

TRANSTECH DEVICES LIMITED

TMB08 INSTALLATION AND USER MANUAL

**Transtech Devices Ltd, Unit 17, Wye Industrial Estate,
London Road, High Wycombe, Bucks. HP11 1LH.
Tel : 0494 464303 Fax : 0494 463686 Telex: 838844**

CONTENTS

1. Introduction

2. Hardware Description

- 2.1 Overview
- 2.2 Transputer Pipeline
- 2.3 TRAM Slot Location & Pipe Jumpers
- 2.4 IMSC004 Crossbar Switch Control
- 2.5 System Services (Reset, Analyse & Error)
- 2.6 The IBM Interface
 - 2.6.1 Polling the Link Adaptor
 - 2.6.2 Direct Memory Access (DMA)
 - 2.6.3 Interupts
 - 2.6.4 TMB08 Interface Registers
- 2.7 The Patch Area
- 2.8 Link Speed Selection

3. Setting up the Configuration

- 3.1 Default Settings
- 3.2 Board Address
- 3.3 Selecting the DMA and Interupt Channels
 - 3.3.1 DMA Channel Selection
 - 3.3.2 Interupt Channel Selection
- 3.4 Link Speed Selection
- 3.5 System Services (Reset, Analyse and Error)
- 3.6 Jumper Locations
- 3.7 Example Configurations
 - 3.7.1 A Single Board Running the TDS
 - 3.7.2 A Single Board as a Target System (3L Languages)
 - 3.7.3 Cascaded Boards

4. Installation

- 4.1 TRAM Fitting & Handling
- 4.2 Installing a TMB08

5. Testing the Board

6. CO04 Configuration Software

- 6.1 Listing of Occam Program for T212
- 6.2 Listing of Pascal Program

APPENDICES

- A. 37 Way D-Type Connector PINOUT
- B. 37 Way Adaptor PINOUT

1. INTRODUCTION

The Transtech range of TRAMs (**TR**ansputer **M**odules) and motherboard provides a flexible solution to building transputer systems, with a wide range of processor and memory configurations.

Transtech TRAM motherboards allow TRAMs to be connected together as well as to be connected to other transputer board products from Transtech and other suppliers, and adhere to a published TRAM standard. This document describes the Transtech TMB08 TRAM motherboard for IBM PC XT/AT systems and how to install the TRAMs.

2. HARDWARE DESCRIPTION

2.1 Overview

The TMB08 is a TRAM motherboard for use in an IBM PC XT/AT or compatibles. It has slots for up to 10 daughterboard TRAMs. TRAMs are daughterboards with some or all of the following:- transputers, memory, peripheral circuitry, and they communicate with motherboards using transputer serial links. Each slot on the motherboard is made up from 16 DIL (Dual-in-line) pins which provide 4 links, reset, analyse, error, power, ground, clock and link speed selection to the TRAM. The 10 slots are arranged in a hard wired pipeline so that links 2 to 1 of consecutively numbered slots are pre-connected. The remaining links can be configured using an IMSC004 crossbar switch which is supplied on the TMB08 and can be set up from a PROGRAM running on a T212 16-bit transputer on the board. Multiple TMB08 boards can be cascaded to build larger systems, with the T212 transputers connected in a chain to allow configuration of a network on multiple motherboards, by sending the relevant configuration information to the IMSC004's connected to each T212 transputer.

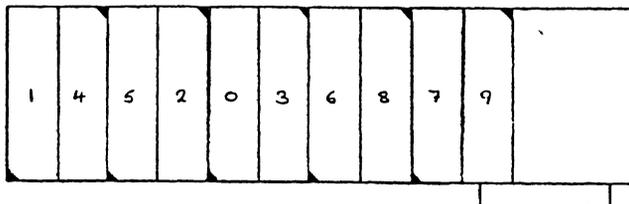
The TMB08 also has an interface to the IBM bus so that a program on the PC can communicate with the TRAMs on the TMB08. There are two options for communication between the IBM bus and transputers, one is via software polling of the link adaptor and the second is via a DMA mechanism which provides a much higher data rate.

2.2 Transputer Pipeline

The TMB08 has 10 slots for TRAM daughterboards. When fully populated with 10 size 1 TRAMs a hardwired pipeline of processors is formed with link 2 from each TRAM connected to Link 1 of the next TRAM in the chain. At the ends of this pipeline are two links, Module 0 link 1 and Module 9 link 2 which are termed Pipehead and Pipetail respectively. Pipetail is taken to the 37 way D-type connector at the edge of the board to allow it to be connected to another transputer board. Pipehead is taken to a 16 way patch area on the board to allow it to be connected to various other links. The remaining links 0 and 3 of each TRAM slot are connected to the IMSC004 crossbar switch to allow more complicated network topologies to be built. Further information on the programming of the C004 is provided in sections 2.4 and 6.

2.3 TRAM Slot Location & Pipe Jumpers

The diagram outlines the physical location of the TRAM slots on the TMB08. It is clearly shown that adjacent TRAM slots have alternate orientation.



ALTERNATE ORIENTATION OF TRAM SLOTS

When some TRAMs are fitted to the TMB08 which are larger than size one, they will physically cover more than one TRAM slot. A larger TRAM will only take signals from one slot but will take power and ground from adjacent slots. The hardwired pipeline will be broken on adjacent slots when a size 2 or larger TRAM is fitted.

Transtech's TRAMs follow two design rules to overcome this problem. The first is to use 8 way connectors, called pipe jumpers, supplied with the TMB08 motherboard. TRAMs can be stacked on top of some of the TRAMs as the signals from the motherboard slot below are propagated through to a further set of DIL pins on top of the TRAM. When stacking of TRAMs is not necessary or not required, a pipe jumper needs to be inserted at one end of the TRAM covering the unused slot. This pipe jumper connects links 1 and 2 on the unused slot to continue the hardwired pipe. The pipe jumper needs to be orientated so that its yellow dot aligns with the white dot on the motherboard, indicating pin 1 of the TRAM slot.

Some of Transtech's TRAMs have a height profile which does not easily allow stacking of TRAMs. These TRAMs incorporate the pipejumper connectors onto the PCB, so that all unused TRAM slots underneath them already have their link 1 and 2 connected, when the TRAM is inserted, to complete the hardwired pipeline.

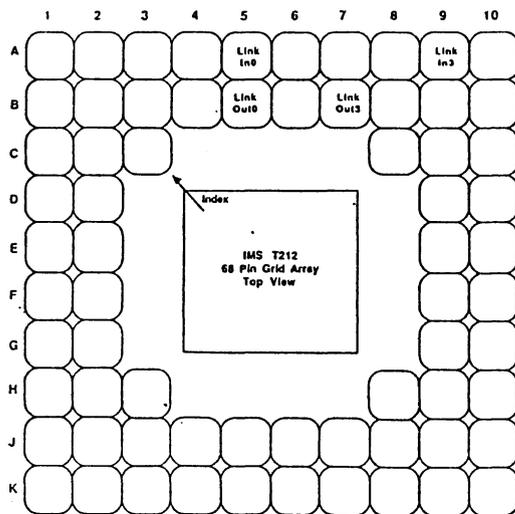
2.4 IMSC004 Crossbar Switch Control

The IMSC004 crossbar switch allows software control of the topology of the transputer network. The C004 is a 32 way link switch with 32 link inputs and outputs and is configured via a configuration link. Links 0 and 3 from TRAM slots 1 to 9 and link 3 from TRAM slot 0 are directly connected to the C004. There are ten other links connected to the C004, two of these go to the 16 way patch area and eight to the 37 way edge connector and are designated Edge Link 0 to Edge Link 7 respectively. TRAM0 link0 can also be connected to the C004 if required, but only via the patch area. This enables TRAM0 link0 to be connected to the PC via the IMSC012 link adaptor which also terminates at the patch area. Full details on setting up the patch connectors are to be found in section 3.7.

There are two methods of programming the IMSC004. Firstly, the configuration link of the IMSC004 is programmed from an IMST212 16-bit transputer. The IMST212 has link 3 connected to the C004, and uses links 1 and 2 to receive and transmit configuration information. Link 1 is termed **configup** and link 2 **configdown**.

Configdown goes to the 37 way edge connector to be connected to the next board in the chain, while configup goes to the patch area to allow the source of the configuration information to be selected. The "master" or first board in the chain has its configup connected to **Pipehead** (TRAM 0 link 1) for information to be downloaded from the TDS running on the root processor. In this way a PROGRAM for a network of T212 transputers can be downloaded to program all C004's as required.

Secondly, the IMSC004 can be programmed via a link from a link adaptor, memory mapped into the PC bus. This method requires a second link adaptor to be installed on the board in a spare socket marked C012(2) above the other IMSC012 which provides the PC interface to TRAM slot 0. The IMST212 needs to be removed and the link from the link adaptor (2) needs to be connected to the C004 crossbar switch. The link from link adaptor (2) is connected to link 0 of the T212, and link 3 of the T212 is connected to the configuration of the C004. Hence with the T212 removed, link 0 on the T212 socket needs to be connected to link 3. These connections are shown in the diagram below.



CONNECT PIN A5 TO B7
AND PIN B5 TO A9

Once the hardware adjustments have been made, the Pascal program supplied on the floppy disk can be edited and compiled to set up the relevant connections on the C004. The program can be included in a batch file on the PC to allow the transputer network configuration to be set up without the need to use the TDS.

Further details on the software provided to support both techniques of programming the C004 are contained in Section 6.

2.5 System Services (Reset, Analyse & Error)

Transputers in a system as well as having links connected together must also have system services connected to control the resetting and analysis of errors in the system. The TRAM in slot 0 can either be reset by the PC bus or by another board via the 37 way edge connector. When the TRAM on slot 0 is reset from the PC bus, it can also be analysed by the PC bus as well as having any error ready by the PC bus. All of these functions can also be controlled via the 37 way edge connector.

TRAMs in slots 1 to 9 can either be controlled by the same source as TRAM 0, (PC bus or 37 way edge connector) or from the subsystem port of TRAM 0. The subsystem port allows TRAM 0 to control the reset, analyse and error functions of TRAMs 1 to 9.

The selection of the source of system services is detailed in section 3.5, and discussed in the example board configuration in section 3.7.

2.6 The IBM Bus Interface

The IBM interface consists of 3 elements, 2 link adaptor systems (1 of which is optional) and a BIOS EPROM which sits at #D000X in the IBM memory map and is 32KBytes long.

The link adaptor systems are for communication between the PC bus and transputer network and optionally a method of programming the C004 crossbar switch as discussed in section 2.4.

The EPROM is not supplied as standard, but by adding it to the available socket on reset of the PC the ROM code can be read and acted upon. This can be useful for end user systems, where the user may not know about transputer systems, but has a developed software package running on the supplied hardware.

There are two ways to use the link adaptor(1) (IMSC012) to communicate between the PC bus and transputer system, described in the following two sections.

2.6.1 Polling the Link Adaptor

This method is the simplest form of data transfer between the PC bus and the TMB08. Earlier Transtech boards used only this method and the TMB08 is totally compatible with them. It is often referred to as a "B004" interface.

The status of the registers of the IMSC012 are continuously polled by software running on the PC. Further details are contained in the IMSC012 data sheet which is included in the Transputer Reference Manual, available from Prentice Hall. Before any data can be sent to the TMB08, it must first be reset by software.

2.6.2 Direct Memory Access (DMA)

The data rate for polling the link adaptor is slow. This is why a DMA interface facility has been developed for the TMB08 to satisfy demands for higher data rates.

This technique uses the 8237 DMA controller chip in the PC, details of which can be found in the 8237 data sheet.

The DMA interface can be used in different ways, once the 8237 has been initialised. A transfer to the TMB08 is started by the program running on the PC writing a 0 to the DMA control register of the TMB08. A "1" is written to the PC to receive data from the TMB08.

The TMB08 makes a DMA request for a single byte at a time, so a byte is transferred between the execution of each instruction on the PC. Hence the

PC appears to be running code at the same time as the DMA transfer is taking place. The end of the DMA transfer is signalled by either the 8237 being polled by reading its status register, giving information about which DMA channels have a transfer pending and which have completed transfers, or by an interrupt being set up. Further details on interrupts appear in the next section.

2.6.3 Interrupts

There are four sources of interrupt that the TMB08 can use to signal to the PC.

1. Interrupt on end of DMA transfer
2. Interrupt on transputer error
3. Interrupt on link data input ready
4. Interrupt on link data output ready

The interrupt control register is used to determine which of these events cause an interrupt. Four bits are used in this write only register (see section 2.6.4). Writing a "1" to a bit interrupts on the relevant event and a "0" disables the interrupt.

To determine the source of an interrupt it is required to read status registers. The **error flag**, **Output Int** and **Input Int** are status link bits present in the TMB08 registers, while the interrupt on DMA end can be detected by reading the status register of the 8237 DMA controller in the PC.

2.6.4 TMB08 Interface Registers

The registers of the TMB08 are located in the PC I/O address space detailed below.

| Board Address | Register |
|-----------------|-------------------------------|
| boardbase + #00 | Input data register |
| boardbase + #01 | Output data register |
| boardbase + #02 | Input status register |
| boardbase + #03 | Output status register |
| boardbase + #10 | Reset register (write only) |
| boardbase + #11 | Analyse register (write only) |
| boardbase + #12 | Error location (read only) |
| boardbase + #13 | DMA request register |
| | Interrupt control register |

Boardbase is the base address and can be #150 or #200 as detailed in section 3.2.

The function of the first four registers is also detailed in the IMSC012 data sheet.

The others which are specific to the TMB08 are detailed here:

Reset Register (Write Only)

Writing bit 0 : to "1" asserts Reset : to "0" deasserts.

Analyse Register (Write Only)

Writing bit 0 : to "1" asserts Analyse : to "0" deasserts.

Error Register (Read Only)

Reading bit 0 : as "1" indicates Error : as "0" no error.

DMA Request Register (Read/Write)

Writing bit 0 : as "0" indicates DMA from PC to TMB08

: as "1" indicates DMA from TMB08 to PC

IRQ control register (Write Only)

Four bits are used to control the function of the register.

Writing "1" to bit 0 enables interrupt on end of DMA.

Writing "1" to bit 1 enables interrupt on error.

Writing "1" to bit 2 enables interrupt when TMB08 is ready to receive byte.

Writing "1" to bit 3 enables interrupt when TMB08 is ready to transmit byte.

Writing "0" to any bit disables the interrupt.

2.7 The Patch Area

The Patch Area is a 16 pin DIL header block which has 8 links terminating at it. (A link consists of two wires). These 8 links allow different configurations to be set up for the TMB08 to be used as a "master" or "slave" board. The links are:

TRAM slot 0 link 0

IMSC012(1)

Pipehead

Configup

Patch link 0

Patch link 1

C004 link 28

C004 link 29

The options on connecting these links are detailed in the configuration section 3.7 showing example configurations.

2.8 Link Speed Selection

The link speed of all links on the TMB08 is selectable for 10 or 20 Mbits/sec and is detailed in the configuration section 3.4.

3. SETTING UP THE CONFIGURATION

The TMB08 is designed to be set up in generally two different modes, firstly, as a "Master" board with a "root" processor TRAM in slot 0 interfaced to the PC bus and secondly, as a slave board forming part of a larger transputer system where all the communication is with other boards and not the PC. The various configurations and options are set up using different jumpers on the board.

3.1 Default Settings

The boards default configuration is to be set up as a Master board and this is how the TMB08 will be supplied from Transtech.

| | |
|-------------------|-------------------------|
| Boardbase address | #150 |
| Interupt channel | 3 |
| DMA channel | 1 |
| Link Speed | 20Mbits/sec |
| Module 0 reset | FROM IBM |
| Module N reset | from Module 0 subsystem |

These settings are discussed further, together with explanations of how and why they should be altered in the following sections.

3.2 Board Address

The default address of the board is between #150 and #163 (# is a hexadecimal number) in the I/O space of the PC. If this clashes with any other cards in the PC there is an option to change the address to be between #200 to #213.

An option of the TMB08 is to have a second link adaptor to allow programming of the IMSC004 crossbar switch directly from the PC bus. This second link adaptor can be at #250 or #300. The second link adaptor option is discussed in section 2.4.

The selection of these addresses is made by using different combinations of jumpers over the pins A1, A2 and A3 as indicated in the table in section 3.6.

3.3 Selecting the DMA and Interupt Channels

Near the bottom edge of the board are a series of single pin sockets to allow the selection of the DMA and interupt channels.

3.3.1 DMA Channel Selection

| | | | | | | |
|---|---------|---------|---|---------|---------|-----------------------|
| 1 | DRQ | 2 | 1 | DMA | 2 | |
| * | * | * | * | * | * | No DMA selection |
| * | *-----* | * | * | *-----* | * | Selects DMA Channel 1 |
| * | * | *-----* | * | * | *-----* | Selects DMA Channel 2 |

The default setting of the board is to use DMA Channel 1 which is available on both PC AT and XT systems. However, DMA channel 1 is often used for ethernet or other networking cards. DMA channel 2 is the floppy disk DMA channel, which can be disabled by writing I/O to memory location #3F2, allowing it to be used by the TMB08 for data transfer. Whenever a floppy disk access is required, the disk BIOS always enables location #3F2, this means that the TMB08 driver must disable location #3F2 and enable its drivers at the start of a DMA transfer, and then disable its drivers at the end of a transfer. Separate software routines are provided for handling the different DMA channels.

3.3.2 Interrupt Channel Selection

IRQ channel 3 is the default setting of the TMB08. This is normally the setting for a second RS232 port, while the other option IRQ channel 6 is associated with the diskette driver. The software routines for DMA channel 1 support IRQ3 and the routines for DMA channel 2 support IRQ6.

| | | | |
|----------|---------|----------|-----------------------|
| Channel3 | IRQ | Channel6 | |
| * | * | * | No IRQ Selection |
| * | *-----* | * | IRQ Channel3 selected |
| * | * | *-----* | IRQ Channel6 selected |

3.4 Link Speed Selection

The TMB08 is supplied with all links running at a default of 20Mbits/second. To change the links running at 10Mbits/second a jumper needs to be inserted between the two pins marked "S". This affects all links on the T212, C004, link adaptors and fitted TRAMs. The location of the link speed jumper is shown in section 3.6.

3.5 System Services (Reset, Analyse & Error)

The default settings assume a TRAM in slot 0 is the Master TRAM connected via its link 0 to link adaptor (1) and hence the PC bus. Consequently the TRAM in slot 0 is reset by the PC, and the remaining TRAM on the board are controlled from TRAM 0's subsystem. The location of the jumpers used to control these functions is shown in section 3.6. More details are to be found in the example configuration in section 3.7.

3.6 Jumper Locations

There are 7 jumper selectors on the main board, and are located under TRAM slot 9 close to the T212 transputer, as shown. They select as indicated.

```

*   *   *   *   *   *   *
I   0   R   1   2   3   S
*   *   *   *   *   *   *
    
```

| JUMPER | FUNCTION | LINK IN |
|--------|--|--|
| | LINK OUT | |
| I | Module 0 reset from PC | Module 0 reset from edge connector |
| 0 | Module 1 to 9 reset from Mod 0 SubSystem | Module 1 to 9 reset from same source as Module 0 |
| R | ROM not selected | ROM selected at #D000X |
| A1 | See address tables below | |
| A2 | See address tables below | |
| A3 | See address tables below | |
| S | Link speed 20Mbits/sec | Link speed 10Mbits/sec |

I, 0 and S are discussed in sections 3.4 and 3.5. R is discussed in section 2.6.

| ADDRESS TABLE | | | | |
|---------------|----|----|---|-------------------|
| A3 | A2 | A1 | boardbase L/A(2) (optional link adaptor) | boardbase L/A (1) |
| 0 | 0 | 0 | - | - |
| 0 | 0 | 1 | - | #200 |
| 0 | 1 | 0 | #250 | - |
| 0 | 1 | 1 | #250 | #200 |
| 1 | 0 | 0 | #300 | #200 |
| 1 | 0 | 1 | #300 | #150 |
| 1 | 1 | 0 | #300 | - |
| 1 | 1 | 1 | - | #150 |

0 indicates link in
1 indicates link out

Boardbase L/A (1) is the address of the IMSC012 which provides the interface to the PC bus from the transputer system as discussed in section 3.2.

Boardbase L/A (2) is the address of the second optional IMSC012 which can be used to program the C004 crossbar switch as discussed in section 2.4.

The jumper I, 0, A1, A2 and A3 also appear near the edge of the board above the 37 way D-type connector as a series of headers. These headers have small link jumpers on one pin equivalent to jumper link out as the default. It is often easier to use these jumpers rather than the ones under slot 9 to set up the configuration.

3.7 Example Configurations

To help in understanding the configuration options, three examples of configured TMB08's are discussed here.

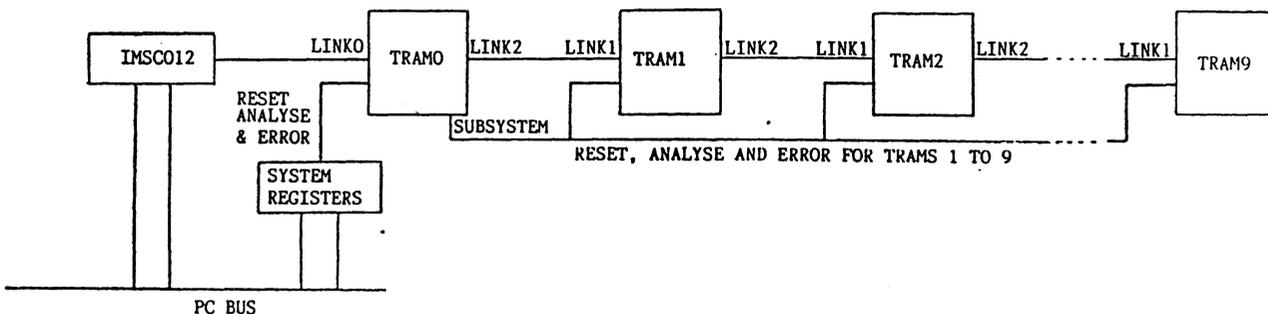
3.7.1 A Single Board Running the TDS

This is the default configuration in which the board is supplied.

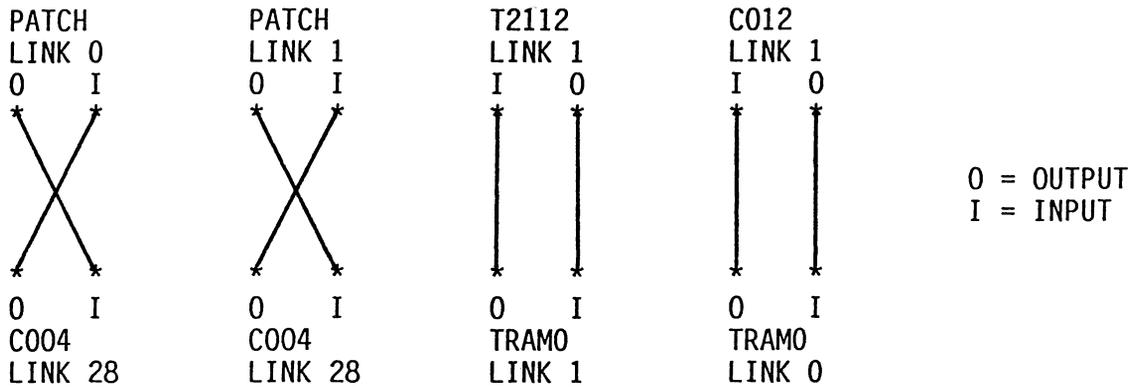
The TDS should be run on a TRAM plugged into slot 0 of the TMB08, with a server program running on the PC under MS.DOS. The two communicate via link adaptor 1, memory mapped onto the PC bus at either #150 or #200, which goes to the patch where it is connected to the TRAM in slot 0 via link 0.

The reset, error and analyse functions for the TRAM in slot 0 are under control from the PC bus (i.e. jumper "I" is out). The TRAM's in slots 1 to 9 and the T212, derive their reset under control of TRAM0's subsystem. This means that TRAM0 can reset and analyse all the transputers in the system as well as read the status of their error (i.e. jumper "0" is out).

A SINGLE BOARD RUNNING THE TDS



The "config up" link of the T212 is its link 1, and this must be connected to link 1 of TRAM 0, known as "pipehead". Consequently, configuration information can be downloaded from TRAM0 to the T212 which in turn passes it to the IMSC004 crossbar switch. These connections are made via the patch area.



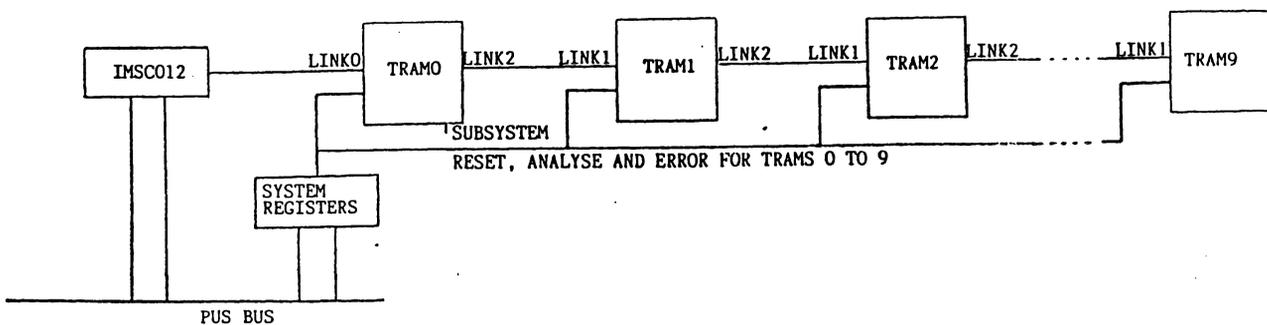
The other connections made on the patch area are Patch Link 0 and 1 connected to C004 links 28 and 29, which provide two more (making 10 in all) link connections from the C004 to the edge connector.

This configuration allows the TDS to run on the root processor and download code to a network of target processors controlled by the root processors subsystem. Each time the command LOAD NETWORK is used the target system is reset by the subsystem immediately prior to the code being loaded. When the root processor is booted it is first reset by the PC. As a processor is reset it will automatically reset its subsystem, hence preventing any code running on processors in the target network from sending unexpected data to the root processor to interfere with the application running on the root processor.

3.7.2 A Single Board as a Target System

This is the configuration required to run the 3L scientific languages, where a bootable code file is loaded onto the transputer network from the PC. It can also be used for bootable code files produced using the TDS which are loaded to the transputer network under server support from the PC.

A SINGLE BOARD AS A TARGET SYSTEM (3L LANGUAGES)



The link between TRAM 0 Link 0 and the PC bus via the Link adaptor 1 is maintained from the previous example as the communication route between the two.

However, the reset analyse and error functions of TRAMs 1 to 9 are now controlled from the same source as TRAM 0, i.e. the PC bus. This means jumper 0 is now in, while jumper "I" is still out allowing control from the PC bus.

This configuration means that a reset to TRAM 0 is propagated directly to TRAMs 1 to 9 and the T212 without going via the subsystem of TRAM 0. The same effect can be achieved in the configuration in 3.7.1 where the reset is propagated via the subsystem hence the PC resets all the transputers at once. The difference with this new configuration is that the PC can also analyse all the TRAMs at once and not just TRAM 0, and that the PC can monitor the error status of all the transputers whose error signals are logically OR'd together, rather than just the error status of TRAM 0.

3L's languages can be run on the configuration in 3.7.1. but in that case it is not possible to receive errors from TRAMs other than TRAM 0 or to analyse TRAMs other than TRAM 0.

Since the TDS cannot reset the T212 immediately before downloading code to configure the T212 in this configuration, it is suggested that the C004 crossbar switch is programmed by the Turbo Pascal program running on the PC addressing the link adaptor 2 as discussed in section 2.4. The patch connection between TRAM 0 link 1 and the T212 link 1 can remain, but since the T212 is removed that connection will not be used. The link adaptor (2) will be connected directly to the C004 by means of connections made to the T212 socket as shown below. The link adaptor 2 can be at boardbase address #250 or #300 and is selectable by reference to the table in section 3.6. The registers for link adaptor 2 are the same as for link adaptor 1 as described in section 2.6.4 except that the boardbase address is different.

3.7.3 Cascaded Boards

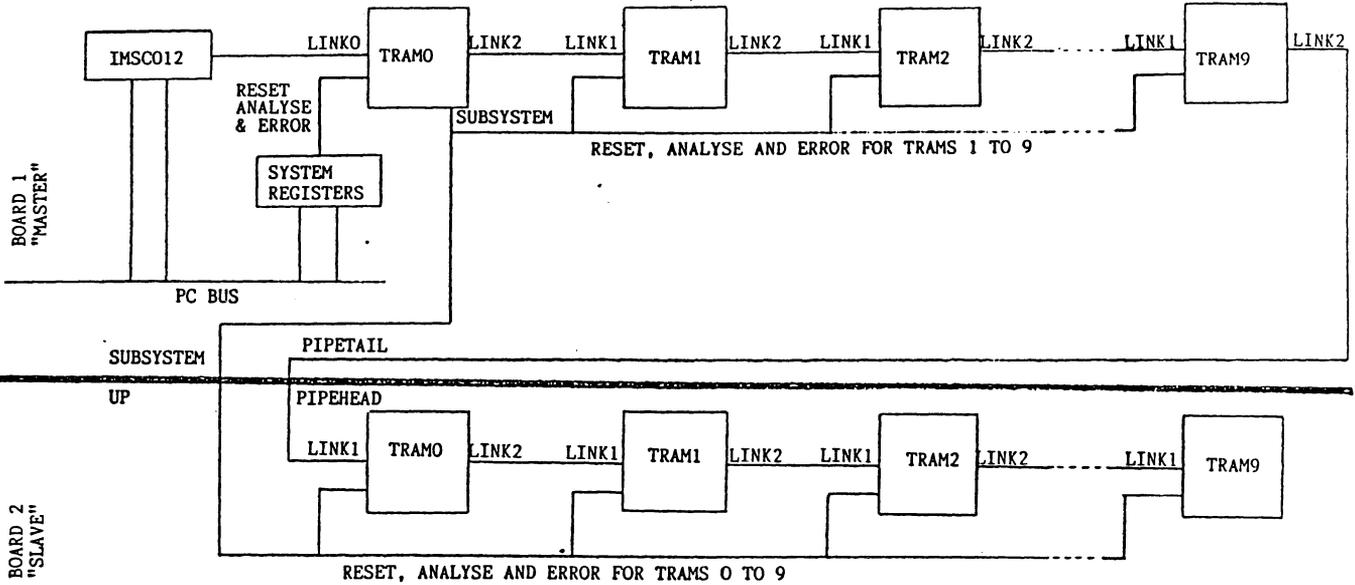
An example of two TMB08 boards cascaded together is described here. The TDS is running on TRAM 0 of a "master" board and another "slave" board with additional TRAMs is connected to it.

The master board should be set up exactly as detailed in section 3.7.1 for the TDS running on a single board.

The "slave" board should be invisible to the PC bus, with its link adaptor and system registers disabled. This is achieved by removing the connections between TRAM 0 link 0 and link adaptor 1.

TRAM 0 link 1 on the slave board should be connected to TRAM 9 link 2 on the "master" to continue the "link 2 to link 1" hardwired pipeline discussed in section 2.2. This is why TRAM 0 link 1 is known as "pipehead". To enable this connection TRAM 0 link 1 should be connected to Patch link 1 on the patch area.

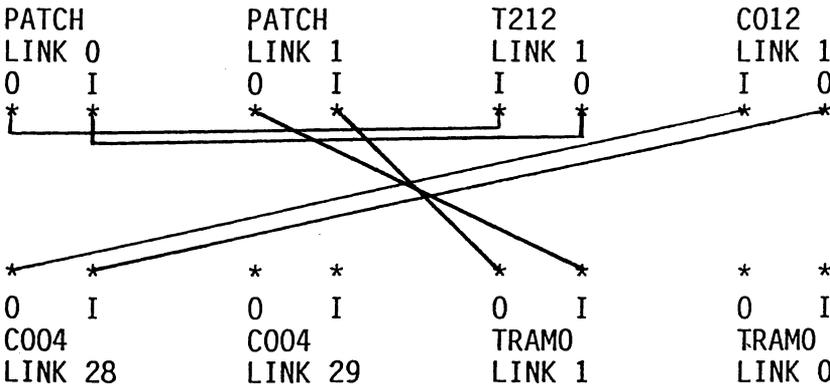
TWO CASCADED BOARDS



To allow the configuration of the slave board to be programmed from the master board the T212 link 1 ("config up") should be connected to the patch link 0 on the patch area, to allow it to be connected to the T212 on the master board via config down on the masterboard's edge connector.

The final adjustment to the patch area is not necessary but may be useful and allows an extra link to be connected to the C004, i.e. TRAM 0 link 0 to C004 link 28.

Hence the patch area should be configured as follows:



TRAMO should derive its system services control from the "UP" on the 37 way edge connector, hence jumper I should be in. TRAMs 1 to 9 and the T212 should also receive control from the "UP" so jumper 0 should be in too.

The slave board is now configured and ready to be connected to the master board. The following connections need to be made between the two boards.

| Master Board | Slave Board |
|--|---|
| Subsystem Pipetail (TRAM 9 LINK 2) Config Down (T212 LINK 2) | Up system Patch link 1 (TRAM 0 LINK 1) Patch link 0 (T212 LINK 1) |

The pin out of the 37 way D-type edge connector is given in appendix A. The pin out of an adaptor board which converts the 37 way connector to the standard Inmos 5 way connectors for links and system services is in appendix B, to allow the supplied cables to be used to make the connections between boards. These cables are designed for development only and it is suggested that when a final configuration is known a customised cable should be assembled to connect the two boards together.

More boards may be cascaded together following similar principles to these detailed here. Other types of transputer boards can also be connected.

4. INSTALLATION

4.1 TRAM Fitting and Handling

Care must be taken when fitting or removing TRAMs from the TMB08, to ensure no damage occurs to the TRAM pins. A white circle in the corner of each TRAM slot indicates Pin 1 of that slot on the TMB08. TRAMs also have an indicator for pin 1, which should be matched with the marking on the motherboard to ensure correct orientation. A TRAM plugged in the wrong way round may result in damage to the TRAM and the motherboard.

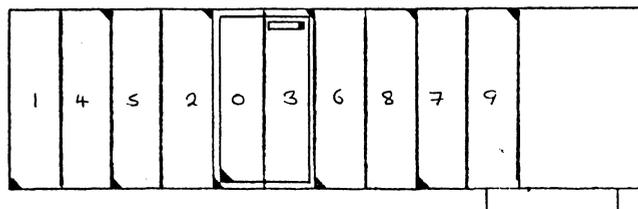
The TMB08 has some components mounted on the board between the TRAM sockets, which foul some of the TRAMs available. These can effect TRAM slots 6, 7, 8 and 9. If a TRAM cannot be fitted without fouling these components, stand off strips or larger pins supplied with the TRAM should be fitted to allow the TRAM to be raised above these components.

4.2 Installing a TMB08

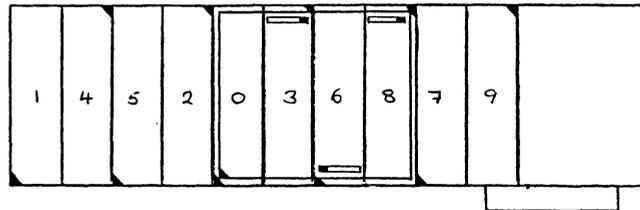
Remove the TRAM that is to be the root processor from its protective packing, observing the appropriate anti-static handling precautions. If the TRAM is to run the TDS its subsystem pins will need to be connected to the 3 sockets marked E.R.A. (Subsystem) near to Pin 1 on TRAM slot 0 of the TMB08. This is achieved using a 3 pin 2 way header supplied with the TMB08 or in some cases with the TRAM. You may also need, depending on the TRAM or length of subsystem pins, to use a 3 way spacing strip supplied with the TMB08. The 3 pin header (subsystem pins) fit into 3 sockets on the underside of the TRAM near Pin 1 so that when the TRAM is fitted to the board the 2 way header makes a connection between the subsystem sockets on the TRAM and the TMB08.

N.B. The subsystem pins are only needed if the Subsystem of TRAM 0 is needed to control other TRAMs in the system. Refer to section 3.7 and the example configurations.

Plug the TRAM into slot 0 ensuring pin 1 matches pin 1 on the TMB08. The 16 pins that carry the signals should fit into the required slot, so a TRAM larger than size 1 that fits over more than one slot will cover adjacent slots to TRAM slot 0, i.e. a size 1 TRAM will cover slots 0 and 3 and a size 4 TRAM will cover slot 0, 3, 6 and 8 as shown in the diagrams below.



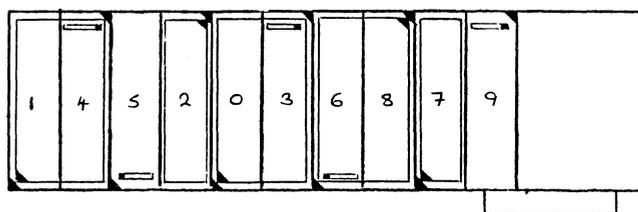
A SIZE 2 TRAM FITTED TO SLOT 0 WITH PIPE JUMPER IN SLOT 3



A SIZE 4 TRAM FITTED TO SLOT 0 WITH PIPE JUMPERS IN SLOTS 3, 6 AND 8

If the TRAM is larger than size 1 insert link jumpers in the slots that do not carry signals to ensure the continuity of the hardwired pipeline. A jumper is an 8 way connector that connects link 1 to link 2 on a slot that does not carry any signals, and should be inserted at one end of the slot with its indicator lining up with the Pin 1 index of the slot. If a TRAM is stacked on top of it the jumpers need to be removed from the slots used for stacking.

Install other TRAMs as required in the other slots of the TMB08, not necessarily in order, provided that the pipeline is maintained as discussed in section 2.2. Pipe jumpers should also be fitted in the slots that are not being used as shown in the following diagram where slots 5 and 9 have been jumped out to complete the pipeline.



SIZE 2 TRAMS FITTED TO SLOTS 0, 1 AND 8, AND SIZE 1 TRAMS IN SLOTS 2 AND 7. SLOTS 5 AND 9 ARE JUMPED OUT. PIPE JUMPERS ARE ALSO IN SLOTS 3, 4 AND 6.

Once all the TRAMs have been fitted follow the instructions in your PC manual for installing an option board.

5. TESTING THE BOARD

A test program to test the functionality of the board and the IBM interface is supplied on floppy disk.

Copy the content of this disk to a directory called TMB08 and run the test program by typing:

:>test

followed by enter.

A prompt will appear asking if the default settings have been changed, if not, type "n" followed by enter.

The program tests various parts of the board briefly and will check if the TMB08 has been set up and installed correctly. If the program reports a fault, it should indicate an error with the relevant jumper settings and or patch connections.

If the default settings have changed type "y" in response to the question and follow the questions inserting the new address and channels etc. (N.B. for address #200 type **\$200**)

Once your board is confirmed as functional refer to your software manuals and start to consider installation of your software system.

6. C004 CONFIGURATION SOFTWARE

There are example programmes supplied on the floppy disk to demonstrate the possible ways to configure the C004 on the TMB08 board. These programs are listed here. Please refer to section 2.4 to decide which program you will use.

Firstly, there is an Occam PROGRAM which is intended to be compiled for the T212 which can be extended to cover any number of T212's in a chain of cascaded boards. This example can be used as a template to set up for any C004 configuration.

C004 Occam PROGRAM

```

{{{ SC C004set.tsr
{{{F C004set.tsr
PROC C004.from.T2.set ()

  {{{ declarations
  INT A, B :
  {{{ constants
  {{{ link declarations
  VAL Link0Out IS 0:
  VAL Link1Out IS 1:
  VAL Link2Out IS 2:
  VAL Link3Out IS 3:
  VAL Link0In IS 4:
  VAL Link1In IS 5:
  VAL Link2In IS 6:
  VAL Link3In IS 7:
  }}}
  CHAN OF BYTE to.C004, from.C004:
  PLACE to.C004 AT Link3Out:
  PLACE from.C004 AT Link3In:
  }}}

```

```

{{{ c004 link definitions
VAL Unused1      IS  0 :
VAL Mod1L0       IS  1 :
VAL Mod2L0       IS  2 :
VAL Mod3L0       IS  3 :
VAL Mod4L0       IS  4 :
VAL Mod5L0       IS  5 :
VAL Mod6L0       IS  6 :
VAL Mod7L0       IS  7 :
VAL Mod8L0       IS  8 :
VAL Mod9L0       IS  9 :
VAL Mod0L3       IS 10 :
VAL Mod1L3       IS 11 :
VAL Mod2L3       IS 12 :
VAL Mod3L3       IS 13 :
VAL Mod4L3       IS 14 :
VAL Mod5L3       IS 15 :
VAL Mod6L3       IS 16 :
VAL Mod7L3       IS 17 :
VAL Mod8L3       IS 18 :
VAL Mod9L3       IS 19 :
VAL EdgeL0       IS 20 :
VAL EdgeL1       IS 21 :
VAL EdgeL2       IS 22 :
VAL EdgeL3       IS 23 :
VAL EdgeL4       IS 24 :
VAL EdgeL5       IS 25 :
VAL EdgeL6       IS 26 :
VAL EdgeL7       IS 27 :
VAL C004L28      IS 28 :
VAL C004L29      IS 29 :
VAL Unused2      IS 30 :
VAL Unused3      IS 31 :
}}}
}}}
}}}
}}} procedures
}}} output.byte ( VAL BYTE byte )
PROC output.byte (VAL BYTE byte)
SEQ
to.C004 ! byte
:
}}}
}}} func0 ( input, output )
PROC func0 ( VAL INT input, output )
SEQ
to.C004 ! BYTE 0
output.byte (BYTE input)
output.byte (BYTE output)
:
}}}}

```

```

{{{ func1 ( link1, link2 )
PROC func1 ( VAL INT link1, link2 )
  SEQ
    to.C004 ! BYTE 1
    output.byte (BYTE link1)
    output.byte (BYTE link2)
:
}}}}
{{{ func3
PROC func3 ()
  SEQ
    to.C004 ! BYTE 3
:
}}}}
{{{ func4
PROC func4 ()
  SEQ
    to.C004 ! BYTE 4
:
}}}}
{{{ func5 ( output )
PROC func5 ( VAL INT output )
  SEQ
    to.C004 ! BYTE 5
    output.byte (BYTE output)
:
}}}}
{{{ func6 ( link1, link2 )
PROC func6 ( VAL INT link1, link2 )
  SEQ
    to.C004 ! BYTE 6
    output.byte (BYTE link1)
    output.byte (BYTE link2)
:
}}}}
SEQ
  func4 ()
  A,B := Mod1L0,Mod1L3
  func1 ( A,B )
  A,B := Mod2L0,EdgeL7
  func1 ( A,B )
  A,B := Mod0L3,Mod2L3
  func1 ( A,B )
  func3 ()
:
PLACED PAR
  PROCESSOR 0 T2
  C004.from.T2.set ()

```

Secondly, there is a Pascal programme which can be run on the PC and access the C004 directly via link adaptor (2). This programme can be used as a template, edited and re-compiled to produce a programme to configure the C004 as required. This programme could be included in a batch file to set up the required configuration at the start of a given application run.

C004SET.PAS Pascal programme

CONST

{ TMB08 REGISTERS }

```

boardbase      :    INTEGER = $0300 ;
inputData      :    INTEGER = 0    ;
outputData     :    INTEGER = 0    ;
inputStatus    :    INTEGER = 0    ;
outputStatus   :    INTEGER = 0    ;
resetM2        :    INTEGER = 0    ;
analyseM2      :    INTEGER = 0    ;

Unused1        :    BYTE  = 0    ;
Mod1L0         :    BYTE  = 1    ;
Mod2L0         :    BYTE  = 2    ;
Mod3L0         :    BYTE  = 3    ;
Mod4L0         :    BYTE  = 4    ;
Mod5L0         :    BYTE  = 5    ;
Mod6L0         :    BYTE  = 6    ;
Mod7L0         :    BYTE  = 7    ;
Mod8L0         :    BYTE  = 8    ;
Mod9L0         :    BYTE  = 9    ;
Mod0L3         :    BYTE  = 10   ;
Mod1L3         :    BYTE  = 11   ;
Mod2L3         :    BYTE  = 12   ;
Mod3L3         :    BYTE  = 13   ;
Mod4L3         :    BYTE  = 14   ;
Mod5L3         :    BYTE  = 15   ;
Mod6L3         :    BYTE  = 16   ;
Mod7L3         :    BYTE  = 17   ;
Mod8L3         :    BYTE  = 18   ;
Mod9L3         :    BYTE  = 19   ;
EdgeL0         :    BYTE  = 20   ;
EdgeL1         :    BYTE  = 21   ;
EdgeL2         :    BYTE  = 22   ;
EdgeL3         :    BYTE  = 23   ;
EdgeL4         :    BYTE  = 24   ;
EdgeL5         :    BYTE  = 25   ;
EdgeL6         :    BYTE  = 26   ;
EdgeL7         :    BYTE  = 27   ;
C004L28        :    BYTE  = 28   ;
C004L29        :    BYTE  = 29   ;
Unused2        :    BYTE  = 30   ;
Unused3        :    BYTE  = 31   ;

```

```

VAR
    A      :      BYTE      ;
    B      :      BYTE      ;

PROCEDURE initconst ;
BEGIN
    inputData      := boardbase      ;
    outputData     := boardbase + 1  ;
    inputStatus    := boardbase + 2  ;
    outputStatus   := boardbase + 3  ;
    resetM2       := boardbase + $10;
    analyseM2     := boardbase + $11;
END ;

PROCEDURE initC012 ;
BEGIN
    PORT [ inputStatus ] := 0 ; { Disable inputInt }
    PORT [ outputStatus ] := 0 ; { Disable inputInt }
END ;

FUNCTION dataPresent : BOOLEAN ;
BEGIN
    dataPresent := ODD ( PORT [ inputStatus ] ) ;
END;

FUNCTION outputReady : BOOLEAN ;
BEGIN
    outputReady := ODD ( PORT [ outputStatus ] ) ;
END;

PROCEDURE outByte ( b : BYTE ) ;
BEGIN
    PORT [ outputData ] := b ;
    WHILE NOT ODD ( PORT [ outputStatus ] ) DO ;
END ;

FUNCTION inByte : BYTE ;
BEGIN
    WHILE NOT ODD ( PORT [inputStatus] ) DO ;
    inByte := PORT [ inputData ] ;
END;

PROCEDURE loopFor ( i : INTEGER ) ;
BEGIN
    WHILE i <> 0 DO i := i-1 ;
END ;

```

```

PROCEDURE func0 ( input , output : BYTE ) ;
BEGIN
    outbyte ( 0 ) ;
    outbyte ( input ) ;
    outbyte ( output ) ;
    WRITELN ( 'Connecting : ',input,' input. To ',output,' output' ) ;
END ;

```

```

PROCEDURE func1 ( link1 , link2 : BYTE ) ;
BEGIN
    outbyte ( 1 ) ;
    outbyte ( link1 ) ;
    outbyte ( link2 ) ;
    WRITELN ( 'Connected : ',link1,' to ',link2 ) ;
END ;

```

```

PROCEDURE func2 ( output , answer : BYTE ) ;
BEGIN
    outbyte ( 2 ) ;
    outbyte ( output ) ;
    answer := inbyte ;
    WRITELN ( 'output : ',output,' is connected to ',answer ) ;
END ;

```

```

PROCEDURE func3 ;
BEGIN
    outbyte ( 3 ) ;
    WRITELN ( 'C004 SET' ) ;
END ;

```

```

PROCEDURE func4 ;
BEGIN
    outbyte ( 4 ) ;
    WRITELN ( 'C004 RESET' ) ;
END ;

```

```

PROCEDURE func5 ( output : BYTE ) ;
BEGIN
    outbyte ( 5 ) ;
    outbyte ( output ) ;
    WRITELN ( 'Disconnecting output : ',output ) ;
END ;

```

```

PROCEDURE func6 ( link1 , link2 : BYTE ) ;
BEGIN
    outbyte ( 6 ) ;
    outbyte ( link1 ) ;
    outbyte ( link2 ) ;
    WRITELN ( ' Disconnecting link : ',link1,' from ',link2 ) ;
END ;

```

```

PROCEDURE doReset ;
BEGIN
    PORT [ analyseM2 ] := 0 ;
    loopFor ( 800 ) ;
    PORT [ resetM2 ] := 1 ;
    loopFor ( 3000 ) ;
    PORT [ resetM2 ] := 0 ;
    loopFor ( 1000 ) ;
END ;

PROCEDURE clearError ;
BEGIN
    doReset ;
    initC012 ;
    outByte ( 4 ) ;
    outByte ( $22 ) ; outByte ( $F9 ) ; outByte ( $60 ) ; outByte ( $0E ) ;
END ;

PROCEDURE SetError ;
BEGIN
    doReset ;
    initC012 ;
    outByte ( 6 ) ;
    outByte ( $22 ) ; outByte ( $F9 ) ; outByte ( $25 ) ;
    outByte ( $F8 ) ; outByte ( $21 ) ; outByte ( $F0 ) ;
END ;

PROCEDURE loadT2code ;
VAR
    data      : BYTE ;
    bootcode  : FILE OF BYTE ;
BEGIN
    ASSIGN ( bootcode , 'listener.m2' ) ;
    RESET ( bootcode ) ;
    WRITELN ( 'Loading bootcode to transputer ... ' ) ;
    REPEAT
        READ ( bootcode , data ) ;
        outByte ( data ) ;
    UNTIL EOF ( bootcode ) = TRUE ;
    WRITELN ( ' Loaded code ' ) ;
END ;

PROCEDURE switches ;
BEGIN
    boardbase := $0200 ;
END ;

```

```
BEGIN
  WRITELN ( ' SET UP FOR C004 CONFIG ' ) ;
  switches ;
  initconst ;
  doreset ;
  initC012 ;
  func4 ;
  A := Mod2L0 ; B := Mod3L3 ; func1 ( A,B ) ;
  A := Mod1L0 ; B := EdgeL0 ; func1 ( A,B ) ;
  A := EdgeL7 ; B := Mod1L3 ; func1 ( A,B ) ;
  func3 ;

END.
```

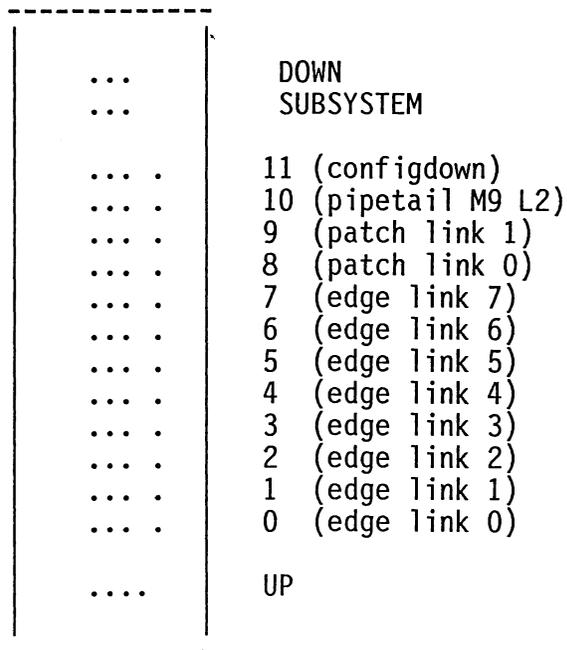
APPENDICIES

A. 37 WAY D-TYPE CONNECTOR PIN OUT

| | | | |
|----------------------|----|----|--------------------|
| down notError | 19 | 37 | down not Analyse |
| down notReset | 18 | 36 | configdown in |
| configdown out | 17 | 35 | pipetail in |
| pipetail out | 16 | 34 | notSubsystem Error |
| notSubsystem Analyse | 15 | 33 | notSubsystem Reset |
| Patch link1 in | 14 | 32 | Patch link1 out |
| Patch link0 in | 13 | 31 | Patch link0 out |
| GND | 12 | 30 | Edge link7 in |
| Edge link7 out | 11 | 29 | Edge link6 in |
| Edge link6 out | 10 | 28 | Edge link5 in |
| Edge link5 out | 9 | 27 | GND |
| Edge link4 in | 8 | 26 | Edge link4 out |
| Edge link3 in | 7 | 25 | Edge link3 out |
| Edge link2 in | 6 | 24 | Edge link2 out |
| GND | 5 | 23 | Edge link1 in |
| Edge link1 out | 4 | 22 | Edge link0 in |
| Edge link0 out | 3 | 21 | up notError |
| up notAnalyse | 2 | 20 | up notReset |
| GND | 1 | | |

B. 37 WAY D-TYPE ADAPTOR

There is a D-type socket adaptor supplied to plug into the 37 way D-type connector at the edge of the board. This adaptor enables standard link cables to be attached. The pin outs of this adaptor are as follows:



C. REFERENCES

1. Transputer Reference Manual, Inmos Limited
Prentice Hall ISBN 0-13-929001-X
2. Transputer Technical Notes, Inmos Limited
Prentice Hall ISBN 0-13-929126-1
3. Intel 8237 data sheet, Intel Limited