CMPT 295
Unit – Processor Design & Instruction Execution
Lecture 29 – Pipelined Execution
Last Lecture - 1

- Sequential execution of machine instructions

- **Processor design #1: Sequential execution**
  - Executing 1 machine instruction per clock cycle

- Issue: Long clock cycle -> slow computer 😞 (small throughput)

- Analysis of processor instruction execution:
  - Latency (propagation delay): Time required to execute a single instruction
  - Throughput: Number of instructions executed per second - GIPS
How to improve throughput

- Divide the execution of instructions into stages: fetch, decode, execute, memory, write back
- Introduce clocked registers (pipeline registers) after each stage
- Processor design #2: Staged execution

Result: Shorter (faster) clock cycle -> faster computer 😊

Issues:
- Added clocked registers increase latency
- Stages may not have equal propagation delay
Today’s Menu

- Instruction Set Architecture (ISA)
  - Definition of ISA
  - ISA design
  - ISA evaluation
    - Improving our ISA -> Decreasing effect of von Neumann bottleneck
      - 3 Strategies

- Execution of machine instructions
  - Intro to logic design, combinational and sequential logic circuits
  - Sequential execution of machine instructions
  - **Pipelined execution** of machine instructions
Pipelining

- Example: fast food counter versus cafeteria

- Start executing a new instruction at every clock cycle
- Effect: Different stages of different instructions overlap
A closer look:

Instruction₁:

ADD r0, r5:

F
IR ← M[PC]
PC ← PC+1

D
valA ← r[0]
valB ← r[5]

E
valE ← valA + valB

M

W
r[5] ← valE
Clock cycle:

1. $F/I_1$
2. $F/I_2$
3. $F/I_3$
4. $F/I_4$
5. $F/I_5$
6. $D/I_5$

Pipelined Execution
Analysis

- **Sequential Execution – 1 stage:**
  - Latency - cost per instruction:
    - 400 ps + 20 ps = 420 ps
  - CPU throughput: 1 / 420 ps = 2.38 GIPS
  - Clock cycle: 400 ps + 20 ps = 420 ps

- **Staged Execution – 5 stages:**
  - Latency - cost per instruction:
    - \( (400/5 \text{ ps} + 20 \text{ ps})^5 = 500 \text{ ps} \)
  - CPU throughput: 1 / 500 ps = 2 GIPS
  - Clock cycle: 80ps + 20ps = 100ps

- **Pipelined Execution – 5 stages:**
  - Latency - cost per instruction: \( (400/5 \text{ ps} + 20 \text{ ps})^5 = 500 \text{ ps} \) has not changed!
  - CPU throughput: 5 / 500 ps = 10 GIPS
  - Clock cycle: 80ps + 20ps = 100ps has not changed!

**Conclusion:**
- Instruction latency: \( \uparrow \)
- CPU throughput: \( \uparrow \)
- Clock cycle: \( \downarrow \)
Limitations of pipelining - 1

1. Would doubling the number of stages (5 -> 10) double the throughput (10 GIPS -> 20 GIPS)?

- Cost of each stage
  -> 400/10 ps + 20 ps = 40 ps + 20 ps = 60 ps
- CPU throughput -> 1 / 60 ps = 16.67 GIPS
- So?
  - Nop! (not quite 20 GIPS)
Limitations of pipelining - 2

2. What if stages have different delays?
   - Sum of delays through all the stages remains the same
   - But, clock cycle limited by delay of slowest stage
Summary

- Pipelining
- Analysis of pipelined execution using best case pipeline scenario (all stages have the same propagation delay)
  - Impact on CPU throughput
- Limitations on pipelined execution
  - Limitation 1
    - Doubling the number of stages does not double the throughput
  - Limitation 2
    - Non-uniform stage delays – not every stage has same delay
    - Longest delay wins all
Next Lecture

- Instruction Set Architecture (ISA)
  - Definition of ISA
  - ISA design
  - ISA evaluation
    - Improving our ISA -> Decreasing effect of von Neumann bottleneck
      - 3 Strategies
  - Improving our ISA - Decreasing effect of von Neumann bottleneck
    - 3 Strategies

- Execution of machine instructions
  - Intro to logic design, combinational and sequential logic circuits
  - Sequential execution of machine instructions
  - Pipelined execution of machine instructions
    - Hazards