

Study Notes: Advanced Pipelining and Modern Processor Architectures

I. Data Hazards and Data Forwarding Recap

- **Data Forwarding:** A technique to reduce stalls caused by data dependencies by forwarding data directly from the output of one instruction to the input of a dependent instruction.
- **Example:** The provided assembly code snippets illustrate the effect of data forwarding on reducing stalls. Without forwarding, significant stalls occur.

II. Loop Unrolling

- **Definition:** A compiler optimization technique that reduces loop overhead by replicating the loop body multiple times.
- **Limitation:** Compilers are limited by the number of available registers.
- **Example:** The `for` loop example demonstrates how loop unrolling can improve performance if the compiler can determine that the loop count is always even.

III. False Dependencies

- **WAR (Write After Read):** A later instruction overwrites the source of an earlier instruction.
- **WAW (Write After Write):** A later instruction overwrites the output of an earlier instruction.
- **Impact:** False dependencies limit out-of-order execution.
- **Resolution:** Register renaming.

IV. Register Renaming

- **Purpose:** Eliminates false dependencies (WAR and WAW hazards) by assigning different registers to each write operation.
- **Mechanism:** Uses a pool of physical registers to map to architectural registers.
- **Benefits:** Enables more efficient out-of-order execution.

V. Out-of-Order Execution (OoO)

- **Components:**
 - Register renaming
 - Reorder Buffer (RoB) / Instruction Queue
 - Speculative execution
- **Process:**
 1. Instructions are fetched and decoded.
 2. Registers are renamed to eliminate false dependencies.
 3. Instructions are executed out of order based on data availability and functional unit availability.
 4. The Reorder Buffer ensures instructions are committed in the correct order.

VI. Superscalar Architecture

- **Definition:** A processor architecture that can fetch, decode, and issue more than one instruction per cycle.
- **Fetch Width:** The number of instructions that can be fetched/decoded each cycle.
- **Issue Width:** The number of instructions that can be issued each cycle.
- **Theoretical CPI:** $1 / \min(\text{issue width, fetch width, decode width})$

VII. Data Hazards and Stalls

- **Data Hazards:** Occur when an instruction depends on the result of a previous instruction that is not yet available.
- **Stalls:** Pipeline stalls are introduced to resolve data hazards, but they reduce performance.
- **Compiler Optimizations:** Can help reduce data hazards, but are limited.

VIII. Modern Processor Characteristics

- Multiple-issue pipelines with multiple functional units (ALUs, Load/Store units).
- Dynamic OoO scheduling.
- Cache (high hit rate).
- Data/instruction prefetcher.
- Branch predictors (Perceptron, TAGE).

IX. Programming for Modern Processors

- Exploit instruction-level parallelism (ILP) to achieve higher instructions per cycle (IPC) or lower cycles per instruction (CPI).
- Avoid data-dependent operations (e.g., pointer chasing).
- Avoid inter-iteration dependencies (e.g., linked lists, trees).

X. Linked Lists vs. Arrays

- **Example:** Comparing array and linked list implementations for counting nodes.
- **Arrays:** Generally outperform linked lists due to better cache locality and fewer data dependencies.
- **Linked Lists:** Performance is often limited by the "critical path" of pointer chasing (e.g., `movq 8(%rdi), %rdi`).

XI. Modern Processor Examples

- **Intel Alder Lake (P-Core):** 5-issue ALU pipeline, 7-issue memory pipeline.
- **Intel Alder Lake (E-Core):** 5-issue pipeline.
- **AMD Zen 3 (RyZen 5000 Series):** 4-issue integer pipeline + 1 branch, 3-issue memory pipeline.
- **Intel Skylake:** 4-issue integer pipeline, 4-issue memory pipeline.

XII. Key Takeaways

- Data hazards and dependencies limit performance.
- Register renaming and OoO execution mitigate false dependencies.
- Superscalar architectures increase instruction throughput.
- Efficient code exploits ILP and minimizes data dependencies.
- Arrays generally outperform linked lists in modern processors.

Glossary

- **CPI (Cycles Per Instruction):** A measure of processor performance, indicating the average number of clock cycles required to execute one instruction.
- **ILP (Instruction-Level Parallelism):** The degree to which independent instructions can be executed in parallel.
- **OoO (Out-of-Order Execution):** A technique where instructions are executed in an order different from the program order to improve performance.
- **Register Renaming:** A technique used to eliminate false data dependencies by assigning different registers to each write operation.
- **Superscalar:** A processor architecture that can execute multiple instructions per clock cycle.
- **Data Forwarding:** A technique to reduce stalls caused by data dependencies by forwarding data directly from the output of one instruction to the input of a dependent instruction.
- **Reorder Buffer (ROB):** A buffer that holds instructions that have been executed out of order but not yet committed, ensuring that they are committed in the correct order.