2. Logické funkce NON, AND, OR a XOR, pravdivostní tabulka, ÚNDF, ÚNKF, Booleova algebra, poloviční a úplná sčítačka, demultiplexor, porovnávací obvod.

HARDWARE A APLIKAČNÍ SOFTWARE

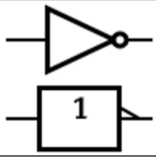
Logické funkce

NOT/NON

- Logická negace (invertor)
- Na výstupu je vždy opačná logická hodnota než na vstupu.
- Matematický zápis: Y = /A

Pravdivostní tabulka	
A	Y
0	1
1	0

Schematická značka

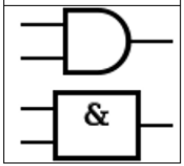


AND

- Logický součin
- Na výstupu log. 1 pouze tehdy, je-li na všech jeho vstupech log. 1.
- Matematický zápis: Y = A * B

Pravdivostní tabulka		
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Schematická značka

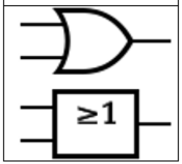


OR

- Logický součet
- Na výstupu log. 1 pouze tehdy, pokud je alespoň na jednom vstupu log. 1.
- Matematický zápis: Y = A + B

Pravdivostní tabulka		
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Schematická značka

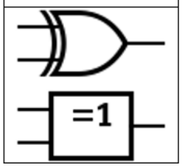


XOR

- Exklusivní logický součet
- Na výstupu log. 1 pouze tehdy, pokud je na vstupech rozdílná log. hodnota.
- Matematický zápis: Y = /(A B)

Pravdivostní tabulka		
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Schematická značka



Pravdivostní tabulka

- Pravdivostní tabulka je jeden ze způsobů zápisu logických funkcí.
- Taková tabulka obsahuje pouze logické proměnné, které nejčastěji nabývají dvou hodnot 0 a 1 (pravda/nepravda, ano/ne).
- Velikost tabulky je dána počtem proměnných a počtem výstupních funkcí.
- Máme-li n proměnných a m výstupních funkcí bude mít tabulka n + m sloupců.
- Řádků bude mít tabulka právě 2^n, což jsou všechny možné kombinace stavů logických proměnných, které mohou nastat.
- Pravdivostní tabulky se v praxi používají v elektronice při návrhu logických obvodů.

ÚNDF

- Úplná normální disjunktní forma
- Skládá se ze **součinů** vstupních proměnných v řádcích, kde má výstupní funkce hodnotu f = 1 tzv. mintermy.

- Všechny tyto mintermy pak sečteme.
- Každá proměnná v součinu je zapsána tak, že pokud nabývá hodnoty log 0, pak ji píšeme s negací, pokud log 1, pak píšeme bez negace.

ÚNKF

- Úplná normální konjunktní forma
- Skládá se ze **součtů** vstupních proměnných v řádcích, kde má výstupní funkce hodnotu f = 0 tzv. maxtermů.
- Všechny tyto maxtermy pak vynásobíme.
- Každá proměnná v součtu je zapsána tak, že pokud nabývá hodnoty log 0, pak ji píšeme bez negace, pokud log 1, pak píšeme s negací.

Booleova algebra

- Zákony Booleovy algebry slouží k zjednodušení a minimalizaci algebraického výrazu.

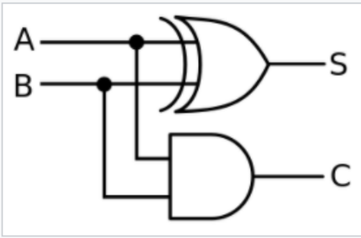
A	B	C
1. *Komutativní zákon* – KO $y = x_1 + x_0 = x_0 + x_1$ $y = x_1 \cdot x_0 = x_0 \cdot x_1$	1. *Zákon dvojité negace* – DN $y = \overline{\overline{x_0}} = x_0$	1. *Demorganovy zákony* – DM $y = \overline{x_1 \cdot x_0} = \overline{x_1} + \overline{x_0}$ $y = \overline{x_1 + x_0} = \overline{x_1} \cdot \overline{x_0}$
2. *Asociativní zákon* – AS $y = (x_2 + x_1) + x_0 = x_2 + (x_1 + x_0)$ $y = (x_2 \cdot x_1) \cdot x_0 = x_2 \cdot (x_1 \cdot x_0)$	2. *Zákon vyloučení třetího* – VT $y = x_0 + \overline{x_0} = 1$ $y = x_0 \cdot \overline{x_0} = 0$	2. *Zákon absorpce negace* – AN $y = x_1 \cdot (\overline{x_1} + x_0) = x_1 \cdot x_0$ $y = x_1 + \overline{x_1} \cdot x_0 = x_1 + x_0$
3. *Distributivní zákon* – DI $y = x_2 \cdot (x_1 + x_0) = x_2 \cdot x_1 + x_2 \cdot x_0)$	3. *Zákon neutrality nuly a jedničky* – NE $y = x_0 \cdot 1 = x_0$ $y = x_0 + 0 = x_0$	
	4. *Zákon agresivity nuly a jedničky* – AG $y = x_0 + 1 = 1$ $y = x_0 \cdot 0 = 0$	
	5. *Zákon absorpce* – AB $y = x_0 + x_0 = x_0$ $y = x_0 \cdot x_0 = x_0$	

Binární sčítačka

- Binární sčítačka (nebo jen sčítačka) je kombinační logický obvod, který realizuje sčítání čísel reprezentovaných ve dvojkové soustavě.
- Je důležitou součástí aritmeticko-logické jednotky (ALU aj.) procesorů (CPU, GPU, DSP, ...) počítačů.
- V případě použití dvojkového doplňku záporných čísel může být sčítačka velmi snadno rozšířena na sčítačku-odčítačku.

Poloviční sčítačka

- Anglicky Half adder
- Realizuje sčítání dvou jednobitových čísel.
- Vstup jsou dva jednobitové sčítance (A, B).
- Výstupem je jednobitový součet (S) a jednobitový příznak přenosu do vyššího řádu (C, nebo též anglicky jako Carry flag).
- Poloviční sčítačka dále přenáší příznak přenosu do vyššího řádu, sama však nedokáže zpracovat přenos z nižšího řádu. Nestačí proto k realizaci vícebitového sčítání.

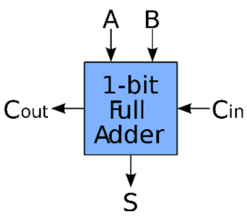


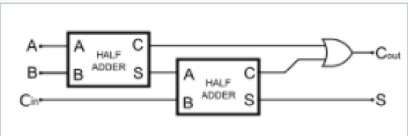
Poloviční sčítačka z hradel XOR a AND

vstup		výstup	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

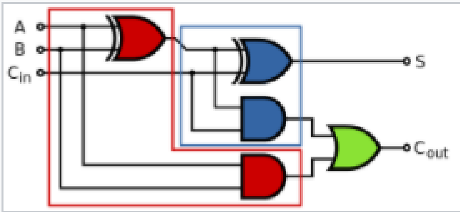
Úplná sčítačka

- Anglicky Full adder
- Realizuje sčítání dvou jednobitových čísel s přihlédnutím k přenosu z předchozího řádu.
- Vstupem jsou tři jednobitové sčítance: A, B, Ci (Ci z anglického Carry-in).
- Výstupem je jednobitový součet (S) a jednobitový příznak přenosu do vyššího řádu (Co z anglického Carry-out).
- Úplnou sčítačku je možné složit ze dvou polovičních sčítaček a hradla OR (viz obrázek).
- Hradlo OR je navíc možné bez vlivu na funkčnost nahradit pomocí hradla XOR, protože kombinace vstupů (1, 1), v němž by se jejich výstupy lišily, nemůže v případě sčítání nastat (buď nastane přenos pouze v první poloviční sčítačce, nebo pouze v druhé).
- Takže k vytvoření úplné sčítačky stačí mít 2 typy hradel, což může být praktické pro realizaci.
- Úplné sčítačky se spolu mohou vedle sebe řetězit (výstup Co jedné sčítačky propojit se vstupem Ci další) a provádět tak sčítání vícebitových čísel.





Úplná sčítačka, složená ze dvou sčítaček a hradla NEBO (blokové schéma)



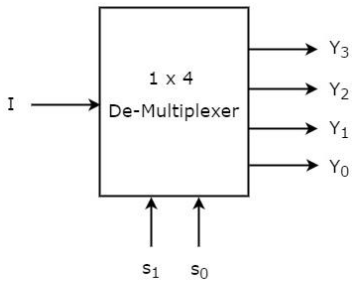
Úplná sčítačka z hradel (2x XOR, 2x AND, 1x NEBO). Dvě poloviční sčítačky jsou ve schématu označeny červeně a modře.

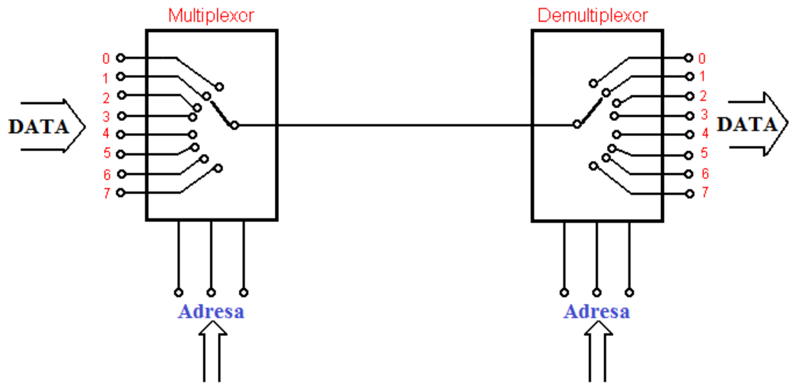
vstup			výstup	
A	B	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Demultiplexor

- Demultiplexor je kombinační logický obvod, která pracuje na opačném principu než multiplexor.
- Demultiplexor má 1 informační vstup (X), n adresních vstupů (A) a 2^n výstupů (y_0, y_1 atd).

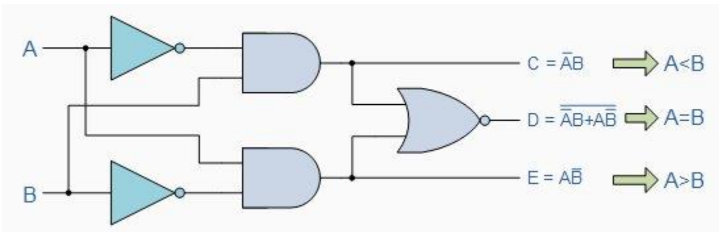
- Demultiplexor ze svého datového vstupu x_0 pomocí nastavené adresy rozesílá informace na příslušný výstup.





Porovnávací obvod

- Kombinační logický obvod, který porovnává dvě více bitová slova a na výstupech generuje signály pro rovnost, větší a menší.



i	B		A		y_2 A=B	y_1 A>B	y_0 A<B
	b_1	b_0	a_1	a_0			
0.	0	0	0	0	1	0	0
1.	0	0	0	1	0	1	0
2.	0	0	1	0	0	1	0
3.	0	0	1	1	0	1	0
4.	0	1	0	0	0	0	1
5.	0	1	0	1	1	0	0
6.	0	1	1	0	0	1	0
7.	0	1	1	1	0	1	0
8.	1	0	0	0	0	0	1
9.	1	0	0	1	0	0	1
10.	1	0	1	0	1	0	0
11.	1	0	1	1	0	1	0
12.	1	1	0	0	0	0	1
13.	1	1	0	1	0	0	1
14.	1	1	1	0	0	0	1
15.	1	1	1	1	1	0	0