Portland State University
ECE 588/688

IBM Power4 System Microarchitecture

IBM Power4 Design Principles

- SMP optimization
  - Designed for high-throughput multitasking environments
- Full system design approach
  - Whole system designed together, processor designed with full system in mind
- High frequency design
  - Important for single-threaded applications
- RAS: Reliability, Availability, and Serviceability
- Balanced scientific vs. commercial performance
  - Good performance for both high-performance computing scientific applications & commercial server applications
- Binary compatibility with previous IBM processors

Power4 Chip Features

- Two processors on a chip (Figure 1, die photo in Figure 2)
- Each processor has private L1 caches
- Both processors share an on-chip L2 cache through a core interface unit (CIU)
  - Crossbar between two processors' I and D L1 caches and three L2 controllers
  - Each L2 controller can feed 32B per cycle
  - Accepts 8B processor stores to L2 controllers
- Each processor has a noncacheable unit (NC)
  - Logically part of L2, handles noncacheable operations
- L3 directory and L3 controller on chip
  - Actual L3 cache on separate chip
  - Fabric controller controls data flow between L2 & L3 controller

Power4 Processor Features

- On-chip, two identical processors provide two-way SMP to software (an example for chip multiprocessing)
- Each processor is a superscalar out-of-order processor
  - Issue width: up to 8, retire width: 5
  - 8 instruction units, each capable of issuing one inst/cycle
  - Two floating point execution units, each can start an FP add and FP multiply every cycle
  - Two load/store units, each can perform address generation arithmetic
  - Two fixed point execution units
  - Branch execution unit
  - Condition register logical execution unit
- Core block diagram: paper figure 3

Power4 Microarchitecture

- Complex branch prediction
  - Branch target and direction prediction
  - Has a selector table to choose between a Local branch history table and global history vector
  - Selective pipeline flush on branch misprediction
- Instructions are decoded, cracked into internal instructions (IOPs), then grouped into five-instruction groups
  - Fifth IOP is always a branch
  - Groups dispatched in order, IOPs in a group issued out of order
  - Whole group committed together (up to 5 IOPs)
- Issue queues: paper table 1, rename resources: table 2
- Pipeline in paper figure 4

Load/Store Unit Operation

- Main structures
  - Load Reorder Queue (LRQ), i.e., load buffer
  - Store Reorder Queue (SRQ), i.e., store address buffer
  - Store Data Queue (SDQ)
- Hazards avoided by Load/store unit
  - Load hit store (RAW1): Younger load executes before older store writes data to memory. Load should get data from SDQ. Possible flush or reissue
  - Store hit load (RAW2): Younger load executes before recognizing an older store will write to same location. Store checks LRQ and flushes all subsequent groups on hit
  - Load hit load (RAR): If younger load got old data, older load should not get new data. Older load checks snooping bit in LRQ for younger loads, flushes all subsequent groups on hit

Memory Hierarchy

- Memory hierarchy details in paper table 3
- L2 logical view in paper figure 5
- L3 logical view in paper figure 6
- Memory subsystem logical view in paper figure 7
- Hardware prefetching
  - Eight sequential stream prefetchers per processor
  - Prefetch data to L1 from L2, to L2 from L3, and to L3 from memory
  - Streams initiated when processor misses sequential cache access
  - L3 prefetches 512B lines

Cache Coherence

- Each L2 controller has four coherency processors to handle requests from either processor's caches or store queues
  - Controls return of data from L2 (hit) or fabric controller (miss) to the requesting processor
  - Updates L2 directory state
  - Issues commands to fabric on L2 misses
  - Controls writing to L2
  - Initiates invalidates to a processor if a processor's store hits a cache line marked as being resident in another processor's L1
- L2 controller has four snoop processors to handle coherency operations from fabric
  - Can source data to another L2 from this L2

Coherence Protocol

- L2 has enhanced version of MESI (paper table 4)
  - I: Invalid
  - SL: Shared, can be sourced to local requesters
    - Entered when processor load or I-fetch misses L2, data is sourced from another L2 or from memory
  - S: Shared, cannot be sourced
    - Entered when another processor snoops cache in SL state
  - M: Modified, can be sourced
    - Entered on processor store
  - Me: Exclusive
  - Mu: Unsolicited modified
    - Entered when data is sourced from another L2 in M state
  - T: Tagged (valid, modified, sourced to another L2)
    - Entered on a snoop read from M state
- L3 has simpler protocol (paper)

Connecting into larger SMPs

- Basic building block is Multi-Chip Module (MCM)
  - Four Power4 chips form an 8-way SMP (paper figure 9)
  - Each chip writes to its own bus (with arbitration among L2, I/O controller and L3 contoller)
  - Each of the four chips snoops all buses
- 1-4 MCMs can be connected to form 8-way, 16-way, 24-way and 32-way SMPs
  - 32-way SMP shown in paper figure 10
  - Intermodule buses act as repeaters, moving requests and responses from one module to another in a ring topology
  - Each chip writes to its own bus but snoops all buses

Reading Assignment

- Monday
  - Erik Lindholm et al., "Nvidia Tesla: A Unified Graphics and Computing Architecture", IEEE Micro, 2008 (Read)
- Wednesday
  - John Mellor-Crummey and Michael Scott, "Synchronization Without Contention," ACM Transactions on Computer Systems,1991 (Read)
  - Thomas Anderson, "The Performance of Spin-Lock Alternatives," IEEE Transactions on Parallel and Distributed Systems, 1990 (Skim)
  - Ravi Rajwar and James Goodman, "Speculative Lock Elision: Enabling Highly-concurrent Multithreaded Execution," Micro 2001 (Skim)
- Project progress report due on Monday