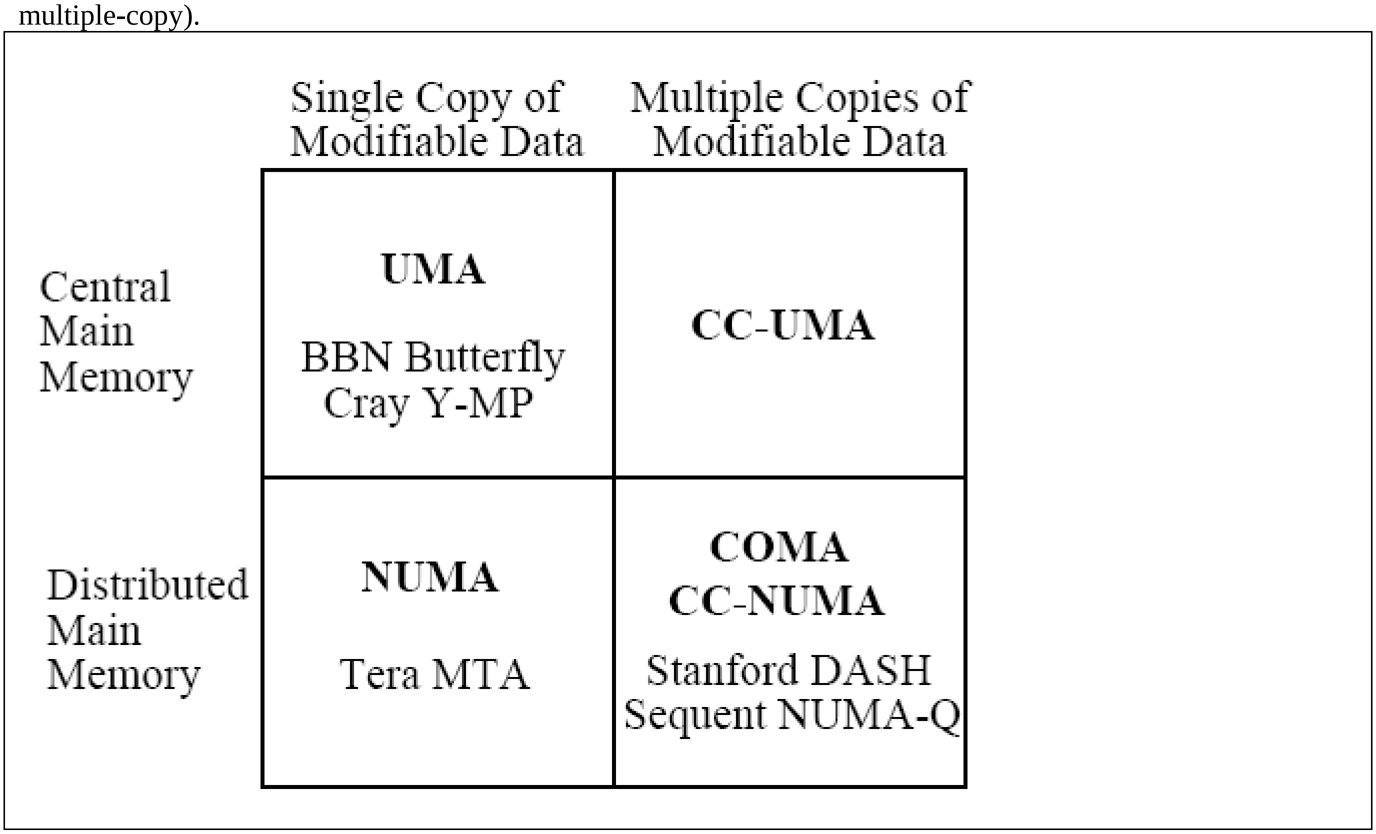
UNIT-5

PARALLEL SYSTEM IMPLEMENTATIONS

SHARED MEMORY MIMD MACHINES

Variations in Shared Memory:
Shared-memory implementations vary greatly in the hardware architecture that they use and in the programming model (logical user view) that they support With respect to hardware architecture, shared-memory implementations can be classified according to the placement of the main memory modules within the system (central or distributed) and whether or not multiple copies of modifiable data are allowed to coexist (single- or multiple-copy).



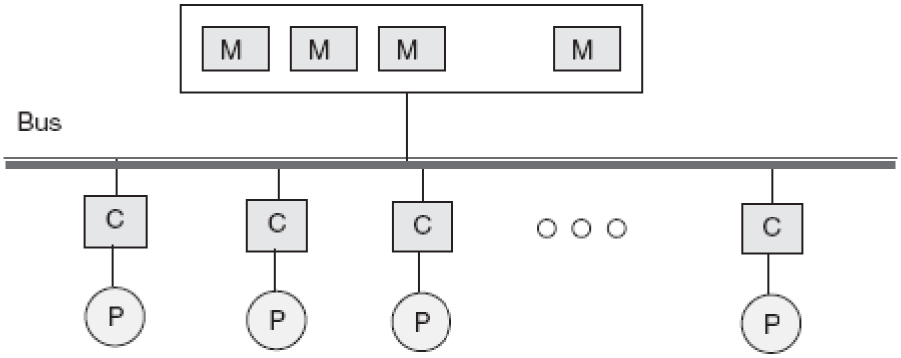
	Single Copy of Modifiable Data	Multiple Copies of Modifiable Data
Central Main Memory	**UMA** BBN Butterfly Cray Y-MP	**CC-UMA**
Distributed Main Memory	**NUMA** Tera MTA	**COMA** **CC-NUMA** Stanford DASH Sequent NUMA-Q

Uniform memory Access (UMA):

With a central main memory, access to all memory addresses takes the same amount of time, leading to the designation uniform memory access (UMA). In such machines, data distribution among the main memory modules is important only to the extent that it leads to more efficient conflict-free parallel access to data items that are likely to be needed in succession.

Because access to shared memory is balanced, these systems are also called SMP (symmetric multiprocessor) systems.

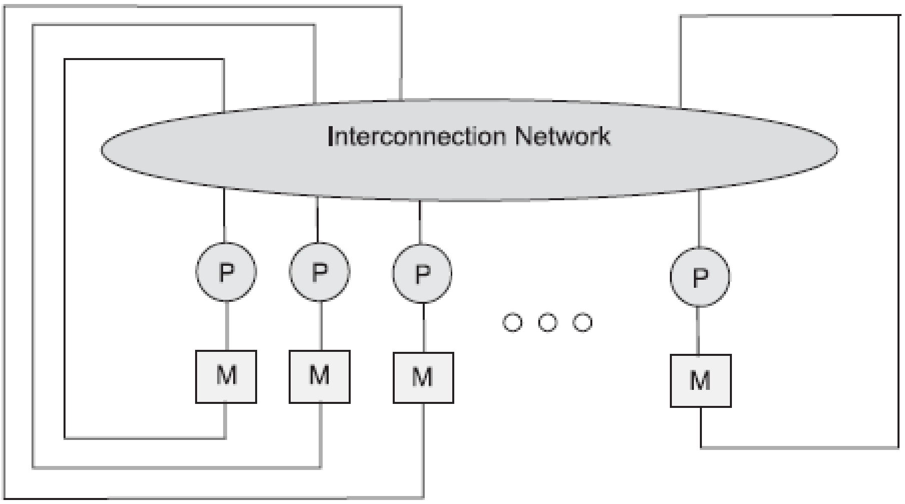
Uniform memory Access (UMA):



Non-Uniform Memory Access (NUMA):

When memory is distributed among processing nodes, access to locations in the global address space will involve different delays depending on the current location of the data. The access delay may range from tens of nanoseconds for locally available data, somewhat higher for data in nearby nodes, and perhaps approaching several microseconds for data located in distant nodes. This variance of access delay has led to the designation non-uniform memory access (NUMA).

Non-Uniform Memory Access (NUMA):



Cache-Coherent NUMA (CC-NUMA):

Another approach for dealing with slow remote accesses is to cache the needed data within individual processors (cache-coherent NUMA or CC-NUMA).

CC-NUMA machines typically enforce cache coherence via a directory-based scheme. The address placed on the memory bus of a processing node is locally translated into the identity of the home directory for the requested data. The node's local cache hierarchy holds copies of data from the local memory as well as from the memory of remote nodes. The penalty for accessing remote memory is paid only if the requested data are not found in the local node or if the enforcement of the coherence protocol (i.e., for writing to shared data) requires coordination with other nodes.

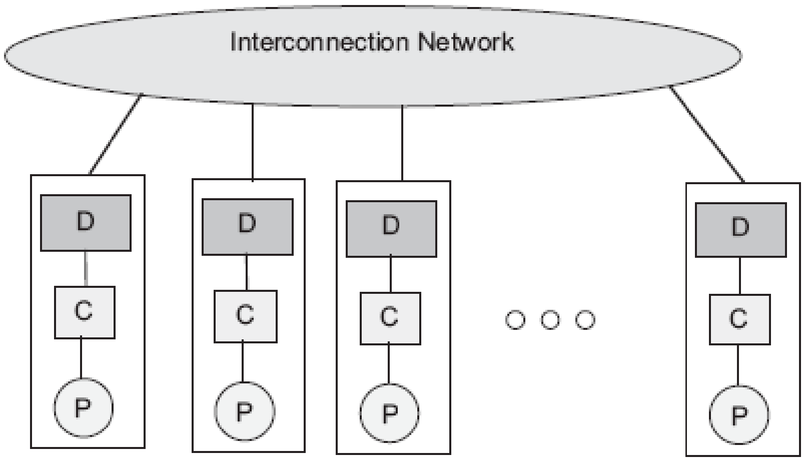
Cache-Only Memory Architecture (COMA):

Similar to the NUMA, each processor has part of the shared memory in the COMA. However, in this case the shared memory consists of cache memory.

A COMA system requires that data be migrated to the processor requesting it.

There is a cache directory (D) that helps in remote cache access.

Cache-Only Memory Architecture (COMA):



MIN-based BBN Butterfly (UMA machine):

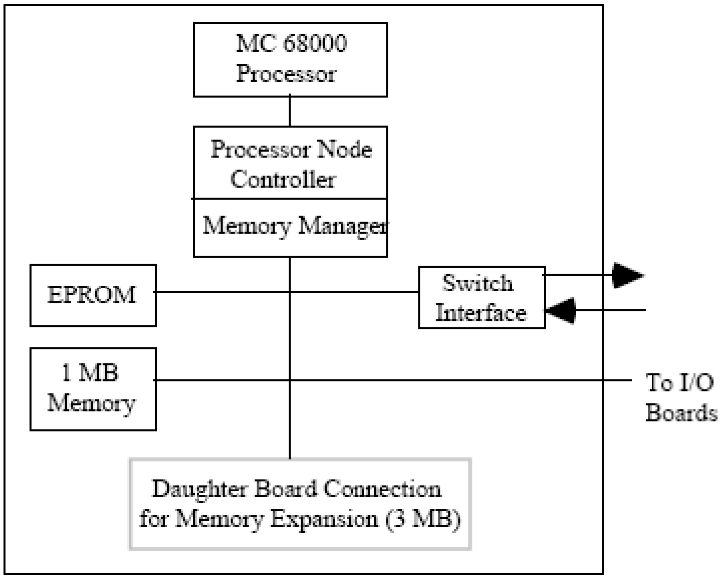
The relatively small difference between the latencies of local and remote memory accesses leads us to classify the BBN Butterfly as a UMA machine.

It is a general-purpose parallel computer that is particularly suitable for signal processing applications.

The BBN Butterfly was built of 2–256 nodes (boards), each holding an MC68000 processor with up to 4 MB of memory, interconnected by a 4-ary wrapped butterfly network.

Typical memory referencing instructions took 2 µs to execute when they accessed local memory, while remote accesses required 6 µs.

Structure of an individual node in a BBN butterfly machine:

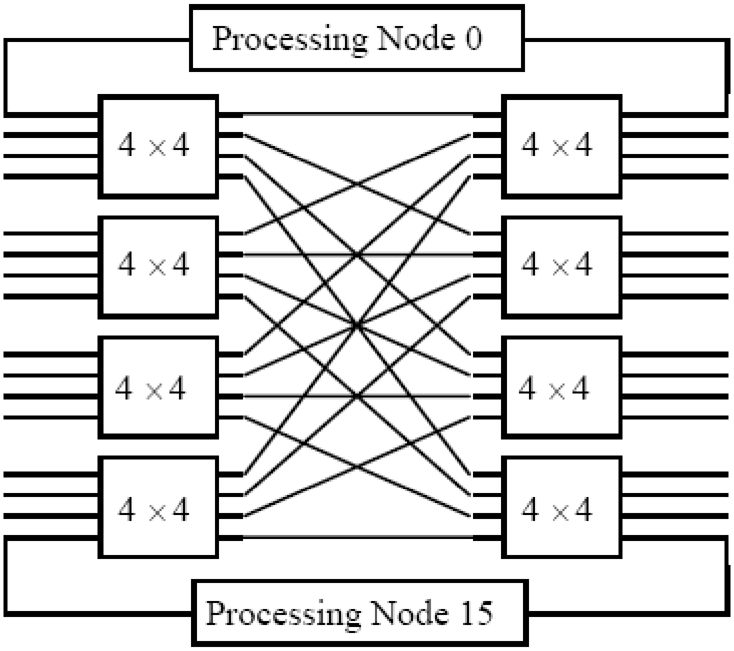


Every individual node in a BBN butterfly machine consists of a microcoded processor node controller (PNC) which is responsible for initiating all messages sent over the switch and for receiving messages from it.

It also handles all memory access requests, using the memory management unit for translating virtual addresses to physical addresses.

PNC also performs the functionality of the main processor in performing operations needed for parallel processing (queuing, scheduling, etc.).

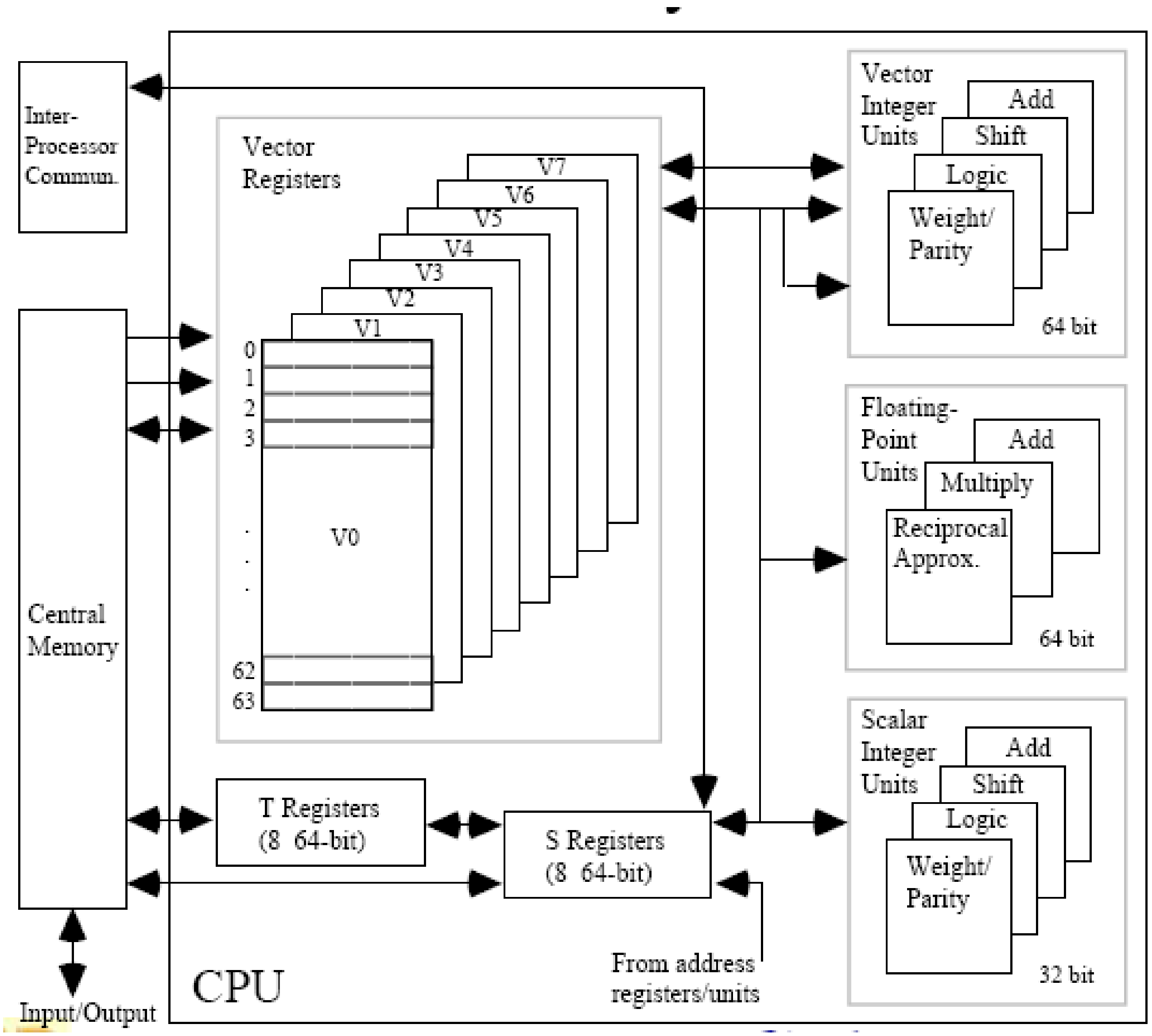
16-node version of the multistage interconnection network of the BBN Butterfly:



VECTOR-PARALLEL CRAY Y-MP

- The Cray Y-MP series of vector-parallel computers were introduced in the late 1980s.
- The Cray Y-MP consisted of a relatively small number (up to eight) of very powerful vector processors. A vector processor essentially executes one instruction on a large number of data items with a great deal of overlap. Such vector processors can thus be viewed as time-multiplexed implementations of SIMD parallel processing. With this view, the Cray Y-MP should be classified as hybrid SIMD/MIMD machine.
- Each processor has four ports to access central memory, with each port capable of delivering 128 bits per clock cycle (4 ns). Thus, a CPU can fetch two operands (a vector element and a scalar), store one value, and perform I/O simultaneously.
- Vector integer operations are performed by separate function units for add/subtract, shift, logic, and bit-counting (e.g., determining the weight or parity of a word).
- Vector floating-point operations are performed by separate function units for add/subtract, multiply, and reciprocal approximation.
- Scalar integer operations are performed by separate integer function units for addition/subtraction, shift, logic, and bit-counting.
- A key component of Cray Y-MP is its processor-to-memory interconnection network This is a multistage crossbar network built of 4 × 4 and 8 × 8 crossbar switches and 1 × 8 demultiplexers. The network uses circuit switching.

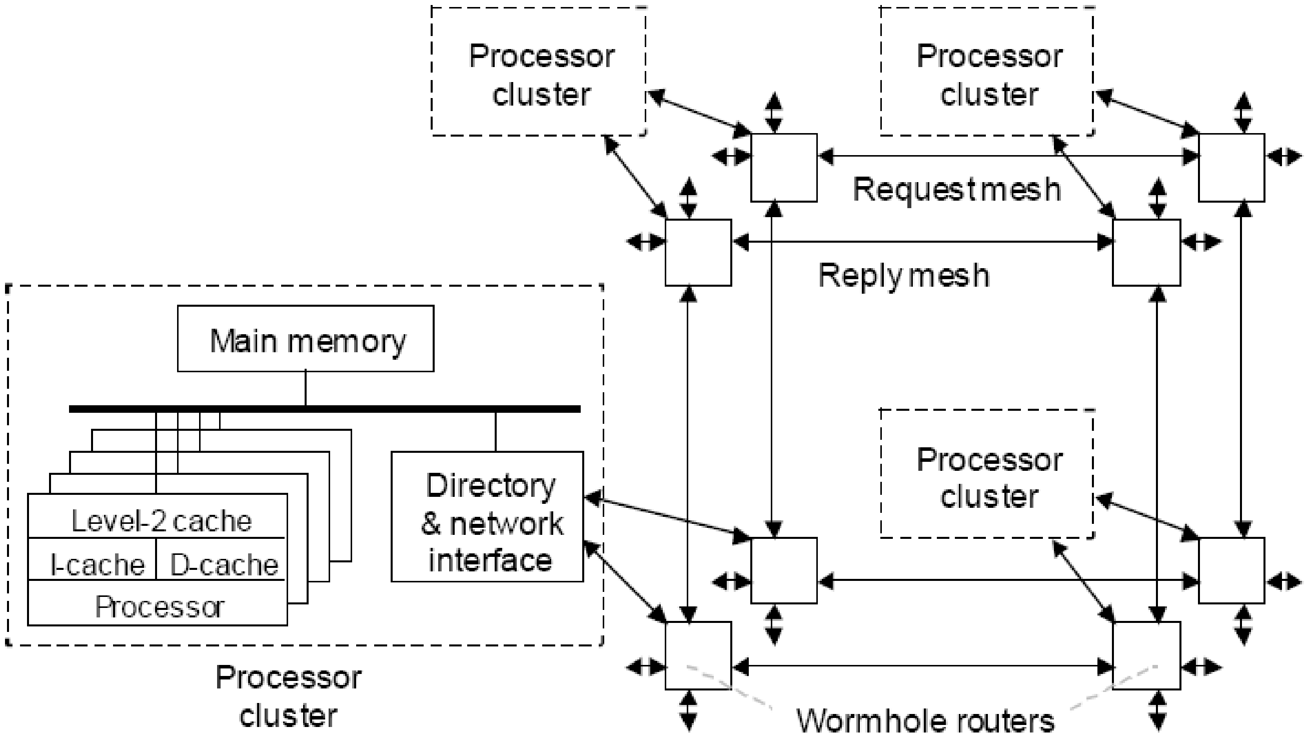
Key elements of Vector-Parallel Cray Y-MP:



CC-NUMA Stanford DASH

- Stanford University's directory architecture for shared memory (DASH) project of the early 1990s had the goal of building an experimental cache-coherent multiprocessor.
- DASH can be classified as a cache-coherent NUMA (CC-NUMA) architecture.
- It has a two-level processor-to-memory interconnection structure and a corresponding two-level cache coherence scheme.
- Within a cluster of 4–16 processors, access to main memory occurs via a shared bus. Each processor in a cluster has a private instruction cache, a separate data cache, and a Level-2 cache.
- The clusters are interconnected by a pair of wormhole-routed 2D mesh networks: a request mesh, which carries remote memory access requests, and a reply mesh, which routes data and acknowledgments back to the requesting cluster.
- Each of the 16 clusters in the DASH prototype is a four-processor Silicon Graphics 4D/340 Powerstation symmetric multiprocessor based on the MIPS R3000 chip.

CC-NUMA Stanford DASH:



MESSAGE-PASSING MIMD MACHINES

Three categories of message-passing MIMD computers are as follows:

- **Coarse-grain parallelism.** Processing nodes are complete (large, multiboard) computers that work on large sized sub problems and communicate or synchronize with each other less often.
- **Medium-grain parallelism.** Processing nodes might be based on standard micros that execute smaller chunks of the application program (e.g., subtasks, processes, threads) and communicate or synchronize more frequently.
- **Fine-grain parallelism.** Processing nodes can be custom-built processing elements (perhaps with multiple PEs fitting on one chip) that execute small pieces of the application and need constant communication or synchronization.

Interconnection networks for message passing:

Interconnection networks for message passing are of three basic types:

1. **Shared-medium networks.**
    - Only one of the units linked to a shared-medium network is allowed to use it at any given time.
    - Nodes connected to the network typically have request, drive, and receive circuits.
    - The two most commonly used shared-medium networks are backplane buses and local area networks (LANs).
    - In bus transactions that involve a request and a response, a split-transaction protocol is often used so that other nodes can use the bus while the request of one node is being processed at the other end.
    - For LANs, the nodes can detect the idle/busy state of the shared medium, transmitting when they observe the idle state, and considering the transmission as having failed when they detect a "collision." Token-based protocols, which implement some form of rotating priority, are also used.
2. **Router-based networks.**
    - Such networks, also known as direct networks, are based on each node (with one or more processors) having a dedicated router that is linked directly to one or more other routers.
    - The local node(s) connected to the router inject messages into the network through the injection channel and remove incoming messages through the ejection channel.
    - The link controllers handle interfacing considerations of the physical channels.
    - The queues hold messages that cannot be forwarded because of contention for the output links.
    - Various switching strategies (e.g., packet or wormhole) and routing algorithms (e.g., tag-based or use of routing tables) can be implemented in the router.
3. **Switch-based networks**.
    - Such networks, also known as indirect networks which are based on crossbars or regularly interconnected (multistage) networks of simpler switches.
    - Typically, the communication path between any two nodes goes through one or more switches.
    - The path to be taken by a message is either predetermined at the source node and included as part of the message header or else it is computed on the fly at intermediate nodes based on the source and destination addresses.
    - Switch-based networks can be classified as unidirectional or bidirectional. In unidirectional networks, each switch port is either input or output, whereas in bidirectional networks, ports can be used for either input or output.

DATA-PARALLEL SIMD MACHINES

Data-parallel SIMD machined are built using Associative Processors (APs). These APs work together and create a parallel processing architecture.

The APs can be classified in four categories:

1. **Fully parallel (word-parallel, bit-parallel)** APs have comparison logic associated with each bit of stored data. In simple exact-match searches, the logic associated with each bit generates a local match or mismatch signal. These local signals are then combined to produce the cell match or mismatch result.
2. **Bit-serial (word-parallel, bit-serial)** systems process an entire bit-slice of data, containing 1 bit of every word, simultaneously, but go through multiple bits of the search field sequentially.
3. **Word-serial (word-serial, bit parallel)** APs based on electronic circulating memories represent the hardware counterparts of programmed linear search.
4. **Block-oriented (block-parallel, word-serial, bit/byte-serial)** systems represent a combination of bit-serial and word-serial systems.

PROCESSOR AND MEMORY TECHNOLOGIES

Over the past two decades, microprocessor clock rates have improved by a factor of 100.

Along with speed, the functionality of microprocessors has also improved drastically. This is a direct result of the larger number of transistors that can be accommodated on one chip.

The Intel P6 has

- A 32-bit architecture,
- Internally using a 64-bit data bus,
- 36-bit addresses,
- An 86-bit floating-point format.
- It is a superscalar and superpipelined system
- It has multiprocessing capability up to four processors that can operate at 150–200 MHz, and has 21M transistors.