Microarchitecture of the UltraSPARC-T1 CPU

Poonacha Kongetira
Director Hardware Engineering
Sun Microsystems, Inc.

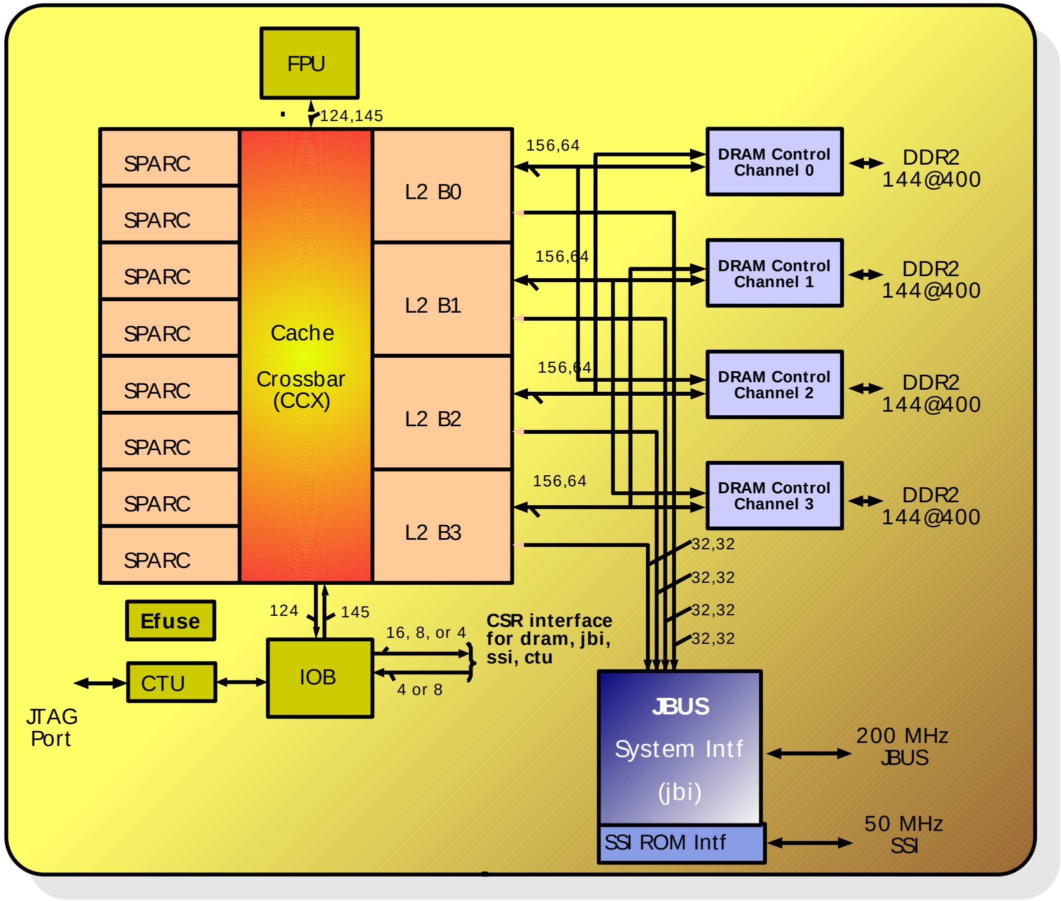
Agenda

- Introduction
- Threading and the Core pipeline
- Sparc Core Microarchitecture
- Memory Subsystem Brief
- Conclusions

Architectural Tradeoffs for Throughput

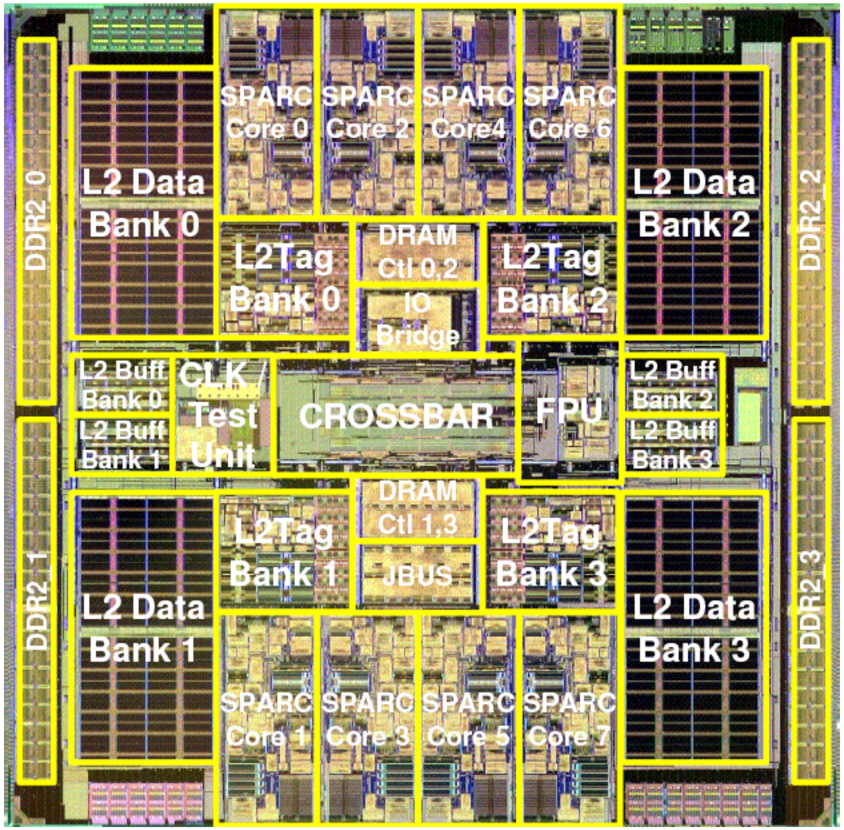
- Maximize number of Threads on die to exploit Thread Level Parallelism
  - Memory and Pipeline stall time hidden by overlapped execution of large number of threads
  - Shared L2 cache for efficient data sharing among cores
- Implement a high b/w memory system to feed the threads
  - High b/w interface to L2 cache for L1 misses
  - Banked and highly associative L2 cache
  - High bandwidth interface to DRAM
- Pick frequency optimized for Performance/Watt

UltraSPARC-T1



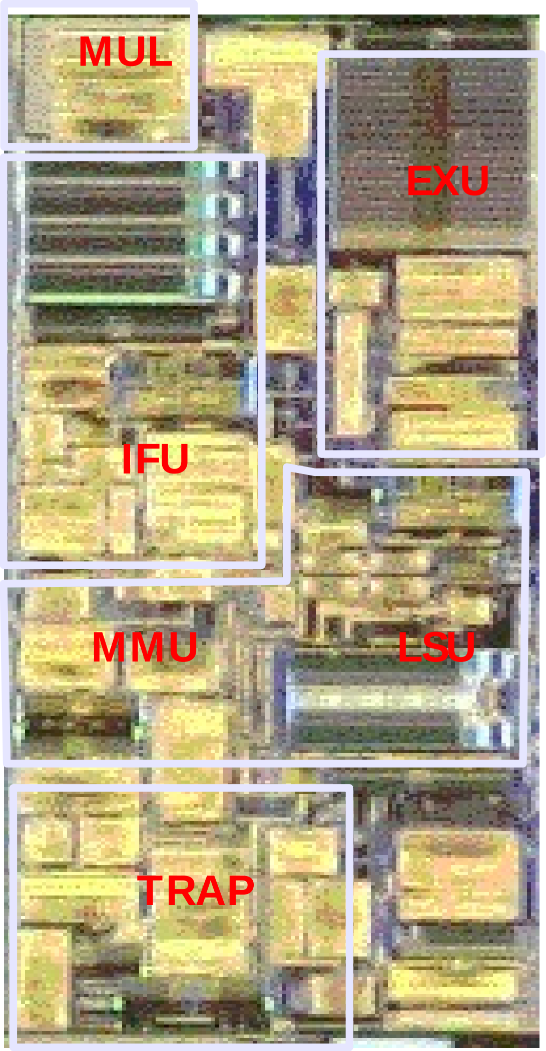
- 8 x 4-way Multithreaded cores for a total of 32 threads
- 134 GB/s crossbar interconnect for on chip communication
- 4 way banked, 12 way associative, 3MB L2
- 4 DDR2 channels (25GB/s)
- Sun Jbus interface to PCI-X/PCIe bridge chip
- Single FPU shared by all cores
- SPARC V9 ISA
- 1.2Ghz frequency

UltraSPARC-T1: Some Design Choices



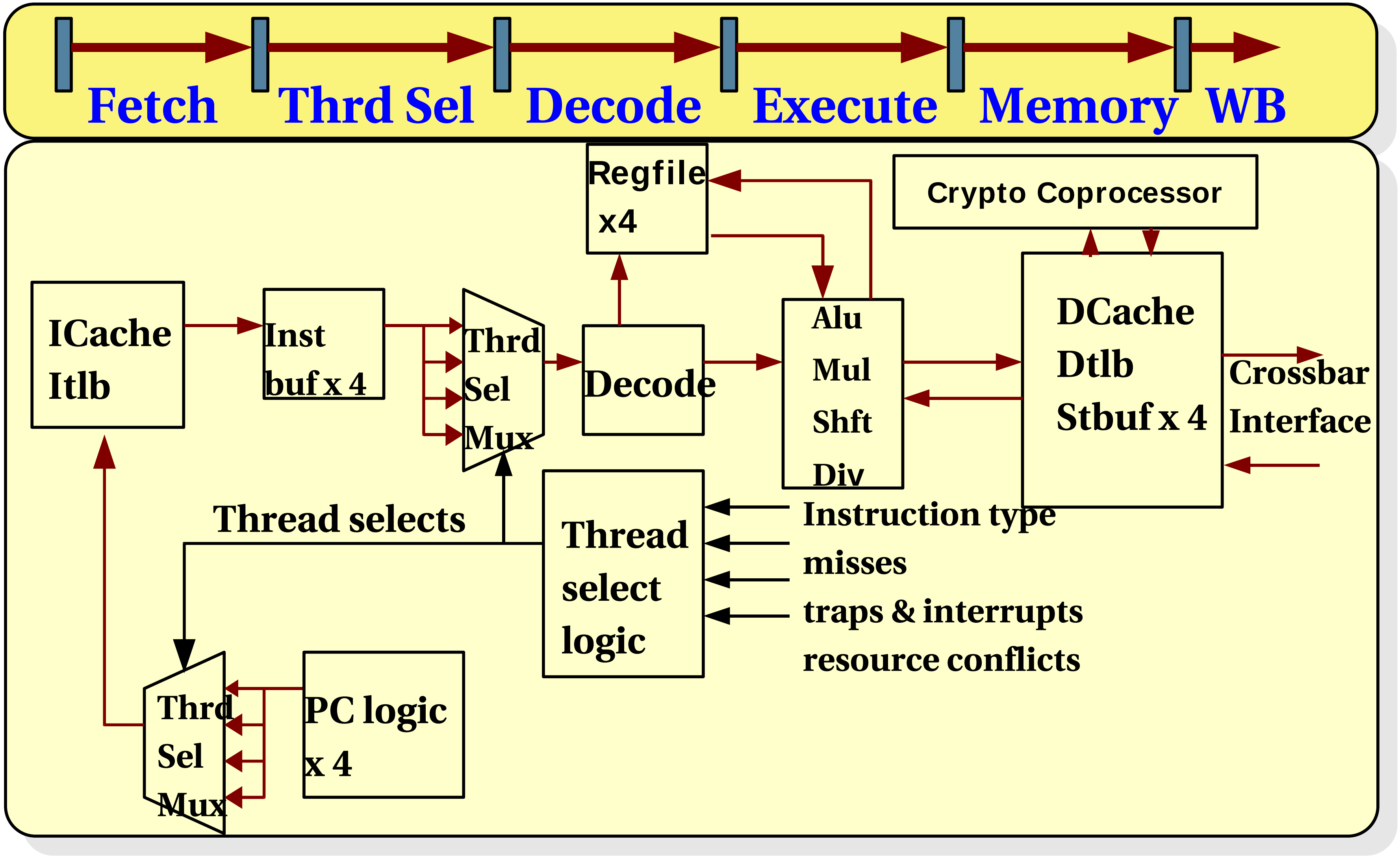
- Simpler core architecture to maximize cores on die
- Caches, dram channels shared across cores give better area utilization
- Shared L2 decreases cost of coherence misses by an order of magnitude
- On die memory controllers reduce miss latency
- Crossbar good for b/w, latency, functional verification
- 378mm2 die in 90nm dissipating ~70W

UltraSPARC-T1 Processor Core



- Four threads per core
- Single issue 6 stage pipeline
- 16KB I-Cache, 8KB D-Cache
  - Unique resources per thread
    - Registers
    - Portions of I-fetch datapath
    - Store and Miss buffers
  - Resources shared by 4 threads
    - Caches, TLBs, Execution Units
    - Pipeline registers and DP
- Core Area = 11mm2 in 90nm
- MT adds ~20% area to core

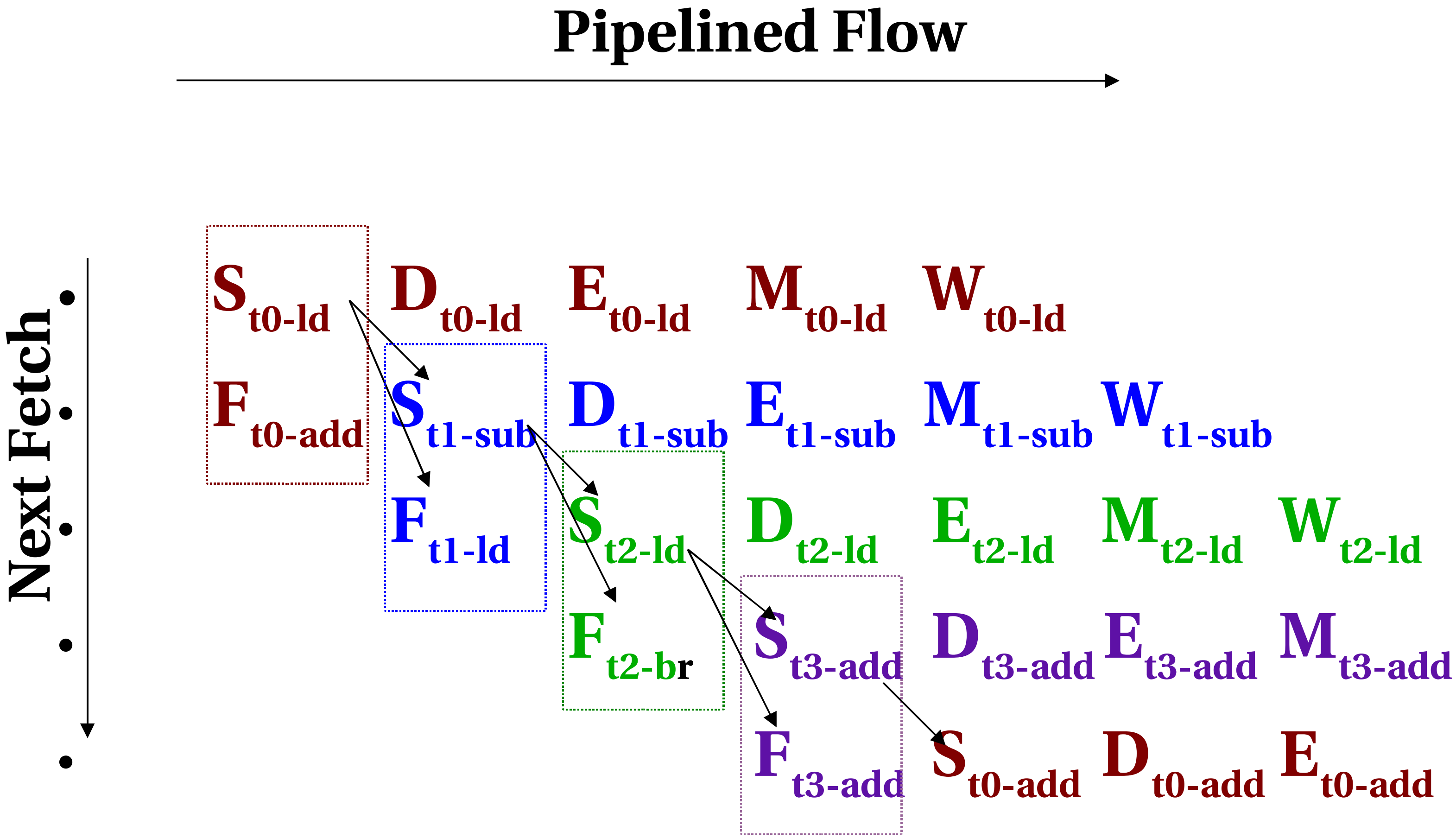
SPARC Core Pipeline



Thread Selection Policy

- Switch between available threads every cycle giving priority to least recently executed thread
- Threads become unavailable due to:
  - Long latency ops like loads, branch, mul, div.
  - Pipeline stalls such as cache misses, traps, and resource conflicts
- Loads are speculated as cache hits, and the thread is switched in with lower priority

Thread Selection – All Threads Ready



Pipelined Flow

Next Fetch

$S_{t0\text{-}ld}$ $D_{t0\text{-}ld}$ $E_{t0\text{-}ld}$ $M_{t0\text{-}ld}$ $W_{t0\text{-}ld}$

$F_{t0\text{-}add}$ $S_{t1\text{-}sub}$ $D_{t1\text{-}sub}$ $E_{t1\text{-}sub}$ $M_{t1\text{-}sub}$ $W_{t1\text{-}sub}$

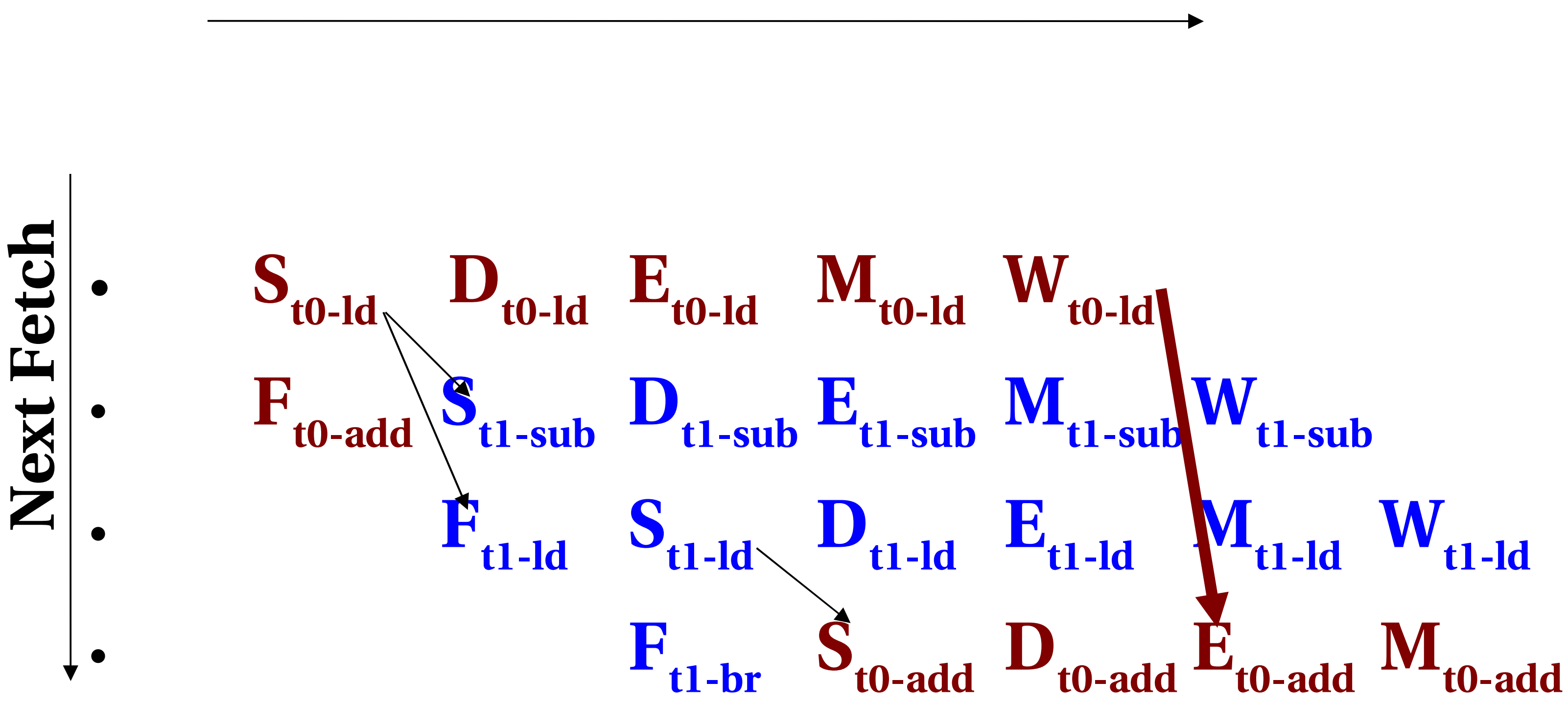
$F_{t1\text{-}ld}$ $S_{t2\text{-}ld}$ $D_{t2\text{-}ld}$ $E_{t2\text{-}ld}$ $M_{t2\text{-}ld}$ $W_{t2\text{-}ld}$

$F_{t2\text{-}br}$ $S_{t3\text{-}add}$ $D_{t3\text{-}add}$ $E_{t3\text{-}add}$ $M_{t3\text{-}add}$

$F_{t3\text{-}add}$ $S_{t0\text{-}add}$ $D_{t0\text{-}add}$ $E_{t0\text{-}add}$

Thread Selection – Two Threads Ready

Pipelined Flow



•	$S_{t0\text{-}ld}$	$D_{t0\text{-}ld}$	$E_{t0\text{-}ld}$	$M_{t0\text{-}ld}$	$W_{t0\text{-}ld}$		
•	$F_{t0\text{-}add}$	$S_{t1\text{-}sub}$	$D_{t1\text{-}sub}$	$E_{t1\text{-}sub}$	$M_{t1\text{-}sub}$	$W_{t1\text{-}sub}$	
•		$F_{t1\text{-}ld}$	$S_{t1\text{-}ld}$	$D_{t1\text{-}ld}$	$E_{t1\text{-}ld}$	$M_{t1\text{-}ld}$	$W_{t1\text{-}ld}$
•			$F_{t1\text{-}br}$	$S_{t0\text{-}add}$	$D_{t0\text{-}add}$	$E_{t0\text{-}add}$	$M_{t0\text{-}add}$

Thread '0' is speculatively switched in before cache hit information is available, in time for the 'load' to bypass data to the 'add'

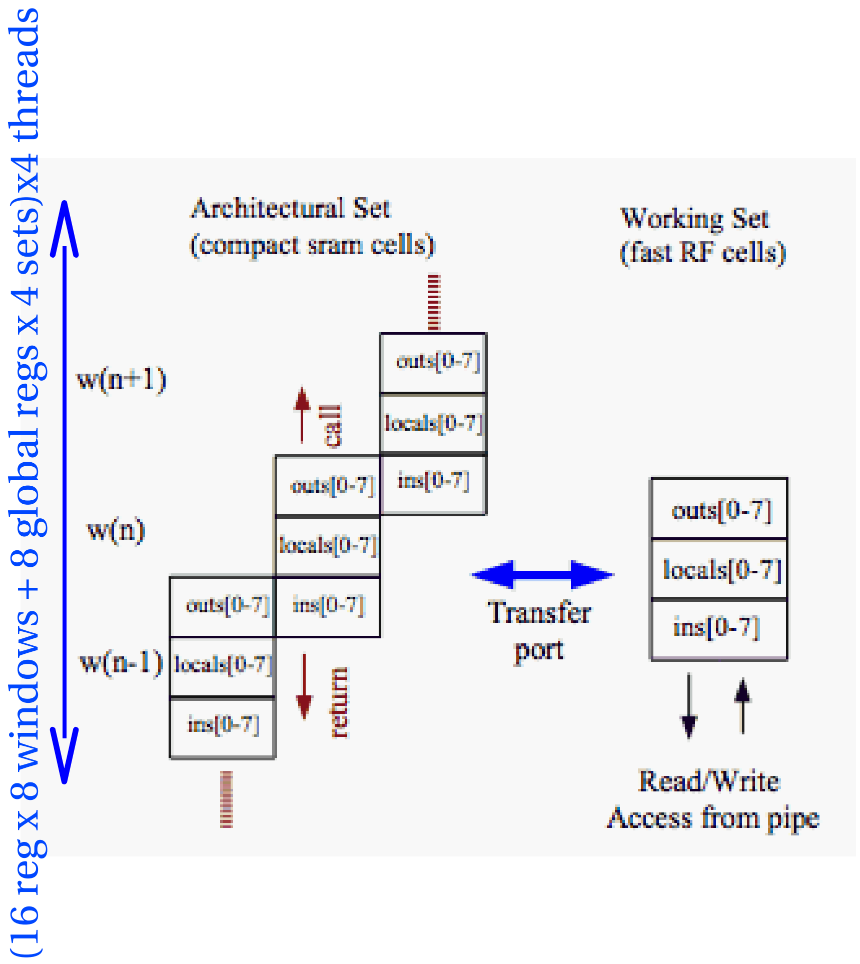
Instruction Fetch/Switch/Decode Unit(IFU)

- I-cache complex
  - 16KB data, 4ways, 32B line size
  - Single ported Instruction Tag.
  - Dual ported(1R/1W) Valid bit array to hold Cache line state of valid/invalid
  - Invalidates access Vbit array not Instruction Tag
  - Pseudo-random replacement
- Fully Associative Instruction TLB
  - 64 entries, Page sizes: 8k,64k, 4M, 256M
  - Pseudo LRU replacement.
  - Multiple hits in TLB prevented by doing autodemap on fill

IFU Functions (cont'd)

- 2 instructions fetched each cycle, though only one is issued/clk. Reduces I$ activity and allows opportunistic line fill.
- 1 outstanding miss/thread, and 4 per core. Duplicate misses do not request to L2
- PC's, NPC's for all live instructions in machine maintained in IFU

Windowed Integer Register File



- 5kB 3R/2W/1T structure
  - 640 64b regs with ECC!
- Only 32 registers from current window visible to thread.
- Window changing in background under thread switch. Other threads continue to access IRF
- Compact design with 6T cells for architectural set & multi ported cell for working set.
- Single cycle R/W access

Execution Units

- Single ALU and Shifter. ALU reused for Branch Address and Virtual Address Calculation
- Integer Multiplier
  - > 5 clock latency, throughput of ½ per cycle for area saving
  - > Contains accumulate function for Mod Arithmetic.
  - > 1 integer mul allowed outstanding per core.
  - > Multiplier shared between Core Pipe and Modular Arithmetic unit on a round robin basis.
- Simple non restoring divider, with one divide outstanding per core.
- Thread issuing a MUL/DIV will rollback and switch out if another thread is occupying the mul/div units.

Load Store Unit(LSU)

- D-Cache complex
  - 8KB data, 4ways, 16B line size
  - Single ported Data Tag.
  - Dual ported(1R/1W) Valid bit array to hold Cache line state of valid/invalid
  - Invalidates access Vbit array but not Data Tag
  - Pseudo-random replacement
  - Loads are allocating, stores are non allocating.
- DTLB: common macro to ITLB(64 entry FA)
- 8 entry store buffer per thread, unified into single 32 entry array, with RAW bypassing.

LSU(cont'd)

- Single load per thread outstanding. Duplicate request for the same line not sent to L2
- Crossbar interface
  - > LSU prioritizes requests to the crossbar for FPops, Streaming ops, I and D misses, stores and interrupts etc.
  - > Request priority: imiss>ldmiss>stores,{fpu,strm,interrupt}.
  - > Packet assembly for pcx.
- Handles returns from crossbar and maintains order for cache updates and invalidates.

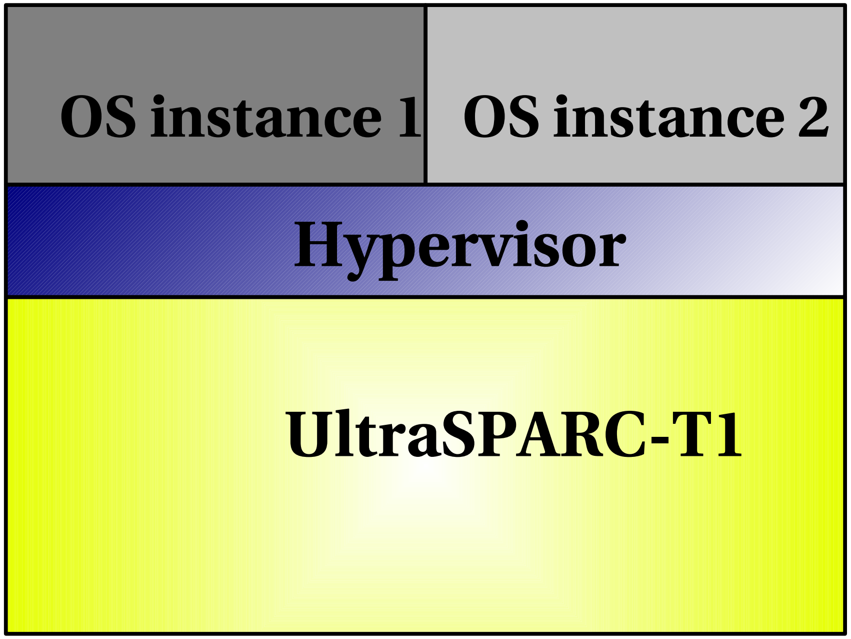
Asynchronous Crypto Co-processor

- One crypto unit per core
  - Supports asymmetric crypto(public key RSA) for upto 2048b size key. Shares integer Multiplier for modular arithmetic operations
  - One thread can use unit at a time
  - Operation set up by store to control register, and thread returns to normal processing
  - Crypto unit initiates streaming load/store to L2 through the crossbar, compute ops to Multiplier
  - Completion by polling or interrupt

Other Functions

- Support for 6 trap levels. Traps cause pipeline flush and thread switch until trap PC is available
- Support for upto 64 pending interrupts per thread
- Floating Point
  - FP registers and decode located within core
  - On detecting an Fpop
    - The thread switches out
    - Fpop is further decoded and FRF is read
    - Fpop with operands are packetized and shipped over the crossbar to the FPU
    - Computation done in FPU and result returned via crossbar
    - Writeback completed to FRF and thread restart

Virtualisation

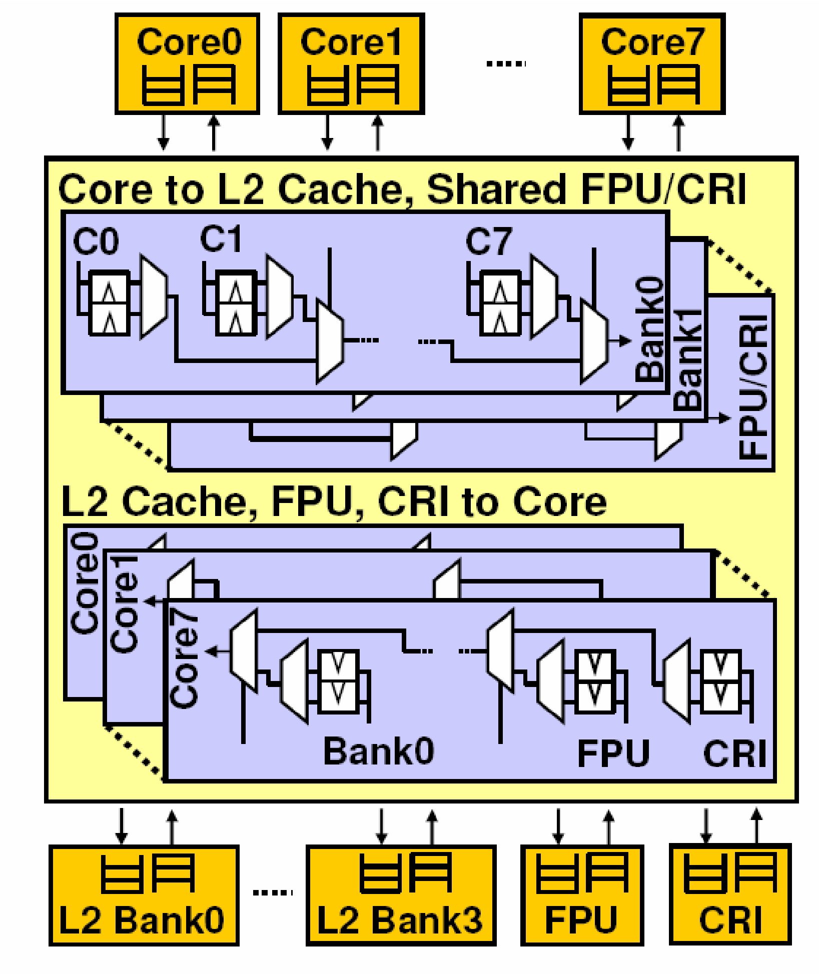


- Hypervisor layer virtualizes CPU
- Multiple OS instances
- Better RAS as failures in one domain do not affect other domain
- Improved OS portability to newer hardware

Virtualisation on UltraSPARC-T1

- Implementation on UltraSPARC-T1
  - Hypervisor uses Physical Addresses
  - Supervisor sees 'Real Addresses' – a PA abstraction
  - VA translated to 'RA' and then PA. Niagara MMU and TLB provides h/w support.
  - Upto 8 partitions can be supported. 3Bit partion ID is part of TLB translation checks
  - Additional trap level added for hypervisor use

Crossbar



- Each requestor queues upto 2 packets per destination.
- 3 stage pipeline: Request, Arbitrate and Transmit
- Centralised arbitration with oldest requestor getting priority
- Core to cache bus optimized for address + doubleword store
- Cache to core bus optimized for 16B line fill. 32B I$ line fill delivered in 2 back to back clks

L2 Cache

- 3MB, 4-way banked, 12way SA, Writeback
- 64B line size, 64B interleaved between banks
- Pipeline latency: 8 clks for Load, 9 clks for I-miss, with critical chunk returned first
- 16 outstanding misses per bank -> 64 total
- Coherence maintained by shadowing L1 tags in an L2 directory structure.
- L2 is point of global visibility. DMA from IO is serialised wrt traffic from cores in L2

L2 Cache – Directory

- Directory shadows L1 tags
  - L1 set index and L2 bank interleaving is such that ¼ of L1 entries come from an L2 bank
  - On an L1 miss, the L1 replacement way and set index identify the physical location of the tag which will be updated by miss address
  - On a store, directory will be cammed.
    - Directory entries collated by set so only 64 entries need to be cammed. Scheme is quite power efficient
    - Invalidates are a pointer to the physical location in the L1, eliminating the need for a tag lookup in L1

Coherence/Ordering

- Loads update directory & fill the L1 on return
- Stores are non allocating in L1
  - Two flavors of stores: TSO, RMO. One TSO store outstanding to L2 per thread to preserve store ordering. No such limitation on RMO stores
  - No tag check done at store buffer insert
  - Stores check directory and determine L1 hit.
  - Directory sends store ack/inv to core
  - Store update happens to D$ on store ack
- Crossbar orders responses across cache banks

On Chip Mem Controller

- 4 independent DDRII DRAM channels
- Can supports memory size of upto 128GB
- 25GB/s peak bandwidth
- Schedules across 8 rds + 8 writes
- Can be programmed to 2 channel mode in reduced configuration
- 128+16b interface, chipkill support, nibble error correction, byte error detection
- Designed to work from 125-200Mhz

Conclusion

- Microarchitecture choices for UltraSPARC-T1 guided by a focus on throughput performance for commercial server workloads
  - Simple threaded cores to maximize number of threads
  - Shared memory subsystem to deliver sufficient bandwidth
  - Focus on Performance/Watt to address power concerns in datacentre installations

Microarchitecture of the UltraSPARC -T1 CPU

Poonacha Kongetira
Director Hardware Engineering
Sun Microsystems Inc