

NXP CUP EMEA 2019 - 2020

JOIN MORE THAN 50 EUROPEAN
UNIVERSITIES TO CREATE YOUR
INTELLIGENT CAR.



FEI-minetors team



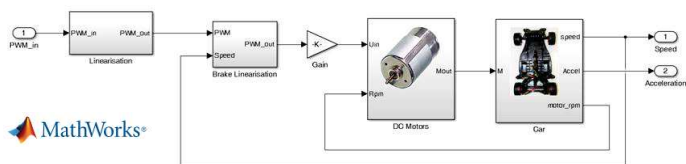
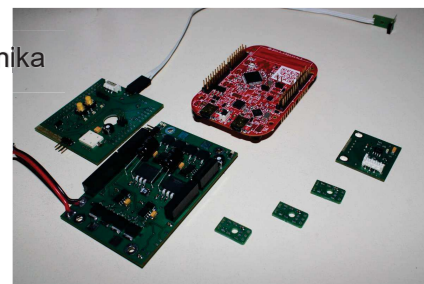
Freescall Cup FEI-minetors Riešenie elektroniky

Vlastný návrh

Freedom Kinetis K20 (ARM cortex M4 @ 50MHz)

Moduly:

- Výkonová elektronika
- Interface
- Interface kamery
- Odometria

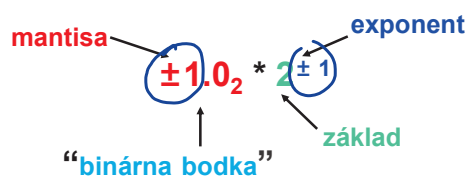


```

86 int main()
87 {
88     PC.baud(115200);
89     TFC_TickerObj.attach_us(&TFC_TickerUpdate, 2000);
90     TFC_Init();
91     for(;;) {
92         PC.printf("Select component to test:\r\n0: Servos\r\n1: Motors\r\n2: Camera (CCDI)\r\n");
93         switch(PC.getc()) {
94             default:
95                 case '0':
96                     PC.printf("Testing Servos. Restart the board to stop.\r\n");
97                     float val = -1; // -0.4 is all the way to the right. 0.4 is all the way to the left
98                     TFC_SetServo(0, val);
99                     while(1) {
100                         val += 0.5;
101                         if(val > 1) {
102                             val = -1;
103                         }
104                         TFC_SetServo(0, val);
105                         wait(1);
106                     }
107                 }
108             }
109     }

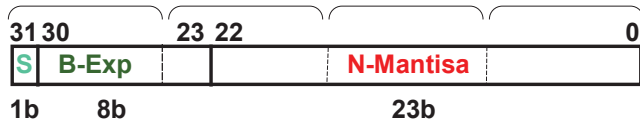
```

Zobrazenie s plávajúcou desatinnou čiarkou



Norma IEEE754 Single Precision

- Formát zápisu: $(+/-)1.xxxxxxxx_2 \cdot 2^{yyyy_2}$
- Počet bitov: 32



Mantisa: (priamy kód)

S - Sign znamienko mantisy

$|Mantisa| = 1.xxxxxxxx$, $xxxxxxx = N-Mantisa$

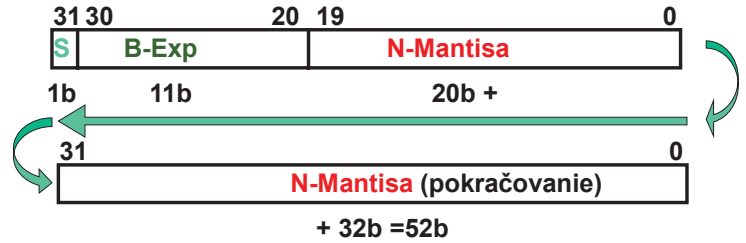
Exponent = B-Exp - Bias, Bias = 127, B-Exp = $\langle 1, 254 \rangle$

Čísla z rozsahu: $2^{-126}(1.0) \sim 2^{+127}(2 - 2^{-23})$

t.j. $1.18 \cdot 10^{-38} \sim 3.40 \cdot 10^{38}$

7

Norma IEEE754 Double precision



Jednoduchá presnosť

Bias: 127

Dvojnásobná presnosť

Bias: 1023

Čísla z rozsahu:

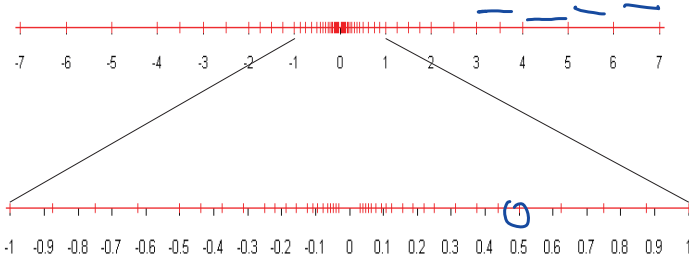
$2^{-126}(1.0) \sim 2^{+127}(2 - 2^{-23})$

$2.0 \cdot 10^{-308} \sim 2.0 \cdot 10^{308}$

Väčšou výhodou je vyššia presnosť

8

Domáca úloha



Viac o FP číslach

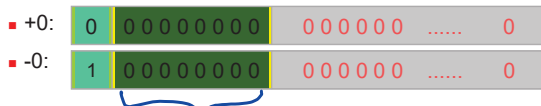
Doteraz sme uvažovali B-Exp v rozsahu $\langle 1 \text{ až } 254 \rangle$

Na čo je použitá „0“ a „255“?

10

Zobrazenie „0“

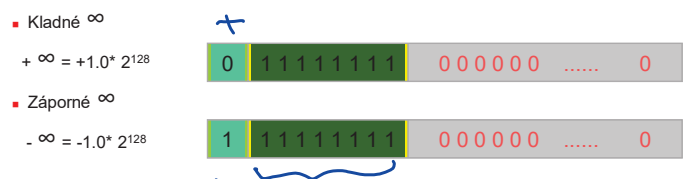
- Posunutý exponent, samé nuly: B-Exp = 0
- rovnako normovaná mantisa samé nuly
- A čo znamienko?



- Prečo dve nuly?
 - Výhodné pri limitných porovnávaniach

Znázornenie $\pm \infty$

- V FP aritmetike, delenie 0 dá $\pm \infty$, nie pretečenie.
- Prečo?
 - Ak existuje v FP aritmetike ∞ potom výraz $X/0 > Y$ je platné porovnanie
 - IEEE 754 vie zobrazit' $\pm \infty$
 - Najkľadnejší exponent B-Exp = 255 je rezervovaný pre ∞
 - N-Mantisa je nulová



11

12

Not a Number (NaN)

- Čo je $\sqrt{-4}$ alebo 0/0?
 - Ak ∞ nie je „chyba“, potom by nemuselo byť ani napr. 0/0.
 - Zaužíval sa názov: Not a Number (NaN)
 - B-Exp = 255, N-Mantisa nenulová
 - Načo je to dobré?
 - Dá sa predpokladať, že NaN sa využijú pri debugovaní

Napr.: $\text{op}(\text{NaN}, X) = \text{NaN}$

Zhrnutie

B-Exp	N-Mantisa	Výsledok (číslo)
0	0	0
0	nenulová ($\neq 0$)	Nenormované FP
1-254	nenulová	Normované FP
255	nulová	$\pm \infty$
255	nenulová ($\neq 0$)	NaN


```

#define FLT_MIN      1.17549435E-38F
#define FLT_MAX      3.40282347E+38F
#define FLT_EPSILON  1.19209290E-07F

#define DBL_MAX      1.7976931348623157E+308
#define DBL_MIN      2.2250738585072014E-308
#define DBL_EPSILON  2.2204460492503131E-016
    
```

14

15

Zaokrúhľovanie

- Výpočty s reálnymi číslami $\Rightarrow$ problém ako číslo umiestniť do odpovedajúceho priestoru.
- FP hardware obsahuje **2 špeciálne bity** pre presnosť (zníženie presnosti $\Rightarrow$ zvýšenie rýchlosti)
 - 00 – 24 bitov (SP)
 - 10 – 53 bitov (DP)
 - 11 – 64 bitov (Extended P (vnútorne FPU 80bitov))
- Zaokrúhľuje sa vždy pri konvertovaní...
 - DP $\Rightarrow$ SP
 - Číslo FP na integer

Zaokrúhľovanie

- Zaokrúhľovanie smerom $+\infty$
Vždy nahor : 2.1 $\Rightarrow$ 3, -2.1 $\Rightarrow$ -2
- Zaokrúhľovanie smerom $-\infty$
Vždy nadol : 1.9 $\Rightarrow$ 1, -1.9 $\Rightarrow$ -2
- Odrezanie
Jednoducho zahod' posledné bity (zaokrúhlenie smerom k 0)
- Zaokrúhlenie na najbližšie číslo (default),
 - vykonáme pripočítaním čísla 1 s váhou o 1 menej, ako je posledný platný rád.
 - resp. párne, ak sú dve najbližšie čísla rovnako vzdialené.
Pr.: 2,5 $\Rightarrow$ 2; 3,5 $\Rightarrow$ 4

16

17

Problémy FP aritmetiky

Presnosť $\longleftrightarrow$ Rozsah

Vedecké výpočty vyžadujú chybový manažment

Nespomenuli sme: Napr.: nie je garantované: $(1/r)*r \neq 1$

FPA aritmetika nie je asociatívna !!!!!

$$(A+B) + C \neq A + (B+C), (A*B) * C \neq A * (B*C)$$

nie je ani vždy distributívna !!!!!

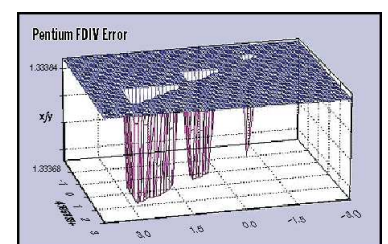
$$(A+B) * C \neq A*B + B*C$$

Implementovanie normy IEEE 754 je ťažké

Pentium bug

$$4\,195\,835 \div 3\,145\,727 = 1.333820449136241$$

1.333739068902037

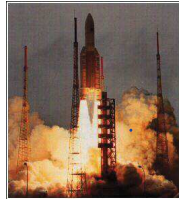


18

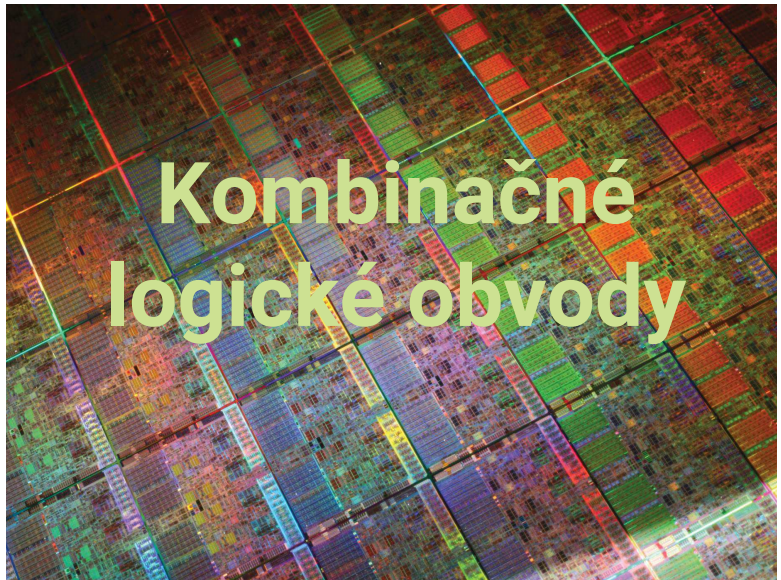
19

Problémy počítačovej aritmetiky

- Zlé implementovanie FPP spôsobuje chyby:
- **Raketa Patriot** – „vojna v zálive“:
Systém protivzdušnej obrany bol „zapnutý“ cca 100 hodín. Čas generovali ako načítavanie 0,1s zapísané binárne do 24 b.
Binárny ekvivalent 0,1s => chyba pri **zaokrúhlení** cca 10^{-7} s.
ALE: Za 100hodín chyba narástla na $100 * 3600 * 10^{-7} = 0,36$ s
ALE: pri rýchlosti 1676 m/s je to cca 600 m.
"Raketa minula priletavajúci iracký Scud", ktorý zasiahol tábor americkej armády.
- **Ariane 5** (v roku 1996) havarovala kvôli chybe pri **konverzii** čísla FP (64b) na signed integer (16b) ("nezместilo sa")
Predpokladaná cena nehody: 500 milión €



20



Všetci Kréťania sú klamári.

Epimenides z Knózosu (cca 600 p.n.l.)



→ Tit 1,12

Výroková logika

Jednoduchý výrok

Dunaj je ^PrieKa.

Zložený výrok

Dunaj je rieka ^{kvôli - P}alebo hotel.



Zdroj: Wikipédia

Výroková logika

modus
ponens

Prší. ^P

Ak prší, potom je cesta mokrá. ^P

Cesta je ^{moкра} ✓

p

$p \Rightarrow q$

q



Výroková logika

Keď poviem, že som uviedol zdroje,
tak potom som okopíroval, a neviem čo.



Základné logické princípy

Princíp dvojhodnotovosti (bivalencie):

Existujú práve dve pravdivostné hodnoty – pravda a nepravda.

Princíp vylúčenia tretieho (tertium non datur):

Pre každý výrok platí, že buď je pravdivý tento výrok, alebo opak tohto výroku.

Princíp vylúčenia sporu:

Pre každý výrok platí, že nie je súčasne pravdivý tento výrok aj opak tohto výroku

67 a. V Shakespearov Benátském kupci vystupuje dívka Porcie, a ta má tři skříňky — zlatou, stříbrnou a olovenou — a v jedné z nich je Porciina podobizna. Kdo se uchází o její ruku, musí určit, v které skříňce podobizna je, a pokud má štěstí (nebo je tak chytrý) a uhodne, smí se s ní oženit. Na víku každé skříňky je nápis, který má nápadníkovi při volbě pomoci.

Dejme tomu, že by si Porcie chtěla vybrat manžela ne podle toho, jak je ctnostný, ale jen podle toho, jak je inteligentní. Dala na skříňky nápisy:

DŮ

Zlatá	Stříbrná	Olovená
PODOBIZNA JE V TĚTO SKŘÍŇCE	PODOBIZNA NENÍ V TĚTO SKŘÍŇCE	PODOBIZNA NENÍ VE ZLATÉ SKŘÍŇCE

Nápadníkovi prozradila, že z těch tří nápisů je nanejvýš jeden pravdivý. Kterou skříňku měl nápadník vybrat?

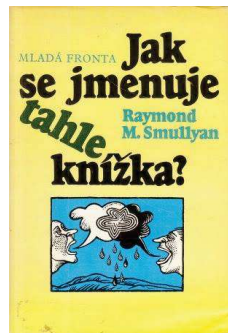
Chcem vedieť viac...

Raymon M SMULLYAN:

Jak se jmenuje tahle knížka?

What is the Name of this Book? (1981)

Mladá fronta : Praha, 1986.



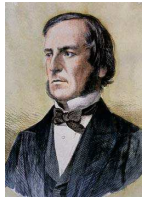
George Boole

* 2. novembra 1815, Lincoln, Anglicko
† 8. decembra 1864, Ballintemple, Írsko

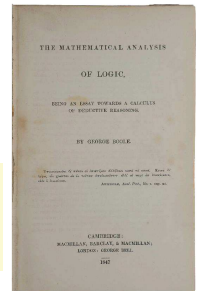
— profesor Queen's College v Írsku

kvaternióny, matematická analýza, logika

Systém na ohodnotenie pravdivostných hodnôt výrazov zložených z logických spojok AND, OR, NOT a logických premenných nadobúdajúcich iba dve hodnoty – 1 (PRAVDA) a 0 (NEPRAVDA).



The class X and the class not- X together make the Universe. But the Universe is 1, and the class X is determined by the symbol x , therefore the class not- X will be determined by the symbol $1 - x$.



Zdroj: Wikipédia

Boolova algebra

$B: X \subseteq \{0, 1\}$

Negácia (komplement) je unárna operácia na množine B , ktorú označujeme $\neg$ a definujeme:

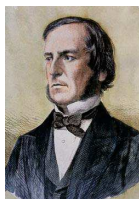
$$\neg 0 = 1 \text{ a } \neg 1 = 0$$

Logický súčet je binárna operácia na množine B , ktorú označujeme $+$ a definujeme takto:

$$0+0=0, \quad 0+1=1, \quad 1+0=1, \quad \boxed{1+1=1}$$

Logický súčin je binárna operácia na množine B , ktorú označujeme $\cdot$ a definujeme takto:

$$0 \cdot 0=0, \quad 0 \cdot 1=0, \quad 1 \cdot 0=0, \quad 1 \cdot 1=1$$



Pre každé $x, y, z \in B$ platí (pozor na preklepy 😊)

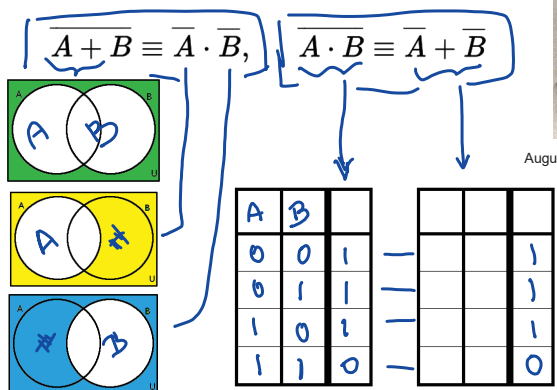
$x + y = y + x$	$x \cdot y = y \cdot x$	komutatívne zákony
$(x + y) + z = x + (y + z)$	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	asociatívne zákony
$(x + y) \cdot z = (x \cdot z) + (y \cdot z)$	$(x \cdot y) + z = (x + z) \cdot (y + z)$	distributívne zákony
$\boxed{x + y = \bar{x} \cdot \bar{y}}$	$\boxed{\bar{x} \cdot \bar{y} = \bar{x + y}}$	de Morganove zákony
$\begin{matrix} x + x = x \\ 0 + 0 = 0 \end{matrix}$	$x \cdot x = x$	zákony idempotentnosti
$x + \bar{x} = 1$	$x \cdot \bar{x} = 0$	zákony komplementárnosti
$\bar{\bar{x}} = x$		zákon involúcie
$x + (x \cdot y) = x$	$x \cdot (x + y) = x$	zákony absorpcie
$x + 0 = x$	$x \cdot 1 = x$	zákony identity
$x + 1 = 1$	$x \cdot 0 = 0$	
Zákon jednotkového sčítavania	Zákon nulového násobenia	

Zdroj: Wikipédia

De Morgan



Augustus De Morgan (1806-1871)



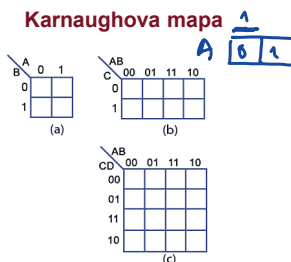
Zdroj: Wikipédia

Negácia

Tabuľka

A	Y
0	1
1	0

Karnaughova mapa



Rovnica

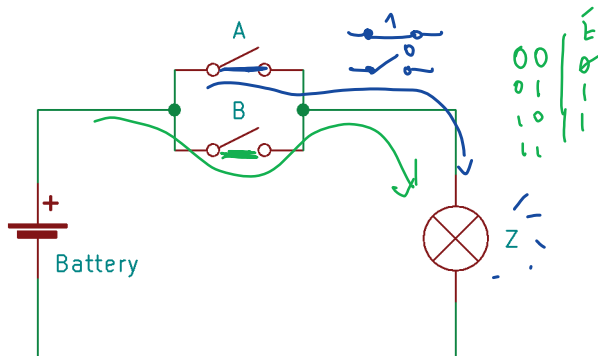
$$Y = \overline{A}$$

Schematická značka



Logický súčet

ALÉBO OR



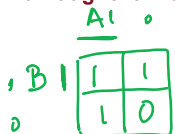
Logický súčet

OR

Tabuľka

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

Karnaughova mapa



Rovnica

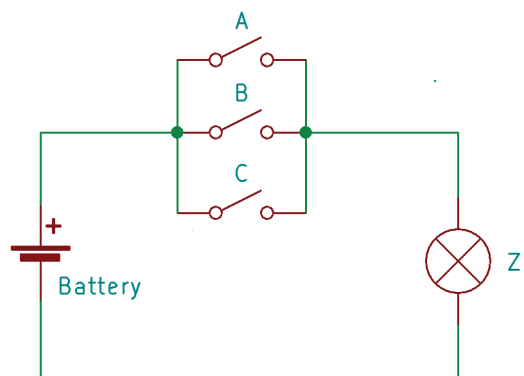
$$Y = A + B$$

Schematická značka



Logický súčet viac premenných

OR



Logický súčet viac premenných

OR

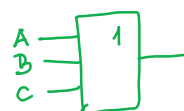
Tabuľka

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

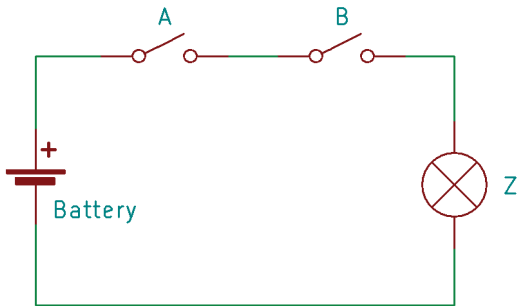
Rovnica

$$Y = A + B + C$$

Schematická značka



Logický súčin ^A AND



Logický súčin AND

Tabuľka

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

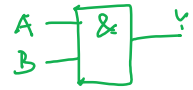
Rovnica

$$Y = A \cdot B$$

Karnaughova mapa

	<u>A</u>	0	1
B	0	0	0
1	0	0	1

Schematická značka



Logický súčin viac premenných AND

Exclusive OR

XOR

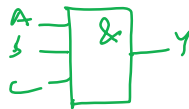
Tabuľka

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

Rovnica

$$Y = A \cdot B \cdot C$$

Schematická značka



Tabuľka

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

Rovnica

$$Y = A \oplus B$$

Karnaughova mapa

	<u>A</u>	0	1
B	0	0	1
1	0	1	0

Schematická značka



Logický súčet ^{NEGOVANÝ} NOR

NOR

Logický súčin ^{NEG} NAND

NAND

Tabuľka

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

Rovnica

$$Y = \overline{A + B}$$

Schematická značka



Tabuľka

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

Rovnica

$$Y = \overline{A \cdot B}$$

Schematická značka



NAND – úplný logický systém

NOT $Y = \overline{A \cdot B} = \overline{A \cdot B} = \overline{A}$



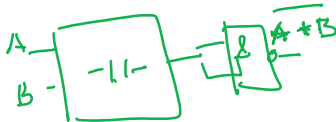
AND $Y = A \cdot B$



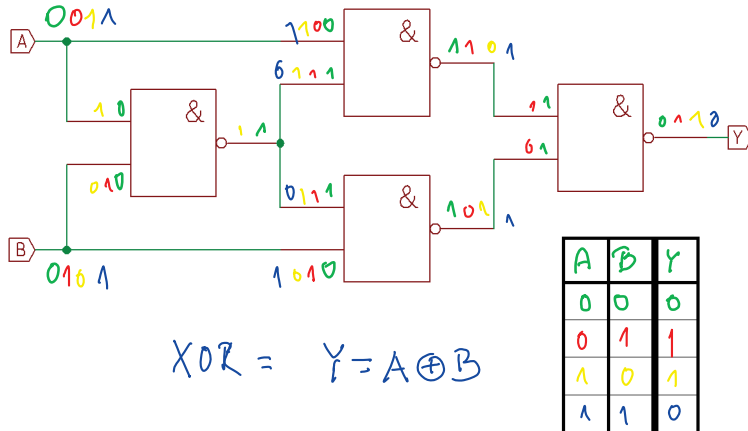
OR $A + B = \overline{\overline{A} \cdot \overline{B}}$



NOR $Y = \overline{A + B}$



NAND – úplný logický systém

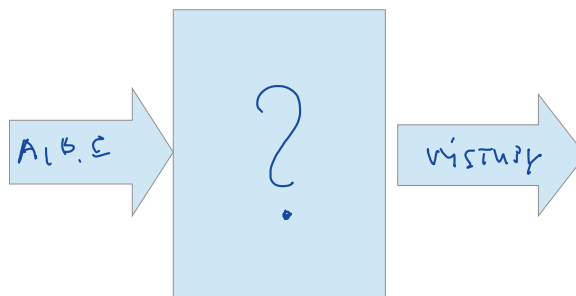


Koľko je logických funkcií?

A	B	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆
0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	1
1	0	0	0	1	0	0	1
1	1	0	0	1	1	1	0

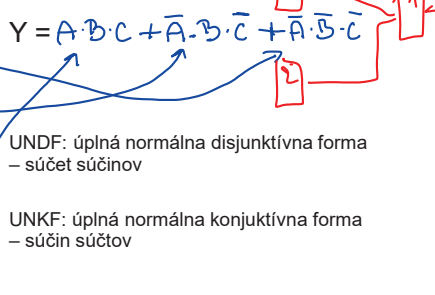
$Y_1 = 0$
 $Y_2 = 1$
 $Y_3 = A \cdot B$
 $Y_4 = B$

Syntéza logických obvodov



Syntéza logických obvodov

	A	B	C	Y
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1



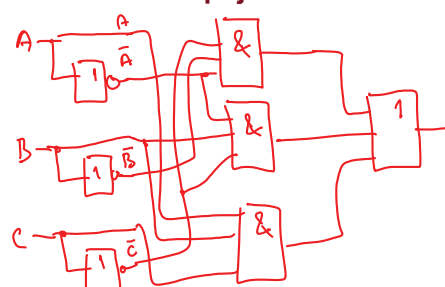
$Y = (A + B + \overline{C}) \cdot (\overline{A} + \overline{B} + C) \cdot (\overline{A} + B + \overline{C}) \cdot (\overline{A} + \overline{B} + C)$

Syntéza logických obvodov

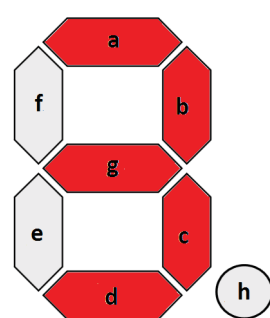
	A	B	C	Y
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

$Y = \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C} + A \cdot B \cdot C$

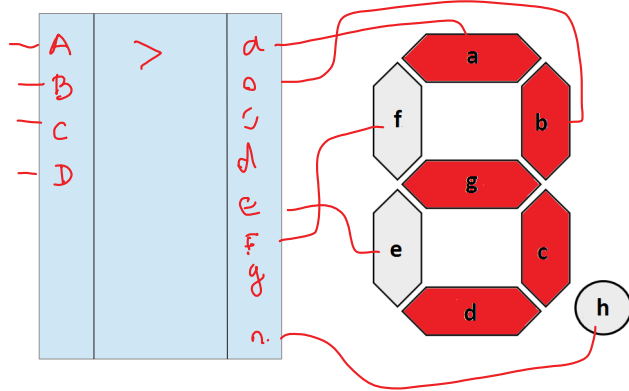
Schéma zapojenia:



7-segmentový dekódér

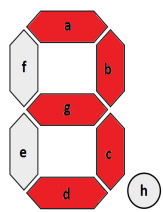


7-segmentový dekódér



7-segmentový dekódér

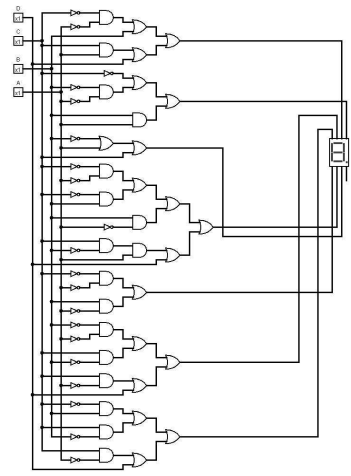
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0							
1	0	0	0	1							
2	0	0	1	0							
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1							
8	1	0	0	0							
9	1	0	0	1							



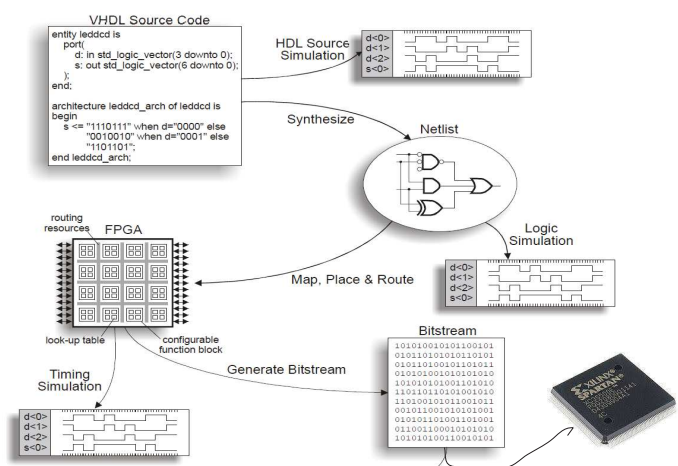
$a = A \cdot \bar{B} + C \cdot \bar{D}$

7-segmentový dekódér

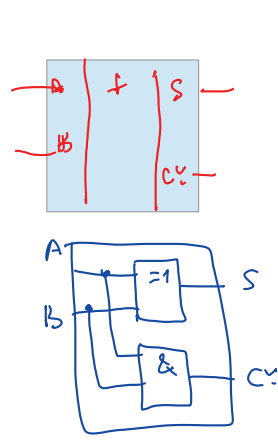
Binary Inputs	Decoder Outputs	7 Segment Display Outputs
DCBA	a b c d e f g	
0000	1 1 1 1 1 1 0	0
0001	0 1 1 0 0 0 0	1
0010	1 1 0 1 1 0 1	2
0011	1 1 1 1 0 0 1	3
0100	0 1 1 0 0 1 1	4
0101	1 0 1 1 0 1 1	5
0110	1 0 1 1 1 1 1	6
0111	1 1 1 0 0 0 0	7
1000	1 1 1 1 1 1 1	8
1001	1 1 1 1 0 1 1	9



Syntéza FPGA (Xilinx)



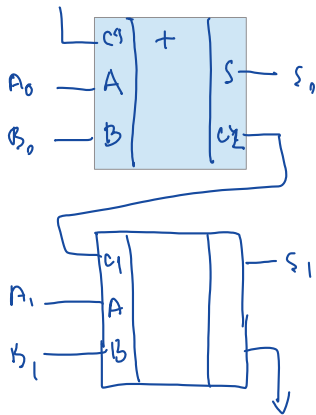
Sčítačka



A	B	S	CY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

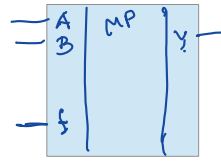
$S = A \oplus B$
 $CY = A \cdot B$

Úplná sčítačka



C_1	A	B	S	C_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

Programovateľný AND/OR člen



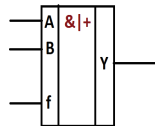
f	A	B	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$f = \begin{cases} 0: Y = A + B \\ 1: Y = A \cdot B \end{cases}$$

$$Y = A \cdot B \cdot C + A \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C$$

Programovateľný AND/OR člen

Schéma zapojenia:



f	A	B	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

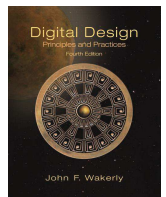
$$Y = f \cdot (A + B) + \bar{f} \cdot A \cdot B$$

Chcem vedieť viac...

John F. WAKERLY:

Digital design.

Stanford University : Prentice Hall, 2006.



Jean-Michel BERNARD, Jean HUGON

a Robert Le Corvec:

Od logických obvodů k mikroprocesorům.

SNTL : Praha, 1982

