SGX DDK for Android

Software Architecture Specification

Filename	:	SGX DDK for Android.Software Architecture Specification.doc
Version	:	1.0.9 External Issue
Issue Date	:	15 Jul 2010
Author	:	Imagination Technologies

Contents

List of Figures

Glossary of Terms

Khronos Group	Body that controls the OpenGL-ES, OpenVG and OpengMAX specifications
EGL 1.4	The Native Platform Graphics Interface implemented for this project.
OpenGL-ES™ 2.0	The programmable 3D API implemented for this project OpenGL-ES is a trademark of Silicon Graphics, Inc.
OpenGL-ES™ 1.1	The fixed function 3D API implemented for this project OpenGL-ES is a trademark of Silicon Graphics, Inc.
SGX	IP that provides 2D/3D/Video acceleration

1. Introduction

1.1. Scope

This document provides a top-level view of the SGX Embedded Systems (ES) DDK software architecture for Android (2.0.1 or greater) with a brief description of the major architectural components. It is assumed that the reader is already familiar with the hardware functionality of SGX and its derivatives.

1.2. Related Documents

OpenGL-ES 1.1 Specification, Khronos Group
OpenGL-ES 2.0 Specification, Khronos Group
EGL 1.4 Specification, Khronos Group
SGX OpenGL ES 2.0 Reference Driver, Software Architecture Specification
SGX OpenGL ES 1.1 Reference Driver, Software Architecture Specification
SGX EGL 1.4 Reference Driver, Software Architecture Specification
PVR2D Reference Driver, Software Architecture Specification
Consumer Services Reference Driver, Software Architecture Specification
Pluggable Window System.Implementation Guide
Pluggable Window System.API
Consumer Services.3rd Party Buffer API
Consumer Services.3rd Party Display API
PVR2D API Specification

2. Product Overview

The SGX embedded systems DDK for Android implements the graphics hardware acceleration infrastructure required to support hardware accelerated rendering and composition under the Android (2.0.1 or greater) operating system.

2.1.1. Goals and Objectives

The major high-level design goals for the SGX ES-DDK for Android software architecture are detailed below.

- Low Implementation Risk. This is achieved by substantial reuse of major IP code blocks, specifically OpenGL-ES 1.1, OpenGL-ES 2.0, Services (including uKernel).
- Does not impede performance. This is achieved by utilising the optimal composition code path available in the Android (2.0.1 or greater) architecture, to allow zero copy HW composition
- Support HW accelerated composition of the Android native windowing system with OpenGL-ES 1.1, OpenGL-ES 2.0 applications through the EGL API.
- Allow integration of a 3rd party display controller to allow high-performance “flipping” for composition presentation.

2.2. Product Environment

The SGX Embedded Systems DDK for Android allows targeting against all supported SGX cores at the associated major DDK revision. The reference implementation has been tested as running against exiting HW including but not limited to SGX530 v1.2.1 & SGX535 v1.1.2. The reference implementation has been built against Intel IA32 CPUs and ARM Cortex CPUs.

3. Architecture Specification

3.1. Overview

The following diagram Figure 1 Architecture Overview details a top-level overview of the SGX ES-DDK for Android architecture.

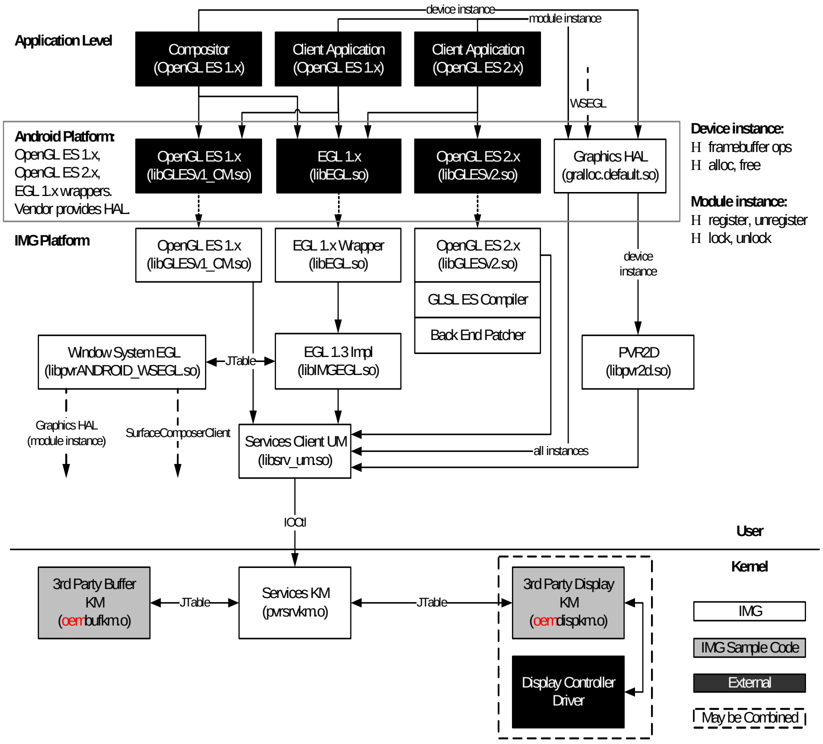
device instance
module instance
Application Level
Compositor (OpenGL ES 1.x)
Client Application (OpenGL ES 1.x)
Client Application (OpenGL ES 2.x)
WSEGL
Android Platform:
OpenGL ES 1.x,
OpenGL ES 2.x,
EGL 1.x wrappers.
Vendor provides HAL.
OpenGL ES 1.x (libGLESv1_CM.so)
EGL 1.x (libEGL.so)
OpenGL ES 2.x (libGLESV2.so)
Graphics HAL (gralloc.default.so)
Device instance:
H framebuffer ops
H alloc, free
Module instance:
H register, unregister
H lock, unlock
IMG Platform
OpenGL ES 1.x (libGLESv1_CM.so)
EGL 1.x Wrapper (libEGL.so)
OpenGL ES 2.x (libGLESV2.so)
GLSL ES Compiler
Back End Patcher
device instance
Window System EGL (libpvrANDROID_WSEGL.so)
JTable
EGL 1.3 Impl (libIMGEGL.so)
PVR2D (libpvr2d.so)
Graphics HAL (module instance)
SurfaceComposerClient
Services Client UM (libsrv_um.so)
all instances
IOCtl
User
Kernel
3rd Party Buffer KM (oembufkm.o)
JTable
Services KM (pvrsrvkm.o)
JTable
3rd Party Display KM (oemdispkm.o)
Display Controller Driver
IMG
IMG Sample Code
External
May be Combined

Figure 1 Architecture Overview

4. Detailed Component Description

OpenGL-ES 1.1	
Component Type	Shared library (.so)
Purpose	Provides the OpenGL-ES 1.1 Common and Common-lite API interfaces to SGX hardware and native platform graphics interface.
Functionality	Provides 3D acceleration of OpenGL ES 1.1 Common and Common-lite. (See “OpenGL ES 1.1 Reference Driver, Software Architecture Specification” for details).
Exported Interfaces	This component provides a ‘C’ interface as defined by the OpenGL ES 1.1 Common Lite profile, Common profile and Native Platform Graphics Interface as defined by the Khronos group.
Imported Interfaces	None
Dependencies and Inter-relationships with other major components	This component interfaces with the EGL implementation component and uses the PowerVR services component. The PowerVR services component is used for memory management, power management, synchronisation, device initialisation and various common utility functions.

OpenGL-ES 2.0	
Component Type	Shared library (.so)
Purpose	Provides the OpenGL ES 2.0 API interface to SGX.
Functionality	Provides 3D acceleration of OpenGL ES 2.0. (See “OpenGL ES 2.0 Reference Driver, Software Architecture Specification” for details).
Exported Interfaces	This component provides a ‘C’ interface as defined by the OpenGL ES 2.0 API as defined by the Khronos group.
Imported Interfaces	None
Dependencies and Inter-relationships with other major components	This component interfaces with the EGL implementation component and uses the PowerVR services component. The PowerVR services component is used for memory management, power management, synchronisation, device initialisation and and various common utility functions.

GLSL ES Compiler	
Component Type	Shared library (.so) / Binary Executable
Purpose	Provides a way to compile GLSL ES source shaders for OpenGL ES 2.0.
Functionality	Compiles GLSL ES source shaders to a proprietary intermediate format. (See “OpenGL ES 2.0 Reference Driver, Software Architecture Specification” for details).
Exported Interfaces	This component may be a binary executable which will execute on a source file input containing the shader and will output a binary output file for use by an OpenGL ES 2.0 application.
Imported Interfaces	None
Dependencies and Inter-relationships with other major components	This component will be both a binary executable which will execute on the host system (not the target) to convert a source file containing a GLSL ES shader to a binary output file for use by an OpenGL ES 2.0 application; and a library for the OpenGL ES 2.0 driver component to support compilation of source shaders directly by the driver.

EGL	
Component Type	Shared Libraries (.so) - Thin Layer and Implementation Layer.
Purpose	Provides the native platform graphics interface (EGL1.2) as well as the implementation of all revisions of the native platform graphics interface.
Functionality	EGL provides mechanisms for creating rendering surfaces onto which OpenGL ES 1.1, OpenGL ES 2.0 and OpenVG can draw, and synchronizing drawing between these client APIs and the native platform. (See “EGL 1.x Reference Driver, Software Architecture Specification” for details).
Exported Interfaces	The thin layer provides a ‘C’ interface as defined by the Native Platform Graphics Interface as defined by the Khronos group.
Imported Interfaces	The thin layer interfaces with the implementation layer

EGL	
Dependencies and Inter-relationships with other major components	The implementation layer interfaces with all client APIs, and also with the Pluggable Window System component provided by the OEM. It also uses the PowerVR services component. The PowerVR services component is used for memory management, power management, synchronisation, device initialisation and various common utility functions.

Graphics HAL	
Component Type	Shared library (.so)
Purpose	Provides an Android graphics API interface to the PowerVR Services component
Functionality	This component implements the types and functions described in the Android driver model documentation.
Exported Interfaces	API defined in the Android EGL Driver Model document
Imported Interfaces	None
Dependencies and Inter-relationships with other major components	This component interfaces with the EGL implementation component and uses the PowerVR services component.

PVR2D	
Component Type	Shared library (.so)
Purpose	Hardware flipping support and surface abstraction.
Functionality	Accelerates operations for the Pluggable Window System component and Surface Flinger. Provides a method of surface allocation and of making previously allocated surfaces visible to the SGX HW. See “PVR2D Reference Driver, Software Architecture Specification” for details).
Exported Interfaces	The PVR2D API provides a ‘C’ interface as defined in the “PVR2D API specification” document.
Imported Interfaces	None
Dependencies and Inter-relationships with other major components	The PowerVR services component is used for memory management, power management, synchronisation, device initialisation and various common utility functions. The 3rd Party Display Class Driver may be used indirectly (through the services) for flipping support.

PowerVR Services	
Component Type	Kernel and user space libraries (.o, .so)
Purpose	The primary aim of this component is to provide a degree of insulation from platform specific detail and manage shared resources, and provide multi-context and multi-process synchronisation without undue software serialisation.

PowerVR Services	
Functionality	Services are provided to initialise the device, to manage memory allocations for all device addressable memory, to synchronise access to shared resources, to synchronise activities on different HW modules, provide power management, resource management, and initialise the device. (See “Consumer Services Reference Driver, Software Architecture Specification” for details).
Exported Interfaces	API defined in the PowerVR Services documentation
Imported Interfaces	The user mode services component interfaces with the kernel mode component through ioctls.
Dependencies and Inter-relationships with other major components	The user and kernel mode services components can interface to 3rd party display/buffer class drivers. (See “3rd Party Buffer API Specification” and “3rd Party Display API Specification” documents for details).

3rd Party Pluggable Window System (OPK)	
Component Type	Shared library (.so)
Purpose	The primary aim of this component is to allow a DDK licensee/OEM to interface a custom windowing system with EGL.
Functionality	The component is responsible for display validation and capability reporting, surface creation/mapping to SGX HW, as well as surface presentation. Example SW will be provided for the “Null” window system and the X Window System. (See “Pluggable Window System.Implementation Guide” for details).
Exported Interfaces	API defined in the “Pluggable Window System.API” document.
Imported Interfaces	
Dependencies and Inter-relationships with other major components	The 3rd Part Pluggable Window System component can interface to the PVR2D module for surface creation, mapping, 2D acceleration and potentially surface flipping.

3rd Party Buffer Class	
Component Type	Kernel and user space libraries (.o, .so)
Purpose	The primary aim of this component is to provide an abstraction of buffer class (aka stream) devices for PowerVR services.
Functionality	This component implements functionality to allow surface synchronisation between SGX and data sourced from 3rd Party stream HW/SW.
Exported Interfaces	API defined in the “Consumer Services.3rd Party Buffer API” document
Imported Interfaces	The user mode services component can interface with the kernel mode component through ioctls.
Dependencies and Inter-relationships with other major components	The 3rd party buffer class driver interfaces with the Services component.

3rd Party Display Class	
Component Type	Kernel and user space libraries (.o, .so)

3rd Party Display Class	
Purpose	The primary aim of this component is to provide an abstraction of display devices for PowerVR services.
Functionality	This component exposes display controller functionality to services clients, and synchronisation of operation between SGX and a 3rd Party Display Controller.
Exported Interfaces	API defined in the “Consumer Services.3rd Party Display API” document
Imported Interfaces	The user mode services component can interface with the kernel mode component through ioctls.
Dependencies and Inter-relationships with other major components	The 3rd party display class driver interfaces with the Services component, and the kernel component may interface to (or in fact be subsumed within) a display controller driver.