Kybernetika

Václav Dvořák

Univerzální logické funkce a syntéza booleovských funkcí pomocí univerzálních modulů

Kybernetika, Vol. 4 (1968), No. 2, (93)--104

Persistent URL: `http://dml.cz/dmlcz/124928`

KYBERNETIKA ČÍSLO 2, ROČNÍK 4/1968

Univerzální logické funkce a syntéza booleovských funkcí pomocí univerzálních modulů

VÁCLAV DVOŘÁK

V článku jsou vyšetřovány univerzální logické funkce n proměnných pro $n \geqq 3$. Je odvozen počet univerzálních funkcí v závislosti na n a ukázán postup při syntéze obecné funkce pomocí dané univerzální funkce.

I. ÚVOD

Jeden ze základních úkolů syntézy kombinačních logických obvodů je výběr takové soustavy logických prvků, která by umožnila realizaci libovolné funkce. Soustav logických prvků s touto vlastností existuje obecně nekonečně mnoho. Volba určitého systému musí být podložena ekonomickými hledisky a přirozenými schopnostmi používaných fyzikálních prvků, případně stavem výrobní technologie. Mikroelektronika přináší stále složitější stavební prvky, které realizují složitější logické funkce. Důvodem je zejména nízká cena a zvýšená spolehlivost těchto složitějších obvodů. Při stále rostoucí prostorové hustotě součástek zůstávají rozměry těchto modulů srovnatelné s rozměry dřívějších diskrétních součástek. Výrobci se snaží, aby moduly měly co nejširší použití. Větší složitost modulu má však za následek jeho méně efektivní použití v řadě aplikací. Problém absolutní minimizace tak ztrácí svůj prvořadý význam. Optimální složitost modulů je z ekonomického hlediska pak určena stavem technologie jejich výroby. V příspěvku je diskutována univerzálnost jistých složitějších modulů a způsob syntézy zadaných logických funkcí pomocí daného univerzálního modulu.

II. FUNKČNĚ ÚPLNÝ SYSTÉM BOOLEOVSKÝCH FUNKCÍ

Chování každého kombinačního obvodu lze popsat booleovskou funkcí obecně n proměnných $f(x_1, x_2, \ldots, x_n)$, která definuje zobrazení z množiny uspořádaných n-tic nul a jedniček, tj. n-násobného kartézského součinu množin $\{0, 1\}$, $\{0, 1\}^n$ do množiny $\{0,1\}$. Sestavování složitých logických obvodů z jednodušších odpovídá tvoření nových booleovských funkcí superpozicí (tj. dosazením jiných funkcí za argumenty dané funkce) nebo substitucí a permutací proměnných. Přitom je třeba dodržet ještě některá další omezení, např. vyloučit smyčky a pod. [1]. Nyní přejdeme k pojmu funkčně úplného systému booleovských funkcí.

Definice 1. Systém booleovských funkcí je funkčně úplný, jestliže superpozicí, substitucí a permutací proměnných lze z funkcí tohoto systému získat libovolnou funkci (včetně konstant 0 a 1).

Poznámka. Někdy se též automaticky předpokládá, že funkce $f = 1$, $g = 0$ jsou k dispozici v uvažovaném systému booleovských funkcí. Přidržíme se však původní definice 1.

Platí následující věta, [2]:

Věta 1. *Aby systém booleovských funkcí byl funkčně úplný, je nutné a stačí, aby obsahoval alespoň*

a) *jednu funkci nezachovávající nulu,*
b) *jednu funkci nezachovávající jedničku,*
c) *jednu funkci, která není komplementární sama k sobě,*
d) *jednu nelineární funkci,*
e) *jednu nemonotonní funkci.*

Připomeneme zde alespoň definici funkce zachovávající nulu (jedničku) a funkce komplementární k sobě samé, které budou potřeba v dalším.

Definice 2. Funkce $f(x_1, x_2, \ldots, x_n)$ zachovává nulu, jestliže platí

$$f(0, 0, \ldots, 0) = 0 .$$

Definice 3. Funkce $f(x_1, x_2, \ldots, x_n)$ zachovává jedničku, jestliže platí

$$f(1, 1, \ldots, 1) = 1 .$$

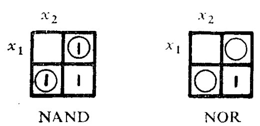
Definice 4. Funkce $f(x_1, x_2, \ldots, x_n)$ je komplementární sama k sobě, jestliže platí

$$f(x_1, x_2, \ldots, x_n) = \overline{f(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)} \overset{\mathrm{df}}{=} f^k(x_1, x_2, \ldots, x_n) .$$

Funkce komplementární sama k sobě nabývá tedy na každých dvou souborech hodnot argumentů x_i, vzájemně komplementárních, různých hodnot. Komplementárním souborům hodnot argumentů odpovídají v mapě funkce komplementární pole. V mapě funkce, která není komplementární sama k sobě, existuje tedy alespoň jedna dvojice komplementárních polí, v nichž funkce nabývá stejné hodnoty buď 0 nebo 1.

Podle věty 1 by se zdálo, že úplný systém booleovských funkcí bude obsahovat alespoň 5 funkcí. Některé funkce však splňují několik požadavků současně, takže lze vystačit s menším počtem funkcí. Bylo dokázáno, že neredundantní systém booleovských funkcí, který je úplný, obsahuje nanejvýš 4 funkce [1]. Příklad takového systému je systém funkcí $\{0, 1, x . y + y . z + x . z, x \oplus y \oplus z\}$, kde znaménko $\oplus$ značí součet mod 2. Některé známé úplné systémy booleovských funkcí jsou např. součin dvou proměnných a negace, součet dvou proměnných a negace, systém prahových funkcí, implikace a konstanta nula, součet mod 2 dvou proměnných, součin dvou proměnných a konstanta 1. Existují funkce dvou proměnných, které mají všechny vlastnosti vyžadované větou 1, např. funkce NAND a NOR a jsou tedy univerzální v tom smyslu, že pomocí nich lze realizovat každou funkci n proměnných.

Definice 5. Booleovská funkce se nazývá univerzální, jestliže tvoří úplný systém ve smyslu definice 1.



Obr. 1. Mapy funkcí NAND a NOR.

Mapy univerzálních funkcí dvou proměnných jsou na obr. 1, v němž je jedna dvojice komplementárních polí označena kroužky. Běžně se těchto funkcí používá k realizaci „součtu součinů" nebo „součinu součtů" vstupních proměnných ve dvou kaskádách podle rovnic

$$\overline{\overline{x_1 . x_2} . \overline{x_3 . x_4}} = x_1 . x_2 + x_3 . x_4 ,$$

případně

$$\overline{\overline{x_1 + x_2} + \overline{x_3 + x_4}} = (x_1 + x_2) . (x_3 + x_4) .$$

V dalším budeme vyšetřovat univerzální funkce n proměnných pro $n \geqq 3$. Např. známá funkce 3 proměnných, majorita, není univerzální funkcí. Úplný systém tvoří teprve majorita, negace a konstanta 0. Z obr. 1 lze vidět následující vlastnosti univerzálních funkcí 2 proměnných:

funkce NOR nebo NAND

(1) a′) nezachovává nulu,

(2) b′) nezachovává jedničku,

(3) c′) není komplementární sama k sobě.

Univerzální funkce obecně musí splňovat všechny podmínky věty 1. Dokážeme však nyní následující větu:

Věta 2. *Booleovská funkce je univerzální právě tehdy, když splňuje podmínky* (1) *až* (3).

Důkaz. a) *Podmínky nutné* (jestliže f je univerzální, pak platí (1) až (3)). Stačí ukázat, že neplatí-li jedna z podmínek (1) až (3), nemůže být f univerzální. Skutečně, jestliže např. f je komplementární sama k sobě, lze snadno ukázat [2], že při superpozici, substituci a permutaci proměnných u takové funkce lze získat právě jen funkce komplementární k sobě samým a žádné jiné. Podobně je tomu při nesplnění podmínky (1) nebo (2), kdy dostáváme zase jen funkce zachovávající nulu nebo jedničku. f tedy nemůže být univerzální.

b) *Podmínky postačující* (jestliže platí (1) až (3), pak je f univerzální). Důkaz dostatečnosti podmínek provedeme tak, že pomocí funkce, která splňuje (1) až (3) sestrojíme univerzální funkci dvou proměnných NOR nebo NAND. Předpokládejme nejdříve, že funkce obsahuje několik párů komplementárních polí, v nichž nabývá hodnoty 0. Vyberme z nich libovolně jeden pár. Tento pár definuje podmnožinu V množiny proměnných x_i

$$V = \{x_i \mid x_i = 0 \quad \text{pro 1. pole}, \quad x_i = 1 \quad \text{pro 2. pole dvojice}\}.$$

Pak hodnoty funkce f ve vybraném páru komplementárních polí jsou

$$\text{1. pole:} \quad [f(x_1, \ldots, x_n)]_{\substack{x_i = 0 \text{ pro } x_i \in V \\ x_i = 1 \text{ pro } x_i \notin V}} = 0\,,$$

$$\text{2. pole:} \quad [f(x_1, \ldots, x_n)]_{\substack{x_i = 1 \text{ pro } x_i \in V \\ x_i = 0 \text{ pro } x_i \notin V}} = 0$$

podle předpokladu. Dále, jelikož f nezachovává nulu ani jedničku, je

$$[f(x_1, \ldots, x_n)]_{\substack{x_i = 0 \text{ pro } x_i \in V \\ x_i = 0 \text{ pro } x_i \notin V}} = 1$$

$$[f(x_1, \ldots, x_n)]_{\substack{x_i = 1 \text{ pro } x_i \in V \\ x_i = 1 \text{ pro } x_i \notin V}} = 0\,.$$

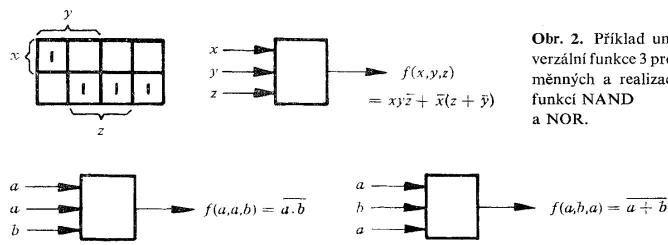
Provedeme-li substituci proměnné x za všechny proměnné $x_i \in V$ a proměnné y za všechny proměnné $x_i \notin V$, pak

$$[f(x_1, \ldots, x_n)]_{\substack{x_i = x \text{ pro } x_i \in V \\ x_i = y \text{ pro } x_i \notin V}} = F(x, y) = \overline{x + y}\,.$$

Podobně se dá ukázat, že v případě, kdy existuje v mapě funkce alespoň jeden pár komplementárních polí, ve kterých nabývá funkce hodnoty 1, lze realizovat funkci $\overline{x \,.\, y}$.

Příklad univerzální funkce 3 proměnných a realizace $\overline{x \,.\, y}$, $\overline{x + y}$ je na obr. 2.

Na základě podmínek (1) až (3) lze snadno najít počet univerzálních funkcí $A(n)$ v závislosti na počtu proměnných n. Jestliže funkce nezachovává nulu a jedničku, jsou tím definovány hodnoty funkce v jednom páru komplementárních polí. Zbývá tedy $p = 2^{n-1} - 1$ volných párů. Obsaďme jeden pár komplementárních polí v souhlasu s podmínkou (3). Máme pro to zřejmě 2^1 možností (buď do obou polí nulu nebo do obou jedničku). Zbývajících $p - 1$ párů obsadíme tak, aby podmínka (3) splněna nebyla, tj. vždy v jednom poli z páru bude jednička, v druhém nula,



Obr. 2. Příklad univerzální funkce 3 proměnných a realizace funkcí NAND a NOR.

což je 2^{p-1} možností. Podobně lze obsadit dva páry komplementárních míst v souhlasu s podmínkou (3) $2^2 = 4$ způsoby a zbývajících $p - 2$ párů 2^{p-2} způsoby tak, aby podmínka (3) neplatila atd., takže celkem máme

$$A(n) = 2^1\binom{p}{1} 2^{p-1} + 2^2\binom{p}{2} 2^{p-2} + \ldots + 2^p\binom{p}{p} 2^{p-p} =$$

$$= 2^p \sum_{k=1}^{p} \binom{p}{k} = 2^p(2^p - 1),$$

$$p = 2^{n-1} - 1.$$

Např. $A(3) = 2^3(2^3 - 1) = 8 . 7 = 56.$

Existuje tedy 56 univerzálních funkcí 3 proměnných. Obecně je

$$\lim_{n\to\infty} \frac{A(n)}{2^{2^n}} = \lim_{n\to\infty} \frac{2^{2p} - 2^p}{2^{2^n}} = \frac{1}{4} - \lim_{n\to\infty} 2^{-(2^{n-1}+1)} = \frac{1}{4}.$$

Tedy pro velká n je počet univerzálních funkcí n proměnných blízký jedné čtvrtině počtu všech funkcí n proměnných. 56 univerzálních funkcí 3 proměnných lze zredukovat na 16 ekvivalentních tříd vzhledem k permutacím vstupů. V tabulce 1 je seznam 16 funkcí, z nichž každá reprezentuje jednu ekvivaleletní třídu. Tyto funkce by se mohly uplatnit ve dvojrozměrných sítích identických logických prvků, v nichž by bylo možné realizovat libovolné funkce. Sítě logických prvků tohoto typu s uniformním propojením prvků nabývají v poslední době významu v technologii integrovaných monolitických obvodů. Práce [3] se např. zabývá sítěmi majoritních hradel a metodami syntézy libovolné funkce při zadaném propojení a rozměrech sítě. Jelikož však majorita není univerzální funkcí, je nutné používat negovaných proměnných a konstant 0 a 1, které již dohromady tvoří úplný systém logických funkcí.

Tabulka 1.

Č. třídy	Standardní součet	Poznámka
1	0	NOR
2	0, 4	
3	0, 6	
4	0, 1, 2	
5	0, 1, 6	
6	0, 4, 5	
7	0, 5, 6	
8	0, 1, 2, 5	
9	0, 3, 4, 6	
10	0, 1, 2, 3, 4	
11	0, 1, 2, 3, 6	
12	0, 2, 3, 4, 5	
13	0, 3, 4, 5, 6	
14	0, 1, 2, 3, 4, 5	
15	0, 1, 2, 3, 5, 6	
16	0, 1, 2, 3, 4, 5, 6	NAND

III. ANALÝZA JEDNODUCHÉHO OBVODU S UNIVERZÁLNÍMI MODULY

Metody analýzy a syntézy obvodů s univerzálními moduly (a s libovolnými moduly vůbec) budou ilustrovány na jednoduchém zapojení podle obr. 3. Při řešení problémů analýzy a syntézy použijeme rovnic s booleovskými maticemi [4]. Uvedeme nejdříve několik potřebných definic.

Booleovská matice (A_{ij}) je obecně nečtvercová matice, jejíž prvky A_{ij} mohou nabývat hodnoty 0 nebo 1, $A_{ij} \in \{0, 1\}$. Definujeme následující relace (znaménka $+$, $\sum$, $.$, $\prod$ značí logické součty a součiny):

a) komplement matice $\overline{(A_{ij})} \overset{\text{df}}{=} (\overline{A_{ij}})$,

b) transpozice matice $(A_{ij})^{\mathrm{T}} \overset{\text{df}}{=} (A_{ji})$,

c) rovnost matic $(A_{ij}) = (B_{ij})$ jestliže $A_{ij} = B_{ij}$,

d) implikace matic $(A_{ij}) \rightarrow (B_{ij})$ jestliže $A_{ij} \rightarrow B_{ij}$,

e) součin matic $(C_{ij}) = (A_{ij}) \cdot (B_{ij})$ jestliže $C_{ij} = A_{ij} \cdot B_{ij}$,

f) součet matic $(C_{ij}) = (A_{ij}) + (B_{ij})$ jestliže $C_{ij} = A_{ij} + B_{ij}$,

g) přímý součin matic $(C_{ij}) = (A_{ik}) \otimes (B_{kj})$ jestliže $C_{ij} = \sum_k A_{ik} \cdot B_{kj}$,

h) Θ-součin matic $(C_{ij}) = (A_{ik})\,\Theta(B_{kj})$ jestliže $C_{ij} = \prod_k (A_{ik} \cdot B_{kj} + \overline{A}_{ik} \cdot \overline{B_{kj}})$,

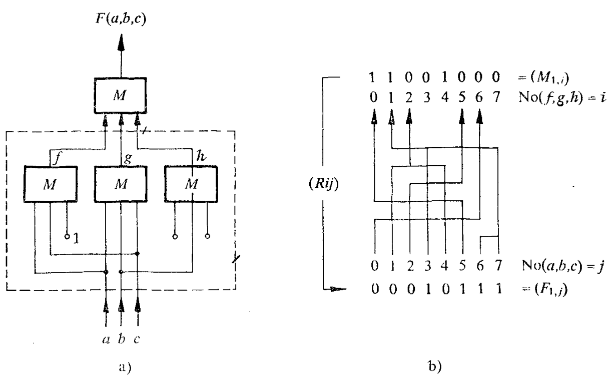
pro všechna i, j.

V operacích c) až f) se předpokládá, že matice (A_{ij}) a (B_{ij}) jsou téhož typu, u operací g), h) se na typ matice klade stejná podmínka jako u běžného násobení matic v algebře.

Přejděme nyní k analýze obvodu na obr. 3. Zde jsou všechny moduly identické, ale realizují různé funkce podle proměnných přivedených na vstupy. Výsledná funkce proměnných a, b, c

$$F(a, b, c) = M(f, g, h),$$

kde M je funkce realizovaná univerzálním modulem. Dekadickou hodnotou $(a \cdot 2^2 + b \cdot 2^1 + c \cdot 2^0)$ binárního čísla abc budeme nazývat konfiguračním číslem (a, b, c) a značit $\mathrm{No}(a, b, c)$. Podobně pro proměnné f, g, h. Kódovým číslem funkce $F(x, y, z)$ budeme nazývat posloupnost jejích funkčních hodnot odpovídající rostoucí posloupnosti konfiguračních čísel (x, y, z) a takto uspořádanou posloupnost funkčních hodnot budeme značit jako řádkovou matici $(F_{1,i})$ s prvky $F_{1,i}$, $i = 0, 1, \ldots, 7$.



Obr. 3. Jednoduchý obvod s univerzálními moduly: a) blokové schéma, b) transformace konfiguračních čísel realizovaná první kaskádou.

Z obr. 3a je vidět, že první kaskáda modulů přiřazuje každému konfiguračnímu číslu (a, b, c) jisté konfigurační číslo (f, g, h). Toto zobrazení je znázorněno schematicky v obr. 3b šipkami. Jdeme-li v obr. 3b proti směru šipek, realizujeme vlastně nejednoznačnou inverzní transformaci množiny konfiguračních čísel (f, g, h) do množiny konfiguračních čísel (a, b, c). Jelikož každému konfiguračnímu číslu (f, g, h) je přiřazena jistá funkční hodnota $M(f, g, h)$ a rovněž každému konfiguračnímu číslu (a, b, c) je přiřazena funkční hodnota $F(a, b, c) = M(f, g, h)$, převádí tato inverzní transformace též kódové číslo funkce $M(f, g, h)$ na kódové číslo funkce $F(a, b, c)$. Použijeme-li pro kódová čísla řádkových matic $(M_{1,i})$, $(F_{1,i})$, je možno inverzní transformaci zapsat takto:

$$(M_{1,i}) = 11001000 \rightarrow (F_{1,i}) = 00010111$$

(šipka značí přiřazení, nikoliv implikaci). Tuto transformaci si lze představit jako násobení matice $(M_{1,i})$ jistou transformační maticí (R_{ij}) zprava,

$$(M_{1,i}) \otimes (R_{ij}) = (F_{1,j}) .$$

Prvek R_{ij} matice (R_{ij}) je roven jedné tehdy a jen tehdy, když konfigurační číslo $\text{No}(a, b, c) = j$ se první kaskádou transformuje na konfigurační číslo $\text{No}(f, g, h) = i$. V každém sloupci matice (R_{ij}) je tedy zřejmě jen jediná jednička, vzhledem k jednoznačnosti této přímé transformace. Matice, která má zmíněnou vlastnost se nazývá unitární; lze tedy danému zapojení modulů v 1. kaskádě jednoznačně přiřadit jistou unitární matici (R_{ij}).

IV. SYNTÉZA DANÉ FUNKCE POMOCÍ UNIVERZÁLNÍCH MODULŮ

Metodu syntézy ukážeme opět jen na jednoduchém obvodu podle obr. 3. Jde vlastně o speciální případ syntézy, kdy zapojení prvků je známo a hledáme proměnné nebo konstanty, které je nutno připojit na vstupy prvků tak, abychom na výstupu obdrželi žádanou funkci. Problém syntézy je pak možno formulovat jako problém řešení booleovské rovnice

$$M[f(a, b, c), g(a, b, c), h(a, b, c)] = F(a, b, c) , \tag{5}$$

kde f, g, h jsou neznámé funkce proměnných a, b, c a F a M zadané funkce. Omezující podmínka je, že funkce f, g, h realizuje jediný univerzální modul $M = M(x, y, z)$. Takto formulovaný problém může mít jedno, několik nebo žádné řešení. Podmínkami existence řešení se zde nebudeme zabývat. Je možno odkázat na [4]. Při řešení booleovské rovnice (5) použijeme booleovských matic [4]. Místo rovnice (5) lze pak řešit maticovou rovnici (4):

$$(M_{1,i}) \otimes (R_{ij}) = (F_{1,j}) ,$$

v níž neznámá matice (R_{ij}) odpovídá systému funkcí f, g, h a je dána vztahem [4]

$$(R_{ij}) = (M_{1,i})^{\mathrm{T}} \ominus (F_{1,j}) . \tag{6}$$

Takto získaná matice (R_{ij}) nemusí být unitární. Jestliže obsahuje více než jednu jedničku v některém sloupci, existuje více systémů funkcí f, g, h splňujících rovnici (5). Matice (R_{ij}) totiž vlastně reprezentuje několik unitárních matic, z nichž každá obsahuje některé jedničky matice (R_{ij}) a sice právě jednu z každého sloupce. Jestliže některý sloupec matice (R_{ij}) neobsahuje žádnou jedničku, pak systém funkcí f, g, h splňující rovnici (5) neexistuje. (Pro podrobné zdůvodnění lze odkázat na [4].)

Přechod od matice (R_{ij}) k systému funkcí f, g, h je obrácením postupu, podle kterého jsme při analýze sestrojovali matici (R_{ij}). Jestliže $R_{ij} = 1$, pak konfiguračnímu číslu $\mathrm{No}(a, b, c) = j$ odpovídá konfigurační číslo $\mathrm{No}(f, g, h) = i$, takže lze přímo sestavovat tabulku kódových čísel funkcí f, g, h. Je-li z tabulky možno vybrat řešení (v případě, že jich existuje více) f, g, h takové, že

$$f, g, h = M(x, y, z),$$

kde x, y, z jsou ze souboru $\{a, b, c, 0, 1\}$, pak je řešen náš původní problém. Postup ukážeme na následujícím příkladě.

Mějme realizovat funkci $F(a, b, c) = ab + ac + bc$ (majorita) s kódovým číslem $(F_{1,i}) = 00010111$ pomocí univerzálního modulu, který realizuje funkci $M(x, y, z) = \bar{y}(\bar{x} + \bar{z})$ s kódovým číslem $(M_{1,i}) = 11001000$. Matici (R_{ij}) spočteme podle rovnice (6):

$$(7) \qquad (R_{ij}) = (M_{1,i})^{\mathrm{T}} \ominus (F_{1,j}) = [1\,1\,0\,0\,1\,0\,0\,0]^{\mathrm{T}} \ominus [0\,0\,0\,1\,0\,1\,1\,1] =$$

$$= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & \mathbf{1} & 1 & 1 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 1 & \mathbf{1} & \mathbf{1} \\ 1 & \mathbf{1} & 1 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & \mathbf{1} & 0 & 1 & 0 & 0 & 0 \\ \mathbf{1} & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Z této matice lze psát ihned tabulku 2 funkcí f, g, h (jednotlivé sloupce zde odpovídají konfiguračním číslům i, přiřazeným konfiguračním číslům j). Řešení (f, g, h) je velmi

Tabulka 2.

j	0	1	2	3	4	5	6	7
f g h	00111 11011 01101	00111 11011 01101	00111 11011 01101	001 000 010	00111 11011 01101	001 000 010	001 000 010	001 000 010

mnoho. Musíme vybrat takové řešení, aby funkce f, g, h byly realizovatelné pomocí daného modulu. **Všechny funkce realizovatelné daným modulem jsou proto shrnuty** v tabulce 3. Nejurčitější tvar má funkce g, jejíž kodové číslo z tabulky 2 je

$$(g_{1,i}) = \emptyset\,\emptyset\,\emptyset\,0\,\emptyset\,0\,0\,0\,.$$

Symbol $\emptyset$ představuje jedničku nebo nulu. Zvolíme-li za něj jedničku, získáme největší volnost při výběru zbývajících funkcí f a h. Z tabulky 3 je vidět, že pro funkci g nejlépe vyhovuje některá z prvních tří funkcí. Zvolíme např. první funkci F_1

$$(g_{1,i}) = 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\,.$$

Tabulka 3.

i	F_i \ x, y, z	x: 0 0 0 0 1 1 1 1 y: 0 0 1 1 0 0 1 1 z: 0 1 0 1 0 1 0 1
1	$\bar{y}(\bar{x} + \bar{z}) = M(x, y, z)$	1 1 0 0 1 0 0 0
2	$\bar{x}(\bar{y} + \bar{z}) = M(y, x, z)$	1 1 1 0 0 0 0 0
3	$\bar{z}(\bar{x} + \bar{y}) = M(x, z, y)$	1 0 1 0 1 0 0 0
4	$\overline{xy} = M(x, 0, y)$	1 1 1 1 1 1 0 0
5	$\overline{xz} = M(x, 0, z)$	1 1 1 1 1 0 1 0
6	$\overline{yz} = M(y, 0, z)$	1 1 1 0 1 1 1 0
7	$\overline{x + y} = M(x, y, 1)$	1 1 0 0 0 0 0 0
8	$\overline{x + z} = M(x, z, 1)$	1 0 1 0 0 0 0 0
9	$\overline{y + z} = M(y, z, 1)$	1 0 0 0 1 0 0 0
10	$\bar{x} = M(x, x, x)$	1 1 1 1 0 0 0 0
11	$\bar{y} = M(y, y, y)$	1 1 0 0 1 1 0 0
12	$\bar{z} = M(z, z, z)$	1 0 1 0 1 0 1 0

Zbývající funkce f, h jsou pak

$$f_{1,i} = \emptyset\ \emptyset\ 1\ x\ \emptyset\ x\ x\ x$$

$$h_{1,i} = \emptyset\ \emptyset\ 1\ x\ \emptyset\ x\ x\ x\,,$$

kde $\begin{matrix} x \\ x \end{matrix}$ značí libovolnou kombinaci 0 a 1 kromě $\begin{matrix} 1 \\ 1 \end{matrix}$. Největší volnost ve výběru poslední funkce získáme, když za x dosadíme 0. Hledáme tedy v tabulce 3 funkci s kódovým číslem $\emptyset\,\emptyset\,1\,0\,\emptyset\,0\,0\,0$. Takové funkce existují 3. Použijeme F_8 s kódovým číslem, které přiřadíme např. funkci f

$$(f_{1,i}) = 1\,0\,1\,0\,0\,0\,0\,0\,.$$

Poznamenejme, že jsme kódové číslo funkce F_8 mohli stejně dobře přiřadit funkci h, poněvadž obě funkce f a h mají stejný tvar. Poslední funkce h má teď tvar $(h_{1,i}) = \emptyset\,\emptyset\,1\,\emptyset\,\emptyset\,\emptyset\,\emptyset\,\emptyset$. Nejjednodušší funkce tohoto typu je zřejmě přímo proměnná y s kódovým číslem

$$(h_{1,i}) = 0\,0\,1\,1\,0\,0\,1\,1\,.$$

Máme tedy řešení:

$$f = \overline{a + c} \quad = M(a, c, 1)\,,$$
$$g = \bar{b}(\bar{a} + \bar{c}) = M(a, b, c)\,,$$
$$h = b\,.$$

Skutečně je

$$F(a, b, c) = \bar{g}(\bar{f} + \bar{h}) = (b + ac)(a + c + \bar{b}) = ab + ac + bc\,.$$

Nenulové prvky unitární matice odpovídající zvolenému řešení jsou v matici (7) vyznačeny polotučně.

V. ZÁVĚR

V příspěvku byly vyšetřovány univerzální funkce n proměnných a na základě pozměněných kritérií úplnosti bylo možno najít i počet univerzálních funkcí n proměnných. Univerzální funkce 3 proměnných byly tabelovány. Při návrhu obvodů s podobnými základními funkcemi bylo použito metody řešení booleovských rovnic pomocí ekvivalentních maticových rovnic. Postup byl ilustrován na poměrně jednoduchém obvodu. V případě složitějších obvodů se však zdá, že tento postup nebude možno použít vzhledem k tomu, že chybí vedlejší podmínky, které by umožňovaly volit systematicky určité řešení z množiny všech možných řešení. Kromě vypracování účinných metod syntézy funkcí více proměnných nabízejí se k řešení další otázky jako minimální počet modulů nutných k realizaci obecné funkce 3 proměnných, případně který univerzální modul je nejefektivnější, které zapojení daných prvků je univerzální (tj. může realizovat libovolnou funkci daného počtu proměnných). Z hlediska technických aplikací v integrovaných obvodech je zvláště důležité studium vlastností dvojrozměrných sítí univerzálních modulů s uniformním propojením.

(Došlo dne 22. prosince 1966.)

LITERATURA

[1] H. H. Loomis, R. H. Wyman: On complete sets of logic primitives. IEEE Trans. on Electronic Computers *EC-14* (April 1965), 173—174.

[2] Вавилов, Портной: Синтез схем электронных цифровых машин. Советское радио, Москва 1963.

[3] R. H. Canaday: Two-dimensional iterative logic. AFIPS Conference Proc. vol. 27, pt. 1. Spartan, Washington D. C. 1965.

[4] R. S. Ledley: Digital computer and control engineering. McGraw Hill, 1960.

General-Purpose Boolean Functions and Synthesis of Boolean Functions by means of General-Purpose Modules

VÁCLAV DVOŘÁK

In this paper there are investigated general-purpose boolean functions of n variables for the case $n \geqq 3$. On the basis of modified conditions for completeness of the system of boolean functions the expression for a number of general-purpose functions is derived. The list of 16 classes of general purpose functions of 3 variables is presented. A method of solution of boolean equations by means of equivalent matrix equations was used for the design of circuits with general-purpose building blocks. The process is illustrated on the example of synthesis of a two-stages-cascade of general-purpose modules.

Ing. Václav Dvořák, Výzkumný ústav matematických strojů, Loretánské nám. 3, Praha 1.