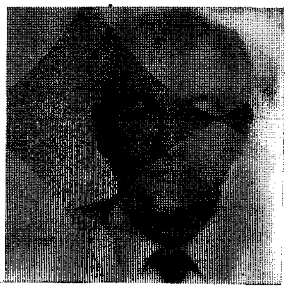
Mikroszámítógépes program bemenetszám-korlátozott NÉS-NÉS kapuhálózat szintézisre

KÁLDI TIBOR
Elektrokémiai Technológiákat Fejlesztő KFT
SZENTIDAY KLÁRA
Kandó Kálmán Villamosipari Műszaki Főiskola



KÁLDI TIBOR

A Jedlik Ányos Közlekedésgépészeti Technikumot 1958-ban végezte el. 8 évig az EIVRT-ben, a TUNGSRAM RT jogelődjénél dolgozott a tranzisztorgyártás szakterületén. 1967-től 1989-ig az OMH Optikai osztályán dolgozott. Jelenleg az Elektrokémiai Technológiákat fejlesztő KFT-ben dolgozik, ahol műszerfejlesztéssel és számítástechnikával foglalkozik.

SZENTIDAY KLÁRA

Oki. fizikus, oki. elektronikai szakmérnök, egy. doktor. Korábban a Híradástechnikai Ipari Kutató Intézet /a MEV jogelődje/ tud. főmunkatársa, jelenleg a KKVMF EATI docense, szakcsoportvezetői. Szakmai tevékenysége: tranzisztor struktúravizsgálatok, optoelektronikai eszközök méréstechnikája, minőségellenőrzés. Oktatási területe a félvezetők konstrukciós számításai, digitális áramkörök.

Összefoglalás

A MOS LSI/VLSI és felhasználói tervezésű áramkörökben a kapuk maximálisan megengedett bemenetszáma erősen korlátozott. Az M.A. Breuer által bevezetett, bemenetszám-korlátozott NÉS-NÉS kapuhálózat szintézis egyik változatát készítettük el ZX Spectrum személyi számítógépen futtatható program formájában. A program kompatibilis az általunk korábban kidolgozott logikai függvényminimalizálási eljárással.

Bevezetés

Korábbi publikációnk [1] mikroszámítógépes programot ismertetett egy és többkimenetű /max. 10/ logikai függvények minimalizálására. A ZX Spectum személyi számítógépen futtatható program a Breuer-féle irredundáns lefedési algoritmusokra [2] épült. Ez a program, ami kétszintű /ÉS-VAGY/ kapuhálózat szintézisére készült, nem veszi figyelembe azt a nyilvánvaló tényt, hogy a gyakorlatban a tervező számára olyan kapuk állnak rendelkezésre, amelyek bemenetszáma /FAN IN/ és terhelhetősége /FAN OUT/ az áramkörök fizikai adottságaitól függően korlátozott.

A hazánkban is üzembe állított Hierarchikus Tervező Rendszer /HTR, [3]/ és a számos egyéb LSI/VLSI CAD/CAM eszköz döntően MOS technológiával készült, kapuhálózat-alapú szeletekkel dolgozik. A terhelhetőségi korlát a MOS áramköröknél viszonylag rugalmasan kezelhető, mivel a terhelés növekedése a kapuk logikai funkcióját nem veszélyezteti, bár hátrányos módon a működési sebességet csökkenti.

Sokkal erősebben korlátozott azonban a bemenetszám, különösen a NÉS, ill. ÉS kapuk esetében. Ekkor – mint ismeretes – az n- vagy *p*-csatornás MOS tranzisztorok vezérlő- /driver/ ág tranzisztorait sorba kell kapcsolni. Ugyanakkor a drain-áram állandósága csak úgy biztosítható, ha a tranzisztorok gate-elektródájának méretarányát /a w/L hányadost/ a bemenetszám arányában megnövelik. Ez számos hátrányt jelent az áramkör felépítésére és elektromos paramétereire nézve, ezért a gyakorlatban általában nem készítenek háromnál több-bemenetes NÉS vagy ÉS kaput. A CMOS áramköröknél – ahol a terhelő-ág a vezérlő-ág duálja – a korlátozás egyéb kapufunkciók /pl. NVAGY/ esetében is fennáll.

Közleményünk ZX Spectrum /48 K/ személyi számítógépre kidolgozott, bemenetszám-korlátozott szintézist ismertet kettőnél többszintű NÉS-NÉS hálózatok kialakításával. A program kompatibilis az [1] minimalizálási eljárással, de csak egy kimenetű hálózat megvalósítására alkalmas. Lehetőség van azonban a többkimenetű minimálalak kimenetei közötti választásra, és a szintézisnek kimenetenkénti elvégzésére. A program ASSEMBLY nyelven készült.

Beérkezett: 1989. II. 20. (H)

1. Alapismeretek

Az n változós Boole-függvényt n dimenziós vektorok (kockák) halmazával adjuk meg. A c vektor összetevői $c = \{0, 1, x\}$ értékűek, és az f függvényt az

$$f = \{c_1, c_2, ..., c_m\}$$

halmazzal definiáljuk. A csonka szorzattagok (mintermek, P-tagok) mint kockák esetében tehát „0" jelenti a negált, „1" a ponált és „x" a közömbös változó szimbólumát.

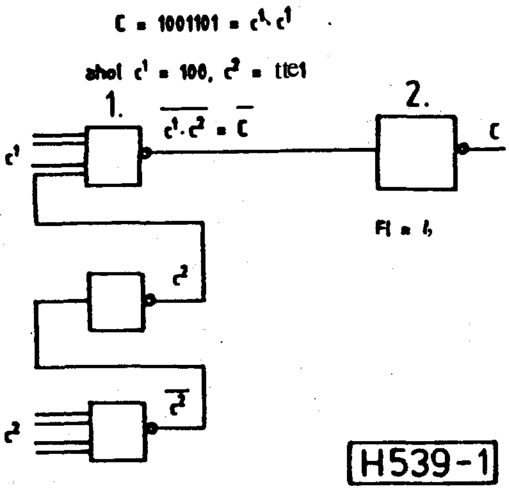
A szorzattagos függvényalak megvalósítására legegyszerűbbnek az ÉS-VAGY kapus realizálás tűnik, ahol a bemenetek szintjén ÉS kapuk és a kimenet (több kimenet esetén kimenetek) szintjén VAGY kapu található. A legtöbb logikai áramkörcsalád előnyben részesíti azonban a negációt végző kaputípust, ami azt jelenti, hogy egyszerűbben, kevesebb áramköri elemmel valósítható meg pl. a NÉS mint az ÉS kapu. A De-Morgan szabályok alkalmazásával az ÉS-VAGY alak egyszerűen NÉS-NÉS alakra hozható. Legyen pl. $f = \overline{AB}C + \overline{C}D$, kétszer negálva a függvényt:

$$f = \overline{\overline{f}} = \overline{\overline{ABC + CD}} = \overline{\overline{ABC} \cdot \overline{CD}}$$

nyerhető.

2. Kettőnél többszintű hálózatstruktúrák

Tekintsük a c = {1001101} kockát, amely egyetlen, hétbemenetű ÉS kapuval volna realizálható. Tegyük fel, hogy a megvalósítás céljára csak NÉS kapuk állnak rendelkezésre és olyanok, amelyek maximálisan négybemenetűek lehetnek. A továbbiakban nevezzük a maximálisan megengedett bemenetszámot (*F*AN IN korlátot) egyszerűen *FI*-nek. Példánkban tehát *FI* = 4. A kocka realizálásához szükséges NÉS hálózatot az *1. ábra* mutatja be. (Megjegyezzük, hogy az ábrákon a NÉS kaput és az invertert az egyszerűség kedvéért a szabványos jelöléstől eltérően „üres doboz" szimbólummal jelöljük.) Első lépésként az eredetileg 7 db, *x*-től különböző változót tartalmazó kockát egy darab 3- és egy darab 4-változós kockára hasítjuk



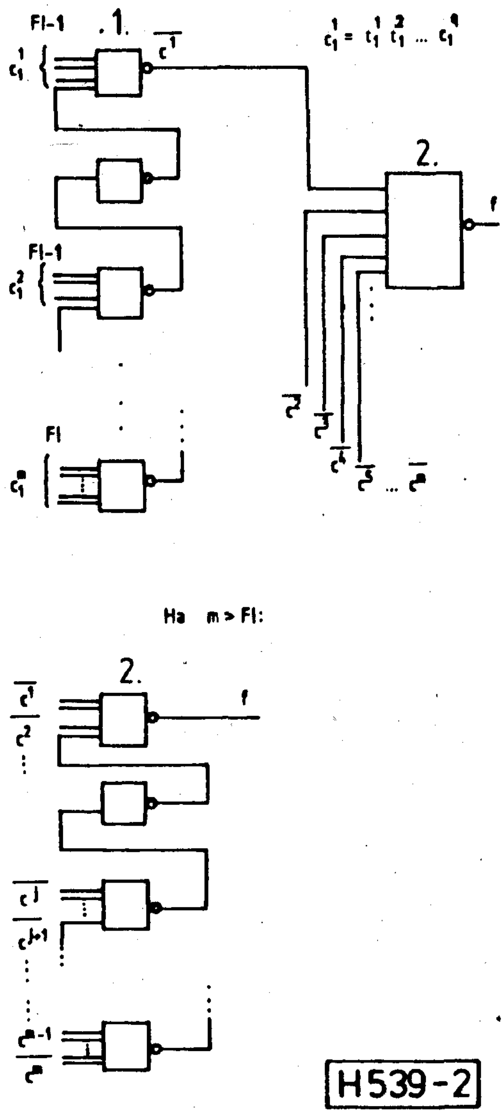
1. ábra. Példa kockahasításra

$$c = c_1 \cdot c_2$$

alapján. Tekintve, hogy a megvalósítást NÉS kapukkal kell elvégezni, gondoskodni kell a NÉS-művelet negálásáról is. Az ábrából látható, hogy az 1. szint kimenetén a kocka negált változata jelenik meg, amelyet a 2. szinten lévő kapu „visszaforgat". Ez a szabály általánosítható a kettőnél többszintű hálózatokra is, ezért minden esetben páros számú szintből kell állnia a NÉS-NÉS hálózatnak.

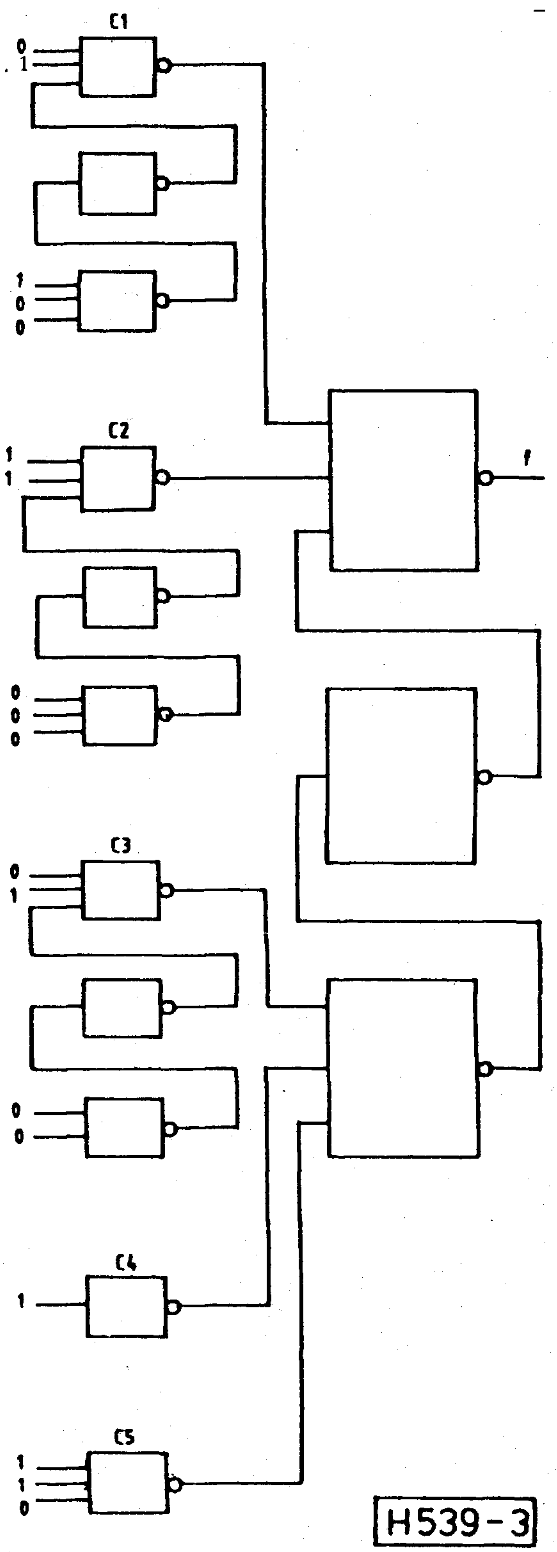
A 2. ábra kockahalmaz realizálási lehetőségét szemlélteti. Ha m > *FI*, a 2. szintnek megfelelő kaput is bontani kell az 1. ábrán bemutatott kockahasítás mintájára.

A hálózatstruktúra sok esetben egyszerűsíthető az ún. *faktorizációs eljárások* segítségével. Ebben az esetben a realizálandó kockahalmazból, vagy annak alkalmasan választott részhalmazaiból a közös elemeket szorzófaktorok formájában kiemeljük. A faktorok és a hányadosok elkülönített megvalósítása általában kevesebb kaput igényel, mint az egyszerű kockahasítás. Az *egyenes faktoros* realizálásnál megkeressük a faktorokat és képezzük a hozzájuk tartozó hányados-halmazt. *Komplemens faktoros* realizálásnál a D_1, D_2, ... hányados-halmazokat negáljuk, majd az így kapott kockahalmazt minimalizáljuk. A komplemens faktoros eljárás bonyolultsága folytán a Spectrum gépen csak igen körülményesen volna megvalósítható.



2. ábra. Faktorizáció nélküli realizálás általános sémája, *f* = {c1, c2, ..., cm}

A logikai függvények minimalizálásánál a célkitűzést a kapubemenetszám csökkentése képezi. A megvalósítás költségei, ill. bonyolultsága a kapuhálózat bemeneti és összekötő huzalozásával áll arányban mind a hagyományosnak tekinthető IC- tokos kártyák, mind a felhasználói tervezésű áramkörök esetében. Bemenetszám-korlátozásnál más a helyzet, itt a kapuk szá-



3. ábra. Példa a C = {01100x, 11x000, 0x1x00, 1xxxxx, 110xxx} kocka faktorizáció nélküli realizálására FI = 3 esetén

mát kell minimalizálni, bár itt sem használják fel minden esetben az összes lehetséges bemenetet. A gyakorlatban elfogadott módon a bemenetszám-korlátozott szintézis *költségfüggvénye* a logikai függvény realizálásához szükséges *kapuszámmal* egyezik meg. Célszerű ezért többféle kapuhálózatstruktúra esetében először a költségfüggvényt meghatározni, majd azt a felépítést választani, amelynek költségfüggvénye a legkisebb.

3. Faktorizáció nélküli hálózatszintézis

Tekintsük például a C= $\{c_1, c_2, c_3, c_4, c_5\}$ = {01100x, 11x000, 0x1x00, 1xxxxx, 110xxx} halmazt és legyen a realizálás feltétele az FI = 3 bemenetszámkorlát. A példában a változószám n = 6 és a kockák száma m = 5. A felépítést és költségfüggvényt az alábbi megfontolások mellett határozzuk meg:

- csak azokat a bemeneteket kell figyelembe venni, amelyek x-től különbözőek;
- feltételezzük, hogy a bemeneteken a változók negált alakban is rendelkezésre állnak, ahol a függvényben 0 szerepel.

A megvalósított hálózat a *3. ábrán* látható. Tekintve, hogy m értéke is nagyobb FI-nél, a második szinten lévő NÉS kaput hasítani kell.

Számítsuk ki a *költségfüggvényt*, vagyis a realizáláshoz szükséges kapuk számát, természetesen a közbenső invertereket is kapunak tekintve. A bemeneti NÉS tömbben egymásba csatlakozó kapu-inverter párokat találunk. Mindegyik kapu FI-1 számú változót képes fogadni, mivel egy bemenetet az inverter igényel. Kivétel az utolsó kapu, ami a maradék bemeneteket fogadja, itt a bemenetszám elérheti az FI értéket. A kimeneti NÉS tömb annyi bemenetet fogad, amennyi a kockák m száma.

Jelöljük a kockák x-től különböző változóinak (bemeneteinek) számát ξ-vel, és vezessük be az alábbi segédfüggvényt:

$$G(\xi) = 0 \qquad \text{ha } \xi = 0;$$
$$G(\xi) = 1 \qquad \text{ha } 1 \le \xi \le FI;$$
$$G(\xi) = 2.\mathrm{CEIL}\left(\frac{\xi - FI}{FI - 1}\right) + 1 \qquad \text{ha } \xi > FI$$

(A CEIL-függvény jelentése hasonló az INTEGER-függvényhez, azonban felfelé kerekítés történik. Pozitív szám esetében pl. CEIL(3,6) = 4 adódik.)

Ha egyetlen kockából áll a függvény (m = 1), akkor a *faktorizáció nélküli költségfüggvény:*

$$\$C(1) = G(\xi^1) + 1.$$

$m > 1$ esetében pedig

$$\$C(1) = \sum_{i=1}^{m} G(\xi^i) + G(m)$$

adódik. Visszatérve az előbbi példára, $\xi^1 = \xi^2 = 5$, $\xi^3 = 4$, $\xi^4 = 1$ és $\xi^5 = 3$ valamint m = 5 helyettesítéssel $C(1) = 14 adódik, ami éppen a *3. ábrán* látható kapuk számát adja.

4. Egyenes faktoros realizálás

A faktorizációt az alábbi egyszerű példával illusztráljuk. Legyen
$f = \bar{A}BCD + \bar{A}B\bar{C}F + \bar{A}BE = \bar{A}B(CD + \bar{C}F + E)$.
Kockahalmazzal kifejezve: C = {0111xx, 010xx1, 01xx1x}.

A halmaz $g(k)$ közös faktora és a közös faktorral elosztott hányadoshalmaz: g(k) = {01xxxx};$C/g(k)$ = {xx11xx, xx0xx1,xxxx1x}. Nagyobb halmazok esetén a közös faktor (ha van) kiemelése után maradt hányadoshalmaz további csoportokra bontható, amelyekből rendre g(1), g(2) stb. faktorok emelhetők ki. Valamely halmaz maximális jóságú faktora (*MF*) az a szorzótényező, amely esetén

$$(w.\,h)_{max} = MF.$$

w jelenti azt az x-től különböző változószámot, amelyet a faktor tartalmaz, míg h azoknak a kockáknak a száma, amelyekben a faktor megtalálható. Az előbbi példában talált $g(k)$ közös faktor jósága $w = 2$ és $h = 3$ alapján e két érték szorzata, tehát 6. w-t a faktor szélességének és h-t a faktor magasságának is nevezik. Végül azok a kockák, amelyek legfeljebb csak a $g(k)$ közös faktort tartalmazzák, a Cm maradékhalmazt alkotják.

Faktorizáció esetén a *költségfüggvény* meglehetősen bonyolult lesz. Jelöljük $\$C(2)$-vel az egyenes faktoros költségfüggvényt, amelyet a következőképpen határozhatunk meg. Ha valamennyi faktortípus létezik, hat fő szint különböztethető meg. Tegyük fel, hogy a $g(k)$ közös faktoron kívül q számú további faktor létezik és létezik a Cm maradékhalmaz is, ami a $g(k)$ közös faktoron kívül más faktort nem tartalmaz.

Az első NÉS kapu-szint a q számú faktorról leválasztott d^j_i hányadoshalmazt realizálja. Annyi kapucsoport van, ahány faktor, ezért az első összegezésnél az i index q-ig fut. A j index a kapucsoportokon belüli kapuszámot jelzi, az első csoport $q1$, a második csoport $q2$ és az í-edik csoport qi számú kaput tartalmaz. A második szint a hányadoshalmaz kapukimeneteit fogja össze. A harmadik szinten lépnek be a faktorok, míg a negyedik szint a Cm maradékhalmaz negáltjának a belépési szintje. Ha nem volna közös faktor, itt fejeződne be a realizáció. Közös faktor létezése esetén $g(k)$ az ötödik szinten lép be, míg a hatodik szint egyetlen inverterből áll. Képletben:

$$\$C(2) = \sum_{i=1}^{q}\sum_{j=1}^{qi} G(\$d_i^j) + \sum_{i=1}^{q} G(\$qi) + \\ + \sum_{i=1}^{q} G(\$g(i)+1) + G(q+x) + \sum_{i=1}^{q} G(\$Cm_i) + \\ + G(\$g(k)+1) + 1.$$

Az $\$$ megjelölés arra vonatkozik, hogy a kérdéses mennyiség kapubemenetszámot jelent. x a Cm maradékhalmaz kockáinak számát jelenti. A képlet jobb megértését a **4.** ábra segíti, ami az egyenes faktoros realizálás általános sémáját szemlélteti.

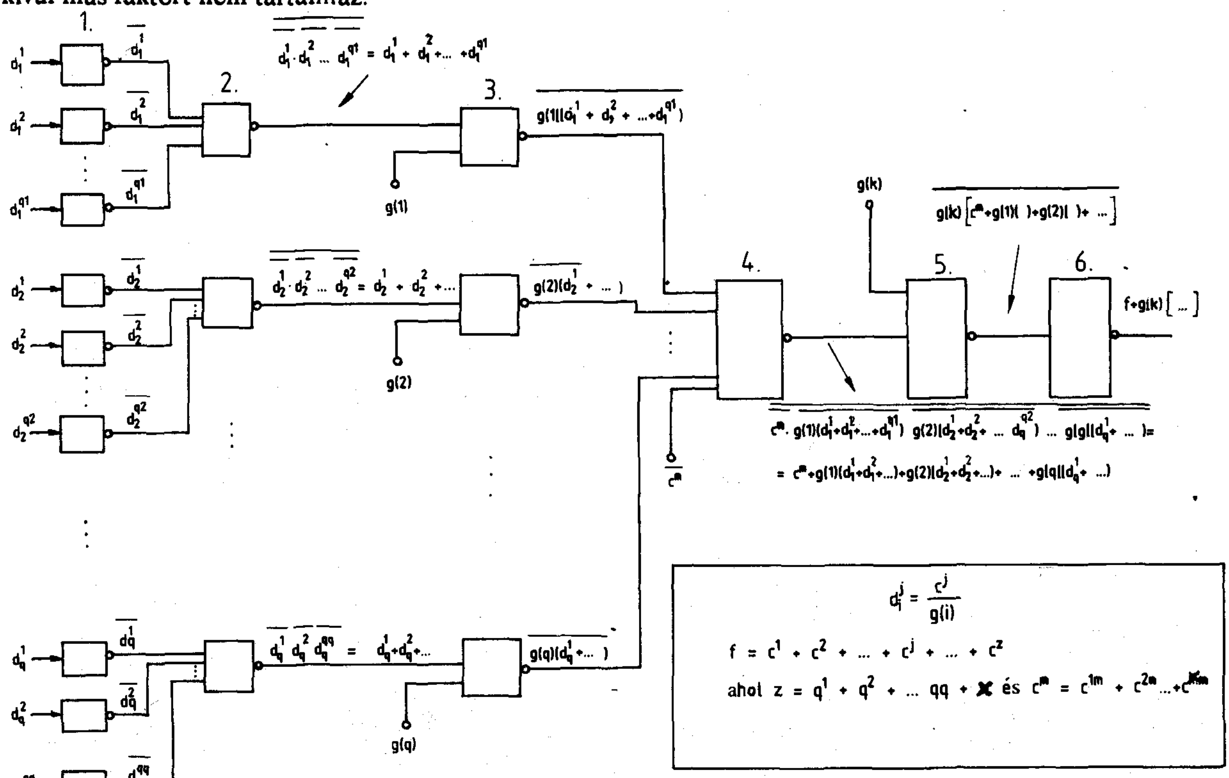
5. Algoritmusok

NÉS művelet

A NÉS műveletet az

$E_n \# c$

operációval algoritmizálhatjuk, ahol E_n az n-edren-



4. ábra. Egyenes faktoros realizálás általános sémája

H539-4

1. táblázat

Az F-operáció koordináta-táblája

		c_i		
	F	0	1	x
c_j	0	0	x	x
	1	x	1	x
	x	x	x	x

dű kocka, amelynek minden változója közömbös érték (x). A „#" szimbólum az éles szorzat műveleti jele (ld. [1]).

Faktorkeresés

A faktorkeresésre az F operációt vezették be, amelynek koordináta-tábláját az 1. táblázat mutatja. Adott $C = c_1, c_2, ..., c_i, c_j, ..., c_z$ halmaz esetén, ami z számú kockából áll, a halmaz faktorát a

$g = (.....(..(..(((c_1Fc_2)Fc_3)..Fc_i)Fc_j)...Fc_z)$

műveletsor elvégzésével kaphatjuk meg. Ha végeredményként g = xxxx....xxxx (tehát csupa x) adódik, akkor a kérdéses halmaznak vagy részhalmaznak nincs közös faktora.

Hányadoshalmaz előállítása

A c/g művelet a 2. táblázat segítségével végezhető el. Ha Ω adódik, a g faktort az i-edik c kocka nem tartalmazza.

2. táblázat

A c/g hányadosképzés koordináta-táblája

			g	
	c/g	0	1	x
c_i	0	x	Ω	0
	1	Ω	x	1
	x	Ω	Ω	x

Maximális jóságú faktor keresése

Adott C halmaz maximális jóságú faktorának megtalálásához valamennyi kockakombinációt végig kellene vizsgálni. Ez azt jelenti, hogy ha a halmaz z számú kockából áll:

$$\binom{z}{2} + \binom{z}{3} + \cdots + \binom{z}{z}$$

számú műveletet kellene elvégezni. Az összehasonlítás céljából az eredményeket tárolni is kellene, ami meghaladja a Spectrum gép tárkapacitását. Ezért egy egyszerűbb, azonban M. A. Breuer által is javasolt megoldást választottunk.

Rendezzük a kockahalmazt mátrixba, és számozzuk meg a pozíciókat. Jelentse $i=1$ a kocka-sor első elemét, vagyis az adatmártrix első oszlopát. Emeljük ki azt a halmazt, ahol az $i=1$ pozícióban 0 van. Ezután az F-operációval megkeressük a $g^o_{i=1}$ közös faktort. Amennyiben $h \geq 2$ és $w \geq 1$, meghatározzuk az $FM(g^o_{i=1})$ jóságot. Ezután kiemeljük azt a halmazt, ahol az $i=1$ pozícióban 1 áll. Meghatározzuk a

$g^1_{i=1}$

közös faktort, majd az előbbi feltételek mellett meghatározzuk az $FM(g^1_{i=1})$ jóságot. Összehasonlítjuk a két jóságot, majd a nagyobbat tároljuk. Ezután $i=2$ esetében megismételjük az eljárást, és az előző eredménnyel összevetve, a nagyobb jóságot tároljuk. A vizsgálatot valamennyi pozícióra, tehát

$i = 1...n$-ig

elvégezzük. A végeredményként legnagyobbnak bizonyuló faktor lesz a $g(1)$ közös faktor. Ezután a c/g operációval meghatározzuk a $g(1)$ faktort tartalmazó kockák hányadoshalmazát.

A következő lépésben az eredeti C^o halmazt két részre osztjuk: C' lesz az, ami tartalmazza $g(1)$-et, és

C^o-C'

a második halmaz. Most a C^o-C' halmazra megismételjük a faktorkeresési eljárást, az eredményül kapott legnagyobb faktor lesz $g(2)$. Ismét meghatározzuk a c/g operációval a $g(2)$ faktort tartalmazó kockák hányadoshalmazát. A halmaz szétválasztást és faktorkeresést mindaddig folytatjuk, ameddig a feltételeknek eleget tevő faktor leválasztható. Végül azok a kockák, amelyek már nem faktorizálhatók, a Cm maradékhalmazt alkotják, és az eljárást ezzel befejeztük.

6. Programszerkesztés

Kiindulási feltételek:

- tetszőleges kezdeti lefedés választható a csonka P tagokat kockákkal kifejezve;
- többkimenetű függvény esetén a szintézis kimenetenként végezhető el.

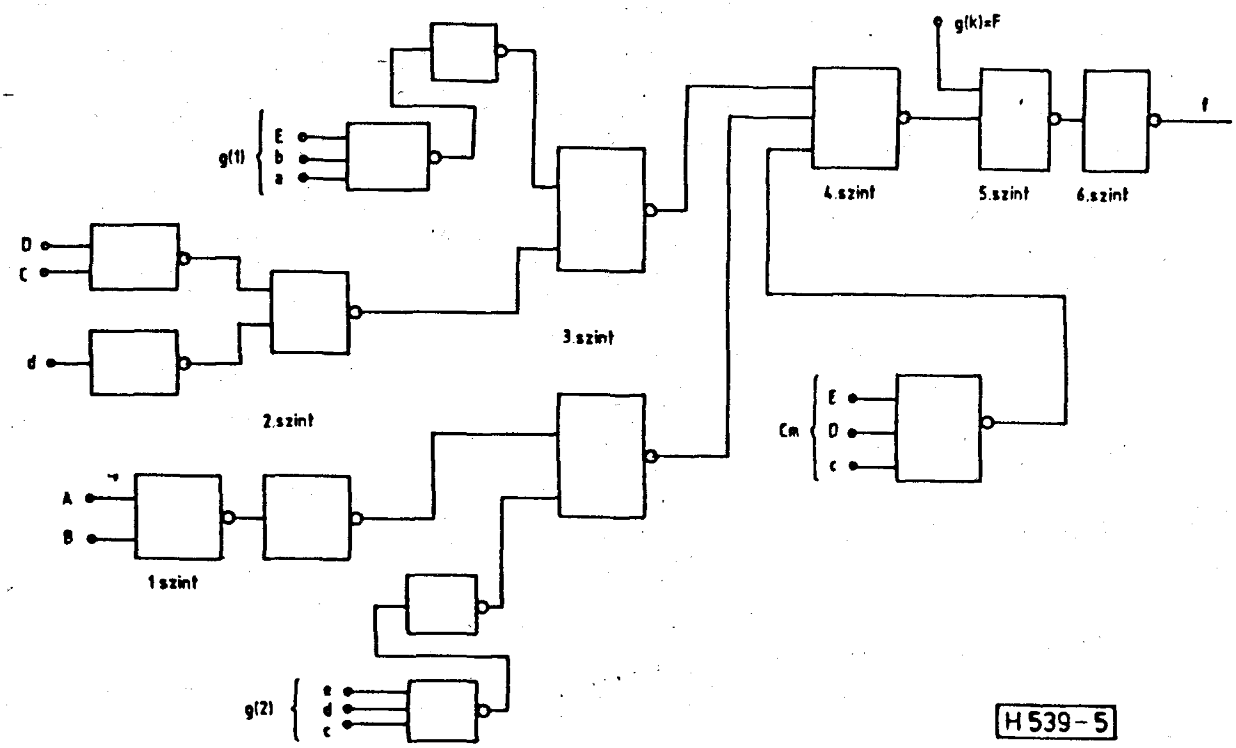
Bemeneti jellemzők

Az adatbevitel kockánként történik a $\{0, 1, x\}$ szimbólumok beírásával. Az egyes kockák elemeinek n száma maximálisan 20 lehet azzal a megkötéssel, hogy a közömbös változók (az x- értékek) száma nem lehet nagyobb 16-nál. A kockák m száma maximálisan 255 lehet.

Kimeneti adatok

A program a faktorizáció nélküli megvalósítás és az egyenes faktoros megvalósítás opciót ajánlja fel menü formájában. Célszerű mindkettőt elvégeztetni, majd a kijelzett $\$C(1)$ és $\$C(2)$ költségfüggvényt összehasonlítva, a kisebb értékűhöz tartozó hálózatstruktúrát választani.

A program a jobb áttekinthetőség és á bemenetek azonosíthatósága érdekében a bemeneteket rendre az ábécé betűivel jelöli meg. Ha a kockahalmazt adatmátrixnak tekintjük, a mátrix első oszlopa kapja az a, második oszlopa a b stb. betűjelet. Mivel az „x" szimbólummal jelzett változókhoz nem tartozik kapubemenet, elegendő csak a 0 és az 1 (tehát a negált és a ponált) változókat kijeleztetni. A negált változókat a program *kisbetűvel*, míg a ponált változókat *nagybetűvel* illeti. A betűkkel való megjelölés szükségességét jól



5. ábra. Példa a C = {111100, 110x00, 1110xx, 1000xx, 100011} halmaz egyenes faktoros realizálására $FI = 3$ esetén

3. táblázat

f	e	d	c	b	a
1	1	1	1	0	0
1	1	0	X	0	0
1	1	1	0	X	X
1	0	0	0	X	X
1	0	0	0	1	1

illusztrálja a 3. *ábra*, ahol csak azt tüntettük fel, hogy a bemeneten a változó 0 vagy 1 alakban szerepel-e. Így igen nehéz kideríteni, hogy melyik bemenet melyik változóhoz tartozik.

Végeredményként a program kijelzi a $g(k)$ közös faktort (ennek hiányában közli, hogy ilyen NINCS), a $g(1)$, $g(2)$ stb. faktorokat a hozzájuk tartozó hányados-halmazokkal, majd a Cm maradékhalmaz elemeit. A kijelzett betűcsoportok alapján a hálózat a 2. vagy 4. ábra mintájára megszerkeszthető.

7. Tervezési példa

Valósítsuk meg az alábbi kockahalmazt többszintű NÉS hálózattal! Legyen C = {111100, 110x00, 1110xx, 1000xx, 100011} és válasszuk az FI =3 bemenetszám-korlátot. A kockahalmaznak megfelelő adatmátrixot a helyiértékek betűjeleivel a 3. táblázat tartalmazza. A változók száma: n = 6 és a kockák száma: m = 5. Lefuttatva a programot

$$\$C(1) = 22$$
$$\$C(2) = 15$$

adódik, ami azt jelenti, hogy faktorizáció esetén már 15 db kapu elegendő a megvalósításhoz. Ez utóbbi megoldást választva, az alábbi betűcsoportok jelennek meg:

$g(k) = F$

$g(1) = Eba$; $\quad C1/g(1) = \{DC, d\}$

$g(2) = edc$; $\quad C2/g(2) = \{BA\}$

$Cm \; = EDc$

A példa egyszerűsége folytán a betűcsoportok jelentése a 3. táblázaton jól felismerhető. A szemléletesség kedvéért a faktorokat keretbe foglaltuk. A hálózat rajzát az 5. ábra szemlélteti.

IRODALOM

[1] *Káldi-Szentiday:* Mikroszámítógépes program logikai függvények minimalizálására. Híradástechnika XXXIX. évf. 10. sz. 1988. p. 446-451

[2] Design Automation of Digital Systems. Chapter Two, Melvin A. Breuer. 1972 by Prentice-Hall, Inc.

[3] Automatikus struktúra szintézis a Hierarchikus Tervező Rendszerhez. Tanulmány. Készült a Mikroelektronikai Kormánybiztos és a MEV megbízásából 1985-ben. Szerzők: Keresztes Péter, Ágotai István stb.