Leveraging OpenSPARC

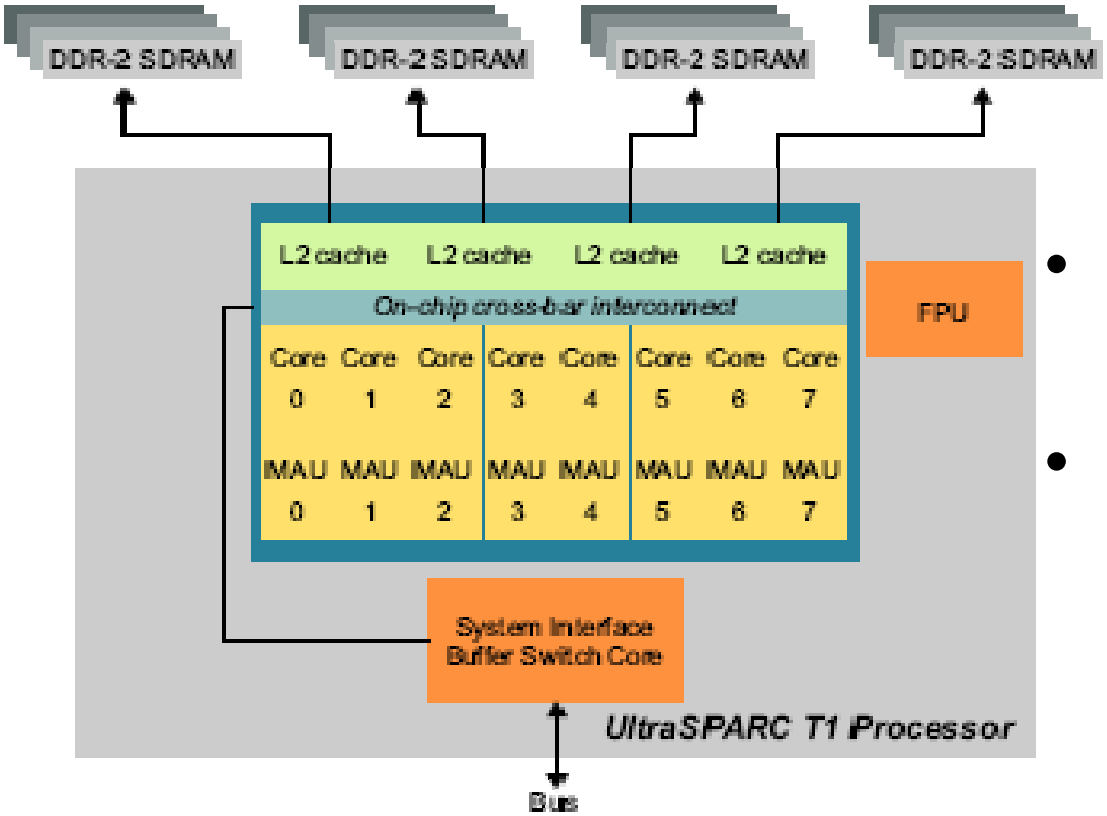
ESA Round Table 2006 on Next Generation Microprocessors for Space Applications

G.Furano, L.Messina – TEC-EDD

OpenSPARC T1

- The T1 is a new-from-the-ground-up SPARC microprocessor implementation that conforms to the **UltraSPARC architecture 2005** specification and executes the full **SPARC V9 instruction set**. Sun has produced two previous multicore processors: UltraSPARC IV and UltraSPARC IV+, but UltraSPARC T1 is its first microprocessor that is both multicore and multithreaded.
- The processor is available with 4, 6 or 8 CPU cores, each core able to handle four threads. Thus the processor is capable of processing up to 32 threads concurrently.
- Designed to lower the energy consumption of server computers, the 8-cores CPU uses typically **72 W** of power at **1.2 GHz**.

72W … 1.2 GHz … 90nm …

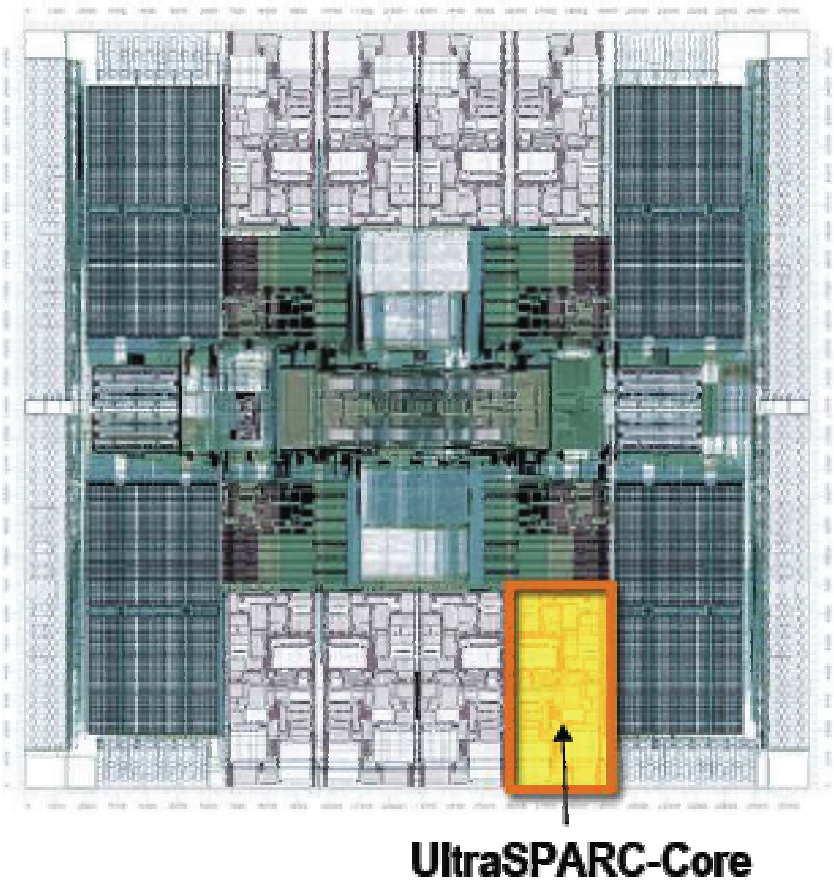


- Is a cutting edge design, targeted for high-end servers.
- NOT FOR SPACE USE
- But, let's see which are the potential spin-in …

Why OPEN ?

On March 21, 2006, Sun made the UltraSPARC T1 processor design available under the GNU General Public License. The published information includes:

- Verilog source code of the UltraSPARC T1 design, including verification suite and simulation models
- ISA specification (UltraSPARC Architecture 2005)
- The Solaris 10 OS simulation images
- Diagnostics tests for OpenSPARC T1
- Scripts, open source and Sun internal tools needed to simulate the design and to do synthesis of the design
- Scripts and documentation to help with FPGA implementation of parts of OpenSPARC T1 design including SPARC core, Floating point Unit, Cross-bar



Target market : driving reqs

- “Walls” facing server CPUs:
  - Memory latency (75%!), load-load dependencies ? Fine multi-threading (1 cycle context switch, 640 64-bit register file, round robin)
  - Branch prediction ? minimal (no BPU, OOO buffers, pipes, bypass)
  - Cache coherency overhead ? Per-core L1 I&D caches (3 cycles) connected to shared L2 (MESI, 12-way assoc, 64B line) and memory (8 cycles) over 200 GB/s interconnect (“SMP on chip”, Dual Opteron)
  - Power consumption ? “Throttling” of threads/cores, lower temperature increases reliability due to lower wearout and drift
- What are they selling ? Many customers invested in Solaris, thread-rich (Java), modularity given per core (Oracle)/chip (uSoft) licensing

Throughput vs. Utilisation

- T1: 5-stage (RISC2) plus Thread Select stage, < 1 CPI -> 9.7 GOPS
- Superpipelining (PPC970) faces memory wall, Instruction Level Parallelism limits
- Large-scale parallel (Beowulf) faces power wall -> smaller clusters
- Superscalar (dynamic) cores limited by complexity required to find parallel instructions being ~proportional to square of number of instructions that can be issued simultaneously.
- Superscalar cores to improve resource usage with coarse/event (IBM Northstar), fine, or simultaneous multi-threading (Intel P4 hyper-threading, IBM POWER5): higher IPC; 2-3x increase in clock speed allows two dual or one quad core "fatter but fewer" CPUs to approach T1

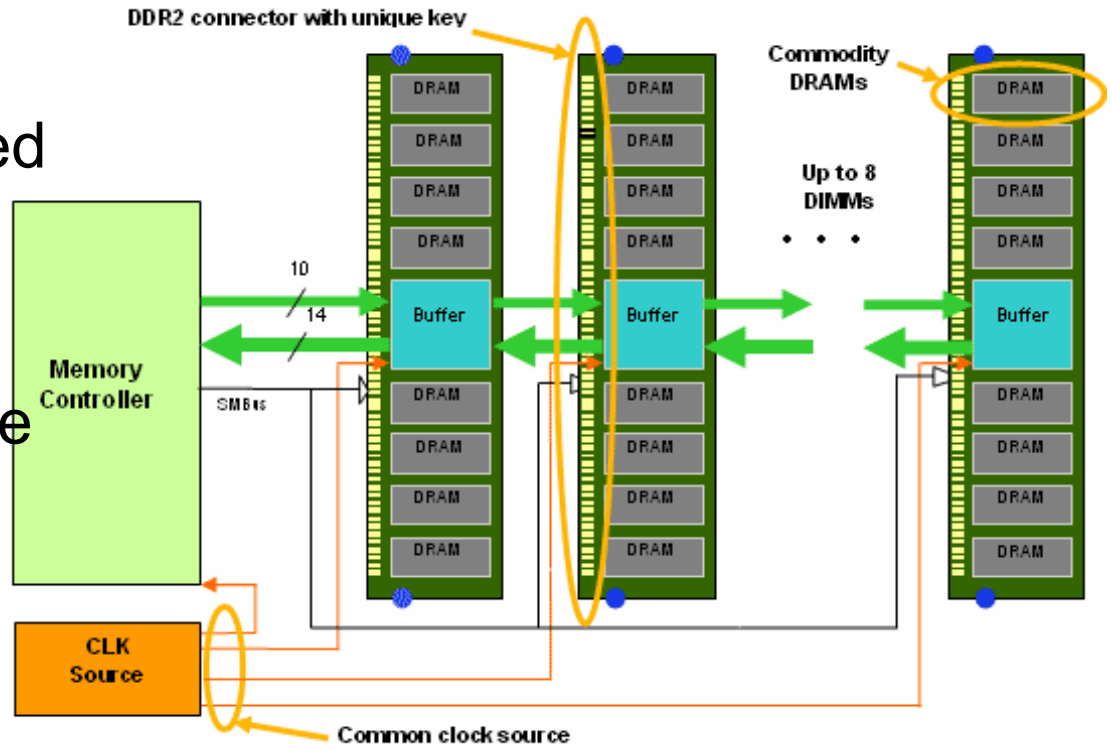
- “Weaknesses” of T1: shallow pipeline, clock speed, large SPARC-defined register window, no FPU

T2: Refining T1

- 90nm, 1.2 GHz T1 -> 65nm T2, 1.4+ GHz
- 4 threads/6-stage pipeline/core -> 8 threads/8 or 12-stage pipeline/2-execution unit (thread group) core
- Memory controllers (4), 400 MHz 128-bit memory bus, DDR2 SDRAM (8 cycles) -> FB-DIMMs
- JBUS -> 8x PCIExpress, two 10 Gigabit ports with 200KB packet classification and filtering (layers 2 thru 4)
- Per core cryptographic coprocessor with MAU and hash and cypher sub-engines
- L2 size increased to 4 MB (8-banks, 16-way associative)
- Shared FPU (<1% instr) -> per core FPU

Fully Buffered DIMM

- JEDEC Standard
- introduces an Advanced Memory Buffer (AMB) between the memory controller and the memory module. Unlike the parallel bus architecture of traditional DRAMs, a FB-DIMM has a serial interface between the memory controller and the AMB



Reliability, Availability, Serviceability

- Registers and L2 with SEC/DED, L1 I- and D-caches and TLBs with parity error correction
- Main memory : correct any error within single nibble, detect any uncorrectable errors within any 2 nibbles (Galois Field for higher bandwidth than Hamming)
- DRAM sparing (degraded protection till service)
- Memory scrubber (64B/line, programmable)
- Hardware retry on cache error

Why this can be interesting for us

- Results of FPGA mapping:

Preliminary results of FPGA mapping to Xilinx XC4VLX200 and Altera EP2S180 parts are listed below.

Xilinx XC4VLX200	SPARC	FPU	CCX
LUTs	134973	13863	25090
Utilization	75%	7%	14%

Altera EP2S180	SPARC	FPU	CCX
Estimated ALMs	74280	6855	19589
Utilization	103%	10%	27%

Some Areas to Consider

- Active, “local” mgmt of power, temp (reliability), soft errors
- Special function integration : networking, cryptography
- Use of SerDes for Ethernet, PCI-Express, FB-DIMM
- T1: “Software-defined”, not “instruction-defined”, architecture
  - Multi-threaded (Sybase), multi-process apps (Oracle, SAP, PeoplSoft)
  - “Bare bones” threaded (Java-based, e.g. JVM)
  - Multi-instance applications
  - Allocate entire core (w/ performance impact) to real-time threads
  - Design independent or cache-positive interactive threads
- Concurrency separates independent control flows, takes advantage of latency : *applications need to be concurrent*
- Improving usage and utilisation : addition of transactional hardware to CMPs (transactional memory soon commercially available) makes concurrent programming models easier

Learn from a code that works

- Learn from the real chip
- Explore the code that works.
- Learn techniques that have been tested to create low-power, highly productive chips.
- Examples of some of the research areas being explored with OpenSPARC technology include:
  - Performance analysis and modeling.
  - EDA development and optimization for Multicore chips.
  - SoftCores for Multicore FPGA implementations.
  - Optimizing and developing SW for multi-core systems.
  - EDA interests areas like simulation/acceleration, formal verification, place and route algorithm, testability, modeling in SystemVerilog, SystemC.
  - Developing multi-threaded EDA tools

What is going on while we discuss

- Simply RISC has shipped the S1 Core, a 64-bit Wishbone-compliant single CPU Core based upon the OpenSPARC T1
- The S1 Core is released under the same license of the T1, the GNU General Public License (GPL); the design is freely downloadable from the Simply RISC website @ www.srisc.com