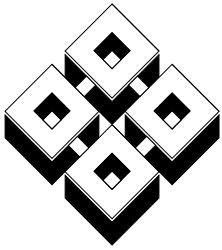


TRANSTECH

Parallel Systems

Transputer Motherboard User Manual

Ref: TMB M 711

Document reference number TMB M 711.

Transtech Parallel Systems Corp
20 Thornwood Drive
Ithaca
NY 14850-1263
USA

tel: 607 257 6502
fax: 607 257 3980

transtech@transtech.com
http://www.transtech.com

Transtech Parallel Systems Ltd
17-19 Manor Court Yard
Hughenden Avenue
High Wycombe
Bucks, HP13 5RE
United Kingdom

tel: +44 (0) 1494 464303
fax: +44 (0) 1494 463686

support@transtech.co.uk
http://www.transtech.co.uk

Table of Contents

Chapter 1
Introduction

This manual describes the Transtech transputer module (TRAM) motherboards.

Chapter 2	gives an introduction to the concepts and nomenclature of transputer modules. All users should read this before attempting to configure a TRAM motherboard.
Chapter 3	gives a detailed description of the TRAM standard for modules and motherboards. This chapter contains more advanced information required for fault-finding, designing compatible hardware or for systems programming.
Chapter 4	describes the TMB03 low-cost motherboard for PC.
Chapter 5	describes the TMB04 motherboard for PC with transputer.
Chapter 6	describes the TMB08 motherboard for PC.
Chapter 7	describes the TMB12 double extended eurocard motherboard.
Chapter 8	describes the TMB14 6U VME slave motherboard.
Chapter 9	describes the TMB16 high performance motherboard for PC.
Chapter 10	describes the TMB17 high performance PCI motherboard for PC.
Chapter 11	describes the test software and utilities provided with the boards.

Chapter 12 provides a detailed trouble-shooting guide.

Chapter 2
Using Transputer Modules

This chapter contains a beginners guide to setting up and using transputer equipment based on TRAMs. It discusses simple configurations of transputers for the most popular programming environments and how to link two motherboards together to construct larger networks. If you have purchased a standard configuration PARAstation it will already be configured for you and the following should only be read to understand the concepts behind your parallel processing system. For more detailed information refer to Chapter 3 *'The TRAM Standard'*.

2.1 Introducing TRAMs

TRAMs are small assemblies based on transputers with a standard electrical and mechanical interface. They plug onto standard motherboards which in turn plug into or can be connected to a range of host computers. This allows TRAMs and motherboards from different vendors to be plugged into a wide variety of computing platforms, giving the user the ability to construct a computing machine which meets their requirements exactly in terms of performance and IO function.

2.1.1 Hardware Description

This section contains a description of the transputer hardware which allows users unfamiliar with TRAMs to understand the installation procedure described in the next section.

The smallest size of TRAM measures about 3.5" long by 1" wide (10cm by 2.5cm). This is called a Size 1 TRAM. The size of larger TRAMs is always a multiple of the Size 1 TRAM. The size 1 TRAM

has 16 pins which plug into sockets on the motherboard. TRAMs are marked in one corner (pin 1) for orientation purposes. See figure 1.

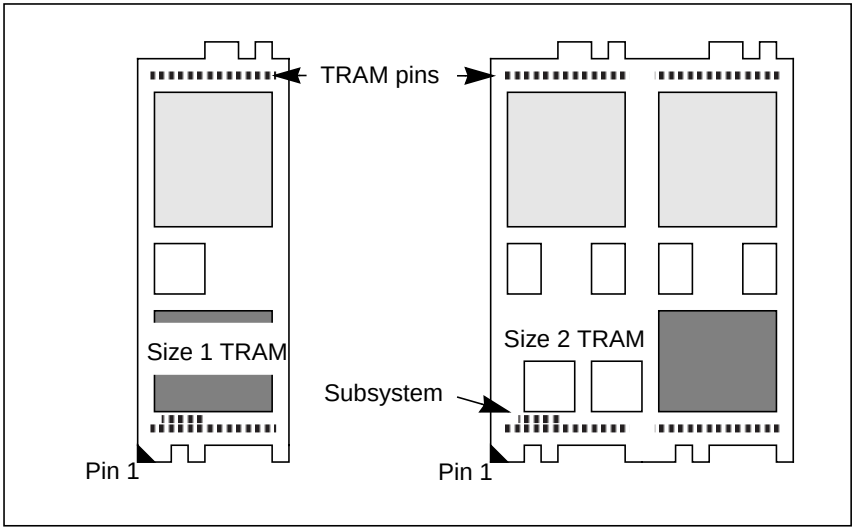


Figure 1. Transputer Modules

Some TRAMs have a subsystem - this consists of three zero profile sockets mounted on the underside of the TRAM in one corner (always next to pin 1). Only slot 0 on the motherboard has the capacity to accept the subsystem from such a TRAM. Figure 2 shows

how to used the supplied double ended connector to plug these sockets into similar sockets on the motherboard.

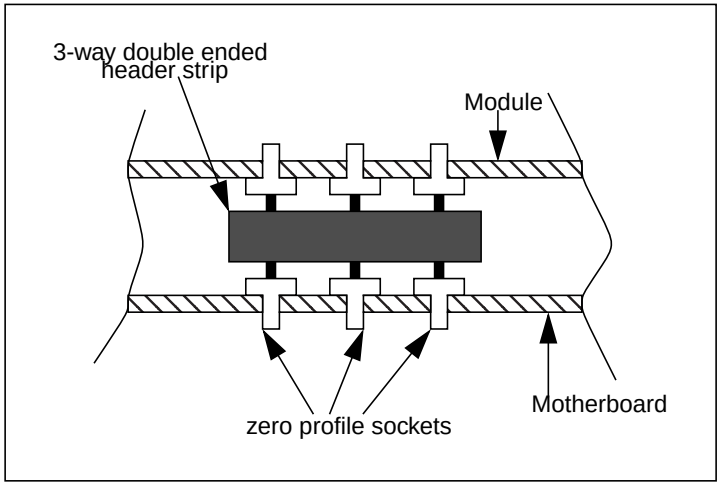


Figure 2. Subsystem Port Connections

Because of the way the subsystem connects to the motherboard it is not compulsory to perform this connection if it is not needed and hence TRAMs with subsystem can plug into any socket on the motherboard.

The location where the TRAM plugs onto a motherboard is called a slot or site. Slots are numbered from zero. Figure 3 shows a slice of a typical motherboard. Note that the slots are not all oriented the same way and that the ordering of slots is not contiguous. This allows better utilization of the motherboard when plugging in TRAMs of different sizes.

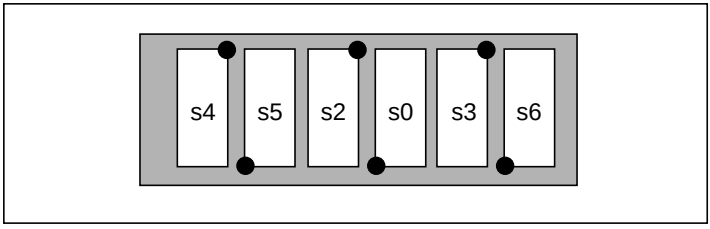


Figure 3. TRAM slots on a motherboard

The motherboard is specially wired so that if it is populated with size 1 TRAMs then the transputers are all connected in a pipeline. This is

achieved by connecting link2 of one transputer to link1 of the next transputer. See figure 4.

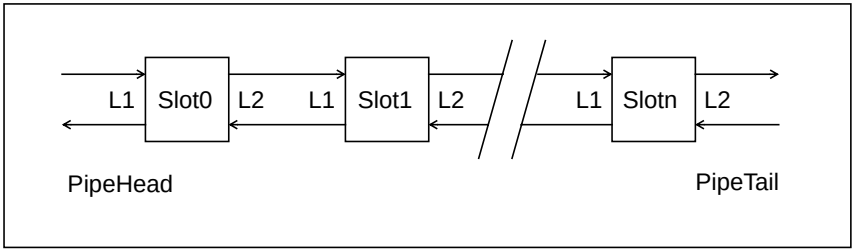


Figure 4. The Default Transputer Pipeline

TRAMs which are larger than size 1 do not use all of the sites underneath them. The only active site is the one below pin 1 of the TRAM. This means that the pipeline is broken at the unused slots underneath the TRAM. To bridge these breaks a special pipe jumper can be used. Figure 5 shows a pipe jumper.

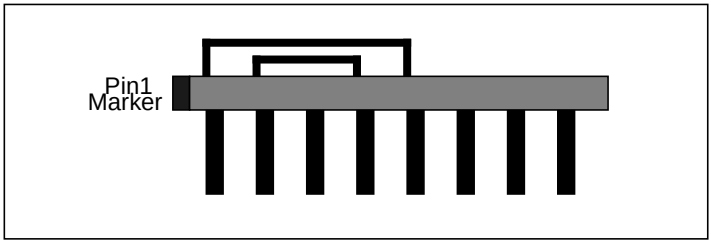


Figure 5. Pipe Jumper

Pipe jumpers are plugged in with the same orientation as TRAMs, i.e. with their pin1 adjoining the orientation mark on the motherboard. Figure 6 shows a situation in which a pipe jumper may be needed.

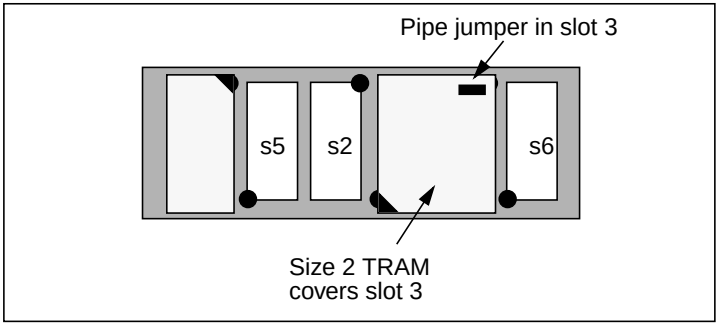


Figure 6. Using pipe jumpers

The diagram shows slot 3 being covered by a large TRAM. If it is required to continue the pipeline on beyond slot 3, in this case there is a TRAM in slot 4, then slot 3 will need to be jumpered.

2.2 Building a Computer from TRAMs

This section describes the setting up of a simple system consisting of a number of transputers on a single module motherboard. Building the system consists of three main stages:

1. Plug the TRAMs onto the motherboard,
2. Configure the motherboard reset structure for the software development system or application being used,

2.2.1 Physically building the System

Transputer modules and motherboards contain components which can be damaged by static electricity. It is therefore advisable to take some simple precautions before handling TRAMs and module motherboards:

- keep the TRAM plugged into its anti-static mat when it is not plugged onto a motherboard,
- keep module motherboards in their anti-static bags when they are not in use,
- before handling TRAMs or motherboards ground yourself by touching an earth (the metal enclosure of electrical equipment is always earthed),

- try to avoid touching the TRAM pins when plugging them in.

Plug the TRAMs into the motherboard slots as required by your application. When doing this note that:

- TRAMs should be oriented the correct way round to avoid permanent damage,
- if you are using the occam TDS, or the interactive debugger supplied with the Inmos C or occam toolsets, you will need to connect the subsystem on the TRAM that plugs into slot 0,
- TRAMs are shipped with spacers attached to their pins to raise them above any components on the motherboard. When there are no components on the motherboard these spacers can be removed if you wish to lower the height profile of the motherboard/TRAM combination. However please bear in mind that adequate airflow under the TRAM is required for cooling. See figure 7.

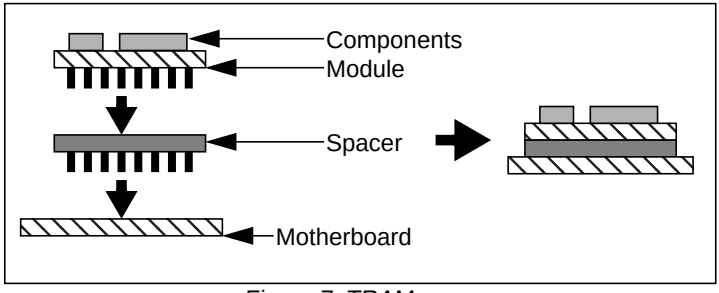


Figure 7. TRAM spacer

2.2.2 Configuring the System

The next stage is to set all the configuration options of the board. Normally, the only options which have to be considered are the control structure and the transputer network.

The following sections discuss the relevant options without reference to a particular motherboard. For details of the actual jumpers/switches to set refer to the hardware chapter in the product specific section of this document.

2.2.2.1 The Control Structure

In order to be able to program and use the TRAMs on the motherboard there is a mechanism for their control. The most important control signal sent to a TRAM is the reset signal. When a TRAM receives this signal, the transputer is reset to an initial state.

The transputer has to be in this state before a user program can be loaded onto it.

There are two basic control architectures that are commonly used:

- the host computer controls all of the processors in the system. This configuration is suitable for the Inmos toolsets and 3L scientific languages where the various tools in the programming environment (compilers, linkers etc.) are invoked from the host computer. Figure 8 shows what this looks like.

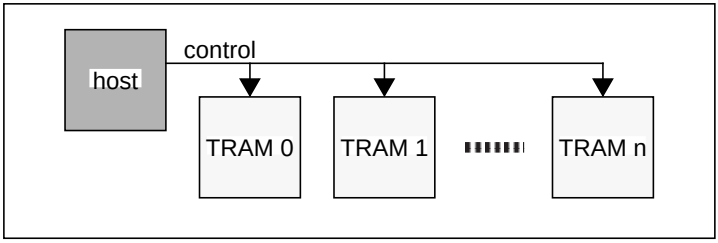


Figure 8. Control architecture for 3L/toolset

- the host computer controls only one transputer - the root or master transputer. This configuration is suitable for occam TDS users and also for Inmos toolset users who wish to use the interactive debugger, where a program (i.e. the TDS) runs on the master processor while other processors in the network are controlled from the master processor's subsystem port and are called a subnetwork. This enables TDS or the interactive debugger to boot/debug the other processors. Figure 9 shows the situation.

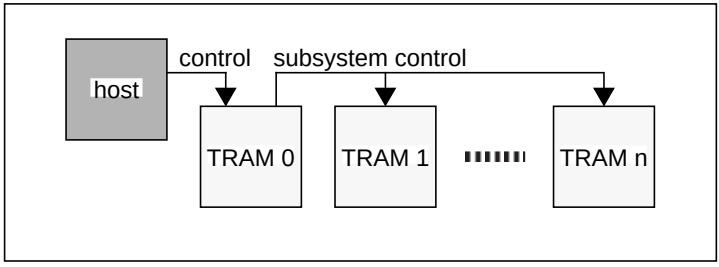


Figure 9. Control architecture for TDS

In order for the host computer to be able to control any transputers the motherboard or transputer host interface board must be set up correctly. See the relevant manual for your host adaptor/motherboard.

2.2.2.2 The Transputer Network

The only thing to be done here is to put in pipe jumpers where they are needed. Some simple rules for determining when pipe jumpers are needed follow:

1. If a large TRAM (greater than size 1) is in the middle of your intended pipeline then the large TRAM will need its unused slots jumpering.
2. If your motherboard is not fully populated and you wish to continue the pipeline onto another motherboard, then the unused slots on the motherboard will need jumpering.

2.3 A More Complex Example

This section shows how you could build up a TRAM network based on a number of motherboards. A common example of this is driving a number of slave TMB12 boards from a motherboard inside a PC. The TMB12 doesn't have a host computer interface and therefore has to be slaved to a master.

There are three aspects to connecting up several motherboards:

1. Sorting out the control system
2. connecting up the transputers
3. connecting up the configuration pipeline

2.3.1 Control Aspects

The control architecture of the single motherboard now has to be connected to other motherboards. Each motherboard has three control ports for this purpose. They are called *subsystem*, *up* and *down*.

The motherboards will normally be connected into a chain. Control is fed into the up port of a board. Normally control is propagated out of the down port. Hence, to connect up a chain the up port of one board is connected to the down port of the next board.

The only exception to this rule arises when the first motherboard in the system is considered. In this case it is desired to control a subsystem from the master processor and the TRAMs on the slave boards should be part of that subsystem. The subsystem port has been provided for just this situation.

Figure 10 shows a reset cable which is used to connect the control ports of different boards together.

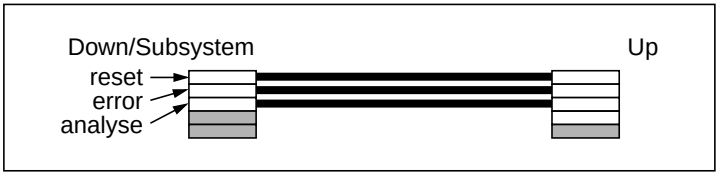


Figure 10. Reset Cable

Figure 11 shows the control connections for the simpler case of the 3L languages and figure 12 shows the connections for users of the occam TDS or Inmos toolsets.

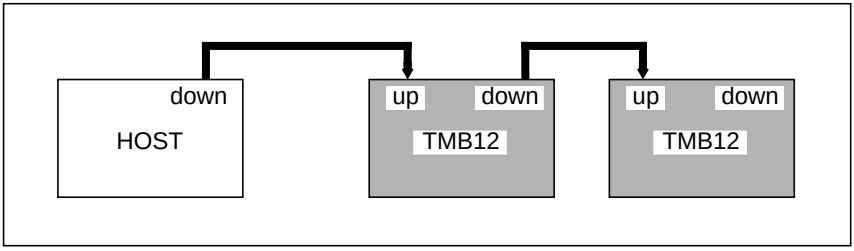


Figure 11. Control in multi-motherboard systems

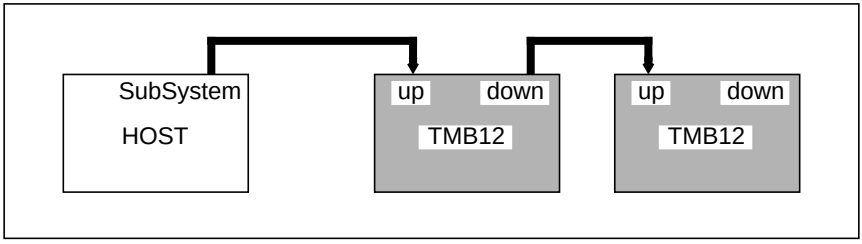


Figure 12. Control in multi-motherboard systems (TDS)

2.3.2 Topology Aspects

In order for the transputers to communicate their links must be connected together. In keeping with the pipeline connections made within a single motherboard, when several motherboards are wired up, all the transputers are connected into a single pipeline.

This is achieved by connecting the end of the pipeline of one motherboard to the beginning of the pipeline on the next motherboard.

The start of the pipeline (slot 0 link 1) is called *PipeHead*. The end of the pipeline (slot n link 2) is called *PipeTail*. Hence you must connect the PipeTail of one board to the PipeHead of the next board.

Because the PipeTail to PipeHead connection is actually a transputer link, the cable used to make this connection is a standard INMOS link cable. See figure 13.

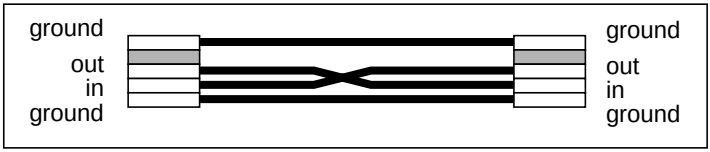


Figure 13. Standard link cable

Figures 14 & 15 illustrate the required connections.

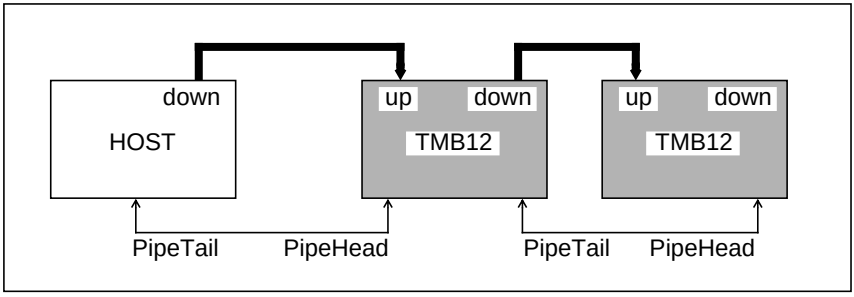


Figure 14. The default pipeline in multi motherboard systems

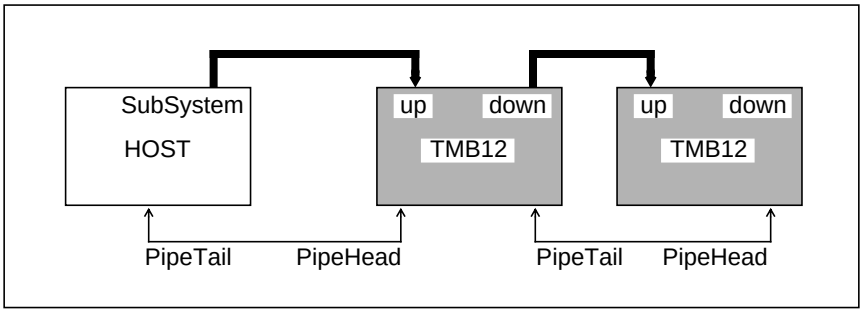


Figure 15. The default pipeline (TDS)

2.3.3 Network Configuration Aspects

A subject which has not been discussed fully in this chapter is that of network configuration. This is the process of connecting up *all* the transputer links to meet the network topology demanded by the application.

The previous section has dealt with the default pipeline, but what hasn't been mentioned yet is how the user can connect up TRAM link 0s & 3s.

In fact, the connection of these links is extremely flexible. On most motherboards there are electronic link switches which perform link0 - link3 connections. The link switches are controlled by transputer(s) on the motherboards called configuration transputers.

The configuration transputers from different motherboards are connected together into a pipeline called the configuration pipeline which is programmable. Transtech TRAM motherboards are shipped with a software system the NCS (Network Configuration Software) to program the link crossbars, Inmos boards are shipped with their own system the MMS (Module Motherboard Software).

Hence, the remaining job to be done is to connect up the *configuration pipeline*, a pipeline of T2 transputers that are used to setup the crossbar switches. Each motherboard has two pipeline connections for this purpose called *ConfigUp* and *ConfigDown*. Figures 16 & 17 show the details.

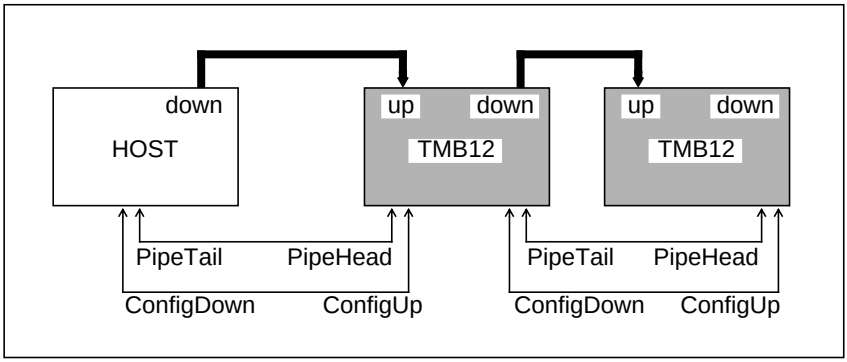


Figure 16. Configuration pipeline in multi motherboard systems

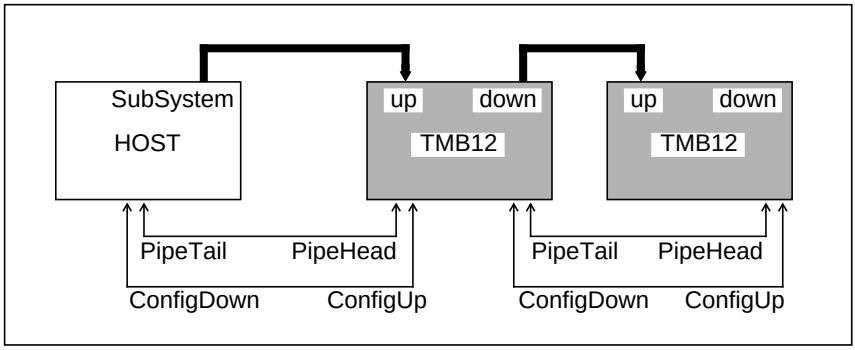


Figure 17. Configuration pipeline in multi motherboard systems (TDS)

The head of the configuration pipeline is connected to link 1 of TRAM slot 0 of the first motherboard.

Chapter 3 contains a full description of network configuration.

2.4 Summary

This chapter has given a beginners overview of Transputer Modules and module motherboards. Users should understand how to construct single motherboard and multi motherboard computing systems from TRAMs.

This chapter has given a statement of the actions to be performed in constructing TRAM based systems, often without explaining exactly

why those actions are necessary. This is in keeping with a beginners introduction. For an explanation of the underlying system, read Chapter 3 which gives details of the TRAM standard and can be regarded as containing a conceptual model of TRAM equipment.

Chapter 3
The TRAM Standard

This chapter forms a reference guide for TRAM equipment. It describes the philosophy behind the TRAM standard before describing Transputer Modules and Module motherboards. The chapter discusses transputer networks and hierarchical control as implemented on multiple motherboards.

The nature of reference dictates that the information be detailed. Inspite of this, all users should aim to understand most of the material in this chapter in order to gain the most out of using their TRAM equipment.

3.1 Introduction

The transputer is a general purpose computational element packaged as a single chip. It contains special dedicated communications circuitry which enables a complete computing engine to be constructed from more than one device. In this sense, the transputer is very much a building block of large parallel computers. In keeping with the building block philosophy of transputers the Transputer Module (TRAM) standard was evolved so that transputer based computing could be purchased in a way that allowed users to customize their computers to their particular application.

The standard allows parallel machines to be constructed with a mix of computational performance and memory tailored for a given application. The flexibility of the standard allows other functionality to be inserted into the parallel machine at the optimum place, for instance disk, graphics and other input/output capability.

To achieve this the standard first defines the Transputer Module. This is a small circuit board which will generally contain a transputer,

some memory and some application specific circuitry. The TRAM is in its own right a Dual-in-Line module which plugs into a purpose designed TRAM motherboard. The motherboard (the second part of the standard) provides power and link connectivity services to the TRAMs mounted on it. Transtech's range of motherboards support up to 16 TRAMs on a single motherboard. The motherboard itself is mounted inside a host computer. Transtech supply motherboards for host computers with a PC AT Bus (IBM AT and all clones) or a VME bus. as well as mother boards with no specific host interface but with links to other motherboards.

3.2 The Transputer Module

3.2.1 Overview

A TRAM is a self contained computing subsystem. Physically it is small and will contain either a transputer or some other device which connects via INMOS links. TRAMs:

- interface to each other via INMOS links
- have a standard pinout

The basic size of a TRAM is 1.05" by 3.66" overall (2.67cm by 9.3cm). This basic size is referred to as Size 1. A size 1 TRAM is big enough to contain a 30MHz T805 transputer with 4MBytes of Dynamic RAM. Often TRAMs are larger than Size 1, e.g. a size 3 TRAM measures 3.15" by 3.66" (8.01cm by 9.3cm), which is big enough for a 30Mhz T805 transputer and 16MBytes of Dynamic RAM.

The properties of the TRAM ensure that the standard is independent of:

- transputer type (e.g., T225, T425, T805)
- transputer speed (15 - 30MHz)
- peripheral function (i860 second processor, graphics output, frame grabber, SCSI interface, etc.)
- memory size (32KBytes to 24MBytes)

3.2.2 Functional Description

3.2.2.1 Pinout of Size 1 Module

Figure 18 shows the pinout of a size 1 TRAM.

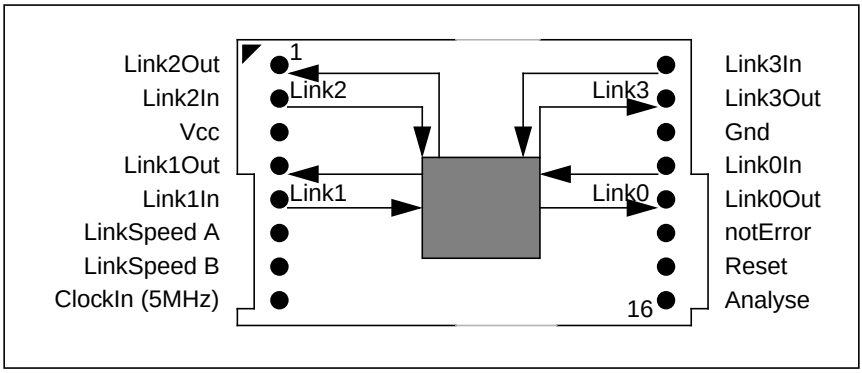


Figure 18. Pinout of Size 1 TRAM (not to scale)

The TRAM brings out all four of the transputer's links for connection to other TRAMs. *LinkSpeedA* and *LinkSpeedB* control the operating speed of the transputer's links. Asserting both of these pins high causes the links to operate at 20Mbits/sec, deasserting both these pins low causes the links to operate at 10Mbits/sec. Other combinations of states of these two signals are illegal.

The transputer's *error*, *analyse* and *reset* signals are brought out from the TRAM. The reset and analyse signals are taken directly to the transputer and allow the transputer to be programmed and debugged. The error output from the transputer is buffered by a transistor on the TRAM. This has the effect of inverting the signal (it becomes *notError*) and also makes the signal an open collector output. This allows wire-ORing of the error signals allowing any TRAM to signal error back to the host computer.

The other pins of the TRAM provide power to the circuitry and a clock for the transputer. Note that the input clock signal for the transputer is multiplied by an on chip Phase Locked Loop to give the final processor clock (typically 20 to 35 MHz).

3.2.2.2 Subsystem Signals

In order to be able to manage a network of transputers the error, analyse and reset signals are taken to the host computer. However, it is often convenient to be able to drive these signals from a *master* transputer. These signals are collectively known as the *subsystem* signals. Most TRAMs have a subsystem port which allows them to

control a (sub)network of transputers. The subsystem port is shown in figure 19.

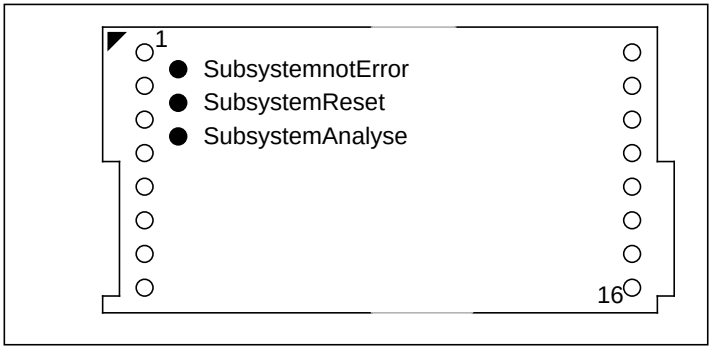


Figure 19. Subsystem Pins

The subsystem port consists of three zero profile sockets on the underside of the TRAM. In order to connect the subsystem to the motherboard a special double ended subsystem port connector is used. This is shown in figure 2.

This arrangement reflects the optionality of subsystem pins on the TRAM and ensures that TRAMs fitted with subsystem pins are still plug compatible with TRAMs not fitted with subsystem pins.

Details of how to program the subsystem port of a TRAM are contained in the next section.

3.2.3 Electrical Description

The basic circuitry of a Transputer Module is shown in figure 20.

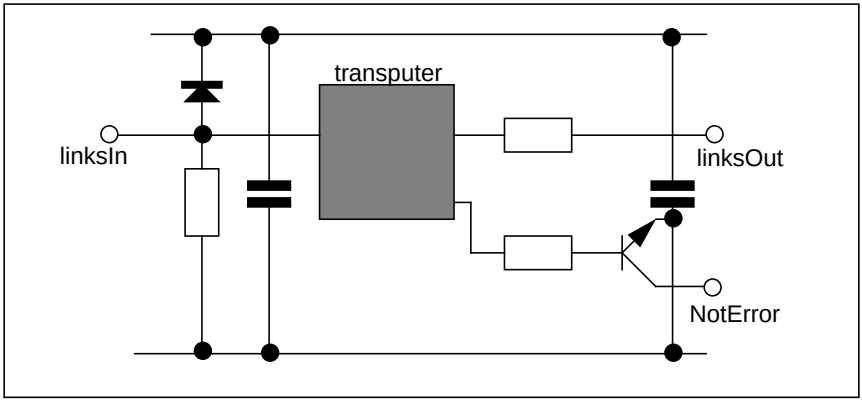


Figure 20. Basic TRAM circuit

An explanation of the features of this circuit follows:

- Link inputs are protected from positive Electrostatic Discharge by the inclusion of a signal diode taken to the positive supply rail. To prevent unconnected link inputs floating a pull down resistor is also included.
- Link outputs have a line resistance to match the output impedance to that of the transmission line to which it is connected.
- The *NotError* output is an inverted, buffered open collector version of the transputer's error flag.

Other connections require no special circuitry

3.3 The Transputer Module Motherboard

3.3.1 Overview

The module motherboard allows access to Transputer Modules from a variety of host machines. The motherboard can usually be divided into two distinct parts: the host specific interface part and the generic TRAM part (although some motherboards do not have a host specific interface).

The design goals of the motherboard standard were that the user should be able to:

- build computing systems consisting of any mix of TRAMs
- configure the transputers into any topology
- chain a number of motherboards together
- run and test applications on transputers without first having to configure the links

The architecture that was evolved to meet these requirements consists of the following important features:

- the modules in a network are connected in a pipeline using two links from each module
- the remaining links from each module are available for user configuration either by direct wiring (via edge connectors) or via a programmable link switch (IMS C004). When using C004 switches:
  - a number of links are taken from the C004 to edge connectors
  - each C004 is controlled by a T2 transputer
  - the T2 transputers are connected together in a separate pipeline
- the first module in the pipeline on a given motherboard can control a subsystem of other modules that may reside on the same motherboard, another motherboard or be distributed across a number of boards (the first module requires a subsystem port)
- an interface may be provided to allow non-transputer based host computers to control and communicate with the TRAMs on the motherboard.

The remainder of this section discusses the above features in greater detail.

3.3.2 Link Configuration

3.3.2.1 Pipeline

Transputer Modules are plugged into module slots on the motherboard. The number of slots on a motherboard depends only on the size of the motherboard. The slots are numbered from zero.

The modules on the motherboard are connected together into a pipeline, as shown in figure 4.

Assuming for the moment that all TRAMs are size 1, then link2 of the module in slot0 is connected to link1 of the module in slot1 (slots are not necessarily connected in number order, see the product specific

section of the manual for the order in which they are connected), and so on for the other slots on the motherboard. Link1 of module 0 (called *PipeHead*) and link2 of module n (called *PipeTail*) are in general brought out to the edge connector to allow pipelines to be constructed across several motherboards (figure 21).

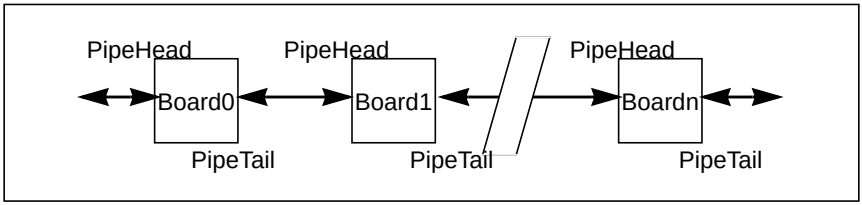


Figure 21. Module pipeline split across several motherboards.

Not all applications will use all of the slots on the motherboard. Some applications will require to use TRAMs which are larger than size1 (but only use one set of 16 pins for interfacing to the motherboard). In both of these cases the pipeline will become broken. To avoid this a special pipeline jumper is supplied which bridges the break (figure 5).

The pipe jumper plugs into unused slots (either on the motherboard or on a TRAM) and connects link1 of that slot to link2 of that slot.

3.3.2.2 Programmable Link Configuration

Some motherboards have a number of C004 link crossbar switches mounted on them. These are devices that allow the topology of the interconnections between transputers to be set electronically under software control.

The links 0 and 3 of each slot on the motherboard are taken to the switches for programmable link connection. The degree of interconnectivity achievable depends on the number of slots and the number of switches on the board. For example, on the TMB12 16 slot motherboard, two switches provide for 64 link connections.

3.3.2.3 The Configuration Pipeline

Each link switch is controlled by a 16 bit T2 transputer. Each T2 can control up to two link switches via its links 0 and 3 (figure 22).

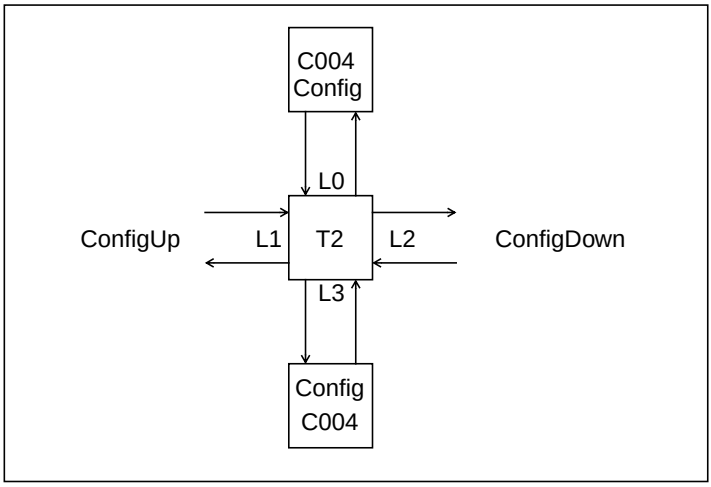


Figure 22. Control of the Link Configuration System

The other links are used to construct a pipeline of configuration transputers. This pipeline may extend across a number of boards to allow configuration to extend throughout a transputer system. Connection to other boards is achieved by edge connections *ConfigUp* and *ConfigDown* (figure 23)

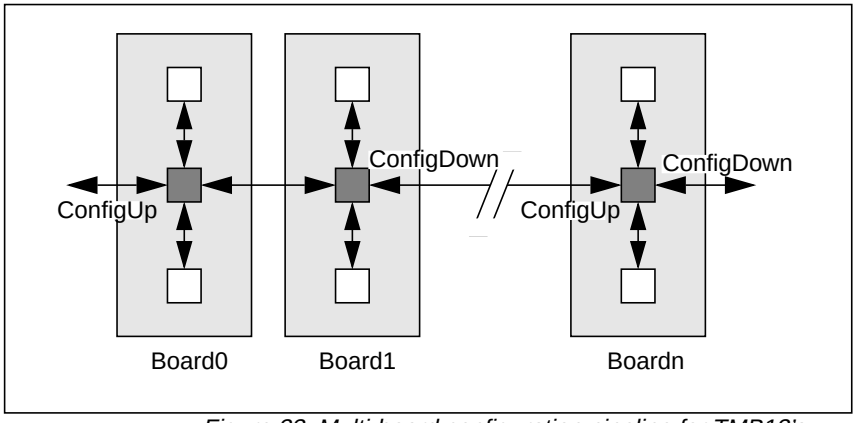


Figure 23. Multi-board configuration pipeline for TMB12's

Usually the link switch configuration data will originate from the a module on the first motherboard in the system. Hence one of the links of that module must be connected to the first T2 of the

configuration pipeline. Figure 24 shows the standard way of achieving this: Link 1 of the module in site0 is taken via a link patch to the T2. Note that this effectively terminates both *PipeHead* and *ConfigUp*.

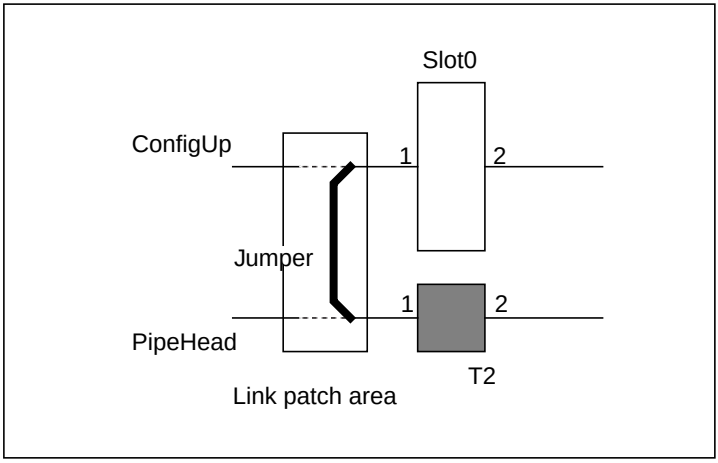


Figure 24. Controlling the configuration pipeline

3.3.2.4 Software for Link Configuration

To configure their network to a desired topology the user can program the T2 pipeline directly or make use of the software system or wiring files supplied with the board.

Note: Transtech module motherboards are currently supplied with the NCS (network Configuration Software) for electronic link configuration.

3.3.3 System Control

The design requirements of the motherboard, part of the TRAM standard, stated that a hierarchical control structure be provided to control networks of transputers. In practice this allows networks to be constructed out of subnetworks. An example might be where each TRAM in a *master* network controls its own subnetwork of transputers.

In order to control a transputer, only three signals are needed: a signal to reset the transputer, a signal to analyse (statically debug) the transputer and a signal from the transputer to indicate that an error has occurred. These signals are collectively called the subsystem signals.

The subsystem signals are driven by a master. The master controls all downstream processors connected to these subsystem signals (i.e. a subnetwork). Normally there are two types of master in a transputer network: the host computer and modules fitted with subsystem ports.

3.3.3.1 Source of Control

For control purposes the modules on a motherboard are divided into two groups: *module0* and *modules1 to n*. The configuration processor on a motherboard (T2 - if there is one) is included into the latter of the above two groups. Note that the error signal of the T2 is left unconnected.

Within a motherboard there are options as to what source controls the TRAMs on that motherboard:

- Module0 can be controlled from either the host computer or from an external source,
- Modules1 to n can be controlled from either a host computer, a module0 TRAM fitted with subsystem or an external source.

In general the external source will in fact be another motherboard. The external source arrives on the motherboard at the *up* connector.

Hence module0 can be controlled either from the *up* port or from the host interface (if there is one). Modules1 to n are controlled either from the up port, from the host computer interface or from module0's subsystem (figure 25).

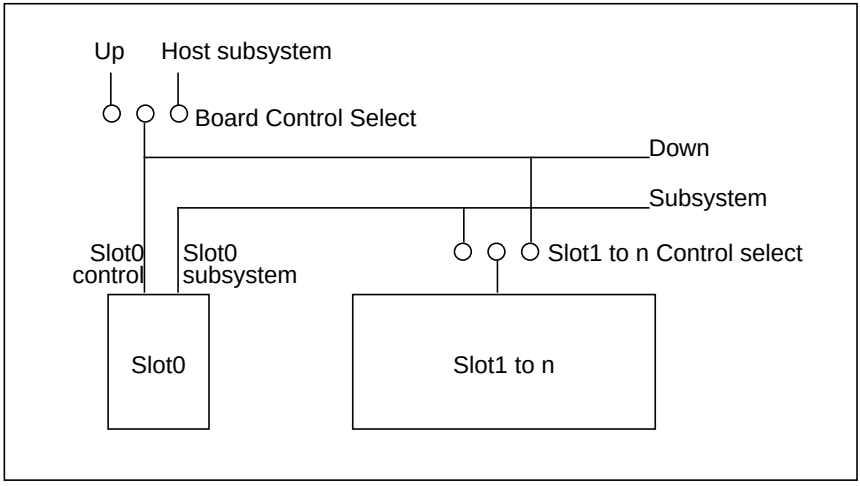


Figure 25. Source of control

The two selection options (board control select & slot1 to n control select) are altered by means of jumpers on the motherboard.

From figure 25 it can be seen that a subsystem of transputers controlled by a master TRAM consists of modules1 to n on this motherboard plus any modules on slaved motherboards. A subsystem of transputers controlled by a host computer consist of all the TRAMs on this motherboard plus any modules on slaved motherboards.

3.3.3.2 Up, Down & Subsystem

Large networks are built up by connecting motherboards together. For this purpose, each motherboard has three control ports:

- *up* (control in), source of external control,
- *down* (control out), echo of the up port,
- *subsystem* (control out), connected to module0's subsystem port.

Figure 26 shows the control ports.

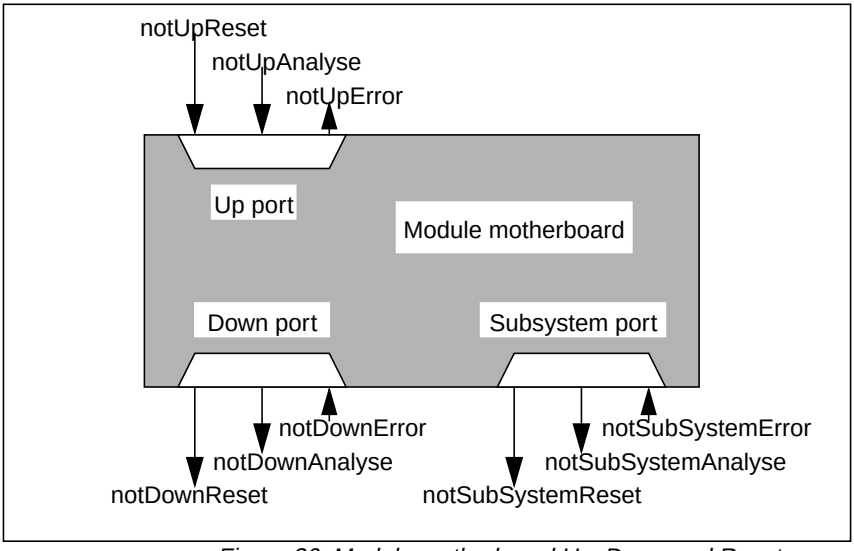


Figure 26. Module motherboard Up, Down and Reset

3.3.3.3 Multi-board Control

This section shows how the control ports are connected to map the subnetwork model of transputer networks onto a number of module motherboards.

The simplest relation between two motherboards is direct connection.Motherboards can be chained together by connecting the down port of one board to the up port of the next. In this case all the boards in the chain are controlled from the same source, i.e. whatever it is that controls the first motherboard in the chain.

The other type of relation between two motherboards is that of master-slave. This allows the module0 of the first motherboard to control a subnetwork of TRAMs, some of which may be on the second motherboard. This is achieved if the subsystem port of the first motherboard is connected to the up port of the second motherboard.

Figure 27 shows a master network consisting of all the of processors on *boarda*, as well as the first processor on each of *boardb* and *boarde*. Two of the processors in this master network control their own subnetworks. The first subnetwork consists of the processor on modules1 to n of *boardb* and all the processors on *boardc/d*. The second subnetwork consists of the processors on modules1 to n of *boarde* and all of the modules on *boardf/g*.

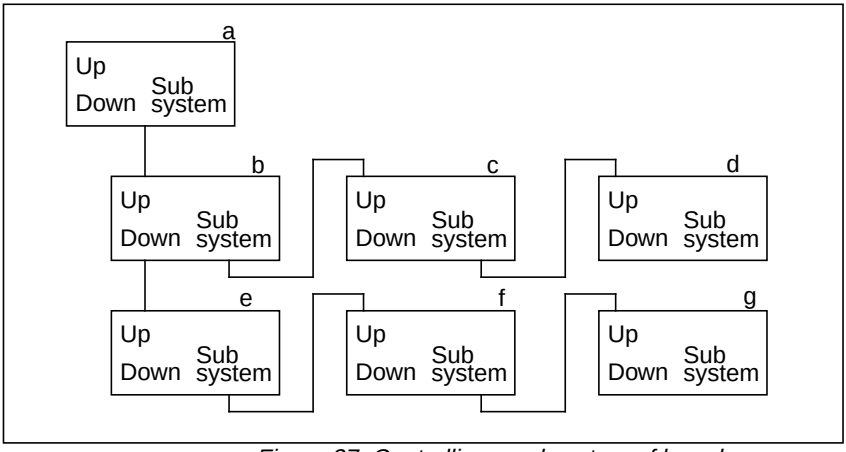


Figure 27. Controlling a subsystem of boards

The notReset and notAnalyse signals flow from the subsystem of a master board to the up port of a slave board. From there they are connected directly to the down port of the slave board. Hence reset and analyse are connected to all boards in a chain allowing the master to control all processors without hindrance. E.g. boardc can be reset/analysed by boarda without interference by boardb in figure 27, above.

The notReset and notAnalyse arriving at the slave's up port are also sent to the slave's subsystem port, but in this case the signals are logically OR-ed with the analyse and reset signals of the boards's subsystem. This means that a board may be reset/analysed by its own master OR by the master of that master. E.g. *Boardd* can be controlled by either *boardb* or *boarda* in figure 27, above.

The notError signal is treated similarly, but flows in the opposite direction. The error from a module on a motherboard is passed to the up port of that motherboard. It is logically OR-ed with any notError signal arriving at that boards down port. This allows error signals to propagate back to the master - it also means that ANY processor in a subsystem can send an error to the master. E.g. processors on *boardc/d* can send error back to their master (*boardb*) in figure 27, above.

The notError signal arriving at a subsystem port is not propagated to the up port, but is handled by that board. It is assumed that a subsystem master can handle errors in its own subsystem.

3.3.3.4 Subsystem Registers

When the source of control is a transputer module in site 0 of some motherboard then the transputer on that module has control of the subsystem via its subsystem pins. These pins are driven by subsystem registers on the TRAM which are in turn mapped into the transputer's memory map.

For 32 bit transputers the memory mapping is as follows:

Register	Mode	Hardware byte address
SubsystemResetLatch	Write Only	#00000000
SubsystemAnalyseLatch	Write Only	#00000004
SubsystemError	Read Only	#00000000

Table 1: memory-mapped registers

The operation of the registers is as follows:

- Writing a 1 into bit 0 of SubsystemResetLatch asserts *subsystemReset*;
- Writing a 0 into bit 0 of SubsystemResetLatch deasserts *subsystemReset*;

- Writing a 1 into bit 0 of SubsystemAnalyseLatch asserts *subsystemAnalyse*;
- Writing a 0 into bit 0 of SubsystemAnalyseLatch deasserts *subsystemAnalyse*;
- Reading a 1 from bit 0 of SubsystemErrorLatch indicates that *subsystemError* is TRUE;
- Reading a 0 from bit 0 of SubsystemErrorLatch indicates that *subsystemError* is FALSE;

The subsystem is reset/analysed under control of the transputer on the TRAM. However, the subsystem must be reset when the TRAM is reset. As already described the following combinational logic is used to propagate the subsystem signals:

- SubsystemReset = UpReset OR SubsystemResetLatch
- SubsystemAnalyse = UpAnalyse OR SubsystemAnalyseLatch
- SubsystemError does NOT propagate

Because of filtering on the motherboard it is recommended that when resetting/analysing subsystems the corresponding signal is held asserted for at least 5 ms. In the case of reset, the subsystem should be left for a further 5 ms before attempting to boot it.

3.4 Host Computer Interface

Most motherboards have an interface to a host computer. The host supplies file services and terminal IO to application programs running on the transputer network. Clearly the structure of the host interface is not generic - the remainder of this section discusses the standard PC interface for PC AT machines and clones. This interface is commonly referred to as a “B004” interface.

3.4.1 Host IO Space

In PC machines interface hardware is mapped into a special address space called the IO space. The driver program running on the PC uses this address space to talk to the add-in card.

The module motherboard employs a block of addresses within the IO space.It is possible to locate the base of this block at one of a number of places in this address space so as to prevent conflicts with other add-in cards. By default the base address is150 hex.

A summary of the register addresses used within the block used is given in table 2. These are all explained in greater detail in the following sections.

IO Address	Register
boardbase + #00	InputDataRegister
boardbase + #01	OutputDataRegister
boardbase + #02	InputStatusRegister
boardbase + #03	OutputStatusRegister
boardbase + #10	Reset register (write only)
boardbase + #11	Analyse register (write only)
boardbase + #10	Error register (read only)
boardbase + #12	DMA request register
boardbase + #13	Interrupt control register

Table 2: IO Registers

3.4.2 Link Interface

The link interface employs an IMS C012 link adapter chip. This device converts between an 8 bit bidirectional port on one side and a transputer serial link on the other, see figure 28.

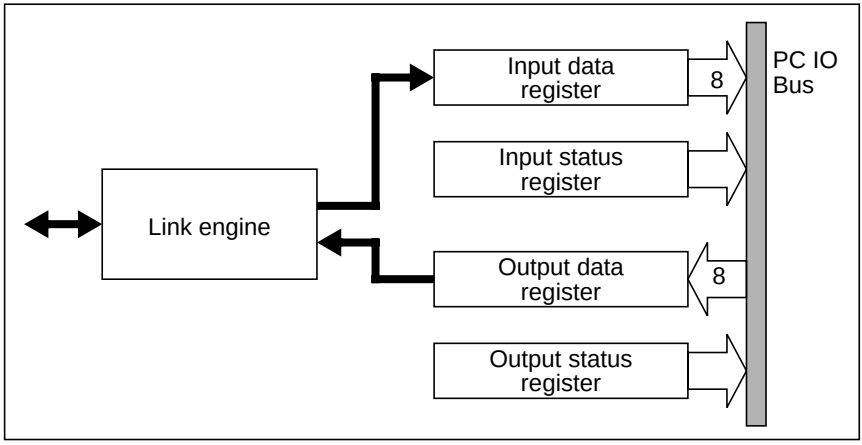


Figure 28. Logical structure of the C012 link adapter

The four registers of the C012 are mapped into the first four addresses in the board's IO space.

Data arriving at the C012 from the serial link is latched into the InputDataRegister. A flag in the InputStatusRegister is set to indicate that the contents of the data register are valid. Current servers poll this flag in software to see if any data has arrived. When the data register is read the flag in the status register is cleared.

Output to the link from the PC follows a similar pattern. An *in use* flag in the OutputStatusRegister indicates whether the OutputDataRegister can be written to. Current servers poll the in use flag in software, sending a byte to the data register as soon as the link becomes free (i.e. the last byte has been sent).

The C012 link connects to link 0 slot0, and on some motherboards it can be connected to the edge connector, allowing the board to be used as a link adaptor to communicate with external transputer systems.

3.4.3 System Control Interface

The next three registers in the IO space are the system control registers. In terms of function they are identical to the subsystem latches on some TRAMs. Only bit 0 of these registers is used:

Reset register:	setting bit 0 to ‘1’ asserts Reset to the motherboard, clearing bit 0 to ‘0’ deasserts Reset.
Analyse register:	Setting bit 0 to ‘1’ asserts analyse to the motherboard, clearing bit 0 to ‘0’ deasserts analyse.
Error register:	Reading a ‘1’ from bit 0 indicates an Error on the motherboard, Reading a ‘0’ from bit 0 indicates no Error.

3.4.4 Interrupts and DMA

Two other registers are provided on Transtech’s range of motherboards are not part of the basic “B004” interface. The two other registers provide support for allowing DMA access to the link hardware. This overcomes the inherent inefficiency of software polling for every single byte read or written to the link adapter, however these features are not often used. No standard software from Transtech uses these features.

The PC contains a DMA controller chip (the 8237) which is employed for these high speed transfers. A DMA transfer to the to the motherboard is initiated by writing a ‘0’ to the DMA request register. A DMA transfer from the motherboard is initiated by writing a ‘1’ to the DMA request register.

To discover when the DMA has completed it is possible to poll the status of the DMA controller chip. However, a better method is to generate an interrupt to the PC. For this purpose the last register in the motherboard IO space is the interrupt control register. This register is a mask register allowing the PC to be interrupted on any combination of four events. Figure 29 shows the structure of this register.

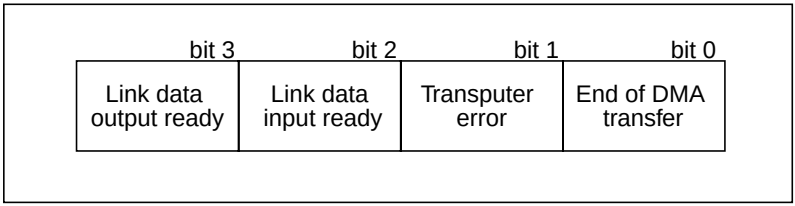


Figure 29. Interrupt control mask register

To allow an interrupt on any of these sources write a ‘1’ to the relevant bit position. To inhibit an interrupt on any of these sources write a ‘0’ to the relevant bit position.

Once the PC has been interrupted the interrupt handler must determine the source of the interrupt. This can be done by reading the Input/OutputStatusRegisters, the Error register and the status registers on the DMA controller chip.

3.4.5 DMA & Interrupt Channels

There is a small degree of flexibility in terms of the interrupt channel and DMA channel that can be used within the PC: the DMA channel can be set to 1 or 2 and the interrupt channel can be set to 3 or 6.

DMA channel 1 is often used by ethernet or other networking cards. DMA channel 2 is used by the PC to communicate with the floppy disk controller. It is possible to share DMA channel 2 with the floppy disk controller by careful programming of the floppy disk controller enable flag located at IO address space #3F2. Whenever a floppy disk access is about to take place the BIOS always enables location #3F2. Hence to use DMA channel 2, the software must disable the controller (by writing a zero to the enable flag) and enable its own drivers. At the end of the DMA transfer the software must disable its own drivers.

The floppy disk subsystem also uses interrupt channel 6. The floppy disk uses this interrupt channel to notify the PC that it is inactive and as such the PC can switch off the “in use” lamp on the front panel. The interrupt is generated about two seconds after the last disk access. Any software using this interrupt channel, should be able to handle such interrupts (or be able to guarantee that such interrupts cannot arrive).

Chapter 4

The TMB03 Motherboard

This chapter gives a detailed hardware description of the TMB03. The various features of the board are described and some examples of configuration are given.

The TMB03 is a small PC hosted TRAM motherboard, with space to plug in up to five Transputer Modules. It has a dummy site allowing a size six TRAM to be fitted

It is possible to use the TMB03 without TRAMs as a driver board (PC-to-link adapter) for external transputer systems via the master link on the edge connector.

Figure 30 shows the layout of this board for reference.

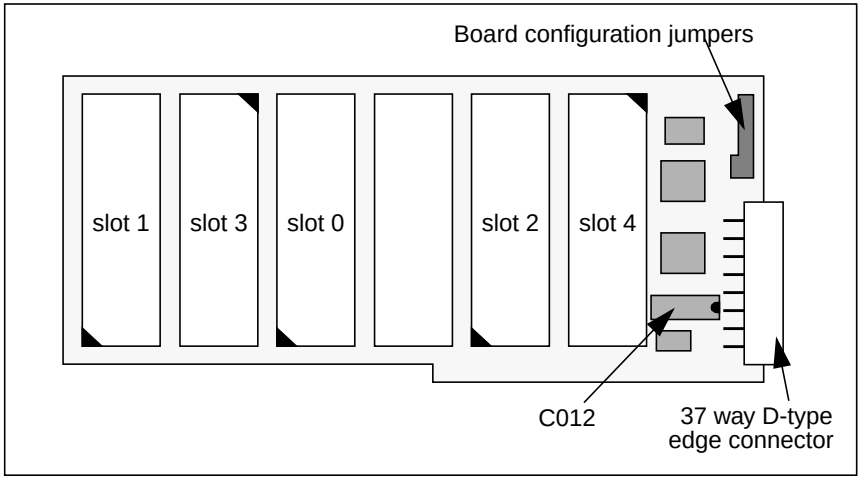


Figure 30. Layout of the TMB03

4.1 Board Configuration Jumpers

This section describes the various board configuration settings and TMB03 specific functionality.

Most of the configuration options are adjusted by the means of a jumper block at the top right corner of the board which is shown in figure 31 for reference.

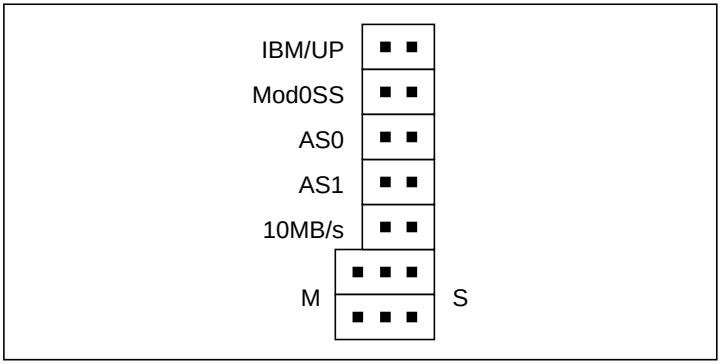


Figure 31. Board configuration jumpers

The jumpers are briefly described below, and in detail in the following sections.

IBM/UP	The source of control for the TRAM in slot0 can be from either the host PC or from the edge connector (UP).
Mod0SS	The TRAMs in slots 1, 2, 3 & 4 can be controlled either from the subsystem of the TRAM in slot 0 or from the source which controls slot 0.
AS0/AS1	The board's IO address can be set to #150, #200 or #300.
10MB/s	The transputer link speeds can be set to 10MHz or 20MHz.
M & S	Allow the board to be used as a *Master* or *Slave* board.

4.1.1 Control Configuration

The board's control configuration jumpers, IBM/UP and Mod0SS, allow the source of control for the module in slot 0 and the modules in slots 1, 2, 3 & 4 to be determined. The control consist of the TRAM signals reset, error and analyse.

Jumper	in/out	Description
IBM/UP	out	Slot 0 is controlled by the PC (default)
	in	Slot 0 is controlled by the edge connector up port

Table 3: Slot 0 control selection

Jumper	in/out	Description
Mod0SS	out	Slots 1 to 4 are controlled from the Slot 0's subsystem port
	in	Slots 1 to 4 are controlled from the same source as Slot 0 (default)

Table 4: Slots 1 to 9 control selection

4.1.2 Board IO Address

Configuration of the board IO base address is achieved via jumpers AS0, AS1, as follows:

AS0	AS1	Description
in	in	Base address 150 hex (default)
in	out	Base address 200 hex
out	in	Base address 300 hex

Table 5: Base address select jumper

4.1.3 Link Speed Configuration

The link speed select jumper controls the speed of all transputer links on the board.

Jumper	in/out	Description
10MB/s	out	20 MBits/s (default)
	in	10 MBits/s

Table 6: Link speed selection jumper

4.1.4 Master and Slave Configuration

The TMB03 can be used either as a master board or a slave board. In master configuration the TMB03 behaves as an ordinary TRAM motherboard with module0 link0 connected to the PC via C012 interface circuitry. It also acts as a PC link interface board to connect to external transputers, with no TRAMs on the TMB03.

In slave configuration, module0 link0 is connected to the edge connector providing a further mechanism to chain motherboards together. (The normal PipeTail - PipeHead connection can be made regardless of the master/slave configuration.)

By altering the M/S jumpers it is possible to connect module0 link0 to the edge connector as an external link, *as long as the C012 is removed* from the board. The C012 is socketed to allow easy removal.

Desired Configuration	Required setup
Connect TRAM slot 0 link 0 to the host link (default)	M/S jumpers set to M, nothing connected to master edge link, TRAM in slot 0, C012 fitted
Connect host link to external transputer	M/S jumpers set to M, link cable connected to master edge link, nothing in slot 0, C012 fitted
Connect external transputer to slot 0 link 0	M/S jumpers set to S, link cable connected to master edge link, TRAM in slot 0, C012 removed

Table 7: Host link, Slot 0 link 0, and Master edge link options

Figure 32 shows the effect of these jumpers and figure 33 shows their use.

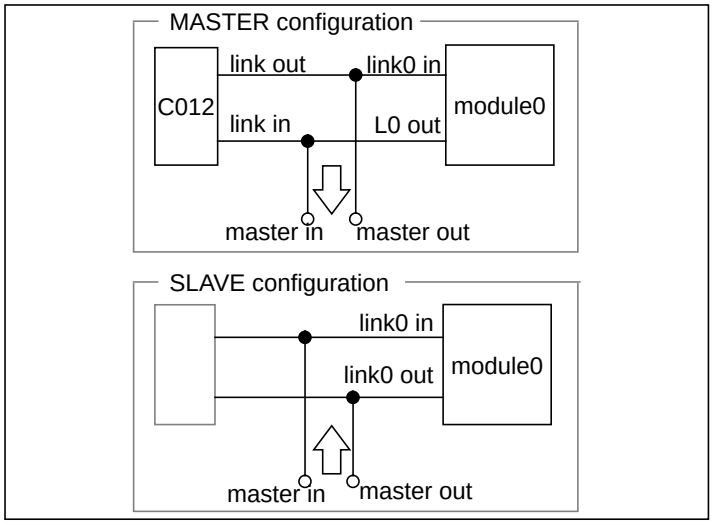


Figure 32. Board Master & Slave Configuration

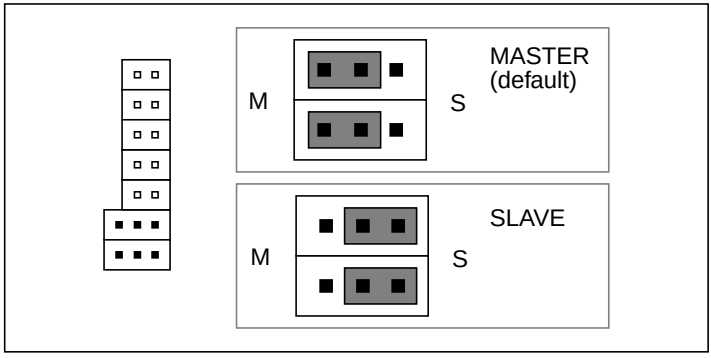


Figure 33. Master & Slave Configuration options

IMPORTANT: note that to use the board configured as a slave according to figure 32, the C012 chip must be removed. See figure 30 on page 35 for the location of the C012.

Permanent damage may result if you do not remove the C012 when configuring the TMB03 as a slave, or if you use the master

edge link when a TRAM is fitted in slot 0 and the board is configured as a master.

4.1.5 IRQ & DMA Selection

The default connections are to use DMA channel 1 and interrupt line 3. These defaults are made by actual copper tracks on the surface of the printed circuit board. To alter the defaults, cut the existing tracks and hook up the solder pads as required. It is possible to configure the board to use DMA channels 1, 2 or 3 and to use interrupts 3, 4, 5, 6, 7 or 9. To disable the Interrupts and DMA, just cut the tracks.

Your warrantee will not be affected by cutting the tracks or soldering between the solder pads.

Figure 34 shows the area and details the default connections.

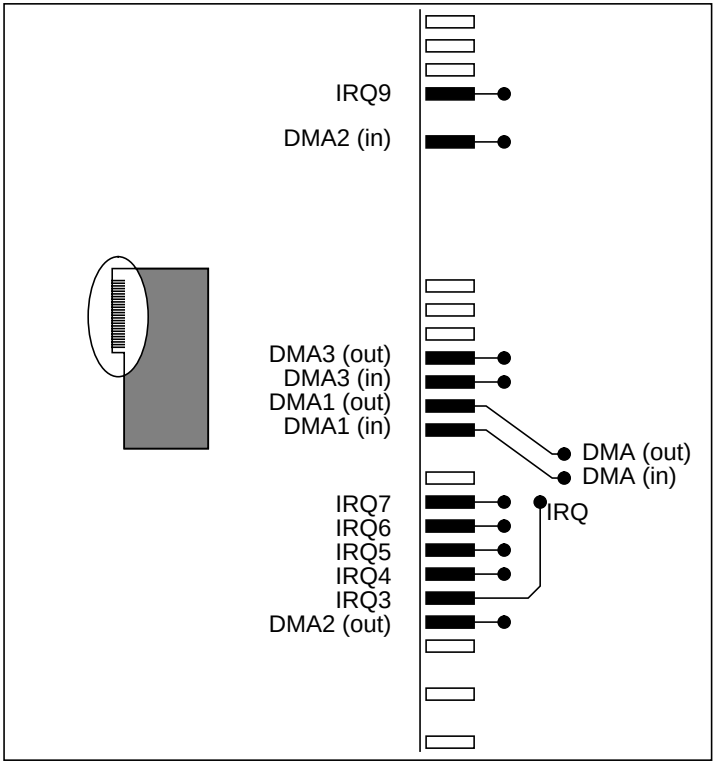


Figure 34. Selection pins for Interrupt/DMA channel.

4.2 The Edge Connector

On the right hand edge of the TMB03 is a 37-way D-type edge connector. On the TMB03 the edge connector is used for two purposes:

- connecting to other motherboards control signals (via up, down and subsystem)
- connecting transputer links to construct the desired network topology.

Figure 35 shows the pinout of the edge connector.

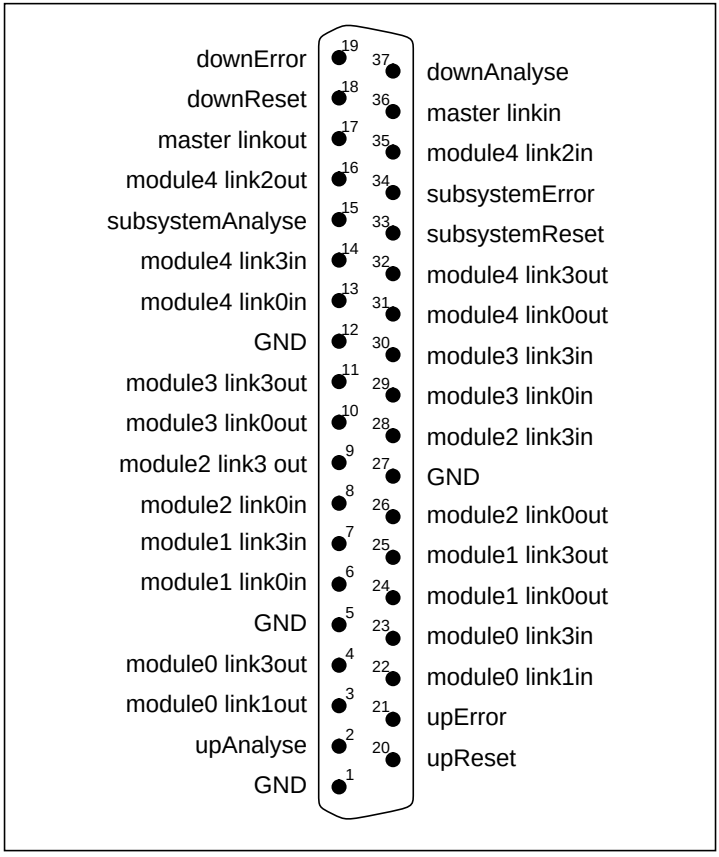


Figure 35. D-type pinout

Included in the accompanying cable pack is a mini-backplane board (“hedgehog”) which plugs into the edge connector and brings out the various links and ports onto standard connectors which accept link cables and reset cables. The pinout of this connector is shown in figure 36.

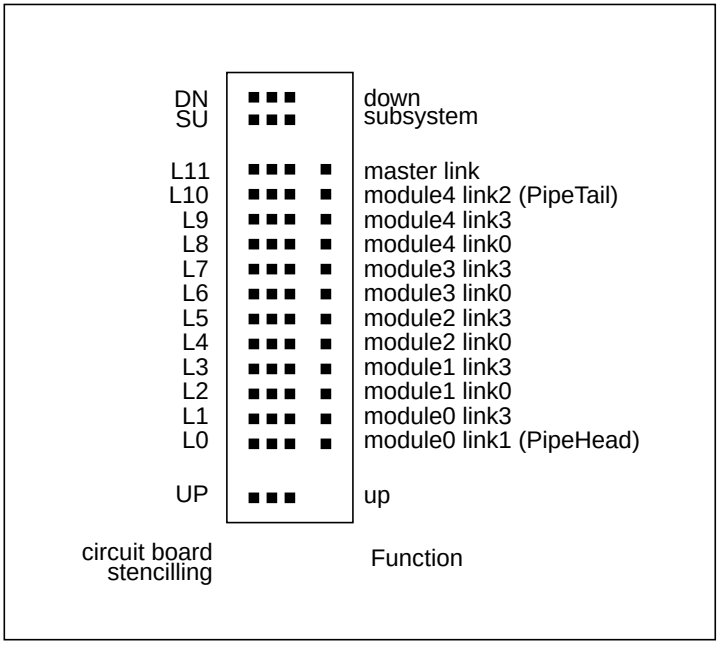


Figure 36. Connections on break out board

Essentially, the connector brings out the three control ports, the link0s and link3s of all the transputers (for network configuration) and the ends of the default pipeline. This is summarized for reference purposes in figure 37.

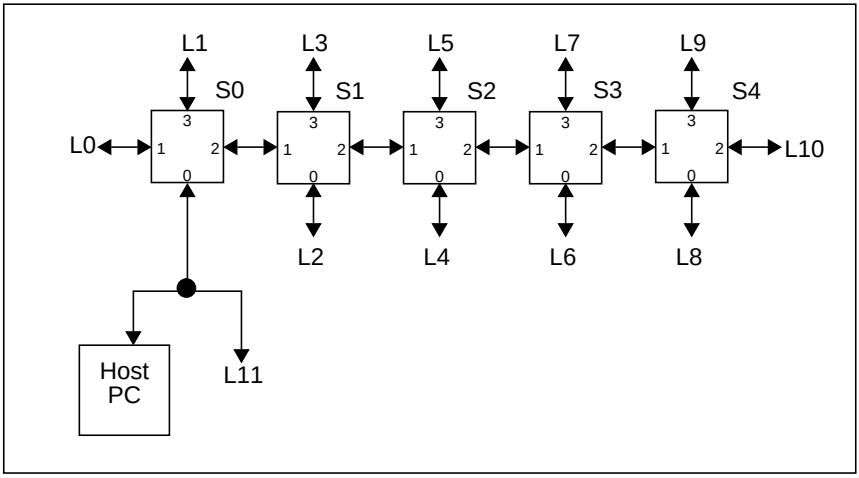


Figure 37. Summary of Network interconnect

4.2.1 Use of the master link

The TMB03's master link provides extra functionality over the normal PC motherboards. However, care must be exercised in its use so as not to damage the hardware. Below is a list of all of the valid configurations:

- board set to master mode, master link not connected to anything & one or more TRAMs plugged onto the board - normal operation.
- board set to master mode, master link connected to an external rack of transputer equipment & no TRAM in slot0 of the board - use as a link adapter card.
- board set to slave mode, master link optionally connected to external transputer equipment, one or more TRAMs plugged onto the board & the C012 removed - slave use.

4.3 Example

The default configuration of the TMB03 is suitable for stand-alone operation with software such as the Inmos Toolsets and 3L. The

example given here shows to configure a system consisting of two TMB03s in the same PC. In this case:

- the two boards must have different addresses
- one board must be slaved to the other
- the default pipeline needs to be connected

Set the jumpers as follows:

Jumper	First board	Second board
IBM/UP	out	in
Mod0SS	in	in
AS0	in	out
AS1	in	in
10MB/s	out	out
M/S	M	M

Table 8: Jumper settings for two TMB03s

Fit TRAMs in all sites (using pipe-jumpers on unused slots and on the inactive slots of TRAMs larger than size 1). No subsystem pins are needed.

Fit the boards in your PC, and using the “hedgehog” breakout boards, make the following connections:

Cable	First board	Second board
Reset Cable	Down	Up
Link Cable	Pipetail (L10)	Pipehead (L0)

Table 9: Connections between TMB03s

Power on the PC and run `check`.

When connecting the TMB03 to motherboards which include electronic link switching, then the TMB03’s PipeHead could be used as ConfigDown (i.e. it may be connected to ConfigUp of the configurable motherboard).

Chapter 5

The TMB04 Motherboard

The TMB04 is a transputer board for PCs with a single T4 or T8 transputer, up to 16 MBytes of memory in SIMMs, a B004-compatible PC link interface, and four TRAM sites.

The board has been designed to support transputer clock speeds from 17.5 to 35 MHz, and the memory access time can be set to 3, 4, 5 or 6 cycles. This allows the matching of many different speeds of memory device to different speeds of transputer.

Figure 38 shows the location of the various jumpers and switches on the board.

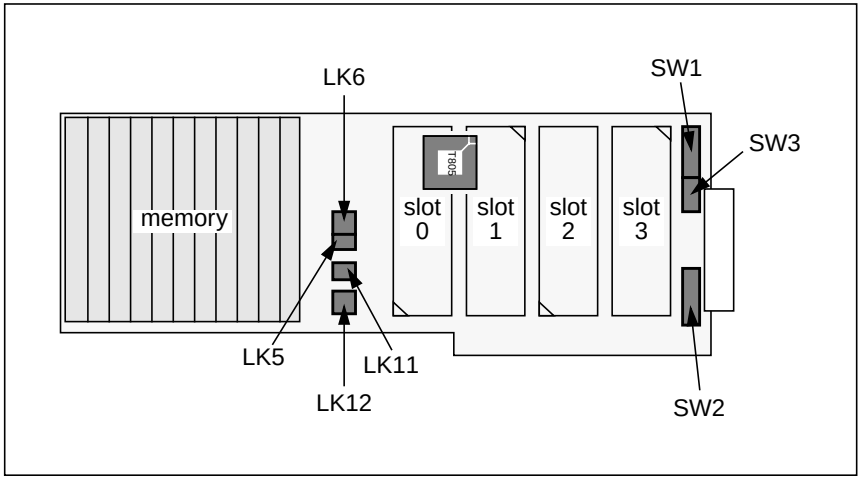


Figure 38. Layout of the TMB04

5.1 Fitting TRAMs

Up to 4 size one TRAMs may be fitted to the TMB04, and the general considerations discussed in section 2.2 apply. However note that:

- the TRAM sites are arranged sequentially (so pipe jumpers are needed if you have TRAMs larger than size one),
- there are large components under the TRAM sites, so SIL spacer strips are needed. When a TRAM is placed over the on-board transputer (in slots 0 and 1), and the TRAM has components on its underside, use of an additional set of spacers is recommended to avoid overheating the transputer,
- there are no holes in the board, so the TRAMs cannot be secured with nylon bolts.

5.2 Fitting memory

The TMB04 can be fitted with 4, 8, 12 or 16 SIMMs, which can be either 256 KByte or 1 MByte. The board is normally supplied with 60 or 70 ns SIMMs. Memory should be expanded in the sequence shown below - populating the SIMM slots in numerical order (the numbers are printed on the PCB):

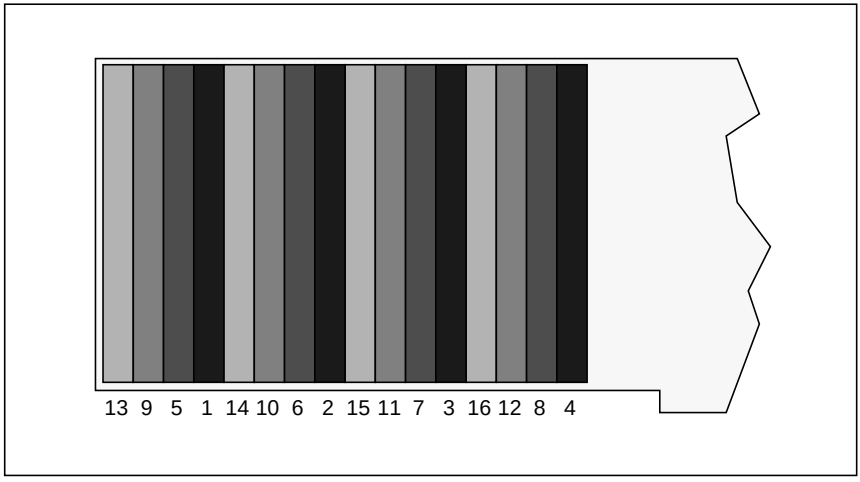


Figure 39. SIMM slot numbering.

See section 5.3.1 below for setting the memory speed selection switches.

5.3 Board Configuration Jumpers

This section describes the various board configuration settings and TMB04 specific functionality.

The jumpers and switches are briefly described below, and in detail in the following sections.

SW1	DMA and IRQ selection, IO base address selection, C012 and TRAM slot link speed selection, reset subsytem selection.
SW2	On-board transputer clock speed and link speed selection.
SW3	Memory speed selection.
LK5	Connect transputer link 0 to host or edge connector.
LK6	The source of control for the on-board transputer can be from either the host PC or from the edge connector (UP).
LK11	Selects the size of the memory SIMMs.
LK12	The TRAMs in slots 0 to 3 can be controlled either from the subsystem of the on-board transputer or from the source which controls the transputer.

In the following sections, it is assumed that you are holding the board component side up, with the PC bus edge connector pointing towards you and the D-type connector to the right.

The jumper groups LK5 and LK11 each consist of two jumpers, and LK6 and LK12 each consist of three jumpers. Each jumper can be in one of two positions, referred to as *left* and *right*. In the left position the jumper connects the central pin to the pin on the left, while in the right position it connects the central pin to the one on the right. All jumpers in a jumper group must be in the same position (left or right).

Each of the switch blocks consists of several switches, numbered from one at the top, downwards. The switches are *on* when the switch is moved to the right and *off* when it is moved to the left. The switch numbers and the on position are marked on the switch blocks.

5.3.1 On-board transputer clock and memory

The processor clock speed of the on-board transputer is selected by SW2, switches 4, 5 and 6, as follows. You may set a clock speed slower than that marked on the transputer, but not faster.

SW2.4	SW2.5	SW2.6	Processor clock
on	off	off	17.5 MHz
on	on	on	20 MHz (default)
on	off	on	25 MHz
off	off	on	30 MHz

Table 10: Processor clock speed select

The number of cycles taken to access local memory of the on-board transputer can be selected by SW3. The correct setting depends on the SIMM's row access time (T_{RAC}), the clock speed of the transputer (as set by SW2), and on the number of SIMMs fitted.

Processor clock	SIMM speed	Cycles	SW 3.1	SW 3.2	SW 3.3	SW 3.4
20 MHz	70 ns	3	on	off	off	off
25 MHz	60 ns					
30 MHz	40 ns					
20 MHz	90 ns	4	off	on	off	off
25 MHZ	70 ns					
30 MHz	60 ns					
20 MHz	120 ns	5	off	off	on	off
25 MHz	90 ns	(default)				
30 MHz	70 ns					
		6	off	off	off	on

Table 11: Memory access time selection (up to 8 MBytes)

If more than 8 MBytes of memory is fitted, add one memory cycle to the above.

Note that only one switch of SW3 should be on, or improper operation will result.

Jumper block LK11 selects the memory size of each SIMM fitted, as follows:

Jumper	Position	Description
LK11	Left	Memory is in 1 MByte SIMMs (default)
	Right	Memory is in 256KByte SIMMs

Table 12: SIMM size selection jumper

5.3.2 Control Configuration

The board's control configuration jumpers, LK6 and LK12, allow the source of control for the on-board transputer and the modules in slots 0 to 3 to be determined. The control consist of the TRAM signals reset, error and analyse.

Note that subsystem pins are not needed with the TMB04 - the on-board transputer acts as the master, and it's subsystem port is wired directly to the jumpers.

Jumper	Position	Description
LK6	Left	Transputer is controlled by the PC (default)
	Right	Transputer is controlled by the edge connector up port

Table 13: On-board transputer control jumper

Jumper	Position	Description
LK12	Left	Slots 0 to 3 are controlled from the transputer's subsystem port (default)
	Right	Slots 0 to 34 are controlled from the same source as the transputer

Table 14: On-board transputer subsystem jumper

By default, the reset line of the on-board transputer's subsystem is asserted whenever the on-board transputer is reset. However, the TMB04 allows you to disable this behavior, so the TRAM slots are not reset when the on-board transputer is reset.

Switch	Position	Description
SW1.8	on	transputer's reset is copied to subsystem (default)
	off	transputer's reset is not copied to subsystem

Table 15: On-board transputer subsystem reset handling

5.3.3 Board IO Address

Configuration of the board IO base address is achieved via switch SW1, as follows:

SW1.4	SW1.5	Description
off	on	Base address 150 hex (default)
on	off	Base address 200 hex
off	off	Base address 300 hex
on	on	Host link interface disabled

Table 16: Base address select switch

5.3.4 Link Speed Configuration

The speeds of various links on the board are set by switches as follows. Wherever two links are connected, they must be set at the same speed.

Switch	Position	Description
SW1.6	on	C012 link at 10 MBits/s
	off	C012 link at 20 MBits/s (default)

Table 17: C012 link speed selection switch

Switch	Position	Description
SW1.7	on	Links at 10 MBits/s
	off	Links at 20 MBits/s (default)

Table 18: TRAM slots 0 to 3 link speed selection switch

SW 2.1	SW 2.2	SW 2.3	on-board transputer link 0	on-board transputer links 1 to 3
on	on	on	10 MBits/s	10 MBits/s
on	on	off	10 MBits/s	5 MBits/s
on	off	on	5 MBits/s	10 MBits/s
on	off	off	5 MBits/s	5 MBits/s
off	on	on	10 MBits/s	10 MBits/s
off	on	off	10 MBits/s	20 MBits/s
off	off	on	20 MBits/s	10 MBits/s
off	off	off	20 MBits/s (default)	20 MBits/s (default)

Table 19: Master transputer link speed selection switch

5.3.5 Master and Slave Configuration

Link 0 of the on-board transputer may be connected either to the PC interface or to the D-type connector.

Jumper	Position	Description
LK5	Left	transputer link 0 connected to PC interface (default)
	Right	transputer link 0 connected to D-type connector (L0 on the hedgehog)

Table 20: On-board transputer link 0 jumper

5.3.6 IRQ & DMA Selection

The DMA channel and IRQ number can be set using SW1 as follows:

SW1.1	SW1.2	Description
on	on	No DMA channel
off	on	DMA channel 1 (default)
on	off	DMA channel 2
off	off	No DMA channel

Table 21: DMA channel selection switch

Switch	Position	Description
SW1.3	on	IRQ number 3 (default)
	off	IRQ number 6

Table 22: IRQ number selection switch

5.4 The Edge Connector

On the right hand edge of the TMB04 is a 37-way D-type edge connector. On the TMB04 the edge connector is used for two purposes:

- connecting to other motherboards control signals (via up, down and subsystem)
- connecting transputer links to construct the desired network topology.

Figure 40 shows the pinout of the edge connector.

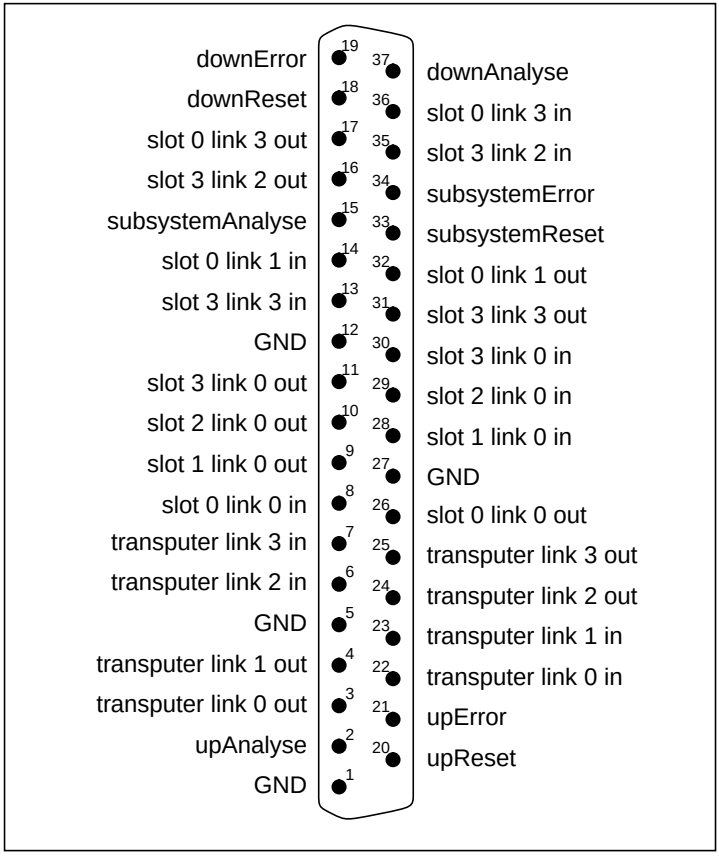


Figure 40. D-type pinout

Included in the accompanying cable pack is a mini-backplane board (“hedgehog”) which plugs into the edge connector and brings out the various links and ports onto standard connectors which accept link cables and reset cables. The pinout of this connector is shown in figure 41.

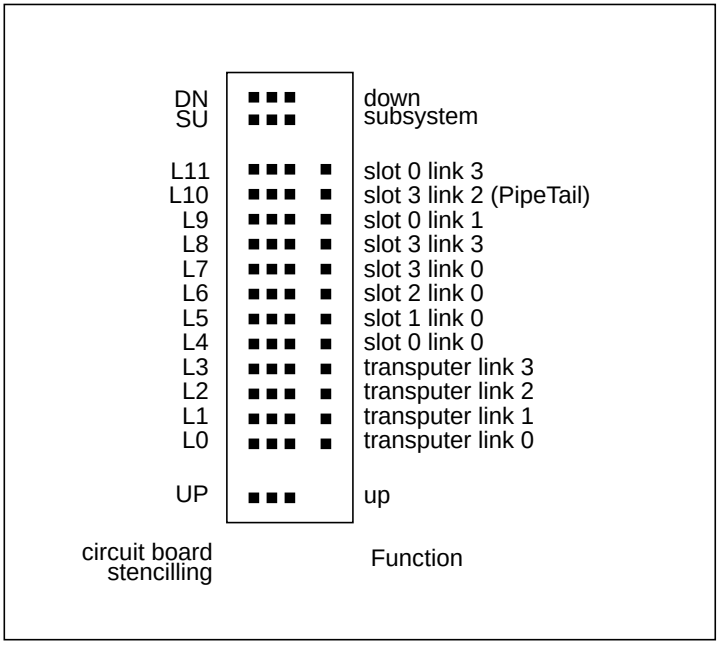


Figure 41. Connections on break out board

Essentially, the connector brings out the three control ports, the link0s and link3s of all the transputers (for network configuration) and the ends of the default pipeline. This is summarized for reference purposes in figure 42.

Note that to connect a link of the on-board transputer to one of the TRAM slots, the link must be looped back at the D-type, or made using a link cable and the hedgehog - there are no links connecting the transputer to the TRAM sites on the board.

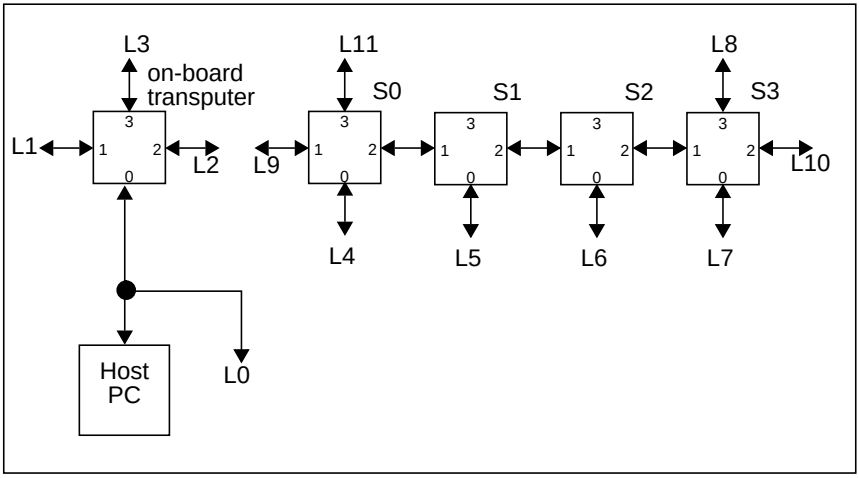


Figure 42. Summary of Network interconnect

Chapter 6
The TMB08 Motherboard

This chapter gives a detailed hardware description of the TMB08. The various features of the board are described and some examples of configuration are given.

6.1 Overview

The TMB08 is a full length PC hosted TRAM motherboard, with space to plug in up to ten Transputer Modules.

The TMB08 is shipped with an IMSC004 link switch, which provides for setting up user defined topologies. The switch is flexible enough to allow any TRAM's link 0 or 3 to be connected to any other TRAM's link 0 or 3 or any one of eight edge connections.

The TMB08 has a link patch area which, amongst other things, allows *PipeHead* and *ConfigUp* to be connected together.

Figure 43 shows the TRAM layout of this board for reference.

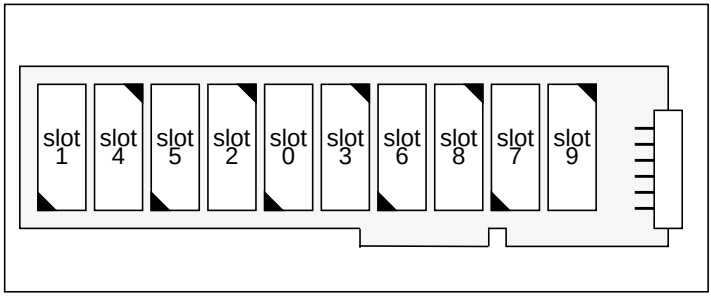


Figure 43. TMB08 TRAM layout

Figure 44 shows the board layout for reference.

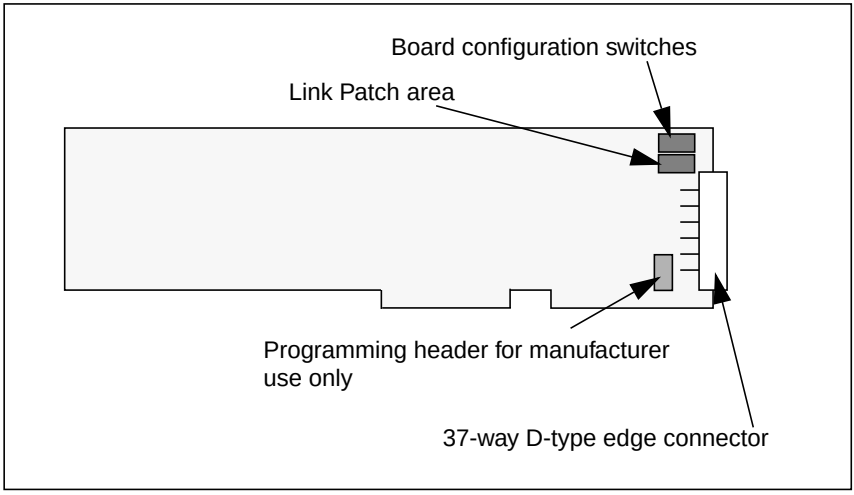


Figure 44. TMB08 board layout

6.2 Network Configuration

This section provides an overview of network configuration on the TMB08. It describes the wiring of the electronic link switch, the patch area and shows the relationship between these and the edge connector.

6.2.1 Electronic Link Configuration

This section describes the organization of the electronic link switch, the IMSC004.

In this application the C004 is used simply as a crossbar switch. The device is connected to 30 links, and can switch any link connected to any other link connected.

The links connected to the C004 are:

- link0 of all TRAM slots except module0 (module0 link0 is always connected to the host PC),
- link3 of all TRAM slots,
- link 0 of the T2 configuration processor,

- eight edge connectors,
- two spare links, which are taken to the patch area.

The C004 is programmed via the T2 configuration processor on the board. This in turn is programmed via *ConfigUp*.

The connections are shown in figure 45 for reference.

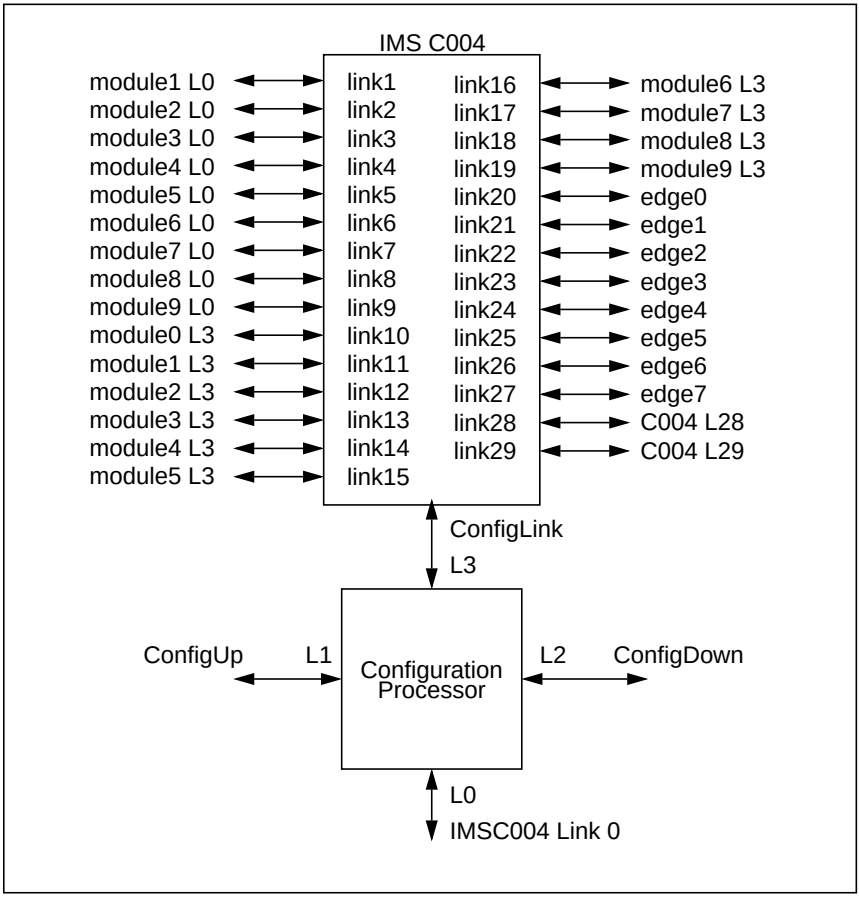


Figure 45. C004 Wiring

6.2.2 The Link Patch Area

The link patch area is a 6x2 jumper header on 0.1" spacing. The required connections can be made by moving push-on shorting links to the correct positions

The primary purpose of the patch area is to allow *PipeHead* and *ConfigUp* to be terminated correctly. Using the patch it is possible to either connect these two together (for the first motherboard in a system) or to take them off the board (for a slave motherboard).

The links taken to the patch area are:

- *ConfigUp*, i.e., link1 of the configuration processor
- *PipeHead*. i.e., Module0 link1
- two of the links from the crossbar switch (*C004L28/29*)
- two links which are taken directly to the edge connector (*patch0/1*)

Figure 46 shows the links attached to the patch area and the default connections made when the board is shipped.

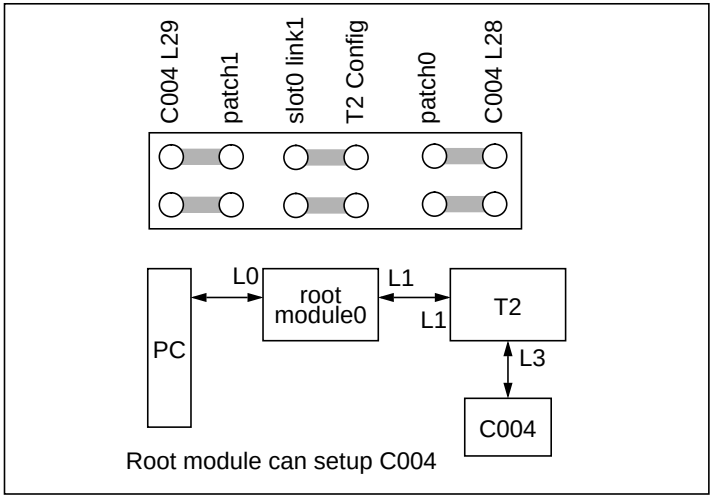


Figure 46. The Link Patch Area for Master Board

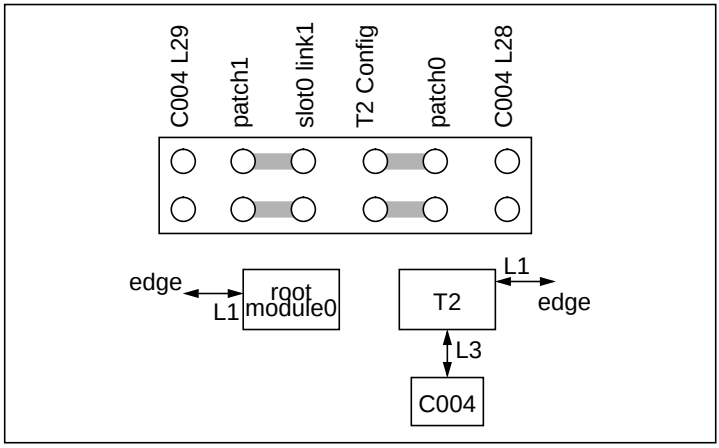


Figure 47. TMB08 Patch Area, connections for slave board

6.2.3 Summary of Network Configuration

Figure 48 shows the interconnection between the module slots, the electronic link switch, the link patch area and the edge connector. It is included for reference.

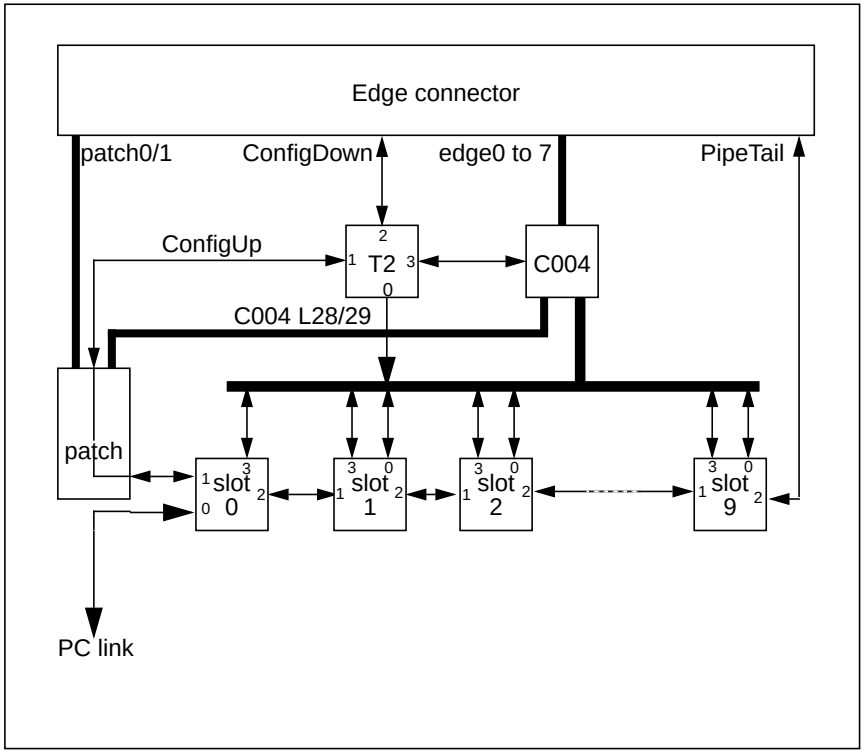


Figure 48. Network configuration summary

6.3 Description

6.3.1 Board Configuration

The basic board configuration is achieved by the use of the configuration switches. In the top right hand corner of the board there is a 6 way switch bank. Switch 1 is on the right, switch 6 left is not used.

The switches control the following functions:

- S1, S2 Board base address
- S3 Module link speed
- S4 Modules 1-9 control source
- S5 Module 0 control source

6.3.1.1 Link Speed

Switch 3 controls the transputer link speed. With the switch off/open (default) the links run at 20MHz. With the switch on/closed the links run at 10MHz.

6.3.1.2 Board Address

Switches S1 and S2 select the board base address. The four possible options are shown in Figure 49.

S2	S1	Board address (Hex)
on	on	150
off	on	200
on	off	250
off	off	300

Figure 49. Board address options

6.3.1.3 Control Configuration

There are two configuration options relating to board control:

- *S4 MD0*, source of control for modules1 to 9.
- *S5 IBM*, source of control for module0

If switch S4 is on, then the source of control for modules1 to 9 is from the same source as module0. If S4 is off then modules1 to 9 are controlled from module0's subsystem.

If switch S5 is on, then the source of control for module0 is from *up* on the edge connector. If S5 is off then the source of control is the host PC.

Figure 50 summarizes for the case of the link area.

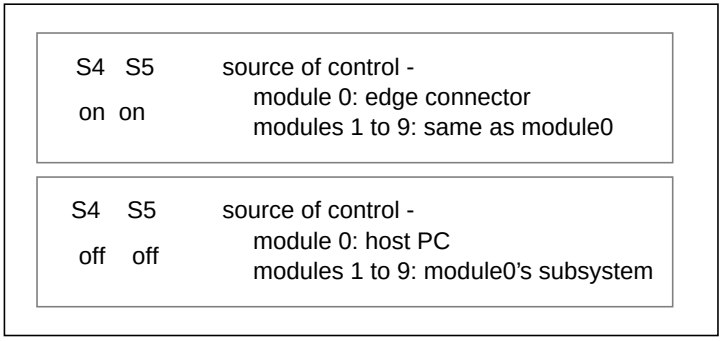


Figure 50. Control configuration (link)

6.3.2 IRQ & DMA Selection

The interrupt request level and DMA channel used are programmed by writing to 4bits in the IRQ and DMA channel select register. At base address +#14. This is a read/write register so that the programmed selection can be read back. The register coding is as shown in figure 51 and figure 52

Bit 1	*Bit 0*	*IRQ*	
0	*0*	*3*	*- reset value*
0	*1*	*5*	
1	*0*	*11*	
1	*1*	*15*	

Figure 51. IRQ Channel select.

.

Bit 3	*Bit 2*	*DMA Channel*
0	*0*	*0 - reset value*
0	*1*	*1*
1	*0*	*DMA Disabled*
1	*1*	*3*

Figure 52. DMA Channel select.

Note: although the TMB08 has a 8 bit (XT) interface the board is that of a 16bit (AT) board. The extra connector (short connector) allows more flexible IRQ and DMA selection. If the board is plugged into a 8 bit only slot then only IQR 3 or 5 and DMA 1 or 3 can be used.

6.3.3 The Edge Connector

On the right hand edge of the TMB08 is a 37-way D-type edge connector.

On the TMB08 the edge connector is used for connection to other motherboards. For this purpose the following are brought out:

- the three control ports and the configuration link,
- twelve transputer links.

Figure 53 shows the pin-out of the edge connector.

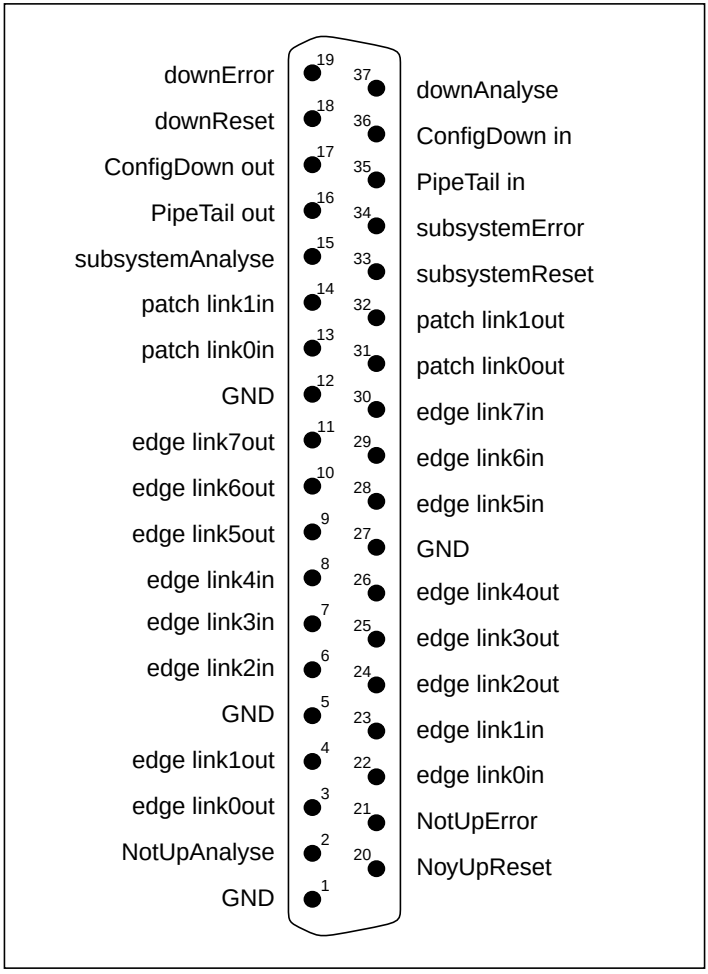


Figure 53. D-type pinout

Included in the accompanying cable pack is a mini-backplane board which plugs into the edge connector and brings out the various links and ports onto standard connectors which accept link cables and reset cables. The pinout of this connector is shown in figure 54.

.

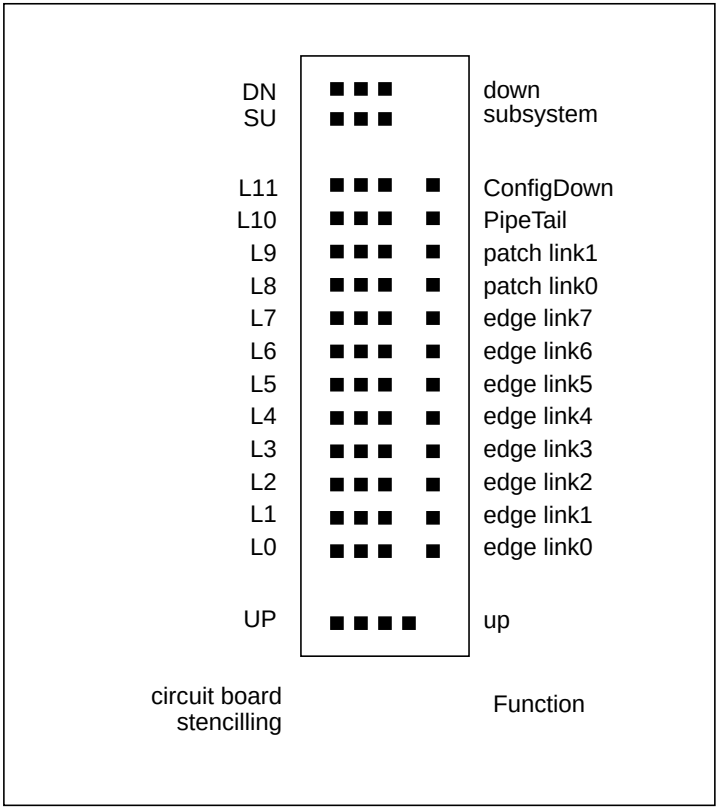


Figure 54. Connections on break out board

6.3.4 The Link Patch Area

The default wiring for the link patch area is to connect:

- *ConfigUp* to *PipeHead*,
- *patch0* to *C004 L28*,
- *patch1* to *C004 L29*.

The *patch0/1* connections allow 10 links to be brought out from the C004 to the edge connector.

The pinout of the link patch area is shown in figure 55 along with the default connections.

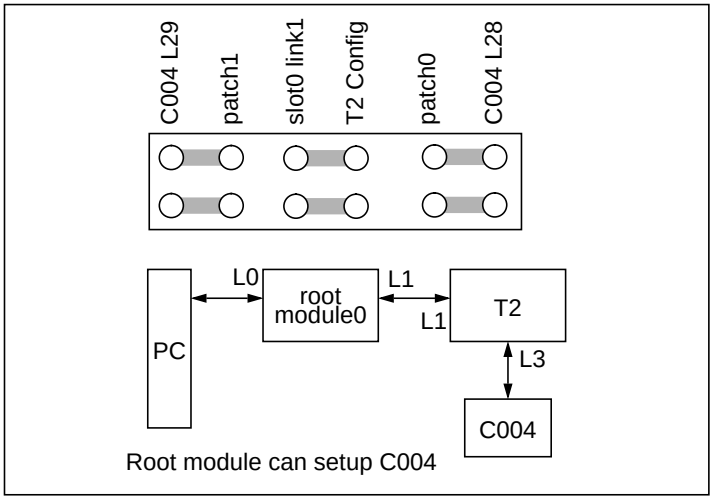


Figure 55. TMB08 Patch Area, connections for master board

6.4 Examples

This section shows how to set the board configuration options for two examples:

- a single TMB08 configured for use with an Inmos Toolset,
- two TMB08s connected as a single system, configured for use with an Inmos Toolset.

6.4.1 Stand-alone TMB08

The most common stand-alone configuration for the TMB08 is for use with the Inmos Toolsets, with every TRAM reset from the PC.To achieve this configuration, set the link the link patch area as shown in figure 55, and make the following setting:

:

Switches	Setting	Description
S1 S2	on on	Base Address #150
S3	off	Use 20 Mbit/s links
S4	on	Slot1-9 reset as slot 0
S5	off	Slot 0 controlled from PC

Table 23: Default settings for stand-alone operation

6.4.2 Multiple TMB08s

As an example of connecting multiple motherboards together consider a system consisting of two TMB08s in the same PC. In this case:

- the two boards must have different addresses
- one board must be slaved to the other
- the default pipeline needs to be connected
- the configuration pipeline needs to be connected

First set up one board with the same configuration as for stand-alone operation (see above). Ensure that pipe-jumpers are used in empty TRAM slots, and in the inactive slots of any TRAMs larger than size 1, so that link 2 of the last TRAM on the motherboard is taken out to pipe tail.

Then set up the second board as follows. The link patch area should be configured as shown in figure 56.

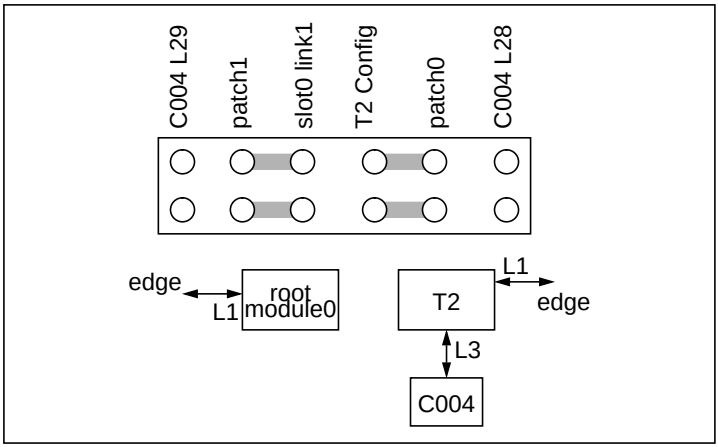


Figure 56. TMB08 Patch Area, connections for slave board

The jumpers and links on the slave board should be set up as follows:

Switches	Setting	Description
S1 S2	on off	Bus address #200
S3	off	Use 20 Mbit/s links
S4	on	Slots 1 to 9 controlled from same source as slot 0
S5	on	Slot 0 controlled from UP

Table 24: Settings for slave operation

Plug both boards into the PC and fit hedgehogs, and make the following connections between the boards:

Cable	First board	Second board
Reset Cable	Down (DN)	Up (UP)
Link Cable	Pipetail (L10)	Patch 1 (L9)
Link Cable	ConfigDown (L11)	Patch 0 (L8)

Table 25: Connections between TMB08s

Chapter 7
The TMB12 Motherboard

7.1 Overview

The TMB12 is a double extended eurocard TRAM motherboard, with space for up to sixteen Transputer Modules.

The board is essentially a stand-alone motherboard in that it contains no interface hardware to any host computer. Connection to other transputer systems is achieved via an edge connector which brings 32 links out of the board. Board services (power, configuration and system services) are also brought out to an edge connector.

The TMB12 is designed to be plugged into a purpose designed rack (the Transrack) which provides mechanical stability, power and cooling services.

The TMB12 has two C004 link switch chips and a controlling T222 transputer on board in order to provide for software control of network configuration.

Figure 57 shows the layout of TRAM sites on the motherboard for reference.

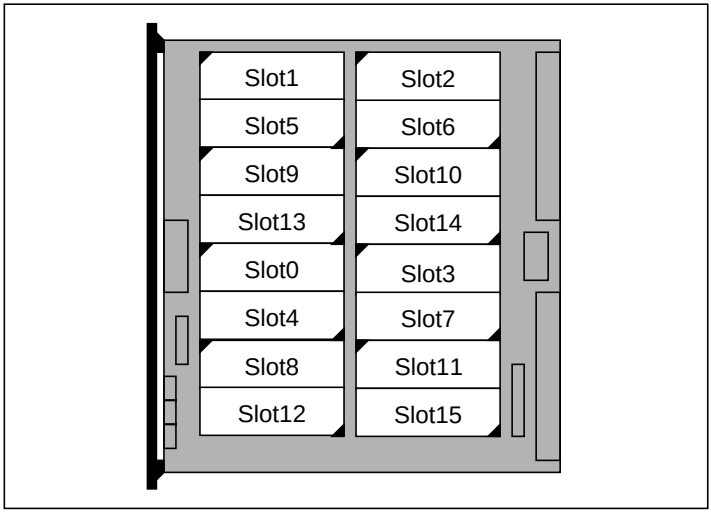


Figure 57. TMB12 TRAM layout

Figure 58 shows the general board layout (including configuration jumpers and switches) for reference.

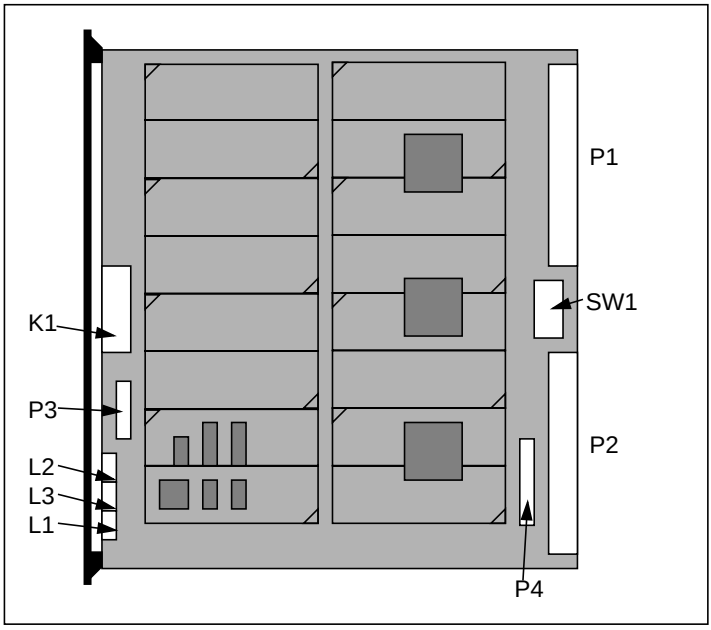


Figure 58. TMB12 board layout

The major components outlined in the figure are described very briefly here:

- P1 - carries 32 transputer links off the board,
- P2 - carries power, pipeline and configuration links, and system control signals off the board,
- K1 - allows the default pipeline to be broken up,
- SW1 - provides for board configuration (mainly link speeds).

7.2 Description

7.2.1 Board Configuration

The TMB12 has a number of hardware options which are selectable by switch bank SW1. Table 26 shows the options that can be selected.

Switch	Setting	Description
SW1.1	off	C004 links at 20Mbits/s (default)
	on	C004 links10Mbits/s
SW1.2 & SW1.3	off	All TRAMs at 20Mbits/s (default)
	on	All TRAMs at 10Mbits/s
SW1.4	off	T2 link 0 at 20Mbits/s (default)
	on	T2 link 0 at 10Mbits/s
SW1.5	off	T2 links 1, 2 & 3 at 20Mbits/s (default)
	on	T2 links 1, 2 & 3 10Mbits/s
SW1.6	off	Slots 1 to 15 controlled from slot 0 subsystem
	on	Slots 1 to 15 controlled from UP (default)

Table 26: TMB12 Link speed and control selection

In nearly all applications, SW1.1 to SW1.5 will be OFF, and SW1.6 will be ON.

Note that it is only sensible to have all the links on the board operating at the same speed.

Slot 0 is always controlled from the UP port.

7.2.2 The P1 Edge Connector

The TMB12 has two edge connectors called P1 and P2. Both of these connectors are standard DIN41612 96 way connectors. Ensure that when wiring to one of these connectors you do not

accidentally wire to the wrong connector - this could cause permanent damage to the board.

Connector P1 carries 32 links from the electronic switches, whilst connector P2 carries a number of board and system services.

The P1 edge connector brings 32 transputer links out of the TMB12 for connection to other boards. Table 27 shows the pinout of this connector.

When looking at the component side of the TMB12 with the DIN connectors to the right, pin 0 of the connector is at the top. This is also true when the TMB12 is mounted into a Transrack. The numbering of the links is consistent with table 30 and the numbering scheme used by NCS.

In order to assist in connecting standard link cables to the edge connector, the TMB12 comes with a break out board in the cable pack. This plugs into the P1 edge connector and brings the links out onto standard link plugs.

7.2.3 The P2 Edge Connector

Edge connector P2 carries:

- the power supply (+5V required, rated @ at least 3Amps),
- up, down & subsystem,
- ConfigUp & ConfigDown,
- PipeHead & PipeTail,
- a link from the C004 switches,
- a link to the K1 header block,
- and a number of uncommitted pins (connected to P4).

Note that under normal operation (the TMB12 mounted in a Transrack), sufficient power and cooling are provided. A power cable is provided with the TMB12 for stand-alone operation, but it is not recommended that stand-alone operation be permanent.

Table 28 gives the detailed pinout of this connector.

In order to allow standard link and reset cables to be attached to the TMB12, a special back-to-back connector is supplied in the cable pack which has a number of keying pins (either pins removed or pins sleeved) which assist in locating the cables to the correct point. This is shown in figure 59.

Pin	c	b	a
edge L0	IC3linkout0	IC2linkin0	GND
edge L1	IC3linkout2	IC2linkin2	GND
edge L2	IC2linkout4	IC3linkin4	GND
edge L3	IC2linkout5	IC3linkin5	GND
edge L4	IC2linkout6	IC3linkin6	GND
edge L5	IC2linkout3	IC3linkin3	GND
edge L6	IC3linkout1	IC2linkin1	GND
edge L7	IC3linkout7	IC2linkin7	GND
edge L8	IC3linkout29	IC2linkin29	GND
edge L9	IC3linkout30	IC2linkin30	GND
edge L10	IC2linkout31	IC3linkin31	GND
edge L11	IC2linkout28	IC3linkin28	GND
edge L12	IC3linkout24	IC2linkin24	GND
edge L13	IC2linkout25	IC3linkin25	GND
edge L14	IC2linkout26	IC3linkin26	GND
edge L15	IC3linkout27	IC2linkin27	GND
edge L16	IC3linkout17	IC2linkin17	GND
edge L17	IC2linkout19	IC3linkin19	GND
edge L18	IC2linkout22	IC3linkin22	GND
edge L19	IC3linkout23	IC2linkin23	GND
edge L20	IC2linkout16	IC3linkin16	GND
edge L21	IC2linkout18	IC3linkin18	GND
edge L22	IC3linkout21	IC2linkin21	GND
edge L23	IC3linkout20	IC2linkin20	GND
edge L24	IC3linkout10	IC2linkin10	GND
edge L25	IC3linkout13	IC2linkin13	GND
edge L26	IC3linkout14	IC2linkin14	GND
edge L27	IC3linkout11	IC2linkin11	GND
edge L28	IC2linkout8	IC3linkin8	GND
edge L29	IC2linkout9	IC3linkin9	GND
edge L30	IC2linkout12	IC3linkin12	GND
edge L31	IC2linkout15	IC3linkin15	GND

Table 27: P1 connector pinout

Pin	c	b	a
0	GND	GND	GND
1	VCC	VCC	VCC
2	PAUX	nc	PAUX
3	VCC	VCC	VCC
4	GND	GND	GND
5	VCC	VCC	VCC
6	GND	GND	GND
7	nc	nc	nc
8	slot0linkout1	slot0linkout0	slot15linkout2
9	slot0linkin1	slot0linkin0	slot15linkin2
10	GND	GND	GND
11	nc	nc	nc
12	GND	GND	GND
13	nc	nc	nc
14	IC1linkout1	IC3linkout22	IC1linkout2
15	IC1linkin1	IC2linkin22	IC1linkin2
16	GND	GND	GND
17	nc	nc	nc
18	P4/3	nc	P4/2
19	P4/4	nc	nc
20	P4/5	GND	nc
21	P4/6	nc	notSubReset
22	P4/7	K1/11	notSubAnalyse
23	P4/8	K1/10	notSubError
24	P4/9	GND	GND
25	P4/10	nc	nc
26	nc	GND	nc
27	notUpReset	nc	notDownReset
28	notUpAnalyse	K1/3	notDownAnalyse
29	notUpError	K1/18	notDownError
30	GND	GND	GND
31	GND	GND	GND

Table 28: P2 connector pinout

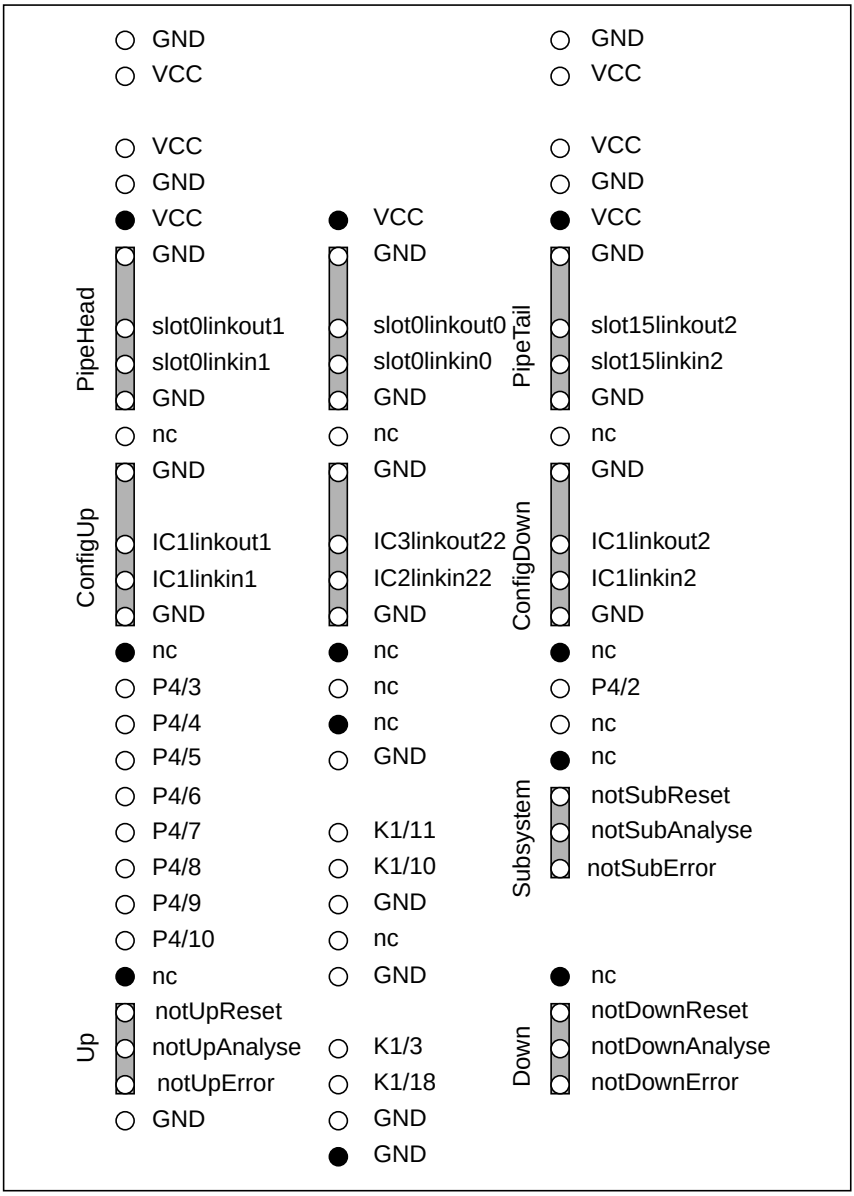


Figure 59. Mini-backplane P2 Connections

7.2.4 Other Hardware

This section describes the remaining hardware outside of the electronic link switches.

7.2.4.1 Error Lights

There are three LEDs mounted on the edge of the TMB12 protuding through the front plate. They monitor the error signals coming from the various slots. Figure 60 summarizes.

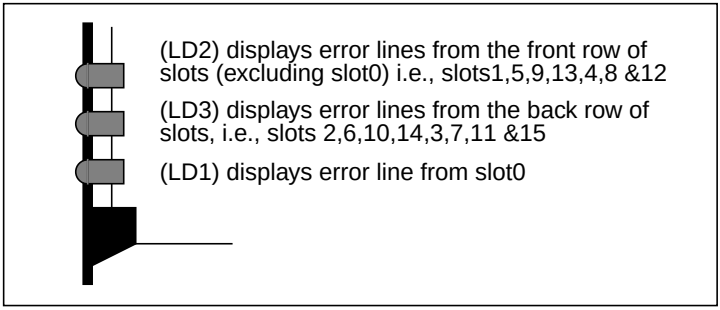


Figure 60. Error lights on the TMB12

7.2.4.2 User Power Connector

There is an optional four-way power connector, P3, mounted at the front of the board. The connector plug is compatible with the power supply socket on most disk units, and can be used to drive such peripherals. From top to bottom the pins are:

Top	connected to P2 3a & 3c (PAUX)
	0V
	0V
Bottom	5V

Table 29: User Power connector, P3

The pins are rated for currents up to 3Amps.

7.2.4.3 Uncommitted Pins

Nine of the pins on edge connector P2 are wired to nine of the pins (2 through 10) of an uncommitted connector, P4, on the board. The two other pins of P4 (1 & 11) are wired to ground. P4 allows

application specific signals to be brought onto the motherboard, e.g., RS232 lines. The individual pins of P4 are rated at 50mA @ 25V with respect to ground. Note that pin1 of P4 is at the top.

7.3 Network Configuration

This section provides an overview of network configuration on the TMB12. It describes the electronic link switching, the pipeline and the relation with the edge connectors.

7.3.1 Electronic Link Switching

This section describes the organization of the electronic link switches.

The two C004s on the TMB12 allow complex transputer topologies to be constructed. They are used in a slightly unusual way to allow the maximum amount of re-connectivity on the TMB12 given the constraints of the hard-wired pipeline.

In general, the link output signals from all the link0s on all the slots (16 signals) are connected to 16 inputs of one of the C004s (IC2). The link input signals from all the link3s on all the slots (16 signals) are connected to 16 of the outputs of the same C004. The C004 can therefore switch any link0 output to any link3 input.

Similarly, the other C004 is connected to the TRAM slot's link0 inputs and link3 outputs. This C004 can therefore switch any link3 output to any link0 input.

This means that each half of a switched link connection between two transputers is routed through a different C004. The result of this wiring scheme is that any link0 of any transputer can be connected to any link3 of any transputer. However, a link0 may not connect directly to another link0.

In a similar fashion to the transputer links, there are 32 edge connector links. Edge links are divided equally into two types:

- type 0: wired as per transputer link0's, i.e., link output to IC2 and link input from IC3
- type3: wired as per transputer link3's, i.e., link output to IC3 and link input from IC2.

Hence, the link switching that can be achieved is:

- any link0 to any link3,
- any link0 to any edge link of type3,

- any link3 to any edge link of type0,
- any edge link of type0 to any edge link of type3.

Where there are 16 link0's, 16 link3's and 16 each of edge links types 0 & 3.

Note that a link0 can be wired to another link0 by connecting the two link0s to two edge connector links and hard-wiring the edge connector links together.

Figure 61 shows the general wiring scheme between the TRAM slots and the switches, figure 62 the connections to the T2 configuration processor, and table 30 shows the link wiring in full detail.

Unless otherwise stated the edge connector referred to is P1. These figures are included for reference only. Programming of the link switches is best achieved with the NCS provided.

Note that the naming of the links connected to the edge connector P1 is from the perspective of the edge connector looking towards the C004s. Hence, *edge L6 out* is a link wire going towards the C004s, and is connected to the input of the appropriate C004.

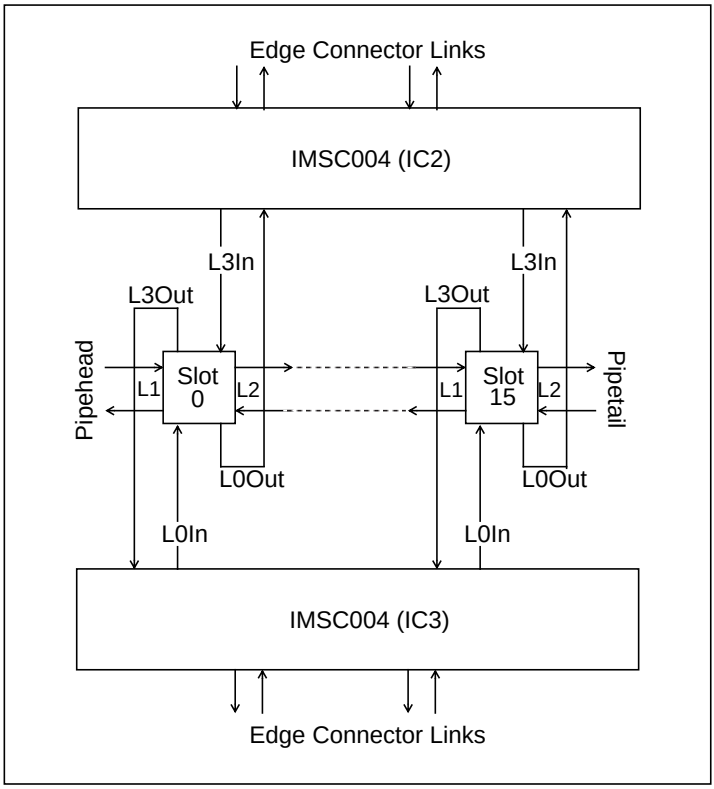


Figure 61. Connection details of the link switches

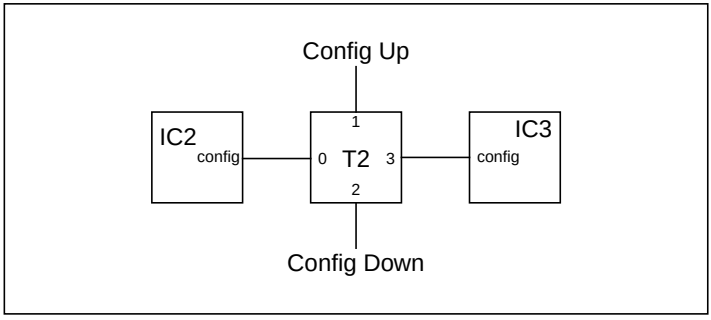


Figure 62. Connections to the configuration processor

Link No.	IC2 link in	IC2 link out	IC3 link in	IC3 link out
0	edge L0 out	slot 2 L3 in	slot 2 L3 out	edge L0 in
1	edge L6 out	slot 5 L3 in	slot 5 L3 out	edge L6 in
2	edge L1 out	slot 1 L3 in	slot 1 L3 out	edge L1 in
3	slot 5 L0 out	edge L5 in	edge L5 out	slot 5 L0 in
4	slot 2 L0 out	edge L2 in	edge L2 out	slot 2 L0 in
5	slot 1 L0 out	edge L3 in	edge L3 out	slot 1 L0 in
6	slot 6 L0 out	edge L4 in	edge L4 out	slot 6 L0 in
7	edge L7 out	slot 6 L3 in	slot 6 L3 out	edge L7 in
8	slot 15 L0 out	edge L28 in	edge L28 out	slot 15 L0 in
9	slot 8 L0 out	edge L29 in	edge L29 out	slot 8 L0 in
10	edge L24 out	slot 15 L3 in	slot 15 L3 out	edge L24 in
11	edge L27 out	slot 11 L3 in	slot 11 L3 out	edge L27 in
12	slot 12 L0 out	edge L30 in	edge L30 out	slot 12 L0 in
13	edge L25 out	slot 12 L3 in	slot 12 L3 out	edge L25 in
14	edge L26 out	slot8 L3 in	slot8 L3 out	edge L26 in
15	slot 11 L0 out	edge L3 in	edge L3 out	slot 11 L0 in
16	slot 7 L0 out	edge L20 in	edge L20 out	slot 7 L0 in
17	edge L16 out	slot 3 L3 in	slot 3 L3 out	edge L16 in
18	slot 4 L0 out	edge L21 in	edge L21 out	slot 4 L0 in
19	slot 3 L0 out	edge L17 in	edge L17 out	slot 3 L0 in
20	edge L23 out	slot 7 L3 in	slot 7 L3 out	edge L23 in
21	edge L22 out	slot 4 L3 in	slot 4 L3 out	edge L22 in
22	P2 pin b15 (slot 0 L0 out)	edge L18 in	edge L18 out	P2 pin b14 (slot 0 L0 in)
23	edge L19 out	K1 pin 20 (slot 0 L3 in)	K1 pin 1 (slot 0 L3 out)	edge L19 in
24	edge L12 out	slot 14 L3 in	slot 14 L3 out	edge L12 in
25	slot 13 L0 out	edge L13 in	edge L13 out	slot 13 L0 in
26	slot 14 L0 out	edge L14 in	edge L14 out	slot 14 L0 in
27	edge L15 out	slot 13 L3 in	slot 13 L3 out	edge L15 in
28	slot 9 L0 out	edge L11 in	edge L11 out	slot 9 L0 in
29	edge L8 out	slot 9 L3 in	slot 9 L3 out	edge L8 in
30	edge L9 out	slot 10 L3 in	slot 10 L3 out	edge L9 in
31	slot 10 L0 out	edge L10 in	edge L10 out	slot 10 L0 in

Table 30: Connections to the C004s

7.3.2 The K1 Header Block

The TMB12 has a slightly modified default pipeline (connecting the TRAM slots in a chain using links 1 and 2). This is designed to allow users to construct complex topologies, whilst maintaining compatibility with the TRAM standard.

The default pipeline is split into four sub-pipelines by the K1 header block. By default this header is jumpered such that the four sub-pipelines are connected together into one long pipeline.

The default pipeline is split at locations: 3-4, 7-8 and 11-12. Also taken to K1 are two links from the P2 edge connector and one link from the electronic switches. Figure 63 illustrates this along with the default connections.

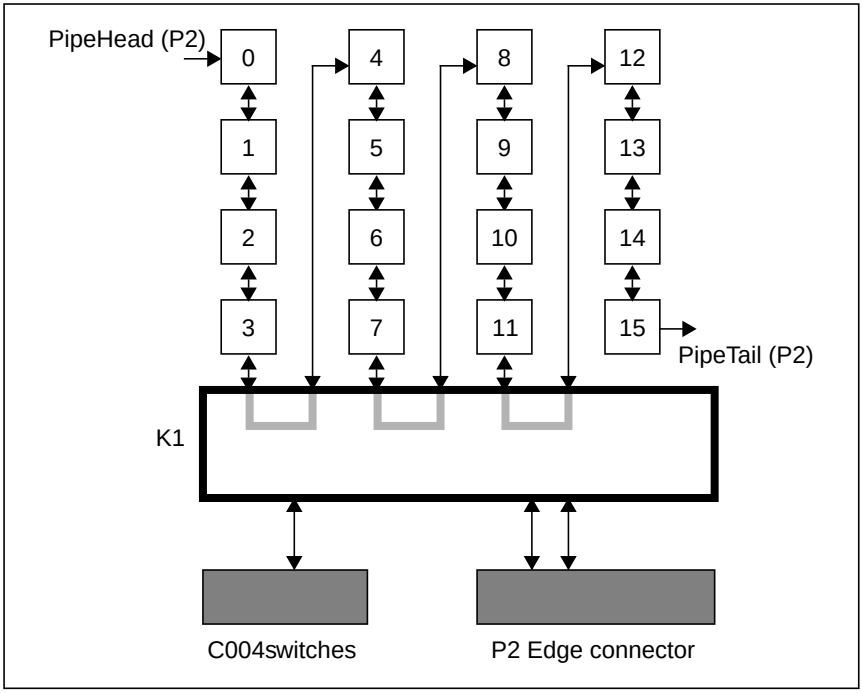


Figure 63. K1 connection diagram

K1 is a 20 pin header block organized 10 by 2 pins. Small jumpers are used to set the configuration.

Figure 64 gives the detailed pinout of the K1 connector and shows the default positions of the jumpers.

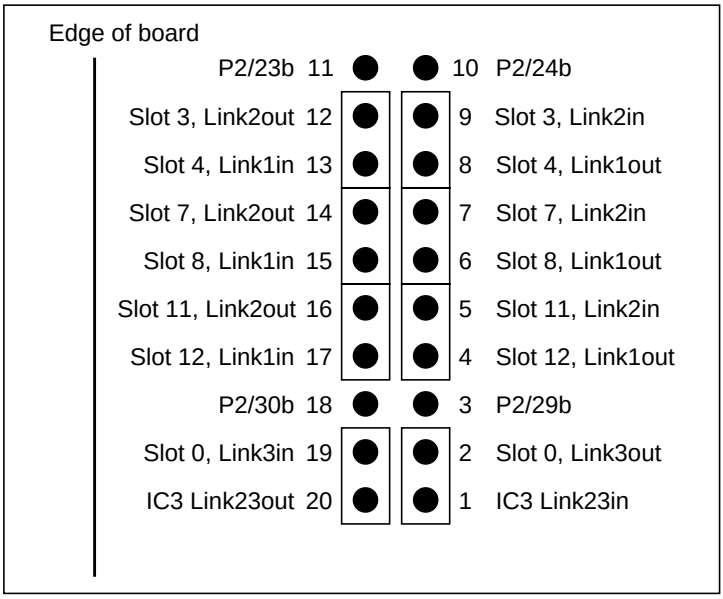


Figure 64. Details of K1 header, showing default jumper settings

7.3.3 The P1 Edge Connector

This section describes the network configuration aspects of the use of the P1 edge connector. For detailed pinouts see section 7.2.2 on page 76.

The edge links of P1 are divided up into type0 links and type3 links. Remember that the C004s can connect a type0 edge link to link 3 of any TRAM slot, and a type3 edge link to link 0 of any TRAM slot. Table 31 classifies all the edge links into one of these two types.

If the desired network topology includes connecting transputer link0's together, then this can be achieved by wiring the transputer links to the edge connector and using a short link cable to jumper the break out board. Clearly transputer link0s can be wired to any edge link of type3. Table 32 shows the recommended wiring to make on the break out board to achieve link0-link0 and link3-link3 connections. Note that this table also describes the default wiring of the break out board on shipping.

P1 edge Link	Link type	P1 edge Link	Link type
edge L0	type 0	edge L16	type 3
edge L1	type 0	edge L17	type 3
edge L2	type 3	edge L18	type 3
edge L3	type 3	edge L19	type 0
edge L4	type 3	edge L20	type 3
edge L5	type 3	edge L21	type 3
edge L6	type 0	edge L22	type 0
edge L7	type 0	edge L23	type 0
edge L8	type 0	edge L24	type 0
edge L9	type 0	edge L25	type 0
edge L10	type 3	edge L26	type 0
edge L11	type 3	edge L27	type 0
edge L12	type 0	edge L28	type 3
edge L13	type 3	edge L29	type 3
edge L14	type 3	edge L30	type 3
edge L15	type 0	edge L31	type 3

Table 31: Edge link classification

Link 0		Link 3	
From	To	From	To
edge L2	edge L3	edge L0	edge L1
edge L4	edge L5	edge L6	edge L7
edge L10	edge L11	edge L8	edge L9
edge L13	edge L14	edge L12	edge L15
edge L17	edge L18	edge L16	edge L19
edge L20	edge L21	edge L22	edge L23
edge L28	edge L29	edge L24	edge L25
edge L30	edge L31	edge L26	edge L27

Table 32: Default P1 link cable connections

7.3.4 Summary

This section summarizes the network configuration of the TMB12 by providing a brief description of all of the aspects described in the preceding sections.

- the link0's and link3s of modules in sites 1 to 15 are taken to an electronic link switch array. Also taken to this array are 32 edge connector links (P1). In general, any module link0 can connect to any module link3 or to one of 16 edge connector links.
- the default pipeline is broken up into four equal length sub-pipelines by header K1. Normally these sub-pipelines are connected together into one long pipeline.
- slot0 link0 is taken directly to edge connector P2. Also taken to P2 is the C004 link that it would have been expected to be wired to. In normal use, these two links would be wired together using a specially provided link jumper (the "yellow plug cable", see figure 65). slot0 link0 is brought directly to the edge connector as it allows those applications that require it to have two directly wired links to other transputer equipment, i.e., links which bypass the C004s.

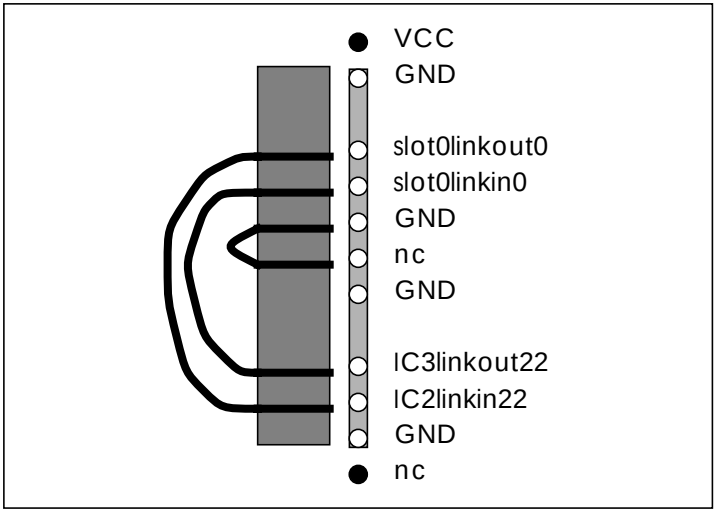


Figure 65. Jumpering slot0 link0 to the C004s

- slot0 link3 is taken to the header block K1. Also taken to K1 is the C004 link that slot0 link3 is normally wired to. Because there is a

link between K1 and P2, this allows slot0 link3 to be taken off the board, bypassing the electronic switches.

Figure 66 shows the main relations between P2, K1 and the link switches.

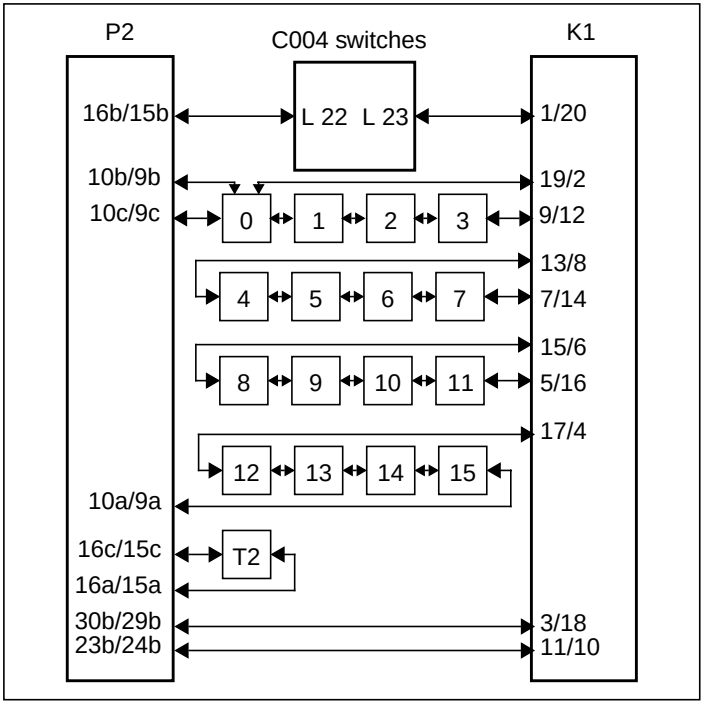


Figure 66. Network configuration

Chapter 8
The TMB14 Motherboard

8.1 Overview

The TMB14 is a 6U (160mm) VME TRAM motherboard, with space for up to eight Transputer Modules.

The link connections between the TRAMs are controlled by a pair of C004 link switches. 24 links are taken from these to edge connectors on the board. This arrangement allows almost any network topology to be adopted.

The board also provides a single bidirectional link and subsystem port which can be controlled through its VMEbus interface by a host computer.

Figure 67 shows the layout of the board with the major components, switches and jumpers highlighted.

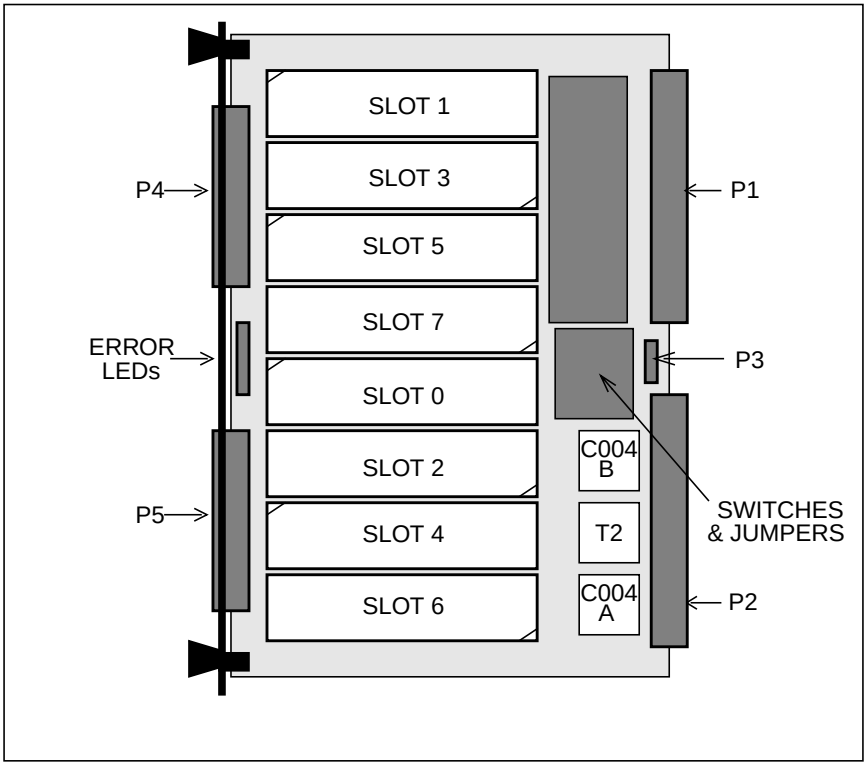


Figure 67. TMB14 board layout

There are five edge connectors:

- P1 - VMEbus connector,
- P2,4,5 - C004, pipeline and configuration links, and system control signals,
- P3 -user power connector.

The T2 and C004's A and B control the link interconnections. Because there are two C004's for eight TRAM slots, *all* the TRAM links are reconfigurable. It is therefore possible to emulate the hard-wired pipeline connecting links 1 and 2 of each TRAM, as used on other motherboards.

The error LEDs reflect the error status of each TRAM slot.

8.2 VMEbus Interface

The VMEbus interface on the TMB14 provides a single bidirectional link, *VMEbusLink*, and subsystem port which may be controlled by a host computer. It also generates interrupts under user defined conditions:

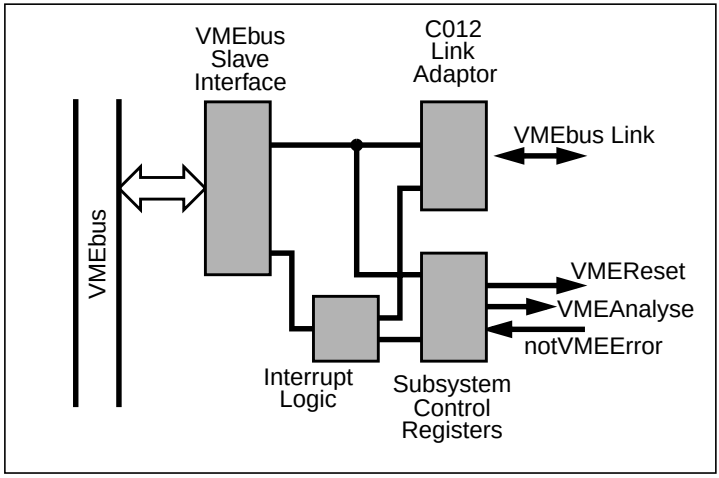


Figure 68. TMB14 VMEbus Interface

The interface implements an A16D08(O) slave and an INT(1-7):DO8(O) interrupter. It maps a number of registers into consecutive (odd) locations in the short address space starting at a base address set by switches SW1,2. These registers correspond to a C012 link adaptor, the subsystem port and the interrupt control logic.

NB. The interface can be completely disabled, if required, by removing a jumper - see section 8.4.1 on page 102.

8.2.1 Link Adaptor Registers

The first four locations are mapped to a C012, as shown in table 33.

Byte Offset	Bits	R/W	Description
#01	7 to 0	Read	Input Data
#03	7 to 0	Write	Output Data
#05	0	Read	Input Data Present
	1	Write	Input Interrupt Enable
#07	0	Read	Output Ready
	1	Write	Output Interrupt Enable

Table 33: C012 Registers

- Input Data Register: this register contains the last data byte received from the VMEbus link. Its contents are not valid unless the Data Present flag in the Input Status Register is asserted (set).
- Input Status Register: this register contains the Data Present flag (bit 0 - least significant bit) and Input Interrupt Enable bit (bit 1).The Data Present flag is set to indicate reception and latching of a data byte. Setting the Input Interrupt Enable bit will cause the C012 to generate an interrupt whenever this condition arises. Bits 2-7 are not used and must be written as zero's.
- Output Data Register: data written to this register is transmitted out of the VMEbus link. Data may only be written when the Output Ready flag in the Output Status register is set.
- Output Status Register: this register contains the Output Ready flag (bit 0 - least significant bit) and the Output Interrupt Enable bit. The Output Ready flag is set to indicate that the Output Data Register is ready for the next byte to be transmitted. Setting the Output Interrupt Enable bit will cause the C012 to generate an interrupt whenever this condition arises. Bits 2-7 are not used and must be written as zero's.

8.2.2 Subsystem Control Registers

The subsystem control registers allow the error states of the eight TRAMs to be determined. They also allow the subsystem port *VMEReset* and *VMEAnalyse* signals to be set and cleared.

Byte Offset	Bits	R/W	Description
#09	0	Read	notVMEError
		Write	VMEReset
#0B	0	Write	VMEAnalyse
#13	0 to 7	Read	TRAM errors

Table 34: Subsystem Control Registers

- Subsystem Reset/Error Register: bit 0 of this register returns the state of the *notVMEError* signal (active low) when read. Writing to this bit controls the state of *VMEReset*. This is asserted when set. Bits 1-7 are not used.
- Subsystem Analyse Register: writing to bit 0 of this register controls the state of *VMEAnalyse*. It is asserted when set. Bits 1-7 are not used.
- TRAM Error Register: reading this register returns the state of the error lines from the eight TRAMs. The bit position corresponds to the TRAM slot number, hence bit 0 reflects slot 0's error line and so on. (These bits are active low).

8.2.3 Interrupt Control Registers

Programming the interrupt control registers allows the user to determine which (if any) of three events will cause a VMEbus

interrupt, the level of such an interrupt and the vector number returned during an interrupt acknowledge cycle:

Byte Offset	Bits	R/W	Description
#0D	1	Write	Error Interrupt Enable
	2	Write	Output Interrupt Enable
	3	Write	Input Interrupt Enable
#0F	0 to 2	R/W	Interrupt Level
#11	0 to 7	Read	Interrupt Status/ID

Table 35: Interrupt Control Registers

- Interrupt Enable Register: bits 1-3 of this register are used to enable interrupts due to active notVMEError, C012 output ready, and C012 input ready respectively. To enable an interrupt the appropriate bit must be set. This register is zeroed at power on or during a VMEbus SYSRESET* condition.
- Interrupt Level Register: the value in bits 0-2 of this register determines the level of any VMEbus interrupt generated by the TMB14. The value corresponds directly to one of the seven interrupt lines IRQ1*-IRQ7*. A value of zero will effectively disable interrupts. This register is zeroed at power on or during a VMEbus SYSRESET* condition. It may be written and read.
- Interrupt Status/ID Register: the value in this 8 bit register is returned during an interrupt acknowledge cycle. It may be written and read. This register is more generally known as the Vme Interrupt Vector

8.3 Link and Control Configuration

Almost all of the link connections between the TRAM slots and the edge connectors are determined by the two C004 link switches. The exceptions are made by hardware jumper settings.

The subsystem connections between the TRAMs, the host computer and the edge connector are configurable by hardware jumper settings.

8.3.1 Links

8.3.1.1 Configuration Processor

The C004's are configured by the on board T222 transputer. This is itself programmed by downloading code in through its link 1, which can be connected to an edge connector signal (ConfigUp on P4 or P2), to TRAM slot 0 or to VMEbusLink. The T222's link 2 is taken to an edge connector signal (ConfigDown P5 or P2). This arrangement allows a configuration pipeline to be implemented - an indefinite number of motherboards, of mixed type if necessary, may be daisy-chained and all of their link switches programmed from the head of the pipeline:

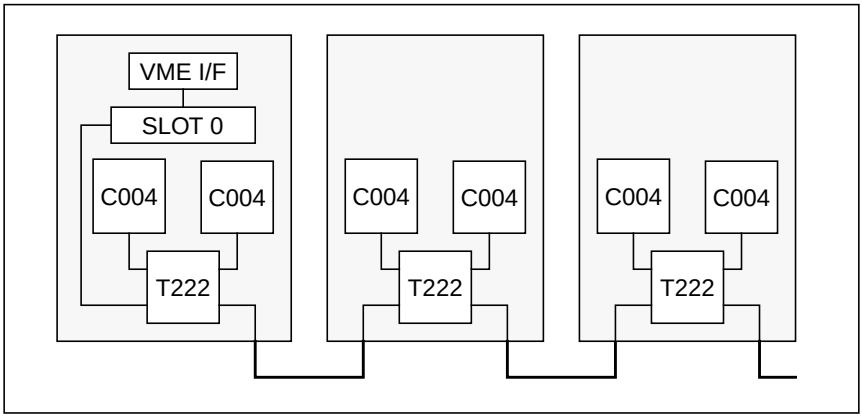


Figure 69. Configuration Pipeline

8.3.1.2 Link Switches

There is a program available to automate the process of making 'soft' link connections through the C004's, called MMS, which is part of the Inmos software package IMS S514 (this package also includes a device driver that is compatible with the TMB14). Obviously this program needs to know the physical connections of the links on the C004's and this information is shown in figure 70.

Note that there is no hard-wired pipeline of links between TRAM sites on the TMB14, so setting of the C004s is essential if more than one transputer is used.

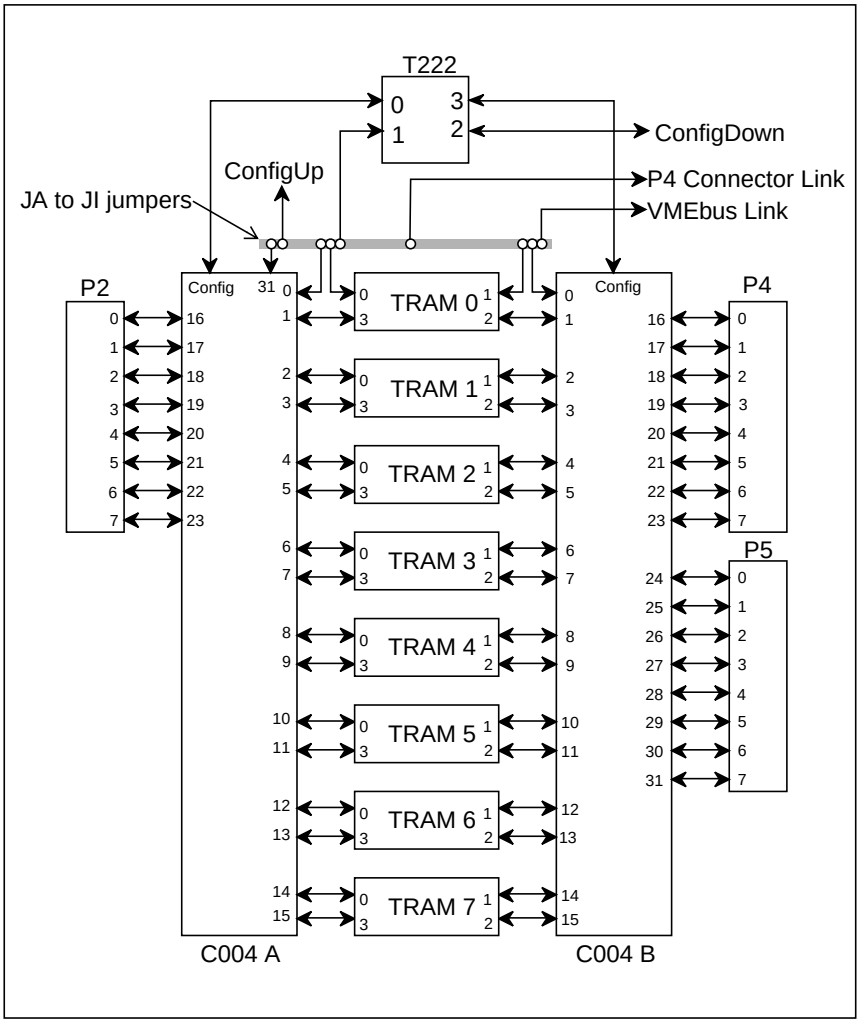


Figure 70. C004 Link Connections

8.3.1.3 Hardware Link Connections

Those link connections which must be made by hardware jumper settings (because they are involved in programming the connections of the other links) are detailed in figure 71. To make one of the 9

possible connections, the relevant pair of jumpers is installed, JA for link connection A and so on. Mutually exclusive settings must not be made as they will prevent the board from operating correctly. This means that there must not be more than one connection to any of the nine links listed.

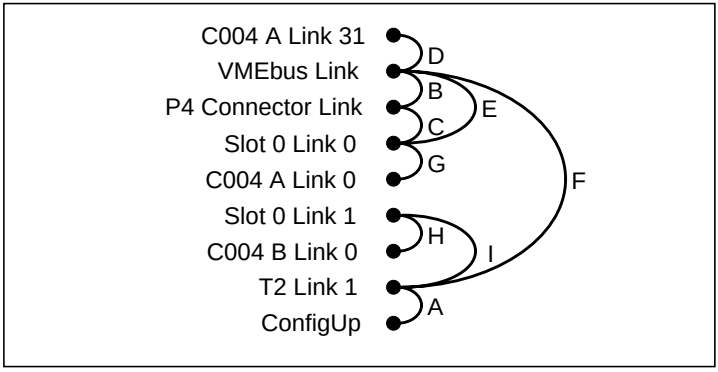


Figure 71. Logical link connections made with JA to JI

The locations of JA-JI are shown on figure 73, and the possible settings are summarized in section 8.4.4 on page 104.

8.3.2 Subsystem

The connection of the control signals (reset, analyse and error) between the VMEbus interface, the edge connectors and the TRAM slots is configured by four hardware jumpers. Figure 72 illustrates the possible connections with the three signals represented as a single bus line.

The switches are shown as they would be with the relevant jumper installed. Removing the jumper moves the switch to the oposite connection.

The ServicesUp/ServicesDown daisy-chain allows many boards to be controlled from one subsystem source, and is analogous to the configuration link pipeline.

The operation of the jumpers is sumarized in section 8.4.3 on page 103.

The error signal from the each TRAM slot is also brought out onto eight leds on the front panel. The top left led is ERROR0, the top right is ERROR1, the bottom left is ERROR6 and the bottom right is ERROR7.

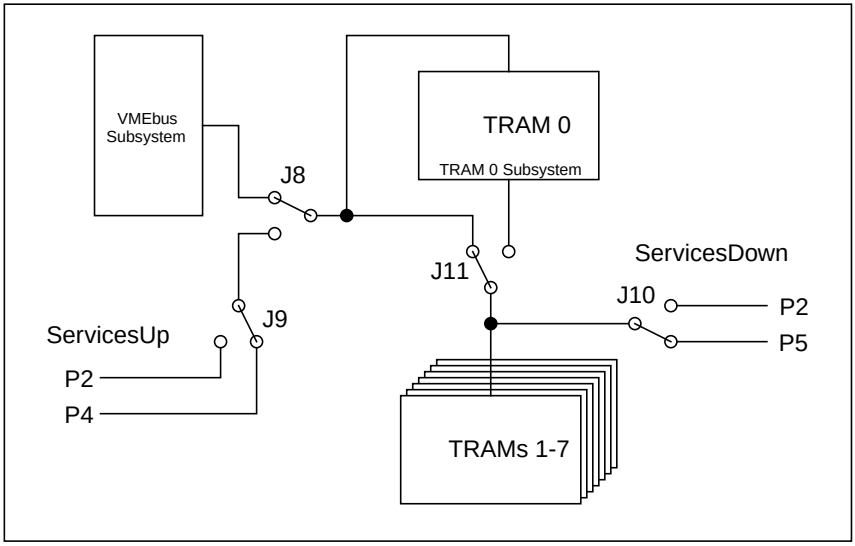
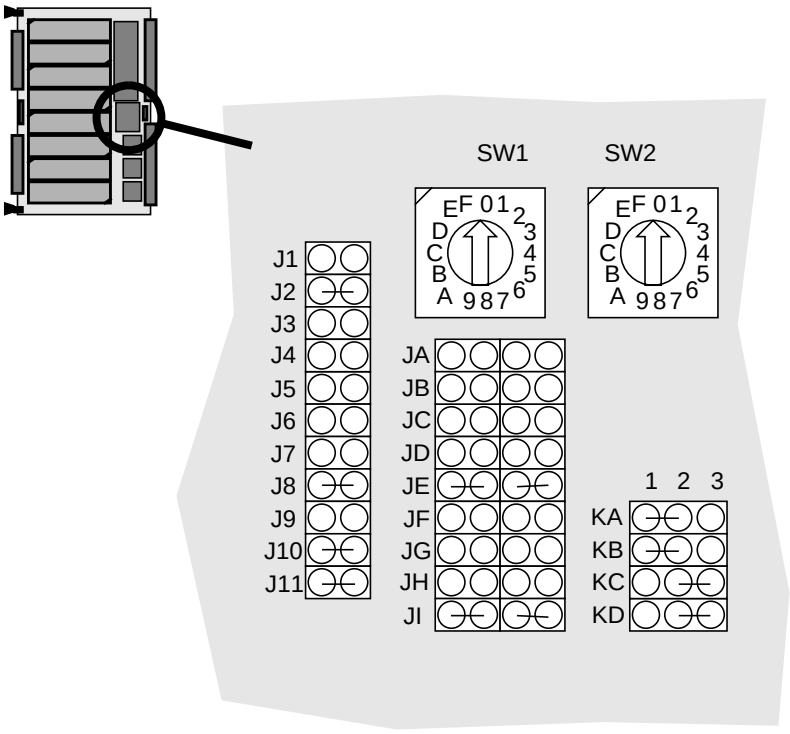


Figure 72. Subsystem Connections

8.4 Board Setup

This section presents a detailed summary of the positions and functions of the jumpers and switches used to configure the board.



SW1 and SW2 are shown in their revision C positions. On earlier pcb revisions they are approx 25mm higher.
Default jumper settings shown on J2, J8 etc.

Figure 73. Jumper and Switch Locations

As shown in figure 73, J1-J11 are 2 pin jumpers, JA-JI are 4 pin (requiring two jumpers to be fitted side-by-side) and KA-JD are 3 pin, requiring a jumper to be fitted between the central pin and the left or right or right hand pin. SW1 and SW2 are sixteen-position rotary switches.

8.4.1 VMEbus Interface

The VMEbus interface can be disabled (to remove the board from the address space, and disable all interrupts), by removing jumper J2:

Jumper	in/out	Description
J2	out	VMEbus interface disabled
	in	VMEbus interface enabled (default)

Table 36: VME interface enable jumper

The address of the board is set using the rotary switches SW1 and SW2, which select the most significant 8 bits of the 16-bit address. If SW1 is set to *X* and SW2 is set to *Y*, on revision C and later the base address in hex is `0x`*XY*`00` and on earlier revisions is `0x`*YX*`00`

8.4.2 Link Speed Configuration

The speeds of the transputer links are set as follows:

Jumper	Function	in	out (default)
J1	VMEbus link speed	10 MBits/s	20 MBits/s
J3	C004 & T2 link speed	10 MBits/s	20 MBits/s
J4 & J5	Slot 0 link speed	10 MBits/s	20 MBits/s
J6 & J7	Slots 1 to 7 link speed	10 MBits/s	20 MBits/s

Table 37:

Note that J4 and J5 must have the same setting, and so must J6 and J7 have the same setting. The default is to have all links running at 20 MBits/s.

8.4.3 Control Configuration

Figure 72 shows the control circuit for the TRAM slots, and the functions of the jumpers J8, J9, J10 and J11. This information is tabulated below

J8	J9	Description
in	don't care	Slot 0 is controlled from the VMEbus (default)
out	in	Slot 0 is controlled by the P4 connector up port
out	out	Slot 0 is controlled by the P2 connector up port

Table 38: Slot 0 control jumpers

Jumper	in/out	Description
J10	out	Down port on P2
	in	Down port on P5 (default)

Table 39: Down port selection jumper

Jumper	in/out	Description
J11	out	Slots 1 to 7 are controlled from the Slot 0's subsystem port
	in	Slots 1 to 7 are controlled from the same source as Slot 0 (default)

Table 40: Slot 1 to 7 control jumper

Unlike most transputer motherboards, which have up, down and subsystem control ports, the TMB14 has only up and down. The up port can be switched between P2 and P4, and the down port can be switched between P2 and P5. The down port always carries the same controls signals as slots 1 to 7, so it is not possible to construct a hierarchical control structure with more than one board.

8.4.4 Link configuration

Jumpers JA to JI act as a link patch area, allowing a variety of hard-wired link connections to be made, as follows:

Jumpers in	Jumpers out	Description
JA	JF, JI	T2 link 1 connected to ConfigUp
JB	JC, JD, JE, JF	VMEbusLink connected to P4 Connector Link
JC	JB, JE, JG	Slot 0 link 0 connected to P4ConnectorLink
JD	JB, JE, JF	VMEbusLink connected to C004 A link 31
JE	JB, JC, JD, JF, JG	VMEbusLink connected to slot 0 link 0 (default)
JF	JA, JB, JD, JE, JI	T2 link 1 connected to VMEbusLink
JG	JC, JE	Slot 0 link 0 connected to C004 A link 0
JH	JI	Slot 0 link 1 connected to C004 B link 0
JI	JA, JF, JH	T2 link 1 connected to slot 0 link 1 (default)

Table 41: Link jumpers

Note that, when the jumpers specified in the *jumpers in* column are fitted, all the jumpers specified in the *jumpers out* column must be removed, or incorrect operation will result.

Jumpers KA to KD allow the up and down links of the configuration pipeline to be taken to either the front or back of the board. In the left

position, the jumpers connect pin 1 to pin 2, while in the right position, they short pin 2 to pin 3.

Jumpers	Position	Description
KA & KB	left	Config down connected to P5 (default)
	right	Config down connected to P2
KC & KD	left	Config up connected to P2
	right	Config up connected to P4 (default)

Table 42: ConfigUp and ConfigDown Selection

8.4.5 Sysreset Lengthening

The power on reset length required by Transputers and associated logic is greater than that specified by the VME standard. The onboard power on reset for TMB14 can be used to drive sysreset on the VME bus by fitting J30, situated in the top right hand corner of the board.

8.5 Connector Pinouts

The following tables give the pin assignments for the TMB14's edge connectors.

Note that P2 is arranged so that standard link cables may be inserted onto the pins at the back of the VME backplane, if desired.

P4 and P5 may be fitted with "hedgehog" breakout boards, allowing standard link cables to be plugged into the front of the board. The resultant pinouts are shown in figure 76 and figure 77.

Pin	c	b	a
1	D08	BBSY*	D00
2	D09	BCLR*	D01
3	D10	ACFAIL*	D02
4	D11	BG0IN*	D03
5	D12	BG0OUT*	D04
6	D13	BG1IN*	D05
7	D14	BG1OUT*	D06
8	D15	BG2IN*	D07
9	GND	BG2OUT*	GND
10	SYSFAIL*	BG3IN*	SYSCLK
11	BERR*	BG3OUT*	GND
12	SYSRESET*	BR0*	DS1*
13	LWORD*	BR1*	DS0*
14	AM5	BR2*	WRITE*
15	A23	BR3*	GND
16	A22	AM0	DTACK*
17	A21	AM1	GND
18	A20	AM2	AS*
19	A19	AM3	GND
20	A18	GND	IACK*
21	A17	SERCLK	IACKIN*
22	A16	SERDAT	IACKOUT*
23	A15	GND	AM4
24	A14	IRQ7*	A07
25	A13	IRQ6*	A06
26	A12	IRQ5*	A05
27	A11	IRQ4*	A04
28	A10	IRQ3*	A03
29	A09	IRQ2*	A02
30	A08	IRQ1*	A01
31	+12V	+5V STDBY	-12V
32	+5V	+5V	+5V

Table 43: P1 connector pinout

Pin	c	b	a
1	GND	+5V	GND
2	nc	GND	nc
3	BackConfigUpOut	RESERVED	BackConfigDownOut
4	BackConfigUpIn	A24	BackConfigDownIn
5	GND	A25	GND
6	GND	A26	GND
7	nc	A27	nc
8	P2Link0Out	A28	P2Link1Out
9	P2Link0In	A29	P2Link1In
10	GND	A30	GND
11	GND	A31	GND
12	nc	GND	nc
13	P2Link2Out	+5V	P2Link3Out
14	P2Link2In	D16	P2Link3In
15	GND	D17	GND
16	GND	D18	GND
17	nc	D19	nc
18	P2Link4Out	D20	P2Link5Out
19	P2Link4In	D21	P2Link5In
20	GND	D22	GND
21	GND	D23	GND
22	nc	GND	nc
23	P2Link6Out	D24	P2Link7Out
24	P2Link6In	D25	P2Link7In
25	GND	D26	GND
26	GND	D27	GND
27	nc	D28	nc
28	notBackUpReset	D29	notBackDownReset
29	notBackUpAnalyse	D30	notBackDownAnalyse
30	notBackUpError	D31	notBackDownError
31	GND	GDN	GND
32	GND	+5V	GND

Table 44: P2 connector pinout

Pin	Description
1	12V
2	-12V
3	GND
4	+5V

Table 45: P3 User power connector pin assignments

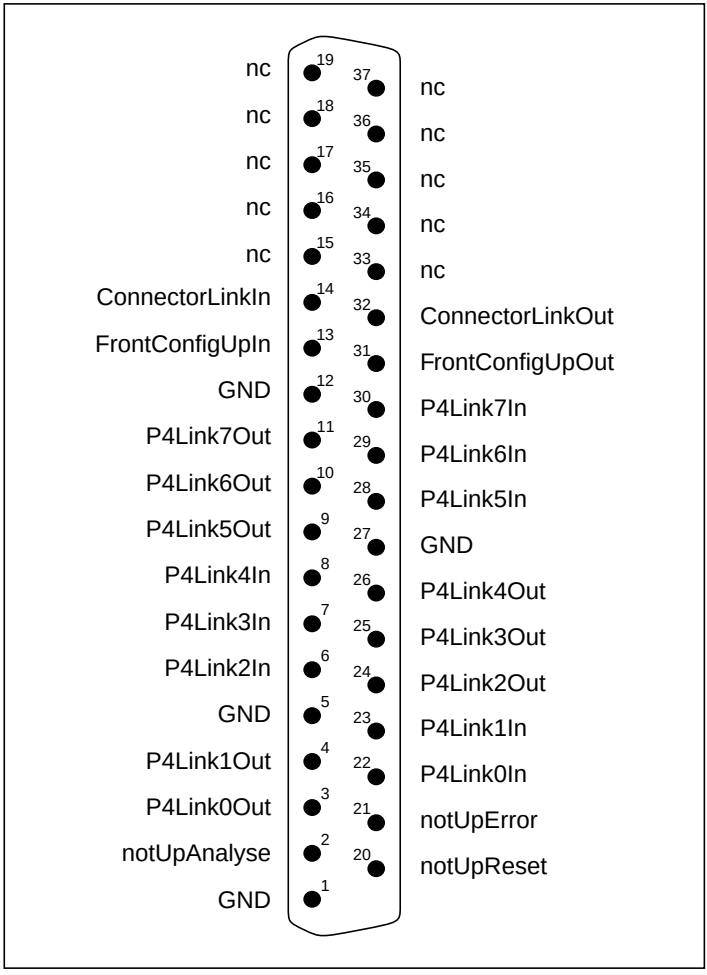


Figure 74. P4 (top) D-type pinout

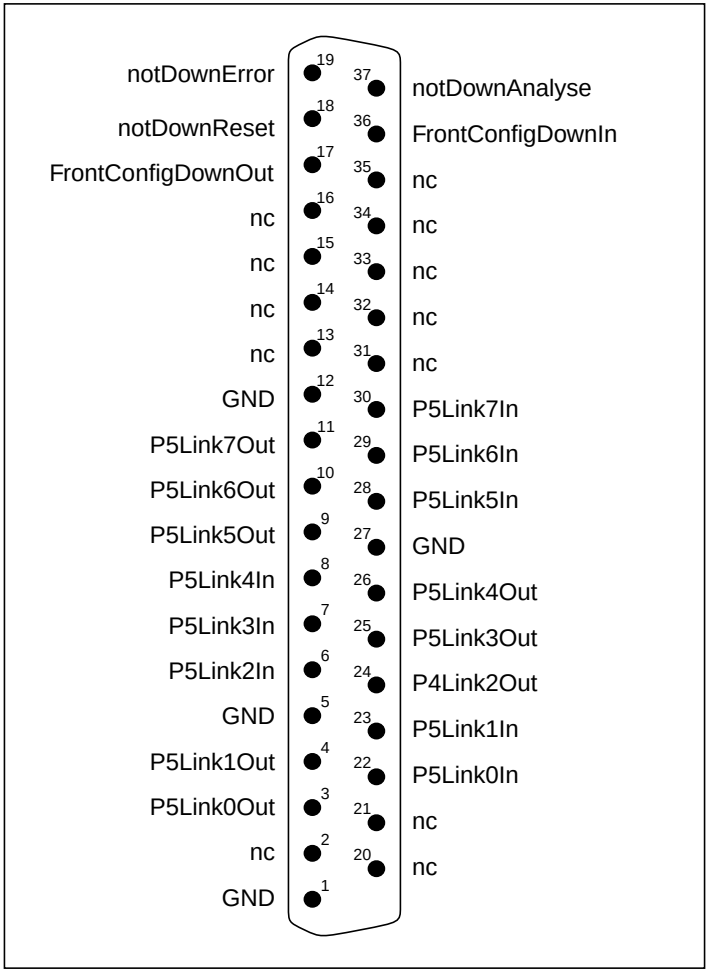


Figure 75. P5 (bottom) D-type pinout

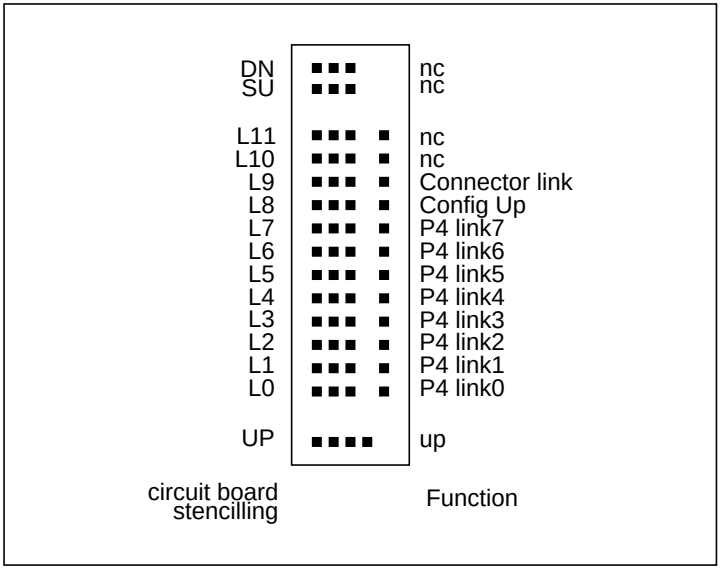


Figure 76. Connections on P4 (top) break out board

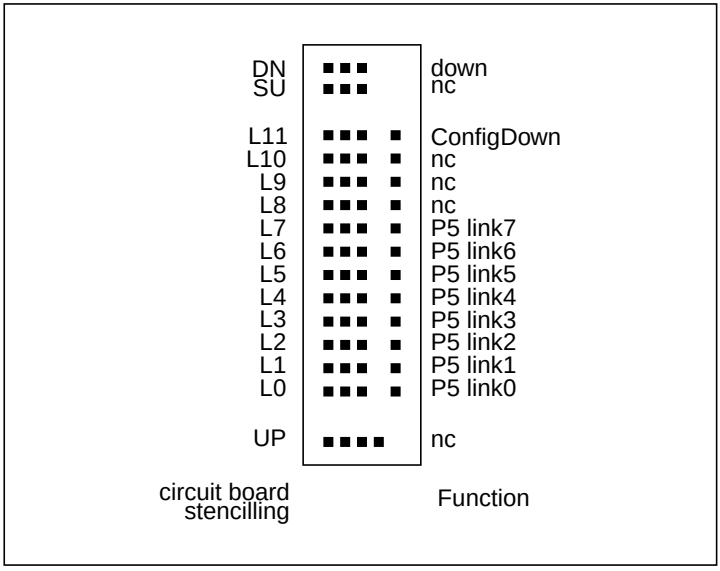


Figure 77. Connections on P5 (bottom) break out board

8.6 Programming

Table 46 summarizes the TMB14 registers, further details of which may be found in section 8.2 on page 93. These appear at the offsets shown from a base address set by SW1,SW2. The base address is #XY00 in the short address space where X is set by SW2 and Y by SW1.

The board responds to address modifier codes #29 and #2D, i.e. short user and supervisor accesses.

Offset (hex)	Register
#01	C012 input data register
#03	C012 output data register
#05	C012 input status register
#07	C012 output status register
#09	subsystem reset/error register
#0B	subsystem analyse register
#0D	interrupt enable register
#0F	interrupt level register
#11	interrupt status/ID register
#13	TRAM error register

Table 46: Register address map

Slave cycles other than odd byte read and write will cause a bus error, as recommended by the VMEbus specification.

Chapter 9
The TMB16 Motherboard

This chapter describes the hardware of the TMB16 motherboard. The description covers the operation of the host interface, details of the link configuration system and details of the various board configuration options.

9.1 Overview

The TMB16 is a full length PC hosted TRAM motherboard, with space to plug in up to ten Transputer Modules.

The TMB16 is has an IMSC004 link switch, which provides for electronic configuration of user defined processor topologies. The switch is flexible enough to allow any TRAM's link 0 or 3 to be connected to any other TRAM's link 0 or 3 or any one of eight edge connectors.

The TMB16 has a further reconfigurability option (via the patch area) which allows it to be the master board in a multi board system, or a slave board in a multi board system. It is also possible to arrange for the board to act as a high speed interface between a PC and an external transputer network in a way which doesn't require there to be any TRAMs on the board.

The TMB16 has an advanced 16bit interface to the host PC giving raw data transfer rates between the PC and the TMB16 of up to 4.9MBytes/sec. In practice, all of this data has to pass down a single transputer link, which limits the available bandwidth (from PC to module0 link0) to around 1.2 MBytes/s.

Whilst giving the option of using the 16bit interface, the TMB16 provides full backwards compatibility with the older B008 & B004 8 bit interfaces. The performance of the TMB16 running in 8bit mode is the same as that achieved by the TMB08 motherboard.

Note that to exploit the 16bit interface of the TMB16 requires the PC's Intel processor to support certain key instructions. These instructions are not supported by the Intel 8086, and hence the TMB16 cannot be used in 16bit mode on XT class machines.

Figure 78 shows the TRAM layout of this board.

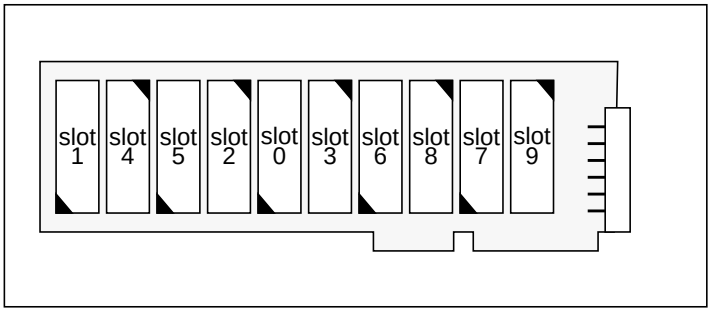


Figure 78. TMB16 TRAM layout

Figure 79 shows the location of the switches, patch area and connectors of the board.

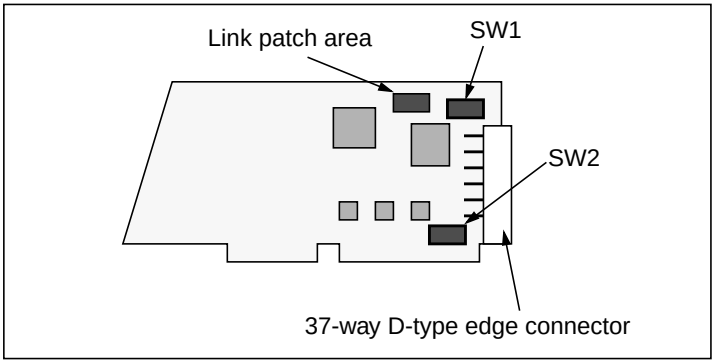


Figure 79. TMB16 board layout

9.2 Network Configuration

This section provides an overview of network configuration on the TMB16. It describes the wiring of the electronic link switch, the patch area and shows the relationship between these and the edge connector.

9.2.1 Electronic Link Configuration

This section describes the organization of the electronic link switch, the IMSC004.

In this application the C004 is used simply as a crossbar switch. The device is connected to 29 links, and can switch any link connected to any other link connected.

The links connected to the C004 are:

- link0 of all TRAM slots except module0 (module0 link0 is connected to the host PC via the T2),
- link3 of all TRAM slots,
- eight edge connectors,
- two spare links, which are taken to the patch area.

The connections made by the C004 are controlled by the T2 transputer acting in the role of a configuration processor. The T2 is, in turn, controlled via ConfigUp (which by default is wired to module0 link1 through the patch area for this purpose). Currently, the only program which can be booted onto the T2 to alter the C004 switch connections is the NCS.

The C004 connections are shown in figure 80.

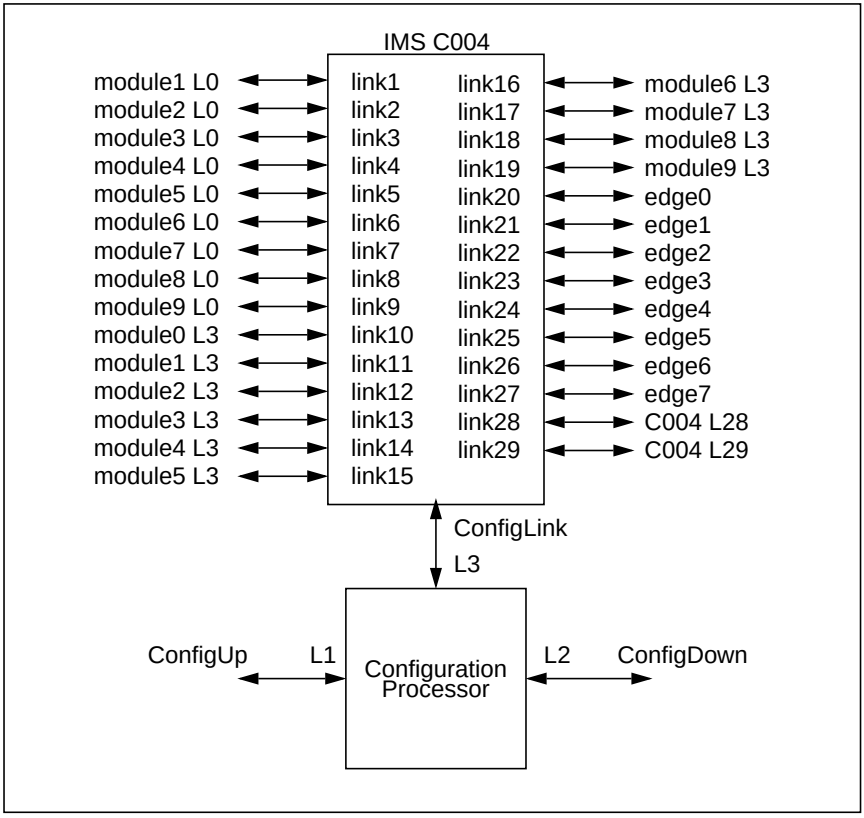


Figure 80. C004 Wiring

9.2.2 The Link Patch Area

The link patch area is in the upper right corner of the board and is labelled on the silk screen printing. It consists of 12 zero profile sockets which may be connected together with small wire jumpers.

The patch area is provided for the more permanent aspects of network configuration, principally, the setting up of the TMB16 as a master board (root processor is connected to the host) in a multi board system or a slave board (root processor is connected to another TRAM motherboard, via PipeHead) in a multi board system. To this end, six transputer links are brought out to the patch area:

- two links from the C004 crossbar switch (C004 L28/29)
- two links from the edge connector (patch0/1)
- the configuration link into the T2 (T2 link1), and
- the root module's configuration link (link1 of board slot0).

Figure 81 and figure 82 show the link patch area and the connections which need to be made to use the TMB16 as a master or slave board.

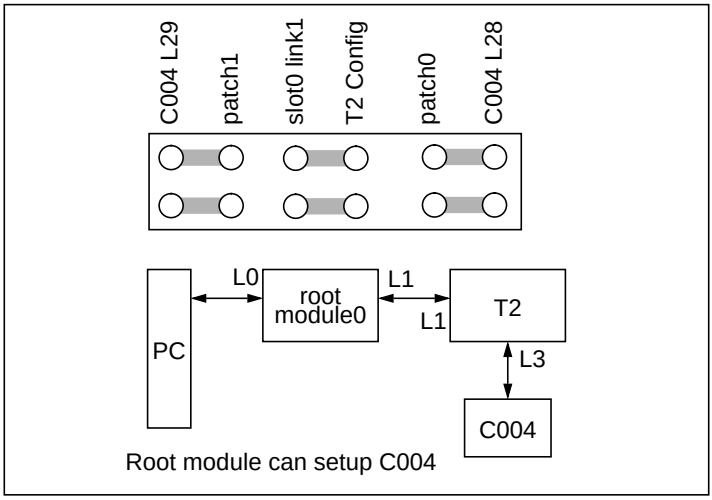


Figure 81. TMB16 Patch Area, connections for master board

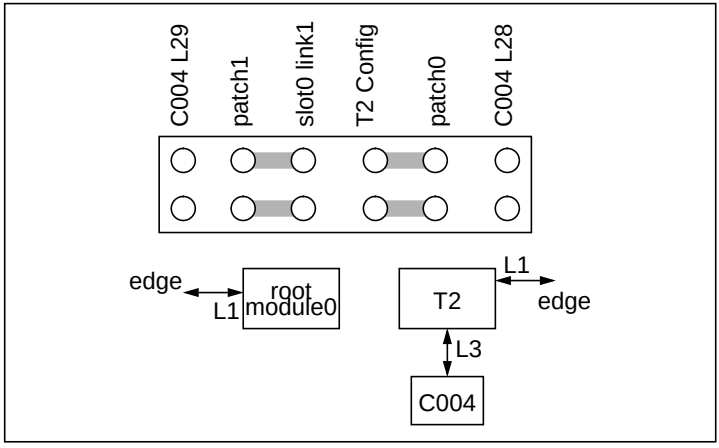


Figure 82. TMB16 Patch Area, connections for slave board

When the TMB16 is configured as a master board, the T2's configuration link (link1) is taken to PipeHead (module0 link1). The patch links (from the edge connector) are wired to the spare C004 positions.

When the TMB16 is configured as a slave board, the T2's configuration link is taken to the edge connector (patch0) and would normally be connected to ConfigUp. Also, the board's PipeHead (module0 link1) is taken to the edge connector at patch1.

9.2.3 Summary of Network Configuration

Figure 83 shows the interconnection between the module slots, the electronic link switch, the link patch area and the edge connector. It is included for reference.

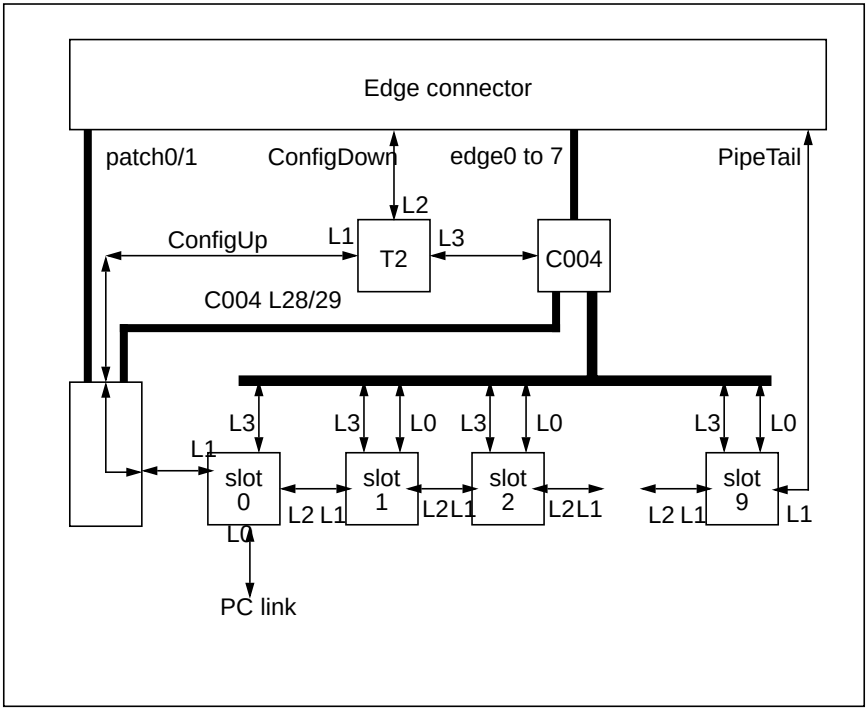


Figure 83. Network configuration summary

9.3 Board Setup

This section describes the operation of the switch banks SW1 and SW2. SW1 is located at the top of the board, near the end plate with the D-type connector, and SW2 is located at the bottom of the board, near the AT-bus connector. Each bank consists of four switches, numbered 1 to 4. When the board is held with the component side facing towards the viewer, the D-type connector to the right, and the AT-bus at the bottom, the switches are numbered from 1 on the left to 4 on the right, and the switch is on when the slider is up, and off when the slider is down. The switch numbers, and the *on* position, are marked on the switch bank.

9.3.1 Control Configuration

The board's control configuration switches, SW1.1 and SW1.2, allow the source of control for the module in slot 0 and the modules in slots 1 to 9 to be determined. The control consist of the TRAM signals reset, error and analyse.

SW1.1	Description
off	Slot 0 is controlled by the PC (default)
on	Slot 0 is controlled by the edge connector up port

Table 47: Slot 0 control selection

SW1.2	Description
off	Slots 1 to 9 are controlled from the Slot 0's subsystem port
on	Slots 1 to 9 are controlled from the same source as Slot 0 (default)

Table 48: Slots 1 to 9 control selection

9.3.2 Board Address

The I/O base address of the board can be set at either 150 hex or 200 hex using SW2.1, as follows:

SW2.1	Description
off	Base address 150 hex (default)
on	Base address 200 hex

Table 49: Board I/O address selection

9.3.3 Link speed

The speed of all the transputer links of the board can be set at either 10 or 20 Mbit/s using SW2.2, as follows:

SW2.2	Description
off	All links run at 20Mbit/s (default)
on	All links run at 10Mbit/s

Table 50: Link speed selection

9.3.4 IRQ & DMA Selection

The interrupt channel of the board may be selected using SW2.3 and 2.4 as follows:

SW2.3	SW2.4	Description
on	off	IRQ channel 3 (default)
off	on	IRQ channel 7
off	off	Interrupts disabled

Table 51: Interrupt selection

The DMA channel of the board is fixed at channel 1.

9.3.5 Reserved switches

Switches SW1.3 and 1.4 are used for factory setup and testing of the board. In normal use they must both be off.

9.4 The Edge Connector

On the right hand edge of the TMB16 is a 37-way D-type edge connector used for connection to other motherboards. For this purpose the following are brought out:

- the three control ports and the configuration link,
- eight transputer links from the C004, and
- two transputer links from the link patch area.

Figure 84 shows the pin-out of the edge connector.

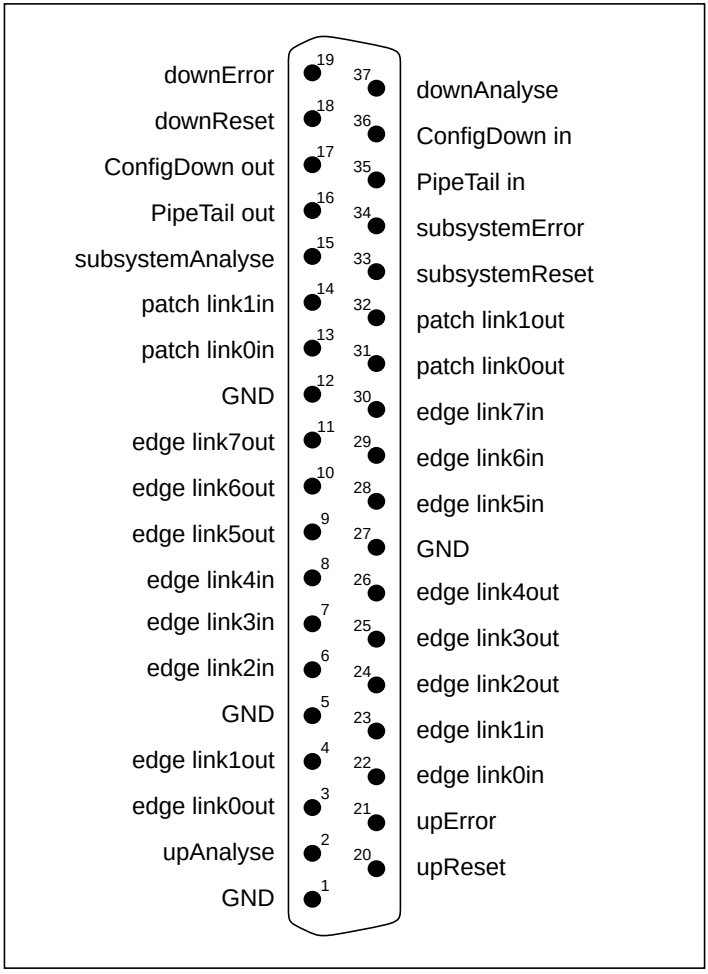


Figure 84. D-type pinout

Included in the accompanying cable pack is a mini-backplane board (sometimes called a “hedgehog” or “break-out board”) which plugs into the edge connector and brings out the various links and ports onto standard connectors which accept link cables and reset cables. The pinout of this connector is shown in figure 85.

.

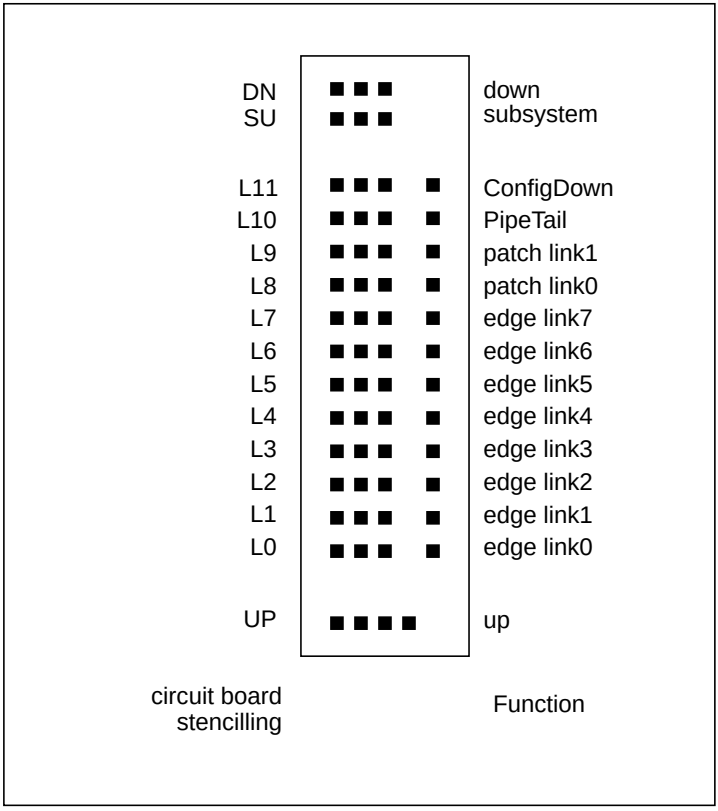


Figure 85. Connections on break out board

9.5 Examples

This section gives the recommended settings and cabling for use in three common configurations: stand-alone (where the TMB16 carries transputer modules and is not connected to other transputer cards), multi-board (where more than one TMB16 is used), and link-adaptor (where the TMB16 is used as an interface between the PC and an external transputer board, without any TRAMs on the TMB16).

9.5.1 Stand-alone

The default settings of the TMB16 are designed for stand-alone operation in a PC. The board is controlled by the PC, and is not connected to other transputer boards.

The switches and patch area should be set as shown in figure 86.

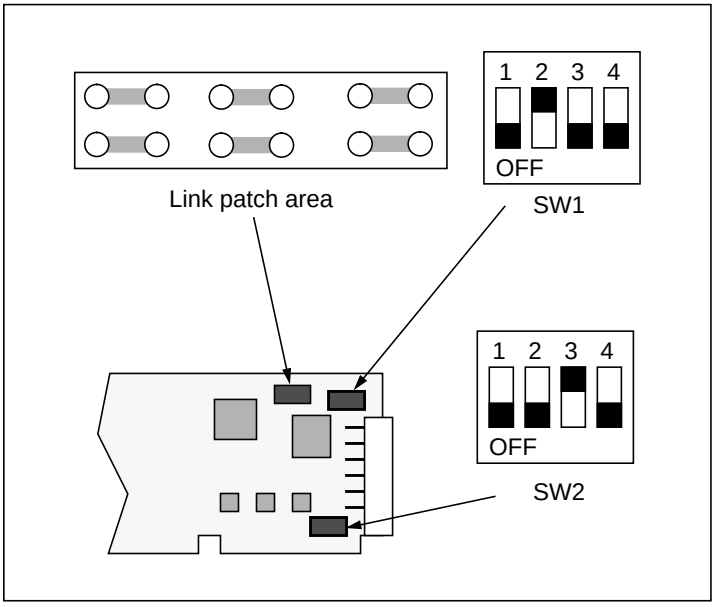


Figure 86. Recommended setup for stand-alone operation

9.5.2 Multiple TMB16s

This examples shows how to connect two TMB16s together to build a larger transputer system. The first board is used as the master, is controlled from the PC, and provides the PC link interface, whilst the second board is the slave.

The switches and link patch area for the first board should be set as shown in figure 86, above, and the second board should be set up as shown in figure 87. Both boards should be inserted into the PC, and hedgehog breakout boards fitted to their D-type connectors. Figure 88 shows the required wiring made with link cables and a reset cable between the two boards.

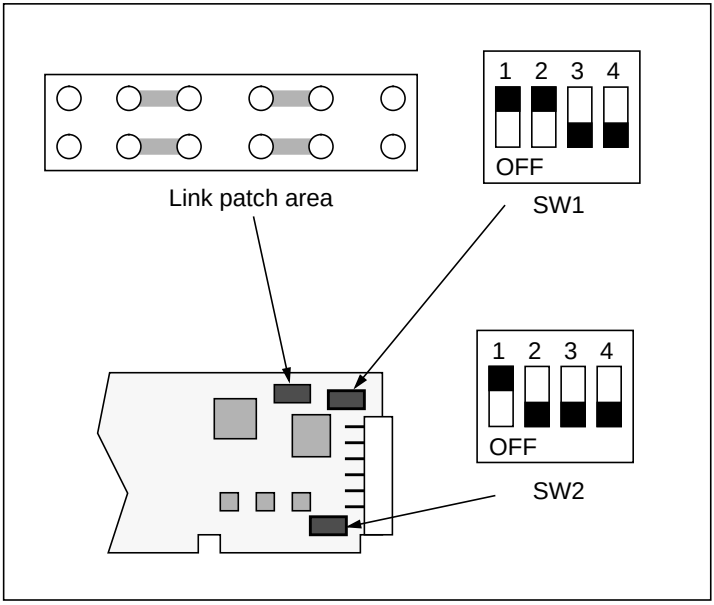


Figure 87. Setup for slave in multi-board operation

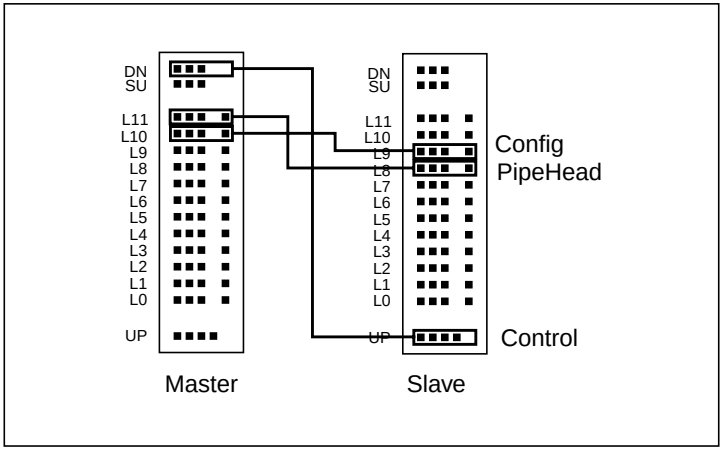


Figure 88. Connections between Master and Slave

Figure 88 shows the minimum link connections between the boards. Additional connections may be made between the edge links, and configured to the TRAMs using the C004, to provide the desired transputer network topology.

9.5.3 Link Adaptor

In this configuration, the TMB16 is used as a link adaptor card, with no TRAMs on-board, but providing a link between the PC and external transputer equipment.

The C004 should be removed from its socket, and wired links made as shown below (it is best to solder these onto a pin grid array socket, which can be inserted into the C004 socket on the TMB16)

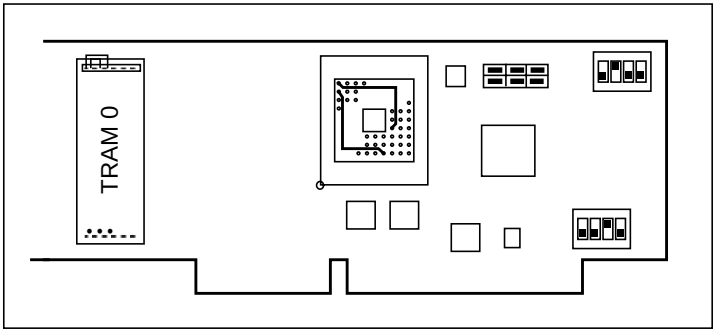


Figure 89. Configuring the TMB16 as a link adaptor

Secondly a pipe jumper should be inserted into TRAM site 0 on the TMB16, not in the normal place for a pipe jumper, but on the opposite side of the board, as shown above.

Then attach the TMB16 Patch 0 link to the host link of the external equipment, and the TMB16 reset Down to the reset Up of the external equipment.

9.6 The Host Interface

The TRAM standard defines that TRAM motherboards have on them a T2 transputer to control the board's IMS C004 electronic link switch. In older designs the T2 was used solely for this purpose and was therefore rather under-utilized. In the TMB16, a portion of this transputer's external memory space is mapped directly to the PC. This allows the transputer to communicate with the PC's Intel micro-

processor directly and at the high bandwidth supported by the PC's backplane. This design fully exploits the presence of the T2, which is in its own right a powerful 16bit processor.

Since the external memory interface of the T2 is 16bits wide, the TMB16 interface transfers 16bit quantities (words) at a time. The PC's backplane is normally clocked at 8MHz. Given that it requires three clock cycles to transfer a word across the backplane, this gives a theoretical backplane bandwidth of (8/3) MWords/sec or 5.33MBytes/sec. Measurements show that the TMB16 can transfer data across the backplane at 3.5MBytes/sec at this bus speed, realizing in practice 70% of the available bandwidth.

The following three subsections describe the operation of the host interface in detail.

9.6.1 Operation of the Hardware

The PC's Intel processor sees the TMB16 as two 16 bit words (ports) in the PC's I/O map. One of the ports is for communications from the PC to the T2 transputer (write port) and the other for communications from the T2 transputer to the PC (read port).

During data transfers from the PC to the T2 transputer, the PC simply writes words to the write port as fast as it can. Using the Intel processor's string write instruction (80286 and later), accesses to the I/O port can be done at full bus speed. This is because the write operation is performed entirely in microcode.

During the same data transfer, the T2 transputer reads data from a block in its external memory. The entire block is memory mapped so that every location in the block maps to the write port in the PC's I/O

space. Hence, the actual reads the transputer performs are reads from the write port. Figure 90 illustrates this operation.

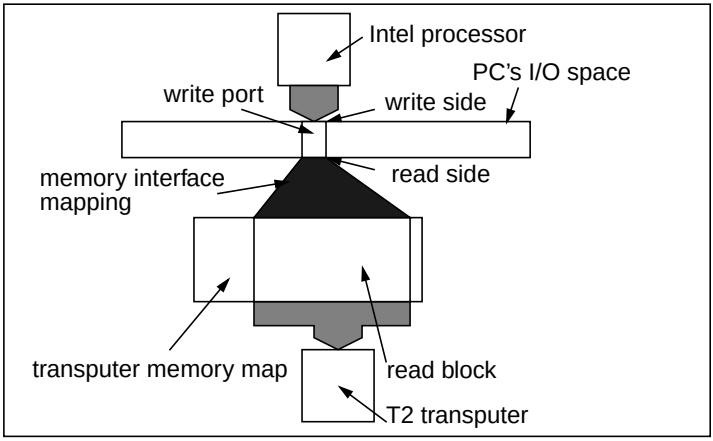


Figure 90. Inter processor communications (write)

For communications from the transputer to the PC a similar process occurs: the transputer writes to a block of memory in its external memory space, this memory is folded by hardware to a single location in the PC's I/O space (the read port) which the PC reads with a string read instruction.

Clearly the operation of the interface depends on synchronization between the two processors. When the Intel processor writes to the write port it cannot perform another write until the transputer has performed a read operation. Similarly, when the transputer writes to the read port, it cannot perform another write until the Intel processor performs a read. This synchronization is achieved via the respective processor's wait line.

9.6.2 Memory Maps

The operation of the interface depends on the folding of half of the transputer's external memory space to two objects in the PC's I/O space. Figure 91 shows the details of the memory map.

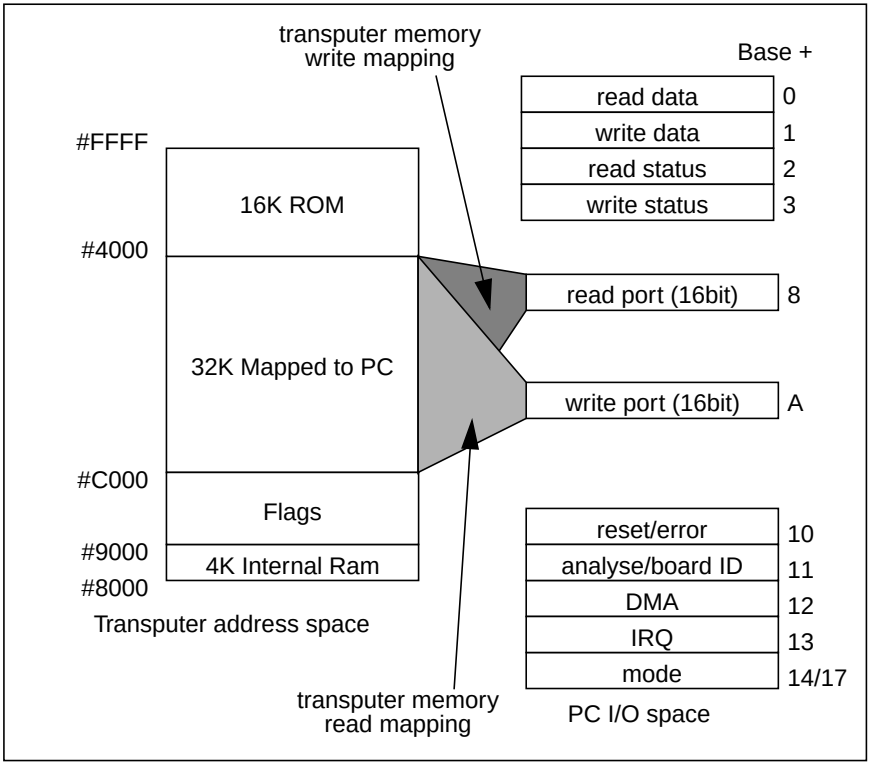


Figure 91. System memory maps

Notes:

- The TMB16 has a flags register which is used during B004 interface emulation. Essentially the register looks like a C012 as far as the PC is concerned. On the transputer side it is mapped

to all locations between #9000 and #BFFF. Table 52 shows the bit allocations.

Bit	Function
0	Data ready to send
1	Data ready to receive
2	1 ⇒ Byte mode, 0 ⇒ string mode

Table 52: TMB16 Flags register

- The transputer has 32KBytes of its memory space mapped to the PC's I/O space. The actual mapping depends on whether the transputer is reading or writing to the memory space.
- The base address of the registers in the PC's I/O space can be set to #150 (default) or #200.
- The PC's I/O space contains registers for DMA & IRQ for compatibility with the B008 style of interface. The TMB16 supports DMA channel 1 and IRQ channels 3 (default) & 7.
- Writing to the mode register in the PC's I/O space determines whether the TMB16 works in full 16bit mode (bit0 set to 0) or B004/B008 compatibility mode (bit0 set to 1). Note that the address of the mode register is either 14 or 17, depending on the contents of the board ID register (see below).
- Reading from the board ID register (base address + 0x11) returns a value as follows

Bit	Function
0	Processor type: 0⇒T222, 1⇒T225
1	Mode reg. offset: 0⇒0x14, 1⇒ox17
2, 3	Reserved: read as 0
4-7	Undefined

Table 53: TMB16 Board ID register

9.6.3 Operation of the Software

The functioning of the interface is controlled by software running on the T2. The software is provided on the TMB16 in non-volatile memory (EPROM). Essentially, the software exploits the available hardware to provide an invisible connection between the PC and the transputers mounted on the motherboard. So as to maintain backwards compatibility with older motherboards, this connection is actually between the PC and link0 of TRAM module0.

At the beginning of any data communication the two processors must agree on the amount of data to be transferred. This is achieved by transmitting a single 16 bit counter. That counter determines the number of bytes that follow. In occam, this protocol may be defined as: `INT16::[]BYTE`

i.e. the same as the iserver protocol. Note that the hardware limits the maximum amount of data that can be transferred in one block to 32KBytes (see the memory map).

As far as the T2 transputer is concerned, access to the PC is achieved by reading/writing its external memory. During a data transfer from the PC to the transputer network the T2 is reading from its external memory and writing data to one of its links. This is achieved with the transputer channel `out` instruction. Similarly, during a data transfer from the transputer network to the PC, the T2 is reading data from a link and writing this data to its external memory, which is achieved with the transputer channel `in`

instruction. Figure 92 summarizes the key software instructions used.

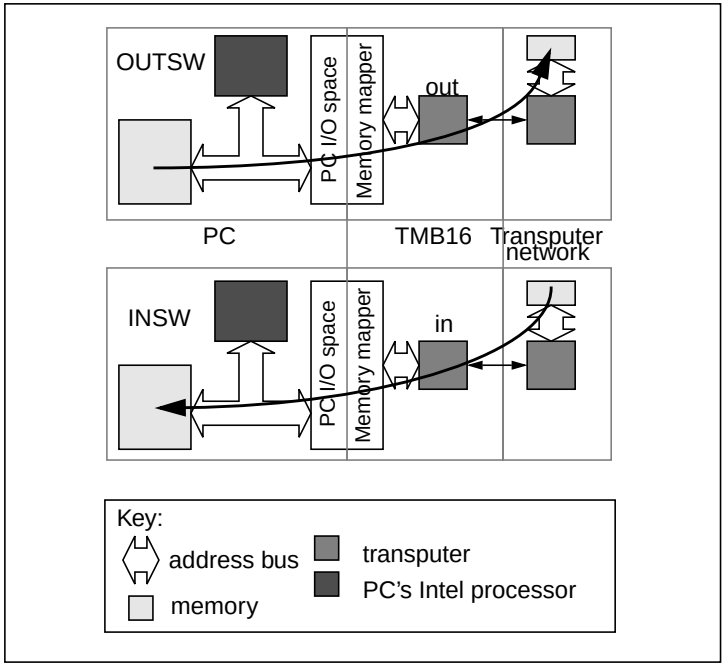


Figure 92. Key software instructions

Chapter 10

The TMB17 Motherboard

This chapter gives a detailed hardware description of the TMB17. The various features of the board are described and some examples of configuration are given.

10.1 Overview

The TMB17 is a full length PCI hosted TRAM motherboard, with space to plug in up to ten Transputer Modules.

The TMB17 is shipped with an IMSC004 link switch, which provides for setting up user defined topologies. The switch is flexible enough to allow any TRAM's link 0 or 3 to be connected to any other TRAM's link 0 or 3 or any one of eight edge connections.

The TMB17 has a link patch area which, amongst other things, allows *PipeHead* and *ConfigUp* to be connected together.

Figure 93 shows the TRAM layout of this board for reference.

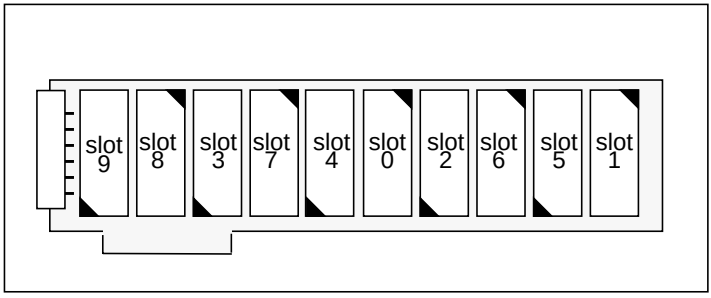


Figure 93. TMB17 TRAM layout

Figure 94 shows the board layout for reference.

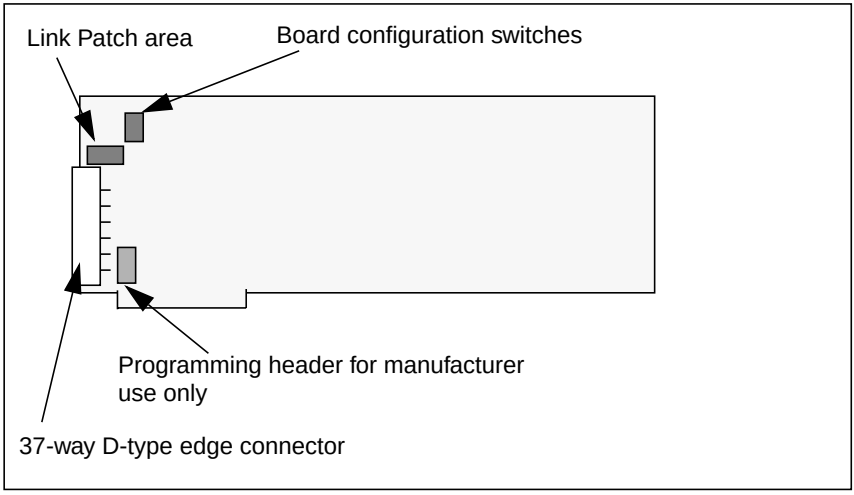


Figure 94. TMB17 board layout

10.2 Windows 95

When Windows 95 is run for the first time after installing a TMB17, a New Hardware Found dialog box is displayed. You should proceed as follows:

1. Click on the Driver from disk radio button and click on OK.
2. An Install From Disk dialog box will appear. Put the supplied CD-ROM in the drive and use the Browse dialog to select the file `tps.inf` in the CD-ROM directory `win95`.
3. Click on OK.

After installation, the I/O base address and the interrupt used by the TMB17 can be determined or altered using the System control panel as follows:

1. From the Start menu, show the control panel by selecting it from the Settings menu.
2. Start up the System control panel.
3. Select the Device Manager page.
4. Under Transtech devices, select the TMB17 entry.

5. Select the Resources page.

10.3 PCI Interface

10.3.1 Hardware Description

Figure 95 is a block diagram of the PCI to Transputer link interface. The PCI controller combines plug and play configuration capabilities with an B004/B008 compatible register map so that standard software works correctly. The FIFO shown between the PCI controller and the link adapter is transparent to this software, but provides an additional high performance interface. This decouples the C012 link bandwidth limitations from the, much faster PCI bus. The high performance interface consists of 32bit access to the FIFO and transmit and receive FIFO level registers. This allows complete packets of information to be sent or received at PCI bandwidth freeing up host processor power.

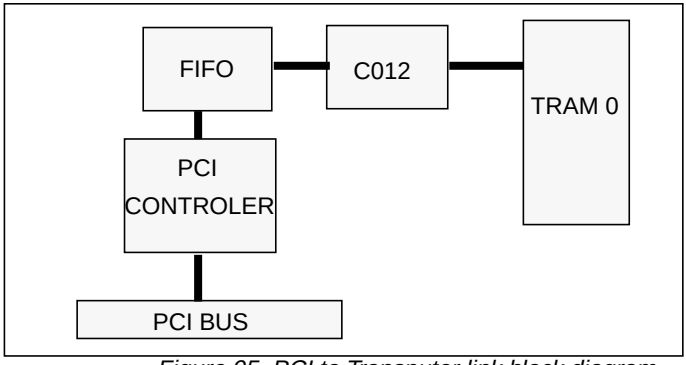


Figure 95. PCI to Transputer link block diagram

10.3.2 Register Map

Table 54 lists the registers available and their offset from the board base address. The base address is set by the PCI Bios routines at host reset time. The value can be read by reading Base Address 0 of the relevant board's configuration space via the PCI Bios routines.

Table 54: Register Map

Base + (hex)	Description	Access Type
00h	Rx Data 8 bit	Byte Read only
01h	Tx Data 8 bit	Byte Write only
02h	Rx Status	Byte Read only
03h	Tx Status	Byte Read only
08h	Rx Data 8/16/32 bit	Byte/Word/Dword Read
0Ch	Tx Data 8/16/32 bit	Byte/Word/Dword Write
10h	Reset write/Error read	Byte Read/Write
11h	Analyse	Byte Read
18h	Rx FIFO Level	Word Read
1Ch	Tx FIFO Level	Word Read

10.3.2.1 Rx Data

Rx FIFO data can be read one byte at a time from offset 00h compatible with B004 standard software. For higher performance the FIFO can be read by WORD or DWORD reads from offset 18h. The data read is in order received from the C012 link (ls byte first). To ensure valid data is read, the number of bytes available must first be determined by reading the Rx level register.

10.3.2.2 Tx Data

Tx FIFO data can be written to one byte at a time by writing to offset 01h compatible with B004 standard software. For higher performance the FIFO can be filled by WORD or DWORD writes to offset 1Ch. The data should be written in the order to be send by the C012 link (ls byte first). To ensure data is not lost, the number of bytes of available space must first be determined by reading the Tx level register.

10.3.2.3 Rx Status

A one read from bit 0 of offset 02h indicates that a byte of data is available to be read from the RX Data register.

10.3.2.4 Tx Status

A one read from bit 0 of offset 03h indicates that a byte of data can be written to the Tx Data register.

10.3.2.5 Rx Level

The value read from offset 18h indicates the number of bytes available to be read from the Rx Data FIFO. To ensure a coherent value is read a 16bit Word read should be used to read this register.

10.3.2.6 Tx Level

The value read from offset 1Ch indicates the number of bytes of space free in the Tx Data FIFO. To ensure a coherent value is read a 16bit Word read should be used to read this register.

10.3.2.7 Reset/error

Writing a one to bit 0 of offset 10h will force Reset to be sent to module 0 and the FIFO link interface, a 0 must be written to clear the Reset. Reading bit 0 will return the state of module 0 error flag.

10.3.2.8 Analyse

Writing a one to bit 0 of offset 11h will force Analyse to be sent to module 0, a 0 must be written to remove Analyse.

10.3.3 PCI Configuration

Programmers should note that Transtech's PCI SIG Vendor ID is 1278 hex. The TMB17 has a Device ID of 1001hex.

10.4 Network Configuration

This section provides an overview of network configuration on the TMB17. It describes the wiring of the electronic link switch, the patch area and shows the relationship between these and the edge connector.

10.4.1 Electronic Link Configuration

This section describes the organization of the electronic link switch, the IMSC004.

In this application the C004 is used simply as a crossbar switch. The device is connected to 30 links, and can switch any link connected to any other link connected.

The links connected to the C004 are:

- link0 of all TRAM slots except module0 (module0 link0 is always connected to the host PC),
- link3 of all TRAM slots,
- link 0 of the T2 configuration processor,
- eight edge connectors,
- two spare links, which are taken to the patch area.

The C004 is programmed via the T2 configuration processor on the board. This in turn is programmed via *ConfigUp*.

The connections are shown in figure 96 for reference.

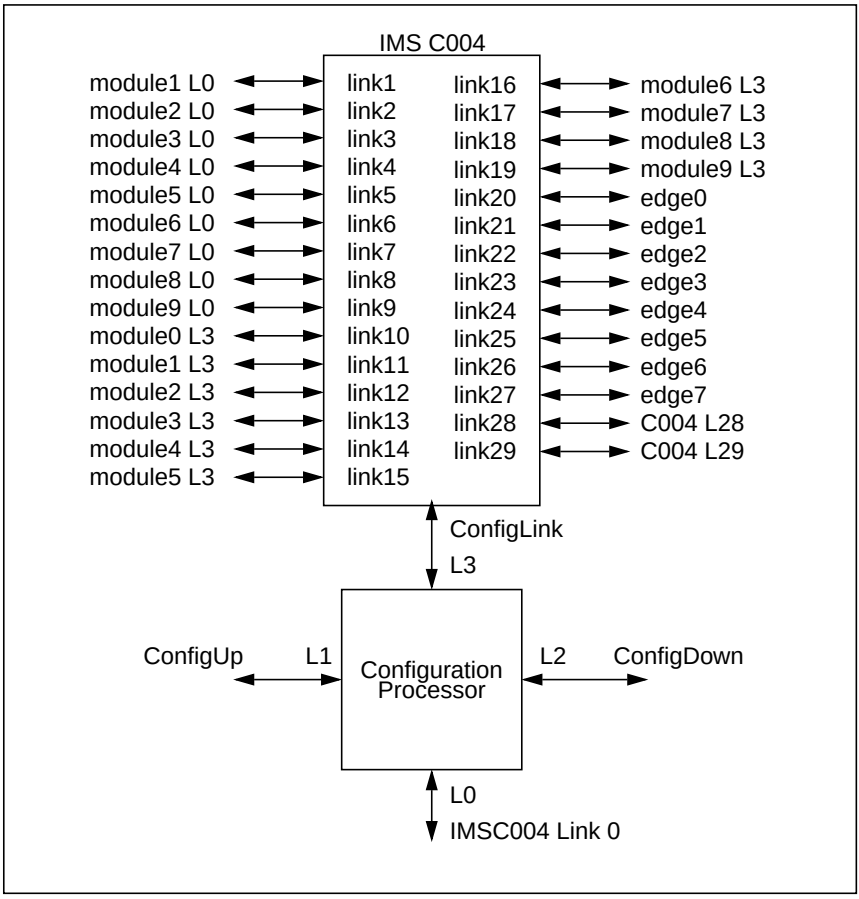


Figure 96. C004 Wiring

10.4.2 The Link Patch Area

The link patch area is a 6x2 jumper header on 0.1" spacing. The required connections can be made by moving push-on shorting links to the correct positions

The primary purpose of the patch area is to allow *PipeHead* and *ConfigUp* to be terminated correctly. Using the patch it is possible to

either connect these two together (for the first motherboard in a system) or to take them off the board (for a slave motherboard).

The links taken to the patch area are:

- *ConfigUp*, i.e., link1 of the configuration processor
- *PipeHead*. i.e., Module0 link1
- two of the links from the crossbar switch (*C004L28/29*)
- two links which are taken directly to the edge connector (*patch0/1*)

Figure 97 shows the links attached to the patch area and the default connections made when the board is shipped.

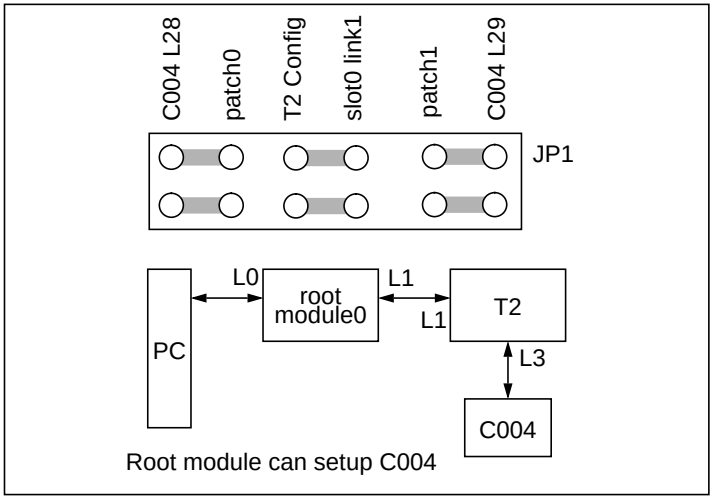


Figure 97. The Link Patch Area for Master Board

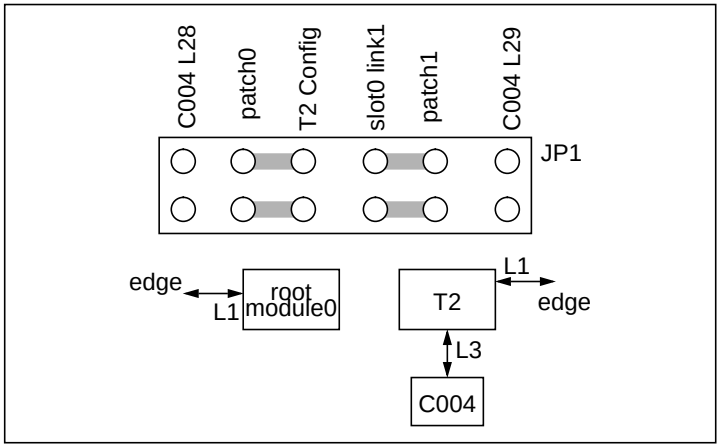


Figure 98. TMB17 Patch Area, connections for slave board

10.4.3 Summary of Network Configuration

Figure 99 shows the interconnection between the module slots, the electronic link switch, the link patch area and the edge connector. It is included for reference.

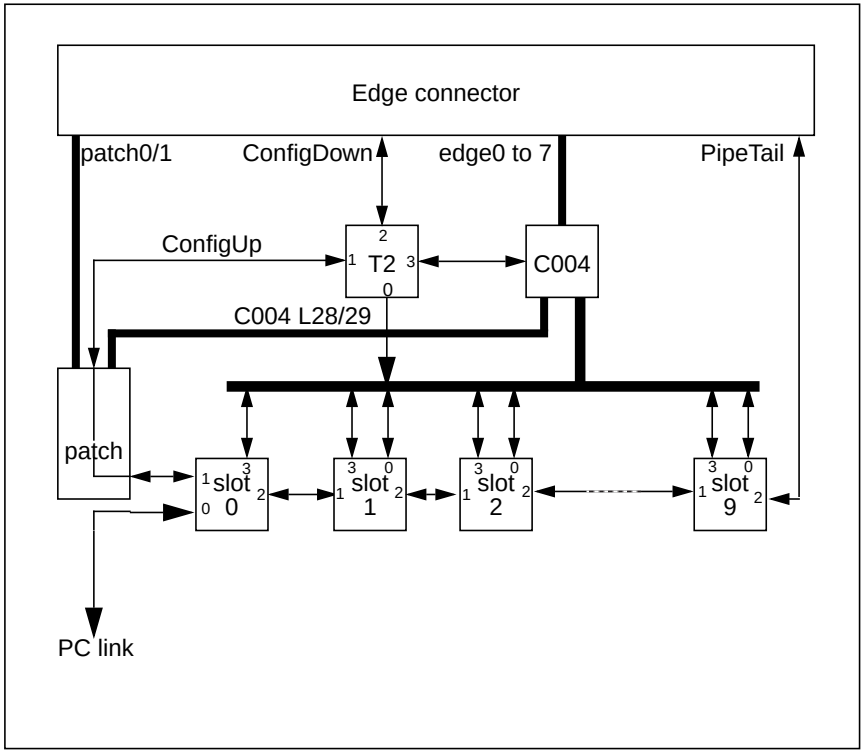


Figure 99. Network configuration summary

10.5 Description

10.5.1 Board Configuration

The basic board configuration is achieved by the use of the configuration switches. In the top left hand corner of the board there is a 4 way switch bank. Switch 1 is on the top, switch 4 bottom is not used.

The switches control the following functions:

- S1 Module link speed
- S2 Module 0 control source
- S3 Modules 1-9 control source

10.5.1.1 Link Speed

Switch 3 controls the transputer link speed. With the switch off/open (default) the links run at 20MHz. With the switch on/closed the links run at 10MHz.

10.5.1.2 Control Configuration

There are two configuration options relating to board control:

- *S2 IBM*, source of control for module0
- *S3 MD0*, source of control for modules1 to 9.

If switch S3 is on, then the source of control for modules1 to 9 is from the same source as module0. If S3 is off then modules1 to 9 are controlled from module0's subsystem.

If switch S2 is on, then the source of control for module0 is from *up* on the edge connector. If S2 is off then the source of control is the host PC.

Figure 100 summarizes for the case of the link area.

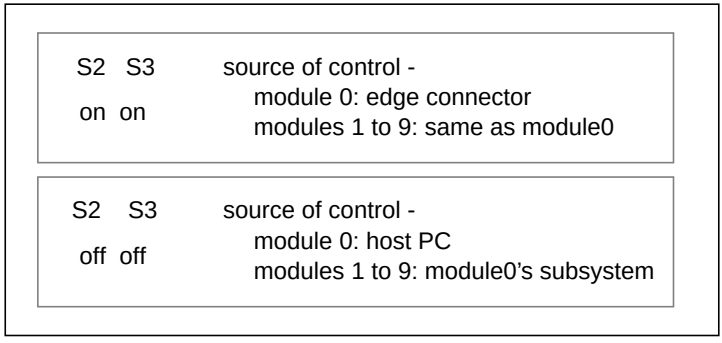


Figure 100. Control configuration (link)

10.5.2 The Edge Connector

On the left hand edge of the TMB17 is a 37-way D-type edge connector.

On the TMB17 the edge connector is used for connection to other motherboards. For this purpose the following are brought out:

- the three control ports and the configuration link,
- twelve transputer links.

Figure 101 shows the pin-out of the edge connector.

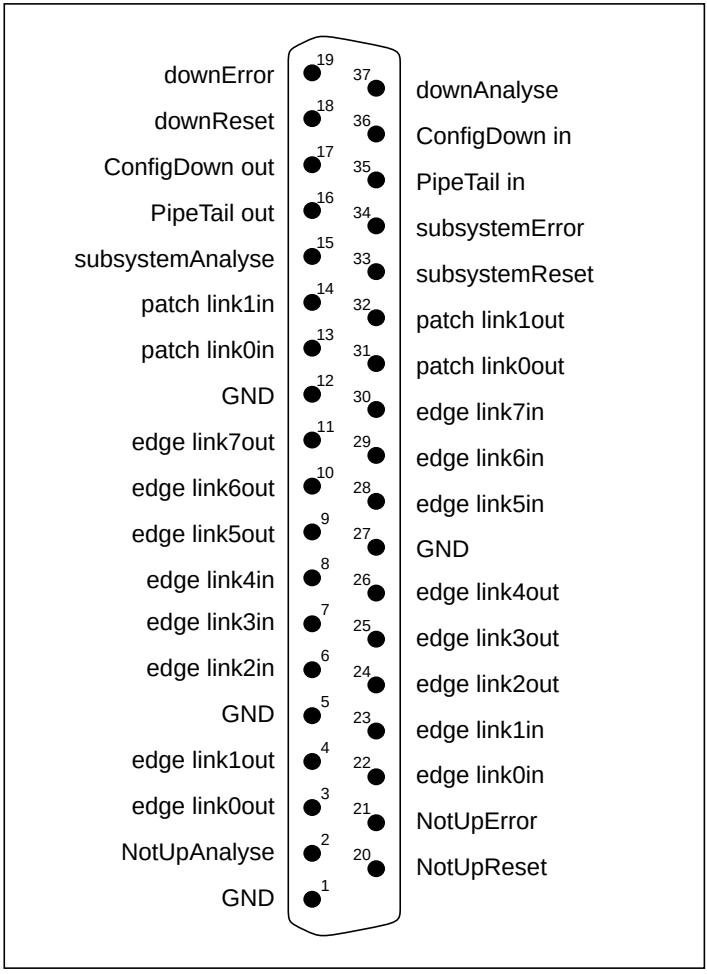


Figure 101. D-type pinout

Included in the accompanying cable pack is a mini-backplane board which plugs into the edge connector and brings out the various links and ports onto standard connectors which accept link cables and reset cables. The pinout of this connector is shown in figure 102.

.

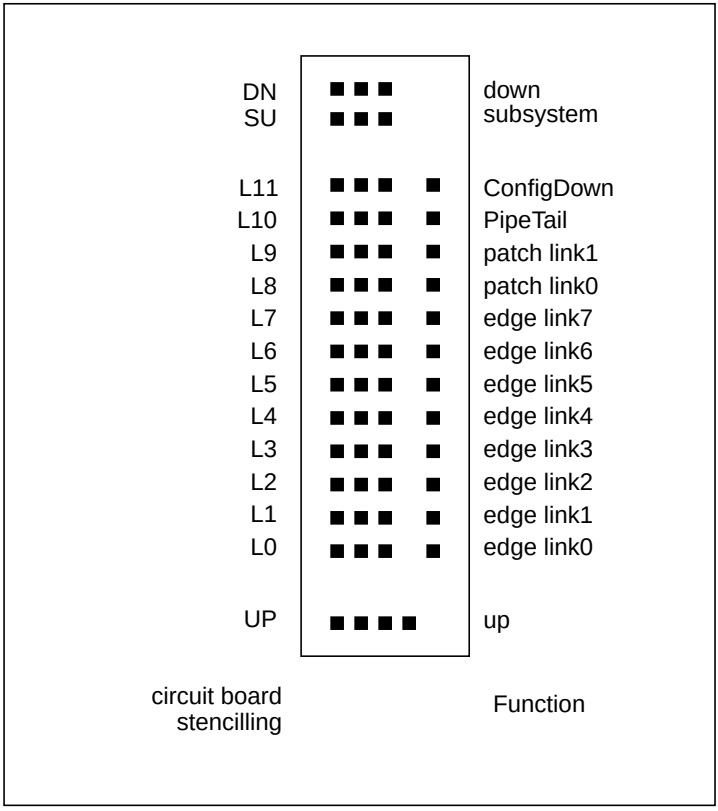


Figure 102. Connections on break out board

10.5.3 The Link Patch Area

The default wiring for the link patch area is to connect:

- *ConfigUp* to *PipeHead*,
- *patch0* to *C004 L28*,
- *patch1* to *C004 L29*.

The *patch0/1* connections allow 10 links to be brought out from the C004 to the edge connector.

The pinout of the link patch area is shown in figure 103 along with the default connections.

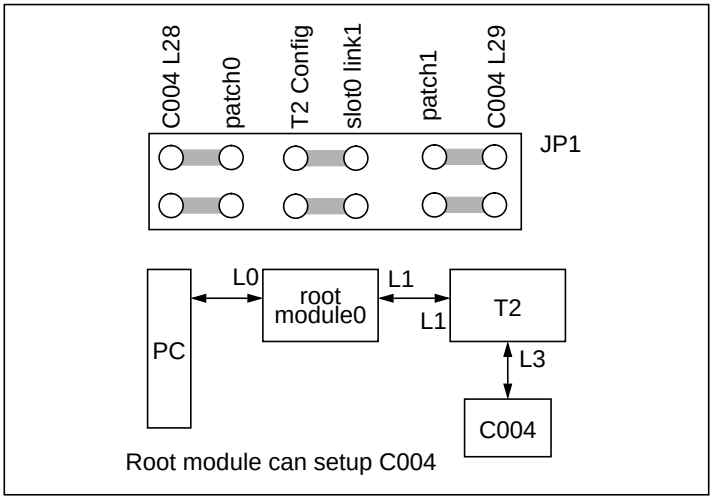


Figure 103. TMB17 Patch Area, connections for master board

10.6 Examples

This section shows how to set the board configuration options for two examples:

- a single TMB17 configured for use with an Inmos Toolset,
- two TMB17s connected as a single system, configured for use with an Inmos Toolset.

10.6.1 Stand-alone TMB17

The most common stand-alone configuration for the TMB17 is for use with the Inmos Toolsets, with every TRAM reset from the PC. To

achieve this configuration, set the link the link patch area as shown in figure 103, and make the following setting:

Switches	Setting	Description
S1	off	Use 20 Mbit/s links
S2	off	Slot 0 controlled from PC
S3	on	Slot1-9 reset as slot 0

Table 55: Default settings for stand-alone operation

10.6.2 Multiple TMB17s

As an example of connecting multiple motherboards together consider a system consisting of two TMB17s in the same PC. In this case:

- the two boards must have different addresses
- one board must be slaved to the other
- the default pipeline needs to be connected
- the configuration pipeline needs to be connected

First set up one board with the same configuration as for stand-alone operation (see above). Ensure that pipe-jumpers are used in empty TRAM slots, and in the inactive slots of any TRAMs larger than size 1, so that link 2 of the last TRAM on the motherboard is taken out to pipe tail.

Then set up the second board as follows. The link patch area should be configured as shown in figure 104.

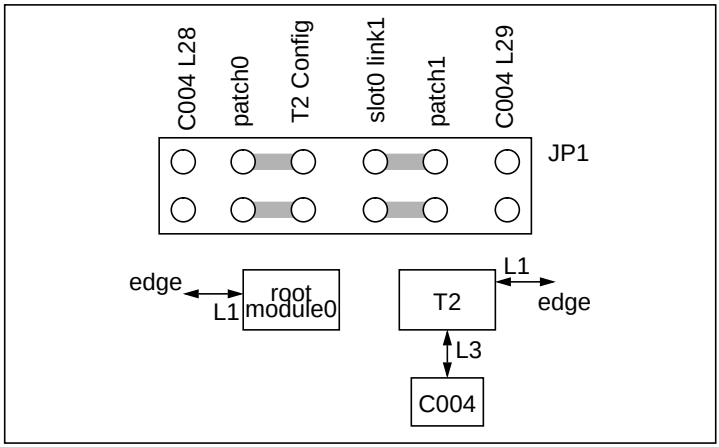


Figure 104. TMB17 Patch Area, connections for slave board

The jumpers and links on the slave board should be set up as follows:

Switches	Setting	Description
S1	off	Use 20 Mbit/s links
S2	on	Slot 0 controlled from UP
S3	on	Slots 1 to 9 controlled from same source as slot 0

Table 56: Settings for slave operation

Plug both boards into the PC and fit hedgehogs, and make the following connections between the boards:

Cable	First board	Second board
Reset Cable	Down (DN)	Up (UP)
Link Cable	Pipetail (L10)	Patch 1 (L9)
Link Cable	ConfigDown (L11)	Patch 0 (L8)

Table 57: Connections between TMB17s

Chapter 11
Utilities Software

This chapter describes the software on the Transputer Utilities CD-ROM supplied with Transtech's Transputer motherboards. The CD-ROM includes:

- Transputer network test software (`check`, `mtest`, etc.)
- `iserver`, suitable for use with the 4th generation Inmos toolset.
- Iserver file I/O cache, multiplexor, and fast I/O libraries, for the 4th generation Inmos toolset.
- Solaris 2 device driver for the TMB14.
- TMB17 support for the PC Inquest tools.
- TMB14 support for the Sun Inquest tools.
- On-line version of this manual.

11.1 PC Installation

The installation allows the software to share the Transtech top-level directory with other Transtech software packages. Installation is as follows under Windows 95:

1. Insert the supplied CD-ROM in the drive.
2. Select Run from the Start menu.
3. Use the Browse dialog to select `setup.exe` from the `win95\tmb95` directory on the CD-ROM. Click on the OK button.
4. After a short delay while the Setup program initializes, a Welcome dialog box appears. Click on the Next button.

5. Select the installation directory. Use the Browse dialog if you wish to change this from the default directory `c:\tps`. Click on the Next button.

   The selected software will be now be installed.

6. When the installation has successfully completed, a Setup Message is displayed. Click on the OK button to exit from Setup.

7. Add the following lines to `autoexec.bat` for an ISA card at I/O base address 150 hex:

```
set TPS=c:\tps
set PATH=%TPS%\bin;%TPS%
set ISEARCH=%TPS%\tp\4libs\ %ISEARCH%
set ICONDB=%TPS%\lib\dos.db
set TRANSPUTER=150
```

   For a TMB17 PCI card, use:

```
set TRANSPUTER=tmb17
```

8. Now re-boot the machine.

11.2 Solaris 2 Installation

11.2.1 FORCE CPU-3CE

If you are using a FORCE SPARCE CPU-3CE card, please read this section carefully when selecting a VME base address for each VME Transputer board such as a TMB14 to be installed.

The FORCE SPARC CPU-3CE card maps a single window of the VME address space into its processor memory map. All of the addresses used for the installed VME cards must be within this window. The PROM monitor variable `vme-a32map` should be set to the top VME address bits of the window.

Also, part of the VME memory map is occupied by memory and registers by the 3CE. You should avoid using base addresses for the VME Transputer card which conflict with these.

For a single TMB14 to be controlled by a 3CE, a VME base address of `0x0800` is often suitable.This can be achieved by setting the TMB14's VME base address switches to `0` and `8`. In this case the value of `vme-a32map` should be set to 15:

```
ok setenv vme-a32map 15
```

11.2.2 FORCE CPU-5V

When using a TMB14 in conjunction with a FORCE SPARC CPU-5V VME controller, you need to enable VME A16 accesses. To do this, set the PROM monitor prompt variable `vme-a16-master-ena?` as follows:

```
ok setenv vme-a16-master-ena? true
```

11.2.3 Software

The Transputer Utilities for Solaris 2 are supplied on CD-ROM. If the SPARC computer does not itself have a CD-ROM drive, then a networked Sun with a CD-ROM drive can be used and the CD-ROM's contents accessed using NFS.

To install the Transputer Motherboard software on SPARC systems running Solaris 2 with a CD-ROM drive:

1. Log into the Sun as root.
2. Put the supplied CD-ROM into the CD-ROM drive.
3. Use `volcheck` to mount the CD-ROM:

   ```
   # volcheck cdrom
   ```

4. Move to the `solaris2` directory on the CD-ROM:

   ```
   # cd /cdrom/cdrom0/solaris2
   ```

5. Add the Solaris 2 packages:

   ```
   # pkgadd -d ‘pwd‘ all
   ```

   When prompted for an installation directory, press Return for the default (`/opt/transtech`) or type in an alternative installation directory.

6. Eject the CD-ROM from the drive:

   ```
   # cd /
   # eject cdrom
   ```

The following lines are an example extract of the file `.login` which can be used to set these environment variables for users of the C shell under Solaris 2:

```
setenv TPS /opt/transtech
setenv ISEARCH “$TPS/tp/4libs/ $ISEARCH”
setenv ICONDB $TPS/lib/tmb.db
setenv ASERVDB $TPS/tp/4libs/aservdb
```

```
setenv TRANSPUTER tmb0
set path = ( $TPS/bin $path )
```

11.2.4 Configuration File

Before the TMB14's device driver software can be used, a configuration file needs to be edited to reflect the hardware parameters of the TMB14 motherboard or motherboards used. This configuration file is described in this section.

The file `/kernel/drv/tmb.conf` is a configuration file which is used to specify the hardware parameters of the installed TMB14 motherboard or motherboards.

Normally, for every Transtech TMB14 motherboard with an enabled VME interface an entry is required in the configuration file. Configuration file entries should not be made for TMB14s where the VME interface is disabled.

An entry in the configuration file consists of a list of the following properties terminated by a semi-colon:

`name`	The device driver name. Must be set to `"tmb"`.
`class`	The type of device driver. Must be set to `"vme"`.
`reg`	A set of three numbers. The corresponding to the TMB14 registers The first number specifies the type of VME accesses used (`0x2d` for A16 mode), the second specifies the VME start address and the third specifies the amount of VME memory space used by the interface registers.
`interrupts`	A set of two numbers, the first specifying the VME IRQ level as set by the interrupt level register on the motherboard, the second specifying the VME interrupt vector to use. The VME interrupt vector number must be different from those used by other installed VME boards.
`vme_mode`	The VME access mode used by the motherboard interface. For a TMB14 this should be set to `"A16D08"`.

`motherboard`	The type of VME TRAM 0 motherboard used i.e. `"TMB14"`.
`controller`	The type of the VME controller used which runs the device driver e.g. `"FORCE-3CE"`.

By uncommenting the following lines in the supplied configuration file, the default setup for a TMB14 can be used:

```
name="tmb" class="vme"
 reg=0x2d,0x0800,0x100 interrupts=1,0x43
 vme_mode="A16D08" motherboard="TMB14" controller="FORCE-3CE";
```

This corresponds to a TMB14 used in A16D08 mode at VME address `0x0800`, IRQ level 1 and interrupt vector 43 hex controlled by a FORCE SPARC CPU-3CE.

After changing the configuration file, the computer needs to be shutdown and then rebooted with the VME motherboard and TRAM modules installed as follows:

```
# init 0
...
ok boot -r
```

Note that the `-r` option is used to rebuild the device directories under `/devices` and `/dev`. This must be done each time the configuration file is changed.

Each entry in the configuration file corresponds to a device file, for example `/dev/tmb/0`. Note that if the entries are changed in the configuration file, the device files that it corresponds to may also change leaving obsolete device files. These obsolete device files can be removed by deleting the appropriate lines in the file `/etc/path_to_inst`.

11.3 Environment Variables

The software uses some environment variables set on the host system. These are:

`TPS`	The path of the top-level Transtech directory, where the software is installed.
`TRANSPUTER`	The name of the transputer link or board to access. The name is used to search the connection database or Aserver database to find the interface type and parameters.

ICONDB	The path of the connection database file.
ASERVDB	The path of the Aserver database file.
IBOARDSIZE	Memory size of root transputer in the network.
ITERM	The file containing terminal keyboard and screen codes. (Used by interactive Inmos software such as idebug).
ISEARCH	The list of directories which iserver will search for certain files if the full pathname is not specified. Each directory path must end in a slash, and the directories are separated by a space.

11.4 Connection Database

The connection database lists the capabilities (resources) available to certain tools such as the host server utility Iserver. The actual resource that the tool is to use is specified by the TRANSPUTER environment variable or command line arguments such as the Iserver sl option. The name of the connection database file to use is the value of the environment variable ICONDB.

The PC connection database file lib/dos.db defines capabilities for an ISA card at IO address 150 and a single TMB17 PCI card. The capability names for the ISA card is chosen to have the same name as the host link as defined above. An extract from this file is:

```
# dos.db

# TMB03 at address #150
# name      | T/F   |
|150        |T      |localhost |150    |b004  |||Host link  |
|#150       |T      |localhost |150    |b004  |||Host link  |

# TMB17

|tmb17      |T      |localhost |0      |tmb17 |||Host link  |
```

The non-comment lines in this file contain the fields:

Capability	The resource name passed, for instance, to iserver using the /sl option or the environment variable TRANSPUTER.

`IsWorking`	Set to `T` if the resource is available. Set to `F` if the resource is not available.
`Machine`	The name of the host on which the resource exists. For a board connected to the local machine, this is set to `localhost`.
`Linkname`	The device name. For the PC ISA cards, this is the I/O address in hex. For PCI cards, this is the board index starting at 0 for the first card.
`Linkdev`	The type of device. Set to `b004` for ISA cards or to `tmb17` for a TMB17 PCI card.
`Mmsfile`	Reserved for future use.
`Mmslink`	Reserved for future use.
`Description`	Descriptive comment.

For example, to use iserver on an ISA board at IO address 150 hex to load and run the transputer executable `run.btl`, use the command

```
C:\> iserver /sl 150 /sb run.btl
```

Alternatively, set the environment variable `TRANSPUTER` to the required capability name. For instance,

```
> set TRANSPUTER=150
> iserver /sb run.btl
```

11.5 Network test utilities

Several utilities are provided to perform diagnostics on networks of transputers:

`check`	A network “worm”, which reports the interconnections between transputers in the network, as well as basic processor characteristics.
`mtest`	A network memory tester, which reports on the amount of memory and the speed of the memory on each transputer.
`ftest`	A functional test utility.

For example, having installed a motherboard with four processors, using the default jumper and switch settings, you can verify that the board is working by typing:

```
c:\> check
Using 150 check 3.0.1
   # Part rate Mb Bt [  Link0  Link1  Link2  Link3 ]
   0 T805d-25 0.43 0 [   HOST    ...    1:1    ... ]
   1 T805d-25 1.75 1 [    ...    0:2    2:1    ... ]
   2 T805d-30 1.77 1 [    ...    1:2    3:1    ... ]
   3 T805d-25 1.79 1 [    ...    2:2    ...    ... ]
```

The memory test is invoked as follows:

```
c:\> check | mtest
Using 150 check 3.0.1 | mtest 3.0.1
   # Part rate Mb Bt [  Link0  Link1  Link2  Link3 ] RAM,cycle
   0 T805d-25 0.43 0 [   HOST    ...    1:1    ... ] 4K,1+4096K,3;
   1 T805d-25 1.75 1 [    ...    0:2    2:1    ... ] 4K,1+4096K,3;
   2 T805d-30 1.77 1 [    ...    1:2    3:1    ... ] 4K,1+4096K,3;
   3 T805d-25 1.79 1 [    ...    2:2    ...    ... ] 4K,1+4096K,3;
```

Interpretation of the output of check normally requires sketching a diagram. The processors are numbered in the order that the worm finds them, which is normally not the most intuitive numbering scheme.

If `check` does not show the network you expect, refer to the trouble-shooting guide in chapter 12.

11.5.1 Link Switch Configuration

The `check` utility can be used to set C004 link switches. This is illustrated in the following example.

Consider a TMB17 with TRAMs fitted into sites 0, 1, 2 and 3. By default, these are connected in a link 2 to link 1 pipeline as shown by `check`:

```
C:\TPS>set TRANSPUTER=tmb17

C:\TPS>check
Using tmb17 check 3.0.3
```

```
   # Part rate Mb Bt [  Link0  Link1  Link2  Link3 ]
   0 T805b-25 1.52 0 [   HOST    1:1    3:1    ... ]
   1 T225c-20 1.74 1 [    ...    0:1    ...   C004 ]
   3 T805b-25 1.77 1 [    ...    0:2    4:1    ... ]
   4 T805b-25 1.75 1 [    ...    3:2    5:1    ... ]
   5 T805b-25 1.75 1 [    ...    4:2    ...    ... ]
```

The C004 link switch settings can be determined using the `check /cl` option and the results stored in the file `wire`:

```
C:\TPS>check /cl > wire

C:\TPS>type wire
Using tmb17 check 3.0.3
   # Part rate Mb Bt [  Link0  Link1  Link2  Link3 ]
   0 T805b-25 1.52 0 [   HOST    1:1    3:1    ... ]
   1 T225c-20 1.74 1 [    ...    0:1    ...    2:C ]
   2 C004b ()
   3 T805b-25 1.75 1 [    ...    0:2    4:1    ... ]
   4 T805b-25 1.75 1 [    ...    3:2    5:1    ... ]
   5 T805b-25 1.75 1 [    ...    4:2    ...    ... ]
```

In this case, no link switch settings have been made.

To create a link 3 to link 0 pipeline between the TRAMs, then C004 link 10 should be connected to link 1, link 11 to link 2 and link 12 to link 3. See figure 96 for details.

Edit the file `wire` and change the line for the C004 link switch to:

```
2 C004b (10-1 11-2 12-3 )
```

This amended file can then be used with the `check /cs` option to set the C004 link switch:

```
C:\TPS>check /cs < wire
Using tmb17 check 3.0.3
   # Part rate Mb Bt [  Link0  Link1  Link2  Link3 ] CHANGES
   0 T805b-25 1.52 0 [   HOST    1:1    3:1    ... ]
   1 T225c-20 1.74 1 [    ...    0:1    ...    2:C ]
   2 C004b   [ -ABC---- --123--- -------- -------- ]
   3 T805b-25 1.75 1 [    ...    0:2    4:1    ... ]
   4 T805b-25 1.75 1 [    ...    3:2    5:1    ... ]
   5 T805b-25 1.75 1 [    ...    4:2    ...    ... ]

C:\TPS>check
```

```
Using tmb17 check 3.0.3
   # Part rate Mb Bt [  Link0  Link1  Link2  Link3 ]
   0 T805b-25 1.52 0 [   HOST    1:1    3:1    3:0 ]
   1 T225c-20 1.75 1 [    ...    0:1    ...   C004 ]
   3 T805b-25 1.77 1 [    0:3    0:2    4:1    4:0 ]
   4 T805b-25 1.75 1 [    3:3    3:2    5:1    5:0 ]
   5 T805b-25 1.75 1 [    4:3    4:2    ...    ... ]
```

11.6 The Inmos server program

The Inmos server, `iserver`, is used to load programs on the transputer network, and provide host services to the program. iserver is normally invoked on a PC as follows, to load and run a program.

```
c:\>iserver /sb prog.btl args...
```

Or, under Unix:

```
% iserver -sb prog.btl args...
```

If `iserver` fails to load the network, use `check` with the `/cfb` option (see the reference page) to verify that the actual network is the same as that specified to the configurer.

11.7 Transputer host I/O utilities

The following files are supplied in the `\tps\tp\4libs` directory for use with the Dx414 toolset:

`genio.h`	Header file for `genio.lib`.
`genio.lib`	Generic I/O library. Used to improve I/O performance of transputer programs using `iserver`. The library allows a compiled program to make use of optimized I/O on a number of different boards, including the TMB16.
`hostmux.lku`	`iserver` host channel multiplexor process, allowing any number of processes, any where in a transputer network to perform host I/O.
`iocache.lku`	Disk cacheing process, providing the multiplexing facilities of `hostmux.lku`, and optimized I/O. Client processes do not need to use genio calls to benefit from optimized I/O.

The genio library and the hostmux and iocache processes are documented in the form of Unix man pages in the following section.

11.8 Inmos Aserver Support

The Inmos Aserver tools (such as rspy and inquest) are supported on PCs running Windows 95 and on SPARC computers running Solaris 2.

For a PC ISA card such as a TMB03 or a TMB16, use the default B004 compatible interface.

For a TMB08, use either the default B004 compatible interface or the faster B008 compatible interface.

For a TMB17 PCI Motherboard:

1. Add the following line to the ASERVDB database:

   ```
   | tmb17 | txcs wtmb17 0 | 1 |
   ```

2. Set the TRANSPUTER variable to tmb17 either in autoexec.bat or using iLaunch, or specify the option /sl tmb17 when starting the utilities.

For a TMB14 VME card controlled by a SPARC VME card running Solaris 2, use the supplied Aserver database file as follows:

```
setenv ASERVDB $TPS/tp/4libs/aservdb
setenv TRANSPUTER tmb0
```

To test the Aserver interface, use the rspy utility. For example, under Solaris 2:

```
% rspy
  # Part-rt  Link0 Link1 Link2 Link3
  0 T805-25   HOST   ...   ...   ...
```

11.9 Solaris 2 Device Driver

For Solaris 2, the device driver object file `/kernel/drv/tmb` is supplied along with its configuration file `/kernel/drv/tmb.conf`.

The device driver implements a B014 compatible software interface to a TMB14. The following interface is implemented:

`open()`	Opens the connection to a TMB14 interface link. The filename used should be of the form `/dev/tmb/`*n* where *n* is the interface link number.
`close()`	Closes the connection to the link.
`read()`	Reads from the link. This function returns when the data transfer has completed or when a time-out, error or signal has occurred.
`write()`	Writes to the link. This function may return before the data transfer has completed or when a time-out, error or signal has occurred.
`ioctl()`	Perform various interface functions described below.

A number of functions can be performed by calling `ioctl()` with the request argument set to `SETFLAGS` and the third argument set to a pointer to an `IMS_IO` structure with the `op` member set to the following values:

`RESET`	Resets the link.
`ANALYSE`	Resets the link with analyse asserted.
`SETTIMEOUT`	Sets the link time-out to the value in the `val` member measured in tenths of a second.

The `IMS_IO` structure and related constants are defined in the include file `ims_bcmd.h`.

For example, the following code fragment opens and resets the TMB14:

```
#include <ims_bcmd.h>
...
       union IMS_IO io;
       int fd;
```

```
/* Open link */
fd = open( "/dev/tmb/0", O_RDWR );
if (fd < 0)
      exit( 1 );

/* Reset link */
io.set.op = RESET;
io.set.val = 0;
if (ioctl( fd, SETFLAGS, &io ))
      exit( 1 );
```

The function `ioctl()` can also be called with a request argument of `READFLAGS` with the third argument set to a pointer to an `IMS_IO` structure. This returns status information.

11.10 Reference Manual Pages

11.10.1 Commands

check(1)

```
NAME
     check - Test a Transputer network

SYNOPSIS
     Unix:

     check [ -c4 ] [ -cl ] [ -cr ] [ -cs ] [ -cfb filename ] [ -h
     ] [ -i ] [ -l name ] [ -m filename ] [ -n ] [ -r ] [ -v ] [
     -x ] [ < filename ]

     DOS:

     check [ /c4 ] [ /cl ] [ /cr ] [ /cs ] [ /cfb filename ] [ /h
     ]  [ /i ] [ /l name ] [ /m filename ] [ /n ] [ /r ] [ /v ] [
     /x ] [ < filename ]

DESCRIPTION
     check is a utility which tests a network of Transputer  pro-
     cessors.   It outputs a list of the processors found and the
     connections between them.

     The output of check can be  used  as  the  input  of  mtest,
     ftest,  or  load.  Alternatively, the output from a previous
     run of check can be piped into subsequent runs.

     The host link connection to use is specified by the l option
     or the TRANSPUTER environment variable.  This corresponds to
     an entry in  the  connection  database  pointed  to  by  the
     environment variable ICONDB.

OPTIONS
     In Unix, options are preceded by `-`.  In DOS,  options  are
     preceded by '/'.

     c4   Read the state of all C004s found.

     cl   Read the state of C004s, long form.

     cr   Reset all C004s found.
```

```
     cs   Set all C004s in file piped into check.

     cfb filename
          Use filename as a configuration binary file.

     h    Print help page.

     i    Information - tells you whats happening.

     l name
          Use this link, else use  TRANSPUTER  environment  vari-
          able.

     m filename
          Use filename as a toolset map file.

     n    Do not reset the root transputer.

     r    Reset the root transputer subsystem.

     v    Leave network in virgin reset state.

     x    Ignores any file piped in to check.

BUGS
     check does not work with T450 processors.
```

ckmon(1)

NAME

ckmon - Transputer hex monitor

SYNOPSIS

Unix:

```
check | ckmon -0 [ -h ] [ -l name ] [ -a ]

ckmon -f filename -n [ -h ] [ -l name ] [ -a ]
```

DOS:

```
check | ckmon /0 [ /h ] [ /l name ] [ /a ]

ckmon /f filename /n [ /h ] [ /l name ] [ /a ]
```

DESCRIPTION

ckmon is a Transputer monitor program. The first Transputer in the network can be monitored by using the 0 option. Other Transputers can be monitored using the n option where n is the number of a processor as defined in the output of check in the file filename as specified using the f option.

The host link connection to use is specified by the l option or the TRANSPUTER environment variable. This corresponds to an entry in the connection database pointed to by the environment variable ICONDB.

OPTIONS

In Unix, options are preceded by '-'. In DOS, options are preceded by '/'.

0 Monitors root processor.

n Monitors processor n.

f filename
Use check output filename.

h Help page.

l name
Use the named link.

a Assert Subsystem Analyse.

ftest(1)

NAME

ftest - Test processors in a Transputer network

SYNOPSIS

Unix:

```
check | ftest [ -t2 ] [ -t4 ] [ -t8 ] [ -l ]
```

DOS:

```
check | ftest [ /t2 ] [ /t4 ] [ /t8 ] [ /l ]
```

DESCRIPTION

ftest is a utility used in conjunction with check which tests processors in a network of Transputers.

The host link connection to use is specified by the l option or the TRANSPUTER environment variable. This corresponds to an entry in the connection database pointed to by the environment variable ICONDB.

OPTIONS

In Unix, options are preceded by '-'. In DOS, options are preceded by '/'.

t2 Test M212s, T2s, T225s only.

t4 Test T414s, T425s only.

t8 Test T800s only.

l Log progress of testing.

iserver(1)

NAME

iserver - Inmos host server program

SYNOPSIS

Unix:

```
iserver [ -sb filename ] [ -si ] [ -se ] [ -sl linkname ] [
-sr ] [ -sa ] [ -sc filename ] [ -sp n ] [ -ss ] [ -sm ] [ -
sk n ] [ -sz[1|2] ] [ [ -st ] arguments...  ]
```

DOS:

```
iserver [ /sb filename ] [ /si ] [ /se ] [ /sl linkname ]  [
/sr  ]  [ /sa ] [ /sc filename ] [ /sp n ] [ /ss ] [ /sm ] [
/sk n ] [ /sz[1|2] ] [ [ /st ] arguments...
```

DESCRIPTION

iserver is the Inmos host server program. It is used to load programs onto Transtech systems and allow them access to the host workstation's keyboard, screen, file system and other services.

The list of available systems (resources) which iserver can use is kept in a file called the connection database. The environment variable ICONDB should be set to the pathname of this file. The resource that iserver should attempt to use is specified by the sl option or the environment variable TRANSPUTER.

The session manager provides a mechanism to allow users continuous access to a resource. It has a simple command line interface allowing users to specify and use the required resource. This interface can be customised by a configuration file specified by the environment variable ISESSION.

OPTIONS

On Unix systems iserver options are preceded by '-'. On DOS systems iserver options are preceded by '/'.

sb filename
: Boot the named file (same as -sr -ss -si -sc filename).

si
: Verbose mode.

se
: Test the error flag.

```
     sl linkname
          Use the named resource.

     sr   Reset the root transputer.

     sa   Analyse and peek the root transputer.

     sc filename
          Copy the named file to the link.

     sp n Set peek size to n Kchars.

     ss   Serve the link.

     sm   Enter the session shell.

     sk interval
          Retry connects every interval (seconds).

     sz[1|2]
          Very verbose debug mode (logs all transactions).

     st   Pass all of the following arguments to the booted  pro-
          gram.

     Options and or  arguments  not  recognised  by  iserver  are
     passed to the booted program.

ENVIRONMENT
     The following environment variables are used by iserver:

     TRANSPUTER
          Specifies the resource to used.  May be  overridden  by
          the sl option.

     ICONDB
          Pathname of the connection database file.

     ISESSION
          Pathname of the  session  manager  configuration  file.
          Default is session.cfg.
```

load(1)

NAME

load - Loads files onto a Transputer in a network

SYNOPSIS

Unix:

```
check | load [ -n File1 File2 ] [ -f filename ] [ -i n ] [ -
l ] [ -h ]
```

DOS:

```
check | load [ /n File1 File2 ] [ /f filename ] [ /i n  ]  [
/l ] [ /h ]
```

DESCRIPTION

load is a utility used in conjunction with check which loads a file or files onto a processor in a Transputer network.

The host link connection to use is specified by the l option or the TRANSPUTER environment variable. This corresponds to an entry in the connection database pointed to by the environment variable ICONDB.

OPTIONS

In Unix, options are preceded by '-'. In DOS, options are preceded by '/'.

n File1 File2
: Load Transputer n with a number of files in sequence.

f filename
: Use check command file.

i n
: Set default iserver path to Transputer n.

l
: Log progress of load.

h
: Display this help page.

mtest(1)

NAME

mtest - Test memory of Transputers in a network

SYNOPSIS

Unix:

```
check | mtest [ -c ] [ -e Kb ] [ -i iter ] [ -l ] [ -t tp  ]
[ -t2 ] [ -t4 ] [ -q ] [ -x ] [ -0 ] [ -h ]
```

DOS:

```
check | mtest [ /c ] [ /e Kb ] [ /i iter ] [ /l ] [ /t tp  ]
[ /t2 ] [ /t4 ] [ /q ] [ /x ] [ /0 ] [ /h ]
```

DESCRIPTION

mtest is a utility used in conjunction with check which tests the memory of processors in a network of Transputers.

The host link connection to use is specified by the l option or the TRANSPUTER environment variable. This corresponds to an entry in the connection database pointed to by the environment variable ICONDB.

OPTIONS

In Unix, options are preceded by '-'. In DOS, options are preceded by '/'.

c Include T2s with C004s on links (default - no).

E Kb Sets ceiling in Kbytes to which memory is tested.

i iter
Number of iterations.

l Log progress of testing.

t tp Test processor tp only.

t2 Test T2s only.

t4 Test T4s and T8s only.

q Quick memory sizing option.

x Extra information on why memory search stopped.

```
0    Do not include root processor in tests.

h    Display help page.
```

11.10.2 Linked Process Units

hostmux(2)

```
NAME
     hostmux - host server channel multiplexor

SYNOPSIS
     process (
       stacksize = 16K,
       heapsize = 64K,
       interface (
          input   from_host,
          output  to_host,
          input     in[size],
          output    out[size],
          int  n=size
          )
       ) mux;

     use “hostmux.lku” for mux;



DESCRIPTION
     The host server channel multiplexor is used where more  than
     one process needs access to host I/O facilities.  The multi-
     plexor takes any number of pairs  of  server  channels,  and
     combines  them  into  a single pair that is connected to the
     host.

     The host connections can be connected  another  multiplexor,
     giving rise to tree-shaped structures, which may be built up
     to any level of complexity.

     A process accessing the host is referred to as a “client” of
     the multiplexor.  Each client is connected to one “in” chan-
     nel, and to the “out” channel with the same array index.

     The multiplexor  receives  requests  on  any  “in”  channel,
     passes  the  request  to  the host, waits for the reply, and
     passes the reply to the corresponding out channel.  Termina-
     tion messages (which cause the iserver to terminate) receive
     special treatment:  a termination message is not  passed  to
     the  host  until  a  total  of  “n” termination messages are
     received from clients.  Hence the iserver will not terminate
     until all clients have terminated.
```

Client channels should not be left unconnected: if they are the iserver will never terminate, because the multiplexor will expect a termination message from the unconnected channel.

The multiplexor is not limited to a fixed size of iserver packet. If a packet is encountered that exceeds the current buffer size, it attempts to allocate a new buffer from the heap.

The multiplexor inputs requests from the clients using a fair ALT. This means that even if there is a continuous stream of requests from one client, requests from any other client are guaranteed to be serviced within a finite time.

EXAMPLE

```
/* loader.cfs
 *
 * This example shows the use of hostmux to
 * provide host services to run860 processes
 * on three processors.
 */

#include “i860tset.cfh”

#include “hardware.cfs”

process ( stacksize=64K, heapsize=64K );

/* process declarations */


val boot_1 “[0].860 “;
val len_1  size(boot_1);

val boot_2 “[1].860 “;
val len_2  size(boot_2);

val boot_3 “[2].860 “;
val len_3  size(boot_3);


process   ( interface ( input HostInput, output HostOutput,
               int flags = NORUN,
               char boot_file[len_1]=boot_1) ) driver_1;
```

```
process   ( interface ( input HostInput, output HostOutput,
               int flags = NORUN,
               char boot_file[len_2]=boot_2) ) driver_2;

process   ( interface ( input HostInput, output HostOutput,
               int flags = NORUN,
               char boot_file[len_3]=boot_3) ) driver_3;

process ( interface ( input HostInput,  output HostOutput,
                      input Input[2],   output Output[2],
                        int Size = 2))           mult_1;

process ( interface ( input HostInput,  output HostOutput,
                      input Input[2],   output Output[2],
                        int Size = 2))           mult_2;



input   HostInput;
output  HostOutput;

connect mult_1.HostInput to      HostInput;
connect mult_1.HostOutput to     HostOutput;

connect mult_1.Input[0]  to      mult_2.HostOutput;
connect mult_1.Output[0] to      mult_2.HostInput;

connect mult_1.Input[1]  to      driver_1.HostOutput;
connect mult_1.Output[1] to      driver_1.HostInput;

connect mult_2.Input[0]  to      driver_3.HostOutput;
connect mult_2.Output[0] to      driver_3.HostInput;

connect mult_2.Input[1]  to      driver_2.HostOutput;
connect mult_2.Output[1] to      driver_2.HostInput;

/* placement */

use “hostmux.lku” for mult_1;
use “hostmux.lku” for mult_2;
use “run860.lku” for driver_1;
use “run860.lku” for driver_2;
use “run860.lku” for driver_3;

place driver_1 on TTM100_1;
place driver_2 on TTM100_2;
place driver_3 on TTM100_3;
```

```
     place mult_1 on TTM100_1;
     place mult_2 on TTM100_2;

     place  HostInput    on host;
     place  HostOutput   on host;



FILES
     hostmux.lku


SEE ALSO
     iserver(1)
```

iocache(2)

```
NAME
     iocache - host file I/O accelerator

SYNOPSIS
     /* io_type */

     val TMB16       16384;
     val TB400       65536;
     val PARAMID     65536;
     val MCP         131072;
     val BIG_IO     0;
     val STD_IO     0;


     /* packet size */
     val DEFAULT_SIZE 0;


     /* cache block size */
     val DEFAULT_BLOCK 16384


     process (
       stacksize = 10K,
       heapsize = 1M,
       interface (
             input   from_host,
             output  to_host,
             input   from_client[n],
             output  to_client[n],
             int    num_chans = n,
             int     io_type = type,
             int     packet_size = size,
             int     block_size = b_size
          )
       ) cache;

     use “iocache.lku” for cache;



DESCRIPTION
     The file cache process improves I/O performance by  cacheing
     recently accessed data.  The size of I/O blocks is converted
     up or down to the optimum, depending on the host server  and
```

interface hardware being used.

Where many processes are reading the same file concurrently (such as when a group of i860s are booting the same program), I/O speed can be dramatically increased. The problem of the iserver running out of file descriptors is also avoided, as the cache only opens each file only once.

The configuration parameters are as follows: “from_host” and “to_host” are the normal iserver channels to the host. “from_client” and “to_client” are arrays of iserver channels to the clients, or application processes. Note that iocache multiplexes the host channels of several processes in a manner similar to hostmux(2).

“io_type” specifies the type of I/O optimization that is performed. It is an integer as defined in the above value declarations. “packet_size” is used to determine the size of iserver packets when the io_type is BIG_IO - it is ignored for all other I/O types. To obtain I/O compatible with a standard iserver, both io_type and packet_size should be set to zero.

“block_size” determines the size of buffer that is allocated for each file opened by a client. A value of 16384 (16K) is recommended. The space available for buffers is controlled by the heapsize given to the process. In general, as much heap as possible should be given to this process.

If the host is a PC running DOS, then text mode files opened by Transputer processes are not cached. This does not apply to files opened by i860s.

The standard input and output are not cached.

When the same file is opened more than once (eg, by different client processes) the cached data can be shared between processes. For this to happen, both the file name and the open mode must match exactly.

FILES

iocache.lku

SEE ALSO

iserver(1) hostmux(2)

11.10.3 Program Function Calls

genio(3)

```
NAME
     genio,  genio_open,  genio_close,  genio_read,  genio_write,
     genio_lseek, genio_control, - optimized file I/O library


SYNOPSIS
     #include <genio.h>

     int genio_open( char *name, int mode );

     int genio_close( int fd );

     int genio_read( int fd, char *buf, int n );

     int genio_write( int fd, char *buf, int n );

     int genio_lseek( int fd, long int offset, int origin );

     void genio_control( int iotype, int param );



DESCRIPTION
     The generic I/O library implements faster  versions  of  the
     Inmos C Toolset low level file I/O functions.  Several tech-
     niques are included that give optimal results  on  different
     kinds of hardware.

     This is done by allowing a larger iserver packet size (up to
     4096  bytes)  to be used than normal (512 bytes), or, in the
     case of the TMB16 motherboard, by DMA  direct  to  the  BIOS
     disk controller routines.

     All operations on files opened using genio_open must be per-
     formed  using  the genio library functions and not using the
     default library  functions.   Similarly  the  genio  library
     functions  should not be used on files opened using the nor-
     mal open function.

     On multi-processor systems, the server  channels  should  be
     multiplexed  using  hostmux.lku  and not the occam procedure
     so.multiplexor.  This routine should also be used instead of
     so.buffer.
```

The open, close, read, write and lseek routines are used in the same way as the corresponding Inmos routines, taking exactly the same parameters.

By default, the library performs standard Iserver I/O with 512 byte packets. genio_control is used to specify what type of I/O optimizations should be used. It can be called to switch between modes whenever desired: I/O will be performed using the mode selected when the file was opened.

genio_control takes two arguments: an I/O type and a secondary parameter. The parameter is ignored, unless otherwise specified. The following I/O types can be used:

STD_IO Standard mode (default)

TMB16_IO Optimize I/O for the TMB16 (NB the calling process must have it's host I/O links connected directly to the TMB16 interface, NOT through any multiplexors)

TB400_IO Use enlarged iserver packets optimized for the TB400.

MCP_IO Use enlarged iserver packets optimized for MCP1000 and MCP500.

BIG_IO Use enlarged iserver packets of the specified size. The second argument specifies the size of packet that may be used in bytes.

FILES

genio.h genio.lib

SEE ALSO

hostmux(2)

RESTRICTIONS

File descriptors used by routines in the genio library cannot be used with the standard file I/O routines and vice-versa.

The server channels from Transputer processes using the genio routines cannot be multiplexed using so.multiplexor or buffered using so.buffer.

If TMB16-optimized I/O is enabled, the calling process must have its host channels connected DIRECTLY to the TMB16 interface link. No multiplexors or buffers are allowed between the library and the host link. The hostmux process can handle enlarged iserver packets.

Chapter 12

Trouble-shooting

In the event of a problem with your transputer equipment, the following sections give some hints for tracking down the cause.

12.1 TRAM checklist

It check reports some, but not all of the transputers, then it is best to draw a picture of the processors and links that are detected. Then you can determine which processors or links are not detected.

The following checklist, based on the signals at the TRAM pins, may be used to find the cause of the problem.

.

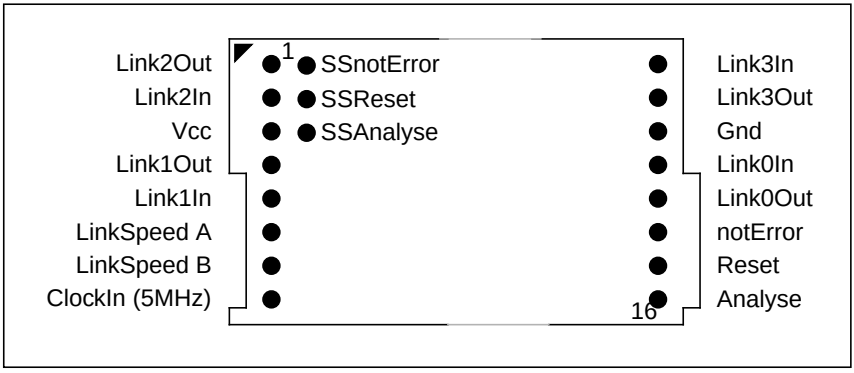


Figure 105. TRAM Pinout

12.1.1 Reset

Check the setting of the control configuration jumpers on the motherboard. With the default settings given in this manual, sub-system pins are not required.

Using a logic probe (with LEDs that light to show high or low TTL signal states) check that the reset pulse can be seen. The pulse is quite short, but should be clearly visible. Most voltmeters are too heavily damped to show the reset pulse.

12.1.2 Links

Pipe-jumpers are a common source of trouble. Check that they are fitted in the right places, and that the SIL strip between the TRAM and the motherboard is fitted.

Check the settings of the link patch area, and cable connections.

Using a continuity tester or resistance meter, check the connection of the link. There should be a 56 Ohm series resistance.

If a transputer cannot be booted from one link, even if the link is correctly connected, it may be because its other links are not held low, but are disconnected and floating high. All motherboards incorporate pull-down resistors to prevent this.

12.1.3 Link speed

Check jumper settings. Check TRAM signals with a voltmeter or logic probe - both link speed signals should be high for the default 20MBit/s operation.

12.1.4 Analyse

Check the TRAM analyse signal is low (not asserted). It should remain low when `check` is run.

12.1.5 Power

Using a voltmeter, check that 5V is present.

12.1.6 Clock

Using an oscilloscope or logic probe, check that the 5MHz clock is present.

12.2 PC Host Interface

If `check` cannot find any transputers, then the problem is may be with the operation of the host interface. It is best to remove all non-essential hardware and software from the machine, repeat the test. Then re-introduce other devices one at a time, resolving any clashes that occur. If you are using more than one transputer card, start with just one card, then introduce the others.

You may need to change the IO address of the transputer card. Remember to update the environment variable TRANSPUTER and the database `\tps\lib\dos.db` before running `check` again.

Note that there are many different versions of check and iserver, released by many organizations. Check that you have the `\tps\bin` directory in your path ahead of any other transputer-related software (e.g., the Inmos toolset).

Index

Symbols

A

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

V

W