

Lecture - 02:

Brief comparison of Computer Architecture and Computer Organization:

Aspect	Computer Architecture	Computer Organization
Definition	The design and functionality of a computer system.	How the components are connected and work together in the system.
Focus	High-level design and structure of systems.	Low-level details and how the system works.
Components	Includes CPU, memory, and input/output devices	Includes hardware like buses, registers, and control units.
Level of Abstraction	More abstract and conceptual.	More concrete and practical
Concerned With	Instruction sets, data formats, and system capabilities.	Hardware details, circuit design, and physical connections.
Examples	Processor Design and Memory hierarchy	RAM, ROM, cache memory, and how they interact
Goal	To define what the computer can do.	To define how the computer does it.

Definitions:

RISC (Reduced Instruction Set Computer):

RISC is a type of computer architecture that uses a small set of simple instructions. It focuses on executing instructions at high speed by using a fixed instruction format and optimizing the pipeline. RISC architectures are designed to perform operations quickly and efficiently, often requiring more instructions to accomplish tasks compared to CISC.

CISC (Complex Instruction Set Computer):

CISC is a type of computer architecture that has a large set of complex instructions. These instructions can execute multiple operations in a single instruction, allowing for more powerful commands. CISC architectures aim to minimize the number of instructions per program, potentially making the coding process easier but at the cost of slower execution.

Differences Between RISC and CISC

Aspect	RISC	CISC
Instruction Set	Small and simple set of instructions.	Large and complex set of instructions.
Instruction Length	Fixed-length instructions (e.g., 32 bits).	Variable-length instructions (e.g., 1 to 15 bytes).
Execution Speed	Generally faster due to simpler instructions and pipelining.	Can be slower due to complex instruction decoding.
Complexity	Simpler design; easier to optimize.	More complex design; can be harder to optimize.
Programming Style	More instructions required for the same task.	Fewer instructions needed for programming tasks.
Memory Usage	Tends to use more memory for code.	Tends to use less memory for code.
Pipeline Efficiency	Highly efficient due to simpler instructions.	Less efficient due to instruction complexity.
Examples	ARM, MIPS, PowerPC.	x86, VAX, Intel 8086.

Definition of Pipeline

Pipeline:

In computing, a pipeline is a technique used to improve the performance of a processor by allowing multiple instructions to be executed simultaneously at different stages of processing. Instead of completing one instruction before starting the next, the pipeline divides the execution process into discrete stages, such as instruction fetch, decode, execute, and write back. This means that while one instruction is being executed, another can be decoded, and a third can be fetched, increasing the overall throughput of the processor.

Key Points

- **Stages:** Typical stages in a pipeline include Instruction Fetch (IF), Instruction Decode (ID), Execute (EX), Memory Access (MEM), and Write Back (WB).
- **Parallelism:** Pipelines exploit instruction-level parallelism by overlapping the execution of multiple instructions.
- **Throughput:** The primary goal of pipelining is to increase the number of instructions processed in a given time, thus improving overall performance.

Pipelining is a fundamental concept in modern CPU design, allowing for more efficient processing and faster execution of programs.

পাইপলাইন (Pipeline):

কম্পিউটিংয়ে, পাইপলাইন একটি প্রযুক্তি যা প্রসেসরের কার্যকারিতা বাড়ানোর জন্য ব্যবহৃত হয়, যা একাধিক নির্দেশকে প্রক্রিয়াকরণের বিভিন্ন পর্যায়ে একই সাথে কার্যকর করার অনুমতি দেয়। একটি নির্দেশ সম্পূর্ণ করার আগে পরবর্তী নির্দেশ শুরু না করে, পাইপলাইন কার্যক্রমকে বিভিন্ন পর্যায়ে বিভক্ত করে, যেমন নির্দেশ সংগ্রহ (Instruction Fetch), ডিকোড (Decode), কার্যকরী (Execute), এবং লেখার জন্য পুনরায় (Write Back)। এর মানে হল যে যখন একটি নির্দেশ কার্যকর হচ্ছে, তখন আরেকটি নির্দেশ ডিকোড করা হচ্ছে এবং তৃতীয়টি সংগ্রহ করা হচ্ছে, যা প্রসেসরের সামগ্রিক ফ্রিউট বাড়ায়।

মূল পয়েন্টসমূহ:

- পর্যায়সমূহ: পাইপলাইনে সাধারণত নির্দেশ সংগ্রহ (IF), নির্দেশ ডিকোড (ID), কার্যকরী (EX), মেমরি অ্যাক্সেস (MEM), এবং লেখার জন্য পুনরায় (WB) অন্তর্ভুক্ত থাকে।
- প্যারালেলাইজম: পাইপলাইনগুলি নির্দেশ স্তরের প্যারালেলাইজমের সুবিধা গ্রহণ করে একাধিক নির্দেশের কার্যক্রমকে ওভারল্যাপ করে।
- ঋণপুট: পাইপলাইনের মূল লক্ষ্য হল নির্দিষ্ট সময়ে প্রক্রিয়াকৃত নির্দেশের সংখ্যা বাড়ানো, ফলে সামগ্রিক কার্যকারিতা উন্নত হয়।

পাইপলাইনিং আধুনিক CPU ডিজাইনের একটি মৌলিক ধারণা, যা আরও কার্যকর প্রক্রিয়াকরণ এবং প্রোগ্রামের দ্রুত কার্যকারিতার জন্য অনুমতি দেয়।

Hazards in Pipelining

Hazards in pipelining refer to situations that prevent the next instruction in the pipeline from executing in the designated clock cycle. Hazards can lead to pipeline stalls, reducing the efficiency gains achieved through pipelining. There are three main types of hazards:

1. Structural Hazards:

- Occur when hardware resources are insufficient to support all active pipeline stages.
- For example, if a single memory unit is shared between instruction fetch and data access, the pipeline may be unable to fetch a new instruction while simultaneously reading or writing data.

2. Data Hazards:

- Occur when an instruction depends on the result of a previous instruction that has not yet completed.
- There are three subtypes:
 - **Read After Write (RAW):** An instruction needs to read a value that a previous instruction has not yet written.
 - **Write After Read (WAR):** An instruction tries to write a value before a previous instruction has read it.
 - **Write After Write (WAW):** Two instructions attempt to write to the same location, causing potential data loss or incorrect results.

3. Control Hazards:

- Occur due to branch instructions that alter the flow of execution.
- If the pipeline fetches the next instruction before the branch decision is known, it may fetch the wrong instruction, causing a misprediction.

পাইপলাইনে বিপদ (Hazards in Pipelining):

পাইপলাইনে বিপদ বোঝায় এমন পরিস্থিতি যা পাইপলাইনের পরবর্তী নির্দেশটি নির্ধারিত ঘড়ির সাইকেলে কার্যকর হতে বাধা দেয়। বিপদগুলি পাইপলাইন স্টল তৈরি করতে পারে, যা পাইপলাইনিং দ্বারা অর্জিত কার্যকারিতা বৃদ্ধিকে হ্রাস করে। প্রধানত তিন ধরনের বিপদ রয়েছে:

1. স্ট্রাকচারাল বিপদ (Structural Hazards):

- যখন হার্ডওয়্যার সম্পদগুলি সব সক্রিয় পাইপলাইন পর্যায়কে সমর্থন করার জন্য যথেষ্ট নয়।
- উদাহরণস্বরূপ, যদি একটি একক মেমরি ইউনিট নির্দেশ সংগ্রহ এবং ডেটা অ্যাক্সেসের মধ্যে ভাগ করা হয়, তবে পাইপলাইন নতুন নির্দেশ সংগ্রহ করতে অক্ষম হতে পারে যখন এটি একই সময়ে ডেটা পড়া বা লেখা করছে।

2. ডেটা বিপদ (Data Hazards):

- যখন একটি নির্দেশ একটি পূর্ববর্তী নির্দেশের ফলাফলের উপর নির্ভর করে যা এখনও সম্পন্ন হয়নি।
- এর তিনটি উপধরন রয়েছে:
 - **রিড আফটার রাইট (Read After Write - RAW):** একটি নির্দেশ এমন একটি মান পড়তে চায় যা একটি পূর্ববর্তী নির্দেশ এখনও লেখেনি।

- রাইট আফটার রিড (**Write After Read - WAR**): একটি নির্দেশ এমন একটি মান লেখার চেষ্টা করে যা একটি পূর্ববর্তী নির্দেশ পড়ার আগে।
- রাইট আফটার রাইট (**Write After Write - WAW**): দুটি নির্দেশ একই স্থানে লেখার চেষ্টা করে, যা সম্ভাব্য ডেটা ক্ষতি বা ভুল ফলাফলের সৃষ্টি করতে পারে।

3. কন্ট্রোল বিপদ (**Control Hazards**):

- ব্রাঞ্চ নির্দেশগুলির কারণে ঘটে যা কার্যক্রমের প্রবাহ পরিবর্তন করে।
- যদি পাইপলাইন ব্রাঞ্চ সিদ্ধান্ত জানার আগে পরবর্তী নির্দেশটি সংগ্রহ করে, তবে এটি ভুল নির্দেশটি সংগ্রহ করতে পারে, যা একটি ভুল পূর্বাভাস সৃষ্টি করে।

Control Hazards and Their Impact on Pipeline Performance

Control hazards occur when branch instructions disrupt the normal flow of execution in a pipeline. Their impact on performance includes:

1. **Stalls**: The pipeline may need to wait to determine the branch outcome, causing delays and wasted clock cycles.
2. **Wrong Path Execution**: Speculatively executing instructions after a branch can lead to incorrect instruction execution, resulting in wasted resources.

Summary

Control hazards decrease pipeline efficiency by increasing latency and reducing throughput. Techniques like branch prediction are used to mitigate these effects.

AVAILABLE AT:

Onebyzero Edu - Organized Learning, Smooth Career

The Comprehensive Academic Study Platform for University Students in Bangladesh (www.onebyzeroedu.com)