

# Párhuzamos programozási platformok



# Parallel számítógép részei

- Hardver
  - Több processzor
  - Több memória
  - Kapcsolatot biztosító hálózat
- Rendszer szoftver
  - Párhuzamos operációs rendszer
  - Konkurenciát biztosító programozási konstrukciók
- Alkalmazás szoftver
  - Párhuzamos algoritmusok

Cél: úgy használjuk a hardvert, a rendszer és az alkalmazás programot, hogy

- Sebességnövekedést érjünk el  $T_p = T_s / p$
- Nagy mennyiségű memóriát igénylő feladatokat is meg tudunk oldani



# Párhuzamos számítási platform

- Logikai felépítés
  - Ahogy a felhasználó látja a gépet, amit a rendszer szoftver „mutat”
- Fizikai felépítés
  - Az aktuális hardver architektúra
- A fizikai architektúra nagyrészt független a logikai felépítéstől

# Logikai felépítés elemei

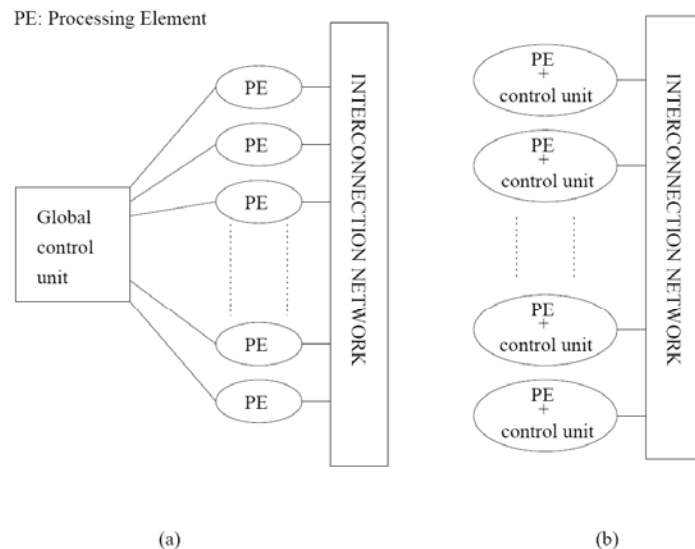
## ■ Vezérlő mechanizmus

### □ SISD/SIMD/MIMD/MISD:

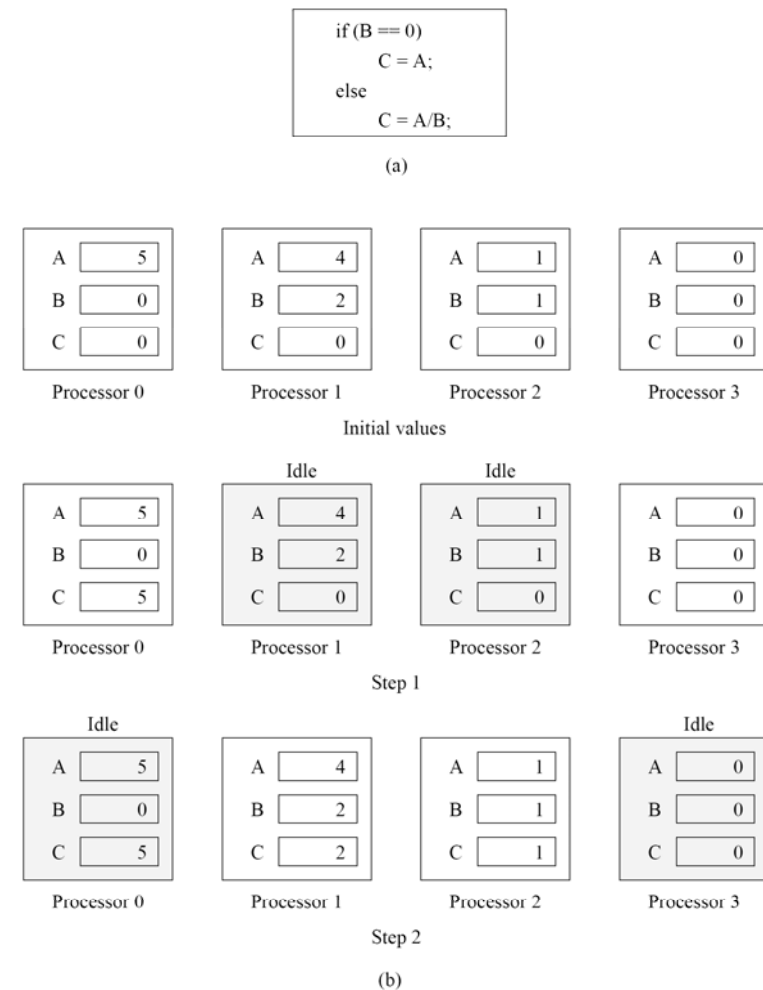
Single/Multiple Instruction Stream  
& Single/Multiple Data Stream

### □ SPMD:

Single Program Multiple Data



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

# Logikai felépítés típusai

- Kommunikációs modell szerint

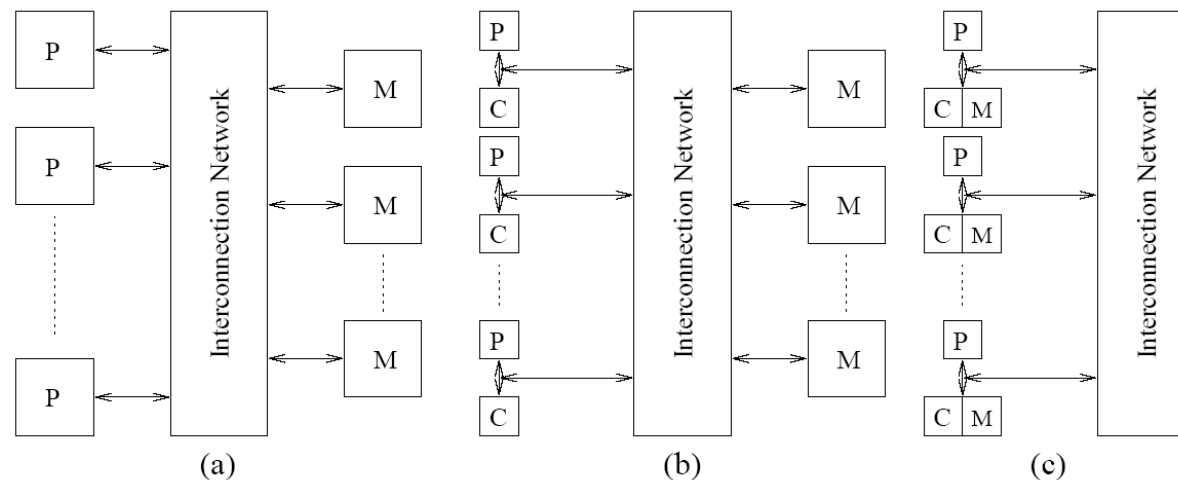
- Shared-Address Space - „Megosztott címterű”

- UMA/NUMA/ccNUMA

- Message-Passing  
Üzenet-továbbítás

- MPI

- P1/M



**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.



# Fizikai szervezés

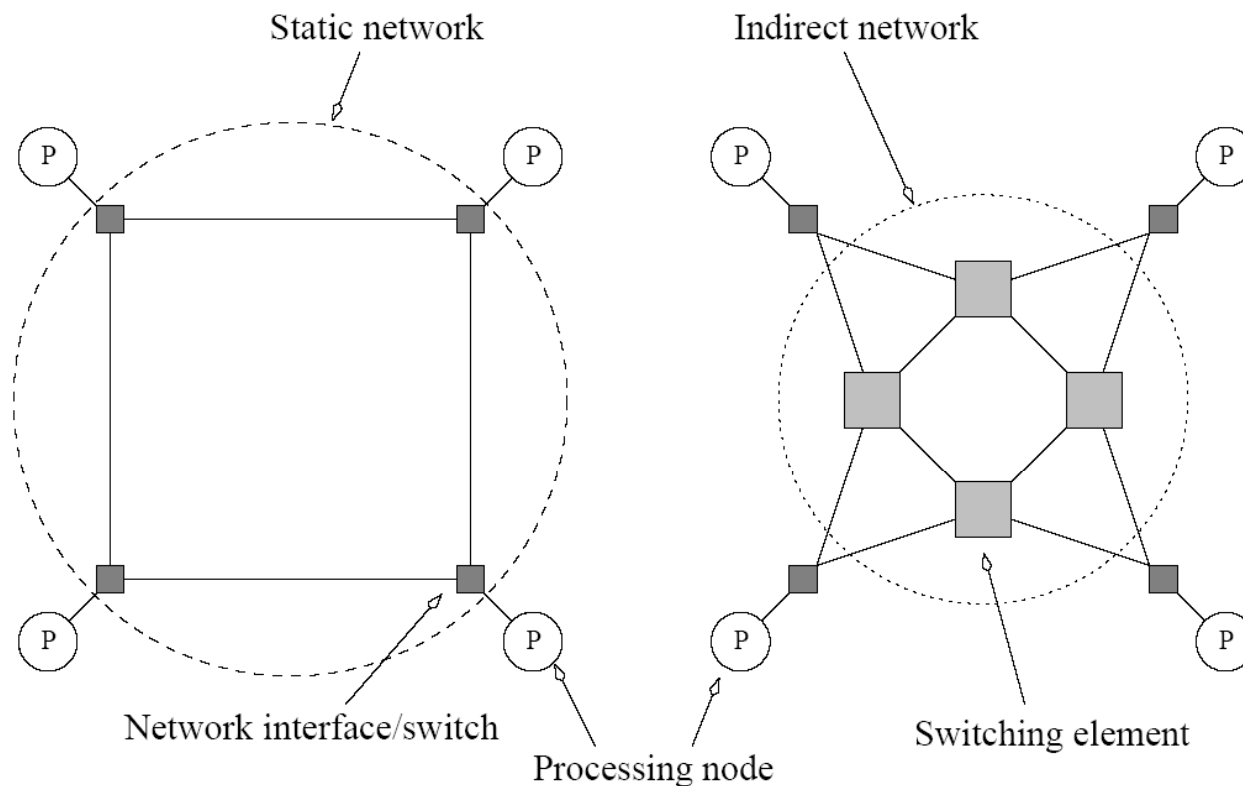
- Ideális párhuzamos számítógép architektúra
  - PRAM: Parallel Random Access Machine
- PRAM modellek
  - EREW/ERCW/CREW/CRCW
    - Exkluzív/konkurens olvasás és/vagy írás
- Konkurens írás a következőket jelentheti
  - Közös: párhuzamos írás engedélyezett, ha minden processzor ugyanazt az azonos értéket szeretné írni
  - Tetszőleges: valamelyik írhat, a többi hibát kap
  - Prioritásos: processzorok prioritás listába rendezettek és ennek megfelelően írnak
  - Összegző: a mennyiségek összege kerül írásra



# Fizikai felépítés

- Kapcsolt hálózatok (Interconnection Networks: ICN)
  - Processzor-processzor és processzor-memória kapcsolatot biztosít
  - Hálózatok osztályozása:
- Statikus hálózatok, vagy direkt hálózatok:  
pont-pont kapcsolatok
- Dinamikus hálózatok, vagy indirekt hálózatok:  
a hálózati kapcsolóelemek kapcsolják a processzorokat

# Statikus és dinamikus ICN



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

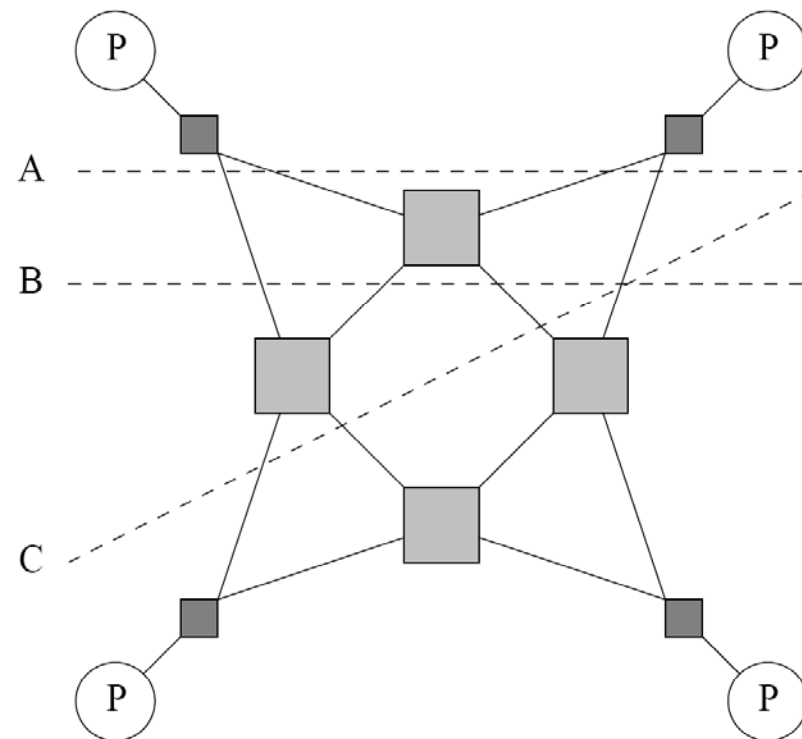




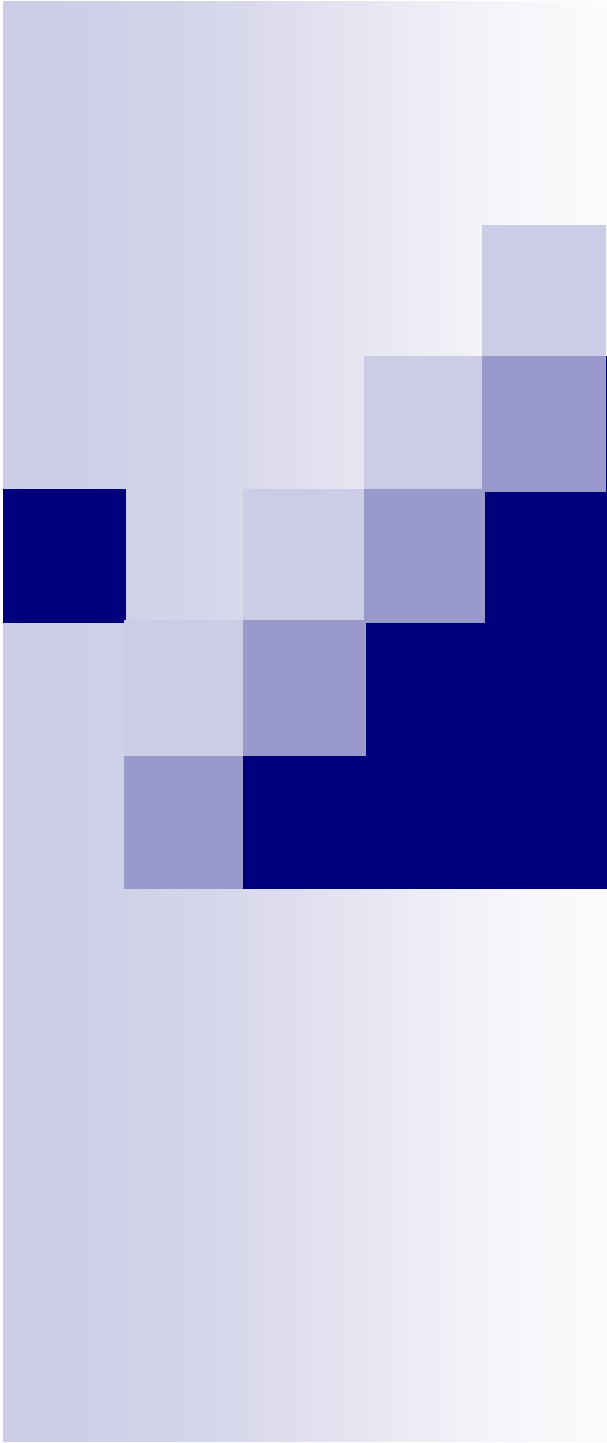
# Hálózatok mérhető jellemzői

- Átmérő
  - Maximális távolság két csomópont között – jobb, ha kisebb
- Összekapcsolódás (connectivity)
  - A legkevesebb él, amit eltávolítva, két hálózatra esik szét a rendszer – jobb, ha nagyobb
  - Az utak multiplicitásának jellemzője
- Félbevágási minimális szélesség (bisection width)
  - A legkevesebb él, amelynek eltávolításával két azonos méretű félre bontható a háló – jobb, ha nagyobb
- Félbevágási minimális sáv szélesség
  - Az él súlya megmutatja, hogy mennyi adat haladhat át rajta
  - A minimális kommunikációs mennyiség, amit két tetszőleges fél között biztosított – jobb, ha nagyobb
- Költség
  - Kapcsolatok száma – kevesebb jobb

# Dinamikus hálózati jellemző



**Figure 2.20** Bisection width of a dynamic network is computed by examining various equipartitions 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.

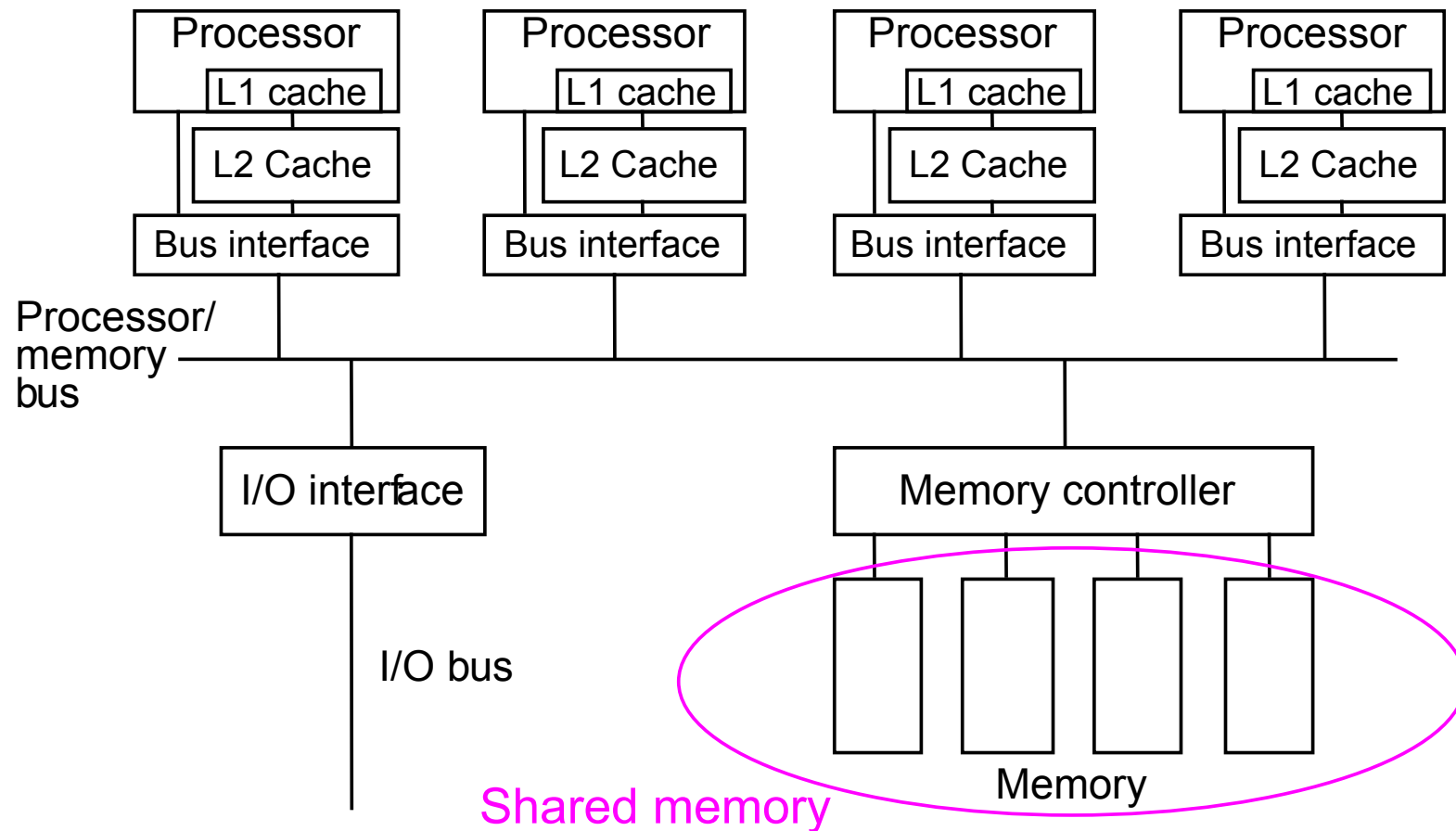


# Párhuzamos számítógép topológiák

Megosztott memória

Szétosztott memória

# Quad Pentium - Shared Memory Multiprocessor



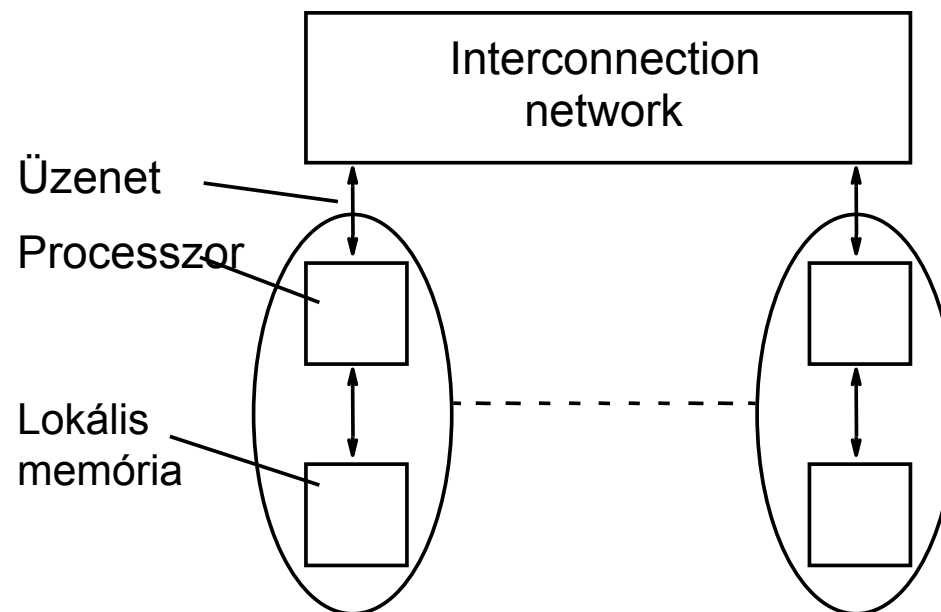


# Párhuzamos programozási lehetőségek

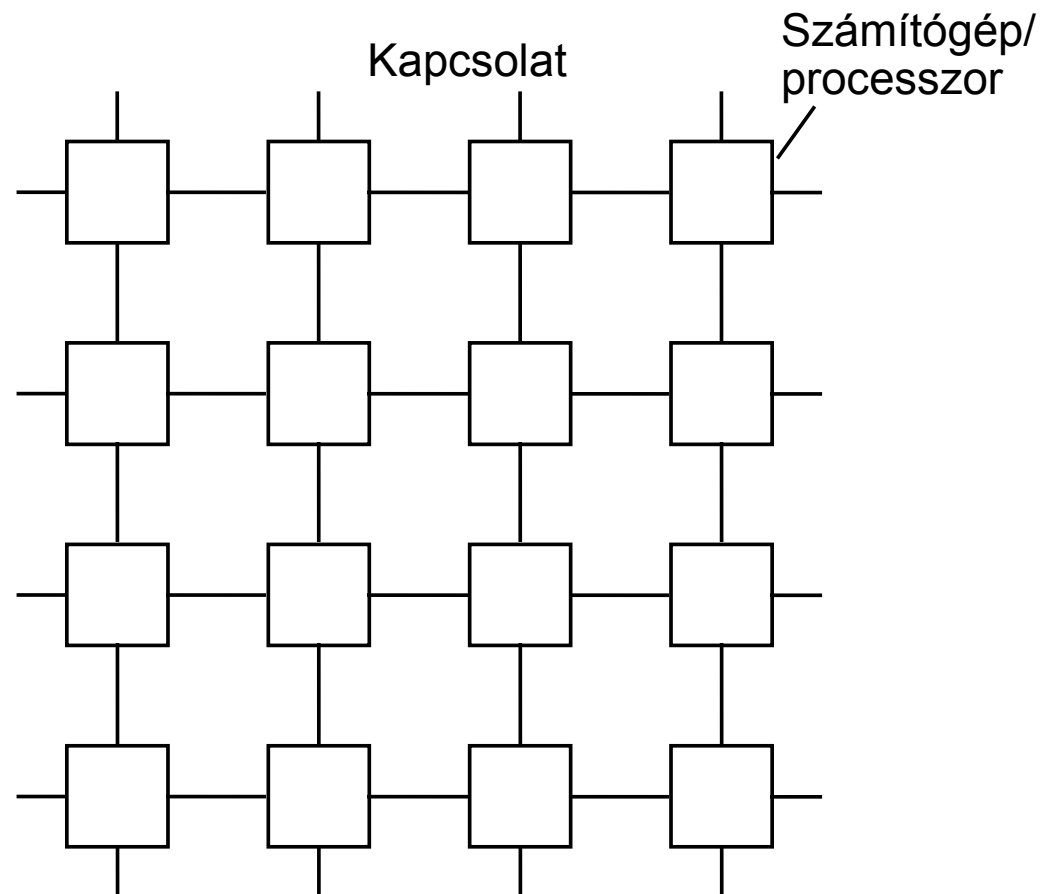
- *Szálakkal* – a szálon kívül definiált változókat is  
Példa: Pthreads
- Soros programnyelv *preprocesszor fordítási direktívákkal*, amelyek megosztott változókat és párhuzamosságot biztosítanak  
Példa: OpenMP - OpenMP compiler
- Soros programnyelv *extra nyelvi elemekkel*, amelyek megosztott változókat és párhuzamosságot biztosítanak  
Példa UPC (Unified Parallel C) - UPC compiler.
- Párhuzamos programozási nyelv párhuzamosságot biztosító szintaktikával – a compiler készíti futatható kódot minden processzorra (kutatási terület)

# Üzenet-küldés (MPI)

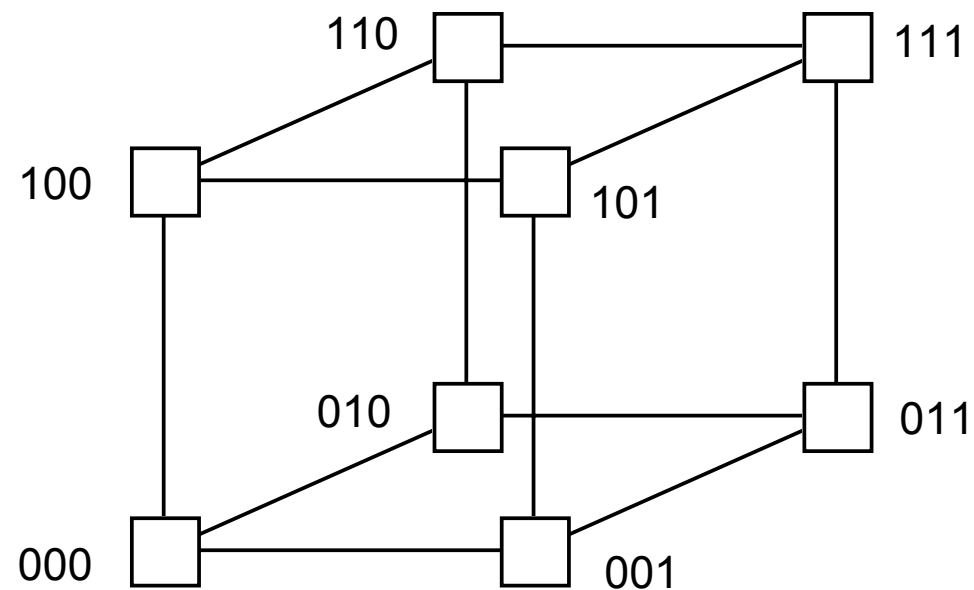
Teljes, önálló számítógépek összekapcsolva



# Két dimenziós háló

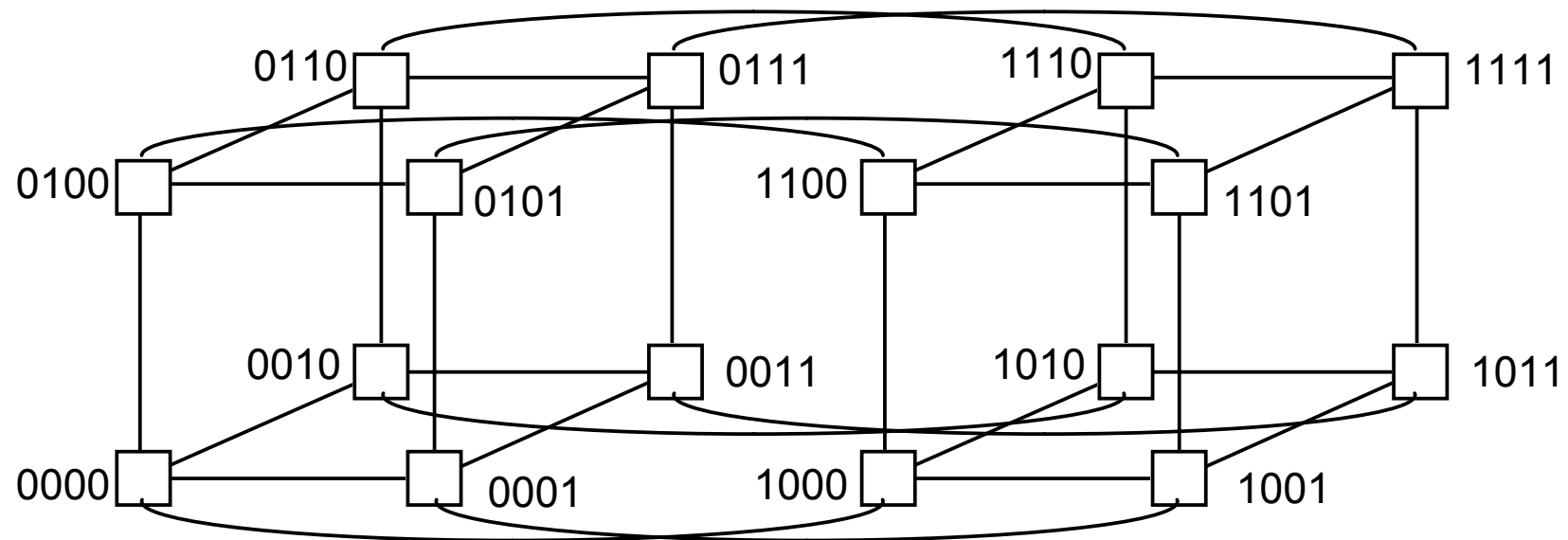


# 3D-s hiperkocka

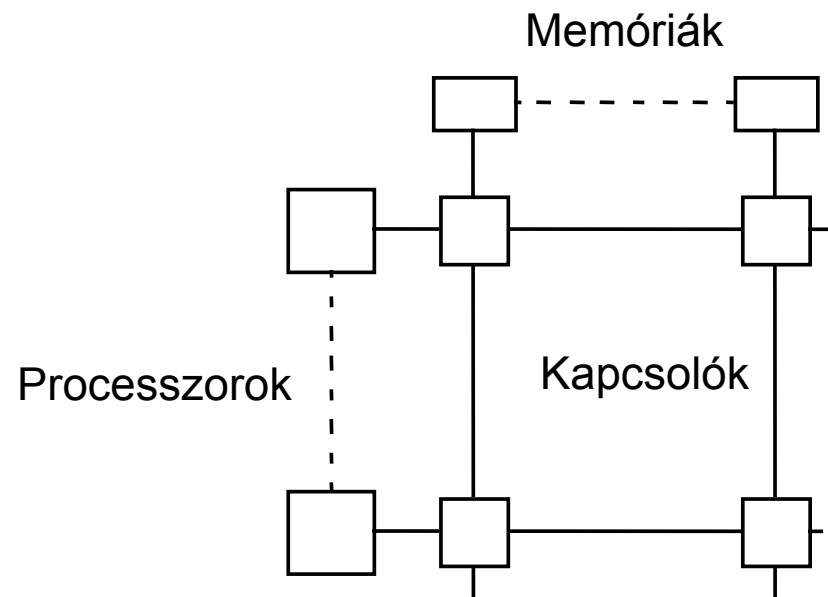




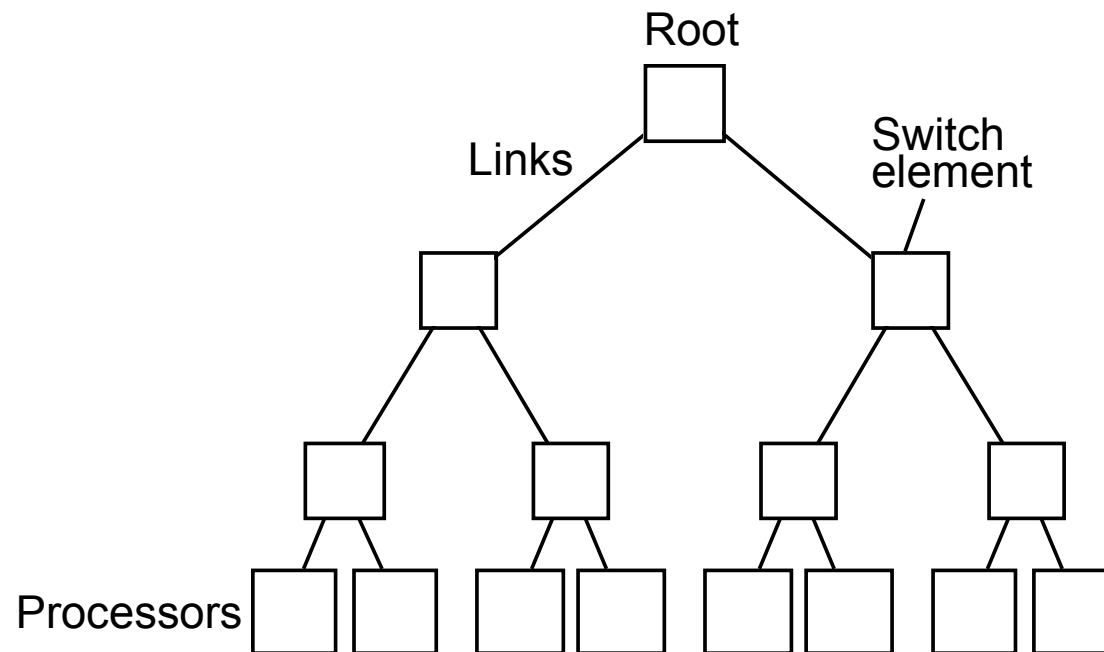
# 4D-s hiperkocka



# Crossbar kapcsolat



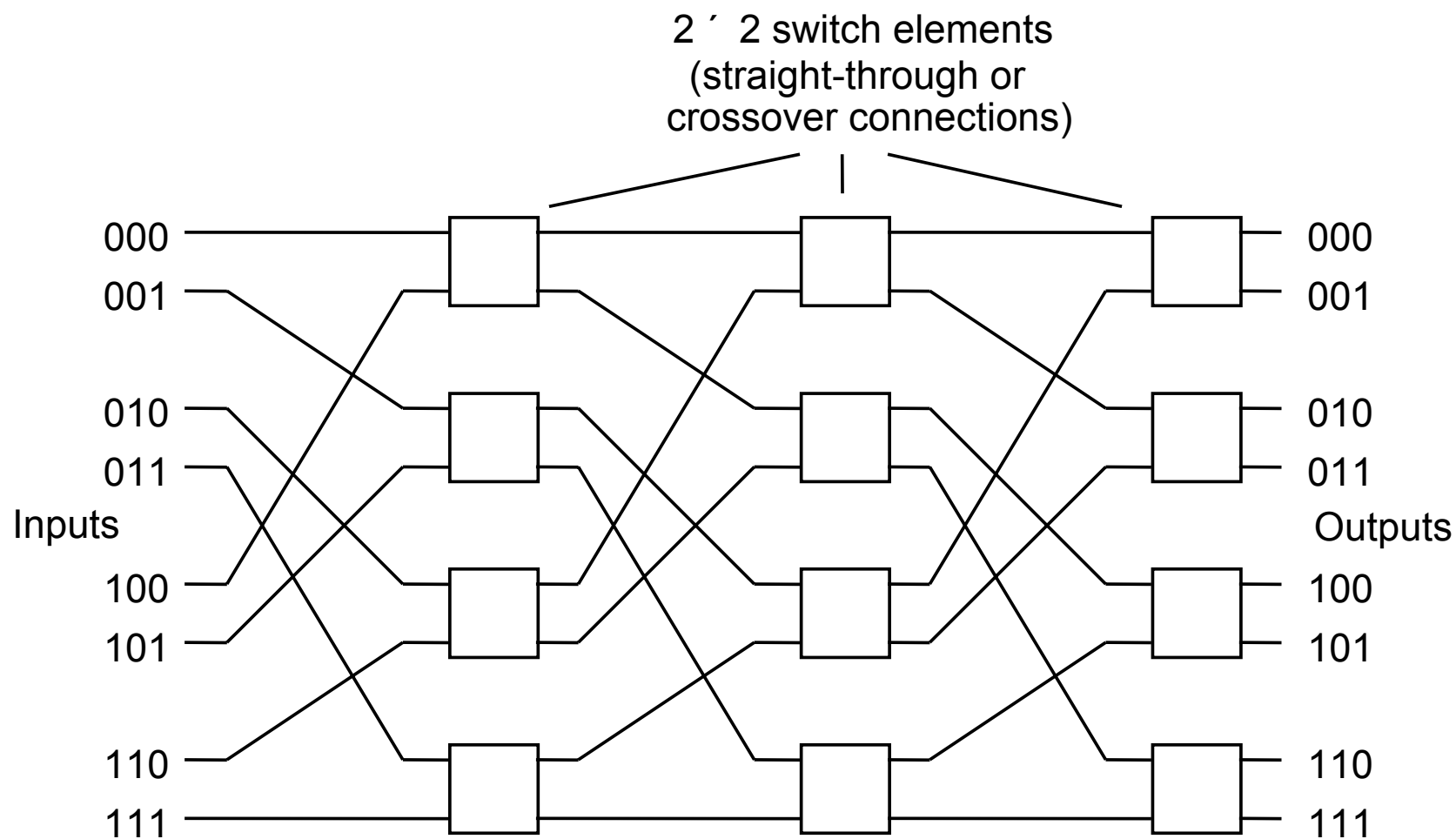
# Fa topológia





# Többállapotú kapcsolt hálózatok

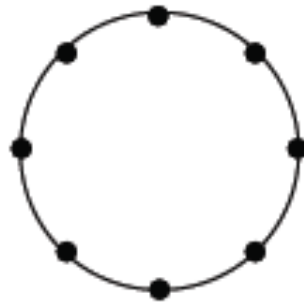
Példa: Omega háló



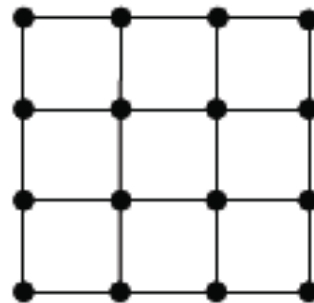
# Topológiák - összefoglaló



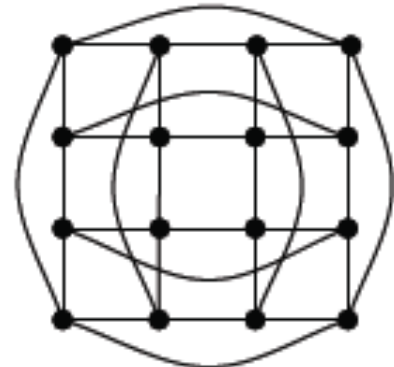
*1-D mesh*



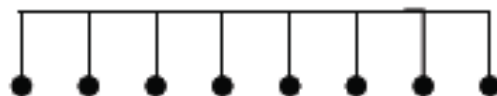
*1-D torus (ring)*



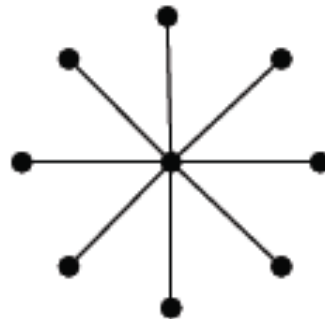
*2-D mesh*



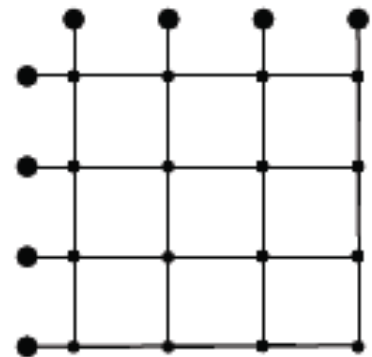
*2-D torus*



*bus*

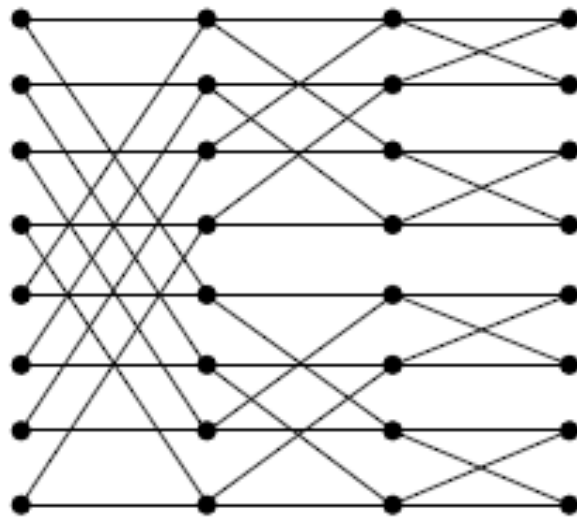


*star*

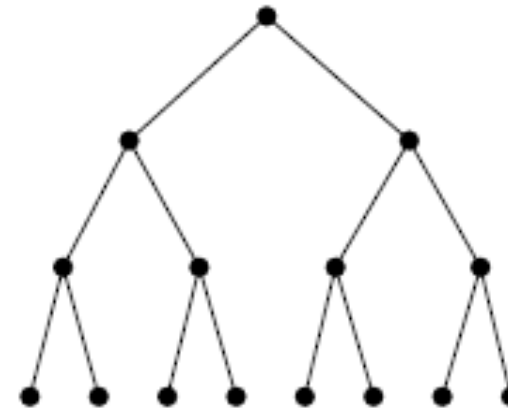


*crossbar*

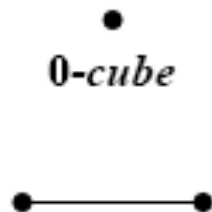
# Topológiák - összefoglaló



*butterfly*

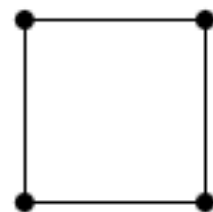


*binary tree*

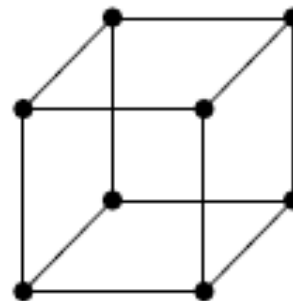


*0-cube*

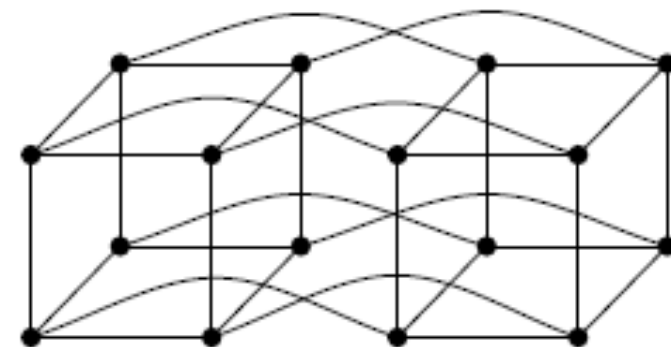
*1-cube*



*2-cube*



*3-cube*



*4-cube*

*hypercubes*