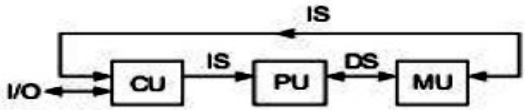
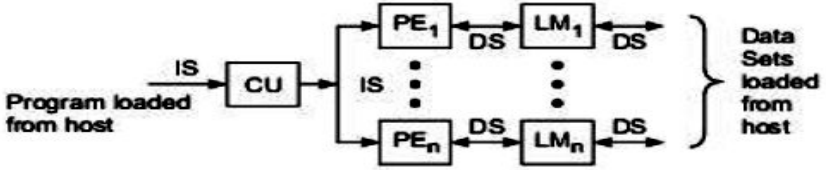
Flynn's Classification

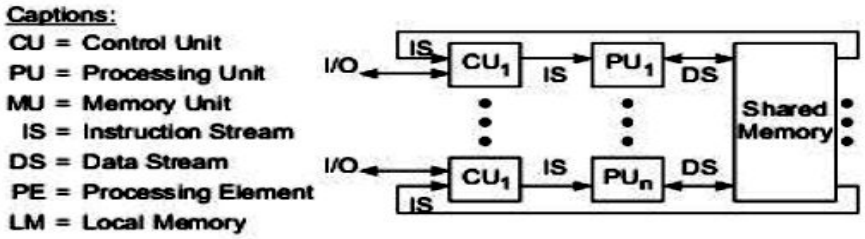


(a) SISD uniprocessor architecture

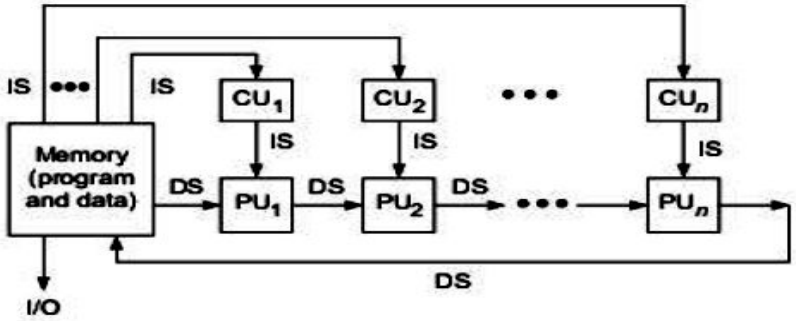


(b) SIMD architecture (with distributed memory)

- Of the four machine models, most parallel computers built in the past assumed the MIMD model for general purpose computations.
- The SIMD and MISD models are more suitable for special-purpose computations.
- For this reason, MIMD is the most popular model, SIMD next, and MISD the least popular model being applied in commercial machines.



(c) MIMD architecture (with shared memory)



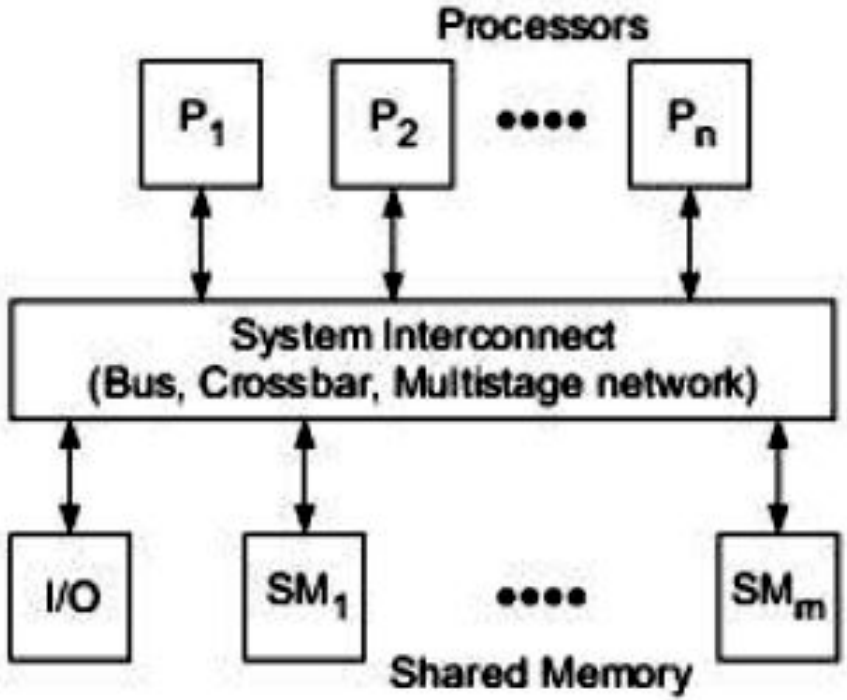
(d) MISD architecture (the systolic array)

Multiprocessors and Multicomputers

- These physical models are distinguished by having a **shared common memory** or **unshared distributed memories**.
- The processors in a multiprocessor system communicate with each other through shared variables in a common memory.
- Each computer node in a multicomputer system has a local memory, unshared with other nodes. Inter-processor communication is done through message passing among the nodes.
- There are **three** types of **shared memory multiprocessor**:
  - **UMA** (Uniform Memory Access)
  - **NUMA** (Non- uniform Memory Access)
  - **COMA** (Cache Only Memory)

Uniform Memory Access (UMA)

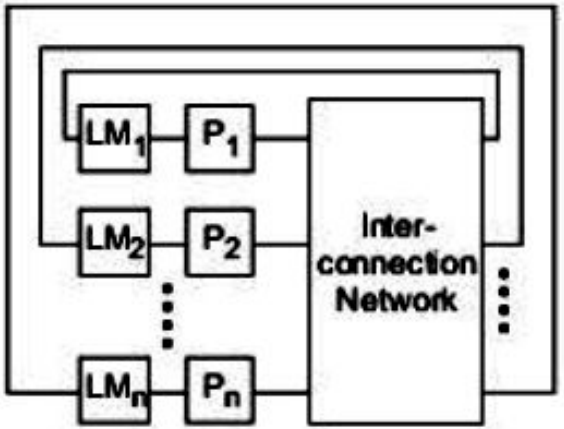
- Most commonly represented today by Symmetric Multiprocessor (SMP) machines.
- Identical processors.
- Equal access and access times to memory.
- Sometimes called CC-UMA - Cache Coherent UMA.
- Cache coherent means if one processor updates a location in shared memory, all the other processors know about the update. Cache coherency is accomplished at the hardware level.
- Multiprocessors are tightly coupled.
- The UMA model is suitable for general-purpose and times haring applications by multiple users.



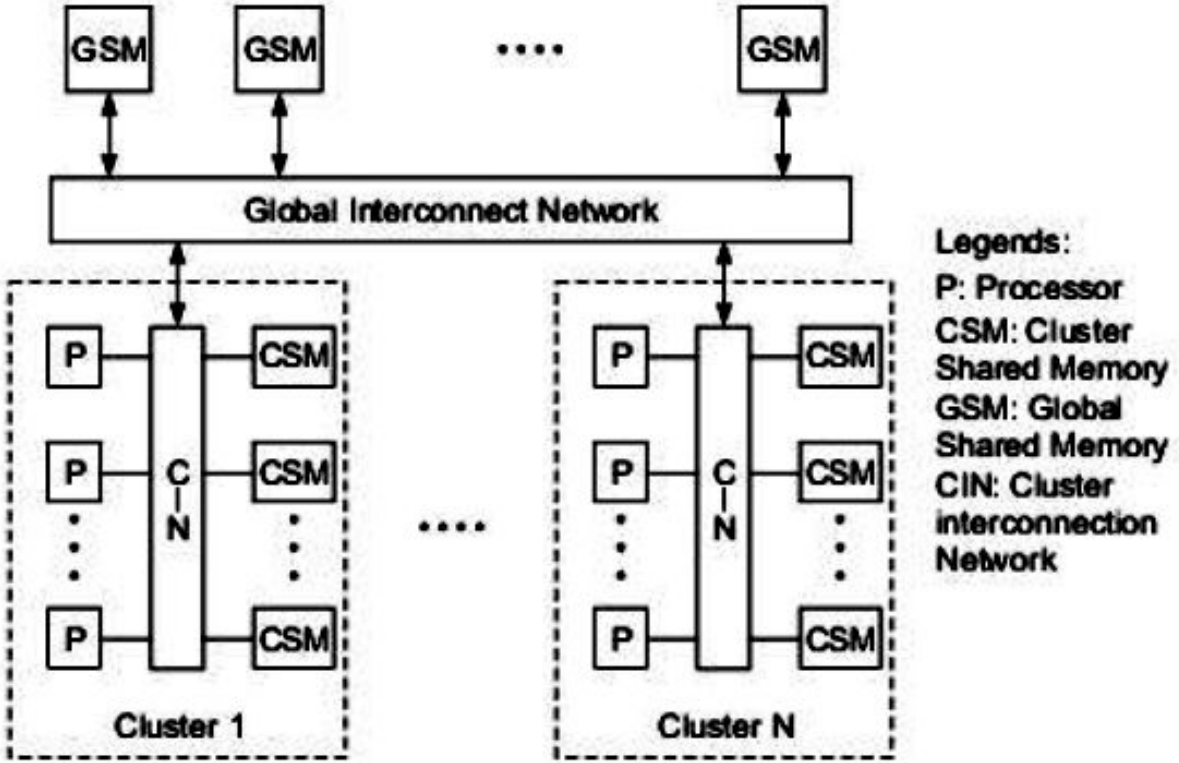
Non-Uniform Memory Access (NUMA)

- Often made by physically linking two or more SMPs
- One SMP can directly access memory of another SMP
- Not all processors have equal access time to all memories
- Memory access across link is slower.
- If cache coherency is maintained, then may also be called CC-NUMA - Cache Coherent NUMA

Non-Uniform Memory Access (NUMA)



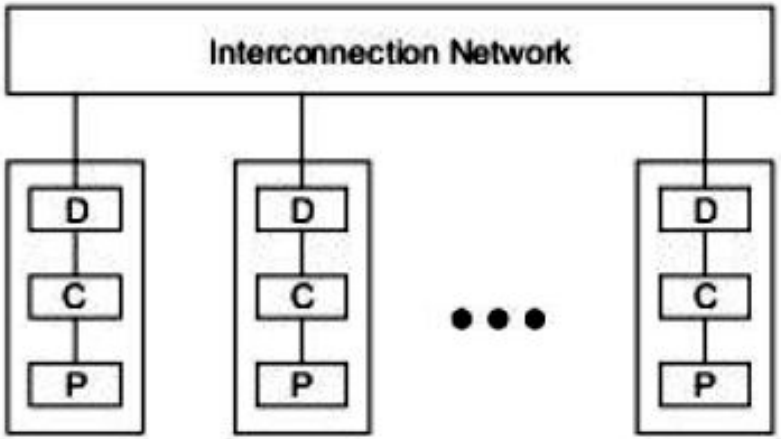
(a) Shared local memories (e.g. the N Butterfly)



(b) A hierarchical cluster model (e.g. the Cedar system at the University of Illinois)

The COMA model (Cache only Memory Access)

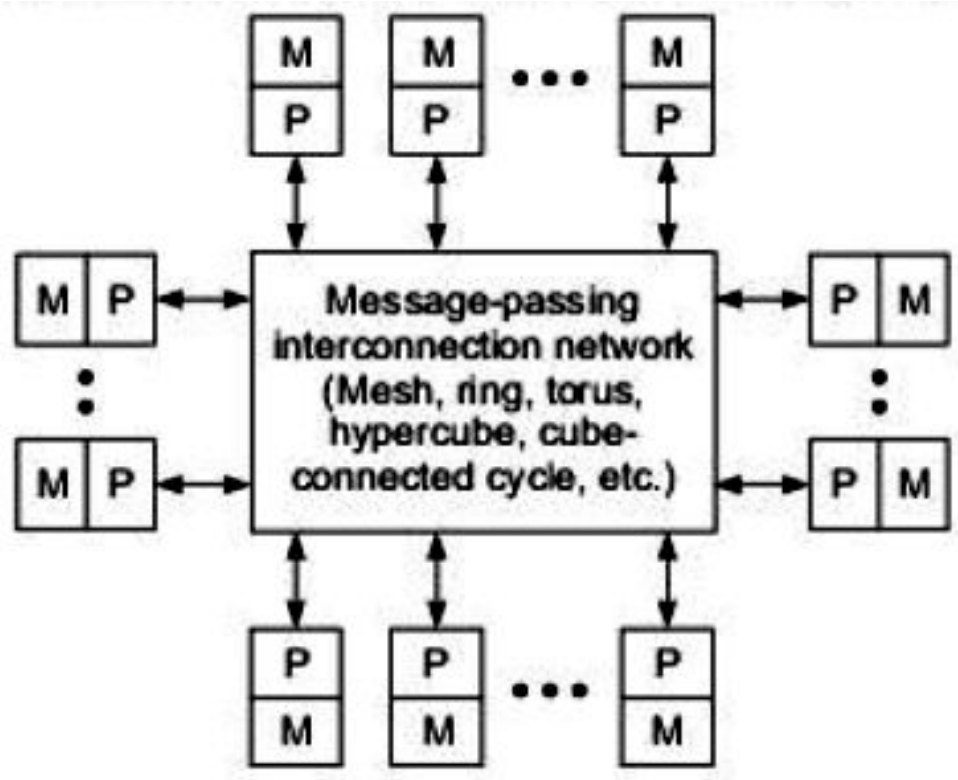
- The COMA model is a special case of NUMA machine in which the distributed main memories are converted to caches.
- All caches form a global address space and there is no memory hierarchy at each processor node.



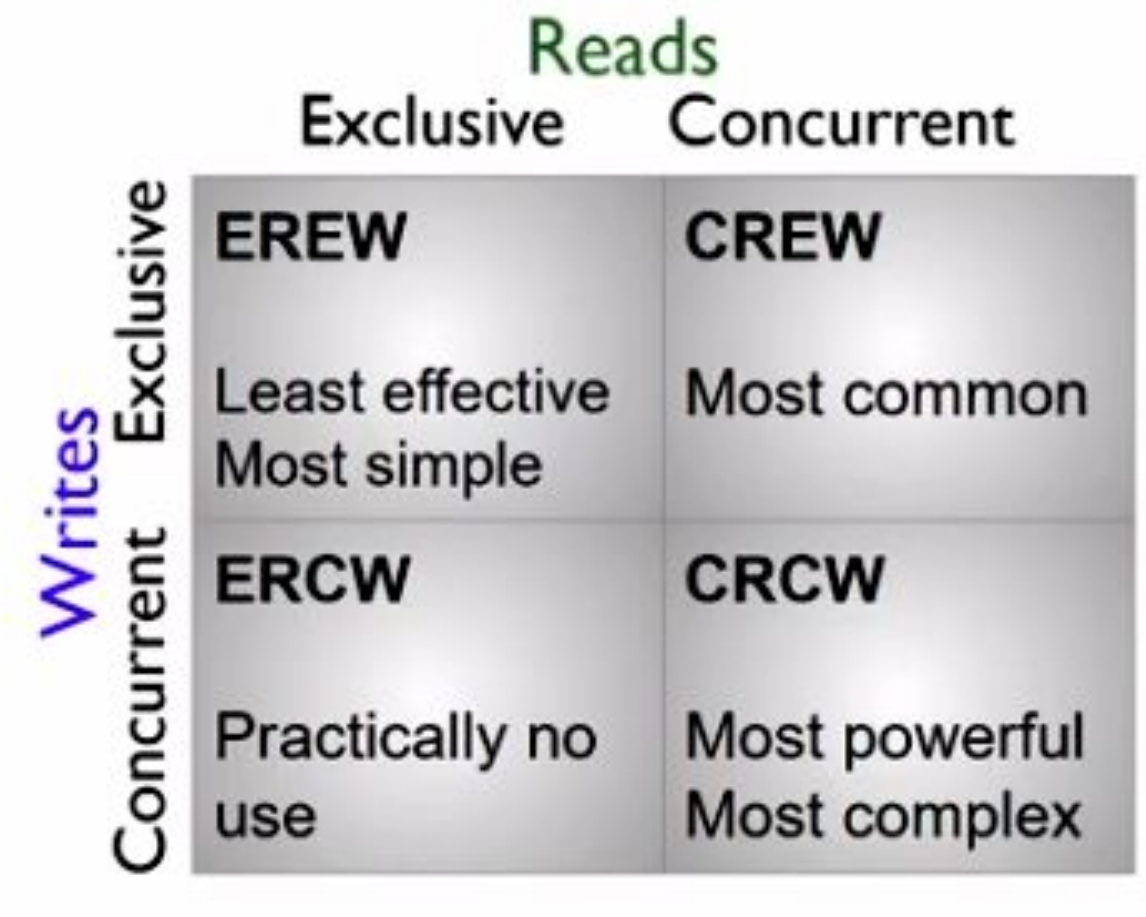
P: Processor; C: Cache; D: Directory

Distributed-Memory Multicomputers

- The system consists of multiple computers, often called nodes, interconnected by a message-passing network.
- Each node is an autonomous computer consisting of a processor, local memory, and sometimes attached disks or I/O peripherals.



PRAM Variants



Parallel Multiplication of nxn matrices on CREW PRAM

- **Input:** Two n x n matrices - A and B
- **Output:** n x n Product C matrix
- **#PEs:** n^3, later reduced to $n^3/\log_2 n$
- **Time Complexity:** $O(\log_2 n)$ for both cases

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & a_{11}b_{12} + \cdots + a_{1n}b_{n2} & \cdots & a_{11}b_{1p} + \cdots + a_{1n}b_{np} \\ a_{21}b_{11} + \cdots + a_{2n}b_{n1} & a_{21}b_{12} + \cdots + a_{2n}b_{n2} & \cdots & a_{21}b_{1p} + \cdots + a_{2n}b_{np} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & a_{m1}b_{12} + \cdots + a_{mn}b_{n2} & \cdots & a_{m1}b_{1p} + \cdots + a_{mn}b_{np} \end{pmatrix}$$

Step 1

1. Read $A(i, k)$
2. Read $B(k, j)$
3. Compute $A(i, k) \times B(k, j)$
4. Store in $C(i, j, k)$

Step 2

1. $\ell \leftarrow n$
2. Repeat
   - $\ell \leftarrow \ell/2$
   - if $(k < \ell)$ then
     - **begin**
       - Read $C(i, j, k)$
       - Read $C(i, j, k + \ell)$
       - Compute $C(i, j, k) + C(i, j, k + \ell)$
       - Store in $C(i, j, k)$
     - **end**

   until $(\ell = 1)$

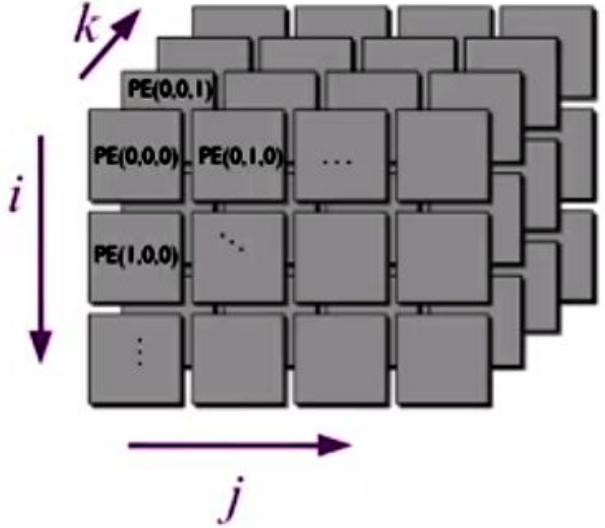
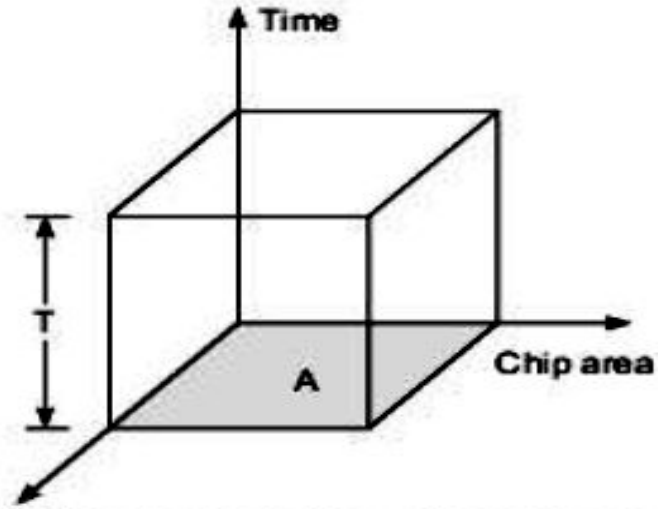


Image source: Wikipedia/Google Image search

VLSI Complexity Model

- Parallel computers rely on the use of VLSI chips to fabricate the major components such as processor arrays, memory arrays, and large-scale switching networks.
- The **AT^2 model** models the constraints while fabricating VLSI chip, these constraints include:
- **Memory Bound on Chip Area:** The amount of information processed by the chip can be visualized as information flow upward across the chip area. Each bit can flow through a unit area of the horizontal chip slice. Thus, the chip area bounds the amount of memory bits stored on the chip.
- **I/O Bound on Volume AT:** The volume of the rectangular cube is represented by the product AT. As information flows through the chip for a period of time T, the number of input bits cannot exceed the volume AT

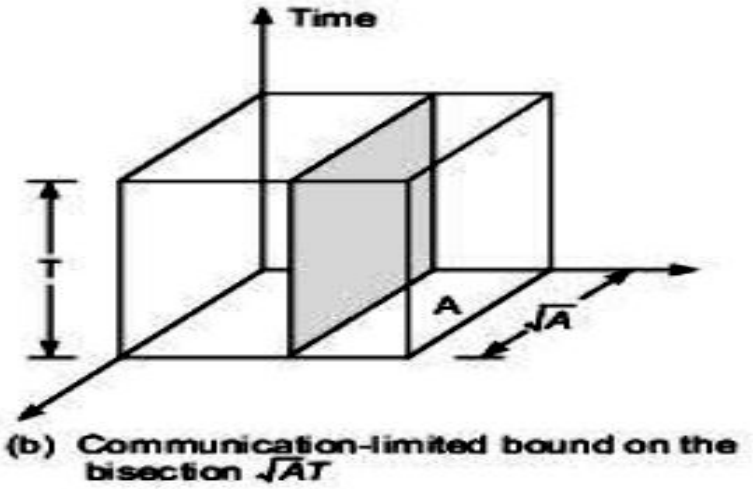


(a) Memory-limited bound on chip area *A* and I/O-limited bound on chip history represented by the volume *AT*

VLSI Complexity Model

- **Bisection Communication Bound:** a communication limited lower bound on the bisection area.
- The bisection area represents the maximum amount of information exchange between the two halves of the chip circuit during the time period T.
- If S be the problem size involved in computation, then it has been seen that there exists a lower bound f(S) such that:

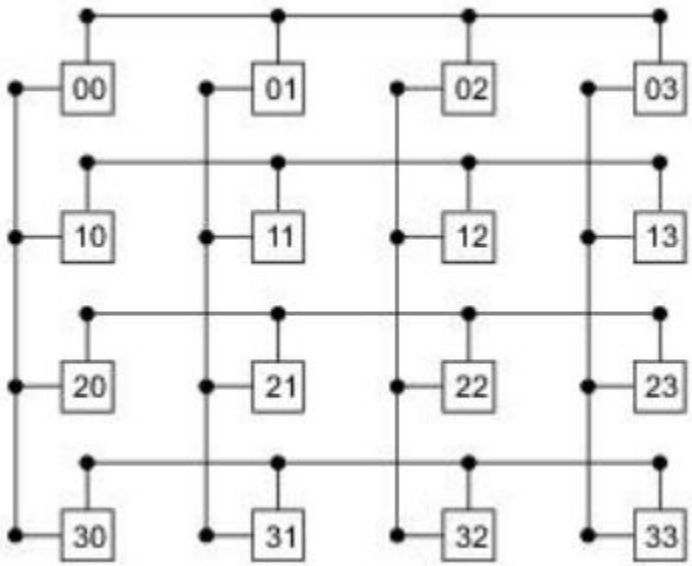
$$O(f(S)) <= AT^2.$$



(b) Communication-limited bound on the bisection $\sqrt{AT}$

Estimating chip area ‘A’ and compute time ‘T’ nxn matrix multiplication

```
      Doall 10 for 0 ≤ i, j ≤ n − 1
10        PE(i, j) sets C(i, j) to 0 /Initialization/
      Do 50 for 0 ≤ k ≤ n − 1
          Doall 20 for 0 ≤ i ≤ n − 1
20            PE(i, k) broadcasts A(i, k) along its row bus
          Doall 30 for 0 ≤ j ≤ n − 1
30            PE(k, j) broadcasts B(k, j) along its column bus
              /PE(i, j) now has A(i, k) and B(k, j), 0 ≤ i, j ≤ n − 1/
          Doall 40 for 0 ≤ i, j ≤ n − 1
40            PE(i, j) computes C(i, j) ← C(i, j) + A(i, k) × B(k, j)
50    Continue
```

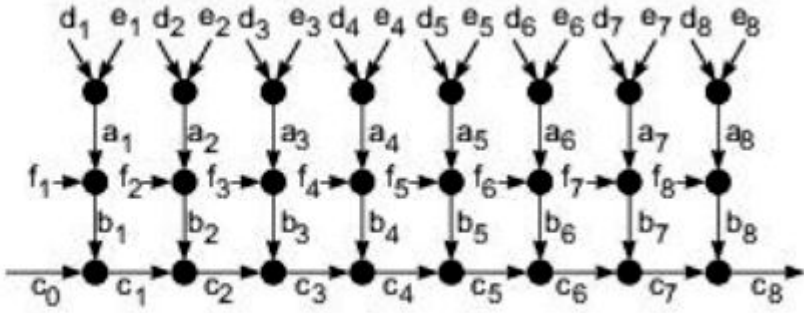


A **4 x 4** mesh of processing-elements (PB) with broadcast buses on each row and on each column

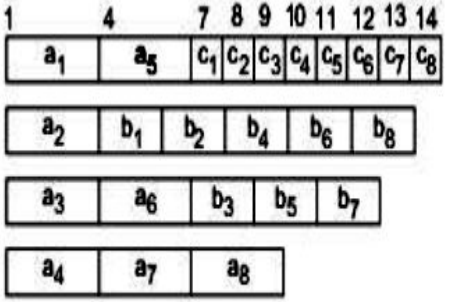
The above algorithm has a sequential loop along the dimension indexed by k. It takes n time units (iterations) in this k-loop. Thus, we have T = O(n). Therefore, $\mathbf{AT^2} = O(n^2).(O(n))^2 = \mathbf{O(n^4)}$

Comparison between dataflow and control-flow computers

```
input d, e, f
  c0 = 0
for i from 1 to 8 do
  begin
    ai := di ÷ ei
    bi := ai * fi
    ci := bi + ci-1
  end
output a, b, c
```

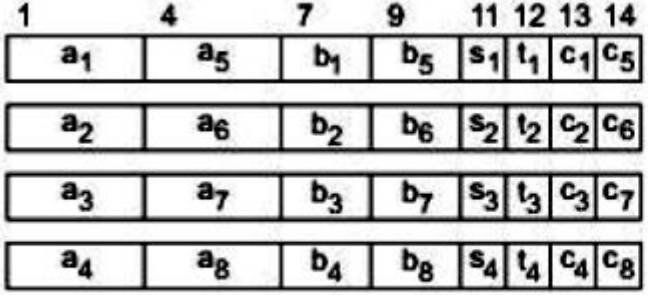


(a) A sample program and its dataflow graph



(c) Data-driven execution on a 4-processor dataflow computer in 14 cycles

(b) Sequential execution on a uniprocessor in 48 cycles



$s_1 = b_2 + b_1,\ t_1 = b_3 + s_1,\ c_1 = b_1 + c_0,\ c_5 = b_5 + c_4$

$s_2 = b_4 + b_3,\ t_2 = s_1 + s_2,\ c_2 = s_1 + c_0,\ c_6 = s_3 + c_4$

$s_3 = b_6 + b_5,\ t_3 = b_7 + s_3,\ c_3 = t_1 + c_0,\ c_7 = t_3 + c_4$

$s_4 = b_8 + b_7,\ t_4 = s_4 + s_3,\ c_4 = t_2 + c_0,\ c_8 = t_4 + c_4$

(d) Parallel execution on a shared-memory 4-processor system in 14 cycles

Machine Model	*Control Flow (control-driven)*	*Dataflow (data-driven)*	*Reduction (demand-driven)*
Basic Definition	Conventional computation; token of control indicates when a statement should be executed	Eager evaluation; statements are executed when all of their operands are available	Lazy evaluation; statements are executed only when their result is required for another computation
Advantages	Full control The most successful model for commercial products	Very high potential for parallelism	Only required instructions are executed
	Complex data and control structures are easily implemented	High throughput	High degree of parallelism
		Free from side effects	Easy manipulation of data structures
Disadvantages	In theory, less efficient than the other two	Time lost waiting for unneeded arguments	Does not support sharing of objects with changing local state
	Difficult in preventing run-time errors	High control overhead	Time needed to propagate demand tokens
		Difficult in manipulating data structures	

SYSTEM INTERCONNECT ARCHITECTURES

- These include networks which are used for interconnecting computer subsystems or for constructing multiprocessors or multicomputers.
- These networks can be used for internal connections among processors, memory modules, and HO adaptors in a centralized system, or for distributed networking of rnulticoniputcr nodes.
- The topology of an interconnection network can be either static or dynamic.
  - **Static networks** are formed of point-to-point direct connections which will not change during program execution.
    - They are used for fined connections among subsystems of e centralized system or multiple computing nodes of a distributed system.
  - **Dynamic networks** are implemented with switched channels, which are dynamically configured to match the communication demand in user programs.
    - They include buses, crossbar switches, multistage networks, and routers which are often used in shared-memory multiprocessors.

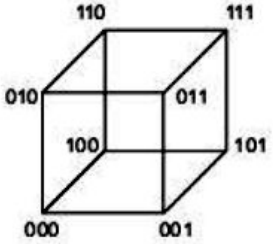
SYSTEM INTERCONNECT ARCHITECTURES

- **Node Degree (d):** The number of edges {links or channels) incident on a node.
  - In the case of unidirectional channels, the number of channels into a node is the indegree, and that out of a node is the outdegree.
  - The node degree should be kept a (small) constant, in order to reduce oost.
- **Diameter (D):** of a network is the maximum shortest path between any two nodes.
  - The path length is measured by the number of links traversed.

SYSTEM INTERCONNECT ARCHITECTURES

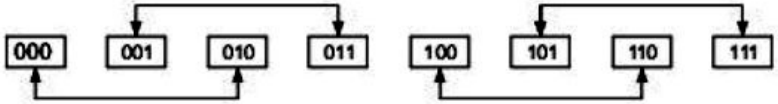
- **Bisection Width (b):** When a given network is cut into two equal halves, the minimum number of edges (channels) along the cut is called the channel bisection width.
  - If a channel has w bit wires, then wire bisection width **B = bw,** reflecting the wiring density of network.
  - If B is fixed, then **w = B/b**, providing a good indicator of the maximum communication bandwidth along the bisection of a network.
- **Data-Routing Functions:** A data-routing network is used for inter-PE data exchange.
  - Commonly seen data-routing functions among the PEs include shifting, rotation, permutation (one-to-one), broadcast (one-to-all), multicast (one-to-many), shuffle, exchange, etc.
  - These routing functions can be implemented on ring, mesh, hypercube, or multistage networks.
  - E.g. permutation **pi = (a,b,c)(d,e)** means **a->b, b->c, c->a, d->e, e->d**, where (a,b,c) has **period** of 3 and (d,e) has period of 2. Combining the two will result into the permutation of cycle 3x2=6.

Hypercube Routing Functions

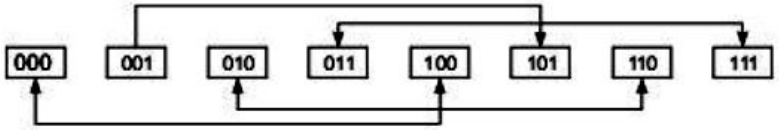


(a) A 3-cube with nodes denoted as $C_2C_1C_0$ in binary

(b) Routing by least significant bit, C_0



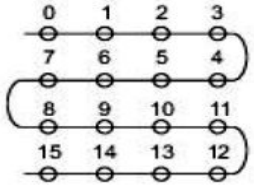
(c) Routing by middle bit, C_1



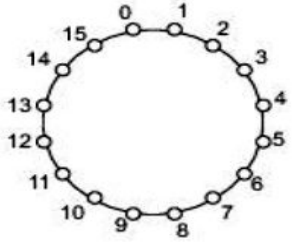
(d) Routing by most significant bit, C_2

Fig. 2.15 Three routing functions defined by a binary 3-cube

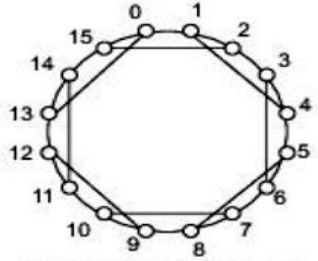
Other static connection networks



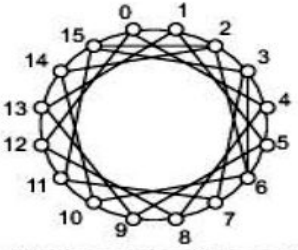
(a) Linear array



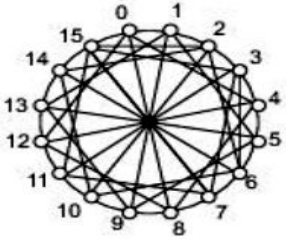
(b) Ring



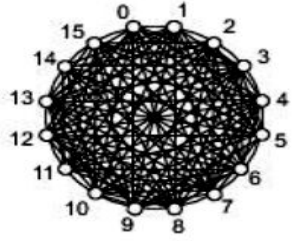
(c) Chordal ring of degree 3



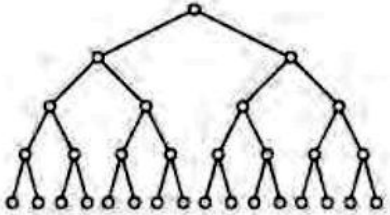
(d) Chordal ring of degree 4 (same as Illiac mesh)



(e) Barrel shifter

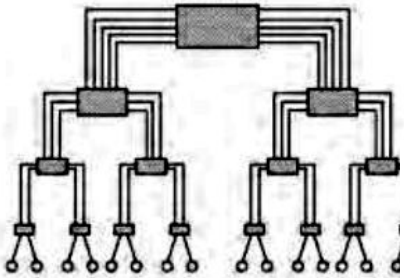


(f) Completely connected

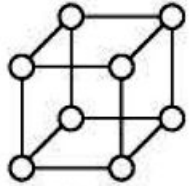


(a) Binary tree

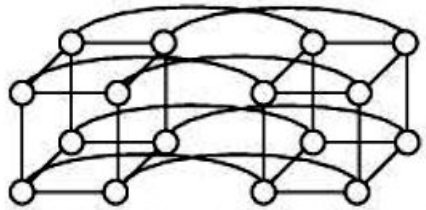
(b) Star



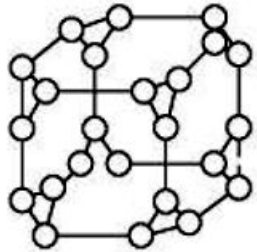
(c) Binary fat tree



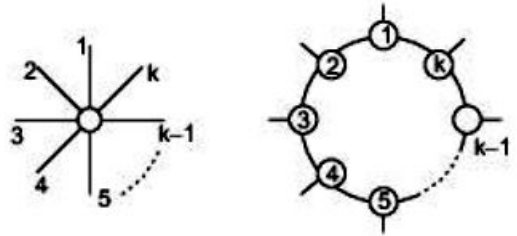
(a) 3-cube



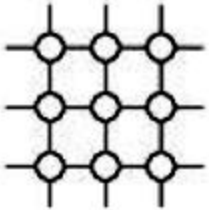
(b) A 4-cube formed by interconnecting two 3-cubes



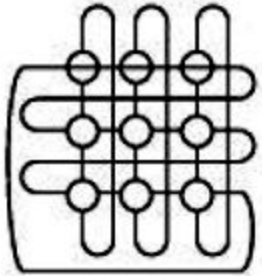
(c) 3-cube-connected cycles



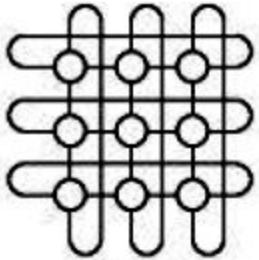
(d) Replacing each node of a *k*-cube by a ring (cycle) of *k* nodes to form the *k*-cube-connected cycles



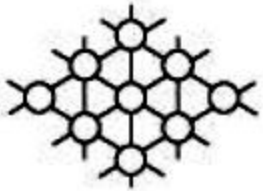
(a) Mesh



(b) Illiac mesh



(c) Torus



(d) Systolic array

Network type	*Node degree, d*	*Network diameter,*	*No. of links, l*	*Bisection width, B*	*Symmetry*	*Remarks on network size*
Linear Array	**2**	$N-1$	$N-1$	**1**	**No**	N nodes
Ring	**2**	$\lfloor N/2 \rfloor$	N	**2**	**Yes**	N nodes
Completely Connected	$N-1$	**1**	$N(N-1)/2$	$(N/2)^2$	**Yes**	N nodes
Binary Tree	**3**	$2(h-1)$	$N-1$	**1**	**No**	Tree height $h = \lceil \log_2 N \rceil$
Star	$N-1$	**2**	$N-1$	$\lfloor N/2 \rfloor$	**No**	N nodes
2D-Mesh	**4**	$2(r-1)$	$2N-2r$	r	**No**	$r \times r$ mesh where $r = \sqrt{N}$
Illiac Mesh	**4**	$r-1$	$2N$	$2r$	**No**	Equivalent to a chordal ring of $r = \sqrt{N}$
2D-Torus	**4**	$2\lfloor r/2 \rfloor$	$2N$	$2r$	**Yes**	$r \times r$ torus where $r = \sqrt{N}$
Hypercube	n	n	$nN/2$	$N/2$	**Yes**	N nodes, $n = \log_2 N$ (dimension)
CCC	**3**	$2k-1+\lfloor k/2 \rfloor$	$3N/2$	$N/(2k)$	**Yes**	$N = k \times 2^k$ nodes with a cycle length $k \geq 3$
k-ary n-cube	$2n$	$n\lfloor k/2 \rfloor$	nN	$2k^{n-1}$	**Yes**	$N = k^n$ nodes