

## Logikai hálózatok számítógépes vizsgálata\*

ETO: 164.1:661.325.65.001.4

A digitális berendezések számára kialakított logikai hálózatok már viszonylag egyszerűbb esetben is olyan sok báziselemet tartalmaznak, hogy a terhelési, késleltetési tulajdonságok vizsgálata, a helyes logikai működés ellenőrzése, az esetleges téves bekötések helyének megkeresése, a meg nem engedett vezérlések kimutatása rendkívül fáradságos feladat. A fenti feladatok jól algoritmizálhatók, ezekre logikai analízist és egyéb vizsgálatokat végző programok készíthetők. Az analízis programok különösen előnyösen használhatók a logikai kapcsolási rajz módosítás utáni ellenőrzésre. A konstrukciós tervezés megkezdése előtt — mind az integrált áramkörtokok, mind a tokokból felépült kártyák esetén — célszerű a helyes logikai működésről számítógépes szimulációval előzőleg meggyőződni.

Gyártás közben és a végellenőrző méréseknél szükséges az egyes kártyák ellenőrzése. Ebben az esetben a hiba kimutatásán túlmenően alapvető követelmény a hiba típusának és helyének meghatározása. Az optimális vizsgáló jelek kiválasztására diagnosztikai tesztek generáló számítógép-programok készíthetők.

A továbbiakban az első feladatkör ellátásra alkalmas programokat analízis programnak, a második feladatcsoporthoz tartozókat pedig diagnosztizáló programnak nevezzük. Jelen dolgozat célja, hogy rövid bevezetést nyújtson a logikai hálózatok vizsgálatának fentiekben vázolt két feladatkörébe.

A logikai hálózatok vizsgálatáról publikált külföldi irodalom a részletekre vonatkozóan elég szűkszavú. Hazánkban több helyen folyik ilyen tevékenység — bizonyos fókig eltérő irányokban — pl. BME Elektroncsövek és Félvezetők Tanszék, MTA-AKI, Távközlési Kutató Intézet. Publikációkat tekintve ez csak a Híradástechnikában közölt számítógépprogram katalógusban tükröződik. Az [5] referencia részletesen kifejti a logikai hálózatok analízisére szolgáló program tulajdonságait, ezért a következőkben a diagnosztikával kapcsolatos kérdéseket fogjuk bővebben tárgyalni. Bemutatjuk, hogy a diagnosztika hiba-táblázat módszerrel, érzékenységi út módszerrel, hiba-szótár módszerrel végezhető el és felhívjuk a figyelmet a Boole differencia jelentőségére a hibák kimutatásánál.

### 1. Logikai analízis

Az elektronikus áramkörök analizálásánál kitüntetett figyelmet kell fordítani (i) a modell alkotásra, (ii) a megengedhető összekapcsolási lehetőségekre,

\* A munka a Számítástechnikai Koordinációs Intézet megbízásából készült.

Beérkezett: 1972. III. 4.

(iii) a bemenet és kimenet (gerjesztés és felelet) kapcsolására és (iv) a felhasználó orientált programrendszer szempontjaira. Tekintsük át ezeket a kérdéseket integrált áramkörtokokból felépített logikai hálózatok esetén.

A logikai hálózat modellezésére elvileg több lehetőség kínálkozik. Elképzelhető ellenállásokból, kapacitásokból, diódákból, tranzisztorokból stb. felépített modell, amely nemlineáris áramkör formájában írja le a hálózatot. Ez a leírás bázis elemek (logikai alárámkörök) esetén szükséges, azonban bonyolultabb hálózatok esetén egyrészt a számítási igény rohamosan nő, másrészt a logikai hálózat működésének megítélése szempontjából túzzottan részletes eredményeket szolgáltat.

Ismeretes, hogy NAND, illetve NOR művelet segítségével minden kombinációs logikai művelet kifejezhető, tehát egyetlen művelet elegendő a kombinációs hálózatok leírásához. Ennek megfelelően olyan modell is felépíthető lenne, amely csupán pl. NAND kapukat és késleltetéseket reprezentáló elemeket tartalmazna. Egy tényleges logikai hálózat ilyen helyettesítő képe kialakítása és a kapott eredmények értelmezése és felhasználása a konkrét hálózatra azonban nagyon nehézkes lenne.

Az integrált áramkörtokokból felépített logikai hálózatok esetén célszerű az integrált áramkörtok készlet által biztosított logikai modellekre támaszkodni, tehát AND, OR, NAND, NOR, NOT stb. kapukat, D, JK tárolókat, sőt nagyobb funkcionális elemeket (ADDER, DECODER stb.) is megengedni. Az időviszonyokat a báziselemek bemenetei és kimenetei között értelmezhető maximális késleltetési idővel lehet modellezni. Az áramviszonyok számítása a terhelések és terhelhetőségek (FAN-IN, FAN-OUT) meghatározására egyszerűsíthető.

A logikai hálózat tényleges számítása előtt, összekapcsolhatóság szempontjából, ellenőrzésre van szükségünk. Ennek keretében meghatározandó

a) az összes bekötési rendellenesség (hiányzó bekötés, nem értelmezhető bekötések stb.),

b) az összes áramkörtok szempontból tilos összeköttetés, amely báziselem károsodásra vezet,

c) az összes logikai szempontból tilos összeköttetés (pl. bizonyos kapuk bemenete nem lehet összekötve saját kimenetével),

d) továbbá, hogy az áramkör kombinációs, szinkron szekvenciális vagy aszinkron szekvenciális működésű-e. E vizsgálatok feleletet adnak olyan fontos kérdésekre, mint pl. van-e visszacsatolás az áramkörben, van-e aszinkron működésű tároló az áramkörben, hogyan történik a tárolók vezérlése stb.

Logikai hálózatoknál a bemenet és kimenet kapcsolata több módszerrel leírható:

a) Meghatározható a logikai hálózat bármely pontjának logikai egyenlete,

b) kiírható a kiválasztott ponton levő logikai értékek a bemenőjelek és a tárolók belső állapotának minden értéke mellett,

c) a hálózat bármely pontpárja között, a két pontot összekötő valamennyi útra meghatározható a késleltetési idő.

E feladatok tényleges megoldása sok nehézségbe ütközik. A logikai egyenletek nagyobb hálózatok esetén áttekinthetetlen hosszúságúak lehetnek, a tárolók figyelembevétele a logikai egyenletek felírásánál nehézkes stb. A két pont közötti késleltetési idő értelmezése az 1—0 és 0—1 átmenetek idejének eltérése következtében, a nagyobb funkcionális egységek különböző bemenetei és kimenetei közötti eltérő késleltetések következtében bizonyos mértékben önkényes.

A logikai hálózat analizisének legfontosabb feladata, hogy adott bemenő jelekhez és belső állapotokhoz előállítsa az összes kimenet logikai értékét. A hálózat bemenő jelei a hálózat bemenő változói, a külső vezérlőjelek és a tárolók vezérlését végző órajelek. A hálózat belső állapotát tárolók valósítják meg. Ezek lehetnek vagy az órajel egy meghatározott szintváltozására billenő szinkron tárolók, vagy a kapcsoléletről, ill. egymásról vezérelt aszinkron vezérlésű tárolók. A bemenőjelek minden adott tetszőleges kombinációja és a szinkron vezérlésű tárolók adott belső állapota esetén meghatározható az aszinkron vezérlésű tárolók és az összes kimenet logikai értéke, továbbá kiszámítható a szinkron vezérlésű tárolók által a következő órajel idején létrejövő belső állapot. Ily módon rendelkezésünkre áll az adott bemeneti vezérlés hatására keletkező kimeneti jel a logikai hálózat tetszőleges pontján, valamint — a tárolók állapotainak kiírásával — a logikai hálózat kódolt állapot-táblázata. Ezáltal olyan eredményeket kapunk, melyek mérés segítségével is csak rendkívül körülményesen határozhatók meg, mivel a belső csomópontok nem állnak rendelkezésre.

A felhasználó orientált programrendszer szempontjából magától érthetődő módon biztosítanunk kell, hogy a program alkalmazása egyszerű legyen, a felhasznált gépidő lehető legkevesebb legyen, az adott számítógép konfiguráció (elsősorban az operatív memória) kihasználása optimális legyen. Az alkalmazás alapvető követelménye, hogy a báziselemek és integrált áramköri tokok számításához szükséges adatai mágnesszalagon tárolva a tervező rendelkezésére álljanak. A felhasználó a katalógusban már megtalálható elemekre a típus megadásával (pl. SN 7472 N) hivatkozik. A programrendszer karbantartója kívánásra az adatbankot elvileg tetszőlegesen, akár nagyobb funkcionális egységekkel is bővítheti. Az a tény, hogy a mérnökök által eddig is használt integrált áramköri katalógus és a mágnesszalagon tárolt adatbank azonos hivatkozást biztosít, a logikai analízis program használatát elősegíti.

A programrendszer használatát járulékos szolgáltatásokkal is célszerű elősegíteni. A logikai hálózat tervezésének több rutin-feladata egyszerűen programozható, pl. a megépítéshez szükséges integrált áramköri tokok számának és típusának összesítése,

a megengedett és tényleges terhelések meghatározása stb. Mindezek szükségesek ahhoz, hogy a logikai analízis-program hatékony eszközzé váljék a tervező kezében.

## 2. Diagnosztika

Az elmúlt években több eljárást dolgoztak ki egyetlen előforduló hiba kimutatására. Az eljárásokat két nagy csoportra oszthatjuk: (i) fix vizsgálati eljárások, melyeknél a vizsgáló jelek függetlenek az előző vizsgálatok eredményétől; (ii) szekvenciális vizsgálati eljárások, melyeknél a vizsgáló jel függ az előző vizsgálat eredményétől. Logikai rendszereknél elsősorban a fix vizsgálati eljárásokat alkalmazzák, ezért csak ezekkel foglalkozunk.

A fix vizsgálati eljárások közül a hiba-táblázat módszert, az érzékenységi út módszert és a hibaszótár módszert ismertetjük.

2.1 A hiba-táblázat módszernél vizsgáljunk egy kombinációs logikai áramkört, melynek  $n$  bemenetét jelölje  $X_1, X_2, \dots, X_n$ , és hibamentes kimenetét pedig  $F$  (az egyszerűség érdekében csupán egyetlen kimenetet tárgyalunk). A különböző hibák hatása az 1. táblázat szerint foglalható össze. A táblázat baloldala a bemeneti változók  $2^n$  lehetséges változatának listája.

A hibatáblázat kiindulási alakja 1. táblázat

$X_n \dots X_2 X_1$	$F$	$F_1 F_2 \dots F_l \dots$
0 ... 0 0	0	1 0 ... 0 ...
0 ... 0 1	1	1 0 ... 0 ...
0 ... 1 0	0	1 0 ... 1 ...
...	...	...
...	...	...
1 ... 1 1	0	0 0 ... 1 ...

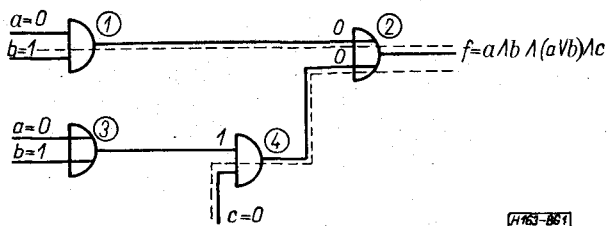
$F$ , mint említettük a hibamentes kimenetet,  $F_1, F_2, \dots, F_j, \dots$  a kimenet abban az esetben, ha valamelyik hibaok fellép. A jobb oldalról egyes oszlopok elhagyhatók, nevezetesen, ha valamelyik  $F_j$  oszlop megegyezik  $F$ -el (vagyis a  $j$ -edik hibának nincs hatása a kimenetre) vagy, ha  $F_i$  és  $F_k$  megegyezik (vagyis két különböző hibának azonos hatása van). Ezen törlések után mindegyik oszlop különböző lesz.

$m$  megkülönböztethető hiba esetén  $m+1$  oszlop áll a jobboldalon  $F, F_1, \dots, F_m$ . Az  $m+1$  oszlopot  $2^n$  soros hiba-táblázatba,  $T$ -be foglalhatjuk. A probléma ezek után  $T$  azon sorainak megtalálása, amelyek alapján az oszlopok megkülönböztethetők. A hibadetektálás céljából annyi sort hagyhatunk ki  $T$ -ből, hogy az első oszlop különböző legyen a többi oszloptól. A hiba-detektálás céljából szükséges vizsgálatok  $N_d$  számára az  $1 \leq N_d \leq m$  korlát érvényes. Kimutatták, hogy a hiba lokalizálására szükséges vizsgálatok  $N_l$  számára az  $1 + 1 \text{ ldm} \leq N_l \leq m$  korlát érvényes.

A hiba-táblázat sorainak minimalizálása analóg a logikai kapcsolóstanból ismert prim implikánsok meghatározásával. A hiba-táblázat kombinációs áramkörökre megszorítás nélkül alkalmazható. Hátránya, hogy a sorok száma exponenciálisan növekedik a bemenet  $n$  számával. Gyakorlatilag  $n \leq 10$  be-

menet, néhány kimenet és  $m \leq 100$  hiba esetén alkalmazható.

2.2 Az érzékenység út módszernél vizsgáljunk egy bázis elemekből álló logikai hálózatot. Azokat a hibákat tekintjük, amelyek abból adódnak, hogy az alap-áramkör a logikai egy vagy a logikai nulla értéken megrekedt. Ezek jelölése  $(s-a-1)$ , illetve  $(s-a-0)$  az angol stuck at 1, és stuck at 0 kifejezésekből adódóan. Az érzékenységi út mentén előforduló hiba esetén a kimenet megváltozik. Az érzékenységi út mentén fellépő  $(s-a-1)$  és  $(s-a-0)$  hibáknál a kimenet különbözni fog a hibamentes áramkör kimenetétől. Például az 1. ábrán látható kapcsolásnál alkalmazzuk az  $a=0, b=1, c=0$  vizsgálójelet. Ekkor az 1 és 2 kapukon keresztül vezető út az  $f$  kimenethez érzékenységi út, mivel a 4 kapu kimenete 0. A 4 és 2 kapukon vezető út is érzékenységi út, mivel a 3 kapu kimenete 1 és  $c=0$ . Tehát az  $a=0, b=1, c=0$  vizsgálójel e két út mentén fellépő hibákat kimutatja. Más vizsgálójelekkel a kapcsolás további útjai ellenőrizhetők.



1. ábra. Az érzékenységi út értelmezése

Az érzékenységi út a lehetséges vizsgálójeleknek csak egy tört részével számol. E vizsgálójelek a hiba táblázat azon sorainak felelnek meg, melyekhez nagy számú hiba tartozik. Az eljárással minden  $(s-a-1)$  és  $(s-a-0)$  hiba kimutatható és a módszer alkalmazható nagy áramkörök esetén is. Az érzékenységi út módszer tetszőleges egyszerű hiba detektálására kiterjeszhető és számítógép segítségével gazdaságosan elvégezhető. Az eljárás nem minimalizálja a vizsgáló jelek számát.

2.3 A hiba-szótár módszerrel kombinációs és szekvenciális áramkörök egyaránt vizsgálhatók. A hiba szótár ismert hibák esetén fellépő vizsgálati eredmények rendszerezett összeállítás. A program a logikai áramkör leírását tartalmazza és a különböző hibákat a program bemenő adatainak változtatásával állítjuk elő. A hálózatra vizsgálójeleket kapcsolunk és a hibás esetekhez tartozó kimeneteket regisztráljuk. Az így kapott hiba-szótár segítségével a ténylegesen fellépő hibák lokalizálhatók. Nagy hálózatok vagy rendszerek esetén a hiba-szótár igen terjedelmes, ezért adat redukáló leképezések alkalmazása válik szükségessé.

A diagnosztikával foglalkozó fejezet lezárásaként a vizsgáló jelek generálásának fontos matematikai módszerét, a Boole-differencia bevezetését tárgyaljuk. Érdekesképpen előljáróban megjegyezzük, hogy a logikai hálózatoknál értelmezett Boole differencia a hálózatelméletben alkalmazott  $S_i = \frac{\partial y}{\partial x_i}$  érzékenységgel analóg fogalom.

Ha valamely logikai hálózat bemenő változói  $X_1, X_2, \dots, X_n$  és kimenő változója  $F$ , akkor az

$$F = F(X_1, X_2, \dots, X_n)$$

logikai egyenletnek az  $X_i$  változótól való függését a

$$D_i F = \frac{dF}{dX_i} = F(X_1, X_2, \dots, X_i, \dots, X_n) \oplus$$

$$\oplus F(X_1, X_2, \dots, \bar{X}_i, \dots, X_n)$$

kifejezésekkel definiált Boole-differencia határozza meg ( $A \oplus$  szimbólum a kizáró VAGY, illetve modulo 2 összeadás jele). A Boole-differencia valamely  $X_i$  ( $1 \leq i \leq n$ ) változóban bekövetkező  $X_i \rightarrow \bar{X}_i$  változás hatását vizsgálja.  $D_i F$  a következőképpen értékelhető ki:

a) Ha  $D_i F = \frac{dF}{dX_i} = 0$ , akkor a hálózatban előforduló egyetlen hiba nincs hatással a kimenetre, a hálózat  $X_i$ -vel szemben invariáns.

b) Ha  $D_i F = \frac{dF}{dX_i} = 1$ , akkor a hálózat kimenete mindig hibás, ha  $X_i \rightarrow \bar{X}_i$  hiba előfordul.  $F$  feltétel nélkül függ  $X_i$ -től.

c) Ha  $D_i F = \frac{dF}{dX_i} = \varphi_i(X_1, \dots, X_n)$ , akkor a hálózat kimenete csak akkor függ  $X_i$ -től, ha:

$$\varphi_i(X_1, \dots, X_n) = 1.$$

Ez az utóbbi logikai egyenlet határozza meg a bemenő változók azon kombinációját, amely mellett  $X_i \rightarrow \bar{X}_i$  hiba a kimeneten kimutatható.

Egyetlen hiba előfordulásának esetére  $D_i F$  könnyen általánosítható. Az eddigiekben kizárólag valamely bemenő változó hatását vizsgáltuk. Ez önmagában is fontos eset, hiszen egy logikai rendszer valamely részhálózatának kimenő jele egy vagy több további hálózat rész bemenetét szolgál az esetek nagy részében. Különös jelentősége lehet ennek a vizsgálatnak abban az esetben is, ha a logikai rendszer párhuzamos működésű, és abban egy vagy több segédváltozó terjed. Erre tipikus példa a rekurzív átvitelképzéssel működő párhuzamos üzemi összeadó egység, amelyben a helyértékek egy adott csoportjára kombinációs hálózat — az összeadás elvégzése előtt — határozza meg az egyes helyértékekhez tartozó átviteli impulzusokat. Az előre meghatározott átviteli impulzusok az egyes helyértékeken az az összeadást elvégző részhálózat bemenő jelei. További példák: átvitel-terjedésen alapuló információfeldolgozó hálózatok (számlánc, összeadó egység, paritásvizsgáló stb.), részekre osztott, kaszkád működésű kombinációs hálózatok stb.

Egyetlen bemenő változóban előforduló hiba kiterjeszhető egyetlen kapu áramkörben fellépő hiba hatásának analizésére is.

Pl. Három bemenetű összeadó  $i$ -edik helyértékén keletkező  $S_i$  összeg és  $C_i$  átvitel, ha az operandusok  $i$ -edik helyértéken levő számjegye  $A_i$ , illetve  $B_i$ :

$$S_i = A_i \oplus B_i \oplus C_{i-1}; \quad C_i = A_i B_i + (A_i + B_i) C_{i-1}$$

Ha  $G_i = A_i B_i$  kapu állítja elő az átvitel keletkezését adó jelet,  $T_i = A_i + B_i$  kapu pedig az átvitel terjedést, akkor:

$$S_i = T_i \bar{G}_i \oplus C_{i-1} \quad \text{és} \quad C_i = G_i + T_i C_{i-1}$$

Az alábbi Boole-differenciák határozzák meg a  $G_i$ , illetve  $T_i$  kapu áramkörök egyikében előforduló hiba hatását a hálózat kimeneteire:

$$\frac{dS_i}{dT_i} = \bar{G}_i; \quad \frac{dS_i}{dG_i} = T_i; \quad \frac{dC_i}{dT_i} = \bar{G}_i C_{i-1}; \quad \frac{dC_i}{dG_i} = \bar{T}_i C_{i-1}$$

A Boole-differencia algebrai eszközökkel a definíció alapján közvetlenül meghatározható. Számítógép segítségével történő meghatározás logikai egyenletet megoldó programrészt kíván.

Egyszerűsíthető a közvetlen meghatározás az alábbi összefüggés felhasználásával:

$$D_i F = \frac{dF}{dX_i} = F(X_1, X_2, \dots, 1, \dots, X_n) \oplus F(X_1, X_2, \dots, 0, \dots, X_n)$$

minden  $X_i$ -re,  $1 \leq i \leq n$  esetén. Számítógép segítségével történő kiszámítás ebben az esetben is logikai egyenletet megoldó programrészt kíván.

Igazolható, hogy az ismert minimalizálási eljárások felhasználásával is előállítható a Boole-differencia.

Pl. Veitch-diagram alapján ez úgy történhet, hogy egy-egy minterm táblán ábrázoljuk  $F(X_1, \dots, X_i, \dots, X_n)$  illetve  $F(X_1, X_2, \dots, \bar{X}_i, \dots, X_n)$  logikai függvényt, majd a két minterm táblából  $D_i F$  számára egy harmadikat készítünk. A  $D_i F$ -et ábrázoló minterm tábla 1-es termjei ott vannak, ahol az előző két tábla termjei antivalensek. Számítógépen történő előállítás ezen módszere a szokásos minimalizálási eljárások kismérvű továbbfejlesztését kívánja, az antivalens termek előállítására.

Tisztán algebrai síkon történő előállítás érdekében az  $F$  függvény részekre is bontható, és az egyes részek Boole-differenciájának ismeretében határozható meg az eredő Boole-differencia.

Kettős Boole-differencia szolgál a logikai rendszerben előforduló kettős hiba detektálására. Kettős hiba esetén a következő eseteket kell megvizsgálni: a) a két hiba együttesen fordul elő; b) a két hiba vagy az egyik, vagy a másik hibaok valamelyikében lép fel, de egyidőben csak az egyik hibaok létezik; c) a két hiba vagy az egyik, vagy a másik hibaok miatt fordul elő, vagy mindkettő egyszerre fellép.

Egyszerűség kedvéért logikai hálózat  $X_i$ , ill.  $X_j$  bemenő változójában fellépő  $X_i \rightarrow \bar{X}_i$ , ill.  $X_j \rightarrow \bar{X}_j$  változás legyen a hiba oka. Vizsgáljuk meg ennek hatását  $F(X_1, X_2, \dots, X_i, \dots, X_j, \dots, X_n)$  logikai függvényre.

Ha  $X_i \rightarrow \bar{X}_i$ ;  $\bar{X}_j \rightarrow X_j$  változás egyidejűleg lép fel, vagyis a két hibaok ÉS kapcsolatban van:

$$D_{ij} F = \frac{d^2 F}{d(X_i X_j)} = F(X_1, \dots, X_i, \dots, X_j, \dots, X_n) \oplus F(X_1, \dots, \bar{X}_i, \dots, \bar{X}_j, \dots, X_n)$$

kifejezéssel definiált kettős Boole-differencia segítségével vizsgálható a hiba hatása. Az előzőekhez hasonlóan  $D_{ij} F$  kiértékelésével határozható meg, hogy

invariancia, a feltétel nélküli függés, illetve feltételes függés esete fordul-e elő.

Ha  $X_i \rightarrow \bar{X}_i$ ,  $X_j \rightarrow \bar{X}_j$  változás egymást kizáró VAGY kapcsolatban van, vagyis a két hibaok egyidőben nem léphet fel, akkor a

$$D_{i \oplus j} F = \frac{d^2 F}{d(X_i \oplus X_j)}$$

lényegében egyváltozós Boole-differencia segítségével analizálható a hiba.

Kimutatható, hogy a hiba-detekció visszavezethető a két hibaok külön-külön történő vizsgálatára a következőképpen:

$$D_{i \oplus j} F = \frac{d^2 F}{d(X_i \oplus X_j)} = \frac{dF}{dX_i} + \frac{dF}{dX_j}$$

Ha a két hibaok közül vagy az egyik, vagy a másik, vagy mindkettő egyszerre léphet fel, akkor a két hibaok vagy kapcsolatban van, hatásának vizsgálatára

$$D_{i+j} F = \frac{d^2 F}{d(X_i + X_j)}$$

kettős Boole-differencia alkalmazható. Kimutatható, hogy a hiba-detekció ebben az esetben az egyidejűleg fellépő és az egymást kizáró kettős hiba analízisére vezethető vissza az alábbi módon:

$$D_{i+j} F = \frac{d^2 F}{d(\bar{X}_i + X_j)} = \frac{d^2 F}{d(X_i + X_j)} + \frac{d^2 F}{d(X_i X_j)}$$

I R O D A L O M

- [1] D. B. Armstrong: On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets. IEEE Trans. on El. Computers, Vol. EC—15, No. 1. pp. 66—73, 1966.
- [2] Bohus M.: Statikus és dinamikus hazard vizsgálat a szinkron működésű logikai hálózatokban. Mérés és Automatika. XVIII. és 4—5. szám 121—126. old. 1970.
- [3] Bohus M.: Digitális rendszerek tervezése. Tankönyvkiadó Budapest 1969.
- [4] Bohus M.: Elektronikus számítógépek. Tankönyvkiadó, Budapest, 1966.
- [5] Bohus M.—Csupák Gy.—Géjther L.—Halász E.—Németh G.—Trón T.—Varró L.: A LOGAN logikai szimulációs program. Híradástechnika (ebben a számban).
- [6] H. Y. Chang—W. Thomis: Method of Interpreting Diagnostic Data for Locating Faults in Digital Machines. BSTJ. Vol. 46. pp. 289—317. February 1964.
- [7] H. Y. Chang: An Algorithm for Selecting an Optimum Set of Diagnostic Tests. IEEE Trans. on El. Computers, Vol. EC—14, No. 5. pp. 706—711. October 1966.
- [8] Csurgay Á.—Géjther K.—Házman I.: Helyzetkép a hálózatelmélet fő fejlődési irányairól. MTA Műszaki Tudományok osztálya, Távközlési Rendszerek Bizottság. 24. old. 1971.
- [9] R. W. Downing—J. S. Nowak—L. S. Tuomenoksa: No. 1. ESS Maintenance Plan. BSTJ. Vol. 43. pp. 1961—2020. September 1964.
- [10] Drasny J.: Automatikus tervezés a Számítástechnikai Koordinációs Intézetben. Híradástechnika (ebben a számban).
- [11] P. C. Garton—S. P. O'Byrne: Applications of Logic Simulation in Large Real Time Systems. IEE Conference Publication No. 51. pp. 51—58. 1969.
- [12] Géjther K.: Számítógép programok katalógusa 1969. Híradástechnika, XXI. évf. 6. szám. 178—184. old.

- [13] *K. Géher*: Theory of Network Tolerances. Akadémiai Kiadó, Budapest, 1971.
- [14] *Géher K.*: Számítógép programok katalógusa 1970. Híradástechnika, XXII. évf. 8. szám. 246—255. old.
- [15] *G. G. Hays*: Computer-Aided Desing: Simulation of Digital Design Logic. IEEE Trans. on Computers, Vol. C—18, No. 1. pp. 1—10. January 1969.
- [16] *G. Insinga*: Simulazione de reti logiche mediante calcolatore numerieo, Alta Frequenza, Vol. XXXIX. No. 3—4. 1970.
- [17] *A. A. Kaposi*: Logic Testing by Simulation. IEE Conference Publication No. 51. pp. 31—40. 1969.
- [18] *W. H. Kautz*: Fault Testing and Diagnosis in Combinational Digital Circuits. IEEE Trans. on Computers, Vol. C—16. pp. 353—366. April 1968.
- [19] *J. B. Krushal—E. R. Hart*: A Geometric Interpretation of Diagnostic Data from a Digital Machine, Based on a Study of the Morris Illinois Electronic Central Office. BSTJ Vol. 45. pp. 1299—1338. October 1966.
- [20] *F. Lee*: An Automatic Self-checking and Fault Locating Method. IRE Trans. on El. Computers, Vol. EC—11, pp. 649—654. October 1964.
- [21] *F. Lerailliez—A. Sarre—B. Waterlot*: CRISMAS: a Tool for Conception, Realisation, Implementation and Simulation of Sequential Synchronous Machines, IEEE Conference Publication 51. pp. 59—71. 1969.
- [22] *C. S. Lorens*: Invertible Boolean Functions, IEEE Trans. on El. Computers, Vol. EC—13, No.5. pp. 529—541. October 1964.
- [23] *P. N. Marinos*: Derivation of Minimal Complete Sets of Test-input Sequences Using Boolean Differences. IEEE Trans. on Computers. Vol. C—20. pp. 25—32. January 1971.
- [24] RACAL Research Limited: REDAP 22. Simulation of an Interconnected System of Logic Elemenst (SISLE). REDAC User's Manual, Vol. 2. 1969.
- [25] *J. P. Roth*: Diagnosis of Automatic Failures: A Caculus and a Method. IBM J. Res. and Dev. Vol. 10. pp. 278—291. July 1966.
- [26] *J. P. Roth—W. G. Bouricius—P. R. Schneider*: Programmed Algorithmus to Compute Test to Detect and Distinguish Between Failures in Logic Circuits. IEEE Trans. on El. Computers. Vol. EC—16. No. 5. pp. 567—580. October 1967.
- [27] *F. F. Sellers—M. Y. Hsiao—L. W. Bearnson*: Errordetecting Logic for Digital Computers, Mc Graw-Hill, New York 1968.
- [28] *Sebestyén B.*: Számítógép-bázisú automatikus ellenőrző rendszerek. Mérés és Automatika 172—176 és 270—274. old. 1970.
- [29] *D. J. K. Wise*: LIDO-An Integrated System for Computer Layout and Documentation of Digital Electronics. IEE Conference Publication No. 51. pp. 72—81. 1969.