3Dlabs®

GLINT *Gamma1*

& Gamma 2

Errata

PROPRIETARY and
CONFIDENTIAL INFORMATION

Issue 4

Proprietary Notice

Email: info@3dlabs.com
Web: http://www.3dlabs.com

3D*labs* Ltd.
Meadlake Place
Thorpe Lea Road, Egham
Surrey, TW20 8HE
United Kingdom
Tel: +44 (0) 1784 470555
Fax: +44 (0) 1784 470699

3D*labs* K.K
Shiroyama JT Mori Bldg 16F
40301 Toranomon
Minato-ku, Tokyo, 105, Japan
Tel: +81-3-5403-4653
Fax: +91-3-5403-4646

3D*labs* Inc.
480 Potrero Avenue
Sunnyvale, CA 94086,
United States
Tel: (408) 530-4700
Fax: (408) 530-4701

Change History

Document	Issue	Date	Change
149.5.3	1	14 May 98	First Issue.
149.5.3	2	18 Jan 99	Added feedback bug.
149.5.3	3	8 April 99	Added output FIFO Disconnect bug.
149.5.3	4	15 July 99	Added Spurious PCI Transfer Concatenation bug.

Introduction

The following information is for the GLINT Gamma device (subsequently just called Gamma). There is an errata for each known problem containing a detailed description and suggested workarounds.

Identification

Gamma may be identified by two means. The first is the physical markings on the part itself, the second is by reading the Vendor ID, Device ID and Revision ID Registers in PCI Configuration Region. Please refer to the Gamma Hardware Reference Manual for further details.

Part Marking	Vendor ID	Device ID	Revision ID
Gamma	3D3Dh	0008h	0001h

Software Drivers and Reference Designs

Please note that 3Dlabs supplied software drivers and reference designs include the appropriate bug fixes and workarounds as described in this document.

Errata Summary

Errata No. GAMEN001

Problem

The operation of the Input FIFO Disconnect from the GLINT 500TX or GLINT MX into the pins GLINTInDis[1:0] of Gamma is such that the graphics pipeline arbiter inside Gamma sometimes fails to operate correctly. The bug occurs when the disconnect lines are asserted and then removed while the secondary PCI bus is still idle or if they are asserted and removed during a secondary bus burst operation. Both of these instances can occur because of the pipelining of data into the input FIFO.

Hardware Workaround

The GLINTInDis[1:0] signals need to by conditioned by external hardware. The disconnect lines are ignored just after a PCI burst completes to ensure the Gamma arbitration circuitry works correctly. If a disconnect is signaled during a PCI burst operation the disconnect is held until the burst completes.

The following Abel format listing shows the workaround currently applied to 3Dlabs reference boards.

```
"-------------------------------------------------------------------------
module _gamma;
title '
'
"      Project:       GLINT Gamma
"      Creation       5th Jan 98 (MEW)
"
"      Last Edit      4th Mar 98 (PS)
"      Revision:      B1
"-------------------------------------------------------------------------
       C,H,L,x,z = .C.,1,0,.X.,.Z.;
"-------------------------------------------------------------------------

       S0                    pin 1;         "State Machine Register
       S1                    pin 2;         "State Machine Register
       S2                    pin 3;         "State Machine Register
       S3                    pin 4;         "State Machine Register

       SCI_CLK               pin 5;         "Secondary PCI Clk
       !FRAME0               pin 15;        "From Gamma
       !IRDY0                pin 16;        "From Gamma
       GLINTINDIS0           pin 10;        "to Gamma
       GLINTINDIS1           pin 11;        "to Gamma

       !RESET                pin 9;         "from Gamma

       G_InDis0              pin 23;        "From Glint 0
```

```
        G_InDis1              pin 38;        "From Glint 1

        GLINTINDIS0           istype         'buffer,reg';  "to Gamma
        GLINTINDIS1           istype         'buffer,reg';  "to Gamma
        S3, S2, S1, S0        istype         'buffer,reg';

        VCC0, VCC1            pin 6,30;
        GND0, GND1            pin 18,42;

"-------------------------------------------------------------------------
" State machine state values


        S               = [S3,S2,S1,S0];

        S_Idle          = [ L, L, L, L];

        S_1             = [ L, L, L, H];
        S_2             = [ L, L, H, L];
        S_3             = [ L, L, H, H];
        S_4             = [ L, H, L, L];
        S_5             = [ L, H, L, H];
        S_DisLock0      = [ L, H, H, L];
        S_DisLock1      = [ L, H, H, H];

"-------------------------------------------------------------------------
equations "State Registers

        [S].clk = SCI_CLK;

        GLINTINDIS0.clk = SCI_CLK;
        GLINTINDIS1.clk = SCI_CLK;

"-------------------------------------------------------------------------
" Disconnect bug fix

GLINTINDIS0 := !RESET & (((S == S_Idle) & !FRAME0 & IRDY0)
                      # (S == S_1)
                      # (S == S_2)
                      # (S == S_3)
                      # (S == S_4)
                      # (S == S_DisLock0)
                    # G_InDis0) ;

GLINTINDIS1 := !RESET & (((S == S_Idle) & !FRAME0 & IRDY0)
                      # (S == S_1)
                      # (S == S_2)
                      # (S == S_3)
```

```
                    # (S == S_4)
                    # (S == S_DisLock1)
                  # G_InDis1) ;

state_diagram S "Disconnect fix
        state S_Idle:           if RESET # (!FRAME0 & !IRDY0)
                                    then S_Idle
                                else if (!FRAME0 & IRDY0)
                                    then S_1
                                else if (G_InDis0 & (FRAME0 # IRDY0))
                                            then S_DisLock0
                                else if (G_InDis1 & (FRAME0 # IRDY0))
                                            then S_DisLock1
                                else S_Idle;

        state S_1:              if RESET
                                    then S_Idle
                                else S_2;

        state S_2:              if RESET
                                    then S_Idle
                                else S_3;

        state S_3:              if RESET
                                    then S_Idle
                                else S_4;

        state S_4:              if RESET
                                    then S_Idle
                                else S_Idle;

         state S_DisLock0:      if RESET
                                    then S_Idle
                                      else if (!FRAME0 & !IRDY0)
                                    then S_1
                                else S_DisLock0;

         state S_DisLock1:      if RESET
                                    then S_Idle
                                      else if (!FRAME0 & !IRDY0)
                                    then S_1
                                else S_DisLock1;
end
"-------------------------------------------------------------------------
```

Errata No. GAMEN002

Alert

When the Gamma is set up to use logical addressing for accessing of DMA data, it is important that the Input FIFO PCI disconnect functionality of the input FIFO is not used if there is a chance that a logical page may not be resident. The reason for this is that the test for residence will generate an interrupt which can not be serviced if the input FIFO is disconnecting and therefore causing the PCI bus to keep retrying.

Software Workaround

Do not use Input FIFO Disconnect when using logical addressing.

Errata No. GAMEN003

Problem

When using PCI DMA, if the InDMA Data Throttle bit (2) is set, control registers in the AGP DMA controller will be set incorrectly. If an AGP DMA is then set off it will fail to work.

Software Workaround

Do not set the InDMA Data Throttle bit (2) when using PCI DMA. This bit has no effect when using PCI DMA.

Errata No. GAMEN004

Problem

If the last operation in a DMA buffer is DMACount or DMACall the DMA operation will fail to start correctly.

Software Workaround

The software should ensure that DMACount or DMACall are not put at the end of DMA buffers. If necessary a 'padding' command pair should be added to the end of the DMA buffer.

Errata No. GAMEN005

Problem

In the Rectangle Write DMA function, when the PackIn bit is True, and the data is non-word aligned at the start of a line, the byteOffest is adjusted incorrectly after the first word of data is written.

Software Workaround

The simplest way to workaround this problem is to ensure that any DMA buffers are allocated on 32-bit boundaries.

If it is not possible to align the destination address (which we will call *destAddr*) then there is another workaround. Suppose we are trying to DMA a rectangle at *{xl, xr, yt, yb}* (left, right, top bottom) from a 16 bpp framebuffer. If *destAddr* is only 16-bit aligned, then we must:

- render a single pixel stripe at *{xl, xl+1, yt, yb}* and perform a rectangle write DMA to *destAddr*.

- render the rest of the rectangle *{xl+1, xr, yt, yb}* and perform a rectangle write DMA to *(destAddr+2)*, which is now a 32-bit aligned address,

A similar algorithm should apply for an 8 bpp framebuffer.

Errata No. GAMEN006

Alert

When using the logical addressing mode for Input DMA care must be taken with the positioning of a DMAReturn. The DMA controller requests data ahead of its current data position. By the time that a DMAReturn is recognized a number of extra words may have been requested which could take you into a new page if the DMAReturn is near the end of a page. If the next logical page isn't valid a DMA error will be generated.

Software Workaround

When using logical addressing, either ensure that a DMAReturn is not placed in the last 128 words of a logical page, or Make sure that the next logical page is a valid page.

Errata No. GAMEN007

Problem

When using the logical addressing mode for Input DMA and if a DMACount or DMACall is placed in the last 128 words of a logical page.

If the DMA controller is servicing a page fault, and a DMACount or DMACall is found by the command unit, the command having worked its way down the input FIFO. The current DMA page is erroneously updated with the new DMA address rather than the address for the page, which was being loaded. This means that the page address has become out of step with the logical address sent by the command unit, but the DMA unit will not give a page fault.

Software Workaround

When using logical addressing ensure that a DMACount or DMACall is not placed in the last 128 words of a logical page.

Errata No. GAMEN008

Problem

When an OpenGL rasterposition is defined, Gamma calculates an associated fog value based on the current fogging state. This fog value is then used for subsequent GeomRect operations (i.e. glDrawPixels and glCopyPixels).

Software Workaround

In these circumstances the software driver should calculate the correct fog values and load them into the GLINT 500TX or GLINT MX.

Errata No. GAMEN009

Problem

However large the MultiGLINT aperture is set to by the Gamma configuration bits Gamma will only respond to addresses up to an offset of 16MBytes.

Hardware / Software Workaround

None

Errata No. GAMEN010

Problem

The Gamma MultiGLINT aperture will not generate an address on the secondary bus with an offset greater than 4MBytes. As each alternate scanline is sent to alternate GLINT chips this maps to a total limit of 8MBytes. Note that the limit is on the secondary side address and so different resolutions will fail at different addresses on the primary side.

To calculate the maximum number of lines available for a particular resolution use the following equation:

NumberOfLines = 0x400000 / post-multiplier / multiply * 2

Where the post-multiplier value and the multiply value come from the MultiGLINT aperture control register.

So for example for 1024x768x32

NumberOfLines = 0x400000 / 256 / 16 * 2 = 2048

Hardware / Software Workaround

None

Errata No. GAMEN011

Problem

When using logical addressing, if an input DMA and an output DMA page faults at the same time, it is possible for the page address for the input DMA page address register to update erroneously with the output DMA page address

Software Workaround

To ensure that the bug doesn't occur: When using logical addressing Input DMA and Output DMA must not be running at the same time. The case where this may happen is when using a feedback output DMA. In this case when using logical addressing, the Data must be read from the output FIFO by the host and the output DMA controller must not be used.

Errata No. GAMEN012

Problem

If you re writing to the Gamma Input FIFO through the GP register space, and using Input FIFO disconnect and you are using Input FIFO disconnect on the PCI bus, it is possible for data written to the FIFO to be thrown away.

Software Workaround

Either always write to the 4Kbyte GP FIFO access area, or do not use Input FIFO disconnect when writing to the GP register space.

Errata No. GAMEN013

Problem

When using the Gamma input FIFO DMA controller in legacy mode (controlled via Command Mode register) you can not issue DMA Commands if the input FIFO has anything in it.

Software Workaround

There are 2 possible workarounds for this bug.

1. Use the Queued DMA mode for the command unit instead. This is the recommended method.
2. If you need to use legacy DMA mode for the command unit, then you should ensure that the input FIFO is empty before loading a new DMACount. On previous products it was sufficient to wait for the DMACount register to become zero.

Errata No. GAMEN014

Problem

When using a PCI master on the Gamma secondary PCI bus there is a possibility of system lock-ups. These lock-ups can occur both for DMA reads and DMA writes.

Hardware Workaround

There are no hardware fixes for this problem

Software Workaround

In general the only way to ensure that lock-ups cannot occur is to not make use of a PCI master on the secondary PCI bus.

Errata No. GAMEN015

Problem

If a polygon (triangle, quad or polygon) is rendered in feedback mode, then the POLYGON_TOKEN in the feedback buffer is followed by the number of vertices (always 3). The correct behavior occurs when the input polygon is not clipped. If the input polygon is clipped then the resulting polygon is decomposed into triangles and each triangle is written into the feedback buffer with the POLYGON_TOKEN and vertex count.

Hardware Workaround

There are no hardware fixes for this problem

Software Workaround

The driver software must check if the word following the POLYGON_TOKEN is not 3.0 and insert 3.0 into the application's buffer. Note that biased coordinates areused so it is impossible for the following vertex data to have the value of 3.0

Errata No. GAMEN016

Problem

The operation of the Output FIFO Disconnect from PERMEDIA 3 and GLINT R3, into the pins GLINTInDis[1:0] of GLINT Gamma is such that the secondary PCI bus state machine inside Gamma sometimes fails to operate correctly. The problem occurs when the disconnect lines are asserted during a PCI access or just after a PCI access is completed and before another starts.

Hardware Workaround

The GLINTOutDis[1:0] signals need to be conditioned by external hardware. The disconnect lines are asserted just after a PCI burst completes for 5 clocks to ensure the Gamma arbitration circuitry works correctly. If a disconnect is signaled during a PCI burst operation the disconnect is not passed through to the Gamma until the cycle is completed (FrameN de-asserted).

Errata No. GAMEN017

Problem

Writes to the Rasterizer chip memory through the Gamma bypass may be written to the wrong memory location under some conditions.

The following access sequences can go wrong if they occur close enough in time that the Gamma attempts to combine the accesses into the same burst. If the second access is the start of a burst, then the whole burst may be misdirected.

Address	0xNNNNN3FC		
followed by	0xNNNNN000	is written to:	0xNNNNN400
Address	0xNNNNN7FC		
followed by	0xNNNNN400	is written to:	0xNNNNN800
Address	0xNNNNNBFC		
followed by	0xNNNNN800	is written to:	0xNNNNNC00

where NNNNN is identical for both addresses.

Hardware Workaround

None

Software Workaround

None, although speeding up the secondary bus will incidentally prevent bursts from developing. Given the unusual nature of the problem this is unlikely to be necessary.