Metal feature set tables
This table lists each current Apple GPU family, its processors, and how each family relates to older feature sets.

### Apple GPUs

<table>
<thead>
<tr>
<th>Apple GPU family</th>
<th>GPUs in family</th>
<th>Corresponding feature sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple2</td>
<td>A8</td>
<td>iOS GPU Family 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tvOS GPU Family 1</td>
</tr>
<tr>
<td>Apple3</td>
<td>A9, A10</td>
<td>iOS GPU Family 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tvOS GPU Family 2</td>
</tr>
<tr>
<td>Apple4</td>
<td>A11</td>
<td>iOS GPU Family 4</td>
</tr>
<tr>
<td>Apple5</td>
<td>A12</td>
<td>iOS GPU Family 5</td>
</tr>
<tr>
<td>Apple6</td>
<td>A13</td>
<td>—</td>
</tr>
<tr>
<td>Apple7</td>
<td>A14</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>M1, M1 Pro, M1 Max, M1 Ultra</td>
<td></td>
</tr>
<tr>
<td>Apple8</td>
<td>A15, A16</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>M2, M2 Pro, M2 Max</td>
<td></td>
</tr>
<tr>
<td>Apple9</td>
<td>A17 Pro</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>M3, M3 Pro, M3 Max</td>
<td></td>
</tr>
</tbody>
</table>

1. See `MTLGPUSamily` for each GPU family’s enumeration constant.

When an Apple GPU is installed in a Mac device (Apple Silicon Mac), the device also reports support for the `mac2` GPU family; these devices support the union of both feature families.
This table lists each current Metal 3 GPU family and the processors in that family.

### Metal 3 GPUs

<table>
<thead>
<tr>
<th>Metal GPU family¹</th>
<th>Platform</th>
<th>GPUs in family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metal3</td>
<td>iOS</td>
<td>A14, A15, A16, A17 Pro</td>
</tr>
<tr>
<td></td>
<td>iPadOS</td>
<td>A14, A15, A16 M1, M2</td>
</tr>
<tr>
<td></td>
<td>macOS</td>
<td>M1, M1 Pro, M1 Max, M1 Ultra M2, M2 Pro, M2 Max M3, M3 Pro, M3 Max AMD Vega AMD 5000-series, 6000-series Intel UHD Graphics 630 Intel Iris Plus Graphics</td>
</tr>
</tbody>
</table>

1. See [MTLGPUFamily](#) for each GPU family’s enumeration constant.
## Metal feature availability by GPU family

<table>
<thead>
<tr>
<th>Feature</th>
<th>Common1</th>
<th>Common2</th>
<th>Common3</th>
<th>Metal3</th>
<th>Apple2</th>
<th>Apple3</th>
<th>Apple4</th>
<th>Apple5</th>
<th>Apple6</th>
<th>Apple7</th>
<th>Apple8</th>
<th>Apple9</th>
<th>Mac2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetalKit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory barriers in indirect command buffers (compute)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory barriers in indirect command buffers (rendering)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Texture barriers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable sample positions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deferred store action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined MSAA store and resolve action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA depth resolve</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counting occlusion query</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base vertex/instance drawing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dual-source blending</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined MSAA store and resolve action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable sample positions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deferred store action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined MSAA store and resolve action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable sample positions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deferred store action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined MSAA store and resolve action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable sample positions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deferred store action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined MSAA store and resolve action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable sample positions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deferred store action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined MSAA store and resolve action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable sample positions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deferred store action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined MSAA store and resolve action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable sample positions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deferred store action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined MSAA store and resolve action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable sample positions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deferred store action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined MSAA store and resolve action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAA blits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPU family</td>
<td>Common1</td>
<td>Common2</td>
<td>Common3</td>
<td>Metal3</td>
<td>Apple2</td>
<td>Apple3</td>
<td>Apple4</td>
<td>Apple5</td>
<td>Apple6</td>
<td>Apple7</td>
<td>Apple8</td>
<td>Apple9</td>
<td>Mac2</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>Tessellation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indirect tessellation arguments</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Tessellation in indirect command buffers</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource heaps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Function specialization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Read/write buffers in functions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Read/write textures in functions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Extract, insert, and reverse bits</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>SIMD barrier</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Indirect draw &amp; dispatch arguments</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Argument buffers tier</td>
<td>Varies</td>
<td>Varies</td>
<td>Varies</td>
<td>Tier 2</td>
<td>Tier 1</td>
<td>Tier 1</td>
<td>Tier 1</td>
<td>Tier 1</td>
<td>Tier 2</td>
<td>Tier 2</td>
<td>Tier 2</td>
<td>Tier 2</td>
<td></td>
</tr>
<tr>
<td>Indirect command buffers (rendering)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Indirect command buffers (compute)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Uniform type</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Imageblocks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Tile shaders</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Imageblock sample coverage control</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Post-depth coverage</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Quad-scoped permute operations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>SIMD-scoped permute operations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>SIMD-scoped reduction operations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>SIMD-scoped matrix multiply operations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Raster order groups</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Varies</td>
</tr>
<tr>
<td>Non-uniform threadgroup size</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Multiple viewports</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Device notifications</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Stencil feedback</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Stencil resolve</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Non-square tile dispatch</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Texture swizzle</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Placement heap</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Primitive ID</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Barycentric coordinates</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Varies</td>
</tr>
<tr>
<td>Read/write cube map textures in functions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Sparse textures</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Sparse depth and stencil textures</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Variable rasterization rate</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Varies</td>
</tr>
<tr>
<td>Vertex amplification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Varies</td>
</tr>
<tr>
<td>64-bit integer math</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

October 31, 2023

Page 4 of 15

Copyright © 2023 Apple Inc. All Rights Reserved.
<table>
<thead>
<tr>
<th>GPU family</th>
<th>Common1</th>
<th>Common2</th>
<th>Common3</th>
<th>Metal3</th>
<th>Apple2</th>
<th>Apple3</th>
<th>Apple4</th>
<th>Apple5</th>
<th>Apple6</th>
<th>Apple7</th>
<th>Apple8</th>
<th>Apple9</th>
<th>Mac2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lossy texture compression</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMD shift and fill</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Render dynamic libraries</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compute dynamic libraries</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mesh shading</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MetalFX spatial upscaling⁸</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>MetalFX temporal upscaling⁹</td>
<td></td>
<td></td>
<td></td>
<td>❌</td>
<td></td>
<td>❌</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fast resource loading</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ray tracing in compute pipelines¹⁰</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Ray tracing in render pipelines¹¹</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Floating point atoms</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Texture atoms</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64-bit atomics¹²</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Query texture LOD¹³</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Binary archives</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Function pointers in compute pipelines¹⁴</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Function pointers in render pipelines¹⁴</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Depth sample compare bias and gradient</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Non-private depth stencil textures</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Dynamic stride for attribute buffers</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>MTLAttributeFormat.floatRGB9E5 and .floatRGB11E14</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>MTLDataType.bfloat (brain float) scalar and vector cases</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
</tbody>
</table>

1. See MTLGPUFamily for each GPU family's enumeration constant.
2. Some GPU devices in the Apple7 and Apple8 families support BC texture compression on iPadOS. You can check whether a GPU supports BC texture compression by inspecting its `MTLDevice.supportsBCTextureCompression` property at runtime.
3. Some GPU devices in the Mac2 family support vertex amplification. You can check whether a GPU supports vertex amplification by calling its `MTLDevice.supportsShaderBarycentricCoordinates` method at runtime.
4. Some GPU devices in the Mac2 family support raster order groups. You can check whether a GPU supports raster order groups by calling the `MTLDevice.rasterOrderGroupsSupported` property at runtime.
5. Some GPU devices in the Mac2 and Metal3 families support barycentric coordinates. You can check whether a GPU supports barycentric coordinates by inspecting its `MTLDevice.supportsShaderBarycentricCoordinates` property at runtime.
6. Some GPU devices in the Mac2 family support variable rasterization rates. You can check whether a GPU supports variable rasterization rates by inspecting its `MTLDevice.supportsRasterizationRateMap` method at runtime.
7. Some GPU devices in the Mac2 family support vertex amplification. You can check whether a GPU supports vertex amplification by calling its `MTLDevice.supportsVertexAmplificationCount(_:)` method at runtime.
8. Some GPU devices in the Mac2 family support ray tracing in compute pipelines. You can check whether a GPU supports ray tracing in compute pipelines by inspecting its `MTLDevice.supportsRayTracing` property at runtime.
9. Some GPU devices in the Mac2 family support ray tracing in render pipelines. You can check whether a GPU supports ray tracing in render pipelines by inspecting its `MTLDevice.supportsRayTracing` property at runtime.
10. Some GPU devices in the Mac2 family support ray tracing in render pipelines. You can check whether a GPU supports ray tracing in render pipelines by inspecting its `MTLDevice.supportsRayTracing` property at runtime.
11. Some GPU devices in the Mac2 family support query texture LOD. You can check whether a GPU supports query texture LOD by inspecting its `MTLDevice.supportsQueryTextureLOD` property at runtime.
<table>
<thead>
<tr>
<th>GPU family1</th>
<th>Apple2</th>
<th>Apple3</th>
<th>Apple4</th>
<th>Apple5</th>
<th>Apple6</th>
<th>Apple7</th>
<th>Apple8</th>
<th>Apple9</th>
<th>Mac2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function arguments</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum number of vertex attributes, per vertex descriptor</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>Maximum number of entries in the buffer argument table, per graphics or kernel function</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>Maximum number of entries in the texture argument table, per graphics or kernel function</td>
<td>31</td>
<td>31</td>
<td>96</td>
<td>96</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>Maximum number of entries in the sampler state argument table, per graphics or kernel function</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Maximum number of entries in the threadgroup memory argument table, per kernel function</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>Maximum number of constant buffer arguments in vertex, fragment, tile, or kernel function</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>14</td>
</tr>
<tr>
<td>Maximum length of constant buffer arguments in vertex, fragment, tile, or kernel function</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
</tr>
<tr>
<td>Maximum threads per threadgroup3</td>
<td>512</td>
<td>512</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Maximum total threadgroup memory allocation</td>
<td>16352 B</td>
<td>16 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
</tr>
<tr>
<td>Maximum total tile memory allocation4</td>
<td>Not accessible</td>
<td>Not accessible</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>Not accessible</td>
</tr>
<tr>
<td>Threadgroup memory length alignment</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
</tr>
<tr>
<td>Maximum function memory allocation for a buffer in the constant address space</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
</tr>
<tr>
<td>Maximum scalars or vectors inputs to a fragment function. (Declare with the [[stage_in]] qualifier4.)</td>
<td>60</td>
<td>60</td>
<td>124</td>
<td>124</td>
<td>124</td>
<td>124</td>
<td>124</td>
<td>124</td>
<td>32</td>
</tr>
<tr>
<td>Maximum number of input components to a fragment function. (Declare with the [[stage_in]] qualifier3.)</td>
<td>60</td>
<td>60</td>
<td>124</td>
<td>124</td>
<td>124</td>
<td>124</td>
<td>124</td>
<td>124</td>
<td>124</td>
</tr>
<tr>
<td>Maximum number of function constants</td>
<td>65536</td>
<td>65536</td>
<td>65536</td>
<td>65536</td>
<td>65536</td>
<td>65536</td>
<td>65536</td>
<td>65536</td>
<td>65536</td>
</tr>
<tr>
<td>Maximum tessellation factor</td>
<td>Not available</td>
<td>16</td>
<td>16</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>Maximum number of viewports and scissor rectangles, per vertex function</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Maximum number of raster order groups, per fragment function</td>
<td>Not available</td>
<td>Not available</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>GPU family1</td>
<td>Apple2</td>
<td>Apple3</td>
<td>Apple4</td>
<td>Apple5</td>
<td>Apple6</td>
<td>Apple7</td>
<td>Apple8</td>
<td>Apple9</td>
<td>Mac2</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>Minimum alignment of buffer layout descriptor stride</td>
<td>4 B</td>
<td>4 B</td>
<td>4 B</td>
<td>1 B</td>
<td>1 B</td>
<td>1 B</td>
<td>1 B</td>
<td>1 B</td>
<td>4 B</td>
</tr>
<tr>
<td>Maximum size of buffer layout descriptor stride</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>No limit</td>
<td>4 KB</td>
</tr>
<tr>
<td><strong>Argument buffers</strong></td>
<td><strong>Argument buffers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum number of buffers you can access, per stage, from an argument buffer</td>
<td>31</td>
<td>31</td>
<td>96</td>
<td>96</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Maximum number of textures you can access, per stage, from an argument buffer</td>
<td>31</td>
<td>31</td>
<td>96</td>
<td>96</td>
<td>1 M</td>
<td>1 M</td>
<td>1 M</td>
<td>1 M</td>
<td>1 M</td>
</tr>
<tr>
<td>Maximum number of samplers you can access, per stage, from an argument buffer</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>500 K</td>
<td>1024</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td><strong>Resources</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum constant buffer offset alignment</td>
<td>4 B</td>
<td>4 B</td>
<td>4 B</td>
<td>4 B</td>
<td>4 B</td>
<td>4 B</td>
<td>4 B</td>
<td>4 B</td>
<td>32 B</td>
</tr>
<tr>
<td>Maximum 1D texture width</td>
<td>8192 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
</tr>
<tr>
<td>Maximum 2D texture width and height</td>
<td>8192 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
</tr>
<tr>
<td>Maximum cube map texture width and height</td>
<td>8192 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
<td>16384 px</td>
</tr>
<tr>
<td>Maximum 3D texture width, height, and depth</td>
<td>2048 px</td>
<td>2048 px</td>
<td>2048 px</td>
<td>2048 px</td>
<td>2048 px</td>
<td>2048 px</td>
<td>2048 px</td>
<td>2048 px</td>
<td>2048 px</td>
</tr>
<tr>
<td>Maximum texture buffer width7</td>
<td>64M px</td>
<td>256M px</td>
<td>256M px</td>
<td>256M px</td>
<td>256M px</td>
<td>256M px</td>
<td>256M px</td>
<td>256M px</td>
<td>256M px</td>
</tr>
<tr>
<td>Maximum number of layers per 1D texture array, 2D texture array, or 3D texture</td>
<td>2048</td>
<td>2048</td>
<td>2048</td>
<td>2048</td>
<td>2048</td>
<td>2048</td>
<td>2048</td>
<td>2048</td>
<td>2048</td>
</tr>
<tr>
<td>Buffer alignment for copying an existing texture to a buffer</td>
<td>64 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>16 B</td>
<td>256 B</td>
</tr>
<tr>
<td>Maximum counter sample buffer length</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>32 KB</td>
<td>No limit</td>
</tr>
<tr>
<td>Maximum number of sample buffers</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>No limit</td>
<td>No limit</td>
</tr>
<tr>
<td><strong>Render targets</strong></td>
<td><strong>Render targets</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum number of color render targets per render pass descriptor</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Maximum size of a point primitive</td>
<td>511</td>
<td>511</td>
<td>511</td>
<td>511</td>
<td>511</td>
<td>511</td>
<td>511</td>
<td>511</td>
<td>511</td>
</tr>
<tr>
<td>Feature limits</td>
<td>Feature limits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Maximum total render target size, per pixel, when using multiple color render targets</strong></td>
<td><strong>256 bits</strong></td>
<td><strong>256 bits</strong></td>
<td><strong>512 bits</strong></td>
<td><strong>512 bits</strong></td>
<td><strong>512 bits</strong></td>
<td><strong>512 bits</strong></td>
<td><strong>512 bits</strong></td>
<td><strong>512 bits</strong></td>
<td><strong>No limit</strong></td>
</tr>
<tr>
<td><strong>Maximum visibility query offset</strong></td>
<td><strong>65528 B</strong></td>
<td><strong>65528 B</strong></td>
<td><strong>65528 B</strong></td>
<td><strong>65528 B</strong></td>
<td><strong>256 KB</strong></td>
<td><strong>256 KB</strong></td>
<td><strong>256 KB</strong></td>
<td><strong>256 KB</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Feature limits</strong></td>
<td><strong>Maximum number of fences</strong></td>
<td><strong>32768</strong></td>
<td><strong>32768</strong></td>
<td><strong>32768</strong></td>
<td><strong>32768</strong></td>
<td><strong>32768</strong></td>
<td><strong>32768</strong></td>
<td><strong>32768</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Maximum number of I/O commands per buffer</strong></td>
<td><strong>8192</strong></td>
<td><strong>8192</strong></td>
<td><strong>8192</strong></td>
<td><strong>8192</strong></td>
<td><strong>8192</strong></td>
<td><strong>8192</strong></td>
<td><strong>8192</strong></td>
<td><strong>8192</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Maximum vertex count for vertex amplification</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>2</strong></td>
<td><strong>8</strong></td>
<td><strong>8</strong></td>
<td><strong>8</strong></td>
<td><strong>Varies</strong></td>
</tr>
<tr>
<td><strong>Maximum threadgroups per object shader grid</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>No limit</strong></td>
<td><strong>No limit</strong></td>
<td><strong>No limit</strong></td>
<td><strong>1024</strong></td>
</tr>
<tr>
<td><strong>Maximum threadgroups per mesh shader grid</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>1024</strong></td>
<td><strong>1024</strong></td>
<td><strong>1048575</strong></td>
<td><strong>1024</strong></td>
</tr>
<tr>
<td><strong>Maximum payload in mesh shader pipeline</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>16384 B</strong></td>
<td><strong>16384 B</strong></td>
<td><strong>16384 B</strong></td>
<td><strong>16384 B</strong></td>
</tr>
<tr>
<td><strong>Largest number of levels a ray-tracing intersector can traverse in an acceleration structure</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>32</strong></td>
<td><strong>32</strong></td>
<td><strong>32</strong></td>
<td><strong>32</strong></td>
<td><strong>32</strong></td>
</tr>
<tr>
<td><strong>Largest number of levels a ray-tracing intersection query can traverse in an acceleration structure</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>Not available</strong></td>
<td><strong>16</strong></td>
<td><strong>16</strong></td>
<td><strong>16</strong></td>
<td><strong>16</strong></td>
<td><strong>16</strong></td>
</tr>
</tbody>
</table>

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 total 240 components.
6. The limits apply to the items you place both in the argument buffers you bind directly and in the argument buffers you can access indirectly through your bound argument buffers.
7. The maximum texture buffer width, in pixels, is also limited by `MTLDevice.maxBufferLength` divided by the size of a pixel, in bytes; as well as available memory.
8. Some GPU devices in the Mac2 family support vertex amplification. You can check an individual GPU's support for a specific vertex amplification count by calling its `MTLDevice.supportsVertexAmplificationCount(_:)` method at runtime.
9. Mesh shaders can use up to 4 GB of payload and mesh geometry per draw for devices in the Apple7 and Apple8 GPU families.
10. Mesh shaders that have a `[[threadgroups_per_grid]]` or `[[threads_per_grid]]` parameter reduce the available payload size by 16 bytes. Viewing a mesh shader's geometry in the Metal debugger (within Xcode) reduces the available payload by 16 bytes. The total payload size reduction can be 32 bytes.
11. The value includes one level for the primitive acceleration structure, which leaves the remaining levels for instance acceleration structures.

---

**October 31, 2023**

---

**Page 8 of 15**

---

**Copyright © 2023 Apple Inc. All Rights Reserved.**
This table lists the GPU’s texture capabilities for each pixel format:

- **Atomic**: The GPU can use atomic operations on textures with the 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.
- **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.

### Texture capabilities by pixel format

<table>
<thead>
<tr>
<th>GPU family</th>
<th>Apple2</th>
<th>Apple3</th>
<th>Apple4</th>
<th>Apple5</th>
<th>Apple6</th>
<th>Apple7</th>
<th>Apple8</th>
<th>Apple9</th>
<th>Mac2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ordinary 8-bit pixel formats</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A8Unorm2</td>
<td>Filter</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>R8Unorm2</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>R8Unorm_sRGB</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>Not available</td>
</tr>
<tr>
<td>R8Snorm</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>R8Uint2</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
</tr>
<tr>
<td>R8Sint2</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
</tr>
<tr>
<td><strong>Ordinary 16-bit pixel formats</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R16Unorm</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>All</td>
</tr>
<tr>
<td>R16Snorm</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>All</td>
</tr>
<tr>
<td>R16Uint2</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>All</td>
</tr>
<tr>
<td>R16Sint2</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>All</td>
</tr>
<tr>
<td>R16Float2</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>GPU family</td>
<td>Apple2</td>
<td>Apple3</td>
<td>Apple4</td>
<td>Apple5</td>
<td>Apple6</td>
<td>Apple7</td>
<td>Apple8</td>
<td>Apple9</td>
<td>Mac2</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>RG8Unorm</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RG8Unorm_sRGB</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>Not available</td>
</tr>
<tr>
<td>RG8Snorm</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RG8Uint</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Color</td>
<td>Color</td>
<td>MSAA</td>
<td>Sparse</td>
<td>Sparse</td>
<td>Sparse</td>
</tr>
<tr>
<td>RG8Sint</td>
<td>Color</td>
<td>Color</td>
<td>Color</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
</tr>
<tr>
<td>Packed 16-bit pixel formats</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
<td>Blend</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
</tr>
<tr>
<td>B5G6R5Unorm</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
<td>Blend</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
</tr>
<tr>
<td>A1BGR5Unorm</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
<td>Blend</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
</tr>
<tr>
<td>ABGR4Unorm</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
<td>Blend</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
</tr>
<tr>
<td>BGR5A1Unorm</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
<td>Blend</td>
<td>Filter</td>
<td>Color</td>
<td>MSAA</td>
<td>Resolve</td>
</tr>
<tr>
<td>Ordinary 32-bit pixel formats</td>
<td>Filter</td>
<td>Write</td>
<td>Color</td>
<td>MSAA</td>
<td>Blend</td>
<td>Filter</td>
<td>Write</td>
<td>Color</td>
<td>MSAA</td>
</tr>
<tr>
<td>R32Uint</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
</tr>
<tr>
<td>R32Sint</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
</tr>
<tr>
<td>R32Float</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
</tr>
<tr>
<td>RG16Unorm</td>
<td>Filter</td>
<td>Write</td>
<td>Color</td>
<td>MSAA</td>
<td>Blend</td>
<td>Filter</td>
<td>Write</td>
<td>Color</td>
<td>MSAA</td>
</tr>
<tr>
<td>RG16Snorm</td>
<td>Filter</td>
<td>Write</td>
<td>Color</td>
<td>MSAA</td>
<td>Blend</td>
<td>Filter</td>
<td>Write</td>
<td>Color</td>
<td>MSAA</td>
</tr>
<tr>
<td>RG16Uint</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
</tr>
<tr>
<td>RG16Sint</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
<td>Write</td>
</tr>
<tr>
<td>RG16Float</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
</tbody>
</table>

October 31, 2023
<table>
<thead>
<tr>
<th>GPU family</th>
<th>Apple2</th>
<th>Apple3</th>
<th>Apple4</th>
<th>Apple5</th>
<th>Apple6</th>
<th>Apple7</th>
<th>Apple8</th>
<th>Apple9</th>
<th>Mac2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGBA8Unorm</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RGBA8Unorm_sRGB</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RGBA8Snorm</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RGBA8Uint</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
</tr>
<tr>
<td>RGBA8Sint</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA Sparse</td>
<td>Write Color MSAA Sparse</td>
<td>Write Color MSAA Sparse</td>
<td>Write Color MSAA Sparse</td>
<td>Write Color MSAA Sparse</td>
<td>Write Color MSAA Sparse</td>
<td>Write Color MSAA Sparse</td>
</tr>
<tr>
<td>BGRA8Unorm</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>BGRA8Unorm_sRGB</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>BGRA8Unorm_sRGB</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
</tbody>
</table>

**Packed 32-bit pixel formats**

Texture capabilities for **packed 32-bit pixel formats** by GPU family

<table>
<thead>
<tr>
<th>RGBA10A2Unorm</th>
<th>Filter Color MSAA Resolve Blend</th>
<th>All</th>
<th>All</th>
<th>All</th>
<th>All</th>
<th>All</th>
<th>All</th>
<th>All</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGR10A2Unorm</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RGB10A2Uint</td>
<td>Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
</tr>
<tr>
<td>RG11B10Float</td>
<td>Filter Color MSAA Resolve Blend</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RGB9E5Float</td>
<td>Filter Color MSAA Resolve Blend</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>Filter</td>
</tr>
<tr>
<td>GPU family</td>
<td>Apple2</td>
<td>Apple3</td>
<td>Apple4</td>
<td>Apple5</td>
<td>Apple6</td>
<td>Apple7</td>
<td>Apple8</td>
<td>Apple9</td>
<td>Mac2</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td><strong>Ordinary 64-bit pixel formats</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RG32Uint⁸</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td></td>
</tr>
<tr>
<td>RG32Sint</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RG32Float⁸</td>
<td>Write Color Blend</td>
<td>Write Color Blend</td>
<td>Write Color Blend</td>
<td>Write Color Blend</td>
<td>Write Color Blend</td>
<td>Write Color Blend</td>
<td>Write Color Blend</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RGBA16Unorm</td>
<td>Filter</td>
<td>Write Color</td>
<td>Filter</td>
<td>Write Color</td>
<td>Filter</td>
<td>Write Color</td>
<td>Filter</td>
<td>Write Color</td>
<td></td>
</tr>
<tr>
<td>RGBA16Snorm</td>
<td>Write Color MSAA Blend</td>
<td>Filter</td>
<td>Write Color MSAA Blend</td>
<td>Filter</td>
<td>Write Color MSAA Blend</td>
<td>Filter</td>
<td>Write Color MSAA Blend</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RGBA16Uint²</td>
<td>Write Color MSAA</td>
<td>Write Color</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>Write Color MSAA</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RGBA16Sint²</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>RGBA16Float²</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td><strong>Ordinary 128-bit pixel formats</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RGBA32Uint²</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td></td>
</tr>
<tr>
<td>RGBA32Sint²</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RGBA32Float²,⁶</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>Write Color</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td><strong>Compressed pixel formats</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PVRTC pixel formats³</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter Sparse</td>
<td>Filter Sparse</td>
<td>Filter Sparse</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td>EAC/ETC pixel formats</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter Sparse</td>
<td>Filter Sparse</td>
<td>Filter Sparse</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td>ASTC pixel formats</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter Sparse</td>
<td>Filter Sparse</td>
<td>Filter Sparse</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td>HDR ASTC pixel formats</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Filter Sparse</td>
<td>Filter Sparse</td>
<td>Filter Sparse</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td>BC pixel formats</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Filter Sparse</td>
<td>Filter</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td>GPU family¹</td>
<td>Apple2</td>
<td>Apple3</td>
<td>Apple4</td>
<td>Apple5</td>
<td>Apple6</td>
<td>Apple7</td>
<td>Apple8</td>
<td>Apple9</td>
<td>Mac2</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td><strong>YUV pixel formats</strong>⁶,⁷</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
<td>Filter</td>
</tr>
<tr>
<td>GBGR422</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGRG422</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Depth and stencil pixel formats</strong>⁷</td>
<td>Filter</td>
<td>MSAA</td>
<td>Filter</td>
<td>MSAA</td>
<td>Filter</td>
<td>MSAA</td>
<td>Filter</td>
<td>MSAA</td>
<td>Filter</td>
</tr>
<tr>
<td>Depth16Unorm</td>
<td>Filter</td>
<td>MSAA</td>
<td>Filter</td>
<td>MSAA</td>
<td>Filter</td>
<td>MSAA</td>
<td>Filter</td>
<td>MSAA</td>
<td>Filter</td>
</tr>
<tr>
<td>Depth32Float</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
</tr>
<tr>
<td>Stencil18</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
</tr>
<tr>
<td>Depth24Unorm_Stencil8⁵</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Filter MSAA Resolve</td>
</tr>
<tr>
<td>Depth32Float_Stencil8</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>Filter MSAA Resolve</td>
</tr>
<tr>
<td>X24_Stencil8</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>Not available</td>
<td>MSAA</td>
</tr>
<tr>
<td>X32_Stencil8</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
<td>MSAA</td>
</tr>
<tr>
<td><strong>Extended range and wide color pixel formats</strong></td>
<td>Not available</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>Not available</td>
</tr>
<tr>
<td>BGRA10_XR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGRA10_XR_sRGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGR10_XR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGR10_XR_sRGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. See [MTLGPUFamily](#) for each GPU family's enumeration constant.
2. 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.
3. Only the GPUs in Apple 3 and Apple 4 families support [MTLSamplerAddressMode.clampToZero](#) for the PVRTC pixel formats.
4. The GPUs in Apple 6 through Apple 9 families don't support sparse textures with YUV pixel formats.
5. Some GPUs support [MTLTextureDescriptor.width24Stencil18PixelFormatSupported](#) You can check an individual GPU’s support for this pixel format by inspecting its [MTLDevice.isDepth24Stencil18PixelFormatSupported](#) property at runtime.
6. Some GPUs in the Apple 7 and Apple 8 families additionally support Filter and Resolve (and therefore, All) 32-bit float capabilities. You can check whether a GPU supports 32-bit float Filter and Resolve by inspecting the [MTLDevice.supports32BitFloatFiltering](#) property at runtime.
7. Formats in this group aren't compatible with lossy texture compression through [MTLTextureDescriptor.compressionType](#).
8. Some GPU devices in the Apple 7 and Apple 8 families support filtering and sparse BC compressed textures on iPadOS. You can check whether a GPU supports BC texture compression by inspecting its [MTLDevice.supportsBCTextureCompression](#) property at runtime.
9. The A8Unorm pixel format is incompatible with imageblocks with explicit layout. Use either an R8Unorm texture view, or imageblocks with implicit layout.
10. You can only apply the RG32Uint format to a ulong texture on a GPU that supports the “64-bit atomics” feature.
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 buffer with 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**: 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.

<table>
<thead>
<tr>
<th>Ordinary 8-bit pixel formats</th>
<th>Format</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>A8Unorm</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>R8Unorm</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>R8Snorm</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>R8Uint and R8Sint</td>
<td>All</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ordinary 16-bit pixel formats</th>
<th>Format</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>R16Unorm and R16Snorm</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>R16Uint and R16Sint</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>R16Float</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>RG8Unorm and RG8Snorm</td>
<td>Read/Write</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ordinary 32-bit pixel formats</th>
<th>Format</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>R32Uint and R32Sint</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>RG16Uint and RG16Sint</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>RG16Float and RG16Snorm</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>RGBA8Unorm</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>RGBA8Snorm</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>RGBA8Uint and RGBA8Sint</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>BGR8Unorm</td>
<td>Read</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ordinary 64-bit pixel formats</th>
<th>Format</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>RG32Uint and RG32Sint</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>RGBA16Unorm and RGBA16Snorm</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>RGBA16Uint and RGBA16Sint</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>RGBA16Float</td>
<td>All</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ordinary 128-bit pixel formats</th>
<th>Format</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGBA32Uint and RGBA32Sint</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>RGBA32Float</td>
<td>All</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Packed 32-bit pixel formats</th>
<th>Format</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGBA10A2Unorm</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>RGBA10A2Uint</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>RG11B10Float</td>
<td>Read/Write</td>
<td></td>
</tr>
</tbody>
</table>

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.depth24Stencil8PixelFormatSupported` property at runtime.

3. GPUs that support texture atomics (see feature availability by GPU family) also support atomics in read/write texture buffers with this pixel format.