



2D Rendering on MIPS® Android

Document Number: MD00809

Revision 01.00

February 3, 2011

This application note refers to Android 2.2 "Froyo" code-base.

Android offers couple of points where 2D graphic rendering can be accelerated using platform specific hardware acceleration provided by destination platform. These points are present in:

- OpenGL module
- Skia module

Android relies on OpenGL ES standard for 3D graphic operations and some basic 2D graphic operations (like blitting two surfaces). This standard is wrapped with two libraries: `libagl` and `libhgl`. These libraries have the same API but implement it in different way. `libagl` is pure software implementation that relies on the CPU (MIPS) for actual processing. `libhgl` is hardware accelerated library that uses GA (graphic accelerator) for acceleration of the 2D/3D graphic operations. Depending on what is present in the system, Android will use either `libagl` or `libhgl`. MIPS Android source tree doesn't have `libhgl` source code, as this library is platform specific and relies on the fact that platform must include GA that supports OpenGL ES standard. `libagl`, on the other hand is implemented and includes MIPS code generation for API functions. In order to truly accelerate 3D (and some basic 2D) rendering, `libhgl` implementation is mandatory. As API for this library is not available, usually GA manufacturers provide their own `libhgl` implementation.

There is one way to accelerate blitting operations (for whole surfaces, especially used during surface composition process in SurfaceFlinger) without implementing `libhgl`. This is possible by implementing library named: `libcopybit`. This library (if present in system) will substitute blitting and stretch blitting operations for surfaces with platform specific implementations (like DirectFB, etc). Library only implements two functions:

- `blit_copybit`
- `stretch_copybit`

Beside `libcopybit` library Android provides another hardware-specific library for managing graphic operations. This library is called: `gralloc`. `gralloc` (short from graphic allocation) library is needed when `/dev/fb*` (or `/dev/graphics/fb*`) device is not available in the system (for any reasons). This is very important, as Android determines/sets screen information and allocates surfaces using `graphics/fb*` commands for that device. In order to avoid this (or if your platform doesn't support this device) you will have to put platform specific implementation of the graphic allocation operations in this library.

The Skia module is primarily used for resource rendering, like icons, images, etc. This module also supports introduction of the platform specific acceleration, but not on the library level, rather on the function level. These functions include:

- `S32_D565_Opaque`
- `S32_D565_Blend`
- `S32A_D565_Opaque`

- S32A_D565_Blend
- S32_D565_Opaque_Dither
- S32_D565_Blend_Dither
- S32A_D565_Opaque_Dither
- S32A_D565_Blend_Dither
- S32_D4444_Opaque
- S32_D4444_Blend
- S32A_D4444_Opaque
- S32A_D4444_Blend
- S32_D4444_Opaque_Dither
- S32_D4444_Blend_Dither
- S32A_D4444_Opaque_Dither
- S32A_D4444_Blend_Dither
- S32_Opaque
- S32_Blend
- S32A_Opaque
- S32A_Blend

These functions perform several different blitting operations (with dither, blend, etc) on the row level for several different color formats. In MIPS Android source tree some of these functions are implemented using MIPS DSP ASE instruction set (both rev 1 and 2). Beside blitting, Skia performs also scaling, rotation and perspective operations which are present in Matrix module. These functions include:

- ClampX_ClampY_nofilter_scale
- ClampX_ClampY _filter_scale
- ClampX_ClampY_nofilter_affine
- ClampX_ClampY_filter_affine
- ClampX_ClampY_nofilter_persp
- ClampX_ClampY_filter_persp
- RepeatX_RepeatY_nofilter_scale

- RepeatX_RepeatY_filter_scale
- RepeatX_RepeatY_nofilter_affine
- RepeatX_RepeatY_filter_affine
- RepeatX_RepeatY_nofilter_persp
- RepeatX_RepeatY_filter_persp
- Filter_32_alpha
- Filter_32_opaque

In MIPS Android source tree these functions are also implemented using MIPS DSP ASE instruction set.

All of previously mentioned Skia routines are on row/pixel level. This approach is not suited for optimization of the Skia library using other hardware accelerated graphic libraries (like DirectFB), as the overhead of inserting new library calls will exceed performance gain for small amount of data. In order to fully utilize acceleration of the other graphic libraries, implementation will have to be on frame/surface/rectangular level, which is complex task.

Document Revision History

Revision	Date	Description
01.00	February 3, 2011	Initial version.

Unpublished rights (if any) reserved under the copyright laws of the United States of America and other countries.

This document contains information that is proprietary to MIPS Tech, LLC, a Wave Computing company ("MIPS") and MIPS' affiliates as applicable. Any copying, reproducing, modifying or use of this information (in whole or in part) that is not expressly permitted in writing by MIPS or MIPS' affiliates as applicable or an authorized third party is strictly prohibited. At a minimum, this information is protected under unfair competition and copyright laws. Violations thereof may result in criminal penalties and fines. Any document provided in source format (i.e., in a modifiable form such as in FrameMaker or Microsoft Word format) is subject to use and distribution restrictions that are independent of and supplemental to any and all confidentiality restrictions. UNDER NO CIRCUMSTANCES MAY A DOCUMENT PROVIDED IN SOURCE FORMAT BE DISTRIBUTED TO A THIRD PARTY IN SOURCE FORMAT WITHOUT THE EXPRESS WRITTEN PERMISSION OF MIPS (AND MIPS' AFFILIATES AS APPLICABLE) reserve the right to change the information contained in this document to improve function, design or otherwise.

MIPS and MIPS' affiliates do not assume any liability arising out of the application or use of this information, or of any error or omission in such information. Any warranties, whether express, statutory, implied or otherwise, including but not limited to the implied warranties of merchantability or fitness for a particular purpose, are excluded. Except as expressly provided in any written license agreement from MIPS or an authorized third party, the furnishing of this document does not give recipient any license to any intellectual property rights, including any patent rights, that cover the information in this document.

The information contained in this document shall not be exported, reexported, transferred, or released, directly or indirectly, in violation of the law of any country or international law, regulation, treaty, Executive Order, statute, amendments or supplements thereto. Should a conflict arise regarding the export, reexport, transfer, or release of the information contained in this document, the laws of the United States of America shall be the governing law.

The information contained in this document constitutes one or more of the following: commercial computer software, commercial computer software documentation or other commercial items. If the user of this information, or any related documentation of any kind, including related technical data or manuals, is an agency, department, or other entity of the United States government ("Government"), the use, duplication, reproduction, release, modification, disclosure, or transfer of this information, or any related documentation of any kind, is restricted in accordance with Federal Acquisition Regulation 12.212 for civilian agencies and Defense Federal Acquisition Regulation Supplement 227.7202 for military agencies. The use of this information by the Government is further restricted in accordance with the terms of the license agreement(s) and/or applicable contract terms and conditions covering this information from MIPS Technologies or an authorized third party.

MIPS, MIPS I, MIPS II, MIPS III, MIPS IV, MIPS V, MIPSr3, MIPS32, MIPS64, microMIPS32, microMIPS64, MIPS-3D, MIPS16, MIPS16e, MIPS-Based, MIPSsim, MIPSpro, MIPS-VERIFIED, Aptiv logo, microAptiv logo, interAptiv logo, microMIPS logo, MIPS Technologies logo, MIPS-VERIFIED logo, proAptiv logo, 4K, 4Kc, 4Km, 4Kp, 4KE, 4KEc, 4KEm, 4KEp, 4KS, 4KSc, 4KSd, M4K, M14K, 5K, 5Kc, 5Kf, 24K, 24Kc, 24Kf, 24KE, 24KEc, 24KEf, 34K, 34Kc, 34Kf, 74K, 74Kc, 74Kf, 1004K, 1004Kc, 1004Kf, 1074K, 1074Kc, 1074Kf, R3000, R4000, R5000, Aptiv, ASMACRO, Atlas, "At the core of the user experience.", BusBridge, Bus Navigator, CLAM, CorExtend, CoreFPGA, CoreLV, EC, FPGA View, FS2, FS2 FIRST SILICON SOLUTIONS logo, FS2 NAVIGATOR, HyperDebug, HyperJTAG, IASim, iFlowtrace, interAptiv, JALGO, Logic Navigator, Malta, MDMX, MED, MGB, microAptiv, microMIPS, Navigator, OCl, PDtrace, the Pipeline, proAptiv, Pro Series, SEAD-3, SmartMIPS, SOC-it, and YAMON are trademarks or registered trademarks of MIPS and MIPS' affiliates as applicable in the United States and other countries.

All other trademarks referred to herein are the property of their respective owners.