1 Introduction

Data-intensive applications such as transaction processing and information retrieval, data mining and analysis and multimedia services have provided a new challenge for the modern generation of parallel platforms. Emerging areas such as computational biology and nanotechnology have implications for algorithms and systems development, while changes in architectures, programming models and applications have implications for how parallel platforms are made available to users in the form of grid-based services.

High performance may come from fast dense circuitry, packaging technology, and parallelism. Parallel processors are computer systems consisting of multiple processing units connected via some interconnection network plus the software needed to make the processing units work together. There are two major factors used to categorize such systems: the processing units themselves, and the interconnection network that ties them together.

- *Uniprocessor* – Single processor supercomputers have achieved great speeds and have been pushing hardware technology to the physical limit of chip manufacturing.
  - Physical and architectural bounds (Lithography, μm size, destructive quantum effects.
  - Proposed solutions are maskless lithography process and nanoimprint lithography for the semiconductor).
  - While clock rates of high-end processors have increased at roughly 40% per year over the past decade, DRAM access times have only improved at the rate of roughly 10% per year over this interval (presents a tremendous performance bottleneck). This growing mismatch between processor speed and DRAM latency is typically bridged by a hierarchy of successively faster memory devices called caches that rely on locality of data reference to deliver higher memory system performance. In addition to the latency, the net effective bandwidth between DRAM and the processor poses other problems for sustained computation rates.
  - Uniprocessor systems can achieve to a limited computational power and not capable of delivering solutions to some problems in reasonable time.
- *Multiprocessor* – Multiple processors cooperate to jointly execute a single computational task in order to speed up its execution.
- New issues arise;

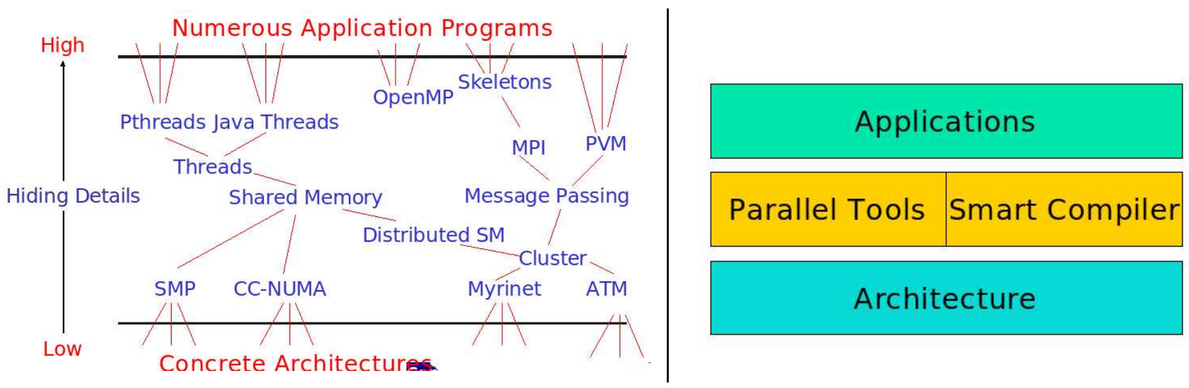


Figure 1: View of the Field and Abstraction Layers

- Multiple threads of control vs. single thread of control
- Partitioning for concurrent execution
- Task Scheduling
- Synchronization
- Performance

- Past Trends in Parallel Architecture (inside the box)
  - Completely custom designed components (processors, memory, interconnects, I/O). The first three are the major components for the aspects of the parallel computation.
    * Longer R&D time (2-3 years).
    * Expensive systems.
    * Quickly becoming outdated.
  - While parallel computing, in the form of internally linked processors, was the main form of parallelism, advances in computer networks has created a new type of parallelism in the form of networked autonomous computers.

- New Trends in Parallel Architecture (outside the box)
  - Instead of putting everything in a single box and *tightly couple* processors to memory, the Internet achieved a kind of parallelism by *loosely* connecting everything outside of the box.
  - Network of PCs and workstations connected via LAN or WAN forms a Parallel System. Compete favorably (cost/performance).

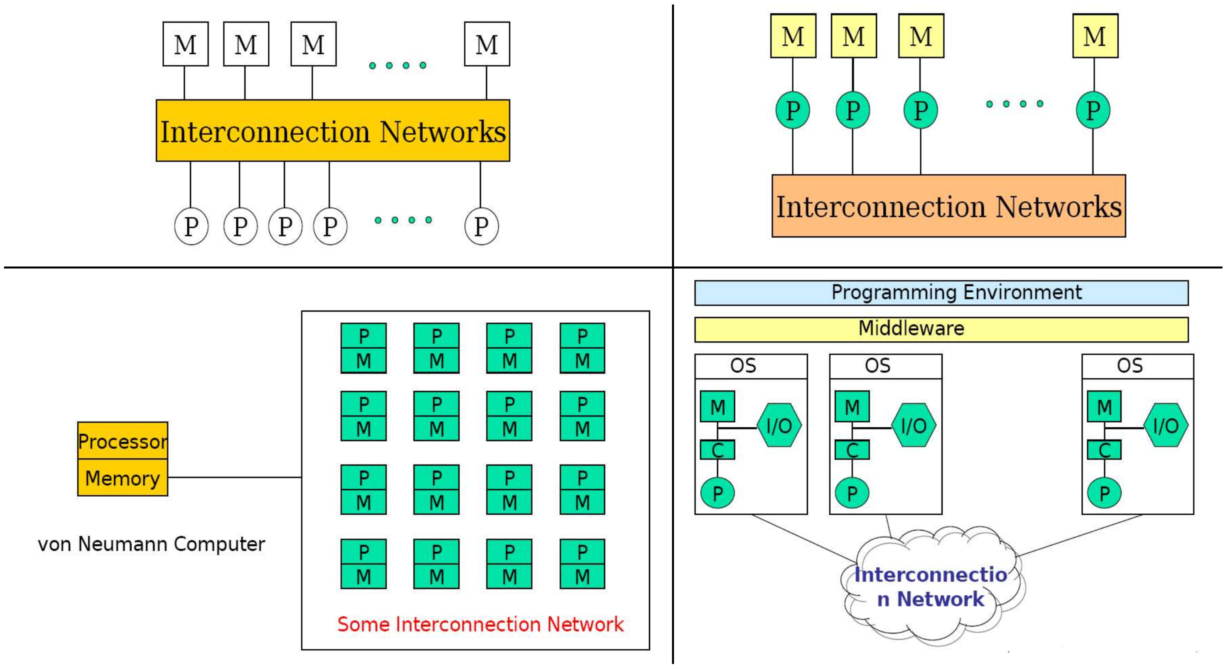
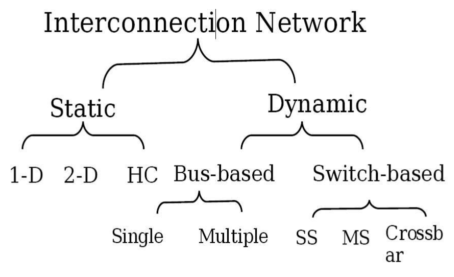
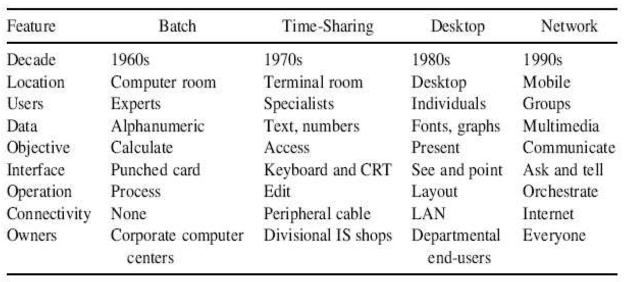


Figure 2: MIMD Shared Memory, MIMD Distributed Memory, SIMD Distributed Computers, and Clusters.

- Utilize unused cycles of systems sitting idle.

- Parallel and Distributed Computers. The processing units can communicate and interact with each other using either shared memory or message passing methods. The interconnection network for shared memory systems can be classified as bus-based versus switch-based.
  - MIMD Shared Memory
  - Bus based
  - Switch based
  - CC-NUMA
  - MIMD Distributed Memory
  - SIMD Computers
  - Clusters
  - Grid Computing
    - Grids are geographically distributed platforms for computation.





Feature	Batch	Time-Sharing	Desktop	Network
Decade	1960s	1970s	1980s	1990s
Location	Computer room	Terminal room	Desktop	Mobile
Users	Experts	Specialists	Individuals	Groups
Data	Alphanumeric	Text, numbers	Fonts, graphs	Multimedia
Objective	Calculate	Access	Present	Communicate
Interface	Punched card	Keyboard and CRT	See and point	Ask and tell
Operation	Process	Edit	Layout	Orchestrate
Connectivity	None	Peripheral cable	LAN	Internet
Owners	Corporate computer centers	Divisional IS shops	Departmental end-users	Everyone

Figure 3: Interconnection Network Taxonomy and Four Decades of Computing.

* They provide dependable, consistent, general, and inexpensive access to high end computational capabilities.

- In message passing systems, the interconnection network is divided into static and dynamic.
  - Static connections have a fixed topology that does not change while programs are running.
  - Dynamic connections create links on the fly as the program executes.

1.1 Four Decades of Computing

Most computer scientists agree that there have been four distinct paradigms or eras of computing. These are: batch, time-sharing, desktop, and network.

1. Batch Era
2. Time-Sharing Era
3. Desktop Era
4. Network Era. They can generally be classified into two main categories: (1) shared memory, and (2) distributed memory systems. The number of processors in a single machine ranged from several in a shared memory computer to hundreds of thousands in a massively parallel system. Examples of parallel computers during this era include Sequent Symmetry, Intel iPSC, nCUBE, Intel Paragon, Thinking Machines (CM-2, CM-5), MsPar (MP), Fujitsu (VPP500), and others.
5. Current Trends: Clusters, Grids.

1.2 Flynn's Taxonomy of Computer Architecture

- The most popular taxonomy of computer architecture was defined by Flynn in 1966. Flynn's classification scheme is based on the notion of a stream of information.

  - Two types of information flow into a processor: instructions and data.
  - The instruction stream is defined as the sequence of instructions performed by the processing unit.
  - The data stream is defined as the data traffic exchanged between the memory and the processing unit.

  According to Flynn's classification, either of the instruction or data streams can be single or multiple. Computer architecture can be classified into the following four distinct categories:

  1. single instruction single data streams (SISD)
  2. single instruction multiple data streams (SIMD)
  3. multiple instruction single data streams (MISD)
  4. multiple instruction multiple data streams (MIMD).

- Parallel computers are either SIMD or MIMD.

  - When there is only one control unit and all processors execute the same instruction in a synchronized fashion, the parallel machine is classified as SIMD.
  - In a MIMD machine, each processor has its own control unit and can execute different instructions on different data.
  - In the MISD category, the same stream of data flows through a linear array of processors executing different instruction streams. In practice, there is no viable MISD machine; however, some authors have considered pipelined machines as examples for MISD.

1.3 SIMD Architecture

- The SIMD model of parallel computing consists of two parts: a front-end computer of the usual von Neumann style, and a processor array as shown in Fig. 2c.

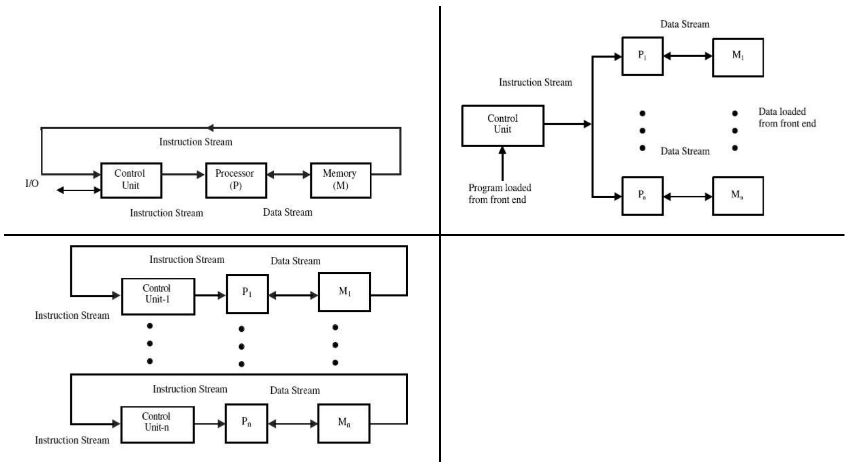


Figure 4: SISD, SIMD, amd MIMD Architectures.

- Each processor in the array has a small amount of local memory where the distributed data resides while it is being processed in parallel.

- The processor array is connected to the memory bus of the front end so that the front end can randomly access the local processor memories as if it were another memory.

- The application program is executed by the front end in the usual serial way, but issues commands to the processor array to carry out SIMD operations in parallel.

- The similarity between serial and data parallel programming is one of the strong points of *data* parallelism.

- Synchronization is made irrelevant by the lock-step synchronization of the processors. Processors either do nothing or exactly the same operations at the same time.

- In SIMD architecture, parallelism is exploited by applying simultaneous operations across large sets of data.

- There are two main configurations that have been used in SIMD machines (see Fig. 5).

    1. In the first scheme, each processor has its own local memory. Processors can communicate with each other through the interconnection network.

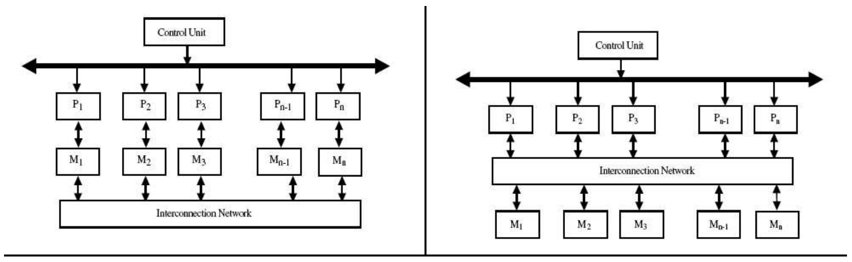


Figure 5: Two SIMD Schemes.

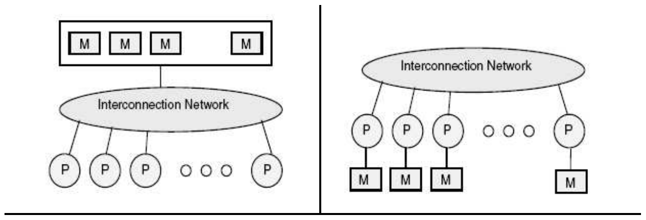


Figure 6: Two MIMD Categories; Shared Memory and Message Passing MIMD Architectures.

  - If the interconnection network does not provide direct connection between a given pair of processors, then this pair can exchange data via an intermediate processor.
  - The ILLIAC IV used such an interconnection scheme. The interconnection network in the ILLIAC IV allowed each processor to communicate directly with four neighboring processors in an 8×8 matrix pattern such that the i th processor can communicate directly with the $(i-1)^{th},(i+1)^{th},(i-8)^{th}, and(i+8)^{th}$ processors.

2. In the second SIMD scheme, processors and memory modules communicate with each other via the interconnection network.
   - Two processors can transfer data between each other via intermediate memory module(s) or possibly via intermediate processor(s). The BSP (Burroughs' Scientific Processor) used the second SIMD scheme.

1.4 MIMD Architecture

- Multiple instruction multiple data streams (MIMD) parallel architectures are made of multiple processors and multiple memory modules connected together via some interconnection network.

- They fall into two broad categories: shared memory or message passing. Figure 6 illustrates the general architecture of these two categories.

- Processors exchange information through their **central shared memory** in shared memory systems, and exchange information through their **interconnection network** in message passing systems.

    1. A shared memory system typically accomplishes interprocessor coordination through a global memory shared by all processors.
        - The bus/cache architecture facilitates the need for expensive multi-ported memories and interface circuitry as well as the need to adopt a message-passing paradigm when developing application software.
        - Because access to shared memory is balanced, these systems are also called SMP (symmetric multiprocessor) systems.
        - Commercial examples of SMPs are Sequent Computer's Balance and Symmetry, Sun Microsystems multiprocessor servers, and Silicon Graphics Inc. multiprocessor servers.
    2. A message passing system (also referred to as distributed memory) typically combines the local memory and processor at each node of the interconnection network.
        - There is no global memory, so it is necessary to move data from one local memory to another by means of message passing. This is typically done by a Send/Receive pair of commands, which must be written into the application software by a programmer (data copying and dealing with consistency issues).
        - Commercial examples of message passing architectures were the nCUBE, iPSC/2, and various Transputer-based systems. These systems eventually gave way to Internet connected systems whereby the processor/memory nodes were either Internet servers or clients on individuals' desktop.

- It was also apparent that distributed memory is the only way efficiently to increase the number of processors managed by a parallel and distributed system. If scalability to larger and larger systems (as measured by the number of processors) was to continue, systems had to use distributed memory techniques.

- These two forces created a conflict: programming in the shared memory model was easier, and designing systems in the message passing model provided scalability.
- The distributed-shared memory (DSM) architecture began to appear in systems like the SGI Origin2000, and others. In such systems, memory is physically distributed; for example, the hardware architecture follows the message passing school of design, but the programming model follows the shared memory school of thought.
- As far as a programmer is concerned, the architecture looks and behaves like a shared memory machine, but a message passing architecture lives underneath the software. Thus, the DSM machine is a *hybrid* that takes advantage of both design schools.

1.4.1 Shared Memory Organization

- Each processor may have registers, buffers, caches, and local memory banks as additional memory resources.
- A number of basic issues in the design of shared memory systems have to be taken into consideration. These include access control, synchronization, protection, and security.
  - Access control determines which process accesses are possible to which resources.
  - Synchronization constraints limit the time of accesses from sharing processes to shared resources.
  - Protection is a system feature that prevents processes from making arbitrary access to resources belonging to other processes.
  - Sharing and protection are incompatible; sharing allows access, whereas protection restricts it.
- The simplest shared memory system consists of one memory module that can be accessed from two processors. Requests arrive at the memory module through its two ports.
- Depending on the interconnection network, a shared memory system leads to systems can be classified as: uniform memory access (UMA), nonuniform memory access (NUMA), and cache-only memory architecture (COMA).

- In the UMA system, a shared memory is accessible by all processors through an interconnection network in the same way a single processor accesses its memory.
    * Therefore, all processors have equal access time to any memory location.
    * The interconnection network used in the UMA can be a single bus, multiple buses, a crossbar, or a multi-port memory.
- In the NUMA system, each processor has part of the shared memory attached.
    * The memory has a single address space. Therefore, any processor could access any memory location directly using its real address.
    * However, the access time to modules depends on the distance to the processor. This results in a nonuniform memory access time.
- 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.

1.4.2 Message Passing Organization

- Message passing systems are a class of multiprocessors in which each processor has access to its own local memory.
- Unlike shared memory systems, communications in message passing systems are performed via send and receive operations.
- A node in such a system consists of a processor and its local memory.
- Nodes are typically able to store messages in buffers (temporary memory locations where messages wait until they can be sent or received), and perform send/receive operations at the same time as processing.
- The processing units of a message passing system may be connected in a variety of ways ranging from architecture-specific interconnection structures to geographically dispersed networks.
    - Of importance are **hypercube** networks, which have received special attention for many years.

  - The nearest neighbor two-dimensional and three-dimensional mesh networks have been used in message passing systems as well.

- Two important design factors must be considered in designing interconnection networks for message passing systems. These are the link bandwidth and the network latency.

  1. The link bandwidth is defined as the number of bits that can be transmitted per unit time (bits/s).
  2. The network latency is defined as the time to complete a message transfer.

1.5 Interconnection Networks

Multiprocessors interconnection networks (INs) can be classified based on a number of criteria.

1. *Mode of Operation.* According to the mode of operation, INs are classified as *synchronous* versus *asynchronous.*

   - In synchronous mode of operation, a single global clock is used by all components in the system such that the whole system is operating in a lock-step manner.
   - Asynchronous mode of operation, on the other hand, does not require a global clock. Handshaking signals are used instead in order to coordinate the operation of asynchronous systems.
   - While synchronous systems tend to be slower compared to asynchronous systems, they are race and hazard-free.

2. *Control Strategy.* According to the control strategy, INs can be classified as *centralized* versus *decentralized.*

   - In centralized control systems, a single central control unit is used to oversee and control the operation of the components of the system. The function and reliability of the central control unit can become the bottleneck.
   - In decentralized control, the control function is distributed among different components in the system.
   - While the crossbar is a centralized system, the multistage interconnection networks are decentralized.

3. *Switching Techniques.* Interconnection networks can be classified according to the switching mechanism as *circuit* versus *packet* switching networks.

    - In the circuit switching mechanism, a complete path has to be established prior to the start of communication between a source and a destination. The established path will remain in existence during the whole communication period.
    - In a packet switching mechanism, communication between a source and destination takes place via messages that are divided into smaller entities, called packets. On their way to the destination, packets can be sent from a node to another in a store-and-forward manner until they reach their destination.
    - While packet switching tends to use the network resources more efficiently compared to circuit switching, it suffers from variable packet delays.

4. *Topology.*An **interconnection network topology** is a mapping function from the set of processors and memories onto the same set of processors and memories.

    -
    - A fully connected topology, for example, is a mapping in which each processor is connected to all other processors in the computer.
    - A ring topology is a mapping that connects processor k to its neighbors, processors $(k-1)$ and $(k+1)$.
    - In general, interconnection networks can be classified as *static* versus *dynamic* networks.
    - In static networks, direct fixed links are established among nodes to form a fixed network, while in dynamic networks, connections are established as needed.
    - *Shared memory systems* can be designed using *bus-based* or *switch-based* INs (see Fig. 7).
        - The simplest IN for shared memory systems is the bus. However, the bus may get saturated if multiple processors are trying to access the shared memory (via the bus) simultaneously.
        - A typical bus-based design uses caches to solve the bus contention problem.

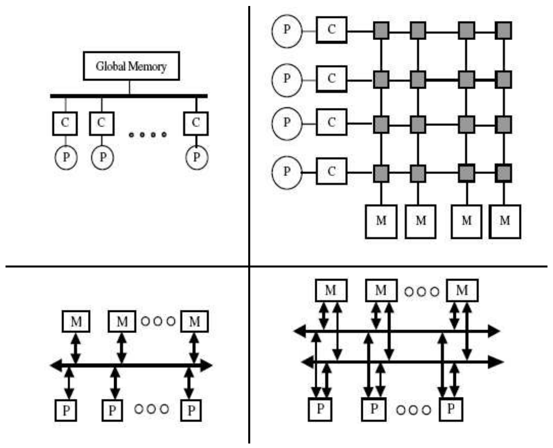


Figure 7: Bus Based and Switch Based Shared Memory INs, Single Bus Based and Multiple Bus Based Shared Memory INs.

- Other shared memory designs rely on switches for interconnection. For example, a crossbar switch can be used to connect multiple processors to multiple memory modules.

- *Message passing* INs can be divided into *static* and *dynamic*.
  - Static networks form all connections when the system is designed rather than when the connection is needed.
  - Dynamic INs establish a connection between two or more nodes on the fly as messages are routed along the links.
  - The number of hops in a path from source to destination node is equal to the number of point-to-point links a message must traverse to reach its destination.
  - The ultimate performance of an interconnection network is greatly influenced by the number of hops taken to traverse the network.
  - Figure 8 shows a number of popular static topologies.

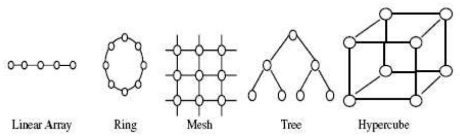


Figure 8: Examples of Static Topologies.

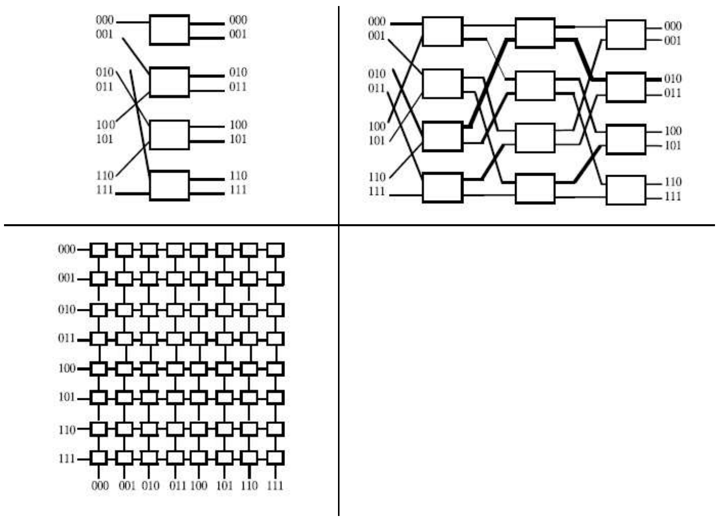
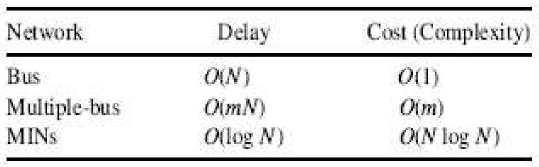
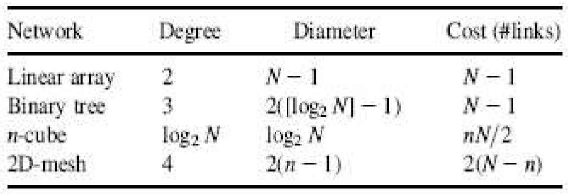


Figure 9: Single-Stage, Multi-Stage and Crossbar Switch Dynamics INs.

- Figure 9 shows examples of dynamic networks. The single-stage interconnection network of Fig. 9a is a simple dynamic network that connects each of the inputs on the left side to some, but not all, outputs on the right side through a single layer of binary switches represented by the rectangles.
- The binary switches can direct the message on the left-side input to one of two possible outputs on the right side.
- If we cascade enough single-stage networks together, they form a completely connected multistage interconnection network (MIN), as shown in Fig. 9b.
- These are dynamic INs because the connection is made on the fly, as needed.
- For example, to connect source 111 to destination 001 in the network, the switches in the first and second stage must be set to connect to the upper output port, while the switch at the third stage must be set to connect to the lower output port (001).
-
- Similarly, the crossbar switch of Figure 9c provides a path from any input or source to any other output or destination by simply selecting a direction on the fly.
- To connect row 111 to column 001 requires only one binary switch at the intersection of the 111 input line and 011 output



Network	Delay	Cost (Complexity)
Bus	$O(N)$	$O(1)$
Multiple-bus	$O(mN)$	$O(m)$
MINs	$O(\log N)$	$O(N \log N)$



Network	Degree	Diameter	Cost (#links)
Linear array	2	$N-1$	$N-1$
Binary tree	3	$2(\lceil \log_2 N \rceil - 1)$	$N-1$
n-cube	$\log_2 N$	$\log_2 N$	$nN/2$
2D-mesh	4	$2(n-1)$	$2(N-n)$

Figure 10: Performance Comparisons of Some Dynamics INs and Performance Characteristics of Static INs.

line to be set.

- The crossbar switch clearly uses more binary switching components; for example, N^2 components are needed to connect $N \times N$ source/destination pairs. The MIN, on the other hand, connects $N \times N$ pairs with $N/(2 * (logN))$ components.
- The major advantage of the crossbar switch is its potential for speed. In one clock, a connection can be made between source and destination. The diameter of the crossbar is one. (Diameter, D, of a network having N nodes is defined as the maximum shortest paths between any two nodes in the network.)
- The MIN, on the other hand requires $logN$ clocks to make a connection. The diameter of the MIN is therefore $logN$.
- If two pairs attempt to communicate at the same time along a shared path, one pair must wait for the other. This is called **blocking**, and such MINs are called blocking networks.
- Figure 10a shows a performance comparison among a number of different dynamic INs. Figure 10b shows a performance comparison among a number of static INs.