Cell Processor

By: Chandana Kalagara
CSC 561
Date: 4/11/2007

Abstract:-

A group of three corporate (Toshiba, Sony and IBM) companies tries another way to come into the market with a new idea, the cell design. The cell processor is a new try to leverage the increasing amount of transistors per die in an efficient way. The new processor is targeted at the game console and consumer electronics market to enhance the quality of these devices. This will lead to a wide spreading, and if everybody has two or more cell processors in TV, game console or PDA, the interesting question comes up: what can I do with these processors? This paper gives a short overview of the architecture and several programming ideas which help to exploit the whole processing power of the cell processor.

Introduction:-

The Cell concept was originally thought up by Sony Computer Entertainment inc. of Japan, for the PlayStation 3. The architecture as it exists today was the work of three companies: Sony, Toshiba and IBM. Sony and Toshiba previously co operated on the PlayStation 2 but this time the plan was more ambitious and went beyond chips for video game consoles. The aim was to build a new general purpose processor for a computer. With that in mind IBM was brought in as their expertise is in computer design.

The development of the processor began in 2000, when the three companies Sony, Toshiba and IBM funded a research group to develop a new processor to fit the demands of multimedia applications. Sony had experience in processor design and programming due to the Play Station 1&2. Their vision was to make the Play Station 3 (PS3) around 100 times faster than its predecessor the PS2. Toshiba offered experiences in the field of development and high volume manufacturing and IBM as the" traditional" processor designer and manufacturer (especially the 90nm SOI technology with copper wiring). IBM also brought its chip design expertise and in this case used a very aggressive approach by producing a fully custom design - the chip's circuitry was designed by hand instead of with automated tools, very few other companies use this approach. IBM also has the industry's leading silicon process which will be used in the manufacturing. Sony and Toshiba bring mass manufacturing capabilities and knowledge. They started their research project in a design center in Austin,
Texas in 2001 with an investment of more than 400 Million dollars. All three companies had future plans for their use of the new technology, Sony wanted to incorporate the Cell into the PS3, Toshiba into HDTV TVs and IBM in server or workstation systems. Their target was to build a high clock-rate and high throughput multi purpose processor.

Description:-

Cell is architecture for high performance distributed computing. This architecture is not fixed, if we have a computer, PS3 and HDTV which have Cell processors they can co-operate on problems. They have been talking about this sort of thing for years of course but the Cell is actually designed to do it. The Cell architecture was patented by Ken Kuturagi (Sony) in 1999. He differentiated between software and hardware cells. A software cell is a program with the associated data, and the hardware cell is an execution unit with the capability to execute a software cell. The Cell’s hardware has been specifically designed to provide sufficient data to the computational elements to enable such performance. According to IBM the Cell performs ten times faster than existing CPUs on many applications. This may sound ludicrous but GPUs (Graphical Processors Units) already deliver similar or even higher sustained performance in many non-graphical applications. Cell processor is a crowded little chip, with a whopping 234 million transistors (compare to AMD64’s 114 million transistors). The potential processing power of Cell blows away existing processors, even supercomputers. One Cell working alone has the potential of reaching 256 GFLOPS (gigaflops, processing 256 billion floating point operations per second). GFLOPS are benchmarked with a program called Linpack and are mostly useful for comparing supercomputers. To compare, our home PC would be extremely lucky to reach 6 GFLOPS, unless we count our graphics card. The most outstanding difference between Cell and current technologies is that a normal CPU in a computer or game console has, of course, one processing unit that churns through one thread of data. Cell can run through many threads of data at a time through the eight data processing units within a single chip.

Main Components in cell processor architecture:-

An individual hardware Cell is made up of a number of elements:

- One Power Processor Element (PPE).
- Eight Synergistic Processor Elements (SPEs).
- Element Interconnect Bus (EIB).
- Direct Memory Access Controller (MIV).
- Two Rambus XDR memory controllers.(just like RAM)
- Rambus FlexIO (Input / Output) interface.
Cell Processor Architecture

Specifications known so far:-

- Capable of running at speeds beyond 4 GHz.
- Memory bandwidth: 25.6 GBytes per second.
- I/O bandwidth: 76.8 GBytes per second.
- 256 GFLOPS (Single precision at 4 GHz).
- 256 GOPS (Integer at 4 GHz).
- 25 GFLOPS (Double precision at 4 GHz).
- 235 square mm.
- 235 million transistors.

Power processing element (PPE):-

The Power Processing Element (PPE) offers the normal PowerPC (PPC) ISA. It is a dual threaded 64 bit power processor which includes VMX. Its architecture is very simple to guarantee high clock rates. The PPE has a cache to hold tasks to be processed on fast localized memory, an L1 of 32 KB and an L2 of 512 KB. Current AMD64 chips have this beaten already, with an L1 of 128 KB and L2 of up to 1 MB. The AMD64 definitely has larger on-processor storage, but this is not yet decided. The AMD64 cache holds data for the one processor to crunch. The
Cell’s PPE cache is more of a temporary holding area, where the PPE parcels the data and sends it off to a Synergistic Processing Element (SPE). That is to say, it is NOT based on the existing 970/G5 or POWER processors. It is a completely different architecture so clock speed comparisons are completely meaningless. Most modern microprocessors devote a large amount of silicon to executing as many instructions as possible at once by executing them "out-of-order" (OOO). This type of design is widely used but it requiring hefty amounts of additional circuitry and consumes large amounts of power. With the PPE, IBM have not done this and have instead gone with a much simpler design which uses considerably less power than other PowerPC devices - even at higher clock rates.

This design will however have the downside of potentially having rather erratic performance on branch laden applications. Such a simple CPU needs the compiler to do a lot of the scheduling work that hardware usually does so a good compiler will be essential. Some of the technology in the PPE has been derived from IBM's high end POWER series of CPUs, Like POWER5 the PPE has the ability to run 2 threads simultaneously. When one thread is stalled and is waiting for data the second thread can issue instructions keeping the instruction units busy. IBM's hyper visor technology [Hyper] is also used allowing the Cell to run multiple operating systems simultaneously. According to IBM the Cell can run a normal OS alongside a real time OS with both functioning correctly.

Another interesting point about the PPE is that it includes support for the VMX vector instructions, (also known as "AltiVec" or "Velocity Engine"). VMX can speed up anything from financial calculations to operating system functions though it (or its PC equivalents) doesn’t appear to be that widely used currently. One company which does use VMX extensively is Apple who uses it to accelerate functions in OS X; it would not have been a huge job for Apple to utilize the PPE in the Cell. The PPE is an interesting processor and it looks likely that similar cores will turn up in systems other than the Cell. The CPU cores used in the XBox360 while different appear to be derived from the same original design.

**Synergistic Processor Elements (SPE):**

Even though the PPE is a powerful processor in its own right, the eight Synergistic Processing Elements (SPEs) are the real workers. The SPEs don’t have any cache, but they do have four 64 KB arrays of private memory, or Load Store (LS) units. This gives them a total of 256 KB of private memory. Current processors automate the use of the memory for tasks like data fetch and branch prediction, but this adds complexity and cost to hardware. The SPEs’ memory is not cache, but instead operates as flexible storage for the little
processor. In Cell, programs must manage how this memory is used, and this potentially makes these functions far more efficient and specialized for the programs.

Cell SPE Architecture

Each SPE is an independent vector CPU capable of 32 GFLOPs or 32 GOPs (32 bit @ 4GHz.)

The SPE is essentially a full blown vector CPU with own RAM. Its ISA is not compatible to VMX and has a fixed length of 32 Bit. Current SPEs have about 21 Million Transistors where two third of them are dedicated to the SRAM (memory). The processor has no branch prediction or scheduling logic, and relies on the
programmer/compiler to find parallelism in the code. As the PPE, it uses two
independent pipelines and issues two instructions per cycle, one SIMD
computation operation and one memory access operation. All instructions are
processed strictly in-order and each instruction works with 128 Bit compound
data items. 4 single precision floating point units and 4 integer units offer up to
32GOp/s each. The single precision floating point units are not IEEE754
compliant in terms of rounding and special values. The single precision units can
also be used to compute double precision floating point numbers which are
compliant to the IEEE754 standard. But their computation is rather slow
(34GFlops). The schematic architecture of a single SPE is shown in the above
figure. The memory layout of the SPE is also quite special, each SPE has its
own 256kb RAM which is called Local Storage (LS). This SRAM storage can be
accessed extremely fast in 128 bit lines. Additionally, each SPE has a large
register file of 128 bit registers which store all available data types. There is no
cache, virtual memory support or coherency for the Local Storage and the data
can be moved with DMA from/to main memory via the EIB. The Memory Flow
Controller (MFC) acts like a MMU in this context and provides virtual memory
translations for main memory access.

It’s expected that Cell will be extremely powerful for stream
processing. In a demo, Toshiba presented of a Cell processor decoding 48
streams of video, only 6 of the SPEs were used for data decoding. Another was
used for scaling the screen and the last can be used for other tasks. The SPEs
can function in unison, like decoding streams together, or they can be dedicated
to completely separate jobs. An SPE is a self contained vector processor which
acts as an independent processor. They each contain 128 x 128 bit registers,
there are also 4 (single precision) floating point units capable of 32 Gigaflops and
4 Integer units capable of 32 GOPS (Billions of integer Operations per Second)
at 4GHz. The SPEs also include a small 256 Kilobyte local store instead of a
cache. According to IBM a single SPE (which is just 15 square millimeters and
consumes less than 5 Watts at 4GHz) can perform as well as a top end (single
core) desktop CPU given the right task.

This is counting Multiply-Adds which count as 2 instructions, hence 4GHz x 4 x 2
= 32 GFLOPS.

32 X 8 SPEs = 256 GFLOPS

The SPEs are vector (or SIMD) [Vector] processors. That is, they
do multiple operations simultaneously with a single instruction. Vector computing
has been used in supercomputers since the 1970s (the Cray 1 was one of the
first to use the technique) and modern CPUs have media accelerators (e.g.
MMX, SSE, VMX / AltiVec) which work on the same principle. Each SPE is
capable of 4 X 32 bit operations per cycle (8 if you count multiply-adds). In order
to take full advantage of the SPEs, the programs running will need to be
"vectorised", this can be done in many application areas such as video, audio,
3D graphics, scientific calculations and can be used at least partially in many other areas.

**Element Interconnect Bus (EIB):**

The EIB is the central communication channel inside a Cell processor; it consists of four 128 bit wide concentric rings. The ring uses buffered point to point communication to transfer the data and is therewith scalable. It can move 96 bytes per cycle and is optimized for 1024 bit data blocks. Additional nodes (e.g. SPEs) can be added easily and increase only the maximal latency of the ring. Each device has a hardware guaranteed bandwidth of \(1/\text{num Devices}\) to enhance the suitability for real time computing.

The EIB carries data traveling from the PPE and L2 cache to the SPEs and back again. It also connects this data to the Memory Interface Controller (MIC) and the FlexIO Front Side Bus (FlexIO FSB). Through the MIC and FlexIO FSB, the EIB moves data from all those inner processors to parts of the computer outside the CPU, including system memory. It’s notable that the MIC is located on-chip. The AMD64 also offers this, but the AMD64’s reasonable memory bandwidth of 6.4 GB/s doesn’t even compare to Cell’s. The MIC on a Cell chip is designed to run at 25.6 GB/s (dual 12.8 GB/s channels). That certainly shouldn’t be a bottleneck

**I/O Interconnect – FlexIO:**

The I/O Interconnect connects the Cell processor (the EIB) to the external world, e.g. other cell processors). It offers 12 unidirectional byte-lanes which are 96 wires. Each lane may transport up to 6.4GB/s, which make 76.8 GB accumulated bandwidth. 7 lanes are outgoing (44.8 GB/s) and 5 lanes incoming (32 GB/s). There are caches coherent (CPU interconnects) and non coherent links (device interconnect) and two cell processors can be connected glueless.

The FlexIO FSB, like any front side bus, is what the processor uses to talk to the other parts of the computer. The FlexIO, however, will have two types of data that it transmits. One, called non-coherent which is used for talking to sound cards, video cards, etc. This is what FSBs have been traditionally for. The other kind will be coherent data traffic, used to send and receive data from other Cell processors. Since the Cell and its software are designed to operate by dividing packets and processing cooperatively, linking up a bunch of Cell processors will make a larger and more efficient cooperative processing system. To be able to allow these speedy processors to communicate with each other and the rest of the machine, the FlexIO FSB has been pretty fast. Between both sets of data traffic, the bandwidth is 76.8 GB/s. To compare to our x86 counterpart, the AMD64 can only handle up to 8 GB/s bandwidth.
The Memory Interface Controller:-

The MIC connects the EIB to the main DRAM memory, which is in this case Rambus XDR memory which offers a bandwidth of 25.2 GB/s. The memory is ECC protected which shows that the cell will be used for more than game consoles and consumer electronics (this could also be for better EMC protection). The MIC offers virtual memory translation to the PPE and the SPEs. The memory itself is not cached, only the PPE has an own cache hierarchy. It's notable that the MIC is located on-chip. The AMD64 also offers this, but the AMD64’s reasonable memory bandwidth of 6.4 GB/s doesn’t even compare to Cell’s. The MIC on a Cell chip is designed to run at 25.6 GB/s (dual 12.8 GB/s channels). That certainly shouldn't be a bottleneck.

Cell Programming:-

The programming of the cell will be as special as the architecture. The big advantage is that there is no abstraction layer between an external ISA and the internal core (cmp. x86). But the strict RISC design moves the effort to generate optimal code up, to the programmer or compiler. And the general problems of parallel or streaming application development stay the same as for multi-core or multi processor machines. The SPEs are programmed in a direct manner, as own autonomous processors with their 256kB Local Storage and 128 Registers.

An Assembly language specification is available from IBM but higher level languages such as C/C++ are also supported for application development. The task distribution and allocation of SPEs is fully done in software. The operating system could use them as a shared resource and virtualize them for many tasks (each task sees their own SPEs, in the whole more than available). The PPE is programmed like a standard PPC970 and Linux is running as is (without SPE support, but a patch is available from IBM). The SPEs can be used in different scenarios. A job queue can be used to processes a fixed amount of independent jobs as fast as possible, the SPEs could also be used in a self multitasking manner, as if the cell were a multi-core or multi CPU system. Stream processing (Pipelining) is also supported and especially very reasonable for media data (e.g. HDTV). The Local Storage could be used as a cache, but has to be managed by the software for this task. Additionally, MPI support is thinkable, where each SPE is a single node. All these different programming models are just ideas, the future will show which models will be used on the new Cell processors.
IBM is trying to accelerate Cell’s development by lifting limitations and revealing its structure; in this way, talented programmers can better understand the processor and program for it more easily than otherwise. This might make it attractive for some companies that want great performance that the Cell delivers without extreme costs. IBM might also get some free hardware and software development from these opportunistic companies and from some helpful individuals. Open hardware could bring flexibility and usefulness to Cell systems that it couldn’t have reached before.

**Play station 2 versus Play station 3:-**

The below are the different specifications used for PS2 and PS3, and we can see how much PS3 is improved by using the cell processor.

<table>
<thead>
<tr>
<th></th>
<th>Sony Emotion Engine</th>
<th>Cell Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU Core ISA</strong></td>
<td>MIP64</td>
<td>64-bit Power Architecture</td>
</tr>
<tr>
<td><strong>Core Issue Rate</strong></td>
<td>Dual</td>
<td>Dual</td>
</tr>
<tr>
<td><strong>Core Frequency</strong></td>
<td>300MHz</td>
<td>~4GHz (est.)</td>
</tr>
<tr>
<td><strong>Core Pipeline</strong></td>
<td>6 stages</td>
<td>21 stages</td>
</tr>
<tr>
<td><strong>Core L1 Cache</strong></td>
<td>16KB I-Cache +</td>
<td>32KB I-Cache +</td>
</tr>
<tr>
<td></td>
<td>8KB D-Cache</td>
<td>32KB D-Cache</td>
</tr>
<tr>
<td><strong>Core Additional Memory</strong></td>
<td>16KB scratch</td>
<td>512KB L2</td>
</tr>
<tr>
<td><strong>Vector Units</strong></td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td><strong>Vector Registers (#, width)</strong></td>
<td>32, 128-bit + 16, 16-bit</td>
<td>128, 128-bit</td>
</tr>
<tr>
<td><strong>Vector Local Memory</strong></td>
<td>4K/16KB I-Cache +</td>
<td>256KB unified</td>
</tr>
<tr>
<td></td>
<td>4K/16KB D-Cache</td>
<td></td>
</tr>
<tr>
<td><strong>Memory Bandwidth</strong></td>
<td>3.2GB/s peak</td>
<td>25.6GB/s peak (est.)</td>
</tr>
<tr>
<td><strong>Total Chip Peak FLOPS</strong></td>
<td>6.2GFLOPS</td>
<td>256GFLOPS</td>
</tr>
<tr>
<td><strong>Transistor Count</strong></td>
<td>10.5 million</td>
<td>235 million</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>15W @ 1.8V</td>
<td>~80W (est.)</td>
</tr>
<tr>
<td><strong>Die Size</strong></td>
<td>240mm²</td>
<td>235mm²</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>250nm, 4LM</td>
<td>90nm, 8LM + LI</td>
</tr>
</tbody>
</table>
In Future:

The Sony-Toshiba-IBM trio expects that Cell will be able to be used in mobile devices, such as laptops and even cell phones. As of now, that’s not possible because the power consumption is too high. Cells clocked at 4 GHz reportedly suck down 80 Watts, not that bad for PC hardware but still too much to plant in a phone or PDA.

The companies have announced no plans to put the processor in mainstream consumer or business computers, though the idea wouldn’t seem beyond Cell’s scope. Once Cells are developed to be fully cooperative, the possibilities for the hardware are limitless. When buying a new computer, we won’t be replacing our old one; we’ll be supplementing it. Instead of computers being completely separate (and often useless after you replace one), old computers could still be very useful for donating extra processing cycles to the new system. Actually, with Cells going into TVs and game consoles, we could ideally network all our Cell-powered electronics and they could share and balance the workload between them.

So we’re getting low FPS in some cutting edge game, say Doom 6 or Half Life 14; give a little boost by connecting your computer to your home entertainment system, microwave, and garage door opener. It can send some extra chunks of data out to these other devices for them to crunch. It will be like having your own makeshift computer array at home. Some say that Cell is only a media device that it is useful in certain applications and is not useful for general computing. This really has absolutely no grounding yet, and it remains to be seen. The way programs are handled on Cell will not be the same as on current processors. They process "cells" of data and not a linear thread. If a stripped down Cell can render games smoothly on a PS3, and a full fledged Cell can clock up to 5.6 GHz already, the PC may have to get ready for a little competition.

Of course, PCs have already crushed a lot hardware that outperformed them in the past. Even if Cell computers are produced and are far superior, the market place may not go their way. The key to Cell’s success is probably marketing its interconnectivity and also starting outside the general consumer market, like in Mercury’s products.

AMD and Intel must be ready for this new technology. The chip makers just released 64-bit CPUs and dual-core CPUs (to a market that isn’t quite ready), which seem to be born from a similar vision. PC manufacturers are already beginning to use the multiprocessor principles to accelerate those processes that will benefit from it. Cell’s structure seems to be pointing to the future of CPU architecture, even in PCs.
However, software developers don’t seem ready to write the new software required for dual-cores, whether it’s because they aren’t comfortable with it or they don’t see enough advantage yet. Needing a new software base to properly take advantage of dual-cores certainly doesn’t put the traditional PC hardware at any advantage. It also doesn’t help that those dual-cores only perform at a fraction of the capacity of a Cell. Though a poor excuse so far, they may be just the beginning. Before the Cell is practical outside of TVs and game consoles, they may look quite a bit more functional.

Conclusions:-

The Cell processor has a new design and consists of a single Power CPU and 8 additional vector processors. The Single Instruction, Multiple Data approach is used together with a strict RISC instruction set. The extremely fast I/O and memory subsystems allow high bandwidth communication with memory and other processors or devices. The hardware offers also real time capabilities which can be used in combination with HDTV stream programming.

Cell is a revolution, a completely new microprocessor architecture which, while it may take some time to get used to, promises a vast performance boost over today’s systems. GPUs can already run 10 times faster than desktop CPUs, Cell will not only bring similar performance but will do so for more applications and it’ll be easier to program. Being produced in large volumes also means the Cell will be cheap. They will likely see wide spread not just in living rooms but in the realm of industry and science as well.

Some have suggested that STI (Sony, Toshiba and IBM) should have gone for a more conventional design such as three PowerPC 970s on a single chip. Such a design would not have addressed the power issues and would, as a result have to of been driven at a relatively low clock rate. Instead, by using simpler designs which use vectors the Cell designers have managed to fit 9 cores on a single chip at a higher clock speed, the potential performance is consequently considerably higher.

The Cell is a new architecture and will seem strange and alien to many used to rather more conventional desktop designs. In order to utilize it properly programmers will have face new problems and devise new ways of solving them. It remains to be seen how much of the Cell’s potential can be achieved and how difficult it is to extract it, but it’s clear that STI are trying to make this as painless as possible.

Many people do not like change, to them Cell represents a threat. For others it represents an opportunity. Let’s see how many take the opportunity, and what other opportunities the other CPU vendors come up with in response. Altogether, the new architecture could be the standard of tomorrow and is (hopefully) integrated into the PS3 and TVs in 2006.
References:-


http://www.realworldtech.com/page.cfm?ArticleID=RW021005084318

http://www.realworldtech.com/page.cfm?ArticleID=RW022805234129

http://www.blachford.info/computer/Cell/Cell0_v2.html