Quiz 1 Review

TA: Mark Jeffrey
Adapted from: Suvinay Subramanian, 2016
EDSAC

• Accumulator based
• No concept Index Registers, PC
• Use self-modifying code for indirect accesses, subroutine calls etc.
Caches

![Diagram of cache structure]

- **Tag**: Log2(cache line size) bits
- **Index**: Log2(num sets) bits
- **Block Offset**: Remainder bits
Address Translation

Parameters

- $P = 2^p$ = page size (bytes).
- $N = 2^n$ = Virtual-address limit
- $M = 2^m$ = Physical-address limit

Page offset bits do not change with translation
Hierarchical Page Tables

Virtual Address

<table>
<thead>
<tr>
<th>Index 1</th>
<th>Index 2</th>
<th>Index 3</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>23</td>
<td>17</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Context Register</th>
</tr>
</thead>
</table>

L1 Table

PTP

L2 Table

PTP

L3 Table

L1 Table

PTP

L2 Table

PTP

L3 Table

PTE

Physical Address

PPN

Offset

TLB:
- Holds VPN --> PPN mapping

TLB miss → Page Walk

6.823 Spring 2017
Physical vs Virtual Address Caches?

- Physical Cache
  - CPU
  - TLB
  - Physical Cache
  - Primary Memory

- Virtual Cache
  - CPU
  - Virtual Cache
  - TLB
  - Primary Memory

- Cache
  - CPU
  - TLB
  - Cache
  - Primary Memory
Concurrent Accesses to TLB and Cache

Virtual Address <40>

Virtual Page Number <28>  Page Offset <12>

L1 index <6>  Block Offset <6>

L1 Tag <28>  L1 Data <512>

TLB Tag <28>  TLB Data <28>

Physical Page Number

TLB Hit/Miss  L1 Hit/Miss

L1 Size = Page Size

To CPU  To CPU
Pipelining
Pipeline Hazards

1. Structural
2. Data
3. Control
Handling Pipeline Hazards: Recipe

1. Stall

2. Bypass

3. Speculate
Instruction Flow Diagram

\[ \text{time} \]

(I_1) \quad r_1 \leftarrow r_0 + 10

(I_2) \quad r_4 \leftarrow r_1 + 17

(I_3)

(I_4)

(I_5)

\[ \begin{align*}
\text{IF}_1 & \quad \text{ID}_1 \\
\text{EX}_1 & \quad \text{MA}_1 \quad \text{WB}_1 \\
\text{IF}_2 & \quad \text{ID}_2 \quad \text{ID}_2 \quad \text{ID}_2 \quad \text{ID}_2 \\
\text{IF}_3 & \quad \text{IF}_3 \quad \text{IF}_3 \quad \text{IF}_3 \quad \text{IF}_3 \\
\text{EX}_2 & \quad \text{MA}_2 \quad \text{WB}_2 \\
\text{IF}_4 & \quad \text{ID}_4 \quad \text{EX}_4 \\
\text{IF}_5 & \quad \text{ID}_5 \\
\end{align*} \]  

\( \text{stalled stages} \)
Compare the source registers of the instruction in the decode stage with the destination register of the uncommitted instructions.
Mechanics of Processor Design

• Analyze ISA
  – Determine data path requirements

• Select components of data path
  – ALU, Register file etc.

• Analyze implementation of instruction
  – Do we need to bypass, stall, speculate?
  – Assemble control signals
That’s All!

I wish you all the best 😊