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

**Part 1**

2024

* **name: Abdul raheman Gotori**
* **Student id: 005029919**
* **COURSE & TITLE: Computer Architecture and Design (MSCS-531-M51)**
* **DATE:** **27th october 2024**

Table of Contents

[**Part 1** 2](#_Toc180876609)

[**References** 4](#_Toc180876610)

### **Part 1**

Innovations aimed at increasing the number of instructions executed per clock cycle have been the driving force behind the evolution of ILP, which has become a fundamental component of contemporary computer architecture.  
  
The first pipelined processor, the IBM System/360 Model 91 was introduced in 1965 enabling the overlapping of multiple stages of instruction execution. In late 1960, Tomasulo's algorithm was introduced to manage out-of-order execution in the IBM System/360 Model 91. It mitigates pipeline stalls by enabling instructions to execute as soon as their operands are available, rather than in the order in which they are specified in the program.  
  
 Intel, IBM, and other companies began incorporating superscalar architectures into general-purpose CPUs in the early 1990s (e.g., Intel Pentium (1993) and IBM PowerPC).  
  
 IBM's POWER processors formalized the use of reorder buffers (ROBs) in 1992 to maintain precise exceptions in out-of-order execution. This allowed for speculative execution while preserving the program order for committed instructions.  
  
 The Alpha 21264 (1996) from Digital Equipment Corporation introduced two-level adaptive branch prediction in 1996. This made mistakes less expensive and gave very good results.  
  
 Intel's Pentium 4 processor introduced Hyper-Threading Technology in 2002. SMT enabled the simultaneous issuance of instructions by multiple threads on the same core thereby enhancing ILP by optimizing the utilization of execution units that would have otherwise been idle.  
  
 In 2003, the processor design began to transition to multicore architectures as a result of the minimal performance benefits that further ILP provided. For example, AMD and Intel began to concentrate on Thread-Level Parallelism (TLP) through multicore designs.

ILP is a potential overlap that is employed to enhance the performance of the system by utilizing pipelines in the execution of instructions. Instruction-level parallelism quantifies the number of operations that are executed concurrently. ILP is comprised of the following three primary components in all contemporary high performance CPUs  
  
- Pipelining is predicated on the notion that a single instruction can frequently require a significant amount of time to execute. It exclusively employs a particular processor region at any given moment.  
  
- Superscalar is the process of duplicating specific CPU features to accommodate multiple instructions in each pipeline stage. For example the CPU can simultaneously decode four instructions by copying the decode stage into four separate decoders.  
  
 - Out-of-order execution (OOE) allows processors to execute instructions in a different order than they are organized in memory provided that the results are accurate.

In order to overcome the declining effectiveness of traditional ILP methods, new architectural paradigms, specialized hardware and clever optimizations will become more and more important in ILP research in the future. The application of machine learning-based optimizations which could improve scheduling efficiency, prediction accuracy, and resource management to lessen pipeline stalls and dependencies, is, in my opinion, the most promising trend for ILP. For instance reinforcement learning-based schedulers and neural branch predictors have already demonstrated potential in improving ILP by facilitating more precise predictions and more intelligent scheduling decisions. Furthermore, it can be employed to accurately predict power consumption with minimal runtime overhead, which is particularly advantageous for applications where traditional simulation models are impractical due to high computational demands based on specific workloads. This technique reduces power consumption in CPU architectures, which is critical for both mobile and data center applications where energy efficiency is paramount.

### **References**

* Hennessy, J. L., & Patterson, D. A. (2017). *Computer Architecture: A Quantitative Approach* (6th ed.). Morgan Kaufmann.
* Ajay Krishna Ananda Kumar , Sami Al-Salamin, Hussam Amrouch (2022). *Machine* *Learning Based Microarchitecture*. Research Gate.
* Mark Oskin (1999). *Exploiting ILP In Page-Based Intelligent Memory*. Research Gate.
* Bilal Ali Ahmad (2020). *Spectre and Meldown attack detection using machine learning and hardware performance counters.* Research Gate.