

Cell Architecture Explained: Introduction

Introduction

Designed for the PlayStation 3, Sony, Toshiba and IBM's new "Cell processor" promises seemingly obscene computing capabilities for what will rapidly become a very low price. In these articles I look at what the Cell architecture is, then I go on to look at the profound implications this new chip has, not for the games market, but for the entire computer industry. Has the PC finally met it's match?

To date the details disclosed by the STI group (Sony, Toshiba, IBM) have been very vague to say the least. Except that is for the patent application* which describes the system in minute detail. Unfortunately this is very difficult to read so the details haven't made it out into general circulation even in the technical community.

I have managed to decipher the patent and in parts 1 and 2 I describe the details of the Cell architecture, from the cell processor to the "software cells" it operates on.

At this point we do not have any performance figures so any figures given are derived from either the figures in the patent or the 4.6GHz given for the chip itself. All performance figures are thus estimates based on hypothetical maximums.

Cell is a vector processing architecture and this in some way limits it's uses, that said there are a huge number of tasks which can benefit from vector processing and in part 3 I look at them.

The first machine on the market with a Cell processor will steal the performance crown from the PC, probably permanently, but PCs have seen much bigger and better competition in the past and have pushed it aside every time. In part 4 I explain why the PC has always won and why the Cell may have the capacity to finally defeat it.

In part 5 I wrap it up with a conclusion and list of references. If you don't want to read all the details in parts 1 and 2 I give a short overview of the Cell architecture.

* This article is based on the interpretation of the original 2002 patent application by Masakazu Suzuoki and Takeshi Yamazaki. **[Cell Patent]**

[Part 1: Inside The Cell](#)

In Parts 1 and 2 I look at what the Cell Architecture is. Part 1 covers the computing hardware in the Cell processor.

- Background
- So what is Cell Architecture?
- The Processor Unit (PU)
- Attached Processor Units (APUs)
- APU Local Memory

[Part 2: Again Inside The Cell](#)

Part 2 continues the look at the insides of the Cell, I look at the setup for stream processing then move on to the other parts of the Cell hardware and software architecture.

- Stream Processing
- Hard Real Time Processing
- The DMAC
- Memory

- Software Cells
- Multi-Cell'd Animals
- Concrete Processing
- DRM In The Hardware
- Other Options And The Future

Part 3: Cellular Computing

Cells are not like normal CPUs and their main performance gains will come from the vector processing APUs, in this section I look at the type of applications which will benefit from the Cells power.

- Cell Applications
- Games
- 3D Graphics
- Video
- Audio
- DSP (Digital Signal Processing)
- SETI
- Scientific
- Super Computing
- Servers
- Stream Processing Applications
- Non Accelerated Applications

Part 4: Cell Vs the PC

x86 PCs own almost the entire computer market despite the fact there have been many other platforms which were superior in many ways. In this section I look at how the PC has come to dominate and why the Cell may be able to knock the king from his throne.

- The Sincerest Form of Flattery is Theft
- Cell V's x86
- Cell V's Software
- Attack
- Cell V's Apple
- Cell V's GPU
- The Cray Factor
- The Result

Part 5: Conclusion and References

- Short Overview
- Conclusion
- References And Further Reading

Part 6: Updates, Clarifications and Missing Bits

Some people have misunderstood parts of this article, this is understandable given it's size, scope and the odd dubious explanation. I have subsequently added an extra section to clear up these issues.

[Rebuttal to Ars article](#)

One example of some of these misunderstandings is Jon "Hannibal" Stokes' article at Arstechnica. He has assumed this is an "analysis" (his words) and that's just the title. He later chastises me for indicating the difference between local memory and cache, maybe he should read the patent. Jon's work is usually excellent, I guess everyone has off days.

© Nicholas Blachford 2005.

Cell Architecture Explained - Part 1: Inside The Cell

Getting the details on Cell is not that easy. The initial announcements were vague to say the least and it wasn't until a patent [Cell Patent] appeared that any details appeared, most people wouldn't have noticed this but the inquirer ran a story on it [INQ].

Unfortunately the patent reads like it was written by a robotic lawyer running Gentoo in text mode, you don't so much read it as decipher it. On top of this the patent does not give the details of what the final system will look like though it does describe a number of different options.

With the recent announcements about a new Cell workstation and some details [Recent Details] and specifications [Specs] being revealed it's now possible to have a look at how a Cell based system may look like in the flesh.

The patent is a long and highly confusing document but I think I've managed to understand it sufficiently to describe the system. It's important to note though that the actual Cell processors may be different from the description I give as the patent does not describe everything and even if it did things can and do change.

Background

Although it's been primarily touted as the technology for the PlayStation 3, Cell is designed for much more. Sony and Toshiba, both being major electronics manufacturers buy in all manner of different components, one of the reasons for Cell's development is they want to save costs by building their own components. Next generation consumer technologies such as BluRay, HDTV, HD Camcorders and of course the PS3 will all require a very high level of computing power and this is going to need chips to provide it. Cell will be used for all of these and more, IBM will also be using the chips in servers and they can also be sold to 3rd party manufacturers [3rd party].

Sony and Toshiba previously co-operated on the PlayStation 2 but this time the designs are a more aggressive and required the help of a third partner to help design and manufacture the new chips. IBM brings not only it's chip design expertise but also it's industry leading silicon process and their ability to get things to work - when even the biggest chip firms in the industry have problems it's IBM who get the call to come and help. the companies they've helped is a who's who of the semiconductor industry.

The amount of money being spent on this project is vast, two 65nm chip fabrication facilities are being built at billions each and Sony has paid IBM hundreds of millions to set up a production line in Fishkill. Then there's a few hundred million on development - all before a single chip rolls of the production lines.

So, what is Cell Architecture

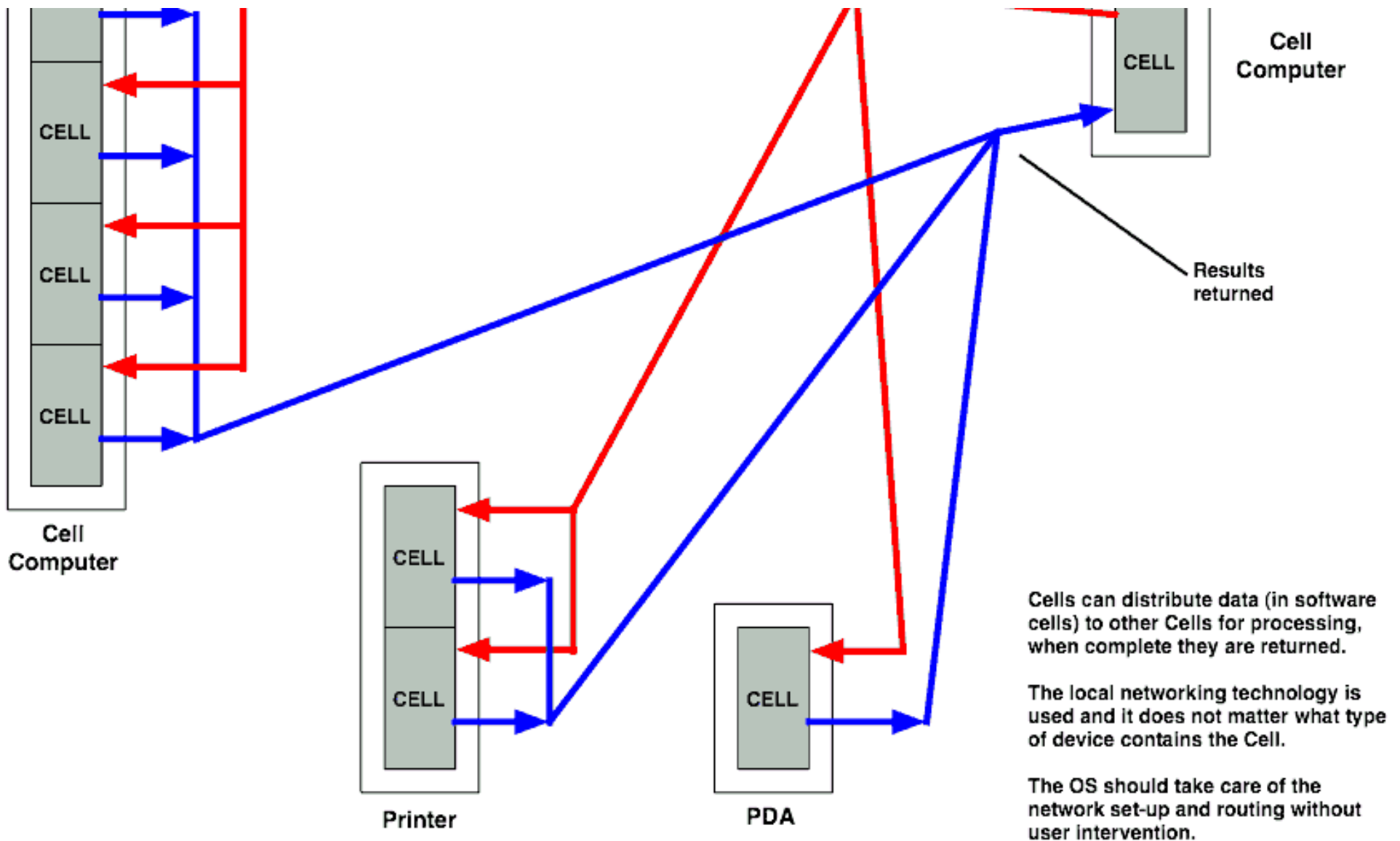
Cell is an architecture for high performance distributed computing. It is comprised of hardware and software Cells, software Cells consist of data and programs (known as apulets), these are sent out to the hardware Cells where they are computed and results returned.

This architecture is not fixed in any way, if you have a computer, PS3 and HDTV which have Cell processors they can co-operate on problems. They've been talking about this sort of thing for years of course but the Cell is actually designed to do it. I for one quite like the idea of watching "Contact" on my TV while a PS3 sits in the background churning through a SETI@home [SETI] unit every 5 minutes. If you know how long a SETI unit takes your jaw should have just hit the floor, suffice to say, Cells are very, very fast [Calc].

It can go further though, there's no reason why your system can't distribute software Cells over a network or even all over the world. The Cell is designed to fit into everything from PDAs up to servers so you can make an ad-hoc Cell computer out of completely different systems.

Distributed Processing with Cells





© Nicholas Blachford 2005

Scaling is just one capability of Cell, the individual systems are going to be potent enough on their own. The single unit of computation in a Cell system is called a Processing Element (PE) and even an individual PE is one hell of a powerful processor, they have a theoretical computing capability of 250 GFLOPS (Billion Floating Point Operations per Second) [GFLOPS]. In the computing world quoted figures (bandwidth, processing, throughput) are often theoretical maximums and rarely if ever met in real life. Cell may be unusual in that given the right type of problem they may actually be able to get close to their maximum computational figure.

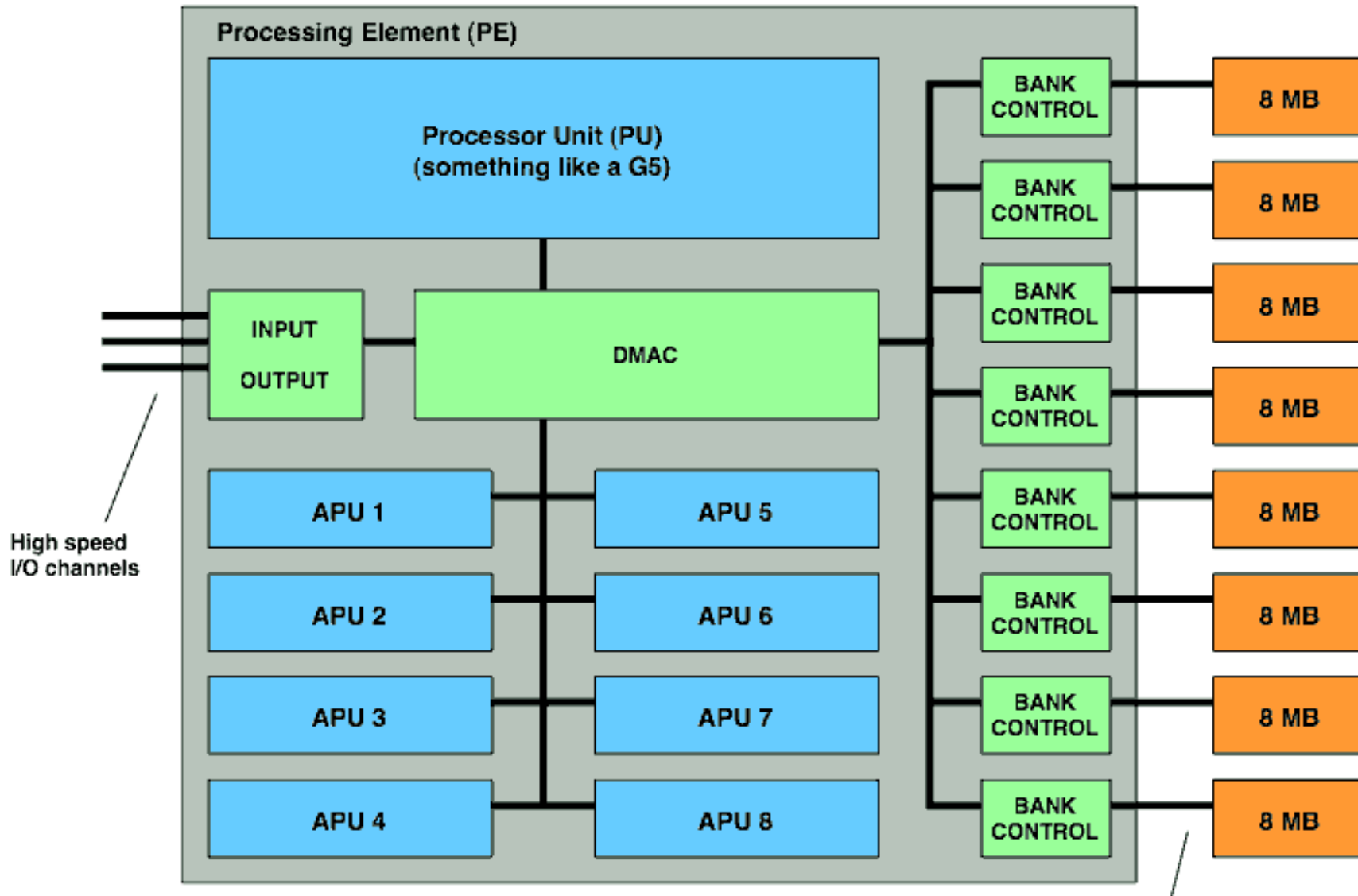
Specifications

An individual Processing Element (i.e. Hardware Cell) is made up of a number of elements:

- 1 Processing Unit (PU)
- 8 X Attached Processing Units (APUs)
- Direct memory Access Controller DMAC
- Input/Output (I/O) Interface

Cell Processor Architecture

(This is a guess since no details have been released as yet)
65nm PS3 chips may have 2 Cells per chip.



© Nicholas Blachford 2005

Multiple high speed memory busses
give Cell huge memory bandwidth.

The full specifications haven't been given out yet but some details [Specs] are out there:

- 4.6 GHz
- 1.3v
- 85 Celcius operation with heat sink
- 6.4 Gigabit / second off-chip communication

All those internal processing units need to be fed so a high speed memory and I/O system is an absolute necessity. for this purpose Sony and Toshiba have licensed the high speed "Yellowstone" and "Redwood" technologies from Rambus [Rambus], the 6.4 Gb/s I/O was also designed in part by Rambus.

The Processor Unit (PU)

As we now know [Recent Details] the PU is a 64bit "Power Architecture" processor. Power Architecture is a catch all term IBM have been using for a while to describe both PowerPC and POWER processors. Currently there's only 3 CPUs which fit this description: POWER5, POWER4 and the PowerPC 970 (aka G5) which itself is a derivation of the POWER4.

The IBM press release indicates the Cell processor is "Multi-thread, multi-core" but since the APUs are almost certainly not multi-threaded it looks like the PU may be based on a POWER5 core - the very same core I expect to turn up in Apple machines in the form of the G6 [G6] in the not too distant future, IBM have acknowledged such a chip is in development but as if to confuse us call it a "next generation 970".

There is of course the possibility that IBM have developed a completely different 64 bit CPU which it's never mentioned before. This isn't a far fetched idea as this is exactly the sort of thing IBM tend to do, i.e. the 440 CPU used in the BlueGene supercomputer is still called a 440 but is very different from the chip you find in embedded systems.

If the PU is based on a POWER design don't expect it to run at a high clock speed, POWER cores tend to be rather power hungry so it may be clocked down to keep power consumption down.

The PlayStation 3 is touted to have 4 Cells so a system could potential have 4 POWER5 based cores. This sounds pretty amazing until you realise that the PUs are really just controllers - the real action is in the APUs...

Attached Processor Units (APU)

Each Cell contains 8 APUs. An APU is a self contained vector processor which acts independently from the others. They contain 128 X 128 bit registers, there are also 4 floating point units capable of 32 GigaFlops and 4 Integer units capable of 32 GOPS (Billions of Operations per Second). The APUs also include a small 128 Kilobyte local memory instead of a cache, there is also no virtual memory system used at runtime.

Independent processing

The APUs are not coprocessors, they are complete independent processors in their own right. The PU sets them up with a software Cell and then "kicks" them into action. Once running the APU executes the apulet in the software Cell until it is complete or it is told to stop. The PU sets up the APUs using Remote Procedure calls, these are not sent directly to the APUs but rather sent via the DMAC which also performs any memory reads or writes required.

Vector processing

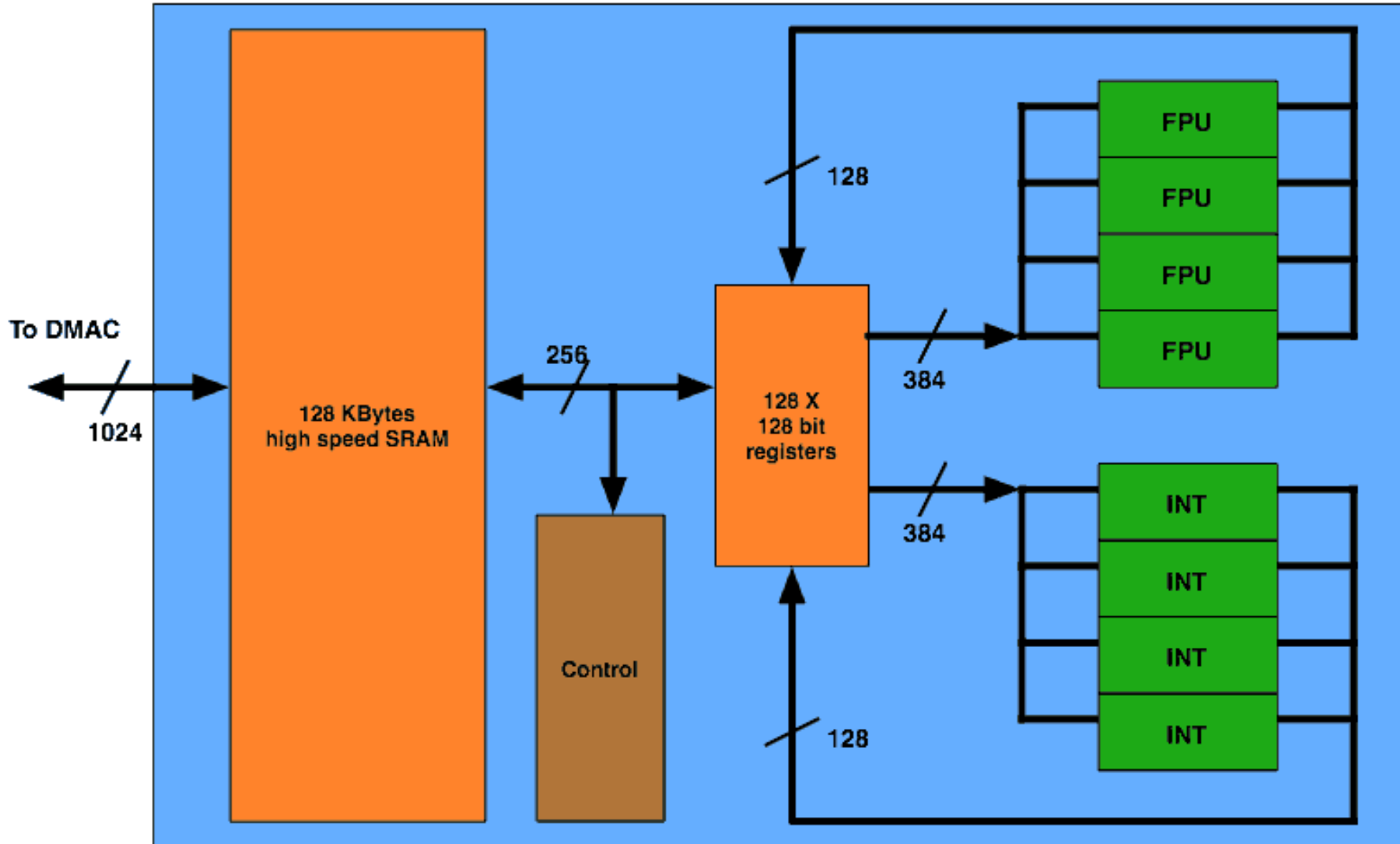
The APUs are vector [Vector] (or SIMD) processors, that is they do multiple operations simultaneously with a single instruction. Vector computing has been used in supercomputers since the 1970s and modern CPUs have media accelerators (e.g. SSE, AltiVec) which work on the same principle. Each APU appears to be capable of 4 X 32 bit operations per cycle, (8 if you count multiply-adds). In order to work, the programs run will need to be "vectorised", this can be done in many application areas such as video, audio, 3D graphics and many scientific areas.

AltiVec?

It has been speculated that the vector units are the same as the AltiVec units found in the PowerPC G4 and G5 processors. I consider this highly unlikely as there are several differences. Firstly the number of registers is 128 instead of AltiVec's 32, secondly the APUs use a local memory whereas AltiVec does not, thirdly AltiVec is an add-on to the existing PowerPC instruction set and operates as part of a PowerPC processor, the APUs are completely independent processors. There will no doubt be a great similarity between the two but don't expect any direct compatibility. It should however be relatively simple to convert between the two.

Cell APU Architecture

Each APU is an independent vector CPU capable of 32 GFLOPs or 32 GOPs.



APU Local memory

The lack of cache and virtual memory systems means the APUs operate in a different way from conventional CPUs. This will likely make them harder to program but they have been designed this way to reduce complexity and increase performance.

Conventional Cache

Conventional CPUs perform all their operations in registers which are directly read from or written to main memory, operating directly on main memory is hundreds of times slower so caches (a fast on chip memory of sorts) are used to hide the effects of going to or from main memory. Caches work by storing part of the memory the processor is working on, if you are working on a 1MB piece of data it is likely only a small fraction of this (perhaps a few hundred bytes) will be present in cache, there are kinds of cache design which can store more or even all the data but these are not used as they are too expensive or too slow.

If data being worked on is not present in the cache the CPU stalls and has to wait for this data to be fetched. This essentially halts the processor for hundreds of cycles. It is estimated that even high end server CPUs (POWER, Itanium, typically with very large fast caches) spend anything up to 80% of their time waiting for memory.

Dual-core CPUs will become common soon and these usually have to share the cache. Additionally, if either of the cores or other system components try to access the same memory address the data in the cache may become out of date and thus needs updated (made coherent).

Supporting all this complexity requires logic and takes time and in doing so this limits the speed that a conventional system can access memory, the more processors there are in a system the more complex this problem becomes. Cache design in conventional CPUs speeds up memory access but compromises are made to get it to work.

APU local memory - no cache

To solve the complexity associated with cache design and to increase performance the Cell designers took the radical approach of not including any. Instead they used a series of local memories, there are 8 of these, 1 in each APU.

The APUs operate on registers which are read from or written to the local memory. This local memory can access main memory in blocks of 1024 bits but the APUs cannot act directly on main memory.

By not using a caching mechanism the designers have removed the need for a lot of the complexity which goes along with a cache. The local memory can only be accessed by the individual APU, there is no coherency mechanism directly connected to the APU or local memory.

This may sound like an inflexible system which will be complex to program and it most likely is but this system will deliver data to the APU registers at a phenomenal rate. If 2 registers can be moved per cycle to or from the local memory it will in it's first incarnation deliver 147 Gigabytes per second. That's for a single APU, the aggregate bandwidth for all local memories will be over a Terabyte per second - no CPU in the consumer market has a cache which will even get close to that figure. The APUs need to be fed with data and by using a local memory based design the Cell designers have provided plenty of it.

Coherency

While there is not coherency mechanism in the APUs a mechanism does exist. To prevent problems occurring when 2 APUs use the same memory, a mechanism is used which involves some extra data stored in the RAM and an extra "busy" bit in the local storage. There are quite a number of diagrams to look at and a detailed explanation in the patent if you wish to read up on the exact mechanism used. However the system is a much simpler system than trying to keep caches up to date since it essentially just marks data as either readable or not and lists which APU tried to get it.

The system can complicate memory access though and slow it down, the additional data stored in RAM could be moved on chip to speed things up but may not be worth the extra silicon and subsequent cost at this point in time.

Little is known at this point about the PUs apart from being "Power architecture" but being a conventional CPU design I think it's safe to assume there will be perfectly normal cache and coherency mechanism used within them (presumably modified for the memory subsystem).

APUs on their own being well fed with data will make for some highly potent processors. But...

APUs can also be chained, that is they can be set up to process data in a stream using multiple APUs in parallel. In this mode a Cell may approach it's theoretical maximum processing speed of 250 GigaFlops. In part 2 I shall look at this, the rest of the internals of the Cell and other aspects of the architecture.

[Introduction and Index](#)

[Part 1: Inside The Cell](#)

[Part 2: Again Inside The Cell](#)

[Part 3: Cellular Computing](#)

[Part 4: Cell Vs the PC](#)

[Part 5: Conclusion and References](#)

[Part 6: Updates, Clarifications and Missing Bits](#)

© Nicholas Blachford 2005.

Cell Architecture Explained - Part 2: Again Inside The Cell

Stream Processing

A big difference in Cells from normal CPUs is the ability of the APUs in a Cell to be chained together to act as a stream processor [Stream]. A stream processor takes data and processes it in a series of steps. Each of these steps can be performed by one or more APUs.

A Cell processor can be set-up to perform streaming operations in a sequence with one or more APUs working on each step. In order to do stream processing an APU reads data from an input into it's local memory, performs the processing step then writes it to a pre-defined part of RAM, the second APU then takes the data just written, processes it and writes to a second part of RAM. This sequence can use many APUs and APUs can read or write different blocks of RAM depending on the application. If the computing power is not enough the APUs in other cells can also be used to form an even longer chain.

Stream processing does not generally require large memory bandwidth but Cell will have it anyway. According to the patent each Cell will have access to 64 Megabytes directly via 8 bank controllers (it indicates this as an "ideal", the maximum may be higher). If the stream processing is set up to use blocks of RAM in different banks, different APUs processing the stream can be reading and writing simultaneously to the different blocks.

So you think your PC is fast...

It is where multiple memory banks are being used and the APUs are working on compute heavy streaming applications that the Cell will be working hardest. It's in these applications that the Cell may get close to it's theoretical maximum performance and perform over an order of magnitude more calculations per second than any desktop processor currently available.

If over clocked sufficiently (over 3.0GHz) and using some very optimised code (SSE assembly), 5 dual core Opterons directly connected via HyperTransport should be able to achieve a similar level of performance in stream processing as a **single** Cell - Admittedly, this is purely theoretical and it depends on the Cell achieving it's performance goals and a "perfect" application being used, it does however demonstrate the sort of processing capability the Cell potentially has.

The PlayStation 3 is expected to have have 4 Cells.

General purpose desktop CPUs are not designed for high performance vector processing. They all have vector units on board in the shape of SSE or AltiVec but this is integrated on board and has to share the CPUs resources. The APUs are dedicated high speed vector processors and with their own memory don't need to share anything other than the memory. Add to this the fact there are 8 of them and you can see why their computational capacity is so large.

Such a large performance difference may sound completely ludicrous but it's not without precedent, in fact if you own a reasonably modern graphics card your existing system is be capable of a lot more than you think:

"For example, the nVIDIA GeForce 6800 Ultra, recently released, has been observed to reach 40 GFlops in fragment processing. In comparison, the theoretical peak performance of the Intel 3GHz Pentium4 using SSE instructions is only 6GFlops." [GPU]

The 3D Graphics chips in computers have long been capable of very much higher performance than general purpose CPUs. Previously they were restricted to 3D graphics processing but since the addition of shaders people have been using them for more general purpose tasks [GPGPU], this has not been without some difficulties but Shader 4.0 parts are expected to be a lot more general purpose than before.

Existing GPUs can provide massive processing power when programmed properly, the difference is the Cell will be cheaper and several times faster.

Hard Real Time Processing

Some stream processing needs to be timed exactly and this has also been considered in the design to allow

"hard" real time data processing. An "absolute timer" is used to ensure a processing operation falls within a specified time limit. This is useful on it's own but also ensures compatibility with faster next generation cells since the timer is independent of the processing itself.

Hard real time processing is usually controlled by specialist operating systems such as QNX which are specially designed for it. Cell's hardware support for it means pretty much any OS will be able to support it to some degree. This will however only apply to tasks using the APUs so I don't see QNX going away anytime soon.

The DMAC

The DMAC (Direct Memory Access Controller) is a very important part of the Cell as it acts as a communications hub. The PU doesn't issue instructions directly to the APUs but rather issues them to the DMAC and it takes the appropriate actions, this makes sense as the actions usually involve loading or saving data. This also removes the need for direct connections between the PU and APUs.

As the DMAC handles all data going into or out of the Cell it needs to communicate via a very high bandwidth bus system. The patent does not specify the exact nature of this bus other than saying it can be either a normal bus or it can be a packet switched network. The packet switched network will take up more silicon but will also have higher bandwidth, I expect they've gone with the latter since this bus will need to transfer 10s of Gigabytes per second. What we do know from the patent is that this bus is huge, the patent specifies it at a whopping 1024 bits wide.

At the time the patent was written it appears the architecture for the DMAC had not been fully worked out so as well as two potential bus designs the DMAC itself has different designs. Distributed and centralised architectures for the DMAC are both mentioned.

It's clear to me that the DMAC is one of the most important parts of the Cell design, it doesn't do processing itself but has to contend with 10's of Gigabytes of memory flowing through it at any one time to many different destinations, if speculation is correct the PS3 will have 100GByte / second memory interface, if this is spread over 4 Cells that means each DMAC will need to handle at least 25 Gigabytes per second. It also has to handle the memory protection scheme and be able to issue memory access orders as well as handling communication between the PU and APUs, it needs to be not only fast but will also be a highly complex piece of engineering.

Memory

As with everything else in the Cell architecture the memory system is designed for raw speed, it will have both low latency and very high bandwidth. As mentioned previously memory is accessed in blocks of 1024 bits. The reason for this is not mentioned in the patent but I have a theory:

While this may reduce flexibility it also decreases memory access latency - the single biggest factor currently holding back computers today. The reason it's faster is the finer the address resolution the more complex the logic and the longer it takes to look it up. The actual looking up may be insignificant on the memory chip but each look-up requires a look-up transaction which involves sending an address from the bank controller to the memory device and this will take time. This time is significant itself as there is one per memory access but what's worse is that every bit of address resolution doubles the number of look-ups required.

If you have 512MB in your PC your RAM look-up resolution is 29 bits*, however the system will read a minimum of 64 bits at a time so resolution is 26 bits. The PC will probably read more than this so you can probably really say 23 bits.

* Note: I'm not counting I/O or graphics address space which will require an extra bit or two.

In the Cell design there are 8 banks of 8MB each and if the minimum read is 1024 bits the resolution is 13 bits. An additional 3 bits are used to select the bank but this is done on-chip so will have little impact. Each bit doubles the number of memory look-ups so the PC will be doing a thousand times more memory look-ups per second than the Cell does. The Cell's memory busses will have more time free to transfer data and thus will work closer to their maximum theoretical transfer rate. I'm not sure my theory is correct but CPU caches use a

similar trick.

What is not theoretical is the fact the Cell will use very high speed memory connections - Sony and Toshiba licensed 3.2GHz memory technology from Rambus in 2003 [Rambus]. If each cell has total bandwidth of 25.6 Gigabytes per second each bank transfers data at 3.2 Gigabytes per second. Even given this the buses are not large (64 data pins for all 8), this is important as it keeps chip manufacturing costs down.

100 Gigabytes per second sounds huge until you consider top end graphics cards are in the region of 50 Gigabytes per second already, doubling over a couple of years sounds fairly reasonable. But these are just the theoretical figures and never get reached, assuming the system I described above is used the bandwidth on the Cell should be much closer to it's theoretical figure than competing systems and thus will perform better.

APUs may need to access memory from different Cells especially if a long stream is set up, thus the Cells include a high speed interconnect. Details of this are not known other than they transfer data at 6.4 Gigabits / second *per wire*. I expect there will be busses of these between each Cell to facilitate the high speed transfer of data to each other. This technology sounds not entirely unlike HyperTransport though the implementation may be very different.

In addition to this a switching system has been devised so if more than 4 Cells are present they too can have fast access to memory. This system may be used in Cell based workstations. It's not clear how more than 8 cells will communicate but I imagine the system could be extended to handle more. IBM have announced a single rack based workstation will be capable of up to 16 TeraFlops, they'll need 64 Cells for this sort of performance so they have obviously found some way of connecting them.

Memory Protection

The memory system also has a memory protection scheme implemented in the DMAC. Memory is divided into "sandboxes" and a mask used to determine which APU or APUs can access it. This checking is performed in the DMAC before any access is performed, if an APU attempts to read or write the wrong sandbox the memory access is forbidden.

Existing CPUs include hardware memory protection system but it is a lot more complex than this. They use page tables which indicate the use of blocks of RAM and also indicate if the data is in RAM or on disc, these tables can become large and don't fit on the CPU all at once, this means in order to read a memory location the CPU may first have to read a page table from memory and read data in from disc - all before the data required is read.

In the Cell the APU can either issue a memory access or not, the table is held in a special SRAM in the DMAC and is never flushed. This system may lack flexibility but is very simple and consistently very fast.

This simple system most likely only applies to the APUs, I expect the PU will have a conventional memory protection system.

Software Cells

Software cells are containers which hold data and programs called apulets as well as other data and instructions required to get the apulet running (memory required, number of APUs used etc.). The cell contains source, destination and reply address fields, the nature of these depends on the network in use so software Cells can be sent around to different hardware Cells. There are also network independent addresses which will define the specific Cell exactly. This allows you to say, send a software Cell to hardware Cell in a specific computer on a network.

The APUs use virtual addresses but these are mapped to a real address as soon as DMA commands are issued. The software Cell contains these DMA commands which retrieve data from memory to process, if APUs are set up to process streams the Cell will contain commands which describe where to read data from and where to write results to. Once set up, the APUs are "kicked" into action.

It's not clear how this system will operate in practice but it would appear to include some adaptively so as to allow Cells to appear and disappear on a network.

This system is in effect a basic Operating System but could be implemented as a layer within an existing OS. There's no reason to believe Cell will have any limitations regarding which Operating Systems can run.

Multi-Cell'd animals

One of the main points of the entire Cell architecture is parallel processing. Software cells can be sent pretty much anywhere and don't depend on a specific transport means. The ability of software Cells to run on hardware Cells determined at runtime is a key feature of the Cell architecture. Want more computing power? Plug in a few more Cells and there you are.

If you have a bunch of cells sitting around talking to each other via WiFi connections the system can use it to distribute software cells for processing. The system was not designed to act like a big iron machine, that is, it is not arranged around a single shared or closely coupled set of memories. All the memory may be addressable but each Cell has it's own memory and they'll work most efficiently in their own memory or at least in small groups of Cells where fast inter-links allow the memory to be shared.

Going above this number of Cells isn't described in detail but the mechanism present in the software Cells to make use of whatever networking technology is in use allows ad-hoc arrangements of Cells to be made without having to worry about rewriting software to take account of different network types.

The parallel processing system essentially moves a lot of complexity which would normally be handled by hardware and moves it into software. This usually slows things down but the benefit is flexibility, you give the system a set of software Cells to compute and it figures out how to distribute them itself. If your system changes (Cells added or removed) the OS should take care of this without user or programmer intervention.

Writing software for parallel processing is usually highly difficult and this helps get around the problem. You still, of course have to parallelise the program into cells but once that's done you don't have to worry if you have one Cell or ten.

In the future, instead of having multiple discrete computers you'll have multiple computers acting as a single system. Upgrading will not mean replacing an old system anymore, it'll mean enhancing it. What's more your "computer" may in reality also include your PDA, TV and Camcorder all co-operating and acting as one.

Concrete Processing

The Cell architecture goes against the grain in many areas but in one area it has gone in the complete opposite direction to the rest of the technology industry. Operating systems started as a rudimentary way for programs to talk to hardware without developers having to write their own drivers every time. As time went on operating systems have evolved and taking on a wide variety of complex tasks, one way it has done this is by abstracting more and more away from the hardware.

Object oriented programming goes further and abstracts individual parts of programs away from each other. This has evolved into Java like technologies which provide their own environment thus abstracting the application away from the individual operating system. Web technologies do the same thing, the platform which is serving you with this page is completely irrelevant, as is the platform viewing it. When writing this I did not have to make a Windows or Mac specific version of the HTML, the underlying hardware, OSs and web browsers are completely abstracted away.

Even hardware manufacturers have taken to abstraction, the Transmeta line of CPUs are sold as x86 CPUs but in reality they are not. They provide an abstraction in software which hides the inner details of the CPU which is not only not x86 but a completely different architecture. This is not unique to Transmeta or even x86, the internal architecture of most modern CPUs is very different from their programming model.

If there is a law in computing, Abstraction is it, it is an essential piece of today's computing technology, much of what we do would not be possible without it. Cell however, has abandoned it. The programming model for the Cell will be *concrete*, when you program an APU you will be programming what is in the APU itself, not some abstraction. You will be "hitting the hardware" so to speak.

While this may sound like sacrilege and there are reasons why it is a bad idea in general there is one big advantage: Performance. Every abstraction layer you add adds computations and not by some small measure, an abstraction can decrease performance by a factor of ten fold. Consider that in any modern system there are multiple abstraction layers on top of one another and you'll begin to see why a 50MHz 486 may of seemed fast years ago but runs like a dog these days, you need a more modern processor to deal with the subsequently added abstractions.

The big disadvantage of removing abstractions is it will significantly add complexity for the developer and it limits how much the hardware designers can change the system. The latter has always been important and is essentially THE reason for abstraction but if you've noticed modern processors haven't really changed much in years. The Cell designers obviously don't expect their architecture to change significantly so have chosen to set it in stone from the beginning. That said there is some flexibility in the system so it can change at least partially.

The Cell approach does give some of the benefits of abstraction though. Java has achieved cross platform compatibility by abstracting the OS and hardware away, it provides a "virtual machine" which is the same across all platforms, the underlying hardware and OS can change but the virtual machine does not.

Cell provides something similar to Java but in a completely different way. Java provides a software based "virtual machine" which is the same on all platforms, Cell provides a machine as well - but they do it in hardware, the equivalent of Java's virtual machine is the Cells physical hardware. If I was to write Cell code on OS X the exact same Cell code would run on Windows, Linux or Zeta because in all cases it is the hardware Cells which execute it.

It should be pointed out that this does not mean you have to program the Cells in assembly, Cells will have compilers just like everything else. Java provides a virtual machine but you don't program it directly either.

DRM In The Hardware

Some will no doubt be turned off by the fact that DRM is built into the Cell hardware. Sony is a media company and like the rest of the industry that arm of the company are no doubt pushing for DRM type solutions. It must also be noted that the Cell is destined for HDTV and BluRay / HD-DVD systems, any high definition recorded content is going to be very strictly controlled by DRM so Sony have to add this capability otherwise they would be effectively locking themselves out of a large chunk of their target market. Hardware DRM is no magic bullet however, hardware systems have been broken before - including Set Top Boxes and even IBM's crypto hardware for their mainframes.

Other Options And The Future

There are plans for future technology in the Cell architecture, optical interconnects appear to be planned, it's doubtful that this will appear in PS3 but clearly the designers are planning for the day when copper wires hit their limit (thought to be around 10GHz) Other materials than Silicon also appear to be being considered for fabrication but this will be an even bigger undertaking.

The design of Cells is not entirely set in stone, there can be variable numbers of APUs and the APUs themselves can include more floating point or integer calculation units. In some cases APUs can be removed and other things such as I/O units or graphics processor placed in their place. Nvidia are proving the graphics hardware for the PS3 so this may be done within a modified Cell at some point.

As Moore's law moves forward and we get yet more transistors per chip I've no doubt the designers will take advantage of this. The idea of having 4 Cells per chip is mentioned in the patent but there are other options also for different applications of the Cell.

When multiple APUs are operating on streaming data it appears they write to RAM and read back again, it would be perfectly feasible however to add buffers to allow direct APU to APU writes. Direct transfers are mentioned in the patent but nothing much is said about them.

To Finish Up

The Cell architecture is essentially a general purpose PowerPC CPU with a set of 8 very high performance vector processors and a fast memory and I / O system, this is coupled with a very clever task distribution system which allows ad-hoc clusters to be set up.

What is not immediately apparent is the aggressiveness of the design. The lack of cache and runtime virtual memory system is highly unusual and has not done on any modern general purpose CPU in the last 20 years. It can only be compared with the sorts of designs Seymour Cray produced. The Cell is not only going to be very fast, but because of the highly aggressive design the rest of the industry is going to have a very hard time catching up with it*.

To sum up there's really only one way of saying it:

This system isn't just going to rock, it's going to play German heavy metal.

* I expand on this line of thought in Part 4

[Introduction and Index](#)

[Part 1: Inside The Cell](#)

[Part 2: Again Inside The Cell](#)

[Part 3: Cellular Computing](#)

[Part 4: Cell Vs the PC](#)

[Part 5: Conclusion and References](#)

[Part 6: Updates, Clarifications and Missing Bits](#)

© Nicholas Blachford 2005.

Cell Architecture Explained - Part 3: Cellular Computing

The Cell is not a fancy graphics chip, it is intended for general purpose computing. As if to confirm this the graphics hardware in the PlayStation 3 is being provided by Nvidia [Nvidia]. The APUs are not truly general purpose like normal microprocessors but the Cell makes up for this by virtue of including a PU which is a normal PowerPC microprocessor.

Cell Applications

As I said in part 1, the Cell is destined for uses other than just the PlayStation 3. But what sort of applications Cell will be good for?

Cell will not work well for everything, some applications cannot be vectorised at all, for others the system of reading memory blocks could potentially cripple performance. In cases like these I expect the PU will be used but that's not entirely clear as the patent seems to assume the PU can only be used by the OS.

Games

Games are an obvious target, the Cell was designed for a games console so if they don't work well there's something wrong! The Cell designers have concentrated on raw computing power and not on graphics, as such we will see hardware functions moved into software and much more flexibility being available to developers. Will the PS3 be the first console to get real-time ray traced games?

3D Graphics

Again this is a field the Cell was largely designed for so expect it to do well here, Graphics is an "embarrassingly parallel", vectorisable and streamable problem so all the APUs will be in full use, the more Cells you use the faster the graphics will be. There is a lot of research into different advanced graphics techniques these days and I expect Cells will be used heavily for these and enable these techniques to make their way into the mainstream. If you think graphics are good already you're in for something of a surprise.

Video

Image manipulations can be vectorised and this can be shown to great effect in Photoshop. Video processing can similarly be accelerated and Apple will be using the capabilities of existing GPUs (Graphics Processor Units) to accelerate video processing in "core image", Cell will almost certainly be able to accelerate anything GPUs can handle.

Video encoding and decoding can also be vectorised so expect format conversions and mastering operations to benefit greatly from a Cell. I expect Cells will turn up in a lot of professional video hardware.

Audio

Audio is one of those areas where you can never have enough power. Today's electronic musicians have multiple virtual synthesisers each of which has multiple voices. Then there's traditionally synthesised, sampled and real instruments. All of these need to be handled and have their own processing needs, that's before you put different effects on each channel. Then you may want global effects and compression per channel and final mixing. Many of these processes can be vectorised. Cell will be an absolute dream for musicians and yet another headache for synthesiser manufacturers who have already seen PCs encroaching on their territory.

DSP (Digital Signal Processing)

The primary algorithm used in DSP is the FFT (Fast Fourier transform) which breaks a signal up into individual frequencies for further processing. The FFT is a highly vectorisable algorithm and is used so much that many vector units and microprocessors contains instructions especially for accelerating this algorithm.

There are thousands of different DSP applications and most of them can be streamed so Cell can be used for many of these applications. Once prices have dropped and power consumption has come down expect the Cell to be used in all manner for different consumer and industrial devices.

SETI

A perfect example of a DSP application, again based on FFTs, a Cell will boost my SETI@home [SETI] score no end! As mentioned elsewhere I estimate a set of 4 Cells will complete a unit in under 5 minutes [Calc]. Numerous other distributed applications will also benefit from the Cell.

Scientific

For conventional (non vectorisable) applications this system will be at least as fast as 4 PowerPC 970s with a fast memory interface. For vectorisable algorithms performance will go onto another planet. A potential problem however will be the relatively limited memory capability (this may be PlayStation 3 only, the Cell may be able to address larger memories). It is possible that even a memory limited Cell could be used perfectly well by streaming data into and out of the I/O unit.

GPUs are already used for scientific computation and Cell will be likely be useable in the same areas: *"Many kinds of computations can be accelerated on GPUs including sparse*

linear system solvers, physical simulation, linear algebra operations, partial difference equations, fast Fourier transform, level-set computation, computational geometry problems, and also non-traditional graphics, such as volume rendering, ray-tracing, and flow visualization. [GPU]

Super Computing

Many modern supercomputers use clusters of commodity PCs because they are cheap and powerful. You currently need in the region of 250 PCs to even get onto the top 500 supercomputer list [Top500]. It should take just 8 Cells to get onto the list and 560 to take the lead*. This is one area where backwards compatibility is completely unimportant and will be one of the first areas to fall, expect Cell based machines to rapidly take over the Top 500 list from PC based clusters.

There are other super computing applications which require large amounts of interprocess communication and do not run well in clusters. The Top500 list does not measure these separately but this is an area where big iron systems do well and Cray rules, PC clusters don't even get a look-in. The Cells have high speed communication links and this makes them ideal for such systems although additional engineering will be required for large numbers of Cells. Cells may not only take over from PC clusters but also expect them to do well here also.

If the Cell has a 64 bit Multiply-add instruction (I'd be very surprised if this wasn't present) it'll take 8000 of them to get a PetaFlop**. That record will be very difficult to beat.

** Based on theoretical values, in reality you'd need more Cells depending on the efficiency.

Servers

This is one area which does not strike me as being terribly vectorisable, indeed XML and similar processing are unlikely to be helped by the APUs at all though the memory architecture may help (which is unusual given how amazingly inefficient XML is). However servers generally do a lot of work in their database backend.

Commercial databases with real life data sets have been studied and found to have been benefited from running on GPUs. You can also expect these to be accelerated by Cells. So yes, even servers can benefit from Cells.

Stream Processing Applications

A big difference from normal CPUs is the ability of the APUs in a cell to be chained together to act as a stream processor [Stream]. A stream processor takes a flow of data and processes it in a series of steps. Each of these steps can be performed by a different APU or even different APUs on different Cells.

An Example: A Digital TV Receiver

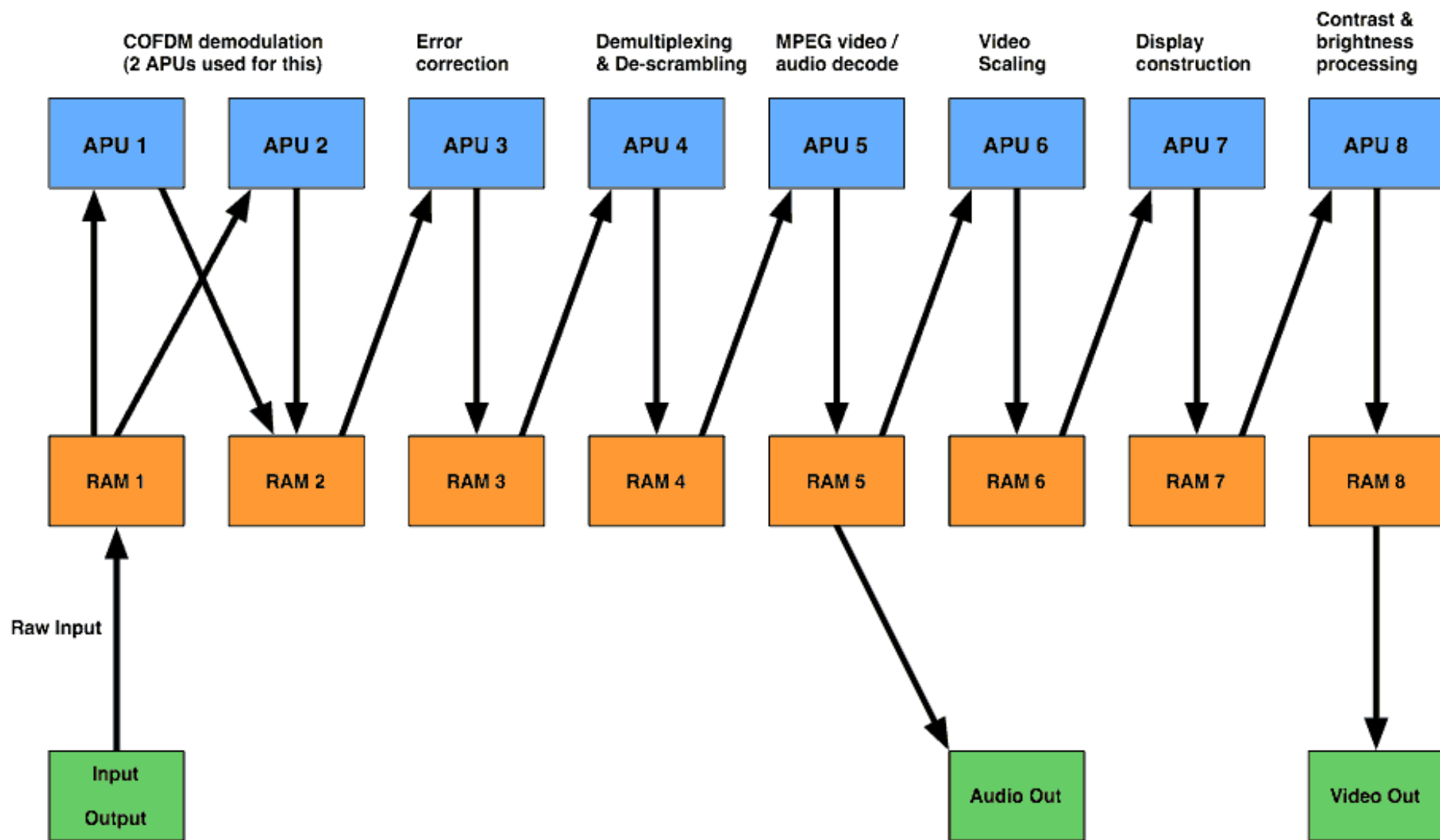
To give an example of stream processing take a Set Top Box for watching Digital TV, this is a lot more complex process than just playing a MPEG movie as a whole host of additional processes are involved. This is what needs to be done before you can watch the latest episode of Star Trek, here's an outline of the processes involved:

- COFDM demodulation
- Error correction
- Demultiplexing
- De-scrambling
- MPEG video decode
- MPEG audio decode
- Video scaling
- Display construction
- Contrast & Brightness processing

These tasks are typically performed using a combination of custom hardware and dedicated DSPs. They can be done in software but it'll take a very powerful CPU if not several of them to do all the processing - and that's just for standard definition MPEG2. HDTV with H.264 will require considerably more processing power. General purpose CPUs tend not to be very efficient so it is generally easier and cheaper to use custom chips, although highly expensive to develop they are cheap when produced in high volumes and consume miniscule amounts of power.

These tasks are vectorisable and working in a sequence are of course streamable. A Cell processor could be set-up to perform these operations in a sequence with one or more APUs working on each step, this means there is no need for custom chip development and new standards can be supported in software. The power of a Cell is such that it is likely that a single Cell will be capable of doing all the processing necessary, even for High definition standards. Toshiba intend on using the Cell for HDTVs.

Stream Processing



© Nicholas Blachford 2005

Non Accelerated Applications

Cell Architecture Explained - Part 3: Cellular Computing

There are going to be many applications which cannot be accelerated by a Cell processor and even those which can may not be ported overnight. I don't for instance expect Cell will even attempt to go after the server market.

But generally PCs either don't need much power or they can be accelerated by the Cell, Intel and AMD will be churning out ever more multi-core'd x86s but what's going to happen if Cells will deliver vastly more power at what will rapidly become a lower price?

The PC is about to have the biggest fight it has ever had. To date it has won with ease every time, this time it will not be so easy. In Part 4 I look at this forthcoming battle royale

[Introduction and Index](#)

[Part 1: Inside The Cell](#)

[Part 2: Again Inside The Cell](#)

[Part 3: Cellular Computing](#)

[Part 4: Cell Vs the PC](#)

[Part 5: Conclusion and References](#)

[Part 6: Updates, Clarifications and Missing Bits](#)

© Nicholas Blachford 2005.

The Cell Processor Explained, Part 4: Cell V's the PC

To date the PC has defeated everything in it's path [PCShare]. No competitor, no matter how good has even got close to replacing it. If the Cell is placed into desktop computers it may be another victim of the PC. However, I think for a number of reasons that the Cell is not only the biggest threat the PC has ever faced, but also one which might actually have the capacity to defeat it.

The Sincerest Form of Flattery is Theft

20 years ago an engineer called Jay Miner who had been working on video games (he designed the Atari 2600 chip) decided to do something better and produce a desktop computer which combined a video game chipset with a workstation CPU. The prototype was called Lorraine and it was eventually released to the market as the Commodore Amiga. The Amiga had hardware accelerated high colour screens, a GUI based multitasking OS, multiple sampled sound channels and a fast 32 bit CPU. At the time PCs had screens displaying text, a speaker which beeped and they ran MSDOS on a 16 bit CPU. The Amiga went on to sell in millions but the manufacturer went bankrupt in 1994.

Like many other platforms which were patently superior to it, the Amiga was swept aside by the PC.

The PC has seen off every competitor that has crossed paths with it, no matter how good the OS or hardware. The Amiga in 1985 was years ahead of the PC, it took more than 5 years for the PC to catch up with the hardware and 10 years to catch up with the OS. Yet the PC still won, as it did against every other platform. The PC has been able to do this because of a huge software base and it's ability to steal the competitors clothes, low prices and high performance were not a factor until much later. If you read the description of the Amiga I gave again you'll find it also describes a modern PC. The Amiga may have introduced specialised chips for graphics acceleration and multitasking to the desktop world but now all computers have them.

In the case of the Amiga it was not the hardware or the price which beat it. It was the vast MSDOS software base which prevented it getting into the business market, Commodore's ability to shoot themselves in the foot finished them off. NeXT came along next with even better hardware and an even better Unix based OS but they couldn't dent the PC either. It was next to be dispatched and again the PC later caught up and stole all it's best features, it took 13 years to bring memory protection to the consumer level PC.

The PC can and does take on the best features of competitors, history has shown that even if this takes a very long time the PC still ultimately wins. Could the PC not just steal the Cell's unique attributes and cast it aside also?

Cell V's x86

This looks like a battle no one can win. x86 has won all of it's battles because when Intel and AMD pushed the x86 architecture they managed to produce very high performance processors and in their volumes they could sell them for low prices. When x86 came up against faster RISC competitors it was able to use the very same RISC technologies to close the speed gap to the point where there was no significant advantage going with RISC.

Three of what were once important RISC families have also been dispatched to the great Fab in the sky. Even Intel's own Itanium has been beaten out of the low / mid server space by the Opteron. Sun have been burned as well, they cancelled the next in the UltraSPARC line, bought in radical new designs and now sell the Opteron which threatened to eclipse their low end. Only POWER seems to be holding it's own but that's because IBM has the resources to pour into it to keep it competitive and it's in the high end market which x86 has never managed to penetrate and may not scale to.

To Intel and AMD's processors Cell presents a completely different kind of competition to what has gone before. The speed difference is so great that nothing short of a complete overhaul of the x86 architecture will be able to bring it even close performance wise. Changes are not unheard of in x86 land but neither Intel or AMD appear to be planning a change even nearly radical enough to catch up. That said Intel recently gained access to many of Nvidia's patents [Intel+Nvidia] and are talking about having dozens of cores per chip so who knows

what Santa Clara are brewing. [Project Z]

Multicore processors are coming to the x86 world soon from both Intel and AMD [MultiCore], but high speed x86 CPUs typically have high power requirements. In order to have 2 Opterons on a single core AMD have had to reduce their clock rate in order to keep them from requiring over a hundred watts, Intel are doing the same for the Pentium 4. The Pentium-M however is a (mostly) high performance low power part and this will go into multi-core devices much easier than the P4, expect to see chips with 2 cores arriving followed by 4 & 8 core designs over the next few years.

Cell will accelerate many commonly used applications by ludicrous proportions compared to PCs. Intel could put 10 cores on a chip and they'll match neither it's performance or price. The APUs are dedicated vector processors, x86 are not. The x86 cores will no doubt include the SSE vector units but these are no match for even a single APU.

Then there's the parallel nature of Cell. If you want more computing power simply add another Cell, the OS will take care of distributing the software Cells to the second or third etc processor. Try that on a PC, yes many OSs will support multiple processors but many applications do not and will need to be modified accordingly - a process which will take many, many years. Cell applications will be written to be scalable from the very beginning as that's how the system works.

Cell may be vastly more powerful than existing x86 processors but history has shown the PC's ability to overcome even vastly better systems. Being faster alone is not enough to topple the PC.

Cell V's Software

The main problem with competing with the PC is not the CPU, it's the software. A new CPU no matter how powerful, is no use without software. The PC has always won because it's always had plenty of software and this has allowed it to see off it's competitors no matter how powerful they were or the advantages they had at the time. The market for high performance systems is very limited, it's the low end systems which sell.

Cell has the power and it will be cheap. But can it challenge the PC without software? The answer to this question would have been simple once, but PC market has changed over time and for a number of reasons Cell is now a threat:

The first reason is Linux. Linux has shown that alternative operating systems can break into the PC software market against Windows, the big difference with Linux though is that it is cross platform. If the software you need runs on linux, switching hardware platforms is no problem as much of the software will still run on different CPUs.

The second reason is cost, other platforms have often used expensive custom components and have been made in smaller numbers. This has put their cost above that of PCs, putting them at immediate disadvantage. Cell may be expensive initially but once Sony and Toshiba's fabs ramp up it will be manufactured in massive volumes forcing the prices down, the fact it's going into the PS3 and TVs is an obvious help for getting the massive volumes that will be required. IBM will also be making Cells and many companies use IBM's silicon process technologies, if truly vast numbers of Cells were required Samsung, Chartered, Infineon and even AMD could manufacture them (provided they had a license of course).

The third reason is power, the vast majority of PCs these don't need the power they provide, Cell will only accentuate this because it will be able to off load most of the intensive stuff to the APUs. What this means is that if you do need to run a specific piece of software you can emulate it. This would have been impossibly slow once but most PC CPUs are already more than enough and with today's advanced JIT based emulators you might not even notice the difference.

The reason many high end PCs are purchased is to accelerate many of the very tasks the Cell will accelerate. You'll also find these power users are more interested in the tools and not the platform, apart from Games these are not areas over which Microsoft has any hold. Given the sheer amount of acceleration a Cell (or set of Cells) can deliver I can see many power users being happy to jump platforms if the software they want is ported or can be emulated.

Cell is going to be cheap, powerful, run many of the same operating systems and if all else fails it can emulate a PC with little noticeable difference, software and price will not be a problem. Availability will also not be a problem, you can buy playstations anywhere. This time round the traditional advantages the PC has held over other systems will not be present, they will have no advantage in performance, software or price. That is not to say that the Cell will walk in and just take over, it's not that simple.

Attack

IBM plan on selling workstations based on the Cell but I don't expect they'll be cheap or sold in any numbers to anyone other than PlayStation developers.

Cell will not just appear in exotic workstations and PlayStations though, I also expect they'll turn up in desktop computers of one kind or another (i.e. I know Genesi are considering doing one). When they do they're going to turn the PC business upside down.

Even with a single Cell it will outgun top end multiprocessor PCs many times over. That's gotta hurt, and it will hurt, Cell is going to effectively make traditional general purpose microprocessors obsolete.

Infection inside

Of course this won't happen overnight and there's nothing to stop PC makers from including a Cell processor on a PCI / PCIe card or even on the motherboard. Microsoft may be less than interested in supporting a competitor but that doesn't mean drivers couldn't be written and support added by the STI partners. Once this is done developers will be able to make use of the Cell in PC applications and this is where it'll get very interesting. With computationally intensive processing moved to the Cell there will be no need for a PC to include a fast x86, a low cost slow one will do just fine.

Some companies however will want to cut costs further and there's a way to do that. The Cell includes at least a PowerPC 970 grade CPU so it'll be a reasonably fast processor. Since there is no need for a fast x86 processor why not just emulate one? Removing the x86 and support chips from a PC will give big cost savings. An x86 computer without an x86 sounds a bit weird but that's never stopped Transmeta who do exactly that, perhaps Transmeta could even provide the x86 emulation technology, they're already thinking of getting out of chip manufacturing [Transmeta].

Cell is a very, very powerful processor. It's also going to become cheap. I fully expect it'll be quite possible to (eventually) build a low cost PC based around a Cell and sell it for a few hundred dollars. If all goes well will Dell sell Cells?

Game on

You could argue gamers will still drive PC performance up but Sony could always pull a fast one and produce a PS3 on a card for the PC. Since it would not depend on the PC's computational or memory resources it's irrelevant how weak or strong they are. Sony could produce a card which turns even the lowest performance PC into a high end gaming machine, If such a product sold in large numbers studios developing for PS3 already may decide they not need to develop a separate version for the PC, the resulting effect on the PC games market could be catastrophic.

While you could use an emulated OS it's always preferable to have a native OS. There's always Linux However Linux isn't really a consumer OS and seems to be having something of a struggle becoming one. There is however another very much consumer ready OS which already runs on a "Power Architecture" CPU: OS X.

Cell V's Apple

The Cell could be Apple's nemesis or their saviour, they are the obvious candidate company to use the Cell. It's perfect for them as it will accelerate all the applications their primary customer base uses and whatever core it uses the the PU will be PowerPC compatible. Cells will not accelerate everything so they could use them as co-processors in their own machines beside a standard G5 / G6 [G6] getting the best of both worlds.

The Core Image technology due to appear in OS X "Tiger" already uses GPUs (Graphics Processor Units) for things other than 3D computations and this same technology could be retargeted at the Cell's APUs. Perhaps

that's why it was there in the first place...

If other companies use Cell to produce computers there is no obvious consumer OS to use, with OS X Apple have - for the second time - the chance to become the new Microsoft. Will they take it? If an industry springs up of Cell based computers not doing so could be very dangerous. When the OS and CPU is different between the Mac and PC there is (well, was) a big gap between systems to jump and a price differential can be justified. If there's a sizeable number of low cost machines capable of running OS X the price differential may prove too much, I doubt even that would be a knockout blow for Apple but it would certainly be bad news (even the PC hasn't managed a knockout).

PC manufacturers don't really care which components they use or OS they run, they just want to sell PCs. If Apple was to "think different" on OS X licensing and get hardware manufacturers using Cells perhaps they could turn Microsoft's clone army against their masters. I'm sure many companies would be only too happy to get released from Microsoft's iron grip. This is especially so if Apple was to undercut them, which they could do easily given the 400% + margins Microsoft makes on their OS.

Licensing OS X wouldn't necessarily destroy Apple's hardware business, there'll always be a market for cooler high end systems [Alien]. Apple also now has a substantial software base and part of this could be used to give added value to their hardware in a similar manner to that done today. Everyone else would just have to pay for it as usual.

In "The Future of Computing" [Future] I argued that the PC industry would come under threat from low cost computers from the far east. The basis of the argument was that in the PC industry Microsoft and Intel both enjoy very large margins. I argued that it's perfectly feasible to make a low cost computer which is "fast enough" for most people's needs and running Linux there would be no Microsoft Tax, provided the system could do what most people need to do it could be made and sold at a sufficiently low price that it will attack the market from below.

A Cell based system running OS X could be nearly as cheap (depending on the price Apple want to charge for OS X) but with Cell's sheer power it will exceed the power of even the most powerful PCs. This system could sell like hot cakes and if it's sufficiently low cost it could be used to sell into the low cost markets which PC makers are now beginning to exploit. There is a huge opportunity for Apple here, I think they'll be stark raving mad not to take it - because if they don't someone else will - Microsoft already have PowerPC experience with the Xbox2 OS...

Cell will have a performance advantage over the PC and will be able to use the PC's advantages as well. With Apple's help it could also run what is arguably the best OS on the market today, at a low price point. The new Mac mini already looks like it's going to sell like hot cakes, imagine what it could do equipped with a Cell...

It looks like the PC could finally have a competitor to take it on, but the PC still has a way of fighting back, PC's are already considerably more powerful than you might think...

The PC Retaliates: Cell V's GPU

The PC does have a weapon with which to respond, the GPU (Graphics Processor Unit). On computational power GPUs will be the only real competitors to the Cell.

GPUs have always been massively more powerful than general purpose processors [PC + GPU][GPU] but since programmable shaders were introduced this power has become available to developers and although designed specifically for graphics some have been using it for other purposes. Future generations of shaders promise even more general purpose capabilities[DirectX Next].

GPUs operate in a similar manner to the Cell in that they contain a number of parallel vector processors called vertex or pixel shaders, these are designed to process a stream of vertices of 3D objects or pixels but many other compute heavy applications can be modified to run instead [EE-GPU].

With aggressive competition between ATI and Nvidia the GPUs are only going to get faster and now "SLI" technology is being used again to pair GPUs together to produce even more computational power.

GPUs will provide the only viable competition to the Cell but even then for a number of reasons I don't think they will be able to catch the Cell.

Cell is designed from the ground up to be more general purpose than GPUs, the APUs are not graphics specific so adapting non 3D algorithms will likely mean less work for developers.

Cell has the main general purpose PU sharing the same fast memory as the APUs. This is distinct from PCs where GPUs have their own high speed memory and can only access main system memory via the AGP bus. PCI Express should speed this up but even this will be limited due to the bus being shared with the CPU. Additionally vendors may not fully support the PCI Express specification, existing GPUs are very slow at moving data from GPU to main memory.

There is another reason I don't think Nvidia or ATI will be able to match the Cell's performance anytime soon. Last time around the PC rapidly caught up with and surpassed the PS2, I think it is one of Sony's aims this time to make that very difficult so, as such Cell has been designed in a highly aggressive manner.

The Cray Factor

The "Cray factor" is something to which Intel, AMD, Nvidia and ATI may have no answer to.

What is apparent from the patent is the approach the designers have taken in developing the Cell architecture. There are many compromises that can be taken when designing a system like this, in almost every case the designers have not compromised and gone for performance, even if the job of the programmers has been made considerably more difficult.

The Cell design is very different from modern microprocessors, seemingly irremovable parts have been changed radically or removed altogether. The rule of computing, fundamental to modern computing - abstraction - is abandoned altogether, no JITs here, you get direct access to the hardware. This is a highly aggressive design strategy, much more aggressive than you'll find in any other system, even in it's heyday the Alpha processor's design was nowhere near this aggressive. In their quest for pure, unadulterated, raw performance the designers have devised a processor which can only be compared to something designed by Seymour Cray [Cray].

To understand why the Cell will be so difficult to catch you have to understand a battle which started way back in the 1960s.

From the 60s to the 90s IBM and Cray battled each other in trying to build the fastest computers. Cray won pretty much every time, he raised the performance bar to the point that the only machines which eventually beat Cray's designs were newer Cray designs.

IBM made flexible business machines, Cray went for less flexible and less feature rich designs in the quest for ultimate speed. If you look at what is planned for future GPUs [DirectX Next] it is very evident they are going for a flexible-features approach - exactly as you'd expect from a system designed by a software company. They are going to be using virtual memory on the GPU and already use a cache for the most commonly used data, in fact GPUs look like they are rapidly becoming like general purpose CPUs.

The Cell approach is the same as the Cray's. Virtual memory takes up space and delays the access to data. Virtual memory is present in the Cell architecture but not at runtime, the OS keeps addresses virtual until a software Cell is executed at which point the real addresses are used for getting to and from memory. Cell also has memory protection but in a limited and simple fashion, a small on-chip memory holds a table indicating which APU can access which memory block, it's small and never flushed, this means it's also very fast.

CPUs and GPUs use a cache memory to hide access to main memory, Cray didn't bother with cache and just made the main memory super fast. Cell uses the same approach, there is no cache in the APUs, only a small but very fast local memory is present. The local RAM does not need concurrency and is directly addressable, the programmer will always know what is present because they had to specify the load. Because of this reduced complexity and the smaller size the local RAM will be very fast, much faster than cache. If it can transfer 2 (256bit) words per cycle at the clock speed they have achieved (4.6GHz) they'll be working at 147

Gigabytes per second - and they'll never have a cache miss...

The aggressiveness in the design of the Cell architecture means that it is going to be very, very difficult to produce a comparably performing part. x86 has no hope of getting there, they ultimately need to duplicate the Cell design in order to match it. GPUs will also have a hard time, they are currently at a 10 fold clock speed disadvantage, generate large amounts of heat and the highest performance parts are made in tiny numbers compared to what cell will be made. It will require a complete rethink of the GPUs design in order to get even close to the Cell's clock rate.

The Cell designers have not made their chips out of gallium arsenide or dipped them in a bath of fluorinert so they're not *quite* as aggressive as Seymour Cray, but then again there's always the PlayStation 4...

The Alternative

There is the possibility that some company out there will produce a high power multi core vector processor using a different design philosophy. This could be done and may get close to the Cell's power. It is possible because the Cell has been designed for a high clock rate and this poses some limitations on the design. If an alternative used a lower clock rate, it would allow the use of slower and more importantly smaller transistors. This means the number of vector units included could be increased and more importantly the amount of on-chip memory could be made much greater. These will make up for the higher clock rate and the smaller memory bandwidth necessary would allow slower but lower cost RAM.

This may not be as powerful as the Cell but could get fairly close due to the processors being better fed with all the additional RAM. Power consumption would be lower than Cell and the scalability wouldn't be needed for all markets. There are plenty of companies in the embedded space who stand to lose a lot from the Cell so we may see this sort of design coming from that sector. The companies in the PC CPU and VPU are certainly capable of this sort of design but how it could be made to work in the existing PC architecture is open to question.

The Result

Cell represents the largest threat the PC has ever faced. The PC can't use it's traditional advantage of software because the Cell can run the same software. It can't get an advantage in price or volume as Cell will also be made in huge volumes. Lastly it can't compete on the basis of Cell being proprietary because it's being made by a set of companies and they can sell to anyone. x86 is no less propriety than Cell. It looks like the PC may have finally met it's match.

The effect on Microsoft is more difficult to judge, if Cells take off MS will have difficulty supporting them as it will not allow the same level of control. Because Cells are a distributed architecture you could end up using a Windows machine as a client and having everything else running Linux or some other OS. Multiple machines not running Windows? I don't think that's something Microsoft is going to like.

Then there's also the issue that the main computations may be performed by the Cell with Windows essentially providing an interface. Porting the interface may take time but anything which runs on the Cell's itself is separate and will not need porting to different OSs, software cells are OS agnostic. I can't see that Microsoft are going to like this either.

Nothing is certain and it's not even clear if going up against the PC is something the STI partners are even interested in. But we can be sure Cell and the PC will eventually clash in one way or another.

However even if Cell does take over as the dominant architecture it's going to do so in a process which will take many years or even decades. Then there are areas where Cells may not have any particular advantage over PCs so irrespective of the outcome you can be sure x86 will still be around for a very, very long time.

Cell threatens the current Wintel dominance of the PC industry. The traditional means Intel and Microsoft have used to defend their turf may not prove effective but that's not to say it's a done deal, these companies should never be underestimated. If nothing else, it's certainly going to be an interesting fight.

[Introduction and Index](#)

[Part 1: Inside The Cell](#)

[Part 2: Again Inside The Cell](#)

[Part 3: Cellular Computing](#)

[Part 4: Cell Vs the PC](#)

[Part 5: Conclusion and References](#)

[Part 6: Updates, Clarifications and Missing Bits](#)

© Nicholas Blachford 2005.

Cell Architecture Explained - Part 5: Conclusion and References

Short Overview

The Cell architecture consists of a number of elements:

The Cell Processor

This is a 9 core processor, one of these cores is something similar to a PowerPC G5 and acts as a controller. The remaining 8 cores are called APUs and these are very high performance vector processors. Each APU contains it's own block of high speed RAM and is capable of 32 GigaFlops (32bit). The APUs are independent processors and can act alone or can be set up to process a stream of data with different APUs working on different stages. This ability to act as a "stream processor" gives access to the full processing power of a Cell which is more than 10 times higher than even the fastest desktop processors.

In addition to the raw processing power the Cell includes a high performance multi-channel memory subsystem and a number of high speed interconnects for connecting to other Cells or I/O devices.

Distributing Processing

Cells are specifically designed to work together. While they can be directly connected via the high speed interconnects they can also be connected in other ways or distributed over a network. The Cells are not gaming or computer specific, they can be in anything from PDAs to TVs and all can be used to effectively act as a single system. The infrastructure for this is built into each Cell as they operate on "Software Cells" which contain routing information as well as programs and data.

Parallel programming is usually complex but in this case the OS will look at the resources it has and distribute tasks accordingly, this process does not involve re-programming. If you want more processing power you simply add more Cells, you do not need to replace the existing ones as the new Cells will augment the existing ones.

Overall the Cell architecture is an architecture for distributed, parallel processing using very powerful computational engines developed using a highly aggressive design strategy. These devices shall be produced in vast numbers so they will provide vast processing resources at a low cost.

Conclusion

The first Cell based desktop computer will be the fastest desktop computer in the industry by a very large margin. Even high end multi-core x86s will not get close. Companies who produce microprocessors or DSPs are going to have a very hard time fighting the power a Cell will deliver. We have never seen a leap in performance like this before and I don't expect we'll ever see one again, It'll send shock-waves through the entire industry and we'll see big changes as a result.

The sheer power and low cost of the Cell means it will present a challenge to the venerable PC. The PC has always been able to beat competition by virtue of it's huge software base, but this base is not as strong as it once was. A lot of software now runs on Linux and this is not dependant on x86 processors or Microsoft. Most PCs now provide more power than is necessary and this fact combined with fast JIT emulators means that if necessary the Cell can provide PC compatibility without the PC.

It will not just attack the PC industry but expect it to be widely used in embedded applications where high performance is required. This means it will be made in numbers potentially many times that of x86 CPUs and this will reduce prices further. This will also hurt PC based vendors' desires to enter the home entertainment space as PC based solutions [Entertainment] will be more complex and cost more than Cell based systems.

This is going to prove difficult for the PC as CPU and GPU suppliers will have essentially nothing to fight back with. All they can hope to do is match a Cell's performance but even that is going to be incredibly difficult given the Cell's aggressive Cray-esqe design strategy.

Cell is going to turn the industry upside down, nobody has ever produced such a leap in performance in one go and certainly not at a low price. The CPU producers will be forced to fight back and irrespective of how well the

Cell actually does in the market you can be sure that in a few short years all CPUs will be providing vastly more processing resources than they do today. Even if the Cell was to fail we shall all gain from it's legacy.

Not all companies will react correctly or in time, this will provide opportunities for newer, smaller and smarter companies. Big changes are coming, they may take years but the Cell means a decade from now the technology world is going to look very different.

References and Further Reading

Microprocessor Report Article on the Cell

I'm not the only one to decipher the patents as Microprocessor Report have published an article on the subject (of course just as I finished this..) There is a [short version](#) but you have to pay \$50 to read the full article. I can't comment on the full article as I haven't read it but it's probably a good read.

[Cell Patent]

The original Cell [patent application](#) by Masakazu Suzuoki and Takeshi Yamazaki of Sony Computer Entertainment inc.

This article is based on the interpretation of this document.

The updated [patent](#)

[INQ]

The inquirer ran a story on the Cell patent in early 2003 [here](#).

[Recent Details]

[Press release 1.](#)

[Press release 2.](#)

[Specs]

Cell production specifications [Photo](#).

[3rd party]

Companies can sell cell to their own customers, mentioned in this [Toshiba press release](#).

[SETI]

SETI@home could benefit from the power of cells.

SETI uses a lot of FFTs and the should benefit tremendously from the Cell's design. The use of local memory in the APUs will especially help. [SETI](#)

[Rambus]

Sony and Toshiba licensed Rambus technology for use in the Cell. [Rambus](#)

[GFLOPS]

It's not entirely clear how these are being counted, I assume this is 32 bit floating point operations. Floating point operations can be 16 bit, 32 bit (single precision) or 64 bit (double precision). The Top500 supercomputer list counts double precision GFLOPS so these are not comparable. Assuming the APUs are capable of it a single PE should be capable of 128 GFLOPS (double precision), still over twenty times faster than any "normal" CPU.

[G6] I speculated on a POWER5 based G6 [here](#)

[Vector]

What is [vector processing](#)

[Stream]

ACM Queue on [Streaming Processors](#)

[PC + GPU]

Nvidia with [16 pixel processors](#).

[GPU]

Interesting papaer on GPUs and clustering them to produce very high performance systems [GPU Cluster](#) (Pdf)

[GPGPU]

GPUs can be used for general purpose computations. All these applications will also be useable on a Cell.

[GPGPU](#)

[ShaderTech](#)

[Nvidia]

PS3 to use Nvidia graphics technology. [Nvidia](#)

[Top500]

The Top 500 Supercomputer list, expect Cell to rapidly take over. [Top500.org](#)

[PCShare]

Computer Market share, C64 was biggest in 1984 [Market Share](#)

[Intel+Nvidia]

Intel and Nvidia get cosy. [invidia?](#)

[Project Z]

Intel does a deal with Nvidia, gets a new team from HP and then project Z is discovered. There is talk of massive parallel CPUs but who knows what Intel is planning. [Project Z](#)

[MultiCore]

8 core opterons [Opteropteron](#)

[Transmeta]

Transmeta is considering getting out of chips. [Transmeta de-chips?](#)

[Alien]

PCs can be expensive as well [Aliens](#)

[Future]

PCs under threat in "The Future of Computing". [Future](#)

[DirectX Next]

[NextX](#)

[EE-GPU]

EETimes has a series of articles on using GPU for general purpose work. [EE-GPU](#)

[Cray]Supercomputer designer Seymour Cray also talked about Cells, of the biological kind: [Interview](#)

[Entertainment]

Intel are preparing PC based media [centres](#).

[Calc]

At this point we do not have any performance figures so any figures given are derived from either the figures in

the patent or the 4.6GHz given for the chip itself. All performance figures are thus estimates based on hypothetical maximums.

5 minutes for a SETI unit? This is based on the difference between a 1.33GHz G4 (6 Hours / unit @ 10 GFlops) and a 4 X 250 GFlops Cells, this assumes the SETI client is using AltiVec on the G4 at full speed and could do the same in the PS3.

SETI uses lots and lots of FFTs and these involve lots and lots of Multiply-Add instructions. If we assume the G4 is processing at it's maximum rate and the 4 Cells are doing the same the 4 Cells should be delivering a result in well under 5 minutes.

G4 @ 1.33GHz gives 10.64 Giga Flops (8 instructions (Mul-Add) per cycle using AltiVec).
6 hours = 21,600 seconds
21,600 X 10.64 = 229,824
229,824 / 1,000 (4 Cells @ 250 GFlops)
= 229 seconds
= 3 minutes 49 seconds.

I rounded up to 5 minutes to be conservative.

Other figures (Opteron V's Cell, Top500) should be taken with similar amounts of salt.

[Introduction and Index](#)

[Part 1: Inside The Cell](#)

[Part 2: Again Inside The Cell](#)

[Part 3: Cellular Computing](#)

[Part 4: Cell Vs the PC](#)

[Part 5: Conclusion and References](#)

[Part 6: Updates, Clarifications and Missing Bits](#)

About the Author

[Nicholas Blachford](#) lives in Paris. He is currently loitering with intent on the [Yoper](#) Linux disto, learning French and writing a softsynth on OS X.

© Nicholas Blachford 2005.

Cell Architecture Explained - Updates, Clarifications and Missing Bits

This is to clarify a few points which have appeared in discussions and correct any errors. Where necessary the original has been modified to make the points below clearer or correct them.

Arstechnica ran a piece critiquing this article, given it's content and tone I felt I should write a [rebuttal](#). I have not seen this sort of criticism from anywhere else so I assume this is a combination of a misunderstanding (see "Technical Analysis" below) and someone having a **really** bad day.

Updates

The ISSCC conference is starting in the US on 6th Feb and more information is expected to be made public on the Cell shortly thereafter. This article will be updated once I've read everything.

In the mean time here's some predictions:

- APUs will contain an addressable "local memory".
- APU ISA is highly striped back ultra-RISC, not AltiVec.
- Internal caches for APU-APU transfer (possibly within the APUs).
- POWER5 based PU (AltiVec probably not included).
- Peak performance just below 300 GFLOPs.
- Closer to 100 GFLOPs in normal usage.
- Stream processing may get close to theoretical max.
- FFTs will go like a bat out of hell.
- So will Linpack.
- I am also initially expecting them to be stupidly expensive at launch but prices to drop rapidly.
- I don't predict Apple using them as a main CPU anytime soon, but I expect they'll be used as co-processors in PowerMacs within a year.
- I **do not** predict having to make any significant changes to my article :-)

Clarifications

I'm too Enthusiastic

I am very enthusiastic about this technology and I believe I have good reason to be.

I wrote this because after the recent announcement were made, it was clear that not many people knew how the Cell operated. I had already read the original Sony patent once and had a good idea of how it worked. So, with nothing better to do I decided to dig it out and write an explanation.

Whilst going over it again I was quite amazed at the aggressiveness of the design, I had recently been reading about Cray's designs and this appears to be very heavily influenced by them. This is important because Cray was very successful in constantly making the world's fastest computers, other companies threw hundreds of people and millions of dollars at trying to beat him but they could never do it. If this influence has carried through to the final designs they are going to be monsters. The reported 4.6GHz clock figure indicates to me that this has happened.

Even if the Cell can only attain a fraction of it's potential processing power it's still going to be the fastest CPU on the market by a very wide margin.

Cell Will Demolish The PC On The First Day

One point that has been made to me is that claiming the Cell will demolish the PC on the first day is inviting criticism.

This claim is based on the fact that the similar technology in GPUs is already demolishing general purpose CPUs by hundreds of percent margins. This is not in theory, this is happening in real life applications today. GPUs are used in a variety of ways to accelerate different research applications, they can be used for a lot more than graphics.

The above claim may seem shocking to many PC users but if you are aware of what GPUs are capable of you'll not find it surprising at all. Your graphics chip can already outgun your CPU by a massive margin. You could consider Cell a more general purpose version of the same technology.

Technical Analysis?

A few people have been making comments about the quality of this "Analysis". If you read the first 4 paragraphs at the beginning of part 1 it's pretty clear that this is **not** a technical analysis.

Numbers

All numbers are either direct quotes or are derived from the patent or the 4.6GHz figure. These are the only figures I have to go on so I couldn't use anything else. This should also be clear given what I wrote at the beginning of part 1.

Cell Equivalent To 5 Dual Opteron

Slashdot liked the article so much they posted it twice, unfortunately first time they used the dodgiest quote in the article... This comparison is based on the Cell reaching it's *theoretical* maximum computing power. We will not know **IF** this is possible until the chip becomes available and even then it will need a "perfect" application which can use all the APUs at full power simultaneously.

SETI in 5 Minutes

This is something of a "calculated guess", again based on the theoretical maximum computing power being achieved (in 4 Cells). It makes a lot of assumptions which may be in error so I will not be the least bit surprised if this figure is miles out... I've added the actual calculation to part 5.

I've thought about this a bit more and wondered how to optimise an FFT (which SETI uses extensively) for a Cell. It looks to me like Cell will be a very good architecture for running FFT based applications.

The Cell Compiler Will Magically Make The Code Parallel

This is not true and I didn't say this. You still have to break up problems into software Cells. I still cannot figure out why people think I said this.

IBM processor

This article was based on the original Sony patent, there have been other patents but I have not looked at them in detail, I understand there may be caching mechanisms to allow transfers between the different processing units. The CPUs have been designed by STI (Sony / Toshiba / IBM) the first CPUs should be available from IBM this year.

Maximum Memory Size

While the patent specifies 8 bank controllers with 8 MB each it does not specify the maximum memory of an individual Cell. So 64 MB is an "ideal" rather than an absolute. I suspect the Cells in the PS3 may be hardwired for 64 MB but not others.

Memory Protection And Paging

The APUs will work on specific chunks of RAM set aside specifically. They can either access it or not, nothing more. There is no paging system for the APUs as there is no need for it.

The PU will likely have a normal memory protection system with paging as it'll be dealing with more complex operations (i.e. running an OS).

Replacing PCs / Emulation

A single Cell will not emulate a top end PC at the same level of performance. What it will be able to emulate is a low end PC, this is useful because many heavy processing tasks can be off-loaded to the APUs so you wont need a high end PC. This will not happen for every possible task but will be a pretty common scenario.

The emulation itself may not work on the APUs unless the instructions are vectorisable or are already in vector form (MMX, SSE).

You Program These Things In Assembly?

No, you don't program virtual machines directly either, the compiler handles that for you. This will work the same way, but for real speed you'll probably want assembler level control.

IBM already has Cell compilers in beta.

This is just hype like the PS2 chip

From reading the patent on the PS2 chip/s, the vector processing engines in the PS2 are powerful but also somewhat specialised. The Cell contains general purpose vector units, that is anything which can be vectorised will run. According to one reader the PS2 processors could have been useful for general purpose vector processing but Sony never made it available. The difference this time is the Cell will not just be used in PS3s.

Untested, Radical New Architecture

Actually it's not. look up Cray XMP or YMP and you'll see another parallel vector processor. The difference is they've put this on a single chip with a massively higher clock rate. What is new is the way the programs (apulets) are distributed and the fact we've never seen anything as powerful as this on the desktop.

Missing Bits

Cell V's Java

I explained the differences in part 2 but I thought of this afterwards (of course):

What would happen if you were to combine them?

The result would be a application which would work on (almost) every Platform / OS and take full advantage of the Cells.

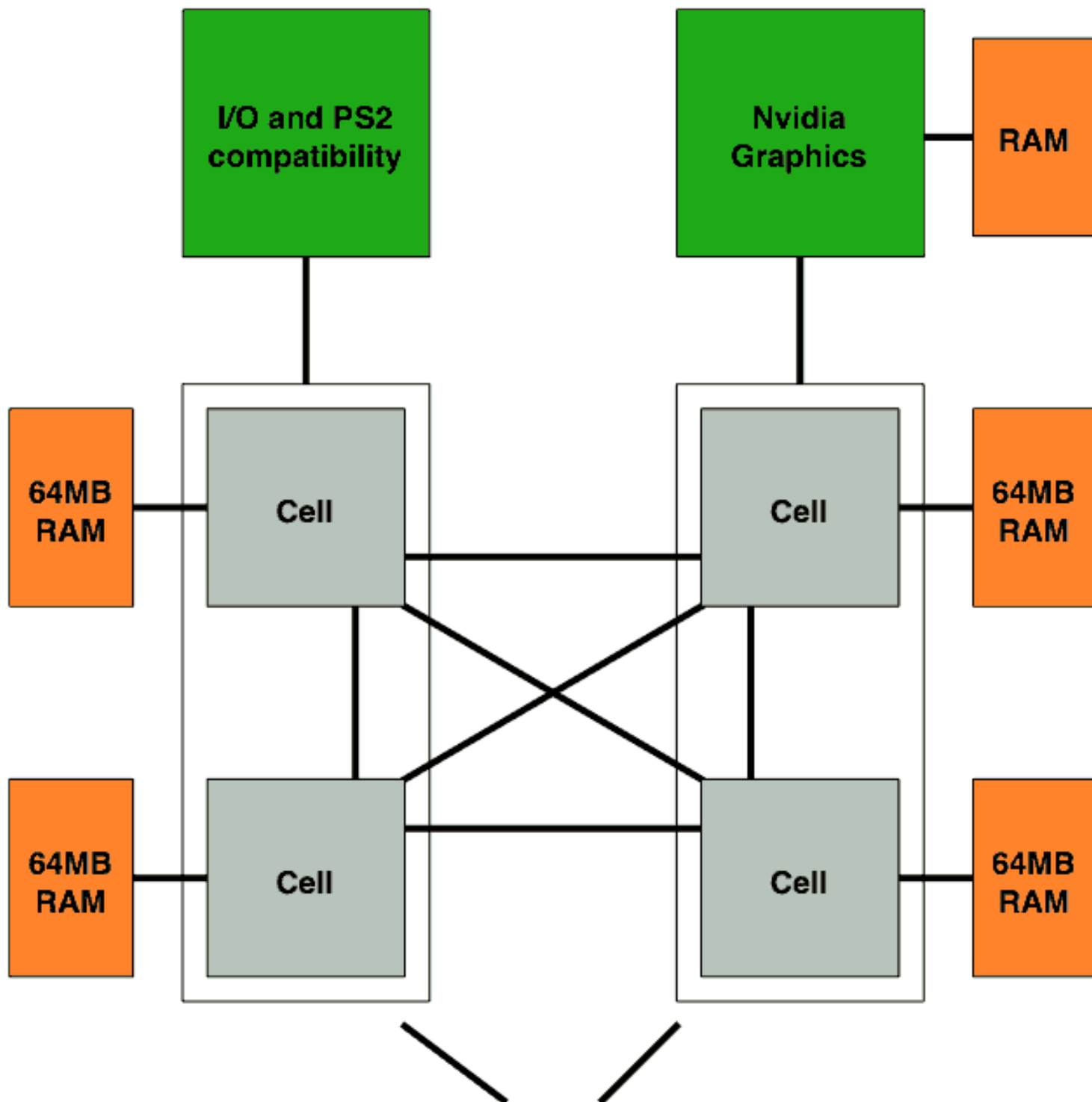
Going further, this could lead to applications in which the GUI and logic is written in a scripting language and the heavy processing in a more complex language. Such applications will be quick and easy to develop and automatically cross platform.

Dissapointed!

This thing gets views by lots and lots of 1337 haX0Rz and nobody notices the hidden diagram! I was worried about bandwidth so I removed this from the article but it was still sitting in the same dir for anyone to pick up. In the event surprisingly little bandwidth was used so, here it is:

PlayStation 3 Architecture ?

This is a guess at what the PS 3 architecture will look like.



Chips have 2 Cells each

© Nicholas Blachford 2005

[Introduction and Index](#)

[Part 1: Inside The Cell](#)

[Part 2: Again Inside The Cell](#)

[Part 3: Cellular Computing](#)

[Part 4: Cell Vs the PC](#)

[Part 5: Conclusion and References](#)

Updated: 25/01/05 (1)

© Nicholas Blachford 2005.