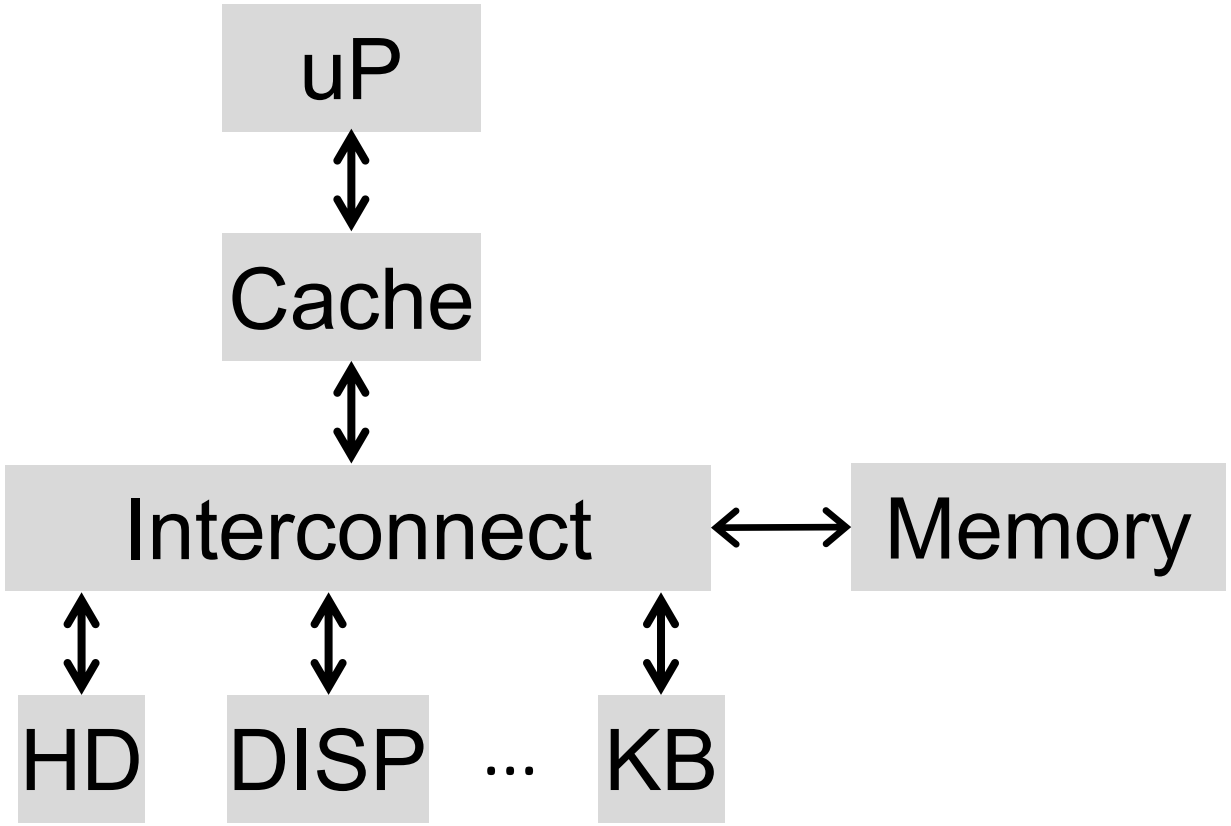
EEL 6764 Principles of Computer Architecture
Final Review

Dr Hao Zheng
Dept. of Comp Sci & Eng
U of South Florida

Computer Architecture – Big Picture



ISA Design

- Different architectures
  - Classified by organizations of operands -- stack, accumulator, GP registers
  - Classified by granularity – CISC vs RISC
- Objectives: performance, implementability, compatibility
- Considerations
  - Number of registers
  - Number of operands
  - Memory addressing mode
  - Types of instructions
  - Instruction encoding – fixed vs flexible
    - impact of code size

ILP and Pipelining

- ILP – overlapped executions of different instructions
- Pipelining – architecture to exploit ILP
- Architecture of 5-stage MIPS pipeline
  - IF, ID, EX, MEM, WB
- Ideal pipeline performance
  - Speedup is close to number of pipeline stages
  - Pipeline hazards, mem & FU latency, pipeline register delays
- Pipeline hazards
  - Structural – a hardware issue
  - Data – dependencies in programs
  - Control - related to branches

ILP and Pipelining

- Hazard handling – stall (simple, but undesirable)
  - Structural – replicate HW, or pipelining slow components
  - Data – RAW, WAR, WAW
    - forwarding for RAW, register renaming for WAR & WAW
  - Branch
    - Branch prediction -- static & dynamic, 1-bit & 2-bit predictors
    - HW speculation

ILP and Pipelining

- **Scheduling** – what is it?
  - Static vs dynamic – issues of static scheduling
  - register renaming – WAW & WAR hazards

- **Tomasulo's algorithm**
  - reg renaming + dynamic scheduling
  - reservation stations, CDB, tags for registers
  - operations – see Figure 3.13.

- **HW speculation** – what is it, and how does it work?
  - What problem does it solve?
  - What additional HW is used and operated? (see Fig. 3.18)
  - Efficiency depends on branch prediction accuracy

ILP and Pipelining

- **Multi-Issue**
  - Goal: reduce CPI to <1
  - Challenge – complexity of issue logic (see Fig 3.22)
- **Multithreading** – target thread-level parallelism
  - Fine-grained
  - Coarse-grained
  - Simultaneous multithreading – additional PCs and registers

Memory Hierarchy Design

- Mem latency – limiting factor of performance
- **Cache** – Reduce average memory access latency
  - spatial & temporal locality
  - Organization
  - Set Associativity
  - Cache misses, and their causes
  - Performance measurement
  - Write policy – write-back vs write-through
  - Optimization techniques
- **Virtual memory** – memory as cache for hard disk
  - Roles: memory management and protection
  - Organization: page, page table, page faults; addr translation
  - TLB – cache for page table

TLP and Multiprocessors

- Parallel execution of instructions from different threads
- Multiprocessor = a set of processors connected together
  - Support MIMD execution model
- **Multiprocessing** vs **multithreading**
- Architectures
  - Symmetric MP, aka, centralized shared memory MP
  - Distributed shared memory MP
- Coordination – shared memory
- Caching – reduce remote memory access latency
- Introduce coherence problem
- **Cache Coherence protocols** – snooping vs directory
  - Write-invalidate vs write-update

Vector Processors

- DLP – parallel operations on different data
- Vector architecture
  - Exploit DLP
  - Support SIMD execution model
- Target applications with many vector operations/loops
- Main architecture features
  - Vector registers,
  - Vector load/store unit, addressing with stride, gather-scatter
  - Pipelined functional units, multiple lanes, chaining
  - Vector length register, strip mining
  - Vector predicate register for conditional execution over vectors
- Differences against superscalar processors