



*Announcer* You're listening to the Sun Microsystems Podcast Network. Welcome to another edition of *Innovating@Sun* with your host Hal Stern. Today's topic: the UltraSPARC T2. And now here's Hal Stern.

**Hal:**

**Hello and welcome to another episode of Innovating@Sun. I'm your host Hal Stern, vice president of Global Systems Engineering, and with me for this show is Rick Hetherington who's distinguished engineer and chief technology officer of Sun's newest business unit, Sun Microelectronics. So, Rick, welcome to the show.**

Rick:

Thanks, Hal. Thanks for having me.

**Hal:**

**So you have possibly one of the more fun jobs is you get to design the chips that, certainly on the SPARC side, power just about everything that we do.**

Rick:

Yeah. It's been a fun several years here with Niagara CMT and we're just thrilled with the success we've had so far with Niagara and Niagara 2 is a great chip, and we're really excited to reveal this to the world.

**Hal:**

**So you talked about Niagara 2, the continuation of the work we started with chip multi-threading and the first Niagara UltraSPARC T1 processor. Clearly one of the things we've been talking about for a while is this notion of a system on a chip, system-level integration down at the silicon level. Tell us what makes that work.**

Rick:

Well, first and foremost, of course, we need a technology that allows us to migrate many cores and system functions onto the die. Niagara 2 is designed in 65-nanometer technology, and with that, we were able to put eight cores, 64 threads, a level 2 cache, memory controllers, network interface unit as well as PCI express root complex all on one piece of silicon. So we've arrived finally and can make a statement that we have a true server on a chip with Niagara 2.

**Hal:**

**In terms of where we're going from the Niagara 1 baseline, new things in Niagara 2, different floating point architecture, memory controller, crypto-interface. Tell us some more about what the secret sauce is there inside the silicon.**

Rick:

Sure. Let's start with the memory controller first and foremost. We deployed DDR2 in Niagara itself and as you pile on the amount of compute capability, it puts a great deal of pressure on memory. So as we went from 32 threads to 64 threads, we definitely had to increase bandwidth throughout the memory hierarchy, and we took the

Level 2 cache and went from four banks in Niagara to eight banks in Niagara 2. And then on the other side of the banks, each of those pairs of banks in Niagara 2, are connected directly to a memory controller, and the memory controller of choice was fully buffered DIMM for Niagara 2. Now, there is a power increase that one does experience with SPDM 2, but from the perspective of providing the capacity required and the bandwidth required fully buffered DIMM was the only choice. We've more than doubled our bandwidth. We've vastly increased our capacity. We didn't have to trade off one against the other, and we actually saved on pins in the long run, so you can see the package benefits from that. So a radical reduction in pin count in Niagara 2, yet a big bump in bandwidth. On the crypto piece we had cryptography in Niagara itself, and so that functioned as accelerating RSA, so we call it a modular arithmetic block. And the concept of a co-processing element within each core was, again, spawned by Niagara, so we embellished that further in Niagara 2 by adding all the favorite bulk ciphers, this is DES, triple DES, RC4. We have hashing functions for authentication, and that includes SHA 1, SHA 256 MD5. Again, we support modular arithmetic for RSA for key exchange as well as elliptic curve cryptography. So we've future-proofed and we think we covered 99.9 percent of all the encryption and security needs for the market. So it is a feature that we are stressing that, as security becomes more and more important, a crypto co-processing element alleviates the cores from having to do that function. And so we have zero-cost security that we can offer our customers.

**Hal:**

**So I want to go back for a second. Talk about thread architecture. With our first multi-threading processor Niagara, we introduced the concept and delivered on it in silicon. Here we go to eight threads and eight cores, so it's a bigger, wider chip. What'd we learn and where do you see that going?**

Rick:

Well, we learned our threading model on Niagara was really a big benefit. We chose that model correctly and we wanted to add to that, not double our performance, our throughput performance, but don't vector away from the holy grail, and that was a very simple, single issue: in-order pipelines that are energy efficient. And then what we actually did was we have a common fetch unit and two independent pipes per core and four threads fixed on each of those cores. So from an efficiency point of view, we added a floating point unit, cryptographic unit and doubled the number of threads on a per-core basis, yet the dimensions of the core are identical. They're about 11 square millimeters in Niagara 1 and 11 square millimeters in Niagara 2, so the dimensions didn't increase even though we added that much functionality. And that's all through the beauty of 65 nanometer versus 90 nanometer.

**Hal:**

**So I would say some of the initial trade-off points you've made with the first Niagara processor in terms of going to multiple cores but with less floating point. Now, in Niagara 2, it looks in many ways much more like a – I never want to say regular processor because I think Niagara's about as regular a processor as you can get, certainly looking forward. But, you know, it looks a lot more balanced in terms of capability. Plus you add in things like the integrated networking, integrated crypto. You know, there is some stuff that goes beyond the typical microprocessor memory.**

Rick:

Yeah. On the floating point, we examined all of the commercial applications that we were targeting Niagara to initially several years back, and they are typically void of floating points. So it was the right trade-off at the time. We had a fairly large dye with a single floating point unit, but we did have to caution our customer base and, in fact, we provided a tool called Cool Test that would examine workloads to report back floating point content to the customers and alert them that perhaps Niagara was an inappropriate processor. That's all behind us now. With Niagara 2, floating point has been strengthened to 10x over Niagara itself, and so they're no longer required

to examine their code. I think we're over the hump on that issue.

**Hal:**

**Okay. There's one other thing I want to bring up before – talking about power and the software side of the world. If you look at the evolution of processors that we've gone through from really thinking about deep pipeline into multi-threaded architecture now, you know, more scalable multi-threaded architecture, your emphasis on fully buffered DIM, on integrating networking. There are a lot of things that become sort of fundamental building blocks that you're putting right into the processor that you're making freely available. I'm struck by people's definition of commodity. To me, a commodity is you take advantage of things that are available at scale and you make them available at scale in some useable form, whether it's people taking oil and refining it and turning it into gasoline or generating electricity and getting it out to your house. Here you're taking the fundamental elements of computing, fast memory, scalable memory, fast crypto, scalable crypto and a truckload of threads and making them available to the typical programmer through an interface that looks like just about any other SPARC processor.**

**Rick:**

Yeah, that's true. I think we're not putting a great deal of pressure on the software community to change their programming model to a great extent. Our performance is really achieved through parallelism, but it's parallelism on workloads that are naturally threaded at the moment. In future, I think we'll begin to see more single-threaded apps that understand the parallelism underneath and be able to take advantage of that. But if we look at the cores and examine the cores themselves, there's a lot of, let's say, retro kind of features that are in there. We've kind of stepped backwards from all of the very complex speculation techniques that sound its way into processor technology as we chase that single-thread performance. And I think we took a step back, and it was a bold move to take a step back and say we can derive ultimate throughput performance by simplifying our cores and providing a wide number of cores and threads in order to achieve that throughput.

**Hal:**

**So if I define commodity as something that's broadly applicable and has a lot of the uses – is fungible from one particular application to another one, you just defined it.**

**Rick:**

I think so, Hal. I think we've not run into very many applications even with Niagara where it wasn't appropriate. So for web-serving functions, app-serving functions, database, and now into the HPC market. So we have a chip that stands up quite well in the technical market.

**Hal:**

**Okay. So you touched on HPC, and clearly people start to think about massive scale and lots of cores, lots of processors there. And before you hinted a little bit at the power dissipation and power usage on Niagara 2. Where does it stand versus Niagara 1 and, again, what did we learn – what do you see as the differentiator here?**

**Rick:**

Well, overall, if you measured absolute power, Niagara 2 is slightly higher than Niagara 1. We've added more functionality in terms of the additional floating point units. And the network-interface unit as well as the root complex for PCI express. So the latter two for IO itself adds about 15 to 20 watts to a chip that you would otherwise find externally as chip sets on our competitors' processors. So from a power perspective, power performance per watt, if we measured in that extent, we definitely have taken – extended – the performance per watt on Niagara 2 by about 50 percent. So the chip itself, it depends on the workload and the frequency, so we'll

be announcing two speed bins, 1.2 and 1.4 gigahertz. At the very high end, let's say 1.4 gigahertz running a floating point-intensive application, you might see power as high as 120 watts. On the lower end at 1.2 gigahertz running a job application, Niagara 2 can be as low as 80 watts, so we operate in that range of power. But one single Niagara 2 chip typically has performance of two sockets of all of our competitors. That's essentially where we're at. And then being the leader in power efficiency, we paid special attention to how we can manage power internally, so we do have features like clock-gating, extensive clock-gating throughout the core. So if one is not running a floating point application, the clocks are not being submitted to that floating point unit. So that is a savings of power there. We have a savings of power in our fully buffered DIM memory where we can control clocks and turn off clocks to ranks that are not being accessed. And so that has a significant drop in power even though SPDM has a reputation for being a power-hungry design.

**Hal:**

**I guess it's that kind of innovation around thinking about there's a certain basic cost of having something powered on, and a lot of the design characteristics you just talked about drive down that powering-it-on cost by looking at what's really being utilized inside the silicon.**

Rick:

That's exactly it. And then, of course, then it required software to manage it as well, and so there's been significant features added to Solaris to understand how to schedule on during those lightly loaded periods so that we can actually migrate workloads down to fewer and fewer cores and be able to disable cores and then save power on that. So that feature and function is working quite well.

**Hal:**

**So to continue down that thread, talk a little bit about virtualization and OS support. What's it run?**

Rick:

It runs Solaris and Solaris 10 will be – shipping with Solaris 10 update 4. We do support containers and logical domains for virtualization. So with logical domains, you can have 64 independent domains, each running their own version of Solaris across the entire chip. So lots of flexibility in virtualization. The customer can choose to use containers or choose the LDOMs or a combination of the two. So a domain could have containers as well. So I think we have an excellent virtualization story on Niagara 2.

**Hal:**

**How about Linux?**

Rick:

There's already been on kernel.org, the main line Linux does have the put-back for supporting Niagara 2 and LDOMs. So I would imagine the next release of Ubuntu will have a distro for Niagara 2.

**Hal:**

**Okay. So I want to hit you with the last question here. We talked about open sourcing Niagara 1. Where are we with open source of essentially the software fabric that underlies the silicon in Niagara 2?**

Rick:

Well, we plan to do exactly what we did in Niagara, and so at announcement, we'll be announcing OpenSPARC for the UltraSPARC T2. So we will definitely open it up. Exactly when, I think we'll announce at a later point, but our full intention is to open SPARC and follow that same pattern that we did with Niagara.

**Hal:**

**Great. So get the IP out there. Get people looking at it. Have people start to think about where we see the world going in terms of very highly threaded, very scalable microprocessors.**

**Rick:**

Yeah. We're all set and ready. The chips in great shape and it's all ready to go.

**Hal:**

**Great. Well, Rick, I'd like to thank you for taking the time to talk about where we are and where we're going with the world of CMP and, of course, some exciting news around the influx of Niagara 2. You've been listening to another episode of Innovating@Sun and I'm your host Hal Stern.**

*Announcer      You've been listening to Innovating@Sun. Join us next time for the latest in innovation from Sun Microsystems only on the Sun Microsystems Podcast Network.*