DATE: 02/05/2024

Submitted By : Junaid Asif
Roll No : BSAI-144
Class : BSAI (IV-Sem)
Submitted To : Ms. Mehwish Zeb
Subject : Coal

Assignment # 03

Q:1 Explain difference between pipelining and parallelism process. Provide examples of how each technique improves CPU performance.

Ans Pipelining:

Pipelining involves breaking down the execution of instructions into multiple stages so that different stages of different instructions can be executed simultaneously. Each stage in the pipeline performs a different operation, and multiple instructions are overlapped in execution. Pipelining improves CPU performance by maximizing CPU utilization, as it allows multiple instructions to be in different stages of execution simultaneously. However it may suffer from pipeline hazards such as data hazards, control hazards, and structural hazards, which can reduce efficiency.

DATE: __/__/____

Parallelism:

Parallelism involves performing multiple tasks simultaneously using multiple processing units or cores. It can be classified into ILP and TLP. Parallelism improves CPU performance by increasing throughput, allowing multiple tasks to be executed concurrently. It enables faster execution of programs by dividing the workload among multiple processing units.

Example of how each technique improves CPU Performance:

Pipelining: Suppose you have a task where you need to bake a cake. Pipelining in this scenario would involving breaking down the process of baking into multiple stages, such as mixing ingredients, preparing the oven, baking the cake, and cooling it down. While the cake is being baked, you can start mixing the ingredients for the next cake, thereby overlapping the stages and reducing the overall time required to bake multiple cakes.

Parallelism: Imagine you have a team of chef in a kitchen, each specializing in a different aspects of cooking. While one chef is preparing the main course, another

DATE: __/__/____

chef can simultaneously work on appetizers, and another chef can focus on deserts. This division of labor allows multiple dishes to be prepared concurrently, reducing the overall time required to serve a complete meal.

Q:-2 Describe stages involved in instruction execution cycle of pipeline process. How does pipelining improves CPU performance?

Ans The instruction execution cycle in a pipelined processor typically consists of several stages, each responsible for a specific operation. These stages are designed to overlap in order to increase throughput and improve CPU performance. Here's a typical breakdown of the stages involved in the instruction execution cycle of a pipelined processor:

1. **Instruction Fetch:** In this stage, the processor fetches the instructions from memory using the address stored in the program counter. The instruction is then placed into an instruction register (IR) for further process.
2. **Instruction Decode:** In this stage, the processor decodes the instruction fetched in the previous stage.

DATE : __/__/____

3. **Execution:** In this stage, the actual operation specified by the instruction is executed. This could involve arithmetic or logic operations, memory accesses, or control flow changes.
4. **Memory Access:** This stage is responsible for accessing memory if needed. For instructions that involve memory operations, such as load and store instructions, data is read from or written to memory in this stage.
5. **Write Back:** In the final stage, the results of the executed instruction are written back to the appropriate register. This could be a general-purpose register or a special-purpose register, depending on the type of instruction executed.

Now, let's discuss how pipelining improves CPU performances:

1. **Increased Throughput:** Pipelining allows multiple instructions to be in different stages of execution simultaneously. As a result, the overall throughput of the CPU is increased because multiple instructions can be processed concurrently.
2. **Reduced Latency:** By breaking down the instruction execution cycle into multiple stages and overlapping them, pipelining reduces the latency of individual instructions. Even though each instruction takes multiple clock cycles to complete, the overall time taken for the entire instruction

DATE: __/__/____

Squence is reduced.

3. **Scalability:** Pipelining ~~facilities~~ facilitates the scaling of CPU performance by enabling the addition of more pipeline stages or by increasing the width of the pipline. This scalability allows for higher clock frequencies and more complex instruction execution without ~~significancy~~ significantly increasing the cycle time.

Overall, pipelining is a fundamental technique used in modern CPU design to improve performance by increasing throughput, reducing latency, and enhancing resource utilization.

Q.3 Discuss the concept of instruction level parallelism and its importance in modern CPU design. Provide examples of ILP technique used in practice.

Ans. ILP is a CPU design concept that focuses on executing multiple instructions simultaneously within a single processor core. It aims to exploit the inherent parallelism present in a programs instruction stream to improve performance.

Importance in modern CPU design:

DATE: __/__/____

1. **Performance Improvement:** ILP allows CPUs to execute multiple instructions concurrently, thereby increasing the overall throughput and performance of the processor.

2. **Utilization of CPU Resources:** ILP ensures that the CPU's resources, such as functional units and execution pipelines, are fully utilized.

3. **Compensation for Memory Latency:** ILP can help mitigate the impact of memory latency by allowing the CPU to execute independent instructions while waiting for memory access to complete.

4. **Scalability:** ILP techniques can be scaled to accommodate the increasing complexity of modern CPUs. By leveraging ILP, CPU designers can continue to improve performance without significantly increasing clock speeds or power consumption.

Example of ILP Techniques used in Practice are:

1. **Superscalar Execution:** Superscalar processors have multiple execution units within a single core, allowing them to execute multiple instructions in parallel.

DATE: __/__/____

2. **Out-of-Order Execution:** In OoOE processors, instructions are dynamically reordered at runtime to maximize parallelism and resource utilization.

3. **Speculative Execution:** Speculative execution involves predicting the outcome of branches or conditional instructions and executing instructions speculatively based on those predictions.

4. **Vectorization:** Vectorization involves executing multiple data elements of a single instruction in parallel using vector processing units.

Overall, ILP techniques play a crucial role in modern CPU design, by allowing processors to exploit parallelism at the instruction level, thereby achieving higher performance, better resources utilization, and improved scalability.