

Logická funkce

- Logická funkce je funkce, která pro konečný počet vstupních parametrů vrací logické hodnoty.
- Přiřazuje kombinaci vstupních logických proměnných nějakou výstupní kombinaci.
- Používá se v oboru teorie řízení a číslicové techniky, v praxi pak například v mikroprocesorové technice.
- Parametry logické funkce jsou logické proměnné. ...

Logická proměnná je taková proměnná, která může nabývat nějakého konečného počtu logických hodnot.

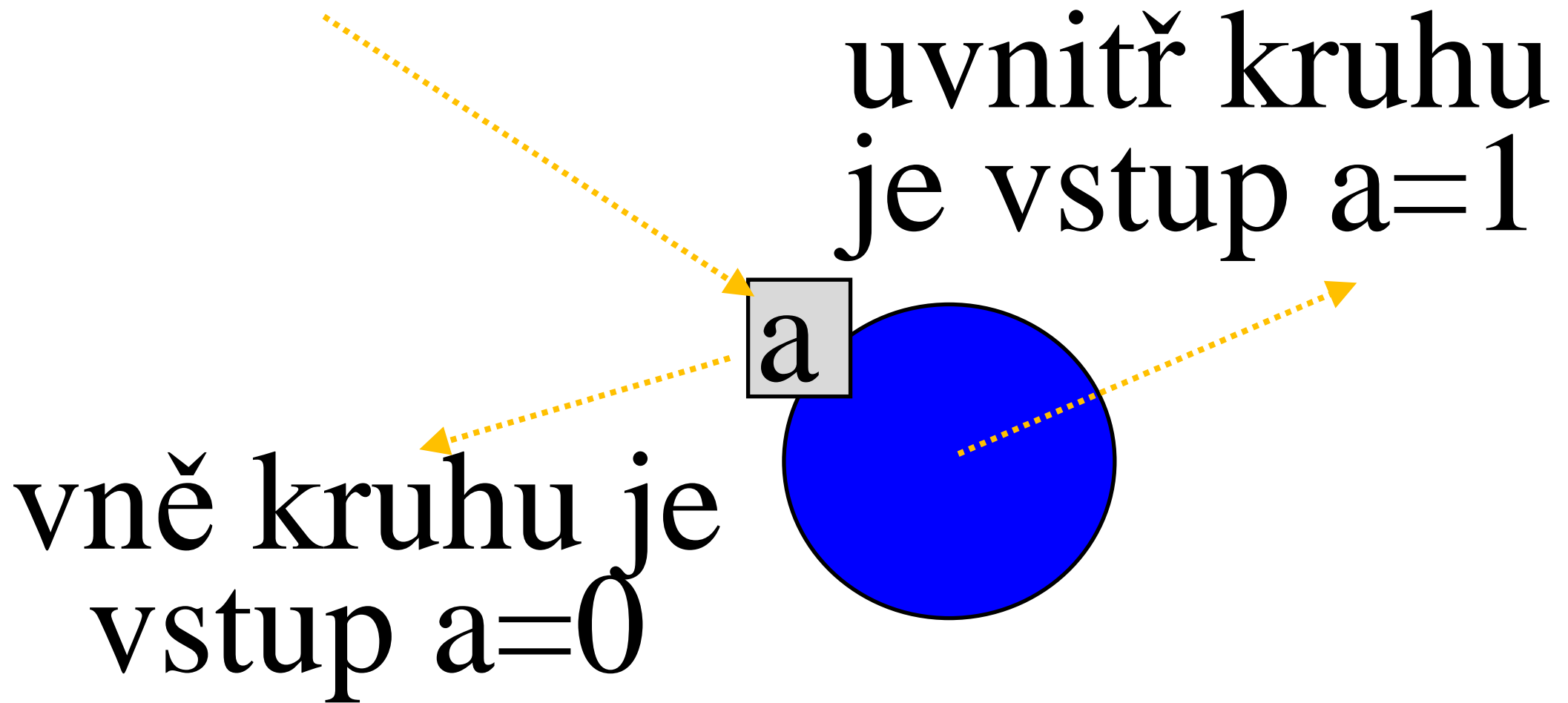
Nejčastěji jsou tyto hodnoty dvě a reprezentují stavy **pravda** a **nepravda**.

Pravda se vyjadřuje jako $1 =$ **logická jednička**

Nepravda se vyjadřuje jako $0 =$ **logická nula**

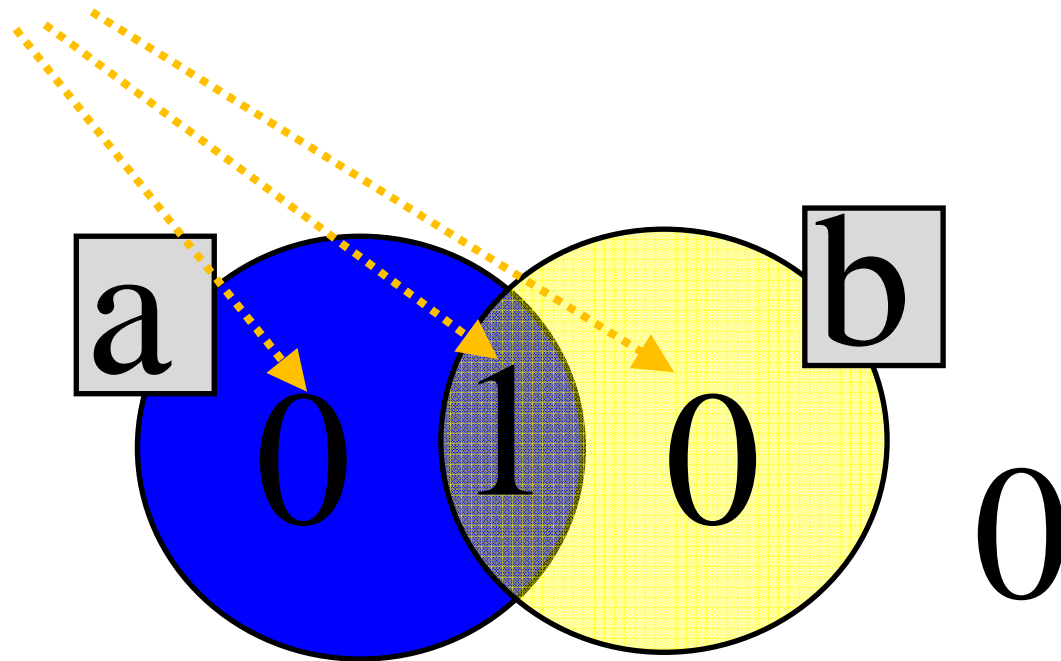
Vennův diagram

Vstupní proměnnou zaznameneáme kruhem (příp.jiným obrazcem), vnitřek představuje hodnotu 1



Vennův diagram

Výsledná hodnota je pro danou kombinaci je označena uvnitř kruhu.



Čtu: funkce bude rovna 1, pokud $a=1$ a zároveň $b=1$.

Logická funkce

Logická operace je taková operace s výroky, jejíž výsledkem je opět výrok, jehož pravdivostní hodnota (PRAVDA nebo NEPRAVDA) závisí na pravdivosti.

Booleova algebra je algebraická struktura, která zobecňuje vlastnosti množinových a logických operací.

Logická funkce

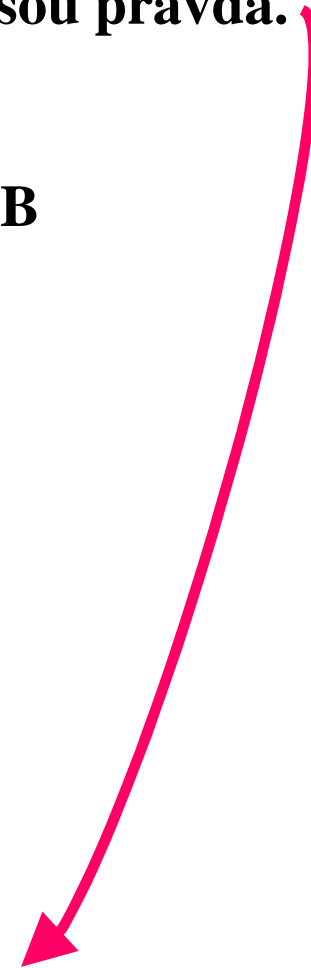
Logická konjunkce (symboly \wedge AND, &) je binární logická operace jejíž hodnota je pravda, právě když obě vstupní hodnoty jsou pravda.

Zápis:

$X = A \text{ AND } B$, nebo $X = A \& B$, $X = A \wedge B$, $X = A * B$

Mohu vyjádřit **pravdivostní tabulkou**:

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1



Logická funkce

Disjunkce v logice znamená logický součet. Výroky jsou spojeny symbolem *OR* nebo \vee) Logický součet nabývá hodnotu pravda , když alespoň jeden z obou vstupních výroků je pravda.

Zápis:

$X = A \text{ OR } B$, nebo $X = A \vee B$, někdy $X = A + B$

Mohu vyjádřit **pravdivostní tabulkou**:

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1



Logická funkce

Logická **negace** (používá se pro ni NOT, popř. se označuje pruhem nad proměnnou) hodnota je nepravda, právě když první vstupní hodnota je pravda a naopak.

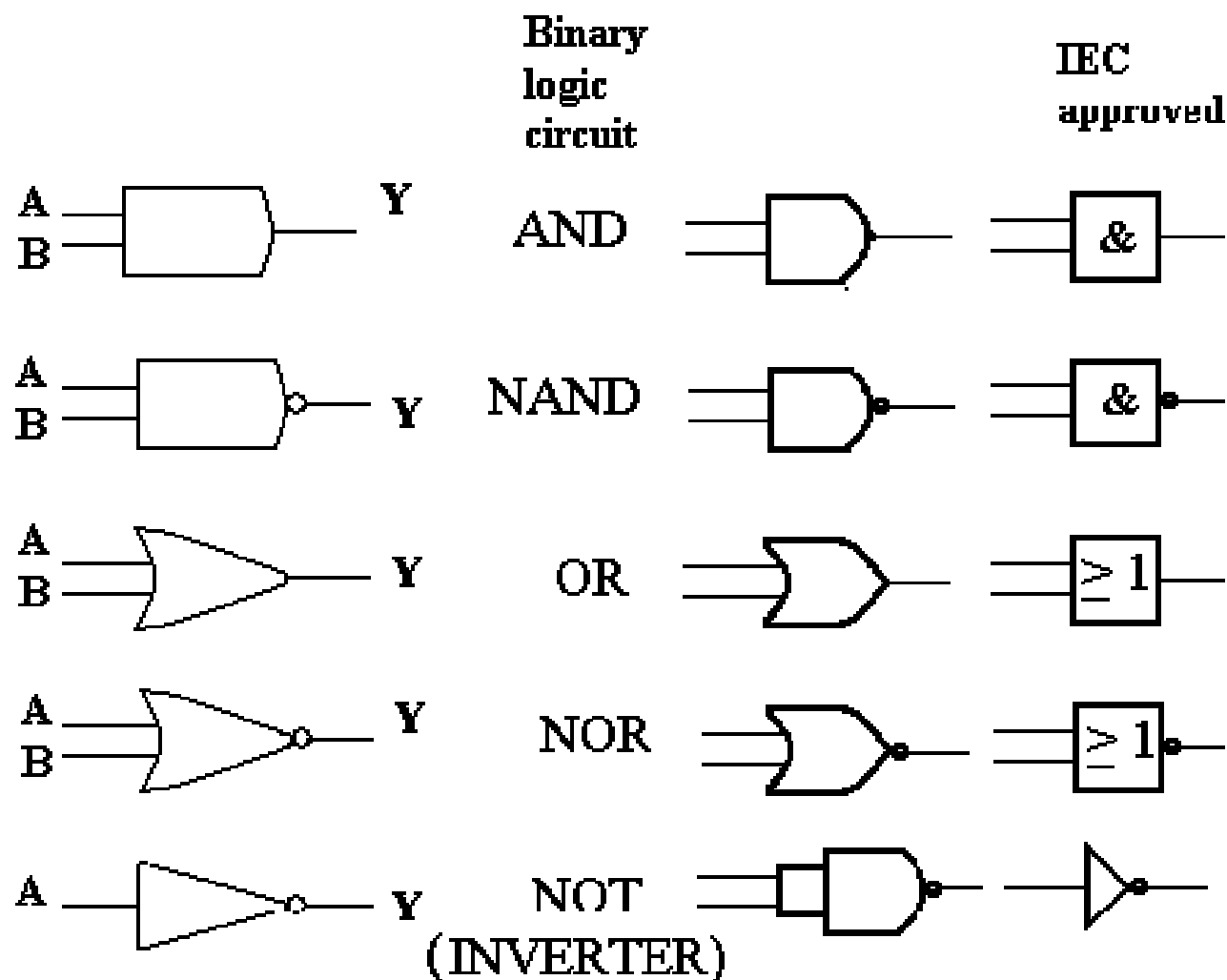
Logická **negace** (používá se pro ni NOT, popř. se označuje pruhem nad proměnnou) hodnota je nepravda, právě když první vstupní hodnota je pravda a naopak.

A	$\neg A$
0	1
1	0

Logické funkce

Používané schematické
značky:

Logic symbols
(IEC = International Electrochemical Commission)



Zákony Booleovy algebry

Booleova algebra

- minimalizace logických funkcí pomocí zákonů a pravidel

požadované znalosti:

- celkový přehled o možnosti zápisu
- umět si odvodit proč $a+1=1$, $a*0=0$... a podobně
- znát de Morganova pravidla

	Zákony součtu	Zákony součinu
Zákon idempotence	$a + a = a$	$a.a = a$
Zákon absorbce	$a + ab = a$	$a.(a + b) = a$
Zákon absorbce negace	$a + \bar{a}.b = a + b$	$a.(\bar{a} + b) = ab$
Zákon komutativní	$a + b = b + a$	$ab = b.a$
Zákon asociativní	$a + (b + c) = (a + b) + c$	$a.(b.c) = (ab).c$
Zákon distributivní	$a + (b.c) = (a + b).(a + c)$	$a.(b + c) = ab + ac$
Neutrálnost nuly a jedničky	$a + 0 = a$	$a.1 = a$
Agresivnost nuly a jedničky	$a + 1 = 1$	$a.0 = 0$
Zákon vyloučeného třetího	$a + \bar{a} = 1$	$a.\bar{a} = 0$
Zákon negace	$\bar{\bar{0}} = 1$	$\bar{\bar{1}} = 0$
Zákon dvojité negace	$\bar{\bar{a}} = a$	
De Morganovy zákony	$\overline{a + b} = \bar{a}\bar{b}$	$\overline{a.b} = \bar{a} + \bar{b}$

▪ Zákon de Morganův

$$\overline{a + b} = \bar{a} \times \bar{b}$$

$$\overline{a \times b} = \bar{a} + \bar{b}$$

Výčtový způsob zadání logické funkce

číslo řádku	c	b	a	Y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Pravdivostní tabulka nám říká, že zorněná log.funkce je rovna "1" pokud proměnné "a", "b", "c" budou mít kombinaci uvedenou v řádcích 1, 2, 5, 6.

Stačí tedy vypsát řádky, kde bude funkce rovna "1".

$$f = \Sigma (1, 2, 5, 6)$$

Pozor: řádky musí být srovnané dle binární hodnoty od nuly

Karnaughova mapa

c	b	a	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

	1		

v těchto políčkách má vstupní proměnná hodnotu "1"

do políček zapisujeme hodnotu kterou má funkce při daných vstupních hodnotách

Zjednodušení logické funkce

Karnaughova mapa je metoda používaná pro minimalizaci logické funkce při její analýze. Jejím principem je zobrazení n-rozměrné tabulky hodnot do dvojrozměrné mapy. Z této mapy lze poté graficky vyčíst minimální funkci.

- ⇒ Každé okénko odpovídá kombinaci vstupních proměnných.
- ⇒ Jen jedna proměnná se mění u sousedního okénka.
- ⇒ Uvnitř okénka je výstupní hodnota, při této kombinaci vstupních proměnných.

		A		B	
C	D	$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$	$A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$	$A \cdot B \cdot \bar{C} \cdot \bar{D}$	$\bar{A} \cdot B \cdot \bar{C} \cdot \bar{D}$
		$\bar{A} \cdot \bar{B} \cdot C \cdot \bar{D}$	$A \cdot \bar{B} \cdot C \cdot \bar{D}$	$A \cdot B \cdot C \cdot \bar{D}$	$\bar{A} \cdot B \cdot C \cdot \bar{D}$
		$\bar{A} \cdot \bar{B} \cdot C \cdot D$	$A \cdot \bar{B} \cdot C \cdot D$	$A \cdot B \cdot C \cdot D$	$\bar{A} \cdot B \cdot C \cdot D$
		$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D$	$A \cdot \bar{B} \cdot \bar{C} \cdot D$	$A \cdot B \cdot \bar{C} \cdot D$	$\bar{A} \cdot B \cdot \bar{C} \cdot D$

		A		B	
C	D	0000	0001	0011	0010
		0100	0101	0111	0110
		1100	1101	1111	1110
		1000	1001	1011	1010

		A		B	
C	D	0	1	3	2
		4	5	7	6
		12	13	15	14
		8	9	11	10

- vytvářím podmapy ze sousedních políček
- podmapa představuje člen součtu
- každý člen součtu je násobek vstupních proměnných, které představuje políčko
- pokud podmapa představuje přímý i negovaný tvar vstupní proměnné, proměnná vypadne
- snažím se vytvořit co největší podmapy
- políčko mohu využít pro víc podmap

Zjednodušení logické funkce

Příklad vytváření podmap.

Pozor: i políčka vyznačená fialově jsou sousední

X jsou stavy u kterých nezáleží na výsledku - mohou je použít jak potřebuji

				A	
				B	
		0	1	X	0
		1	0	X	1
		1	1	X	X
		0	0	X	X
C	D				

Zjednodušení logické funkce

Pravdivostní tabulka (zadání):

$$Q = (\bar{x} \cdot \bar{y} \cdot \bar{z}) + (\bar{x} \cdot y \cdot \bar{z}) + (x \cdot \bar{y} \cdot \bar{z})$$

Odovídá:

x	y	z	Q
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Řešení:

poskydává:

				y	
				z	
x	1	0	0	1	
	1	0	0	0	

ní:

				y	
				z	
x	1	0	0	1	
	1	0	0	0	

 +

				y	
				z	
x	1	0	0	1	
	1	0	0	0	

$Q_1 = \bar{y} \cdot \bar{z}$
 $Q_2 = \bar{x} \cdot \bar{z}$

Řešení:

$$Q(x, y, z) = (\bar{y} \cdot \bar{z}) + (\bar{x} \cdot \bar{z})$$

Kombinační logické obvody

- logické obvody, ve kterých stavy na výstupech závisí pouze na okamžitých kombinacích vstupních proměnných a nezávisí na jejich předchozích hodnotách, s výjimkou krátkého přechodového děje
- nemají žádnou paměť předchozích stavů, takže jedné kombinaci vstupních proměnných odpovídá právě jediná výstupní kombinace funkčních hodnot
- závislost výstupních funkčních hodnot na hodnotách vstupních proměnných popisuje pravdivostní tabulkou nebo pomocí logických výrazů

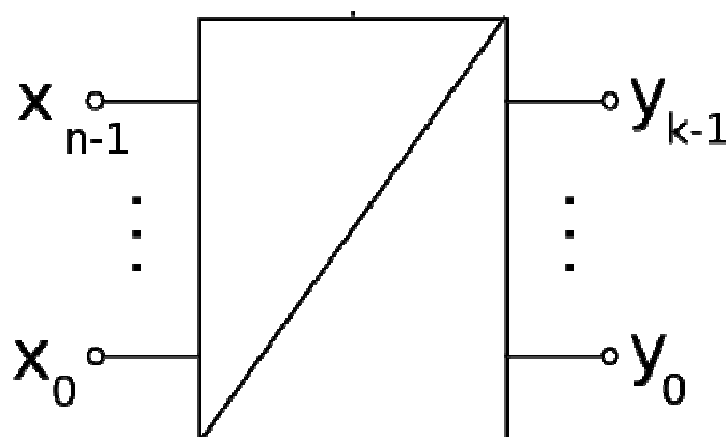
Pro realizaci kombinačních obvodů je možné použít:

- pevné paměti
- programovatelná logická pole
- základní logické členy: NAND, AND, NOR, OR, apod.

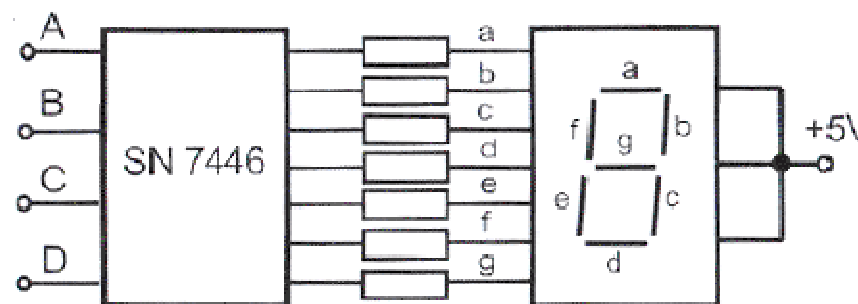
Kombinační logické obvody

Kodér

- kombinační logický obvod, který na základě kombinační tabulky z kombinace vstupních dat (x), vstupního kódu vytváří na výstupu (y) kód jiný. Funkce kodéru je inverzní k funkci dekodéru
- má n vstupních signálů $x_0 \dots x_{n-1}$ jejichž kombinace vytváří na k výstupech $y_0 \dots y_{k-1}$ jinou kombinaci signálů. Obecně platí, že $n > k$. Volitelně je možno použít strobovací výstup s pro vzorkování signálu nebo signál e pro uvolnění



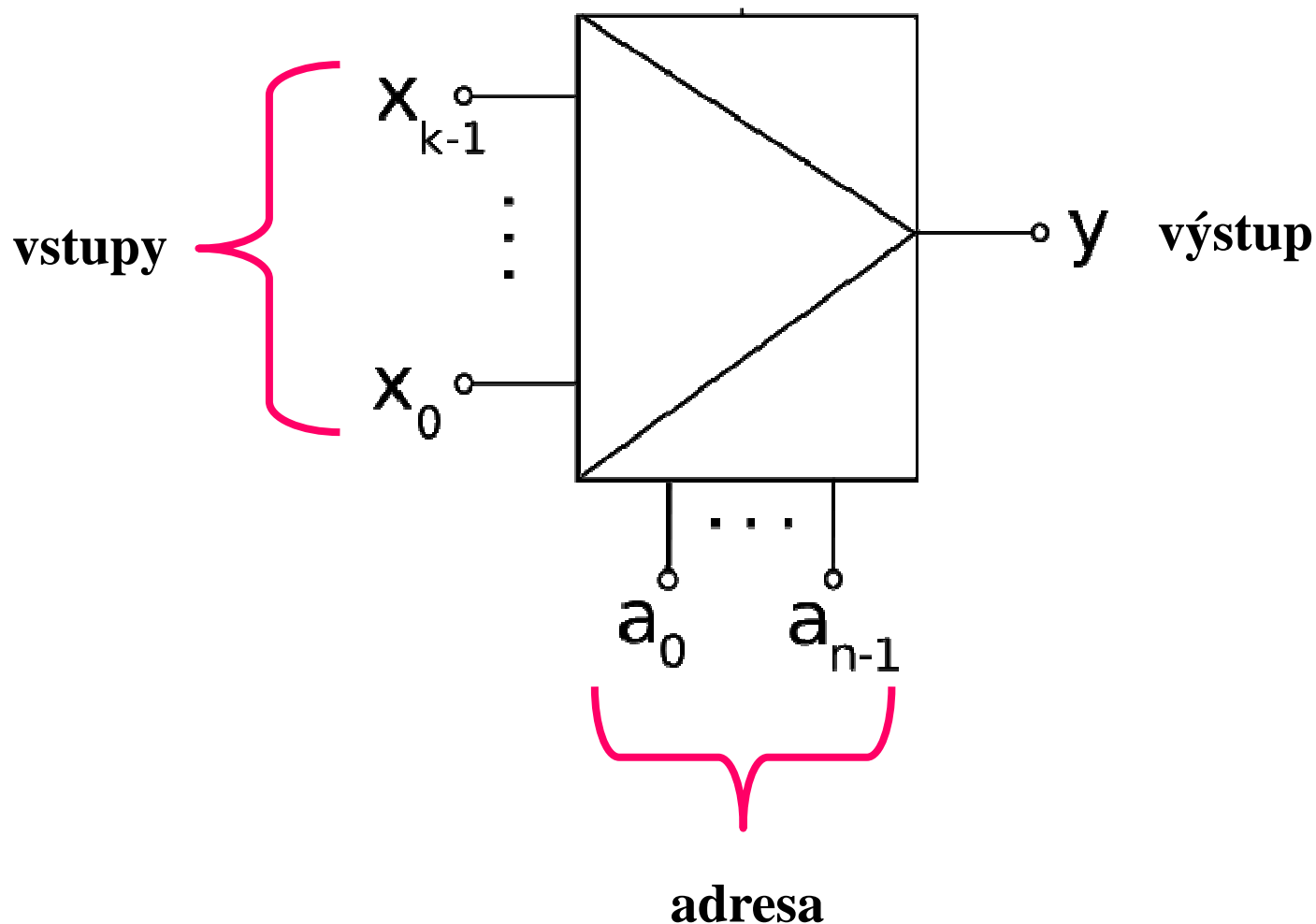
**Příklad - kodér hexadecimální/
sedmissegmentovka:**



Kombinační logické obvody

Multiplexer

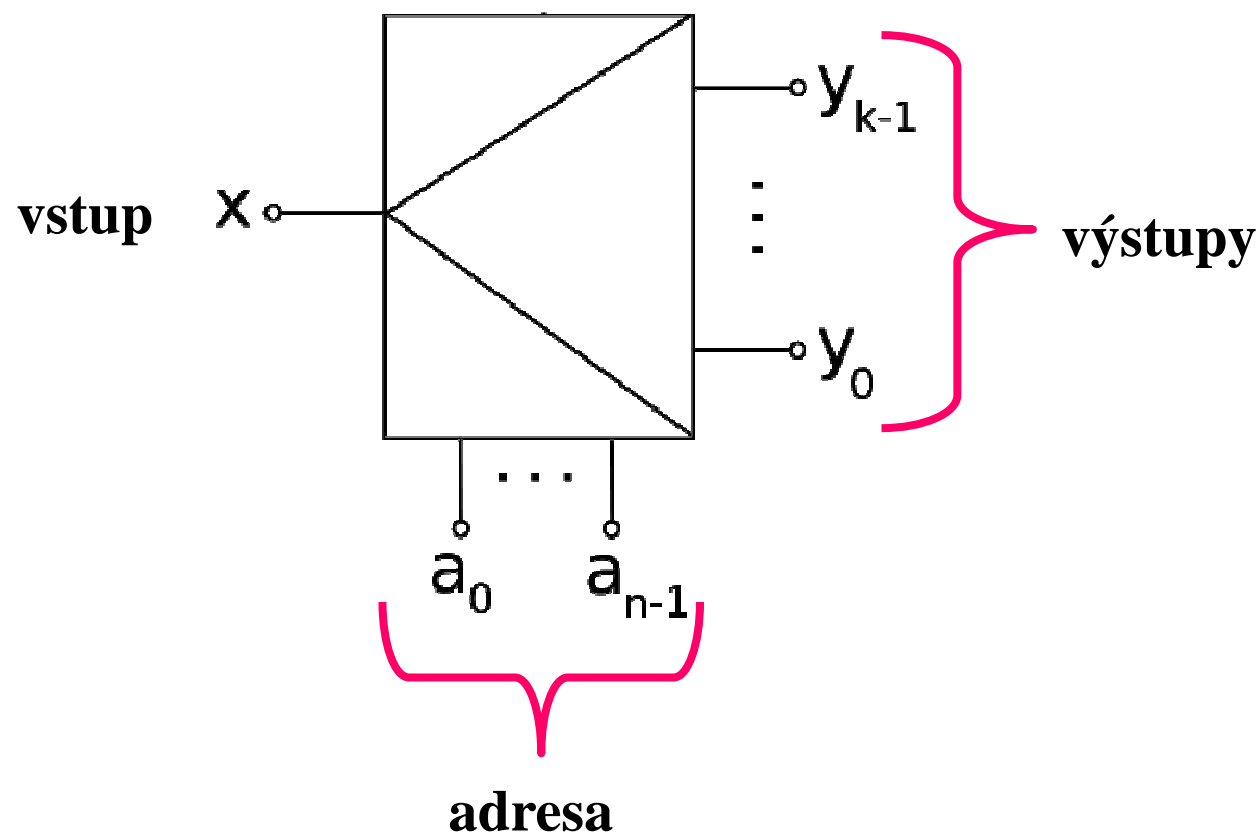
je elektronický člen, fungující na principu přepínače, kdy je podle řídicích signálů (a) přiváděn na výstup (y) jeden ze vstupních signálů (x)



Kombinační logické obvody

Demultiplexer

je elektronický člen, fungující na principu přepínače, kdy je podle řídicích signálů (a) přiváděn na výstup (y) jeden ze vstupních signálů (x)



Kombinační logické obvody

Sčítačka

- kombinační obvod
- obvod, který sčítá 2 a více čísel

Ve dvojkové soustavě se provádí sčítání podle klíčů:

Úplná má

- vstupy sčítaných bitů a_1 , b_1 a přenos c
- výstupy součet s_1 a přenos c

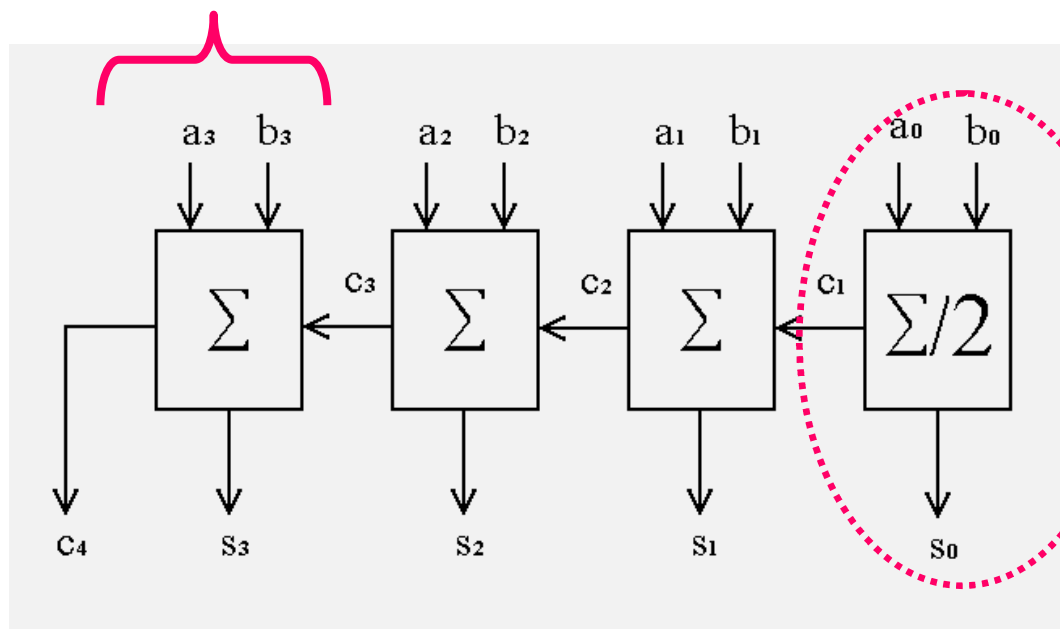
$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

↖ přenos do vyššího řádu



Poloviční má

- vstupy sčítaných bitů a_1 , b_1
- výstupy součet s_1 a přenos c

Chybí přenos z bitu, protože je první.

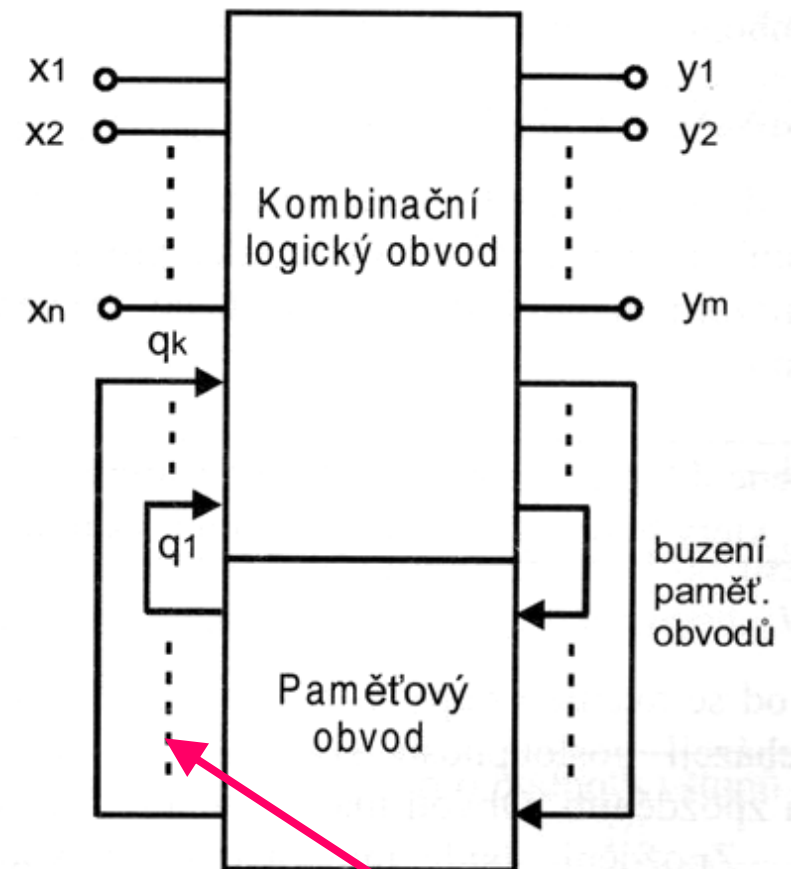
Sekvenční logické obvody

výstup závisí na

- vstupních proměnných
- vnitřním stavu (nastaveném v předchozím ději)

Skládá se z:

- kombinační části
 - paměti
 - zpětné vazby
- Abychom mohli určit hodnotu výstupní proměnné, je potřeba u sekvenčních obvodů sledovat kromě vstupních proměnných ještě i jeho vnitřní proměnné – **vnitřní stav**. Jsou to proměnné, které jsou uchovány v paměťových členech.
- Existence vnitřních proměnných způsobuje, že **stejně hodnoty vstupních** proměnných přivedené na vstup obvodu, **nevyvolávají vždy stejnou odezvu na výstupu** obvodu.



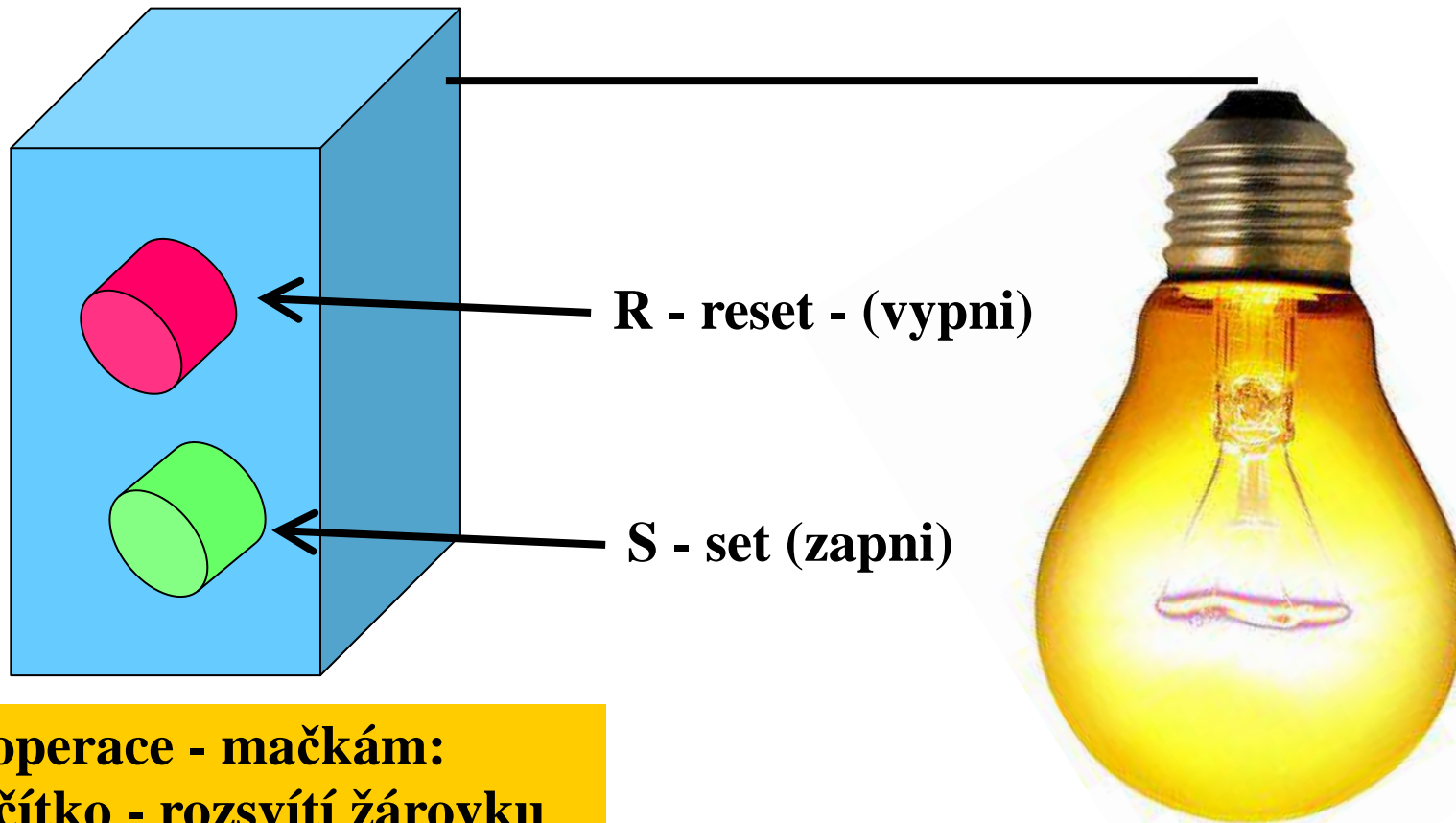
R – S
klopný obvod

Sekvenční logické obvody

vstupy

S - set (nastavuje stav výstupu do logické 1)

R - reset (nastavuje stav výstupu do logické 0 - nuluje výstup)



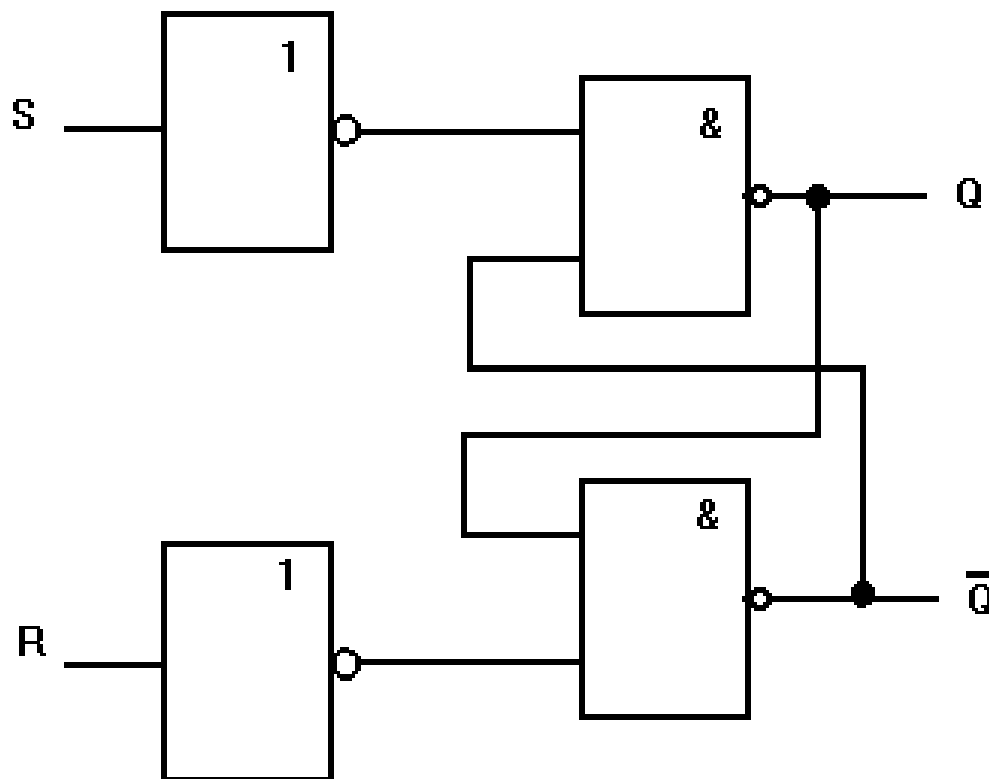
Povolené operace - mačkám:
Zelené tlačítko - rozsvítí žárovku
Červené tlačítko - zhasne žárovku

Zakázané operace - mačkám:
Obě tlačítka

R – S klopný obvod

Sekvenční logické obvody

Realizace funkce RS



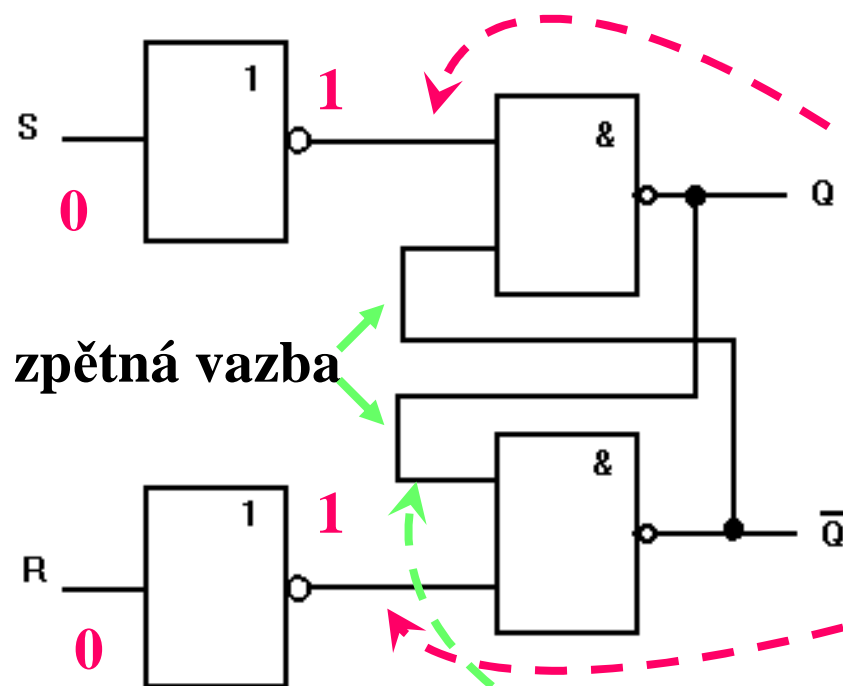
Pravdivostní tabulka RS klopného obvodu

S	R	Q^{t+1}	$\overline{Q^{t+1}}$
0	0	Q^t	$\overline{Q^t}$
0	1	0	1
1	0	1	0
1	1	(1)	(1)

Zakázaný stav

R – S klopný obvod

Sekvenční logické obvody



Pravdivostní
tabulka funkce
NAND

a	b	Y
0	0	1
0	1	1
1	0	1
1	1	0

Zde jsou „0“

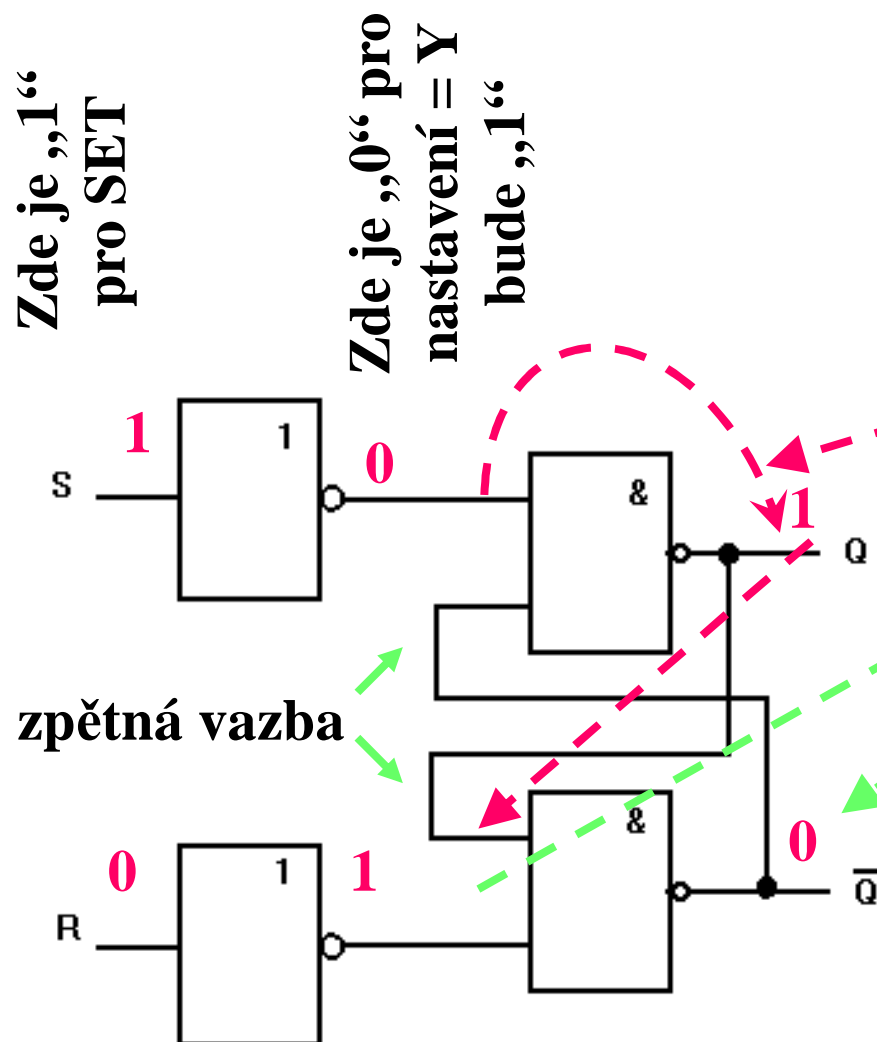
Zde jsou „1“

Zpětná vazba
má rozhodující
vliv na výstup.

R i S vstup jsou „0“ - výstup se nemění

R – S klopný obvod

Sekvenční logické obvody



Pravdivostní
tabulka funkce
NAND

a	b	Y
0	0	1
0	1	1
1	0	1
1	1	0

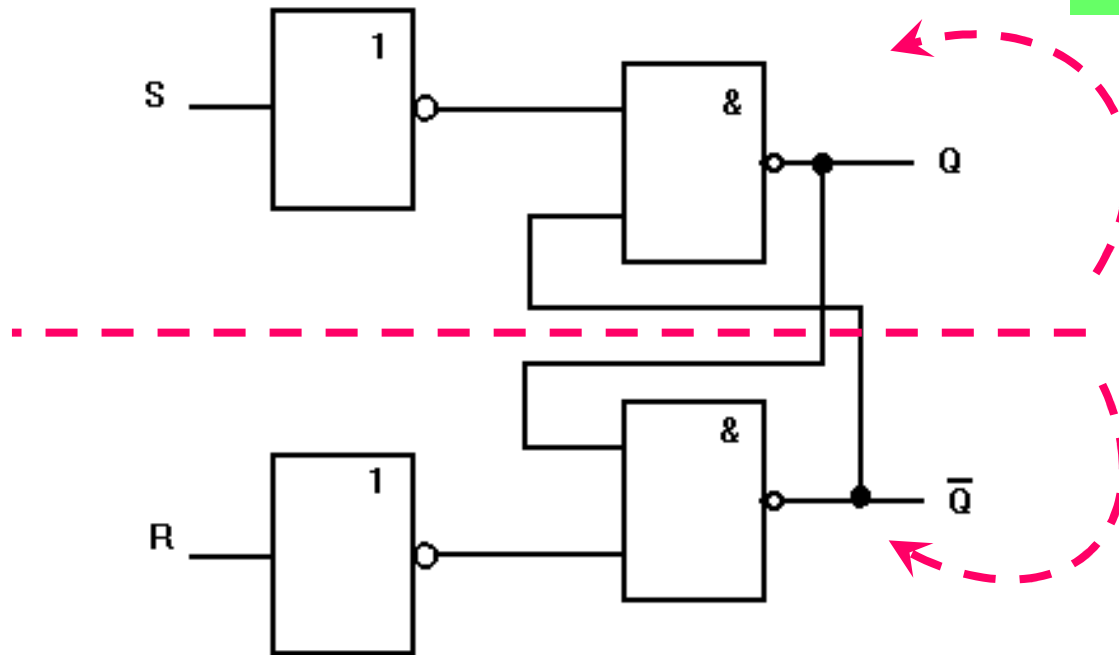
R vstup je „1“ - výstup Q = „1“

R – S klopný obvod

Sekvenční logické obvody

Stejný princip platí pro R jako S.
(Zrcadlové zapojení.)

S vstup je „1“ - výstup Q = „0“



Zapojení je
zrcadlové
pro SET i
RESET

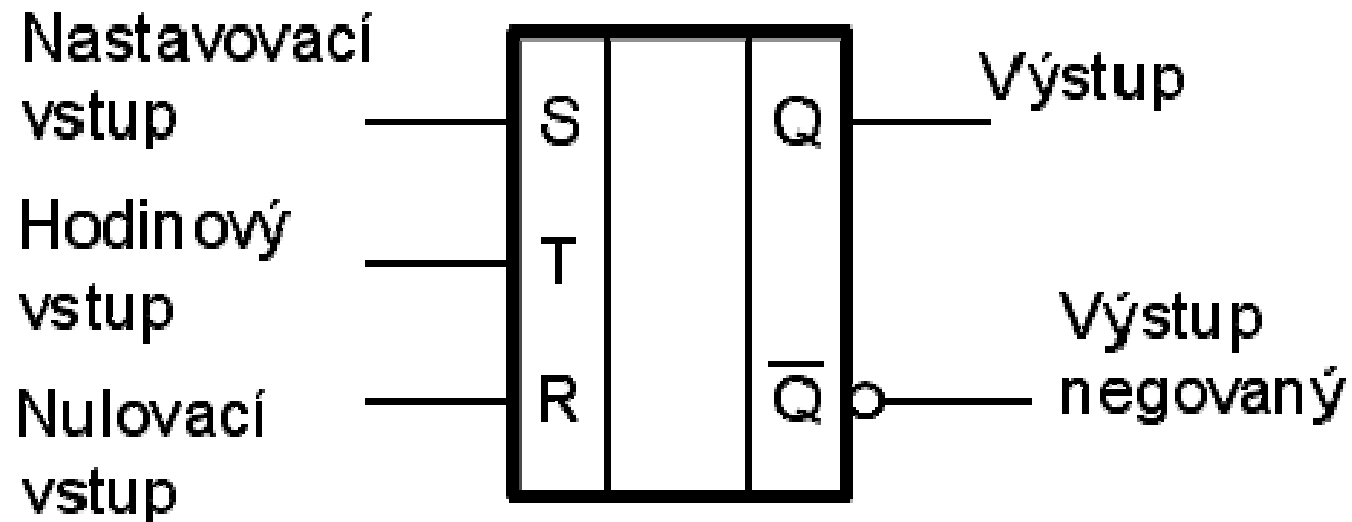
R i S v „0“ - nic se neděje
R „0“, S „1“ - nastavení do „1“
R „1“, S „0“ - nastavení do „0“

RST
klopný obvod

Sekvenční logické obvody

RS obvod doplněn o vstup T (časové pulzy)

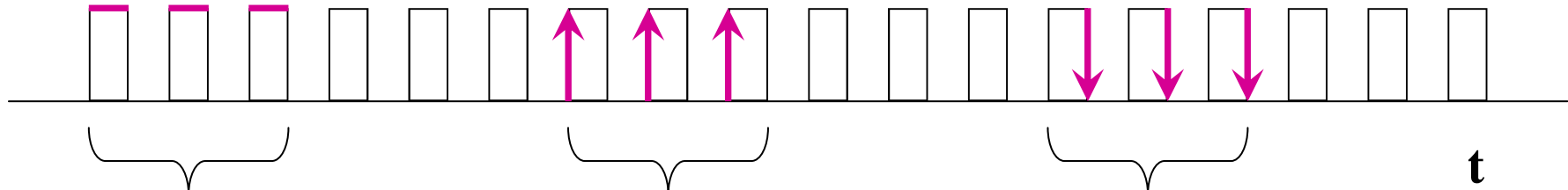
Pracuje stejně jako RS, jen v době, kdy
to dovolí vstup T.



Používá se tehdy, kdy musí být vstupy RS necitlivé a změna jen tehdy, kdy potřebujeme.

Sekvenční logické obvody

Obvody řízené časovými pulzy



**Obvody řízené úrovní
časového pulzu**

**Nevýhoda: dlouho trvá
citlivý stav vstupů**

**Obvody řízené ná-
běžnou hranou ča-
sového pulzu**

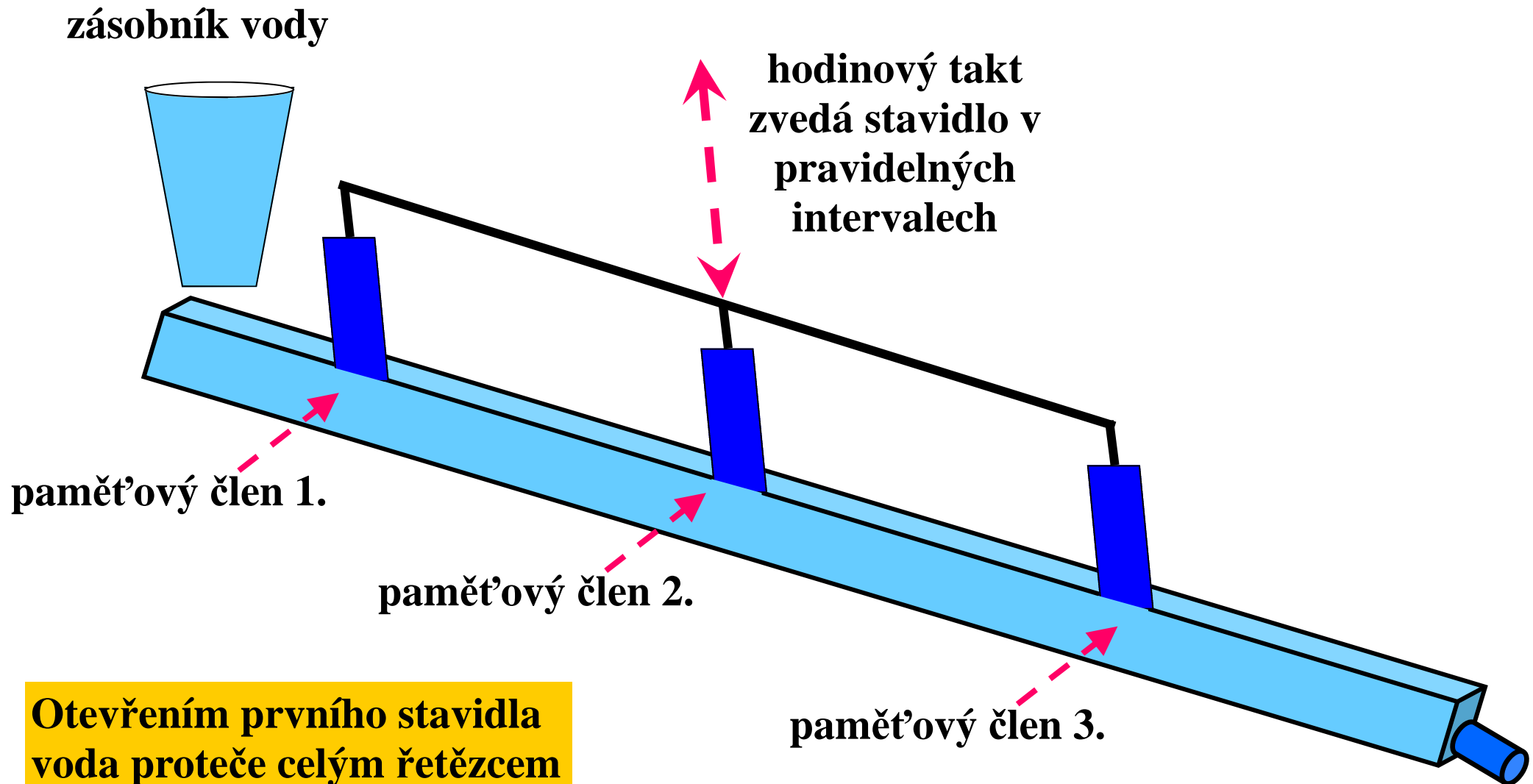
**citlivý stav vstupů
jen krátkou dobu
náběhu hodinového
puzu**

**Obvody řízené se-
stupnou hranou ča-
sového pulzu**

**citlivý stav vstupů
jen krátkou dobu
náběhu hodinového
puzu**

**Obvody řízené jak sestupnou tak ná-
běžnou hranou časového pulzu**

Pracují dvojnásobnou rychlostí.



**Otevřením prvního stavidla
voda proteče celým řetězcem
až na výstup - chybné řešení.**

zásobník vody

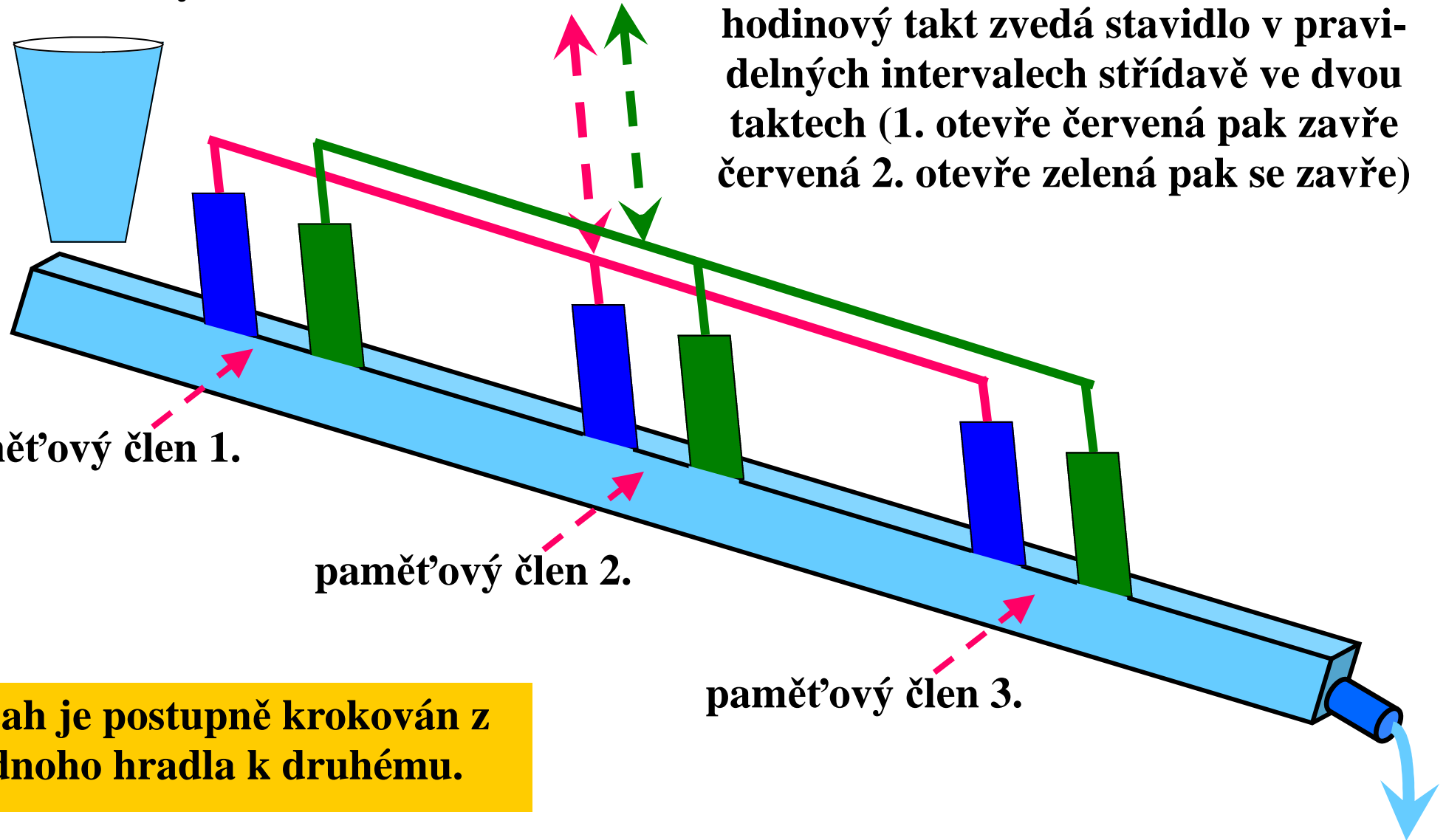
hodinový takt zvedá stavidlo v pravidelných intervalech střídavě ve dvou taktech (1. otevře červená pak zavře červená 2. otevře zelená pak se zavře)

paměťový člen 1.

paměťový člen 2.

paměťový člen 3.

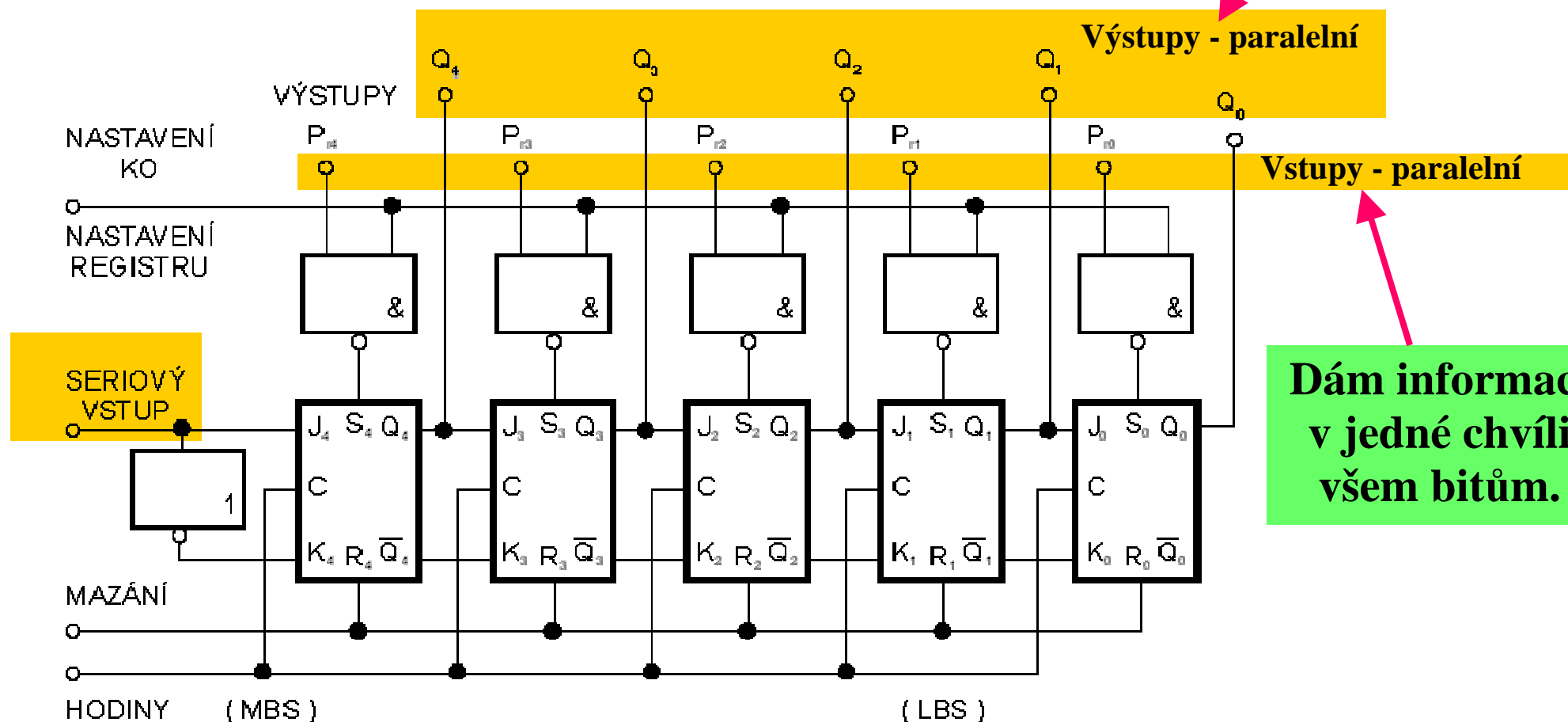
Obsah je postupně krokován z jednoho hradla k druhému.



Sekvenční logické obvody

Kombinací n klopných obvodů, schopnou **zapamatovat** si n-bitovou informaci, nazýváme registrem.

Dám informaci postupně bit po bitu.



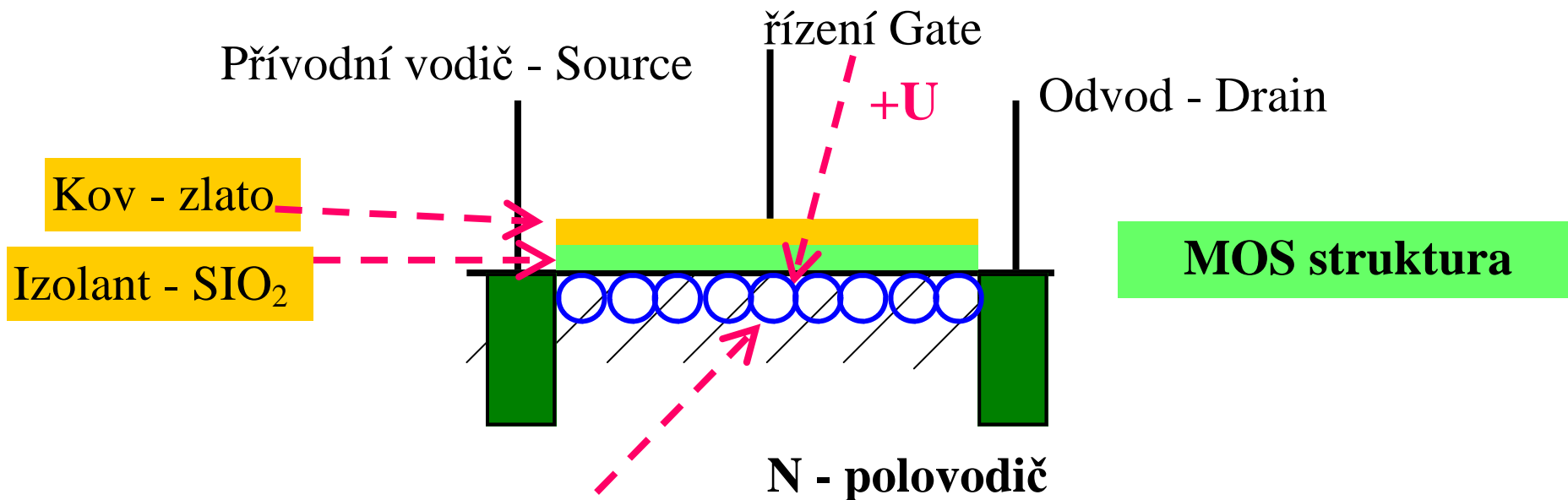
Dám informaci v jedné chvíli všem bitům.

Sériový vstup dat

- chceme zapsat do registru binární číslo 01011

číslo hod. impulsu	bit	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀
1	1 →	1	0	0	0	0
2	1 →	1	1	0	0	0
3	0 →	0	1	1	0	0
4	1 →	1	0	1	1	0
5	0 →	0	1	0	1	1

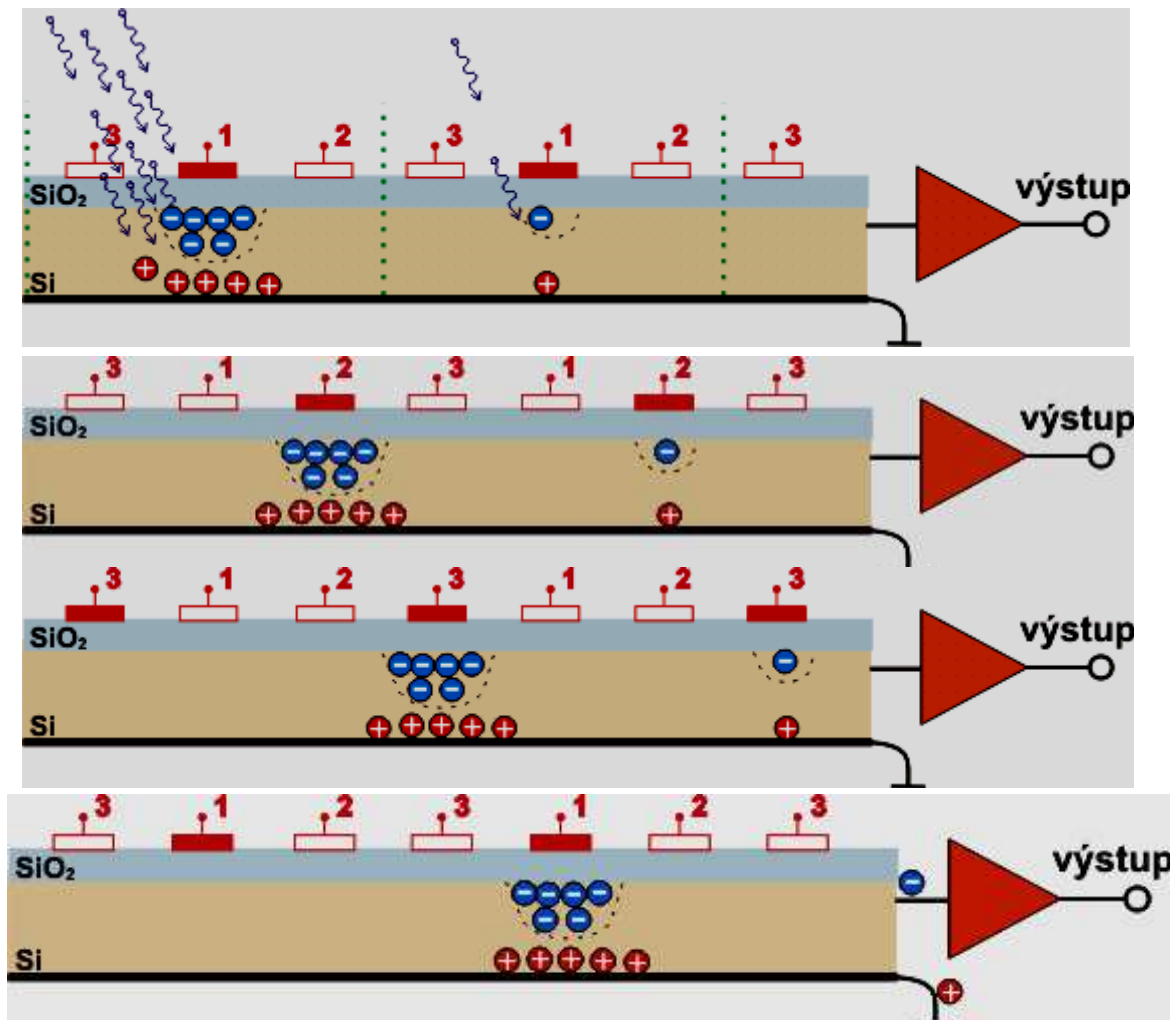
- Liší se od předchozích (statických) posuvných registrů svou dynamickou strukturou.
- Dynamické posuvné registry mají informace uloženy v kapacitě hradla (paměťová buňka) tranzistoru MOS.
- Náboj se však v malých kapacitách neudrží dlouho a proto musí být
- obnovován současně s posouváním dat



+U - (přitáhne) vytvoří kanál elektronů
ty povedou proud mezi S a D elektrodou

- posuv logické informace ve struktuře MOS lze uskutečnit pomocí efektu indukce potenciálové jámy
- nábojově vázané prvky vznikly jako počítačové paměti, ale jejich schopnost převádět světlo na elektrický signál z nich vytvořila kvalitní detektory světla
- na povrchu polovodiče je vytvořena řada kovových elektrod vzájemně izolovaných (oxid křemíku) od polovodiče i mezi sebou
- uspořádány lineárně nebo do matice
- pokud se na elektrody přivede různé napětí, elektrony mohou být „přelévány“ z jedné nábojové studny do sousední
- tak je možné náboj posouvat po ploše čipu
- tento proces je používán, když je třeba informaci z CCD čipu vyčíst.
- balíky elektronů, reprezentující jednotlivé pixely, jsou posouvány do výstupního zesilovače, kde je elektrický náboj převeden na napětí
- toto napětí se objeví na výstupním pinu CCD čipu

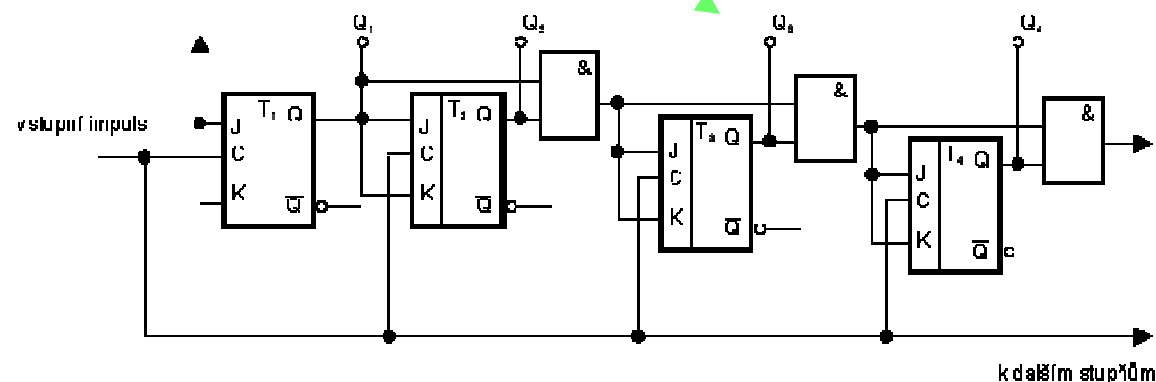
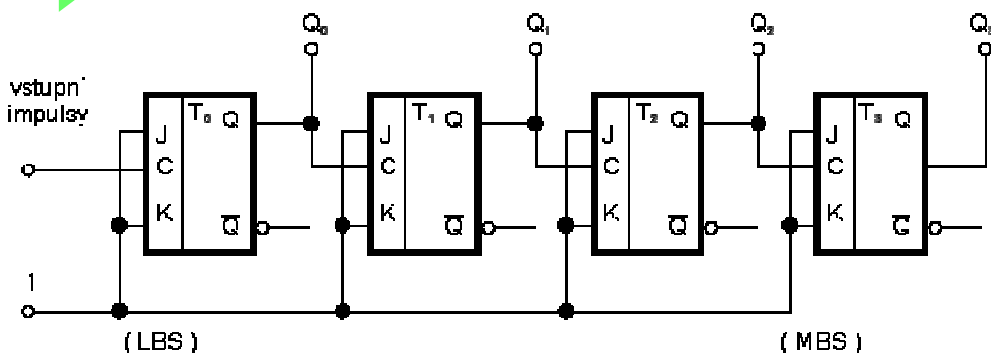
Posouvání náboje pod buňkami CCD struktury:



Sériový vstup klopného obvodu (odpovídá bitu s nejnižší váhou) umožňuje čtyři různé možnosti ovládání vkládání informací:

- **Náhrada nulami** - bity se v registru při posuvu uvolňují od nejnižší váhy. Spojíme-li sériový vstup s nulovou logickou úrovní, pak jsou na uvolněná místa vkládány nuly.
- **Náhrada jedničkami** - sériový vstup je spojen s logickou úrovní odpovídající jedničce.
- **Sériový zápis informací** - sériový vstup je připojen k externímu zdroji dat. Posuvný registr a externí datový zdroj jsou pomocí hodinových impulsů synchronizovány.
- **Kruhový posuv** - tzv. kruhové registry = paměť s kolujícími daty.
 - ⇒ posun vpravo → výstup bitu s nejvyšší váhou je spojen se vstupem bitu s nejnižší váhou
 - ⇒ posun vlevo → výstup bitu s nejnižší váhou je spojen se vstupem bitu s nejvyšší váhou

- ⇒ čítá počet vstupních impulsů a vyjadřuje jejich počet pomocí buď binárního nebo jiného kódu
- ⇒ **asynchronní čítač** - změna stavu z 1 do 0 předcházejícího obvodu teprve působí změnu stavu následujícího obvodu (může být pomalý proces proti rychlosti vstupních pulzů)
- ⇒ **synchronní čítač** - ze stavu **kombinace** předcházejících výstupů obvodů se určuje logická úroveň vstupu
- ⇒ **vratný čítač** - čítání buď vpřed nebo vzad

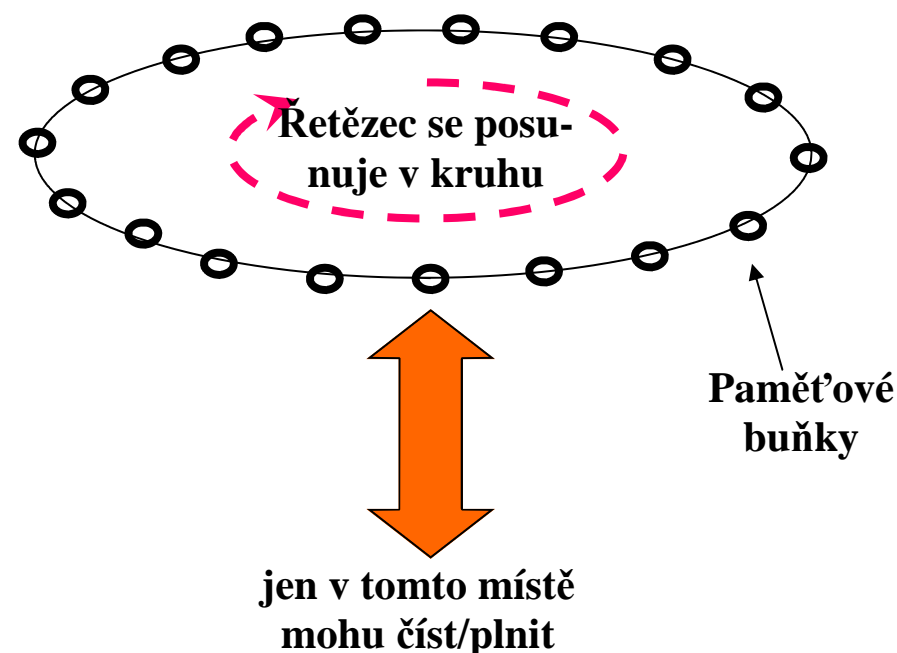
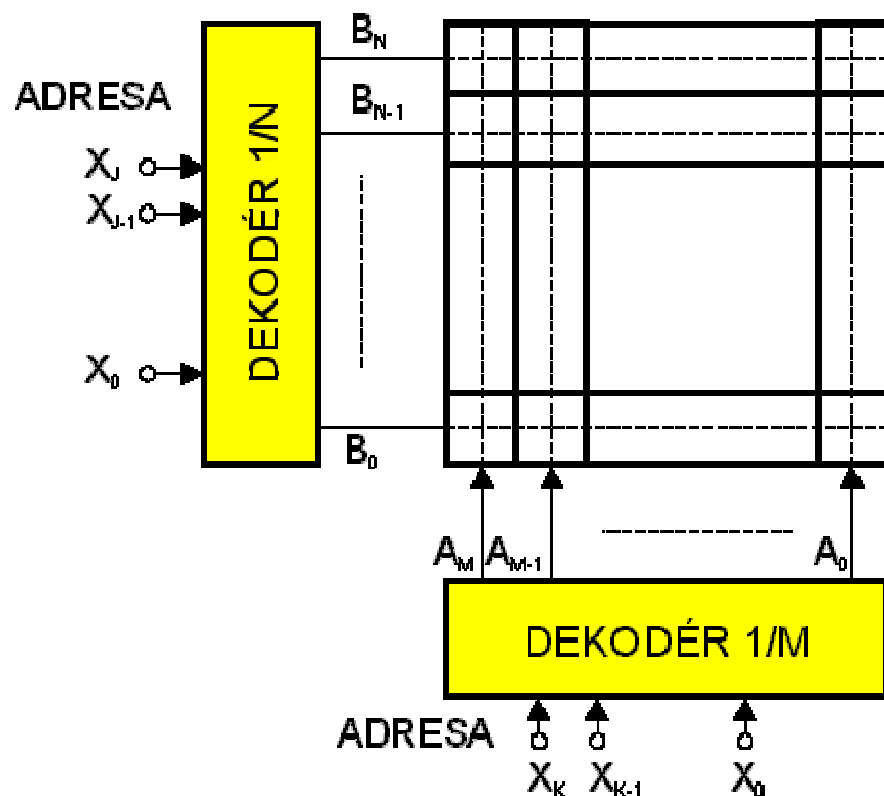


Polovodičové paměti

Podle typu přístupu mohou být paměti rozděleny na:

⇒ **RAM** (Random Access Memory) - paměti s libovolným přístupem

⇒ **SAM** (Serial Access Memory) - paměti se sériovým přístupem.

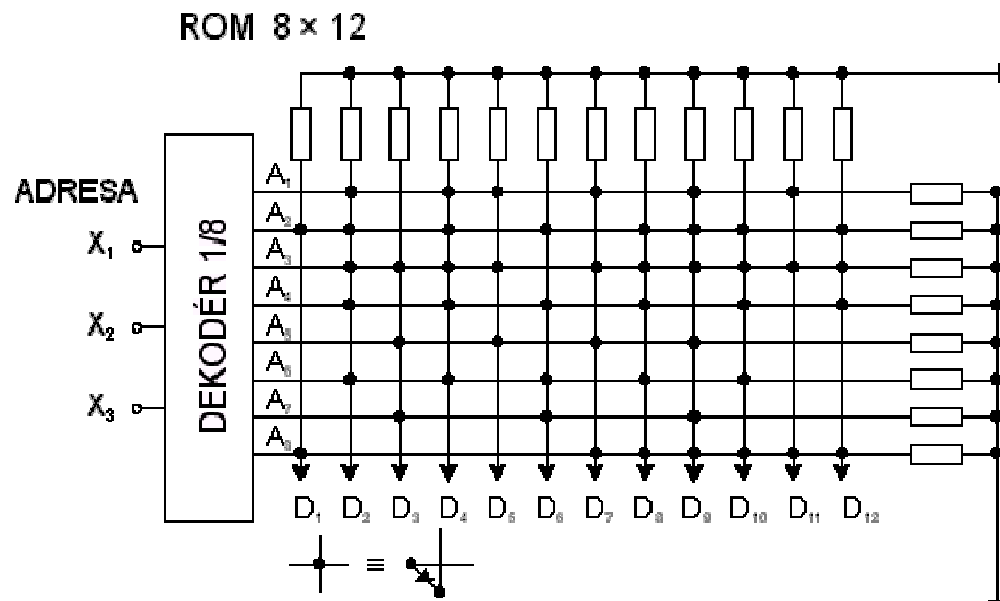


Polovodičové paměti

Podle možnosti zápisu/čtení mohou být paměti rozděleny na:

⇒ **ROM** (Read Only Memory) - paměti pouze pro čtení

⇒ **RWM** (Read Write Memory) - paměti pro zápis i čtení



⇒ **statické** hodně součástek - malé kapacity

⇒ **dynamické** jedna součástka = jeden bit - velké kapacity

Polovodičové paměti

⇒ **RWM** (Read Write Memory) - paměti pro zápis i čtení

⇒ **Statické SRAM** hodně součástek - malé kapacity

⇒ **Dynamické DRAM** jedna součástka
= jeden bit - velké kapacity
Informace = náboj kapacity tzn. musím obnovovat = **refresh**

