Virtual Memory (and Cache)

Team Members: Bonnie Ishiguro, Apurva Raman, Shruti Iyer

Description:

We would like to explore how virtual memory and caches work and how they fit into a CPU design. We aim to understand the various ways virtual memory and caching can be implemented.

References:

1. Computer Principles and Design in Verilog HDL by Yamin Li
2. Computer Organization and Design by David Patterson and John L. Hennessy
3. Lecture 12: Memory hierarchy & caches The University of Edinburgh

Deliverables:

Minimum

We would like to create a textbook-chapter-style guide that describes the theory of virtual memory, possible implementations with analysis, and commonly used optimization methods.

Planned

In addition to our research on virtual memory, we plan to explore optimization by implementing a cache replacement policy algorithm (Random Selection, First In First Out, or Least Recently Used) and demonstrate it on a series of registers that will act as the cache levels typically found in processors.

Stretch

Our stretch goal is to implement the virtual memory in Verilog.

Workplan:

Research: **13.5 hr**

Memory hierarchy: **2 hr**

- Properties of types of memory (disk, DRAM, SRAM, etc.) **2 hr**
    - Features of types of memory **1.25 hr**
    - Timing for types of memory **0.75 hr**

Virtual memory: **10 hr**

- Pages and segments **2.5 hr**
- Space, faults, sources of error **1 hr**
- Translation table **1.5 hr**
- Page placement policy **2 hr**
- Translation lookaside buffer **3 hr**

Memory cache: **1.5 hr**

- Cache replacement policy algorithms **1 hr**
- Page size vs cache size **0.5 hr**

Draft Write-Up: **3 hr + 2 hr / person**

Outline: **0.5 hr** 11/29
Teaching: **1.5 hr** 12/4
Draft: **2 hr / person** 12/4
Figures / formatting: **0.5 hr**
Citations: **0.5 hr**

Implementation (**13.25 hr ?**)

- Choose a cache replacement policy algorithm (Random, FIFO, or LRU) **0.25 hr**
- Choose processor **1 hr**
- Build memory hierarchy **1 hr**
    - Cache
- Simulate timing of write/read to caches **3 hr (?)**

    - Control circuitry
- Simulate OS or instruction set to move data between caches **4 hr (?)**
- Implement cache replacement policy algorithm **2-4 hr**

Documentation: **4.75 hr**

Project abstract **0.5 hr**
Motivation **0.5 hr**
Process **1 hr**
Building on where we left off **2.75 hr**

- Link to virtual memory write-up **(-)**
- Code **(-)**
- Schematics **1 hr**
- Scripts and build instructions **0.25 hr**
- Attribution for resources **(-)**
- Difficulties and “gotchas” **0.5 hr**
- Reflection on project and work plan **0.5 hr**
- Possible TODOs to extend project depth **0.5 hr**

Poster: **2-4 hr**

Timeline:

Timeline (due dates):

11/29

- Outline for virtual memory write-up

12/1

- Schedule midpoint check-in

12/4

- Finished research
- First draft of virtual memory write-up
- Choose style of documentation

12/5

- Midpoint check-in

- Schedule implementation meeting with Ben

12/8

- Discuss roadblocks

12/12

- Poster printed
- Cache replacement algorithm
- Simulate

12/15:

- Presentation