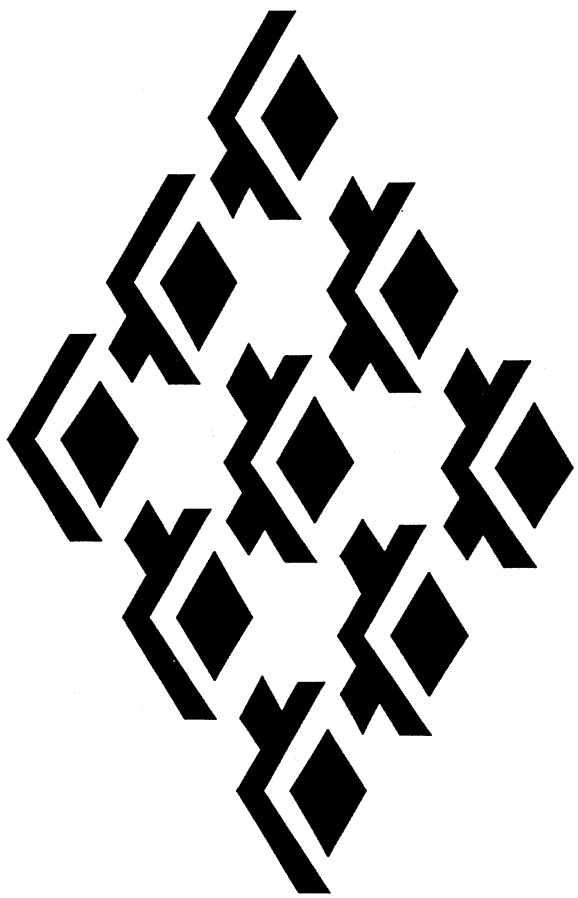
TRANSTECH TTG3

HIGH RESOLUTION GRAPHICS TRAM
USER MANUAL

VERSION 1.0 4:9:1989



TRANSTECH DEVICES LIMITED
UNIT 17, WYE INDUSTRIAL ESTATE
LONDON ROAD
HIGH WYCOMBE
BUCKINGHAMSHIRE
HP11 1LH
ENGLAND

TELEPHONE (+44) 0494 464303
FAX (+44) 0494 463686

Contents

Chapter 1

Introduction

The TTG3 is a revolutionary new high resolution graphics subsystem compressed onto an extremely small format TRAM (size 4). The TTG3 provides all the performance and functionality of comparable systems, but in a much smaller package.

The use of the latest surface mount technology, and the Inmos G300 Colour Video Controller (CVC), has kept the component count to a minimum. This level of integration results in increased performance and better reliability.

The TTG3 uses the IMS T800 as a general purpose graphics processor. A total of 2 Mbytes of DRAM is provided for program and data. The four bi-directional serial links on the transputer are ideally suited for downloading data from a distributed image store into the frame store. The micro-port of the G300 is also directly addressable by the T800 enabling user control of all G300 parameters.

The frame store consists of 2 Mbytes of interleaved Video RAM. The Video Timing Generation performed by the G300 is totally user programmable, which ensures the TTG3 can support a wide range of display resolutions, and monitors. The G300 can be configured to use external synchronising signals, which may be from another TTG3, thus supporting multiple synchronised framestores.

The pixel port of the G300 has two modes of operation. In mode 1 (8 bits/pixel) consecutive 32 bit words are clocked through the pixel port, and each 8 bit pixel defines an offset in the 256 element Colour Lookup Table. Each table entry gives 24 bits of colour, which is used to drive three 8 bit DACs. Mode 1 therefore provides 256 displayable colours from a total palette of 16 million. In mode 2 (24 bits/pixel) the pixel data is supplied at full video rate. The bottom 3 bytes of every word are sent direct to the DACs for display, giving a total of 16 million colours displayable.

The TTG3 is in the most part compatible with the Inmos G300 TRAM as described in [1].

Chapter 2

Installation

2.1 Package Contents

TTG3
User Guide
Two 360K (MSDOS) Disks
Video cable

2.2 Installing the TTG3

The TRAM conforms to the Inmos TRAM format as defined by Inmos Technical Note [3]. As such the TRAM may be installed on any module mother board designed for Inmos format TRAMs, e.g. Transtech MCP1000-3, IMS B008, IMS B014, etc.

The TTG3 is a size 4 TRAM, but only the 16 pins on site 1 are actually used (see block diagram Appendix B). These pins provide the following functions:

Pin	Function
1	Link2Out
2	Link2In
3	VCC
4	Link1Out
5	Link1In
6	LinkSpeedA
7	LinkSpeedB
8	ClockIn(5MHz)
9	Link3In
10	Link3Out
11	GND
12	Link0In
13	Link0Out
14	notError
15	Reset
16	Analyse

LinkSpeedA is connected to Link0special on the processor, and **LinkSpeedB** is connected to Link123Special.

Connect the three SMB connectors on the supplied cable to the RGB SMB connectors on the TTG3, and the other end to your monitor, see Appendix B Block Diagram for position of R,G, and B.

You are now ready to install the set-up software, if required.

2.3 Installing Software

It is recommended that a copy is made of both the distribution disks before proceeding any further.

Insert distribution disk 1 into drive A: and run the install batch file. It will ask for the target directory to be defined. This disk contains the TTG Server and should typically be copied to `\TTG3\TTGSERVER` .

The server directory includes a standalone server program and all the occam source. For further details refer to the server guide TTGD001.

Insert distribution disk 2 into drive A: and run the install batch file. It will ask for the target directory to be defined. This disk contains s simple hardware test program (**htest.btl**) and a VTG set-up program (**vtgsetup.btl**). These should normally be installed in the directory `\TTG3` . Both programs require a two processor configuration as defined in the next section.

2.4 VTG Set-up

For those familiar with such things turn straight to Section 3.5 for an explanation of the VTG parameters. For those less familiar and or more anxious to see something on the screen proceed as below.

To determine the parameters necessary to best drive your monitor a simple set-up program is provided. This is a standalone program designed to run from the Inmos afserver. It is configured for a two processor system, in which the root processor is connected to the TTG3 via a single link. Link2 of the root processor must be connected to Link1 of the TTG3. If this configuration is not achievable, refer to section 4.1, which describes the program in more detail.

Apply the necessary power and start the set-up program.

```
afserver -:b vtgsetup.btl [options]

valid options   /f <frame_rate>
                /l <line_freq>
                /p <pixel_clock>
```

```
/x <x_size>
/y <y_size>
```

The optional arguments can be used to define the intial values for the five defineable parameters.

The following line is displayed on the Host monitor

```
i)ncrement f)rame l)ine p)ixelclock x)size y)size
```

To drive the program, select any of the options from the menu (with a single key press). Whenever p,f,x or y are selected the parameter is incremented/decremented by the current increment value. To change this select i)ncrement and increase/decrease the increment.

For a full description of the set-up program see section 4.1. The four parameters which need to be defined to initiate display, include **frame rate**,**line rate**, **pixel clock**, **xsize**, and **ysize**. These parameters are then used by the program to calculate the final VTG parameters etc.. The basic function of these parameters can be summarised as follows:

line defines the line frequency which will typically be between 40 and 64KHz. If you have a multisync monitor then the line frequency can be set to any value within the sync. range of the monitor, otherwise this must be set to the exact sync. value of the monitor.

frame defines the frame frequency which will typically be between 40 and 70Hz. If you have a multisync monitor then the frame rate can be set to any value within the sync. range of the monitor, otherwise this must be set to the exact sync. value of the monitor.

pixelclock defines the rate at which pixels are output to the screen, some value between 30 and 120MHz, depending on the screen resolution, and monitor.

xsize defines the size of the screen in x.

ysize defines the size of the screen in y.

If the screen clears to grey then proceed as follows, else there is probably some problem with the RGB leads, or monitor.

1] Adjust **xsize** and **ysize** to the desired values.

2] Select the required **pixelclock** frequency, if in doubt select 80MHz.

3] Adjust **line rate** until the picture stabilises horizontally (Around 60 Khz).

4] Adjust **frame rate** until the picture stabilises vertically (Around 60 Hz).

The current parameter settings are displayed on the status line so make a note of these once you are happy with the display. These parameters will be required for any future software development.

For a full listing of the relevant VTG parameters turn to section 3.5.2.

Chapter 3

Hardware Description

3.1 Overview

The TTG3 supports both 25 and 30 MHz (when available) T800's. The module is configured with 2 Mbytes 80 ns DRAM (upgradeable to 8 Mbytes with 4 Mbit ram), the cycle time of varies between 3 and 6 cycles depending on the access type. The DRAM is subdivided into two interleaved 1 (4) Mbyte banks. This enables fast contiguous writes/reads.

A full 2 Mbytes of VRAM is provided, which can support a wide range of display resolutions. The VRAM also supports a very high speed block clearing option, using the VRAM serial port.

The Module can be used with a wide range of display monitors, and is fully programmable through the use of the G300. The Module can generate 256 levels of each of the primary colours, thus enabling very subtle colouring, and grey scaling.

The Module supports both 8 bit pixels and full 32 bit pixels. In the former case, the Colour Look-up table within the G300 is used to convert an 8 bit address to a 24 bit colour, which is then fed direct to the DAC's. In the 32 bit mode, the bottom 24 bits of every pixel (32 bit word) are fed direct to the DAC's.

3.2 Display Resolution

The display resolution is limited only by the amount of VRAM, the maximum dot rate of the G300, and the access time of the serial port on the VRAM. The TTG3 supports displays of upto 1280 * 1024 pixels.
Possible screen sizes include:

1280 * 1024	8 bits/pixel
1024 * 1024	8 bits/pixel double buffered
768 * 768	8 bits/pixel treble buffered
640 * 480	32 bits/pixel
512 * 512	32 bits/pixel double buffered

Table 3.1: Sample Resolutions

The display resolution is defined by the line timing parameters, and frame timing parameters as defined below 3.5.2, and also by the G300 mode and Video Data Rate determined by the frequency of **PllClkin**. Unfortunately all these parameters are closely related, and setting the required resolution involves initialising all these parameters to the relevant value. This is discussed in detail in section 3.5, but the proceedure is involved, and therefore it is recommended that the **initCRTC** library routine is used (see ??). This uses only the four basic parameters

Mode	JP3	Video Clock Source	Pixel Route
1	OFF	output of on-chip PLL	through LUT
1	ON	**pixClkIn**	through LUT
2	OFF	not available	-
2	ON	**pixClkIn**	direct to DACs

Table 3.2: Clock and pixel port options

derived from the VTG set-up program vtgset.tg3, frame rate, pixel clock frequency, xsize and ysize.

However it is worth mentioning at this stage the two modes of operation of the G300. As intimated earlier, the G300 actually has two quite different modes, which define the operation of the pixel port.

The pixel port takes in pixel data from the video RAM and outputs it either via the look-up table in Mode 1, or direct to the DACs in Mode 2. The mode is defined by a single bit in the G300 control register (see 3.4.4) The clock source to the G300 is controlled by a jumper JP3 on the TTG3 board. The clock input options and resultant modes are illustrated the two modes is illustrated in table 3.2

The jumper JP3 is ON when jumper in place, OFF when jumper removed.

On the TTG3 the external Pixel Clock input **pixClkIn** is driven directly from the Processor Clock **ProcClockOut**. This means that in Mode 2 operation the input clock **pixClkIn** will always run at 25MHz. This can only be changed by removing a resistor (R51) and attaching a different clock to this pin.

3.3 Memory Map

The memory on the board is subdivided into two non-contiguous areas. Figure 1 shows how the Video and Dynamic RAM is mapped within the address space of the T800, and the control addresses for the Subsystem and G300.

The G300 has a memory mapped architecture, which enables fast access from the graphics processor for operations such as dynamic colour palette changes, and VTG changes. All configurable parameters are controlled by a block of 512 32 bit read/write registers, called the micro-port. The micro-port appears within the memory map of the TTG3 T800 at address #40000000.

The memory map is illustrated in Figure 3.1

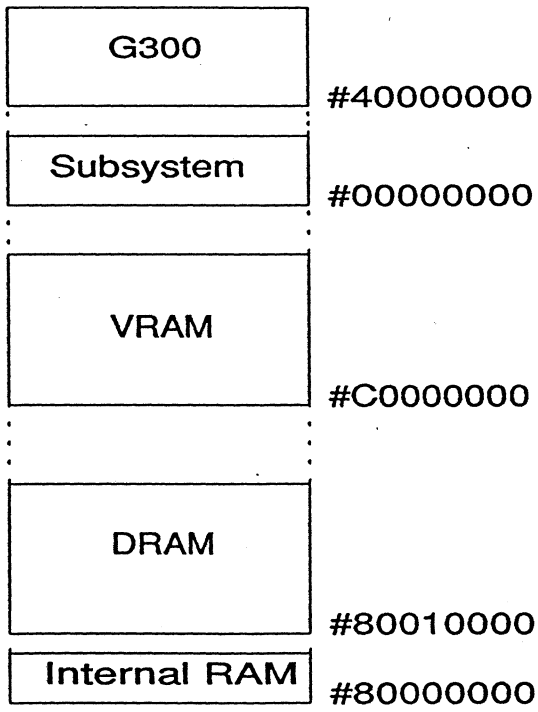


Figure 3.1: TTG3 Memory Map

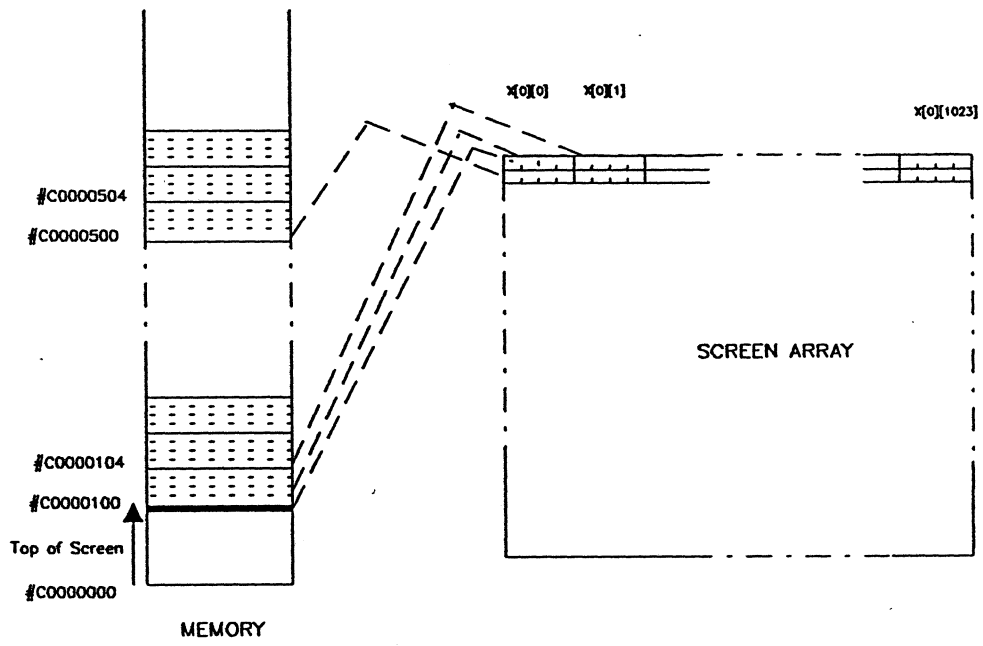


Figure 3.2: TTG3 Screen Addressing

3.3.1 Screen Addressing

The offset of the base of screen is programmable using the **Top of Screen** register (see 3.4.3), and is an offset from #C0000000.

The screen can be addressed as either a two dimensional byte array, in Mode 1, or a two dimensional word array, in Mode 2. In both cases the top left of screen X[0],Y[0] is at the lowest physical address. This is illustrated in Figure 3.2. The whole frame store is contiguous, so consecutive scan lines are storred consecutively.

NB In this example the Top Of Screen register (3.4.3) has been set to #100 to offset the screen array by 256 bytes from the start of VRAM. Normally the screen array will be located at #C0000000 which is the start of VRAM.

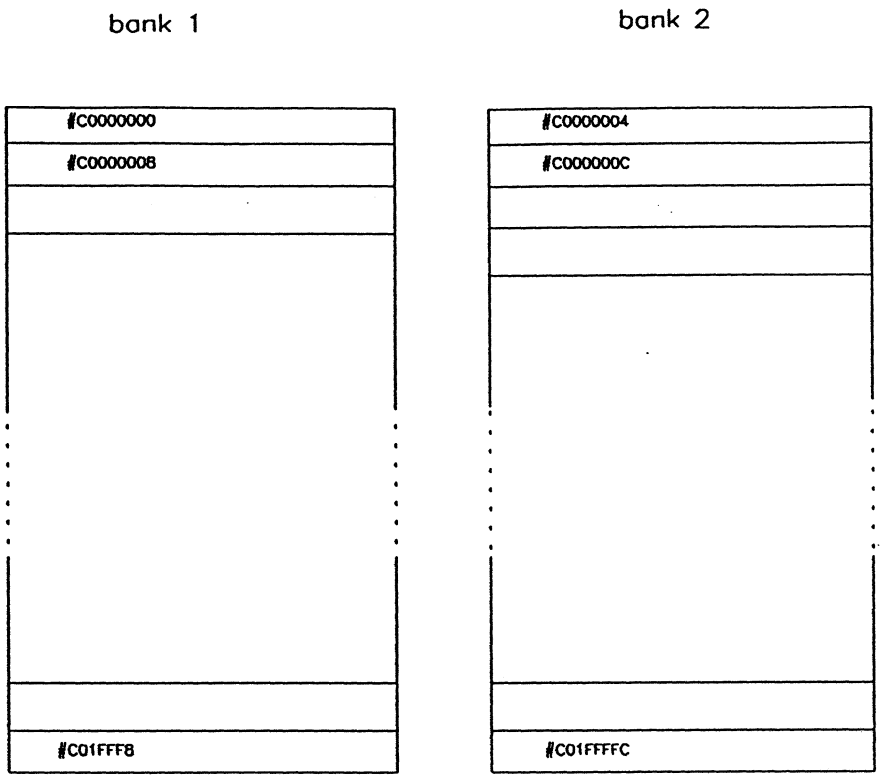


Figure 3.3: Memory Interleave

3.3.2 Interleaved Memory

Both the VRAM and DRAM on the TTG3 are organised as two interleaved banks of memory. In both cases the banks are word interleaved. This means that consecutive word read/writes access different memory banks. While one bank is being addressed, the last bank is being precharged, so that next consecutive access completes in 3 cycles as opposed to 5 cycles for a non-interleaved access. This buys very significant performance improvements for incremental array accesses. The interleaved banks are illustrated in Figure 3.3.

Register	Address	Comments
Boot Location	#400001A0	Startup location to which must be written the clock multiplication factor in PLL mode. Writing to this location sets the byte counter to zero.
Top of Screen	#40000180	Read/Write register giving the ability to re-program the top of screen.
Control Register	#40000160	Read/Write register. Contains all configuration information. Used to start and stop timing generation.
Mask Register	#40000140	Read/Write mask register. masks each pixel address byte.
DataPath Register	#40000121 to #4000012C	Read/Write registers containing the screen description parameters. Accessible only when the timing generator is not running.
Byte Counter	#40000100	Incrementing counter used in byte access mode. This may be set by writing the current byte number to it. On startup, it is set to zero by the bootstrap routine.
Colour Palette	#40000000	256 (32 bit) locations of 24 bit colours read/rite accessible at all times.

Table 3.3: Register Functions

3.4 G300

This section covers the basic features of the IMS G300 applicable to the TTG3, the reader is referred to the Inmos Graphics Databook (3), for further details. The basic issues of frame store addressing, colour palette manipulation, and Video Timing Generation are covered elsewhere, and it is intended that the support software provided should in the most part hide the complexity of the G300, and provide a flexible interface to the device. However there are features of the G300 which are not supported by these library routines, and which demand direct addressing of micro-port.

3.4.1 Addressing the micro-port

The micro-port is a 24 bit interface which can be configured to operate in byte wide, or 24 bit wide mode (word mode). The mode is set by the *micro port mode* bit in the control register as defined in table 3.5. It should be noted that the G300 defaults to word mode on reset and so all the registers should be addressed as word registers from reset.

The G300 register functions are outlined in Figure 3.3.

These are the only addresses in the range (512 words) which may be written to. Writing to other locations may have unpredictable results.

Video Data Rate	Multiplication
30	6
70	14
100	20
120	24

Table 3.4: Video Data Rate Selction

3.4.2 The Bootstrap Location

On startup, after reset a configuration pattern must be written to the bootstrap location. The effect of this is to set the Phase Locked Loop multiplication factor. The multiplication value used defines the video data rate (pixel clock speed).

Speed selection

As mentioned above the video data rate (pixel clock speed), is defined in Mode 1 by a multiplication of the input clock **PllClkIn**.

Once written, the multiplication factor is set until the G300 is reset, at which point the boot location can be re-written with another multiplication factor if required. Only **ADBus[4:0]** are valid as data during these write cycles, valid factors are 5–32. The G300 on the TTG3 shares the processors' 5 MHz clock, and as such the relevant multiplication factor must be applied to generate the required video data rate. Table 3.4 gives the relevant multiplication factors for a range of video rates.

NBB multiplication factors less than 5 are invalid.

3.4.3 Top of Screen

The top of screen register defines the first address in each frame. The first 22 bits of the top of screen register define the frame store start offset within the VRAM array.

```
31          21                          0
0000000000nnnnnnnnnnnnnnnnnnnnnn
```

This register can be addressed at any time (other than during a row transfer), and can be used for efficient panning and scrolling. It is recommended that top of screen is only updated during flame flyback, as signalled by flyback event.

3.4.4 Control Register

The bit pattern written to this 32 bit register determines the operating mode of the G300. The function of each of the bits is given in Table 3.5.

Bit	Function	Comments
23-16	Reserved	Not valid on read, write zero
15	Turn off blanking	1 = blanking disabled 0 = blanking enabled
14	Turn off DMA	1 = No VRAM management 0 = DMA VRAM operational
13	Reserved	Write zero
12	Black level	Selects blanking level 0 = blank = black level
11-9	Delay value	Delays internal Sync and Blank by 0-7 clock cycles
8	Pixel port mode	0 = model, 1 = mode2
7	Micro port mode	0 = word mode, 1 = Byte mode
6	Reserved	Write zero
5	Frame flyback pattern	1 = plain synchronising waveform 0 = tesselated synchronising waveform
4	Digital sync format	0 = mixed sync, 1 = separate sync
3	Analogue video format	1 = video only 0 = video and sync composite
2	Device operating mode	0 = master mode, 1 = slave mode
1	Screen format	0 = non-interlaced, 1 = interlaced
0	Enable VTG	0 = VTG disabled, 1 = VTG running

Table 3.5: Control Register bit allocations

3.4.5 Mask Register

The mask register is only applicable to mode 1. It is used to mask all incoming pixel addresses to the CLUT. the contents of this register are logically ANDed with the incoming pixel stream. This register can be used to achieve rapid colour changes, by masking sections of the CLUT, and even completely blanking the display.

3.4.6 DataPath Registers

There are 12 single byte datapath registers as follows:

For a more detailed description of these registers, the reader is referred to the Inmos "Graphics Databook" [2].

3.4.7 Byte Counter

These registers can be used for test purposes when the VTG is not running.

Address	Function
#40000121	HalfSync
#40000122	BackPorch
#40000123	Display
#40000124	ShortDisplay
#40000125	BroadPulse
#40000126	VSync
#40000127	VBlank
#40000128	VDisplay
#40000129	Linetime
#4000012A	LineStart
#4000012B	MemInit
#4000012C	TransferDelay

Table 3.6: DataPath Registers

Address	Function
#40000012D	HIncrementer
#40000012E	VIncrementer
#40000012F	TBIncrementer

Table 3.7: Byte counter registers

3.4.8 Colour Palette

The 256 CLUT locations are accessed via the micro-port to enable block moves of whole tables into the CLUT. The CLUT can thus be addressed like any other memory location. Only the bottom 24 bits of each location are significant.

3.5 The Video Timing Generator

The IMS G300 includes a fully programmable Video Timing Generator ("VTG"). This section covers the more detailed features of this VTG where relevant to the TTG3. For a more detailed description of the VTG the reader is referred to [2]. The VTG provides composite sync and blanking to the on-chip video DACs, it controls the timing of **BusReq** and **Transfer** and it starts and stops **notShiftClk** to control the flow of pixels onto the screen. It also provides a **FrameInactive** signal which is asserted whenever the display enters frame flyback.

Although the VTG can be configured to generate both interlaced, and non-interlaced synchronising waveforms, the TTG3 does not fully support interlaced displays. The TTG3 has an advanced interleaved memory structure which enables fast contiguous read/writes, and unfortunately this presents problems in the generation of interlaced output.

3.5.1 The display screen

A raster scan screen can be subdivided into a number of raster lines. The majority of these lines are visible because the display remains un-blanked, the remainder are blanked frame flyback

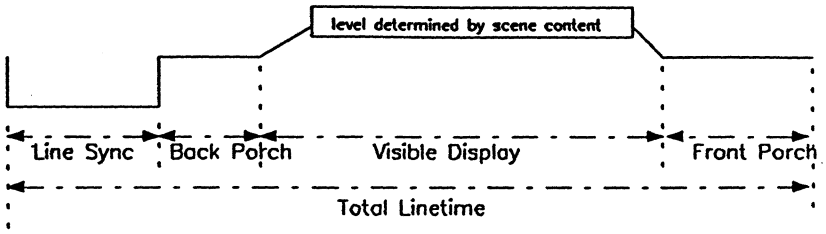


Figure 3.4: raster line segments

lines. Each of the displayed lines has a display period, and a blanked flyback period, made up of front, back porch and line sync. The total line time is the sum of the displayed and blanked periods. This raster line structure is illustrated in Figure 3.4.

3.5.2 Line timing parameters

There are a total of six line description parameters:

HalfSync

Backporch

Display

ShortDisplay

Broadpulse

Linetime

These line timing periods are used to program the VTG and correspond to the Datapath Registers. All line timing parameters are specified in units of four pixels, so a line with a total

linetime of 1024 pixels is defined as 256 line units.

The line segments shown in Figure 3.4 are used directly to program the VTG, with two exceptions. The Line Sync pulse is split into two states of equal duration, called **Halfsync**. Also there is no programmable delay for **FrontPorch**, instead the total line time (**Linetime**) is used to determine frontporch.

3.5.3 Frame timing parameters

All frame timing parameters are specified in terms of half line times. Thus a non-interlaced display of 1024 lines has the value 2048 written to the **VDisplay** register.

There are a total of three frame description parameters:

VSync,

VBlank,

VDisplay.

The **VSync** parameter defines the period of the vertical synchronisation pulse for the display.

The **VBlank** parameter defines the period of vertical blanking, and is the total period for which vertical blanking is asserted.

The **VDisplay** register defines the number of displayed lines.

3.5.4 Calculation of VTG parameters

The following equations can be used to derive the VTG parameters:

During a full line cycle (VBlank, VDisplay)

```
HalfSync     =   Horizontal Sync/2
BackPorch    =   BackPorch
Display      =   Display
FrontPorch   =   Linetime - (2*HalfSync + BackPorch+Display)
```

During an Equalisation Cycle

```
Low period   =   HalfSync
ShortDisplay =   LineTime/2 - (2*HalfSync+BackPorch+FrontPorch)
High period  =   HalfSync + BackPorch+ShortDisplay+FrontPorch
FrontPorch   =   LineTime/2 - (2*HalfSync+BackPorch+ShortDisplay)
```

During a VSync cycle

```
Low period    =    BroadPulse
High period   =    FrontPorch
FrontPorch    =    Linetime/2 - BroadPulse
```

The following restrictions on parameter values must be observed:

1] all parameters must be non-zero,

2] Linetime must be an even multiple of the period of **notSerialClk**,

3] the Halfline point must fall within the active display period with at least one **notSerialClk** period of display either side of it,

4] the total number of displayed lines in each frame must be a whole number ,

5] the vertical blanking period must be a whole number of lines,

6] **Backporch** must exceed **TransferDelay** by at least one **notSerialClk** period,

7] **TransferDelay** must not exceed **ShortDisplay**.

A set of sample parameters for the Hitachi HM-4219/4119 are given in Appendix A.

3.6 The Startup Sequence

Once the G300 has been reset, the user must ensure that the relevant multiplication factor is written to the G300 bootstrap location. This also sets the micro-port into a usable state. Following this the user must set the micro-port mode by writing the relevant bits in the control register.

The whole startup sequence can be summarised as follows:

```
1 Assert, then deassert Reset.
2 Write configuration pattern to bootstrap location
  (e.g.#00000010 for 80MHz operation).
3 Write to control register to set micro-port to desired state.
4 Initialise Screen parameters.
5 Set VTG enable bit in control register to start VTG.
6 Initialise CLUT.
```

In order to simplify this procedure a simple library routine is provided with the TTG3 to initialise the VTG etc. This routine performs all the functions described as startup sequence 2 to 5 inclusive. The library routine **initCRTC** is described in more detail in Section 4.2.1.

3.7 Event Handling

The FrameInactive signal of the G300 has been used to generate event requests. This enables the use of frame flyback events for synchronising double buffered animated sequences.

Register	Address
SubSystemReset (write only)	#00000000
SubSystemAnalyse (write only)	#00000004
SubSystemError (read only)	#00000000

Table 3.8: Subsystem Registers

A single event is generated at the beginning of frame flyback and it is up to the user to ensure that this event is serviced before the active display period is entered. For a 1024 * 1024 screen at 60 Hz the user has approximately 580 Micro seconds to handle the event.

It should be noted that the T800 event request logic will only be cleared on an event acknowledge. It is advisable to handle frame events with a high priority process to ensure that the event is serviced within the non-display window. Care must be taken if other high priority processes may be scheduled at this time.

3.8 Subsystem Control

The TTG3 is provided with a complete set of subsystem controls. These can be used to control a network of transputers. Three signals are provided: **SubSystemReset**, **SubSystemAnalyse**, and **SubSystemError**.

To maintain software compatibility between TRAMs the subsystem registers start at hardware address #00000000 , and are mapped as follows:

Setting bit 0 of the relevant register asserts reset/analyse. A '1' read form bit '0' of SubSystemError indicates ERROR.

A further register is provided which enable the user to reset the G300 without resetting the processor.

writing a '1' into bit 0 of #000000F0 asserts G300 reset.
writing a '0' into bit 0 of #000000F0 deasserts G300 reset.

3.9 Advanced Features

3.9.1 Interleaved Memory

The basic design of the interleaved memory system has been described in section 3.3, but it is worth stressing at this point how significant carefull management of array allocation and access can be. Unpre-charged accesses are 40% faster than precharged accesses, and this can be achieved on all regular consecutive array accesses, provided that the arrays are word aligned.

3.9.2 Fast Screen Clear

The TTG3 supports an extremely efficient screen fill operation. This is achieved using the VRAM serial port. A library proceedure **fast.clear()** is provided within the TRANSTECH graphics library, which uses the transputer block move instruction and the serial port, to clear a 1 Mbyte block of VRAM to a selected colour in under 400 us.

3.9.3 Memory Upgrades

The TTG3 has been designed to support the new generation of 4 Mbit DRAMs, so that the module is upgradable to 8 Mbytes. This is ideal for running advanced high level windowing systems or graphics libraries such as X Windows, Display Postscript interpretters, Doré, etc.

Chapter 4

Software Description

4.1 VTG tuning program

The TTG3 is supplied with a simple setup program, designed to ease the task of deriving the optimum VTG parameters for a particular monitor. The source for this program (written in occam) is provided in TDS (D700D) format (**toplevel.top**), and also as a standalone program (**vtgsetup.btl**) which can be run from the Inmos **afserver**.

The standalone version expects a two processor configuration, with the root processor Link2 connected to the TTG3 Link0. If this is not suitable then a new version must be generated from the source provided.

There are five parameters which must be adjusted interactively within **vtgsetup**, and these are defined below.

increment select increment to change the current increment/decrement value. The program maintains a single delta value called **increment** which may be applied to any of the five definable parameters, to adjust their current value accordingly. So to increment the frame rate simply type **f** when the incrementor is set to 1. To change the incrementor, the i)incrementor option should be selected. Typing **l** will multiply the incrementor by 10, typing **s** will divide the incrementor by 10, and **n** will negate the incrementor.

line defines the line rate which will typically be between 40 and 64KHz. If you have a multisync monitor then the frame rate can be set to any value within the sync. range of the monitor, otherwise this must be set to the exact sync. value of the monitor.

frame defines the frame rate which will typically be between 40 and 70Hz. If you have a multisync monitor then the frame rate can be set to any value within the sync. range of the monitor, otherwise this must be set to the exact sync. value of the monitor.

pixelclock defines the rate at which pixels are output to the screen, some value between 30 and 120MHz, depending on the screen resolution, and monitor.

xsize defines the size of the screen in x.

ysize defines the size of the screen in y.

The initial state of any of the five parameters may be defined by arguments passed to the program e.g.

```
afserver -:b vtgsetup.btl [options]

valid options   /f <frame_rate>
                /l <line_freq>
                /p <pixel_clock>
                /x <x_size>
                /y <y_size>
```

4.2 TTG3 Graphics Server

The TTG3 comes complete with a low level Graphics Server, which supports basic primitives such as draw.circle, draw.line, fill.polygon, write.text etc. This Server provides a common interface to the whole range of TRANSTECH graphics devices.

The TRANSTECH Server **TTGS** is documented in [4].

Server functions of particular importance to the TTG3 are discussed below.

4.2.1 initCRTC

One of the library routines provided in this Server is the initCRTC routine. This is a general purpose Cathode Ray Controller setup routine and the interface is compatible with both the medium resolution TTG1 and the high resolution TTG3.

A full listing of the procedure is given in Appendix C (in occam). The proceedure interface is defined below:

```
initCRTC(lineFreq,frameRate,pixelClock,scrxSize,scrySize,interlace)
```

The parameters lineFreq,frameRate,pixelClock,scrXsize, and scrYsize, are defined above for the VTG set-up routine vtgsetup(). The values extracted from this set-up procedure should be plugged direct into initCRTC to setup the required display.

4.2.2 fastclear

The TTG3 offers a fast block clear option, which can be used to clear the screen to a defined background colour.

This routine uses the serial port of the VRAM to perform ultra fast block clear. The basic operation is to clear 4 Kbytes of VRAM (this corresponds to 8 RAMs worth of serial shift buffers). This 4 Kbytes is then copied into the serial shift buffer, which is then replicated through the rest of VRAM. Timing of this routine is highly critical and the function is therefore provided as a library routine, which should not be changed.

The routine is called with a **bank** which takes value 0, or 1. The bank number defines the 1K * 1K frame buffer to be cleared. The **colour** defines the target colour.

```
fast.clear(VAL INT bank, VAL BYTE colour)
```

Appendix A

Sample VTG parameters

This appendix covers the basic calculation of all the G300 VTG register values. This proceedure will only be necessary when you wish to set up the G300 directly, and do not use the initCRTC routine provided. It therefore assumes a certain level of knowledge about Video Timing Generation etc. and is not for the faint hearted.

Screen parameters for the Hitachi HM-4219/4119 are as follows:

```
HalfSync        21
BackPorch       54
Display         320
LineTime        422
ShortDisplay    109
BroadPulse      205
VSync           6
VBlank          56
VDisplay        2048
```

The remaining three parameters are concerned with management of the Video RAM bitmap. The TTG3 uses 256K * 4bit VRAMs, therefore the shift register length is 256 nibbles. The sum of the parameters **MemInit** and **TransferDelay** must not exceed this figure.

Thus:

```
Meminit + TransferDelay = 256
```

Transfer delay is

```
System DMA latency + VRAM access time + 1 Screen unit
```

Assume DMA latency to be around 500ns and the VRAM access time is known to be 80ns then:

```
TransferDelay = 580ns / 37.184ns + 1 = 17.60
              = 18 Screen units
```

Which obeys the conditions for TransferDelay that:

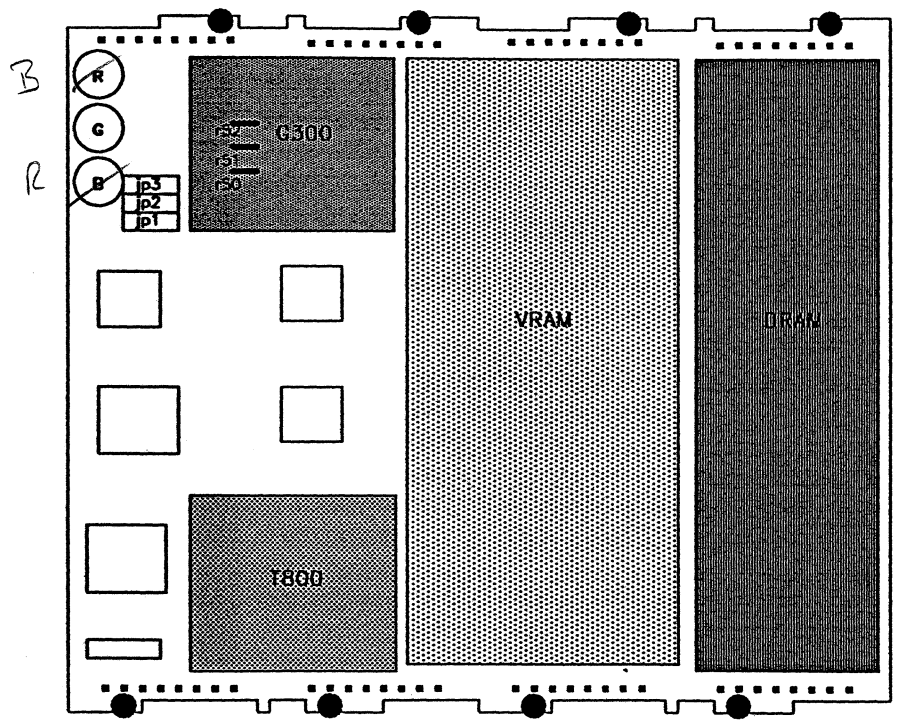
```
TransferDelay < BackPorch
TransferDelay < ShortDisplay
```

This gives

```
MemInit = 256 - 18 = 238 Screen units.
```

Appendix B

Block Diagram



R51,R52,R53 are actually on the reverse side of the pcb under the G300

Figure B.1: TTG3 Block Diagram

Appendix C

initCRTC

Listing of initCRTC routine.

```
PROC initCRTC(VAL INT lineFreq,frameRate,pixelClock,scrxSize,scrySize,
                      VAL BOOL interlace)
  #USE snglmath
  VAL PLLClockIn IS 5000000: --5Mhz
  [#1BF]PORT OF INT registers:
  VAL g300.register.base IS #30000000 (INT):
  PLACE registers AT g300.register.base:
  VAL INT boot.location    IS #1A0:
  VAL INT top.of.screen    IS #180:
  VAL INT control.register IS #160:
  VAL INT mask.register    IS #140:
  VAL INT half.sync        IS #121:
  VAL INT back.porch       IS #122:
  VAL INT display          IS #123:
  VAL INT short.display    IS #124:
  VAL INT broad.pulse      IS #125:
  VAL INT v.sync           IS #126:
  VAL INT v.blank          IS #127:
  VAL INT v.display        IS #128:
  VAL INT line.time        IS #129:
  VAL INT line.start       IS #12A:
  VAL INT mem.init         IS #12B:
  VAL INT transfer.delay   IS #12C:
  VAL INT h.incrementer    IS #12D:
  VAL INT v.incrementer    IS #12E:
  VAL INT tb.incrementer   IS #12F:
  VAL INT byte.counter     IS #100:
  VAL INT colour.palette   IS #000:
  VAL REAL32 HorizPosition  IS 0.20 (REAL32):  -- percent backporch
  VAL INT Display           IS scrxSize/4:
  VAL INT vertSyncWidth     IS 6: -- lines
```

```
VAL INT TransferDelay     IS 21:
PROC resetG300 ()
  [#FF]INT reset:
  VAL G3Reset IS #F0:
  PLACE reset AT #20000000:
  SEQ
    VAL timeout IS 20:   --1.28 ms
    INT timenow:
    TIMER clock:
    SEQ
      clock ? timenow
      clock ? AFTER timenow PLUS timeout
    reset[G3Reset] := 1
    VAL timeout IS 20:   --1.28 ms
    INT timenow:
    TIMER clock:
    SEQ
      clock ? timenow
      clock ? AFTER timenow PLUS timeout
    reset[G3Reset] := 0
:

INT PLLfactor:
REAL32 lineTime :
REAL32 pixTime  :
REAL32 shiftTime :
REAL32 SyncWidth.real:
REAL32 lineTime:
INT half.sync.data:
INT back.porch.data:
INT display.data:
INT short.display.data:
INT broad.pulse.data:
INT v.sync.data:
INT v.blank.data:
INT v.display.data:
INT line.time.data:
INT line.start.data:
INT mem.init.data:
INT transfer.delay.data:
INT mask.register.data :
INT top.of.screen.data :
INT TotalHorizClocks:
INT totalLines:
INT SyncWidth:
INT BlankWidth:
INT BackPorch:
INT FrontPorch:
INT vertSync:
INT vertBlank:
SEQ
```

```
SEQ
  PLLfactor := pixelClock / PLLClockIn
  pixTime   := 1.0 (REAL32) /(REAL32 ROUND pixelClock)
  shiftTime := pixTime * 4.0 (REAL32)
  lineTime  := 1.0 (REAL32) / (REAL32 ROUND lineFreq)
SEQ
  TotalHorizClocks   := INT ROUND ((lineTime / shiftTime)+1.5(REAL32))
  TotalHorizClocks   := TotalHorizClocks /\ (~1)  --Ensure even mult.
  IF
    TotalHorizClocks  <  ((scrxSize/4) + (scrxSize/(4*20)))
      TotalHorizClocks  := ((scrxSize/4) + (scrxSize/(4*20))) --bottom out
    TRUE
      SKIP
  SyncWidth.real     :=  (REAL32 ROUND TotalHorizClocks) * 0.1(REAL32)
  SyncWidth    := (INT ROUND (SyncWidth.real + 0.5(REAL32))) --round this up
  SyncWidth    := SyncWidth /\ (~1) --even No.
  BlankWidth   := (TotalHorizClocks - (Display + SyncWidth))
  FrontPorch   := INT ROUND ((REAL32 ROUND BlankWidth) * HorizPosition)
  BackPorch    := BlankWidth - FrontPorch
  IF
    BackPorch <= TransferDelay
      SEQ
        BackPorch  := TransferDelay + 1
        FrontPorch := TotalHorizClocks - (SyncWidth + (BackPorch + Display))
    TRUE
      SKIP
SEQ
  totalLines  := lineFreq/frameRate
  IF
    totalLines < (scrySize + (3 + (scrySize/20))) --must have some v blanking
      totalLines := scrySize + (3 + (scrySize/20))
    TRUE
      SKIP
  vertBlank   := totalLines - scrySize
half.sync.data     :=  SyncWidth /2
transfer.delay.data:=  TransferDelay
back.porch.data    :=  BackPorch
display.data       :=  Display
v.sync.data        :=  vertSyncWidth
v.blank.data       :=  vertBlank
v.display.data     :=  scrySize * 2
line.time.data     :=  TotalHorizClocks
short.display.data :=  (line.time.data/2)-((SyncWidth + BackPorch) + FrontPorch)
broad.pulse.data   :=  164  --not used
line.start.data    :=  0
mem.init.data      :=  512 - transfer.delay.data
mask.register.data :=  255
top.of.screen.data :=  0
VAL VTGsetup IS [[ half.sync,       half.sync.data     ],
                 [ back.porch,      back.porch.data    ],
                 [ display,         display.data       ],
```

```
                    [ short.display,   short.display.data ],
                    [ broad.pulse,     broad.pulse.data   ],
                    [ v.sync,          v.sync.data        ],
                    [ v.blank,         v.blank.data       ],
                    [ v.display,       v.display.data     ],
                    [ line.time,       line.time.data     ],
                    [ line.start,      line.start.data    ],
                    [ mem.init,        mem.init.data      ],
                    [ transfer.delay,  transfer.delay.data],
                    [ mask.register,   mask.register.data ],
                    [ top.of.screen,   top.of.screen.data ]] :

  SEQ
    resetG300 ()
    registers [ boot.location] ! PLLfactor
    SEQ i = 0 FOR SIZE VTGsetup
      VAL register IS VTGsetup [i][0] :
      VAL contents IS VTGsetup [i][1] :
      registers [register] ! contents
    registers [control.register] ! 0
    registers [control.register] ! 1
:
```

Bibliography

[1] **"The Design of a high resolution graphics system using the IMS G300 Colour Video Controller"** *Inmos Technical Note 62.*

[2] **"THE GRAPHICS DATABOOK"** *INMOS Limited, Inmos Databook Series.*

[3] Inmos Technical Note 25.

[4] **TRANSTECH GRAPHICS SERVER (TTGS)** *Transtech Devices document ref. TTGD001.*