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

  ![Combinational logic circuits diagram](image)

  - 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

- Staged Execution
- Pipelined Execution

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

Instruction 1:

ADD r0, r5:

IR ← M[PC]
PC ← PC+1
valA ← r[0]
valB ← r[5]
valE ← valA + valB
r[5] ← valE

valA ← r[0]
valB ← r[5]
valA + valB
valE ← valA + valB
r[5] ← valE
Pipelined Execution

Clock cycle:

1. F/I₁
2. F/I₈ D/I₁
3. F/I₃ D/I₂ E/I₁
4. F/I₄ D/I₃ E/I₂ M/I₁
5. F/I₅ D/I₄ E/I₃ M/I₂ W/I₁
6. D/I₅ E/I₄ M/I₃ W/I₂
Analysis

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

- **Staged Execution – 5 stages:**
  - Latency - cost per instruction:
  - CPU throughput:
  - Clock cycle:

- **Pipelined Execution – 5 stages:**
  - Latency - cost per instruction:
  - CPU throughput:
  - Clock cycle:

**Conclusion:**
- Instruction latency:
- CPU throughput:
- Clock cycle:
1. Would doubling the number of stages (5 -> 10) double the throughput (10 GIPS -> 20 GIPS)?
   - Cost of each stage
     ->
   - CPU throughput
   - 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
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
  - Hazards

Next Lecture