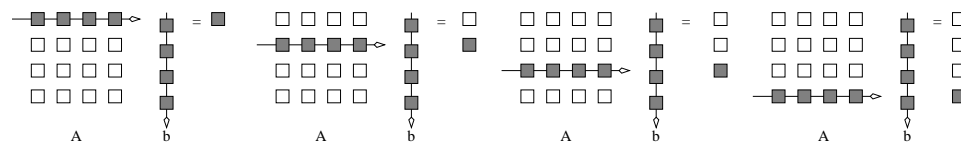
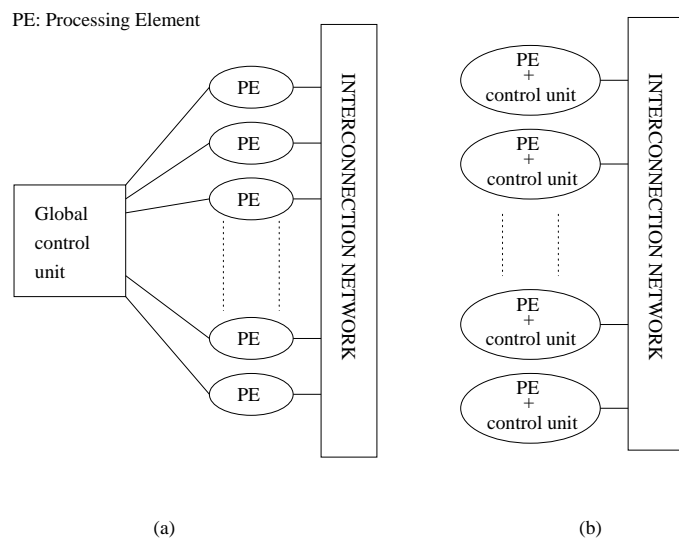


(a) Column major data access



(b) Row major data access.

**Figure 2.2** Multiplying a matrix with a vector: (a) multiplying column-by-column, keeping a running sum; (b) computing each element of the result as a dot product of a row of the matrix with the vector.



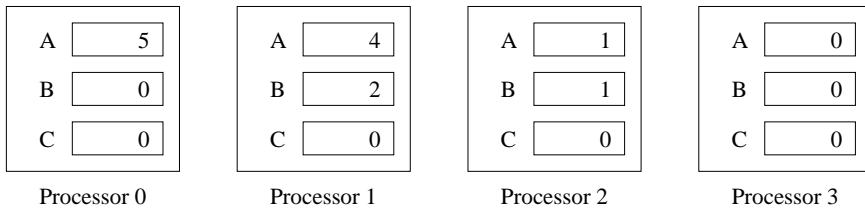
**Figure 2.3** A typical SIMD architecture (a) and a typical MIMD architecture (b).

```

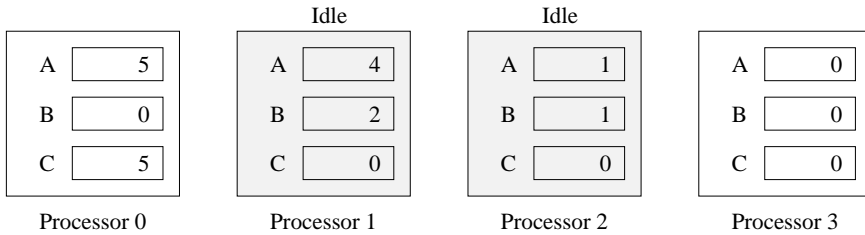
if (B == 0)
    C = A;
else
    C = A/B;

```

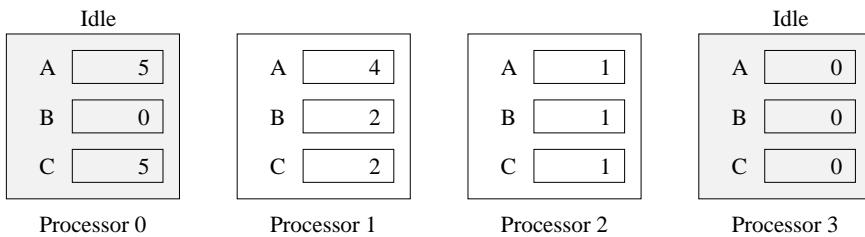
(a)



Initial values



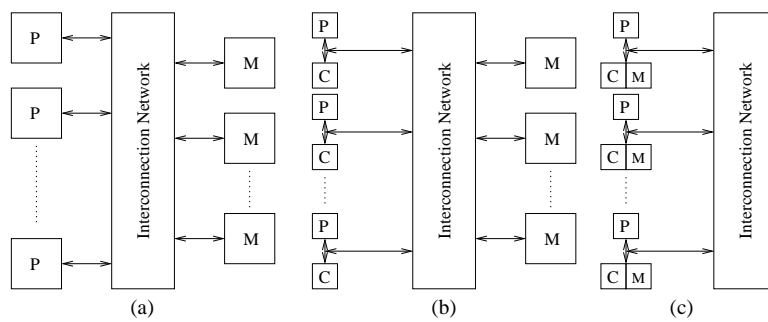
Step 1



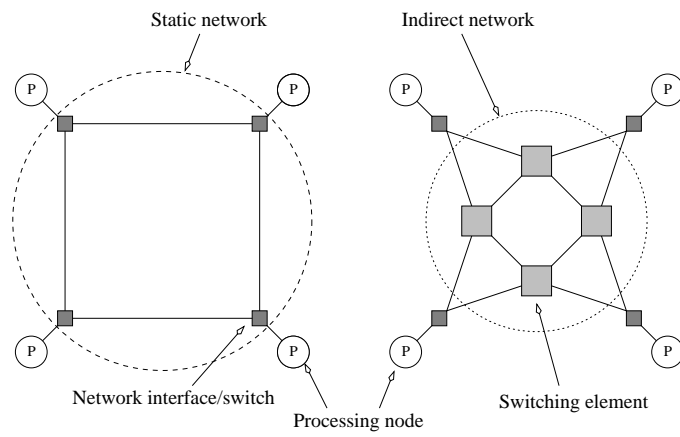
Step 2

(b)

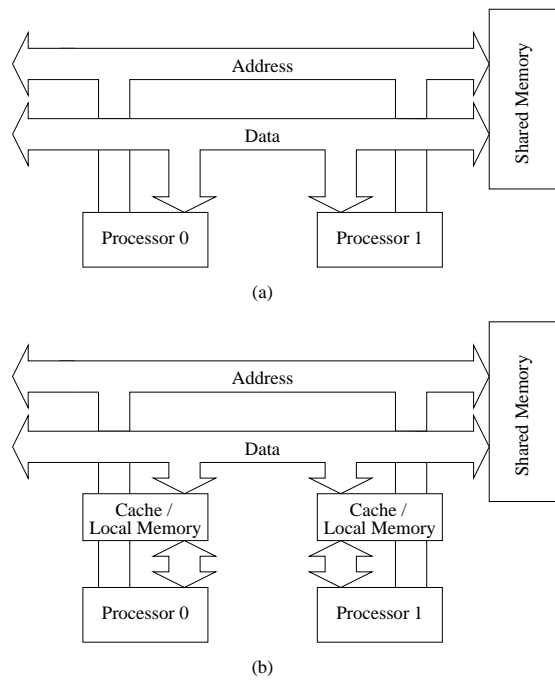
**Figure 2.4** Executing a conditional statement on an SIMD computer with four processors: (a) the conditional statement; (b) the execution of the statement in two steps.



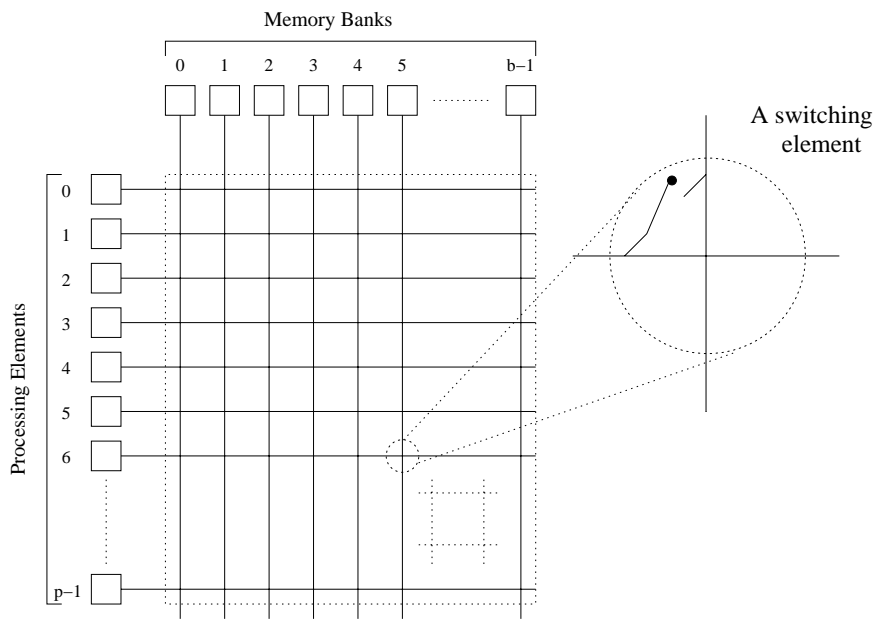
**Figure 2.5** Typical shared-address-space architectures: (a) Uniform-memory-access shared-address-space computer; (b) Uniform-memory-access shared-address-space computer with caches and memories; (c) Non-uniform-memory-access shared-address-space computer with local memory only.



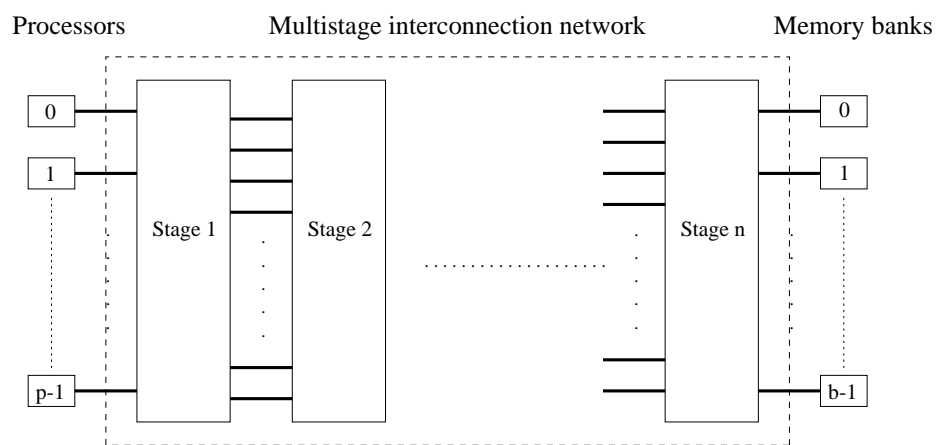
**Figure 2.6** Classification of interconnection networks: (a) a static network; and (b) a dynamic network.



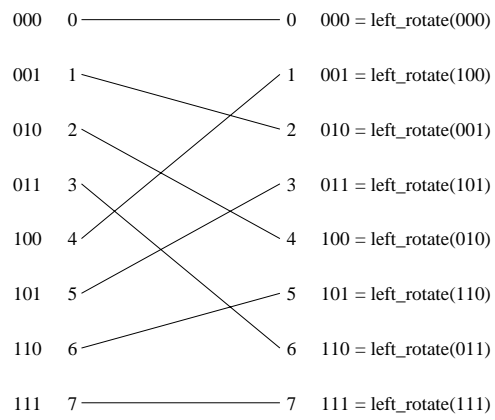
**Figure 2.7** Bus-based interconnects (a) with no local caches; (b) with local memory/caches.



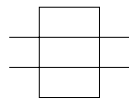
**Figure 2.8** A completely non-blocking crossbar network connecting  $p$  processors to  $b$  memory banks.



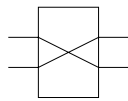
**Figure 2.9** The schematic of a typical multistage interconnection network.



**Figure 2.10** A perfect shuffle interconnection for eight inputs and outputs.



(a)



(b)

**Figure 2.11** Two switching configurations of the  $2 \times 2$  switch: (a) Pass-through; (b) Cross-over.

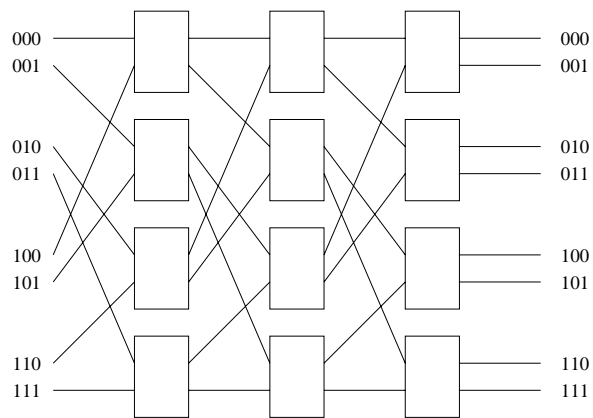
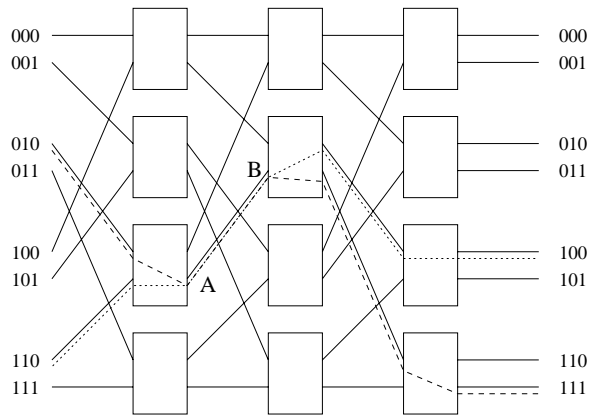
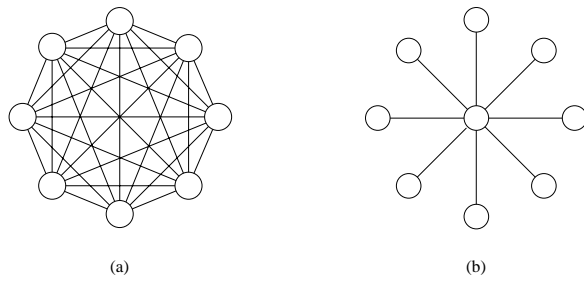


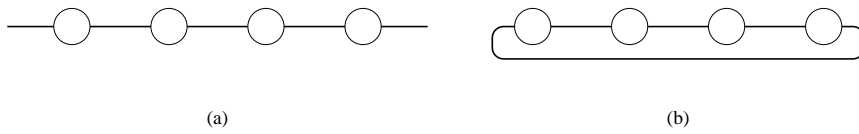
Figure 2.12 A complete omega network connecting eight inputs and eight outputs.



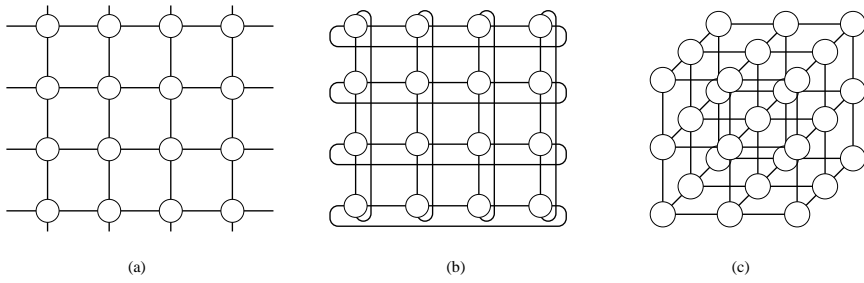
**Figure 2.13** An example of blocking in omega network: one of the messages (010 to 111 or 110 to 100) is blocked at link AB.



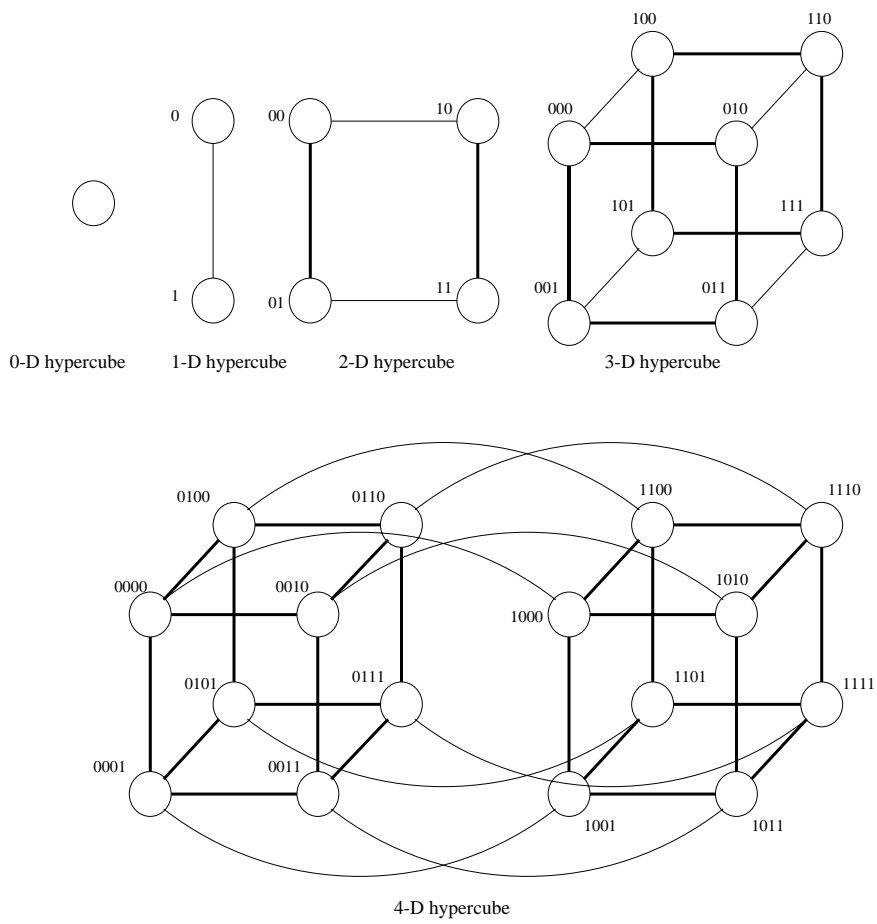
**Figure 2.14** (a) A completely-connected network of eight nodes; (b) a star connected network of nine nodes.



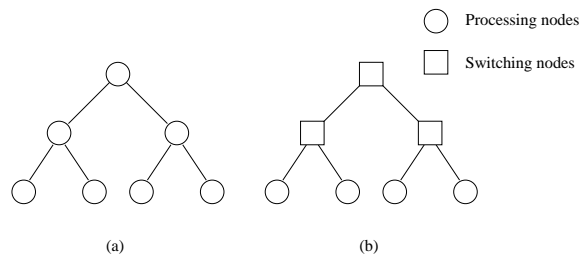
**Figure 2.15** Linear arrays: (a) with no wraparound links; (b) with wraparound link.



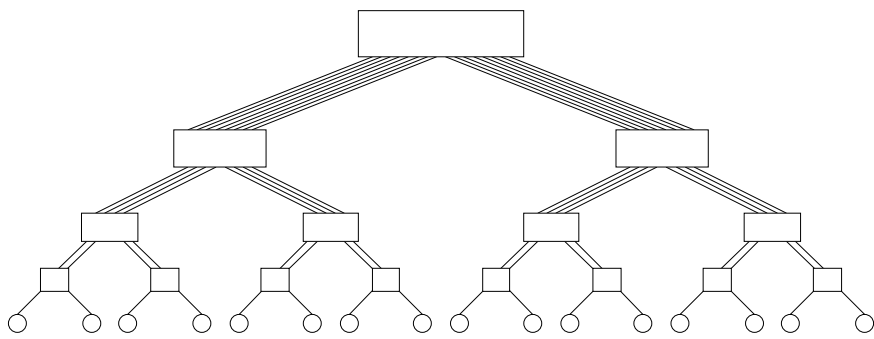
**Figure 2.16** Two and three dimensional meshes: (a) 2-D mesh with no wraparound; (b) 2-D mesh with wraparound link (2-D torus); and (c) a 3-D mesh with no wraparound.



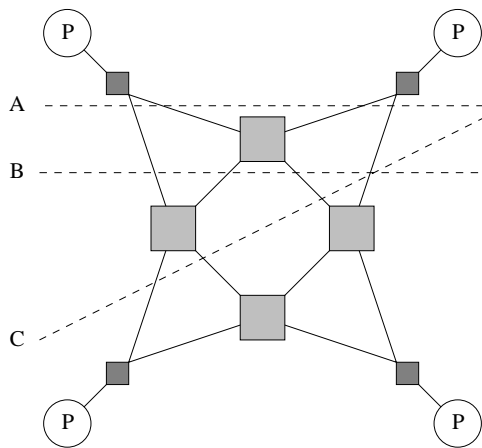
**Figure 2.17** Construction of hypercubes from hypercubes of lower dimension.



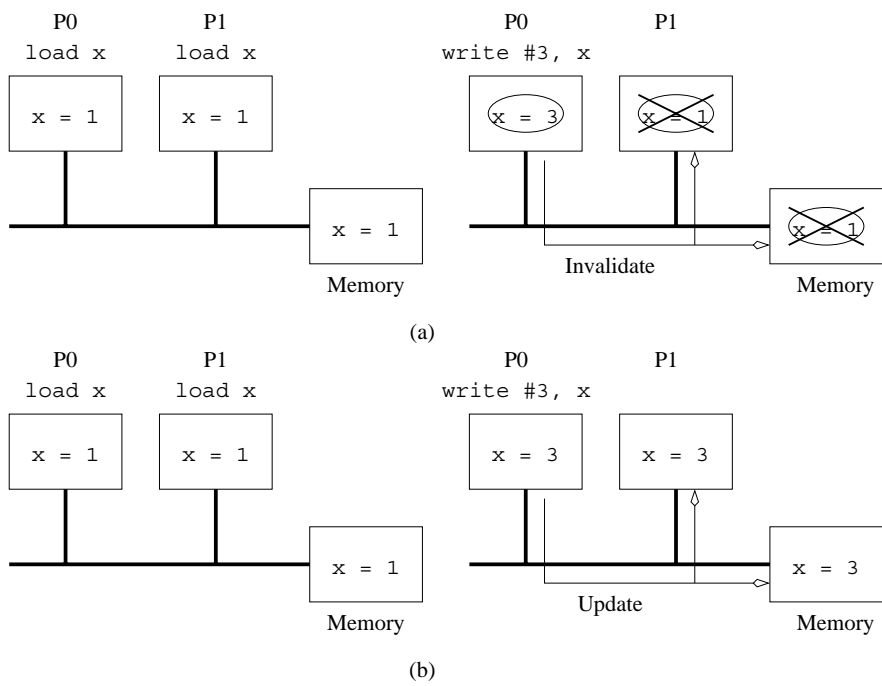
**Figure 2.18** Complete binary tree networks: (a) a static tree network; and (b) a dynamic tree network.



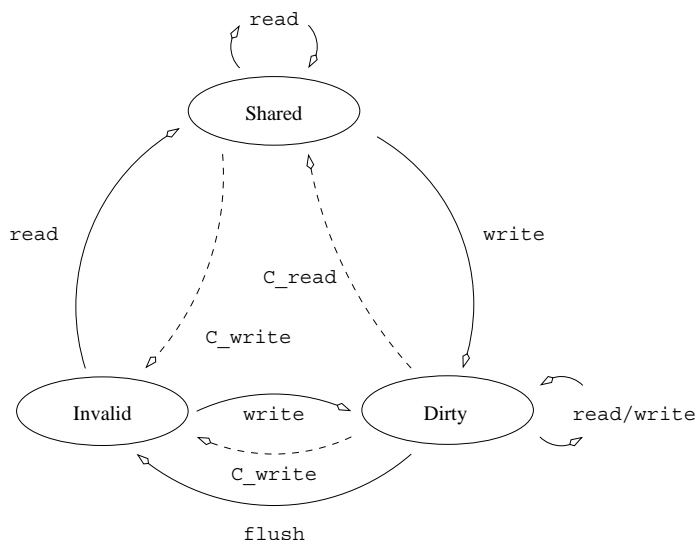
**Figure 2.19** A fat tree network of 16 processing nodes.



**Figure 2.20** Bisection width of a dynamic network is computed by examining various equi-partitions of the processing nodes and selecting the minimum number of edges crossing the partition. In this case, each partition yields an edge cut of four. Therefore, the bisection width of this graph is four.



**Figure 2.21** Cache coherence in multiprocessor systems: (a) Invalidate protocol; (b) Update protocol for shared variables.

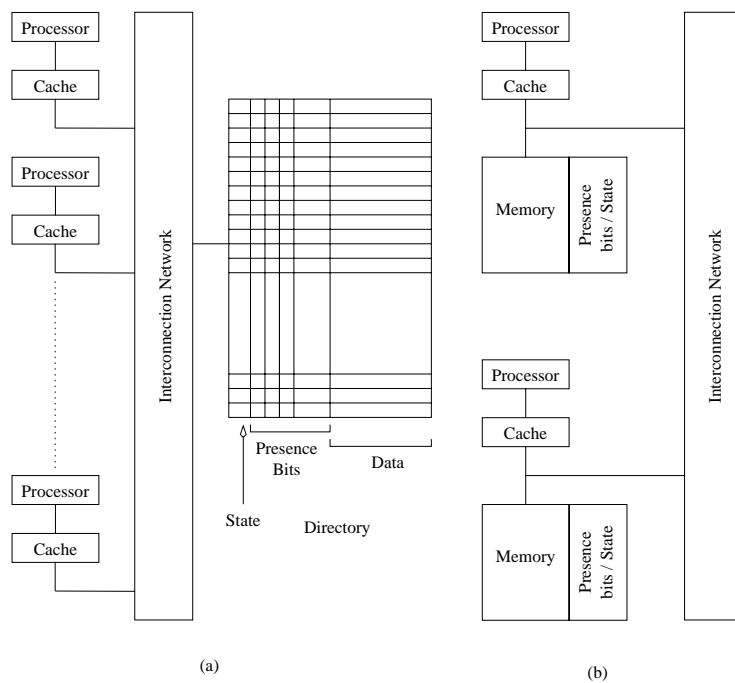


**Figure 2.22** State diagram of a simple three-state coherence protocol.

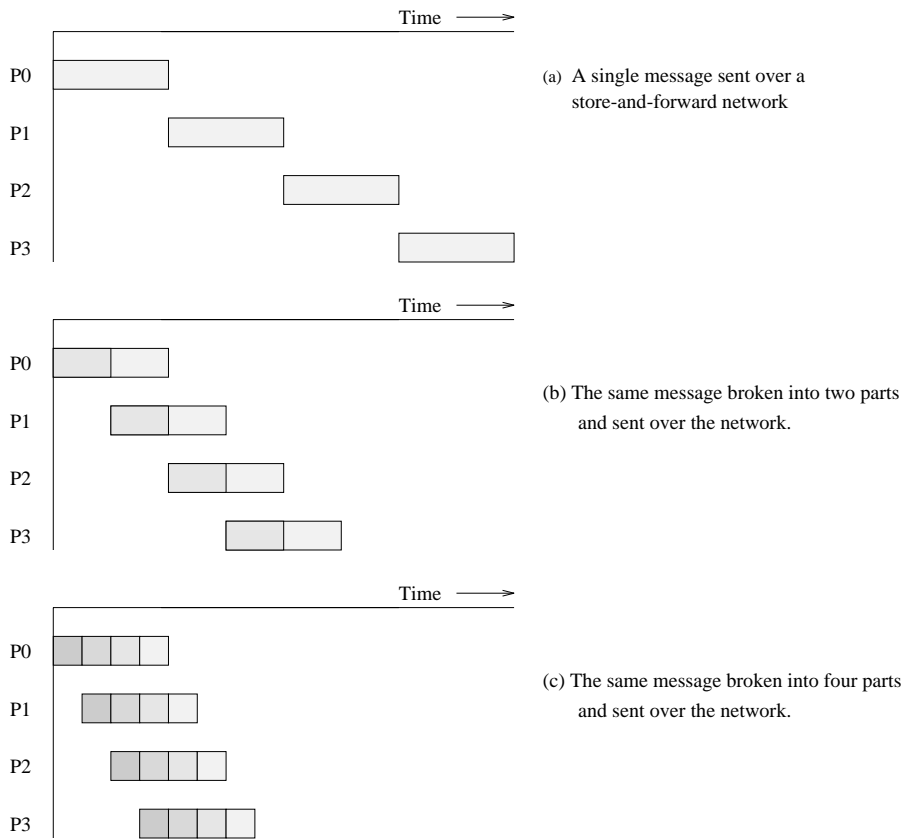
Time ↓ ∇	Instruction at Processor 0	Instruction at Processor 1	Variables and their states at Processor 0	Variables and their states at Processor 1	Variables and their states in Global mem.
					x = 5, D y = 12, D
	read x	read y	x = 5, S	y = 12, S	x = 5, S y = 12, S
	x = x + 1	y = y + 1	x = 6, D	y = 13, D	x = 5, I y = 12, I
	read y	read x	y = 13, S	y = 13, S	y = 13, S
	x = x + y	y = x + y	x = 6, S	x = 6, S	x = 6, S
			x = 19, D	x = 6, I	x = 6, I
	x = x + 1	y = y + 1	y = 13, I	y = 19, D	y = 13, I
			x = 20, D	y = 20, D	x = 6, I y = 13, I

**Figure 2.23** Example of parallel program execution with the simple three-state coherence protocol discussed in Section ??.

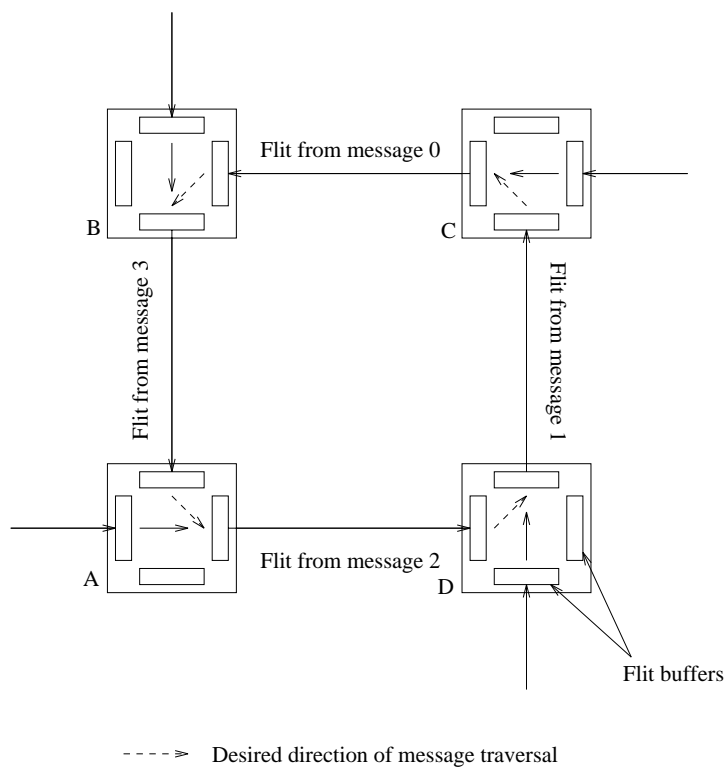




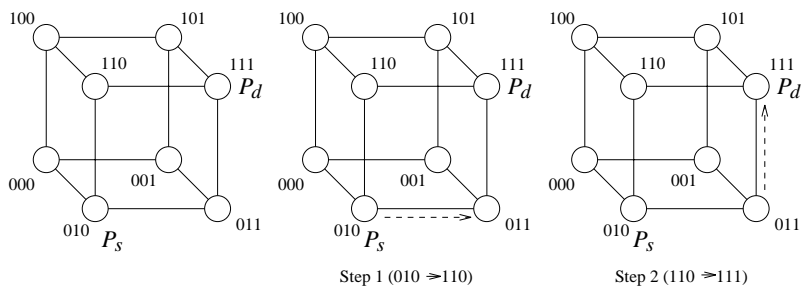
**Figure 2.25** Architecture of typical directory based systems: (a) a centralized directory; and (b) a distributed directory.



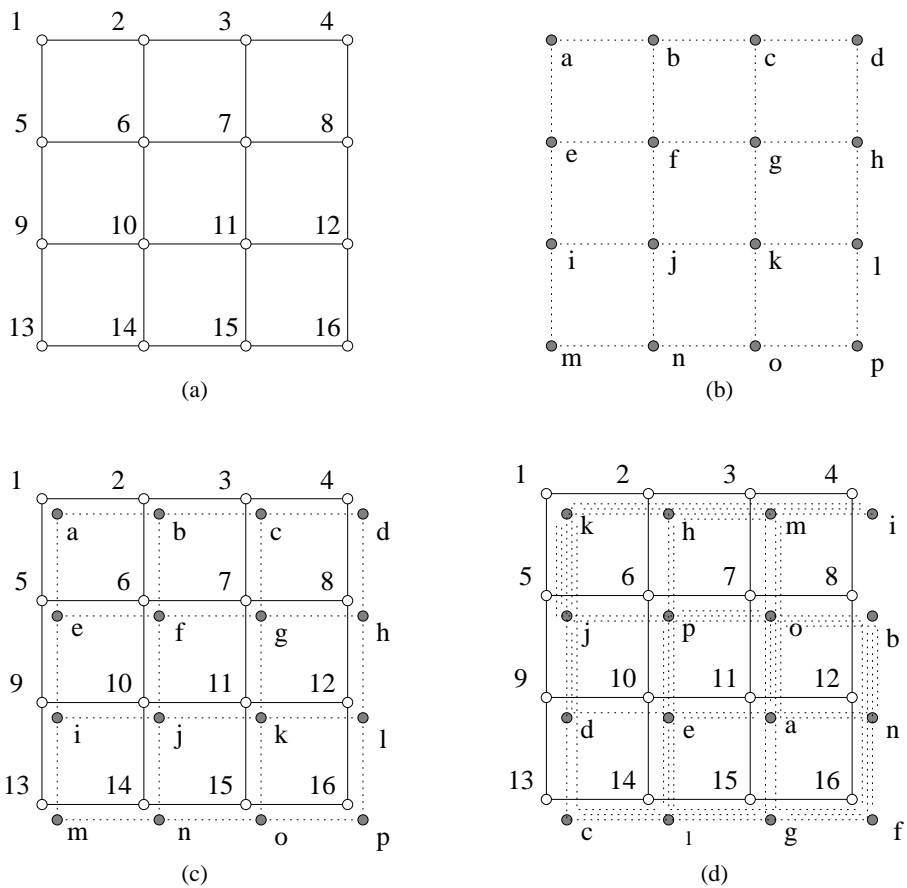
**Figure 2.26** Passing a message from node  $P_0$  to  $P_3$  (a) through a store-and-forward communication network; (b) and (c) extending the concept to cut-through routing. The shaded regions represent the time that the message is in transit. The startup time associated with this message transfer is assumed to be zero.



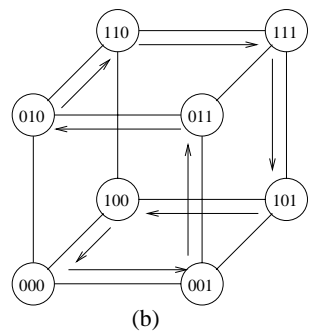
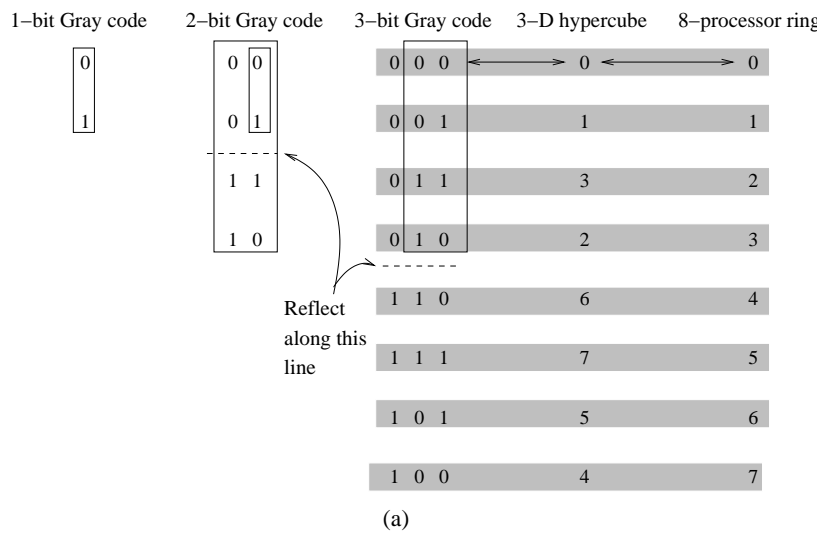
**Figure 2.27** An example of deadlock in a cut-through routing network.



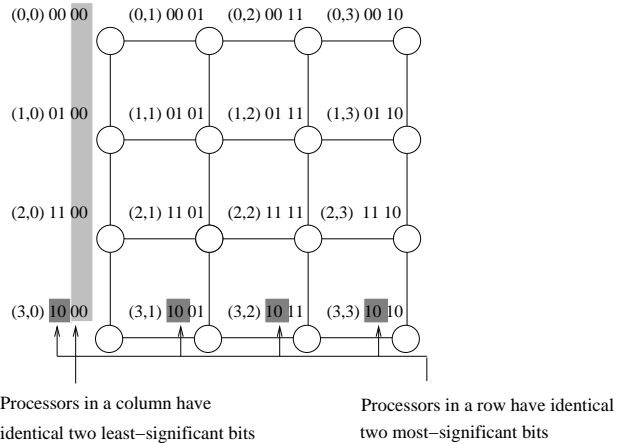
**Figure 2.28** Routing a message from node  $P_s$  (010) to node  $P_d$  (111) in a three-dimensional hypercube using E-cube routing.



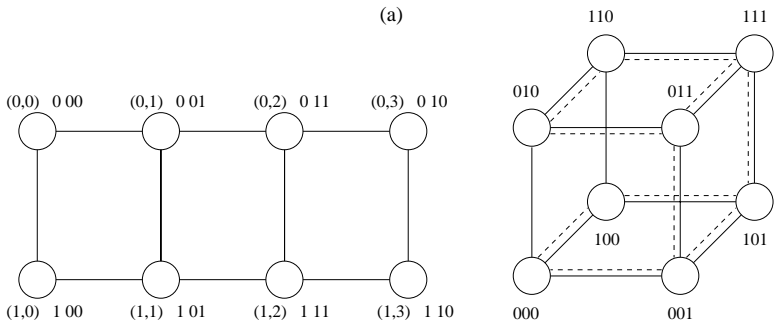
**Figure 2.29** Impact of process mapping on performance: (a) underlying architecture; (b) processes and their interactions; (c) an intuitive mapping of processes to nodes; and (d) a random mapping of processes to nodes.



**Figure 2.30** (a) A three-bit reflected Gray code ring; and (b) its embedding into a three-dimensional hypercube.

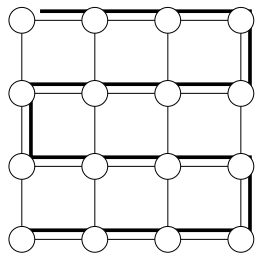


(a)

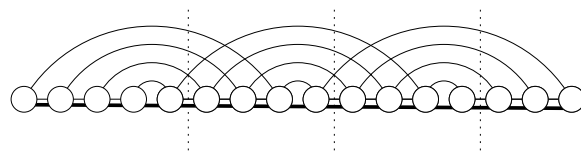


(b)

**Figure 2.31** (a) A  $4 \times 4$  mesh illustrating the mapping of mesh nodes to the nodes in a four-dimensional hypercube; and (b) a  $2 \times 4$  mesh embedded into a three-dimensional hypercube.

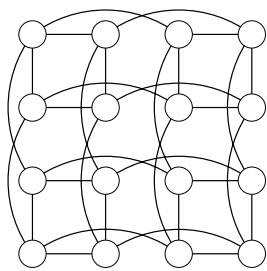


(a) Mapping a linear array into a 2D mesh (congestion 1).

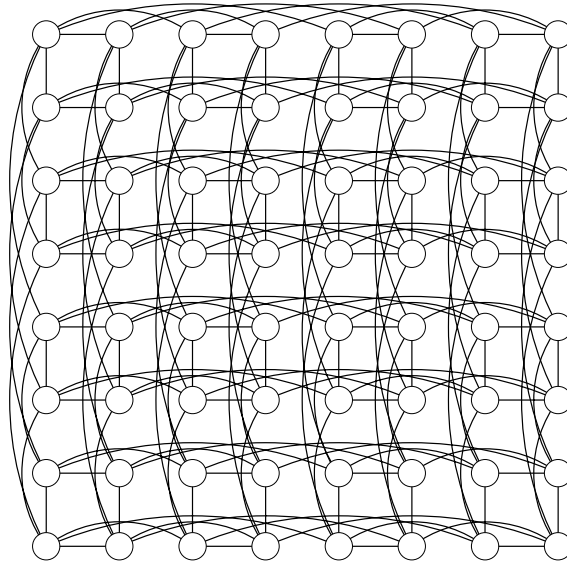


(b) Inverting the mapping – mapping a 2D mesh into a linear array (congestion 5)

**Figure 2.32** (a) Embedding a 16 node linear array into a 2-D mesh; and (b) the inverse of the mapping. Solid lines correspond to links in the linear array and normal lines to links in the mesh.

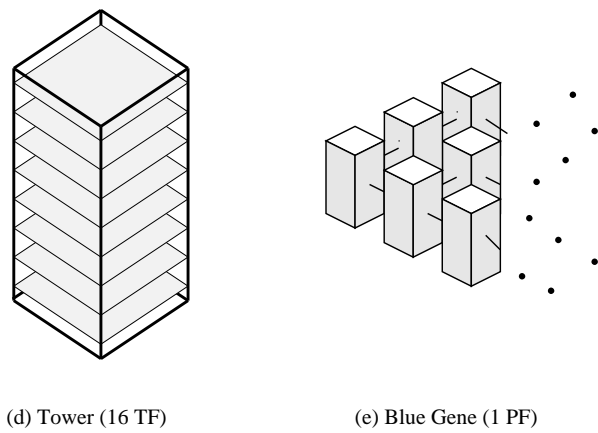
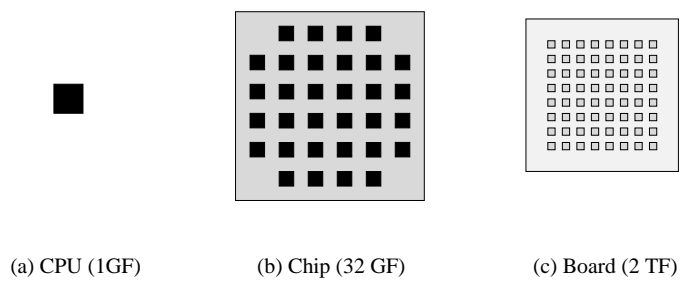


(a)  $P = 16$

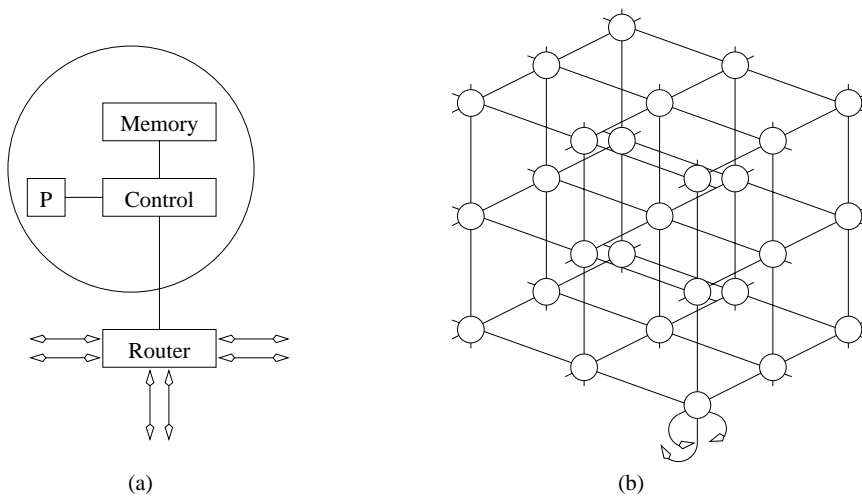


(b)  $P = 32$

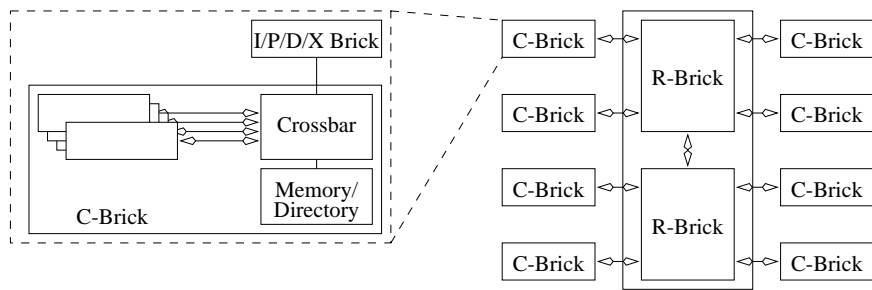
**Figure 2.33** Embedding a hypercube into a 2-D mesh.



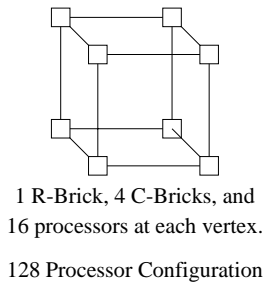
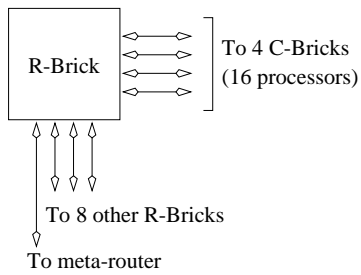
**Figure 2.34** The hierarchical architecture of Blue Gene.



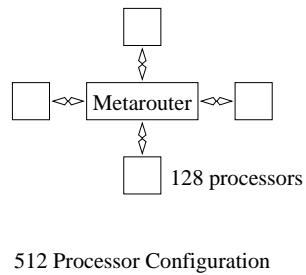
**Figure 2.35** Interconnection network of the Cray T3E: (a) node architecture; (b) network topology.



32 Processor Configuration

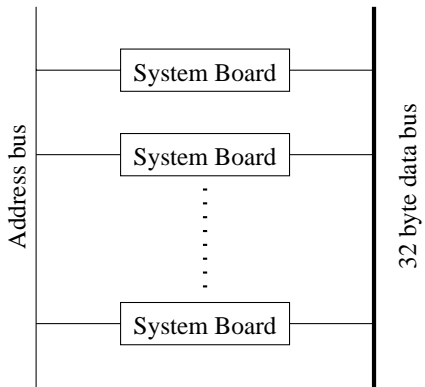


128 Processor Configuration

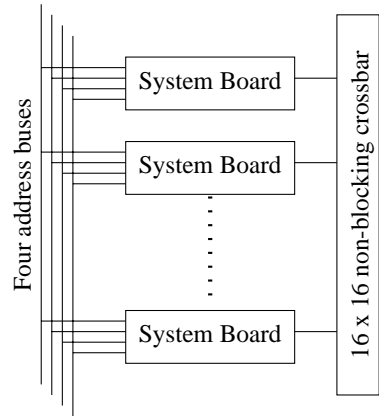


512 Processor Configuration

Figure 2.36 Architecture of the SGI Origin 3000 family of servers.

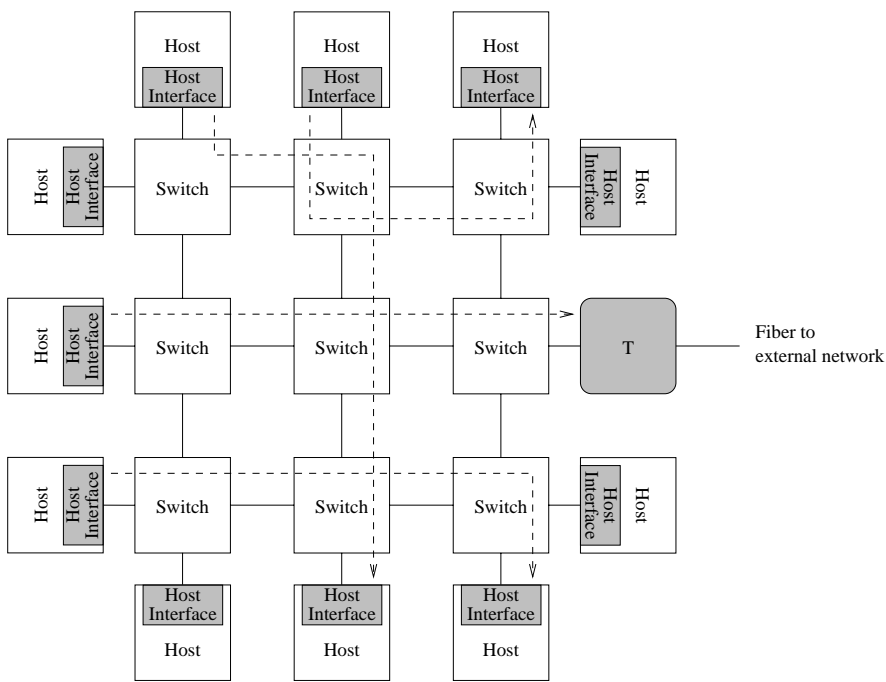


Sun Ultra 6000 (6 - 30 processors)

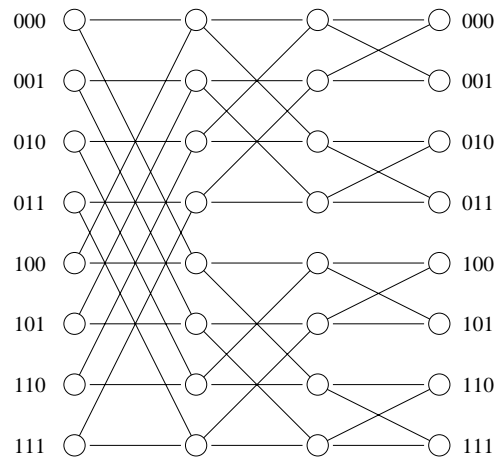


Starfire Ultra 1000 (up to 64 processors)

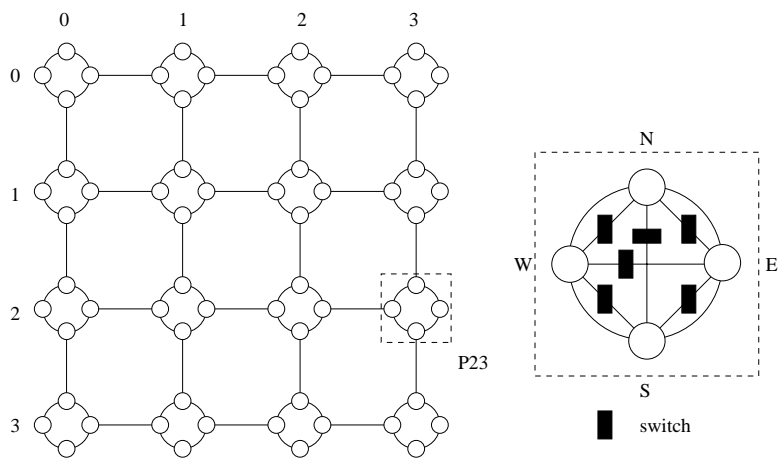
**Figure 2.37** Architecture of the Sun Enterprise family of servers.



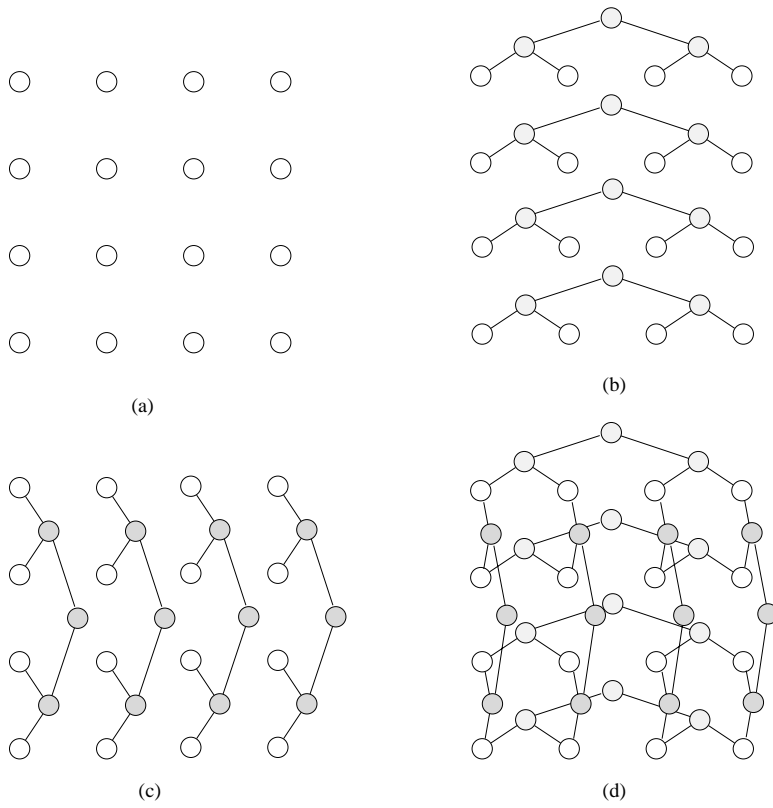
**Figure 2.38** A typical connection pattern for switches and hosts in a Myrinet. The figure also illustrates routing of messages between hosts. At any point of time, multiple pairs of processors with non-conflicting paths may be communicate with each other.



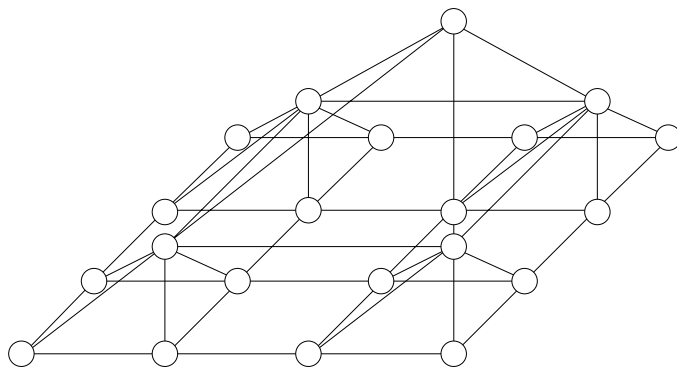
**Figure 2.39** A Butterfly network with eight processing nodes.



**Figure 2.40** Switch connection patterns in a reconfigurable mesh.



**Figure 2.41** The construction of a  $4 \times 4$  mesh of trees: (a) a  $4 \times 4$  grid, (b) complete binary trees imposed over individual rows, (c) complete binary trees imposed over each column, and (d) the complete  $4 \times 4$  mesh of trees.



**Figure 2.42** A  $4 \times 4$  pyramidal mesh.