

# Metal Feature Set Tables

This table lists each current Apple GPU family, the processors in that family, and how they relate to older feature sets.

## Apple GPUs

Apple GPU Family <sup>1</sup>	GPUs in Family	Corresponding Feature Sets
Apple1	A7	iOS GPU Family 1
Apple2	A8	iOS GPU Family 2 tvOS GPU Family 1
Apple3	A9, A10	iOS GPU Family 3 tvOS GPU Family 2
Apple4	A11	iOS GPU Family 4
Apple5	A12	iOS GPU Family 5
Apple6	A13	—
Apple7	A14 M1, M1 Pro, M1 Max, M1 Ultra	—
Apple8	A15	—

1. See [MTLGPUFamily](#) for each GPU family's enumeration constant.

### Metal Feature Availability by GPU Family

GPU Family <sup>1</sup>	Common1	Common2	Common3	Apple1	Apple2	Apple3	Apple4	Apple5	Apple6	Apple7	Apple8	Mac1	Mac2	Catalyst1	Catalyst2
Feature	Available in Family														
MetalKit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Metal Performance Shaders		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Programmable blending				✓	✓	✓	✓	✓	✓	✓	✓				
PVRTC pixel formats				✓	✓	✓	✓	✓	✓	✓	✓				
EAC/ETC pixel formats				✓	✓	✓	✓	✓	✓	✓	✓				
ASTC pixel formats					✓	✓	✓	✓	✓	✓	✓				
BC pixel formats												✓	✓	✓	✓
Compressed volume texture formats		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Extended range pixel formats						✓	✓	✓	✓	✓	✓				
Wide color pixel format	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Depth-16 pixel format	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Linear textures	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSAA depth resolve			✓			✓	✓	✓	✓	✓	✓		✓		✓
Array of textures (read)		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Array of textures (write)		✓	✓						✓	✓	✓	✓	✓	✓	✓
Cube map texture arrays		✓	✓				✓	✓	✓	✓	✓	✓	✓	✓	✓
Stencil texture views	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Array of samplers		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sampler max anisotropy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sampler LOD clamp	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MTLSamplerState support for comparison functions		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
16-bit unsigned integer coordinates	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Border color										✓	✓	✓	✓	✓	✓
Counting occlusion query		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Base vertex/instance drawing		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Layered rendering			✓					✓	✓	✓	✓	✓	✓	✓	✓
Layered rendering to multisample textures			✓							✓	✓		✓		✓
Memoryless render targets				✓	✓	✓	✓	✓	✓	✓	✓				
Dual-source blending	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Combined MSAA store and resolve action		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSAA bits	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Programmable sample positions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Deferred store action	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Texture barriers												✓	✓	✓	✓
Memory barriers <sup>2</sup>						✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tessellation		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Indirect tessellation arguments								✓	✓	✓	✓	✓	✓	✓	✓
Tessellation in indirect command buffers								✓	✓	✓	✓	✓	✓	✓	✓
Resource heaps	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Function specialization	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Function buffer read-writes		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Function texture read-writes			✓				✓	✓	✓	✓	✓	✓	✓	✓	✓
Extract, insert, and reverse bits	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SIMD barrier	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Indirect draw & dispatch arguments		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

GPU Family <sup>1</sup>	Common1	Common2	Common3	Apple1	Apple2	Apple3	Apple4	Apple5	Apple6	Apple7	Apple8	Mac1	Mac2	Catalyst1	Catalyst2
Argument buffers tier	Varies	Varies	Varies	Tier 1	Tier 1	Tier 1	Tier 1	Tier 1	Tier 2	Tier 2	Tier 2	Varies	Varies	Varies	Varies
Indirect command buffers (rendering)		✓	✓			✓	✓	✓	✓	✓	✓		✓		✓
Indirect command buffers (compute)		✓	✓			✓	✓	✓	✓	✓	✓				
Uniform type	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Imageblocks							✓	✓	✓	✓	✓				
Tile shaders							✓	✓	✓	✓	✓				
Imageblock sample coverage control							✓	✓	✓	✓	✓				
Post-depth coverage							✓	✓	✓	✓	✓				
Quad-scoped permute operations			✓				✓	✓	✓	✓	✓		✓		✓
SIMD-scoped permute operations									✓	✓	✓	✓	✓	✓	✓
SIMD-scoped reduction operations										✓	✓		✓		✓
SIMD-scoped matrix multiply operations										✓	✓				
Raster order groups <sup>3</sup>			✓				✓	✓	✓	✓	✓	Varies	Varies	Varies	Varies
Non-uniform threadgroup size			✓				✓	✓	✓	✓	✓	✓	✓	✓	✓
Multiple viewports			✓					✓	✓	✓	✓	✓	✓	✓	✓
Device notifications												✓	✓	✓	✓
Stencil feedback			✓					✓	✓	✓	✓		✓		✓
Stencil resolve			✓					✓	✓	✓	✓		✓		✓
Non-square tile dispatch								✓	✓	✓	✓				
Texture swizzle				✓	✓	✓	✓	✓	✓	✓	✓		✓		✓
Placement heap				✓	✓	✓	✓	✓	✓	✓	✓		✓		✓
Pre-clipped primitive id and barycentric coordinates										✓	✓		✓		✓
Read or write cube map textures							✓	✓	✓	✓	✓	✓	✓	✓	✓
Sparse textures									✓	✓	✓				
Sparse depth & stencil textures											✓				
Variable rasterization rate									✓	✓	✓				
Vertex amplification									✓	✓	✓				
64-bit integer math						✓	✓	✓	✓	✓	✓				
Lossy texture compression											✓				
Simd shift and fill											✓				
Render dynamic libraries									✓	✓	✓		✓		✓
Compute dynamic libraries									✓	✓	✓		✓		✓

1. See `MTLGPUFamily` for each GPU family's enumeration constant.

2. The GPUs in Apple3 through Apple8 families only support memory barriers for compute command encoders, and for vertex-to-vertex and vertex-to-fragment stages of render command encoders.

3. Some GPU devices in Mac and Mac Catalyst families support raster order groups. You can check an individual GPU's support for this feature by inspecting its `MTLDevice.rasterOrderGroupsSupported` property at runtime.

## Implementation Limits by GPU Family

GPU Family <sup>1</sup>	Apple1	Apple2	Apple3	Apple4	Apple5	Apple6	Apple7	Apple8	Mac1	Mac2	Catalyst1	Catalyst2
<b>Function Arguments</b>	<b>Function Arguments</b>											
Maximum number of vertex attributes, per vertex descriptor	31	31	31	31	31	31	31	31	31	31	31	31
Maximum number of entries in the buffer argument table, per graphics or kernel function	31	31	31	31	31	31	31	31	31	31	31	31
Maximum number of entries in the texture argument table, per graphics or kernel function	31	31	31	96	96	128	128	128	128	128	128	128
Maximum number of entries in the sampler state argument table, per graphics or kernel function <sup>2</sup>	16	16	16	16	16	16	16	16	16	16	16	16
Maximum number of entries in the threadgroup memory argument table, per kernel function	31	31	31	31	31	31	31	31	31	31	31	31
Maximum number of constant buffer arguments in vertex, fragment, tile, or kernel function	31	31	31	31	31	31	31	31	14	14	14	14
Maximum length of constant buffer arguments in vertex, fragment, tile, or kernel function	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB
Maximum threads per threadgroup <sup>3</sup>	512	512	512	1024	1024	1024	1024	1024	1024	1024	1024	1024
Maximum total threadgroup memory allocation	16352 B	16352 B	16 KB	32 KB	32 KB	32 KB	32 KB	32 KB	32 KB	32 KB	32 KB	32 KB
Maximum total tile memory allocation <sup>4</sup>	Not accessible	Not accessible	Not accessible	32 KB	32 KB	32 KB	32 KB	32 KB	Not accessible	Not accessible	Not accessible	Not accessible
Threadgroup memory length alignment	16 B	16 B	16 B	16 B	16 B	16 B	16 B	16 B	16 B	16 B	16 B	16 B
Maximum function memory allocation for a buffer in the constant address space	No limit	No limit	No limit	No limit	No limit	No limit	No limit	No limit	64 KB	64 KB	64 KB	64 KB
Maximum number of inputs (scalars or vectors) to a fragment function, declared with the stage_in qualifier <sup>4</sup>	60	60	60	124	124	124	124	124	32	32	32	32
Maximum number of input components to a fragment function, declared with the stage_in qualifier <sup>5</sup>	60	60	60	124	124	124	124	124	124	124	124	124
Maximum number of function constants	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536	65536
Maximum tessellation factor	Not available	Not available	16	16	64	64	64	64	64	64	64	64
Maximum number of viewports and scissor rectangles, per vertex function	1	1	1	1	16	16	16	16	16	16	16	16
Maximum number of raster order groups, per fragment function	Not available	Not available	Not available	8	8	8	8	8	8	8	8	8
<b>Argument Buffers<sup>6</sup></b>	<b>Argument Buffers</b>											
Maximum number of buffers inside an argument buffer	31	31	31	96	96	500000	500000	500000	Varies by tier	Varies by tier	Varies by tier	Varies by tier
Maximum number of textures inside an argument buffer	31	31	31	96	96	500000	500000	500000	Varies by tier	Varies by tier	Varies by tier	Varies by tier
Maximum number of samplers inside an argument buffer	16	16	16	16	16	1024	1024	1024	Varies by tier	Varies by tier	Varies by tier	Varies by tier

GPU Family <sup>1</sup>	Apple1	Apple2	Apple3	Apple4	Apple5	Apple6	Apple7	Apple8	Mac1	Mac2	Catalyst1	Catalyst2
<b>Resources</b>	<b>Resources</b>											
Minimum constant buffer offset alignment	4 B	4 B	4 B	4 B	4 B	4 B	4 B	4 B	256 B	32 B	256 B	32 B
Maximum 1D texture width	8192 px	8192 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px
Maximum 2D texture width and height	8192 px	8192 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px
Maximum cube map texture width and height	8192 px	8192 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px	16384 px
Maximum 3D texture width, height, and depth	2048 px	2048 px	2048 px	2048 px	2048 px	2048 px	2048 px	2048 px	2048 px	2048 px	2048 px	2048 px
Maximum number of layers per 1D texture array, 2D texture array, or 3D texture	2048	2048	2048	2048	2048	2048	2048	2048	2048	2048	2048	2048
Buffer alignment for copying an existing texture to a buffer	64 B	64 B	16 B	16 B	16 B	16 B	16 B	16 B	256 B	256 B	256 B	256 B
<b>Render Targets</b>	<b>Render Targets</b>											
Maximum number of color render targets per render pass descriptor	4	8	8	8	8	8	8	8	8	8	8	8
Maximum size of a point primitive	54	54	54	54	54	54	54	54	54	54	54	54
Maximum total render target size, per pixel, when using multiple color render targets	128 bits	256 bits	256 bits	512 bits	512 bits	512 bits	512 bits	512 bits	No limit	No limit	No limit	No limit
Maximum visibility query offset	65528 B	65528 B	65528 B	65528 B	65528 B	65528 B	256 KB	256 KB	256 KB	256 KB	256 KB	256 KB
<b>Synchronization</b>	<b>Synchronization</b>											
Maximum number of fences	32768	32768	32768	32768	32768	32768	32768	32768	32768	32768	32768	32768
Maximum vertex count for vertex amplification	Not available	Not available	Not available	Not available	Not available	2	2	2	Not available	Not available	Not available	Not available

1. See `MTLGPUFamily` for each GPU family's enumeration constant.
2. Inline, `constexpr` samplers that you declare in `Metal Shading Language` (MSL) code count against the limit. For example, for a feature set limit of 16, you can have 12 API samplers and 4 language samplers (16 total), but you can't have 12 API samplers and 6 language samplers (18 total).
3. The values in this row are the theoretical maximum number of threads per threadgroup. Check the actual maximum by inspecting the `MTLComputePipelineState.maxTotalThreadsPerThreadgroup` property at runtime.
4. You can allocate memory between `imageblock` and `threadgroup` memory, but the sum of these allocations can't exceed the maximum total tile memory limit. Some feature sets can't access tile memory directly, but they can access threadgroup memory.
5. A vector counts as  $n$  scalars, where  $n$  is the number of components in the vector. The iOS and tvOS feature sets only reach the maximum number of inputs if you don't exceed the maximum number of input components. For example, you can have 60 float inputs (components) but you can't have 60 `float4` inputs, which totals 240 components.
6. For GPUs in Apple6 through Apple8 families, the limit applies to the number of buffers and textures, combined. The macOS GPU families (Mac1, Mac2, Catalyst1, and Catalyst2) with Tier 1 argument buffers support 64 buffers, 128 textures, and 16 samplers. The macOS GPU families with Tier 2 argument buffers support 500,000 buffers and textures, and 1,024 unique samplers.



This table lists the GPU's texture capabilities for each pixel format:

- **All:** The GPU has all of the texture capabilities below for the pixel format.
- **Filter:** The GPU can filter a texture with the pixel format during sampling.
- **Write:** The GPU can write to a texture on a per-pixel basis with the pixel format.<sup>1</sup>
- **Color:** The GPU can use a texture with the pixel format as a color render target.
- **Blend:** The GPU can blend a texture with the pixel format.
- **MSAA:** The GPU can use a texture with the pixel format as a destination for multisample antialias (MSAA) data.
- **Sparse:** The GPU supports sparse-texture allocations for textures with the pixel format.
- **Resolve:** The GPU can use a texture with the pixel format as a source for multisample antialias (MSAA) resolve operations.

**Note**

All graphics and compute kernels can read or sample a texture with any pixel format.

### GPU Texture Capabilities by Pixel Format

GPU Family <sup>2</sup>	Apple1	Apple2	Apple3	Apple4	Apple5	Apple6	Apple7	Apple8	Mac1	Mac2	Catalyst1	Catalyst2
<b>Ordinary 8-bit pixel formats</b>	Texture capabilities for <b>ordinary 8-bit pixel formats</b> by GPU Family											
A8Unorm	Filter	Filter	Filter	Filter	Filter	Filter Sparse	Filter Sparse	Filter Sparse	Filter	Filter	Filter	Filter
R8Unorm <sup>1</sup>	All	All	All	All	All	All	All	All	All	All	All	All
R8Unorm_sRGB	Filter Color MSAA Resolve Blend	All	All	All	All	All	All	All	Not available	Not available	Not available	Not available
R8Snorm	Filter Write Color MSAA Blend	All	All	All	All	All	All	All	All	All	All	All
R8Uint <sup>1</sup> R8Sint <sup>1</sup>	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
<b>Ordinary 16-bit pixel formats</b>	Texture capabilities for <b>ordinary 16-bit pixel formats</b> by GPU family											
R16Unorm R16Snorm	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend Sparse	Filter Write Color MSAA Blend Sparse	Filter Write Color MSAA Blend Sparse	All	All	All	All
R16Uint <sup>1</sup> R16Sint <sup>1</sup>	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
R16Float <sup>1</sup>	All	All	All	All	All	All	All	All	All	All	All	All
RG8Unorm	All	All	All	All	All	All	All	All	All	All	All	All
RG8Unorm_sRGB	Filter Color MSAA Resolve Blend	All	All	All	All	All	All	All	Not available	Not available	Not available	Not available
RG8Snorm	Filter Write Color MSAA Blend	All	All	All	All	All	All	All	All	All	All	All
RG8Uint RG8Sint	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA

GPU Family <sup>2</sup>	Apple1	Apple2	Apple3	Apple4	Apple5	Apple6	Apple7	Apple8	Mac1	Mac2	Catalyst1	Catalyst2
<b>Packed 16-bit pixel formats</b>	Texture capabilities for <b>packed 16-bit pixel formats</b> by GPU family											
B5G6R5Unorm A1BGR5Unorm ABGR4Unorm BGR5A1Unorm	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend Sparse	Filter Color MSAA Resolve Blend Sparse	Filter Color MSAA Resolve Blend Sparse	Not available	Not available	Not available	Not available
<b>Ordinary 32-bit pixel formats</b>	Texture capabilities for <b>ordinary 32-bit pixel formats</b> by GPU family											
R32Uint <sup>1</sup> R32Sint <sup>1</sup>	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color Sparse	Write Color Sparse	Write Color Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
R32Float <sup>1</sup>	Write Color MSAA Blend	Write Color MSAA Blend	Write Color MSAA Blend	Write Color MSAA Blend	Write Color MSAA Blend	Write Color MSAA Blend Sparse	Write Color MSAA Blend Sparse	Write Color MSAA Blend Sparse	All	All	All	All
RG16Unorm RG16Snorm	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend Sparse	Filter Write Color MSAA Blend Sparse	Filter Write Color MSAA Blend Sparse	All	All	All	All
RG16Uint RG16Sint	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
RG16Float	All	All	All	All	All	All	All	All	All	All	All	All
RGBA8Unorm <sup>1</sup>	All	All	All	All	All	All	All	All	All	All	All	All
RGBA8Unorm_sRGB	Filter Color MSAA Resolve Blend	All	All	All	All	All	All	All	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend
RGBA8Snorm	Filter Write Color MSAA Blend	All	All	All	All	All	All	All	All	All	All	All
RGBA8Uint <sup>1</sup> RGBA8Sint <sup>1</sup>	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
BGRA8Unorm	All	All	All	All	All	All	All	All	All	All	All	All
BGRA8Unorm_sRGB	Filter Color MSAA Resolve Blend	All	All	All	All	All	All	All	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend



GPU Family <sup>2</sup>	Apple1	Apple2	Apple3	Apple4	Apple5	Apple6	Apple7	Apple8	Mac1	Mac2	Catalyst1	Catalyst2
<b>Packed 32-bit pixel formats</b>	Texture capabilities for <b>packed 32-bit pixel formats</b> by GPU family											
RGB10A2Unorm	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	All	All	All	All	All	All	All	All	All	All
RGB10A2Uint	Color MSAA	Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
RG4B10Float	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	All	All	All	All	All	All	All	All	All	All
RGB9E5Float	Filter Color MSAA Resolve Blend	Filter Color MSAA Resolve Blend	All	All	All	All	All	All	Filter	Filter	Filter	Filter
<b>Ordinary 64-bit pixel formats</b>	Texture capabilities for <b>ordinary 64-bit pixel formats</b> by GPU family											
RG32Uint RG32Sint	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color Sparse	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
RG32Float	Write Color Blend	Write Color Blend	Write Color Blend	Write Color Blend	Write Color Blend	Write Color Blend Sparse	Write Color MSAA Blend Sparse	Write Color MSAA Blend Sparse	All	All	All	All
RGBA16Unorm RGBA16Snorm	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend	Filter Write Color MSAA Blend Sparse	Filter Write Color MSAA Blend Sparse	Filter Write Color MSAA Blend Sparse	All	All	All	All
RGBA16Uint <sup>1</sup> RGBA16Sint <sup>1</sup>	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
RGBA16Float <sup>1</sup>	All	All	All	All	All	All	All	All	All	All	All	All
<b>Ordinary 128-bit pixel formats</b>	Texture capabilities for <b>ordinary 128-bit pixel formats</b> by GPU family											
RGBA32Uint <sup>1</sup> RGBA32Sint <sup>1</sup>	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color Sparse	Write Color Sparse	Write Color Sparse	Write Color MSAA	Write Color MSAA	Write Color MSAA	Write Color MSAA
RGBA32Float <sup>1</sup>	Write Color	Write Color	Write Color	Write Color	Write Color	Write Color Sparse	Write Color MSAA Sparse	Write Color MSAA Sparse	All	All	All	All
<b>Compressed pixel formats</b>	Texture capabilities for <b>compressed pixel formats</b> by GPU family											
PVRTC pixel formats <sup>3</sup>	Filter	Filter	Filter	Filter	Filter	Filter Sparse	Filter Sparse	Filter Sparse	Not available	Not available	Not available	Not available
EAC/ETC pixel formats	Filter	Filter	Filter	Filter	Filter	Filter Sparse	Filter Sparse	Filter Sparse	Not available	Not available	Not available	Not available
ASTC pixel formats	Not available	Filter	Filter	Filter	Filter	Filter Sparse	Filter Sparse	Filter Sparse	Not available	Not available	Not available	Not available
HDR ASTC pixel formats	Not available	Not available	Not available	Not available	Not available	Filter Sparse	Filter Sparse	Filter Sparse	Not available	Not available	Not available	Not available
BC pixel formats	Not available	Not available	Not available	Not available	Not available	Not available	Not available	Not available	Filter	Filter	Filter	Filter

GPU Family <sup>2</sup>	Apple1	Apple2	Apple3	Apple4	Apple5	Apple6	Apple7	Apple8	Mac1	Mac2	Catalyst1	Catalyst2
<b>YUV pixel formats<sup>4</sup></b>	Texture capabilities for <b>YUV pixel formats</b> by GPU family											
GBGR422 BGRG422	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
<b>Depth and stencil pixel formats</b>	Texture capabilities for <b>depth and stencil pixel formats</b> by GPU family											
Depth16Unorm	Filter MSAA	Filter MSAA	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve Sparse	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve
Depth32Float	MSAA	MSAA	MSAA Resolve	MSAA Resolve	MSAA Resolve	MSAA Resolve	MSAA Resolve	MSAA Resolve Sparse	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve
Stencil8	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA Sparse	MSAA	MSAA	MSAA	MSAA
Depth24Unorm_Stencil8 <sup>5</sup>	Not available	Not available	Not available	Not available	Not available	Not available	Not available	Not available	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve
Depth32Float_Stencil8	MSAA	MSAA	MSAA Resolve	MSAA Resolve	MSAA Resolve	MSAA Resolve	MSAA Resolve	MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve	Filter MSAA Resolve
X24_Stencil8	Not available	Not available	Not available	Not available	Not available	Not available	Not available	Not available	MSAA	MSAA	MSAA	MSAA
X32_Stencil8	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA	MSAA
<b>Extended range and wide color pixel formats</b>	Texture capabilities for <b>extended range and wide color formats</b> by GPU family											
BGRA10_XR BGRA10_XR_sRGB BGR10_XR BGR10_XR_sRGB	Not available	Not available	All	All	All	All	All	All	Not available	Not available	Not available	Not available
BGR10A2Unorm	All	All	All	All	All	All	All	All	Filter Color MSAA Resolve Blend	All	Filter Color MSAA Resolve Blend	All

1. Some GPUs support read-write textures — where a kernel can both read from and write to a texture. You can check an individual GPU's support for this feature by inspecting its `MTLDevice.readWriteTextureSupport` property at runtime.
2. See `MTLGPUFamily` for each GPU family's enumeration constant.
3. Only the GPUs in Apple3 and Apple4 families support `MTLSamplerAddressMode.clampToZero` for the PVRTC pixel formats.
4. The GPUs in Apple6 through Apple8 families don't support sparse textures with YUV pixel formats.
5. Some GPUs support `MTLPixelFormat.depth24Unorm_stencil8`. You can check an individual GPU's support for this pixel format by inspecting its `MTLDevice.isDepth24Stencil8PixelFormatSupported` property at runtime.

# Texture Buffer Pixel Formats

These tables list the pixel formats that texture buffers support and the GPU's read/write access to textures with those formats:

- **All:** The GPU can use all the accesses below for a texture in the pixel format.
- **Read:** The GPU can use read access for a texture buffer with the pixel format.
- **Write:** The GPU can use write access for a texture buffer with the pixel format.
- **Read/Write<sup>1</sup>:** The GPU can use read\_write access for a texture buffer with the pixel format.

## Note

The GPU capabilities are generally the same across all hardware families, but some GPUs have additional options.<sup>2</sup>

Ordinary 8-bit pixel formats	
Format	Access
A8Unorm	Read
R8Unorm	All
R8Snorm	Read Write
R8Uint R8Sint	All

Ordinary 16-bit pixel formats	
Format	Access
R16Unorm R16Snorm	Read Write
R16Uint R16Sint	All
R16Float	All
RG8Unorm	Read Write
RG8Snorm	Read Write
RG8Uint RG8Sint	Read Write

Ordinary 32-bit Pixel Formats	
Format	Access
R32Uint R32Sint	All
R32Float	All
RG16Unorm RG16Snorm	Read Write
RG16Uint RG16Sint	Read Write
RG16Float	Read Write
RGBA8Unorm	All
RGBA8Snorm	Read Write
RGBA8Uint RGBA8Sint	All
BGRA8Unorm	Read

Packed 32-bit Pixel Formats	
Format	Access
RGB10A2Unorm	Read Write
RGB10A2Uint	Read Write
RG4B10Float	Read Write

Ordinary 64-bit pixel formats	
Format	Access
RG32Uint RG32Sint	Read Write
RG32Float	Read Write
RGBA16Unorm RGBA16Snorm	Read Write
RGBA16Uint RGBA16Sint	All
RGBA16Float	All

Ordinary 128-bit pixel formats	
Format	Access
RGBA32Uint RGBA32Sint	All
RGBA32Float	All

1. GPUs with the Tier 2 feature set support read\_write access to textures. You can check an individual GPU's support for this pixel format by inspecting its `MTLDevice.readWriteTextureSupport` property at runtime.

2. Some devices support this pixel format. Check a device by inspecting its `MTLDevice.depth24Stencil18PixelFormatSupported` property at runtime.



Apple Inc.  
Copyright © 2022 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer or device for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-branded products.

Apple Inc.  
One Apple Park Way  
Cupertino, CA 95014

Apple is a trademark of Apple Inc., registered in the U.S. and other countries.

**APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT, ERROR OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**Some jurisdictions do not allow the exclusion of implied warranties or liability, so the above exclusion may not apply to you.**