Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Sistemas de Computação
Laboratório de Sistemas Distribuídos e Programação Concorrente

Notas de Aulas da Disciplina
SSC0903 – Computação de Alto Desempenho

Módulo 2 – Arquiteturas Paralelas

por Paulo Sérgio Lopes de Souza

São Carlos/SP – Brasil – 2019

Conteúdo

2 Arquiteturas Paralelas 1
2.1 Considerações Iniciais 1
2.1.1 Paralelismo no nível de bit 1
2.1.2 Paralelismo por pipeline 1
2.1.3 Paralelismo por múltiplas unidades funcionais 2
2.1.4 Paralelismo no nível de processos ou threads 4
2.2 Classificação de Arquiteturas Paralelas 5
2.3 MIMD 6
2.3.1 Multiprocessadores – MIMD com Memória Compartilhada 7
Semântica de Memória 9
Coerência de Cache em multiprocessadores 10
Falso compartilhamento 12
2.3.2 Multicomputadores – MIMD com Memória Distribuída 13
Clusters 13
Massively Parallel Processors 14
2.4 SIMD 15
2.5 GPUs 18
2.6 Redes de Conexão nas Arquiteturas Paralelas 20

2 Arquiteturas Paralelas

2.1 Considerações Iniciais

(Rauber & Rünger, 2013)

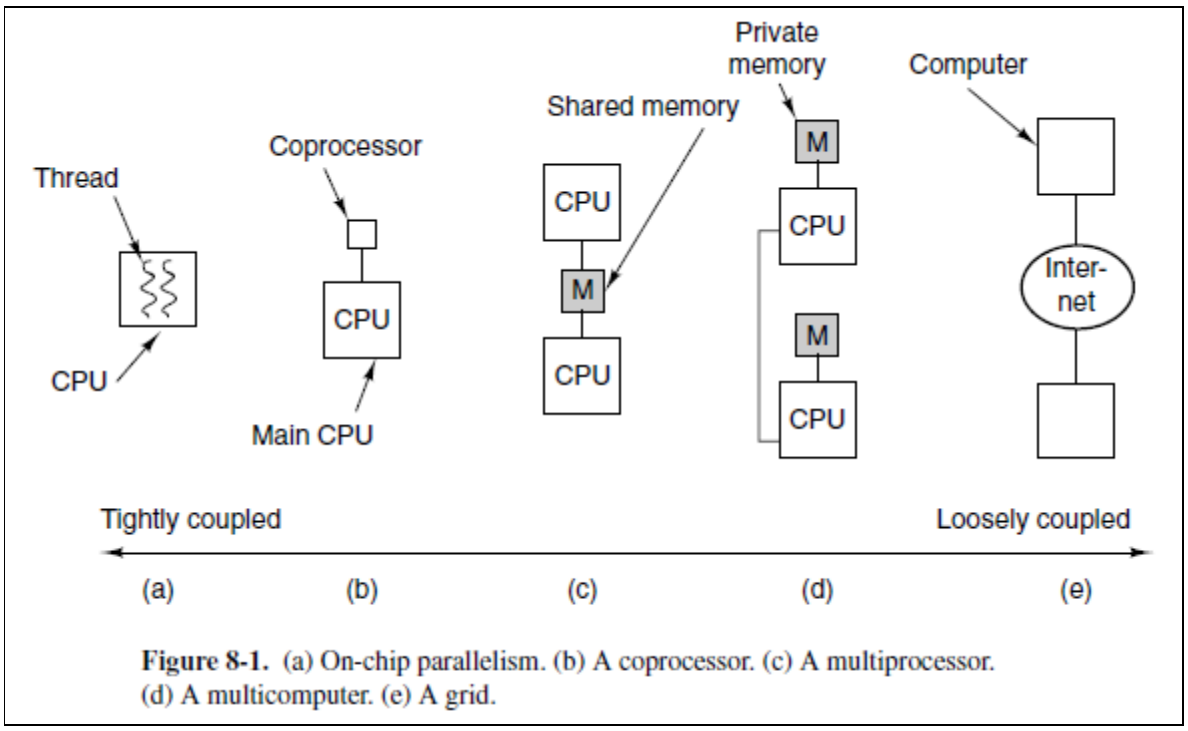


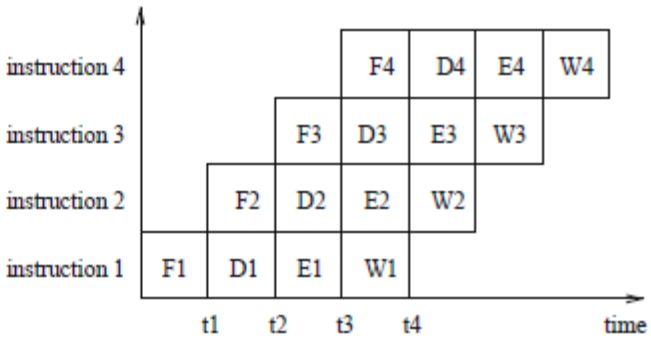
Figure 8-1. (a) On-chip parallelism. (b) A coprocessor. (c) A multiprocessor. (d) A multicomputer. (e) A grid.

(Tanenbaum, 2013)

2.1.1 Paralelismo no nível de bit

2.1.2 Paralelismo por pipeline

Fig. 2.1 Overlapping execution of four independent instructions by pipelining. The execution of each instruction is split into four stages: *fetch* (F), *decode* (D), *execute* (E), and *write back* (W).



(Rauber & Rünger, 2013)

2.1.3 Paralelismo por múltiplas unidades funcionais

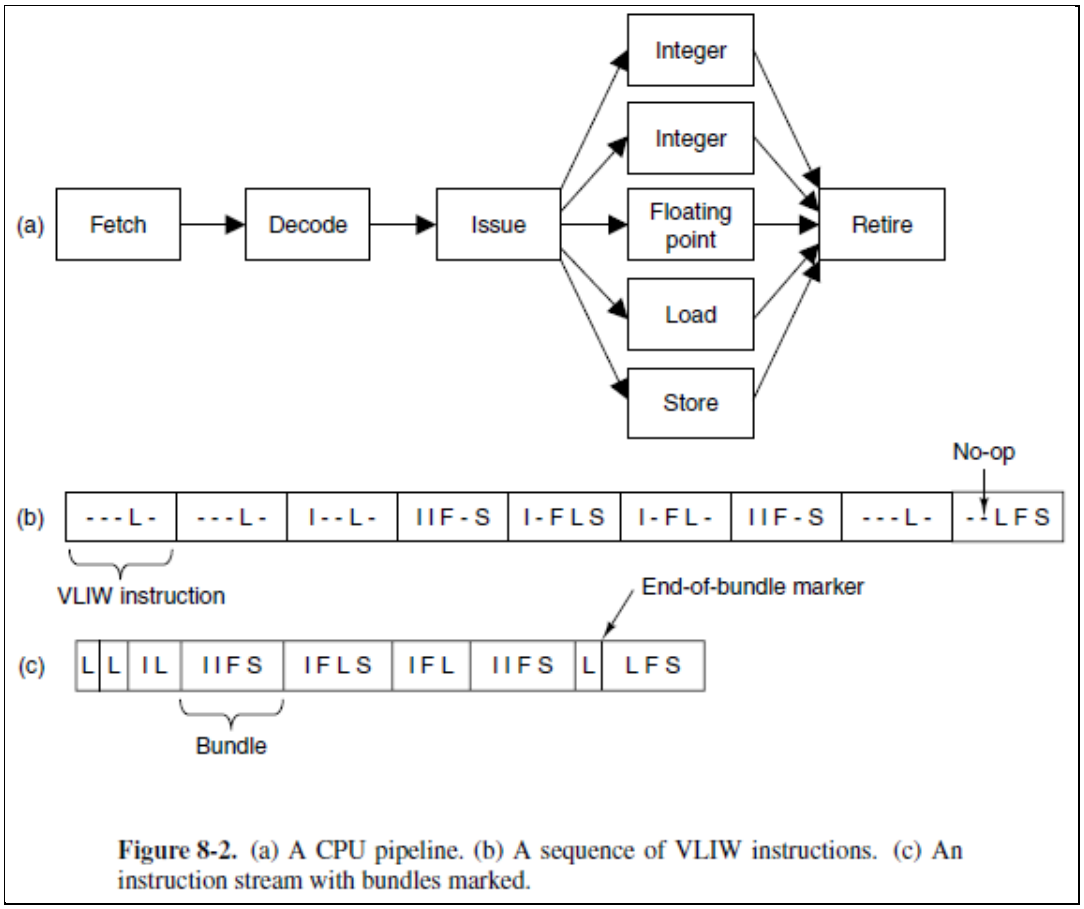


Figure 8-2. (a) A CPU pipeline. (b) A sequence of VLIW instructions. (c) An instruction stream with bundles marked.

(Tanenbaum, 2013)

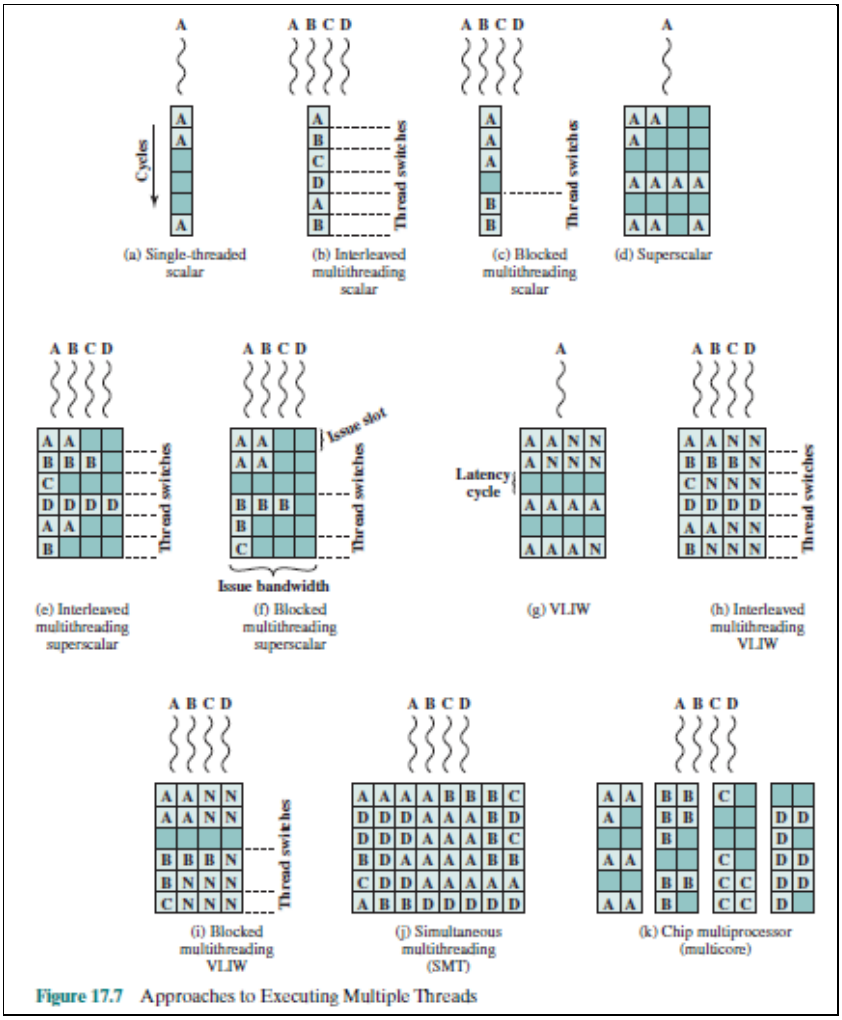


Figure 17.7 Approaches to Executing Multiple Threads

(Stallings, 2013)

2.1.4 Paralelismo no nível de processos ou threads

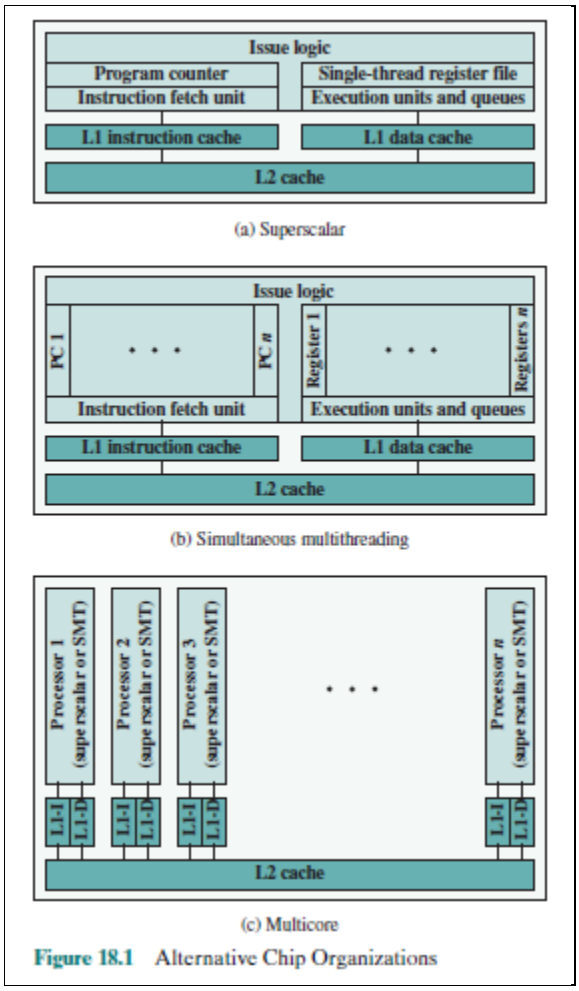
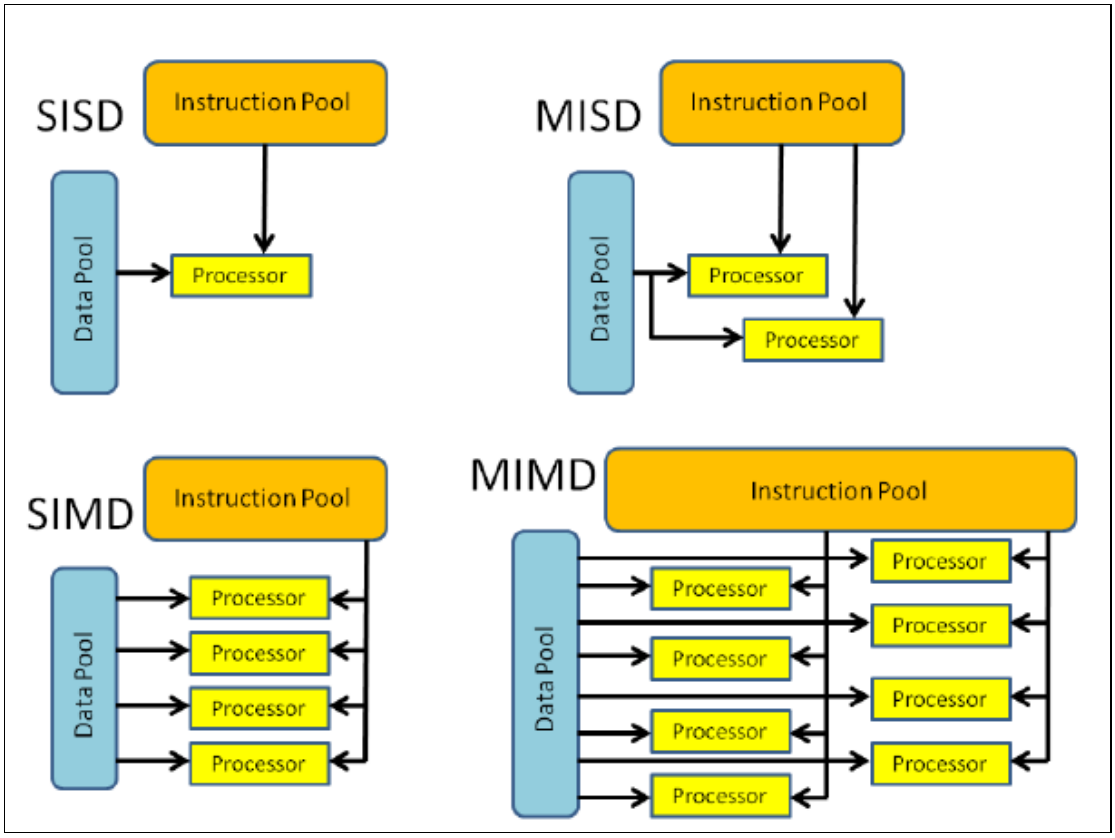


Figure 18.1 Alternative Chip Organizations

(Stallings, 2013)

2.2 Classificação de Arquiteturas Paralelas



Fonte WEB

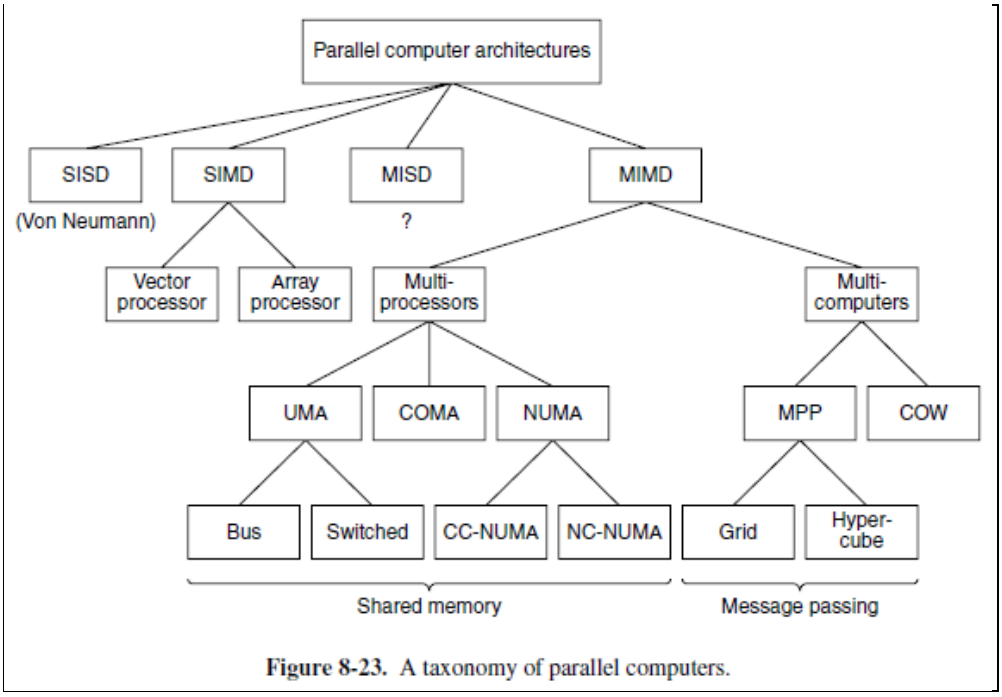
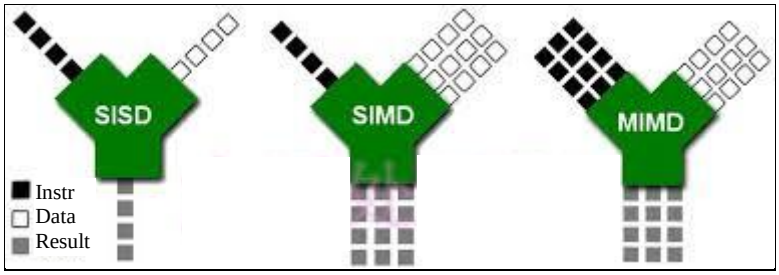
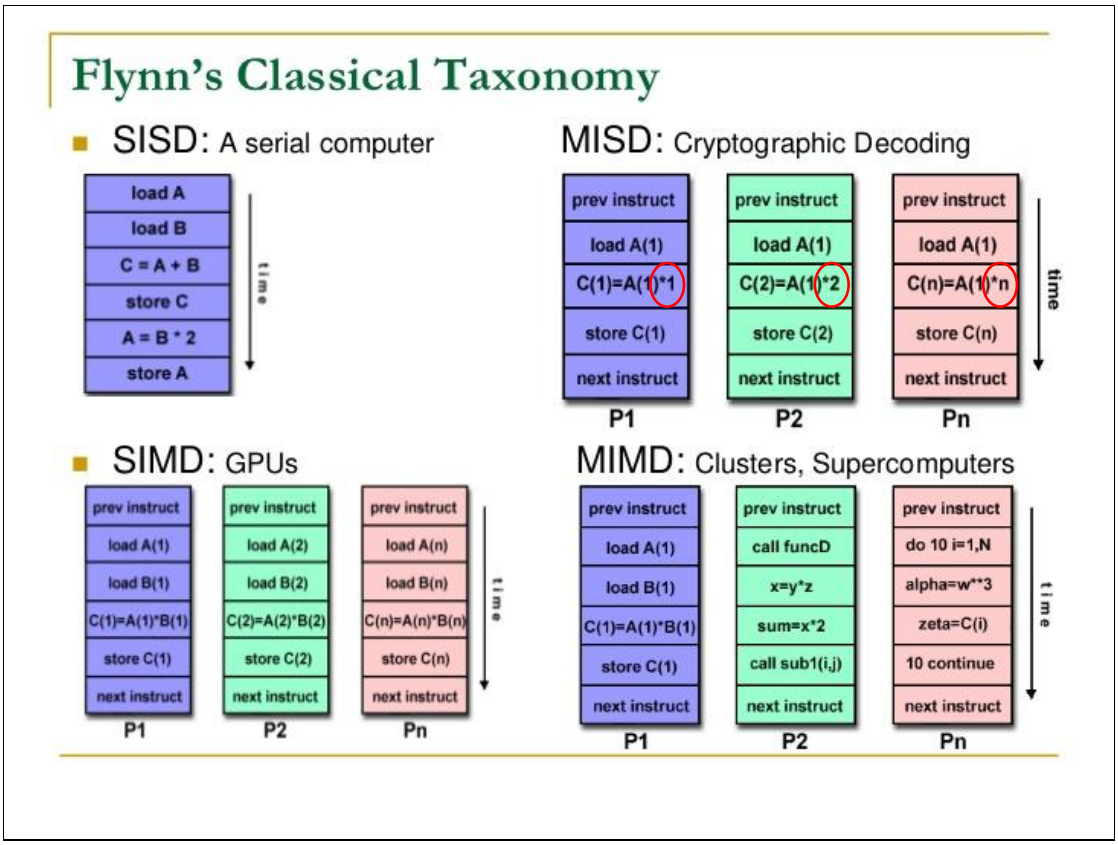


Figure 8-23. A taxonomy of parallel computers.

(Tanenbaum, 2013)



Fonte WEB



Fonte WEB

2.3 MIMD

Memória Compartilhada x Memória Distribuída

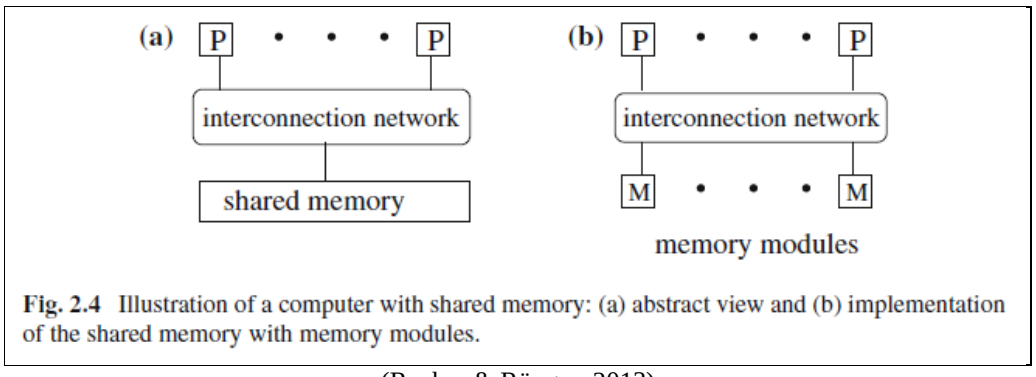


Fig. 2.4 Illustration of a computer with shared memory: (a) abstract view and (b) implementation of the shared memory with memory modules.

(Rauber & Rünger, 2013)

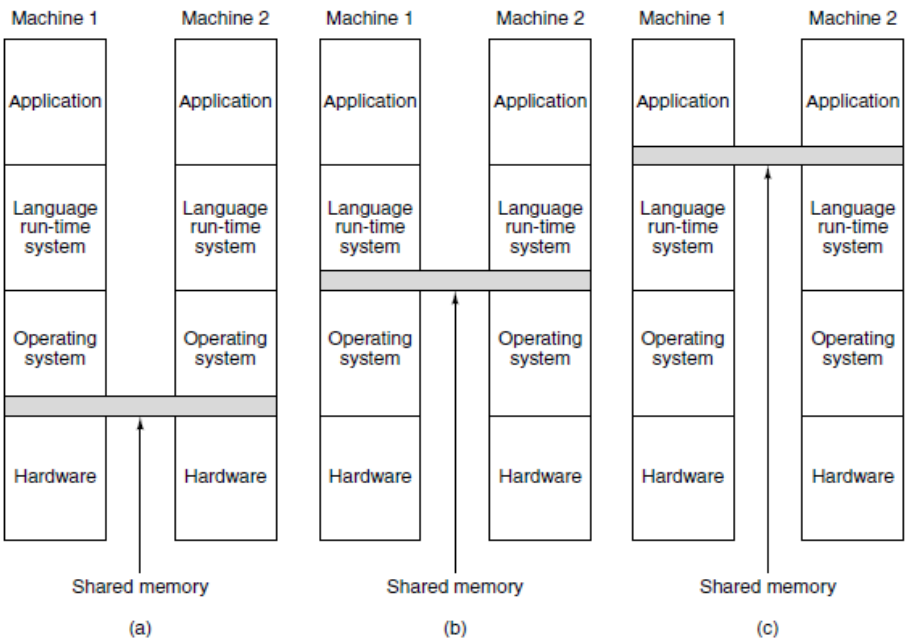


Figure 8-21. Various layers where shared memory can be implemented. (a) The hardware. (b) The operating system. (c) The language run-time system.

(Tanenbaum, 2013)

2.3.1 Multiprocessadores – MIMD com Memória Compartilhada

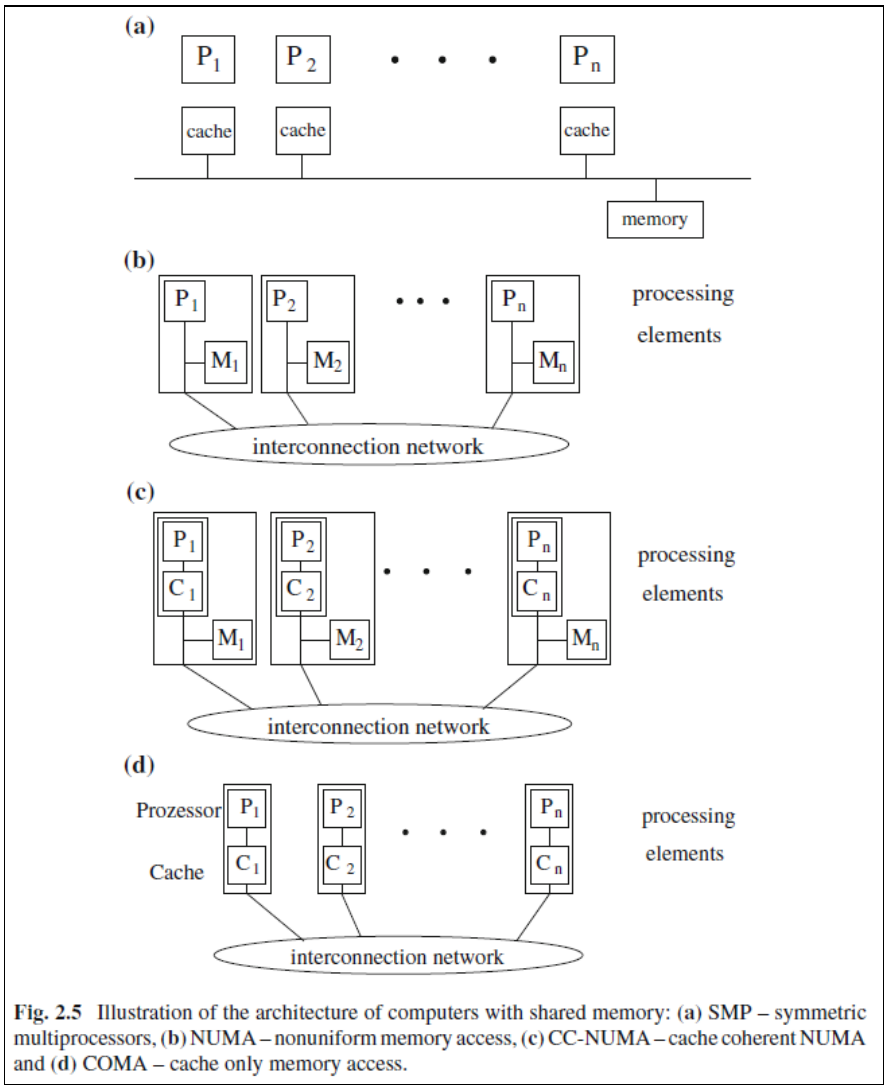


Fig. 2.5 Illustration of the architecture of computers with shared memory: (a) SMP – symmetric multiprocessors, (b) NUMA – nonuniform memory access, (c) CC-NUMA – cache coherent NUMA and (d) COMA – cache only memory access.

(Rauber & Rünger, 2013)

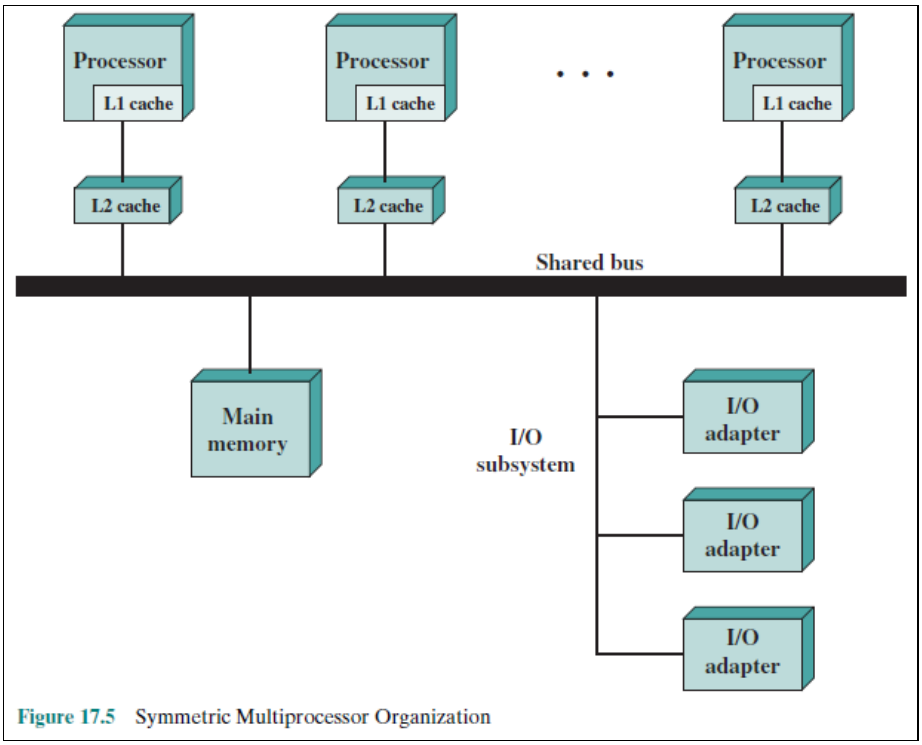


Figure 17.5 Symmetric Multiprocessor Organization

(Stallings, 2013)

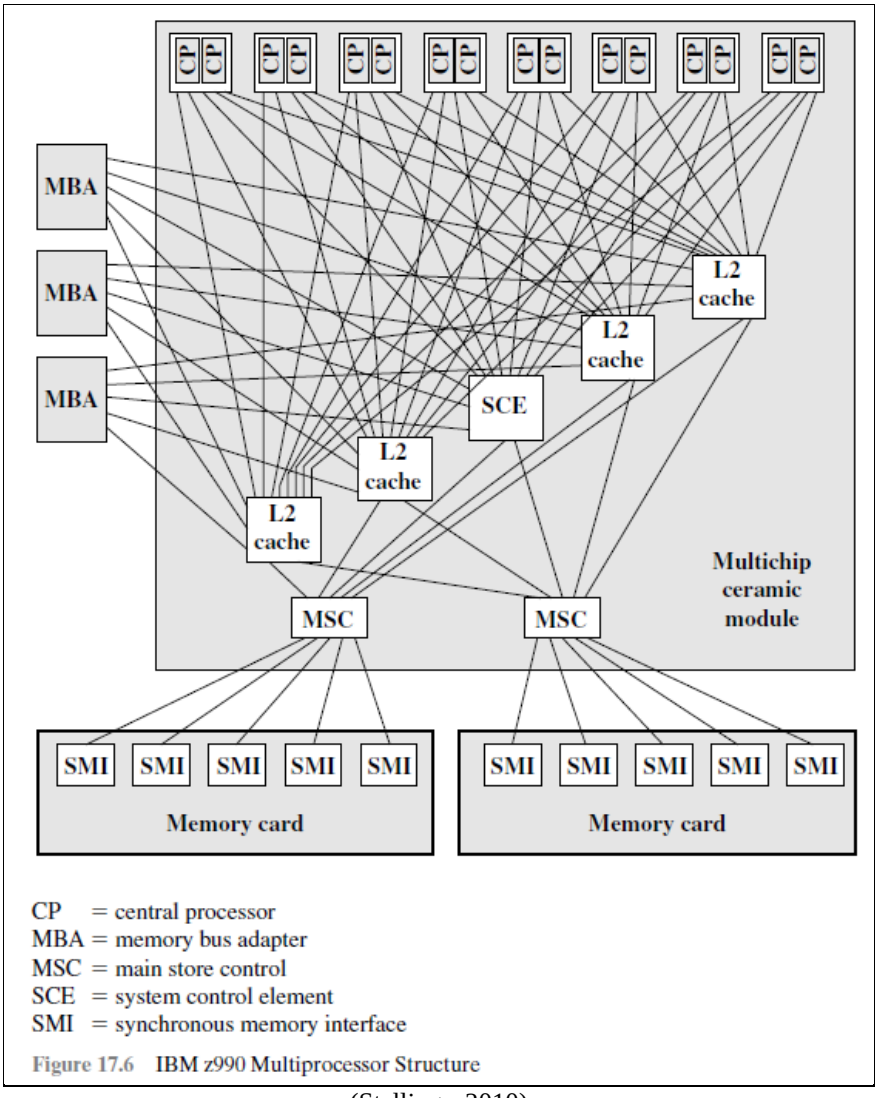


Figure 17.6 IBM z990 Multiprocessor Structure

(Stallings, 2010)

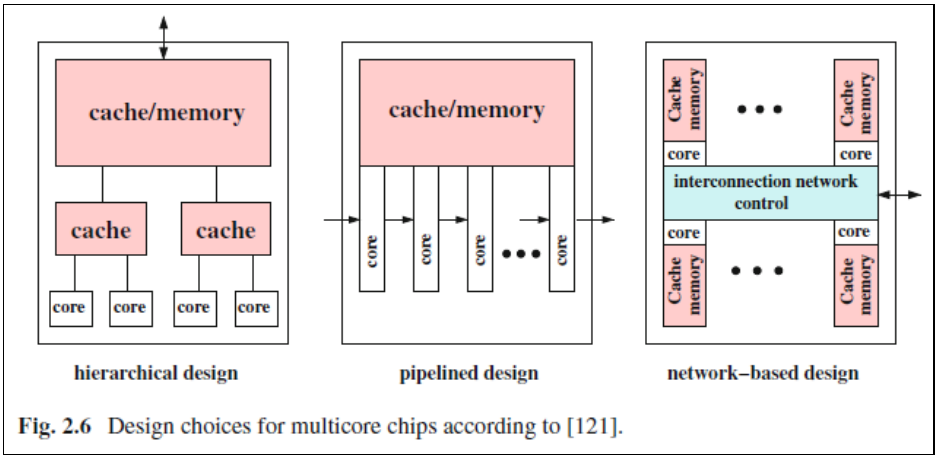


Fig. 2.6 Design choices for multicore chips according to [121].

(Rauber & Rünger, 2013)

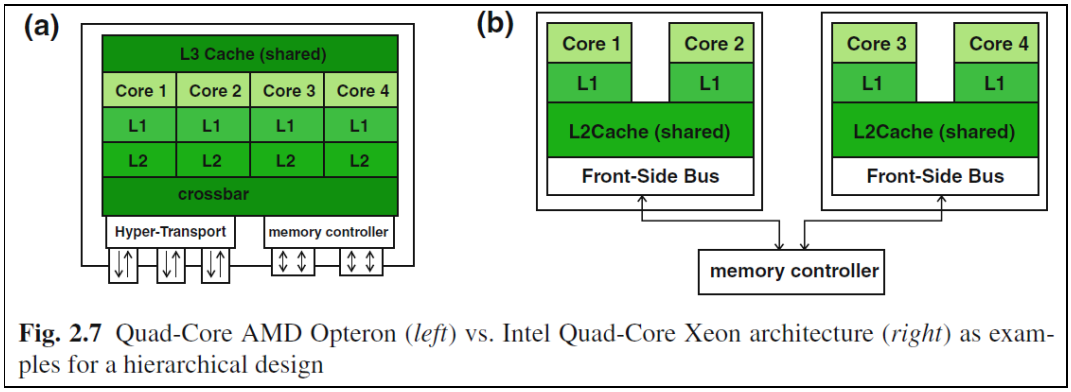


Fig. 2.7 Quad-Core AMD Opteron (*left*) vs. Intel Quad-Core Xeon architecture (*right*) as examples for a hierarchical design

(Rauber & Rünger, 2010)

Semântica de Memória

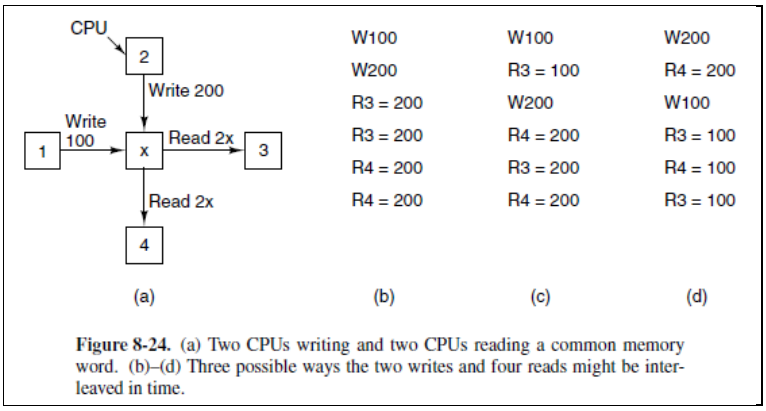


Figure 8-24. (a) Two CPUs writing and two CPUs reading a common memory word. (b)–(d) Three possible ways the two writes and four reads might be interleaved in time.

(Tanenbaum, 2013)

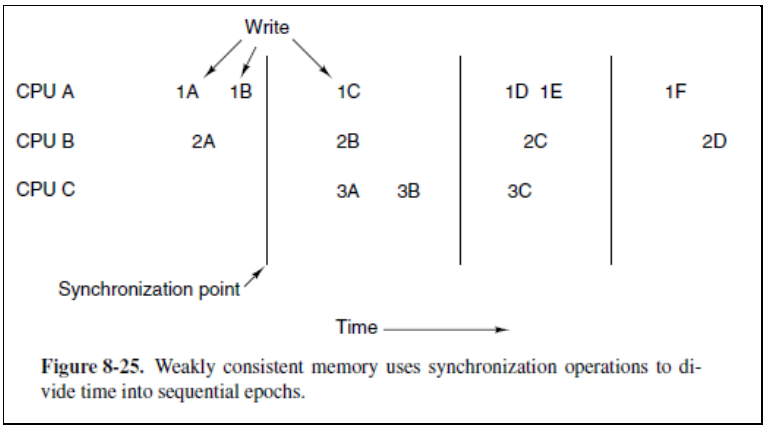


Figure 8-25. Weakly consistent memory uses synchronization operations to divide time into sequential epochs.

(Tanenbaum, 2013)

Coerência de Cache em multiprocessadores

Table 17.1 MESI Cache Line States

	M Modified	E Exclusive	S Shared	I Invalid
This cache line valid?	Yes	Yes	Yes	No
The memory copy is ...	out of date	valid	valid	—
Copies exist in other caches?	No	No	Maybe	Maybe
A write to this line ...	does not go to bus	does not go to bus	goes to bus and updates cache	goes directly to bus

(Stallings, 2013)

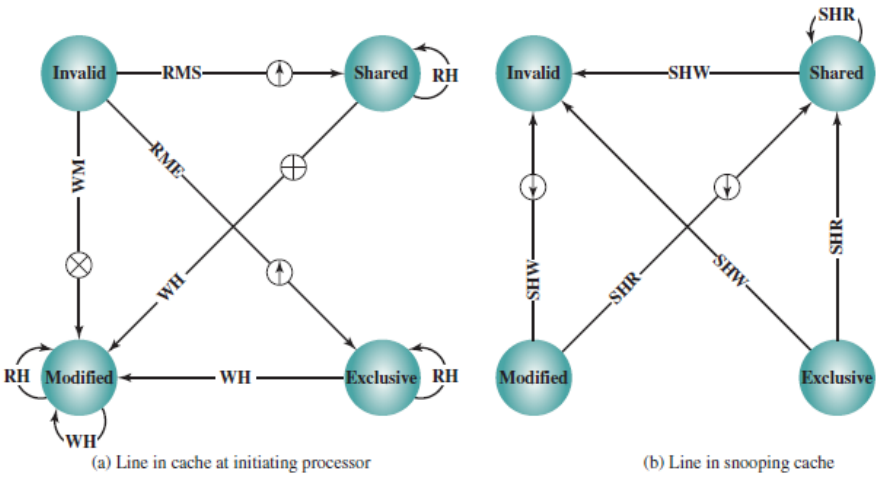


Figure 17.6 MESI State Transition Diagram

(Stallings, 2013)

Implementação em hardware desses protocolos:

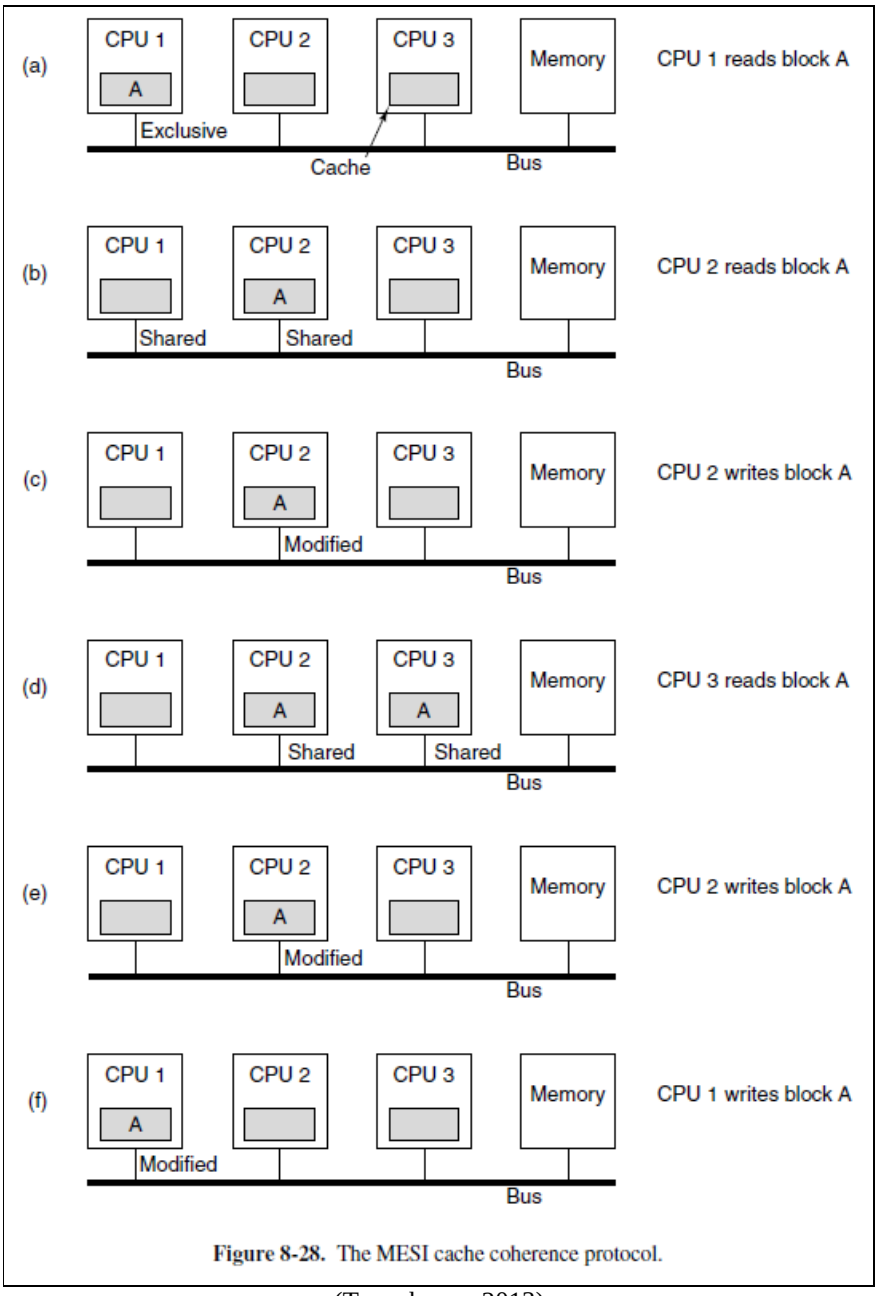


Figure 8-28. The MESI cache coherence protocol.

(Tanenbaum, 2013)

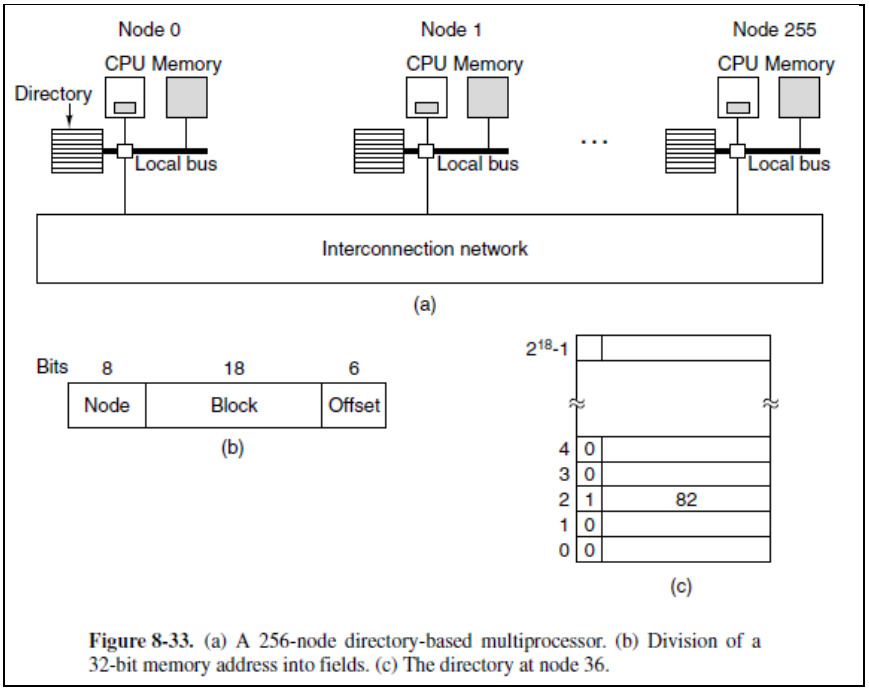
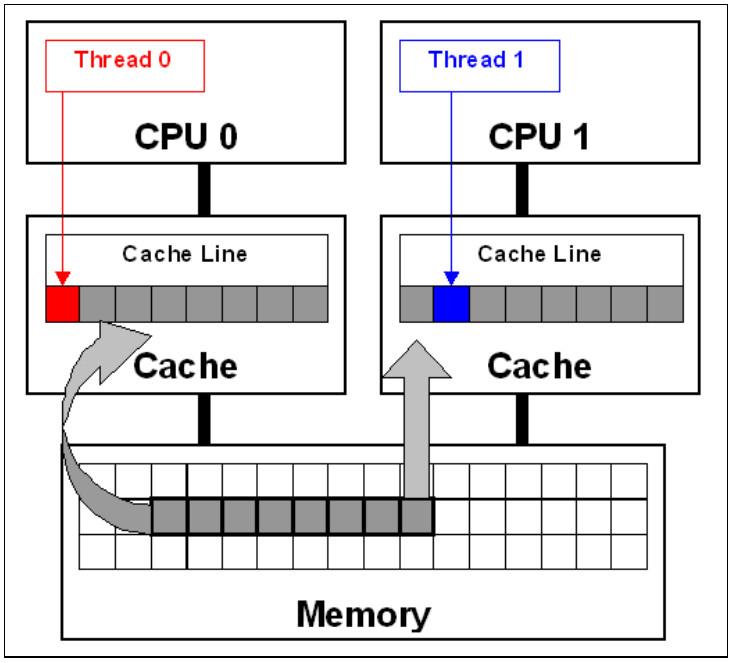


Figure 8-33. (a) A 256-node directory-based multiprocessor. (b) Division of a 32-bit memory address into fields. (c) The directory at node 36.

(Tanenbaum, 2013)

Falso compartilhamento



Fonte WEB

2.3.2 Multicomputadores – MIMD com Memória Distribuída

Clusters

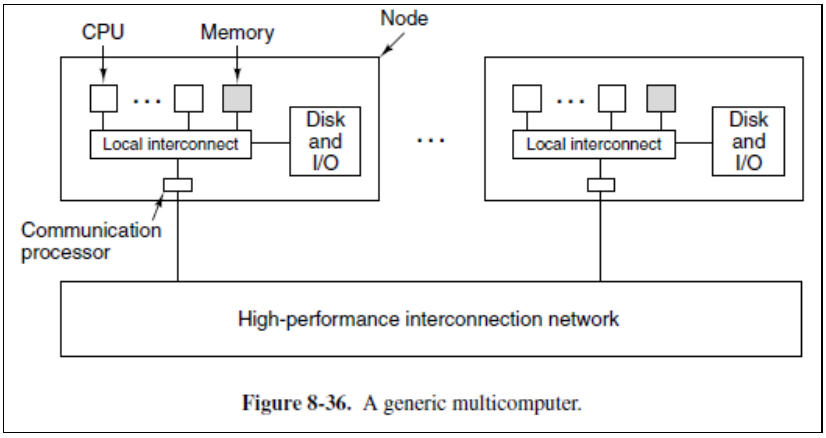


Figure 8-36. A generic multicomputer.

(Tanenbaum, 2013)

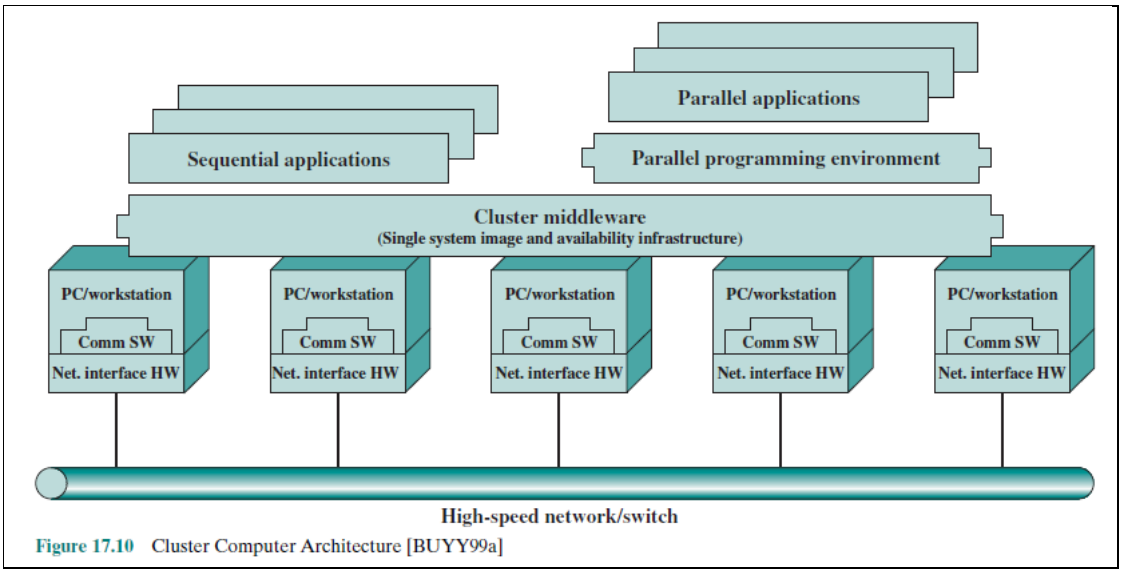


Figure 17.10 Cluster Computer Architecture [BUYY99a]

(Stallings, 2013)

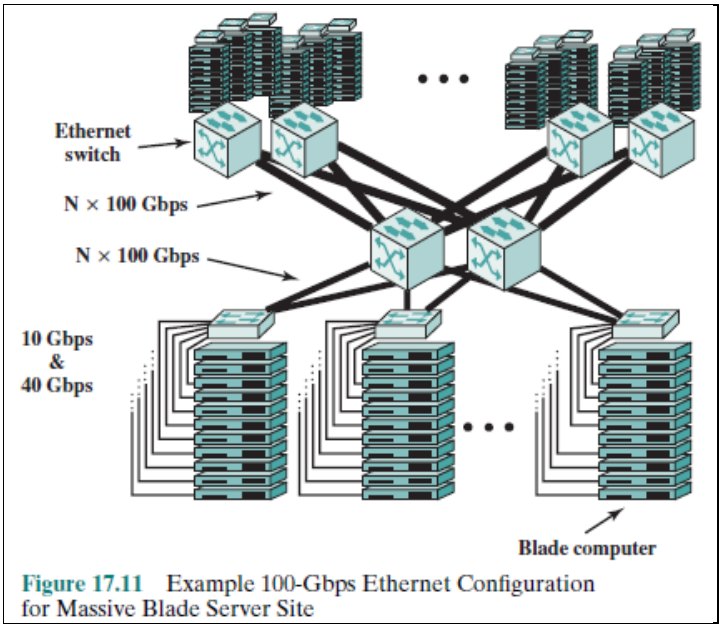


Figure 17.11 Example 100-Gbps Ethernet Configuration for Massive Blade Server Site

(Stallings, 2013)

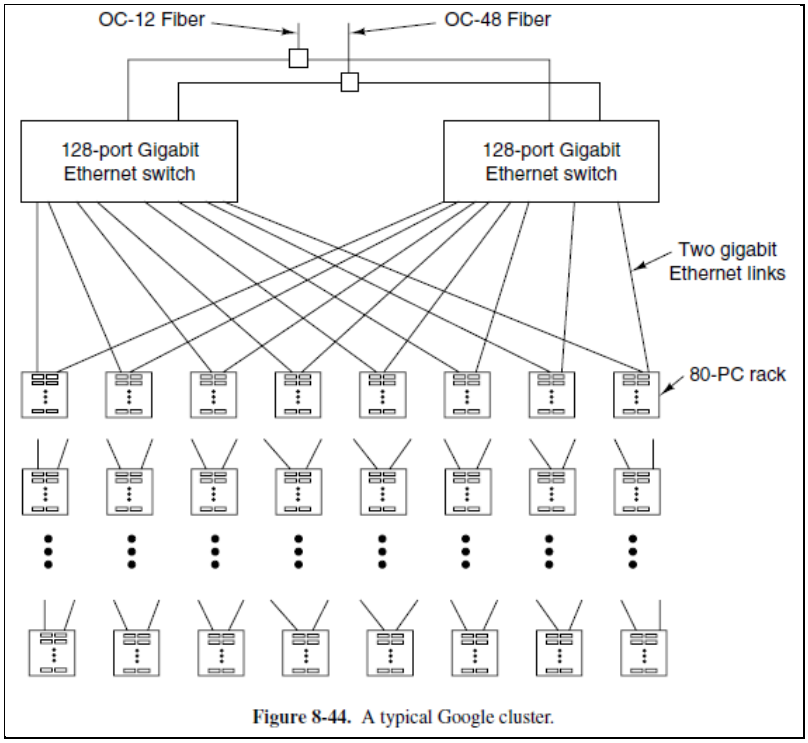


Figure 8-44. A typical Google cluster.

(Tanenbaum, 2013)

Massively Parallel Processors

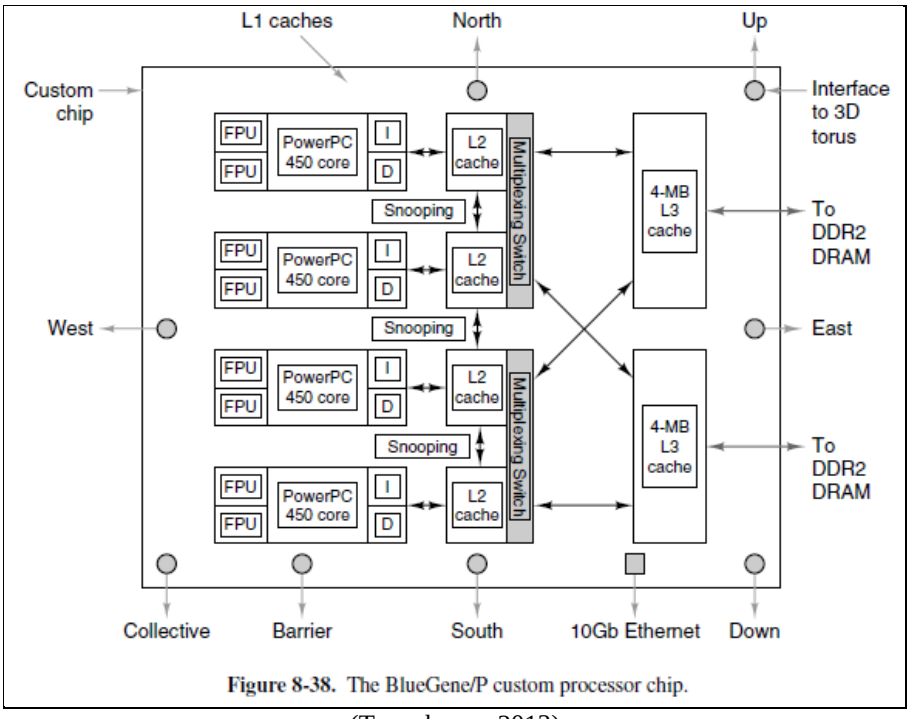


Figure 8-38. The BlueGene/P custom processor chip.

(Tanenbaum, 2013)

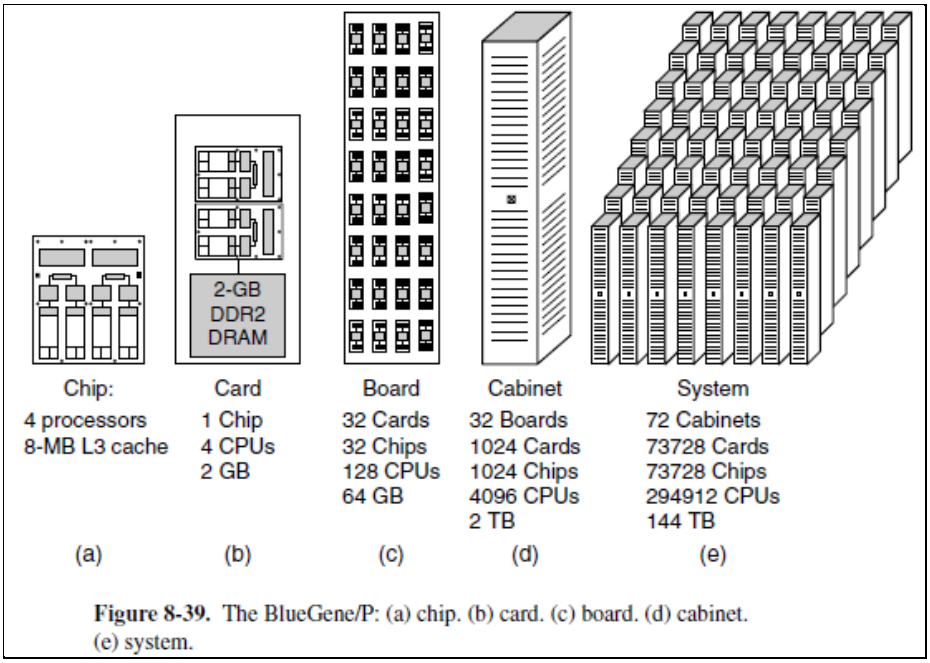


Figure 8-39. The BlueGene/P: (a) chip. (b) card. (c) board. (d) cabinet. (e) system.

(Tanenbaum, 2013)

2.4 SIMD

Fonte WEB

```
      DO 100 I = 1, N
      DO 100 J = 1, N
      C(I, J) = 0.0
      DO 100 K = 1, N
      C(I, J) = C(I, J) + A(I, K) + B(K, J)
100   CONTINUE
```

(a) Scalar processing

```
      DO 100 I = 1, N
      C(I, J) = 0.0 (J = 1, N)
      DO 100 K = 1, N
      C(I, J) = C(I, J) + A(I, K) + B(K, J) (J = 1, N)
100   CONTINUE
```

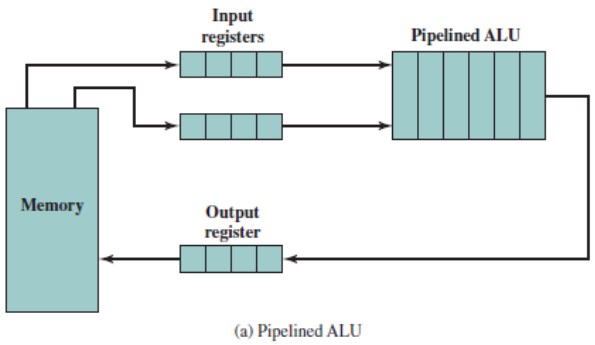
(b) Vector processing

```
      DO 50 J = 1, N − 1
      FORK 100
50    CONTINUE
      J = N
100   DO 200 I = 1, N
      C(I, J) = 0.0
      DO 200 K = 1, N
      C(I, J) = C(I, J) + A(I, K) + B(K, J)
200   CONTINUE
      JOIN N
```

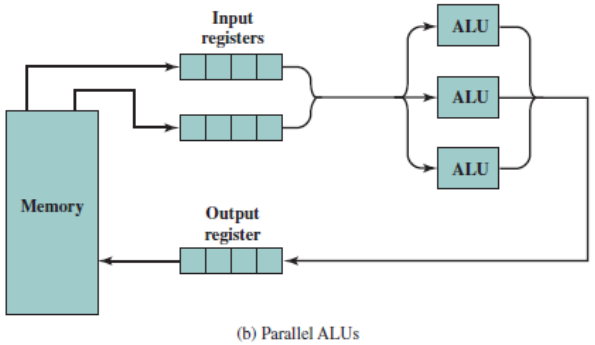
(c) Parallel processing

Figure 17.14 Matrix Multiplication (C = A × B)

(Stallings, 2013)



(a) Pipelined ALU



(b) Parallel ALUs

Figure 17.15 Approaches to Vector Computation

(Stallings, 2013)

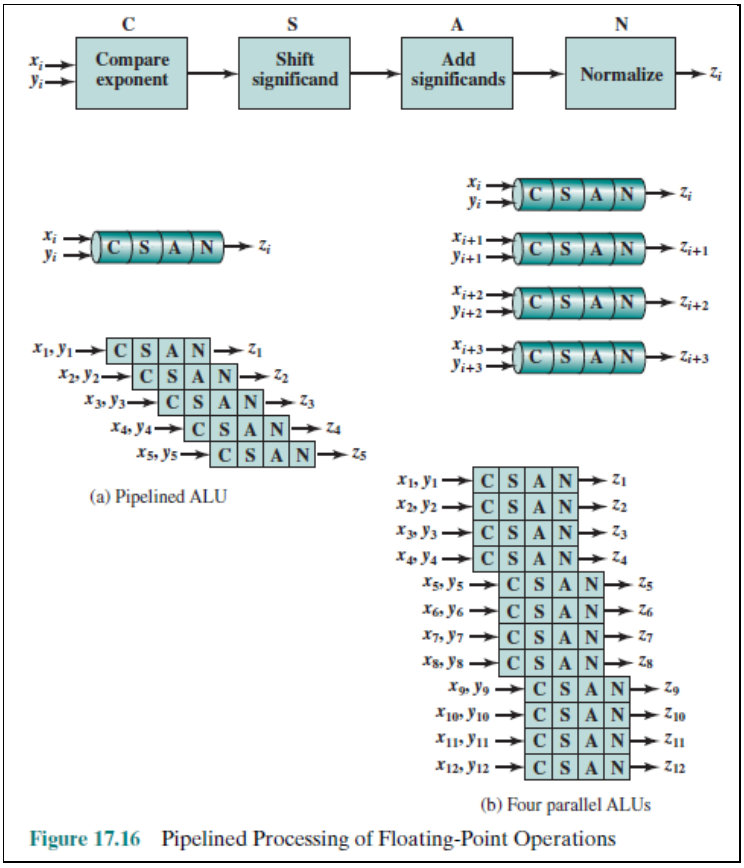
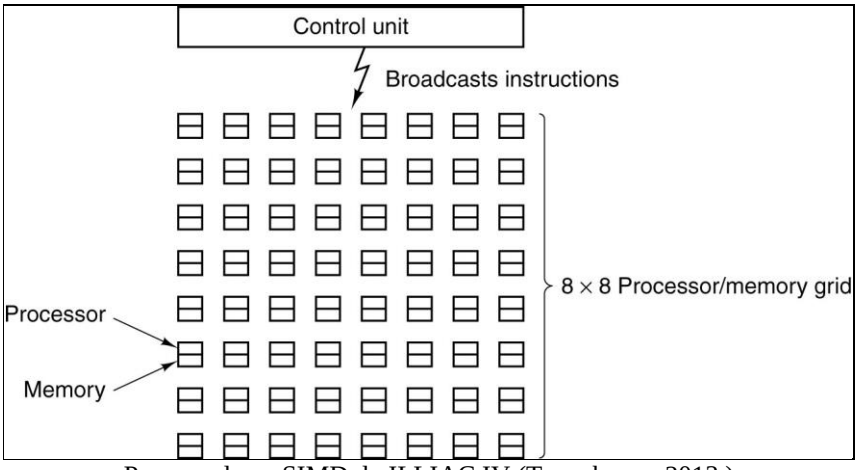


Figure 17.16 Pipelined Processing of Floating-Point Operations

(Stallings, 2013)



Processadores SIMD do ILLIAC IV (Tanenbaum, 2013)

2.5 GPUs

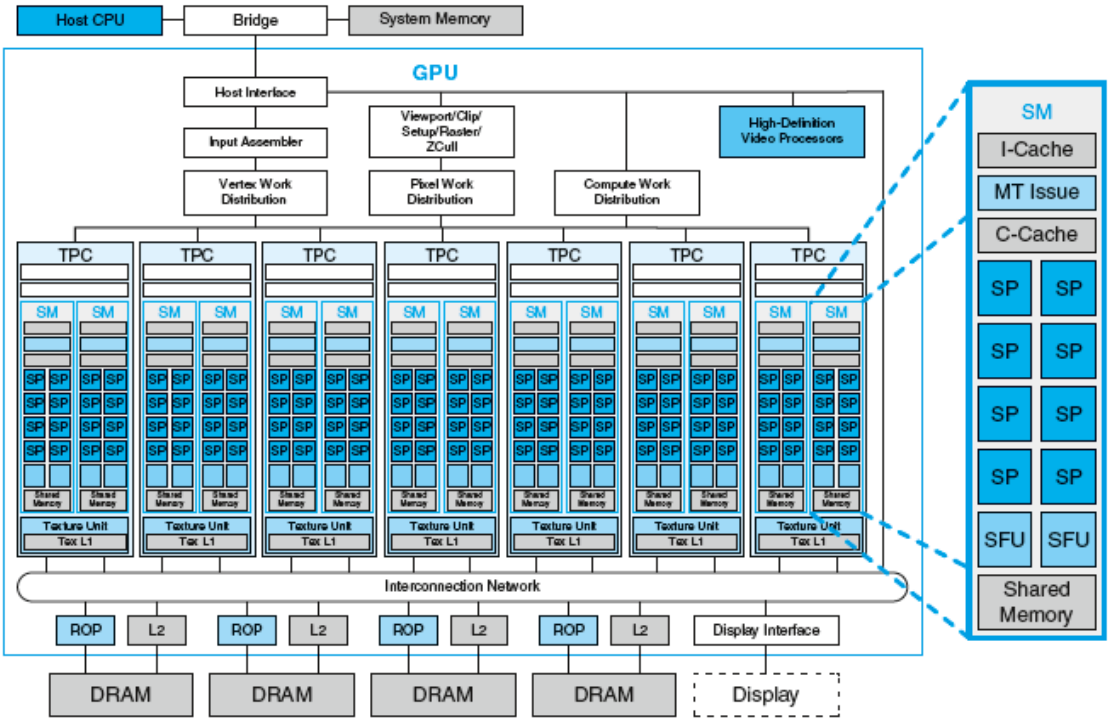
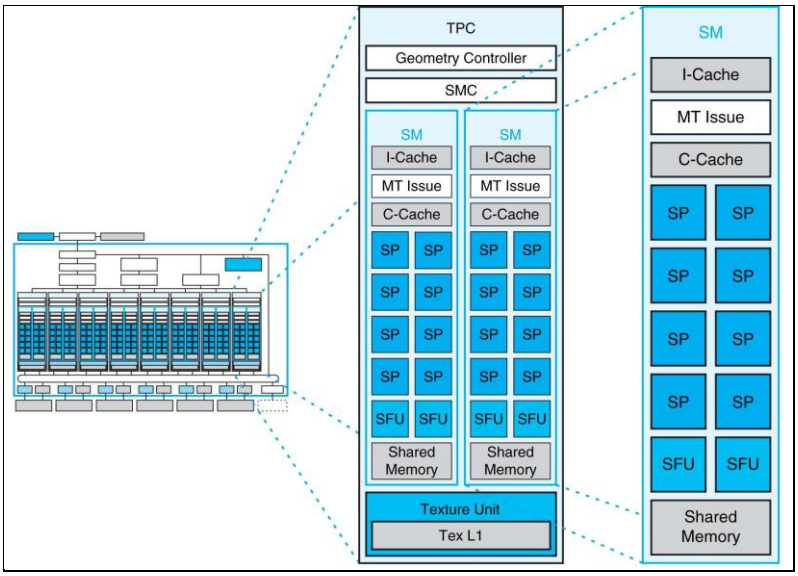


FIGURE C.2.5 Basic unified GPU architecture. Example GPU with 112 streaming processor (SP) cores organized in 14 streaming multiprocessors (SMs); the cores are highly multithreaded. It has the basic Tesla architecture of an NVIDIA GeForce 8800. The processors connect with four 64-bit-wide DRAM partitions via an interconnection network. Each SM has eight SP cores, two special function units (SFUs), instruction and constant caches, a multithreaded instruction unit, and a shared memory.

(Patterson & Hennessy, 2014)



(Patterson & Hennessy, 2014)

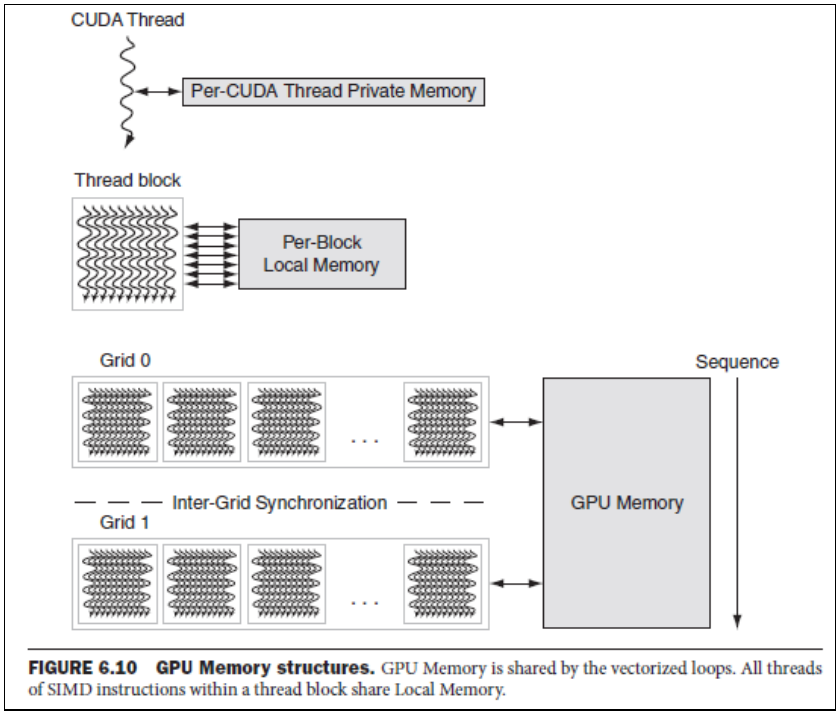
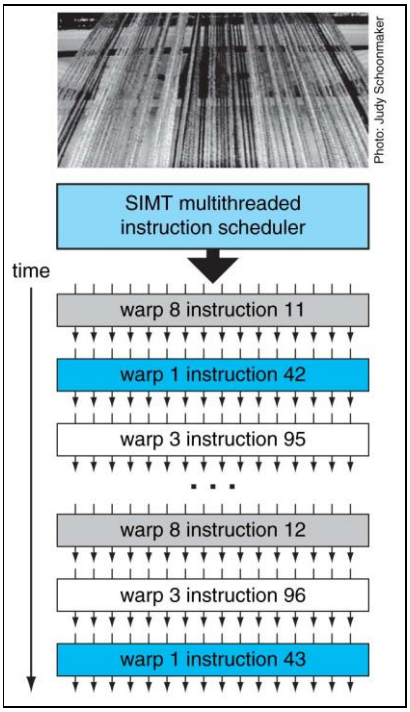


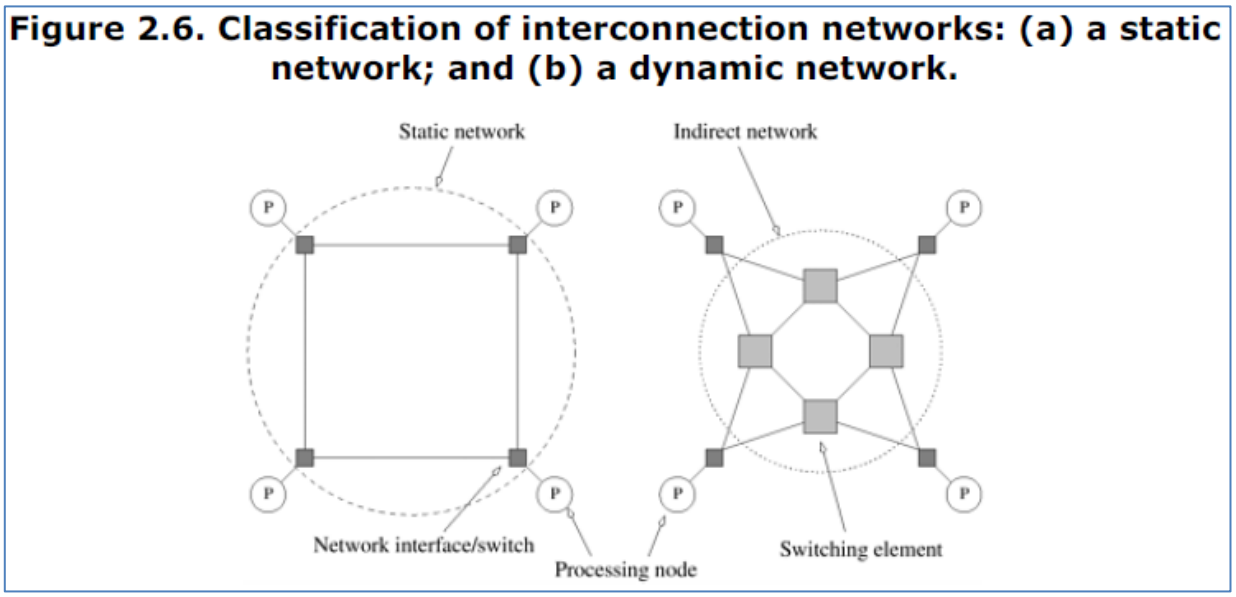
FIGURE 6.10 GPU Memory structures. GPU Memory is shared by the vectorized loops. All threads of SIMD instructions within a thread block share Local Memory.

(Patterson & Hennessy, 2014)

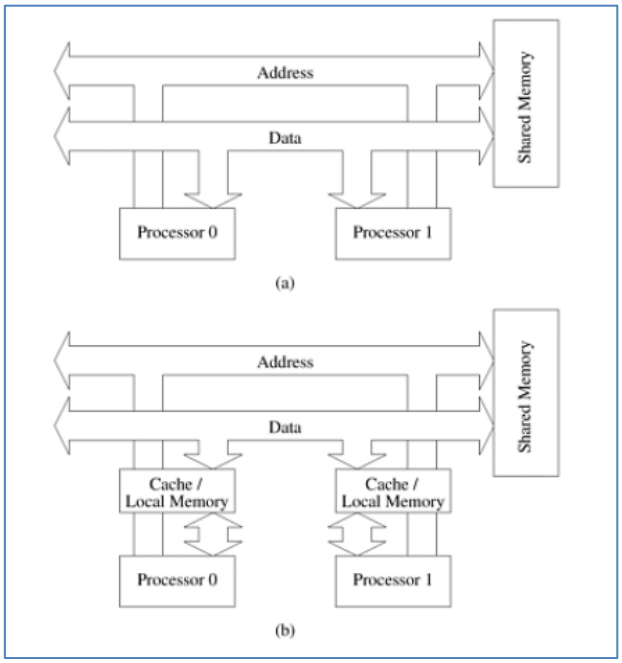


(Patterson & Hennessy, 2014)

2.6 Redes de Conexão nas Arquiteturas Paralelas



(*Grama et al., 2003)*



(Grama et al., 2003), Seção 2.4.3, Figura 2.7

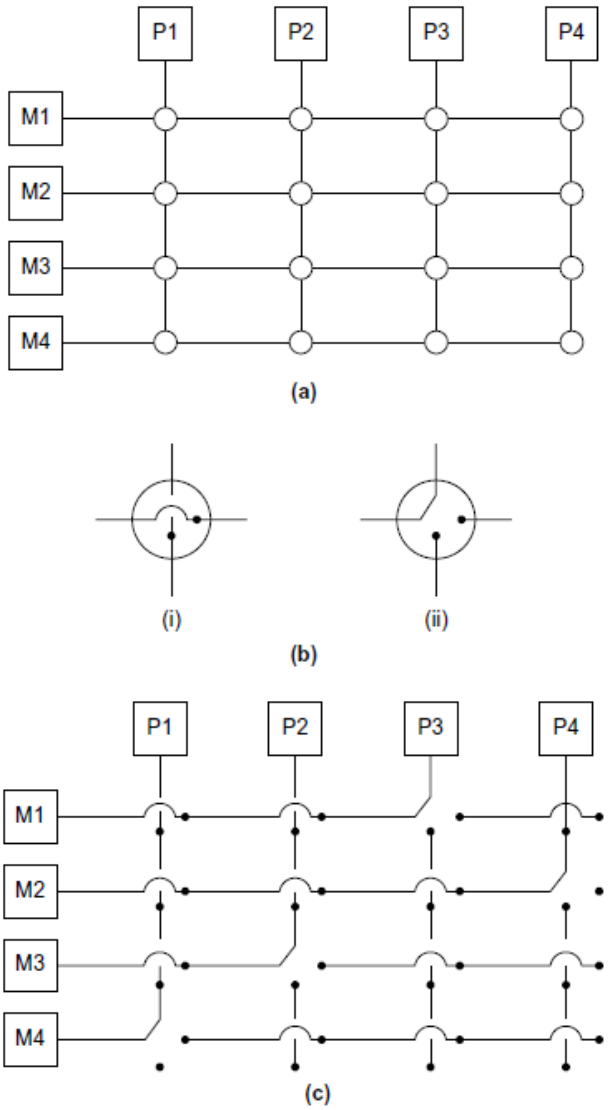
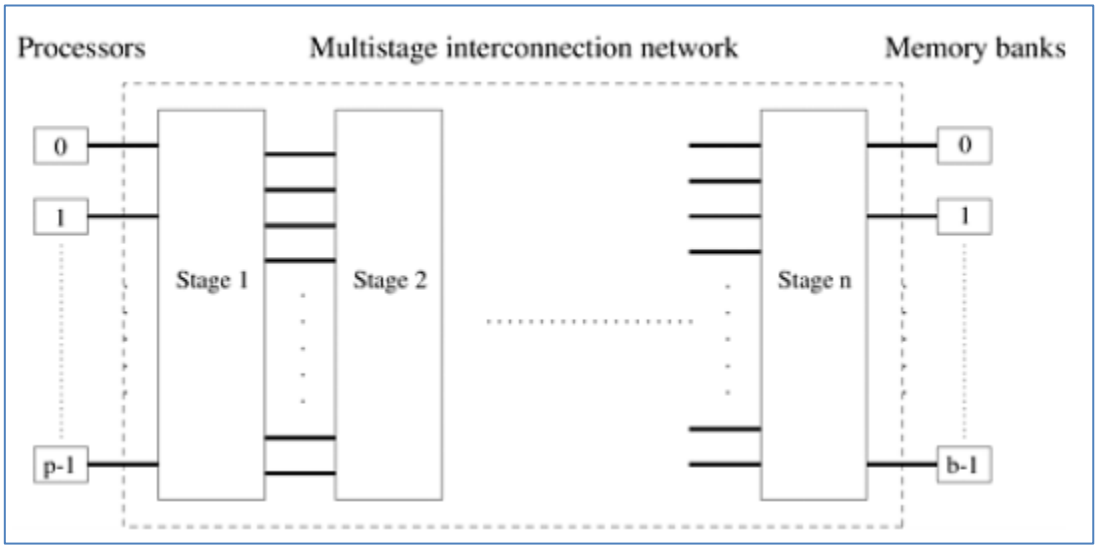


FIGURE 2.7

(a) A crossbar switch connecting four processors (P_i) and four memory modules (M_j); (b) configuration of internal switches in a crossbar; (c) simultaneous memory accesses by the processors

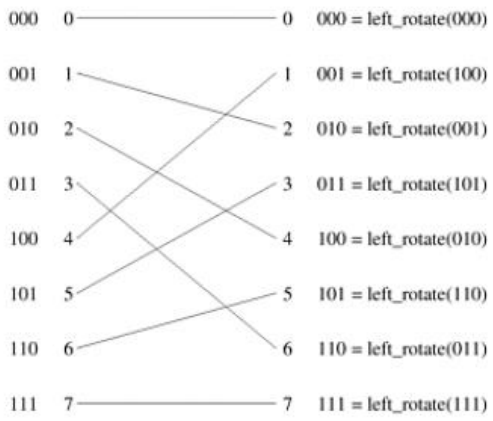
(Pacheco, 2011)

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



(Grama et al., 2003)

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

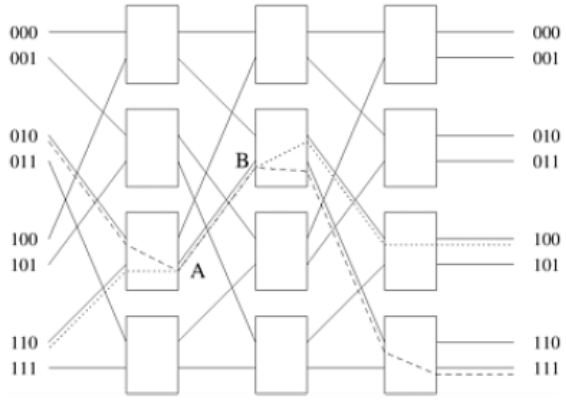


(Grama et al., 2003)

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

(Grama et al., 2003)

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.



(Grama et al., 2003)

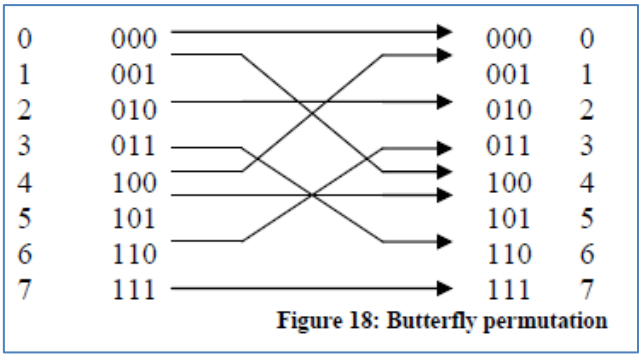
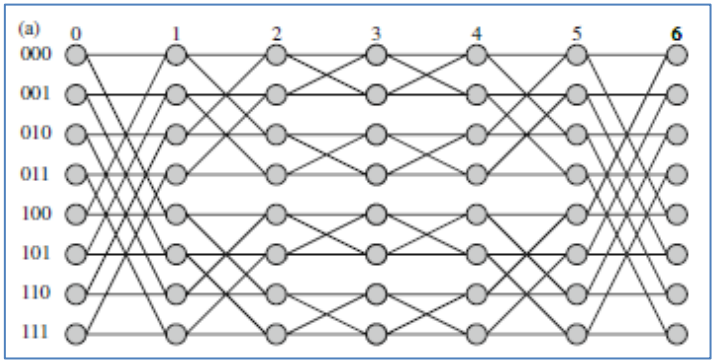


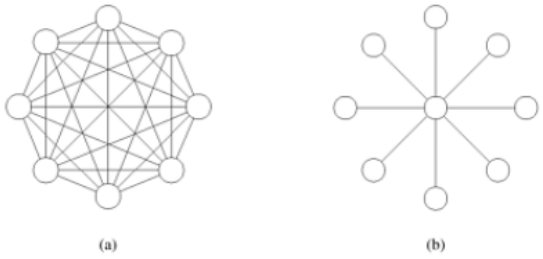
Figure 18: Butterfly permutation

Material WEB, Unit 3, pág 57, Figura 18



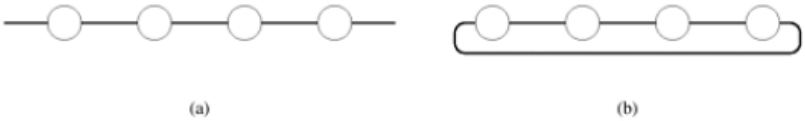
(Rauber & Rünger, 2010) - Figura 2.20 Rede Benes não bloqueante

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



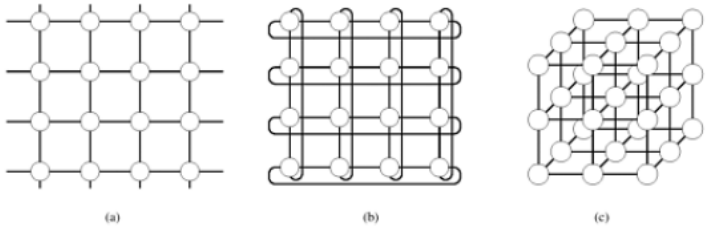
(*Grama et al. 2003)*

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



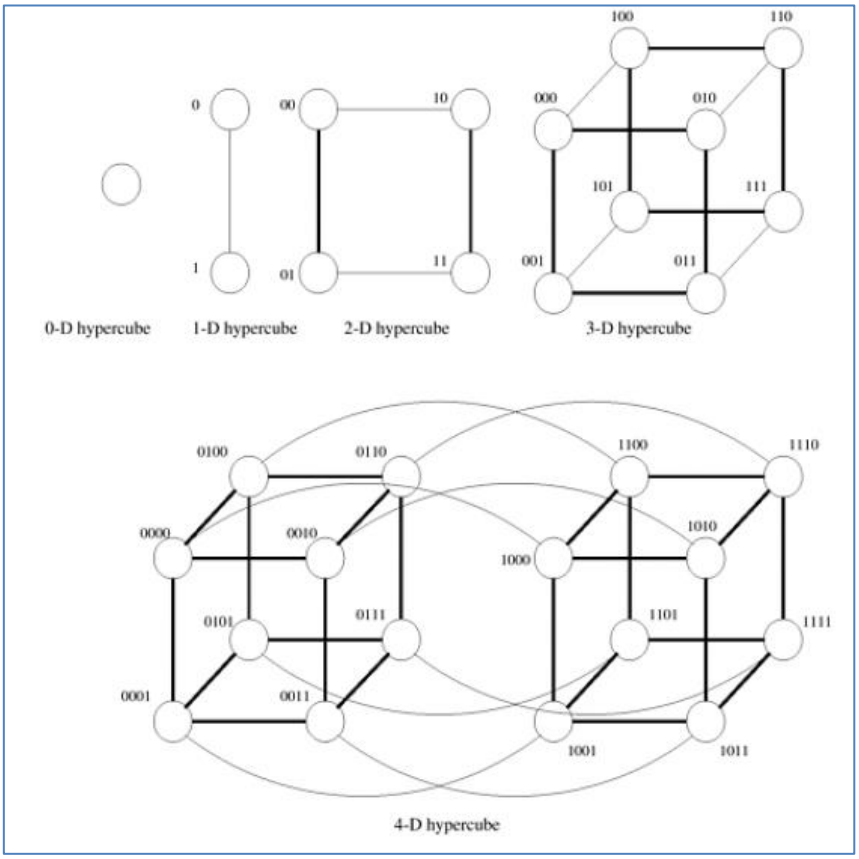
(*Grama et al. 2003)*

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.



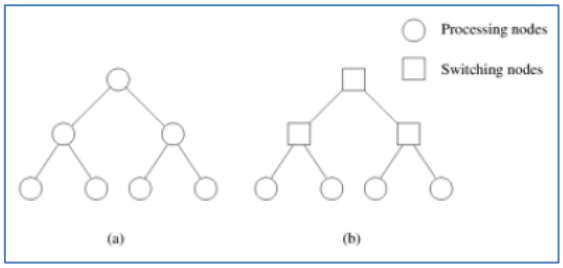
(*Grama et al. 2003)*

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



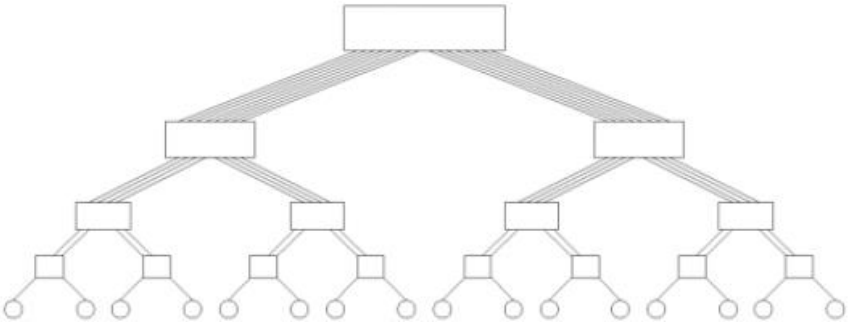
(*Grama et al. 2003)*

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



(*Grama et al. 2003)*

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



(*Grama et al. 2003)*