# Assignment 4: Exploring Instruction-Level Parallelism (ILP) in Modern Processors

## Objective:

This assignment aims to provide students with a practical understanding of Instruction-Level Parallelism (ILP) in computer architecture. Students will engage with theoretical underpinnings, analyze practical techniques, evaluate trade-offs, and scrutinize real-world implementations. The goal is to foster a nuanced understanding of how ILP influences processor design and performance, ultimately preparing students to contribute to future advancements in the field.

## Part 1: Understanding Instruction-Level Parallelism

### 1. Introduction to ILP:

Reading Assignment: Study the provided material on Instruction-Level Parallelism, focusing on the fundamental concepts, challenges, and techniques used to exploit ILP.

**Research Literature Review Assignment:**

**Contemporary Research**

Using available library resources to identify and read 3-5 recent peer-reviewed research papers (published within the last 5 years) that explore current challenges, novel approaches, or future directions in ILP. Focus on papers published in reputable computer architecture conferences or journals from IEEE and/or ACM.

Consider the following components for your Critical Review and Synthesis of information

Requirements: Write a comprehensive review (4-5 pages) that integrates your findings of the contemporary research.

Your review should:

* Trace the Evolution: Chart the historical development of ILP, highlighting key milestones, influential ideas, and paradigm shifts in the field.
* Analyze Core Concepts: Provide an in-depth analysis of fundamental ILP concepts, including: Parallelism detection and exploitation: How do modern processors identify and utilize potential parallelism within instruction streams?
* ILP limitations: What are the fundamental constraints on ILP, such as data dependencies, control flow dependencies, and resource limitations?
* Performance metrics: How is ILP effectiveness measured? What are the trade-offs between different metrics (e.g., throughput, latency, power consumption)?
* **Critique Current Challenges:** What are the major challenges facing ILP in contemporary processor design? (Consider issues like increasing complexity, diminishing returns from traditional techniques, power constraints, etc.)
* How are researchers addressing these challenges? What novel techniques or approaches are being explored to overcome them?
* **Synthesize Future Directions:** Based on your review, what are the most promising future directions for ILP research? What emerging trends or technologies could significantly impact the future of ILP? (Consider areas like heterogeneous architectures, specialized accelerators, machine learning-based optimizations, etc.)

## Part 2: Practical Exploration of ILP Techniques

Part 2 of this assignment focuses on understanding and applying Instruction-Level Parallelism (ILP) techniques through hands-on experimentation with the gem5 simulator. You'll build on your understanding of pipelining, branch prediction, multiple issue, and multithreading to analyze how these techniques impact performance.

**Basic Pipeline Simulation**

**gem5 Configuration:** In your gem5 configuration file:

Define a simple pipeline with distinct fetch, decode, execute, memory, and writeback stages.

Choose a workload (set of instructions or a small program) to simulate.

**Run Simulation:** Execute the simulation in gem5 and observe how instructions progress through each pipeline stage. Analyze the behavior cycle-by-cycle.

**Visualize:** Use gem5's visualization tools (e.g., the graphical pipeline viewer) to get a visual representation of the pipeline and how instructions flow through it.

**Performance Metrics**

**Data Collection:** Use gem5's built-in statistics gathering mechanisms to track:

**Instruction Throughput:** The number of instructions completed per cycle.

**Instruction Latency:** The average number of cycles it takes for an instruction to complete.

**Reporting:**

Present your findings clearly, showing the throughput and latency values you obtained. Explain any interesting patterns or trends you observe.

**Impact of Branch Prediction**

**Add Branch Prediction:** Extend your gem5 configuration to include a simple branch prediction mechanism (e.g., a static predictor or a basic dynamic predictor).

**Comparison:** Run the same workload with and without branch prediction enabled.

**Analysis:** Compare the performance metrics (throughput, latency) for both scenarios. Discuss how branch prediction affects pipeline efficiency. Explain why accurate prediction is crucial.

**Multiple Issue Simulation**

**Superscalar Configuration:** Modify your gem5 configuration to create a superscalar processor that can issue multiple instructions per cycle.

**Benchmarks:** Run a variety of benchmark programs that exercise different instruction types (integer, floating-point, memory operations, etc.).

**Performance Gains:** Analyze the performance improvement compared to the single-issue pipeline. Are there specific benchmarks where superscalar shines?

**Multithreading**

**Enable SMT:** Configure gem5 to use Simultaneous Multithreading (SMT), where multiple threads share the same processor resources.

**Resource Utilization:** Monitor how resources like the pipeline, registers, and functional units are shared among threads. Are there bottlenecks or contention points?

**Overall Throughput:** Measure the overall system throughput. Does SMT significantly improve the number of instructions completed per cycle?

**Key Questions to Consider**

Throughout your experiments, think about:

* How do different ILP techniques interact with each other?
* What are the limitations of these techniques?
* How can you balance the complexity of ILP with the performance benefits?

### Submission and Evaluation

### Submission:

Deliverables:

* Submit appropriate documents for parts 1 and parts 2.
* Include screenshots of your gem5 simulation outputs, configuration files, and any graphs or charts used to present data and a link to your github repository.

**Evaluation Criteria:**

* Research literature Review:  Appropriate and detailed Memory Hierarchy Discussion
* Programming and Development Accuracy: Correct execution of the "Hello World" program in gem5.
* Screenshots: Report accurately provides screenshots depicting output and each step.
* Documentation and APA Guidelines: Clarity and completeness of the report.
* Troubleshooting: Appropriate discussion and documentation on the ability to identify and resolve issues encountered during the process