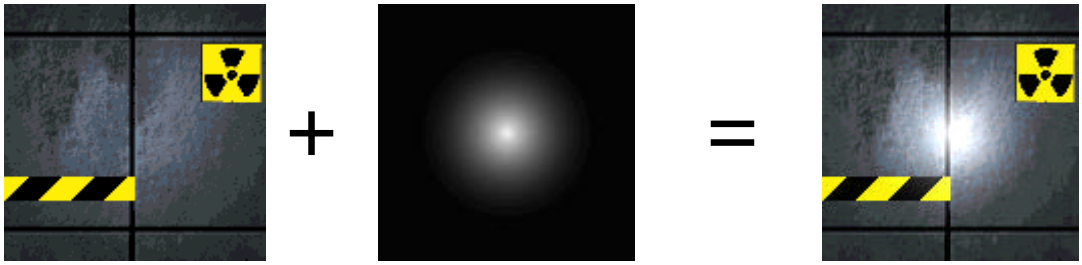
8 Layer Multi-Texturing

REALISTIC ENVIRONMENTS WITH KYRO'S 8 - LAYER MULTI-TEXTURING

KYRO fully supports 8-layer multitexturing, significantly improving on the dual, or triple texturing capabilities of current generation traditional accelerators. With its support of multiple texture layers KYRO allows more detailed backgrounds, realistically lit models and more plausible environments to be created. When compared with traditional accelerators, KYRO's 8-layer multitexturing results in significant performance and quality improvements

Multi-texturing is the action of combining different textures together in order to obtain more realistic environments. One example might be a light map can be added to a texture (base map) to make it more realistic. Other examples of multi-texturing effects include a shadow map, a light map, a bump map, an exclusion map, or a fog map.

Multi-Texturing Example - Figure 1 shows how textures combine together to result in a more detailed map.



Multi-Texturing Example - Figure 1

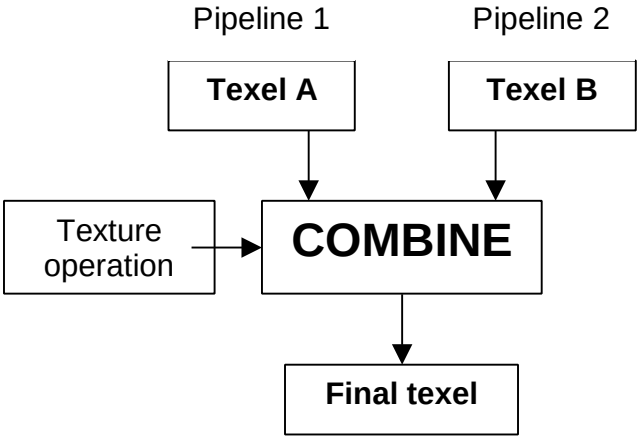
DirectX Multi-Texturing

DirectX supports a maximum of 8 multi-texturing layers at once. This means that up to 8 textures can be combined together to form a more detailed and realistic texture

Traditional 3D Accelerators and Multi-Texturing

Traditional hardware renderers may include two or three parallel texture pipelines in their design in order to be able to process two or three textures at once. This restricts their multi-texturing ability to the number of texture pipelines supported in hardware.

Traditional 2 Pipeline Multi-Texturing - Figure 2 shows a dual-pipeline on a traditional hardware renderer.



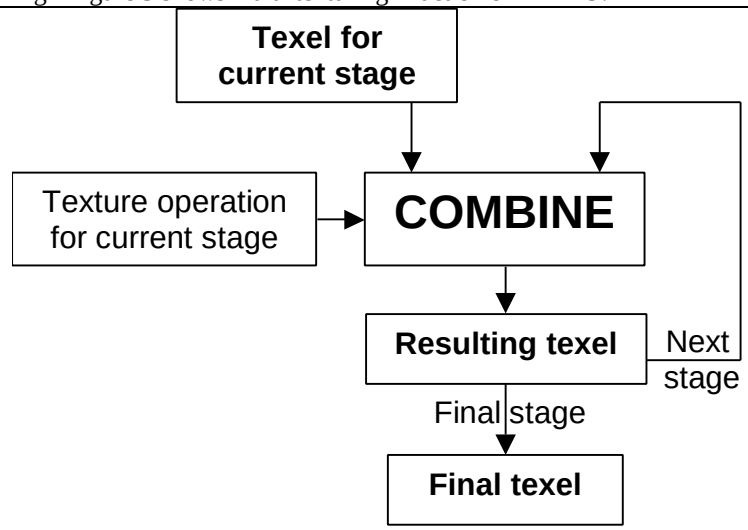
Traditional 2 Pipeline Multi-Texturing - Figure 2

When the hardware doesn't support the required number of texture stages, the only solution is to perform multi-pass rendering. Multi-pass rendering involves re-sending the exact same primitive to be blended with the background.

KYRO Multi-Texturing

KYRO supports 8-layer multitexturing. KYRO is a tile-based renderer, which means that each tile that contains a multi-textured triangle can build up the multi-texture image in a local tile buffer, eliminating the need for a separate pipeline for each texture layer. Each texture is processed in turn until the final image is created.

KYRO Multi-Texturing - Figure 3 shows multi-texturing in action on KYRO.



KYRO Multi-Texturing - Figure 3

Benefits of KYRO Multi-Texturing

8-layer multi-texture support results in numerous advantages when compared to traditional hardware. In any scene, if the desired number of texture layers exceeds the number supported by the hardware, multi-pass rendering has to be used. For instance, if five texture layers are required to render a given primitive on hardware that supports only two textures, three passes will be necessary. Supporting 8 layers completely eliminates the need for multi-pass rendering.

The following sections explain the advantages of KYRO's 8-layer multi-texturing compared to multi-pass rendering on traditional accelerators.

Performance Increase due to Reduced Bus and CPU Loading

Multi-pass rendering on traditional hardwares will require the polygon data to be sent to the HW several times. KYRO's multi-texturing only requires the polygon data to be sent once. As a result system bus bandwidth and CPU loading are much lower with KYRO's multi-texture hardware

Performance Increase due to Reduce Z-buffer and Framebuffer Access

Multi-pass rendering on traditional 3D accelerators involves blending the current primitive onto the previous primitive contained in the Z-buffer and framebuffer. Additional Z-Buffer Read/Write and Framebuffer Read/Write memory accesses are necessary, reducing performance and consuming fill rate. KYRO's multi-texturing support renders interim stages into the internal tile buffer eliminating the need for external memory accesses until the tile is complete and ready to be written to the framebuffer.

Quality Increase due to Internal True Color™

The Framebuffer bit depth affects all multi-pass blending operations on traditional hardware. When a 16-bit Framebuffer is used, color precision is reduced with each blending pass. All of KYRO's multi-texturing operations take place in 32-bit, with a result that no loss of precision occurs between texture stages and only the final 32-bit color values are converted to the 16-bit Framebuffer. See the KYRO whitepaper on Internal True Color for more details.

Special Effects using Destination Alpha

If a transparent texture is part of a multi-texture stage and destination blending is required, then it becomes impossible to emulate the resulting texture using traditional multi-pass rendering, unless the framebuffer format contains an alpha channel (typically 8888 in a 32-bit framebuffer). KYRO's multitexture hardware means that these effects are available in all framebuffer depths.