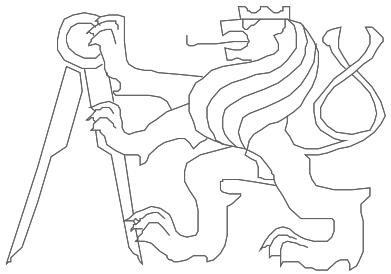
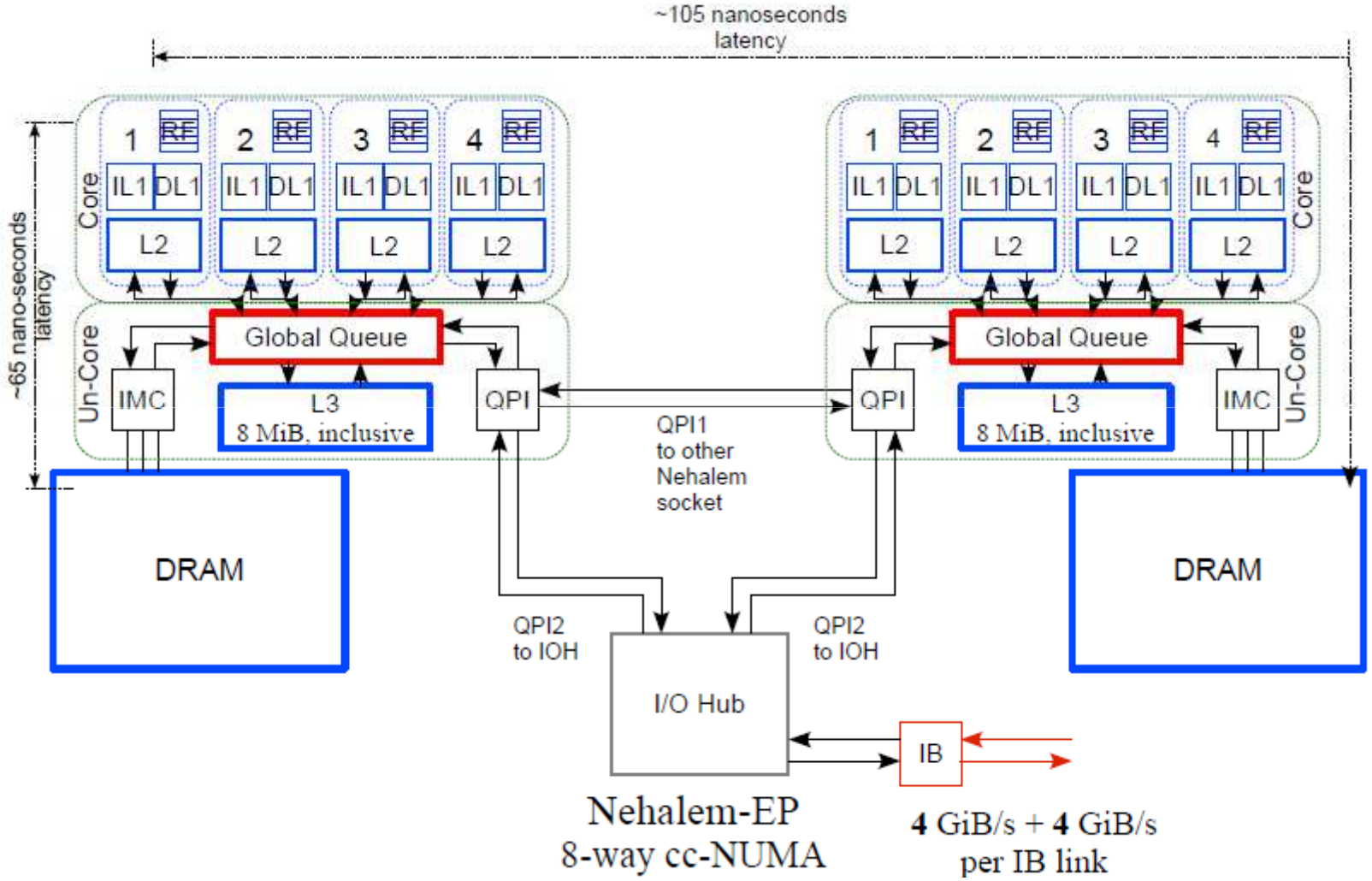
Pokročilé architektury počítačů

Propojovací sítě

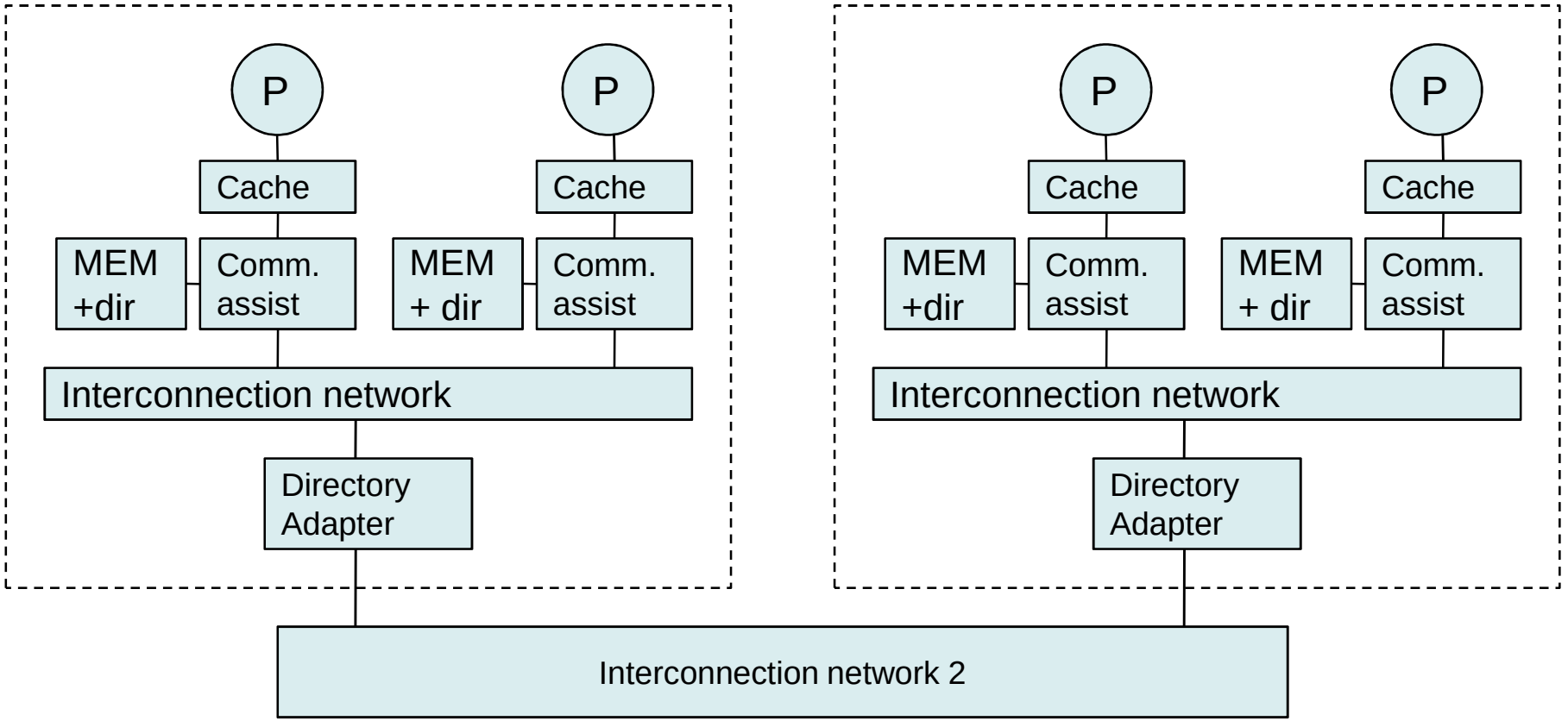


České vysoké učení technické, Fakulta elektrotechnická

Motivace: Propojení dvou procesorů pomocí QPI



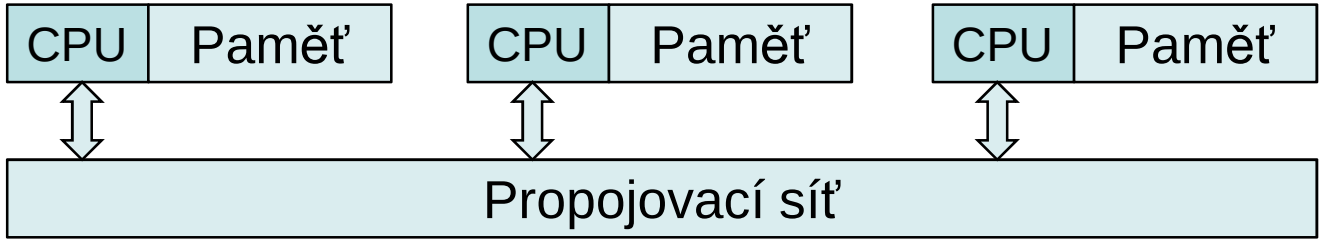
Two-level cache coherent system



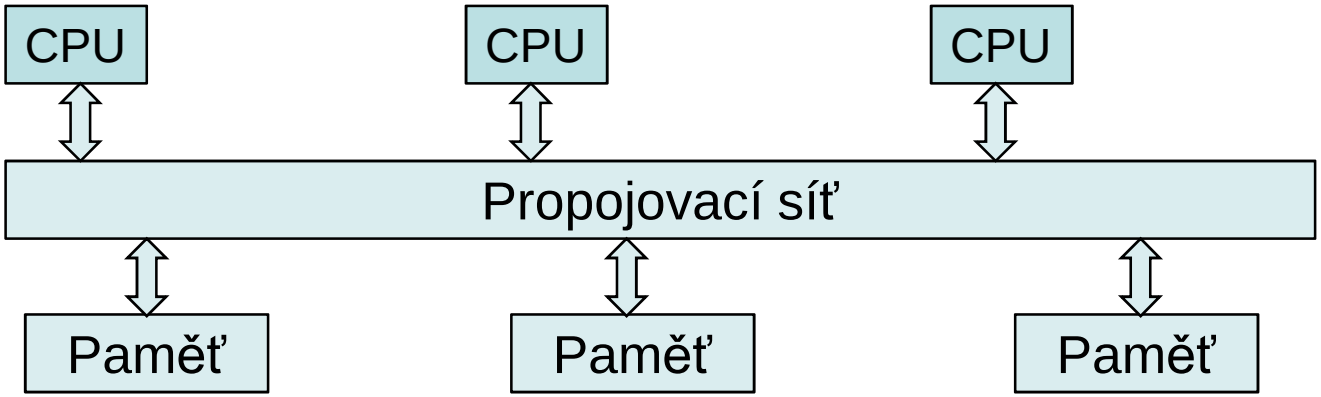
- Directory-directory
- Alternatives: Snooping-Snooping, Snooping-Direcory, Directory-Snooping

Paměťové architektury paralelních procesorů

- **Volně vázaný systém** (Loosely coupled) – paměť je distribuována mezi uzly a každý uzel může přistoupit k paměti v jiném uzlu.
  Pozn.: Pokud by tuto možnost neměl, pak se nejedná o model se sdílenou pamětí.



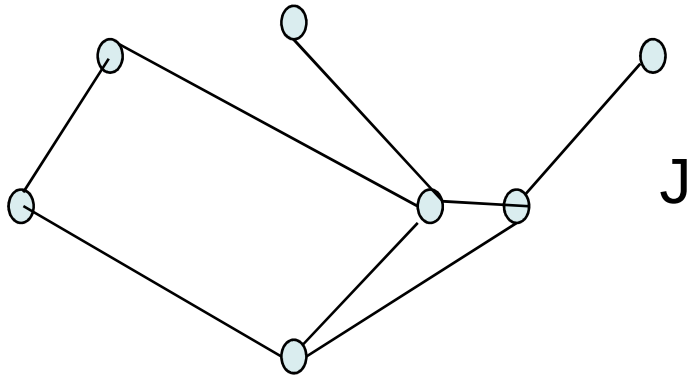
- **Těstně vazaný systém** (Tightly coupled) – paměť je lokalizována centrálně. Procesor může obsahovat vlastní paměť/cache.



SW pohled na propojovací sítě

Síť musí/by měla podporovat:

- One-to-one komunikaci
- One-to-all broadcast a All-to-one redukci
- All-to-all broadcast a All-to-all redukci
- Scatter a Gather



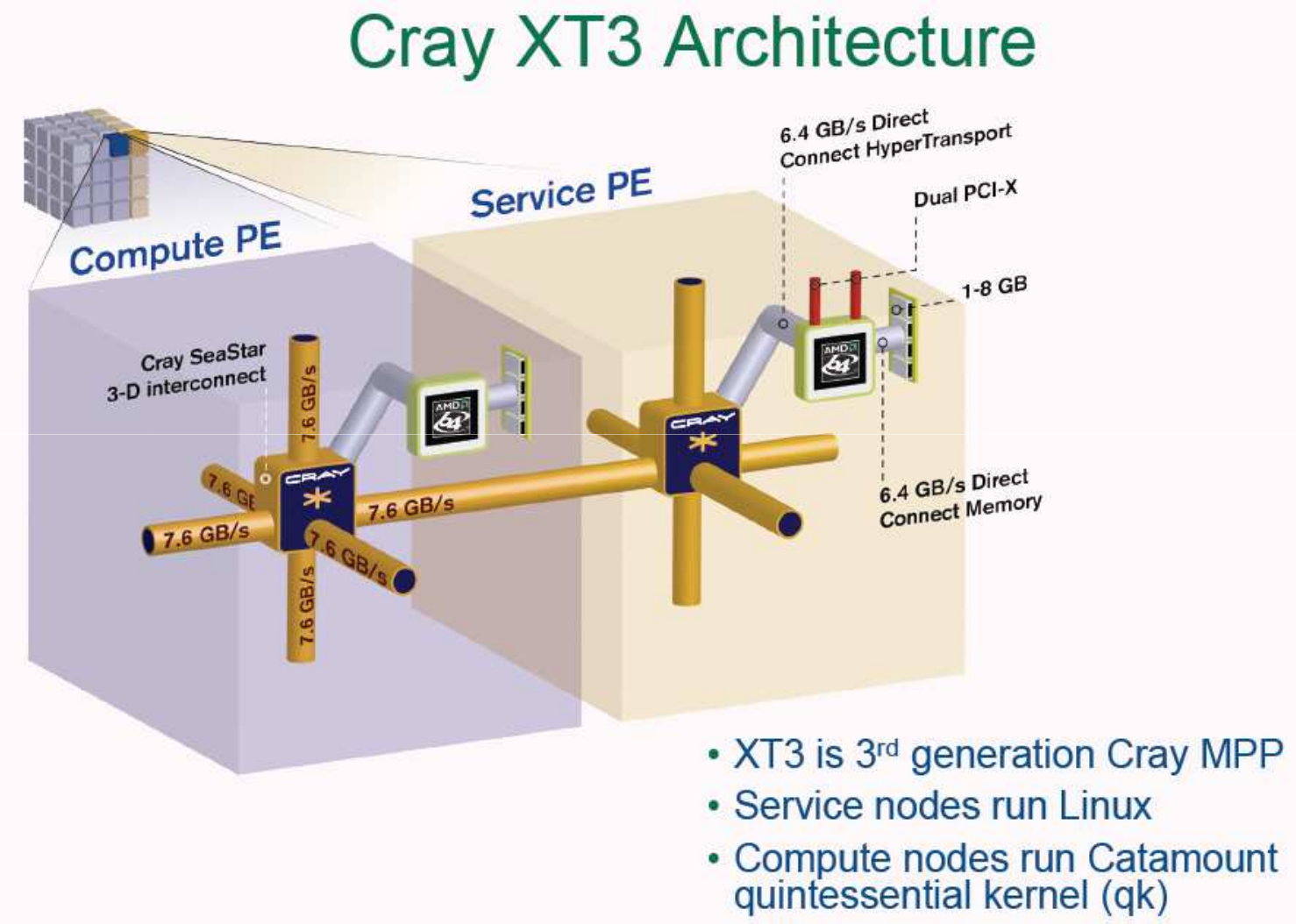
Jak vypadá síť, která optimalizuje tuto komunikaci??

Motivace - Cray XT3

Cray XT3 (20.5 TFlops) v Oak Ridge National Laboratory:

- 56 cabinets, 5212 výpočtových procesorových elementů (PEs), 82 servisních procesorových elementů
- výpočtový uzel – je tvořen 4 CPU (2GB/PE)
- CPU: 64-bitový 2.4GHz AMD Optreon

Motivace - Cray XT3



Topologie

Topologie propojovacích sítí

Statické sítě

- **lineární pole (linear array)**
- **kruh (ring)**
- **Chordálový kruh (Chordal ring)**
- **binary tree**
- **fat tree**
- **2D, 3D mesh**
- **2D, 3D torus**
- **hypercube**
- **Cube Connected Cycles (CCC)**

Dynamické sítě

- **Bus network**
- **jednostupňové sítě (křížové přepínače - crossbars)**
- **vícestupňové sítě (omega, Banyan, Cantor, Clos,...)**

Topologie propojovacích sítí

Přímé sítě

Každý uzel má svůj přepínač a obráceně

Nepřímé sítě

Některé přepínače nemají přiřazen uzel, pouze směrují provoz k dalším přepínačům

Statické sítě

Parametry

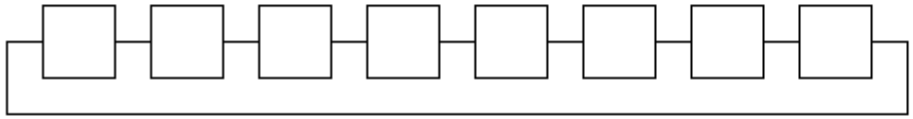
- **velikost sítě N (Network size): počet uzlů v síti,**
- **stupeň uzlu d (Node degree): počet hran vstupujících nebo vystupujících z uzlu,**
- **bisekční šířka B (Bisection width): minimální počet hran, které musíme přerušit při rozdělení sítě na dvě stejné poloviny,**
- **průměr sítě D (Network diameter): počet hran maximální nejkratší cesty mezi dvěma libovolnými uzly sítě - ukazuje nejdelší komunikaci,**
- **cena C (Cost): počet komunikačních linek (hran)**

Linear array

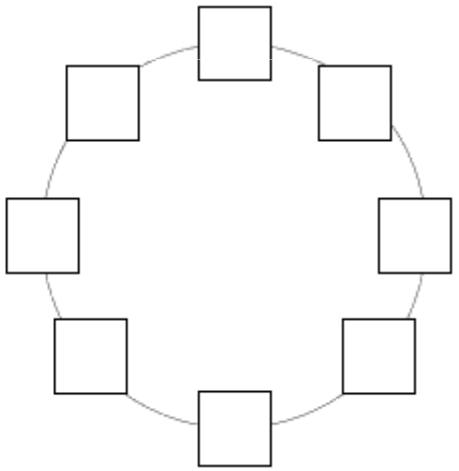
Statické sítě

Linear array

d: 1 (pre koncové uzly), 2 (pre ostatné)
D: N-1
B: 1
C: N-1



Ring



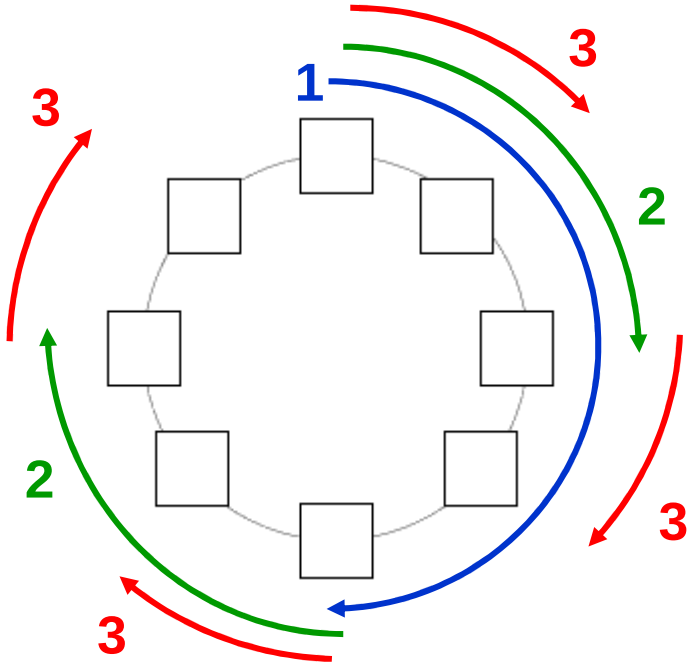
d: 2
D: N/2
B: 2
C: N

Ring – příklad komunikace

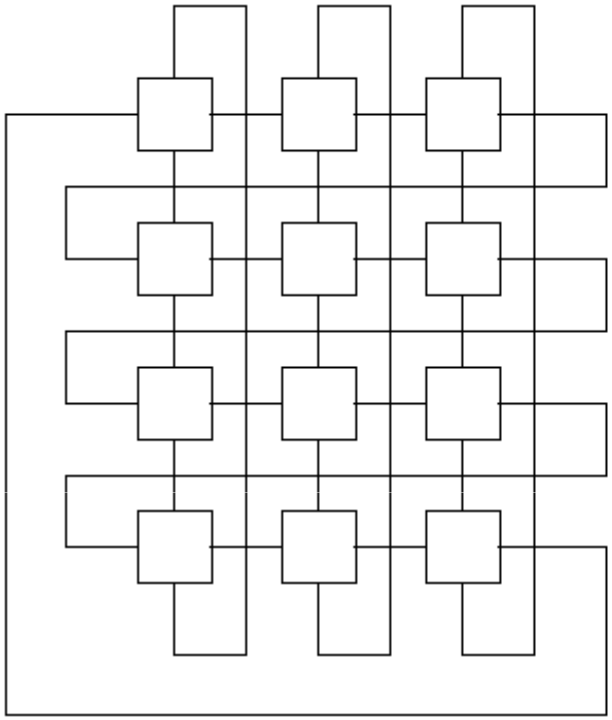
One-to-all broadcast a All-to-one redukce

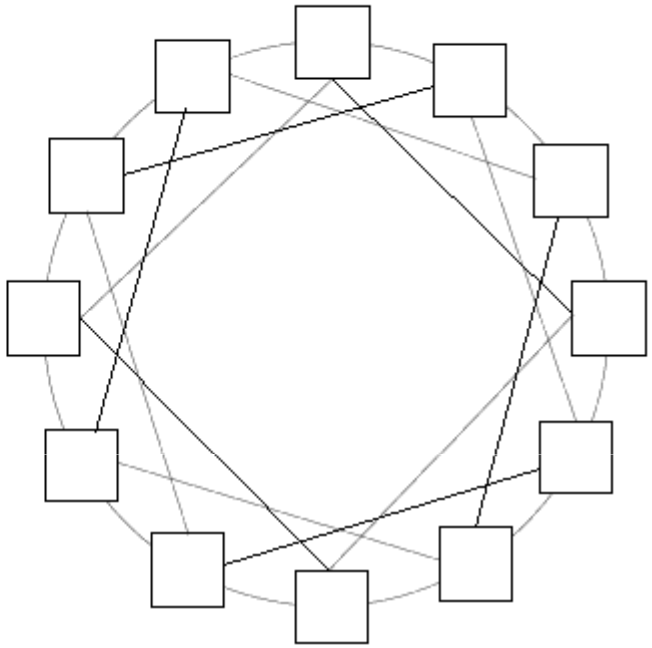
- Nejjednodušší je poslat N-1 zpráv ze zdroje do N-1 cílů (procesorů, uzlů). Není moc efektivní…
- Použij algoritmus rekurzivní zdvojení (recursive doubling) – zdroj pošle zprávu do vybraného uzlu. Tím se problém rozdělí na dva.
- Nebo lineárně sousedovi

- Redukce – obráceně. Data musí však zpracovat…



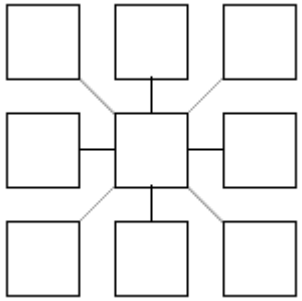
Statické sítě





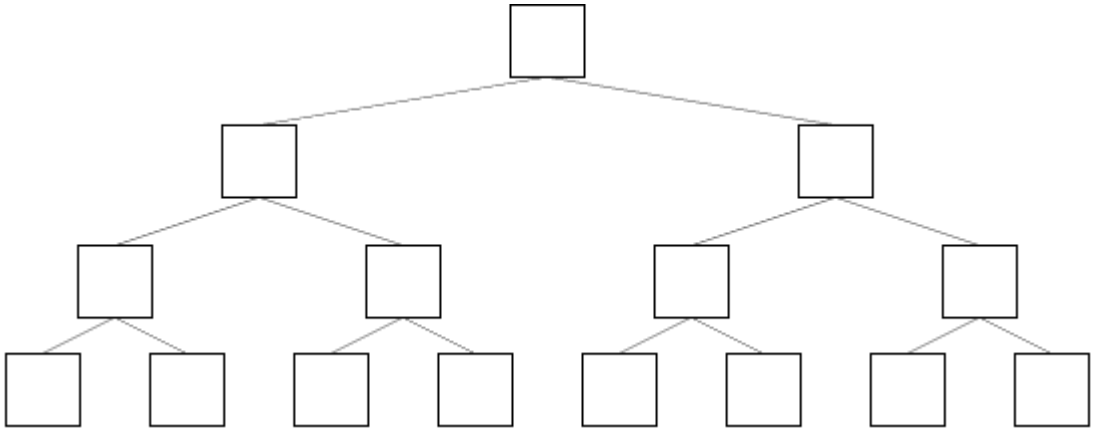
Chordal ring

Statické sítě



Star

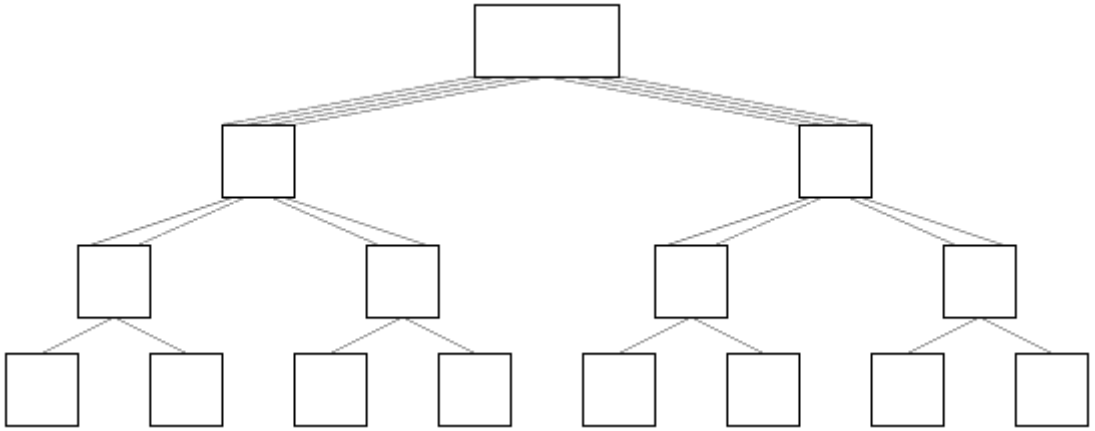
d: 1 , N-1
D: 2
B: 1
C: N-1



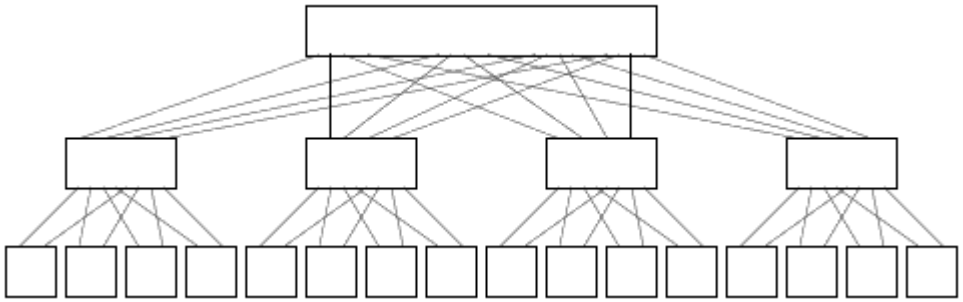
Binary tree

d: 1 (koncové uzly), 2 (kořenový), 3 (ostatní)
D: 2log((N+1)/2)
B: 1
C: N-1

Statické sítě



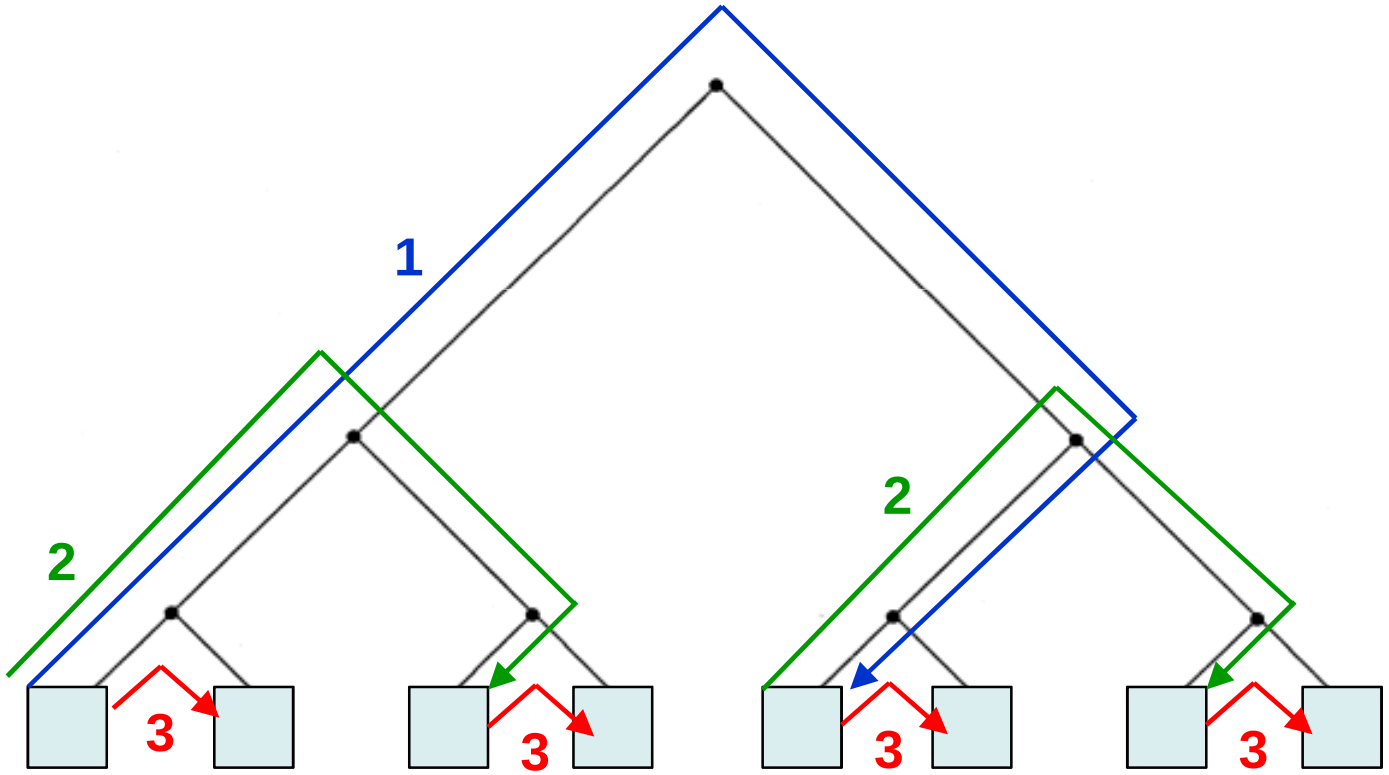
Binary fat tree



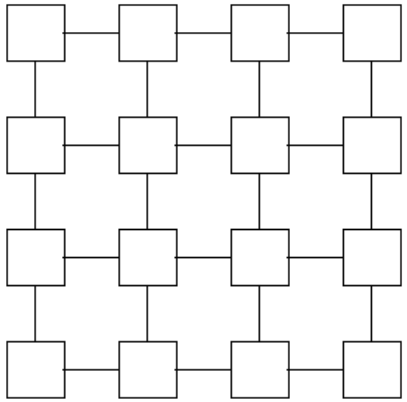
4-ary fat tree

Nepřímý Binární strom – příklad komunikace

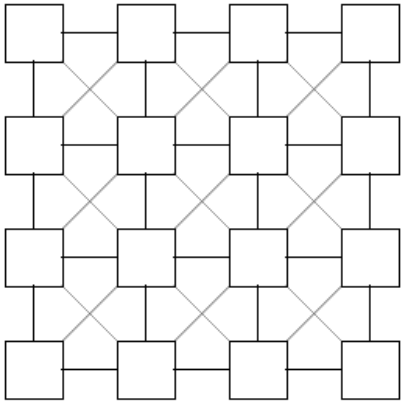
- V nepřímém stromu jsou výpočetní uzly v až listech…
- One-to-all broadcast a All-to-one redukce



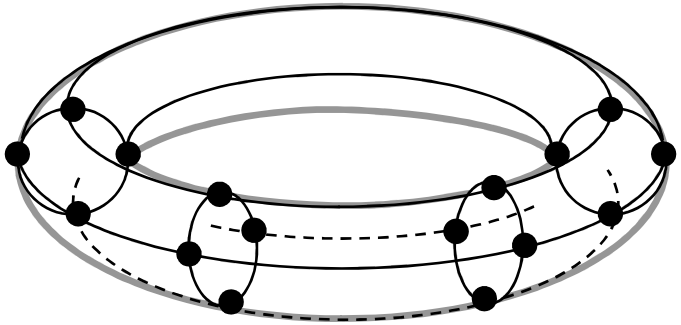
Statické sítě

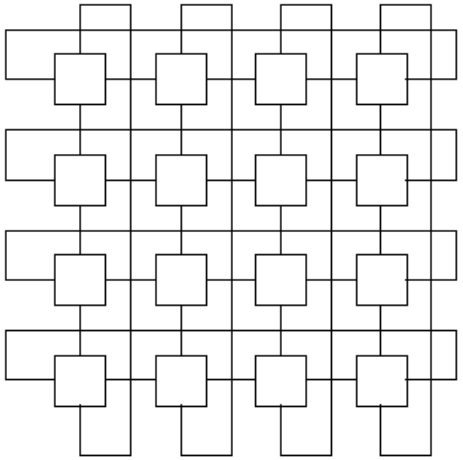


4-connected 2D mesh



8-connected 2D mesh



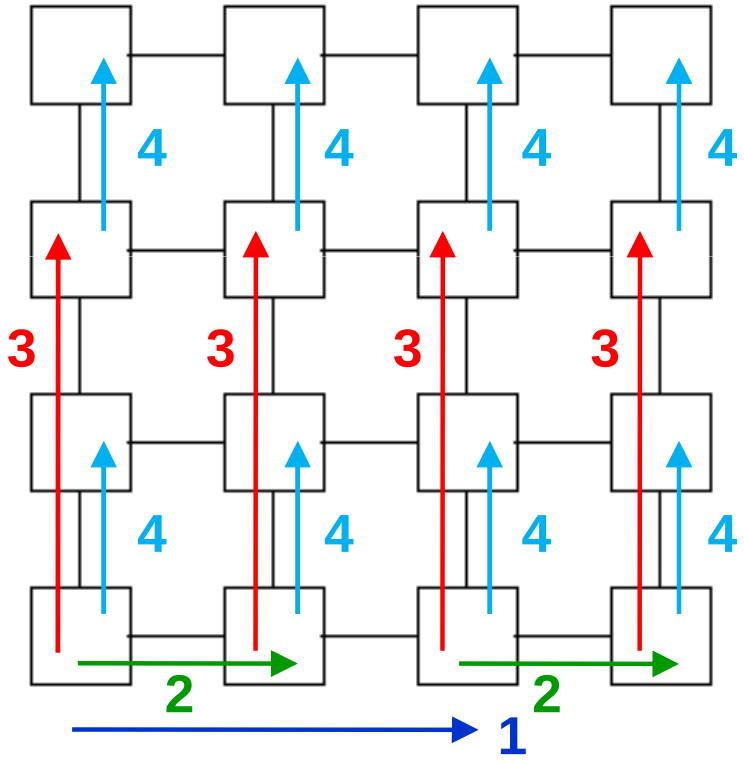


d: 4
D: $N^{1/2}$
B: $2N^{1/2}$
C: 2N

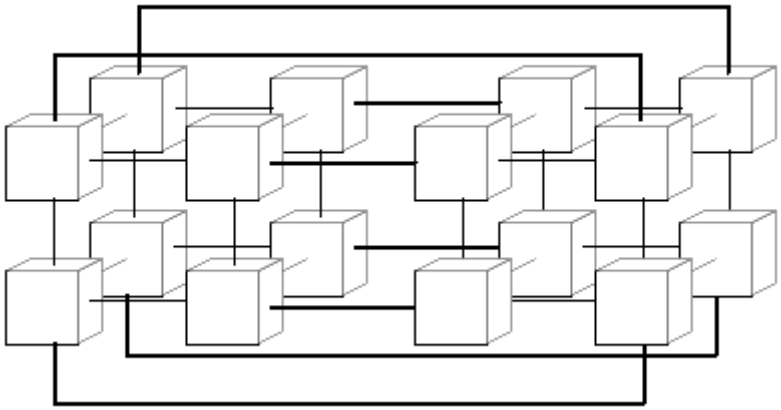
2D Torus (anuloid)

Mesh – příklad komunikace

- One-to-all broadcast a All-to-one redukce

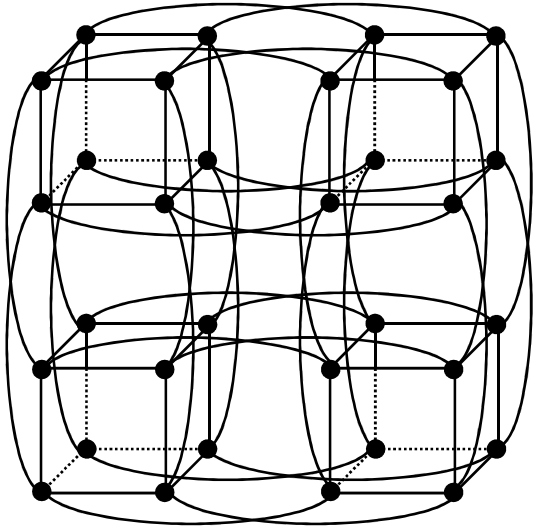


Statické sítě

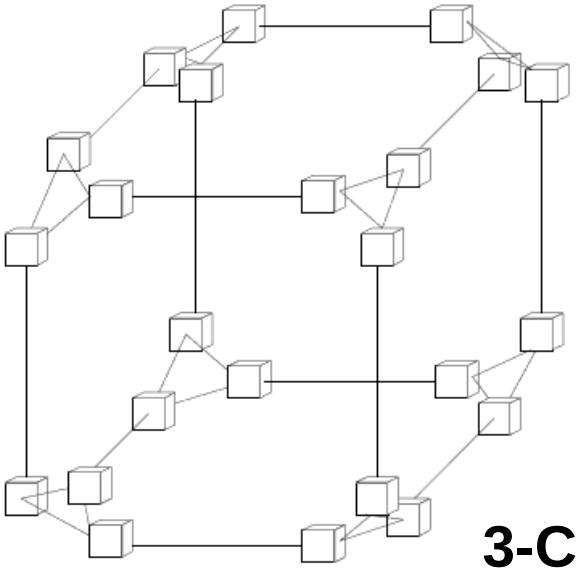


N=4

Hypercube



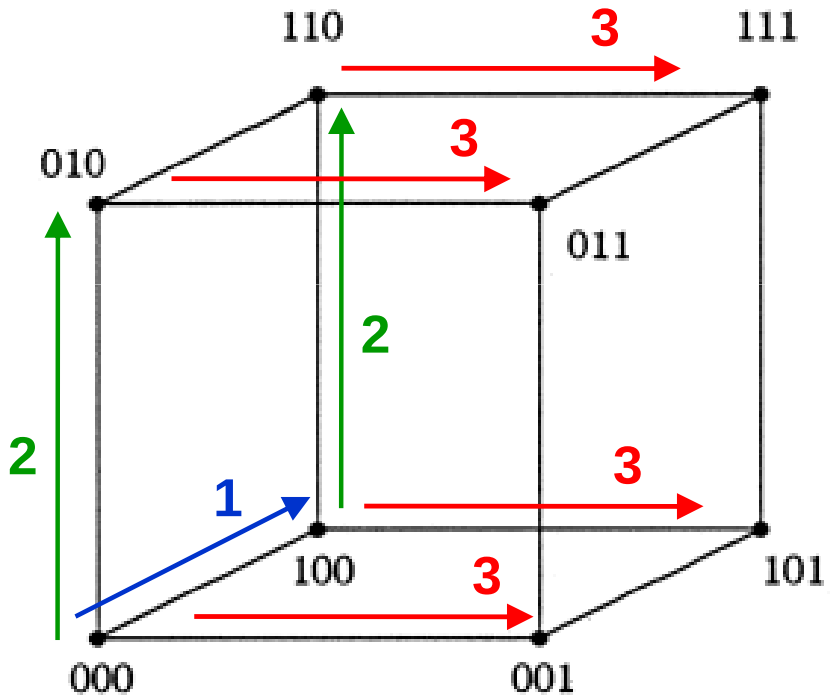
N=5



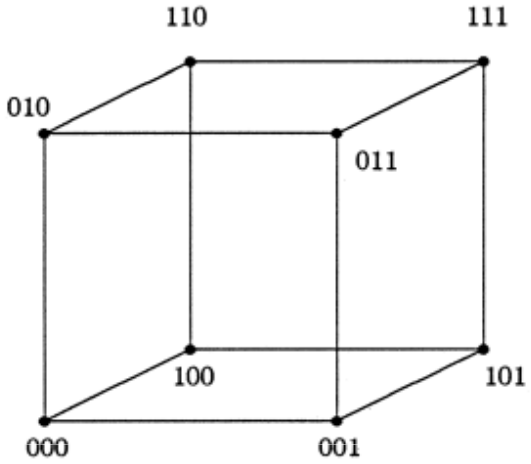
3-Cube Connected Cycles

Hypercube – příklad komunikace

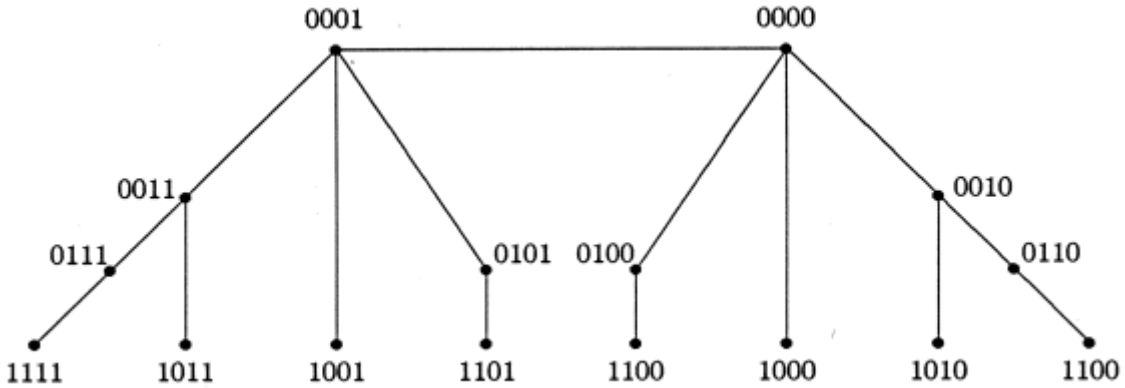
- One-to-all broadcast a All-to-one redukce



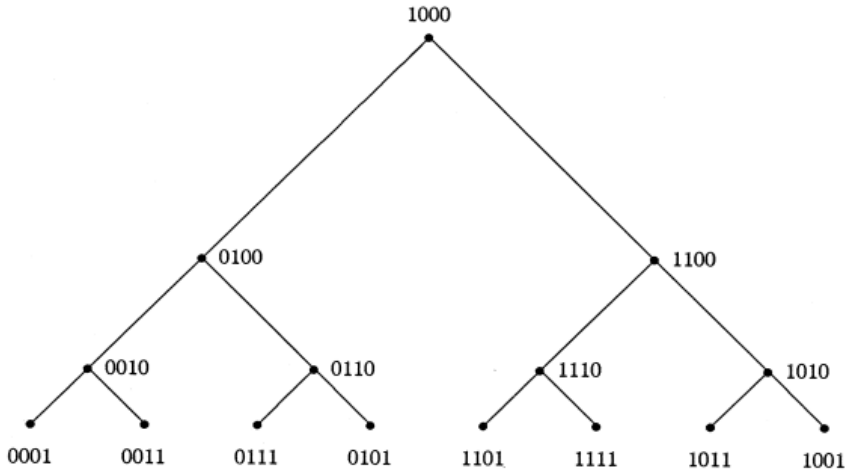
Hypercube – vztah k jiným sítím



Stromová struktura 4-dimenzionální krychle:

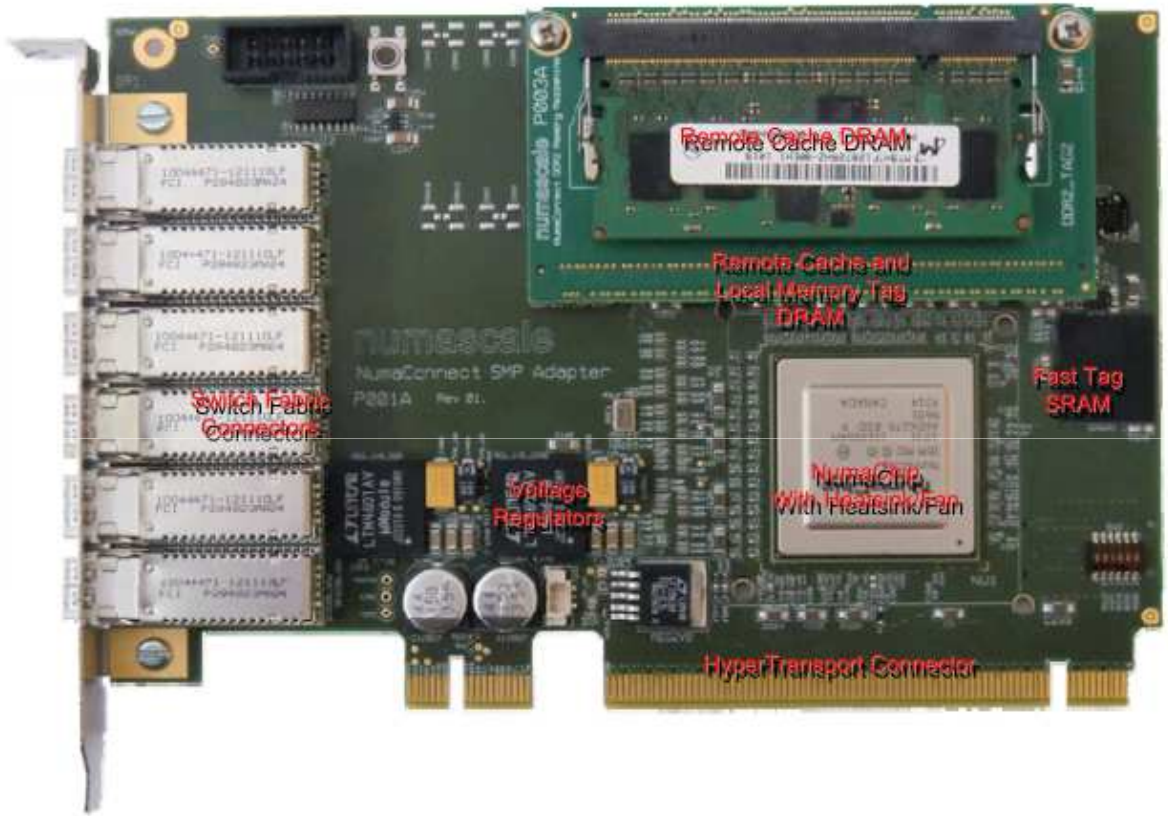


Pro n>=3 je možné mapovat n-úrovňový binární strom do podgrafu získaného z hyperkrychle při odstranění některého z uzlů. Hrana pak reprezentuje buď hranu hypekrychle nebo cestu délky 2.



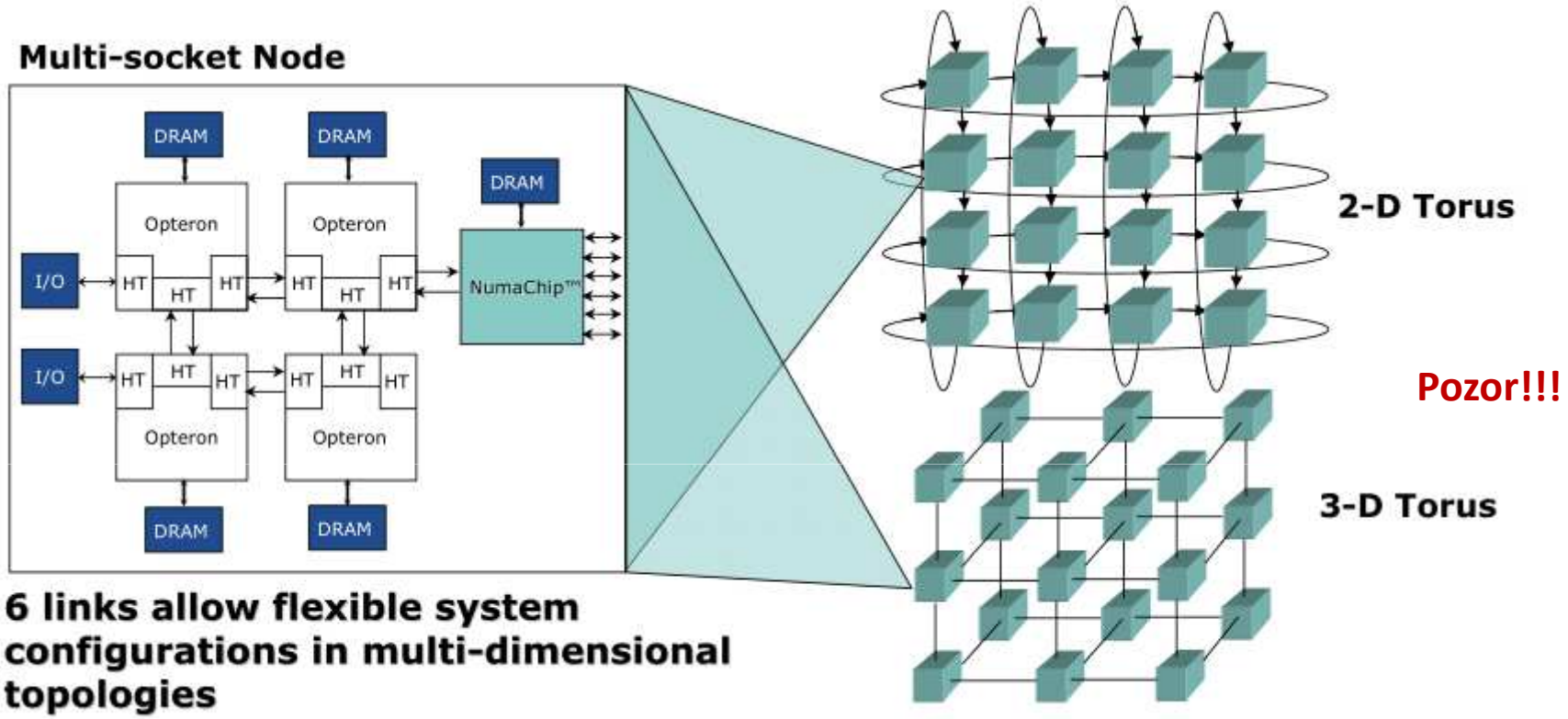
Diskuze – Tím to však nekončí...

numascale



- Scalable directory based cache coherence in hardware
- Support for 4096 Nodes
- 4GB Cache 8 GB Tag (supports 240 GB Local Node RAM)

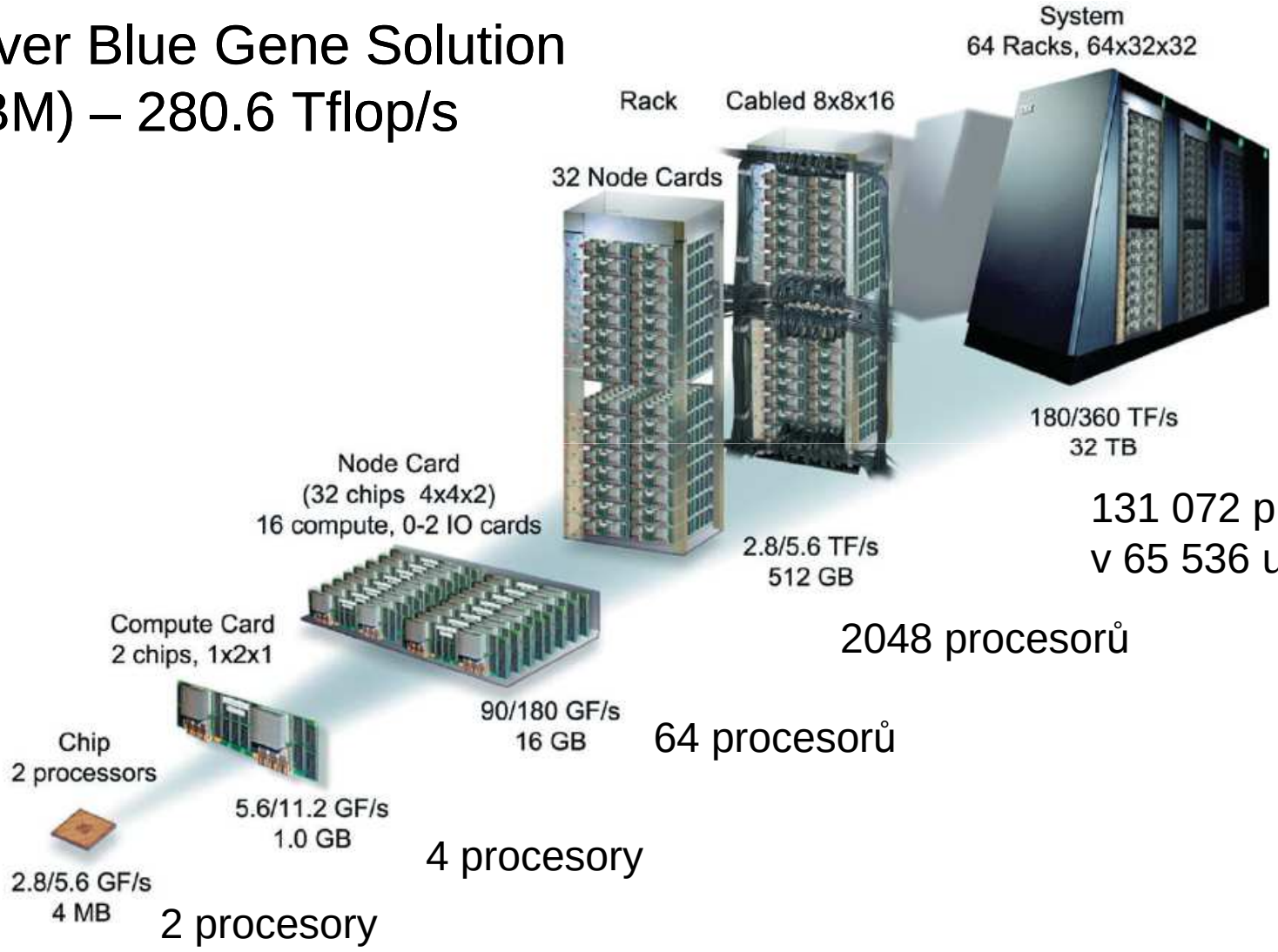
Diskuze – Tím to však nekončí...



- Use any Programming Model Available for the Node on the whole System (OpenMP, MPI, Threads, ...)
- NO Application Changes Required!

Motivace - Blue Gene Solution

eServer Blue Gene Solution
(IBM) – 280.6 Tflop/s



131 072 procesorů
v 65 536 uzlech

2048 procesorů

64 procesorů

4 procesory

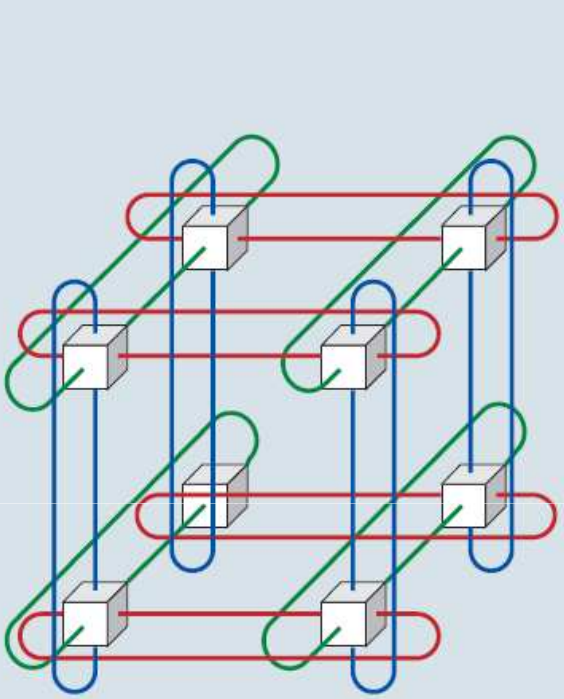
2 procesory

Motivace - Blue Gene Solution

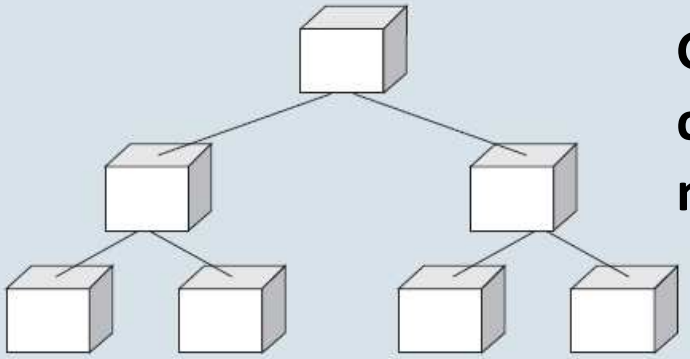
Využívá 5 sítí pro propojení uzlů

- **3D torus pro point-to-point komunikaci mezi uzly (175 MBps v každém směru),**
- **globální hromadní síť pro hromadné operace (Global collective network) – 350 MBps, 1.5μs latence (data z jednoho uzlů mohou být rozeslána všem ostatním uzlům – broadcast, nebo jenom některým),**
- **global barrier and interrupt network,**
- **control network (system boot, debug, monitoring stavu teploty, ventilátorů,...)**
- **gigabit Ethernet network pro řízení a I/O operace**

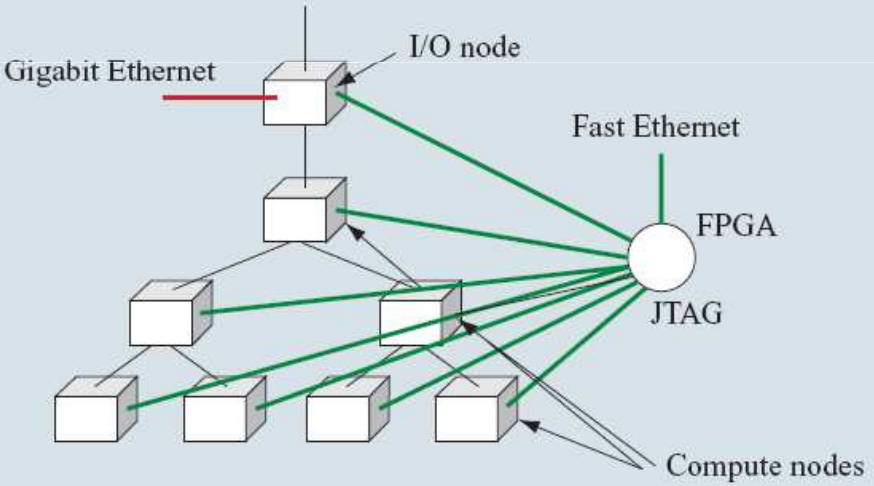
Motivace - Blue Gene Solution



3D Torus



Global collective network



Blue Gene/L control system network and Gigabit Ethernet networks

Topologie

Topologie propojovacích sítí

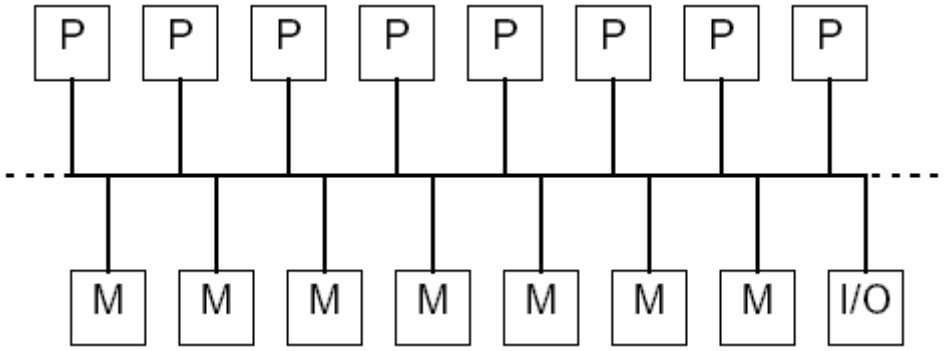
Statické sítě

- lineární pole (linear array)
- kruh (ring)
- Chordálový kruh (Chordal ring)
- binary tree
- fat tree
- 2D, 3D mesh
- 2D, 3D torus
- hypercube
- Cube Connected Cycles (CCC)

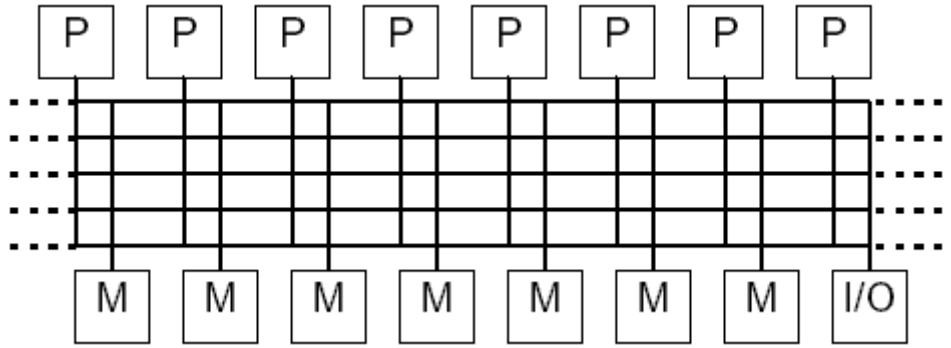
Dynamické sítě

- Bus network
- jednostupňové sítě (křížové přepínače - crossbars)
- vícestupňové sítě (omega, Banyan, Cantor, Clos,...)

Dynamické sítě

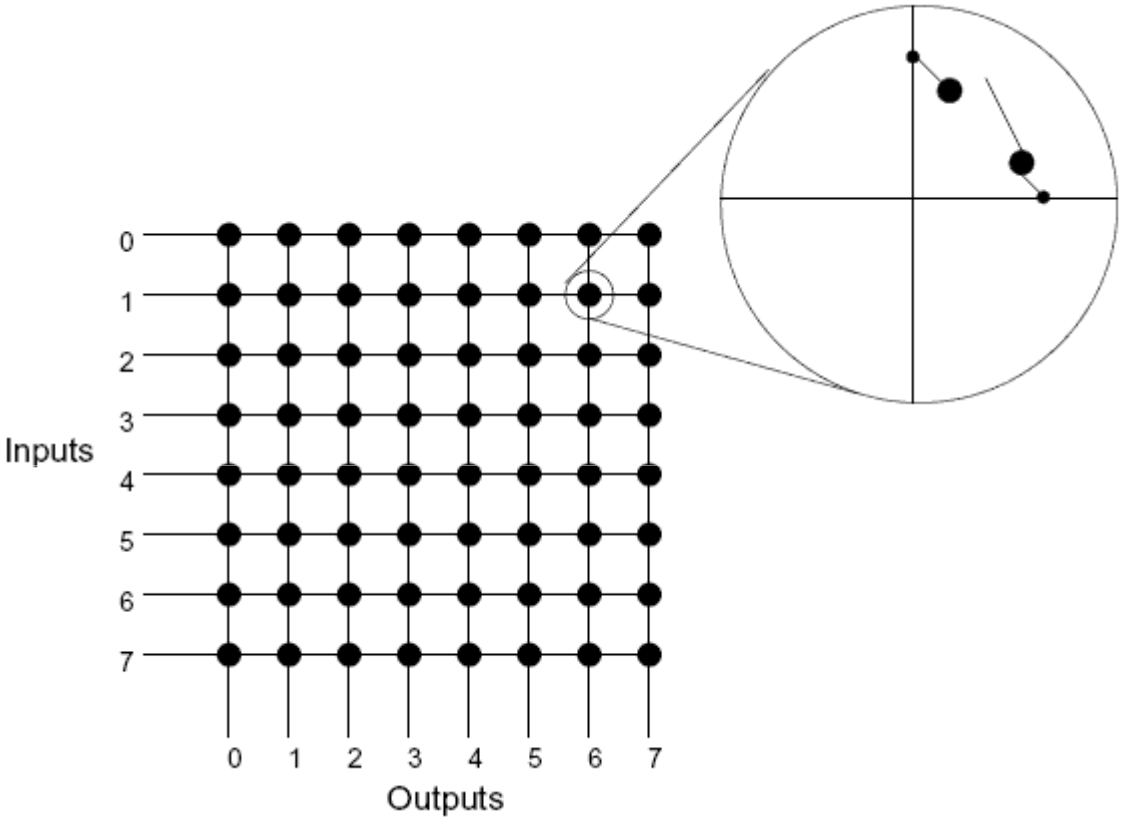


Bus-based multiprocessor system



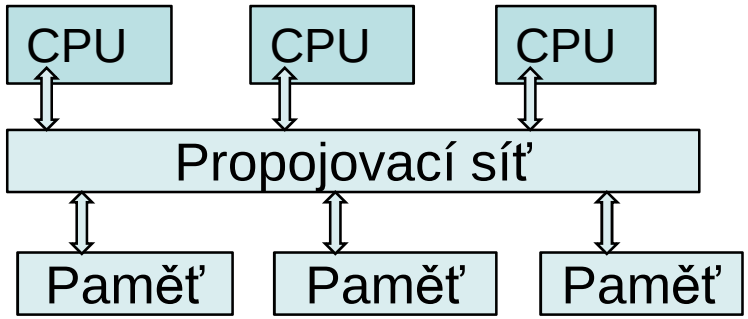
Multiple bus architecture

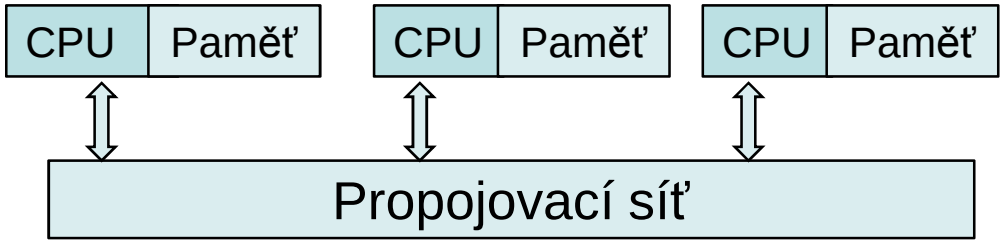
Dynamické sítě – Single stage network

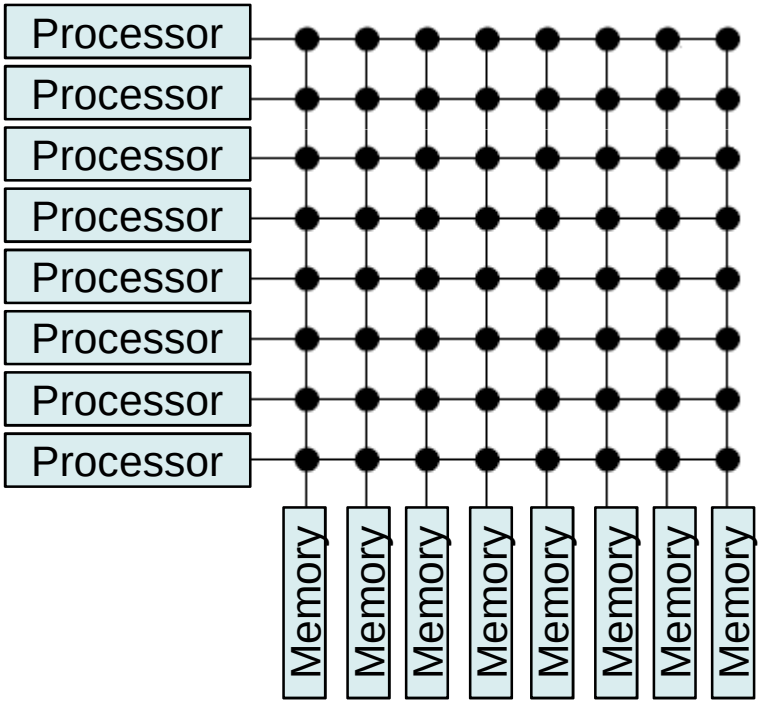


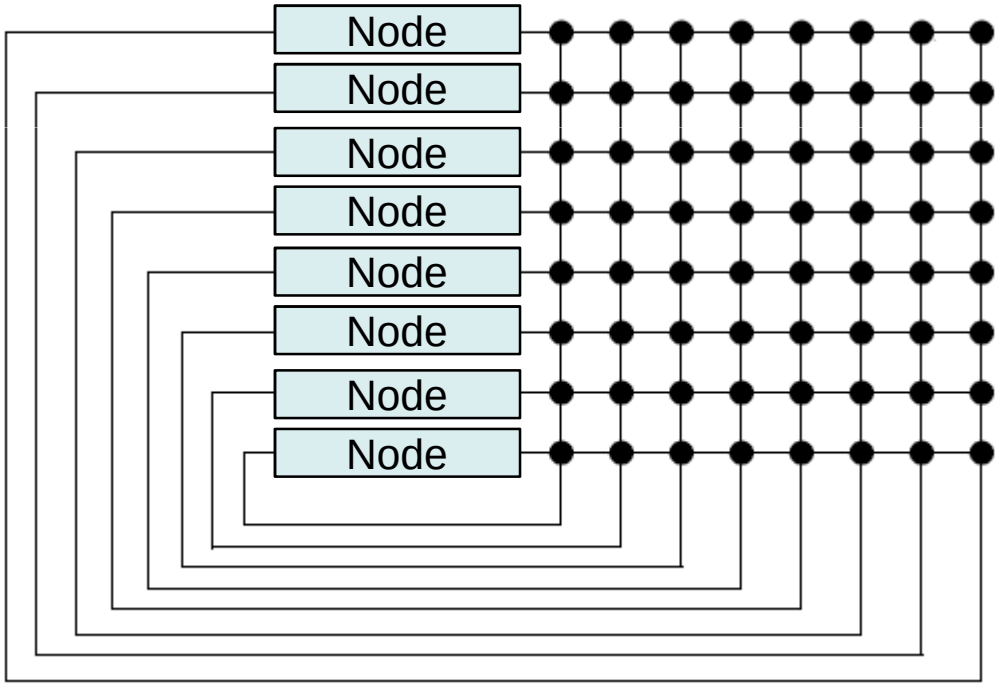
Crossbar – Křížový přepínač – Cena N^2

Dynamické sítě – Single stage network

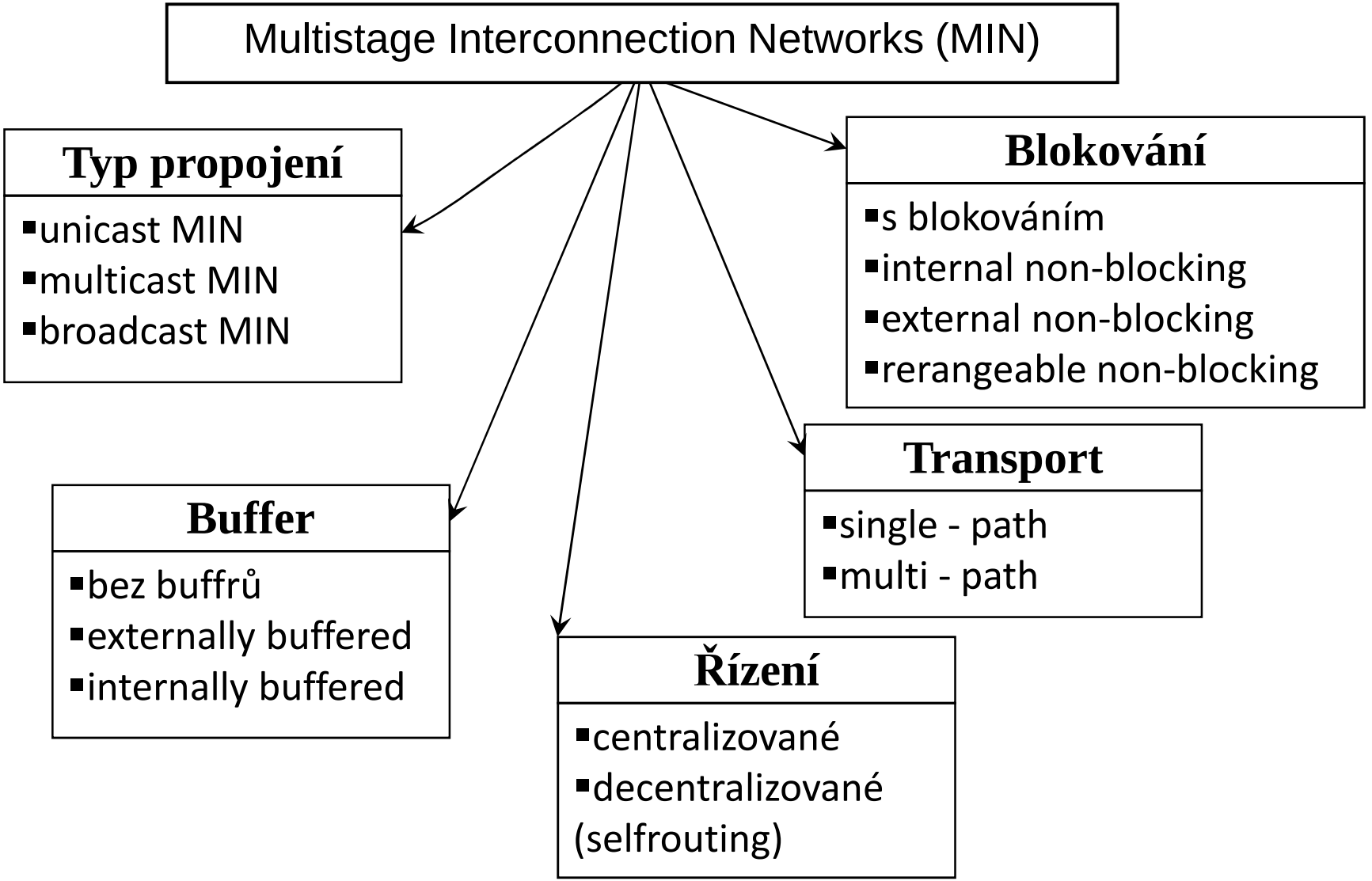








Dynamické sítě – Multistagestage interconnection networks



Dynamické sítě – Multistagestage interconnection networks

- Unicast MIN (one-to-one, point-to-point) - je síť, která propojuje v téže době právě jeden vstup s právě jedním výstupem.
- Multicast MIN (multipoint) - propojí v téže době jeden vstup sítě s více výstupy sítě
- Broadcast MIN - propojuje jeden vstup sítě se všemi výstupy sítě v téže době. Broadcast MIN je speciálním případem Multicast MIN.

- Jednocestní MIN (Single-Path) -všechny pakety v rámci jednoho virtuálního spojení používají stejnou cestu.
- Vícecestní MIN (Multi-Path) -pakety v rámci jednoho virtuálního spojení používají několik cest. Pakety jsou náhodně distribuovány. Vnitřní provoz ve spojovací síti se stane nezávislým na vnějším provozu (tj na poměru služeb, které jsou propojovány v síti). Taková síť vyžaduje opatření pro řazení buněk u východu ze sítě.

Dynamické sítě – Multistagestage interconnection networks

- MIN s centralizovaným řízením -řízení je prováděno centrálním procesorem. Procesor určuje výběr cesty v síti, pokud jsou dány požadavky na spojení.
- MIN s decentralizovaným řízením -řízení je distribuováno na spojovací elementy (samosměrování - self routing). V tomto případě potřebujeme po vstupu do sítě pro každý paket přídavnou informaci, tag, na základě které se v každém elementu určuje kombinace vstupu a výstupu.

Dynamické sítě – Multistagestage interconnection networks

- Sítě s blokováním - jsou sítě kdy se konkrétní vstup nemůže propojit s konkrétním výstupem, přesto že výstup není obsazen jiným spojením. Blokování je spojeno se ztrátou informace, nebo její zpožděním. Pro zajištění kvality spojování je žádoucí blokování vyloučit, nebo jej minimalizovat na specifikovanou úroveň.
- Sítě bez vnitřního blokování (Internal non-blocking) - jsou sítě, které zajišťují propojení libovolného vstupu s libovolným výstupem bez zrušení, nebo rekonfigurace jiné vnitřní cesty v síti. Může však nastat případ, že i když síť je bez vnitřního blokování, dva, nebo více vstupů může chtít použít stejný výstup v téže době - vzniká konflikt.
- Sítě bez vnějšího blokování (External non-blocking) - jsou sítě schopné propojení libovolného vstupu s libovolným výstupem v každém případě. V síti je nutná v případě konfliktu funkce uložení informace, tj vyrovnávací paměť.
- Rekonfigurovatelné (preuspořádatelné) sítě bez blokování (Rearrangeable non-blocking) - jsou zvláštním případem blokovacích sítí - ovšem dokáží vždy realizovat spojení z libovolného vstupu na libovolný výstup. V případě konfliktu dochází k restrukturalizaci (přeuspořádání) stávajících cest a k vytvoření nové konfigurace spojení.

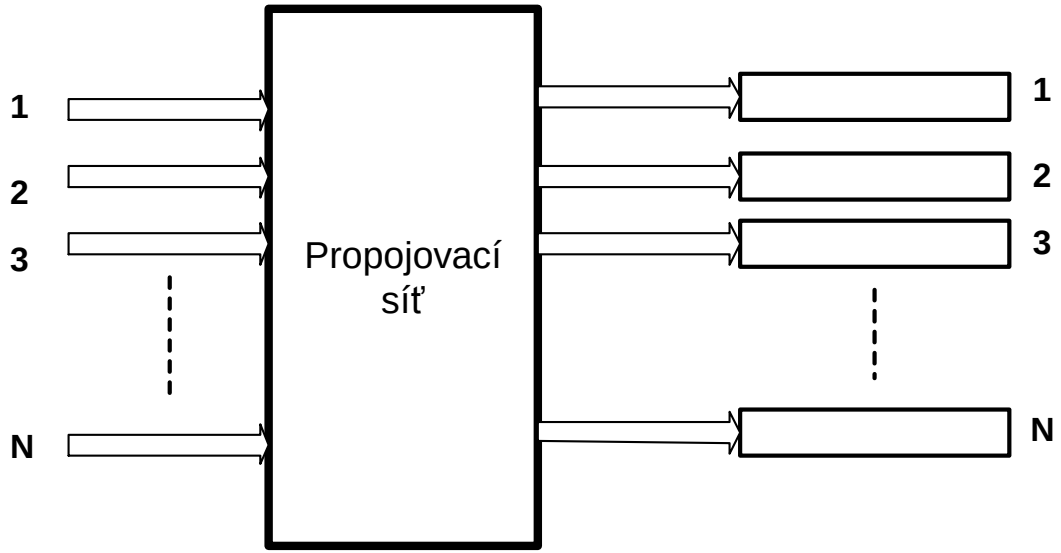
Dynamické sítě – Multistagestage interconnection networks

- MIN s vnější vyrovnávací pamětí (externally Buffered) - vyrovnávací paměti jsou umístěny na vstupech sítě, na výstupech sítě, nebo kombinovaně.
- MIN s vnitřní vyrovnávací pamětí (Internally Buffered) - vyrovnávací paměti jsou v jednotlivých spojovacích elementech, čili uvnitř sítě. I v spojovacím elementu může být vyrovnávací paměť na vstupu (Input Buffering), na výstupu (Output Buffering), nebo uprostřed elementu (Central Buffering)

Hlavní způsoby řazení (ukládání) paketů:

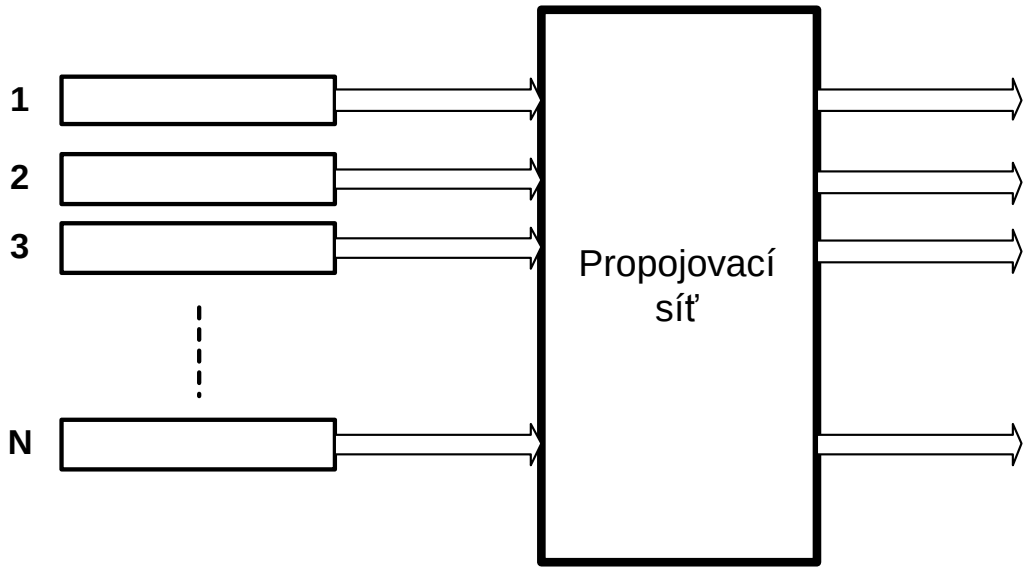
- výstupní řazení (OQ)
- vstupní řazení (IQ)
- kombinované vstupně-výstupní řazení (CIOQ)
- centralizované sdílené řazení (CSQ)
- virtuální výstupní řazení (VOQ)

Výstupní řazení paketů: (IQ - output queuing)



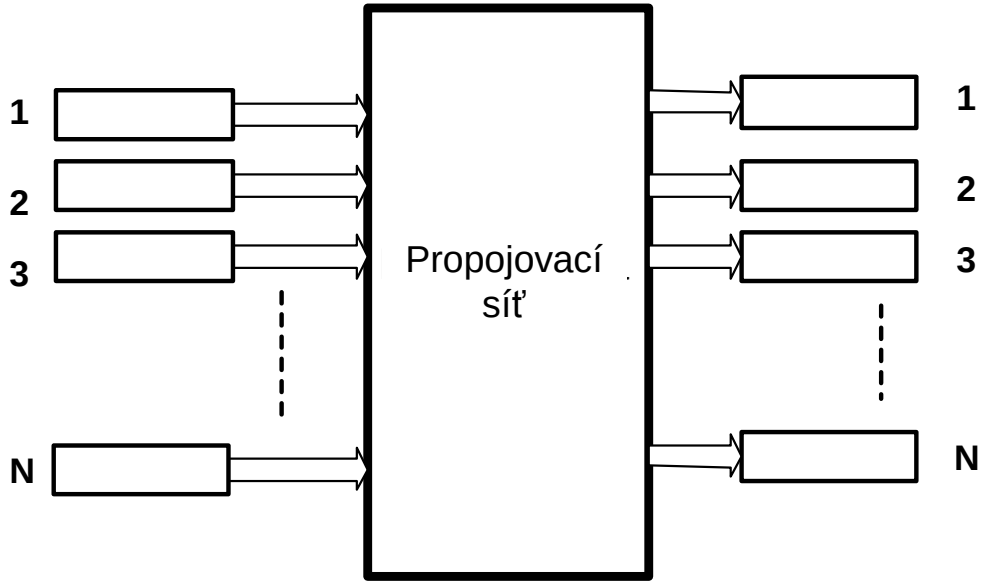
Když paket dorazí na vstupní port, je okamžitě (po průchodu sítí) uložen do bufferu, který je na příslušném výstupním portu. Protože pakety určené pro stejný výstupní port mohou přijít současně z mnoha vstupních portů, výstupní buffer potřebuje radit pakety mnohem větší rychlostí než vstupní port. V nejhorším možném případě to může být až N krát rychleji (kde N je počet vstupních portů), a to tehdy pokud pakety ze všech vstupních portů jsou určeny pro jeden určitý výstupní port. Rychlost, kterou můžeme přistupovat do výstupního bufferu je však limitována.

Vstupní řazení paketů: (IQ - output queuing)



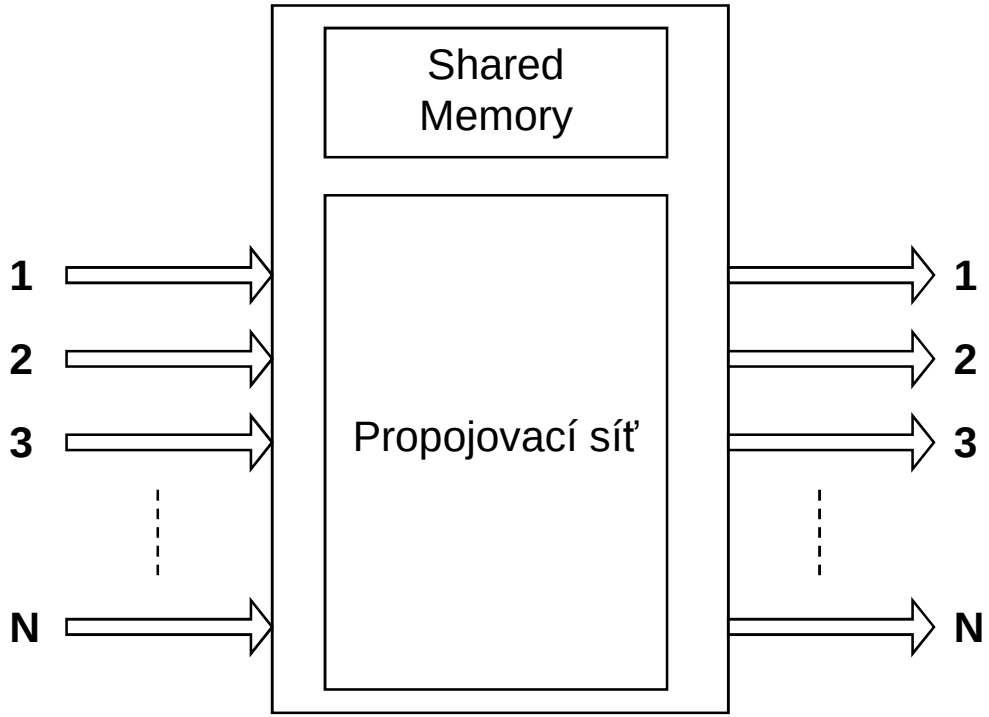
Vstupní řazení nemá limity jaké má například výstupní řazení nebo centralizované sdílené řazení. V této architektuře má každý vstupní port FIFO zásobník buněk, a pouze první buňka ve frontě je způsobilá pro přenos v průběhu daného časového úseku. Nevýhodou FIFO řazení je, že když buňka na čele fronty se zablokuje, zablokují se všechny buňky za ní a zabrání se tak jejich přenosu, dokonce i když je výstupní port volný. Toto se nazývá Head-of-line blokování. Matematickou analýzou a počítačovou simulací bylo ukázáno, že HOL blokování omezuje propustnost každého vstupního portu na maximálně 58.6 procent a to při náhodné hustotě provozu, a tato hodnota je ještě mnohem nižší při velmi hustém provozu.

Kombinované vstupně-výstupní řazení paketů: (COQ - combined input-output queuing)



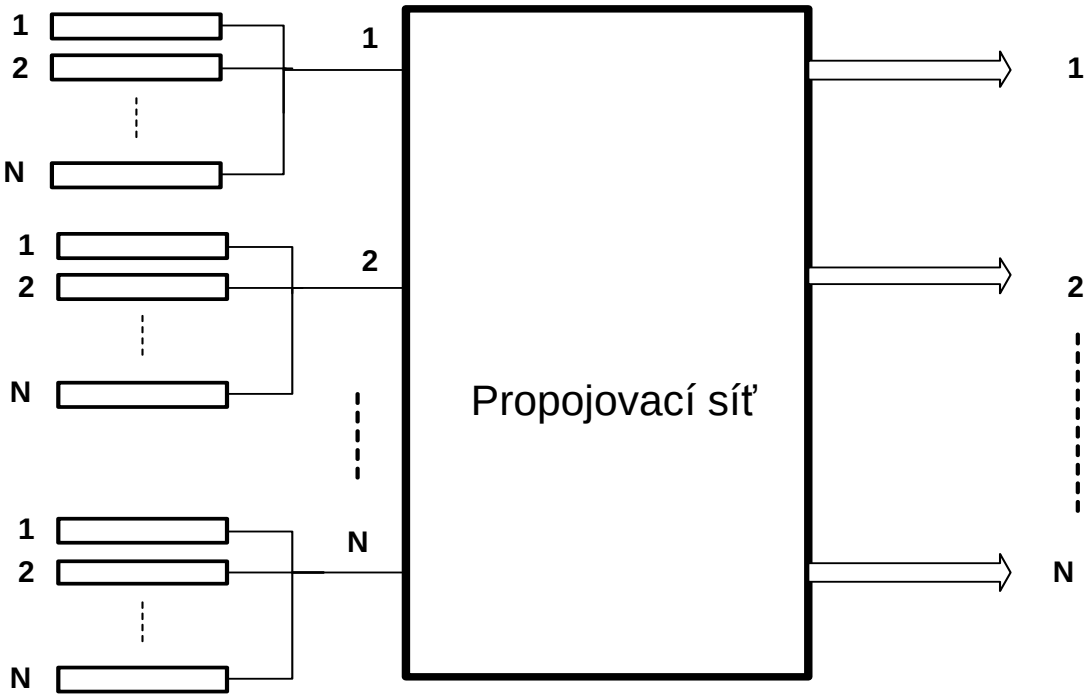
Toto schéma řazení je kombinací vstupního a výstupního řazení. Je to dobrý kompromis mezi výkonem a možností rozšiřování OQ a IQ přepínačů. Pro vstupně řazeny přepínače, nanejvýš jeden paket může být doručen na výstupní port v jednom časovém úseku. Pro výstup frontované přepínače, až N paketů může být doručených na výstupní port za jeden časový úsek. Použitím CIOQ, namísto těchto dvou extrémních možností, můžeme si zvolit kompromis mezi nimi.

Centralizované sdílené řazení paketů: (Centralized shared queuing)



Máme buffer sdílený všemi výstupními porty přepínače, na který se můžeme dívat jako na sdílenou paměťovou jednotku, a který má N souběžných přístupů k zápisu pro N vstupních portů, a N souběžných přístupů na čtení pro N výstupních portů. Protože pakety určené pro stejný výstupní port mohou přicházet současně z více vstupních portů, výstupní port musí být schopen načítat mnohem rychleji než vstupní port dokáže zapisovat údaje.

Virtuální výstupní řazení paketů: (VO - virtual output queuing)



Toto schéma řazení zvládá Head-of-line blokování a zároveň si udržuje svou výhodu rozšiřitelnosti. Při této metodě každý vstupní port udržuje izolovanou frontu pro každý výstupní port. Klíčovým faktorem pro dosažení vysokého výkonu použitím VOQ přepínačů je plánovací algoritmus, který je zodpovědný za výběr paketů, které by měly být přeneseny v každé časové jednotce ze vstupních porotv na výstupní. Několik takových algoritmů již bylo navrženo, jako např. PIM (parallel iterativní matching - paralelní iterační párování), Islip a RPA. Bylo ukázáno, že s méně než čtyřmi opakováními výše uvedeného řídícího algoritmu PIM, propustnost přepínače přesáhne 99 procent.

Základní typy vícestupňových sítí

Unicast sítě s blokováním jednocestné (Single-Path)

- Unicast spojovací sítě provádějí **one-to-one** spojení. Jednocestné spojovací sítě (Single-Path) mají právě jen jednu cestu a tím jen jednu možnost spojení mezi libovolným vstupem a libovolným výstupem.
  - Baseline síť
  - Banyan síť
  - Delta síť
  - Omega síť

Unicast sítě s blokováním vícecestné (Multi-Path)

- Vícecestné sítě zajišťují alternativní cesty mezi vstupy a výstupy, přičemž zachovávají samosměřovací vlastnosti sítě a jen minimálně komplikují její časovou složitost. Zlepší se spolehlivost i propustnost sítě.
  - Banyan síť s dělenou zátěží
  - Data Manipulator - DM
  - Augmented Data Manipulator - ADM
  - Reverse Augmented Data Manipulator - IADM
  - gama sítě

Základní typy vícestupňových sítí

Unicast sítě bez blokování

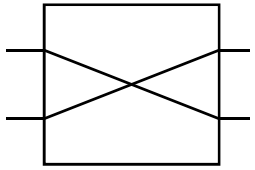
- Všechny sítě s blokováním vyžadují opatření na potlačení blokování. Nejčastěji je to umístění vyrovnávacích pamětí do sítě.
- Problém blokování je možné řešit již při návrhu sítě, vytvořením sítě bez blokování. Rozlišujeme dvě možnosti. Buď jsou sítě topologicky bez blokování, tj jejich architektura minimalizuje pravděpodobnost blokování, nebo jsou s řízením bez blokování, tj mechanismus řízení sítě odstraňuje blokování. Druhý případ nazýváme i rekonfigurovatelné sítě.
  - Benešova síť
  - Paralelní baseline síť
  - Closova síť
  - Batcherova síť
  - Batcher-banyan síť

Základní typy vícestupňových sítí

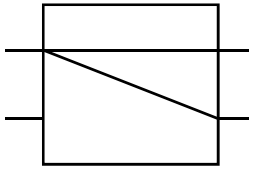
Multicast sítě

- Multicast vícestupňové spojovací sítě jsou obvykle sítě s N vstupy a N výstupy, přičemž **libovolná skupina vstupů se může propojit s libovolnou skupinou výstupů**. Každý vstupní port může být spojen s více než jedním výstupem, ale každý výstupní port je obvykle spojen s nejvíce jedním vstupním portem.
- Multicast sítě mohou uskutečnit N^N různých spojení, mají tedy větší výkon než unicast sítě, které uskutečňují one-to-one permutaci vstupů na výstupy a realizují maximálně N! různých spojení.
- V podstatě jako multicast síť může fungovat každá unicast síť pokud její spojovací elementy dokáží propojit své vstupy na několik výstupů.

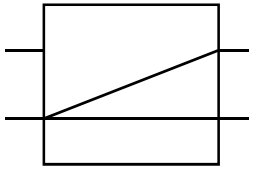
Straight-trough



Exchange

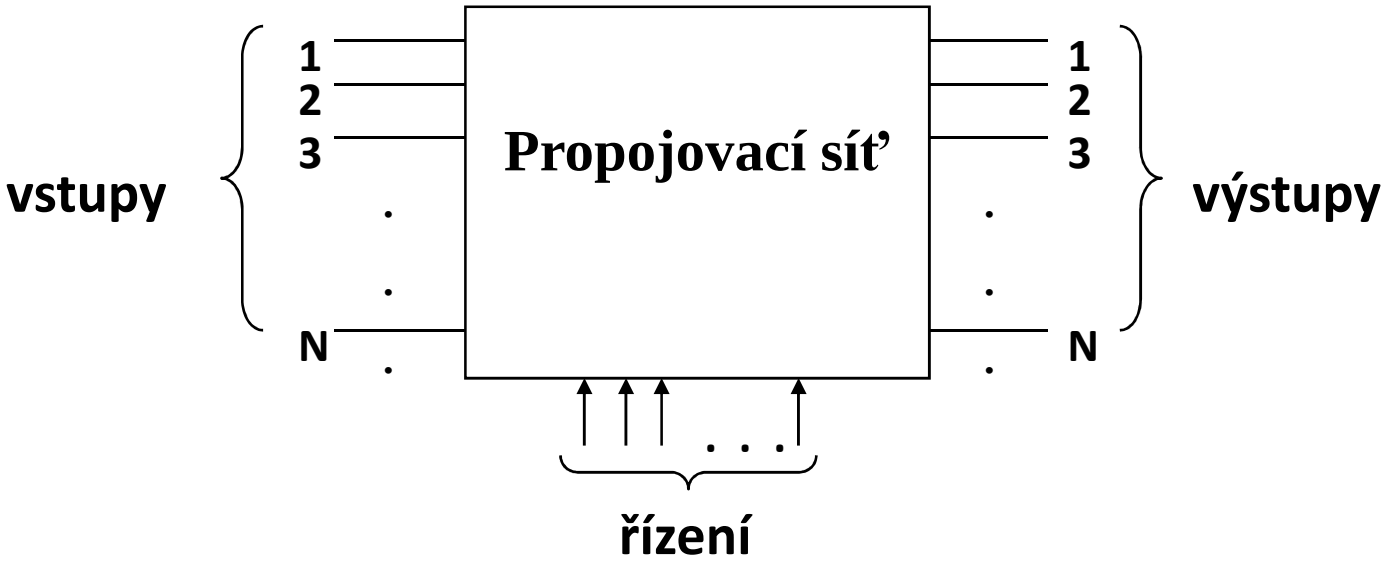


Upper-broadcast



Lower-broadcast

Blokové schéma propojovací sítě



- Propojovací síť - PS [N x N] je prostředek, který umožňuje propojit libovolný ze svých N vstupů (X0, X1, ..., XN-1) s libovolným ze svých N výstupů (Y0, Y1, ... YN-1).
- Základní význam mají PS, které realizují pouze navzájem jednojednoznačné přiřazení, tzn. s jedním vstupem může být spojen pouze jediný výstup -> **permutační sítě**

Základní permutace

Základní permutace:

- **permutace dokonalého promíchání** (**σ**) - perfect shuffle
- **motýlková permutace** (β) - Butterfly
- **permutace reverzace** (ρ)
- **permutace výměny** (**E**) - exchange

- Permutace dokonalého promíchání:

$$\sigma(j) = \left(2j + \left\lfloor \frac{2j}{N} \right\rfloor\right) mod\ N$$

- Obecně je operace promíchání definovaná ve tvaru:

$$\sigma(j,K) = \left(Kj + \left\lfloor \frac{Kj}{N} \right\rfloor\right) mod\ N$$

- kde K je "počet rukou" potřebných na promíchání, pro které platí K = 2^k, k = 0, 1, ..., n-1.

Permutace dokonalého promíchání

- Pokud máme číslo *x* v binární reprezentaci, pak permutace dokonalého promíchání odpovídá cyklickému posuvu binární reprezentace o jeden bit vlevo:

$$\sigma(x) = \sigma([x_n \; x_{n-1} \; x_{n-2} \ldots x_2 \; x_1]) = [x_{n-1} \; x_{n-2} \ldots x_1 \; x_n]$$

- Permutace dokonalého k-tého podpromíchání $\sigma_{(k)}(x)$ - je definována pro $1 \leq k \leq n$ následovně:

$$\sigma_{(k)}(x) = \sigma_{(k)}([x_n \; x_{n-1} \; x_{n-2} \ldots x_2 \; x_1]) = [x_n \; \ldots x_{k+1} \; x_{k-1} \ldots x_1 \; x_k]$$

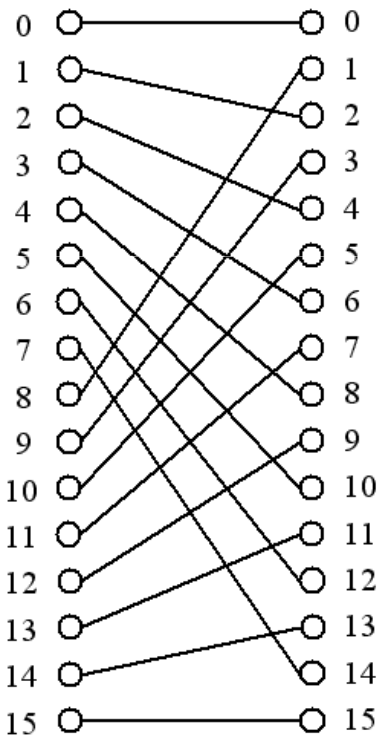
- Takže platí:

$$\sigma_{(1)}(x) \equiv x, \quad \sigma_{(n)}(x) \equiv \sigma(x)$$

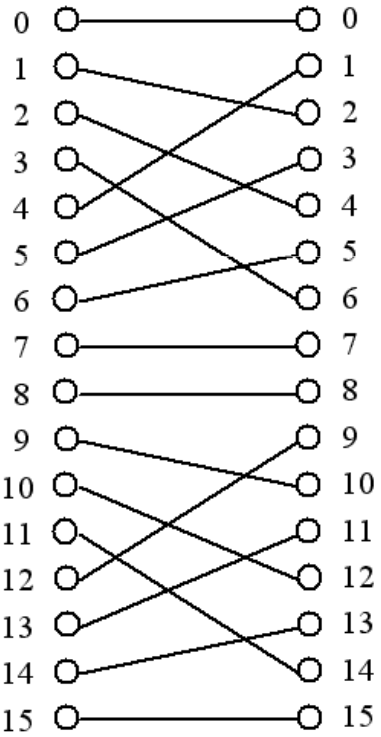
Permutace dokonalého promíchání

Například pro N=16, n=$\log_2$N=4:

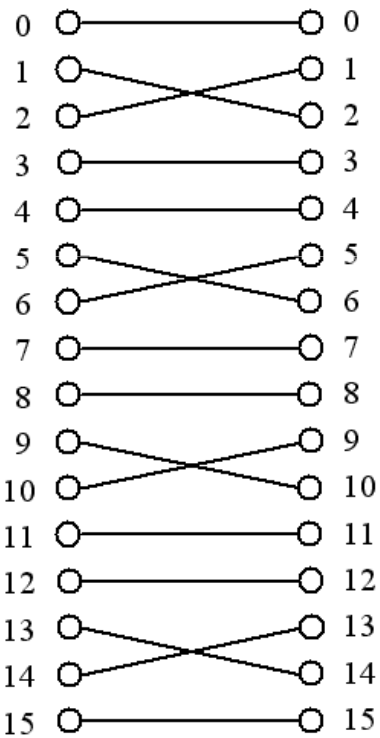
- $\sigma_{(4)}(0) = \sigma_{(4)}([0000]) = [0000] = 0$
- $\sigma_{(4)}(1) = \sigma_{(4)}([0001]) = [0010] = 2$
- $\sigma_{(4)}(2) = \sigma_{(4)}([0010]) = [0100] = 4$ atd.



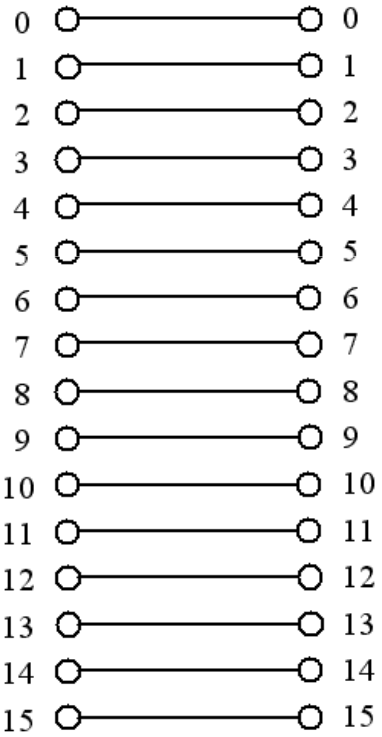
$\sigma_{(4)}(x) \equiv \sigma(x)$



$\sigma_{(3)}(x)$



$\sigma_{(2)}(x)$

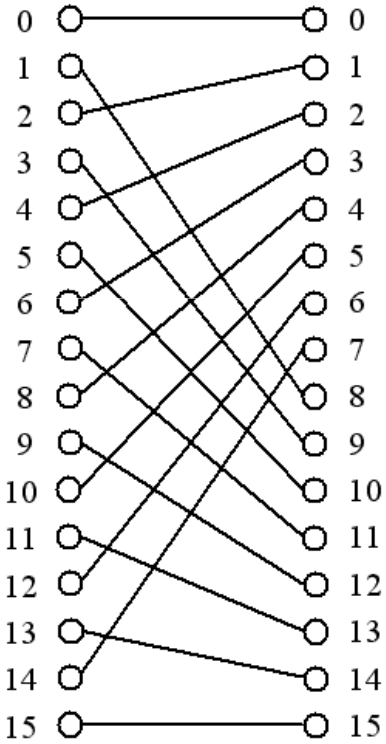


$\sigma_{(1)}(x) \equiv x$

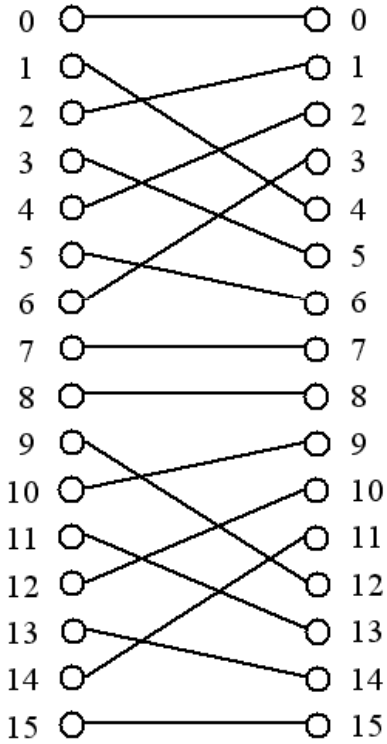
Permutace dokonalého promíchání

Inverzní dokonalé promíchání odpovídá cyklickému posuvu binární reprezentace o jeden bit vpravo:

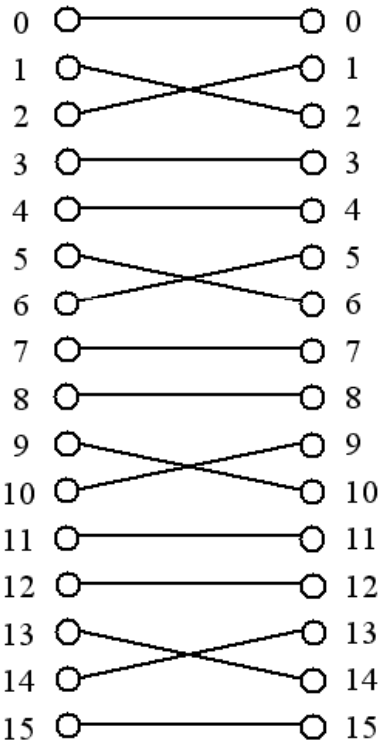
$$\sigma^{-1}(x) = \sigma^{-1}([x_n \; x_{n-1} \; x_{n-2} \ldots x_2 \; x_1]) = [x_1 \; x_n \; x_{n-1} \; x_{n-2} \ldots x_2]$$



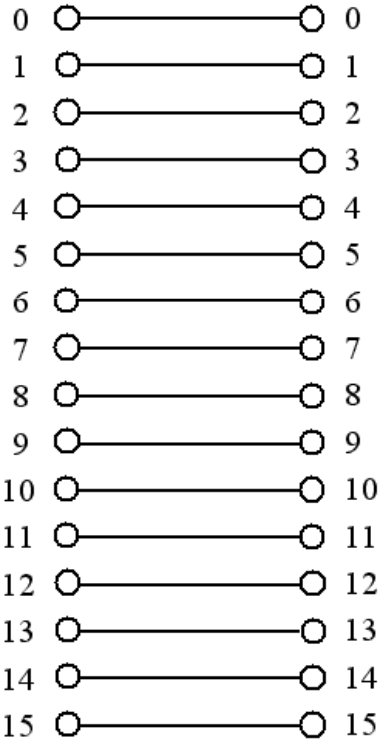
$\sigma_{(4)}^{-1}(x) \equiv \sigma^{-1}(x)$



$\sigma_{(3)}^{-1}(x)$



$\sigma_{(2)}^{-1}(x)$



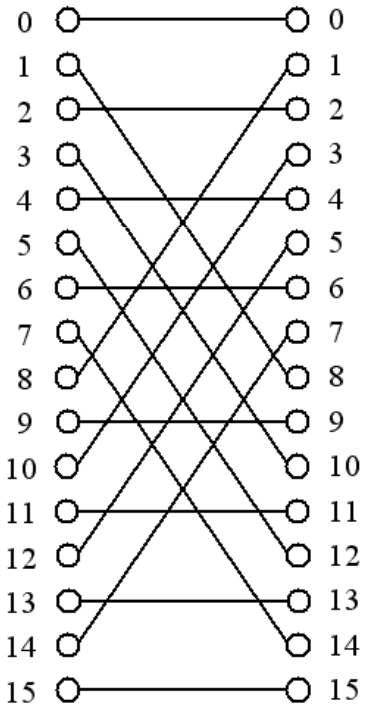
$\sigma_{(1)}^{-1}(x) \equiv x$

Motýlková permutace

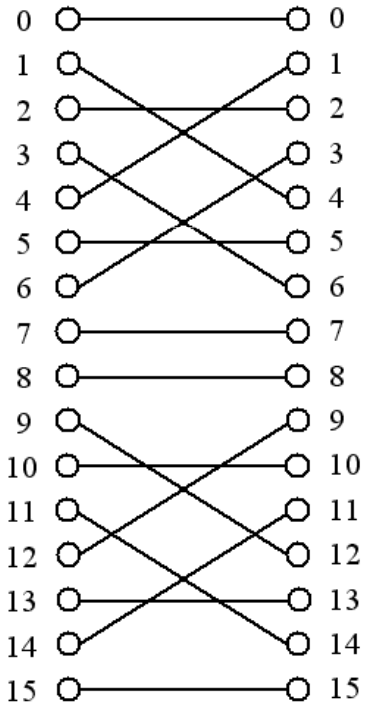
Motýlková permutace (Butterfly):

k-ta motýlková permutace $\beta_{(k)}(x)$ pro $1 \le k \le n$, $n=\log_2 N$ se dostane tak, že se vzájemně vymění první a k-tý bit:

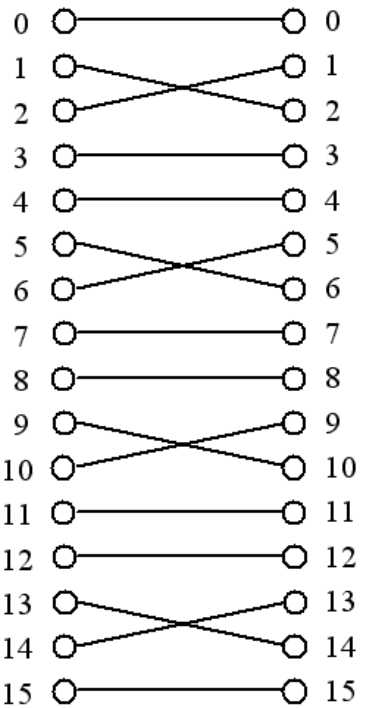
$$\beta_{(k)}(x) = \beta_{(k)}([x_n \; x_{n-1} \; x_{n-2} \ldots x_2 \; x_1]) = [x_n \; x_{n-1} \ldots x_{k+1} \; x_1 \; x_{k-1} \ldots x_2 \; x_k]$$



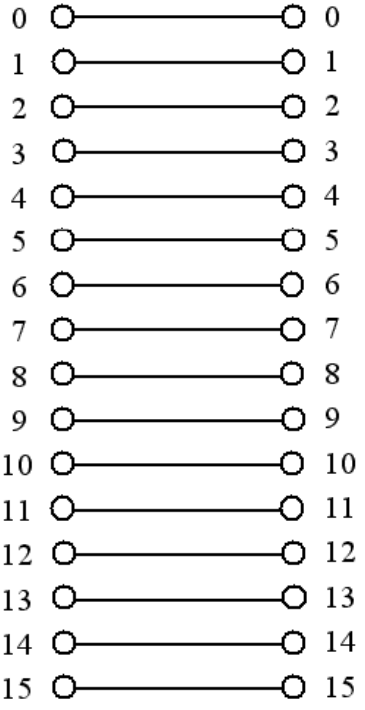
$\beta_{(4)}(x) \equiv \beta_{(4)}^{-1}(x)$



$\beta_{(3)}(x) \equiv \beta_{(3)}^{-1}(x)$



$\beta_{(2)}(x) \equiv \beta_{(2)}^{-1}(x)$



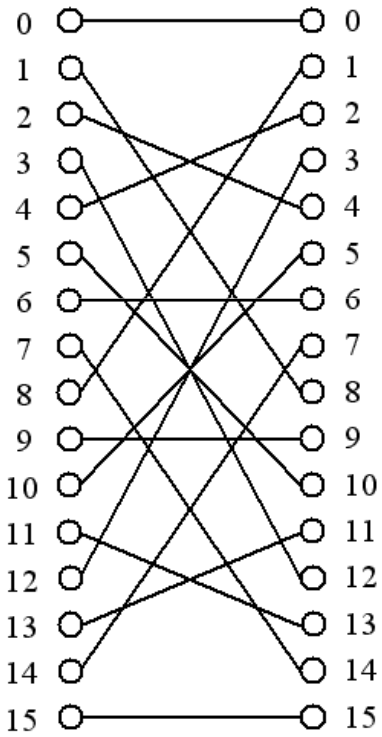
$\beta_{(1)}(x) \equiv \beta_{(1)}^{-1}(x)$

Permutace reverzace

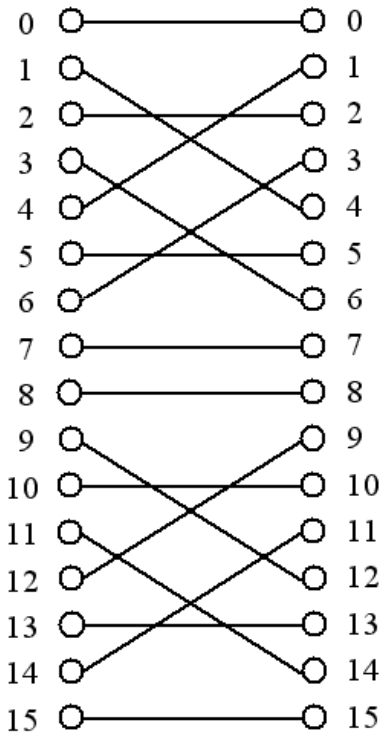
Permutace reverzace:

je dána vzájemnou výměnou bitů vyšších a nižších (zrcadlení):

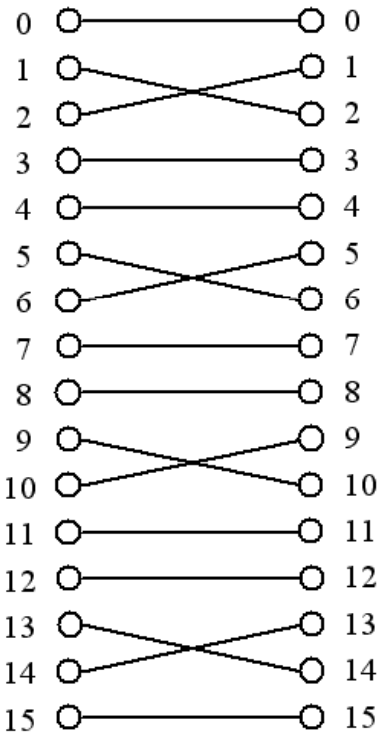
$$\rho(x) = \rho([x_n \; x_{n-1} \; \ldots \; x_2 \; x_1]) = [x_1 \; x_2 \ldots x_{n-1} \; x_n]$$



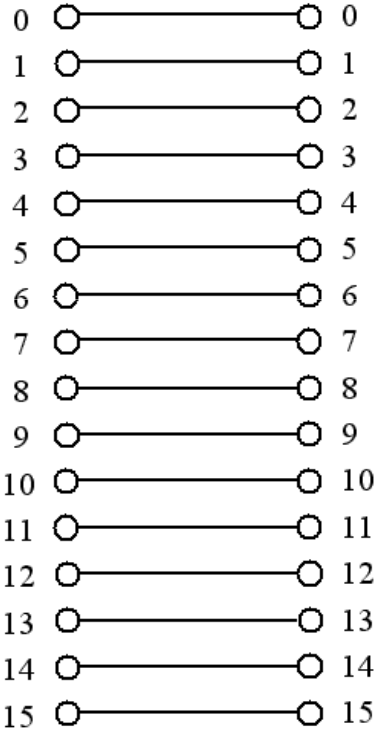
$\rho = \rho_4$



ρ_3



ρ_2



$\rho_1 = x$

Permutace výměny

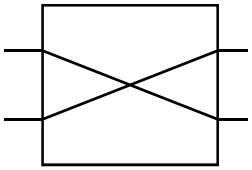
Permutace výměny:

Definujme $\widehat{x}$ vztahem $\widehat{x} = [x_n \; x_{n-1} \; \ldots \; x_2 \; \overline{x_1}]$, kde $\overline{x_1}$ znamená inverzi prvního (nejnižšího) bitu. Pak množina výměnných permutace je:

$$\textbf{bud'} \quad e(x) = x \quad \textbf{a} \quad e(\widehat{x}) = \widehat{x}\,,$$
$$\textbf{nebo} \quad e(x) = \widehat{x} \quad \textbf{a} \quad e(\widehat{x}) = x\,.$$

Straight-trough

(na přímo)



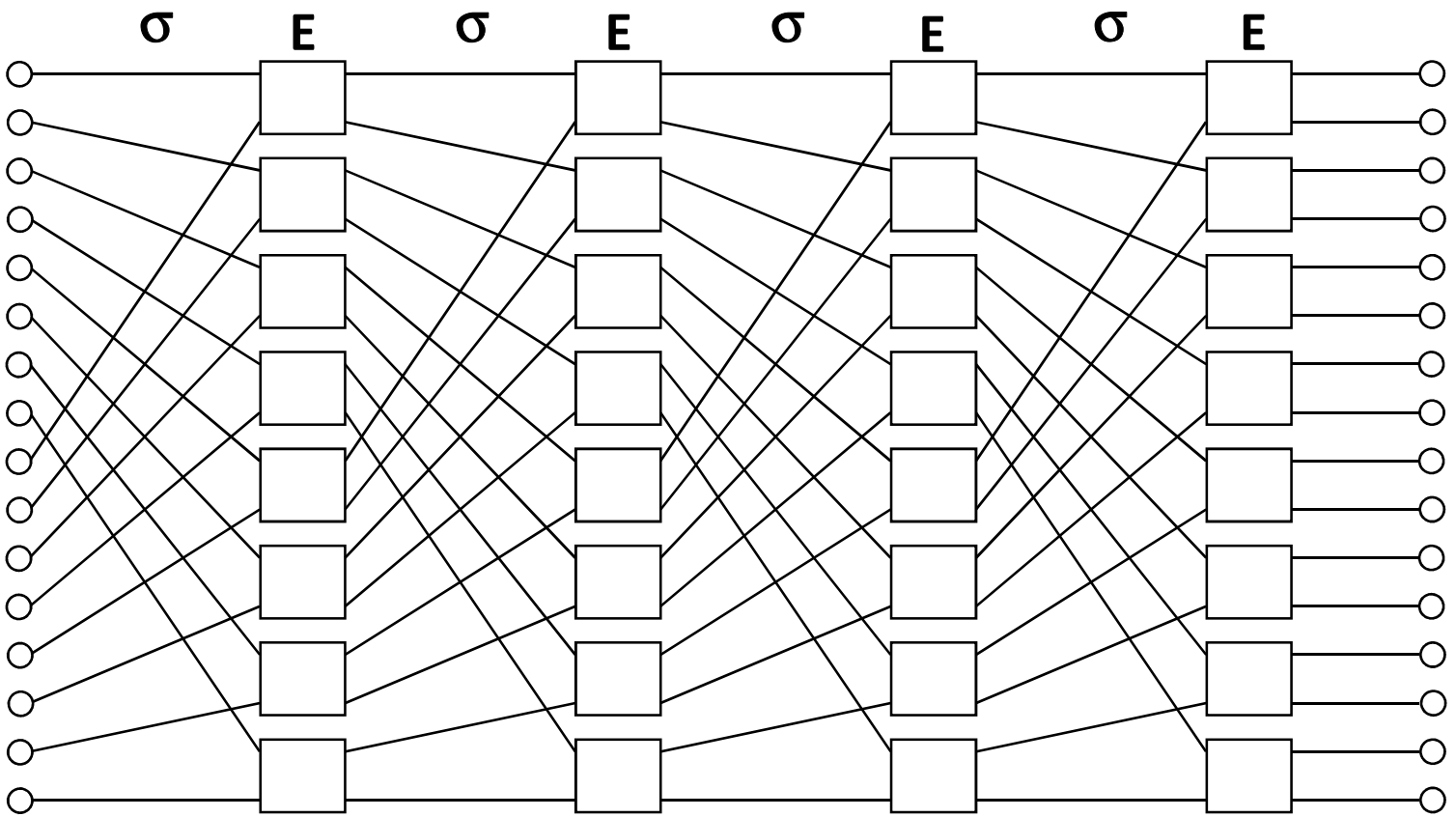
Exchange

(výměna)

Omega síť

$\Omega_N = \sigma E \sigma E \ldots \sigma E = (\sigma E)^n \quad n=\log_2 N$

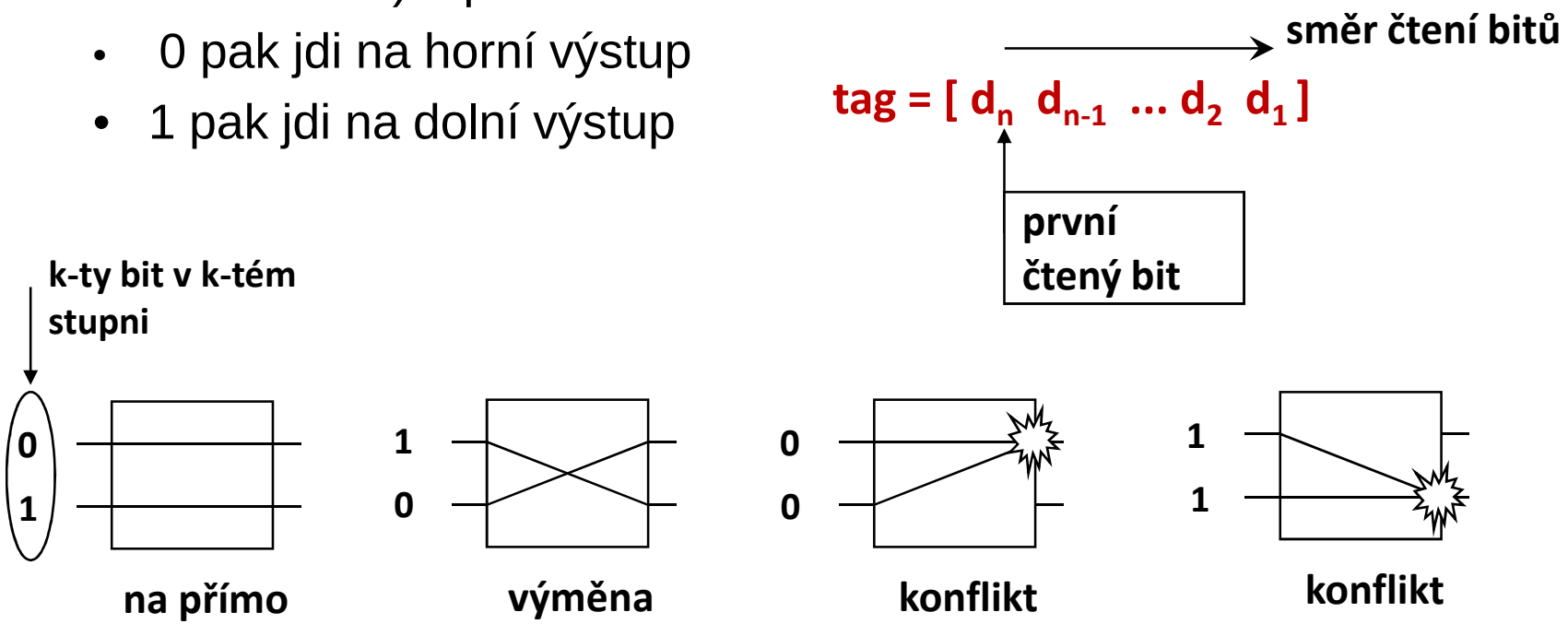
- Napr. pre N=16, n=4: $\Omega_{16} = (\sigma E)^4$



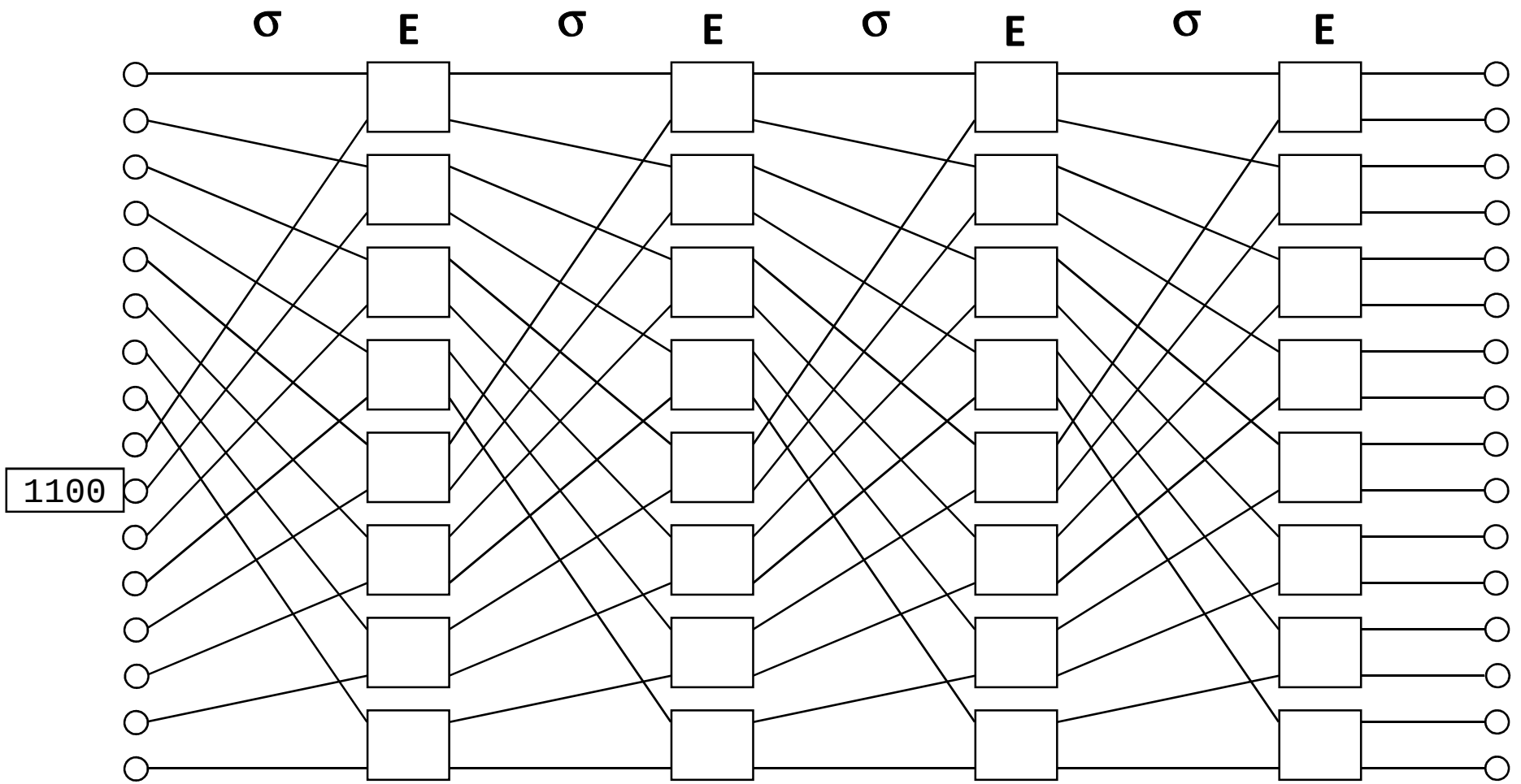
Omega síť

Samosměrovací algoritmus:

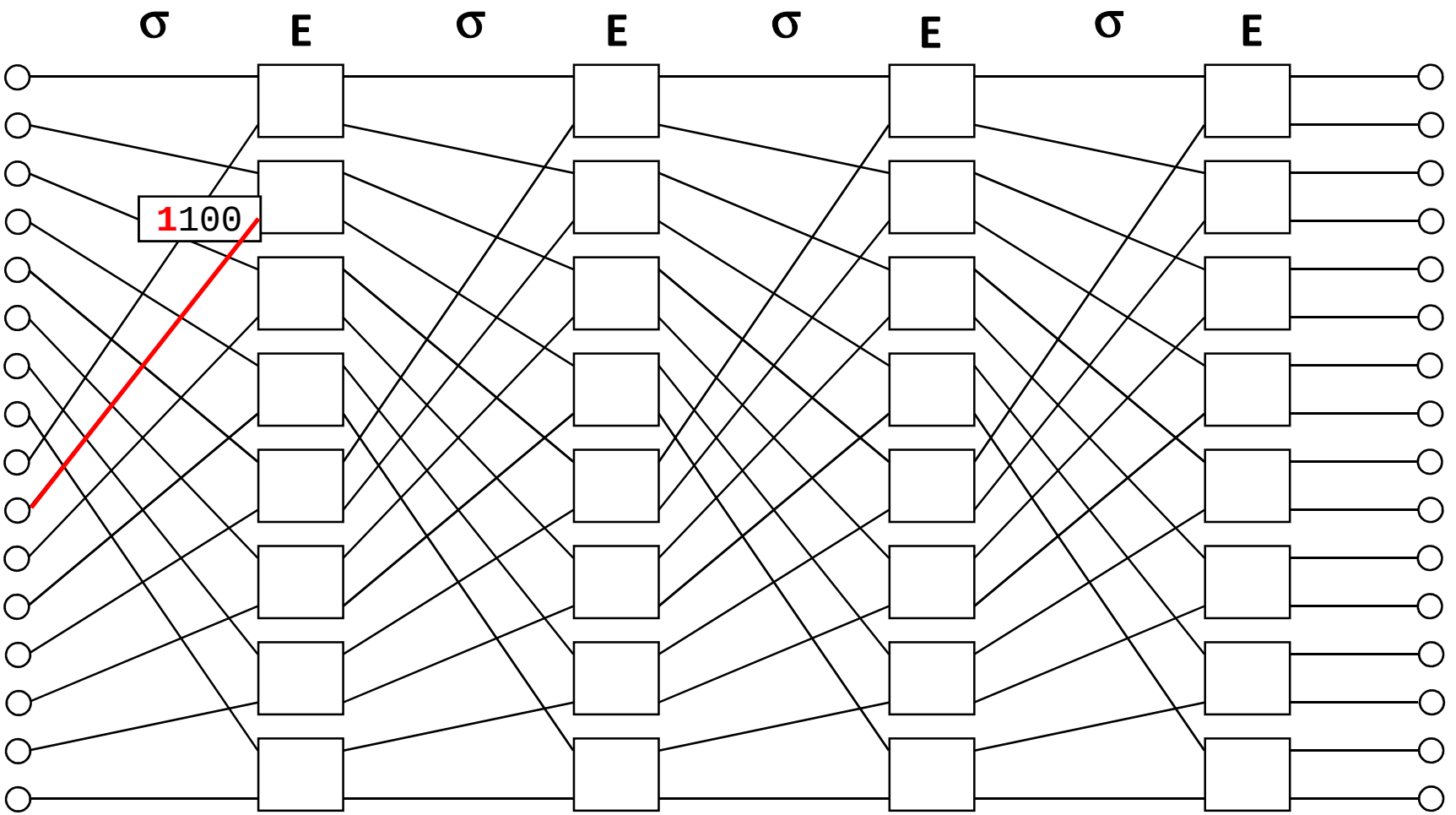
- Na základě adresy určení, tj binární reprezentace čísla výstupního portu (routing tag = destination port id) je možné řídit nastavení přepínačů v jednotlivých stupních sítě následovně. Přepínač v k-tém stupni sítě čte k-tý bit směrovacího návěští (počínaje od MSB směrem k LSB) a pokud se tento rovná:
  - 0 pak jdi na horní výstup
  - 1 pak jdi na dolní výstup



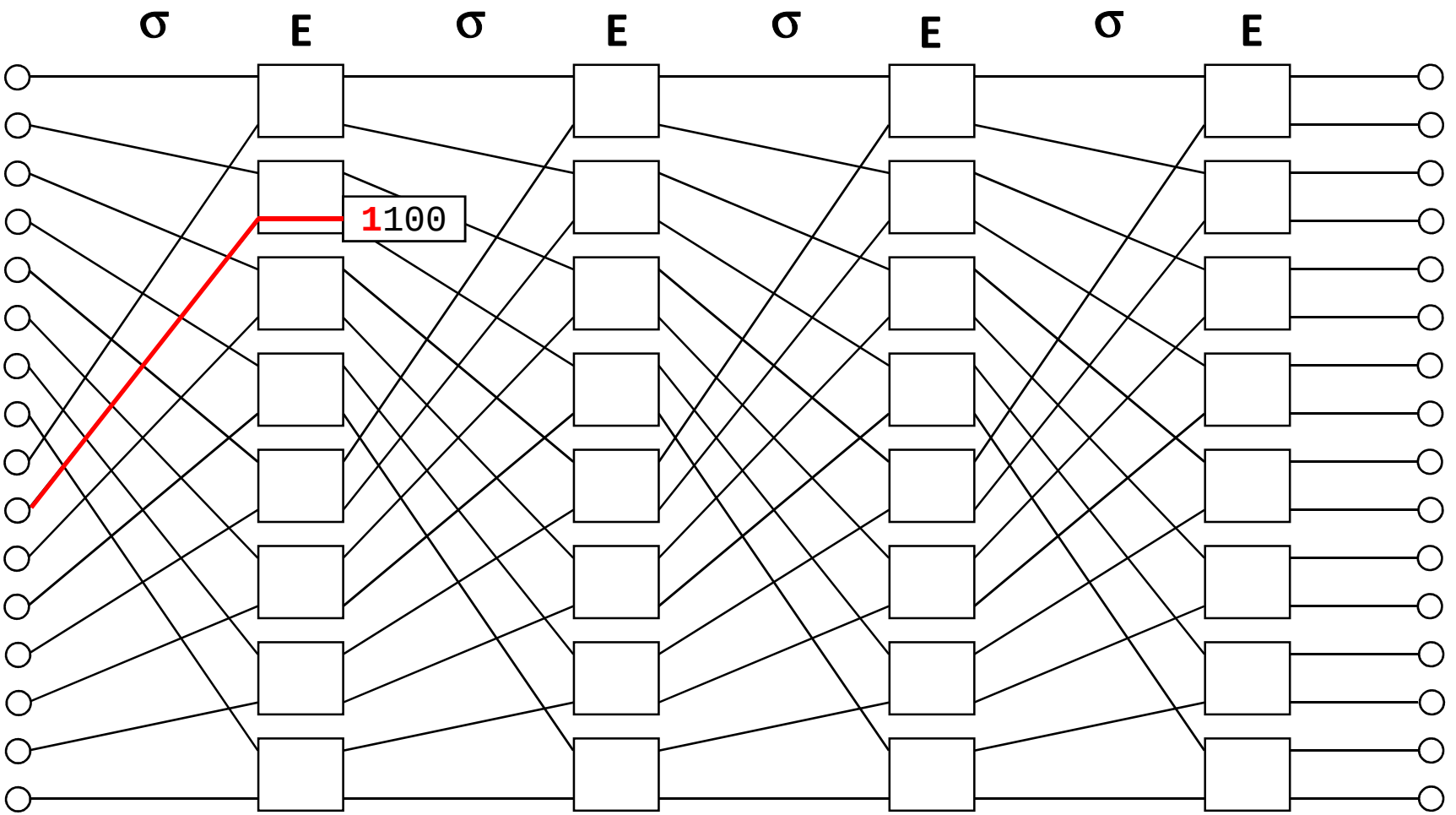
Omega síť



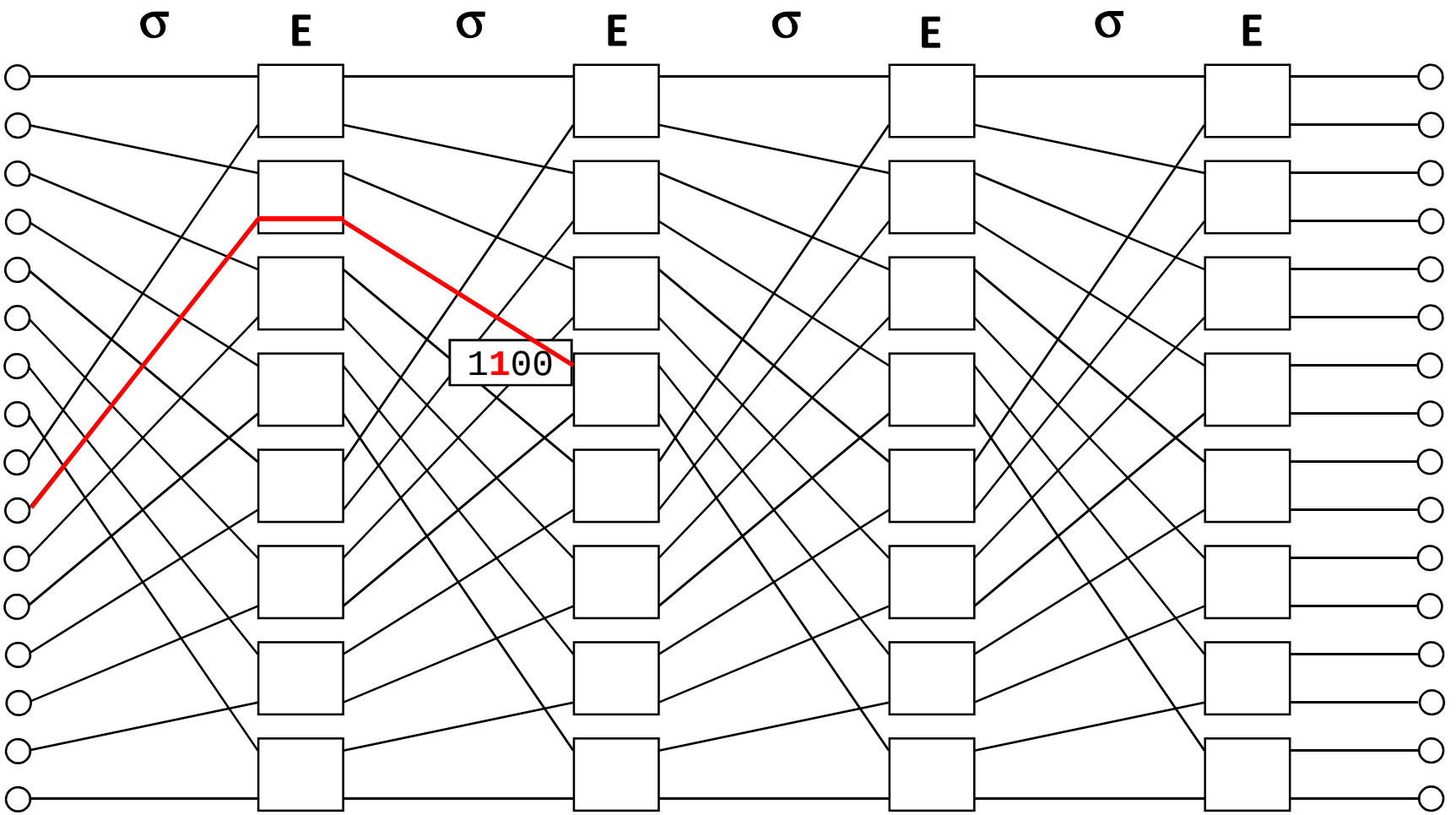
Omega síť



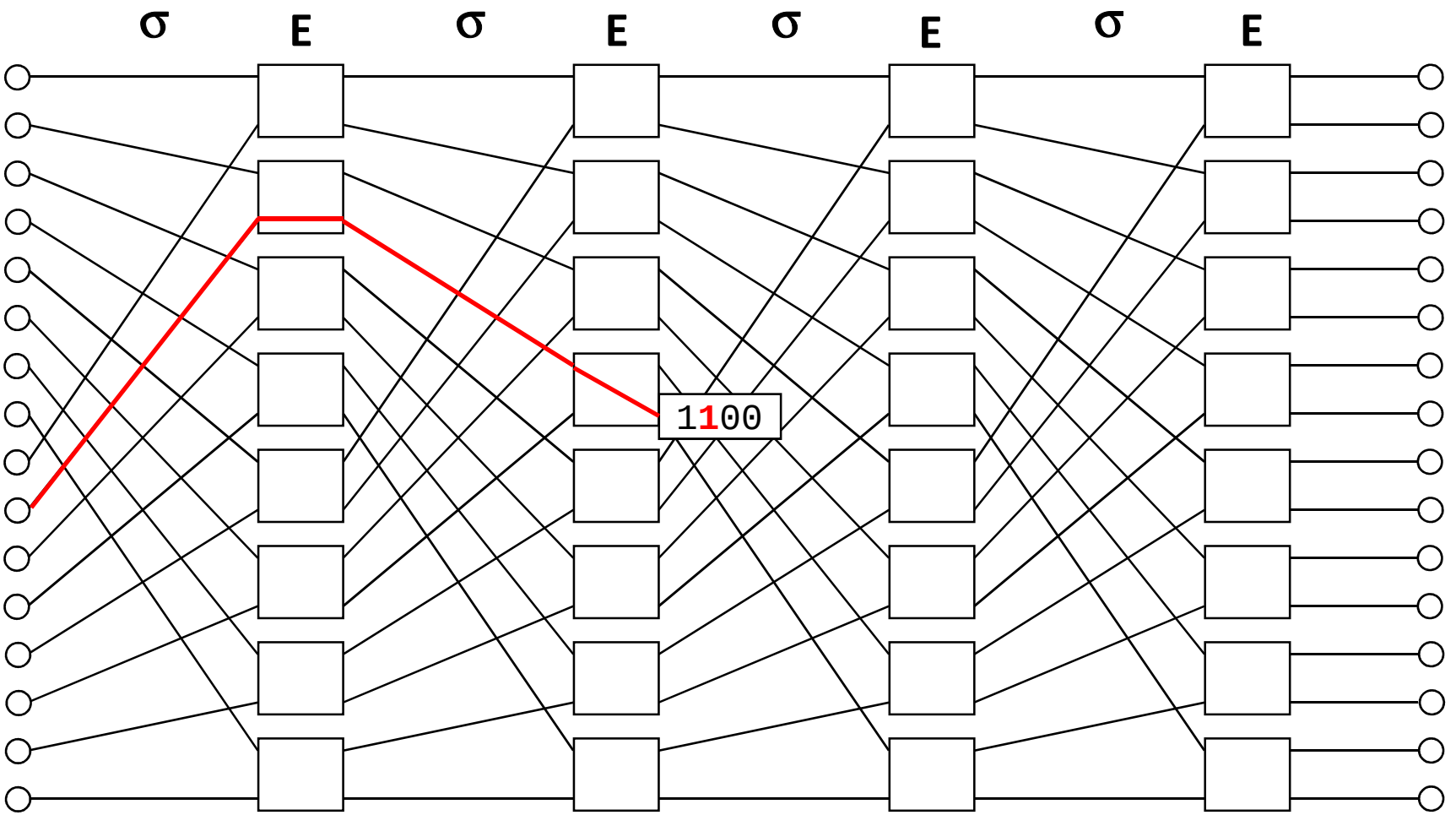
Omega síť



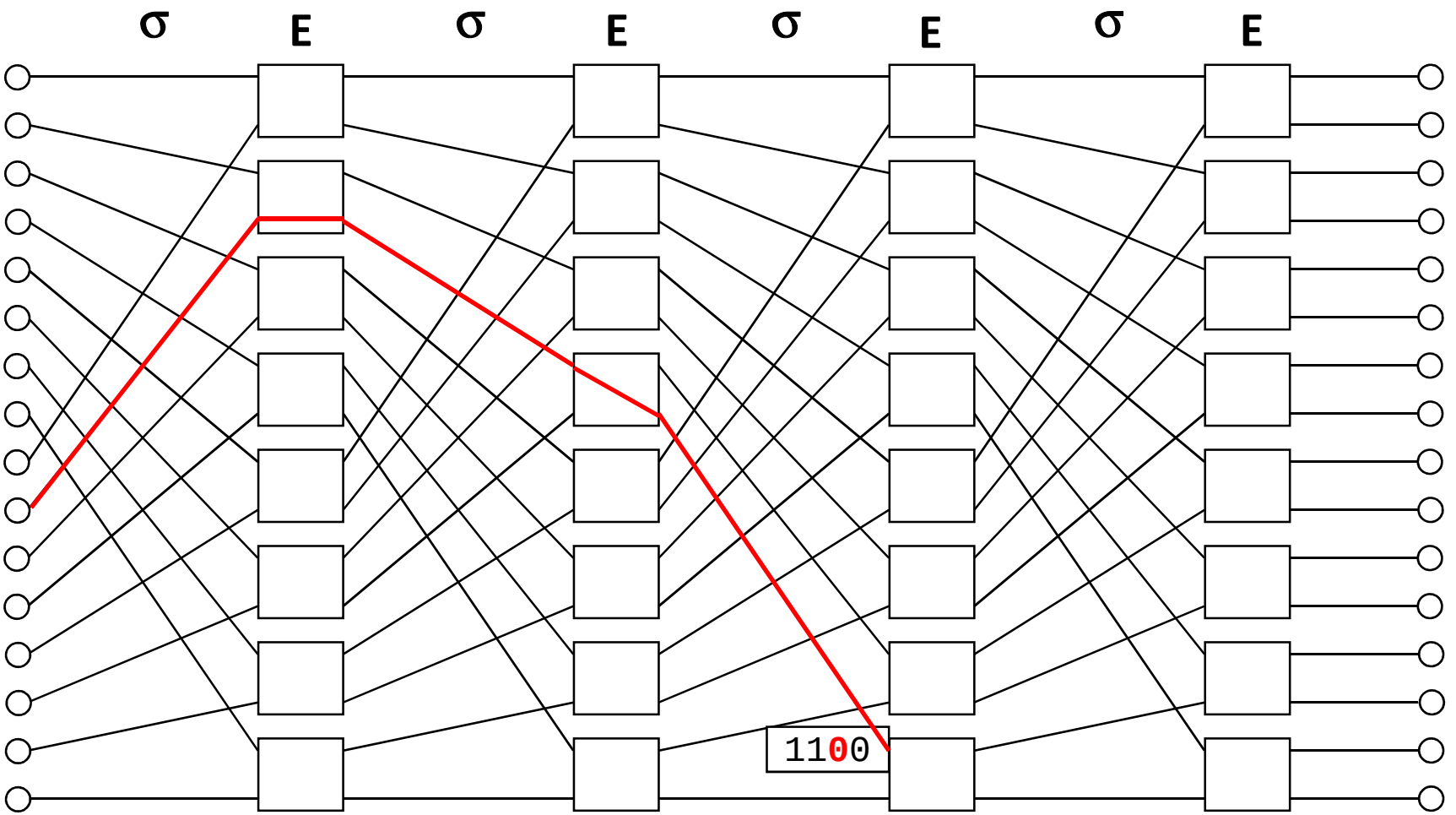
Omega síť



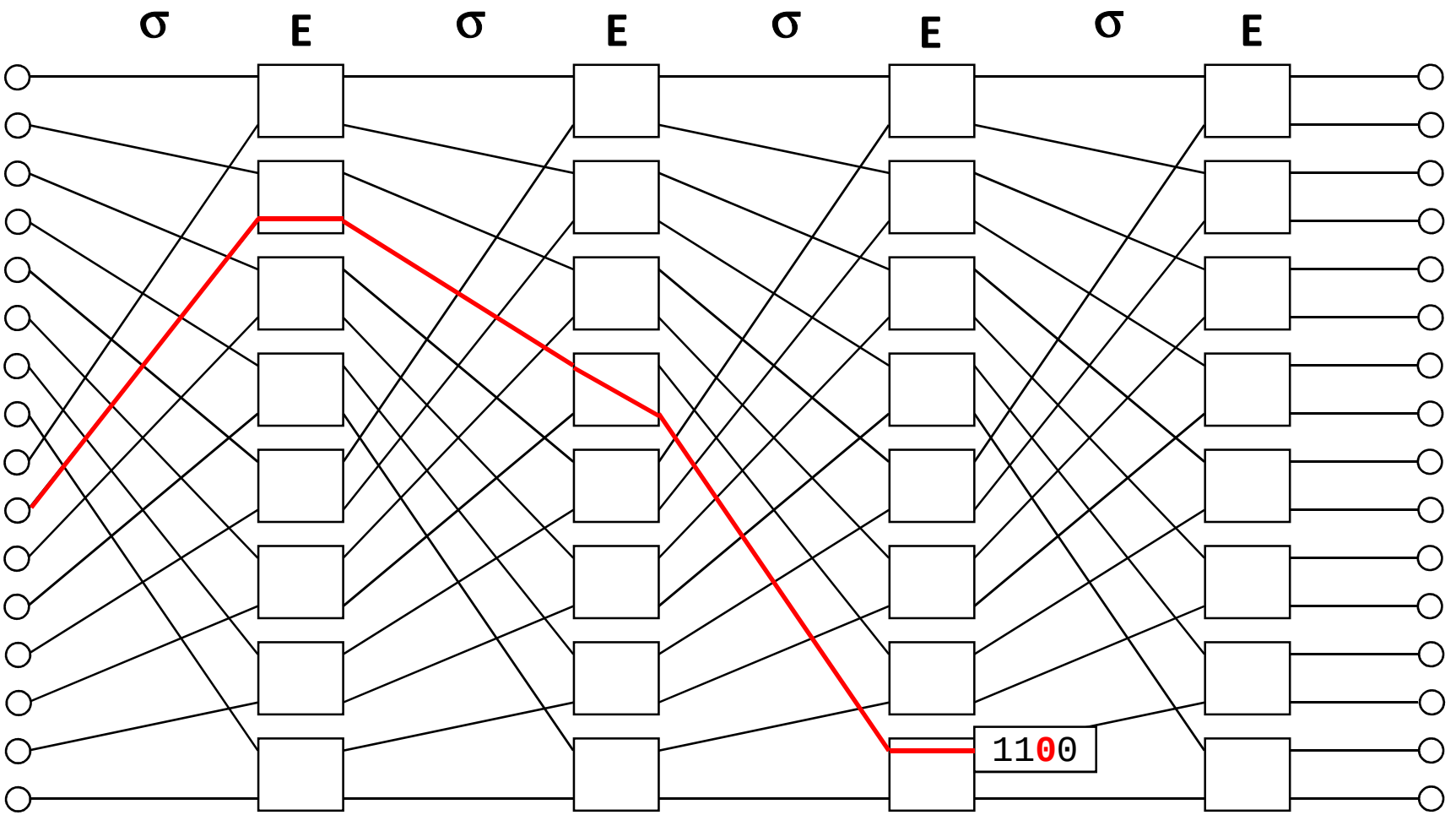
Omega síť



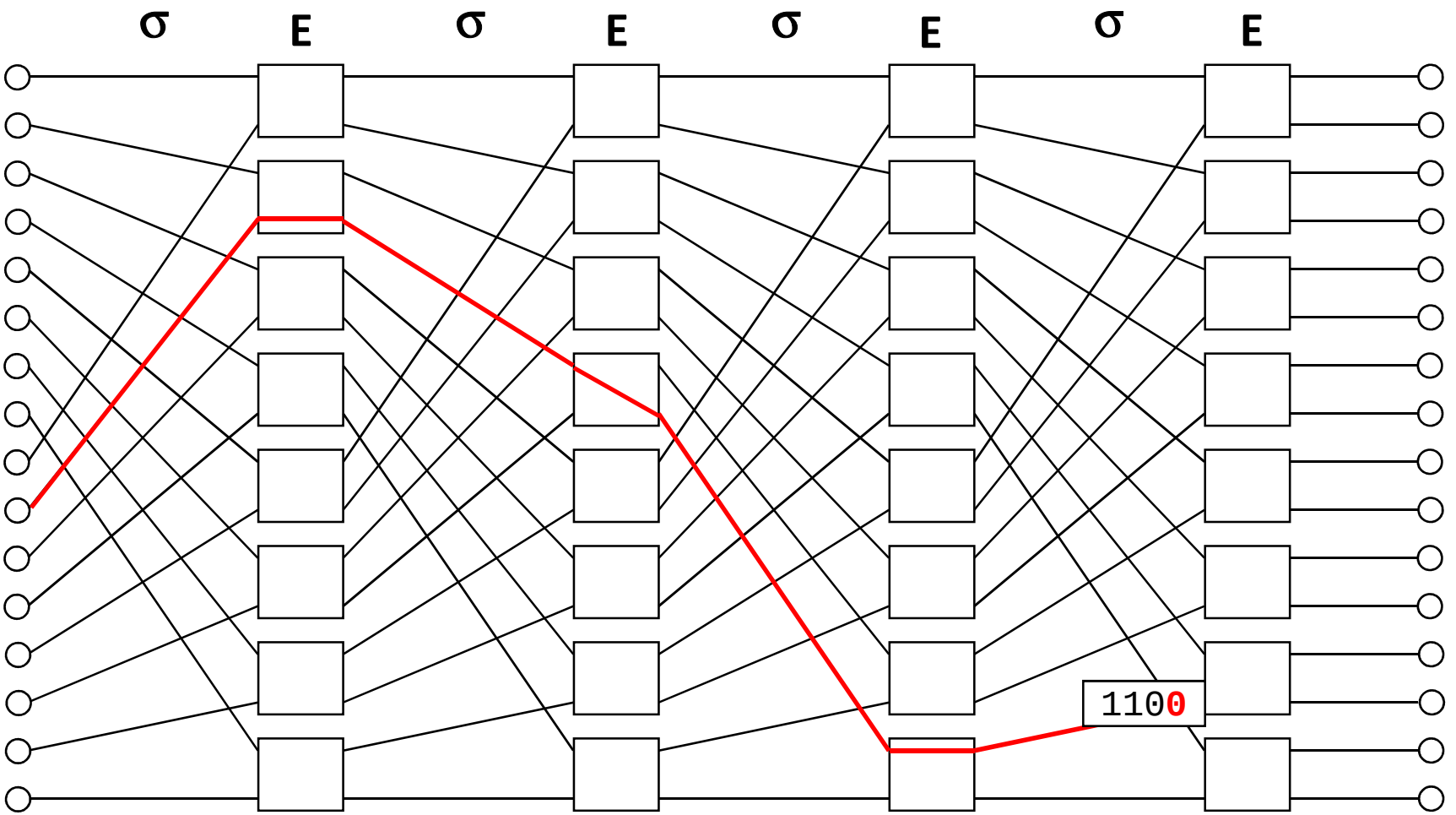
Omega síť



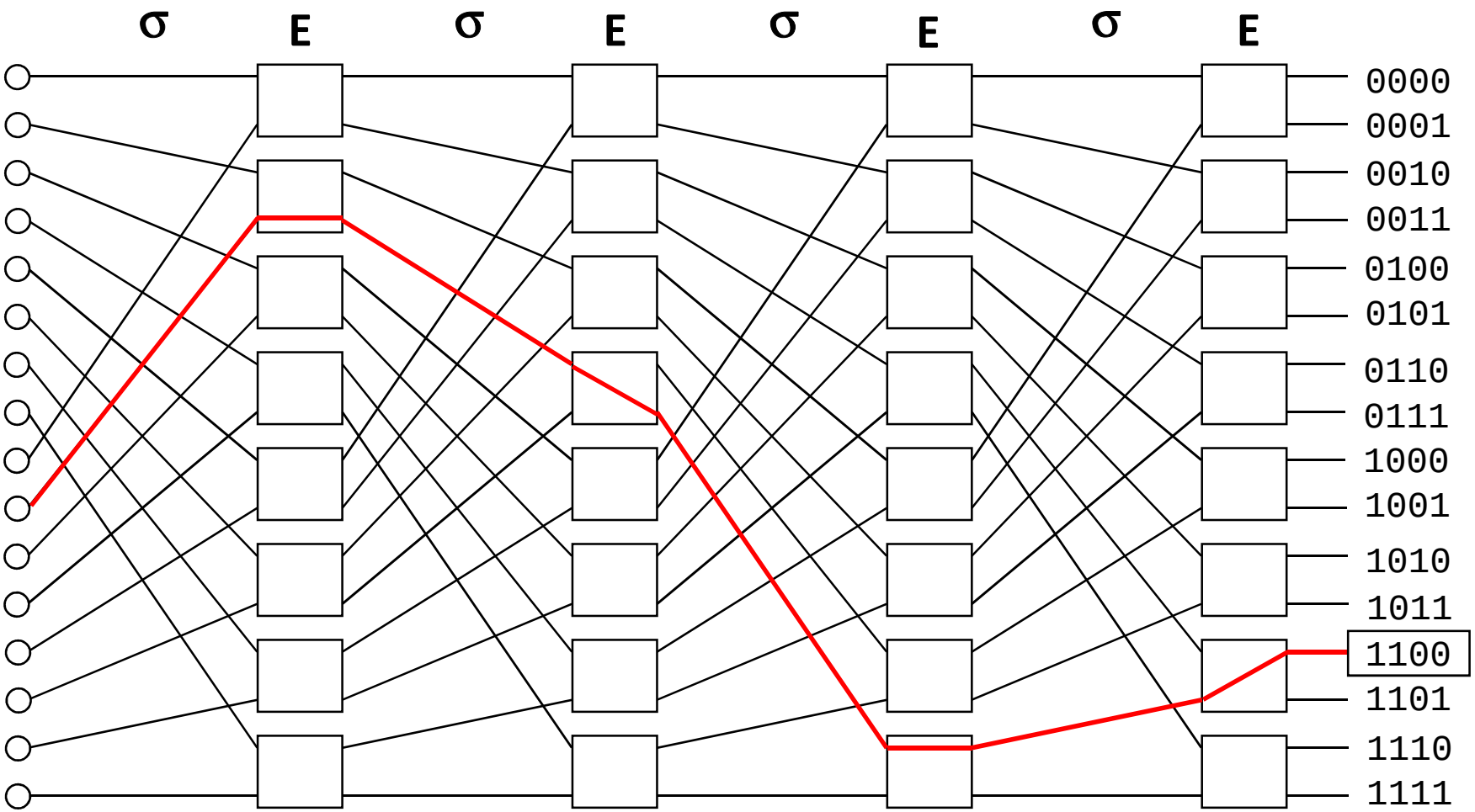
Omega síť



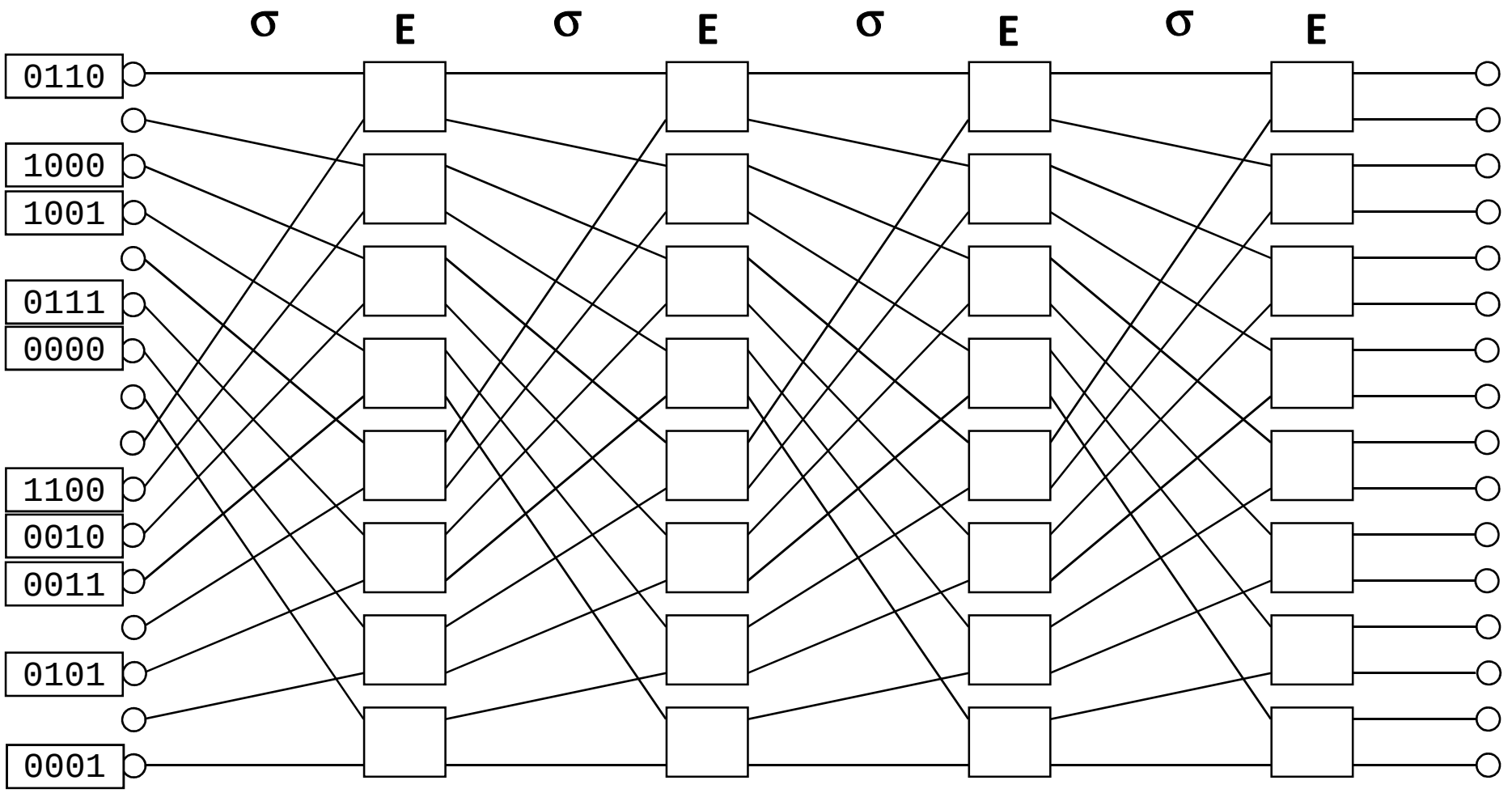
Omega síť



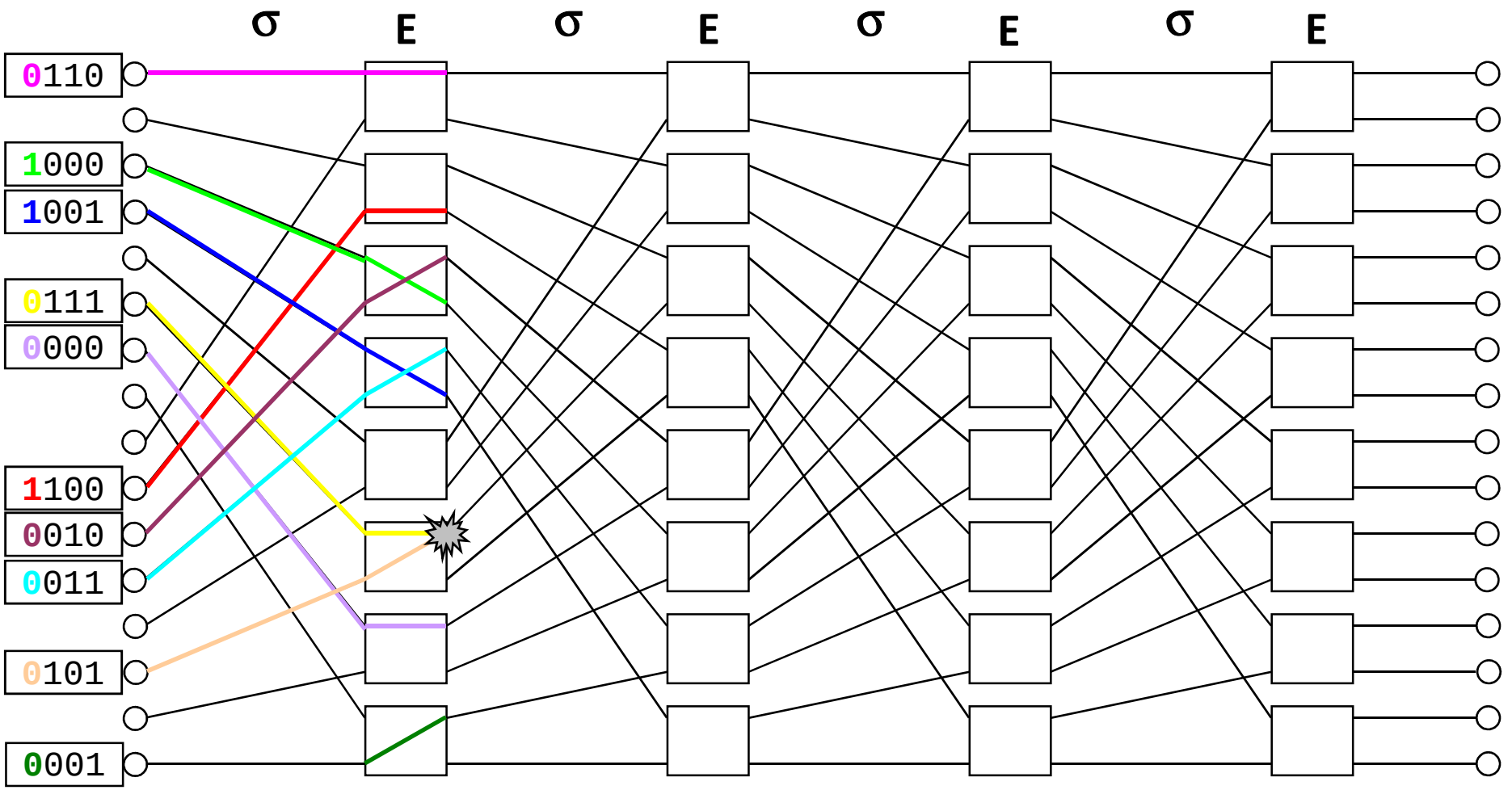
Omega síť


σ
E
σ
E
σ
E
σ
E
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

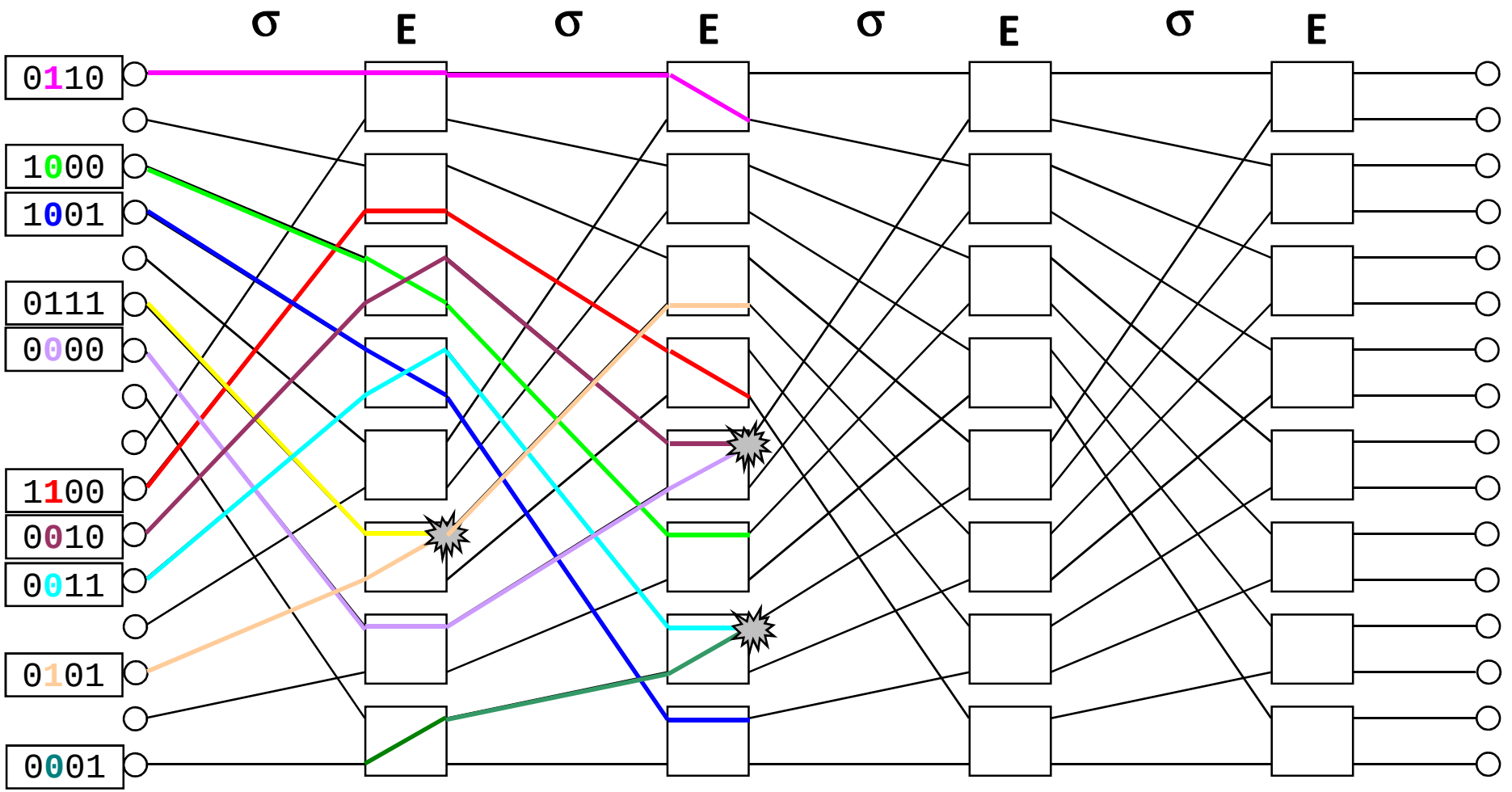
Omega síť



Omega síť

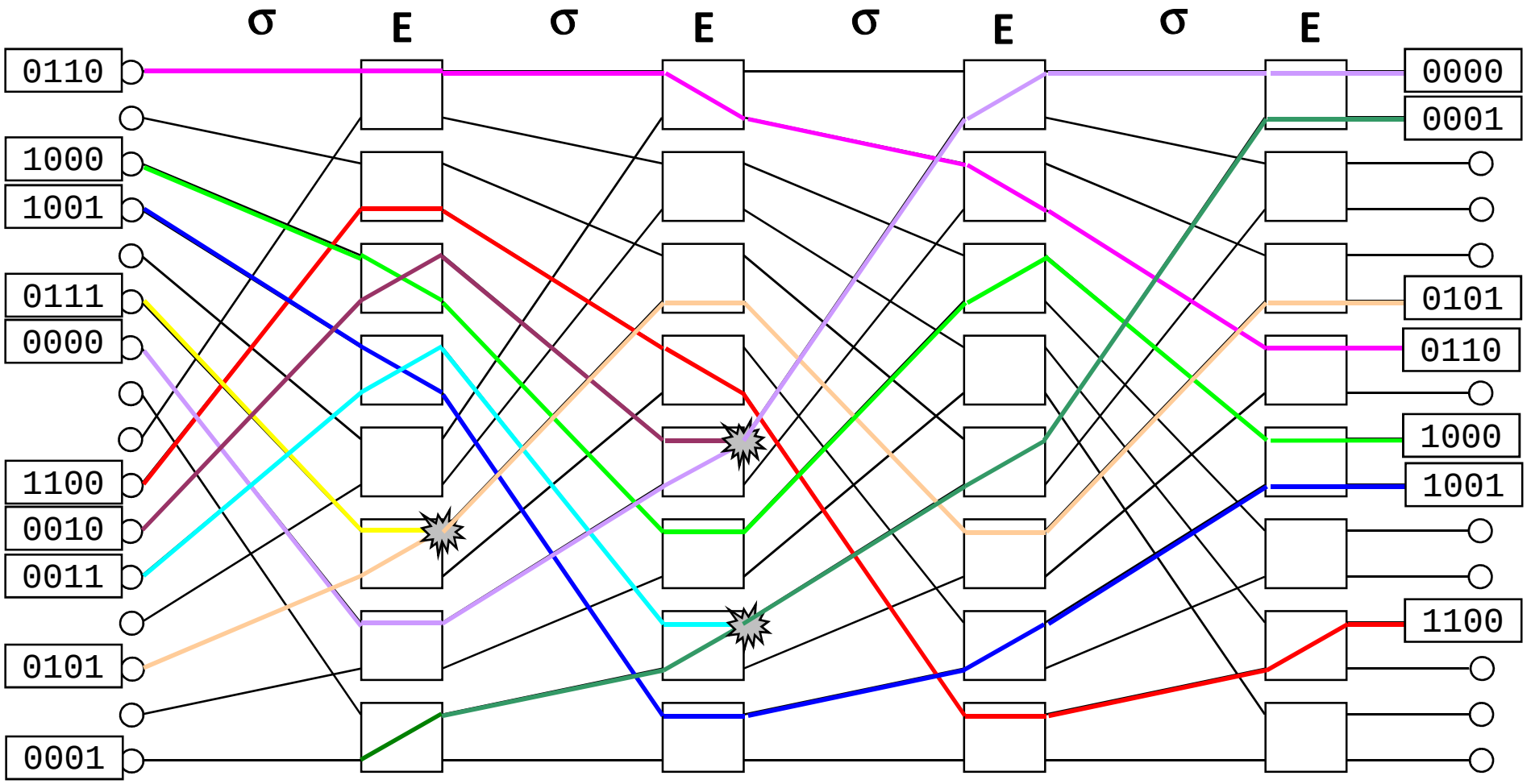


Omega síť



Omega síť

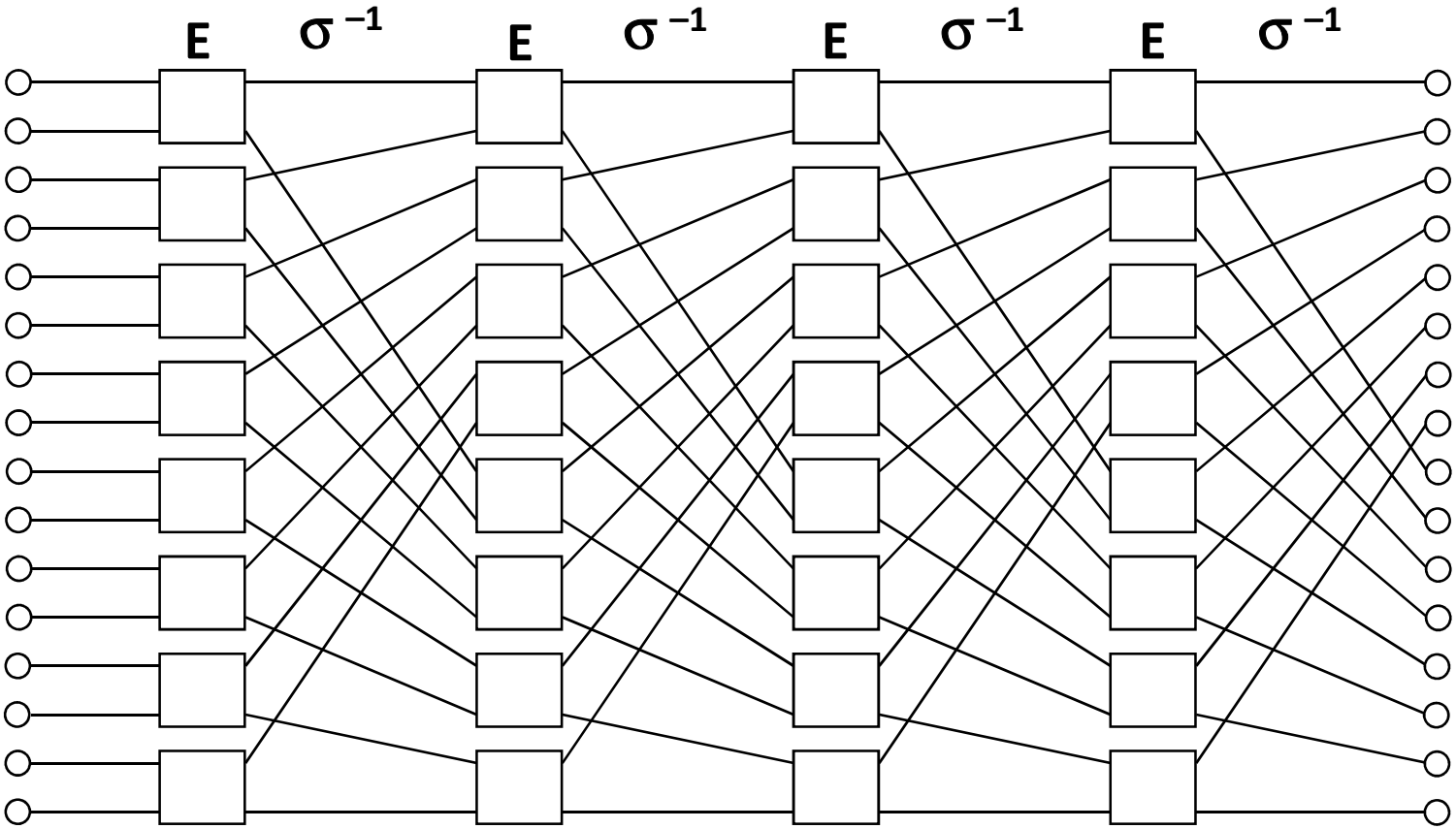
k výstupu dorazilo 7 z 10



Inverzní omega síť

$$\Omega_N^{-1} = (\Omega_N)^{-1} = ((\sigma E)^n)^{-1} = ((\sigma E)^{-1})^n = (E^{-1}\sigma^{-1})^n = (E\,\sigma^{-1})^n$$

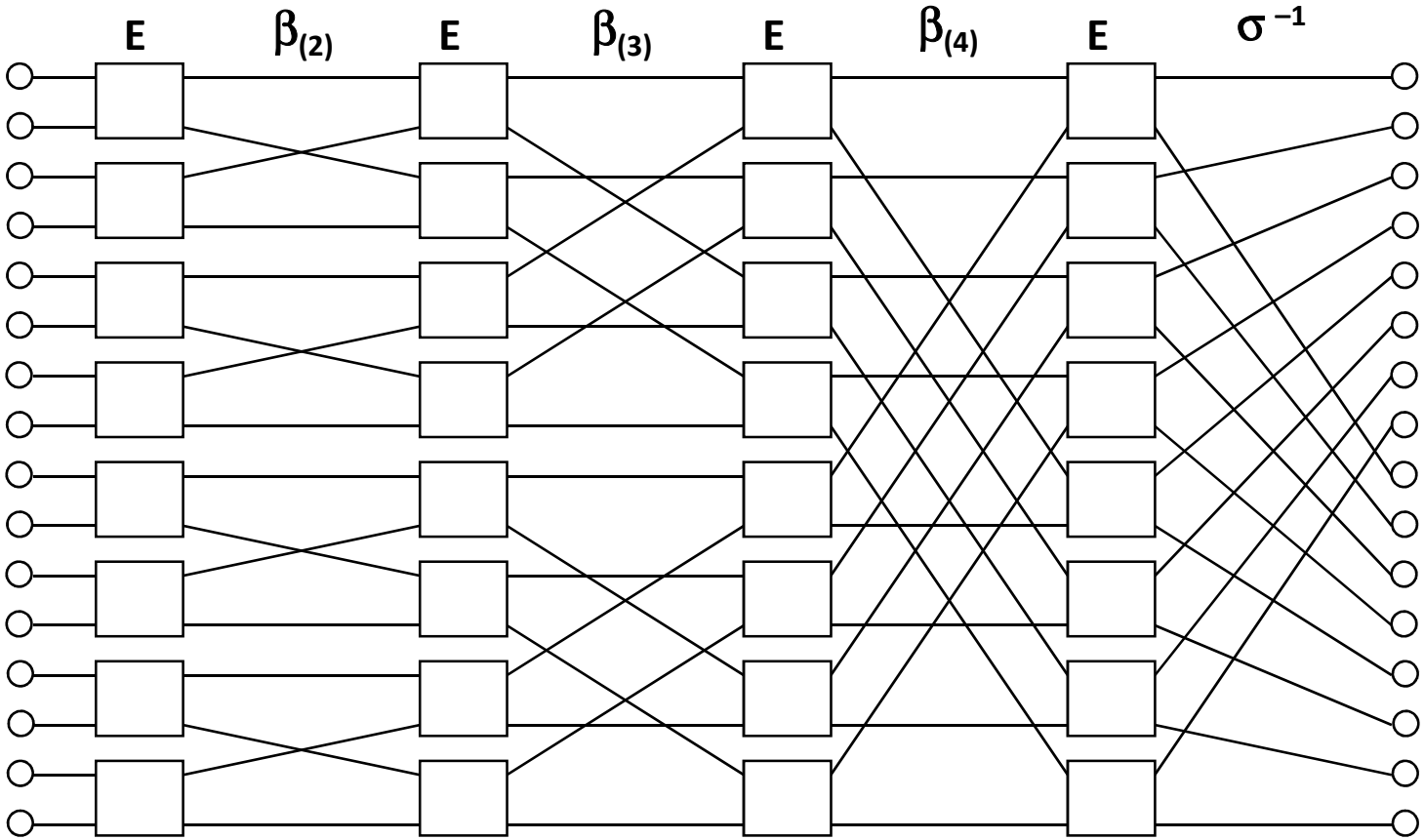
Takže $\Omega_N^{-1} = (E\,\sigma^{-1})^n$



Nepřímá binární n-kubická síť (Indirect binary cube)

$$C_N = E\beta_{(2)}E\beta_{(3)}E\beta_{(4)} \ldots E\beta_{(n)}E\sigma^{-1}$$

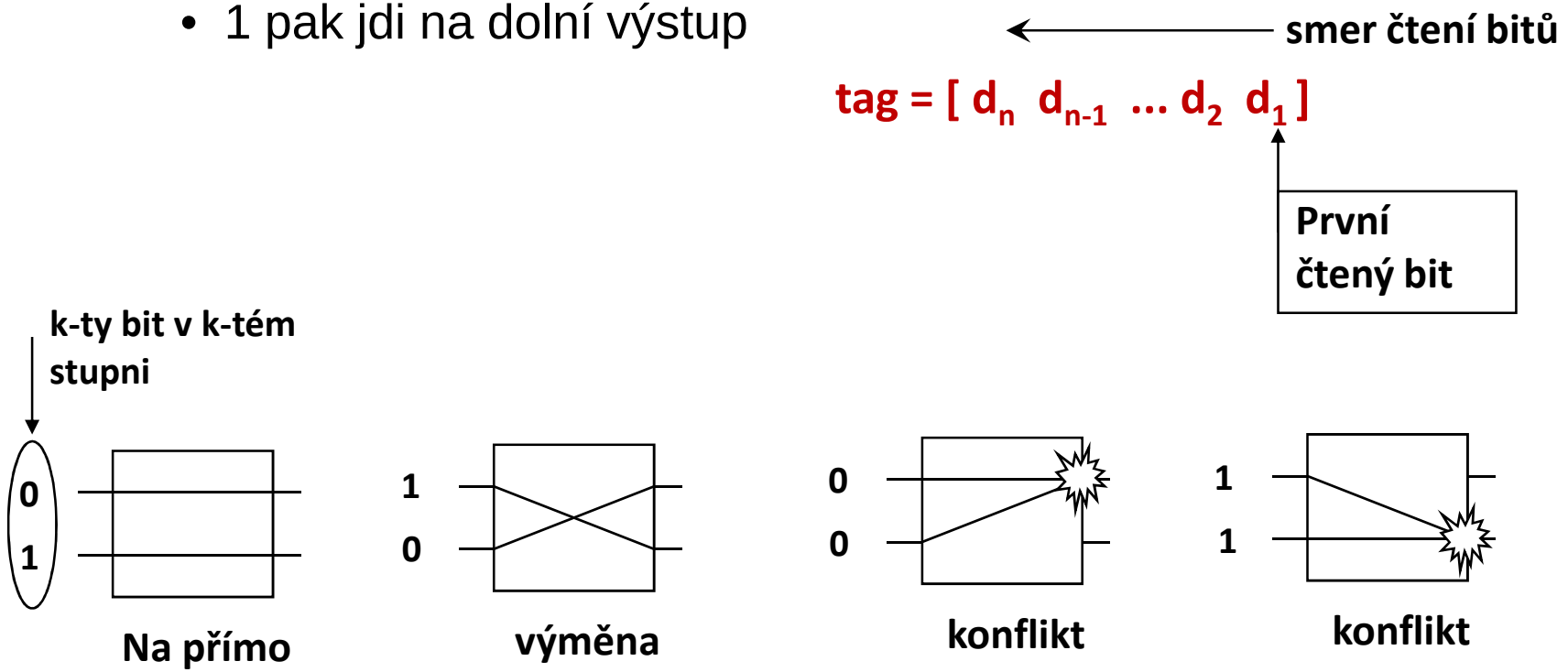
Napr. pro N=16, n=4 $\rightarrow C_{16} = E\beta_{(2)}\, E\beta_{(3)}\, E\beta_{(4)}\, E\sigma^{-1}$



Nepřímá binární n-kubická síť (Indirect binary cube)

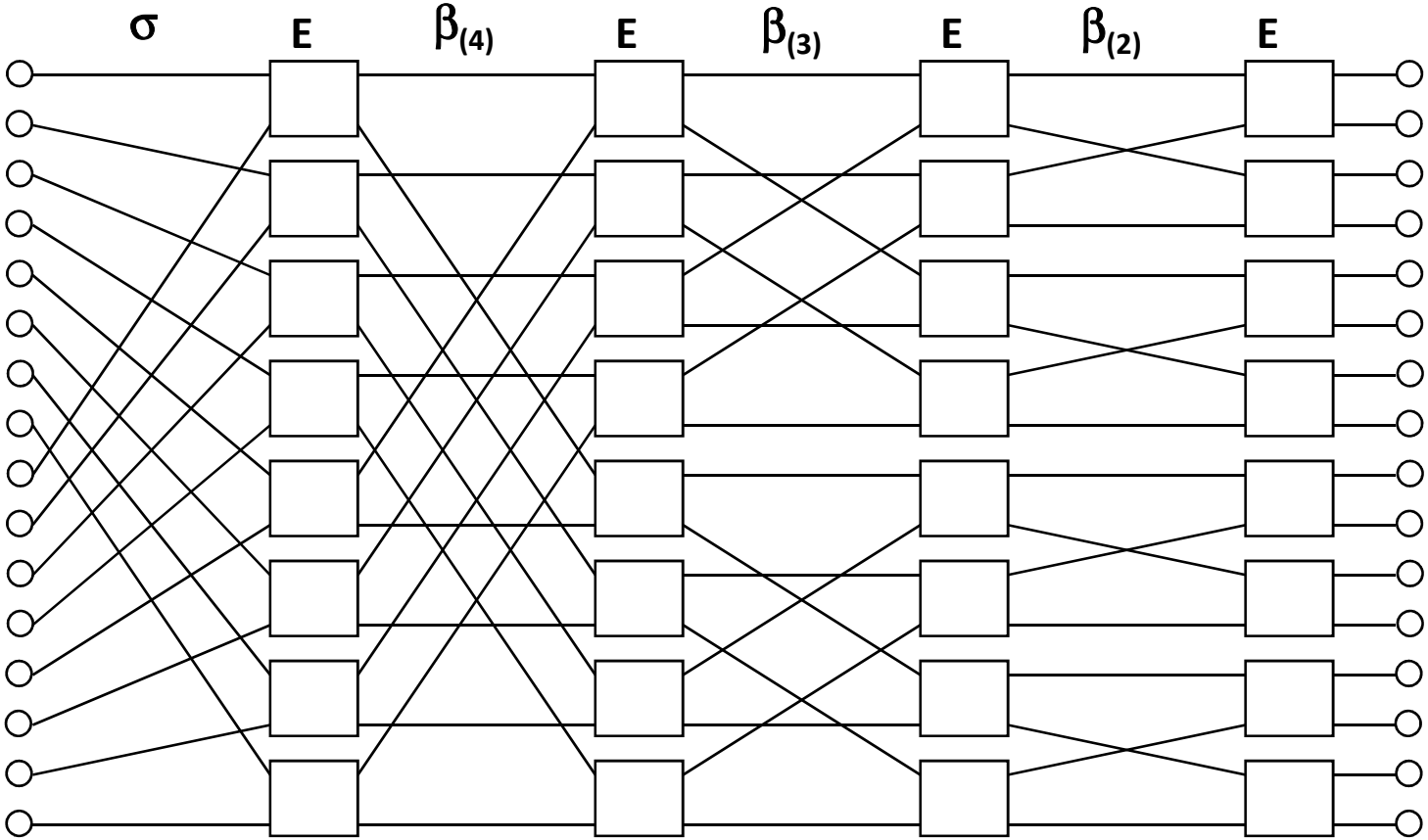
Samosměrovací algoritmus:

- Přepínač v k-tém stupni sítě čte k-tý bit směrovacího návěští (počínaje LSB směrem k MSB) a pokud se tento rovná:
  - 0 pak jdi na horní výstup
  - 1 pak jdi na dolní výstup



Inverzní nepřímá binární n-kubická síť

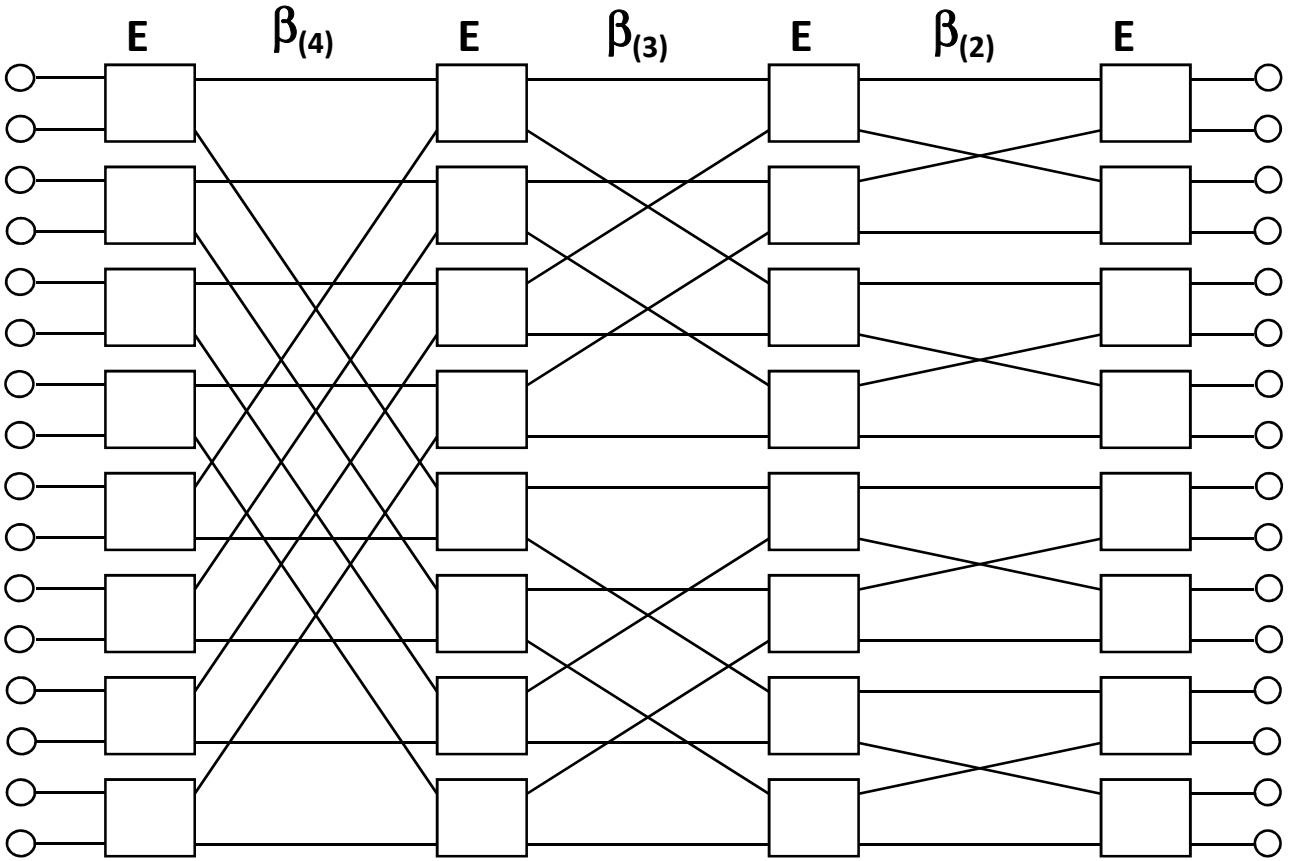
$$C_N^{-1} = (E\beta_{(2)}E\beta_{(3)}E\beta_{(4)} \ldots E\beta_{(n)}E\sigma^{-1})^{-1} = \sigma\, E\beta_{(n)} \ldots E\beta_{(4)}\, E\beta_{(3)}\, E\beta_{(2)}\, E$$



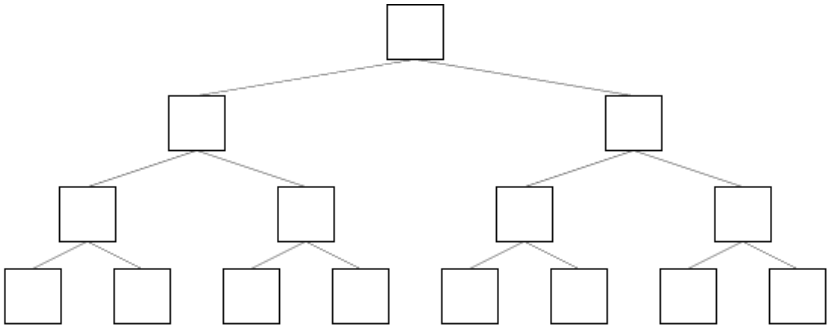
Butterfly network

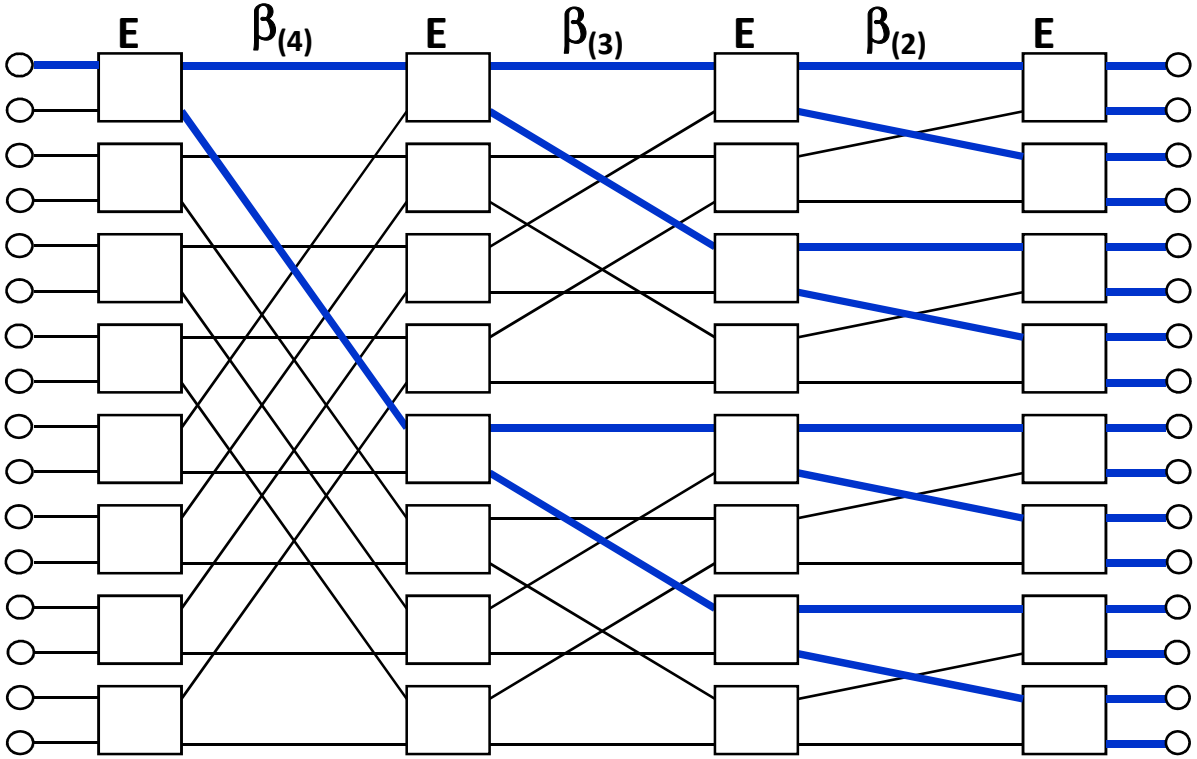
$$C_N^{-1} = (E\beta_{(2)}E\beta_{(3)}E\beta_{(4)} \ldots E\beta_{(n)}E\sigma^{-1})^{-1} = \sigma E\beta_{(n)} \ldots E\beta_{(4)}\, E\beta_{(3)}\, E\beta_{(2)}\, E$$

$$\mathbf{B = E\beta_{(n)} \ldots E\beta_{(4)}\, E\beta_{(3)}\, E\beta_{(2)}\, E}$$



Butterfly network

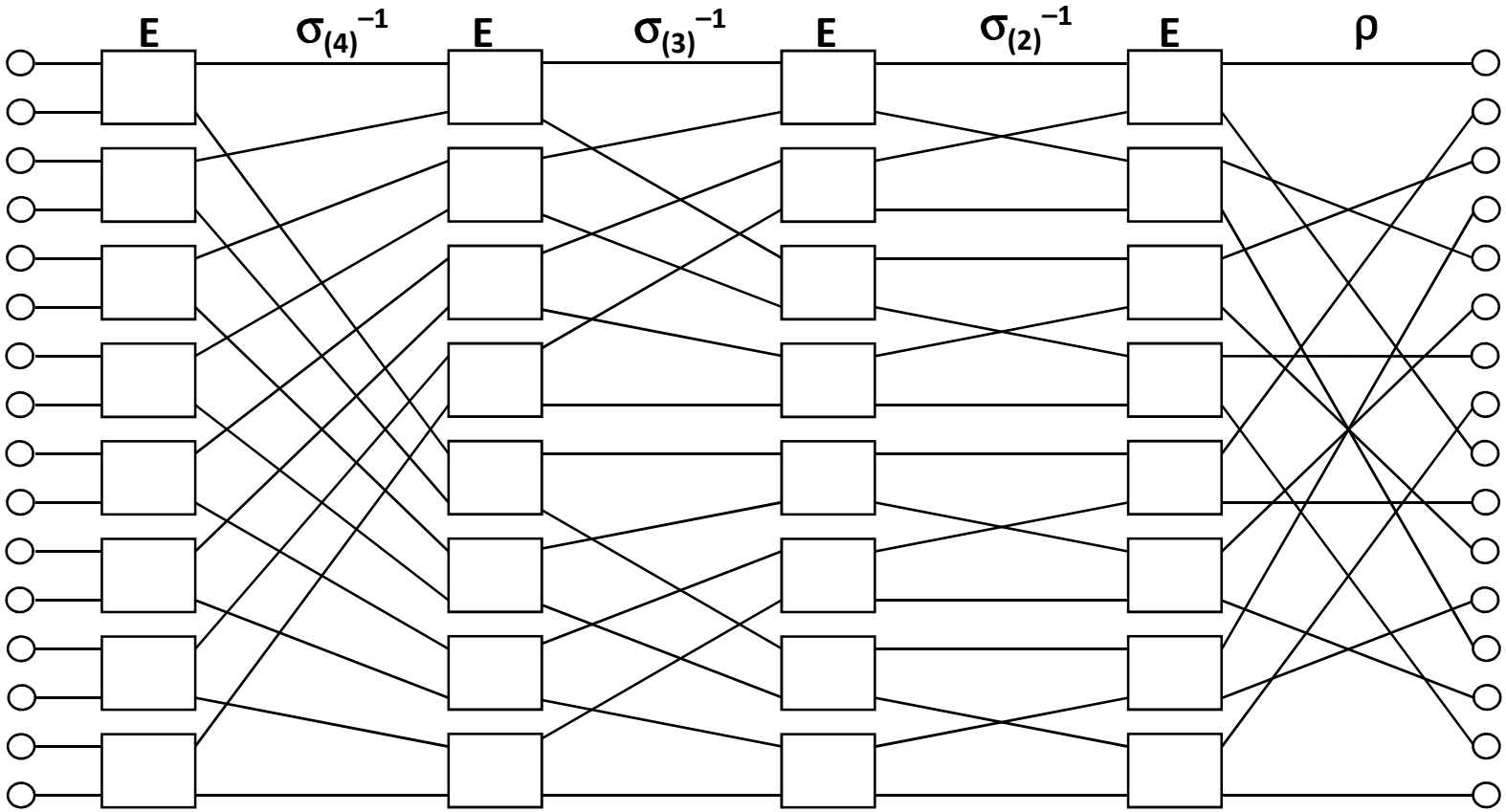




R-síť (R-network)

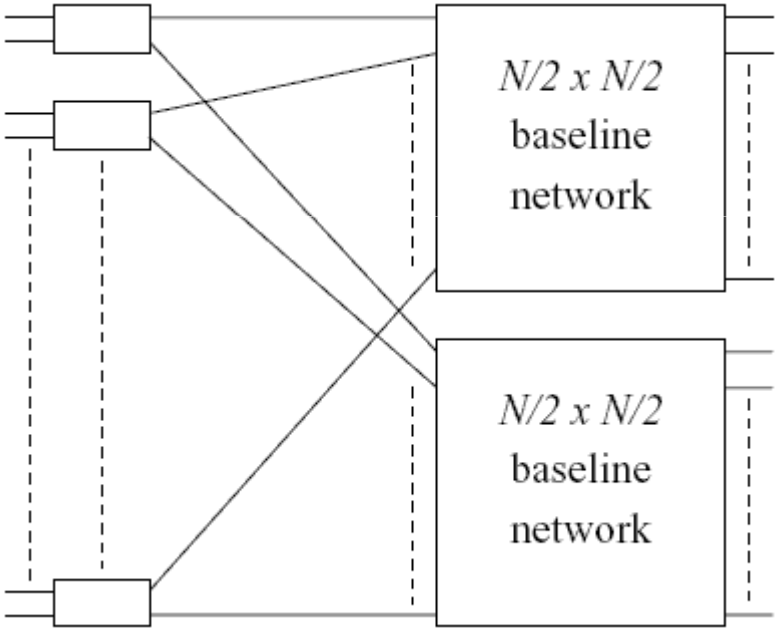
$$R_N = E\sigma_{(n)}^{-1} E\sigma_{(n-1)}^{-1} \ldots E\sigma_{(2)}^{-1} E\rho$$

Napr. pro N=16, n=4 $\rightarrow$ $R_{16} = E\sigma_{(4)}^{-1} E\sigma_{(3)}^{-1} E\sigma_{(2)}^{-1} E\rho$



Baseline síť (Baseline network)

- Topologie baseline sítě je definována rekurzivní způsobem podle následujícího obrázku:



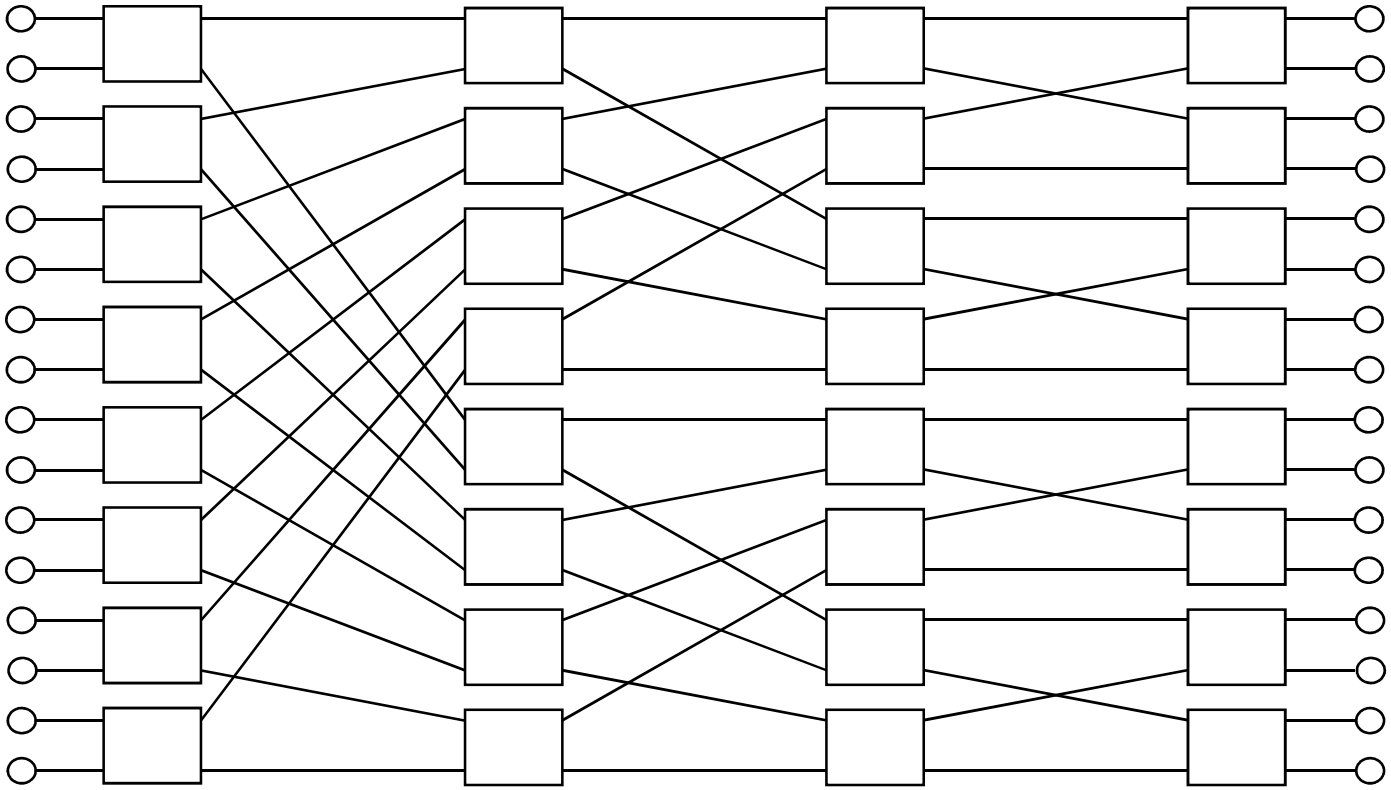
- Spojovací funkce mezi dvěma stupni je reverzní míchání **(σ^{-1}).**

Baseline síť (Baseline network)

- Rekurzivní proces se ukončí pokud dosáhneme elementy 2x2. Rozepsáním rekurze můžeme získat zápis baseline sítě:

$$\mathbf{Baseline_N = E\sigma_{(n)}^{-1} E\sigma_{(n-1)}^{-1} \ldots E\sigma_{(2)}^{-1} E}, \quad \text{kde } \mathbf{n=\log_2 N}$$

- Např. pro N = 16, n = 4:

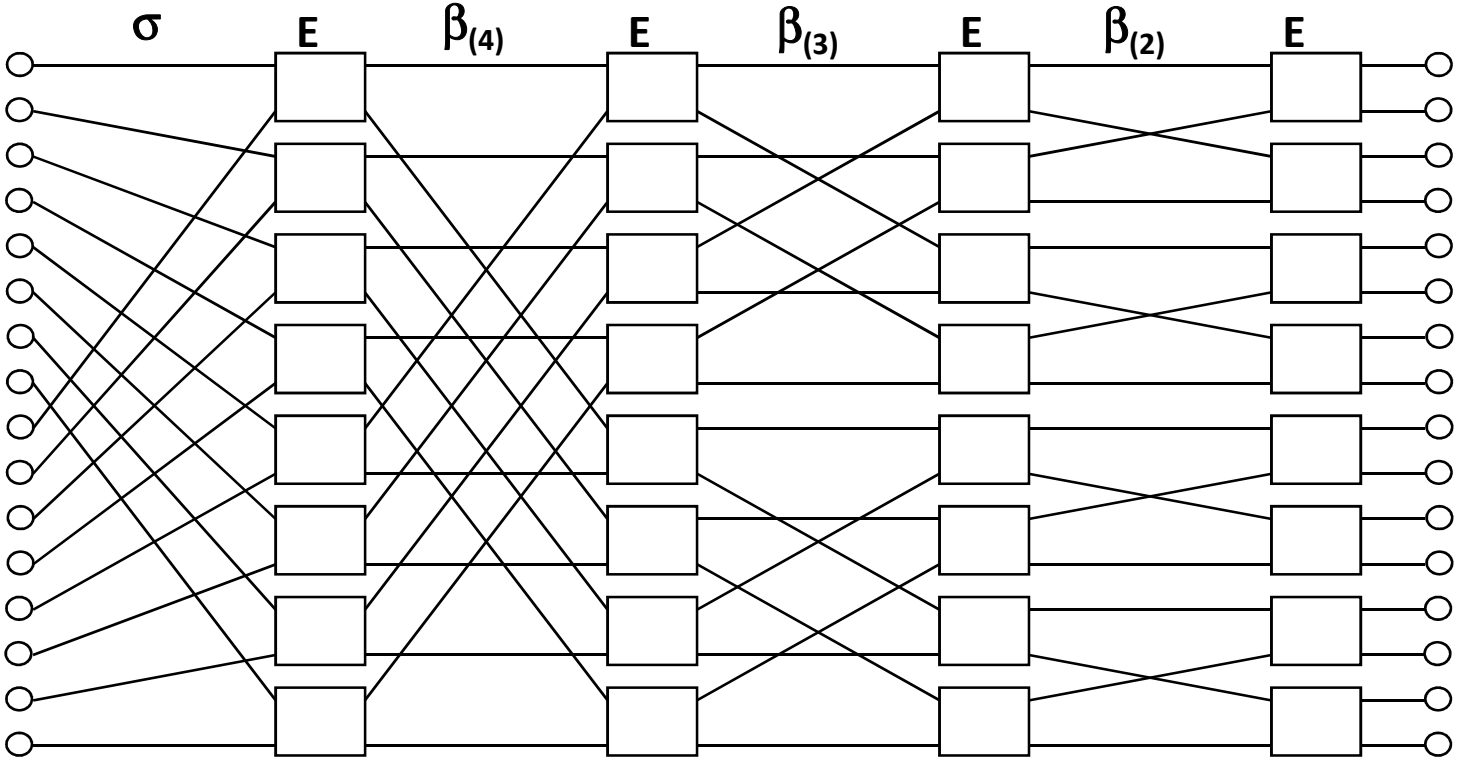


Banyan network

- Banyan síť, někdy také nazývaná generalizovaná kubická síť (generalized cube network) je síť definována předpisem:

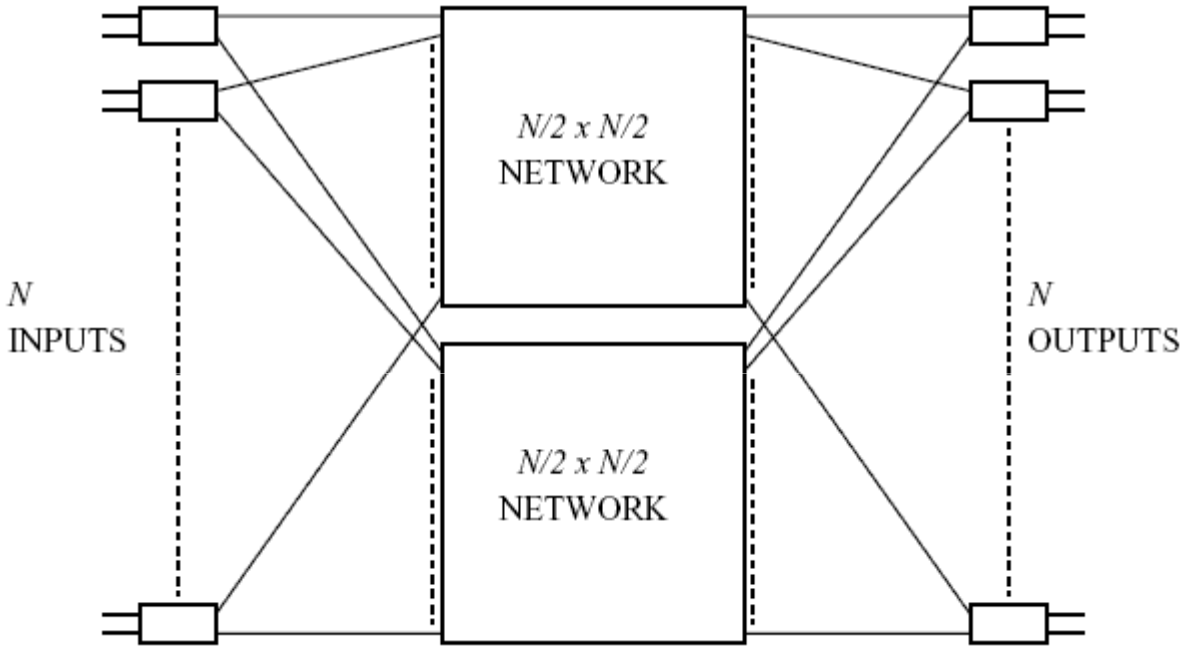
  $\mathbf{Banyan_N = \sigma\, E\beta_{(n)} \ldots E\beta_{(4)}\, E\beta_{(3)}\, E\beta_{(2)}\, E,}$ kde n = log2N

  a tedy se jedná o inverzní nepřímou binární n-kubickou síť.



Benešova síť (Beneš network)

- Rekurzivní definice Benešovy sítě:

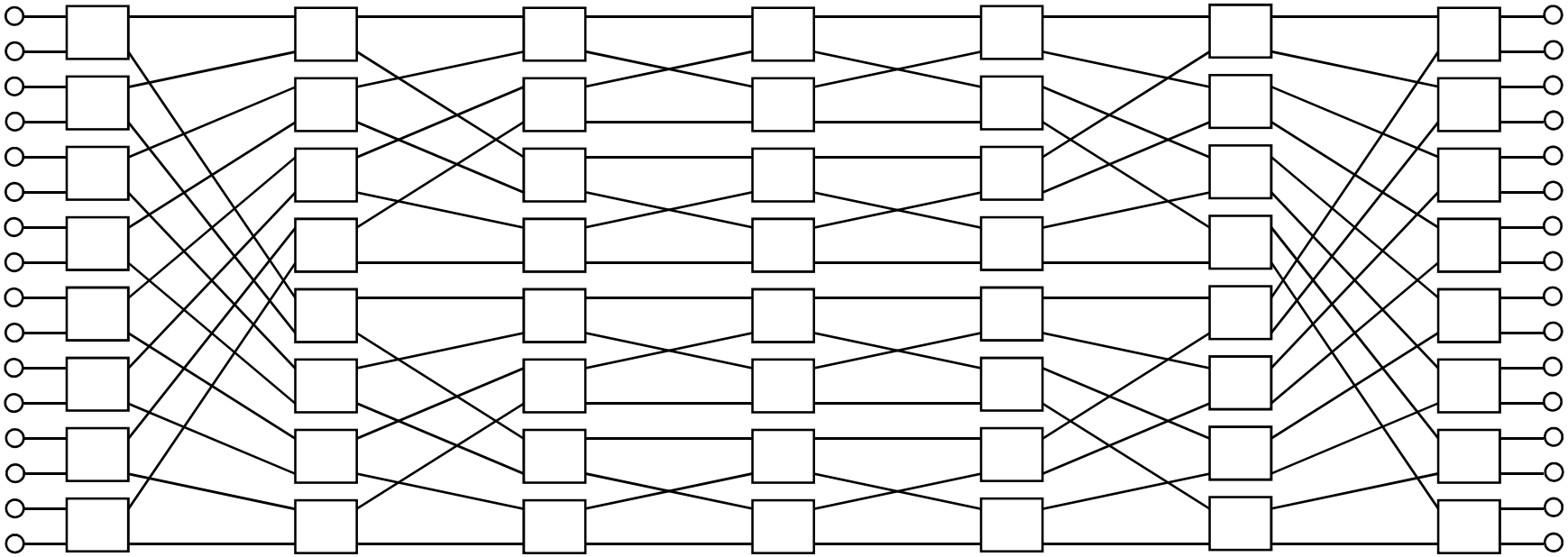


- Když je rekurzivní proces ukončen síť má $2\log_2 N - 1$ stupňů a v každém stupni N/2 spojovacích elementů. Benešova síť patří mezi **rekonfigurovatelné sítě bez blokování** a proto vyžaduje algoritmus pro rekonfiguračný proces - centrální řízení. Existují také samosměrovací algoritmy částečně potlačující blokování v síti.

Benešova síť (Beneš network)

$$\text{Beneš}_N = E\sigma_{(k)}^{-1} E\sigma_{(k-1)}^{-1} E \ldots \sigma_{(2)}^{-1} \; E \; \sigma_{(2)} \ldots E\sigma_{(k-1)} E\sigma_{(k)} E,$$

- Např. pro N = 16, stupňů: n = 2log2N-1 = 7:



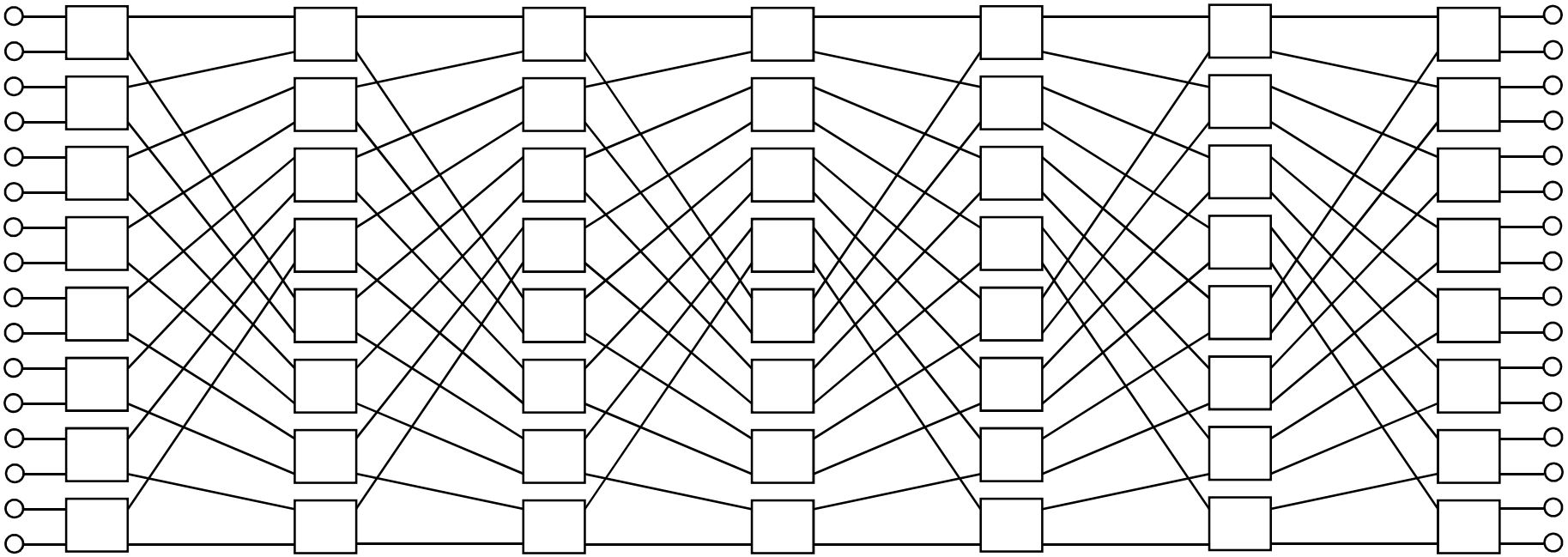
- Na Benešovu síť se také můžeme dívat jako na zařazení dvou baseline sítí za sebou (baseline síť a reverzní baseline síť). Proto bývají Benešově sítě někdy také nazývané "sériové baseline sítě".

Benešova síť (Beneš network)

Alternativní definice:

$Beneš_N = E\sigma^{-1} E\sigma^{-1} E \ldots \sigma^{-1} E \sigma \ldots E \sigma E \sigma E = (E\sigma^{-1})^{k-1} E (\sigma E)^{k-1}$,

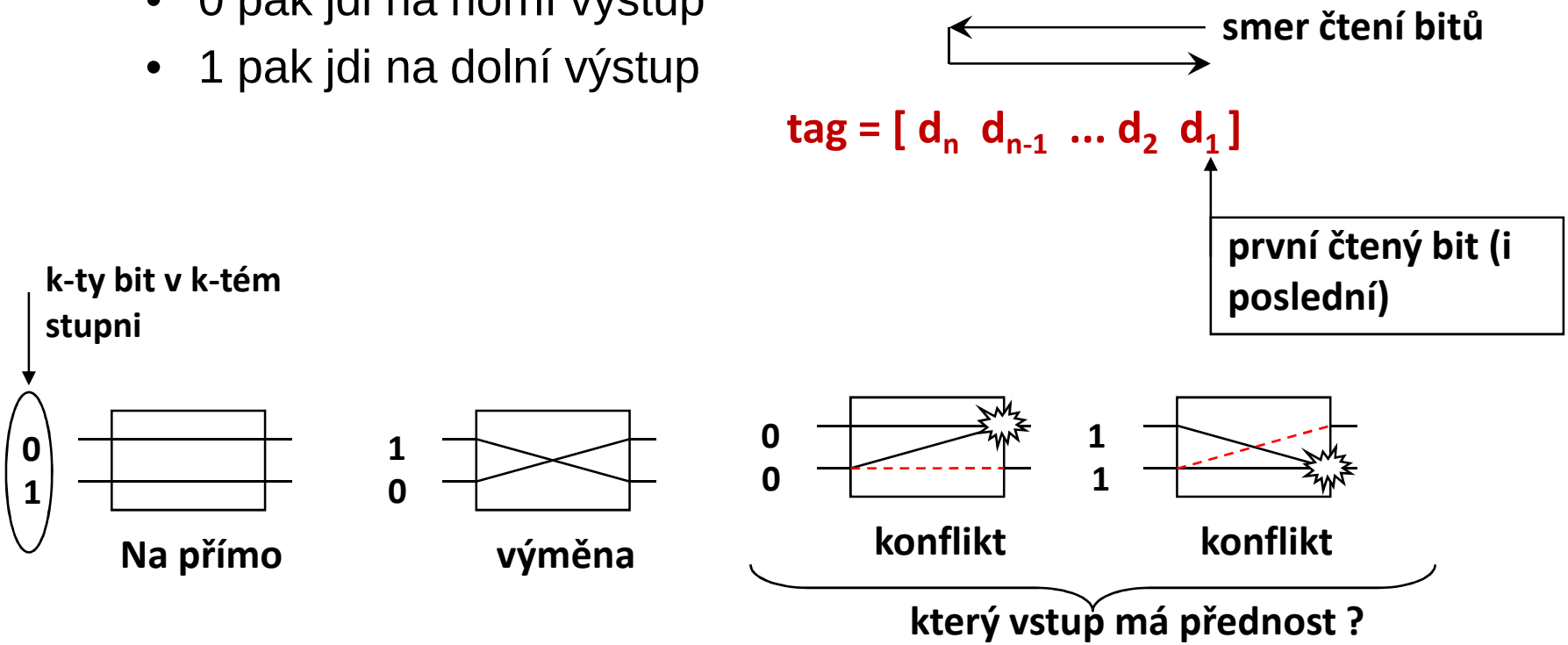
- **kde $k=\log_2 N$**



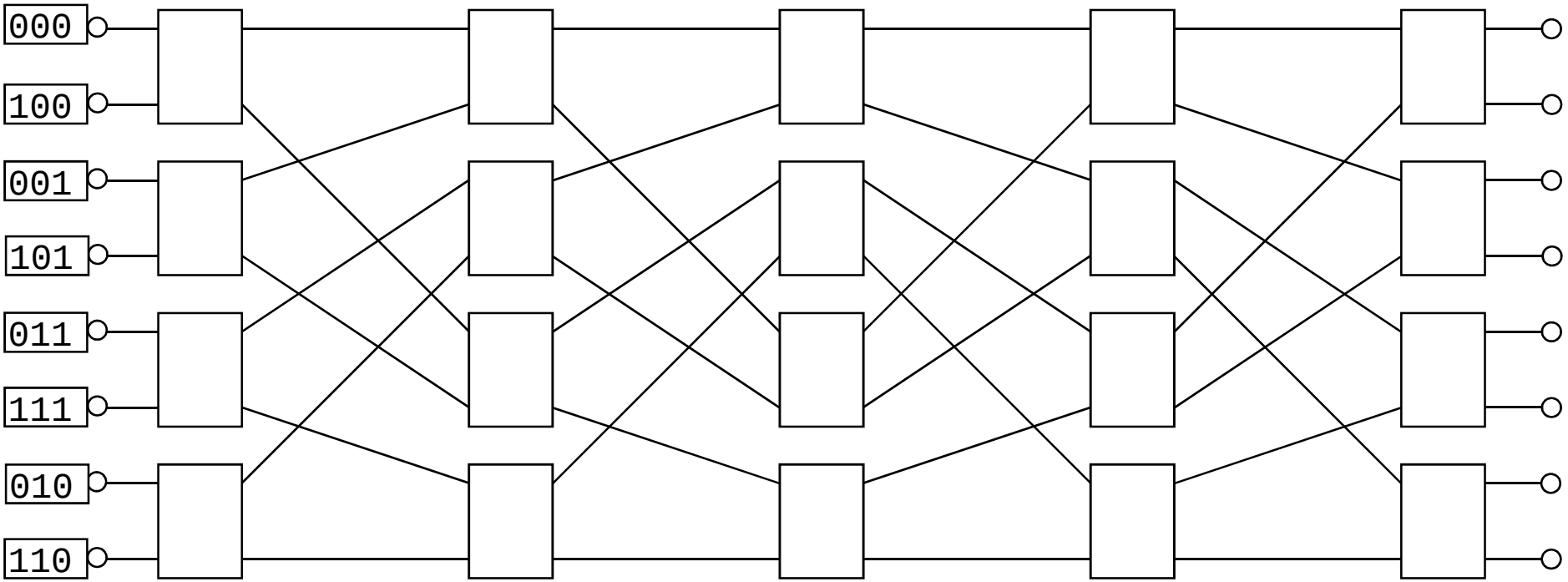
Benešova síť (Beneš network)

Samosměrovací algoritmus:

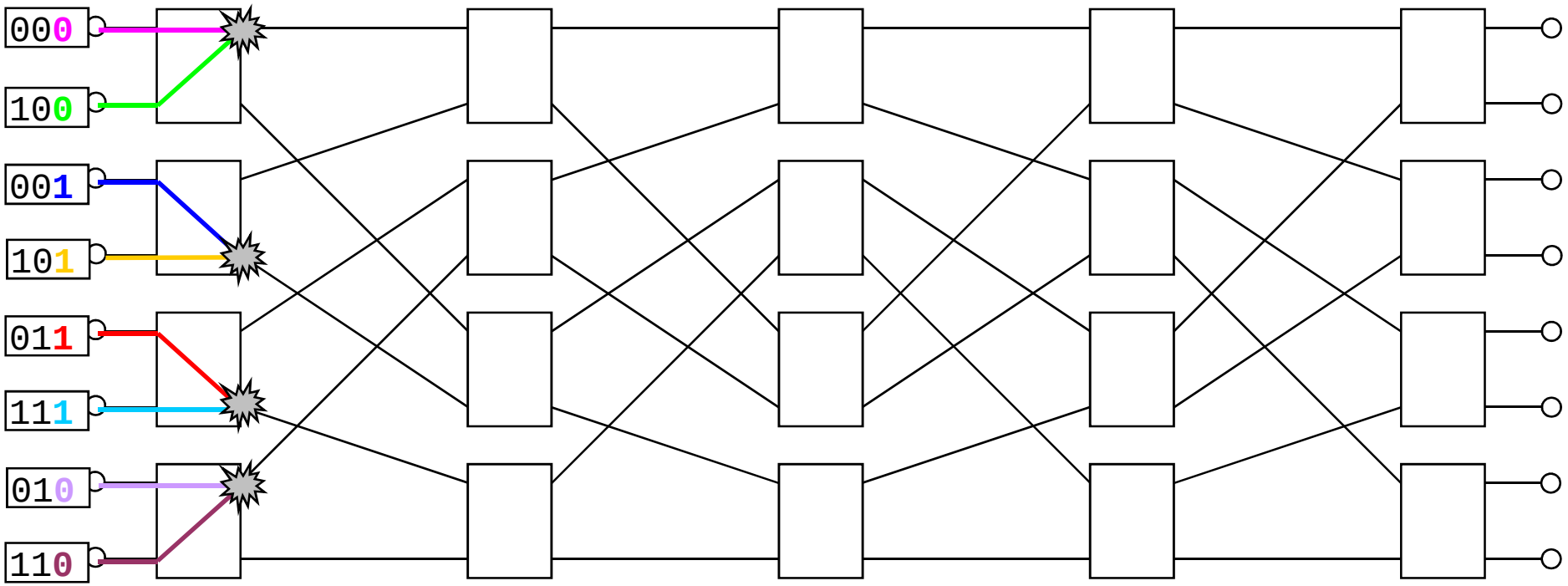
- Přepínač v i-tém, resp. j-tém stupni ($1 \le i \le k$, resp. $k < j \le 2k-1$, kde $k = \log_2 N$) sítě čte i-tý bit, resp. (2k-j) tý bit směrovacího návěští (počínaje LSB směrem k MSB) a pokud se tento rovná:
  - 0 pak jdi na horní výstup
  - 1 pak jdi na dolní výstup



Benešova síť (Beneš network)



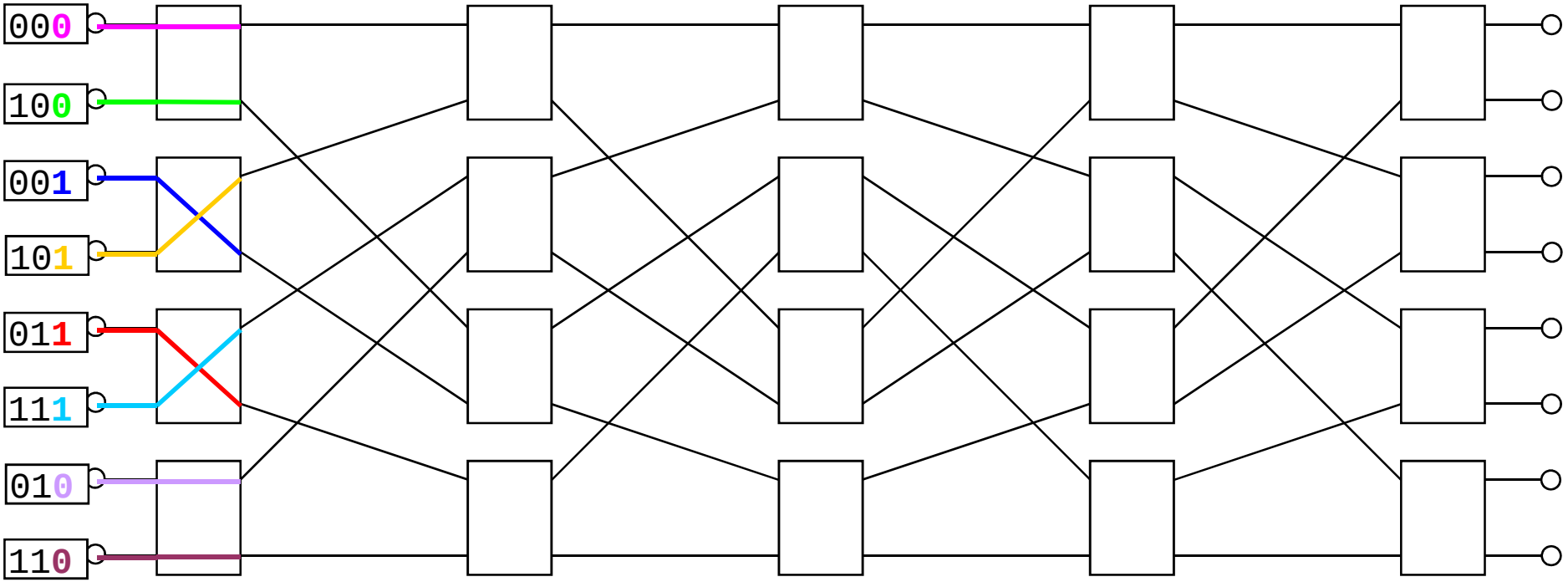
Benešova síť (Beneš network)



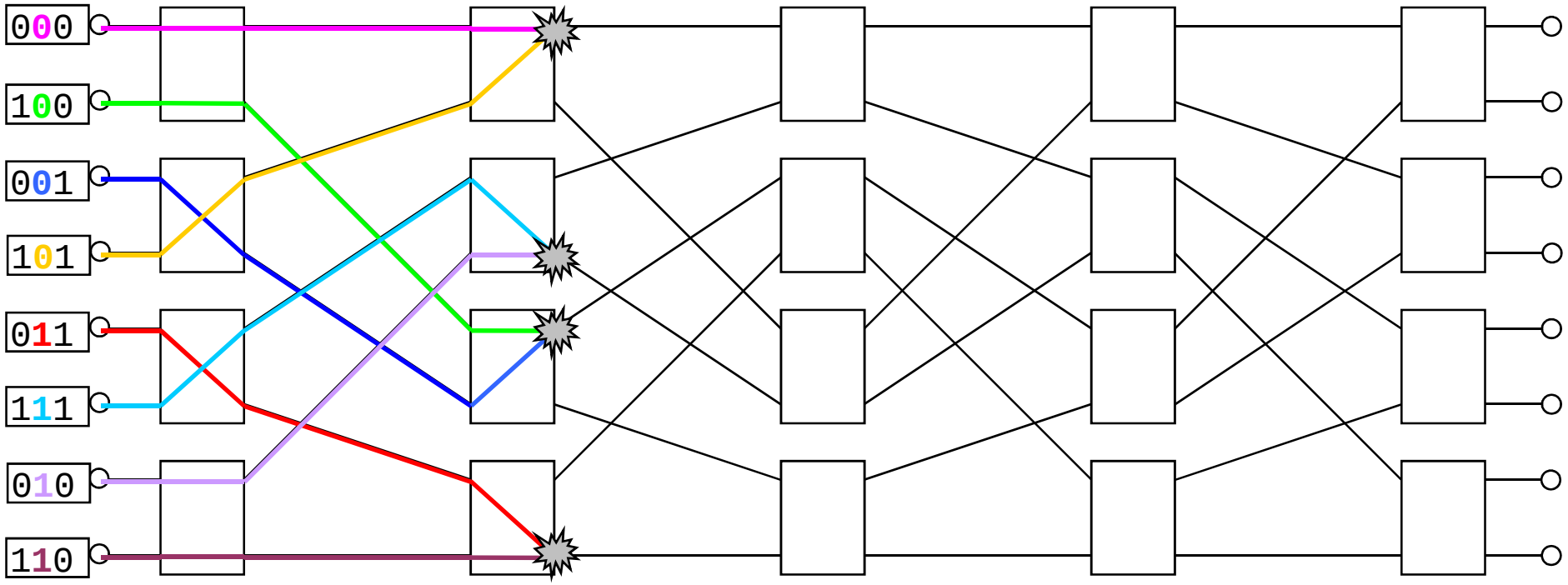
Co teď ???

Napr. Prednost má ten, kterého hodnota tagu (adresa určení) je menší...

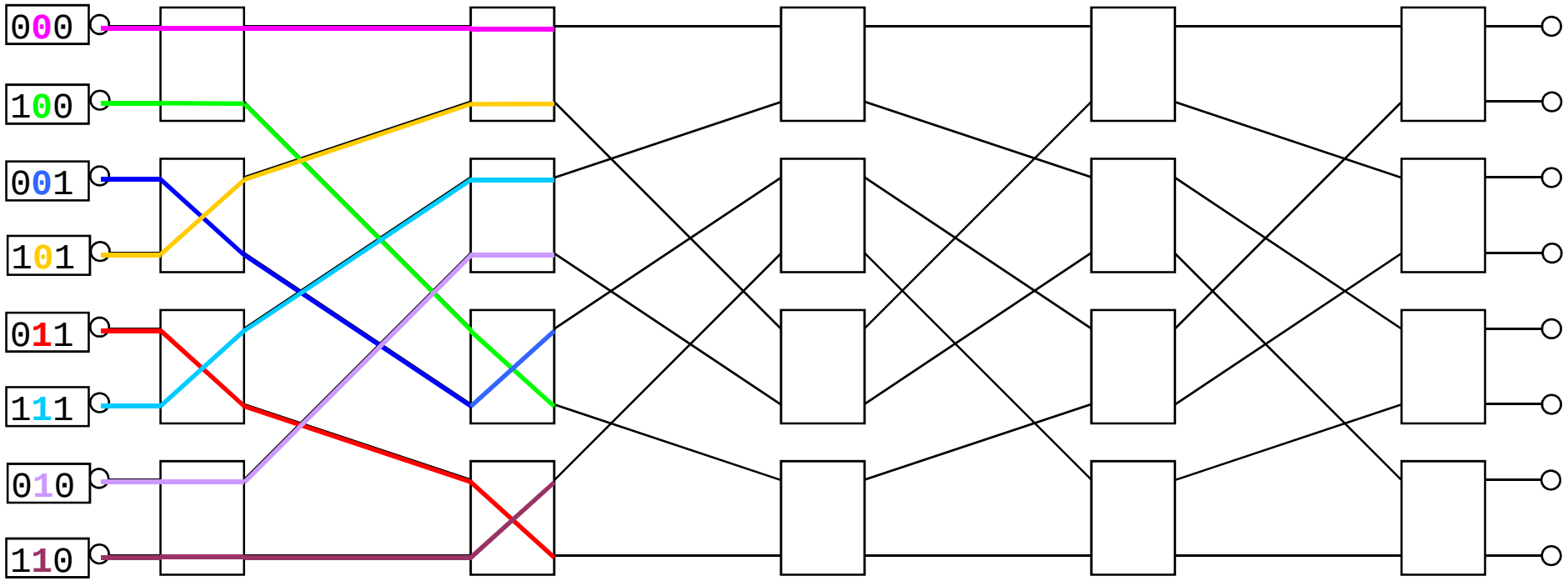
Benešova síť (Beneš network)



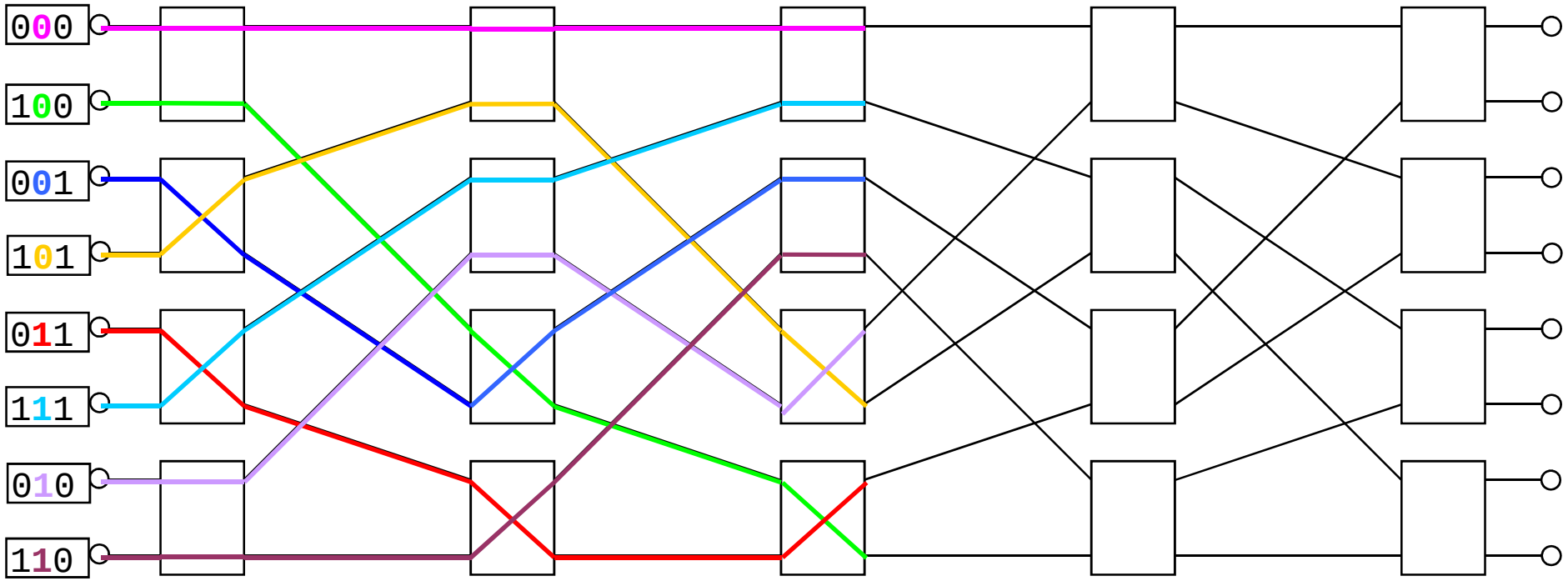
Benešova síť (Beneš network)



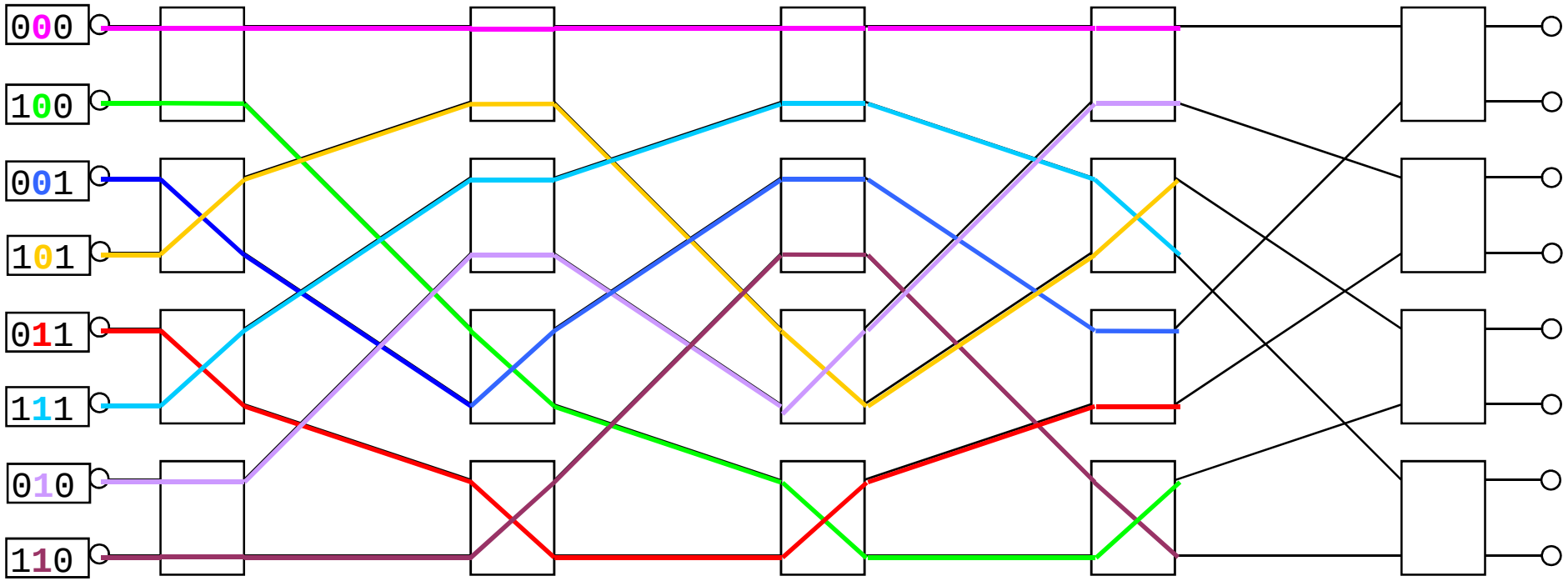
Benešova síť (Beneš network)



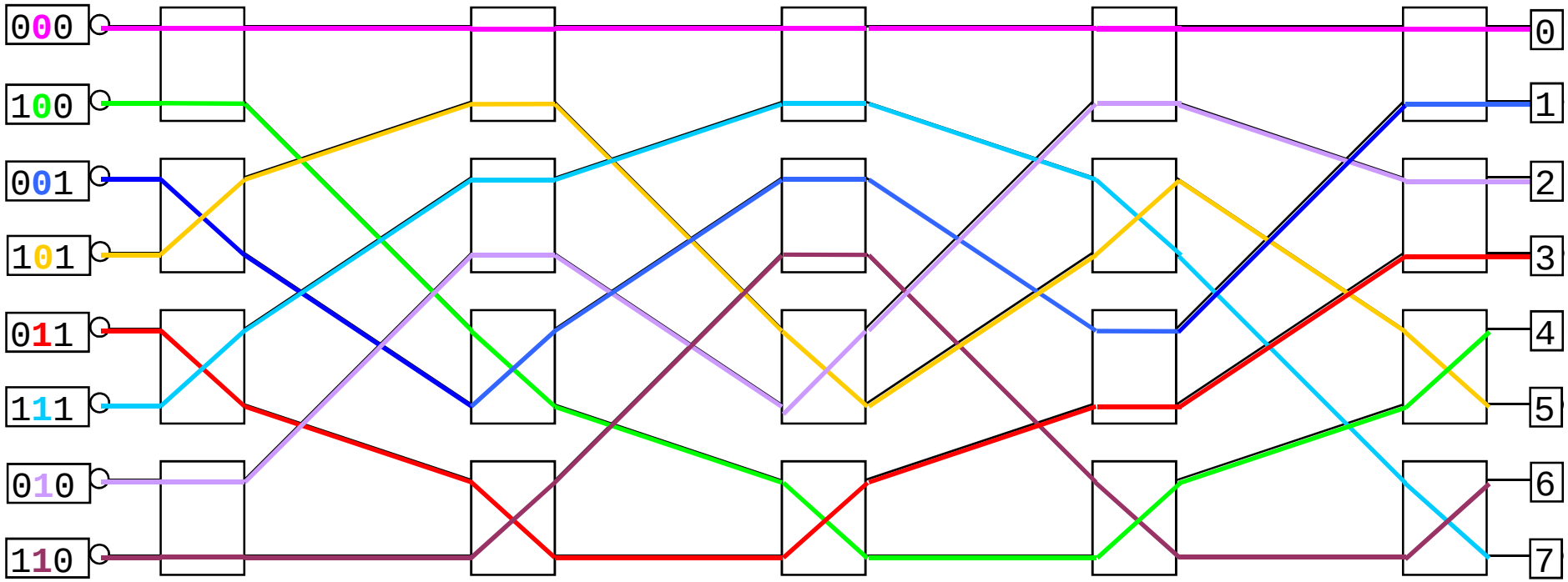
Benešova síť (Beneš network)



Benešova síť (Beneš network)

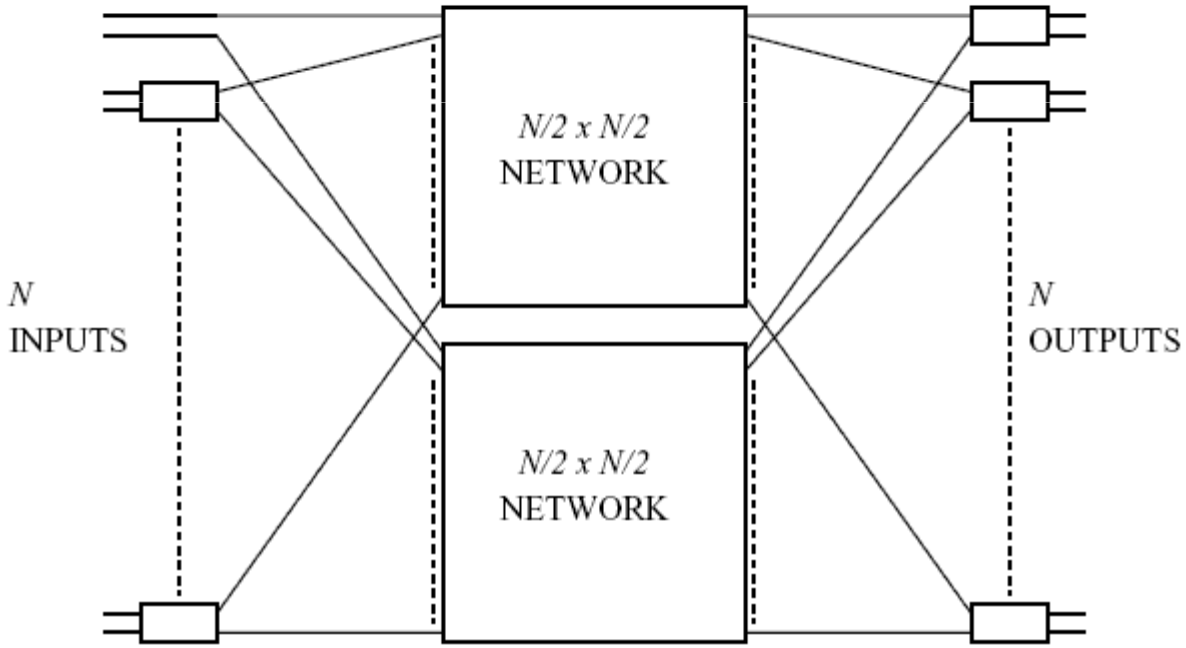


Benešova síť (Beneš network)



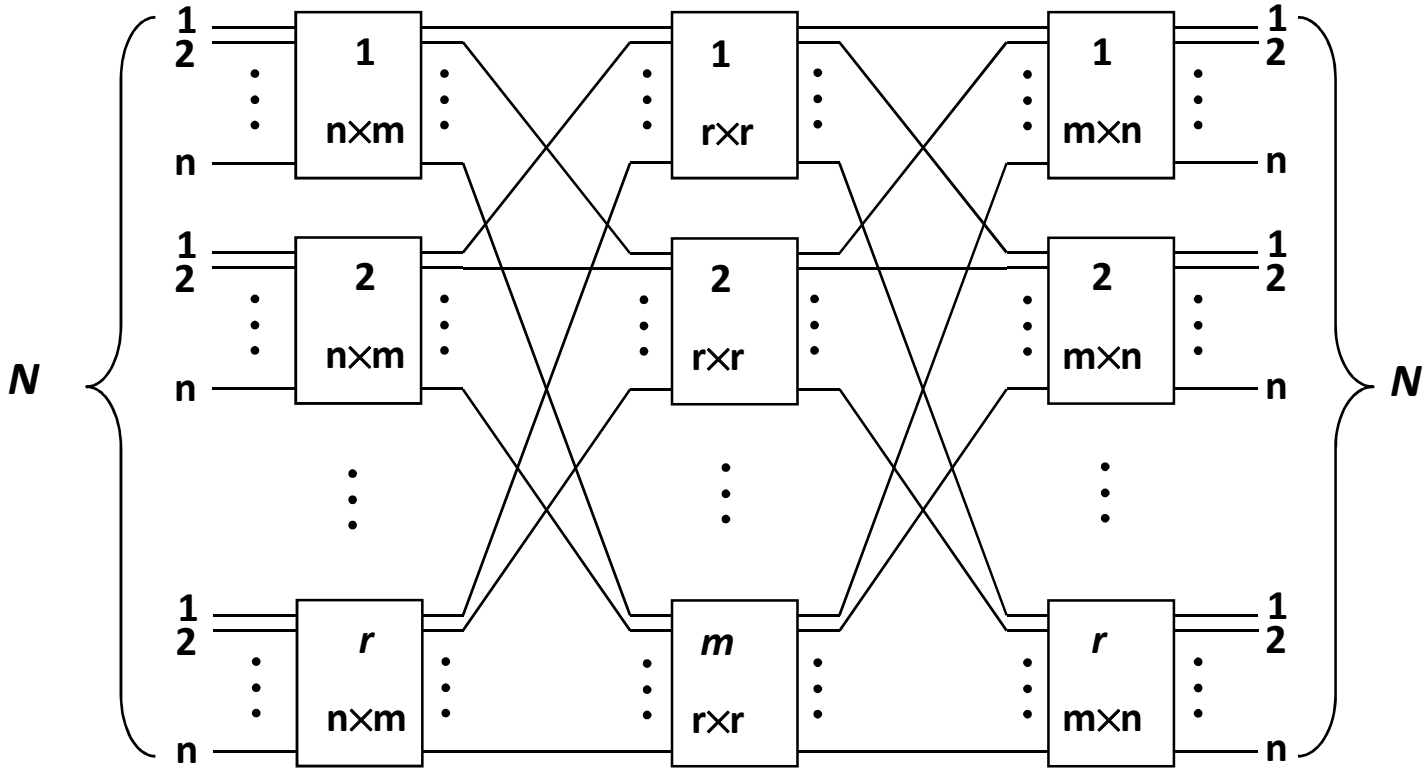
Waksmanova síť (Waksman network)

- Použitím modifikované verze Slepian-Duguidovho teorému může být z Benešovy sítě vynecháním jednoho vstupního nebo výstupního elementu vytvořena tzv. Waksmanova síť. Waksmanova síť používá méně přepínacích elementů než Benešova síť.



Closova síť

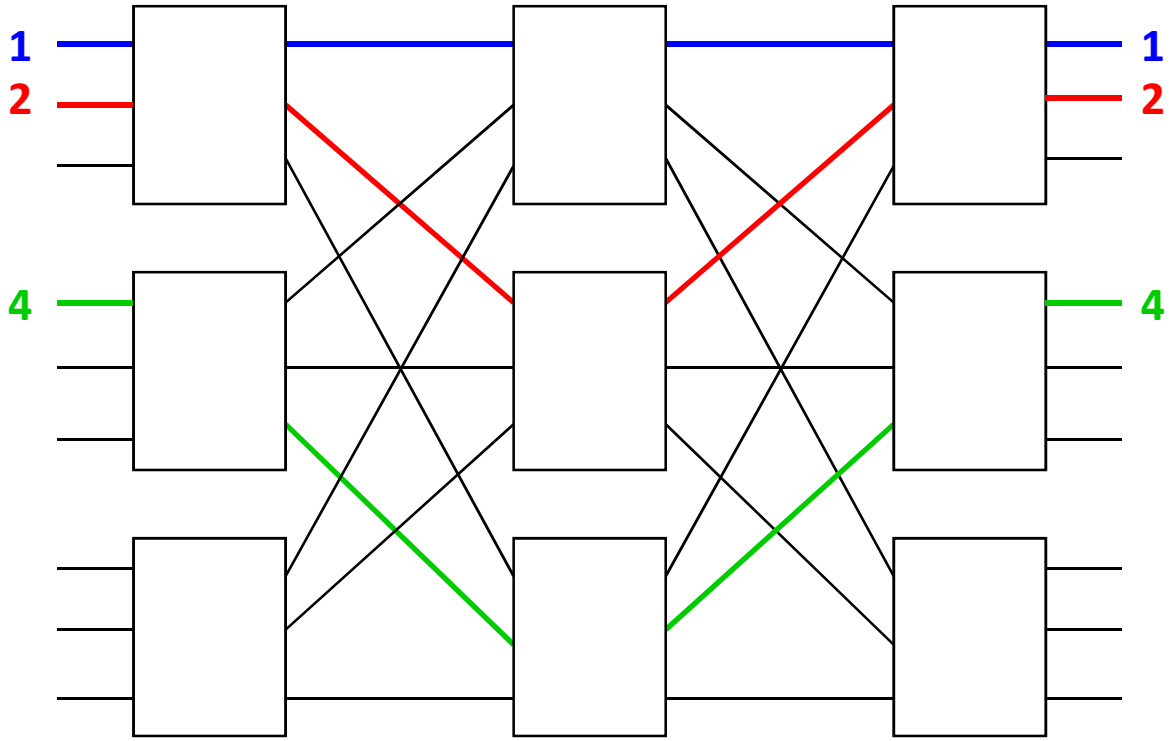
- Symetrická 3-stupňová Closova síť s N vstupními a N výstupními porty má r přepínacích modulů velikosti nxm na prvním stupni, m přepínacích modulů velikosti rxr na středním stupni ar přepínacích modulů velikosti mxn na třetím (výstupním) stupni. Taková 3 stupňová síť je označována jako C (m, n, r).



Closova síť

Je Closova síť s m = n neblokující?

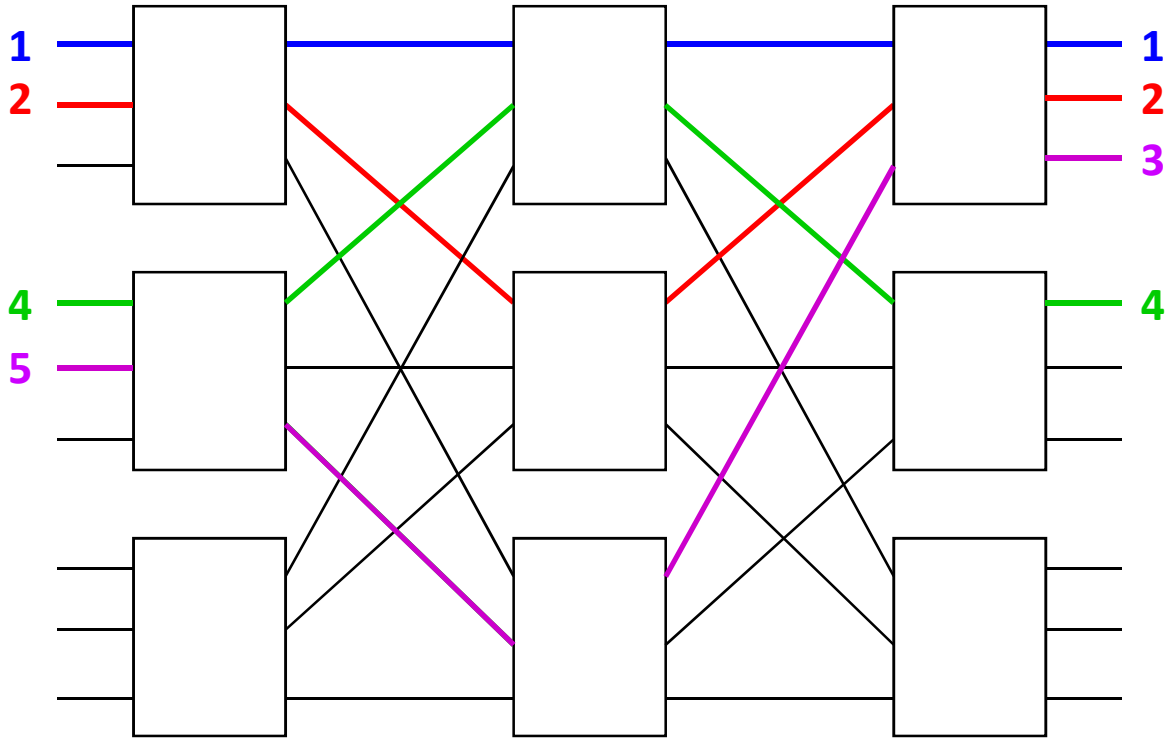
- **Příklad propojení v síti: (1,1), (2,2), (4,4), (5,3), ...**



Closova síť

Je Closova síť s m = n neblokující?

- **Příklad propojení v síti: (1,1), (2,2), (4,4), (5,3), ...**



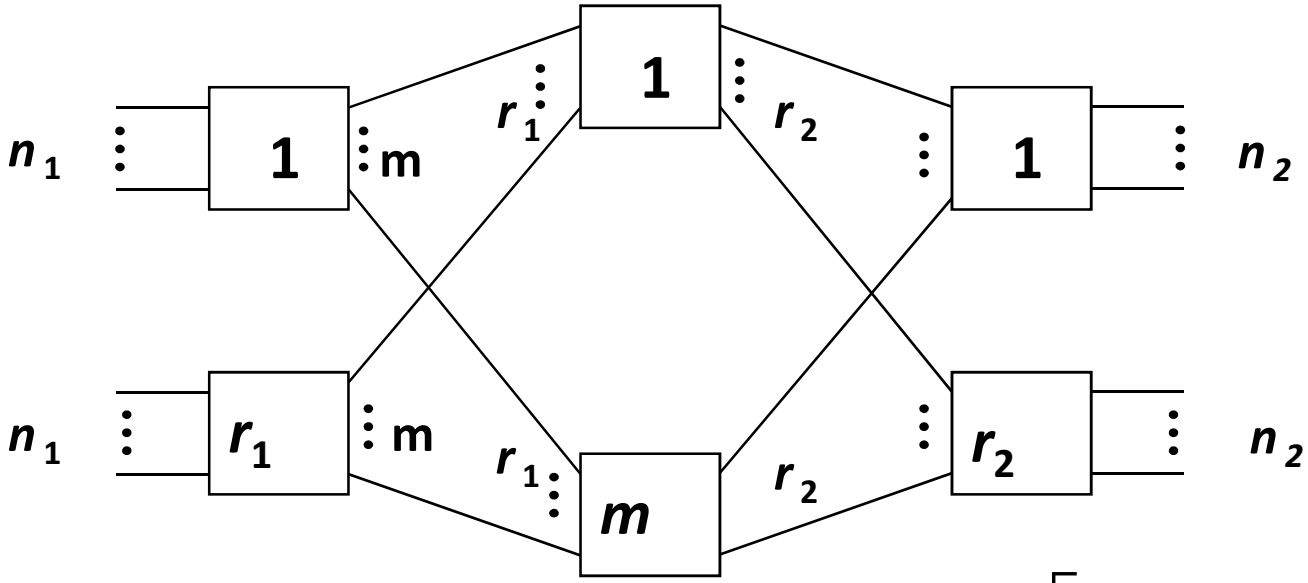
Řešením problému je přeuspořádat již existující spojení v síti

Closova síť

- Propojovací schopnost Closovej sítě je závislá na parametrech n, ra m. Pro dané n, r a měnící se m je mnoho možností propojování. Neblokující operace 3-stupňové sítě můžeme dosáhnout více než jedním způsobem. Rozlišujeme 3 módy neblokujících Closových sítí:
  - **striktně (přísně) neblokující síť** (Strictly Nonblocking - SNB), podmínka: **$m \geq 2n-1$**,
  - **široce neblokující síť** (Wide-Sense Nonblocking - WSN),
  - **přeuspořádatelná neblokující síť** (Rearrangeably Nonblocking - RNB), podmínka: **$m \geq n$**

- Obecně třístupňová Closova síť C (n1, r1, m, n2, r2) je 3-stupňová síť, jejíž první stupeň tvoří r1 přepínačů velikosti n1xm, třetí stupeň má r2 přepínačů dimenze mxn2, a střední stupeň tvoří m přepínačů velikosti r1xr2. Pokud platí n1 = n2, r1 = r2 pak hovoříme o symetrické Closovej 3-stupňové síti. C (n, r, m, n, r) značíme C (m, n, r).

Closova síť



- Pro Closovu síť C (n_1, r_1, m, n_2, r_2) můžeme požadovaný stav sítě - zadaný permutací P přepsat do matice o r_1 řádcích a r_2 sloupcích

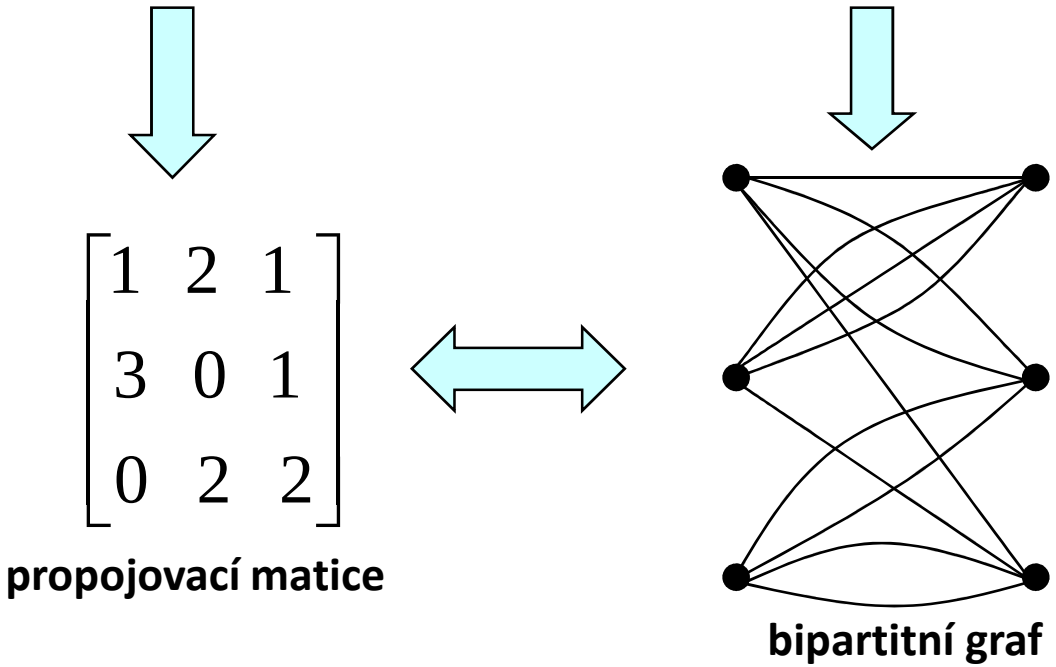
$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,r2} \\ \vdots & & \vdots \\ a_{r1,1} & \cdots & a_{r1,r2} \end{bmatrix}$$

Pro matici A platí: $\sum_{i=1}^{r_1} a_{ij} = n_2 \quad pro\ \forall j = konšt.$ $\sum_{j=1}^{r_2} a_{ij} = n_1 \quad pro\ \forall i = konšt.$

Closova síť

- Mějme permutaci P a síť typu C (4,4,3):

$$P = \left(\begin{array}{cccc:cccc:cccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 3 & 7 & 5 & 11 & 4 & 9 & 1 & 2 & 10 & 12 & 6 & 8 \end{array}\right)$$



Closova síť

Podle zvolené reprezentace vyplývají algoritmy:

- rozklad permutační matice
- rozklad bipartitního grafu

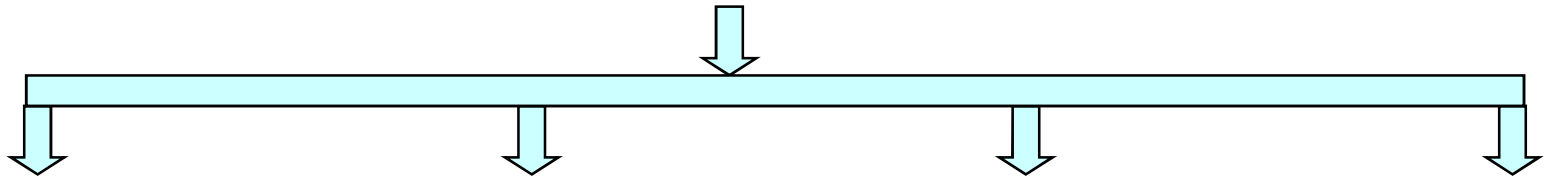
Rozklad propojovací matice:

- Rozkladem propojovací matice A na dílčí matice B_p , $p \in \{1, 2, \ldots, q\}$, také že platí $A = B_1 + B_2 + \ldots + B_p + \ldots + B_q$ a zároveň pre $\forall B_p$ platí
  - $b_{ij} \in \{0, 1\}$
  - $\sum_{i=1}^{r_1} b_{ij} = 1$ príp. $\sum_{i=1}^{r_1} b_{ij} \leq 1$
  - $\sum_{j=1}^{r_2} b_{ij} = 1$ príp. $\sum_{j=1}^{r_2} b_{ij} \leq 1$
- získáme konfiguraci přepínačů v druhém stupni sítě. Pro konkrétní rozklad je q! různých nastavení přepínačů v druhém stupni.

Closova síť

- V našem případě:

$$\begin{bmatrix} 1 & 2 & 1 \\ 3 & 0 & 1 \\ 0 & 2 & 2 \end{bmatrix}$$

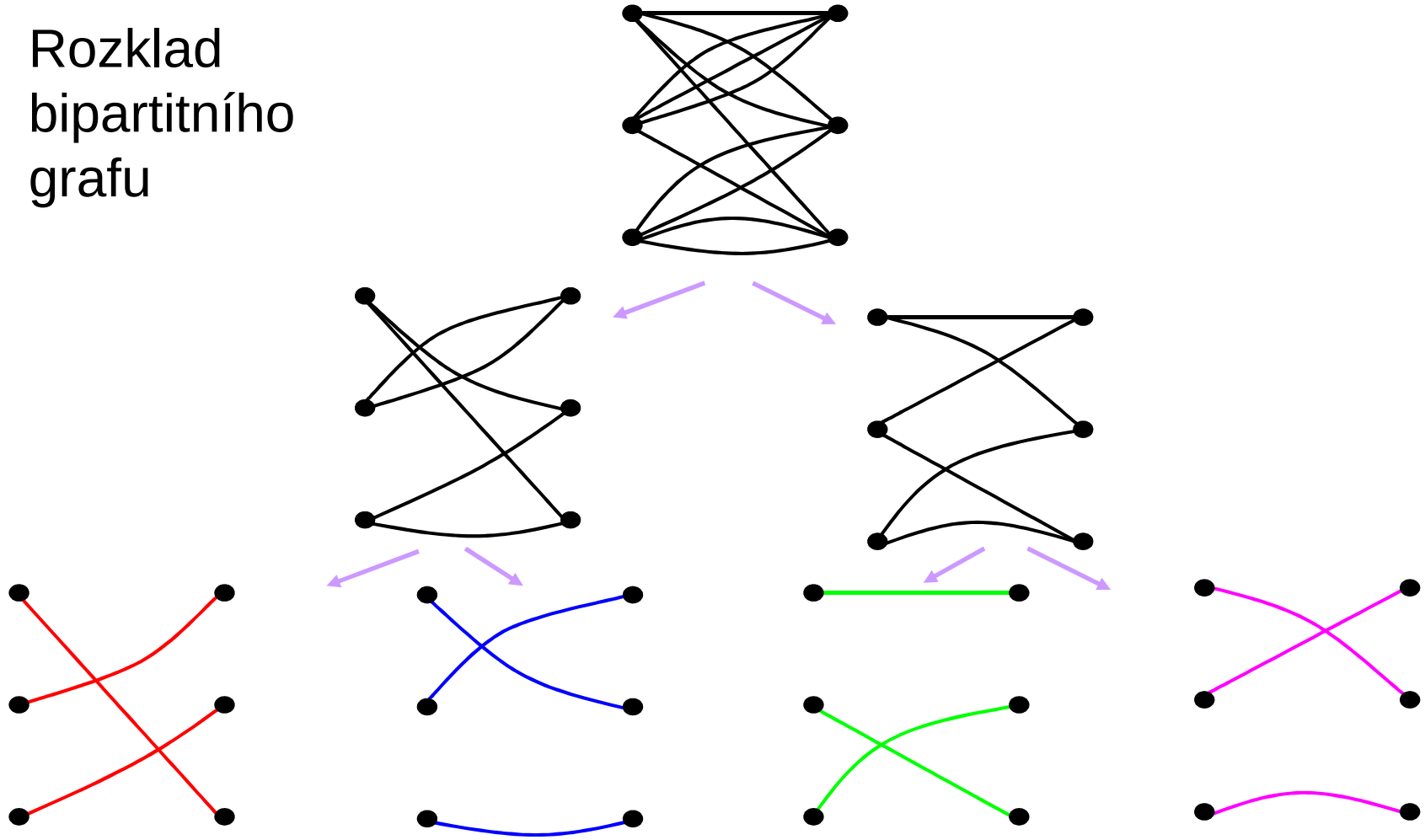


$$S_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad S_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad S_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A = S1 + S2 + S3 + S4

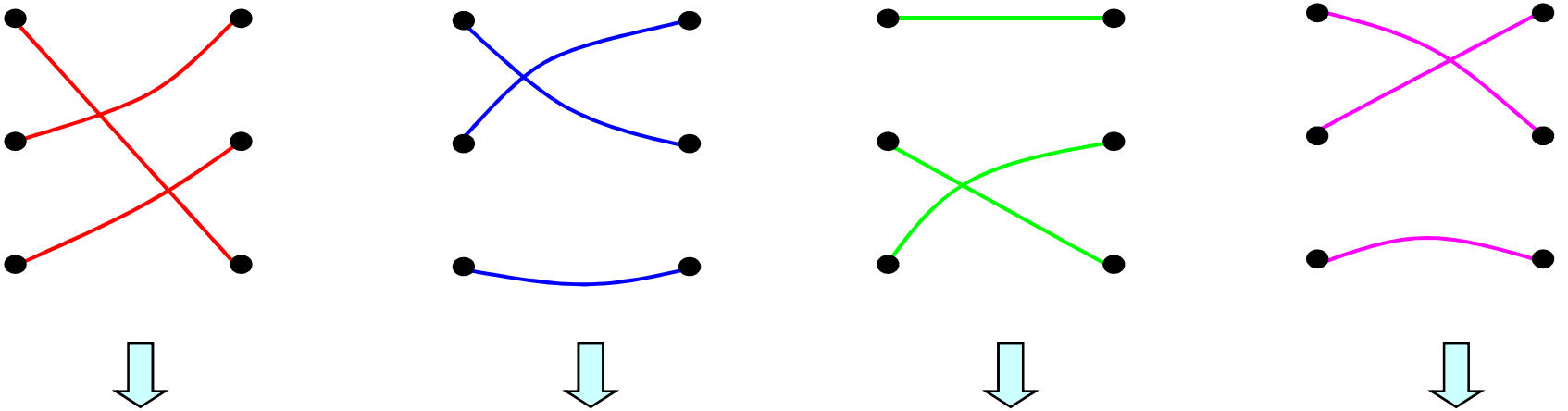
Closova síť

- Rozklad bipartitního grafu



Closova síť

- Rozklad bipartitního grafu



$$S_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad S_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad S_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

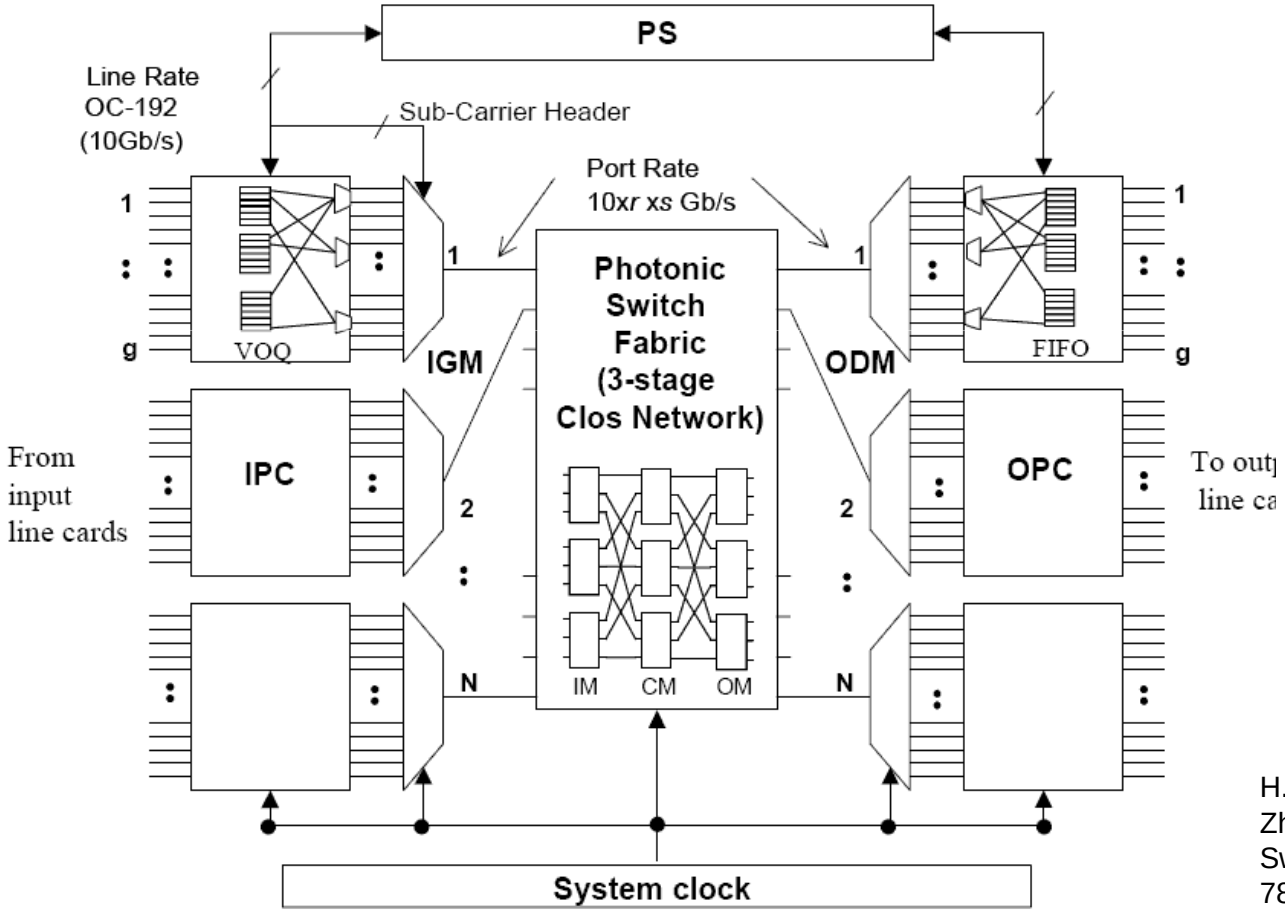
Closova síť

- Matice vzniklé rozkladem (ať už grafu nebo propojovací matice) umožňují **přímo** nastavit přepínače ve středním stupni sítě.
- Pokud známe nastavení přepínačů středního stupně není náročné nastavit přepínače v celé síti.

- Pozn: Benešova síť je speciální případ Closovej sítě. Proto pokud chceme zajistit bezkonfliknosť Benešovy sítě můžeme využít výše uvedené algoritmy pro Closovu síť.

Closova síť

- Například architektura peta bytového paketového optického přepínače P3S dimenze 6400x6400 je založena na Closovej 3-stupňové síti dosahuje celkově 1.024 petabit / s (160 Gbit / s na každý port)

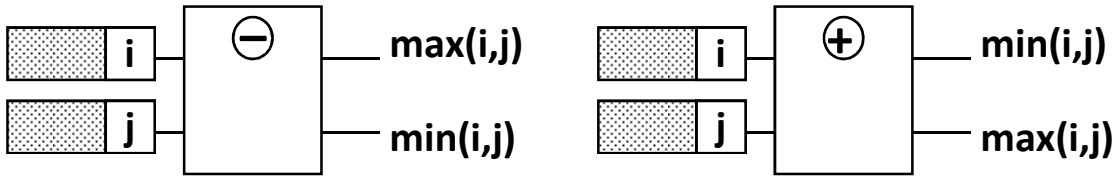


IPC: Input Port Interface Card
IGM: Input Grooming Module
ODM: Output Demultiplexng Module
OPC: Output Port Interface Card
PS: Packet Scheduler
VOQ: Virtual Output Queue
r: Cell Number / s: Speedup
g: Input Line Number
PSF: Photonic Switching Fabric
IM: Input Module
CM: Central Module
OM: Output Module

H. Jonathan Chao, Kung-Li Deng, and Zhigang Jing: A Petabit Photonic Packet Switch (P^3S), IEEE INFOCOM 2003, 0-7803-7753-2/03

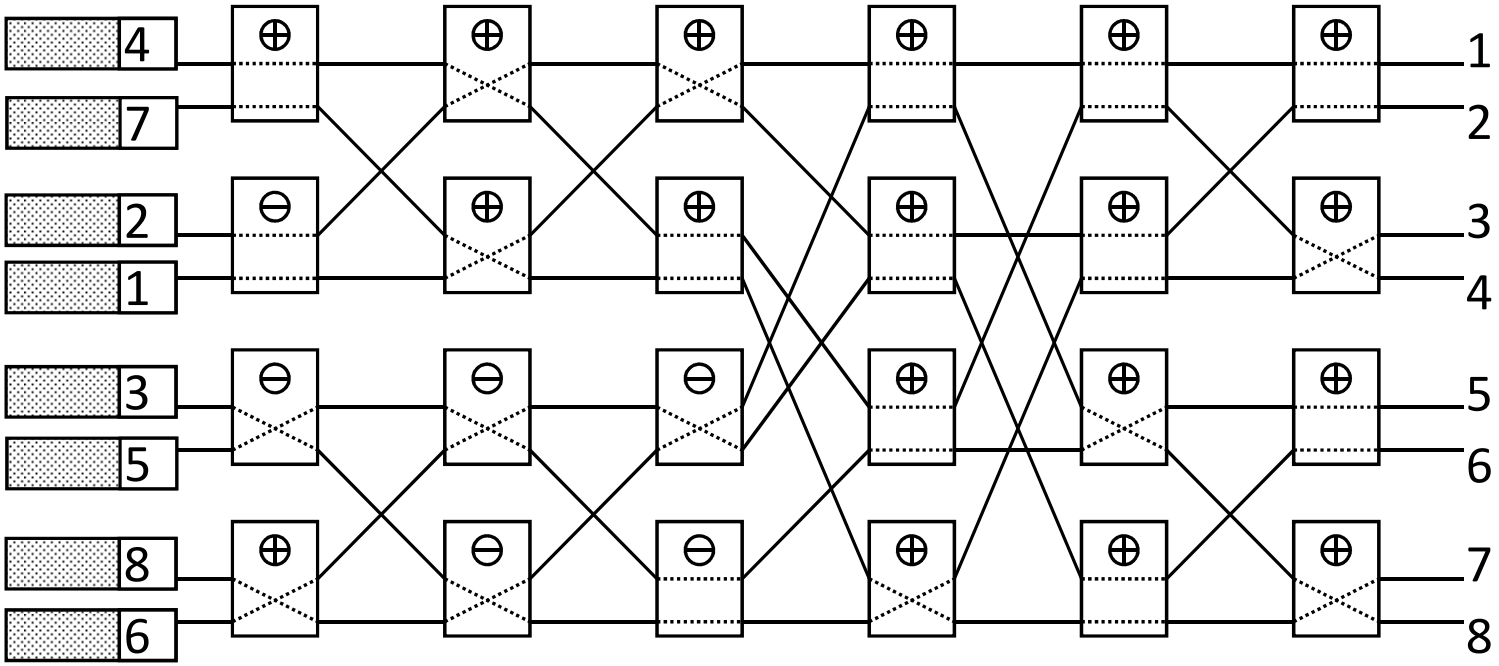
Batcherova síť

- Batcherova, nebo třídící (sorting) síť řadí vstupující pakety podle jejich adresy výstupu a to od nejmenších adres k největším.
- Batcherova třídící síť je složena z elementů:



- Vhodným seřazením takových elementů dostaneme Batcherovu síť. Celá třídící síť je složena z posloupnosti **bitonických** řadičů (bitonic sorters), které řadí své výstupy sestupně, nebo vzestupně. V našem případě jsou to třídiče velikosti 2 x 2. Pokud je na vstupu elementu jeden jen paket a tedy jen jedna adresa, uvažuje se jak by měla nižší hodnotu.

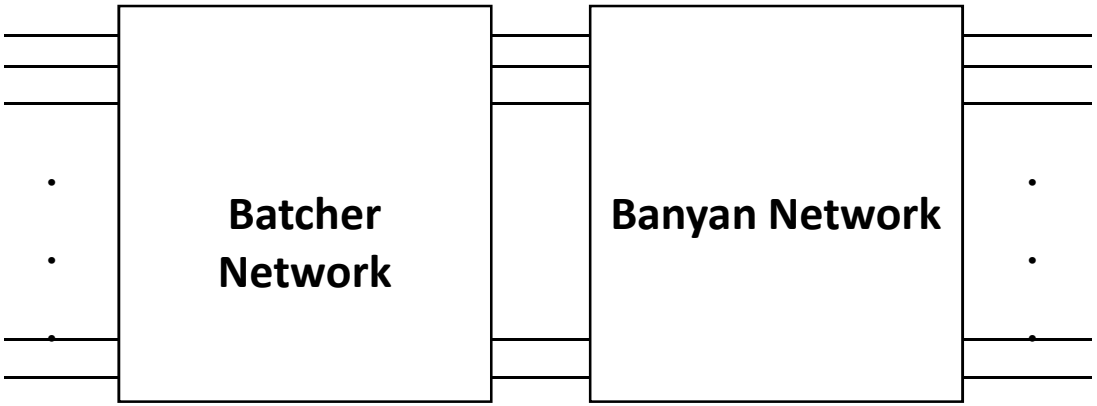
Batcherova síť



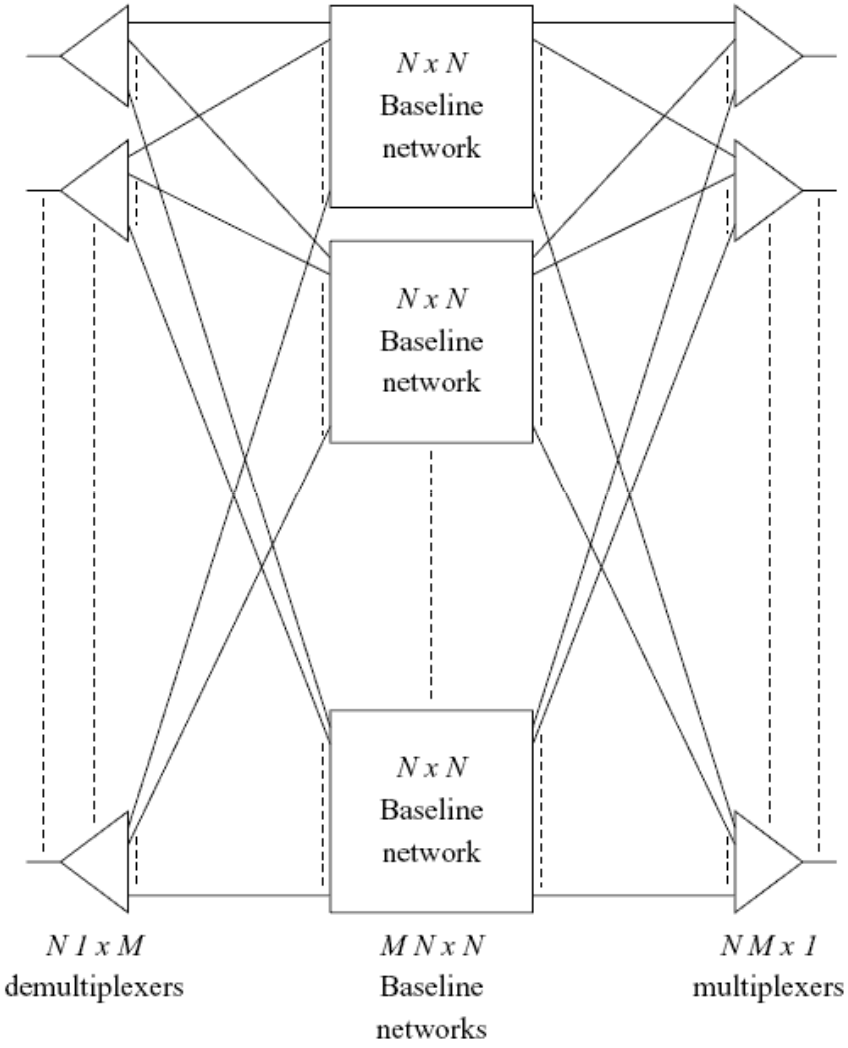
- Na výstupu Batcherovej sítě jsou pakety seřazeny podle vzestupných adres, ale nedosáhnou správný výstup podle své adresy určení (pokud nejsou obsazeny všechny vstupy). Proto se Batcherova síť dále kombinuje s banyan sítí.

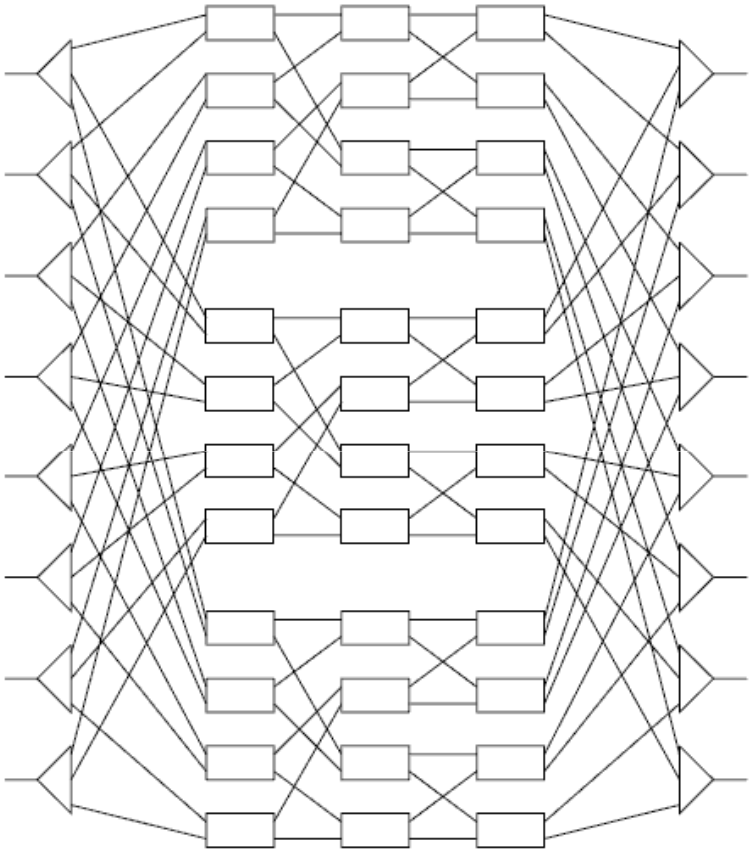
Batcher-Banyan síť

- Banyan síť zapojena za Batcherovou sítí má samosmerovacie vlastnosti a dopraví tak pakety ke správnému výstupu ze sítě. Zařazení třídící sítě před banyan síť vyloučí blokování HOL pokud předpokládáme, že žádné dva pakety nejsou směrovány k témuž výstupu. Vzhledem k tomu, že pakety jsou roztříděna v třídící síti, nevznikne ani vnitřní blokování v banyan síti. V případě pokud existuje možnost stejných výstupních adres na vstupu sítě, je třeba použít vyrovnávací paměti.



Paralelní Baseline síť





Paralelní Baseline síť

- Na vstupu v výstupu sítě paralelní baseline sítě (někdy také označované jako **Multi-Log$_2$N** network) je realizována funkce expanze a koncentrace. Na vstupu sítě jsou spojovací elementy 1xm, které expandují provoz a na výstupu elementy mx1, které provoz koncentrují.

- Pokud nepočítáme vstupní a výstupní stupně (expanze a koncentrace), počet stupňů v síti je **n=log$_2$N**. Z teorie grafů je možné dokázat, že pokud je počet paralelních baseline podsítí

  $$m \geq 2^{(n/2)}$$

  pak paralelní baseline síť je rekonfigurovatelná bez blokády. To zároveň znamená, že pro optimální paralelní baseline síť potřebujeme více spojovacích elementů než pro sériovou baseline síť (Benešovu síť). Na druhou stranu, čas přechodu paralelními baseline sítí je menší než při sériové baseline.

Cantor network

- Podobně jako Closova síť, i Cantorova síť patří mezi striktně neblokující. Cantorova síť NxN může být vytvořena z log2N Benešových sítí, N demultiplexů a N multiplexorů.