



Linux OpenGL ES DDK package
Linux OpenGL ES DDK package (GX910)
OpenGL ES
Errata Notice

This document contains all errata known at the date of issue in releases up to and including revision r0p2 of Linux OpenGL ES DDK package

Proprietary notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM Limited in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

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

INTRODUCTION	7
ERRATA SUMMARY TABLE	10
ERRATA - CATEGORY 1	13
594535: GLES 1: Paletted textures with mipmaps does not work	13
595316: GLES 2: Running out of memory in glRenderbufferStorage can cause a segfault	14
595678: Running out of memory in glReadPixels can cause a segfault	15
ERRATA - CATEGORY 2	17
580918: GLES 2: Mixing CPU and GPU-based texture updates gives incorrect results	17
589866: GLES 2: glBinaryShader does not specify a standalone format	18
589867: GLES 2: Missing SW workaround for HW issue: Unaligned input data to vertex shader may corrupt previous input stream	19
594530: glSampleCoverage invert flag has wrong effect	20
594770: Many flushing operations without eglSwapBuffers or glClear can give incorrect results	21
594869: GLES 2: glBindFramebuffer disables multisampling until the next call to eglSwapBuffers	22
594871: GLES 2: Matrix varyings do not work	23
595318: GLES 2: Cubemap faces of different formats cause false GL_OUT_OF_MEMORY	24
595623: GLES 2: Updating an FBO-bound texture with GL texture functions can lead to segfault	25
595716: GLES 2: Vertex shaders that write to gl_PointSize may end up having other varyings corrupted	26
595766: GLES 1: Points with sizes set with glPointSize are rendered too small in 16xAA	27
595970: GLES 1: Enabling point sprites when rendering polygons or lines breaks texturing	28
596020: GLES 2: Rendering to FBOs and using the image as a texture can fail on multi-core Mali setups	29
602216: GLES 2: Varying arrays of mat2 type do not work	30
602217: GLES 2: Address calculation in a shader is incorrect when more than one dynamic index is used	31
602272: GLES 2: Vertex shader clamp() builtin with constant high/low values may be compiled incorrectly	33

602818:	GLES2: Sampler inside struct in fragment shader and struct in vertex shader leads to segfault	34
ERRATA - CATEGORY 3		36
589267:	GLES2: Shader linker does not do invariance checks for built-in varyings	36
589268:	glDrawElements is limited to 24bit index count	37
589269:	Multisample information may be lost	38
589270:	Previously issued draw-calls may be lost when running out of memory	39
589271:	GLES 2: gl_FragCoord upside-down when rendering to an FBO	40
589272:	Driver relies on mutex behaviour that is undefined	41
589273:	GLES 2: Calling glUniform1i on a sampler with a value larger than 7 or smaller than 0 gives segfault	42
589868:	Missing SW workaround for HW issue: MaliGP2 does not support normalized inputs	43
594532:	GLES 1: Incomplete textures output black instead of disabling their texture-unit	44
594727:	GLES 2: Incorrect return values from glGetRenderbufferParameter	45
594728:	GLES 1: Fixed-point state getters convert enum states incorrectly	46
594769:	GLES 2: glGetFramebufferAttachmentParameteriv returns the wrong value for cubemap faces if they are 2D textures	47
594867:	GLES2: Calling glDeleteFramebuffer on the bound FBO does not cause framebuffer dependent values to change	48
595972:	glClear might clear only parts of the framebuffer	49
596018:	GLES 2: Incorrect return values from glGetIntegerv after glBindFramebuffer	50
596019:	Creating EGLImages from texture mip level 10 or above results in EGL_BAD_MATCH	51
596068:	Non-integer sized points are sometimes rendered non-square	52
596069:	Modifying an EGLImage created from a pixmap within a frame can give wrong result	53
602218:	GLES 2: Compilation of a vertex shader may fail with the error "Register allocation failed for vertex shader"	54
602219:	GLES 2: Shader compiler preprocessor accepts some illegal combinations of preprocessor directives	55
602220:	GLES 2: Shader compilation may fail when the client application sets the locale to non-US	56
602370:	GLES 2: Failing to write gl_Position is reported as a shader compilation error	57
602374:	GLES 2: Shader compiler extension macros only defined for enabled extensions	58
602375:	GLES 2: Function overloading based on array sizes not supported	59
602967:	glReadPixels does not work with direct rendering	60

Linux OpenGL ES DDK package (GX910)
OpenGL ES

Date of Issue: **07-Oct-2008**

ARM Errata Notice

Document Revision **12.0**

ERRATA - DOCUMENTATION	62
There are no Errata in this Category	62
ERRATA – DRIVER SOFTWARE	63
There are no Errata in this Category	63

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

07 Oct 2008: Changes in Document v1

Page	Status	ID	Cat	Summary
13	New	594535	Cat 1	GL ES 1: Paletted textures with mipmaps does not work
14	New	595316	Cat 1	GL ES 2: Running out of memory in glRenderbufferStorage can cause a segfault
15	New	595678	Cat 1	Running out of memory in glReadPixels can cause a segfault
17	New	580918	Cat 2	GL ES 2: Mixing CPU and GPU-based texture updates gives incorrect results
18	New	589866	Cat 2	GL ES 2: glBinaryShader does not specify a standalone format
19	New	589867	Cat 2	GL ES 2: Missing SW workaround for HW issue: Unaligned input data to vertex shader may corrupt previous input stream
20	New	594530	Cat 2	glSampleCoverage invert flag has wrong effect
21	New	594770	Cat 2	Many flushing operations without eglSwapBuffers or glClear can give incorrect results
22	New	594869	Cat 2	GL ES 2: glBindFramebuffer disables multisampling until the next call to eglSwapBuffers
23	New	594871	Cat 2	GL ES 2: Matrix varyings do not work
24	New	595318	Cat 2	GL ES 2: Cubemap faces of different formats cause false GL_OUT_OF_MEMORY
25	New	595623	Cat 2	GL ES 2: Updating an FBO-bound texture with GL texture functions can lead to segfault
26	New	595716	Cat 2	GL ES 2: Vertex shaders that write to gl_PointSize may end up having other varyings corrupted
27	New	595766	Cat 2	GL ES 1: Points with sizes set with glPointSize are rendered too small in 16xAA
28	New	595970	Cat 2	GL ES 1: Enabling point sprites when rendering polygons or lines breaks texturing
29	New	596020	Cat 2	GL ES 2: Rendering to FBOs and using the image as a texture can fail on multi-core Mali setups
30	New	602216	Cat 2	GL ES 2: Varying arrays of mat2 type do not work
31	New	602217	Cat 2	GL ES 2: Address calculation in a shader is incorrect when more than one dynamic index is used
33	New	602272	Cat 2	GL ES 2: Vertex shader clamp() builtin with constant high/low values may be compiled incorrectly
34	New	602818	Cat 2	GL ES2: Sampler inside struct in fragment shader and struct in vertex shader leads to segfault
36	New	589267	Cat 3	GL ES2: Shader linker does not do invariance checks for built-in varyings
37	New	589268	Cat 3	glDrawElements is limited to 24bit index count

Linux OpenGL ES DDK package (GX910)
OpenGL ES

Date of Issue: **07-Oct-2008**

ARM Errata Notice

Document Revision **12.0**

38	New	589269	Cat 3	Multisample information may be lost
39	New	589270	Cat 3	Previously issued draw-calls may be lost when running out of memory
40	New	589271	Cat 3	GL ES 2: gl_FragCoord upside-down when rendering to an FBO
41	New	589272	Cat 3	Driver relies on mutex behaviour that is undefined
42	New	589273	Cat 3	GL ES 2: Calling glUniform1i on a sampler with a value larger than 7 or smaller than 0 gives segfault
43	New	589868	Cat 3	Missing SW workaround for HW issue: MaliGP2 does not support normalized inputs
44	New	594532	Cat 3	GL ES 1: Incomplete textures output black instead of disabling their texture-unit
45	New	594727	Cat 3	GL ES 2: Incorrect return values from glGetRenderbufferParameter
46	New	594728	Cat 3	GL ES 1: Fixed-point state getters convert enum states incorrectly
47	New	594769	Cat 3	GL ES 2: glGetFramebufferAttachmentParameteriv returns the wrong value for cubemap faces if they are 2D textures
48	New	594867	Cat 3	GL ES2: Calling glDeleteFramebuffer on the bound FBO does not cause framebuffer dependent values to change
49	New	595972	Cat 3	glClear might clear only parts of the framebuffer
50	New	596018	Cat 3	GL ES 2: Incorrect return values from glGetIntegerv after glBindFramebuffer
51	New	596019	Cat 3	Creating EGLImages from texture mip level 10 or above results in EGL_BAD_MATCH
52	New	596068	Cat 3	Non-integer sized points are sometimes rendered non-square
53	New	596069	Cat 3	Modifying an EGLImage created from a pixmap within a frame can give wrong result
54	New	602218	Cat 3	GL ES 2: Compilation of a vertex shader may fail with the error "Register allocation failed for vertex shader"
55	New	602219	Cat 3	GL ES 2: Shader compiler preprocessor accepts some illegal combinations of preprocessor directives
56	New	602220	Cat 3	GL ES 2: Shader compilation may fail when the client application sets the locale to non-US
57	New	602370	Cat 3	GL ES 2: Failing to write gl_Position is reported as a shader compilation error
58	New	602374	Cat 3	GL ES 2: Shader compiler extension macros only defined for enabled extensions
59	New	602375	Cat 3	GL ES 2: Function overloading based on array sizes not supported
60	New	602967	Cat 3	glReadPixels does not work with direct rendering

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	r0p1	r0p2
594535	Cat 1	GL ES 1: Paletted textures with mipmaps does not work	X	
595316	Cat 1	GL ES 2: Running out of memory in glRenderbufferStorage can cause a segfault	X	
595678	Cat 1	Running out of memory in glReadPixels can cause a segfault	X	
580918	Cat 2	GL ES 2: Mixing CPU and GPU-based texture updates gives incorrect results	X	X
589866	Cat 2	GL ES 2: glBinaryShader does not specify a standalone format	X	
589867	Cat 2	GL ES 2: Missing SW workaround for HW issue: Unaligned input data to vertex shader may corrupt previous input stream	X	X
594530	Cat 2	glSampleCoverage invert flag has wrong effect	X	
594770	Cat 2	Many flushing operations without eglSwapBuffers or glClear can give incorrect results	X	
594869	Cat 2	GL ES 2: glBindFramebuffer disables multisampling until the next call to eglSwapBuffers	X	
594871	Cat 2	GL ES 2: Matrix varyings do not work	X	
595318	Cat 2	GL ES 2: Cubemap faces of different formats cause false GL_OUT_OF_MEMORY	X	
595623	Cat 2	GL ES 2: Updating an FBO-bound texture with GL texture functions can lead to segfault	X	
595716	Cat 2	GL ES 2: Vertex shaders that write to gl_PointSize may end up having other varyings corrupted	X	
595766	Cat 2	GL ES 1: Points with sizes set with glPointSize are rendered too small in 16xAA	X	
595970	Cat 2	GL ES 1: Enabling point sprites when rendering polygons or lines breaks texturing	X	
596020	Cat 2	GL ES 2: Rendering to FBOs and using the image as a texture can fail on multi-core Mali setups	X	X
602216	Cat 2	GL ES 2: Varying arrays of mat2 type do not work	X	
602217	Cat 2	GL ES 2: Address calculation in a shader is incorrect when more than one dynamic index is used	X	
602272	Cat 2	GL ES 2: Vertex shader clamp() builtin with constant high/low values may be compiled incorrectly	X	

Linux OpenGL ES DDK package (GX910)
OpenGL ES

Date of Issue: **07-Oct-2008**

ARM Errata Notice

Document Revision **12.0**

ID	Cat	Summary of Erratum	r0p1	r0p2
602818	Cat 2	GLES2: Sampler inside struct in fragment shader and struct in vertex shader leads to segfault	X	
589267	Cat 3	GLES2: Shader linker does not do invariance checks for built-in varyings	X	X
589268	Cat 3	glDrawElements is limited to 24bit index count	X	X
589269	Cat 3	Multisample information may be lost	X	X
589270	Cat 3	Previously issued draw-calls may be lost when running out of memory	X	X
589271	Cat 3	GLES 2: gl_FragCoord upside-down when rendering to an FBO	X	
589272	Cat 3	Driver relies on mutex behaviour that is undefined	X	
589273	Cat 3	GLES 2: Calling glUniform1i on a sampler with a value larger than 7 or smaller than 0 gives segfault	X	
589868	Cat 3	Missing SW workaround for HW issue: MaliGP2 does not support normalized inputs	X	X
594532	Cat 3	GLES 1: Incomplete textures output black instead of disabling their texture-unit	X	
594727	Cat 3	GLES 2: Incorrect return values from glGetRenderbufferParameter	X	
594728	Cat 3	GLES 1: Fixed-point state getters convert enum states incorrectly	X	
594769	Cat 3	GLES 2: glGetFramebufferAttachmentParameteriv returns the wrong value for cubemap faces if they are 2D textures	X	
594867	Cat 3	GLES2: Calling glDeleteFramebuffer on the bound FBO does not cause framebuffer dependent values to change	X	
595972	Cat 3	glClear might clear only parts of the framebuffer	X	
596018	Cat 3	GLES 2: Incorrect return values from glGetIntegerv after glBindFramebuffer	X	X
596019	Cat 3	Creating EGLImages from texture mip level 10 or above results in EGL_BAD_MATCH		X
596068	Cat 3	Non-integer sized points are sometimes rendered non-square	X	X
596069	Cat 3	Modifying an EGLImage created from a pixmap within a frame can give wrong result		X
602218	Cat 3	GLES 2: Compilation of a vertex shader may fail with the error "Register allocation failed for vertex shader"	X	X
602219	Cat 3	GLES 2: Shader compiler preprocessor accepts some illegal combinations of preprocessor directives	X	X
602220	Cat 3	GLES 2: Shader compilation may fail when the client application sets the locale to non-US	X	X
602370	Cat 3	GLES 2: Failing to write gl_Position is reported as a shader compilation error	X	X
602374	Cat 3	GLES 2: Shader compiler extension macros only defined for enabled extensions	X	X
602375	Cat 3	GLES 2: Function overloading based on array sizes not supported	X	X

Linux OpenGL ES DDK package (GX910)
OpenGL ES

Date of Issue: **07-Oct-2008**

ARM Errata Notice

Document Revision **12.0**

ID	Cat	Summary of Erratum	r0p1	r0p2
602967	Cat 3	glReadPixels does not work with direct rendering	X	

Errata - Category 1

594535: GLES 1: Paletted textures with mipmaps does not work

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 1, Present in: r0p1, Fixed in r0p2.

Description

When specifying a paletted texture image with mipmap levels, the driver will read texels from outside the supplied memory. This can lead to an access violation if the MMU-pages following the texture map are not allocated.

Implications

Potential segfault when loading paletted textures with mipmaps and rendering errors when reading levels above 0.

Workaround

Unpack paletted textures to non-paletted textures before submitting them using `glTexImage2D`.

595316: GLES 2: Running out of memory in glRenderbufferStorage can cause a segfault

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 1, Present in: r0p1, Fixed in r0p2.

Description

A failing memory-allocation inside glRenderbufferStorage can cause a segfault in the following calls to glDrawElements, glDrawArrays or glClear.

Implications

Segfault when drawing.

Workaround

none

595678: Running out of memory in glReadPixels can cause a segfault

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 1, Present in: r0p1, Fixed in r0p2.

Description

Running out of memory during glReadPixels can lead to segfault in the driver.

Implications

Segfault

Workaround

none

Errata - Category 2

580918: GLES 2: Mixing CPU and GPU-based texture updates gives incorrect results

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, r0p2, Open.

Description

Mixing different methods of updating a texture gives incorrect result. The different methods are:

- OpenGL ES texture update functions: `glTexImage2D`, `glCopyTexImage2D`, `glTexSubImage2D`, and `glCopyTexSubImage2D`.
- Render to texture through Framebuffer Objects (FBOs).
- Render to texture through `EGLImage`.
- Updating pixmaps that GLES has created textures from that are bound to a framebuffer object. (`EGLImage`)
- Updating pixmaps that GLES has created renderbuffers from. (`EGLImage`)

If an application mixes update methods for the same texture, some of the modifications may be lost.

Implications

Some of the texture modifications are lost.

Workaround

Perform all modifications to the contents of a texture using only one method.

589866: GLES 2: glBinaryShader does not specify a standalone format

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

The only way to load a binary shader is through the GL_PLATFORM_BINARY format. This is illegal according to OpenGL ES 2.0 specification, section 2.15.2:

"Since OpenGL ES provides no specific binary formats, using the generic i.e. PLATFORM_BINARY format will result in a GL_INVALID_ENUM error. For all other binary formats, the binary image will be decoded according to the specification defining the binaryformat token."

Implications

No way for applications to load a binary shader.

Workaround

The driver allows the GL_PLATFORM_BINARY format to be used even though it is illegal according to the OpenGL ES 2.0 specification.

589867: GLES 2: Missing SW workaround for HW issue: Unaligned input data to vertex shader may corrupt previous input stream

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1,r0p2, Open.

Description

MaliGP2 might output corrupt data if vertex shader input data is crossing 128 bit boundaries. GX525 HW errata, defect 475765: Vertex shader data crossing 128 bit boundaries may corrupt other data, states:

"The number of usable input data stream may be reduced to any number between 16 and 8, depending on the exact data formats and alignments used. Some APIs may query the number of available streams without specifying the stream parameters, in which case only the worst case of 8 streams may be exposed.

Alternatively, if padding the input streams to avoid the line crossing issue, performance will be affected. The padding will increase the bandwidth required to read the data stream, and the software driver may need to copy the data in order to insert padding."

No software workaround for this issue has been implemented.

Implications

Incorrect rendering.

Workaround

The issue will only occur when input data streams are not aligned to 128 bit boundaries. This will only happen for 3-component and packed input streams. A workaround is therefore to use only non-packed 1, 2, or 4 component input data streams.

594530: glSampleCoverage invert flag has wrong effect

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

If glSampleCoverage has been called with the invert-parameter as GL_TRUE, the sample mask does not get inverted as expected. Instead, if GL_ALPHA_TO_COVERAGE is enabled, the coverage generated from the fragment alpha gets inverted.

Implications

Wrong coverage value will be used in rendering.

Workaround

A possible workaround for cases where the fragment alpha is not needed is to enable GL_ALPHA_TO_COVERAGE and pass the coverage value through the fragment pipeline as the fragment-alpha. For OpenGL ES 2.0, this means to simply assign the coverage value to gl_FragColor.a from a uniform variable at the end of the fragment shader. For OpenGL ES 1.x, this requires setting up a separate texture-stage and using the constant colors alpha as a source.

For cases where alpha blending or alpha testing is enabled, or there are no available texture stages, there is no known workaround.

594770: Many flushing operations without eglSwapBuffers or glClear can give incorrect results

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

If more than 100 GL calls that flush the pipeline (glReadPixels, glFlush and glFinish) are interleaved with drawing calls (this can be any combination of glDrawElements, glDrawArrays, and glClear without all buffers bits or with scissoring enabled) are issued without any calls to glClear (with all buffer bits set and scissoring disabled) or eglSwapBuffers in between, random rendering artifacts can appear. This is due to hardware errata 524463, and applies only to drivers built for Mali-200 r0p1 hardware.

Implications

Random rendering errors.

Workaround

none

594869: GLES 2: glBindFramebuffer disables multisampling until the next call to eglSwapBuffers

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

Calling glBindFramebuffer will cause multisampling to be disabled for the default framebuffer for all consecutive rendering until the next call to eglSwapBuffers.

Implications

Multisampling is disabled.

Workaround

none

594871: GLES 2: Matrix varyings do not work

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

Reading a matrix varying ("varying mat2", "varying mat3", or "varying mat4") from a fragment shader does not work correctly, only the first row of the matrix will be interpolated over the polygon.

Implications

Wrong values read from varying variables in the fragment shader.

Workaround

Use an array of vectors.

varying mat3x3 matrix;

becomes:

varying vec3 matrix[3];

**595318: GLES 2: Cubemap faces of different formats cause false
GL_OUT_OF_MEMORY**

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

Uploading cubemap faces with different formats can cause an GL_OUT_OF_MEMORY error to be set on the consecutive draw-calls using it.

Implications

Calls to glDrawElements or glDrawArrays ignored by the driver, GL_OUT_OF_MEMORY reported by glGetError.

Workaround

Upload all cubemap faces in the same format.

595623: GLES 2: Updating an FBO-bound texture with GL texture functions can lead to segfault

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

Updating a texture that is bound to a color attachment in an FBO can leave the FBO in a corrupted state. This can lead to crashes on subsequent FBO operations, including rendering (glDrawElements, glDrawArrays, or glClear) and glReadPixels.

Updates to textures can be done by issuing calls to either glTexImage2D, glTexSubImage2D, glCopyTexImage2D, glCopySubTexImage2D, glCompressedTexImage2D or glCompressedTexSubImage2D, and all of these functions can cause the issue.

Implications

Segfault.

Workaround

Detach texture from framebuffer object before modifying texture and re-attach afterwards.

595716: GLES 2: Vertex shaders that write to `gl_PointSize` may end up having other varyings corrupted

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

When writing to `gl_PointSize` from a vertex shader, unrelated float, vec2, or vec3 varyings in the same shader program might be corrupted.

Implications

Wrong values interpolated over the primitive in the fragment shader.

Workaround

Only use vec4 varyings when writing to `gl_PointSize`.

595766: GLES 1: Points with sizes set with glPointSize are rendered too small in 16xAA

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

When rendered in 16xAA mode, points with sizes set by glPointSize() gets drawn with a size that is 1/4th as big as specified.

Implications

Points rendered too small in 16xAA.

Workaround

Use point-size streams for all point sizes.

595970: GLES 1: Enabling point sprites when rendering polygons or lines breaks texturing

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

If GL_POINT_SPRITE_OES is enabled when drawing non-point primitives, all texture stages that have the GL_COORD_REPLACE_OES state enabled will get their texture coordinates discarded, and have some other implicit coordinates applied instead. The implicit coordinates does not make any sense for the purpose of texturing polygons or lines.

Implications

The supplied texture coordinates are not used, and the texturing result will look wrong.

Workaround

Disable GL_POINT_SPRITE_OES by calling glDisable(GL_POINT_SPRITE_OES) before drawing non-point primitives, and re-enable it again by calling glEnable(GL_POINT_SPRITE_OES) afterwards.

596020: GLES 2: Rendering to FBOs and using the image as a texture can fail on multi-core Mali setups

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1,r0p2, Open.

Description

On multi-core Mali configurations, rendering to an FBO with a texture attachment and subsequently using the same texture in draw-calls can cause an intermediate result to be used for texturing.

Implications

A mixture of old and new texture content is returned from the texture-sampler.

Workaround

Call glFinish before calling glBindFramebuffer when switching from the problematic framebuffer object.

602216: GLES 2: Varying arrays of mat2 type do not work

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

The shader compiler uses an incorrect array stride for mat2 varying arrays, which causes the second column of a matrix in the array to overlap with the first column of the next matrix element in the array.

Implications

The second column of the mat2 elements will contain incorrect data.

Workaround

Use a varying array of vec2s or vec4s instead.

602217: GLES 2: Address calculation in a shader is incorrect when more than one dynamic index is used

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

When there is more than one dynamic index (indexing with a non-constant expression) in a row, the address calculation ends up multiplying together the address contributions instead of adding them. This gives incorrect results for most indices.

The following counts as dynamic indexing:

- Array indexing
- Matrix column indexing (apart from attribute matrices)

The following does not count as dynamic indexing:

- Indexing of individual elements in a vector
- Attribute matrix column indexing

Implications

Indexing with more than one dynamic index gives incorrect results.

Workaround

Do the indexing one step at a time. For instance, if you are indexing an array of matrices you can replace:

```
uniform mat2 array[2];
```

```
...
```

```
vec2 res = array[i][j];
```

with:

```
uniform mat2 array[2];
```

```
...
```

```
mat2 tmp = array[i];
```

```
vec2 res = tmp[j];
```

For stores, a similar workaround using read-modify-write can be employed:

```
mat2 array[2];
```

```
...
```

```
array[i][j] = vec2(1.0, 2.0);
```

can be replaced with:

```
mat2 array[2];
```

```
...
```

```
mat2 tmp = array[i];
```

```
tmp[j] = vec2(1.0, 2.0);
```

```
array[i] = tmp;
```

For cases with arrays inside structs, no workarounds exist.

602272: GLES 2: Vertex shader clamp() builtin with constant high/low values may be compiled incorrectly

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

If a vertex shader uses clamp(v, min_val, max_val) where min_val and max_val are constants, and the values given for min_val and max_val are both in use in other parts of the shader, the compiler can in rare cases change the values used for min_val and max_val.

Implications

If the shader uses clamp() with constant clamps, the value might be clamped to something else. As the compiler inserts clamp(v, -1.0e10, 1.0e10) for the perspective transform, shaders that use both -1.0e10 and 1.0e10 may have the vertex positions changed.

Workaround

Replace uses of clamp(v, min_val, max_val) with min(max(v, min_val), max_val), and replace any uses of the constants -1.0e10 and 1.0e10 with values loaded from uniforms.

602818: GLES2: Sampler inside struct in fragment shader and struct in vertex shader leads to segfault

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 2, Present in: r0p1, Fixed in r0p2.

Description

Draw-calls using a program that has a struct in both the vertex and fragment shader, and a sampler inside the fragment-shader struct give a segfault.

Implications

Segfault when calling glDrawElements or glDrawArrays.

Workaround

none

Errata - Category 3

589267: GLES2: Shader linker does not do invariance checks for built-in varyings

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

Section 4.6.4 of the OpenGL ES 2.0 specification, Invariance and Linkage, requires invariance checking between the built-in shader varyings `gl_FragCoord`, `gl_Position`, `gl_PointCoord`, `gl_PointSize`, and `gl_FrontFacing`. This is not being done by the linker. Instead, all varyings are being treated as if they were invariant.

Implications

This defect has no implications for valid programs.

Workaround

Do not declare any of the built-in varyings invariant.

589268: glDrawElements is limited to 24bit index count

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

glDrawElements does not support an index count larger than $2^{24}-1$.

Implications

The index count in glDrawElements will be limited to $2^{24}-1$, meaning that indices will be discarded if the count is above this value.

Workaround

If an index count of 2^{24} or more is needed, the call to glDrawElements should be split into several calls with index count less than 2^{24} .

589269: Multisample information may be lost

Status

Affects: product Linux OpenGL ES DDK package.
Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

Calling glDrawElements or glDrawArrays followed by a glReadPixels, glFlush or glFinish more than 100 times, without calling glClear, will cause the driver to lose multisample information.

Implications

Multisample buffers will not be preserved when glDrawElements or glDrawArrays followed by a glReadPixels, glFlush or glFinish are called more than 100 times in succession.

Workaround

If glDrawElements or glDrawArrays followed by a glReadPixels, glFlush or glFinish is called more than hundred times a glClear must be called in between to make sure that multisample buffers are preserved.

589270: Previously issued draw-calls may be lost when running out of memory

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

If more than 100 GL calls that flush the pipeline (glReadPixels, glFlush and glFinish) are interleaved with drawing calls (this can be any combination of glDrawElements, glDrawArrays, and glClear without all buffers bits or with scissoring enabled) are issued without any calls to glClear (with all buffer bits set and scissoring disabled) or eglSwapBuffers in between, and the driver runs out of memory, the result of previously issued drawing calls can be lost.

Implications

Previously rendered content is lost.

Workaround

Avoid excessive calls to glReadPixels, and avoid all calls to glFlush. glFlush is called implicitly at eglSwapBuffers.

589271: GLES 2: gl_FragCoord upside-down when rendering to an FBO

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

Rendering to an FBO is implemented, among other things, by flipping the viewport upside-down. This means that `gl_FragCoord.y` should be also be flipped. This is not done, which causes `gl_FragCoord.y` to have an incorrect value.

Implications

`gl_FragCoord.y` will be incorrect when rendering to an FBO.

Workaround

Manually flip `gl_FragCoord.y` in the shader: `correct_value = HEIGHT - gl_FragCoord.y`, where `HEIGHT` is the height of the render target.

589272: Driver relies on mutex behaviour that is undefined

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

The driver relies on mutexes that can be unlocked from a different thread of execution than the thread that locked it. This behavior is undefined according to the pthread documentation:

"Attempting to unlock the mutex if it was not locked by the calling thread results in undefined behaviour.

Attempting to unlock the mutex if it is not locked results in undefined behaviour."

See http://opengroup.org/onlinepubs/007908775/xsh/pthread_mutex_lock.html for more information.

Implications

The undefined behaviour affects the portability of the driver, because other mutex implementation can have different behaviour.

Workaround

None

589273: GLES 2: Calling glUniform1i on a sampler with a value larger than 7 or smaller than 0 gives segfault

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

Calling glUniform1i on a sampler with value larger than 7 or smaller than 0, triggers an assertion in debug builds of the driver during the following draw call. In release mode this gives segfault.

Implications

An assertion will be triggered in debug builds and a crash occurs in release builds.

Workaround

Avoid calling glUniform1i with incorrect values for samplers.

589868: Missing SW workaround for HW issue: MaliGP2 does not support normalized inputs

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, r0p2, Open.

Description

The vertex loader in MaliGP2 (r0p4 and earlier) does not support normalized inputs. GX525 HW errata, defect 509016: MaliGP2 does not support normalized inputs, states:

"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."

No software workaround has been implemented for this issue.

Implications

Normalized vertex attribute data is divided by the maximum value of the data-type plus one, and not the maximum value as the standard specifies. This means that for normalized vertex attributes, all values will be slightly smaller than they should, and it is impossible to represent the exact value of 1.0.

Workaround

Do not use GL_UNSIGNED_BYTE on normalized vertex attribute data, or normalize data before use. In OpenGL ES 2.0, data can be normalized in the vertex shader by adding this to the shader:

```
attribute float byte_attrib;

// your code

void main()
{
    // your code
    float corrected_value;
    corrected_value = (byte_attrib * 256)/255;
}
```

594532: GLES 1: Incomplete textures output black instead of disabling their texture-unit

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

When a texture stage has an incomplete texture enabled, the texture samples a constant black color instead of disabling the texture unit.

Implications

Incorrect rendering.

Workaround

Disable incomplete texture units.

594727: GLES 2: Incorrect return values from glGetRenderbufferParameter

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

glGetRenderbufferParameter returns the component sizes requested rather than the ones actually obtained for GL_RENDERBUFFER_RED_SIZE, GL_RENDERBUFFER_GREEN_SIZE, GL_RENDERBUFFER_BLUE_SIZE, GL_RENDERBUFFER_ALPHA_SIZE, GL_RENDERBUFFER_DEPTH_SIZE, and GL_RENDERBUFFER_STENCIL_SIZE.

Implications

Wrong value returned.

Workaround

none

594728: GLES 1: Fixed-point state getters convert enum states incorrectly

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

Fixed-point getters (glGetTexEnvv() etc.) converts enum state incorrectly to fixed-point by multiplying them by 65536 as if they were integers.

Implications

Wrong value returned by fixed-point getter.

Workaround

Divide the returned value by 65536.

594769: GLES 2: glGetFramebufferAttachmentParameteriv returns the wrong value for cubemap faces if they are 2D textures

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

glGetFramebufferAttachmentParameteriv returns GL_TEXTURE_2D when trying to query the GL_FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE state from a 2D texture attachment, when it should have returned 0.

Implications

Wrong value returned by glGetFramebufferAttachmentParameteriv.

Workaround

none

594867: GLES2: Calling glDeleteFramebuffer on the bound FBO does not cause framebuffer dependent values to change

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

Calling glDeleteFramebuffer on the currently bound framebuffer object cause the default framebuffer object to be bound, but does not update the framebuffer dependent values (GL_RED_BITS, GL_STENCIL_BITS, etc).

Implications

Wrong value returned by glGetIntegerv.

Workaround

none

595972: glClear might clear only parts of the framebuffer

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

glClear can, in the cases where scissor testing is used or not all buffers bits are set, end up clearing only parts of the expected area.

Implications

Incorrect rendering.

Workaround

Always clear the entire screen (without scissor testing, and with all buffer bits set). If this cannot be done, clear by drawing a quad that covers the entire screen. Depth and stencil compares must be set to GL_ALWAYS, and depth and stencil writes should only be enabled if those buffers are to be cleared.

596018: GLES 2: Incorrect return values from glGetIntegerv after glBindFramebuffer

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

If glGetInteger is called with the pname parameter GL_RED_BITS, GL_GREEN_BITS, GL_BLUE_BITS, or GL_ALPHA_BITS and no draw-calls have been made since the previous call to glBindFramebuffer, the value 0 is incorrectly returned instead of the expected value.

Implications

Wrong value returned.

Workaround

Draw a single, off-screen point between the call to glBindFramebuffer and the call to glGetIntegerv.

596019: Creating EGLImages from texture mip level 10 or above results in EGL_BAD_MATCH

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p2, Open.

Description

Creating an EGLImage from an OpenGL ES texture with a mip level of 10 or above will give an EGL_BAD_MATCH error instead of working as expected. This is due to a driver-limitation.

Implications

Mip levels 10 or above cannot be shared through EGLImages.

Workaround

none

596068: Non-integer sized points are sometimes rendered non-square

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

Under some conditions, point with non-integer size can be rasterized non-square by Mali-200, and this is not worked around by the OpenGL ES driver.

Implications

Some points rasterized as non-square.

Workaround

Round the point-size to an integer in the vertex-shader in OpenGL ES 2.0.

No workaround is known for OpenGL ES 1.x.

596069: Modifying an EGLImage created from a pixmap within a frame can give wrong result

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p2, Open.

Description

If a texture is created from an EGLImage that was created from a pixmap and that texture is used for rendering and then modified in the same frame, the rendering will use the modified texture instead of the original.

Implications

The modified texture is used for draw-calls that happened before the modification took place.

Workaround

none

602218: GLES 2: Compilation of a vertex shader may fail with the error "Register allocation failed for vertex shader"

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

MaliGP2 has only limited internal bandwidth between its registers and execution units. In some rare cases, the register allocator for MaliGP2 in the shader compiler runs into a situation where there are more operations executed in one cycle than can be fed from the registers simultaneously. The compiler aborts the compilation in this case.

Implications

Some very big or very complex vertex shaders fail to compile.

Workaround

Changing the shader code slightly will often eliminate the problem. Try to rewrite some of the shader to do the calculations differently.

602219: GLES 2: Shader compiler preprocessor accepts some illegal combinations of preprocessor directives

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

The ESSL compiler may in some cases accept shaders containing illegal combinations of `#if` / `#ifdef` / `#ifndef` / `#elif` / `#else` / `#endif` as valid.

Implications

This defect has no implications for valid programs.

Workaround

No workaround needed.

602220: GLES 2: Shader compilation may fail when the client application sets the locale to non-US

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

The ESSL compiler uses strtod() for parsing floating point literals. Therefore, if the online compiler is used with an OpenGL ES application and the application either sets the locale explicitly to a locale with different numerical conventions than the US with e.g.

```
setlocale(LC_NUMERIC, "nb_NO");
```

or picks up the current locale from environment variables with e.g.

```
setlocale(LC_NUMERIC, "");
```

and the environment variables are set in a way that indicates a locale with a different numerical convention than the US, the parser will fail with errors such as

```
0:28: L0001: Error while parsing floating point literal '0.0'
```

This is because the strtod() function now is looking for literals with a different decimal separator.

Implications

Online compilation of valid shaders using floating-point literals may fail.

Workaround

Either use the off-line compiler and load shader binaries, or surround each call to glCompileShader with setlocale calls, e.g.

```
char *prev_locale = setlocale(LC_NUMERIC, "C"); /* reset numeric locale to default,
```

```
save the old locale */
```

```
glCompileShader(...);
```

```
setlocale(LC_NUMERIC, prev_locale); /* restore the old locale */
```


602370: GLES 2: Failing to write gl_Position is reported as a shader compilation error

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

If the `gl_Position` built-in variable is not written anywhere in a vertex shader, the compiler reports an error. This error should only be a warning, since this (useless) situation is not explicitly forbidden by the language specification.

Implications

No implications for any useful shader code.

Workaround

No workaround needed.

602374: GLES 2: Shader compiler extension macros only defined for enabled extensions

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1,r0p2, Open.

Description

For each language extension supported by an implementation, a macro with the extension name is automatically defined by the compiler to indicate that the extension is supported. However, the compiler erroneously defines these macros only for extensions that are explicitly enabled.

Implications

Only code that is written to compile on several different implementations supporting different extensions are affected. In such code, it is not possible to explicitly test for support of an extension and conditionally compile different code depending on whether it is supported or not.

Workaround

Unconditionally enable all extensions attempted used in the shader program.

602375: GLES 2: Function overloading based on array sizes not supported

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, r0p2, Open.

Description

The ESSL language allows functions to be overloaded based on array sizes in their parameter types, i.e. one can have two functions with identical names and identical parameter types except for the sizes of array types. The compiler treats functions with same parameter types except for array sizes as having the same signature.

Implications

Shaders containing functions with identical names and identical parameter names except for array sizes are rejected by the compiler. For instance, compilation of the program

```
vec4 foo(float a[2])
{
    return vec4(0.0, 1.0, 0.0, 0.0);
}
vec4 foo(float a[3])
{
    return vec4(1.0, 0.0, 0.0, 0.0);
}
void main()
{
    float a[2];
    gl_Position = foo(a);
}
```

will fail with the error message

0:9: S0023: Function 'foo' redefined

Workaround

Change the function signatures to use different names.

602967: glReadPixels does not work with direct rendering

Status

Affects: product Linux OpenGL ES DDK package.

Fault status: Cat 3, Present in: r0p1, Fixed in r0p2.

Description

If direct rendering is used and eglMakeCurrent has been used to change the read surface since the previous draw-call, then glReadPixels does not work correctly.

Direct rendering is a memory bandwidth optimization enabled in some builds of the driver. See the Release Notes for information on how to enable direct rendering.

Implications

Outdated framebuffer data is returned by glReadPixels.

Workaround

Issue a draw-call that does not produce any fragments before calling glReadPixels.

Errata - Documentation

There are no Errata in this Category

Errata – Driver Software

There are no Errata in this Category