Multicore Processor

A processor that has more than one core is called Multicore Processor. In this article we will learn about multicore processor in detail. A multicore processor is an integrated circuit for faster processing of several tasks like decreased power consumption and increased performance. it has two or more processors that read and perform program instructions.

In other words, a multicore processor comprises multiple processing units, or "Cores," each of which has the potential to do distinct tasks. For example, if you are doing many tasks at a time such as watching a movie and using other applications. one core processor will handle activities like watching a movie while the multicore processor handles other responsibilities at the same time.

History of Multicore Processors

The companies who created the chip based processors could only put one CPU on one chip. Chipmakers were able to create chip with more circuits as chip making technology also chipmakers were able to generate multi core chip with more than one processor. The first multicore CPU was invented by Kunle Olukotun in 1998 who was a professor of electrical engineering at Stanford. Multicore chips were first accessible in 2005 from Advanced Micro Devices and Intel.

Uses of Multicore Processor

Multicore processors are used in many devices like desktop, laptop, smartphone, and gaming systems. Some applications which use multicore processor are as below.

- Multicore Processor is used in high graphics games like Overwatch and Star Wars Battlefront, and other 3D games.
- The multicore processor is more appropriate used in video editing software like Adobe Photoshop, and iMovie.
- Multicore Processor is used solidworks with computer-aided design (CAD).
- Database servers also handled by multicore CPU.
- Multicore CPU is used in high network traffic.

- Embedded systems can handle by multicore processor.

Architecture of Multicore Processor

A multi-core processor's design enables communication between all existing cores, and they divide and assign all processing duties appropriately. each core's processed data is through back to the Motherboard by a single common gateway once all of the operations have been finished. In terms of total performance this technique beats a single core CPU.

Advantages of Multicore Processor

- **Performance:** A multi-core CPU can perform more work as compared to a one-core processor. so multi core processor performance is better.
- **Reliability:** The software is always assigned to different cores in multi core processor. If one piece of software fails others remain unaffected.
- **Software Interactions:** If a software is running on many cores. it will communicate with each other.
- **Multitasking:** Multi core CPU can perform multiple tasks at a same time even if many applications may be run at the same time.
- **Power Consumption:** A multi-core processor consumes less power. Only a part of the CPU that produces heat will be used. Due to low battery utilization the power consumption is automatically reduced.

Disadvantages of Multicore Processors

- **Application Speed:** A multi-core processor is designed for multitasking, its performance is not enough. when an software is processing It jump from one core to the next. so the result is the cache fills up, increasing its speed.
- **Jitter:** In a multi-core CPU, the cores increase more interference happens due to this result in excessive jitters. operating system's program performance may suffer, and failures frequency may increase .

- **Analysis:** When you are doing multiple task at once, you will need to add memory. In a multi-core CPU, this is tough to analysis it.
- **Resource Sharing:** A multi-core CPU shares multiple resources, both internal and external. Like networks, system buses, and main memory are some resources. Any software running on the same core again and again it will interrupted.

Simultaneous Multithreading

Simultaneous Multithreading (SMT) is a hardware-based technique that allows a single physical CPU core to execute multiple independent threads of execution simultaneously. By keeping multiple thread contexts in hardware, SMT lets the core issue instructions from different threads to functional units that would otherwise be idle, increasing overall processor throughput and efficiency by overlapping computation with memory latency. A common example of SMT is Intel's Hyper-Threading Technology (HTT), which enables one physical core to appear as two virtual cores to the operating system.

How SMT Works

1. **Hardware Support:**

   SMT processors include hardware to maintain the architectural state (like register files and program counters) for multiple threads.

2. **Resource Sharing:**

   While some resources are replicated (like fetch queues), others, such as physical registers and functional units, are shared among the threads.

3. **Simultaneous Instruction Issue:**

   During each clock cycle, the processor can issue instructions from different threads to the execution units, effectively utilizing parallelism.

4. **Latency Hiding:**

   When one thread stalls due to a memory access (high latency), the processor can switch to another thread and execute its instructions, keeping the core busy.

   Benefits of SMT

- **Improved Throughput:**

  More threads can be processed concurrently on a single core, leading to higher overall system performance.

- **Enhanced Efficiency:**

  SMT better utilizes the existing resources within a superscalar CPU, reducing idle time and increasing the efficiency of the processor.

- **Increased Parallelism:**

  It allows for a greater degree of both instruction-level parallelism (within a single thread) and thread-level parallelism (across multiple threads).

  Limitations

- **Resource Partitioning:**

  Because resources are shared, SMT can lead to competition for scarce resources like caches, potentially reducing the performance of individual threads.

- **Increased Complexity:**

  Implementing SMT adds complexity to the processor design, requiring careful management of thread contexts and resource allocation.