

Prepojovacie siete – motivácia

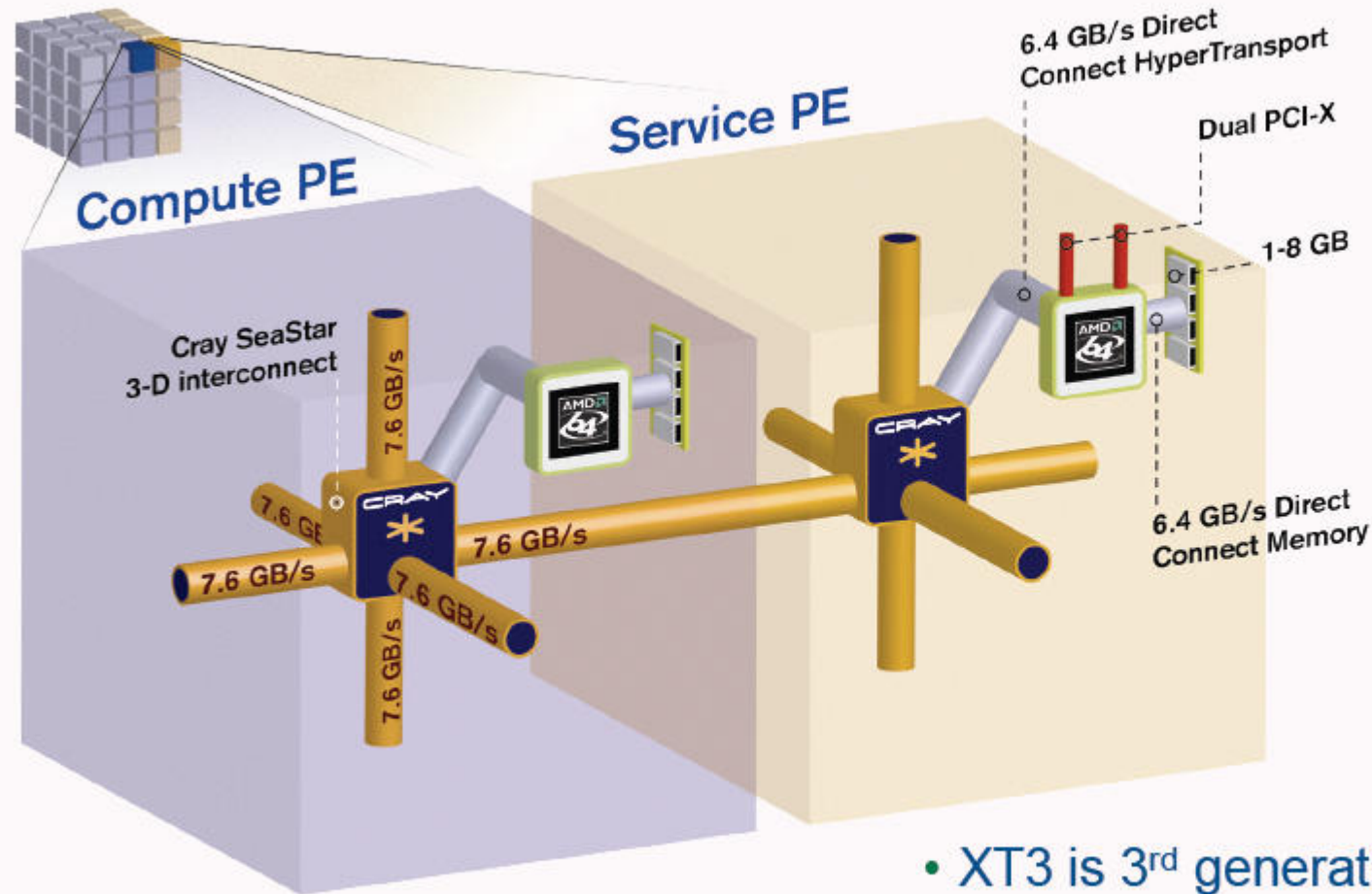
→ v dnešných superpočítačoch zohrávajú kľúčovú úlohu prepojovacie siete

Napríklad Cray XT3 (20.5 TFlops – jeden z najvýkonnejších superpočítačov na svete) v *Oak Ridge National Laboratory* pozostáva z:

- 56 skríň (cabinets), 5212 výpočtových procesorových elementov (PEs), 82 servisných procesorových elementov
- procesorový element: 64-bitový 2.4GHz AMD Optreon
- výpočtový uzol – je tvorený 4 procesorovými elementmi (2GB/PE)

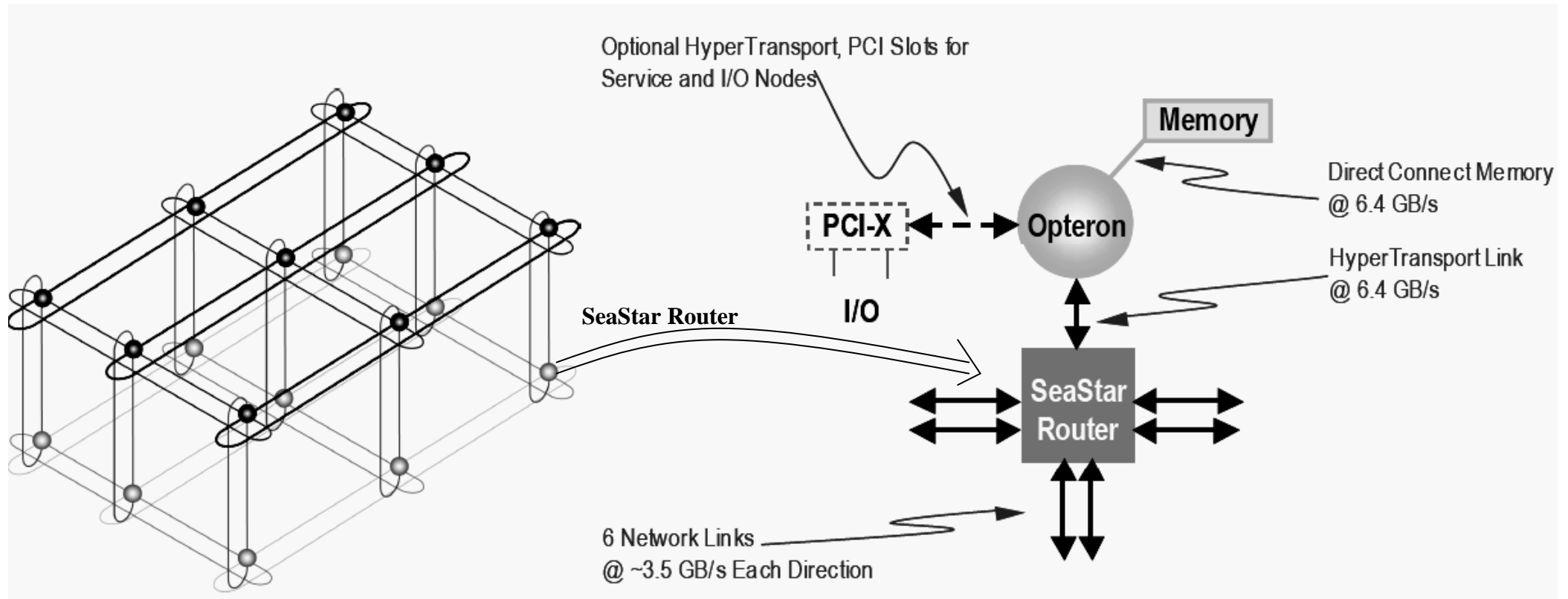


Cray XT3 Architecture

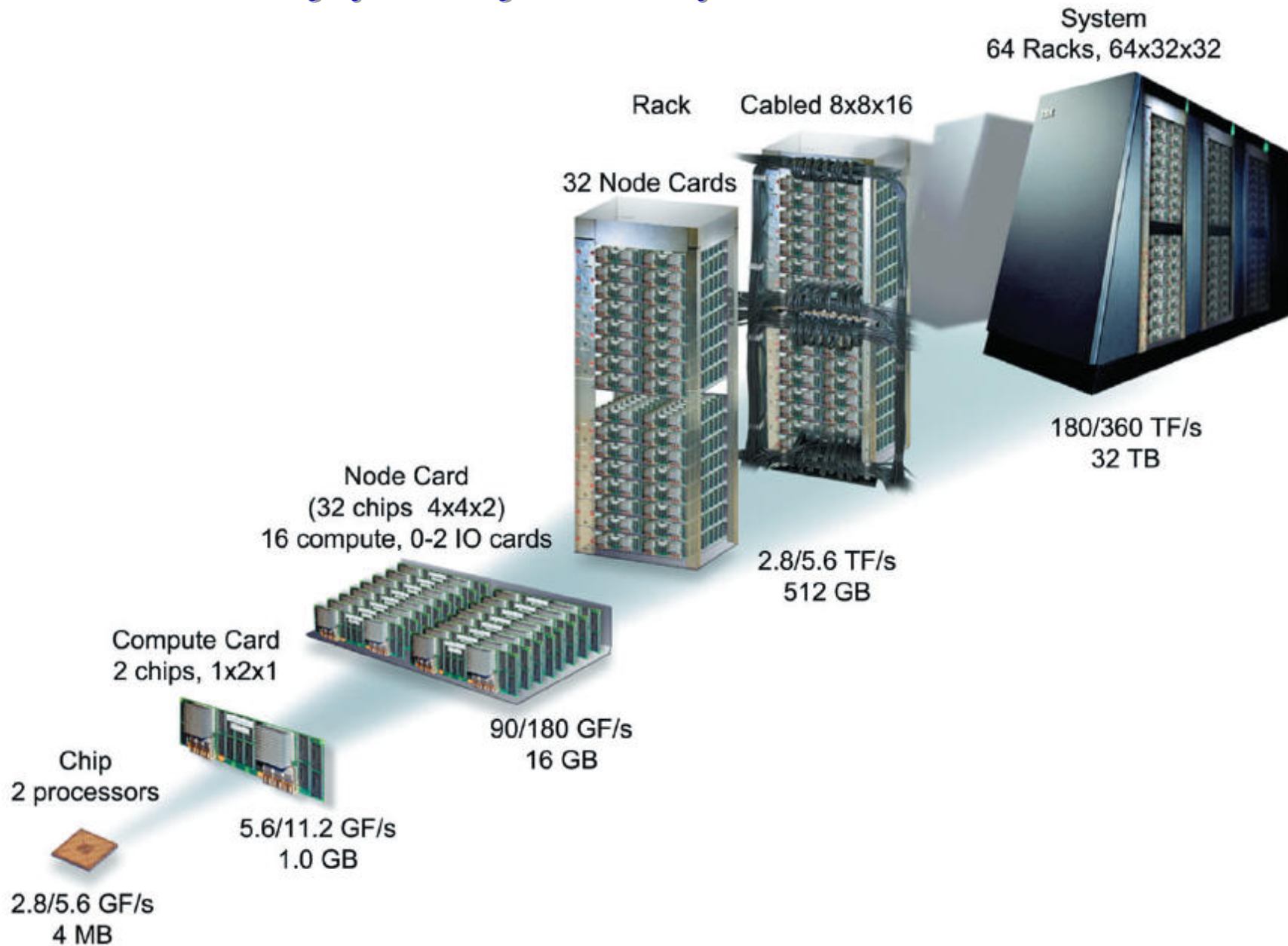


- XT3 is 3rd generation Cray MPP
- Service nodes run Linux
- Compute nodes run Catamount quintessential kernel (qk)

Each Opteron processor is directly connected to the Cray XT3 interconnect via its Cray SeaStar™ routing and communications chip over a 6.4 GB/s HyperTransport™ path .The router in the Cray SeaStar chip provides six high speed network links to connect to six neighbors in the 3D torus topology



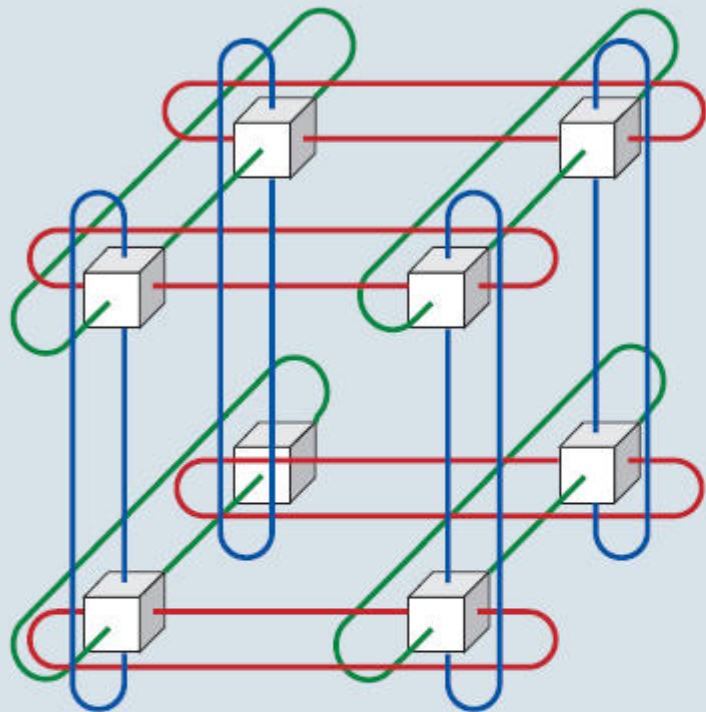
eServer Blue Gene Solution (IBM) – 280.6 Tflop/s – momentálne najvýkonnejší MPP systém (november 2006)



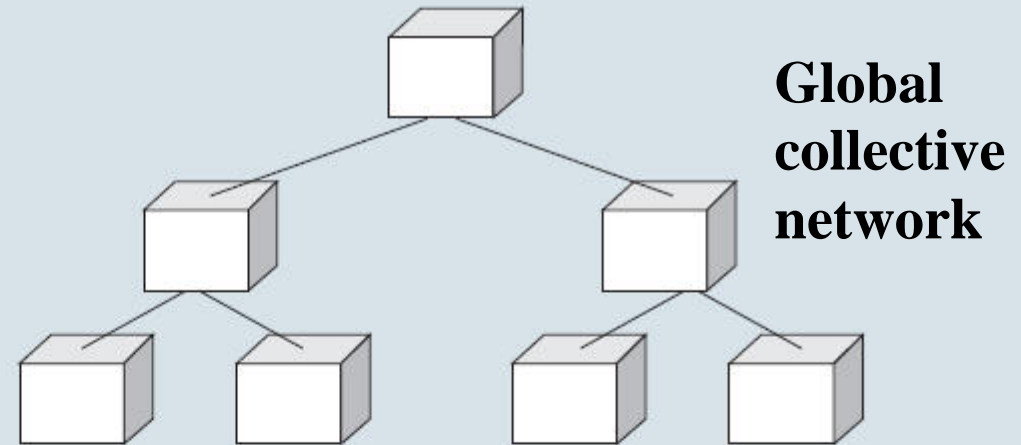
- **2 procesory IBM PowerPC 440 (700MHz) na každý uzol,**
- **2 uzly na výpočtovú kartu – (t.j. výpočtová karta obsahuje 4 procesory),**
- **16 kariet na uzlovú dosku – (t.j. uzlová doska – 64 procesorov) ,**
- **32 uzlových dosiek na skriňu (rack) – (2048 procesorov),**
- **64 skriň tvorí celý systém, takže máme 131 072 procesorov v 65536 uzloch.**

Uzly sú pospájané prostredníctvom piatich sietí:

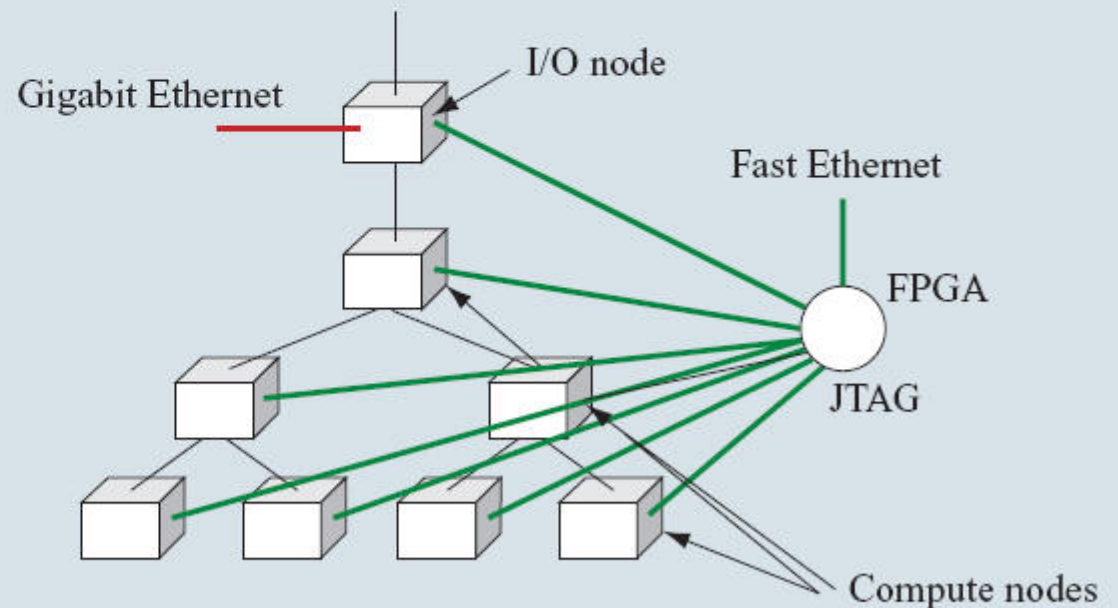
- **3D torus pre point-to-point komunikáciu medzi uzlami (175 MBps v každom smere),**
- **globálna hromadná sieť pre hromadné operácie (Global collective network) – 350 MBps, 1.5 μ s latencia (napríklad, touto sieťou dáta z jedného uzla môžu byť poslané všetkým ostatným uzlom – broadcast, alebo len niektorým),**
- **global barrier and interrupt network,**
- **control network (system boot, debug, monitoring stavu teploty, ventilátorov,...)**
- **gigabit Ethernet network pre riadenie a I/O operácie**



3D Torus



**Global
collective
network**



**Blue Gene/L control system network
and Gigabit Ethernet networks**

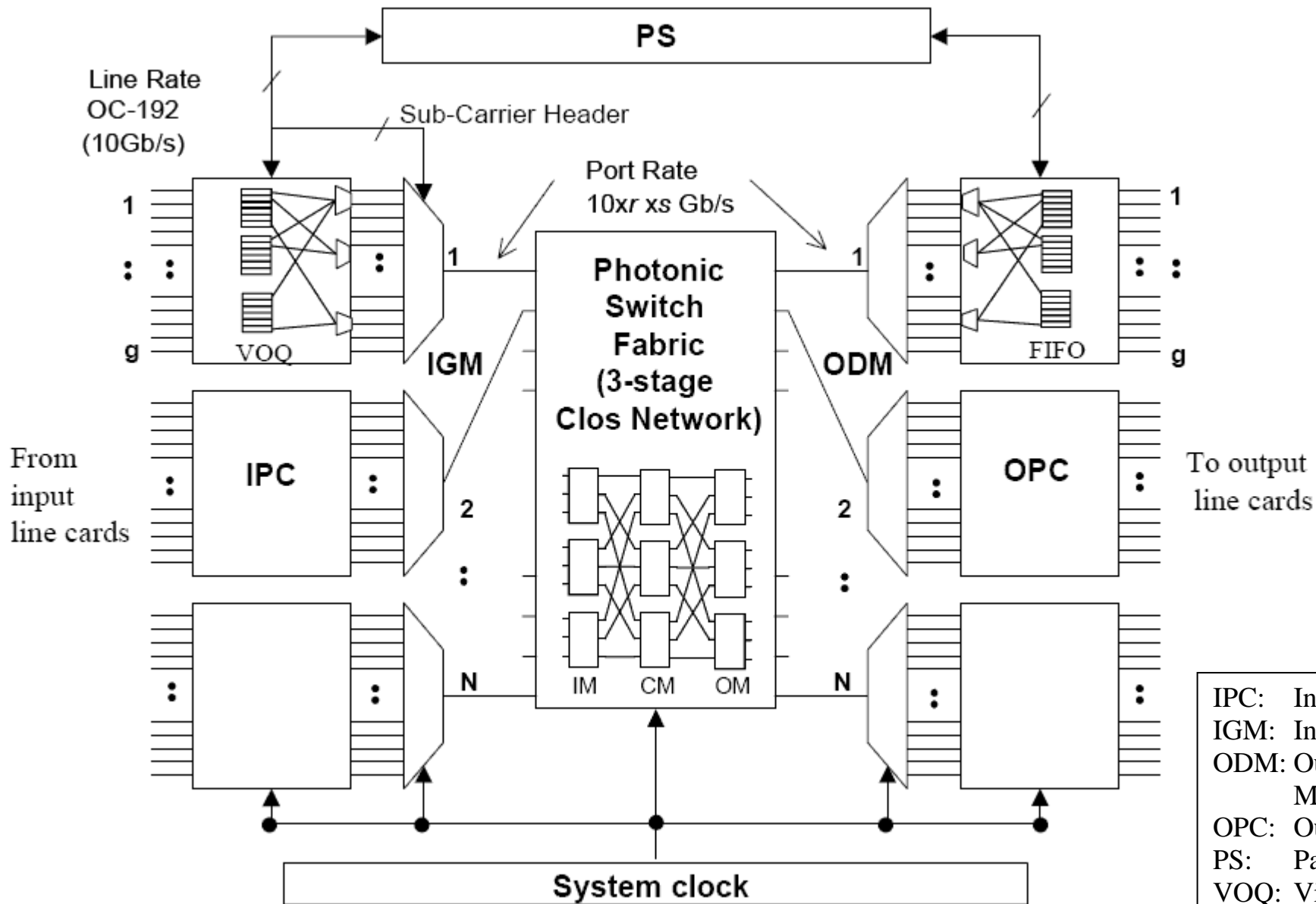
→ dnešné výkonné prepínače sú založené na viacstupňových prepojuvacích sieťach

- Pre potreby rýchleho paketového prepájania (prípadne aj burst a circuit switching) je žiadúce dosahovať čo najmenší prepínací čas.
- Súčasné potreby sú väčšie ako kapacita dnešných elektrických prepínačov.
- Preto, tam kde sú bežné elektrické (opticko-elektricko-optické) prepínače nedostačujúce, je potrebné zaradiť plne optický (OOO) prepínač

Dnešné technológie, ktoré dosahujú veľmi nízke prepínacie časy (v rádoch ns až μ s) zvyčajne neumožňujú vyrobiť prepínač požadovanej dimenzie (len do 32×32). Preto ak chceme vybudovať prepínač väčšej dimenzie (1000×1000 a viac) ako umožňuje daná technológia, ako rozumné (a jediné) riešenie sa javí použitie viacstupňových prepojuvacích sietí.

Napríklad architektúra petabitového paketového optického prepínača P³S dimenzie 6400×6400 je založená na Closovej 3-stupňovej sieti dosahujúcej celkovo 1.024 petabit/s (160 Gbit/s na každý port)

(1 petabit/s = 10^6 Gbit/s)



- IPC: Input Port Interface Card
- IGM: Input Grooming Module
- ODM: Output Demultiplexing 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

Topológie prepojovacích sietí



Statické siete

- **lineárne 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 siete

- **siete založené na zberniciach (bus network)**
- **jednostupňové siete (krížové prepínače -crossbars)**
- **viacstupňové siete (omega, Banyan, Cantor, Clos,...)**

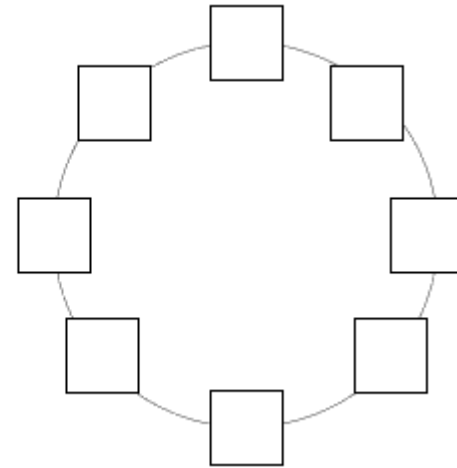
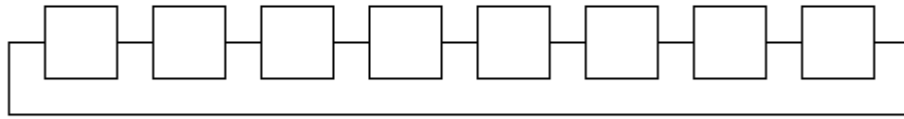
Statické siete

- **veľkosť siete N** (Network size): počet uzlov v sieti,
- **stupeň uzla d** (Node degree): počet hrán vstupujúcich alebo vystupujúcich z uzla,
- **bisekčná šírka B** (Bisection width): minimálny počet hrán, ktoré musíme prerušiť pri rozdelení siete na dve rovnaké polovice,
- **priemer siete D** (Network diameter): počet hrán maximálnej najkratšej cesty medzi dvoma ľubovoľnými uzlami siete – ukazuje najdlhšiu komunikáciu,
- **cena C** (Cost): počet komunikačných liniek (hrán)



Linear array

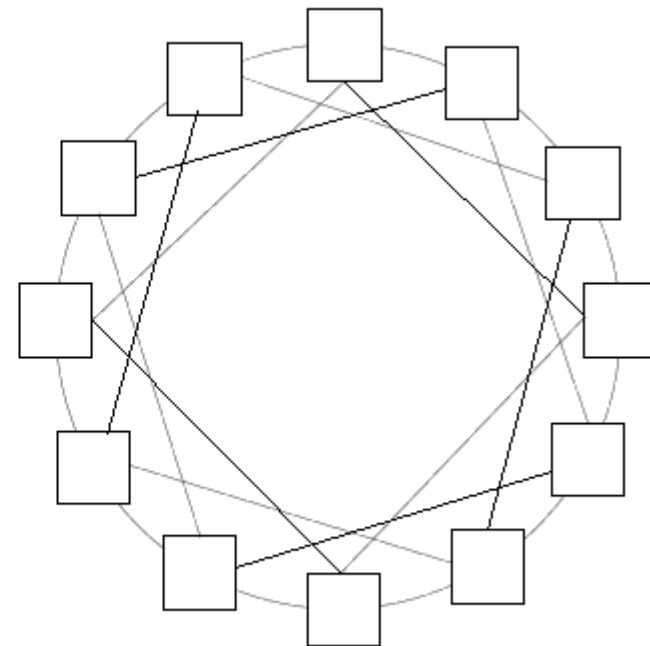
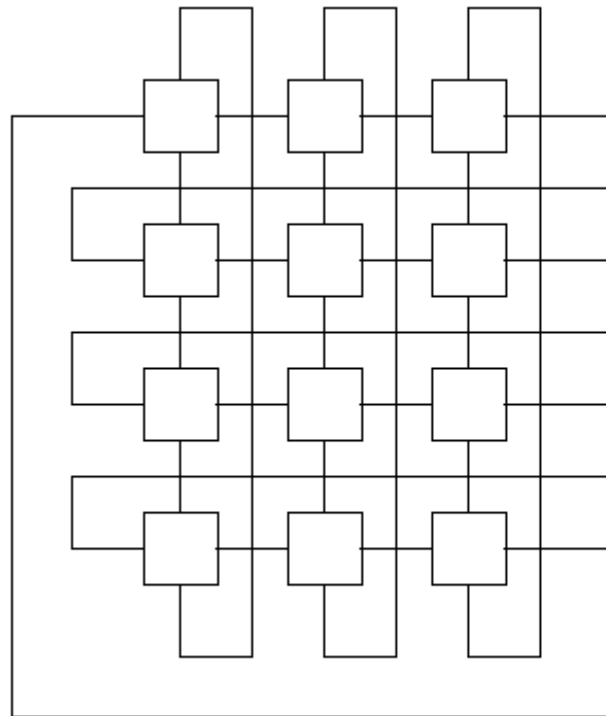
d: 1 (pre koncové uzly),
2 (pre ostatné)
D: $N-1$
B: 1
C: $N-1$



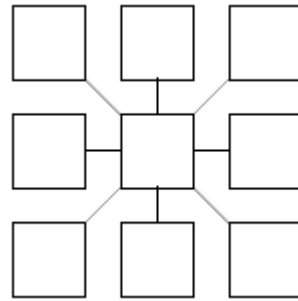
d: 2
D: $N/2$
B: 2
C: N

Ring

**alternatívna
 reprezentácia**

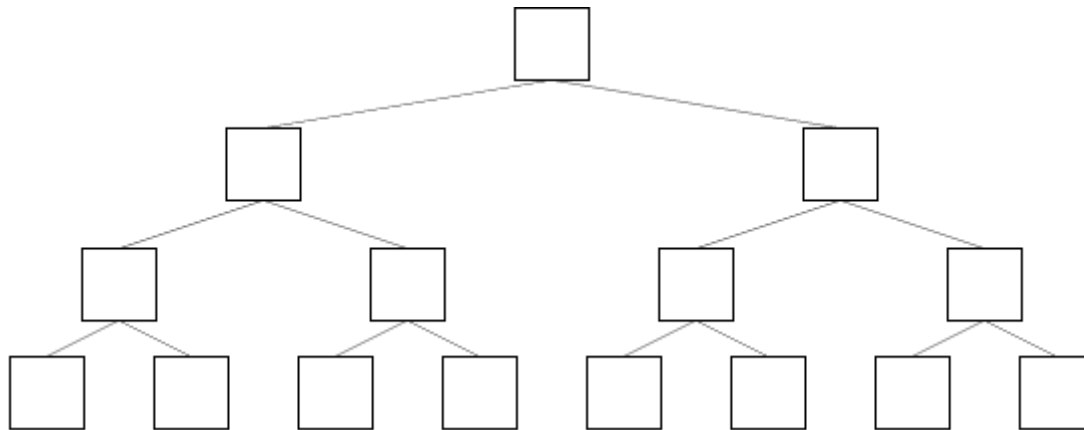


Chordal ring



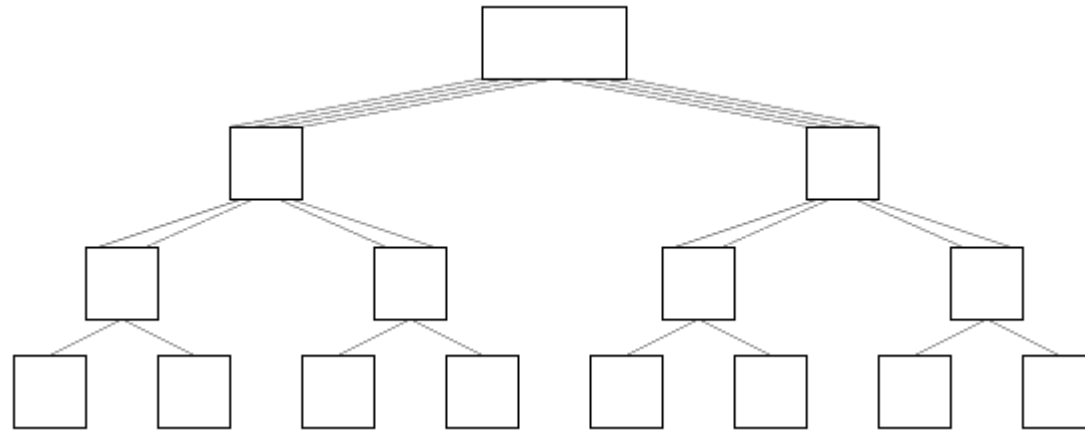
Star

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

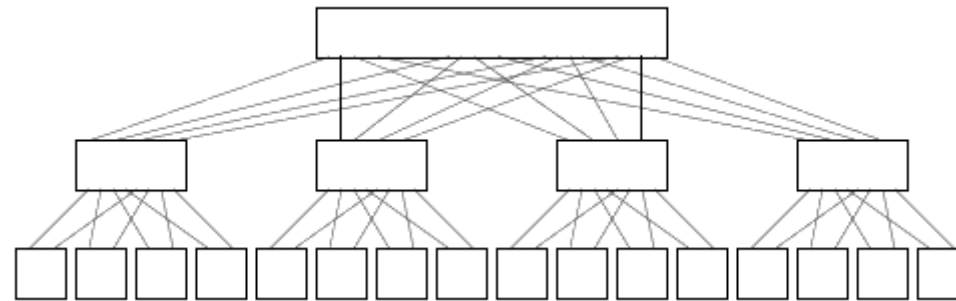


Binary tree

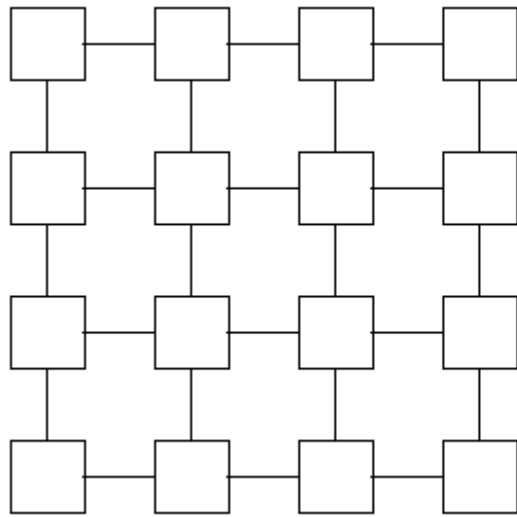
d: 1 (koncové uzly), 2 (koreňový), 3 (ostatné)
D: $2\log((N+1)/2)$
B: 1
C: N-1



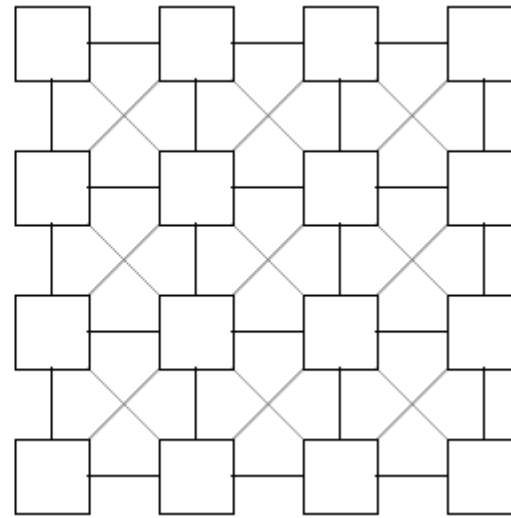
Binary fat tree



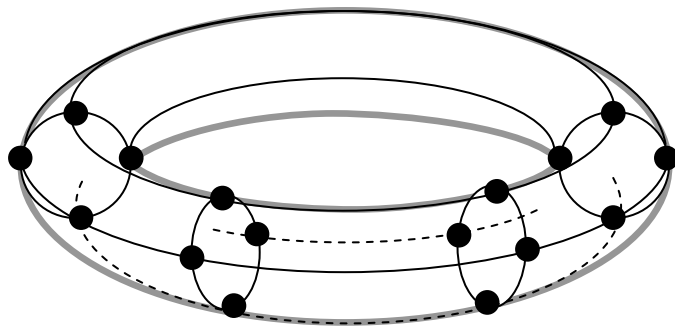
4-ary fat tree



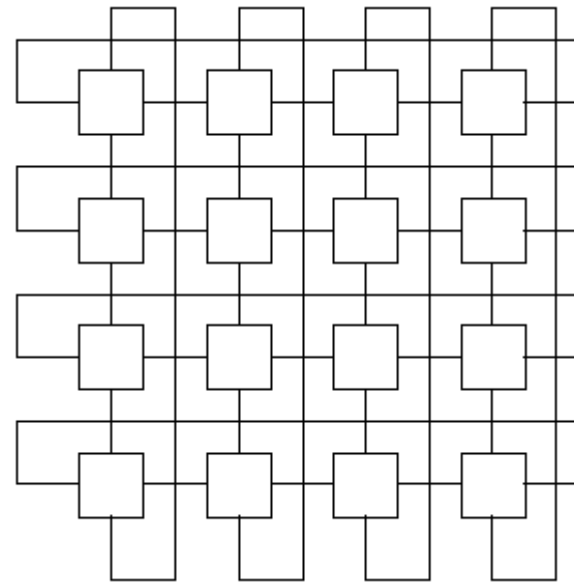
4-connected 2D mesh



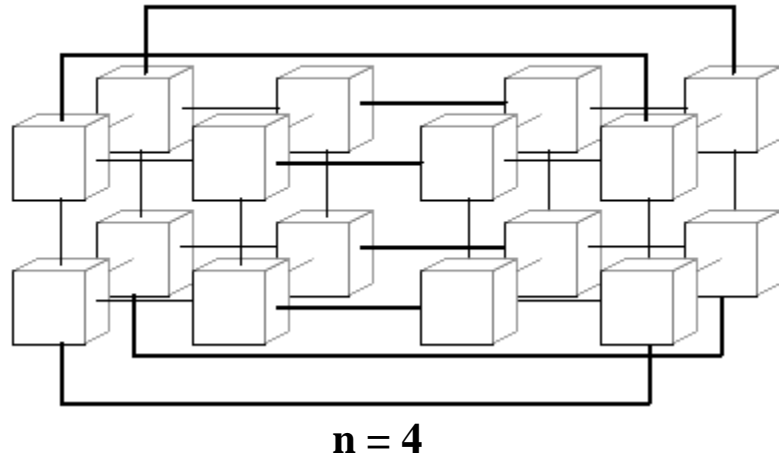
8-connected 2D mesh



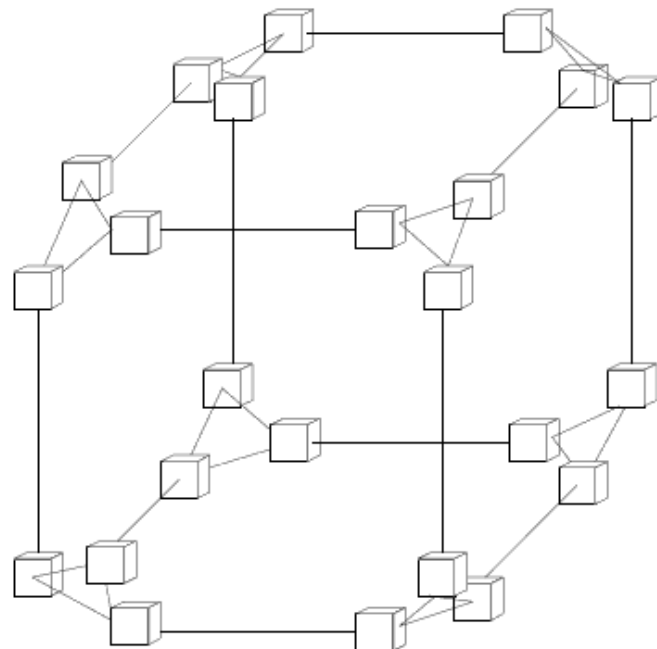
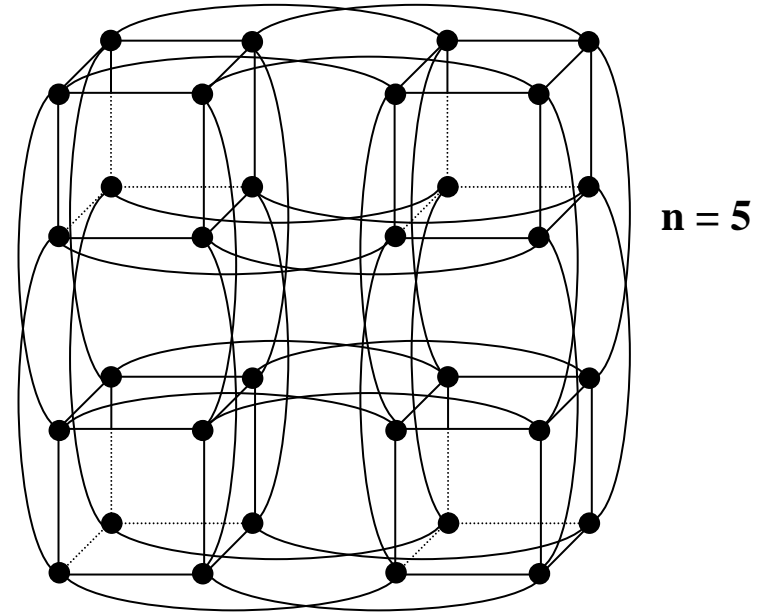
2D Torus (anuloid)



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

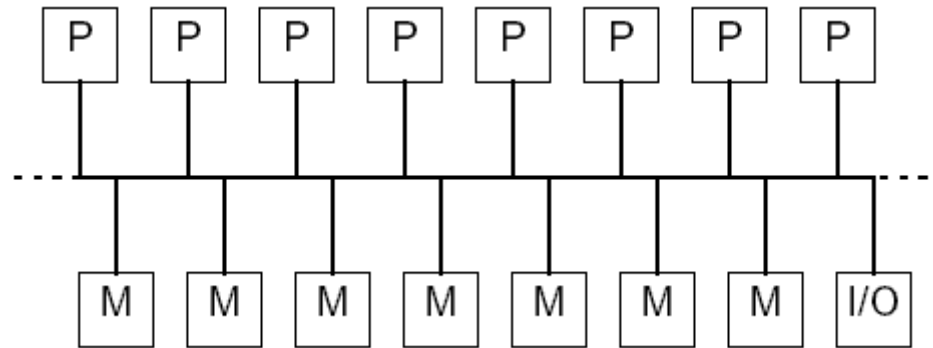


Hypercube

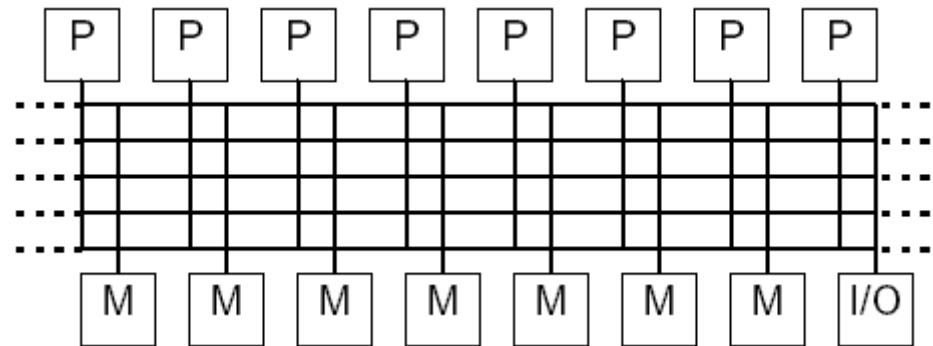


3-Cube Connected Cycles

Bus networks

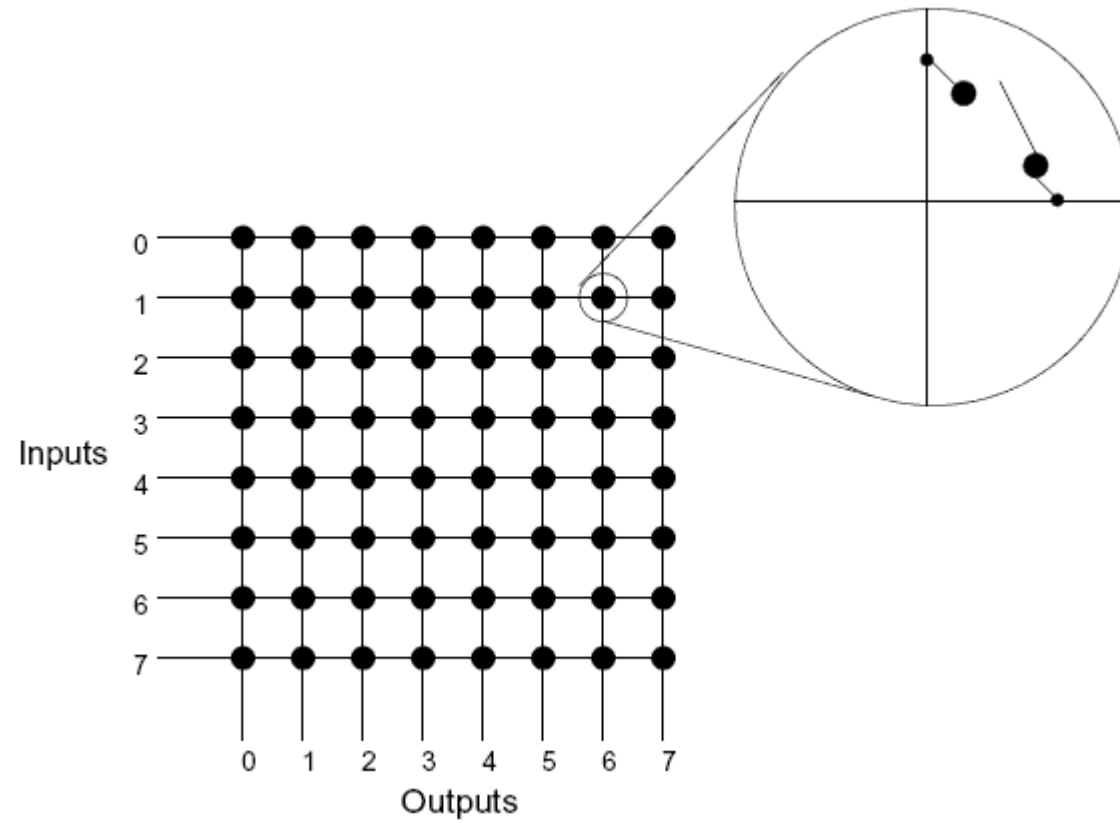


Bus-based multiprocessor system



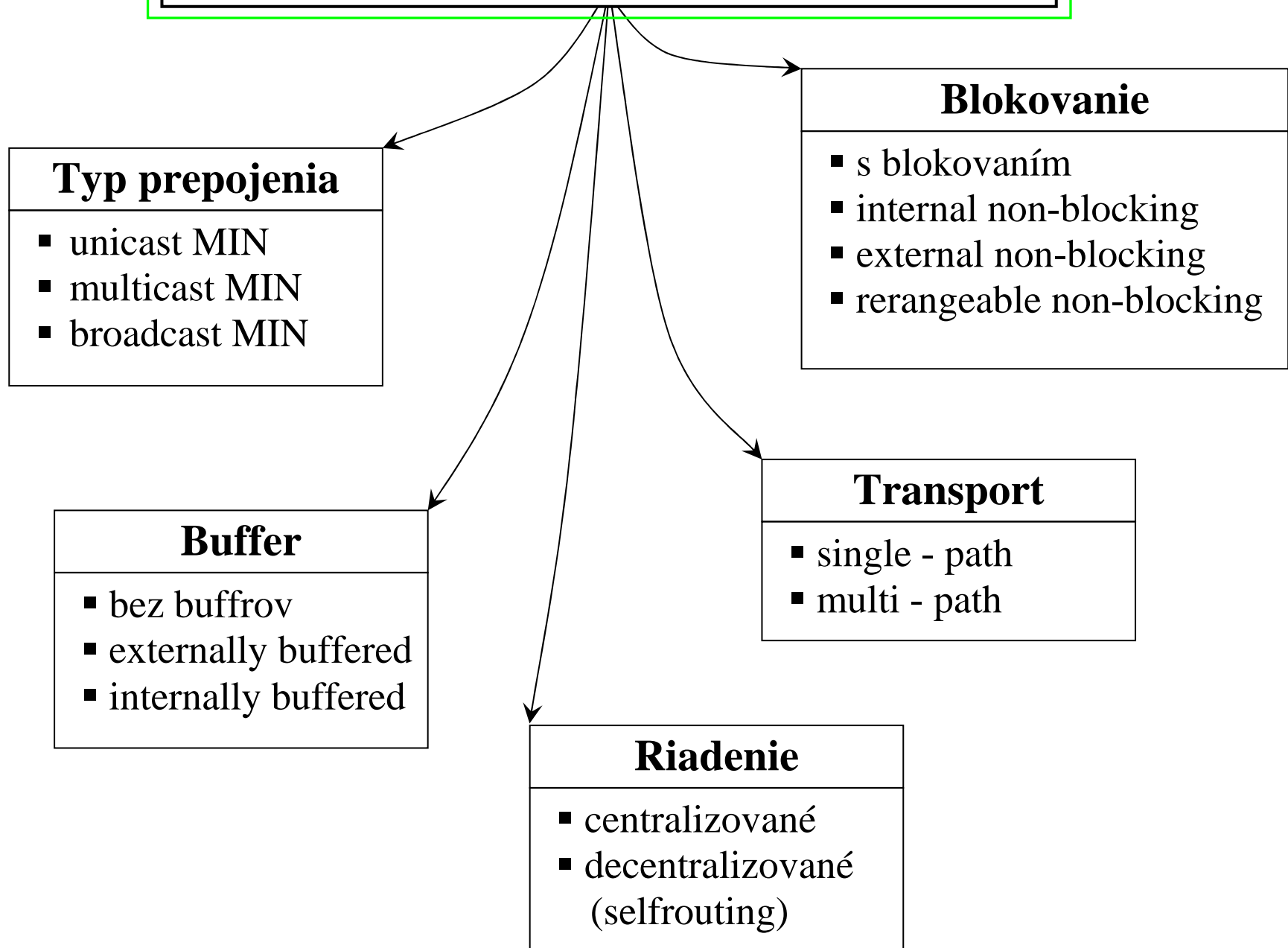
Multiple bus architecture

Single Stage Networks



Crossbar

Viacstupňové prepožovacie siete (MIN)



-
- **Unicast MIN** (one-to-one, poin-to-point) – je sieť, ktorá prepája v tom istom čase práve jeden vstup s práve jedným výstupom.
 - **Multicast MIN** (multipoint) – prepojí v tom istom čase jeden vstup siete s viacerými výstupmi siete
 - **Broadcast MIN** – prepája jeden vstup siete so všetkými výstupmi siete v tom istom čase. Broadcast MIN je špeciálnym prípadom Multicast MIN.
-
- **Jednocestná MIN** (*Single-Path*) – všetky pakety v rámci jedného virtuálneho spojenia používajú tú istú cestu.
 - **Viaccestná MIN** (*Multi-Path*) – pakety v rámci jedného virtuálneho spojenia používajú viacero ciest. Pakety sú náhodne distribuované. Vnútoraná prevádzka v spojovacej sieti sa stane nezávislou na vonkajšej prevádzke (t.j. na pomere služieb, ktoré sú prepájané v sieti). Takáto sieť vyžaduje opatrenia pre radenie buniek pri východe zo siete.
-

- **MIN s centralizovaným riadením** –riadenie je vykonávané centrálnym procesorom. Procesor určuje výber cesty v sieti, ak sú dané požiadavky na spojenie.
 - **MIN s decentralizovaným riadením** –riadenie je distribuované na spojovacie elementy (samosmerovanie – *self routing*). V tomto prípade potrebujeme po vstupe do siete každý paket prídavnú informáciu, nazývanú *tag*, na základe ktorej sa v každom elemente určuje kombinácia vstupu a výstupu.
-
- **Siete s blokováním** – sú siete kedy sa konkrétny vstup nemôže prepojiť s konkrétnym výstupom, napriek tomu že výstup nie je obsadený iným spojením. Blokovanie je spojené so stratou informácie, alebo jej oneskorením. Pre zabezpečenie kvality spojovania je žiadúce blokovanie vylúčiť, alebo ho minimalizovať na špecifikovanú úroveň.
 - **Siete bez vnútorného blokovania** (Internal non-blocking) – sú siete, ktoré zabezpečujú prepojenie ľubovoľného vstupu s ľubovoľným výstupom bez zrušenia, alebo rekonfigurácie inej vnútornej cesty v sieti. Môže však nastať prípad, že aj keď sieť je bez vnútorného blokovania, dva, alebo viac vstupov môže chcieť použiť ten istý výstup v tom istom čase – vzniká konflikt .

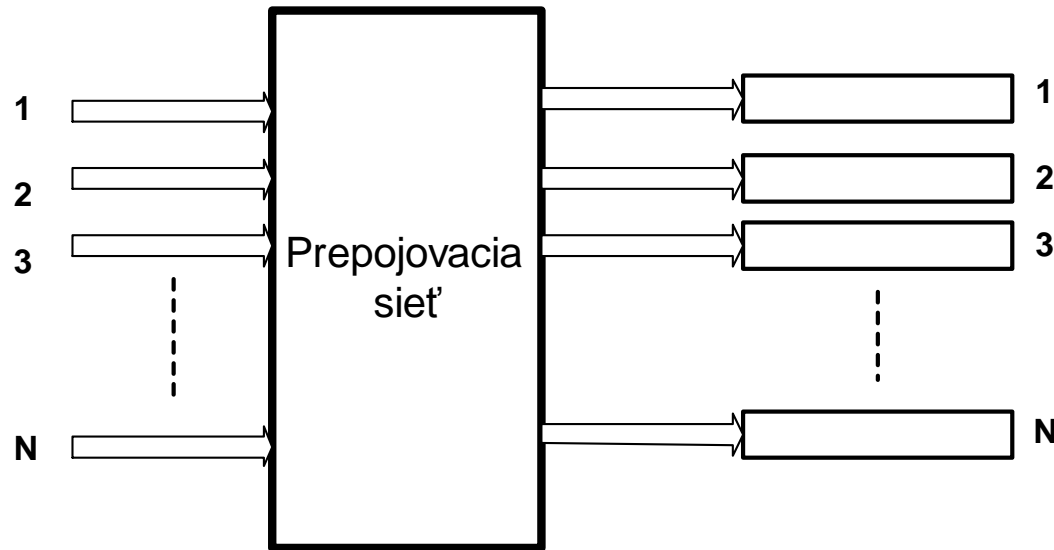
- **Siete bez vonkajšieho blokovania** (External non-blocking) – sú siete schopné prepojenia ľubovlného vstupu s ľubovlným výstupom v každom prípade. V sieti je potrebná v prípade konfliktu funkcia uloženia informácie, t.j. vyrovnávacia pamäť.
 - **Rekonfigurovateľné (preusporiadateľné) siete** bez blokovania (Rearrangeable non-blocking) – sú osobitným prípadom blokovacích sietí – avšak dokážu vždy realizovať spojenie z ľubovlného vstupu na ľubovlný výstup. V prípade konfliktu dochádza k rekonfigurácii (preusporiadaní) existujúcich ciest a k vytvoreniu novej konfigurácie spojení.
-
-

- **MIN s vonkajšou vyrovnávacou pamäťou** (Externally Buffered) – vyrovnávacie pamäte sú umiestnené na *vstupoch siete, na výstupoch siete, alebo kombinovane.*
 - **MIN s vnútornou vyrovnávacou pamäťou** (Internally Buffered) – vyrovnávacie pamäte sú v jednotlivých spojovacích elementoch, čiže vo vnútri siete. Aj v spojovacom elemente môže byť vyrovnávacia pamäť na vstupe (Input Buffering), na výstupe (Output Buffering), alebo v strede elementu (Central Buffering)
-
-

Hlavné spôsoby radenia (ukladania) paketov:

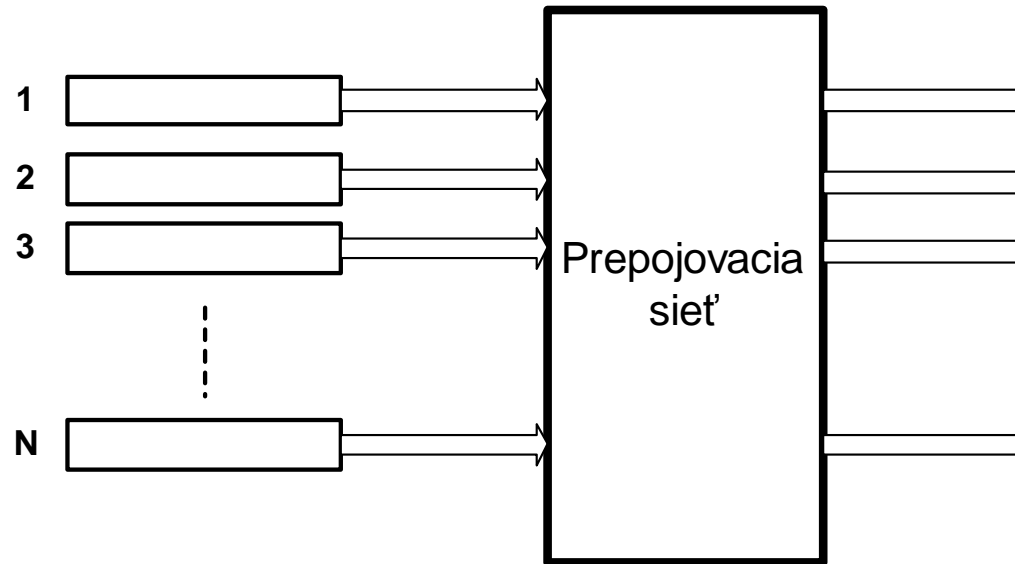
- **výstupné radenie (OQ)**
- **vstupné radenie (IQ)**
- **kombinované vstupno-výstupné radenie (CIOQ)**
- **centralizované zdieľané radenie (CSQ)**
- **virtuálne výstupné radenie (VOQ)**

Výstupné radenie paketov: (OQ - output queuing)



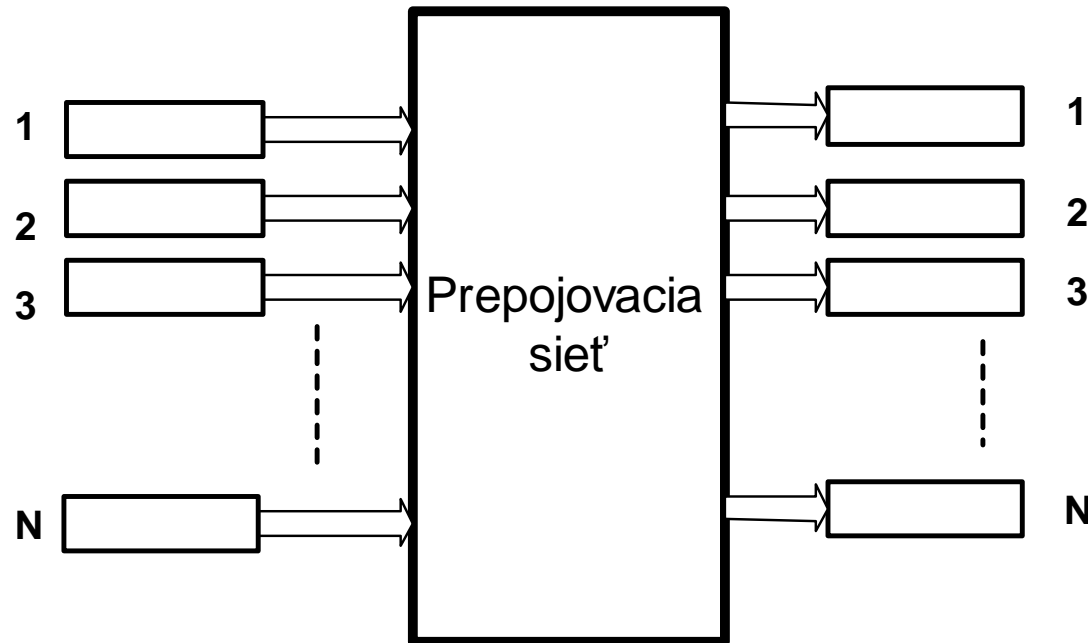
Ked' paket dôjde na vstupný port, je okamžite uložený do buffera, ktorý je na príslušnom výstupnom porte. Pretože pakety určené pre ten istý výstupný port môžu prísť súčasne z mnohých vstupných portov, výstupný buffer potrebuje radiť pakety oveľa väčšou rýchlosťou než vstupný port. V najhoršom možnom prípade to môže byť až N krát rýchlejšie (kde N je počet vstupných portov), a to vtedy ak pakety zo všetkých vstupných portov sú určené pre jeden určitý výstupný port. Rýchlosť, ktorou môžeme pristupovať do výstupného buffera je limitovaná.

Vstupné radenie paketov: (IQ - input queuing)



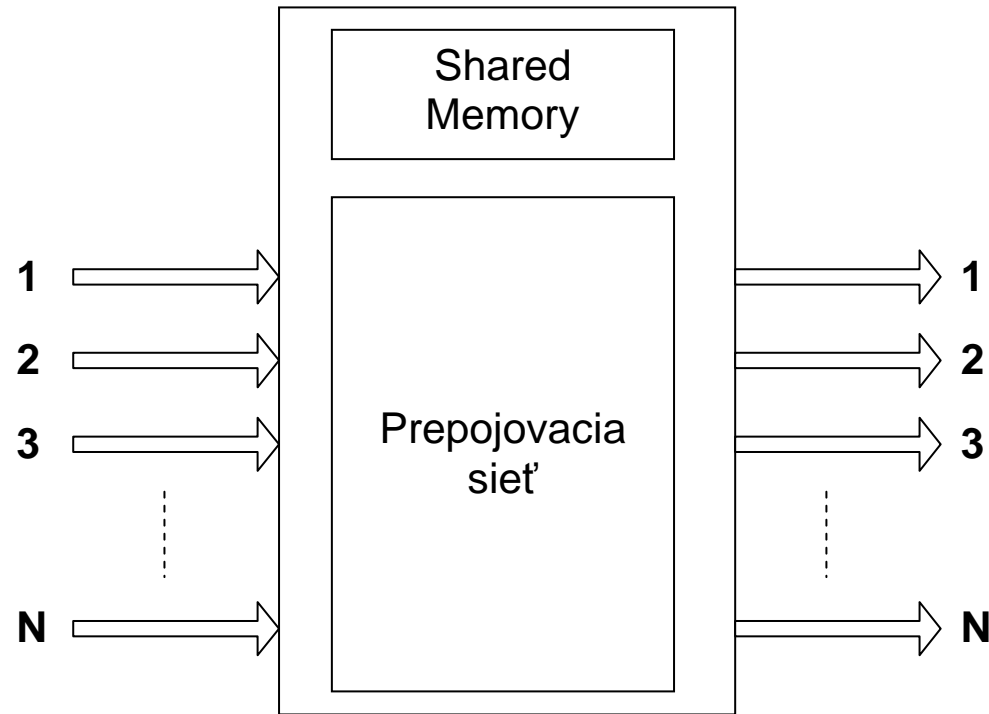
Vstupné radenie nemá limity aké má napríklad výstupné radenie alebo centralizované zdieľané radenie. V tejto architektúre má každý vstupný port FIFO zásobník buniek, a len prvá bunka vo fronte je spôsobilá na prenos v priebehu daného časového úseku. Nevýhodou FIFO radenia je, že keď bunka na čele fronty sa zablokuje, zablokujú sa všetky bunky za ňou a zabráni sa tak ich prenosu, dokonca aj keď je výstupný port voľný. Toto sa nazýva Head-of-line blokovanie. Matematickou analýzou a počítačovou simuláciou bolo ukázané, že HOL blokovanie obmedzuje priepustnosť každého vstupného portu na maximálne 58.6 percent a to pri náhodnej hustote prepravy, a táto hodnota je ešte omnoho nižšia pri veľmi hustej preprave.

Kombinované vstupno-výstupné radenie paketov: (CIOQ - combined input-output queuing)



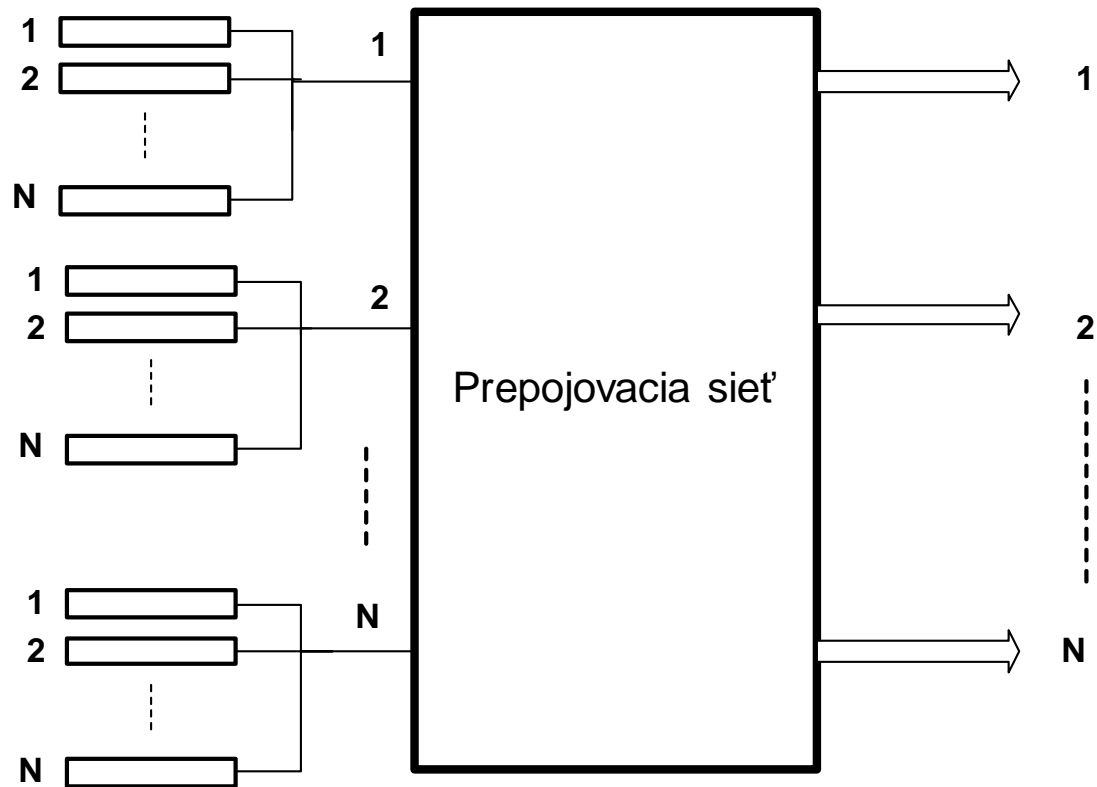
Táto schéma radenia je kombináciou vstupného a výstupného radenia. Je to dobrý kompromis medzi výkonom a možnosťou rozširovania OQ a IQ prepínačov. Pre vstupno radené prepínače, nanajvýš jeden paket môže byť doručený na výstupný port v jednom časovom úseku. Pre výstupne frontované prepínače, až N paketov môže byť doručených na výstupný port za jeden časový úsek. Použitím CIOQ, namiesto týchto dvoch extrémnych možností, môžeme si zvoliť kompromis medzi nimi.

Centralizované zdieľané radenie paketov : (Centralized shared queuing)



Máme buffer zdieľaný všetkými vstupnými portmi prepínača, na ktorý sa môžeme pozerat' ako na zdieľanú pamäťovú jednotku, a ktorý má N súbežných prístupov na zápis pre N vstupných portov, a N súbežných prístupov na čítanie pre N výstupných portov. Pretože pakety určené pre ten istý výstupný port môžu prichádzať súčasne z viacerých vstupných portov, výstupný port musí byť schopný načítavať omnoho rýchlejšie ako vstupný port dokáže zapisovať údaje.

Virtuálne výstupné radenie paketov : (VOQ - virtual output queuing)



Táto schéma radenia zvláda Head-of-line blokovanie a zároveň si udržuje svoju výhodu rozšíriteľnosti. Pri tejto metóde každý vstupný port udržuje izolovanú frontu pre každý výstupný port. Kľúčovým faktorom pre dosiahnutie vysokého výkonu použitím VOQ prepínačov je plánovací algoritmus, ktorý je zodpovedný za výber paketov, ktoré by mali byť prenesené v každej časovej jednotke zo vstupných porotv na výstupné. Niekoľko takýchto algoritmov už bolo navrhnutých, ako napr. PIM (parallel iterative matching - paralelné iteračné párovanie), iSLIP a RPA. Bolo ukázané, že s menej ako štyrmi opakovaniami vyššie uvedeného riadiaceho algoritmu PIM, priepustnosť prepínača presiahne 99 percent.

Existuje veľa riadiacich algoritmov vyvinutých pre virtuálne radenie paketov, ktoré vychádzajú z vyššie uvedených algoritmov a prinášajú značné zlepšenie v otázke riadenia prepínača.

Základné typy viacstupňových sietí

Unicast siete s blokovaním jednocestné (Single-Path)

Unicast spojovacie siete vykonávajú one-to-one spojenie, t.j. spájajú jeden vstup s jedným výstupom v tom istom čase. Jednocestné spojovacie siete (Single-Path) majú práve len jednu cestu a tým len jednu možnosť spojenia medzi ľubovoľným vstupom a ľubovoľným výstupom.

- **Baseline sieť**
- **Banyan sieť**
- **Delta sieť**
- **Omega sieť**

Unicast siete s blokovaním viaccestné (Multi-Path)

Viaccestné siete zabezpečujú alternatívne cesty medzi vstupmi a výstupmi, pričom zachovávajú samosmerovacie vlastnosti siete a len minimálne komplikujú jej časovú zložitosť. Zlepší sa spoľahlivosť aj priepustnosť siete.

- Banyan sieť s delenou záťažou
- Data Manipulator – DM
- Augmented Data Manipulator – ADM
- Reverse Augmented Data Manipulator – IADM
- gama siete

Unicast siete bez blokovania

Všetky siete s blokovaním vyžadujú opatrenia na potlačenie blokovania. Najčastejšie je to umiestnenie vyrovnávacích pamätí do siete. V takomto prípade pribúda problém radenia buniek pri vytváraní radov v pamätiach. Najznámejší problém radenia je HOL (Head-of-Line). Nastane vtedy ak v pamäti FIFO je na čele radu paket, ktorý bráni v ceste za ním čakajúcim paketom, hoci tie by

v ďalšej sieti neboli blokované. Je to typický problém sietí s umiestnením vyrovnávacích pamätí na vstupoch.

Problém blokovania je možné riešiť už pri návrhu siete vytvorením siete bez blokovania. Pri tom rozlišujeme dve možnosti. Bud' sú siete topologicky bez blokovania, t.j. ich architektúra minimalizuje pravdepodobnosť blokovania, alebo sú s riadením bez blokovania, t.j. mechanizmus riadenia siete odstraňuje blokovanie. Druhý prípad nazývame aj rekonfigurovateľné siete.

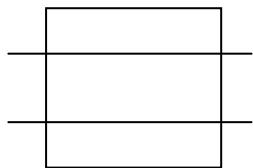
- **Benešova sieť**
- **Paralelná baseline sieť**
- **Closova sieť**
- **Batcherova sieť**
- **Batcher-banyan sieť**

Multicast siete

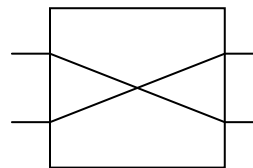
Multicast viacstupňové spojovacie siete sú zvyčajne siete s N vstupmi a N výstupmi, pričom ľubovoľná skupina vstupov sa môže prepojiť s ľubovoľnou skupinou výstupov. Každý vstupný port môže byť spojený s viac ako jedným výstupom, ale každý výstupný port je zvyčajne spojený s najviac jedným vstupným portom.

Multicast siete môžu uskutočniť N^N rôznych spojení, majú teda väčší výkon ako unicast siete, ktoré uskutočňujú ono-to-one permutáciu vstupov na výstupy a zrealizujú maximálne $N!$ rôznych spojení.

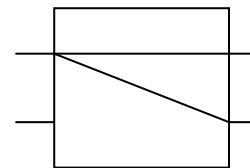
V podstate ako multicast sieť môže fungovať každá unicast sieť ak jej spojovacie elementy dokážu prepojiť svoje vstupy na viacero výstupov.



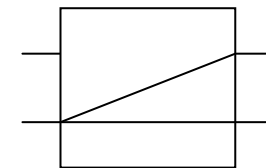
Straight-trough



Exchange

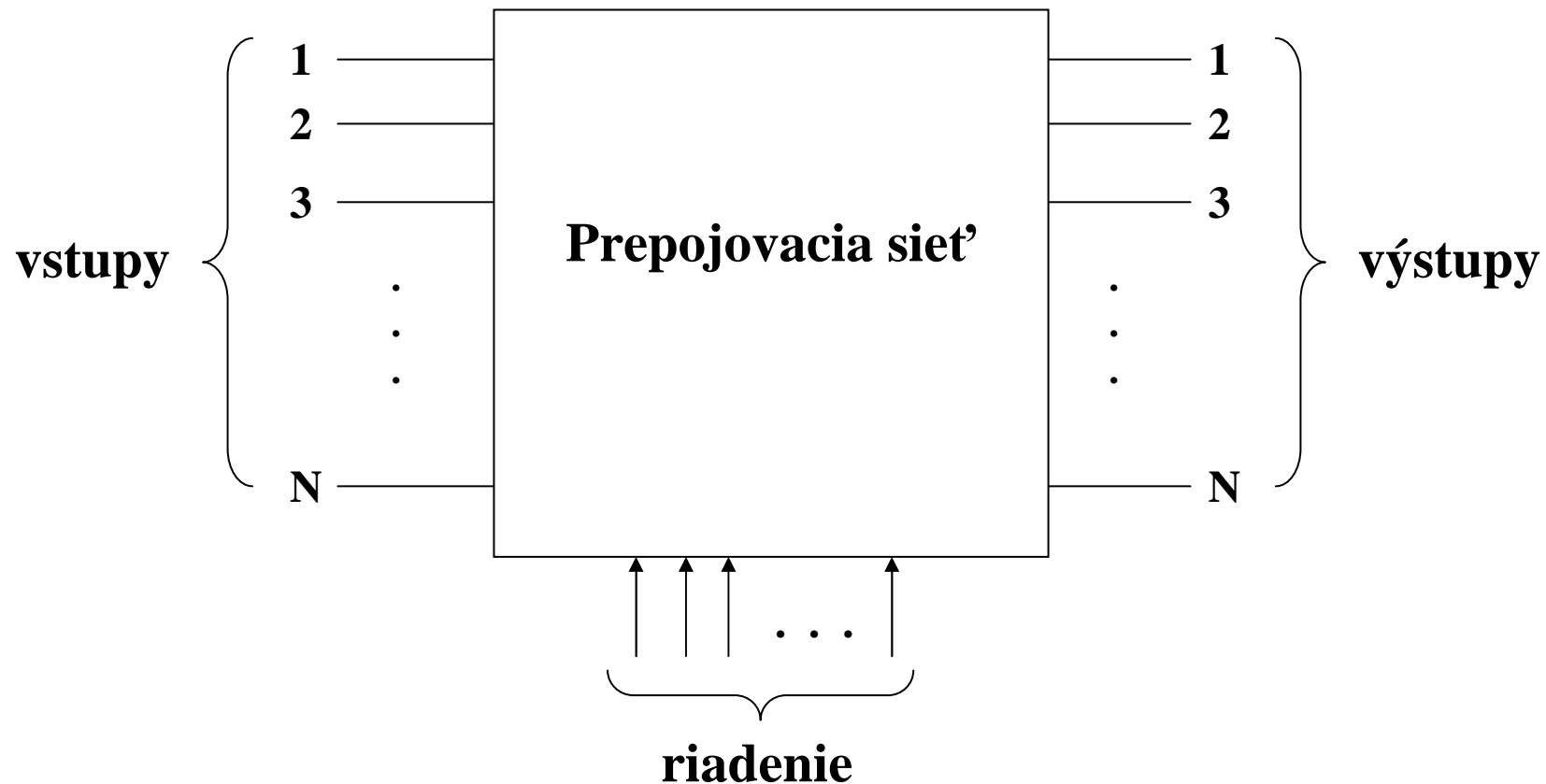


Upper-broadcast



Lower-broadcast

Bloková schéma prepojovacej siete



→ Prepojovacia sieť – PS $[N \times N]$ je prostriedok, ktorý umožňuje prepojiť ľubovoľný zo svojich N vstupov $(X_0, X_1, \dots, X_{N-1})$ s ľubovoľným zo svojich N výstupov $(Y_0, Y_1, \dots, Y_{N-1})$.

→ Vo všeobecnom prípade možno realizovať siete, ktorých stavy predstavujú ľubovoľné priradenie vstupov a výstupov. Základný význam majú PS, ktoré realizujú iba navzájom jednojednoznačné priradenie, tzn. *s jedným vstupom môže byť spojený iba jediný výstup.*

→ Najväčší význam pri tvorbe prepojovacích sietí majú *permutačné siete*, ktoré vo viacstupňovom vyhotovení umožňujú realizovať množstvo prepojovacích štruktúr so širokým spektrom vlastností, využívaných nielen pre potreby organizácie prenosu údajov, ale aj na vykonanie operácií zameraných na manipuláciu údajov.

→ Permutácia σ množiny X je bijekcia tejto množiny na samú seba.

$$\sigma: X \rightarrow X$$

Pozn.: Bijekcia je funkcia, ktorá je súčasne injektívna a surjektívna. Injektívna funkcia zobrazuje rôzne prvky svojho oboru na rôzne prvky svojho kooboru, inak povedané injekcia $S \rightarrow T$ je jednojednoznačné (prosté) zobrazenie množiny S do T . Ďalej funkcia je surjektívna ak obrazom jej oboru je celý koobor, inak povedané surjektívna $S \rightarrow T$ je zobrazenie množiny S na T . Takže môžeme povedať, že bijekcia je $S \rightarrow T$ jednojednoznačné zobrazenie množiny S na T .

Základné permutácie:

- permutácia dokonalého premiešania (σ) – perfect shuffle
- motýliková permutácia (β) – Butterfly
- permutácia reverzácie (ρ)
- permutácia výmeny (E) – exchange

Permutácia dokonalého premiešania:

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

Všeobecne je operácia premiešania definovaná v tvare:

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

kde K je „počet rúk“ potrebných na premiešame, pre ktoré platí $K = 2^k$,
 $k = 0, 1, \dots, n-1$.

Ak máme číslo x v binárnej reprezentácii, potom permutácia dokonalého premiešania zodpovedá cyklickému posuvu binárnej reprezentácie o jeden bit vľavo:

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

Permutácia dokonalého k -teho podpremiešania $\sigma_{(k)}(\mathbf{x})$ – je definovaná pre $1 \leq k \leq n$ nasledovne:

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

Takže platí:

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

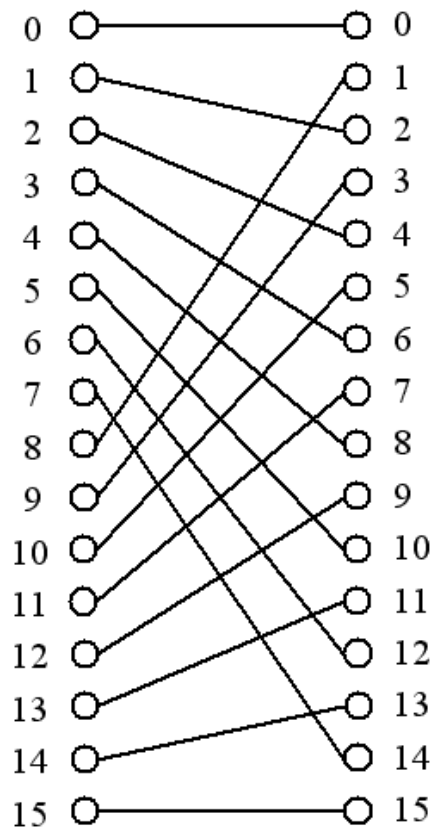
Napríklad pre $N=16$, $n=\log_2 N=4$:

$$\sigma_{(4)}(\mathbf{0}) = \sigma_{(4)}([0000]) = [0000] = \mathbf{0}$$

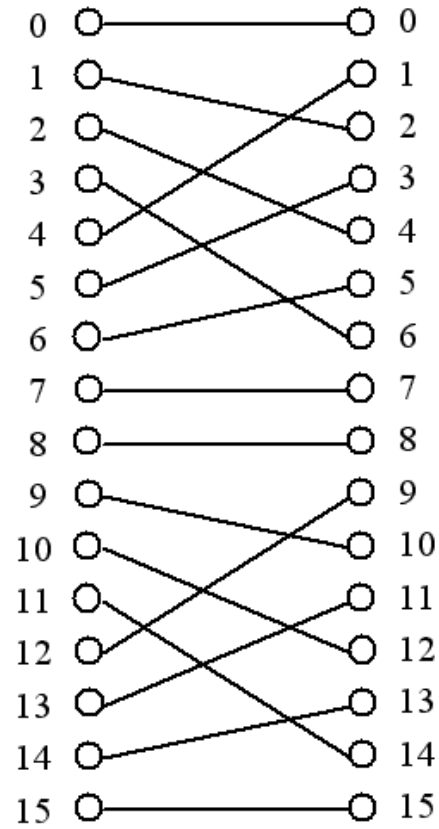
$$\sigma_{(4)}(\mathbf{1}) = \sigma_{(4)}([0001]) = [0010] = \mathbf{2}$$

$$\sigma_{(4)}(\mathbf{2}) = \sigma_{(4)}([0010]) = [0100] = \mathbf{4}$$

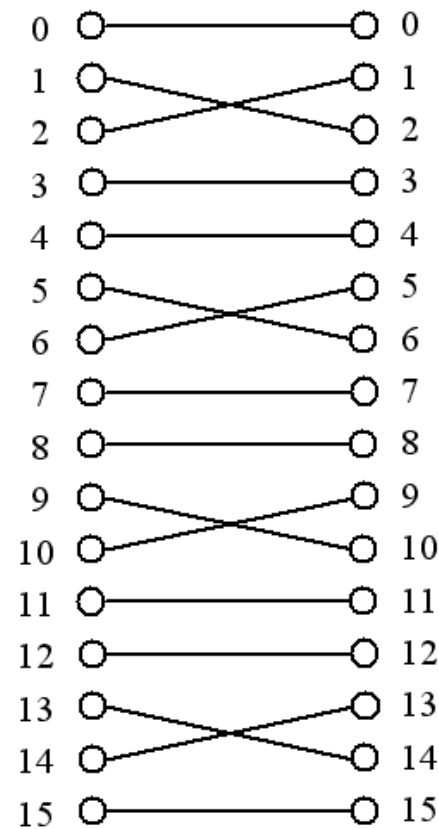
$$\sigma_{(4)}(\mathbf{3}) = \sigma_{(4)}([0011]) = [0110] = \mathbf{6} \text{ atd'}$$



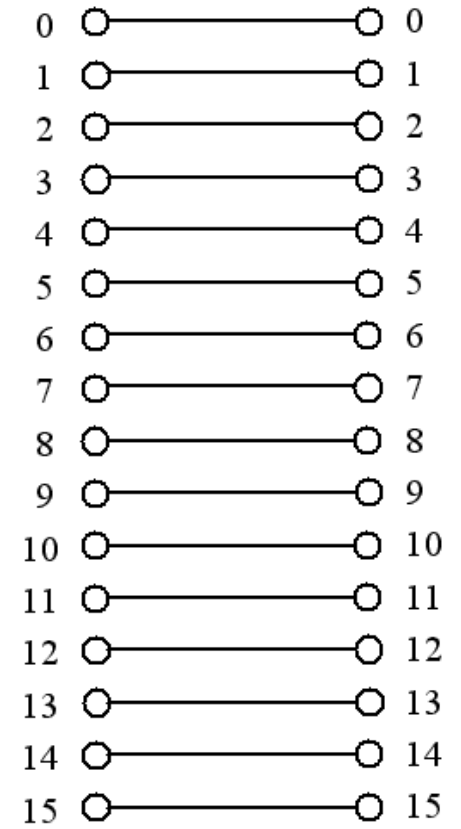
$$\sigma_{(4)}(\mathbf{x}) \equiv \sigma(\mathbf{x})$$



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



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



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

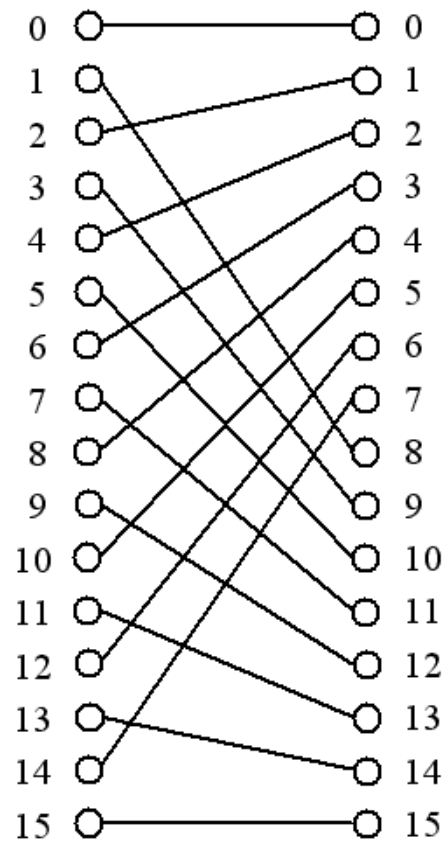
Inverzné dokonalé premiešanie zodpovedá cyklickému posuvu binárnej reprezentácie o jeden bit vpravo:

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

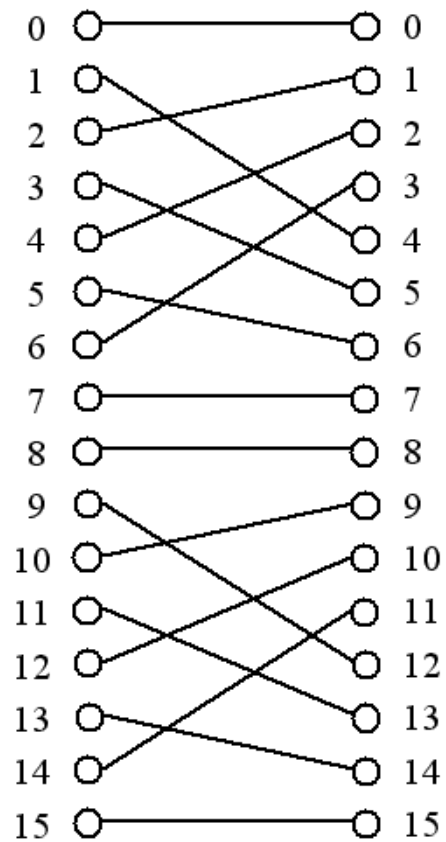
Analogicky, permutácia k-teho inverzného dokonalého podpremiešania $\sigma_{(k)}^{-1}(\mathbf{x})$ je definovaná pre $1 \leq k \leq n$ nasledovne:

$$\sigma_{(k)}^{-1}(\mathbf{x}) = \sigma_{(k)}^{-1}([x_n \ x_{n-1} \ x_{n-2} \ \dots \ x_2 \ x_1]) = [x_n \ \dots \ x_1 \ x_k \ x_{k-1} \ \dots \ x_2]$$

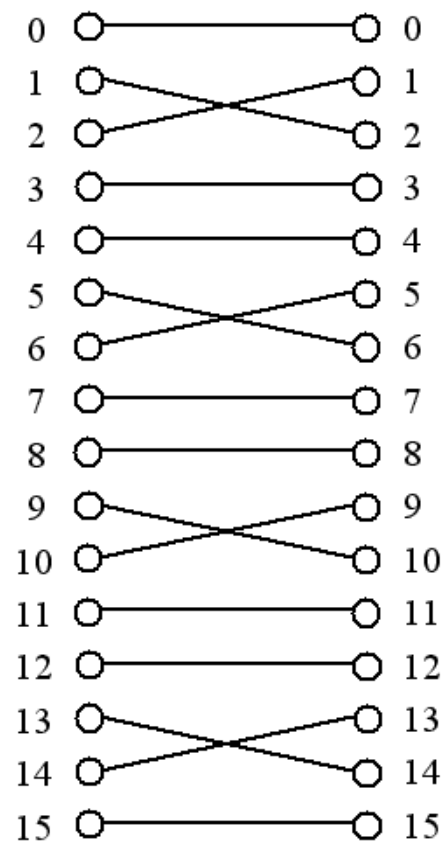
Napríklad pre $N=16$, $n=\log_2 N=4$:



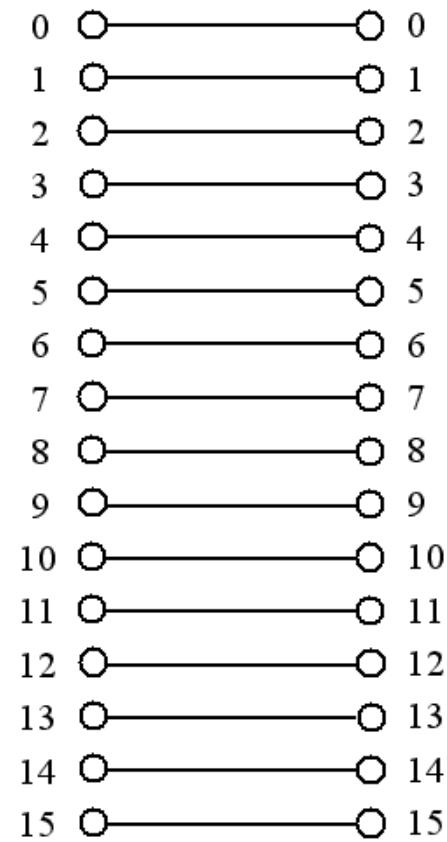
$$\sigma_{(4)}^{-1}(\mathbf{x}) \equiv \sigma^{-1}(\mathbf{x})$$



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



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



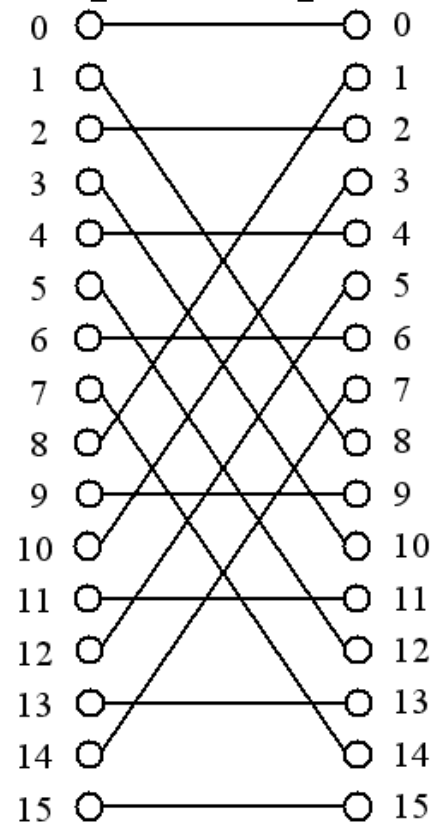
$$\sigma_{(1)}^{-1}(\mathbf{x}) \equiv \mathbf{x}$$

Motýliková permutácia (Butterfly):

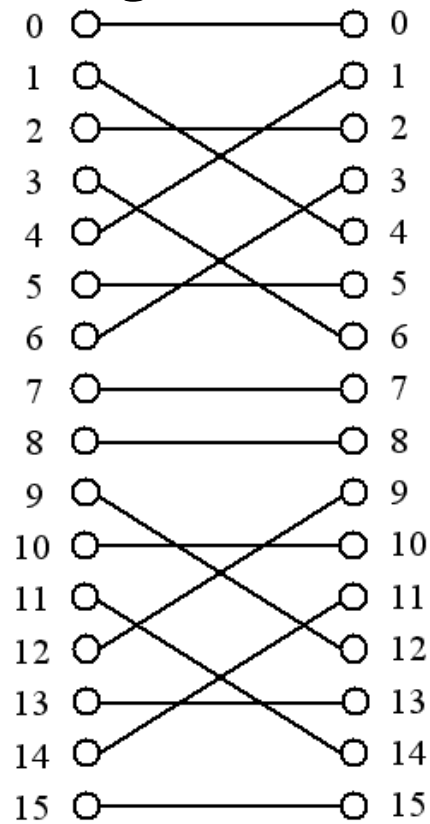
k -ta motýliková permutácia $\beta_{(k)}(\mathbf{x})$ pre $1 \leq k \leq n$, $n = \log_2 N$ sa dostane tak, že sa vzájomne vymenia prvý a k -ty bit:

$$\beta_{(k)}(\mathbf{x}) = \beta_{(k)}([\mathbf{x}_n \ \mathbf{x}_{n-1} \ \mathbf{x}_{n-2} \ \dots \ \mathbf{x}_2 \ \mathbf{x}_1]) = [\mathbf{x}_n \ \mathbf{x}_{n-1} \ \dots \ \mathbf{x}_{k+1} \ \mathbf{x}_1 \ \mathbf{x}_{k-1} \ \dots \ \mathbf{x}_2 \ \mathbf{x}_k]$$

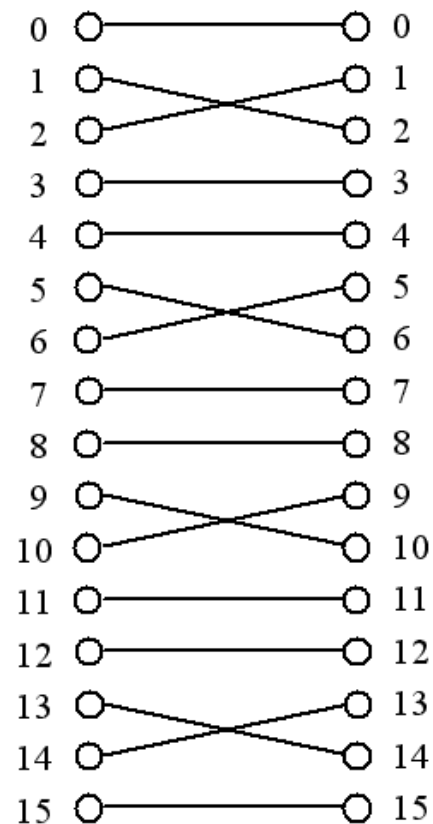
Napríklad pre $N=16$, $n = \log_2 N = 4$:



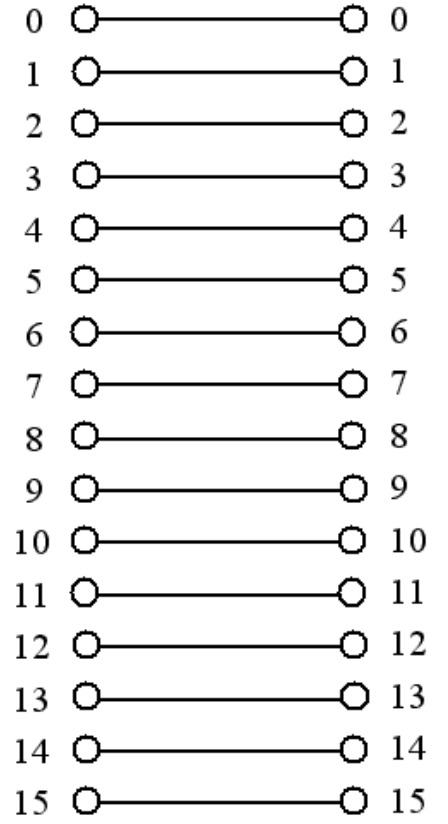
$$\beta_{(4)}(\mathbf{x}) \equiv \beta_{(4)}^{-1}(\mathbf{x})$$



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



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



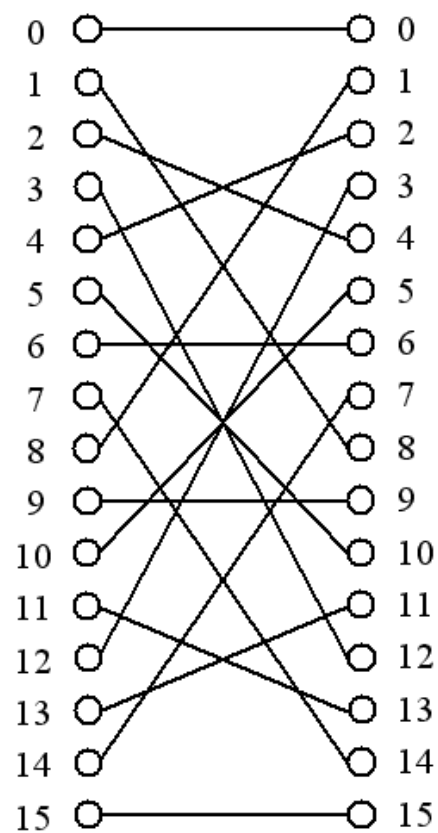
$$\beta_{(1)}(\mathbf{x}) \equiv \beta_{(1)}^{-1}(\mathbf{x})$$

Permutácia reverzácie:

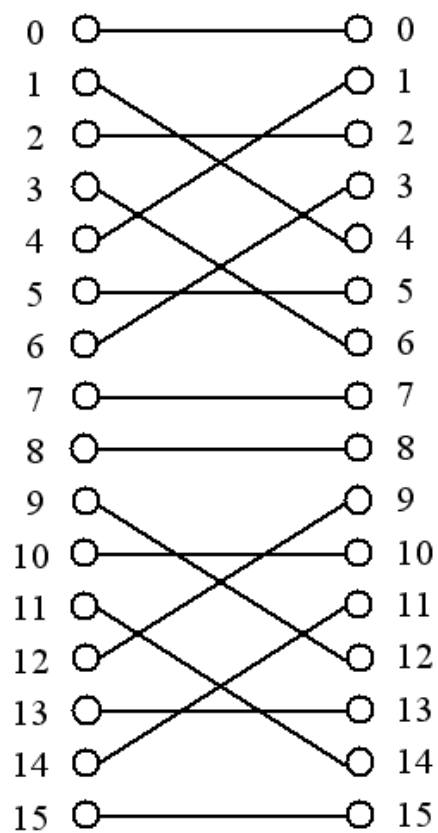
je daná vzájomnou výmenou bitov vyšších a nižších (zrkadlenie):

$$\rho(\mathbf{x}) = \rho([\mathbf{x}_n \ \mathbf{x}_{n-1} \ \dots \ \mathbf{x}_2 \ \mathbf{x}_1]) = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_{n-1} \ \mathbf{x}_n]$$

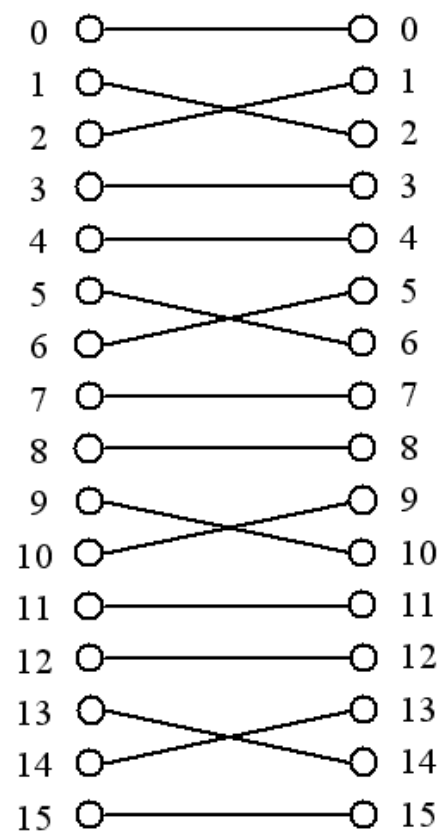
Napríklad pre $N=16$, $n=\log_2 N=4$:



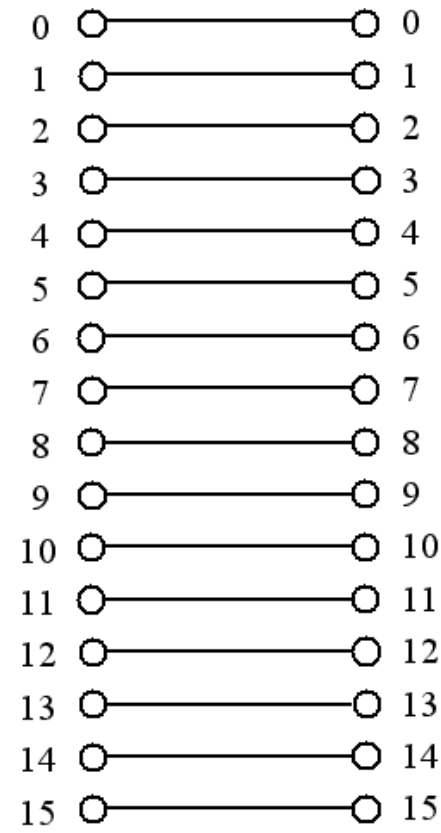
$\rho = \rho_4$



ρ_3



ρ_2

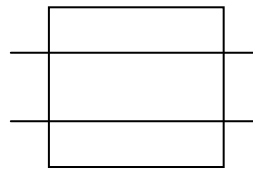


$\rho_1 = \mathbf{x}$

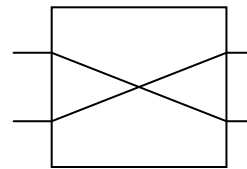
Permutácia výmeny:

Definujme \hat{x} vzt'ahom $\hat{x} = [x_n \ x_{n-1} \ \dots \ x_2 \ \bar{x}_1]$, kde \bar{x}_1 značí inverziu prvého (najnižšieho) bitu. Potom množina výmenných permutácií je:

bud' $e(x) = x$ a $e(\hat{x}) = \hat{x}$,
alebo $e(x) = \hat{x}$ a $e(\hat{x}) = x$.

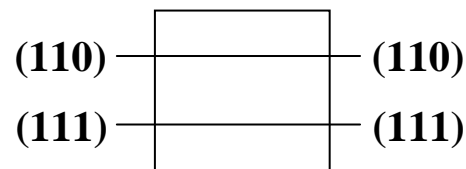


Straight-trough
(priamo)

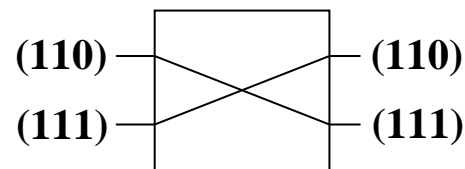


Exchange
(výmena)

Napr.



priamo



výmena

Základné permutačné prepojovacie siete:

Omega sieť

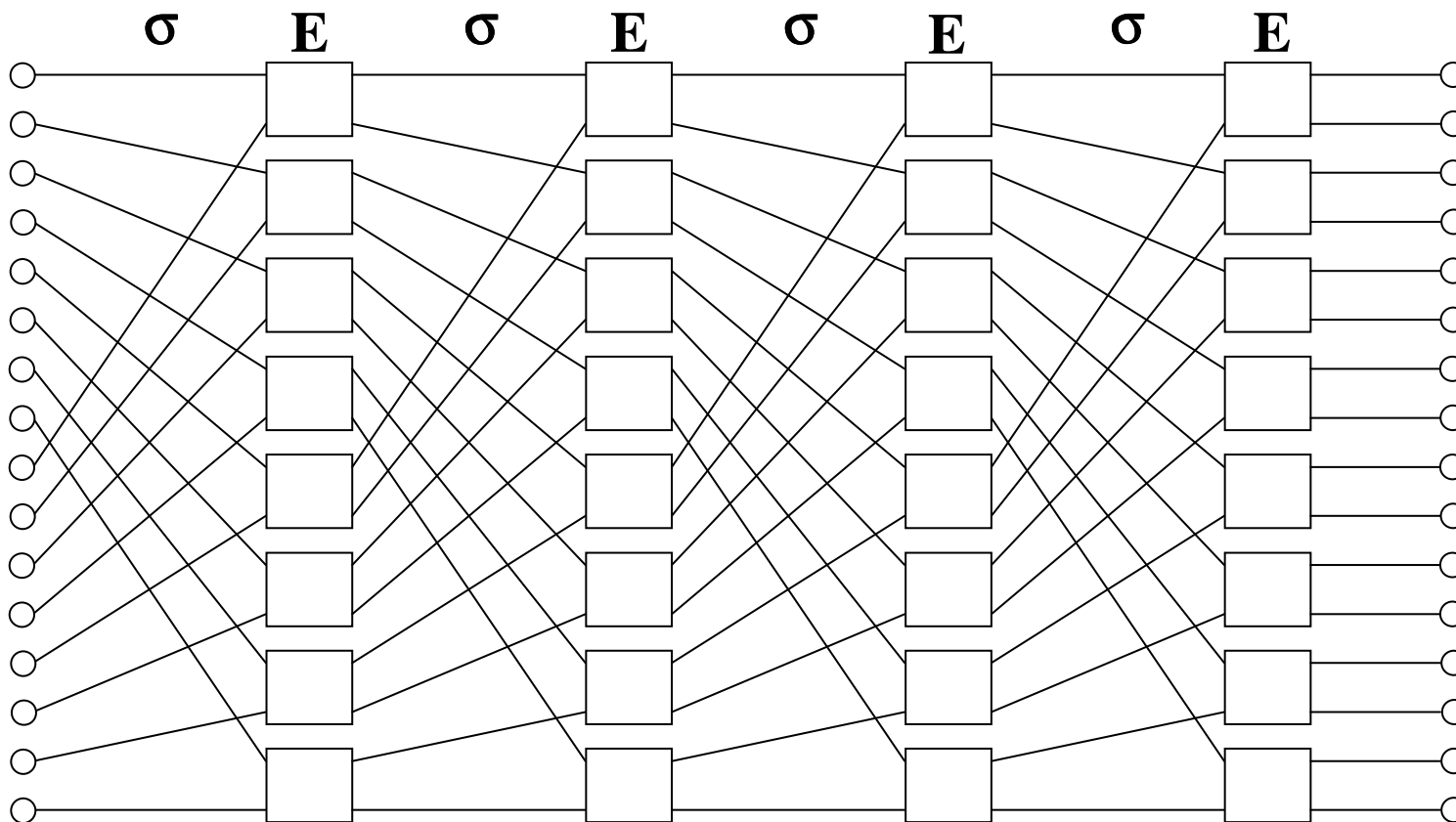
Omega sieť je založená na dvoch základných permutáciách: na permutácii dokonalého premiešania σ a permutácii výmeny E .

$$\Omega_N = \sigma E \sigma E \dots \sigma E = (\sigma E)^n$$

$$n = \log_2 N$$

Napr. pre $N=16$, $n=4$:

$$\Omega_{16} = (\sigma E)^4$$

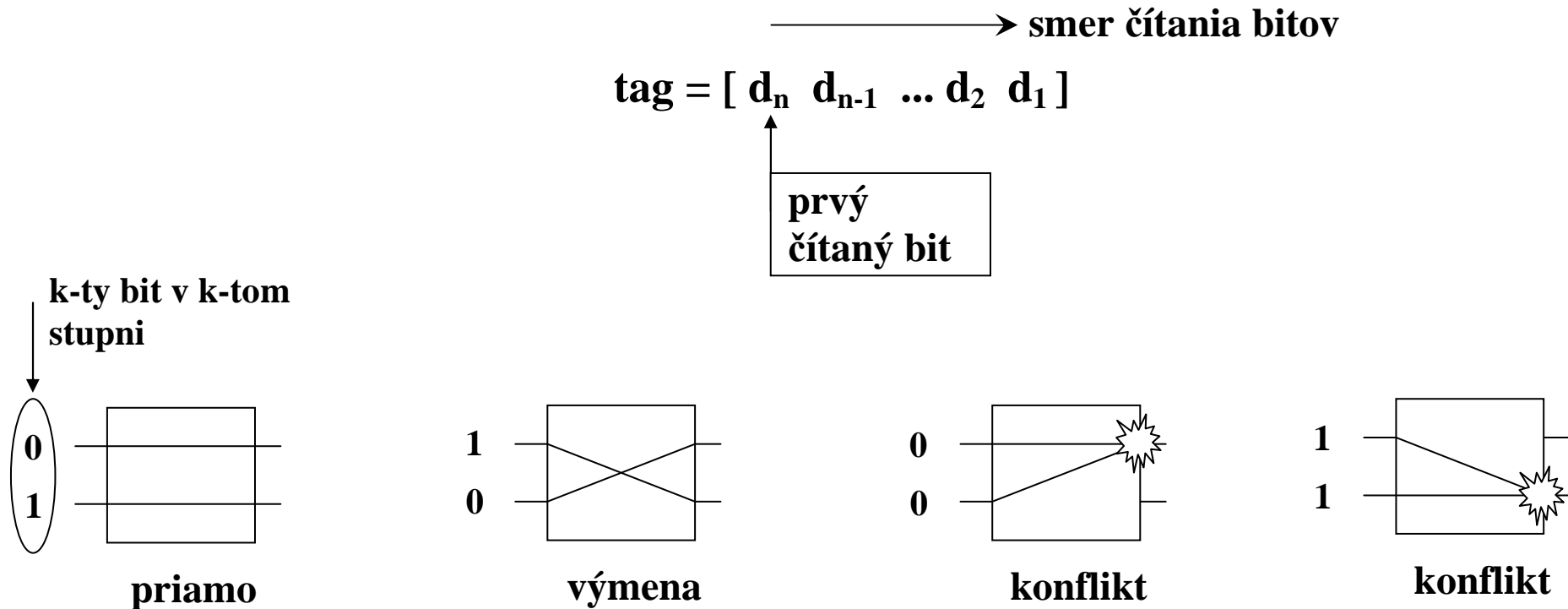


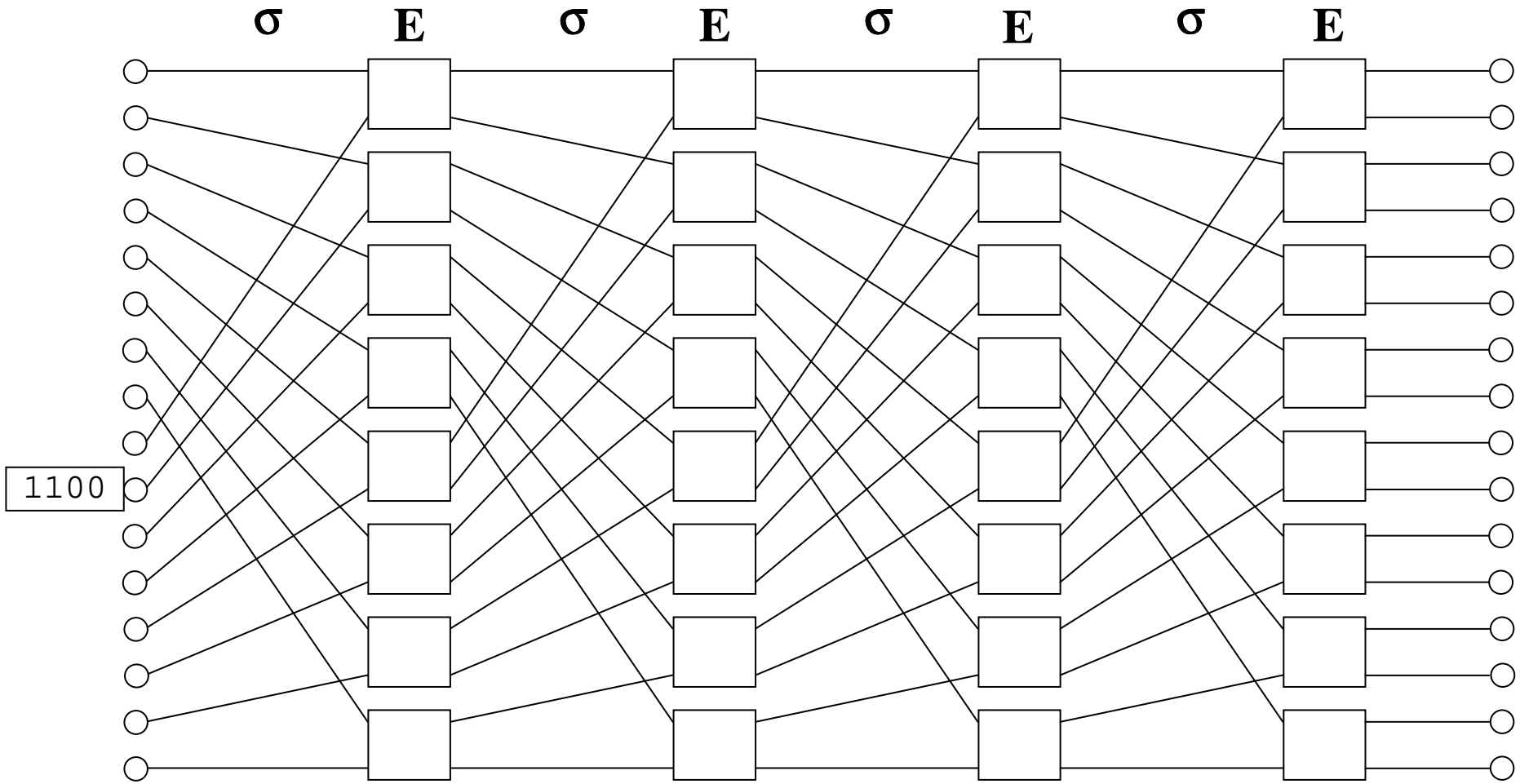
Samosmerovací algoritmus:

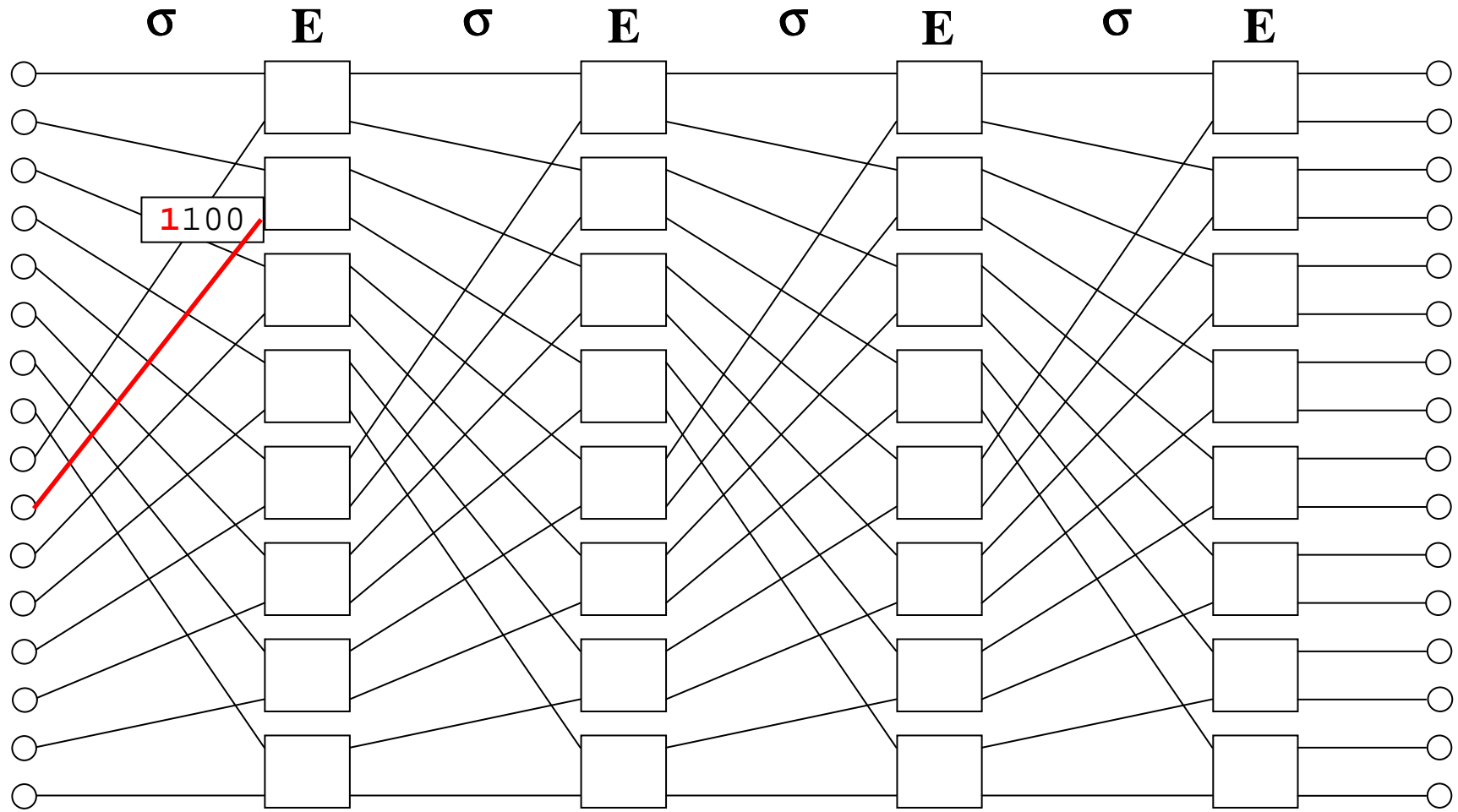
Na základe adresy určenia, t.j. binárnej reprezentácie čísla výstupného portu (routing tag = destination port id) je možné riadiť nastavenie prepínačov v jednotlivých stupňoch siete nasledovne. Prepínač v k-tom stupni siete číta k-ty bit smerovacieho návestia (začínajúc od MSB smerom k LSB) a ak sa tento rovná:

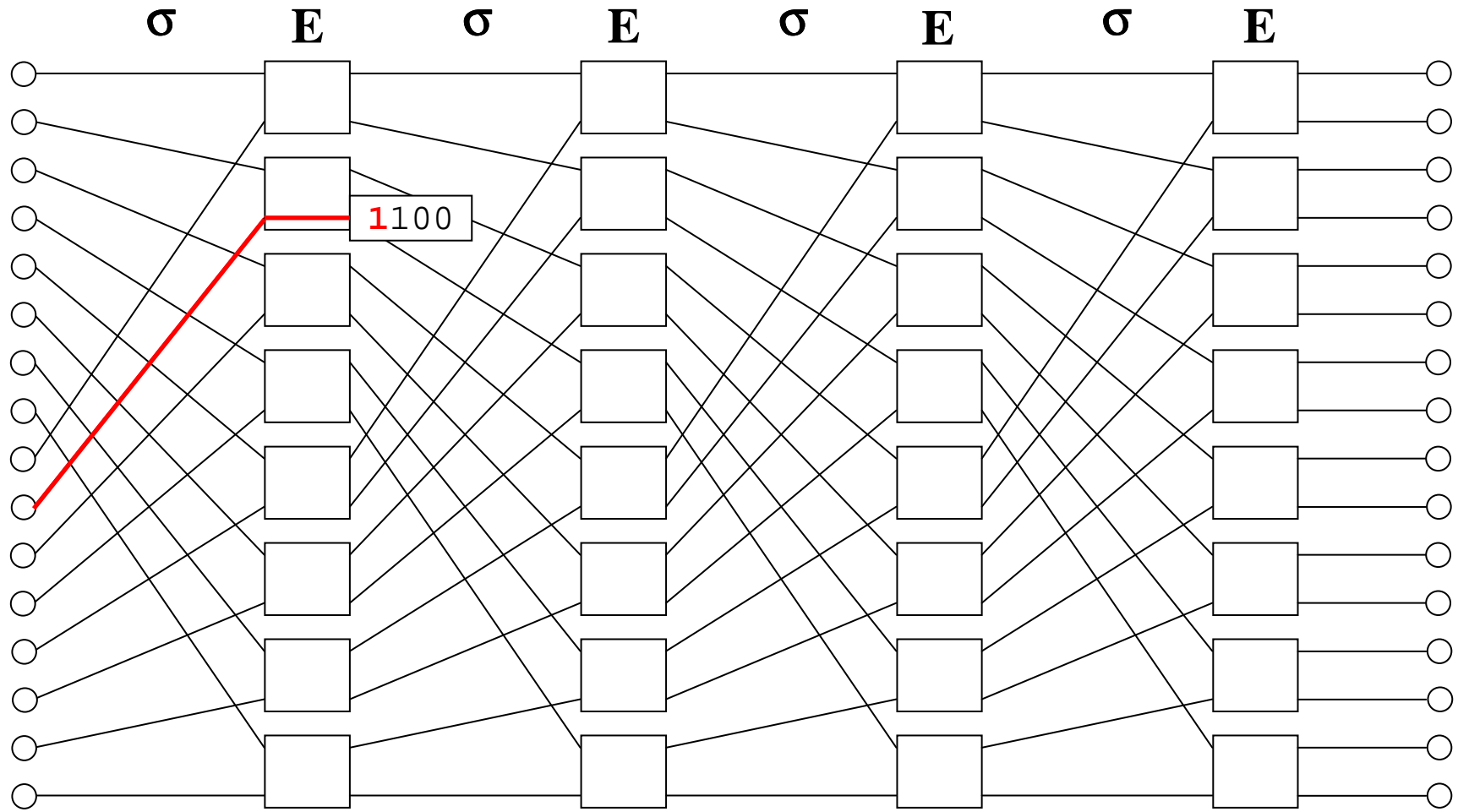
0 potom chod' na horný výstup

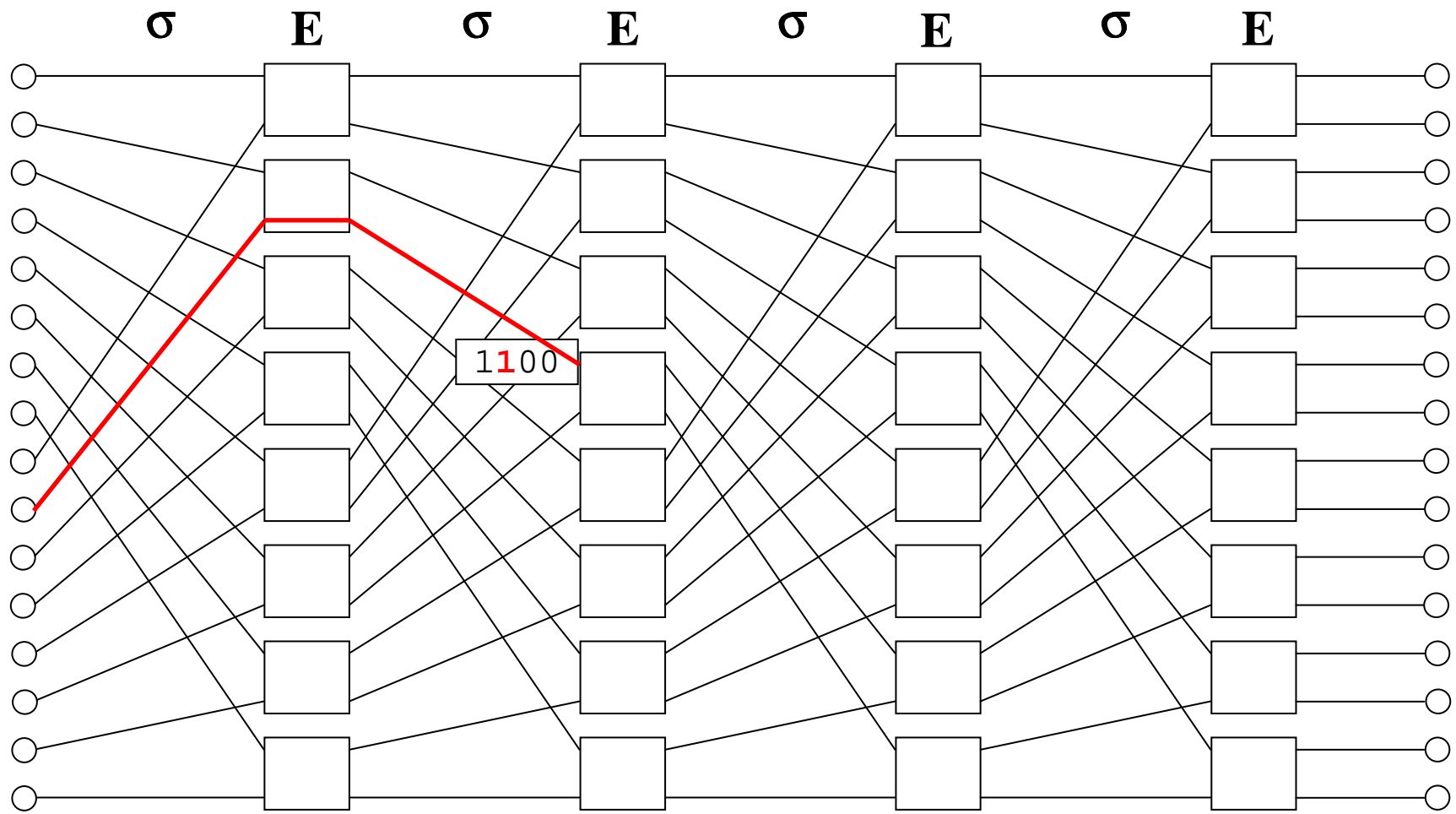
1 potom chod' na dolný výstup

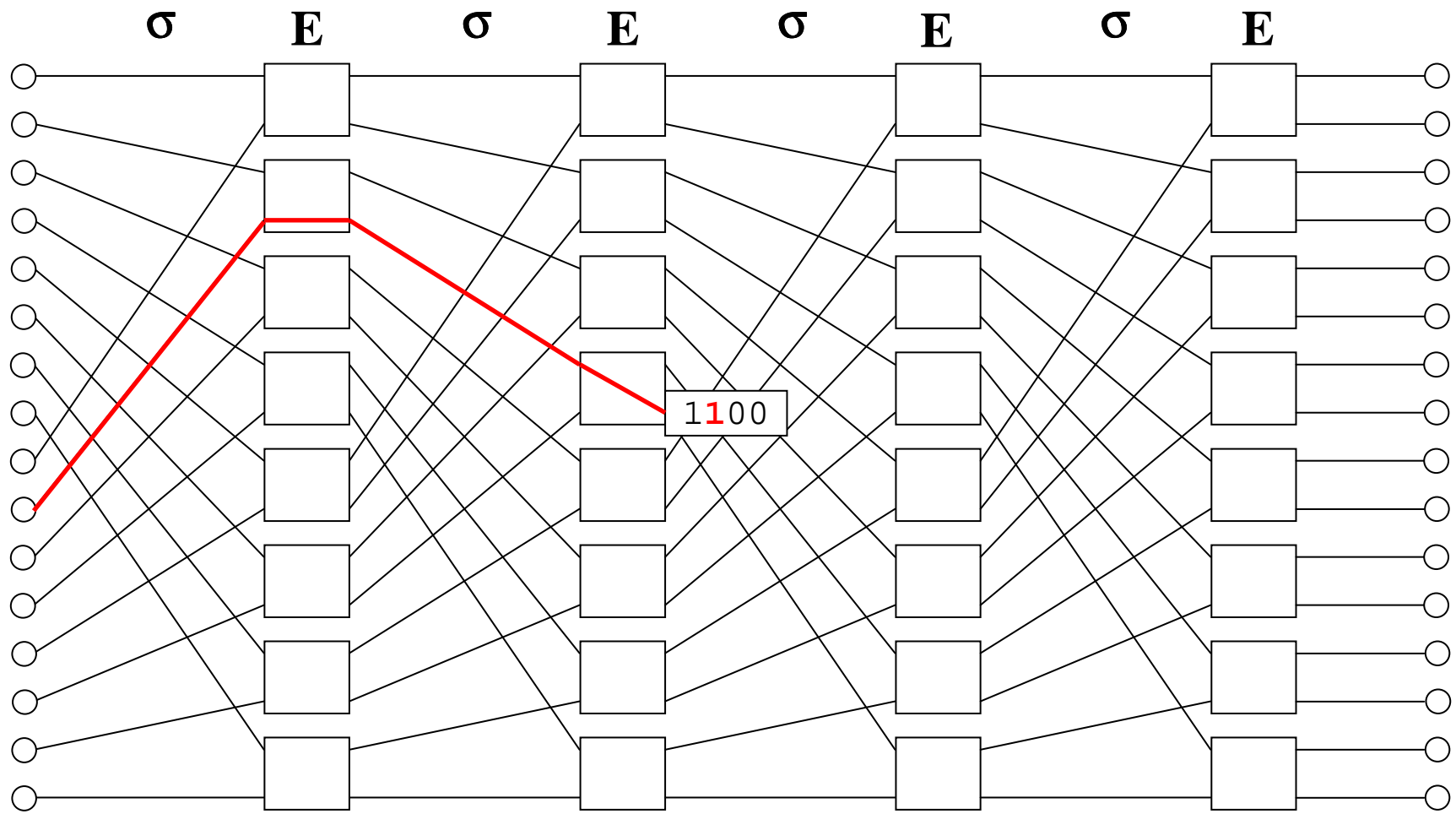


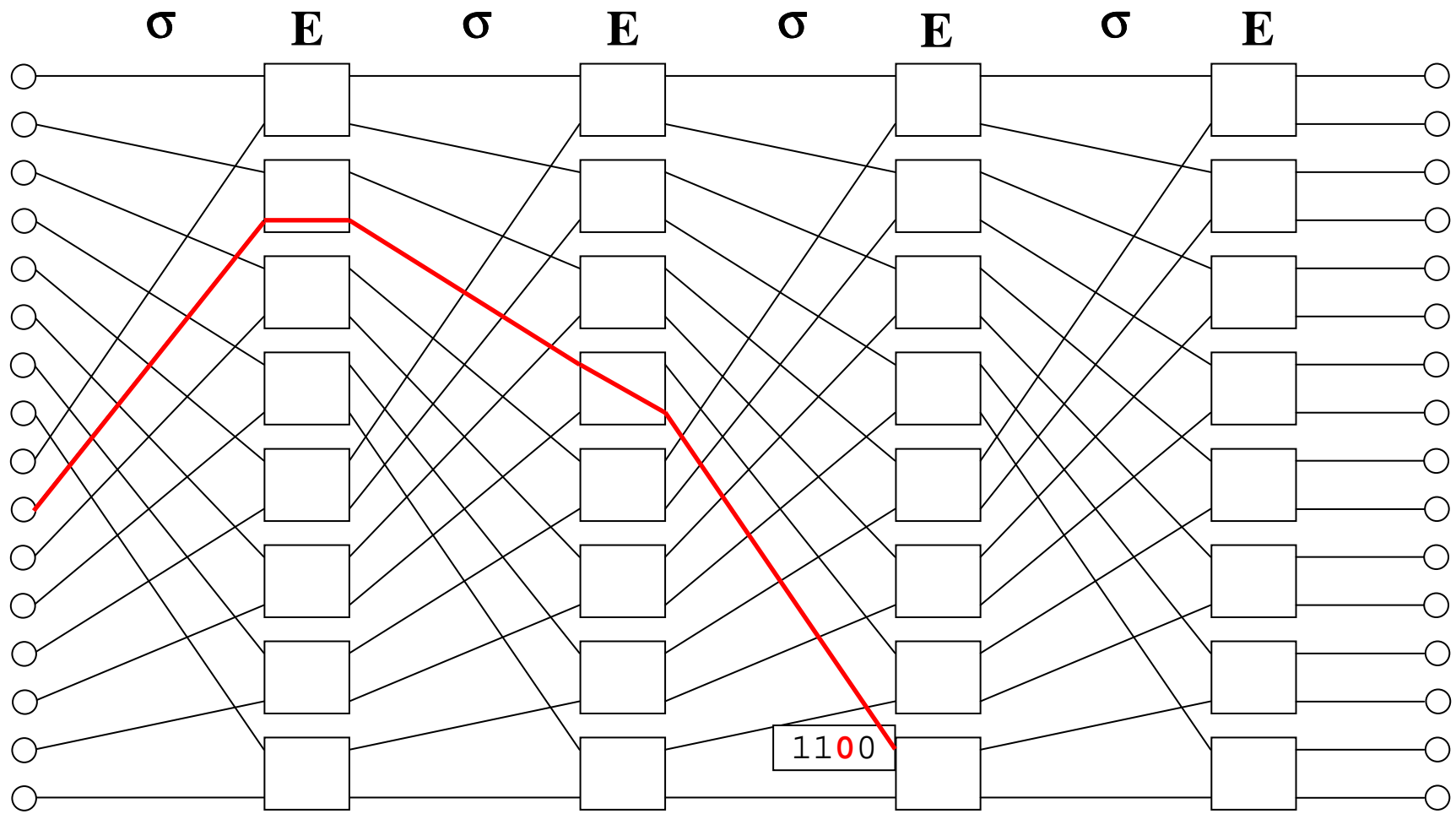


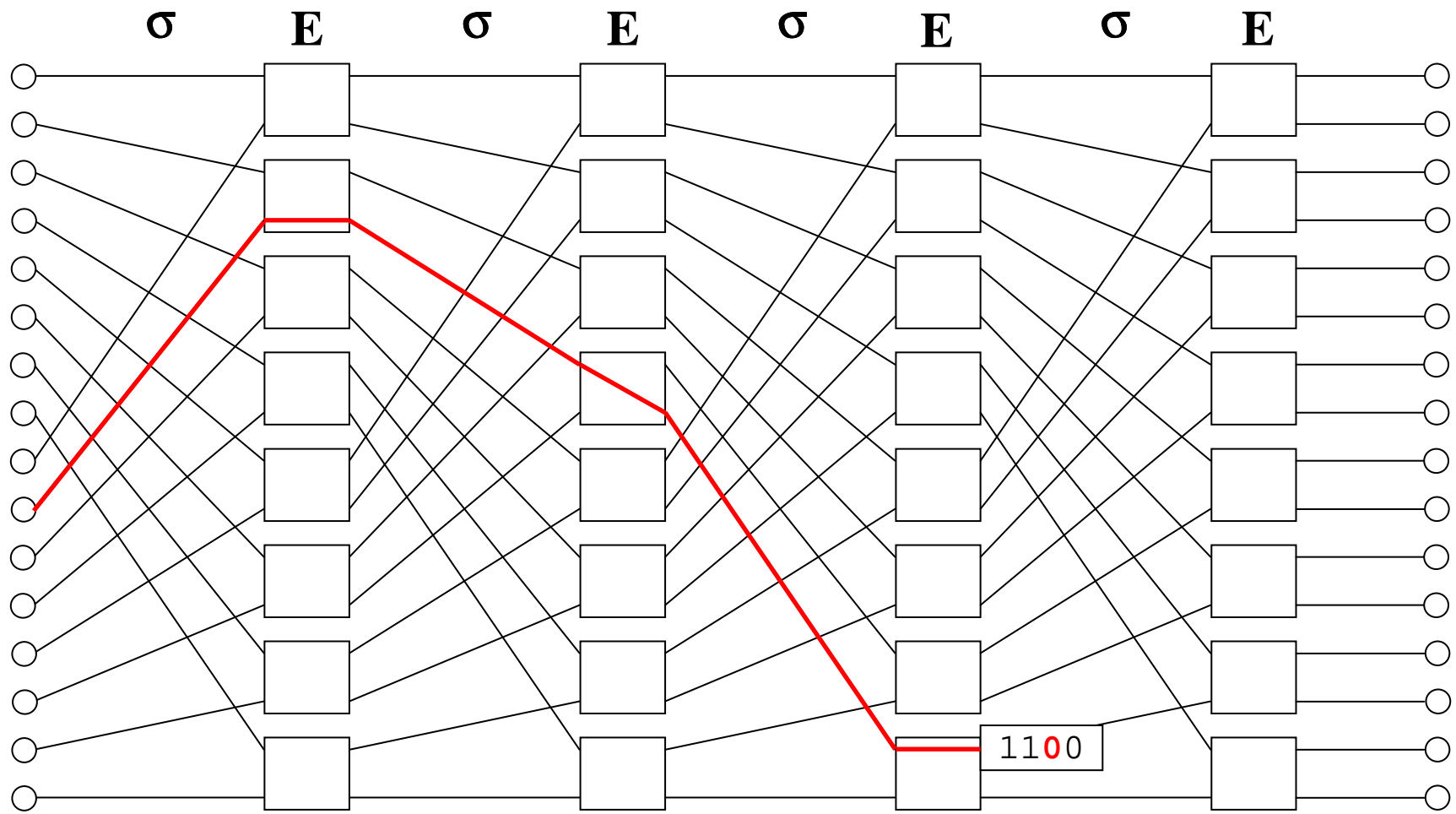


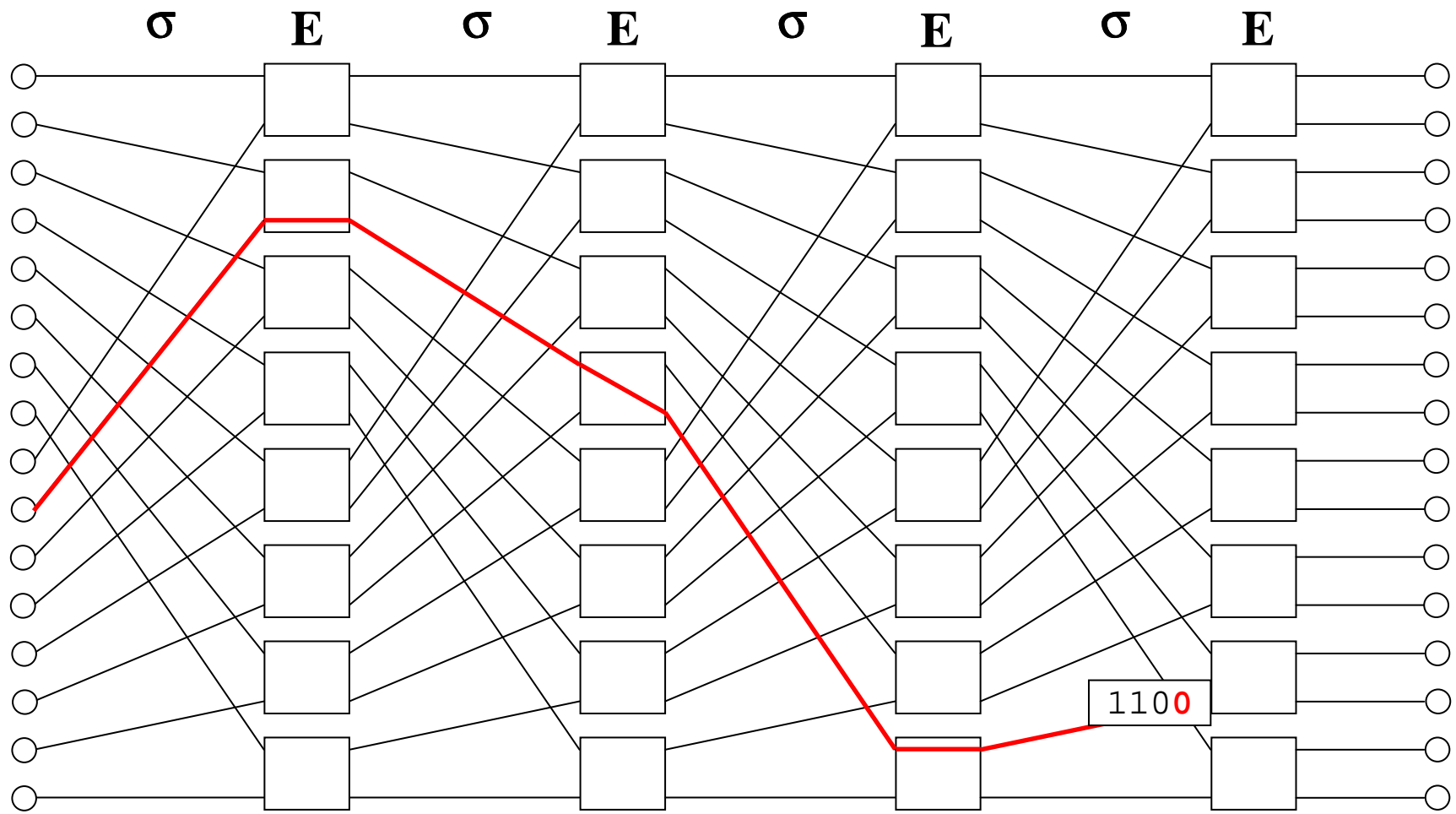


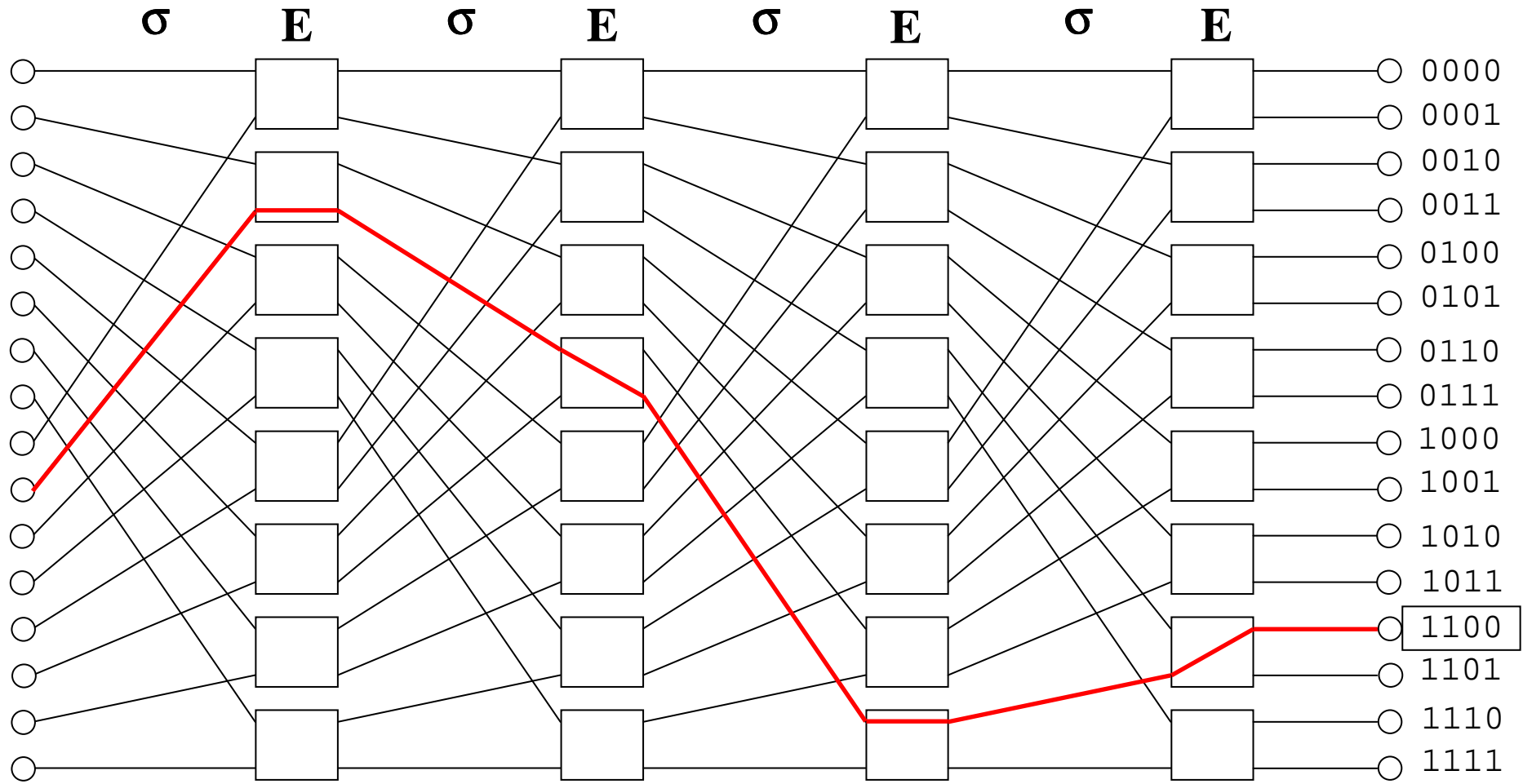


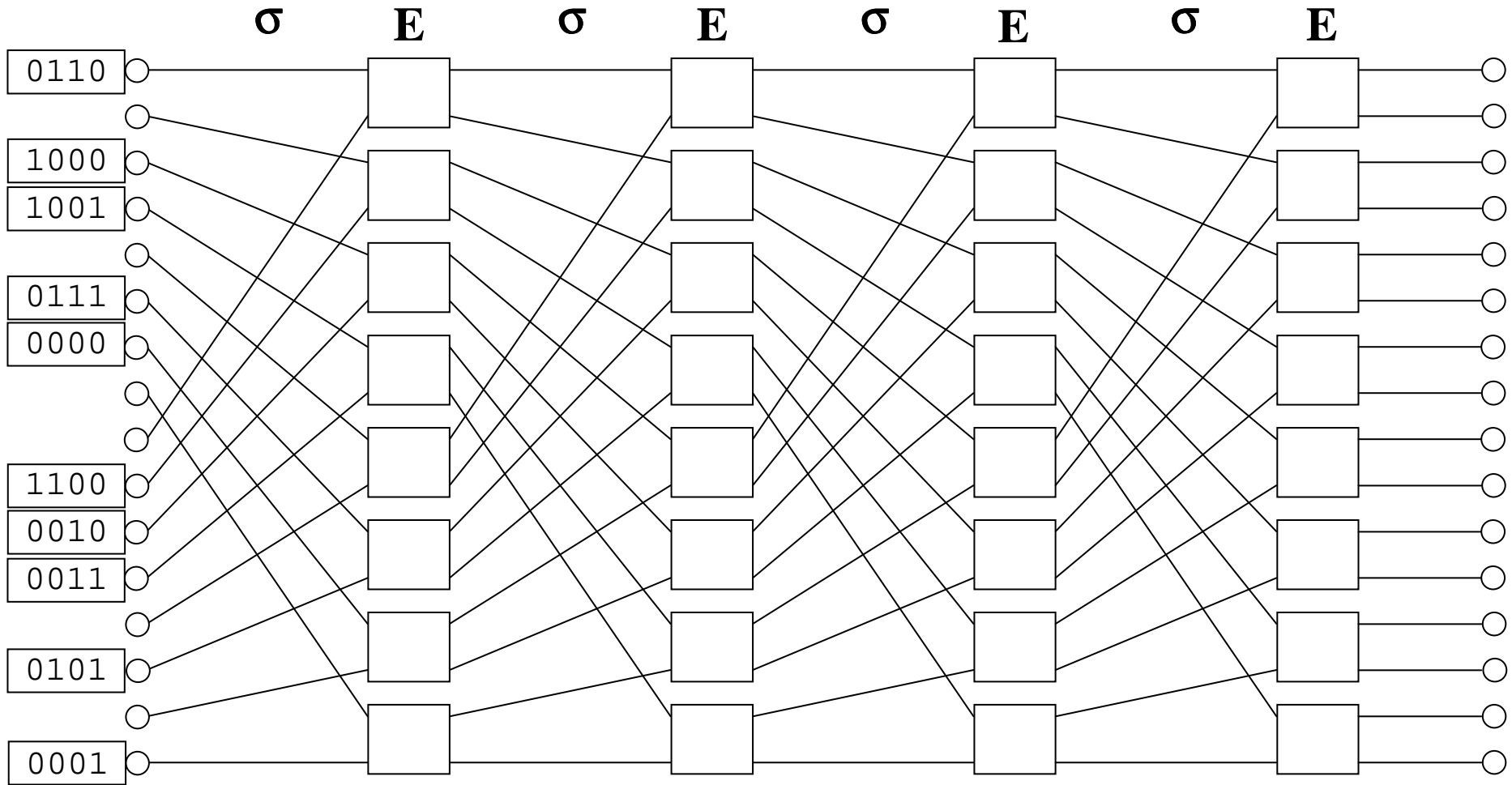


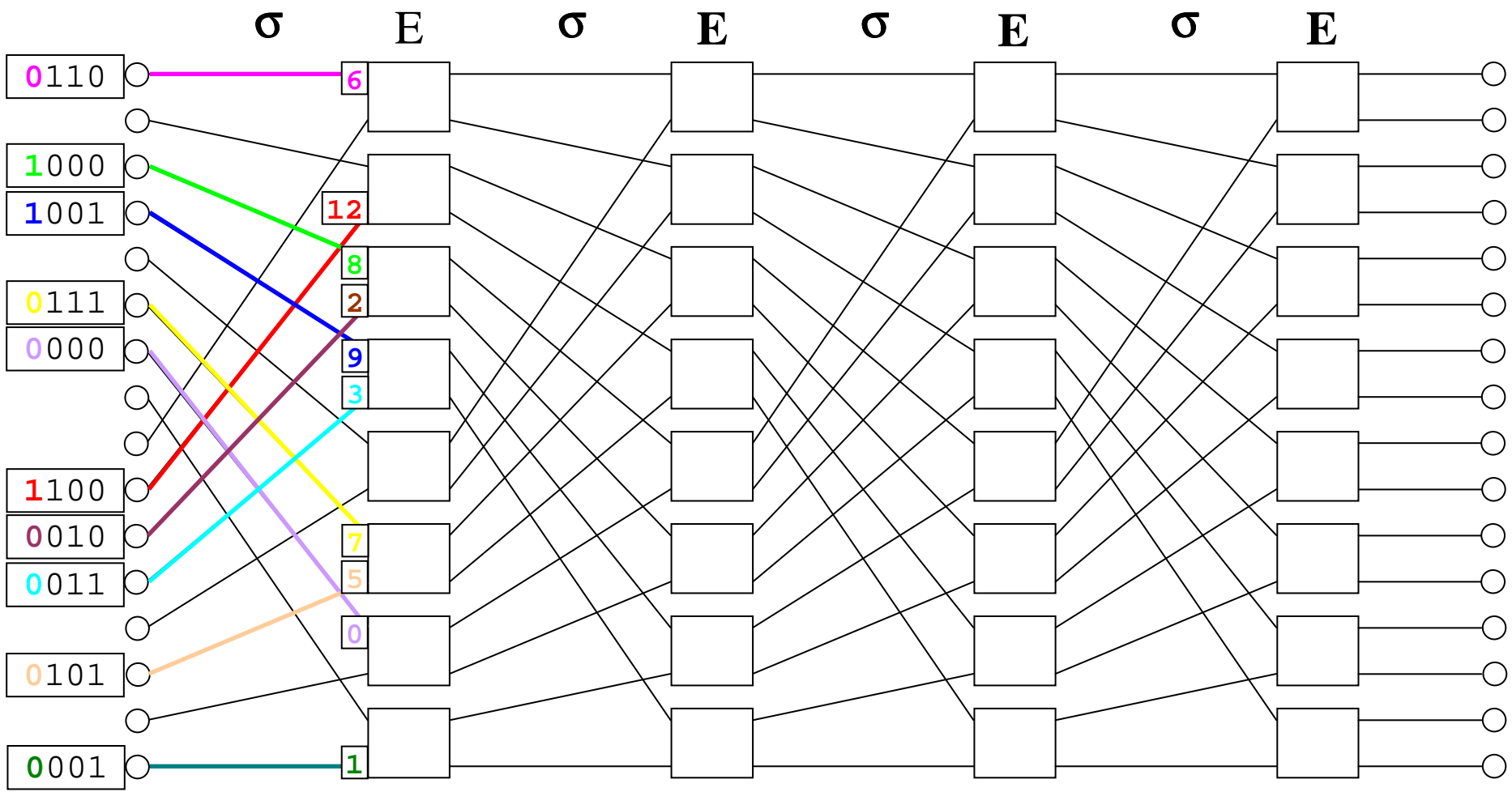


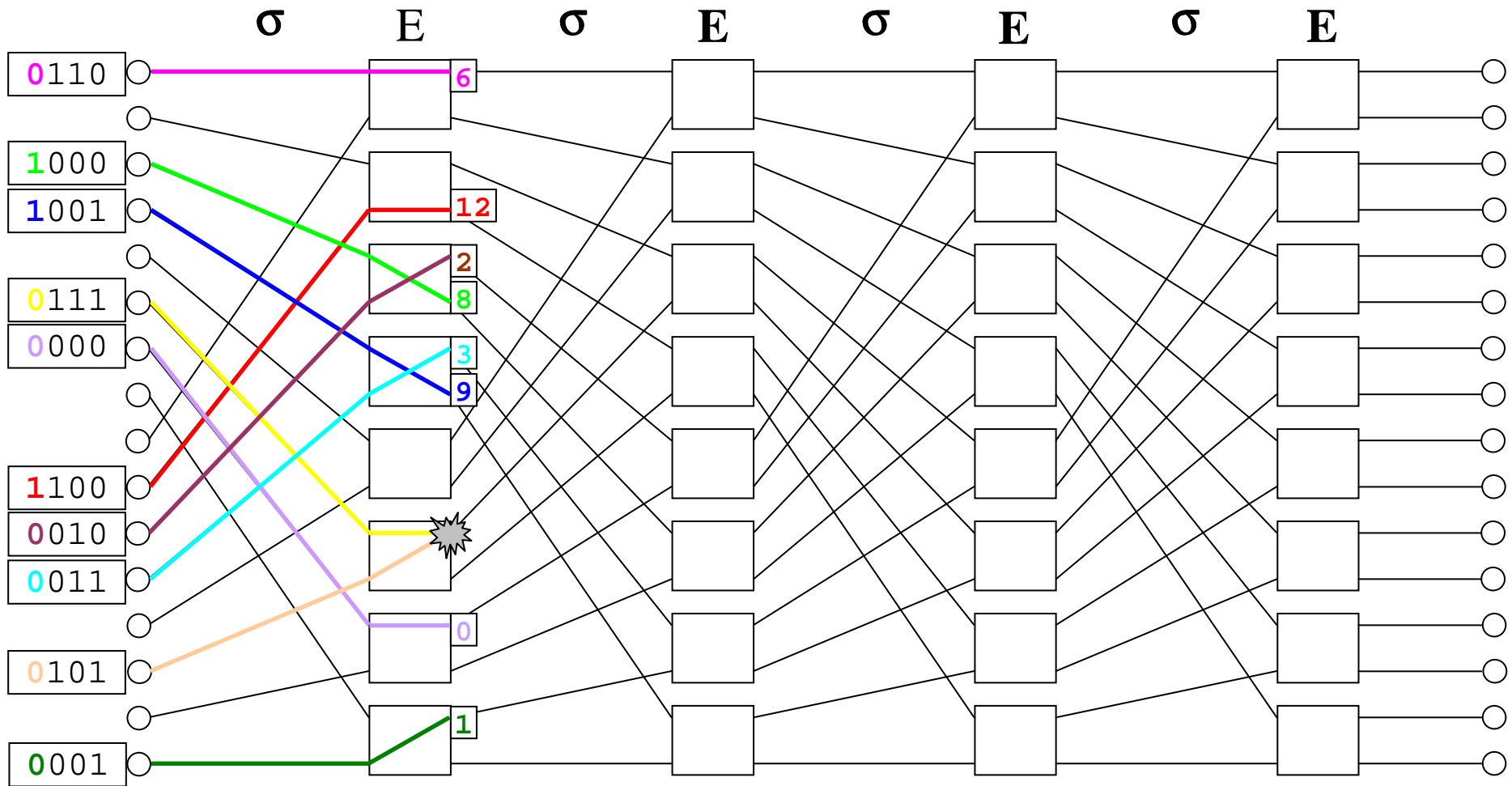


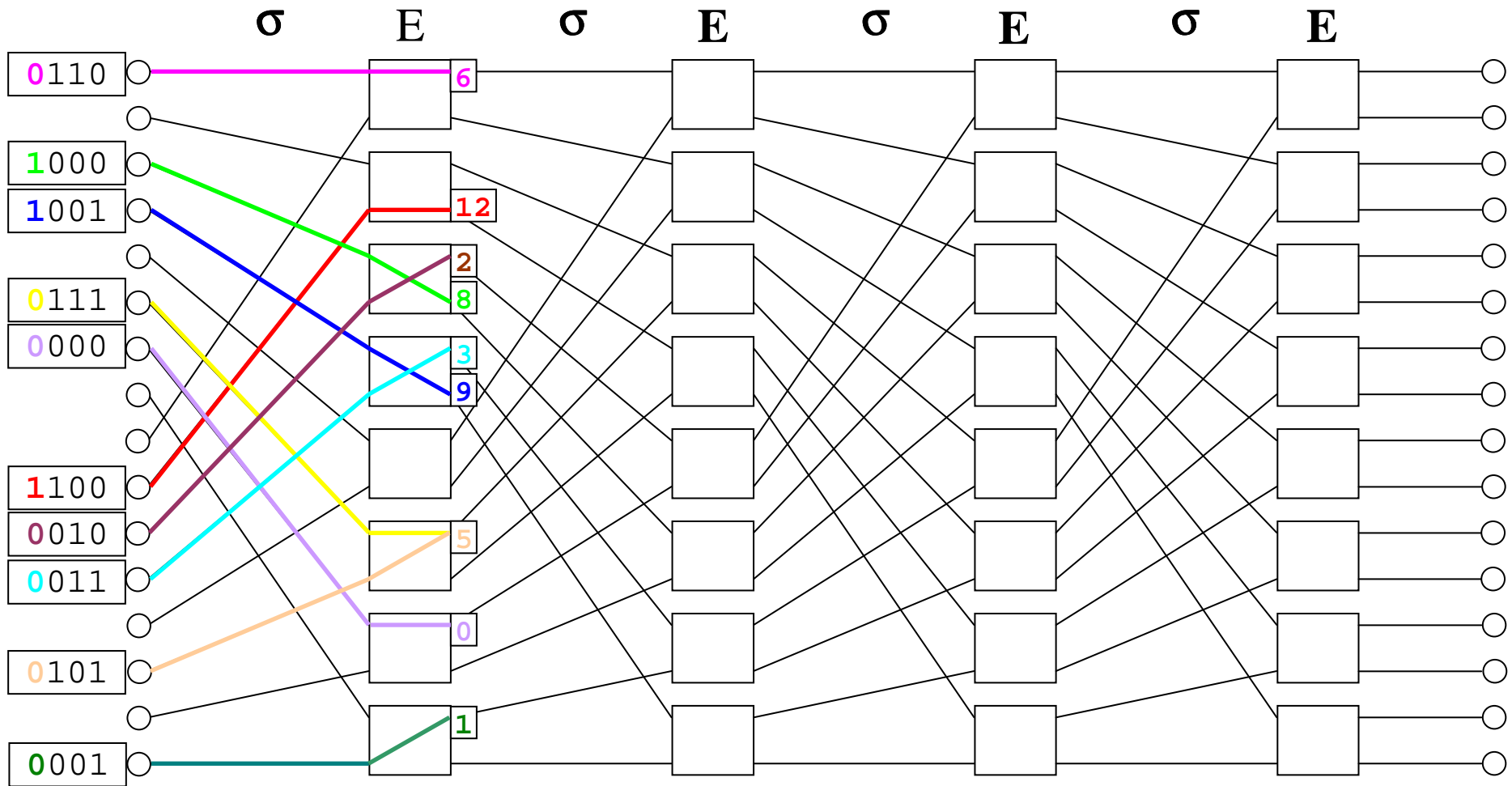


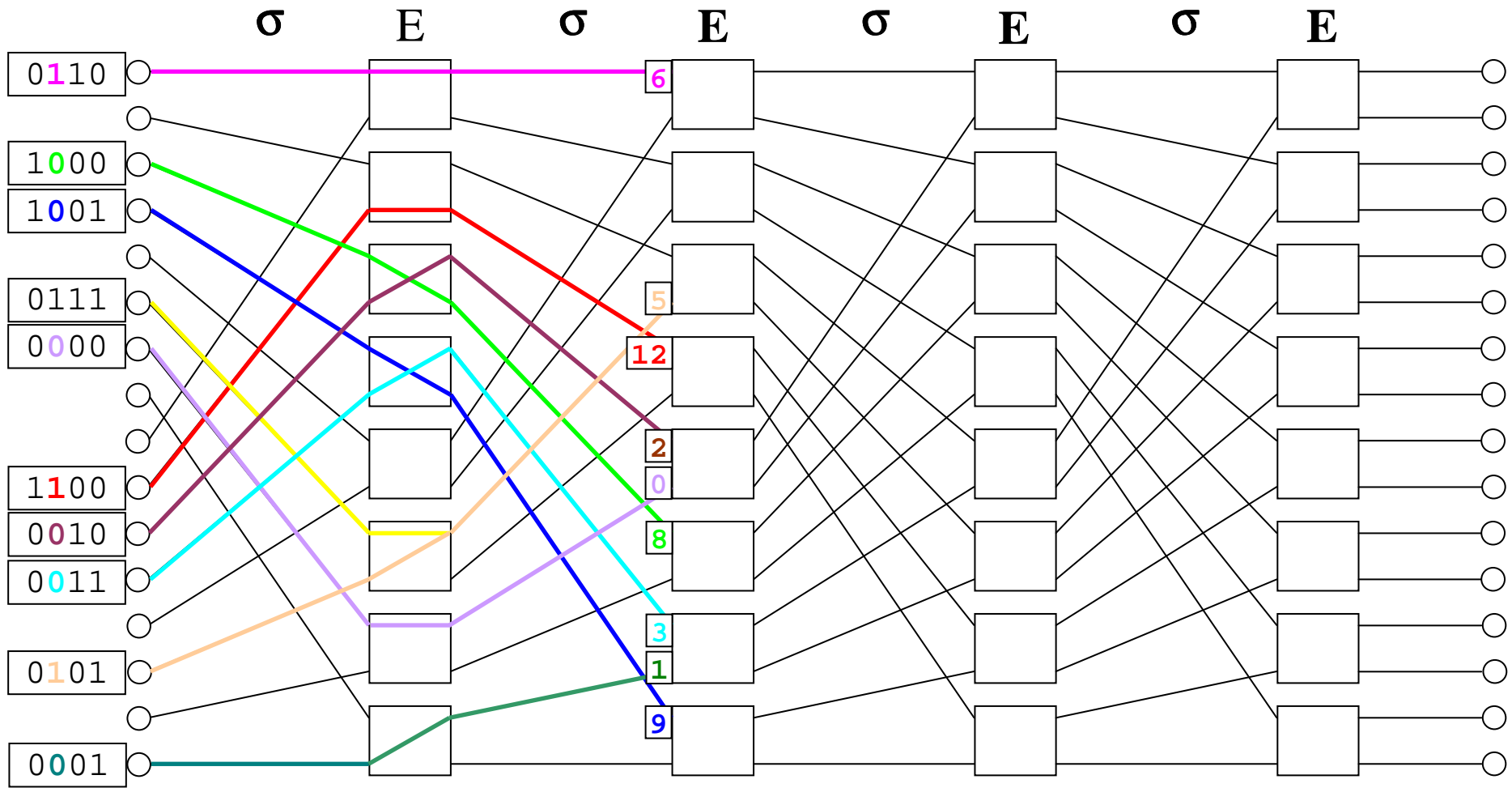


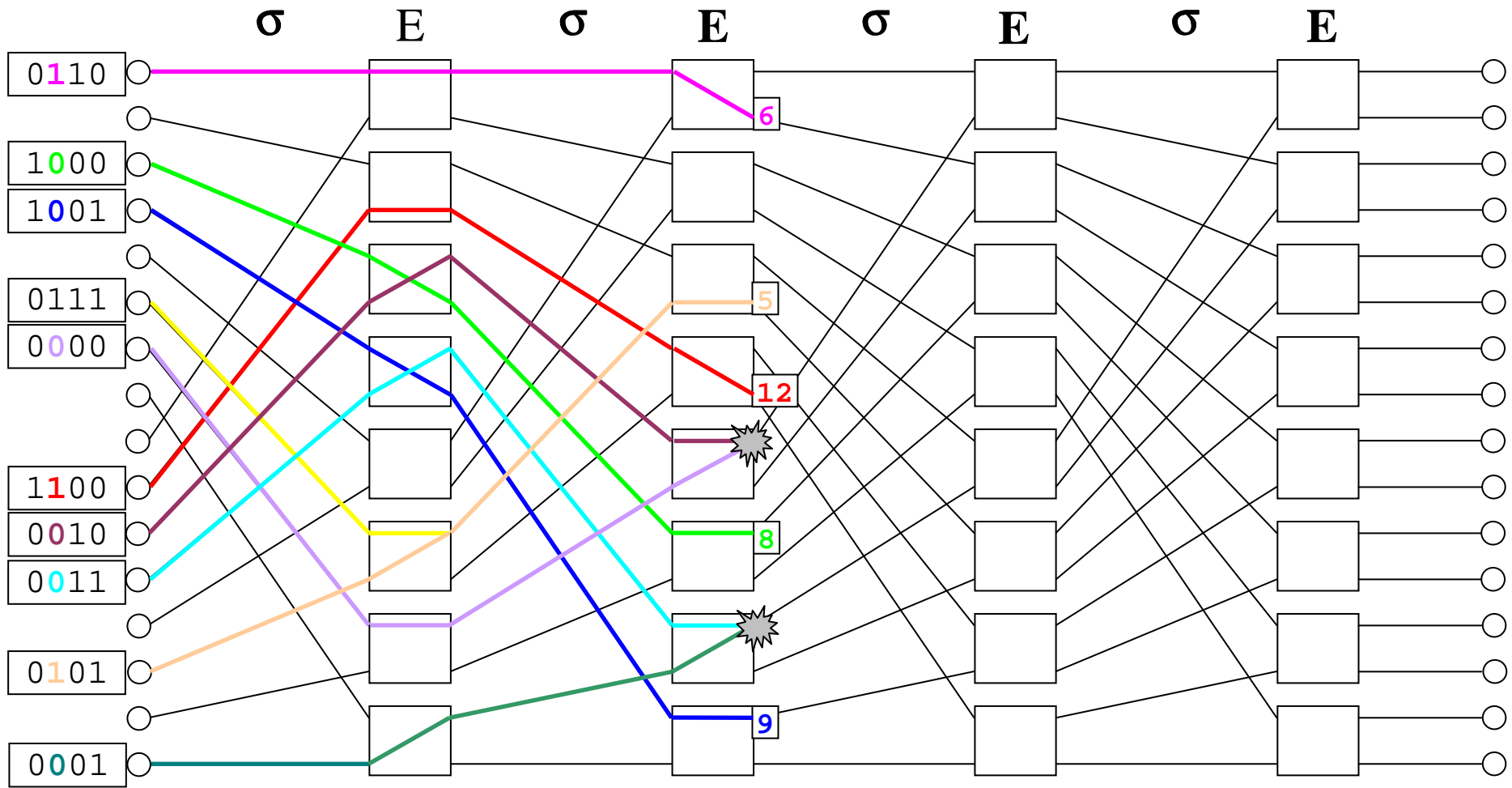


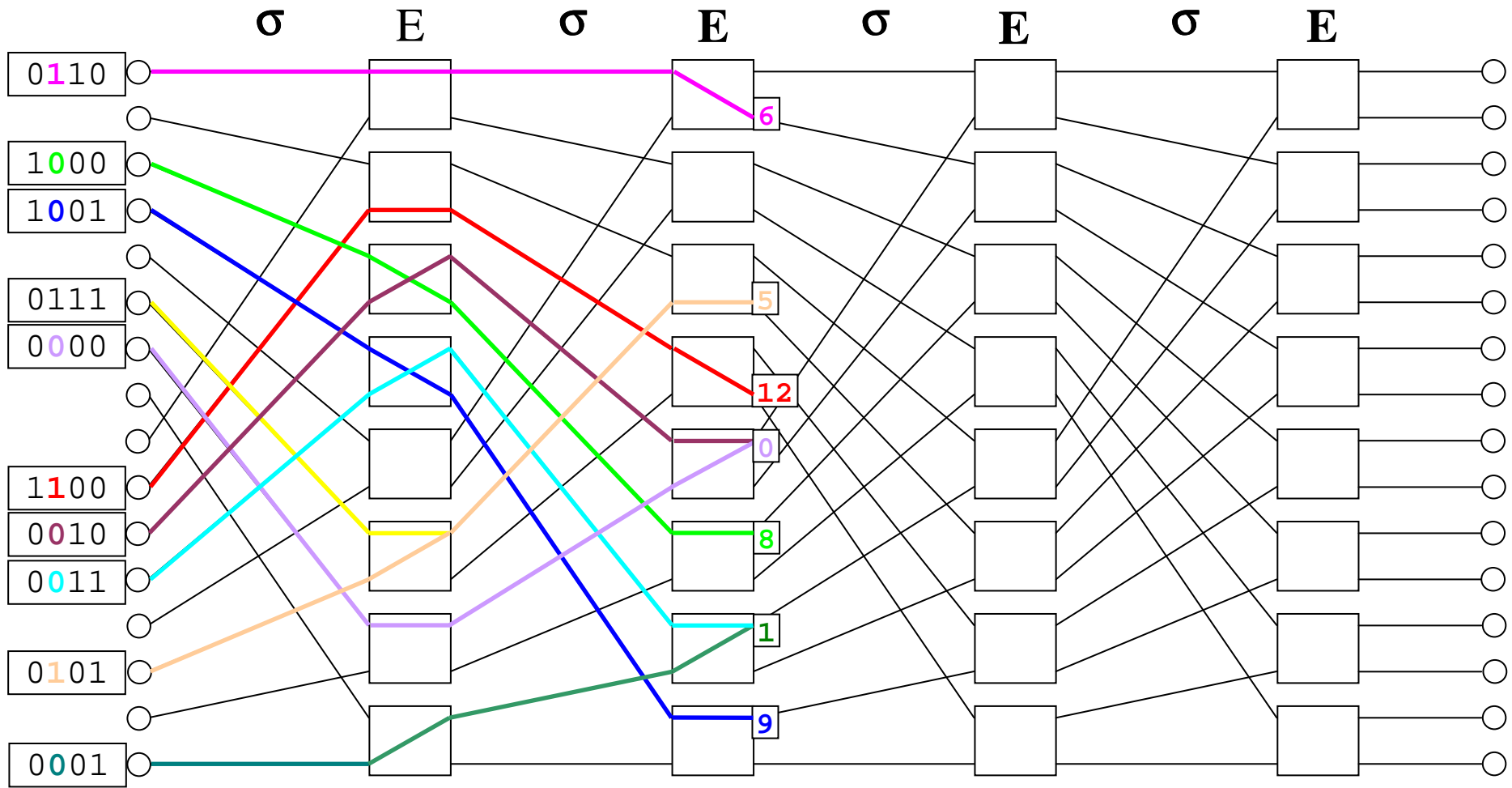


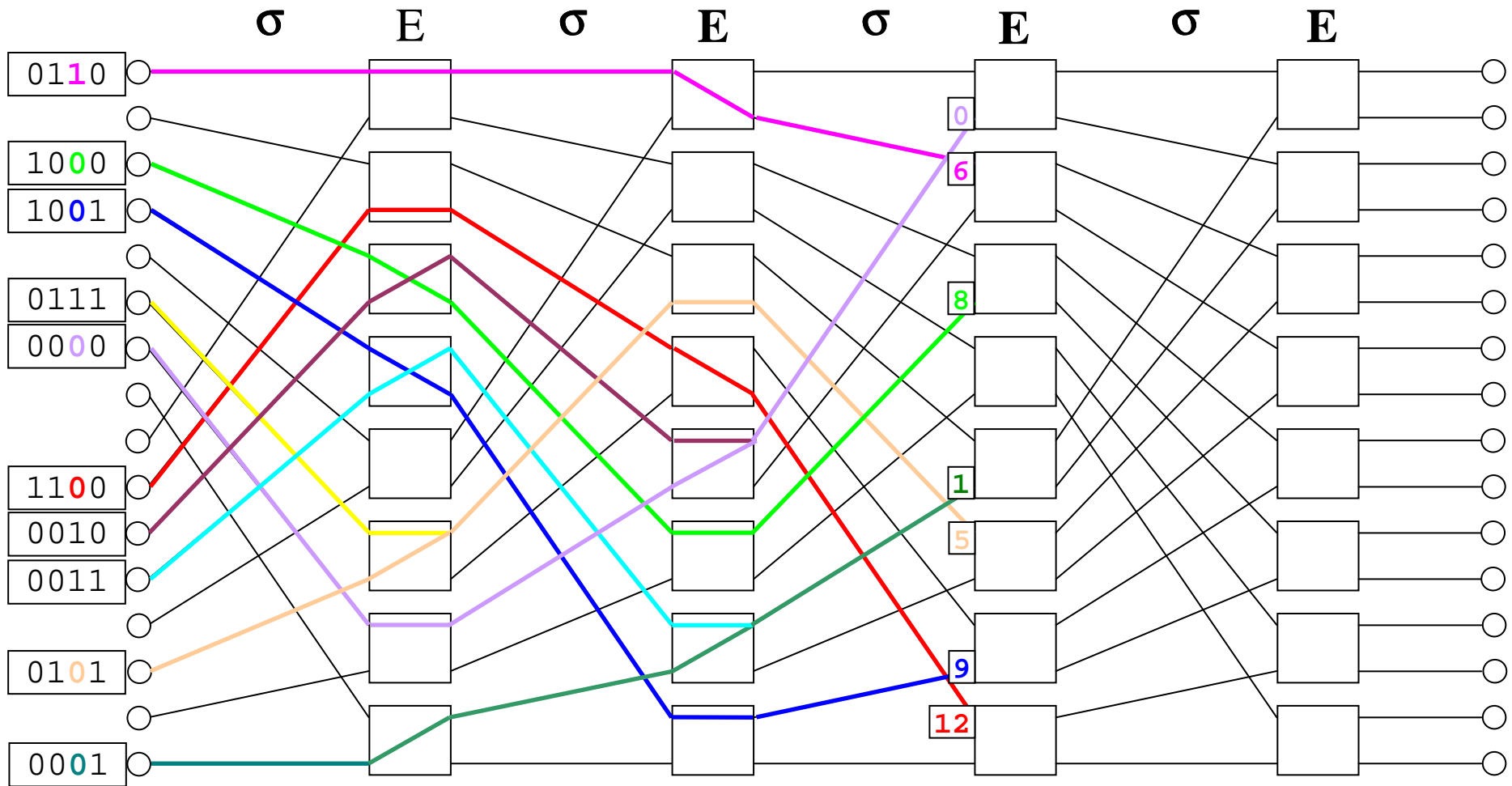


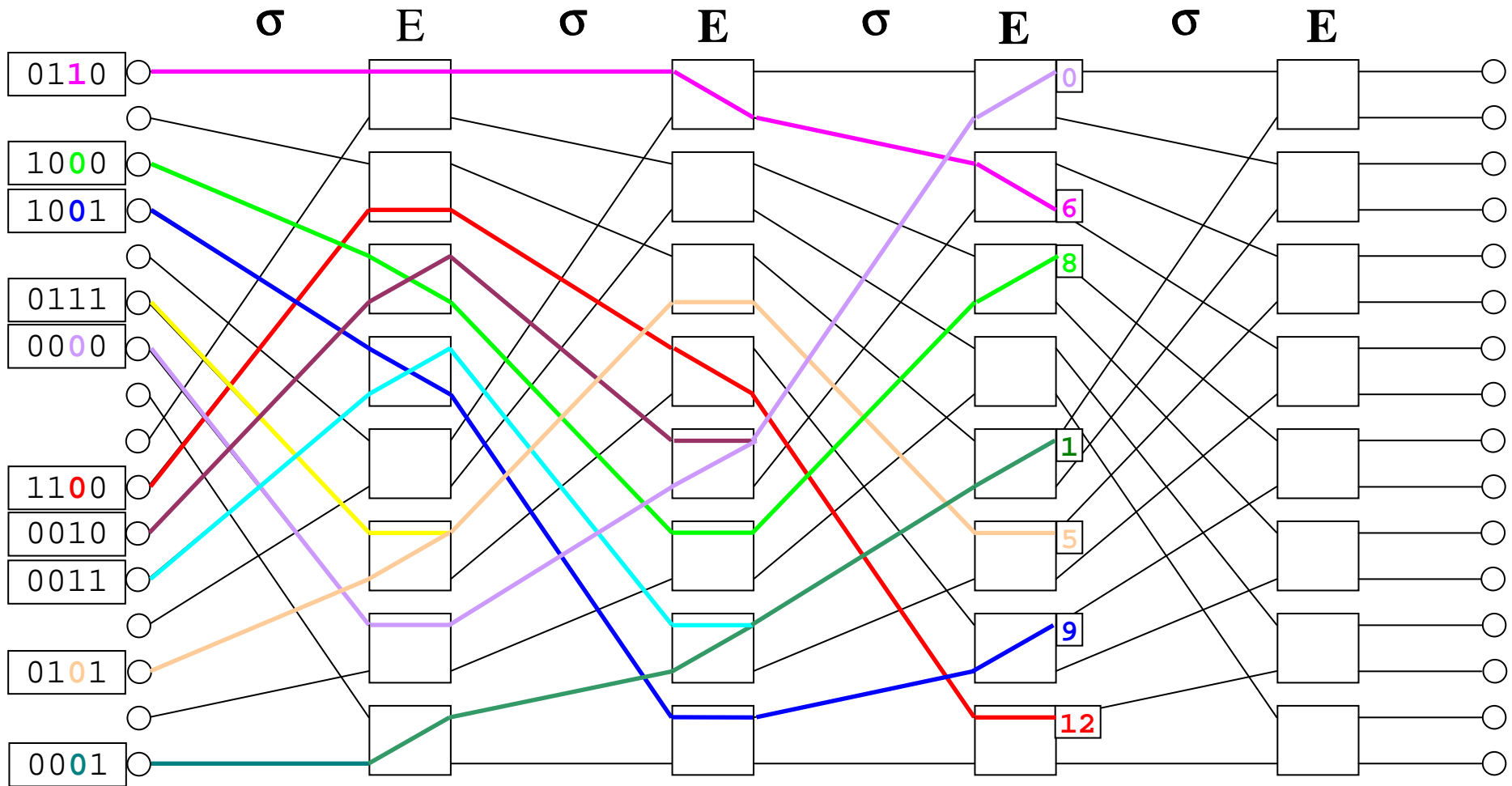


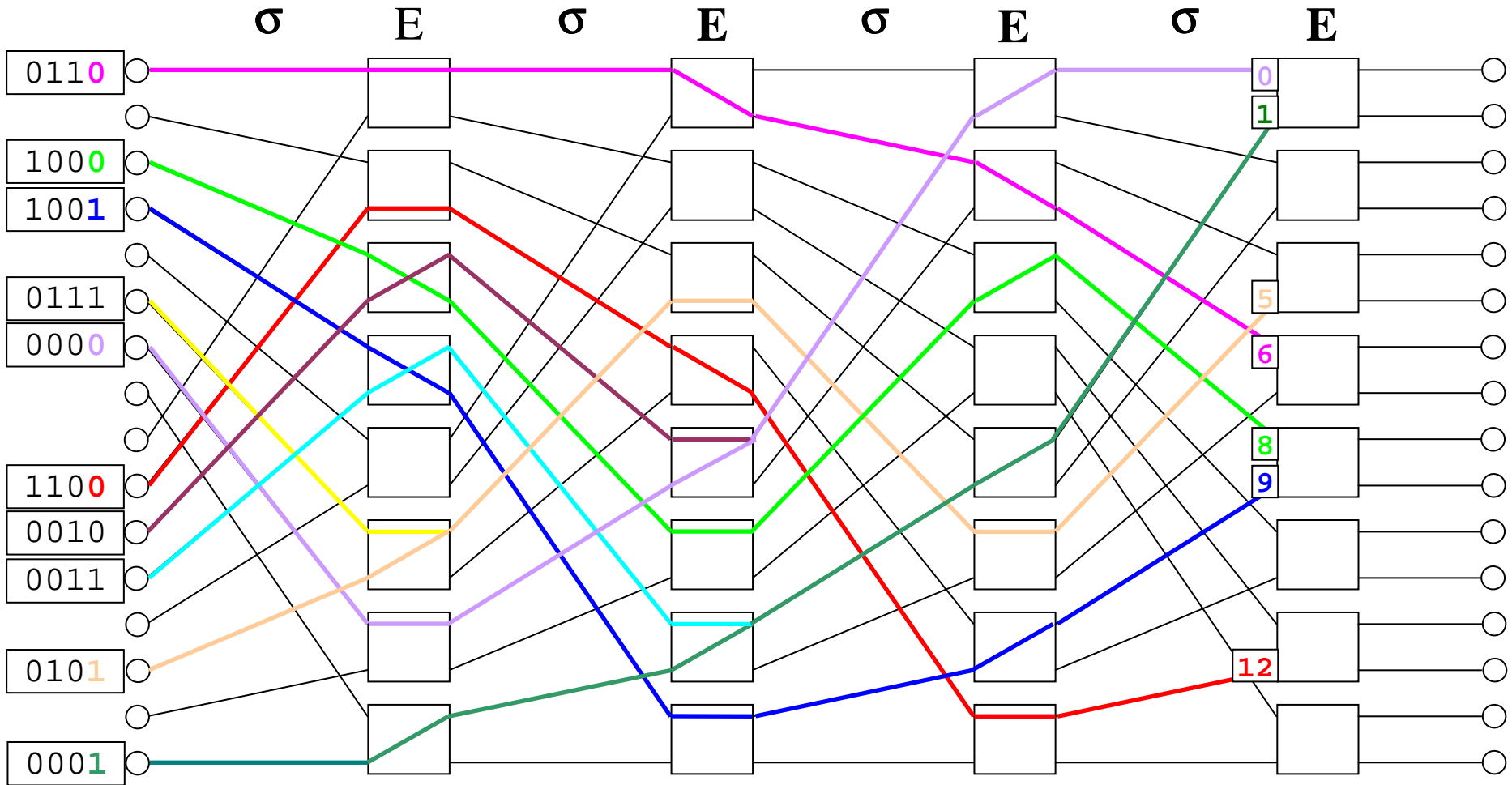


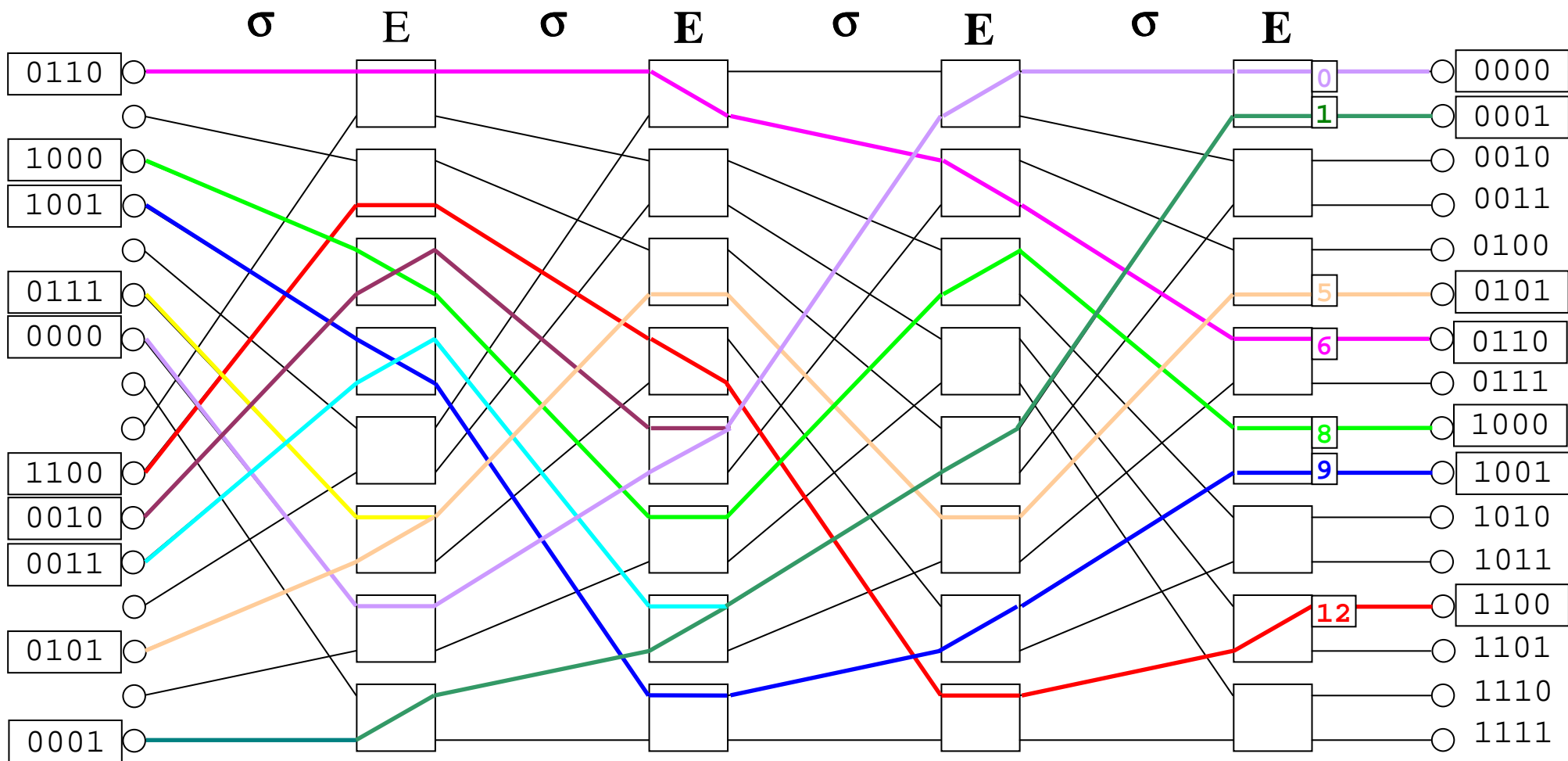












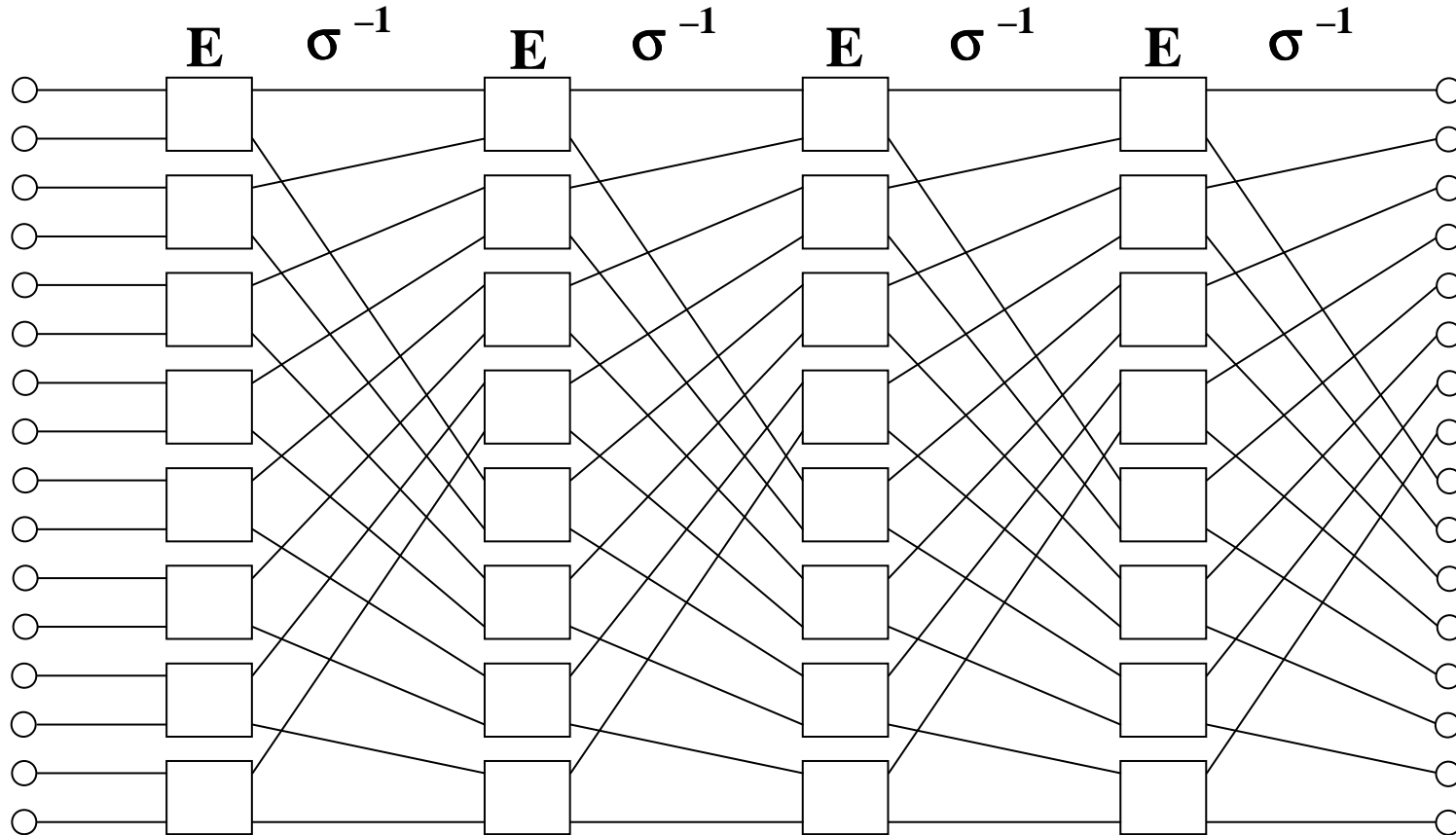
→ k výstupu dorazilo 7 z 10

Inverzná Ω -sieť:

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

takže:

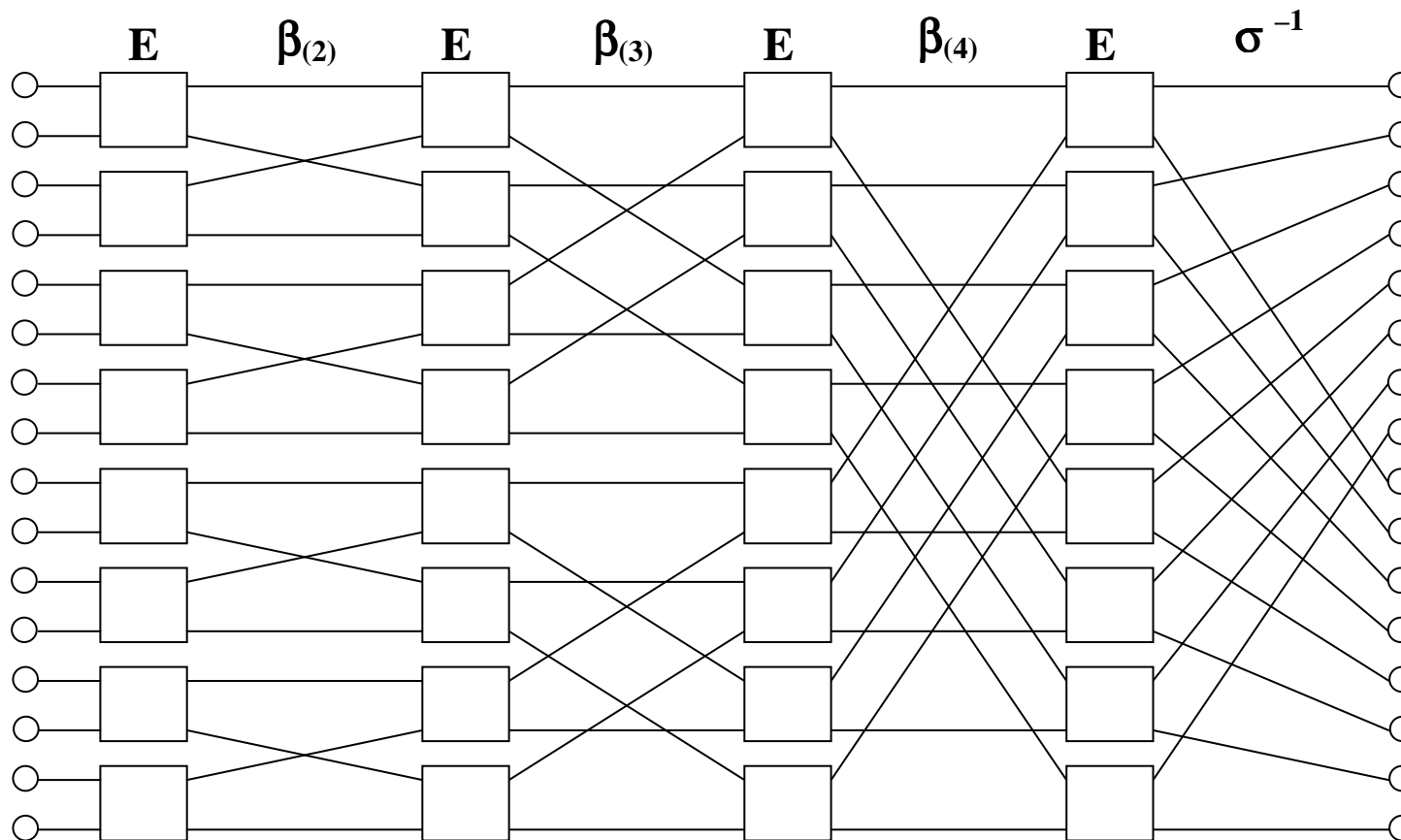
$$\Omega_N^{-1} = (\mathbf{E} \sigma^{-1})^n$$



Nepriama binárna n-kubická sieť (Indirect binary cube)

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

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



Samosmerovací algoritmus:

Prepínač v k-tom stupni siete číta k-ty bit smerovacieho návestia (začínajúc od LSB smerom k MSB) a ak sa tento rovná:

0 potom choď na horný výstup

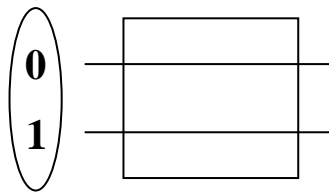
1 potom choď na dolný výstup

← smer čítania bitov

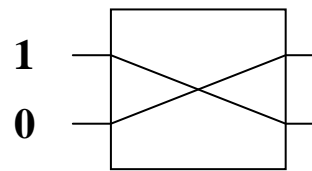
tag = [d_n d_{n-1} ... d_2 d_1]

prvý
čítaný bit

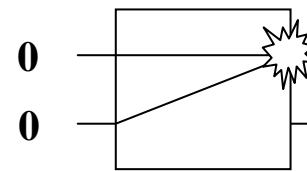
k-ty bit v k-tom
stupni



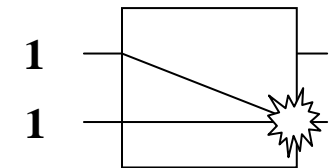
priamo



výmena



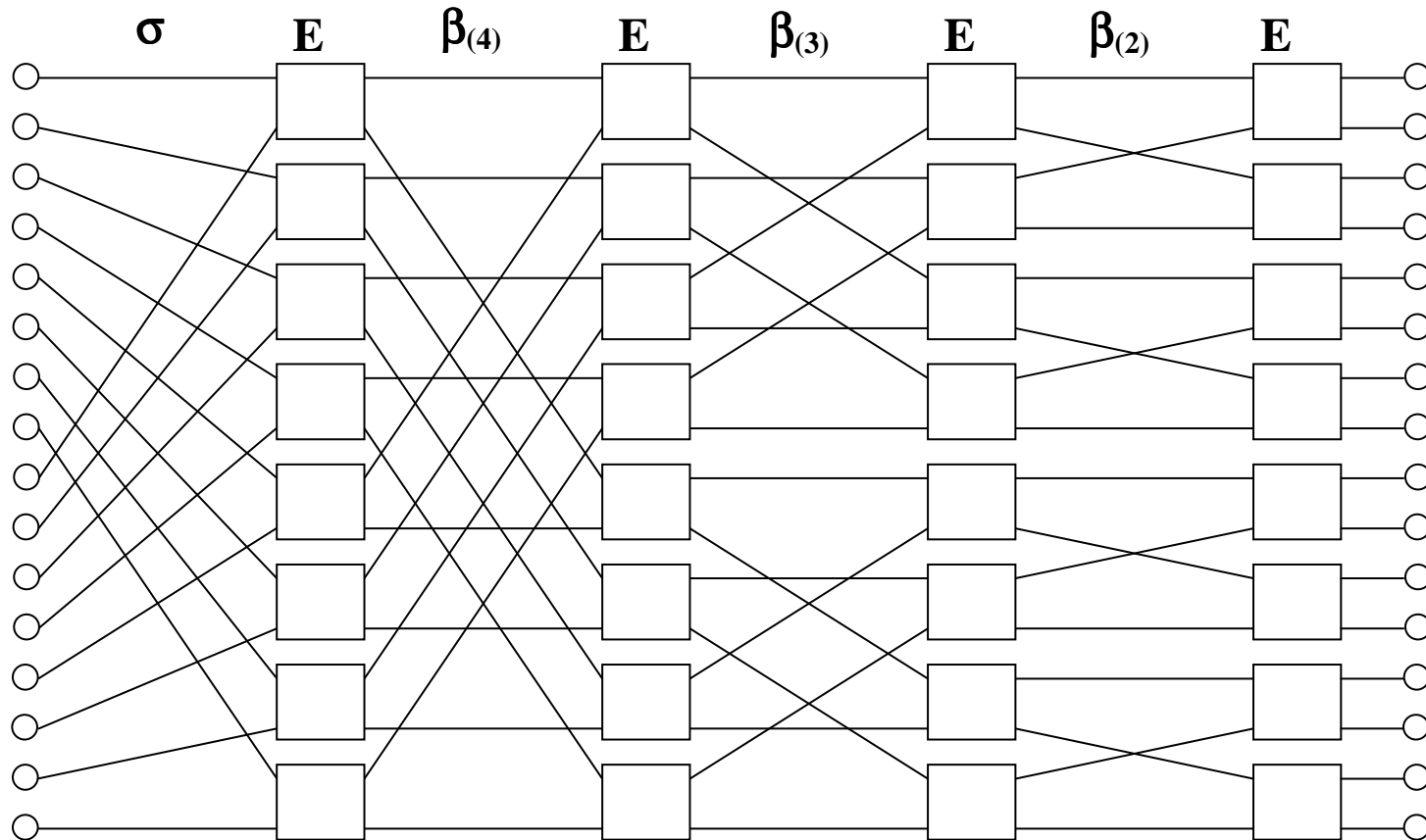
konflikt



konflikt

Priama binárna n-kubická sieť:

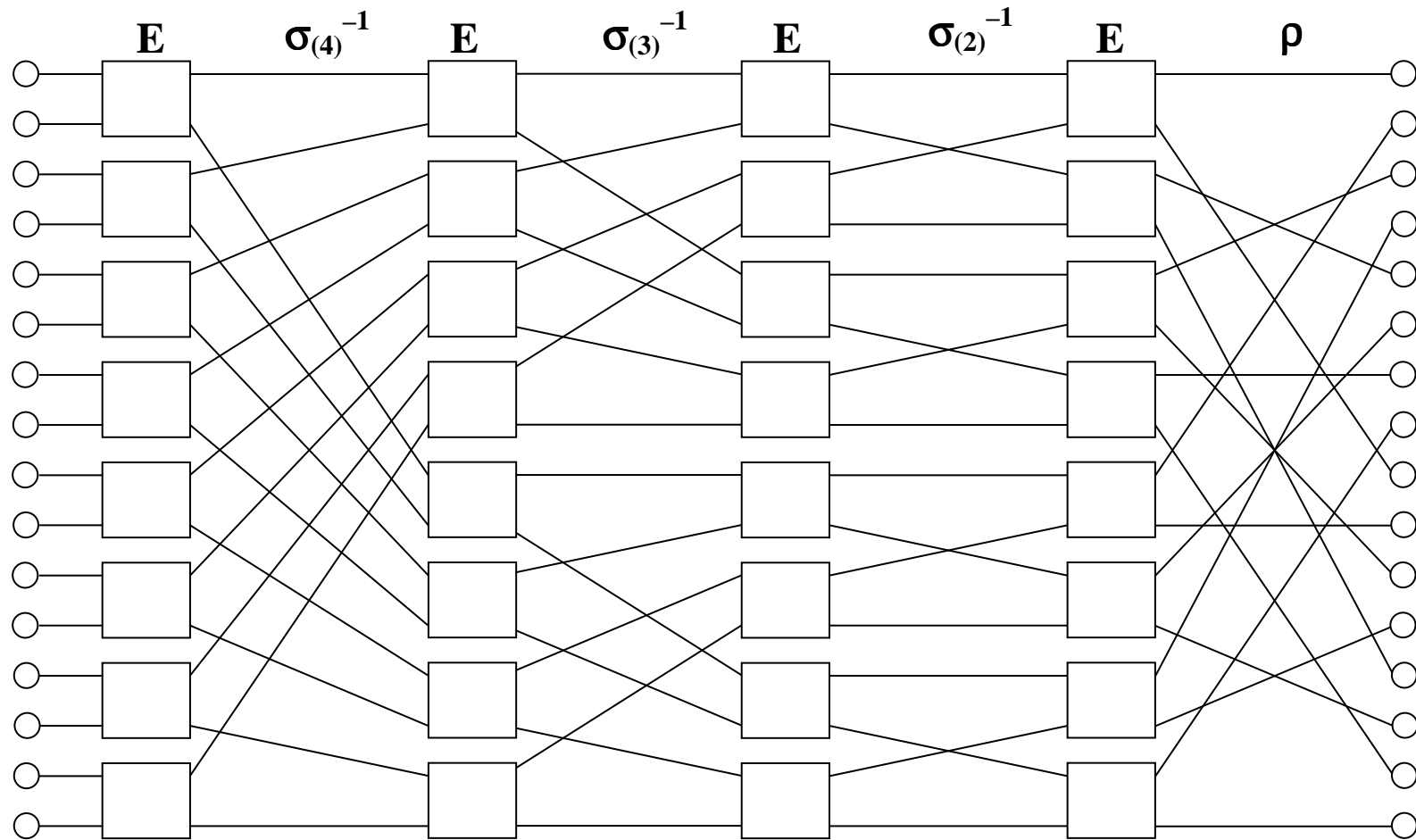
$$C_N^{-1} = (\mathbf{E}\beta_{(2)}\mathbf{E}\beta_{(3)}\mathbf{E}\beta_{(4)} \dots \mathbf{E}\beta_{(n)}\mathbf{E}\sigma^{-1})^{-1} = \sigma \mathbf{E}\beta_{(n)} \dots \mathbf{E}\beta_{(4)} \mathbf{E}\beta_{(3)} \mathbf{E}\beta_{(2)} \mathbf{E}$$



Preusporiadajúca R-sieť (R-network)

$$\mathbf{R}_N = \mathbf{E}\sigma_{(n)}^{-1} \mathbf{E}\sigma_{(n-1)}^{-1} \dots \mathbf{E}\sigma_{(2)}^{-1} \mathbf{E}\rho$$

Napr. pre $N=16$, $n=4 \rightarrow \mathbf{R}_{16} = \mathbf{E}\sigma_{(4)}^{-1} \mathbf{E}\sigma_{(3)}^{-1} \mathbf{E}\sigma_{(2)}^{-1} \mathbf{E}\rho$



Samosmerovací algoritmus (totožný s algoritmom pre C_N):

Prepínač v k -tom stupni siete číta k -ty bit smerovacieho návestia (začínajúc od LSB smerom k MSB) a ak sa tento rovná:

0 potom chod' na horný výstup

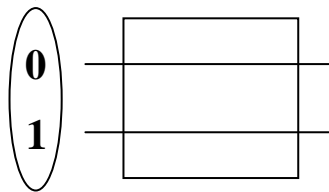
1 potom chod' na dolný výstup

← smer čítania bitov

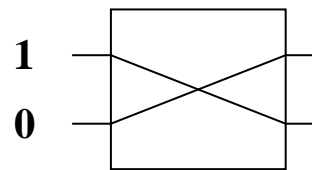
tag = [d_n d_{n-1} ... d_2 d_1]

prvý
čítaný bit

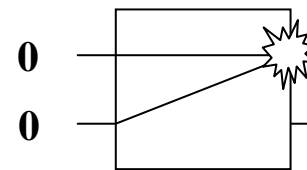
k-ty bit v k-tom
stupni



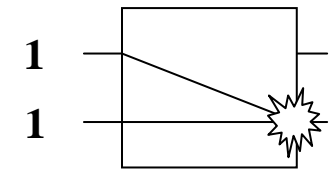
priamo



výmena



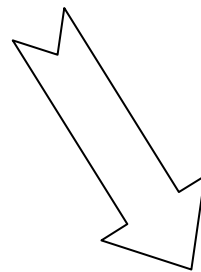
konflikt



konflikt

Dá sa dokázať, že uvedené tri siete (Ω_N^{-1} , C_N , R_N) sú funkčne a topologicky ekvivalentné, t.j. realizujú tú istú permutáciu a ich grafy sú izomorfné (všetky tri siete sú len rôznymi grafmi tej istej siete).

$$\Omega_N^{-1} = C_N = R_N$$

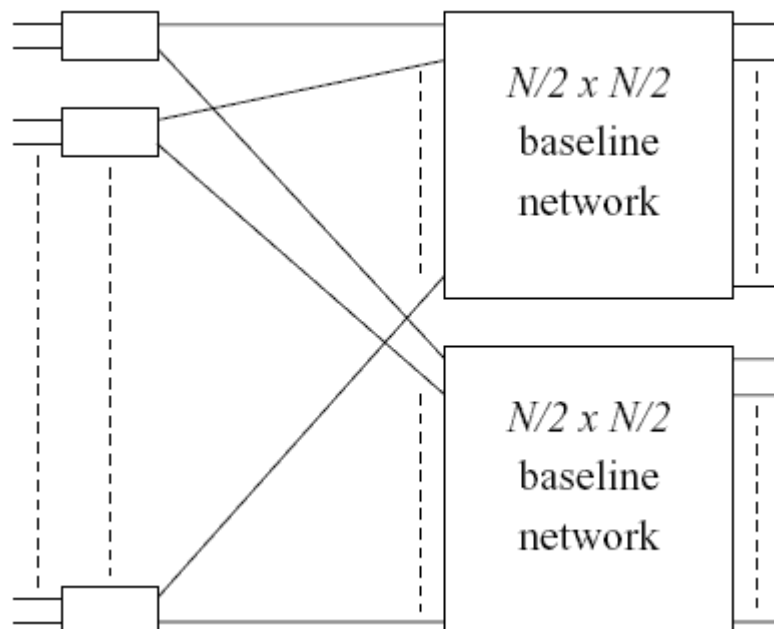


**Domáca úloha:
urobiť dôkaz**

Ďalšie permutačné prepojovacie siete:

Baseline sieť (Baseline network)

Topológia baseline siete je definovaná rekurzívnym spôsobom podľa nasledovného obrázku:



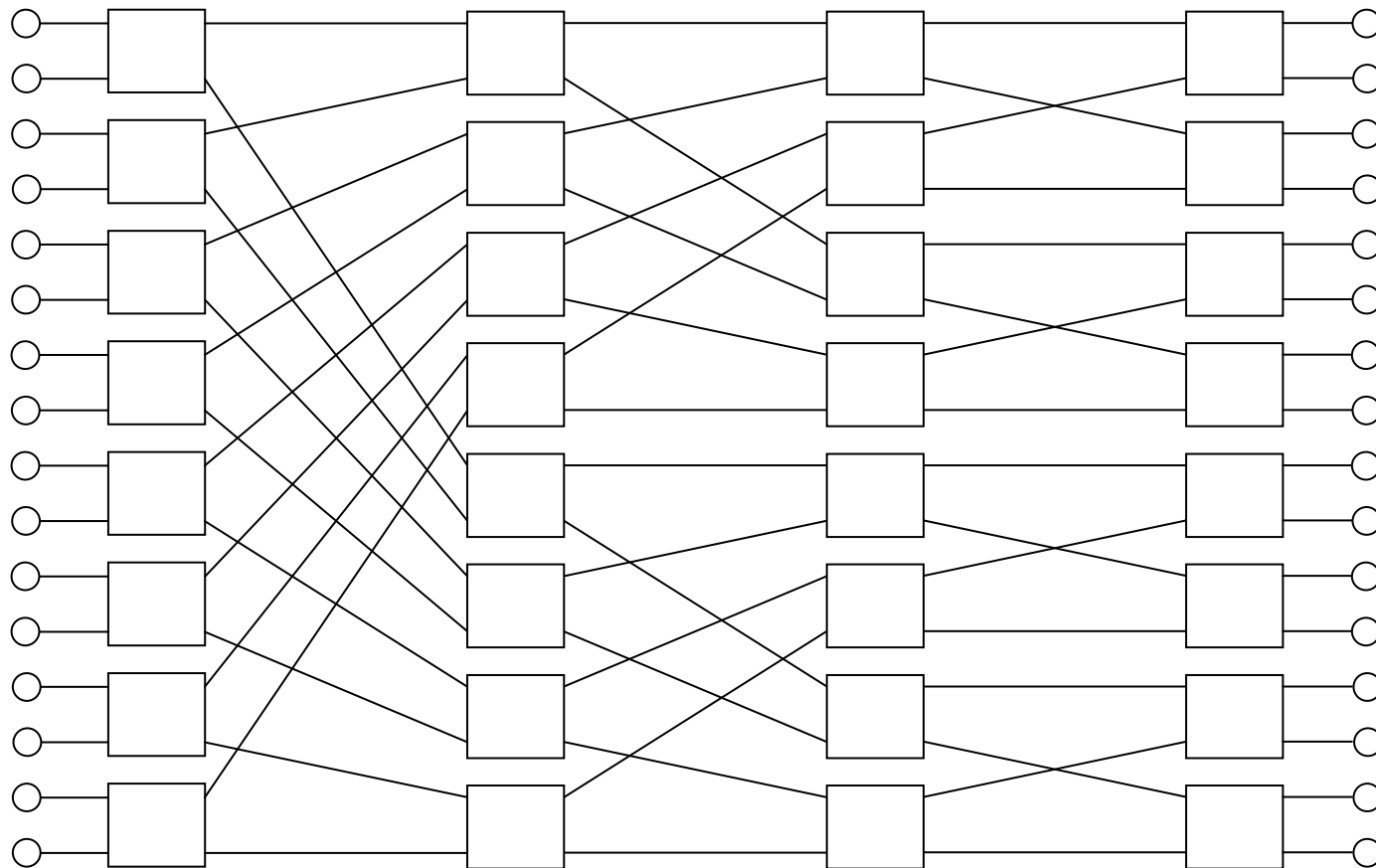
Spojovacia funkcia medzi dvomi stupňami je reverzné miešanie (σ^{-1}).

Rekurzívny proces sa ukončí ak dosiahneme elementy 2×2 . Rozpísaním rekurzie môžeme získať zápis baseline siete:

$$\text{Baseline}_N = \mathbf{E}\sigma_{(n)}^{-1} \mathbf{E}\sigma_{(n-1)}^{-1} \dots \mathbf{E}\sigma_{(2)}^{-1} \mathbf{E},$$

kde $n = \log_2 N$

Napr. pre $N=16$, $n=4$:



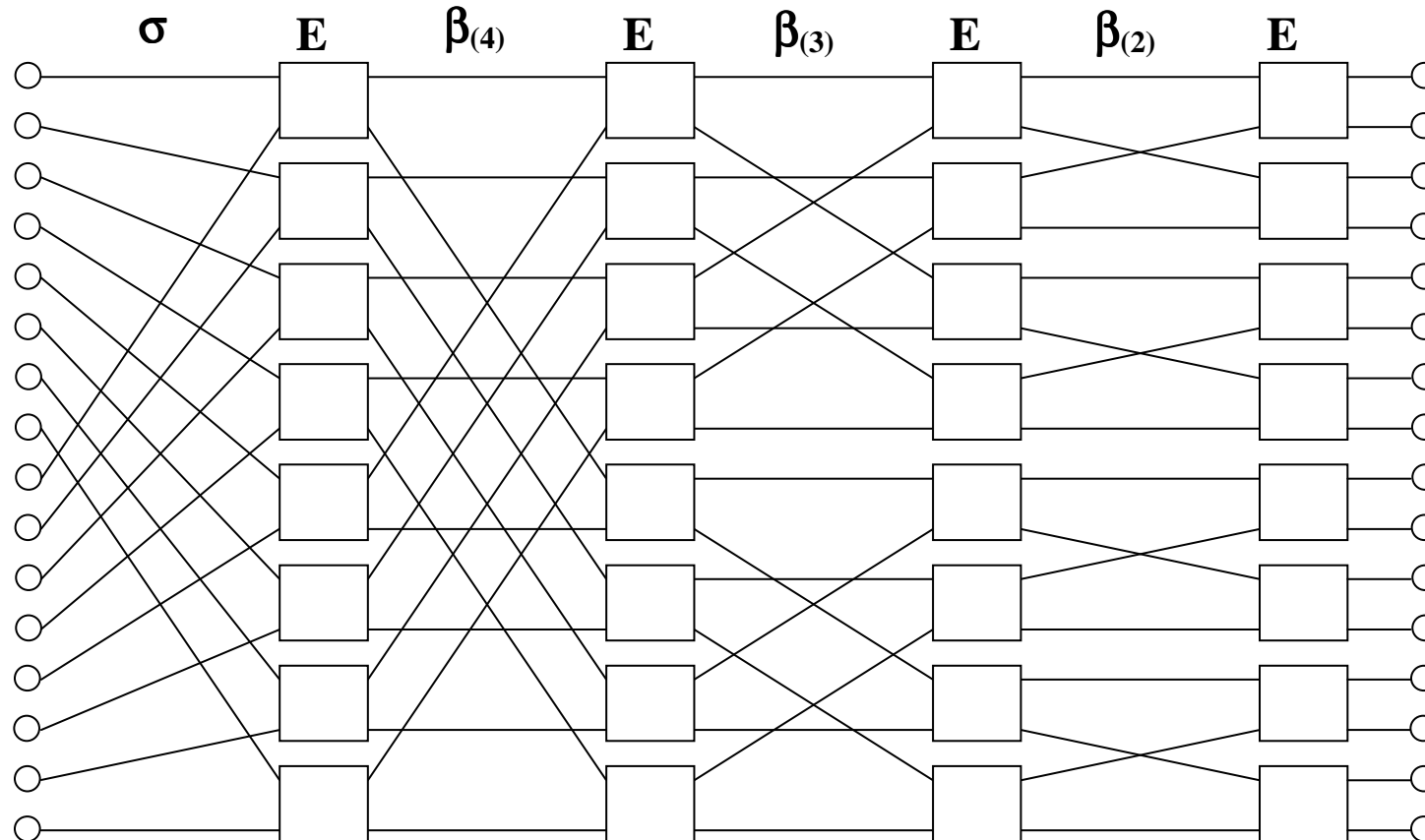
Banyan network

Banyan sieť, niekedy tiež nazývaná generalizovaná kubická sieť (generalized cube network) je sieť definovaná predpisom:

$$\text{Banyan}_N = \sigma E\beta_{(n)} \dots E\beta_{(4)} E\beta_{(3)} E\beta_{(2)} E,$$

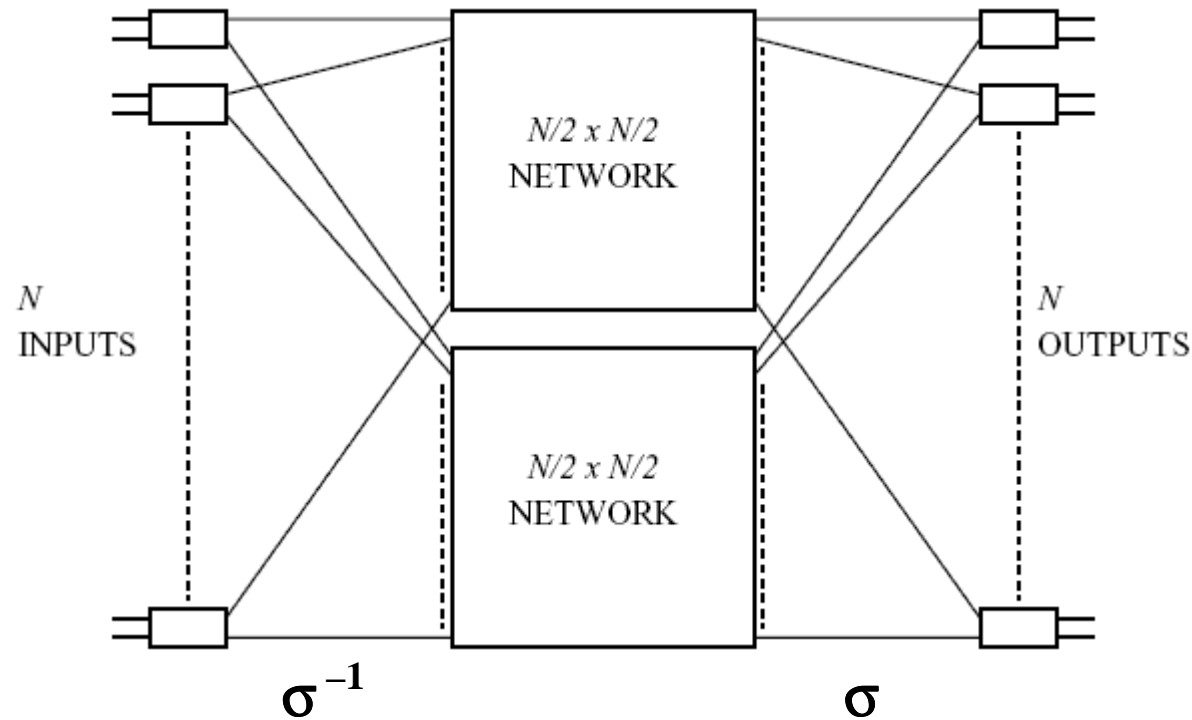
kde $n = \log_2 N$

a teda sa jedná o priamu binárnu n -kubickú sieť (inverznú k nepriamej – C_N^{-1})



Benešova sieť

Rekurzívna definícia Benešovej siete:



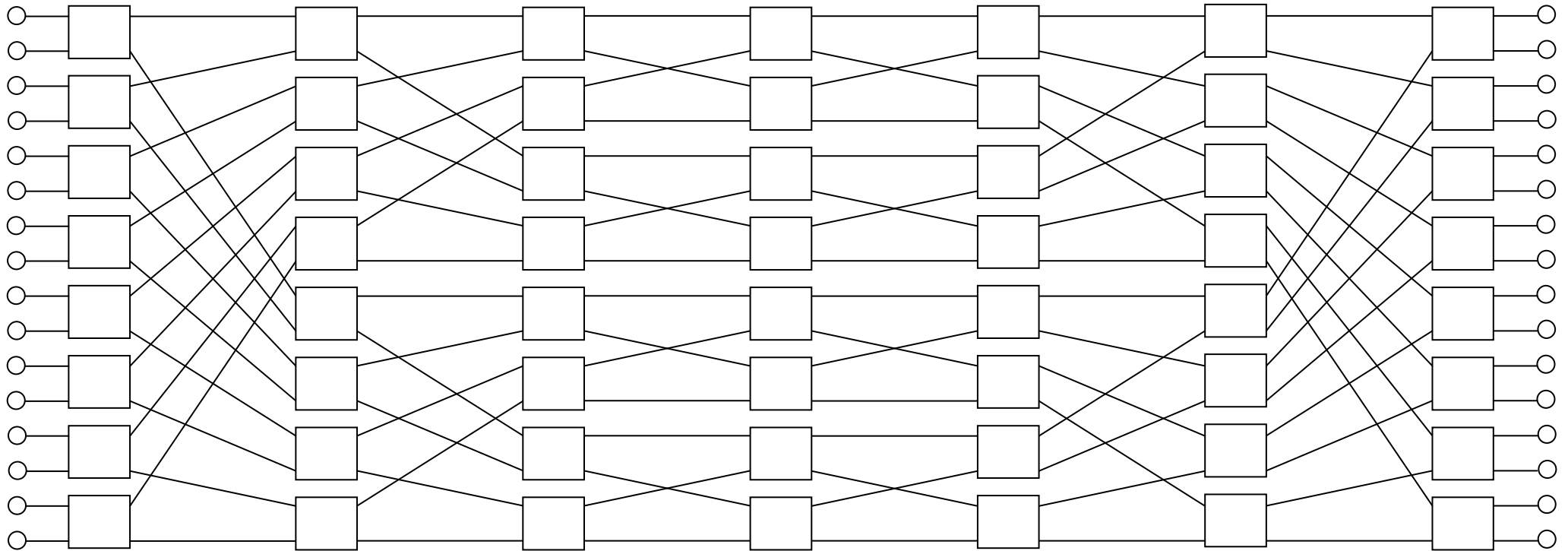
Keď je rekurzívny proces ukončený sieť má $2\log_2 N - 1$ stupňov a v každom stupni $N/2$ spojovacích elementov. Benešova sieť patrí medzi rekonfigurovatel'né siete bez blokovania a preto vyžaduje algoritmus pre rekonfiguračný proces – centrálné riadenie. Existujú tiež samosmerovacie algoritmy čiastočne potláčajúce blokovanie v sieti.

Predpis:

$$\mathbf{Beneš}_N = \mathbf{E}\sigma_{(k)}^{-1} \mathbf{E}\sigma_{(k-1)}^{-1} \mathbf{E} \dots \sigma_{(2)}^{-1} \mathbf{E} \sigma_{(2)} \dots \mathbf{E}\sigma_{(k-1)} \mathbf{E}\sigma_{(k)} \mathbf{E},$$

kde $k=\log_2 N$

Napr. pre $N=16$, stupňov: $n=2\log_2 N-1=7$:

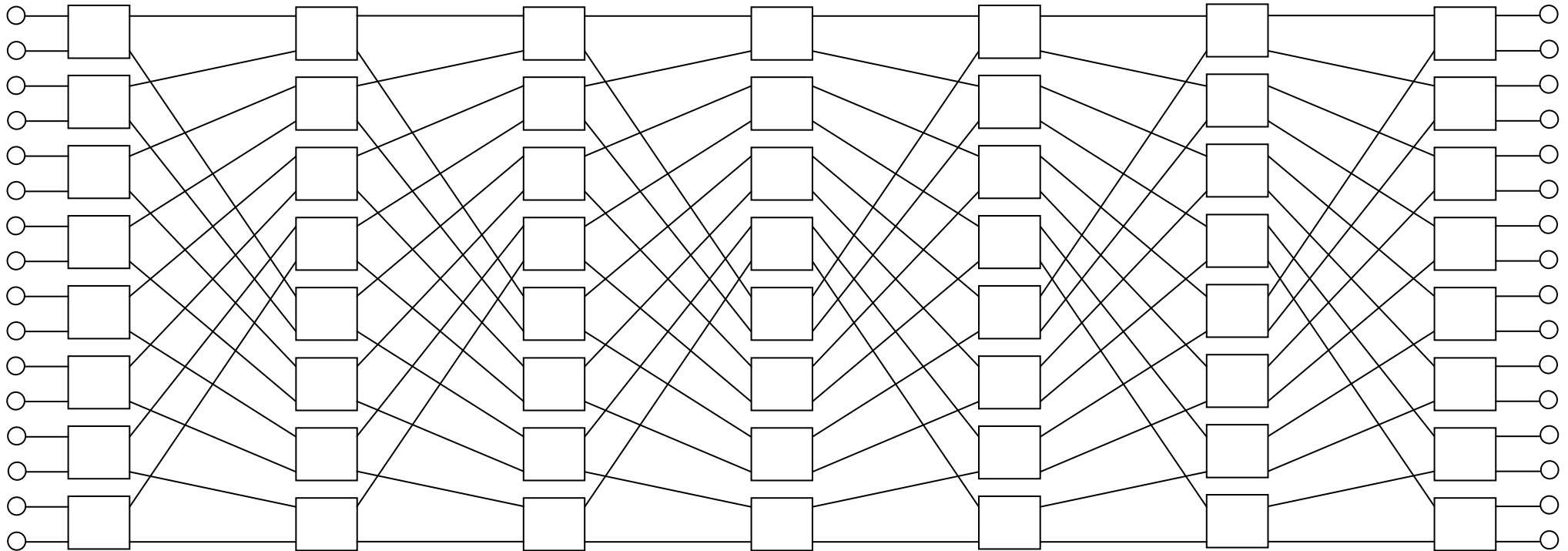


Na Benešovú sieť sa tiež môžeme pozerat' ako na zaradenie dvoch baseline sietí za sebou (baseline sieť a reverzná baseline sieť). Preto bývajú Benešove siete niekedy tiež nazývané "sériové baseline siete".

Alternatívne je možné použiť definíciu Benešovej siete podľa nasledovného predpisu:

$$\mathbf{Beneš}_N = \mathbf{E} \sigma^{-1} \mathbf{E} \sigma^{-1} \mathbf{E} \dots \sigma^{-1} \mathbf{E} \sigma \dots \mathbf{E} \sigma \mathbf{E} \sigma \mathbf{E} = (\mathbf{E} \sigma^{-1})^{k-1} \mathbf{E} (\sigma \mathbf{E})^{k-1},$$

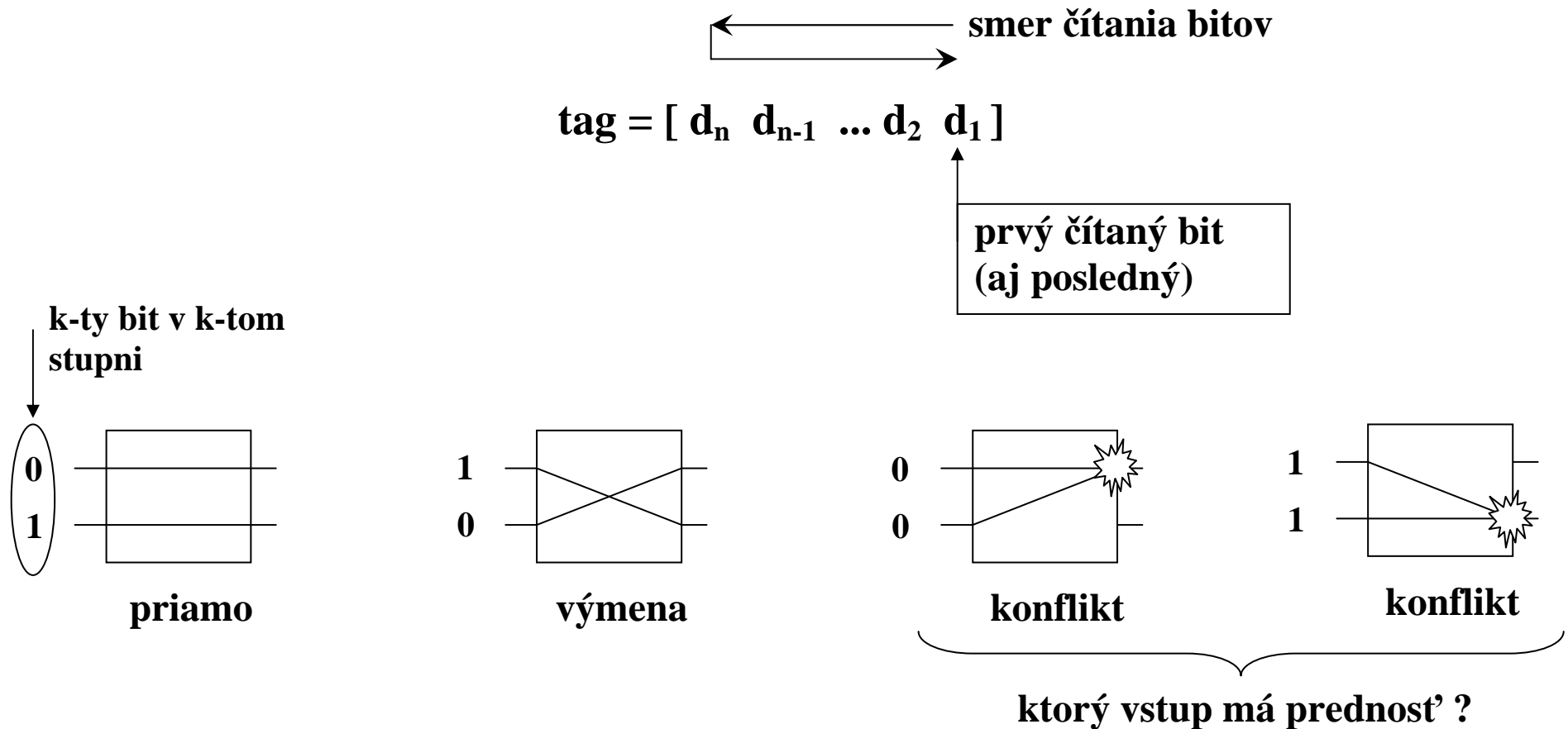
kde $k = \log_2 N$

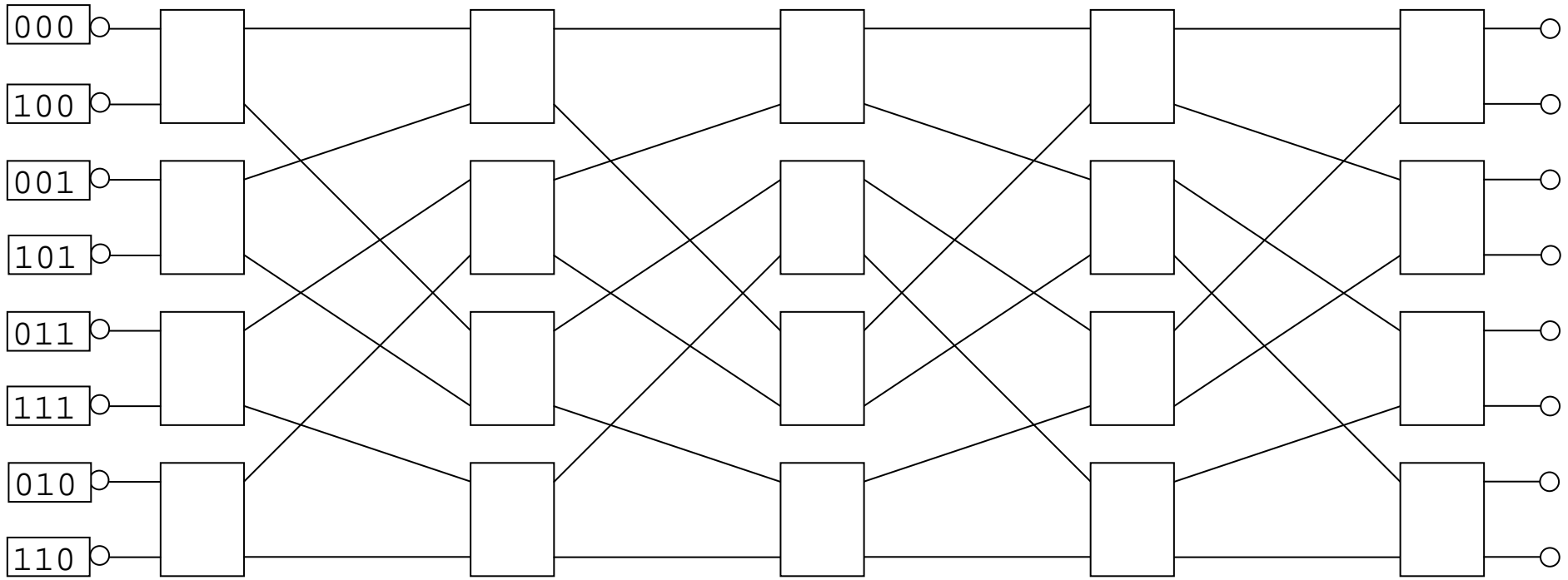


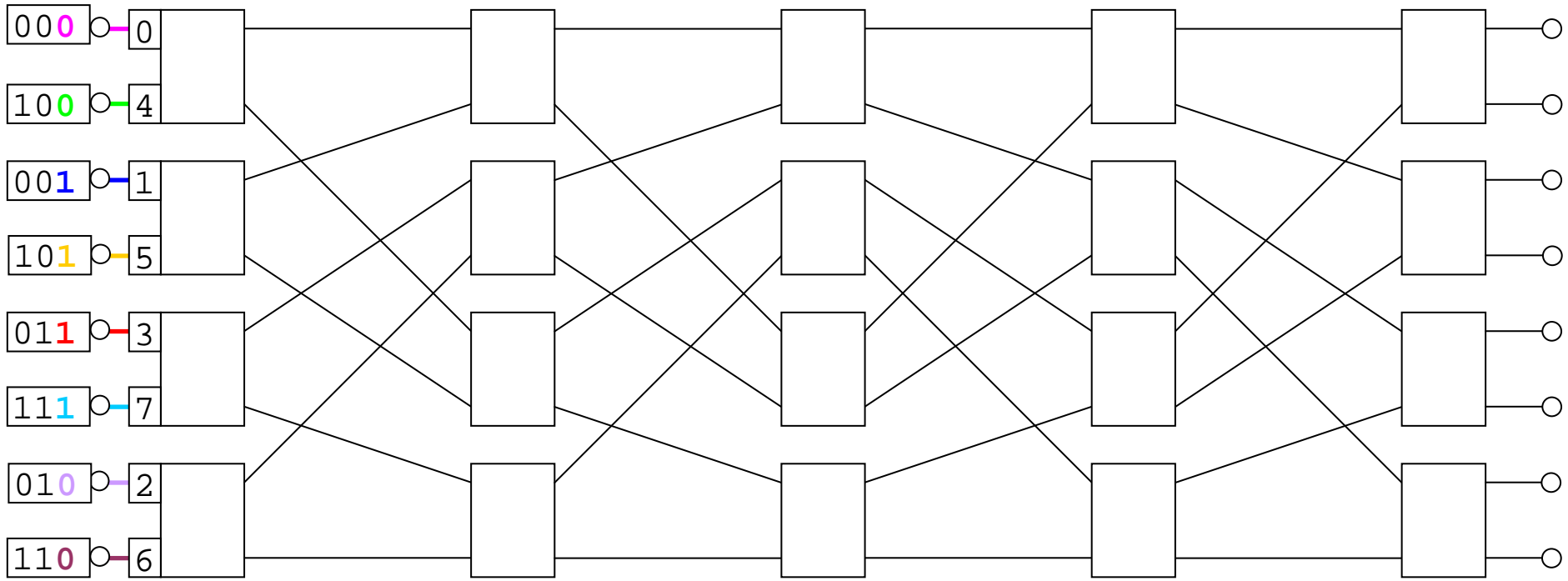
Samosmerovací algoritmus:

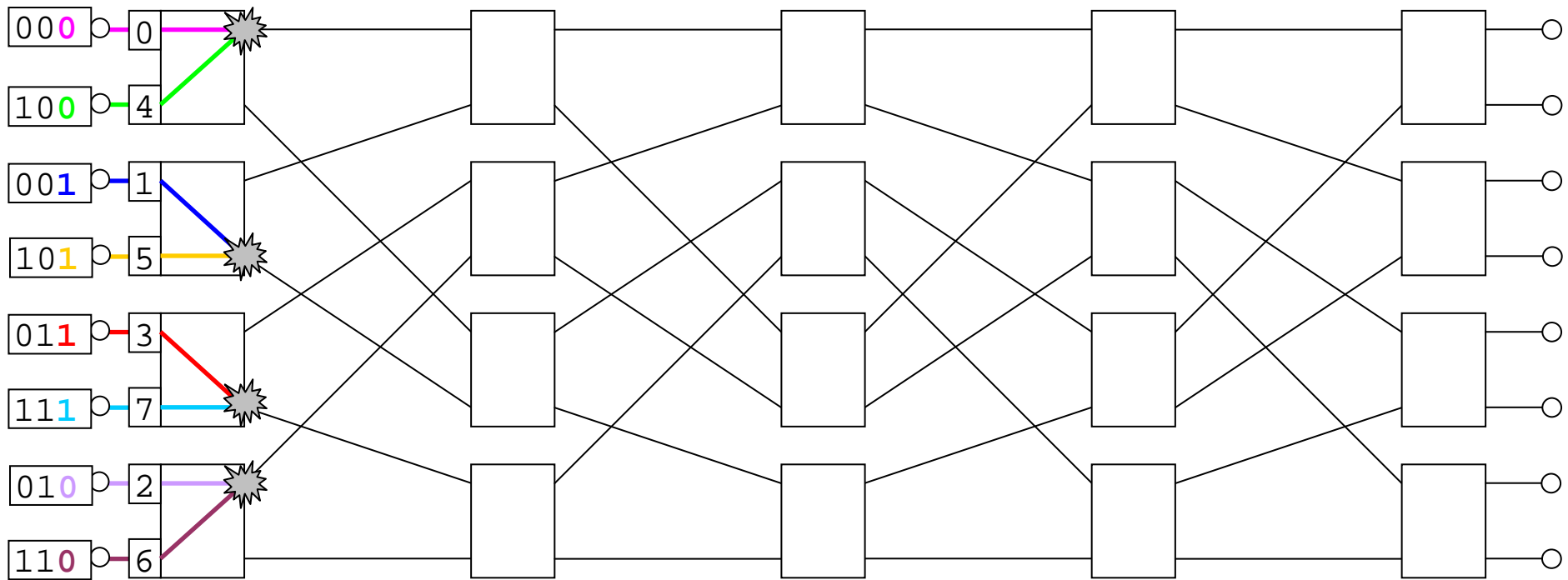
Prepínač v i -tom, resp. j -tom stupni ($1 \leq i \leq k$, resp. $k < j \leq 2k-1$, kde $k = \log_2 N$) siete číta i -ty bit, resp. $(2k-j)$ -ty bit smerovacieho návestia (začínajúc od LSB smerom k MSB) a ak sa tento rovná:

0 potom chod' na horný výstup
1 potom chod' na dolný výstup



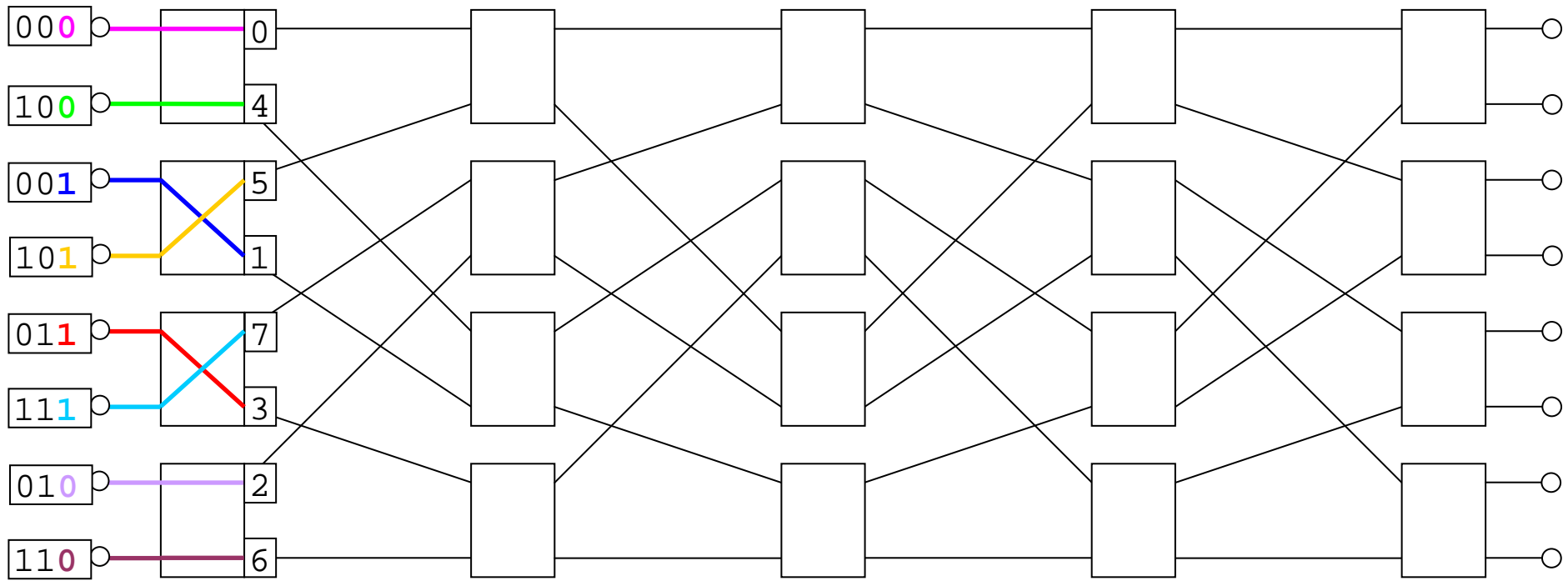


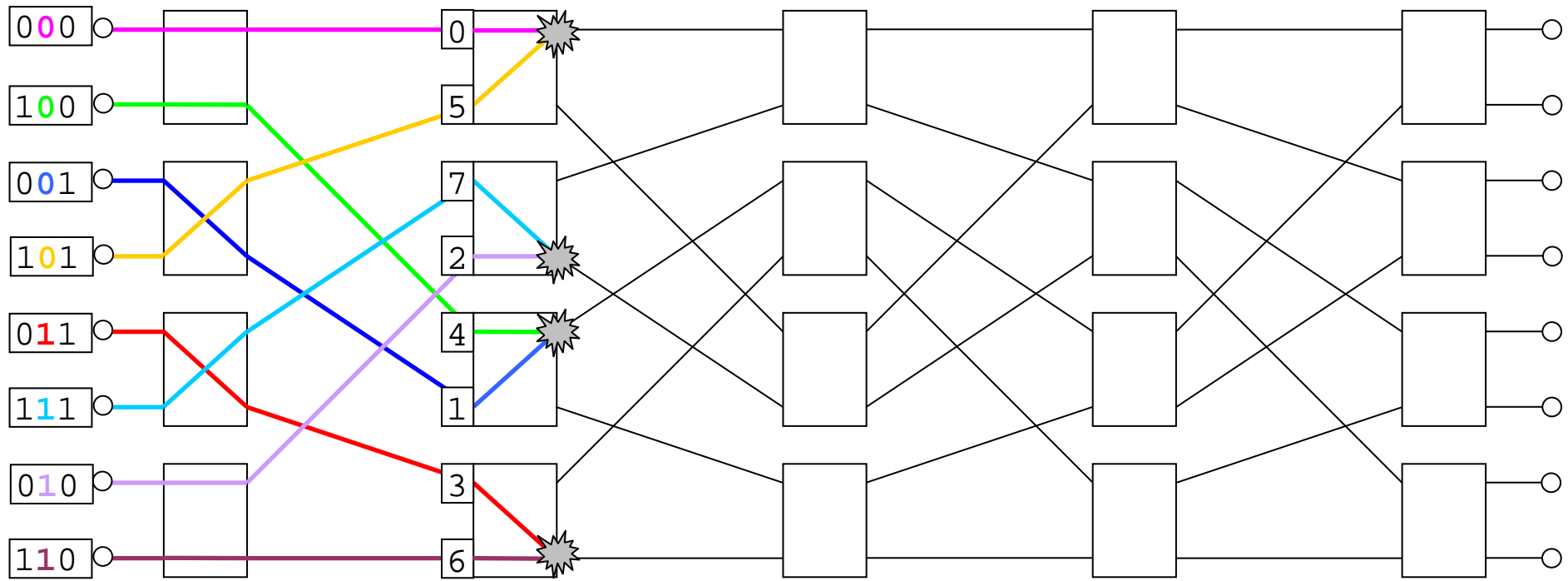


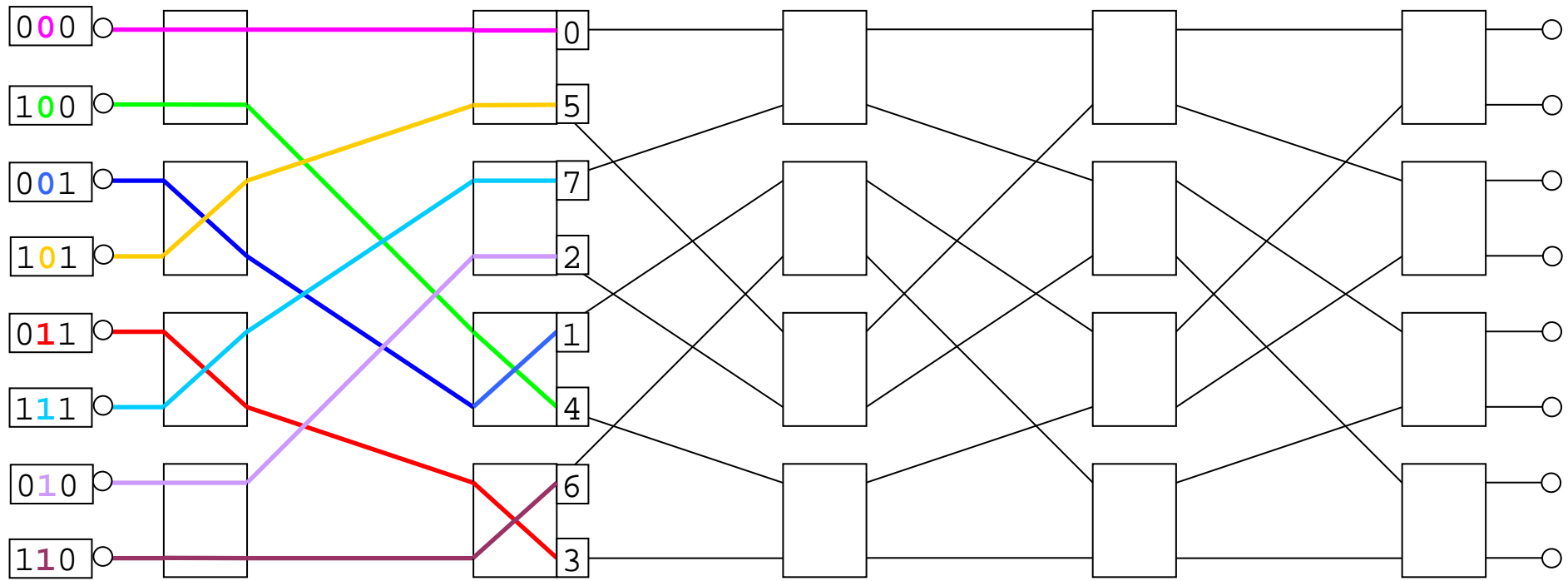


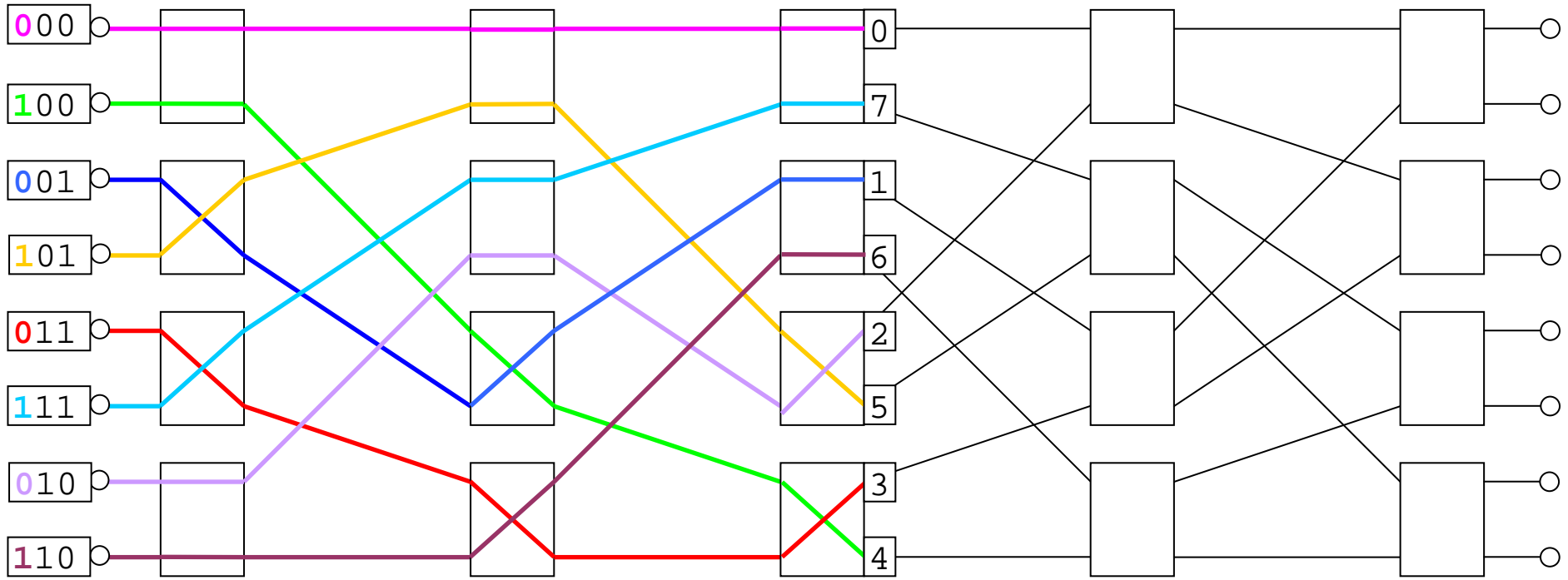
Čo teraz?

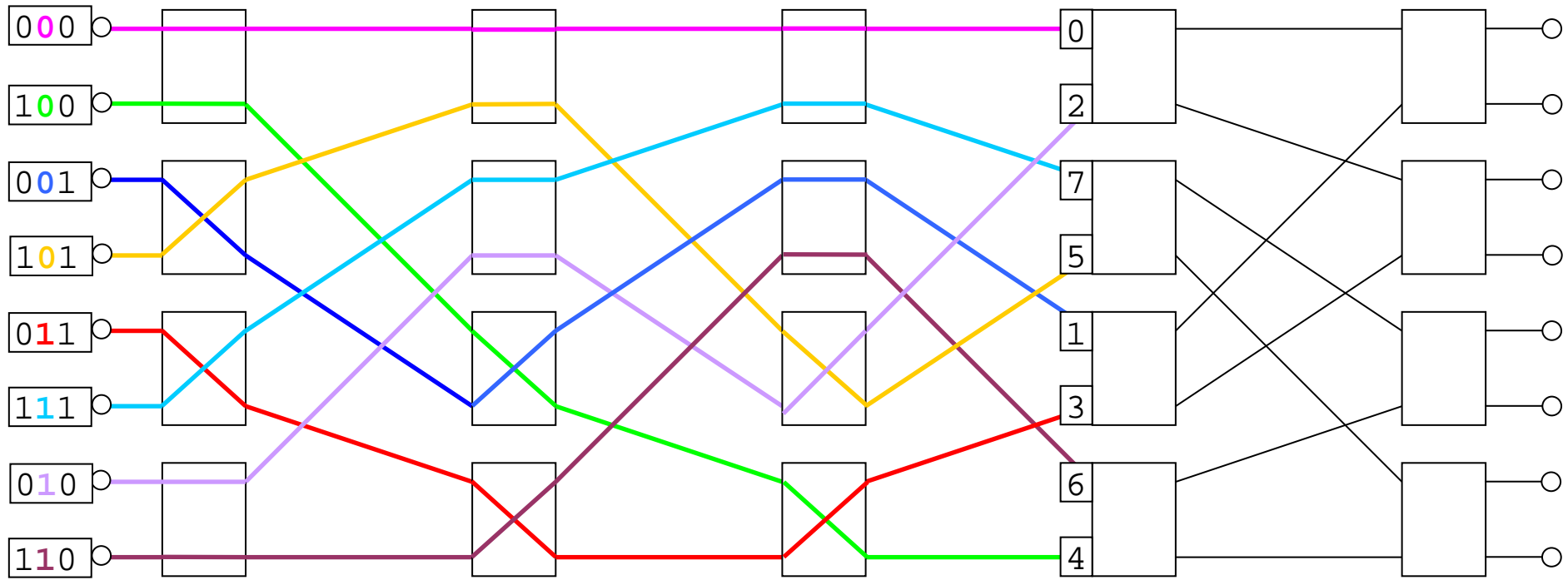
Prednosť má ten, ktorého hodnota tagu (adresa určenia) je menšia...

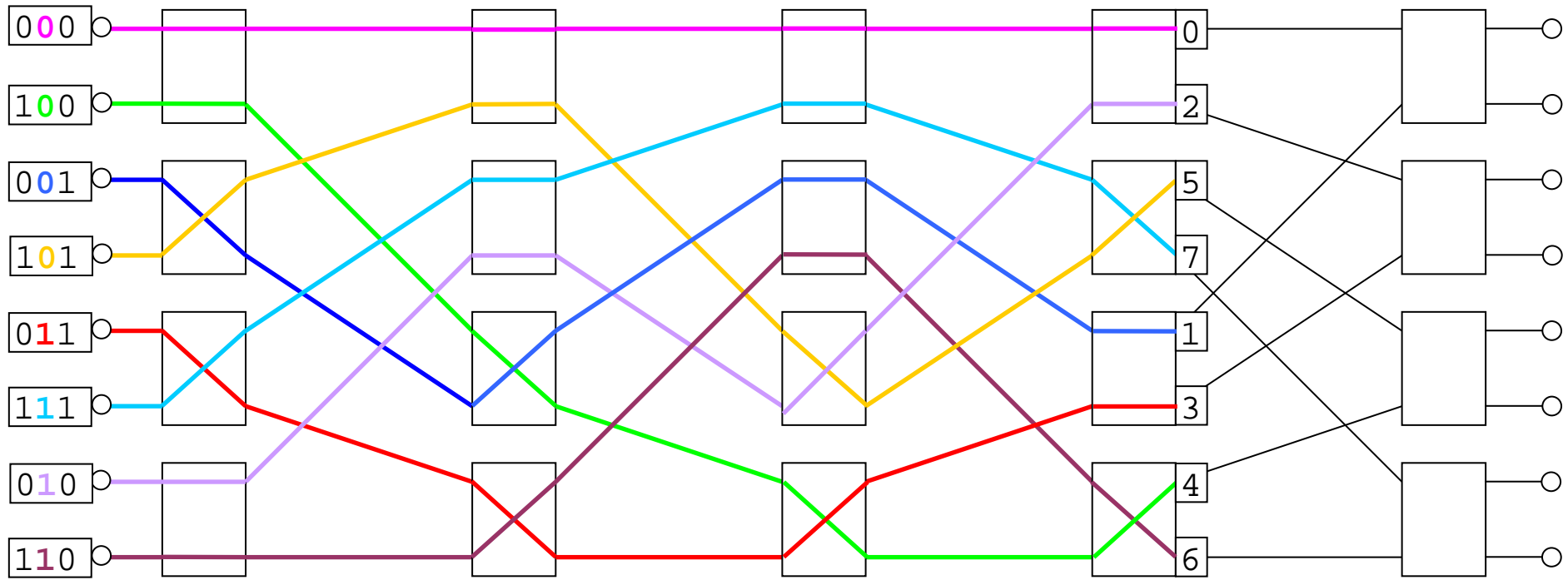


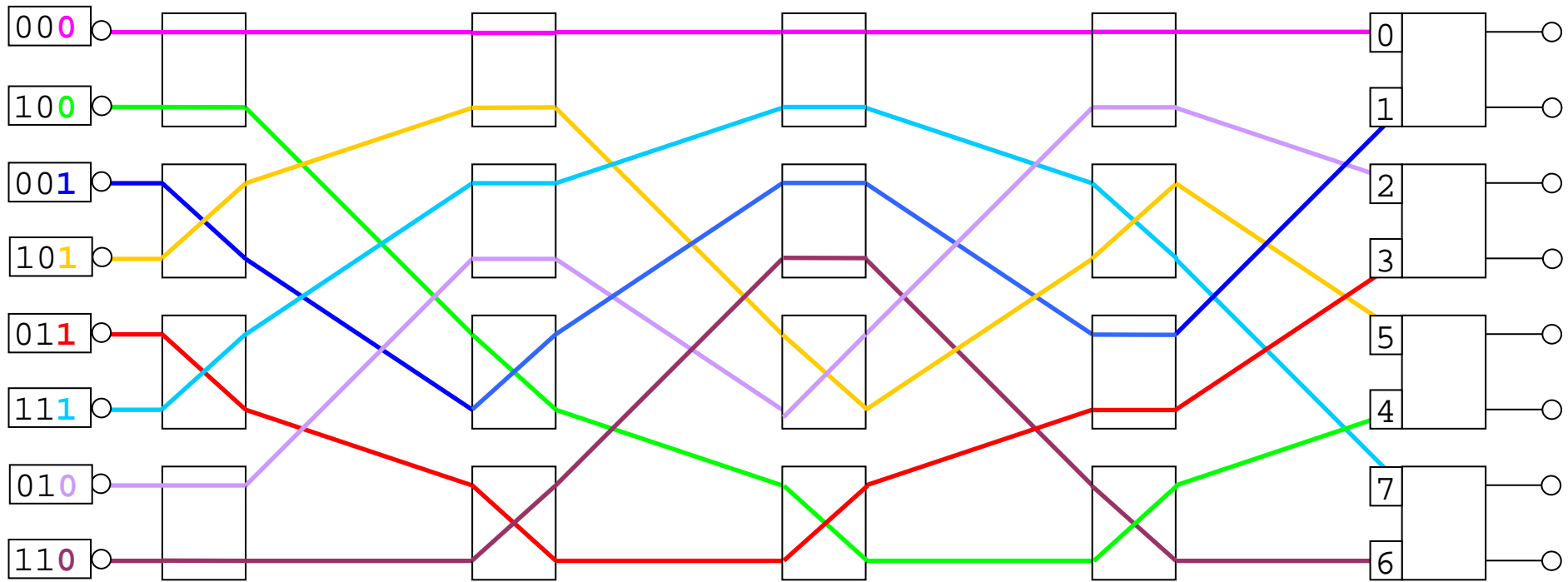


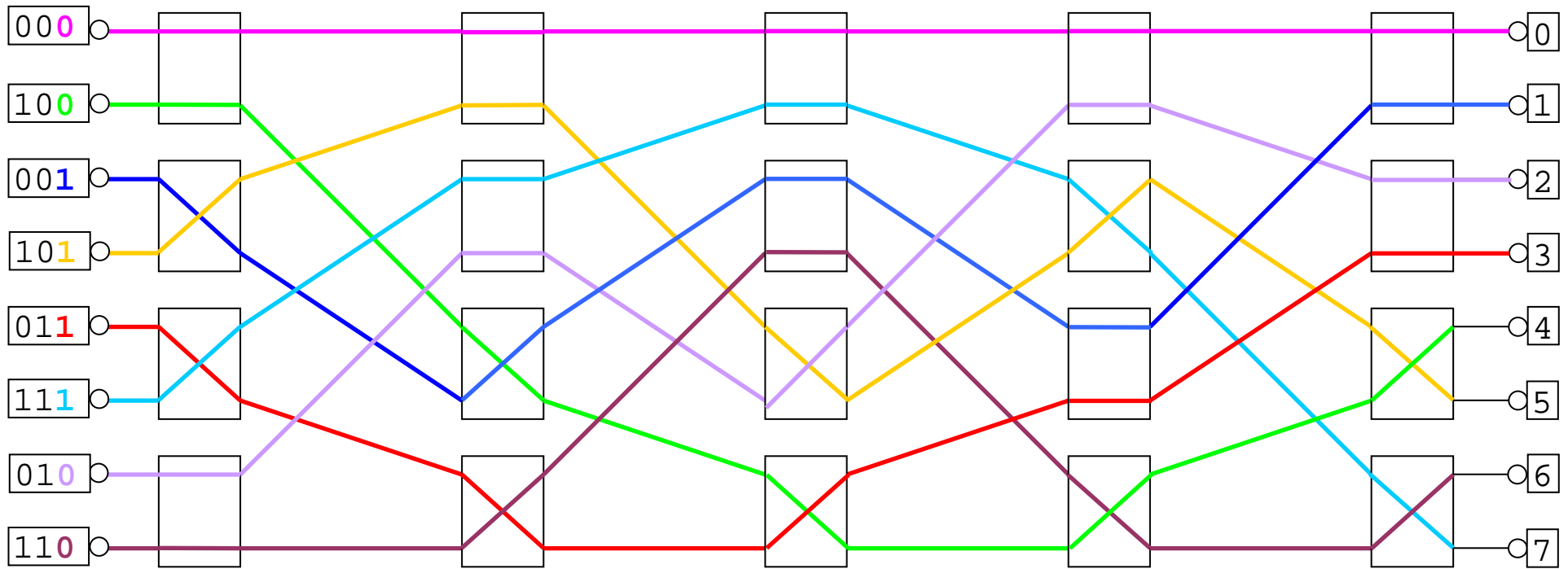






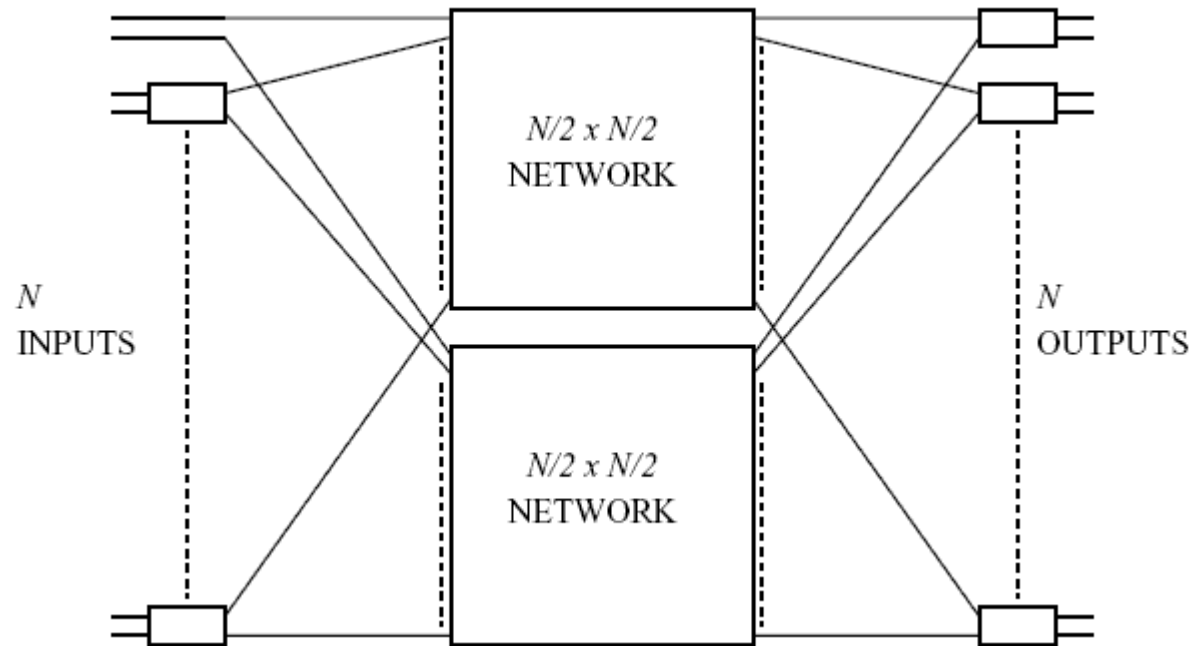




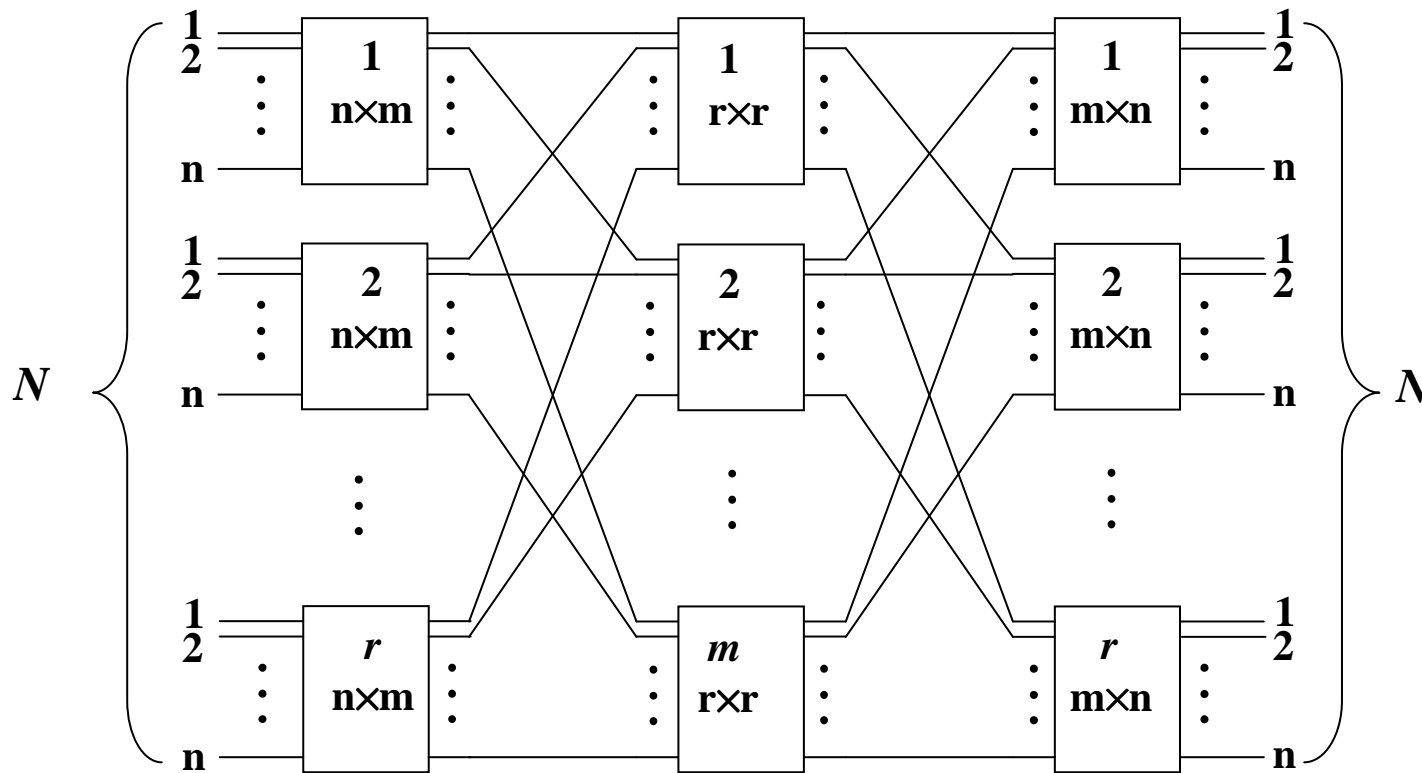


Waksmanova sieť (Waksman network)

Použitím modifikovanej verzie Slepian-Duguidovho teorému môže byť z Benešovej siete vynechaním jedného vstupného alebo výstupného elementu vytvorená tzv. Waksmanova sieť. Waksmanova sieť používa menej prepínacích elementov ako Benešova sieť.



Closova sieť



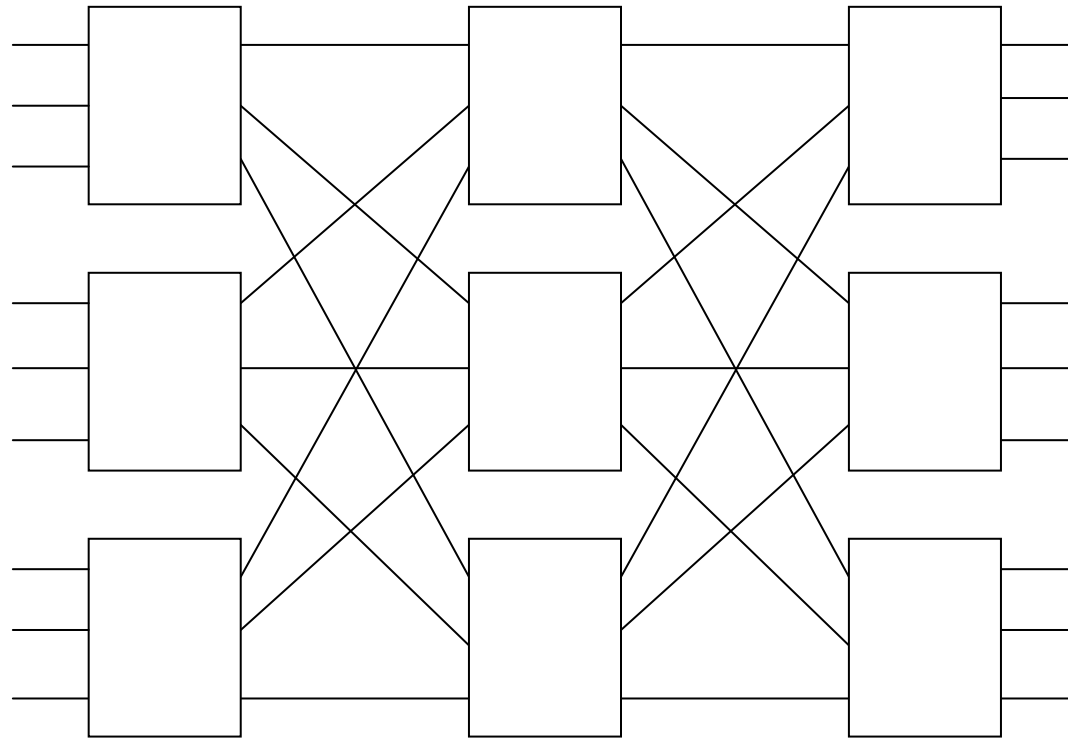
Symetrická Closova sieť

Symetrická 3-stupňová Closova sieť s N vstupnými a N výstupnými portami má r prepínacích modulov veľkosti $n \times m$ na prvom stupni, m prepínacích modulov veľkosti $r \times r$ na strednom stupni a r prepínacích modulov veľkosti $m \times n$ na treťom (výstupnom) stupni. Takáto 3 stupňová sieť je označovaná ako $C(m,n,r)$.

Je Closova sieť s $m=n$ neblokujúca?

Príklad prepojenia v sieti:

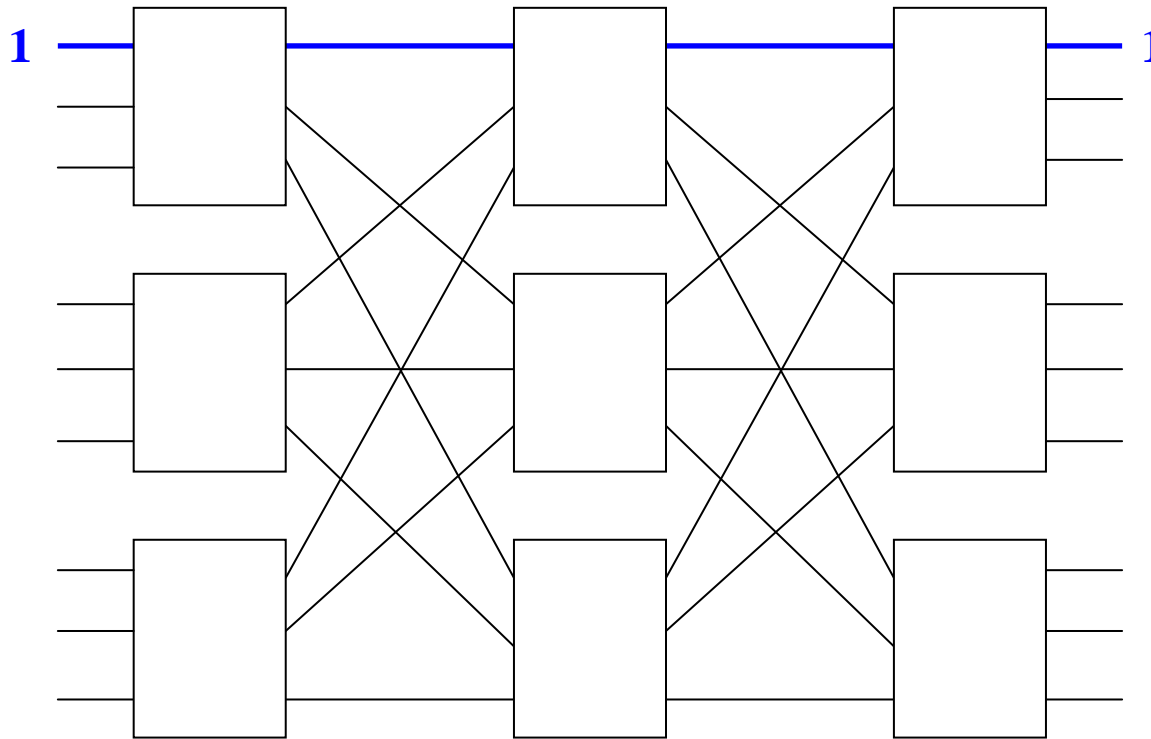
(1,1), (2,2), (4,4), (5,3), ...



Je Closova sieť s $m=n$ neblokujúca?

Príklad prepojenia v sieti:

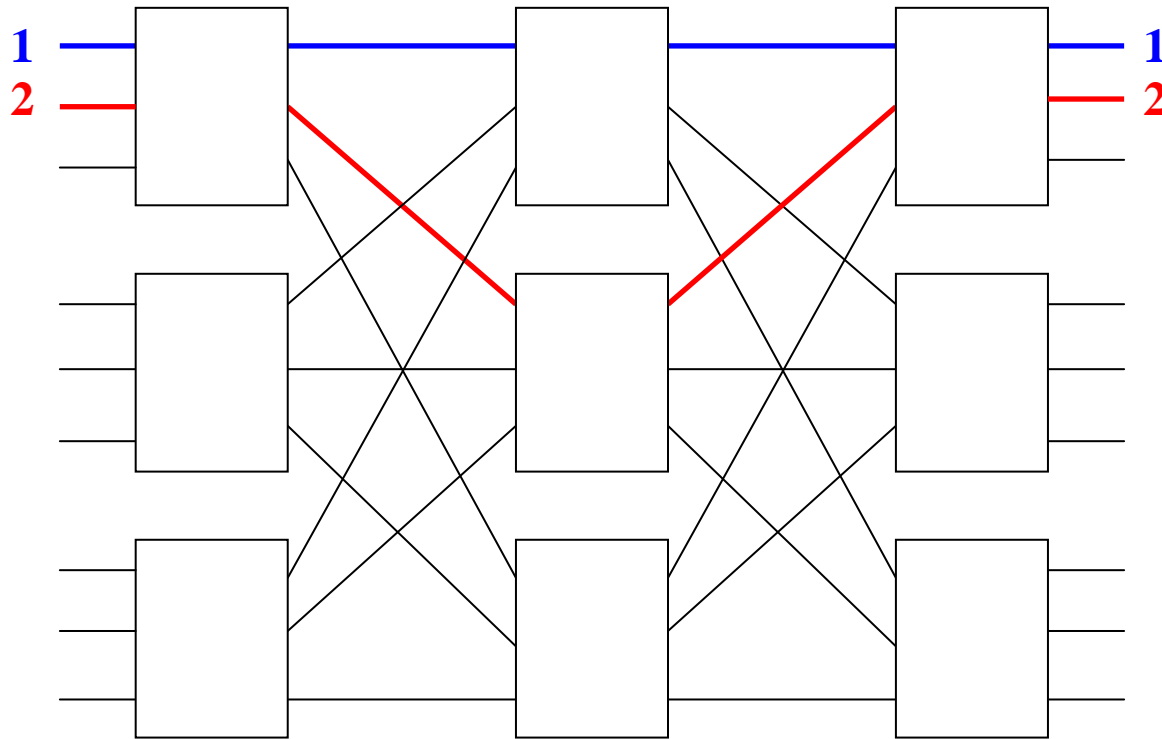
(1,1), (2,2), (4,4), (5,3), ...



Je Closova sieť s $m=n$ neblokujúca?

Príklad prepojenia v sieti:

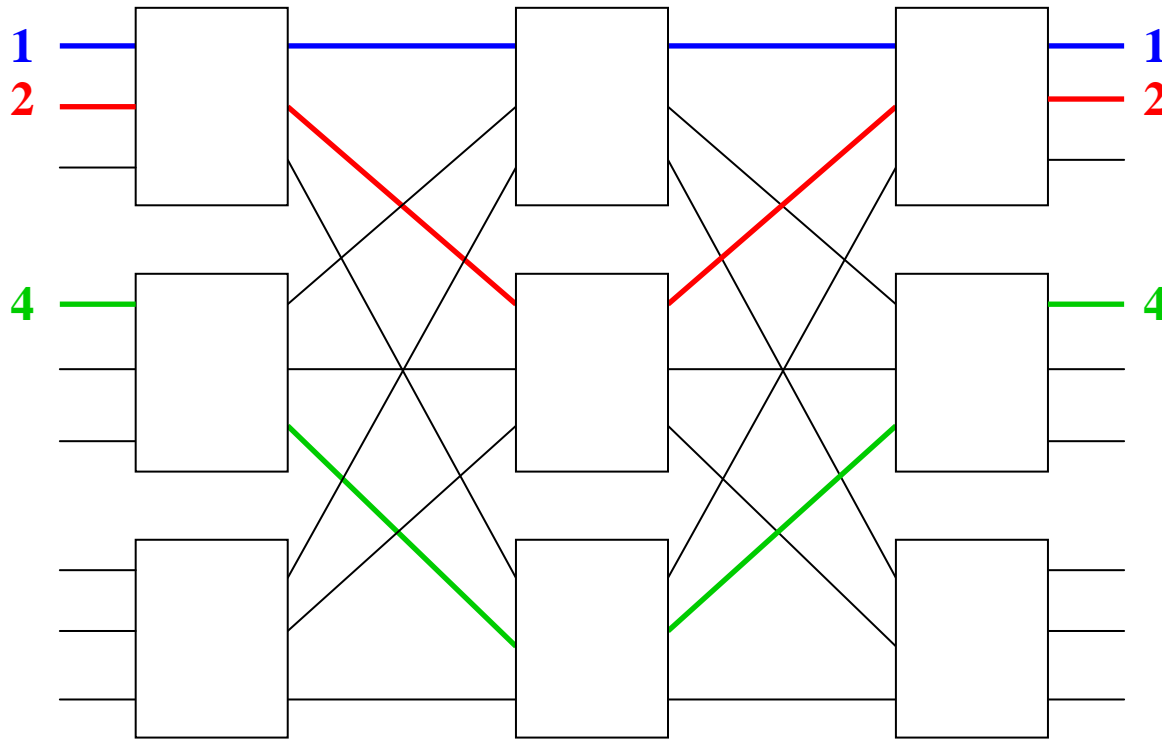
(1,1), (2,2), (4,4), (5,3), ...



Je Closova sieť s $m=n$ neblokujúca?

Príklad prepojenia v sieti:

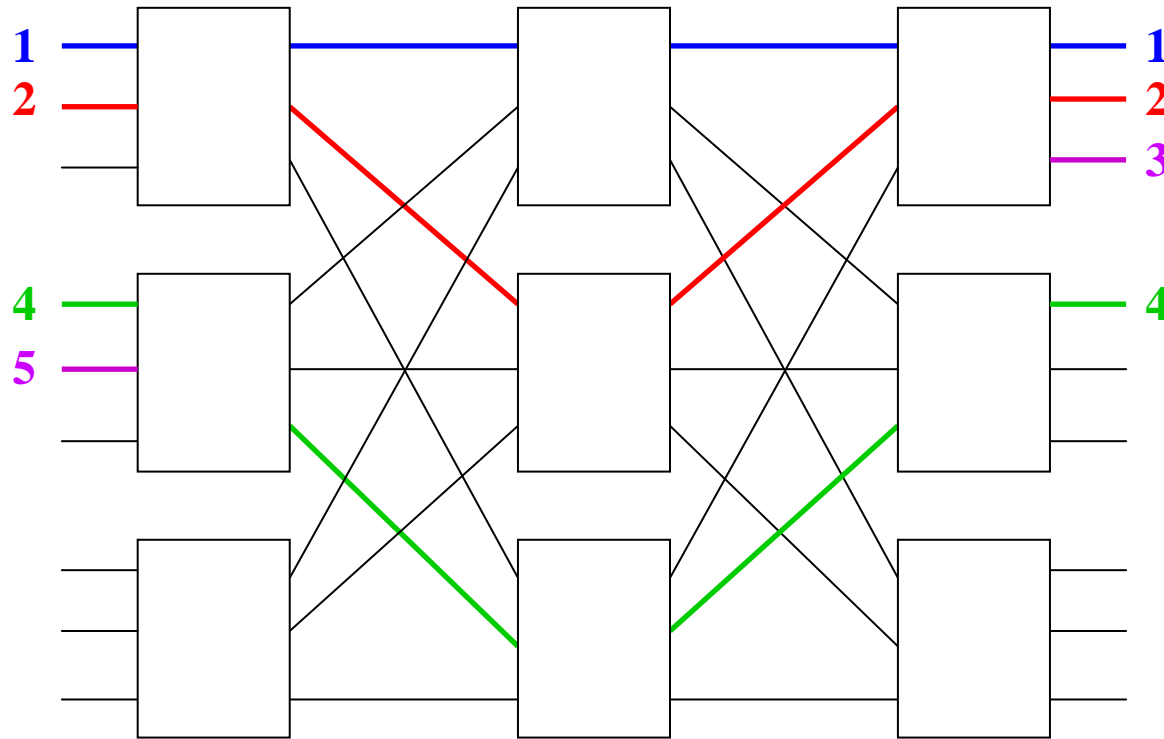
(1,1), (2,2), (4,4), (5,3), ...



Je Closova sieť s $m=n$ neblokujúca?

Príklad prepojenia v sieti:

(1,1), (2,2), (4,4), (5,3), ...

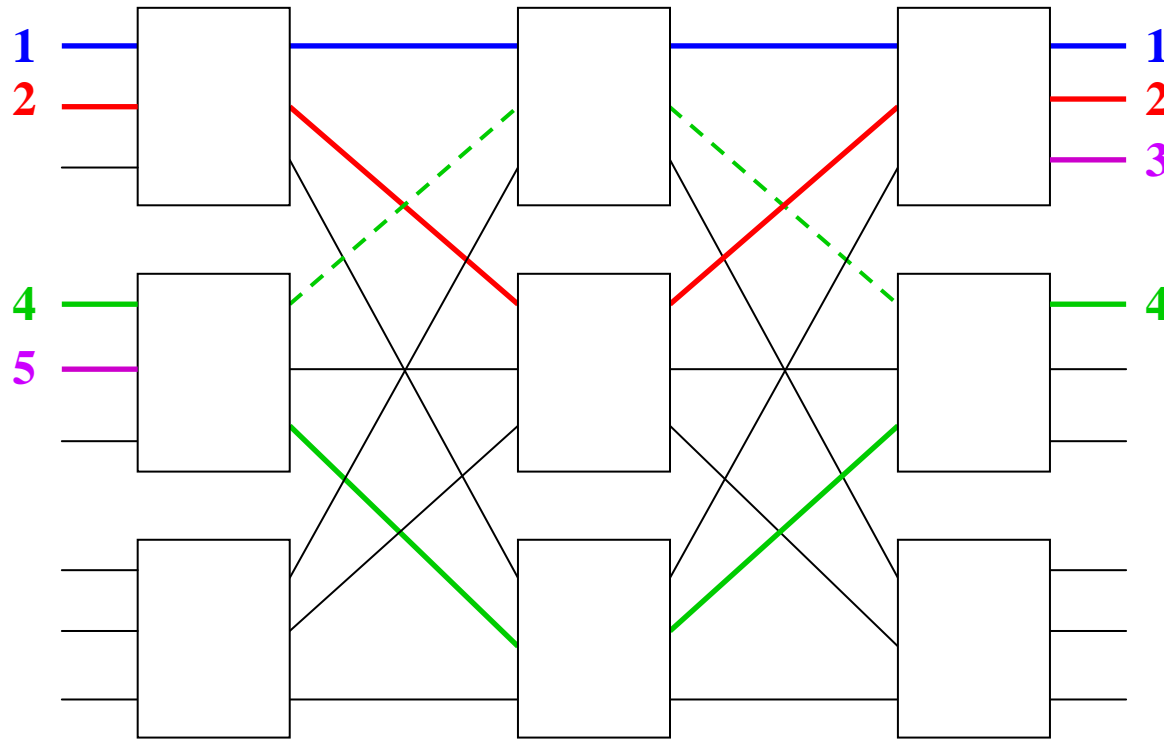


?

Je Closova sieť s $m=n$ neblokujúca?

Príklad prepojenia v sieti:

(1,1), (2,2), (4,4), (5,3), ...

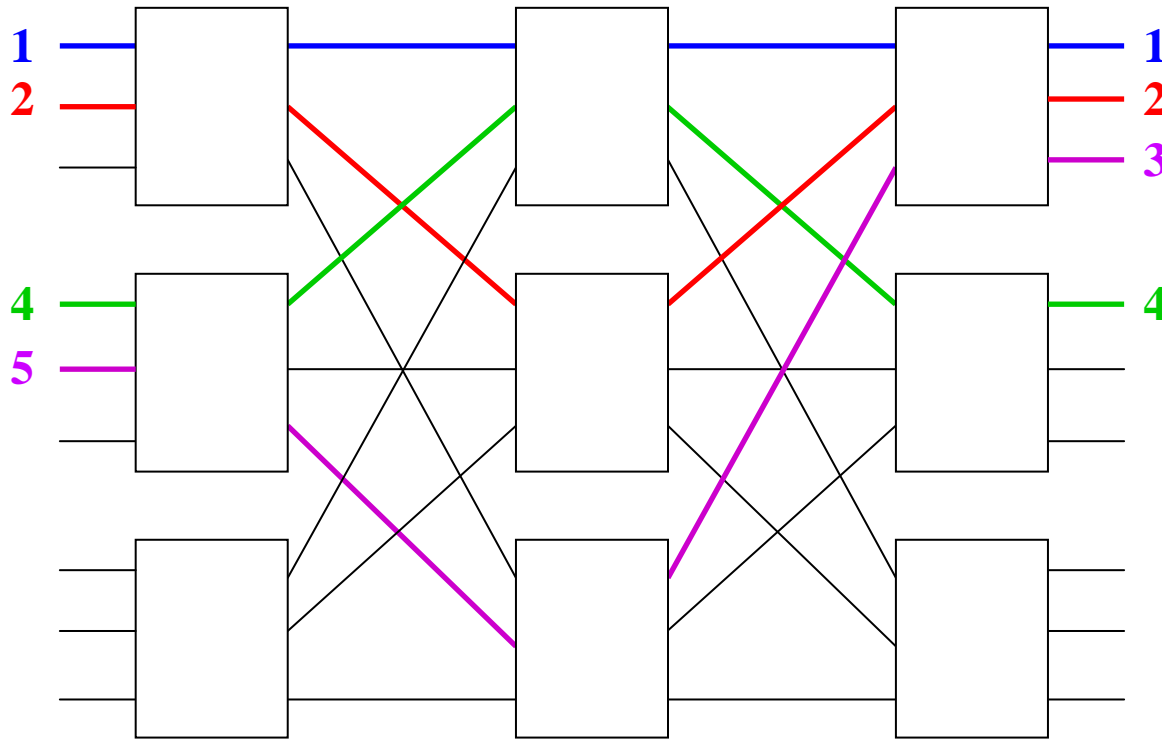


Riešením daného problému je preusporiadať už existujúce spojenia v sieti...
Napríklad spojenie (4,4)

Je Closova sieť s $m=n$ neblokujúca?

Príklad prepojenia v sieti:

(1,1), (2,2), (4,4), (5,3), ...

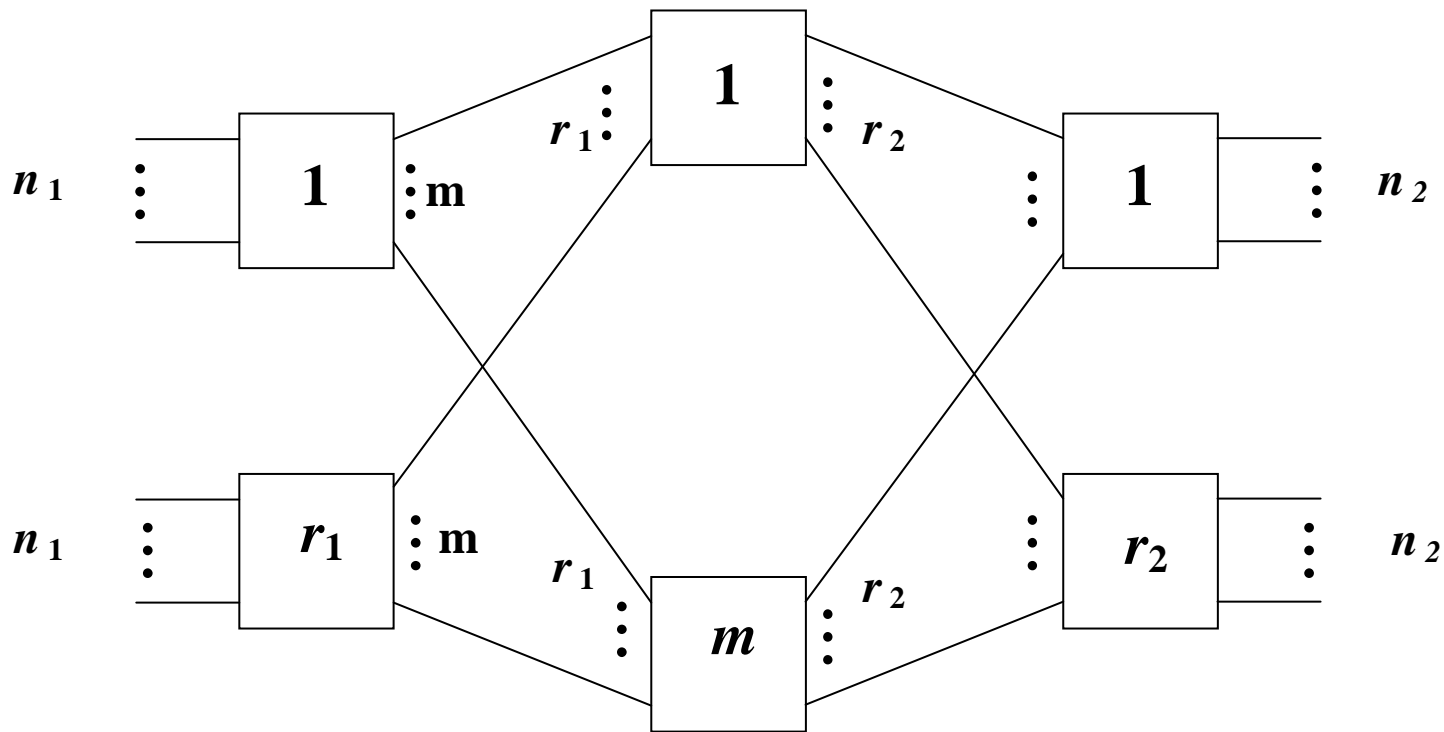


Prepájacia schopnosť Closovej siete je závislá na parametroch n , r a m . Pre dané n , r a meniace sa m je veľa možností prepájania. Neblokujúce operácie 3-stupňovej siete môžeme dosiahnuť viacej ako jedným spôsobom. Rozlišujeme 3 módy neblokujúcich Closových sietí:

- **striktne (prísne) neblokujúca sieť**
(Strictly Nonblocking – SNB), → podmienka: $m \geq 2n - 1$,
- **široko neblokujúca sieť** (Wide-Sense Nonblocking – WSN),
- **preusporiadateľná neblokujúca sieť**
(Rearrangeably Nonblocking – RNB), → podmienka: $m \geq n$

Vo všeobecnosti trojstupňová Closova sieť $C(n_1, r_1, m, n_2, r_2)$ je 3-stupňová sieť, ktorej prvý stupeň tvorí r_1 prepínačov veľkosti $n_1 \times m$, tretí stupeň má r_2 prepínačov dimenzie $m \times n_2$, a stredný stupeň tvorí m prepínačov veľkosti $r_1 \times r_2$. Ak platí $n_1 = n_2$, $r_1 = r_2$ potom hovoríme o symetrickej Closovej 3-stupňovej sieti. $C(n, r, m, n, r)$ značíme $C(m, n, r)$.

všeobecná schéma Closovej siete



Pre Closovu sieť $C(n_1, r_1, m, n_2, r_2)$ môžeme požadovaný stav siete – zadaný permutáciou P prepísať do matice o r_1 riadkoch a r_2 stĺpcoch:

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,r_2} \\ \vdots & & \vdots \\ a_{r_1,1} & \cdots & a_{r_1,r_2} \end{bmatrix}$$

Pre maticu A ďalej (nielen) platí:

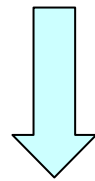
$$\sum_{i=1}^{r_1} a_{ij} = n_2 \quad (\text{pre } \forall j = \text{konšt.})$$

$$\sum_{j=1}^{r_2} a_{ij} = n_1 \quad (\text{pre } \forall i = \text{konšt.})$$

Pojem prepojovacej funkcie a súvislosť s permutáciou:

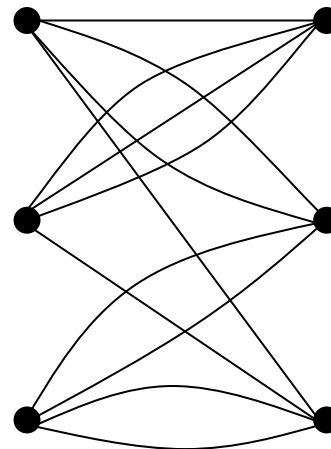
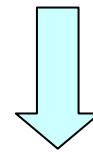
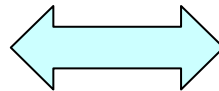
Majme permutáciu a sieť typu C(4,4,3):

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



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

prepojovacia matica



bipartitný graf

Podľa zvolenej reprezentácie vyplývajú algoritmy:

- rozklad permutačnej matice
- rozklad bipartitného grafu

Rozklad prepojovacej matice:

Rozkladom prepojovacej matice A na dielčie matice B_p , $p \in \{1, 2, \dots, q\}$, také že platí

$$A = B_1 + B_2 + \dots + B_p + \dots + B_q$$

a zároveň pre $\forall B_p$ platí

$$- b_{ij} \in \{0, 1\}$$

$$- \sum_{i=1}^{r_1} b_{ij} = 1 \quad \text{príp.} \quad \sum_{i=1}^{r_1} b_{ij} \leq 1$$

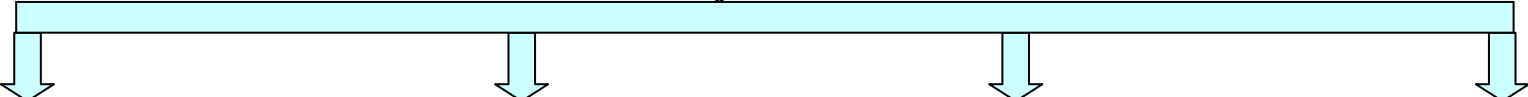
$$- \sum_{j=1}^{r_2} b_{ij} = 1 \quad \text{príp.} \quad \sum_{j=1}^{r_2} b_{ij} \leq 1$$

získame konfiguráciu prepínačov v druhom stupni siete. Pre konkrétny rozklad prepojovacej matice na dielčie matice je $q!$ rôznych nastavení prepínačov v druhom stupni.

V našom prípade:

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




$$S_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

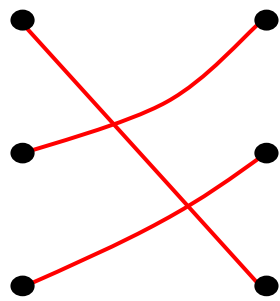
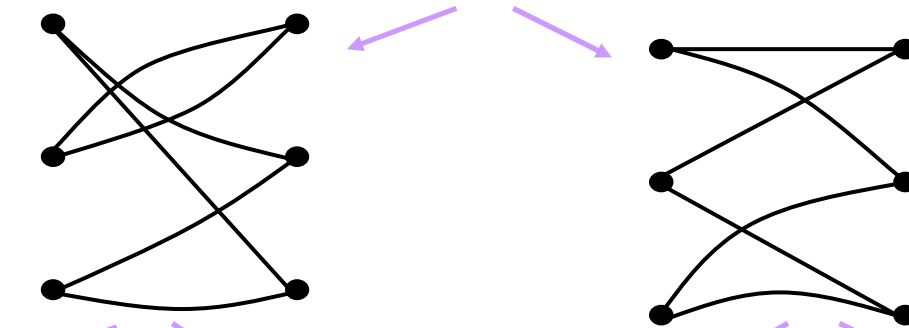
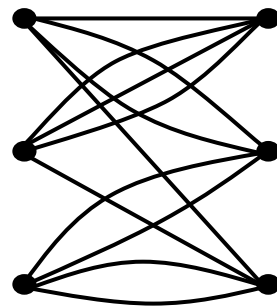
$$S_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$S_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

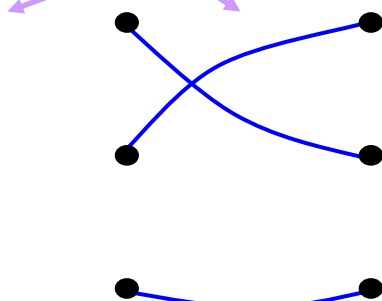
$$\mathbf{A} = \mathbf{S1} + \mathbf{S2} + \mathbf{S3} + \mathbf{S4}$$

Rozklad bipartitného grafu:



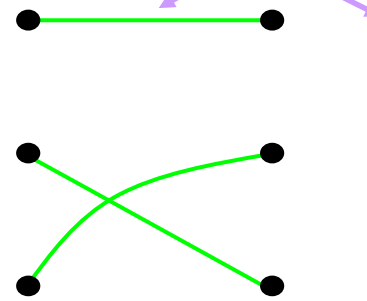
↓

$$S_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$



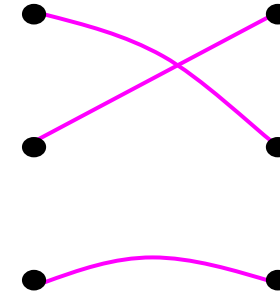
↓

$$S_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



↓

$$S_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



↓

$$S_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

→ Matice vzniknuté rozkladom (či už grafu alebo prepojovacej matice) umožňujú priamo nastaviť prepínače v strednom stupni siete.

→ Ak poznáme nastavenie prepínačov stredného stupňa nie je náročné nastaviť prepínače v celej sieti.

Príklad 2:

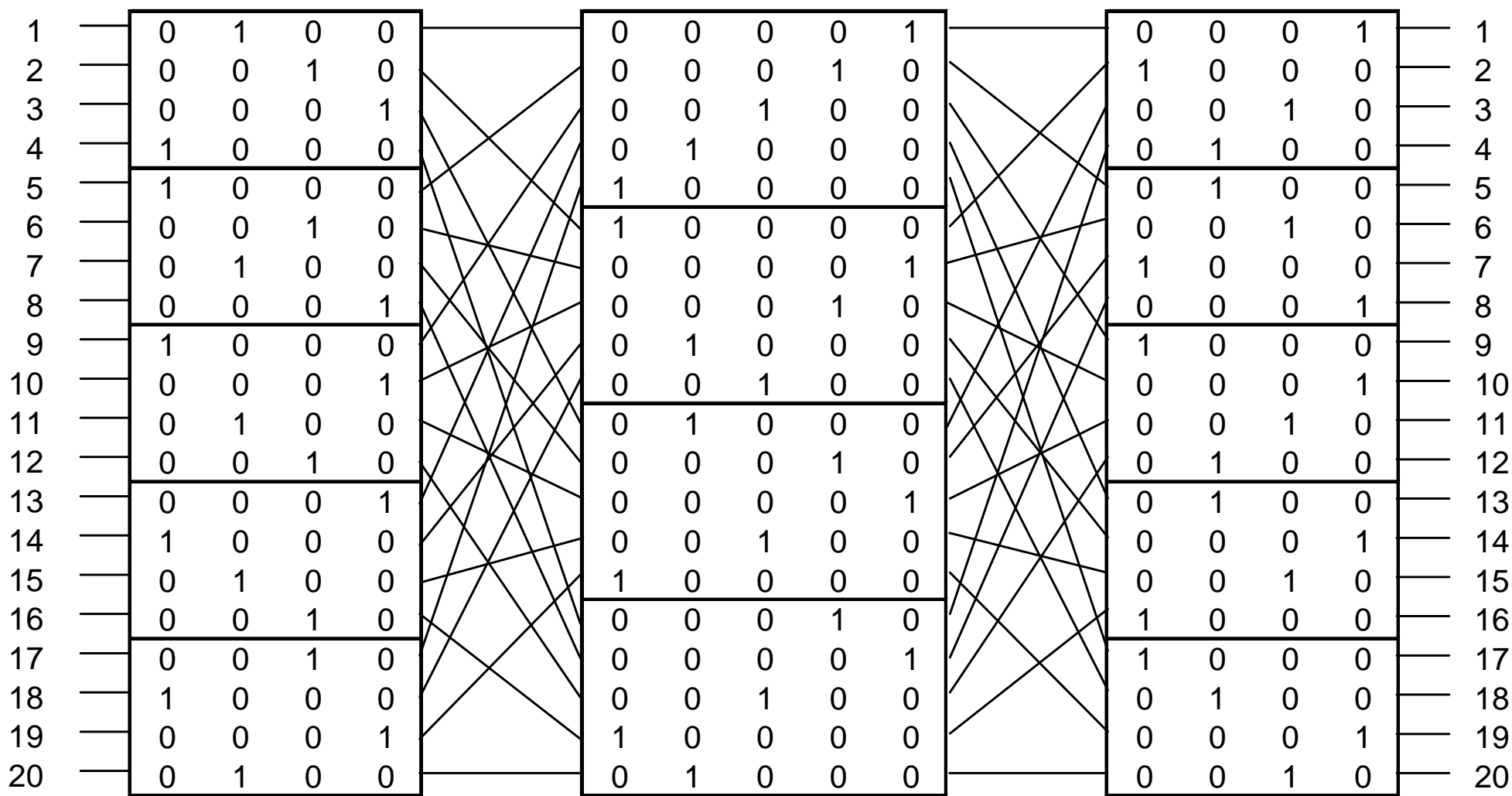
Majme symetrickú Closovu sieť $C(4,4,5)$ (t.j. $C(4,5,4,4,5)$) a permutáciu

$$P = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 1 & 5 & 13 & 17 & 14 & 15 & 18 & 19 & 9 & 10 & 16 & 20 & 2 & 6 & 7 & 11 & 3 & 4 & 8 & 12 \end{pmatrix}$$

potom prepojovacia matica bude:

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 1 & 1 \\ 1 & 2 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Čomu zodpovedá nastavenie siete podľa nasledujúceho obrázku:

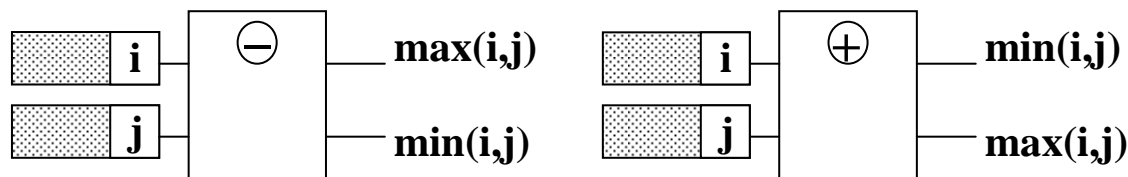


Pozn: Benešova sieť je špeciálny prípad Closovej siete. Preto ak chceme zabezpečiť bezkonfliktnosť Benešovej siete môžeme využiť vyššie uvedené algoritmy pre Closovu sieť.

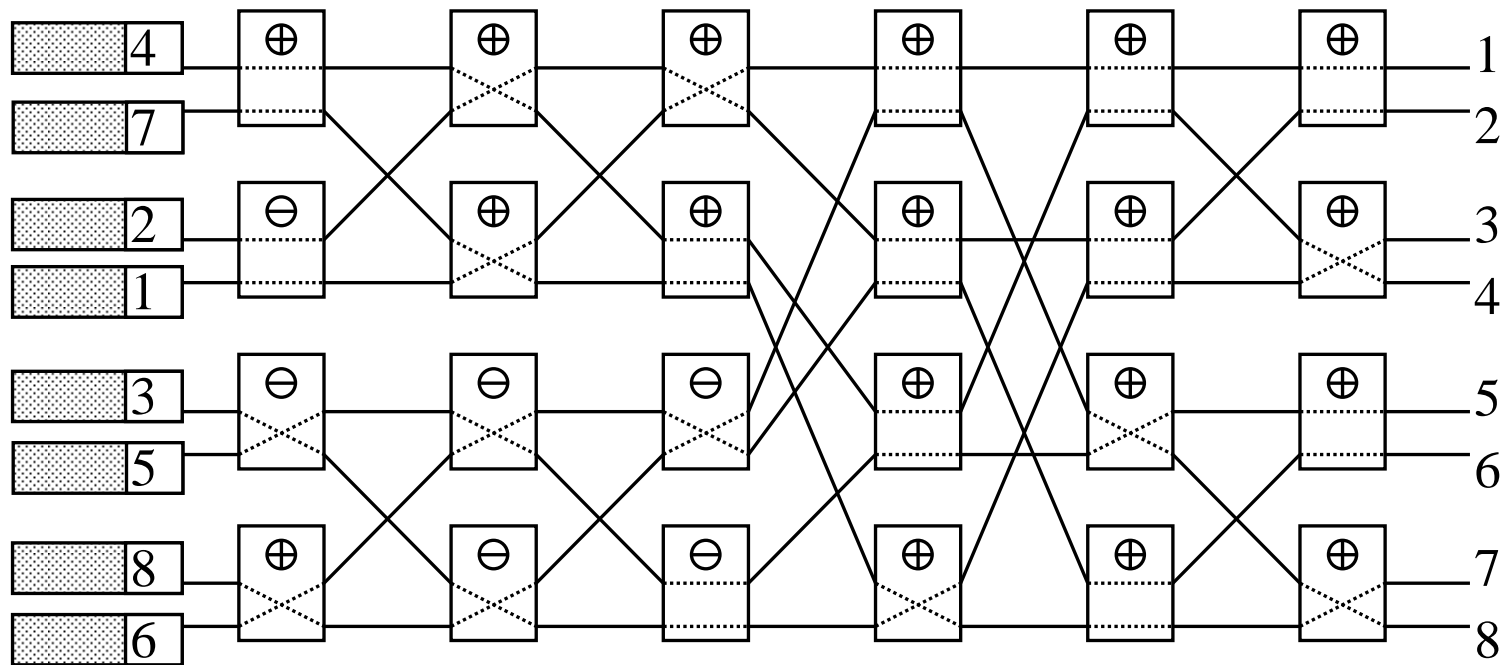
Batcherova sieť

Batcherova, alebo triediaca (sorting) sieť triedi vstupujúce pakety podľa ich adresy výstupu a to od najmenších adries k najväčším.

Batcherova triediaca sieť je zložená z elementov, ktoré sú znázornené na obr.



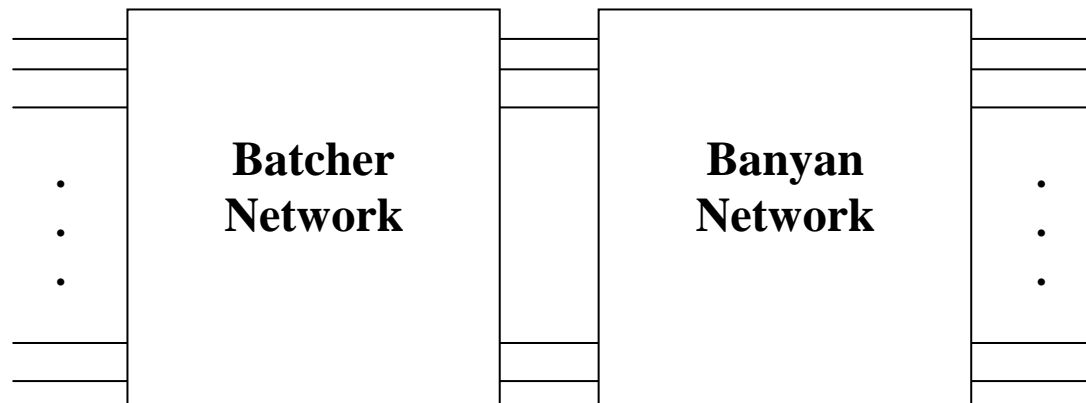
Element má dva vstupy a dva výstupy. Existujú dva druhy elementov. Element označený ako $-$ prepojí na vrchný výstup paket s vyššou adresou a na spodný výstup paket s nižšou adresou (a). Element označený ako $+$ zoradí výstupy opačne (b). Vhodným zoradením takýchto elementov dostaneme Batcherovu sieť. Celá triediaca sieť je zložená z postupnosti bitonických triedičov (bitonic sorters), ktoré triedia svoje výstupy zostupne, alebo vzostupne. V našom prípade sú to triediče veľkosti 2×2 . Ak je na vstupe elementu jeden len paket a teda len jedna adresa, uvažuje sa ako by mala nižšiu hodnotu.



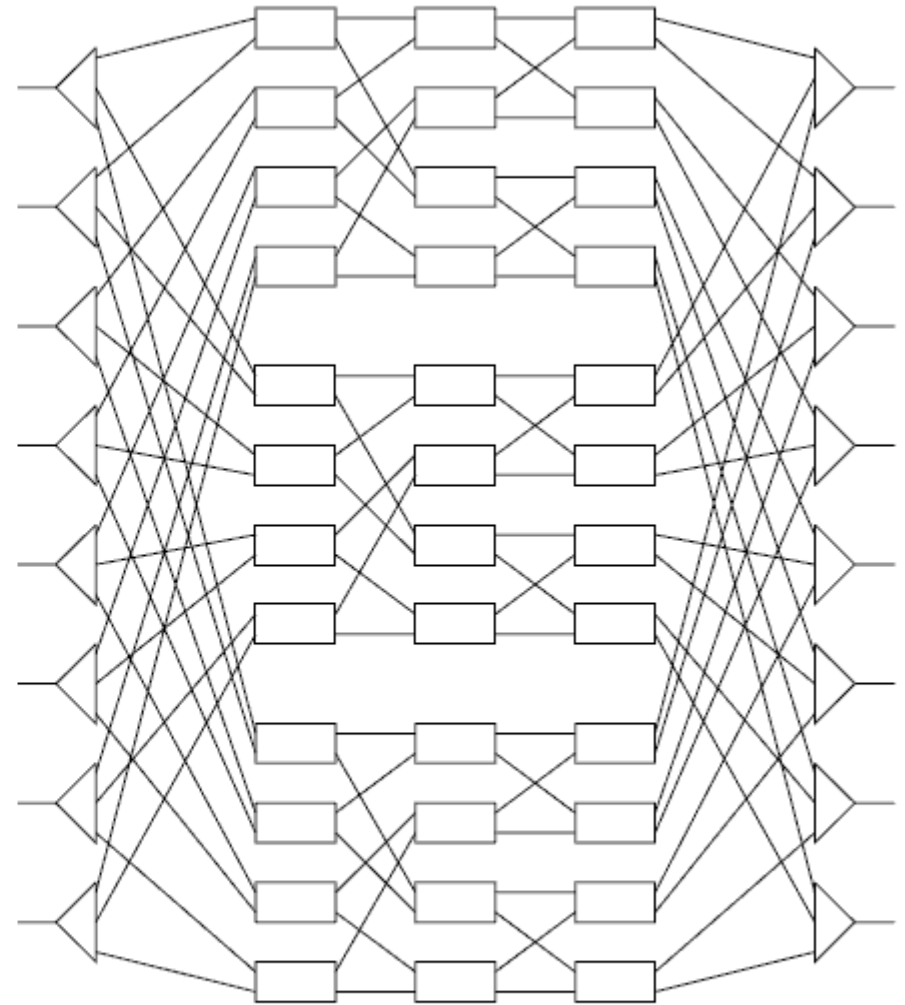
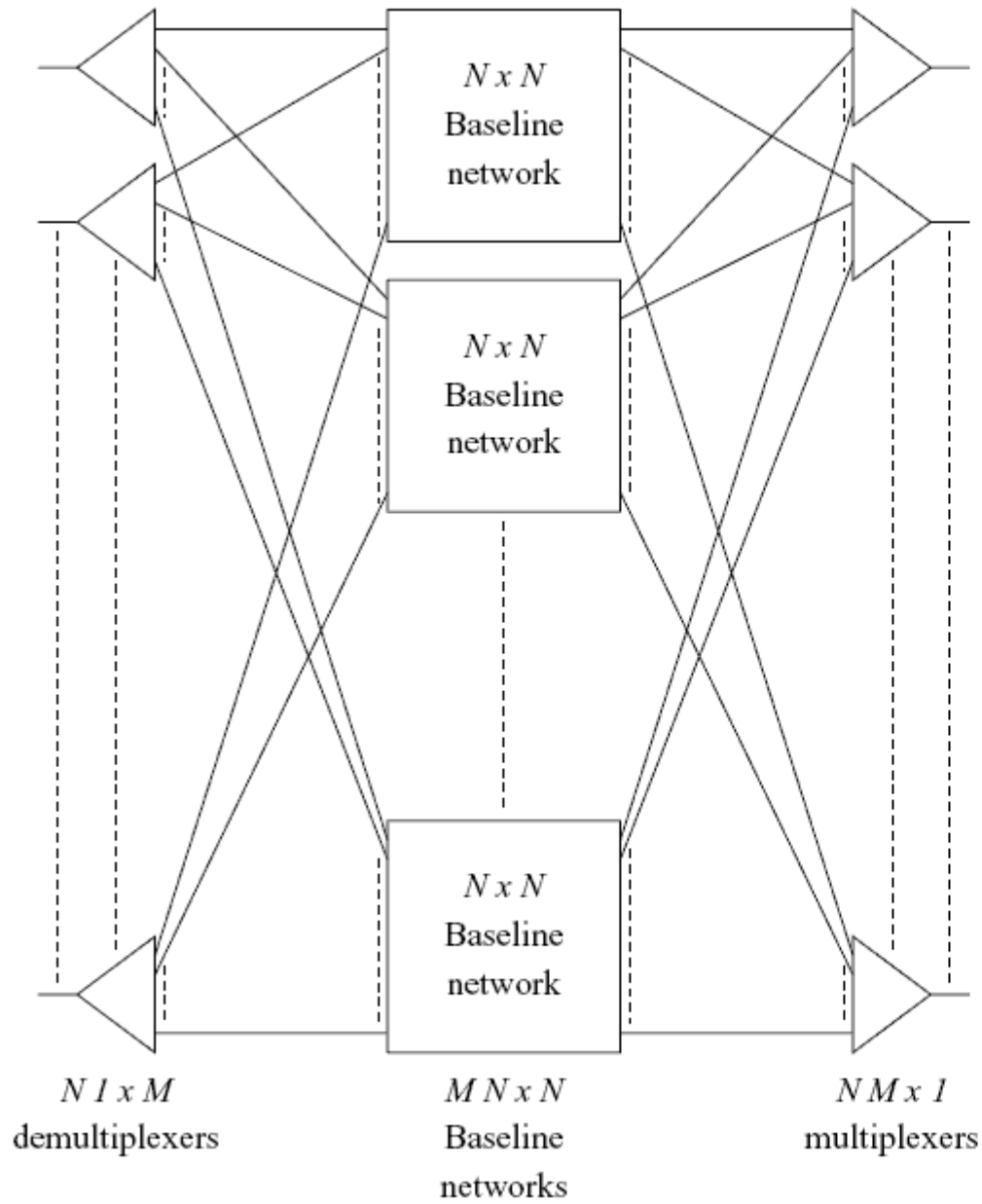
Na výstupe Batcherovej siete sú pakety zoradené podľa vzostupných adries, ale nedosiahnu správny výstup podľa svojej adresy určenia (ak nie sú obsadené všetky vstupy). Preto sa Batcherova sieť ďalej kombinuje s banyan sieťou.

Batcher-Banyan sieť

Banyan sieť zapojená za Batcherovou sieťou má samosmerovacie vlastnosti a dopraví tak pakety k správneému výstupu zo siete. Zaradenie triediacej siete pred banyan sieťou vylúči blokovanie HOL ak predpokladáme, že žiadne dva pakety nie sú smerované k tomu istému výstupu. Vzhľadom na to, že pakety sú pretriedené v triediacej sieti, nevznikne ani vnútorné blokovanie v banyan sieti. V prípade ak existuje možnosť rovnakých výstupných adries na vstupe siete, je potrebné použiť vyrovnávacie pamäti.



Paralelná Baseline sieť



Na vstupe v výstupe siete paralelnej baseline siete (niekedy tiež označovanej ako Multi-Log₂N network) je realizovaná funkcia expanzie a koncentrácie. Na vstupe siete sú spojovacie elementy 1×m, ktoré expandujú prevádzku a na výstupe elementy m×1, ktoré prevádzku koncentrujú.

Ak nerátame vstupné a výstupné stupne (expanzia a koncentrácia), počet stupňov v sieti je $n = \log_2 N$. Z teórie grafov je možné dokázať, že ak je počet paralelných baseline podsietí

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

potom paralelná baseline sieť je rekonfigurovateľná bez blokády. To zároveň znamená, že pre optimálnu paralelnú baseline sieť potrebujeme viac spojovacích elementov ako pre sériovú baseline sieť (Benešovu sieť). Na druhú stranu, čas prechodu paralelnou baseline sieťou je menší ako pri sériovej baseline.

Cantor network

Podobne ako Closova sieť, aj Cantorova sieť patrí medzi striktné neblokujúce. Cantorova sieť $N \times N$ môže byť vytvorená z $\log_2 N$ Benešových sietí, N demultiplexorov a N multiplexorov.

