

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest





MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

FEJEZETEK A RÉSZECSEFIZIKA ELEKTRONIKUS KÍSÉRLETEINEK
ADATGYŰJTŐ, -FELDOLGOZÓ RENDSZEREI KÖRÉBŐL

- Irta:

SEBESTYÉN BÉLA

A kiadásért felelős:

DR VÁMOS TIBOR

ISBN 963 311 162 5

ISSN 0324 2951

ÖSSZEFOGLALÁS

A dolgozat a részecskefizikai kísérlet lényegi vonásainak összefoglalása, a részecskefizikai detektorok és detektorrendszerek rövid áttekintése, majd az elektronikus kísérlet jellemzőinek, felépítésmódjának és építőelemeinek ismertetése után részletesen foglalkozik a CAMAC rendszerrel, ezen belül mindenekelőtt a CAMAC - számítógép kapcsolat kérdéseivel, végül a GYORSSIN /FASTBUS/ leírásával zárul.

SUMMARY

Following the recapitulation of the main features of the particle physics experiment, the short review of the particle physics detectors and detector systems, the description of the characteristics, structures and building blocks of the electronic experiments, the paper deal with the details of the CAMAC system, first of all with the problems of the CAMAC - computer coupleing. The work is closed by the description of the FASTBUS system.

Аннотация: После обобщения важнейших характеристик эксперимента физики элементарных частиц, обзора детекторов и систем детекторов элементарных частиц, ознакомления с параметрами, построением и главными блоками электронного эксперимента, статья подробно занимается системой КАМАК и прежде всего вопросами связи КАМАК - вычислительная машина. Работа заканчивается описанием системы FASTBUS.

T A R T A L O M

	oldal
BEVEZETÉS	7
1. A RÉSZECSEFIZIKAI KÍSÉRLET	10
2. RÉSZECSEFIZIKAI DETEKTOROK ÉS DETEKTOR-RENDSZEREK	16
3. AZ ELEKTRONIKUS KÍSÉRLETI RENDSZER	25
- A kísérleti berendezés	27
- A kísérleti elektronika	29
A trigger elektronika	29
Az esemény-elektronika	32
Az adatgyűjtő blokk	39
A helyi vezérlő	39
Az adatkezelő	40
- A feldolgozó lánc	43
- Két példa	45
4. A CAMAC RENDSZER	50
4.1. CAMAC ALAPOK	51
- Az adatut	51
- Főutak	55
Párhuzamos főut	56
Többágu hálózatok	57
A soros főut	60
4.2. CAMAC VEZÉRLŐK	62
- Passzív vezérlők	62
CCA-1 típusu vezérlő	63
U-vezérlők	63
Ágvezérlők	69
Soros keretvezérlők	70
Hurokvezérlők	72

	oldal
- Automómiával rendelkező vezérlők	74
Parancsgenerálás a programfüggetlen csatornán át	75
Parancsgenerálás parancslistával	76
Blokktvitel	77
Hurokvezérlő autonómiával	82
CAMAC processzorok	82
- Rendszervezérlők, többágú rendszerek egy és több programforrással	85
- CAMAC processzorok általános felhasználási lehetőségei	95
4.3. ELOSZTOTT INTELLIGENCIA	97
4.4. CAMAC MIKROGÉPEK	98
4.5. CAMAC KERET TÖBBSZÖRÖS VEZÉRLÉSSEL	104
4.6. CAMAC PROGRAMOZÁS, CAMAC NYELVEK	106
4.7. NIM-CAMAC EGYÜTTES	119
4.8. AZ IEC-625 CSATLAKOZÁSI RENDSZER	120
4.9. CAMAC VEGYELS	123
- Paralel és soros főutak együttes alkalmazása	123
- CAMAC - külsőkészülék csatlakozás	125
- Modulok tervezési irányelvei	125
- Modulok és rendszer ellenőrzése	126
4.10. ALKALMAZÁSI PÉLDÁK	130
- Multiplexelt analóg-digitális átalakító kezelése és kiolvasása	130
- Proporcionális-kamra kiolvasó rendszerek	132
4.11. CAMAC MÓDOSULATOK	133
5. A GYORSSÍN	141
5.1. A GYORSSIN ÁLTALÁNOS TULAJDONSÁGAI	142

	oldal
5.2. A SZEGMENS	144
- A sin	145
- A tartozék-áramkörök	148
- Az arbitráció	149
- Vezérlő és állapot-regiszterek	150
5.3. CIMZÉS	152
- Logikai címzés	152
- Földrajzi címzés	154
- Közzétett címzés	154
5.4. MŰVELETEK	155
- Null-adatciklusú művelet	
- Egyadatciklusú művelet	157
- Többadatciklusú műveletek, blokkátvitel	158
- Vegyes és többszörös műveletek	159
- Szórványadat-felkeresés	160
5.5. GYORSSIN HÁLÓZAT SZEGMENSEKBŐL	160
5.6. ÁLLAPOTVISSZAJELZÉS	164
5.7. SZOLGÁLATKÉRÉS-KEZELÉS	166
- Szolgálatkérés megszakítási üzenettel	166
- Megszakítás SR útján	167
5.8. DIAGNOSZTIKAI ESZKÖZÖK	168
5.9. A FEJLESZTÉSI MUNKA SAJÁTOSSÁGAI ÉS FŐBB TERÜLETEI	169

IRODALOMJEGYZÉK

ÁBRÁK

BEVEZETÉS

A részecskefizika kísérleti apparátusa az elmúlt évtized során a bonyolulttól a még bonyolultabb irányába fejlődött. Ez a folyamat - legalábbis ha a problémák jelen elméleti és gyakorlati megközelítésmódját szükségszerűnek tekintjük - természetes. Annak érdekében, hogy a fizikus a részecskék jelenségvilágába egyre mélyebbre hatolva kutathassa azok természetét s egymáshoz való viszonyulásaik törvényszerűségeit, kísérleti berendezéseiben mind újabb típusu, ill. energiatartományu kölcsönhatásokat kell előidéznie, és az ezekben létrejövő eseményeket mind finomabb részleteiben kell megfigyelnie. Egy-egy bizonyos típusu kölcsönhatást számtalanszor - esetleg milliárdnyszer - kell előidéznie ahhoz, hogy a statisztikák alapján általános érvényű következtetéseket tehessen, máskor meg azért, hogy valamely igen csekély valószínűséggel előforduló deviánsra bukkanhasson. A kísérlet hatalmas nyers adattömeget szolgáltat, s arra már régóta gondolni sem lehet, hogy erről a fizikus igényelte információt az emberi intellektus közvetlenül válassza le. Ezért kellett áttérni a munka gépesíthető folyamatainak automatizálására. A kísérleti folyamat levezetése, az adatok begyűjtése, szűrése, rendszerezése és kiértékelése mindinkább az elektronikus - tulajdonképpen digitális - rendszerekre hárult, melyek komponenseinek skálája az igen egyszerűtől a szuperszámítógépekig terjed.

A gyorsítótól, az esetleges másodlagos részecskeforráson, a detektorokon és ezek közvetlen elektronikáján, az adatszűrő, -gyűjtő és -feldolgozó egységeken át, a kísérlet eredményeit összegző nagyszámítógépekig - általánosabb értelemben mindezek - együttesen képezik a kísérleti rendszert. Jellemző módon a korszerű laboratórium egyes kísérleti rendszerei nincsenek egymástól hermetikusan elszigetelve, ellenkezőleg, egyetlen nagy hálózatra csatlakoznak, melyben nemcsak a gyorsító, hanem az adatfeldolgozás számos eleme is a különböző kísérletek egész sorát nívótott (gyakran szimultán) kiszolgálni, és pedig az egyes kísérletek információáramait szigorúan külön kezelve.

A fejlődés panorámáját szemlélve, a jelen állapotot a husz év előttihez viszonyítva, a kísérletek automatizálása mellett - vagy inkább azzal szoros összefüggésben - a legjellemzőbb jelenségnek az individuális munkamódszerekről a laboratórium (az intézet) szintjén általánosított módszerekre való áttérés látszik. Bizonyos anyagi és szellemi erőforrásokat központosítani kellett, és ezek éppugy az egyes fizikuscsoportok közös rendelkezésére állnak, mint pl. a gyorsítók is. Az egykor meglehetősen függetlenül tevékenykedő kutató csoportok metodikai szempontból napjainkra igen szoros kapcsolatba kerültek a laboratórium (intézet) integrálódó központi szervezetével. Korszerű munkát csak e szervezet igénybevételével végezhetnek. Haszonélvezői e szervezetnek, de függnek is tőle. A centralizált szervezet és a fizikus csoport kapcsolata nem egyszerű - számos hurokkal kötődnek egymáshoz. Míg korábban a fizikusok munkájának eredményességét főként saját kvalitásaik határozták meg, ma azt döntően befolyásolja a szervezet s a szervezettel való együttműködés minősége is. A tudományosan legkitünőbbben irányított, a legnagyobb elkötelezettség alapján működő fizikusgárda adottságai sem érvényesülhetnek, hacsak nem áll mögöttük oldajozottan működő organizáció és nem kielégítő az ezen organizációval való együttműködés és kölcsönös kommunikáció.

Viszonylag igen rövid idő alatt fejlődtek ki meglehetősen bonyolult szervezeti struktúrák és rendkívül összetett mérőrendszer-struktúrák. E fejlődési folyamat dinamikáját az jellemzi, hogy mielőtt még bármilyen állapot, bármilyen megoldás a maga szintjén tökéletesedhetett volna, máris olyan forradalmi változások következtek be, melyek új munkamódszert követeltek meg, és a tegnapi messze túllépő technikai megoldásokat kínálták. Ebben a helyzetben - különösen a pályára lépő fizikus, számítástechnikus és mérnök számára - nem könnyű az eligazodás. De nem könnyű az eligazodás annak számára sem, aki egy kutatási témán belül egy-egy részletkérdésre koncentrálna - legalábbis egy-egy időszakra kényszerűen - kiejti látóteréből az oldalsávokat.

A jelen anyag az áttekintést igyekszik javítani, de - talán a GYORSSIN leírásától eltekintve - általános értelemben semmi-

képpen sem bevezetésnek, sokkal inkább szintézisnek tekinthető. A részecskefizikai kísérlet lényegi vonásainak tömör körvonalazását a részecskefizikai detektorok és detektorrendszerek rövid áttekintése követi, majd az elektronikus kísérleti rendszer jellemzőinek, felépítésmódjának és elemeinek ismertetése. Ezután a CAMAC rendszerrel foglalkozik, s a CAMAC-on túllépő GYORSIN hálózat leírásával zárul.

Az idestova tizennégy esztendőös multa vissztakintő CAMAC-ról - a szabvány-leírásoktól eltekintve is - számos bevezető jellegű ismertetés jelent meg, ezek sorában egy hazai monográfia is [26]. Semmiképpen sem lehet cél az ezekbe fektetett munkát megismételni. Leírásunk elsősorban a CAMAC funkcionális tulajdonságainak feltárására és az összefüggések megvilágítására törekszik. Fel kell tételeznünk azonban, hogy az olvasó a CAMAC elemeivel más forrásokból előzőleg már megismerkedett. A CAMAC alapoknak a szövegben szereplő célirányos átisméltése nem több, mint a már meglévő ismeretekre való emlékeztetés. A 4. fejezet gerincét a CAMAC-számítógép kapcsolat szervezésmódjának taglálása képezi, mely szervezésmód a CAMAC-hálózat különböző vezérlőinek felépítés- és működésmódjában testesül meg. A CAMAC-hálózat effektív adatátviteli sebessége, a CAMAC-számítógép együttes programozásának módja közvetett kapcsolatban van a választott, ill. a rendelkezésre álló vezérlők tulajdonságaival. Az ebben az irányban való elmélyülés az irodalomból hiányzik, ill. csak szét-szórt töredékekben lelhető fel. Rá kell mutatni azonban arra, hogy CAMAC-számítógép kapcsolaton ma sokkal szervesebb összefüggést kell értenünk, mint kezdetben, tekintettel arra, hogy a mikroprocesszorok korszaka felé haladva a gépi intelligencia a számítógépi oldalról lassanként mintegy átdiffundált a kezdetben passzív hálózatba, ahol végül is mikroprocesszorok, ill. mikrogépek alakjában öltött fizikai formát.

A GYORSIN (eredeti angol nevén FASTBUS) a kísérleti fizikának legalábbis bizonyos területein a CAMAC-ot hivatott felváltani. Formális megfogalmazása után alkotói ugyan még nem tettek pontot, de a jelen szöveg megírásakor már közel áll ahhoz, hogy szabvánnyá válják. A vele való megismerkedés tehát mindenképpen aktuális.

1. A RÉSZECSEKÉFIZIKAI KÍSÉRLET

A részecskefizikai kísérlet általában és lényegében abból áll, hogy mozgási impulzussal rendelkező elemi részecskéket nyugalomban lévő céltárgy részecskéivel vagy tárológyűrűben egymással szemben keringő részecskéket ütköztetnek, miközben regisztrálják, majd elemzik a lejátszódó jelenségeket. Az ütközés impulzussal bíró résztvevői esetenként nem közvetlenül a gyorsító-nyalábból, hanem e nyaláb által gerjesztett másodforrásból származnak.

Az ütközések során új részecskék keletkezhetnek, részecskék semmisülhetnek meg, szóródhatnak egymáson: részecskefizikai események játszódnak le. Az elsődlegesen ütköző részecskéket primer, a reakciók során létrejövőket szekunder részecskéknek nevezik. Az események színhelye lehet semleges (a föld természetes terei) gyakran azonban mesterségesen előállított mágneses tér, ami a töltött részecskékre gyakorolt eltérítő hatása révén a jelenségekben való mélyebb bepillantást tesz lehetővé (impulzus, töltés előjele megállapítható). A kísérlet folyamán az események bizonyos statisztikai eloszlásnak eleget tévő szabályossággal, meghatározott geometriai konfigurációkkal és kinematikai törvényszerűséggel zajlanak le.

Minden kísérlet valamely fizikai modellen alapul. A modell lehet elméletileg megfogalmazott, de lehet lazábban körülírt, intuitív jellegű is. Minden modell ismert adatokra és a priori feltételezésekre épül. A kísérlet igazolhatja vagy elvetheti a feltételezéseket.

A részecskefizikai események lejátszódásának színterét detektorok figyelik, az eseményeket detektorok érzékelik, s a detektorok szolgáltatják az egyes eseményekre vonatkozó információt is.

Az események sorában az adott kísérlet szempontjából hasznos, de haszontalan események is egyaránt találhatóak. Egyes esetekben egyugyanazon részecske-kölcsönhatás más-más kategóriába tartozó eseményeket is kiválthat, másrészt az előírt kölcsönhatás mellett a kísérlettől idegen részecske-kölcsönhatások is létrejöhetnek. A kísérlet lehet tágabb vagy szűkebb aperturájú.

Kiterjedhet több eseménytípusra is, de egy eseménykategórián belül is beszűkülhet valamely korlátozó feltétel alapján. Hasznos eseménynek az számít, amelyik a megfigyelés előre rögzített látóterébe esik. A többi —a véletlenszerű (pl. kozmikus részecskék által keltett) zajjal együtt— kiküszöbölendő háttér.

Az információ-feldolgozás szempontjából kívánatos, hogy az alkalmazott detektorrendszer a megfigyelt eseményekre minél szelektivebb legyen. Vannak azonban olyan detektortípusok, amelyek az eseményeket egyáltalán nem szelektálják. A szelektíóképtelenség többnyire hátrányos, s csak abban az esetben kívánatos, ha a modell igen hevenyészett, a részecske reakciók várhatóan nem egészen úgy zajlanak majd le, mint ahogy azt a tapogatózó elképzelés feltételezi, vagy amikor pl. új részecsketípust keresnek.

Az eseményekről a detektorok által szolgáltatott információ kiértékeléséhez szükség van bizonyos járulékos adatokra is, mint amilyenek pl. elektronikus kísérletnél egyes csatornák kalibrációs adatai, mágneses térerősség, detektor pozíciók stb. A számos eseményről szóló információ e járulékos adatokkal együtt képezi a kísérlet analízisének nyers adatbázisát.

A szorosabb értelemben vett analízisnek két szintje különböztethető meg:

- az esemény-analízis és
- a kísérlet átfogó analízise.

Az esemény-analízis az egyes eseményekről nyert információt a járulékos adatok figyelembevételével egyenként elemzi. Célja az, hogy megállapítsa az egyes események jellemzőit, ennek alapján a hasznos eseményeket elkülönítse a háttértől, s végül, hogy az eseményeket a kísérlet átfogó analíziséhez szükséges módon írja le.

Az esemény-analízis főbb részfeladatai:

- eseményszelekció
- geometriai analízis
- kinematikai analízis
- az egyes eseményekre vonatkozó adatok adatállományba való összesítése, az átfogó analízis adatbázisának előkészítése.

A felsorolás nem rögzíti egyuttal a szigorú időbeli sorrendet is, de még az egyes munkafázisok időbeli elkülönülését sem. Az eseményszelekció részben megelőzheti a többi feladatot, de általában azok végrehajtásával szorosan össze is fonódik.

A geometriai rekonstrukció célja: a rendelkezésre álló mérési adatok alapján az egyes események térbeli konfigurációjának geometriai leírása. Eredményül a kölcsönhatásba lépő és az abból származó részecskék trajektóriáinak, e trajektóriák elágazási pontjainak (csucskok) háromdimenziós koordinátáit nyerik, a mérési és számítási hibák valószínű értékeivel együtt. Amennyiben mágneses tér is jelen van, a geometriai számítások impulzusérték számításokkal egészülnek ki.

A kinematikai vizsgálatok feladata: az ismert geometriai adatok, impulzus értékek és ezek hibaértékei alapján, a megmaradási tételre támaszkodva identifikálni az eseményeket. A kölcsönhatásban szereplő részecskék egy része ismert, a többiek identitását az esemény modellje alapján feltételezzük. Az ismert és a feltételezett adatokat a geometriai rekonstrukció eredményeihez illesztik, töltéssel nem bíró (tehát nyomot nem hagyó) részecskék felléptének lehetőségeit is figyelembe véve. A részfeladatok aszerint variálódnak, hogy a vizsgált esemény modelljében szereplő egyes részecskék tömege pontosan, avagy kevésbé pontosan ismert-e. A kinematikai illesztést az egyes eseményekre iteratív módon optimalizálni kell, miközben a geometriai rekonstrukció is finomodik. Végeredményként a trajektóriák korrigált leírása adódik a hibaértékekkel, valamint a kísérlet által vizsgált eseményben szereplő részecskék azonosítása (tulajdonságaik leírásával) történik meg.

A kinematikai analízis kiegészülhet bizonyos további számításokkal is. Ilyen lehet pl. a szóródási szögnek a laboratórium koordináta rendszeréből a részecskék valamely tömegközéppont koordináta rendszerébe való áttranszformálása.

A különböző típusú kísérleteknél más-más módon, de már a geometriai rekonstrukciót megelőzően sor kerülhet az események részleges szelektálására. A kinematikai analízis lezárultával

viszont a hasznos és haszontalan események végleges elkülönítésének is meg kell történnie. A szelekció kritériumainak együttesét a modell definiálja, s az kísérletenként, sőt egyvalamely kísérlet különböző fázisaiban is, más és más lehet. Adott kísérlethez a szelekciós kritériumoknak több, egymástól eltérő együttese is definiálható, miáltal nemcsak hasznos és haszontalan események különülnek el, hanem a hasznos események is több kategóriába sorolhatók. Az általános szelekciós kritériumok közül néhány jellegzetes:

- a kölcsönhatási pontnak a beeső sugár - céltárgy vagy sugár - sugár metszésének helyére kell esnie;
- a mért koordinátáknak értelmezhető geometriai térgörbék mentén kell feküdniök, az általuk leirt görbe trajektória kell, hogy legyen;
- a részecske trajektóriák térbeli konfigurációjának a kísérlet modellje által feltételezett topológiával kell rendelkeznie;
- a részecske-trajektóriáknak síkba illeszkedőknek kell lenniök.

Az eseményszelekció során igen óvatosan járnak el, a visszautasított események leírását is megőrzik, egyrészt hogy a fizikus céljára ellenőrzés szempontjából rendelkezésre álljon, másrészt mert éppen a visszautasított események között fordulhat elő olyan deviáns, amelyik fizikai szempontból nagyon is érdekes lehet. Az óvatosság különösen indokolt abban az esetben, amikor a szelekció kritériumai nem egészen világosak; mindig fennáll annak a veszélye, hogy a szelekció a kísérlet eredményeit meghamisítja. Általában, de különösen ilyen kétes esetekben jó, ha a fizikus az események geometriai konfigurációit közvetlenül is megszemlélheti. Buborékkamra kísérletnél erre közvetlen lehetőség van, míg elektronikus kísérletek esetében csak a geometriai rekonstrukció után, s grafikus megjelenítő eszközök igénybevételével lehetséges.

A további feldolgozásra kiválasztott eseményadatok tömegét az esemény-analízis utolsó fázisa olyan adatállományba egyesíti,

ami a kísérlet átfogó analizisének adatbázisául szolgálhat. Itt már minden adat fizikai egységben megadva szerepel.

A kísérlet átfogó analizisének két feladatköre:

- a statisztikai analisis és
- az eredményeknek a kísérlet modelljével való összevetése.

A statisztikailag analizálandó események lehetnek egyszerűek vagy összetettek, mely utóbbi esetben az események egyes rész-halmazai külön-külön eseménykategóriákba tartoznak, másrészt adott topológiájú események egynél többféle értelmezése is lehetséges. Bármely vizsgálandó eseményhalmaz eseményeinek fizikai paraméterei közül egy-egy szelektív módon kiválasztható, statisztikai bázis létrehozása céljából. Egy ill. több dimenziós analizist különböztetnek meg, az egyidejűleg vizsgálat alá vont mennyiségek számától függően. A statisztikai analisis eredményeként hisztogramok, eloszlási görbék, képletek, szórási diagramok, értékek adódnak, melyek szögeloszlásra hatáskeresztmetszetre, elágazási arányokra és más fizikai mennyiségekre vonatkozó információt nyújtanak; ezek alapján a részecske-kölcsönhatásokat illető ésszerű következtetések vonhatók le.

Amint azt már fentebb megjegyeztük, esetenként a kísérleti modell meglehetősen lazán van körvonalazva. Lehetséges pl. hogy bizonyos eloszlási görbékre van előzetes feltételezés (parametrikus analisis), máskor hiányzik az erre vonatkozó feltételezés is, a feladatok egyike ilyenkor éppen az ismeretlen eloszlás törvényszerűségeinek felderítése (nem parametrikus analisis).

A valóban hasznos események valamely eloszlásának feltárása igen nehéz abban az esetben (különösen amikor a reakciókban ismeretlen mennyiségű semleges részek is keletkeznek), ha nem hasznos események formailag, sőt kinematikailag is kielégítik a modell által hasznosnak minősített esemény kritériumait.

A kísérleti rendszer szolgáltatja eredmények az elméleti uton megalkotott modellel. csak akkor vethetők össze, ha az eredmények és a modell kimenetei azonos alakban fejezhetők ki. A meg egyezésnek ez a feltétele általában csak jelentős munka révén, s

számos számítógépi óra igénybevétele árán elégíthető ki. Az intuitív modell alapján nyert eredmények igazolása még nehezebb, mivel ez esetben mindenekelőtt a kísérletet is felölelő jelenségcsoport elméletének kidolgozása és összevetésre alkalmas alakban való kifejtése is szükséges.

Hogy valamely kísérlet ideája termékeny, a neki megfelelő modell helyes-e, s végül hogy vajon a modell alapján elképzelt kísérleti berendezés megfelelő-e, az még a kísérlet megindítása előtt jórészt eldönthető számítógépes szimuláció útján. Számítógép útján generálhatók és analizálhatók elképzelt események, "kísérleti adatok" is kaphatók, miknek alapján megítélhető az elképzelések minősége, s finomítható a modell is. De szimulálható a detektorrendszer is, miáltal lehetővé válik a detektorrendszernek a vizsgált eseménytípushoz való illesztése, sőt a nyomfelderítő és illesztő algoritmusok optimalizálása is. A valószínű detektorrendszer tervezése a szimuláció eredményei alapján történhet, de a szimulációs munkák kapcsán alakíthatók ki első közelítésben a kísérleti analízist végző programok is. Noha a szimulációs eljárás különleges felkészülést igényel, és végigvitele időveszteséggel jár, a befektetés a kísérleti munka során bőven visszatérül, általában pedig a kísérlet levezetése szilárdabb alapokon és tudatosabban végezhető el. A szimulációs technika ezért az utóbbi években hangsúlyozottan előtérbe került [1].

A rövid, s a részecskefizikai kísérletet csupán általánosságban és nagy vonalaiban jellemző leírást néhány, a témát érintő összefoglaló cikk [2,3,4,5] idézésével zárjuk.

2. RÉSZECSEFIZIKAI DETEKTOROK ÉS DETEKTOR RENDSZEREK

Noha manapság a részecskefizikai kísérletekben csaknem minden eddig megalkotott detektortípust alkalmaznak (melyiknél melyiket), célszerű mégis e detektorok tulajdonságainak bemutatását kialakulásuk fejlődéstörténetének felvázolása keretében elvégezni.

Vizuális jellegű és jelkimenetű detektorok különböztethetők meg. A vizuális detektor az általa érzékelt eseményről képi információt ad, a jelkimenetű detektor az információt közvetlenül villamos jelre ültetve nyújtja. Jelátalakítóval a vizuális detektor "optikai jele" is villamos jelre transzformálható. A transzformáció nem minden esetben problémamentes, máskor viszont detektor és átalakítója (pl. szcintillációs kristály és elektron-sokszorozó) a gyakorlatban oly mértékben összeforr, hogy együttesüket említik detektorként. Az elmúlt évtizedekben állandó volt az a törekvés, hogy minden vizuális detektorhoz megfelelő átalakítót találjanak, máskor pedig a detektálás elvét ugyan nem változtatták meg, de a detektálás fizikai folyamatával együttjáró jelenségek közül nem az optikai, hanem más, pl. az akusztikai vagy az elektromágneses hatásokat használták fel az információ leválasztására. Vannak összetett detektorok is, amelyek több, különböző alaptípusú detektorból épülnek fel.

Egy másik osztályozásmód szerint triggerelhető és nem triggerelhető detektorokat különböztetnek meg. Ám az eseményszelekciót végző trigger már inkább az elektronikus áramkörökkel is kiegészített detektorok, detektor-rendszerek, semmint az egyes detektor alaptípusok jellemzője.

Az ötvenes éveket, de még a hatvanas évek kezdetét is, jószerevével a buborékkamra uralja. A folyékony hidrogénnel vagy valamely nehézfolyadékkal töltött kamra általában mágneses térben helyezkedik el. Klasszikus alakjában egyidejűleg tölti be a céltárgy és a detektor szerepkörét is. Minden, a térfogatán belül lejátszódó eseményt bizonyos koordinációs vonatkoztatási hálózaton látható nyomvonalak alakjában jelenít meg; a vizuális, nem triggerelhető detektorok legjellegzetesebb képviselője.

A kamra belsejében lejátszódó eseményeket, bizonyos időközönként, 3-8 kamera fényképfelvételek alakjában rögzíti. Az egyes felvételeket követően a kamrát mindig "tisztítják". Az összetartozó felvételek birtokában máris — közvetlenül — végezhető némi értékelés, a pontos feldolgozáshoz azonban a kamrafelvételek digitalizálása és számítógépes kiértékelése szükséges.

A kamrának nincs szelektálási képessége, s így az érzékeny ideje alatt benne lejátszódó minden hasznos és haszontalan eseményt egyaránt rögzít, s igen zavaros regisztrátumot szolgáltat. Ezért az időegység alatt létrejövő események számát a behatoló sugár intenzitásának alacsony szinten tartásával korlátozni kell. A kísérleti követelményeket kielégítő statisztika összegyűjtése emiatt igen hosszú ideig tart.

A csekély sugárintenzitás csökkenti a mesterségesen előidézett reakciók felvételenkénti számát, de nem befolyásolja pl. a kozmikus háttérrel, a felvételek ezért mindenképpen nehezen áttekinthetők maradnak, kiértékelésük nehézkes digitalizálási, alakfelismerési, válogatási feladatok rutinszerű végrehajtását igényli, mely feladatokat, a folyamatos erőfeszítések ellenére máig sem sikerült teljes mértékben automatizálni — az ember jelentős részvétele továbbra is nélkülözhetetlen.

A kamrafelvételek kiértékeléséhez mindenekelőtt a hasznos információt kell — a lehetőség szerint — a háttértől elkülöníteni. Először is elvetendőek azok a felvételek, amelyek hasznos eseményt eleve nem tartalmaznak. Az ember szerepe különösen a durva válogatásban jelentős, mivel a legfejlettebb automatikáknál is sokkal fejlettebb alakfelismerési képességekkel rendelkezik. A válogatás megkönnyítése érdekében a felvételek mindegyikét e célra szolgáló munkaasztalok síkjára vetítik ki. Válogatás után a megmaradó felvételeket digitalizáló asztalokon tapogattják le, melyek síkján a regisztrátum kinagyított képe jelenik meg. A letapogatás az idők folyamán a manuális off-line módszerektől a számítógéppel vezérelt automatikus nyomkövető eljárásokig fejlődött, azt azonban a legfejlettebb módszerek esetében is az ember választja ki, hogy melyik csúcspontból kiindulva, mely nyomvonalakat kell digitalizálni. Az egész munkafázis ne-

nézkéssége akkor válik világossá, ha arra gondolunk, hogy kísérletenként százezernyi felvételt több dimenzióban kell feldolgozni annak érdekében, hogy a kísérlet nyers adatbázisa előálljon.

A digitalizálásnak egy a nyomvonalonkénti eljárástól eltérő másik, a felvételeknek televíziós technikára emlékeztető módon, soronként fényponttal való letapogatása módszerét is kifejlesztették. Az alakfelismerési feladat itt azonban igen nagy nehézségekkel jár még akkor is, ha (mint teszik) a számítógépi programok dolgát bizonyos — digitalizáló asztalon nyert — (a csúcokra és nyomvonalakra vonatkozó) utbaigazító adatok közlésével könnyítik meg. Ugyanezen nehézségek leküzdésére speciális alakfelismerő processzorokat is kidolgoztak [6].

Jelenleg csak egyetlen olyan eszköz ismeretes, mely a kamra által nyújtott optikai információ közvetlen, film nélküli digitalizálását lehetővé teszi, a TV kamera (vidikon v. ortikon). Az ézidőszerint rendelkezésre álló TV kamerák azonban egyrészt csak kisméretű kamrák esetében nyújtanak kielégítő geometriai felbontást, másrészt mivel alkalmazásuk esetén mindenféle előválogatás elmarad, az alakfelismerési feladatok a lehető legösszetettebb formában jelentkeznek, a többdimenziós képtartalmak igen redundánsak, hatalmas adattömegük nagyon nagy tároló kapacitást és feldolgozási időt igényel, ezért ez a módszer általánosságban nem terjedt el, inkább csak speciális esetekben alkalmazzák.

Noha a buborékkamra, különösen kezdetben, igen hatásos és pontos kutatási eszköznek bizonyult, és számos új részecske felfedezéséhez járult hozzá, tulajdonságai a fejlődés hozta követelményekhez nem mindig illeszkedtek. Az energiahatár kitolódásával mind nagyobb kamrákat kellett építeni (30 m³-ig) miközben a regisztrálás pontossága a folyadék-turbulenciák és az optikai leképzés pontatlansága miatt romlott, de meg azért is, mert az alkalmazható legnagyobb mágneses térerősség esetén is a részecskének a kamra térfogatán belülielhajlása (görbülete) csekély. Az a körülmény pedig, hogy a buborékkamra kísérletek adatgyűjtését nem lehetett on-line módon megoldani, lehetetlenné tette az adatoknak legalább részbeni időreális módon való feldolgo-

zását. A fizikus ezért a kísérlet folyamán nem kaphat ugyszólván semmiféle információt a dolgok menetéről; hiányzik az informatív visszacsatolás.

Miközben a fizikusok egy része kitartott a buborékkamra mellett, s azt igyekezett tökéletesíteni, más részük — perspektívájában látva a nehézségeket — új technika után nézett. A teljesen automatizált, tisztán elektronikus kísérlet irányába tett első lépés a már korábban is használt szikrakamra tökéletesítésével vette kezdetét (1958-60).

A szikrakamra érzékeny terén átfutó részecskék, nyomvonalak mentén ionizálják a teret kitöltő gáz molekuláit. A szikrakamra elektródjaira vitt impulzusszerű nagyfeszültség hatására az ionizált utszakaszok mentén szikrakisülés jön létre, a nyomvonalak ezzel láthatóvá válnak, és lefényképezhetők. A szikrakamra ezen változatai ugyancsak vizuális kimenetűek, de közöttük és a buborékkamra között mégis jelentős különbségek vannak. A leglényegesebb ezek között az, hogy itt a tulajdonképpeni kamrát egy a vizsgált események feltételezett topológiáját, az eseményben résztvevő részecskék energiáját és más paramétereit figyelembevevő, részecskeszámlálókból és elektronikájukból álló triggerrendszer egészíti ki. A triggerrendszer az eseményeket két osztályra bontja, a trigger által jelzett első osztályba tartoznak (maradék háttérrel együtt) a hasznos események, míg a másik, a trigger által kirekesztett események osztályába mind haszontalan események sorolódnak. Az első osztályba tartozó esemény érzékelését triggerjel jelzi. A kamrára a triggerjel kapcsolja rá a nagyfeszültséget, szikrakisülés csak hatására léte-sülhet. De a triggerjel működteti a kamerákat is, melyek így csak meghatározott eseményekről készítenek felvételeket. A triggeretechnika alkalmazása révén a buborékkamrákéinál sokkal tisztább felvételek adódnak, miáltal nem csak a nyom-digitalizálási, hanem az alakfelismerési feladatok is lényegesen leegyszerűsödnek. Ezen eredmények birtokában fokozni lehetett a sugár intenzitását, növelve az időegységre eső hasznos események számát is; azonos statisztikákat rövidebb idő alatt lehetett begyűjteni, s az adatfeldolgozás időtartama is lerövidült.

A szikrakamra gáztöltése már csupán a nyomvonalak érzékelésére szolgál, a céltárgy szerepkörét a kísérlet célkitűzéseinek megfelelően választott más anyag tölti be.

A buborékkamra konstrukciója a vele végzett kísérletek során mindig egy és ugyanaz - a szikrakamra felépítésmódja kísérletenként más és más lehet, mindig az adott kísérlet modelljére optimalizálható.

A szikrakamrának az idők folyamán igen sokféle, sik-, ill. hengerelektrodás, átlátszatlan fólia-, ill. az eseményeket minden irányból megpillanthatóvá tevő transzparans-elektrodás változata jött létre, s ezek felépítés- és alkalmazásmódjuktól függően két vagy háromdimenziós regisztrátumokat szolgáltathattak. Kis tértartományra szorító eseményeket a buborékkamra pontosabban regisztrál, a nagy tértartomány eseményekre nézve azonban a szikrakamra nyújt jobb térbeli felbontóképességet.

A több részecske széles térszögű divergenciájával együttjáró kölcsönhatások vizsgálata megkövetelheti ugyan, hogy a szikrakamra mágneses térben helyezkedjék el, mágneses tér alkalmazása azonban nem minden esetben szükségszerű, már csak azért sem, mivel a szikrakamrához társítható Cserenkov számlálók sebesség- és töltés-azonosító információt is nyújtanak.

A szikrakamra vizuális változata, ha lényegesen könnyített formában is, de mégis csak implikálja azokat a nehézségeket, miket a fénykép-technikának az adatgyűjtő-feldolgozó láncban való jelenléte hoz magával. A detektálási módszerek fejlődésvonalán ezért ismét egy elágazási pont jelent meg. A szikrakamrával dolgozó fizikusok egy része kitart a fénykép-technika mellett, s a buborékkamrákkal együtt a felvételek kiértékelési eljárásainak tökéletesítésén fáradozik, nem is eredménytelenül. A CERN-ben pl. már 1964-ben rendelkezésre állt egy olyan fénypont letapogatású digitalizálóval kombinált számítógépes képanalizátor, amelyik rugalmas szóródási kísérletből származó mintegy 10^5 szikrakamra felvételt 10^3 /óra teljesítménnyel dolgozott fel. A kutatók más része továbbhalad a tisztán elektronikus detektálás irányába. Kisebb szikrakamrákhoz eredményesen alkalmazák a TV kamerát, de létrehozzák az akusztikus szikrakamrákat

is, melyek nem a szikra fényhatása, hanem hanghatása alapján lokalizálják a nyomvonalat. A szikra helyzetét a kamrába beépített mikrofonok érzékelik, a hang terjedési sebességének figyelembevételével. Az eredendően analóg kimenőérték digitalizálása digitális időmérővel történik. Itt már nincs fénykép, a kamra által nyújtott információ teljességében villamos jelekből áll.

A szikrakamrák egyes változatainál (nagy elektródaközű kamrák) nehézségbe ütközhet a nyomvégződés pontos helyének lokalizálása. Az akusztikus detektálásba hibákat vihet be az is, hogy a hang terjedési sebessége a fizikai feltételekkel változik. Továbbá, a trajektória helymeghatározása elveszíti egyértelműségét, na több mint egy részecskét kell nyomon követni, s a többértelműség feloldása nem éppen egyszerű feladat. Részben ezen elégtelenségek kiküszöbölésére, részben pedig a tér és időbeli felbontóképesség fokozása érdekében jöttek létre a huzal-szikrakamrák.

A huzal-szikrakamra legegyszerűbb változata egy síkelektrodából, s egy ezzel párhuzamos síkban kifeszített, paralel huzalok képezte másik elektródából áll. A részecske áthaladási helyét a nyomvonalával szomszédos huzalt megütő szikra, ill. a szikrától megszólaló huzal jelzi. Két ilyen egyszerű szikrakamrát olyképp egyesítve, hogy huzaljaik iránya egymással derékszöget zárjon be, a részecske áthaladási helye a sík x , y koordinátaival határozódik meg. A többretegű kamrák térbeli információt nyújtanak. Az egyes kamrasíkok által szolgáltatott információt lényegében a benne megszólaló huzal vagy huzalok pozíciószáma - címe - adja meg, melynek megállapítása különböző módokon történhet. A ferrittárolós kamra esetében, minden egyes huzalhoz egy ferritgyűrű tartozik; a szikra által megütött huzalok ferritmagja átmágneseződik (átbillen). A ferritmagokat sorban, cím szerint, egy letapogató rendszer kérdezi végig, mely a külvilággal azoknak a huzaloknak a címét közli, amelyeknek ferritmagját átbillent állapotban találja. A magnetostriktív szikrakamránál (1. ábra) a huzalsík összes huzalja egy magnetostriktív késleltető vonalra csatlakozik. A vonal kimenetén fekvő érzékelő a megütött huzal helyétől függő késéssel érzékeli a szikrabeccsapódást. E késési időt digitalizáló TD átalakító kimenetén a megütött huzal pozícióját leíró adat jelenik meg.

A szikrakamrák kezdetben "erősáramu" tartományban, nagyintenzitású szikrával dolgoztak. A nagyintenzitású szikrának csupán másodlagos hátránya, hogy a detektorelektronikában zavart kelt. Nagyobb hibája: nagy, több ms holtideje. A helyreállási időt a szikra intenzitásának csökkentésével, mintegy két nagyságrenddel sikerült leszorítani, majd a szikrakisülési tartományból a proporcionális tartományra tértek át, létrehozva a proporcionális huzalkamrákat. Ez utóbbiak időbeli felbontóképessége a ns tartomány alsó határa közelébe esik. A proporcionális huzalkamra időbeli felbontóképessége minden jelenlegi igényt kielégít, viszont igen kicsi a huzaljele (1-5 mV), ezért minden huzalhoz erősítőt kell alkalmazni. A térbeli felbontóképesség érdekében igyekeztek csökkenteni a huzalközöket, egy-egy detektorrendszerben több tízezer huzal jelent meg, az ennek megfelelő huzal-elektronikával együtt. Tekintetbe véve az itt felmerülő áramköri sebességi követelményeket is, világos hogy egy-egy ilyen detektor-rendszer közvetlen elektronikája még akkor is rengetegbe kerül, ha az építőelemeknek bizonyos csoportba való integrálása lehetséges is.

Jó térbeli felbontóképességet ritkásan elhelyezett huzalokkal is sikerült elérni, a sodrási kamrák létrehozása révén. Ezen ugyancsak proporcionális tartományban működő kamrák esetében a megszólaló huzal pozícióján túl, a részecske áthaladási és az általa keltett töltésnek a huzalra való beérkezési időpontjai között eltelt időtartam is jellemző az áthaladás helyére. Az áthaladás helyének meghatározásához tehát, meg kell állapítani egyrészt a megszólaló huzal címét, másrészt mérni (digitalizálni) kell a fent említett időtartamot. Az utóbbi feladatra TD átalakítókat kell alkalmazni. Érdekességként meg kell említeni, hogy a sodrási kamra időbeli felbontóképessége nem rosszabb, mint az egyszerű proporcionális kamráé, noha a töltések beérkezési ideje az 1 μ s tartományig terjedhet.

A proporcionális és a sodrási kamrák — éppugy, mint a szikra-huzalkamrák is — az egyes kísérletek szükségleteire optimalizálva, igen sokféle méretben és geometriai konstrukcióval készülnek.

A proporcionális (sodrási) kamrának nincs szüksége nagyfeszültségű impulzusforrásra — táplálására egyenfeszültség szolgál. A tápforrást tehát nem kell triggerelni, ezért az e kamrák mellett jelenlévő trigger-rendszer nem annyira a kamrának, mint inkább a kamrát is felölelő detektor-rendszer egészének tartozéka. Tekintettel arra, hogy a kamrák működési sebessége a néhány ns időtartamok tartományába esik, jeleik bevonhatók az események gyors szelekciójába, a triggerképzésbe is.

A proporcionális (sodrási) kamrák a kísérlet követelményeinek megfelelően mágneses térben is elhelyezkedhetnek.

A huzalkamrák és huzalkamra etyüttesek: részecske trajektoriakat érzékelő koordináta-detektorok. Koordináta-detektorként még a huzalkamrák megjelenése előtt, de jelenleg is használnak szcintillációs számlálókból felépített együtteseket is. Ezekben az elrendezésekben a kamrahuzal szerepét vékony szcintillátor csik és a hozzátartozó fényvezető és elektronsokszorozó játssza. E detektortípusnak főleg a proporcionális kamrák megjelenéséig volt nagyobb jelentősége; azokon a helyeken alkalmazták, ahol a szikrakamra időbeli felbontóképessége nem volt keilégítő. A szcintillációs koordináta detektor valóban gyors, de térbeli felbontóképessége szerény, a szcintillátorok mérete egy határon túl alig csökkenthető, számuk nehezen fokozható. Térbeli felbontóképessége legfeljebb ± 3 mm, míg ez már a szikrakamrák kezdeti változatainál is elérte a $\pm 0,5$ mm-t. A szcintillációs számláló és sokszorozó e mellett igen drága, több tizezer darab alkalmazása szóba sem jöhet. A proporcionális kamrákkal szemben a szcintillációs detektor a részecskéknek a szcintillátorokon való többszörös szóródásának, ill. abszorbciónak nagyobb valószínűsége miatt is hátrányban van. A szcintillációs koordináta detektorokat ma inkább trigger-rendszer hodoszkópjaként alkalmazzák.

Időközben a klasszikus buborékkamrák is továbbfejlődtek, több specializált típusuk jelent meg, így a kriogén, a gyors ciklusu és a nagy felbontóképességű kamrák. Megjelentek továbbá a hibrid kamra-rendszerek, melyekben buborékkamrákat alkalmaz-

nak más, közvetlen jelkimenetű kamratípusokkal. Ez utóbbiakról [7], részletesebben [4] tudósít. Továbbfejlődtek a vizuális kamrák és a TV technika összekapcsolásának módszerei is [A1].

A jelen és a jövő lehetőségeinek mérlegelése tekintetében, e közlemény aspektusa szempontjából, a fenti vázlatos ismertetésen túl szükségtelen a detektálás technikájának vagy a fejlődés egyes fázisainak változataira részletesebben kitérni. Elsősorban a két évtized előtti és a jelenlegi helyzett közötti különbséget kell hangsúlyozni és azokat a tényezőket szem előtt tartani, amelyeknek serkentésére a változások létrejöttek. A buborékkamra helyét mára, a kísérletek többségénél, a különböző huzalkamrákból, egyéb — egyszerűbb — részecske-detektorokból, a környezeti feltételeket figyelő érzékelőkből komponált (esetenként igen sokrétű) rendszerek foglalták el. A fénykép-szelektáló laboránsok hadának, a manufakturális módon dolgozó fizikusoknak a helyét automatikák vették át, a jelkimenetű detektor-rendszerekhez közvetlenül csatlakozó on-line adatgyűjtő, feldolgozó láncok. A korábbi kamra-kísérlettől való megkülönböztetésül nevezik az utóbbi kísérlettipust "elektronikus kísérlet"-nek.

3. AZ ELEKTRONIKUS KÍSÉRLETI RENDSZER

Az elektronikus kísérleti rendszer lényegében azzal jellemezhető, hogy a benne feltalálható összes részecskefizikai detektor — közvetlenül vagy átalakítója révén közvetve — villamos jelkimenetű, s hogy kísérleti adatainak legalábbis az összegyűjtése és tárolása mindig on-line, időreális módon történik, de általában időreálisan történik az adatoknak valamely részleges vagy az adatok összeségéből kiemelt mintának teljes analizise is.

Az elektronikus kísérleti rendszer három fő funkcionális része:

- a kísérleti berendezés
- a kísérleti elektronika
- a feldolgozó lánc.

A kísérleti rendszer négyféle,

- a) eseményleiró
- b) feltételleiró
- c) folyamatleiró
- d) ellenőrző

információ forrása, ill. felvevője és feldolgozója is. Az esemény-információt a kísérleti berendezések részecskefizikai detektorai, a kísérleti feltételeket leiró információt pedig részben nem magfizikai érzékelői szolgáltatják. A kísérleti feltételeket leiró információ más része adatok alakjában állhat rendelkezésre (pl. detektorgeometriára vonatkozó adatok), továbbá a részecskeforrástól származik, ugyancsak adatok vagy jelek formájában. A kísérleti folyamatot jellemző információt a főként a kísérleti elektronikában elhelyezkedő regisztráló eszközök (pl. triggerszámláló) gyűjtik össze, míg az ellenőrző információt a rendszer különböző pontjain található figyelő - ellenőrző áramkörök nyújtják, valamint maga a rendszer generálja saját gépi ellenőrző programjainak hatására.

A b) és c) típusu információ a kísérleti információ részét képezi és pedig a b) típusura főként az eseményanalízis a c) típusura inkább az összesítő analízis kapcsán van szükség. E kettőt összefoglalóan járulékos vagy kísérő információknak nevezzük.

Az ellenőrző információ generálására a kísérlet menetében, a rendszer munkájának ellenőrzésére, megbízhatóságának fokozása érdekében van szükség. Kezelése és feldolgozása a kísérleti információétól elkülönítve történik.

Az elektronikus kísérleti rendszer felépítésmódját és a részecskeforrással való kapcsolatát a legáltalánosabb módon a 2. ábra szemlélteti.

Részecskeforráson a gyorsító, a nyalábkivívő (beam transport) és az esetleges másodforrás együttese értendő. Ha a gyorsító egyidejűleg több különálló kísérletet lát el, az egyes kísérletektől függetlenül üzemel még az esetben is, ha kísérleteket ellátó számítógépek szolgáltatásait igénybe venné is. Fordítva azonban a gyorsító vezérlőjelei hatnak a kísérleti rendszerekre, s az adatok feldolgozásakor szükség lehet bizonyos gyorsító-adatok ismeretére is. A nyalábkivívők és esetleges másodforrások tágabb értelemben az egyes kísérletek, kísérleti rendszerek tartozékait képezik, s így közöttük és a kísérlet között szoros informatív kapcsolat van.

A részecskeforrás és a kísérleti rendszer közötti informatív kölcsönkapcsolat a forrásvezérlőn keresztül létesül. A 2. ábra a forrásvezérlőt épp csak jelzi; a valóságban az egyszerű téglalap mögött huzódik meg az egész bonyolult (számítógépes) gyorsító automatika, a nyalábkivívő és a másodforrás szabályozó-vezérlő rendszerével együtt. Ezeknek részleteire a továbbiak nem térnek ki.

A fizikus a részecskeforrás és a kísérleti rendszer működési paramétereit, a kísérleti feltételeket és a kísérlet menetét közvetlenül (α, β, γ csatorna) vagy a feldolgozó láncon át követve (δ csatorna) kísérheti nyomon. Másrészt, ugyanezekre

közvetlen (a, b, c csatorna) vagy közvetett (d csatorna) befolyást is gyakorolhat.

A terminál a kísérleti rendszer legmagasabbszintű információ-megjelenítő, ill. beavatkozó eszköze. A fejlett kísérleti rendszer rajta át közölt paranccsal indítható, ill. állítható le, segítségével adhatók be egyes kísérleti és számítástechnikai paraméterek, általa követhető nyomon a legpontosabban a kísérlet menete, ellenőrizhető a rendszer munkája, végezhető — szükség esetén — hibavizsgálat stb. A terminálon át érheti el a fizikus a folyamatban lévő adatfeldolgozás különböző aspektusú összesítő eredményeit is, éspedig időreális módon, a kísérlet menetideje alatt. A terminál a kísérlet egészének lefutását befolyásolni képes, s az emberen keresztül záródó informatív visszacsatolás legmagasabbszintű eszköze, amennyiben az általa nyújtott komplex adatok a kísérlet fizikai tartalmába engednek betekintést.

A kísérleti berendezés érzékeny térfogata az a hely, melyben a mesterségesen létrehozott részecskefizikai kölcsönhatások, a kölcsönhatásokkal járó események lezajlanak. Gerincét a detektorrendszer képezi, melyet azonban nem részecskefizikai jellegű (analóg és digitális kimenetű) érzékelők, mágnesek, tápforrások, mechanikai helyzetérzékelők, ill. - beállítók, vákuumtechnikai eszközök, automatika elemek, ellenőrzés feladatkörű elemek stb. egészíthetnek ki.

A részecskefizikai detektorok kétféle feladatkört, esemény-szelekciót és eseményregisztrációt, láthatnak el. Az előbbieket trigger-detektoroknak, az utóbbiakat esemény-detektoroknak nevezzük, megjegyezve, hogy az elhatárolás nem éles — egyes detektorok ui. mindkét feladatkör ellátásában részt vehetnek.

A trigger-detektorok (a hozzájuk tartozó döntési elektronikával együtt) annak érzékelésére szolgálnak, vajon a kísérleti berendezésben lejátszódó esemény a hasznost is implikáló eseménykategóriába tartozik-e vagy eleve haszontalan. A trigger-detektorok meglehetősen durva térbeli-kinematikai, de igen jó

időbeli-energetikai felbontóképességü együtttest alkotnak.

Az esemény-detektorok a trigger által elfogadott eseményről szóló információt szolgálhatják. E detektoroktól, jobban mondva együttesüktől, elsősorban minél jobb térbeli-kinematikai felbontóképességet követelnek meg, bár számos kísérletnél az esemény-detektorok jó időbeli felbontóképessége is követelmény.

Jelenleg, különösen a terjedelmesebb kísérleti berendezésekben, a detektoroknak csaknem minden típusa megtalálható: huzalkamrák változatai, szcintillációs, Cserenkov és félvezető detektorok, kaloriméterek, sőt még TV-kamerával kiegészített vizuális kamrák is [7].

A nem részecskefizikai mennyiségek érzékelésére termopárok, nyomásérzékelők, villamos és mágneses mennyiségek érzékelői, pozíció-érzékelők stb. szolgálnak. A kísérleti berendezés által nyújtott kísérő információ ezektől származik.

A legtöbb kísérleti berendezés nem kizárólag rögzített (az egész kísérlet során változatlan állapotú) elemekből épül fel, de általában még a belső és külső üzemi feltételek sem mindig azonosak. Szükség lehet mozgatható elemek (pl. detektorok, detektor-csoportok) helyzetének, egyes kísérleti paraméterek (pl. klíma meghatározók, feszültség, áram, gáznyomás és összetétel) értékének előírt program szerinti változtatására. Az egyes munkapontokban a kísérleti feltételek stabilizációját is megkövetelik. E célokra szolgálnak a kísérleti berendezés beavatkozó szervei, melyek kézi uton és/vagy jelvezérlés útján működtethetők.

Működés ellenőrzésére szükség lehet a kísérleti berendezésben állapotjelzőket elhelyezni, melyek vagy folyamatosan szolgáltatnak kimenőjelet, vagy amelyek kimenőjele időszakonként lekérdezhető. A kísérleti berendezésben olyan elemek is helyet találhatnak, melyek a kísérleti rendszer átfogóbb ellenőrzésében játszanak szerepet. Ilyenek pl. a szcintillációs detektorok s e detektorokból álló nagyobb detektorcsoportok munkájának ellenőrzésében szerepet játszó fénydiódák.

A kísérleti elektronika az adott kísérlet, az adott kísérleti berendezés követelményeihez illeszkedő sajátos áramköri együttes. Sajátos, ám nem szükségképpen minden részletében egyedi is. Napjaink kísérleti elektronikája viszonylag kevés egyedi áramkört tartalmaz, nagyjából intézeti, még inkább nemzetközi szabványoknak eleget tévő szokványos modulokból épül fel.

A kísérleti elektronika horizontálisan és vertikálisan tagozódik. A 3. ábra, mely főként a horizontális bontást tükrözi, öt funkcionális részt különböztet meg. Ezek:

- trigger-elektronika
- esemény-elektronika
- adatgyűjtő
- helyi vezérlő
- adatkezelő.

A valóságban az áramkörök szétválása nem ennyire tiszta. Egyes áramkörök egyszer egy, máskor másféle, sőt egyidejűleg többféle funkciót is elláthatnak. Így lapolódnak át esetenként a trigger-, ill. az esemény-elektronika egyes részei, nem is beszélve pl. az adatgyűjtő blokkokról és a helyi vezérlőről, amelyek korszerű alakban egyetlen összevont adatgyűjtő-folyamatirányító egységet képezhetnek.

A trigger-elektronika feladata annak eldöntése, vajon adott esetben a trigger-detektorok együttes megszólalása a hasznos eseményt is magába foglaló esemény-kategória feltételeinek felel-e meg vagy sem. E feltételek térbeli, időbeli, részecske-identitásra vonatkozó előírásokat ölelhetnek fel. A detektorokból és elektronikából álló triggerrendszer a hasznos események hátterét lecsökkenti ugyan, de teljesen ki nem küszöböli, így a triggerrel kiválasztott események halmazában továbbra is találhatóak (a trigger feltételeknek azonban megfelelő) haszontalan események is.

Az áramköri döntésnek az esemény, illetve a reá vonatkozó információ rögzíthetőségének időléptékében, tehát igen gyorsan kell megtörténnie. A trigger-elektronika lényegében koincidencia

hálózat, melyhez azonban erősítő, formáló, késleltető, diszkriminátorok, limiterek, kiágazók (fan-out) jelösszevonók (fan-in), kapuk és további járulékos áramkörök valamely együttese is tartozik.

Esemény elfogadását a trigger-elektronika kimenőjele (röviden a trigger) révén közli; a trigger elmaradása egy adott esemény kapcsán, annak elutasítását jelenti. Az esemény-elektronika, ennek megfelelően, a hozzátartozó detektorok kimenőjeleit csak a trigger hatására fogadja és kezeli, a triggerrel nem kiváltó eseményt feljegyzetlenül hagyja. A trigger ad parancsot az eseményenként feljegyzendő kísérő információ begyűjtésére, regisztrálására és kezelésére is. A trigger vezérli (ha ilyen a kísérleti rendszerben egyáltalán van) a triggerrelendő detektorokat is.

A trigger-elektronika az esemény-információ feljegyzését megelőző szelekciót végez. Mindinkább szaporodnak azonban az olyan kísérleti rendszerek, amelyekben nem csak előszelektáló, hanem az esemény regisztrálását követő, utószelektálást végző áramköri egységeket is alkalmaznak. Az utószelektáló áramköröket esetenként lassu triggereknek is nevezik. Kétségtelen, hogy maga az utószelektálás időtartama lényegesen hosszabb lehet az előszelektálásénál, azonban ezt az időt mindenképpen igyekeznek leszorítani, s ezért e célra is gyors áramköröket alkalmaznak; a "lassu" megjelölés tehát félrevezető lehet. Az utószelekcióval kapcsolatban a trigger kifejezés sem szerencsés, mivel a szelektálást gyakran processzor áramkörök végzik, a döntés processzálás eredményeként születik. Megkülönböztetésül, a továbbiakban csak az előszelektálást végző áramkört nevezzük triggernek, míg az utószelektálást végzőt: esemény-szűrőnek.

Az imént leírtakat szem előtt tartva egy- és többfokozatu trigger különböztethető meg. Az egyfokozatu trigger az összes feltételt együttesen vizsgálva dönt, ezzel szemben a többfokozatu — több időbeli lépésben. A 4. ábrán példaképpen bemutatott többfokozatu trigger [8,9] első fokozata a beeső részecskék azonosítását végzi, mintegy az előszelekció előszelek-

cióját. A második fokozat triggerjelet csak akkor adhat, ha saját, az eseményre vonatkozó — térbeli-időbeli — feltételeinek teljesülése mellett az első fokozat előirt részecskét i-
dentifikál. Az adott konkrét esetben a részecske azonosság 3 ns, az összes triggerfeltétel egybevetése 10 ns időt igényel.

A 3. ábrán láthatóan, a triggerjel végzi a kísérleti be-
rendezés triggerjelet igénylő detektorainak "gyújtását" is,
ezenkívül az adatgyűjtő blokk ugyancsak a triggerjel hatására
végzi el az eseményenként szükséges járulékos információ begyűj-
tését.

A trigger-elektronikának minden triggert követően retesz-
lődni kell arra az időre, míg az esemény-információ, és a hozzá
járuló kísérő információ rögzítése, majd az adatkezelő általi
átvétele (továbitása) meg nem történik. A reteszelés az adat-
kezelő végzi, egyrészt érzékelve a triggerjelet, másrészt azt a
pillanatot is, melyben a kísérleti elektronikától az elfogadott
esemény információjának átvétele lezárult, és ez már kész újabb
esemény regisztrálására. Célszerűen, a reteszelés nem a trigger-
jel képzésére, hanem kibocsátására vonatkozik. A 4. ábrán fel-
vázolt konkrét rendszer esetében [8,9] is ily módon jártak el.
Itt a trigger-elektronika két számlálóval egészül ki, az egyik
az adott idő alatt létrejövő triggerek számát, a másik az ugyan-
azon idő alatt kibocsátott triggerek számát rögzíti. A két szám
különbsége az eseményregisztráció okozta holtidő veszteség.

A trigger-elektronika üzemi paramétereinek értéke eseten-
ként változtatható kell, hogy legyen, továbbá a kiértékelés kap-
csán szükség lehet a koincidencia feltételek ismeretére is.
Ilyen esetben kézzel (3. ábra c' csatorna) vagy programvezér-
lés útján állíthatók pl. a diszkriminátorok (küszöbértékek),
késleltetők (idő), impulzusszélesség (koincidencia kimenőjel),
tápfeszültség (elektronsokszorozó munkapontja) stb., a koinci-
dencia információ kezelésére pedig amplitudó- és időkonverterek,
tároló elemek stb. állhatnak rendelkezésre [10]. Mód lehet a
koincidencia bemenetek és a trigger-detektor kimenetek kézi-
vagy programvezérlés általi kombinatív összekapcsolására is,

az esetenként változó triggerfeltételekhez való alkalmazkodás megkönnyítése céljából [10].

A fentiekből kitűnik, hogy a trigger-elektronika, esetenként nem pusztán triggerjelet, hanem esemény- és kísérő információt (koincidencia-információ) is képez, ha pedig üzemi paraméterei változtathatók, akkor vezérlő adatokat is igényelhet.

Triggerjelekre csak a gyorsító impulzusainak idején van szükség, a szünetekben a triggerképzést tiltani kell. Az engedélyezést és tiltást ugyancsak az adatkezelő végzi. A reteszelés, ill. engedélyezés/tiltás hatásvonalát a 3. ábrán a TILT/ENGED összeköttetés jelzi. A 5. ábra a reteszelés, ill. tiltás/engedélyezés idődiagramját szemlélteti.

Az esemény-elektronika az esemény-detektorok analóg jeleinek vételére, erősítésére és formálására, a jelek hordozta információ digitalizálására, tárolására és előírt formátumu adatok alakjában való előkészítésére szolgál. Elemi áramkörei tipikusan: erősítők, formálók, (logikai és lineáris) kapuk, késleltetők, analóg-digitális (amplitudó, töltés, idő) átalakítók, huzalpozíció-azonosító, kódoló és tároló áramkörök.

Az esemény-detektorok analóg jeleit esetenként trigger képzésre is felhasználják (erősítést és formálást követően), ilyenkor az esemény-elektronika egyes áramkörei résztvesznek a triggerjel képzésében is.

Az eseményelektronikának is lehetnek kiegészítő, így pl. állapotjelzésre, esemény-szimulációra, az üzemi paramétereknek kézi uton vagy programvezérlés útján történő módosítására szolgáló áramkörei.

Az esemény-szimulációra szolgáló áramkörök, a feldolgozó láncról érkező rendelkezések (és adatok) alapján az esemény-elektronika bemenetein meghatározott jellegű, hamis és igaz eseményeknek megfelelő jelkombinációk sorozatát szolgáltatják. E sorozat hordozta információ esemény elemző – statisztikai-feldolgozása pontosan meghatározott eredményt kell, hogy adjon;

az ettől való eltérés a rendszer hibás működésére vall. Az esemény-szimuláció lehetővé teszi a rendszer egészének átfogó ellenőrzését.

Esemény-szimuláció a kísérleti rendszer más-más szervezési szintjein is lehetséges; végezhető pl. nem csak a detektorok és az elektronika csatlakozási síkján, hanem az adatkezelő bemene-tén is, de kiterjeszhető a detektorokra (vagy inkább átalaki-tókra) is. Szcintillációs effektusok utánózhatók pl. fénydió-dákkal.

A detektor szintű esemény-szimuláció kiterjeszhető egyéb-ként a trigger-rendszerre is, a triggerszituációk és a trigger-áramkörök beállítása és ellenőrzése céljából.

Az ellenőrző áramkörök jelenléte nem csak a kísérlet folya-matában, hanem a kísérletre való felkészülés idején is igen meg-könnyítheti a fizikus munkáját.

Napjaink kísérleti rendszereit a proporcionális és sodrási kamrák -- általában a huzalkamrák -- elterjedt alkalmazása jellemzi; nem ritka a több ezer, sőt több tizezer huzalszámu kamrarendszer sem. A nagy huzalkamra együtteseket felölelő kísérleti rendszerek elektronikus áramköreinek sorában súlyará-nyánál fogva különleges szerepet tölt be a kamra-elektronika. E súlyarányra jellemzőképpen az "Európai muon kollaboráció" kísérleti rendszerét említjük meg [11], melynél a kamraelekt-ronika 16400 információ-csatornájával szemben a nem kamra jel-legű detektorok csatornaszáma mindössze 1892. Jelentősége miatt szükséges a kamra-elektronika némely kérdésével külön is foglal-kozni.

A kamra-elektronika horizontális irányban kamrák szerint, az egyes kamrákon belül síkok szerint (vagy hengeres geometria esetén hengerek szerint) oszlik meg. Az egyes síkok (hengerek) modulárisan tovább oszthatók szektorokra. Vertikális irányban a kamraelektronika huzal-elektronikából, kódoló és tároló áramkö-rökből épül fel.

A huzalelektronika erősítőből, formálóból, késleltető, kapu és tároló áramkörökből áll, melyek a sodrási kamra esetében még a sodrási időt mérő digitális időmérővel is kiegészülnek. Szemléltetésül a 6. ábra a proporcionális kamra huzalelektronikájának két változatát [12,13] mutatja be. Az analóg huzaljel mindkét esetben erősítő plusz formálóra kerül, mely [12] esetben diszkriminátorként is működik, állítható küszöbvel. A küszöbbemenet szolgál ellenőrzés céljára is, amennyiben adott esetben ezen át szimulálható huzaljel. A huzalelektronika kimenetén mindkét oldalán kapuzott 1 bit-es tároló bistabil található. A bemenő erősítő és a kimenő bistabil között a két, [12] és [13], áramkör bizonyos — nem lényegi — eltérést mutat. Az analóg huzaljel a monostabil késleltetőn alakul át logikai szintté és ez a jel iródik kapuzójel hatására a tároló bistabilba. A kapuzást a triggerjel, ill. ennek származéka végzi el. A közbenső késleltetésnek a triggerjel képzés idejével kell arányban állnia.

A [13]-ban leírt huzalelektronika mellett a trigger-elektronikához tartozó gyors és lassu döntési blokkok is láthatók, amelyek az analóg és a digitális kamrajeleket belevonják a triggerképzésbe. Bár bővebb utalás erre vonatkozóan [13]-ban nem található, fel kell tételeznünk, hogy a huzalelektronikához csatlakozó gyors és "lassu" döntési blokkot még egy nagyon gyors trigger-áramkör is meglözi. Az ábrán látható megoldás minden esetre a trigger fokozatok egymást követésének sajátos módját szemlélteti.

A huzalkamra és így a több kamrából álló kamraegyüttesek minden huzaljához tartozik egy-egy tároló bistabil, minden egyes huzalnak és a hozzátartozó bistabilnak van egy saját és csakis őt jellemző, az irodalom által huzalcimnek is nevezett száma. A huzalelektronika bistabiljainak összessége, első közelítésben, egy a huzalok számával megegyező bitszámú regiszterként is felfogható. E regiszter azon bitjei, amelyeknek huzaljai megszólalnak, trigger jel hatására átállítódnak, a többi bistabil pedig alapállapotban marad. Ilymódon a kamra-információ a kamra-regiszterbe iródik át. A következő feladat megállapítani az

esemény kapcsán megszólalt huzalok címét. Legegyszerűbb módon ezt a növekvő sorszám szerinti végigtapogatással lehet elvégezni. Egy ilyen univerzális — soros — letapogatót, mely egyúttal a huzalcímek bináris kódját is szolgáltatja ismertet [14].

A [14] letapogató összesen 8192 huzalkimenetet fogadhat. A huzal-bistabilok egyetlen léptető regiszterbe szerveződnek. Az alapelemekhez tartozik még egy, a léptetőregiszter tartalmát mozgató óraimpulzusok regisztrálását végző szinkronszámláló, valamint egy puffertároló és ennek bemenő regisztere is.

A letapogató működési elve a 7. ábra alapján megérthető. Az esemény regisztráció, majd kis késéssel a letapogatás folyamata a triggerjel hatására indul meg. A megszólaló huzalok helyén az óra regiszterbe mindenhová bináris '1' érték íródik. Rövid idővel ezután az óra jelei az ÓRAJEL bemenetre jutnak, s megkezdődik egyrészt a léptetőregiszter tartalmának a nyíl irányában való mozgatása, másrészt az óraimpulzus jeleinek számlálása is. Ahol a léptetőregiszter soros kimenete az ÉS kapun megszólalt huzalt jelez, ott a szinkronszámláló tartalma a bemenő regiszter közvetítésével a puffertároló soron következő rekeszébe íródik. E tartalom pedig éppen a megszólalt huzal sorszámának — címének — bináris kódja. Mint látható tehát, a bemutatott áramkör nem csak letapogat, hanem bináris kódolást és tárolást is végez. A valóságban ez a letapogató áramkör valamivel bonyolultabb, mint ahogy leirtuk; nem rögzíti ui. az egymás szomszédságában megszólaló valamennyi huzal címét, hanem azok közül csak egyét, melyhez 3 bites járulékként csatolja a vele egy csoportban megszólaló huzalok számát. A puffertárolóban a címek 16 bites kódok alakjában tárolódnak. A járulékos tag alkalmazásával jelentős tároló-kapacitás megtakarítást lehetett elérni.

A letapogató 40 MHz max. órafrekvenciával dolgozik, s így a teljes címskála végigjárásához 205 μ s idő szükséges, ami igen jelentős holtidő komponens. A soros letapogatás tehát meg lehetőségen lassu eljárás.

A soros letapogató a kamrahuzalok cimtartományát egyetlen strukturálatlan mezőként kezeli. Mőd van azonban a cimtartomány-
nak kamránkénti és sikonkénti strukturálására is. A teljes cim-
tartomány természetes módon kisebb résztartományokra bomlik fel.
Minden résztartomány konstans azonosítóval látható el, és az
egy-egy résztartományokon belül a letapogató az előbbi módon, de
egy-egy külön letapogatóval végezhető. Az így adódó paralel-
-soros rendszer a tisztán sorosnál már lényegesen gyorsabb, de
ennek az az ára, hogy már nem egy, hanem a résztartományoknak
megfelelő számú letapogató-kódoló szükséges. Ha pl. a 8192 hu-
zalt tartalmazó rendszer 32 - egyenként 256 huzalt felölelő -
tartományra bomlik, úgy a letapogató ideje ugyan 6,4 μ s-ra
csökken, viszont 32 drb letapogatóra van szükség.

A cimtartomány strukturálásának következő lépése a szektorok-
ra bontás. Ilyen, szektoronként 16, egészében 1024 huzalra kiter-
jedő tartományu kódolót [15, A2] szemléltet a 8. ábra, és pedig
az a/ ábra az egy szektorhoz tartozó modult, a b/ ábra pedig a
teljes kódolót mutatja be. Minden egyes modulra 16-16 huzal
elektronikájának kimenete csatlakozik. A 16 huzal által képviselt
kamrainformáció kapujel hatására a modul bemenő regiszterébe
íródik. E regiszter kimenete kétfelé ágazik el, egyrészt a
CAMAC adatut, másrészt a szomszédos megszólalások elnyomása blokk
irányába. A CAMAC adatuton a nyers kamrainformáció közvetlenül
hozzáférhető. A szomszédos megszólalások elnyomása blokkokra
proporcionális kamra esetén van szükség, amelynél egy áthaladó
részecke hatására több szomszédos huzal is megszólalhat. E
blokk a pillanatnyi kívánalmaknak megfelelően érvényesíthető
vagy hatástalanítható. A 16 bit-es kamrainformáció, mely egynél
több huzal (huzalcsoport) megszólalásait is tükrözheti, ezekután
a soros letapogató és a paralel kódoló együttesére kerül, ahon-
nan a megszólalt huzalok címei bináris kódok alakjában a gyors
adatvonalra kerülnek át. Az egyes modulok lekérdezése soros
módon történik, először az 1. majd a 2. végül a 64. modul lekér-
dezése fut le. A tiltás vonal kezdetben az 1. kivételével
minden modult tilt. Az 1. modul lekérdezése után a 2. modul
szabadul fel, és így tovább. A lekérdezés az egyes modulokon
belül is sorrendi; több megszólaló huzal esetén először

legalacsonyabb értékű cím kerül a gyors sinre, majd sorban a magasabb értékűek. A teljes huzalcím olyképpen adódik, hogy a modulon belüli 4 bites kódhoz (16 bemenet) hozzáfűződik a modul 6 bites (64 modul) azonosítója is. Így az 1024 huzalra kiterjedő teljes címtartománynak megfelelő 10 bites címek adódnak. A letapogatást 40 MHz-es órafrekvenciával végzi. Az adott megoldás csak akkor jelenthet előnyt a [14]-ben leírt soros letapogatóval szemben, ha nem az összes címet, hanem csak azokat kérdezi le, amelyek megszólaló huzalt jeleznek, a meg nem szólaló huzalok címét pedig átugorja. Feltételezve pl. hogy az 1024 huzal közül csak 7 szólal meg, így a lekérdezés ideje mindössze is 175 ns, mely valamelyest még kibővül az információ kódolásra való előkészítésének idejével is. Szerzői az adott áramkört elsősorban gyors szűrőprocesszorok adatelőkészítése céljára hozták létre. Feltételezték, hogy a processzálas első fázisaként nem magukat az adatokat, hanem a címtartományon belül megszólaló huzalok (részcscék) számát kell ellenőrizni. E szám képzésére szolgálnak a modulokban az aszinkron működő összeadók, melyek közül a 64. kimenetén a címtartományon belül megszólaló összes huzal száma ~40 ns idő alatt áll rendelkezésre. A binárisan kódolt címek (10 bit) és az eseményszám (4 bit) gyűjtő tárolóra kerül. Tekintettel arra, hogy e tároló adott esetben csak nyolc kódolt címet képes befogadni, a kódoló címtartományán belül egyidejűleg legfeljebb nyolc megszólaló huzal címét képes szolgáltatni.

A kódolás problémája nagyon leegyszerűsödne, ha szektoronként csak egy huzal (huzalcsoport) megszólalásával kéne számolni. A feladat ekkor egyszerű kombinációs áramkörökkel vagy prioritás logikájú kombinációs áramkörrel volna megoldható.

A kamrainformáció kódolása — tömörítése — a huzalkimenetek logikai kombinációjával is lehetséges a [8,9]-ben ismertett (1. még jelen tanulmányban a "Két példa" c. pont alatt) szcintálliciós hodoszkópoknál alkalmazott módon.

A fentebb leírt — erősítőből-bistabilig terjedő — huzalelektronika csupán egyetlen adatot közvetít: megszólalt-e a hoz-

zá csatlakozó huzal vagy sem. Létrehoztak olyan huzal-elektro-
nikát is, amelyik a huzalpozícióra vonatkozó puszta igen/nem
adaton túl, energiára és időre vonatkozó információt is képes
felvenni. Egy ilyen, nagyobb huzalcsoporra kiterjedő rendszert
a 9. ábra szemléltet [16]. A huzal-jelek ez esetben igen nagy
sebességgel lépkedő analóg léptető regiszterekre kerülnek, me-
lyek híven megőrzik az egyes huzalok által szolgáltatott esemény-
-impulzusoknak nem csak amplitudóját, hanem egymáshoz viszonyi-
tott időbeli helyzetét is. Miután az esemény lezajlik, a gyors
léptető jeleket szolgáltató forrás leáll, majd a regiszterek
tartalmát a begyűjtésinél lényegesen kisebb frekvencia tapogatja
le. Ezenközben digitalizálódik az egyes vonalokról származó im-
pulzusoknak nem csak az amplitudója, hanem egymáshoz viszonyított
térbeli és időbeli helyzete is. Az amplitudó digitalizálását az
ábrán látható A/D átalakító, a térbeli és időbeli digitalizálást
pedig az analóg multiplexert és az A/D átalakítót irányító ve-
zérlőegység végzi el.

Külön is említésre méltó a fenti kapcsolásban alkalmazott
analóg léptető regiszter. E regiszter a töltéscsatolás elvén
alapul: n-típusú szilíciumlapkán kiképzett azonos kapacitású
kondenzátorok sorából áll, elején és végén injektáló, ill. ki-
vevő áramkörrel [17, 18, 19]. A beinjektált töltés a kondenzá-
tor-sorra adott két vagy háromfázisú órafrekvencia hatására
— megfelelő körülmények között lényeges veszteség nélkül —
lépésenként a kivevő áramkörig továbbbitódik.

A kamra-információ strukturálása, a szektorokra bontás elve
sugallja az áramköröknek modulokba tömörítését (l. [15, A2])
technológiai szempontok alapján is. A huzaláramkörök felépítés-
módja az esetek nagy részében a 6. ábrán bemutatottakéhoz ha-
sonló. Az esetenként szükséges több tízezer ilyen áramköri egy-
ség előállítási munkái és az előállítással egybekötött költsé-
gei korántsem csekélyek. Ésszerűnek látszik tehát az integrálás
lehetőségeit feltárni, és azzal élni. Az integrálás lehet huza-
lonkénti vagy bizonyos számú huzalból álló csoportra kiterjedő,
amint azt [12] szemlélteti. A csoportosítást és integrálást cél-
szerűen a választott kódoló áramköri követelményeihez kell il-

leszteni, s szükség esetén a vegyes (hibrid) technológiák alkalmazására is sor kerülhet.

A sodrási kamrák huzalszáma ugyan kisebb, mint az azonos felbontóképességű proporcionális kamráké, s ennek arányában csökken a huzalonkénti áramkörök száma, viszont a 6. ábrán látható elemek még a sodrási idő mérésére szolgáló T/D átalakítókkal is kiegészülnek, amiért is az egy huzalra eső elektronika terjedelmesebbé válik.

Korántsem kimerítve, inkább csak érzékeltetve a lehetőségeket, a fentiekben elsősorban arra kívántuk ráirányítani a figyelmet, mennyire fontos az a módszer, amelyik szerint a kamra-információ kivonása és kódolással való tömörítése történik. E kérdésekre a fizikusoknak nem csak a kísérleti elektronika, hanem már a kamrák tervezése során is gondot kell fordítaniok.

Az adatgyűjtő blokkba összevont elektronikus áramkörök szerepe, hogy külső vezérlés irányítása alatt közreműködjenek a nem magfizikai érzékelőktől és a forrásvezérlőtől származó kísérő információ begyűjtésében és digitalizálásában, fogadják és továbbítsák a kísérleti berendezés állapotleíró, valamint a forrásvezérlő vezérlő jeleit (pl. a gyorsító impulzus kezdő/záró jelét). Ennek a feladatának megfelelően az adatgyűjtő blokkban jelkondicionáló, erősítő, zajszűrő, analóg és digitális multiplexer, mintavevő, A/D átalakító, kódoló és tároló, valamint az ezek kezelését ellátó áramkörök találhatók.

A helyi vezérlő részben saját algoritmusai, részben a feldolgozó láncról vagy a fizikustól közvetlenül kapott parancsok alapján gondoskodik egyrészt a kísérlet előirt feltételeinek betartásáról, másrészt e feltételek előirt program szerinti megváltoztatásáról. A helyi vezérlő állítja be, ill. módosítja a tápfeszültségeket, mágnesező áramokat, hőmérsékletet, gáznyomást, mechanikai pozíciókat stb. Az adatgyűjtővel együtt hatva a kísérleti paraméterek stabilizálására szolgáló zárt szabályozóláncot alkot. A helyi vezérlőn át (c''' csatorna 3. ábra) hajthat végre a fizikus manuális beállítási műveleteket, s végül a

helyi vezérlő láthatja el a kísérleti berendezés szimulációt szolgáló elemeit (pl. fénydiódákat) előirt eseményeket szinlelő ingerjelekkel.

A helyi vezérlő lehetséges elemei: D/A átalakítók, demultiplexerek, tároló regiszterek, elektromechanikus eszközök (pl. léptető-motorok) hajtó áramkörei, vezérlő áramkörök stb., továbbá az adatkezelővel való kapcsolat ellátására szolgáló logikai áramkörök. Mint a kísérleti rendszer minden funkcionális egységénél, általános esetben itt is megtalálhatók a helyi vezérlő működésének nyomonkövetésére szolgáló ellenőrző áramkörök.

Az adatkezelő a kísérleti elektronika és a feldolgozó lánc közötti csatlakozás és kommunikáció eszköze. Fogadja a kísérleti elektornikától érkező jelzéseket, továbbítja a feldolgozó láncre a kísérleti elektronika által előkészített adatokat, közvetíti a feldolgozó lánc kibocsátotta, s a kísérleti elektronikának szóló vezérlő adatokat, parancsokat és ellenőrzés rendeltetésű adatokat. Az adatkezelő dolgozza fel továbbá a forrásvezérlőtől származó vezérlőjeleket, s e jelek figyelembevételével tiltja vagy engedélyezi a triggerképzést, s ugyancsak ő gondoskodik a trigger eseményenkénti reteszeléséről is.

Az adatkezelő a kísérleti elektronikától az adatokat előirt átviteli formátummal és átvitel módban veszi át, és az átvételétől esetleg eltérő formátummal és átvitel módban továbbítja.

Az átvitt adatok a kísérleti láncre valamely a kísérlet helyéhez közelebb vagy távolabb eső pontján, de minden körülmények között puffertárolóba kerülnek, s gyakran feldolgozásuk is már e puffertárolóban elkezdődhet. A feldolgozó láncre több puffertároló is elhelyezkedhet, de magának az adatkezelőnek is lehet puffertárolója. Ebben az értelemben tároló nélküli vagy tároló adatkezelő különböztethető meg. Tároló adatkezelőről akkor lehet beszélni, ha puffere egynél több esemény adatainak felhalmozására képes. Szélső esetben az adatkezelő puffere az egy gyorsító impulzus időtartama alatt generálódó összes adatot befogadhatja. Az ilyen adatkezelő a kísérleti elektronika többi egységével

együtt általánosabb értelemben vett adatgyűjtőt alkot.

A feldolgozó láncnak (esetenként az adatkezelőnek) az adat-átbocsátó képessége véges, vagyis gyorsító-impulzusonként valamely maximális számú bitből álló adathalmaz átvételére képes. Ezzel együtt a gyorsító-impulzusonként regisztrálható események száma is maximálva van. Az adatkezelőnek ezért ellenőriznie kell a regisztrált események számát, és ha e szám eléri a maximumot, a trigger minden más tényezőtől függetlenül le kell tiltania. Az 5. ábrán a trigger engedélyező jel hátsó frontján szaggatott vonal jelöli azt az esetet, amikor az eseményszám már a gyorsító-impulzus lezárulta előtt eléri a maximumot.

Alapesetben az adatkezelő lényegében csupán passzív, (esetenként azonban igen kiterjedt) adatgyűjtő hálózat, mely különböző sinekből és autonómiával nem rendelkező sinvezérlők-ből áll, ami néhány speciális (pl. trigger reteszelő-tiltó/engedélyező) elektronikus blokkal egészül ki. Az adatgyűjtést mint aktivitást e hálózat igénybevételével a feldolgozó lánc végzi, az általa a sinvezérlőkhöz kibocsátott parancsok útján. A fejlődés további fokán az adatgyűjtést végző hálózat adatszétosztásra is alkalmazható. Az elektronikus kísérletek fejlődését tekintve, abban — igen sok más (és részben már említett) tényezőn túl — jelentős szerepet játszik magának az adatkezelőnek a fejlődése. E fejlődés fő jellemzője, hogy az adatkezelő először is az adatgyűjtési és szétosztási műveletek végrehajtásában mind több szerepet játszik. A puffertolt adatkezelő pl. kell hogy rendelkezzen olyan egyszerű vezérlővel is, ami a kísérleti adatoknak a pufferbe való begyűjtését végzi. Az adatkezelő korszerűbb formái már rendelkeznek az adatok blokkonkénti, vagy csomagonkénti gyors továbbításának lehetőségével, az adatkezelőben alkalmazott puffert pedig kézenfekvően sugallta az adatkezelőnek processzorral való kiegészítését is. A korszerű, helyi tárolókkal és processzorokkal is ellátott, eseményszűrést és előszelektálást is végző adatkezelők a CAMAC és NIM rendszer keretében és az integrált-áramköri technológia eredményeinek hatása alatt és ezek felhasználásával alakultak ki.

A kísérleti elektronikáról az eddigiekben igen általánosító, egyszersmint azonban leegyszerűsítő képet is adtunk. A cél azonban nem az, hogy minden meglévő rendszert, minden megszületett technikai megoldást maradéktalanul az általunk megadott vázlat körvonalai közé kényszerítsünk, hanem, hogy a kísérleti rendszer funkcionális szervezémódjának főbb vonásaira mutassunk rá. A konkrét rendszerek elektronikáiból számos itt leírt komponens hiányozhat, míg mások kiegészülhetnek olyanokkal, melyekre eddig csak utaltunk, de amelyekre részletesebben csak később térünk ki. A trigger-elektronika és az esemény-elektronika mindig megtalálható (noha gyakorta igen csupasz alakban), a kísérő információ begyűjtésére és a helyi vezérlésre szolgáló elemek azonban (különösen a rövid átfutási idejű egyszerűbb strukturájú kísérleteknél) többnyire csak kezdetleges alakban vannak jelen; a kísérő információ felvételére a fizikus által közvetlenül leolvasható műszerek és digitális számmegjelenítők, a kísérleti feltételek beállítására, ill. módosítására manuális beavatkozók állnak rendelkezésre. Hiányozhatnak az állapotjelzők is, vagy állapotjelzőként esetleg egyszerű indikátorok szolgálnak. Noha az áramkörök funkciók szerinti csoportosítása a 3. ábrának megfelelő módon mindig elvégezhető, és a kísérlet tervezése és előkészítése során a világos elhatárolás célszerű is, a megépített rendszerek valóságában a funkcionális elválasztást követő fizikai elhatárolás nem mindig történik meg; különböző funkciót végző elemek topológiailag keveredhetnek.

A 3. ábra a kísérleti elektronika szervezémódjának elsősorban horizontális vonásait emeli ki, azonban a gyakorlatban általában a bemutatottnál sokkal többértübb a vertikális tagozódás. A detektoroktól a feldolgozó lánc irányába tekintve, többnyire külön csoportosulnak az igen gyors áramkörök és külön a mérsékelt sebességűek. Az adatkezelőnek is több szintje lehetséges, melyek közül a legfelső az utószelekciót végző szűrőegységek, azután az adatgyűjtő-szétosztó hálózat elemi sinjei és processzoraik stb. következnek a feldolgozó lánc küszöbéig. Ám kérdés: hol található az ilyen komplex rendszerek esetében a feldolgozó lánc küszöbe? Sehol. Az elosztott intelligenciájú kísérleti rendszerek esetében a feldolgozó lánc elemei

behatolnak a kísérleti elektronika területére; kísérleti-elektronika és feldolgozó lánc már-már egymástól elválaszthatatlan.

A feldolgozó lánc a kísérleti berendezés által nyújtott és a kísérleti elektronika által adatok alakjában előkészített információ begyűjtésére és analizisére szolgáló rendszer, mely alapvető feladatán túl másodlagos, így pl. ellenőrző és vezérlési funkciókat, továbbá legkülönbözőbb háttérfeladatokat is elláthat. Legalább egy számítógép tartozik hozzá; igen sokféle szervezőmódja ismeretes és lehetséges.

A kísérlet automatizálása terén jól megfigyelhető két, egymással szemben ható, de végül is egymással dialektikusan összefonódó - az erőforrások centralizációjára, ill. decentralizációjára irányuló - törekvés. A megvalósított feldolgozó láncok közül sok aszerint jött létre, hogy e törekvések közül melyik jutott túlsúlyba. Mégis, a feldolgozó láncok megvalósítása kapcsán - különösen a fejlődés korábbi szakaszában - gyakran nem elvi megfontolások, hanem az adott pillanat és hely lehetőségei domináltak.

Centralizált feldolgozó láncról akkor beszélhetünk, ha a kísérlet minden számítógépes szolgáltatását az intézet valamely közös használatu számítógépe (gépei) látja (látják) el. Ennek tipikus esetét a *10.a ábra* szemlélteti. Ez esetben az adatgyűjtés kísérletenként on-line időreális módon mágnesszalagra történik, majd a mágnesszalagot átviszik az intézet (esetleg több intézet) közös számítóközpontjába, és ott adatait kötegelt feldolgozásmódban értékelik ki. A feldolgozó lánc egy része ebben az esetben nyitott (mágnesszalag-számítóközpont).

Decentralizált feldolgozó láncról (*10.b ábra*) akkor beszélhetünk, ha az egyes kísérletekhez önálló és csakis az adott kísérlethez rendelt számítógép tartozik, amelyik az esemény-analízist és a kísérlet átfogó analizisét is elvégzi úgy, hogy további számítógépes szolgáltatás nem szükséges. Maga az, hogy a számítógép egy adott kísérlethez tartozik, s hogy a vele kapcsolatos összes számítástechnikai feladatot is ellátja, még nem jelenti egyúttal azt is, hogy minden munka on-line időreális

módon történik. Ellenkezőleg, nagyon is valószínű, hogy a gép időreális módon csak részleges analízist végez vagy csak az adatok egy részén végez analízist, az összes adat teljes analízisét pedig csak az egyes szeánszokat követően végzi el. A decentralizált feldolgozásmódnak az a bökkenője, hogy minden egyes kísérlethez lehetetlen komplex feladatok jó hatásfoku megoldásához szükséges közép-nagygépet vagy ezt helyettesíteni képes többgépes kisszámítógép-rendszert hozzárendelni, a kisgépeknek pedig még jelenlegi igen fejlett változatai sem képesek mindig megbirkózni a részecskefizika bonyolult feladatstruktúráival. Ezért az ilyen decentralizált, minden mástól független feldolgozó-lánc csak viszonylag egyszerű kísérletekhez alkalmazható.

A két szélsőséges típus mellett változatok, közöttük pedig átmeneti jellegű feldolgozó láncok találhatóak. A centralizált feldolgozó lánc egy változata látható a 10. c. ábrán, melynél egy központi gép on-line adatgyűjtést és részleges analízist is végez, háttérfeladatként pedig egyazon kísérlet vagy valamely más kísérlet teljes analízisén dolgozik.

A ma részecskefizikáját leginkább a megosztott erőforrásu és a feladatelosztásu, tehát a centralizáció és decentralizáció jegyeit egyaránt magán viselő feldolgozó lánc különböző variánsai jellemzik (11. ábra). Az ilyen típusu feldolgozó láncban nem egy, hanem esetenként egész sor számítógép található, melyek közül egyesek kizárólag az adott kísérlethez, tehát az adott feldolgozólánchoz tartoznak, mások egy egész laboratóriumot szolgálnak ki, s ismét továbbiak intézeti, sőt több intézetet kiszolgáló számítóközpont gépei lehetnek. A legfejlettebb feldolgozó láncok több intézmény számítóközpontjainak szolgáltatásait is igénybevevő nagy rendszerek. A feldolgozó lánc egyes számítógépei egymással közvetlen vagy —információhordozók (pl. mágnesszalag) révén — közvetett kapcsolatban állnak, s egy feldolgozó láncon belül mindkét kapcsolttípus is előfordulhat.

Hangsúlyozni kell, hogy a "feldolgozó lánc" nem azonos valamely meglévő számítógéphálózattal. A feldolgozó lánc az egy kísérlethez tartozó információáram utvonala, amint azt a 10. és

a 11. ábra is kiemeli.

Az imént az átmeneti típusok kettős jellegéről, a feladatelosztásról és az erőforrások megosztásáról esett szó. A kettő nem ugyanaz. Feladatelosztáson az egy kísérlethez tartozó munkának vagy feladatnak több számítógépre való szétosztását, míg erőforrás megosztáson egy számítógép erőforrásainak több kísérlet közötti szétosztását értik. Feladatelosztás több kisgépet felölelő decentralizált feldolgozó láncot is jellemezhet, míg az erőforrás megosztás inkább a centralizáltság jellemzője.

Amennyiben egy számítógépet több kísérlet egyidejűleg használ, úgy e gép többprogramu üzemmódban kell, hogy dolgozzék. Az egyidejűség nem jelent időazonosságot is, vagyis ha bármely időpontot választunk is ki, abban a központi gép processzora mindig csak egy kísérletet szolgál ki. Hosszabb időtartamra nézve mégis minden kísérlet irányából nézve úgy látszik, mintha a központi gép egyedül hozzá tartoznék. A gép erőforrásai időben megoszlanak, s így minden kísérlet egy valóságban létezőnél csökkentebb képességű számítógépet mondhat magáénak. Ez a csökkentebb adottságu de már kizárólag az adott kísérlethez tartozó számítógép áll a feldolgozó lánc végpontján.

Azokat a számítógép-rendszereket, amelyekre az egyes kísérletek helyi gépei akár laboratóriumi, akár általánosabb érvényű előírások (protokoll) szerint közvetlenül csatlakozhatnak, nevezik on-line számítógép hálózatnak. Az on-line hálózat korlátozódhat egy-egy laboratórium vagy intézet területére (helyi hálózat), de több, földrajzilag egymástól távoleső intézetet is összeköthet (12. ábra). A hálózat a kísérletek mellett általában más feladatokat is elláthat, így pl. kiszolgálhatja a hozzá csatlakozó, programelőkészítés ill. - futtatás céljára szolgáló terminálokat, és háttérmunkákat is végezhet. A számítógép-hálózat a kísérlet feldolgozó láncában virtuális számítógépként vesz részt, erőforrásainak egy részét oly módon bocsátva egy-egy kísérlet rendelkezésére, mintha egyidejűleg más szolgáltatásokat nem is végezne.

Két példa. A leirtak plasztikusabbá tétele céljából és ki egészítéseként, tekintsük át egy egyszerűbb és egy valamivel bonyolultabb kísérleti rendszer szervezőmódját.

A [8,9]-ben leirt és a 4. ábrán felvázolt kísérleti rendszer néhány részletéről az eddigiekben már szó volt, most nézzük meg egészében. A leegyszerűsített vázlaton a kísérleti berendezésnek csak a detektorai láthatók. A trigger-detektorok részecske azonosítását végző, ill. topológiai jellegű detektorok két csoportjára oszlanak el, a topológiai jellegű előszelekcióhoz azonban az esemény-detektorok jeleit is felhasználják. A kétfokozatu trigger egyes fokozatainak szerepkörére, valamint a triggerjel képzésének módjára a korábbiakban már kitértünk, most még az esemény regisztrálásának módjával, az esemény-elektronikával és az adatkezelő felépítésével és működésével kell megismerkednünk.

Az esemény detektorok együttese három — H_1 , H_2 , H_3 — szcintillációs hodoszkópból áll. H_1 hodoszkópnak 28, H_2 -nek 80, H_3 -nak 96 eleme, az egésznek tehát összesen 204 kimenete van. A detektorjelek kapuáramkörökön át gyorsműködésű regiszterre kerülnek, és ott tárolódnak, ha az eseményt a trigger elfogadja, és jele nyitja a kapuáramköröket. A gyorsregiszter szélessége 96-bit, közvetlenül tehát nem fogadhatja a 204 detektor-kimenetet; a detektor-információt kódolják. A kódolásmód a H_3 hodoszkóppal kapcsolatban szemléltetve a következő: A H_3 négy egymás mögött fekvő hodoszkópegységéből, minden egység két hodoszkóp síkból, a síkok 12-12 szcintillátor csikból állnak. Az egymást követő szomszédos síkok szcintillátor csikjai egymásra merőlegesek. Az azonos pozícióju "vizzintes" detektorok 4-4 kimenetét egyesítve, 12 vizzintes és 4x12 függőleges, azaz 96 helyett mindössze 60 kimenet marad, mely a részecske trajektóriákra vonatkozó információt sértetlenül nyújtja. Hasonló elven kódolják nemcsak H_3 hanem H_1 és H_2 kimeneteit is.

Valamely esemény elfogadását a trigger nemcsak az esemény-elektronikával, hanem az adatkezelővel is közli, mely E/T hatásvonalán át a triggerrel azonnal reteszeli. Az adatkezelő 96 bites (be- és kimenő funkciót egyaránt ellátó) regiszterből, 32 szó/96 bit kapacitású pufferből és vezérlőből áll. Az írás/olvasás bemenet a gyorsító impulzusának idejére az adatkezelőt írás üzemmódba, a gyorsító szünetében olvasás üzemmódba kapcsolja. Az utóbbi üzemmódban a trigger — éppugy, mint ahogy az egyes ada-

toknak az esemény-elektronikából az adatkezelőbe való átvitele idejére is — tiltva van.

Írás üzemmódban az adatkezelő mindig kész az eseményadat átvételére, hacsak egy előző adat átvételével nincs elfoglalva. Feltételezve, hogy az adatkezelő szabad, a triggerjel kapuzza az eseményadatot, egyuttal jelzést küld az adatkezelőnek, mely reteszeli a triggert, majd L lehvívőjele utján az adatot az esemény-elektronikából saját be/kimenő regiszterébe helyezi át, ahonnt az a puffer címmutatója által kijelölt, soronkövetkező pufferrekeszbe továbbbitódik. Ezután a trigger reteszélése feloldódik. Amennyiben a 32 pufferrekesz feltöltődnek, az adatkezelő a további triggerképzést letiltja (E/T vonal). Amikor az adatkezelő a gyorsító impulzus zárójelének hatására olvasás üzemmódra kapcsol át, megkezdődik a pufferrekeszek tartalmának a be/kimenő regiszterbe való egyenkinti visszairása, és onnan mágnesszalag tárolóra, egyuttal pedig számítógép bemenetre is való továbbbitása. Az adattovábbítás nem 96, hanem 6 bit széles byte-ok alakjában történik; a paralel-soros átalakítás a be/kimenő regiszter feladata.

Az adott kísérlet kapcsán a feldolgozó-lánc (számítógép) vezérlési feladatokat nem végez, és kísérő információt sem gyűjt. A vezérlési - beállítási funkciókat manuálisan intézik, és ugyancsak manuálisan regisztrálják a kísérő infomációt is, hogy majd ez utóbbit a számítógéppel közöljék. A számítógépes feldolgozással paralel készülő mágnesszalagos regisztrátum a kísérleti adatok archivumául szolgál, ami különösen értékes az esetben, ha netán a számítógép a kísérlet során meghibásodnék.

A 13. ábra a második példaként választott kísérlet [A4] elektronikájának szervezőmódját mutatja be. Ennek érdekessége, hogy nemcsak elő- hanem utószelektáló (szűrő) áramkörei is vannak, s hogy esemény-azonosító detektorai két csoportra oszlanak. Az első csoportba a tulajdonképpeni trigger-detektorok tartoznak, melyek kizárólag a beeső részecskék identifikálására szolgálnak, s így szerepük nagyjából megegyezik az előző példa 1. trigger-fokozatának detektoraiéval. Az esemény-azonosító detektorok 2. csoportja az utószelekcióhoz szolgáltat információt, a hozzá-

tartozó áramkörökkel együtt F4 feltételt vizsgálva. Az utószelekcióhoz a proporcionális kamrák nyújtotta információ is alapul szolgál, éspedig az F1, F2 feltételek vizsgálatához a 2. kamracsoport, az F3 vizsgálatához az 1. kamracsoport kimenete.

Az eseményregisztráció — az 1. és a 2. kamracsoport által szolgáltatott információ rögzítése — az FO triggerfeltétel teljesülésével történik. A triggerjel a programmal beállítható paraméterü PROGR. FORMÁLÓ-n áthaladva ad kapujelet az 1. és 2. sz. kamraelektronikának. Az utószelekciót az F1, F2, F3, F4 feltétel összevetésével a PROGRAMOZOTT DÖNTÉS blokk végzi. Ez a blokk az F1...F4 feltételek teljesültének igen-nem kimeneteit különböző — számítógépi program által meghatározott — kombinációkban vetheti össze. Ezenképpen mód van rá, hogy az eseményszelekció más-más, a fizikus által megválasztható feltételegyüttes alapján történjék, vagyis hogy egynél több eseménykategóriába tartozó események statisztikáit is lehessen képezni.

Egy esemény regisztrálásának, a feltételek előkészítésének és a döntésnek, vagyis az utószelekciónak együttes ideje $\sim 0,8 \mu\text{s}$. Ennek megfelelően minden eseményt követően a triggerelektronika $\sim 1 \mu\text{s}$ időre automatikusan reteszeli. Az utószelekció az eseményt elfogadja (a DÖNTÉS-blokk IGEN kimenőjelet ad) vagy elutasítja (NEM kimenőjel). Az utóbbi esetben a VISSZAÁLLÍTÓ-áramkör a már meglévő esemény-regisztrátumot mint nem kívánatos az esemény-elektronikából törli, s a trigger-elektronika reteszélése is feloldódik. Amennyiben azonban az utószelekció kimenete pozitív, úgy a DÖNTÉS-blokk IGEN kimenőjele további $\sim 2 \text{ ms}$ időre reteszeli a trigger-elektronikát, egyúttal az adatkezelő gyűjtő-elosztó sinjén szolgálatkérő jelként jelentkezik. E jel hatására az esemény-információ átvitele a számítógép irányába megindul. Az átvitel és a vele kapcsolatos program-műveletek végrehajtása $< 2 \text{ ms}$ alatt megy végbe; lezárultáról a számítógép a gyűjtő-elosztó sinhez kapcsolódó PROGR.VEZÉRLŐ útján ad értesítést, mely STOP jelével feloldja a 2 ms RETESZT-t. Egyúttal az egész kísérleti elektronika alaphelyzetébe állítódik vissza.

A paralel REGISZTER a triggerképzéssel járó, alapjában véve logikai jellegű jelek bizonyos készletének regisztrálására, míg

a számlálók a kiválasztás különböző szintjein jelentkező eseményszámok rögzítésére szolgálnak (kísérő információ).

Az F1, ill. F3 feltételek teljesültének vizsgálatát az f_1 , ill. f_3 , az F2 feltételét pedig az f_x , f_y VAGY áramkörök végzik. A 2. triggercsoport jeleit a KÜSZÖB (diszkriminátor) blokk vizsgálja; e jelek egyidejűleg töltés nagyságának értékét digitalizáló A/D átalakítóra is rákerülnek.

Az eseményinformáció lineáris pozíciókódban eseményenként adódik át a számítógépre. A 2. sz. KAMRA ELEKTRONIKA utószelekció céljára kódoló áramköröket is felölel.

A továbbiak sz szempontjából ki kell emelnünk azt a fontos körülményt, hogy az utószelekcióval elutasított események kapcsán mindössze is csak $\sim 1 \mu s$ regisztrálási holtidő adódik, és csak az elfogadott események adatkezelése igényel $\sim 2 ms$ -t, azaz idéz elő ugyanennyi holtidőt.

A display mint interaktív terminál az általános részben körvonalazott feladatokra szolgál. Többek között a fizikus rajta keresztül adhatja meg az utószelekciót végző DÖNTÉS blokknak a kiválasztás együttes feltételeit, miket azután a számítógép meghatározott program szerint időről-időre automatikusan módosíthat.

Az adott kísérleti berendezés adatkezelője nem rendelkezik pufferral, ezért az adatok eseményenkénti blokkokban kerülnek egyenesen a számítógép memóriájába. A memória pufferterelete 18 Kbyte, a számítógép ennél nagyobb mennyiségű adatot gyorsító-ciklusonként nem fogadhat.

4. A CAMAC RENDSZER

Az adatkezelő a kísérleti elektronika (ill. a forrásvezérlő) és a feldolgozó lánc közötti kommunikáció eszköze. Fő feladata, hogy az elektronikus blokkoktól adatokat vigyen át a feldolgozó láncra, s fordítva, a feldolgozó lánctól adatokat juttasson el az elektronikus blokkokra. Ebben az értelemben, adagyűjtő-szétosztó hálózatnak is tekinthető. Elvárt tulajdonsága: lehetővé tenni az elektronikus blokkok egységes csatlakoztatását, valamint a blokkok és a feldolgozó lánc egységes eljárás (protokoll) szerinti kommunikációját. A kísérleti berendezés és elektronikája egyrészt, a feldolgozó lánc másrészt autonómiával rendelkezik. A két fél az adatkezelőn át fizikai kapcsolatban áll egymással, általában nincs azonban logikai kapcsolatuk. Logikai kapcsolat az adatkezelőn át, az adatkezelő kapcsolati szabályai szerint létesülhet, akár az elektronikus blokkok szolgálatigénylő jelei, akár a feldolgozó lánctól érkező kezdeményezés alapján.

A fenti szerepkört betöltő, s a leírt tulajdonságokkal rendelkező adatkezelő rendszerek kialakítására már a hatvanas évek elején történtek erőfeszítések. Eredmények: Európában az ESONE [20], az U.S.A.-ban a NIM [21] rendszerek jöttek létre. A számítógépek előtérbe kerültével 1969-re kialakult a kifejezetten számítógépi együttműködést figyelembe vevő CAMAC szabvány első változata [22]. Ez az ESONE [20] rendszert teljesen kiszorította, a NIM szabvány alkalmazását pedig a gyors elektronika területére zsugorította, ahol azt jelenleg is használják.

Az idők folyamán az eredeti CAMAC szabvány kisebb módosításokon ment át, s emellett számos új fejezettel ki is bővült. Jelenlegi előírásait az európai (EURATOM, ESONE), az amerikai (ANSI, IEEE, DOE et TID), valamint a nemzetközi (IEC) szabványosítás különböző szintjein kodifikálták. A CAMAC, mint általában más szabványok is, sohasem elégített ki minden lehetséges követelményt, ezért már létrejött idején [23,24], de később is [25] vele párhuzamosan megjelentek egyéb javaslatok is. Mindez nem hátráltatta a CAMAC rendszer termékeny kibontakozá-

sát, és igen széleskörű elterjedését sem. Noha a részecskefizika újabb követelményei gyakran beléütköznek technikai korlátaiba, a CAMAC jelenleg, és bizonyosan még jó néhány éven át továbbra is, fontos szerepet tölt be e területen. E jelentősége indokolja, hogy ezen a helyen is megkíséreljük tömören összefoglalni leglényegesebb vonásait. Az irodalomjegyzékben néhány alapvető CAMAC közleményt idézünk [26-29], melyek a formális CAMAC leírásokra is utalnak.

4.1 CAMAC ALAPOK

Alapszabványa megfogalmazása szerint, a CAMAC: egységes fizikai és logikai csatlakozást előíró passzív, hierarchikus adatgyűjtő - szétosztó hálózat. A passzív megjelölés arra utal, hogy a hálózaton belül nincsenek CAMAC műveleteket vagy e műveletek sorából álló programokat generálni képes elemek; CAMAC műveletek egyedül a feldolgozólánchoz tartozó, a hálózaton kívülfekvő programforrástól származó parancsok alapján hajthatódnak végre.

A CAMAC szabványok két szintet határoznak meg: az adatutat és a hierarchiában felette álló főutat.

Az *adatut* max. 23 elektronikus blokk - a CAMAC szóhasználat szerint: modul - csatlakoztatását teszi lehetővé (14. ábra). Az irodalomban általában sinként ábrázolt adatut voltaképpen radiális szervezésű álsin (14.b. ábra), mivel - noha legtöbb vezetéke sinrendszerű - minden egyes modulnak egyedi címvezetéke (N) és egyedi igénylő vezetéke (L) van. Ezért is nem lehetséges modulok közötti közvetlen kommunikáció az adatuton. Az adatut a szabványos méretű keret háthuzalozásaként realizálódik, és az egyes modulok számára szabványos csatlakozón át hozzáférhető. A modulok mérete a keret méreteihez igazodva ugyancsak szabványos: mélység és magasságméretük kötött, szélességük pedig egy előirt egységmértet egészszámu többszöröse. Az egységnyi modulszélesség a keret teljes szélességének 1/25-öde. Az adatut vezetékei a keretvezérlőből ágaznak ki, ill. hozzá futnak be. A szabványos keretvezérlők két modulszéles-

ségük, ennek megfelelően a keretben max. 23 modul foglalhat helyet. Nem szabványos szélességű keretvezérlő vagy a keretvezérlőhöz kapcsolódó járulékos egységek használata esetén a modulok rendelkezésére álló hely csökken. Az elhelyezhető modulok száma 23-nál kevesebb lesz akkor is, ha a modulok között egységnyinél szélesebbek is előfordulnak. A max. 23 modul egyenkénti megszólítására keretenként 23 cím, ezenkívül modulcimenként 16 alcím áll rendelkezésre. A modulcímet, vagy helyesebb szóhasználattal pozíciószámot a CAMAC irodalom N-el, az alcímet A-val jelöli. Kettős vagy többszörös szélességű modulokhoz egynél több pozíciószám és pozíciószámonként 16 alcím tartozik. A modulok (pozíciók) maximális számának megfelelően keretenként 23 igénylő vezeték van, pozícióként azonban több igénylő is lehetséges, melyek egyidejű igénylőjele (LAM) az adott pozíció L vezetékére nézve VAGY kapcsolatban jelentkezik.

Az adatutatót a keretvezérlővel és a keretvezérlő jobb oldalára csatlakozó (s a későbbiekben ismereteketendő) párhuzamos főúttal együtt a 15. ábra szemlélteti. Az adatutató vonalai négy csoportba sorolhatók. Ezek: az adatvonalak, a parancsvonalak, a vezérlő vonalak és az igénylő vonalak.

Az adatutatónak két, egyenként 24 bit szélességű egyirányú adatvonala van, az egyik olvasás, a másik írás céljára. Az olvasás/írás kifejezés itt az adatutatóhoz csatlakozó modulokra vonatkozik. A 24 bitnyi adatszélesség lehetőség, de nem kötelező; 1-24 bit között bármely adatformátum választható. Az adatátvitel szinkron, a szinkronizálásra S1, S2 vezeték (mikor melyik) jele szolgál. A maximális átviteli sebesség 1 MHz, azaz 3 Mbyte/s.

Az adatutatón pontosan meghatározott műveletek hajthatóak végre, a keretvezérlő bemenetére érkező parancsok alapján. A műveletek lebonyolítására az N, A, F, Z, és C vezeték szolgál. Az adatutatón közvetlen és közvetett parancsok különböztethetők meg.

A közvetlen parancsok végrehajtása az N, A, F vezetékkel történik. Az NAF formátumu közvetlen parancsokat a keretvezérlő külső oldaláról nézve (formailag) olyan N(i) címresz jellem-

zi, melynél $1 \leq i \leq 23$, tehát amelynek N része a 23 pozícióvezeték egyikét jelöli ki. E parancsok számára a keretvezérlő transzparens: a binárisan kódolt N értéke egy pozícióvezetékre képződik le, A és F pedig változtatás nélkül kifut az A és F vezetékre. Az NAF parancs 14 bit szélességű, ezen belül N 5 bit. A 4 bit, F 5 bit helyet foglal el. Az 5 bit által megengedett, összesen 32 feladatkód közül az adatpályára 18 van értelmezve (16. ábra), 6-ot a szabvány további fejlesztés céljára tartálékol, a maradék tetszés szerint alkalmazható.

A közvetett parancsokat az jellemzi, hogy $N(i)$ címreszük nem modul, hanem a keretvezérlő egy helyét címezi meg, $24 \leq i \leq 31$ értékkel (az $i=0$ fenntartott cím). Ide tartoznak az egyidejűleg több pozíciónak szóló, valamint a Z és C vezetéket működtető parancsok. A szabványos $CCA-1$ keretvezérlő esetében pl. az $N(26)A(j)F(k)$ parancs az $F(k)$ adatut-műveletet az összes N pozíció $A(j)$ alcimén egyidejűleg végrehajtja, az $N(24)A(j)F(k)$ parancs pedig preszelektált modulokon hajt végre egyidejű műveletet. Az utóbbi műveletben érintett modulok kijelölése egy bináris vektor bitjeivel történik, mely vektort a parancs végrehajtása előtt kell a keretvezérlő egy erre a célra szolgáló regiszterében elhelyezni. A Z vezetéket, melynek jelével a keret összes áramkörei meghatározott kezdőhelyzetbe állíthatók az $N(28)A(8)F(26)$, a csupán bizonyos - a tervező által előre meghatározott - modulokhoz, ill. áramkörökhöz szóló C törlő vezetéket pedig az $N(28)A(9)F(26)$ parancs működteti. Közvetett parancsokkal működtethető továbbá az I állapotvezeték, mely átállított állapotában azokra az áramkörökre, amelyekre nézve értelmezve van, tiltó jellel hat, alaphelyzetében viszont hatástalan. E vezeték átállítására $N(30)A(9)F(26)$, visszaállítására $N(30)A(9)F(24)$ parancs szolgál.

Az adatut közvetett parancsaival formálisan megegyeznek a keretvezérlő belső parancsai, melyek az adatpályára közvetlen hatást nem gyakorolnak. Ezek címrésze a $CCA-1$ esetén mindig $N(28)$ vagy $N(30)$. Ide tartoznak a keretigényt tiltó, ill. engedélyező, valamint az állapotvizsgáló parancsok, továbbá a

keretvezérlő egyes belső regisztereit megcímző írás - olvasás parancsok.

Külön ki kell emelni, hogy a közvetett, ill. a keretvezérlő belső parancsok formailag keretvezérlőnként változhatnak; a [32] közleményben leírt, az adatutató közvetlen számítógéphez csatlakoztató keretvezérlő esetében pl. az összes közvetett és belső parancsot N(24) cím fogja össze és az A alcím diszkriminálja.

Az adatutató-parancsok végrehajtásának idejére az adatutató a keretvezérlő a B vezetékekkel foglalja le, mely egyúttal letiltja a modulok esetleges igénylő jeleit is. A modul az X vezeték jelével igazolja vissza a parancs elfogadását. Az X jel hiánya a parancs végre nem hajthatóságát jelzi.

A Q vonal egyes CAMAC műveletekkel kapcsolatos visszajelentő vonal. Elsősorban adatok blokkban történő átvitele, a modulok állapotvizsgálata, valamint szolgálatigénylő források azonosítása kapcsán alkalmazzák.

A 23 L vezeték az egyes modulpozíciók L-kimenetén megjelenő igényjeleket (LAM) szállítja a keretvezérlőre. A CCA-1 keretvezérlőben e vezetékek egyenesen egy a keretvezérlő hátlapján lévő csatlakozóra futnak, majd onnan egy ún. igényosztályozóra kerülnek, s megfelelő újrendezés után ugyanazon csatlakozón át vissza a keretvezérlőre. Legegyszerűbb esetben az igényosztályozó a bemenő L-vezetékeket változtatás nélkül képezi le kimenetére, másrészt bizonyos prioritási előírásnak megfelelően a bemenő sorrendhez képest megváltoztathatja a kimenő sorrendet, de sok más lehetőség is van, az adott helyzetnek megfelelően. Ha pl. egyvalamely keretben az igénylő források száma csak 5, akkor az ezeknek megfelelő vezetékek prioritási sorrendben egymás mellé rendezhetők, máskor pedig az a követelmény, hogy az igényosztályozó számos L-pont jelét logikai áramkörei segítségével az igénylő pontok számánál lényegesen kevesebb bitbe egyesítse. Az igényosztályozó fogadhat és a keretvezérlőbe továbbíthat külső forrástól származó igényjel is. A keretvezérlő egyrészt az összes igénylő jelét VAGY kapcsolatba foglalja, miáltal lehetősége nyílik arra, hogy az

adatfeldolgozó láncot bárhonnán jövő igény beérkeztéről értesítse, másrészt rendelkezik azzal a lehetőséggel is, hogy az igényeket összefoglaló bináris vektort, az igényosztályozó által összeállított alakban, az adatfeldolgozó láncra továbbítsa, forrásazonosítás céljából. Esetenként az igényosztályozó a keretben egységnyi szélességű modulként a vezérlő mellé társul. A keret összesítő igényjele programozás útján maszkolható (felüggeszthető) vagy engedélyezhető, igényvonala és maszk állapota ugyancsak programozás útján megvizsgálható.

Rá kell még mutatni az adatut-parancsok (ide számítva a keretvezérlőnek szólókat is) némely, eddig még nem említett sajátosságára, mindenekelőtt arra, hogy a parancsok NAF együttesként értelmezendők. Erre azért van szükség, mivel F ciminformációt NA pedig feladatinformációt is hordoz. A 16. ábrán látható, hogy néhány műveletnek két - 1., ill. 2. regisztercsoportra vonatkozó-feladatkódja van. E feladatkódok különbözősége nem érinti magát a műveletet, csupán csak A-t. Az A-nak kétszer 16 cimből álló tartománya van és, hogy a parancsban szereplő cím melyik tartományba esik, azt F értéke dönti el. Máskor N értéke döntheti el F jelentését. Pl. az F(26) kódot közvetlen parancsban ($N \leq 23$) alkalmazva, értelme: "engedélyezés", míg ugyanez a kód CCA-1 vezérlő esetében az N(28) A(8) cím mellett kezdőhelyzetbe állítást jelent. A NAF formula tágabb értelemben is újjáértelmezhető, ha a szükség ezt kívánja. Ez esetben általában ki kell lépni a CAMAC szabvány keretei közül. Ha pl. egy modul kezeléséhez kevés művelet, de sok cím szükséges, úgy az F kód néhány bitjét el lehet tulajdonítani és A-hoz lehet kapcsolni. A cimmező - olvasás-műveletek kapcsán - még a W-vonalak felhasználásával is kibővíthető. A NAF parancsok átértelmezése azonban programozási szempontból nem kívánatos, s ezért azt általában kerülni kell.

Főutak. A több adatutat felölelő rendszerben a keretvezérlők külső oldalukkal gyűjtősínre kapcsolódnak. Gyűjtősínként szolgálhat a feldolgozólánc CAMAC oldali számítógépének B/K sinje (17.a. ábra) vagy a CAMAC szabványban meghatározott főutak egyike. A CAMAC kétféle, párhuzamos (17.b. ábra) és soros

(17.c. ábra) főutat definiál. A párhuzamos főut ág-, a soros főut hurokszervezésű. A párhuzamos főut ágvezérlőn, a soros főut hurokvezérlőn át kapcsolódik a feldolgozó lánchoz, illetve a feldolgozó lánc azon számítógépéhez, amelyik CAMAC programforrásul is szolgál. Az adatutat az ággal összekötő, CAMAC szabvány szerinti keretvezérlőt CCA-1 jellel, a soros főutal összekötőt pedig SCC-L jellel jelölik. Az utóbbinak egymástól csekély mértékben eltérő két változata: SCC-L1 és SCC-L2. A számítógéphez közvetlenül csatlakozó keretvezérlőket U vezérlőknek nevezik. Az U vezérlőknek csak adatuti oldala szabványos, a külső oldal számítógépenként (és tervezőként) más és más.

A párhuzamos főut (melynek megnevezésére az "ág" szinonimát is használják) szerkezete a 15. ábrán látható. Vonalai - ugyanugy, mint az adatutái is - adat-, vezérlő-, parancs- és igényvonalak csoportjába sorolhatók.

Egyetlen kétirányú 24 bit szélességű adatvonala van, melyen az átvitel a BTA és BTB kapcsoló vonalak révén aszinkron módon történik. Mivel az adott aszinkron parola (handshake) üzemmódban egy átvitel a küldő és a fogadó közötti négyeszeri kölcsönhatást igényel, már viszonylag nem túl hosszú ág-kábelezés esetén is, egy adat átvitele néhány μ s-t tehet ki, a hosszabb kábelek késleltetési hatásáról már nem is szólván. A párhuzamos főutat akkor kell alkalmazni, ha nagy átviteli sebesség szükséges, ezért egyúttal minimális kábelhosszra is kell törekedni; az adatut által megengedett 1 MHz sebesség az ágon azonban még így is csak kivételes körülmények között érhető el.

Az ágra max. 7 keret (7 keretvezérlő) csatlakozhat. Mivel bármely parancs végrehajtásakor azt valamelyik konkrét keret-höz kell irányítani, az ágra irányuló parancsok ki kell, hogy egészüljenek egy 3 bites keretcimmal. A parancs ilyenképpen 14-ről 17 bit szélességűre bővül, s alakja: CNAF. A C-jel az eredeti angol szó (crate=keret) rövidítése. A C dekódolása nem a keret-, hanem az ágvezérlőben történik, az értékének megfelelően BCR sugárvezeték kijelölésével, éppugy, mint ahogy N dekódolása a keretvezérlőn belül megy végbe.

Az ág minden vonalát a "Branch=ág"-ra való utalásul B betű jelöli meg. Erre azért van szükség, mivel az ágnak néhány vonala (igy a BA, BF, BZ, BX, BQ) az adatut megfelelő vonalával kongruens, különbözésük és egymásnak megfelelésük feltüntetése tehát egyaránt szükséges. Ami a kongruens vezetékeket illeti, világos, hogy ezek szerepe azonos, mint az adatut megfelelő vezetékéié. Az előzőekben láthattuk, hogy a Z, C és I vonalak közvetett parancsokkal kezelhetők. BZ vonal útján azonban a nullázás művelete közvetlenül is elvégezhető. A kétféle nullázás szerepköre különböző. Míg a BZ vonallal az egész ág minden egysége nullázható, az egy kerethez szóló nullázó parancs érvénye csak arra a bizonyos keretre terjed ki. Az már más kérdés, hogy az ágvezérlő külső oldaláról a Z vonal ismét csak - ágvezérlőnek címzett - közvetett paranccsal működtethető.

Többágu hálózatok. Számos részecskefizikai kísérlet esetében hét keret nem elegendő a kísérleti elektronika blokkjainak csatlakoztatására, ill. befogadására. A CAMAC szabványokban többágu hálózatok nem szerepelnek, ám egyes intézetek saját céljaikra kidolgoztak különböző olyan rendszervezérlőket, amelyekhez egynél több ág is csatlakoztatható. A rendszervezérlő legegyszerűbb megoldásmódja, melyhez néhány ág, de csak egy számítógép kapcsolható, a 18. ábrán szemlélhető. Az ágvezérlő ágoldali és számítógéppoldali része itt különválnak, és az ágoldali rész (ÁH) az ágak számának megfelelően multiplifikálódik, a rendszer-vezérlő (R-vezérlő) pedig a keret sinrendszerével együtt a multiplexer-demultiplexer szerepkörét is betölti. Maga a keret a CAMAC szabvánnyal oly mértékben egybeváogó, hogy ágvezérlőkkel nem foglalt helyire CAMAC modulok (SCM) dugaszolhatóak. Az R-vezérlő ez utóbbiakkal U-vezérlő módjára kommunikál.

Többágu hálózatban a programforrásnak BrCNAF alakú parancsokat kell generálnia, ahol új elem a Br ágcím (Branch address). Bár a (legfeljebb 2-3 bitnyi) Br kód az N pozícióvezetékekre leképezhető, a rendszerkeret azonban a CN komponens átvitelére szolgáló vezetékekkel nem rendelkezik. Az ebből következő nehézséget a keret kétállapotúsága révén kerülnek el.

Egyik üzemi állapotában a keret normális adatutként szolgál, másik üzemállapotában pedig az ágakat kezelő speciális sinrendszerként. Egyes esetekben [30] a kétállapotúságot az eredeti kerethuzalozás módosítása nélkül, míg máskor az adatut járulékos sinnel való kibővítése révén [31] oldották meg. A 18. ábra járulékos sinnel rendelkező rendszerkeretet mutat be. Vannak végül olyan rendszervezérlők is, amelyeknek mechanikai felépítésmódja a CAMAC-tól eltér [46].

Igénykezelés a párhuzamos hálózatban. Az adatutra csatlakozó modulok meghatározott szolgálati igényvel léphetnek fel (LAM jel), mely igényt a CAMAC hálózat előírászerűen továbbítja a feldolgozó lánchoz. Minden egységnyi szélességű modulnak egy igény-kimenete van (L) (a többszörös szélességűnek több), mely sugárvezetékével a már leírt módon csatlakozik a keretvezérlőre. Az igénykezelésnek három fő fázisa van:

1. Az igénynek a feldolgozó láncra való továbbítása, s az utóbbi általi észlelése.
2. Az igénylő azonosítása.
3. Az igényelt szolgálat elvégzése.

A CAMAC séma a modulon belüli összes igénylő jelét L kimenetére, a keret összes L-jelét pedig a keretvezérlő BD (Branch Demand) kimenetére vonatkozóan VAGY kapcsolatban egyesíti. Huzalozott VAGY kapcsolatban állnak továbbá az ágra kapcsolódó keretek BD kimenetei is. Végül, ha több ág van, ezek igényjelei is VAGY kapcsolatban egyesülhetnek, bár esetenként az adott számítógép megszakitási rendszere más lehetőségeket is megenged. A VAGY kapcsolat ezuttal annyit jelent, hogy a mögötte lévő hálózat egyidejűleg jelentkező igénylőitől (függetlenül ezek számától) csupán egy (közös) igénylőjel továbbítódik.

Igény észlelésekor, a "VAGY séma" használata folytán, nem áll rendelkezésre információ az igénylő (v. igénylők) identitásáról, azt tehát valami módon azonosítani kell.

Egy ág esetén, az azonosítás kezdőlépéseként, a BD jelre válaszul, a feldolgozólánc az összes keretvezérlőnek szóló BG-jel aktivizálja. Ennek eredményeképpen az ágvezérlőre BRW-n át az összes kerettől származó 24 bites bináris LAM vektor érkezik.

Feltételezve, hogy a potenciális igénylők száma ≤ 24 , valameny-nyi igénylőhöz e vektor egy-egy bitje rendelhető, éspedig a prioritási hierarchiának megfelelő sorrendben. E sorrendben végigvizsgálva a biteket, a feldolgozólánc azonosíthatja a legnagyobb prioritású igénylőt, majd rátérhet az ez által igényelt szolgálati program elvégzésére; ezt követően a következőre, és így tovább. Ha a potenciális igénylők száma > 24 , a bináris vektort fel kell osztani az egyes keretek között. Ilyenkor legalább egy keretnek bizonyosan kevesebb bit jut, mint potenciális igénylőinek száma. Bizonyos stratégiai megfontolás szükséges tehát annak érdekében, hogy az egyes kereteknek jutó zónákba mi módon kódoltassék az igénylő információ. Az igénylők azonosítása azután e stratégia által eleve meghatározott, s a prioritási rendet is magában foglaló séma alapján történik. Az IGÉNYVIZSGÁLAT művelettel (l. 16. ábra) végigkérdezhetőek nem csak az egyes modulok, hanem a modulokon belüli potenciális igénylők is. A lekérdezgetésre az aktiv (és nem maszkolt) igénylők a Q vezeték útján pozitív választ adnak. Az egy modulon belüli igénylők egyidejűleg is lekérdezhetőek, ekkor azonban válaszként nem Q, hanem a lehetséges igénylők számának megfelelő, de max. 24 bites bináris vektor adódik BRW vonalon át. Az aktiv igénylőket e vektor bináris 1 helyei jelölik.

Az igények minden szinten maszkolhatóak. A maszkolás mikéntjének szervesen bele kell illeszkednie a forrásazonosítás egészének stratégiájába, bizonyos feladatok végrehajtása során időszakosan felfüggesztve azoknak az igénylőknek az aktivitását, amelyek kiszolgálása valamely adott feladat végrehajtása során elodázható.

Az igénykezeléssel kapcsolatos műveletek a 16. ábrán felsoroltak közül az

- igényvizsgálat
- igénytörlés
- felfüggesztés (1 bites maszkolás)
- engedélyezés
- szelektív írás (maszkolás)
- szelektív törlés

- állapotvizsgálat
- 2. RCS olvasás,

és pedig az első négy kizárólagosan.

Rendszerkeret alkalmazása esetén az igénykezelés mechaniz-
musa még egy további hierarchikus szinttel bővül. Az újabb
szint az ághajtók és az R-vezérlő adottságaitól függő egyedi
módszerekkel élhet, de az alsóbb szinteken alkalmazott egysé-
gesített eljárások érintése nélkül. PDP-11-hez csatlakozó R-ve-
zérlő esetén pl. elképzelhető olyan megoldás, hogy az igénylő
ág már eleve azonosítójával jelentkezik (vektorális megszakí-
tórendszer); ez esetben az igénylő ág külön azonosítására
nincs szükség.

A soros főút: zárt hurok, bitsoros vagy byte-soros szink-
ron átvitel móddal. A főúton az adattovábbítás üzenetek alakjá-
ban történik. A bitsoros főút kétvezetékű (egy adat, egy szink-
ron vezeték), a byte-soros pedig kilencvezetékű (nyolc adat,
egy szinkron vezeték). Az átvitel módtól függetlenül, az üzene-
tek mindig byte-szervezésűek. Háromféle üzenet van:

- a hurokvezérlő által kibocsátott parancsüzenet, mely i-
rás művelet alkalmával a CAMAC parancson kívül a beiran-
dó adatot is magával viszi,
- a keretvezérlő által a parancsra adott válaszüzenet,
mely olvasás műveletkor az olvasott adatot is magával
hozza, és
- a keretvezérlő által kibocsátott igény-üzenet.

Az üzenetek a hurkon mindig egyirányba haladnak.

A soros főút maximális átviteli sebessége 5 MHz, azaz a
bitsoros változatnál 5 Mbit/s, a byte-sorosnál 5 Mbyte/s. Te-
kintettel arra, hogy minden 24 bites szó átvitele az adatutató
a hurokvezérlő felé vagy fordítva 12 byte kezelését igényli,
s figyelembe véve az adatutató 1 μ s ciklusidejét is, egy átvite-
li ciklus ideje a keretvezérlőnél bitsoros átvitel esetén 25
 μ s, byte-sorosnál pedig 3,5 μ s. Az adatok egyenkénti átvitele-
kor ezen értékhez még a hurok (kábel plusz keretek) késlelte-
tése is hozzájárul. Világos, hogy a byte-soros főutat költsé-
gessége miatt csak 5 MHz közelébe eső átviteli sebességnél ér-

demes használni, tehát ott, ahol nagy áthidalandó távolság mellett is még viszonylag jelentékeny adatárammal kell számolni. A bitsoros főút az olcsó, csekély adatáramu nagytávolságu összeköttetés eszköze.

A soros főút üzemét fenyegető sajátos veszély: ha bármelyik keretvezérlő meghibásodik, az üzenetlánc megszakad, hiszen a hurok valamennyi keretvezérlőn áthalad. A hurok működőképességének megőrzése, ill. gyors helyreállítása érdekében különböző módszereket dolgoztak ki [32].

Adott esetben a soros főútnak a párhuzamoséhoz viszonyított hátrányait némely előnyös tulajdonsága kompenzálhatja: Míg a párhuzamos főút mint átviteli csatorna hibavédelemmel nem rendelkezik, a soros főúton az egyes üzenetelemeket paritásbit, az egész üzenetet pedig ellenőrző végösszeg egészíti ki. Ezért a soros vonal zajos környezetben is eredményesen alkalmazható, átviteli távolsága pedig szinte korlátlan. Átviteli közegként használható egyedi telefonvonal vagy kábel, közületi telefonvonal (MODEM-mel is), fényút (közvetlen vagy szálvezetős), lézersugár és rádióhullám, végül egyetlen rendszeren belül ezek együttese is.

A soros főútnak további előnye, hogy hozzá jelentős számú keret (max. 62) kapcsolható.

Igénykezelés a soros főúton. Az SCC-L típusu keretvezérlő az igényt a feldolgozó láncsal a főútra kibocsátott igény-üzenet révén tudathatja. Az igény-üzenet nem pusztán jelzés; 5 bit információt is magával visz, melyet a keretvezérlőhöz társított igényosztályozó állít össze, az igénykezelés előre lefektetett stratégiájának megfelelően. Mivel minden egyes keretvezérlő kibocsáthat igénylő üzenetet, az említett 5 bitnyi információ mindig csak arra a keretre jellemző, amelyik kibocsátja. Tekintve, hogy mindig van valamelyes valószínűsége annak, hogy a hurokra kimenő üzenet elveszik, ezért a soros keretvezérlő rendelkezik azzal a képességgel is, hogy speciális igény-üzenettel jelezze, ha ki nem elégített igénye áll fenn.

Akár soros, akár párhuzamos főútra csatlakozik is a keretvezérlő, az adatutról nézve ez nem érzékelhető, legfeljebb az

átviteli sebesség különbözőségében tükröződik. Az adatuti műveletek tehát soros főut, soros keretvezérlő esetében is ugyan-
ugy futnak le, mint a párhuzamos főut esetében.

4.2 CAMAC VEZÉRLŐK

A CAMAC vezérlőknek számos változata ismeretes. A hálózat természetének megfelelően párhuzamos- és soros-, a hálózat szintjének megfelelően keret-, ág-, ill. hurok- és rendszer-
vezérlőket különböztethetünk meg.

Az eredeti CAMAC előírások passzív vezérlőket definiálnak, melyek maguk nem generálnak CAMAC parancsokat, csupán továbbit-
ják, ill. végrehajtják azokat.

A fejlődés magával hozta az autonóm vezérlők kialakulását. Ezek sorában korlátozott és teljes autonómiával rendelkező e-
gyedeket találunk. Az autonóm vezérlő önmaga is képes CAMAC programok generálására. Legfejlettebb változatai, a mikroszá-
mitógép szintjén, a feldolgozó láncból mintegy független dolgoz-
nak, azok pedig, amelyeknek autonómiája korlátozott, csak a feldolgozó lánc-
tal együttműködve tudnak programokat generálni.

Vannak egycsatornás és többcsatornás vezérlők. A többcsa-
tornás vezérlőket az egycsatornásokkal szemben az jellemzi, hogy egymástól kvazi független adatcsatornákat multiplikálnak, vagy-
is e csatornákon át szimultán adatátvitelt végezhetnek.

Az alábbiakban a számos variáns és átmeneti vezérlőtípus közül a legalapvetőbbeket és a legjellegzetesebbeket ismertet-
jük.

Passzív CAMAC vezérlők. A passzív CAMAC vezérlő önállóan nem tud CAMAC utasításokat (s így CAMAC programokat sem) gene-
rálni; egyetlen feladata, hogy a bementére valamely külső for-
rásból érkező parancsokat értelmezze, s hogy a hálózati hier-
archia alatta álló szintjére leképezze. A keretvezérlő eseté-
ben a parancsok leképezése nem egyszerűen a parancsok informá-
ciótartalmának továbbításából áll, (mint pl. az ágvezérlő ese-
tében), hanem egyúttal a parancsoknak a modulokkal együttműkö-
dve való végrehajtásából is.

A CCA-1 típusu vezérlő a párhuzamos főutat és az adatutat összekötő, CAMAC előírásoknak megfelelő passzív vezérlő. Címfelismerő áramkörre nincs szüksége, mivel az ágon lévő 7 keret közül mindegyiknek külön címvonala (BCR) van. Az ágvezérlő e címvonalak egyikével jelöli ki, melyik kerethez szól az általa az ágra küldött CAMAC parancs (ill. adat).

A keretvezérlőnek a hozzá címzett parancsok értelmezésére szolgáló áramkörei különbséget tesznek a közvetlen, a közvetett és a belső (a keretvezérlőnek szóló) parancsok között, s a parancs természetének megfelelően bonyolítják le a CAMAC műveleteket. Dekódoló szolgál N, A és F kifejtésére. Közvetlen parancsok végrehajtásakor az A és F kódolt alakban fut ki az adatutara, közvetett és belső parancsok esetén azonban mindkét parancselemet a keretvezérlő kell, hogy dekódolja. A CAMAC műveletek de facto végrehajtása az adatuton a vezérlő S_1 , S_2 szinkronizáló impulzusainak irányításával történik, a B foglaltsági jel kíséretében. Közbenső adattároló regiszterek nincsenek; az adatvonalak pusztán kapuáramkörökön futnak át, és nem tekintve arra, vajon az A és F dekódolását a keretvezérlő vagy valamely CAMAC modul végzi-e, az A és F vonal is tároló elemeket nem tartalmazva halad át a vezérlőn. Egy adatregiszter mégis van, az $N(24) A(j) F(k)$ közvetett parancs végrehajtása kapcsán a processzált N pozíciók kijelölésére. A vezérlőnek vannak még állapotbeállító (I) és maszkolást végző (LAM maszk) bistabiljai, valamint a LAM jelek összefogására, az osztályozott LAM vektor lehívására, továbbá kapuzására, stb. szolgáló logikai áramkörei is.

A párhuzamos főút az adatutal meglehetősen kongruens, így a főuton érkező parancsokat a CCA-1 vezérlő könnyedén képezi le az adatutra, az igényeket egyszerű módon továbbítja az adatutról a főutra; felépítése így viszonylag egyszerű.

U-vezérlők. Az U-vezérlők felépítése és működése a CCA-1 vezérlőénél lényegesen nehézkesebb, mivel a CAMAC hálózatot vezérlő számítógépek és ezek periferikus készülékei közötti kommunikáció szabályai a CAMAC protokolltól teljesen eltérnek. Eltérnek a sinstrukturák, általában eltér az átvitel tárgyát ké-

pező adatok szélessége, továbbá az igény-érzékelés és -azonosítás módja is.

Bár nem kizárólagosan, ám az esetek többségében: a CAMAC hálózatot közvetlenül vezérlő programforrás valamilyen kisseámítógép. A kisgépek B/K strukturája sokféle, mégis közülük két jellegzetes típus emelhető ki. Az első típust a külön B/K-sin, és az utasításkészlet külön B/K műveletei jellemzik, (l. pl. PDP-8). A másik csoportba az egysínű gépek tartoznak (pl. PDP-11); ezeknek külön B/K utasításaik nincsenek, a periférikus készülékek a memória címtartománya egy erre rendelt szektorának címeivel azonosíthatók, egyébként (elvileg) az utasításkészlet minden utasításával kezelhetők. A két géptípus a megszakításkérés (amelyet CAMAC vonatkozásban az "igény" szinonimával jelölünk) kezelése szempontjából is jellegzetesen eltér egymástól. Az egyik oldalon az egyetlen szerkezeti megszakítási szintű, az igénylő azonosítását programozási eszközökkel végző egyszerűbb megoldás, a másik oldalon a vektoriális módszer áll. Az utóbbi esetében a számítógép az elsőbbség megállapítását követően az igényt forrás-azonosítójával együtt akceptálja. Az igénykezelés e két szélsőséges módja között és mellett átmeneti megoldások sokasága ill. variánsok is találhatóak.

A számítógép programkapcsolatu csatornájára csatlakozó keret periférikus készülékként tekintendő, melyre a számítógép irányából nézve az adott gép előírásai mérvadók. A CAMAC és a számítógépi világ közötti különbözőség a kommunikációs előírásokéinál mélyebben fekvő; a számítógép utasításai formájukban, tartalmukban és feladatukban teljesen eltérnek a CAMAC parancskéitől. Nem lehet tehát olyan számítógépi programokat elképzelni, amelyeket a passzív U-vezérlő közvetlenül CAMAC programokra tudnak leképezni. Általában különböznek a számítógép és a CAMAC hálózat adatformátumai is.

A legkézenfekvőbb megoldás: az előirt CAMAC algoritmusok generálásához szükséges CAMAC parancskészletek tárolása a memóriában, a CAMAC adatok fogadásához, ill. kiadásához szükséges pufferverületek, ill. a feldolgozásukhoz szükséges munka-

területek kijelölése, majd annak a számítógépi utasításokból álló programnak az összeállítása és tárolása, amelyik az előkészített CAMAC utasítások és adatok alapján az előírt CAMAC algoritmusokat generálja.

Miként történhet ennek a sémának az alapján nagy vonalakban pl. egy 24 bites adat kivitele a számítógépből az adatuttra? Olyan számítógépi utasításokkal, amelyek mindenekelőtt a szóban forgó adatot a számítógépből a keretvezérlő e célra rendelt regiszterébe, majd a megfelelő NAF írás utasítást a memóriából a keretvezérlő programregiszterébe töltik át, ennek utolsó elemeivel aktivizálva az előírt műveletet. A CAMAC ciklust állapotellenőrzés követi.

A CCA-1 vezérlővel szemben, itt a közbenső tárolóregiszterek megszorodása, a számítógép-program általi előkészítő és záró operációk hosszadalmassága, az egy CAMAC műveletre eső számítógépi utasítások sokasága feltűnő, ez utóbbi különösen írás-olvasás műveletek végrehajtása kapcsán. A helyzet egyrészt a gépi utasítások, ill. a CAMAC parancsok, a gépi műveletek és a CAMAC műveletek, a gépi adatformátumok, ill. a CAMAC adatformátumok lehető minél nagyobb átfedésével enyhíthető. Meg kell jegyezni, hogy a PDP-11 és a hozzá hasonló gépek esetében a kifogásolt problémák eleve sokkal enyhébb formában jelentkeznek, mint a rövidszavas PDP-8-nál, ezért a tervezői fogásokkal elérhető javulás is sokkal kisebb mértékű lehet, mint az utóbbi géptípusnál. A PDP-8-nál eredményesen alkalmazott és jelentős javulást hozó tervezői megoldások alkalmazása a PDP-11 és más hasonló jellegű gépek esetében nemcsak eredménytelen, de esetenként kártokozó is lehet.

Tekintsük át az itt felmerülő kérdéseket részletesebben egy PDP-8 jellegű számítógép és az adatut egy lehetséges összeköttetésmódjának ismertetéséből kiindulva.

A 19. ábra az 1001 TPA-i vázlatos felépítését (a), a gép B/K utasításainak formátumát (b), a B/K utasítások CAMAC értelmezését (c), a CAMAC parancsok címrezésének a gép 12 bit szóméretéhez való illesztését (d), valamint a vezérlő - állapot-regiszter tartalmának értelmezését (e) szemlélteti. A 12 bit

széles B/K utasítások (b. ábra) három zónára oszlanak: Balról az első a 3 bites utasításkód (6_8), a következő 6 bit (két oktális számjegy) a periférikus készülék címe, az utolsó három bit (egy oktális számjegy) az adott esetben végrehajtandó műveletet jelöli ki. Adatátvitellel járó B/K műveletek esetén mindig a periférikus készülék adatregisztere és a számítógép akkumulátora a végpontok. Amennyiben tehát a periférikus készülékhez kell kivinni adatot, azt előzőleg adatmozgató utasítással az akkumulátorba kell tölteni, s fordítva, a periférikus készüléktől az akkumulátorba érkező adatot onnét külön utasítással kell a memóriába tárolni. Meg kell jegyezni továbbá, hogy a B/K utasítás lehívásakor az utasításkód a gép utasításregiszterébe, a cím és a műveletkód pedig a memória adatregiszterébe kerül.

A CAMAC szükségleteinek megfelelően, a számítógép eredeti B/K utasításait újra kell értelmezni, egyuttal a keretvezérlőt a periférikus készülékek eredeti csatlakoztatási szabálytól eltérő módon kell illeszteni. A CAMAC parancs két részre válik; a CNA címet a keretvezérlő a számítógép akkumulátorán át (d. ábra), az F kódot a B/K utasítás (c. ábra) részeként a memória adatregiszterén át kapja meg. Az 5 bites F kód a B/K utasításban csak úgy helyezhető el, ha a címrészből egy oktális számot levágunk (a 6. pozíción lévő bit mindig 0 értékű). A maradék címzóna tartalma ez esetben nem készülék cím (hiszen azt C jelzi), hanem a számítógéphez csatlakozó összes keret csoportcím.

A 20. ábra folyamatábrái szemléltetik a CAMAC vezérlés jellegű, adatbeviteli, valamint adatkiviteli parancsok végrehajtásának számítógép-CAMAC együttes eljárásait. Megjegyezzük, hogy a B/K műveletek CAMAC módja mindig CAMAC műveleteket indít, de e műveletek előkészítése és utólagos lezárása az eredeti PDP-8 B/K utasításokkal történik. Ezekkel kell feltölteni ill. kiolvasni a keretvezérlő regisztereit, végezni a keret-igények maszkolását, ill. engedélyezését. A folyamatábrák a preszelektált modulokhoz szóló közvetett címzés mód lehetőségét a lényeg jobb áttekinthetősége érdekében figyelmen kívül hagyják. A CAMAC parancsok normális végrehajtása mindig az X

vonallal visszaigazolójával jár együtt. Ha ez a jel elmarad, valami hiba van. Ezért minden parancs végrehajtásakor X-t ellenőrizni kell. Az ellenőrzés történhet automatikusan, megszakításkérésrel ERR jelzés útján, de történhet X értékének megvizsgálásával is. Ha a program a megszakításkérést maszkolja, akkor minden esetre X-vizsgálatot kell végeznie. A folyamatábrán szaggatott vonallal jelölt téglalapok csak >12 bit adatszélesség esetében vannak jelen. Az ábrán az "1. FÉLADAT" megjelölés rövidszavas átvitelnél maga az adat, >12 bit széles szavak átvitelénél pedig az alsó 12 bit; a 12 biten felüli részt a "2.FÉLADAT" jelöli meg.

A módosított PDP-8/CAMAC utasítások használata mind az egy CAMAC parancsra eső gépi utasítások, mind a keretvezérlőben alkalmazott regiszterek mennyisége tekintetében jelentős megtakarítással jár. A megtakarítás annak következménye, hogy a CAMAC ciklusban végzett adatátvitelre nézve a keretvezérlő rövidszavas átvitelnél teljesen, hosszuszavas átvitelnél pedig az 1. félszóra nézve transzparens; a transzparens adatátvitel a CAMAC modul és a PDP-8 akkumulátora között közvetlenül megy végbe, tárolóregisztert és e tárolóregiszterrel kapcsolatos gépi műveleteket tehát nem igényel. De vezérlés jellegű és adatbevitellel járó CAMAC művelet végrehajtásához nem szükséges CNA-nak a keretvezérlőben való előzetes tárolása sem; e műveletek végrehajtásakor a keretvezérlő CNA-ra nézve is transzparens. A fent leírttól eltérő PDP-8-CAMAC megoldások is vannak, pl. [33], ezek lényege azonban a fent leírtakkal egybevág.

A PDP-11 tulajdonságu gépeknél a tervezési fogások nem nyújtanak olyan látványos eredményeket, mint a PDP-8 esetében, mivel a gép adottságai a CAMAC-hoz eleve jobban illeszkednek; a 16 bit szélesség sokkal több igényt kielégít, mint a 12 bites, s ezért az alkalmazási esetek nagyobb részében kielégítő a rövidszavas átvitel mód is. Ahol viszont hosszuszavas átvitel mód szükséges, ott hasonlóképpen kell eljárni, mint a PDP-8 esetében, ám a 2. félszó ekkor mindössze 8 bit (ugyanilyen széles tehát a keretvezérlő adatregisztere is). Mivel a PDP-11-nek

kétcimű utasításai vannak, a transzparens módon átvitt adatok közvetlenül a CAMAC modul és a számítógép memóriája között mozognak (akkumulátor az átvitelben nem szerepel). A CNAF együttes 17 bitje azonban nem fér el sem adatként a memóriában, sem cimként az utasításban, és ezért valami módon ez esetben is két részre kell bontani. A bontást [34] pl. oly módon végzi el, hogy CNA a PDP-11 utasítás címébe kerül, az F pedig "adatként" szerepel. Így a gépi utasítás közvetlenül címzi meg a CAMAC modulok elemeit (A). Ennek a megoldásnak az a hátránya, hogy a számítógép memória cimtartományának jelentős részét foglalja, ha a keretek száma nő. Egy keret kezelése ui, 512 külső címét igényel, s nyolc keret már 4K címet foglal le. Ezenkívül az F kódot, adat be- és kiviteli műveletek kapcsán egyaránt, előzetesen át kell vinni a keretvezérlőbe. A műveletekben és regiszterekben elérhető megtakarítás így nem számottevő. Ezen a helyzeten az sem változtat, ha - mint az a [35]-ben le van írva - a számítógépi utasítás a keretet előírászerűen, periférikus készülékként címezi, a parancs NAF komponenseit a számítógép adatként kezeli. Az egyetlen így elért eredmény: a memória cimtartományát a CAMAC modulok nem terhelik. Ennek a megoldásnak azonban van némely hátránya is az előzővel szemben. Ha az NA CAMAC cím a PDP-11 utasítás cím részében helyezkedik el, úgy a PDP-11 utasítások egy kereten belüli modul-modul műveleteket is végrehajthatnak, pl. egy adatot átvihetnek a keret egyik moduljának valamely regiszteréből egy másik moduljának valamely regiszterébe. Az NA-t PDP-11 adatként kezelve, ez nem lehetséges. Azt is figyelembe kell venni, hogy míg F tárolására a memória félrekesze is elegendő, az NAF elhelyezése már teljes rekeszt igényel.

A PDP-11 sinrendszere, az UNIBUS, sokkal inkább rokon az adatuttal, mint a PDP-8 programozott B/K sinje. Ezért viszonylag egyszerű adapter-áramkörrel a CCA-1 keretvezérlő közvetlenül is rácsatlakoztatható az UNIBUS-ra [36].

A CAMAC igények kezelése a PDP-8-al meglehetősen nehézkes. A gépnek csak egyetlen megszakítási szintje van, minden igény erre az egyetlen szintre érkezik be, s a beérkező igények közül az előnyt élvezőt, előre rögzített prioritási séma alapján,

program azonosítja. A prioritási sémának az igényosztályozók logikájával összhangban kell állnia. Az azonosítás a hierarchia felső szintjéről lefelé haladva történik: először a keret, majd a modul, végül a modulon belüli elem identifikálódik. A párhuzamos főut viszonyaihoz képest itt annyi a különbség, hogy az azonosítási folyamatban szereplő GL igényvektorok csak egyetlen keretre vonatkozó információt hordoznak.

A PDP-11 vektorális megszakitási rendszerének az a sajátossága, hogy az igénylő periferikus készülék bináris címének beküldésével kell, hogy azonosítsa magát. Ez a bináris cím, amely mutatóként irányul az igényelt szolgálati rutinra, nem tévesztendő össze az igényosztályozók által összeállított bináris vektorral, melyet pl. a CCA-1 keretvezérlő a BG lehívó jelre ad válaszként. Ez utóbbi ui. nem szám, nem cím, hanem olyan lineáris pozíciókód, amelynek minden bitje önálló értelemmel bír, amelynek minden bitje meghatározott igénylőre vagy igénylők együttesére utal, s ez igénylőt vagy igénylőket a vektoron belül elfoglalt helyzete jelöli meg. A PDP-11 logika alapján minden U-vezérlőnek legalább egy bináris azonosítója kell, hogy legyen, amelyik az adott keret szolgálati rutinjára mutat. A kereten belüli igénylők azonosíthatók programozási úton is, ami azonban nehézkes. Célszerűbb az igényjelre válaszul: GL funkcióval ekvivalens műveletet szerkezeti úton végrehajtani, s a kapott GL-vektort prioritási kódolóval kezelni. A prioritási kódoló az előnyt élvező igénylő bináris címét adja, amelynek alapján a PDP-11 közvetlenül konkrét szolgálati rutinjára térhet rá. A GL-vektor elemeinek bináris címekre való leképezését, a keretvezérlőhöz társított igényosztályozó végezheti el [34]. Egy bináris cím mögött ez esetben is több igénylő húzódhat meg, amelyek közül a prioritást élvező programozási úton határozható meg közelebbről.

Ágvezérlők: Az ágvezérlő a paralel főutat illeszti a számítógéphez hasonló módon, mint amiként az U-vezérlő az adatut. A passzív ágvezérlő nem generál parancsokat, másrészt az ágparancsokra ($C > 0$) nézve transzparens, azokat a keretvezérlőkre továbbítja. Ugyanugy, mint a keretvezérlőnek, az ágvezér-

lőnek belső parancsai. A CNAF formátumban az ágvezérlő belső parancsait C=0 érték jelöli meg. Ugyancsak C=0 jelöli meg az ág BZ vonalát működtető közvetett ágparancsot is. A BG vonal, ha működtetése nem szerkezeti uton, hanem programmal történik, belső parancssal aktivizálható.

Az ágvezérlő esetében a CAMAC parancsoknak számítógépi utasításokkal való végrehajtása az U-vezérlőnél már bemutatott módon történik. A [37] közleményben ismertetett vezérlő, a [34] és [35] -ben választott megoldásoknak mintegy kompromisszumaként, a CNAF parancsok csak az NA-részét értelmezi a PDP-11 utasítások címkomponenseként, a CF-et pedig a gép adatként kezeli. Ennek megfelelően pl. adatkivitelnél először CF átvitelére kerül sor a memóriából az ágvezérlő 'CF' regiszterébe, majd a CAMAC ciklust, és ezzel az adatkivitel is, az NA-t tartalmazó gépi utasítás kezdeményezi, ill. hajtja végre.

Az ágvezérlő a BD igényeket kezelheti programozási uton is, de a számítógép munkáját általában bizonyos egyszerű áramkörök alkalmazásával megkönnyítik. A [37]-ben ismertetett ágvezérlő esetében a BD jelre nem a számítógép, hanem az ágvezérlő válaszol BG jel képzésével, minek hatására az összes az ágon fekvő keret az adatutvonalon át együttes GL vektort irt be az ágvezérlő adatregiszterébe. A GL-vektor prioritás kódolóra kerül, mely az előnyben részesülő bitnek megfelelő bináris cím-azonosítót továbbít a számítógéphez. A részletesebb azonosítás - amennyiben ez szükséges - a CAMAC eljárásoknak megfelelően, program útján történik. Az igénylő azonosítását megkönnyítő áramkörök az ágvezérlő organikus részét képezik.

Soros keretvezérlők. Az ágvezérlő és az A1 típusu keretvezérlő között ponttól-ponthoz kapcsolat áll fenn. Ezzel szemben a hurokvezérlő és a soros keretvezérlő közötti kommunikáció üzenetek alakjában történik. A CAMAC parancsokat a hurokvezérlő parancsüzenetek alakjában küldi a hurokszervezésű soros vonalra. A nem neki szóló üzenetet a soros keretvezérlő (SCC) változatlan alakban továbbítja a mögötte fekvő vonalszakaszra, a neki szóló üzenetet viszont elcsipi és végrehajtja. A parancsüzenet (éppugy mint a soros rendszer másik két üzenettípusa is)

byte-onkénti paritásbitet, valamint ellenőrző-összeget is tartalmaz; a soros vezérlő ennek megfelelően rendelkezik a horizontális és vertikális ellenőrzés áramköri elemeivel. Adatkivitelkor a parancsüzenet az adatot is magával viszi. A keretvezérlő a bit-sorosán, ill. byte-sorosán érkező üzenetet az adatutra párhuzamosítja. E célra soros-paralel átalakítóval rendelkezik, mely szükségképpen CNAF- és adatregisztert is kell, hogy felöleljen. Az SCC-L vezérlők mind bitsoros, mind byte-soros üzenetek kezelésére alkalmasak; az üzemmódot a keretvezérlő használója választhatja meg.

A hurok-szinkronizmus egészének fenntartása, és az üzenetek egyértelműségének érdekében az üzenetek közeit ún. várakozási byte-ok töltik ki. Mialatt a megcímzett keretvezérlő a neki szóló üzenetet feldolgozza, ő maga bocsát várakozási byte-okat a mögötte fekvő vonalszakaszra.

A CAMAC parancs végrehajtására csak a teljes parancsüzenet vétele és ellenőrzése után kerül sor. A keretvezérlő minden elfogadott parancsüzenetre válaszüzenettel reagál. A válaszüzenet mindig tartalmazza az érintett keret (C) címét, valamint egy állapotvektort, melynek Q és X értékén kívüli két további bite hibátlan vagy hibás működésre utal. A válaszüzenet olvasás művelet kapcsán továbbá tartalmazza még a (24 bites) adatot is. A válaszüzenet kibocsátása bizonyos időt vesz igénybe. Annak érdekében, hogy a byte-szinkronizáció ezen idő alatt is fennmaradjon, a hurokvezérlő minden parancsüzenet végéhez térköz-byte-okat csatol. Az így keletkező "válasz-térközt" VÉGE-byte zárja le, jelezve a parancs - válasz szekvencia lezárultát, egyúttal üzenetszinkronizációs jelet is képezvén a soros főúton lévő összes keretvezérlő részére. Ugyancsak határolójelként szolgál a válaszüzenetet záró VÉGÖSSZEG is.

A soros keretvezérlő a kereten belüli (esetleg külső) igény felléptét igényüzenet kibocsátásával jelzi. Az igényüzenetet be kell sorolni a soros főútra, de mivel az igény véletlenszerűen jelentkezik, ez nem megy minden további nélkül. Igényüzenetet a keretvezérlő csak akkor bocsáthat ki, ha nem hajt végre műveletet vagy üzenet nem halad át rajta. Az igényüzenet kibocsá-

tására a főut byte-szinkronizációjának megfelelően csak egy háttároló jelet követően, s valamely üzenet kezdetét megelőzően kerülhet sor. Az igényüzenet három byte-ból áll. Hogy kibocsátása idején az időközben esetleg a főutról a keretvezérlőhöz érkező üzenet ne zavarjon, az érkező üzenetet a keretvezérlő - bemeneténél - megkéslelteti. Így az igényüzenet sohasem ütközhet a hurokvezérlő vagy másik keretvezérlő által kibocsátott üzenetekkel.

Az igényüzenet az öt kibocsátó keret címén kívül egy a kereten belüli igénylőket közelebbről meghatározó 5 bit széles bináris vektort is tartalmaz, amint azt már korábban említettük. A bináris vektor kódolása kétféle lehet: vagy lineáris pozíciókód vagy bináris szám. Amennyiben pozíciókód, úgy minden egyes bitje meghatározott igénylőt vagy igénylőket képvisel. Amennyiben bináris szám, akkor az egyidejűleg jelentkező több igénylő közül az előnyben részesülő igénylő azonosítója. A kódolást éppugy, mint a párhuzamos keretvezérlők esetében, itt is a vezérlőhöz társuló, és a mindenkori viszonyokhoz igazodó igényosztályozó végzi, és pedig bináris szám-kódolás esetében prioritás kódoló igénybevételével.

A SCC byte- és üzenet-szinkronizációval egyaránt rendelkezik. Hálózati bekapcsoláskor speciális szinkronizációs szekvenciát futtat le, amelyik mintegy behelyezi a keretvezérlőt a főut forgalmába.

A különböző SCC funkciók ellenőrzésére és vezérlésére speciális CAMAC parancsok szolgálnak. A nullázás (Z) törlés (C), tiltás (I), off line/on line, mellékszár és hurok-bénulás vezérlés, stb. funkciók végzésére külön parancsok helyett az SCC egy státuszregisztere szolgál, melynek bitjei szabadon olvashatók, beírhatók, szelektíven állíthatók, ill. törölhetőek.

A soros főutat definiáló EUR 6100e, ill. ANSI/IEEE 595-1976 előírások a főut specifikációjának mindenben megfelelő keretvezérlőt is definiálnak és leírják, s a keretvezérlőt SCC-L2-vel jelölik.

Hurokvezérlők. A hurokvezérlő fő funkcionális részei: a számítógép-csatlakozó, a hurok-kimenet és a hurok-bemenet. Az u-

többi kettő a soros főút előírásainak megfelelő üzenetek kibocsátására, ill. fogadására szolgál, a számítógép-csatlakozó a mindenkori számítógéppel való kapcsolatot látja el. A hurokvezérlő a számítógéphez annak a periferikus készülékekre előirt normái szerint csatlakozik. Mielőtt parancs-üzenet képzéséhez és kibocsátásához lehetne fogni, a CNAF parancsot, s adatkivitel esetén még a kiviendő adatot is, gépi utasításokkal a számítógép-csatlakozó e célra szolgáló regisztereibe kell áttölteni.

A hurok-kimenet lényegében (bit soros vagy byte soros) paralel-soros átalakító, mely azonban kibővül paritásképző és ellenőrző-összeg képző, formátum képző, határoló byte, térköz- és várakozás-byte képző áramkörökkel is. A kibocsátott üzenet hossza függ az üzenet által hordozott CAMAC parancs értelmétől. Van adat nélküli rövid és van adatot is hordozó hosszú üzenet, s a térköz byte-ok száma is változhat, ezért a formátumképző áramkörnek a parancsot, még az üzenet összeállítása előtt, ebből a szempontból értelmeznie kell. A hurok-kimenethez tartozik végül az óraáramkör, mely az egész soros főút adatforgalmát szinkronizálja.

A hurok-bemenet soros-paralel átalakítást kell, hogy végezzon, előkészítve a válasz-üzenet, ill. az igényüzenet által hordott információt a gép által hasznosítható formában. A válaszüzenet tartalma a hurokcsatlakozó állapot-üzemmód regiszterének állapot részébe kerül, az esetlegesen bevitt adat pedig adatregiszterébe. Mielőtt azonban erre sor kerülne, a hurok-bemenetnek meg kell határoznia az üzenet típusát, hogy az üzenettípusnak megfelelő akciót fejthesse ki.

Az igényüzenet (cim + igényvektor) a hurokcsatlakozó igényregiszterébe tárolódik.

A hurok-bemenetnek paritás- és összeg-ellenőrző áramkörökkel kell rendelkeznie.

A hurokvezérlő állapot - üzemmód regisztere részben az üzenetek által közvetített állapot-biteket veszi fel, részben az üzemmódot meghatározó, s a számítógép által küldött biteket. Az előbbiek sorába tartoznak a Q és X, valamint a soros keretvezérlő által meghatározott hibabitek és a hurokvezérlő hiba-

bitjei, az utóbbiakhoz a megszakítási forrásokat maszkoló és az üzemmódot befolyásoló (megválasztó) más esetleges bitek.

A megszakításkérések kezelése az adott számítógép előírásainak megfelelően kell történnük, s esetenként a számítógépcsatlakozóban elhelyezkedő megszakítás előkészítő áramkört igényel. Megszakításkérés számos okból fordulhat elő. Előhívhatja igényüznetet, de számos hibaforrás is, pl.: a válaszüzenetben lévő hibajelzések, a szinkronizáció hibája, a paritás és kontroll-összeg ellenőrző áramkörök, de egy időmérő áramkör is, amelyik akkor jelez, ha valamely üzenet kibocsátását követően meghatározott időn belül nem érkezik válaszüzenet.

Az üzenetek kibocsátásának, ill. vételének, valamint a számítógéppel végzett egyidejű kommunikációnak a lebonyolítása a hurokvezérlő legfelsőbb szintű vezérlési funkciója.

Autonómiával rendelkező vezérlők. Valamely kísérlet lebonyolításával kapcsolatos adatgyűjtési, adatszétosztási, vezérlési, igényazonosítási, stb. algoritmusok a CAMAC hálózatban CAMAC parancsokból álló programok végrehajtásaként valósulnak meg. Passzív vezérlőket alkalmazva, e programok kizárólagos forrása a hálózat mögött álló programforrás - feltételezésünk szerint: számítógép. A CAMAC parancsokat a számítógép kimenő adatként kezeli, s mint ilyet továbbítja a vezérlőbe. Egy parancs aktivizálása s a vele esetleg együttjáró adat kezelése általában több számítógépi utasítást igényel, ezért a fentiekben már bemutatott tervezői fogásokkal élnek a gépi műveletek idejének csökkentése céljából. Az e fogásokkal elérhető eredmény azonban csekély, mivel a passzív vezérlők a számítógép programozott csatornájára csatlakoznak, melynek átbecsátóképessége eleve szerény. Célszerűbb a CAMAC parancsokat a gépi utasításoktól függetleníteni, és a teljesítőképességet egyrészt a programfüggetlen csatorna (közvetlen memória kapcsolat: DMA) igénybevételével, másrészt a CAMAC programok generálási feladatának a CAMAC vezérlőre való áthárításával fokozni. Ez a módszer azonban csak olyan vezérlőknél alkalmazható, amelyek a számítógéphez közvetlenül csatlakoznak, tehát U-vezérlők, ág-vezérlők és hurokvezérlők esetében.

A CAMAC parancsokat önállóan vagy önállóan is generálni képes vezérlőket autonómiával rendelkező vezérlőknek nevezzük. A programfüggetlen csatorna felhasználása és a helyi vezérlőn belüli programgenerálásra való áttérés a különböző vezérlők-nél más-más mérvű lehet, aminek megfelelően az autonómia foka is változhat. A legfejlettebb vezérlők, a CAMAC processzor és a CAMAC mikrogép, saját memóriával és utasításkészlettel rendelkeznek, a programgenerálást és adatkezelést önállóan végzik el, s ezért általában keretvezérlőként is alkalmazhatók. A CAMAC mikrogépek nem csak az adatgyűjtés, -szétosztással kapcsolatos feladatokat, de adatprocesszálást is végezhetnek.

Parancs-generálás a programfüggetlen csatornán át. A programfüggetlen csatornán át történő parancsgenerálás elemi módját a 21. ábra [39] szemlélteti. A vezérlő a programozott és a programfüggetlen csatornával egyaránt összeköttetésben áll, s munkáját a parancsgenerátor irányítja. A CNAF CAMAC parancs, adatmezőivel s a parancs végrehajtásának feltételeit meghatározó, lefutását regisztráló vezérlő - állapot vektorral (VÁV) együtt a számítógép memóriájában foglal helyet. A számítógép - B/K utasításával - csupán egyetlen címet (X) közöl a vezérlővel, mely a VÁV és a CNAF-t tartalmazó memóriaszegmensre mutat. A címet közlő utasítás hatására a parancsgenerátor a programfüggetlen csatornán át automatikusan lehívja a VÁV-t és CNAF-t (U-vezérlő esetén csak NAF-t) és ez utóbbit a VÁV tartalmának megfelelő feltételek szerint aktivizálja. Ha a parancs olvasás/írás, az általa kijelölt adatátvitel ugyancsak a programfüggetlen csatorna útján történik meg. CNAF általában ez esetben sem fér el egy memóriarekeszben. A gyakorlatban az ebből származó problémát úgy hidalják át, hogy CNAF egy darabkáját a VÁV részére fenntartott memóriarekeszben helyezik el [39].

Programozástechnikai szempontból, de az ésszerű vezérlő-architektúra kialakítása érdekében is, célszerűbb a parancsok egyenkénti kezelése helyett a meghatározott részfeladatok végrehajtására szolgáló CAMAC parancsok összességét egyetlen organizációs egységként kezelni. Egy-egy ilyen egység a számító-

gép memóriájában CAMAC programszegmenst képez. Valamely programszegmenst, passzív vezérlő alkalmazása esetén, a számítógép készülékkezelő programjához tartozó egy (a generálási paraméterekkel konkretizált) rutin aktivizál (generál) [38]. Korlátozott autonómiájú vezérlő esetében a számítógép készülékkezelő rutinjainak eme szerepkörét a vezérlő parancsgenerátora veszi át, de a CAMAC szegmens továbbra is a számítógép memóriájában marad. A CAMAC szegmens parancsait a DMA csatornán át a vezérlő parancsgenerátora hívja le, és az előzőleg megkapott paramétereknek megfelelő módon aktivizálja.

Parancsgenerálás parancs listával. A CAMAC programszegmens egymás mögé sorakozó parancsai listát képeznek. Nem szükségszerű minden egyes parancsot számítógépi utasítással aktivizálni; az utasításokat a parancsgenerátor sorban, automatikusan is lehívhatja. Ennek a gyakorlatnak megfelelő parancsgenerálási sémát mutat be a 22. ábra. A parancsgenerálást valamely állapot bekövetkeztét vagy operátori utasítást jelző LAM jel indítja. Hatására a parancsgenerátor mindenekelőtt a paramétertömbbel feltölti a vezérlő megfelelő regisztereit. Általános esetben a paramétertömb a vezérlő - állapot vektort és a parancslista, ill. adatlista kezdőcímét tartalmazza. A paramétertömb betöltését követően a parancsgenerátor lehívja az X helyen lévő első parancsot és aktivizálja azt. Adatbevitel esetén ekkor (CAMAC olvasás művelet) az első adat az Y címre tárolódik. Ezt követően mind az adat- mind pedig a parancscím-regiszter tartalma inkrementálódik (I), majd CNAF2 aktivizálásával megtörténik a második adat bevitele, a nevezett regiszterek tartalma ismét inkrementálódik, és így tovább. Adatkivitel esetén az átvitel iránya fordított. Az automatikusan ismétlődő folyamat akkor zárul le, amikor a CAMAC szegmens utolsó, speciális, a vezérlőnek szóló záró parancsa (CNAF i) végrehajtódik. Amennyiben a szegmens pl. tisztán vezérlő utasításokból állna (pl. törlés sorozat) a parancsgenerálás sem nem képez, sem nem használ adatlistát, s így a paramétertömbből az adatcím elmarad.

A parancslistával képzett parancsgenerálásnak a később tárgyalandó blokkátvitellel szemben megvan az a tulajdonsága, hogy

nem kell ragaszkodnia a CAMAC címek meghatározott sorrendjéhez, s így vele rendezetlen felkeresési sorrendű művelet sor is végrehajtható [37].

A 22. ábra feltételezi, hogy a parancsok és az adatok egy-egy rekeszben teljes egészükben elférnek. Ha ez nem így van, a vezérlő tervezése során műfogásokkal élnek. Az adatokat felső és alsó félre kell választani és egymásutáni két memóriaciklussal kell beírni, ill. kiolvasni egymással szomszédos rekeszekbe, ill. rekeszekből. A 17 bites CNAF elhelyezési problémáját <17 bit rekeszszélességű memória esetében két részre való bontással, vagy a parancsok bináris strukturájából kiinduló megfontolások alapján oldják meg. Az összes írás/olvasás parancs F8 helyértékén 0, a vezérlés-vizsgálat rendeltetésű parancsok F8 helyértékén 1 áll. Különválasztva a CAMAC írás/olvasás programokat (szegmenseket) a vezérlési-ellenőrzési szegmensektől a program végrehajtását kezdeményező LAM jel (vagy gépi utasítás) már eleve az F8 értékét is implicite magában hordozza [41,42], mely az adott szegmensre vonatkozóan állandó. Amennyiben a programszegmens csak egytipusú parancsokból áll, pl. olvasás rendezetlen sorrendű címekkel, úgy az F-kód a paramétertömbben helyezhető el, s a parancslista helyett CNA címlista alkalmazható (CNA 12 bit széles).

A blokkátvitel n -számu adathól álló blokkoknak a CAMAC hálózatból a memóriába (vagy fordítva) való átviteléből áll. Az n értéke, azaz a blokk hossza, általában előre meghatározott, egyes esetekben azonban a blokkok hosszúsága változik. A blokk átviteléhez homogén parancssort kell végrehajtani, ami egy és ugyanazon (írás vagy olvasás) művelet ismételtetéséből áll, miközben a memóriacím s esetleg, bár nem mindig, a CAMAC cím is állandó inkrementumokkal műveletenként változik. A blokkátvitel művelet sorának homogenitása a parancsgenerálás feladatát igen leegyszerűsíti.

Meglehetősen sokféle szabálygyűttesnek eleget tévő blokkátvitelmód ismeretes, amelyek közül itt a CAMAC ajánlások négy legfontosabbikát ismertetjük (IEEE 683-1976; EUR 4100 suppl.). A négy közül háromnak, a

- blokkátvitel végigjárással
- egycimű blokkátvitel
- blokkátvitel állj jelzéssel

közös vonása, hogy a műveletléptetés a Q válaszjel használatán alapul. A negyedik, mely ugyancsak egy-CAMAC-cimű, műveletléptetés céljára a CAMAC átviteli végpont LAM igényjelét használja.

A végigjárással végzett blokkátvitel két - előre megadott - kezdő és végcim közötti CAMAC cimtartomány összes cimeit sorban felkeresi abból a célból, hogy az e cimeken található regiszterek tartalmát kijelölt memóriaterületre vigye át vagy fordítva. Kiindulási parancsként, a vezérlőnek a blokkátvitelhez ismernie kell a CAMAC kezdő és végcimeit, a kijelölt memóriaterület kezdőcímét, valamint az átvendő adatok formátumát (CAMAC szélességű vagy csatorna szélességű-e). A parancsgenerálás F ismétlésével, s a CNA (vagy BrCNA), valamint a memóriacim inkrementálásával történik. A CAMAC végcim helyett - a körülményektől függően - megadható az átvendő adatok száma is. A blokkátvitel akkor ér véget, ha a CAMAC cim összeesik az előre megadott végcimmal vagy ha az átvitt adatok száma eléri az előre rögzített értéket. A végigjárással végzett blokkátvitel folyamán, minden olyan esetben, amikor a momentán cim valóban meglévő modul meglévő regiszterére mutat, vagyis amikor tényleges adatátadás történik, a vezérlő Q=1 (Q-t l. a 15. ábrán) visszajelzést kap. Ha a megcímzett helyen nincs regiszter, Q=0 jelzés adódik. A CAMAC cim így is továbbugrik, azonban anélkül, hogy közben a memóriacim is inkrementálódna. Ilyenképpen a CAMAC hálózatnak számos modulja, kerete sőt ága is folyamatosan végigjárható, még ha a fizikai cimtartományban hézagok volnának is. Ez az átvitel mód feltételezi, hogy az átvitel megkezdése előtt már minden érintett CAMAC regiszter átadásra, ill. fogadásra készen áll. Az átvitel ritmusát ezért a keretvezérlő határozza meg.

Egycimű blokkátvitelnél az egymás után következő adatok átvitele egy állandó CAMAC cim és kijelölt memóriaterület között történik, a CAMAC modul által diktált ütemben. A vezérlő az át-

adás megkezdése előtt kell, hogy ismerje a memóriaterület kezdőcímét, a CAMAC címet, továbbá az átvendő adatok számát és formátumát. A modul $Q=1$ -el jelzi, ha az adatot átadta vagy átvette, $Q=0$ -al pedig, ha a vezérlő átviteli kísérletét visszautasítja. $Q=0$ esetén a memóriacím inkrementálása nélkül addig ismétlődik a parancs, amíg csak végül $Q=1$ jel adódik. $Q=1$ -el a memóriacím inkrementálódik. Előírt számu $Q=0$ választ adó sikertelen próbálkozást vagy előírt időtartamot követően, a vezérlő hibajelet kell, hogy képezzen, különben meddő hurokba ragad. A blokkátvitel az előkészített adatszám elérésével ér véget.

Az állj jelzéssel végzett blokkátvitel ugyancsak egycimű, azaz rögzített CAMAC címmel történik, de az átvitel ütemét itt a vezérlő diktálja. A modul minden adat átvitelkor $Q=1$ jelzést ad, a blokkot pedig $Q=0$ jelzés zárja.

A LAM-szinronizált átvitel a Q -szinkronizálttól abban különbözik, hogy az egyes adatok átvitelére LAM-al a modul ad jelt, s így a vezérlőnek nem kell lekötve lennie, s állandóan ismételtetnie a CAMAC parancsot (míg $Q=1$ -t nem kap). Az átvitel az előre megadott adatszám elérésével vagy a $Q=0$ jel vezérlő általi vételével zárul.

Míg a végigjárással végzett, valamint az egycimű blokkátvitelmód csak fixhosszusú blokkok, az állj jelzéssel végzett, valamint a LAM szinkronizált átvitel mód előre meg nem határozott hosszúsú blokkok esetében is alkalmazható. A LAM-szinkronizált átvitel mód különösen csekély adatáram és többcsatornás vezérlő esetében használható előnnyel, mivel a hosszú időre elnyúló adatblokk nem köti le a vezérlőt, csupán az egyes adatok ritkásan bekövetkező átviteli ciklusainak idejére. A LAM-szinkronizált átvitel mód eképpen lehetővé teszi az adatmultiplexelést, míg a Q -szinkronizált átvitel mód csak blokkonkénti multiplexelést engedélyez.

Egyes vezérlők [34,40] a blokkátvitelmódban olvasott adatokkal többféleképpen járhatnak el:

- az adatok, a már ismertetett módon, listaszerűen a memóriába íródnak

- az adatok sorban hozzáadódnak a memóriarekeszek már meglévő tartalmához
- a kiolvasott adatokat a vezérlő nem adat, hanem memóriacimként értelmezi, s minden olvasás művelet kapcsán a momentán címmel jelölt memóriarekesz tartalmát eggyel növeli (hisztogram képzés).

Vannak számítógépek, amelyeknek programfüggetlen csatornája e lehetőségekkel maga is rendelkezik. Az SPC-16 pl. a fenti három üzemmódon kívül még egy negyediket is ismer, a kiolvasott adat a listában soron következő rekesz tartalmához logikailag adódik hozzá.

A blokkátvitelt mindig CAMAC igény kezdeményezi, és az, az előre meghatározott feltételek szerint, a parancsgenerátor irányítása alatt automatikusan fut és zárul le. A zárófeltétel beálltát a vezérlő érzékeli, s jelzi a számítógépnek. Hatására a számítógép előírt zárórutint futtat le.

A CAMAC igény (LAM) kétféleképpen kezdeményezhet blokkátvitelt: közvetlenül, és közvetett módon. Közvetlen kezdeményezés esetén LAM egyenesen a vezérlőre hat [37,41,42] közvetett kezdeményezés esetén pedig a számítógép megszakítórendszerén az igény hatására lefutó megszakítási rutinon át [39,43].

A blokkátvitelt, megkezdése előtt, a paramétereknek a vezérlő paraméter-regisztereibe való betöltésével, elő kell készíteni. Az előkészítés történhet a blokkátvitelt kezdeményező LAM jel megjelente előtt [41,42], de történhet a LAM jel hatására is a programfüggetlen csatornán át [37] a programgenerátor akciójaként vagy a LAM által kiváltott megszakítási rutin útján a programozott vagy programfüggetlen csatornán át [43], ill. [39].

Egy és többcsatornás vezérlők ismeretesek. Az egycsatornás vezérlő a CAMAC cimterület és a memória cimterület között csak egyetlen kapcsolatot létesíthet. A többcsatornás ezzel szemben több egymástól független kapcsolatot multiplikál. Van többcsatornás vezérlő, amelyik cserélgetéssel a paraméter-regisztereknek egyetlen készletét használja valamennyi csatorna számára [37], míg más vezérlőknél [35,41,42] a regiszterkészletek száma megegyezik a csatornák számával.

Az egyszerűbb vezérlők üzemi lehetőségei korlátozottak, míg más vezérlők felhasználóik választása szerint a fentiekben felsorolt csaknem minden üzemmódban (különböző blokkátviteli módok, utasításlista kezelés stb.) dolgozni képesek.

A 23. ábra egy közepesen bonyolult vezérlő-architektúrát szemléltet [39]. A vezérlő Q ill. LAM szinkronizált blokkátviteli üzemmódban egyaránt dolgozhat, a MICRO-16 V 16 bit szószélességű számítógép mellett. Az üzemmód a kettős szóhosszuságu VÁV vezérlő-állapot vektor útján választható meg, mely a BrCNAF (kezdő) parancsot is tartalmazza. A vezérlő előkészítése a blokkátvitelt kezdeményező LAM jel megjelenése után a programfüggetlen csatornán át történik. A LAM jel a számítógép megszakitási rutinja révén közvetve indítja a blokkátvitelt. Az indítás egyszerű, a paramétertömb címének közléséből áll, minek alapján a parancsgenerátor lehívja a paramétertömböt. A blokkátvitel a VÁV-ban meghatározott üzemmódban, autonóm fut le.

Itt jegyezzük meg, hogy azoknak a számítógépeknek az esetében, amelyeknek programfüggetlen csatornája háromciklusú üzemmódban is dolgozik, a blokkátvitelű vezérlők adatszám és blokkcím regiszterét a memória erre kijelölt rekeszei helyettesíthetik. E lehetőség felhasználása egyszerűsítheti a vezérlő architektúráját, ám a háromciklusú üzemmódban dolgozó csatorna átbocsátóképessége az egyciklusúénál lényegesen kisebb.

Parancsgenerátorok. A vezérlőnek az előbbieken ismertetettken túl, még más autonóm vonásai is lehetségesek [44], így pl. a CAMAC modultól CAMAC modulhoz való blokkátvitel a memória érintése nélkül; vezérlő-vizsgáló utasítások szekvenciális végrehajtása, stb.

A vezérlők vezérlője, a programgenerátor, éppoly számosféle lehet, mint a vezérlő által megvalósítható átvitel módok. Az egyik oldalon a csupán igen egyszerű algoritmusokat végrehajtó, kötött programozású (huzalozott program), a másik oldalon a többféle átviteli algoritmust megvalósítani, sokféle feltételt figyelembe venni képes, aritmetikai-logikai műveleteket is végző változtatható programu parancsgenerátorokat találjuk. A progra-

mok elhelyezkedhetnek fix tárban, ami egyes vezérlőknél pl. dugaszolható diódamtárix, másoknál olvasható (ROM)-tár, de elhelyezkedhetnek olvas/ir tárban is. A fix táruk alkalmazásának az a hátránya, hogy programjaikat nehézkes a kísérlet körülményeinek és feltételeinek megfelelően változtatni; az olvas/ir táruk programjai viszont a számítógépből egyszerűen áttölthetők, így könnyedén módosíthatók és cserélhetők.

Hurokvezérlő autonómiával. A hurokvezérlőnek is lehet autonómiája. A számítógép a hurokvezérlő állapot-üzemmód regisztre üzemmód bitjeinek beállításával döntheti el, hogy a vezérlő passzív vagy autonóm üzemmódban működjek-e. Egy adott hurokvezérlőnél [45] pl. e célra az üzemmód regiszter négy bitje közül kettő, az ismétlés-mód és a végigjárás-mód bit szolgál. Ha a számítógép az ismétlés-mód bitet állítja át, a hurokvezérlő az elsőként kibocsátott üzenetet ismételteti mindaddig, míg hibamentes válaszüzeneteket kap. A végigjárás mód bitjének átállításával a hurokvezérlő ugyancsak ismételteti az első parancs-üzenetet, de az egymásután kibocsátott parancsüzenetekben a CAMAC cím folyamatosan inkrementálódik.

CAMAC processzorok. CAMAC processzoron olyan autonóm vezérlőt értünk, melynek parancsgenerátora erre a célra szolgáló, saját olvas/ir memóriával ellátott, speciális utasításkészlettel rendelkező processzor. Gyakran mikroprogramozott vezérlőnek is nevezik, a CAMAC parancsoktól és a számítógép utasításaitól így akarván megkülönböztetni a vezérlő processzorának "mikroutasításai"-t. A mikroutasítás, mikroprogram elnevezés azonban legalábbis nem mindig indokolt, a terminológia eredeti értelmében. Az alábbiakban példaképpen leírt MBD-11-nek pl. ilyen utasításai vannak: adatmozgatás, inkrementálás, összeadás, kivonás, logikai VAGY, ÉS, elágazás-ha-igaz stb., vagyis nem mikro, hanem alapszintű utasítások. Le kell rögzítenünk, hogy a CAMAC processzor sokrétűsége ellenére sem tesz mást mint, hogy adatokat továbbít, és részt vesz a CAMAC oldalról jövő igények feldolgozásában, és pedig a számítógép által meghatározott módon. A CAMAC processzor a lehető legnagyobb mértékben tehermentesíti a számítógép processzorát - esetleg és részben memóriáját

is - az adatgyűjtéssel, -szétosztással, vezérléssel, igénykezeléssel járó feladatoktól, de az adatokat mindkét irányban érintetlenül bocsátja át, rajtuk processzállást nem végez. Ez alól csak a már említett hisztrogram-processzállási és az "add a már meglévő memóriatartalomhoz" automatikus üzemmódok képeznek kivételt, mely üzemmódok azonban esetenként az autonóm vezérlők igen egyszerű változatainál is megtalálhatók.

A CAMAC processzorok felépítésmódját az MBD-11 ágvezérlő ismertetésével mutatjuk be, melynek egyszerűbb kivitelét [41], továbbfejlesztett módusulatát [42] írja le.

Az MBD-11 a CAMAC ág és PDP-11 között létesít átviteli kapcsolatot. Három fő funkcionális egységből áll, s ezek:

1. A PDP-11 csatlakozás.
2. Az ághajtó.
3. Az 1. és 2. egység közötti kapcsolatot létesítő processzor és ennek helyi memóriája (egyszerűség kedvéért a kettőt együtt nevezzük processzornak).

Az MBD-11 a PDP-11-hez olyan periférikus készülékként csatlakozik, melyet egyrészt a számítógép programkapcsolatu átvitel módban kezelhet, másrészt amelyik közvetlen memóriakapcsolatot is létesíthet. A közvetlen memóriakapcsolatra az MBD-11 processzora max. 8 CAMAC csatornát multiplikál. Az egyes csatornákhöz sajátos, a mérőrendszer tervezője által megválasztott átvitel módok rendelhetőek. Az egyes csatornákra választott átvitel módokat a processzor helyi memóriájában helyet foglaló csatornaprogramok realizálják. E csatornaprogramokat a processzor utasításkészletének ismeretében a rendszer tervezője írja meg. A programok a PDP-11-ből csatornánként töltődnek át, és azok a rendszer használata közben meg is változtathatók. A processzor, a csatornák számának megfelelően, 8 regisztertömbbel rendelkezik az egyes csatornák átviteli paramétereinek felvételére. Minden egyes csatorna regisztertömbjében megtalálható: adatlista cimregiszter, parancslista cimregiszter, adatszám regiszter, CAMAC parancs regiszter, csatorna-cim mutató és két munkaregiszter. A csatorna-cim mutatóban egyrészt az adott csatornaprogram helyi memória kezdőcíme, másrészt a csatornaprogram

végrehajtásmódjára vonatkozó információ található. Két munkaregiszterrel a programozó szabadon rendelkezhet. Minden egyes regisztertömbnek van egy másolata is, mely az eredeti átviteli paramétereket őrzzi, s melyből e paraméterek eredeti értéke minden munkaciklus után újra nyerhető anélkül, hogy a processzornak a PDP-11-hez kéne fordulnia.

Az egyes csatornaprogramokat a hozzájuk rendelt CAMAC igényszintek jelei indíthatják. Első fokon a CAMAC igényeket az ág-hajtó kezeli. Ez az ág BD jelére BG jellel válaszol, mire beérkezik az osztályozott igényvektor. Ennek 24 bitjéből az alsó 8 a csatornához van rendelve, a többi 16 pedig egyéb szolgálatkérés. A csatornáknak szóló igényjelek prioritás osztályozóra kerülnek, mely a legmagasabbszintű csatorna-igényt fogadja el, s ennek programját aktivizálja. Az aktivizált program nem szakítható meg egy másik - netán magasabb prioritásszintű - csatorna igénye alapján sem. A csatornaprogram a PDP-11-nek szóló megszakító jellel végződik.

A nem csatorna rendeltetésű 16 igényszint prioritási sorrendben közvetlenül a PDP-11-hez van hozzárendelve. A PDP-11-hez összesen 25, prioritási sorrendbe állított, megszakítási szint tartozik a CAMAC processzor részéről. A legmagasabb szintű a processzor belső üzemi zavarait jelző hibajel, majd a csatornaprogramok 8 zárójele következik, s végül az osztályozott LAM vektor 16 bitje. A CAMAC processzor PDP-11 csatlakozó egysége, e jeleknek megfelelően, a számítógép megszakítási rendszere által előírt (vektoriális) címmutatókat állítja elő, a prioritási rendnek is eleget téve. A 25 megszakítási szint a számítógépnek csupán egyetlen megszakítási vonalát foglalja le.

A programozásmódtól és a kiépítettségétől (memória-kapacitás) függően a CAMAC processzor többféle üzemmódban dolgozhat, s az üzemmód csatornánként is változhat. Dolgozhat a PDP-11 memóriájában elhelyezkedő parancs- és adatlistával. A CAMAC parancsok és adott esetben az adatlista is elhelyezkedhet azonban a helyi memóriában is. E mellett a blokkátvitel-típusok bármelyike realizálható. A CAMAC processzor új minőséget jelent az autonóm vezérlők sorában. Azonban ez az új minőség nem csak a

szabad programozhatóság, a csatornák multiplikálhatósága és a flexibilitás tényezőiből áll össze. Igen fontos vonása az is, hogy helyi memóriával rendelkezik. Ha a helyi memória kapacitását megfelelően kibővítik, befogadhatja nem csak a CAMAC parancsokból álló szegmenseket, hanem a begyűjtött vagy kiviendő adatokat is. Ebben az esetben munkája a számítógéptől szinte teljesen független. Az ilyen felsőszintű autonómiával rendelkező CAMAC vezérlőnek már nem kell közvetlenül a számítógéphez csatlakoznia; olyan keretvezérlőként is alkalmazható, amely a kerettel kapcsolatos minden funkciót elláthat. Adatbegyűjtésnél pl. először az adatokat saját memóriájában tárolja, majd a begyűjtési ciklus végeztével azokat egy tömbben a számítógéphez továbbíthatja. Ugyanugy, egy tömbben kaphatja meg a számítógéptől saját programjait, a CAMAC szegmenst, a kerethez továbbítandó adatokat is. A CAMAC processzornak a keretben való elhelyezése azzal a kézzelfogható előnnyel is jár, hogy pl. az adatbegyűjtés a keret által megengedhető maximális sebességgel végezhető a begyűjtési ciklusban, majd az adatok továbbítása egy a sebesség szempontjából nem kritikus időszakban (pl. gyorsító szünete) lassabb tempóban továbbítható a számítógép irányába. Az a korábban felhozható aggály, hogy a helyi memória költséges, az egy bitre eső tárköltés rohamos esése folytán mára érvényét veszítette. A CAMAC processzortól csupán egyetlen lépés az intelligens CAMAC vezérlő, amelyik mielőtt a már begyűjtött adatokat továbbítaná, azokon előprocesszálást is végez.

Rendszervezők, többágu rendszerek egy és több programforrással.

Egy vagy egynél több ágnak egy vagy több programforrással (számítógéppel) való kommunikációs kapcsolatát a rendszervező (R-vezérlő) látja el. Amint a keret-, ág- és hurokvezérlőnek, ugy a rendszervezőnek is számos változatát fejlesztették ki [30,31,46-49,A5]. A rendszervező, mely az ágak számának megfelelő számú ághajtóval (AH) egészül ki, általában (ha nem is mindig) rendszerkeretben foglal helyet, ezért hasz-

nálják gyakran a "rendszervezérlő" szinonimájaként a "rendszerkeret" elnevezést. Magáról a rendszerkeretről már leirtuk, hogy az vagy nem szabványos módon használt szabványos- vagy a szabványoshoz képest bizonyos vezetékekkel kiegészített CAMAC keret.

A több ágat ellátó, de egyetlen számítógéphez csatlakozó rendszervezérlő szervezésének egy lehetséges sémáját (rendszerkeret használatát feltételezve) már a 18. ábrán szemléltettük. A rendszerkeretben standard CAMAC modulok is jelen vannak. A CAMAC modulokkal az R-vezérlő a CCA-1-el kongruens módon, az AH blokkokkal azonban nem standard módon kommunikál; ez utóbbihoz a szabvány adatuton kívül még egy járulékos sint (JS) is használ.

Nem CAMAC keretben elhelyezkedő, többágu, de csupán egy számítógéphez (CDC 3100) csatlakozást adó rendszervezérlőt ír le [46]. E vezérlő felépítésmódján jól kirajzolódnak a többágu rendszerek legjellegzetesebb sajátosságai. A vezérlő egy a számítógép csatorna-rendszeréhez kapcsolódó csatlakozó egységből és áganként egy-egy ághajtó egységből épül fel. Az ágak maximum lehetséges száma: 8. Az ághajtó a CAMAC előírásainak megfelelően kommunikál a hozzá csatlakozó ágakkal, a számítógép-csatlakozó pedig a CDC előírásoknak tesz eleget. Számítógéppel programozott és programfüggetlen átviteli kapcsolat egyaránt lehetséges. A rendszervezérlő többcsatornás autonóm átvitel módokra képes; az egyes csatornák átvitel módját és átviteli paramétereit a számítógép programozási uton tölti be a vezérlőbe.

A többágu rendszer általános jellemzője, hogy parancsai BrCNAF alakúak. Adott esetben a nyolc ág kiválasztására 3 bites Br kód szükséges. Az így adódó 20 bit széles parancsok átvitele itt nem okoz nehézséget, mivel a CDC 3100 tip. számítógép csatornájának adatvonala 24 bit széles.

A rendszervezérlőnek összesen 16 autonóm csatornája van. A csatornákat LAM jelek indítják. Az osztályozott LAM vektor 24 bitjéből így 16-ot az autonóm csatornák foglalnak le, 7 a kereteket identifikálja és 1 a rendszervezérlőhöz tartozik. A

keretek 7 és a rendszervezérlő 1 LAM-jele a számítógép egy megszakítási szintjéhez tartozik.

A számítógép-rendszervezérlő kapcsolat 12 megszakítási szintet ölel fel: áganként egy-egy (összesen 8), továbbá a READY, END OF BLOCK TRANSFER, ADDRESS OVERFLOW és ABNORMAL END OF OPERATION. A megszakítási szintek többszörös hierarchiája miatt a megszakítást kérő forrás azonosítása elég hosszadalmas. Az ilyen terjedelmes rendszereknél ezért különösen előnyös, ha az autonóm átvitelt a LAM jelek közvetlenül indítják.

Az általánosabb jellegű rendszervezérlők CAMAC keretben elhelyezkedő funkcionális blokkokból épülnek fel, és egynél több programforrással való kapcsolatot is lehetővé tesznek. A rendszerkeretként szabványos CAMAC keretet használó vezérlők tipikus képviselője [30]; a B (Busy=foglalt) vonal két állapota ezeknél a keret két állapotát definiálja. B=1 a normál állapot, melyben a keret vezetékezése a 15. ábrán leírt módon funkcionál; B=0 pedig a rendszer-állapot.

Rendszerállapotban a szabványoshoz képest változatlan marad a Z, I, X, S1 és Q funkciója, a többi vezeték feladatköre azonban megváltozik, éspedig:

a W1...W24 vonal kétirányúvá válik, azaz nem csak írásra, hanem olvasásra is szolgál. Az R-vonal új feladatai:

- R1-R20: A BrCNAF parancs átvitele
- R21 : Többszörös címzés mód; jelölése: M (Multi addressing)
- R22 : Osztályozott LAM kérés; jele G
- R23 : Q és az olvasott adatok átvitelével kapcsolatos; jele S
- R24 : Megfogó vonal; jelölése: H (Hold).

Az adatut C vonala a H vonal visszaállítására szolgál. Az L vezeték egyrészt a programforrások, másrészt az ághajtók jelentkezésére, az N vezeték a programforrások indítására szolgál. Az S2 vonalnak egy a szövegben még alább értelmezett visszajelentési feladata van.

A rendszerkeretben, legáltalánosabb esetben, négy különböző funkciójú blokk található:

1. R-vezérlő.
2. Ághajtó.
3. Programforrás.
4. Normál CAMAC modul.

R-vezérlő mindig csak egy van, míg az ághajtók száma 1-7-ig terjedhet. Programforrás egy vagy több lehet (a max. számot [30] nem rögzíti). A keretben elhelyezhető CAMAC modulok száma az adott konfiguráció esetén rendelkezésre álló hely nagyságától és az elhelyezendő modulok méreteitől függ.

A rendszerkeretben a CAMAC modulok jelenléte csak lehetőség, de nem kötelező. Ha CAMAC modulok jelenlétével eleve nem kell számolni, akkor az R-vezérlő funkciója a prioritás arbiter (PA) szerepkörére korlátozódik.

A keretvezérlő programforrásai lehetnek külső számítógépek, melyeknek csupán csatlakoztató egységei foglalnak helyet a rendszerkeretben, lehetnek olyan CAMAC processzorok, amelyek az előre beléjük tárolt információ alapján maguk generálják a CAMAC parancsokat, szolgáltatják, ill. fogadják az egyes átviteli ciklusokhoz tartozó kimenő-, ill. bemenő adatokat, lehetnek azonban komplett mikrogépek is. A "számítógép" helyett e helyen azért célszerűbb a "programforrás" kifejezést használni, mivel az a szituáció is elképzelhető, melyben programforrásként olyan CAMAC processzorok szolgálnak, melyek külső oldalukkal egyetlen számítógépre csatlakoznak. Egyetlen szorosabb értelemben vett számítógép van tehát csupán, a programforrások száma mégis egynél több.

Mi az a lényeges különbség, mely a többcsatornás autonóm ágvezérlő és a fent vázolt összeállítású rendszerkeret között mutatkozik? Az ágvezérlő egyetlen ág több adatforrását, ill. adatnyelőjét multiplikálja egy számítógépre, és egy számítógép parancsait irányítja egyetlen ágra (a parancsok végső soron mindig a számítógéptől származnak). A rendszervezérlő több ág adatforrásait, ill. adatnyelőit multiplikálja több program-

forrásra (általános esetben több számítógépre) és több programforrás CAMAC parancsait multiplikálja több ágra.

A programforrás "hálózati szolgálatra" a keret B=0 állapotában a hozzátartozó L vezetéken át jelentkezhet. Az L és N vezetékek a keret (jobb szélső) pozícióin elhelyezkedő R-vezérlőhöz futnak be, ill. onnan ágaznak ki. Általában egyidejűleg több programforrás jelentkezhet hálózati szolgálatra. A programforrások prioritási rendbe sorolódnak és az R-vezérlő mint prioritás arbiter dönti el, melyikük jelentkezését fogadja el. Amelyiket elfogadja, annak N vezetékeire MEHET jelet küld ki. Az N jel hatására az előnyben részesülő programforrás a keret R vonalának 1-20. számú vezetékegyüttesére bocsátja a BrCNAF parancsot (s ha szükséges a kiviendő adatot a W-vonalra). Ha az üzemmód egyciklusú, a parancs végrehajtásával a művelet le is zárul, s az adott programforrás L jele törlődik, de ha a programforrás egy ciklusnál hosszabb ideig kíván szolgálatban maradni, a H jelet állítja, amivel monopolizálja helyzetét.

A BrCNAF parancsot - végrehajtás előtt - az R-vezérlő értelmezi. El kell ugyanis döntenie, hogy ágra viendő, ill. a rendszerkereten belül fekvő CAMAC modulnak, netán magának az R-vezérlőnek szóló parancsról van-e szó. Ha a parancs ágra viendő, azt az R-vezérlő, az esetleges adattal együtt, a Br értéke által kijelölt ágra (ághajtón át) S2 jellel kapuzza. A művelet lefutását egy az ághajtó L kimenetére közösített BTB parolajel jelzi. A BTB jel az ághajtótól az R-vezérlőre fut, egyúttal azt is jelezve, hogy a Q válaszjel is beérkezett. Erről az R-vezérlő az S jel kibocsátásával értesíti az aktiv programforrást. Néhány adminisztratív lépés után, mely a parolajelek visszaállításából és a MEHET jel visszavonásából áll, az átviteli ciklus lezárul, újra minden nyitottá válik, és a prioritás arbitrációtól kiindulva ismételhető, hacsak a szóban forgó programforrás H jellel hosszabb szolgálatra le nem foglalja a rendszerkeretet.

Amennyiben az R-vezérlő a rendszerkereten belüli CAMAC modulnak szóló parancsot észlel, úgy még B=0 állapotban először ő maga veszi át a parancsot, majd normál keretvezérlővé kap-

csolva át magát, B=1 keretállapotban közvetíti a parancsot a CAMAC modulnak. Ezután ismét B=0 állapotban végzi el a záró-ciklust az aktiv programforrással.

Ha több keretnek szóló parancsot ad ki valamelyik programforrás, akkor az érintett ághajtó TÖBB-KERET regiszterét először egy speciális parancs útján fel kell töltenie a kiválasztott kereteket kijelölő bináris vektorral. Az M(1) jel kíséréteiben kibocsátott parancs ezek után az összes, a TÖBB-KERET regiszter bitjei által kijelölt, keretben végrehajtódik.

A rendszervezérlő a CAMAC hálózat különböző ágairól jelentkező igényeket mindenekelőtt az egyes ágak BD jelei útján érzékeli, mely jelek az ághajtókra futnak be. Az ághajtók legalább is két modulszélességűek, és legalább két L vezetékük csatlakozik az R-vezérlőre. Az egyik L vezeték, mint jeleztük, a közösített BTB parolajel foglalja le, a másik szolgál a BD jelnek az R-vezérlőhöz való továbbítására. A rendszerkeretben lévő CAMAC modulok L kimenetüket normális módon használják. Az ágakról és a helyi CAMAC modulokról érkező igények, az R-vezérlőben rögzítődnek. A jelek prioritáskódolóra kerülnek, mely - egyidejűség esetén - a legnagyobb prioritású igénylő azonosító-kódját - címmutatóját - B=0 keretállapotban a rendszerkeret adatutjára küldi. Ezt a globális címmutatót az összes programforrás megvizsgálja. Közülük az amelyiknek szól, azt ágcimregiszterébe feljegyzi. Ez a (második) cimregiszter a CAMAC parancs Br címét hordozó regisztertől független. Hogy az igénylő címmutatójával mi történik, az függ a programforrás felépítés- és működésmódjától. A cím indíthat egy számítógépi megszakítási rutint, de kezdeményezhet a rendszerkereten belüli autonóm akciót is. Mindkét esetben a következő lépés célja az, hogy osztályozott LAM vektor álljon rendelkezésre az igényt bejelentő ágról. Ennek érdekében az érintett programforrás, a maga L-vezetékén át, szolgálatra jelentkezik, amelynek elfogadása a már leírt módon történik. Miután jelentkezésére a MEHET jelet megkapta, a GBr(x) vezérlő szavat küldi ki, ahol is a G elem a B=0 rendszerállapotu R22 adatut vezetéken fut ki. Az így megcímzett ághajtó végrehajtja az osztályozott LAM műveletet, behiva az

osztályozott LAM vektort, mely ugyanugy vivődik át a programforráshoz, mint olvasás során az adat. Az osztályozott LAM vektor a már leírt kezelésmódok valamelyikével értékelődik ki; bitjei indíthatnak direkt akciót, és létrehozhatnak számítógépi megszakítást is. Egyes bitek tartalmának további értelmezése lekérdezgetéssel végezhető, a CAMAC eljárásoknak megfelelő módon.

Mint korábban említettük, vannak olyan rendszervezérlők is, amelyek ugyancsak szabványos CAMAC keretben helyezkednek el, de amelyek az adatut vonalain kívül, néhány kiegészítő vezeték is használnak. A GEC-Elliott [31] rendszer keretében alkalmazott négy külön vezeték pl., mely a programforrásokat az R-vezérlővel köti össze, arbitráció és rendszerállapot meghatározása céljára szolgál. Itt, a források nem az L pontokon, hanem egy közös vezetékre adott jellel jelentkeznek szolgálatra, és prioritásukat egy gyöngyvezetékre (daisy chain) felfűzött sorrendjük szabja meg; e vezeték adja meg a kiválasztódó programforrásnak a munkára való engedélyt is.

Feltéve, hogy az engedélyt kapott programforrás a rendszerkereten belüli CAMAC modult címez meg, (Br=0), a BrCNAF parancs a rendszerállapotu adatuton át először az R-vezérlőben tárolódik. Rendszerállapotban az adatut használata itt sem szabványos. A program-elemek közül A és F ugyan közvetlenül az adatut A és F vonalaira kerül, de BrCN a W-vonal vezetékein fut át az R-vezérlőre. Br=0 hatására az R-vezérlő az adatut CAMAC állapotba kapcsolja át B=1 és N kibocsátásával. B=1 hatására egyrészt az R-vezérlő összeköti az R és W adatvonalat, másrészt az aktív programforrás a W-vonalról visszavonja az BrCN parancselemeket. Ha olvasás műveletről van szó, az R vonalon át a programforrás közvetlenül veszi a kiolvasott adatot, ha írás művelet hajtódik végre, akkor pedig az R vonalon és az R-vezérlőn át a CAMAC modulba irandó adat a W-vonalra fut. Az R-vezérlő ebben a vonatkozásban ellátja a CCA-1 minden funkcióját: a CAMAC modulok a szabvány szerint csatlakoznak, s nem érzékelik, hogy rendszerkeretben vannak. S2 jellel végződik a ciklus, és ezzel együtt az R-vezérlő felszabadítja a rendszerkeretet, ő maga pedig kész a további szolgálatra.

A két modulszélességű ághajtók a rendszerkeret előre meghatározott helyein foglalnak helyet. Hogy az arbiter által kiválasztott programforrás BrCNAF parancsa ezen ágvezérlők valamelyikének, ill. ezen át a hálózatban kint fekvő CAMAC modulnak szól-e, azt az R-vezérlő az $1 \leq Br \leq 7$ érték alapján ismeri fel. Külső - ág - ciklusban a keret az R-vezérlő B jele után is rendszerállapotban marad, s a programforrás is fenntartja a W-vonalon a BrCN parancselemeket (természetesen A és F-t is). A megcímzett ágat (ágvezérlőt) Br dekódolásával a rendszerkeret N vezetékeinek egyikével B jel kíséretében az R-vezérlő választja ki, mintegy az ágra kapuzva a CNAF parancsot. A rendszerkeret R vonala ez esetben éppugy kétirányú, mint az ág adatvonala is. Adatátadással járó művelet során ezen át halad, a parancs által kijelölt irányban a programforráshoz vagy a forrástól, az adat. Az ághajtókban itt is van TÖBB-KERET regiszter, melynek feltöltését a programforrás végzi el, s melynek tartalmától függően egyvalamely parancs egynél több keretben is végrehajtható.

Az adat be- ill. kivitel (Q és X) adminisztrálása nagyban hasonló módon történik, mint [30] esetében,

A programforrások igen változatos módon komponálhatók. Erre, s az adat valamint LAM kezeléssel kapcsolatos lehetőségekre is példaként, most a GEC-Elliot rendszervezérlő egy konkrét alkalmazásmódjával ismerkedünk meg [49] (24. ábra). A rendszervezérlő a programforrás, ill. programforrások kialakíthatósága tekintetében, bizonyos rendelkezésre álló blokkok felhasználhatósága révén, igen flexibilis. A rendszeréhez NORD 10 típusú számítógép csatlakozik. A NORD 10 mikroprogram architektúrájú 16 bit szélességű, max. 64K, virtuálisan 256K memóriakapacitású számítógép, melynek 16 prioritási szintje van, s minden szinthez egy-egy önálló regiszterkészlet tartozik, ami az egyik szintről valamely másikra való gyors átkapcsolást tesz lehetővé. A gépnek 2048 készülékcíme és ugyanannyi vektoriális címmutatója van.

A NORD 10 programkapcsolatu csatlakozására a rendszerkeret PIO egysége, az egyes ágakon végrehajtott osztályozott LAM ké-

rés automatikus lebonyolítására IVG vektorgenerátor szolgál (BG ciklus). Az IVG a kapott bináris LAM mintázatokat prioritáskódolással 8 bites megszakítási címmutatókká transzformálja, s maszkolás is általa végezhető. A PIO egység programozott átvitel módban CAMAC parancsokat és megszakításkéréseket kezel. Háromféle megszakítási szint lehetséges.

1. LAM eredetű
2. NIM eredetű
3. Hiba eredetű.

A NIM eredetű megszakításkérések az általában a NIM egységként megépített gyors triggeráramköröktől származnak, és közvetlenül hatnak a számítógépre. Ésszerűtlen lett volna ezeket a jeleket a CAMAC LAM hálózatba terelni. A hiba eredetű megszakításkérés főként Q és X eredetű.

A számítógép 16 programszintje közül négy külső vektoriális megszakításmóddal érhető el. A különböző PIO megszakítási szinteknek e programszintekkel való összekapcsolása, a PIO egység erre szolgáló vezérlő regisztereinek előzetes programozása révén dinamikusan végezhető.

A PIO egység minden regisztere feltölthető és kiolvasható programozási uton. Ez, valamint az a körülmény, hogy megszakítási jelek is program útján szimulálhatók, elősegíti az egységek és számítógéppel való kapcsolatuk ellenőrzését.

A DMA egység a számítógép memóriájával való közvetlen autonóm forgalmat bonyolítja le. A DMA egység belső regisztereinek címzése és megszakításkérésének kezelése a rendszerkeret adatútján át történik, s ezáltal nem csak a NORD 10, hanem más programforrás által is előkészíthető. A DMA egységnek nyolc különböző üzemmódja lehetséges, mik közül egy-egy, erre rendelt üzemmód-regiszter tartalmának beállításával, választható.

A háromszoros modulszélességű egység kétféle LAM jelet állíthat. Az egyiknek jelentése: "minden rendben", azaz valamely átvitel sikeresen lezárult, és az egység ismét indítható. Ha a soronkövetkező blokkátvitel az őt megelőző ismétlése, akkor a záró LAM el is hagyható, az átvitel pedig valamely start-impul-

zussal indítható. Ehhez az kell, hogy az átviteli paraméterek rendelkezésre álljanak minden kezdethez felújítási eljárás nélkül is. A másik LAM akkor áll elő, ha az átvitel során a DMA egység valami hibát regisztrál vagy az átvitel felfüggesztődik. Ezen a módon, ha egy puffer több blokkot vesz fel, az adatátvitelre kijelölt egész memóriaterület külön beavatkozás nélkül feltölthető, és a "minden rendben" LAM csak a puffer feltöltődésével állítandó. A kiegészítő információk - így a puffer fejléce, az esemény-cimkék vagy jelző szavak - a blokkok között automatikusan generálhatók a DMA egység, de valamely erre a célra szolgáló más egység által is. A rendszervezérlő kiegészülhet egy speciális CAMAC processzorral. Ilyenkor egy ún. "indirekt csatorna" mechanizmus funkcionál. A DMA a CAMAC parancssal 6 bites kódot állíthat a W-vonalra, ami a rendszerkeretben lévő összes blokknak jelzi, hogy valamely specifikus blokkátvitel van folyamatban. Eközben a jelenlévő CAMAC-processzor adatokat vehet fel az R vonalról a DMA egységgel együtt, és figyelheti ezek átvitelének menetét. Egynél több DMA egység használatával jelentős idő takarítható meg. Pl. az egyik egység valamely adatátvivő vonalat, míg a másik a kísérlet gyors adatgyűjtését szolgálja ki.

Valamely másodlagos számítógép jelenléte igen sok előnnyel jár. E másodlagos gép mint mikroszámítógép a rendszerkeretben foglalhat helyet. Egy ilyen gép még mindig csak adatátvitellel és igénykezeléssel kapcsolatos adminisztrációt végez, de nem végez az adatokon processzálást. Adott esetben a CAMAC-processzorként alkalmazott INTEL 3000 végzi az igénykezelést attól kezdve, hogy az IVG jelzi, hogy osztályozott LAM műveletet hajt végre. Az osztályozott LAM mintázat (LAM vektor) a rendszerkeret adatutjáról egyszerűen "felcsíphető". A mikroprocesszor alkalmazható a DMA egység paramétertömbjének feltöltésére, az adat-fejlécek és jelző szavak beszurására és a folyamatban lévő átvitel figyelésére abból a célból, hogy a hibás átvitelt leállíthassa. A processzor és a hozzá tartozó memória 24 bit szószélességű; a memória max. 32K szó kapacitású. A memória-vezérlőnek nyolc CAMAC B/K csatornája van, melyek ciklus-

elvéttel működnek; egy közvetlen csatornája szolgál a processzorral való kommunikációra és egy csatorna az indirekt csatorna-mechanizmust használó blokkátvitel adatainak felvételére. A saját memóriába történő blokkátvitel is a rendszerkeret DMA egységén át történik indirekt csatorna üzemmódban. A DMA kimenete egy adott vezérlőbit átállításával irányítható a külső számítógép memóriájától a CAMAC-processzor memóriájához.

A CAMAC-processzor LAM jelekkel vagy az előlapján lévő hat bemenet valamelyikére érkező NIM impulzussal, vagy egy a rendszerkeretben lévő másik programforrás útján szakítható meg. 24 prioritási szintje, automatikus megszakítási rendszere van. Tíz NIM kimenet szolgáltat állapot információt és "foglalt" jelet a kísérletnek.

A CAMAC processzorok általános felhasználási lehetőségei.
A nagykapacitású memóriával is rendelkező CAMAC processzort joggal lehetne CAMAC számítógépnek is nevezni. E processzor saját memóriájában tárolja nem csak a generálandó CAMAC programot, hanem a begyűjtött, ill. kiviendő adatokat is. Az ilyen processzor teljesen önállóan képes végrehajtani programokat. A kísérletek természete olyan, hogy azonos programot esetleg ciklikusan, hosszú időn át kell ismételtetni anélkül, hogy változtatására szükség volna. Így, elég csupán egy alkalommal, a kísérlet megkezdése előtt, az adott programot a külső programforrásból a CAMAC processzor helyi memóriájába átvinni. Ha bármi oknál fogva időközben mégis (a kísérleti feltételek megváltoztatása miatt) egy más programot kéne alkalmazni, ez az előző helyére egyszerűen betölthető. A processzor nem csak egy, hanem több feladat végrehajtásával is megbízható, s ennek megfelelően memóriájában az egyes feladatoknak megfelelő több programot tárolhat. Ilyen feladatok lehetnek pl.

- a gyorsító-impulzust megelőző rendszervizsgálat;
- a gyorsító impulzusa alatt az egyes események adatainak begyűjtése menetközbeni ellenőrzéssel;
- a gyorsító-impulzus végeztével a kísérő információ begyűjtése menetközbeni ellenőrzéssel;

- a gyorsító szünetében a fizikus által igényelt (és a számítógép által már kiszámított) mérési eredmények adatainak kivitele a CAMAC hálózat meghatározott pontjára.

Ez meglehetősen egyszerű feladategyüttes; előfordulnak ennél lényegesen bonyolultabbak is. Éppen az ilyen többfeladatu szituációhoz illeszkednek jól a többcsatornás CAMAC processzorok.

Arra való tekintettel, hogy a CAMAC processzor helyi memóriája véges kapacitású, adott esetben időről-időre azt üriteni kell. Adatgyűjtésnél pl. (a helyzettől függően) az adatpuffert vagy az egyes események adatainak begyűjtését követően, vagy a gyorsító-impulzus lezárultával kell üriteni, hogy az a következő adatgyűjtési ciklus rendelkezésére állhasson. Az ürités a külső programforrás (számítógép) memóriájába vagy valamely nagykapacitású tárolóba történhet.

Korábban említettük már, hogy az olyan mérsékelt autonómiájú vezérlő, amelyik a CAMAC utasításokat a számítógépi memória közvetlen csatornáján át generálja és nem rendelkezik helyi adatpufferrel sem, csupán ágvezérlőként vagy számítógéphez direkt csatlakozó U-vezérlőként alkalmazható, az ágra, ill. hurrokra csatlakozó keretvezérlőként azonban nem, ami közvetlenül belátható. Nem ez a helyzet a fejlett, nagykapacitású helyi memóriával rendelkező CAMAC processzorokkal, melyek nem csak az önálló programgenerálás, hanem a közbenső tárolás szerepkörét is elláthatják. Korábban a processzorok CCA-1 vezérlő helyett való alkalmazását költségességük nehezítette. Az árak csökkenésével ma a keretenkénti CAMAC processzorok alkalmazása nem ütközik akadályba.

A keretben helyetfoglaló CAMAC processzor az adatgyűjtéssel és -kivittel, de számos más helyi funkcióval, így pl. igénykezeléssel kapcsolatos feladatokat is elláthat, lényegesen leegyszerűsítve az ágvezérlő, a rendszervezérlő és a számítógép CAMAC műveletekkel kapcsolatos adminisztrációját. A CAMAC processzornak keretvezérlőként való alkalmazása azzal az előnnyel is járhat, hogy az adatgyűjtés a keret által megengedhető maximális sebességgel és több keretben szimultán történhet, az

adatpufferok kiürítése (pl. a gyorsító szünetében) a hálózat adottságai által korlátozott, lassabb tempóban, az egyes keretek puffereinek egymás után való kiürítésével végezhető. Különösen előnyös lehet keretvezérlőként CAMAC processzort alkalmazni soros hálózatban, ahol a soros főút átviteli sebessége lényegesen kisebb az adatuténál.

4.3 ELOSZTOTT INTELLIGENCIA

Mindeddig csupán olyan vezérlőkkel foglalkoztunk, amelyeknek feladatköre legfeljebb a CAMAC parancsok önálló generálását, a CAMAC műveletek lebonyolítását és az igénykezeléssel kapcsolatos adminisztrációt foglalta magában, de amelyek a rajtuk áthaladó adatokon processzálást nem végeztek. Az adatok e vezérlőkből ugyanolyan alakban távoznak, mint ahogyan odaérkeztek. A legmagasabb szintű vezérlők, a CAMAC processzorok, egyes változatai megközelítik az időközben kialakult mikroprocesszorok, ill. mikrogépek bonyolultságát. Kézenfekvő volt arra gondolni, hogy a CAMAC vezérlők tervezéséhez mikroprocesszorokat használjanak fel, s hogy egyuttal ezek bőséges utasításkészletének birtokában helyi adatprocesszálást is lehetővé tevő megoldásokat alkalmazzanak. Az adatgyűjtés és -szétosztás végrehajtásmódjával szemben fokozódó követelmények a helyi számítástechnikai lehetőségek kihasználását nem csak hogy indokolják, de ki is kényszerítik. Ilyenképpen alakultak ki az egyes pontjaikon mikroszámítógépeket adatprocesszálásra is használó, elosztott intelligenciájú hálózatok.

Az elosztott intelligenciájú hálózatot alkalmazó kísérleti rendszer strukturája már nem egyezik meg a 2. ábrán felvázoltéval, mivel a helyi processzorok alkalmazása folytán a feldolgozó lánc elemei mintegy benyomulnak a kísérleti elektronika körzetébe; a kísérleti elektronika és a feldolgozó lánc átlapolják egymást.

Néhány dologra itt fel kell hívni a figyelmet: A kísérleti elektronika adatkezelő rendszere egyes esetekben lehet ugyan tisztán CAMAC, általánosságban azonban - ahogy azt már emli-

tettük is - a lassabb elektronikát összefogó CAMAC hálózat ki-egészül még egy, a gyors elektronikát összefogó, CAMAC-idegen (rendszerint NIM) hálózattal is. Amikor helyi intelligenciáról beszélünk, meg kell különböztetnünk a CAMAC mikrogépeket a gyors elektronikához tartozó vagy a gyors hálózat és a CAMAC hálózat közé iktató CAMAC-idegen helyi számítógépektől. A mikrogép és mikroprocesszor fogalmával is óvatosan kell bánnunk. Noha az elosztott intelligenciájú hálózatok gyors fejlődéséhez a kommerciális mikroprocesszorok elterjedése adta a legnagyobb lökést, e processzorok nem küszöbölték ki, és nem is küszöbölhették ki a kis-, közepes- és nagyintegráltságú áramkörök együtteseiből komponált speciális processzorokat, de még a korábban alkalmazott magasszintű CAMAC vezérlőket (CAMAC processzorokat) sem. A speciális processzorokra főként a CAMAC-idegen gyors hálózatban van szükség, ahol a követelmények az elvégzendő feladatok és a sebesség tekintetében egyaránt sajátosak.

Ami a CAMAC hálózatot illeti, egyrészt általában annak nem minden szintjén szükségeltetik intelligencia, másrészt az elmúlt évtized kommerciális mikroprocesszorai gyakran még a CAMAC hálózati igényeket sem elégítik ki (legalábbis az adatgyűjtés sebességi követelményei szempontjából), s ezért kombinatív megoldások is születtek.

A kommerciális mikroprocesszorra épülő, s a CAMAC rendszer keretein belül fekvő mikrogépek lehetnek ugyan keretvezérlők, ágvezérlők, hurokvezérlők vagy rendszervezérlők is, de nem mindig és nem szükségképpen vezérlők abban az értelemben, ahogyan azt a CAMAC szabványok korábban meghatározták. Egy-egy keret-
hez egynél több processzor is tartozhat, melyek mindegyike rendelkezhet saját memóriával vagy amelyek közül több rendelkezhet közös memóriával is.

4.4 CAMAC MIKROGÉPEK

A speciális, nagysebességű, CAMAC-idegen processzorok szerepkörével, felépítésmódjával és a mérőrendszeren belüli elhelyezésmódjával a jelen anyagban nem foglalkozunk. Egyelőre a CAMAC

rendszer keretei között maradvá, tekintsük át annak a folyamatnak az eredményeit, amelyet a nagy sorozatban gyártott kommerciális mikroprocesszorok, mikrotárák áramköreinek megjelenése váltott ki.

A CAMAC hálózatban modul-, keret-, ág (vagy hurok)- és rendszer-szintű helyi mikrogépek alkalmazhatók. Mikrogépről azért beszélünk, mert eleve feltételezzük, hogy az alkalmazott mikroprocesszor több-kevesebb memóriával egészül ki. A folyton változó fogalmak világában nehéz az eligazodás, ha nem ragaszkodunk a szavak, kifejezések értelmének rögzítéséhez. Korábban a legmagasabb szintű vezérlő elnevezésére a "CAMAC-processzor"-t használták, noha ezek logikai-aritmetikai egységeihez is tartozik - akár igen nagykapacitású - memória is. A "CAMAC-processzor" elnevezést az irodalom széltében használja, ne változtassunk hát rajta és tartalmán sem, de tudnunk kell, hogy az tulajdonképpen speciális mikrogép. Megkülönböztetésül, azokat az egységeket, amelyek akár CAMAC vezérlőként, akár csupán számítógépként helyi adatprocesszálást is végeznek, és amelyek ha nem is mindig, de jellemző módon kommerciális mikroprocesszor és mikrotár elemekből épülnek fel, CAMAC mikrogépeknek nevezzük. Összefoglalva tehát: A CAMAC processzor olyan speciális processzor- és tárelemekből komponált CAMAC vezérlő, mely csupán CAMAC műveleteket végez, de a rajta áthaladó adatokon processzálást nem hajt végre.

A CAMAC mikrogép mikroprocesszorból és mikrotárból felépülő számítógép, amely adatprocesszálást is végezhet, s ha a CAMAC vezérlő szerepkörét tölti be, úgy CAMAC műveleteket is végrehajt.

A különbségtételt nehezítheti, ha pl. CAMAC mikrogépet CAMAC processzorként alkalmaznak. Az osztályozásnál azonban nem az a mérvadó, hogy az adott egység adott esetben mit végez, hanem, hogy mik a tulajdonságai, mire képes.

Az elmúlt évtized során megvalósított számos CAMAC mikrogép noha egy technika előfutárjaként jelenik meg, a részecskefizika szempontjából teljesítmény és gazdaságosság tekintetében aligha nevezhető előrelépésnek a már más eszközökkel elért



szinvonához képest. E mikrogépek többségének alapegysége, autonóm vezérlőként, a keret vezérlőpozícióit foglalja el (általában két vagy három pozíciót), de sem a CAMAC főutak valamelyikére, sem U-vezérlőként valamely külső számítógép B/K csatornájára való csatlakozást nem tesz lehetővé [50-54].

Ha az alapegység kellő kapacitású memóriát is magában foglal, és írógép-csatlakozással is rendelkezik, alkalmas egyetlen keret által felölelt olyan mérőrendszer szükségleteinek ellátására, amelyik felsőbb szintű számítógépes feldolgozást nem igényel. Baj csak akkor van, ha - amint ez megesik - e mikrogépeket nagy mérőrendszerekben kívánják alkalmazni. Mi szükséges ahhoz, hogy nagy rendszerben alkalmazhassák? Mindenekelőtt bővíteni kell a tárkapacitást a keretben elhelyezett memória modulokkal, alkalmazni kell néhány periferikus készüléknek ugyancsak a keretben helyet foglaló csatlakozó egységét, és növelhetővé tenni a mikrogép irányítása alatt álló CAMAC modulok számát is további keretek hozzákapcsolásával, vagyis főutat és főut-vezérlő egységet, s ez utóbbin belül, vagy e mellett, blokkátvitelt lehetővé tévő (DMA) elemeket is kell alkalmazni. Esetleg lehetővé kell tenni valamely felsőbb szintű számítógéphez történő csatlakozást is. Ilyen módon alakult ki pl. a 25. ábrán felvázolt rendszer [50]. Az ilyen irányú fejlesztés eredménye: a korábban a kereten kívül álló kisszámítógépnek CAMAC keretben való elhelyezése; a hálózat struktúrája teljesen azonos a 17.a. ábrán bemutatott klasszikus hálózatéval. Jelent-e azonban bármilyen előnyt is az, hogy kommerciális kisgépeket (vagy azok készen kapható mikrováltozatait) a laboratóriumban épített mikrogép váltsa ki. A válasz e kérdésre aligha lehet igenlő. Akár a DEC gépcsaládjait, akár más, már hosszabb időn át kialakult, gyártott és időről-időre korszerűsített gépcsaládot nézünk is, annak kiforrottsága, megbízhatósága, software ellátottsága, készülék-vezérlőkkel való felszereltsége - ez utóbbiak soraiba számítva a különböző CAMAC vezérlőket is - sokkal magasabb színvonalu, mint ami a fizikai laboratóriumban ad hoc elérhető. A felsorolt mikrogépek, de további, némely szempontból esetleg többet tudó változataik

[55-57] sem alkalmazhatók a sokkeretes CAMAC hálózatban elosztott intelligenciát nyújtó elemként. A kisgépek velük való kiváltása pedig a részecskefizika területén sem technikai, sem gazdaságossági szempontok alapján nem indokolt. Ami a gazdaságosságot illeti, bármennyire olcsók is a mikrogép-áramkörök, a házilag létrehozott általános célu mikrogép bizonyára még mindig lényegesen többbe kerül, mint a sorozatban gyártott. Az ilyen duplikációk helyett, a laboratórium anyagi és szellemi erőforrásait célszerűbb a külvilágban még meg nem oldott, de a kísérletekhez mindenképpen szükséges speciális feladatokra összpontosítani, pl. olyan mikrogép változatok kidolgozására, amelyek többprocesszoros rendszerekbe szervezhetőek.

Mostanság gyakorta felmerülő kérdés, vajon megmaradhatunk-e a CAMAC hálózat keretei között, vajon a továbbiakban is képes-e az kielégíteni a kísérleti fizika fokozódó követelményeit. Anélkül, hogy a CAMAC rendszer nehézségeit és jövőjét már itt taglalni kívánnánk, annyit kétség kívül meg kell állapítani, hogy egyes kísérletek során a CAMAC bizonyos olyan korlátokat állított, melyeket csak módosításával vagy kiküszöbölésével sikerült elhárítani. Ez azonban még nem ok arra, hogy visszatérjünk a CAMAC-ot megelőző kaotikus állapotokra, esetleg éppen a CAMAC zászlaja alatt létrehozva olyan egyedi megoldások sokaságát, amelyek a mechanikai méretek azonosságán túl gyakran már aligha tekinthetőek CAMAC-nak. CAMAC mikrogépekről szólván, azokat a megoldásokat fogadhatjuk el CAMAC kompatibilisként, amelyek a CAMAC szabványokkal és ajánlásokkal valóban összeváganak, amelyeknek alkalmazása tehát nem teszi lehetetlenné CAMAC hálózat szervezését, az alsóbb szinteknek a felsőbb szintekhez szabványos CAMAC protokollok szerinti vagy U-vezérlőként való csatlakoztatását, s amelyek nem zárják ki a már eddig elért eredmények hasznosítását sem. A CAMAC mikrogépek tekintetében két, a fenti követelményeknek megfelelő megoldás látható jogosultnak, a mikrogéppé bővített, de a CAMAC főutakra reákapcsolható keretvezérlő és a többszörös vezérlésű keretrendszer.

A mikrogép-keretvezérlő feltételezésünk szerint egyetlen keretet szolgál ki, s csatlakoztatható a párhuzamos vagy - al-

ternative - a soros főútra. Felépítés- és működésmódját, valamint a keretben való elhelyezkedésmódját a 26.a. ábra alapján világítjuk meg. A mikrogép a vezérlő pozíciókat elfoglaló, általában bizonyos alapkapacitású tárt is tartalmazó processzoregységből és a hozzá társuló operatív memóriából áll. Az utóbbi moduláris felépítésű, s az adott szükségletnek megfelelően bővithető lehet. Feltételezzük, hogy a keretben ez az egyetlen mikrogép, mely egyúttal a keretvezérlő szerepkörét is ellátja. Célszerűen, a mikrogép az adatutatót csak adatutóként használja, a processzort és a memóriát járulékos sín köti össze, mely elhelyezhető akár a keret hátrészen, akár a frontján. Bármely periferikus készülék (talán a soros csatlakozású írógép kivételével) az adatutóra CAMAC modulként csatlakozik, s a mikrogéppel az adatutó protokollja szerint kommunikál. Ismeretesek olyan mikrogép változatok is, amelyek a rendszerkeretnél kidolgozott megoldásokat szem előtt tartva, az adatutó kettős állapotát definiálják: egyik állapotban az adatutó vezetékeit adatutóként, a másik állapotban a mikrogép belső sinjeként használva fel. Mivel az ilyen megoldás adott esetben a mikrogép által nyújtható lehetőségeket zsugorítja, alkalmazása nem célszerű.

Az imént leírt szervezésű mikrogép lehetőségei ideális esetben a következők:

- tetszőleges CAMAC programszegmens generálása
- ismétléssel végzett parancsgenerálás, adatátvitel a korábban ismertetett blokkátvitelmódok valamelyikében
- parancsgenerálás LAM jelek indítványozása alapján
- parancsgenerálás a felsőbb szintről jövő rendelkezés alapján
- egyes CAMAC parancsok kiadása program útján
- egyes felsőbb szintről érkező CAMAC parancsok végrehajtása
- nem adatátvitelt kezdeményező LAM jelek feldolgozása az igénylők prioritási előírásnak megfelelő azonosítása útján
- CAMAC protokollnak megfelelő kommunikáció a főúttal a főúton át magasabb szintű számítógépekkel

- modul-modul adatforgalom lehetősége.

A mikrogépnek lehetnek ROM tárban rögzített állandó programjai vagy programjai helyi készüléken (pl. írógép) vagy felsőbb szintről tölthetők és cserélhetők. A mikrogép a felsőbb szintről esetenként szubrutinokat hívhat (pl. ellenőrző rutinokat, matematikai rutinokat stb). Feltételezzük, hogy a mikrogépnek olyan DMA elemei is vannak, melyek a gyors adatbegyűjtést és a különböző (esetleg többszörös) blokkátviteli módokat lehetővé teszik. A kettős - programozott és programfüggetlen - kapcsolat megengedi, hogy míg a DMA utvonalon gyors adatbegyűjtés folyik, a processzor egyidejűleg már korábban begyűjtött adatokat processzáljon. A lehetőségek skálája igen széles. Amennyiben a mikrogép egységek a keretnek túl sok pozícióját foglalják el, egyszerű sin-ismétlővel egy második számú keret az előzőhöz kapcsolható azzal a megkötéssel, hogy a mikrogép által kezelt modulok N és L pontjainak száma a két keretben együttesen nem haladhatja meg a 23-t. A sinismétlőt célszerűen ugyancsak a processzor-egységben kell elhelyezni, miáltal az N és L vezetékek átvitele nem jelent nehézséget (l. 26.b. ábra).

A CAMAC mikrogépnek lehetőleg 24 bit szószélességgel kell rendelkeznie, s erre a célra általában a mikroprogramozott, bitszeletelésű kommerciális mikroprocesszor-típusok a legalkalmasabbak. Már a 12 és 16 bit szószélességű gépek, de még inkább a byte adatszélességű INTEL 8080 típusu, ill. ehhez hasonló mikrogépek CAMAC egybevághóságának hiánya, a gyakorlatban akut és műfogásokkal is legfeljebb mérsékelhető nehézségeket okoz.

A fent körülírt mikrogéphez hasonló rendszert ír le párhuzamos főút esetére [58], soros főút esetére pedig [59], bár az általunk megfogalmazott ideálistól több szempontból elmaradnak.

A mikrogép-keretvezérlő bizonyos értelemben aktualitását veszítette a többszörös vezérléssel rendelkező CAMAC keret-szabvány kidolgozását követően, mivel ez utóbbi az előbbit, jobban mondva az előbbi által nyújtott lehetőségeket magában foglalja.

4.5 CAMAC KERET TÖBBSZÖRÖS VEZÉRLÉSSEL

Az eredeti EUR 4100e dokumentummal meghatározott CAMAC keretben csupán egyetlen vezérlő foglalhat helyet. A külvilággal való kommunikáció, de a keret két modulja közötti kommunikáció is csak e keretvezérlőn át történhet. Más szóval, az adatutató a keretben helyet foglaló elektronikus blokkok közül mindig csak a keretvezérlő uralhatja. Már a CAMAC alkalmazásának első éveiben kialakultak azonban olyan megoldások is, amelyek lehetővé tették egyetlen keretben több programforrás alkalmazását. Sokáig az ezen a téren elért eredményeket csak a rendszerkeret szintjén hasznosították. A mikroprocesszor elterjedésével, az intelligens hálózat koncepciójának kialakultával azonban már nem csak egymás fölé rendelt CAMAC processzorokkal és hálózati mikrogépekkel kellett számolni, hanem egy kereten belül egynél több mikroprocesszor ill. mikrogép alkalmazásának lehetőségével is. E szempontok figyelembevételével az ESONE és NIM bizottságok kidolgozták az egynél több vezérlő alkalmazását lehetővé tévő CAMAC keret alapelveit, majd 1978-ban kibocsátották az ezen alapelveket összefoglaló szabványt is [60]. E szabvány olyan keretet használ, amellyel az eredeti CAMAC keret egybevág, de amelyik az utóbbihoz képest egy járulékos sinnel, az un. társvezérlő sinnel (Auxiliary Controller Bus - ABC) egészül ki. Ez annyit jelent, hogy ha e - vezetékezése szempontjából bővített - keretbe egy CCA-1 vagy SCC-L vezérlőt helyeznek el, és rajta kívül más vezérlő nincs, akkor a keret az eredeti CAMAC szabvány által definiált módon funkcionál. Ha viszont egynél több vezérlőt kívánnak alkalmazni, a vezérlő pozíciókon lévő keretvezérlőt ki kell cserélni más, a társvezérlő sint is kezelni képes egységgel. Ha a hálózat eredeti képét (adatut, fővonal) tartjuk szem előtt, akkor a vezérlő pozícióban lévő egység lehet CCA-1 vagy SCC-L-szerű vezérlő, kiegészítve a társvezérlő sint kezelő arbitrációs elemekkel. Ilyen arbitrációs elemekkel kiegészített vezérlő pl. a [60] leírásban definiált CCA-2.

A társvezérlő sin a következő vonalakkal áll (27. ábra):

- RQ (Request): Folyamodás
- RI (Request Inhibit): Folyamodás tiltás
- ACL (Auxiliary Controller Lock Out): Társvezérlő kizárás
- AL (Auxiliary-L): Társvezérlő L-vonal
- EN (Encoded-N): Kódolt N-vonal
- Átadó (Grant) vonal

E vonalak közül az első öt társas sinvezetékként (party lines) az átadó vonal pedig a társvezérlőkön áthaladó gyöngyvezeték (daisy chain) alakjában realizálódik. Az egyes vonalak vezetékszámát az ábra feltünteti.

A társvezérlő az RQ vonalon át folyamodik az adatutért. Egyidejűleg több vezérlő is folyamodhat, azonban jeleik egy közös jelként (logikai VAGY) lépnek fel az RQ vonalon. Az egyes vezérlők RQ jelei ugyanakkor az előlapjukon lévő FOLYAMODÁS jelű csatlakozó kontaktuson is megjelennek. A legmagasabb prioritásrangu vezérlőnek ez a FOLYAMODÁS csatlakozópontja össze van kötve saját ÁTADÓ-BE csatlakozópontjával, ÁTADÓ-KI csatlakozópontja pedig a prioritási rangban alatta álló vezérlő ÁTADÓ-BE pontjára lép, és így tovább, a 27. ábrán látható módon, ahol is a vezérlők közül pillanatnyi feltételezésünk szerint a keretvezérlő a legmagasabb prioritásrangu. A prioritásrang az adott esetben tehát $A \rightarrow B \rightarrow C \rightarrow D$ irányban csökken.

A közös RQ, jobban mondva most már ÁTADÓ-jel, a legmagasabb prioritásrangutól az alacsonyabb prioritásranguk felé haladva, az egyidejűleg jelentkezők közül a legmagasabb rangú vezérlőnél elakad, s egyszersmind e vezérlőt aktivizálja is. Az aktivizált vezérlő, RI FOLYAMAODÁS TILTÁS jelének kibocsátásával, lefoglalja a teljes sinrendszert, s mint ilyen a keretben vezető (master) vezérlő lesz, mely a többi - a sinen lévő - egységekhez parancsokat és adatokat továbbíthat, ill. azoktól adatokat fogadhat el. A FOLYAMODÁS TILTÁS jelet kibocsátó vezérlő törli saját jelentkezését, a többi vezérlőt pedig esetleges RQ jelének időszakos visszavonására kényszeríti. Amikor az aktiv vezérlő munkáját befejezi, visszavonja FOLYAMODÁS TILTÁS jelét; ezzel megszűnik a sinrendszer feletti uralma. Együttal a többi vezérlő időszakosan visszavont RQ jele ismét a FOLYAMODÁS vonalra lép, s a jelentkezők közül a legmagasabb rangú kiválasz-

tásával végződő arbitrációs ciklus megismétlődik.

Az egyébként vezetésre jogosult vezérlő a sinrendszert késleltetve kapja meg, ha oly időpontban folyomodik, amikor a sinrendszert egy másik vezérlő már éppen használja. Amennyiben az ujonnan jelentkező az adatfőúttal állna kapcsolatban, a főútnak e késéhez kell tudnia alkalmazkodni. Ilyen alkalmazkodó készsége van a paralel főútnak, ám a soros főút, amelyik saját időrendje szerint jár el, nem tud alkalmazkodni. Azok a keretvezérlők (főutak), amelyek nem akceptálják a késést, használják az ACL TÁRSVEZÉRLŐ-KIZÁRÁS vonalat. Ezt a vonalat egy kereten belül csak egyetlen vezérlő használhatja, de a vonal minden vezérlőhöz csatlakozik. Az ACL jelet vevő vezérlő, ha már szolgálatban van, saját munkáját vagy rögtön befejezi, vagy a folyamatban lévő műveletet elvetéli, s az ACL jelet kibocsátó vezérlőnek engedi át a keret sinrendszerét.

A Z és C műveleteket bármelyik vezérlő az adatut birtokbavétele után közvetlenül végezheti. Az I vonalra nem csak vezérlők, hanem modulok is bocsáthatnak ki tiltó jelet. Az N-t tartalmazó CAMAC parancsok társvezérlő által olyképpen hajtódnak végre, hogy a sinrendszert uraló vezérlő az R, W, A, F, S1, S2 vonalakat közvetlenül használja, az N cím képzése viszont a társvezérlő sin 5 vezetékes vonalára N kód kibocsátásával, s e kódnak a keretvezérlőben (CCA-2) való dekódolásával, tehát az N sugárvezetékek egyikére való leképzésével történik.

Nem csak a társvezérlők folyomodhatnak szolgáltatattételre, hanem hozzájuk is lehet szolgálatigénnyel fordulni. A keret LAM jeleit a keretvezérlő gyűjti össze, és azok a társvezérlő sin 24 vezetékes AL vonalain át jutnak a társvezérlőkhöz. Minden társvezérlő a hozzá rendelt LAM jelet v. jeleket kezeli, és saját LAM jele v. jelei alapján kezd kijelölt szolgálati feladatai elvégzéséhez.

4.6 CAMAC PROGRAMOZÁS, CAMAC NYELVEK

A korszerű számítógépek operációs rendszereinek B/K programjai felölelik mindazokat a rutinokat, amelyek segítségével az operációs rendszer automatikusan elláthatja a periférikus

készülékek működtetésével, az adatátvitellel és pufferolással, továbbá a megszakításkérések kezelésével és az állapotvizsgálatokkal, sőt egy magasabb szinten még a fizikai adatblokkoknak logikai előkészítésével, ill. a logikai strukturáknak fizikai adatblokká történő leképzésével kapcsolatos feladatokat is. A szokványos periférikus készülékek számára ezek a rutinok eleve rendelkezésre állnak, ami a magasabb nyelvi szinten dolgozó programozót teljes mértékben felmenti az egyes készülékek részletismeretének, a készülékek és adatok kezelésének aprólékos feladatai alól. Az operációs rendszerek felépítése általában megengedi a B/K rendszer moduláris bővítését, újabb, a géppel eredetileg rendelkezésre nem álló készülékek csatlakoztatását, ill. e készülékek B/K rutinjainak a már meglévőkhöz való hozzáfűzését. E rutinokat a rendszerprogramozónak egyszer kell kidolgoznia, s azt követően a számítógép használója részére - mintegy a készülék és a magasabbszintű programnyelv közötti kapocsként mindenkor rendelkezésre áll.

A CAMAC modulok a számítógép oldaláról nézve - periférikus készülékek, a CAMAC vezérlő vagy azok egy hálózaton belüli együttese pedig - a multiplexelés feladatát is ellátó készülékvezérlő. Multiplexelést is végző készülékvezérlőt a hagyományos készülékek körében is alkalmaznak; adott esetben egyetlen vezérlő látja el pl. egy csoport mágnesszalagos egység vagy diszk multiplexelését. Mindezek alapján kézenfekvő volna a gondolat, hogy a már kiformált elvek alapján minden egyes CAMAC modulhoz egy-egy készülékkezelő rutint rendeljenek. A gyakorlatban ez az út inkább csak kivételesen, de nem általánosságban követhető, aminek több oka is van. Az egyik ezek közül: az egy rendszerben alkalmazott modulok lehetséges igen nagy (esetenként több százra is rugó) száma. Egy másik: a modulok többsége igen egyszerű átviteli műveletet igényel, melynek individuális kezelésmódja programozási szempontból ésszerűtlenül terjedelmes volna. A legfontosabb ok azonban mindenképpen az, hogy gyakran a CAMAC modulok sora összefüggő adattömböt adó kontinuumot képez. Gondoljunk csak pl. a huzalkamrák által szolgáltatott adattömbökre. Ilyen esetben egy-egy B/K művelet nem egy, hanem

számos CAMAC modult érinthet. Problematikus az is, hogy az új és újabb CAMAC modulok megjelenése új és még újabb készülék-rutinok kidolgozását tenné szükségessé. Leginkább a fent felsorolt okok miatt a rendszerprogramozók figyelme a készülék rutinok helyett a CAMAC hálózatban lefolytatható akciók illetve ez akciók egységesítése irányába terelődött.

Az 'akció' CAMAC művelet(ek)et magába foglaló, de a műveletnél szélesebb körű fogalom. Valamely modul egy adatának a számítógép memóriájába való beolvasásához pl. szükséges egy CAMAC olvasás parancs - olvasás művelet - végrehajtása. A számítógép azonban, általános esetben, nem tudja ezt a parancsot egy lépésben végrehajtani, hanem (amint azt a CAMAC vezérlők ismertetésénél leírtuk) néhány utasításból álló programcskát - akciót - kell elvégeznie. Az akció és művelet közötti különbség még világosabban kitűnik, ha pl. a blokkátvitelre gondolunk. A blokkátvitel egyetlen akció, de benne számos CAMAC művelet hajtodik végre. CAMAC akció pl. egy adott modul adott igényének azonosítása és az igénynek megfelelő szolgálati rutinra való rátérés is.

CAMAC akción tehát általános értelemben a programozás szintjén végzett aktivitás magasabbszintű alapegységét értjük [61]. Az akció felölelhet egy vagy több műveletet; lényegében egy algoritmus, melyet a rendszer ismer, s mely a rendszer használója által aktivizálható.

A CAMAC akcióknak a rendszer használója által definiált rendezett sorozatát CAMAC szekvenciának nevezik.

A CAMAC rendszer igen rugalmas, megkötéseinek tág határai között igen sokféle különböző akció képzelhető el. Programozási szempontból mégis célszerű a lefolytatható akciók körének bizonyos mérvű korlátozása, s a korlátokon belül maradó akciók pontos meghatározása, mindemellett arra törekedve, hogy a rendszer felxibilitása se csorbuljon tulságosan.

Az ESONE és AEC NIM software munkacsoportok a 28. ábrán bemutatott CAMAC akciókat választották alapul [62]. Modul akciókat, LAM akciókat és rendszer akciókat különböztetnek meg. A modul akciók CAMAC modullal vagy modulokkal kapcsolatosak. Az

egyszeres modul-akciók lényegében egy-egy CAMAC műveletnek felelnek meg, egy-egy CAMAC parancsot hajtanak végre, és pedíg a CAMAC parancsok bármelyikét. Adott esetben a CAMAC parancsok három, azaz írás, olvasás és működtetés osztályba sorolódnak. Az írás osztályba az F(16)...F(23), az olvasás osztályba az F(0)...F(7), a működtetés osztályba az F(8)...F(15), ill. F(24)...F(31) feladatkóddal definiált parancsok tartoznak. Az állapotvizsgálat akció azokhoz a modul akciókhoz kapcsolódik, amelyek F(8) vagy F(27) feladatkódot aktivizálva Q értékét szolgáltatják. Végrehajtásának eredményeképp Q-értéke a memóriába vivődik át. Az írás, olvasás, működtetés akcióknak oly módosítai is lehetségesek, melyek az akció első része által kiváltott Q értékétől függően programelágazást hajthatnak végre.

Az egykészülékű blokkátvitel akciók egy CAMAC modulcím (BrCNA) és a memória adott területe között hajtanak végre blokkátvitelt (a modul vagy a memória igényába) a korábban ismertett egycímű blokkátvitelmódok valamelyikének megfelelően, azaz modulszinkronizált (Q), vezérlőszinkronizált, vagy LAM szinkronizált módon.

A többkészülékű blokkátviteli akcióknak két típusát definiálták. Az elsőhöz azok az akciók tartoznak, amelyek adott CAMAC regisztertömb és adott memóriaterület között hajtanak végre adatátvitelt, ill. adott CAMAC regisztertömböt "működtetnek". A második tipushoz a bizonyos CAMAC címtartomány végigjárásával végzett blokkátviteli, illetve működtetés-sorozat akciók sorolódnak.

A LAM akciók az egyes LAM igények engedélyezésére, felfüggesztésére, törörlésére és vizsgálatára szolgálnak. Ezek mindig egyszeres akciók.

Külön csoportot képeznek az ugyancsak egyszeres rendszerakciók, amelyek kizárólag a CAMAC vezérlőkre (keret-, ág-, hurok-, rendszer-vezérlők) irányulnak.

Minden akció egy-egy algoritmus, amelyet passzív CAMAC vezérlők alkalmazása esetén mindig program hajt végre, de amelyek közül egyeseket autonóm vezérlőknél részben vagy egészében, a vezérlő bonyolít le. Azon programok (rutinok) és vezérlők együt-

tesét, melyek a CAMAC akciók előirt készletének lebonyolítására szolgálnak csatolónak (coupler) nevezik [63].

Az egyes akciókat megvalósító (szoftver-hardver) algoritmusok végrehajtásához szükséges bizonyos specifikus paraméterek értékének ismerete is. Egy adat kiolvasása pl. csak akkor bonyolítható le, ha ismeretes az adatátadásban résztvevő CAMAC elem (BrCNA), ill. memóriahely címe, valamint az átviendő adat formátuma (16 v. 24 bit).

Az egyes akciók - szemantikailag - a lebonyolításában résztvevő számítógép és vezérlő(k) tulajdonságainak figyelmen kívül hagyásával írhatók le. Ezen az úton haladva olyan gépfüggetlen nyelv fogalmazható meg, melyben a CAMAC akciók és paramétereik szimbólikus nevekkel azonosíthatók. E nyelv utasításai egy-egy akciót jelölnek meg, argumentumaik pedig az akció paramétereit. Hasonló meggondolások alapján alkották meg a CAMAC IML (Inter-Mediate Language) un. "közbenső" nyelvet az ESONE és AEC NIM CAMAC software munkacsoportjai [62]. Eredetileg a gépi nyelv és valamely magasabbszintű nyelv közötti közvetítő nyelvnek szánták, pl. úgy hogy a felsőbb szintű CAMAC utasításokat a compiler először IML-re fordítaná. A gyakorlat nem követte ezeket az elképzeléseket, és az IML-nél magasabb szintű CAMAC nyelvet [64, 65] sem fogadta el.

Az IML nyelv négyféle utasítás típust ismer, melyek közül három a 28. ábrán felsorolt akciótípusoknak felel meg. Vannak tehát modul-akció, LAM-akció, és rendszer akció utasításai. A negyedik típus: a deklarációs utasítások. Ezeknek feladata a [62] definíciója szerint: "...hogy attributumokat rendeljen a (nyelvi) entitásokhoz, és hogy memória-területet rezerváljon futás-idő alatti reprezentációjához, ha ez helyénvaló". Lényegében a deklarációk konkretizálják az utasításokban szereplő szimbólikus paraméterek konkrét tartalmát, így pl. egy CAMAC regiszter vagy regisztercsoport, valamely memóriahely vagy memóriaterület címét és kiterjedését (ha ez utóbbi szükséges), egy névvel megnevezett LAM forrás identitását, stb. Az IML nyelvvel a már említett forrásdokumentumon [62] kívül egy sor más közlemény is foglalkozik, melyek közül az olvasó figyelmé-

be ajánlható [69] és [26]. Az előbbi: rövid bevezető, mely azonban jól használható annak ellenére, hogy a végső forrásdokumentumot időben megelőzve, attól bizonyos mértékben eltér. Az utóbbi a nyelvet részletesen ismerteti. A továbbiakban itt nem az IML leírásával, hanem inkább alkalmazási lehetőségeivel foglalkozunk.

Alkalmazásával kapcsolatban mindenekelőtt azt kell leszögezni, hogy az IML nem önálló programnyelv. Önmagában nem használható, csupán valamely más - vendéglátó - programnyelvhez társítva.

Az assembly nyelvekhez többféleképpen is illeszthető. Mindenek előtt az assembly nyelven irt programokban használható CAMAC akciók megjelölésére. A program elkészülte után azonban az utasításokat a programozónak kell kifejtenie és a szövegbe behelyettesítenie. A kifejtéshez, ami lényegében a kijelölt CAMAC akciók assembly nyelven való leírása, pontosan ismerni kell az adott számítógép és az alkalmazott CAMAC vezérlők architektúráját. Itt azokat a rutinokat kell létrehozni, amelyek a CAMAC vezérlőket CAMAC csatolóvá tágítják. E rutinok gépenként és vezérlő architektúránként más-más szerkezetűek. Olyan környezetben, ahol a CAMAC hálózathoz közvetlenül csatlakozó számítógép típusa és az alkalmazott vezérlők egységesíthetők, a CAMAC akciókat leíró rutinok is egységesíthetők. A programozói feladat kettéválasztható; rendszerprogramozó dolgozza ki a CAMAC akciók rutinjait és az assembly nyelvű programba való beültetésük szabályait, az egyes kísérleti rendszerek programozói pedig e rutinokat a megadott szabályoknak megfelelően ismételten beépíthetik. A programozók utóbbi, szélesebbkörű csoportja nem kell, hogy foglalkozzék a CAMAC akciók részletkérdéseivel.

A kísérleten assembly nyelven programozók munkája tovább egyszerűsödik, ha nem kell a rutinok beültetésével sem foglalkozniuk, ha az assembler a leirt deklarációkat és az akció utasításokat értelmezve, maga építi be a programba az akció rutinokat. Ez kétféleképpen történhet. Az első megoldás szerint: az egyes akcióknak megfelelő rutinok az assemblerben eleve rendel-

kezésre állnak, és IML utasítások észlelésekor beépülnek a programba. Ez a sajátosan CAMAC assembler esete [66]. E megoldás hátránya, hogy merev, az utasításkészlet bővítése csupán az assembler módosításával oldható meg, és a felhasználó szerkezeti szempontból is kötve van. Az assembler új makrókkal való bővítésének lehetőségét [67] ismerteti.

A második megoldás olyan assembler választása, mely makróutasítások definiálását és generálását egyaránt lehetővé teszi. Az egyes akcióknak megfelelő makrók ez esetben a konkrét CAMAC-számítógépi környezet figyelembevételével, programozási szinten definiálhatók, ezért ez a megoldás az előbbinél rugalmasabb. A feladat itt is szétválasztható, a makrók definícióját a CAMAC-számítógépi környezet részletismeretivel rendelkező rendszerprogramozó végezheti; a kísérlet felkészítése során e definíciókat a programozó - használó egyszerűen az általa írt programokhoz kapcsolhatja, s azt követően a definiált akció-utasításokat az előírt szabályoknak megfelelően szövegében használhatja.

Itt kell rámutatni arra, hogy az eredendően szemantikailag meghatározott IML-nyelv szintaktikai meghatározása a vendéglátó nyelvhez való adaptálása során, ez utóbbi szintaktikai szabályaival egyeztetve történik. Az IML egy makro szintaxisa a [62] leírás A) függelékében található. További makro implementációkat [70, 71] ismertet.

Mint ismeretes, az assembly makrók processzálása kétféle módon történhet. Alkalmazható külön makroprocesszor, mely egy előfutamban a makródefiníciókat értelmezi és a makrogenerálást elvégzi. Olyan kimenetet ad, amelyben már csak assembly nyelvű utasítások találhatóak. Ennek a makroprocesszornak lehetnek az assembly nyelvtől eltérő szintaktikai szabályai is, a lényeg az, hogy az általa generált szöveg az assembler szintaktikai szabályainak megfelelően. Amennyiben az előprocesszor beépített definíciókkal bír hasonlóan, mint pl. a [66] alatt idézett assembler, úgy ez a preprocessor egy magasabb szintű nyelvről alacsonyabb szintű (assembly) nyelvre való fordítást végez, s mint ilyen IML complier szerepkörét tölti be [68]. Már említett hátránya (mervesége) mellett ez a megoldás a ki-

sérlet felkészítésén dolgozó programozó munkáját minden esetre megkönnyíti. Továbbá, a speciális compilerbe több lehetőség építhető be, mint amivel az átlagos makro implementáció rendelkezik: formailag egyszerűsíthetők az IML utasítások, és az ellenőrzési és diagnosztikai adottságok is javíthatók, stb. [69].

A CAMAC rendszer programozásának egy másik megoldásmódja az akcióknak az eljárásorientált nyelvek magasabb szintjén való realizálása. Az implementáció a választott nyelv és a választott módtól függően többféle lehet, s a megvalósítások egymástól abban is eltérhetnek, vajon IML vagy attól eltérő jelölésmódot, ill. előírásokat követnek-e. Az idők során számos CAMAC nyelvi implementációt dolgoztak ki, hogy többek között itt csak a FOCAL [72] és a PEARL [73] implementációkat említsük meg, de az e téren végzett munkák zöme mégis leginkább két nyelv, a FORTRAN és a BASIC területére esik.

A CAMAC akciók implementációja compilert feltételező magasabb szintű nyelven kétféleképpen lehetséges. Az első: a magasabb szintű nyelv adott implementációjának CAMAC irányu kitágítása, a szintaxis és a fordítóprogram módosításával. A második: az akcióknak rutinokkal (eljárásokkal) való realizációja. A szubrutinokkal való realizációknak az az előnye, hogy nem kell a meglévő fordítóprogramot módosítani, a CAMAC implementáció nyelvi szinten végezhető, másrészt a memóriaterület felhasználása szempontjából is gazdaságos, mivel az egyes CAMAC rutinoknak csupán egy-egy kópiája szerepel a tárgyprogramban.

CAMAC akcióknak megfelelő FORTRAN szubrutinokat elsőként a NIM software munkacsoportja dolgozott ki körülbelül az IML nyelv létrehozásával egyidejűleg, de attól függetlenül [74,75]. Implementációjuk jellemzője, hogy minden egyes akcióutasításnak egy-egy szubrutin felel meg. A szubrutin paramétereit a szubrutint hívó utasításnak kell tartalmaznia. Egyes esetekben a fárasztóan számos paraméter jelenléte programozói hibalehetőségek forrása lehet. E megoldásmódnak van azonban más hátrányos vonása is. A paramétereknek a szubrutinokhoz való átadása és a hibavizsgálat is a futási idő alatt történik, ami jelentős idővesztést okoz, különösen egyszerű akciók elvégzé-

sének tényleges idejéhez viszonyítva. A különböző FORTRAN implementációk általában csupán GOTO, a CALL SUBROUTINE és RETURN utasításokkal rendelkeznek a program menetének irányítása céljára, s ezért az IML nyelvben szereplő Q-függő feltételes átadási utasítások (akciók), valamint a LAM kezelési utasítások FORTRAN kivitele probematikus, mivel a FORTRAN nem rendelkezik címkéknek egyik szubrutinból másikba való átvitelének vagy címkékhez való visszatérés lehetőségével, ami ez esetben szükséges volna.

A FORTRAN-nal kapcsolatos nehézségek áthidalásának egy módja lehet: az akció utasításokat külön kezelő processzor alkalmazása. IML nyelv használatát feltételezve tehát a program a vendéglátó nyelv és az IML utasításaiból állna össze, miközben az IML utasításokat a FORTRAN (vagy más vendéglátó nyelv) szintaxisától idegen jelölés különítené el [69].

Egy másik lehetőség: a NIM szubrutin szerkezet kiterjesztése kétszintűre úgy, hogy nem csak végrehajtási (akció) szubrutinokat, hanem a változók meghatározására és vizsgálatára szolgáló deklarációs szubrutinokat is alkalmaznak. Ez utóbbiak feladata, hogy előkészítsék a paraméterek azon memória blokkjait, amelyekre azután az akció utasítások hivatkoznak, egyúttal pedig hibavizsgálatot is végezzenek. E megoldás révén az akció utasítások egyszerűsödnek, és végrehajtási idejük is lerövidül. A deklarációs rutinokat csak egyszer kell végrehajtani, a program alkalmazásának kezdeti szakaszában, amikor a kísérleti rendszer felkészítése folyik [69, 76, 77, 78].

A felsőbb szintű nyelvek másik nagy csoportja: az interpretációs nyelvek. Közülük CAMAC realizáció szempontjából különösen a BASIC állt az érdeklődés homlokterében. A BASIC [79] egyszerűsége és könnyű kezelhetősége már hosszabb ideje magára vonta a mérés technikusok figyelmét [80, 81, 82, 83, 27]. A BASIC nyelven készült programot a számítógép nem fordítja le gépi nyelvre. A program forrásnyelvű utasításait egy a memóriában lévő interpreter fogadja, értelmezi és közvetlenül végre is hajtja. A BASIC interaktív nyelv, mely a programozó és a rendszer között szoros, élő kapcsolatot tesz lehetővé. Leglényegesebb jellemzői:

- A BASIC nyelven irt program utasításai előtt sorszám áll, és az utasítások e számok növekvő sorrendjében hajtódnak végre még akkor is, ha az utasítások nem a növekvő sorszámoknak megfelelően vannak "begépelve". Ezért utólagosan is iktatható be utasítás, ha sorszámát két utasítás sorszáma közé szurják. (A sorszám-inkrementum e lehetőség fenntartása érdekében >1-re választható, pl. 10-re).
- Nincs szükség összekapcsolási eljárásra, mivel a szubrutinokat az interpreter bocsátja rendelkezésre.
- Az interpreter szintaktikailag egyenként megvizsgálja az utasításokat, és ha szükséges, hibajelzést ad. Egyébként STOP utasítások közbeszurásával mód van időszakonkénti megállásra, és egyes programszakaszok átvizsgálására is.

A BASIC-nak különböző implementációi vannak. A HP BASIC programrendszer pl. interpreterből és előkészítő programból (Prepare BASIC System-PBS) áll. Az előkészítő program tartalmazza a periférikus készülékek kezelő rutinjait, továbbá azokat a rutinokat, melyek a kezelő rutinokat az interpreterrel kapcsolatba hozva HP BASIC-be szervezik. Mód van arra is, hogy a programrendszer törzséhez assembly nyelven írott szubrutinokat is lehessen kapcsolni. A későbbiekben ezek a járulékos szubrutinok CALL utasítással hívhatók, mely utasítások vonzatában (argumentum) a szubrutin azonosítóján túl, a szubrutin paramétereit találhatók. A CAMAC akciók realizációjának egyik lehetséges módja éppen ez, az adott gépi környezetet figyelembe vevő assembly szintű CAMAC rutinok összeállítása, és a BASIC rendszerhez kapcsolása. E rutinok a BASIC programokban azután a reájuk hivatkozó CALL utasításokkal aktivizálhatók

A NIM és az ESONE CAMAC software munkacsoportok által kidolgozott és CAMAC szabvánnyá emelt időreális BASIC [85, 86]. nem a szubrutin megoldás útján halad, hanem a B/K műveletekre, a megszakításkezelőre és a deklarációkra új utasításokat vezet be, s bitkezelés lehetőségével is rendelkezik. E nyelvi implementáció a CAMAC regisztereket változókként kezeli, rendelkezik Q és X vizsgálatának lehetőségével, általa időreális "ese-

mények" LAM-oknak megfeleltetve definiálhatók, tömbök kezelhetőek, blokkátvitel végezhető.

A NIM-ESONE BASIC-hoz hasonló tulajdonságu és feladatkörű a CATY nyelv is [87,88].

Az interpreter alkalmazásának igen nagy hátránya lassúsága, másrészt az interpretatív nyelvek meglehetősen szegényesek, nagyobb programok írására nemigen alkalmasak. Abban a körben ahol mégis alkalmazhatók, s ahol előnyös vonásaik hasznosan kamatoztathatók, törekszenek a munkasebesség fokozására. Alkalmazható pl. kétszintű nyelvi implementáció. Az első szinten a forrás-program egyszerű interpretatív nyelvre fordítódik, majd e közbenső programot interpreter hajtja végre. Egy másik lehetőség az interpreternek mikroprogram szintű realizációja. Harmadik megoldás: a programoknak interpreter környezetben való előkészítése, majd a már verifikált, kicsiszolt programnak fordító-programmal gépi kódra való áttétele, s az on-line üzemből már e gépi kód használata. A felsoroltak inkább potenciális lehetőségek, miket a gyakorlatban még kevésbé kamatoztattak. Ugy tetszik, hogy a kísérleti berendezés vezérlésére szolgáló programokat célszerűbb a hagyományos fordítóprogram implementációju, széles körben használt nyelvek valamelyikén (pl. FORTRAN) írni, míg a kisebb rendszerekhez, főleg a rendszerek felkészítéséhez, de különösen a CAMAC építőelemek vizsgálatához alkalmasabbak a BASIC típusu nyelvek. A fentiekből látható azonban, hogy számos átmeneti lehetőség is rendelkezésre áll.

A CAMAC software fejlesztési munkái során talán a szükségesnél nagyobb figyelmet szenteltek a nyelv kérdésének bizonyos egyéb megfontolások rovására. Vitathatatlan, hogy egy adott kísérlet időreális programozása akkor a legkényelmesebb, ha a CAMAC akciók minél magasabbszintű nyelven állnak rendelkezésre. E kényelem birtokába azonban a fizikus csak akkor juthat, ha valaki előzőleg helyette már kidolgozta azokat a rendszerprogramokat, melyek a CAMAC akcióutasításokat értelmezni és végrehajtani képesek. Ezt a munkát - legalább egyszer - mindenképpen el kell végezni, s jó volna, ha a számítógépet eladó vállalat végezné el, s a szükséges programcsomagokat eleve a gép programrendszeréhez csatolná. Nem szabad elfeledni azonban,

hogy a CAMAC akcióprogramok mindig környezetfüggők, s pontosan meghatározott CAMAC vezérlőket tételeznek fel. Az elmúlt évtizedben viszont éppen a vezérlés technikája ment át a legjelentősebb változáson, miközben az autonómia a hálózatot irányító kisszámítógéptől lassan a vezérlők irányába csuszott át. Ez olyan fejlődési lehetőség volt, amelyről nem lehetett lemondani a rögzített kontextus megőrzése érdekében. Az akcióprogramok egységesítése, egységesített akcióprogramok kidolgozása az imént említett körülményeken túl, még nagyobb nehézségbe ütközik a laboratórium szintjén, hiszen ott nem csak a vezérlők, de még az alkalmazott kisszámítógépek típusa is kísérletenként más és más lehet, s az alkalmazott gépek programozási szempontból gyakran nem is egybevághók. Az a kérdés is felmerül, vajon célszerű-e egyáltalán a kísérlet teljes programegyüttesét magasabb szintű pl. FORTRAN nyelven írni, célszerű-e egyáltalán magasabb szintű programok előkészítését azon a kisszámítógépen végzni, amelyik a CAMAC hálózatot az adatfeldolgozó lánchoz kapcsolja. Ezekre a kérdésekre nagyon nehéz egyértelmű választ adni, hiszen a helyes megoldás az elmúlt évtizedben rohamosan változó környezetben évenként más és más, és nagymértékben függ az adott kísérlet feltételeitől és a választott rendszerfilozófiától is. Függ az adatfeldolgozó lánc centralizált vagy decentralizált felépítésmódjától is.

A fenti körülmények hatására, a nyelvi kérdés kiélezése helyett hajlanak a többretű programozásra. Ennek értelmében a kísérlet vitelével kapcsolatos munkát jól definiált feladatok sorára bontják szét, és a feladatokat szint szerint csoportosítják. Legalul találhatóak a CAMAC akciók, melyeket az egyes kísérletekhez az adott számítógépi-CAMAC környezet figyelembevételével a CAMAC részletkérdéseit jól ismerő specialisták dolgozzák ki. A CAMAC akcióprogramok könnyű megírhatóságát ez esetben nem a nyelv egyszerűsége, hanem a CAMAC programozói specialisták körében felhalmozódott tapasztalat teszi lehetővé. E tapasztalatok teszik lehetővé a feladatoknak a lehetőség mértékéig való egységesítését is, a rugalmasság teljes fenntartása mellett. Ezen a területen nincs szükség magasabb szintű nyelvekre.

A következő feladatcsoport az időreális adatgyűjtés (és adatkivitel). Az adatgyűjtő programok feladata a hálózat különböző pontjairól pufferokba, majd háttértárolóba szállítani a kísérleti adatokat. Az ezen a szinten programozó hasznosíthatja leginkább a CAMAC nyelvek fejlesztése terén elért eredményeket. Az akció utasításokat használva, nem kell a B/K rendszer konkrét részleteivel törődni. E feladatcsoporthoz a CAMAC-ot az akcióutasítások szintjén ismerő specialisták kapcsolódnak.

A legfelső szinten az adatfeldolgozó programokat találhatjuk, melyek az adatstruktúra ismeretében a felsőbb szintű nyelveken meghonosított állomány-kezelési módszerekkel rendelkeznek. Ezen a szinten a kísérlethez közel álló, a kísérlet fizikai tartalmát, matematikai modelljét jól ismerő fizikus - matematikus programozók dolgoznak. CAMAC ismeretek itt már egyáltalán nem szükségesek.

A többretű programozás gyakorlata nem nagyon kedvez azoknak, akik a kísérlettel kapcsolatos minden munkát a fizikus csoporton belül kívánják végezni. A legalacsonyabb szintű programozási feladatok pl. mindenképpen egy a laboratórium összes fizikus csoportját ellátó, s inkább elektronika semmint kísérlet orientációju csoportra hárulnak. E csoport munkája csupán az adott kísérlet szervezési koncepciójától függ, pl. a választott számítógép(ek) típusától, a vezérlők autonómia szintjétől, stb., de nem függ a kísérlet tartalmától.

Célszerű, ha az adatgyűjtéssel kapcsolatos munkákat is központi szolgáltató együttes végzi, különös tekintettel arra is, hogy a fejlődés iránya a központosított hálózatok irányába mutat. E specialistáknak nem csak egyetlen kísérlet problematikáját kell szem előtt tartaniok. Az adott kísérlet időbeli lefutását az egész laboratórium munkájának idő szerkezetébe kell beleilleszteniök, az átvivő csatornák áteresztőképességét is figyelembevéve.

Miután a jelen pont bekezdésében kifejtettük, hogy a CAMAC modulokat miért célszerűtlen és mért nem lehet a hagyományos periférikus készülékek módjára kezelni programozási szempontból, most viszont arra kell rámutatnunk, miért lehet előnyös

a hagyományos készülékeket a CAMAC-on keresztül csatlakoztatni [89]. A CAMAC lehetővé teszi, hogy egy olyan laboratóriumban, amelyik számos különböző típusu számítógépet használ, a periférikus készülékek ezekhez mégis egységesen csatlakozhassanak. Az egységes csatlakoztatás nem ment fel azonban a programozással járó feladatok alól; az egyes készülékek CAMAC kezelő rutinjait a mindenkori szituációhoz igazodva ki kell dolgozni. E rutinok pedig célszerűen nem akciórutinok, hanem akció rutinokból felépülő és az operációs rendszer felügyelete alá tartozó készülékkezelők. A periférikus készülékek CAMAC csatlakoztatása éppen e készülékkezelők kidolgozásával járó terhek miatt nem mindig egyértelműen előnyös, hiszen a legtöbb periférikus készüléknek a számítógép csatorna adottságait figyelembevevő kezelője a számítógép operációs rendszerével együtt, már eleve rendelkezésre áll. A CAMAC csatlakoztatás kétséget kizáróan hasznos a mérőszobában, ha a kísérlethez tartozó készülékek a legközelebbi számítógéptől távolabb, a kísérleti elektronika mellett helyezkednek el, vagy esetleg nem csak egy, de egy hálózat több számítógépével is kommunikálniuk kell.

Célszerű lehet a hagyományos készülékekhez hasonló módon kezelni a kísérleti rendszerek bizonyos elemeit, így a beavatkozó elemeket is. A léptető motor esetében pl. egy adott szögelfordulást CAMAC fogalmakkal leírni - meglehetősen kényes. Ésszerűbb pl. egy CALL (M1,-125,6) típusu utasítást (és a mögöttes álló rutint) alkalmazni, ami az érintett motort közvetlenül nevezi meg (M1), s argumentumként a szögelfordulás nagyságát és irányát tartalmazza [76].

4.7 NIM-CAMAC EGYÜTTES

A NIM készülékcsalád-szabvány és e szabvány szerint épített készülékrendszerek alkalmazása (amint azt e fejezet bevezetésében már említettük) időben megelőzte a CAMAC szabványt. A CAMAC a NIM rendszert számos területről kiszorította, főként azért, mivel a NIM nem tette lehetővé az általa definiált modulok számítógépi csatlakoztatását. A NIM azonban rendelkezett

olyan tulajdonságokkal is, melyek a CAMAC-kal szemben kitűntettek, s amelyek mai napig való alkalmazását indokolják. A CAMAC-énál jobb földelési rendszere alkalmassá teszi kisjelű környezetben való használatát, felépítési elvei pedig megengedik moduljainak nagysebességű áramkörökként való kialakítását. Hátsó sokpólusú csatlakozói koncentrikus kábelcsatlakozókkal is elvannak látva, az egységnyi modul szélessége kb. kétszerese a CAMAC modul egység szélességének, ami megkönnyíti kábelcsatlakozók és kezelőszervek elhelyezését.

A NIM rendszert általában a CAMAC-al együtt alkalmazzák. NIM modulként alakítják ki a kísérleti elektronika analóg ill. gyors szakaszait, a CAMAC pedig egyrészt fogadja a NIM elektronika jeleit, másrészt - ha szükséges - elvégzi annak vezérlését is, s mintegy az adatfeldolgozó lánchoz kapcsolódó digitális csatlakozó (begyűjtő és szétosztó hálózat) szerepkörét tölti be. A CAMAC elektronika vezérli (állítja be) pl. a NIM modulok vezérlő szerveire hatva, az esetenként választható és változó koincidencia feltételeket, a CAMAC végzi el a rendszer-ellenőrzéssel kapcsolatos feladatokat, pl. a fénydiódákkal végzett eseménystimulációt. Vannak azután olyan feladatok is, amelyeket a CAMAC a NIM részvétele nélkül lát el, így pl. a járulékos információ begyűjtését, a kísérleti berendezés állítható elemeinek helyzetváltoztatását, a kísérleti feltételek stabilizálását, változtatását, szabályozását.

A 29. ábra döntési sémája a NIM/CAMAC választás kérdésében - lényegében az elmondottakat összefoglalva - ad első durva megközelítést [90].

4.8 AZ IEC-625 CSATLAKOZÁSI RENDSZER

Az IEC-625 (IEEE 488-1975) szabvány [91] vagy más néven GPIB, azt a másik csatlakozási rendszert írja le, amelyik a CAMAC mellett a nemzetközi porondon a legnagyobb jelentőségre tett szert. Míg a CAMAC elsősorban sok kisebb elemből felépülő nagykiterjedésű mérőrendszer összefogására, mechanikus és kommunikációs csatlakoztatására szolgál, addig az IEC-625 fel-

adata: kevesebb, de nagyobb, zárt feladatkörű, komplex mérőkészülékek mérőrendszerbe szervezésére szolgál. A CAMAC alkalmazási területe elsősorban a kísérleti fizika, az űrkutatás és a folyamatirányítás, az IEC-625-t pedig az elektronikus fejlesztő laboratóriumban, a zárt feladatkörű ipari mérőberendezésekben és hasonló területeken alkalmazzák.

Az IEC-625 csatlakozási rendszer (melynek rövid magyarnyelvű leírása [27] -ben található) a CAMAC-tól alapvetően eltér. Sinrendszere három vezetékcsoporthból épül fel. Ezek:

- a 8 vezetékes adatvonal,
- a három vezetékes byteátvitel vezérlő vonal,
- az öt vezetékes csatlakozáskezelő vonal.

Az adatvonal kétirányú, aszinkron kapcsolatu, s mérési adatok, programadatok, állapotvektorok és parancsjellegű irányító adatok átvitelére egyaránt szolgál. Noha maximális átviteli sebessége 1 Mbyte/sec, többcélúsága és keskenysége miatt adatátbocsátó képessége lényegesen kisebb a CAMAC-énál.

Mechanikai méretmegkötések nincsenek, egyetlen szabványos mechanikai eleme a csatlakozó, melynek kiképzése olyan, hogy az egyes készülékek kábelei akár csillagba, akár egymáshoz láncszerűen csatlakoztatva helyezkedhetnek el. A sin összes vezetéke társas vezeték (party line), ez teszi lehetővé a csillag/soros alternatívát. A sinre csatlakozó készülékek lehetnek: vezérlők, beszélők és hallgatók. Képességeit tekintve valamely készülék akár mindhárom lehetőséggel rendelkezhet, de egy adott időben csak egyetlen szerepkört tölthet be, s hogy melyiket, azt a rendszervezérlő szerepkörét ellátó készülék külön parancsok útján határozza meg. A beszélő a hallgatónak közvetlen üzenetet, adatokat, adhat át. Értelemszerűen a sinen adott pillanatban csak egy készülék töltheti be a beszélő szerepkörét, de több hallgatója is lehet. A rendszervezérlő nem feltétlenül programozza a készülékeket vagy végez adatfeldolgozást, elsőrendű feladatköre a feladatkiosztás. A feladatkiosztás azonban a készülékek kézi beállításával is elvégezhető, ha a készülékek feladatköre állandó. Adott esetben tehát a rendszervezérlő nélkülözhető is.

Az IEC-625 rendszernek a CAMAC-hoz viszonyítva szembetűnő különbsége kicsiny cimmezése, mely csupán 31 címet ölel fel. Vonalterhelhetőségi korlátok miatt a sinhez csatlakoztatható készülékek száma még ennél is kisebb, mindössze 15. Igen szerények igénybejelentési (szolgáltatási) és igénykezelési lehetőségei is, s a blokkátvitelre sem fordít különösebb figyelmet.

Felsorolt tulajdonságai miatt az IEC-625 a részecskefizika területén a CAMAC alternatívájaként nem jöhet szóba. E mellett azonban kár volna lemondani olyan IEC-625 csatlakozású mérőkészülékeknek a kísérleti rendszerben való alkalmazási lehetőségeiről, amelyekre ott esetenként szükség lehet. E lehetőségek birtoklása érdekében speciális CAMAC modulokat hoztak létre (pl. Kinetic System 3380 GPIB interface), amelyek CAMAC adatot - IEC-625 sin transzformációt végeznek. Ily modulok birtokában lehetséges a 31-34. ábrán látható vagy ezekhez hasonló hálózatok kiépítése.

A 31. ábra a CAMAC keretvezérlő - számítógép együttesnek IEC-625 rendszervezérlőként való alkalmazását mutatja be. Mivel IEC-625 rendszervezérlő önmagát is programozhatja hallgatóként vagy beszélőként így a CAMAC hálózat az IEC-625 sinnel a csatlakozó egységen át szabadon kommunikálhat, és pedig a külső sinen az IEC-625 protokoll, az adatuton a CAMAC protokoll szerint. Mivel egy külső sinre csak 14 készülék csatlakoztatható, ennél több készülék esetén mód van egynél több CAMAC - IEC-625 blokk alkalmazására is (32. ábra).

Korábban a CAMAC hálózati konfigurációk ismertetésénél alapjában véve csak a lényegesebb összeállításokat vázoltuk, a lehetőségeket azonban korántsem merítettük ki. Egy ilyen korábban nem említett lehetőség pl. valamely alapjában véve kis területre koncentrálódó párhuzamos főútra szervezett CAMAC hálózathoz olyan az adatutra csatlakozó soros vonalhajtó alkalmazása, melynek segítségével egy a mérőrendszerrel távolabb eső keret, keretcsoport vagy mérőkészülék, ill. mérőkészülék együttes soros főúton át a mérőrendszerhez fűzhető, a 30. ábrán bemutatott módon [92]. Hasonló megoldási lehetőséget szemléltet az IEC-625 szabvány szerinti készülékek távcsatlakoztatására a 33. ábra.

Nem csak a CAMAC programforrás szolgálhat külső sinen lévő IEC készülékek irányítására, hanem a külső sinre kapcsolódó IEC rendszervezérlő is betöltheti a CAMAC adatuton a társprocesszor szerepkörét, ha az ehhez szükséges CAMAC elemek rendelkezésre állnak, amint azt a 34. ábra szemlélteti.

4.9 CAMAC VEGYES

Paralel és soros főutak együttes alkalmazása. A főutak párosítása kétféleképpen történhet. Vagy egy alapvetően paralel rendszerhez házasítanak soros főutat, vagy fordítva.

A soros főutnak paralel rendszerhez való illesztését az IEC sin CAMAC adaptációjának ismertetésekor az előző pontban már röviden érintettük, most - [92] alapján - e csatlakozásmóddal valamivel részletesebben is megismerkedünk (l. 30. ábra).

A soros főút nem csak közvetlenül hurokvezérlőn át irányítható számítógéppel, hanem közvetve oly konverter modul útján is, amelyik egy párhuzamos hálózat bármely keretének bármely előre meghatározott pozícióján csatlakoztatható. Ez a paralel-soros átalakító egység (PSAE) a hurokra vonatkozóan mindazokat a feladatokat ellátja, miket a hurokvezérlő is, másik oldalával pedig a CAMAC adatut csatlakozási protokollja előírásainak tesz eleget. Választhatóan bit-soros és byte-soros üzemmódban dolgozik, képezi és értelmezi a soros vonalra kimenő, ill. onnan bejövő üzeneteket, képezi a szinkronizáló jeleket és a várakozási (WAIT) byte-okat, elvégzi a longitudinális és horizontális paritásképzést, ill. ellenőrzést, valamint a soros hálózat igényeinek kezelését is ellátja.

A PSAE modulnak négyféle - A, B, C és D betűkkel jelölt - üzemmódja választható. Az A a parancsonkénti programozott adatátvitel-mód, melyben a számítógép programozási uton kezdeményezhet egyciklusú CAMAC műveleteket, de egyszerűsített módon hajthat végre homogén műveletsorokat is (egyazon parancs ismételtetése vagy egyazon művelet ismételtetése ciminkrementálással).

A B megfelel valamely azonos modul-konfigurációju normál soros hálózat blokkátvitelmódjának. Ebben az üzemmódban végigkérdezéssel végzett blokkátvitel történik, miközben a CAMAC pa-

rancs feladatkódja változatlan, címrésze pedig inkrementálódik. A PSAE rendelkezik azonos sorrendű pufferral, melyből az adatok olvasása ugyanolyan sorrendben történik, mint amilyenbe beírásuk történt (FIFO). A blokkátvitel során nyert adatok ebbe a pufferbe olvasódnak be, ill. ebből vivődnek át a soros hálózatra. Blokk-kivitel megelőzően tehát a puffert az átviendő adatokkal az adatuton át fel kell tölteni.

Az A üzemmódot a PSAE munkaregisztereinek útján szolgálja ki, a B üzemmódot paraméter- és állapotregisztereivel. A C üzemmódban sem a munkaregiszterek, sem a paraméter-regiszterek nem vesznek részt; a soros parancsok (végrehajtásuk tervezett sorrendjében) a PSAE azonos sorrendű puffertárolójában helyezkednek el. Az írás parancsokhoz mindjárt az általuk átviendő adat is hozzátartozik. A válaszüzenetek ugyancsak a pufferbe kerülnek, az esetlegesen kiolvasott adattal együtt. Lehetőség van arra is (és ez a lehetőség megengedhető vagy felfüggeszthető), hogy az igényüzenetek ugyancsak a pufferbe kerüljenek.

Míg a C üzemmódban igényüzenetnek a pufferbe való beírása a CAMAC parancsszegmens végrehajtási ciklusában is lehetséges, a D módban nem. A D mód lehetővé teszi max. N számú forrás adatainak beírását a pufferbe, de igényüzenet beírása csak az adatátviteli ciklus befejeztével lehetséges. Az igényüzenet regisztrálása ez esetben is megengedhető vagy felfüggeszthető.

A párhuzamos főút párosítása soros hurokhoz (35. ábra) soros-paralel átalakító egység (SPAЕ) útján lehetséges, mely egység a hurok irányából nézve úgy működik, mint valami keretvezérlő, a paralel főútra nézve viszont az ágvezérlő szerepkörét tölti be [93]. Az SPAЕ, ha a hozzátartozó ágnak szóló üzenetet kap, az üzenetet először "begyűjti", majd parola műveletet kezd az ág felé, egyuttal előkészíti a válaszüzenet első byte-ját is. Ezután megkezdődik a párhuzamos soros átalakítás és a válaszüzenet további byte-jainak a hurokra való juttatása. Az igényüzenet öt bit-es vektora lehet bináris cím, de lehet bitenként önálló értelemmel bíró prioritáskód is.

CAMAC - külső készülék csatlakozás. Valamely szokványos periférikus készüléket vagy a kísérleti elektronika bizonyos nem CAMAC egységeit, mint amilyenek egyes A/D átalakítók, digitálisan programozott erősítők, diszkriminátorok, koincidencia-antikoincidencia elemek, időkonverterek, léptetőmotorok stb. (egy részük NIM fiókban helyezkedhet el) külön erre a célra szolgáló modullal kell az adatuthoz csatlakoztatni. A modul belső csatlakozása a CAMAC szabványt, külső csatlakozása pedig az esetenként változó külső készülék előírásait követi. (1.36. ábra). Célszerűnek látszanék egy közbenső csatlakozási sík kialakítása, hasonló módon, mint ahogy az a számítástechnikában pl. a BSI csatlakozó alakjában öltött testet. Azt lehetne kérdezni, mi szükség volna a CAMAC-on kívül még egy újabb csatlakozási szabványra? Miért nem lehet a külső készüléket egyenesen CAMAC protokollt követő csatlakozó egységgel ellátni? Ez utóbbi kérdésre válaszolva - természetesen lehet, de a megoldásmód nem gazdaságos. A CAMAC csatlakozás sokkal több lehetőséggel rendelkezik, mint amit a külső (periférikus) készülékek legtöbbször megkiván, s ezért célszerűtlen minden egyes készülék csatlakoztatása esetében az egész CAMAC problematikába belebonyolódni. Az egyszerűbb közbenső csatlakozásmód a CAMAC problematikát hivatott kiküszöbölni. A CAMAC - közbenső csatlakozást egymásra leképező modult csak egyszer kell kidolgozni, az egyes készülékek csatlakoztatása során pedig már csupán csak az egyszerű közbenső csatlakozás feladata hárul a tervezőre. Noha az idők során a közbenső csatlakozás módjára több javaslatot is publikáltak [94, 95], általánosan elfogadott CAMAC szabvány erre végül is nem született.

Modulok tervezési irányelvei. A modulok tervezésének egyszerűsítése, de mindenekelőtt a B/K kezelő rutinok egyöntetűsége érdekében az igen tág CAMAC lehetőségek tartományán belül célszerű a modulok bizonyos vonásait egységesíteni [96]. Egységesíthető pl. az állapotjelzés és -hozzáférés módja, a modul vezérlés, stb.

Az állapot-információ max. 24 bit-ből állhat. Adott esetben az állapot bitek két csoportra oszthatók. Az első csoportba a LAM jelzést is adó fontosabb állapotok bitjei tartoznak,

a második csoportba pedig a LAM jelzést nem adó, de CAMAC művelettel kiolvasható bit-ek. Programozási szempontból fontos az állapotregiszter modulon belüli A címének rögzítése, pl. A(15) Célszerűen, e címen minden modulban az állapotregiszter található. A LAM jelet adó állapotbitek VAGY kapcsolatban egyesülve hatnak a modul LAM bistabiljának átállító bemenetére. A LAM jelzés hatására (egy igénykezelési algoritmuson belüli) A(15)F(1) parancs olvassa az állapotregisztert. Ennek az olvasás műveletnek az S2 jele a LAM bistabilt visszaállítja. Az olvasásra A(15)F(0) parancs is használható, mely az előbbtől abban különbözik, hogy a LAM bistabilt nem állítja vissza. A LAM állapotbitek F(10)-el visszaállítható LAM forrásokként tekinthetők, melyek mögött a tulajdonképpeni igénylők húzódnak meg. Ha valamelyik LAM forrás állapota éppen az állapotregiszter tartalmának kiolvasása alatt változnék meg, akkor ez az L jel csak az olvasás művelet lezárulta után jelentkezik igényként. Csak így érhető el, hogy egyetlen L jel se vesszen el, de hogy ne is jelentkezzen kétszeri igényként.

A vezérlő regiszter vezérlő biteket és maszkot tartalmaz. Legfeljebb 24 bit széles. Alcime A(14). Ahány LAM forrás van, annyi maszk bit is szükséges. A regiszter F(1)-el olvasható, F(17) és F(19)-el beírható. F(17) a regiszter vezérlő részébe végez beírást, F(19) pedig a maszk részébe. Az összes maszkbit visszaállítható I jellel, mit az S2 részjel kapuz.

Célszerű a modul LAM képesítés/felfüggesztés (enable/disable) műveletket is egységesíteni.

A [96]-ban szereplő javaslatok nem emelkedtek ugyan szabvánnyá, sőt más helyen [97] más konvenciókkal találkozunk, de az érintett modul-tervezési alapelvek figyelemre méltók.

Modulok és rendszer ellenőrzése. A CAMAC modulok munkájának, üzemképességének ellenőrzése egyáltalán nem elhanyagolhatóan egyszerű feladat. Az ellenőrzés problematikájával mindenekelőtt az új modulok, új típusú vezérlők tervezése során találkozunk. Általában az elektronikus blokk tervezésének különösen fáradságos szakasza a prototípus megépítése után annak felélesztése, s a felélesztett blokknak egy sor módosítással (ez több-

nyire elkerülhetetlen) a kitűzött specifikációhoz való közeli-tése. Így van ez a CAMAC esetében is. E munka a blokkok újból és újból való részletes átvizsgálását követeli meg, s az ellenőrzési eljárások esetenként meglehetősen hosszadalmasak. A CAMAC blokkok ellenőrzésének további területei e blokkok sorozatgyártásával, műszaki átadásával, használat közbeni karbantartásával, valamint hibamegállapításával és elhárításával kapcsolatosak.

Kezdetben a CAMAC egységeket - e célra szolgáló speciális vezérlők, monitorok és áglezárók útján - manuális módszerekkel ellenőrizték; az ellenőrzési algoritmusokat manuálisan hajtották végre, az algoritmusok szerkezete a fent gondolatjelben felsorolt eszközök minémiségétől és az ellenőrzést megszervező személy szubjektumától függött. Az ellenőrzés megbízhatósága ily módon a végrehajtó személyzet adottságainak függvénye, s a mérési eredmény az elvárt értékekkel nehézkesen vethető össze. A manuális módszer hosszadalmas. A manuális módszer nehézkesége különösen világossá válik, ha meggondoljuk, hogy egy akár egyszerű hiba elhárítása során is, az egység ellenőrzését többször is el kell végezni. Először fel kell fedni a hibát, el kell hárítani, és ezt követően az egységet újra kell vizsgálni. Célszerű tartós égetést is végezni, majd ismételten ellenőrzést végezni. A hibadiagnózis igen képzett szakembert kíván, aki azonban nem lelkesedik a kötött algoritmus ismételtetéséért, hiszen ez nem kreatív munka. A manuális vizsgálat további hátránya: lassúsága; egyes tulajdonságok részletes elemzése el sem végezhető, hiszen ha csak egy egyszerű 16 bites számlálóra gondolunk is, az már mintegy 64000 állapot megvizsgálását igényelné. A fent felsorolt indokok miatt a gyakorlatban már hosszabb ideje áttértek a CAMAC egységek számítógépes ellenőrzésének útjára, ami egyébként megfelel az elektronika területén kialakult ellenőrzéstechnikai tendenciának. A számítógépes ellenőrzési módszerek mellett azonban olyan esetekben ma is alkalmazzák a manuális eljárásokat, amikor a számítógépes ellenőrzés felkészítése nehézkesnek látszanék.

A számítógépes ellenőrzés ellenőrző programok segítségével történik. Ha egy adott számítógépi konfigurációra (ideértve a

számítógép - CAMAC csatlakozót is) a CAMAC akciók rutinjai már rendelkezésre állnak, és a nyelv, amelyen ezek az akciók aktivizálhatók rögzítve van, akkor az egyes CAMAC egységekre igen egyszerűen összeállíthatók a működésüket ellenőrző programok. Az egységesítés az ellenőrzéstechnikában sokkal könnyebben valósítható meg, mint a kísérletek vonatkozásában, hiszen az ellenőrzéshez alkalmazott apparátus csak töredéke a kísérleti rendszer számtalan és a kísérletenként változó összetételű berendezés együtteseknek. A tervezőktől a gyártókon át a felhasználóig egységes ellenőrzéstechnika előnyeit pedig bővebben aligha kell ecsetelni. Elég ha a tervezők kidolgozzák az ellenőrzési programokat, amelyek azután a gyártás, az átadás, az üzemfenntartás és használat közbeni hibaelhárítás terén egyaránt alkalmazhatók, ha mindezekben a szinteken egyébként rendelkezésre áll a programok által feltételezett, ellenőrzés céljára szolgáló hardver környezet (softver kompatibilis számítógép és CAMAC csatlakozás).

A CAMAC akciók realizációja assembly nyelven vagy assembly nyelvbe illeszkedő makrók alakjában nem elégítette ki az ellenőrzéstechnika kívánalmait. A probléma megoldására a BASIC nyelv látszott legalkalmasabbnak, de amint már a CAMAC programozás tárgyalása során említettük, az eredeti BASIC interpreter implementációju. Noha az ellenőrzési feladatok általában nem igényelnek olyan munkasebességet, mint a kísérletek, az interpreter lassúsága mégis az ellenőrzés szempontjából sem mondható előnyösnek. A BASIC nyelvből kiindulva a Daresbury Laboratory és az Applied Computer Systems Limited munkatársai dolgozták ki a CATY nyelvet [87, 88], mely tulajdonképpen több mint nyelv - olyan programozási rendszer, mely a CAMAC akcióknak megfelelő közvetlen utasításokat ölel fel. A CATY rendszert oly módon alakították ki, hogy kevés költséggel összeállítható számítógépi konfiguráción is futtatható legyen; 4K (16 bit) memória már elegendő. A rendszer 3K tárkapacitást foglal el; 1K marad az ellenőrző programoknak. Ha ez a memória-terület nem volna elegendő, rálapolási (overlay) technikát alkalmaznak. A minimál konfiguráció számítógépe írógéppel vagy

KS-csőves megjelenítővel és lyukszalag készülékekkel egészül ki.

A CATY megőrizte a BASIC interaktív tulajdonságát az un. COMMAND üzemmódban, RUN üzemmódban azonban nem interpretálja, hanem kompilálja a programot. A rendszer munkasebessége a tiszta gépi program és az interpreter futási sebessége között helyezkedik el: kb. tízszer lassabb, mint a gépi kód, de mintegy ezerszer gyorsabb, mint az interpreter [98].

A gyártó által nyújtott — géptípus - vezérlő együttesre megírt — ellenőrző programok a CAMAC egységekkel együtt dokumentáltan szállíthatók, ami nem csak egységes kezelésmódot és jól áttekinthető helyzetet teremt, de minden szinten igen nagy szellemi energia megtakarítást tesz lehetővé.

Az általános ellenőrzési specifikáció az ellenőrző programon túl egyéb, a tápfeszültségekre, áramköri pontok feszültségére, hullámformákra és azok időzítésére, összeköttetési lehetőségekre vonatkozó előírásokat is tartalmaz, melyek vizsgálata jellemzően manuálisan történik, individuális mérőeszközök igénybevételével. A manuális vizsgálat az ellenőrzés első szakasza, s ezt követi a számítógépes végigkérdezés. Nem csak a manuális, a számítógépes vizsgálat is általában bizonyos hardver segédeszközök használatát is igényli. A CCA-1 vezérlő ellenőrzéséhez pl. szükségesek olyan egységek, amelyek révén regisztrálni lehet a keretvezérlőn áthaladó ág-jeleket, számítógépi úton generálni lehet az L jeleket és ellenőrizni a keretvezérlő által kibocsátott N jeleket és a LAM mintázatot.

A 37. ábra példaképpen egy olyan CAMAC összeállítást mutat be, mely egy mikroprocesszor bázisu ellenőrző modult alkalmazva, elsősorban a paralel főút és a főút mentén elhelyezkedő keretvezérlők működésének vizsgálatára szolgál [99]. E modul nem csak egyedi egységek, hanem nagyobb hálózatok átvizsgálását is lehetővé teszi. Itt kell megjegyezni, hogy a CAMAC modulok ellenőrzése viszonylag az egyszerűbb feladat, ha már biztosak lehetünk abban, hogy az ág és a keretvezérlő hibátlanul működik.

Általában nehéz megkülönböztetni a vezérlők által okozott információ-torzulásokat a modul hibáitól. A 37. ábrán bemuta-

tott elrendezés elkerüli a kétértelműségeket, mivel a benne használt ellenőrző modul egy külön, a paralel főuttól független kommunikációs utvonallal is rendelkezik. A normál CAMAC pozícióban elhelyezkedő ellenőrző modul érzékeli az adatut összes jeleit, de működését nem ezek a jelek irányítják, ahogy a közönséges modulok esetében. Az ellenőrző modulon a paralel főut is áthalad, s így a modul ennek állapotait is megfigyelheti. Az ellenőrző modul külön kommunikációs vonalán kapja utasításait, és ezen utasításoknak megfelelően ellenőrizheti akár az ág, akár az adatut működését. Rendelkezik egy 256 szó/80 bit kapacitású belső memóriával, ebbe gyűjti be az ellenőrzött vonalak állapotait. Az ellenőrző modul vezérlését saját mikroprocesszora látja el. Kommunikációs vonala soros, mely akár a CAMAC rendszert irányító számítógéphez, akár írógéphez kapcsolható. Ha számítógéphez csatlakozik, automatikus ellenőrzés végezhető. Az ellenőrző modulnak a számítógép parancsot ad, pl. az ág működésének vizsgálatára, majd CAMAC parancsokat bocsát ki az ágra. Az ágnak a parancsokra adott jelei az ellenőrző modul tárolójába rögzítődnek, ahonnan a számítógép a soros vonalon azokat bekérheti a referenciával való összehasonlítás céljából. A detektált hibákról nyomtatott riportot ad. Amennyiben az ág működése hibátlan, hasonló módon elvégezhető az adatut - a keretvezérlő - vizsgálata is.

Az ellenőrző modul a funkcionáló mérőrendszerben is alkalmazható, ahol állandóan gyűjtheti az ág vagy az adatut jeleit (állapotait). Üzemzavar esetén a rendszer letörését megelőző 256 tranzakció állapotai a számítógépbe továbbítódnak hibamegállapítás céljából.

4.10 ALKALMAZÁSI PÉLDÁK

Multiplexelt analóg-digitális átalakító kezelése és kioldása a CAMAC hálózaton át [100]. Amikor egy esemény kapcsán egyidejűleg több amplitudó értékét kell regisztrálni, célszerű ezeket az amplitudókat analóg tárolókba rögzíteni, majd azután egy A/D átalakító segítségével időben egymás után az amplitudó értékeket digitalizálni; az analóg tárolókat az A/D át-

alakító bementére analóg multiplexer (ciklikus letapogató) kapcsolja rá. A kísérleti berendezés két fő részből, a számítógéphez csatlakozó CAMAC hálózathoz és az analóg tárolókat (A), a multiplexert (MX), az A/D átalakítót, valamint ezek segédáramköreit felölelő NIM egységekből áll, amint azt a 38. ábra szemlélteti. A NIM egységek és a CAMAC hálózat közötti kapcsolatot egy speciális, az adott feladatra szolgáló ún. PACAM modul látja el. A NIM egységek és a PACAM közötti kommunikáció tisztán digitális jelek útján történik.

A NIM egységek két csoportba tömörülnek. Az elsőbe az analóg tárolók (jelnyújtók), a multiplexer és az A/D átalakító tartozik, a másodikba az analóg tárolóktól független (tag) információkat felvevő és digitálisan kódoló egységek (R), valamint egy véglezáró (T). A PACAM modul max. 8 analóg tárolót képes ellátni, de egy CAMAC hálózatban több ilyen modul is alkalmazható. A PACAM négy egységmodul szélességű, és így négy független igénylő pontja (L) van, melyek mindegyikét felhasználja. Többféle üzemmódja lehetséges; üzemmódját a számítógép programozási uton választja meg, de bizonyos üzemmódok manuálisan is választhatók.

A PACAM irányítja a NIM egységek működését, illetve végzi el az információ átvételt ezektől az egységektől. A PACAM munkáját az esemény által kiváltott "adat kész" jel indítja, melynek hatására puffertárolójába felveszi az első analóg tároló már digitalizált jelét, egyúttal LAM-al jelzi az autonóm ágvezérlőnek az adatátadás megindulását. Amíg az AD átalakító a következő analóg tároló jelét digitalizálja, az autonóm ágvezérlő a számítógép programfüggetlen csatornáján át a PACAM puffertárolójából az adatot automatikusan átviszi a memóriába. Az itt alkalmazott ágvezérlőnek nincs hisztogramképzési üzemmódja, ezért az átvitt adatok nem spektrumot hanem listát képeznek. Kis eseményszám, elegendő idő rendelkezésre állása esetén a PACAM egy az előzőtől eltérő üzemmódjában lehetséges az egyes LAM jeleknek megszakításkérésként való kezelése, amikor is az egyes digitalizált értékeket és csatolt információjukat megszakítási program kezeli. Esetleges hibaállapotról, üzemképtelen állapotról, stb. a CAMAC-számítógép együttest külön 'hiba-

-LAM' jel tudatja. Az információ-feldolgozás idejére az analóg tárolók bemenetei lezárulnak. A rendszer programozási uton kezdőhelyzetbe állítható és indító paranccsal aktivizálható.

Proporcionális-kamra kiolvasó rendszerek. A [101] -ben leirt, s eseményselekcziót is végző proporcionális kamra kiolvasó rendszer a kamrához közvetlenül tartozó gyors elektronikából és a kamrától mintegy 40 m távolságra fekvő CAMAC együttesből épül fel; a két részt kábelek kötik össze, amint az a 39. ábrán látható. Az ilyen módon két tömbre oszló esemény-elektronika gyors része lényegében a 6. ábra felső részén bemutatotthoz hasonló huzalerősítőket ölel fel. Az ezek kimenetén fekvő tároló elemek léptetőregiszterekbe szerveződnek. Az esemény regisztrálása és a megütött huzalok címének identifikálása a [14] -ben leirt, és a 7. ábrán szemléltetett elv alapján történik (1. 3. fejezet "Az esemény-elektronika" rész alatt). Soros letapogatás helyett azonban itt síkonkénti, ill. koordinátánkénti paralel-soros letapogatás történik, minden egyes sik X és Y koordinátájához saját léptetőregiszter, ill. letapogató tartozik a gyors elektronika oldalán, a CAMAC oldalon pedig koordinátánként 16 szavas puffert tartalmazó kódoló egység (R0). A letapogatás 8 MHz órafrekvenciával, koordinátánként paralel történik. A gyors rész és a CAMAC összeköttetése viszonylag egyszerű. A kábelezésen az órajelek haladnak a gyors elektronika irányába és a léptetőregiszterek kimenő jelei a CAMAC irányába. A primer regisztrátumot így egyáltalán nem kell a CAMAC-hoz átvinni. Az órafrekvenciát számláló bináris számláló (amelyik tulajdonképpen a megütött huzalok bináris címét képezi) az R0 kódoló egységben foglal helyet. Több, egymás szomszédságában megütött huzal száma itt is függelékként csatlakozik a regisztrált huzalcimhez, mely ez esetben a megütött huzalok közül a középsőt identifikálja. Egy koordinátán belül végül is max. 16 megütött huzal, ill. huzalcsoport azonosítható. Az egyes koordinátákhoz tartozó R0 egységeket egy e célra szolgáló vezérlő irányítja egy a CAMAC adatut felett a keret hátlapján elhelyezkedő külön sinrendszer útján. A vezérlő működését a triggervezérelt részel generátor indítja, mely egyéb-

ként a primer kamrainformációnak a léptető regiszterbe való paralel beírását is kapuzza.

A huzalcím 10 bit, a csoportszélesség 5 bit nagyságu. A kettő egymáshoz fűzve kerül a pufferbe, ahonnet az eseményinformáció kódolt tárolásának megtörténte, ill. a puffer feltöltődése után F(0) v. F(2) feladatkódot és A(0)...A(15) alcimeket tartalmazó CAMAC parancsokkal az R1-R15 olvasó vezetékeken át olvasható ki.

A rendszer különböző feltételeknek engedelmeskedő eseményszelekciót képes végezni. Ezek közül az egyik: a CAMAC parancsokkal kiválasztott sikkordináták és külső bemenetek közötti közvetlen koincidencia. A külső bemenetek rendszerint szcintillációs számlálók, melyek a részecske egy meghatározott pályáját figyelik.

Egy másik szelekciós mechanizmus az R0 egységek részét képező csoport logikák multiplicitását veszi alapul, az általa vizsgált eseményeket on-line, off-line és elutasított kategóriába sorolva.

Az R0 modulok előlapján egy erre a célra szolgáló csatlakozón hozzáférhetők a puffer címvezetékei külső vezérlés, adatvezetékei adatkiolvasás céljára, miáltal mód nyílik arra, hogy speciális gyors processzor számításokat hajtson végre a begyűjtött adatokon, esemény szűrés céljából.

A rendszer működőképességének ellenőrzése céljából lehetséges eseményszimulációt is végezni, azaz a CAMAC adatuton át ismert adatokat írni a pufferokba, majd ezeket az adatokat kiolvasva analizálni, és az eredményt összevetni az elvárt értékekkel.

A fentiekben leirttól lényegesen eltérő elven működik az [A6] -ban ismerttetett kamra kiolvasó-regisztráló rendszer, melynek felépítésmódját a 40. ábra szemlélteti. (1. még hasonló rendszert [A2],[15] -ben). A kiolvasó elektronika itt is közvetlenül a kamrára épül rá. Szervezésmódja a következő: A csatornánként erősítőből, diszkriminátorból, késleltetőből és tároló bistabilból álló csatornaelektronika modulokba szerveződik. Egy modul 256 csatornát szolgál ki; elemei 16 áramköri

lapra oszlanak el, egy áramköri lap tehát 16 csatorna elektronikáját tartalmazza. A csatornalelektronika elemein kívül az egyes modulokhoz még kódoló áramkör és pufferáramkör is tartozik. A rendszer 64 modult ölelhet fel, s így a max. lehetséges csatornaszám: 16384. A modulok egy soros-párhuzamos ágra csatlakoznak, mely ágon a kamrainformáció 26 bit széles szó alakjában halad az un. vevő egység bementére. Az információ a 16 bites lineáris pozíciókódból (R1-R16), a modult azonosító 6 bites bináris kódból ($N_{k1} - N_{k6}$) és a modulon belüli lapot azonosító 4 bites bináris kódból ($N_{n1} - N_{n4}$) tevődik össze. Az egy modulon belüli lapok letapogatása óragenerátor jeleivel történik, és csupán annak a lapnak a regisztrátuma vivődik át, amelyik beütést észlelt.

A vevő egység bemenetén regiszter helyezkedik el, ez veszi át az órajel hatására beérkező adatokat. A lineáris kód (R1-R16) prioritáskódolóra kerül, mely a beérkező adatokat sorban bináris kóddá alakítja át. Az információáram mérséklése és a számítógép munkaidejének csökkentése céljából lehetőség van a megszólaló szomszédos huzalok csoportjának elnyomására. E mellett a csoportból csak a második megszólaló huzal címe regisztrálódik, a 2, 3 és 4 huzalból álló csoportok "középpontja". A vevő CAMAC keretben helyezkedik el. A binárisan kódolt szavakat akár az adatutra, akár egy mellette lévő puffertárolóba továbbíthatja. Az egység előirt munkatartománya programozás útján vagy manuálisan választható meg. Az óragenerátor frekvenciája 625 KHz - 10 MHz között állítható; választható pufferrel vagy számítógéppel való üzemmód, ill. az ellenőrzési tartomány. Ez utóbbiban a bemenő regiszterre vagy tisztán 0-ás vagy tisztán 1-es bitek érkeznek.

Amennyiben rövid idő alatt kell összegyűjteni a kamrainformációt, úgy azt a vevő modulból nem az adatutra, hanem a pufferba kell irányítani. Pufferolt adatgyűjtés esetén a max. órafrekvenencia 30 MHz. Az adatoknak a pufferből a számítógépbe való továbbítása "blokkátvitel állj jelzéssel" üzemmódban történik. Az "állj"-t jelző Q=0 jellel együtt a vevő egységből az adatutra kerül az összes regisztrált kamrahuzalok száma (mennyisége) bináris kód alakjában.

Jellemző adatként megjegyzendő, hogy amennyiben a kísérleti rendszerben a max. lehetséges huzalszám(16348) van jelen, s a huzalok 5 százaléka szólal meg, a tőlük származó információ kiolvasása, kódolása és rögzítése 90 μ sec időt igényel.

4.11 CAMAC MÓDOSULATOK

A kísérleti fizika igen sokszerű igényének döntő többségét a CAMAC a mögöttünk álló évtizedben kielégítette. Gyakran felmerültek azonban olyan feladatok is, amelyek esetenként a CAMAC szabvány bizonyos módosításával lényegesen egyszerűbben voltak megoldhatók, mintha e szabványokhoz mereven ragaszkodtak volna vagy máskor megoldásuk csak a CAMAC keretein kívül volt lehetséges. Ilyen módosítás leginkább akkor engedhető meg, ha a CAMAC kompatibilitás általa nem sérül, s a szabványos és a szabványon kívül eső elemek együtt használhatók.

Az irodalomból ismeretesek olyan módosítások is, amelyek közvetlenül a szabványt érintik. A Szovjetunióban kialakított VEKTOR-rendszer logikája pl. azonos a CAMAC-éval, a mechanikai méretek azonban eltérnek; míg a CAMAC hüvelyk méretrendszerű, a VEKTOR metrikus. Más esetben [103] a CAMAC adatut átviteli sebessége bizonyult kevésnek, amin a TTL adatutnak ECL adatuttal való felcserélése révén segítettek.

A CAMAC mérsékelt átviteli sebessége mellett esetenként a kötött terjedelem jelentkezett korlátozó tényezőként, s készítette a rendszerszervezőt új megoldások keresésére és létrehozására. Noha a rendszerkeret szintjén a CAMAC-ot nem szabványosították, a gyakorlatban hosszú esztendőkön át olyan megoldások alakultak ki, melyeknél egy rendszerkerethez legfeljebb négy ág csatlakozhatott, összesen tehát 28 keret, keretenként 23, a teljes hálózat tehát összesen 644 modulpozícióval. Ez nem minden esetben bizonyult elégségesnek egy-egy kísérlet igényeinek kielégítésére.

A Stanford Linear Accelerator Center (SLAC) által létrehozott módosított CAMAC rendszer [102] a rendszerkeret és a számítógép közé egy további szintet iktat be, miáltal lehetővé válik max. 8 rendszerkeretet 56 ágat és 392 keretet felölelő fa-struk-

turáju hálózat kialakítása. Egy rendszerkeretnek a DEC VAX 11/780 típusu számítógéphez való csatlakoztatásmódját a 41. ábra szemlélteti. A szokványos CAMAC hálózathoz viszonyítva itt három új elem jelenik meg: a VAX-CAMAC csatornaprocesszor (VCC), a SLAC szimmetrikus paralel főút és az ez utóbbit aszimmetrikus ágra leképező ágfogadó. A SLAC szimmetrikus főutat az ághajtók és keretek között éppúgy alkalmazzák, mint a csatorna és a rendszerkeretek között. E főút hossza 1 km-ig terjedhet, ami lehetőséget ad térben nagyobb területre eloszló mérőhálózat kialakítására. Az adatátvitel ritmusa programozási úton illeszthető az adott kábelhossz által meghatározott konkrét körülményekhez. A VCC processzor lényegében a 16 bit szélességű bitszeletelt AMD 2900 bipoláris mikroprocesszorból, továbbá a VAX ill. a CAMAC csatlakozóegységből épül fel. A csatornaprocesszor, a VAX-tól kapott parancs és paraméterek alapján előkészíti az ághajtókat, majd ez utóbbiakkal együttműködve lebonyolítja az előírt CAMAC műveletet ill. műveletsort, pl. meghatározott számú adat átvitelét a CAMAC modulok és a VAX memóriája között. A blokkátvitellel kapcsolatos végigjárást (címletapogatást), a távolsági adatátvitelt ezenközben az ághajtók bonyolítják le.

Az elvégzendő feladatokat a felsőbb nyelvi szinten programozó rendszerhasználó FORTRAN rutinok megnevezésével írja elő. E rutinok, lényegében, "csatorna program"-ok, melyek adat- és állapotpufferokat definiálnak. E pufferokra vonatkozó előírásokat a VAX operációs rendszere adja át megfelelő készülék-hajtó programjának, mely utóbbi azután kijelöli a tényleges fizikai memóriahelyeket (pufferek). A műveletekkel kapcsolatos paramétereket a készülék-hajtó adja át VCC-nek. A feladatok kezdeményezésén (beleértve a paraméterek átadását is) túl további VAX tevékenységre nincs szükség, minden egyéb munkát a VCC csatornaprocesszor, az ághajtókkal együttműködve, saját maga véggez el.

A VAX mindig olyan VCC programokat aktivizál, melyek CAMAC műveletek pontosan meghatározott sorát (akció) bonyolítják le. A feladatok megváltozása, új feladatok felbukkanása esetén te-

hát VCC programjait is módosítani kell, ill. programrendszerét új programokkal kell kiegészíteni.

A részecskefizikában alkalmazott kísérleti berendezéseknek, általában a kísérleteknek, a fejlődés során néhány olyan strukturális sajátossága került előtérbe, melyeket a CAMAC rendszer kialakításakor még nem vehettek figyelembe:

- Számos keretet azonos típusu modulok foglalnak el.
- Sok, különböző típusu modulnak a kiolvasási jellegzetességei egymással megegyeznek.
- A kísérletben résztvevő modulok legnagyobb része igen korlátozott számú (kiolvasási) műveletet képes végezni.
- Az egy kísérletre eső modulok száma oly nagy lehet, hogy összefogásukra a CAMAC hálózat terjedelme nem elegendő.

E vonások figyelembe vételével és a mögöttük meghuzódó kényszerítő hatások sarkalására jött létre a CAMAC-tól lényegesen eltérő, de bizonyos mértékben CAMAC kompatibilis és azal együtt is használható ROMULUS/REMUS rendszer [104].

A ROMOLUS/REMUS rendszer alapgondolata az, hogy a nagy homogén áramköri területekre kiterjedő egyszerű olvasási műveleteket külön kell választani az írás műveletektől, továbbá, hogy az egyszerű olvasás-írás műveleteknél bonyolultabb műveletcsoportokat is külön kell végezni. A kísérleti elektronika e koncepció szerint három részre osztható, az egyszerű olvasás műveleteket a ROMULUS, az írás műveleteket a REMUS, az összetettebb feladatokat pedig a CAMAC hálózat végzi. E három hálózattípus kombinációjával építhető fel a kísérlet teljes adatgyűjtő-szétosztó rendszere.

A keretei között végezhető műveletek szempontjából a ROMULUS a CAMAC alrendszereként fogható fel. Ágakból és adatutakból szervezett sokszintű, kiterjedt fa-hálózattá építhető ki, a 42. ábrán szemléltetett módon. Alap-építőelemei: az ághajtó, a vertikális sín, a keretvezérlő és a horizontális sín. Ez utóbbi az esetek többségében azonos lehet a CAMAC adatutal, melytől azonban egyes speciális keretvezérlők alkalmazása esetén el kell térni. A vertikális sín a CAMAC párhuzamos fő-

uttól lényegesen különbözik, ennek megfelelően az ághajtók és keretvezérlők is sajátosak. A ROMULUS hálózatszervezés első szembeötlő eltérése a CAMAC-hoz viszonyítva, hogy míg a CAMAC keretvezérlők az ág akármelyik pontjára elhelyezhetők, a ROMULUS keretvezérlők helye topológiailag kötött, ami a hálózat topológiailag rögzített kiolvasási sorrendjéből következik.

A ROMULUS hálózatban mindössze néhány CAMAC parancs hajtható végre (nem CAMAC módon), melyek közül a legkarakterisztikusabb az olvasás művelet. Olvasás azonban csak blokk-üzemmódban végezhető; egyes modulok egy-egy adatának magában álló kiolvasása nem lehetséges.

A parancsok a hálózatra nézve transzparenssek, amennyiben valamely ághajtóhoz érkező parancs automatikusan "szétszóródik" a hozzátartozó részhálózat keretvezérlőihez, ill. moduljaihoz. Az olvasás művelet végrehajtása pl. a következő módon történik: Az ághajtó az ág egy e célra szolgáló OLVASÁS vonalán jelet generál, mely jel minden a vonal útjában fekvő vezérlőn áthalad. Az első olyan vezérlő, melynél kiolvasásra váró érvényes adat várakozik, az OLVASÁS jelet időszakosan feltartóztatja, egyúttal megindítva adatainak blokk-üzemmódu átadását az ághajtóhoz. Az adatblokk széljelzővel (marker) zárul. A széljelző megjelenését követően az OLVASÁS jel a láncon továbbhalad a következő vezérlő irányába. Miután az utolsó keret adatainak kiolvasása is megtörtént, az OLVASÁS jel OLVASÁS-VISSZA jel alakjában tér vissza az őt kibocsátó ághajtóhoz.

Az ághajtó minden egyes adat átvitelénél inkrementálja egy erre a célra szolgáló számlálóját, melynek regisztrátumát az adatblokkhoz csatolja amikor annak széljelzője megjelenik, majd a számlálót nullázza. Egy másik számláló az ágon végzett összes átvitelt számlálja, mely regisztrátum az OLVASÁS-VISSZA jel megérkeztekor fűződik az adat blokkok sorának végére. Az elmondottak értelmében pl. 42. ábrán felvázolt hálózatban a kiolvasási sorrend a következő:

A1-A2(B1-Bn, C1-Cn) -An, D1-D2(E1-En)-Dn, F1-F2-F3.

A vertikális sin adatvonalának szélessége 16 bit.

A vertikális sinnek nem csak az olvasás, hanem a többi, a hálózatban végrehajtható parancs számára is külön egyedi vezetéke van.

Mivel a modulok felkeresése a műveletek kapcsán topológiai-
lag meghatározott sorrendben, automatikusan történik, külön cím-
zésre a ROMULUS hálózatban általános esetben nincs szükség. A
vertikális sinben azonban mégis megtalálható az A1...A4 és az
N1...N5 címvezeték. E vezetékeknek többféle szerep juthat. Egy-
részt minden a hálózatra kiadott parancshoz A0 cím járul, ez
továbbbitódik, másrészt ha arra szükség van, a keretvezérlő e
vezetékek igénybevételével az éppen kiolvasott adathoz függeszt-
heti az adatforrás címét. Bizonyos kivételes esetekben lehet-
séges az A és N vezetékekre az ághajtó által továbbított címnek
az adatutra való leképezése is.

A ROMULUS rendszer nem szolgáltat állapot információt; fel-
tételzés szerint a kiadott parancsok végrehajtódnak, de hogy
végre is hajtódtak, arról nincs visszajelzés. Az egyetlen ki-
vétel az OLVASÁS, melynek jele "megvizsgálja" a keretvezérlő
állapotát, már ti. hogy várakozik e kiolvasandó adat az adott
keretvezérlő hatáskörében.

A REMUS csupán abban különbözik ROMULUS-tól, hogy bizonyos
korlátozott irási lehetőségekkel is rendelkezik. Ugyanazt a
vertikális sint alkalmazza, de REMUS üzemmódban a sin egyes
vezetékeinek szerepe a ROMULUS értelmezésétől különbözik. A
két rendszer logikailag egymáshoz illeszkedik; ROMULUS/REMUS
hálózat alakítható ki, de a két rendszer keverése egy ágon be-
lül nem célszerű, s mivel a ROMULUS hálózati elemek IRÁS pa-
rancsokat nem tudnak továbbítani, a REMUS ágakat a fa-struktu-
ra tövában kell elhelyezni. Minden irás parancshoz saját azon
cím járul, amelyikre az adatot be kell írni. Az irás parancsok
nem osztódnak szét automatikusan a hálózat egészére, miként a
ROMULUS OLVASÁS parancsok.

Az ághajtó alaptípusai a 243 típuszámú ROMULUS és a 283
típuszámú REMUS vezérlő. Ezek a REMUS irási lehetőségeitől
eltekintve logikailag azonosak. Az ághajtó CAMAC keretben mo-
dulként helyezkedik el, és a hozzá tartozó ROMULUS/REMUS rész-
hálózatot látja el. A CAMAC keret maga a fa struktúra különböző

szintjein foglalhat helyet, esetleg a ROMULUS/REMUS-al együtt alkalmazott CAMAC részhálózatban.

Ugyancsak két alaptipusa - ROMULUS, ill. REMUS - van a keretvezérlőnek is, melyek az IRÁS művelet kivételével logikailag hasonló tulajdonságúak. Előlap-kapcsolókkal választható ki a keretvezérlő predeterminált üzemmódja, pl. OLVASÁS vagy OLVASÁS TÖRLÉSSEL; blokkátvitel megállással vagy ismétléssel, vagy végigkérdezéssel stb.

Az ághajtónak és a keretvezérlőnek egyaránt van saját kiskapacitású adatpufferje, de külön puffer-egység hozzátársításával a puffer kapacitás 16 K szóig bővíthető.

Mind az ághajtónak, mind a keretvezérlőnek kidolgozták az alaptípusoktól eltérő, speciális követelményekhez illeszkedő, változatait is. A hálózat kiegészülhet különböző járulékos blokkokkal is, így: a vertikális sín utjában fekvő ROMULUS processzorral, ROMULUS utválasztó (routing) egységgel, keret társvezérlővel stb.

5. A GYORSSIN

A CAMAC születésének idején a számítógépnek mérőrendszerben való alkalmazásáról még kevés tapasztalat állt rendelkezésre. A CAMAC sikere, máig tartó széleskörű alkalmazása, tervezői jövőbe tekintő koncepciójának helyességét igazolja. Közben a fizikai kísérlet követelményei szakadatlanul fokozódtak, s az utóbbi években mind gyakrabban haladták meg a CAMAC korlátait. Az integrált áramköri technika és a számítástechnika expanziója során sok olyan új lehetőség tárult fel, melyek hasznosítására a CAMAC alig vagy egyáltalán nem alkalmas. Mai tapasztalataink fényében az is látható, hogy a CAMAC-ban alkalmazott egyes technikai megoldásoknál vannak már lényegesen jobbak is. Mindezen tényezők serkentésére az U.S. NIM bizottság már 1975-ben létrehozott Advanced System Study Group /ASSG/ néven egy munkacsoportot, azzal a céllal, hogy egy újabb időszakra előretekintve, megfogalmazza a nagyenergiás fizika mérésadatgyűjtő rendszereivel szemben támasztott általános követelményeket. E követelményeket 1977-ben tették közzé [105], s azok tömören az alábbiakban foglalhatók össze [106].

1. <100 ns adatátviteli idő.
2. Nagysebességű blokkátvitelt elősegítő szervezőmód.
3. Lehetőség a hálózat szórványadatainak gyors kiolvasására.
4. Modularitás; tág határok között való kiépíthetőség.
5. Az építőelemek helyfüggetlen címzési lehetősége.
6. Nagy rendszer számos elemének közvetlen címezhetőségét lehetővé tevő tág címtartomány.
7. Széles adatvonal a minél nagyobb adatáram elérése érdekében.
8. Az egész hálózatra nézve egységes, az alkalmazott áramköri megoldásoktól független kommunikációs protokoll, transzparens hálózat.
9. Többprocesszoros szervezőmód.
10. Hibadetektálási lehetőségek.
11. Hajlékony táp és mechanikai rendszer.

12. A jövőben megszülető technikai eredmények befogadására való alkalmasság.
13. A gyártói - alkalmazói közeghez való illeszkedés.

E követelmények megfogalmazását követően világossá vált, hogy a CAMAC az elkövetkező évek igényeit nem tudja kielégíteni, s hogy tehát szükség van egy új rendszer kidolgozására és szabványosítására. A CAMAC szerepe (bizonyos területeken) mindenképpen zsugorodnék, újabb, megfelelőbb szabvány híján az egyedi megoldások kerülnének előtérbe, s ismét csak a sokféleség hátrányos következményeivel kéne szembenézni. Az új szabvány kidolgozására 1976-ban az U.S.A.-ban munkacsoportot hoztak létre, melynek munkájában való részvételre felkérték az európai CAMAC bizottság és egyes független európai és kanadai intézmények szakértőit is. A munkák koordinálását az U.S. NIM bizottság (Louis Costrell, Chairman NIM Committee) végzi. Az új rendszer a 'FASTBUS' (GYORSSIN) elnevezést kapta.

Az alábbiak a GYORSSIN-t néhány róla szóló közlemény [107-110], valamint a legújabban kiadott előzetes specifikáció [111] alapján ismertetik. A szabvány szövege még nem végleges, így megeshet, hogy egy az olvasó kezébe kerülő újabb szövegváltozat az itt leírtakhoz képest egyben-másban eltér. A több éve tartó munka folyamán igen sok módosítást hajtottak végre az eredeti anyagon, de az, ami a jövőre nézve leginkább fontos - a sinrendszer szervezési koncepciója - nagyjából már ki-kristályosodottnak tekinthető. Így a következő sorok abban a reményben vethetők papírra, hogy egyrészt segítséget nyújtanak a szabvány várható végleges szövegének gyorsabb megértéséhez, másrészt a jövő fizikai mérőrendszereinek előtervezéséhez.

5.1 A GYORSSIN ÁLTALÁNOS TULAJDONSÁGAI

A 'GYORSSIN' elnevezést nem szabad szó szerint venni. A szó nem sint, hanem elemi sinszakaszokból a mindenkori szükségletnek megfelelően modulárisan összeállítható, szűkíthető-tágítható hálózatot jelöl. Az elemi sinszakaszt SZEGMENS-nek nevezik. A szegmens sinből és tartozék áramkörökből áll; a

szabványnak megfelelő módon a legkülönbözőbb készülékek csatlakoztathatók rá, melyek egymással a GYORSSIN előírásai - protokolljai - szerint kommunikálhatnak. Hálózat több szegmens egymáshoz kapcsolása révén alakítható ki. A különböző szegmensekre csatlakozó egységek ugyancsak a GYORSSIN protokolljainak megfelelően kommunikálhatnak egymással. A szegmensnek két meghatározási szintje van: logikai és fizikai, s két fizikai megvalósításmódja - KERETSZEGMENS és KÁBELSZEGMENS. Ezek logikailag megegyeznek, egymástól csupán sajátos szerepkörükkel összefüggő apróbb, de nem lényegbe vágó részletekben térnek el. A keretszegmens a CAMAC keretre emlékeztet, feladata a mérőrendszer készülékeinek koncentrált összefogása, a kábelszegmens viszont elsősorban az egymástól távolabb fekvő készülékek, keretszegmensek, keretszegmens csoportok összeköttetésére szolgál. A kábelszegmens és a CAMAC főutak között azonban nincs analógia. A kábelszegmens protokolljai megegyeznek a keretszegmensével olyannyira, hogy hozzá közvetlenül is kapcsolhatók a GYORSSIN csatlakozási előírásainak eleget tévő készülékek.

A GYORSSIN átviteli sebessége - előírás szerint - legalább egy nagyságrenddel nagyobb, mint a CAMAC-é, tehát minimum 10 MHz. Általában azonban csak a terjedési és logikai késés idők limitálják. Az eddigi vizsgálatok azt mutatják, hogy blokkátviteli üzemmódban adatárama meghaladja a 80 Mbyte/sec, rendezetlen (random) címzés mód esetén pedig eléri a 40Mbyte/sec értéket.

A GYORSSIN adatátvitelmódja paralel; az adatszélesség: 32 bit.

Címzés módja földrajzi (azaz helyhez kötött) és logikai; címátvitelmódja paralel; címszélessége 32 bit. A földrajzi cím a szegmenshez, a szegmensekből álló hálózat adott csatlakozási helyéhez tartozik, a logikai cím a GYORSSIN-re csatlakozó készülékhez, melyet ez a rendszer kezdeti beállítása - kezdetelése - során kap meg. Egy adott szegmens adott helyére bármely készüléket csatlakoztatnak is, ez az e helyhez tartozó földrajzi címmel hívható. Viszont egy adott logikai címmel ellátott készülék mindig saját logikai címére hallgat.

A GYORSSIN alapjában és eleve többprocesszoros rendszerek szervezésére szolgál, és ennek megfelelő arbitrációs mechanizmusa is van.

5.2 A SZEGMENS

A szegmens, mint említettük, a GYORSSIN szervezési alap-egysége; sinből és tartozék áramkörökből áll. Max. 32 készülék (önálló funkcióval bíró blokk) csatlakoztatható hozzá egységes módon. A készülékek két fő típusa: a VEZETŐ (MASTER) és a VEZETETT (SLAVE). A kettőt az különbözteti meg egymástól, hogy míg a vezető kisajátíthatja a sint, és irányíthat más készülékeket, addig a vezetett soha nem uralhatja a sint, csak alárendelt lehet. Azt követően, hogy egyvalamely vezető a sint kisajátította, bármely másik készülékkel - vezetettel, de vezetővel is - kommunikálhat. Két vezető kommunikációs kapcsolatában mindig a sint uraló vezető a fölérendelt, a másik pedig az alárendelt. Ilyenkor az alárendelt vezető vezetett szerepkörét tölti be.

Egy szegmenshez egynél több (elvileg max. 31) vezető tartozhat. A vezetőknek a sinért folyó veresengéséből származó konfliktusok arbitráció útján oldódnak fel. Minden vezetőnek meghatározott arbitrációs szintje van, és az egyszerre folyomódók közül mindig a magasabb szintű nyeri el a sint. Az arbitrációs séma csupán arra szolgál, hogy megakadályozza a sinnek két vezető általi egyidejű használatát, de a birtokon belül lévő vezetőtől egy időközben jelentkező - s nála magasabb szintű - a sint nem veheti el. Ez azt jelenti, hogy - hacsak nincs tulterhelés - a vezetők a sint mindig jelentkezésük sorrendjében használják. Egyéb intézkedés híján azonban a magasabb szintű vezető mégis előnyhöz juthatna az alacsonyabb szintűekkel szemben. Ha ez nem kívánatos, a GYORSSIN olyan protokollt használ, mely egy vezetőnek az általa végzett munkaciklust követően csak akkor engedi meg a sin újbóli kisajátítását, ha az adott szegmenshez tartozó összes többi folyamódó is már egy-egy fordulót végigcsinált. Más esetben éppen az lehet

szükséges, hogy egyes vezetőknek sinelvételi előjoguk legyen. Ennek lehetővé tételére a GYORSSIN "biztosított eltérési protokoll"-ja szolgál. A GYORSSIN-nek tehát többféle arbitrációs mechanizmusa van.

Az arbitrációs szintet a vezető egy erre a célra szolgáló regiszterében őrzi, melynek tartalma átírható. Mód van tehát arra, hogy az erre kijelölt vezető a sint kisajátítva, a többi vezető arbitrációs szintjét meghatározza, ill. újra írja, azaz lehetőség van az arbitrációs szintek dinamikus kiosztására.

Amennyiben egy szegmenshez túl sok vezető csatlakozik, a várakozási idők hossza nyúlnak. Az idővesztés csökkentése érdekében ezért tanácsos a vezetőket a hozzájuk tartozó információ-forrásokkal együtt a hálózat szegmensei között minél egyenletesebben elosztani.

Vezetett a sint nem sajátíthatja ki, de szolgálatkérő jele útján kérheti a rendszerhez tartozó vezetőt (azok egyikét) meghatározott feladat elvégzésére. Amennyiben a szegmensnek csak egy megszakításkérő forrása van, és annak kérése mindig csak egyugyanazon vezetőhöz szól, úgy a megszakításkérő jel a sin SR szolgálatkérő vonalán át közvetlenül a megcélzott vezetőhöz továbbítható; a jel kibocsátóját és címzettjét ilyenkor egyaránt az SR vonal azonosítja. Általában azonban ennél bonyolultabb a helyzet. A szegmens szolgálatkérő jelei a közös SR vonalra lépnek, melynek állapotát a szegmenshez tartozó SZOLGÁLATKÉRÉS-KEZELŐ figyeli. Ez - ugyanugy mintha vezető volna - kisajátíthatja a sint. Ha szolgálatkérő jel hatására a sint kisajátította, identifikálja a megszakításkérő forrást, ami történhet akár lekérdezgetéssel, akár a TP kapcsok (l. 43. ábra) útján, vagy valami más alkalmas módon. Olyan szolgálati igény is jelentkezhet, amelyet maga a szolgálatkérés-kezelő is elvégezhet, általában azonban ún. megszakítási üzenet kibocsátása révén szólítja fel a szolgálatkérő jel által megcélzott vezetőt előírt feladatának elvégzésére.

A *sin* vonalait a 43. ábra sorolja fel jelöléseikkel, vezetékszámokkal, jeltípusuk és a rajtuk haladó jelet kibocsátó

lehetséges források megnevezésével. A táblázatot két részre osztó vastagabb vonal feletti 60 vezeték a keretszegmensnél és a kábelszegmensnél egyaránt megtalálható. Ezek egytől egyig valamennyi készülék egy-egy pontjára csatlakozó társasvezetékek (party lines). A táblázat alsó részében felsorolt vezetékek és kapcsok csak a keretszegmensnél vannak meg.

Funkcióikat tekintve

- arbitrációs
- szolgálatkérő
- kijelölő
- változó szerepkörű
- szolgálati

vezetékek csoportjai különíthetők el. A keretszegmensnél ezen kívül találjuk még a TX, RX vezetéket, melyeket diagnosztikai modulok és programok használhatnak, továbbá a két (egymástól független) gyöngyvezeték-láncot (daisy chain) alkotó DL, DR vezetékcsoportot, mely a normális GYORSSIN protokolltól független módon való információ továbbításra szolgál. A szegmensen belül minden csatlakozóhelynek (tehát nem a készüléknek!) egyetlen bináris földrajzi címe van, mely a keretszegmensnél a GA kapocspontok huzalozott kódolása révén rögzíthető. A TP kapocspont esetenként más-más információt hordozhat. Jelezheti pl. egy adott készülék szolgálatkérését (az SR-re adott jel mellett); e pontok letapogatása útján a szolgálatkérő készülék azonosítása végezhető.

A sin-arbitráció vonala a vezetők sinkisajátítási protokolljának eszközéül szolgál. Az egyes vezetékek szerepkörének leírására az arbitrációs eljárás ismertetése folyamán kerül sor.

A változó szerepkörű vezetékek csoportja képezi a sin gerincét. Az adatok és címek továbbítására egyaránt a 32 bit széles kétirányú AD átvivő vonal szolgál. E vonal feladatát - az átvitel irányától függetlenül - mindig a szint uraló vezető jelöli ki, a kijelölő vezetékek segítségével. Ha a vezető AS jelet bocsát ki, az AD címvonalként, ha pedig DS jelet, akkor adatvonalként működik. Az AS ill. DS jelre a kapcsolatban résztvevő vezetett az AK, ill. DK nyugtázó jellel válaszol.

Az AS, AK, DS, DK vezetékek nem csak a változó szerepkörű vezetékek szerepkijelölését, nem csak a vezető-vezetett kapcsolat kialakítását, hanem - szinkronizáló vezetékként - az átviteli folyamat időbeli lefutásának irányítását is végzik.

Azt a munkafázist, amikor az AD vonal cimvonalként funkcionál cimciklusnak, amikor pedig adatvonalként - adatciklusnak nevezik. A GYORSSIN-nek többféle címzés módja és többféle adatátvitelmódja van, és hogy adott esetben melyikről van szó, azt az MS vezeték harmas jelöli ki. MS jeltartalmának tehát a cimciklusban és az adatciklusban más-más jelentése van.

A PA paritásvezeték jelét mindig az az egység képezi, amelyik az AD vezetékre információt bocsát ki (adatot, címet, stb.), s mindig az AD párosszámu logikai 1 szintje mellé állít logikai 1 szintet (páros paritás). A PE jelet az AD vonal páratlan paritásának jelzésére állítja a szegmens készüléke; a vezető cim és írás ciklusokban, a vezetett olvasás ciklusokban generálja. Az a készülék, amelyik PE jelzését észleli, nem kell hogy paritásvizsgálatot végezzen.

Az RD vonallal az őt állító vezető a vele kapcsolatba kerülő készülékkel (vezetettel v. adott esetben vezetettként működő vezetővel) az AD, PE, PA vonalak átviteli irányát közli. Cim, ill. írás adatciklusban $RD=0$, ilyenkor a vezető állítja az AD, PE és PA vonalakat. Olvasás adatciklusban $RD=1$, ekkor a vonalakat a vezetett állítja.

Az SS vezetékek a vezetővel kapcsolatba lépő készülék mint vezetett állapotát leíró vektort továbbítják a vezetőhöz. Az AK vagy DK jellel válaszul érkező $SS=0$ állapot az jelenti, hogy az átvitel sikeres volt.

Az SR szolgálatkérő vonal szerepét az előző pont végén a szolgálatkérés mechanizmusának rövid leírásakor már ismertettük. Vissza van még a WT, RB, BH, EG szolgálati vezetékek feladatának tisztázása.

A WT vezeték jele visszaállíthatja, ill. leállíthatja a vezető időkifutásjelét és visszatarthatja a vezető időzítő átmeneteit, egyúttal indikálva, hogy a szokásosnál nagyobb, de megengedett késleltetés várható a várt válasz beérkeztéig. WT

diagnosztikai célra való felhasználása lehetővé teszi a rendszer lépésenkénti vizsgálatát.

Az RB jelet egy a rendszer üzemállapotáról megfelelő ismerettel rendelkező vezető (gazda) állíthatja abból a célból, hogy azt alapállapotába vigye vissza (nullázás).

A BH jelet a szegmens tartozék áramkörének részét képező logikai MENET/ÁLLÁS eleme állítja, ha MENET/ÁLLÁS kapcsolójának ÁLLÁS igényét észleli. Az AK-val együtt keltődik, a szegmens összes készülékének az inaktív ÁLLÁS állapotot jelezve. E jel a sin véletlenszerű zavaraitól (pl. hamis RB-től) véd.

A földrajzi cím képesítő EG vonalat vagy vezető, vagy a szegmens tartozék-áramköre állíthatja. Jelének hatására a sin-re csatlakozó készülékek hozzáfognak a pozíciójukon lévő, s a GA kapcsok kódolásával rögzített földrajzi címükkel való összehasonlításhoz (kábelaszegmensnél a hely szerinti címet kapcsolókkal kódolják).

A *tartozék-áramkörök (Ancillary Logic - Anc)* együttese ugyanúgy állandó alkotórésze a szegmensnek, mint a sin. A sin-re csatlakozó minden készülékre nézve közös, és különböző elemeirészt vesznek különböző szegmens-műveletekben. A 'tartozék áramkör' fedőnév alatt az alábbi funkcionális egységek csoportosulnak:

- arbitráció időzítő vezérlő (Arbitration Timing Control-ATC)
- földrajzi cím vezérlő
- rendszer-parola generáló (handshake generation)
- MENET/ÁLLÁS vezérlő (RUN/HALT control)
- sinlezárók
- soros-hálózati vonalcsatlakozás.

Az arbitráció időzítő vezérlő fogadja az AR vonalon át a vezetők részéről kibocsátott folyamodást, a vezetővel, ill. vezetőkkel együttműködve végrehajtja az arbitrációt, melynek eredményeképpen a sint az arbitrációs protokollban meghatározott módon átengedi a legmagasabb szintű folyamodónak.

A földrajzi cím vezérlő: speciális vezetett készülék. Feladata a földrajzi cím detektálása és az EG földrajzi képesítő

jel kibocsátása. EG jelkeltőből, szegmens-cim regiszterből, készülék azonosító regiszterből (ID), egy ugynevezett NTA regiszterből, vezetett modul vezérlőből, dekóldolóból és SS válaszlogikából épül fel. E felsorolt elemek egy része különben minden olyan modulban megtalálható, amelyik vezetett üzemmódban dolgozhat. Ha a földrajzi cím vezérlő az AD vonalon földrajzi címet észlel, EG jelet bocsát ki.

A rendszer-parola generáló az un. közzétett (broadcast) címzésmód használatakor szolgáltatja az AK, DK parolajeleket.

A MENET/ÁLLÁS vezérlő a MENET/ÁLLÁS kapcsolót figyeli, s annak ÁLLÁS helyzetét, azaz a szegmens ÁLLÁS állapotát az AK és BH vonalak állítása útján jelzi. A mondottakból következik, hogy minden szegmensnek van egy MENET/ÁLLÁS kapcsolója. E kapcsoló ÁLLÁS helyzetében nemcsak hogy mindenféle sinművelet lehetetlen, de a sinre csatlakozó készülékek a hamis jelek hatásától is védve vannak.

A GYORSSIN összes társasvezetékéhez, mindkét oldalon, illesztett sinlezáró csatlakozik.

A keretszegmens TX, TR soros vonalainak kábelcsatlakozási részleteit a szabványtervezet egyelőre még nem definiálta.

A kábelszegmenshez ugyanaz a tartozékáramkör járul, mint a keretszegmenshez, a soros-hálózati vonalcsatlakozás kivételével.

Az arbitráció az ATC tartozék áramkör, a sin-arbitráció vezetékai és a szegmenshez tartozó vezetők együttműködésével történik. Az eljárás megkezdésének előfeltétele a GK vonal alaphelyzete. Az ATC áramkör e feltétel mellett akcióba lép, ha az AR vonalon egy vagy több vezető egyidejűleg a sinért folyomodik. Az arbitráció ciklus az AG jel kibocsátásával kezdődik. Minden vezető előre meghatározott arbitrációs szinttel rendelkezik, amelyet képviselő 6 bites kódot a folyomodók az AG jel hatására egyidejűleg az AL vonalra adják. Következő lépésként valamennyi folyomodó - a nagyobb helyiértékek irányából indulva - bitenként végigvizsgálja az AL vonalon található *eredő* kódot, mely az egyes folyomodók által kibocsátott szintkódok VAGY kapcsolatu logikai egyesítéséből jön létre. A

vizsgálat folyamán az eredő kód és a folyamódó kódjának bitenkénti összehasonlítása történik. Az a folyamódó, amelyik az eredő kód valamelyik helyiértékén 1-t talál ott, ahol saját szintkódjában 0 áll, az kiáll a sinért folyó versenyből. Végül a legmagasabb arbitrációs szintű folyamódó marad egyedül. Mihelyt az ATC azt érzékeli, hogy a sin szabad ($AS=AK=WT=GK=0$), visszavonja AG jelét, mire a győztes folyamódó GK állításával válaszol, egyidejűleg elnyerve a sint. A győztes vezető mindaddig állítja GK-t, amíg csak nem hajlandó újabb arbitráció ciklust engedélyezni. Ez a pillanat rendszerint az általa lefuttatott szekvencia utolsó cím-fázisa után következik el, ami lehetővé teszi a következő arbitráció ciklus elkezdését, mielőtt még a sint momentán uraló vezető végleg befejezné munkáját.

A vázolt arbitrációs eljárás alapján két - egymással összhangban lévő - protokoll választható. Az egyik csak az arbitrációs szinteket veszi figyelembe, ami azzal a következménnyel járhat, hogy egyes alacsonyabb szintű vezetők elfogadhatatlanul hosszú időre kizárulnak a sin használatából. A másik az ún. "biztosított elérésű" protokoll. Ennél csak akkor bocsátható ki újabb AR jel, ha az AI arbitráció folyamódás tiltó vonal logikai 0 állapotú. Az AI vonalat ATC állítja át az arbitráció ciklus kezdetén, és csak akkor állítja vissza, ha már nincs folyamódó. Így valamennyi az AI átállítása előtt folyamódó vezető sinigénye kielégítést nyer, és e folyamatot egy újabb vagy egy már kiszolgált vezető ismételt folyamódása nem zavarhatja meg. Egyugyanazon szegmensben belül mindkét protokollnak eleget tévő vezetők is lehetnek; ezek csak az AR állításának feltételében különböznek egymástól.

Vezérlő és állapot regiszterek (Control and Status Registers - CSR). CSR regiszternek a szegmensre csatlakozó készülékek azon speciális célú regisztereit nevezik, amelyek nem a modul alapfeladatával kapcsolatos adatok, hanem a modul tulajdonságára, állapotára, a feladatvégrehajtás módjára vonatkozó és más szolgálati információ (időszakos) őrzésére szolgálnak. A GYORSSIN szabvány az adatregiszterek és a CSR regiszterek kö-

zött még kezelésmódjukat illetően is éles különbséget tesz, és pedig sokkal határozottabban, mint a CAMAC.

A CSR regiszterek négy típusa különböztethető meg.

- alap CSR
- program CSR
- paraméter CSR
- használói CSR

A szabvány mindnégy regisztertípus számára olyan nagy címtartományt tart fenn, melynek alapján a készüléktervező méltán azt érezheti, hogy az általa alkalmazható CSR regiszterek száma tekintetében nincs korlátozva.

Az alap CSR címtartományba a tulajdonképpeni vezérlő és állapotregiszterek tartoznak, melyek közül nem egynek használata kötelező.

A program CSR tartományba tartozó regisztereket főként maga a készülék használja; programok és táblázatok őrzésére szolgálnak.

A paraméter CSR tartomány regiszterei az adott készülékre vonatkozó statikus vagy ritkán változó információt tárolhatnak pl. kalibrációs állandókat és gyártói adatokat. Ezen információk a GYORSSIN-en át többnyire nem változtathatók (csak olvashatók), átírásuk általában speciális módon (PROM átírás) történik.

A használói CSR tartomány szabadon felhasználható.

Az alap CSR címtartományon belül a regiszterek négy alcsoportba oszlanak el. Az első nyolc címen azok találhatóak, amelyeket főként vezetettek, a következő nyolc címen pedig azok, amelyeket főként vezetők használnak. A további helyeken a használó rendelkezésére álló címek, a szabvány által további fejlesztés céljára tartalékolt címek, SR forrás és maszk regisztereinek címei, valamint a több szegmensből álló hálózat kapcsolat-mechanizmusának regisztercímei találhatóak.

Minden készülékben megtalálható a CSRO, minden logikailag címezhető készülékben a CSR3 logikai címregiszter, minden vezetőben a CSR8 arbitráció szint regiszter, minden időzítővel ellátott készülékben a CSR9 időzítésvezérlő regiszter.

A legegyszerűbb funkciókat teljesítő készülékben is meg kell lennie a CSR0 regiszternek. E regiszter két, egyenként 16-16 bit szélességű része egymástól független információt hordoz. A CSR0 [31:16] bit az adott készülék ID azonosító (identification) kódját tartalmazza, mely minden készüléktípusra nézve egyedi. Ha az adott készüléket csak a legkisebb mértékben is módosítják, azzal együtt azonosító kódját is módosítani kell. A 16-ból a magasabb helyiértékű 12 bit tartalmát minden egyes készüléktípushoz a NIM bizottság adja ki, és 4 bit, azaz 16 variációs-módosítási lehetőség marad a használó részére. Amikor a szegmens egyvalamely konkrét hálózatban foglal helyet, az egyes pozíciókon található készülékek ID kódja kiolvasható, miáltal a rendszert kezelő számítógép pontosan azonosíthatja az egyes helyeken lévő készülékek típusát (és variánsát).

A CSR0 [15:0] szakasza állapot és vezérlő biteket tartalmaz, melyek mindegyikének pontos szerepét a szabvány rögzíti.

Valamennyi CSR regiszter részletes felsorolása tulmenne általános jellegű leírásunk keretein.

5.4 CIMZÉSMÓDOK

A szegmensekből szervezett hálózat készülékei többféle módon érhetők el. Hivhatók a hálózat földrajzi címeivel, amikor is mindig az a készülék válaszol, amelyik a hívó által kibocsátott földrajzi címnek megfelelő hálózati ponton csatlakozik, vagy hívhatók logikai címekkel, mely esetben mindig az a készülék válaszol, amelyiknek a logikai címe momentán a vonalon van. Míg tehát a földrajzi cím a hálózathoz tartozik, a logikai cím a készülékhez.

Logikai címzés. A logikai cím a készüléké ugyan, de nincs ahhoz rögzítve. A hálózathoz kiemelt készüléknek nincs semmiféle címe (csak belső címei), a logikai címet a hálózathoz való csatlakoztatást követően, a rendszer kezdeti előkészítésekor (kezdetelés) kapja CSR3 regiszterébe való betöltéssel, s e regiszter tartalma a rendszer működése során meg is változtatható.

A 32 bites logikai cím két fő zónája: a DA készülékcím és az IA belső cím (44. ábra). A készülékcímnek két alzónája van, melyek közül a GP csoportcím a rendszeren belül a szegmenst jelöli meg, az MA pedig modulcím, mely a készüléket a szegmensten belül választja ki.

A zónák elhelyezkedési sorrendje a nagyobb helyiértékű bitek irányából a kisebbek felé haladva: GP, MA, IA, amint azt az ábra is mutatja. A 32 bitnek az egyes zónák közti megosztását a szabvány nem rögzíti, az a mindenkori körülményekhez alkalmazkodva választható. Egyszerű esetben pl. a készülék belső címeinek IA zónája nullára zsugorodhat, ha egynél több elérendő elemmel nem kell számolni. Változhat a GP és MA zónák szélességének egymáshoz viszonyított aránya is.

Meg kell jegyezni, hogy a logikai címnek tulajdonképpen csupán a DA zónája logikai. Az IA belső cím értékei funkcionálisan a készülék tervezésekor rögzítődnek, azaz IA egy adott modulon belül, mindig ugyanazt az elemet (pl. regisztert) képviseli, és ezt az identitást a készülék logikai átcímzése nem érinti.

A logikai cím jelentését a MS vonalak tartalma közelebbről értelmezi. E szerint a cím lehet egy készüléknek szóló, és lehet több készüléknek egyidejűleg szóló, un. "közzétett" (broadcast) cím.

Ugyancsak MS két egymástól teljesen független logikai címet határoz meg. Az egyik címtérben a logikai cím adathelyet jelöl meg, a másik címtérben CSR regisztert.

A GYORSSIN alapműveletei mindig címciklussal kezdődnek, és egy vagy több adatsiklussal folytatódnak. A kezdő címciklust elsődleges címzésnek nevezik. A 45. ábrából kitűnik, hogy az MS mely értékeihez milyen címtér, ill. milyen címzés mód tartozik.

Amennyiben az elsődleges címzést követő adatsiklust MS<2> minősíti, akkor a vezető által e ciklusban állított adat nem adat, hanem címként értelmeződik. E címet másodcímnek, a vele végrehajtott műveletet másodcímzésnek nevezik. A másodcímzés egy-egy 32 bites új belső címtartományt tesz hozzáférhetővé,

mégpedig az adattér és a CSR tér számára egyaránt.

Az adattér cimmezésének a CSR térben való megismétlése nemcsak a belső címtartomány megkétszereződését, hanem az adatregiszterek és a CSR regiszterek kezelésének különválasztását is maga után vonja, ami a rendszer megbízható működésének fontos feltétele.

A teljes logikai cím és az esetleges másodcím teljes egészében a megcímezett készüléken belül dekódolódik. A készülék csak akkor lép kommunikációs kapcsolatba hívójával, ha CSR3 regisztere DA zónájának tartalma megegyezik az AD vonal megfelelő vezetékének információtartalmával. A nullás készülékcím logikai címként nem használható, nehogy összetéveszthető legyen a földrajzi címekkel.

Földrajzi címzés. A 32 bites elsődleges cím 2^{32} kombinációja közül a 8 bittel leírható első 256-ot a szabvány speciális célokra tartja fenn. Ezek közül is az 5 bittel leírható első 32 kombináció, azaz a 0 - 31 cím: szegmens belüli speciális készülékcím, a szegmens készülékeinek földrajzi címe. Készülék a címet földrajzikiént csak abban az esetben érzékeli, ha egyidejűleg az EG vonal erre képesíti. Az EG vonalat vagy a megcímezett készülékekkel egy szegmenshez tartozó vezető, vagy a szegmens tartozék áramkörei közül a földrajzi cím vezérlő állítja. Az utóbbi akkor állítja az EG vonalat, ha az elsődleges címzés során az AD vonalon a 46. ábrán látható cim-alakok egyikét észleli.

Az EG jelének hatására valamennyi a sinhez csatlakozó készülék összehasonlítja a cimvonalára érkező címet csatlakozóhelyének rögzített földrajzi címével. Az a készülék, amelyik megegyezést talál, kapcsolatba lép a címet kibocsátó forrással. A kapcsolat létrejötté, azaz az elsődleges (földrajzi) címzés ciklusának befejezését követően, másodlagos címzési fázis vagy szabályos adatciklus következhet.

Közzétett címzés. Az elsődleges címzés nem csak egy, hanem egyidejűleg több készüléket is megszólíthat és kapcsolatba hozhat a cím kibocsátójával közzétett címzési mód használata esetén. A címet az öt kibocsátó vezető az MS vonalak útján teheti köz-

zé, megválasztva egyuttal az adat-, ill. CSR teret is. Közzétett címzéssel végzett jellegezetes műveletek: a készülékek szinkronizálása és a regisztertömbök törlése.

A közzétett cím két feladatot jelöl ki: Meghatározza, vajon a hálózaton belül a közzétett cím egy bizonyos szegmensnek, a rendszerben rögzített információ vezérelte minta szerint valamennyi szegmensnek vagy egy specifikáltnak és azon túl valamennyi szegmensnek szól-e. A másik feladat, hogy az érintett szegmenseken meghatározzon egyvalamely készülékosztályt vagy végrehajtandó műveletet. Előirt műveletek lehetnek: szórvány adatok lekérdezése, TP kapocs föltétel nélküli állítása, szolgáltatáskérésrel kapcsolatos TP kapocs állítás, készülékcímzés a TP kapocs vezető általi állítása révén. A készüléktervező számára nyolc feladatkód áll rendelkezésre speciális feladatok kitűzésére.

Közzétett címzésmód használatakor az egyes munkaciklusok megfelelő parola válaszeleit a megszólított készülékek szabályszerűen képezik ugyan, de e jelek nem jutnak ki az AK, DK vonalra. A parola válaszeleket ilyen esetben a rendszer-parola jelgeneráló tartozék áramkör képezi.

5.4 MŰVELETEK

A GYORSSIN műveletek során valamely vezető és egy vagy több vezetett közötti információ-átvitel történik. Az "információ-átvitel" kifejezést itt az "adatátvitel"-nél tágabb értelemben kell értelmezni.

A GYORSSIN műveletet mindig vezető indítja, s az minden esetben elsődleges címzesciklussal kezdődik. Megkülönböztethető: null adatciklus, egyadatciklus, több adatciklus, vegyes és többszörös művelet.

A nulladatciklus művelet esetében az elsődleges címzést nem követi adatciklus. Lévén, hogy a cím ciklus parola üzemmódban zajlik le, e művelet egyrészt egyvalamely készülék jelenlétének megvizsgálására, ill. e készülékkel való kapcsolatteremtés lehetőségének ellenőrzésére szolgálhat, még mielőtt a datátvitelre kerülne sor.

Az egyadatciklusú művelet, mint nevéből is következik, egyetlen adatnak tetszőleges címre vagy címről való átvitelére szolgál (rendezetlen elérési sorrendű átvitel – random access transfer).

A többadatciklusú műveletek esetében az elsődleges címzést mindig egynél több adatciklus követi. Ebbe a kategóriába tartoznak a parola üzemmódban, ill. a szinkron üzemmódban végrehajtott blokkátviteli műveletek.

A vegyes műveletek esetében az elsődleges címzést egy sor egyes, ill. blokkátvitel, írás, ill. olvasás keverten követi.

Vegyes művelet pl. az olvasd-módóítsd-írd (ready-modify write). Vegyes műveletek továbbá a másodcímzéssel végzett műveletek is.

A felsoroltakon kívül vannak még a többszörös műveletek. Többszörös GYORSSIN műveletet végezhet a vezető, ha nem enged el a már kisajátított sint. Ilyen módon egyes, blokk, ill. vegyes adatciklusokból álló sorozat jöhet létre különböző vezetettek készleteivel anélkül, hogy más potenciális sinvezető azt megszakíthatná. A vezető az ilyen műveletsort oly módon bonyolíthatja le, hogy a GK=1 jelet (1. arbitráció) fenntartja.

Adatciklusok végrehajtása során azok az adatbitek, amelyek a (megcímzett) vezetettnél nem szerepelnek (nincsenek), olvasás művelet során nullát adnak, írás műveletek kapcsán pedig figyelmen kívül maradnak.

Elsődleges címzés. A sint birtokoló vezető és a vezetett vagy vezetettek közötti kapcsolat az elsődleges címzés fázisában jön létre. A vezető a műveletet csak akkor kezdheti el, ha előzőleg GK=1 jelet állítja. Ennek megtörténte után állítja RD=0 mellett az MS és AD vonalat, s szükség szerint a PA, PE és EG vonalat. A nevezett vonalak beállítást kivárva, a vezető az AS cimszinkron jelet bocsátja ki, melynek hatására a vezetettek megvizsgálják az AD vonal tartalmát, az MS és EG vonalak állapota által előírt módon vetve össze saját címükkel. Ha a vezető csupán egyetlen vezetett szolgáltatást igényli, úgy ez felismerve saját címét, AK címnyugtázó jelet küld a hívó

vezetőnek, s ezzel egyuttal a hívó és a hívott fél logikai összeköttetésbe kerül. Közzétett címzés használatakor az AK jelet a parolajel generáló tartozék-áramkör szolgáltatja, ami ez esetben a cím kibocsátó vezető részére a cím közzétételének megtörténtét jelzi.

A GYORSSIN előírásának megfelelően az AS és AK jel az elsődleges címzéssel megkezdett művelet egész időtartama alatt állítva marad. Az AS/AK "zár" megakadályozza, hogy bármely készülék beleavatkozhasnék a logikai kapcsolatban lévő készülékpár (készülékek) munkájába; mindaddig amíg az AS/AK zár érvényesül, kereteik között tetszőleges (célszerűen szabványos) protokoll alkalmazható.

Az elsődleges címzéssel megkezdett művelet akkor végződik, amikor a vezető visszaállítja az AS=0 szintet, s válaszképpen az AK-t adó egység (vezetett vagy tartozékáramkör) AK=0-t állít.

AS=AK=0 beállítást követően, bizonyos idő múltán, a vezető újabb elsődleges címzési ciklust kezdhet, ha egyébként fenntartotta a GK=1 állapotot.

Egyadatciklusu művelet. A vezető, miután az elsődleges ciklus folyamán megkapja az AK nyugtázó jelet, az AD vonalról visszavonja a címet, hogy e vonalat adatátvitel céljára használja. Irásciklusban RD=0, olvasásciklusban RD=1-t állít. Irásciklus esetén az AD vonalra helyezi az átviendő adatot, s ha paritásképzés szükséges PE=1-t a PA vonalra pedig AD tartalmától függő 0/1 paritásbitet. Az MS vonalra '0' kód kerül. Az adatátvitelt (írás) ezután a DS adatszinkron jel kibocsátása indítja. DS hatására az AD-n lévő adat a vezetett megcímzett regiszterébe iródik, amiről a vezetett DK nyugtázó jelével értesíti a vezetőt.

Olvasás (RD=1) esetén minden ugyanugy zajlik le, csupán az adatot a vezetett szolgáltatja a hozzá érkező DS hatására, s küldi DK társaságában az őt hívó készülékhez.

Egyadatciklusu parolázó olvasás művelet impulzus-ábráját szemlélteti a 47. ábra, úgy, ahogy az a vezető oldalán látható.

Korábban az MS kód jelentését inkább a címzéssel kapcsolatban érintettük, most a különböző adatátvitel-módok áttekintésével kapcsolatban ismételten ráirányítjuk a figyelmet a 45. ábra táblázatának tartalmára.

Többadateciklusu műveletek, blokkátvitel. Ide tartoznak a parola üzemmódu és a szinkron üzemmódu blokkátviteli műveletek. A blokkátvitel elsődleges címciklust követő, azonos típusu, tetszőleges számú (vagy csak írás vagy csak olvasás) adateciklusból áll.

Parola üzemmódu blokkátvitelnél az ismétlődő adateciklusok DS, DK jelek kíséretében hasonló módon zajlanak le, mint az egyadateciklusu műveleteknél. A sebesség növelése érdekében a DS/DK jelpár mindkét éle igénybe vehető; a parola védelem ebben az esetben is fennmarad.

A parola üzemmód minden adatátviteli ideje legalább kétszerese a vezetőt és vezetettet összekötő utszakasz késleltetési idejének. Ahhoz ui., hogy a vezető egy újabb adatot bocsáthasson ki, előbb meg kell kapnia a DK jelet. Olvasásnál is DS kiküldése után az adat DK társaságában az összekötő ut kétszeres késleltetési idejével érkezik meg.

Amennyiben a parola üzemmód lehetőségeit meghaladó adatátviteli sebesség szükséges, szinkron blokkátvitel alkalmazható. Szinkron üzemmódban az adatátvitel ritmusát csupán a vezető ill. a vezetett adó, ill. vevő áramköreinek sebességi korlátai, ill. a közöttük fekvő átvivőszakasz sáv szélessége határoolja. Az összekötő szakaszon így egyidejűleg (egymást követően) több adat is haladhat. Írás műveletnél ekkor DS nem parolajelként, hanem szinkron részjelként szolgál, DK pedig figyelmen kívül marad, vagy a vezető a lefutott adateciklusok számlálására használhatja. Olvasás művelet esetén DS - órajelként - vezérli a vezetett adat kibocsátási ritmusát, míg DK a vezető oldalán a szinkron részjel szerepkörét tölti be.

Noha a parola üzemmód kisebb adatátviteli sebességet tesz lehetővé, rendelkezik néhány olyan előnyös tulajdonsággal, amit a szinkron mód nélkülöz. Elsősorban is megbízhatóbb. Ha szükséges, bármelyik oldal szünetet tarthat (pl. dinamikus tár-

elemeinek felfrissítése céljából). Lehetővé teszi továbbá, hogy a kommunikáló felek bármelyike a tervezettnél korábban (pl. puffer túlfolyás esetén) befejezze működését és pedig úgy, hogy emellett mindkét fél pontosan ismeri a sikeresen átvitt szavak számát. Szinkron üzemmódban a vezetőknek pontosan ismernie kell a vele kapcsolatban álló vezetett adottságait, általában a vele kapcsolatban álló egész adatösvény átbocsátó képességét; csak ezen ismeret birtokában választhatja meg a DS órajel frekvenciáját. Ha az áramköri adottságok nem tesznek lehetővé lényegesen nagyobb átviteli sebességet szinkron üzemmódban, mint ami parola üzemmódban lehetséges - szinkron üzem módot alkalmazni értelmetlen.

A vezető, ha munkájának befejezése nem sürgős, és a blokkátvitel ideje alatt egy másik AR folyamódását észleli, a sint átengedheti, s majd a félbeszakított blokkátvitelt később befejezheti.

Vegyes és többszörös műveletek. A vegyes művelet az elsődleges cimciklust követő tetszőleges számú másodlagos cim-, ill. adaticiklusból áll. Pl. az olvasás-módosítás-írás művelet során a vezető kibocsátja a címet, olvassa az adatot az RD vonalat $1 \rightarrow 0$ állítja és a módosított adatot a változatlan (vezetett) címre írja (48. ábra). Mivel az egész művelet során az AS/AK zár hat, más készülék közben nem használhatja a sint.

Ebbe a csoportba tartoznak a másodlagos címzést használó műveletek is, melyek során MS értelmezése alapján az elsődleges címzést követő adaticiklusban a vezető által kibocsátott "adat"-ot a vezetett címként kezeli.

A vegyes műveletek blokkátviteli szakaszokat éppúgy magukba foglalhatnak, mint egyes adatok átvitelének sorozatát. Az egyes blokkátviteli szakaszok, az egyes adatok átviteli ciklusai között az átvitel iránya változtatható.

A többszörös műveletek során egy sor - általában vegyes - művelet hajtódik végre egymás után a nélkül, hogy a vezető közben lemondana a sin birtoklásáról. Ez a művelet típus jól alkalmazható a vezetettek bizonyos oly készletének szinkronizálására, mely készlethez tartozó készülékek egyébként több procesz-

szor között oszlanak meg. A szinkronizálás anélkül hajtható végre, hogy közben más processzorok beavatkozhatnának. Ez a mechanizmus a GYORSSIN hálózat különböző szegmensein csatlakozó vezetettekre is kiterjed.

Szórványadat felkeresés. Mód van arra, hogy a szegmens készülékei által szolgáltatott szórványadatok helyét a vezető felderítse. E célra a sinrendszer T kapcsait használják fel. A T kapocs maga jelezheti, ha a készülék szórványadatot őriz. A vezető által közzétett parancs hatására a T kapcsok állapota az AD vonalra kerül, mégpedig úgy, hogy az n. készülék T kapcsa AD n. vonalára kapcsolódik. AD tartalmának kiolvasása révén a vezető informálódik arról, hogy mely készülékek őriznek kiolvasandó adatot.

A műveletek lefolyásáról a címzettek a vezetőt az SS-kód kibocsátása útján informálják. SS révén kap teljes képet a vezető a nulladatciklusú művelet lefolyásáról is. Mivel SS sokkal inkább a szegmensekből álló hálózatra, semmint az egyes szegmensre vonatkozó információt nyújt, e kód egyes értékeinek jelentését valamivel később ismertetjük.

5.5 GYORSSIN HÁLÓZAT SZEGMENSEKBŐL

A szegmensből mint alapelemből tetszőleges alakzatú hálózatok szervezhetők, melyen belül a szegmensek egymás közötti kapcsolatai szabadon választhatók meg. A szegmensek páronkénti összekötésére a SZEGMENSKAPOCS (SEGMENT INTERCONNECT-SI) szolgál, amint azt a 49. ábra szemlélteti. Az itt alkalmazott szegmenskapocs szimmetrikus, mindkét irányból és mindkét irányban azonos tulajdonságú. A szegmenskapocs igénybevételével az 1. szegmens vezetője hívhatja a 2. szegmens készülékét, a 2. szegmens készüléke pedig igényelheti az 1. szegmens vezetőjének szolgálatát, s a szimmetriából következőleg ugyanez lehetséges fordítva is. A szimmetrikus SI ezért mindkét irányban kell hogy rendelkezzen a vezető és a vezetett tulajdonságaival is. A szegmenskapocsnak mint vezetettnek van címfelismerő áramköre. A cím, amit az 1. szegmens oldaláról felismer nem más,

mint a 2. szegmens GP címe (l. 44. ábra), amit pedig a 2. szegmens irányából ismer fel, az az 1. szegmens GP címe. Ha az 1. szegmenshez tartozó vezető a 2. szegmens készülékét hívja, az SI e hívást (címet) felismerve, a 2. szegmens sinjéért folya-
modik ugyanugy, mint a 2. szegmens bármely vezetője. A kívánt kapcsolat akkor jön létre, amikor az arbitrációs mechanizmus SI-nek juttatja a 2. sint.

Az SI mint általános célra szolgáló hálózati elem eleve nem ismeri sem az 1., sem a 2. szegmens címét. E címeket a rendszer kezdetelése során kapja meg.

Az 50. ábra az SI alkalmazásával szegmensekből szervezett hálózat egy általánosabb példáját mutatja be. Látható, hogy a szegmens nem csak egy, hanem több más szegmenssel is közvetlen összeköttetésben állhat, de minden közvetlen összeköttetéshez külön szegmenskapocs szükséges. Valamely szegmens a hálózat egy más szegmensével nem csupán közvetlen kapcsolatban állhat, az egymástól távolfekvő szegmensek a közöttük lévő szegmenseken át is logikai kapcsolatba léphetnek egymással. Ez oly módon történik, hogy a hívó vezető a hívott vezetett irányába vivő utvonal mentén először a szomszédos szegmens sinjéért folya-
modik, majd ezt elnyerve ezen át az utvonal mentén fekvő második sinért, és így tovább, míg végül azt a sint is el nem nyeri, melynek hatáskörébe a hívott vezetett tartozik. Látható, hogy a logikai összeköttetés arbitrációs eljárások során át valósul meg.

Egy szegmenstől egy másikhoz több utvonal vezethet, másrészt a hívó vezető a hívott készülék címét bocsátja ki, melyben GP a cél-szegmens címe, nem pedig a szomszédos szegmenseké. Az arbitráció hosszadalmassága miatt fontos, hogy a hívó és a hívott közötti kapcsolat a hálózat lehetséges utvonalai közül a lehető legrövidebb kiválasztásával jöjjön létre. De hogyan ismerik fel egyáltalán a közbenső szakaszok szegmenskapcsai a távolabbi szegmenshez irányuló címet, és miként teljesül a legrövidebb ut kiválasztásának feltétele? Mindez a szegmenskapcsokba a rendszer kezdetelése során tárolt iránytérkép alapján történik. A szegmenskapcsok mindkét portája

(bejárata) rendelkezik bizonyos kapacitású tárolóval. E tárolóba mindazon szegmenscímek elhelyezhetők, melyekre nézve egy adott szegmenskapocsnak szabad utat kell adnia. Az 50. ábra hálózatán pl. az 1. szegmenstől a 2.-hoz öt különböző utvonal vezet.

1. utvonal SI12
2. utvonal SI15 → SI45 → SI24
3. utvonal SI15 → SI56 → SI26
4. utvonal SI14 → SI24
5. utvonal SI14 → SI45 → SI56 → SI26

Az iránytérképet úgy kell létrehozni, hogy konfliktusok ne adódhassanak. Adott esetben az öt közül csak egy valósítható meg. Ha a választott utvonal a 2. akkor az SI15, SI45, SI24 szegmenskapcsok mindegyikének az 1. szegmens felé eső oldala fel kell hogy ismerje és az arbitrációs láncon át kell hogy továbbítani tudja a 2. szegmens hívott készülékének címét. Ésszerűbb természetesen a legrövidebb (1) utvonalat választani, mely esetünkben az SI12-n át vezet. A szegmensportákban primitív esetben huzalozással, kapcsolók útján, célszerűbben ROM tárban rögzíthető az iránytérkép, általánosabb esetben pedig RAM tárba írható be, ami dinamikus átprogramozást tesz lehetővé. Az egységes programozás érdekében célszerű, ha az egyes szegmenskapcsok portáin elhelyezett információ alakilag azonos, s felöleli az összes a hálózatban helyetfoglaló szegmens címét. E címek közvetlenül címzik a portatárat. Ha egy adott címen a portatár tartalma 1, ezt a címet az adott szegmenskapocs továbbítja a következő szegmensre (arbitráció után), ahol pedig 0 ott elzárja az utat.

Egymástól távol eső szegmensek, szegmensekből álló részhálózatok kábellel egyesíthetők. A kábel egy-egy szegmenskapocs külső pontjára csatlakozik, amint azt az 51. ábra szemlélteti. A kábelre és csatlakozásmódjára a szabvány nem ad előírást, az egységesség érdekében azonban célszerű kábelszegmenst alkalmazni. Kábelszegmens alkalmazása egyebek mellett azzal az előnnyel is jár, hogy reá nem csak szegmenskapcsok (szegmensek), hanem különböző, a GYORSSIN előírásainak eleget tévő, készülékek

közvetlenül is csatlakozhatnak.

A GYORSSIN előírásai alapján szervezett bármely hálózathoz tartoznia kell egy olyan számítógépnek, mely teljes mértékben ismeri a rendszer strukturáját. E gépnek hozzá kell férnie a hálózat összes szegmenséhez, és pontosan tudnia kell a szegmensek összeköttetésének módját - az iránytérképet. Ez a számítógép, melyet GAZDAGÉP-nek vagy röviden GAZDÁ-nak neveznek, kezelheti a rendszert, tudatva az összes szegmenskapocs mindkét portájával, milyen műveleteket (cimeket) kell továbbítani a tuloldalán fekvő szegmensre. Földrajzi címzést használva, a GAZDA azonosíthatja a hálózat egyes pontjain fekvő készülékeket, és szükség szerint az egyes készülékekhez logikai cimeket rendelhet.

A gazda csatlakozásmódja a hálózat alakzatától függően többféle lehet. Egy lehetőséget az 51. ábra mutat; a gazda itt PI processzor-csatlakozón át közvetlenül a hálózat kábelszegmensére csatlakozik.

Kábelszegmens használható a (pl. fa strukturájú) hálózat két egymástól távolos szegmense közötti közvetlen kapcsolat fenntartására, mely szegmensek egymással gyakori kapcsolatban állnak.

Midőn a vezetők valamelyike gyorsin műveletbe kezd, egyúttal mindig megindítja belső válaszidő-mérő óráját, mely annak a szegmensnek a kifutási idejére van beállítva, amelyikhez a szóban forgó vezető tartozik. Ha a megkezdett művelet saját szegmensen belüli, a megcímzett készüléknek a kifutási idő lejártá előtt kell reagálnia. Ha azonban a művelet végrehajtásához egy vagy több szegmenskapcsón kell "áthatolni", vagyis a megcímzett valamely távolabbi szegmensen fekszik, úgy a helyi kifutási idő nyilvánvalóan nem lesz elegendő az egyébként normálisan folyó művelet során a megcímzett válaszához beérkezéséhez. Ilyenkor a hívó vezetőt figyelmeztetni kell, s ez az által történik, hogy minden a művelet által érintett SI szegmenskapocs WT várató jelet állít a hozzá forduló szegmensre. Egyúttal, kifutási időmérőt indít azon szegmensre irányulóan, amelyekre a műveletet átvezeti. Ezek a várató jelek az SI egysé-

geken át láncszerűen érvényesülnek, egészen a műveletet indító vezetőig. Amelyik szegmens WT jel áll, ott az időmérés szünetel. Kezdetben tehát csak a lánc utolsó szemét képező SI időmérője dolgozik, majd, ahogy a hívásra adott válasz visszafelé halad és az egyes szegmensek WT jelei visszaállítódnak, indulnak az órák egyik a másik után, végül a hívó vezetőé is. A WT jelek sorjában akkor is visszaállítódnak, ha pl. a láncszem utolsó tagját képező szegmensnél az időmérő kifutási időt jelez, vagy ha valamelyik közbenső szegmenskapocs nem állít várható jelet. A lehetőségek végiggondolhatók.

A hálózat az iránytérkép által kijelölt utvonalakon transzparens. Valamely vezető az engedélyezett utvonal mentén fekvő, tőle független szegmens készülékét ugyanugy hívhatja, mintha az saját szegmenséhez tartoznék. Azonos módon, azonos protokoll szerint történik e "távoli" készülék címzése (földrajzi, logikai, közzétett), valamint a vele végzett adatátvitel (egyciklusú, blokkátvitel parola, ill. szinkron üzemmódban; vegyes és többszörös műveletek) mintha a készülék a hívó vezetővel egy szegmenshez tartoznék.

5.6 ÁLLAPOTVISSZAJELZÉS

A GYORSSIN műveletek egyes fázisainak lefolyásáról a műveletet indító vezető elemi módon az AK, DK jelek, részletesebben pedig az SS kód útján kap információt. Nyilvánvaló, hogy ha az AK, DK jel elmarad, ez a megkezdett művelet elakadását jelzi. Ilyen eset áll elő pl. ha az elsődleges cimciklusban PE=1 paritásvizsgálatra képesít, s e vizsgálat kimenete (a címzettnél) negatív; AK válaszjel nem képződik.

Az SS kód különböző értékeinek jelentését az 52. ábra sorolja fel az elsődleges cimciklusra és az adaticiklusokra vonatkozóan egyaránt. E kódot vagy a hívó szegmenséhez tartozó megcímzett készülék, vagy az ide tartozó s a címet felismerő szegmenskapocs küldi a hívóhoz.

Elsődleges cimciklusban az SS=0, ill. 5 válasz (bár egymástól eltérő értelemben) azt jelenti, hogy a hívó és a hívott

között a kapcsolat létrejött. Az SS=1,2 és 3 válasz a hívás különböző okokból (1. 52. ábrát) való visszautasítását jelzi.

Adatciklusban:

- SS=0 Hibátlan művelet. E mellett előfordulhat, hogy a vezető a kiolvasott adat paritáshibáját észleli.
- SS=1 Foglaltság jelzés. A hívott sem nem fogad, sem nem állít adatot.
- SS=2 Blokk-végződés. Blokk átvitel a végéhez ért. Adat nem fogadható és nem állítható.
- SS=3 Érvénytelen művelet. RD, MS vonalak és a belső cím (IA) a megcímzett készülék által végre nem hajtható műveletet határoz meg. Pl. írás művelet csak olvasható regiszter címére; blokkátviteli utasítás ilyen üzemmódot teljesíteni nem képes készüléknek; nem létező belső cím használata. Érvénytelen műveletek adódhatnak MS értékének sérüléséből vagy helytelen megválasztásából, helytelen címválasztásból stb. kifolyóan is.
- SS=4 Adathiba (elvetéssel). Irásciklusban a vezetett a paritással vonatkozásban nem lévő adathibát észlel, s ezért visszautasítja az adatot. A megcímzett regiszter tartalma e mellett változatlan marad.
- SS=5 Adathiba (elfogadással). A vezetett a paritással vonatkozásban nem lévő adathibát észlel. Írás művelet esetén az elfogadott adat hibás minősítésű. Olvasásnál a vezetett az általa nyújtott adatot hibásnak nyilvánítja. Ez az SS válasz érvénytelen másodlagos cím írásánál ill. olvasásánál adódik.

SS=6 Paritáshiba (elvetéssel). A címzett - irás művelet során - PE=1 értéke mellett, az adat paritáshibáját észleli. A címzett regiszter tartalma változatlan marad.

SS=7 Paritáshiba (elfogadással). Írásnál jelentése ugyanaz, mint SS=6-é azzal a különbséggel, hogy a címzett az adatot az általa észlelt paritáshiba ellenére elfogadja. Olvasás művelet esetén a vezetett jelzi, hogy az általa kibocsátott adat paritáshibás.

A belső cím címmutatója automatikusan általában csak akkor változhat, ha a vezetett elfogadta vagy továbbította az adatot.

5.7 SZOLGÁLATKÉRÉS KEZELÉS

Általános értelemben a GYORSSIN "megszakítás": valamely készülék szolgálatának egy másik általi igénylése vagy egy készüléknek másik általi valamire való figyelmeztetése.

A szolgálatra felkért készülék az esetek többségében a GYORSSIN-től független megszakítási mechanizmussal rendelkező processzor vagy számítógép. A GYORSSIN szolgálatkérését előírt módon kell illeszteni a számítógép (processzor) megszakítási mechanizmusához. A GYORSSIN szabvány (tervezet) két módszert ajánl:

Az első módszer valamely szolgálatkérés kezelő készülék (számítógép, processzor) speciális vezérlő regisztereibe való beírási műveleteket tételez fel. A másik módszer a csupán vezetett szerepkörét betölteni képes készülékek valamelyikének az erre szolgáló SR vonalra állított szolgálatkérő jeléből indul ki.

Szolgálatkérés megszakítási üzenettel. Szolgálatot, de üzenetátadást is csupán vezető típusu készülék hajthat végre. Megszakítási üzenettel tehát csak vezető kérhet vezetőtől szolgálatot. A megszakítási üzenetnek pontosan meghatározott formája és tartalma van, s az a szolgálatot végrehajtó vezető

rögzített című CSR regisztereibe kerül. A szolgálatkérés oly módon történik, hogy az igénylő készülék a sínért folyamodik, majd amikor azt elnyeri, a megszakítási üzenetet szavanként vagy blokkban a szolgálatra felkért készülék e célra rendelt CSR regisztereibe írja. Az üzenet tartalma és hossza bizonyos korlátok között változhat. Informative tartalmazza saját hosszát továbbá mindazon paramétert, mely a szolgálat mikéntjét és módját meghatározza. Tartalmazza a szolgálatkérő készülék címét, továbbá az adott szolgálatkérő forrás állapotvektorának CSR címét.

A szolgálatkérés végrehajtása csak az üzenetátadás műveletének lezárultát követően kezdődhet el.

Megszakítás SR útján. Az SR vonal azoknak a készülékeknek a megszakításkérését közvetíti, amelyek a sín nem tulajdoníthatják el. Az ilyen - vezetett - készülékek mindegyikének max. 256 belső megszakításkérő forrása lehet, de ezen belül akár-mennyi, az mindegyik kell, hogy rendelkezzen a saját átállító/visszaállító, képesítő/felfüggesztő (enable/disable) bitekkel, valamint forrás és maszk állapot bitekkel. A maszkolható források VAGY kapcsolatban állnak az SR bitre vonatkozóan. Az SR bit maga is vezérelhető képesítő/felfüggesztő és átállító/visszaállító bitekkel. Az SR bit visszaállítása az összes képesített belső forrást törli. Ha valamely készüléknek csak egyetlen szolgálatkérő forrása van, akkor ez kérését közvetlenül az SR bithez irányíthatja.

A szolgálatkérés kezelő (egy e feladattal megbízott vezető) az SR vonal állítását érzékelvén, a szolgálatkérést különböző eljárások alapján kezelheti. Minden esetben azonban első lépésként folyamodnia kell a sínért, és azt el kell nyernie. Miután a sín birtokába jutott, az egyes készülékeket végigkérdezheti, vajon melyik állított szolgálatkérő jelet.

Egy másik lehetséges megoldás: közzétett paranccsal a T-kapcsok (megszakításkéréssel kongruens) állapotát az AD vonalon bit mintázat alakjában kiolvasni és a szolgálatkérő készülék(ek)et ily módon azonosítani.

Miután a szolgálatkérés kezelő meghatározza, hogy melyik készülék kér kiszolgálást, e készüléket felhívja (megcimzi), hogy megvizsgálja állapotregisztereit az SR kérés közelebbi okának megállapítása céljából. A készüléknek számos belső szolgálatkérő forrása lehet, ezért a szolgálat megkezdésére csak a belső forrás azonosítása után kerülhet sor. Az azonosítás alapelveit az alapelvek teljesíthetőségének előfeltételeit a szabvány (tervezet) rögzíti.

A szolgálatkérés kezelését végző vezető nem szükségképpen tudja elvégezni az igényelt szolgálatot, ellenben ilyen esetben megszakítási üzenetet generálhat egy a szolgálatra rendelt processzor részére. A szolgálati algoritmusok, ill. megszakítási üzenetek a megszakításkérő forrásoknak megfelelően a rendszer kezdetelése folyamán töltődnek a szolgálatkérés kezelő memóriájába. Ugyancsak a kezdetelés során történik meg a szolgálatkérő források előkészítése is az egyes források maszk bitjeinek beállításával.

Az adott forrásnak szolgálatot teljesítő (megszakított) processzor, feladata végeztével, visszaállítja az SR bitet és átállítja (ha szükséges) a maszk bitet. Közben a szolgálatkérés kezelő már hozzákezdhet egy másik forrás kérésének kezeléséhez. Minden forrás logikailag önálló entitás, mely a többi forrástól független kiszolgálást kap.

5.8 DIAGNOSZTIKAI ESZKÖZÖK

Tekintettel arra, hogy minden GYORSSIN jel a keretszegmens minden pozícióján megjelenik, a sinre csatlakoztatható egy olyan készülék is, amelynek kizárólagos feladata a GYORSSIN tevékenységének figyelése. E nyomkövető készülék, mely a GYORSSIN terminológiában a SNOOP DEVICE (fűrésző készülék) nevet kapta [112], nem csak a pillanatnyi helyzetet regisztrálhatja, hanem tárolójában egy hosszabb üzemi szakasz lefolyását is följegyezheti, későbbi kiértékelés céljából.

5.9 A FEJLESZTÉSI MUNKA SAJÁTOSSÁGAI ÉS FŐBB TERÜLETEI

A CAMAC kialakítása során még sokakat megtéveszthetett az a csalóka látszat, hogy lényegében nincs másról szó, mint a mérőrendszer moduljainak egységes csatlakoztatását lehetővé tevő sinrendszerről. Ez a látásmód jelenleg, a GYORSSIN-nel kapcsolatban, pusztán illúzióknak sem illenek be. Az egyszerű csatlakozási feladaton innen és túl a sinhez kapcsolódó - vezetett és vezetőképű tulajdonságú - készülékek közötti kommunikáció és koordináció bonyolult problematikájának egész világa huzódik meg. A ma rendszertervezője számára édeskeveset érne egy olyan új sinhálózat, mely csupán a logikai kapcsolatteremtés, az elektromos paraméterek és a mechanikai méretek egységesítésére törekednék. A GYORSSIN kidolgozása ezért sokkal szélesebb látószögű munka keretében folyik, mint ami folyt annak idején a CAMAC létrehozta. Ez nem csak erénye e tevékenységnek, hanem nehézsége is. Igen sokváltozós rendszert kell mozgatni célparaméterei irányába, miközben a célparaméterek folyékonyak, egymásra hatnak, s éppen mozgás közben szilárdulnak lassacskán végső "értékükké". A munka nagysága és a feladatok sokrétűsége nyilvánul meg a reáfordítás számos évében is, hiszen a kezdet 1977, a korszerű mérésadatgyűjtő rendszerrel szemben támasztott követelmények közzétételének [105] éve. A GYORSSIN leírás egymást követő szöveg vázlatai, noha egyrészt az elvi alapok megszilárdulását jelzik, másrészt jelentős módosulásaikkal arra figyelmeztetnek, hogy a végleges szövegre egy ideig még várni kell.

A fejlesztéssel kapcsolatos problémák kiterjedtségét és súlyát érzékeltetendő, tekintsük át azokat a területeket, amelyeken jelenleg a GYORSSIN témakörben munka folyik.

Végleges rögzítésére vár a CSR regisztertér címének és biteinek feladat hozzárendelése. E nélkül egy nagy rendszer kaotikus, kezelhetetlen volna. A rendszerszoftvernek az általános funkciókkal összefüggő információt egy és ugyanazon a helyen kell találnia, ill. ugyanarra a helyre kell elhelyeznie minden készüléknél. Az egységesítés nem zárja ki azt, hogy a

készülékek speciális funkcióihoz a CSR tér külön regisztereit és bitjeit lehessen hozzárendelni.

A megszakítási üzenet formátumát a jelen szöveg már tartalmazza. A hibajavítás mechanizmusa azonban még nincs kiérlelve. Az általános elv az, hogy a hibásnak észlelt adatot nem szabad átvinni, hanem az elvetélt adatsiklust meg kell ismételni. Am olvasás ciklusokban a forrás nem tud arról, hogy az általa kibocsátott adat hibásan érkezett a vezetőhöz. A megoldás: a vezető újra címez és újra olvassa az adatot. Ha azonban az olvasás törléssel jár együtt, ez a módszer nem alkalmazható, s nem alkalmazható azonos sorrendű (FIFO) tárolóknál sem, ahol nincs belső cím. A jelen elképzelés szerint az utolsó olvasott adatot külön regiszter őrzi, ahonnan újra olvasható. A tervezők hajlanak arra, hogy a GYORSSIN használatából kizárják azokat az elemeket, amelyek csak törléssel olvashatók, mivel az ilyen elemek használata a diagnosztizálhatóságot is gátolja.

Annak érdekében, hogy a rendszerben helyet foglaló készülékek típusát és helyét regisztrálni lehessen, hogy a készülékek részére szimbolikus neveket lehessen kiosztani, hogy általában rendelkezésre álljon minden szükséges információ a rendszer kezdetelésére, ezen belül a címkiosztáshoz és az iránytérkép generálásához is, minden esetben létre kell hozni a rendszer-adatállományt. Célszerű azonban ennek szerkezetét - formátumát - egységesíteni, ami azután másrészt lehetővé teszi rendszer-adminisztrációra szolgáló egységes segédrutinok (utility) készletének alkalmazását.

Amint a CAMAC esetében, itt is szükség van a GYORSSIN akciók szubrutinjainak kidolgozására, ill. e szubrutinoknak magasabb szintű nyelvekbe (FORTRAN, BASIC, PASCAL) való beépítésére. Ezáltal a GYORSSIN kezelése nem csak egyszerűbbé, de programozása közérthetőbbé is válik, s javulnak az egyes feladatokon dolgozó munkatársak közötti kommunikációs kapcsolat feltételei is.

Bonyolult rendszerről lévén szó, mindig szükség van valamilyen általános rendszerszoftverre is, mely a GYORSSIN kezelését megkönnyíti. Egyedi munkák helyett célszerű volna ezt a

szoftvert, ill. strukturájának általános elveit kidolgozni és egységesíteni. Ez a rendszerszoftver a fent említett rendszer-adatbázis köré épülne fel, tartalmazná az adatállomány létrehozására és fenntartására szolgáló segéd rutinokat, egyeztetné az adatbázist a meglévő szerkezeti összeállítással, előkészítené a szegmenskapcsok számára az iránytérkép elemeit, kiosztaná a címeket és prioritási szinteket, meghatározná a közzétett üzenetek utvonalait és kezdetelné a hardvert.

Kidolgozásra várnak a többprocesszoros üzemmód programozásának GYORSSIN alapelvei is.

Rögzíteni kell a soros diagnosztikai hálózat protokolljait, és a protokollokhoz tartozó hardver elemeket. Szükség van általános diagnosztikai programrendszer létrehozására. Diagnosztikai feladatokra máris alkalmazzák a "FORTH FASTBUS" nyelvet, és kidolgozták az FDL nyelvet. A diagnosztika céljára szolgáló nyomkövető (snoop) készülékkel, és a FORTH nyelv hozzá való alkalmazásával kapcsolatos munkák több helyen is folynak.

A terjedelmes GYORSSIN rendszerek tervezése során a munka bonyolultsága megköveteli az egységes leiró és kommunikációs elvek rögzítését. Erre a célra a GYORSSIN-t leiró speciális nyelv szolgál. E nyelv alapelveinek kidolgozása folyamatban van.

Nagy rendszerekről lévén szó, ugyancsak hasznos és szükséges lehet szimuláció. Szimulációt lehet használni a munkák jelen állásában a GYORSSIN variációk tulajdonságainak tanulmányozására, a későbbiekben pedig valamely megtervezett rendszer megépítés előtti vizsgálatára.

A fenti irányokban folyó munkákról [113-116] számolnak be részletesebben.

A GYORSSIN-el kapcsolatos tevékenység fenti áttekintése semmiképpen sem ad pontos képet a munkák jelen állásáról, hiszen a bármily friss publikációk megjelenése óta sem állt az óramutató.

Viszont a puszta felsorolás is fogalmat ad a tevékenység méreteiről, s rámutat azokra a feladatokra, amelyekre a GYORSSIN-t a jövőben alkalmazni kívánó fizikusnak, számítás-

technikusnak és elektronikusnak fel kell készülnie.

Az irodalomjegyzék [117-121] tételei további, a fentiekben még nem említett GYORSSIN vonatkozású közlemények címét sorolják fel.

IRODALOMJEGYZÉK

- [1] Rodrigue, G.-Giroux, D.E. - Pratt, M.: Perspectives on large-scale scientific computation. *Computer*, Oct. (1980) 65-80.
- [2] Friedman, J.H.;: Data analysis techniques for high energy particle physics. Proc. of the 1974 CERN school of computing. (1974) 271-366.
- [3] Drijard, D.: Design of experiments. Proc. of the 1978 CERN school of computing. (1978) 116-123.
- [4] Mermikides, M.E.: Data analysis for bubble chamber and hybrid systems. Proc. of the 1980 CERN school of computing (1981) 106-135.
- [5] Grote, H.: Data analysis of electronic experiments. Proc. of the 1980 CERN school of computing (1981) 136-181.
- [6] Verkerk, C.: Special purpose processors. Proc. of the 1974 CERN school of computing (1974) 223-262.
- [7] Rosner, R.A.: On line computers in high energy physics. Proc. conf. "On line computing in the laboratory" Imperial College, London. 11-12 Sept. (1975) 7-29. Advance Publ. Limited.
- [8] Foley, K.J. - Lindenbaum, S.J. - Love, W.A. - Ozaki, S. - Russel, J.J. - Yuan, L.C.L;: A counter hodoscope digital data handling and on-line computer system used in high energy scattering experiment. *Nuclear Instrum. and Methods - Vol. 30* (1964) No.1.45-60.
- [9] Lindenbaum, S.J.: The on-line computer counter and digitalized spark chamber technique. *Physics Today*, April (1965) 19-28.
- [10] Požar, F.: Computer controlled multiscaler experiment. *Nuclear Instrum. and Methods* 91 (1971) 253-265.

- [11] Dobinson, R.W.: Practical data-acquisition problems in large high-energy physics experiments. Proc. of the 1980 CERN school of computing (1981) 325-361.
- [12] Larsen, R.S.: Interlaboratory development of an integrated circuit for multiwire proportional chambers. IEEE Trans. Nucl. Sci. NS-19. No.1. (1972) 483-494.
- [13] Bell, W. - Coopex, R. - Gagliardi, F. - Heck, B. - McCulloch, L. - Sciré, M: The data distribution system for multi-experiment operation of the SFM facility. Proc. Conf. On-line Computing in the Laboratory (Advance Publication Ltd.) London, 11-12, Sept. (1975) 320-332.
- [14] Nunamaker, T.A.: Universal wire detector CAMAC scanner. Nuclear Instrum. and Methods 106 (1973) 557-561.
- [15] Basiladze, S.G. - Parfionov, A.N.: Fast readout and encoding system for real-time processors. Real-time data handling and process control. Proc. of the first European symposium, Berlin (west) 23-25 Oct. (1979) 245-248.
- [16] Yazgan, E. - Kirsten, F.: Characterization of charge-coupled analog memories for nuclear data acquisition. IEEE Trans., NS-25, No.1. (1978) 730.
- [17] Amelio, G.F.: Charge-coupled devices. Scientific American, Feb. (1974) 22-31.
- [18] Kosonocky, W.F.: Charge-coupled devices - An overview. 1974. WESCON Technical Papers, Vol. 18. Sept. (1974) 2/1, 1-20.
- [19] Kosonocky, W.F. - Sauer, D.J.: Consider CCDs for a wide range of uses. Electronic Design 6, March 15 (1976) 70-78.
- [20] EUR 1831e. ESONE system of nuclear electronics. EURATOM 1964.
- [21] TID-20893. Standard nuclear instrument modules. AEC NIM Committee. Első kiadás 1964., utolsó változat 1974.

- [22] EUR 4100e. CAMAC a modular instrumentation system for data handling. EURATOM. Első kiadás 1969., utolsó változat 1972.
- [23] LaSalle, R.A.: A modular digital interface system utilizing a bus structure. IEEE Trans. Nucl. Sci. NS-17, Nr.1.Febr. (1970) 467-475.
- [24] Kirsten, F.A.: Some thoughts on the data acquisition problem of high-energy physics experiment. IEEE Trans. Nucl. Sci. NS-17. Nr.1. Feb (1970) 452-457.
- [25] Marple, M.J. - Culvahouse, J.W.: The digital instrumentation node: an alternative to CAMAC. IEEE Trans. Nucl. Sci., NS-22, Feb (1975) 502-507.
- [26] Biri J. - Lukács J.: CAMAC perifériarendszer. Műszaki Könyvkiadó, Budapest, (1976).
- [27] Sebestyén B.: Számítógép-irányításu mérőrendszerek. Műszaki Könyvkiadó, Budapest (1976) 179-191.
- [28] Stuckenberg, H.J.: CAMAC for Newcomers. CAMAC bulletin ISSUE No. 13. Sep (1975) Supplement A
- [29] TID-26618. CAMAC tutorial articles. U.S. NIM Committee Oct (1976).
- [30] Crawthraw, M.J.: The system crate - A modular method of controlling CAMAC systems. Rutherford Laboratoy Rpt. No. RHEL/R 246 (1972).
- [31] Toy, N.V. - Drury, D.M. - Smith, K.R.E.: A modular method of multiplexing program souces to branch drivers in CAMAC systems. CAMAC bulletin, Issue No. 7. July (1973) 21-22.
- [32] Klessmann, H.: CAMAC system configurations. Proc. 2nd Internat. Symp. on CAMAC. Brussels, Oct. 14-16 (1975) 51-61.
- [33] Whitehead, N.P.: CAMAC dataway control for the PDP-8 computer family. AERE-R 6673 (1970), Harwell.

- [34] Halling, H. - Zvoll, K. - Müller, K.D.: A versatile PDP-11 CAMAC crate controller for nuclear data acquisition and processing. CAMAC bulletin No.2. (1971) 17-18.
- [35] Barthel, H. - Heep, W. - Ottens, J.G.: Specification of a CAMAC single crate controller with DMA option for PDP-11 computers. Nucl. Instrum. Meth. 138 (1976) 691-694.
- [36] Stüber, W.: Direct connection of CAMAC crate controllers type "A" to the PDP-11 unibus. CAMAC bulletin No.4. (1972) 25-26.
- [37] Reisser, P.: A universal CAMAC branch highway interface for PDP-11. CAMAC bulletin No.7. (1973) 13-14.
- [38] Whitehead, N.P.: A parallel processing coupler for CAMAC - computer systems. Harwell AERE - R 6817 (1971).
- [39] Blake, H.D. - Folwell, D.J.: An interface between CAMAC and the DIGICO MICRO-16 V computer. CAMAC bulletin No.13 (1975) 11-12.
- [40] Hellmann, G. - Ottens, J.G.: An efficient CAMAC single-crate controller for PDP-8/E. CAMAC bulletin No.7. (1973) 15-16.
- [41] Biswell, L.R.: A microprogrammed branch driver for a PDP-11 computer. CAMAC bulletin No.5. (1972) 21-23.
- [42] MBD-11 Microprogrammed branch driver for PDP-11 computers. Manual operation and specifications. Bi Ra Systems, Inc.
- [43] BORER ELECTRONICS AG. NPR controller type 1542. Ref. 603.3.053.10.73
- [44] Richards, J.M.: Harwell 7000 series CAMAC controllers. Harwell, AERE-R 6723 (1971).
- [45] Messing, G. - Stolte, J. - Kwakkel E.: A CAMAC driver-receiver. CAMAC bulletin No.12. (1975) 17-18.
- [46] Karbstein, W. - Kögel, B.: CAMAC multi-branch system for Control Data 3100 computers. CAMAC bulletin No.10. (1974) 25-26.

- [47] Iselin, F. - Lang, A. - Löfstedt, B.A. - Maurer, A. - Nicolaysen, O.P. - Orève, A. - Ponting, Ph. - Rivollet, R. - Vanuxem, J.P.: System Controller I. CERN-NP CAMAC Note 21-00, Jan (1970)
- [48] Configuring CAMAC system controllers, GEC-ELLIOTT Proc. Automation Ltd. CAMAC catalogue No. A 2951-1 (1974).
- [49] Drury, D. - Eck, C. - Vanuxem, J.P. - Meyer, J.M.: The application of a modular system controller concept to the development of advanced data acquisition and control system in CAMAC. Proc. 2nd Internat. Symp. on CAMAC, Brussels, Oct. 14-16. (1975) 83-86.
- [50] Nemes, T. - Rettelbusch, L. - Rapp, H.: Real-time application examples of the intelligent crate controller type KKI-661 at the JINR Dubna. Proc. Real-time Data Handling and Proc. Contr. First European Symposium. Berlin (West) 23-25 Oct (1979) 583-586.
- [51] Nemes, T.: An autonomous KKI-661 crate-controller. 10-12106/1979. JINR. Dubna.
- [52] CAMAC intelligent crate controller family. ICC user's manual. KFKI (1979) No. 79-254.
- [53] Gallice, P. - Mathis, M.: Autonomous crate-controller (JCAM 10) with Intel 8080 microprocessor. CAMAC bulletin No 14 (1975) 7-10.
- [54] Abbott, D.L.: Microprocessors and CAMAC-modular peripheral interfaces with distributed intelligence. COMPCON 74. Feb. 26-28. San Francisco, Calif. (1974) 89-91.
- [55] Coppo, N. - De Grandi, G. - Notagni, M. - Stanchi, L.: Microprocessor-based CAMAC data acquisition and processing system. IEEE Transactions on Nuc. Sci., Vol. NS-22, No.5., Oct (1975) 2078-2083.
- [56] Kollbach, D. - Schmidt, V.: CAMOPS-CAMAC modular processor system. CAMAC bulletin No.14. (1975) 11-12.

- [57] Schöberl, E.: CMC 8080: A CAMAC crate controller with Intel 8080 microprocessor. CAMAC bulletin No.14. (1975) 12-14.
- [58] Iselin, F. - Löfstedt, B. - Ponting, P.: Localized processing power for CAMAC systems. IEEE Transactions on Nucl. Sci., Vol. NS-22, Feb (1975) 484-487.
- [59] Halling, H.: CAMAC serial system with programmable crate controller. IEEE Transactions on Nucl. Sci., Vol. NS-21, No.1. (1974) 886-888.
- [60] Multiple Controllers in a CAMAC Crate. EUR 6500e és DOE/EV-0007, ESONE, ill. US. NIM Committee, March (1978).
- [61] Hooton, I.N.: The run time characteristics of CAMAC-computer systems. Proc. 1974. IFAC-IFIP Workshop on Real-Time Programming. Budapest, March 25-27 (1974) 135-141.
- [62] The Definition of IML, a Language for Use in CAMAC System. ESONE/IML/01, Oct. (1974); Y3AT 7: 22/TID-26615 Jan (1975).
- [63] Hooton, I.N. - Lewis, A. - Whitehead, N.P.: Implementing CAMAC-computer systems, Harwell, AERE-R 6664 (1970).
- [64] Proposal for a CAMAC Language, ESONE Committee Software Working Group. ESONE SWG 14/72, Sept (1972).
- [65] Hooton, I.N.: Standard software for CAMAC - computer system. ESONE SWG 17/72.
- [66] CAMAS Programming Language, TPA-AP-01 MA A, MTA KFKI.
- [67] Golding, F.R. - Peatfield, A.C. - Spurling, K.: CAMACRO - An aid to CAMAC interface programming. CAMAC bulletin No.5. (1972) 29-30.
- [68] Kneis, W.: Implementation of CAMAC Intermediate Language (IML) in an assembler language environment. CAMAC bulletin, No.10 (1974) 28-30.
- [69] Lewis, A.: The implementation of the CAMAC intermediate language in real-time computer systems. Proc. 1974.

IFAC-IFIP Workshop on Real-Time Programming, Budapest, March 25-27 (1974) 143-150.

- [70] Kubitz, M. - Kind, R.: Macro-IML implementations for the PDP-11 computer. CAMAC bulletin, No.13. (1975) 17-18.
- [71] Kubitz, M. - Kind, R.: Macro-IML implementations for the PDP-11 computer. Proc. 2nd Internat. Symp. on CAMAC. Brussels, Oct (1975) 103-107.
- [72] May, F. - Marschik, W. - Halling H.: A FOCAL interrupt handler for CAMAC. CAMAC bulletin, No.6. (1973) 21-22.
- [73] Haase, V.: CAMAC in PEARL - Comments in standardization in laboratory automation and process control. First Internat. Symp. on CAMAC in Real-Time Computer Applications. Luxemburg, Dec. (1973).
- [74] Thomas, R.F. Jr.: Some aspects of CAMAC software. IEEE Trans. on Nuclear Science. Vol. NS-20/2 (1973) 50-68.
- [75] Thomas, R.F. Jr.: Specifications for standard CAMAC sub-routines. CAMAC bulletin No.6. (1973) 23-26.
- [76] Michelson, J. - Halling, H.: Procedure calls - a pragmatic approach. First Internat. Symp. on CAMAC in Real-Time Computer Applications. Luxemburg, Dec (1973) 63-66.
- [77] Hagan, P-J.: A view of IML. SWG 25/73.
- [78] Kneis, W. - Karbstein W.: CAMAC with FORTRAN on a CDC 3100, an approach based on IML. CAMAC bulletin, No.13. (1975) 18-20.
- [79] Lőcs Gy. - Sarkadi Nagy I. - Szlankó J.: A BASIC programozási nyelv.
- [80] Peterson, G.L.: BASIC the language of time sharing. Hewlett-Packard Journal, Nov. (1968) 2-8.
- [81] Moley, R.M.: BASIC at Hewlett-Packard. Hewlett-Packard Journal, Nov (1968) 9-13.
- [82] Gudniz, L. - Tsuda, H.: General-purpose test system gets digital capability. Hewlett-Packard Journal, July (1971) 2-9.

- [83] A pocket guide to the 2100 computers. Hewlett-Packard Co. 9/72, 5951-4423.
- [84] Visschersr, J.L.-ten Hertog, A.: CAMAC extension for BASIC 2/3 on the ALPHA-LSI computer. CAMAC bulletin, No. 12. (1975) 27-28.
- [85] Real-Time BASIC for CAMAC. TID-26619, Feb (1977) ESONE/RTB/02.
- [86] Bianchi, G. - Kubitz, M. - Lewis, A.: An implementation of real-time BASIC for CAMAC on a PDP-11. Proc. 2nd Internat. Symp. on CAMAC. Oct. (1975) 91-95.
- [87] "CATY A CAMAC testing aid". GEC-Elliott Process Automation Limited Product Specification No. C1030 (1975)
- [88] "Language Definition of CATY-1", UK CAMAC Association (1977).
- [89] Peatfield, A.C. - Spurling, K. - Zacharov, B.: CAMAC as a computer peripheral interface system. IEEE Trans. on Nucl. Sci. NS-21. Nr.1 (1974) 867-869.
- [90] SEN ELECTRONIQUE CAMAC NOTES AND NEWS. The NIM-CAMAC marriage. Apr. (1970).
- [91] Az eredeti szabványleírásokon túl 1. még PHILIPS. Digital instrument course, Part 4. IEC Bus Interface.
- [92] Brenner, A.E. - Martin, R.G.: CAMAC extended branch serial driver. IEEE Transactions on Nuclear Science, Vol. NS-22. Feb (1975) 511-516.
- [93] Kwakkel, E. - Messing, G.: A CAMAC serial branch adapter. CAMAC bulletin, No.9. (1974) 15-16.
- [94] Klessmann, H. - Pangrizt, H. - Wawer, W.: A standard part for communication with CAMAC peripherals. CAMAC bulletin, No.3. (1972) 15-16.
- [95] McPherson, G.: A standard interface port for CAMAC. Proc. 2nd Internat. Symp. on CAMAC. Brussels, Oct (1975) 127-131.

- [96] Heep, W. - Ottes, J.G. - Tradowsky, K.: Design characteristics for CAMAC modules. CAMAC bulletin, No.2. (1971) 15-16.
- [97] Kirsten, F.A.: Operational characteristics of the CAMAC dataway. CAMAC tutorial Articles. TID-26618, Oct (1976) 15-26
- [98] Drury, D.: The testing of CAMAC modules - A description of current industrial practice. Proc. 2nd Internat. Symp. on CAMAC, Brussels, Oct (1975) 223-228.
- [99] Dawson, W.K. - Gjovig, A. - Naivar, F. - Potter, J. - Smith, W.: CAMAC system test module. IEEE Transactions on Nuclear Science, Vol. NS-28, No.1., Feb (1981) 375.
- [100] Van Dellen, K. - Sporrel, F. - Taff, L.M; A CAMAC buffered readout module for a multiplexed ADC. Nuclear Instrum. and Methods, 125 (1975) 525-530.
- [101] Buchanan, J.A.: CAMAC MWPC readout system with event selection logic. IEEE Transactions on Nucl. Sci., Vol. NS-22, Feb (1975) 543-546.
- [102] Nelson, D.J. - Breidenbach, M. - Granieri, C.D. - Grund, J.E. - Patrick, J. F. - Weaver, L.J.: The VAX CAMAC channel. IEEE Transactions on Nucl. Sci., Vol. NS-28, No. 1. Feb (1981) 336-340.
- [103] Lindsay, J.B. - Millerin, C. - Tarlé, J. C. - Verweij, H. - Wandler, H.: A fast and felxible data acquisition system for multiwire proportional chambers and other detectors. Nuclear Instrum. and Methods, 156 (1978) 329-333.
- [104] Ponting, P. J.: A guide to ROMULUS/REMUS data acquisition systems, CERN EP-Electronics Note 80-01. Feb (1980).
- [105] Future data bus requirements for laboratory high speed data acquisition systems, U.S. NIM Committee USERDA TID 26621, June (1977).

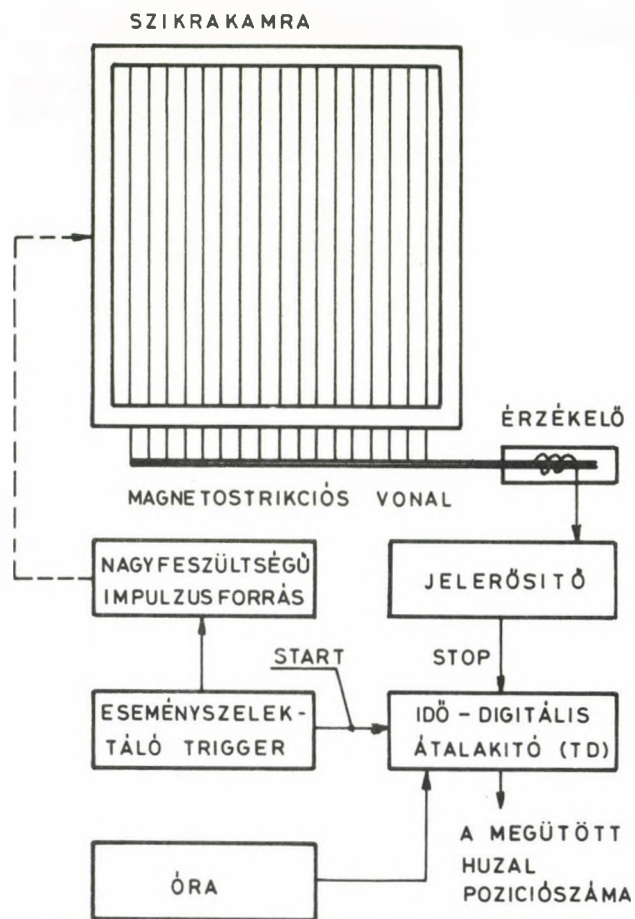
- [106] Larsen, R.S.: "FASTBUS" - Status of development of a standard high speed data acquisition bus for high energy physics. IEEE Transactions on Nuclear Sci., Vol. NS-25, No.1, Feb (1978) 735-739.
- [107] Larsen, R.S.: Status of the FASTBUS standard data bus. IEEE Transactions on Nuclear, Sci., Vol. NS-28, No.1. Feb (1981) 322-329.
- [108] Paffrath, L.: The FASTBUS standard - A review of the current status. IEEE Transactions on Nuclear Sci. Vol. NS-28, No.5, Oct (1981) 3786-3788.
- [109] Logg, C.A. - Paffrath, L. - Bertolucci, B. - Horelick, D: FASTBUS introduction and demonstration, IEEE Transactions on Nuclear Sci., Vol. NS-28, No.5. Oct(1981) 3801-3806
- [110] Gustavson, D.B.: FASTBUS status from a system designer's point of view. IEEE Transactions on Nuclear Sci., Vol. NS-28, No.5. Oct (1981) 3796-3800.
- [111] FASTBUS Modular High Speed Data Acquisition System for High Energy Physics and other Applications. Working Group Document - Tentative Specification U.S. NIM Committee, 20. Jan (1982)
- [112] Walz, H.V.: FASTBUS SNOOP diagnostic module. IEEE Transactions on Nuclear Sci., Vol. NS-28, No.1. Feb (1981) 380-384.
- [113] Gustavson, D.B.: FASTBUS software status. IEEE Transactions on Nuclear Sci., Vol. NS-28, No.1. Feb (1981) 330-332.
- [114] Nater, K.D.: FDL - A high level diagnostic language for FASTBUS. IEEE Transaction on Nuclear Sci., Vol. NS-28. No.5. Oct (1981) 3807-3809.
- [115] Pordes, R. - Gannon, S. - Taf, L.: Standard software routines for FASTBUS. IEEE Transactions on Nuclear Sci., Vol. NS-28. No.5. Oct (1981) 3810-3811.

- [116] Christopher, T. - El-Dessouki, O. - Evens, M. - Kabat, W. - Wagle, S.: A FASTBUS description language. Proc. Distributed Computing COMPCON FALL 80. Sept. 23-25 (1980) 543-550.
- [117] Dowing, R.W.: FASTBUS slaves - A designers view. IEEE Transactions on Nuclear Sci., Vol. NS-28, No.5. Oct (1981) 3789-3795.
- [118] Leipuner, L.B. - Fuhrmann, J. - Larsen, R. - Makowiecki, D. - Morse, W. - Rudolf, T. - Sims, W. - Blatt, S. - Campbell, M. - Kasha, H. - Schmidt, M.: A FASTBUS system used in a high energy experiment. IEEE Transactions on Nuclear Sci., Vol. NS-28. No.1, Feb (1981) 333-335.
- [119] Gustavson, D.B. - Holmes, T.L. - Paffrath, L. - Steffani, J.P.: A "Front Panel" human interface for FASTBUS. IEEE Transactions on Nuclear Sci., Vol. NS-28, No.1., Feb (1981) 343-345.
- [120] Campbell, M. - Blatt, S. - Kasha, H. - Schmidt, M.: High speed processor for FASTBUS. IEEE Transactions on Nuclear Sci., Vol. NS-28. No.1., Feb (1981) 369-371.
- [121] Larwill, M. - Barsotti, E. - Lagerlund, T.D. - Taff, L.M. - Franzen, J.: A UNIBUS processor interface for a FASTBUS data acquisition system. IEEE Transactions on Nuclear Sci., Vol. NS-28. No.1. Feb (1981) 385-389.
- [A1] Нетраков, А.В.: Развитие телевизионного бесфильмового метода измерения координат быстропротекающих процессов /обзор/. ПТЭ, 6 /1980/ 5-17.
- [A2] Басиладзе, С.Г. - Парфенов, А.В.: Универсальный быстродействующий цифровой шифратор для процессорных систем отбора событий. ПТЭ, 1 /1980/ 88-92.
- [A3] Басиладзе, С.Г.: Электронная регистрирующая аппаратура в стандарте КАМАК. ПТЭ, № 2 /1981/ 7-21.
- [A4] Аблеев, В.Г. и др. /24 автора/: Одноплечевой магнитный спектрометр установки "АЛЬФА". Описание и характери-

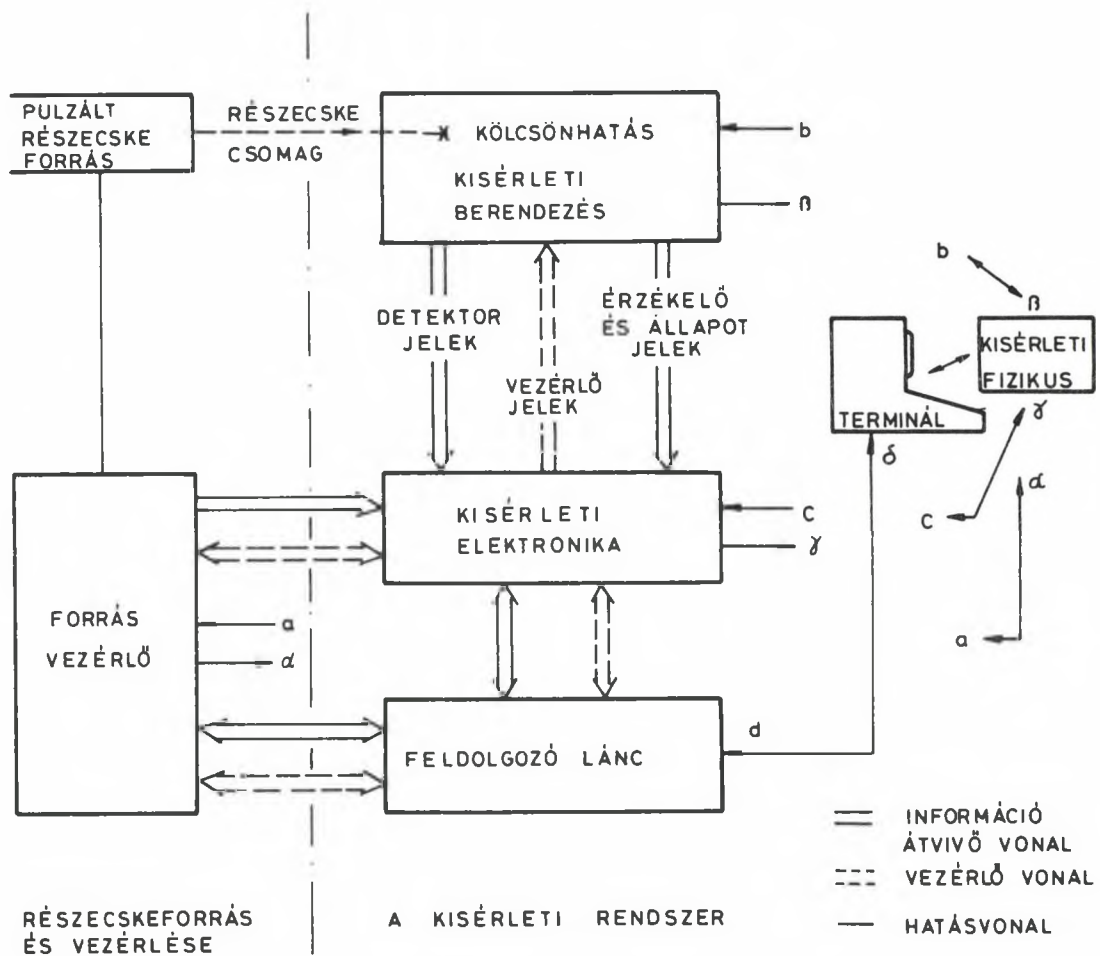
ки спектрометра. Организация работы. ПТЭ. № 2 /1978/
63-70.

- [A5] Нгуен, Фук: Автореф. дисс. 10-8990 /1975/ ОИЯИ Дубна.
- [A6] Базиладзе, С.Г. - Юдин, В.К.: Система регистрации информации с пропорциональных камер. ПТЭ, № 5 /1981/ 68-75.
- [A7] Базиладзе, С.Г. - Лохоняи, Л.: Система регистрации информации с дрейфовых камер. ПТЭ, № 5 /1981/ 62-68.

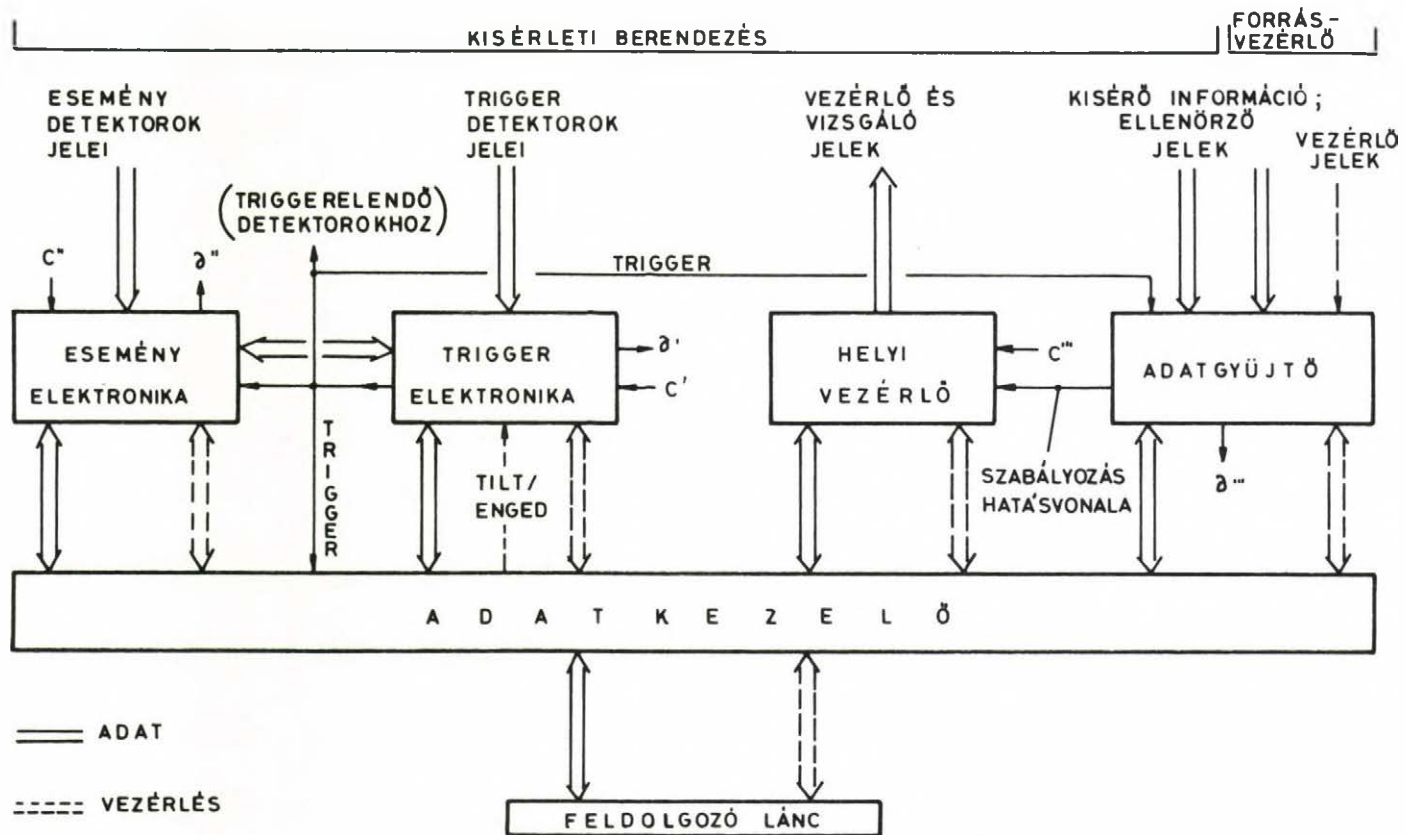
Á B R Á K



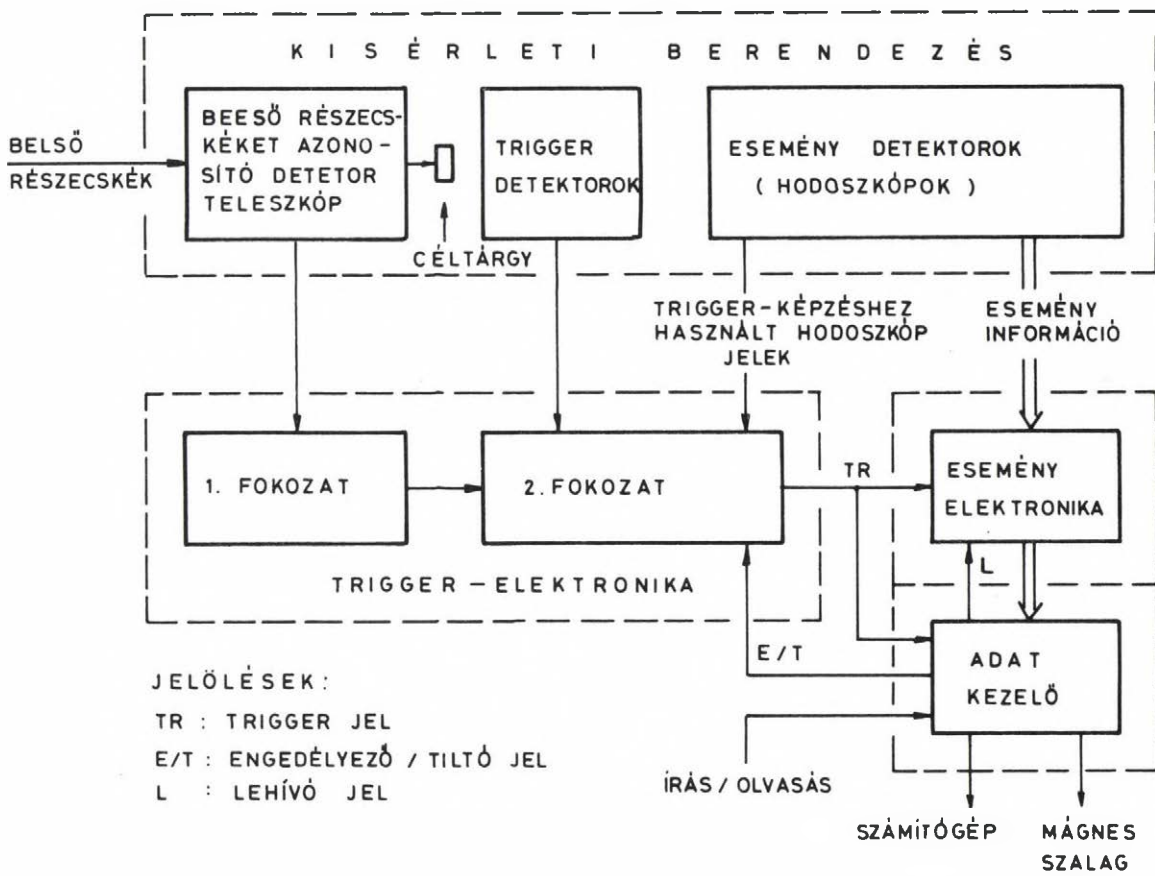
1. ábra.



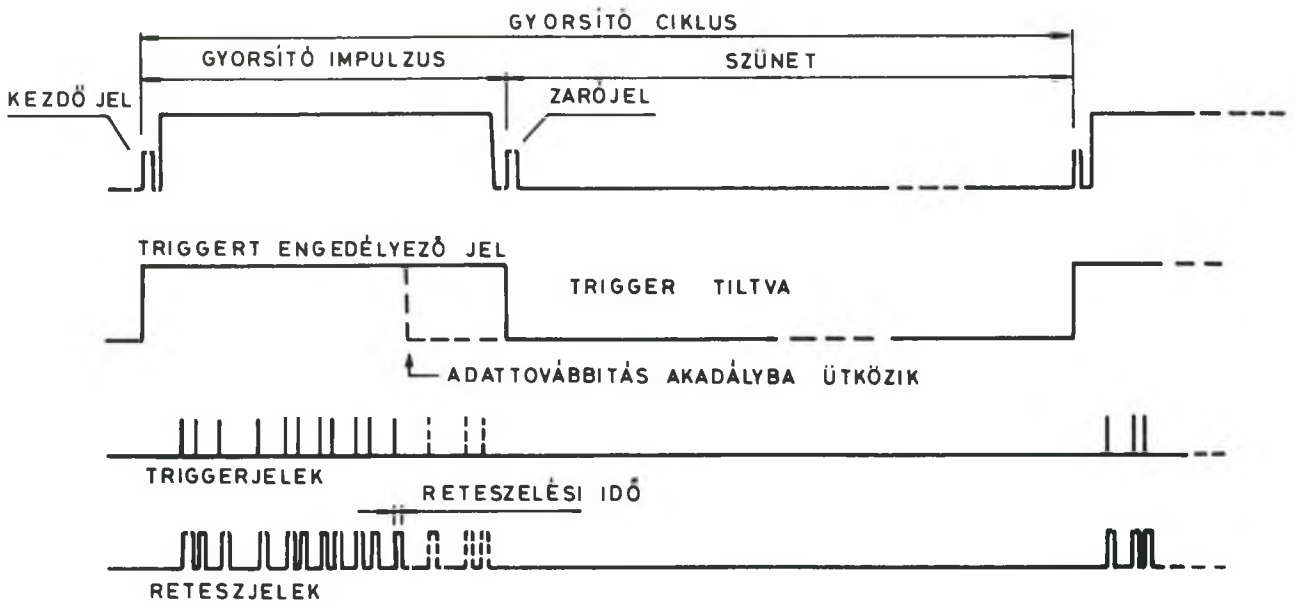
2. ábra.



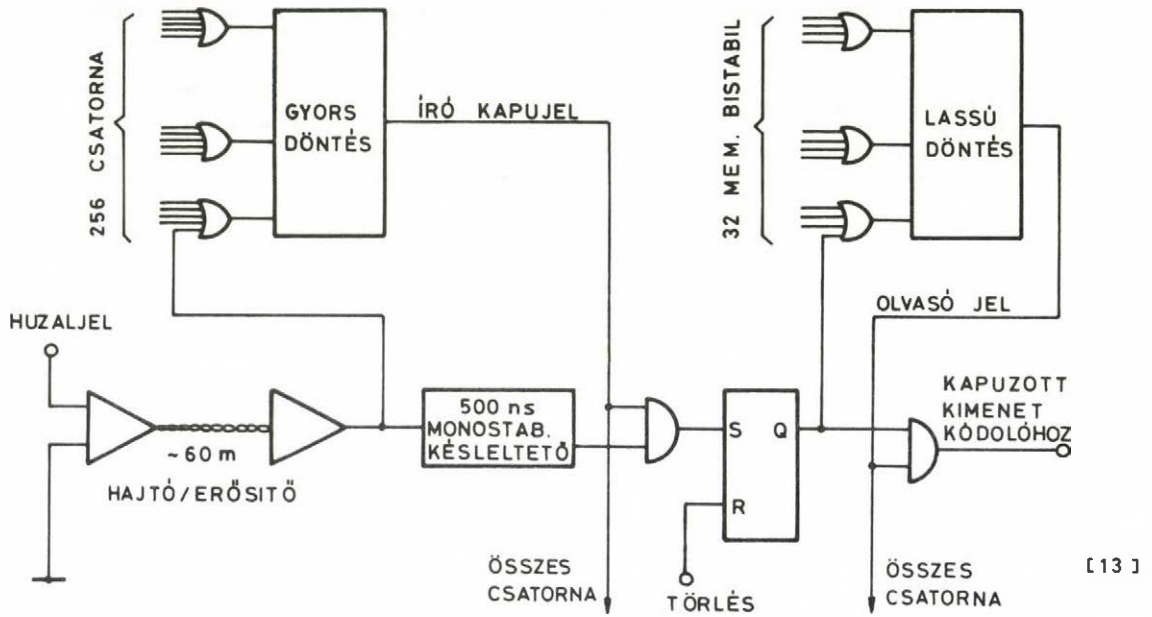
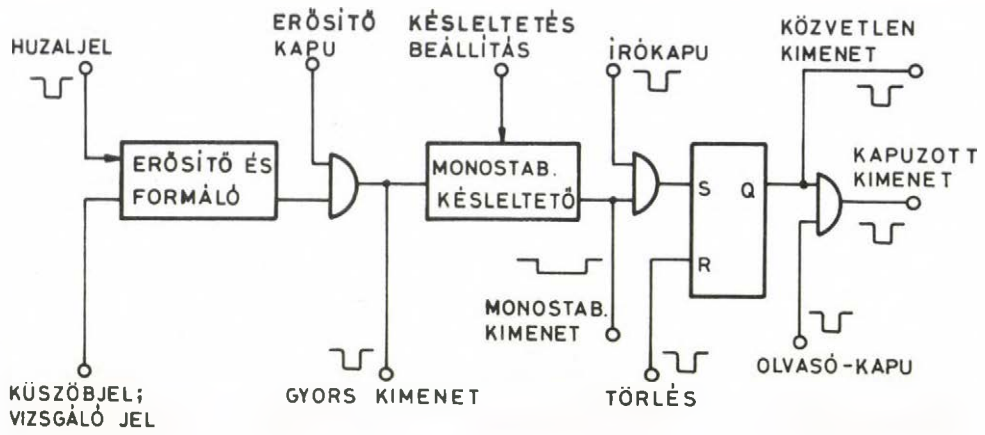
3. ábra.



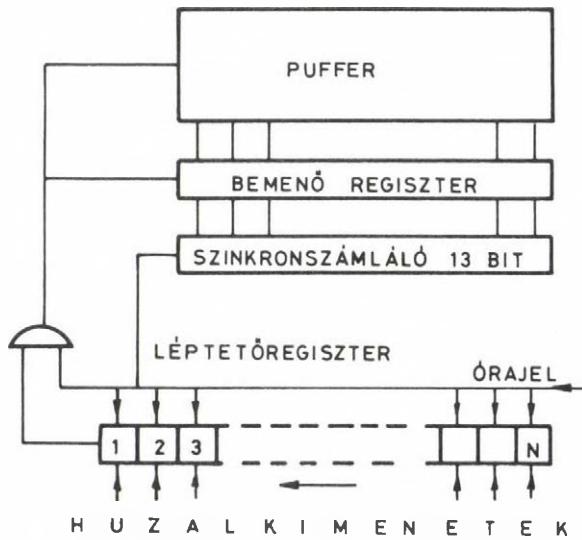
4. ábra.



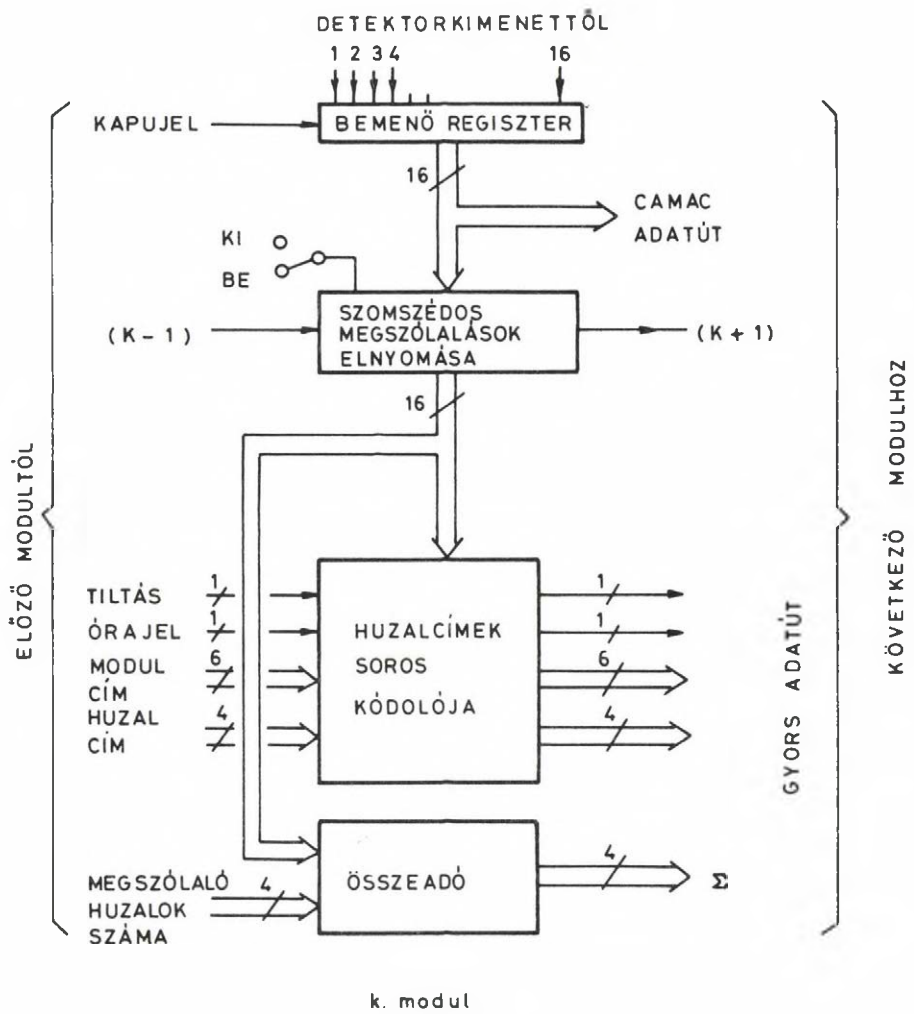
5. ábra.



6. ábra.

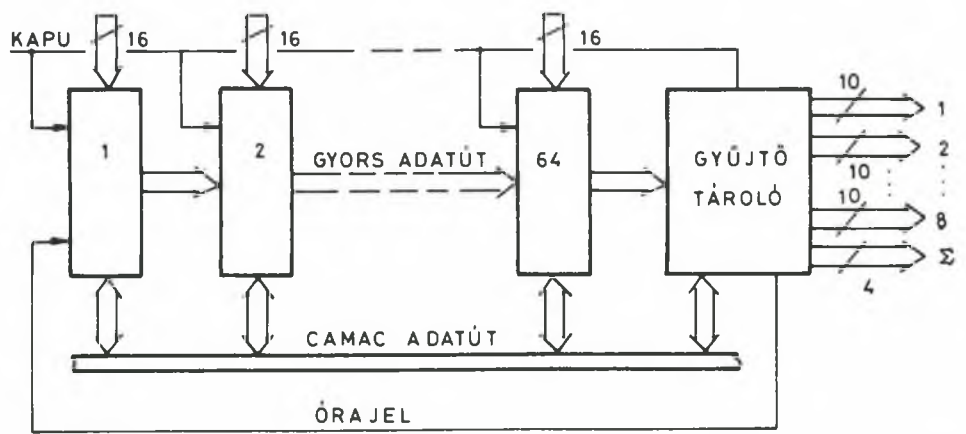


7. ábra.



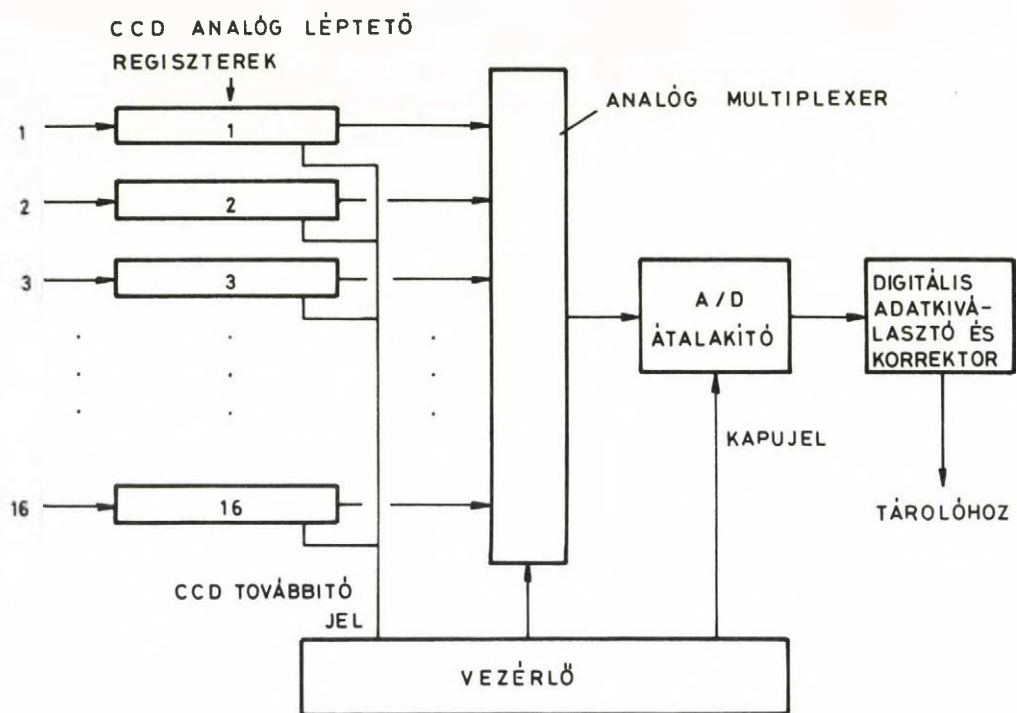
k. modul

a.)

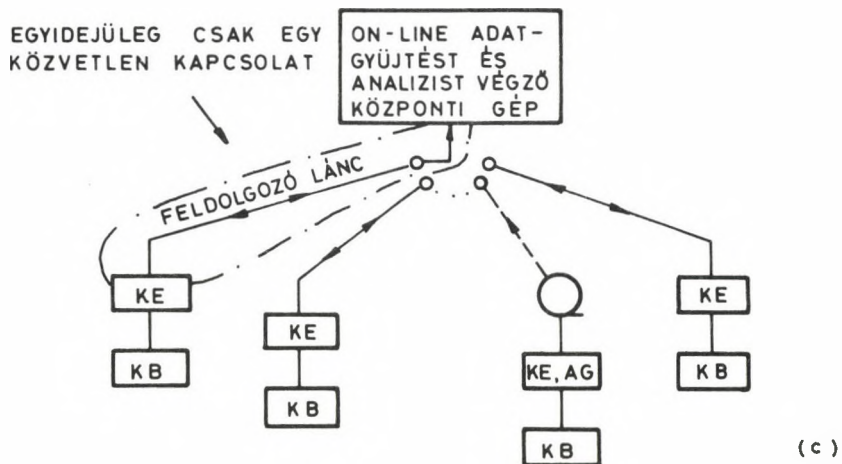
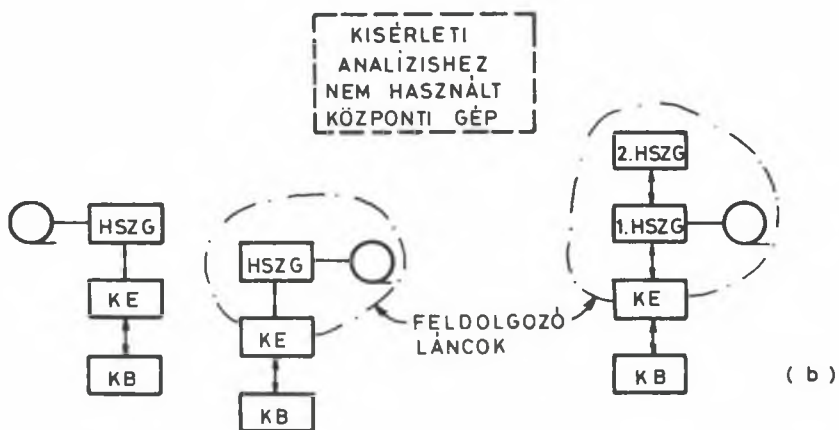
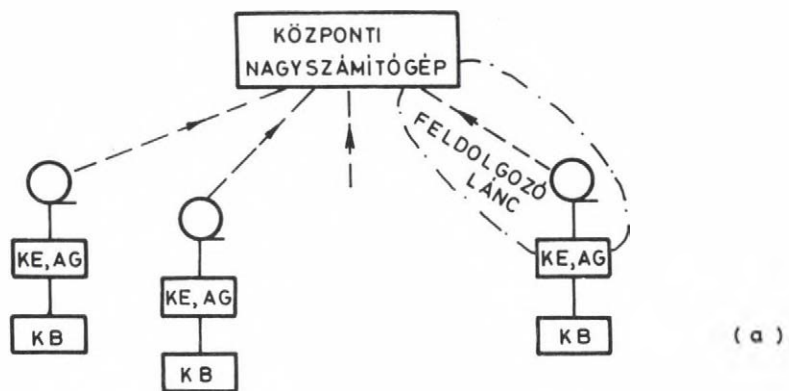


b.)

8. ábra.

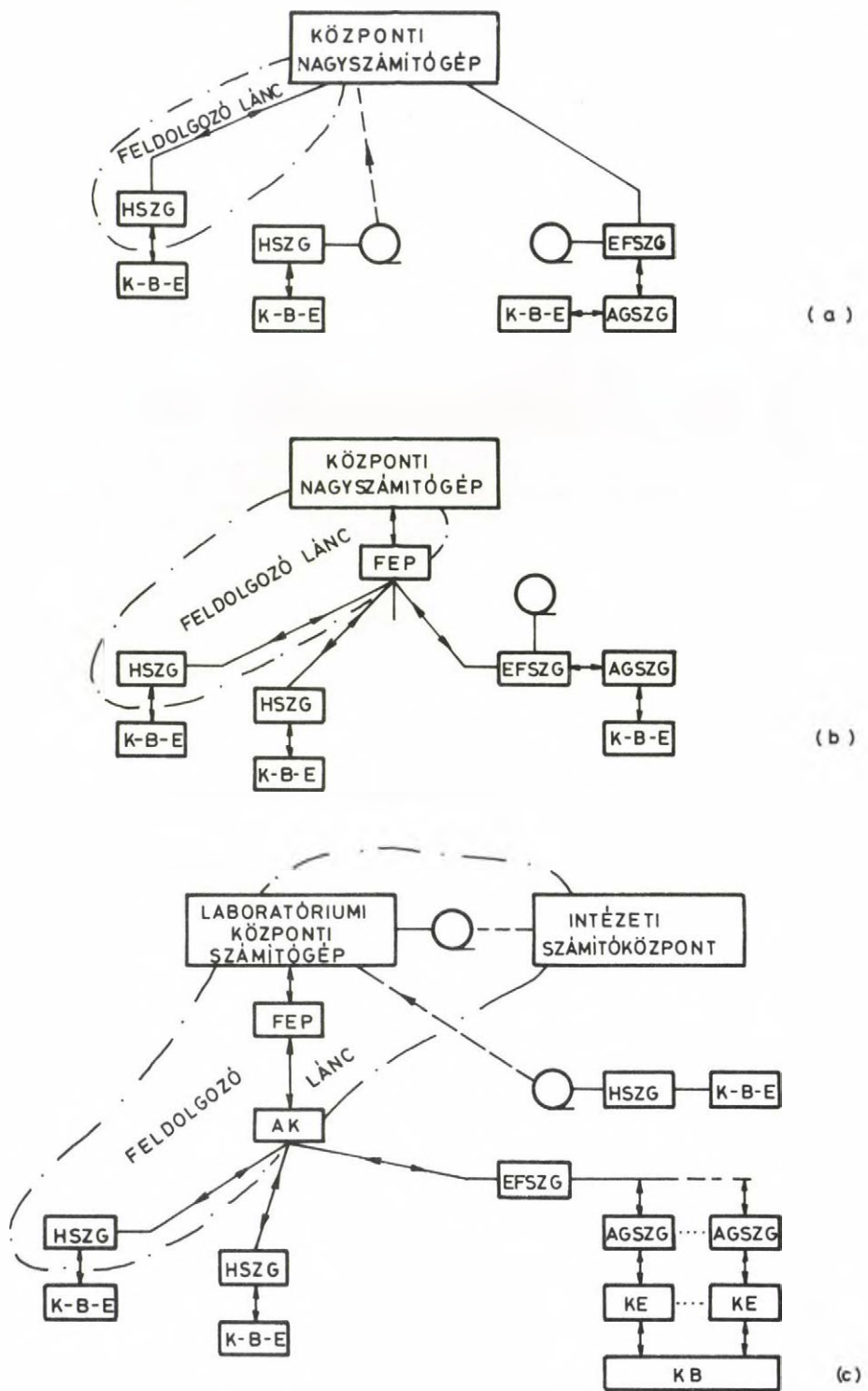


9. ábra.



JELÖLÉSEK : KB - KISÉRLETI BERENDEZÉS ; HSZG - HELYI SZÁMÍTÓGÉP
 KE - KISÉRLETI ELEKTRONIKA ;
 AG - ADATGYŰJTŐ ;
 ----- KÖZVETETT KAPCSOLAT ;
 ——— KÖZVETLEN KAPCSOLAT ;

10. ábra.



JELÖLÉSEK :

K - B - E - KÍSÉRLETI BERENDEZÉS + ELEKTRONIKA

KB - KÍSÉRLETI BERENDEZÉS

KE - KÍSÉRLETI ELEKTRONIKA

HSZG - HELYI SZÁMÍTÓGÉP

FEP - FRONT - END PROCESSOR

— KÖZVETLEN KAPCSOLAT

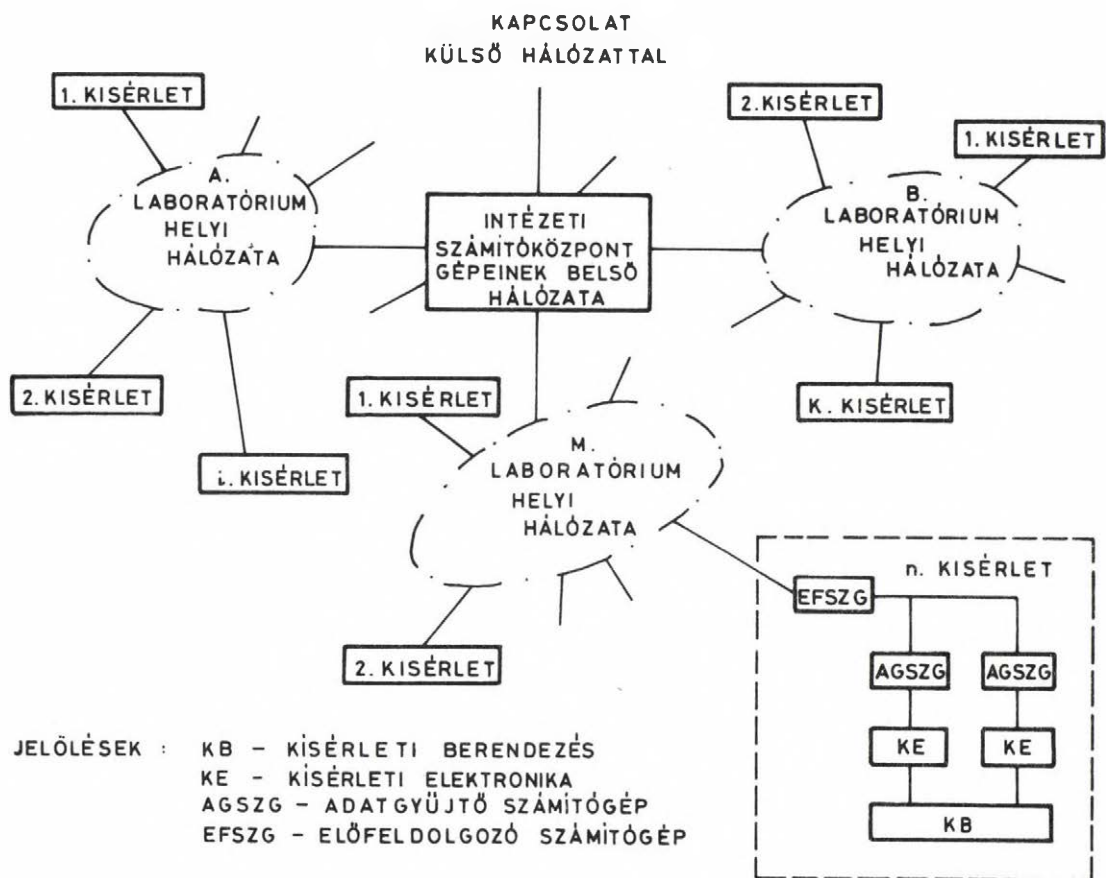
AGSZG - ADATGYŰJTŐ SZGÉP

EFSZG - ELŐFELDOLGOZÓ SZGÉP

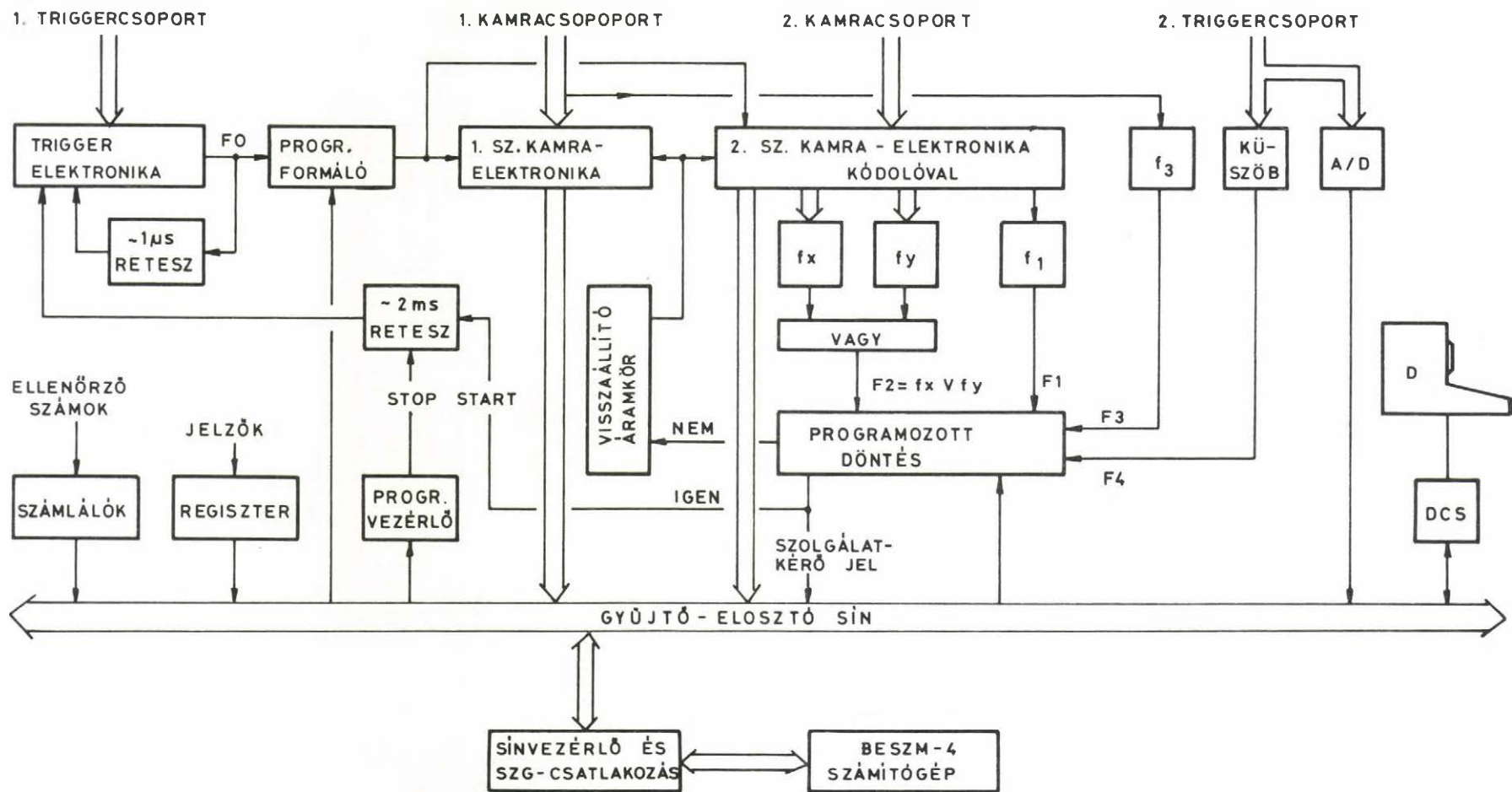
AK - ADATKONCENTRÁTOR

----- KÖZVETETT KAPCSOLAT

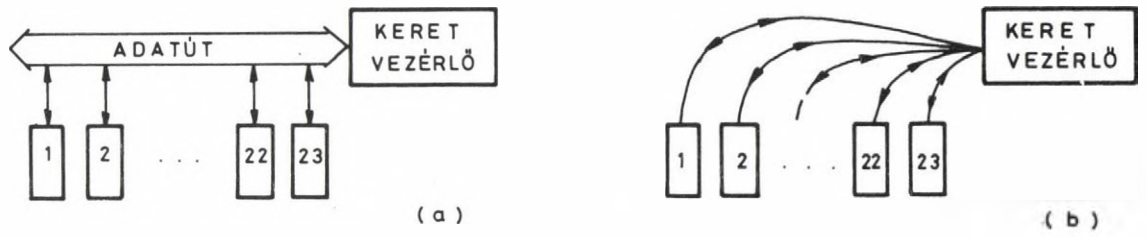
11. ábra.



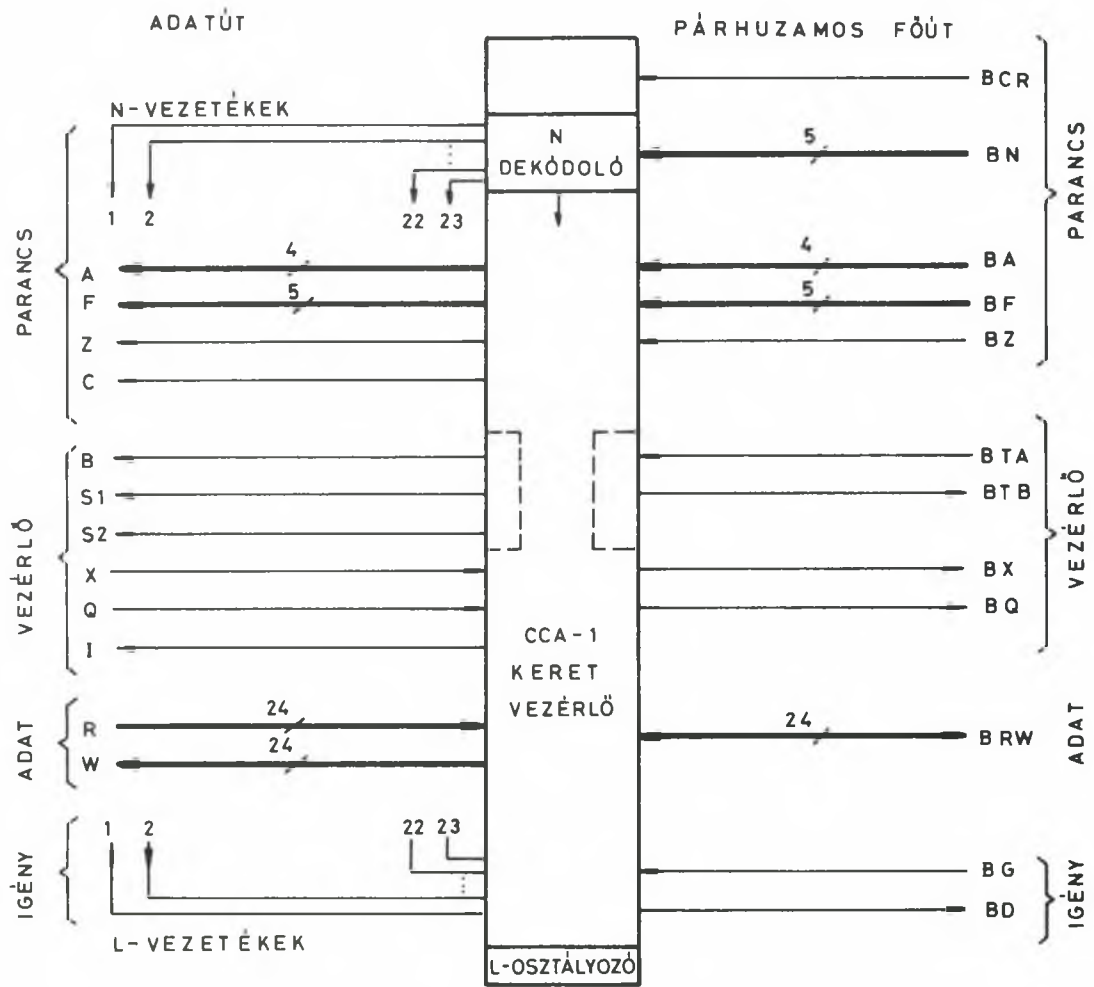
12. ábra.



13. ábra.



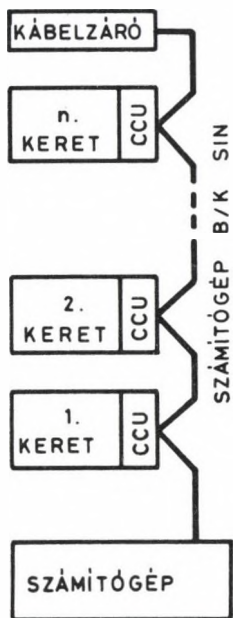
14. ábra.



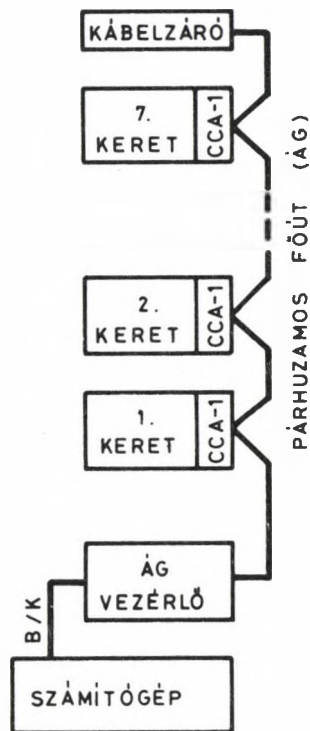
15. ábra.

ADATÚT MŰVELETEK	F FELADATKÓD	
	1. RCS	2. RCS
OLVASÁS	00000	00001
OLVASÁS TÖRLÉssel	00010	—
KOMPLEMENTIS OLVASÁS	00011	—
ÍRÁS	10000	10001
SZELEKTÍV ÍRÁS	10010	10011
SZELEKTÍV TÖRLÉS	10101	10111
TÖRLÉS	01001	01011
IGÉNYVIZSGÁLAT	01000	
IGÉNYTÖRLÉS	01010	
ÁLLAPOTVIZSGÁLAT	11011	
FELFÜGGESZTÉS	11000	
ENGEDÉLYEZÉS	11010	
VÉGREHAJTÁS	11001	
F JÖVŐ FEJLESZTÉS CÉLJÁRA FENNTARTOTT ÉRTÉKEI : 00101 ; 00111 ; 01101 ; 01111 ; 11101 ; 11111		
NEM STANDARD F ÉRTÉKEK : 00100 ; 00110 ; 01100 ; 01110 ; 10100 ; 10110 ; 11100 ; 11110		
1. RCS ; 2. RCS — KÉT KÜLÖNBÖZŐ REGISZTERCSOPORT		

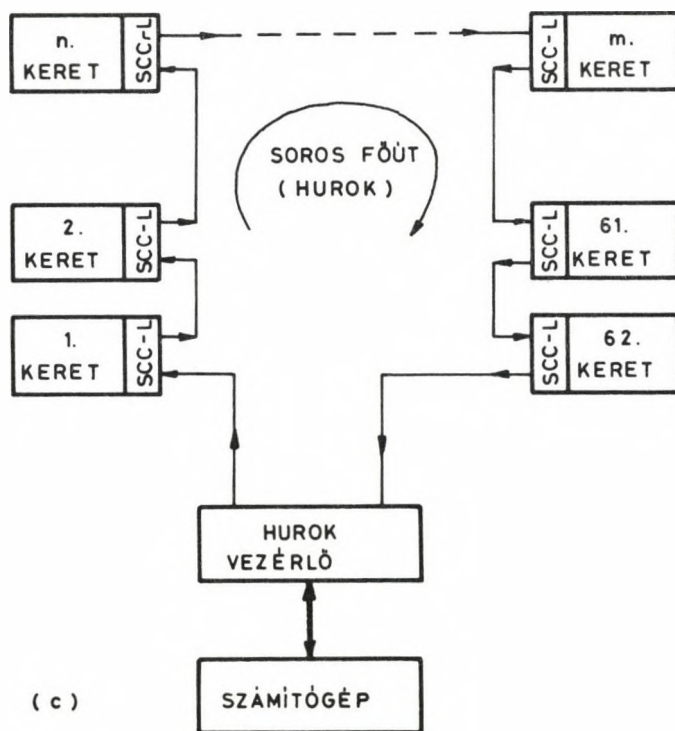
16. ábra.



(a)

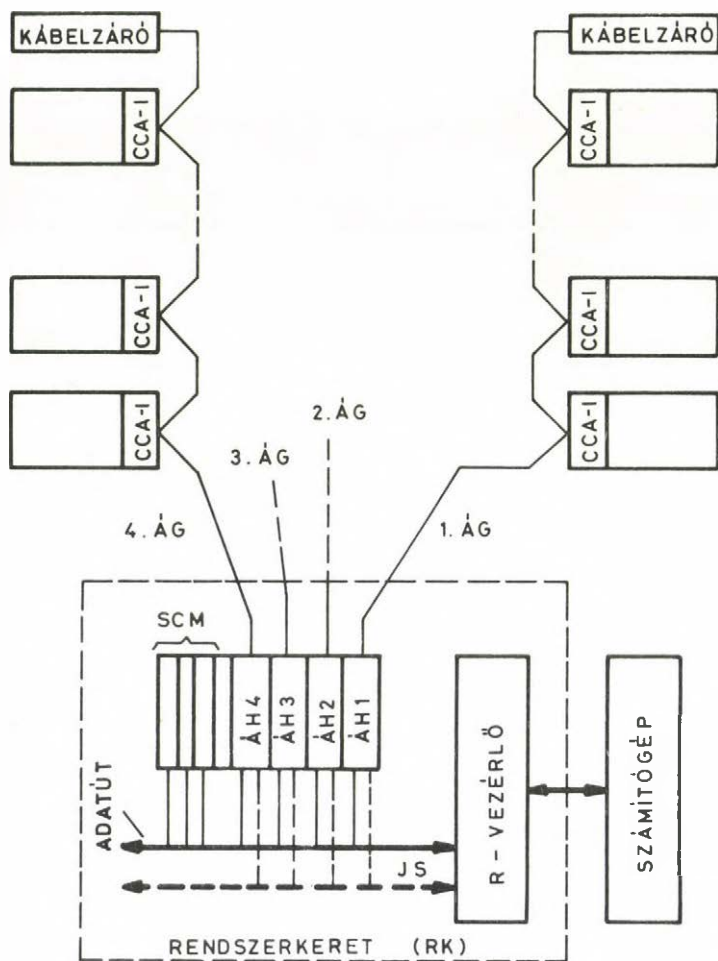


(b)



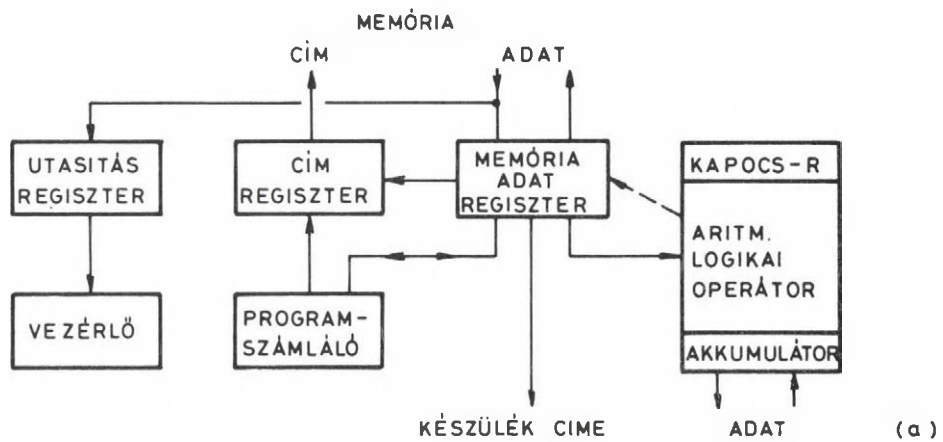
(c)

17. ábra.

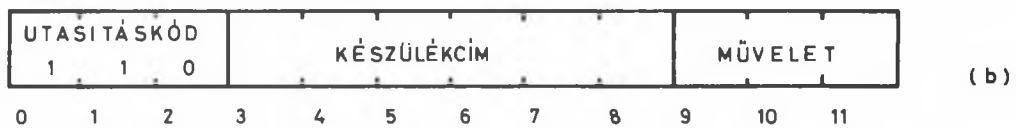


JS - JÁRULÉKOS SÍN ; ÁH - ÁGHAJTÓ
 SCM - STANDARD CAMAC MODULOK

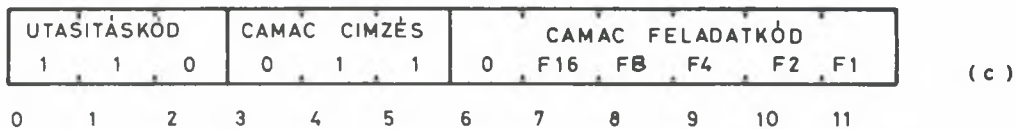
18. ábra.



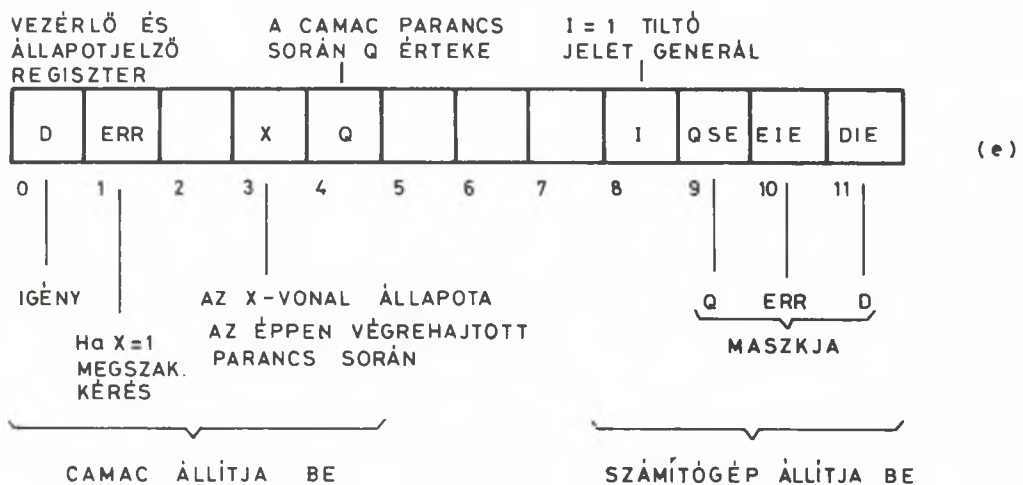
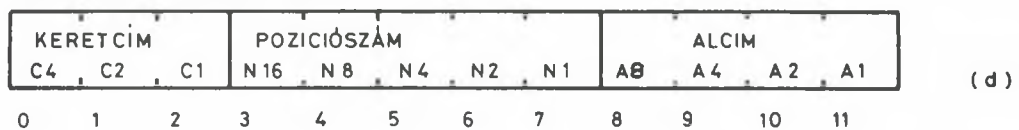
1001 TPA-l B/K UTASÍTÁS



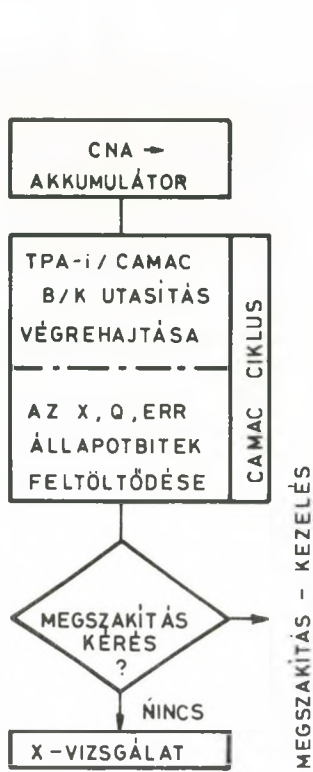
1001 TPA-i / CAMAC B/K UTASÍTÁS



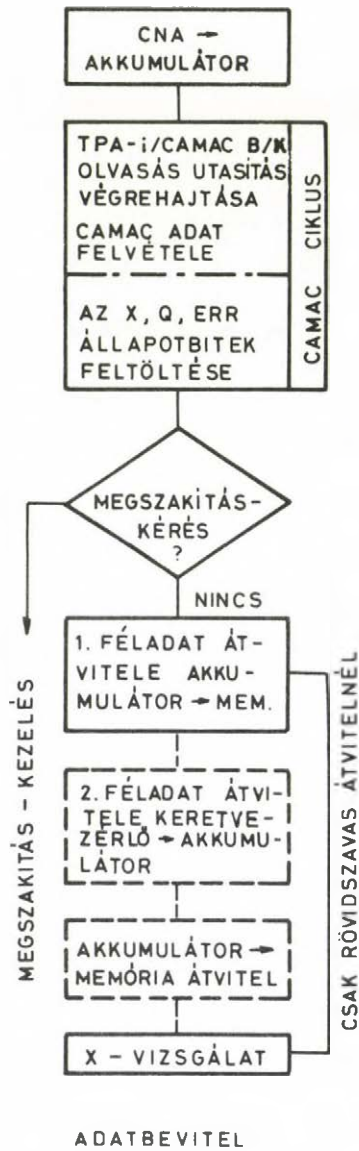
1001 TPA-l CAMAC CÍM



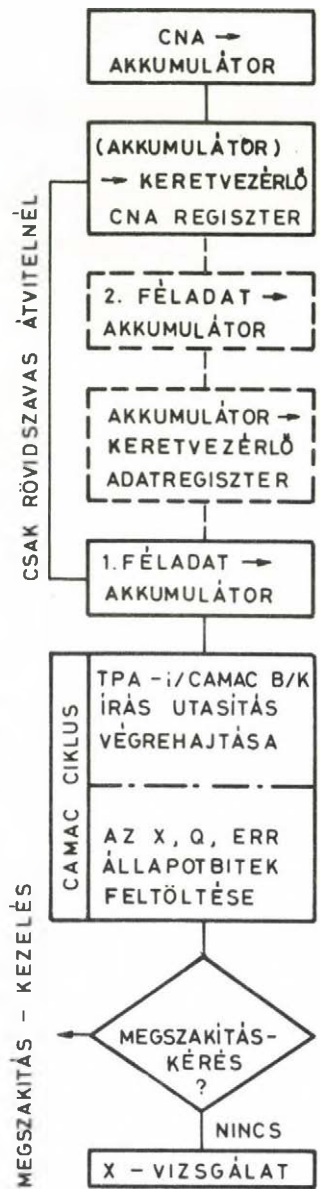
19. ábra.



VEZÉRLÉS JELLEGŰ
CAMAC PARANCSONK

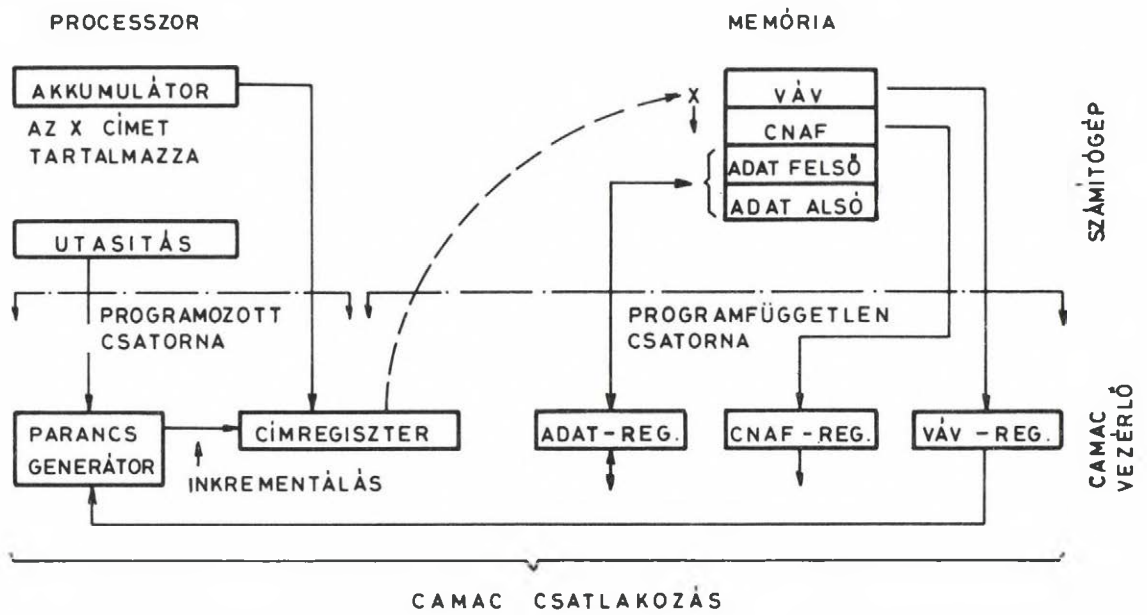


ADATBEVITEL

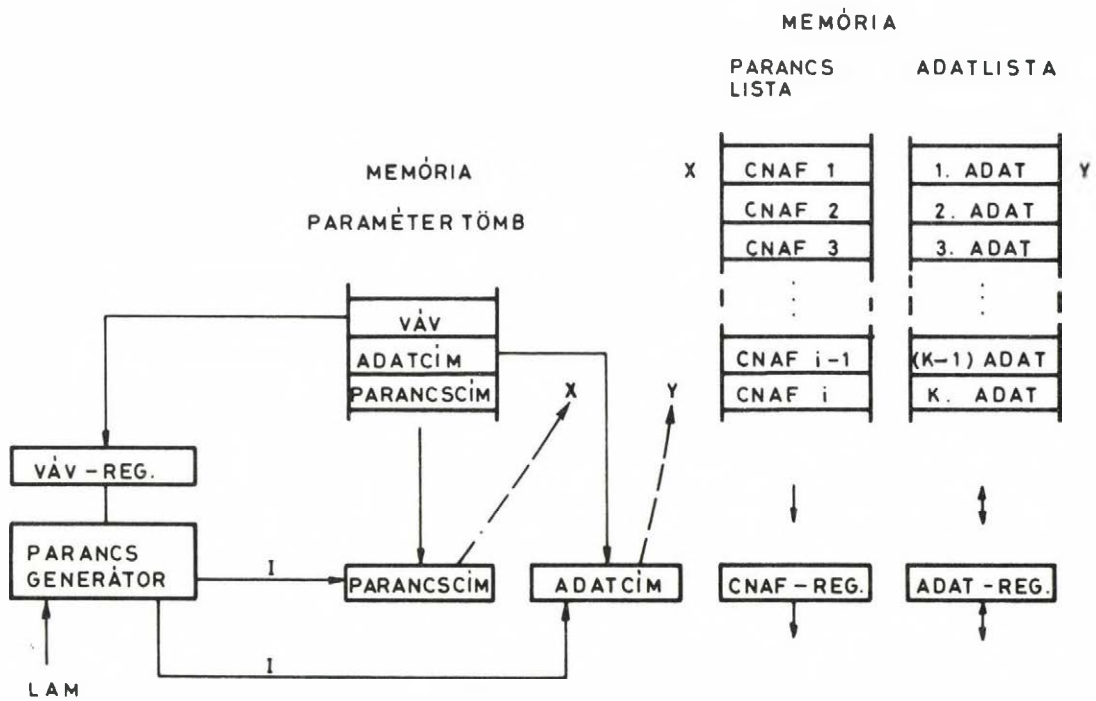


ADATKIVITEL

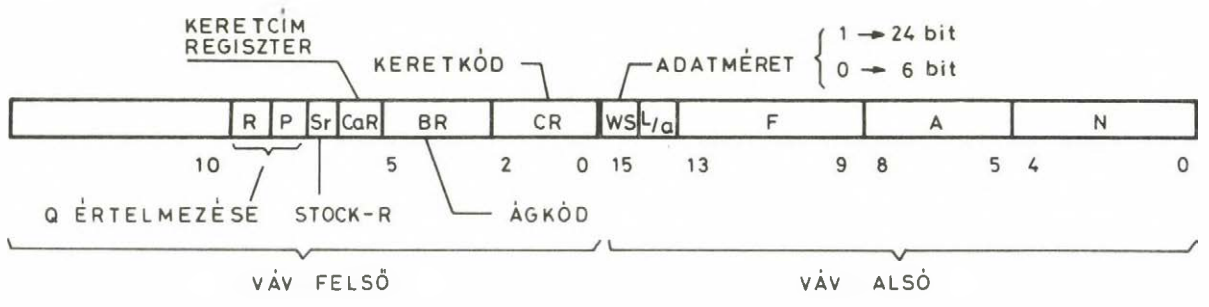
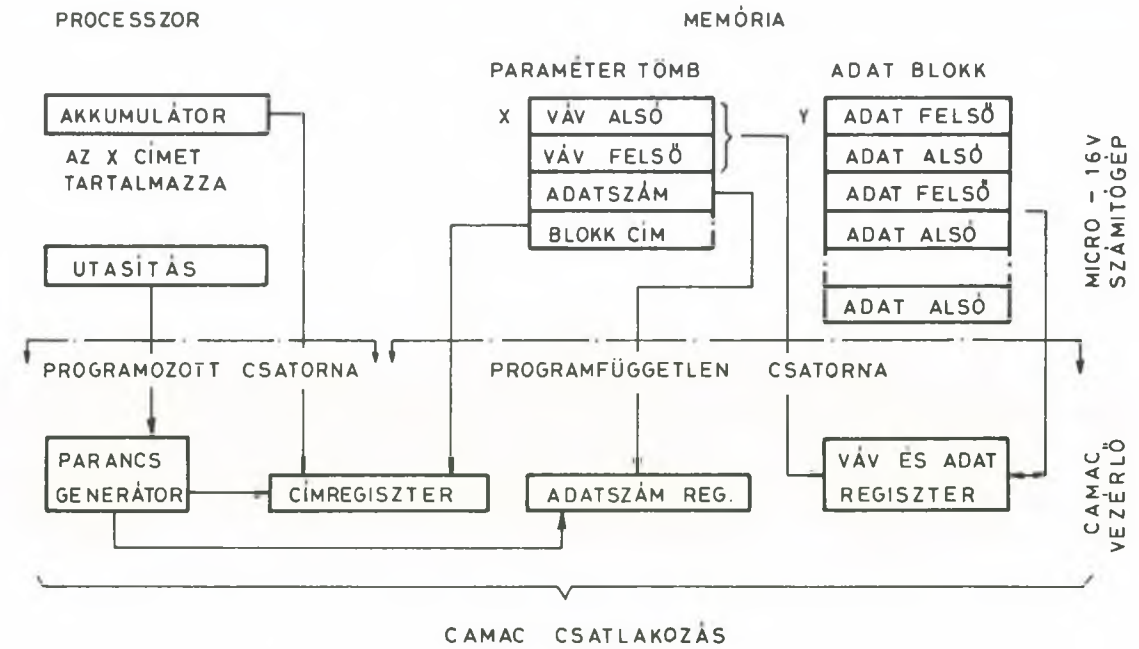
20. ábra.



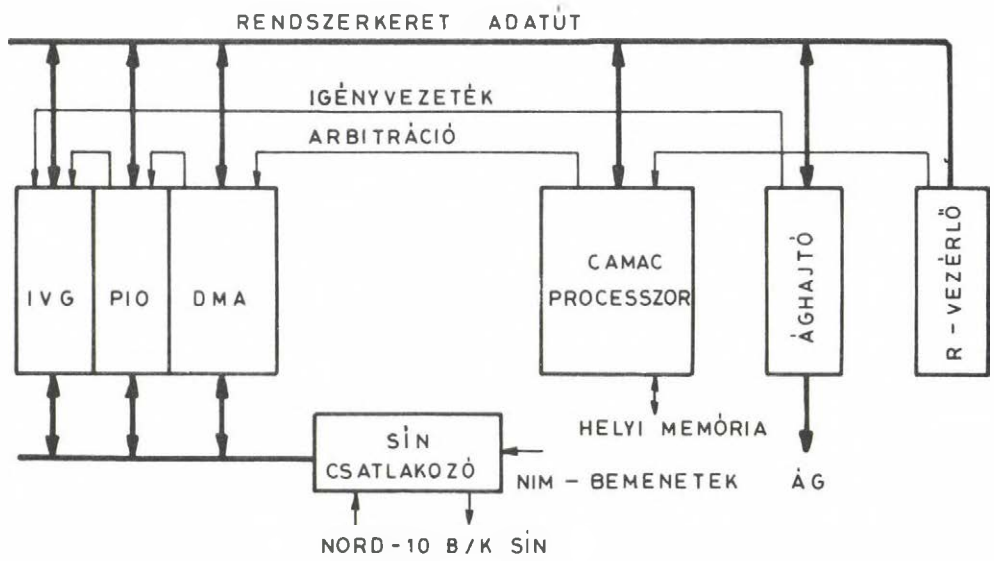
21. ábra.



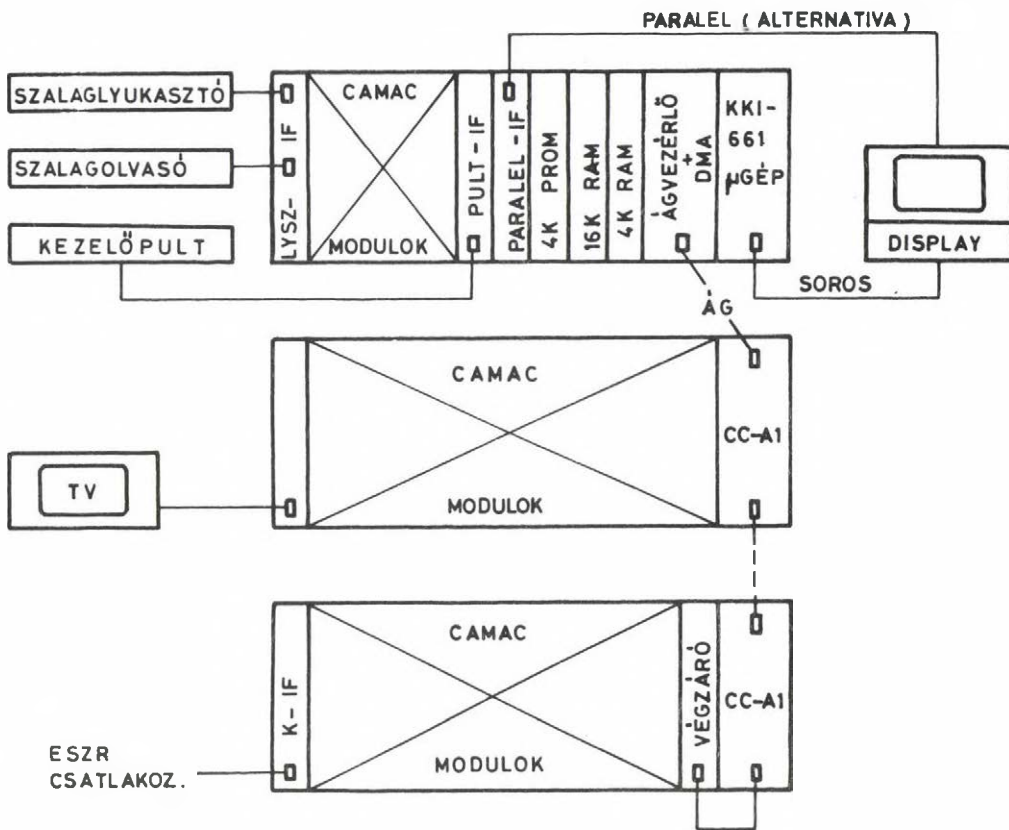
22. ábra.



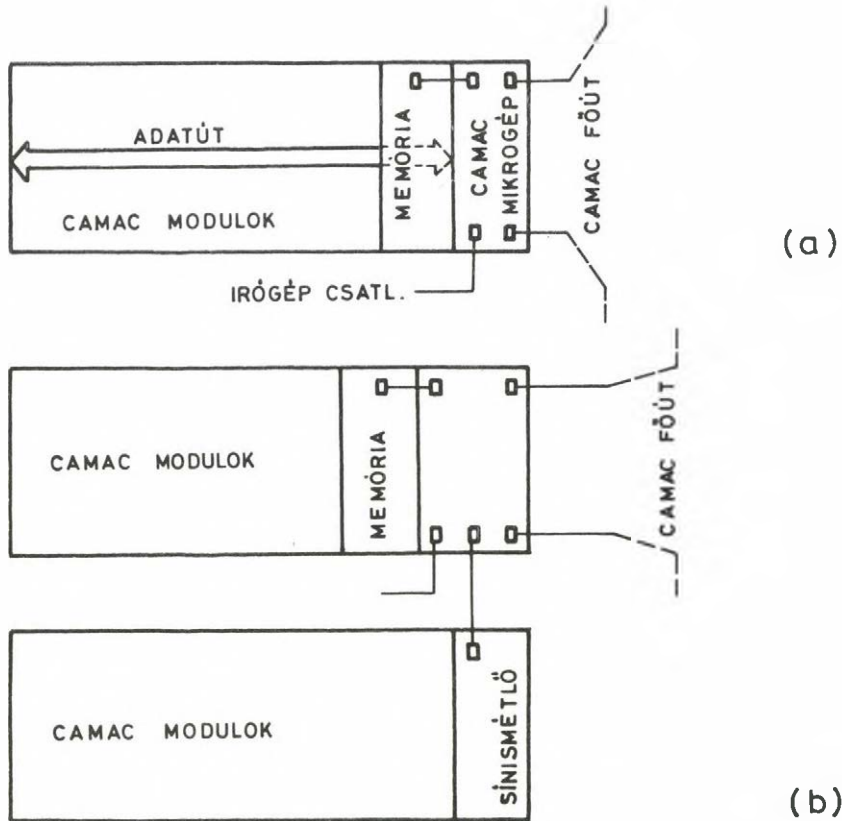
23. ábra.



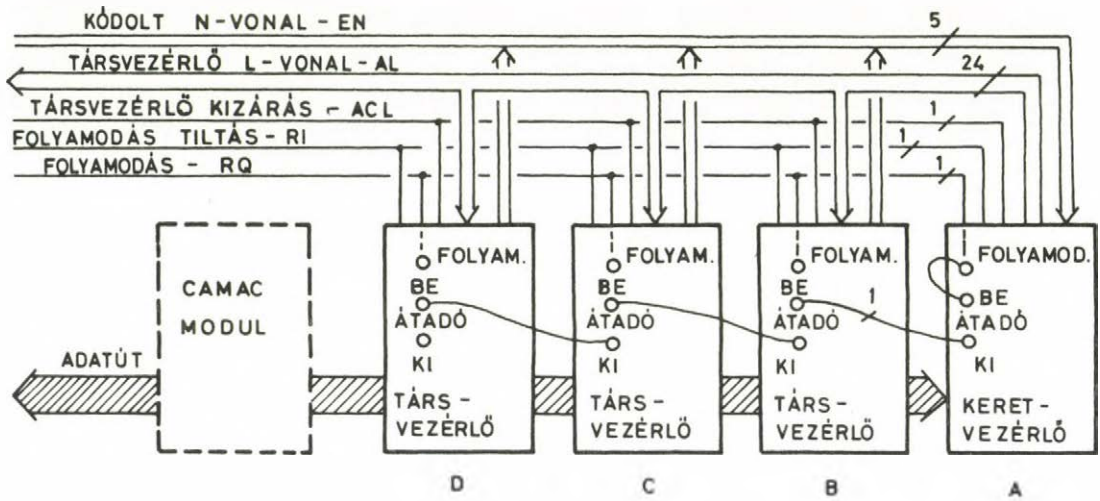
24. ábra.



25. ábra.



26. ábra.



27. ábra.

CAMAC AKCIÓK

MODUL - AKCIÓK			
EGYSZERES	EGYKÉSZÜLÉKŰ BLOKKÁTVITEL	TÖBBKÉSZÜLÉKŰ BLOKKÁTVITEL	
- ÍRÁS - OLVASÁS - MŰKÖDTETÉS - ÁLLAPOT OLVASÁS	- ÍRÁS - OLVASÁS	- ÍRÁS - OLVASÁS - MŰKÖDTETÉS	VÉGIGJÁRÁSSAL
		- ÍRÁS - OLVASÁS - MŰKÖDTETÉS	

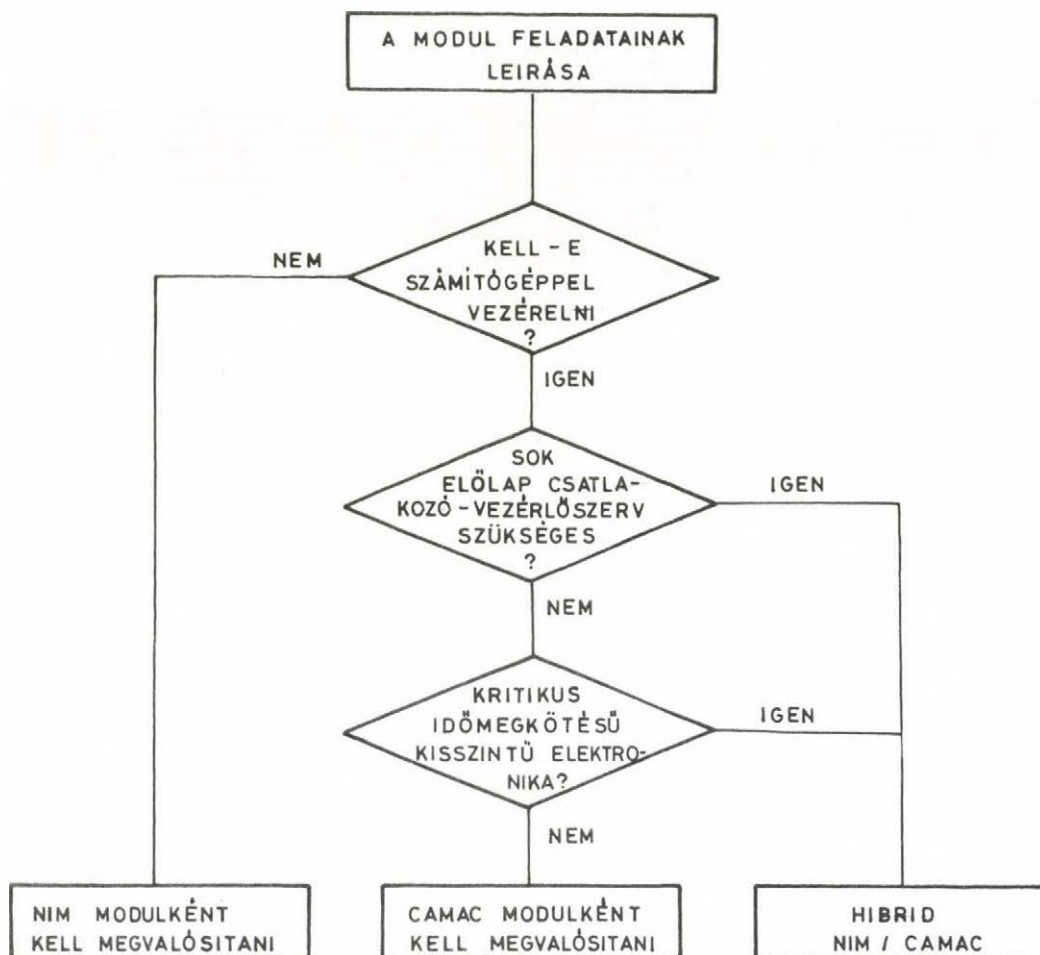
LAM AKCIÓK

- ENGEDÉLYEZÉS
- FELFÜGGESZTÉS
- TÖRLÉS
- FELTÉTELES UGRÁS
(NÉGYFÉLE)
- OLVASÁS
- ÁLLAPOTOLVASÁS
- LAM REGISZTER
OLVASÁS
- MASZK OLVASÁS
- MASZK ÍRÁS

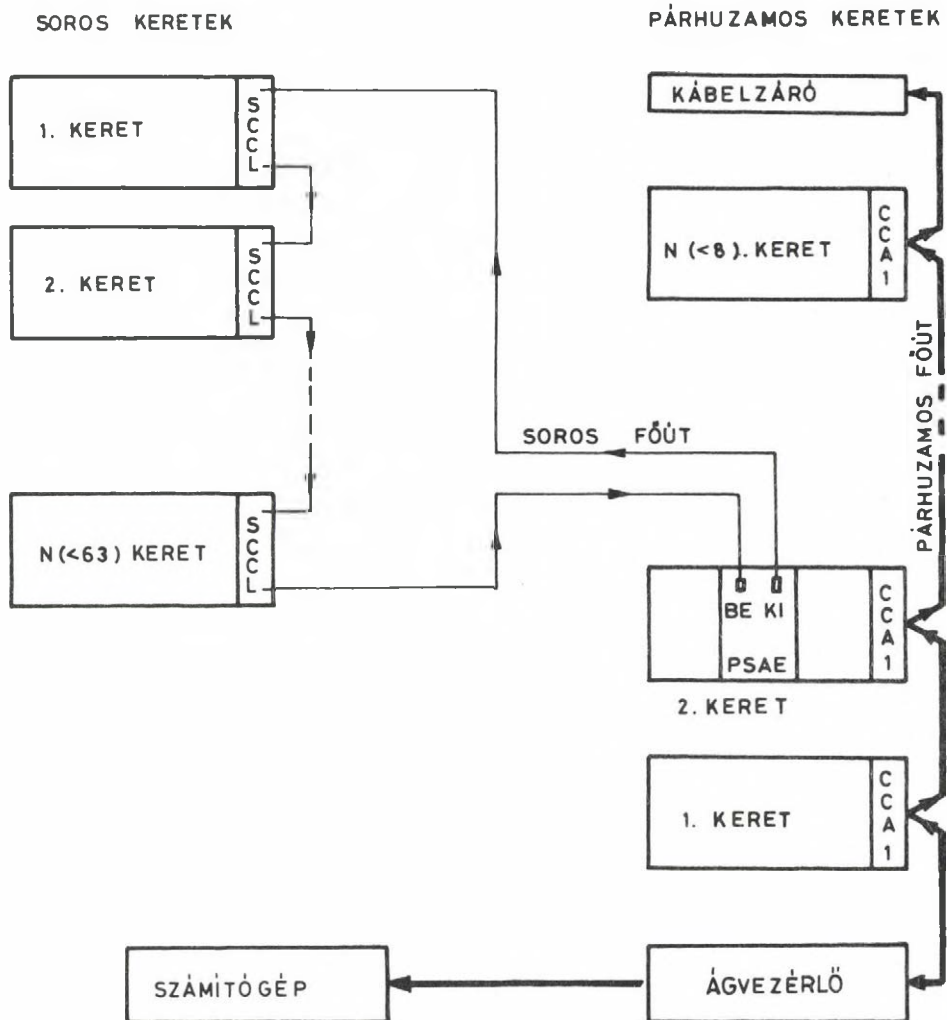
RENDSZER AKCIÓK

KERETVEZÉRLŐ	RENDSZERVEZ.
- ALAPBEÁLLÍTÁS	- ALAPBEÁLLÍTÁS
- KERET TÖRLÉS	- IGÉNY ENGEDÉLYEZÉS
- POZÍCIÖREGISZ- TER TÖRLÉS	- IGÉNY FELFÜGGESZTÉS
- TILTÁS BEÁLLÍTÁS	- X-HIBA ÉRZÉKEL. ENGEDÉLYEZÉS
- TILTÁS VISSZA- ÁLLÍTÁS	- X-HIBA ÉRZÉKEL. FELFÜGGESZTÉS
- KERETIGÉNY ENGEDÉLYEZÉS	- GL MINTÁZAT OLVASÁSA
- KERETIGÉNY FELFÜGGESZTÉS	- Q - ÁLLAPOT OLVASÁSA
- FELTÉTELES UG- RÁS (HATFÉLE)	- Q - ÁLLAPOT VIZSGÁLATA
- KERETIGÉNY KIOLVASÁS	
- ÁLLAPOTSZÓ KIOLVASÁS	

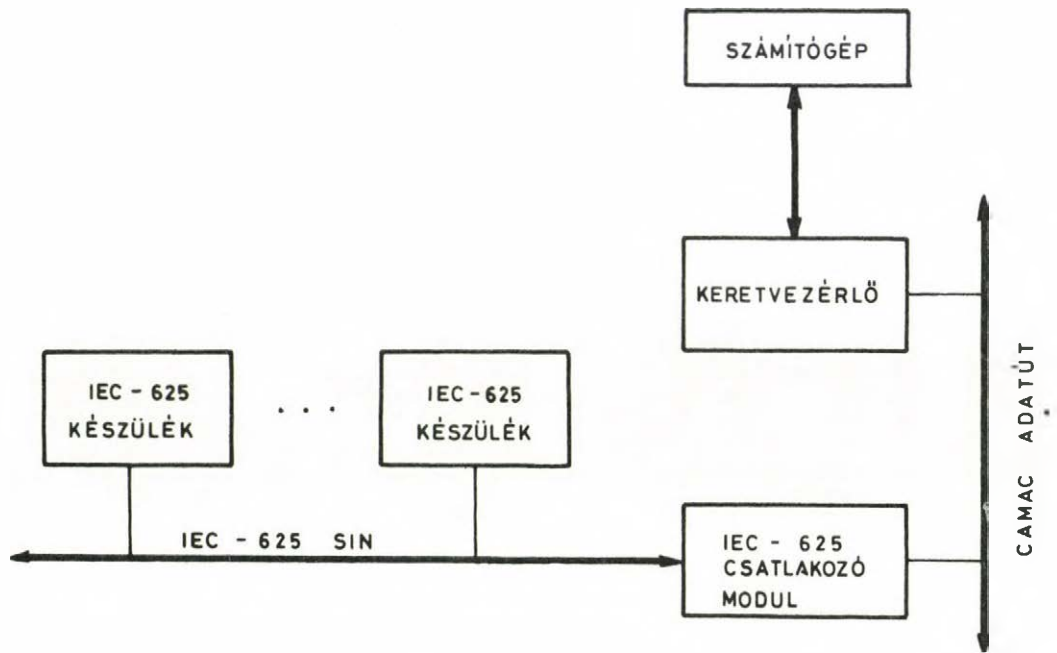
28. ábra.



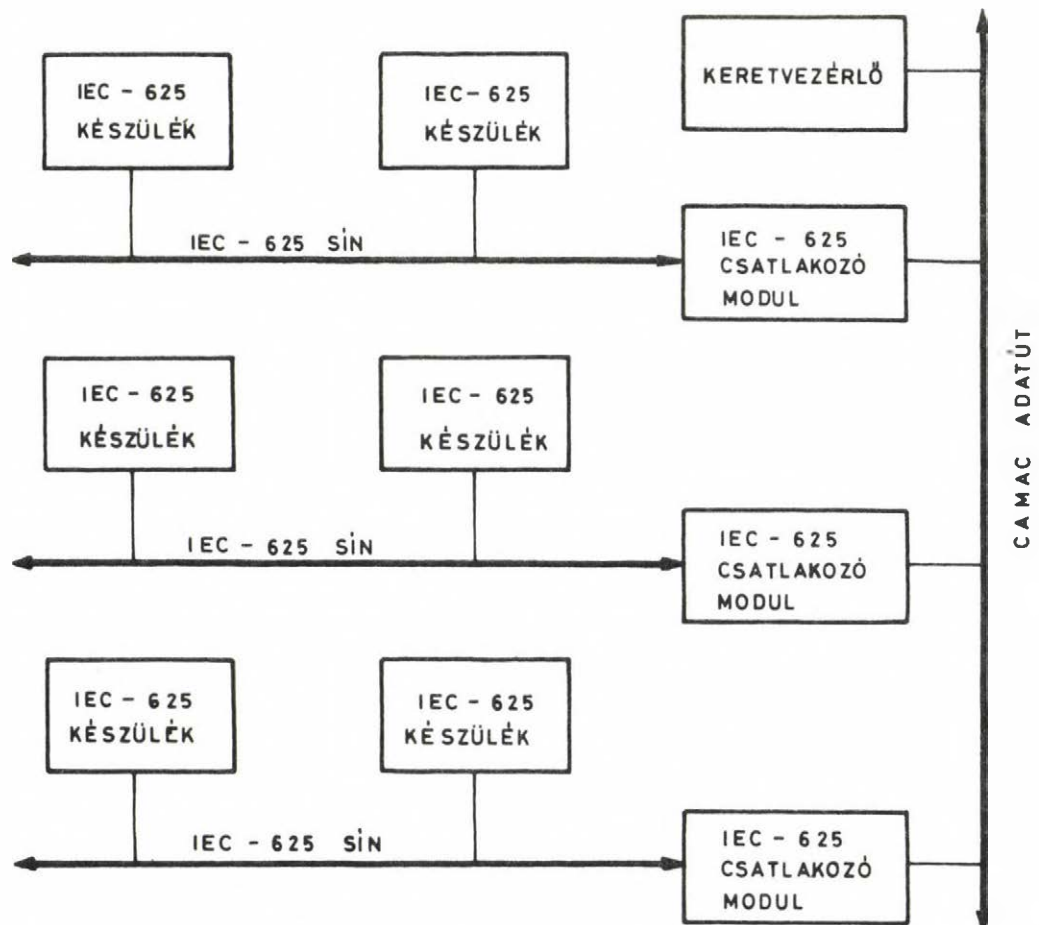
29. ábra.



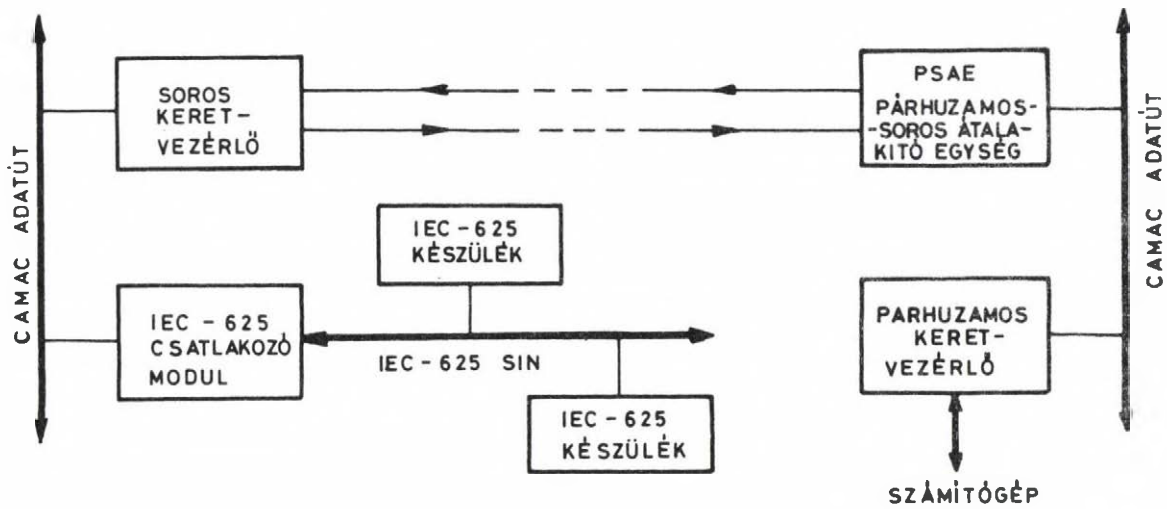
30. ábra.



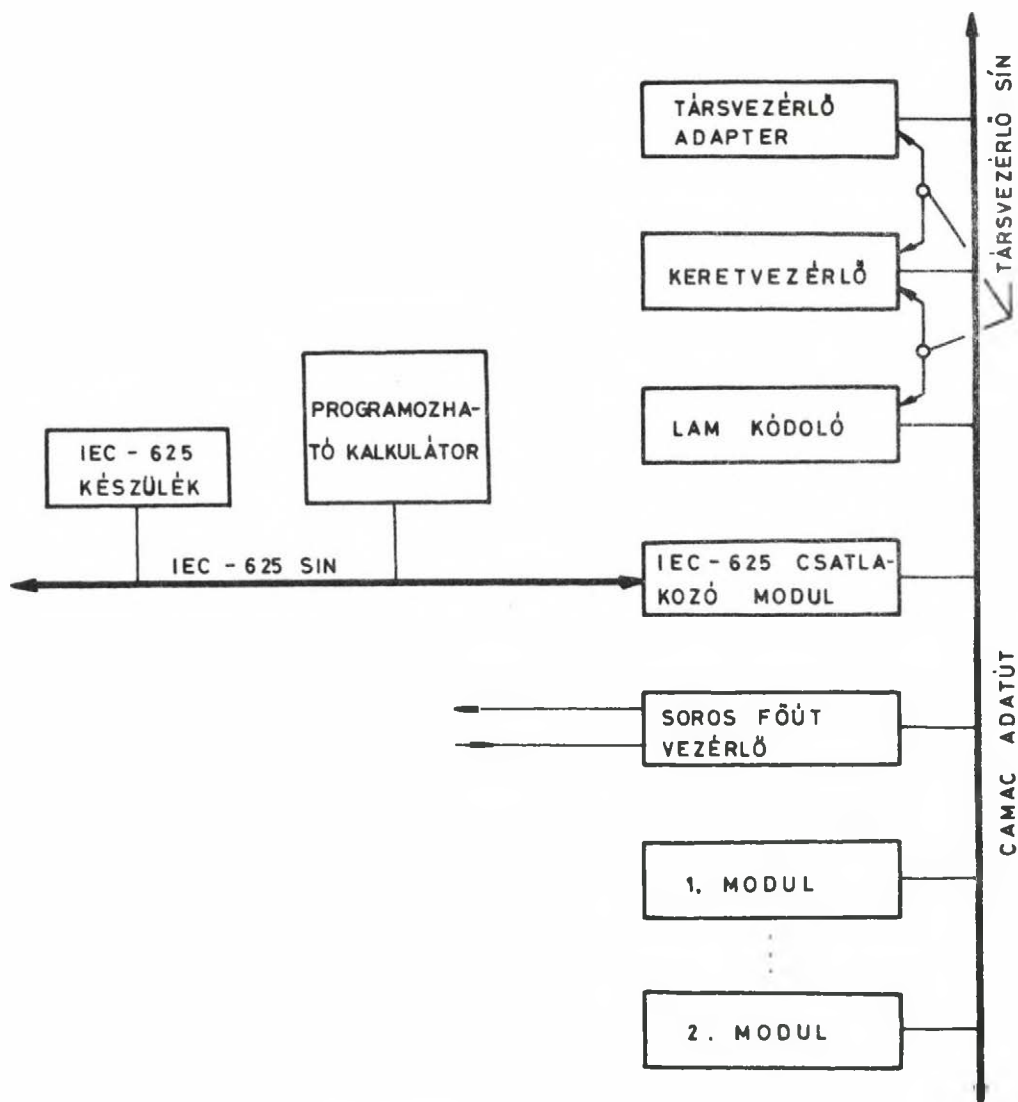
31. ábra.



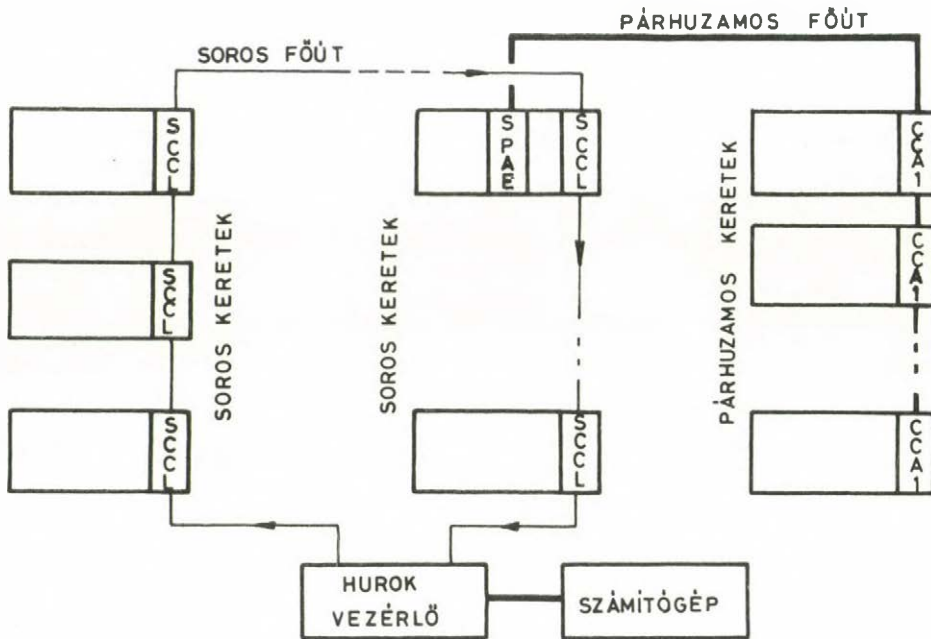
32. ábra.



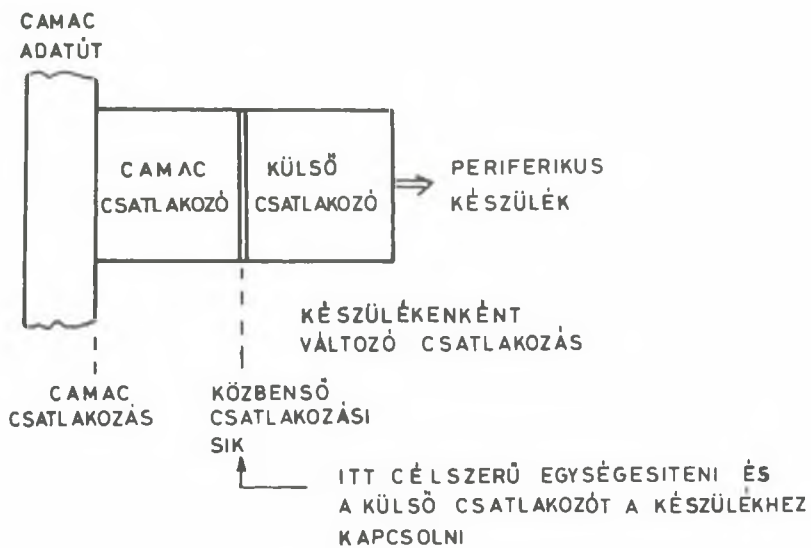
33. ábra.



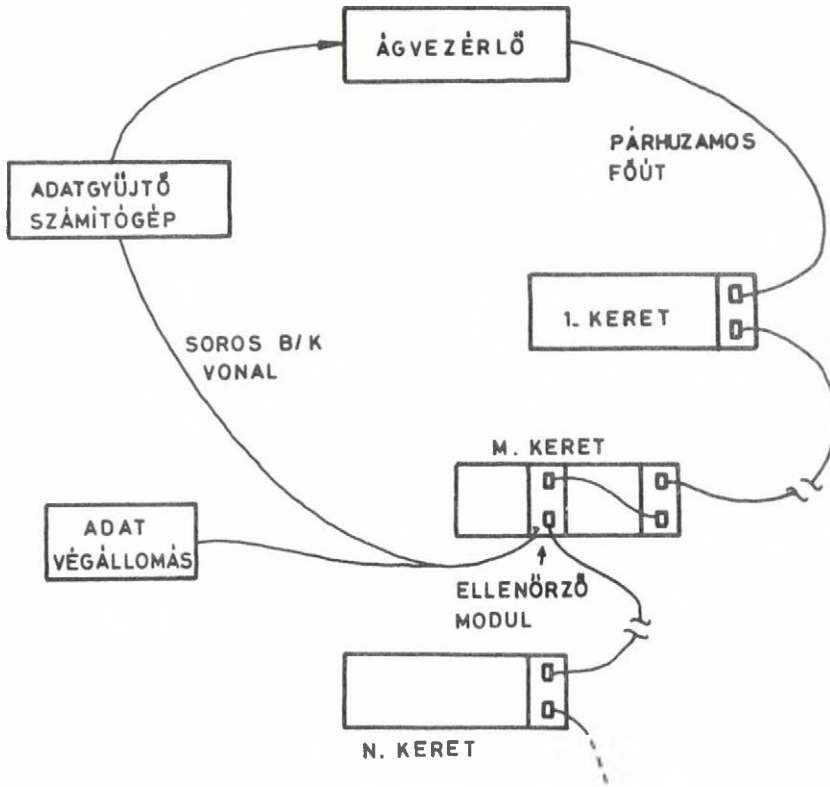
34. ábra.



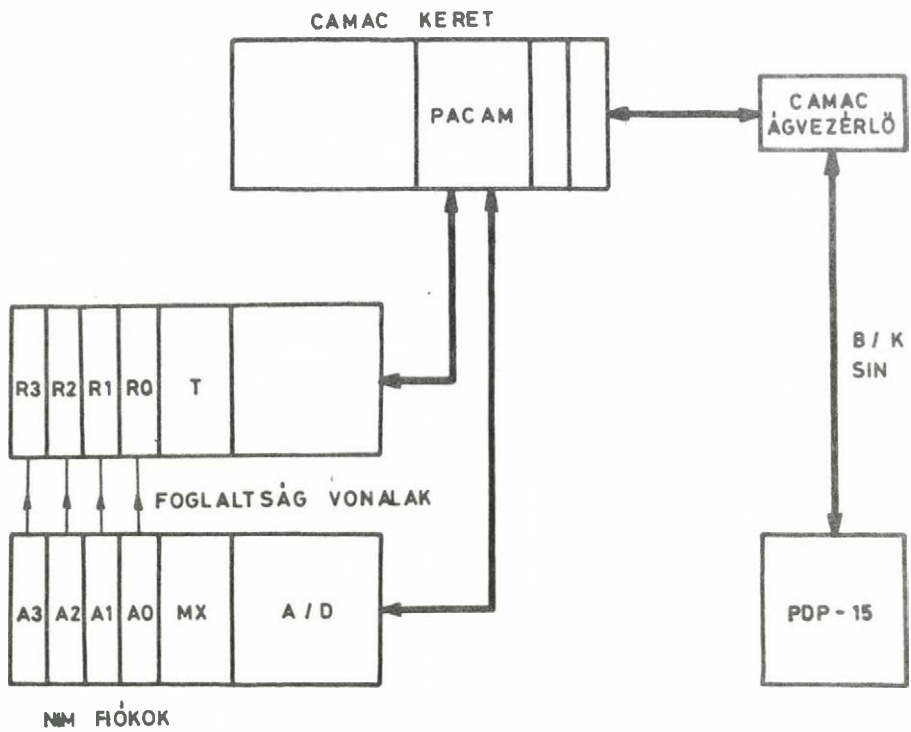
35. ábra.



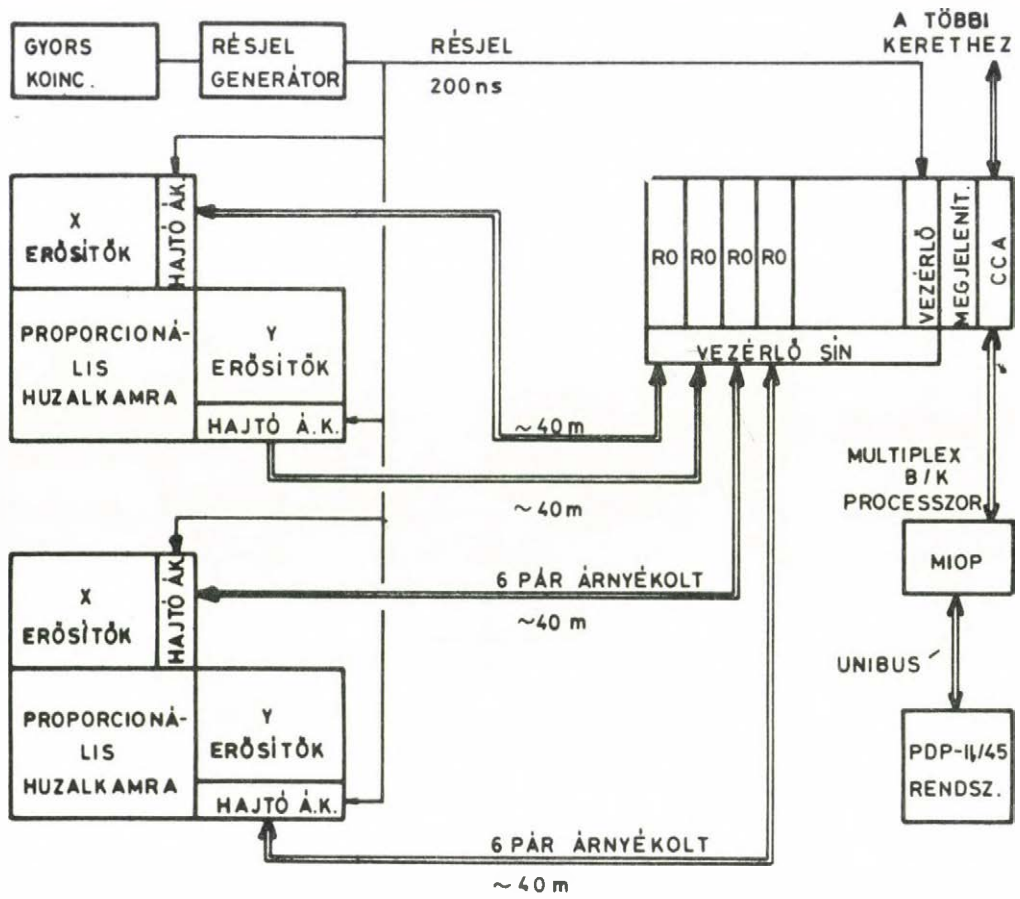
36. ábra.



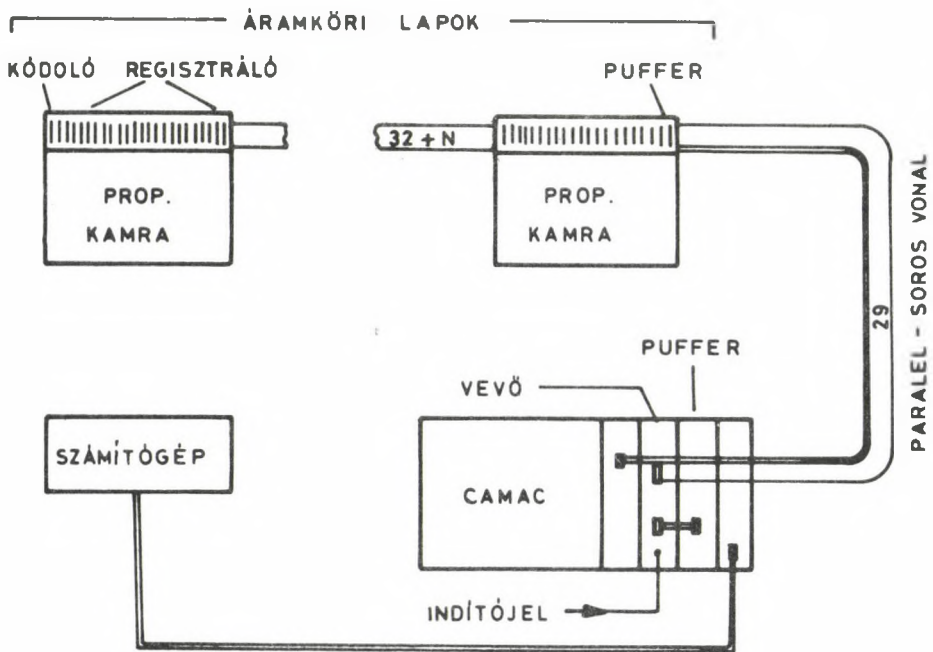
37. ábra.



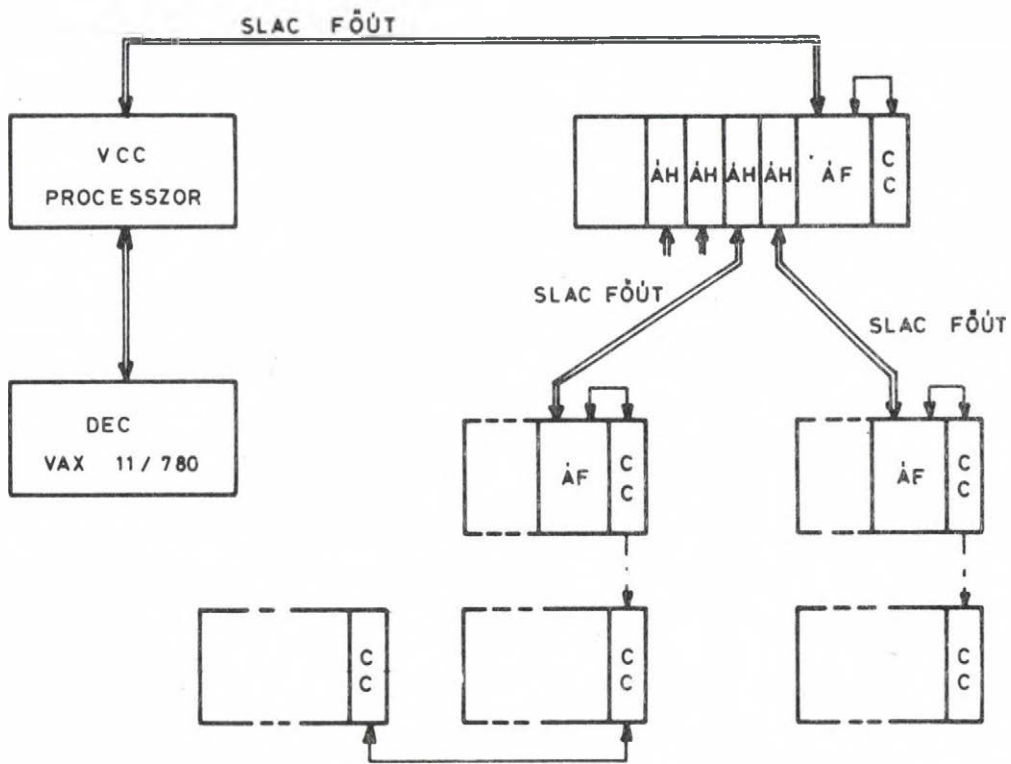
38. ábra.



39. ábra.

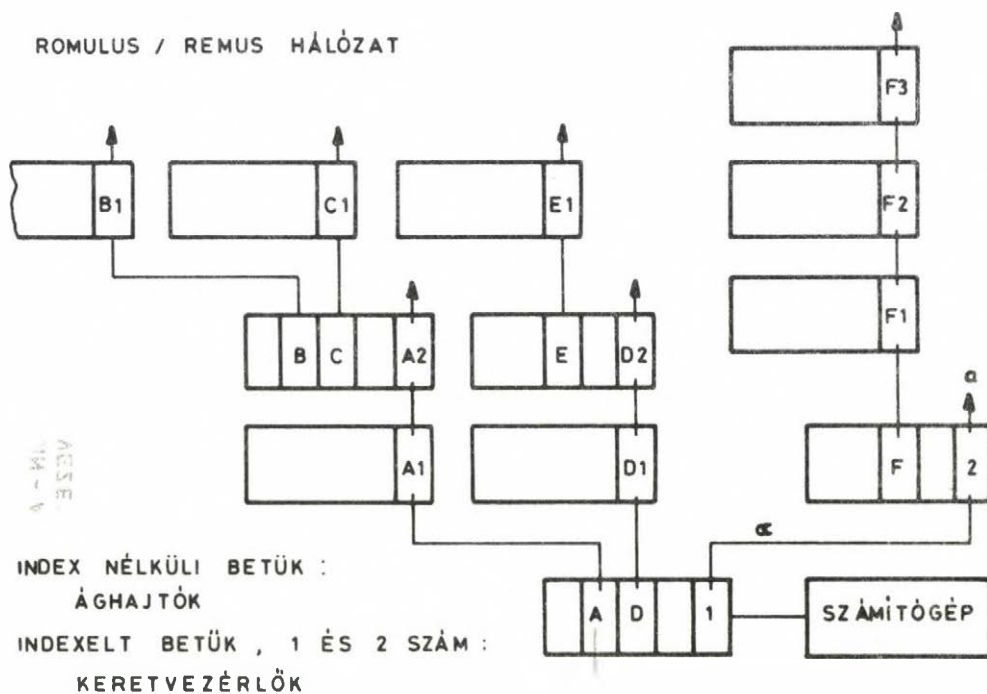


40. ábra.



ÁF - ÁGFOGADÓ ; ÁH - ÁGHAJTÓ ; CC - KERETVEZÉRLŐ

41. ábra.

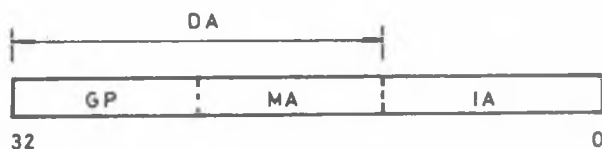


42. ábra.

JELÖLÉS	MEGNEVEZÉS		VEZETÉK SZÁM	JEL TÍPUS	FORRÁS		
	ANGOL	MAGYAR					
AR	ARBITRATION REQUEST	ARBITRÁCIÓ FOLYAMODÓ	1	A	M	} SÍN - ARBITRÁCIÓ VEZETÉKEI	
AG	ARBITRATION GRANT	ARBITRÁCIÓ ÁTENGEDŐ	1	TA	Anc		
GK	GRANT ACKNOWLEDGE	ÁTENGEDÉS NYUGTÁZÓ	1	TA	M		
AI	ARBITRATION REQU. INHIBIT	ARBITR. FOLYAMODÁS TILTÓ	1	CA	Anc		
AL	ARBITRATION LEVEL	ARBITRÁCIÓ SZINT	6	IA	M		
AS	ADDRESS SYNC	CÍMSZINKRON	1	T	M	} CÍMVÁLASZTÁS, SZINKRONIZÁLÁS	KIJELÖLŐ VEZETÉK
AK	ADDRESS ACKNOWLEDGE	CÍMNYUGTÁZÓ	1	T	S / Anc		
DS	DATA SYNC	ADATSZINKRON	1	T	M	} ADATVÁLASZTÁS, SZINKRONIZÁLÁS	
DK	DATA ACKNOWLEDGE	ADATNYUGTÁZÓ	1	T	S		
MS	MODE SELECT	MÓDVÁLASZTÓ	3	C	M	} VÁLTOZÓ SZEREPKÖRŰ VEZETÉKEK	
RD	READ	OLVASÁS / ÍRÁS	1	C	M		
SS	SLAVE STATUS	VEZETETT ÁLLAPOTA	3	I	S		
AD	ADDRESS / DATA	CÍM / ADAT	32	I	M/S		
PA	PARITY	PARITÁS	1	I	M/S		
PE	PARITY ENABLE	PARITÁS - KÉPESÍTŐ	1	I	M/S		
SR	SERVICE REQUEST	SZOLGÁLTATKÉRŐ	1	A	M/S	} KÜLÖNBÖZŐ SZOLGÁLTATI VONALAK	
WT	WAIT	VÁRATÓ	1	A	M/S		
RB	RESET BUS	SÍN - VISSZAÁLLÍTÓ	1	A	H		
BH	BUS HALTED	SÍNZÁRÓ	1	C	Anc		
EG	ENABLE GEOGRAPHICAL	FÖLDRAJZI KÉPESÍTŐ	1	CD	M / Anc		
TX	SERIAL LINE TRANSMIT	SOROS VONALADÓ	1	A	M/S	} DIAGNOSZTIKAI CÉLOKRA SOROS VONAL	
RX	SERIAL LINE RECEIVE	SOROS VONALVEVŐ	1	A	M/S		
DL	DAISY CHAIN LEFT	GYÖNGYVEZETÉK BAL (KAPOCS)	3	I	M/S	} NORMÁLIS GYORSSÍN PROTOKOLLTÓL FÜGGETLEN GYÖNGYVEZETÉKEK	
DR	DAISY CHAIN RIGHT	GYÖNGYVEZETÉK JOBB (KAPOCS)	3	I	M/S		
GA	GEOGRAPHICAL ADDRESS PINS	FÖLDRAJZI-CÍM KAPCSOK	5	F	HUZALOZ.	} BINÁRISAN KÓDOLT FIX CÍM SPEC. CÉLŰ	
TP	T - PIN (NOT BUSSED)	T - KAPOCS (NEM SINVEZETÉK)	1	I	S		
R	RESERVED	FENNTARTOTT	24				

A KERETSZEGMENSNÉL ÉS A KÁBELSZEGMENSNÉL EGYARÁNT MEGLÉVŐ VONALAK - ALAPSÍN
CSAK A KERETSZEGMENSNÉL

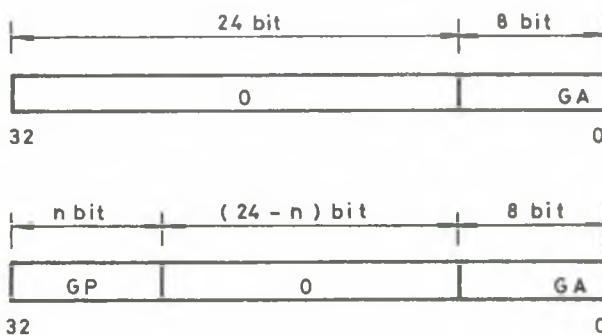
JELÖLÉSEK : A - ASZINKRON ; C - VEZÉRLŐ ; CA - ARBITRÁCIÓ - VEZÉRLŐ ; CD - KÉSLELTETETT VEZÉRLÉS ; F - KONST.
I - INFORMÁCIÓ ; IA - ARBITRÁCIÓ - INFORMÁCIÓ ; T - IDŐZÍTŐ - VÁLASZTÓ ; TA - ARBITRÁCIÓ IDŐZÍTŐ
JELFORRÁSOK JELÖLÉSEI : Anc - TARTOZÉK ÁRAMKÖR ; H - GAZDA ; M - VEZETŐ ; S - VEZETETT ;



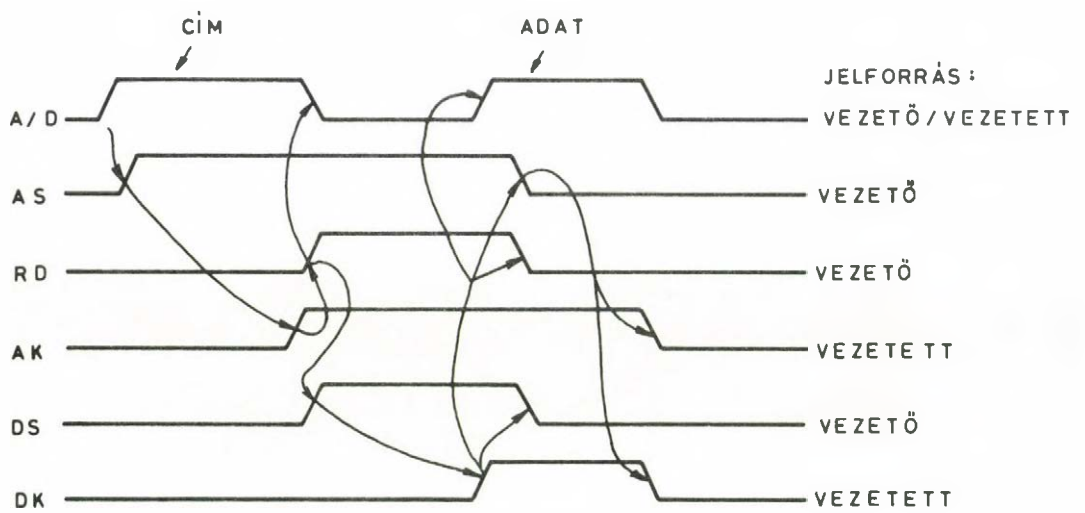
44. ábra.

MS	MS JELENTÉSE	
	ELSŐDLEGES CÍMCIKLUSBAN	ADATCIKLUSBAN
< 2 : 0 >		
0	KÉSZÜLÉKCÍM - ADATTÉR	SORREND NÉLKÜLI ADAT
1	KÉSZÜLÉKCÍM - CSR TÉR	BLOKKÁTVITEL - PAROLA MÓD
2	KÖZZÉTETT CÍM - ADATTÉR	MÁSODLAGOS CÍM
3	KÖZZÉTETT CÍM - CSR TÉR	BLOKKÁTVITEL - SZINKRON MÓD
4 - 7	TARTALÉKOLVA	TARTALÉKOLVA

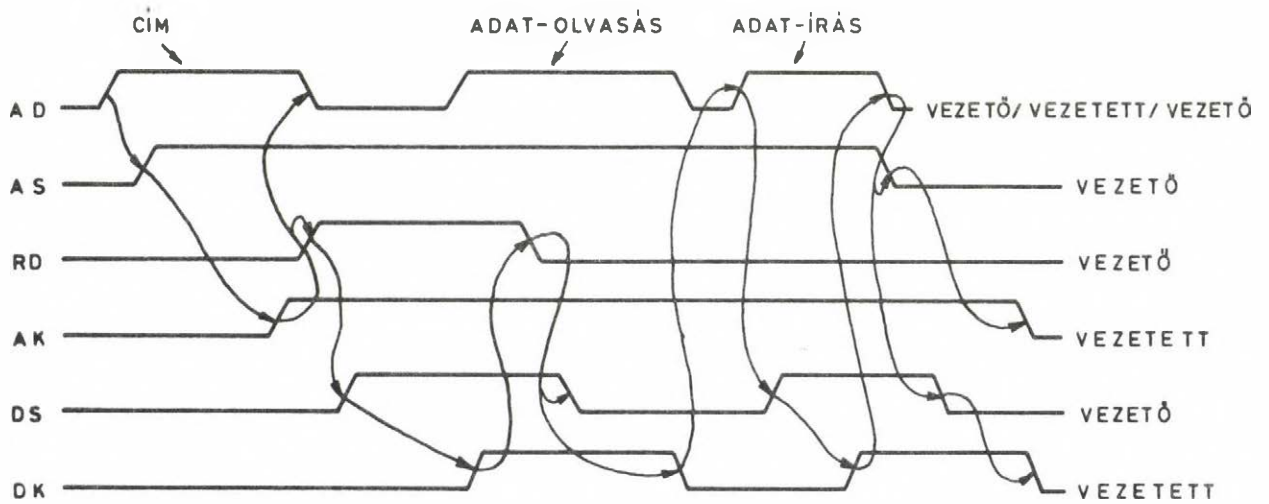
45. ábra.



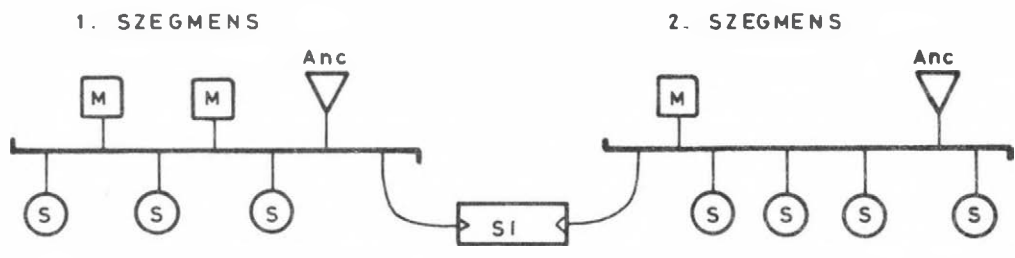
46. ábra.



47. ábra.

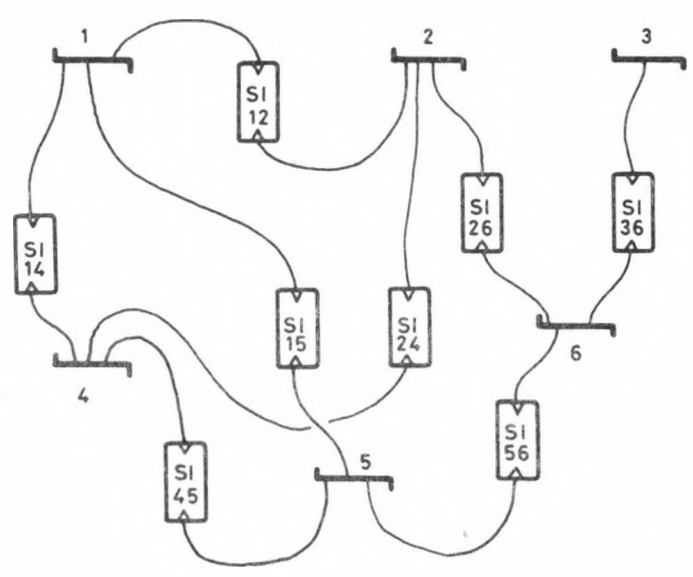


48. ábra.

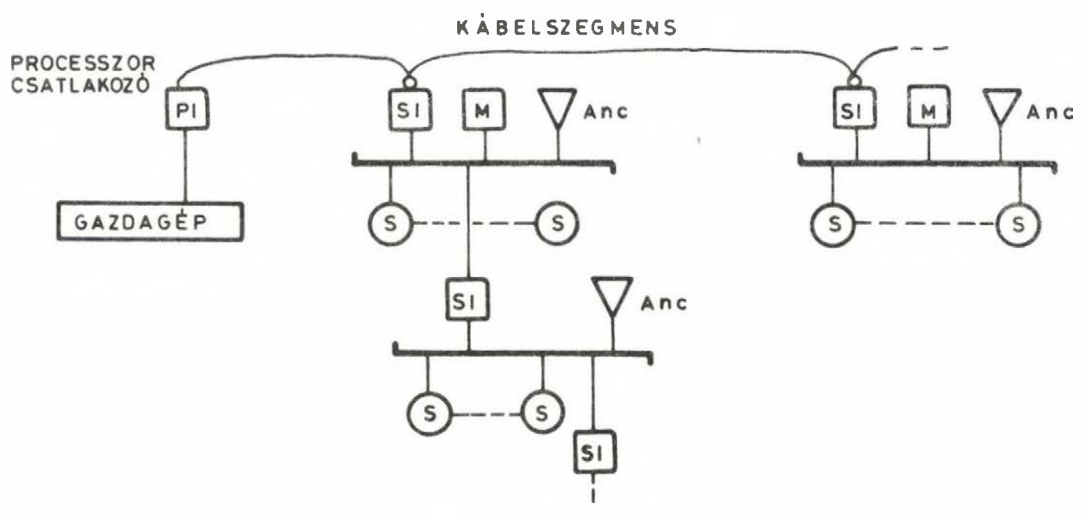


JELÖLÉSEK : M - VEZETŐ ; S - VEZETETT ; Anc - TARTOZÉK ÁK.

49. ábra.



50. ábra.



51. ábra.

SS <2:0>	SS ÉRTELMEZÉSE					
	VÁLASZ CÍMCIKLUSBAN AK(U) - VAL	VEZETETT VÁLASZA ADATCIKLUSBAN DK(t) - VEL				
		ADAT VAGY CSR TÉR		MÁSODCÍM		BELSŐ CÍM VÁLTOZÁS
		ÍRÁS	OLVASÁS	ÍRÁS	OLVASÁS	
0	CÍM FELISMERVE	ELFOGADVA	ADAT ÁLLITVA	ELFOGADVA	ADAT ÁLLITVA	MEGENGEDVE
1	HÁLÓZAT FOGLALT	VISSZAVETVE	NINCS ADAT	VISSZAVETVE	NINCS ADAT	NINCS
2	HÁLÓZATI HIBA	VISSZAVETVE	NINCS ADAT	VISSZAVETVE	N/A	NINCS
3	HÁLÓZAT ELVET	VISSZAVETVE	NINCS ADAT	VISSZAVETVE	NINCS ADAT	NINCS
4	NEM STANDARD	VISSZAVETVE	N/A	VISSZAVETVE	N/A	NINCS
5	HIBÁS BELSŐ CÍM	ELFOGADVA	ADAT ÁLLITVA	ELFOGADVA	ADAT ÁLLITVA	MEGENGEDVE
6	NEM STANDARD	VISSZAVETVE	N/A	VISSZAVETVE	N/A	NINCS
7	TARTALÉKOLT	ELFOGADVA	ADAT ÁLLITVA	ELFOGADVA	ADAT ÁLLITVA	MEGENGEDVE

52. ábra.

A TANULMÁNYSOROZATBAN 1982-BEN MEGJELENTEK

- 130/1982 Barabás Miklós - Tőkés Szabolcs: A lézer printer képalkotás hibái és optikai korrekciójuk
- 131/1982 RG-II/KNVVT "Szisztemü upravlenija bazani dannüh i informacionrüe szisztemü" Szbornik naucsno-iszszledovatel'szkih rabot rabocsej gruppü RG-II KNVVT, Bp. 1979. T o m I.
- 132/1982 RG-II/KNVVT T o m II.
- 133/1982 RG-II KNVVT T o m III.
- 134/1982 Knuth Előd - Rónyai Lajos: Az SDLA/SET adatbázis lekérdező nyelv alapjai /orosz nyelven/
- 135/1982 Néhány feladat a tervezés-automatizálás területéről. Örmény-magyar közös cikkgyűjtemény
- 136/1982 Somló János: Forgácsoló megmunkálások folyamatainak optimalási és irányítási problémái
- 137/1982 KGST I-15.1. Szakbizottság 1979. és 80. évi előadásai
- 138/1982 Kovács László: Számítógép-hálózati protokollok formális specifikálása és verifikálása
- 139/1982 Operációs rendszerek elmélete 7. visegrádi téli iskola

A TANULMÁNYSOROZATEAN 1983-BAN MEGJELENTEK

- 140/1983 Operation Research Software Descriptions (Vol.1.)
Szerkesztette: Prékopa András és Kéri Gerzson
- 141/1983 Ngo The Khanh: Prefix-mentes nyelvek és egyszerű
determinisztikus gépek
- 142/1983 Pikler Gyula: Dialógussal vezérelt interaktiv
gépészeti CAD rendszerek elméleti és gyakorlati
megfogalmazása
- 143/1983 Márkus Zsuzsanna: Modellelméleti és univerzális
algebrai eszközök a természetes és formális nyelvek
szemantikaelméletében
- 144/1983 PUBLIKÁCIÓK '81 Szerkesztette: Petróczy Judit
- 145/1983 Telcs András: Belső állapotú bolyongások
- 146/1983 Varga Gyula: Numerical Methods for Computation of
the Generalized Inverse of Rectangular Matrices
- 147/1983 Proceedings of the joint Bulgarian-Hungarian work-
shop on "Mathematical Cybernetics and data Processing"
Szerkesztette: Uhrin Béla



