

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- Dave Kirk - NVIDIA Tesla Launch
- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- A speculative look on the Wii U GPU
- Andy Keane Interview & Tesla Adoption/Deployment
- Diving into Anti-Aliasing
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- Ian Buck - NVIDIA Tesla Launch

ATI Xenos: Xbox 360 Graphics Demystified - Page 1

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

Foreword

Those that have followed the development of 3D graphics over the past ten years or so will have seen a continual development of the capabilities of the processors, but fundamentally following the path of OpenGL pipeline model. 3dfx really ignited the market with their "Voodoo Graphics" add-in boards, which were not much more than just a raster engine: it utilised one chip for texture sampling and another for pixel processing (a simple Render Output unit - ROP); 3dfx further evolved that by adding an extra texture unit, allowing for slightly more complex effects in the raster pipeline. And so it was that this model was followed for a number of years with the main developments being the number of pixel pipelines and textures supported per pipeline, until NVIDIA took the step of moving further forward on OpenGL pipeline and giving accelerated support to the Transformation and Lighting process with GeForce 256. Whilst graphics processors had varying degrees of the geometry process, from clipping to setup, handled in hardware, adding a T&L engine was a significant step up the OpenGL pipeline, but didn't really fundamentally change our thinking of graphics processors.

At the same time as the graphics vendors started giving us T&L engines the pixel processors gradually increased in flexibility as well, up until the point that "programmable shader architectures" were all anyone could talk about. The pixel pipelines became more flexible such that they had limited programmability, as did vertex processing, with vertex shaders operating in parallel with T&L engines. Nowadays the level of programmability of both vertex and pixel shaders has increased significantly with each vertex shaders enveloping the T&L processors entirely and pixel shaders consuming the texture processors. However, despite an increasingly important onus being placed on the arrangement and capabilities of the shader Arithmetic Logic Units (ALU's) in this programmable era, the designs of contemporary graphics processors still bear the fundamental similarities to their forebears: vertex processing up one end of the pipeline, pixel processing down the other and still very much aligned with multiples of pixel pipelines.

Conceivably there is no reason why this development model couldn't continue to exist in the PC space and it certainly seems like it will from all vendors for at least the next year. However, ATI have multiple design teams working on different architectures concurrently, so whilst their PC processors may follow a fairly familiar lineage other parts of the company have been talking this shader era with a completely fresh perspective in order to consider the needs of a "Programmable Graphics Processor" and extract as much of the potential of the ALU's as possible by trying to minimise the

AMD1044_0279902
AMD1318_0158231
ATI Ex. 2124
IPR2023-00922
Page 1 of 33

wasted cycles. In doing so they will force us to reconsider how we think of the overall pipeline and make initial performance assessments based upon "pipelines" alone.

Introduction

Ever since the announcement that ATI were working with Microsoft on "Future XBox technologies" the rumour mill has been working overtime as to the graphics behind it. Some of the messages since the announcement of the XBOX 360, the eventual console ATI's work will appear in, have not necessarily been reflective of the actual operation and even a little contradictory from representatives directly from ATI. With strict NDA's and designs being built for two different competitive consoles, very tight controls of what could be talked about had to be implemented within ATI, and the XBOX group operated very much within their own silo; it wasn't until Microsoft lifted the NDA's that ATI could even speak of it on a wider internal basis, let along externally, and even then there is a lot of information to gather.

Since XBOX 360's announcement and ATI's unleashing from the non disclosure agreements we've had the chance to not just chat with Robert Feldstein, VP of Engineering, but also Joe Cox, Director of Engineering overseeing the XBOX graphics design team, and two lead architects of the graphics processor, Clay Taylor and Mark Fowler. Here we hope to accurately impart a slightly deeper understanding of the XBOX 360 graphics processor, how it sits within the system, understand more about its operation as well as give some insights into the capabilities of the processor. Bear in mind that we are under NDA for some of the operational details of the graphics processor to gain an understanding of how it differs from current platforms however some of the specifics won't be revealed in full detail in this article.

Click for a bigger version



Throughout this article we'll attempt to piece together the operation of the graphics processor based on our conversations with ATI and some developers who have already had some knowledge of XBOX 360's capabilities, however we'll also offer some opinions on certain elements. Sections typed in blue indicate Beyond3D's suppositions and have not been directly indicated to us by ATI.

Page Navigation

Page 1 ▼ ▷

Site width adjust

Reset width

Privacy Policy

AMD1318_0158232
ATI Ex. 2124
IPR2023-00922
Page 2 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- A speculative look on the Wii U GPU
- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- Ian Buck - NVIDIA Tesla Launch
- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- Andy Keane Interview & Tesla Adoption/Deployment
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- Dave Kirk - NVIDIA Tesla Launch
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- Diving into Anti-Aliasing

ATI Xenos: Xbox 360 Graphics Demystified - Page 2

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

Xbox 360 System Overview

The "XBOX 360" console was officially unveiled at a show on MTV the week prior to E3 2005, and at the unveiling Microsoft revealed a few technical details of the platform. The primary specifications for the system are:

- **3.2GHz Custom IBM Central Processor**
  - Three CPU Cores
  - Two Threads Per core
  - VMX Unit Per Core
  - 128 VMX Registers Per Thread
  - 1MB L2 Cache (Lockable by Graphics Processor)
- **500MHz Custom ATI Graphics Processor**
  - Unified Shader Core
  - 48 ALU's for Vertex or Pixel Shader processing
  - 16 Filtered & 16 Unfiltered Texture samples per clock
  - 10MB eDRAM Framebuffer
- **512MB System RAM**
  - Unified Memory Architecture (UMA)
  - 128-bit interface
  - 700MHz GDDR3 RAM

Of these core components obviously we are going to be most concerned with the graphics processing element. Whilst the graphics processor is different from others seen before in the PC space, and is very different from even ATI's impending new PC graphics components, it will be interesting to take a look at the graphics processor for the very reason that it doesn't directly correspond to any current graphics processor but also because we feel that this will give hints as to the architectural direction ATI are likely to be taking in the future for PC and other applications.

ATI C1 / Xenos

A name that has long since been mentioned in relation to the graphics behind Xenon (the development name for XBOX 360) is R500. Although this name has appeared from various sources, the actual development name ATI uses for Xenon's graphics is "C1", whilst the more "PR friendly" codename that has surfaced is "Xenos". ATI are probably fairly keen not to use the R500 name as this draws parallels with their upcoming series of PC graphics processors starting with R520, however R520 and Xenos are very

AMD1044_0279904
AMD1318_0158233
ATI Ex. 2124
IPR2023-00922
Page 3 of 33

distinct parts. R520's aim is obviously designed to meet the needs of the PC space and have Shader Model 3.0 capabilities as this is currently the highest DirectX API specification available on the PC, and as such these new parts still have their lineage derived from the R300 core, with discrete Vertex and Pixel Shaders; Xenos, on the other hand, is a custom design specifically built to address the needs and unique characteristics of the game console. ATI had a clean slate with which to design on and no specified API to target. These factors have led to the Unified Shader design, something which ATI have prototyped and tested prior to its eventual implementation (*with the rumoured R400 development ?*) , with capabilities that don't fall within any corresponding API specification. Whilst ostensibly Xenos has been hailed as a Shader Model 3.0 part, its capabilities don't fall directly inline with it and exceed it in some areas giving this more than a whiff of WGF2.0 (Windows Graphics Foundation 2.0 - the new name for DirectX Next / DirectX 10) about it.

Click for a bigger version



Click for a bigger version



The Xenos graphics processor is not a single element, but actually consists of two distinct elements: the graphics core (shader core) and the eDRAM module. The shader core is a 90nm chip manufactured by TSMC and is currently slated to run at 500MHz*, whilst the eDRAM module is another 90nm chip, manufactured by NEC and runs at 500MHz* as well. These two chips both exist side by side, together on a single package, ensuring a fast interlink between the two. The main graphics chip, the parent core, could be considered as a "shader core" as this is one of its primary tasks. The eDRAM module is a separate, daughter chip which contains the elements for reading and writing color, z and stencil and performing all of the alpha blending and z and stencil ops, including the FSAA logic. We'll explore the capabilities and operations of both these chips in greater detail throughout the article.

(*) Note: We understand the clockspeeds for the shader core and daughter die are target clockspeeds at present and there may be some room for small movement either way on both dies dependant on yields. As Microsoft have now announced 500MHz speeds it is more likely that these will be the eventual release speeds.

One element that has been reported on is the number of 150M transistors in relation to the graphics processing elements of Xenon, however according to ATI this is not correct as the shader core itself is comprised from in the order of 232M transistors. It may be that the 150M transistor figure pertains only to the eDRAM module as with 10MB of DRAM, requiring one transistor per bit, 80M transistors will be dedicated to just the memory; when we add the memory control logic, Render Output Controllers (ROP's) and FSAA logic on top of that it may be conceivable to see an extra 70M transistors of logic in the eDRAM module.

AMD1318_0158234
ATI Ex. 2124
IPR2023-00922
Page 4 of 33

Click for a bigger version



Update: *We've recently been given an image of the Xenos graphics chip package (above) that highlights the dual die nature, with the parent die quite clearly to the centre of the package and the daughter over to the left. While the 232M transistor figure for the parent was given to us by ATI we are still trying to establish a more official figure for the daughter (even though these types of transistor counts are very much estimates anyway). We've speculated that the 150M figure that appeared when XBOX 360 was first announced may just relate to daughter die, however another figure that has arisen is 100M - judging from the die sizes the daughter die doesn't have more than half the area of the parent, which would giv indications towards the 100M side although 80M of those transistors are DRAM which may be more dense than the logic circuitry that will dominate the parent die. We are trying to get further clarification.*

One of the mistakes that Microsoft made with the original XBox was to contract their component providers into supplying entire chips with, evidently, no development path - at least, this was the case with NVIDIA NV2A graphics processor, which resulted in Microsoft and NVIDIA going through a legal arbitration process. Although the components in the XBOX 360 in its initial form are hardly low cost, the cost of the unit over the course of its lifetime is one that has quite obviously been addressed with contracts that pay via royalties for chips sold and with Microsoft in charge of ordering the chips from the various Fabs, however the original semiconductor manufacturers are likely to still be in charge of further developments in terms of putting the cores on to smaller processes and we believe that this is part of the contract that ATI has with Microsoft. An obvious area for cost reduction of the Xenos processor is by merging the shader and daughter die on to a single core - we suspect that this will not happen until there is a process shrink available (that can also cater for both the complex logic and eDRAM) as two cores on 90nm mitigate some of the yield risks of a single, large die on 90nm.

Page Navigation

◁ Page 2 ▽ ▷

AMD1044_0279906
AMD1318_0158235
ATI Ex. 2124
IPR2023-00922
Page 5 of 33

Site width adjust

Reset width

Privacy Policy

AMD1318_0158236
ATI Ex. 2124
IPR2023-00922
Page 6 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- A speculative look on the Wii U GPU
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- Diving into Anti-Aliasing
- Ian Buck - NVIDIA Tesla Launch
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- Dave Kirk - NVIDIA Tesla Launch
- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- Andy Keane Interview & Tesla Adoption/Deployment
- AMD's John Bridgman on Radeon, Linux and Open Source 3D

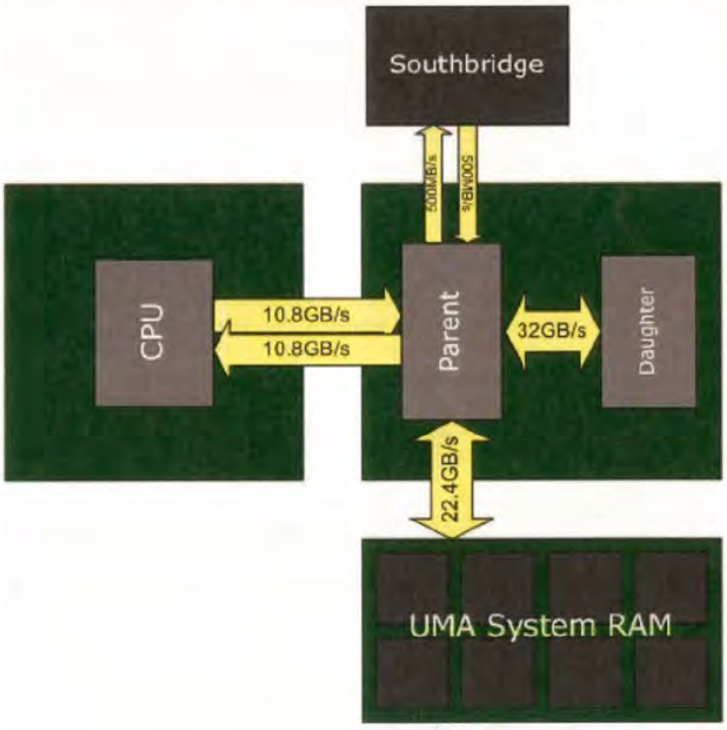
ATI Xenos: Xbox 360 Graphics Demystified - Page 3

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

Bandwidths and Interconnects

When creating a high performance computing platform bandwidth between components and operations is highly important, especially when creating a system that has to last for 3-5 years before a new version comes about, such is the world of consoles. With the Xenos processor being both a high performance graphics processing element of the XBOX 360 as well as the "Northbridge" component of the system, which is essentially the communication hub for the other components of the system, it has many interconnects and bandwidths to deal with. Below is a diagram highlighting the connection bandwidths between the most important elements it is connected to:

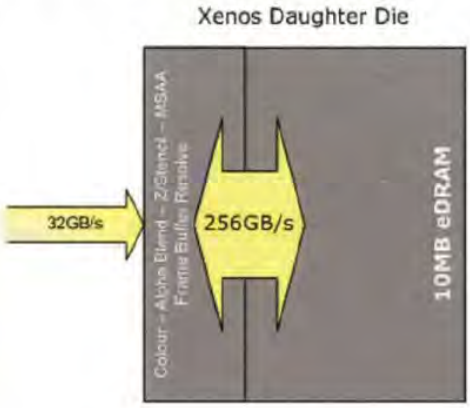
AMD1044_0279908
AMD1318_0158237
ATI Ex. 2124
IPR2023-00922
Page 7 of 33



As we discussed earlier, the XBOX 360 carries a unified memory architecture and Xenos's parent die is acting as the Northbridge controller as well as the graphics processing device. The system memory bandwidth is 22.4GB/s courtesy of the 128-bit GDDR3 memory interface running at 700MHz. At 232M transistors the Xenos parent die isn't an enormous chip so internal memory communication isn't going to be too latency bound, hence the memory interface only needs to be a standard crossbar, which is partitioned into two 64-bit blocks. Xenos's parent die also has a 32GB/s connection to the daughter, eDRAM die Connection to the Southbridge audio and I/O controller is achieved via two PCI Express lanes which results in 500MB/s of both upstream and downstream bandwidth.

As the CPU is going to be using Xenos to handle all its memory transfers, the connection between the two has 10.8GB/s of bandwidth both upstream and downstream simultaneously. Additionally the Xenos graphics processor is able to directly lock the cache of the CPU in order to retrieve data directly from it without it having to go to system memory beforehand. The purpose of this is that one (or more, if wanted) of the three CPU cores could be generating very high levels of geometry that the developer doesn't want to, or can't, preserve in the memory footprints available on the system when in use. High-resolution dynamic geometry such as grass, leaves, hair, particles, water droplets and explosion effects are all examples of one type of scenario that the cache locking may be used in.

AMD1318_0158238
ATI Ex. 2124
IPR2023-00922
Page 8 of 33



The one key area of bandwidth, that has caused a fair quantity of controversy in its inclusion of specifications, is that of bandwidth available from the ROPS to the eDRAM, which stands at 256GB/s. The eDRAM is always going to be the primary location for any of the bandwidth intensive frame buffer operations and so it is specifically designed to remove the frame buffer memory bandwidth bottleneck - additionally, Z and colour access patterns tend not to be particularly optimal for traditional DRAM controllers where they are frequent read/write penalties, so by placing all of these operations in the eDRAM daughter die, aside from the system calls, this leaves the system memory bus free for texture and vertex data fetches which are both read only and are therefore highly efficient. Of course, with 10MB of frame buffer space available this isn't sufficient to fit the entire frame buffer in with 4x FSAA enabled at High Definition resolutions and we'll cover how this is handled later in the article.

Both XBOX 360 and Playstation 3 feature UMA and graphics busses, respectively, that have been announced to use fairly fast 700MHz GDDR3 memory, but both only have a 128-bit interface. Whilst this is less of a surprise for XBOX 360 as Xenos's use of eDRAM will move the vast majority of the frame buffer bandwidth to the EDRAM interface leaving the system memory bandwidth available primarily for texturing bandwidth. It does seem odd that by the time the consoles will be released the likelihood is that high end PC graphics will using at least the same speed RAM but on double wide busses. The primary issue here is, again, one of cost - the lifetimes of a console will be much greater than that of PC graphics and process shrinks are used to reduce the costs of the internal components; 256-bit busses may actually prevent process shrinks beyond a certain level as with the number of pins required to support busses this width could quickly become pad limited as the die size is reduced. 128-bit busses result in far fewer pins than 256-bit busses, thus allowing the chip to shrink to smaller die sizes before becoming pad limited - by this point it is also likely that Xenos's daughter die will have been integrated into the shader core, further reducing the number of pins that are required.

Page Navigation

◁ Page 3 ▾ ▷

AMD1044_0279910
AMD1318_0158239
ATI Ex. 2124
IPR2023-00922
Page 9 of 33

Site width adjust

Reset width

Privacy Policy

AMD1318_0158240
ATI Ex. 2124
IPR2023-00922
Page 10 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- Ian Buck - NVIDIA Tesla Launch
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- Andy Keane Interview & Tesla Adoption/Deployment
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- A speculative look on the Wii U GPU
- Dave Kirk - NVIDIA Tesla Launch
- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- Diving into Anti-Aliasing
- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2

ATI Xenos: Xbox 360 Graphics Demystified - Page 4

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

Pixel and eDRAM Operation

Despite references to 192 processing elements in to the ROP's within the eDRAM we can actually resolve that to equating to 8 pixels writes per cycle, as well as having the capability to double the Z rate when there are no colour operations. However, as the ROP's have been targeted to provide 4x Multi-Sampling FSAA at no penalty this equates to a total capability of 32 colour samples or 64 Z and stencil operations per cycle.

Most PC graphics processors have to balance their output with the available bandwidth and as such their ROP units usually only cater for 2 Multi-Samples per pixel in a single cycle, and the Z output doesn't double with the number of Multi-Samples being produced either. Z and colour compression techniques are also employed in order to get close to the output capabilities with the bandwidth available. ATI's calculations lead to a colour and z bandwidth demand of around 26-134GB/s at 8 pixels with 4x Multi-Sampling AA enabled at High Definition TV resolutions. The lower end of that bandwidth figure is derived from having 4:1 colour and Z compression, however the lossless compression techniques are only optimal when there are no triangle edges intersecting a pixel, but with the presumed high geometry detail within a next generation console titles the opportunities for achieving this compression ratio across the entire frame will be reduced. So, with 256GB/s of bandwidth available in the eDRAM frame buffer there should always be sufficient bandwidth for achieving 8 pixels per clock with 4x Multi-Sampling FSAA enabled and as such this also means that Xenos does not need any lossless compression routines for Z or colour when writing to the eDRAM frame buffer.

So, as far as the operation is concerned, once pixel data has come through the shader array and is ready to be processed into colour values in memory the Z data of the pixel is matched with the correct colour data coming out of the shaders. Xenos supports an "Alpha to Mask" feature, which allows for the use of Multi-Sampling for sort-independent translucency. All of this processing is performed on the parent die and the pixels are then transferred to the daughter die in the form of source colour per pixel and loss-less compressed Z, per 2x2 pixel quad. The interconnect bandwidth between the parent and daughter die is only an eighth of the eDRAM bandwidth because the source colour data value is common to all samples of a pixel here, and the Z is compressed. Once on the daughter die the pixels are unpacked to their Multi-Sample level and each sample is driven through their Z and Alpha computations and the final data is stored on the eDRAM until either the entire frame or current tile (we'll cover this in more detail later) being rendered is finished.

AMD1044_0279912
AMD1318_0158241
ATI Ex. 2124
IPR2023-00922
Page 11 of 33

When the frame or tile has finished rendering, the colour data will then be resolved on the daughter die, with the Multi-Samples being blended down to their pixel level. The resolved buffer information is then passed back from the daughter die to the parent which then outputs to system RAM such that, when all the tiles are finished, this can then be outputted to the display device. Although the resolved colour data has to be stored in system RAM, which uses some bandwidth during the transfer, the efficiency of the write as the resolved data comes out of the daughter die to be written to system RAM is very high. This high efficiency is due to the fact that it is dealing with a significant quantity of non-fragmented data and the bus isn't as busy with lots of other bandwidth consuming, high frequency and inefficient frame buffer read / write / modify operations for the back buffer. This helps in alleviating the fact that the parent die is also handling system memory requests. Also note that data can be written to the eDRAM at the same time as it is being cleared from the previous data that resided there, meaning there should be little to no wait when removing the previous data from the eDRAM (*We've heard comments from developers familiar to both designs that this element of Xenos bears similarities to the "Flipper" design for Nintendo's Gamecube, a part that was originally designed by ArtX, who of course were subsequently purchase by ATI, however ATI are keen to point out that while there may be apparent similarities the designs are entirely independent as there are distinct virtual and physical barriers between the groups working on the various console developments, past and present, and no members of the Flipper architecture team were involved in Xenos's development*).

As all the sampling units for frame buffer operations are multiplied to work optimally with 4x FSAA this is actually the maximum mode available. Although the developer can choose to use 2x or no FSAA, there are no FSAA levels available higher than 4x. The sampling pattern is not programmable but fixed, although it does use a sample pattern that doesn't have any of the sample points intersecting one or another on either the vertical or horizontal axis. Although we don't know the exact sample pattern shape, we suspect it will be similar to that seen on other sparse sampled / jittered / rotated grid FSAA mechanisms we've seen over the past few years, such as this.

The ROP's can handle several different formats, including a special FP10 mode. FP10 is a floating point precision mode in the format of 10-10-10-2 (bits for Red, Green, Blue, Alpha). The 10 bit colour storage has a 3 bit exponent and 7 bit mantissa, with an available range of -32.0 to 32.0. Whilst this mode does have some limitations it can offer HDR effects but at the same cost in performance and size as standard 32-bit (8-8-8-8) integer formats which will probably result in this format being used quite frequently on XBOX 360 titles. Other formats such as INT16 and FP16 are also available, but they obviously have space implications. Like the resolution of the MSAA samples, there is a conversion step to change the front buffer format to a displayable 8-8-8-8 format when moving the completed frame buffer portion from the eDRAM memory out to system RAM.

The ROP's are fully orthogonal so Multi-Sampling can operate with all pixel formats supported.

Render to texture operations will also be rendered out to the eDRAM first and then read out to UMA memory, when complete, in order to be used as a texture surface for the final frame rendering. Render to texture operations can also have Multi-Sample FSAA applied and the result can either be resolved on the way out to system memory or kept at the high resolution Multi-Sample level. As with standard pixel operations, the eDRAM memory can be written to with either another render to texture operation or pixel data whilst the data from the previous render to texture is being pushed out to UMA memory.

Page Navigation

◁ Page 4 ▼ ▷

AMD1318_0158242
ATI Ex. 2124
IPR2023-00922
Page 12 of 33

Site width adjust

Reset width

Privacy Policy

AMD1044_0279914
AMD1318_0158243
ATI Ex. 2124
IPR2023-00922
Page 13 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- Dave Kirk - NVIDIA Tesla Launch
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- Diving into Anti-Aliasing
- Andy Keane Interview & Tesla Adoption/Deployment
- A speculative look on the Wii U GPU
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- Ian Buck - NVIDIA Tesla Launch

ATI Xenos: Xbox 360 Graphics Demystified - Page 5

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

Z-Only Rendering Pass

Some games these days make use of graphics chips abilities to fast reject workload based on Z information. Engines such as Doom 3 or Source have the capabilities to, on each frame, run a geometry only pass which is for the purpose of pre-filling the Z buffer with the final Z depths of that frame. When the full frame is ready to be rendered, pixel information that has a higher Z depth than the information in the Z buffer is rejected before any pixel operations are carried out on it, meaning that there are no pixels written that are wasted due to overdraw. This z-only prepass is expected to be commonly used on Xenos as it has additional advantages for tiling, explained later.

A geometry pass to populate Z information is going to gain from a processor that has double the Z compare / write units in relation to its pure pixel fill-rate, which Xenos's does. However another factor is that this pass is actually going to require geometry processing over the vertex shaders. In a traditional shader capable graphics processor the number of vertex units can often be many times less than the pixel shader ALU's, however in the case of Xenos all of the shader units will be tasked purely with the geometry processing which should also ensure a fast operation of this early Z pass.

As with ATI's current desktop parts, Xenos features a Hierarchical Z buffer. Hierarchical Z buffers contain "coarser" Z information than the full resolution Z buffer - usually Hierarchical Z buffers are tiled down versions of the full resolution Z buffers and the highest Z value of that tile is stored for that group of pixels. In Xenos's case the Hierarchical Z Buffer stores down to 16 sample groups, which equates to 2x2 pixel groupings with 4x FSAA enabled. Once a triangle is setup, its pixel coverage areas can be compared against the Hierarchical Z buffer and if all of their Z values are greater than the value on the tile then they can all be rejected before any work is carried out, however if some are lower then they will be compared against the full resolution Z buffer. Because the Hierarchical Z Buffer exists on chip the checking operation is very fast and can also reject numerous pixel groups in a single cycle. Xenos can discard up to 64 pixels per clock cycle based on hierarchical z. As the Hierarchical Z buffer is populated on the Z only pass it will have the final Z values for its tile coverage when the full pass is done. This will result in more efficient use of the Hierarchical Z buffer in comparison to normal (PC) graphics processors on software that doesn't have an early Z only rendering pass built within the engine.

Something to note here is that with current PC parts the size of the on-chip ZCULL capabilities usually scales with the number of quads it is processing, and at least has to

AMD1318_0158244
ATI Ex. 2124
IPR2023-00922
Page 14 of 33

cater for the range of common PC resolutions, which are larger than those of even high definition TV sets. Being designed directly for the needs of a console Xenos can make some die savings in this area as the on chip Hierarchical Z buffer only needs to cater for a Z buffer size of these high definition TV resolutions.

Tiled Rendering

When FSAA is involved, the pixels always have to be stored at their sample levels until the frame is fully rendered. As the scene is rendered blends will be occurring on samples and, as the sub-samples for the pixels can contain different colour values, pixels cannot be down-sampled temporarily and then up-sampled if more blends have occurred. Basically, the down-sampling (resolve) step can only occur once it is known that all the operations for a given pixel are finished for a frame. The upshot of this is that in most traditional rendering cases the FSAA resolve is only done once the frame is finished and the back-buffer is written to the front-buffer (or even directly in the DAC's in some cases).

With the eDRAM being the primary rendering target for Xenos there looks to be a potential issue with rendering FSAA at High Definition TV (HDTV) resolutions: space. With only 10MB of rendering space available, the resolutions and FSAA depths that can be natively supported by the eDRAM could be limited. If we look back to our 512MB Radeon X800 XL review we see that the calculation for the size of frame-buffer requirements with FSAA goes along the following lines:

Back-Buffer = Pixels * FSAA Depth * (Pixel Colour Depth + Z Buffer Depth)
Front-Buffer = Pixels * (Pixel Colour Depth + Z Buffer Depth)
Total = Back-Buffer + Front-Buffer

Now, in the case Xenos the front-buffer only exists in UMA memory, so only the back-buffer size is of concern for the eDRAM space.

At the moment XBOX 360 is supporting 720p (progressive scan) and 1080i (interlaced) resolutions - 720p equates to 1280x720 pixels and 1080i equates to 1920x1080 pixels, however interlacing means that only the odd horizontal lines are refreshed on one cycle and the even lives on the next, which means that the frame buffer is only ever needing to handle 1920x540 pixels per refresh.

Here are the frame-buffer sizes for these HDTV resolutions and 640x480 with a colour depth of 32-bit (which will cover both the standard integer 32-bit format and the FP10) and a 32-bit Z/stencil buffer. Naturally, the sizes will increase if a higher Z-Buffer depth or a higher bit colour depth is used:

Framebuffer sizes

As we can see, with these bit depths, all the resolutions will fit into the 10MB of eDRAM without FSAA and at 640x480 a 4x FSAA depth will stay within the eDRAM memory size, with these colour and Z depths. However, at HDTV resolutions nothing can fit into the 10MB of eDRAM with any mode FSAA enabled. Xenos was specifically designed to perform very well in these cases by dividing the screen into multiple portions that fit within the eDRAM render buffer space. This is similar to prior tile-based renderers, but with a much larger base tile and with additional functionality to optimize the tiling approach.

Tiling mechanisms can operate in a number of ways. With immediate mode rendering (i.e. the pixels being rendered are for the same frame as the geometry being sent) it is never known what pixels the geometry is going to be mapped to when the commands begin processing. This is not known until all the vertex processing is complete, setup has occurred and each primitive is scan converted. So if you wanted to tile the screen with an immediate mode rendering system, the geometry may need to be processed, setup and then discarded if it is found not to relate to pixels that are to be rendered in

AMD1044_0279916
AMD1318_0158245
ATI Ex. 2124
IPR2023-00922
Page 15 of 33

the current buffer space. The net result here is that geometry needs to be recalculated multiple times for each of the buffers. Another method for tiling would be to use Tile Based Deferred Rendering which processes the geometry and "bins" it into graphics RAM, saving which render "tile" the geometry affects as it does so - these mechanisms have traditionally operated by deferring the actual rendering by a frame in order to parallelise the geometry processing / binning and the rendering (you may wish to take a refresher on PowerVR's tile based deferred rendering process in our article here).

ATI and Microsoft decided to take advantage of the Z only rendering pass which is the expected performance path independent of tiling. They found a way to use this Z only pass to assist with tiling the screen to optimise the eDRAM utilisation. During the Z only rendering pass the max extents within the screen space of each object is calculated and saved in order to alleviate the necessity for calculation of the geometry multiple times. Each command is tagged with a header of which screen tile(s) it will affect. After the Z only rendering pass the Hierarchical Z Buffer is fully populated for the entire screen which results in the render order not being an issue. When rendering a particular tile the command fetching processor looks at the header that was applied in the Z only rendering pass to see whether its resultant data will fall into the tile it is currently processing and if so it will queue it, if not it will discard it until the next tile is ready to render. This process is repeated for each tile that requires rendering. Once the first tile has been fully rendered the tile can be resolved (FSAA down-sample) and that tile of the back-buffer data can be written to system RAM; the next tile can begin rendering whilst the first is still being resolved. In essence this process has similarities with tile based deferred rendering, except that it is not deferring for a frame and that the "tile" it is operating on is order of magnitudes larger than most other tilers have utilised before.

There is going to be an increase in cost here as the resultant data of some objects in the command queue may intersect multiple tiles, in which case the geometry will be processed for each tile (note that once it is transformed and setup the pixels that fall outside of the current rendering tile can be clipped and no further processing is required), however with the very large size of the tiles this will, for the most part, reduce the number of commands that span multiple tiles and need to be processed more than once. Bear in mind that going from one FSAA depth to the next one up in the same resolution shouldn't affect Xenos too much in terms of sample processing as the ROP's and bandwidth are designed to operate with 4x FSAA all the time, so there is no extra cost in terms of sub sample read / write / blends, although there is a small cost in the shaders where extra colour samples will need to be calculated for pixels that cover geometry edges. So in terms of supporting FSAA the developers really only need to care about whether they wish to utilise this tiling solution or not when deciding what depth of FSAA to use (with consideration to the depth of the buffers they require as well). ATI have been quoted as suggesting that 720p resolutions with 4x FSAA, which would require three tiles, has about 95% of the performance of 2x FSAA.

Taking the previous sampling requirements, the memory quantities required resolved to the following number of tiles being required:

Number of tiles

Render to texture operations that have space requirements beyond 10MB can also operate in the tiled mode, however given that Xenos is going into a closed box environment its likely that developers of the system will consider what best fits the design of the console when they are developing their titles.

Page Navigation

◁ Page 5 ▾ ▷

AMD1318_0158246
ATI Ex. 2124
IPR2023-00922
Page 16 of 33

Site width adjust

Reset width

Privacy Policy

AMD1044_0279916
AMD1318_0158247
ATI Ex. 2124
IPR2023-00922
Page 17 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- A speculative look on the Wii U GPU
- Diving into Anti-Aliasing
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- Ian Buck - NVIDIA Tesla Launch
- Dave Kirk - NVIDIA Tesla Launch
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- Andy Keane Interview & Tesla Adoption/Deployment

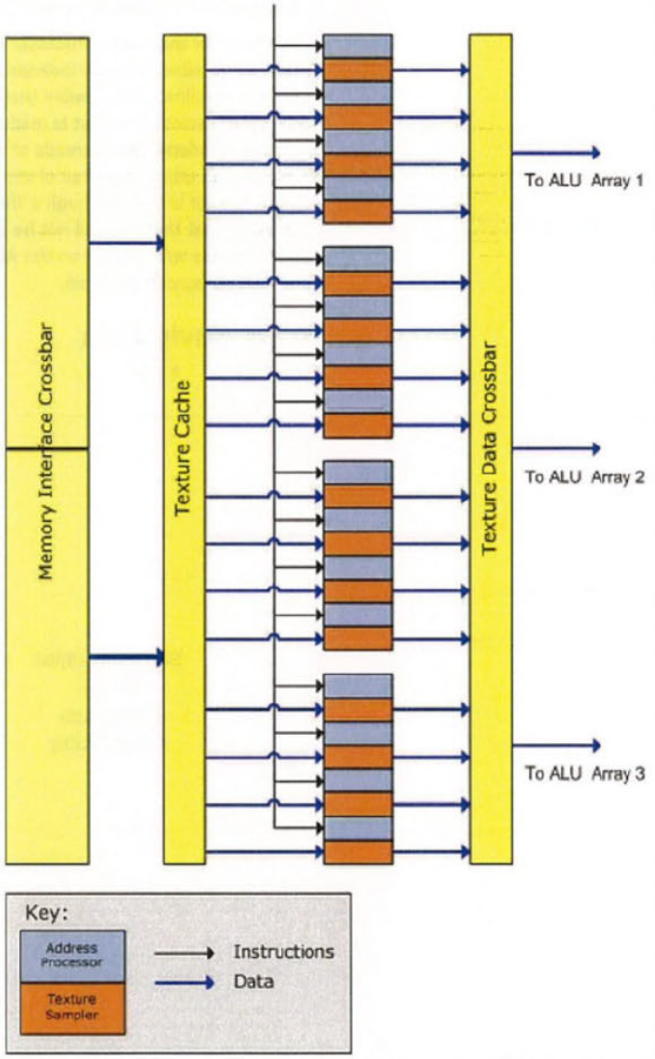
ATI Xenos: Xbox 360 Graphics Demystified - Page 6

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

Texture Processing

There are both 16 texture fetch units (filtered texture units, with LOD) and 16 vertex fetch units (unfiltered / point sample units) giving 16 of each type of texture samplers. Note that as the output data from the texture samplers is supplied to the unified shader arrays both types of texture lookups are available to either Vertex or Pixel Shader programs, if needed, and there are no limitations on the number of dependant texture reads. All of the texture address processing is handled locally by the texture processing array with each texture unit having its own texture address processor, so this is functionality that does not consume any cycles in the ALU shader array.

AMD1318_0158248
ATI Ex. 2124
IPR2023-00922
Page 18 of 33



Each of the filtered texture units have Bilinear sampling capabilities per clock and for Trilinear and other higher order (Anisotropic) filtering techniques each individual unit will loop through multiple cycles of sampling until the requested sampling and filtering level is complete. The texture address processor has some general purpose shader ability and is able to apply offsets from the input texture co-ordinates which can be used with higher order filtering techniques. The Anisotropic filtering capabilities adapts the number of samples taken dependant on the gradient of the surface that it is sampling, which is fairly normal for Anisotropic filtering mechanisms, ATI says that the anisotropic filtering quality is improved from previous generations of hardware. As Xenos is the controller of a UMA, the entirety of system RAM is available to the texture samplers, although they will not perform any operations on the eDRAM memory.

Xenos texture capabilities include support for DXTC (S3TC) texture compression routines as well as various other compression routines that are DXTC like in their operation. ATI2N (3Dc) is supported, as this is more or less just a twist of DXTC operation, as well as other compression formats that would be useful for normal maps. There are no compression methods available for float texture formats, although there are a total of 64 different texture formats supported.

AMD1044_0279920
AMD1318_0158249
ATI Ex. 2124
IPR2023-00922
Page 19 of 33

The design of the Xenos processor is such that latency within operations is hidden as much as possible. Texture lookups are usually one of highest latency operations in a graphics pipeline, and possibly the least predicable in terms of the variation in the number of cycles a request is made to the data becoming available. Xenos uses a large number of independent threads of vertex and pixel workload interleaved in order to achieve high utilization of all of the processing units while hiding the latency of fetches. The net result is that although a thread may need to wait for a texture sample to be achieved, that thread need not be stalling the ALU's waiting for texture data, instead other threads will operate on the ALU's which should maximise the available texture and ALU resources available.

Page Navigation

◁ Page 6 ▾ ▷

Site width adjust

Reset width
Privacy Policy

AMD1318_0158250
ATI Ex. 2124
IPR2023-00922
Page 20 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- Andy Keane Interview & Tesla Adoption/Deployment
- Diving into Anti-Aliasing
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- Dave Kirk - NVIDIA Tesla Launch
- Ian Buck - NVIDIA Tesla Launch
- A speculative look on the Wii U GPU

ATI Xenos: Xbox 360 Graphics Demystified - Page 7

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

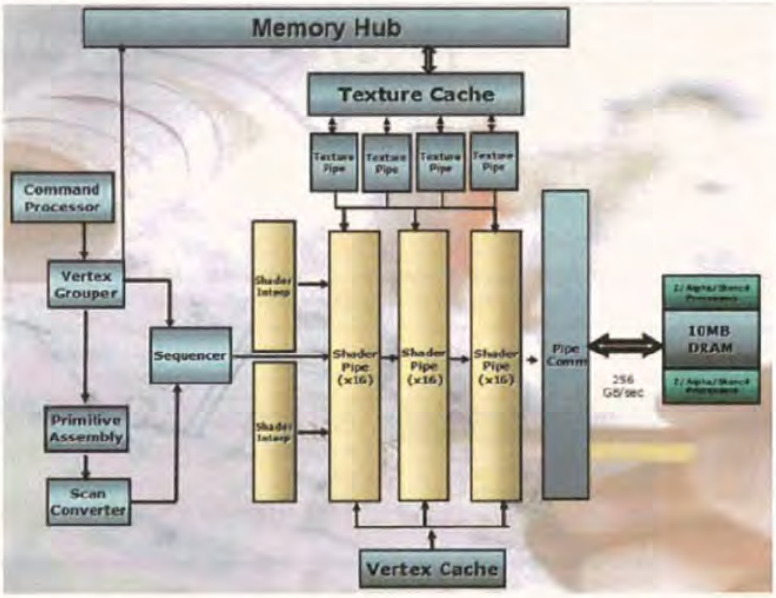
The Shader Pipeline Array

As we mentioned before the graphics pipeline has been an evolving entity but in this shader dominated environment that we are approaching shader ALU (Arithmetic Logic Unit) capabilities and organisation is fast becoming one of the most determinate factors in the overall performance of the graphics processor - if the processor has a poor shader pipeline, it doesn't matter how fast the rest of the chip is as the mix of graphics usage in today's graphics titles is shifting more and more to the shader pipeline.

With its unified shader pipeline Xenos has a fundamental difference with virtually all current shader capable graphics processors and whilst shader processing are enveloping both the geometry and raster pipeline, until now they have done both those elements distinctly. On Xenos there is a logical disconnect between the old OpenGL pipeline, which is basically the evolution path most graphics processors followed, as now the geometry and pixel shader processing are moved on to a single processing element of the chip as all the shader ALU's can dynamically be tasked with either vertex shader programs or pixel shader programs.

AMD1044_0279922
AMD1318_0158251
ATI Ex. 2124
IPR2023-00922
Page 21 of 33

Click for a bigger version



At this years E3 ATI held a press conference to briefly outline what they did for the XBOX 360 platform and highlight a few of the capabilities of the Xenos processor. From this press conference the diagram above emerged which gives an indication of the type of functional arrangement of the Xenos processor, however it is not entirely reflective of how the unified shader pipeline actually operates. Some other misconceptions of the pipeline operation has also risen since then so here we'll go into a little more depth to explain the unified processing that is occurring with Xenos.

Its been said that Xenos's shader processor is an array of 48 ALU's, however it is more correct to say that that it is 3 separate arrays of SIMD (Single Instruction Multiple Data) ALU's. Each one of the 48 ALU's can co-issue a vector (Vec4) and a scalar instruction simultaneously, essentially allowing a "5D" operation per cycle. Each one of the ALU's is a complete instruction duplicate of the others and are all single precision IEEE floating point 32-bit compliant. The ALU's will process everything in FP32 internal precision and there are no internal partial precision requirements for FP16. Additional to the 48 ALU's is specific logic that performs all the pixel shader interpolation calculations which ATI suggests equates to about an extra 33% of pixels shader computational capability.

The arrows on ATI's diagram above indicates that there is some dependency from one of the shader arrays to another, almost as though they are pipelined; this is in fact not the case and each ALU array is working independently of the other and the data is not pipelined between them. This being the case there is no dependency between what programs, or types of programs, are being executed on each of the three ALU arrays - at a snapshot in time they could, potentially, all be vertex processing, all be pixel processing or there can be a mixture of both vertex processing and pixel processing occurring on the three different 16 ALU arrays.

Page Navigation

◁ Page 7 ▼ ▷

AMD1318_0158252
ATI Ex. 2124
IPR2023-00922
Page 22 of 33

Site width adjust

Reset width

Privacy Policy

AMD1044_0279924
AMD1318_0158253
ATI Ex. 2124
IPR2023-00922
Page 23 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- A speculative look on the Wii U GPU
- Andy Keane Interview & Tesla Adoption/Deployment
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- Diving into Anti-Aliasing
- Ian Buck - NVIDIA Tesla Launch
- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- Dave Kirk - NVIDIA Tesla Launch

ATI Xenos: Xbox 360 Graphics Demystified - Page 8

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

Thread Handling

To keep all of the available units active as efficiently as possible there has to be some fairly complex thread management, and the diagram ATI displayed doesn't attempt to do any justice as to what is occurring.

The thread processing does bear similarities to the patent unearthed previously which has two "reservation stations", one for pixel shader instructions and another for vertex shaders. However, beyond that there are multiple arbiters and sequencers for each of the different workload types (ALU instruction operations, texture fetches and vertex fetches). These arbiters and sequencers interleave execution from the pools of instructions from the reservations stations in order to optimise the utilisation of the available processing elements whilst hiding the latencies of dependant operations within each thread, be they texture or shader instruction oriented. Additionally there are algorithms designed to prioritise the thread execution order and for transitioning threads from one workload type to another (i.e. a shader program first requiring some texture data input then requiring ALU instruction operations).

With this complex organisation, the threading mechanisms, the number of threads that are active, or ready to be active so the system hides latency effectively, ATI's testing indicates an average of about 95% efficiency over the shader array in general purpose graphics usage conditions. The throughput of the system is such that ATI expect to be able to achieve two loops, two texture instructions and 6 ALU instructions per pixel, per cycle at Xenos's peak fill-rate.

Shader Type Load Balancing

When trying to prioritise one pixel shader program over another pixel shader, or one vertex shader over another vertex shader the best choice is nearly always first in first out. With a unified shader architecture, though, where same ALU's will be presented with both pixel and vertex shader programs over time, the prioritisation between whether vertex shading or pixel shading should be done is a little more complex.

ATI, probably understandably, weren't too keen on giving many details out in regards to the prioritisation methodology, probably because there is some fairly proprietary logic behind it, but also because for the most part you shouldn't need to know much about it other than "it happens". From ATI's comments it sounds like a fairly complicated procedure, but conceptually it appears to monitor the vertex buffer and pixel export buffer (just before the transfer to the daughter die) and, depending on

AMD1318_0158254
ATI Ex. 2124
IPR2023-00922
Page 24 of 33

application program mix, there is an equation that prioritises between pixel shading and vertex shading dependant on the size of the buffers and how full they are.

This load balancing equation is inherently weighted, however information from the OS or even the application itself, which is obviously given the control by the developer, can alter that weighting a little in order to affect the prioritisation of the vertex and pixel shader programs. ATI's experiments show that the algorithm gives a quite optimal throughput and they expect only a few teir-1 developers will actually look into the altering the weighting of the algorithm.

ATI states that there will never be an unused shader array (or texture sampler for that matter) if there are any threads that are available to use it. Whilst the load balancing is like an arbiter, it only operates on threads that are ready to go in that decision.

Page Navigation

◁ Page 8 ▽ ▷

Site width adjust

Reset width
Privacy Policy

AMD1044_0279926
AMD1318_0158255
ATI Ex. 2124
IPR2023-00922
Page 25 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- Andy Keane Interview & Tesla Adoption/Deployment
- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- A speculative look on the Wii U GPU
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- Diving into Anti-Aliasing
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- Dave Kirk - NVIDIA Tesla Launch
- Ian Buck - NVIDIA Tesla Launch

ATI Xenos: Xbox 360 Graphics Demystified - Page 9

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

Capabilities

As we mentioned Xenos has capabilities that exceed those of a pure Shader Model 3.0, in DirectX terms, implementation. Whilst ATI are not yet giving out the full instruction set openly, they have broken down a number of the capabilities of Xenos and so that we can compare them against the Shader Model 2.0 and 3.0 capabilities. Note that the table below breaks the operations into Pixel and Vertex Shader models for SM2.0 and SM3.0 as the capabilities are still quite distinct between the two, however with the unified shader architecture on Xenos these differences are removed such that the capabilities that are available to one type of shader processing are available to the other as well.

ATI Xenos Capabilities

** Note: We are listing here Xenos hardware capabilities, which may or may not be the same as that is exposed through the API for the XBOX 360 hardware. However, as this is a closed system with a custom API for the hardware we would expect them to be exposed for use by developers.*

Some additional capabilities that are included on the Xenos graphics processor are:

- **Multiple Render Targets (MRT)**
  4 render target outputs are supported as output and, as an addition to current processors, each target can have different blend capabilities.
- **Hierarchical Stencil Buffer**
  Operates similar to the Hierarchical Z buffer to quickly cull unnecessary stencil writes.
- **Alpha-to-Mask**
  Converts Pixel Shader output alpha value to a sample mask for sort-independent translucency.

An additional functional element that Xenos provides to developers is a Geometry Tessellation Unit. The tessellation unit is a fixed function engine that accepts triangles, rectangles and quads as its primitive input, along with a tessellation level per edge such that the level of tessellation is completely variable across the surface of the original primitive.

AMD1318_0158256
ATI Ex. 2124
IPR2023-00922
Page 26 of 33

Current graphics processor architectures can mark to "kill" a pixel in the pixel shader and this is the case with Xenos. However, as the architecture unifies the shaders the capabilities of both the shader program types (vertex and pixel) are available to each other, so the kill command will also operate for vertices. Although the vertex isn't retired in the ALU as it goes through the rest of the geometry pipeline to be set up vertices marked as killed will be ignored, effectively reducing the level of detail in the resultant geometry.

Although 4000 is a reasonably large number of instructions to support in a single code block, this is a limitation on the number of instructions that can be applied to a single shader program because the full program is stored on the chip and never partially retrieved from memory. However, should the developer wish to exceed that in a single block then ATI's F-Buffer technology is included to increase the shader length. Alternatively ATI's "MEMEXPORT" (see "MEMEXPORT" section) could be used to increase the length of a shader program beyond the nominal 4000 instructions.

The combination of the shader array and tessellation unit can now make the, oft spoken of but rarely seen, capability of displacement mapping an attainable method to use as this truly becomes a single pass algorithm for Xenos. A simple primitive can be sent to the tessellation unit which is then subdivided into a vertex mesh and then that can be applied to a vertex shader program that does displacement map lookups via the vertex fetch texture units and then the geometry mesh altered according to the sampled values from the texture sampler. Alternatively, if the screen-space projection of the input primitive to the tessellation unit is calculated prior to tessellation then the per-edge tessellation level can be figured out dependant on that projection such that displacement mapping with correct, dynamic level of detail can be achieved.

Page Navigation

◁ Page 9 ▾ ▷

Site width adjust

Reset width

Privacy Policy

AMD1044_0279926
AMD1318_0158257
ATI Ex. 2124
IPR2023-00922
Page 27 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- Dave Kirk - NVIDIA Tesla Launch
- Diving into Anti-Aliasing
- A speculative look on the Wii U GPU
- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- Ian Buck - NVIDIA Tesla Launch
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- Andy Keane Interview & Tesla Adoption/Deployment

ATI Xenos: Xbox 360 Graphics Demystified - Page 10

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 3rd Mar 2007

MEMEXPORT

In addition to its other capabilities Xenos has a special instruction which is presently unique to this graphics processor and may not necessarily even be available in WGF2.0 and this is the MEMEXPORT function. In simple terms the MEMEXPORT function is a method by which Xenos can push and pull vectorised data directly to and from system RAM. This becomes very useful with vertex shader programs as with the capabilities to scatter and gather to and from system RAM the graphics processor suddenly becomes a very wide processor for general purpose floating point operations. For instance, if a shader operation could be run with the results passed out to memory and then another shader can be performed on the output of the first shader with the first shader's results becoming the input to the subsequent shader.

MEMEXPORT expands the graphics pipeline further forward and in a general purpose and programmable way. For instance, one example of its operation could be to tessellate an object as well as to skin it by applying a shader to a vertex buffer, writing the results to memory as another vertex buffer, then using that buffer run a tessellation render, then run another vertex shader on that for skinning. MEMEXPORT could potentially be used to provide input to the tessellation unit itself by running a shader that calculates the tessellation factor by transforming the edges to screen space and then calculates the tessellation factor on each of the edges dependant on its screen space and feeds those results into the tessellation unit, resulting in a dynamic, screen space based tessellation routine. Other examples for its use could be to provide image based operations such as compositing, animating particles, or even operations that can alternate between the CPU and graphics processor.

With the capability to fetch from anywhere in memory, perform arbitrary ALU operations and write the results back to memory, in conjunction with the raw floating point performance of the large shader ALU array, the MEMEXPORT facility does have the capability to achieve a wide range of fairly complex and general purpose operations; basically any operation that can be mapped to a wide SIMD array can be fairly efficiently achieved and in comparison to previous graphics pipelines it is achieved in fewer cycles and with lower latencies. For instance, this is probably the first time that general purpose physics calculation would be achievable, with a reasonable degree of success, on a graphics processor and is a big step towards the graphics processor becoming much more like a vector co-processor to the CPU.

AMD1318_0158258
ATI Ex. 2124
IPR2023-00922
Page 28 of 33

colour or Z information which is likely to limit its usefulness.

ATI indicate that MEMEXPORT functions can still operate in parallel with both vertex fetch and filtered texture operations.

Display

Generally speaking we are used to the graphics processors being responsible for the display output capabilities, however in the case of the XBOX 360 that is not the case. Xenos itself outputs the frame-buffer digitally to another display device of Microsoft's choosing.

Power and Die Savings

ATI state that Xenos is a fairly low power consumption design for several reasons. For starters, the mechanism of the ALU's is designed to operate by reducing latencies which, if fully successful, should increase the efficiency of the operation of the chip. Likewise, a unified pipeline also increases efficiency by removing cases where the vertex shader is idle, waiting for the pixel shader to have available slots, or the pixel shader is idle, waiting for the vertex shader to produce data. If such efficiencies are fully realised in relation to current graphics processing methodologies, this can result in either a smaller chip (hence cheaper) with the same performance as larger chips with a traditional architecture, or the same sized chip with more ALU's dedicated to processing, hence higher performance. Of course, this does depend on exactly how "inefficient" current graphics processors really are and whether future processors that use distinct vertex and pixel shaders don't find alternative methods for increasing efficiencies. Although it could be implemented in future designs, current graphics processors may not be able to clock gate between the vertex and pixel shader units, which results in power burnt if one end of the pipeline is waiting for the other; this inherently isn't an issue with a unified platform.

ATI also believe that Xenos specifically has the most advanced power management features of any chip they have produced so far. There is a top level power management system that can be controlled by the OS that allows for various elements of the pipeline to be turned off for various operations, such as DVD playback for instance. There are low power modes that regulate the speeds and voltages and, when inactive, the data is held in stasis rather than just switching transistors on and off to keep the data. However, in the graphics core itself there localized power ma
applied at the block level to minimize power consumption du
periods.

Cookie Control Beyond3D uses Cookies to store information on your computer I'm fine with this

When we factor in the savings for both power and die size savings we can see that this potentially has some advantages over traditional architectures. In the case of the XBOX 360 not only does this result in a relatively smaller die size for a fairly high performance ratio but also means that the graphics need only be air cooled, without the use of its own additional fan. Beyond the immediate application we can see that unified designs that are bound for the PC could have smaller die sizes for equivalent performances as current discrete solutions or more silicon dedicated to either more ALU's for higher performance, or other transistors dedicated to other functionality.

Page Navigation

◁ Page 10 ▾ ▷

AMD1044_0279930
AMD1318_0158259
ATI Ex. 2124
IPR2023-00922
Page 29 of 33

Site width adjust

Reset width

Privacy Policy

AMD1318_0158260
ATI Ex. 2124
IPR2023-00922
Page 30 of 33

Latest Reviews

- NVIDIA Fermi GPU and Architecture Analysis
- ATI Cypress Gaming Performance Analysis
- ATI Cypress GPU and Architecture analysis
- ATI RV740 GPU and Architecture Analysis
- NVIDIA GT200 GPU and Architecture Analysis

Latest Articles

- Was Harry Potter actually into rendering? Beyond3D's first ever book review
- Q&A with Visceral's Technical Art Director Doug Brooks on Dead Space 2
- Dave Kirk - NVIDIA Tesla Launch
- Travelling in Style: Beyond3D's C++ AMP contest - WINNER ANNOUNCED
- E3 2011: Behind Closed Doors - Witcher 2 Xbox 360... and a Nurse
- Ian Buck - NVIDIA Tesla Launch
- AMD's John Bridgman on Radeon, Linux and Open Source 3D
- Diving into Anti-Aliasing
- A speculative look on the Wii U GPU
- Andy Keane Interview & Tesla Adoption/Deployment

ATI Xenos: Xbox 360 Graphics Demystified - Page 11

Published on 13th Jun 2005, written by Dave Baumann for Consumer Graphics - Last updated: 21st Mar 2007

So, Why Not Now for the PC?

The most immediate question that comes to mind is if all the graphics elements that are seen within the XBOX 360 are so good, why aren't we seeing it in the PC space yet?

Xenos's particular range of features are going into a closed box environment, hence the API can be tailored to expose all of the features of the chip, however on the PC space graphics processors really need to be tailored to the capabilities of the current DirectX release. This is where Xenos has an issue in that its features and capabilities are clearly beyond the current Shader Model 3.0 DirectX9 specification while it lacks features that are expected to be a requirement for WGF2.0.

WGF2.0 has requirements for virtualisation, and whilst Xenos has the luxury of being able to access the entire system memory this is by virtue of the fact that it is the system memory controller. Part of the virtualised requirements of WGF2.0 appear to be able to include unlimited length shaders, where Xenos has some hard coded limits here and, whilst large and defeat-able through a couple of methods, probably wouldn't meet the requirements for WGF2.0 here. When we looked into WGF2.0 in our DirectX Next article there was, at that point, a suggested requirement to the graphics pipeline to have a fully integer instruction set as well as the floating point pipelines, however Xenos's ALU's are purely float in operation.

The shader processing design is clearly very different from today's graphics processors, but then there is the fact that PC's will be catering to a greater range of utilisation of features as there is a quicker evolution cycle as far as graphics are concerned - some titles being released even now are very limited in their shader use, whilst others are utilising them extensively; Xenos's design is likely to be most beneficial when the majority weight of processing requirements goes towards shaders as opposed to the more fixed functionality elements of the pipeline. Arguably, though that balance is already shifting, and if Xenos is actually as good at shader processing as it purports it still begs the question as to why ATI are looking towards more traditional shader pipeline over the next 12-18 months instead of using this, even though it has slightly greater capabilities than current PC API's allow. Perhaps the answer lies in the fact that this is such a big change that trialing it in a closed box environment, where developers will have more time to tailor specifically to the hardware requirements, as the hardware will stay the same for the next 3-5 years, makes sense as they can also use the experience gained from that to assist in the development of a PC architecture based on a similar processing methodology.

AMD1044_0279932
AMD1318_0158261
ATI Ex. 2124
IPR2023-00922
Page 31 of 33

One area of PC graphics processing that a unified architecture will be sure to benefit from immensely is that of the Workstation market. PC graphics processors are primarily designed for desktop PC's, hence their main target is for gaming which biases the workload very much more to the raster (rendering) pipeline rather than the geometry pipeline - current high end graphics processors have two to three times the math logic dedicated to pixel shading than vertex processing. Many workstation applications, such as CAD and CAM use, put the onus heavily on the geometry processing as they will be often rendering very detailed geometric representations of objects and frequently viewed in wire frame mode, however most workstation graphics processors sold are derived directly from desktop processors, which isn't necessarily optimal as they are designed for pushing pixels. With a unified shader architecture the graphics processor is neither biased towards either pixel or vertex processing in terms of the ALU math capabilities and is much more versatile in its potential usage - workstation graphics that use such an architecture can suddenly find themselves with many times the geometry throughput performance, at more or less the same costs, as the utilisation is automatically balanced and spread across the entire array of ALU's that are available on the entire graphics processor.

However, possibly one of the most immediate (without WGF2.0 for Windows being here) application for unified shaders is actually outside of the PC space and in mobile phone 3D graphics engines. Presently ATI are yet to produce a fully shader capable "Imageon" graphics processor for the handheld markets and are not expected to until 2006, however with the onus on minimal power utilisation in minimal die size in the handset space anything that mitigates wastage is going to be a welcome element, and with slightly less rigid specification targets to meet in the handheld arena a unified shader architecture may be the ideal approach when, inevitably, ATI choose to create shader enabled handset parts.

Conclusion

Overall it looks as though Xenos represents some highly interesting design choices on many fronts and clearly seems as though ATI have attempted to come up with a very different architecture to, at the very least, target the specific needs of the XBOX 360 console platform. It will be very interesting to see the performance and quality of graphics it is able to produce once developers have had decent access to development kits based on the final hardware, however we suspect that it won't be until the second generation of XBOX 360 titles before we see developers being able to seriously scratch the surface of understanding the processing capabilities of Xenos and the XBOX 360 as a whole, given that most of the first generation titles will not have been developed on the final hardware. That being said, though, much of the graphics architecture is transparent to the developer and they shouldn't need to concern themselves much with the types of workloads they are requesting of graphics processor as this will all be handled automatically, without stalling any part of the pipeline.

Apart from the interesting use of eDRAM in this design, which is clearly targeted towards the console environment (although from its operation even this could potentially be moved into other the PC space if the driver forced a Z only pass, however this may be a little risky) the design of the ALU arrays, texture processing and threaded nature of the system is clearly a large departure from any of the shader architecture we've seen so far. Despite having a raw ALU quantity that exceeds any platform currently available, the primary key to the design of the processing is that of "efficiency" when processing shader programs, by organising the workloads in a threaded manner in order to try and constantly keep the available processing elements active, interleaving latency bound dependant operations and having a unified platform that is agnostic to whether it is processing Vertex or Pixel Shaders and never having one type of operation stalling the other. The primary question here is exactly how "inefficient" are current architectures in relation to this one, which is a difficult question to answer because no hardware vendor is going to tell you their graphics processors are inefficient. All we can say at the moment is that clearly Xenos's shader processing architecture is fundamentally and significantly different from current platforms and ATI

AMD1318_0158262
ATI Ex. 2124
IPR2023-00922
Page 32 of 33

obviously must have perceived an issue with current methodologies otherwise they wouldn't have gone to these lengths to change the pipeline.

In the future, with WGF2.0's unified shader language, it would be hard not to see this type of threaded shader architecture not make its way across to ATI's PC products.

Page Navigation

◁ Page 11 ▼

Site width adjust

Reset width
Privacy Policy

AMD1044_0279934
AMD1318_0158263
ATI Ex. 2124
IPR2023-00922
Page 33 of 33