Software Test Procedure

SGX

Copyright © Imagination Technologies Ltd. All Rights Reserved.

This document is confidential. Neither the whole nor any part of the information contained in, nor the product described in, this document may be adapted or reproduced in any material form except with the written permission of Imagination Technologies Ltd. All other logos, products, trademarks and registered trademarks are the property of their respective owners. This document can only be distributed subject to the terms of a Non-Disclosure Agreement or Licence with Imagination Technologies Ltd.

Filename	:	Software Test Procedure.SGX.doc
Version	:	1.0.35 External Issue ANDROID.
Issue Date	:	17 Jun 2011
Author	:	Imagination Technologies

Contents

1. Introduction

1.1. Scope

This document outlines the procedures used during SQA, to test specific features of the current PowerVR SGX driver.

1.2. Related Documentation

Document
SGX Android OMAP4 Blaze Platform Guide (1.7 DDK)
SGX DDK for Android Software Functional Specification
SGX OpenGL ES 1.1 Reference Driver Software Functional Specification
SGX OpenGL ES 2.0 Reference Driver Software Functional Specification
SGX Services 4 Software Functional Specification

1.3. Apphints and powervr.ini

The powervr.ini file and the use of ‘apphints’ allow SQA to manipulate the driver to allow completion of certain tests and features.

For Android the file is read from $DISCIMAGE/system/etc/powervr.ini or from the current directory when a binary is executed. The latter location takes precedence over the former.

2. Khronos Conformance Tests

2.1. OpenGL ES 1.1

SQA uses the current RC3 version of the OGLES1 conformance tests. This is visible on the Khronos website as ESCTS-1.1.3.0-20100210.tgz. No Khronos patches are in use at this time.

Conform is executed with the following parameters:

```
$ ./conform -v2 -p 0 -f testlist -r 32555
$ ./conform -v2 -p 1 -f testlist -r 32556
$ ./conform -v2 -p 2 -f testlist -r 32557
$ ./conform -v2 -p 3 -f testlist -r 32558
```

For covgl, covegl and primtest no command line parameters are passed.

It is necessary to use the following apphints when running conform, covgl, covegl and primtest:

```
[conform]
ExternalZBufferMode=2
[covgl]
ExternalZBufferMode=2
[covegl]
ExternalZBufferMode=2
[primtest]
ExternalZBufferMode=2
```

2.2. OpenGL ES 2.0

SQA uses the current RC3 version of the OGLES2 conformance tests. This is visible on the Khronos website as ESCTS-2.0.3.0-20091116.tgz. The following table shows the patches that have been applied to this version of the conformance test:

Item	Type of Change	Files Modified	Notes
BRN31095 Bugzilla 7681 (Android source issue)	Patch (IMG)	GTFExtensionTestEGLImage.c	Specific to Android only. Modifies the tests to call the extension entry points directly.

The following command line parameters are passed to GTF:

```
$ ./GTF -width=64 -height=64 -run=mustpass.run
$ ./GTF -width=113 -height=47 -run=mustpass.run
$ ./GTF -width=[max_width] -height=[max_height] -run=mustpass.run
```

Set [max_width] and [max_height] to the maximum renderable window size supported on the platform being tested. For the [max_width] and [max_height] resolution, android:theme="@android:style/Theme.NoTitleBar" was used.

The following apphints are used for GTF:

```
[GTF]
ExternalZBufferMode=2
```

3. PDUMP

3.1. Obtaining PDUMP Out Files

See the platform guide for details of building the DDK with PDUMP for a hardware or a no hardware target.

See section “Android-specific PDUMP Information” in the document “SGX DDK for Android.Software Functional Specification” for details of how to configure an Android system and capture PDUMP data.

This will generate three PDUMP2 out.* files. Copy these files to a separate machine running Windows XP or later, in a new directory, for example c:\pdump. The files needed are:

```
out2.txt
out2.prm
out_drvinfo.txt
```

3.2. Additional Files Needed

To convert these files into PDUMP1 out.* files, you will need the following executables. These can be found in the ‘PowerVrToolsExternalPackage1.00.10.XXXX.zip’. Copy these to c:\pdump

```
pdump2topdump1.exe
pdumppostprocess.exe
```

Other necessary files can be found in the relevant Simulator Package, applicable to your SGX core and core revision in your platform.

For SGX Blaze platform, it is ‘SGX540_V1_2_0.1.XXXX.zip’. These files can be found in the \win32simulator and \win32simulator\tools directories. Copy to c:\pdump. They are:

```
SGX540_V1_2_0_mt.exe
pthreadVC.dll
convert.bat
sgx535cnf.txt
```

3.3. Converting PDUMP2 In to PDUMP1

Open a CMD prompt and navigate to the folder directory where you have placed all the necessary files (c:\pdump). Execute the convert.bat with the following params. This will automatically convert your PDUMP2 out files into PDUMP1 out files.

```
convert.bat sgx540
```

3.4. Simulation

After converting you can then run the simulator. This will generate a params directory and a series of bitmap images showing the application being rendered.

```
SGX540_V1_2_0_mt.exe
```

4. SPM

To induce Smart Parameter Management (SPM) and create an ‘OUTOFMEMORY’ event, the buffer size is reduced to the driver default of 10 pages, through the use of an apphint:

```
[name_of_app]
ParamBufferSize=40960
```

For multi-application SPM testing, a powervr.ini file will be created with several entries, with each application being assigned a particular buffer size:

```
[name_of_app]
ParamBufferSize=40960
[name_of_app_2]
ParamBufferSize=81920 /* 20 pages */
[name_of_app_3]
ParamBufferSize=122880 /* 30 pages */
```

For DDK branches 1.6 or higher an additional apphint is needed to overcome the ‘Buffer Grow/Shrink’ driver feature and forcefully restrict the buffer from increasing. The apphint is below:

```
[name_of_app]
MaxParamBufferSize=40960
```

Alternatively the default keyword can be used to restrict all application buffer sizes for the platform, but this is not advised:

```
[default]
ParamBufferSize=40960
[default]
MaxParamBufferSize=40960
```