Freescale Semiconductor

Document Number: IMX6EGLOGLES
Rev. L3.0.35_4.1.0, 09/2013

i.MX 6 EGL & OGL Extension Support

Contents

1 Introduction

The following tables list the level of support for EGL and OES extensions available with i.MX 6 hardware and software. Support levels are current as of the date of the document and subject to change.

Two tables are provided. The first table lists the EGL interface extensions. The second table lists extensions for both OpenGL ES 1.1 and OpenGL ES 2.0.

Key:
Extension Name and Number: Each listed extension is derived from the relevant khronos.org webpage list and includes the extension number as well as a hyperlink to the khronos description of the extension.
Yes: Support is currently available.
No: Support is not available. (Reasons for lack of support may vary: the extension may be proprietary or obsolete, or not applicable to the specified OES version.)
N/A: Support is not provided as the extension is not applicable in this and subsequent versions of the specification.

2 EGL Extension Support

The following table includes the list of all current EGL Extensions and indicates their support level.

(list from http://www.khronos.org/registry/egl/ as of 1/24/2013)

EGL Extension Number, Name, and hyperlink	Supported?
1. EGL_KHR_config_attribs	No
2. EGL_KHR_lock_surface	Yes
3. EGL_KHR_image	Yes
4. EGL_KHR_vg_parent_image	No
5. EGL_KHR_gl_texture_2D_image EGL_KHR_gl_texture_cubemap_image EGL_KHR_gl_texture_3D_image EGL_KHR_gl_renderbuffer_image	Yes Yes No Yes
6. EGL_KHR_reusable_sync	Yes
8. EGL_KHR_image_base	Yes
9. EGL_KHR_image_pixmap	Yes
10. EGL_IMG_context_priority	No
16. EGL_KHR_lock_surface2	No
17. EGL_NV_coverage_sample	No
18. EGL_NV_depth_nonlinear	No
19. EGL_NV_sync	No
20. EGL_KHR_fence_sync	Yes
24. EGL_HI_clientpixmap	No
25. EGL_HI_colorformats	No
26. EGL_MESA_drm_image	No
27. EGL_NV_post_sub_buffer	No
28. EGL_ANGLE_query_surface_pointer	No
29. EGL_ANGLE_surface_d3d_texture_2d_share_handle	No
30. EGL_NV_coverage_sample_resolve	No
31. EGL_NV_system_time	No
32. EGL_KHR_stream	No
33. EGL_KHR_stream_consumer_gltexture	No
34. EGL_KHR_stream_producer_eglsurface	No
35. EGL_KHR_stream_producer_aldatalocator	No
36. EGL_KHR_stream_fifo	No
37. EGL_EXT_create_context_robustness	No
38. EGL_ANGLE_d3d_share_handle_client_buffer	No
39. EGL_KHR_create_context	No
40. EGL_KHR_surfaceless_context	No
41. EGL_KHR_stream_cross_process_fd	No
42. EGL_EXT_multiview_window	No
43. EGL_KHR_wait_sync	No
44. EGL_NV_post_convert_rounding	No
45. EGL_NV_native_query	No
46. EGL_NV_3dvision_surface	No
47. EGL_ANDROID_framebuffer_target	No
48. EGL_ANDROID_blob_cache	No

49. EGL_ANDROID_image_native_buffer	Yes
50. EGL_ANDROID_native_fence_sync	No
51. EGL_ANDROID_recordable	No
52. EGL_EXT_buffer_age	No
53. EGL_EXT_image_dma_buf_import	No
54. EGL_ARM_pixmap_multisample_discard	No
55. EGL_ANDROID_get_render_buffer	Yes
56. EGL_ANDROID_swap_rectangle	Yes

3 OpenGL ES Extension Support

The following table includes the list of all current OpenGL ES Extensions and indicates their support level.

(list from http://www.khronos.org/registry/gles/ as of 9/27/2012)

Extension Number, Name and hyperlink	ES1.1	ES2.0
1. GL_OES_blend_equation_separate	Yes	N/A
2. GL_OES_blend_func_separate	Yes	N/A
3. GL_OES_blend_subtract	Yes	N/A
4. GL_OES_byte_coordinates	Yes	N/A
5. GL_OES_compressed_ETC1_RGB8_texture	Yes	Yes
6. GL_OES_compressed_paletted_texture	Yes	Yes
7. GL_OES_draw_texture	Yes	N/A
8. GL_OES_extended_matrix_palette	Yes	No
9. GL_OES_fixed_point	Yes	No
10. GL_OES_framebuffer_object	Yes	N/A
11. GL_OES_matrix_get	Yes	N/A
12. GL_OES_matrix_palette	Yes	N/A
14. GL_OES_point_size_array	Yes	No
15. GL_OES_point_sprite	Yes	No
16. GL_OES_query_matrix	Yes	N/A
17. GL_OES_read_format	Yes	No
18. GL_OES_single_precision	Yes	No
19. GL_OES_stencil_wrap	Yes	No
20. GL_OES_texture_cube_map	Yes	N/A
21. GL_OES_texture_env_crossbar	No	No
22. GL_OES_texture_mirrored_repeat	Yes	N/A
23. GL_OES_EGL_image	Yes	Yes
24. GL_OES_depth24	Yes	Yes
25. GL_OES_depth32	No	No
26. GL_OES_element_index_uint	Yes	Yes
27. GL_OES_fbo_render_mipmap	Yes	Yes
28. GL_OES_fragment_precision_high	No	Yes
29. GL_OES_mapbuffer	Yes	Yes
30. GL_OES_rgb8_rgba8	Yes	Yes
31. GL_OES_stencil1	Yes	Yes
32. GL_OES_stencil4	Yes	Yes
33. GL_OES_stencil8	Yes	N/A
34. GL_OES_texture_3D	No	No
35. GL_OES_texture_float_linear GL_OES_texture_half_float_linear	No No	No No
36. GL_OES_texture_float GL_OES_texture_half_float	No No	No No
37. GL_OES_texture_npot	Yes	Yes
38. GL_OES_vertex_half_float	Yes	Yes
39. GL_AMD_compressed_3DC_texture	No	No

Extension Number, Name and hyperlink	**ES1.1**	**ES2.0**
40. GL_AMD_compressed_ATC_texture	No	No
41. GL_EXT_texture_filter_anisotropic	Yes	Yes
42. GL_EXT_texture_type_2_10_10_10_REV	No	Yes
43. GL_OES_depth_texture	No	Yes
44. GL_OES_packed_depth_stencil	Yes	Yes
45. GL_OES_standard_derivatives	No	Yes
46. GL_OES_vertex_type_10_10_10_2	No	Yes
47. GL_OES_get_program_binary	No	Yes
48. GL_AMD_program_binary_Z400	No	No
49. GL_EXT_texture_compression_dxt1		
50. GL_AMD_performance_monitor	No	No
51. GL_EXT_texture_format_BGRA8888	Yes	Yes
52. GL_NV_fence	No	No
53. GL_IMG_read_format	No	No
54. GL_IMG_texture_compression_pvrtc	No	No
55. GL_QCOM_driver_control	No	No
56. GL_QCOM_performance_monitor_global_mode	No	No
57. GL_IMG_user_clip_plane	No	No
58. GL_IMG_texture_env_enhanced_fixed_function	No	No
59. GL_APPLE_texture_2D_limited_npot	No	No
60. GL_EXT_texture_lod_bias	Yes	N/A
61. GL_QCOM_writeonly_rendering	No	No
62. GL_QCOM_extended_get	No	No
63. GL_QCOM_extended_get2	No	No
64. GL_EXT_discard_framebuffer	No	Yes
65. GL_EXT_blend_minmax	Yes	Yes
66. GL_EXT_read_format_bgra	Yes	Yes
67. GL_IMG_program_binary	No	No
68. GL_IMG_shader_binary	No	No
69. GL_EXT_multi_draw_arrays GL_SUN_multi_draw_arrays	Yes No	Yes No
70. GL_QCOM_tiled_rendering	No	No
71. GL_OES_vertex_array_object	No	No
72. GL_NV_coverage_sample	No	No
73. GL_NV_depth_nonlinear	No	No
74. GL_IMG_multisampled_render_to_texture	No	No
75. GL_OES_EGL_sync	Yes	N/A
76. GL_APPLE_rgb_422	No	No
77. GL_EXT_shader_texture_lod	No	No
78. GL_APPLE_framebuffer_multisample	No	No
79. GL_APPLE_texture_format_BGRA8888	No	No
80. GL_APPLE_texture_max_level	No	No
81. GL_ARM_mali_shader_binary	No	No
82. GL_ARM_rgba8	No	No
83. GL_ANGLE_framebuffer_blit	No	No
84. GL_ANGLE_framebuffer_multisample	No	No

Extension Number, Name and hyperlink	ES1.1	ES2.0
85. GL_VIV_shader_binary	No	Yes
86. GL_EXT_frag_depth	No	Yes
87. GL_OES_EGL_image_external	Yes	Yes
88. GL_DMP_shader_binary	No	No
89. GL_QCOM_alpha_test	No	No
90. GL_EXT_unpack_subimage	No	N/A
91. GL_NV_draw_buffers	No	No
92. GL_NV_fbo_color_attachments	No	No
93. GL_NV_read_buffer	No	No
94. GL_NV_read_depth_stencil	No	No
95. GL_NV_texture_compression_s3tc_update	No	No
96. GL_NV_texture_npot_2D_mipmap	No	No
97. GL_EXT_color_buffer_half_float	No	No
98. GL_EXT_debug_label	No	No
99. GL_EXT_debug_marker	No	No
100. GL_EXT_occlusion_query_boolean	No	No
101. GL_EXT_separate_shader_objects	No	No
102. GL_EXT_shadow_samplers	No	No
103. GL_EXT_texture_rg	No	No
104. GL_NV_EGL_stream_consumer_external	No	No
105. GL_EXT_sRGB	No	No
106. GL_EXT_multisampled_render_to_texture	No	Yes
107. GL_EXT_robustness	No	No
108. GL_EXT_texture_storage	No	No
109. GL_ANGLE_instanced_arrays	No	No
110. GL_ANGLE_pack_reverse_row_order	No	No
111. GL_ANGLE_texture_compression_dxt3 GL_ANGLE_texture_compression_dxt5	No No	No No
112. GL_ANGLE_texture_usage	No	No
113. GL_ANGLE_translated_shader_source	No	No
114. GL_FJ_shader_binary_GCCSO	No	No
115. GL_OES_required_internalformat	No	No
116. GL_OES_surfaceless_context	No	No
117. GL_KHR_texture_compression_astc_ldr	No	No
118. GL_KHR_debug	No	No
119. GL_QCOM_binning_control	No	No
120. GL_ARM_mali_program_binary	No	No
121. GL_EXT_map_buffer_range	No	No
122. GL_EXT_shader_framebuffer_fetch	No	No
123. GL_APPLE_copy_texture_levels	No	No
124. GL_APPLE_sync	No	No
125. GL_EXT_multiview_draw_buffers	No	No
126. GL_NV_draw_texture	No	No
127. GL_NV_packed_float	No	No
128. GL_NV_texture_compression_s3tc	No	No
129. GL_NV_3dvision_settings	No	No

Extension Number, Name and hyperlink	ES1.1	ES2.0
130. GL_NV_texture_compression_latc	No	No
131. GL_NV_platform_binary	No	No
132. GL_NV_pack_subimage	No	No
133. GL_NV_texture_array	No	No
134. GL_NV_pixel_buffer_object	No	No
135. GL_NV_bgr	No	No
136. GL_OES_depth_texture_cube_map	No	No
137. GL_EXT_color_buffer_float	No	No
138. GL_ANGLE_depth_texture	No	No
139. GL_ANGLE_program_binary	No	No
140. GL_IMG_texture_compression_pvrtc2	No	No
141. GL_NV_framebuffer_blit	No	No
142. GL_NV_framebuffer_multisample	No	No
143. GL_NV_generate_mipmap_sRGB	No	No
144. GL_NV_instanced_arrays	No	No
145. GL_NV_shadow_samplers_array	No	No
146. GL_NV_shadow_samplers_cube	No	No
147. GL_NV_sRGB_formats	No	No
148. GL_NV_texture_border_clamp	No	No
149. GL_VIV_direct_texture	Yes	Yes
150. GL_VIV_texture_border_clamp	No	Yes

4 Extension GL_VIV_direct_texture

Name

VIV_direct_texture

Name strings

GL_VIV_direct_texture

IPStatus

Contact Freescale Semiconductor regarding any intellectual property questions associated with this extension.

Status

Implemented: July, 2011

Version

Last modified: 29 July, 2011
Revision: 2

Number

Unassigned

Dependencies

OpenGL ES 1.1 is required. OpenGL ES 2.0 support is available.

Overview

Create a texture with direct access support. This is useful when an application desires to use the same texture over and over while frequently updating its content. It could also be used for mapping live video to a texture. A video decoder could write its result directly to the texture and then the texture could be directly rendered onto a 3D shape. `glTexDirectVIVMap` is similar to `glTexDirectVIV`. The only difference is that it has two inputs, “Logical” and “Physical,” which support mapping a user space memory or a physical address into the texture surface.

New Procedures and Functions

glTexDirectVIV

Syntax:

```
GL_API void GL_APIENTRY
```

```
glTexDirectVIV (
       GLenum              Target,
       GLsizei             Width,
       GLsizei             Height,
       GLenum              Format,
       GLvoid **           Pixels
);
```

Parameters

Target Target texture. Must be GL_TEXTURE_2D.

Width
Height Size of LOD 0. Width must be 16 pixel aligned. The width and height of LOD 0 of the texture is specified by the Width and Height parameters. The driver may auto-generate the rest of LODs if the hardware supports high quality scaling (for non-power of 2 textures) and LOD generation. If the hardware does not support high quality scaling and LOD generation, the texture will remain a single-LOD texture.

Format You can choose the format of the pixel data from the following formats: GL_VIV_YV12, GL_VIV_NV12, GL_VIV_NV21, GL_VIV_YUY2, GL_VIV_UYVY, GL_RGBA, and GL_BGRA_EXT.

- If the format is GL_VIV_YV12, glTexDirectVIV creates a planar YV12 4:2:0 texture and the format of the Pixels array is as follows: Yplane, Vplane, Uplane.
- If the format is GL_VIV_NV12, glTexDirectVIV creates a planar NV12 4:2:0 texture and the format of the Pixels array is as follows: Yplane, UVplane.
- If the format is GL_VIV_NV21, glTexDirectVIV creates a planar NV21 4:2:0 texture and the format of the Pixels array is as follows: Yplane, VUplane.
- If the format is GL_VIV_YUY2 or GL_VIV_UYVY, glTexDirectVIV creates a packed 4:2:2 texture and the Pixels array contains only one pointer to the packed YUV texture.
- If Format is GL_RGBA, glTexDirectVIV creates a pixel array with four GL_UNSIGNED_BYTE components: the first byte for red pixels, the second byte for green pixels, the third byte for blue, and the fourth byte for alpha.
- If Format is GL_BGRA_EXT, glTexDirectVIV creates a pixel array with four GL_UNSIGNED_BYTE components: the first byte for blue pixels, the second byte for green pixels, the third byte for red, and the fourth byte for alpha.

Pixels Stores the memory pointer created by the driver.

Output

If the function succeeds, it returns a pointer, or, for some YUV formats, it returns a set of pointers that directly point to the texture. The pointer(s) will be returned in the user-allocated array pointed to by the Pixels parameter.

GlTexDirectVIVMap

Syntax:

```
GL_API void GL_APIENTRY
glTexDirectVIVMap (
       Glenum              Target,
       Glsizei             Width,
       Glsizei             Height,
       Glenum              Format,
       Glvoid **           Logical,
       const Gluint *      Physical
);
```

Parameters

Target	Target texture. Must be GL_TEXTURE_2D.
Width **Height**	Size of LOD 0. Width must be 16 pixel aligned. See glTexDirectVIV.
Format	Same as `glTexDirectVIV` Format.
Logical	Pointer to the logical address of the application-defined texture buffer. Logical address must be 64 bit (8 byte) aligned.
Physical	Pointer to the physical address of the application-defined buffer to the texture, or ~0 if no physical address has been provided.

GlTexDirectInvalidateVIV

Syntax:

```
GL_API void GL_APIENTRY
glTexDirectInvalidateVIV (
       Glenum              Target
);
```

Parameters

Target	Target texture. Must be GL_TEXTURE_2D.

New Tokens

GL_VIV_YV12	0x8FC0
GL_VIV_NV12	0x8FC1
GL_VIV_YUY2	0x8FC2
GL_VIV_UYVY	0x8FC3
GL_VIV_NV21	0x8FC4

Error codes

GL_INVALID_ENUM	Target is not GL_TEXTURE_2D, or format is not a valid format.

GL_INVALID_VALUE	Width or Height parameter is less than 1.
GL_OUT_OF_MEMORY	A memory allocation error occurred.
GL_INVALID_OPERATION	Specified format is not supported by the hardware, or no texture is bound to the active texture unit, or some other error occurs during the call.

Example 1.

First, call `glTexDirectVIV` to get a pointer.
Second, copy the texture data to this memory address.
Then, call `glTexDirectInvalidateVIV` to apply the texture before you draw something with that texture.

```
… …
glTexDirectVIV(GL_TEXUTURE_2D, 512, 512, GL_VIV_YV12, &texels);
… …
GlTexDirectInvalidateVIV(GL_TEXTURE_2D);
…
glDrawArrays(…);
…
```

Example 2.

First, call `glTexDirectVIVMap` to map Logical and Physical address to the texture.
Second, you can modify Logical and Physical data.
Then, call `glTexDirectInvalidateVIV` to apply the texture before you draw something with that texture.

```
… …
char *Logical = (char*) malloc (sizeof(char)*size);
Gluint physical = ~0U;
glTexDirectVIVMap(GL_TEXUTURE_2D, 512, 512, GL_VIV_YV12,
    (void**)&Logical, &13hysical);
… …
GlTexDirectInvalidateVIV(GL_TEXTURE_2D);
…
glDrawArrays(…);
```

Issues

None

5 Extension GL_VIV_texture_border_clamp

Name

VIV_texture_border_clamp

Name Strings

GL_VIV_texture_border_clamp

Status

Implemented September 2012.

Version

Last modified: 27 September 2012
Vivante revision: 1

Number

Unassigned

Dependencies

This extension is implemented for use with OpenGL ES 1.1 and OpenGL ES 2.0.

This extension is based on OpenGL ARB Extension #13: GL_ARB_texture_border_clamp: http://www.opengl.org/registry/specs/ARB/texture_border_clamp.txt. See also vendor extension GL_SGIS_texture_border_clamp, http://www.opengl.org/registry/specs/SGIS/texture_border_clamp.txt.

Overview

This extension was adapted from the OpenGL extension for use with OpenGL ES implementations. The OpenGL ARB Extension 13 description applies here as well:

> "The base OpenGL provides clamping such that the texture coordinates are limited to exactly the range [0,1]. When a texture coordinate is clamped using this algorithm, the texture sampling filter straddles the edge of the texture image, taking 1/2 its sample values from within the texture image, and the other 1/2 from the texture border. It is sometimes desirable for a texture to be clamped to the border color, rather than to an average of the border and edge colors.
>
> This extension defines an additional texture clamping algorithm. CLAMP_TO_BORDER_[VIV] clamps texture coordinates at all mipmap levels such that NEAREST and LINEAR filters return only the color of the border texels."

The color returned is derived only from border texels and cannot be configured.

Issues

None

New Tokens

Accepted by the `<param>` parameter of `TexParameteri` and `TexParameterf`, and by the `<params>` parameter of `TexParameteriv` and `TexParameterfv`, when their `<pname>` parameter is TEXTURE_WRAP_S, TEXTURE_WRAP_T, or TEXTURE_WRAP_R:

CLAMP_TO_BORDER_VIV	0x812D

Errors

None.

New State

Only the type information changes for these parameters.

See OES 2.0 Specification Section 3.7.4, page 75-76, Table 3.10, “Texture parameters and their values.”

6 Revision History

This section describes top level differences between document revisions.

Version	Date	Driver Version	Notes
1.4	2013-07-22	4.6.9-p12	Inclusion of 4.6.9p12
1.3	2013-04-15	4.6.9-p11.1	Restore indication of support for EGL: - #6 EGL_KHR_reusable_sync - #20 EGL_KHR_fence_sync. Add EGL extension 54 to list. Formatting adjustments GL_VIV_direct_texture: Formats: - Expand descriptions for GL_RGBA, GL_BGRA_EXT. - Remove GL_RGB and GL_RGB565 as supported formats. Add indication of support for GL 1.1: - GL_OES_texture_npot Add indication of support for GL 2.0: - GL_EXT_discard_framebuffer Remove indication of support for GL 1.1: - GL_EXT_unpack_subimage - GL_VIV_texture_border_clamp (remains for GL2) Change level of support for GL 2.0: - GL_OES_texture_half_float_linear, change from YES to NO Add EGL extension 53-54 to list. Add OES extensions 138-149 to list
1.2	2013-03-01	4.6.9-p11	• Added the following EGL extensions: ○ EGL_EXT_multiview_window ○ EGL_KHR_wait_sync ○ EGL_NV_post_convert_rounding ○ EGL_NV_native_query ○ EGL_NV_3dvision_surface ○ EGL_ANDROID_framebuffer_target ○ EGL_ANDROID_blob_cache ○ EGL_ANDROID_native_fence_sync ○ EGL_ANDROID_recordable ○ EGL_EXT_buffer_age ○ EGL_ANDROID_get_render_buffer ○ EGL_ANDROID_swap_rectangle • Removed EGL_KHR_reusable_sync and EGL_KHR_fence_sync support for EGL • Added the following OES extensions: ○ GL_EXT_multiview_draw_buffers ○ GL_NV_draw_texture ○ GL_NV_packed_float ○ GL_NV_texture_compression_s3tc ○ GL_NV_3dvision_settings

			○ GL_NV_texture_compression_latc ○ GL_NV_platform_binary ○ GL_NV_pack_subimage ○ GL_NV_texture_array ○ GL_NV_pixel_buffer_object ○ GL_NV_bgr ○ GL_OES_depth_texture_cube_map ○ GL_EXT_color_buffer_float • Added GL_EXT_multisampled_render_to_texture support for OES 2.0 • Added GL_BGRA_EXT format for GL_VIV_direct_texture • Modified GL_VIV_texure_border_clamp extension
1.1	2013-01-09	4.6.9-p9	• Added OES extensions 120-124 • Added RGB support for GL_VIV_direct_texture • Corrected GL_VIV_direct_texture Example 2 code • Added GL_VIV_texture_border_clamp extension
1.0	2012-10-04	4.6.9-p6	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.