**Module-6**

**Performance Enhancement of Processor**

It can be well anticipated that organization enhancements to the CPU can improve performance, e.g., multiple registers rather than a single accumulator, use of multi-port memory etc. The taxonomy first introduced by Flynn is still the most common way of categorizing systems with parallel processing capability. Flynn proposed the following categories of parallel computer systems: Single instruction multiple data (SIMD) system, Multiple instruction single data (MISD) system and Multiple instruction multiple data (MIMD) System.

Another organizational approach, which is quite common, is instruction pipelining. Pipelining is particularly an effective way of organizing parallel activity in a computer system. One of the major issues of pipeline is that of conditional instructions. As the outcome of conditional instructions cannot be known before actual execution, this can result in nullifying several instructions that are sequentially pre-fetched. In such a case, pipeline stalls; this is called control hazard. There are typically three types of hazards namely, Data Hazard, Control Hazard and Structural Hazard.

Some basic approaches for dealing with conditional branches are Multiple streams, Pre-fetch branch target, Loop buffer, Delayed Branching etc. Various techniques can be used to predict whether a branch will be taken or not. The most common techniques are: Predict never taken, Predict always taken, Predict by op-code, Taken/not taken switch and Branch history table.

In a multi-processor (parallel) system, communication between any two pair of nodes is required. Several architectures for such network based communication have been proposed namely, time shared bus, crossbar network, multistage network, hypercube network, mesh network etc.

In contemporary multiprocessor system, it is customary to have one or two levels of cache associated with each processor. This organization is essential to achieve high performance. Cache creates a problem, which is known as the cache coherence problem. The cache coherence problem is: Multiple copies of the same data can exist in different caches simultaneously, and if processors are allowed to update their own copies freely, an inconsistent view of memory can result. There are software and hardware approaches for maintaining cache coherence.