ARM Graphics

Mali200 with MaliGP2 Core (GX525)

Errata Notice

This document contains all errata known at the date of issue in releases from revision r0p4 up to and including revision r0p5 of Mali200 with MaliGP2 Core

Proprietary notice

Document confidentiality status

This document is Non Confidential.

Web address

http://www.arm.com/

Feedback on the product

If you have any comments or suggestions about this product, contact your supplier giving:

- The product name
- A concise explanation of your comments.

Feedback on this document

If you have any comments on about this document, please send email to mailto:errata@arm.com giving:

- The document title
- The documents number
- The page number(s) to which your comments refer
- A concise explanation of your comments

General suggestion for additions and improvements are also welcome.

Contents

There are no Errata in this Category **69**

Introduction

Scope

This document describes errata categorised by level of severity. Each description includes:

- the current status of the defect
- where the implementation deviates from the specification and the conditions under which erroneous behavior occurs
- the implications of the erratum with respect to typical applications
- the application and limitations of a 'work-around' where possible

Categorisation of Errata

Errata recorded in this document are split into three levels of severity:

Category 1 Behavior that is impossible to work around and that severely restricts the use of the product in all, or the majority of applications, rendering the device unusable.

Category 2 Behavior that contravenes the specified behavior and that might limit or severely impair the intended use of specified features, but does not render the product unusable in all or the majority of applications.

Category 3 Behavior that was not the originally intended behavior but should not cause any problems in applications.

Change Control

25 jun 2009: Changes in Document v16

Page	Status	ID	Cat	Summary
17	Updated	653167	Cat 1	The MMU_AXI module misses the RLAST signal in some conditions
25	New	716163	Cat 2	The Geometry Processor does not wait for BRESP
27	New	719242	Cat 2	Varying interpolation get reduced precision for large triangles
47	Updated	548681	Cat 3	Impossible to start bus after it has been stopped by performance counter event.
31	Updated	476265	Cat 3	Reading of tile writeback big endian bit fails
65	New	719239	Cat 3	Rasterization fails for large triangles
64	New	719238	Cat 3	Writeback-targets with different downsampling and bounding box does not work in some cases
63	New	719237	Cat 3	Incorrect performance counter 28 and 46 for the pixel processor
62	New	719236	Cat 3	Points and lines following a zero area triangle will be discarded
61	New	719233	Cat 3	Tile write-back might use too short bursts
60	New	718652	Cat 3	WBx_TARGET_PIXEL_FORMAT 4 and 5 do not work when writing back the alpha component

24 jun 2009: Changes in Document v15

Removed all errata that were only relevant to versions prior to r0p4

09 jan 2009: Changes in Document v14

Page	Status	ID	Cat	Summary
17	New	653167	Cat 1	The MMU_AXI module misses the RLAST signal in some conditions
24	New	650468	Cat 2	Shader-program driven texture prefetch can cause hang
59	New	655369	Cat 3	The point size data type is too wide
58	New	654867	Cat 3	Exponent wrap around for scaled FP16 additions
57	New	654667	Cat 3	The early-z performance counter value is too low

14 nov 2008: Changes in Document v13

Only changes to errata prior to revision r0p4

31 okt 2008: Changes in Document v12

Page	Status	ID	Cat	Summary
16	New	618780	Cat 1	ActiveQuads register reset by internally generated combinatorial reset
15	New	564465	Cat 1	ActiveQuads register reset by internally generated combinatorial reset
37	Updated	499914	Cat 3	Inaccurate rasterization
44	Updated	547565	Cat 3	OpenGL ES conformant dithering is only supported for formats with 8-bit per channel
41	Updated	524463	Cat 3	Verbatim32 textures give unexpected results
53	New	618017	Cat 3	Aliased points with non integer size are not OpenGL ES conformant

52	New	618016	Cat 3	nRESET used as input to flip-flops
55	New	618674	Cat 3	Random mipmaplevel and centroid mapping when some fragments in the pixel quad has ended execution
54	New	618666	Cat 3	Too much data might be written to the framebuffer when using the WBx_DIRTY_BIT feature
56	New	618716	Cat 3	Performance counter "Instruction failed rendezvous count" counts incorrectly

27 jun 2008: Changes in Document v11

Page	Status	ID	Cat	Summary
45	Updated	547566	Cat 3	Paletted formats cannot be used together with non-paletted formats
44	Updated	547565	Cat 3	OpenGL-ES conformant dithering is only supported for formats with 8-bit per channel
48	Updated	549015	Cat 3	Rendezvous falsely terminates pixels
46	Updated	547567	Cat 3	Palette cache livelocks.
51	New	559920	Cat 3	Discontinuous line strips or line loops
67	New	559615	Doc	Wrong reset values in the TRM for viewport and screensize registers in the PLBU unit

19 jun 2008: Changes in Document v10

Formatting changes

19 jun 2008: Changes in Document v9

Formatting changes

19 jun 2008: Changes in Document v8

Page	Status	ID	Cat	Summary
23	New	555370	Cat 2	Vertex storer cache can loose data and write data to the wrong address

13 jun 2008: Document v7

Page	Status	ID	Cat	Summary
14	New	549166	Cat 1	AXI masters don't properly handle write transactions correctly in some cases
22	New	551322	Cat 2	Blending doesn't work with FP16 tile-buffers
50	New	551265	Cat 3	Dirty bit write-back doesn't work for Z/stencil buffer
49	New	551122	Cat 3	Load stencil buffer from fragment shader doesn't work.
48	New	549015	Cat 3	Rendezvous falsely terminates pixels
47	New	548681	Cat 3	Impossible to start bus after it has been stopped by performance counter event.
46	New	547567	Cat 3	Pallette cache livelocks.
45	New	547566	Cat 3	Paletted formats cannot be used together with non-paletted formats
44	New	547565	Cat 3	GLES Conformant dithering not supported for other formats than RGBA8888
41	Updated	524463	Cat 3	Verbatim32 textures give unexpected results
39	Updated	502122	Cat 3	Writeback bounding box does not work with interleaved output

23 apr 2008: Changes in Document v6

Page	Status	ID	Cat	Summary
20	New	509016	Cat 2	MaliGP2 doesn't support normalized inputs
21	New	526763	Cat 2	Occasional Early-z errors
34	New	499664	Cat 3	Occasional wrong fragment depth
35	New	499713	Cat 3	Writeback bounding box combined with downsampling may hang
36	New	499714	Cat 3	The Polygon List Builder Initial command list address register contains wrong value.
37	New	499914	Cat 3	Inaccurate rasterization
38	New	502119	Cat 3	Dithering is lost when downsampling
39	New	502122	Cat 3	Writeback bounding box does not work with interleaved output
40	New	509019	Cat 3	Tile buffer writeback may use unaligned AXI burst for some settings
41	New	524463	Cat 3	Verbatim32 textures give unexpected results
42	New	524466	Cat 3	Numeric instability in Z-calculation
43	New	526764	Cat 3	PLBU heap address not completely updated when written to memory

17 des 2007: Changes in Document v5

Page	Status	ID	Cat	Summary
29	New	475813	Cat 3	When out of memory, the list builder may repeatedly write to a single address
30	New	475814	Cat 3	MaliGP2 does not enforce alignment in all cases
31	New	476265	Cat 3	Reading of tile writeback big endian bit fails
32	New	481018	Cat 3	Mali200 may hang with some tile writeback configurations
33	New	481021	Cat 3	Texture data may be read for textures of type NO_TEXTURE

23 okt 2007: Changes in Document v4

Page	Status	ID	Cat	Summary
19	New	460049	Cat 2	Depth clipping can be done wrong when combined with polygon offset or shaders that write to Z

27 sep 2007: Changes in Document v1-v3

Only changes to errata prior to revision r0p4

Errata Summary Table

The errata associated with this product affect product versions as below.

A cell shown thus **X** indicates that the defect affects the revision shown at the top of that column.

ID	Cat	Summary of Erratum	r0p4	r0p4-rel1	r0p4-rel2	r0p5	r0p5-rel1	r0p5-02rel0
559615	Doc	Wrong reset values in the TRM for viewport and screensize registers in the PLBU unit	X	X	X	X	X	X
549166	Cat 1	AXI masters don't properly handle write transactions correctly in some cases	X					
564465	Cat 1	ActiveQuads register reset by internally generated combinatorial reset				X		
618780	Cat 1	ActiveQuads register reset by internally generated combinatorial reset	X	X				
653167	Cat 1	The MMU_AXI module misses the RLAST signal in some conditions	X	X	X	X	X	
460049	Cat 2	Depth clipping can be done wrong when combined with polygon offset or shaders that write to Z	X	X	X	X	X	X
509016	Cat 2	MaliGP2 doesn't support normalized inputs	X	X	X			
526763	Cat 2	Occasional Early-z errors	X	X	X			
551322	Cat 2	Blending doesn't work with FP16 tile-buffers	X	X	X			
555370	Cat 2	Vertex storer cache can loose data and write data to the wrong address	X	X	X			
650468	Cat 2	Shader-program driven texture prefetch can cause hang	X	X	X	X	X	X
716163	Cat 2	The Geometry Processor does not wait for BRESP	X	X	X	X	X	X
719242	Cat 2	Varying interpolation get reduced precision for large triangles	X	X	X	X	X	X
475813	Cat 3	When out of memory, the list builder may repeatedly write to a single address	X	X	X			
475814	Cat 3	MaliGP2 does not enforce alignment in all cases	X	X	X			
476265	Cat 3	Reading of tile writeback big endian bit fails	X	X	X			
481018	Cat 3	Mali200 may hang with some tile writeback configurations	X	X	X			

ID	Cat	Summary of Erratum	r0p4	r0p4-rel1	r0p4-rel2	r0p5	r0p5-rel1	r0p5-02rel0
481021	Cat 3	Texture data may be read for textures of type NO_TEXTURE	X	X	X			
499664	Cat 3	Occasional wrong fragment depth	X	X	X			
499713	Cat 3	Writeback bounding box combined with downsampling may hang	X	X	X			
499714	Cat 3	The Polygon List Builder Initial command list address register contains wrong value.	X	X	X			
499914	Cat 3	Inaccurate rasterization	X	X	X			
502119	Cat 3	Dithering is lost when downsampling	X	X	X			
502122	Cat 3	Writeback bounding box does not work with interleaved output	X	X	X			
509019	Cat 3	Tile buffer writeback may use unaligned AXI burst for some settings	X	X	X			
524463	Cat 3	Verbatim32 textures give unexpected results	X	X	X	X	X	X
524466	Cat 3	Numeric instability in Z-calculation	X	X	X			
526764	Cat 3	PLBU heap address not completely updated when written to memory	X	X	X			
547565	Cat 3	OpenGL ES conformant dithering is only supported for formats with 8-bit per channel	X	X	X			
547566	Cat 3	Paletted formats cannot be used together with non-paletted formats	X	X	X	X	X	X
547567	Cat 3	Palette cache livelocks.	X	X	X	X	X	X
548681	Cat 3	Impossible to start bus after it has been stopped by performance counter event.	X	X	X	X	X	X
549015	Cat 3	Rendezvous falsely terminates pixels	X	X	X	X	X	X
551122	Cat 3	Load stencil buffer from fragment shader doesn't work.	X	X	X			
551265	Cat 3	Dirty bit write-back doesn't work for Z/stencil buffer	X	X	X			
559920	Cat 3	Discontinuous line strips or line loops	X	X	X	X	X	X
618016	Cat 3	nRESET used as input to flip-flops	X	X	X	X	X	X
618017	Cat 3	Aliased points with non integer size are not OpenGL ES conformant	X	X	X	X	X	X
618666	Cat 3	Too much data might be written to the framebuffer when using the WBx_DIRTY_BIT feature	X	X	X	X	X	X

ID	Cat	Summary of Erratum	r0p4	r0p4-rel1	r0p4-rel2	r0p5	r0p5-rel1	r0p5-02rel0
618674	Cat 3	Random mipmaplevel and centroid mapping when some fragments in the pixel quad has ended execution	X	X	X	X	X	X
618716	Cat 3	Performance counter "Instruction failed rendezvous count" counts incorrectly	X	X	X	X	X	X
654667	Cat 3	The early-z performance counter value is too low	X	X	X	X	X	X
654867	Cat 3	Exponent wrap around for scaled FP16 additions	X	X	X	X	X	X
655369	Cat 3	The point size data type is too wide	X	X	X	X	X	X
718652	Cat 3	WBx_TARGET_PIXEL_FORMAT 4 and 5 do not work when writing back the alpha component	X	X	X	X	X	X
719233	Cat 3	Tile write-back might use too short bursts	X	X	X	X	X	X
719236	Cat 3	Points and lines following a zero area triangle will be discarded	X	X	X	X	X	X
719237	Cat 3	Incorrect performance counter 28 and 46 for the pixel processor	X	X	X	X	X	X
719238	Cat 3	Writeback-targets with different downsampling and bounding box does not work in some cases	X	X	X	X	X	X
719239	Cat 3	Rasterization fails for large triangles	X	X	X	X	X	X

Errata - Category 1

549166: AXI masters don't properly handle write transactions correctly in some cases

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 1, Present in: r0p4, Fixed in r0p4-rel1.

Description

The problem is a failure of the Mali AXI masters to properly handle write transactions in the case where the AXI interconnect and memory sub-system connected to the master port of GX525 is able to absorb a full data burst on the data write channel (W*) before accepting the corresponding address on the write address channel (AW*) at the interface of GX525. If then an internal master in a Mali core requests a new write transaction, the AXI master may loose the address of the first write transaction. This will result in address/data mismatch between the AXI write data and address channels which can create graphics core or, in worst case, system lock-up.

The bug is present in two location in GX525, one in the vertex processor AXI master and one in the fragment processor AXI master.

Implications

To trigger this bug, the following criterias has to be met:

- The MMU transaction queue has to be full
- The system bus must be able to receive the whole write data burst before address is received. (One burst can be only one beat)
- No write transactions can already be in the MMU queue. The MMU transaction queue is combined for both reads and writes. If there is a write transaction in the queue, write data from the second transaction is most likely not accepted on the system bus. (not valid in all systems)

When the above criteria are met, wrong data can be written and there will be loose data on the bus that can hang the AXI bus, and eventually the whole system.

Workaround

No workaround available.

564465: ActiveQuads register reset by internally generated combinatorial reset

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 1, Present in: r0p5, Fixed in r0p5-rel1.

Description

Note: This errata is the same as errata 618780, but is duplicated to track it separately for r0p4 and r0p5.

The 8-bit ActiveQuads register in the M200_toplevel_livelock_monitor module has an internally generated asynchronous reset that is generated from a logical OR of a four bit state register. This has implications both in scan mode and in normal mode.

Implications

Scan Mode

If the ActiveQuads register is included in the scan chain, the internal generated reset will corrupt the scan chain if it is not gated during scan. This will normally be handled by the scan insertion tool.

Normal Mode

The internally generated asynchronous reset on the ActiveQuads register can produce a glitch when the state register changes from one value to another. The worst case scenario happens if this glitch occurs when GX525 is in a live lock situation, and the glitch resets all but the most significant bit. The resulting value for the ActiveQuads register will then be zero which is an illegal value. This can make the GX525 stop processing pixels giving an incomplete frame as a result. The built-in watchdog timer will trigger after the number of cycles defined in the WATCHDOG_TIMEOUT register, resulting in a hang interrupt. The GX525 can then be restarted normally after a reset of the core.

Workaround

None

618780: ActiveQuads register reset by internally generated combinatorial reset

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 1, Present in: r0p4,r0p4-rel1, Fixed in r0p4-rel2.

Description

Note: This errata is the same as errata 564465, but is duplicated to track it separately for r0p4 and r0p5.

The 8-bit ActiveQuads register in the M200_toplevel_livelock_monitor module has an internally generated asynchronous reset that is generated from a logical OR of a four bit state register. This has implications both in scan mode and in normal mode.

Implications

Scan Mode

If the ActiveQuads register is included in the scan chain, the internal generated reset will corrupt the scan chain if it is not gated during scan. This will normally be handled by the scan insertion tool.

Normal Mode

The internally generated asynchronous reset on the ActiveQuads register can produce a glitch when the state register changes from one value to another. The worst case scenario happens if this glitch occurs when GX525 is in a live lock situation, and the glitch resets all but the most significant bit. The resulting value for the ActiveQuads register will then be zero which is an illegal value. This can make the GX525 stop processing pixels giving an incomplete frame as a result. The built-in watchdog timer will trigger after the number of cycles defined in the WATCHDOG_TIMEOUT register, resulting in a hang interrupt. The GX525 can then be restarted normally after a reset of the core.

Workaround

None

653167: The MMU_AXI module misses the RLAST signal in some conditions

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 1, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1, Fixed in r0p5-02rel0.

Description

The MMU always reads data (page table data) using 4 beat bursts on the AXI bus. The MMU waits for such transactions to end by looking for RLAST and RVALID going high in the same cycle qualified by the bus ID for the MMU. However there is an error in this logic so that the MMU will miss the end of a burst if RLAST was high in the cycle before.

This situation can occur due to two different scenarios:

1. When RVALID is not asserted RLAST can according to the AXI specification be of any value, so if it happens to be high in the cycle before the cycle where both RLAST and RVALID are asserted, the bug is triggered.

This situation can:

Either come from the way the slave is designed, it can e.g. assert the RLAST after the second last beat has been delivered, and then later assert RVALID when the last beat is delivered.

Or if the slave that delivers the data to the MMU is slow, depending on the actual bus matrix used, another slave can be selected while waiting for the last beat, if RLAST on the other slave was high when it switched back to the MMU slave, RLAST would then have been high in the cycle before the last beat, and thus triggering the bug.

2. If a transaction with a different ID ended in the cycle before the MMU transaction, both RVALID and RLAST would be high, so the MMU would then not detect the last beat of the MMU transaction that comes in the next cycle since RLAST was high in the previous cycle. If read interleaving is not used in the system, this particular scenario will never happen.

If interleaved reads are not supported in the system and one can in addition guarantee that RLAST is always low in the cycle before the last beat in the MMU read bursts, the bug will not be present in that particular system.

Implications

When the bug is triggered the page table data will not be updated correctly, and some MMU requests might never complete which will in turn result in a GX525 hang.

If a hang situation occurs the watchdog timer will trigger and a hang interrupt will be issued, the GX525 can then recover from the hang after a top level reset of the core. If the core hangs it will not hang or lock the AXI bus.

Workaround

Do not use the MMU inside GX525. It is possible to compile the Mali software drivers so that they don't use the MMU. This imposes limitations on how graphics memory can be allocated in the system, but by disabling the MMU the described bug will never be triggered.

Errata - Category 2

460049: Depth clipping can be done wrong when combined with polygon offset or shaders that write to Z

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 2, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

According to the OpenGL standard, clipping against the Z near/far planes are logically done before Polygon-Offset and replacement of Z value by shader output. Mali200 does not do this, instead rejecting against clip planes AFTER Polygon-Offset and Z-replacement.

Implications

When using polygon offset or writing to Z, fragments can be incorrectly clipped.

Workaround

None

509016: MaliGP2 doesn't support normalized inputs

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 2, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

The OpenGL specification allows the user to specify some values in normalized format, where the input is interpreted as a value in the range [0, 1] or [-1, 1]. MaliGP2 does not support these formats directly, making it perform worse than otherwise possible for these kinds of inputs.

Implications

Normalized inputs will be converted with a small percentage error. This error will usually give a slightly different output color of a pixel. However, in cases where a normalized input is later used in the shader to calculate an index, severe artifacts can occur.

Workaround

For a correct workaround, treat the input as fixed point, then multiply with an appropriate correction factor in the shader.

For a fast workaround, treat the input as a fixed point number with zero integer bits. This will approximate the correct value well enough for most uses. If the value is used as an index, this method may be too inaccurate.

526763: Occasional Early-z errors

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 2, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

Early-z calculates zmin and zmax-values for each level of the hierarcical rasterizer. These values are calculated using the formula z=a*x+b*y+c. If zmin/zmax are in the range [0,1] and abs(c) > 1, the zmin/zmax-values may be calculated wrong.

The incorrect values for zmin/zmax can be calculated as follows:

z_wrong = z_right / 2^(1+floor(log2(abs(c))))

This does not affect the final z value in any way, only the earlyz-mechanism in the hierarchical rasterizer.

zmin and zmax together forms a range of z-values a primitive can be within. This z-range dictates whether or not a primitive should be future subdivided by the hierarcichal rasterizer. When both zmin and zmax falls outside of the range of z-values for a patch and the z-function is set to discard those z-values, the primitive is discarded. Thus, when both the zmin and zmax are between [0,1] for a given patch and at the same time falls on the opposite side of the z-function, a patch may wrongly be discarded.

The problem has high probability for triangles with completely random z-values, but for normal scenes, the probability is very low.

- The error can only occur on half of the pixels on the screen due to implementaion details.
- When abs(c)>=2 the chance of the error to occur is greatly reduced because then only one of min,zmax will be wrong within each patch.
- Normal scenes has z-values close to each other, and when the z-values are close, the error can only happen when z is very close to 1, and for these values, the probability of the error is
- near zero.
- Since the incorrect z-values are always less than the correct value, the error will never happen when the scene is composed of triangles with z-function "less than" unless the "error-z" covered a patch completely, and there is a lot of overdraw where the last drawn thing is between the error-z and the correct-z
- For z-function "greater than" (which is not the common case), the error can happen more often, as it does not require an error-z that completely cover a whole patch to be drawn first.

Implications

Occational fragments may be lost when early-z test is enabled.

Workaround

Disable the EARLYZ_ENABLE bit in the FEATURE_ENABLE register.

551322: Blending doesn't work with FP16 tile-buffers

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 2, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

Blending doesn't work with FP16 tile buffers because of an error in the FP16 adder used.

Implications

The resulting color of a blended pixel will be wrong when using FP16 format.

Workaround

Perform floating point blending in the fragment shader, or don't use floating point tile buffer.

555370: Vertex storer cache can loose data and write data to the wrong address

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 2, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

With some output stream configurations where varyings for one vertex are located far apart in memory there is a possibility that the vertex storer cache loses data or writes it to the wrong address.

Implications

For some output stream configurations where varyings for one vertex are located far apart in memory there is a possibility that the vertex storer cache loses data or writes it to the wrong address. This will eventually result in a wrong vertex value beeing read by the pixel processor. In the end this will give visible artifacts, vertices misplaced or with wrong color etc.

Workaround

None.

650468: Shader-program driven texture prefetch can cause hang

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 2, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The TRM says that the ignore texture unit output flag and the reset texture summation flag can, together, be used to perform a shader-program software driven texture prefetch. This feature does not work and should not be used. It is not used in either the OpenVG or the OpenGL ES drivers.

Implications

If shader-program software driven texture prefetch is used, the texture bandwidth will increase and can also cause M200 to hang.

If a hang situation occurs the watchdog timer will trigger and a hang interrupt will be issued, the GX525 can then recover from the hang after a top level reset of the core. If the core hangs it will not hang or lock the AXI bus.

Workaround

Do not use the Shader-program driven texture prefetch feature. More specifically in the texture instruction, bit [43] (Ignore texture unit output) must always be 0, and bit [42] (Reset texture summation) must always be 1.

716163: The Geometry Processor does not wait for BRESP

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 2, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

Since the Geometry Processor does not wait for BRESP to be signaled, it can read old random data if it reads from the same address/addresses as it has recently written to. Write data is not guarantied to be stored in memory before BRESP is signaled for the given write transaction.

There are three known cases where this bug can result in unwanted behaviour.

Case 1: Store/Load of Vertices

If the vertex loader read a recently stored vertex, it might read random data instead of the recenty stored vertex.

Case 2: Store/Load of PLBU configuration registers.

If a recently stored configuration register is loaded back, a random register value might be loaded.

Case 3: Store/Load of PLBU pointer array.

If a recently stored pointer array is loaded back, random pointers might be loaded.

Implications

Case 1:

It will lead to wrong vertex data being used which in the end can result in various artefacts in the final output. If e.g. the vertex data in question is a part of a big triangle that is in the foreground it can be quite visible.

Case2:

The Geometry Processor could use errounous register values and operate in non predictable manners.

Case 3:

The PLBU can write the command list to random addresses.

Workaround

To ensure that all useful data is written to memory before it is read back, dummy write transactions must be added to the same AXI ID. Since the AXI specification ensures ordering of write data to the same ID, the last useful data is guarantied to be stored in memory if the number of dummy transactions added is equal or higher than the systems total write acceptance capability.

Case 1:

A dummy vertex shader must executed after each normal vertex shader job that uses DrawElements. The dummy shader must generate as many write transactions as the systems total write acceptance capability.

Case 2:

When PLBU configuration registers are stored, repeat the last store at least as many times as the systems total write acceptance capability.

Case 3:

When the pointer array is flushed, add a second flush command. This will add 21 write transactions. If the systems total write acceptance capability is even higher than 21, several flush commands can be inserted.

719242: Varying interpolation get reduced precision for large triangles

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 2, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

Triangles with large x or y - coordinates in window-space (usually caused by w close or equal to 0) may cause precision issues during triangle setup. This will in turn affect varying interpolation. Varying interpolation will lose more and more precision when at least one vertex has a screen-space coordinate greater than 64k pixels from origo and fail completely when it is greater than 640k pixels from origo. The primitive must intersect the screen to cause visible failures.

Implications

Precision for varying interpolation is gradually lost for triangles wider than / higher than 64k pixels up to 640k pixels in windows space coordinates. Triangles wider than 640k pixels will have completely random varying values.

Workaround

In the vertex-shader, low clamp abs(w) >= epsilon * max(abs(x), abs(y), abs(z)) before the perspective division

epsilon should ideally be as small as possible, but must be >=1.5e-6 to remove varying interpolation failures. >=1.5e-5 should guarantee high precision for varying interpolation.

If the workaround above is considered to expensive, and the maximum range of the input-values is known, max(abs(x), abs(y), abs(z)) can be replaced with the maximum range of the input values and incorporate it into the constant, such that the workaround would be abs(w) >= epsilon instead.

Errata - Category 3

475813: When out of memory, the list builder may repeatedly write to a single address

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

When the polygon list builder runs out of memory, it will signal an interrupt and wait for software to allocate more memory. While waiting for this, the list builder will repeatedly write the last completed command to the last allocated address, wasting memory bandwidth.

Implications

Memory bandwidth will be wasted. If resetting the core while it signals out of memory, the bus must be explicitly stopped to avoid leaving the bus in an inconsistent state.

Workaround

If handling the out of memory condition takes a long time and the vertex shader is idle, memory bandwidth consumption may be reduced by stopping the bus interface until memory has been allocated.

475814: MaliGP2 does not enforce alignment in all cases

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

In some cases, it is possible to specify addresses that do not match the required alignment, causing unintended or invalid bus transactions.

When loading data that are more than 32 or 64 bytes long but not 32 or 64 byte aligned, the processor may do unaligned bursts. These bursts will be legal AXI transfers, but the Mali cores are supposed to align all bursts to the size of the burst. This applies to e.g. loading input and output configuration, loading constant registers, or loading shader programs.

When loading or flushing the polygon list builder pointer array, the base address should be aligned to a 64 byte boundary. The hardware does 64 byte bursts, but only enforces 16 byte alignment. If specifying a base address that is not 64-byte aligned, the core will do unaligned bursts. If the unaligned address modulo 4096 is larger than 2880, the core may additionally burst across a 4 kB boundary in violation of the AXI specification.

Implications

Invalid configurations may cause MaliGP2 to behave differently on the bus than intended.

Workaround

Always ensure that correctly aligned addresses are used.

476265: Reading of tile writeback big endian bit fails

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

The Big Endian bit in the WBx Target Flags registers always reads as zero.

Implications

The big endian writeback mode can be correctly configured, but it is not possible to check which mode is currently configured.

Workaround

Instead of querying the core, remember whether each stream is configured as little or big endian.

481018: Mali200 may hang with some tile writeback configurations

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

Mali200 has 3 tile writeback units. If all of them are disabled, or they are configured such that a disabled writeback unit is followed by an enabled one, there is a small chance that the core may deadlock.

Implications

Not all possible combinations of tile writeback settings may be safely used.

Workaround

Ensure that writeback unit 0 always is enabled, and that writeback unit 1 always is enabled if writeback unit 2 is enabled.

481021: Texture data may be read for textures of type NO_TEXTURE

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

Texture samplers of type NO_TEXTURE always reads as black, but the texture mapper may still read data from memory.

Implications

Texture descriptors with type NO_TEXTURE must still point to a valid and readable memory location. If they do not, memory access exceptions may occur. If they do, a small amount of unneccesary texture reads are performed.

Workaround

Set texture size to 1x1, and use the address of the texture descriptor as the texture data address.

499664: Occasional wrong fragment depth

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

Mali200 uses an early-z technique to discard fragments that will be entirely hidden behind already-drawn fragments. This speeds up rendering by not having to run the fragment shaders on fragments that cannot be visible. In most cases both the depth test and updating the depth buffer can be done before the shader is run. In some cases, e.g. if the shader can modify the fragment depth, the depth test and depth buffer update must be postponed until the shader completes.

The early-z module has a race condition where, in some rare cases, a fragment may use the early-z path even though another fragment at the same position uses the late-z path. When this happens, the two fragments may get wrong depth values, and may cause rare pixel artifacts.

The full conditions required for this to occur are:

- Two pixel quads covering the same position in the frame buffer enter the early-z module with exactly 12 cycles between the quads.
- The first quad has the early-z-update flag disabled.
- The first quad uses the early-z path, i.e. there are no other active quads that force it into the late-z path.
- The second quad has the early-z-update flag enabled.
- The second quad is also allowed to use the early-z path, erronously ignoring the first quad.
- The second quad has a Z value causing the Z buffer to be updated.
- The first quad has a Z value between the original value in the Z buffer and the Z value written by the second quad.

Implications

There is a small but non-zero chance that rendered images may have single-pixel rendering artifacts.

Workaround

Disable the early-z-update bit and the early-z-test bit in all Render State Words (RSWs).

499713: Writeback bounding box combined with downsampling may hang

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

If a writeback unit is set to downsample the rendered image, and the bounding box is set to a height that is not a multiple of the downsampled tile size, the writeback unit will hang.

Implications

Not all combinations of bounding boxes and downsampling can be used.

Workaround

Depending on the exact requirements, several workarounds are possible:

- Allocate frame buffers that are a multiple of effective tile size, and disable the bounding box.
- Render the last tile row partially overlapping the previous row, so that the entire height of the last row can be drawn.
- Render the last row to a separate buffer, and copy the required portions of this buffer to the frame buffer.

499714: The Polygon List Builder Initial command list address register contains wrong value.

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

The GP_CONTR_REG_PLBCL_INITIAL_ADDR register should contain the value originally written into the GP_CONTR_REG_PLBCL_START_ADDR register, but only contains bits [28:0] left shifted 3 bits.

Implications

The read back value of the GP_CONTR_REG_PLBCL_INITIAL_ADDR register cannot be used.

Workaround

Remember the address written to the GP_CONTR_REG_PLBCL_START_ADDR register.

499914: Inaccurate rasterization

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

The Mali200 rasterizer hierarchically splits primitives into 8x8, 4x4, and 2x2 patches, until 1x1 fragments leave the rasterizer. In some rare cases, the rasterizer could wrongly split these patches so that all the fragments in a patch could be lost.

This happens if:

- One vertex is within 1/16th of a pixel from the upper left corner of the patch.
- The second vertex is within 1/16th of a pixel from the upper right corner, and the third vertex is within 1/16th pixel of the right edge of the patch, or
- The second vertex is within 1/16th of a pixel from the lower left corner, and the third vertex is within 1/16th pixel of the bottom edge of the patch.
- The polygon does not cover the lower right corner of the patch.

Implications

In rare cases, all fragments from a primitive may be lost in an 8x8, 4x4, or 2x2 area of the screen.

Workaround

Set the subpixel specifier to 1/16th pixel using the SUBPIXEL_SPECIFIER register.

502119: Dithering is lost when downsampling

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

In OpenGL, dithering is a fragment operation that is done right after blending. Mali200 can dither before writing blended values back to the tile buffer.

When downsampling is enabled, multiple tile buffer pixels are averaged to form a single color buffer pixel. When this is done, the previously applied dither pattern is averaged away.

Implications

The dithering pattern is reduced on 4x downsampling and lost when using 16x or more on downsampling.

Workaround

If all fragments have dithering enabled, enable dithering in the writeback module.

If only some fragments have dithering enabled, no strictly compliant workaround exists.

502122: Writeback bounding box does not work with interleaved output

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

The writeback bounding box does not work correctly if the frame buffer is configured as interleaved. Both pixels inside and outside the bounding box may be written with wrong data.

Implications

Writeback bounding box does not work with interleaved output.

Workaround

Do not use writeback bounding box on interleaved color buffers.

509019: Tile buffer writeback may use unaligned AXI burst for some settings

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

The tile writeback is supposed to do all it's writes using aligned power-of-two-length bursts. In some cases it may do unaligned bursts, potentially violating the AXI specification's restrictions regarding bursting across 4 KiB boundaries.

The issue may occur if a linear framebuffer is aligned to a 64-byte boundary, but the scan line length is not a multiple of 64 bytes.

Implications

Mali200 will violate the AXI specification restrictions regarding bursting across 4 KiB boundaries.

Workaround

Make sure the frame buffer base address is not 64-byte aligned in the cases where the issue may occur. The table below describes the settings that may cause the issue and corresponding workaround.

Downsampling	Scanline length	Frame buffer format	Workaround
disabled, 2x	Bit [3] is set	8-64bit	WBx_TARGET_ADDR bit [3] must be set
disabled, 2x	Bit [4] is set	16-64bit	WBx_TARGET_ADDR bit [4] must be set
disabled, 2x	Bit [5] is set	32-64bit	WBx_TARGET_ADDR bit [5] must be set
4x, 8x	Bit [3] is set	16-32bit	WBx_TARGET_ADDR bit [3] must be set
4x, 8x	Bit [4] is set	32bit	WBx_TARGET_ADDR bit [4] must be set
16x, 32x	Bit [3] is set	32bit	WBx_TARGET_ADDR bit [3] must be set

524463: Verbatim32 textures give unexpected results

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

Sampling texels from a texture using the 32-bit verbatim copy texture format may occasionally return wrong data. When the texture sampling misses in the texture cache, the x, z, and w components may be logically OR'ed with the value 0x3c00.

Implications

The verbatim32 copy texture format is used to load Z and texture buffers for multi-pass rendering. This may cause the loaded Z-value to be incorrect, causing occasional rendering glitches.

Workaround

For r0p1 and all prior versions there are no workarounds (see defect #486268).

For all versions after r0p1:

Use two texture descriptors pointing to the same verbatim-texture, one of them with the "full order invert" bit set. Sample from both descriptors, and use the y component from both samples. The y component with full order invert is the same as the x component without the inversion, but will always be correct.

For r0p5:

When there is 1:1 mapping between texels and pixels a valid workaround is to add 0.25 to the s and t texture-coordinates for the verbatim-texture.

524466: Numeric instability in Z-calculation

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

Before interpolating Z values across a polygon, Mali200 calculates the derivatives dz/dx and dz/dy. If two Z-values are very close, cancelation causes the dz value to have few bits of precision. If these two vertices are also very close in the x or y plane, the corresponding dx or dy is very small, causing dz/dx or dz/dy to amplify any rounding errors and loss of precision.

When these derivatives are used to interpolate depth values, the interpolated result may be off by some bits.

This is most likely to occur for long narrow triangles. The typical error is on the order of 2^-20, but is potentially unbounded if the triangle becomes much less than 1 pixel wide.

The issue has been improved for the r0p5 release, but a complete fix require infinite precision in these calculations.

Implications

Some fragments may deviate slightly from their correct depth value.

Workaround

Avoid long narrow polygons with the short axis significantly less than one pixel.

526764: PLBU heap address not completely updated when written to memory

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

If the current PLBU heap allocation address is written to memory, and a new heap block is allocated by one of the last three primitives, the wrong address is written to memory.

If this is later read back for a new draw call, the same block may be reallocated, and data may be corrupted.

Implications

Polygon lists may be corrupted, causing rendered image to be wrong.

Workaround

Insert dummy commands, e.g. set base address commands, to ensure all important commands are written before writing the heap address. If 3 or more dummy commands are inserted, only dummy commands can be lost, and these should not affect the rendered image.

547565: OpenGL ES conformant dithering is only supported for formats with 8-bit per channel

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

OpenGL ES conformant dithering is only supported for formats with 8-bit per channel such as RGBA8888. The effect of dithering is lost if write back to a target format with less precision is performed.

Implications

No dithering is available in OpenGL ES for formats with less than 8 bits per channel.

Workaround

None.

547566: Paletted formats cannot be used together with non-paletted formats

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

Paletted formats cannot be used together with non-paletted formats.

Implications

Texture data may be corrupted.

Workaround

Convert the paletted textures to a non paletted format.

547567: Palette cache livelocks.

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The palette cache has a possibility to livelock.

Implications

Livelock can occur when using paletted texture formats. The livelock will stop the rendering process and the core will eventually give a hang interrupt.

Workaround

Convert the paletted textures to a non paletted format.

548681: Impossible to start bus after it has been stopped by performance counter event.

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The bug only manifests itself when both the the PERF_CNT_X_LIM_EN and PERF_CTN_ENABLE bit is cleared at the same time after the limit has triggered. If they are both cleared at the same time the bus will hang.

Implications

Its is impossible to start bus after it has been stopped by performance counter event.

Workaround

Turn off the performance counter in two steps as follows:

- Turn off the PERF_CNT_0_LIM_EN bit
- Turn off the PERF_CNT_0_ENABLE bit

549015: Rendezvous falsely terminates pixels

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

none

Implications

The problem can occur whenever all of the following conditions are met for the

4 threads in a 4-pixel group:

- None of the threads have been terminated
- All of the threads are executing the same instruction
- This instruction has the rendezvous flag set
- At least one of the threads will terminate right after executing this instruction (having this instruction as the last instruction executed)
- At least one (but not all) of the threads is marked as killed by an earlier instruction.

When all of the above conditions exist the color of some of the pixels in a pixel quad will be wrong.

Workaround

If there exists an instruction which can execute a pixel kill or alternative pixel kill without terminating the program, then any instruction which can be the last instruction executed must not have the rendezvous flag set.

551122: Load stencil buffer from fragment shader doesn't work.

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

Reading back the Z and stencil buffer from the fragment shader does not work.

Implications

The value returned from the Read framebuffer Z and stencil command is wrong, so this command can't be used.

Workaround

None.

551265: Dirty bit write-back doesn't work for Z/stencil buffer

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2, Fixed in r0p5.

Description

Dirty bit write-back doesn't work for the Z/stencil buffer. More specifically, setting the WBx_DIRTY_BIT_ENABLE bit, and with WBx_SOURCE_SELECT = 1 (Z/Stencil buffer).

Implications

The resolution of the dirty bits are not per pixel based as they should. If one of the pixels in a 2x2 quad is dirty, the whole quad will be written back. The lower left pixel quad in each tile will also always be written back even if it is not dirty.

Workaround

None.

559920: Discontinuous line strips or line loops

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

To approximate the GLES diamond-exit rule, all aliased lines are offset by a small amount along the direction of the line. This offsetting may move a line by one pixel, which in itself is not a problem. However, if two connected lines make a ~90 degree angle, the offset will be in two different directions for the first and second line. This might lead to a discontinuity between the two lines. The issue is only relevant when both lines has the same major-direction, which means that the first line must have an angle close to and greater than 45+90*n degrees while the other line must have an angle close to but less than 45+90*(n+1) degrees.

Implications

When two aliased lines sharing the same vertex have the same major direction, a discontinuity may occur when the two lines form an edge of close to but not more than 90 degrees.

Workaround

None.

618016: nRESET used as input to flip-flops

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The following three one bit registers have the asynchronous reset signal connected to the data input:

MOD_M200_M120PP_toplevel_1/M200_MOD_M120PP_core_0/M200_MOD_Rasterizer_5/M200_MOD_rast_fifos_2/M200_MOD_rast_fifos_3_2/FIFO_Full

MOD_M200_M120PP_toplevel_1/M200_MOD_M120PP_core_0/M200_MOD_Rasterizer_5/M200_MOD_rast_fifos_2/M200_MOD_rast_fifos_2_1/FIFO_Full

MOD_M200_M120PP_toplevel_1/M200_MOD_M120PP_core_0/M200_MOD_Rasterizer_5/M200_MOD_rast_fifos_2/M200_MOD_rast_fifos_2_1/FIFO_Full

This violates DFT and linting rules. As a DFT violation it is classified as a warning.

Implications

As long the setup and hold requirements for the D input of the flip-flops are held, there are no real issues.

Workaround

No workaround needed.

618017: Aliased points with non integer size are not OpenGL ES conformant

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

For non-integer point sizes, the points might end up non-quadratic (by one pixel only) due to the order in which calculations are done on aliased point sprites.

Implications

For non-integer point sizes, the points might end up non-quadratic (by one pixel only).

Workaround

Round the point-sizes of all aliased points to integer values in the vertex shader.

618666: Too much data might be written to the framebuffer when using the WBx_DIRTY_BIT feature

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The dirty bit is not correctly masked with the RGB write mask and "pixel killed", when RSW early-Z testing is enabled.

Implications

Some pixels that don't have the dirty bit set might be written to the frame buffer even if the WBx_DIRTY_BIT_ENABLE flag is enabled.

Workaround

Disable RSW early-Z testing, (subword 13, bits 8 and 9), when the shader uses pixel-kill and you want to use dirty bit write back.

618674: Random mipmaplevel and centroid mapping when some fragments in the pixel quad has ended execution

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The selected mipmap level and the centroid mapping result is non deterministic when some fragments in the pixel quad has ended execution before texturing. This can only be triggered by fragment shader programs with a conditional pixel kill before a texturing instruction. Such a program will by the OpenGL ES standard result in undefined result and should not be used.

Implications

OpenGL-ES defines the result to be undefined for these cases, for GX525 this value can be indeterministic as well.

Workaround

Do one of the following:

Either:

- Move the conditional pixel kill instruction after the texturing instruction.

Or,

- Always have an empty varying-instruction at the end of each program if there exist varying-lookups inside conditional code-paths, and make sure the last instruction is the one with the highest memory-address of all instructions in the program.
- Do not set the end-flag on a pixel kill instruction.
- Make all potential exits of the program exit at the same instruction.

618716: Performance counter "Instruction failed rendezvous count" counts incorrectly

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The performance counter number 38 "Instruction failed rendezvous count" counts incorrectly.

Implications

The value for the performance counter number 38 "Instruction failed rendezvous count" will be higher than it should have been.

Workaround

None.

654667: The early-z performance counter value is too low

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The Performance counter number 35 (Patches rejected early z/stencil count), only counts some of the rejected patches so the counter value will in general be too low.

Implications

The early-z counter is only used for performance analysis, so the only implication is that the performance analysis will report a lower number of early-z cullings than what is actually culled.

Workaround

None.

654867: Exponent wrap around for scaled FP16 additions

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The scalar and vector adder in the fragment shader can give exponent wrap around when using scaled FP16 additions, operation(1, 2, 3, 5, 6, and 7). It will only happen for the two cases described below.

- If the result before scaling is less than infinity, and the expected result after scaling is infinity, the result will be 0 instead of infinity.
- If the result before scaling is greater than zero, and the expected result after scaling is zero, the result will be infinity instead of zero.

Implications

If one of the two cases described above occur during the execution of a fragment shader program, artifacts may occur.

Workaround

A general workaround would be to not use scaled additions. More specifically, this means adder operations 1, 2, 3, 5, 6, and 7. Instead of using scaled additions use the normal adder operation (operation 0) together with a multiply operation.

The workaround is not required if the absolute value of the result is known to be greater than zero and less than infinity.

655369: The point size data type is too wide

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

Data type 33, (GP_VS_VSTREAM_FORMAT_POINT_SIZE), specified in the vertex data format register, (GP_VS_CONF_REG_OUTP_SPEC(X)), specifies a 32 bit data type, but it is interpreted as a 128 bit data type.

Implications

One point size might overwrite an existing point size. Some points can then have the wrong size in the resulting image.

Workaround

- For glDrawArrays mode, four 32 bit words need to be allocated for outputting the point size instead of only one word. The allocated memory must be 16-byte aligned.
- For glDrawElements mode, data type 0 (GP_VS_VSTREAM_FORMAT_1_FP32) should be used instead of data type 33 (GP_VS_VSTREAM_FORMAT_POINT_SIZE).

718652: WBx_TARGET_PIXEL_FORMAT 4 and 5 do not work when writing back the alpha component

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The Alpha component is forced to one when writing back using the fixedpoint writeback formats with one or two components (writeback formats 4 and 5).

Implications

When writing back a single or two components of the color-buffer using WBx_TARGET_PIXEL_FORMAT 4 and 5, the alpha channel cannot be selected.

Workaround

When less than four components of the color-buffer are being used, use the RGB-components to store the values and not the alpha component.

When four components of the color-buffer are being used, use writeback format 3 ARGB8888 to write all of the components in one block and separate the alpha component by other means afterwards.

719233: Tile write-back might use too short bursts

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

Calculation of maximum burst size for tile write-back is wrong when using the WBx_TARGET_LAYOUT register is set to 2 (interleaved blocks) and the resolution is not a multiple of 128.

Implications

The Pixel Processor will use many short bursts instead of the maximum allowed size when the WBx_TARGET_LAYOUT control register, is set to 2 (Interleaved blocks) and the framebuffer size is not a multiple of 128. This should in most cases not affect the performance, since tile write back is seldom the bottleneck. It will however add extra load on the memory sub-system.

Workaround

Two workarounds are available:

- Use another write-back target layout. The layout can be configured by writing to the WBx_TARGET_LAYOUT control register. Use target layout 0 (linear layout) or layout 1 (interleaved layout) as these layouts does not have the defect.
- Pad the framebuffer in x-direction to a multiple of 8 blocks when the write-back target layout must be 2 (block-interleaved). This will give maximum burst length for the write-back. The number of blocks in the x-direction is set by the WBx_TARGET_SCANLINE_LENGTH register.

719236: Points and lines following a zero area triangle will be discarded

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

A flag indicating that a primitive should be discarded due to zero area is not properly updated for points and lines.

Implications

Points and lines following a zero area triangle will be discarded. In most situations zero area triangles will be discarded early in the pipeline, so the defect will only rarely be triggered.

Workaround

Before each draw call with points or lines, draw a fullscreen quad (or with the size of the bounding box of the point-array). This fullscreen quad should have z-function "never" and shader length=0. EarlyZ (register) should be enabled for this workaround.

or:

Clamp the vertices of every triangle/quad to the same grid in the vertex shader as the one used by triangle setup.

719237: Incorrect performance counter 28 and 46 for the pixel processor

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

Performance counter 28 (Stall cycles PolygonListReader) counts the number of items in the polygon list instead of the number of cycles the polygon list reader is stalled.

Performance counter 46 (Texture mapper multipass count) does not count all events it's supposed to.

Implications

Performance counter 28 (Stall cycles PolygonListReader) and 46 (Texture mapper multipass count) for the pixel processor can not be used.

Workaround

N/A

719238: Writeback-targets with different downsampling and bounding box does not work in some cases

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

If WBx_TARGET_AA_FORMAT differs between two writeback targets or between two frames where the Pixel Processor was not reset in between, one WBx_TARGET_AA_FORMAT has >= 32x downsampling and at least one of the BOUNDING_BOX_LEFT_RIGHT or BOUNDING_BOX_BOTTOM registers are used, the frame buffer will in some cases be shuffled within the tiles rendered after a tile using the bounding box.

Implications

Bounding boxes cannot be used when different writeback targets have different downsampling and at least one of them uses >= 32x downsampling.

Workaround

In any event:

- Reset the core between each time the downsampling has changed between frames.

And if possible:

- Turn off the bounding box when different writeback targets has different downsampling (and it is >= 32x).

otherwise:

- Render the scene twice, one for each downsampling target.

or:

- Render the scene once with lowest downsampling, then downsample it in another pass.

719239: Rasterization fails for large triangles

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Cat 3, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

Triangles with large x or y - coordinates in window-space (usually caused by w close or equal to 0) may cause precision issues during triangle setup. This will in turn affect rasterization. Rasterization will fail when two vertices are ~equally far from the screen in both directions (+/- 320M pixels) and the edge is intersecting the screen. Failures will be represented as triangles with edges moved or duplicated a multiple by 16 pixels. The number of 16 pixel blocks the edge can be moved starts at 1 and increases by a multiple of 2 for each power of 2 increase in the triangle size.

Implications

Edges of triangles wider than / higher than 640M pixels in window-space coordinates will not be rendered correctly.

Workaround

In the vertex-shader, low clamp abs(w) >= epsilon * max(abs(x), abs(y), abs(z)) before the perspective division

epsilon should ideally be as small as possible, but must be >=3.1e-9 to remove any rasterization artifacts.

If the workaround above is considered to expensive, and the maximum range of the input-values is known, max(abs(x), abs(y), abs(z)) can be replaced with the maximum range of the input values and incorporate it into the constant, such that the workaround would be abs(w) >= epsilon instead.

Errata - Documentation

559615: Wrong reset values in the TRM for viewport and screensize registers in the PLBU unit

Status

Affects: product Mali200 with MaliGP2 Core.

Fault status: Doc, Present in: r0p4,r0p4-rel1,r0p4-rel2,r0p5,r0p5-rel1,r0p5-02rel0, Open.

Description

The TRM states that the viewport and screensize registers in the PLBU are reset to 0, but they do not have a reset value. The registers are: GP_PLB_CONF_REG_VIEWPORT_TOP, GP_PLB_CONF_REG_VIEWPORT_BOTTOM, GP_PLB_CONF_REG_VIEWPORT_LEFT, GP_PLB_CONF_REG_VIEWPORT_RIGHT and GP_PLB_CONF_REG_SCREENSIZE.

Implications

If the Viewport or Screen Size registers are not set before they are used the behavior is undefined.

Workaround

The following registers have to be set before they are used: GP_PLB_CONF_REG_VIEWPORT_TOP, GP_PLB_CONF_REG_VIEWPORT_BOTTOM, GP_PLB_CONF_REG_VIEWPORT_LEFT, GP_PLB_CONF_REG_VIEWPORT_RIGHT and GP_PLB_CONF_REG_SCREENSIZE.

Errata – Driver Software

There are no Errata in this Category