Members
Welcome!
Sign Up
Your Account
Preferences
Logon

Navigation
Home
About
Introduction
+GameCube Review
-Specification
-Bigger Is Better?
-Gekko
-Flipper
-Embedded Memory
-System Memory
-Main Memory
-A-RAM
+XBox Review
Comparison
Conclusions
References
Acknowledgments
Disclaimer

Search
Search

The GameCube Review

Specification

System Specification	
CPU:	Custom IBM PowerPC "Gekko"
Internal Cache:	L1:Instruction 32KB, Data 32KB (8 way)
	L2: 256KB (2 way)
Memory Subsystem	
Main Memory:	24 MB MoSys 1T-SRAM, Approximately 10ns Sustainable Latency
System Memory:	40MB
Main Memory Bandwidth:	2.6GB/second (Peak)
Graphics Subsystem	
Graphics Processor:	Custom ATI/Nintendo "Flipper"
Clock Frequency:	162 MHz
Real-world polygon:	6 million to 12 million polygons/second (Peak)
Texture Read Bandwidth:	10.4GB/second (Peak)
Audio Subsystem	
Sound Processor:	Custom Macronix 16-bit DSP
Sampling Frequency:	48KHz

Back To Top

Bigger is Better?

Just like a conventional home PC, when asking about games consoles, people want to know how fast it is? What are the graphics like? Some people may even ask about polygons-per-second! And this is fine; except the wee man in the high street electrical retail store is going to say one thing: XBox is better because XBox has a higher clock rating, therefore it's faster. Rubbish! The more astute among you may be aware that clock rating is not the be all and end all of processors, and that case certainly applies here.

The IBM PowerPC 750CXe, code-named "Gekko", is the brains behind the beauty of GameCube. The main aims in the design of the Gekko were to make it powerful, cheap to manufacture and it had to run cool enough that it could fit into a small enclosure i.e. the GameCube casing.

The processor used in the console is actually a derivative of the above, the extras being an addition of forty or so SIMD (Single Instruction stream, Multiple Data streams) floating point instructions, specifically included to aid in the processing of the data required to move someone through a virtual world as smoothly as possible. Utilising Motorola's AltiVec (a SIMD pseudonym), the instruction set was geared towards the types of fast processes required by the designers of the console. It is said in computer architecture and design, make the common case fast. Well this is exactly what has been done by using AltiVec. With the AltiVec SIMD instruction set, support is provided for the transfer of 4 32-bit floating-point instructions which greatly accelerates all graphical calculations. The Gekko also has a four stage basic integer pipeline, this is what enables the processor to run at very low clock speeds. As I mentioned before though it is the Gekkos ability to handle floating point instructions that is more important for gaming. The Gekkos floating point pipeline is 7 stages long and since the Gekko is a RISC processor and not CISC like the X-Box it does not have to spend long in the fetch/decoading stages of the pipeline. In fact once the instructions are fetched they are instantly dispatched and an amazing one clock cycle later they are sent to be executed.

The PowerPC architecture used in the Gekko is a 64-bit architecture with a 32-bit subset. This supports 32-bit addresses and also has two 32-bit integer ALUs. In addition to this there is a further 64-bit FPU that is capable of managing 64-bit or 32-bit floats using its 32 64-bit floating point registers. This is not where the enormous number of registers stops the Gekko also boasts 32 general purpose registers(GPR) which blows the X-box's tiny 8 GPRs out of the water. The Gekko is built using the 0.18 micron process as is the X-Box but it is held back by its short pipeline which makes processor speeds of over 500MHz impossible. The Gekko though using copper connects instead of utilising the dated aluminium ones in the X-Box. This ensures that the electricity in the system is more efficiently conducted. The clock speed of the Gekko is a flimsy 485MHz or three times its 162MHz front side bus frequency. This is ofcourse why the Gekko has a very short pipieline i.e. having a short pipeline means that more instructions can be processed in the limited number of clock cycles available. This slow clock speed will not necessarily mean slow performance as all the Computer Architecture & Design students will know. The X-Box's CPU could be ten times faster than the Gekko but if it is bottlenecked somewhere it will simply not be as fast.

As I have mentioned the Gekko does not seem to be a mammoth processor, instead of raw MHz power the guys at Nintendo have seemingly opted to concentrate more on the physical characteristics of the procerssor. Although the Gekko has a massive on die L1 and L2 cache (64Kb/256Kb) compared with the X-Box's (32Kb/128Kb) and huge 21 million transistors compared with the X-Box's 9 million, the dimensions are still only 45mm^2. As planned the Gekko is a very cool running processor, mainly down the its slow clock

low heat dissapation ensures that it is a very cheap processor to manufacture. This was Nintendos goal all along, to make a processor that was powerfull enough whilst also being cheap to manufacture and cool running enough that it could fit in the GameCube shell.

Back To Top

Flipper

A faster processor by using less transistors for cheaper? Well ATI have pulled it off and Nintendo's choice was a good one. Rather than using conventional SRAM for the main graphics cache, Nintendo decided to use this exciting new technology. By deciding to use ATI's chip, they were able to bring the GPU dimensions down by about 30% of its rivals to just 106mm^2, which sits proudly above the gekko.

Rather than go with the flow, GameCube designers decided against the obvious choice of DDR memory for any of its memory subsystems. The reason for this is MoSys's revolutionary 1T-SRAM. More analysis of this technology is given later, but the justification is simple. It's smaller. The manufacturing process permits the use of one transistor rather than the standard 4 or 6 and so the density is at least 4 times better than anything else GeForce or Raedon can produce, although the flipper chip itself is about twice as big as the Gekko, as seen below.

Image from ExtremeTech.

Similarly built in the 0.18 micron fashion of Gekko, Flipper manages to squeeze 51 million transistors onto the die. The entire Flipper chip runs at 162MHz which means that not only the graphics, but also all I/O is processed at 162Mhz. The only thing not processed at this rate is the audio stream,

latency times and guaranteeing the previous statement that everything on the Flipper chip will indeed run at 162Mhz; this also includes the graphics core. GameCube is therefore virtually guaranteed to reach its advertised 12 million due to the efficiency created by the way the DRAM has been embedded on the chip.

This is also boosted by the fact that Gekko has plenty Front Side Bus bandwidth at its disposal, simply because its FSB is running at 162MHz. This results in a 1.3GB/s connection between Gekko and the North Bridge, which in this case is integrated into a single chip along with the graphics core, aka Flipper, helping the processing of graphics and of course, system memory which is discussed later.

Back To Top

Embedded Memory

GameCube's designers have managed to squeeze a rather large 2Mb Z-buffer and equally large 1MB texture buffer on chip. This technique is useful as it reduces latency and greatly increases performance per millions of pixels. This is not where the graphics efficiency ends however, as early Z-tests in the GPU's pipeline also help increase the performance as this reduces any pixel processing load that later pipeline stages will have to handle.

To explain further, these different types of Z-tests are performed to determine whether or not a particular pixel will be visible, shaded, fogged out, in focus etc. As one might imagine, doing this for every pixel can be very computationally exhaustive and that's where this large buffer comes into play. Some might call it GameCube's "ace in the hole", but whatever you call it, paired with the 4 pixel pipeline flipper provides, it certainly helps minimise latency.

The 1MB texture cache helps performance but the impact is negligible compared to the 2MB Z-buffer. Flipper offers 32 1T-SRAM devices (256Kbit each) offering a peak, and therefore generally unobtainable 10.4GB/s of bandwidth to this cache. This is attained through the utilisation of 4 pixel pipes, as mentioned earlier, all running at the peak memory main bandwidth of 2.6GB/s.

The most interesting thing about the technology used here is that soon it will be getting smaller. As mentioned above, both Flipper and Gekko were manufactured using 0.18 micron production techniques but it has become apparent that 0.15 and 0.13 micron is now possible, reducing the size of the die by half and leaving a hole to fill with more of MoSys's transistor technology.

Back To Top

System Memory

Further functionality comes from the use of System Memory as a whole The GameCube designer?s once again thought about functionality by not making use of a shared memory system, or Unified Memory Architecture (UMA). In doing this GameCube?s 24Mb again sounds like too little, but this is just

Back To Top

Main Memory

By making the decision to continue to use MoSys outside of flipper, GameCube?s designers once again take the brains over brawn approach to their machine. It is understandable that most people would have expected DDR (Double Data Rate) SDRAM to be used as the main memory module, as the not-inexpensive DDR can produce much higher frequencies at a similar cost per size ratio and cost to the pocket. However Nintendo?s key rationale for using MoSys again lies in latency, or rather the lack of it.

The off-chip RAM is synchronised with gekko?s FSB, as well as flipper?s operating frequency and in this case is also twice as fast - 162MHz x2 resulting in 324MHz or, with a quick back of the envelope calculation, 2.6GB/s of bandwidth to main memory, again two times the external bus bandwidth. See below for a photo of the off-chip, motherboard-integrated MoSys main memory chip. Both 128-bit chips can also be viewed in the earlier picture under the label C.

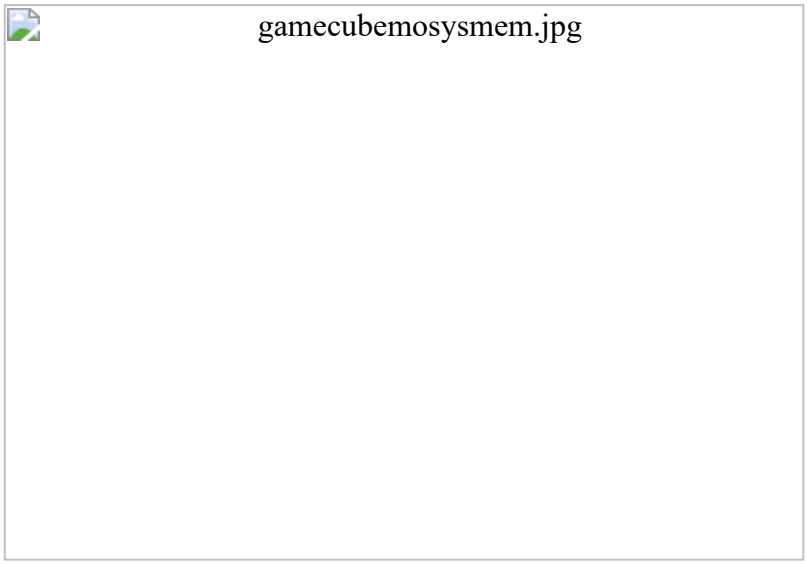

Image from Anandtech.

Those amongst you who were, and still are avid followers of the Geforce family of GPU?s will be aware that this equates to the same amount as bandwidth as an original Geforce 256. What must be remembered though is the real powerhouse of the GameCube is not the gekko or even the memory sub-system. This is not a PC and upgrading the RAM and CPU will not help here. The flipper is constantly working to stay well ahead of the main memory which means by the time the off-chip MoSys sees any of the data, all of the Z-buffer accesses have been done by its embedded counterpart and the frame being drawn is done initially by the on-die 2MB buffer. Obviously this is a significant drop in bandwidth as most PC gamers are used to dealing with, but this is due to this crucial time saving fore-thought of the GameCube?s designers.

The benefit the GameCube gains from using 1T-SRAM as its main memory is low latency access and thus better bus utilisation; due to the synchronisation

Back To Top

A-RAM

The remaining 16Mb is slower A-Memory, or Audio Memory, constructed from bargain basement, bog standard DRAM and runs at 81MHz or half of flipper?s operating frequency. The title Audio Memory is a bit deceptive as GameCube?s A-RAM also handles low priority operations that flipper can relegate due to non-graphics intensive nature of the processes. It was decided to make this slower due to cost reasons, but also to make sure there was plenty left over to make the other key pieces of memory fast, which turns out to be GameCube?s best point. The A-RAM is shown below and is listed as D in the previous Motherboard layout picture.

Image from Anandtech.

Most people who build their PC?s as easily as they did their Lego sets when they were younger, will notice that 16Mb is a fair amount of memory to be used for just audio processing. And they?d be right. 16Mb is infact considered so much that the amounts that would normally have been redundant are subtly and efficiently used by gekko and the GameCube?s audio processor, a custom Macronix DSP(Digital Signal Processor) which is directly connected to the A-RAM through an 8-bit bus.

The Macronix is not clearly visible in any diagram or photo presented here as it resides as a co-processor within flipper. The Macronix itself is a 16-bit DSP, similar in power to the 16-bit PCI cards anyone can install in their PC. Except this is of course a custom version which has some very neat extras.

Most, if not all gamers these days crave a large helping of clear-as-a-bell sound with their Big Mac-Graphics. Unfortunately, this is where GameCube hits a snag. It does not provide a digital output which means no Dolby 5.1. Except that it does. To explain, we need to know what Dolby is all about.

Everyone knows what Dolby surround sound is these days, and everyone also knows that Dolby are the best sound engineers around. So when Nintendo, and a few others, presented them with the problem of extracting 5 channels from just two, scratching of heads took place. A discrete (normal) 5.1 signal is decoded by each sound being mapped to its own channel, giving the full bandwidth support for complete channel separation with targeting of a specific sound to a specific channel. What Dolby came up with however, was Dolby Pro Logic II, the successor to that all too familiar technology of the

negligibly inferior surround sound.

To those who were paying attention earlier, you should be thinking 16MB is a lot of memory for just audio processing alone. And once again you?d be right. Hardcore Nintendo fans and those who just happened to own a N64 console may remember the buzz created when it was announced N64 software was to remain ROM cartridge based. This was the result of another good day of thinking about the problem by Nintendo. Anyone who?s ever played one will know that there was virtually no loading times and this was all down to allowing reasonably fast access to game bits on the ROM cartridge. Now that Nintendo have decided to buckle to tradition, in a non-traditional manner typical of Nintendo, the access times have become slower. Orders of magnitude slower actually, down to the fact that CD?s (or unconventional half-discs) are now used rather than ROM cartridges because of the amount of data storage required.

The problem here is latency. Whereas the other components of the GameCube were designed to combat any and all latency, the one thing Nintendo couldn?t change was the read latency on the CDROM drive. Except they did. They made use of a technique called Constant Angular Velocity which, unlike its opposite Constant Linear Velocity, keeps the motor rotating at the same speed no matter which track is being read. CLA is advantageous over CLV because CLV incurred a brief latency whenever the drive needed to change the rotational speed.

However, even this minor tweak isn?t going to solve the problem of: What happens to the gameplay if we need a bit of data now, but it is only to be found on the CD. This is where our semi-redundant A-RAM comes into play. Developers for the GameCube have been allowed access to this ?extra? memory to use it to create a CD-ROM cache as a kind of "virtual" ROM cartridge. This allows the fast access Nintendo gamers have been used to and once again demonstrates the careful architecture of this machine.

Of course, flipper tends to keep most of the audio processing chores using its integrated Macronix DSP core freeing up most of the 16Mb for caching non-graphics intensive data.

So it now becomes clear that GameCube actually has a total of 43Mb system memory available to it, made up of the 16Mb A-RAM, 24Mb main memory and the 3Mb hidden away on-chip for flipper?s cache. All tolled, this provides the gekko with ample space to deal with much of what developers are sure to throw at