDS), using either the host Personal Computer, or running on another machine and connected to the board via an INMOS link. Reference should be made to the transputer development system user manual (supplied with the development system), for details of how to compile and load programs for a network of boards.

This manual assumes familiarity with MSDOS.

Contents

Contents

Getting started with IMS B004

1.1   Introduction   The IMS B004 board is logically divided into three distinct parts:

1   The transputer, with buffered links and one or two megabytes of RAM

2   The PC subsystem logic, which allows a program running on the Personal Computer to reset and analyse systems

3   The IMS C002 link adaptor, which interfaces to a parallel address/data bus, such as the one provided on the system expansion slots within an IBM Personal Computer. The link adaptor is accessed by a program running on the Personal Computer to transfer data to and from the transputer.

These three distinct parts of the board are joined together, if required, by the two jumpers provided. The 'Reset' jumper allows the PC subsystem to respond to addresses from the Personal Computer, and connects the transputer's reset, analyse, and error signals to those controlled by the Personal Computer. The 'Link' jumper connects the link adaptor to one of the transputer's links, and allows the Link Adaptor to respond to addresses from the Personal Computer.
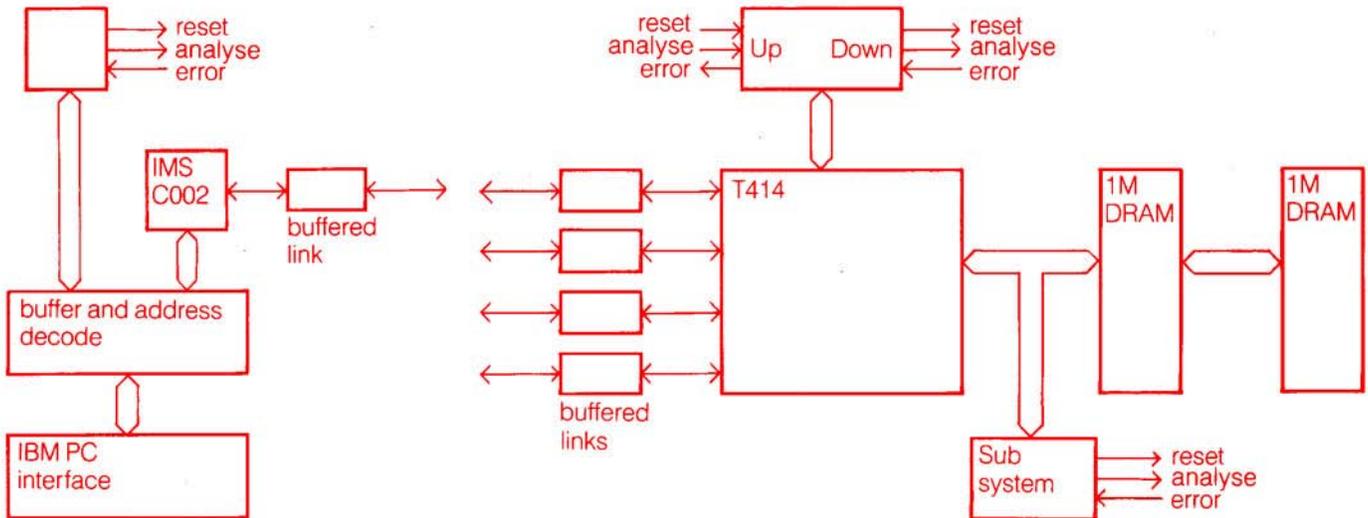
Getting started with IMS B004

1.2 Opening the Box

When you open the box, you will find

1 Some documentation, including this manual

2 A floppy disk, which contains some test software

3 A number of cables, consisting of two link cables, a reset cable, and two jumper plugs in a polythene bag

4 An anti static bag, containing the add-in board.
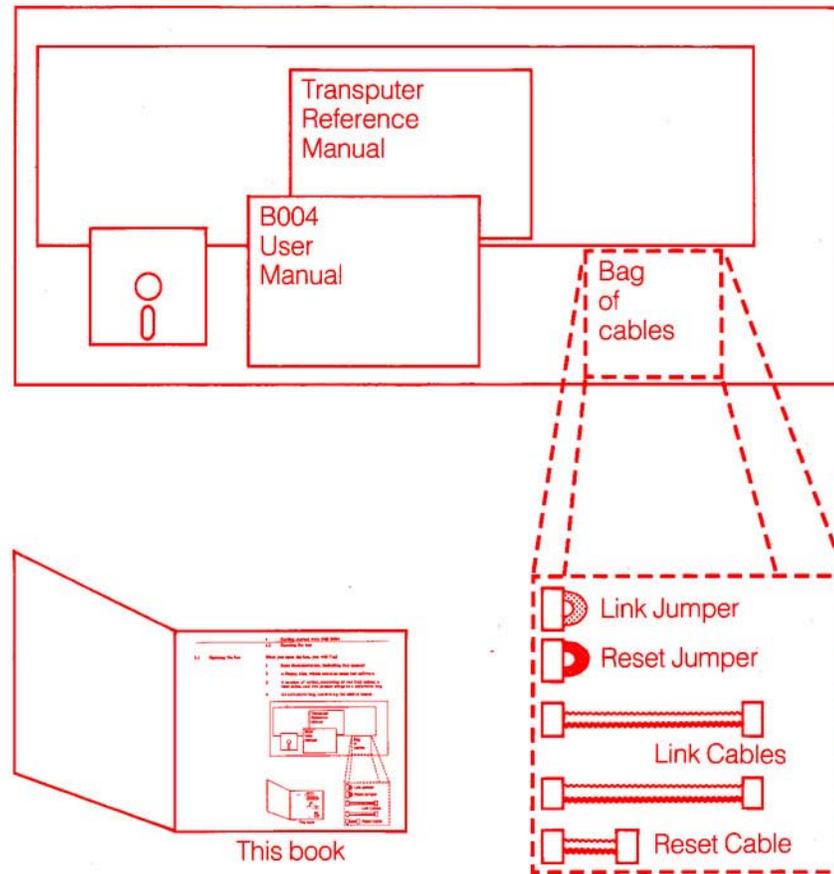
1.3	**Removing the cover of your unit**	Before removing the board from the bag, the cover must be removed from the host IBM Personal Computer. Refer to the section 'Cover Removal' for specific details of your unit. This section can be found in the 'Installation and Setup' manual, which is supplied with the IBM Personal Computer.

1   Ensure your system unit's power, and any externally connected devices, are switched OFF.

2   Disconnect all cables from the back of your machine, noting their locations.

3   Remove the five cover mounting screws from the rear panel.

Cover screws

Cover screws
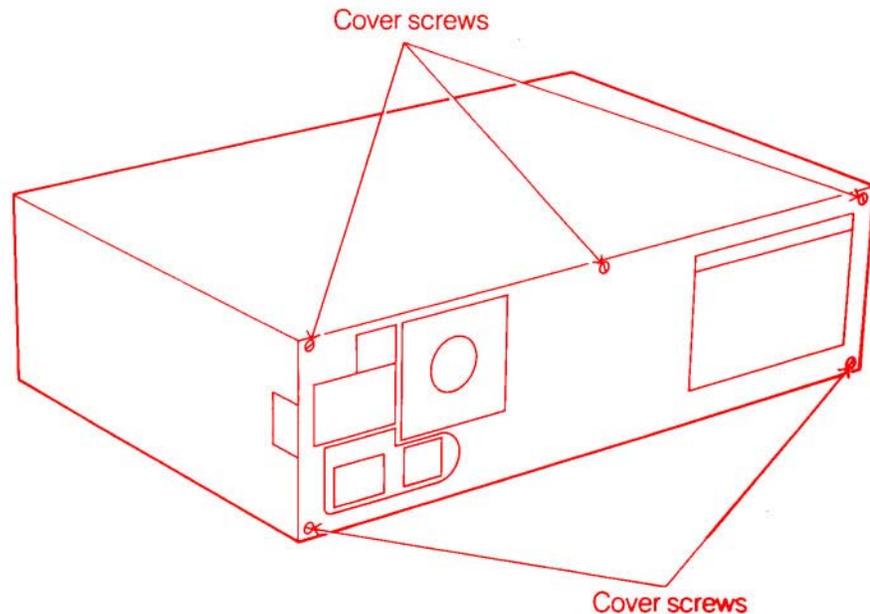
Getting started with IMS B004

4   Remove the cover, by lifting slightly, and pulling forwards. Be careful not to catch any of the internal wires with the cover.
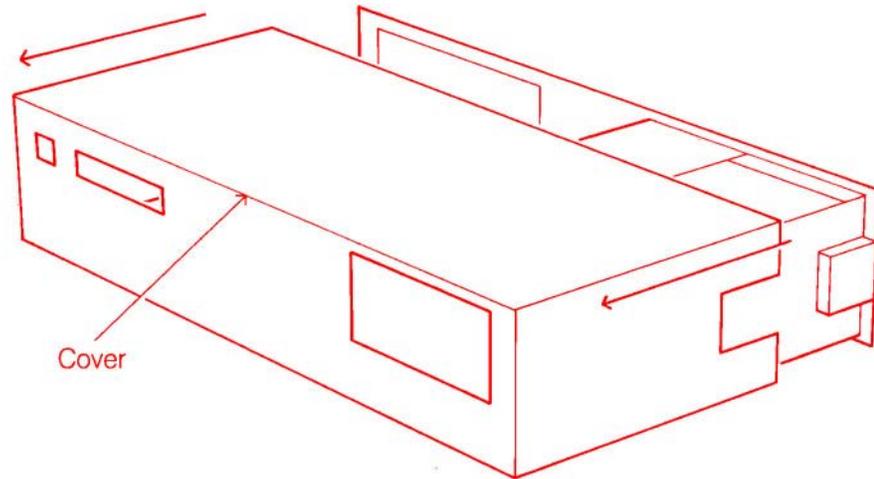
Cover

Having removed the cover, you should now be able to see a number of expansion slots at the rear left of your unit.

1 2 3 4 5 6 7 8

Rear panel
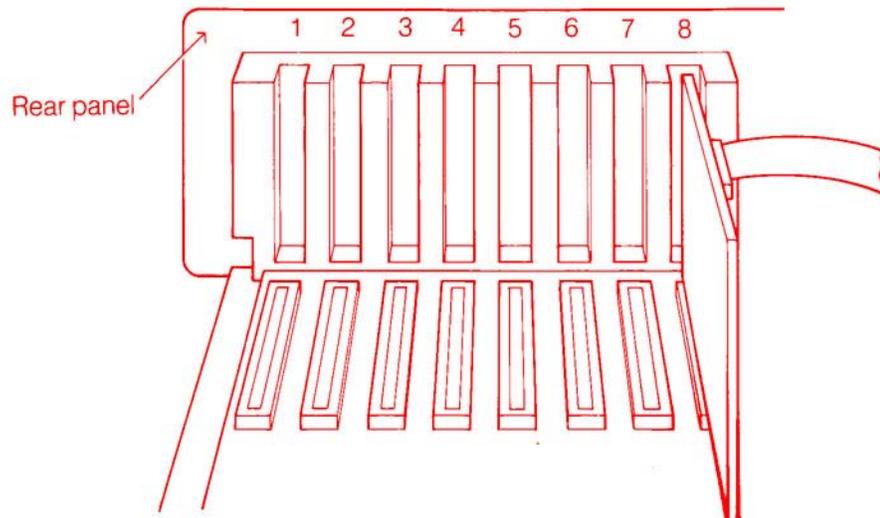
Getting started with IMS B004

1.4 Removing the B004 board from its bag

To prevent damage from static charge, certain precautions should be taken when removing the board from its protective bag.

1   While holding the board in one hand still in its bag touch any metal part of your machine with the other hand

2   Carefully remove the board from the bag, holding the board by the edges only. Avoid touching any components or connections.

3   When inserting the board into the system board. hold the board by its top edge or upper corners.

1.5	**Installing the board**	The IMS B004 board can be installed in any of the available expansion slots within the Personal Computer. While installing the board, also refer to the 'Internal option installation' chapter in the `Installation and Setup' manual for your Personal Computer.

1   Remove the screw that holds the expansion slot's cover in place, and remove the cover. The screw should be saved for the installation of the B004 board.
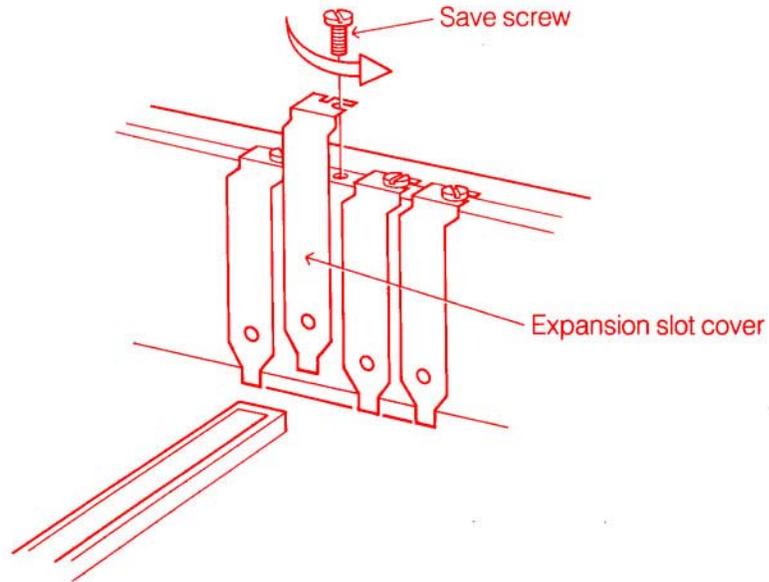
Getting started with IMS B004

2   Holding the board by the top, ensure that the edge connector of the board is aligned with the expansion slot of the machine. Press the board firmly into the slot Also ensure the rear of the board is situated correctly in the support brackets, which are attached to the inside of the front panel.

B004 board
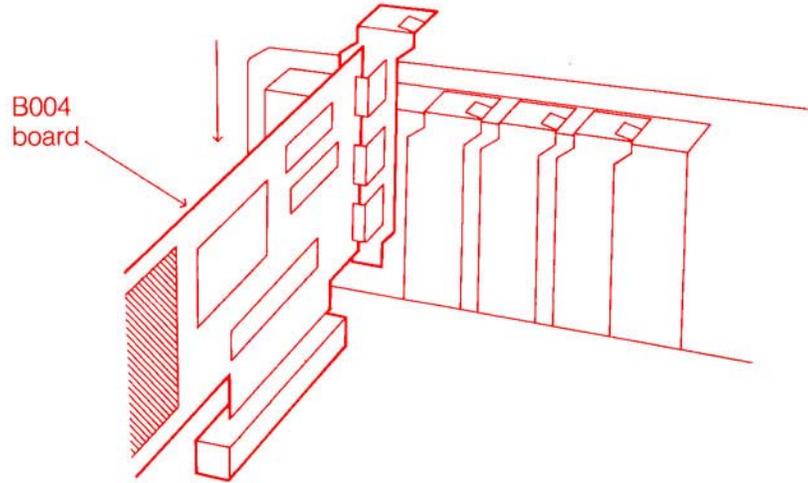
3   Refit the screw you removed in step 1.

Getting started with IMS B004

1.6   Refitting the cover   Before replacing the cover to your unit, make sure that the board edge connector is correctly inserted in the expansion slot, and the board is firmly held within the support brackets at the front of the case.

Once the board has been fitted correctly, the cover can be refitted. Whilst refitting the cover, refer to the chapter 'Cover installation' in the 'Installation and Setup' manual for specific details of your machine.

1   Ensure your Personal Computer is powered off. Push the power cables (beside the power supply) down and out of the way. They should be no higher than the power supply.
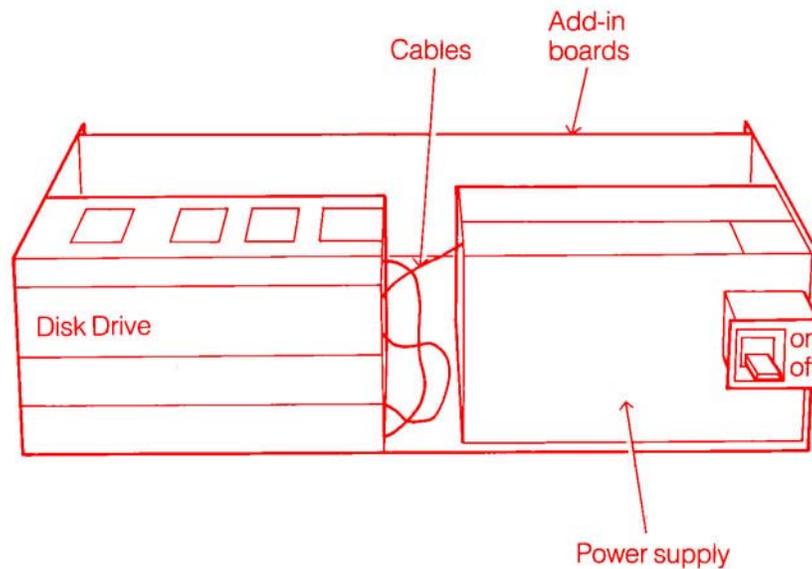
Getting started with IMS B004

2   Refit the cover Ensure the cover is under the rail on the main frame of the Personal Computer Lift the cover up against the rail, and push the cover over the main unit

Cover

Rail
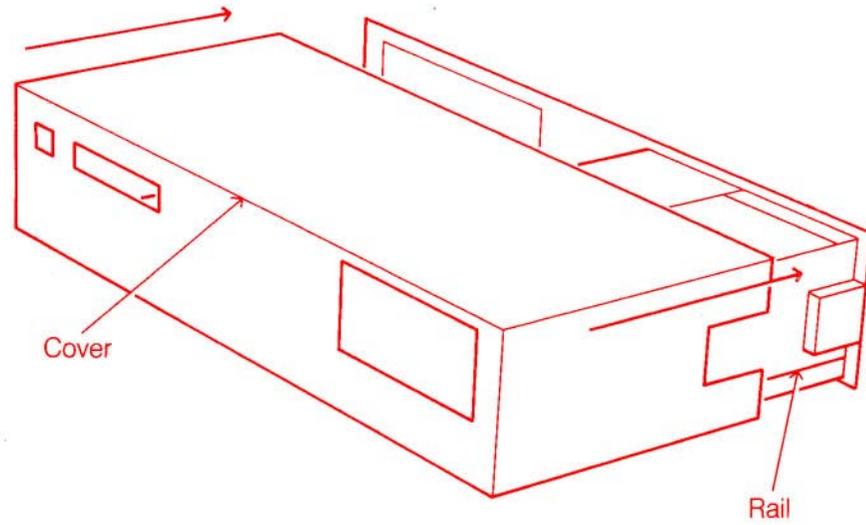
3   Install the five mounting screws.

4   Refit all the leads removed previously, ensuring they are connected to the correct sockets BEFORE switching your machine on.

1.7 **Selecting the master board**	Before any program can be downloaded to a B004 board from a Personal Computer, two jumper sockets must be fitted correctly. The use of these jumpers allows more than one B004 to be present within a Personal Computer, but allowing only one of them to respond to the TDS. The board which has the jumpers fitted is designated the Master and any number of other INMOS evaluation boards can be attached to this one via the links.

The diagram in the margin shows the connectors on the rear of the board. The view is looking at the rear of the Personal Computer, after the board and cover have been fitted. To select a board to be the master TDS board, the connectors called PCLink and PCSystem (see section 1.9) are used in conjunction with the Link and Reset jumper plugs to allow the board to communicate with the Personal Computer.

The Link jumper is the 11 way connector with the yellow leads. It is used to connect the link from the link adaptor (which handles all communication with the PC parallel bus) to link 0 of the transputer. The 11 way connector is designed such that when plugged into PCLink and link 0, a signal is asserted which, in effect, attaches the link adaptor to the parallel bus of the Personal Computer. When the board is set up with this connector in place, all communications to the Personal Computer are possible, using standard occam channel communications on link 0 of the transputer.
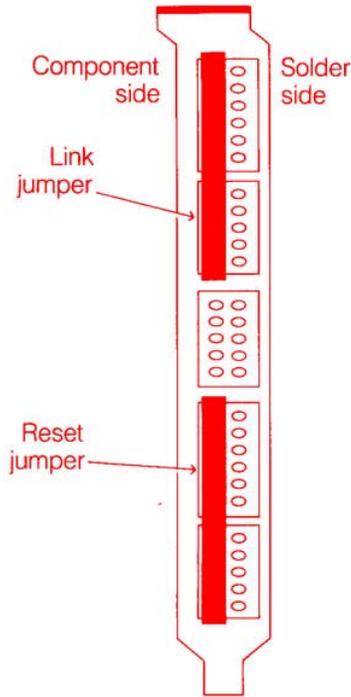
The Reset jumper is the 11 way connector with the black leads. It is used to connect the Personal Computer to the system service signals provided at the Up connector. When the jumper is in place it is possible for the Personal Computer to assert both the Up reset a-Up analyse signals, and read the Up error signal. This jumper must be in place on the Master board, as the TDS must be able to reset the entire system before loading any transputer code. If the connector is not used, the board will ignore any system commands from the Personal Computer.

The Link jumper must be inserted into the PCLink and LinkO connectors (the jumper is keyed to prevent it being placed down). The Reset jumper must be inserted into the PCSystem and Up connectors. Note that this jumper is also keyed, and cannot swapped with the Link jumper.

1.8   The selected board   The diagram below shows a view of the rear of the board after the Link and Reset jumpers have been fitted. The board is now set up to receive commands and code from the Personal Computer.

Reset jumper   Link jumper
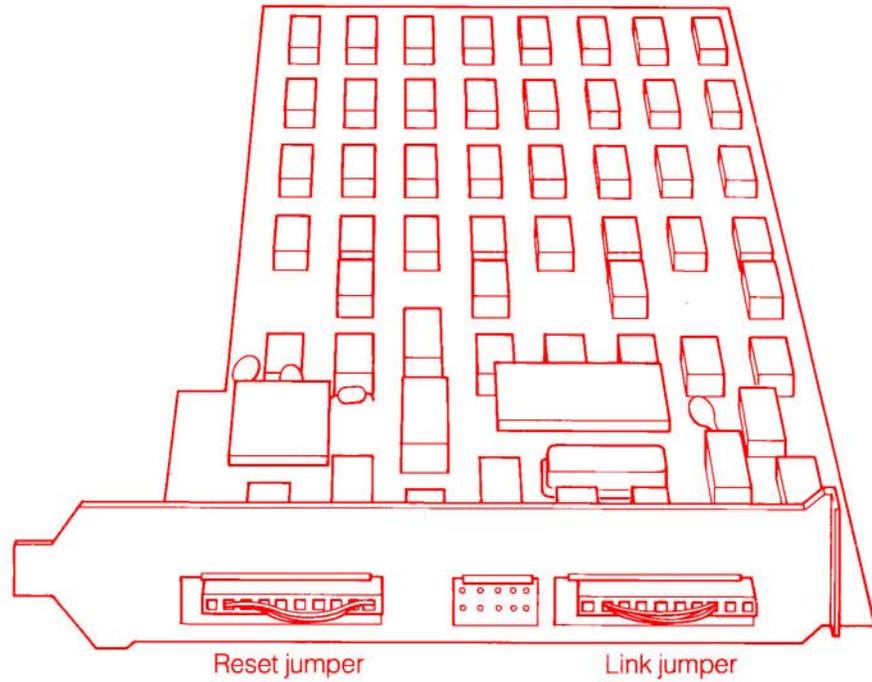
1.9   Connecting multiple boards together
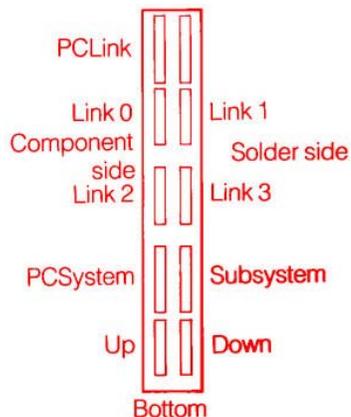
The diagram in the margin shows the rear edge connectors of the B004, looking from the rear of the board. As can be seen, there are two columns of pins, and these are grouped into sets of five, suitable for the five way sockets which terminate the various cables supplied.

The link sockets are self explanatory. The Up, Down and Subsystem sockets are concerned with system control, initialisation and error handling. The simplest way to use them is to connect the DOWN socket of the Master LIDS board to the Up socket of the next board with the Reset cable provided, and to then daisy chain the Down from each board to the Up of the next. This method ensures that when the LIDS resets the first board, all others in the chain are also reset (see the diagram below).

The Subsystem socket is used to allow one board to control its own system of boards. The subsystem Reset, Analyse and Error signals are all controllable by software on the transputer. More details of the use of the subsystem is given in section 4.

One further thing you need to do if you are using evaluation boards other that the B004 is to set them to Boot from link. See the user manual for the particular boards you are using for details.
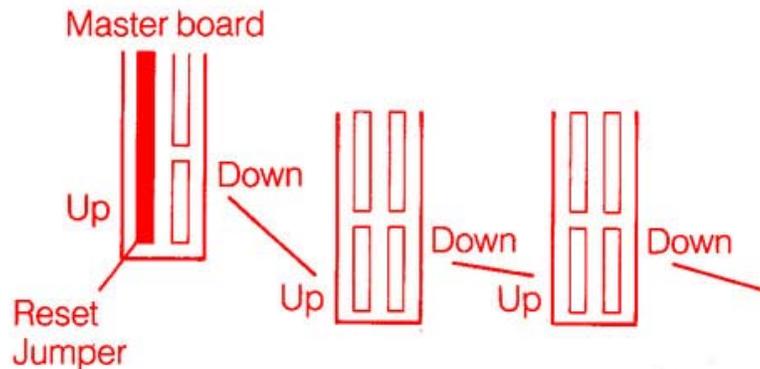
Getting started with IMS B004

1.10   Powering up

Once the board has been fitted correctly. and the cover of *your* machine replaced, then the machine can be switched on. The test software provided with the board can be found on the floppy disk which you will have found in the top of the box, along with this manual

To retrieve the programs from the floppy disk, the following should be done

1   place the floppy disk in the disk drive.

2   enter the top level directory (by typing `cd \`). The files will be placed in their own sub-directory.

3   type the command `restore a: c : \ /s`. This will read the test programs into their directory, `B4TEST`.

4   type `cd B4TEST` to enter the test directory

Typing `dir` will display the contents of the directory, which should read

```
Rmtest.b4
Linktest.b4
Ramtest.exe
Linktest.exe
Linkcom.occ
B4procs.occ
```

The use of these programs is described in more detail in section 3.

Board Architecture

The IMS B004 and other evaluation boards share a common architecture which includes the edge connector pinout, link buffering, and system control functions. The architecture is independent of the type of transputer used, its speed, and of the number of transputers on the board.

The memory map of each board is different. and the memory map for the B004 is given in section 5.1.

2.1   Links   The transputer's links are brought out to the edge connector, for the user to configure his system as he wishes. A pair of link cables is supplied with each board, giving four ends of cables which are suitable for connecting to the four links of the transputer

The four link sockets on the edge connector are:

LinkO   pins b7 to b11
Link1   pins a7 to a11
Link2   pins b13 to b17
Link3   pins a13 to a17

The link sockets and cables are coded to make it difficult to plug link cables into sockets other than link sockets.

Board Architecture

2.2   System control
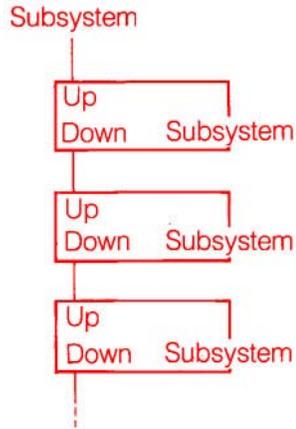
The IMS B004 and other evaluation boards share a common architecture which includes the edge connector pinout, link buffering, and system control functions. The architecture is independent of the type of transputer used, its speed, and of the number of transputers on the board.

The memory map of each board is different. and the memory map for the B004 is given in section 5.1.

The aim of the system control functions of the evaluation boards is that it should be possible to initialize, and analyse errors in an arbitrarily large system built with the boards In particular one board in a cabinet or a rack must be able to control all the other hoards in the rack.

This control is necessarily independent of the serial links between boards

2.2.1   Plugging boards together

One board, or the host Personal Computer, controls a subsystem of an arbitrary number of other boards. This is achieved with three sockets which we call Up, Down and Subsystem. The figure on the left shows a board at the top with its Subsystem socket wired to the Up socket of the board below. Subsequent boards are daisy chained by wiring the Down socket of one board to the Up socket of the board below.

Each of the boards in a subsystem can itself control a further subsystem, as in the hierarchy of boards shown below.

A cable is supplied with each board which may be used to connect the board's Up socket to either the Down socket or the Subsystem socket of the board above.

The Up socket is implemented as edge connector pins b28-32,and the Down socket as pins a28-32, the subsystem socket as pins a22-26.

15
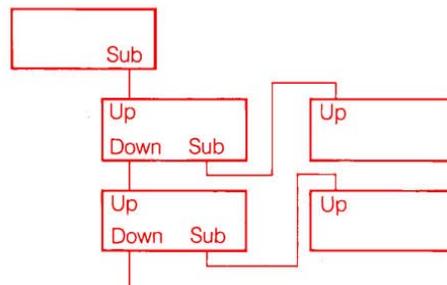
Board Architecture

2.2.2   The signals on the sockets

Each of the sockets includes the signals notReset, notAnalyse and notError.

The notReset and notAnalyse signals flow from the Up port to the Down port, but not to the Subsystem port. The notError signal flows in the reverse direction from Down to Up, and indicates that an error has occurred on this board or on a board further down from this board. All the inputs are biased so that if a socket is not used, the signals it receives are in their inactive state. The Subsystem error signal is not propagated up because the board controlling the subsystem should deal with the error.

2.2.3   Control of the board at the top of the hierarchy

One board (in this case the B004 board set up as the master board, with the reset and link jumpers in place) can be reset or analysed from the system running the TDS. These signals appear as memory locations within the Personal Computer. The error signal can also be read by the Personal Computer. A detailed description of procedures to access these locations is given in section 4.

Board Architecture

2.2.4   Logic on the board to provide the system control functions

The logic relationships between the signals on the sockets and on the transputer are shown below (ignoring logic polarities).

Reset signals

Transputer   = Up

Down   = Up

Subsystem   = Addressable Latch

Analyse signals

Transputer   = Up

Down   = Up

Subsystem   = Addressable Latch

Error signals

Up   = Transputer OR Down

Addressable input = Subsystem

Subsystem latches

The addressable latches for subsystem reset and subsystem analyse are initialized at power-on to be inactive i.e. not resetting and not analysing the subsystem. This must be core controlling the subsystem.

Board Architecture

2.3 Edge connector pinout	Pin	b	a
	1	GND	NC
	2	(missing)	(missing)
The B004 board uses a group	3	PCLinkOut	NC
of 5 way connectors, to	4	PCLinkIn	NC
simplify the location of the	5	GND	NC
various leads for a system.	6	NotLink	NC
	7	GND	GND
The NotLink (b6) and	8	(missing)	(missing)
NotSystem (b27) are used in	9	LinkOut 0	LinkOut 1
conjunction with the Link and	10	LinkIn 0	LinkIn 1
Reset jumpers described	11	GND	GND
previously. When these signals			
are at logic 0, they select the	12	(gap)	(gap)
functions associated with			
either reset or link to respond	13	GND	GND
to signals from the PC.	14	(missing)	(missing)
	15	LinkOut 2	LinkOut 3
	16	LinkIn 2	LinkIn 3
	17	GND	GND
	18	(gap)	(gap)
	19	(gap)	(gap)
	20	(gap)	(gap)
	21	(gap)	(gap)
	22	PCNotReset	SubsystemNotReset
	23	PCNotAnalyse	SubsystemNotAnalyse
	24	PCNotError	SubsystemNotError
	25	GND	GND(missing)
	26	(missing)	(missing)
	27	NotSystem	NC
	28	UpNotReset	DownNotReset
	29	UpNotAnalyse	DownNotAnalyse
	30	UpNotError	DownNotError
	31	GND	GND(missing)
	32	GND(missing)	GND(missing)

B004 Software

3.1 Using the board

Component side / Solder side

Link jumper

Reset jumper
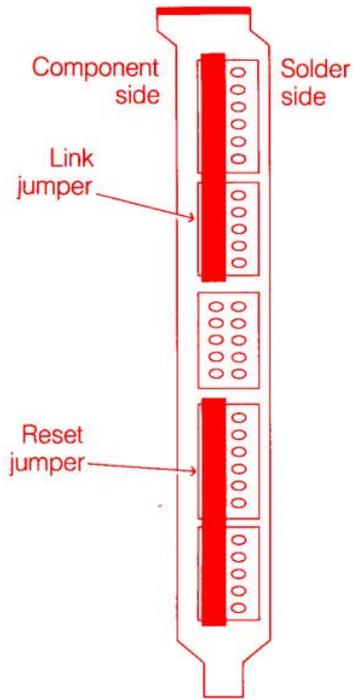
The IMS B004 add--in board is controlled solely by the host Personal Computer. To test the board, there are two test programs, which can be run independently of the transputer development system. These programs, when running, will produce a message of the form

```
Type ControL-D to resume MSDOS.
```

If this message does not appear, three things must be checked

1   The board must be correctly seated within the PC expansion slot. Refer to section 1 for installation details.

2   Ensure the Link jumper is firmly situated in the top left sockets (as looking at the rear of the board). The top diagram in the margin shows this view, and where the jumper should be placed.

3   The Reset jumper must also be correctly fitted in the bottom left sockets (as shown in the lower diagram in the margin).

If these conditions have been satisfied, and there is still no response from the board, then there is either a fault in the Personal Computer, or on the board.

B004 Software

3.2   **Testing the board**   Two programs are supplied to test the board. These are a memory test program, and a program to test the links on the rear connector. Both programs will loop indefinitely, and so by typing Control D, their execution can be halted, and control returned to the Personal Computer.

3.2.1   Ram test

To run this program, the command

Ramtest

should be entered on a normal DOS command line. The start-up message

Type Control-D to resume MSDOS

should appear, followed by

Memory size in MBytes (1 or 2) ?

Reply by typing the digit **1** or **2** as appropriate.

The program then cycles through the memory, testing the on board RAM. As it proceeds through each cycle of tests, the following sequence of messages is output

```
Filled with addresses
Checking
Filling with NOT addresses Checking
Zeroed RAM
Domino roll up memory, checking
filling with odd parity,checking
Passes = number : failures = number
```

The parity will be checked during each phase, and parity errors, along with any other errors detected during any test, will be reported to the terminal.

B004 Software

3.2.2   Link test

This test will exercise the links of the transputer. As link 0 is used to communicate with the Personal Computer, and has been used to transfer the program to the transputer, it will have already have been tested. To test the remaining links, the command

```
Linktest
```

should be entered from the DOS command line. The message

```
Type Control-D to resume MSDOS
Link Test...
Connect Link 1 to Link 2 . . . . . then press any key
```

should appear. Having connected the two link connectors at the rear edge connector with one of the link cables supplied, press a key. The test transfers 12KByte blocks of data down both directions of the links in parallel, and does this 100 times, updating the count on the screen after each pass. If a link is not operating correctly, no information will be printed on the screen. If the link is working, but an error occurs during transmission of data, the error count for that link will be incremented.

After 100 cycles, the message

```
Connect Link 1 to Link 3 . . . . . then press any key
```

will appear. This test is the same as the first, just using different links.

3.3   Example programs   Some example programs are provided with the transputer development system. The purpose of these is to help familiarize the user with the use of the boards and to act as a check that the boards are working. A suitable first example to try is the **multiboard echo** program. This and other examples are described in the TDS system manual.

IBM PC Software

The following routines can be used in conjunction with the occam programming system, to allow the Personal Computer user to communicate with the INMOS C002 Link adaptor on the B004 (refer to the Product Data in the enclosed transputer reference manual), and also control the system signals, such as reset etc. These procedures will only function correctly if the Link and Reset jumpers are in place on the Master TDS board. The source code can be found on the floppy disk supplied with the board.

Details of the operation of the procedures can be found in the source code of the routines.

4.1   Link adaptor access   Two routines are provided. one to write and one to read data to and from the link adaptor. The source code, in occam fold format, can be found in the file

```
linkcom.occ
```

Writing to the C002   The procedure

```
write.byte.to.link(VALUE data.word)
```

should be called when the user wishes to pass data to the transputer from the host Personal Computer. The parameter **data.word** will be truncated to the least significant byte of the word, and this byte written to the link adaptor, followed by three zero bytes to create a full transputer word.

Reading from the C002

The procedure

```
read.byte.from.link(VAR data.word)
```

should be called when reading data from the link adaptor The routine reads a transputer word (four bytes) from the link adaptor, but only the first (least significant) byte is passed back to the calling process in the lowest significant byte of the variable parameter. The rest of the parameter word is set to zero.

The three other bytes that made up the word from the transputer are discarded.

IBM PC Software

4.2 **IBM PC system control**

Three routines are supplied, to control reset, analyse, and read error.

Reset

The routine

```
system.reset(VALUE reset.status)
```

is used to control the reset signal. The parameter passed should have the value TRUE is the reset signal is to be asserted, and FALSE to de-assert the signal.

Analyse

The routine

```
system.analyse(VALUE analyse.status)
```

controls the analyse signal. As with reset, a value TRUE passed as the parameter will assert the analyse signal, and FALSE will de-assert it.

Errors

Reading the error signal can be accomplished using the procedure

```
read.system.error(VAR error.status)
```

If the parameter returned reads TRUE, an error has occured, else no error has been reported.

5.1   Memory map

The address space is only partially decoded using A31 and A20. so that, within the negative half of the address space. the memory map is repeated every two megabytes. If your B004 is fitted with 1 MByte of RAM, the memory map of the board is

Addresses **80000000 to 800007FF**
2KBytes of internal RAM

Addresses **80000800 to 800FFFFF**
1 MByte of external RAM

giving a total of 1 Mbyte of RAM.

If your board has two megabytes of RAM fitted, the memory map is as follows

Addresses **80000000 to 800007FF**
2KBytes of internal RAM

Addresses **80000800 to 802007FF**
2MBytes of external RAM

giving a total of over 2MBytes of memory.

All external memory has a cycle time of five processor cycles.

5.2   **Parity and Subsystem**

Routines are also supplied for use with the transputer development system to generated code to use the parity checking and subsystem port of the IMS B004. These routines can be found on the floppy disk in the file

b4procs.occ

5.2.1   Subsystem

Reset

The procedure

```
subsystem.reset(VALUE reset.status)
```

controls the subsystem reset signal. A value TRUE passed as the parameter will assert the reset signal, and the value FALSE will de-assert it.

Analyse

The procedure

```
subsystem.analyse(VALUE analyse.status)
```

controls the analyse signal, in a similar way to the reset procedure above.

Error

To read the subsystem error signal, the procedure

```
subsystem.error(VAR error.status)
```

should be used. A value TRUE returned in the variable parameter indicates an error has been signalled from the subsystem.

5.2.2   Parity

The external memory has a parity check bit for each byte of RAM If parity is enabled, and a parity error occurs, the position of that error, in terms of which byte in which bank of memory, is latched. The procedures described below allow the user to enable/disable parity and to read the parity latch to check for parity errors.

No parity checking is done on the internal memory of the transputer.

Enabling Parity   The procedure

```
enable.parity(VALUE enable)
```

enables or disables the parity checker. On passing a value TRUE in the parameter, the parity checker will be enabled, and will report the first parity error, should one occur. On error, no event occurs, and so the parity latch should be polled occasionally to check for errors. No further parity errors will be latched until the parity latch has been reset (by disabling, and then enabling parity).

On passing the value FALSE to the routine, the parity will be disabled and the parity latch cleared.

The EnableParity flag may power up in any state. The status of the EnableParity flag may not be read.

IBM PC Software

ParityToUP   It is possible to pass the parity error flag to the UP error port of the board. In this situation any parity error that occurs (when parity is enabled) will be passed up through the system to the board or host controlling the system. The procedure

```
send.parity.to.up(VALUE to.up)
```

controls the routing of the parity error flag. A value TRUE passed as a parameter will route the parity error flag to the UP port (at the rear edge connector), and a value FALSE will disconnect the flag from the port.

ReadParityError   By calling the procedure

```
read.parity.error(VAR error,byte,bank)
```

the parity latch may be read, and the status of the parity error flag determined.
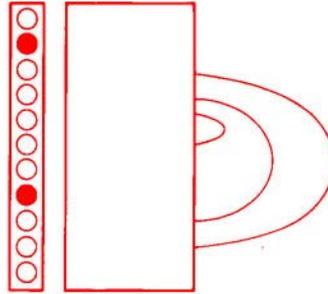
NOTE   The parameters returned by this procedure are valid only if parity is ENABLED. If this is not the case, then the value of the 'error' parameter should be ignored.
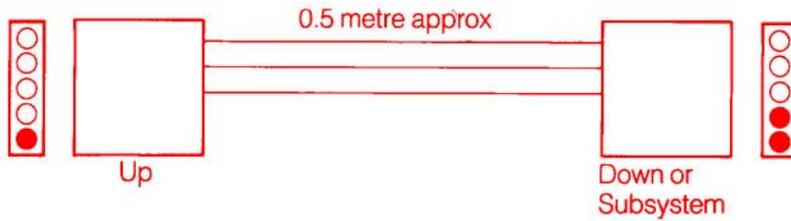
If `error' is TRUE, and parity is enabled, then the location of where the error occured can be found, in terms of the bank of memory (bank 0 s the first 1 MByte of external memory, and bank 1 the second), and the byte (0 to 3).

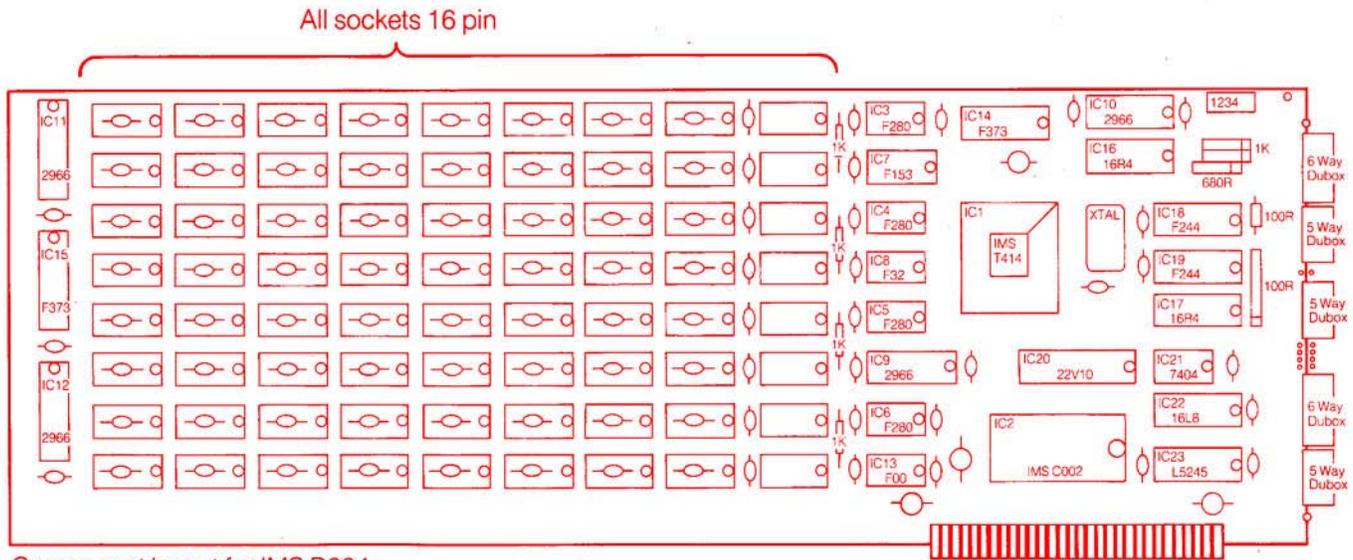# IMS B004 Evaluation board accessories
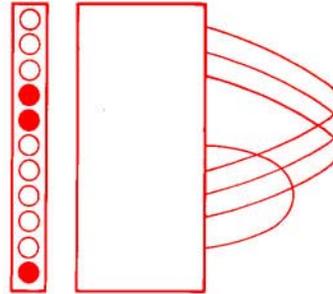
IMS B004 Component layout
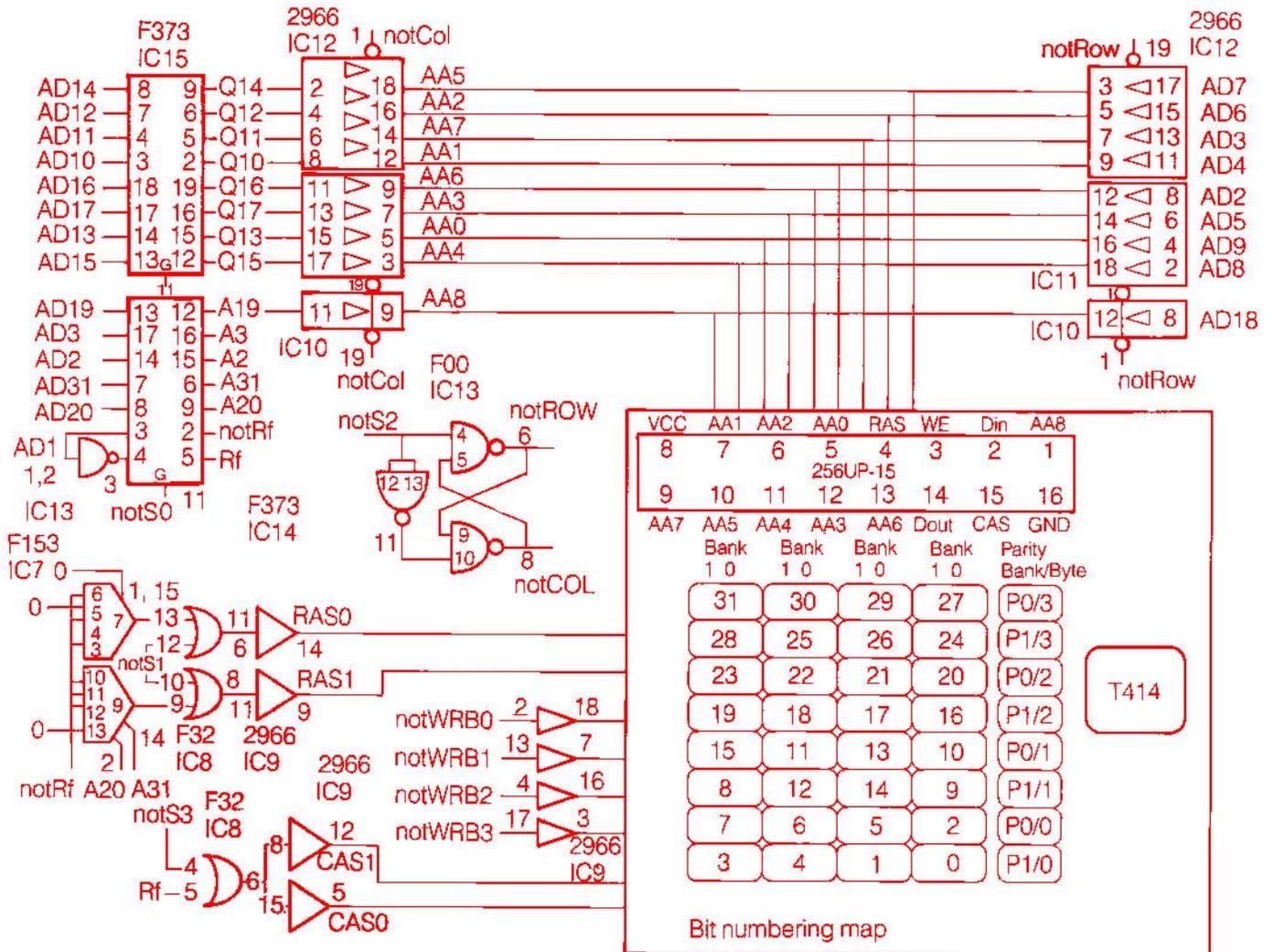

Component layout for IMS B004
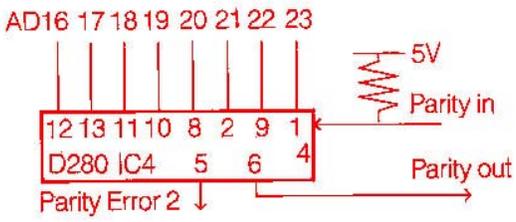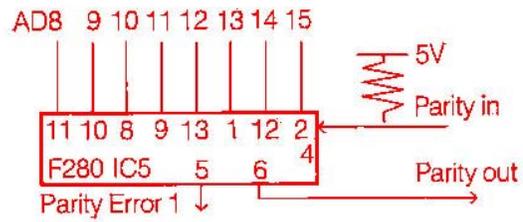For arrangement of bits in memory array, see logic diagram —O— 10uF Tant

Ø 0.1uF AVX caps

IMS B004 logic diagram

IMS B004 logic diagram

T4Link0out 15 5
T4Link1out 6 14
T4Link2out 4 16
T4Link3out 2 18

F244
IC19

100R

T4Link0in 16 4
T4Link1in 18 2
T4Link2in 3 17
T4Link3in 14 6

F244
IC18   Gnd

+5   1K

UPresent   1 2   3 4   T4reset (E3)

DNreset

F04   F04
IC21   IC21

+5

7404   T4analyse (F2)

UPanalyse   13 12 11   10   DNanalyse

F04   F04
IC21   IC21

Link

25 Mhz   8   25 Mhz   Adaptor
X tal   Clk
osc   T4Clk (B2)

Hold to GND	A2
LOSpecial	B4
L 123Special	A4
MemReq	B9
Hold to GND	E1
EventReq	B8
Hold to GND	D3
Hold to GND	D2
Hold to GND	F1
BootFromRom	E2
MemWait	A10

GROUND

X 1PLL B1
VDD

AD0 1 2 3 4 5 6 7

5V

Parity in

13 10 9 11 12 1 8 2
F280 IC6   5   6   4

Parity out

Parity Error 0 ↓

AD8 9 10 11 12 13 14 15

5V

Parity in

11 10 8 9 13 1 12 2
4
F280 IC5   5   6

Parity out

Parity Error 1 ↓

AD16 17 18 19 20 21 22 23

5V

Parity in

12 13 11 10 8 2 9 1
4
D280 IC4   5   6

Parity out

Parity Error 2 ↓

AD24 25 26 27 28 29 30 31

5V

Parity in

11 13 10 1 12 9 2 8
4
F280 IC3   5   6

Parity out

Parity Error 3 ↓

32

IMS B004 logic diagram

IMS B004 logic diagram

A31 → 6
A2 → 5
A3 → 4
AD4 → 13
AD5 → 11
AD6 → 10
notRd → 8

A31
A2
A3
AD4
AD5
AD6
AD7
— MemConfig

A2
A31
notRd
— notReadParity

A31
A2
A3

AD0

notSSReset
notSSAnalyse
Enable Parity
ParityToUp

+5V

notDNError → 9
notSSError → 3
ParityError → 2
T4Error → 7

T4Error
notDNError
ParityToUp
ParityError
— notUPError

A2
A31
notRd

notSSError — ADO

PAL 22V10
IC20

1
notWRBO

18 → MemConfig
22 → notReadParity
17 → notUPError
19 → notSSReset
20 → notSSAnalyse
21 → Enable Parity
23 → ParityToUp
15 → AD0
14 → AD7

PAL 16R4
IC17

Enable Parity — 9
A31 — 6

Parity0 — 3   14 — Pstat0 – Which Byte 0
Parity1 — 5   16 — Pstat1 – Which Byte 1
Parity2 — 7   17 — Pstat2 – Which Bank
Parity3 — 4   15 — Pstat3 – Error
A20 — 8   13 — ParityError

1   11
notRd   notReadParity