

Trending

- Get your wallets ready for the return of CEE to Suntec Singapore at the end of the week
- Single8 joins the fray in offering 5Gbps broadband to the home
- Beat the heat: Affordable alternatives to air-con units to keep you cool

Feature Articles

NVIDIA GeForce 8800 GTX / GTS (G80) - The World's First DX10 GPU

By Vijay Anand - 9 Nov 2006

Page 2 of 21 - Embracing a Unified Shader Architecture

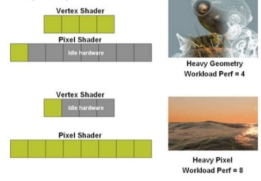
Embracing a Unified Shader Architecture

One of the most prominently sighted improvements in next generation graphics engines was the unification of the traditional fragmented and linear shader processing units in the GPU. On conventional GPU pipelines of the DirectX 8/9 era, which practically encompasses all modern graphics engines, there are two major shader units that take the brunt of most of the initial processing workload - the vertex shader and the pixel shader. The vertex shader manipulates vertices information which then passes the data to the triangle setup engine to actually map them in a 3D space (forming the 3D object). The pixel shader is the next stage in line that actually fills up the triangle fragments that make up the 3D object with individual colored pixels that have undergone complex computations such as lighting models or other complex transformation effects to derive the final pixel color applied. So far, both the vertex and pixel shaders have had different hardware feature sets and consequently work on distinct instruction sets.

Increasingly in recent times with game developers embracing the programmable nature of Shader Models (SM) version 2 and then version 3, games became more and more complex, requiring greater processing power. To meet those demands GPU designs have been increasing the rendering pipelines available, which normally encompass the vertex and pixel shader units. The number of these shader units possessed in a GPU is a really big deal and plays a heavy role (among others) in defining whether a GPU is passed off as a low, mid or high-end model. In fact, ATI was so certain that pixel shaders was going to play an even stronger role than ever before when complex shader programs were embraced in a big way that their Radeon XTK series of GPUs were heavily skewed toward pixel shader processing power. Thus with the rise of programmable graphics pipeline, these shader units play a major role as pinpointed above and occupy significant portion of the GPU die. Herein lies an issue. With the increasing number of shader units packed into the GPU for vertex shader and pixel shader functionality, there's bound to be many a time when a number of these resources go idle.

Depicted below by NVIDIA are two directly opposite workloads run on a fictitious traditional GPU to better understand the situation at hand. The first scene has a very complex 3D mesh (naturally requiring thousands of vertices), resulting in a very high polygon count. This is a case where the vertex shader is highly used to build the scene whereas the pixel shader is sparingly used, at least while the scene is being constructed. This could take a while as performance is being bottlenecked with the few vertex units available. The scene below that with the water body looks realistic and nice, but if you switch to the wire frame view, it would look sparse and simple with a very low polygon count as the pixel shader performs complex calculations to render this scene accurately, carefully coloring each pixel. It could have several pixel shader programs operating simultaneously to handle the water, waves, sky, lighting, refraction and many more. Thus, here we have little use of the vertex shader units but we could use more resources on the pixel shader front. From these two extreme scenarios (which aren't far-fetched in reality), you can see that the traditional architecture with a fixed set of vertex and pixel shader units are highly inefficient at times, leading to performance bottlenecks. Furthermore as NVIDIA cites, it's not efficient from a power (performance/watt) or die size and cost (performance/sq-millimeter) perspective either.

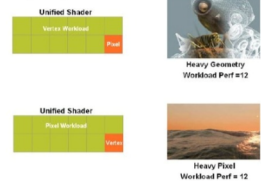
Why unify?



Inefficient use of vertex and pixel shader units in a traditional GPU architecture and its associated 'net performance' in these worst-case workloads.

Thus the proposal of the unified shader architecture which uses an array of shader processors, not specifically tuned for either pixel or vertex, but one that can handle both instruction types and can tackle both tasks simultaneously assigning various processors for each kind of task on the fly. As depicted below, this greatly boosts the efficiency of the shader units and increases overall performance.

Why unify?



Efficient use of all shader processors in a unified architecture and its associated 'net performance' in this ideal scenario.

ATI was the first of the two graphics giants to come forward with the first ever implementation of a GPU with a unified shader architecture and that's the ATI Xenos GPU used on the Xbox 360 console. Despite the number of rumors floated around the Internet of ATI's next generation graphics cards for the PC utilizing the unified shader architecture model, the R600, NVIDIA actually surprised those outside of the development world with the world's first unified shader architecture for the PC industry and it's the first ever to be fully DirectX 10 compliant at that too. In fact when ATI first announced that their next fully ground-up GPU would be taking the unified route, NVIDIA actually played it coy by replying they don't actually see the need to venture that path right now. At least not when things are going fine, so why 'fix' it? Turns out that it was just to play down ATI's announcement while they worked on the secretive G80 project that was destined to be using this unified architecture.

Page 2 of 21 - Embracing a Unified Shader Architecture

Join HWZ's Telegram channel here and catch all the latest tech news!

Our articles may contain affiliate links. If you buy through these links, we may earn a small commission.

Have feedback on the article for the editorial team? You can reach out to them here.