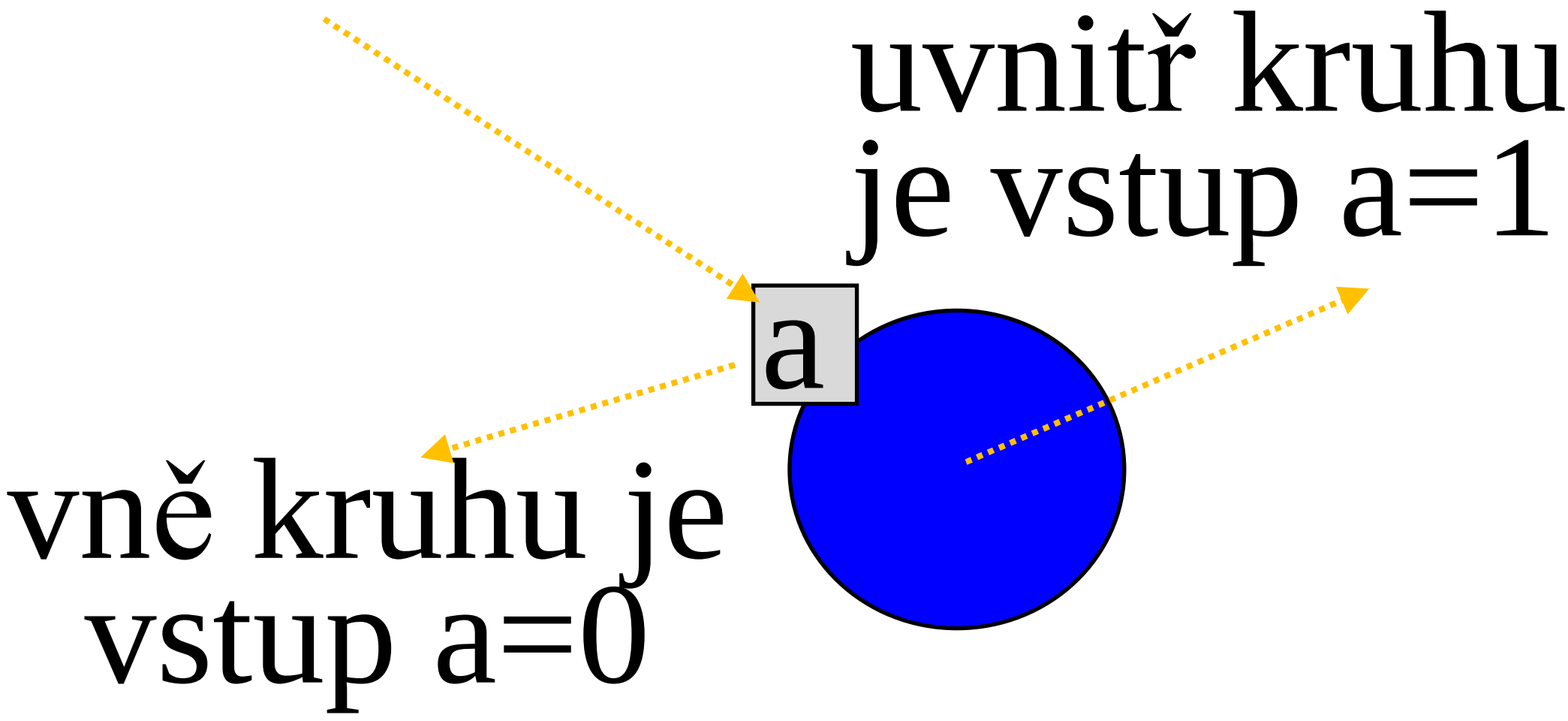
Logická funkce

- **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

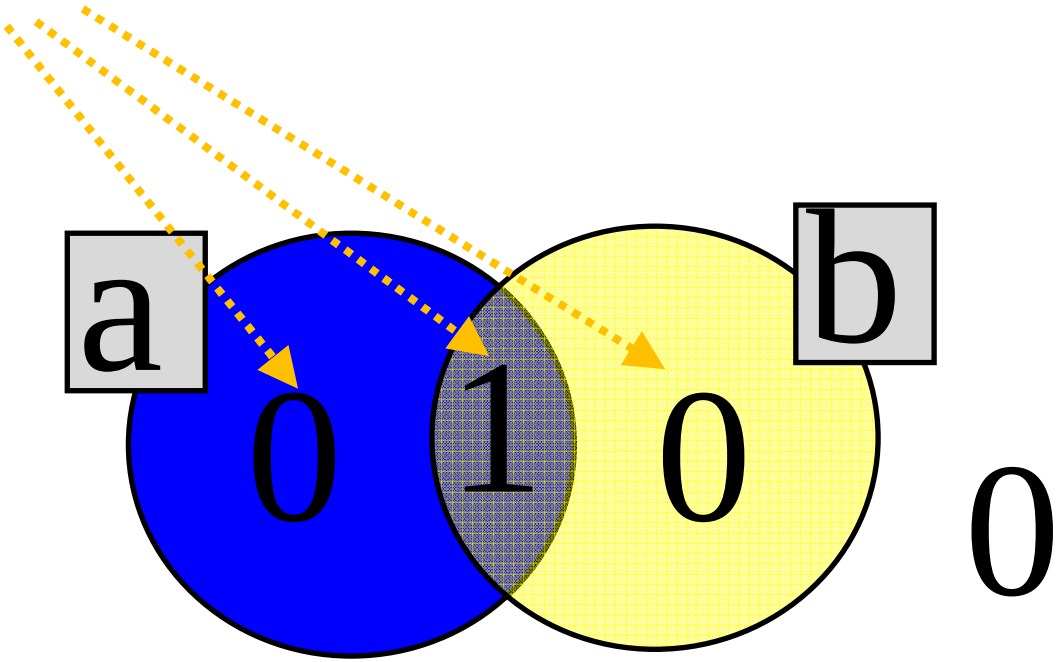
Vennův diagram

Vstupní proměnnou zaznamenáme kruhem (příp.jiným obraz-cem), 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 ∧ AND, &)je binární logická operace jejíž hodnota je pravda, právě když obě vstupní hodnoty jsou pravda.

Zápis:
X = A AND B, nebo X = A & B , X = A ∧ B, X=A*B

Mohu vyjádřit pravdivostní tabulkou:

A	B	A ∧ 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 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

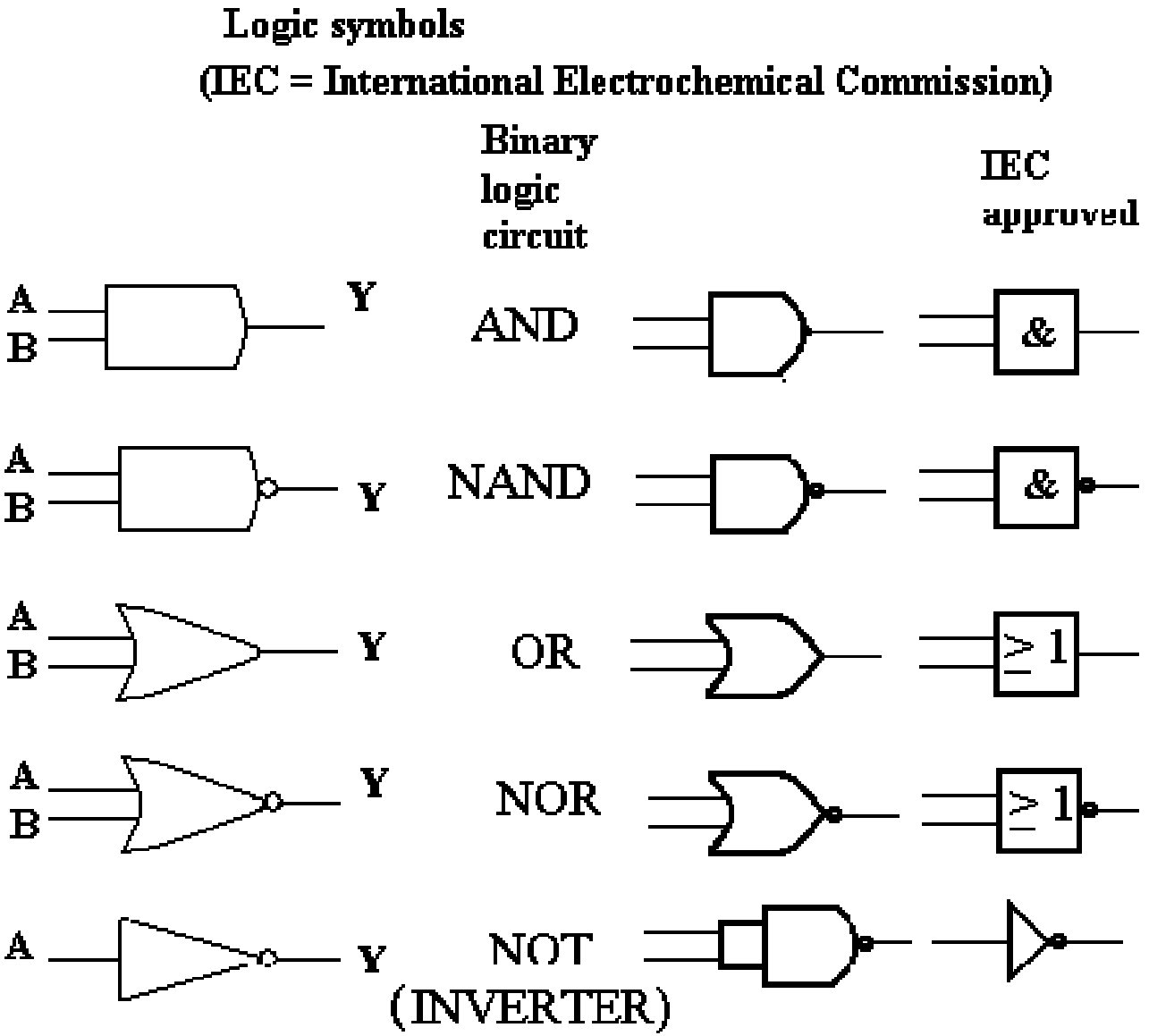
Logická negace (používá se pro ni NOT, popř. se označuje pruhem nad proměnnou) hodnota je nepravda, právě když první vstupní hodnota je pravda a naopak.

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:



Zákony Booleovy algebry

požadované znalosti:
- **celkový přehled o možnosti zápisu**
- **umět si odvodit proč a+1=1, a*0=0 … a podobně**
- **znát de Morganova pravidla**

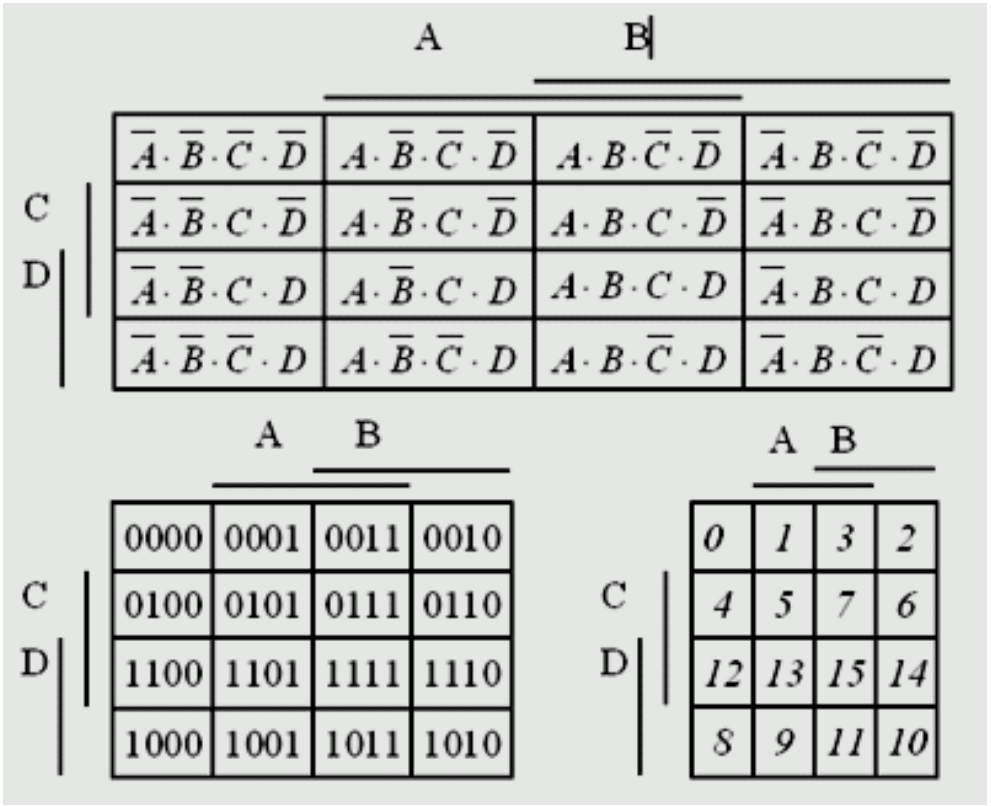
Booleova algebra
- minimalizace logických funkcí pomocí zákonů a pravidel

Zákon komutativní	$a + b = b + a$	$a \times b = b \times a$	
Zákon asociativní	$a + (b + c) = (b + a) + c$		
Zákon distributivní	$a \times (b + c) = (a \times b) + (a \times c)$	$a + (b \times c) = (a + b) \times (a + c)$	
Zákon idempotentní	$a + a = a$	$a + 0 = a$	$a + 1 = 1$
	$a \times a = a$	$a \times 0 = 0$	$a \times 1 = a$
Zákon doplňku	$a + \bar{a} = 1$	$a \times \bar{a} = 0$	
Zákon involuce	$a = \bar{\bar{a}}$		
Zákon absorpce	$a \times (a + b) = a$	$a + (a \times b) = a$	
Zákon absorpce negace	$a + (\bar{a} \times b) = a + b$	$\bar{a} \times (a + b) = \bar{a} \times b$	
Zákon de Morganův	$\overline{a + b} = \bar{a} \times \bar{b}$	$\overline{a \times b} = \bar{a} + \bar{b}$	

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.**



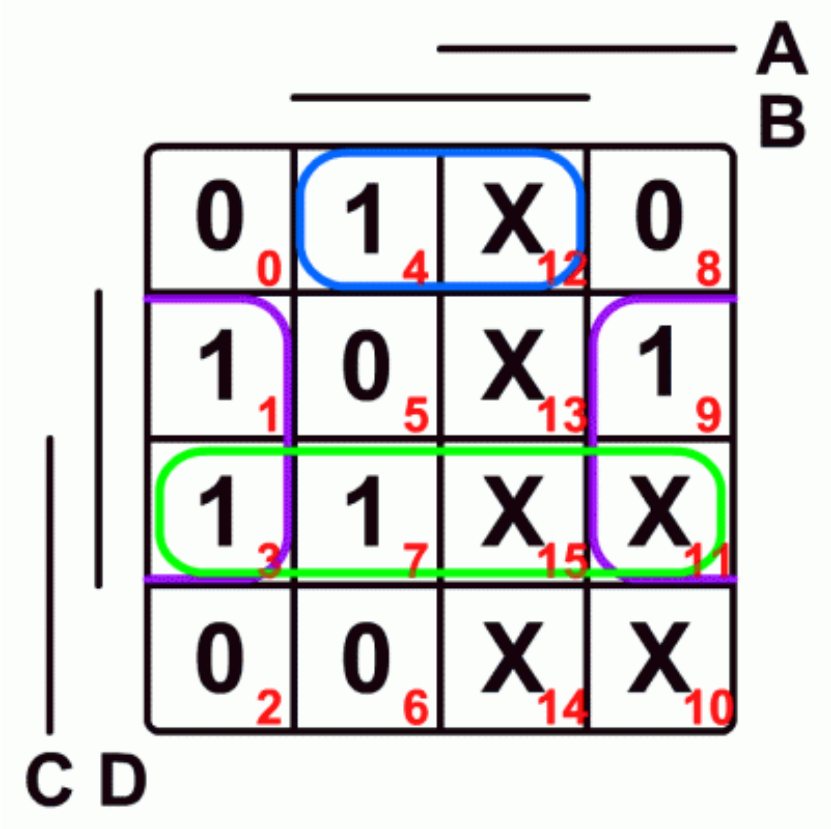
- **vytvářím podmapy ze sousedních políček**
- **podmapa představuje člen součtu**
- **každý člen součtu je násobek vstupních proměnných, které představuje políčko**
- **pokud podmapa představuje přímý i negovaný tvar vstupní proměnné, proměnná vypadne**
- **snažím se vytvořit co největší podmapy**
-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



Zjednodušení logické funkce

Pravdivostní tabulka (zadání):

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

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

Odpovídá:

		y			
		z			
x		1	0	0	1
		1	0	0	0

Řešení:

		y			
		z			
x		1	0	0	1
		1	0	0	0

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

+

		y			
		z			
x		1	0	0	1
		1	0	0	0

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

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**

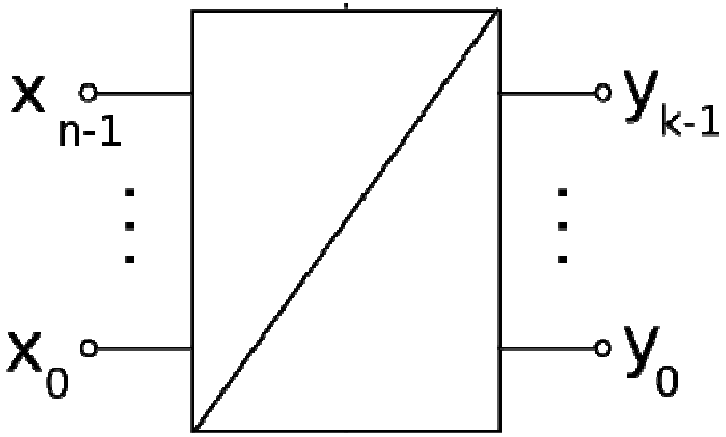
Pro realizaci kombinačních obvodů je možné použít:

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

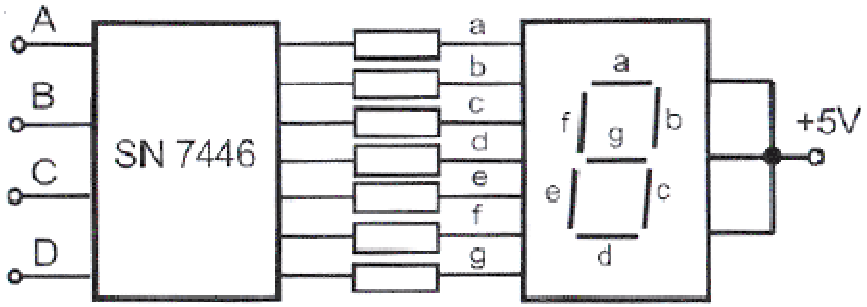
Kombinační logické obvody

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 x0 … xn-1 jejichž kombinace vytváří na k výstupech y0 … yk-1 jinou kombinaci signálů. Obecně platí, že n > k. Volitelně je možno použít strobovací výstup s pro vzorkování signálu nebo signál e pro uvolnění**



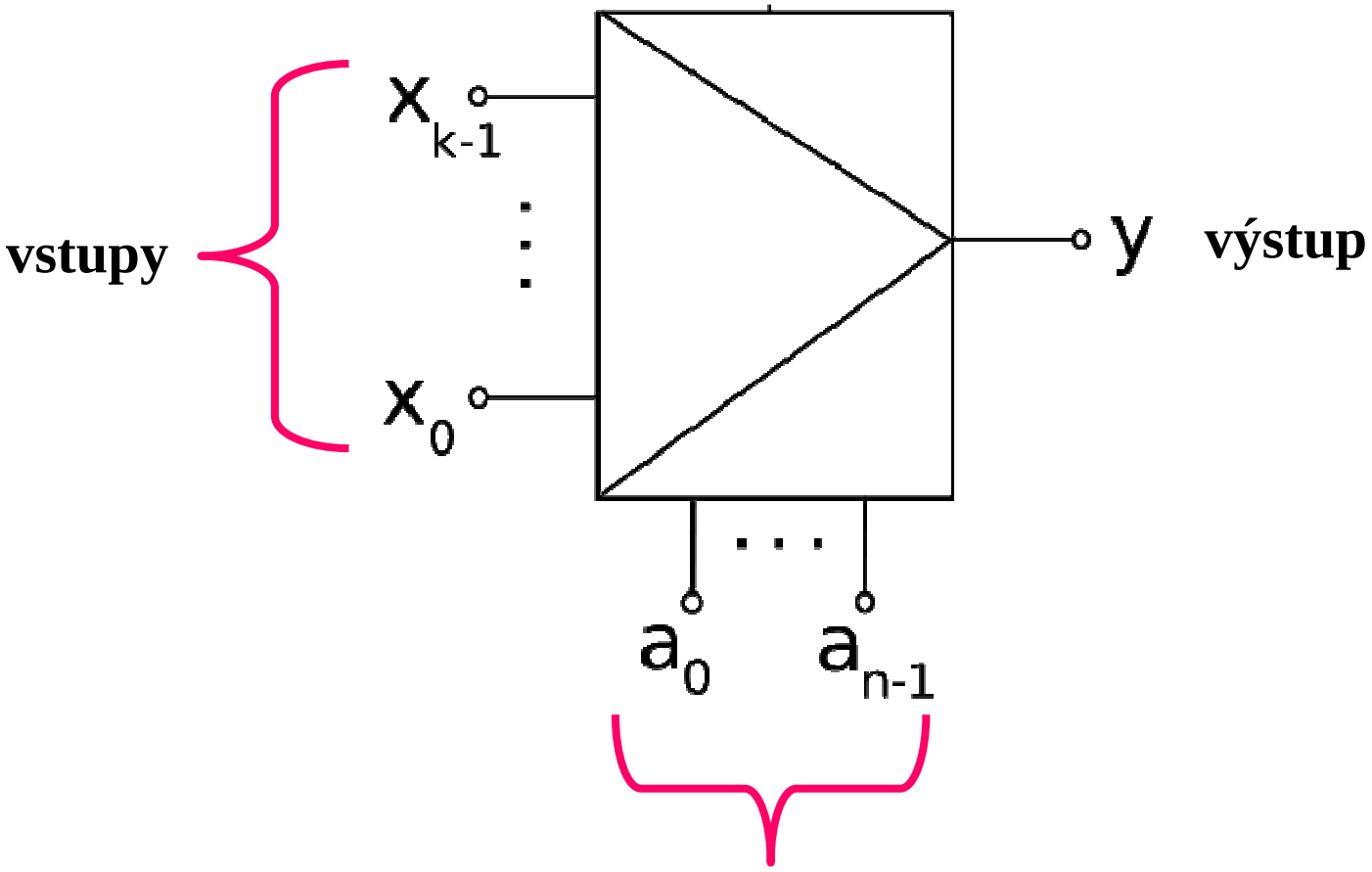
Příklad - kodér hexadecimální/ sedmisegmentovka:



Kombinační logické obvody

Multiplexer

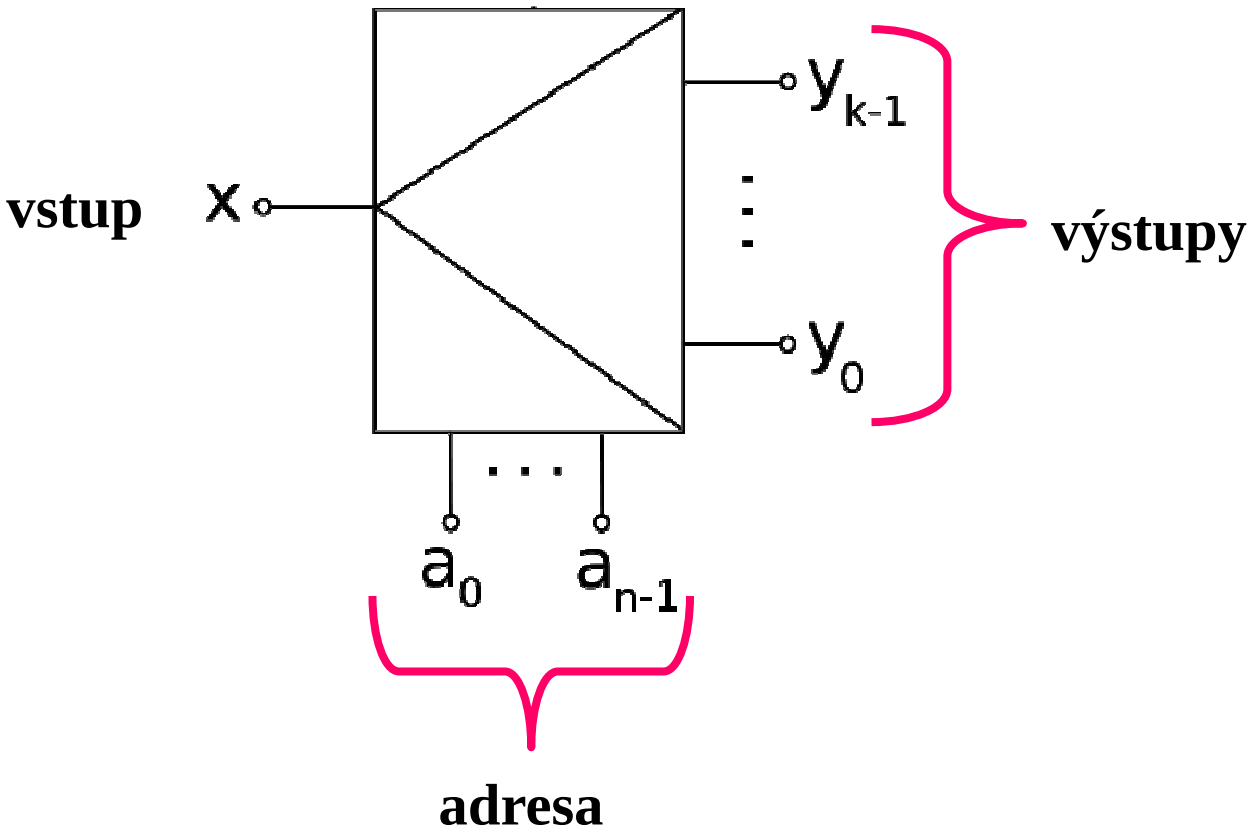
je elektronický člen, fungující na principu přepínače, kdy je podle řídících signálů (a) přiváděn na výstup (y) jeden ze vstupních signálů (x)



Kombinační logické obvody

Demultiplexer

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



Kombinační logické obvody

Sčítačka

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

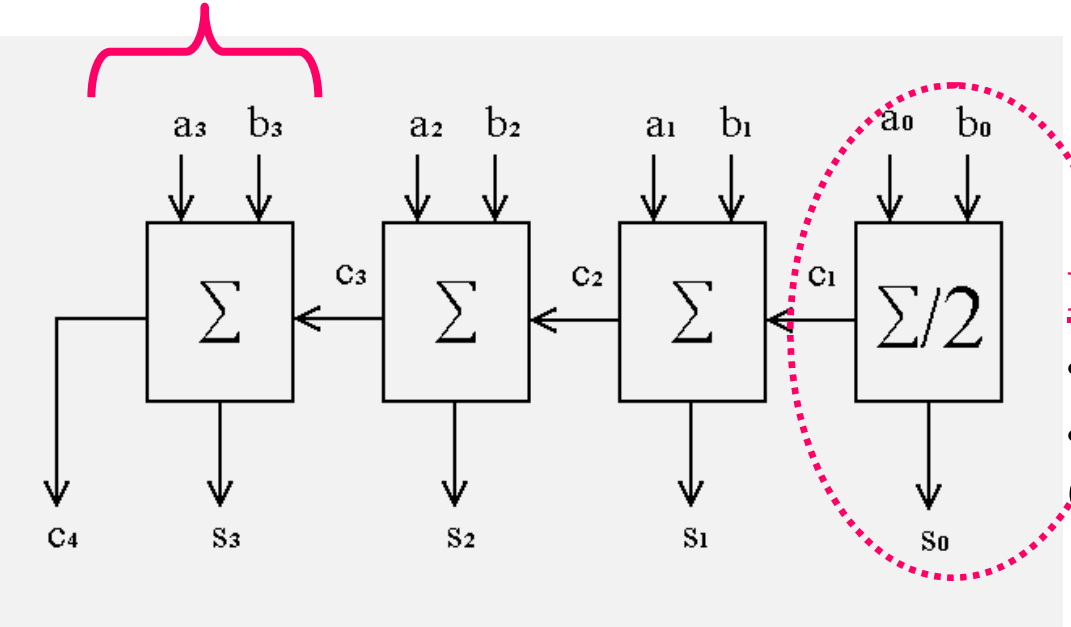
Ve dvojkové soustavě se provádí sčítání podle klíčů:

$0+0=0$
$0+1=1$
$1+0=1$
$1+1=10$

přenos do vyššího řádu

Úplná má

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



Poloviční má

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

Chybí přenos z bitu, protože je první.

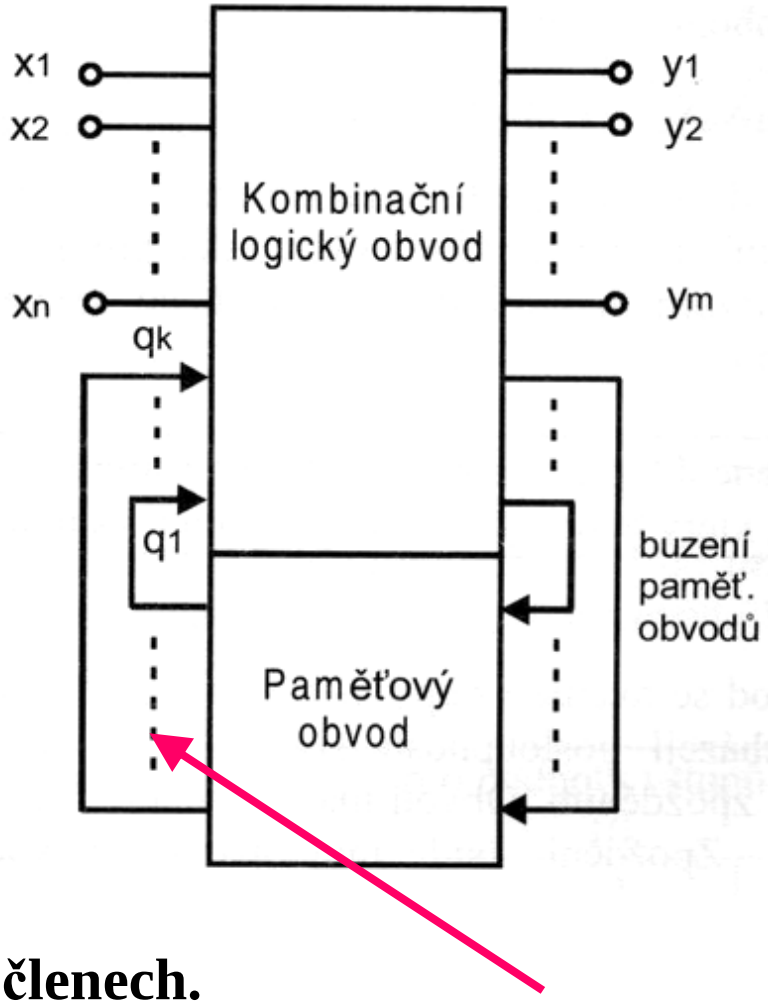
Sekvenční logické obvody

výstup závisí na

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

Skládá se z:

- **kombinační části**
- **paměti**
- **zpětné vazby**

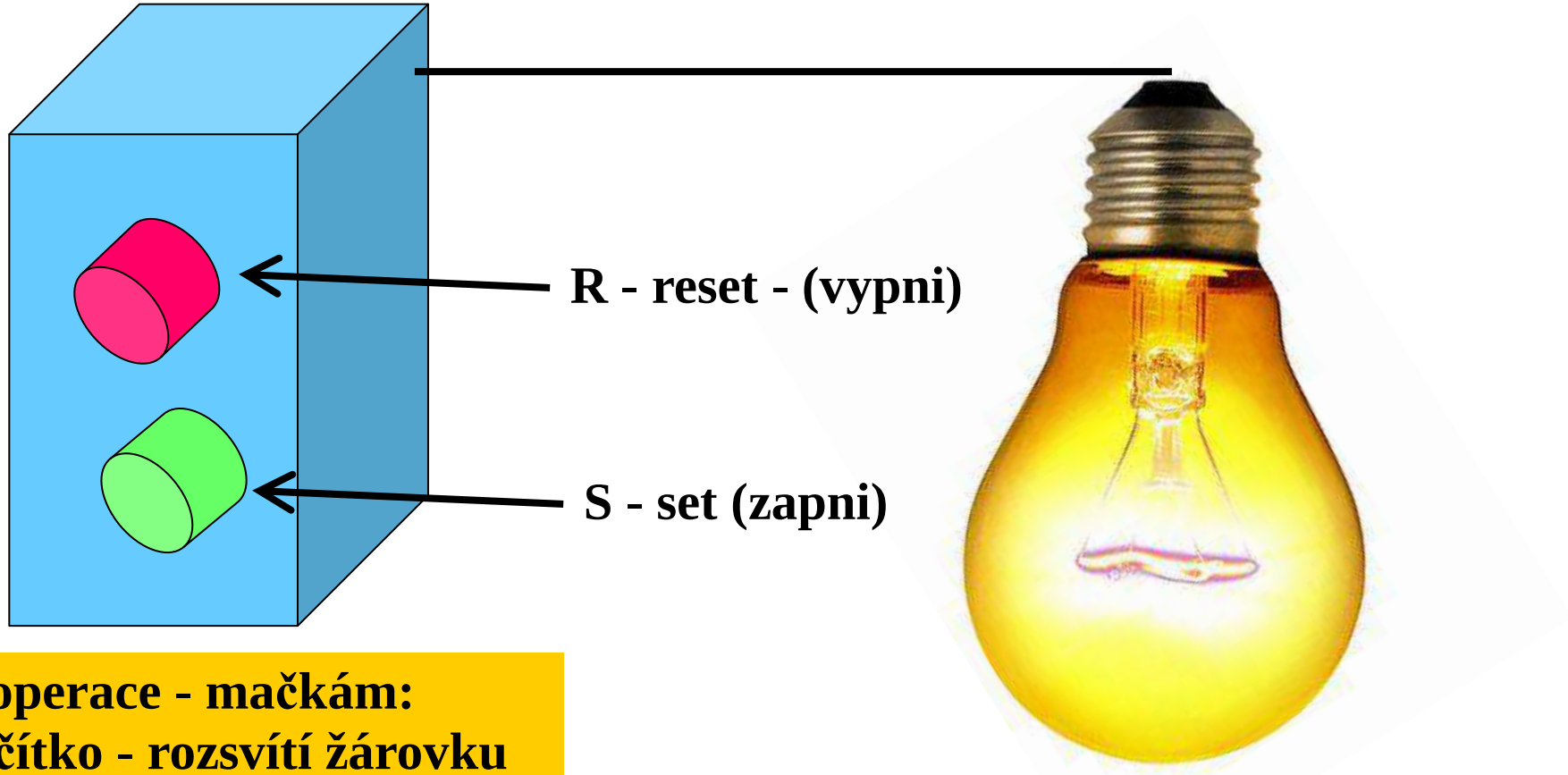


- **Abychom mohli určit hodnotu výstupní proměnné, je potřeba u sekvenčních obvodů sledovat kromě vstupních proměnných ještě i jeho vnitřní proměnné – vnitřní stav. Jsou to proměnné, které jsou uchovány v paměťových členech.**
- **Existence vnitřních proměnných způsobuje, že stejné hodnoty vstupních 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)



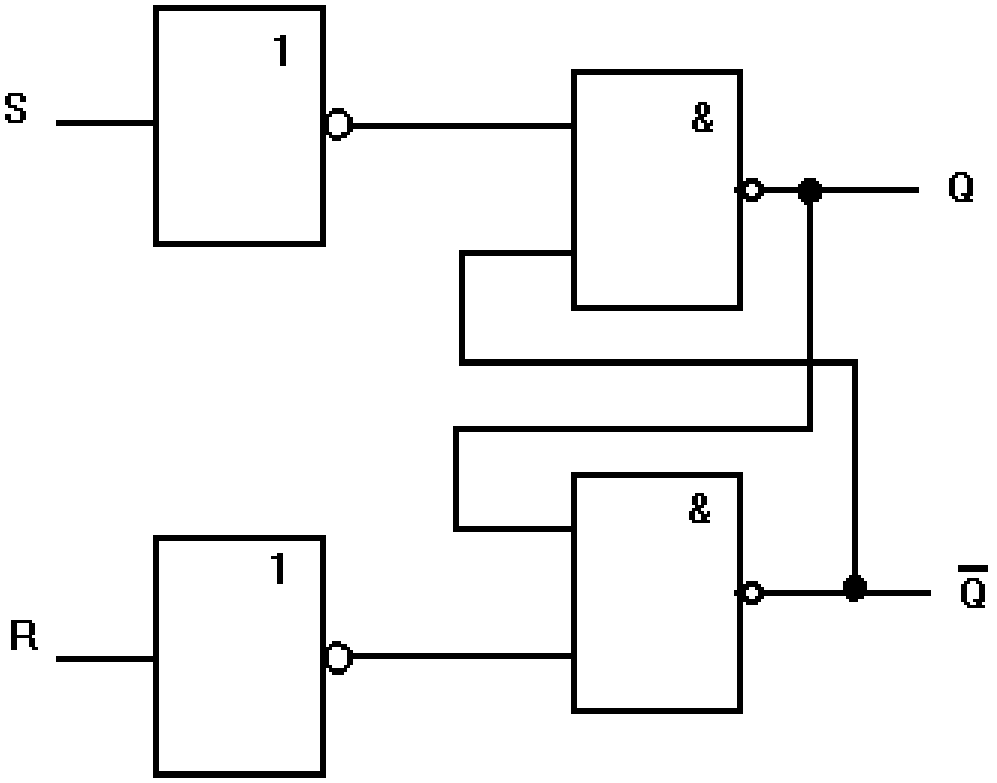
Povolené operace - mačkám:
Zelené tlačítko - rozsvítí žárovku
Červené tlačítko - zhasne žárovku

Zakázané operace - mačkám:
Obě tlačítka

R – S klopný obvod

Sekvenční logické obvody

Realizace funkce RS



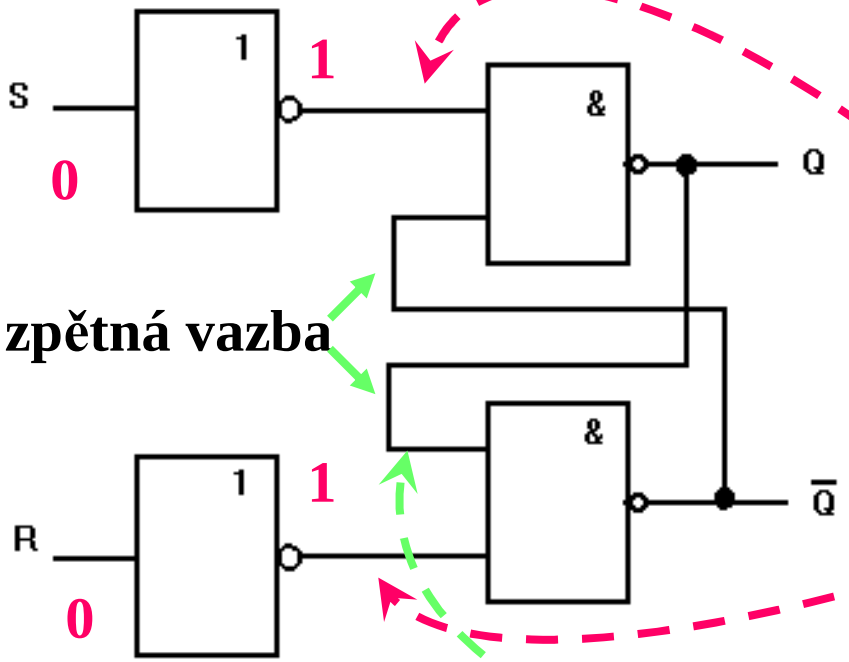
Pravdivostní tabulka RS klopného obvodu

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



zpětná vazba

Pravdivostní tabulka funkce NAND

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

Zde jsou „0“

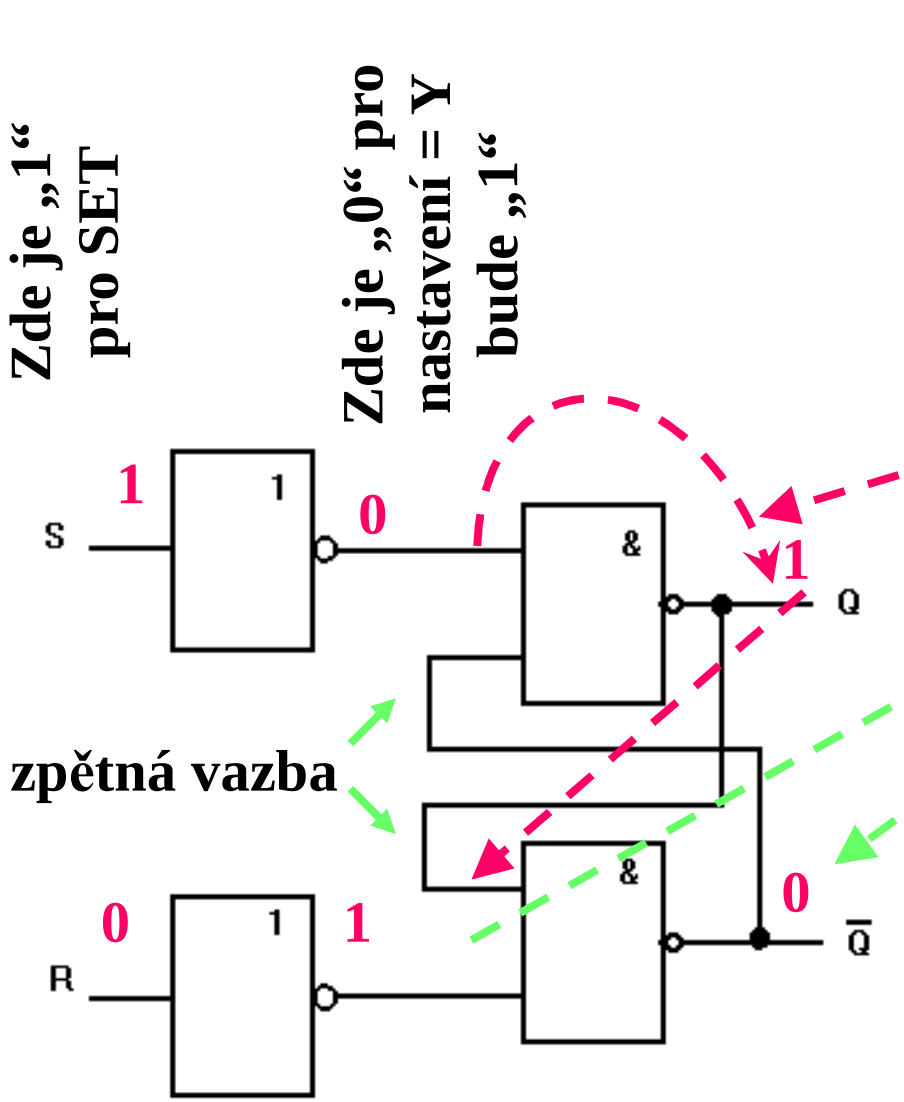
Zde jsou „1“

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

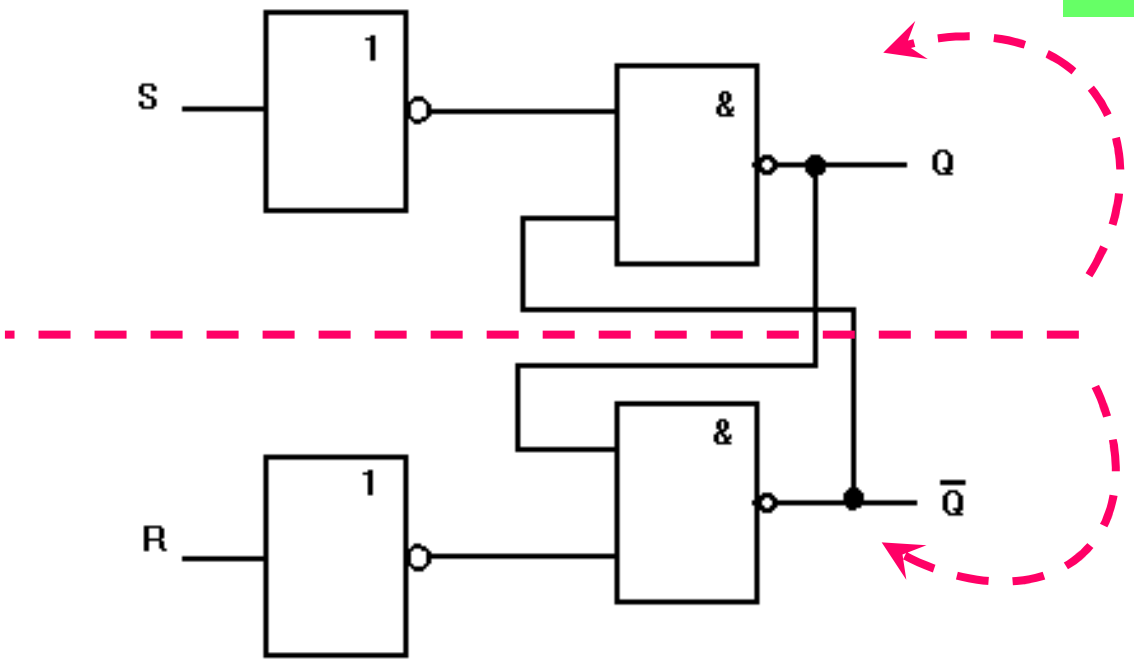
a	b	Y
0	0	1
0	1	1
1	0	1
1	1	0

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

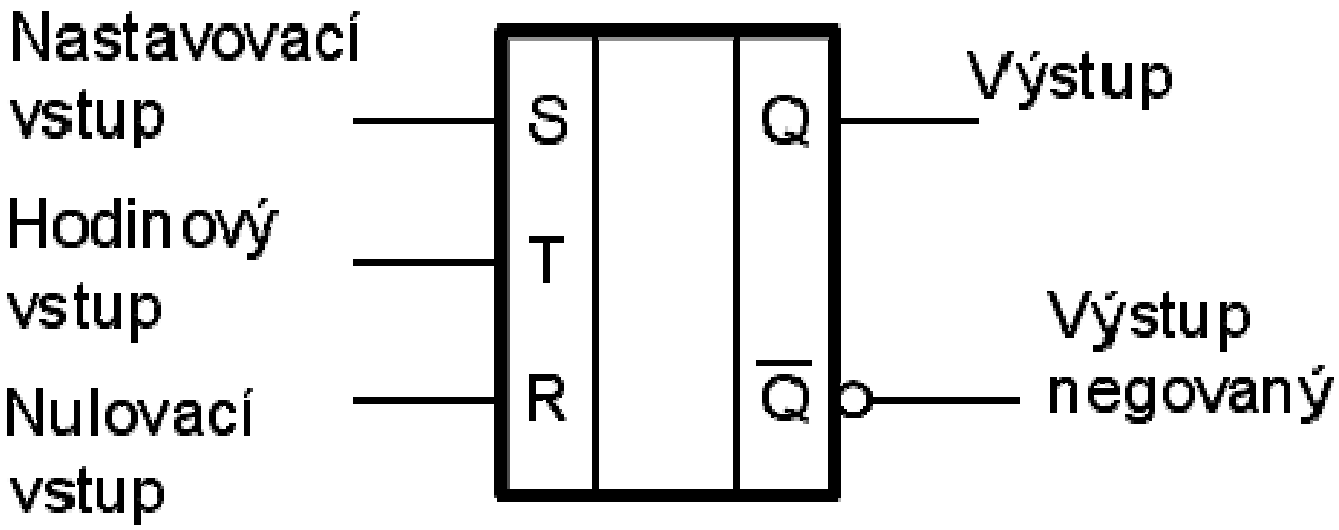
R i S v „0“ - nic se neděje
R „0“, S „1“ - nastavení do „1“
R „1“, S „0“ - nastavení do „0“

RST klopný obvod

Sekvenční logické obvody

RS obvod doplněn o vstup T (časové pulzy)

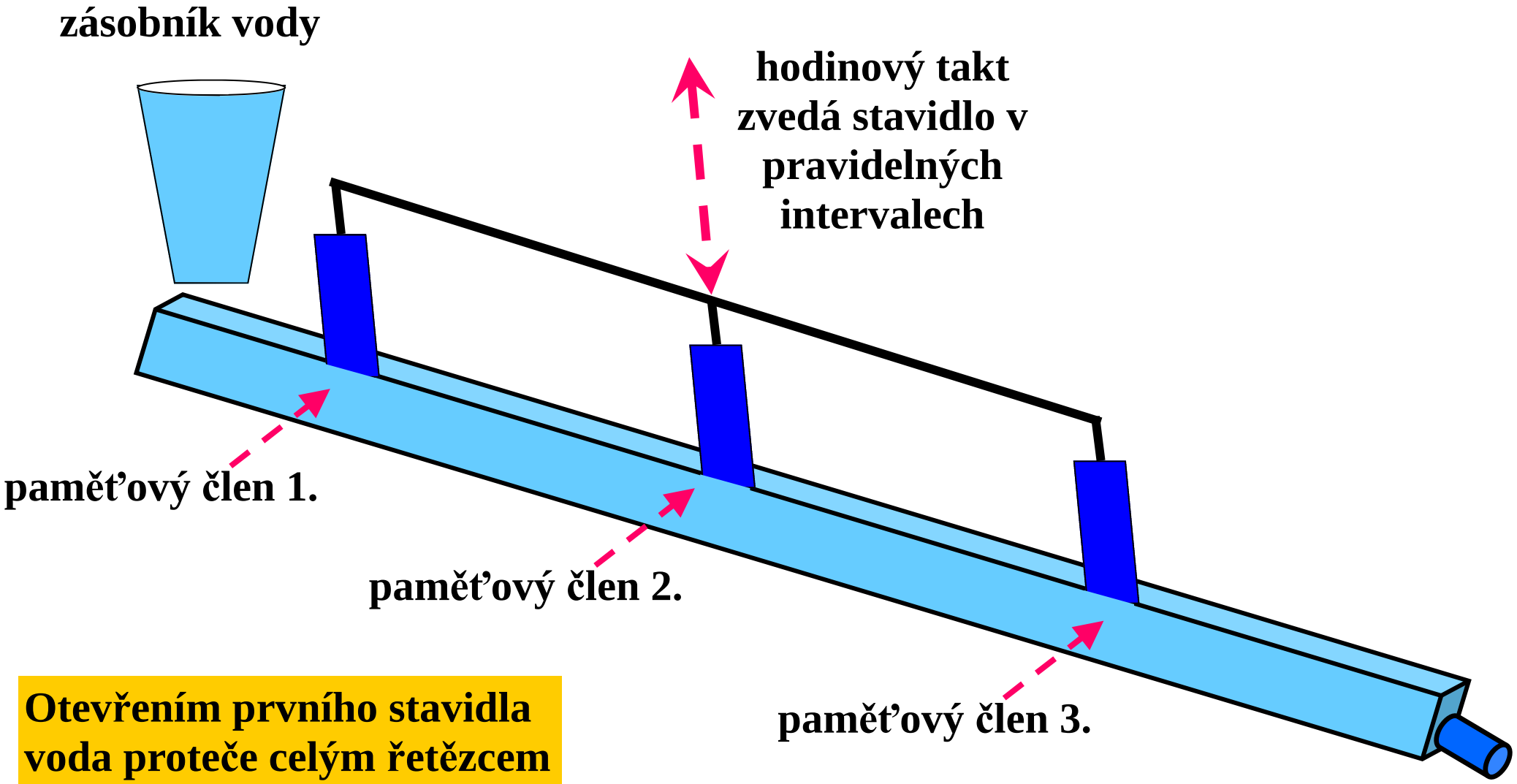
Pracuje stejně jako RS, jen v době, kdy to dovolí vstup T.



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

Dvojčinné klopné obvody

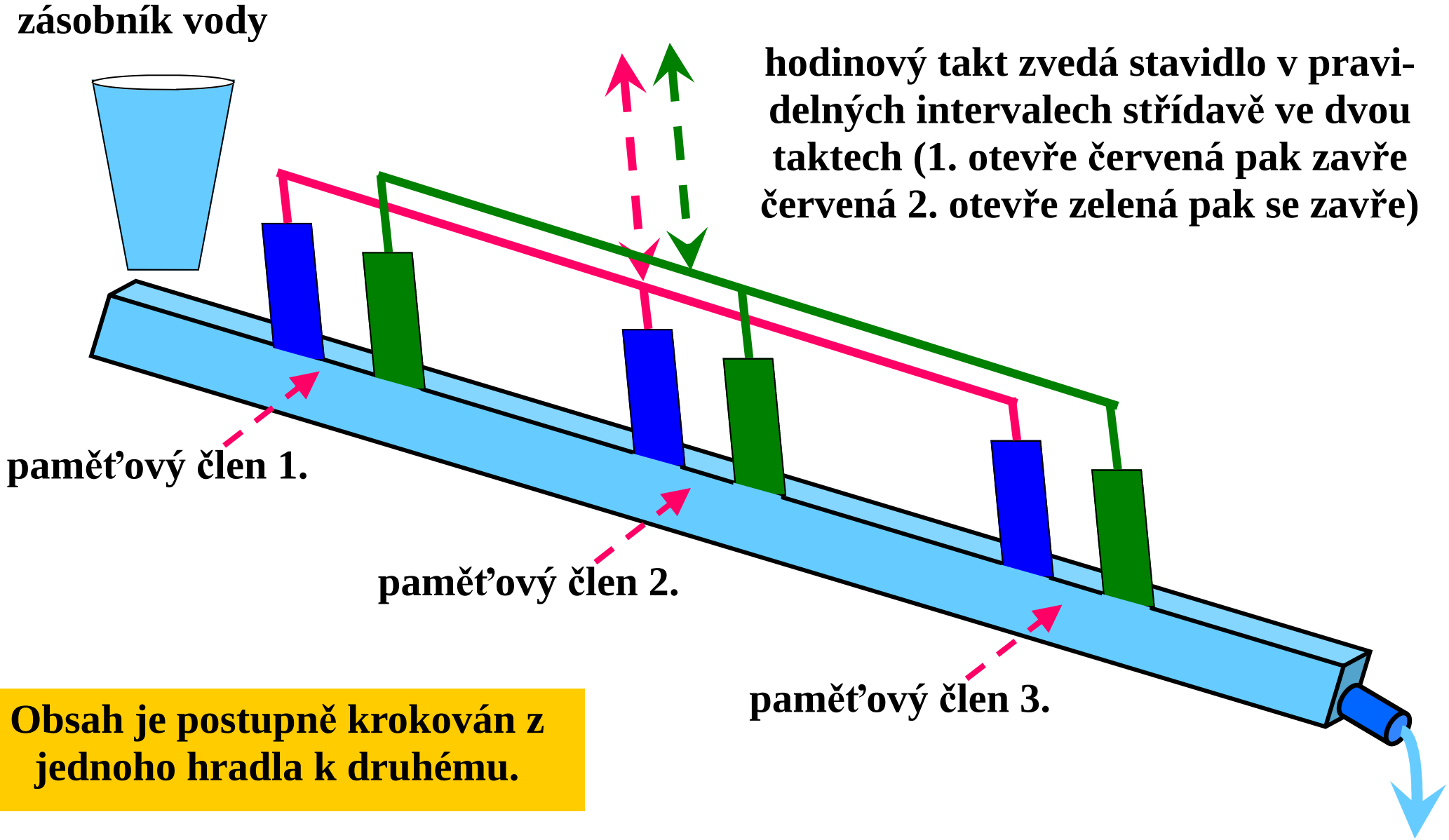
Sekvenční logické obvody



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

Dvojčinné klopné obvody

Sekvenční logické obvody

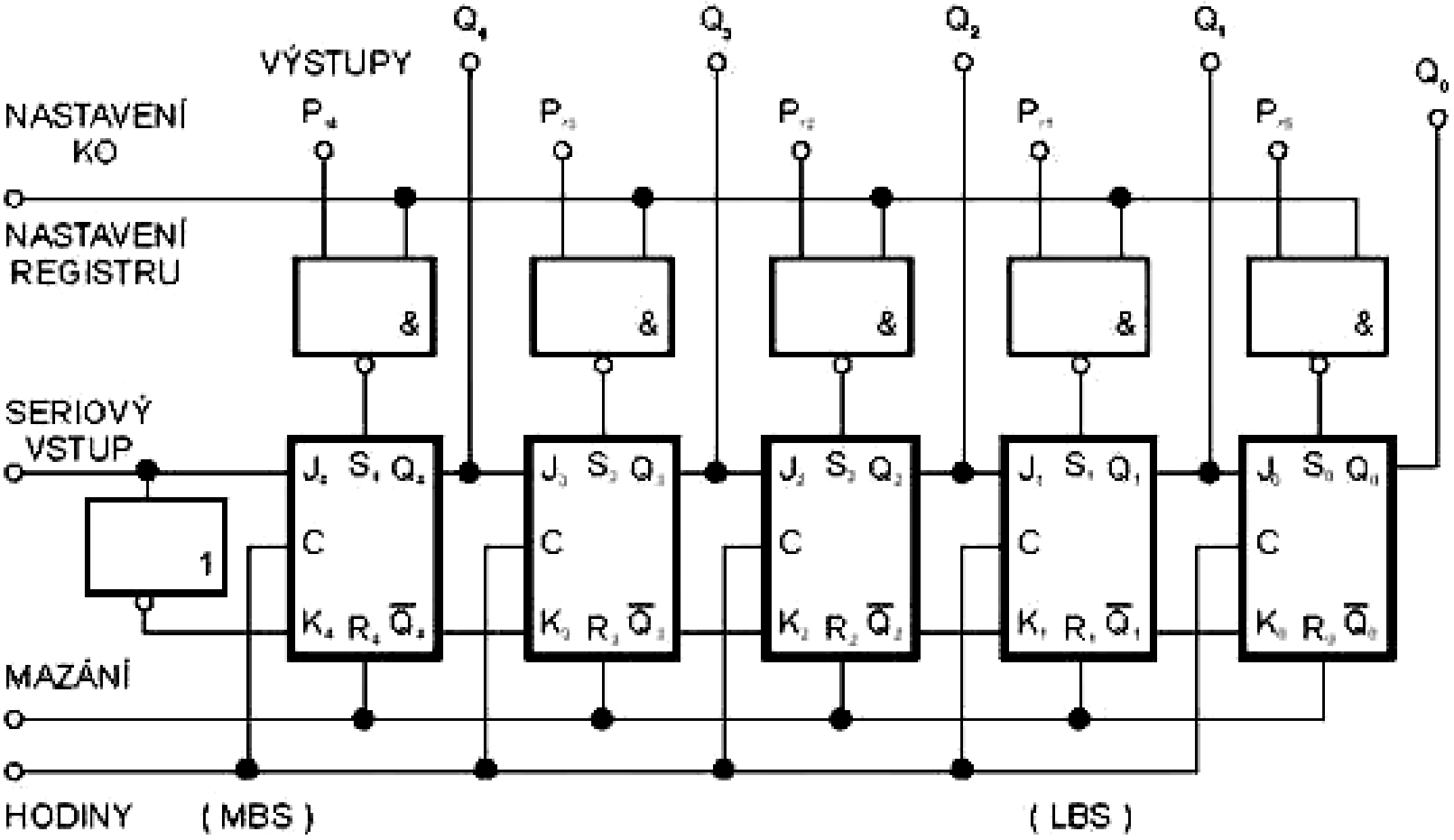


Obsah je postupně krokován z jednoho hradla k druhému.

Dvojčinné klopné obvody

Sekvenční logické obvody

Řetězec za sebou zapojených dvojčinných klopných obvodů. Dvojčinný obvod je řízen sestupnou a vzestupnou hranou pulzu.

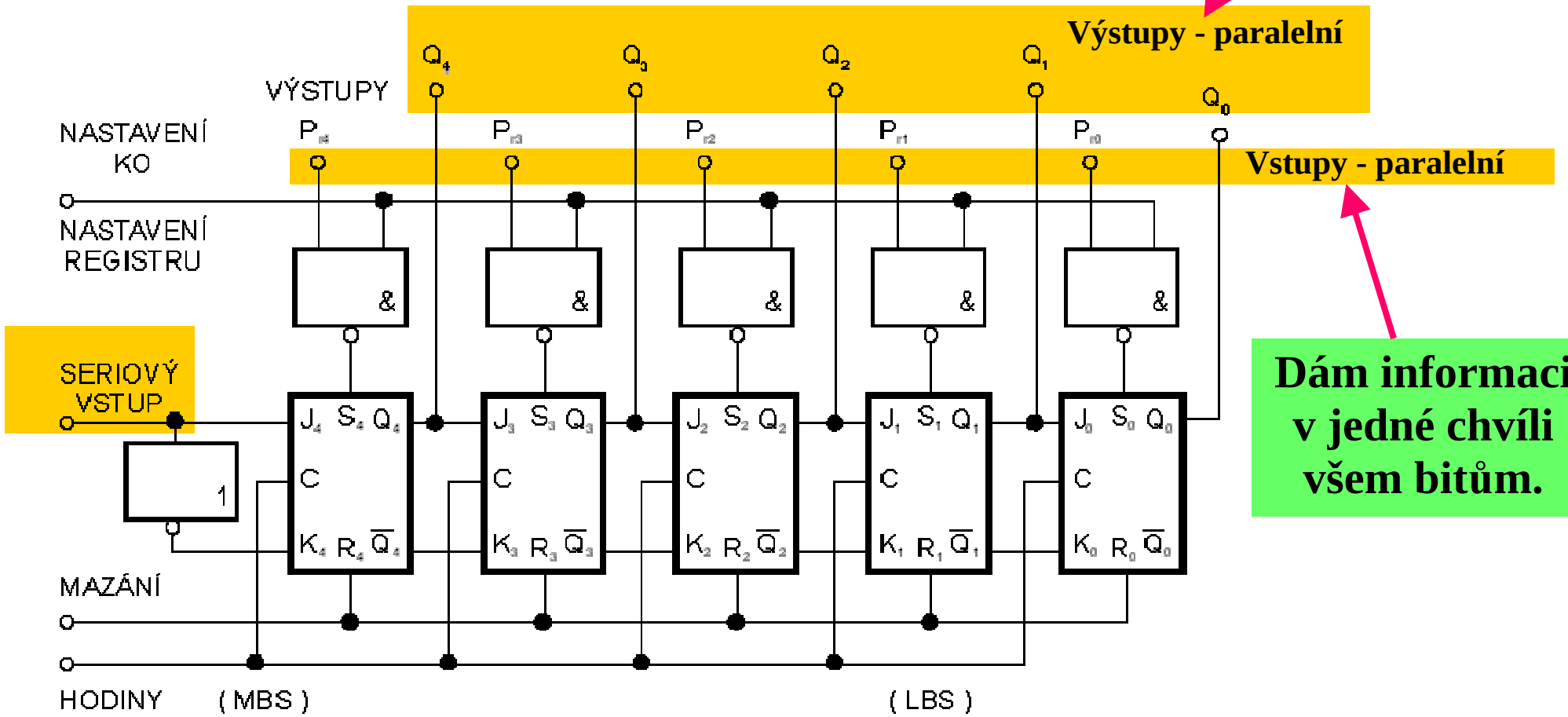


Statický registr

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.

Sekvenční logické obvody

Sériový vstup dat

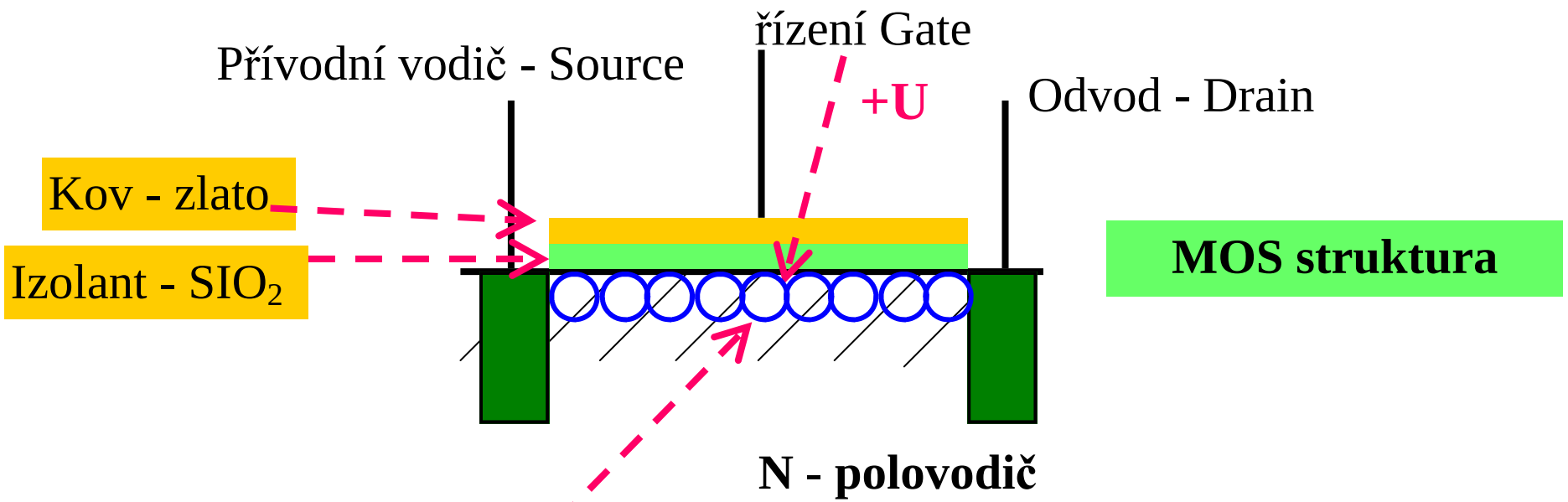
- **chceme zapsat do registru binární číslo 01011**

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

Dynamický registr

Sekvenční logické obvody

- **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

Sekvenční logické obvody

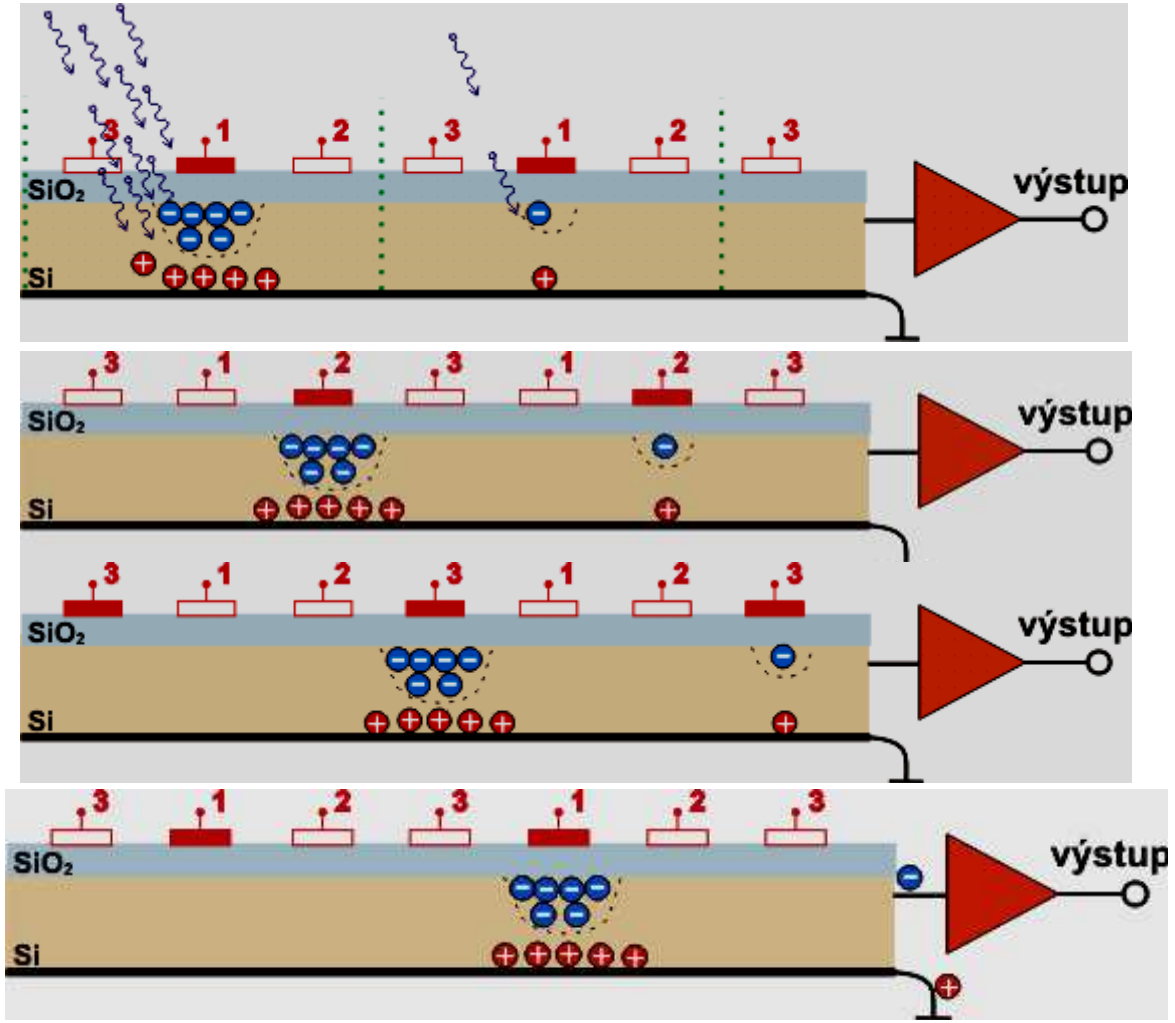
Struktura CCD (Charge Coupled Device = nábojově vázané struktury)

- 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

Dynamický registr

Sekvenční logické obvody

Posouvání náboje pod buňkami CCD struktury:



Sekvenční logické obvody

Vkládání informací do posuvného registru

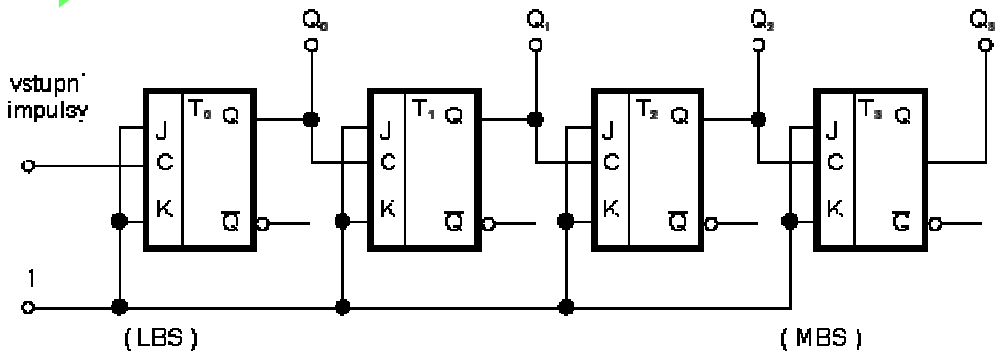
Sériový vstup klopného obvodu (odpovídá bitu s nejnižší váhou) umožňuje čtyři různé možnosti ovládání vkládání informací:

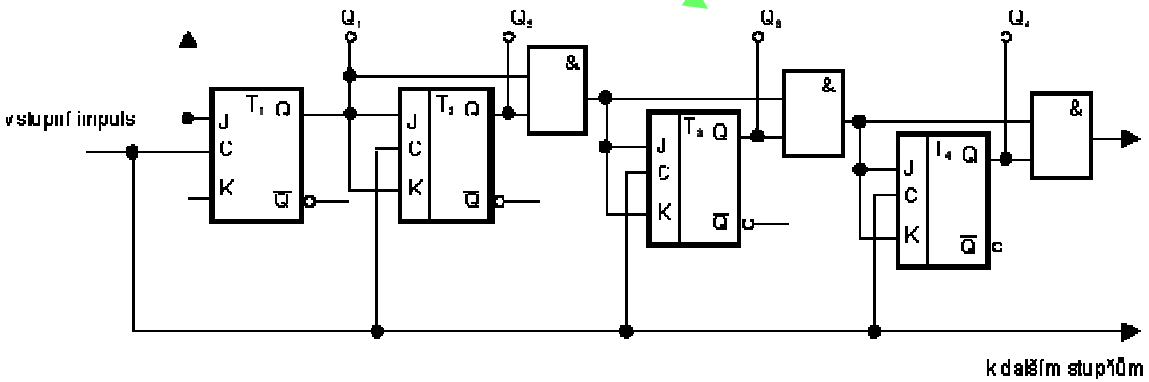
- **Náhrada nulami - bity se v registru při posuvu uvolňují od nejnižší váhy. Spojíme-li sériový vstup s nulovou logickou úrovní, pak jsou na uvolněná místa vkládány nuly.**
- **Náhrada jedničkami - sériový vstup je spojen s logickou úrovní odpovídající jedničce.**
- **Sériový zápis informací - sériový 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**

Sekvenční logické obvody

- ⇒ 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

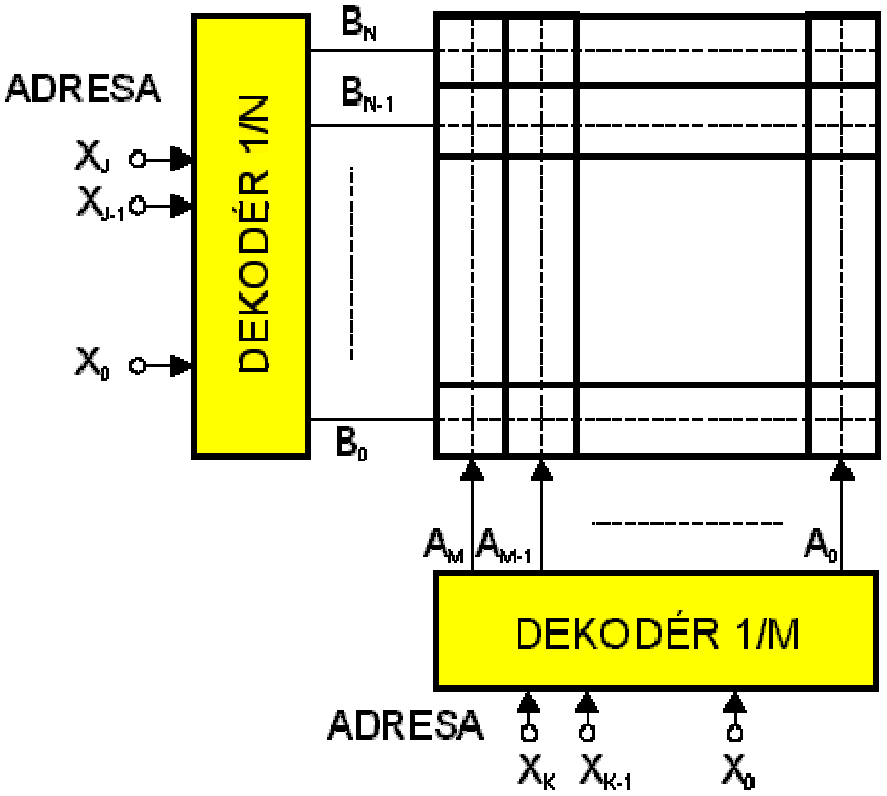


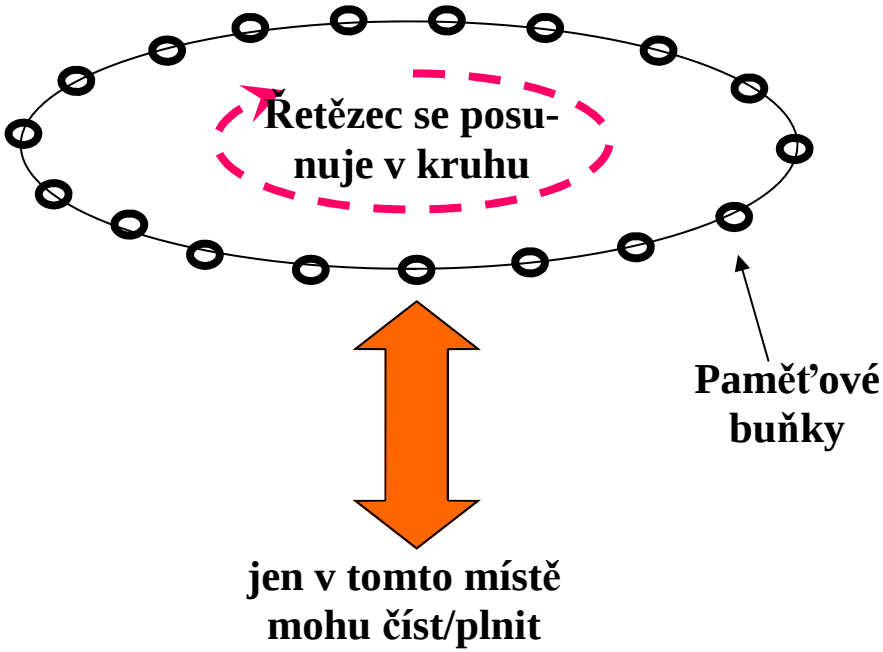


Polovodičové paměti

Podle typu přístupu mohou být paměti rozděleny na:

- **RAM (Random Access Memory) - paměti s libovolným přístupem**
- **SAM (Serail Access Memory) - paměti se sériovým přístupem.**

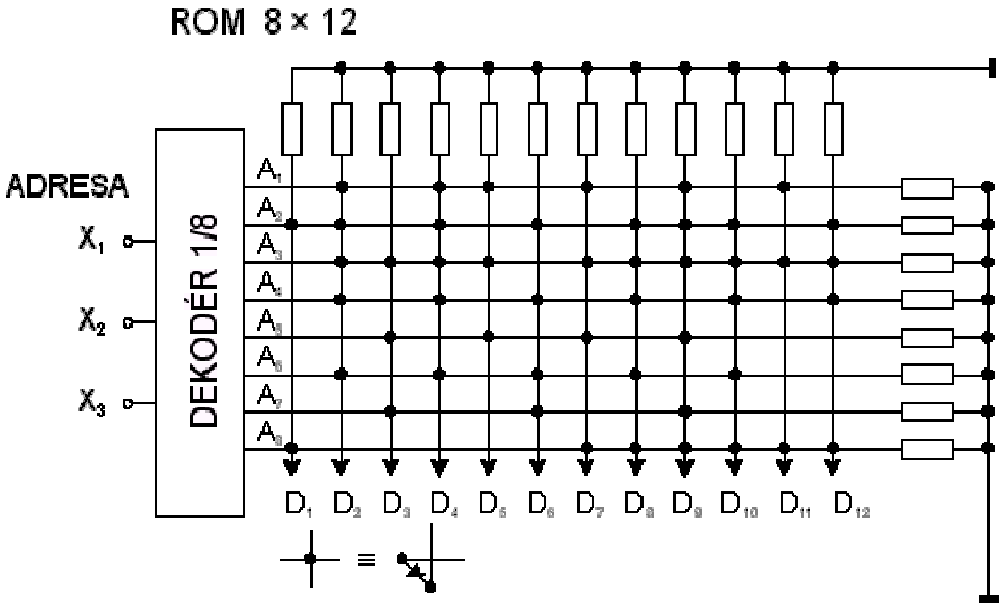




Polovodičové paměti

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**

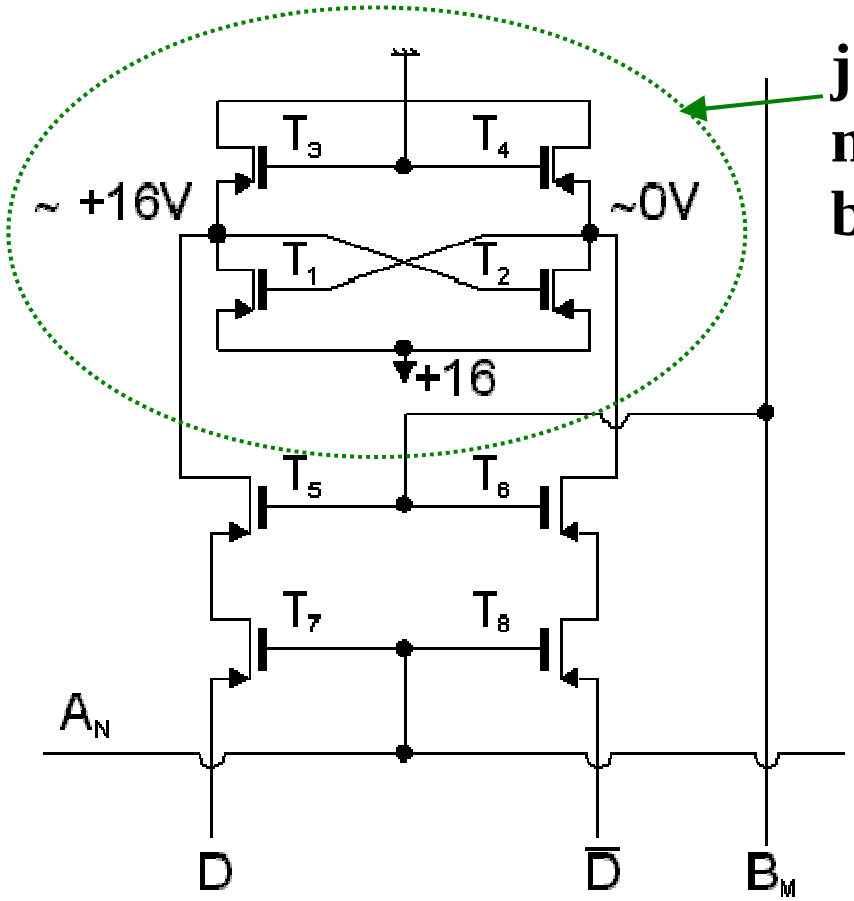
Polovodičové paměti

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

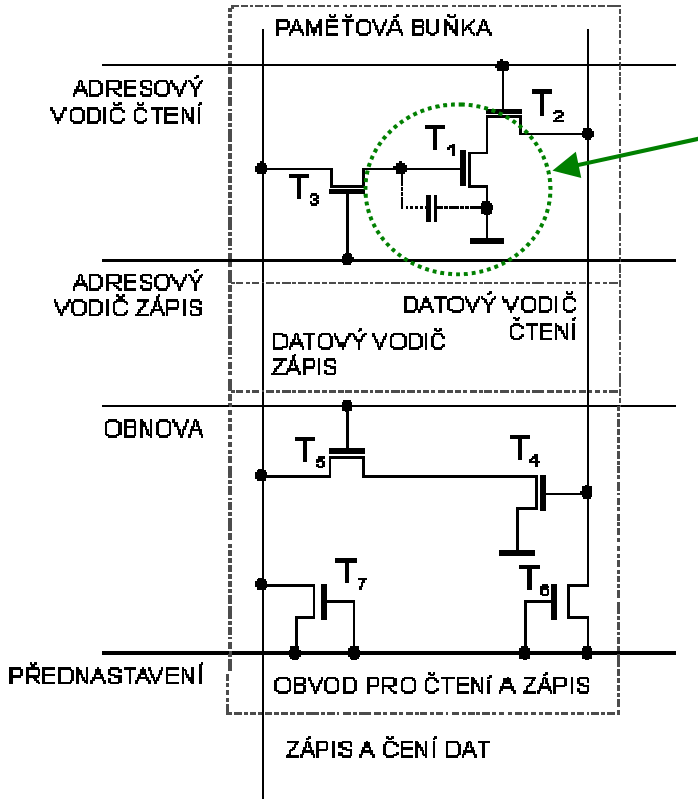
⇒**Statické SRAM hodně součástek - malé kapacity**

⇒**Dynamické DRAM jedna součástka = jeden bit - velké kapacity**

Informace = náboj kapacity tzn. musím obnovovat = refresh



jedna paměťová buňka



jedna paměťová buňka