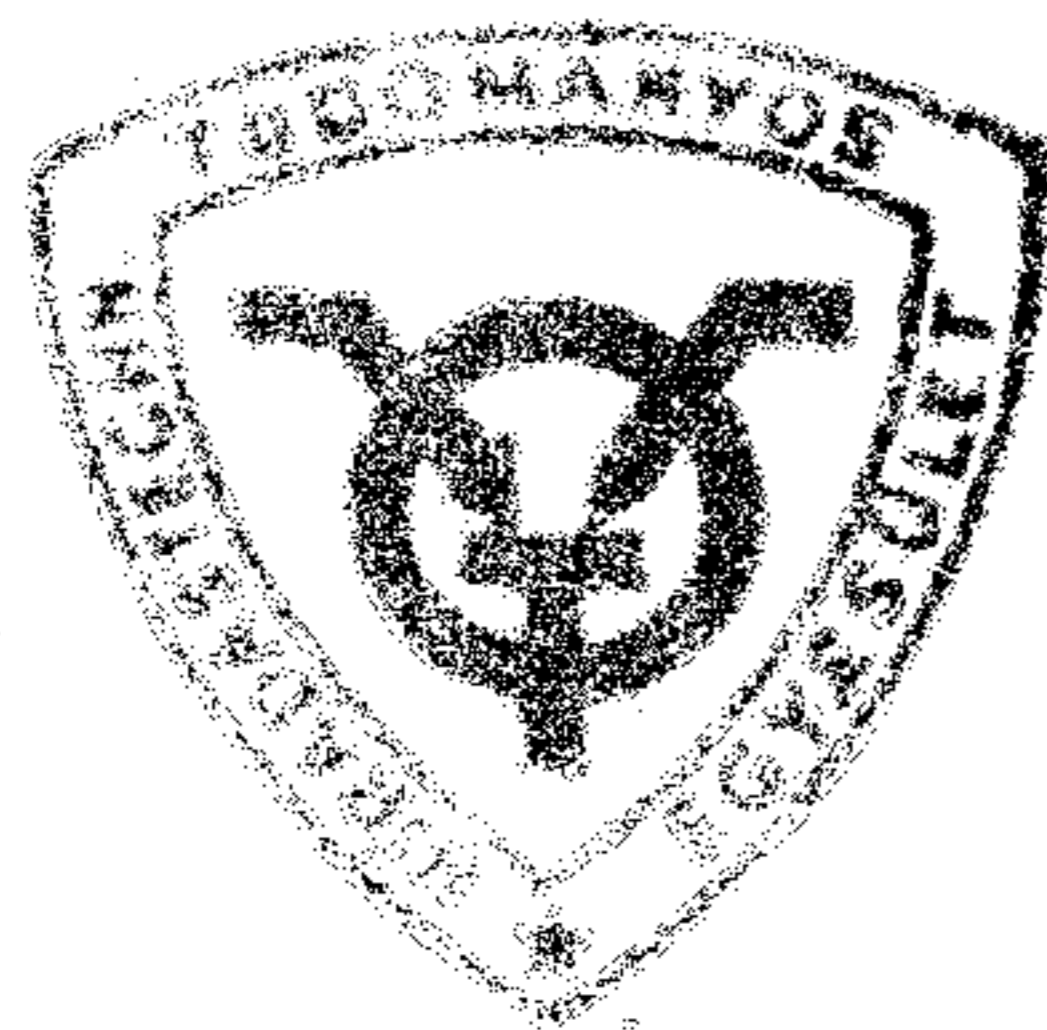


# HÍRADÁS- TECHNIKA

7



# HÍRADÁS- TECHNIKA

---

A HÍRADÁSTECHNIKAI TUDOMÁNYOS EGYESÜLET LAPJA

---

## TARTALOM

|   |                         |
|---|-------------------------|
| DRASNY JÓZSEF: Számítógépek automatizált tervezése .....  | 193                     |
| BOHUS MIKLÓS—DR. GÉHER KÁROLY: Logikai hálózatok számítógépes vizsgálata .....  | 199                     |
| BOHUS MIKLÓS—CSOPAKI GYULA—GEFFERTH LÁSZLÓ—HALÁSZ EDIT—DR. NÉMETH<br>GÁBOR—TRÓN TIBOR—VARRÓ LÁSZLÓ: A LOGAN logikai szimulációs program ..... | 204                     |
| DRASNY JÓZSEF—KOVÁCS MIKLÓS—MEZEY GYULA: Nyomtatási rajzok készítése a TAL—1<br>programmal .....  | 210                     |
| KOSZÓ GÁBOR—SZIRAY JÓZSEF: Hátoldali huzalozás tervezése. A TAL—2 program .....   | 215                     |
| Szaktmérnöki szakok indítása .....  | 220                     |
| Szemle .....  | 198, 203, 209, 214, 221 |
| Tartalmi összefoglalók .....  | 222                     |
| Обобщения .....   | 222                     |
| Zusammenfassungen .....   | 223                     |
| Summaries .....   | 223                     |
| Résumés .....   | 224                     |

Szerkesztőség: BOGLÁR GYULA főszerkesztő, SZÖLLŐSI GYÖRGYNE szerkesztőségi titkár, BALOGH PÁL, DR. SÁRKÖZI GÉZA kandidátus és MAY PÉTER tudományos szerkesztők, DR. FLESCHE ISTVÁN, DR. RUPPENTHAL PÉTER szerkesztőségi munkatársak. — A szerkesztőség címe: Budapest II., Mártírok útja 85. I. em. 140. Telefon: 154-859 — A Híradástechnikai Tudományos Egyesület címe: Budapest V., Szabadság tér 17. Telefon 113-027

Szerkeszti a szerkesztő bizottság

**INDEX: 25.375**

## HÍRADÁSTECHNIKA

Kiadja a Lapkiadó Vállalat, Budapest VII., Lenin körút 9–11. Telefon: 221-285. Felelős kiadó: SALASÁNDOR igazgató. Terjeszti a Magyar Posta. Előfizethető bármely postahivatalnál, a kézbesítőknél, a Posta hírlapüzleteiben és a Posta Központi Hírlap Irodánál (KHI, Budapest V., József nádor tér 1.) vagy közvetlenül postautalványon, valamint átutalással a KHI 215—96162 pénzforgalmi jelzőszámra. Előfizetési díj: félévre 36 Ft, egész évre 72 Ft. Egyes szám ára: 6 Ft. Megjelenik havonta. A folyóirat külföldre előfizethető: „KULTURA” P. O. B. 149 Budapest, 62.

72.7918 Egyetemi Nyomda, Budapest. Felelős vezető: JANKA GYULA igazgató

DRASNY JÓZSEF

Számítástechnikai Koordinációs Intézet

## Számítógépek automatizált tervezése

ETO: 681.3.001.2;681.3.06

Napjainkban a számítógépek a legbonyolultabb műszaki berendezések közé tartoznak, és így tervezésük igen sok szellemi munkát igényel. Ugyanakkor a számítógépek térhódítása a szellemi munka gépesítése területén egyre növekszik. Kézenfekvő a megoldás: használjunk számítógépeket a számítógépek tervezésére, legyen a tervezés automatizált.

Az elmúlt évtizedben elsősorban a nagy számítógépgyártó cégek (IBM, ICL) dolgoztak ki programrendszereket a tervezés automatizálása céljából. Ezek a programok a cég saját konstrukcióinak tervezésére készültek; másirányú felhasználásra nincs mód, elsősorban a titoktartás, másrészt a speciális alkalmazási terület miatt. A 60-as évek végén megjelenő software-cégek viszont már eladásra és főleg felhasználásra kínálnak olyan programokat, amelyek a tervezés egy-egy részfeladatát meg tudják oldani. A legtöbb ilyen program áramkörök analízisével és szintézisével, valamint nyomtatott áramkörök tervezésével és rajzolásával kapcsolatos.

Magyarországon az elektronikus áramkörök számítógépes tervezésére komoly és megalapozott tevékenység indult meg, és már számottevő eredmények vannak ezen a területen. A programokat a készítő intézmények elsősorban saját céljaikra készítették, de foglalkoznak azok értékesítésével is (TKI, MIKI). A hazai számítógépgyártás terén mostanáig hiányoztak az adottságok arra, hogy egy nagyobb tervezési szakasz automatizálása megtörténjen, de igények sem jelentkeztek erre különösebben. A most végrehajtás alatt álló számítástechnikai program viszont lehetővé és sürgetővé teszi ilyen programrendszer létrehozását. Jelenleg több helyen is folyik a számítógépek konstrukciós tervezését elősegítő programrendszer kidolgozása, de sehol sem a gyártó cégnél. Ennek oka az, hogy a gyártás terén hagyományokkal rendelkező intézményeknek nincs erre a célra elegendő kapacitásuk.

A hazai helyzet specialitása továbbá, hogy több vállalatnál készülnek digitális berendezések azonos konstrukciós elvek szerint, ami lehetővé teszi az elkészült programok több helyen történő hasznosítását.

Tulajdonképpen ez teszi vagy teheti kifizetődővé a programok kidolgozói számára a rendkívül nagy munka- és gépidő-befektetést.

E cikk beszámol a Számítástechnikai Koordinációs Intézetben kidolgozás alatt álló és részben már megvalósított programrendszer általános tulajdonságairól. Ismerteti a teljes tömbvázlatot, és magyarázatot fűz az egyes műveletekhez.

A programok kipróbálásához és futtatásához rendelkezésre áll az SzKI Siemens 4004/45-ös számítógépe. Ennek operatív tárkapacitása 128 kbyte, ezenkívül 4 mágnesszalag és 3 mágneslemez-tároló is tartozik a konfigurációba. Az 1972-ben várható bővítés következtében a lemezes tárolók száma 4-re vagy 5-re növekszik és ez már teljes mértékben meg fog felelni a programrendszer által támasztott összes igényeknek.

### 1. Általános szempontok

A programrendszer kialakításakor alapvető szempont az volt, hogy az egész konstrukciós tervezést vagy annak lehetőleg nagy, összefüggő szakaszát lefedje. Abban az esetben, ha a tervezési folyamatban kézi és gépi szakaszok váltogatják egymást, a gépi módszerek erősen vesztenek hatékonyságukból. A két tervezési mód más-más formában igényli a bemeneti adatokat, és másféle formában adja az eredményeket. A különböző módszerekkel végrehajtott szakaszok közti átmeneteknél felesleges munkaként jelentkezik a szükséges formai átalakítás, amely még pótlólagos hibák forrása is lehet. Ennek elkerülése végett olyan tervezési munkákat is automatizálni kell, amelyek a tervező számára nem jelentenek különösebb problémát vagy nem igényelnek hosszabb időt. A kitűzött közvetlen cél az, hogy egy adott integrált áramköri rendszer (Texas SN 74 N) báziselemeiből felépített logikai terv alapján számítógéppel lehessen előállítani a nyomtatott áramköri lapok (kártyák) és a kártyákból összeállított kártyablokkok gyártási dokumentációját, az üzemeltetési dokumentációk közül pedig azokat, amelyeket géppel lehet elkészíteni, és meg lehessen oldani a kártyák és kártyablok-

kok gyártás utáni automatikus ellenőrzését. Mindezt úgy kell elérni, hogy valamennyi tervezési szakaszt automatizáljuk.

A mondottak ellenére egyes tervezési szakaszokat olyan formában kell programozni, hogy ezek a programok a teljes rendszertől függetlenül, önállóan is alkalmazhatók legyenek. Tipikus példa erre a nyomtatott áramköri lapok nyomtatásának a megtervezése. Ez a részprogram önállóan is felhasználható, aminek több előnye is van. Olyan kis berendezések kártyái is megtervezhetők, ahol a teljesen automatizált tervezésnek nem lenne értelme vagy lehetősége. A program már akkor is alkalmazható, amikor a teljes programrendszer még nincs készen. Ezért ezt a feladatot érdemes általánosabb esetekre is megoldani.

Az automatizált tervezés nem zárja ki, sőt határozottan igényli a tervezés menetébe való emberi beavatkozást. A programok nagy része heurisztikus eljárásokon alapszik, és így sok esetben nem lehet jobb eredményt kapni az ember által elérhetőnél. Ilyenkor szükség van az eredmények módosítására, figyelembe véve az első lépésben kapott megoldást. Számos esetben nem lehet teljesen hűen algoritmizálni az egyes tervezési szakaszokat, sokszor pedig el kell hagyni bizonyos szempontokat a megoldás optimalizálásakor, mert a program túl hosszú vagy túl bonyolult lenne. A program által ilyen okokból elkövetett hibákat ki kell javítani. A legkedvezőbb módszer a számítógéppel való állandó, közvetlen, kétirányú kapcsolat létrehozása lenne, grafikus display és fényceruza segítségével. Megfelelő eszközök hiányában erről le kellett mondanunk. A jelenlegi megoldás szerint a program szakaszos felépítésű, és minden szakasz lefutása után lehetőség van a sornyomtatón kapott eredmények ellenőrzésére, a szükséges változtatások bevitelére és a programszakasz ismételt futtatására kedvezőbb eredmények elérése céljából.

Nemcsak a tervezés folyamán, hanem annak befejeztével is szükség lehet változtatások bevezetésére. A gyártás vagy az élesztés és bemérés során gyakran derülnek ki olyan hibák, amelyek csak a tervek módosításával, esetleg csak a logikai terv megváltoztatásával küszöbölhetők ki. Ilyen esetekben túl költséges lenne a teljes programrendszert újra lefuttatni, ezért külön programrészek gondoskodnak a változás hatálya alá eső tervek módosításáról és a változtatások következetes bevezetéséről a gépen belül és a dokumentációkban.

A programok igen sok bemeneti adatot igényelnek és sok a kimeneti adat is. Az adatok a legkülönbözőbb formában jelentkeznek és a gép különböző formában adja ki ezeket (rajzok, táblázatok, szövegek). A magas szintű programozási nyelvek közül a FORTRAN nyújtja ezen a téren a legkedvezőbb leírási módot, ezért célszerű ezt a nyelvet használni a programok megírásához. Sok esetben szükség van azonban alacsonyabb szintű nyelv használatára is; ezeket a részeket Assembler nyelven lehet írni.

## 2. Folyamatábra

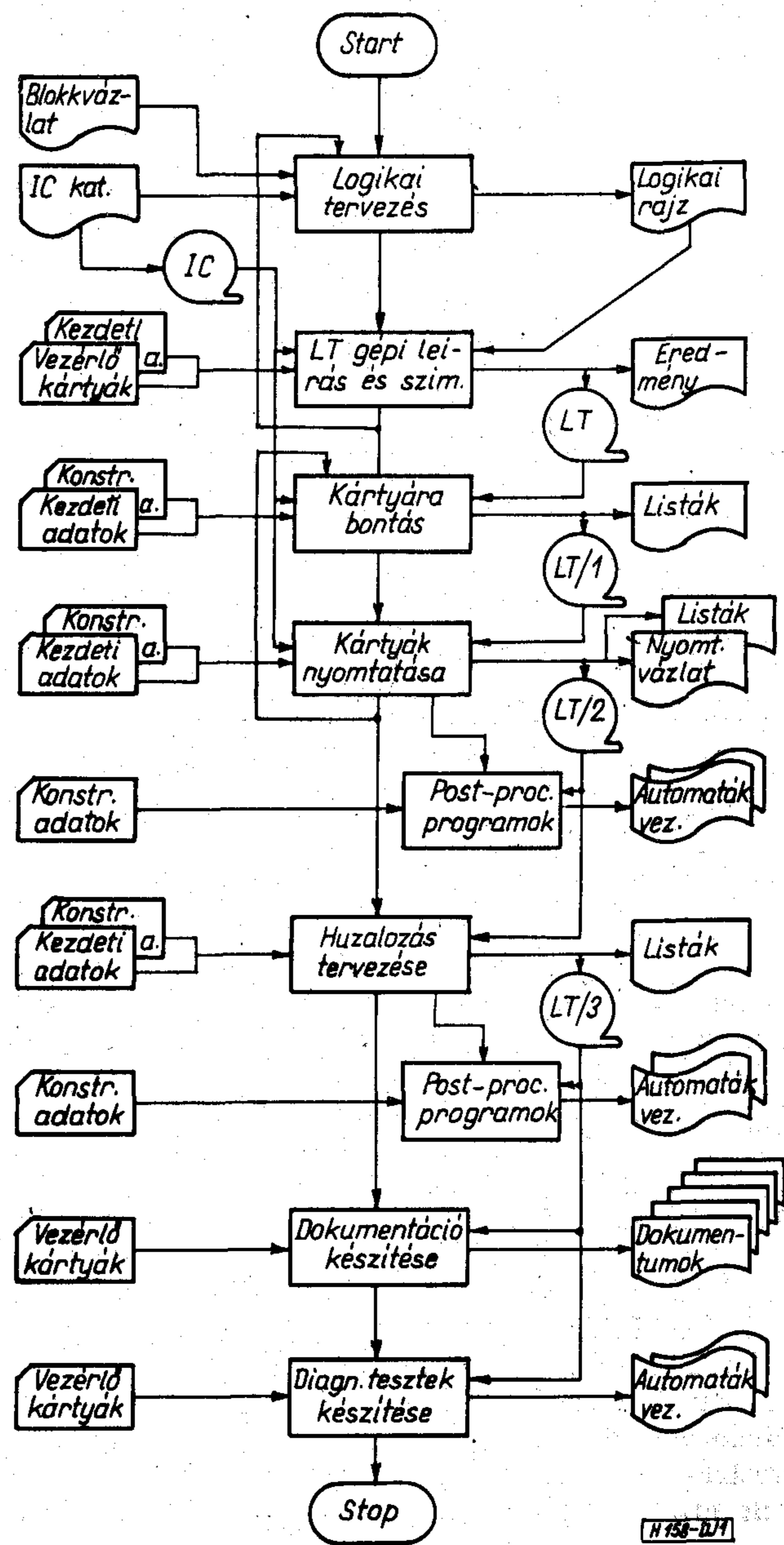
A teljes programrendszer folyamatábrája az 1. ábrán látható. Ezen külön blokkban tüntettük fel a nagyobb programrészeket, amelyek valamely jól

definiálható tervezési szakasszal kapcsolatosak. Egy-egy blokk megkapja a logikai terv gépen belüli leírását a már kiszámított valamennyi adattal együtt, ezenkívül a tervezőnek kell külön megadnia a feladat megoldásához szükséges konstrukciós adatokat, a tervező előírásait képviselő kezdeti adatokat és a program futását vezérlő adatokat, ha az nem rögzített módon történik.

Az ábrán csak a legfontosabb és jellemző bemeneti és kimeneti adatokat tüntettük fel, úgyszintén az egyes blokkok közti kapcsolatokat is leegyszerűsített formában jelöltük.

### 2.1 Logikai tervezés

A logikai tervezés szakasza a folyamatábrán csak a teljesség kedvéért szerepel, mivel a logikai terv előállítása hagyományos módon történik. A logikai terv általában és itt is feltételezeten logikai rajz formájában készül el, a digitális rendszer strukturális



1. ábra. A programrendszer folyamatábrája

tervéből kiindulva. A tervezésnél figyelembe kell venni a realizáláshoz megadott integrált áramköri készletet (IC katalógus), továbbá más, itt nem részletezett előírásokat (specifikáció, szabványok stb.).

## 2.2 A logikai terv gépi leírása és szimulációja

Ahhoz, hogy egy logikai rendszer konstrukciós tervét elkészíthessük, szükség van a logikai tervnek a számítógépen belüli leírására, vagyis a logikai terv (LT) modelljének a felépítésére. Ez a modell fog kiindulási alapul szolgálni a további tervezéshez.

A logikai tervet, amely logikai rajzként áll rendelkezésre, számítógépes feldolgozásra alkalmas formára kell átalakítani. Ennek az előkészítő munkának az egyik szakasza a logikai rajz leírása, azaz az ismertetésre kerülő programrendszer esetében a táblázatos formára való átalakítás. Az átalakítás meghatározott szabályok szerint történik, ennek során a logikai terv adatait adatlapokra kell rávezetni. Ezután az adatlapok tartalmát lyukkártyára kell lyukasztani és a számítógéppel beolvastatni.

Az előkészítéshez tartozik még egy integrált áramköri katalógus gépen belüli létrehozása is. Ezt külön program készíti el, amelyet az ábrán nem tüntettünk fel. Az IC katalógus jelenleg a Texas SN 74 N sorozat 20 leggyakrabban használt típusát tartalmazza, de lehetőség van tetszőleges további bővítésekre.

Az adatkártyák beolvasása során megtörténik az adatok szintaktikai ellenőrzése is. A program az adatmegadás hibáit ellenőrzi, és az elkövetett hibákat sornyomtatón kijelzi. Adathiba esetén a program futása leáll és a hibák kijavítása után újra kell kezdeni a beolvasást.

Ha a program a szintaktikai ellenőrzés során nem talált hibát, akkor hozzáfog az alapadatok és a leírt kapcsolás ellenőrzéséhez. Felismeri és kijelzi a következő hibákat:

- két különböző jel ugyanazon az érintkezőn,
- bekötési rendellenességek,
- tilos összekötések,
- aszinkron hálózatot eredményező visszacsatolás.

Az áramkör ellenőrzése után újra kell kezdeni a futtatást, ha a program hibát jelzett, és azokat a tervező kijavította. A hibajegyzékből látható, hogy a program az aszinkron csatolásokat hibának tekinti, és nem tudja kezelni. Az ellenőrzés és a szimuláció csak szinkron szekvenciális hálózatokra alkalmazható.

A kétszeres ellenőrzés után a program a logikai hálózat adatait a számíthatóság szempontjából alkalmas formába rendezi és így tárolja. A logikai tervnek ezt a formáját tekintjük alapnak a további számításokhoz, ez a logikai terv gépi leírása, a logikai terv modellje (LT).

A logikai terv modelljének birtokában a program különböző áramköri vizsgálatokat tud végezni, és megoldható a működés szimulálása is. Ezeknek a vizsgálatoknak a célja az, hogy a kapcsolásban és a működésben rejlő hibák még a konstrukciós tervezés megkezdése előtt kiderüljenek. Így elkerülhető a hibás tervek realizálása, a legyártott berendezésen való hibakeresés művelete, valamint a hibás konstrukciós tervek megvalósításához szükséges munka.

Az áramköri vizsgálatok során vezérlő kártyákkal lehet kérni bizonyos feladatok elvégzését. A tervező így maga választja meg, hogy milyen adatokat kér és milyeneket nem. Bizonyos vizsgálatokhoz kezdeti adatokat kell megadni, ez is a tervező feladata. Az elvégezhető vizsgálatok a következők:

- A megépítéshez szükséges IC tokok számának meghatározása típusonként és összesítve.
- Minden báziselem tényleges terhelésének meghatározása. A túlterhelt elemek kijelzése. A szabad kollektoros kimenetekre csatlakozó ellenállások értékének kiszámítása.
- Késleltetési idő kiszámítása bármely két megadott pontpár között, valamennyi útra nézve.
- Logikai egyenlet felírása bármely adott pontra.
- Azon báziselemek kiírása, amelyek kimenetén a logikai szint értéke a vezérléstől függetlenül állandó.

A vizsgálatok eredményét a program sornyomtatón kiadja, a hibákat külön kijelzi.

A logikai szimuláció elvégzését szintén a tervező kérheti egy vezérlő kártya segítségével. A szimuláció során megállapítható bármely csomóponton lévő logikai jel értéke a bemenőjelek és a tárolók belső állapotának tetszőleges kombinációja esetén. Kezdeti adatként szükség van a tároló elemek kezdeti állapotára és a bemeneti változók értékére. Bemeneti jelsorozatok megadása esetén a program eredménye a vizsgált pontokon lévő jelek idődiagramja.

A szimuláció után a tervező — a vizsgálatok eredményeinek felhasználásával — eldöntheti, hogy szükséges-e a logikai terven változtatást végrehajtani. Ha igen, akkor a folyamatban a megfelelő helyen, valószínűleg még a logikai tervben el kell végeznie a megfelelő módosításokat. Ha nem, akkor el lehet kezdeni a konstrukciós tervezési feladatokat.

A szimulációs eljárásokról és a programról e szám más helyein részletes ismertetés található [9], [10].

## 2.3 Kártyára bontás

A logikai tervet báziselemek összekötött halmaza alkotja. A konstrukciós tervezés során a báziselemeket IC tokokban kell elhelyezni, az IC tokokból kártyákat, a kártyákból kártyablokkokat kell összeállítani. A tárgyalt tervezési programrendszer az egy kártyablokknál nem nagyobb konstrukciós egységben elhelyezhető logikai rendszerek tervezését oldja meg. Ez összhangban van a várható hazai fejlesztésekkel, mivel a kifejlesztésre kerülő berendezések túlnyomó többsége nem tartalmaz ezer IC toknál többet, és így nem haladja meg a kártyablokk méretét. Az ennél nagyobb berendezéseket kézi úton részletekre lehet bontani, és azután az egyes részekre alkalmazható a tárgyalt tervezési eljárás.

A konstrukciós tervezés első feladata a logikai rendszer olyan részekre történő felbontása, amelyek egy kártyán elhelyezhetők. A cél az, hogy minél kevesebb kártyára legyen szükség, ezen belül pedig az, hogy a kártyák közti összekötő vezetékek száma minimális legyen.

A számítógépes tervezésnél nem vettük figyelembe azokat a szempontokat, amelyek kézi tervezésnél mindig szerepet játszanak: a felbontás funkcionális

egységek szerint történjék, és lehetőleg minél több azonos felépítésű kártya legyen. A tervező azonban adhat meg kezdeti előírásokat egyes kártyák tartalmára vonatkozóan, és így irányíthatja a tervezés menetét.

A programban állandó korlátozó feltételként szerepelnek a következők:

- Az egy kártyán lévő báziselemek teljesen kitöltött IC tokokat eredményezzenek. Részlegesen kitöltött tok a teljes rendszerben csak típusonként egy darab lehet.
- Az egy kártyán lévő IC tokok száma nem haladhat meg egy adott értéket.
- Az egy kártyáról kimenő vezetékek száma nem haladhat meg egy adott értéket.

Ezek a korlátozások a konstrukciós adatokból (kártyaméret, csatlakozótípus), továbbá az IC katalógus adataiból adódnak.

A program eredménye sornymutatón jelenik meg és azokat a tervező vagy módosítja, vagy elfogadja. Az utóbbi esetben a programot tovább kell folytatni és ekkor az újonnan számított adatok kerülnek be a logikai terv leírásába. Így jön létre a logikai terv egy bővített modellje, az LT/1 modell.

#### 2.4 Kártyák nyomtatása

A kártyára bontás eredményeként kiadódott az egy kártyára kerülő báziselemek listája, az LT/1-ből így megállapítható a kártyán belüli összekötések és a kártyáról kimenő jelek leírása is. Ezek alapján elvégezhető a kártyák konstrukciós tervezése:

- az IC tokok összeállítása báziselemekből,
- az IC tokok helyének meghatározása,
- az IC tokok közti összekötések nyomvonalának megtervezése, kétoldalas nyomtatott lapokat feltételezve.

A program kiindulásként megkapja az LT/1 modellt és az IC katalógust, amelynek adatai gépen belül vannak tárolva. Lyukkártyán kell beadni a kártya kialakítására vonatkozó konstrukciós adatokat, valamint az IC tokok elhelyezésére és a nyomvonalak kijelölésére vonatkozó előírásokat és korlátozásokat (kezdeti adatok).

A program eredményeként kapott nyomtatási tervről sornymutatón vázlatot kaphatunk, és ennek alapján tud ismét a tervező dönteni a továbbfolytatásról vagy változtatásról. Előfordulhat, hogy egy kártyán belül sem a program, sem a tervező nem tudja valamennyi összekötést megvalósítani. Ebben az esetben nemcsak a nyomtatás tervezését, hanem a kártyára kerülő elemek számát is módosítani kell, azaz vissza kell térni a kártyára bontás műveletéhez.

A nyomtatás sikeres megtervezése után előállítható az LT/2 modell, amely az egyes kártyák konstrukciójára vonatkozó összes adatokat tartalmazza.

A nyomtatást tervező programról más helyen részletesebb leírás található, ezért itt nem térünk ki további részletekre [13].

#### 2.5 Post-processor programok (kártya)

A nyomtatási útvonalak kijelölésével a kártyák konstrukciós tervezése megoldódott, de a sornymutatón kiadott nyomtatási vázlat vagy nyomtatási

terv nem tekinthető gyártási dokumentációnak. A kártyák gyártásának megkezdéséhez mérethelyes film vagy méretarányos nyomtatási rajz (fotosablon) szükséges. A rajzot kézzel is el lehet készíteni a kapott topológiai elrendezés birtokában, de a munka meglehetősen hosszadalmas, és a pontosság is sok kívánnivalót hagy maga után.

Rajzgép alkalmazásával időt lehet megtakarítani, és a hiba is megfelelő értéken tartható. Az LT/2 modell adatai alapján bármely rajzgéphez elő lehet állítani a vezérlő információt akár lyukszalagon, akár mágnesszalagon.

Még 1970-ben elkészült a *TAL-1* nevű program, amely a ZUSE gyártmányú Graphomat rajzgéphez állítja elő a vezérlő lyukszalagot, kézzel tervezett nyomtatási vázlat adatai alapján. További post-processor program készíthető numerikus vezérlésű fúrógépek vezérlő szalagjainak az előállítására is.

#### 2.6 Hátoldali huzalozás

A nyomtatás tervezése során meghatározásra kerültek azok az érintkezők, amelyek a kártyákról kimenő jeleket közvetítik. Így az LT/2 modell ismeretében megállapítható, hogy mely kártyák mely érintkezőin van ugyanaz a jel, vagyis ismertek az ekvipotenciális érintkezők csoportjai. Ezek után megtervezhetők a kártyacsatlakozók közötti huzalos összekötések. A huzalozás tervezésekor két feladatot kell megoldani:

- a kártyák helyének meghatározása a kártyablokkban,
- a huzalos összekötések huzalvezetésének megtervezése, wire-wrap technológiát feltételezve.

A program mágnesszalagon megkapja az LT/2 modellt, lyukkártyán kell megadni a kártyablokk kialakítására vonatkozó konstrukciós adatokat és kezdeti adatként a tervező előírásait és korlátozásait a kártyák helyére és a huzalozásra vonatkozóan.

A program eredményeként a következő táblázatokat kapjuk:

- elhelyezési táblázat,
- összekötési táblázat.

Ezek alapján a tervező dönti el, hogy végez-e változtatásokat, vagy felvéteti az új eredményeket a logikai terv leírásába, és így létrehozza az LT/3 modellt.

A huzalozást tervező programmal és az alkalmazott eljárásokkal kapcsolatban utalunk a részletesebb leírást nyújtó irodalomra [14].

#### 2.7 Post-processor programok (kártyablokk)

A hátoldali huzalozás elvégzését és ellenőrzését egyre inkább automata és félautomata berendezések veszik át, különösen azokban a termelő üzemekben, ahol nagy darabszámban kell a berendezéseket előállítani. A félautomaták a huzalozást végző szerelő munkáját ellenőrzik és irányítják, az automaták magát a huzalozási műveletet is elvégzik. Vezérlésükre általában lyukszalagot használnak, amely tartalmazza a huzalozáshoz szükséges információkat. Ezeket a lyukszalagokat számítógéppel lehet előállítani.

Az ismertetés tárgyát képező programrendszerben az LT/3 modell tartalmazza a huzalozásra vonatkozó tervezési adatokat. A huzalozás kivitelezéséhez szükség lehet még más konstrukciós adatokra is, amelyek lyukkártyáról olvashatók be az automaták működtetéséhez szükséges vezérlő adatokkal együtt.

### 2.8 Dokumentáció készítése

Az LT/3 modell a konstrukcióra vonatkozó valamennyi olyan adatot tartalmazza, amely az egyes kártyák és a teljes kártyablokk gyártási dokumentációjának elkészítéséhez szükséges. A gyártási dokumentációk nagy része elkészíthető számítógép segítségével akár sornyomtatót, akár rajzgépet használunk a táblázatok és rajzok kivitelezéséhez.

A dokumentációt készítő programrendszer kidolgozása előtt szükség van természetesen az egyes dokumentumok tartalmi és formai követelményeinek a kidolgozására. Különösen fontos ez azon gyártási dokumentumok esetében, amelyek később üzemeltetési dokumentációként is szolgálnak. Ezeknél már nemcsak a gyártó vállalat előírásait kell figyelembe venni, hanem a megrendelő követelményeit is.

A dokumentációs rendszerrel szemben támasztott egyik alapvető követelmény az, hogy a változások könnyen bevezethetők legyenek. A programnak lehetővé kell tennie ezeknek a változtatásoknak következetes végrehajtását is.

A dokumentáció két fő csoportra osztható: kártya és kártyablokk dokumentációra. Itt felsoroljuk ezek közül azokat, amelyeknél lehetőség látszik a tartalmi és formai szabályok rögzítésére.

#### Kártyadokumentáció:

- logikai rajz (a báziselemekkel realizált logikai rajz, feltüntetve a helyes működéshez szükséges diszkrét elemeket is),
- ültetési rajz (az IC tokok és a diszkrét alkatrészek elhelyezési rajza),
- darabjegyzék.

#### Kártyablokk dokumentáció:

- idődiagramok,
- ültetési rajz (a kártyák elhelyezése a kártyablokkban),
- jeltáblázat (a jelek összes konstrukciós címei és rajzcímei),
- jelazonosítók táblázata (az összes jel felsorolása),
- realizációs táblázatok (a vezetékek listája a huzalok bekötési sorrendjében),
- darabjegyzék.

### 2.9 Diagnosztikus tesztek készítése

Az LT/3 modell szolgál alapul a kártyák és a berendezés gyártás utáni ellenőrzéséhez. Az ellenőrzés során megállapítható, hogy a kártya vagy kártyablokk jól funkcionál-e vagy sem, és információt kell kapni a hiba helyére vonatkozóan is.

Számítógépes módszerekkel kidolgozhatók olyan optimális bemeneti jelsorozatok (tesztek), amelyeket a kártya bemenetére adva és vizsgálva a kimeneti jeleket, megállapítható a hiba jelenléte és esetleg helye vagy környezete is. Az optimalitás elsősorban a

jelsorozat hosszára vonatkozik, azaz olyan minimális hosszúságú jelsorozatot kell kialakítani, amely a logikai hálózatban lévő valamennyi hibát képes kimutatni. Amennyiben a hiba helyét is meg akarjuk állapítani, úgy további vizsgáló jelekre van szükség, de ezek számát is minimalizálni kell.

Magát a vizsgálatot nem célszerű nagy számítógépen on-line üzemen végezni, de a tesztek generálásához szükség van a nagy gépre. Ezért a tesztprogramokat az előállítás után adathordozókra kell rávinni, és kis számítógépen vagy célgépen kell elvégezni velük a vizsgálatokat. A kis számítógép a tesztprogramot leolvassa, kidolgozza és kiadja a szükséges jeleket a vizsgált kártyára vagy berendezésre, érzékeli a kapott válaszjeleket, és összehasonlítja a jónak számított válaszokkal. Hiba esetén jelzést ad a hiba jelenlétéről vagy a vizsgálat módszerétől függően a helyéről és jellegéről is.

### 3. Strukturális szimuláció

Az 1. és 2. pontokban ismertetett programrendszer a logikai tervből kiindulva állítja elő a berendezés konstrukciós, technológiai és üzemeltetési dokumentációit feltételezve, hogy a logikai terv elkészítése hagyományos módon, kézzel történik. A tervezés korábbi szakaszaiban, a rendszertervezés és a strukturális tervezés során is van lehetőség azonban számítógépek igénybevételére.

A különböző formális nyelvek lehetővé teszik a rendszer- vagy strukturális szinten megtervezett berendezés működésének leírását, a nyelvekhez készült transzlátorok segítségével pedig a formális leírás átalakítható a számítógép saját belső utasításaira. A gépen belüli leírás birtokában ezek után a tervezendő berendezés működése szimulálható, sőt egy jól megszerkesztett formális nyelv lehetővé teheti a strukturális leírástól a logikai terv leírására történő átmenetet, azaz a logikai tervezés számítógépes megoldását.

Az SzKI-ban és az SzKI megbízásából a BME Vezetékes és Vezetéknélküli Híradástechnikai Tanszékein 1970 óta folyik a munka a Szovjetunióban kidolgozott formális nyelv, az OSzSz-2 transzlátorának elkészítése érdekében. Erről a témakörrel és az elkészült programokról a [11] és [12] közlemények számolnak be.

### 4. Megjegyzések

Az ismertetett tervezési programrendszer és az OSzSz-2 transzlátor kidolgozását az SzKI Tervezés Automatizálási Laboratóriuma és az SzKI-val kötött együttműködési szerződések keretében a BME Vezetékes Híradástechnika Tanszéke és a Vezetéknélküli Híradástechnika Tanszéke végzi.

A programrendszer jelenleg még kidolgozás alatt áll; egyes részei már elkészültek, mások készítése folyamatban vagy az előkészítés stádiumában van. Elképzelhető, hogy a jelenlegitől eltérő igények jelentkezése esetén vagy a fejlődés során egyes rész-megoldások megváltoznak. Ezért is igyekeztünk a

programok kialakításának általános szempontjait kiemelni és nem tárgyaltuk az egyes megoldási módokat.

A jelenlegi szám további közleményei a már megvalósított vagy befejezéshez közeli programokkal részletesen is foglalkoznak, és megfelelő információt nyújtanak a kivitelezésre és a felhasználásra vonatkozóan.

I R O D A L O M

- [1] Géher K.: Számológép programok áramkörök tervezésére. Híradástechnika, 1968. jún. 169—173. oldal.
- [2] Géher K.: Számítógép programok katalógusa 1968. Híradástechnika, 1969. aug. 238—251. oldal.
- [3] Géher K.: Számítógép programok katalógusa 1969. Híradástechnika, 1970. jún. 178—184. oldal.
- [4] Géher K.: Számítógép programok katalógusa 1970. Híradástechnika, 1971.
- [5] Géher K.—Csurgay Á.—Máté L.—Németh J.—Roska T. Sárossy J.—Szabó Lné.: Elektronikus áramkörök automatikus tervezése és gyártása. OMFB 16—803/3. Budapest, 1970.
- [6] International Conference on Computer Aided Design. Southampton, 1969. április 15—18. IEE Conference Publication, No. 51.
- [7] Special issue on computer aided design. Proc. IEEE. 1967. november.
- [8] Матюхин, Н. Я.: Применение вычислительных машин для проектирования цифровых устройств. Советское Радио, Москва, 1968.
- [9] Bohus M.: Szimbolikus nyelvek felhasználása a digitális rendszerek funkcionális és parametrikus szimulációjára. Híradástechnika, 1972. 6. sz.
- [10] Bohus M.—Németh G.—Trón T.—Varró L.: A SUBSET szimulációs nyelv digitális rendszerek funkcionális vizsgálatára. Híradástechnika, 1972. 6. sz.
- [11] Bohus M.—Géher K.: Logikai hálózatok számítógépes vizsgálata. Híradástechnika, ebben a számban.
- [12] Bohus M.—Németh G.—Geffert L.—Trón T.—Halász E. Csopaki Gy.—Varró L.: A LOGAN logikai szimulációs program. Híradástechnika, ebben a számban.
- [13] Drasny J.—Kovács M.—Mezey Gy.: Nyomtatási rajzok készítése a TAL—1 programmal. Híradástechnika, ebben a számban.
- [14] Koszó G.—Sziray J.: Hátoldali huzalozás tervezése. A TAL—2 program. Híradástechnika, ebben a számban.

S Z E M L E

Összeállította: BALOGH PÁL

LD 30 típusmegjelöléssel hozta forgalomba a Siemens cég az új GaAsP lumineszcens diódáját. Ezt mint optikai indikátort javasolják felhasználni a vastag- és vékonyréteg áramköröknél, valamint a diszkrét és integrált elemeket tartalmazó kapcsolási összeállításoknál. (Funktechnik, 1971. 16. sz.)

A Saunders-Roe Developments Ltd. betalignel elnevezéssel olyan fényforrást kínál, ami nem igényel tápforrást. Az új lámpa tulajdonképpen tríciumgázzal töltött üvegbúra, ami lényegesen kevesebb radioaktív sugárzást bocsát ki, mint az órák világító számlapja. A 20 év élettartamú fényforrás a legkülönbözőbb színekben, alakokban és méreteken előállítható. Műszerlapok és skálák megvilágítására éppen úgy alkalmas, mint fotometrikus kalibrálóberendezésekben referencia-fényforrásként. (Elektronik, 1971. 20. k. 1. sz.)

Az NDK-ban a növekvő munkaerőhiány következtében az elektronikus adatfeldolgozó berendezéseket elsősorban olyan feladatkörökben fogják alkalmazni, ahol ezáltal munkaerőt szabadíthatnak fel, illetve megakadályozhatják a munkaerőhiány további növekedését. Ez elsősorban az ügykezelés területére, a mérési- és ellenőrzési folyamatokra és a sok munkát igénylő számviteli és statisztikai rutinmunkákra érvényes. (Die Computer Zeitung, 1971. szept.)

Bulgáriában a mikroelektronika területén új kapacitások és az elektronikus számítástechnika megteremtése céljára az összegépipari beruházások 28%-át irányozták elő.

Tervek szerint SIT 1020 típusú computer berendezést, valamint mágnesszalag és mágneslemez tárolókat fognak gyártani. A távlati elképzelések szerint a Bulgáriában léte-

tendő számítóközpontok 70%-át saját gyártmányú számítógépekkel akarják felszerelni. (Die Computer Zeitung, 1971. szept.)

A félvezetőárak további esését mutatja az SGS Deutschland GmbH 1970 októberében kiadott árlistája, ami az 1970. január 21-i árjegyzékhez viszonyítva a diszkrét félvezetőeszközöket 44%-kal, az integrált áramköröket 30—60%-kal olcsóbban kínálja. (Elektrotechnische Zeitschrift, 1971. 23. k. 2. sz.)

A francia CII cég Varsóban, kétnapos szemináriumon a lengyel szakembereknek bemutatta IRIS típusú computereit és tájékoztatta őket a berendezés felhasználási lehetőségeiről. (Die Computer Zeitung, 1971. szept.)

Új sztereo készülékekkel jelentkezett a Philips cég. A szakértők egybehangzó véleménye szerint komoly érdeklődésre tarthat számot az Electrophon GF 805, a Stereo-Wechsler-Electrophon GF 447, a Stereo-Electrophon GF 604 (telepről és hálózatról működtethető), a Stereo-Wechsler-Electrophon GF 560 és a Stereo-Electrophon GF 808. A készülékek mind kiviteli koncepciójukban, mind pedig áramköri technikájukban jelentős előrelépést mutatnak. (Funktechnik, 1971. 17. sz.)

Új kazettás video-rekordert hozott forgalomba a Philips cég, N 1500 típuszámmal. A rendszer súlya 16 kg, 60 perces lejátszást tesz lehetővé. Kazettamérete 13×15×3,5 cm, a benne futó szalag 12,7 mm széles, a lejátszási sebesség 14,29 cm/sec. (Funktechnik, 1971. 16. sz.)

(Folytatás a 203. oldalon)



## Logikai hálózatok számítógépes vizsgálata\*

ETO: 164.1:681.325.65.001.4

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 áramköri tokok, 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 feladatsoporthoz 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 fokig 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,

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

A logikai hálózat modellezésére elvileg több lehetőség kínálkozik. Elképzelhető ellenállásokból, kapacitásokból, diódákból, tranzistorokból stb. felépített modell, amely nemlineáris áramkör formájában írja le a hálózatot. Ez a leírás bázis elemek (logikai alapáramkörök) esetén szükséges, azonban bonyolultabb hálózatok esetén egyrészt a számítási igény rohamosan nő, másrészt a logikai hálózat működésének megítélése szempontjából túzottan 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 áramköri tokokból felépített logikai hálózatok esetén célszerű az integrált áramköri 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 időekkel 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 áramköri 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 munka a Számítástechnikai Koordinációs Intézet megbízásából készült.

Beérkezett: 1972. III. 4.

a) Meghatározható a logikai hálózat bármely pontjának logikai egyenlete,

b) kiírható a kiválasztott ponton levő logikai érték 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 analízisének 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 kapcsolékról, 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 kívánságra 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 hibátáblázat kiindulási alakja 1. táblázat

| $X_n \dots X_2 X_1$ | $F$ | $F_1 F_2 \dots F_j \dots$ |
|---------------------|-----|---------------------------|
| 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 ...             |

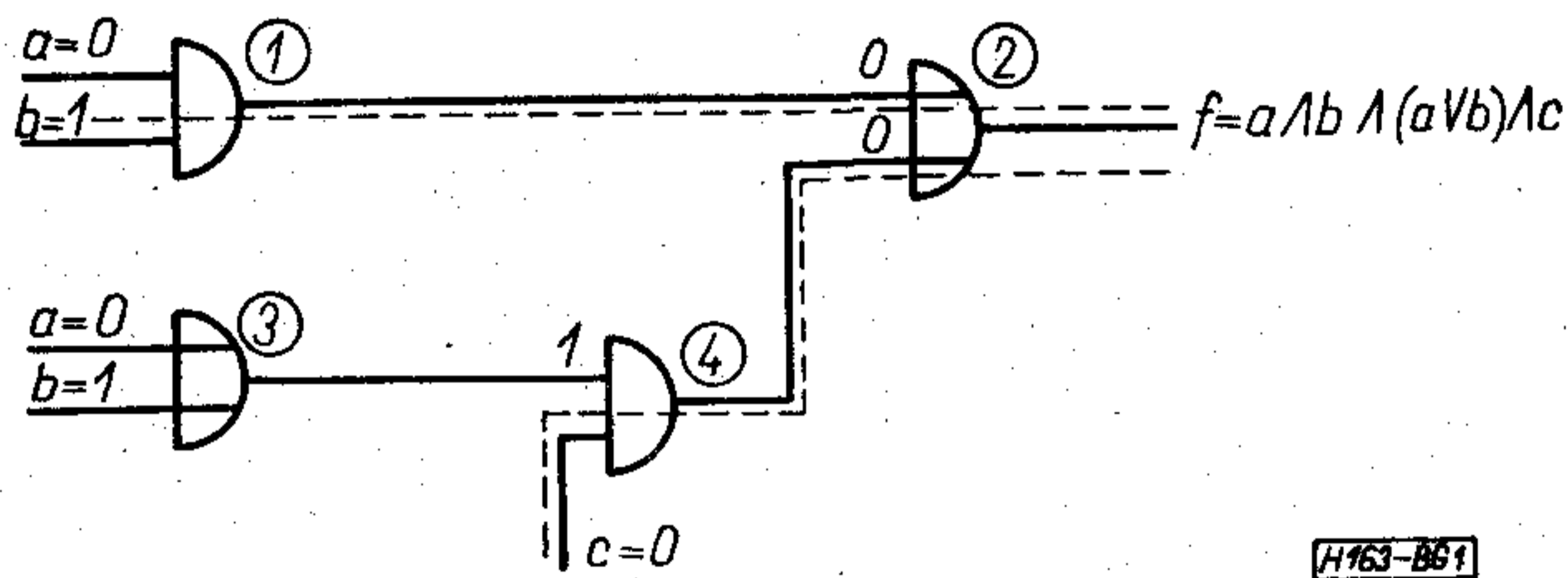
$F$ , mint említettük a hibamentes kimenetet,  $F_1, F_2, \dots, F_j, \dots$  a kimenet abban az esetben, ha valamelyik hibák fellép. A jobb oldalról egyes oszlopok elhagyhatók, nevezetesen, ha valamelyik  $F_j$  oszlop megegyezik  $F$ -el (vagyis a  $j$ -edik hibának nincs hatása a kimenetre) vagy, 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 kapcsolástanból 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 alapáramkör 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 egyszeri hiba detektálására kiterjeszhető és számítógép segítségével gazdaságosan elvégezhető. Az eljárás nem minimalizálja a vizsgálójelek számát.

2.3 A hiba-szótár módszerrel kombinációs és szekvenciális áramkörök egyaránt vizsgálhatók. A hiba szótár ismert hibák esetén fellépő vizsgálati eredmények rendszerezett összeállítása. 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 leképzések 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 bemenetétül 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 üzemű 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 információ feldolgozó 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 analízisére 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:

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

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:

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

#### 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 hazárd vizsgálata aszinkron 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.—Csopaki Gy.—Gefferth 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.—Géher 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] Géher 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 Design: 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 numerico, 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 Elements (SISLE). REDAC User's Manual, Vol. 2. 1969.
- [25] J. P. Roth: Diagnosis of Automatic Failures: A Calculus 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: Error-detecting 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.

## SZEMLE

(Folytatás a 198. oldalról)

1974-re fejeződik be a CANTAT II. elnevezésű mélytengeri kábel lefektetése, amely az angliai Widemouth Bay (Cornwall) és az Új Skóciában levő (Canada) Halifax között létesít kapcsolatot. A rendszer kiépítését a SEL egyik leányvállalata a Standard Telephones and Cables Ltd. végzi. Több mint 5 ezer kilométert fognak áthidalni, 1840 beszédesatorna áll majd rendelkezésre, a tervezett átviteli szélesség 14 MHz-es lesz. 455 közbeiktatott erősítőt helyeznek el a kábel mentén, egymástól mintegy 11 km-es távolságban. Eddigi ismereteink szerint a bejelentés váratlanul hat, hiszen már korábban több megállapodás született arra vonatkozóan, hogy gazdaságossági okok miatt az interkontinentális kábeles hírvitel helyett a továbbiakban csaknem kizárólag a műholdas hírközlésre fognak áttérni. (*Funktechnik*, 1971. 16. sz.)

A Dow Corning szilikonkaucsuk-alapú védőlakkot fejlesztett ki félvezetőeszközök bevonására. Alkalmazása révén különösen a műanyagtokozású MOS-áramkörök megbízhatósága növelhető jelentős mértékben.

A MOS-eszközök statikus feltöltődés által okozott, szerelés közben bekövetkező tönkremenetelét az Emerson & Cuming által forgalombahozott, elektromosan vezető habszivaccsal lehet meggátolni. Egyrészt ezzel vonják be a szerelőasztalok lapját, másrészt ezt használják csomagolóanyagként. A tranzistorok vagy integrált áramkörök lábait egyszerűen benyomják az ún. ECCOSORB-ba, ami rövidrezárja azokat, és ennek révén megakadályozza feltöltődésüket. (*Elektronik* 1971. 20. k. 1. sz.)

Az ICL elnöke közölte, reméli, hogy rövidesen sor kerül a lengyel Építésügyi Minisztériummal létrehozott műszaki kooperációs megállapodás ratifikálására.

A szóban forgó egyezmény ez esetben a harmadik lenne azok sorában, melyet az ICL lengyel vállalatokkal hozott létre. Az ICL egyébként ez ideig 13 számítógépet értékesített Lengyelországban és most kapott rendelést a 14. berendezésre.

Az ICL elnöke, Sir John Wall kétnapos megbeszélést folytatott a műszaki kooperáció témájában. Amennyiben az egyezmény elvi kérdéseiben megállapodás jön létre, a következő hónapokban műszaki szakértői delegáció keresi fel Lengyelországot, hogy a kooperáció részleteit kidolgozzák. (*Finantial Times*, 1971. aug.)

A General Electric új IC-szerelési technológiát dolgozott ki: a szokványos módon elkészített IC-morzsákat 35 mmi-es mozi-

filmhez hasonló filmre helyezik, amire előzetesen felvitték a fémes kivezetéseket. Az elemeket különleges módszerrel csatlakoztatják a fémezésekhez. Az így kialakított struktúrát vékony epoxifóliával védik.

A feltekeresélhető lyukasztott film, amire az IC-morzsákat szerelik, gépi úton vagy ollóval könnyen darabolható.

Az új IC-kivitel „minimod”-nak nevezték el. (*Electronics World*, 1971. 85. k. 3. sz.)

690-es típuszámmal új színes monitort hozott forgalomba az oszcilloszkópjairól híres Tektronix cég. A monitor érdekessége, a beépített 30 cm-es trinitron. Az automatikus irányítású rendszert jelenleg NTSC-szisztémára dolgozták ki. Előreláthatólag hamarosan forgalomba kerül a PAL és a SECAM rendszerű színes trinitronos monitor is. (*Fernseh und Kino-technik*, 1971. 6. sz.)

1971. decemberében hozta forgalomba a Tektronix cég a 211-es típuszámú új oszcilloszkópját. A szakértők egyhangzó véleménye szerint ez a kis méretű, hordozható vizsgáló egység, amely teljesen félvezetős felépítésű, új korszakot nyit az elektronikus vizsgálórendszerek fejlődésében. A hazai készülégyártás számára is feltétlenül hasznosak azok a tervezési és kivitelezési alapelvek, amelyek 211-es típus kialakításánál felhasználásra kerültek. Röviden ismertetjük az oszcilloszkóp fontosabb jellemzőit, amelyek együttesen, minden további megjegyzés nélkül indokolják a szakemberek előzőekben említett ritkán hallható véleményét.

A legmodernebb formatervezési szempontoknak megfelelő készülék súlya mindössze 1,4 kg, méretei 7,6 cm × 13,3 cm × 22,6 cm. Az 500 kHz-ig alkalmazható vizsgálórendszer az IEC szabványok értelmében II. osztályú berendezésnek számít. Eltérítési tényezője 1 mV/osztástól 50 V/osztásig folyamatosan változtatható, 5 · 10<sup>-6</sup> s és 200 · 10<sup>-3</sup> s közötti futattási idővel. Bemenő impedanciája 1 MΩ paralel 130 pF. Kettős szigeteléssel látták el, mind hálózatról mind pedig telepről működhet. Amennyiben a szokásos, nemzetközi szabványoknak megfelelő telepről üzemeltetik, 5 órás folyamatos működésre alkalmas. Kialakításánál arra törekedtek, hogy egészen extrém körülmények között is megbízhatóan használni lehessen: még 95%-os relatív páratartam esetén is változatlan paraméterekkel működik, -40 °C és 60 °C közötti hőmérsékleten tartósan tárolható. Viszonylag széles környezeti hőmérséklet-tartományban üzemképes: -15 °C és 55 °C között paraméter-változás nélkül alkalmazható. (*Tektronix*, 1971. dec.)

(Folytatás a 209. oldalon)

## A LOGAN logikai szimulációs program\*

ETO 621.3.049.7—111:681.3.06 LOGAN

Digitális rendszerek realizálása során elengedhetetlen gazdaságossági követelmény, hogy a rendszer konkrét kivitelezése, gyártása előtt meggyőződjünk az elkészített tervek helyességéről. Az ellenőrzés hagyományos „kézi” módja rendkívül fáradtságos és sok időt igénylő munka, hiszen még az egyes funkcionális egységek is nagyon sok elemből állhatnak. A tervek fokozatos részletezése, illetve a logikai kapcsolási rajzok többszöri átrajzolása során esetleg előforduló elkötések, téves bekötések helyének kimutatása, a még megengedhető terhelési és késleltetési viszonyok vizsgálata, a meg nem engedett vezérlések és állapotok felderítése részletes és alapos ellenőrzést kíván. Egy számítógépes logikai szimulációs program létjogosultságát a felsoroltakon túlmenően az is indokolja, hogy a tervek kisebb-nagyobb módosításának hatása — ha csak nem elemi változtatásról van szó — csak a tervezés ismételt elvégzésével azonos nagyságrendű munkával állapítható meg.

A Számítástechnikai Koordinációs Intézet megbízásából a Budapesti Műszaki Egyetem Híradástechnikai Elektronika Intézetében kidolgozott és most ismertetésre kerülő LOGAN programrendszer [1] alkalmas szinkron működésű szekvenciális hálózatok logikai, áramköri és vezérlési tulajdonságainak vizsgálatára. A programrendszer kidolgozásánál az alábbi szempontokat vettük alapul:

- A vizsgálati eljárás legyen teljes, tegye lehetővé bármely elem és a teljes hálózat működésének ellenőrzését;
- a hálózat elemeinek a számításokhoz szükséges adatai egy mágnesszalag-katalóguson álljanak rendelkezésre, hogy azokat ne kelljen a felhasználónak minden egyes feladat során megadnia;
- a katalógusba felvett elemkészlet elegendően bő választékot biztosítson;
- legyen független az elemek konkrét fizikai tulajdonságaitól (sebesség, hőmérséklet határ, nagyjelű és nagy zavarérzékenységű elemek stb);
- bármely feladathoz tartozó adatok gépi módon tárolhatók legyenek, hogy esetleges későbbi részletesebb vizsgálatokhoz újból egyszerűen hozzáférhetők legyenek;
- a programrendszer által biztosított vizsgálatok, ellenőrzések közül a felhasználó kívánsága szerint (fakultatíve) választhasson;
- a programrendszer felépítése tegye lehetővé a későbbi bővítéseket mind a vizsgálható hálózatok (pl. aszinkron), mind a felhasználható elem-

- készlet (pl. funkcionális egységek, MSI elemek, nagysebességű elemek) terén;
- szolgáljon alapul a konstrukciós tervezéshez;
- biztosítsa több hálózat egymás után sorozatban történő vizsgálatát külön külső beavatkozás nélkül is;
- a programrendszer legyen felhasználó orientált, alkalmazása ne igényeljen programozási ismereteket, használható legyen a programrendszer részletes ismerete nélkül is.

### 1. A programrendszer felépítése, működése, szolgáltatásai

A programrendszerrel végezhető ellenőrzések, vizsgálatok két nagyobb csoportra oszthatók.

A hálózat felépítésére és az áramköri működés ellenőrzésére vonatkozó vizsgálatok során kimutathatók és meghatározhatóak

- a hiányzó, nem értelmezhető és tilos összeköttetések,
- a forrás nélküli bemenetek és fogyasztó nélküli kimenetek,
- a megépítéshez szükséges integrált áramköri tokok száma,
- az egyes elemek terhelési viszonyai és
- a hálózaton belüli késleltetések.

A hálózat logikai analízise során meghatározható

- a hálózat bármely pontján levő jel logikai értéke a bemenő jelek és a tárolók belső állapotának tetszőleges értéke esetén,
- adott bemeneti vezérléssorozat hatására létrejövő jelsorozat a hálózat bármely pontján vagy kimenetén,
- a bemeneti vezérlés hatására kialakuló tárolóállapotok,
- a hálózat minden olyan eleme, melynek kimenetén a vezérléstől függetlenül állandó logikai szint található,
- a hálózat tetszőleges pontjához rendelhető logikai egyenlet.

A fenti vizsgálatok előtt minden esetben szükséges a hálózatra vonatkozó adatok beolvasása és ellenőrzése, továbbá — kiegészítve a katalógus adatokkal — a programrészek számára egységes adatbázissá alakítása.

A felsorolt vizsgálatokat, feladatokat a programrendszer egy-egy szubrutinja végzi, melyek egymástól függetlenül működnek és hívhatók. Aktivizálásukat — részben a felhasználó igényeinek megfelelően (vezérkártyák) — egy vezér programszegmens

Beérkezett: 1972. III. 4.

\* A munka a Számítástechnikai Koordinációs Intézet megbízásából készült.

biztosítja. A szubrutinok közös adatmezőből (COMMON) dolgoznak, a vizsgálatok eredményeit a soronyomatón közlik. Egyes szubrutinok külön input rendszerrel is rendelkeznek a közös alapadatokon kívül szükséges további információk beolvasására.

A programrendszer szerves részét képező katalógusszalag bővítésére, elemeinek cseréjére vagy törlésére, a katalógusba felvett elemek listájának vagy a katalógus teljes tartalmának kiírására külön program (KATAL) készült, mivel ezek a feladatok az áramköri és logikai analízistől elkülönülő egységet képeznek.

A LOGAN programrendszer a fentebb elmondottak alapján az alábbi egységekből épül fel:

#### 1. VEZER programszegmens:

Az egész programrendszer keretprogramja, irányítja az egyes vizsgálatokat végző szubrutinokat. A vizsgálatok elvégzésével biztosítja — amennyiben szükséges — a hálózat összes adatának mágnesszalagra vitelét, illetve újbóli vizsgálat során onnan beolvasását, valamint a programrendszer futásának újraindítását, ha további hálózat vizsgálatára is szükség van.

#### 2. INPUT szubrutin:

Új feladat esetén beolvassa és ellenőrzi a vizsgálandó hálózat alapadatait, közli az adatmegadás hibáit. Hibátlan adatmegadásnál összeállítja a közös adatbázist az alapadatokat és a katalógusszalag alapján.

#### 3. ELLEN szubrutin:

A hálózat működésére vonatkozó vizsgálatok előtt ellenőrzi a hálózat felépítését. Megkeresi az összes bekötési rendellenességet (hiányzó bekötés, nem értelmezhető bekötés stb.), az összes áramköri szempontból tilos összeköttetést (melyek esetén az egyes elemekben károsodás állhat elő) és az összes logikai szempontból tilos összeköttetést (pl. egy kapu bemenete és saját kimenete nem köthető össze) a hiba típusának és helyének megjelölésével.

#### 4. TOK szubrutin:

Meghatározza a hálózat megépítéséhez szükséges integrált áramköri tokok számát típusonként és összeítve. Hasonló számítást végez a szabadon maradt (be nem kötött) tokon belüli elemekre nézve is.

#### 5. TERH szubrutin:

Meghatározza minden egyes elem kimenetének tényleges terhelését, ezt összehasonlítja az elemre megengedett terhelhetőséggel. A számítás TTL egységterhelésben történik. A túlterhelt elemekről külön listát is ad. A szabad kollektoros kimenetekhez kiszámítja (beépített adatok alapján) az alkalmazandó ellenállás értékének alsó és felső határát. Végül megadja a hálózat bemeneteinek terhelését és a kimenetek terhelhetőségét.

#### 6. KESES szubrutin:

A hálózat bármely pontpárja között, a két pontot a kombinációs részhálózatban összekötő valamennyi útra nézve kiszámítja a maximális késleltetési időt és kiírja az utat alkotó elemeket. A kezdőpontokhoz kezdeti késleltetés rendelhető.

#### 7. EXAMKO szubrutin:

Azokat az elemeket keresi meg, melyek kimenetén a vezérléstől függetlenül állandó logikai szint van. Megadja az egyes elemek azon bemeneteit is, melyek az állandó kimeneti szintet okozzák.

#### 8. LOGEGY szubrutin:

A hálózat kiválasztott pontjához rendelhető logikai egyenletet adja meg. Az egyenlet változói a hálózat bemenetei és a tárolókimenetek, formátuma az ún. lengyel írásmód egy módosított változata. Az egyenletek felírásánál semmilyen egyszerűsítés vagy minimalizálás nincs figyelembe véve, vagyis a jel terjedésének útjához tartozó, a hálózatot felépítése alapján jellemző logikai egyenletet adja meg.

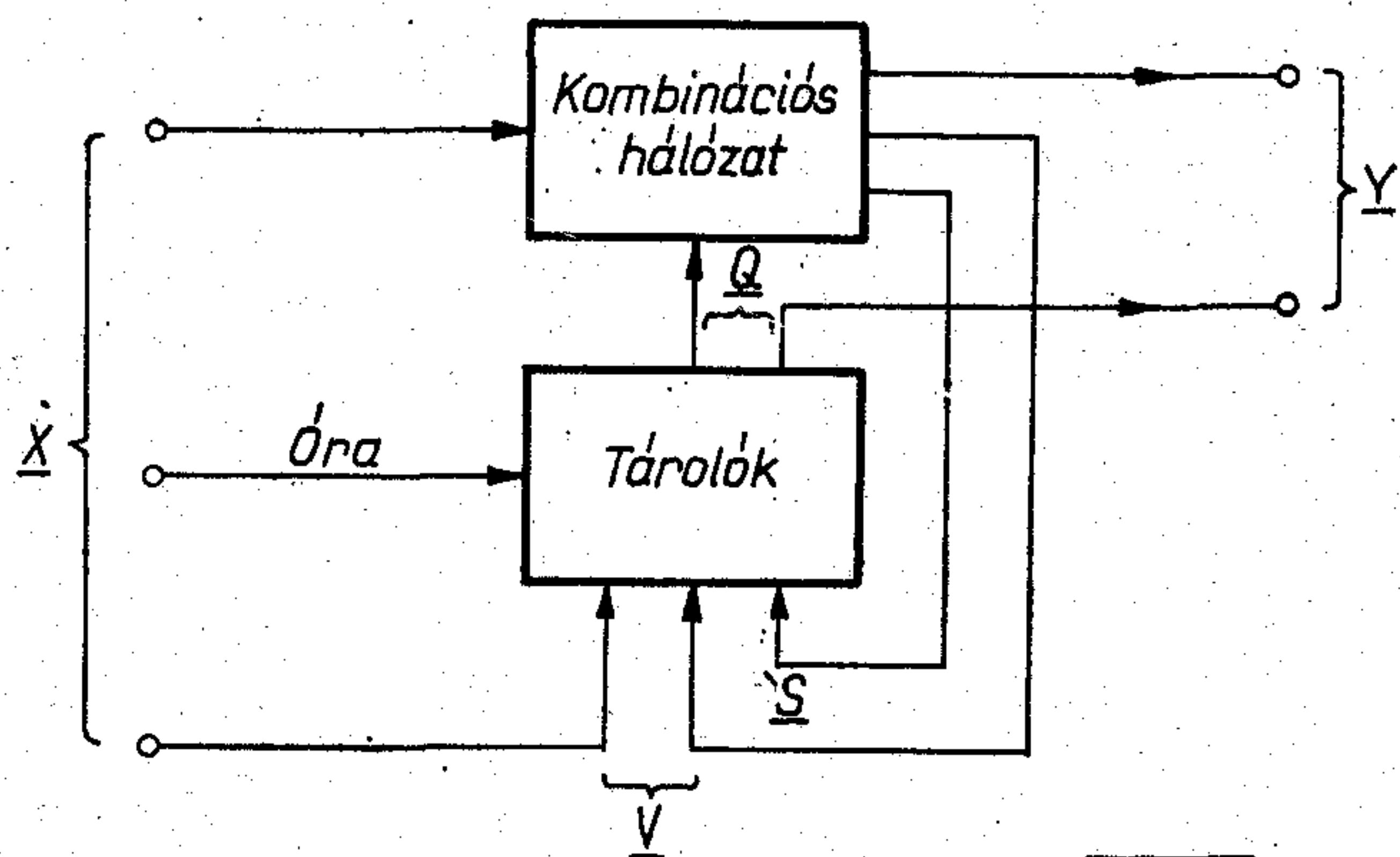
#### 9. LIZA szubrutin:

Meghatározza a hálózat kívánt pontjainak logikai értékét a hálózat bemenő jeleinek és a tárolók belső állapotának előírt értéke mellett. A tárolók tiltott vezérlését üzenet formájában közli. A bemenetek sorozatos vezérlése esetén a kiválasztott pontok logikai értékének kiírásával a hálózat időrendi működése ellenőrizhető.

### 2. A programrendszerrel vizsgálható hálózat

A LOGAN programrendszer szinkron működésű szekvenciális hálózatok vizsgálatára alkalmas. A vizsgált hálózat két fő részre bontható: kombinációs hálózatra és a hálózat belső állapotait realizáló tárolókra (1. ábra). A tárolók lehetnek mind szinkron, mind aszinkron vezérlésűek.

A hálózat szinkron működése feltételezi, hogy minden változás azonos időben, az órajelek egy meghatározott szintváltozásának pillanatában következzen be. A szinkron vezérelt tárolók órajelet csak a hálózat bemenetéről kaphatnak, azaz az órajelek kapuzása nincs megengedve, továbbá minden szinkron vezérelt tárolónak az órajelet azonos szintváltozására kell átbillennie. Az aszinkron vezérlésű tárolók vagy közvetlenül a hálózat bemenetéről, vagy a kombinációs hálózaton keresztül egymásról vezérelhetők. A hálózat bemenő jelei — így az aszinkron tárolók és a teljes rendszer időzítő vagy vezérlő jelei is — csak az órajelet idején változhatnak. Az egymásról történő aszinkron vezérlés hatására bekövetkező



1. ábra

H154-B1

állapotváltozásokat a program szintén az órajellel azonos idejűnek tekinti (a késleltetéseket figyelmen kívül hagyja). A számítás megfelelő sorrendjéről a program maga gondoskodik (lásd 3. fejezet).

A programrendszer megengedi a tárolók vegyes (szinkron és aszinkron) vezérlését is. A kétféle vezérlés egyidejű fellépte esetén az aszinkron vezérlést tekinti meghatározónak. Ily módon előfordulhat, hogy a program egyes tárolókat vezérlésük időbeli lefolyása alapján egyszer szinkron, másszor aszinkron tárolóként kezel.

A kombinációs hálózatrész a fentiek alapján egyrészt vezérli az aszinkron tárolókat, másrészt biztosítja a szinkron tárolók kijelölő vezérlését, ami meghatározza a tárolók következő órajel idején felvett állapotát, továbbá a hálózat bemenő jeleiből a tárolók állapota alapján előállítja a hálózat kimenő jeleit.

A hálózat szinkron működése csak akkor biztosított, ha a kombinációs hálózat elemei között nincs visszacsatolás és az aszinkron vezérelt tárolóknak egymásról — esetleg a kombinációs hálózaton keresztül — való vezérlése szintén egy nyílt lánc mentén történik. A programrendszer a logikai analízis előtt a hálózat elemeit (kapukat és tárolókat egyaránt) ún. kiszámíthatósági sorrendbe rendezi, ami megfelel a logikai jelek „terjedésének”. Ha ezt nem tudja elvégezni, mert egy elem logikai állapotának meghatározásához szükséges lenne kimenetének logikai értéke (pl. kapukból kialakított RS tároló), a hálózat aszinkron működésű. Az ilyen aszinkron működést a programrendszer felismeri és kimenő üzenet formájában közli, hogy a hálózat meg nem engedett belső visszacsatolást tartalmaz.

### 3. A logikai analízis algoritmus

A programrendszer — mint említettük — a hálózat bármely pontján a logikai érték megváltozását az órajel adott szintváltozásával azonos idejűnek tekinti. Az analízis során tehát nincs időkezelés, a változások ütemét az órajel frekvencia szabja meg, az órajelek közötti időben fellépő esetleges házardok és az aszinkron vezérelt tárolók átmeneti állapotainak kimutatására nincs lehetőség. Az analízis feltételezi, hogy a logikai értékek változása az órajel idején „végigfut” az egész hálózaton. Nagy hálózatok esetén az órajelperiódus nagyságrendjébe is eshető kritikus késleltetésekről a késleltetési időket vizsgáló részprogram eredményei alapján lehet képet kapni.

A logikai analízis algoritmusának egyszerűbb bemutatása érdekében vezessük be a következő logikai vektorokat (1. ábra). Jelölje a hálózat bemenő jeleit az

$$\underline{X} = [X_1, \dots, X_i],$$

a hálózat kimenő jeleit az

$$\underline{Y} = [Y_1, \dots, Y_j],$$

a tárolók állapotát a

$$\underline{Q} = [Q_1, \dots, Q_l]$$

vektor, továbbá a szinkron vezérlésű tárolók kijelölését (J, K, D bemenetek) az

$$\underline{S} = [S_1, \dots, S_m]$$

és az aszinkron vezérlő jeleket (CLEAR, PRESET, R, S bemenetek) a

$$\underline{V} = [V_1, \dots, V_n]$$

vektorok. Az egyes vektorok  $k$ -adik ütemben felvett értékét felső indexezéssel jelöljük (pl.  $X^k$ ).

A tárolók állapotát a  $k$ -adik ütemben szinkron vezérlés esetén az

$$\underline{S}^{k-1} = F_s(\underline{X}^{k-1}, \underline{Q}^{k-1})$$

kijelölés és az előző ütem során felvett állapot határozza meg a

$$\underline{Q}^k = F_{QS}(\underline{S}^{k-1}, \underline{Q}^{k-1}, \text{CLOCK})$$

függvény alapján, míg aszinkron vezérlés esetén a

$$\underline{V}^k = F_V(\underline{X}^k, \underline{Q}^k)$$

vezérlő jelek a

$$\underline{Q}^k = F_{QV}(\underline{V}^k)$$

függvény alapján. A kimenetek logikai értéke az

$$\underline{Y}^k = F_Y(\underline{X}^k, \underline{Q}^k)$$

függvényből határozhatók meg.

A programrendszer az egymás utáni órajelek ( $k=1, 2, \dots, N$ ) hatására meghatározza minden előre megadott  $\underline{X}^k$  bemenő vektorhoz tartozó  $\underline{Q}^k$  állapot- és  $\underline{Y}^k$  kimenő-vektort, valamint a szinkron vezérlésű tárolók következő ütembeli állapotát eldöntő  $\underline{S}^k$  kijelölő vektort. A számítások elvégzéséhez tárolni kell a  $(k-1)$ -edik ütemhez tartozó  $\underline{S}^{k-1}$  és  $\underline{Q}^{k-1}$  vektorokat.

A hálózat logikai analíziséhez meg kell adni a bemenő változók mindazon kombinációját, melyek mellett a logikai működést vizsgálni akarjuk. Az analízis megkezdéséhez szükséges ezenkívül a tárolók kezdeti állapotának ( $\underline{Q}^0$ ) és a szinkron tárolók kezdeti kijelölésének ( $\underline{S}^0$ ) az ismerete. Utóbbi a bemenő vektor kezdeti értékének ( $\underline{X}^0$ ) ismeretében már a program határozza meg. Az aszinkron vezérlésű tárolók kiindulási állapota tetszőleges lehet, mivel azok  $\underline{X}^0$ -val és egymásról is vezérelhetők, így a programrendszer  $\underline{X}^0$  ismeretében beállítja a bemenő változók kezdő értékének megfelelő helyes kiinduló állapotokat.

A programrendszer tehát a megadott  $\underline{X}^0, \underline{Q}^0$  adatokból meghatározza a hálózat kiindulási (vezérlés előtti) állapotát ( $\underline{Q}^0, \underline{Y}^0, \underline{S}^0$ ). Az első órajel idején érvényes  $\underline{X}^1$  bemenő vektor és a hálózat kiindulási állapota alapján kiszámítja az új állapotnak megfelelő  $\underline{Q}^1, \underline{Y}^1, \underline{S}^1$  vektorokat és további órajelek hatására a számítás ciklikusan ismétlődik.

A számítás eredményeit a program a kiválasztott pontok logikai értékének oszlopfolytonos kiírásával közli. A kiírás formátuma (a két lehetséges logikai



érték egymáshoz képest eltolva jelenik meg) alapján a megadott pontokon lévő logikai jelek idődiagramja is rendelkezésre áll.

#### 4. A programrendszer elemkészlete és korlátozásai

A programrendszer — használatának (adatmegadás) megkönnyítése érdekében — egy katalógusszalaggal rendelkezik. Ezen megtalálhatók a szinkron működésű hálózatokban leggyakrabban használt logikai elemek minden olyan adata, melyre a programrendszer által elvégezhető vizsgálatok során szükség van. Az elemkészlet kialakításánál a hazai viszonyokra való tekintettel célszerűnek látszott a TEXAS TTL integrált áramköri készlet SN74..N sorozatát alapul venni. A katalógus-szalag jelenleg az 1. táblázatban megadott elemek adatait tartalmazza.

1. táblázat

| Típus  | Logikai funkció   |
|--|---|
| SN7404N, SN7405N   | inverterek  |
| SN7400N, SN7401N,<br>SN7403N, SN7410N,<br>SN7420N, SN7430N,<br>SN7440N | 2-, 3-, 4-, 8- bemenetű<br>NAND kapuk                             |
| SN7402N  | 2-bemenetű NOR kapuk  |
| SN7450N, SN7451N,<br>SN7453N, SN7454N                                  | 2×2-, 4×2- bemenetű<br>AND-OR-INVERT ka-<br>puk (a bővíthetők is) |
| SN7460N  | 4-bemenetű EXPANDER-<br>ek  |
| SN7472N, SN7473N,<br>SN7476N   | J-K-MASTER-SLAVE<br>FLIP-FLOP-ok                                  |
| SN7474N  | D-FLIP-FLOP-ok  |

Megjegyzés: az aláhúzott típusok szabadkollektoros kimenettel rendelkeznek

2. táblázat

|  |           |
|--|-----------|
| A hálózat bemeneteinek és kimeneteinek együttes száma              | max. 100  |
| A hálózatban lévő tárolók száma                                    | max. 100  |
| A hálózatban lévő elemek száma                                     | max. 200  |
| Az elemtípusok száma   | max. 20   |
| A hálózat csomópontjainak száma                                    | max. 300  |
| Az elemek bekötött lábainak száma                                  | max. 1000 |
| A logikai analízis során egyszerre kiírható logikai változók száma | max. 30   |
| A kiírható leghosszabb logikai egyenlet karaktereinek száma        | max. 4000 |

A katalógusba felvehető elemek száma tetszőleges lehet. Bővítés esetén, ha az új elem logikai funkciója egy már meglévővel megegyezik, csak az új elem katalógusadataira van szükség. Ha az új elem bevezetése új logikai funkció megjelenésével jár, szükséges az egyes vizsgáló részprogramok kibővítése is.

A programrendszer jelenlegi kiépítésében csak bizonyos korlátozások betartásával használható. A korlátozások egyrészt a felhasznált algoritmusból adódnak (tiltott órajel kapuzás, meg nem engedett aszinkron visszacsatolás), másrészt abból, hogy a számítógép kapacitása alapvetően meghatározza a vizsgálható hálózat méreteit. Utóbbi korlátokat a 2. táblázatban tüntettük fel.

#### 5. Alkalmazási lehetőségek

Jelen fejezetben — a teljesség igénye nélkül, felsorolásszerűen — megkíséreljük összefoglalni azokat a területeket és problémaköröket, ahol a programrendszer a tervezési vagy ellenőrzési munkában hasznos segítséget nyújthat.

Rendszertechnikai szempontból nézve a programrendszer segítségével a digitális áramkörök széles köre vizsgálható. Digitális számítógépek és vezérlő rendszerek területén pl. aritmetikai egységek, vezérlő egységek és rendszerek, szelekciós áramkörök (mágneslemezes, mágnesdobos, ferritmagos tárolók kiválasztó áramkörei), valamint ellenőrző logikai hálózatok működésének szimulációja és ellenőrzése végezhető el. Vizsgálhatók digitális jelátviteli rendszerek hibajelző, hibajavító áramkörei; ipari szekvenciális vezérlő rendszerek; szabályozó szerveket realizáló logikai hálózatok — pl. PD, PI, PID stb. algoritmusok ellenőrizhetők. Használható a program továbbá digitális rendszerek elemi funkcionális egységeinek (számláncok, kódkonverterek, üzemmód átalakítók, időzítő áramkörök stb.) vagy egy hálózat tetszés szerint kiválasztott részének a vizsgálatára.

A hálózatok ellenőrzésére — az időbeni sorrendet tekintve — először a tervezés folyamán van szükség. A logikai tervek elkészülésekor, a további konstrukciós tervezés előtt meg kell győződnünk a tervezés helyességéről, az áramkör működőképességéről. Az ellenőrzés folyamán nemcsak a logikai működés, hanem a logikai tervezés során figyelembe nem vehető szempontok alapján az áramköri működés is vizsgálható (terhelési viszonyok, késleltetések, meg nem engedett, illetve hiányzó bekötések). A késleltetési idők ismeretében meghatározható a maximális órajel frekvencia, vagy adott frekvencia esetén ellenőrizhető, hogy a tranziszterek lezajlanak-e két órajel között, illetve megfelelő-e az időtartalék. Ugyancsak így ellenőrizhető a tárolók egyszerre történő billenésének feltétele. A kezdeti késleltetések megadásával figyelembe vehetők az elektromos szerelés okozta késleltetések is. Kimutathatók a forrás vagy fogyasztó nélküli csomópontok, az áramkör belső aszinkron jellegű visszacsatolásai.

A hálózat logikai analízise során meghatározható a tárolók vezérlési sorrendje, felvehető az állapot-táblázat, kimutathatók a meg nem engedett állapot átmenetek, ellenőrizhető, hogy a hálózat előre felvett állapota fenn áll-e a bemenetek adott értékénél. Kimutathatók a vezérléstől függetlenül állandó logikai értéket képviselő csomópontok és elemek. Bármely belső pont logikai értékének változása vizsgálható a bemeneti vezérlés hatására, előállítható a logikai jelek idődiagramja.

Részekre bontott nagyobb hálózat esetén egyszerűen meghatározhatók a következő részek bemenő jelei, a külső vezérlő jelek könnyen figyelembe vehetők. Megadhatók, hogy az egyes egységek milyen terhelést fejtenek ki egymásra.

A hálózatban végrehajtott minden változtatásnak, egy-egy rész cseréjének vagy új elem bevezetésének hatása minden követelmény alapján vizsgálható (logikai működés, terhelési és késleltetési viszonyok, IC tok igény stb.). Az elemkészletben nem szereplő bonyolultabb integrált áramköri elemek (dekódolók, összeadók, paritás-ellenőrzők) fiktív elemekre bontva helyettesíthetők; így a logikai analízis ilyen hálózatokra is elvégezhető.

Használható a programrendszer már meglévő hálózatok élesztésével kapcsolatos automatizált vizsgálatok előkészítésénél, karbantartási dokumentációk készítésénél, különböző teszt-programok kialakításánál és a hálózatok diagnosztikai vizsgálatánál. Hat hatós segítséget nyújthat a diagnosztikához szükséges hibaszótár elkészítéséhez. Mivel a program segítségével a hálózat bármely belső pontja „elérhető”, az áramkör jelentéktelen módosításával bármely belső pont logikai értéke tetszés szerint beállítható, így valamely hiba hatása adott bemeneti vezérlés mellett a hiba helyétől a kimenetekig bárhol vizsgálható és kimutatható. Ez lehetőséget ad hardware mérésekkel való összehasonlításra, sőt a belső pontok vizsgálhatósága révén ahhoz képest jelentős többletet is nyújt.

**6. A vizsgálandó hálózat megadása**

A hálózat megadása a program számára — amennyiben a hálózat elemeit a meglévő elemkészletből vesszük — tulajdonképpen a hálózat struktúrájának leírására és az egyes elemek típusainak közlésére korlátozódik. Az adatmegadás további egyszerűsítését szolgálja a szimbolikus nevek használata.

A felhasználónak névvel (azonosítóval — max. 4 karakter) kell ellátnia a hálózat minden egyes bemenetét, kimenetét és elemét. Szintén szimbolikus névvel (max. 12 karakter) hivatkozhatunk az egyes elemek típusára (pl. SN7400N). Az elemek be- és kimeneteit ún. lábtámasszal látjuk el, mely megegyezik az integrált áramköri tokok katalógusokban közölt lábszámával. A nem elemhez tartozó, logikai funkciót nem teljesítő kivezetések (tápfeszültség, föld, üres lábak) figyelmen kívül hagyandók. A minden szempontból azonosnak tekinthető elemekre azonos lábszámok használhatók.

A hálózat megadásához ezek után fel kell sorolni:

1. a hálózat külső csatlakozási pontjait (csatlakozási lista),
2. típusonként a hálózat elemeit (típuslista),
3. csomópontként az egymással összekötött elemkivezetéseket és külső csatlakozási pontokat (név, vagy név és lábszám — kötési lista).

A kötési listában — a nem használt tároló kimeneteken kívül — minden elemkivezetésnek szerepelnie kell. Az elemek nem vezérelt vagy „felesleges” bemeneteit vagy egymással, vagy a tápfeszültség, illet-

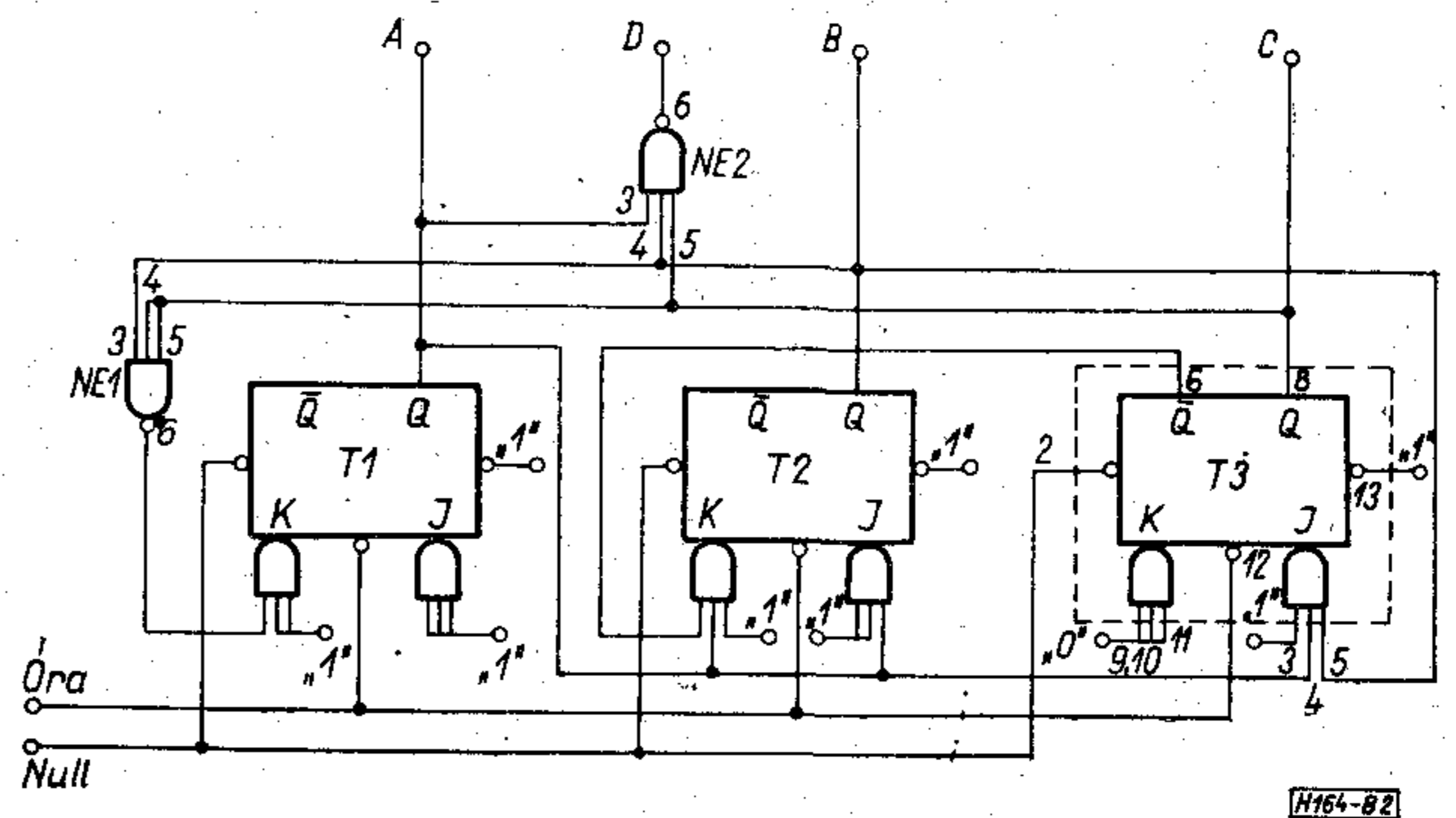
ve föld pontokkal kell összekötni. A tápfeszültségre, illetve földre kötött elem lábakat a kötési lista utolsó két csomópontjaként kell megadnia.

Az adatmegadás formátumával, továbbá az egyes áramköri és logikai vizsgálatok végrehajtásához szükséges további adatok megadásával kapcsolatban a LOGAN programrendszer használati utasítására utalunk [2].

A program az eredmények közlésénél a vizsgálat helyére szintén szimbolikus nevekkkel és lábszámokkal hivatkozik.

**7. Példa**

A programrendszer használatára vonatkozóan tekintsük illusztratív példaként a 2. ábrán megadott hálózatot. A hálózat egy 7-ig számláló, majd önmagától leálló szinkron számlánc, újraindítása a NULL bemeneten keresztüli aszinkron törléssel lehetséges.



2. ábra

Az ábrán feltüntettük a szimbolikus nevet és az elemek lábszámozását is (a tárolóknál a lábszám azonos). A hálózat megadása az alábbi módon történhet:

Csatlakozási lista:

ÓRA, NULL, A, B, C, D

Típuslista:

SN7472N T1, T2, T3  
SN7410N NE1, NE2

Kötési lista:

|       |     |     |     |                             |     |                        |         |    |    |    |
|-------|-----|-----|-----|-----------------------------|-----|------------------------|---------|----|----|----|
| ÓRA,  | T1  | 12, | T2  | 12,                         | T3  | 12                     |         |    |    |    |
| NULL, | T1  | 2,  | T2  | 2,                          | T3  | 2                      |         |    |    |    |
| A,    | T1  | 8,  | T2  | 10,                         | T2  | 5,                     | T4, NE2 | 3  |    |    |
| B,    | T2  | 8,  | T3  | 5,                          | NE1 | 3,                     | NE2     | 4  |    |    |
| C,    | T3  | 8,  | NE1 | 4,                          | NE1 | 5,                     | NE2     | 5  |    |    |
| D,    | NE2 | 6   |     |                             |     |                        |         |    |    |    |
| T3    | 6,  | T2  | 9   |                             |     |                        |         |    |    |    |
| T1    | 9,  | NE1 | 6   |                             |     |                        |         |    |    |    |
| T3    | 9,  | T3  | 10, | T3                          | 11  | — földpont (logikai 0) |         |    |    |    |
| T1    | 13, | T2  | 13, | T3                          | 13, | T1                     | 3,      | T1 | 4, | T1 |
|       |     | T1  | 10, | T1                          | 11, | T2                     | 3,      | T2 | 4, | T2 |
|       |     | T3  | 3   | — tápfeszültség (logikai 1) |     |                        |         |    |    |    |

A logikai analízis megindításához szükséges kiinduló tárolóállapotok megadása tetszőleges lehet (pl. olyan is, mely a normális működés során nem fordulhat elő), ha a bemenetek kiinduló értékének megadásánál törlőjelet adunk a NULL bemenetre. Az

3. táblázat

| Sorszám | ÓRA | NULL | T1<br>8 | T2<br>8 | T3<br>8 | D |
|---------|-----|------|---------|---------|---------|---|
| 0       | 0   | 0    | 0       | 0       | 0       | 1 |
| 1       | 1   | 1    | 1       | 0       | 0       | 1 |
| 2       | 1   | 1    | 0       | 1       | 0       | 1 |
| 3       | 1   | 1    | 1       | 1       | 0       | 1 |
| 4       | 1   | 1    | 0       | 0       | 1       | 1 |
| 5       | 1   | 1    | 1       | 0       | 1       | 1 |
| 6       | 1   | 1    | 0       | 1       | 1       | 1 |
| 7       | 1   | 1    | 1       | 1       | 1       | 0 |
| 8       | 1   | 1    | 1       | 1       | 1       | 0 |
| 9       | 1   | 1    | 1       | 1       | 1       | 0 |
| .       |     |      |         |         |         |   |
| .       |     |      |         |         |         |   |
| .       |     |      |         |         |         |   |
| n-1     | 1   | 1    | 1       | 1       | 1       | 0 |
| n       | 1   | 0    | 0       | 0       | 0       | 1 |
| n+1     | 1   | 1    | 1       | 0       | 0       | 1 |
| .       |     |      |         |         |         |   |
| .       |     |      |         |         |         |   |
| .       |     |      |         |         |         |   |

elsődlegesnek tekintett aszinkron vezérlés hatására ugyanis a program automatikusan beállítja a helyes alapállapotot.

A logikai működést az analízis eredményeként a 3. táblázat alapján követhetjük. A bemenetek vezérlését meg kell adni, a program azonban csak akkor írja ki, ha a kiíratandó pontok listáján a táblázat teljes fejlécét felsoroljuk. A futtatás során az A, B, és C kimenetek helyett a tárolók Q-kimeneteit irattuk ki.

### 8. Programnyelv, helyfoglalás, futási idő

A LOGAN programrendszer FORTRAN—IV nyelven készült az SZKI SIEMENS 4004/45-ös számítógépre. Helyfoglalása a gép memóriájában 56K szó

(32 bites szóhossz). A felhasznált perifériák: kártya-olvasó, sornyomtató és két mágnesszalag egység. Tartozékai: két mágnesszalag, egyrészt a programrendszer tárolására, másrészt a katalógus és a vizsgált hálózatok adatainak tárolására.

A KATAL katalógus kezelő program nyelve szintén FORTRAN—IV, helyfoglalása 21K, ugyanazokat a perifériákat használja, mint a LOGAN programrendszer.

A programrendszer szalagról beolvasása és részeinek összefűzése 4 percet vesz igénybe. A példaként bemutatott számlánc esetében az áramkör felépítésére és logikai működésére vonatkozó vizsgálatokhoz 20 s gépidő szükséges.

Egy bonyolultabb feladat várható időigényének megbecslésére közöljük még, hogy egy bináris, négy bites, párhuzamos üzemmódú összeadó egység, hibaellenőrzéssel (21 csatlakozási pont, 46 elem, 6 elem-típus, 62 csomópont) és egy BCD kódú összeadó egy decimális helyértékét kezelő részének (14 csatlakozási pont, 25 elem, 6 elem-típus, 47 csomópont) egymás utáni teljes vizsgálatához 4 perc gépidőre volt szükség (tisztá futási idő) [2].

A szerzők köszönetüket fejezik ki Farkas György, dr. Flesch István, dr. Géher Károly, Jagudits László, Pápai Zsolt, dr. Szittyá Ottó és Theisz Péter kollegáinknak a programrendszer kidolgozása során tőlük kapott hasznos tanácsokért és az értékes eszmecsereért.

### I R O D A L O M

- [1] A LOGAN logikai áramkört vizsgáló számítógép programrendszer. Tanulmány az SZKI számára, 1970.
- [2] Használati utasítás a LOGAN logikai hálózatok analízisére szolgáló programrendszerhez. 1971.
- [3] TTL Integrated Circuits Catalog from Texas Instruments.
- [4] Bohus M.—Géher K.: Logikai hálózatok számítógépes vizsgálata. Híradástechnika (ebben a számban).

### S Z E M L E

(Folytatás a 203. oldalról)

A Systems 71 kiállításon a részt vevő cégek több, mint kétharmada amerikai kiállító volt, akik az egyszerű, bemeneti termináloktól kezdve, a mágnesszalag- és lemezegységeken át különböző típusú és árfekvésű képernyős megjelenítő készülékeket mutattak be. Az amerikai perifériagyártó cégek semmilyen személyi és anyagi ráfordítást nem sajnáltak annak érdekében, hogy betörjenek az európai, elsősorban az nyugat-német piacra.

A kiállítással egyidőben rendezett szimpóziumon F. Sherwood, a Diebold Európa SA alelnöke, igen érdekes beszámolót tartott, amelyből a következőket idézzük:

Az utóbbi időben végzett felmérések szerint a forgalomban levő számítógépek sem az USA-ban, sem Európában nem felelnek meg a követelményeknek. A Diebold szerint végzett vizsgálatok azt mutatják, hogy az USA-ban a felhasználók mindössze 40 %-a, Európában 30 %-a használja ki berendezéseit. Az európai számítógépipar fejlesztésével összefüggő tenivalókkal kapcsolatban Sherwood rámutatott arra, hogy végül is az európai számítógép gyártóknak az európai felhasználó cégek sok tekintetben fontos partnerei. Arra is lehetőség nyílna, hogy az európai technológiai fejlődés jövője szempontjából oly fontos együttműködés alapjait lerakják. A jelenlegi tények azonban azt mutatják, hogy ezen fontos iparág fejlődése

igen lassú és egyre inkább nemzeti, mintsem nemzetközi jellegű. A helyzetet még az is súlyosbítja, hogy egy országon belül sem tudnak e tekintetben megfelelő egyezségekre jutni. Joggal merül fel a kérdés, ha az űrhajózás területén meg tudták teremteni az európai együttműködést, miért ne lehetne ez a számítógépek vonatkozásában is így? Az e területen megvalósított összefogás gazdaságos lenne; egy európai egyesülés olyan hardware-t tudna előállítani, amely az amerikai berendezésekkel versenyképes lenne.

A szimpóziumon felmerült a szakemberhiány kérdés is, ami igen komoly problémát jelent. Becslések szerint 1975-re az NSZK-ban a szükségesnél 35 000 adatfeldolgozó szakemberrel lesz kevesebb. A számszerűsége túlmenően, még a meglévő szakemberek képzésére is figyelemmel kell lenni. Szakmai körökben az a vélemény, miszerint a legtöbb rendszerelemző, programozó és operátor képzettségi szintje nem kielégítő. A szakemberhiány a magas fizetések és a kedvező perspektívák ígéretében sok szakképzetlen és egyéb szakterületeken sem megfelelően helytálló embereket vonz.

Az elektronikus adatfeldolgozó specialisták nem megfelelő képzettségéhez az is hozzájárul, hogy nem áll rendelkezésre kellő számú oktató és az oktatási intézmények színvonala sem mondható kiválóknak.

(Folytatás a 214. oldalon)

## Nyomatási rajzok készítése a TAL—1 programmal

ETO: 621.3.049.75.001.2 : 681.3.06 TAL—1

A nyomtatott áramköri lapok tervezése a konstrukció egyik legtöbb munkát igénylő szakasza. A berendezések tervezése során leginkább ebben a fázisban jelentkezik szűk keresztmetszet, ezért van a legnagyobb szükség új módszerek bevezetésére. Ha a feladat megoldására számítógépet alkalmazunk, akkor időben igen jelentős nyereség érhető el. A SEMA Metra International által kidolgozott TACTIC 2 nevű program CDC 6600-as gépen például 2—5 perc alatt tervezi meg egy 20 IC tokot tartalmazó, kétoldalas nyomtatású kártya nyomtatási vonalait. Ugyanezt a munkát egy gyakorlott technikus 10—20 óra alatt tudja csak elvégezni. Körülbelül hasonló nagyságú időnyereség érhető el a kártyák fotosablonjainak gépi rajzolása esetén is a kézi rajzolással szemben. A 10—20 órás rajzmunkát egy jó rajzgép a kártya bonyolultságától, a nagyítási aránytól és a rajzolási sebességtől függően 1/2—1 óra alatt végzi el.

A kártyák tervezése a berendezés tervezésének jól elhatárolható szakasza. Jól definiáltak a be- és kimeneti adatok, illetve eredmények is. Ugyanakkor ebben a szakaszban érhető el a legnagyobb időnyereség a számítógépes tervezés alkalmazása révén. Mindebből következik, hogy a kártyák tervezését akkor is lehet és érdemes automatizálni, ha a tervezés teljes folyamata hagyományos módszerekkel történik. Ennek megfelelően valóban az a helyzet, hogy a konstrukciós software-ek legtöbbje a kártyák tervezésére vonatkozik. Ezeket a programokat sokszor valamilyen célgéppel együtt ajánlják (rajzgép, fúró-automata stb.), és a gép megvásárlása esetén a vevő rendelkezésére bocsátják, vagy pedig a program tulajdonosa szolgáltatásként azonnal a fotosablont, sőt az elkészített nyomtatott lapot szállítja. A célgéphez kapcsolódó és azzal együtt szállított programok legnagyobb része még nem oldja meg a nyomtatott vonalak útvonalának megtervezését, hanem azt előzőleg kézzel kell elvégezni, és a program számára alkalmas formában megadni. Az alkalmas forma itt azt jelenti, hogy a felhasználó egyszerűen és könnyen tudja leírni a nyomtatási vonalak adatait a program számára. Az ilyen programok felhasználóinak nagy száma mutatja, hogy a tervezésnek még ezt a rövid szakaszát is érdemes és kell automatizálni, akár a többitől függetlenül is.

A SzKI-ban kidolgozás alatt álló programrendszer készítői a konstrukciós tervezés egészének automatizálását tűzték ki célul. A munkák során elsőként a TAL—1 nevű program készült el, amely nyomtatási terv adatai alapján a ZUSE Graphomat rajzgép számára készít vezérlő lyukszalagot. A fentiek értelmében célszerűnek látszott ezt a programot úgy kialakítani, hogy önmagában is felhasználható legyen, és a

kézzel elkészített nyomtatási terv lyukkártyán megadott adatait is értelmezni tudja. Így a TAL—1 program a programrendszer további részei nélkül is jó eredménnyel alkalmazható.

Ugyanilyen kialakítású a kártyákat tervező program is: tagja egy programrendszernek, azonban attól függetlenül vagy a TAL—1 programmal összefűzve is felhasználható.

A továbbiakban ismertetjük a kártyatervező program kialakításánál figyelembe vett szempontokat, valamint a TAL—1 program felépítését és alkalmazását.

### 1. Kártyák nyomtatási tervének elkészítése

Nyomatási tervnek nevezzük azt a rajzot vagy leírást, amely a kártyára kerülő valamennyi nyomtatott vezeték (vagy vezetékalkazat) elhelyezéséhez és kialakításához szükséges információt tartalmazza. A nyomtatási rajz a készítendő kártya mérethelyes vagy méretarányos képe, amely tehát tartalmazza az összes nyomtatott vezetékeket a konkrét geometriai kialakítás szerint. A nyomtatási tervnek nem kell kifejezetten tartalmaznia olyan adatokat, mint például a jelvezetékek szélessége, a furatmezők átmérője vagy az érintkezők geometriai méretei; ezeket csak a nyomtatási rajzon kell feltüntetni. Természetesen a tervezés során ezeket is figyelembe kell venni és a tervet ennek megfelelően kell elkészíteni.

A nyomtatási terv elkészítése három feladat megoldásából áll, ezek a következők:

- a) IC tokok összeállítása,
- b) az áramköri elemek elhelyezése,
- c) az összekötő vezetékek útvonalának megtervezése.

Kézi, azaz nem számítógépes tervezés esetén a tervező ezt a három feladatot nem egymástól függetlenül, különállóan oldja meg, hanem szinte teljesen párhuzamosan. Egy-egy rész megoldásnál állandóan figyelembe veszi a kölcsönhatásokat, és a legkedvezőbb megoldás érdekében mindhárom feladat korábbi részeredményeit módosítja. Számítógépre ez a komplex tervezési mód nem alkalmazható, mivel az algoritmizálás nagy nehézségekbe ütközik. A feladatok külön-külön, egymás után történő megoldása viszont általában nem a legkedvezőbb eredményt adja, és ez is egyik oka annak, hogy a gépi tervezés nem mindig éri el a gyakorlott technikus munkájának színvonalát. Az ismertetésre kerülő program az eredményesség javítása érdekében az a) és b) feladatot együttesen oldja meg, de a c) feladatot már azoktól különállóan, az eredményeket felhasználva, de nem módosítva. Módosításra a tervezőnek van lehetősége,

egyszer az *a*) és *b*) feladat, másodszer a *c*) feladat befejezése után. A változtatásokat a géppel közölni lehet, és a program azokat figyelembe tudja venni. Az 1.1 pont az *a*) és *b*) feladat, az 1.2 pont pedig a *c*) feladat megoldását ismerteti.

### 1.1 Összeállítás és elhelyezés

A nyomtatás tervezésének megkezdésekor a tervező rendelkezésére áll a kártyára kerülő áramkör logikai terve. Ez lehet gépen belüli leírás, ha a korábbi tervezést is géppel végeztük vagy az áramkör logikai rajza, amelyet először be kell vinni a gépbe.

Az utóbbi esetben előkészítő munkára van szükség, amelynek során a logikai rajz adatait a gép számára alkalmas formára alakítjuk át. Szükség van ezek után még egy programra is, amely az előkészített adatokat beolvassa, ellenőrzi és elkészíti belőlük azt a leírást, amely az első esetben már készen áll.

A gépen belüli leírásnak a következőket kell tartalmaznia:

#### a) A logikai tervben előforduló báziselemek azonosítói

A továbbiakban báziselemnek nevezzük egy integrált áramköri toknak azt a legkisebb áramköri részét, amely önálló funkció elvégzésére alkalmas. Egy IC tok tehát egy vagy több báziselemet tartalmazhat. A kártya logikai tervében minden báziselemhez azonosítót kell rendelnünk, amely megkülönbözteti az összes többi báziselemtől. A logikai terv nem tartalmazhat olyan elemeket, amelyek nem egyeznek meg a gépben tárolt IC katalógus (lásd *c*) pont) valamely báziselemével.

#### b) A logikai tervben előforduló integrált áramkörök típusazonosítói

Ebben a listában fel kell sorolni a logikai terv valamennyi IC típusazonosítóját és minden típushoz a hozzá tartozó báziselemek azonosítóját. A gép ebből a listából állapítja meg, hogy valamely báziselemet milyen típusú IC tokban kell elhelyeznie.

#### c) Integrált áramkörök katalógusa

Ez tartalmazza a felhasználásra kerülő integrált áramkörök katalógusadatait. Külön összeállításra általában nincs szükség, mivel mágnesszalagon van állandóan tárolva és onnan beolvasható. Jelenlegi állapotában a leggyakrabban használt „Texas SN 74 N” integrált áramkörök adatait foglalja magában, de tetszés szerint bővíthető.

#### d) Kötési lista

Ez tartalmazza a báziselemek közti összekötések leírását, azaz megadja a fémesen összekötendő pontok listáját. Egy ekvipotenciális vezeték leírásában szerepelnek az összekötendő elemek (báziselem azonosító és a láb sorszáma), egy jelzés arra vonatkozóan, hogy a jelet ki kell-e vezetni a kártya valamelyik érintkezőjére, valamint a jel azonosítója, ha ilyen van (a báziselemek lábait itt egyelőre csak feltételes sorszámmal kell ellátni).

Az összeállító és elhelyező program a közölteken kívül még további adatokat is igényel, amelyek részben a kártya konstrukciós felépítésére vonatkoznak, részben pedig a tervező előírásait tartalmazzák. Ilyenek lehetnek a következők:

#### e) Konstrukciós adatok

- az IC tokok megengedett helyei,
- az érintkezők helye.

Ezeket a konstrukciós adatokat a program tartalmazza, tehát nem lehet őket lyukkártyán megadni. A program esetleges általánosítása esetén azonban bemeneti adatok lehetnek.

#### f) Kezdeti adatok

- bizonyos báziselemek előírt helye,
- tiltott helyek,
- érintkezők kijelölése,
- a diszkrét elemek adatai.

Amennyiben a kártyán ilyenek is szerepelnek, itt kell megadni a bekötésükre és elhelyezésükre vonatkozó előírásokat.

A program a gépen belüli leírás, a konstrukció és a kezdeti adatok alapján a megengedett helyek mindegyikére kijelöl egy vagy több báziselemet. Több báziselemet akkor, ha a kiválasztott típusból több fér el egy IC tokban. Ily módon egyidejűleg történik az IC tokok összeállítása és az elhelyezési feladat megoldása.

A báziselemek és a beültetendő helyek kiválasztása egy célfüggvény szerint történik. A célfüggvény minden báziselemhez hozzárendel egy számértéket, és a program a legnagyobb értékű báziselemet (vagy báziselemeket) helyezi el a soron következő helyek valamelyikére. Így egy kezdeti elhelyezést kapunk, amely minden megengedett helyhez hozzárendel egy IC tokot és megadja, hogy egy IC tokon belül milyen báziselemek milyen elrendezésben foglalnak helyet.

Ezután egy optimáló eljárás az egyenértékű bemenetek cseréjével igyekszik javítani az elhelyezéset úgy, hogy csökkenti az átmenő furatok számának várható értékét és a vezetékek várható hosszát.

A program eredménye a sornyomtatón kapott elhelyezési terv. Ez a kártya vázlatos rajza, amely a megengedett helyeken mutatja az ott elhelyezett IC típusszámát és a benne levő báziselemek azonosítóit. Ugyanennek a rajznak az adatai táblázatosan, összesítve is megkaphatók. A további tervezés számára a gépen belül tároljuk az elhelyezési és összekötési adatokat, most már nem báziselemekre, hanem IC tokokra vonatkoztatva.

Az eredmények kinyomtatása után a tervező dönthet, hogy elfogadja-e a kapott megoldást, vagy pedig módosításokat kíván végrehajtani. Az utóbbi esetben a változtatásokat új kezdeti adatként kell kezelni, és ezzel a programot újra kell futtatni. Az így létrejött ismétlések száma tetszőleges lehet. Ha az eredmények elfogadhatók, akkor át lehet térni a következő programra, amely megtervezi az összekötő vezetékek útvonalát.

## 1.2 Útvonalak tervezése

Az útvonalakat tervező programrész az elhelyezendő és összekötő program eredményeit használja fel, amelyekből a következőkre van szükség:

- a) Elhelyezési terv:  
az IC tokok helye a kártyán.
- b) Összekötési terv:  
az összekötendő pontok listája. Ebben az IC tokok lábainak sorszáma, valamint egy jelzés szerepel; hogy ki kell-e vezetni a jelet a kártya érintkezőire.

A program számára a következő adatokat kell még a kártyán megadni:

- c) Konstruktív adatok:
  - az összekötő vonalak szélessége,
  - a furatmezők átmérője,
  - az érintkezők méretei,
  - a különböző vezetékek közti megengedett legkisebb távolság stb.
- d) Kezdeti adatok:
  - vezeték számára tiltott helyek (felirat, furat, érintkező),
  - előre elhelyezett vezetékek helyei (föld, tápvezeték stb.),
  - adott jelhez kijelölt érintkező sorszáma,
  - foglalt vagy tilos érintkezők sorszáma,
  - a kártyára kerülő diszkrét elemek bekötése.

A program a felsorolt adatok birtokában kijelöli az összekötő vezetékek útvonalait és erről sornyomatón vázlatot készít. Előfordul, hogy a program nem tudja valamennyi vezeték számára megadni az útvonalat, ezeknek a vezetékeknek az adatait a gép kinyomtatja. A tervező ezek után ki tud törölni már elhelyezett vezetékeket és el tud helyezni el nem helyezett vezetékeket a gép által készített nyomtatási terv alapján. Esetleg új kezdeti adatokkal újra kezdi az útvonal tervezését vagy akár az elhelyező és összekötő programot is. A módosítások után előállított elfogadható nyomtatási terv leírását a gép tárolja a további tervezési szakaszokban történő felhasználás céljára.

## 2. A nyomtatási rajz elkészítése

A tervezés automatizálásának munkái során már 1970-ben elkészült és azóta is állandó használatban van a *TAL-1* program, amely a nyomtatási terv adatai alapján elkészíti a nyomtatási rajzot, illetve pontosabban: vezérlő lyukszalagot készít a rajzgép számára. A program úgy készül, hogy kézzel készített nyomtatási tervek feldolgozására legyen alkalmas, de a későbbiekben felhasználható legyen mint a teljes tervezés automatizálási programrendszernek egy részprogramja.

A program alkalmazása révén már most is időmegtakarítás érhető el: a nyomtatási rajz szerkesztési munkáinak egy része és a teljes rajzolás munkája géppel végezhető. A nyomtatási rajz tervezését és a vázlat készítését továbbra is a tervezőnek kell elvégez-

nie, de a program felhasználása ebben a szakaszban nem tesz szükségessé semmilyen változtatást a megszokott módszerekhez képest.

### 2.1 A nyomtatási terv előkészítése

Ahhoz, hogy a *TAL-1* program számára a nyomtatási terv adatai könnyen megadhatók legyenek, a tervet meghatározott szabályok szerint kell elkészíteni, ez a forma azonban megegyezik az általában szokásos nyomtatási tervekével.

A nyomtatási tervet (vázlatot) négyzethálójával (raszterrel) ellátott úrlapra kell rajzolni oly módon, hogy arról valamennyi vezeték és vezetékalakzat (furat, érintkező, felirat stb.) helye pontosan leolvasható legyen. E célból koordináta-rendszert kell a rajzhoz hozzárendelni úgy, hogy origója a kártyarajz jobb alsó sarkával essék egybe, a koordináták egysége pedig a rasztervonalak távolságával legyen egyenlő.

A kártyán csak egyenes szakaszokból álló vezetékek helyezhetők el, és az egyes szakaszok kezdő- és végpontjai csak egész koordinátájú pontok lehetnek. Ugyancsak egész koordinátájú pontokban lehet elhelyezni a furatok és felirati jelek középpontját. Kétoldalas nyomtatású kártyáknál jelölni kell, hogy a vezeték a kártya melyik oldalán halad.

Sok előírás lehetővé teszi, hogy a vezetékek a rasztervonalaknál sűrűbben haladjanak, és így töréspontok nemcsak egész koordinátájú pontokban lehetnek. A program ezt figyelembe veszi, és elfogad olyan koordinátákat is, amelyek törtrészt is tartalmaznak, a törtrész azonban csak az egység  $1/4$ -e vagy annak egész számú többszöröse lehet.

### 2.2 Adatok megadása

A nyomtatási vázlat adatait 80 oszlopos adatlapokra kell rávezetni, majd ezekről történik a lyukkártyák lyukasztása.

A program számára a következő adatokat kell megadni:

- a) Általános adatok  
A kártya méretei és a rajzolásra vonatkozó vezérlő adatok.
- b) Modulok azonosítói és helyei  
Az IC tokok lábfuratainak megrajzolásához szükséges adatok.
- c) Egyéb elemek és helyeik  
A diszkrét elemek lábfuratainak megrajzolásához szükséges adatok.
- d) Felirati jelek és helyeik  
A kártyára kerülő feliratok megadása.
- e) Vonalak helye  
A különböző vonalas vezetékek leírása.
- f) Ismétlés kérése  
Javítás utáni újrafuttatás esetére.

Abban az esetben, ha több olyan kártya rajzát kell elkészíteni, amelyen ismétlődő részek (IC tokok lábfuratai, föld- és tápfeszültség vezeték, érintkezők stb.) vannak, akkor nagyon sok idő megtakarítható a második és további kártyák rajzolásánál. Az ismétlődő részek adatait csak egyszer kell megadni, és er-

ről külön lyukszalagot lehet készíteni. Így minden kártyánál csak a változó részek adatait kell külön leírni és feldolgozni. Rögzített konstrukciós előírások vagy szabványok használata nagyon előnyös, mert így rendszerint csak a vonalak és feliratok helyét kell egy-egy kártyánál megadni, a többi adat minden kártyánál ugyanaz.

A nyomtatott kártyára rajzolható vonalas vezetékeket a kezdő- és végpontok, valamint a közbeeső töréspontok, átmenő furatok és láb furatok sorozatával kell megadni. A pontokat olyan sorrendben kell felírni, ahogyan azokat össze kell kötni. Egy-egy pont megadásához a következő adatok szükségesek:

- a pont típusa (láb furat, átmenő furat vagy töréspont),
- a pont helye (koordinátákkal vagy elemnévvel és lábszámmal megadva),
- a kártya oldalának jele (melyik oldalon halad a vezeték).

### 2.3 Futtatás, eredmények

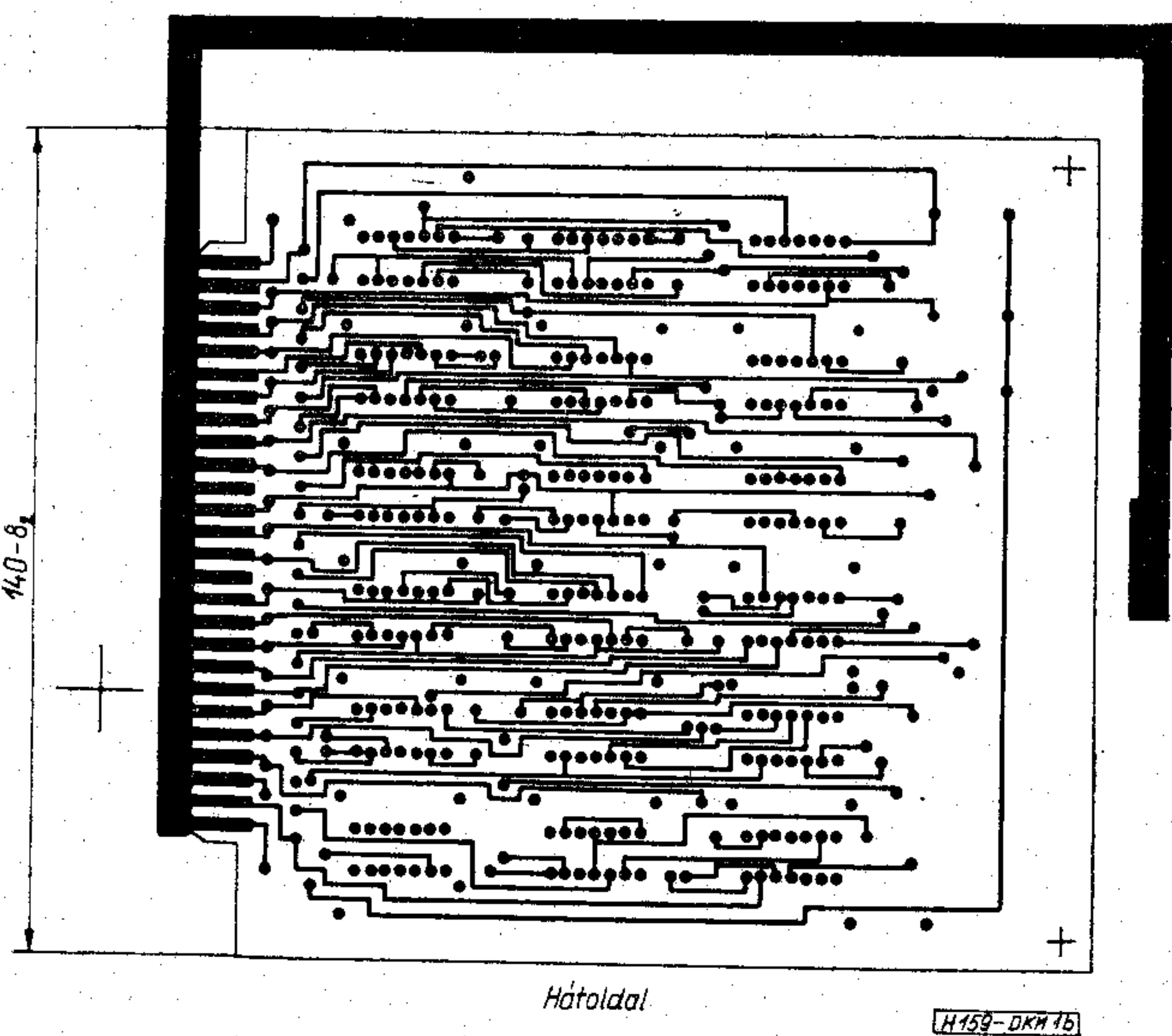
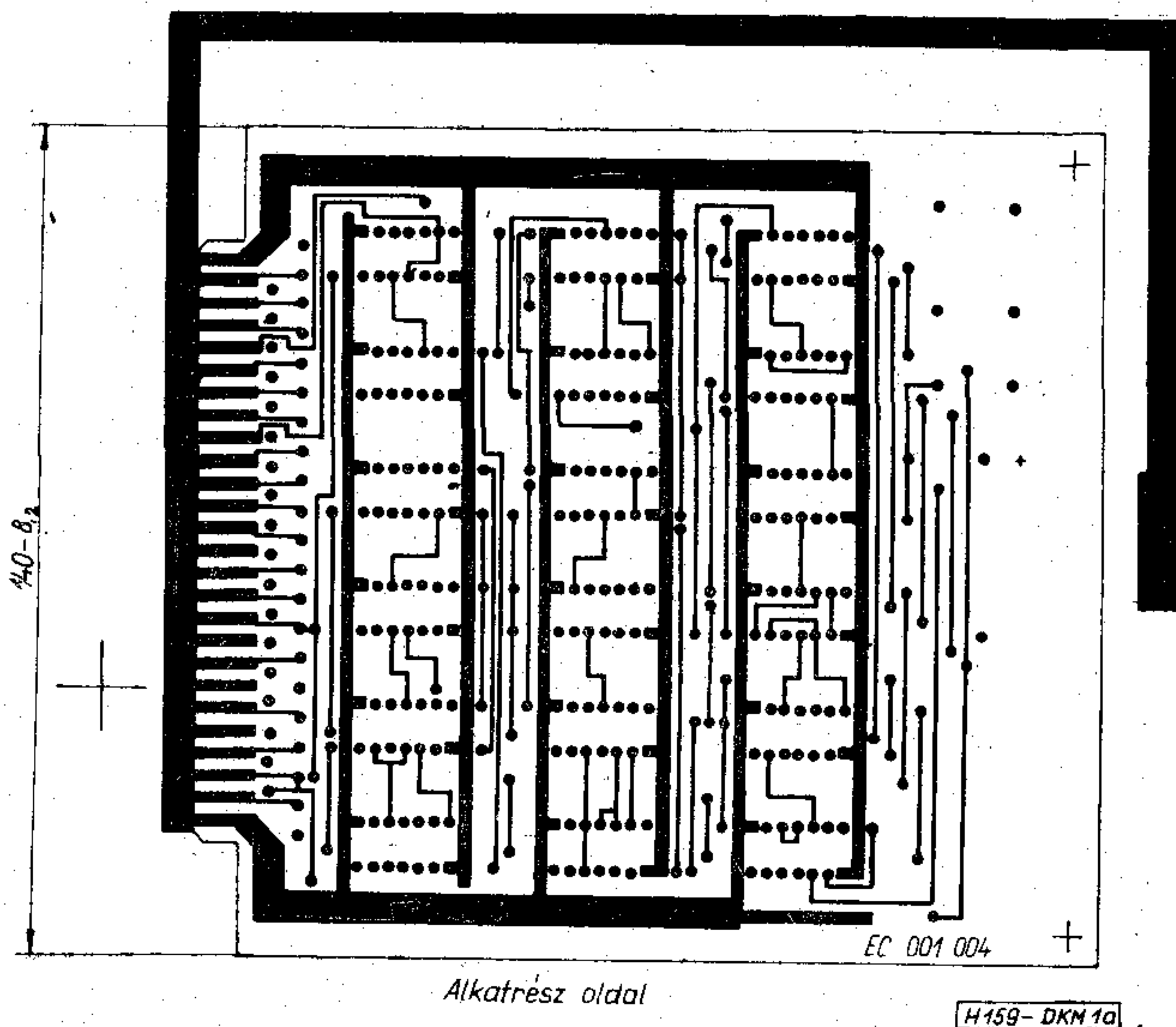
Az adatok beolvasásával egyidejűleg a program szintaktikai ellenőrzést is végez, majd a futtatás folyamán további ellenőrzésre kerül sor. Összesen 38-féle hibát képes észrevenni és kijelézni. A hibák kijelzése után a program futása leáll, és a javított adatokkal újra kell kezdeni a futtatást.

Helyes működés esetén a program kiírja a nyomtatott kártya adatait, az elhelyezett elemek típusait és helyüket a kártyán. Ezután még egy ellenőrzés van mód, amelyet azonban csak a felhasználó kérésére, azaz egy vezérlő adat megadására végez el a program. Elő lehet írni ugyanis, hogy a program sornyomtatón készítse el a kártya nyomtatási vázlatát. Ebből teljes biztonsággal meg lehet állapítani, hogy az adatmegadás során történt-e hiba vagy sem. Ha az ellenőrzések során nem történt hibajelzés, illetve a nyomtatási vázlat megegyezik a kézzel készített nyomtatási tervvel, akkor a program elkészíti a rajzgép vezérléséhez szükséges lyukszalagot. Jelenleg a ZUSE Graphomat rajzgép vezérléséhez szükséges formára tudja a program a vezetékek belső leírását átkódolni, de csekély munkával más gépek vezérlőutasításait is elő lehet állítani. A Graphomat rajzgépen készített nyomtatási rajz látható az 1. ábrán.

A programot az SzKI Siemens 4004/45-ös gépre írtuk, legnagyobb részben FORTRAN, egyes szubrutinokat ASSEMBLER nyelven. A futási idő az adatok beolvasásával együtt körülbelül 1 perc. A nyomtatási vázlat készítése sornyomtatón 3–4 perc, míg a lyukszalag készítése 10–15 perc. A futási idők kb. 20 IC tokot tartalmazó, átlagos bonyolultságú kártyára vonatkoznak. A lyukszalag készítési idejének kb. 2/3-ad része a láb furatok és érintkezők adatainak lyukasztása. Tipizált kártyák esetén ezt elég egyszer elkészíteni, és a további kártyáknál a lyukszalagot újra fel lehet használni. A lyukszalag készítési ideje így 3–4 percre lecsökkenthető.

### 2.4 Hasznosítás

A TAL—1 programot elkészítése óta alkalmazzák külső felhasználók is. A féléves felhasználás során a



1. ábra

program következő előnyös tulajdonságai bizonyosodtak be:

#### a) Hibátlan rajz

Ha a nyomtatási terv készítésekor és az adatmegadásban nem történt hiba, akkor a rajz is hibátlan lesz. A rajzoló mint hibaforrás így kiküszöbölődött. Igaz ugyan, hogy helyette az adatelőkészítés során van lehetőség a hibák elkövetésére, de ezeket a hibákat a számítógép megbízhatóan kijelzi.

#### b) Pontos rajz

A kétoldalas nyomtatású kártyáknál problémát jelent a két oldal nyomtatási rajzának pontos illesztése. Kézi rajzoláskor lehetetlen a kívánt, kb. 50  $\mu$ -os pontosságot elérni. Rajzgép alkalmazásával, 2:1 nagyítási arány esetén ez az érték betartható. Így megszűnt a rajzok és a kész kártyák utólagos javítása, káparása.

#### c) Munkaerő és munkaidő megtakarítás

Az adatelőkészítésre és a futtatásra fordított idő kártyánként kb. 10 órával kevesebb, mint kézi rajz-

zolás esetén. A megtakarítás csaknem azonos a teljes rajzolás idejével, mert az adatelőkészítésre fordított idő megtérül a rajz és a kártya utólagos javításának elmaradása révén. A számítógépes módszernél az adatelőkészítés során elkövetett hibák javítása, a program újbóli futtatása jelent idővesztést, amely azonban kellő gyakorlattal kiküszöbölhető, vagy erősen csökkenthető.

#### d) Költségek

Az automatizált kártyarajzolás a tapasztalatok szerint nem jelent megtakarítást a közvetlen költségekben. Ha egybevetjük a kézi rajzolásra fordított munkaerő árát a számítógépes módszer költségeivel (adatelőkészítés, számítógép ideje, rajzgép ideje), akkor a két költség kb. azonos. Sok hiba esetén (többszöri futtatás a számítógépen) a kézi módszer olcsóbb, tipizált kártyák csomagban történő feldolgozása viszont a gépi eljárást mutatja előnyösebbnek. Figyelembe kell venni azonban, hogy a munkaidőben jelentkező megtakarítás esetleg egy határidő elcsúszását akadályozza meg, a hibátlan és pontos rajzok a legyártott berendezés élesztésénél okoznak jelentős könnyebbséget. Az így elért megtakarítás nem mindig mérhető, de bizonyosan az automatizált módszer javára billenti a mérleget a költségek szempontjából is.

Ebből a szempontból még további javulás várható a teljes kártyatervezés automatizálásának megoldása után.

### 3. Megjegyzések

A nyomtatási rajzot készítő *TAL-1* programmal szerzett tapasztalatok azt mutatják, hogy annak alkalmazása a felhasználók számára sok szempontból előnyös. A program jelenleg is bármely intézmény számára hozzáférhető.

A számítógépes módszer előnyei sokkal jobban megmutatkoznak, ha a kártyák tervezését is géppel lehet végezni. Ezt a feladatot a *TAL-3* nevű program oldja meg, amely 1971. decemberében készült el, és 1972. első negyedévében került használatba.

A *TAL-3* program tovább csökkenti a digitális berendezések tervezőinek és gyártóinak problémáit, és nagy lépést jelent előre a műszaki tervezés teljes automatizálása felé.

- [1] *Shostack, K.*: Computerized methods for the routing of printed circuit boards. *Computer Design*, July 1970. pp. 80, 82, 84, 87.
- [2] *Wiseman, N. E.*: Application of list-processing methods to the design of interconnections for a fast logic system. *The Computer Journal*, January 1964. pp. 321—327.
- [3] *Dunne, G. V.*: The design of printed circuit layouts by computer. *Proc. 3rd Australian Computer Conference*, May 1966. pp. 419—423, 442.
- [4] *Saedler, J.—Schäller, K.*: Zu einigen Problemen der rechnerunterstützten Konstruktion von Leiterplatten. *Nachrichtentechnik*. H. 4. 1970. pp. 153—160.
- [5] *Götz, E.—Pabst, W.*: Automatisierte Herstellung von Unterlagen zur Leiterplattenfertigung für Elektronische Schaltungen. *Werkstatt und Betrieb*. H. 10. 1968. pp. 597—602.
- [6] *Bader, W.*: Das topologische Problem der gedruckten Schaltung und seine Lösung. *Archiv für Elektronik*, H. 1. 1964. pp. 2—12.
- [7] *Fisk, C. J.—Caskey, D. L.—West, L. E.*: ACCEL: Automated circuit card etching layout. *Proc. IEEE*. November 1967. pp. 1971—1982.
- [8] *Lee, C. Y.*: An algorithm for path connection and its applications. *IRE Trans. on EC*. September 1961. pp. 346—365.
- [9] *Akers, S. B.*: A modification of Lee's path connection algorithm. *IEEE Trans. on EC*. February 1967. pp. 97—98.
- [10] *Peze, F. A.*: Program for semi-automatic tracing of printed circuit connection. *IEEE Conference Publication*. No. 51. 1969. pp. 89—96.
- [11] *Leavers, D. F. A.*: The use of a graphical display in the automatic design of printed circuit boards. *IEEE Conference Publications*, No. 51. 1969. pp. 11—20.
- [12] *Koichi Mikami—Kinya Tabuchi*: A computer program for optimal routing of printed circuit conductors. *Information Processing 68*. North-Holland Publishing Co., Amsterdam, 1969. pp. 1475—1478.
- [13] *Aakhus, M. D.—Seeman, M. D.—Ptak, E. J.*: ACCLAIM: A computer aided design system. *Computer Design*. May 1968. pp. 64—71.
- [14] *Mamelak, J. S.*: The placement of computer logic modules. *Journal of ACM*, October 1968. pp. 615—629.
- [15] *Lass*: Automated printed circuit routing with a stepping aperture. *Comm. of ACM*, May 1969. pp. 262—265.
- [16] *Шуро, Г. Э.—Ланудис, Л. И.*: Метод трассировки печатных соединений. В сборнике статей «Применение вычислительных машин для проектирования цифровых устройств», ред. Матюхин, Н. Я. Изд. «Советское Радио», Москва, 1968. стр. 199—215.

### SZEMLE

(Folytatás a 209. oldalról)

Már napjainkban is észlelhető az aktív félvezető eszközökön belül, az integrált áramkörök fokozatos térhódítása. 1969-ben az Amerikai Egyesült Államokban a félvezető eszközökből és integrált áramkörökből származó bevétel mintegy 1300 millió dollár volt. Ebből az integrált áramkörök kb. 40%-ban részesedtek. Az előrejelzések szerint az amerikai piacokon 1973-ban értékesítésre kerülő aktív félvezető eszközöknek több mint 60%-át az integrált áramkörök jelentik.

A fejlett nyugat-európai államokban — az amerikai források szerint — a nemzetközi élvonaltól mintegy 2—4 éves lemaradás mutatkozik. Nyugat-Európa 1969. évi félvezető-eszköz és integrált áramkör forgalmának csak mintegy 20%-át képezte az integrált áramkörök részaránya. Ennek ellenére 1973-ra e részarány 40—50%-ra való felfutásával számolnak.

A szocialista országok között a félvezetők és integrált áramkörök gyártásában és fejlesztésében a Szovjetunió jár élen. A többi országban általában 5—6 éves lemaradás jelentkezik a fejlett ipari országokhoz képest. Az elmúlt 2 évben jelentős előrelépést tettek Csehszlovákiában a félvezetőeszközök és egyes integrált áramkörök gyártása terén. Több digitális és analóg áramkör fejlesztését fejezték be. Jelentős erők koncentrációjával folyik Lengyelországban az integrált áramkörök fejlesztése. Bulgáriában 1971-ben indult meg a MOS-típusú félvezető áramkörök kísérleti gyártása. Az NDK e vonatkozásban különleges helyet foglal el. Fejlődését több licenc és know-how vásárlással siettette, a félvezető és integrált áramkör gyártás megszervezésében erőteljesen támaszkodott a viszonylag magasszintű finommechanikai és gépipari adottságokra. (*KGM—MTTI információ, 1971.jan.*)

(Folytatás a 221. oldalon)



# Hátoldali huzalozástervezése. A TAL-2 program

ETO 621.049.75.001.2:681.3.06 TAL-2

A kártyák nyomtatási tervének elkészülése után a következő feladat a kártyák közti huzalozás tervezése. A huzalozási tervek elkészítéséhez szükséges kiindulási adatok egyrésze a kártyára bontás során, a másik része pedig a kártyák nyomtatásának tervezése során állt elő.

A kártyára bontás eredménye a kártyák közti összekötések táblázata, amely megadja, hogy az egyes jelek melyik kártyákat kötik össze. A nyomtatás eredményeként pedig meg lehet tudni, hogy egy adott jelet a kártya milyen sorszámú érintkezőjéhez kell csatlakoztatni. A tervezés feladata a kártyák kereten belüli elhelyezkedésének a meghatározása és a közvetlenül huzallal összekötendő pontpárok kijelölése, mindez valamely optimalitási feltételt kielégítve.

A következőkben foglalkozunk a probléma egy elméleti megoldási lehetőségével, majd ismertetjük az eljárást alkalmazó TAL-2 programot.

## 1. Az elhelyezési probléma egy megoldása

A blokszerű felépítésű elektronikus berendezéseknél fontos feladat a keretekben elhelyezkedő kártyák csatlakozói közti huzalozás optimális megtervezése. Az általunk kitüntetett optimalitási elv a következő: a kártyák egymáshoz képesti elhelyezkedése és az előírtan ekvipotenciális érintkezők közti huzalösszekötések minimális összhuzalhossz eredményezzenek, csavart huzalozási (wire-wrap) technológiára.

Adott kártyaelhelyezésnél az ekvipotenciális érintkezők közti minimális összhosszúságú kötések a gráfelméletből ismert minimális fák megkeresésével kaphatjuk meg. Itt egy fa csúcsait az ekvipotenciális érintkezőcsúcsok, a fa éleit a csúcspárok közti huzalösszekötések, az élekhez rendelt hossz pedig az összekötött csúcspárok geometriai távolságai képviselik.

Legyen  $p$  egy adott elrendezés, ahol elrendezésen a kártyák és a keretekben levő pozíciók kölcsönösen egyértelmű egymáshoz rendelését értjük;  $f$  egy adott huzalozás, az ekvipotenciális érintkezőcsúcsok összekötéseinek egy megadása;  $\varphi$  a fák száma;  $v_i$  az  $i$ -edik fa csúcsainak száma;  $w_i$  az  $i$ -edik fához rendelt súly;  $e_{ij}^f$  az  $i$ -edik fa  $j$ -edik élének hossza, adott  $p$  és  $f$  mellett. Ekkor a feladat formálisan a következő: keresendő az a  $p=p^*$ ,  $f=f^*$ , melyre az

$$L(p, f) = \sum_{i=1}^{\varphi} \sum_{j=1}^{v_i-1} w_i e_{ij}^f \quad (1)$$

célfüggvény minimális értéket vesz fel.

## 1.2. Irodalmi áttekintés

A feladat bonyolultsága következtében mindeddig nem áll rendelkezésre számítástechnikailag gazdaságos, az abszolút optimumot megtaláló eljárás. Az irodalomban javasolt valamennyi módszer, a nagy mennyiségű számítás miatt eleve számítógépes feldolgozást igényel. Ezek az algoritmusok az (1) célfüggvényt illetően azzal az elhanyagolással élnek, hogy a keretekben elhelyezkedő kártyákat pontszerű elemekként kezelik. A feladatot a következőképpen fogalmazzák meg:

Adott  $n$  elem és  $n$  pozíció, ahol meghatározottnak és rögzítettnek tekintik az elempárok közti összekötéseket. Ha  $C(i, j)$  az  $i$  és  $j$  elemek közti összekötések száma,  $P(j)$  a  $j$  pozícióban elhelyezett elem,  $d(i, j)$  az  $i$  és  $j$  pozíciók egymástól mért távolsága, akkor keresendő az  $n$  elemnek az  $n$  pozícióhoz való egy egyértelmű  $p=p^*$  hozzárendelése, melyre

$$L(p) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n C\{P(i), P(j)\} d(i, j) \quad (2)$$

minimális. Látható, hogy a pontszerűségtől eltekintve (2) nem más, mint (1) rögzített  $f$ -re.

Ennek a feladatnak a közelítő megoldására több javasolt eljárás található az irodalomban. Ezek az eljárások alapjukban véve heurisztikusak.

A probléma egy másik megközelítése lehet az, hogy az elrendezést tekintjük meghatározottnak és ehhez keresünk egy olyan huzalozást, mely minimális vezetékosszat biztosít. Ezt a feltételt maradéktalanul kielégítik a minimális fák, viszont ettől még az eredmény az optimumtól távolinak is adódhat. Itt nyilván arról van szó, hogy (1)-ben  $p$ -t rögzítettük. Mivel összesen  $n!$  különböző elrendezés létezik  $n$  kártyára és  $n$  pozícióra, azonnal látható, hogy nem biztosíthat kielégítő optimumkeresést ezek közül egynek a figyelembevétele. Ez annál is inkább így van, mivel eddigi tapasztalataink szerint a gyakorlatban a legtöbb fa kevés élből áll, s ilyenkor a megoldás jóságát főleg az elrendezés, és nem a fák felépítése szabja meg.

Ami  $f$  rögzítését illeti, az összes fa lehetséges felépítéseinek száma

$$\sum_{i=1}^{\varphi} v_i^{v_i-2}$$

és ebből a  $\varphi$  számú fára egy-egy felépítést választanak kiindulásként. Erre az esetre javasolt algoritmusok már nem tudják egyértelműen biztosítani az adott huzalozáshoz tartozó optimális elrendezést, nem beszélve arról, hogy a pontszerű közelítés igen durva elhanyagolást jelent, másrészt  $f$  megválasztására nincs semmiféle támpont.

1.2. Eljárás adott feladat megoldására

Az (1) általános célfüggvényt a következő feltételek mellett kívánjuk optimalizálni:

a) A keret két egymás fölött lévő, egyenként 20 kártyapozíciót tartalmazó sorból áll.

A megoldás során két, a kártyák elhelyezésére vonatkozó lehetséges megkötést kell figyelembe venni:

- b) Egy vagy több kártya előírt pozícióba kerüljön véglegesen.
- c) Két vagy több kártya egymás melletti, illetve egymás alatti pozícióba kerüljön.

A keret bemeneti, illetve kimeneti jelei a keretek mellett rögzített közvetítő csatlakozókról futnak a kártyacsatlakozókra, ezért amennyiben kívánatos,

- d) az ilyen közvetítő csatlakozókat úgy kell tekinteni, mint a b) szerint rögzített csatlakozókat, s ekkor a kártyák érintkezőcsúcsaiból hozzájuk menő vezetékek hosszai beleszámítanak a célfüggvénybe.
- e) A csavart huzalozás megvalósíthatósága érdekében egy érintkezőcsúcsra maximálisan 3 vezeték köthető, vagyis a fák csúcsainak fokszáma 3-nál nem lehet nagyobb.

Az utóbbi feltétel figyelembevételéül úgy történik, hogy először felépítjük az e) pont szerinti korlátozást nem tartalmazó minimális fákat, aztán a 3-nál nagyobb fokszámú csúcsokról azokat a többletet jelentő éleket „választjuk le”, amelyek helyett új megengedhető és a minimálisához legközelebbi összhuzalhosszt biztosító élek vehetők fel.

Az alábbiakban leírt eljárás az irodalomban ismertektől eltérően az elrendezést és a huzalozást nem egymástól függetlenül kezeli, s ezáltal a végső huzalozás csak a végső elrendezésnél jön létre. Ezzel azokat a hátrányokat kívánjuk elkerülni, melyek a célfüggvény egyik változójának egyszer s mindenkorra való rögzítéséből adódnak.

A további tárgyalásunk folyamán mind a kártyák csatlakozóit, mind a közvetítő csatlakozókat egységesen kezeljük, közös nevük: csatlakozó. Minden kártyát egy csatlakozóval reprezentálunk, ahol az illető kártya beültetésén a csatlakozójának egy keretpozícióhoz való hozzárendelését értjük. Kiindulásként ismereteseek azok az adatok, melyek előírják, hogy mely csatlakozók érintkezőcsúcsai ekvipotenciálisak. Az ilyen egy fához tartozó csúcsokról azt is mondjuk, hogy egy jelhez tartoznak, ahol a fához rendelt súly a jel súlya. A jelek száma tehát azonos a fák számával,  $\varphi$ -vel. Rendeljük minden csatlakozóhoz egy  $\varphi$  komponensből álló ún. jelvektort, amelyek  $i$ -edik komponense az  $i$ -edik jel súlya, ha az  $i$ -edik jel szerepel a csatlakozón, különben zérus. Legyen a  $k$ -adik csatlakozóhoz rendelt jelvektor  $S_k$ , ennek az  $i$ -edik komponense pedig  $c_{ki}$ .

A jelvektorokra definiáljuk a  $\oplus$  és  $\odot$  műveleteket a következőképpen:

az  $S_k \oplus S_l$  egy olyan  $\varphi$  komponensű jelvektor, melynek  $i$ -edik komponense

$$c_{si} = c_{ki} = c_{li}, \text{ ha } c_{ki} = c_{li}, \text{ másrészt } c_{si} = c_{ki}, \text{ ha } c_{li} = 0, \text{ ill. } c_{si} = c_{li}, \text{ ha } c_{ki} = 0.$$

$$S_k \odot S_l = \sum_{i=1}^{\varphi} c_{pi}, \text{ ahol}$$

$$c_{pi} = c_{ki} = c_{li}, \text{ ha } c_{ki} = c_{li}, \text{ ill. } c_{pi} = 0, \text{ ha } c_{ki} \neq c_{li}.$$

Az alábbiakban leírt eljárás a szekvenciális optimalizálás elvét követi. Az eljárás kiválasztási és beültetési fázisok egymást követő sorozatából áll, ahol az előbbi fázis egy vagy több csatlakozó beültetésre való kijelölését jelenti, az utóbbi pedig a beültetés végrehajtását.

Legyen a csatlakozók számozása olyan, hogy ha összesen  $g$  számú rögzített csatlakozó van, akkor ezek 1-től  $g$ -ig számozzanak, a nem rögzített csatlakozók pedig  $g+1$ -től  $n$ -ig.

Nevezük gócnak a keret azon pozícióit, melyek egy rögzített csatlakozó körül helyezkednek el.

Előírjuk, hogy mindegyik góc olyan pozíciómezőt foglaljon le, mely nem tartalmaz közös pozíciót semelyik más góc pozíciómezőjével. Két pozíciót szomszédosnak mondunk, ha a befoglaló téglalapjaik valamelyik oldala közös. Egy góc mezőjében lefoglalt, ill. üres pozíciók vannak aszerint, hogy van-e oda beültetett csatlakozó, illetve nincs. A góc lefoglalt pozícióinak befoglaló téglalapjai területileg zárt egyseget képviselnek: mindegyik téglalapnak van közös éle egy másikkal. A góc üres pozícióit a lefoglalt mező határoló oldalaival közös oldalú téglalapok jelölik meg.

A kiválasztási és beültetési fázis állandóan ismétlődik, mígnem az összes csatlakozó beültetésre kerül. Egy lépésnek fogjuk nevezni egy kiválasztás és beültetés végrehajtását.

Legyen  $Q = \{1, 2, \dots, n\}$  a beültetendő kártyák halmaza. Legyen  $s$  lépés után ( $s=0, 1, 2, \dots$ )

$$B(s) \cup K(s) = Q,$$

ahol  $B(s)$  az  $s$ . lépés után beültetett kártyák halmaza,  $K(s)$  a még be nem ültetett kártyák halmaza. Legyen továbbá  $G(s) = \{1, 2, \dots\}$  az  $s$ . lépés utáni gócok halmaza. Jelöljük  $U_k(s)$ -sel az  $s$ . lépés után a  $k$ . góchoz tartozó jelvektorok  $\oplus$  összegét.

I. Tegyük fel, hogy  $g > 1$  és kiindulásként legyen  $G(0) = \{1, 2, \dots, g\}$ ,  $B(0) = G(0)$ ,  $U_k(0) = S_k$ ,  $k \in G(0)$ . Az  $s$ . lépésben minden  $k \in G(s-1)$  és  $l \in K(s-1)$ -re kiszámoljuk a következő  $\eta_{kl}$  értékét.

$$\eta_{kl} = U_k(s-1) \odot S_l - \sum_{j \in G(s-1), j \neq k} U_j(s-1) \odot S_l,$$

ahol minden rögzített  $k$  érték mellett  $l$  felveszi a  $K(s-1)$  halmaz értékeit.

Legyen

$$\beta_k = \max_l \eta_{kl} \tag{3}$$

és végül legyen

$$\beta_j = \max_k \beta_k. \tag{4}$$

Látható, hogy (3) és (4) az  $s$ . lépésben egy vagy több jelvektor kiválasztását írja elő. Amennyiben (4)-nél több maximumhely van (nemcsak egy  $j$  értékre teljesül (4)), a következő szelektálást hajtjuk végre:

Ha egy  $j$  jelű maximumhelyhez több  $S_l$  tartozna, ezek közül azt választjuk ki, melyre az

$$U_j(s-1) \odot S_l \quad (5)$$

érték a legnagyobb. Ha nemcsak egy ilyen legnagyobb érték adódna, további szelekciós feltételként a megmaradt jelvektorokra számítandó

$$\delta_l(s) = S_l \odot S_l - \sum_{i \in G(s-1)} U_i(s-1) \odot S_l \quad (6)$$

és amelyekre ezek közül  $\delta_l(s)$  minimális, azt rendeljük az illető  $j$ -hez. Ha még így is egynél több jelvektor választódna ki, akkor most már ezek közül az egyiket jelöljük ki tetszőleges választással.

Ha több  $j$ -hez egy  $S_l$  tartozna, úgy a jelvektorhoz tartozó csatlakozó ezen lépésben nem kerül kiválasztásra. Az ilyen jelvektort semlegesnek nevezzük. Ilyenkor azokra a  $j$ -kre, amelyekre még „maradt” olyan jelvektor, ami nem semleges, s egynél több ilyen van, az előbb leírt pótszelekcióval választjuk ki a megfelelőt. Ha valamennyi  $j$ -hez csak semleges jelvektor választható ki, akkor tetszőleges kijelöléssel az egyik ilyen jelvektort valamelyik neki megfelelő  $j$ -hez rendeljük.

Miután a kiválasztás megtörtént, azokra a  $j$  értékekre, melyekhez van kiválasztott jelvektor, legyen

$$U_j(s) = U_j(s-1) \oplus S_l \quad (7)$$

és

$$l \in B(s)$$

míg az összes többi  $k \in G(s-1)$  értékre továbbra is

$$U_k(s) = U_k(s-1). \quad (8)$$

A (7) összefüggésről: az  $l$ -edik csatlakozó a  $j$ -edik góchoz került, ahol  $S_l$  az  $s$ -edik lépésben a  $j$ -edik góchoz maximálisan kötődő jelvektor, míg  $U_j(s)$  a  $j$ -edik góc  $s$ -edik lépés utáni ún. egyesített jelvektora.

Nem adtunk még szabályt az  $s$ . lépés utáni  $G(s)$  halmaz rekurzív meghatározására. Ehhez előbb az  $s$ -edik lépéshez tartozó beültetési és huzalozási eljárást kell ismertetnünk.

Valamely góchoz kiválasztott csatlakozó beültetési pozícióját a következő módon határozzuk meg: A csatlakozót elhelyezzük a góchoz tartozó üres helyek mindegyikére, és egy-egy ilyen helyen felépítjük a megfelelő minimális fákat az adott elrendezéshez. Ezután a csatlakozót a góc lefoglalt mezőjében levő egyik csatlakozó helyére tesszük be, míg az ott levő csatlakozót a góc összes üres helyére beültetve külön-külön mindegyik elrendezésnél felépítjük a megfelelő minimális fákat. Ezt az eljárást a beültetésre kiválasztott csatlakozóval a góc összes előzőleg lefoglalt pozíciójára (a rögzített csatlakozóé kivételével) végrehajtva, a végső beültetés ebben a lépésben az lesz, amelyik az előbb leírtak közül a legrövidebb összhuzalhosszt eredményezte.

Ha egy adott lépésben egynél több góchoz választódott ki csatlakozó, úgy ezek beültetését ebben a lépésben kell végrehajtani, de egyszerre csak egy gócra végezzük el a beültetést. Ha az  $s$ -edik lépésbeni összes beültetés után két gócra vagy többre nézve szomszédos üres vagy lefoglalt pozíció keletkezne, akkor a továbbiakban ezek a gócok egyazon gócnak

tekintendők. Ha az  $i$ - és  $j$ -edik gócról van szó  $i < j$ -re, akkor a gócok sorszámait tartalmazó

$$G(s-1) \text{ és } G(s)$$

abban és csakis abban különböznek, hogy  $G(s)$  már nem tartalmazza a  $j$  elemet. Másrészt a két egyesített gócra  $U_j(s)$  a két góc (7) és (8) szerint számított egyesített jelvektorainak  $\oplus$  összege lesz. Hasonlóan járunk el kettőnél több góc egyesítésekor.

A fent leírt eljárást mindaddig folytatjuk, míg az egyesítések egyetlen gócot nem eredményeznek. Amint ez bekövetkezik, azon túl a

$$P_l(s) = \max_{l \in K(s-1)} (U_l(s-1) \odot S_l) \quad (9)$$

feltétel szerint választjuk ki a beültetendő  $l$ -edik csatlakozót. Ha (9)-nél több maximum adódna, azok közül azt választjuk, amelyikre a (6) feltétel szerint minimumot kaptunk. Ha még ezek után is több csatlakozó maradna, közülük szabad választással jelölünk ki egyet. Ennek beültetése ugyanúgy történik, mint több góc esetén. Mihelyt az összes csatlakozó beültetést nyert, az egész eljárás befejeződik.

**II.** Ha a kiindulásnál egyetlen rögzített csatlakozó van, akkor ugyanúgy járunk el végig, ahogy az egy gócra redukálódott esetben tettük.

**III.** Ha egyetlen csatlakozót sem kell rögzíteni (ilyenkor természetesen a közvetítő csatlakozókkal sem törődünk), akkor elsőként az a csatlakozó kerül kiválasztásra, amelyen a legtöbb jel van. Ha több ilyen csatlakozó van, tetszőlegesen választunk közülük.

A további kiválasztás ugyanúgy történik, mint egy góc esetén. Ugyanez vonatkozik a beültetésre is, azzal a különbséggel, hogy itt nem lesz rögzített csatlakozó, továbbá a két kártyasort úgy tekintjük, hogy végtelen számú pozíciót tartalmaznak. Ekkor csak arra kell tekintettel lennünk, hogy az összes csatlakozóra legvégül kialakuló beültetést a két egymás fölötti 20—20 pozíciós sorba lehessen befoglalni.

### 1.3. Egy korlátozás figyelembevétele

Ha két vagy több csatlakozónak előírás szerint egymás mellett kell elhelyezkedni, akkor ezen csatlakozók jelvektoraihoz egy fiktív, a többi összetevőhöz képest több nagyságrenddel nagyobb kiegészítő összetevőt rendelünk, és előírjuk, hogy az ehhez tartozó fiktív jel a csatlakozók geometriai középpontján elhelyezkedő fiktív érintkezőcsúcsokon legyen.

Kötelező egymás alatti elhelyezkedés esetén a felső sorban levő csatlakozók alsó részén, az alsó sorban levő csatlakozók felső részén definiáljuk az extrém nagy súlyú fiktív jelet tartalmazó érintkezőcsúcsot.

Annyi pótlólagos fiktív összetevővel kell a jelvektorokat bővíteni, ahány — a leírt módon egymáshoz rendelt csatlakozókból álló — csoport létezik.

Ezáltal az előző fejezetben ismertetett elhelyezési és huzalozási eljárással egységesen kezelhetjük a csatlakozókat: a jelvektorok fiktív összetevői biztosítani fogják az előírtan szomszédos csatlakozók egymást közvetlenül követő kiválasztását, a fiktív érintkezőcsúcsok és a faélek pedig a megfelelő beültetésüket.

## 2. A TAL-2 program

Az SZKI-ban 1970-ben elkészült a TAL-2 nevű program, amely kártyablokkok hátoldali huzalozását tervezi meg. A program jelenleg önállóan használható, a bemenő adatokat lyukkártyáról olvassa be. A későbbiek során a teljes konstrukciós tervezést megoldó programrendszer egy részprogramja lesz, és fel tudja használni a már korábban kapott és gépen belül tárolt eredményeket is.

A TAL-2 program elvégzi a kártyák kártyablokkokban történő elrendezését és megtervezi a kártyák érintkezői közti wire-wrap huzalösszekötéseket úgy, hogy a kártyablokkra kiadódó összes huzalhossz minimális legyen. Eközben figyelembe vesz bizonyos előírásokat és korlátozásokat.

A program eredménye egy elhelyezési rajz és egy összekötési táblázat, amely a számított adatokat tartalmazza. A gépben tárolt adatok alapján a továbbiakban nem jelent problémát, hogy a felhasználók igényeinek megfelelően gyártási vagy üzemeltetési dokumentációként szolgáló táblázatokat lehessen kiadni, bármilyen kívánt formában. Jelenleg egyféle realizációs táblázatot ad ki a program, amely a huzalokat a csatlakozók és azon belül az érintkezők sorszáma szerint rendezve tartalmazza. Ez a táblázat gyártási dokumentációként szolgál.

A következőkben röviden ismertetjük a TAL-2 program felépítését és a használatával összefüggő tudnivalókat.

### 2.1. Az adatok előkészítése

A programot a tervezés azon szakaszában lehet igénybevenni, amikor már a rendszer kártyákra bontása megtörtént és az összekötendő érintkezők sorszáma is meghatározott.

Az elhelyezési és az összekötési terv elkészítése céljából ekkor a program számára meg kell adni az egymással összekötendő (ekvipotenciális) érintkezők listáját. Az egyes érintkezőkre a következő két adattal kell hivatkozni:

- a kártya azonosítója
- az érintkező sorszáma

Az egyes ekvipotenciális érintkező csoportokhoz lehet, de nem kötelező egy jelazonosítót is hozzárendelni.

A következőkben felsoroljuk a programhoz szükséges, illetve lehetséges bemenő adatokat:

#### a) Általános adatok

- a kártyablokk konstrukciós adatai
- vezérlő adatok: milyen táblázat kell, milyen legyen az optimalizálás foka ( $p$ ), van-e rögzített kártyapozíció?

#### b) Rögzített kártyát definiáló adatok.

Ha van olyan kártya, amelynek a helyét a tervező előre kijelöli, akkor ezek adatait is meg kell adni:

- a kártya azonosítója,
- a kijelölt pozíció.

A kártyablokkból kimenő vezetékek számára szolgáló csatlakozót rögzített kártyaként kell kezelni és itt kell a helyét definiálni.

#### c) Jelet és ekvipotenciális érintkezőket definiáló adatok.

A kártyablokkban szereplő valamennyi jelet definiálni kell és fel kell sorolni a hozzátartozó érintkezőket.

Egy jel adatai a következők:

- a jel neve (nem kötelező adat),
- a jel vagy huzal típusa (nem kötelező),
- a jel súlya (ha nincs megadva, akkor 1),
- az érintkező pontok listája:
  - a kártya azonosítója,
  - az érintkező sorszáma,
  - a jelforrás jele (nem kötelező),
  - az érintkezőn elhelyezhető vezetékek száma (ha nincs megadva, akkor 3).

### 2.2. A program leírása

A TAL-2 program blokkvázlata az 1. ábrán látható. Az ábra alapján a program működése különböző magyarázatot nem igényel, az eljárás a 2. pontban ismertetett módszert követi.

A felhasználás során szerzett tapasztalatok azt mutatják, hogy a beültetésre kiválasztott kártya helyének keresése során nem szükséges valamennyi, már beültetett kártyahelyet is megvizsgálni. Ez egyrészt rendkívül sok időt igényel, másrészt alig javít az eredményeken. A program számára ezért egy  $p$  paraméterrel (lásd 2.1a pont) meg lehet adni, hogy csak a kiválasztott góc üres helyeit vizsgálja végig ( $p=0$ ), vagy vizsgálja ugyan a már beültetett helyeket, de ezek közül csak az üres hellyel szomszédosakat ( $p=1$ ) vagy az üres hellyel szomszédos helyet és annak a szomszédját is ( $p=2$ ). A  $p$  az úgynevezett cserélési mélység, amelyet az általános adatok között kell megadni.

A program nyelve FORTRAN és az SZKI Siemens 4004/45-ös gépére van írva. A futási idő természetesen függ a kártyák, a bekötött érintkezők és jelek számától, továbbá erősen függ a cserélési mélységtől.

Egy jellemző futási adat: 20 kártya elhelyezése, 220 jel bekötése 0 cserélési mélység esetén 7,5 percet vett igénybe.

### 2.3. Eredmények

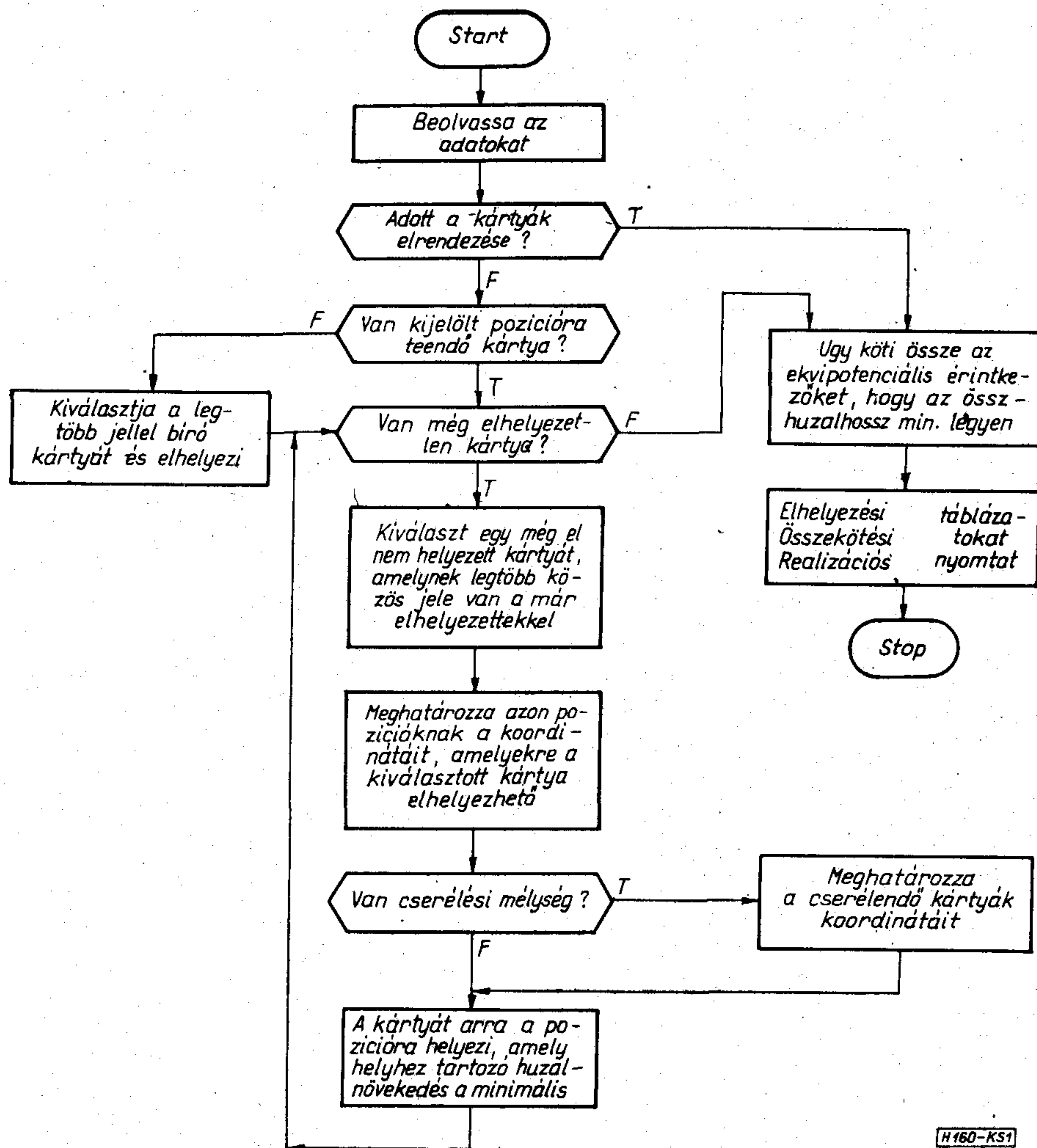
A program az eredményeket sornyomtatón adja ki, táblázatok formájában.

#### a) Elhelyezési táblázat

A táblázat három sorból áll. Az első sorban a kártyapozíciók sorszámai állnak, a következő két sor pedig a kártyablokk első, illetve második sorában a megfelelő kártyapozícióba elhelyezett kártyák azonosítóit tartalmazza.

#### b) Összekötési táblázat

Ez a táblázat a beolvasás sorrendjében felsorolja az egyes jeleket, részben a beadott, részben a számított



1. ábra. A TAL-2 program blokkvázlata

huzalozási adatokkal együtt. Egy jel a következő adatokkal szerepel a táblázatban:

- a jel sorszáma, amelyet a program ad a beolvasás sorrendjében,
- a jel neve (ha van),
- a jel vagy huzal típusa (ha van),
- érintkezőpárok listája. Az egy sorba írt két-két érintkezőt közvetlenül össze kell kötni. Egy érintkezőpár adatai a következők:
  - honnan (kártyapozíció és érintkező sorszám),
  - hova (kártyapozíció és érintkező sorszám),
  - kötési hossz (mm),
  - jelforrás jele (ha van),
  - az érintkezőpár (a huzal) sorszáma,
- a fa hossza.

Az összekötési táblázat végén kiírásra kerül még az összegzett teljes huzalhossz.

### c) Realizációs táblázat

Csak a huzalozás elkészítéséhez szükséges „honnan-hova” adatokat tartalmazza, a jelekre vonatkozó információk nélkül. Az esetleges azonosítás céljából a két adathoz egy harmadik csatlakozik, a huzal sorszáma, amely az összekötési táblázatban is szerepel. Mivel ennek első fele a jel sorszáma, így a huzal-

hoz tartozó jel könnyen kikereshető és így a további adatok felderíthetők.

A huzalok felsorolása a kártyapozíciók, ezen belül pedig az érintkezők sorrendjében történik. Minden huzal csak egyszer szerepel a táblázatban, és pedig annak a végének a bekötési pontjánál, amelyhez kisebb kártyapozíció sorszám tartozik, illetve kártyán belüli összekötéseknél a kisebb sorszámú érintkezőknél.

A huzalok csoportosítva vannak egy-egy kártyasoron belüli, illetve a kártyasorok közti összekötések szempontjából. Egy-egy ilyen csoporton belül újabb csoportosítás lehetséges, mégpedig vezeték típusok szerint, ha ezt a felhasználó kívánja. A felsorolási elv minden csoporton belül érvényesül.

### I R O D A L O M

- [1] Cooper, L.: Heuristic methods for location-allocation problems. SIAM Review, January 1964. pp. 37—53.
- [2] Steinberg, L.: The backboard wiring problem: A placement algorithm. SIAM Review, January 1961. pp. 37—50.
- [3] Rutman, R. A.: An algorithm for placement of interconnected elements based on minimum wire length. Proc. SJCC, April 1964. pp. 477—491.
- [4] Nicholson, T. A. J.: Permutation procedure for minimizing the number of crossings in a network. Proc. IEE. January 1968. pp. 21—26.

- [5] *Kakita, H. H.*: Automatic wiring verifier. Computer Design, June 1968. pp. 50—54.
- [6] *Herbst, R. T.*: Designing equipments with computers. Bell Laboratories Record, April 1966. pp. 129—134.
- [7] *Prim, R. C.*: Shortest connection networks and some generalizations. The Bell System Tech. Journal. November 1957. pp. 1389—1401.
- [8] *Pomentale, T.*: An algorithm for minimizing backboard wiring functions. Comm. of ACM. November 1965. pp. 699—703.
- [9] *Garside, R. G.—Nicholson, T. A. J.*: Permutation procedure for the backboard-wiring problem. Proc. IEE. January 1968. pp. 27—30.
- [10] *Koch*: Computer-assistierter Entwurf von Geräteverdrahtungen. Nachrichtentechnik, H. 9., 1967. pp. 351—357.
- [11] *Merkin, N.*: Unsnarl your complex wiring problems. Electronic Design, November 1968. pp. 72—76.
- [12] *Loberman, H.—Weinberger, A.*: Formal procedures for connecting terminals with a minimum total wire lengths. Journal of ACM. April 1957. pp. 428—433.
- [13] *Gilmore, P. C.*: Optimal and suboptimal algorithms for the quadratic assignment problem. Journal of SIAM, June 1962. pp. 305—313.
- [14] *Houghton, J.*: A system for the placement of circuit modules. IEEE Conference Publications, No 51. 1969. pp. 82—88.
- [15] *Помазанов, В. М.*: К задаче о размещении ячеек в панели. В сборнике статей «Применение вычислительных машин для проектирования цифровых устройств», ред. Матюхин, Н. Я., Изд. «Советское Радио», Москва, 1968, стр. 165—182.
- [16] *Штеин, М. Е.—Гайдаенко, В. С.*: О задачах трассировки монтажных соединений. См. (15).

## Szakmérnöki szakok indítása

**A Budapesti Műszaki Egyetem Villamosmérnöki Kara 1973. februárban az alábbi szakmérnöki szakokat indítja:**

### Rádió műsorszoró és hírközlő szak:

A szak oktatási célkitűzése a rádió- és TV-adókat, rádió hírközlő berendezéseket gyártó ipari, valamint a Posta, MÁV, kisebb mértékben a honvédség, MAHART, valamint az autóközlekedési vállalatok részére speciális tudású szakemberek képzése. Elsősorban a fix és mozgó összeköttetések tervezésére és üzemeltetésére alkalmas szakmérnökök képzéséről van szó. Ezenkívül a műsorszoró és rádió hírközlő ipar területén a berendezések, áramkörök tervezési, fejlesztési feladatainak megoldására is képesít.

### Átviteltechnikai szak:

A szak oktatási célkitűzése a villamosmérnöki, gépészmérnöki vagy természettudományi (matematikus, fizikus) diplomával rendelkezők továbbképzése, elsősorban a híradástechnikai iparok és a posta számára biztosítva speciális képzettségű szakembereket. Az elektronikus műszeriparban dolgozók és a híradástechnikai nagyberendezéseket üzemeltetők számára is ajánlható a szak.

A tantárgyak a beszéd, zene, adat- és képátvitel általános kérdéseit és a vezetékes híradástechnika műszaki berendezéseit tárgyalják. Foglalkoznak a híradástechnikai alkatrészek kiválasztásával, a különböző fontosabb áramkörök átviteltechnikai berendezések és a teljes hírközlő rendszer tervezésével, gyártásával és üzemeltetésével.

### Digitális elektronika szak:

A szak célja villamosmérnöki, ill. fizikusi oklevéllel rendelkezők továbbképzése a félvezető eszközöknek a híradástechnikában, mérés technikában, számítás-

technikában és az adatfeldolgozás területén történő alkalmazásával kapcsolatos speciális új ismeretekben. Ennek érdekében a matematikai és ferrites impulzustechnikai és digitális technikai képzést nyernek a hallgatók. A képzés a félvezetős impulzustechnikai és logikai alapáramkörök mellett a nanoszekundumos impulzustechnikára és impulzustechnikai mérésekre is kitér. A digitális rendszerek alapjainak tárgyalása után a digitális számítógépek, digitális mérés technika, elektronikus telefonközpontok, adatfeldolgozás ismertetésére kerül sor. A szak a szubminiatűr technika ismertetésével és félvezetős elektronikai berendezések tervezésével zárul. A gyakorlati képzést a tárgyakhoz szervesen kapcsolódó laboratóriumi mérések biztosítják.

### Irányítástechnikai szak:

A szak oktatási célja okleveles gépész-, vegyész- és villamosmérnökök továbbképzése a következő három ágazatban:

#### a) Folyamatszabályozási ágazat

Az ágazat oktatási célja, hogy a különböző folytonos technológiával rendelkező folyamatok irányítási feladatainak ellátására alkalmas szakmérnököket képezzen. A szak elsősorban a vegyi és az azzal rokon iparágak komplex automatizálási kérdéseivel foglalkozik.

#### b) Erőművi és hálózati ágazat

Az ágazat oktatási célja, hogy az energiaiparban dolgozó villamos- és gépészmérnökök részére olyan irányítástechnikai ismereteket adjon, hogy ezen mérnökök képesek legyenek irányítástechnikai egységeket üzembe helyezni, üzemeltetni és az alkalmazás szempontjából megtervezni.

#### c) Hajtásszabályozási ágazat

Az ágazat oktatási célja, hogy az erősáramú iparban dolgozó mérnökök részére olyan irányítástechnikai ismereteket adjon, hogy ezen

mérnökök képesek legyenek irányítástechnikai egységeket üzembe helyezni, üzemeltetni és az alkalmazás szempontjából megtervezni.

### Elektronikai technológia szak:

Az oktatás célja az, hogy a professzionális elektronikus berendezések legújabb speciális technológiáival megismertesse az ezen a területen dolgozó mérnököket.

Az oktatás elsősorban az új technológiákhoz szükséges speciális matematikai és anyagszerkezeti ismereteket nyújtja, megvilágítja a technológia és az áramköri, illetve szerkezeti konstrukció közötti kapcsolatokat. Ezután alapot ad a legújabb technológiai folyamatok tervezéséhez, eszköztervezéséhez, illetve kiválasztásához, az előbbiekhöz kapcsolódó kutatómunka megindításához.

A jelentkezés feltétele: híradás- és műszeripari technológiai, híradástechnikai vagy műszerszakos villamosmérnöki oklevél.

### Fényforrások szak:

A szak célja, hogy a fényerjesztés fizikai és a fényforrás technológiai, valamint a vonatkozó vizsgálati, mérési és minőségellenőrzési területen olyan korszerű és elmélyült ismereteket nyújtson, amelyek a jelzett tudomány-, illetve iparág önálló, tudományos igényű alkalmazásához, gyakorlatához, fejlesztéséhez, valamint irányításához szükségesek.

### Hálózat és térelmélet szak:

A szak célja villamosmérnökök továbbképzése a hálózatelméletben, ill. a térelméletben felvetett új problémák területén.

A képzés főbb területei a hálózatelméletben a lineáris, invariáns, koncentrált paraméterű, passzív hálózatok analizálásával szemben az ilyen hálózatok szintézise, a nemlineáris, a változó paraméterű, az elosztott paraméterű és az aktív hálózatok analizálása, ezek szintézise. A bonyolultabb hálózatok, ill. hálózattípusok tárgyalása a gráfelméleti módszerek és a rendszerek állapotváltozós jellemzése útján történik.

A térelméleti vizsgálatok középpontjában a speciális anyagok hatása áll (giromágnes, negatív vezetőképességű stb. közegek). Nagy szerepet kapnak a képzésben a számítógépes eljárások. Fontos elvi kérdésként jelentkezik az elektromágneses tér és a villamos hálózatok kapcsolata.

### Számítógépes szimuláció erősáramú hálózatokban szak:

A szak oktatási célkitűzése villamosmérnökök vagy gépészmérnökök (B tagozatos) továbbképzése az erősáramú hálózatok tervezésével és üzemeltetésével foglalkozó mérnökök megismertetése a korszerű számítógépes eljárások matematikai alapjaival — különös tekintettel a szimulációs módszerekre —, továbbá azok konkrét alkalmazási lehetőségeivel erősáramú hálózatok üzemi és tervezési problémáinak megoldásánál. Szűkebb oktatási célkitűzések: az említett szakterületeken dolgozó mérnökök továbbképzése annak érdekében, hogy munkájukat magasabb szinten, a kibernetika és a korszerű matematikai eljárások teljes mértékben történő felhasználásával legyenek képesek ellátni.

Jelentkezni a Budapesti Műszaki Egyetem Villamosmérnöki Karának Dékáni Hivatalában lehet. Jelentkezési határidő 1972. szeptember 15.

## SZEMLE

(Folytatás a 214. oldalról)

A felmérések szerint a fekete-fehér televízió vevőkészülékek értékesítése ma már nemcsak az Amerikai Egyesült Államokban, hanem egyes nyugat-európai országokban is rohamosan csökken. Néhány éven belül e piacokon már csak pótlásra és másodkészülékként fognak ilyen vevőkészülékeket vásárolni. A színes televízió vevőkészülékek megjelenése azokban az országokban is lelassította a fekete-fehér készülékek értékesítését, ahol a fekete-fehér vevőkészülékekkel való ellátottság viszonylag kicsi, de néhány éven belül bevezetik a színes adást. A színes vevőkészülékek hatása világszerte megnyilvánul a kis képcsőméretű, fekete-fehér vevőkészülékek értékesítésének növekedésében. Ez a legjobban az eddig nagy képcsőmérethez ragaszkodó Nyugat-Európában mérhető le.

Néhány országban (USA, Dánia, Anglia, NSZK, Svédország stb.) már több színes vevőkészüléket gyártanak, mint fekete-fehéret. Franciaországban pl. 1975-re minden második gyártott készülék színes lesz, sőt az NSZK-ban már 1971-ben a készülékgyártás 75%-át a színes televízióvevők előállítására jelentette. A piac telítettségéről ennek ellenére nem beszélhetünk.

A színes vevőkészülékek értékesítésére kitűnő piacot jelent India, ahol 1970-ben mindössze 8717 db színes vevőkészülék

üzemel. Indiában meg nem erősített közlések szerint — nemrégiben hagyták jóvá a televízióhálózat bővítésének tervét. Ennek kapcsán olyan javaslatot dolgoztak ki, amely állami támogatással kívánja biztosítani a televízió — elsősorban a színes vevők — gyors elterjedését. (KGM — MTTI információ, 1972 jan.)

\* \* \*

1971 első negyedében az amerikai elektroncső-ipar 1,64 millió elektroncsövet állított elő, 1%-kal kevesebbet, mint 1970 első negyedében. Az időközben végrehajtott árleszálítások következtében a forgalom ennek ellenére 15,3%-kal elmarad — az értékesített elektroncsövek összárát tekintve — az 1970-estől. Az elektroncső forgalom ilyen kis mértékű csökkenésére nem számítottak. Azt várták, hogy a második értékesítési hullám-maximum elmúltával erőteljesebb forgalomcsökkenés fog jelentkezni. Pillanatnyilag újraelemzik az elektroncső-forgalmat és a várható felhasználást. Amennyiben stabilizáció jelentkezik — 3%-os ingadozást megengedve — át kell értékelni azokat a felméréseket és prognosztikákat, amelyek ellenkező értékesítési helyzet feltételezésén alapulnak. (Radio-Mentor Elektronik, 1971. 9. sz.)

## Tartalmi összefoglalások

ETO 681.3.001.2:681.3.06

Drasny J.:

## Számítógépek automatizált tervezése

HÍRADÁSTECHNIKA XXIII. (1972) 7. sz.

Az SZKI foglalkozik a számítógépek tervezésének automatizálásával és elkészít egy olyan programrendszert, amely a logikai tervet szimulálja és az így ellenőrzött logikai terv adatai alapján konstrukciós, technológiai és üzemeltetési dokumentációkat állít elő. A cikk ismerteti a programrendszer felépítését és az egyes részprogramok általános tulajdonságait.

ETO 164.1:681.325.65.001.4

Bohus M.—Dr. Géher K.:

## Logikai hálózatok számítógépes vizsgálata

HÍRADÁSTECHNIKA XXIII. (1972) 7. sz.

A logikai hálózatok számítógépes vizsgálatán két feladatcsoportot értünk; (i) a logikai hálózat analízisét és (ii) a diagnosztikai tesztek generálását. A dolgozat ezekről nyújt áttekintést. A tervező szempontjából az alapadatok megadásának legcélszerűbb módja az integrált áramköri készlet katalógusához csatlakozó leírás. A logikai analízis program kiindulásul szolgál a diagnosztikához is. Ez a hibatáblázat módszerrel, érzékenységi út módszerrel, hibaszótár módszerrel végezhető el. A Boole-differencia előnyösen használható egyszerű és kettős hibák kimutatására.

ETO 621.3.049.7—111.681:3.06 LOGAN

Bohus M.—Csopaki Gy.—Gefferth L.—Halász E.—  
Dr. Németh G.—Trón T.—Varró L.:

## A LOGAN logikai szimulációs program

HÍRADÁSTECHNIKA XXIII. (1972) 7. sz.

A LOGAN programrendszer segítségével integrált áramköri tokokból felépített szinkron szekvenciális hálózatok áramköri, vezérlési és logikai működése vizsgálható. A programrendszer FORTRAN—IV nyelven készült a SIEMENS 4004/45 számítógépre a TEXAS TTL integrált áramköri készlet alapul vételével. A dolgozat utal a programrendszer sokrétű alkalmazási lehetőségeire.

ETO 621.3.049.75.001.2:681.3.06 TAL-1

Drasny J.—Kovács M.—Mezey Gy.:

## Nyomatási rajzok készítése a TAL-1 programmal

HÍRADÁSTECHNIKA XXIII. (1972) 7. sz.

A cikk ismerteti a nyomtatott kártyák számítógépes tervezésénél figyelembe veendő szempontokat, különös tekintettel az SZKI-ban kidolgozott kártyatervező program tulajdonságaira. A kártyatervezés utáni fotosablon-készítés a TAL-1 program segítségével történik, amely 1970 óta használatban van. A közlemény beszámol a felhasználási lehetőségekről és módokról, valamint a felhasználás során szerzett tapasztalatokról.

ETO 621.3.049.75.001.2:681.3.06 TAL-2

Koszó G.—Sziray J.:

## Hátoldali huzalozás tervezése. A TAL-2 program

HÍRADÁSTECHNIKA XXIII. (1972) 7. sz.

A nyomtatott kártyák keretben történő elhelyezése nem minden szempontból megoldott probléma. A cikk egy új heurisztikus jellegű eljárást ismerteti a minimális közeli összhuzalhossz eredményező megoldás elérésére. Az eljárás figyelembe veszi a csatlakozók geometriai méreteit és a minimális fák az elrendezés során fokozatosan alakítja ki. A TAL-2 program a leírt módszert alkalmazza és huzalozási dokumentációkat állít elő.

## Обобщения

ДК 681.3.001.2:681.3.06

Драшни, Й.:

## Автоматическое проектирование ЭВМ

HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXIII.  
(1972) № 7

Координационный Институт Вычислительной Техники (SZKI) занимается автоматизацией проектирования ЭВМ и изготавливает систему программ, симулирующую логический план и на основе данных, таким образом контролируемых, производит документацию по конструкции, технологии и эксплуатации. Статья излагает построение системы программ и общие свойства отдельных частичных программ.

ДК 164.1:681.325.65.001.4

Бохуш, М.—Д-р Гехер, К.:

## Испытание ЭВМ логических схем

HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXIII.  
(1972) № 7

Испытание ЭВМ логических схем включает в себе две группы задач: (i) анализ логической схемы, (ii) генерацию диагностических испытаний. Статья дает обзор об этих. С точки зрения проектировщика наиболее целесообразным методом определения основных параметров является применение описания в каталоге комплекта интегральных схем. Программа логического анализа служит тоже как исходный пункт в диагностике. Она может быть сделана методом таблицы ошибок, пути чувствительности или словаря ошибок. Разница Була применяется к показанию однократных и двукратных ошибок.

ДК 621.3.049.7-111:681.3.06 LOGAN

Бохуш, М.—Чопаки, Г.—Гефферт, Л.—Халас, Е.—  
Д-р Немец, Г.—Трон, Т.—Варро, Л.:

## Программа логическая симуляции LOGAN

HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXIII.  
(1972) № 7

С помощью системы программ LOGAN испытывается работа схемы, ее управления и логическая функция синхронных последовательных сетей, построенных из интегральных схем с корпусами. Система программ разработана на программном языке FORTRAN-IV для ЭВМ SIEMENS 4004/45 на основе комплекта интегральных схем TEXAS TTL. Статья излагает многосторонние возможности применения системы программ.

ДК 621.3.049.75.001.2:681.3.06 TAL-1

Драшни, Й.—Ковач, М.—Мезеи, Г.:

## Изготовление чертежей для печатания программы ТАЛ-1

HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXIII.  
(1972) № 7

Статья излагает точки зрения, которые должны быть учтены при проектировании карточек печатных схем, особенно имея в виду свойства программы, изготовленной в SZKI для проектирования печатных схем. После проектирования карточки изготовление фотошаблона производится с помощью программы TAL-1, примененной с 1970 г. Статья дает отчет о возможностях и способах применения, а также об опытах, полученных в течение применения.

ДК 621.3.049.75.001.2:681.3.06 TAL-2

Косо, Г.—Сираи, Й.:

Проектирование расположения проводов задней стороны.  
Программа ТАЛ-2HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXIII.  
(1972) № 7

Расположение печатных схем в рамке является проблемой нерешенной со всех точек зрения. Статья излагает новый метод эвристического характера с целью достижения решения, имеющего последствие приблизительно минимальную общую длину проводов. Метод принимает во внимание геометрические размеры соединителей и образует минимальное число деревьев постепенно в течение расположения. Программа ТАЛ-2 применяет вышеуказанный метод и изготавливает документацию по расположению проводов.



DK 681.3.001.2:681.3.06

Drasny, J.:

**Automatisierter Entwurf von Rechenmaschinen**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

Die SZKI (Institut für die Koordinierung der Computer-Technik) beschäftigt sich mit der Automatisierung des Entwurfes von Rechenmaschinen und macht solch ein Programmsystem, welches den logischen Entwurf simuliert und auf Grund der auf solcher Weise kontrollierten Angaben des Logik-Planes Instruktions-, technologische und funktionelle Dokumentationen herstellt. In dem Artikel werden die Struktur des Programmsystems und die allgemeinen Eigenschaften der einzelnen Teilprogramme erörtert.

DK 164.1:681.325.65.001.4

Bohus, M.—Dr. Géher, K.:

**Untersuchung von logischen Netzwerken mit Rechenmaschine**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

Die Untersuchung von logischen Netzwerken mit Rechenmaschine umfasst zwei Aufgabengruppen, bzw.; (i) Analyse des logischen Netzwerkes und Erzeugung der diagnostischen Prüfungen (Tests). Dieser Artikel gibt einen Überblick über die oben erwähnten Probleme. Die zweckentsprechendeste Methode der Bestimmung der Grundangaben ist vom Gesichtspunkt des Entwurfes die Beschreibung im Katalog der Auswahl von integrierten Schaltungen. Das logische Analysenprogramm dient auch als Ausgangspunkt der Diagnostik. Das kann mit Fehlertabellenmethode, Empfindlichkeitswegmethode und Fehlerwörterbuchmethode ausgeführt werden. Die Boole'sche Differenzmethode kann vorzüglich zum einfachen und zweifachen Fehleranzeigen angewendet werden.

DK 621.3.049.7—111:681.3.06 LOGAN

Bohus, M.—Csopaki, Gy.—Gefferth, L.—Halász, E.—  
—Dr. Németh, G.—Trón, T.—Varró, L.:**LOGAN Logik-Simulationsprogramm**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

Mit Hilfe des Programmsystems LOGAN kann die Stromkreis-, Steuerungs-, und Logik-Funktion von Synchron-Sequenz-Netzen, aufgebaut, aus integrierten Stromkreisen untersucht werden. Das Programmsystem wurde auf FORTRAN—IV Sprache ausgearbeitet zur Rechenmaschine SIEMENS 4004/45 auf dem Grund des TEXAS TTL integrierten Stromkreissatzes. In dem Artikel wird auf die vielfältige Anwendungsmöglichkeiten des Systems hingewiesen.

DK 621.3.049.75.001.2:681.3.06 TAL—1

Drasny, J.—Kovács, M.—Mezey, Gy.:

**Erzeugung von Vordruckzeichnungen mit Programm TAL—1**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

In dem Artikel werden die beim Entwurf von gedruckten Karten mit Rechenmaschine die in Acht zu nehmenden Gesichtspunkte mit besonderer Rücksicht auf die Eigenschaften des im SZKI (Institut für die Koordinierung der Computer-Technik) ausgearbeiteten Kartenentwurfprogramms erörtert. Die Herstellung der Photoschablonen nach dem Kartenentwurf erfolgt mit Hilfe des Programms TAL—1 und ist seit 1970 in Anwendung. In dem Artikel werden die Möglichkeiten und Methoden der Anwendung und die im Gebrauch errungenen Erfahrungen beschrieben.

DK 621.3.049.75.001.2:681.3.06 TAL—2

Koszó, G.—Sziray, J.:

**Entwurf rückseitiger Verdrahtung. Programm TAL—2**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

Das Einsetzen in Rahmen von gedruckten Karten ist noch nicht von allen Gesichtspunkten gelöstes Problem. In dem Artikel wird ein neues Verfahren von heuristischem Charakter erörtert zur Erreichung einer Lösung, die eine annähernd minimale Gesamtdrahtlänge ergibt. In dem Verfahren werden die geometrischen Abmessungen der Anschlüsse in Acht genommen und die minimale Bäume werden während der Anordnung stufenweise entwickelt. Das Programm TAL—2 wendet die beschriebene Methode an und stellt Schaltdrahtverbindungs-dokumentationen her.

UDC 681.3.001.2:681.3.06

Drasny, J.:

**Automated Desing of Computers**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

The SZKI (Intstitute for Coordination of Computing Techniques) deals with the automation of the design of computers and prepares a program system, simulating the logic design and makes constructional, technological and operational documentations on the basis of the data of the logic plan checked in this way: The composition of the program system and the general properties of certain partial programs are presented.

UDC 164.1:681.325.65.001.4

Bohus, M.—Dr. Géher, K.:

**Examination of Logic Networks by Computer**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

Two groups of tasks are understood on the examination of logic networks by computer; (i) analysis of the logic networks and (ii) generation of diagnostic tests. The paper gives a review over the above mentioned tasks. From the point of view of the designer the most practical method of determining the basic data is the description enclosed in the catalogue of the assortment of integrated circuits. The analysis program of logic network is the starting point of the diagnostic test. This can be accomplished by the error table method, the sensitivity path method and error dictionary method. The Boole difference can be advantageously used to display the single and double errors.

UDC 621.3.049.7—111:681.3.06 LOGAN

Bohus, M.—Csopaki, Gy.—Gefferth, L.—Halász, E.—  
—Dr. Németh, G.—Trón, T.—Varró, L.:**LOGAN Logic Simulation Program**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

By means of the LOGAN program system the circuit, control and logic operation of synchronous sequential networks built up of integrated circuits can be examined. The program system was made in language FORTRAN—IV for the computer SIEMENS 4004/45 on the basis of the assortment of TEXAS TTL integrated circuits. The paper presents the manifold fields of application of the program system.

UDC 621.3.049.75.001.2:681.3.06 TAL—1

Drasny, J.—Kovács, M.—Mezey, Gy.:

**Preparation of Drawings for Printing with Program TAL—1**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

The paper presents the aspects to be taken into account at the design of printed cards by a computer, with special regard to the properties of card design program elaborated by the SZKI (Institute for Coordination of Computing Techniques). The preparation of photo chablons following card design is made by help of the program TAL—1, which has been in use since 1970. The paper gives an account of the possibilities and methods of utilization and of the experiences obtained in the course of application.

UDC 621.3.049.75.001.2:681.3.06 TAL—2

Koszó, G.—Sziray, J.:

**Desing of Rear-Side Wiring. Program TAL—2**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

To insert the printed cards into frame is a problem not yet solved from all aspects. The paper presents a new producedure of heuristic character to obtain approximately minimum total wire length. The procedure takes into account the geometric dimensions of the connectors and develops the minimal trees step by step during the arrangement. The Program TAL—2 applies the described method and prepares wiring documentations.

## Résumés

CDU 681.3.001.2:681.3.06

Drasny, J.:

**Projet automatique des ordinateurs**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

SZKI s'occupe de l'automatisation du projet des ordinateurs et prépare un système des programmes, simulant le plan logique et produit des documentations de construction, technologie et exploitation, basées sur les données du plan logique, tellement contrôlé. L'article expose la composition du système des programmes et les propriétés générales des programmes partiels.

CDU 164.1:681.325.65.001.4

Bohus, M.—Dr. Géher, K.:

**Examen des réseaux logiques par un ordinateur**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

Examen des réseaux logiques par un ordinateur comprend deux groupes des tâches; (i) l'analyse du réseau logique, (ii) la génération des essais (tests) diagnostiques. L'article donne une revue de celles-ci. En ce qui concerne le dessinateur-projeteur, la méthode la plus convenable est l'utilisation des données comprises dans la description du catalogue de l'assortiment des circuits intégrés. Le programme de l'analyse logique fournit les données de départ au diagnostic, qui peut être fait par la méthode du tableau des erreurs, de la voie de sensibilité, ou bien du dictionnaire des erreurs. La différence Boole peut être utilisée avantageusement pour identifier les erreurs simples et doubles.

CDU 621.3.049.7—111:681.3.06 LOGAN

Bohus, M.—Csopaki, Gy.—Gefferth, L.—Halász, E.—  
— Dr. Németh, G.—Trón, T.—Varró, L.:**Programme de simulation logique LOGAN**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

A l'aide du système des programmes LOGAN on peut examiner la fonction logique et de commande de circuits des réseaux synchronies

séquentiels composés de blocs de circuits intégrés scellés. Le système de programmes est fait en utilisant la langue FORTAN—IV pour l'ordinateur SIEMENS 4004/45, basé sur l'assortiment des circuits intégrés TEXAS TTL. L'article donne une revue des possibilités diverses d'utilisation du système de programmes

CDU 621.3.049.75.001.2:681.3.06 TAL—1

Drasny, J.—Kovács, M.—Mezey, G.:

**Préparation des dessins d'imprimerie a l'aide du programme TAL—1**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

L'article expose les points de vue en projetant avec un ordinateur des cartes imprimées, considérant particulièrement les propriétés du programme pour projeter des cartes, élaboré par SZKI (Institut de coordination pour la technique de calcul). La réalisation du chablon de photographie après le projet des cartes est faite à l'aide du programme TAL—1, utilisé depuis 1970. La publication rend compte des possibilités et méthodes de l'utilisation ainsi que des expériences obtenues au cours de l'utilisation.

CDU 621.3.049.75.001.2:681.3.06 TAL—2

Koszó, G.—Sziray, J.:

**Projet du cablage du côté d'arrière. Le programme TAL—2**

HÍRADÁSTECHNIKA (Budapest) XXIII. (1972) N° 7

La mise en place des cartes imprimées dans un cadre n'est pas un problème résolu de tous les points de vue. L'article expose une méthode du caractère heuristique ayant pour résultat une longueur totale des fils presque minimale. La méthode prend en considération les dimensions géométriques des dispositifs de connexion et les arbres minimaux sont formés progressivement au cours de l'arrangement. Le programme TAL—2 applique la méthode décrite et prépare des documentations de cablage.

