Module 18: "TLP on Chip: HT/SMT and CMP"

Lecture 40: "Case Studies: IBM Power4 and IBM Power5"

TLP on Chip: HT/SMT and CMP

- IBM POWER4
- 4-chip 8-way NUMA
- 32-way: ring bus
- POWER4 core
- POWER4 pipeline
- POWER4 caches
- POWER4 L2 cache
- POWER4 L3 cache
- POWER4 die photo
- IBM POWER5
- POWER5 die photo

Previous Next

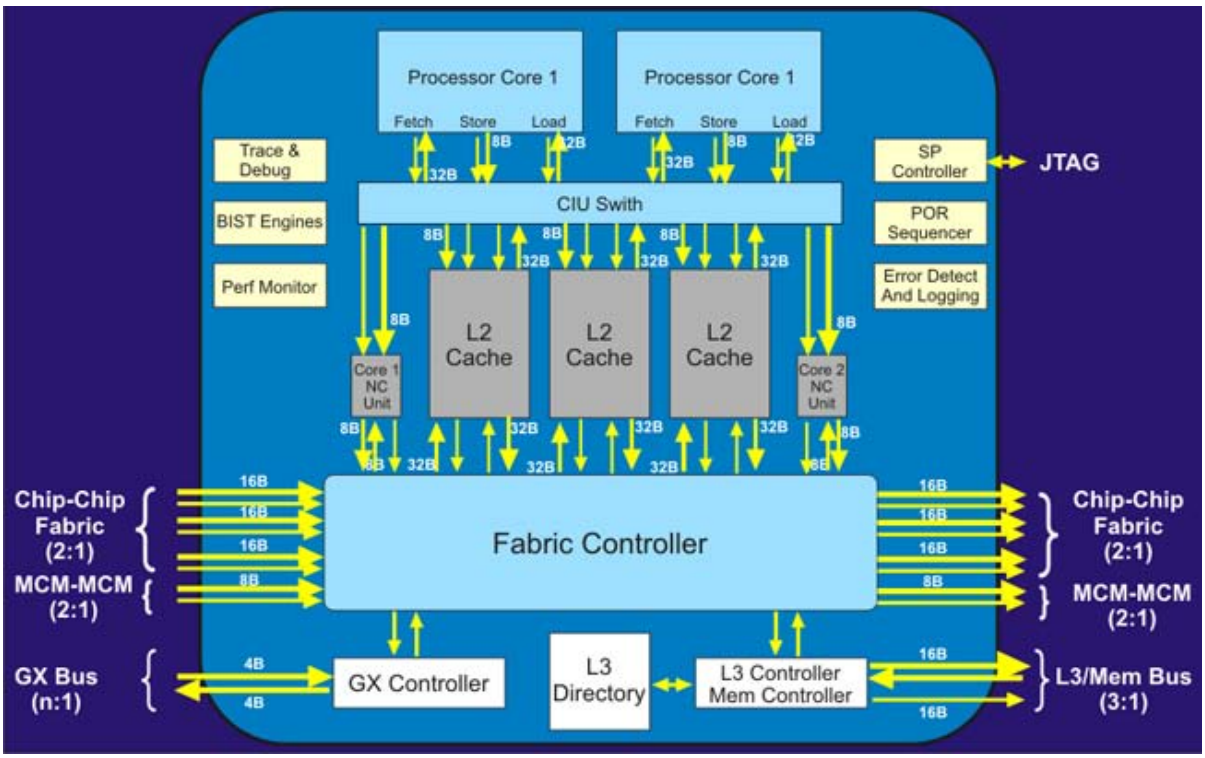
Module 18: "TLP on Chip: HT/SMT and CMP"

Lecture 40: "Case Studies: IBM Power4 and IBM Power5"

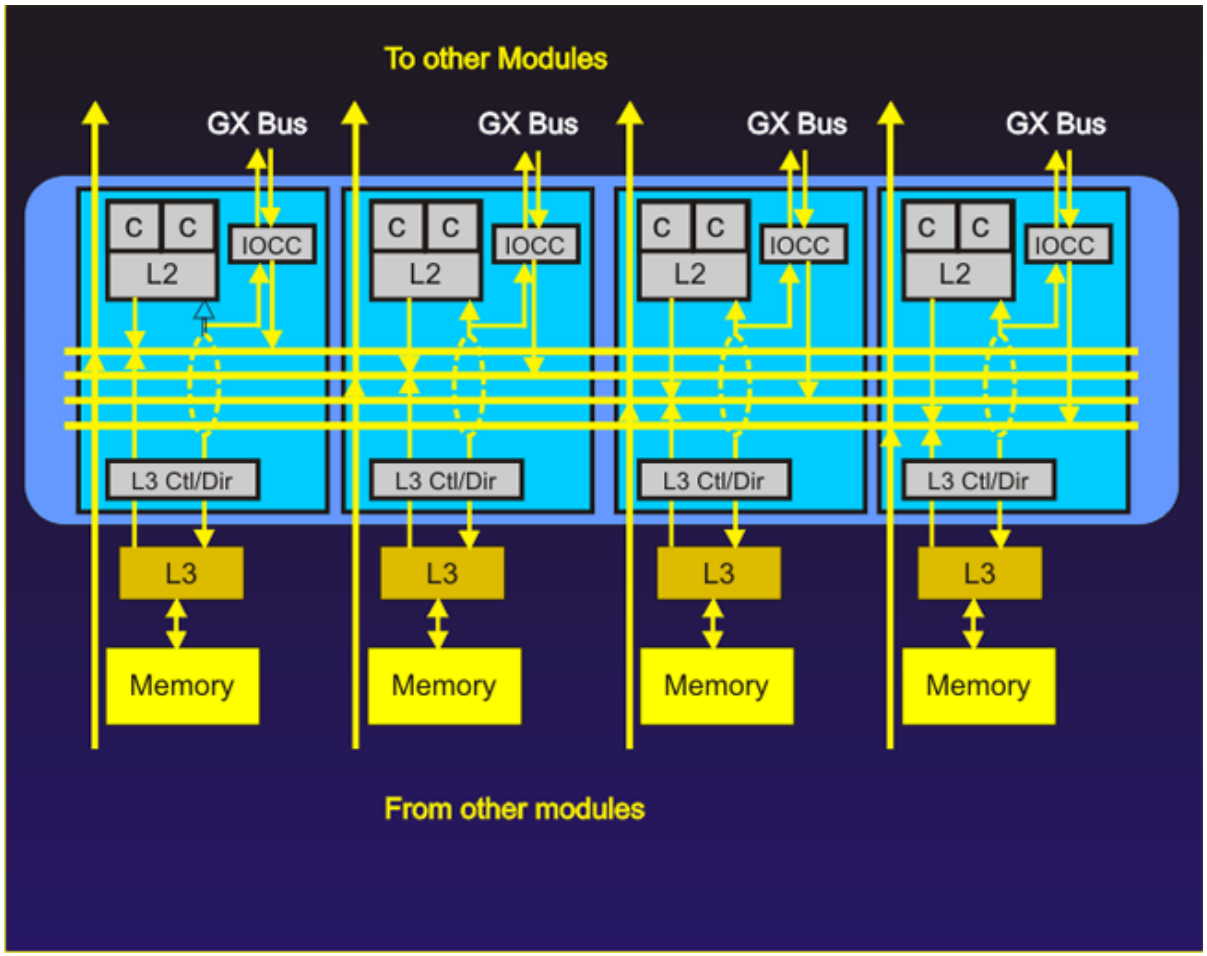
IBM POWER4

IBM POWER4

- Dual-core chip multiprocessor



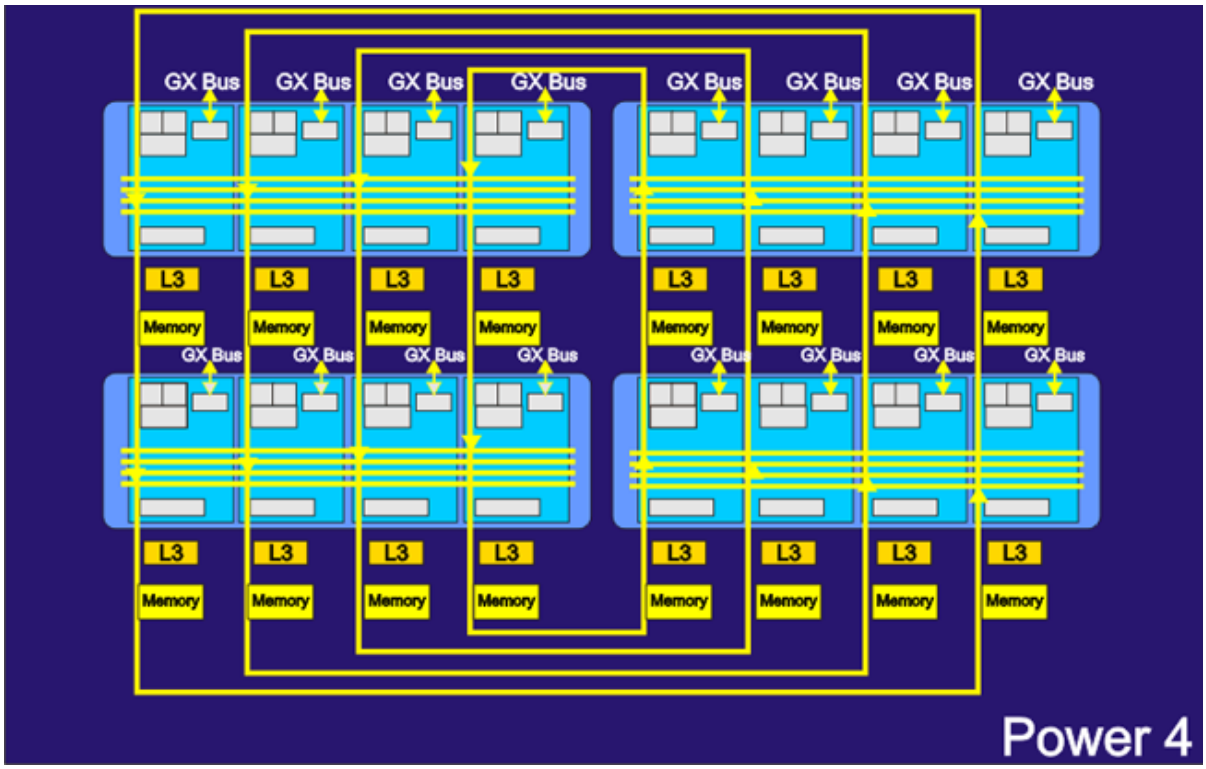
4-chip 8-way NUMA



Parallel application case studies

- Steps in writing a parallel program
- Example

32-way: ring bus

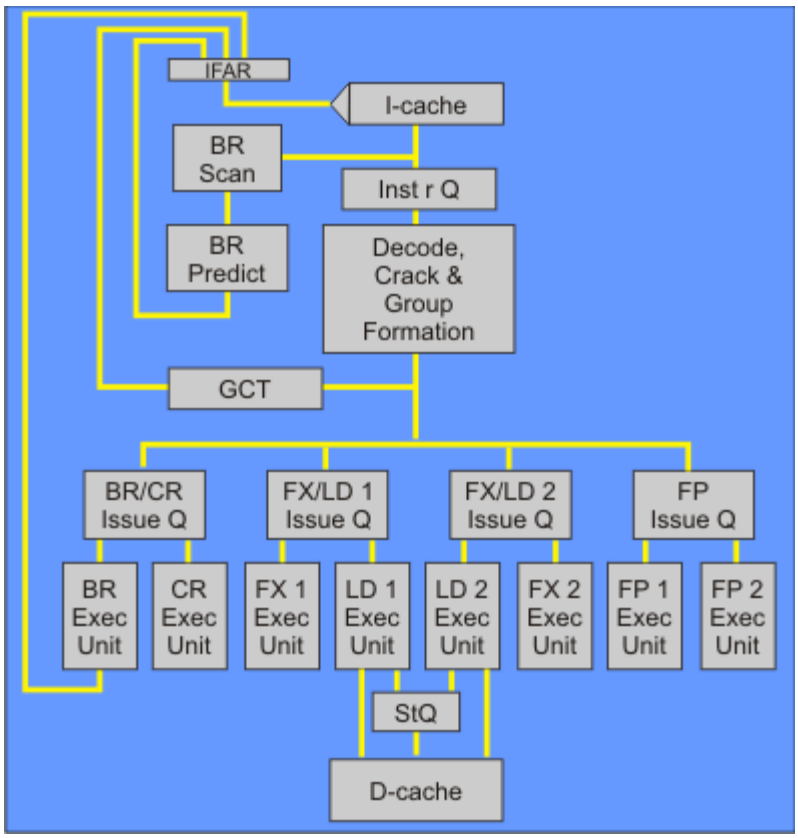


Previous Next

Module 18: "TLP on Chip: HT/SMT and CMP"
Lecture 40: "Case Studies: IBM Power4 and IBM Power5"

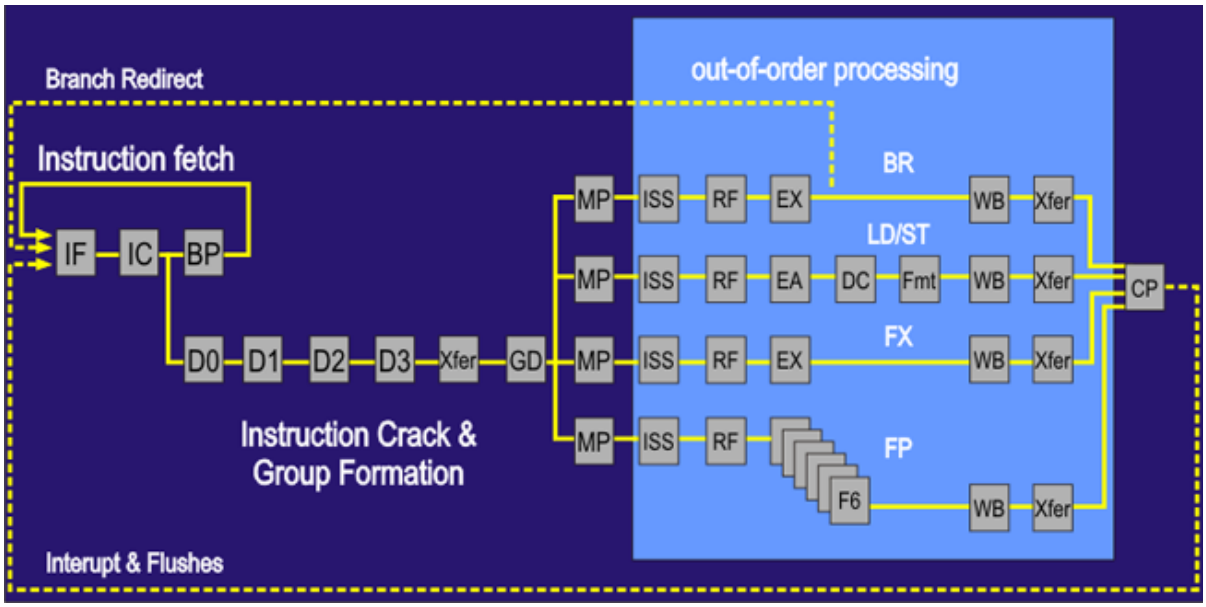
POWER4 core

- 8-wide fetch, 8-wide issue, 5-wide commit
  - Features out-of-order issue with renaming and branch prediction (bimodal+gshare hybrid)
  - Allows 20 groups of at most 5 instructions each to be in-flight beyond dispatch (100 instructions)



POWER4 pipeline

- Relatively short pipe
  - Clocked at more than 1 GHz for 0.18µm technology
  - Minimum 15 cycles for integer instructions
  - Minimum 12-cycle branch misprediction penalty
  - 11 small parallel issue queues (divided into four groups) for fast selection
  - Back-to-back issue of dependent instructions not allowed: slow bypass or bypass absent? Requires at least one cycle gap
  - Out-of-order load issue, load-load and load-store replay; load-load replay optimized with load queue snoop bit
  - Write through write no allocate private L1 data cache; at most 8 outstanding L1 load misses
  - Inclusion maintained between L2 and L1

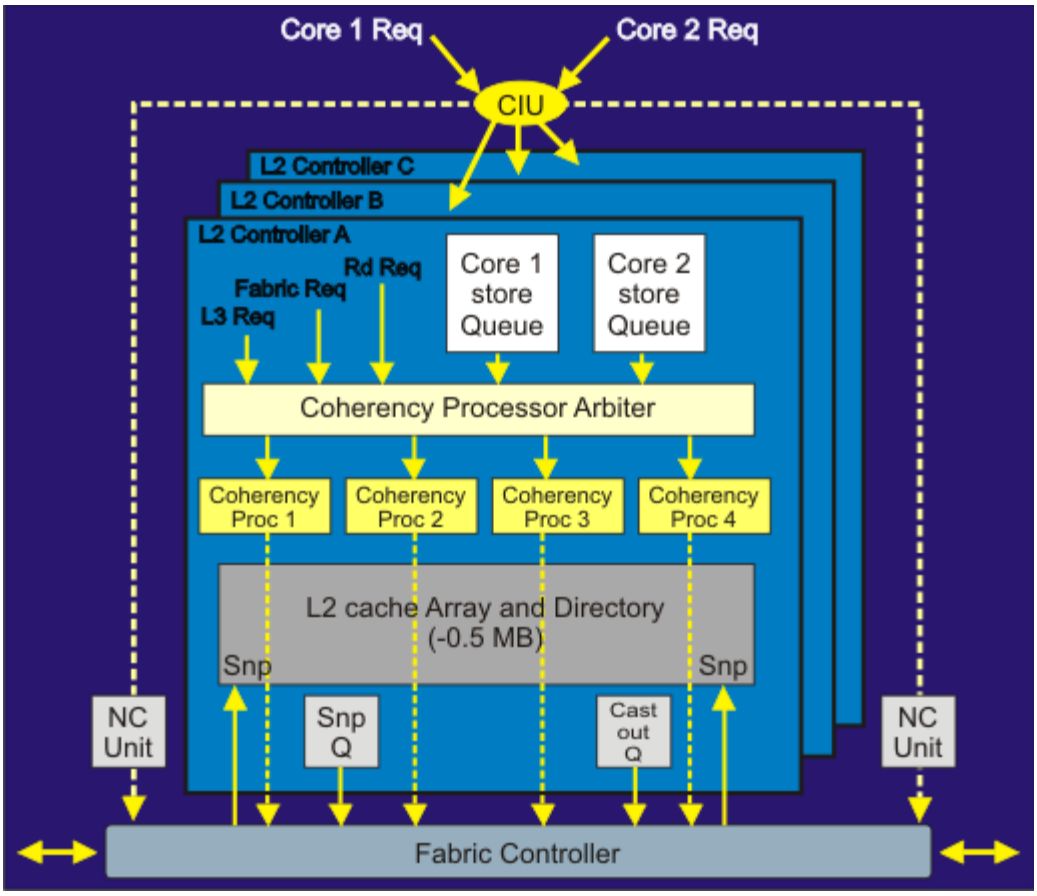


Previous Next

POWER4 caches

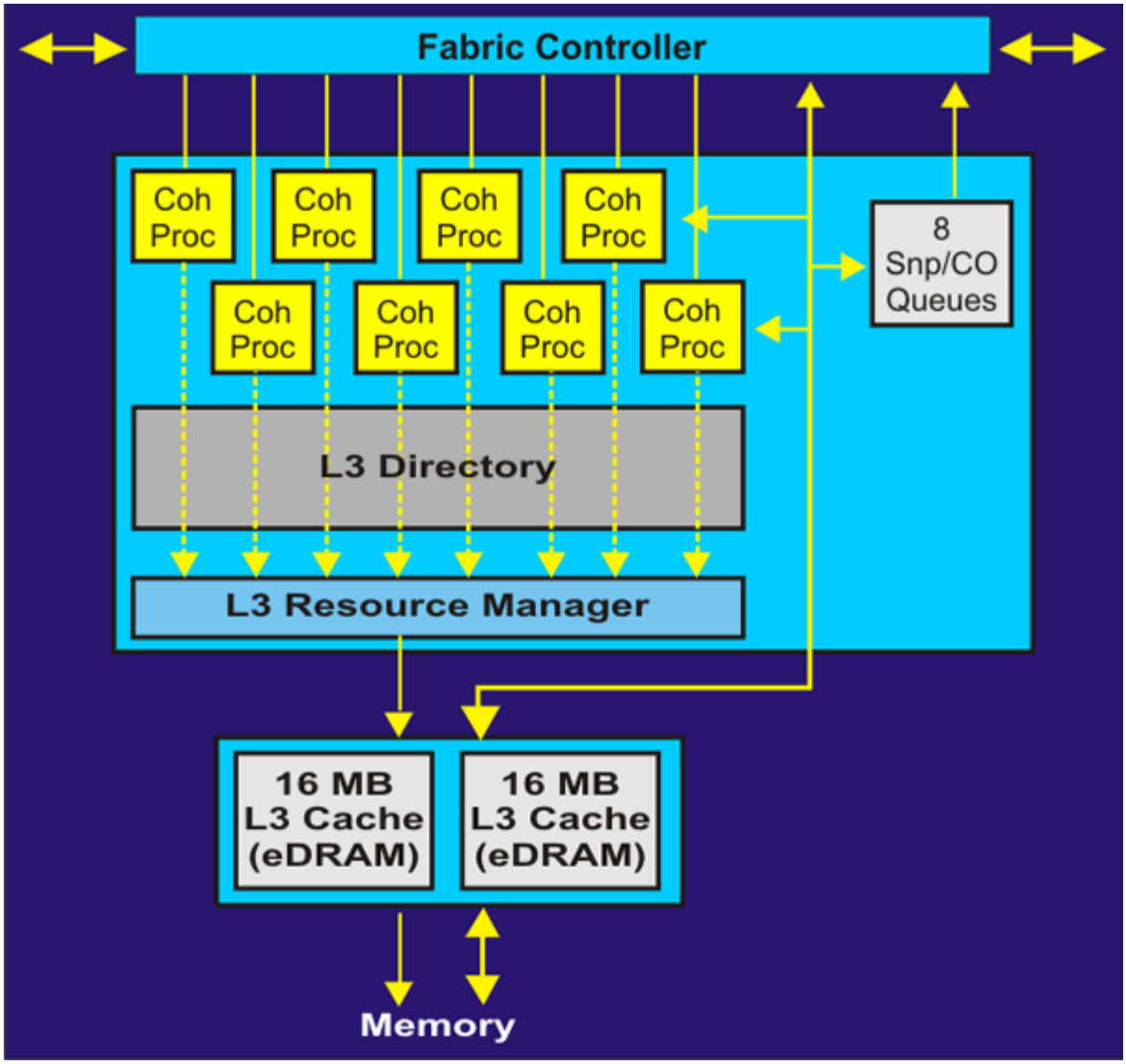
- Private L1 instruction and data caches (on chip)
  - L1 icache: 64 KB/direct mapped/128 bytes line
  - L1 dcache: 32 KB/2-way associative/128 bytes line/LRU
  - No M state in L1 data cache (write through)
- On-chip shared L2 (on-chip coherence point)
  - 1.5 MB/8-way associative/128 bytes line/pseudo LRU
  - For on-chip coherence, L2 tag is augmented with a two-bit sharer vector; used to invalidate L1 on other core's write
  - Three L2 controllers and each L2 controller has four local coherence units; each L2 controller handles roughly 512 KB of data divided into four SRAM partitions
  - For off-chip coherence, each L2 controller has four snoop engines; executes enhanced MESI with seven states

POWER4 L2 cache

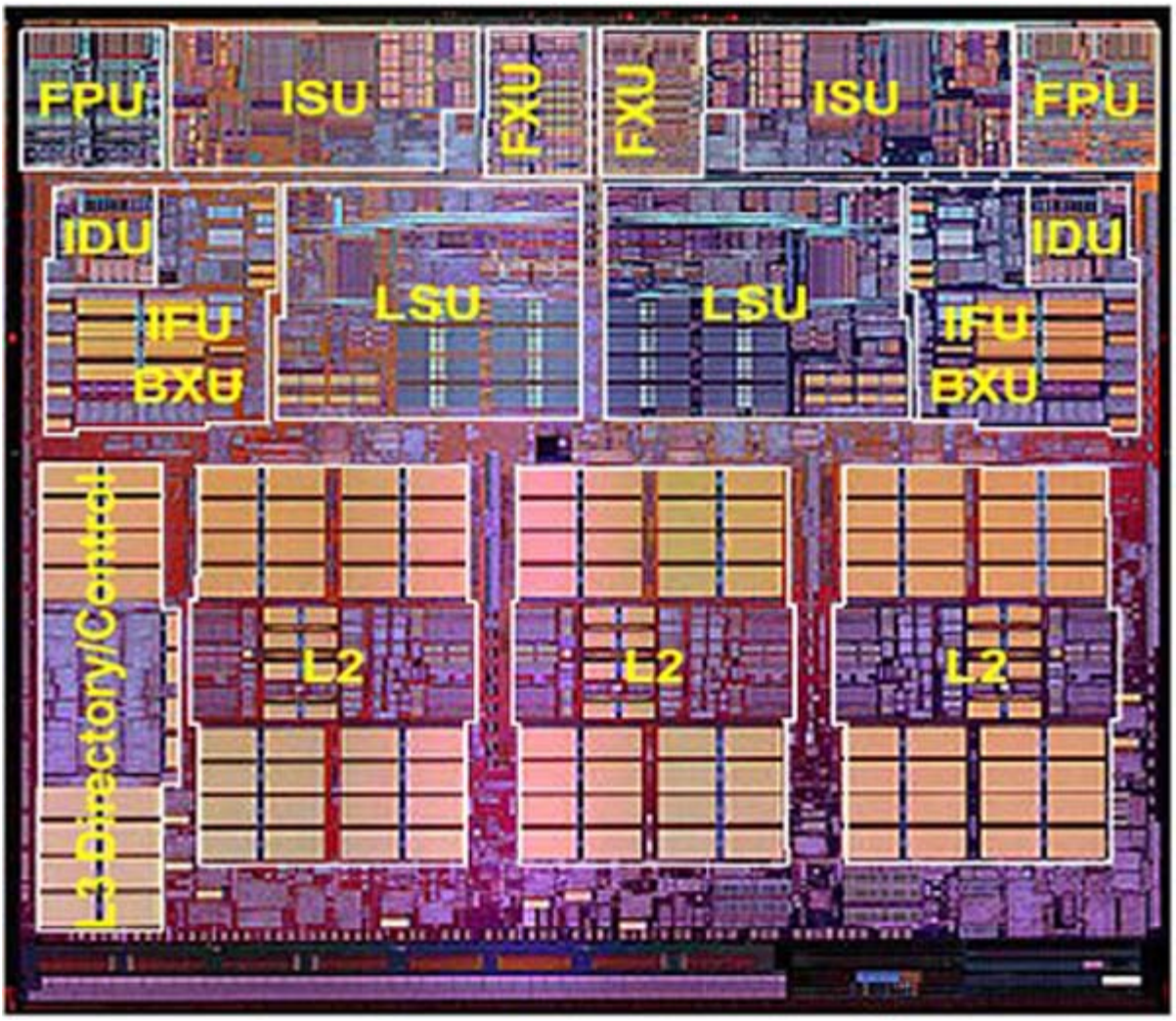


POWER4 L3 cache

- On-chip tag (IBM calls it directory), off-chip data
  - 32 MB/8-way associative/512 bytes line
  - Contains eight coherence/snoop controllers
  - Does not maintain inclusion with L2: requires L3 to snoop fabric interconnect also
  - Maintains five coherence states
  - Putting the L3 cache on the other side of the fabric requires every L2 cache miss (even local miss) to cross the fabric: increases latency quite a bit



POWER4 die photo



Previous Next

IBM POWER5

IBM POWER5

- Carries on POWER4 to the next generation
  - Each core of the dual-core chip is 2-way SMT: 24% area growth per core
  - More than two threads not only add complexity, may not provide extra performance benefit; in fact, performance may degrade because of resource contention and cache thrashing unless all shared resources are scaled up accordingly (hits a complexity wall)
  - L3 cache is moved to the processor side so that L2 cache can directly talk to it: reduces bandwidth demand on the interconnect (L3 hits at least do not go on bus)
  - This change enabled POWER5 designers to scale to 64-processor systems (i.e. 32 chips with a total of 128 threads)
  - Bigger L2 and L3 caches: 1.875 MB L2, 36 MB L3
  - On-chip memory controller

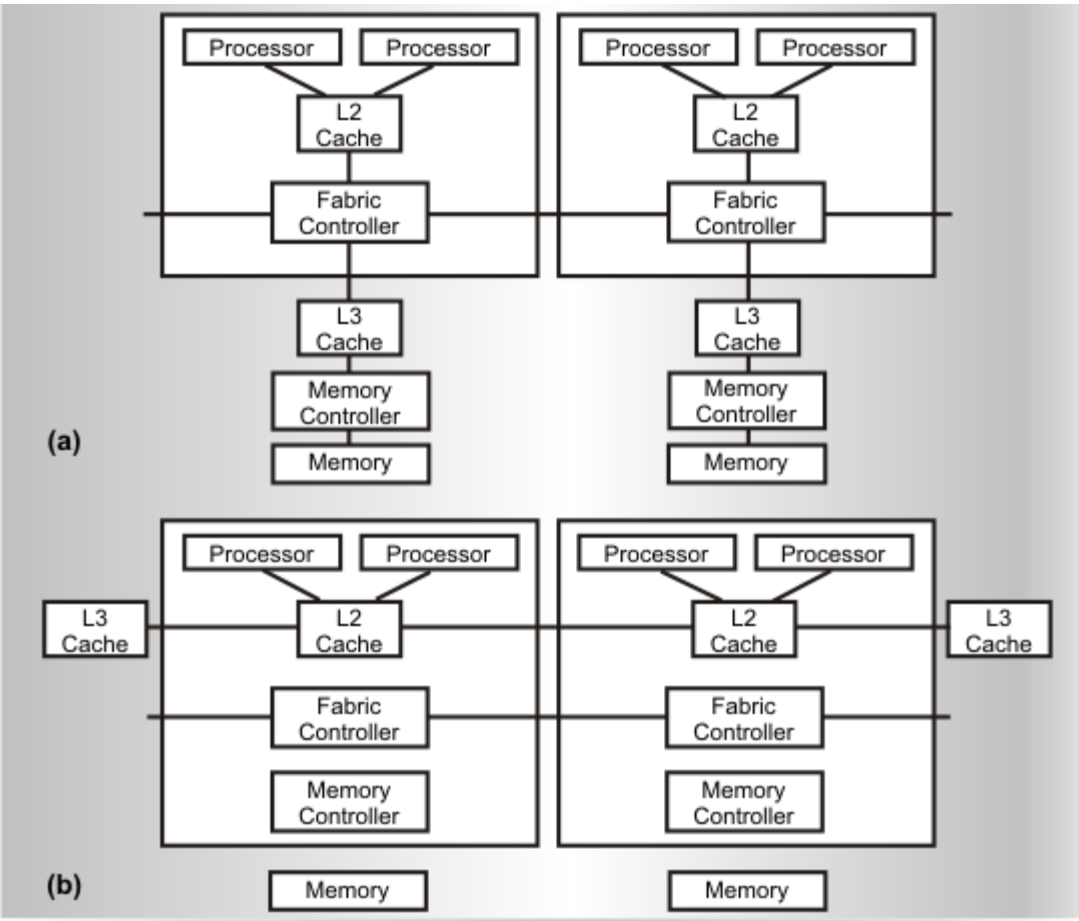


Figure 1. Power4 (a) and Power5 (b) system structures.

Reproduced from IEEE Micro

- Same pipeline structure as POWER4
  - Added SMT facility
  - Like Pentium 4, fetches from each thread in alternate cycles (8-instruction fetch per cycle just like POWER4)
  - Threads share ITLB and ICache

- Increased size of register file compared to POWER4 to support two threads: 120 integer and floating-point registers (POWER4 has 80 integer and 72 floating-point registers): improves single-thread performance compared to POWER4; smaller technology (0.13 µm) made it possible to access a bigger register file in same or shorter time leading to same pipeline as POWER4
- Doubled associativity of L1 caches to reduce conflict misses: icache is 2-way and dcache is 4-way

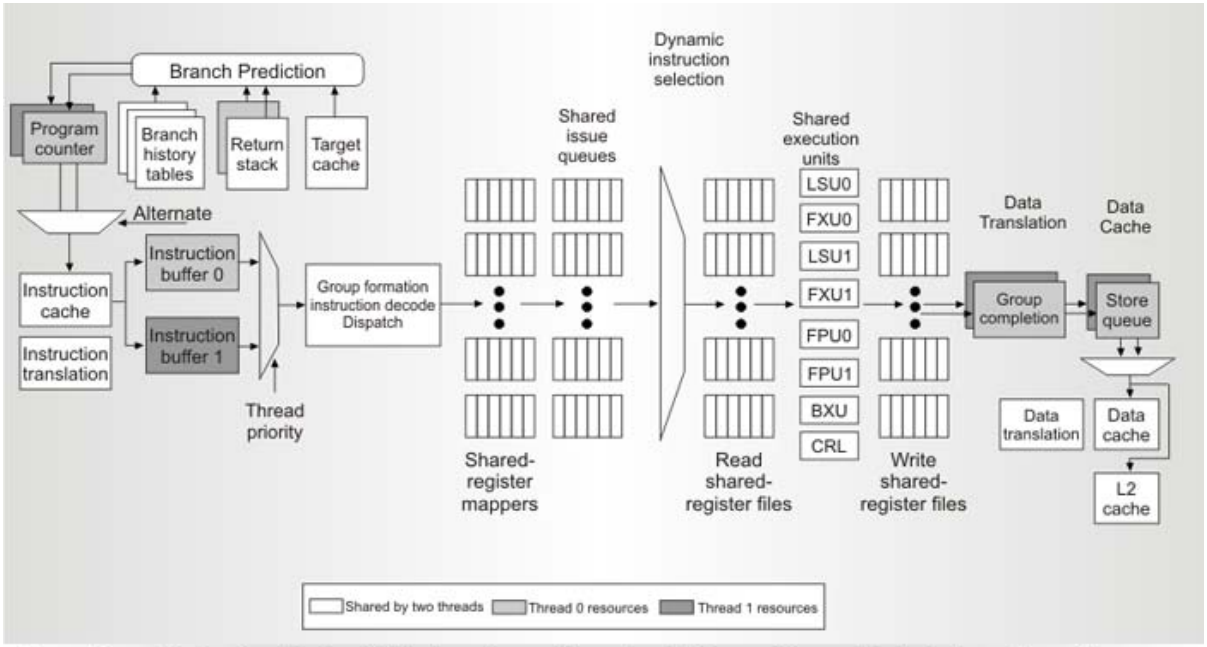


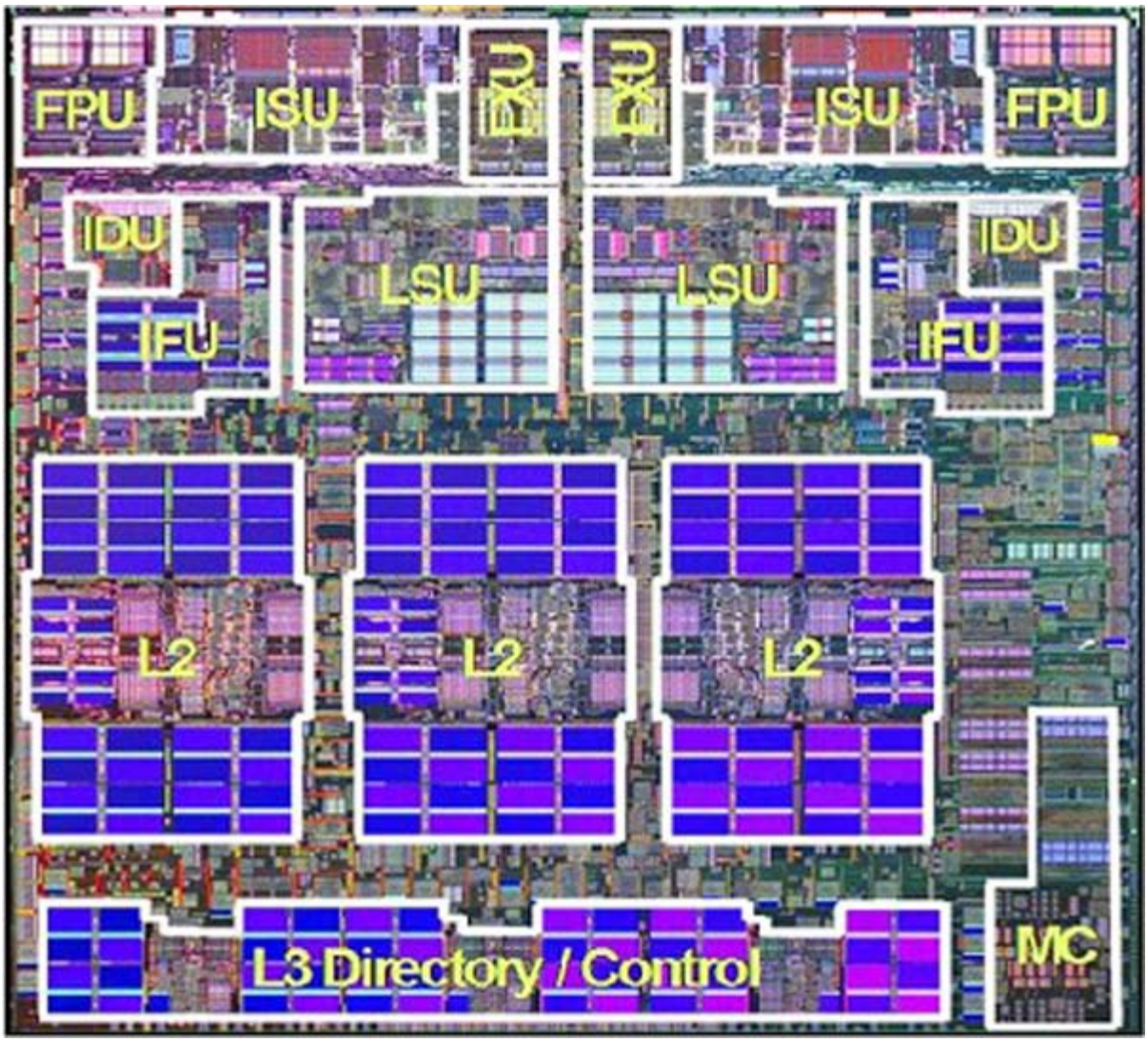
Figure 4. Power5 instruction data flow (BXU = branch execution unit and CRL = condition register logical execution unit).

Reproduced from IEEE Micro

IBM POWER5

- Thread priority
  - Software can set priority of a thread and the hardware (essentially the decoder) reads these priority registers to decide which thread to process in a given cycle
  - Higher priority thread gets more decode cycles in the long run i.e. injects more instructions into the pipe
  - Eight priority levels for each thread: level 0 means idle
  - Real time tasks get higher priority while a thread looping on a spin-lock will get lower priority
  - Level 1 is the lowest priority for an active thread; if both threads are running at level 1 the processor throttles the overall decode rate to save dynamic power
- Adaptive resource balancing
  - Mainly three hardware mechanisms used by POWER5 to make sure that one thread is not hogging too much
  - If one thread is found to consume too many GCT entries i.e. has too many in-flight instructions (one GCT entry is at most 5 instructions), that thread will get less decode cycles until GCT occupancy reaches a balanced state (note the difference with ICOUNT)
  - If a thread has too many outstanding L2 cache misses, that thread will be given less decode cycles (why?)
  - If a thread is executing a sync, all instructions belonging to that thread that are waiting in the pipe at the dispatch stage will be flushed and fetching from that thread will be inhibited until sync finishes (why?)
- Dynamic power management
  - With SMT and CMP average number of switching per cycle increases leading to more power consumption
  - Need to reduce power consumption without losing performance: simple solution is to clock it at a slower frequency, but that hurts performance
  - POWER5 employs fine-grain clock-gating: in every cycle the power management logic decides if a certain latch will be used in the next cycle; if not, it disables or gates the clock for that latch so that it will not unnecessarily switch in the next cycle
  - Clock-gating and power management logic themselves should be very simple
  - If both threads are running at priority level 1, the processor switches to a low power mode where it dispatches instructions at a much slower pace

POWER5 die photo



Previous Next