Parallelism via Multithreaded and Multicore CPUs.

By-Osarumwense Obayagbonna

Multithreading is the process whereby operating systems (OS) use context switching to give the illusion of multiple processes executing at once. On a single processor, multithreading occurs by time division multiplexing. The processor switches between different threads commonly called a context switch. Multithreading can be done in both hardware and software.

There are different types of multithreading in hardware which include Simple multithreading, Interleaved multithreading, and Simultaneous multithreading.

Simplest form of multithreading occurs when a thread runs until it is blocked by an event that causes the pipeline to stall e.g. cache miss could takes 100s of clock cycles, switching to a ready-to-go thread can cover the latency, thereby increasing throughput. Multithreading hardware allows fast switching between a stalled thread and a ready-to-go thread. A cost of this approach is that all state registers will be duplicated.

Interleaved multithreading involves the removal of all control hazards. With IMT, the next instruction in the pipeline is always the correct instruction for the other thread. In IMT, each pipeline stage must track the thread ID of instruction it is processing.

Simultaneous Multithreading (SMT) applies only to superscalar processors. Multiple instructions are issued from multiple threads every cycle unlike in the normal case were multiple instructions are issued from a single thread per cycle.

Multicore CPU design pairs two or more single-core chip designs together, and only the control and execution units are replicated. Cache, memory controller, and secondary processing units (such as FPUs) are shared. Each core may also be multithreaded. Certain advantages of this include the fact that power is reduced linearly with reduced complexity and cores may be individually power-tuned. Core clocks frequencies can also be scaled individually.

Conclusively, it should be noted that multithreaded and multicore CPUs can significantly improve processor throughput. Although multicore performance is limited by memory bandwidth, adding multithreading to each core can improve performance over multicore single-thread. Therefore, a combination of multicore and multithreaded provides the best power/performance characteristics for most general computing applications