CS3210

Parallel Computing

Changes from Monday in Green

Tut 1

Mon (4pm)
Tues (2pm)

Keven 2020

Admin

Updates

- Lab 1 submission
  - If you have not submitted yet, please do so ASAP (you can submit your current work - even if it has a few issues)

- Assignment 1 released on LumiNUS
  - Part 1 (process, POSIX threads) - due 14 Sep, 11am
  - Part 2 (OpenMP) - due 28 Sep, 11am
  - Do not start on the assignment just days before the deadline
  - Check the FAQ before posting your questions on LumiNUS

Admin

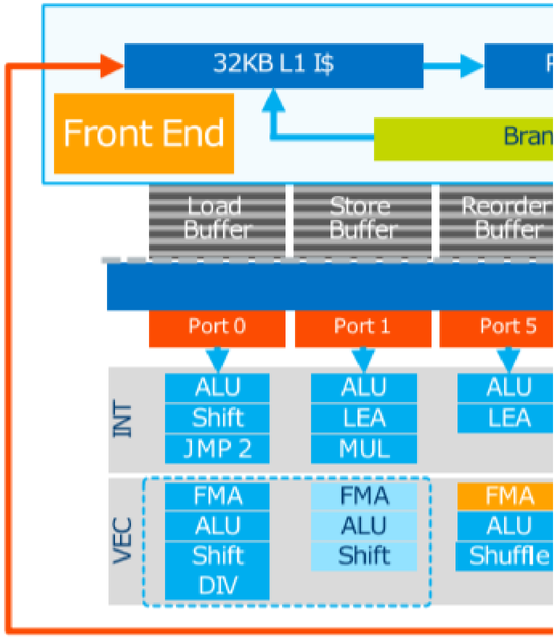
Roadmap

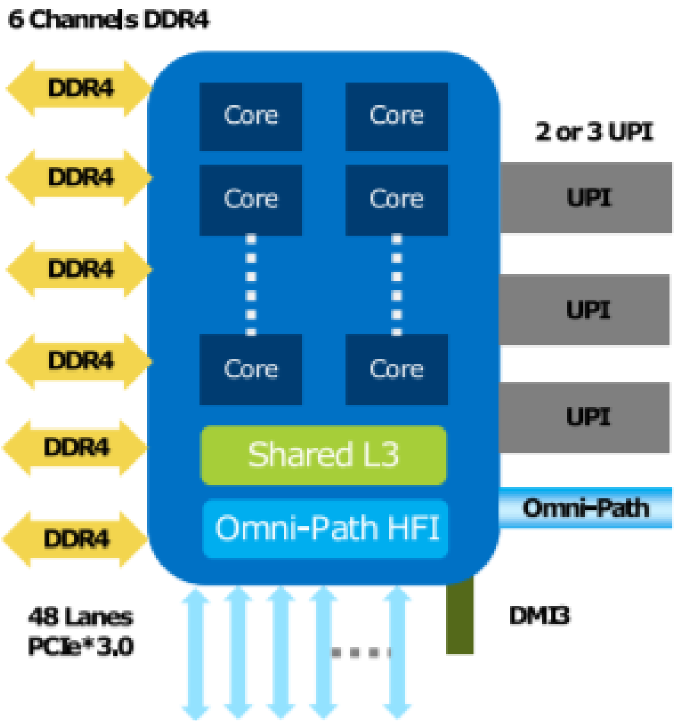
- Today’s tutorial
  - Part 1: questions on Parallel Computer Architecture
  - Part 2: short MCQ quiz
  - Part 3: pre-requisite lab exercise for Assignment 1
  - Part 4: Lab 1 review

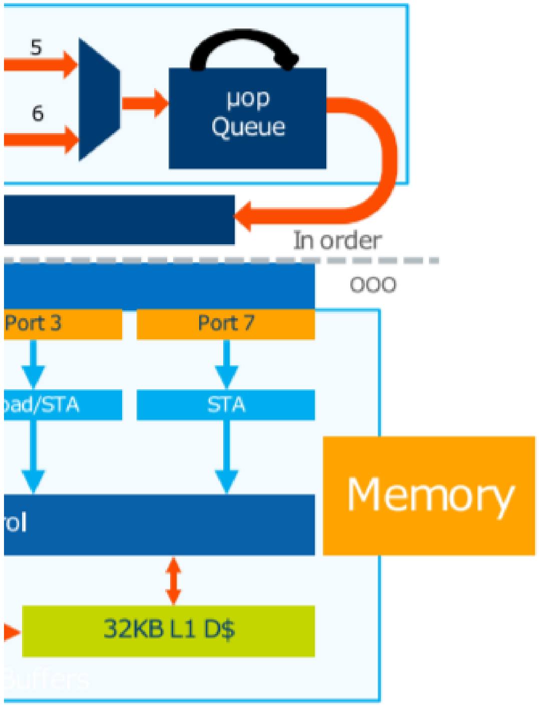
- If you still have issues with your SOC account or accessing Sunfire, please drop us a message
  - **Parallel Computing lab nodes should be up 24/7 from now until Assignment 1 is over**

Q1
Levels of Parallelism

- What levels of parallelism can you identify below?







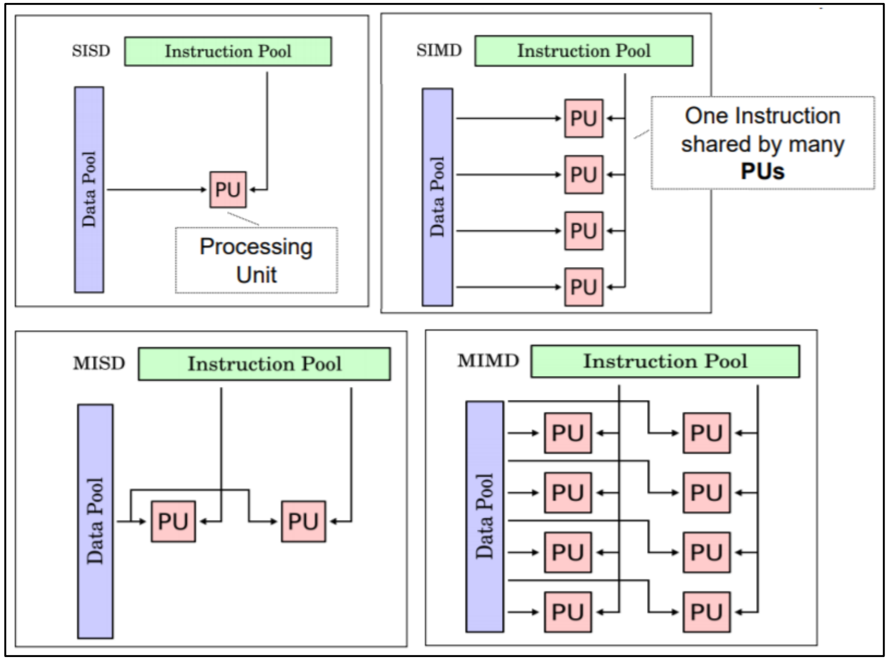
Q1
Levels of Parallelism

- Levels of parallelism
  - Bit-level parallelism? **Not really, can't infer 64-bit word size**
  - Instruction-level parallelism? **Yes! μops pipeline can be duplicated (superscalar execution)**
  - Thread-level? **Yes! Simultaneous multi-threading (SMT) - hardware support for threads**
  - Process-level parallelism? **Yes! Multiple cores; can cooperate through shared memory, cache or message-passing**
  - Processor-level parallelism? **Yes! Interconnects for shared-memory (Intel UltraPath) and distributed (Intel Omni-Path) systems**

Review

Flynn's Taxonomy

- How does Flynn's Taxonomy categorise systems?
  - By the number of independent instruction streams and data streams -> four types (SISD, SIMD, MISD, MIMD)



Q2
Flynn's Taxonomy (1)

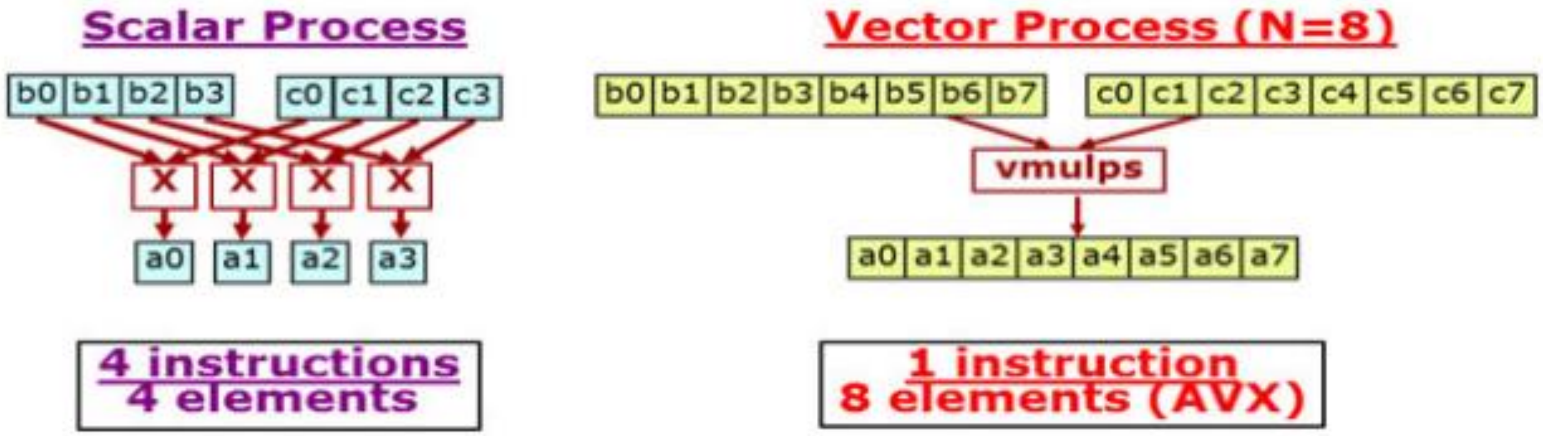
- How would you thus classify the following?

- A personal computer from the 1980s?
  - ➢ SISD: most computers in that era are uni-processor (single-core) systems, and only capable of executing one instruction stream on a single data stream

- A laptop with a multi-core processor?
  - ➢ MIMD: each core can execute different threads, which can operate on different data streams

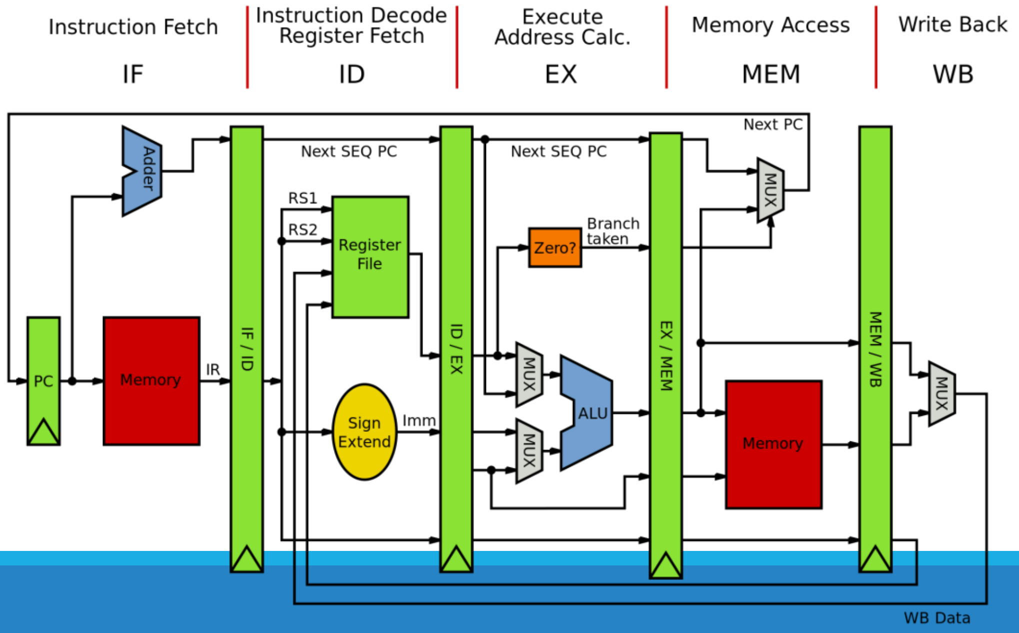
Q2
Flynn’s Taxonomy (2)

- Intel’s AVX instruction set
  - SIMD: one instruction stream (e.g. vmulps) operates on different sets of data at the same time (VLIW)



Q2
Flynn's Taxonomy (3)

- A uniprocessor (single-core) with pipelining (ILP)
  - SISD: despite pipelining, it is still executing from the same instruction and data stream, coming from the same program
  - Question does not mention superscalar execution (where the full pipeline or parts thereof are duplicated in one core)



Q2
Flynn's Taxonomy (4)

- Students attempting the same exam in an exam hall
  - MISD (?): same exam script for all students, but students are working on different parts of the exam paper at any single moment (multiple instruction streams)
  - How else can we view this?

Q3
Memory and Multicore Arch. (1)

- Does a shared-memory system imply a UMA architecture?
  - No! There are also NUMA and COMA architectures in shared-memory systems
  - Shared-memory vs distributed memory: describes the view of memory (the address space) by processors
  - UMA, NUMA, COMA: describes the physical layout of memory modules in the entire system and its implications

Q3
Memory and Multicore Arch. (2)

- On a NUMA architecture, the actual location of data has no impact on a distributed shared-memory program
  - ➢ What is a *distributed shared-memory program*?
  - ➢ Distributed tasks can access local (private) data, and can transfer data to other tasks
  - ➢ **No! The memory is a unified address space (negotiated by a shared memory provider), but with non-uniform access as the physical memory is distributed across all PUs**
  - ➢ Different pieces of data are located on different PUs → translates to different overheads for a distributed task

Q3
Memory and Multicore Arch. (3)

- In the hierarchical design of multicore architecture, the memory organization is hybrid (distributed-shared memory)
  - What do you consider as part of the memory organization?
  - True? If we consider cache as part of memory organization - low-level caches are distributed, higher-level caches/main memory is shared
  - False? If we focus only on main memory - not guaranteed the architecture has shared memory (some supercomputing networks)

Q4
Processes and Threads

- A semaphore can be used to replace a mutex without affecting the correctness of a program
  - True! Initialise the semaphore with initial value of 1 - might require minor changes in the program logic
  - Converse not true (mutex has ownership, semaphore doesn't)

- Implementing an algorithm using multiple threads **always** runs faster than using multiple processes
  - False! Slower if using user-level threads (mapped to one process), and overhead introduced by library-level scheduling

Admin

Tutorial Quiz

- Go to Luminus > Quiz
  - 1% of your grade
  - 8 minutes for 5 MCQ questions
  - **Don’t be stressed - full credit for ≥ 60% correct (you should be able to do this after the tutorial, well, hopefully)**

- 10 minutes now to attempt it

Admin

Lab Task

- SSH into any lab machine via Sunfire (or SoCVPN)
  - Test if nodes are online with `ping -c 5 <node hostname>`

- Set up your credentials for password-less access
  - Generate SSH key pair: `ssh-keygen`
  - Copy your **PUBLIC KEY** over the network to another node: `ssh-copy-id <username>@<other node hostname>`
  - SSH into that node to verify it works

- Test distributed (OpenMPI) program

CS3210

Parallel Computing

Thank you! Any questions?

Tut 1

Mon (4pm)
Tues (2pm)