Základy číslicovej techniky.

Vývojom elektroniky sa ľudstvo dopracovalo k zdokonaleniu elektronických zariadení, v ktorých sa čoraz viac a v dnešnej dobe väčšinou používajú súčiastky pracujúce na princípe číslicových metód. **Číslicové obvody** sú obvody, v ktorých sa signál v čase mení nespojito (skokom). Pod pojmom **logický systém** rozumieme systém, pri ktorom každá vstupná a výstupná veličina sa nemôže meniť spojito, ale môže nadobúdať len dva stavy, hodnoty označované 0 a 1.

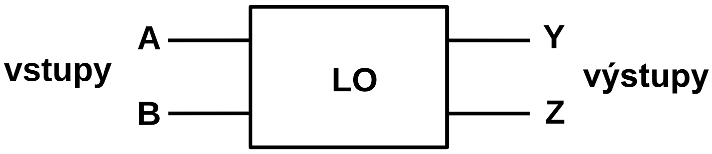
Dôvody pre používanie číslicovej techniky:

- použitím číslicových integrovaných obvodov sa zmenšila hmotnosť, veľkosť a cena obvodov,
- IO (integrované obvody) nahradili zložité obvody (v niektorých prípadoch aj celé prístroje),
- zväčšila sa presnosť a výkonnosť v prezentácií výsledkov aj pri samotnom vykonávaní merania,
- zväčšil sa dynamický rozsah,
- zväčšila sa stabilita,
- pohodlnosť (priame desiatkové odčítanie dát vylučuje nesprávne odčítanie meraných hodnôt),
- automatizácia (pomocou IO alebo PC – programovanie operácií),
- jednoduchá konštrukcia,
- nové prístupy pri riešení konštrukčných problémov elektronických zariadení.

Boolova algebra

Na opis správania logických systémov používame matematický model pre prácu s dvojhodnotovými (binárnymi) veličinami. **Boolová algebra** je dvojhodnotová logická algebra, ktorá používa disjunkciu (logický súčet), konjunkciu (logický súčin) a negáciu (logická negácia) ako úplný súbor základných logických funkcií a slúži na matematický opis zákonov a pravidiel výrokovej logiky, ktoré riešia vzťahy medzi pravdivými a nepravdivými výrokmi:

- pravdivý výrok - priradená hodnota logická 1 (TRUE alebo PRAVDA)
- nepravdivý výrok - priradená hodnota logická 0 (FALSE alebo NEPRAVDA).



Logická funkcia F je zobrazenie z množiny výrokov do množiny pravdivostných hodnôt {0,1}, ktoré výroku priraďuje pravdivostnú hodnotu, pričom ak je výrok V

- pravdivý: $F(V)$ = LOG1 t.j. $Y = 1$,
- nepravdivý: $F(V)$ = LOG0 t.j. $Y = 0$.

Logická funkcia môže byť zadaná rôznymi spôsobmi – slovne, pravdivostnou tabuľkou, v tvare logického výrazu alebo graficky pomocou máp a diagramov.

Pravdivostná tabuľka je tabuľka, v ktorej sú uvedené všetky možné kombinácie hodnôt nezávislých premenných a im odpovedajúce hodnoty závislých premenných. Táto tabuľka má toľko stĺpcov, koľko je uvažovaných logických premenných a toľko riadkov, koľko existuje kombinácií pravdivostných hodnôt nezávislých premenných.

Pre súbor *n* logických premenných *Xi*, i = 1, 2, ... *n* existuje:

- 2^n rôznych vstupných kombinácií týchto premenných,
- 2^{2n} rôznych logických funkcií *Yj* (pretože každá z nezávislých logických premenných môže nadobúdať dve hodnoty: 0, 1), teda *j* = 1, 2, ... 2^{2n}.

S rastúcim počtom logických premenných (n) teda počet navzájom rôznych logických funkcií, ktoré možno týmto premenným priradiť prudko narastá. Je však dokázané, že akúkoľvek logickú funkciu *Y* konečného počtu logických premenných je možné vyjadriť súborom vhodne vybraných logických funkcií *Yi* jednej alebo dvoch logických premenných. Tento súbor logických funkcií nazývame **úplný súbor logických funkcií**. Na každom úplnom súbore logických funkcií možno vybudovať logickú algebru.

Z množstva logických funkcií dvoch logických premenných majú **zásadný význam** funkcie *Y*1 ¸ *Y*16, ktorých hodnoty pre jednotlivé vstupy vyjadruje pravdivostná tabuľka:

A	**B**	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}	Y_{16}
0	**0**	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	**1**	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	**0**	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	**1**	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

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

Y_{16}	konštanta verum	je pravda	1

V Boolovej algebre sú definované **3 základné operácie**, pomocou ktorých môžeme vyjadriť ľubovoľnú logickú operáciu:

- logický súčet – **disjunkcia,**
- logický súčin – **konjunkcia,**
- logická negácia – **negácia.**

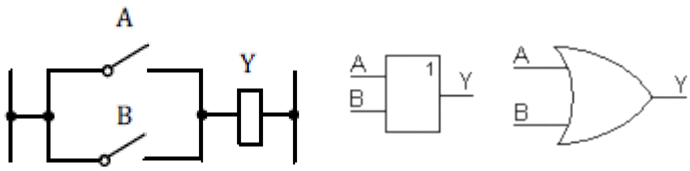
Hodnoty premennej Y, závislej od jednotlivých kombinácií nezávisle premenných A**,** B pre základné logické operácie znázorníme v **pravdivostnej tabuľke**. V ľavej časti tabuľky sú zapísané všetky kombinácie nezávislé premenných. V pravej časti tabuľky sú zapísané všetky stavy funkčných hodnôt výstupnej premennej Y.

Logický súčet je operácia disjunkcie alebo zjednotenia a označuje sa znakom ∨. V praxi sa často používa algebraický zápis:

$$\boldsymbol{Y = A + B \ alebo \ Y = A \vee B}$$

Pre operáciu logického súčtu je charakteristická spojka „alebo“ (anglicky „or“). Logický člen, ktorý realizuje logický súčet, sa nazýva **člen OR**. Logický súčet sa **rovná jednotke**, keď **aspoň jedna** nezávisle premenná má hodnotu 1.

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



Pre logický súčet platí:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 s prenosom 1 do vyššieho rádu

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

PRÍKLAD:

0 1 1 0 0 1 1	1 + 1 = 2 → 10 => 0 zapíšeme,
<u>0 0 1 1 0 1 1</u>	1 prechádza do vyššieho rádu
1 0 0 1 1 1 0	1 + 1 + 1 = 3 → 11 => 1 zapíšeme,

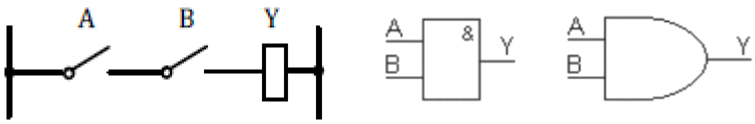
1 prechádza do vyššieho rádu,
1 + 0 + 0 = 1 => 1 zapíšeme,
1 + 0 = 1 => 1 zapíšeme,
1 + 1 = 2 → 10 => 0 zapíšeme,
1 prechádza do vyššieho rádu
1 + 1 = 2 → 10 => 0 zapíšeme,
1 prechádza do vyššieho rádu.

Logický súčin je operácia konjunkcie alebo prieniku, označuje sa znakom ∧. V praxi sa často používa algebraický zápis:

$$\boldsymbol{Y = A.B \; alebo \; Y = A \wedge B}$$

Pre operáciu logického súčinu je charakteristická spojka „aj“, „i“(anglicky „and“, „&”). Logický člen, ktorý realizuje logický súčin, sa nazýva **člen AND**. Logický súčin sa **rovná jednotke len vtedy**, keď **všetky** nezávisle premenné majú hodnotu 1. Ak aspoň jedna vstupná premenná má hodnotu 0, výsledná hodnota sa rovná 0.

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



Pre logický súčin platí:

0 . 0 = 0
0 . 1 = 0
1 . 0 = 0
1 . 1 = 1

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:

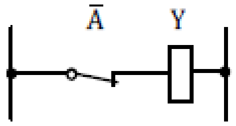
```
    1 1 0 1 0
       . 1 0 1
    1 1 0 1 0
  0 0 0 0 0
1 1 0 1 0
1 0 0 0 0 0 1 0
```

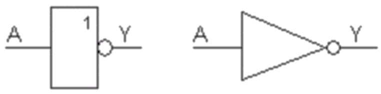
Logická negácia je operácia negácie, záporu. V praxi sa často používa algebraický zápis:

$$\boldsymbol{Y = \overline{A}}$$

Pre operáciu logickej negácie je charakteristický zápis „nie“, („not“). Logický člen, ktorý realizuje logickú negáciu, sa nazýva **člen NOT, invertor** alebo **negátor**. Logická negácia mení hodnotu nezávislej premennej na opačnú. Jej výsledok je rovný 1**,** ak A = 0 a opačne.

A	Y
0	1
1	0





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 *Xi* 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 *Xi* 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 $Y = A \Rightarrow (B \Rightarrow C)$, ktorej pravdivostná tabuľka je:

Technické prostriedky automatizovaného riadenia
Prednáška 2

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

Napíšte:

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

Riešenie:

1. **Súčtová (disjunktívna) normálová forma** tejto funkcie je

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

2. Súčinová (konjunktívna) normálová forma tejto funkcie je

$$Y = (\bar{A} + B + \bar{C}).(\bar{A} + \bar{B} + C)$$

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.

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

absorpcie	$A + (A.B) = A$	$A.(A + B) = A$
absorpcie negácie	$A + \bar{A}.B = A + B$	$A.(\bar{A} + B) = A.B$
vylúčenia	$A + \bar{A} = 1$	$A.\bar{A} = 0$
idempotencie	$A + A = A$	$A.A = A$
neutrálnosti 0 a 1	$A + 0 = A$	$A.1 = A$
agresívnosti 0 a 1	$A + 1 = 1$	$A.0 = 0$
dvojitej negácie A	$\bar{\bar{A}} = A$	$\bar{\bar{A}} = A$

Zápis logickej funkcie do pravdivostnej tabuľky

Logickú funkciu zapisujeme do pravdivostnej tabuľky tak, že hodnoty vstupných veličín (nezávisle premenných) prostredníctvom algebraického výrazu reprezentujú výstupnú veličinu (závisle premennú).

PRÍKLAD:

Vytvorte pravdivostnú tabuľku z daných logických funkcií:

$$Y1 = (A + B + C).(A + \bar{B} + C).(\bar{A} + B + \bar{C})$$
$$Y2 = (\bar{A}.\bar{B}.C) + (A.B.\bar{C}) + (A.\bar{B}.\bar{C})$$

Obidve funkcie môžeme pre zjednodušenie zapísať do jednej tabuľky. Vytvoríme teda tabuľku so šiestimi stĺpcami (stavové indexy, premenné A, B, C a výstupné hodnoty Y1, Y2) a ôsmimi riadkami (2^n). Do stĺpcov pre premenné A, B, C vpíšeme všetky kombinácie možných vstupných hodnôt (0 a 1). Do stĺpcov pre výstupy Y1 a Y2 vpíšeme výstupnú hodnotu danej funkcie pri daných hodnotách nezávisle premenných pre každý riadok tabuľky (nezávisle premenné dosadíme do funkcie a pomocou základných operácií Boolovej algebry určíme výslednú hodnotu).

N	A	B	C	Y1	Y2
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	0	0
3	0	1	1	1	0
4	1	0	0	1	1
5	1	0	1	0	0
6	1	1	0	1	1
7	1	1	1	1	0

Zo zadanej pravdivostnej tabuľky sa najčastejšie vypisuje funkcia ako logický súčet základných logických súčinov.

Základný logický súčin (minterm) je súčin všetkých nezávisle premenných, pričom vstupné premenné, ktoré majú stav 0, sa vyjadrujú negované. Vyberáme len tie súčiny z tých riadkov, kde hodnota funkcie nadobúda stav 1.

Niekedy môže byť výhodnejšie (napríklad ak je výstupná hodnota funkcie vo väčšine riadkov tabuľky rovná 0 zapísať funkciu v tvare logického súčinu základných logických súčtov.

Základný logický súčet (maxterm) je súčet všetkých nezávisle premenných, pričom vstupné premenné, ktoré majú stav 1, sa vyjadrujú v negovanom stave. Vyberáme len tie súčty z tých riadkov, kde hodnota funkcie nadobúda stav 0.

PRÍKLAD:

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

Pre súčet základných logických súčinov z tabuľky vypíšeme kombinácie vstupných premenných z tých riadkov, kde je hodnota funkcie logická 1.

riadok č. 2: A=0, B=1, C=0 – základný logický súčin $\bar{A}.B.\bar{C}$
riadok č. 3: A=0, B=1, C=1 – základný logický súčin $\bar{A}.B.C$
riadok č. 6: A=1, B=1, C=0 – základný logický súčin $A.B.\bar{C}$
riadok č. 7: A=1, B=1, C=1 – základný logický súčin $A.B.C$

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

Pre súčin základných logických súčtov vypíšeme z tabuľky tie riadky, v ktorých má funkcia hodnotu 0:

riadok č. 0: A=0, B=0, C=0 – základný logický súčet $A + B + C$
riadok č. 1: A=0, B=0, C=1 – základný logický súčet $A + B + \bar{C}$
riadok č. 4: A=1, B=0, C=0 – základný logický súčet $\bar{A} + B + C$
riadok č. 5: A=1, B=0, C=1– základný logický súčet $\bar{A} + B + \bar{C}$

$$Y = (A + B + C).(A + B + \bar{C}).(\bar{A} + B + C).(\bar{A} + B + \bar{C})$$

Zápis logickej funkcie do Karnaughovej mapy

Karnaughova mapa je metóda používaná na minimalizáciu logickej funkcie. Jej princípom je zobrazenie n-rozmernej tabuľky hodnôt do dvojrozmernej mapy. Z tejto mapy možno potom graficky vyčítať minimálnu funkciu. Každej kombinácií nezávisle premenných je v Karnaughovej mape pridelený jeden štvorec, do ktorého sa vpisuje stav výstupnej funkcie. Teda počet štvorcov v Karnaughovej mape sa musí rovnať počtu riadkov v pravdivostnej tabuľke a naopak, každému riadku pravdivostnej tabuľky zodpovedá jeden štvorec v Karnaughovej mape.

Priradenie vstupných premenných pre jednotlivé riadky a stĺpce sa vykonáva:

- **algebraickým označením** - po obvode Karnaughovej mapy sa napíšu hodnoty nezávisle premenných, ktoré prislúchajú danému riadku/stĺpcu. Tento spôsob sa používa len zriedkavo.
- **svorkami** - po okrajoch Karnaughovej mapy sa vyznačia čiarami označenými príslušnou nezávisle premennou tie riadky alebo stĺpce, pre ktoré má daná nezávisle premenná hodnotu 1.

Postup pri minimalizácií z Karnaughovej mapy pre NDF (NKF) je nasledovný:

1) Jednotky (nuly) v mape uzatvárame pomocou slučiek, ktoré obsahujú 2^n (teda jedno, dve, štyri, osem, atď.) susedných políčok. Pritom si treba zapamätať, že čím väčšie slučky vytvoríme, tým jednoduchšie budú zodpovedajúce algebraické výrazy.
2) Karnaughovu mapu si treba pri vytváraní slučiek predstaviť napr. ako guľu, pretože do jednej slučky môžu patriť aj políčka z náprotivných strán Karnaughovej mapy. Do jednej slučky môžu patriť aj políčka v rohoch Karnaughovej mapy.
3) Každá jednotka (nula) musí byť zahrnutá aspoň v jednej slučke.
4) Slučky sa môžu navzájom pretínať.
5) Každej slučke zodpovedá iba jeden výraz, ktorý neobsahuje tie premenné, pre ktoré daná slučka nadobúda hodnotu 0 **aj** 1.
6) Slučky s jednotkami zodpovedajú výslednému výrazu v podobe súčtu súčinov, zatiaľ čo slučky s nulami výrazu v podobe súčinu súčtov.

PRÍKLAD:

Algebraická minimalizácia logických funkcií predstavuje algebraické upravovanie logického výrazu podľa zákonov a pravidiel Boolovej algebry. Výsledkom je funkcia s rovnakým správaním, vyjadrená však jednoduchšie, čo pri realizácii znamená jednoduchšiu štruktúru logického obvodu.

Funkciu $Y = A.B + A.\bar{B} + \bar{A}.\bar{B}$ je možné upraviť nasledovne:
Z prvého a druhého súčinu môžeme pred zátvorku vybrať A: $Y = A(B + \bar{B}) + \bar{A}.\bar{B}$
Na výraz $(B + \bar{B})$ môžeme použiť zákon vylúčenia: $Y = A + \bar{A}.\bar{B}$
Nakoniec použijeme zákon absorpcie negácie a dostaneme výsledný výraz: $Y = A + \bar{B}$

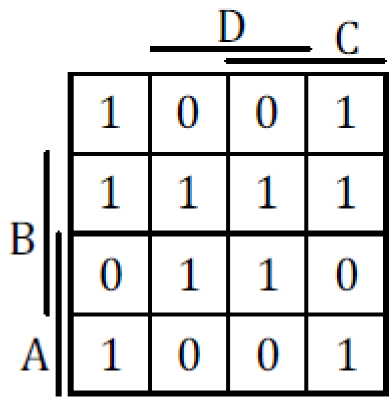
Ponúka aj možnosť minimalizácie pomocou Karnaughovej mapy. Pretože dve jednotky spolu susedia, môžeme pristúpiť okamžite k napísaniu jedného výrazu pre obe políčka a tak dostaneme priamo minimalizovaný výraz. Uvedený postup môžeme rozšíriť na väčší počet susedných políčok.

		B
	1	
A	1	1

Ak robíme minimalizáciu funkcie pomocou jednotiek, hovoríme o **Normálnej disjunktívnej forme – NDF**, alebo o **Normálnej konjunktívnej forme – NKF**, ak minimalizujeme z núl.

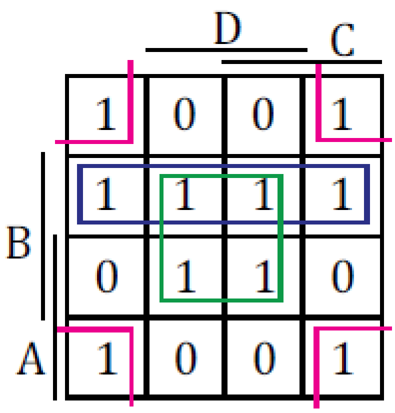
PRÍKLAD

Minimalizujte funkciu danú Karnaughovou mapou:



Riešenie pomocou NDF:

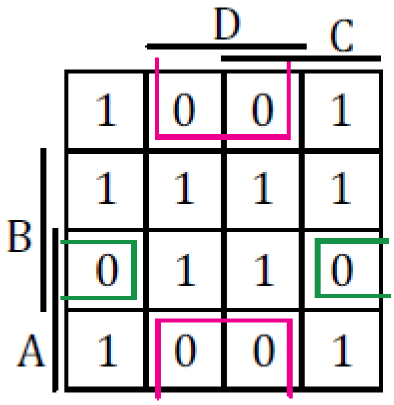
Na mape vytvoríme slučky obsahujúce jednotky. Slučky robíme čo najväčšie pre zjednodušenie výrazu. Vypíšeme jednotlivé slučky ako súčet základných súčinov :



$$Y = B.D + \bar{B}.\bar{D} + \bar{A}.B$$

Riešenie pomocou NKF:

Vytvoríme slučky z núl. Opäť sa snažíme vytvoriť čo najväčšie slučky. Vypíšeme jednotlivé slučky ako súčin základných súčtov, pričom nezabúdame, že výstupy sú negované:



$$Y = (B + \overline{D}).(\bar{A} + \bar{B} + D)$$