Seminar Report
On

AMD'S BULLDOZER ARCHITECTURE

Submitted by

Mr. SAYYAN N SHAIKH
R.No: 50 USN: 2VD07CS041

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KARNATAK LAW SOCIETY'S
VISHWANATHRAO DESHPANDE RURAL INSTITUTE OF TECHNOLOGY, HALIYAL-581329

2010-11

KARNATAK LAW SOCIETY'S
VISHWANATHRAO DESHPANDE RURAL INSTITUTE OF TECHNOLOGY, HALIYAL-581329

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the seminar work entitled **"AMD'S BULLDOZER ARCHITECTURE"** has been carried out and presented by **SAYYAN N SHAIKH (2VD07CS041)**, in partial fulfillment of the degree of Bachelor of Engineering, in Computer Science and Engineering of Visvesvaraya Technological University Belgaum, during the academic year 2010-11.The seminar report has been approved as it satisfies the academic requirements in respect of seminar.

COORDINATOR(S) **HOD**

1)

2)

ACKNOWLEDGEMENT

The magnitude of this seminar demanded the co-operation, guidance and assistance from a number of people and I have been fortunate to have this, and I take this opportunity to thank all those who have helped me in this seminar.

I express a deep sense of gratitude to **Prof. Saleem Hebbal** and **Prof. Damodar Hotkar** for their constant timely advice, valuable suggestions and help given from time to time.

I also express a deep sense of gratitude to **Prof. Poornima Raikar** for her valuable suggestions and help given.

I take this opportunity to thank **Prof. A.V. Kolaki** HOD of Computer Science and Engineering Dept, VDRIT Haliyal providing the inspiration for taking this seminar.

I will be failing in my duty if I don't thank our principal **Dr. G.R. Udupi** for providing healthy environment in the college that helped in concentrating on the task.

Then I would also thank all teaching and non teaching staff of the department who directly or indirectly contributed in accomplishing this task.

Last but not the least I express my sincere thanks to all persons who have directly or indirectly assisted my endeavour.

Sayyan N Shaikh

CONTENTS

LIST OF FIGURES

ABSTRACT

Bulldozer is the codename AMD has given to one of the next-generation CPU cores after the K10 microarchitecture for the company's M-SPACE design methodology, with the core specifically aimed at 10 watt to 125 watt TDP computing products.

Basically, it is a monolithic dual core building block that supports two threads of execution and is intended for deployment in everything from mainstream clients (including desktops and notebooks) to servers.

The bulldozer architecture has two dedicated integer cores. Each of these consist of 2 Arithmetic Logic Unit 2 AGU which together can execute 4 independent arithmetic or memory operations per clock cycle per core. Core module of AMD bulldozer shares portions of a traditional core—including the instruction fetch, decode, and floating-point units and L2 cache—between two otherwise-complete processor cores. The integer core has duplicating integer schedulers. The execution pipeline offers dedicated hardware significantly increasing performance in multi-threaded integer applications.

CHAPTER -1

INTRODUCTION

AMD is unveiling today the new processor architecture that will be used in their new CPUs starting in 2011. Codenamed Bulldozer, this architecture is completely different from the current AMD64 architecture that AMD has been using since the introduction of the very first Athlon 64 CPU back in 2003.

Bulldozer will be the first major redesign of AMD's processor architecture since 2003, when the firm launched its Athlon 64/ Opteron (K8) processors, and will feature two 128-bit FMA-capable FPUswhich can be combined into one 256-bit FPU. This design is accompanied by two integer cores each with 4 pipelines (the fetch/decode stage is shared). Bulldozer will also introduce shared L2 cache in the new architecture. AMD calls this design a "**Bulldozer module**". A 16-core processor design would feature eight of these modules, but the operating system will recognize each module as two physical cores.

The module, described as two cores, can be compared to a single Intel core with HyperThreading. The difference between the two approaches is that Bulldozer provides dedicated schedulers and integer units for each thread, whereas in Intel's core each thread can access all available resources, except for the individual thread state information.

The Bulldozer architecture will inherit some features introduced with the AMD64 architecture, such as the integrated memory controller and the use of the Hyper Transport bus for communication between the CPU and the chipset.

Bulldozer is the codename for the architecture, not for a specific processor. AMD has proposed the first desktop CPUs based on the Bulldozer architecture will require a new CPU socket, called AM3+, which will also be compatible with current socket AM3 processors. Socket AM3+ CPUs, however, won't be compatible with socket AM3 motherboards.

The Bulldozer architecture will have an equivalent of the Intel Turbo Boost technology, allowing the CPU to overclock itself if you are running CPU-intensive programs and if the thermal dissipation is still within specs.

Bulldozer is the first x86 design to share substantial hardware between multiple cores, in some cases blurring the traditional notion of a core. Current x86 designs share the last level cache, power management and external interfaces such as the memory controllers, coherent interconnects and other I/O between 2-8 cores. Bulldozer is a hierarchical design with sharing at nearly every level. Each module or compute unit (i.e. a pair of cores) share an L1I cache, floating point unit (FPU) and L2 cache, saving area and power to pack in more cores and attain higher throughput - albeit with a slight cost in terms of per-core performance. All modules in a chip share the L3 cache, Hypertransport links and other system components.

While sharing the front-end and FPU seems radical compared to today's x86 designs, it is a natural evolution in the multi-core era. The front-end has to deal with a lot of the complexity of the x86 instruction set, which leads to power hungry decoders and large structures like microcode. Floating point hardware also consumes a great deal of area and power and is rarely utilized over 40% - so sharing between two cores is an excellent way to gain back area and power with a minor performance loss. In many ways, AMD is positioning this high degree of sharing as an alternative to multi-threading, which is used by almost every other high performance CPU and there is some truth to this claim.

Bulldozer itself is somewhat of a contradiction – it is a substantial departure from the previous generation Istanbul, yet in most parts of the design, it is also clearly descended from AMD's previous work. Before going through the internals of the Bulldozer architecture, let's first see the instruction sets supported by this new architecture.

CHAPTER -2

INSTRUCTION SETS

The Bulldozer architecture, besides being compatible with the standard x86 instructions, will support the following additional instruction sets:

- ✓ SSE4.1 and SSE4.2
- ✓ AVX (Advanced Vector Extensions) with two additional subsets, called XOP and FMA4
- ✓ AES (Advanced Encryption Standard)
- ✓ LWP (Light Weight Profiling)

2.1. SSE4.1 and SSE4.2

Finally AMD CPUs will support SSE4 instructions. Currently AMD CPUs don't support these instruction sets, which increase speed in multimedia applications (images and video processing) that support it. Current AMD CPUs support a proprietary instruction set called SSE4a, which isn't the same thing as SSE4.

2.2. AVX (Advanced Vector Extensions)

A while ago, AMD proposed an SSE5 instruction set. Because Intel decided to create its own implementation of what would be the SSE5 instructions, called AVX (Advanced Vector Extensions), AMD added this instruction set to the Bulldozer architecture.

The AVX instructions will also be supported by forthcoming CPUs from Intel based in their Sandy Bridge architecture, and use the same SIMD (Single Instruction, Multiple Data) concept introduced with the MMX instruction set and used by the SSE (Streaming SIMD Extensions) instructions. This concept consists in using a single big register to store several small-sized data and then process all data with a single instruction, speeding up processing.

The AVX instruction set adds 12 new instructions and increases the size of the XMM registers from 128 bits to 256 bits.

In the Bulldozer architecture, AMD has decided to add some of the instructions they had originally proposed for the SSE5 instruction set. Therefore, the AVX implementation in the Bulldozer architecture is more complete than Intel's. These additional instructions are called the XOP and FMA4 instructions, in their Bulldozer presentations AMD is announcing the AVX instruction set as "also" having the FMAC (Fused Multiply Accumulate) subset, but this subset of instructions is actually part of the XOP instructions. The "AMD 4-operand form" being announced in the AMD presentations is simply the new format used by the XOP instructions and mentioning this is also completely redundant.

2.3. AES (Advanced Encryption Standard)

This instruction set is already being used in the new Intel CPUs based on the "**Westmere**" architecture and newer (except Core i3), and consists of six new instructions to deal specifically with encryption. Intel calls this instruction set AES-NI.

2.4. LWP (Light Weight Profiling)

The LWP instructions allow programs to easily monitor software performance, which will help developers to fine-tune programs for best performance, for example. This additional instruction set has six new instructions.

CHAPTER -3

THE CPU BUILDING BLOCKS

AMD decided to take a completely different approach to build the new Bulldozer architecture. The Bulldozer architecture has “dual-core” module that shares some resources (the front-end engine, the floating-point unit, and the L2 memory cache, see Figure 1) and, therefore, are not completely independent from each other.

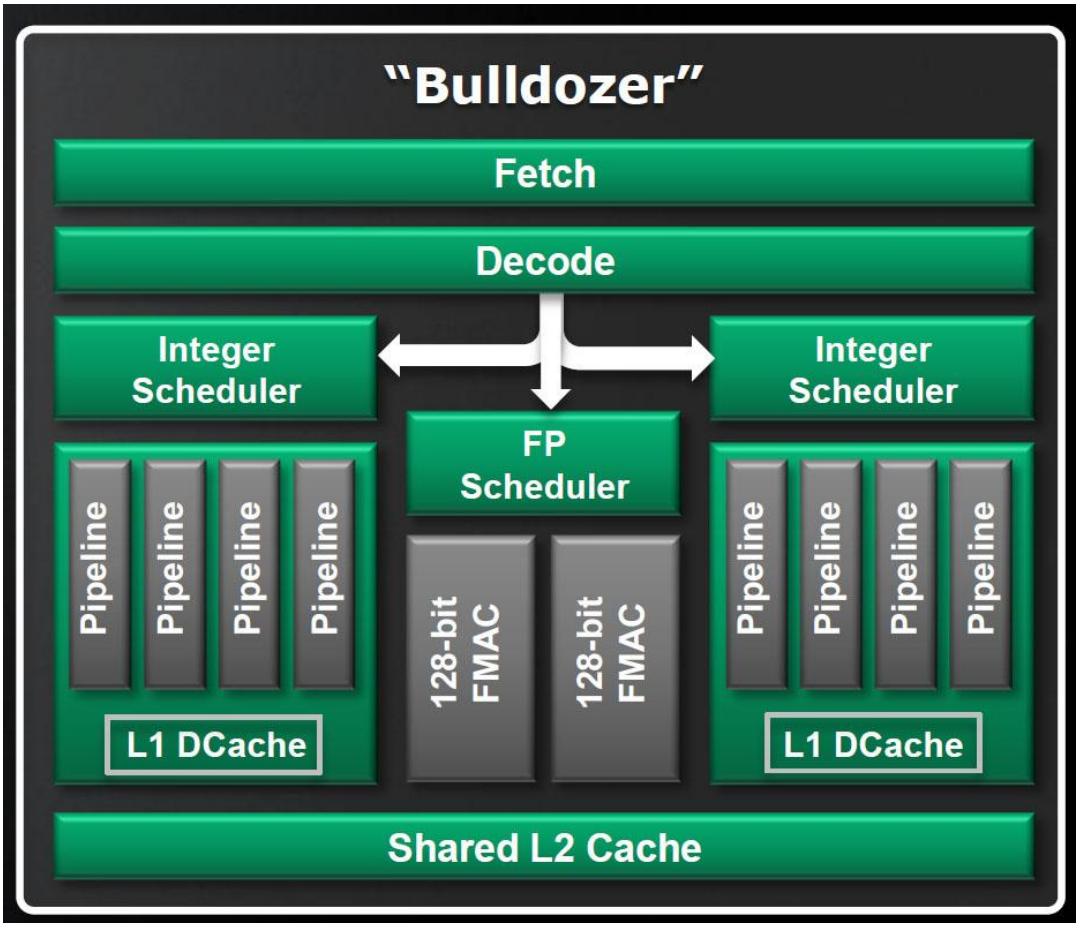

Figure 3.1 Bulldozer building block

According to AMD this changes has been done in order to optimize the CPU and, at the same time, cut costs. The optimization comes from the fact that on a typical multi-core CPU several units inside the CPU remain idle, and these units could be combined in the Bulldozer architecture. And since the CPU will have less units, it can be smaller, which reduces the amount of material necessary to build the CPU, reducing costs. Having fewer units also help saving energy and reducing the amount of generated heat.

So while AMD will call a CPU that has one of these modules a "dual-core" CPU, in reality the CPU isn't true a dual-core product, because there aren't two complete and complete CPUs inside the product. The "dual-core" name in this case will be used for marketing purposes, to make sure the consumer understands that although this Bulldozer-based CPU isn't a true "dual-core" model, it should perform like one.

Going further, for making a "quad-core" CPU, AMD will get two of these blocks and put together, so while physically speaking the processor has actually two "CPUs" inside (two of the building blocks shown in Figure 1), and not four, AMD will still call it a "quad-core" product. In Figure 2, you can see how an "eight-core" CPU based on the Bulldozer architecture would look like.

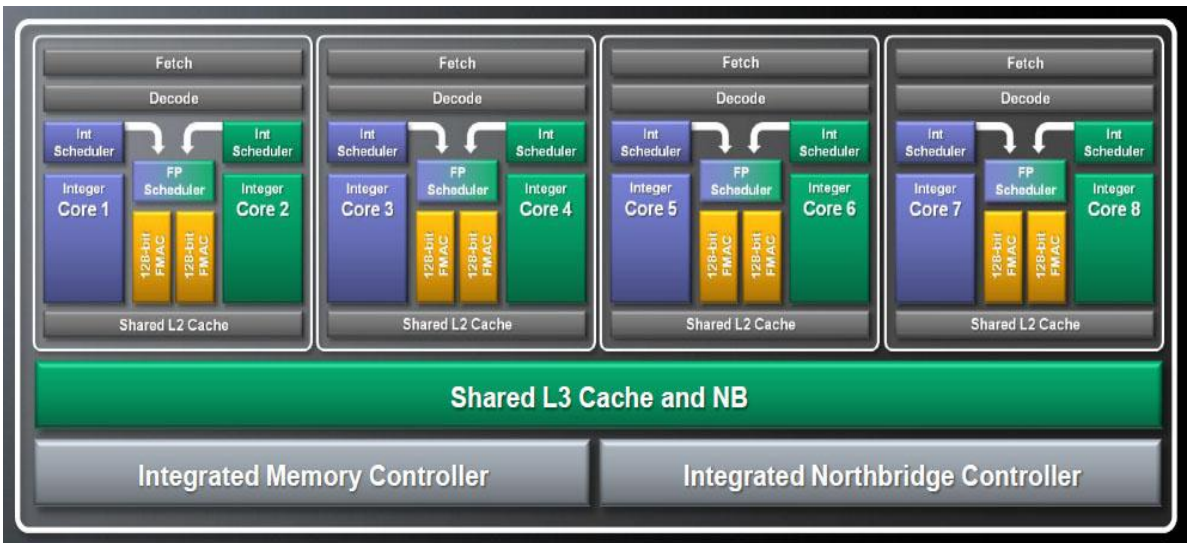

Figure 3.2 Eight-core CPU based on the Bulldozer architecture

CHAPTER -4

THE FETCH AND DECODE UNITS

The Fetch unit is in charge of getting the next instruction to be decoded from the RAM or memory cache.

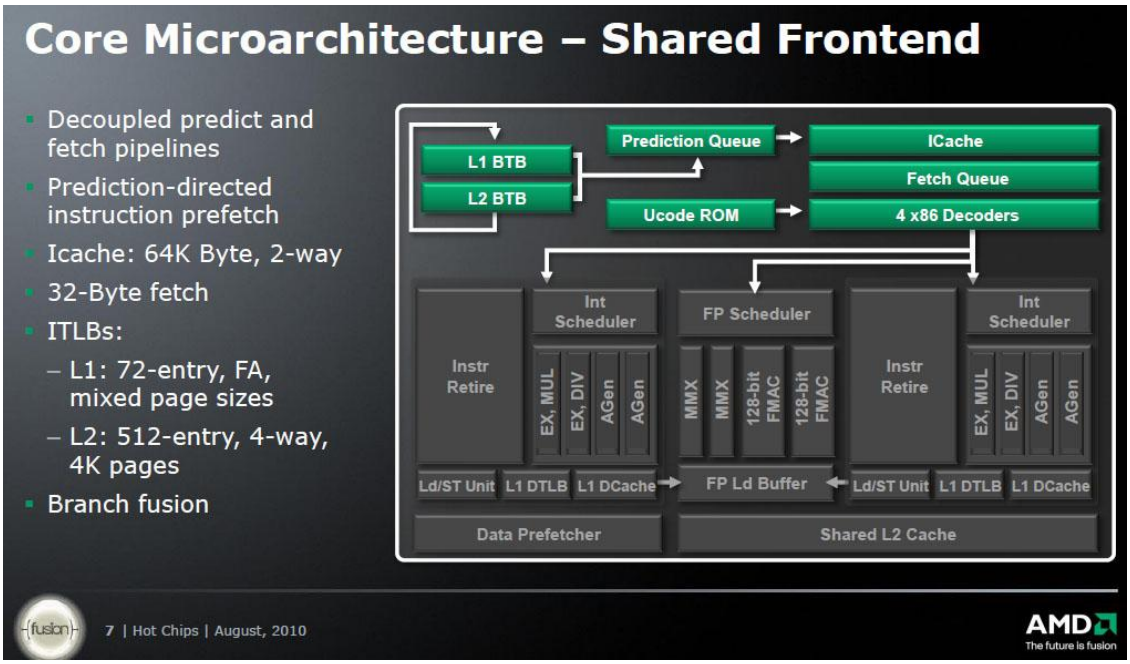

Figure 4.1 The Fetch and Decode units

As shown in the figure, the Fetch unit is shared by the two "cores" available in each Bulldozer module. The L1 instruction cache is also shared by the two "cores", because it is an essential part of the fetch unit, but each CPU "core" has its own L1 data cache. Interesting enough AMD has already announced that the L1 instruction cache used in the Bulldozer architecture is a two-way set associative 64 KB cache, the same configuration used by CPUs based on the AMD64 architecture, with the obvious difference that while AMD64 CPUs have one L1 memory cache per core, Bulldozer-based CPUs will have one L1 memory cache per each pair of cores. However, the data cache used by each "core" will be of only 16 KB, which is considerably smaller than the 64 KB per core currently used by CPUs based on the AMD64 architecture.

At this moment AMD hasn't made public the size of Bulldozer's BTBs (Branch Target Buffers), which is a small memory that lists all identified branches in the program, used by the branch prediction mechanism of the CPU.

The sizes of the TLBs (Translation Look-aside Buffers), on the other hand, have been disclosed, as you can see in Figure 3. These buffers are a small memory to help the conversion between virtual addresses and physical addresses, used mainly by the virtual memory circuit (virtual memory, also known as swap file, is a technique where the CPU simulates that it has more RAM memory than you actually have installed by using a file in the hard drive).

PC programs are written using x86 instructions, but nowadays the CPU Execution unit only understands proprietary RISC-like instructions. So the Decode unit is in charge of converting the x86 instructions provided by the program running into these RISC-like microinstructions, which are the kind of instruction understood by the Execution unit of the CPU. The Bulldozer architecture has four decoders, but at this moment AMD didn't give a lot of information on what kind of instructions each decoder can handle. Usually at least one of these decoders handles exclusively complex instructions, using the provided microcode ROM (in the slide "Ucode" should be read as "μcode", or "microcode"). The decoding of complex instructions takes several clock cycles to be completed, because they are converted into several microinstructions. Simple instructions, however, are usually converted in only one clock cycle because they are translated into a single microinstruction. Usually processor manufacturers optimize their CPUs to decode the most common instructions as fast as possible, in just one clock cycle.

CHAPTER -5

THE EXECUTION UNITS

After the instructions are decoded, they are sent to the appropriate scheduler, integer or floating-point. The Bulldozer architecture has only one floating-point unit, which is shared between the two "cores" available. On the other hand, it has two completely independent integer units, the so-called "cores".

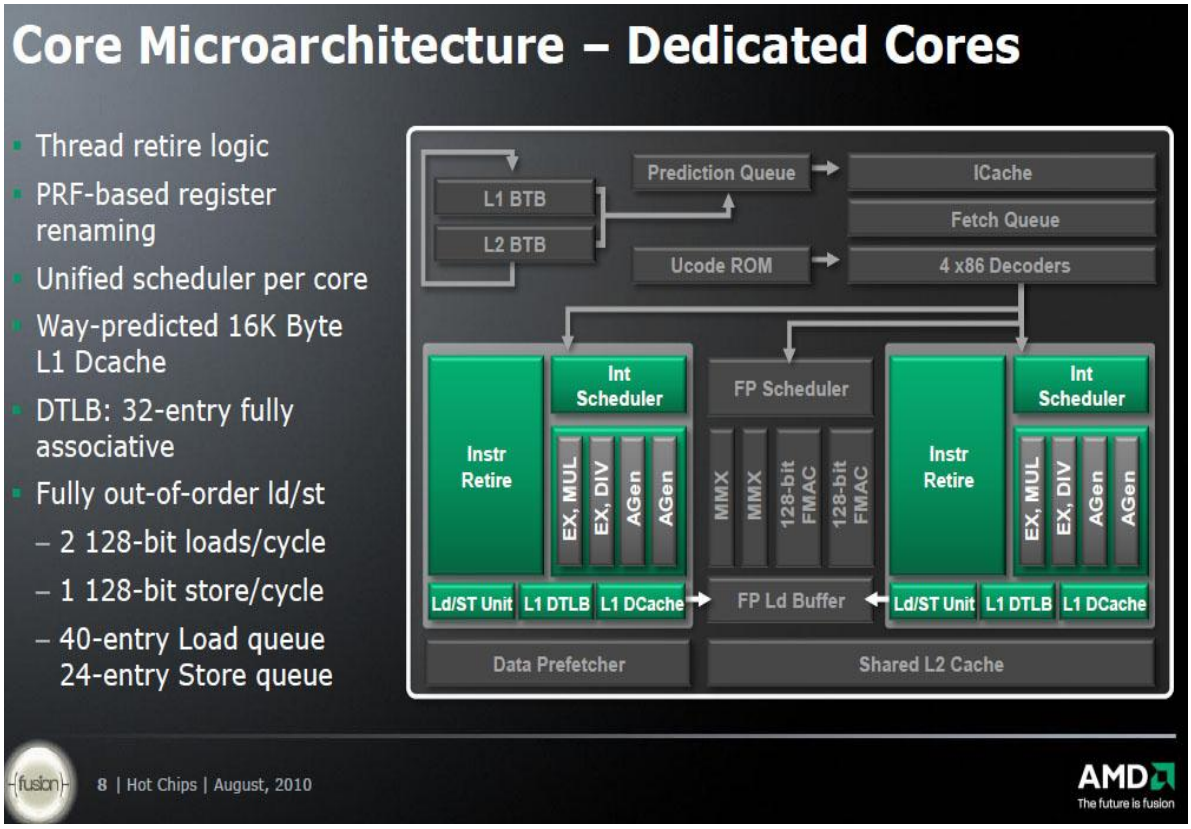

Figure 5.1 The **Execution units**

Each integer engine has four Execution units, labeled as:

- ✓ EX, MUL: Can execute any kind of integer instruction, including multiplication, but not division.
- ✓ EX, DIV: Can execute any kind of integer instruction, including division, but not multiplication.
- ✓ AGen: Address generation, a.k.a. AGU or Address Generation Unit, used to generate the address the CPU will get or store a data.
- ✓ It also has a Load/Store unit ("Ld/ST"), which is in charge of getting from the memory or storing in the memory a data requested by an instruction. Usually this unit is drawn side-by-side with the units listed above, but somehow in this presentation AMD decided to draw it separately.

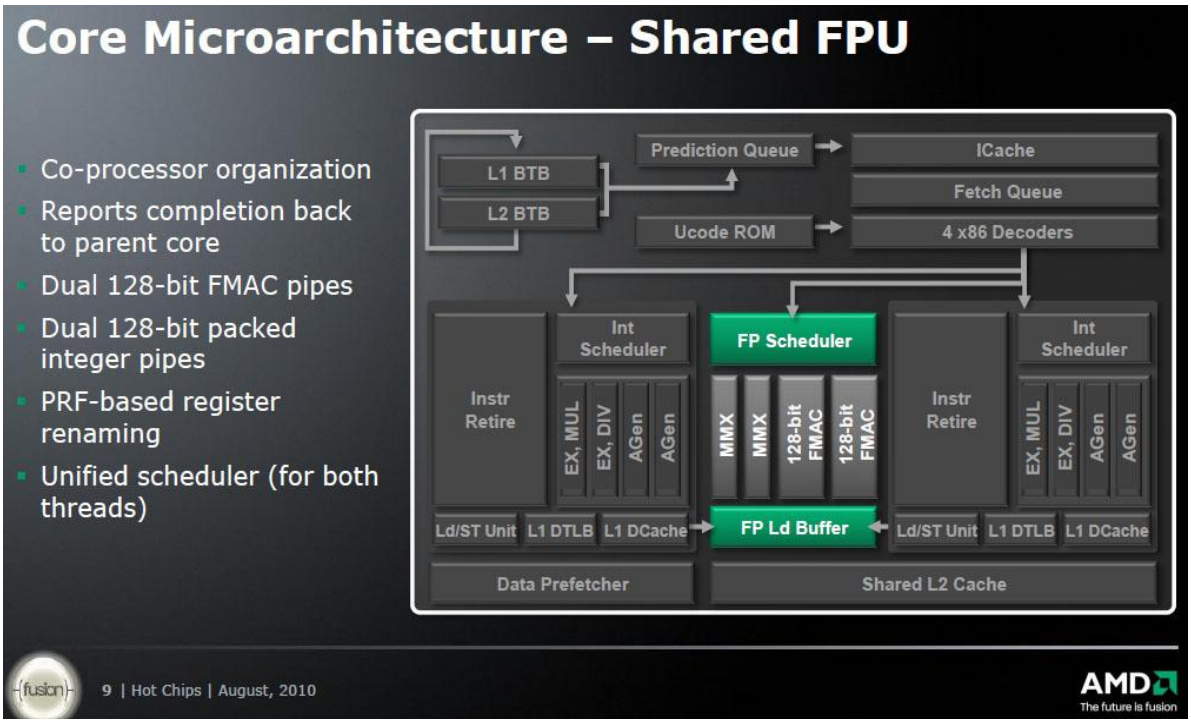

Figure 5.2 The floating point unit

The Bulldozer architecture uses an out-of-order execution engine, like AMD64 CPUs and Intel CPUs since the Pentium Pro (P6 architecture). Because not all execution engines can process all kinds of instructions, if there wasn't an out-of-order engine some of the execution units would be idle sometimes. Let's say the next instruction to be executed is an integer division, but the unit that is able to process this kind of instruction is busy processing another instruction. Instead of waiting for this unit to be free, the scheduler will look for an instruction that can be executed right away in one of the other units, if they are free, of course. So the role of the scheduler is to keep all execution unit as busy as possible.

After integer instructions are executed, they are sent to the Retire unit, where the CPU will put them back in the correct order.

In a similar fashion, there is a small floating point load buffer (not shown above) which acts as an analogous conduit for loads between the load-store units and the FP cluster. The FP cluster can execute two 128-bit loads per cycle, and one of the purposes of the FP load buffer is to smooth the bandwidth between the two cores. For example, if the two cores simultaneously send data for four 128-bit loads to the FP cluster, the buffer would release 256-bits of data in the first cycle, and then 256-bits of data in the next cycle.

The floating point unit also has four Execution units, labeled as:

- ✓ MMX: Can execute all basic floating-point instructions (x87 instructions), including MMX.
- ✓ 128-bit FMAC: Can execute all floating-point instructions.

CHAPTER -6

THE MEMORY UNITS

Perhaps the most profound change in Bulldozer is the load-store pipeline and caches. Other sections of the chip were rearchitected for efficiency or for modest performance gains. However, the load store units were totally redesigned and improved across the board. In tandem, the inner portions of the cache hierarchy have been redone, and for the first time, AMD is fielding competitive prefetchers.

The memory pipeline for each Bulldozer core starts with the load and store queues and the integer scheduler. Any loads or stores in flight must be allocated an entry in the appropriate memory queues. This is necessary to maintain the relatively strong x86 memory ordering model. Previously, Istanbul had a somewhat complex two level load-store queue, where different functions were performed in each level. Bulldozer has a conceptually simpler microarchitecture with a separate 40 entry load queue, and a 24 entry store queue. In total, this means that each Bulldozer core can have 33% more memory operations in flight compared to the previous generation and about 20-30% less than Nehalem or Westmere.

The Bulldozer architecture will have a shared L2 memory cache for each two "cores". An L3 memory cache will be available, shared between all "cores". The L2 memory cache will be a 16-way set associative cache, with a 1,024-entry TLB (Translation Look-aside Buffer).

Bulldozer's L2 cache is shared between the two cores in a module and is mostly inclusive of the L1D caches (recall that the L1D is write-through). The size is implementation dependent and early versions will be either 1MB or 2MB. Open64 compiler optimization notes indicate that Interlagos probably uses a 2MB and 16-way associative design. The load-to-use latency for Bulldozer is surprisingly high: 18-20 cycles, again reflecting a focus on high frequency

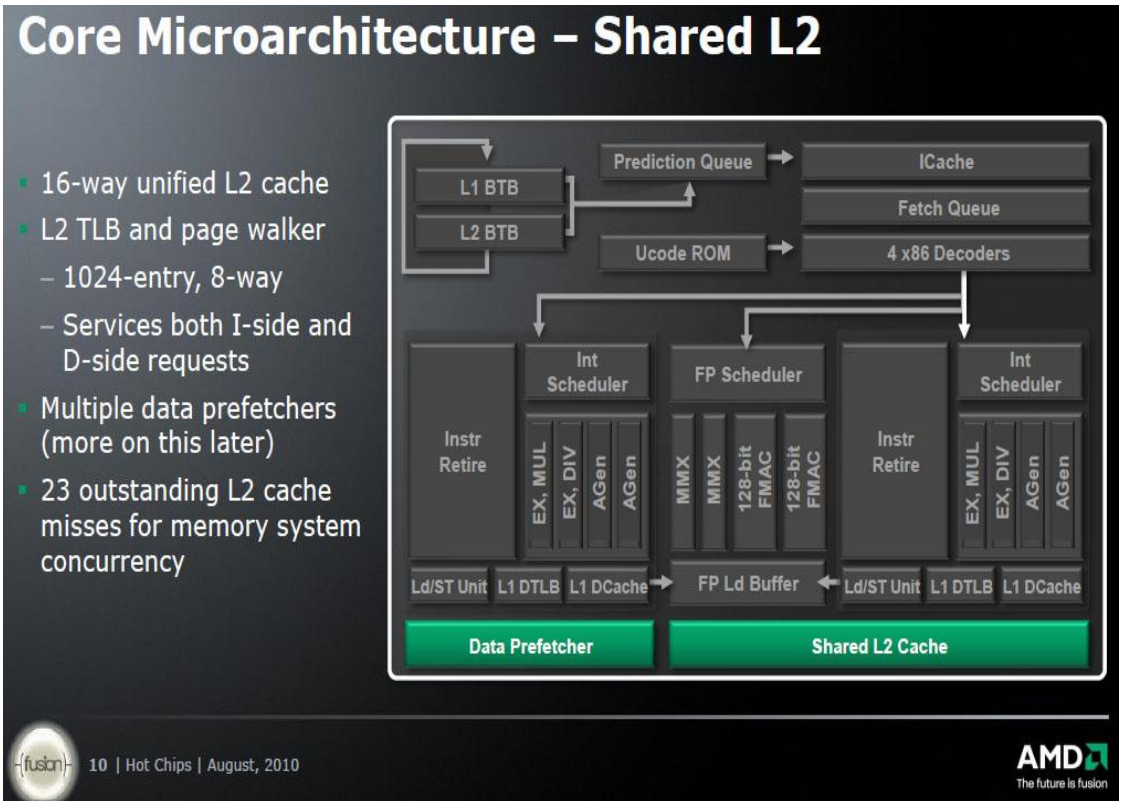

Figure 6.1 The L2 memory cache

The L2 cache can have as many as 23 outstanding misses concurrently, which is a somewhat peculiar number compared to the usual powers of 2. This suggests that some outstanding miss requests may be dedicated for certain purposes. For example, there might be 8 misses outstanding for each L1D cache, with the remainder for use by the L1I cache and prefetchers.

CHAPTER -7

POWER MANAGEMENT

AMD added some interesting features for managing power in their Bulldozer architecture, the most important being "power gating", which allows the CPU to simply cut the power from unused CPU units to save power. It can also completely turn off any unused CPU "core". AMD also added a feature to measure CPU activity to estimate power being dissipated. The phrase "Hardware uses higher frequency when power limit allows" is an indication that AMD is adding a technology similar to Intel's Turbo Boost, which automatically overclocks the CPU if thermal dissipation remains within specs.

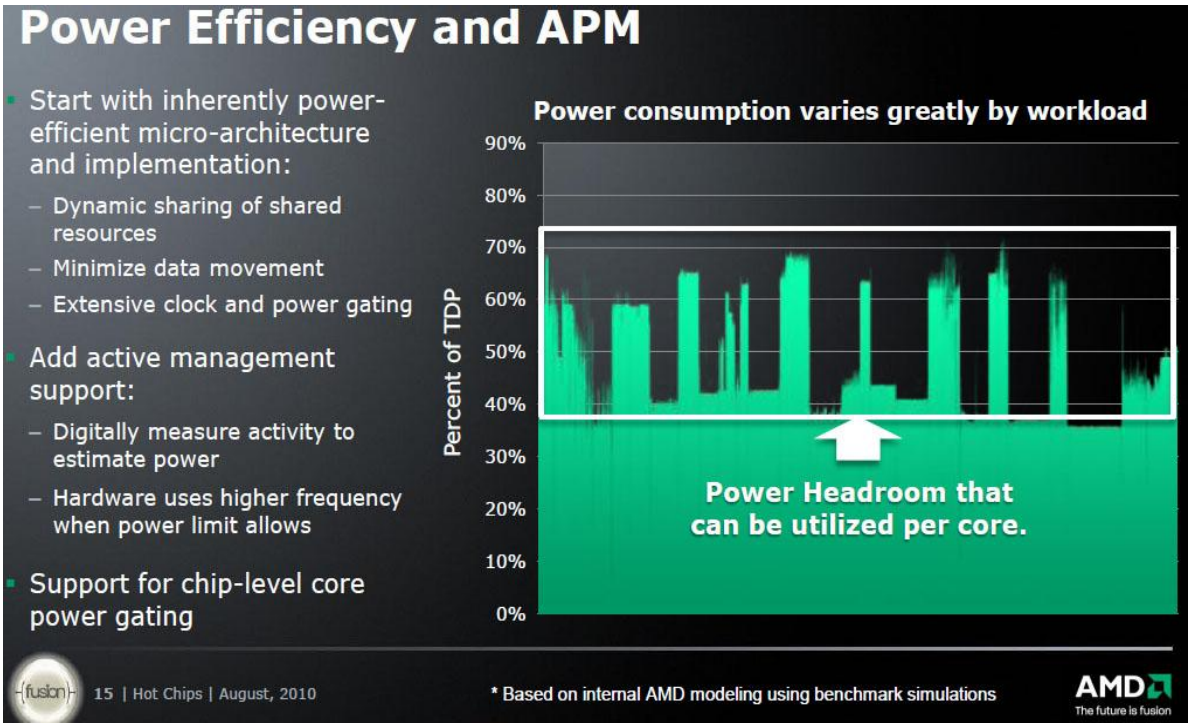

Figure 7.1 Power management

CONCLUSION

This initial peek at Bulldozer reveals some truly new thinking about CPU microarchitecture, and it's undeniably promising in theory. Done well, Bulldozer could restore AMD's competitiveness in both server/workstation processors and high-end desktops, and it could serve as a foundation for continued success for years to come. Unfortunately, it's way too early to speculate on the prospects for products based on this architecture. Purely by looking at Barcelona on paper, one might have expected it to outperform the competing Core 2-based processors and to match up well with Nehalem. The reality was far different from that. Bulldozer's future will hinge on whether AMD can effectively implement the concepts it has introduced here, and we have no crystal ball to tell us what to expect on that front.

REFERENCES

[1] Butler, Mike. "**Bulldozer**" A new approach to multithreaded compute performance. Hot Chips XXII, August 2010.

[2] GCC Mailing list discussion. *http://gcc.gnu.org/ml/gcc/2010-06/msg00402.html*

[3] Jotwani, R. et al. "**An x86-64 Core Implemented in 32nm SOI CMOS**," Proceedings of International Solid State Circuits Conference, pp 106-107, February 2010.

[4] Conway, P. et al. Blade Computing with the AMD Opteron Processor ("Magny-Cours"). Hot Chips XXI, August 2009.