# Způsoby vyjádření logických funkcí

## 1. Zjednodušení (minimalizace) logické funkce

- **Jak minimalizovat logické funkce:**
  - Zapsat funkci v algebraickém tvaru a použít pravidla Booleovy algebry.
  - Použít Karnaughovu mapu, která umožňuje minimalizaci pomocí grafické metody.

## 2. Nakreslení schématu logického obvodu

- **Jak vytvořit obvod:**
  - Použít základní logické členy jako AND, OR, NOT nebo jiné, například EXOR.
  - Použít pouze NAND nebo NOR členy. To vyžaduje převod funkce podle De Morganových zákonů.

## Jak zapsat logickou funkci v algebraickém tvaru

1. **Úplná normální disjunktní forma (UNDF):**
   - Vytvoříme součiny vstupních proměnných (mintermy) z řádků pravdivostní tabulky, kde je výstup 1.
   - Tyto mintermy sečteme. Proměnné se zapisují takto:
     - Hodnota 0: s negací.
     - Hodnota 1: bez negace.
2. **Úplná normální konjunktní forma (UNKF):**
   - Vytvoříme součty vstupních proměnných (maxtermy) z řádků pravdivostní tabulky, kde je výstup 0.
   - Tyto maxtermy vynásobíme. Proměnné se zapisují takto:
     - Hodnota 0: bez negace.
     - Hodnota 1: s negací.

**Příklad:**

| *s* | *b* | *a* | *f* | *slovní popis* | *minterm* | *maxterm* |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | *žádný větrák není rozbitý* | - | $a+b$ |
| 1 | 0 | 1 | 1 | *první větrák se rozbil* | $a\bar{b}$ | - |
| 2 | 1 | 0 | 1 | *druhý větrák se rozbil* | $\bar{a}b$ | - |
| 3 | 1 | 1 | 1 | *oba větráky se rozbily* | $ab$ | - |

ÚNDF: $f = a\bar{b} + \bar{a}b + ab$

ÚNKF: $f = a + b$

Samozřejmě teď bychom mohli nakreslit schéma obvodu, ale vždy je dobré výstupní funkci zkusit zjednodušit. Již dříve jsme využívali pravidla Booleovy algebry, takže jen zopakujeme

## Minimalizace pomocí Booleovy algebry

Pomocí pravidel Booleovy algebry lze zjednodušit funkci:

- Příklad zjednodušení:

$$f = a\bar{b} + \bar{a}b + ab = a(\bar{b} + b) + \bar{a}b = a + \bar{a}b = a + b$$

Jak vidíme, zjednodušená funkce je stejná jako ÚNKF (Úplná Normální Konjunktní Forma). To není překvapivé, protože obě formy vyjadřují stejnou logickou funkci. Co je zajímavé, je fakt, že ÚNKF byla už od začátku minimální, protože obsahovala pouze jeden řádek (tj. jedinou nulovou hodnotu funkce).

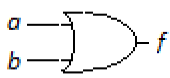
### Princip minimalizace:

- Minimalizace spočívá ve vhodném slučování:
  - **Jedniček** v případě ÚNDF (Úplná Normální Disjunktní Forma).
  - **Nul** v případě ÚNKF.
- Minimalizace se většinou provádí pro ÚNDF, protože práce s jedničkami bývá jednodušší. ÚNKF se používá pouze tehdy, pokud je nul výrazně méně než jedniček.

---

## Schéma výstupní funkce ze základních logických členů

Nyní můžeme nakreslit schéma obvodu pomocí základních logických členů. Tento příklad je velmi jednoduchý, takže stačí použít jeden logický člen **OR**




Raději uvedu ještě jeden příklad, který je zadán pravdivostní tabulkou a opět máme vypsat ÚNDF a ÚNKF:

| $s$ | $c$ | $b$ | $a$ | $f$ | *minterm* | *maxterm* |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | - | $a+b+c$ |
| 1 | 0 | 0 | 1 | 0 | - | $\bar{a}+b+c$ |
| 2 | 0 | 1 | 0 | 0 | - | $a+\bar{b}+c$ |
| 3 | 0 | 1 | 1 | 1 | $ab\bar{c}$ | |
| 4 | 1 | 0 | 0 | 0 | - | $a+b+\bar{c}$ |
| 5 | 1 | 0 | 1 | 1 | $a\bar{b}c$ | |
| 6 | 1 | 1 | 0 | 1 | $\bar{a}bc$ | |
| 7 | 1 | 1 | 1 | 1 | $abc$ | |

ÚNDF: $f = ab\bar{c} + a\bar{b}c + \bar{a}bc + abc$

ÚNKF: $f = (a+b+c)(\bar{a}+b+c)(a+\bar{b}+c)(a+b+\bar{c})$

# Minimalizace ÚNDF podle Booleovy algebry

$$f = ab\bar{c} + a\bar{b}c + \bar{a}bc + abc = ab(\bar{c}+c) + a\bar{b}c + \bar{a}bc = ab + a\bar{b}c + \bar{a}bc = a(b+\bar{b}c) + \bar{a}bc = ab + ac + \bar{a}bc = ab + c(a+\bar{a}b) =$$
$$= ab + ac + bc$$

Tak tady už byla úprava mnohem zajímavější, ale realizovatelná.

Pokud bychom chtěli zjednodušovat ÚNKF, bylo by nutné závorky mezi sebou roznásobit a .... to už samo o sobě je hrozná představa, takže to dělat nebudeme

# Schéma výstupní funkce ze základních logických členů

Ještě schéma naší funkce. Ze základních členů budeme potřebovat 3x dvouvstupový AND a 2x dvouvstupový OR.

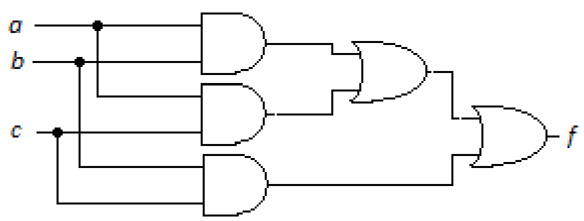



Schéma vypadá dobře a není ani moc složité, ale každý typ logického členu je obsažen v jiném integrovaném obvodu. Konkrétně dvouvstupový AND v TTL 7408, kde jsou čtyři tyto členy a dvouvstupový OR zase v TTL 7432, kde jsou také čtyři členy. Což tedy znamená celkem dva integrované obvody.

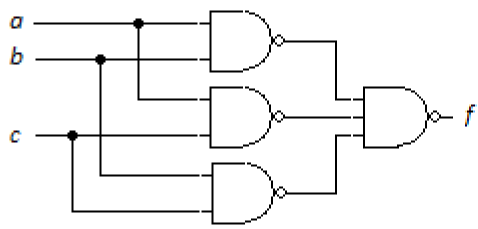
# Realizace výstupní funkce pouze pomocí členů NAND

V rámci úspor (místa i financí) se minimalizace týká i omezení typů použitých členů. Takže se pak provádí převedení již minimalizované funkce na jen NAND nebo jen NOR a to pomocí De Morganových pravidel.

Já to trošku zjednoduším. Chceme-li použít jen NAND, musíme se zbavit všech součtů ve funkci a naopak chceme-li funkci realizovat jen NOR, pak je potřeba odstranit součiny. To provedeme "dvojitou negací" nad součtem (nebo součinem). Otázka je proč dvojitá negace, když podle De Morgana stačí jen jedna negace pro změnu součtu na součin (nebo součinu na součet). Tak pokud bychom naší funkci znegovali pouze jednou, tak vlastně realizujeme funkci právě opačnou než jsme chtěli. Takže ta druhá negace je tam proto, abychom nezměnili původní funkci. Pro vlastní demorganování ji nepoužijeme, jen tam zůstane.

$$f = ab + ac + bc = \overline{\overline{ab + ac + bc}} = \overline{\overline{ab} \cdot \overline{ac} \cdot \overline{bc}}$$

# Schéma výstupní funkce jen z logických členů NAND




Když na to teď koukám, tak jsme si moc nepomohli, protože zase budeme potřebovat dva integrované obvody (TTL 7400 a TTL 7411). Sice je dvou i třívstupové NAND stejný typ, ale bohužel se rozlišuje i počet vstupů, ale doufám, že princip je jasný .

# Minimalizace výstupní funkce pomocí Karnaugovy mapy

Cílem minimalizace je nalézt co nejjednodušší vyjádření zadané logické funkce. Tato metoda je vhodná maximálně pro 4 až 5 proměnných, ale je rychlá a výsledná funkce je vždy v minimálním tvaru.

Hledáme:

**MNDF**, minimální normální disjunktní formu - logický součet minimálního počtu minimálních součinů (mintermů)

nebo

**MNKF**, minimální normální konjunktní formu - logický součin minimálního počtu minimálních součtů (maxtermů)

## Postup minimalizace pomocí Karnaughovy mapy

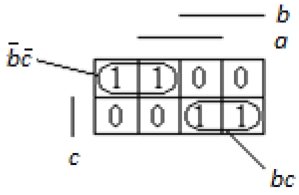
1. Zapíšeme výstupní funkci do mapy.
2. Vytvoříme smyčky. Pro hledání **MNDF** vytváříme smyčky přibližně ve tvaru čtverce nebo obdélníku, které obsahují, co největší počet „sousedních jedničkových stavů“. Počet těchto stavů musí být vždy mocninou čísla 2 (tj. 1, 2, 4, 8, 16, … atd.). Sousední jedničkové stavy jsou „jedničky“, které spolu sousedí hranou, a to i přes okraje mapy. Smyčky se mohou navzájem překrývat, neděláme však smyčky nadbytečné. Všechny jedničky musí být popsány buď v rámci některé smyčky, nebo samostatně.
3. Jednotlivé smyčky se popisují mintermy, složenými pouze z těch proměnných, které se během celé smyčky nemění (proměnná a bude v mintermu obsažena, pokud pro všechny „1“ ve smyčce nabývá stejné hodnoty, např. a = 1). Termy dvou sousedních polí se od sebe liší jen ve stavu jedné proměnné a o tuto proměnnou lze funkci, při sloučení těchto polí do smyčky, zjednodušit. Čím více sousedních „1“ je ve smyčce, tím méně proměnných bude v příslušném mintermu.
4. Výsledná **MNDF** je součtem takto vytvořených minimálních mintermů.

Pro **MNKF** platí totéž, s tím, že smyčky vytváříme kolem „sousedních nulových stavů“ a smyčky se popisují pomocí maxtermů. Výsledná **MNKF** je pak součinem těchto minimálních maxtermů.

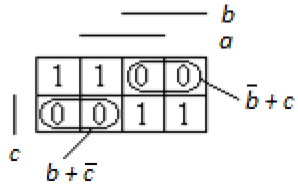
Př.: Napište pravdivostní tabulku, Karnaughovu mapu, MNDF a MNKF funkce, zadané pomocí stavových indexů.

$$f(c,b,a)=\sum(0,1,6,7)$$

| s | c | b | a | f |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |



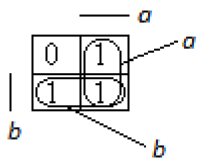

MNDF: $f=\bar b\bar c + bc$



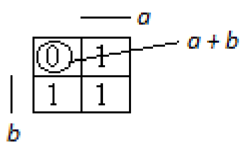

MNKF: $f=(\bar b+c)(b+\bar c)$

Př.: Vytvořte pravdivostní tabulku, Karnaughovu mapu, MNDF a MNKF logického členu OR. Uvažujte dvě vstupní proměnné.

| s | b | a | f |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 |




MNDF: $f = a + b$



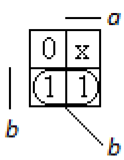

MNKF: $f = a + b$

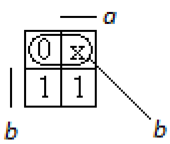
# Minimalizace neúplně zadaných funkcí

Při minimalizaci neúplně zadaných funkcí postupujeme shodně jako při minimalizaci funkcí úplně zadaných s tím, že neurčené stavy **x** zahrnujeme buď do smyček s „jedničkami“ nebo do smyček s „nulami“ nebo je nemusíme zahrnout do žádné smyčky. Vždy hledíme na výhodnost jejich polohy pro tu kterou minimalizaci. Stručně řečeno, pokud je výhodné zahrnout neurčitý stav a získat tak větší smyčku (tedy menší počet proměnných) učiníme tak, jinak ne.

Př.: pravdivostní tabulka a Karnaughova mapa funkce neúplně zadané

| s | b | a | f |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | x |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 |



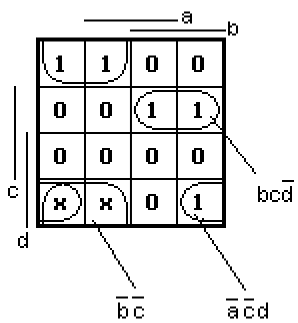




Pro **MNDF** není výhodné neurčený stav zahrnovat, vedlo by to pouze k vytvoření další smyčky a tím dalších proměnných.

Pro **MNKF** je však smyčka s **x** výhodnější neboť je popsána jen proměnnou ***b***, oproti popisu samotné „0“, jež by vedlo k popisu ***(a + b)***.

Př.: Minimalizujte neúplně zadanou funkci pomocí Karnaughovy mapy a zapište ve tvaru MNDF. Funkce je zadána stavovým indexem.

$$f(d,c,b,a) = \sum(0,1,6,7,10) + \sum_x(8,9)$$




$$MNDF: f = \bar{b}\cdot\bar{c} + bc\bar{d} + \bar{a}\cdot\bar{c}d$$