inmos®

*i*X package installation and user guide

Incorporating the IMS B020 and IMS S720A products.

SGS-THOMSON
MICROELECTRONICS

INMOS is a member of the SGS–THOMSON Microelectronics Group

72 TRN 234 00
October 1991

INMOS is a member of the SGS–THOMSON Microelectronics Group.

72 TRN 234 00 October 1991

Contents

1 Introduction

The *i*X package is an INMOS software and hardware development that adds a high performance colour X-terminal capability to a PC host. The product is intended to offer a low cost upgrade path for those users who are currently PC based, albeit on a network, but who wish to take advantage of a growing UNIX environment of workstations and workstation software around them. With *i*X, users can access this powerful environment without investing in a workstation, losing access to their current PC or sacrificing additional desk space for a dedicated X-terminal.

The hardware component of *i*X – the IMS B020, is also capable of acting as a general purpose transputer development and graphics application platform capable of expansion using on-board TRAnsputer Modules (TRAMs) sockets.

1.1 Product features

- Uses an optimised AT graphics card for high performance colour X-terminal implementation without needing Megabytes of PC memory.
- X windows server version 11.4 fully supported - 256 colours out of a palette of 16.7 million and multiple client application windows.
- Programmable screen resolutions up to 1024×768 for workstation quality displays as well as supporting current VGA or Super VGA scan rates.
- Uses existing PC ethernet interface card and networking software.
- Switch back to DOS environment with a hot-key.
- Optional font file storage on remote server machine.
- Diagnostic test and configuration software included.
- Easy to use installation script.
- Transputer module expansion capability for compute intensive applications and parallel processing development.

2 Product components

The product, as delivered, includes:

- A transputer based PC add in card, IMS B020, as described below with 4 Mbyte DRAM option fitted.
- A set of 5 1/4" (360kbyte) and 3 1/2" (720kbyte) floppy disks for the IMS 720A software.
- Installation and User Guide (this document)
- VGA loopback cable

To use the *i*X product, it will also be necessary to have:

- An IBM PC or compatible (8086,80286,80386,80486 etc.) with a PC, XT or AT style ISA expansion bus. A copy of DOS at version 3.2 or higher must be installed. At least one full length slot will need to be free to install the card. The PC should have a hard disk drive with about 9MBytes of free space to install all *i*X options.
- A VGA, Super-VGA, 8514A, or multi-sync colour monitor. The software offers configuration options to cope with the display on most types of PC monitor, but example configurations are only supplied for a limited range of monitors of the above types. With an X-display, the larger resolution options are much easier to use, but need not be installed immediately.
- A three-button mouse or 'pointing device', either serial or parallel interface, with a Microsoft, or Microsoft compatible (INT33 type) software driver. While three buttons are not strictly necessary to support any X-windows applications, many will expect this capability. With a two button mouse *i*X will interpret left and right button clicks as corresponding to left and right buttons on a three button device, ie there will be no access to a middle button.
- An ethernet adaptor card and a supporting package of TCP/IP networking software. The supplied release of the *i*X software only supports the use of the PC/TCP package from FTP Inc. In the future other networking packages offering similar networking facilities will be supported, but performance and ease of use may vary. The ethernet interface card can be any of those supported by the networking package.

Although it is not required for installation and operation of *i*X, the system clearly offers the most in a UNIX networked environment where applications using X-window displays are established. The X-window system is intended as a medium for transparently using applications running on both local and remote machines. To get the best out of the *i*X system, work with your local UNIX system administrators to set up and make available the maximum range of X-windows applications and utilities.

A diagram of the *i*X board is shown in figure 3.1.

Alternative hardware that may be used to run *i*X software includes an IMS B419 graphics TRAM mounted on a an IMS B008 or other PC motherboard. The software is compatible with this run-time environment, but cannot offer such a wide range of support for different display options. Information about using this kind of hardware is included in Appendix D.

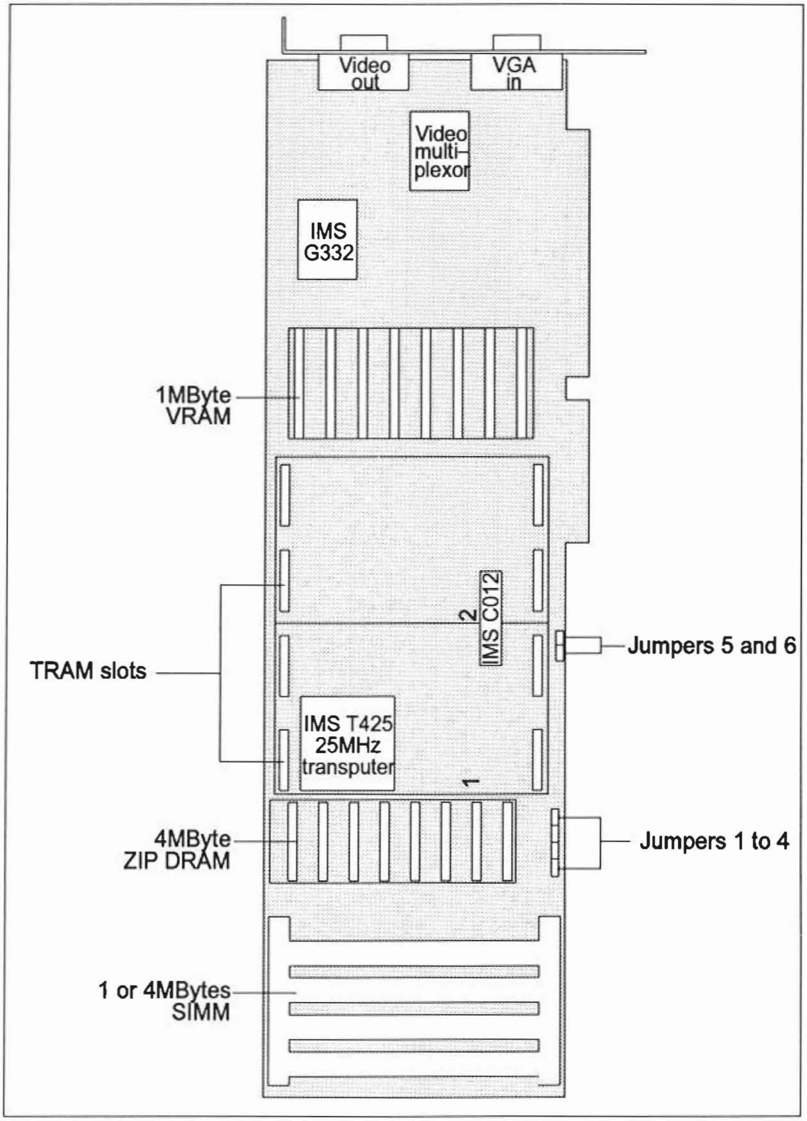


Figure 3.1 *i*X board layout

3 System overview

This chapter gives an introduction to the X-window system, intended simply as a background to the development, and an overview of the approach taken to implementing an X-terminal using the *i*X package.

3.1 The X-window system

The X-window system was originally developed by the Massachusetts Institute of Technology as a methodology for making workstation applications independent of the display technology used in the various UNIX systems within their development environment. It was intended initially as a UNIX interface tool but, as various versions have been developed, it has become more and more a portable display medium, independent of target hardware or operating system environments. In particular, X-terminals that interact with X-windows applications do not have to run UNIX themselves. This is one of the reasons that *i*X can offer a high quality X-window display running in a PC and MSDOS environment.

The X-window environment is nominally divided into two parts: the X-server and the X application clients. All display actions required by a client are performed on its behalf by the server. This includes all keyboard and mouse input requests. Both clients and server can reside on the same computer, as workstation, or be physically separated and joined only by a network connection. Within the *i*X system, the X-server runs in the PC environment on a dedicated graphics card, and the X-windows applications run on hosts elsewhere in the network. Since it is primarily a networked display mechanism, the X-window system gives complete flexibility in the environment used by the application. It can be a super-computer or another PC. All computational resources needed by a particular client are supplied by its environment, leaving the *i*X processor free to perform graphics operations and track the mouse and keyboard inputs. This flexibility extends to the capability of supporting clients from different host environments on the same screen at the same time, even exchanging data between them – a truly inter-operable computer network environment.

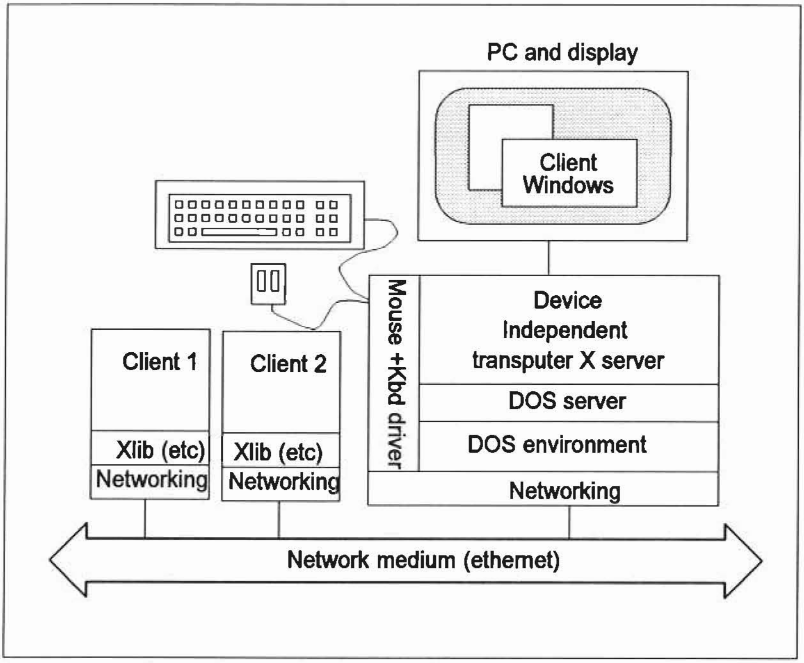


Figure 4.1 Using the X-window system in a network

A fuller description of the X-window system can be found in several books now published on the subject, for example see reference [5].

3.2 The design of *i*X

Several guiding principles were followed when designing this product:

- *i*X is intended to offer a low cost upgrade path for those users who are currently PC based, with a network connection, and who want to take advantage of a growing environment of workstations and workstation software around them without incurring the cost of acquiring a workstation or additional desk space of a dedicated X-terminal. Typically, these users will be interested in maintaining access to current PC facilities and existing files.

- Users may well be non-technical and wish to gain access to network based facilities via a convenient and friendly Graphical User Interface environment with a sophisticated appearance and applications - probably OSF/ Motif based. They will not wish to be concerned with the intricacies of the system setup or user configuration.

- If a PC is already connected to ethernet and using TCP/IP it should be possible to make use of the existing investment in the interface card and networking software. Where re-use of the software is impossible, the existing interface hardware should at least be re-usable.

- Re-use of the existing screen, if it is VGA resolution or better, should be possible to minimize the cost of the upgrade. The user may find 640×480 resolution unsatisfactory for any kind of extensive X-window usage but this should be an option for casual users or as a first step.

- As with commercial X-terminals, the majority of the setup and working data should be held on a network host where this is possible. It should however be a user option to install all necessary files on the PC disk.

To satisfy all these types of considerations, and to provide the highest performance X-server implementation, a specially designed, transputer based add-in card has been used to run the X-server code. The existing file system, networking support, screen, keyboard and mouse facilities of the PC are used, through a host server task, in conjunction with this high performance card to implement an X-window graphical terminal environment.

The hardware design of the IMS B020 is more generic than required for a simple X implementation. Other potential applications include its use for MS Windows graphics acceleration, transputer based multi-media tools and postscript printing utilities. It is also possible to use the card for hosting standard INMOS transputer development tools for C, Fortran and occam parallel programming. For further details of these capabilities see chapter 10.1.

3.3 Networking support

To offer an X-window environment using *i*X, it is necessary to have some access to a networked environment based on the TCP/IP protocol. Their are a number of excellent network packages available for PCs together with a range of low cost ethernet interface cards. The *i*X product eventually aims to support as many as possible of these implementations, but initially can only be used with the PC/TCP product from FTP Inc. There are a number of items to be considered when looking at a network package for this application:

- How many sockets can the network package support? Each X-window client requires a distinct network connection, or socket, to the server If the number of sockets is limited, this will also limit the number of distinct clients that are able to use the screen at any one time. A typical X-window session may be using 5-10 clients. When the housekeeping needs of the software are also taken into account, a minimum of 8-16 sockets may be required.

- How much memory does the package require? In order to use *i*X, the network software and the PC server must be running in the existing 640k memory limit of the PC. The PC server can use up to 180kbytes of memory.

- Does the package support a PC client implementation of NFS? The local storage of certain X-server files (discussed below) may require quite large amounts of PC disk space. If these needs can be supplied from a remote disk via the Network File System it may be convenient to take advantage of the facility.

The interface card and networking package should be installed, following carefully their own instructions, before *i*X installation commences. It should be possible to access other nodes on the network using the test facilities of the package. Appendix C gives additional information about TCP/IP installation and optimisation.

3.4 Fonts and support files

The X-server requires a set of support files to be accessible during its normal operation. At start-up it requires access to a configuration file, a colour database file and a host access file. During application operation, it will read any number of font files to support the display of text characters in the graphics environment. These files can be based anywhere accessible to the X-server as it starts operation and can be read from a number of 'font search paths'. The font files, in particular, can be installed on the PC, but require about 7Mbytes of disk space in expanded form. During installation, it may be better to consider access to a font server node over the network using NFS. Your system administrator will be able to install the fonts on a central machine and make them accessible to all PCs running the *i*X package. For more details on this procedure see chapter 9.

4 Hardware components and installation

This chapter gives a short description of the *i*X hardware and describe how to install the card in a host PC.

4.1 *i*X Hardware

*i*X uses a custom designed, transputer based PC, XT or AT graphics card with between 4 and 12 Mbytes of DRAM to support a high performance X-server implementation. 1 Mbyte of VRAM is fitted to support screen resolutions of up to 1024×768 in 256 distinct colours, out of a palette of over 16.7 million. The 10 MIPs IMS T425 Transputer is interfaced to the latest Colour Video Controller device, the IMS G332 to provide exceptionally flexible capabilities in a simple and highly reliable design. Using the flexibility of the G332, many different monitor resolutions and frame rates can be configured, allowing the re-use of existing PC displays where desired. The normal operation of the PC is supported by passing VGA signals to the display through a video switch when it is being used for DOS application. When the X-server is started, the display mode changes to use the transputer based graphics engine. The main features of the design are:

- VGA compatible PC display adaptor supporting resolutions up to 1024X768.
- Analogue switch between VGA and X display video signals under transputer control.
- IMS C012 link adaptor for PC bus interface.
- Two size two TRAM sites for applications expansion.

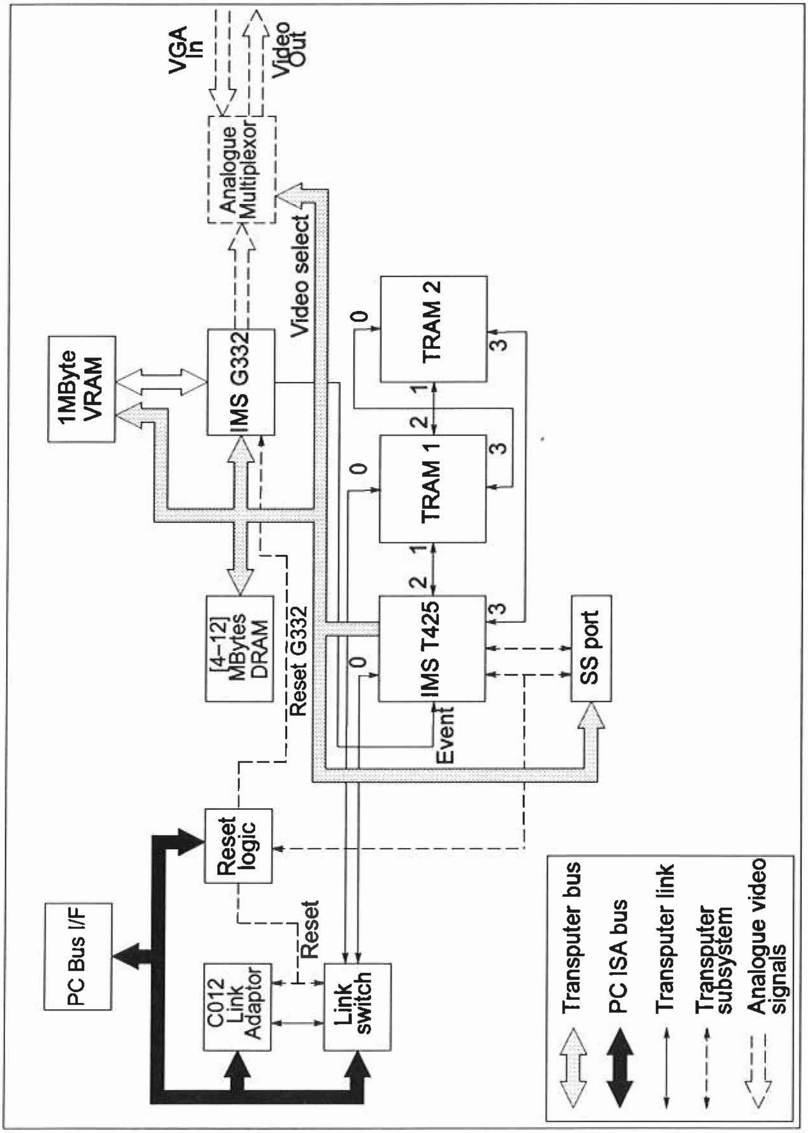


Figure 5.1 Diagram of IMS B020 *i*X card

A block diagram of the *i*X board is shown in figure 5.1. It can be seen from this diagram that a variety of TRAM expansion schemes are possible using the IMS B020 hardware, but in its base configuration it will simply contain an IMS T425 transputer that controls the video and switch option which supports the X-server operation. The transputer itself is under control of a PC hosted program that loads the X-server code and supports its operation by supplying files, network support and device inputs.

4.2 Inspection and preparation

4.2.1 Handling precautions

To prevent damage from static discharge, take special care when handling the *i*X board and inserting it into the PC.

1. While holding the board in its bag, touch an earthed metal surface with the other hand.
2. Carefully remove board from its protective bag and hold by the edges only. Avoid touching components or connections.
3. While setting jumpers or installing memory upgrades, hold the board by the edges only.
4. If the board is to be shipped or stored out of the PC, then it should be wrapped in its protective bag.

4.2.2 How to install additional SIMMs

Two sizes of expansion SIMM are supported - 256kbx8 and 1Mbx8. Four SIMMS will be needed to expand the memory space for the transputer and these should be inserted carefully in the socket provided at one end of the board. The jumpers J3 or J4 (positioned on the lower edge of the card) should be inserted to select the size of SIMM installed. J4 is used to select the smaller size options, J3 selects the larger size option. Do not insert both jumpers at the same time. If no memory expansion is fitted, then the position of the jumper is not important. Refer to Appendix section A.5.

4.2.3 I/O Jumper positions

The other jumper selectable option is for PC I/O address selection. The factory set default address of the card is #150. This may clash with other options installed in your PC and may therefore need to be changed. As shown in the diagram below a single jumper should be used to select an address option, installed in position J5,J6 or J5/J6. These jumpers are found on the lower edge of the PCB.

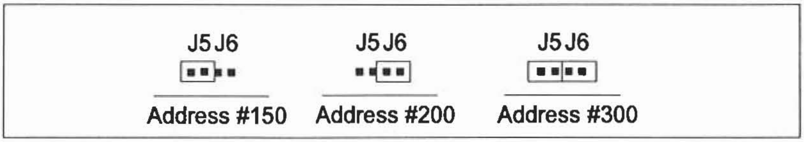


Figure 5.2 Address jumper positions

Note: # = hexadecimal address.

4.3 Board installation

Carefully remove the outer case of the PC to expose the expansion slots into which the board will be inserted.

Choose the slot to be used for the *i*X board. This should be a full length slot with a full set of bus signals. In an XT compatible machine, the smaller set of edge connectors may not fit into a socket. This is acceptable and does not affect board operation.

Remove the expansion slot cover from the rear panel of the chosen slot, retaining the fixing screw. Carefully insert the board, holding it by its edges only and press it home into the slot sockets. Secure the bracket on the board to the back panel using the fixing screw.

Using a VGA adaptor and display in the PC, take the video cable from the PC monitor, unplug it from the existing VGA adaptor and insert it into the top socket on the *i*X board. Take the supplied VGA loopback cable (see Appendix section A.7) and plug it into the VGA input socket on the *i*X board, and route it to the output of the VGA adaptor. This is shown in diagrammatic form in figure 5.3.

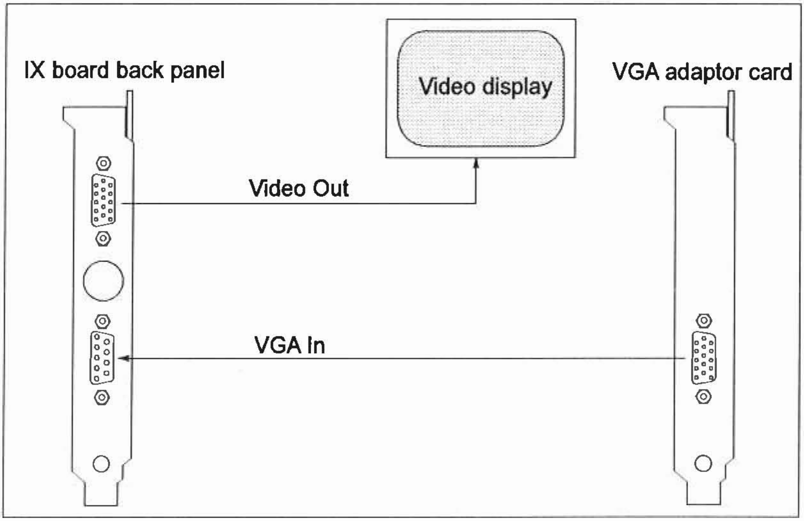


Figure 5.3 Video wiring detail

The wiring shown in figure 5.3 allows use of the existing VGA display screen for both X-windows operation and normal PC use. If a separate monitor is desired for *i*X use, or the current video adaptor of your PC is not VGA or VGA compatible, then simply route a video cable from the *i*X card back panel to the chosen display.

Replace the case of the PC.

Switch the PC on and look for the normal DOS boot sequence to be displayed on the monitor screen. If any problems occur, refer to the troubleshooting section below.

4.4 Troubleshooting

- **PC does not boot.** If the PC fails to initialise correctly – no disk activity or boot up messages on the screen, then there is probably an address clash between the *i*X card and some other device. Remove the card, change the address jumpers and try again. If no address can be found that gives correct results in combination with other installed cards, then remove other devices with conflicting addresses.

- **No picture on screen.** If the PC executes its normal boot sequence with associated disk activity etc, but no messages appear on the screen, then check the video wiring for correct and firm connections.

5 Software components and installation

This chapter will give an overview of the software components within the product and detail the step by step installation process.

5.1 Software components

Software is supplied on a set of 51/4" 360k disks and 31/2" 720k disks. The major software components on these disks are as follows:

- Transputer executable of X-server and interface code.
- PC hosted server executable supporting communications and device I/O.
- Installation script.
- *i***X** configuration files for specific monitor types.
- X runtime files and fonts compiled for transputer server.
- Diagnostic and test software.

The *i*X-server is primarily a port of the MIT Release 11.4 X-server code optimized for an IMS T425 transputer and a IMS G332 CVC device. The server can use memory configurations of between 2 and 12 Mbytes DRAM and features a variety of run-time configuration options.1 Mbyte of video memory supports a maximum display resolution of 1024×768 pixels, using 256 colours out of a palette of 16.7 million.

The XDMCP display control protocol is supported by the server under user configuration control. The initial display protocol requests will be directed towards a user defined XDM manager process typically on a UNIX host. Client access rights to the X-display are controlled via a combination of host access control (hosts file) and XDM magic cookie authentication. For more details on these features, see the X-windows documentation.

5.2 Installation procedure

The software installation process for *i***X** has been made as simple as possible and is executed using a friendly question and answer sequence to allow user configuration information to be performed as the process progresses.

1 Insert disk number 1 from package (disk of correct size) into floppy drive.

2 Close disk drive door, if required.

3 Type 'A:' at the DOS prompt and press the Enter key.

4 The DOS prompt is now A:> Type 'install' and press the Enter key.

Within a few seconds the install process will start. The process offers a full description of the operations being performed and will require little further explanation here.

The install process offers menus with various selections. Use the keyboard cursor keys to move up and down between selections and the use the Enter key to choose each option. Adequate defaults are provided for each option where the information is unknown at install time. Other information will be requested in the form of dialogue boxes. Names of files and directory paths will default to those used by convention.

The installation options include:

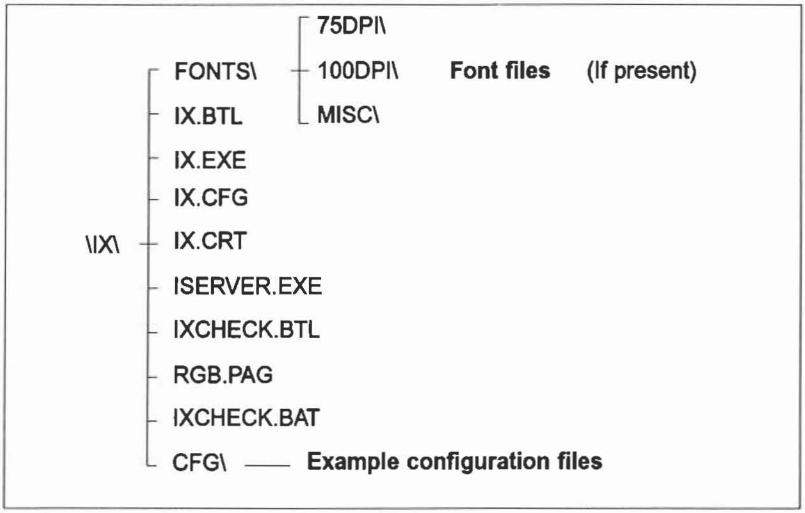
- Installation directory name and disk drive. The script will check for adequate space on the selected drive. It is suggested that the product is installed in an *i*X subdirectory on the main hard disk drive.
- Whether local X-server fonts are required.
- Font search paths, if files are not local.
- The kind of monitor used for the X display
- How much DRAM is installed on the board.
- Name of XDM server machine in the network, if any.
- Screen saver timeout values.
- X-server backing store mode.
- Hot-key selection for toggle to DOS and *i*X termination.

The user will be prompted to insert each additional disk in turn by the installation program. Depending on the options chosen, certain disks may not be used during the running of the installation process. In particular if you chose not to install local font files then the installation process will be considerably shorter. If font files exist from a previously installed version of the software then these can be re-used but all other components of the software release will be overwritten.

The result of running this script will be the installation of the product directory tree and the creation of two files 'IX.CFG' and 'IX.CRT' in the installation directory which will contain, in ASCII readable format, the results of this parameter selection combined with a larger amount of default information and comments selected for the board and monitor that are in use. The alternative installation options are also copied to the disk, but stored in a subdirectory for possible use when upgrading the installation.

5.3 Installation directory structure

The directory structure of the *i*X product after installation will be as shown below:



```
                        ┌ 75DPI\
            ┌ FONTS\  ──┼ 100DPI\      Font files    (If present)
            ├ IX.BTL    └ MISC\
            ├ IX.EXE
            ├ IX.CFG
    \IX\  ──┼ IX.CRT
            ├ ISERVER.EXE
            ├ IXCHECK.BTL
            ├ RGB.PAG
            ├ IXCHECK.BAT
            └ CFG\  ──── Example configuration files
```

Figure 6.1 Installed directory structure

5.4 Changes to DOS start-up files

As an additional function the install process will offer to update the PC's CONFIG.SYS file. There are optimal settings for the following two parameters in this file, shown below:

Files = 20
Buffers = 32

6 Diagnostic software

As an aid to user confidence, and in order to validate the screen configuration parameters, a simple diagnostic test program has been provided in executable form. This chapter will explain how to run the software to give confidence in *i*X board operation and how to use its further facilities to set up the *i*X software to use a wide variety of PC monitor options.

6.1 Starting the diagnostics

To run the diagnostic follow the following procedure.

Change your current DOS directory position to the *i*X installation directory using the 'cd' command.

At the DOS prompt type 'ixcheck' and then press the Enter key.

The PC screen will display a short message:

```
Booting root transputer /
```

followed by a display of the contents of the video configuration file IX.CRT as read by the software. Press any key on the keyboard to continue execution and the screen will clear and display the following menu:

```
  iX diagnostic software version 2.7
  Top level menu
0 Exit

1 Generate testcard
2 Confidence tests
3 Diagnostics
4 Monitor configuration
5 Demos
Esc to toggle VGA/iX output to monitor.
```

The desired option is selected by typing a single number. If the Enter key, or any other unrecognised key is pressed, then the display will remain.

At any time during the diagnostics the escape key can be used to toggle the display between the PC menu selections and the current video display generated by the *i*X card.The selection mechanism still operates when the *i*X display is on the screen, so pressing the escape key repeatedly will result in toggling backward and forward between VGA and *i*X screens. If the testcard or demos options are chosen the display is automatically set to the *i*X picture. Pressing the escape key will return to the menu selection. The following paragraphs describe in detail the test options available.

6.2 Testcard

The most commonly used of the test options is the display of the video testcard. It is selected by pressing '1' when the top level menu is displayed and will result in a video pattern being displayed on the monitor. The picture here is a comprehensive test of the correct performance of all aspects of the *i*X video hardware and a revealing test of the correct setup of the configuration parameter set for the type of display being used. If the picture is distorted, rolling vertically or horizontally or has excessive borders, then some re-configuration of monitor parameters may be needed. The configuration file selected during initial software installation should contain values compatible with your display, and this is programmed into the *i*X hardware by the diagnostics before setting up the picture. Each model of display and manufacturer may however differ in small aspects of display characteristics, which require some fine tuning to the configuration parameters. Press the escape key to toggle back to the display menu.

6.3 Confidence tests

The *i*X hardware can be checked at a detailed level by the diagnostic test described here. Selecting this option from the top level menu will result in a sun menu being displayed:

```
Confidence Tests
0 Exit
1 Full test sequence
2 Test video memory
3 Test Video controller registers
```

The full test sequence will test the memory used for the video display, the base working memory on the board and check any installed memory extensions. It will then exercise both DRAM and VRAM memory space for correct operation.

The video memory test will set up and read back a set of video patterns that exercise the display memory fully.

The video controller chip is exercised by using the register test option that repeatedly reads and writes patterns to check for any interface errors.

6.4 Diagnostic tests

From the top level menu, invoking this option results in the display of a limited sub-menu:

```
Diagnostics
0 Exit
1 Video memory
```

The Video memory option allows a hex display of the memory contents to be edited at will with the option to display the modified video contents on the *i*X display.

6.5 Demos

The demonstration sub-menu is as follows:

```
Demos
0 Exit
1 Random squares
2 Random circles
```

The two options allow the display of vivid patterns that exercise the video controller operation and colour palette update functions in a thorough fashion. Each function can be terminated by pressing any key on the keyboard.

6.6 Monitor Configuration

Unless the user has a thorough understanding of the IMS G332 video controller used by the *i*X hardware, the use of the re-configuration options is difficult to follow. Appendix B contains a description of the IMS G332 device and its control registers which should be carefully studied before changing any parameters.

It is also possible to overwrite existing values of display parameters in the configuration file using this software, so before using this option, save any working configuration files by making copies of them in the local directory.

From the top level menu, the following steps can be used to set up new configurations or modify existing configuration values.

Generate a testcard display to set a pattern into video memory.

Choose monitor configuration from the menu. A full description of current video controller and monitor parameters will be displayed in a formatted form together with an option menu to load or save a file of values, update individual values or quit the display.

By using the cursor keys, individual parameter values can be selected then adjusted. At any time, hitting the escape key will toggle to the updated *i*X display to see the effect of changes. Many changes in values will result in illegal combinations and a lack of *i*X video picture, so some care and pre-calculation is required.As any change is made, the register will be re-programmed and the video re-initialised, other parameters may change as the software re-calculates the dependent variables. By choosing the testcard option on the main menu, the effect of the change can be seen immediately. When changes have been made, the results can be saved back to the initial file, or any other, by choosing this option on the display sub-menu. The file of values will be written in a form consistent with that required for *i*X operation and with a comment header identifying the monitor type, manufacture and date of creation. The data can be written back by default into the file that was read at diagnostic start-up but the original contents of that file will be lost.

If the 'ixcheck' command is invoked giving as a parameter the name of the configuration file to be tested, then configurations other than the current default of 'IX.CRT' can be created. If the file name specified does not exist or cannot be read by the software as it starts up, default values will be used as the starting video configuration. By default, the 'ixcheck' software will always attempt to read and use parameters contained in the IX.CRT file, which is consistent with normal operation of the *i*X-server software.

7 *i*X operation

This chapter gives a short overview of the initial operation of the *i*X-server and how to run X clients that display on the *i*X-server screen.

7.1 Starting *i*X

Before starting the *i*X-server, it is necessary to check for initialization of the mouse and network software. Both may have been already installed and started during the PC boot procedure. If this is not the case, then initialize their operation as described in their respective instruction manuals.

After installation and checkout, *i*X can be executed by typing 'IX' at the DOS command line. The current default directory must be that where *i*X is installed. The initial response will be a message:

```
        Loading iX Server
```

After a few seconds the screen will switch from VGA mode into *i*X mode and the picture will change to the grey X 'root weave' pattern with an X default cursor in the centre of the pattern. The X-server is fully operational within 5 or 6 seconds. After this point, the user will be able to toggle back to VGA text display, using the pre-defined hot-key and observe the X-server informative messages that are displayed. When the mouse is moved the cursor on the X screen should respond correctly. Check that the cursor moves to the top, bottom and edges of the screen without vanishing and that the response speed of the cursor is adequate for the size of display. Any problems that are seen here will necessitate changes in the configuration file.

Check that the X-server is finding its font directories correctly. Error messages will be displayed on the VGA text screen if the expected files are not found in their predefined locations.

Check also that hitting the termination hot-key and the DOS hot-key perform correctly. In both instances the VGA screen should be restored and the keyboard should function normally. Incorrect operation of these functions may leave the keyboard in a non-operational state.

7.2 Initial Use of display

What happens after the initial screen is displayed depends on the overall setup of the system, both that of the PC and host UNIX network. The root screen is of little use by itself - there is no capability to access any application or remote system. One of the following techniques must be employed to use the server in conjunction with X-windows application.

- The first and preferred option is to use the XDM options in the configuration file to start up an XDM session with a remote server machine. The UNIX server machine must have the 'xdm' display manager service running as part of its setup. A correct setup will result in an X display login prompt for the remote UNIX system appearing on the X-server screen a few seconds after startup. From this point all client display characteristics will be determined by this remote system and the configuration files within the users directory space. Logging into the system will give a default X display session for that particular user.
- It is also possible to start an X client such as Xterm from a separate login to a UNIX machine. The X client needs to be manually started with its display parameter directed at the named *i***X** node on the network. After the command line window has been displayed on the *i***X** screen, it will be necessary to set the shell environment variable DISPLAy. When this is set to the identifier of the local X screen, additional X clients started from the shell command line will appear on the same display.

7.3 X window utilities.

An overall description of common X clients and utilities is provided in the X window documentation, for instance see [5]. The following clients are, however, worth a separate mention here since they can be used to modify particular facets of the X-server operation at run-time.

7.3.1 xset

The 'xset' utility modifies some basic operational characteristics of the X-server to which it is addressed. In particular it can modify:

- Screen-saver timeout and action
- Font paths
- Cursor speed
- Bell volume and keyclick

xset can also be used to display the current settings of these controls.

7.3.2 xhost

This utility can be used to modify access rights to the X-server. Normally only a limited named set of hosts can used to run applications that use the *i***X**-server display. The list can be displayed and modified by the 'xhost' command or even disabled completely.

7.3.3 xmodmap

This utility can be used to change the mapping of keyboard keycaps onto displayed characters. Used to cope with the differences in keyboard layout and character assignments for different countries.

7.3.4 xterm

This is a text application display tool and a terminal emulator for the X-window system which can give a basic display similar to that on an alphanumeric VDU system. Applications that have not been written to use X-window graphics can be used alongside those that have.

7.3.5 mwm

One of a class of so called window manager clients. These clients start to impose a look and feel over the display area and allow windows from other applications to be placed, moved and deleted. They also allow, via pull-down or pop-up menus, other applications to be started. The Motif window manager is one of the most sophisticated of these applications and exhibits a style consistent with the standard Motif toolkit used by an increasing number of applications.

7.4 Hot-key to DOS

Pressing this user configurable hot-key combination will return the PC screen to the VGA display and run a DOS command shell. X-window client activity will be suspended without loss of data for a period of some minutes. The DOS shell can be used to execute all normal commands, but memory will be limited for large applications. Note also that the network software will still be in use and the transputer card will be inaccessible.

To resume X display, end the DOS shell by typing 'exit' followed by the enter key. *i*X will resume control of the screen and the X client sessions can continue.

7.5 Termination of *i*X

Pressing this user configurable hot-key will terminate all X-clients and X-server and return display to VGA mode, terminating the *i*X program. All data from current X-window clients will be lost. A typical key used for this purpose is 'scroll-lock'. If *i*X is configured to run 'one session' (see section 8.2) then *i*X will be terminated by logging out of any session that was commenced with the XDM manager.

7.6 Troubleshooting

If problems are encountered starting *i*X, the following suggestions may help solve some of the more common errors.

- **Please load mouse driver.** During start-up, the *i*X-server looks for mouse input from a previously loaded driver and will halt if the driver has not been loaded. It is recommended that the appropriate driver for the installed mouse is loaded at start-up in the AUTOEXEC.BAT file.

- **No network driver loaded.** The network support software has not been loaded or has failed to initialise properly. Make sure problems due to the network are fixed before running *i*X. A quick verification of network setup is to use the 'ping' network command to test access to your chosen XDM server machine. If this fails then refer to the network package manual.

- **Unstable X video display**. If after adjusting horizontal and vertical controls on your monitor, the *i*X card does not produce a stable video display, then the installed parameter settings for the video generator are potentially unsuitable for the chosen monitor. It is possible using the 'ixcheck' diagnostic supplied with the product, to adjust the parameters to suit individual monitors, but this requires knowledge of both the monitor involved and the characteristics of the video generator chip, the IMS G332. If this is desired, then the documentation given in Appendix B should be consulted.

- **Get host by name failure.** If the XDM server named in your IX.CFG file cannot be found on the network, then the *i*X-server will halt with an error Check that the name given is valid and that the network software has an appropriate 'home server' configured. Check this by using the 'ping' command.

- **Blank X screen.** If the server initialises properly, but simply displays a grey weave pattern (with the cursor responding to mouse movement), then the *i*X-server is not making contact with an X display manager on your named server machine. Check that the XDM daemon is in fact running and that the server has access rights (via the X host control mechanism) to the *i*X display.

- **Cursor does not move.** If the display initialises properly but the cursor does not respond to mouse movement, check that the correct driver has been loaded for the particular type of mouse. Also check that the mouse is correctly plugged into the PC.

- **Wrong cursor speed.** If cursor movement is too fast or too slow in response to mouse motion, change the pointer acceleration parameters in the IX.CFG file. The values are described later in this document.

- **Fatal X-server error.** The X-server will fail to operate correctly if its font paths are not appropriately set up. Check that the specified paths each contain a FONTS.DIR and a FONTS.ALI file.

- **Clients cannot open windows on display**. If the server initialises properly but clients are refused access to the display then:
  –check the host control list, if any, in the XHOST.0 file.
  –check that the network software has been installed allowing sufficient connections to be
  established
  –check that enough file descriptors have been set up in the CONFIG.SYS file. The figure should be set to the maximum of 20.

8 *i*X configuration file

The run-time configuration of the *i*X-server is controlled by two ASCII format files containing keyword/value combinations. These files can be edited at will to change the contents. This editing is a simple task once the significance of the parameters is explained. The contents of this chapter will attempt to do this. Major changes may result in the *i*X-server working incorrectly when booted - any unrecognised words or values are simply ignored and defaults are substituted. To avoid problems of this type, always keep a copy of the last working version of the file when editing the contents. In the event of disaster, the old copy can be renamed to replace the faulty version.

When the *i*X server is started, it first reads a IX.CFG file followed by the IX.CRT file.

8.1 Monitor configuration

The first set of keyword value pairs describe the parameters of the Colour Vdeo Controller device on the *i*X card. These values will be held in the IX.CRT file.

screen-width, screen-height

The display resolution expressed in pixels. The accepted range is from 640×480 to 1024×768. The ratio of width to height should be 4:3 in most cases to preserve the correct aspect ration for many application displays.

g3xx-xxxxxx-val

Defines the values written to the CVC registers to initialise and control the display timings. The full set of these parameters are calculated according to the display characteristics and the resolution required. The full set are:

```
g3xx-controla-val
g332-controlb-val
g3xx-boot-val
g3xx-halfsync-val
g3xx-backporch-val
g3xx-display-val
g3xx-shortdisplay-val
g3xx-broadpulse-val
g3xx-vsync-val
g3xx-vblank-val
g3xx-vdisplay-val
g3xx-linetime-val
g3xx-linestart-val
```

```
g3xx-meminit-val
g3xx-transferdelay-val
g3xx-mask-val
g3xx-saver-mask-val
g332-cursorstore-val
g332-vpostequalize-val
g332-vpreequalize-val
g332-cursorstore-size
```

Sample values for these registers are in the CFG subdirectory or installation directory. Refer also to the IMS G332 overview in Appendix B and the IMS G300 documentation [4].

ix-graphicscontrol-val

Select the graphics options written to the control register on the IMS B020. Default is **0xc3.**

ix-hotkeymuxselect-val

Controls the multiplexor setting on the IMS B020. Default is **0xffffffff.**

ix-clockselectport-val

Controls the source of the IMS G332 clock. Default is **0x38.**

ix-interruptcontrol-val

Selects the source of IMS T425 events. Default is **0x01.**

8.2 X-server configuration

All values from this point should be contained in the IX.CFG file, such that a change in display hardware need not force re-specification of X-server characteristics.

ix-display-hardware

Set to **b020** for the *i*X hardware to indicate the CVC device type. Set to **b419** for other display devices based on standard IMS B419 designs.

g3xx-register-base

Defines the hardware address of the CVC device. Set to **0x40000000** for the *i*X card.

g332-topofscreen-val

Value written to IMS G332 top of screen register. Default is **0.**

g3xx-vram-base

Defines the hardware address of the video RAM for the card. Set to **0xc0000000** for the *i*X card.

g3xx-vram-end

Set to the last valid address in the video RAM space of the card. Set to **0xc00c0000** for the *i*X card.

g3xx-vram-size

Set to the size of the video RAM space of the card. Set to **0x100000** for the *i*X card.

g3xx-blue-shift, g3xx-green-shift, g3xx-red-shift

Set to the bit offset of the colour description fields within the CVC device. For the IMS G332 device these are **0, 8, 16** respectively.

font-path

The font path is used to locate the individual font files used by the server for text display. Each directory specified, in a comma separated list, must contain a FONTS.DIR file and a FONTS.ALI file listing the other font types compiled in the directory. The directory specified may be on a local or remote disk drive. The default setting is **IX/FONTS/MISC/, IX/FONTS/75DPI/, IX/FONTS/100DPI/** on a local disk.

rgb-file

The path and filename of the file used to describe the named colour database, without its three letter extension code. The default is **IX/RGB**

permit-old-x-bugs

Sets a mode in which the X-server retains compatibility with the characteristics of the X11R3 code, even when this was incorrect. Default is **no.**

retain-x-settings

Sets a mode in which the X-server will return to its default settings after the end of each session. A session is determined to have ended when the last client terminates and the X-server is reset – equivalent to logging out of XDM session. Default is **no.**

backing-store, save-unders

The use of backing store and save unders enables faster response to screen updates since portion of unseen windows can be retained in server memory. The options here are **yes** or **no**. The disadvantage of using these options is additional use of server memory.

screen-saver-implemented

The screen save options allow control over behaviour of the X-server when no input events are received for some interval. This option is set by default to **using-control.** The other possible values are **using-mask** and **no.**

screen-saver-interval

Sets preference for time before screen saver cuts in. The units are milliseconds. Set by default to **60000.**

pointer-numerator, pointer-denominator, pointer-threshold

These parameters set the response of the cursor to mouse movements. Different mice will have different sensitivities and require adjustment of these values. The cursor will move 'numerator/denominator' times as fast when it travels more than 'threshold' pixels in a short time. This way, the mouse can be used for precise alignment when it is moved slowly, yet it can be set to travel across the screen rapidly when desired. Default values are **2, 1, 4.**

8.3 XDM configuration

The values in this section are also set within the IX.CFG file.

virtual-terminal-at-reset

Sets default action of server when started. Set to **xdm** to attempt to find an XDM server according to the following parameters. If successful then a login window will appear on the display after a few seconds. Set to **none** to disable XDM operation.

xdm-init-state

Determines the method of finding the XDM server machine. Set to **query** to directly ask a named machine to service the terminal. Set to **broadcast** to ask any machine to respond.This number is an allocated address for XDM use.

xdm-one-session

If set to **yes**, then after the last client in the current session has terminated (i.e. the user has logged out) then XDM will not be re-invoked to re-display a login window. Set to **no** for *iX* to cycle round and display the XDM prompt once again.

xdm-display-class

The display Class is used to identify distinct sets of terminal types to the X display management daemon. The classes can be used to distinguish different types of XDM behaviour. Set by default to **INMOS-IX.**

xdm-magic-cookie

Provides an initial Magic number for use by the X display manager. When not using this feature of XDMCP set to **0.**

xdm-display-id

Used by X display manager to distinguish displays of a given class. Set to **0.**

xdm-host-name

Sets the network name of the host machine which will be asked to manage the display using XDMCP. The machine must be running an XDM daemon to respond. If not using XDM then set to **unknown.**

8.4 PC configuration values

These setting will be contained in the IX.CFG file and are used by the PC server side before installing the X-server software.

ix-board-address

Set to the hex address of the card in the PC I/O bus. Set to **#150** by default.

ix-memory-size

Set to the value that represents the amount of server memory on the card. Default value is **#400000.**

ix-quit-key

Set to the keyboard scan code of the key used for termination of the ***i*X** program. Default value is **0x046.**

ix-shell-key

Set to the keyboard scan code of the key used for hot-key to DOS during ***i*X** operation. Default value is **0x146.**

ix-btl-file

Set to the filename of the transputer executable used by ***i*X**.Default value is **\IX\IX.BTL.**

ix-trace-level

This is a debug flag. Default set to **1.**

hosts-file

Defines the location of the X-server host access control file, if it exists. This file can be used to list host machines from applications can display on the X-server screen. Default is **\IX.**

9 Font files, rgb files and servers

This chapter gives an introduction to the run-time support files used by the X-server and outlines how the files could be maintained in a centralised location under a UNIX operating system.

9.1 Font file overview

Fonts are descriptive binary format files that describe the shape and spacing of individual families of letters, numbers and symbols. Font descriptions are usually stored as individual files in directories accessible to the X-server at runtime, and are read when an application requests the use of a particular font.

The list of directories in which the server looks when trying to open a font is controlled by the X-server font path selection. For *i*X the font path is set in the local configuration file and, when the fonts are installed locally on the PC, the paths point to a subdirectory of the installation tree. The font path can be changed at any time using an X client utility 'xset', but three default directory paths are usually set up at installation time.

- IX\FONTS\MISC - This directory contains several miscellaneous fonts that are useful on all systems. It contains a very small family of fixed-width fonts (6x10, 6x12, 6x13, 8x13, 8x13 bold, and 9x15) and the cursor font. It also has font name aliases for the commonly used fonts, fixed and variable.
- IX\FONTS\75DPI - This directory contains fonts contributed by Adobe Systems Incorporated , Digital Equipment Corporation and by Bitstream Incorporated for 75 dots per inch displays. An integrated selection of sizes, styles, and weights are provided for each family.
- IX\FONTS\100DPI - This directory contains versions of the fonts in the 75dpi directory for 100 dots per inch displays.

Font databases are set up within these directories by creating the FONTS.DIR file as part of the directory. This file is the output of a utility program called 'mkfontdir' when it is run in the directory containing the compiled versions of the fonts (the '.snf' files) and is the primary file that the server reads to determine what fonts are available to it. The addition of a particular directory in the font search path will not be accepted un less it contains a valid FONTS.DIR file. Whenever new fonts are added to a directory, 'mkfontdir' should be rerun so that the server can find the new fonts. 'mkfontdir' is normally a UNIX based utility and is not supplied with the *i*X product.

9.2 Storing fonts under UNIX

To save space on the local PC disk, it may be useful to store font files on a remote machine and access the information across the network. With centralised font storage, updates and additions to the fonts families will also be rather easier. To do this, it will be necessary to create a distinct set of directories for storage of *i*X fonts on a UNIX server machine and make that directory tree accessible across the network, typically using the facilities of the Network File System (NFS). Networks which are running X already may have an existing font installation. Unfortunately, this will probably be in-appropriate since the font files for particular servers may differ considerably in their format. The key to this difference lies in the conversion process from the generalised font distribution format (typically files ending in '.bdf) to those actually used by the server (with a '.snf' suffix). All machines with X-window system installed will have a set of original distribution font files and a set of directories with server fonts files used by the existing X-servers in the system. For typical installations the server font files will be stored in the path /usr/lib/X11/fonts/. To make a set of font files for the *i*X-server available across the network, install a subdirectory in this area called *i*X and store font files within this structure below the subdirectory. (/usr/lib/X11/fonts/IX/fonts/misc/ and so on). The fonts can either be obtained from the *i*X distribution disks or can be compiled from the original font description files supplied with an X-windows release using the 'bdftosnf' utility. The options required to compile for *i*X are:

```
bdftosnf -L -l -p4 -u4
```

When each font directory is complete, run mkfontdir to construct the fonts.dir file and make the directory readable and mountable for remote machines.

On the PC, use the remote mount command to install an additional disk drive on the system that is mapped to the fonts directory that has just been created. A change to the font path used by *i*X can then be made such that the *i*X-server uses the remote file system for reading the fonts.

There are two complications that make this a slightly less than straightforward process. They both arise from the fact that PC filename conventions are less flexible than those of UNIX and therefore PC based NFS implementations sometimes need to invoke a mapping between a UNIX filename and its representation on the PC. This must be avoided if the *i*X-server is to function correctly with remote mounted fonts.

- All filenames on the UNIX machine must be all lower case. Some of the font files distributed with X whilst restricting themselves to 8 character names with a three character extension, used a mixture of cases within their names. These can be trivially converted to use all lower case letters without conflict.

- The file 'fonts.alias' within a UNIX font directory must be renamed or copied to a file 'fonts.ali' to match DOS conventions

9.3 RGB files

Additional configuration of X-server characteristics at run time can be made by altering its colour database. The database, stored in a file 'rgb.pag', is a mapping of colour names to RGB intensity and is highly subjective in nature. In particular, different display monitors may appear to reproduce very different colours when given the same RGB data. The file can be changed at will by editing its text form, 'rgb.txt', and converting the data to binary form using the rgb utility. Again this utility is normally run in a UNIX environment. It may be appropriate to store the rgb files in the same remote file system as the fonts. This will ease maintenance and update problems. It is suggested that the file be stored in the iX directory as created above. The local configuration file will need to be changed to reflect the search path for this file.

10 Using the transputer expansion slots

The IMS B020 is one example of a set of TRAM motherboards, conforming to a standard transputer expansion architecture. The TRAM system is fully described in *Dual-in-Line Transputer Modules* [1] published as INMOS technical note 29 and also included in *The transputer development and* **iq** *systems databook* [2]. For information on the transputer itself, refer to the *Transputer Databook* [3].

The IMS B020 is designed to be compatible with a more generic PC motherboard, the IMS B008. All transputer development tools and driver software that has been written for the IMS B008 will work with the IMS B020. This compatibility includes the provision of expansion sockets to accept additional TRAMs and the capability to drive links and sub-system signals to external transputer systems through a back panel connector.

10.1 INMOS development toolkit

The transputer architecture is designed to enable the easy construction of multi-processor application systems. To support this, INMOS has developed a system of software development tools that support both the basic processor architecture and the TRAM system that supports it. Single or multi-processor application programs can be written in C, C++, Fortran or the INMOS designed language occam. The toolset supports compilation, linking, and configuration over a network in all languages, as well as access to I/O facilities via a comprehensive run-time library. Multi-processor debugging is supported at the source level in run-time or post-mortem operation. For further information on these development tools see the [2] *iq systems Databook* or contact an INMOS or SGS-THOMSON sales office.

A variety of other vendors have also developed run-time and development systems for the transputer. These include Ready Systems (VRTX), Perihelion (Helios) and JMI (C-exec). All of these vendors support applications written for single and multi-processor systems including the IMS B020.

10.2 Adding TRAM functions

The link and subsystem wiring on the IMS B020 is described in Appendix A It allows the addition of one size 4 or two size 2 TRAM functions onto the basic IMS B020 in addition to the extension of both links and control signals to external transputer modules or subsystems. The additional TRAMs can be used for computational or I/O functions. The transputer on the IMS B020 can also be used for any function, but clearly is most suitable for graphic display purposes. This transputer, in addition to other functions can reset, analyse, and monitor error signal of the additional TRAMs independent of host activity.

Note that there is no IMS C004 linkswitch capability on the IMS B020.

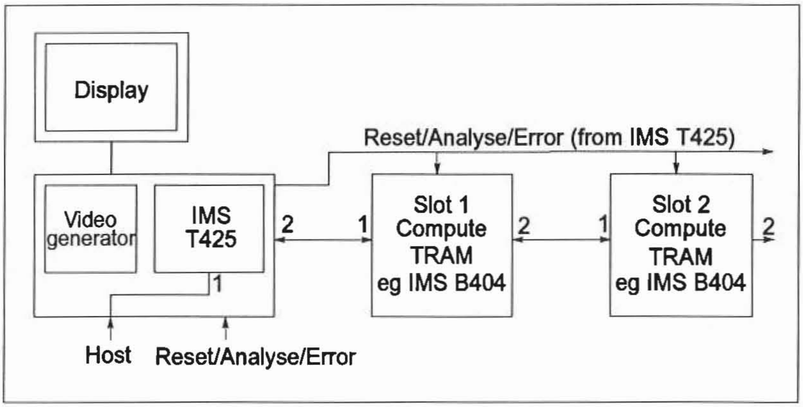

Display
Reset/Analyse/Error (from IMS T425)
Video generator
IMS T425
2
1
Slot 1 Compute TRAM eg IMS B404
2
1
Slot 2 Compute TRAM eg IMS B404
2
1
Host
Reset/Analyse/Error

Figure 11.1 Using additional TRAMs for compute intensive graphics generation

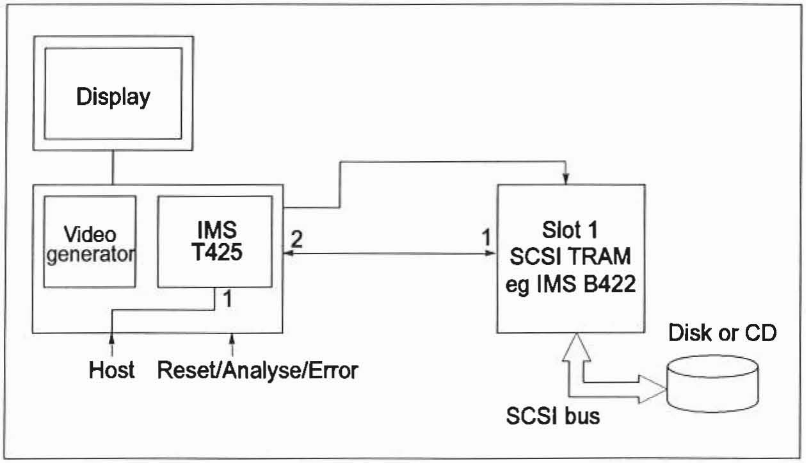


Figure 11.2 Using an I/O TRAM for multi media applications based on disk or CD images

11 References

1 *Dual-in-line transputer modules* published as INMOS technical note 29

2 *The transputer development and iq systems databook,* INMOS Ltd, second edition 1991

3 *The transputer databook*, INMOS Ltd, second edition 1989

4 *The graphics databook,* INMOS Ltd, second edition 1990

5 *X-window system user's guide (Volume 3)*, Quercia and O'Reilly, O'Reilly & Associates Inc, Motif edition 1991

Appendices

A IMS B020 engineering data

A.1 Introduction

The IMS B020 is a transputer based IBM PC/AT compatible high performance colour graphics board. It has been specifically designed to permit a PC to implement X-Terminal functionality at low cost with high resolution colour graphics provided by an IMS G332 Colour Video Controller (CVC). In addition, the IMS B020 may also be used as a conventional PC transputer development environment in place of an IMS B008 with the added benefit of high resolution graphics included. It can also accommodate a range of size 1, 2 and 4 INMOS TRAnsputer Modules (TRAMs).

The frame store consists of 1Mbyte of dual ported Video RAM which supports displays of arbitrary resolution at 8 bits/pixel. The resolution of the system is programmable and is only limited by the CVCs maximum dot rate and the access time of the serial port on the VRAM. The IMS B020 supports a dot rate up to 85 MHz. The CVC is configured by the on board IMS T425 transputer. The transputer has access to the VRAM and 4MBytes of ZIP DRAM. In addition a further 1 or 4 MBytes of DRAM in the form of Single In-line Memory Modules (SIMM's) may be added by the user providing a maximum DRAM configuration of 8MBytes The processor can be used to implement graphic primitives directly or as an intelligent channel, receiving image data from other transputers via its bidirectional links.

A.2 Detailed description

A.2.1 IBM PC bus interface

The PC bus has become a de-facto standard after appearing in the IBM PC. Since then a large number of other machines have become available that incorporate the PC bus. The IMS B020 has been designed to work when plugged into either a PC/AT bus slot or a PC bus slot. Table A.1 illustrates the I/O memory map that the IMS B020 presents to the PC.

Base Address	Register	Comments
+ #00	IMS C012 Input data register	
+ #01	IMS C012 Output data register	
+ #02	IMS C012 Input status register	
+ #03	IMS C012 Ouput status register	
+ #10	Reset/Error register	The following occurs when reset is asserted: • Reset IMS T425 transputer • Reset IMS C012 • Reset subsystem Reset and Analyse ports • Reset IMS T425 Event from PC register • Reset IMS G332
+ #11	Analyse register	
+ #18	Hardware reset register	Toggles D0 (high pulse) to generate a hardware reset on the IMS B020. Same effect as a reset signal on the PC bus. The following occurs: • Reset output selection to VGA loopback • Reset Linkspeed selection to 20MHz
+ #19	Linkspeed register	D0 selects 10MHz or 20MHz for linkspeed (IMS T4 links 1, 2+3 and TRAMs). Initialise to 20MHz at power on and board reset. • D0 = '0' for 10MHz • D0 = '1' for 20MHz
+ #1C	Boot link and system services selection	D0 Selects whether the IMS T425 or TRAM slot 0 is connected to the PC • D0 = '0' for link to IMS T425 • D0 = '1' for link to TRAM D1 selects whether th PC host or the IMS T425 controls the system services for TRAMs 1 and 2 and external transputers • D1 = '0' for PC host • D1 = '1' for IMS T425
+ #1D	Version register	Read 8 bit code D0-D7 (stored in PAL)

Table A.1 PC bus memory map

A.2.2 Host system services

A port is provided by the PC bus interface to allow software on the PC to provide 'system services' to TRAMs mounted on the IMS B020. The port appears as two registers in the I/O map of the PC. See section A.3for a detailed description of the system services provided by the IMS B020.

A.2.3 Link Speeds

The link speeds of the TRAMs and the IMS C012 can be set to 10 or 20 Mbits/s under software control via the link speed register in the PC I/O address space. The default link speed is 20Mbits/s.

A.2.4 Transputer and graphics

The memory map of the IMS T425 (figure A.1) and IMS G332 is shown in the following sections. Note that the contiguous DRAM memory space at the low end of memory is extendable using standard SIMMs.

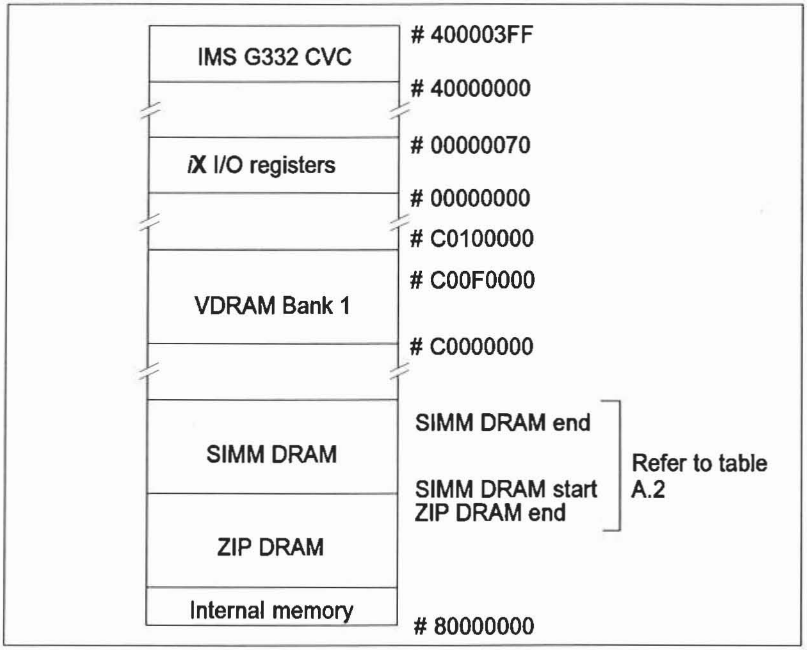


Figure A.1 IMS T425 address map

ZIP DRAM size (MByte)	ZIP DRAM end	SIMM DRAM start
4	#803FFFFF	#80400000
SIMM DRAM end = SIMM DRAM start + [1 or 4] MBytes		

Table A.2 DRAM start and end registers

A.2.5 *i*X I/O registers and description

Register group	Hex address	Register	Read/ Write
*i*X subsystem registers	#00000000	Subsystem Reset/Error	R/W
	#00000004	Subsystem analyse	W
*i*X graphics control register	#00000060	Graphics control	W
*i*X interrupt registers	#00000070	Interrupt control	R/W

Table A.3 *i*X registers

*i*X subsystem registers

A standard transputer subsystem port as found on TRAM modules (eg IMS B404) is provided at the standard address as detailed in section A.3.

*i*X graphics control registers

Two registers are provided to configure the graphics subsystem of IMS B020 in addition to those provided by the IMS G332.

Graphics control register
The graphics control register is described in table A.4. All bits are set to 0 on board reset.

	Bit 1	Bit 0
D0	VSync true	Invert G332 VSync
D1	HSync true	Invert G332 HSync
D2	Interleaved addressing	No interleaved addressing
D6	G332 output selected	VGA loopback mode
D7	G332 as sync master	Sync mux-G332 as sync slave

Table A.4 Graphics control register

Bits 0 and 1 are used to invert VSync and GSync respectively. The user should consult their monitor specification when deciding if this is required.

Bit 2 is used to select interleaved addressing mode. This is required for 15/16 bit colour operation for boards upgraded to support 2MBytes of VRAM. Only 1MByte of VRAM is fitted to the IMS B020 as standard.

Bit 6 is used to switch either VGA or IMS G332 video output to the monitor via the multiplexor. This bit must be set in conjunction with bit 7.

Bit 7 is used to determine whether the IMS G332 is sync master or sync slave mode. This bit must be set in conjunction with bit 6.

Note
All unused bits are reserved and should be set to 0.

iX interrupt control register

The on-board transputer may be interrupted from any one or more of 4 external events. These events are listed in table A.5. In order to determine which are the interrupt sources, the transputer must interrogate the Interrupt control register by reading its contents. In order to enable or disable interrupt sources the transputer may write a mask as shown in table A.6.

Event on transputer generated as positive pulse from:

- VSync low
- HSync low

Read mode- Status:

Bit	Set to 1
D0	G332 VSync
D1	G332 HSync

Table A.5 iX interrupt control register-read mode

Write mode- Mask:

	Bit 1	Bit 0
D0	Enable G332 VSync	Disable G332 VSync
D1	Enable G332 HSync	Disable G332 HSync

Table A.6 iX interrupt control register-write mode

Note
All unused bits are reserved and should be set to 0.

A.3 System Services

On all INMOS board products the term 'system services' refers to the collection of the reset, analyse, and error signals. On the IMS B020 the system services for the on board transputer or the TRAMs in slot 1 and 2 are connected to the system services controlled by the PC bus interface. System services for the other TRAMs can be connected to the PC, or to the subsystem port of on-board transputer.

Reset

Resets the transputer, and other circuitry. **Reset** should be asserted for a minimum of 100ms. After **Reset** is de-asserted a further 100ms should elapse before communication is attempted on any link. After this time, the transputer on this TRAM is ready to accept a boot packet on any of its links.

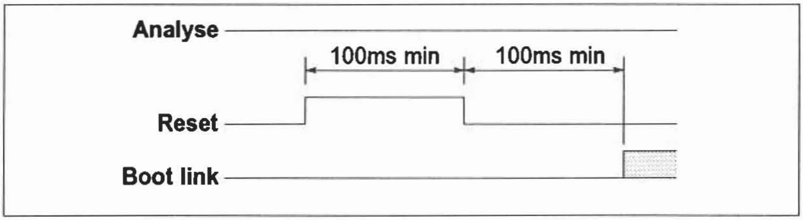


Figure A.2 Reset timing

Analyse

This is used, in conjunction with **Reset**, to stop the transputer. It allows internal state to be examined so that the cause of an error may be determined. **Reset** and **Analyse** are used as shown in figure A.3. A processor in analyse mode can be interrogated on any of its links.

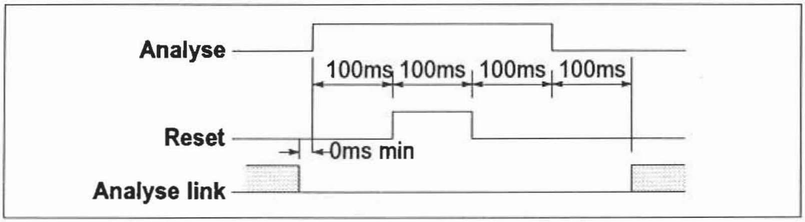


Figure A.3 Analyse timing

notError

An open collector output which is pulled low when the transputer asserts its Error pin. **notError** should be pulled high by a 10kW resistor to **VCC**. Up to 10 **notError** signals can be wired together. The combined error signal will be low when any of the contributing signals is low.

A.4 TRAM Slots

The IMS B020 has two locations for TRAM expansion. Each slot can accommodate a size 1 or 2 TRAM. Alternatively a single size 4 TRAM may be fitted. Each of the two slots on the IMS B020 has connections for four INMOS links. Links are numbered 0 to 3 and slots are numbered 1 to 2 . TRAM slots 3 and 4, although present are for mechanical fixing of size 2 and 4 TRAMs only. Power and ground are, however, supplied to all TRAM sockets. For exact specifications of different sized TRAMs refer to [1] or [2].

A.4.1 Topology

The transputer link topology of the IMS B020 is shown in figure A.4. The link switch function permits TRAM 1 to act as the 'root' transputer instead of the on-board IMS T425. This is useful when using the INMOS interactive debugger in conjunction with graphics software running on the IMS T425. In such cases a TRAM in slot 1 can be used to boot and run the debugger software.

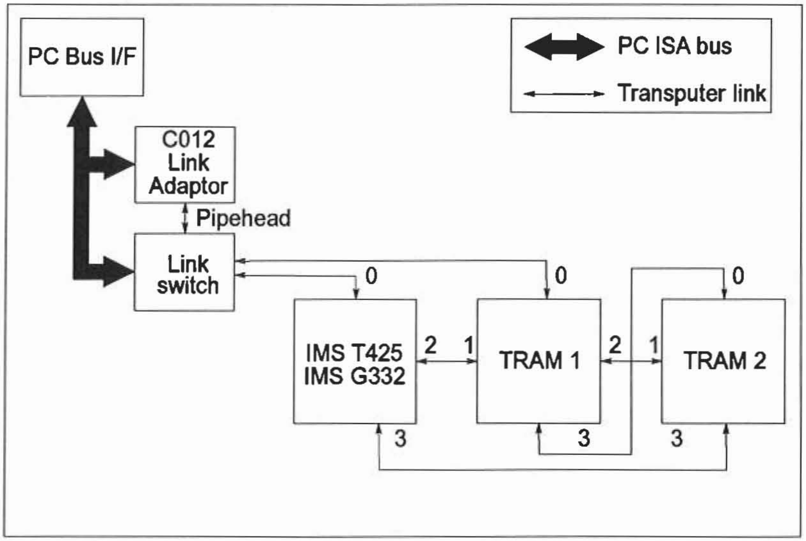


Figure A.4 Board topology

In instances when not all the TRAMs are fitted, to maintain the pipeline connection, *pipejumpers* must be used, plugged into each un-occupied slot, or the TRAM covering that slot. These pipejumpers connect link 1 to link 2 of the same slot. They are plugged into the pin 1 end of the TRAM slot, with the triangle marked on the corner. The pipejumpers have a mark on them which must be pointing towards the pin 1 marker triangle.

A.5 Jumpers

There are a minimal number of jumpers on the board, so as to ensure against software incompatibilities. In general it should not be necessary to change any of the factory defaults except when either upgrading with SIMM memory modules or changing the PC I/O base address.

Jumper	Description	Notes
J1	1M×4 ZIP DRAM	Factory fitted
J2	256K×4 ZIP DRAM	Not fitted
J3	1M×8 SIMM DRAM	4MByte expansion
J4	256K×8 SIMM DRAM	1MByte expansion
J5	IMS B020 address	J5 -#150 J6 -#200 J5+J6 -#300
J6	IMS B020 address	J5 -#150 J6 -#200 J5+J6 -#300
J25	5MHz PLL clock to IMS G332	Factory fitted

Table A.7 Jumper configuration

Note: # = hexadecimal address.

A.6 Memory expansion using SIMMs

Two sizes of expansion SIMM are supported – 256kbits×8 and 1Mbit×8. Four SIMMS will be needed to expand the memory space for the transputer and these should be inserted carefully in the sockets provided.

A.7 Rear panel connectors

A.7.1 Video connections

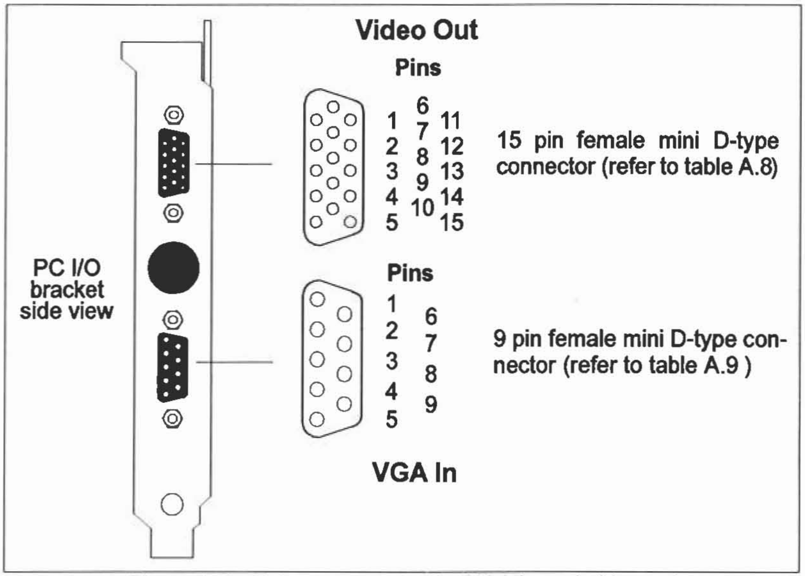

Video Out
Pins
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
15 pin female mini D-type connector (refer to table A.8)
PC I/O bracket side view
Pins
1 2 3 4 5
6 7 8 9
9 pin female mini D-type connector (refer to table A.9)
VGA In

Figure A.5 D-type connector for VGA in and video out

The video connections of IMS B020 are shown in figure A.5and tables A.8and A.9. The 15 pin mini D-type connector is the output to the monitor and conforms to the VGA specification. The 9 pin mini D-type connector permits the output of a VGA card to be looped through the IMS B020 via the video multiplexor.

Pin	Name	Pin	Name
1	Red	9	N/C
2	Green	10	GND
3	Blue	11	MonitorSense1
4	MonitorSense2	12	MonitorSense0
5	N/C	13	HSync
6	Red GND	14	VSync
7	Green GND	15	N/C
8	Blue GND		

Table A.8 15 pin **female** mini D-type connector **Video Out**

Pin	Name	Pin	Name
1	Red	6	Red GND
2	Green	7	Green GND
3	Blue	8	Blue GND
4	HSync	9	GND
5	VSync		

Table A.9 9 pin **female** mini D-type connector **VGA In**

VGA loopback cable

In order to facilitate the video connection between the users VGA card and the IMS B020, a VGA loopback cable is provided as shown in figure A.6. The 15 pin and 9 pin male D-type connectors of the cable are connected to the 15 way female connector of the user's VGA card and the 9 way female connector of the IMS B020 respectively.

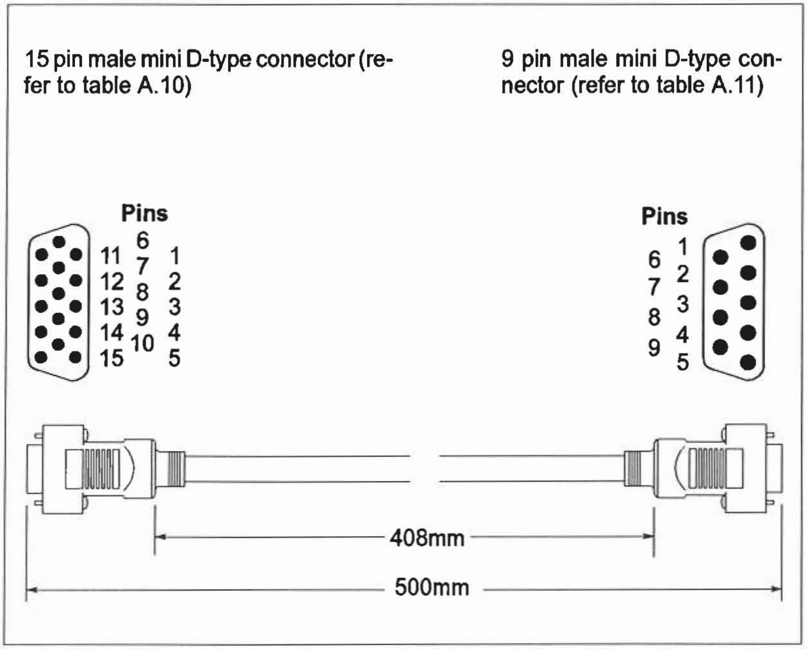

15 pin male mini D-type connector (refer to table A.10)
9 pin male mini D-type connector (refer to table A.11)
Pins
6
11 7 1
12 8 2
13 9 3
14 10 4
15 5
Pins
1
6 2
7 3
8 4
9 5
408mm
500mm

Figure A.6 VGA loopback cable

Pin	Name	Pin	Name
1	Red	9	N/C
2	Green	10	GND
3	Blue	11	GND*
4	N/C	12	N/C
5	N/C	13	HSync
6	Red GND	14	VSync
7	Green GND	15	N/C
8	Blue GND		
*** Pin 11 is connected to pin 10**			

Table A.10 15 pin **male** mini D-type connector

Pin	Name	Pin	Name
1	Red	6	Red GND
2	Green	7	Green GND
3	Blue	8	Blue GND
4	HSync	9	GND
5	VSync		

Table A.11 9 pin **male** mini D-type connector for **VGA In**

Parameter	Rating
Current rating	2A max.
Operating voltage	300V max.
Insulating resistance	100MW
Temperature rating	0-60°C
Contact resistance	15mΩ
Mating cycles	500 cycles

Table A.12 Cable specification

A.8 Mechanical details

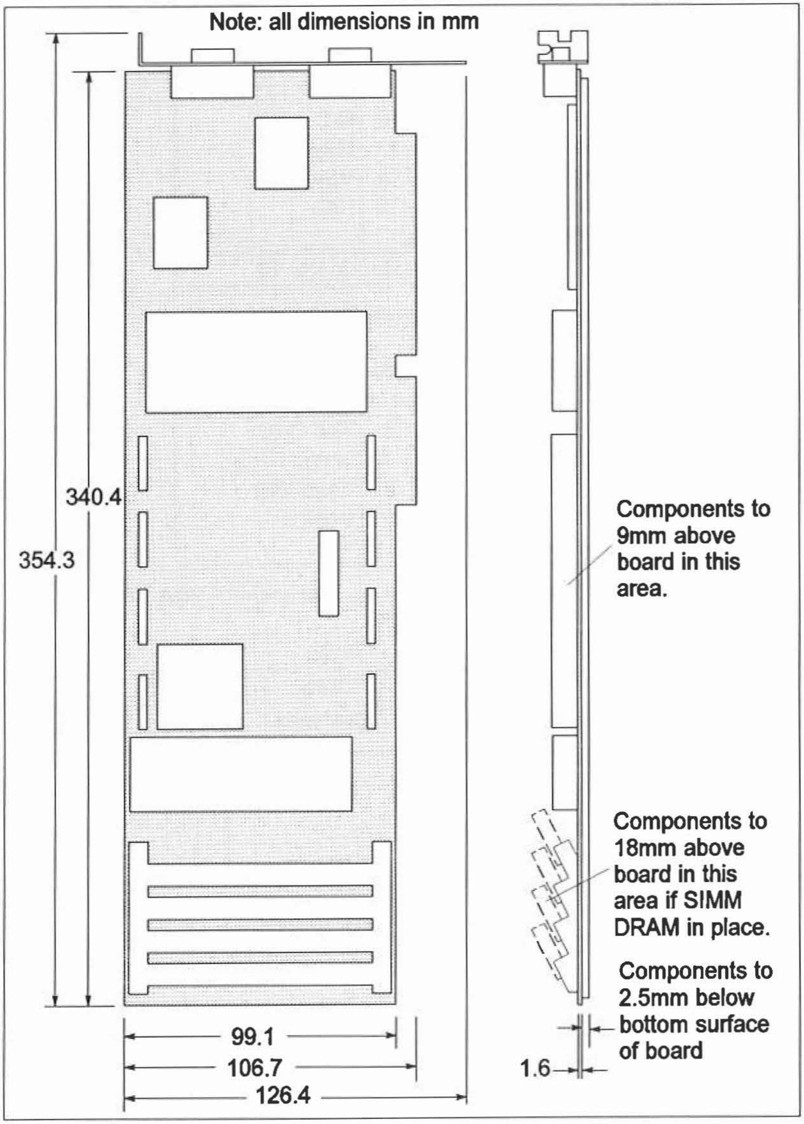

Note: all dimensions in mm
340.4
354.3
99.1
106.7
126.4
Components to 9mm above board in this area.
Components to 18mm above board in this area if SIMM DRAM in place.
Components to 2.5mm below bottom surface of board
1.6

Figure A.7 Mechanical drawing of the IMS B020

The IMS B020 is a PC/AT format board and is nominally 354mm by 126mm by 22mm overall. The PCB thickness is nominally 1.6mm. The board includes a metal

PC I/O bracket on which the rear panel connectors are mounted. This bracket serves two functions to ensure the board is held rigidly at the edge connector end and to maintain the integrity of the shielding of the PC. To enable the bracket to perform these functions it must be securely fixed to the backpanel metalwork of the PC by a screw passing through the slot on the right hand side of the bracket, as viewed with the board towards you in the PC. A mechanical drawing of the IMS B020 is shown in figure A.7.

The IMS B020 weighs 344g without any TRAMs or SIMMs fitted.

A.8.1 Thermal Information

The IMS B020 with no TRAMs or SIMMs installed will dissipate not more than 4.3W.

When installing the IMS B020 in a PC , consideration needs to be given to cooling airflow not only across the IMS B020 itself but also any TRAMs fitted to it. It is the responsibility of the user to ensure that the operating environment limits for the IMS B020 listed in table A.13 are not exceeded. This will not occur as long as there are not a large number of high dissipation boards also present in the PC.

To ensure good airflow in the PC the blank backpanels should be present in any slots that are empty.

A.8.2 Operating and Storage Environments

The IMS B020 is designed to be operated and stored in the environments in table A.13.

Parameter	Operating	Storage
Ambient air temperature	0 to +50°C	-55 to +85°C
Relative Humidity	95% non condensing	95% non condensing
Thermal Shock	<0.08°C/s	<0.15°C/s
Altitude	-300 to +3000m	-300 to +16000m

Table A.13 Environmental details

A.9 Electrical details

The IMS B020 requires a +5 V dc supply which must be between 4.75V and 5.25V with less than 50mV peak-peak noise and ripple between dc and 10MHz. The IMS B020 does not incorporate protection against incorrect power supplies. Major damage will result from connecting a supply to the board which is outside its power supply range.

Voltage supply	Typical current consumption
-5V	90mA
+12V	20mA
+5V	1960mA

Table A.14 Worst case power consumption

B IMS G332 overview and programmer's model

For complete technical information please refer to the IMS G332 datasheet, INMOS document number 42 1461 02. For convenience, applicable parts of the IMS G332 datasheet are reproduced below.

B.1 Introduction

The IMS G332 provides all the necessary functions to control real-time operation of a raster scan video display, using dual-ported video RAMs.

The device consists of a 32 bit variable format multiplexed pixel interface, a programmable video timing generator (VTG), a 256 location colour lookup table (LUT) with variable addressing modes, triple 8 bit video Digital-to-Analogue Convertors (video DACs), a 64×64×2 bit cursor store and 3 location cursor LUT, a programmable cursor positioning/insertion controller, a video memory control system and phase-locked loop clock generator.

A brief description of each functional block follows, for more detailed information please refer to the relevant section in the datasheet.

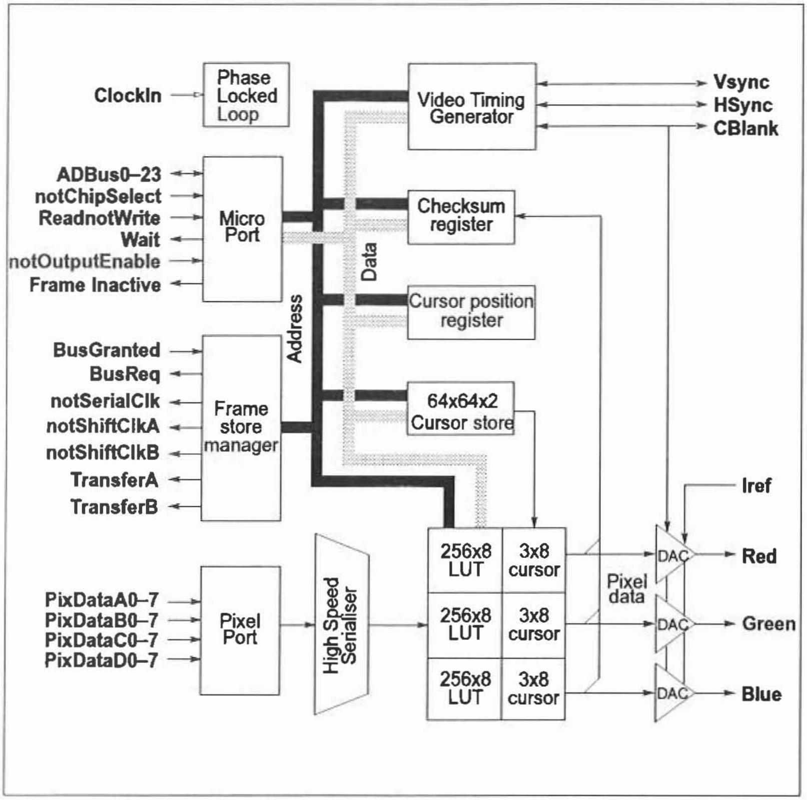


Figure B.1 IMS G332 Block Diagram

B.2 Register function reference guide

Register	Word address (hex)	Page No.	Description
Boot Location	#X000	78	Startup location to which must be written the clock multiplication factor, whether PLL or x1 mode, and the 32/64 bit address alignment selection.
Screen Description Registers	#X021 to #X02B		Read/write registers containing the screen description parameters. These are accessible only when the timing generator is not running.
Top of Screen #1	#X02C		Read/write register giving ability to reprogram the top of screen pointer while the VTG is disabled.
Framestore description Registers	#X02D, #X02E		Read/write registers containing effective VRAM register length and DMA latency figures.
Mask Register	#X040	69	24 bit pixel address mask register. Read/write accessible at all times. (Operates only on pseudo colour pixels)
Control Register A Control Register B[1]	#X060 #X070	76	Read/write control registers contain configuration information. Unassigned bits must be written with zero and are not valid on read. Read/write accessible at all times.
Top of Screen #2[2]	#X080		Read/write register giving ability to reprogram the top of screen pointer while the VTG is enabled.
Cursor palette	#X0A1 to #X0A3	65	3x24 bit cursor colour registers. Read/write accessible at all times.
Checksum registers	#X0C0 to #X0C2	69	RGB frame checksums.
Cursor position	#X0C7	65	24 bit register storing the x-y position of the cursor.
Colour Palette	#X100 to #X1FF	67	256 locations of 24 bit colours read/write accessible at all times.
Cursor store	#X200 to #X3FF	65	512 locations of 16 bit words, each containing 8 packed 2-bit pixel colour values.

N.B. All other addresses in the range are reserved and must not be written to.

[1] Control Register B is for future expansion only. For forward compatibility this register must not be written to after the start-up procedure has been completed.

[2] Both Top of Screen locations address the same physical hardware

The transfer address and top of screen pointers

The IMS G332 outputs a new 22 bit address on **ADBus2-23** during every transfer cycle it initiates irrespective of whether the micro port address alignment is set to 32 or 64 bit mode. The first address in each frame is specified in the Top of Screen register, which is programmed on startup but which can be modified at any time.

Note that this register appears at two separate locations, #X02C and #X080. #X02C is accessible only when the VTG is disabled, #X080 only when it is running.

The current row address is incremented by the amount specified in bits 13-12 of the control register, used in conjunction with the 'Interlace' bit (bit 2). These bits specify the VRAM step length and the screen format. Refer to Table B.1 for bit assignments.

Changes to the Top of Screen pointer become effective from the top of the subsequent screen (or field in an interlace system).

The framestore format for interlace is identical to that for non-interlace, the address offset for the second field being added automatically by the IMS G332. Address ordering depends on the standard selected. CCIR scans even lines first, EIA scans odd lines first.

In interlace, the first half of Line Zero is always blanked at the video DACs but the IMS G332 will clock the VRAM shift registers as though visible. This preserves compatibility between interlace and non-interlace.

Register Bit		Non-interlace	Interlace	
13	12	Increment	Increment	Second field offset
0	0	1	1	1
0	1	256	2	1
1	0	512	1024	512
1	1	1024	2048	1024

Table B.1 VRAM address increment as set by Control Register A, bits 12 and 13

B.2.1 Non-interleaved pixel sampling

The IMS G332 supports four depths of pixel using a non-interleaved memory system.

Register bits				Bits per	ShiftClk	MUX	Use of LUT
22	21	20	18	pixel	period	ratio	
0	1	1	0	8	1 SClk	4:1	Pseudo colour
0	1	0	0	4	2 SClks	8:1	Pseudo colour
0	0	1	0	2	4 SClks	16:1	Pseudo colour
0	0	0	0	1	8 SClks	32:1	Pseudo colour

Table B.2 Non-interleaved pixel modes

There are two sampling modes available in a non-interleaved memory system, standard and delayed, set by bit 19 in Control Register A.

B.2.2 Interleaved pixel sampling

In interleaved mode, the IMS G332 supports six depths of pixel as shown in Table B.3. Interleaved sampling is necessary to implement the true colour modes, and the remaining modes are supported to allow for a consistent framestore design.

Register bits				Bits per	ShiftClk	MUX	Use of LUT
22	21	20	18	pixel	period	ratio	
1	0	1	1	16	1 SClk	2:1	True colour
1	0	0	1	15	1 SClk	2:1	True colour
0	1	1	1	8	2 SClks	4:1	Pseudo colour
0	1	0	1	4	4 SClks	8:1	Pseudo colour
0	0	1	1	2	8 SClks	16:1	Pseudo colour
0	0	0	1	1	16 SClks	32:1	Pseudo colour

Table B.3 Interleaved pixel modes

B.2.3 Pixel formats

True colour modes (15 and 16 bits per pixel)

Each pair of pixel ports (A and B, and C and D) supplies a two-byte pixel value which is split into red, green and blue fields as illustrated below The ports are used in the order A,B then C,D. In 15 bits per pixel mode bits 0-4 are blue, 5-9 are green, and 10-14 are red. In 16 bits per pixel mode bits 0-3 are blue, 4-9 are green, and 10-15 are red.

The resulting Red, Green and Blue pixel values are then passed through the colour palette, which can act as a gamma correcting table, before being displayed by the DACs.

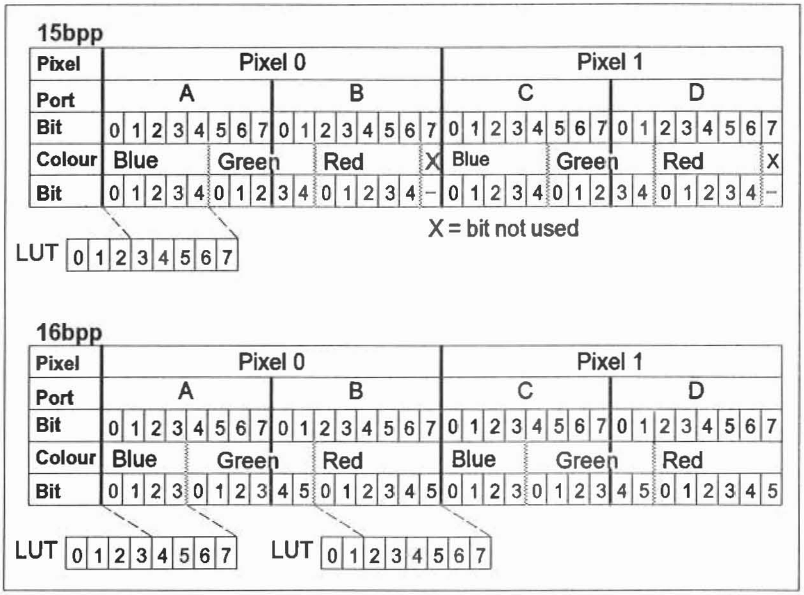


Figure B.2 Pixel mapping for 15bpp and 16bpp

Pseudo colour modes (1, 2, 4 and 8 bits per pixel)

The latched pixel data is serialised to the colour palette in the order Port A, B, C, and D. Each pseudo pixel is masked by the 24-bit mask register before acting as a pointer to a pre-loaded 24-bit colour in the colour palette.

Reduced pixel format sizes use the lower order address bits and therefore access the lower colour palette locations.

Pixel mode	Pixels per port	1st pixel	Last pixel	LUT address bits	LUT locations used
8	1	Bits 0-7	–	0-7	0-255
4	2	Bits 0-3	Bits 4-7	0-3	0-15
2	4	Bits 0-1	Bits 6-7	0-1	0-3
1	8	Bit 0	Bit 7	0	0-1

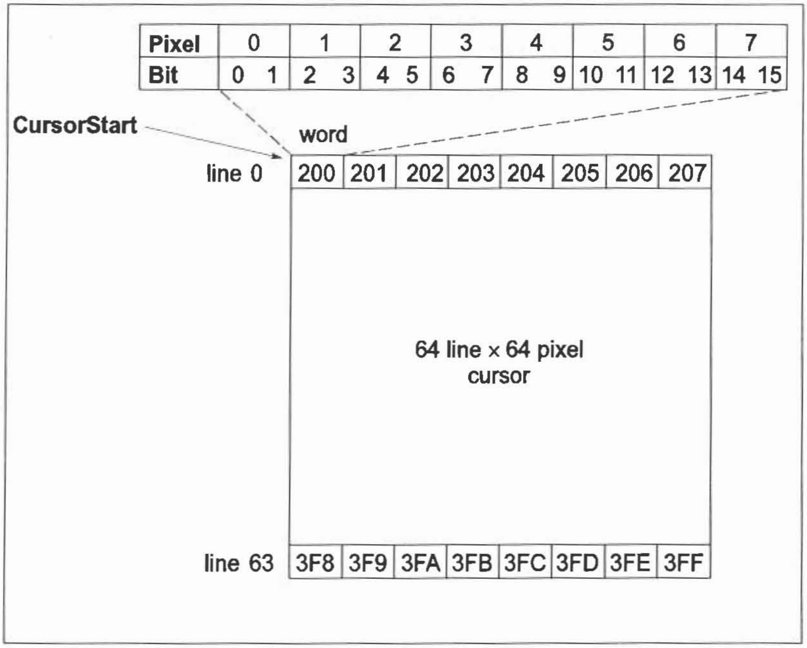
Table B.4 Pseudo colour modes

B.3 Hardware cursor

To reduce the amount of external hardware and limit software overhead, an on-chip cursor store is provided on the IMS G332. This is memory mapped through addresses #X200-3ff. The result is a 64 line×64 pixel three colour cursor, with a transparency option. This allows the underlying display to show through any cursor pixel. The cursor is enabled or disabled by bit 23 in Control Register A (Address #X060).

Cursor store

The cursor store is addressed as 512 × 16 bit words, each word being formatted into 8 pixels × 2 bits.



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

Figure B.3 Cursor store format

Cursor palette

The cursor palette consists of 3 × 24 bit colour registers.

Bit	23 – 16	15 – 8	7 – 0
Colour	Red	Green	Blue

Table B.5 Cursor palette colour register

The 2 bit cursor pixels address the palette registers as follows:

Pixel value	Format
0 0	Cursor transparent. Background colour displayed.
0 1	Colour from cursor palette register 1
1 0	Colour from cursor palette register 2
1 1	Colour from cursor palette register 3

Table B.6 Cursor palette colour register addressing

Cursor position

Cursor position is held in a single 24 bit register as an x-y location relative to the top left (0,0) of the screen. The position defined is that of the topmost, leftmost pixel of the cursor (**CursorStart**). The x-address and y-address are two's complement values in the range -64 to 2047.

Bit	23 12	11 0
	x–address (horizontal)	y–address (vertical)

Table B.7 Cursor position register

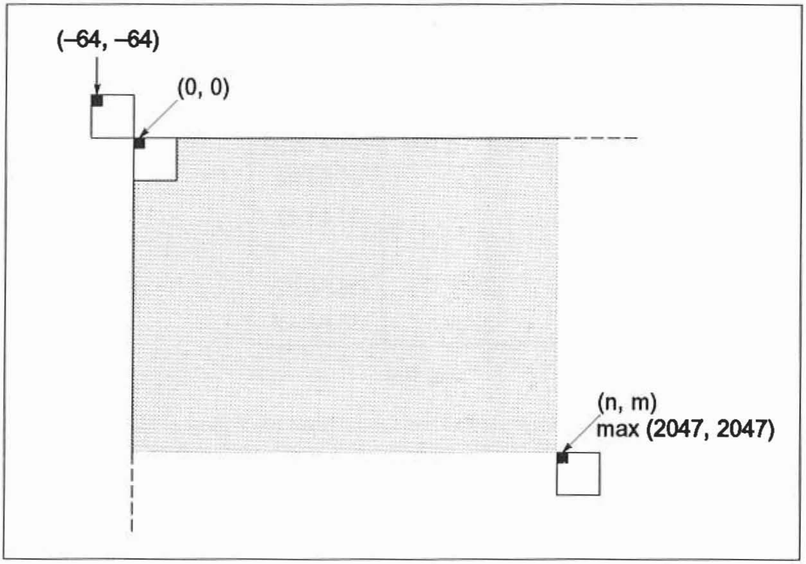


Figure B.4 Cursor position

Maximum X-address

The X position of **CursorStart** must never exceed n+1 where n is the number of pixels on a line. The absolute limit is n=2046

Maximum Y-address

The Y position of **CursorStart** may be any value up to 2047 irrespective of screen dimensions.

B.4 Anti-sparkle colour palette

The colour palette consists of 256 × 24 bit colour registers.

Bit	23 – 16	15 – 8	7 – 0
Colour	Red	Green	Blue

Table B.8 Colour palette - colour register

These read/write registers are located at Micro port address #X100 – #X1FF.

The colour palette is used both by the pseudo colour pixels (as an expansion table) and by the true colour pixels (each component can address a different location, at which there can be a gamma correction value).

If it is required to display an expanded pseudo colour picture and a true colour picture without reloading the palette in between, then the simplest solution is to write the colour number to each of the Red, Green and Blue portions of each colour palette register. This is shown in Table B.9.

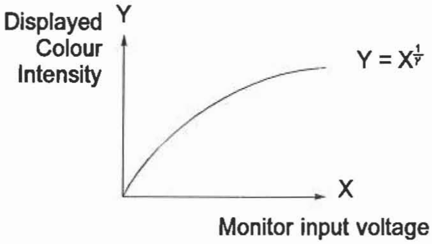
Colour register	Value (hex)
0	00 00 00
1	01 01 01
...	...
254	FE FE FE
255	FF FF FF

Table B.9

When using the pseudo colour modes this would produce a grey scale from black (0) to white (255). In true colour modes this would allow each colour to be varied from low intensity (0) to high intensity (255).

B.4.1 Gamma correction

Due to the nature of the phospors used on colour monitor screens, colour intensity displayed is not a linear function of the applied electron beam energy. The relationship between monitor input voltage and the observed intensity on the screen is approximated by the power curve shown in Figure B.5.



where γ (gamma) is a constant of the particular phosphor used in the monitor, typically approximated to 2.3

Figure B.5 Gamma correction power curve

To create a gamma corrected colour palette each register value is calculated as follows:

$$\text{Corrected colour value} = 255 \times \left(\frac{\text{colour value}}{255}\right)^{\frac{1}{\gamma}}$$

Colour value	0	25	50	75	100	125	150	175	200	225	255
Corrected value	0	92	126	150	170	187	202	216	229	241	255
Corrected value (hex)	00	5C	7E	96	AA	BB	CA	D8	E5	F1	ff

Table B.10 Example results for $\gamma = 2.3$

B.4.2 Mask register

The 24 bit mask register masks the pseudo colour pixel inputs to the three LUTs. Setting a bit in the mask register to zero causes the corresponding LUT address bit to be set to zero. The mask register has no effect on the picture when the IMS G332 is operating in true colour modes.

B.5 Checksum registers

There are three 24-bit checksum registers, one for each colour channel. Their purpose is to facilitate testing the device and systems containing it. The checksum is located directly before the DACs and after the colour and cursor palettes.

The checksum registers are reset by the falling edge of **FrameInactive**. They accumulate only those pixels which are visible on the screen, i.e. those pixels which are unblanked. The checksum value is dependent on the cursor position and whether or not interlaced mode is selected, but independent of sync modes and flyback patterns. The registers should be read during the first part of frame flyback. At the end of this period they are reset, and at other times they are accumulating and are consequently invalid.

The checksum registers are addressed from the microport as 24 bit words containing low, middle and high bytes as follows:

Micro port address	Bits 16-23	Bits 8-15	Bits 0-7
#X0C0	Red bits 0-7	Green bits 0-7	Blue bits 0-7
#X0C1	Red bits 8-15	Green bits 8-15	Blue bits 8-15
#X0C2	Red bits 16-23	Green bits 16-23	Blue bits 16-23

Table B.11 Checksum registers

The checksum is a shift register that uses feedback to create a pseudo-random sequence. As each pixel arrives, its 8-bit value is XOR-ed with the lowest 8 bits of the register, and the register is shifted. The feedback minimises the risk of two errors in the pixel stream cancelling to give a correct checksum value.

B.5.1 Programming the screen description registers

The screen description registers are memory mapped as shown in Table B.12. They contain the line timing parameters, which are programmed in screen units of four pixels duration (also referred to as one 'SClk') and also the frame timing parameters, which are programmed in units of half a linetime.

These registers can only be written to while the Video Timing Generator is turned off (this is controlled by writing to Control Register A (Address #X060)).

A summary of the screen description parameters is given in Table B.12.

Line timing parameters during blanked and displayed lines

The constituent periods of a display line map directly to timing generator registers with two exceptions. Firstly, the line synchronising pulse is split into two periods of equal duration which are used in immediate succession – the parameter used for this is 'halfsync'. Secondly, there is no register for frontporch, rather the total line time is programmed into a separate register and the end of the scan line occurs when Linetime expires.

All line timing parameters are specified in 'Screen Units' of four pixels, so that for example a 1024 pixel screen width is described as 256 SU. The line description parameters are illustrated in Figure B.6.

Figure B.6 (a) shows the parameters relating to a full scan line. The duration of frontporch is not directly defined but is determined by Linetime minus the duration of the other scan line components. Halfsync is used twice in succession to construct the line sync pulse.

Figure B.6 (b) shows the parameters relating to a short scan line (occurring only in interlaced displays). The overall duration is determined by Linetime. As for a full scan line the duration of the frontporch is not directly defined, hence it can differ from the implied Frontporch value of a full scan line.

Line timing parameters during frame flyback

During frame flyback, the VTG uses two variants of the basic linescan sequence. In VPreEqualise and VPostEqualise the waveform produced is either constant high or tesselated as shown in Figure B.6 (c). The internal cycle, however, uses the parameter sequence as shown in Figure B.6 (b) so the parameter ShortDisplay must contain a valid value even though interlace is not being used.

In VSync, the waveform produced is either constant low or tesselated as shown in Figure B.6 (d). The internal cycle uses both Broadpulse and Linetime so the parameter Broadpulse must contain a valid value, even if flyback tesselations are not being used.

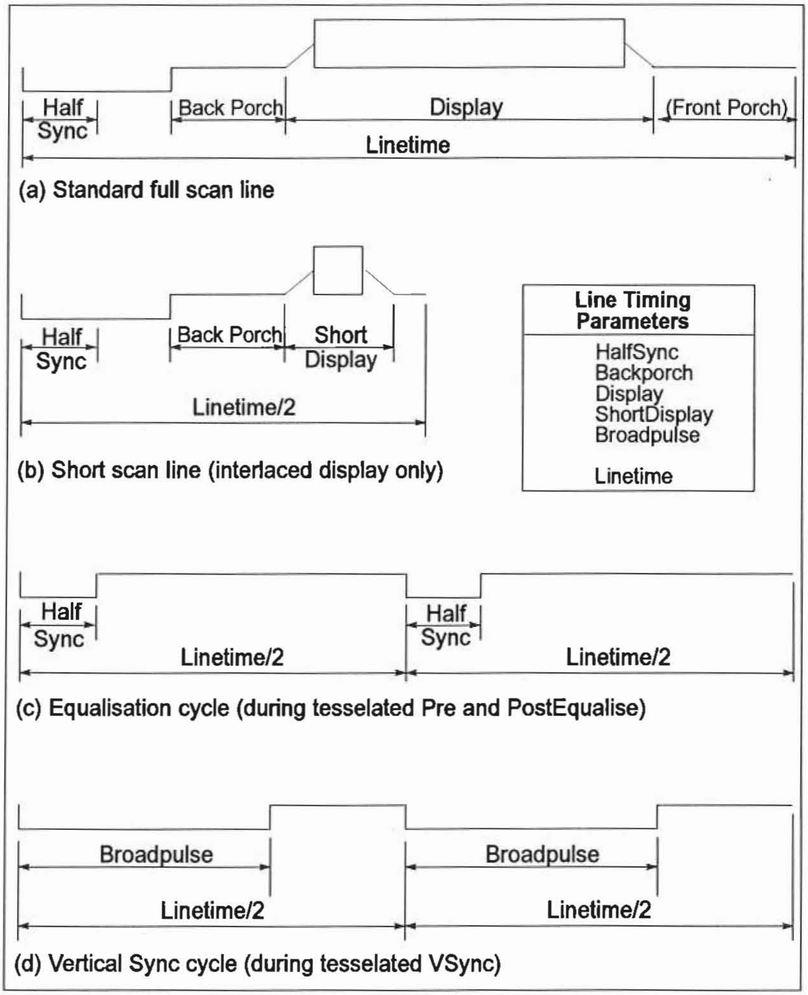


Figure B.6 Line timing parameters

Frame timing parameters

The IMS G332 generates synchronising signal timings and levels conforming to both broadcast and closed circuit television standards. This means that, as well as being capable of generating frame sync patterns associated with non-interlaced computer graphics systems, it is also able to produce tesselated sync signals for an interlaced television system (see Figure B.7).

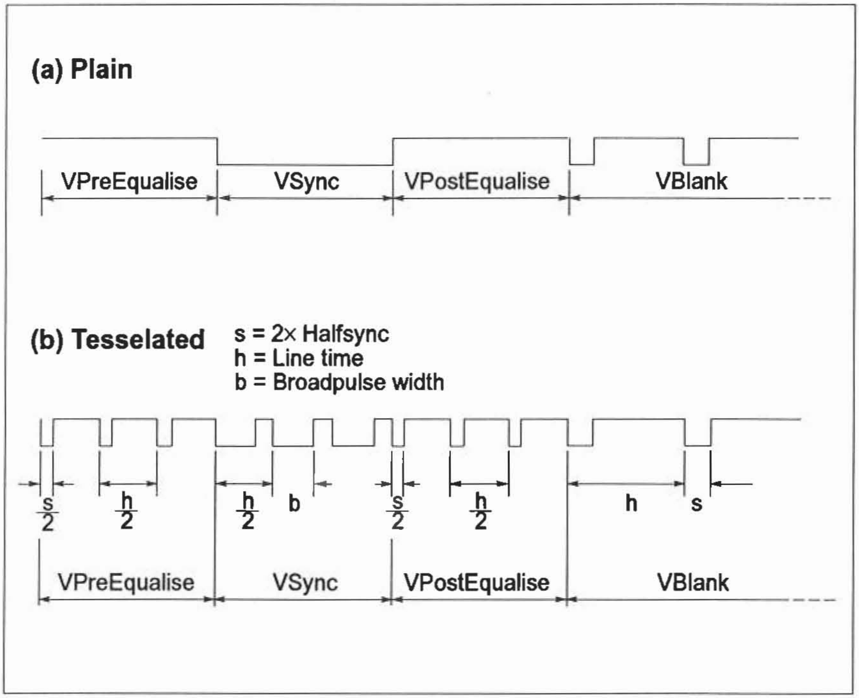


Figure B.7 Composite Sync frame flyback waveforms

In non-interlaced mode the VBlank and VDisplay parameters must be an even number of lines. The remaining parameters may be odd or even, depending on the monitor specification.

In interlaced mode, VDisplay must be odd and Vblank may be even or odd, depending on the standard being followed. The remaining parameters must all be even. Note that in interlaced mode the VDisplay parameter describes the number of lines in each field, not the number in each frame, so that, for a 625-line TV frame, the VDisplay parameter is 625 Linetime periods.

In non-interlace there is only one field, so that a 1024-line frame would have 2048 Linetime periods.

Pa-ram-eter	Notes	Unit	Address
Line timing parameters			
Linetime	Determines the total line time (the sum of the displayed and blanked periods of a single line). For interlaced displays Linetime/2 also determines the duration of short (half) scan lines (see Figure B.6). Linetime > (2xHalfSync + BackPorch + Display) Linetime must be an even multiple of the period of **notSerialClk***	SU	#X02B
HalfSync	Determines the line sync pulse duration (2xHalfSync). HalfSync also determines the low period in pre and post equalise cycles of a tesselated frame flyback waveform.	SU	#X021
Back-porch	Determines the duration of Backporch (Figure B.6). Note that Backporch must exceed TransferDelay by at least one **notSerialClk*** period; also it must exceed 16 SClk periods in total.	SU	#X022
Display	Determines the duration of the visible portion of the scan line.	SU	#X023
Front-porch	The duration of Frontporch (shown in Figure B.6) is implied by the following parameters: Frontporch = Linetime - (Display + Backporch + (2xHalf-Sync))		
Short Display	Determines the duration of the visible portion of short (half) scan lines. ShortDisplay < Linetime/2 - (2xHalfSync + BackPorch) Transfer delay must not exceed ShortDisplay.	SU	#X024
Frame timing parameters			
VDisplay	Describes the vertical display field (visible lines). For a non-interlaced display VDisplay = 2x(Lines per frame). For an interlaced display VDisplay = Lines per frame The total number of displayed lines in each frame must be a whole number. In interlace, this must be an odd whole number.	Half lines	#X02A
VBlank	Determines the number of blanked lines during frame flyback. The total frame blanking period is the sum of VBlank, VSync, VPreEqualise and VPostEqualise.	Half lines	#X029
VSync	Determines the duration of the frame sync pulse and, for a tesselated frame flyback waveform, the number of VSync cycles.	Half lines	#X026
VPre Equalise	Determines the duration of PreEqualise and, for a tesselated frame flyback waveform, the number of PreEqualise cycles.	Half lines	#X027
VPost Equalise	Determines the duration of PostEqualise and, for a tesselated frame flyback waveform, the number of PostEqualise cycles.	Half lines	#X028
Broad-Pulse	Determines the low period in a Vsync cycle of a tesselated frame flyback waveform.	SU	#X025
	* Note: SU=1 **notSerialClk** period=4 pixel periods		

Table B.12 Summary of screen description parameters

Line timing parameter calculation

The line timing parameters can be calculated using the equations given in Table B.13. Note that if the monitor waveforms are specified in microseconds, they must first be converted to screen units by dividing by the period of **notSerialClk**.

During a full line cycle (VBlank, VDisplay)		
Halfsync	=	Horizontal Sync/2
BackPorch	=	BackPorch
Display	=	Display
Linetime	>	(2xHalfSync + BackPorch + Display)
During an equalisation cycle		
ShortDisplay	<	Linetime/2 - (2xHalfSync + BackPorch)
Low period	=	HalfSync
High period	=	Linetime/2 - HalfSync
During a VSync cycle		
BroadPulse	=	Linetime/2 - Pulse width*
Low Period	=	BroadPulse
High period	=	Pulse width
* Note: Pulse width = duration of serration pulse high time		

Table B.13 Screen description line parameter equations

The following restrictions on parameter values **must** be observed:

- All parameters must be greater than 1.
- Linetime must be an even multiple of the period of **notSerialClk**.
- 2xHalfSync + BackPorch + Display > Linetime/2 > 2xHalfSync + BackPorch.
- The total number of displayed lines in each frame must be a whole number. In interlace, this must be an odd whole number.
- Backporch must exceed TransferDelay by at least one **notSerialClk** period; also it must exceed 16 SClk periods in total.
- Transfer delay must not exceed ShortDisplay.

Frame timing parameter calculation (non-interlaced)

If the monitor waveforms are specified in milliseconds, then they must be converted to Linetime units as follows:

$$\text{Total number of lines per frame} = \frac{\text{frame period (ms)}}{\text{line period } (\mu s)}$$

Number of flyback lines = Total number of lines - number of displayed lines

The VTG programming rules are:

Number of flyback lines×2 = VPreEqualise + VSync + VPostEqualise +VBlank

and:

VBlank must be even.

Once VSync has been expressed in terms of Linetimes then specification of the remaining parameters is determined by the monitor requirement for the location of VSync in the blanking period.

B.5.2 The VTG startup sequence

Reading from and writing to the VTG registers, which are memory mapped, is only possible while the timing generator is disabled. Enabling and disabling the VTG is achieved by a write to Control Register A (Address = #X060).

On startup, after reset, the host processor must write a configuration pattern to the IMS G332 bootstrap location. The effect of this is to set the PLL multiplication factor, clock source (PLL or external crystal) and microport address alignment. It must then initialise the VTG by writing a 0 to bit 0 in Control Register A.

Startup sequence:

1 Assert, then deassert **Reset**.
–Wait 50ns

2 Write configuration pattern to bootstrap location.
–Wait 20μs to allow PLL to stabilise

3 Write #0000 to Control Register A

4 Write #0000 to Control Register B

After this the screen parameters, colour table, cursor pattern, etc. can be written to the appropriate locations in any order. The processor must then make another write to Control Register A to enable the VTG which will then start up. The IMS G332 can be reprogrammed without asserting Reset.

The reprogramming sequence has three steps:

1 Write zero to bit 0 of Control Register A, disabling VTG.

2 Write to the screen parameter registers chosen for redefinition.

3 Write one to bit 0 of Control Register A, (redefining modes if necessary by modifying the relevant register bits) and enabling the VTG.

If only the operating mode is to be changed, step 2 only may be omitted, the remainder of the address space is programmed without disabling the VTG. If the clock multiplication factor is to be changed the full startup procedure must be followed, including reset.

Control Register B must not be written to except as part of the startup sequence.

B.6 Register reference guide

B.6.1 The control registers and boot location

Control Register A

See table B.14.

Bit	Function	Comments
0	Enable VTG	0 = VTG disabled 1 = VTG enabled
1	Screen format	0 = non-interlaced 1 = interlaced
2	Interlace standard	0 = EIA format 1 = CCIR format
3	Operating mode	0 = master mode 1 = slave mode
4	Frame flyback pattern	0 = tesselated sync 1 = plain sync
5	Digital sync format	0 = composite sync 1 = separate sync
6	Analogue video format	0 = composite video + sync 1 = video only
7	Blank level	0 = no blank pedestal 1 = blanking pedestal
8	Blank I/O	0 = CBlank is output 1 = CBlank is input
9	Blank function switch	0 = delayed CBlank at pad 1 = Undelayed ClkDisable, at pad
10	Force blanking (irrespective of bit 11)	0 = no action 1 = screen blanked
11	Turn off blanking	0 = blanking enabled 1 = blanking disabled
12-13	VRAM address increment	see definition
14	Turn off transfer cycles	0 = DMA transfer cycles enabled 1 = DMA transfer cycles disabled
15	Pixel pipeline delay	As for bits16-17
16-17	Pixel pipeline delay	Delays sampling point by 0 - 7 VTG clock cycles
18	Pixel port interleaving	0 = non-interleaved 1 = interleaved
19	Delayed sampling	0 = disabled 1 = enabled – see definition (Section B.2.1)
20-22	Bits per pixel	see definition (Section B.2.1)
23	Cursor disable	0 = cursor enabled 1 = cursor disabled

Table B.14 Control Register A bit allocations (Address #X060)

Control Register B

Bit	Function	Comments
0-23	All bits reserved	Write zero on startup only

Table B.15 Control Register B bit allocations (Address #X070)

Boot location

Bit	Function	Comments
0-4	PLL multiplier	Binary coded PLL multiplication factor
5	Clock source select	0 = external (×1) clock 1 = PLL clock
6	Micro port address alignment	0 = 32 bit 1 = 64 bit
7-23	Reserved	Write zero

Table B.16 Boot location bit allocations (Address #X000)

The boot location must be written on power-up and after any reset before attempting to access any other locations from the micro port.

C Network Support Configuration

Various options need to be set on particular networking packages to obtain best performance from the socket connections. The following data may assist in the process of installing and configuring the networking package.

PC-TCP

Installations of this software have been done using the Clarkson packet drivers supplied with the PC-TCP product and a selection of 3COM interface cards.

The CONFIG.SYS file will need to contain entries for the two drivers used by the TCP software

```
device=\pcftp\ipcust.sys
device=\pcftp\ifcust.sys
```

Allow the maximum number of files (20) and a large number of buffers (32).

A typical startup file for PC/TCP will include:

```
\clarkson\3c503 0x60 3 0x350
\pcftp\ethdrv -t 8 -p 24
```

or

```
\clarkson\3c501 0x60
\pcftp\ethdrv -t 8 -p 24
```

The additional options on the TCP resident software allow additional buffering and additional client connections. Performance will suffer if these are not used.

D Using *i*X software with an IMS B419

The *i***X** software can be used on an IMS B419 graphics TRAM mounted on an IMS B008 PC motherboard, provided that a suitable separate display is available. The majority of installation and setup steps are the same, including the parameter selection and base address options. The **ix-memory-size** parameter in the IX.CFG file should be set to 2MBytes or greater, according to the display resolution that is required. Refer to section 8.4.

Install the IMS B008 as normal with an IMS B419 in TRAM slot zero. The IMS B419 should be jumpered such that video memory is contiguous with DRAM. This gives additional space for the X-server code and allows additional client operation. This is selected by using jumper JP4. With a 1024×768 pixel display, an additional 1MB of video memory can be used to supplement the DRAM on the card. The video memory used is placed at the upper extreme of the VRAM address space.

Example IX.CRT files for use with this configuration are provided on the installation disk and are copied to the CFG sub directory of the installation tree.

Sales Offices

EUROPE

DENMARK

2730 HERLEV
Herlev Torv, 4
Tel. (45–42) 94.85.33
Telex:35411
Telefax: (45–42) 948694

FINLAND

LOHJA SF–08150
Karjalankatu, 2
Tel. 12.155.11
Telefax: 12.155.66

FRANCE

94253 GENTILLY Cedex
7, Avenue Gallieni – BP 93
Tel. (33–1) 47.40.75.75
Telex: 632570 STMHQ
Telefax: (33–1) 47.40.79.10

67000 STRASBOURG
20, Place des Halles
Tel. (33) 88.75.50.66
Telex: 870001F
Telefax: (33) 88.22.29.32

GERMANY

6000 FRANKFURT
Gutleutstrasse, 322
Tel. (49–69) 237492
Telex: 176997 689
Telefax: (49–69) 231957
Teletex:6997689=STVBP

8011 GRASBRUNN
Bretonischer Ring, 4
Neukerloh Technopark
Tel. (49–89) 46006–0
Telex: 528211
Telefax: (49–89) 4605454
Teletex:897107=STDISTR

5000 HANNOVER 51
Rotenburgerstrasse, 28A
Tel. (49–511) 615960
Telex: 175118418
Telefax: (49–511) 6151243

8500 NURNBERG 20
Erlenstegenstrasse, 72
Tel. (49–911) 59893–0
Telex: 626243
Telefax: (49–911) 5980701

5200 SIEGBURG
Frankfurter Str. 22a
Tel. (49–2241) 660 84–86
Telex: 889510
Telefax: (49–2241) 67584

7000 STUTTGART
Oberer Kirchhaldenweg, 135
Tel. (49–711) 692041
Telex: 721718
Telefax: (49–711) 691408

ITALY

20090 ASSAGO (MI)
V.le Milanofiori – Strada 4 –
Palazzo A/4/A
Tel. (39–2) 89213.1 (10 lines)
Telex: 330131 – 330141
SGSAGR
Telefax: (39–2) 8250449

40033 CASALECCHIO DI RENO (BO)
Via R. Fucini, 12
Tel. (39–51) 591914
Telex: 512442
Telefax: (39–51) 591305

00161 ROMA
Via A. Torlonia, 15
Tel. (39–6) 8443341
Telex: 620653 SGSATE I
Telefax: (39–6) 8444474

NETHERLANDS

5652 AM EINDHOVEN
Meerenakkerweg, 1
Tel. (31–40) 550015
Telex: 51186
Telefax: (31–40) 528835

SPAIN

08021 BARCELONA
Calle Platon, 6, 4th Floor, 5th Door
Tel. (34–3) 4143300 – 4143361
Telefax: (34–3) 2021461

28027 MADRID
Calle Albacete, 5
Tel. (34–1) 4051615
Telex: 27060 TCCEE
Telefax: (34–1) 4031 134

SWEDEN

S–16421 KISTA
Borgarfjordsgatan, 13 – Box 1094
Tel. (46–8) 7939220
Telex: 12078 THSWS
Telefax: (46–8) 7504950

SWITZERLAND

1218 GRAND–SACONNEX (GENEVA)
CheminFrançois-Lehmann18/A
Tel. (41–22) 7986462
Telex: 415493 STM CH
Telefax: (41–22) 7984869

United Kingdom And Eire

MARLOW, BUCKS SL7 1YL
Planar House, Parkway
Globe Park
Tel. (44–628) 890800
Telex: 847458
Telefax: (44–628) 890391

AMERICAS

BRAZIL

05413 SAO PAULO
R. Henrique Schaumann
286–CJ33
Tel. (55–11) 883–5455
Telex: (391) 11–37988
"UMBR BR"
Telefax: 11–551–128–22367

CANADA

BRAMPTON, ONTARIO
341, Main St. North
Tel. (416) 455–0505
Telefax:416–455–2606

USA

NORTH & SOUTH AMERICAN MARKETING HEADQUARTERS
1000, East Bell Road
Phoenix, AZ 85022
(1)–(602)867–6100

SALES COVERAGE BY STATE

ALABAMA

303, Williams Avenue,
Suite 1031,
Huntsville, AL 35801–5104
Tel.(205)533–5995

ARIZONA

1000, East Bell Road
Phoenix, AZ 85022
Tel. (602) 867–6100

CALIFORNIA

200 East Sandpointe,
Suite 120,
Santa Ana, CA 92707
Tel. (714) 957–6018

2055, Gateway Place,
Suite 300
San José, CA95110
Tel. (408) 452–9122

COLORADO

1898, S. Flatiron Ct.
Boulder, CO 80301
Tel.(303)449–9000

FLORIDA

902 Clint Moore Road
Congress Corporate Plaza II
Bldg. 3 – Suite 220
Boca Raton, FL 33487
Tel.(407)997–7233

GEORGIA

6025, G.Atlantic Blvd.
Norcross, GA 30071
Tel. (404) 242 –7444

ILLINOIS

600, North Meacham
Suite 304,
Schaumburg, ILL 60173–4941
Tel. (708) 517–1890

INDIANA

1716, South Plate St.
Kokomo, IN 46902
Tel. (317) 459–4700

MASSACHUSETTS

55, Old Bedford Road
Lincoln North
Lincoln, MA 01773
Tel. (617) 259–0300

MICHIGAN

17197, N. Laurel Park Drive
Suite 253,
Livonia, MI 48152
Tel. (313) 462–4030

MINNESOTA

7805, Telegraph Road
Suite 112
Bloomington, MN 55438
Tel. (612) 944–0098

NEW JERSEY

Staffordshire Professional Ctr.
1307, White Horse Road Bldg. F.
Voorhees, NJ 08043
Tel. (609) 772–6222

NEW YORK

2–4, Austin Court
Poughkeepsie, NY 12603–3633
Tel. (914) 454–8813

NORTH CAROLINA

4505, Fair Meadow Lane
Suite 220
Raleigh, NC 27607
Tel. (919) 787–6555

TEXAS

1310, Electronics Drive
Carrollton, TX 75006
Tel. (214) 466–7402

ASIA/PACIFIC

AUSTRALIA

NSW 2027 EDGECLIFF
Suite 211, Edgecliff Centre
203–233, New South Head Road
Tel. (61–2) 327.39.22
Telex: 071 126911 TCAUS
Telefax: (61–2) 327.61.76

HONG KONG

WANCHAI
22nd Floor – Hopewell Centre
183, Queen's Road East
Tel. (852–5) 8615788
Telex: 60955 ESGIES HX
Telefax: (852–5) 8656589

INDIA

NEW DELHI 110001
Liaison Office
62, Upper Ground Floor
World Trade Centre
Barakhamba Lane
Tel. 3715191
Telex: 031–66816 STMI IN
Telefax: 3715192

KOREA

SEOUL 121
8th Floor Shinwon Building
823–14, Yuksam-Dong
Kang-Nam-Gu
Tel. (82–2) 553–0399
Telex: SGSKOR K29998
Telefax: (82–2) 552–1051

MALAYSIA

PULAU PINANG 10400
4th Floor, Suite 4–03
Bangunan FOP, 123D Jalan Anson
Tel. (04) 379735
Telefax: (04) 379816

SINGAPORE

SINGAPORE 2056
28 Ang Mo Kio – Industrial Park, 2
Tel. (65) 48214 11
Telex: RS 55201 ESGIES
Telefax: (65) 4820240

TAIWAN

TAIPEI
12th Floor
571, Tun Hua South Road
Tel. (886–2) 755–4111
Telex: 10310 ESGIE TW
Telefax: (886–2) 755–4008)

JAPAN

TOKYO 108
Nisseki Takanawa Bld. 4F
2–18–10 Takanawa
Minato-ku
Tel. (81–3) 3280–4125
Telefax: (81–3) 3280–4131

SOFTWARE PROBLEM REPORT

inmos

This form should be used to report a problem with INMOS software to Software Support, INMOS Business Centre, SGS-THOMSON Microelectronics at one of the following addresses:

Asia/Pacific : 28 Ang Mo Kio Industrial Park 2, Singapore 2056.
France : 7 Avenue Gallieni, BP 93, 94253 Gentilly Cedex.
Germany : Bretonischer Ring 4, 8011 Grasbrunn, München.
Italy : V.le Milanofiori, Strada 4, Palazzo A/4/A, 20090 Assago (MI).
Japan : Nisseki Takanawa Building, 4th Floor, 18-10 Takanawa 2-chome, Minato-ku, Tokyo 108.
Scandanavia : Borgarfjordsgaten, 13 Box 1094, S-16421 Kista, Sweden.
UK : Planar House, Parkway Globe Park, Marlow, Bucks SL7 1YL.
USA : 2225 Executive Circle, PO Box 16000, Colorado Springs, Colorado 80935-6000.

Name of Field Application Engineer:
Date of report:
Originator of report:

(please include name, address
and telephone number)

Has FAE been informed?
Is a reply required?
Originator's reference:

Product name and number, version number and date (Use special form for TDS):

Identity of software component with problem:

Host hardware and OS version:

Master transputer type and memory size:

Tick one of the following items:

Documentation error or inadequacy □
Serious software bug (system crashes) □
Other software bug □
Suggestion for design change □

Description of problem (one problem only) *continue on separate sheet if necessary*

(If there is something you cannot do that you expected to be able to do please describe this)