# CSE 5600
# Computer Systems Architecture

Exam 1 Review

1

## Logistics

- Date and Time
  - Monday, Oct 12, in class, 2:30pm to 3:50pm, starting right at 2:30!
- Location is in-person here in Lopata 101

2

## Style

- Questions
  - Question 1 will be a collection of short answer things (e.g., true/false, fill in the blank, quick definition)
  - Questions 2 through N will be longer (going more in depth on a particular subject)
- Closed book, closed note, NO partners!
- One-page "crib sheet" is allowed
  - 8.5 x 11 sheet, front and back, whatever you want to include (content-wise)
- Calculator is not allowed (not needed), so show your work

3

## Practice Problems

- Designed to be practice for exam
- Wording style mimics exam questions
  - Many are old exam questions, or derived from them
- One distinction is length
  - Some practice problems are longer than what is reasonable to ask in an exam question

4

## Instruction Set Architecture

- Definition
- Components
  - Machine instructions
  - Programmer-level memory model
  - Operating Modes
- What isn't part of the ISA, e.g.,
  - Pipelined execution
  - Branch predictors
  - Caches

5

## Qualities of a Good ISA

- Programmability
- Implementability
- Compatibility

6

## Specifics of ISAs

- Instruction Format
- Register File
- Main Memory
  - Little endian, big endian
- Addressing Modes
- Conditional Branching
  - Condition Codes
  - Test and Branch
- RISC vs. CISC

7

## Performance

- Metrics
  - Latency
  - Throughput
- Fundamental Equation

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{cycle}}$$

  - Calculation of CPI accounting for stalls
- Benchmarks
- Architectural Simulation

8

## Reporting Performance

- Aggregating data
  - When to use
    - Arithmetic mean
    - Harmonic mean
    - Geometric mean
  - How to compute them

9

## Pipelining

- 5-stage example pipeline
  - What happens where
  - Data dependencies
  - Hazards
  - Bypassing (forwarding)
- Pipeline diagrams
- Multi-cycle operations
  - Pipelined functional units (e.g., multiplier)

10

## Branch Prediction

- Why?
- How?
  - Classic types
- Where?
  - Esp. in context of 5-stage pipeline
- Performance implications?

11

## Parallel Pipelines

- Superscalar
  - In-order execution
- Scheduling
  - By the compiler
  - Single issue and multiple issue
  - Loop unrolling
- Pipeline diagrams
  - Emphasis on when does the X stage happen

12

## Out-of-Order Execution

- Register Renaming
  - Free list
- Dynamic Scheduling
  - Instruction queue
  - Issue logic
- Commit
  - Reorder buffer
- Memory – load and store queues
- OoO Pipeline Diagrams

13

## Time for Q&A

- We'll start right at 2:30pm
- Closed book, closed notes, NO partners
- One page "crib sheet" allowed (both sides), it does not have to be handwritten
- Calculator not required
- Read questions carefully (esp. pipeline diagrams)

14