Kapitola 3

Zátvorkové formy. Návrh zložitých kombinačných systémov. Hazardy v logických systémoch.

Princípy zjednodušenia zapojení s použitím zátvorkových foriem Booleovskej algebry. Návrh zložitých kombinačných systémov s opakovanou štruktúrou. Štrukturálna dekompozícia.

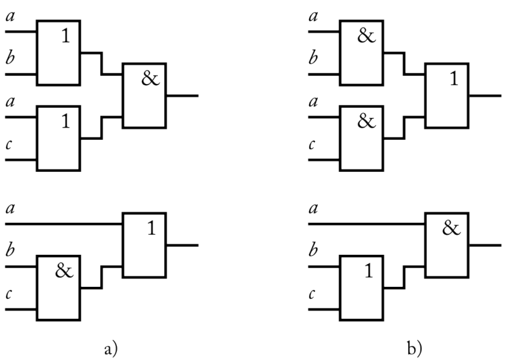
Princípy hľadania „optimálneho" riešenia

V predošlých častiach sme si uviedli postup ako dekomponovať a zapísať kombinačnú úlohu do Karnaughovej mapy, vytvoriť pravidelné konfigurácie, prepísať výrazy s použitím pravidiel Booleovej algebry do normálnej formy a zakresliť štrukturálnu schému. I keď sme počas návrhu určili optimálne konfigurácie, tak výsledný algebrický zápis nemusí viesť k minimálnej elektrickej schéme. Výrazy je možné často ďalej zjednodušovať. Jednou z možností je použitie *zátvorkových pravidiel.*
V praxi sú obmedzenia dané prevažne použitou súčiastkovou základňou a požiadavkami na vlastnosti zapojenia akým je napr. rýchlosť.
V súčasnosti vieme riešiť exaktne úlohy len s malým počtom premenných nakoľko zložitosť výpočtu rýchlo rastie.
Pri hľadaní optimálnych konfigurácií v logickom systéme s viacerými výstupmi je možné aplikovať *skupinovú optimalizáciu.* Jej princíp spočíva vo vytváraní takých pravidelných konfigurácií, ktoré sa dajú aplikovať vo viacerých Karnaughových mapách súčasne.

Zátvorkové formy

Uveďme si zátvorkové pravidlá Booleovej algebry. Ich reprezentácia je na obr. 1.

$$(a+b)\cdot(a+c)=a+b\cdot c$$
$$a\cdot b+a\cdot c=a\cdot(b+c)$$



Obrázok 1. Elektrické schémy zátvorkových pravidiel.

Príklad 3.1

Aplikujte zátvorkové pravidlá na zadanú NDF, ktorej zapojenie je na obr. 2a.

$$y=a\cdot\bar{c}+b\cdot d+\bar{c}\cdot d$$

Riešenie

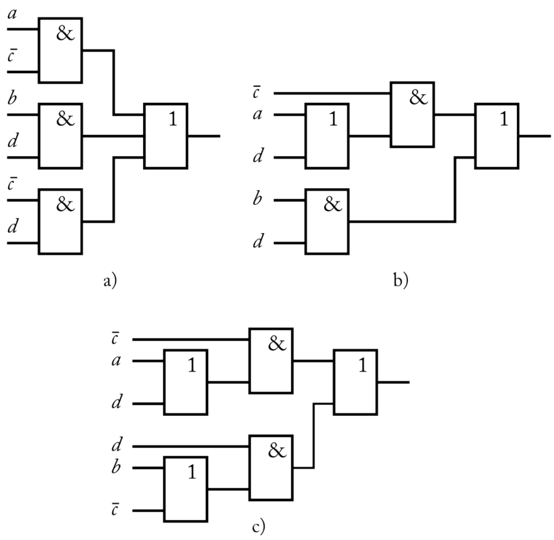
Pre aplikovanie pravidla máme dve možnosti, premenné $\bar{c}$ a d. Aplikujme pravidlo na prvý a posledný súčin:

$$y=\bar{c}\cdot(a+d)+b\cdot d$$

Výsledok zjednodušenia je na obr. 2b. Pokračujme aplikovaním pravidla po druhý krát. K výrazu najskôr pripočítajme $\bar{c}\cdot d$:

$$\begin{aligned} y&=\bar{c}\cdot(a+d)+b\cdot d+\bar{c}\cdot d\\ &=\bar{c}\cdot(a+d)+d\cdot(b+\bar{c}) \end{aligned}$$

Výsledok druhého zjednodušenia je na obr. 2c.



Obrázok 2. Aplikovanie zátvorkových pravidiel na výraz v NDF – a) pôvodný výraz, b) pravidlo aplikované jedenkrát a c) použitie pravidla na rovnaký súčin druhý raz.

Záver

Ak je nejaký súčin použitý v zátvorkovej forme je nevhodné použiť rovnaký súčin v ďalšej zátvorkovej forme, viď. počet hradiel v zapojení na obr. 2b a 2c.

Príklad 3.2

Aplikujte zátvorkové pravidlá na zadanú NKF, ktorej zapojenie je na obr. 3a. Zapíšte výsledok do 1. NPF a 2. NSF.

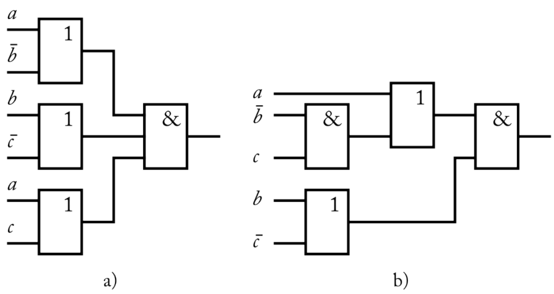
$$y = \left(a + \bar{b}\right) \cdot (b + \bar{c}) \cdot (a + c)$$

Riešenie

Pre aplikovanie pravidla máme jednu možnosť, premennú a. Aplikujme pravidlo na prvý a posledný súčet:

$$y = \left(a + \bar{b} \cdot c\right) \cdot (b + \bar{c})$$

Výsledok zjednodušenia je na obr. 3b.



Obrázok 3. Aplikovanie zátvorkových pravidiel na výraz v NKF.

Platí rovnaký „záver“ ako pri NDF.

Úpravu do 1. NPF prevedieme za pomoci substitúcie $K = \bar{b} \cdot c$:

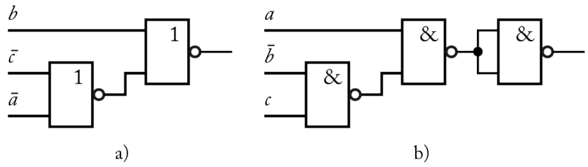
$$\begin{aligned} y &= (a + K) \cdot (b + \bar{c}) \\ &= (a \downarrow K) \downarrow (b \downarrow \bar{c}) \\ &= (a + K) \cdot \bar{K} \\ &= a \cdot \bar{K} = \bar{a} \downarrow K \end{aligned}$$

kde $K = \bar{b} \cdot c = \overline{\overline{\bar{b} \cdot c}} = \overline{b \vee \bar{c}} = b \downarrow \bar{c}$ si upravíme s použitím De Morganovho pravidla a zákona absorpcie po dosadení dostaneme

$$y = \bar{a} \downarrow (b \downarrow \bar{c})$$

Úpravu do 2. NSF prevedieme podobne za pomoci substitúcie $K = \bar{b} \cdot c = \overline{\overline{\bar{b} \cdot c}} = (\bar{b}|c)|$ a získame:

$$y = [a|(\bar{b}|c)]|$$



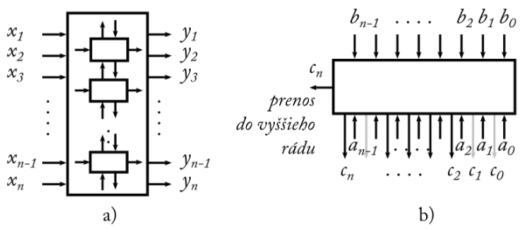
Obrázok 4. Štrukturálna schéma funkcie $y = f(a, b, c)$ vytvorená z hradiel NOR a NAND.

Návrh zložitých kombinačných systémov, štrukturálna dekompozícia

Existujú logické obvody, ktoré vo svojej štruktúre obsahujú jednoduchší blok, ktorý sa opakuje. Tento blok nazývame *iteratív*. Pri návrhu logického obvodu s opakovanou štruktúrou najskôr hľadáme popis správania sa *iteratívu*. Snažíme sa vytvoriť *iteratív* čo najmenší a s minimálnym počtom vstupných signálov. Definujeme vzťahy medzi blokmi. Tomuto spôsobu návrhu hovoríme *štrukturálna dekompozícia*. Tento prístup vedie na pomalšie systémy. Medzi takéto obvody patria napr. sčítačky, násobičky, deliče frekvencie, čítače a iné.

Príklad 3.3

Navrhnite a zakreslite schému 8-bitovej binárnej sčítačky metódou štrukturálnej dekompozície, obr. 5b. Určte celkové oneskorenie sčítačky.



Obrázok 5. Princíp logického systému s opakovanou štruktúrou – a), *n*–bitová binárna sčítačka.

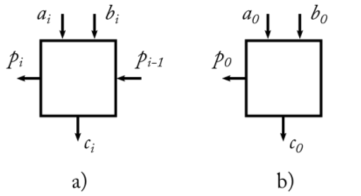
Riešenie

Uveďme si matematický princíp sčítavania dvoch čísel bez znamienka.

$$\begin{array}{rl} & a_7 \qquad\qquad a_0 \\ & b_7 \qquad\qquad b_0 \\ \text{A:} & 1011\ 0101 \\ +\ \text{B:} & 0011\ 0100 \\ \hline \textit{prenos:} & 0\ 0110\ 100 \\ \text{C:} & 1110\ 1001 \end{array}$$

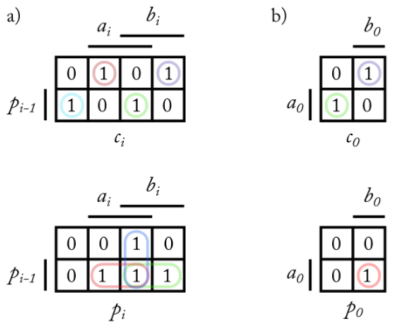
Obrázok 6. Princíp sčítavania binárnych čísel.

Z princípu sčítavania je zrejmá štruktúra iteratívu, ktorá je na obr. 7a. Prípad nultého bitu môže byť vyriešený samostatne, obr. 7b.



Obrázok 7. Navrhnutý iteratív – a) a špeciálny prípad pre nultý bit výsledku – b), kde premenná p_i predstavuje prenos z nižšieho rádu do nasledujúceho.

Na obr. 8 máme zakreslené Karnaughove mapy pre oba prípady spolu s pravidelnými konfiguráciami.



Obrázok 8. Pravidelné konfigurácie v sčítačke a špeciálny prípad.

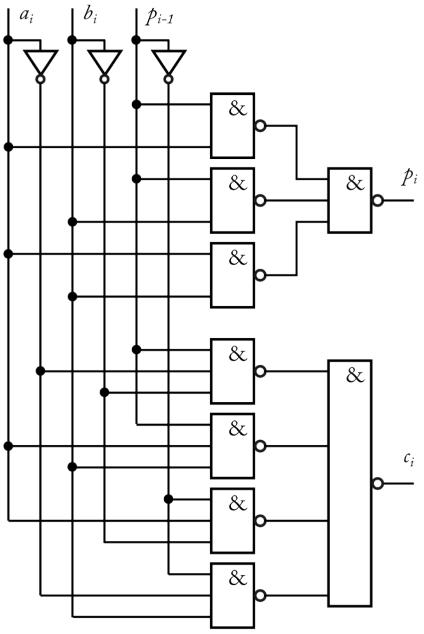
Zapíšme si NDF všetkých výstupných premenných

$$c_i = p_{i-1} \cdot \bar{a}_i \cdot \bar{b}_i + p_{i-1} \cdot a_i \cdot b_i + \bar{p}_{i-1} \cdot a_i \cdot \bar{b}_i + \bar{p}_{i-1} \cdot \bar{a}_i \cdot b_i$$
$$p_i = p_{i-1} \cdot a_i + p_{i-1} \cdot b_i + a_i \cdot b_i$$
$$c_0 = a_0 \cdot \bar{b}_0 + \bar{a}_0 \cdot b_0$$
$$p_0 = a_0 \cdot b_0$$

a preveďme ich do 1. NSF

$$c_i = (p_{i-1}|\bar{a}_i|\bar{b}_i)|(p_{i-1}|a_i|b_i)|(\bar{p}_{i-1}|a_i|\bar{b}_i)(\bar{p}_{i-1}|\bar{a}_i|b_i)$$
$$p_i = (p_{i-1}|a_i)|(p_{i-1}|b_i)|(a_i|b_i)$$
$$c_0 = (a_0|\bar{b}_0)|(\bar{a}_0|b_0)$$
$$p_0 = (a_0|b_0)|$$

Výsledné zapojenie jedného iteratívu je na obr. 9. Jedná sa o zapojenie *jednobitovej plnej sčítačky*. Naopak špeciálny prípad (bez prenosového vstupu) popísaný v obr. 8b predstavuje *polovičnú sčítačku*.



Obrázok 9. Jednobitová plná sčítačka, realizácia použitím logických hradiel NAND.

Ak uvažujeme jednotkové obneskorenie každého hradla je celkové oneskorenie 8-bitovej sčítačky rovné 3+2.7 = 17 časových jednotiek od okamžiku pripojenia vstupných čísel a, b až po získanie platného výsledku c. To je cena za jednoduchý návrh.

Praktická aplikácia

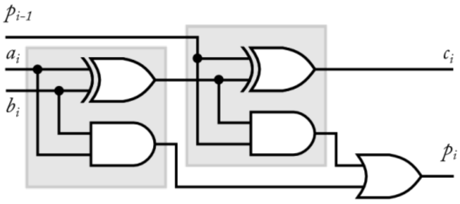
V praxi sa častejšie používa zapojenie s hradlami XOR. Zapíšme si výrazy pre polovičnú a plnú sčítačku z obr. 8.

Polovičná sčítačka: $c_i = a_i \oplus b_i$

$p_i = a_i \cdot b_i$

Plná sčítačka: $c_i = (a_i \oplus b_i) \oplus p_{i-1} = a_i \oplus b_i \oplus p_{i-1}$

$p_i = a_i \cdot b_i + p_{i-1} \cdot (a_i \oplus b_i)$



Obrázok 10. Plná sčítačka, realizácia s použitím logických hradiel XOR vytvorená z dvoch polovičných sčítačiek.