



Niagara 2: More Heft in the Weft

Research Note

Gordon Haff

24 August 2007

Licensed to Sun Microsystems, Inc. for web posting. Do not reproduce. All opinions and conclusions herein represent the independent perspective of Illuminata and its analysts.

Multiple cores on each physical microprocessor chip are now the norm in processor design. Vendors still use a variety of design and manufacturing approaches to wring the best performance out of a transistor budget that continues to increase in accordance with Moore's Law. However, today, performance gains come far more through core addition and other ways of handling multiple program threads in parallel than was the historic norm. Indeed, it seems likely that we are now at a point where notable improvements in how quickly we can process an individual thread will increasingly be the exception; most speedups will instead take place at the level of the aggregated threads that make up most applications.

However, with its "Niagara" line, Sun has taken a more radical approach to multi-threading. The likes of AMD, IBM, and Intel are adapting their designs to multi-core thinking, but keeping individual cores pretty much as fat and fast as technology allows. The Fujitsu-designed SPARC64 VI "Olympus" processor in the Sun/Fujitsu "APL" product line—like Sun's own UltraSPARC IV+ chip—takes a

similarly incrementalist approach.¹ There's nothing incremental about Niagara. The UltraSPARC T1, the first product in the lineup, was formally unveiled in late 2005; it can switch among four threads on each of eight cores, for a total of 32 simultaneous threads.² In design philosophy, the T1 has much more in common with many-core network processors or Azul's Vega processor for Java processing than it does historic general-purpose microprocessor designs.³

Sun is now rolling out its second-generation Niagara processor, the UltraSPARC T2. This generation stays with eight cores but doubles the number of threads per core to eight. It also adds more processing resources to both the individual cores and the chip as a whole. These include dedicated floating point units for each core—all cores share one floating



¹ See our [Does APL Matter?](#).

² Some versions sold support fewer than 8 cores. See our [Gradations of Threading](#) for an in-depth discussion of multi-threading on Niagara and other processors.

³ See our [Azul's Assiduous Advance](#) and [Azul: A New Shade of Server](#).

point unit in the T1—and a variety of network and security protocol accelerators. The T2 remains extremely multithreaded by conventional standards. However, while retaining its focus on thread-level parallelism (TLP), the T2's additional hardware represents a significant evolution from a design that was *solely* focused on TLP to one that's well-equipped to handle a considerable range of thread-rich workloads.

From Latency to Watts

Sun's main pitch for its UltraSPARC T1 has been "eco-friendliness," which is to say, power efficiency. Given that datacenter power and cooling concerns are one of the hip, hot themes of the moment, and that the T1 indeed offers attractive performance-per-watt metrics, it's neither surprising nor risible that Sun has decked out the T1 in green garb. However, many of the enhancements that Sun has made in the T2 generation are better understood viewed through a different lens—the one through which Sun initially presented its Chip Multithreading (CMT) strategy.

The UltraSPARC T1 was derived in part from technology that Sun gained when it purchased Afara Websystems in 2002, coupled with learnings from Sun's own MAJC graphics chip. It was undoubtedly intended to deliver good bang for the watt—but power tradeoffs are part of all processor designs. While the T1's 72 watt typical power draw is on the low side for processors used in servers, it's not strikingly low—especially compared to lower-speed, and hence lower-powered, versions of the competition. Rather, the T1's strength is in the performance it delivers on its target workloads relative to the power it consumes. It was that performance that was the real design target for the T1 rather than its power draw. Multi-threaded performance *at the system level* ameliorated the effects of delays associated with getting instructions and data from memory. That the T1 could be recast in terms of power efficiency was something of a felicitous side effect—albeit one

that Sun marketing exploited to highly effective (and amusing) effect.⁴

The fundamental issue is that while memory *capacity* has roughly kept pace with processor performance—in large part because both are driven by increasing transistor densities—memory *performance* has not. Bandwidth (the amount of data that can be transferred in a given amount of time) has more or less kept up, but latency (the time that it takes to return a result from memory) has improved much more slowly. Memory latencies over the past decade have lagged processor performance by at least an order of magnitude. As a result, even using techniques such as on-processor caches and out-of-order execution, processors have tended to spend more and more time spinning idly, waiting for memory to give them data to process.⁵

Thread-level parallelism (TLP) is one approach to deal with this disparity between processor speed and memory speed. With TLP, the chip handles several chains of instructions at once, efficiently switching away from tasks that are waiting for data to arrive, working instead on tasks that already have their data ready for processing. Underlying TLP is the fact that modern server programs tend to be extensively multi-threaded, containing many streams of code that can execute more-or-less autonomously. TLP can't speed up memory, but it can help optimize for a world in which memory is slower than the processors. If a processor core waiting for data is one of many lightweight units, rather than one of a small number of heavier editions, the overall performance cost to the system is much reduced. If it's only one execution slot on a multi-threaded core doing the waiting, the penalty is less still.

It is this desire to juggle a great number of threads in parallel that underlies the designs of the Niagara family. For example, the T2 doubles the number of threads that it can handle relative to its predecessor, while keeping the core count the same. This provides twice the number of "slots" that the chip

⁴ See our [Sun Finds a New Evil Empire in Round Rock](#).

⁵ For a more detailed discussion, see our [Latency Matters!](#) and [Breaking Up The Microprocessor Monolith](#).

Processor Name	UltraSPARC T1	UltraSPARC T2
Codename	Niagara	Niagara 2
Cores x Threads per Core	8 x 4	8 x 8
Process technology	90 nm	65 nm
Floating point units	1 (shared among 8 cores)	8 (1 per core)
On-chip networking & I/O	None	Dual 10 Gigabit Ethernet PCI Express
Cryptographic acceleration (per-core)	Modular Arithmetic Unit	Modular Arithmetic Unit Cipher/Hash Unit
Frequency	1.0 and 1.2 GHz	900MHz to 1.4 GHz
L2 cache	3 MB	4 MB
Memory	Four memory controllers, up to 16 DDR2 DIMMs	Four memory controllers, up to 64 FB-DIMMs
Power consumption	Up to 79 watts	Up to 123 watts

can keep in play while waiting for data to arrive from memory. The T2 is still fairly aggressive relative to mainstream competition in terms of the number of cores that it contains—but it's in the number of threads that each core can switch among where it truly puts the hammer down. The T2 also adds specialist co-processors for floating point, security, and networking acceleration. In short, the T2 is all about boosting performance in a multi-threaded environment.

The Basics

At the heart of the UltraSPARC T2 are its eight 1.2 GHz or 1.4 GHz cores, each of which can switch among up to eight threads, for a total of 64 threads per chip. Each core has two integer execution units with an eight-stage, in-order pipeline. This is a simple design by current microprocessor standards, reflecting the T2's emphasis on thread-level rather than instruction-level parallelism. TLP is achieved by more cores and more threads, whereas ILP is achieved by increasing the complexity and frequency of individual cores. By comparison,

Intel's current Core 2 generation has a 14-stage pipeline; various of its "NetBurst" predecessors (Pentium 4 and Xeon) used up to 31 stages.

In addition to the two integer units, each T2 core also has a floating point unit with a 12-stage pipeline.⁶ This represents a considerable advance over the T1, which shared just one floating point unit among all eight cores on the processor, and thus was not suitable for applications that need more than a modicum of floating point computations. By using the additional transistors that the T2's 65 nm process technology makes available (relative to the T1's 90 nm process) for seven additional floating point units, Sun has effectively crafted the T2 into a more general-purpose processor than its predecessor.

Each core connects to L2 caches and the rest of the memory hierarchy through a pipelined, non-blocking eight-port crossbar switch that can handle eight load/store requests and eight data returns

⁶ The pipeline for division and square root operations is somewhat longer.

simultaneously. The processor as a whole has 4 MB of L2 cache, shared among all eight cores. Cache is divided into eight banks but any core can directly access any L2 bank through the crossbar switch—that is, a given L2 bank has no special affinity with any particular core. This may seem an exceptionally small amount of cache—and it is, by any competitive standard. But this is where the many-threads-per-core, TLP design kicks in most fully; the whole point of the TLP design is to work around memory latency, rather than try to hide it with large caches.

Architecturally, the T2 can support up to 512 GB of Fully-Buffered DIMM (FB-DIMM) memory—although the physical limit will depend upon the implementation of a given server design. FB-DIMMs might seem a bit of an odd choice, given that current FB implementations consume more power than parallel DDR2 or DDR3 memory designs. However, Sun elected to go with FB-DIMMs to keep the processor's pin count reasonable even for large memory configurations.⁷ This is another example of the T2's design's being driven by overall system capabilities rather than solely a frugal power budget.

In general, the memory hierarchy is designed for performance at the multiple thread level rather than for maximizing performance of any given thread. We've already noted the relatively small cache size. In addition, the crossbar used to connect the cores to the L2 caches imposes a high latency on L2 cache accesses—about 20 cycles.⁸ The design is optimized to handle a multiple of such accesses in parallel rather than to make any single one as fast as possible.

Like the T1, the T2 also implements its own flavor of virtual machines, Logical Domains (LDoms). LDoms are implemented with a thin layer of firmware coupled with hardware extensions that provide isolation between partitions. Each server

based on an UltraSPARC T2 chip can have up to 64 LDoms—i.e. one per thread. LDoms are complementary to the Containers (a.k.a. zones) that Sun introduced as part of Solaris 10. Containers are an extension to Unix resource management groups and are created within a single operating system copy; they're sometimes referred to as operating system virtualization for this reason. LDoms, by contrast, are a virtual machine implementation akin to what VMware ESX Server does in software; each domain hosts its own independent copy of an operating system. Both Solaris and Ubuntu Linux support LDoms.⁹

Accelerating Cryptography

Beyond its fundamental multi-threading orientation, the T2 also aggressively integrates a variety of specialized hardware acceleration functions. The T1 took some initial steps in this direction by including a Modular Arithmetic Unit (MAU) in each core. The MAU handles operations associated some of the more popular cryptographic functions such as RSA (a public key encryption algorithm), DSA (an algorithm used for digital signatures), and DH (the Diffie-Hellman key exchange used when initiating key exchanges to establish encrypted links like SSL that enable secure access to web sites).

The T2 adds a cipher/hash unit to complement the MAU. This adds support for bulk encryption (DES, 3DES, RC4, AES) and secure hash (MD5, SHA-1, SHA-256) algorithms that are widely used in a variety of security functions and protocols such as IPsec. The T2 also adds hardware acceleration for Elliptic Curve Cryptography, which many view as an important direction for public key cryptography because it can provide equivalent levels of protection as older algorithms without requiring extremely long key lengths.¹⁰ These functions would typically be accessed through the Solaris Cryptographic Framework in the same manner as

⁷ The UltraSPARC T2 package has 711 signal I/O pins out of 1,831 pins overall.

⁸ If the required data isn't in the L2 cache, it takes a minimum of about 104 cycles to load from main memory.

⁹ Ubuntu is Sun's preferred, if unsupported, Linux distribution for SPARC.

¹⁰ See, for example, this discussion: www.nsa.gov/ia/industry/crypto_elliptic_curve.cfm

they would were the operations being handled entirely in software.

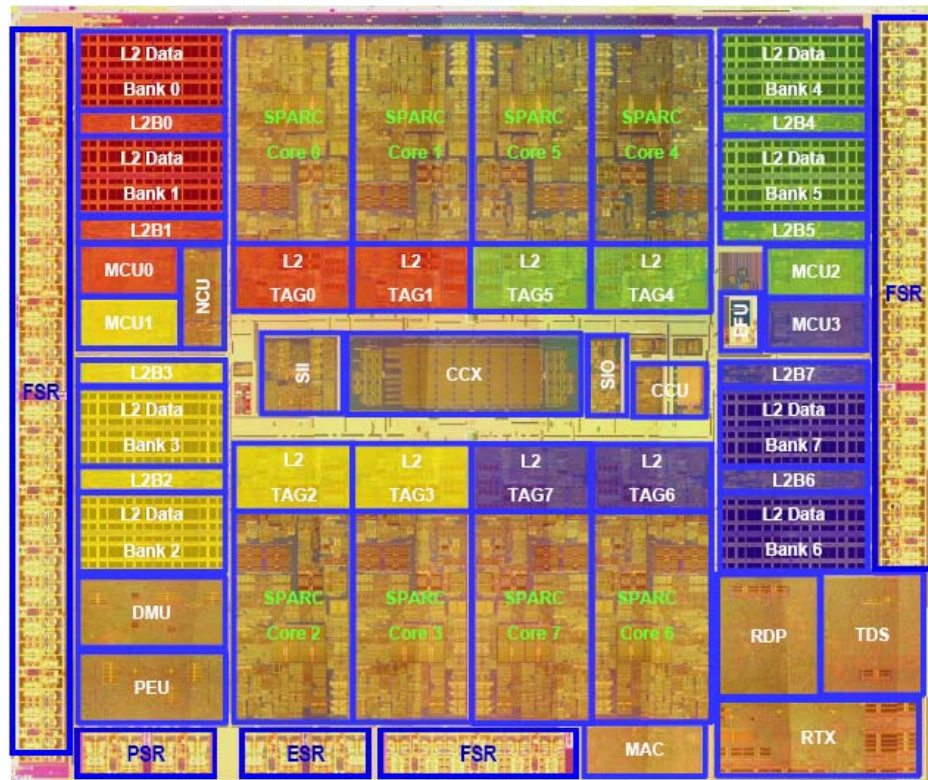
Cryptographic offload in servers has mostly been of interest in systems where the general purpose “MIPS” are relatively expensive—such as on the IBM System z. Although cryptographic functions tend to be computationally intensive, the cost of buying additional specialized security hardware for racks of cheap x86 servers has tended to mean that secure links and other forms of security only get used when it’s really needed—such as for financial transactions. However, if such capabilities are built into the base silicon as in the case of the T2, that’s an attractive feature in a world where there are pressures to encrypt more and more data and secure more and more communications. The performance advantages of hardware acceleration depend on many factors, but Sun’s benchmarks suggest up to about 10x better performance than general purpose x86 processors—with far less CPU overhead.

Networking On-Chip

The T2 also puts dual 10 Gigabit Ethernet ports onto the chip. (The ports can also run at 1 Gigabit/sec.) This is effectively the integration of Sun’s “Neptune” ASIC onto the microprocessor.¹¹ In part, this is about reducing the number of external support chips needed in the system as a whole—reducing both power and cost. However, the T2’s integrated networking also works in conjunction with Solaris 10 and the T2’s multi-core

architecture to provide a form of I/O virtualization. The basic idea is to take the many packets and connections associated with the external network, break them up into smaller workloads, and parcel them out in an efficient manner to the chip’s many threads. Sun calls the overall architecture “Project Crossbow.”

The process starts with classifying the packets as they arrive using the packet classification engine that is part of the on-chip network hardware.



Packets can, for example, be classified on the basis of IP addresses or an upper-level protocol and assigned to a virtual NIC. Using this information, incoming network traffic can then be associated with a “flow” which corresponds to a given protocol, operating system service, or a virtual machine (such as a Logical Domain or Solaris Container). The *squeues* introduced in Solaris 10 as part of “FireEngine” have been extended to control the resources used by flows. They replace the interrupt-driven packet processing by a lower-overhead polling mechanism by which the *squeue* fetches packets from the hardware.

¹¹ On the chip photograph, the RDP, TDS, RTX areas in the lower right corner are the integrated networking.

The net result is to create dedicated lanes through the protocol stack from the processor. This process helps to organize network traffic and reduce the opportunities for contention over resources. Sun's data shows the most significant gains with small (64 byte) packet sizes, although there are some performance gains with larger packets as well.

Conclusion

These days, there's nothing particularly controversial about the UltraSPARC T2's basic thread-centric design approach.

In part, this reflects an industry that now acknowledges—essentially universally—that wringing performance out of future processors will increasingly depend on techniques oriented to executing many threads in parallel rather than handling just a few with great speed. That's not to say that there aren't the usual disputes over the best ways to implement thread-level parallelism, or the optimal balances to strike between competing concerns such as power, processor size, and various performance metrics. And, indeed, those optimal balances depend in no small part on the workloads and applications on which a given vendor has chosen to concentrate.¹² However, there's little dispute that, as a general concept, multi-threading is the future. The Niagara line—and especially the

¹² For example, the need to support a huge desktop application base—which tends to be a less multi-threaded environment than in the case of servers—was a major factor in Intel's being slow to embrace thread-level parallelism.

second-generation T2—implements the TLP concept well.

At the same time, many of the UltraSPARC T2's new features make it not just more capable than its predecessor, but a more general-purpose processor as well. The addition of floating point units to each core, enhancements to cryptographic processing, larger memory support, and advanced virtualized networking add capabilities that go beyond being able to just handle a lot of threads in parallel. While the T2 supports twice as many threads as the T1, most of the additional transistors gained from the 90nm to 65nm process transition go to speeding up widely-used operations such as floating point, cryptography, and network packet processing. The transistors are also devoted to putting system-level interfaces such as 10 GbE and PCIe on-chip.

The UltraSPARC T2 still pushes the boundaries of thread-level parallelism further than most. Coupled with the crypto acceleration and on-chip networking, this makes the T2 an even better foundation for the network facing workloads that Sun so covets. But Niagara is no longer purely fixated on network workloads. Sun has evolved this processor family into something considerably more general-purpose. What were once fringes are now the mainstream.



Through subscription research, advisory services, speaking engagements, strategic planning, product selection assistance, and custom research, Illuminata helps enterprises and service providers establish successful information technology.