

Niagara-2: A Highly Threaded Server-on-a-Chip

Greg Grohoski
Distinguished Engineer
Sun Microsystems

Authors

- Jama Barreh
- Jeff Brooks
- Robert Golla
- Greg Grohoski
- Rick Hetherington
- Paul Jordan
- Mark Luttrell
- Chris Olson
- Manish Shah

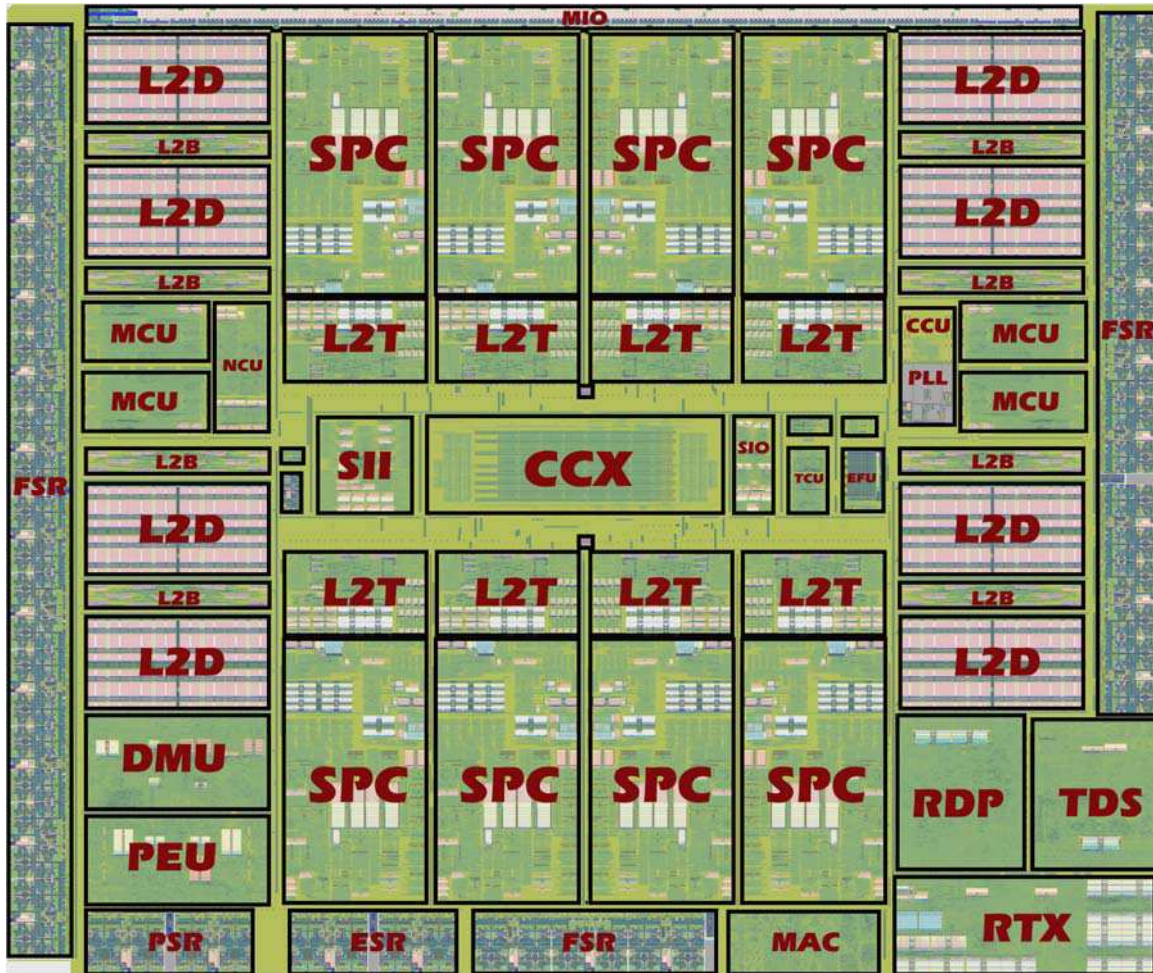
Agenda

- Chip overview
- Sparc core
 - > Execution Units
 - > Power
 - > RAS
- Summary

Niagara-2 Chip Goals

- Double throughput versus UltraSparc T1
 - > Maintain Sparc binary compatibility
 - > <http://opensparc.sunsource.net/nonav/index.html>
- Improve throughput / watt
- Improve single-thread performance
- Integrate important SOC components
 - > Networking
 - > Cryptography

Niagara-2 Chip Overview



- 8 Sparc cores, 8 threads each
- Shared 4MB L2, 8-banks, 16-way associative
- Four dual-channel FBDIMM memory controllers
- Two 10/1 Gb Enet ports w/onboard packet classification and filtering
- One PCI-E x8 1.0 port
- 711 signal I/O, 1831 total

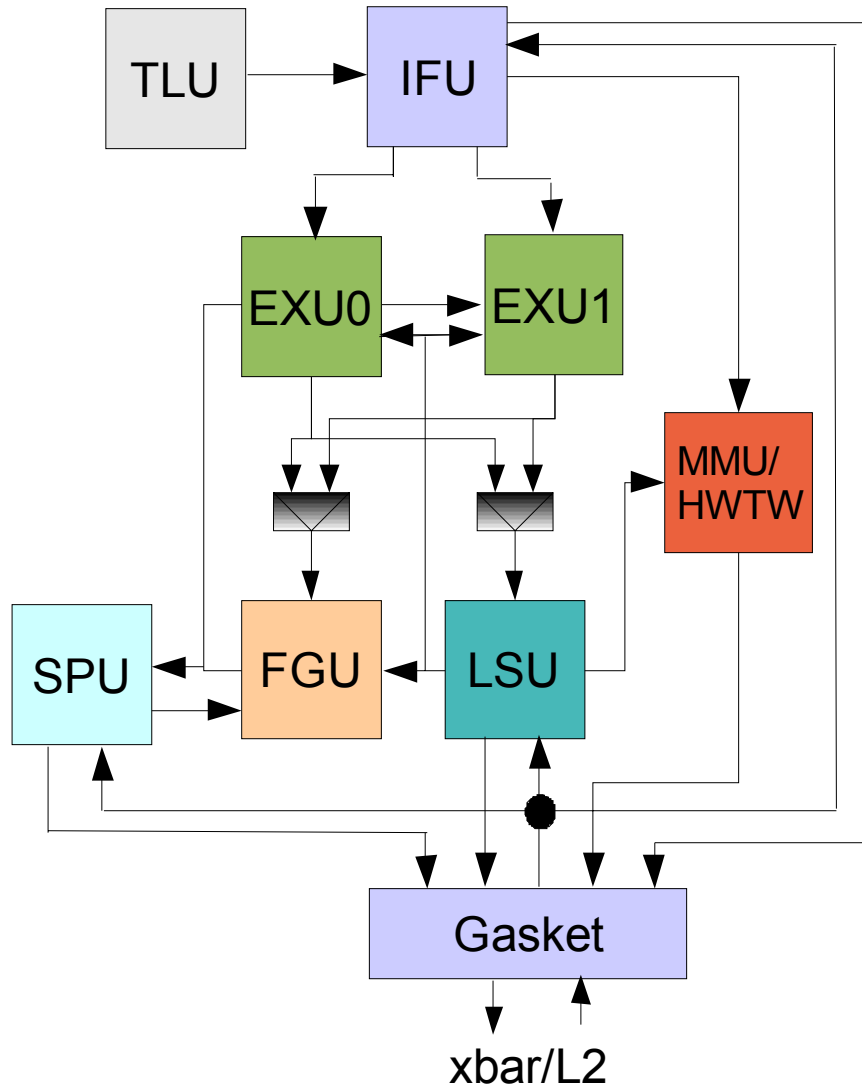
Glossary

- CCX – Crossbar
- CCU – Clock control
- DMU/PEU – PCI Express
- EFU – Efuse (redundancy)
- ESR – Ethernet SERDES
- FSR – FBDIMM SERDES
- L2B – L2 write-back buffers
- L2D – L2 Data
- L2T – L2 tags
- MCU – Memory controller
- MIO – Miscellaneous I/O
- PSR – PCI-Express SERDES
- RDP/TDS/RTX/MAC – Ethernet
- SII/SIO – I/O datapath in/out to memory
- SPC – Sparc core
- TCU – Test control unit

Sparc Core Goals

- >2x throughput of UltraSparc T1
- Improve integer, floating-point performance
- Extend cryptographic support
 - > Support relevant ciphers, hashes
 - > Enable “free” encryption
- Optimum throughput/area and throughput/watt
 - > Doubling cores vs. increasing threads/core
 - > Utilization of execution units

Sparc Core Block Diagram

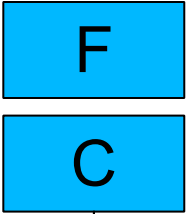


- IFU – Instruction Fetch Unit
 - > 16 KB I\$, 32B lines, 8-way SA
 - > 64-entry fully-associative ITLB
- EXU0/1 – Integer Execution Units
 - > 4 threads share each unit
 - > Executes one integer instruction/cycle
- LSU – Load/Store Unit
 - > 8KB D\$, 16B lines, 4-way SA
 - > 128-entry fully-associative DTLB
- FGU – Floating/Graphics Unit
- SPU – Stream Processing Unit
 - > Cryptographic acceleration
- TLU – Trap Logic Unit
 - > Updates machine state, handles exceptions and interrupts
- MMU – Memory Management Unit
 - > Hardware tablewalk (HWTW)
 - > 8KB, 64KB, 4MB, 256MB pages

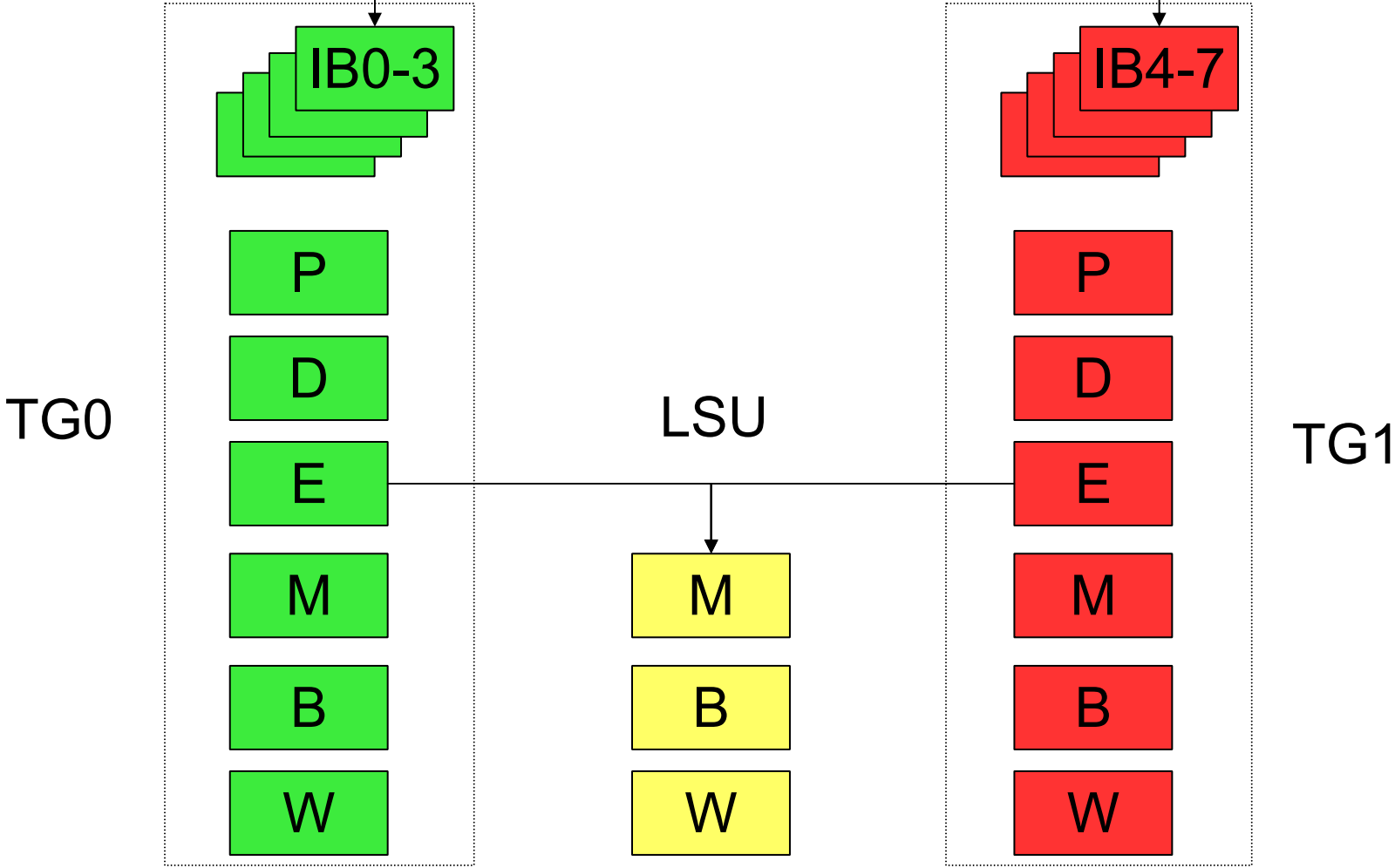
Core Pipeline

- 8 stages for integer operations:
 - > Fetch, Cache, Pick, Decode, Execute, Memory, Bypass, Writeback
 - > 3-cycle load-use
 - > Memory (translation, tag/data access)
 - > Bypass (late select, formatting)
- 12 stages for floating-point:
 - > Fetch, Cache, Pick, Decode, Execute, FX1, FX2, FX3, FX4, FX5, FB, FW
 - > 6-cycle latency for dependent FP ops
 - > Longer pipeline for divide/sqrt

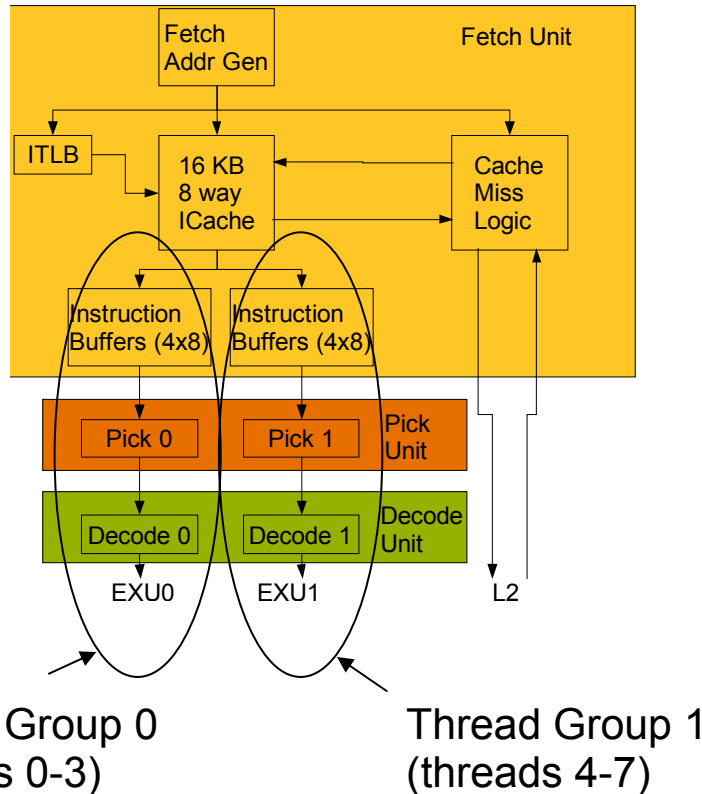
Integer/LSU Pipeline



IFU

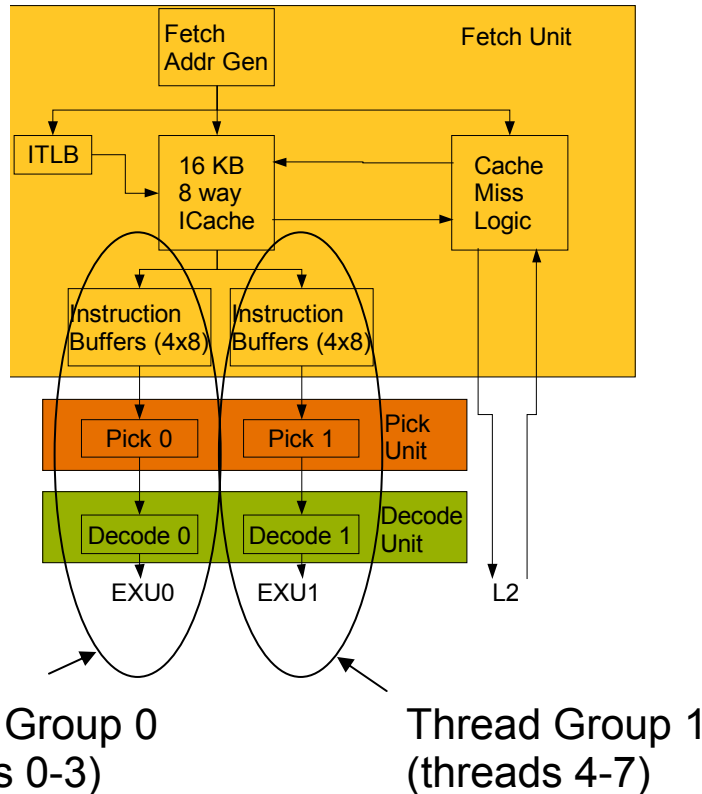


IFU Block Diagram



- Instruction cache, fetch/pick/decode logic for 8 threads
- Fetch up to 4 instructions from I\$
 - > Threads either in ready or wait state
 - > Wait states: TLB miss, cache miss, instruction buffer full
 - > Least-recently fetched among ready threads
 - > One instruction buffer/thread
- No branch prediction
 - > Predict not-taken, 5-cycle penalty
- Limited I\$ miss prefetching

IFU Block Diagram

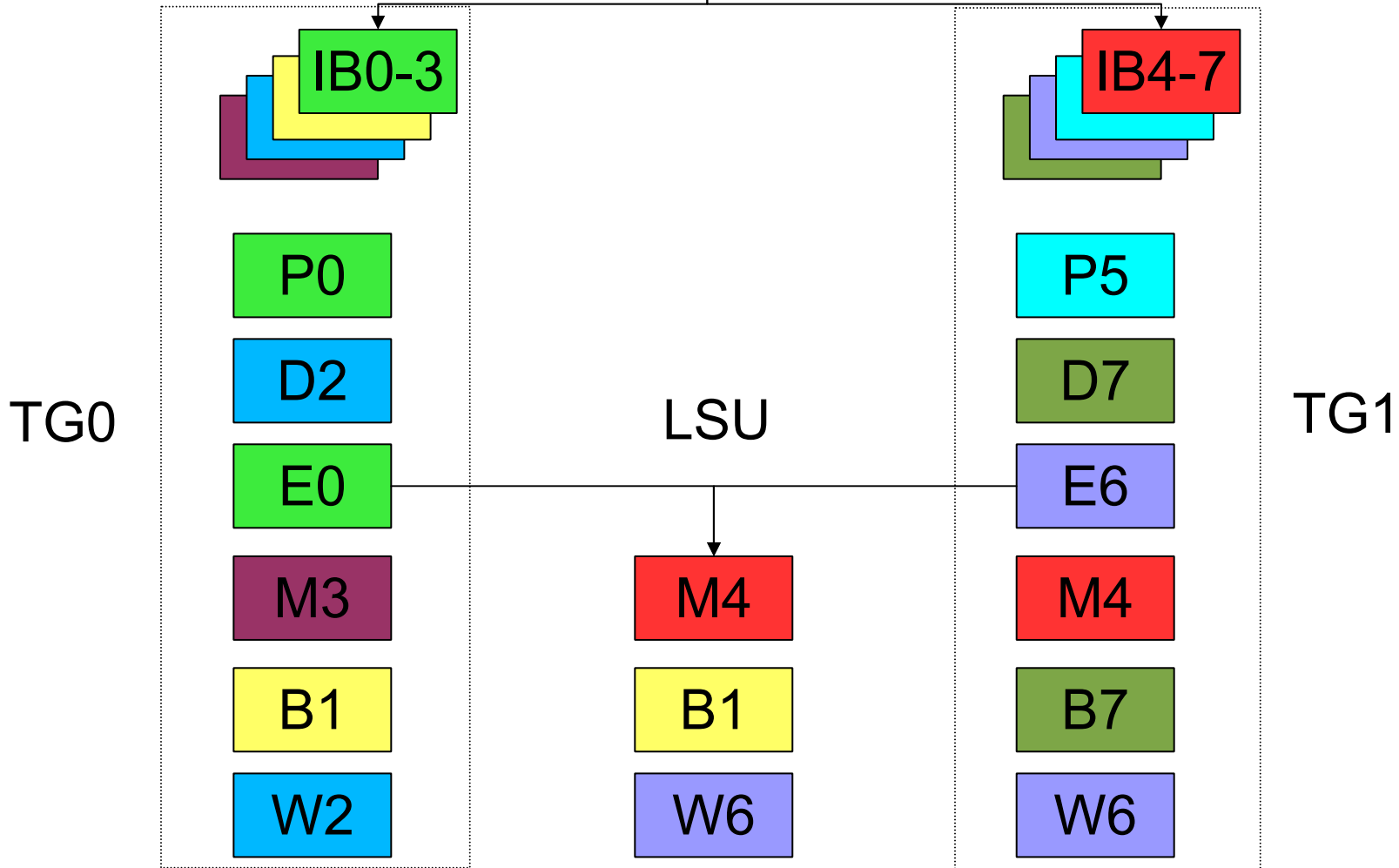


- Fetch decoupled from Pick
- Threads divided into 2 groups of 4 threads each
- One instruction from each thread group picked each cycle
 - > Least-recently picked within a thread group among ready threads
 - > Wait states: dependency, D\$ miss, DTLB miss, divide/sqrt, ...
 - > Gives priority to non-speculative threads (e.g. non-load hit)
- Decode resolves conflicts
 - > Each thread group picks independently of the other
 - > Both thread groups pick load/store or FGU instructions

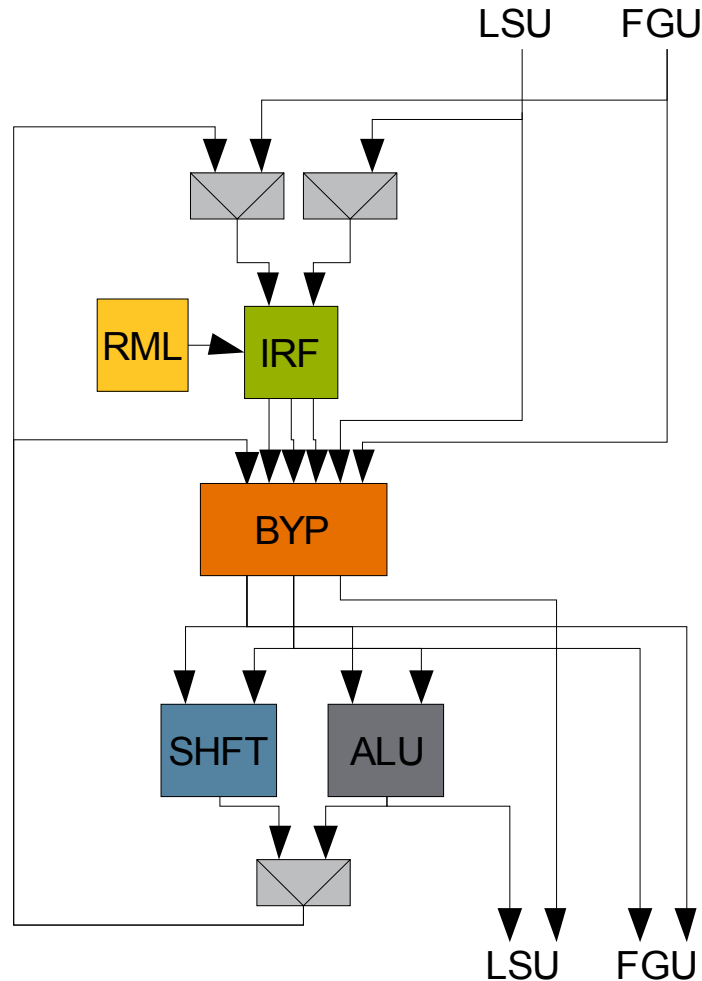
Threaded Execution



IFU

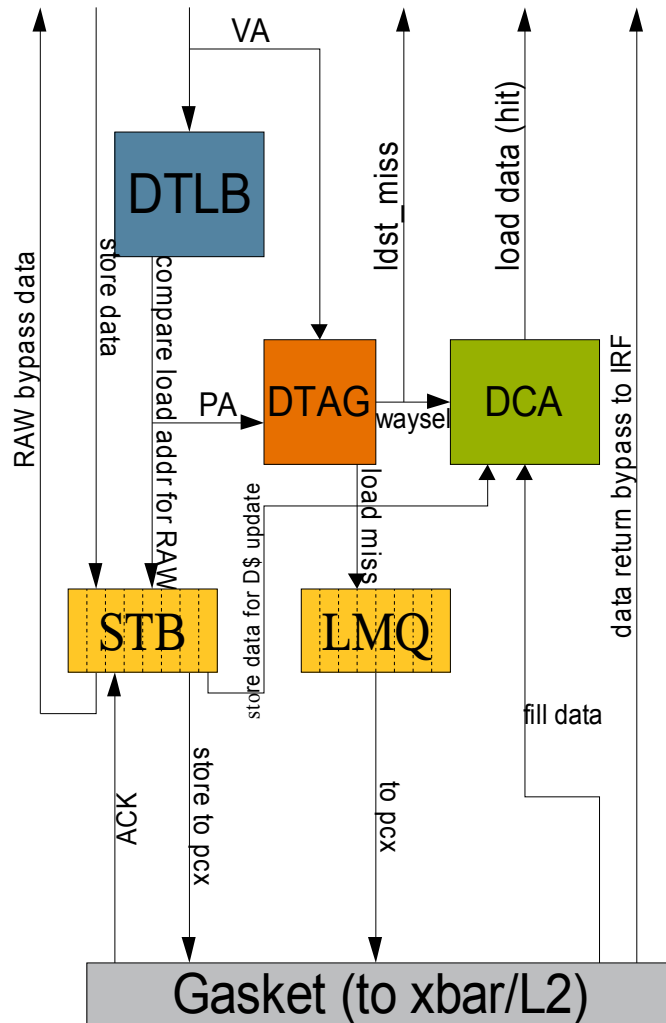


EXU



- Executes integer operations
 - > Some graphics operations
- Each EXU contains state for 4 threads
 - > Integer register file (IRF) contains 8 register windows per thread
 - > Window management logic (RML)
 - > Adder, shifter
- Generates addresses for load/store operations

LSU

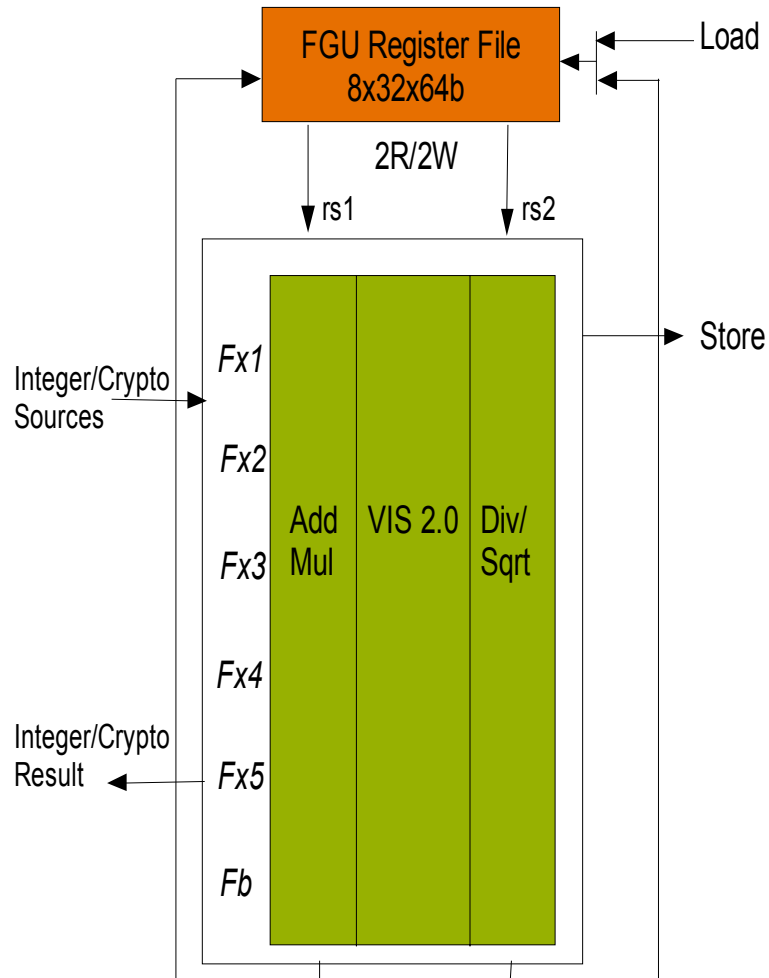


- One load or store per cycle
 - > D\$ is store-through
 - > D\$ fills in parallel with stores
- Load Miss Queue (LMQ)
 - > One pending load miss per thread
 - > D\$ allocates on load misses, updates on store hits
- Store buffer (STB) contains 8 stores/thread
 - > Stores to same L2 cache line are pipelined to L2
- Arbitrates between load misses, stores for crossbar
 - > Fairness algorithm

MMU

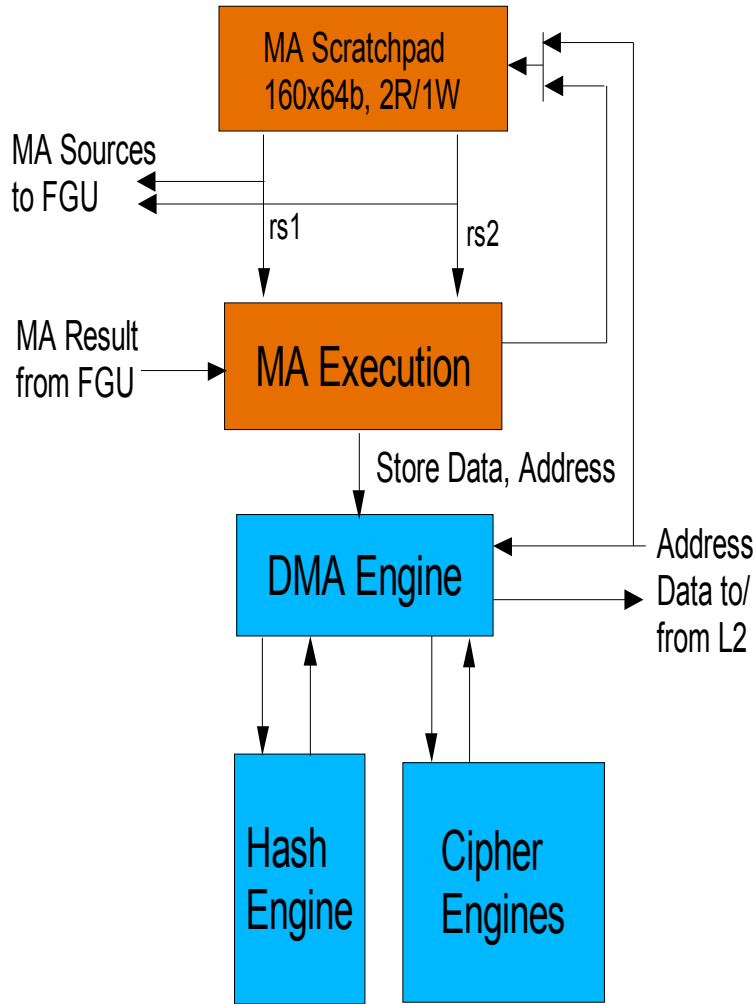
- Hardware tablewalk of up to 4 page tables
- Each page table supports one page size
- Three search modes:
 - > Sequential – search page tables in order
 - > Burst – search page tables in parallel
 - > Prediction – predict page table to search based upon VA
 - > Two-bit predictor for ordering first two page table searches
- Up to 8 pending misses
 - > ITLB or DTLB miss per thread

FGU



- Fully-pipelined (except divide/sqrt)
 - > Divide/sqrt in parallel with add or multiply operations of other threads
- FGU performs integer multiply, divide, population count
- Multiplier enhancements for modular arithmetic operations
 - > Built-in accumulator
 - > XOR multiply

SPU



- Cryptographic coprocessor
 - > Runs in parallel w/core at same frequency
- Two independent sub-units
 - > Modular Arithmetic
 - > RSA, binary and integer polynomial elliptic curve (ECC)
 - > Shares FGU multiplier
 - > Ciphers / Hashes
 - > RC4, DES/3DES, AES-128/192/256
 - > MD5, SHA-1, SHA-256
 - > Designed to achieve wire-speed on both 10Gb Ethernet ports
- DMA engine shares crossbar port w/core

Core Power Management

- Minimal speculation
 - > Next sequential I\$ line prefetch
 - > Predict branches not-taken
 - > Predict loads hit in D\$
 - > Hardware tablewalk search control
- Extensive clock gating
 - > Datapath
 - > Control blocks
 - > Arrays
- External power throttling
 - > Add wait states at decode stage

Core Reliability and Serviceability

- Extensive RAS features
 - > Parity-protection on I\$, D\$ tags and data, ITLB, DTLB CAM and data, modular arithmetic memory, store buffer address
 - > ECC on integer RF, floating-point RF, store buffer data, trap stack, other internal arrays
- Combination of hardware and software correction flows
 - > Hardware re-fetch for I\$, D\$
 - > Software recovery for other errors
 - > Offline a thread, group of threads, or physical core if error rate too high

Summary

- Niagara-2 combines all major server functions on one chip
- >2x throughput and throughput/watt vs. UltraSparc T1
- Greatly improved floating-point performance
- Significantly improved integer performance
- Embedded wire-speed cryptographic acceleration
- Enables new generation of power-efficient, fully-secure datacenters

Thank you ...

gregory.grohoski@sun.com