3D Architecture White Paper

Introduction

The RADEON™ X800 series of Visual Processing Units are the most advanced graphics processors ever created. Comprised of 160 million transistors, they build upon the highly acclaimed RADEON 9800 architecture and extend it by providing major increases in bandwidth, parallelism, efficiency, image quality, and programmability. The result is a series of products that defines a new era of High Definition graphics. The performance of the new architecture is staggering, doubling anything on the market today.

	RADEON 9800 PRO	RADEON 9800 XT	RADEON X800 PRO	RADEON X800 XT
Transistors	110 million	110 million	**160 million**	**160 million**
Clock Speeds (engine/memory, MHz)	380e/680m	412e/730m	**475e/900m**	**500+e/1000+m**
Memory Bandwidth	21.8 GB/sec	23.4 GB/sec	**28.8 GB/sec**	**32+ GB/sec**
Vertex Pipelines	4	4	**6**	**6**
Pixel Pipelines	8	8	**12**	**16**
Transform Rate	380 million vertices/sec	412 million vertices/sec	**712 million vertices/sec**	**750+ million vertices/sec**
Pixel Fill Rate	3.0 GigaPixels/sec	3.3 GigaPixels/sec	**5.7 GigaPixels/sec**	**8+ GigaPixels/sec**
Anti-Aliased Fill Rate	18.2 billion AA samples/sec	19.8 billion AA samples/sec	**34.2 billion AA samples/sec**	**48+ billion AA samples/sec**
Shader Performance	16 billion instructions/sec	17.3 billion instructions/sec	**34.2 billion instructions/sec**	**46+ billion instructions/sec**

Table 1: RADEON X800 Series Performance Statistics

The RADEON X800 architecture is designed to deliver a major leap in performance over existing graphics processors, while inheriting all of the strengths of its successful predecessors. The technology was extended and scaled to achieve the goal of twice the frame rate performance of the previous flagship VPU, without significant increases in power consumption, heat generation, or cost.

This required the design to be made as lean and efficient as possible, using the latest manufacturing technologies to maximize performance per transistor. In fact, the RADEON X800 XT is capable of processing approximately 100 times as many floating point operations per clock cycle as an Intel® Pentium® 4 3.0 GHz CPU. A range of innovative techniques are employed to make the most efficient possible use of precious memory bandwidth, to optimize performance at high display resolutions, and to improve image quality.

Another important characteristic of the RADEON X800 architecture is the ease with which applications can take full advantage of its capabilities. Its feature set and architecture are designed to avoid exposing developers to potential performance pitfalls, thus simplifying the task of performance optimization. This allows more time to be spent on adding new features to games instead of tracking down and working around performance issues.

This document provides an overview of the RADEON X800 architecture and its key features.
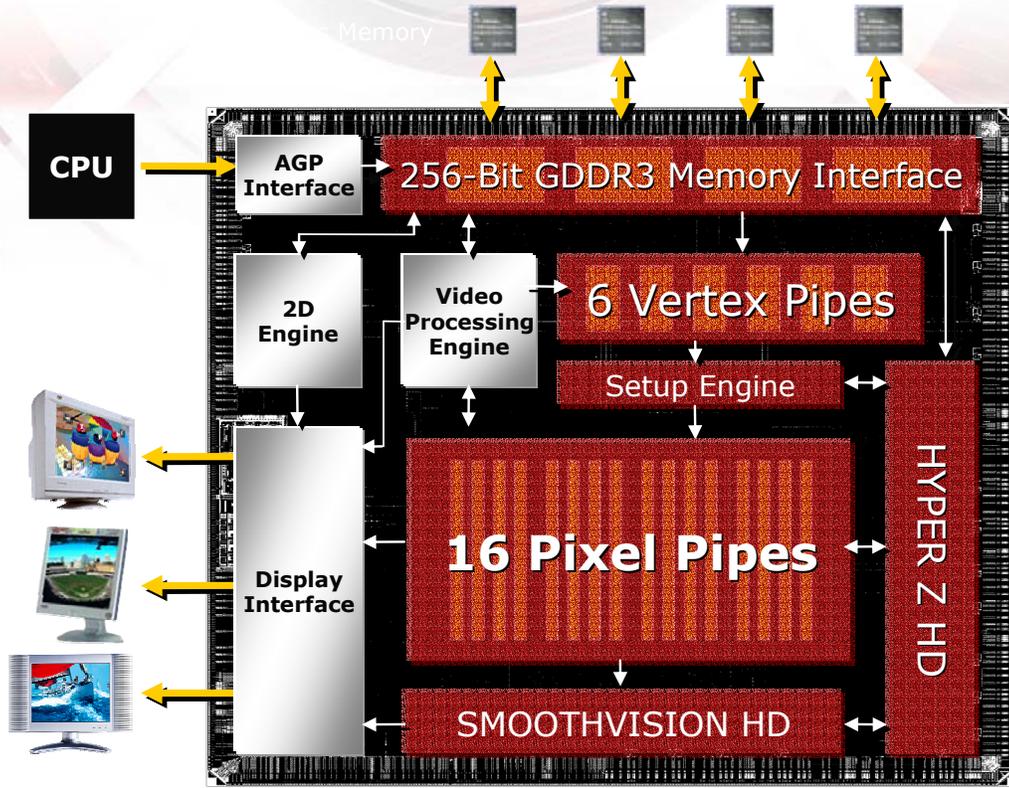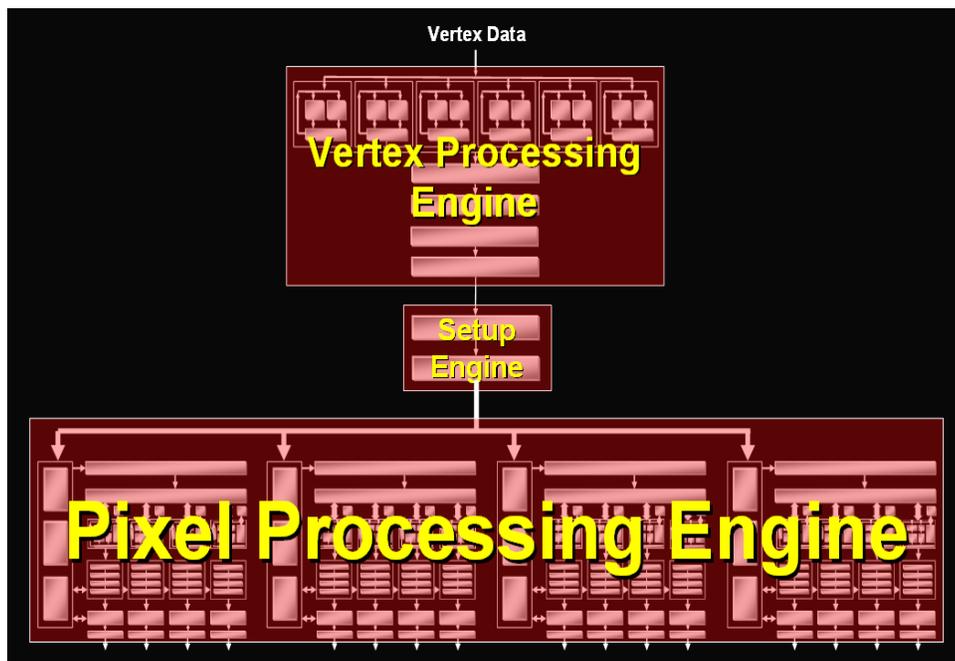
Figure 1: RADEON X800 Simplified Block Diagram

Figure 2: RADEON X800 3D Core

Silicon Technology

The RADEON X800 series VPUs are manufactured by Taiwan Semiconductor Manufacturing Co. (TSMC), using their latest 0.13 micron low-k copper fabrication process. This process allows about 33% more transistors per unit area than the 0.15 micron technology used for the RADEON 9800 series. Furthermore, it allows the transistors to run over 100 MHz faster with no increase in power consumption or heat generation.

One of the key technologies responsible for these improvements is the low-k Black Diamond dielectric material used by TSMC. This refers to the insulating material that separates the conducting wires on the chips. Any two signal-carrying conductors separated by an insulator generate capacitance, a phenomenon which can degrade signal quality and increase power consumption. Different insulating materials exhibit different levels of capacitance, and this characteristic value is referred to as k. Thus low-k materials are highly beneficial for high performance chip design.

The use of copper interconnects is also partly responsible for the improved performance of the RADEON X800 series. Compared with the aluminum interconnects used in the 0.15 micron process, copper has significantly lower resistance. This means electrical signals can travel faster, and less energy is wasted as heat.

Another RADEON X800 manufacturing technology innovation is its highly scalable design. The 3D rendering core is divided into four independent blocks of four pixel pipelines each. If a manufacturing defect occurs in one or more of these blocks, the affected blocks can be disabled without impacting the rest of the chip. When rendering a 3D scene, the pixel processing workload is efficiently divided among all of the functional pipelines.

This capability means that the X800 architecture can operate in 4, 8, 12, and 16 pipeline configurations, with performance scaling accordingly. Since the 3D core takes up a majority of the RADEON X800 die area, a substantial proportion of the dies that would normally have to be discarded can be recovered.

Memory Interface

The RADEON X800 VPU incorporates a new high-performance 256-bit GDDR3 memory interface, capable of providing over 32 GB/sec of graphics memory bandwidth. It includes four independent 64-bit memory channels, each of which can be simultaneously writing data to memory, or reading data back into the graphics processor. Sophisticated sequencer logic ensures that all four channels are being utilized for maximum efficiency.

GDDR3 memory is designed to operate at higher clock speeds than DDR & DDR2 SDRAM memories, while consuming less power. In order to operate at increasing frequencies, graphics DRAM typically requires termination circuitry to prevent signal degradation at the I/O interface. This circuitry can add significant cost and takes up valuable space on the graphics card. GDDR3 memory deals with these issues by integrating the termination circuitry into the memory devices themselves, thus simplifying board design and reducing cost.
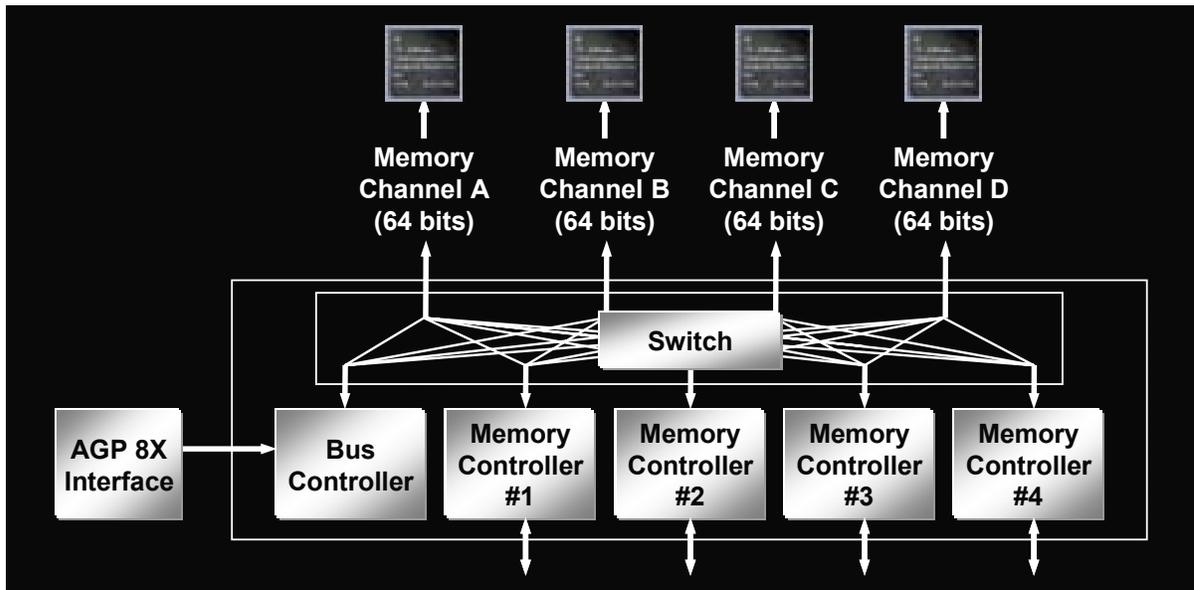

Figure 3: RADEON X800 Memory Interface

Vertex Processing Engine

A 3D scene is composed of interlocking groups of triangles that make up all visible surfaces. By performing mathematical operations on the vertices at the corners of each triangle, the vertex processing engine can place, orient, animate, color, and light every object and surface that needs to be drawn. The process is controlled by small programs called vertex shaders that are uploaded to the graphics chip and executed by the vertex processing engine.
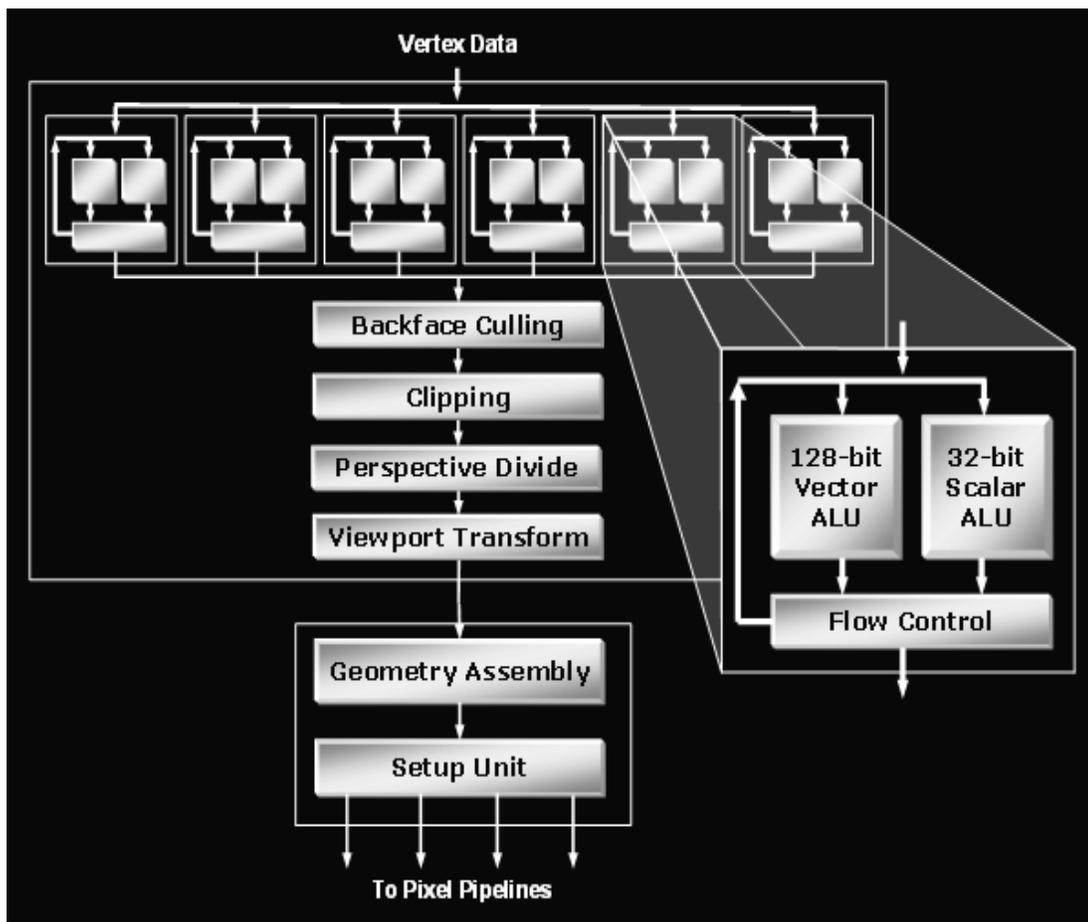

Figure 4: RADEON X800 Vertex Processing Engine and Setup Engine

The RADEON X800 vertex processing engine consists of six programmable vertex processing units, plus a series of fixed function stages. Each vertex processing unit actually includes two Arithmetic Logic Units (ALUs): a 128-bit floating point vector ALU plus a 32-bit floating point scalar ALU. There is also flow control circuitry that allows loops, subroutines, and branches to be executed.

The data received by the vertex processing units includes three-dimensional position information, texture co-ordinates, normal vectors, colors, and material information. The most basic function of these units is to perform transformation operations on the position data, which typically consists of

four 32-bit values for each vertex (*x*, *y* & *z* co-ordinates, plus a *w* co-ordinate used for calculating perspective). This type of transformation requires a 4x4 matrix multiplication, which can be broken down into four dot product instructions. Each 128-bit vector ALU in the RADEON X800 is designed to execute one of these dot product instructions every clock cycle. Since there are six of these ALUs, a total of 1.5 vertex transformations can be processed every clock cycle.

The fixed function portion of the vertex processing engine handles vertex data after any vertex shader programs have completed. First, backface culling removes any triangles that are facing away from the viewpoint, since they will not be visible in the final image. Next, the clipping stage detects triangles that lie partly outside of the viewing window (or any other application-defined area), and discards the invisible portions of these triangles. The persepective divide stage modifies vertices to account for perspective in order to create a better sense of depth, and the viewport transform stage projects the 3D polygons on to a flat 2D display area.

Setup Engine

Once all necessary vertex processing has been performed, the vertices must be re-assembled into triangles. This step is accomplished by the Setup Engine.

The first part of the Setup Engine is called Geometry Assembly. This is where groups of three vertices are connected to form triangles. An index buffer contains lists of which vertices belong to each polygon. The Geometry Assembly unit also handles point sprites (also known as particles), which are defined by just a single vertex. These are converted into rectangles consisting of two triangles joined together, centered on the original vertex.

The Setup Unit is responsible for assigning any required parameters to the newly assembled triangles, such as texture co-ordinates, color values, and Z-buffer information. It is also responsible for dividing the triangles into tiles, and then distributing these tiles to the pixel pipelines to continue the rendering process. The RADEON X800 setup engine has been enhanced to handle the efficient distribution of workload among all functional pipelines in the chip. It also adds support for varying rather than fixed tile sizes, allowing the graphics driver to choose optimal tile size for any given situation.
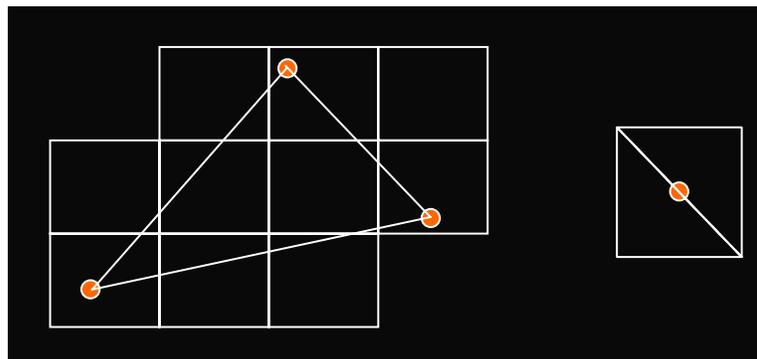

Figure 5: A triangle created from three vertices and split into multiple tiles (left). Each tile can be sent to a different quad pipeline for rendering. On the right is a point sprite or particle created from a single vertex.

Pixel Processing Engine

Once each of the triangles that make up a 3D scene have been arranged and lit as necessary, the next step is to fill them in with individual pixels. The color of each pixel is determined by the textures that are applied, the lighting conditions, and the material properties assigned to the triangle. This process is controlled by pixel shaders, which are small programs that are uploaded to the graphics chip and executed by the pixel processing engine.
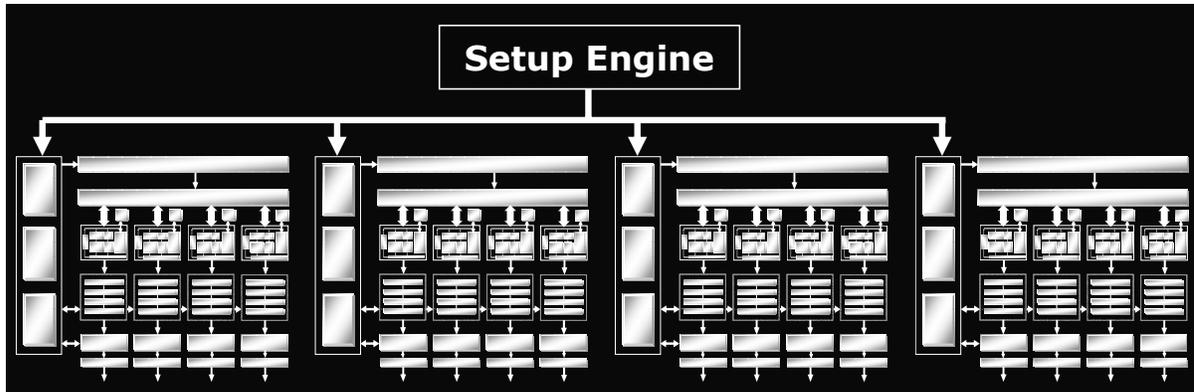

Figure 6: RADEON X800 Pixel Processing Engine, with four independent quad pipelines

The RADEON X800 architecture includes sixteen parallel floating point pixel processing pipelines, each with its own pixel shader unit and texture unit. These pipelines are grouped into four groups of four, each referred to as a quad pipeline. Each quad pipeline is an independent unit with its own dedicated resources. Different members of the RADEON X800 product family have different numbers of these quad pixel pipelines enabled.
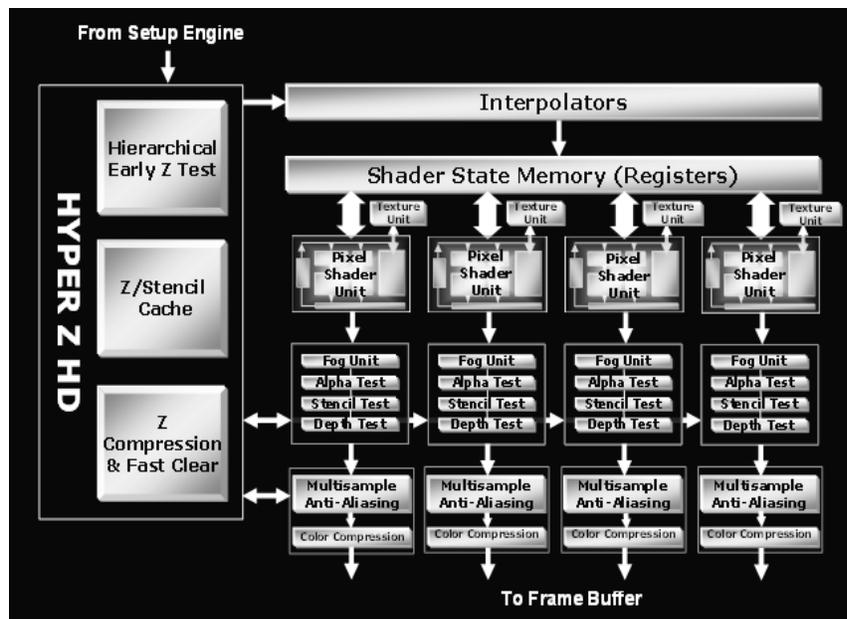

Figure 7: Detail of a quad pipeline

The Setup Engine passes each quad pipeline a tile containing part of the current triangle being rendered. The first task at this stage is to determine which pixels in the tile will be visible, and which will be hidden behind other existing triangles. This is handled by the Hierarchical Z unit of the HYPER Z HD block, which subdivides the Z-Buffer data for the current tile into blocks of pixels. These blocks can then be rapidly checked to see if they will be visible in the final image. If an entire block will be hidden, it is discarded and the renderer moves on to the next block. If some portions of the block will be visible, it is subdivided into smaller blocks and the process is repeated.

While Hierarchical Z can usually catch and discard most of the occluded pixels in an image, the fact that it operates on blocks of pixels means that some will always be able to slip through. These pixels must then pass all the way through the rendering engine before a final Z test is performed and they are discarded. The Early Z Test feature of HYPER Z HD is capable of quickly testing the visibility of individual pixels before they are sent to the rendering engine, effectively eliminating overdraw and maximizing rendering efficiency.
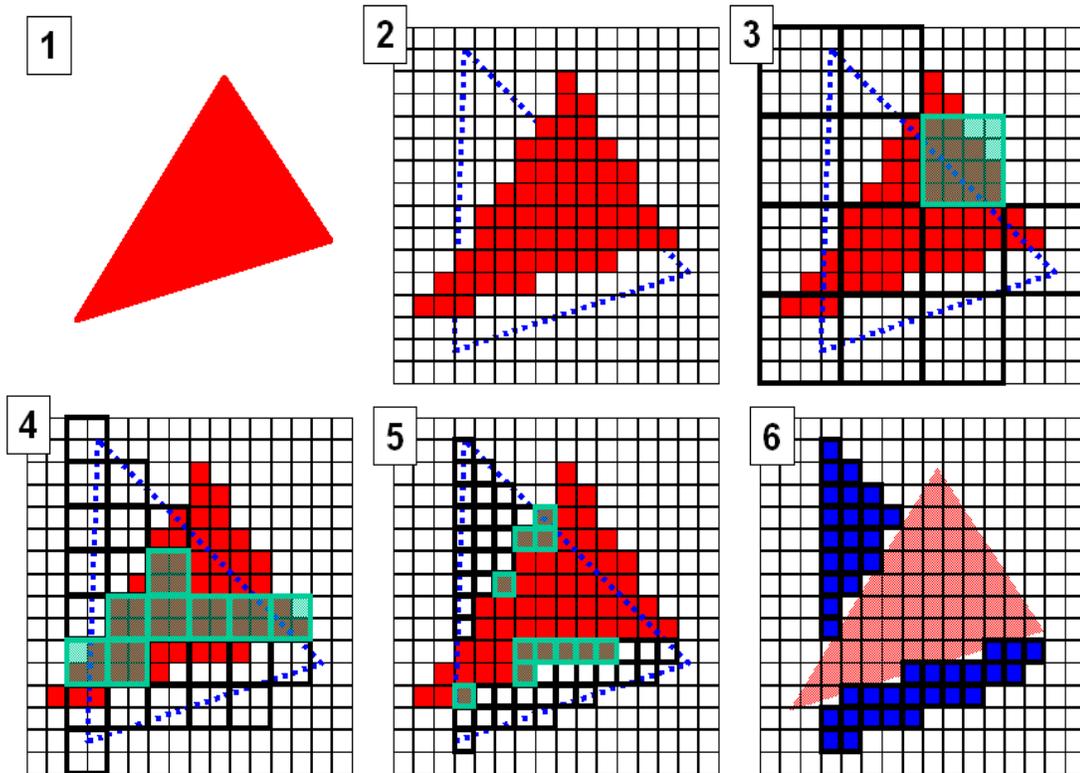

Figure 8: HYPER Z HD Hierarchical Z and Early Z Test. The red triangle represents an existing polygon, with a new polygon outlined in blue being drawn behind it. The green boxes represent areas where the blue polygon is fully occluded, and thus these pixels

Once the visible pixels have been determined, the next step is to assign them initial values for basic parameters such as color, depth, transparency (alpha), and texture co-ordinates. The initial values are determined by looking at the values assigned to each vertex of the current triangle, and interpolating them to the location of the current pixel. The interpolated values are stored in registers that are used by the pixel shader units.
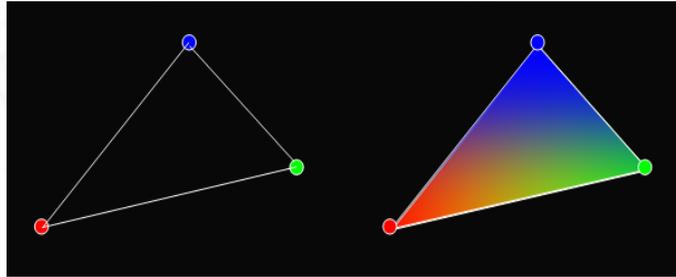
Figure 9: A triangle with initial color values at each vertex (left), and with interpolated color values (right)

Each pixel shader unit in the RADEON X800 actually consists of five distinct ALUs: two 72-bit floating point vector ALUs, two 24-bit floating point scalar ALUs, and a 96-bit texture address ALU. These ALUs access a high speed datapath to shader state memory, which consists of 32 temporary registers, 32 constant registers, a few special purpose registers, and a set of interpolated values for color, position, texture co-ordinates, etc. There is also an output combiner as well as F-buffer logic for handling multi-pass shaders.
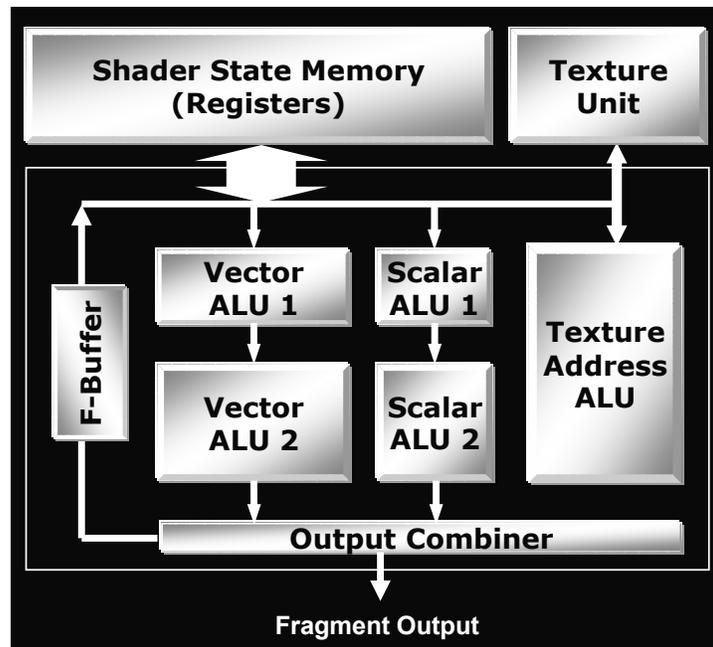

Figure 10: Detail of a RADEON X800 Pixel Shader Unit

The design of the pixel shader units is optimized for handling color data in RGBA (Red/Green/Blue/Alpha) format, although it can handle a wide range of alternative data formats. The vector ALUs perform operations on the three color components (RGB), while the scalar ALUs perform operations on the alpha or transparency component (A). The separate texture address ALU allows texture accesses to occur independently of other pixel shader operations. All operations are processed with a full 24 bits of precision for each component. The RADEON X800 driver software ensures optimal utilization of these ALUs by analyzing and reordering incoming pixel shader instructions wherever possible, without affecting the output.

The texture unit of each pixel pipeline can sample up to 16 textures in a single rendering pass. These textures can be one-, two-, or three-dimensional, with bilinear, trilinear, or anisotropic filtering applied depending on the desired quality level. Textures can be sampled with up to 32 bits of floating point precision per component. The texture units also support automatic decompression of DXTC/S3TC and 3Dc compressed texture formats.

The F-buffer is a portion of graphics memory reserved to store data for a series of pixels as soon as they are output from the pixel shader unit. These pixels can then be read back directly into the pixel shader and undergo further processing without having to go through all of the other stages of the rendering pipeline. This allows multi-pass shaders to execute much more efficiently than would otherwise be possible.

A key advantage of the RADEON X800 pixel shader architecture over competing technologies is its ability to maintain high performance regardless of the amount of resources used by a shader. There are no significant penalties for using large numbers of temporary registers, using a high proportion of texture instructions vs. math instructions, or using full floating point precision. This allows developers to get predictable performance out of their code, and reduces debugging time.
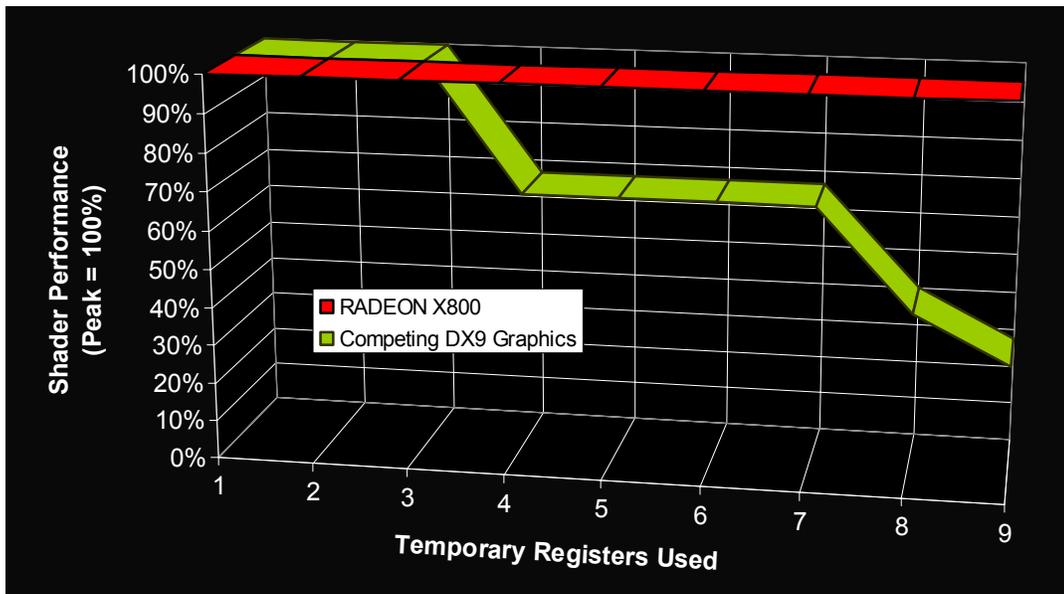

Figure 11: An example of a performance pitfall that can occur in some competing pixel shader architectures. Note that the RADEON X800 pixel shader units maintain peak performance in all cases.

Once pixel shading is complete, pixel data is passed on to the rasterization portion of the rendering pipeline. Here it has fog values blended in before undergoing a series of visibility tests including alpha (transparency), stencil (commonly used for shadow volumes) and depth or Z (determines if pixel is occluded). Each pixel pipeline can normally read each of these values, perform a comparison, and write back a modified to value to memory for one pixel every clock cycle. If multi-sample anti-aliasing is enabled, however, the number of stencil and Z operations is doubled to two per clock cycle.
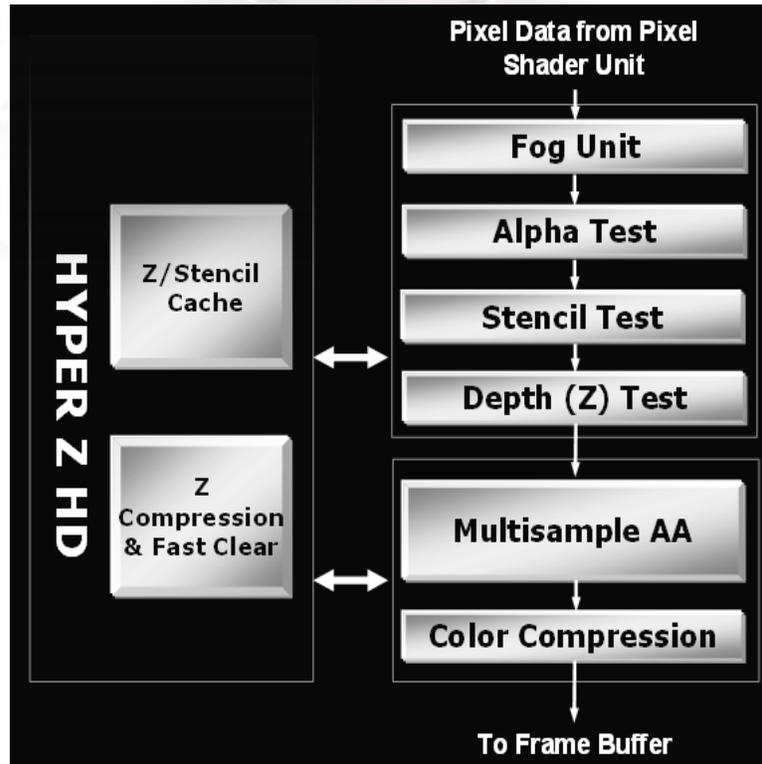
Figure 12: Detail of the rasterization and anti-aliasing portion of a pixel pipeline

The RADEON X800 multi-sample anti-aliasing unit can perform Z tests at 2, 4, or 6 different locations per pixel to determine what proportion is covered by the current triangle. The sample locations are read from a programmable lookup table, and can be varied from frame to frame. Color values are calculated only once per pixel, but up to 6 different colors can be stored for each pixel to handle cases where multiple triangles intersect and overlap. To accommodate the varying number of possible colors stored for each pixel, a special compressed frame buffer format is used. This allows color compression of up to 6:1 in the typical case where most pixels require only a single color value to be stored.

When a frame is complete, the color values for each pixel are blended to produce the final output color. This blending is done with gamma correction, to ensure that gradients along anti-aliased edges appear smooth when displayed on the screen. The results can be written to one of up to four different render targets for use in subsequent rendering passes, or sent to the display engine for output.

Summary

The RADEON X800 3D architecture sets new standards for computing power and flexibility with massive parallelism, unprecedented efficiency, and a high degree of scalability. By creating major discontinuities in pixel fill rate, vertex processing, shader processing, and memory bandwidth performance, it is able to deliver real application performance up to double the previous industry leader at maximum image quality settings. This in turn makes it possible to introduce new visual effects that were never before possible in real time.
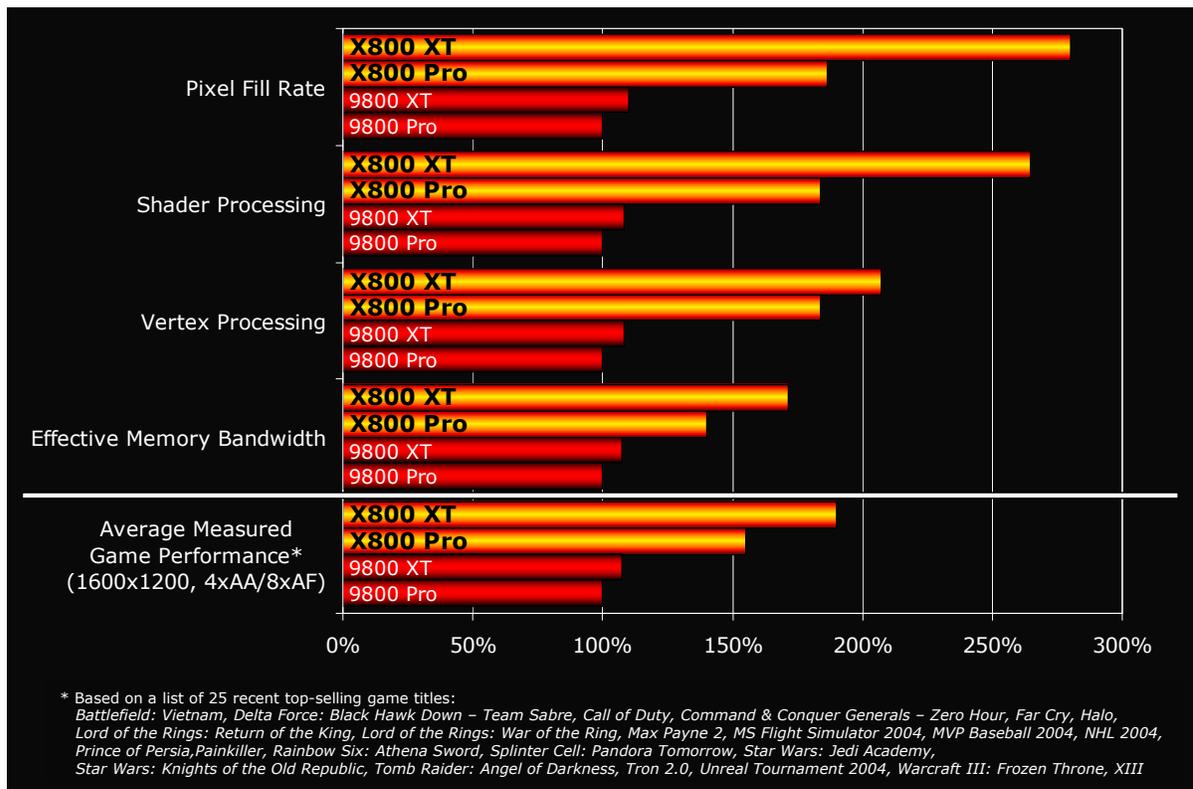

Figure 13: Theoretical and measured performance figures for the RADEON X800 series VPUs compared with the previous industry performance leaders, the RADEON 9800 series.