

A szerkesztő bizottság elnöke: HORVÁTH IMRE

Főszerkesztő: ANGYAL LÁSZLÓ

### SZERKESZTŐ BIZOTTSÁG

#### BHG

Laczkó Endre  
Bernhardt Richárd  
Dr. Eisler Péter  
Dr. Gosztony Géza  
Honti Ottó  
Klug Miklós  
Tölgyesi László

#### ORION

Jakubik Béla  
Csernoch János  
Froemel Károly  
Sass Károly  
Szabó Károly  
Szász Gerő

#### TERTA

Bánsághi Pál  
Baján Tibor  
Benedek Elek  
Halmi Gábor  
Hutter Mihály

BHG ORION TERTA

# MŰSZAKI KÖZLEMÉNYEK

XXVIII. évfolyam

1982

5. szám

## TPV-telefonközpontok hívásfeldolgozó rendszerének funkcionális specifikációja

MAKAY ATTILA  
BHG

### 1. Bevezetés

Jelen dolgozat címében a „specifikáció” helyett alkalmazhattuk volna a „leírás” vagy az „identifikálás” kifejezést is, mint ahogy ezt pl. Horváth Gyula tette hasonló témájú tanulmányában [2]. Hogy mégis a fenti címet választottuk, annak az az oka, hogy a témát kifejezetten a tervezés szemszögéből kívánjuk megközelíteni, és a tervezéshez szükséges input adatok, vagyis a specifikációk megadásának, a vezérlési feladatok pontos leírásának szükségessége a TPV-központok esetén sokkal hangsúlyozottabban kerül előtérbe, mint eddig bármikor. A számítógépekre egyre bonyolultabb feladatokat bízunk és az ezekhez szükséges programrendszerek tervezése és működésük ellenőrzése, kézben tartása nem lehetséges pontos, jól áttekinthető specifikációk nélkül. Ezért világszerte jelentős erők foglalkoznak általában is programrendszerek specifikálására alkalmas módszerek, nyelvek kidolgozásával.

Az előzmények, a problémakör felvetése és a telefonközpontok speciális viszonyaira való megfogalmazása a már idézett [2] alatti irodalomban, valamint a szerző egyéb munkáiban [3] megtalálható. Az ezekben lefektetett meghatározások, alapelvek képezik jelen dolgozat alapját is. A bemutatandó módszer azonban alkalmazkodik az azóta megszületett CCITT ajánlásokhoz [1], valamint tükrözi a TPV-központok software tervezésében a BHG-ban eddig szerzett tapasztalatokat. Egyformán alkalmazható a rendszertervezés kezdeti, specifikációs szakaszában, valamint a már realizált rendszer működési leírásában. Rögtön hangsúlyozni kívánjuk, hogy működési specifikációról lesz szó, vagyis a megvalósítandó telefonos szolgáltatásoknak már egy bizonyos mértékig adott rendszer keretein belüli specifikálását tűztük ki célul. Azaz, a specifikációs határterületet nem a telefonközpont vonalai és a központ között vontuk meg, hanem a központon belül. A TPV-rendszer specifikációjának első lépése éppen annak meghatározása, hogy a belső határfelületeket hol és hogyan vonjuk meg. Mivel a TPV-közpon-

tok hardware architektúrája a telefonos szolgáltatásokból csak a legalapvetőbbeket tükrözi (mikrofon-áram adása, csengetés, hangok, információgyűjtés stb.), a hívások lebonyolításával járó minden funkció egyértelműen a vezérlő számítógép(ek) memóriájában realizálódik, amelyen belül már nehezebb szemléletes határfelületeket vonni. A vonalak és a központ közötti határfelület specifikációjával, vagyis a telefonos szolgáltatások berendezésfüggetlen specifikációjával itt nem kívánunk bővebben foglalkozni. Ezeket a berendezésfüggetlen specifikációkat röviden külső specifikációknak nevezhetjük, amelyek természetesen minden újonnan kifejlesztendő rendszer kiinduló pontját képezik, úgy is mondhatnánk, hogy a rendszerspecifikáció legfelső szintjét képviselik. A hívásfeldolgozást tekintve ilyenek például a vonalakról igénybevehető szolgáltatások és ezek lebonyolítási módjainak „algoritmusai” (a készülékek és egyéb perifériális berendezések használati utasításai), a megvalósítandó számozási, irányítási, csillapítás-vezérlési stb. rendszerek specifikációi. Ezek konkrét kiviteli formája az eddigi gyakorlatban többnyire a verbális, szöveges leírás. Az említett CCITT ajánlások elterjedése még csak ezután várható postai körökben. Ezek alkalmazásba vétele pedig igen fontos lenne, mivel az említett verbális módszerek egyszerűen nem teszik lehetővé pl. az előfizetői szolgáltatások precíz, minden normális és abnormális viselkedésre kiterjedő megadását. Ez pedig a fejlesztésnek kifejezetten akadálya, mivel a hiányos specifikáció „pótlása” a tervezés alacsonyabb szintjén sokkal nehezebb és az esetleges hibás döntés hatása nehezebben ellenőrizhető.

### 2. Alapelvek

A következőkben röviden összefoglaljuk a TPV-központok hívásfeldolgozó programrendszerének tervezése során a BHG-ban eddig alkalmazott alapelveket.

A feladatok megosztására a *hívásosztás* elvét választottuk. Ez, mint ismeretes, [5] azt jelenti, hogy a feldolgozás során minden egyes hívás külön önálló jobot vagy összetartozó jobokat képvisel, és a szükséges adatok tárolására a többi hívásától elkülönített tárolók, az ún. *hívástárak* szolgálnak. A hívástár nem feltétlenül egyetlen összefüggő memóriaterületen helyezkedik el, hanem pl. multiprocesszoros vezérlés esetén általában különböző processzorok memóriájában, több részből tevődik össze. Az egyes részek méretére és felépítésére célszerű tipizált egységeket választani. Ezeket a tipizált, a hívástárak felépítésére szolgáló elemi egységeket a következőkben röviden *erőforrásoknak* nevezzük. Az előzőekből következik, hogy egy hívástár mérete nem szükségképpen mindig ugyanaz; a hívásfajtától függően pl. mindig a szükségletnek megfelelő számú erőforrásból fűzhetjük őket össze.

A másik általánosan ismert, általunk is alkalmazott elv az *eseményvezérlés*. Ez röviden a következőket jelenti. A hívásokhoz rendelt jobok a hívás idejének döntő többségében inaktívak. Az operációs rendszer akkor aktivizálja őket, ha a hívásban résztvevő valamelyik külső végződésről (vonalról) valamilyen esemény vagy tevékenység folytán jelzés, üzenet érkezik. Az esemény hatására a job állapota általában megváltozik, az állapotmenet során meghatározott műveletek kerülnek elvégzésre, majd ezek után a job az új állapotában újra inaktív lesz, és itt várja meg a következő eseményt. Ez így megy a hívás, vagyis a job keletkezésétől megszűnéséig.

Ezek után önként adódik, hogy a legfelső, berendezésfüggetlen specifikációs szintet követő szinten a hívásfeldolgozás további specifikációjának magára a hívásra, annak belső eseményeire kell irányulnia. Az előzőek értelmében a hívások különböző belső állapotait és állapotátmeneteit kell úgy megállapítani, hogy azokkal az adott konkrét szolgáltatások megvalósulása leírható legyen. Más szóval a hívásokat állapotátmeneteik segítségével kívánjuk specifikálni. Ezt a specifikációs szintet más néven *modellezési fázisnak* nevezzük, melyben absztrakció útján létrehozott logikai modellek segítségével írjuk le a ténylegesen megvalósuló folyamatok belső, strukturális részleteit. Mint ismeretes [2], [6], az információfeldolgozó rendszerek esetében — amilyenek egy TPV-telefonközpont vezérlője is tekinthető — e modellek céljára legalkalmasabbak a véges automaták. A specifikáció ezek után a következő lépésekből áll:

- a modellalkotás, vagyis az alkalmas automaták kiválasztása
- a választott automaták állapotainak és állapotátmeneteinek meghatározása. Ezek segítségével határozzuk meg, vagyis specifikáljuk a real-time operációs rendszer keretein belül végrehajtandó jobokat. Az ezen specifikációk alapján írt különféle programok képviselik a hívásfeldolgozó rendszernek a fizikailag is létező, a végrehajtó számítógépek memóriájában, valamint a programlistákon „megfogható” részét, a hivatkozott automaták csupán a modellalkotás eszközei, a valóságban nem léteznek.

### 3. A hívásfeldolgozás automatái

Az eddigieknek megfelelően tehát a telefonközpontban lezajló hívások logikai modelljeül a véges automatákat (Mealy automata) választottuk. Az egyszerű hívásosztás elvéből az „egy hívás — egy automata” választás adódna. Azonban ez, mint a tapasztalat mutatta [4] igen nagyszámú állapot felvételéhez vezet, ami a gyakorlatban az áttekinthetőséget erősen rontja, ezért használata igen nehézkes. Az egyes hívások leírására ezért általában nem egy, hanem funkcionális osztás révén több, egymással együttműködő automatát használunk. Ilyen módon a hívásosztás elvét megtartva, a funkcionális osztásból származó előnyöket is kihasználjuk. Ennek eredményeképpen egy-egy automata típusra valóban „véges” számú, jól deklaráható állapot adódik, melyekkel a hívások különböző szituációi szemléletesen, jól követhetően írhatók le.

Az alábbiakban a BHG legújabb nagy kapacitású, elosztott vezérlésű telefon alközpontja számára kifejlesztett hívásfeldolgozó rendszer példáján mutatjuk be a módszer gyakorlati alkalmazását.

Az automaták megválasztásához az előző, közepes kapacitású központ fejlesztési tapasztalatai nyújtottak kiindulási alapot. Ezen tapasztalatok lényege, hogy a [4]-ben tárgyalt hívásautomata számára végül is még mindig igen sok állapotot kellett felvenni. Ez egyrészt abból adódott, hogy a „vonali” automata funkcióinak körét igen szűkre választottuk, vagyis az elvégzendő feladatokból több hárult a hívásautomatára, valamint a hívásautomatát két végpont — a hívó és a hívott — együttes viselkedésének leírására alkalmaztuk, ami ugyancsak a lehetséges állapotok számát növelte. Ennek alapján a tervezendő rendszer számára a következő automaták felvétele látszott célszerűnek — amit a konkrét multiprocesszoros, elosztott vezérlésű hardware architektúra is indokolt.

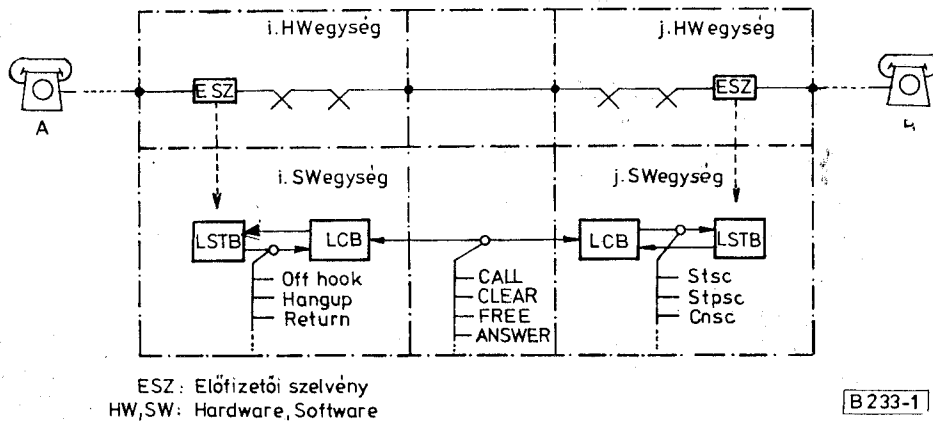
#### 3.1. Jellevő automaták (LSTB, TRSB, ASTB stb.)

Funkcionálisan a [4]-ben tárgyalt vonali automatáknak felelnek meg. A különböző végzések felől (vonalak, trunkök, kezelő asztal stb.) érkező külső, fizikai jelzéseket veszik és továbbítják egységesen megválasztott és értelmezett belső jelekként a többi automata felé.

Így pl. a vonalak felől érkező jelekből (hurok zár és nyit) az LSTB automata előállítja a „hívás” (Call), a „számjegy” (Digit), „bontás” (Hangup), „hívott jelentkezik” (Offhook), „földelőgomb nyomás” (Return) jelzéseket (lásd 1. 2. ábra.)

#### 3.2 Vonali automata (LCB)

Egyetlen végpont közvetlen vezérlését végzi a hívás ideje alatt. Hatáskörébe tartozik a hangok, csengetések kiadása, a kapcsolóút felépítése, bontása. Működése által valósul meg a jellevő automata vezérlése, valamint a hívásban résztvevő másik végpontot vezérlő automatával való együttműködés. (Egy normál beszédkapcsolatban két LCB automata vesz részt, lásd. 1. ábra)



1. ábra. Normál beszédkapcsolatokban résztvevő automaták; tipikus jelek és jelutak

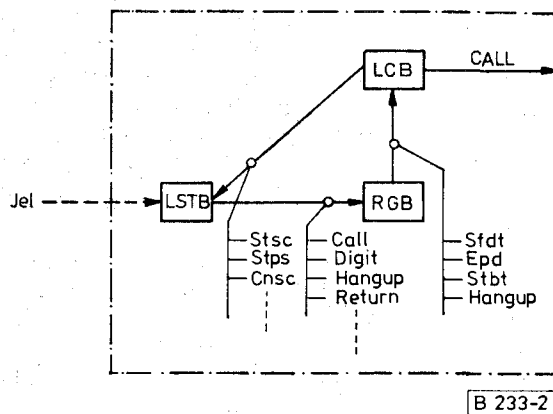
### 3.3 Regiszter automata (RGB)

A hívásfelépítés alatt a hívószámokkal kapcsolatos logikai és vezérlési feladatokat látja el. Funkciója a hagyományos központok regisztereivel analóg. A hívásfelépítés során normál esetben egy LCB automatával működik együtt (lásd 2. ábra).

### 3.4 Szervező automata (COMB)

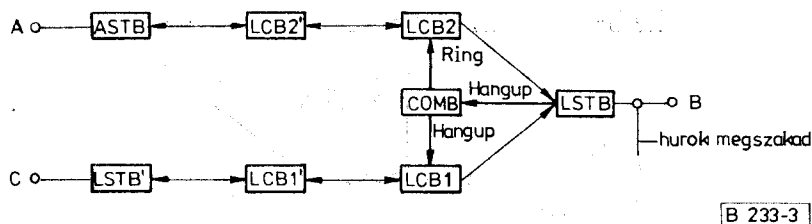
Az alapvető, hívó—hívott kapcsolatot vezérlő LCB—LCB, vagy a kezdeti hívásfelépítést vezérlő LCB—RGB automata konfigurációkon kívül bonyolultabb forgalmi helyzetek modellezéséhez nyilván bonyolultabb konfigurációkra is szükség van. Ezen bonyolultabb forgalmi helyzeteket formailag általában az jellemzi, hogy egyrészt kettőnél több végpont tartozik valamilyen módon egy híváshoz, vagy két hívás között kell valamilyen kapcsolatot teremteni. Ezekben az esetekben a szóbanforgó forgalmi helyzet vezérlésében tehát több LCB vagy RGB automata is részt vesz, melyek együttműködésének irányítására, a szükséges információk továbbítására hivatott a COMB automata (innen a neve, *communication block*).

Szervező funkcióját a COMB olyan módon kell ellátni, hogy fel tudja használni az egyszerű alapeseteket realizáló LCB és RGB automaták „tudományát”, és a bonyolult forgalmi helyzetek lebonyolításához szükséges algoritmusokból az egyszerű esetekhez képesti „többlet” lehetőleg a saját működése révén realizálódjon. Ez azt jelenti, hogy egy bonyo-



2. ábra. Hívásfelépítés konfigurációja a hívó oldalon

lultabb konfigurációban részt vevő LCB vagy RGB nem is tudja, hogy milyen hívásfajtában működik közre, a hozzákapcsolódó COMB automata révén ő mindig csak az egyszerű eseteknek megfelelő együttműködő partnerét látja. Így pl. a 3. ábrán egy befigyelő hívásban résztvevő automatákat láthatunk (A—B: kezelő-mellékállomás, B—C: mellék-mellék-kapcsolat), ahol a keresett fél, a B pont által küldött jeleket a COMB automata közvetíti az LSTB helyett a két kapcsolat megfelelő LCB automatáinak, persze a kívánt működésnek megfelelően módosítva azokat.



3. ábra. Kezelő (A) befigyel foglalt mellékállomás (B) beszélgetésébe, aki elfogadja a felajánlott hívást és letesz

#### 4. Az automaták együttműködési mechanizmusa

A hívások lebonyolításának menetét az előzőekben definiált automaták együttműködésével tudjuk leírni. Az automaták állapotátmeneteik során — melyek természetesen mindig valamilyen input jel hatására következnek be — output jeleket is generálhatnak, melyek egy másik automata számára input jelként jelentkeznek. A központon belül egyidejűleg sok hívás élhet, azaz az LCB, RGB ... stb. automatáknak egyidejűleg sok implementációja létezhet. Amikor tehát egy adott hívás viselkedését írjuk le az automaták közötti jelzést váltások segítségével, hallgatólagosan feltételezzük, hogy az adott híváshoz tartozó automaták ismerik egymást, vagyis az output jelzések „célba” találnak. Az automaták híváspecifikus implementációja a 2. alatt említett erőforrások segítségével valósul meg. Minden egyes automatahoz, deklarációja pillanatában konkrét memória allokáció révén egy-egy erőforrást rendelünk, melyek az automata állapotátmeneteit végrehajtó reentrant programok híváspecifikus adatbázisát képviselik. Ezen erőforrások területén elhelyezett mutatók (pointerek) segítségével láncoljuk össze az egy hívásban résztvevő automatákat (illetve ezek adatbázisait). Az állapotátmenetek során kiadott jelzések tehát ezen pointerek segítségével jutnak el együttműködő partnereikhez.

Egy adott hívásban résztvevő minden automata azonban közvetlenül nem ismerheti összes társát. Ehhez egyrészt nagyobb memóriaigény (sok pointer!), valamint bonyolult mechanizmus tartozna. Célszerű volt az automatákat kapcsolataik számát és típusát illetően is tipizálni. A már említett multiprocesszoros konfigurációt alapul véve a mutatók maguk is kétfélék lehetnek, úgymint belső (processzoron belüli erőforrást jelölő) és külső (más processzor memóriaterületén levő erőforrást jelölő) mutatók. Ezek után a 3. alatti automatákat kapcsolataik szerint az alábbi módon jellemezhetjük:

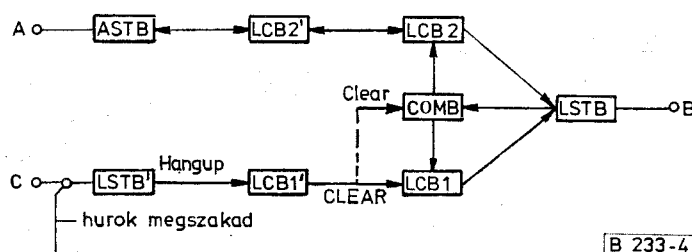
- A jellevő automatának egy belső kapcsolata lehet, vagyis egyetlen belső mutatója van, amivel a hívás során LCB, RGB vagy COMB automata erőforrására mutathat.
- A vonali automata egy belső és egy külső kapcsolatot létesíthet; a belső mutatója mindig egy jellevő automatára (a hozzá tartozó végpontra) mutat, a külső pedig egy másik vonali automatára.

- A regiszternek két belső mutatója lehet, melyekkel LCB vagy COMB automatákra mutathat.
- A COMB automata kapcsolatainak számát elvileg nem korlátoztuk, az eddigi kialakult mechanizmusok szerint egy LCB-vel azonos szerepű külső-belső mutató páron kívül még maximum három belső mutatója lehet, melyekkel RGB, LCB és COMB típusú automatákra mutathat.

Ha már a modellezési rendszerünk alapját képező automata típusokat a 3. pont példái szerint meghatároztuk, akkor a további specifikáció előtti első lépés az együttműködési mechanizmus alapjainak tisztázása, ami pl. az előzőekben elmondottak szerinti döntésekben nyilvánul meg. Ezután meg kell vizsgálni néhány tipikus esetet, aminek elemzéséből a jelzéstviteli mechanizmussal szembeni esetleges különleges követelményeinket kell levezetni. Ha ez túl bonyolult rendszert eredményezne, akkor célszerű az alapmodellrel változtatni. Pl. azt, hogy az egyszerű, magától értetődő mechanizmus, hogy a kiadandó jeleket a vonatkozó pointer által kijelölt automatának kell továbbítani, nem elegendő, azt az alábbi egyszerű példán mutatjuk meg:

Tekintsük ismét a 3. ábrát, mely a befigyelő hívásban résztvevő automatákat mutatja. Ha a C oldal bont, akkor a 4. ábra szerinti jelzések továbbítása következik be. A COMB automata erről nem értesül, mivel LCB1-öt nem ismeri, és csak a normál hívásban szükséges tennivalókat végzi (B foglaltsági hangot kapna). Viszont erre a jelzésre COMB-nak szüksége van, ha az ilyenkor megkívánt szolgáltatást — vagyis a kezelő és a keresett mellék normál beszédkapcsolatba való összekapcsolását — biztosítani akarjuk. Tehát olyan jelzéstviteli mechanizmusra lenne szükség, mely a CLEAR jelzést a COMB automatának is eljuttatja. Hogy egy ilyen jelzéstvitelésre mikor van szükség, azt aránylag egyszerű módon el lehet dönteni, csak a címzett LCB1-hez tartozó végpont (LSTB) mutatóját kell megvizsgálni. Ha az nem magára az LCB1-re mutat, akkor a jelzést továbbítani kell az általa kijelölt erőforrás automatájának is, vagyis ebben az esetben a COMB-nak.

Az eddigi feldolgozott szolgáltatások során a jelzéstviteli rendszerrel szemben ennél bonyolultabb követelmény nem adódott, sőt ezt is csak a processzorok közötti, vagyis a külső mutatók által kijelölt kapcsolatokban elküldött jelzésekre kellett előírni. Ezért az



4. ábra. Befigyelési állásban a keresett fél (B) partnere (C) bont

eddiggi és az ezután ábráinkon a belső illetve külső jelzéseket meg is különböztetjük (kis-, illetve nagybetűvel írjuk a neveiket).

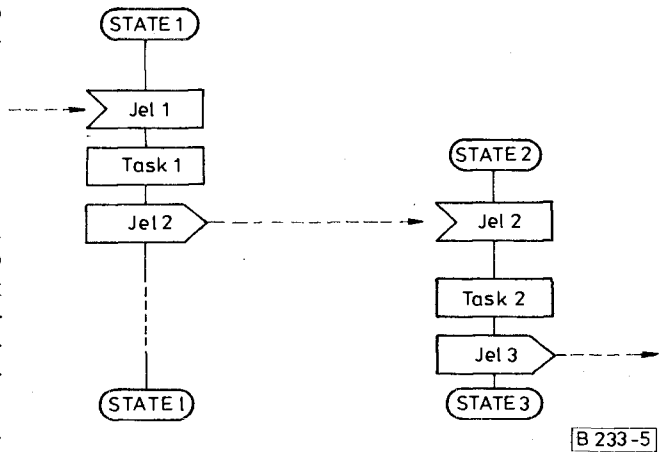
## 5. A hívások lebonyolításának specifikációja

Az egyes hívásfajták lebonyolításának menetét, a normális és különleges forgalmi eseteket az előző pontban definiált automaták együttműködésének leírásával adjuk meg. Ehhez szükségünk van bizonyos formai eszközökre, hiszen itt a verbális módszerek nem elég szemléletesek és főleg igen terjedelmeseek.

Maguknak az automatáknak leírására a CCITT által ajánlott SDL nyelvet alkalmazzuk [1]. Azonban a gyakorlat azt mutatta, hogy szükség van olyan eszközre is, amely az automaták együttműködésének ábrázolására helyezi a hangsúlyt. Az együttműködés ábrázolása bizonyos mértékig az SDL-ábrák segítségével is lehetséges — nevezetesen az input és output jelek grafikus jelölt összekapcsolásával (lásd 5. ábra) — de ez bonyolultabb esetekben terjedelmes rajzokat eredményez. Tömörebb, áttekinthetőbb ábrázolásra van szükség. Ilyet a telefontechnika szakemberei már régen kitaláltak, — létra diagramnak hívják —, ami egy kis formai módosítással választott modelljeink, a véges automaták együttműködésének ábrázolására is kiválóan alkalmasnak bizonyult. A kívánalom itt is, akárcsak a jelfogós áramkörök esetén az, hogy a külső és belső jelek, állapotváltozások ok-okozati, tehát sorrendi viszonyait egyértelműen ábrázoló rajzok szülessenek, mivel a rendszer működését legtöbbször ezek határozzák meg (a jelek sorrendisége a legtöbbször önmagában döntést eredményez a működésekben, kombinációs alternatívák ritkábban fordulnak elő, lásd [2]-ben is). A jelfogós áramkörök esetén a létra, mint leíró eszköz, alkalmas arra, hogy követni lehessen egy változás, esemény hatásának terjedését a rendszeren belül. Aki pedig már tervezett bonyolult jelfogós áramkört, az tudja, hogy a létra mint tervezési segédeszköz is kiváló, egy jó tervező ugyanis először a tervezendő áramkör létráját rajzolja meg, és ehhez tervez jelfogót és kontaktushálózatot. Mi is mint tervezési segédeszközt vezettük be a létrát, amit a bemutatásra kerülő formában *flowgrammnak* kereszteltünk el, utalva arra, hogy az események folyamatainak ábrázolására hivatottak.

A flowgrammokon függőlegesen, felülről lefelé az egyes automatáknak a vízszintesen feltüntetett jelzések hatására végbemenő állapotátmeneteit láthatjuk. A tervezőnek ilyen módon kell megadnia, hogy milyen állapotokból milyen jelzés hatására milyen másik állapotba kívánja hozni a tárgyalt automatáit. Emellett egyszerűen feltüntetheti azt is, hogy az egyes állapotmenetekben melyik, a szolgáltatást igénybevevő végződésről is észlelhető telefonos tevékenység indítását, illetve leállítását (pl. csengetés, hang, időzítés) kívánja elvezettetni.

A hívásfeldolgozó rendszert akkor tekintjük specifikáltnak, ha minden elképzelhető forgalmi szituációra megszülettek a flowgrammok. Ez egyúttal azt



5. ábra. Input és output jelek grafikus jelölt összekapcsolásai SDL diagramokon

jelenti, hogy minden állapotot és jelzést definiálunk. Az ezután következő lépés a flowgrammok szintézise SDL diagramokká, vagyis az egyes automata-állapotokban előforduló átmenetek összesítése és megtervezése. A tervezés alatt az átmenetekben elvégzendő „task”-ok specifikációjának elkészítését értjük. (Az SDL diagramokon ezen task-ok nevei szerepelnek csupán).

Példaképpen a 6–10. ábrákon az egyszerű hívásfelépítés alapeseteit, a 11. ábrán pedig a bonyolultabb forgalmi szituációk közül a már említett befigyelő hívásszituációt ábrázoljuk.

Az ábrák önmagukért beszélnek, különösebb magyarázatot nem igényelnek. Az egyes automata állapotok és jelzések rövid jellemzését a melléklet tartalmazza.

A flowgrammok után elkészített SDL diagramok (12, 13. ábra) és az azokban definiált taskok képezik a tényleges programozói munka kiinduló pontját. Erről, vagyis az „automaták” programozásáról azonban már egy következő cikkben fogunk beszámolni.

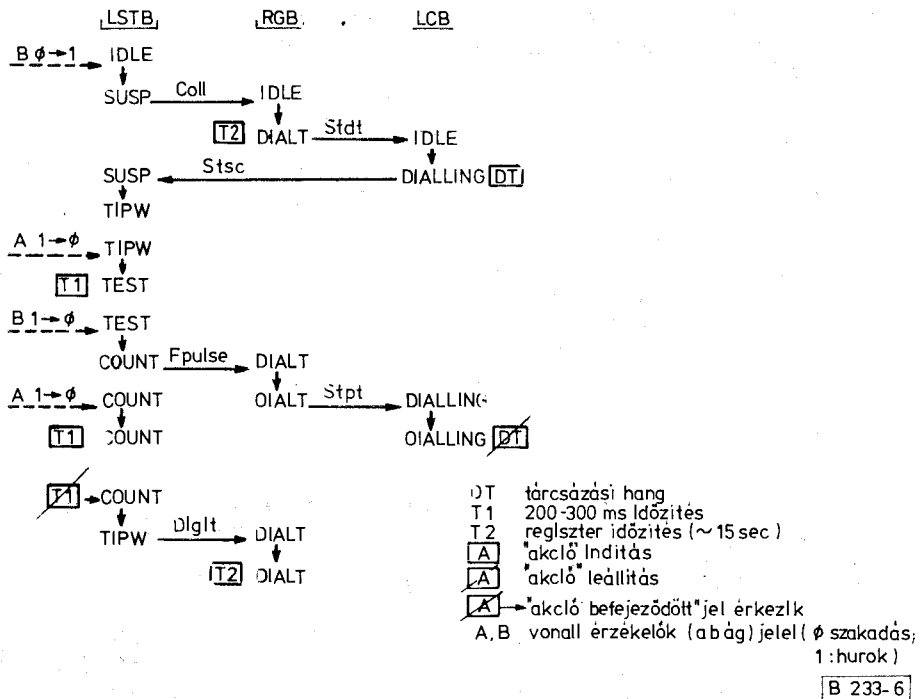
### Melléklet

#### 1. Jellevő automata állapotai:

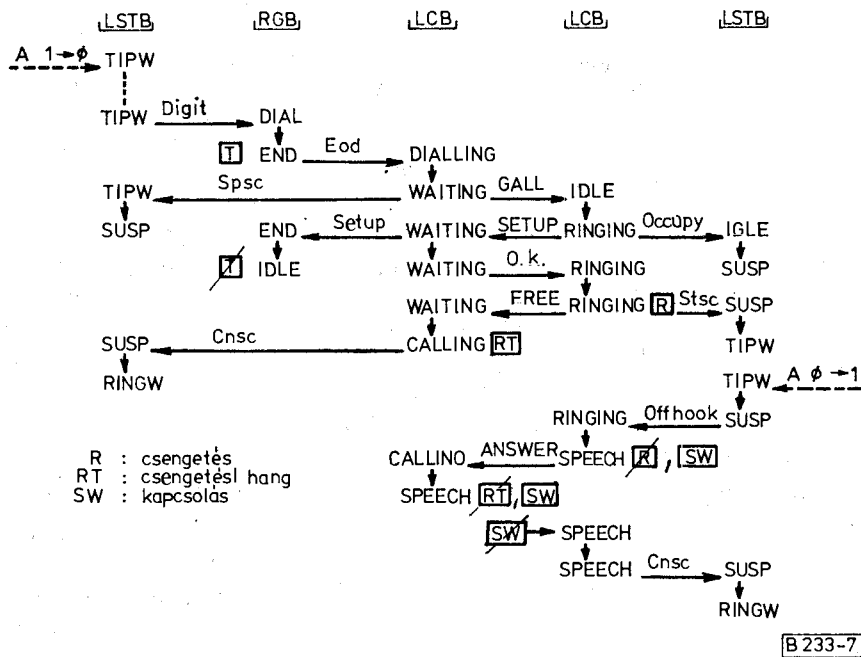
IDLE	Nyugalmi állapot, híváskereső folyik
SUSP	Bejövő jelek érzékelése felfüggesztve
TIPW	Jelek érzékelése mindkét ágon folyik
TEST	A vonali állapot megváltozása után létrejövő állapot: itt kell eldönteni, hogy milyen jelzés indult (szakadás vagy földelőgomb)
COUNT	Számbevételezés folyik
RINGW	Bontásfigyelés folyik

#### 2. Vonali automata állapotai:

IDLE	Szabad, nincs deklarálv
DIALLING	Az RGB működése alatti állapot, funkciója a tárcsázási hang vezérlése



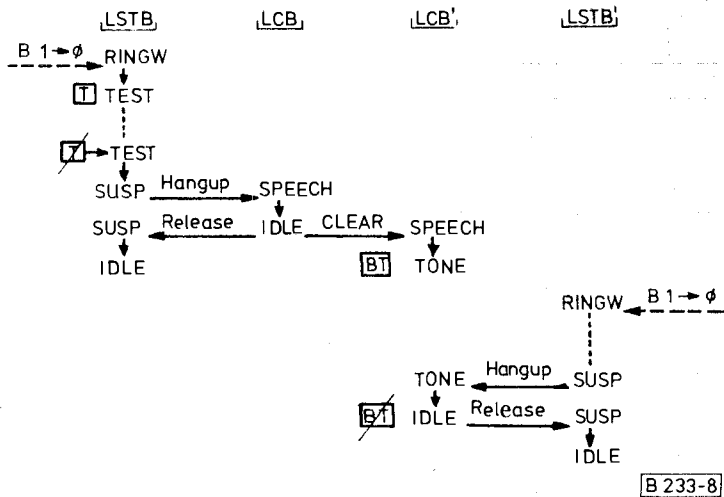
6. ábra. Előfizető hív, első számjegy (2-es) beadása



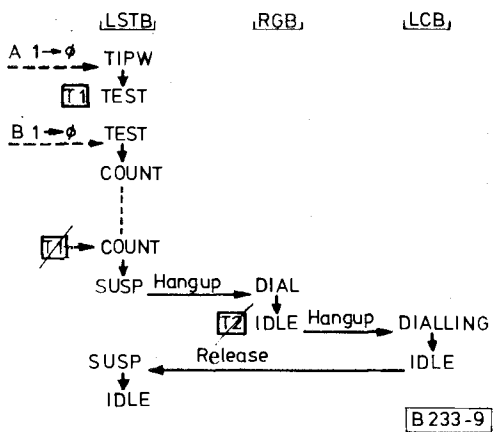
7. ábra. Helyi hívás: utolsó számjegy beadása, hívott csengetése, hívott jelentkezik

WAITING A szabadútkeresés alatti átmeneti állapot  
CALLING Csengetési hang adása folyik  
RINGING Csengetés, hívott oldali útkeresés  
SPEECH Beszédállapot

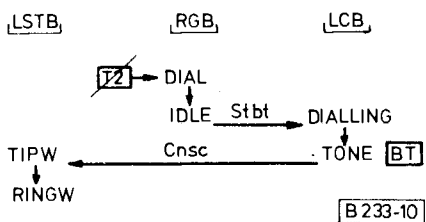
TONE Foglaltsági hang adása  
PATH Közvetítő állás  
OVERRIDE Befigyelési állapot, hívó oldal (Figyelmeztető hang adás)



8. ábra. Helyi beszélgetés hontása



9. ábra. Korai bontás



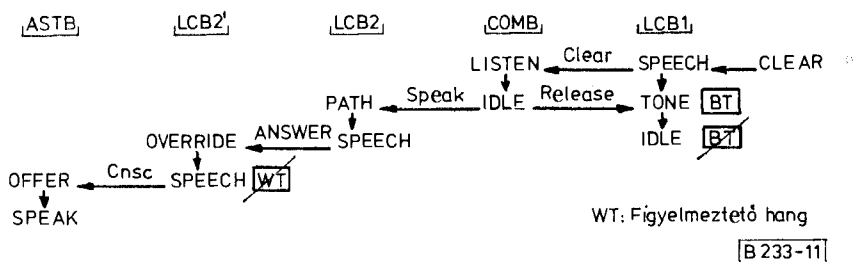
10. ábra. Regiszter „leldőzít”

3. Regiszter automata állapotai:

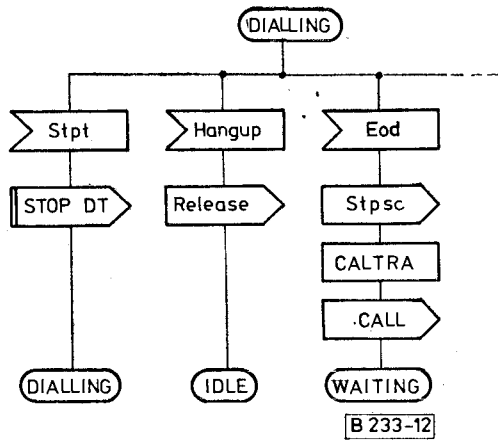
- DIALT Az első számjegy érkezését várja
- DIAL Számjegybevételezés. Időzítés számjegyenként indul
- END Választási információ rendelkezésre áll

4. A külső kapcsolatokban alkalmazott jelzések (üzenetek)

- CALL A hívott és hívó adatait tartalmazó üzenet, a hívott oldali tevékenység indítását kéri
- SETUP A hívott oldali szabad kapcsolóút információt ajánlja fel a hívónak választás céljára
- O. K. A hívó oldal tudott választani, és a kiválasztott útszakaszt közli hívott oldallal
- FREE Hívott oldali csengetés indul, a hívó oldali választást hívott oldal elfogadja
- ANSWER Hívott jelentkezik, beszédállapot létrejött
- CLEAR Bontás (hívó vagy hívott oldal)



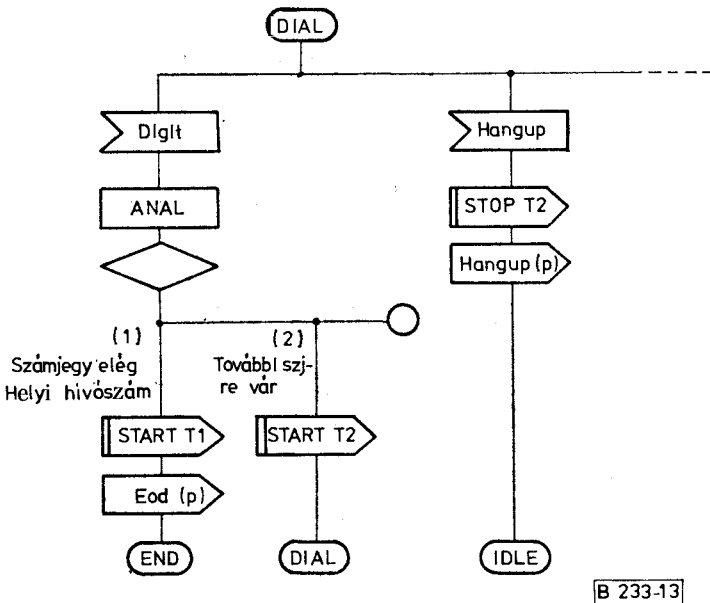
11. ábra. Befigyelési állásban hívott m. a. partnere bont



12. ábra. LCB „DIALING” átmeneteinek SDL diagramja

5. Belső kapcsolatok jelzései:

Call	Hívás keletkezett
Stdt	Tárcsázási hang indítása (Start dial-tone)
Stsc	Számbevételezés indítása (Start scanning)
Fpulse	Első hurokmegszakítás (First pulse)
Stpt	Hang adás leállítás (Stop tone)
Hangup	Végpont bontott
Release	Végpont-felszabadítás
Cnsc	Bontásfigyelés indul (Continue scanning)
Speak	Beszédállásba kapcsol
Stbt	Foglaltságihang-adás indul (Start busy tone)
Stpsc	Jelérzékelés felfüggesztése (Stop scanning)
Eod	Számjegybevételezés vége (End of dialling)
Ring	Csengetés indul



13. ábra. RGB „DIAL” átmeneteinek SDL diagramja

I R O D A L O M

- [1] CCITT-ajánlások; Orange Book, Vol. VI./4. Z101–104
- [2] Horváth Gyula: Távbeszélő központ identifikálása (BHG tanulmány; 1973)
- [3] Horváth Gyula: Telefonközponti ismereteink tudományos feldolgozásának problémái Híradástechnika 20. évf. (1969), (197–202.)
- [4] Makay A.: QA96 programvezérlő rendszere BHG ORION TERTA Műszaki Közlemények. 1980. 11. szám
- [5] M. T. Hills, S. Kano: Programming electronic switching systems 1976. Published by Peter Peregrinu Ltd.
- [6] N. E. Kobrinckij; B. A. Trakhtenbrot: Introduction to the theory of Finite Automata North-Holland Publishing Company, Amsterdam, 1965.
- [7] Horváth Gyula: Kapcsoló központok identifikálása. Mérés és Automatika 24. évf. (1976) 12. szám. (454–458.)
- [8] Horváth Gyula: Távbeszélő központok folyamatábrái Híradástechnika 29. évf. (1978), (39–46. o.)