

6. LOGICKÉ OBVODY

6.1. ZÁKLADNÉ POJMY

A. Definície základných pojmov

Logický obvod je hmotný systém realizujúci prostredníctvom logických signálov logickú funkciu

pričom

logická funkcia F konečného počtu nezávislých logických premenných je akékoľvek jednoznačné priradenie pravdivostnej hodnoty závislej logickej premennej Y pravdivostným hodnotám nezávislých logických premenných $X_1, X_2, X_3, \dots, X_n$:

$$Y = F(X_1, X_2, X_3, \dots, X_n);$$

logický signál je fyzikálny prostriedok, pomocou ktorého sa prenáša informácia o pravdivostnej hodnote logickej premennej, ktorého fyzikálna podstata môže byť rôzna a závisí od toho, ktorý fyzikálny jav alebo vlastnosť sa využíva pri realizácii logickej funkcie hmotným systémom.

Základné pojmy:

Výrok je každé tvrdenie, o ktorom je možné jednoznačne rozhodnúť, či je pravdivé alebo nepravdivé, pričom žiaden iný informačný obsah tohto tvrdenia nevyhodnocujeme.

Logická (pravdivostná) hodnota výroku označovaná ako 1 bit (**binary digit**) je informácia o tom, ktorý z dvoch možných prípadov pri vyhodnocovaní výroku nastal:

- výrok je pravdivý,
- výrok je nepravdivý.

Jednoduchý výrok je taký výrok, ktorého pravdivostná hodnota je daná len jeho obsahom. Pravdivostná hodnota jednoduchého výroku sa nazýva nezávislá logická premenná.

Zložený výrok je taký výrok, ktorý vznikne spájaním jednoduchých výrokov pomocou logických spojok a jeho pravdivostná hodnota závisí od

- pravdivostných hodnôt jednoduchých výrokov (logických operandov) z ktorých je spojený,
- zvolených logických spojok (logických operátorov).

Popis týchto funkcií je:

Symbol	Názov	Výrokový funktor	Booleov tvar
Y_1	konštanta falzum	nie je pravda	0
Y_2	konjunkcia (AND)	je A aj B	$A.B$
Y_3	priama inhibícia	je A a nie je B	$A.\bar{B}$
Y_4	asercia (opakovanie) A	je A	A
Y_5	spätná inhibícia	nie je A a je B	$\bar{A}.B$
Y_6	asercia (opakovanie) B	je B	B
Y_7	neekvivalencia (XOR)	je len A alebo len B	$A.\bar{B} + \bar{A}.B$
Y_8	disjunkcia (OR)	je A alebo B	$A+B$
Y_9	Piercova funkcia (NOR)	nie je A ani B	$\overline{A+B}$
Y_{10}	ekvivalencia	je A vtedy a len vtedy keď je B	$A.B + \bar{A}.\bar{B}$
Y_{11}	negácia (NOT) B	nie je pravda, že je B	\bar{B}
Y_{12}	spätná implikácia	ak je B, potom je A	$A + \bar{B}$
Y_{13}	negácia (NOT) A	nie je pravda, že je A	\bar{A}
Y_{14}	priama implikácia	ak je A, potom je B	$\bar{A} + B$
Y_{15}	Shefferova funkcia (NAND)	nie je A ani B	$\overline{A.B}$
Y_{16}	konštanta verum	je pravda	1

XNOR

$\overline{\overline{A.B} + \overline{A}.\bar{B}}$

Z logických funkcií uvedených v týchto tabuľkách je možné vybrať niekoľko úplných súborov a vybudovať na nich logickú algebru. Najznámejší z úplných súborov, na ktorých bola vybudovaná logická algebra je súbor logických funkcií Y_2 , Y_8 , Y_{11} , čiže logické funkcie AND, OR a NOT. Logická algebra vybudovaná na týchto funkciách sa podľa svojho autora Georgea Boolea nazýva **Booleova** alebo **booleovská algebra**.

B. Booleovská algebra

Všetky operácie v logických obvodoch prebiehajú v binárnom kóde, teda v dvojkovej (binárnej) číselnej sústave.

a) Binárna číselná sústava

V tejto sústave každá číslica predstavuje jeden bit, pričom určovanie poradia bitov začína posledným bitom, ktorý je 0-tý (predposledný bit je 1., ďalší je 2. atď.). Každému bitu prislúcha hodnota 2^n , kde n je poradie bitu.

Prevod čísel z decimálnej do binárnej sústavy získame postupným celočíselným delením decimálneho čísla číslom 2 až po výsledok 0, pričom zvyšky zapisujeme pod príslušného delenca. Tieto zvyšky v opačnom poradí predstavujú binárny tvar tohto decimálneho čísla.

PRÍKLAD

Binárny tvar decimálneho čísla 215:

$$215 : 2 = 107 : 2 = 53 : 2 = 26 : 2 = 13 : 2 = 6 : 2 = 3 : 2 = 1 : 2 = 0,$$

$$1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1$$

binárny tvar decimálneho čísla 215 teda je 11010111.

Prevod čísel z binárnej do decimálnej sústavy získame pomocou násobenia každého bitu jemu odpovedajúcou hodnotou 2^n . Súčet týchto súčinov predstavuje decimálny tvar tohto binárneho čísla.

PRÍKLAD

Dekadický tvar binárneho čísla 11010111:

$$1.2^0 + 1.2^1 + 1.2^2 + 0.2^3 + 1.2^4 + 0.2^5 + 1.2^6 + 1.2^7 =$$

$$= 1 + 2 + 4 + 0 + 16 + 0 + 64 + 128 = 215,$$

decimálny tvar binárneho čísla 11010111 je teda 215.

Základné operácie v binárnom kóde: základnými operáciami v booleovskej algebre sú súčet a súčin, pričom pre

- súčet platí: $0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 0$ s prenosom 1 do vyššieho rádu

a binárne sčítanie robíme tak, že binárne čísla sčítame po bitoch a ak vznikol medzi susednými bitmi prenos, pripočítame ho do vyššieho rádu.

PRÍKLAD:

$\begin{array}{r} 110011 \\ \underline{11011} \\ 100110 \end{array}$	$1 + 1 = 2 \rightarrow 10 \Rightarrow 0$ zapíšeme, 1 prechádza do vyššieho rádu
	$1 + 1 + 1 = 3 \rightarrow 11 \Rightarrow 1$ zapíšeme, 1 prechádza do vyššieho rádu,
	$1 + 0 + 0 = 1 \Rightarrow 1$ zapíšeme, $1 + 0 = 1 \Rightarrow 1$ zapíšeme,
	$1 + 1 = 2 \rightarrow 10 \Rightarrow 0$ zapíšeme, 1 prechádza do vyššieho rádu
	$1 + 1 = 2 \rightarrow 10 \Rightarrow 0$ zapíšeme, 1 prechádza do vyššieho rádu.

- súčin platí: $0 \cdot 0 = 0,$
 $0 \cdot 1 = 0,$
 $1 \cdot 0 = 0,$
 $1 \cdot 1 = 1.$

a binárne násobenie nahrádzame rovnako ako v dekadickej sústave posunom a sčítaním. Násobíme postupne a začíname poslednou číslicou násobiteľa, ktorou vynásobíme všetky číslice násobenca a výsledok zapíšeme tak, aby jeho posledná číslica bola zapísaná pod číslicou násobiteľa, ktorou sme násobili. Ďalej pokračujeme v násobení sprava doľava a jednotlivé čiastkové výsledky zapisujeme pod seba, pričom v každom ďalšom riadku pribúda jedna medzera sprava. Nakoniec získané čiastkové výsledky binárne sčítame.

PRÍKLAD:

$$\begin{array}{r}
 11010 \\
 \cdot \quad 101 \\
 \hline
 11010 \\
 00000 \\
 11010 \\
 \hline
 10000010
 \end{array}$$

b) Základné tvary booleovských funkcií

Booleovský súbor logických funkcií je úplný, teda vieme získať zápis ľubovoľnej logickej funkcie pomocou booleovských funkcií a to v zásade v dvoch tvaroch:

- štandardný súčtový tvar,
- štandardný súčinový tvar.

Štandardný súčtový tvar získame tak, že v každom riadku pravdivostnej tabuľky, kde má závislá premenná pravdivostnú hodnotu 1

- vynásobíme priamu alebo negovanú pravdivostnú hodnotu každej nezávislej premennej X_i podľa toho, či má táto premenná pravdivostnú hodnotu 1 alebo 0 prislúchajúcou premennou,
- získané súčiny vynásobíme,
- takto získané súčiny sčítame.

Štandardný súčinový tvar získame tak, že v každom riadku pravdivostnej tabuľky, kde má závislá premenná pravdivostnú hodnotu 0

- vynásobíme priamu alebo negovanú pravdivostnú hodnotu každej nezávislej premennej X_i podľa toho, či má táto premenná pravdivostnú hodnotu 1 alebo 0 prislúchajúcou premennou,

- získané súčiny sčítame,
- takto získané súčty vynásobíme.

Vo všeobecnosti možno povedať, že ak nezávislá premenná nadobúda prevažne pravdivostné hodnoty 0, použijeme súčtovú normálovú formu, ak nadobúda prevažne pravdivostné hodnoty 1, použijeme súčinovú normálovú formu

PRÍKLAD

Majme funkciu $Y = A \Rightarrow (B \Leftrightarrow C)$, ktorej pravdivostná

tabuľka je:

Napíšte

1. súčtovú normálovú formu,
 2. súčinovú normálovú formu
- tejto funkcie.

A	B	C	Y
1	1	1	1
1	0	1	0
1	1	0	0
1	0	0	1
0	1	1	1
0	0	1	1
0	1	0	1
0	0	0	1

1. **Súčtová normálová forma** tejto funkcie je

$$Y = A.B.C + A.B.C + A.B.C + A.B.C + A.B.C + A.B.C$$

2. **Súčinová normálová forma** tejto funkcie je

$$Y = (A + B + C).(A + B + C).$$

c) Pravidlá a zákony zjednodušovania booleovských funkcií

Pri určovaní pravdivostnej logickej funkcie je dôležité, aby bola táto funkcia čo najjednoduchšia, preto po procese jej zostavenia nasleduje proces jej zjednodušenia. Podobne ako v normálnej algebre, aj v booleovskej algebre je možné upravovať a zjednodušovať logické funkcie podľa určitých pravidiel a zákonov.

Pravidlá zjednodušovania logických funkcií:

- najprv upravujeme logické výrazy vo vnútri zátvoriek,
- potom upravujeme logickú funkciu podľa priority logických operátorov, pričom priority jednotlivých logických operátorov sú:
 1. negácia,
 2. logický súčin,
 3. logický súčet.

Zákony zjednodušovania logických funkcií:

Zákon	Súčtový tvar	Súčinový tvar
komutatívny	$A+B=B+A$	$A.B=B.A$
asociatívny	$A+(B+C)=(A+B)+C$	$A.(B.C)=(A.B).C$
distributívny	$A+(B.C)=(A+B).(A+C)$	$A+(B.C)=(A.B)+(A.C)$

De Morganov absorbcie absorbcie negácie	$\overline{A + B} = \overline{A} \cdot \overline{B}$ $A + (A \cdot B) = A$ $A + \overline{A} \cdot B = A + B$	$\overline{A \cdot B} = \overline{A} + \overline{B}$ $A \cdot (A + B) = A$ $A \cdot (\overline{A} + B) = A \cdot B$
vylúčenia	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
idempotencie	$A + A = A$	$A \cdot A = A$
neutrálnosti 0 a 1	$A + 0 = A$	$A \cdot 1 = A$
agresívnosti 0 a 1	$A + 1 = 1$	$A \cdot 0 = 0$
dvojitaj negácie A	$\neg\neg A = A$	$\neg\neg A = A$

d) Elektrické modely operácií booleovskej algebry

Definícia logických obvodov nehovorí nič o tom, na akom fyzikálnom princípe má logický obvod fungovať, ani akým spôsobom má byť realizovaný. Jedna z možností realizácie logických obvodov pomocou elektrického modelu. Vo všeobecnosti môže byť elektrický model logického obvodu realizovaný pomocou prvkov

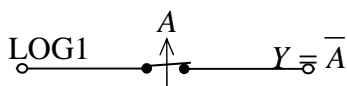
- analógových,
- číslicových (digitálnych).

Najjednoduchšia realizácia elektrického modelu logického obvodu je pomocou spínačov, kde

- logickým signálom je elektrický prúd,
- LOG0 znamená „netečie prúd“, LOG1 znamená „tečie prúd“.

Potom realizácia základných funkcií booleovskej algebry je:

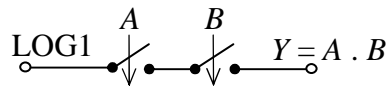
Funkcia NOT



Pravdivostná tabuľka:

VSTUP	VÝSTUP
A	NOT A
0	1
1	0

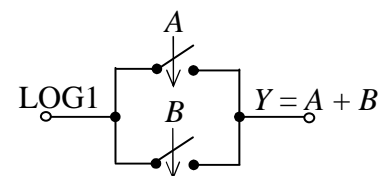
Funkcia AND



Pravdivostná tabuľka

VSTUPY		VÝSTUP
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Funkcia OR



Pravdivostná tabuľka

VSTUPY		VÝSTUP
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

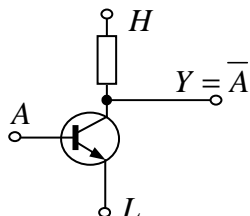
Iná realizácia elektrického modelu logických obvodov je pomocou polovodičových prvkov, ktoré výrazne menia svoj odpor v závislosti od vstupného

napätia a majú jednoducho definované dva stavy, v ktorých môžu dlhodobo zotrvať: stav, keď sú

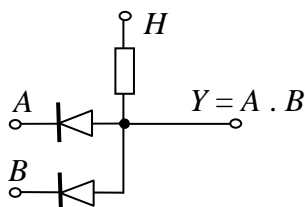
- úplne otvorené
- úplne zatvorené.

Potom realizácia základných funkcií booleovskej algebry je:

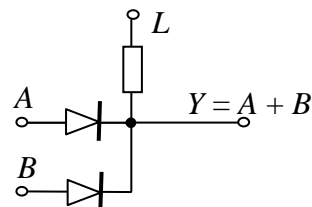
Funkcia NOT



Funkcia AND



Funkcia OR



V ďalšom sa budeme zaoberať len digitálnymi elektrickými modelmi logických obvodov.

6.2. NÁVRH LOGICKÝCH OBVODOV

Návrh alebo syntéza logických obvodov je jedna z dvoch základných typov úloh:

- analýza logického obvodu (určenie správania sa obvodu),
- syntéza logického obvodu (návrh obvodu s požadovaným správaním sa).

Analýza obvodu má vždy jednoznačné riešenie, syntéza obvodu vo všeobecnosti jednoznačné riešenie nemá (vždy existuje viac druhov logických obvodov s požadovaným správaním sa).

Logické obvody možno navrhnuť ako obvody s

- pevnou logikou (logika navrhnutá pre jednu logickú funkciu, ktorá sa nemení),
- flexibilnou logikou (logika navrhnutá tak, aby bola schopná prispôbiť sa realizácii viacerých logických funkcií).

Všeobecný algoritmus postupu návrhu logického obvodu:

Krok	Pevná logika	Flexibilná logika
1.	Definícia problému	Definícia problému
2.	Voľba súčiastkovej základne	Voľba univerzálneho mikropočítača
3.	Návrh schémy zapojenia obvodu	Vývojový diagram riešenia úlohy
4.	Návrh plošného spoja	Text zdrojového programu
5.	Výroba a osadenie dosky	Zavedenie programu do pamäte
6.	Test a oživovanie dosky	Ladenie programu

7.	Ak nastala chyba pri oživovaní, chod' na bod 6	Ak nastala chyba pri ladení, chod' na bod 6
8.	Ak vznikli nové požiadavky na systém, chod' na bod 3	Ak vznikli nové požiadavky na systém, chod' na bod 3
9.	Využívanie systému	Využívanie systému

Základné kroky pri návrhu obvodu:

1. Slovné zadanie rozložíme na výroky a priradíme im premenné (nezávislé aj závislé).
2. zložený výrok prepíšeme do tvaru logickej funkcie, pričom čiastkové výroky pospájame pomocou logických spojok (tento proces je náchylný na chyby, najmä pri zložitých zadaniach, preto sa niekedy vynecháva a nahrádza sa krokmi 3 a 4).
3. Zostrojíme pravdivostnú tabuľku zadania úlohy podľa pravidla, že ak je čiastkový výrok pravdivý, nadobúba premenná, ktorá je mu priradená logickú hodnotu 1, v opačnom prípade logickú hodnotu 0.
4. Zadanie logickej úlohy vyjadrené pravdivostnou tabuľkou prevedieme na štandardný súčtový alebo štandardný súčinový tvar.
5. Získaný algebraický výraz zjednodušíme.
6. Poskladáme obvod z booleovských členov a prepojíme vstupy a výstupy podľa predpisu, získaného z algebraického tvaru výrazu po kroku 5.
7. Testujeme činnosť obvodu (vykonávame jeho analýzu).

6.3. KLASIFIKÁCIA LOGICKÝCH OBVODOV

Logické obvody delíme do dvoch základných skupín:

- kombinačné obvody,
- sekvenčné obvody.

Kombinačné logické obvody sú také logické obvody, ktorých stav výstupov je jednoznačne daný stavom ich aktuálnych vstupov, teda v každom čase je možné priradiť akejkolvek kombinácii vstupov vždy tú istú príslušnú kombináciu výstupov.

Sekvenčné logické obvody (nazývané aj **sekvenčné automaty**) sú také logické obvody, u ktorých závisí stav výstupov okrem aktuálneho stavu vstupov aj od minulého stavu vstupov, teda sekvenčné obvody majú pamäť.

Kombinačné obvody ďalej delíme na

- jednoduché kombinačné obvody (hradlá),
- zložitejšie kombinačné obvody

a rovnako **sekvenčné obvody** delíme na

- jednoduché sekvenčné obvody (preklápacie alebo klopné obvody),
- zložitejšie sekvenčné obvody.

Jednoduché kombinačné obvody (hradlá) slúžia na realizáciu základných logických operácií a základnými sú:

hradlo NOT,

hradlo AND, NAND,

hradlo OR, NOR,

hradlo XOR, XNOR,

hradlo AND OR INVERT.

Zložitejšie kombinačné obvody (aritmetické jednotky) slúžia na realizáciu zložitejších aritmetických logických operácií a základnými sú:

sčítačka,

násobička,

multiplexor, demultiplexor,

prepínač,

komparátor,

kóder, dekodekér,

generátor parity,

aritmeticko – logická jednotka.

Jednoduché sekvenčné obvody (preklápacie resp. klopné) ďalej delíme podľa

- **stavov** na
 - bistabilné (Flip-flop), ktoré majú dva stabilné stavy, žiaden nestabilný stav; nachádzajú sa v jednom zo stabilných stavov a z jedného stabilného stavu do druhého stabilného stavu ich možno preklápať vstupom;
 - monostabilné (Monoflop), ktoré majú jeden stabilný stav a jeden nestabilný stav; nachádzajú sa v stabilnom stave, do nestabilného stavu ich možno preklopiť vstupom a obvod sa po čase sám preklopí do stabilného stavu;
 - astabilné alebo multivibrátory (Multivibrators), ktoré majú dva nestabilné stavy, žiaden stabilný stav; periodicky sa medzi týmito dvoma nestabilnými stavmi preklápajú, pričom doba, počas ktorej obvod zotrúva v danom stave sa nazýva časovou konštantou stavu, ktorá môže byť pre oba stavy rovnaká (symetrický obvod) alebo rôzna (nesymetrický obvod);
- **synchronizácie** na
 - asynchrónne alebo transparentné (Latch, Transparent latch), ktoré sa preklopiť ihneď po zmene úrovne na niektorom vstupe;

- asynchrónne, ktoré sa preklapia len v súčinnosti so synchronizačným vstupom (hodinovým, taktovacím), pričom **synchronizácia** môže byť
 - * úrovňová (hladinová), ktorá je realizovaná úrovňou (Gated latch, Clocked latch) hodinového signálu,
 - * kladná derivačná, ktorá je realizovaná nábežnou hranou hodinového (v tomto prípade taktovacieho) signálu (Positive edge triggered flip-flop),
 - * záporná derivačná, ktorá je realizovaná úbežnou hranou hodinového (aj v tomto prípade taktovacieho) signálu (Negative edge triggered flip-flop).

Základnými preklápacími obvodmi sú RS (SL, EL), RST, JK (T, D)

Zložitejšie sekvenčné obvody sú:

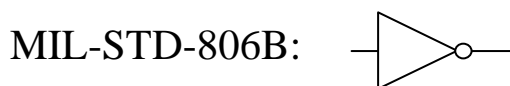
počítadlo inak čítač (nastaviteľný nahor, nadol, obojsmerný, binárny), register (posuvný, záchytný).

A. Jednoduché kombinačné obvody

Hradlo NOT (logická negácia $\rightarrow \bar{A}$)

Funkcia: jeho výstup je negáciou jeho vstupu; nazýva sa aj logický invertor.

Značka:



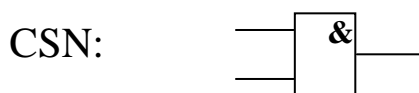
Pravdivostná tabuľka:

VSTUP		VÝSTUP
A		NOT A
0		1
1		0

Hradlo AND (logický súčin $\rightarrow A.B$)

Funkcia: jeho výstup je logickým súčinom všetkých jeho vstupov.

Značka:



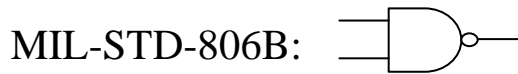
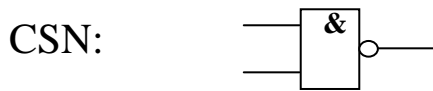
Pravdivostná tabuľka:

VSTUP		VÝSTUP
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Hradlo NAND (negovaný logický súčin $\rightarrow \overline{A.B}$)

Funkcia: jeho výstup je negáciou logického súčinu všetkých jeho vstupov.

Značka:



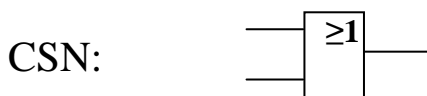
Pravdivostná tabuľka:

VSTUP		VÝSTUP
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Hradlo OR (logický súčet $\rightarrow A + B$)

Funkcia: jeho výstup je logickým súčtom všetkých jeho vstupov.

Značka:



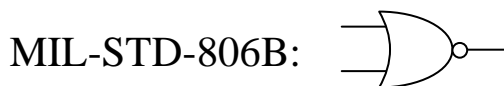
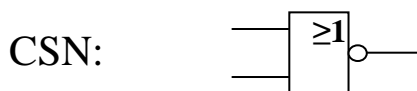
Pravdivostná tabuľka:

VSTUP		VÝSTUP
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Hradlo NOR (negovaný logický súčet – Peirceova funkcia $\rightarrow \overline{A + B}$)

Funkcia: jeho výstup je negácia logického súčtu všetkých jeho vstupov.

Značka:



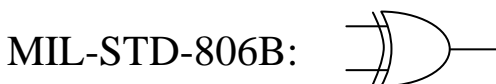
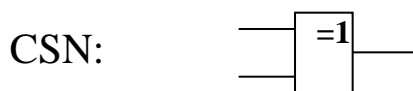
Pravdivostná tabuľka:

VSTUP		VÝSTUP
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Hradlo XOR (exkluzívny logický súčet $\rightarrow A \oplus B$)

Funkcia: jeho výstup je exkluzívnym logickým súčtom všetkých jeho vstupov

Značka:



Pravdivostná tabuľka:

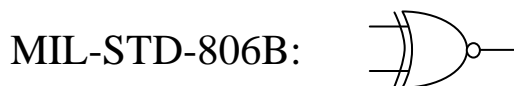
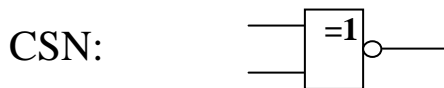
VSTUP		VÝSTUP
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

(výstup = log1 vtedy a len vtedy, ak sa logické hodnoty jeho vstupov líšia).

Hradlo XNOR (negácia exkluzívneho logického súčtu $\rightarrow \overline{A \oplus B}$)

Funkcia: jeho výstup je negácia exkluzívneho logického súčtu všetkých jeho vstupov.

Značka:



Pravdivostná tabuľka:

VSTUP		VÝSTUP
A	B	$A \text{ XNOR } B$
0	0	1
0	1	0
1	0	0
1	1	1

B. Zložitejšie kombinačné obvody (aritmetické jednotky)

Aritmetické jednotky podľa ich realizácie možno rozdeliť na

- paralelné – sú to kombinačné logické obvody, ktoré pracujú asynchrónne a zostavené bývajú z hradiel a sčítačiek;
- sériové – sú to sekvenčné logické obvody, ktoré pracujú synchronne a zostavené bývajú z generátorov hodinových pulzov, posuvných registrov, preklápacích obvodov a úplných jednobitových sčítačiek; sú pomalšie ako paralelné aritmetické jednotky, ale vyžadujú menej vstupov ako paralelné aritmetické jednotky.

V tejto kapitole sa budeme zaoberať **len paralelnými aritmetickými jednotkami**.

Sčítačka, odčítačka

Funkcia:

Sčítačka (binárna sčítačka) umožňuje sčítanie dvoch čísel A a B ($A + B$) reprezentovaných v binárnej číslicovej sústave.

Sčítačky podľa typu ich realizácie delíme na

- jednobitové sčítačky (polovičná sčítačka, úplná sčítačka),
- viacbitové sčítačky (sčítačka s propagáciou prenosu, sčítačka s predikciou prenosu).

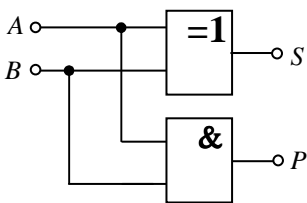
Odčítačka (binárna odčítačka) umožňuje odčítanie dvoch čísel A a B ($A - B$) reprezentovaných v binárnej číslicovej sústave a realizuje sa pomocou sčítačky, v ktorej sa použije inverzia k operandu, ktorý má byť odčítaný a ako pôvodný prenos sa použije

- * 1 ak je $A > B$,
- * 0 ak je $A < B$.

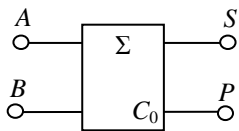
Jednabitové sčítačky

Polovičná sčítačka (Half Adder) umožňuje sčítať dve jednomiestne binárne čísla (dve binárne číslice). **Vstupom** sú dva 1-bitové sčítance A a B , **výstupom** sú 1-bitový aritmetický súčet S (hradlo XOR) a 1-bitový príznak prenosu do vyššieho rádu C (Carry Flag) v MIL-STD-806B resp. P v CSN (hradlo AND). Táto sčítačka síce odovzdáva ďalej príznak prenosu do vyššieho rádu, sama však nedokáže spracovať prenos z predchádzajúceho rádu, nestačí preto na realizáciu viacbitového sčítania.

Realizácia:



Schematická značka



Pravdivostná tabuľka:

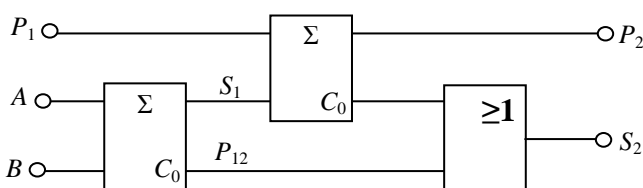
VSTUP		VÝSTUP	
B	A	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Booleovský zápis:

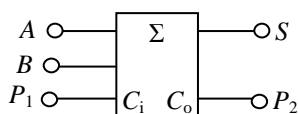
$$S = \bar{A} \cdot B + A \cdot \bar{B} \quad C = A \cdot B$$

Úplná sčítačka (Full Adder) umožňuje sčítanie dvoch 1-bitových binárnych čísel s pripočítaním prenosu z predchádzajúceho rádu (tri binárne číslice). **Vstupom** sú tri 1-bitové sčítance A , B , C_{in} (Carry-in) resp. P_1 , **výstupom** sú 1-bitový súčet S a 1-bitový príznak prenosu do vyššieho rádu C_{out} (Carry-out) resp. P_2 .

Realizácia:



Schématická značka:



Pravdivostná tabuľka:

VSTUP			VÝSTUP	
P_1	B	A	S	P_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Booleovský zápis:

$$S = (A + B) + P_1$$

$$P_2 = (A \cdot B) + (P_1 \cdot (A + B)) = (A \cdot B) + (B \cdot P_1) + (P_1 \cdot A) ???$$

Viacbitové sčítačky

***N*-bitová sčítačka s propagáciou prenosu resp. so sériovým prenosom (Ripple Carry Adder – **RCA**)** je zostavená z *N* úplných 1-bitových sčítačiek ich jednoduchým zreťazením. **Vstupom** tejto sčítačky sú potom dve *N*-bitové čísla + 1-bitový prenos z predchádzajúceho rádu, **výstupom** sú *N*-bitový súčet vstupných *N*-bitových čísel + 1-bitový prenos z predchádzajúceho rádu (spolu *N*+1 bitov).

Výhody tejto sčítačky:

- jednoduchý návrh,
- jednoduchá realizácia (počet hradieľ $\propto N$),
- rozšíriteľnosť na teoreticky ľubovoľný počet zreťazených úplných 1-bitových sčítačiek, t.j. na teoreticky ľubovoľný počet bitov.

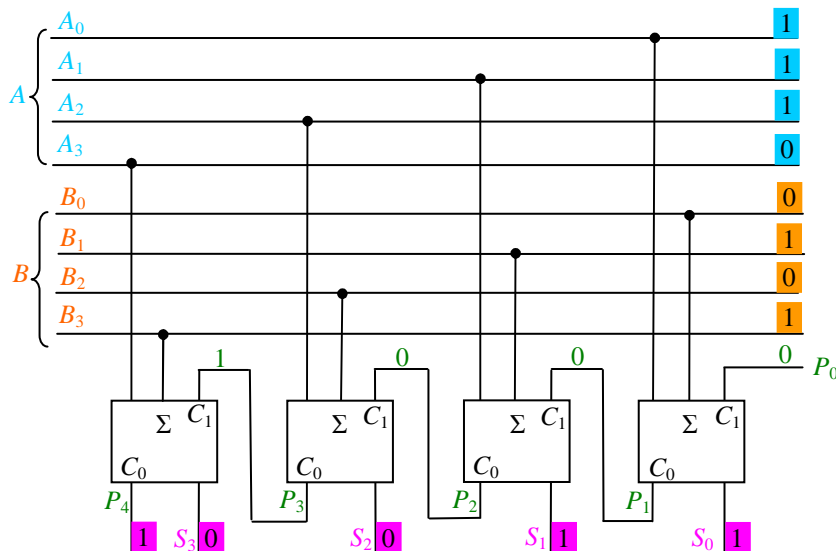
Nevýhody tejto sčítačky:

- s rastúcim počtom bitov priamo úmerne narastá celkové oneskorenie, ktoré je úmerné počtu hradlových oneskorení jednotlivých 1-bitových sčítačiek, (oneskorenie $\propto N$), pretože každá sčítačka musí „čakať“ na príznak prenosu od sčítačky predchádzajúceho bitu a toto oneskorenie výrazne obmedzuje jej priepustnosť t.j. maximálny možný počet sčítaní za jednotku času.

Pre **4-bitovú paralelnú sčítačku** so sériovým prenosom je

- sčítanec: $A = 1\ 1\ 1\ 0$
sčítanec: $B = \underline{0\ 1\ 0\ 1}$
súčet $\quad 1\ 0\ 0\ 1\ 1$
 $\quad 1\ 1\ 0\ 0 \quad \leftarrow$ prenos
 $\quad P_4\ P_3\ P_2\ P_1\ P_0$

- **realizácia:**



***N*-bitová sčítačka s predikciou prenosu (Carry Lookahead Adder – CLA)** je zostavená z

- * *N* úplných 1-bitových sčítačiek ich jednoduchým zret'azením,
- * jednotkou predikcie prenosu (Look-ahead Carry Unit – LCU), čo je kombinačný obvod, ktorý vypočítava všetky prenosy medzi rádmi súčasne.

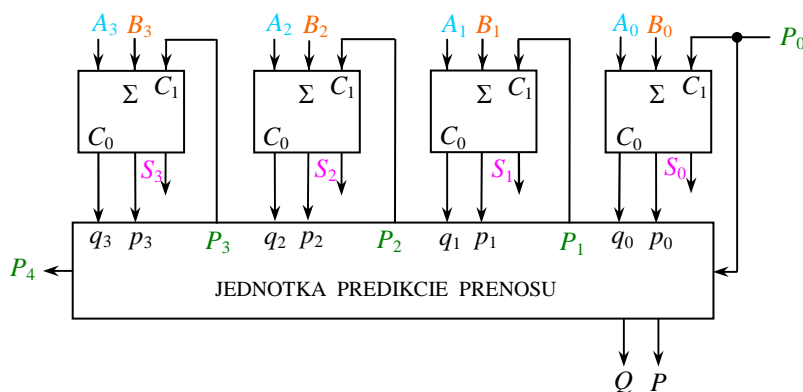
Výhody tejto sčítačky:

- je schopná vypočítať prenos v kratšom čase (v jednotkách hradlového oneskorenia) ako sčítačka s propagáciou prenosu, keďže každá 1-bitová sčítačka nemusí „čakať“ na korektný príznak prenosu od sčítačky predchádzajúceho bitu (oneskorenie $\square 1$);

Nevýhody tejto sčítačky:

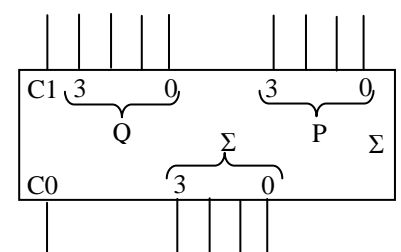
- vyššie nároky na celkový počet hradiel (počet hradiel $\square N^2$) a počet tranzistorov (počet tranzistorov $\square N^3$),
- od určitého počtu bitov prestáva byť prakticky realizovateľná (kvôli počtu hradiel a výrobným obmedzeniam na počet vstupov jedného hradla); široké sčítačky je preto potrebné realizovať viacúrovňovo – kombinovaním princípu predikcie prenosu (na nižšej úrovni) s propagáciou prenosu (na vyššej úrovni), prípadne použitím princípu predikcie prenosu na dvoch (alebo viacerých) úrovniach.

Realizácia:



Poznámka:

Všeobecný symbol 4-bitovej paralelnej sčítačky je:



Násobička

Funkcia: (Binárna) násobička (Binary Multiplier) umožňuje násobenie dvoch čísel *A* a *B* (*A*·*B*) reprezentovaných v binárnej číslicovej sústave. Paralelná násobička postupne násobí násobencov jednotlivými číslicami násobiteľ'a a takto vytvorené čiastočné násobky postupne sčíta.

Realizácia:

Násobička môže byť zostavená viacerými spôsobmi, ale pretože násobenie je v binárnej sústave nahrádzané posunom a sčítaním, najčastejšie pozostáva z

- * hradíel AND,
- * sčítačiek

ktorých počet je určený počtom bitov násobených binárnych čísel.

Pre **4-bitovú paralelnú násobičku** bude:

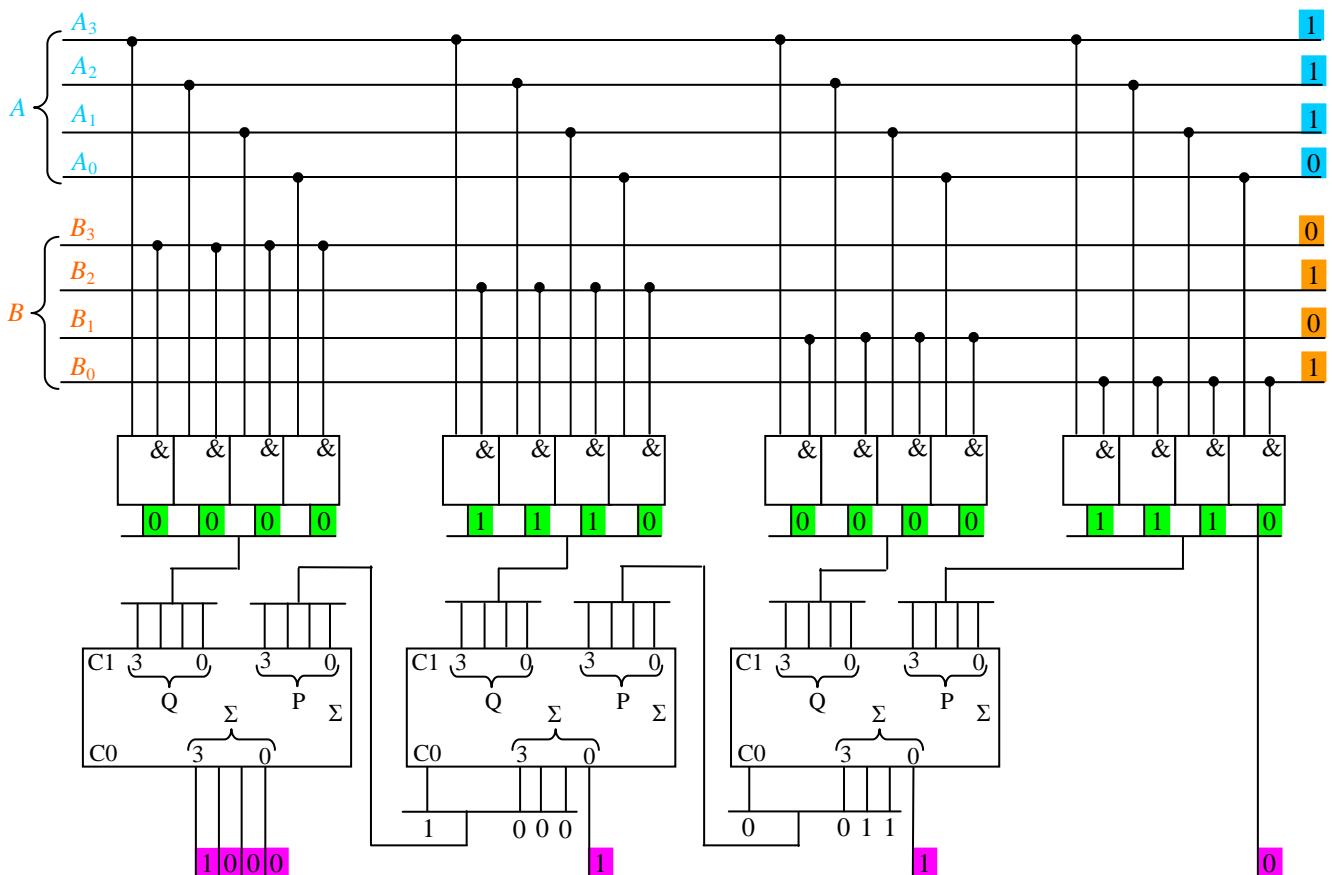
- **počet hradíel a sčítačiek:**

násobenec: $A = 1110$

násobiteľ: $B = 0101$

$$\begin{array}{r}
 B_0 = 1: 1.1110 = \quad \quad \quad 1110 \Rightarrow 4 \text{ hradlá AND,} \\
 B_1 = 0: 0.1110 = \quad \quad \quad 0000 \Rightarrow 4 \text{ hradlá AND,} \\
 \quad \quad \quad \quad \quad \quad \quad 0111 \Rightarrow 4\text{-bitová sčítačka,} \\
 B_2 = 1: 1.1110 = \quad \quad \quad 1110 \Rightarrow 4 \text{ hradlá AND,} \\
 \quad \quad \quad \quad \quad \quad \quad 10001 \Rightarrow 4\text{-bitová sčítačka,} \\
 B_3 = 0: 0.1110 = \quad \quad \quad 0000 \Rightarrow 4 \text{ hradlá AND,} \\
 \quad \quad \quad \quad \quad \quad \quad 1000 \downarrow \downarrow \downarrow \\
 \text{súčin} \quad \quad \quad 1000 \quad 11 \quad 0 \Rightarrow 4\text{-bitová sčítačka.}
 \end{array}$$

- **realizácia:**



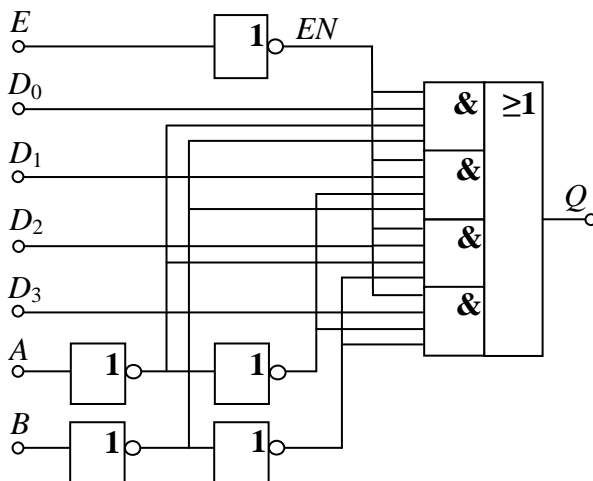
Multiplexor

Funkcia: Multiplexor prevádza paralelný binárny signál na sériový binárny signál. Má N adresových vstupov a_i , maximálne 2^N dátových informačných vstupov $X_i (\equiv D_i)$ ($i = 0, 1, 2, \dots, N - 1$) a 1 výstup $Y (\equiv Q)$. Umožňuje preniesť informáciu z niektorého z N adresových vstupov do výstupu pričom stav výstupu je zhodný so stavom toho adresového vstupu, ktorý je určený adresovými vstupmi a_i .

Pre 4-kanálový multiplexor:

- **tabuľka** bude obsahovať:
 - 2 adresové vstupy B, A (B má najväčšiu váhu, A má najmenšiu váhu)
 - 4 dátové informačné vstupy D_0, D_1, D_2, D_3 ,
 - 1 výstup Q ;
- **realizácia** je:

VSTUP			VÝSTUP
EN	B	A	Q
0	X	X	0
1	0	0	D_0
1	0	1	D_1
1	1	0	D_2
1	1	1	D_3



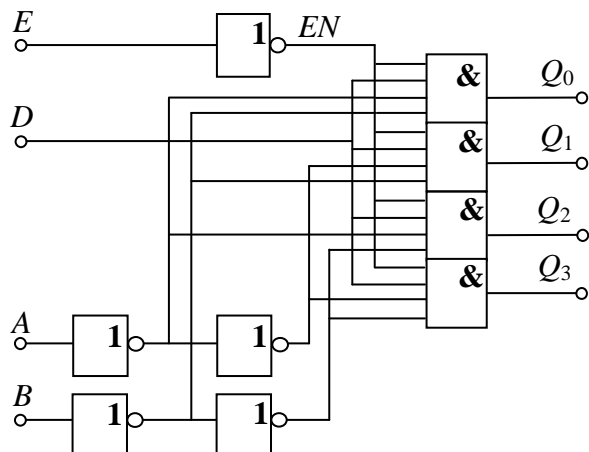
Demultiplexor

Funkcia: Demultiplexor prevádza sériový binárny signál na paralelný binárny signál. Umožňuje preniesť informáciu z jedného vstupu $X (\equiv D)$ na niektorý z N výstupov $Y_i (\equiv Q_i)$, ($i = 0, 1, 2, \dots, N - 1$), ktorý je určený adresovými vstupmi a_i .

Pre 4-kanálový demultiplexor:

- **tabuľka** bude obsahovať:
 - 2 adresové vstupy B, A ,
 - 1 dátový informačný vstup D ,
 - 4 výstupy Q_0, Q_1, Q_2, Q_3 ;
- **realizácia** je:

VSTUP			VÝSTUP			
EN	B	A	Q_0	Q_1	Q_2	Q_3
1	0	0	D	0	0	0
1	0	1	0	D	0	0
1	1	0	0	0	D	0
1	1	1	0	0	0	D



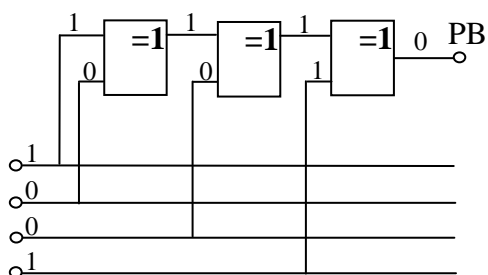
Generátor parity

Funkcia: Generátor parity (Parity Generator) je kombinačný logický obvod, generuje jednoduchý paritný výstup tzv. paritný bit (PB).

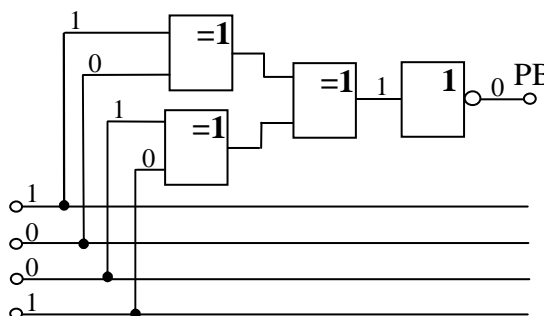
Jeho **vstupom** je počet jednotiek (9 vstupov), **výstupom** je parita vstupu, pričom ak je počet 1 vo vstupoch párný, pre párnú paritu $PB = 0$
nepárnu paritu $PB = 1$

Realizácia:

kaskáda XOR hradíel



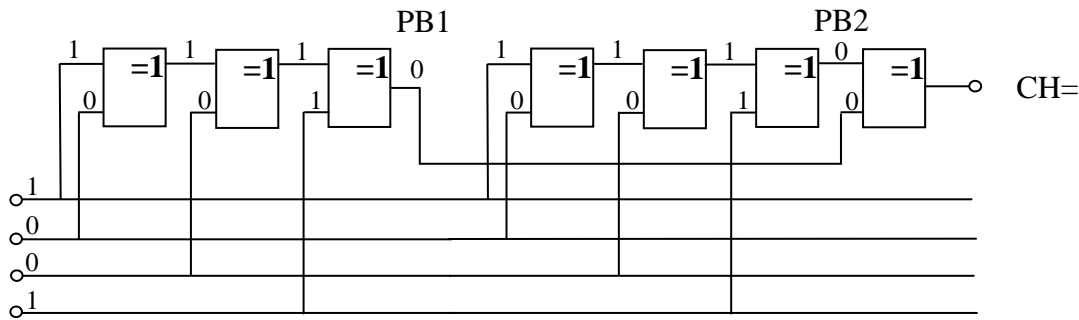
paralelné XOR hradlá a hradlo NOR



Pravdivostná tabuľka:

POČET 1 VO VSTUPOCH	PÁRNÝ VÝSTUP	NEPÁRNÝ VÝSTUP
0, 2, 4, 6, 8	0	1
1, 3, 5, 7, 9	1	0

Používa sa na identifikáciu vzniku chyby v kombinácii binárnych premenných pri jej prenose alebo uschovaní. Pri prehliadke kombinácie binárnych premenných generuje paritný bit tejto kombinácie, ktorý je prenášaný alebo uschovaný spolu s touto kombináciou a použije sa ku jej kontrole pri prenesení alebo uschovaní.



Aritmeticko – logická jednotka

Aritmeticko – logická jednotka (**A**rithmeti**C** **L**ogic **U**nit – **ALU**) je centrálna časť procesora, v ktorej sa vykonávajú základné aritmetické a logické operácie s číslami: sčítanie, odčítanie, násobenie, delenie, logický posun, negácia, komplement atď. s dvoma operandmi.

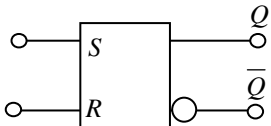
Typickými subsystémami tejto jednotky sú sčítačka, odčítačka, násobička, delička, negátor, logický súčet, posuvný register.

C. Jednoduché sekvenčné obvody

a) Bistabilné preklápacie obvody asynchrónne

Preklápací obvod RS (Reset - nulovanie, Set - nastavenie)

Značka:



kde S a R sú **vstupy** (nastavovací S a nulovací R , ktorých základný stav je log 0;

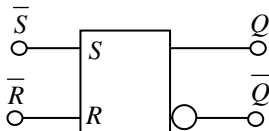
Q a \bar{Q} sú **výstupy**, ktorých logické hodnoty sú opačné, pričom uložená hodnota je k dispozícii na výstupe Q .

Ak je tento obvod

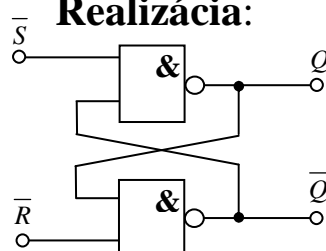
- realizovaný dvoma hradlami NAND,
- nastavovaný (do 1 na výstupe Q) log. 0, je nastavovací vstup označovaný \bar{S} a nulovací vstup označovaný \bar{R}

a je to **preklápací obvod $\bar{R}\bar{S}$** .

Značka:



Realizácia:



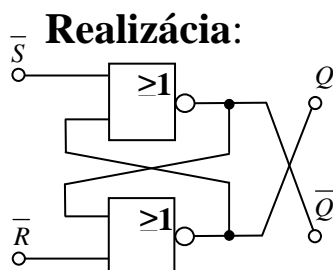
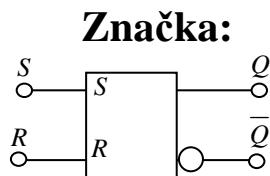
Pravdivostná tabuľka

\bar{S}	\bar{R}	Q_{n+1}	
0	0	Q_n	zachovanie stavu
0	1	0	vynulovanie
1	0	1	nastavenie
1	1	?	zakázaný stav

Ak je tento obvod

- realizovaný dvoma hradlami NOR,
- nastavovaný (do 1 na výstupe Q) log. 1, je nastavovací vstup označovaný S a nulovací vstup označovaný R

a je to **preklápací obvod RS**.



Pravdivostná tabuľka

S	R	Q_{n+1}	
0	0	Q_n	zachovanie stavu
0	1	0	nulovanie
1	0	1	nastavenie
1	1	?	zakázaný stav

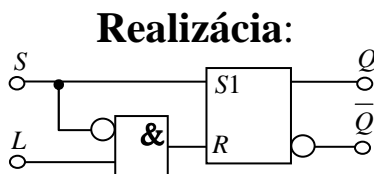
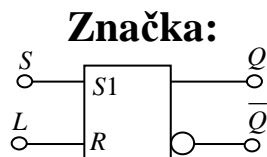
Poznámka:

Preklápacie obvody \overline{RS} aj RS

- majú jeden zakázaný stav,
- pri súhlasnom buzení oboch vstupov sa môžu dostať do neurčitého stavu.

Preklápací obvod SL (Set - nastavenie, Latch – blokovanie)

je RS preklápací obvod s dominantným S vstupom.



Pravdivostná tabuľka

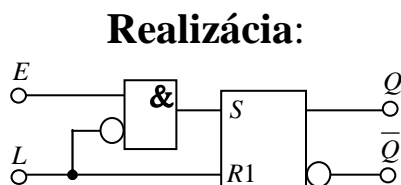
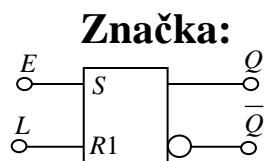
S	L	Q_{n+1}	
0	0	Q_n	zachovanie stavu
0	1	0	nulovanie
1	0	1	nastavenie
1	1	1	nastavenie

Poznámka:

Preklápací obvod SL nemá žiaden zakázaný ani neurčitý stav

Preklápací obvod EL (Erase – vymazanie, Latch – blokovanie)

je RS preklápací obvod s dominantným R vstupom.



Pravdivostná tabuľka

E	L	Q_{n+1}	
0	0	Q_n	zachovanie stavu
0	1	0	nulovanie
1	0	1	nastavenie
1	1	0	nulovanie

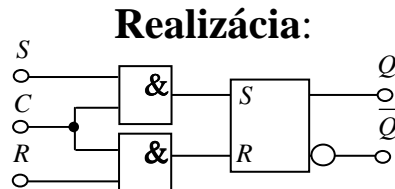
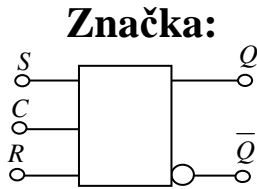
Poznámka:

Preklápací obvod EL tiež nemá žiaden zakázaný ani neurčitý stav.

b) Bistabilné preklápacie obvody synchronne

Preklápací obvod RST (Gated RS latch)

je synchronný variant obvodu RS.



Pravdivostná tabuľka

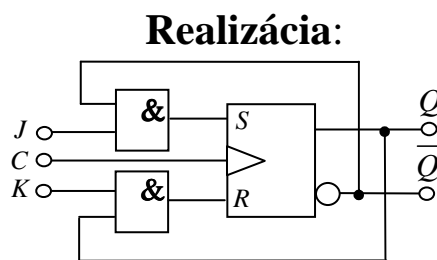
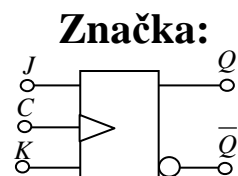
S	R	C	Q_{n+1}	
X	X	0	Q_n	zachovanie stavu
0	0	1	Q_n	zachovanie stavu
0	1	1	0	nulovanie
1	0	1	1	nastavenie
1	1	1	?	zakázaný stav

K preklopeniu obvodu dochádza len v závislosti od hodnoty signálu na hodinovom vstupe **C** (Clock).

Obvod je synchronizovaný úrovňou hodinového signálu (hladinová synchronizácia); stav je možné meniť po celú dobu trvania hodinového impulzu.

Preklápací obvod JK (Jack Kilby; J = S, K = R)

je základným synchronným preklápacím obvodom. V podstate je rozšírením obvodu RST o interpretáciu vstupnej kombinácie $R = S = 1$, ktorá je v RS aj RST obvodoch zakázaná; pri tejto kombinácii obvod JK invertuje (neguje) uloženú hodnotu.



Pravdivostná tabuľka

J	K	C	Q_{n+1}	
X	X	0	Q_n	zachovanie stavu
0	0	1/↑	Q_n	zachovanie stavu
0	1	1/↑	0	nulovanie
1	0	1/↑	1	nastavenie
1	1	1/↑	$\overline{Q_n}$	negovanie stavu (preklopenie)

Obvod JK môže byť synchronizovaný

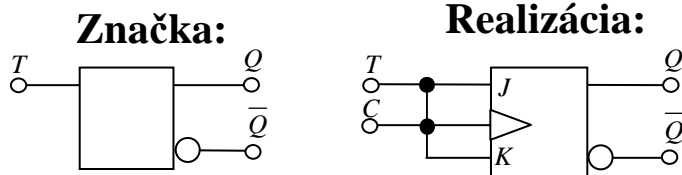
- nábežnou hranou (↑) taktovacieho,
- úbežnou hranou (↓) taktovacieho,
- úrovňou hodinového signálu.

Ak je JK preklápací obvod synchronizovaný jednou hranou taktovacieho signálu, jeden taktovací signál (nábežná alebo úbežná hrana taktovacieho signálu) spôsobí jedno jeho preklopenie a to ak $J = K = 1$.

Preklápačí obvod T (Toggle – preklopiť)

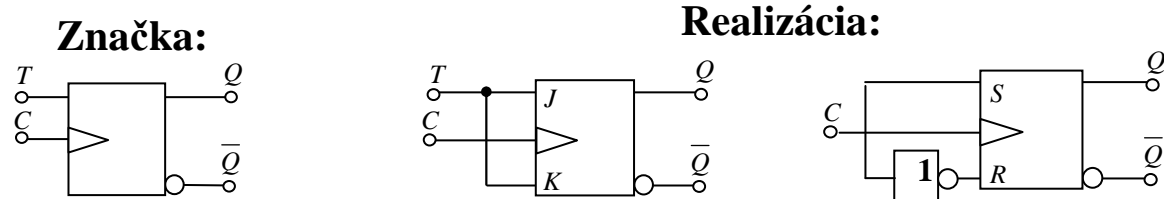
- asynchrónny je preklápačí obvod s jediným vstupom T
- synchronný je preklápačí obvod so vstupom T a hodinovým vstupom C .

Asynchrónny preklápačí obvod T je možné vytvoriť napríklad z obvodu JK prepojením všetkých jeho vstupov J , K a C do jedného: $J = K = C = T$.



Synchronný preklápačí obvod T je možné vytvoriť napríklad z

- obvodu JK prepojením jeho vstupov J a K do jedného: $J = K = T$,
- hranou synchronizovaného preklápacieho obvodu RST a invertora.



Pravdivostná tabuľka

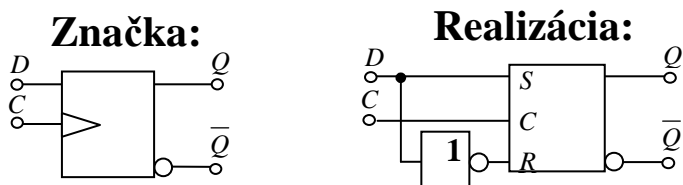
T	Q_n	Q_{n+1}	
0	0	Q_n	zachovanie stavu
0	1	Q_n	zachovanie stavu
1	0	1	negovanie stavu
1	1	0	negovanie stavu

Obvod T tvorí základ čítačov a deličiek frekvencie. Po privedení pravouhlého signálu s frekvenciou f na vstup T asynchrónneho obvodu T (prípadne na vstup C synchronného obvodu pri súčasnom nastavení vstupu T na log.1), dostaneme na jeho výstupe signál s frekvenciou $f/2$.

Preklápačí obvod D (Delay – zdržanie)

je synchronný preklápačí obvod s dátovým D (Data) a hodinovým C (Clock) vstupom.

Možno ho vytvoriť z obvodu RST tak, že na vstup S privedieme priamo hodnotu vstupu D a na vstup R jeho negovanú hodnotu.



Pravdivostná tabuľka

D	C	Q_{n+1}	
X	0	Q_n	zachovanie stavu
0	1/↑	0	nulovanie
1	1/↑	1	nastavenie

Obvod D realizuje jednobitovú pamäť a je základom posuvných registrov.

C. Zložitejšie sekvenčné obvody

Počítadlo (čítač)

V sekvenčne (sériovo) zapojených preklápacích obvodoch sa informácie, nachádzajúce sa v 1-bitových pamätiach posúvajú z výstupov obvodov na vstupy susedných obvodov, takže celok funguje ako **binárny čítač impulzov** (v binárnom kóde).

Podľa toho, ako sa preklápacie obvody, z ktorých sa čítače skladajú uvádzajú do činnosti (prijatie a odovzdanie informácie) rozlišujeme čítače

- **asynchrónne**, ktoré sú spúšťané samotnými vstupnými impulzmi,
- **synchrónne**, ktoré sú spúšťané synchrónnymi hodinovými impulzmi.

Pre k -bitový (k -miestny) binárny čítač je

- rozsah čísel: $n_{\max} = 2^k$, t.j. $0 \div k-1$ binárnych čísel;
- počet preklápacích obvodov: $k \geq \log_2 n_{\max}$,

teda napr. pre **2-bitový (2-miestny) binárny čítač** bude:

- rozsah čísel: $n_{\max} = 2^2 = 4$, t.j. $0 \div 3$ binárne čísla;
- počet preklápacích obvodov: $k \geq \log_2 4 = 2$.

Čítače možno zostrojiť z preklápacích obvodov

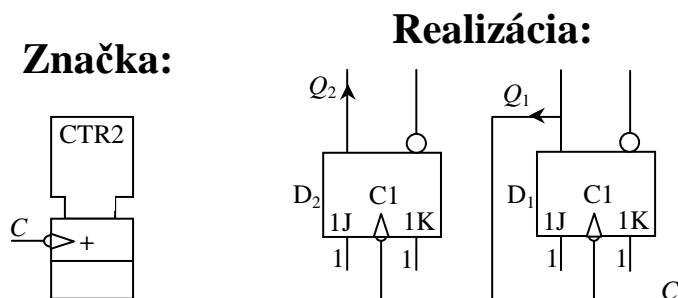
- JK,
- T,

ale čítače zostavené z T preklápacích obvodov vyžadujú viac pomocných hradiel.

Asynchrónne čítače

V asynchrónnom čítači je hodinový impulz privedený na hodinový vstup C prvého preklápacieho obvodu a ďalšie preklápacie obvody sú na svojich vstupoch budené výstupnými impulzmi svojich predchodcov.

Dvojbíťový asynchrónny čítač je:

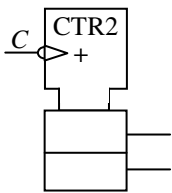


Synchrónne čítače

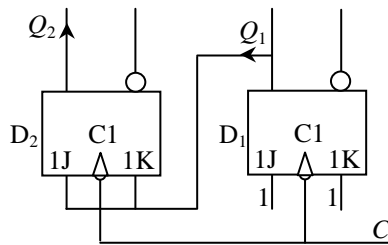
V synchrónnom čítači je spoločný hodinový impulz privedený paralelne na ho-

dinové vstupy všetkých preklápacích obvodov, takže informácie nachádzajúce sa v 1-bitových pamätiach sa posúvajú z výstupov obvodov na vstupy susedných obvodov synchronne s hodinami. takže synchronný čítač pracuje ako **posuvný pamäťový register**.

Značka:



Realizácia:



Pravdivostná tabuľka:

t_n		t_{n+1}	
Q_{2n}	Q_{1n}	Q_{2n+1}	Q_{1n+1}
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Pamäťové registre

Pamäťové registre možno chápať ako **špeciálny prípad synchronného čítača**. Podľa ich zapojenia ich delíme na

- **posuvné** (serial in, serial out – SISO, paralel in, serial out – PISO),
- **záchytné** (paralel in, paralel out – PIPO),
- **kombinované** (serial in, paralel out – SIPO).

Posuvné registre

V 4-bitovom posuvnom registri so **sériovým dátovým vstupom**

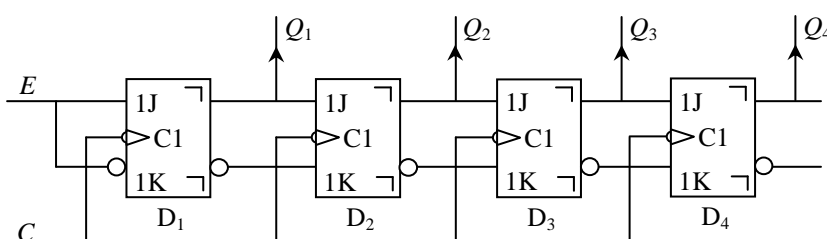
- informácia je privádzaná na vstupy J, K preklápacieho odvodu D_1 , výstupy obvodu D_1 sú vstupmi obvodu D_2 , výstupy obvodu D_2 sú vstupmi obvodu D_3 a výstupy obvodu D_3 sú vstupmi obvodu D_4 ;
- na úplné načítanie celej informácie sú potrebné štyri hodinové (taktovacie) impulzy

a ak sa štyri bity informácie posúvajú na výstup registra

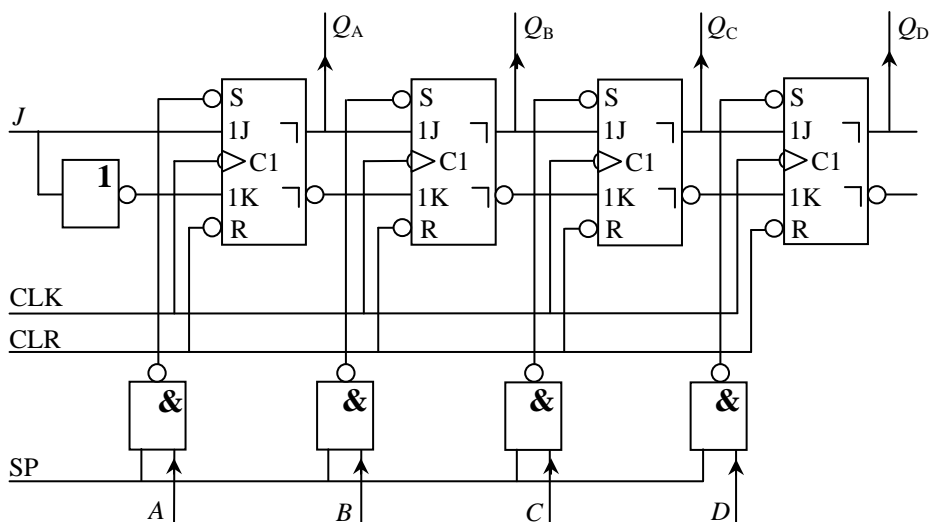
- **postupne** po jednotlivých bitoch, t.j. pri 4. takte sa na výstup registra presunie prvý bit, pri 5. takte druhý bit, pri 6. takte tretí bit a pri 7. takte štvrtý bit, **výstup je sériový**,
- **súčasne** všetky štyri bity, **výstup je paralelný**.

Realizácia:

sériový vstup, paralelný výstup:



kombinovaný register s nulovaním:



J – sériový vstup,
 A, B, C, D – paralelné vstupy,
 Q_A, Q_B, Q_C, Q_D – paralelné výstupy,

SP – paralelné nastavenie,
 S – nastavenie (set),
 R – nulovanie (reset).