Mali Offline Shader Compiler Errata

Document Number:	PR389-PRDC-011263 2.0
Date of Issue:	17th January 2012
Product:	Mali Offline Shader Compiler
Product Version:	3.0.0

Abstract

This document describes the known errata in the EAC release of the Mali Offline Shader Compiler version 3.0.0.

This is a working document throughout the product lifecycle and, as such, the content may be modified as new information is uncovered.

Release Information

Proprietary Notice

Document Confidentiality Status

ARM Web Address

The ARM website is located at the following address: http://www.arm.com

Feedback

ARM welcomes feedback on this product and its documentation.

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give the following:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms if appropriate.

Feedback on this document

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

- The document title.
- The document 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

1 PREFACE

1.1 Change Control

Issue	*Date*	*Change*
1.0	15th September 2009	Initial release.
2.0	17th January 2012	Changes for product version 3.0.0.

1.2 Scope

This document describes the errata discovered in the implementation of the Mali Offline Shader Compiler, categorized by level of severity. Each description includes:

- 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

2 CATEGORISATION OF ERRATA

Errata recorded in this document are split into three groups:

Category 1 Features which are impossible to work around and severely restricts the use of the software in all or the majority of applications rendering the software unusable.

Category 2 Features which contravene the specified behavior and may limit or severely impair the intended use of specified features but does not render the software unusable in all or the majority of applications.

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

2.1 Errata Summary

The following tables summarize all errata associated with this product.

Mali Offline Shader Compiler Errata Summary

ID	***Category***	***Summary of Erratum***
717175	Category 3	Shader compilation may fail when the client application sets the locale to non-US
717177	Category 3	Compilation of a vertex shader may fail with the error "Register allocation failed for vertex shader"

3 CATEGORY 1 ERRATA

No Category 1 errata

4 CATEGORY 2 ERRATA

No Category 2 errata

5 CATEGORY 3 ERRATA

5.1 717175: Shader compilation may fail when the client application sets the locate to non-US

Status

Affects:	Mali Offline Shader Compiler
Fault status:	Category 3, Present in: 2.2, 3.0.0
Platforms Affected:	Windows, Linux, Mac OS X

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.

```
/* reset numeric locale to default and save the old locale */
char *prev_locale = setlocale(LC_NUMERIC, "C");
glCompileShader(...);
setlocale(LC_NUMERIC, prev_locale); /* restore the old locale */.
```

5.2 717177: Compilation of a vertex shader may fail with the error "Register allocation failed for vertex shader"

Status

Affects:	Mali Offline Shader Compiler
Fault status:	Category 3, Present in: 2.2, 3.0.0
Platforms Affected:	Windows, Linux, Mac OS X

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.