

Biri János - Lukács József

GAZDAG

perifériarendszer

BIRI JÁNOS — LUKÁCS JÓZSEF

CAMAC PERIFÉRIARENDSZER

A mérések, ipari folyamatok automatizálásánál gyorsan terjednek a számítógépes megoldások. A számítógépeket a „külvilághoz” — tehát a laboratóriumi, ipari mérőberendezésekhez, beavatkozó szervekhez az ún. real-time perifériák kapcsolják: ezek alakítják át a különböző analóg és digitális jeleket a számítógép által „érthető” formába. Ilyen jellegű számítógépes csatlakoztatásra dolgozták ki a nemzetközileg elfogadott CAMAC szabványt.

A könyv ismerteti a CAMAC szabványokon alapuló rendszerek előírásait és a használat során kialakult gyakorlati tapasztalatokat. Részletesen foglalkozik a különböző hardware és software elemekkel és végül gyakorlati feladatok megoldását, alkalmazási példákat is bemutat.

A könyvet azon rendszertervezők, felhasználók és mérnökök részére ajánljuk, akik számítógépes mérési, automatizálási, ipari folyamatszabályozási feladatokkal foglalkoznak, de hasznos gyakorlati segédkönyvet is jelent az egyetemi és felsőoktatásban is.

MŰSZAKI KÖNYVKIADÓ

Biri János • Lukács József

CAMAC perifériarendszer

Biri János • Lukács József

CAMAC

perifériarendszer

Műszaki Könyvkiadó, Budapest, 1976

Lektorálta:

Horváth István

okl. villamosmérnök

© Biri János — Lukács József, 1976

ETO: 681.337

ISBN: 963 10 1273 5

Felelős kiadó: Solt Sándor igazgató

Felelős szerkesztő: Kemény Tamásné okl. villamosmérnök

Tartalomjegyzék

- 1. A CAMAC rendszer helye és szerepe 9
 - 1.1. A real-time perifériák szerepe, a CAMAC rendszer kialakulása 9
 - 1.2. Általános fejlődési irányok a mérésautomatizálással és folyamatszabályozással kapcsolatban 11
 - 1.3. CAMAC az általános fejlődés tükrében 12
 - 1.3.1. CAMAC és a korszerű hardware-software rendszerek 12
 - 1.3.2. CAMAC az újabb alkalmazási területeken 14
 - 1.3.3. Szervezeti változások 15
 - 1.4. CAMAC a KGST-ben 15
- 2. Általános elvek, CAMAC előírások 17
 - 2.1. A CAMAC rendszer főbb jellemzői 17
 - 2.1.1. CAMAC egységek 17
 - 2.1.2. CAMAC rendszerek, FŐÚT-ak 17
 - 2.1.3. Üzenetformátumok, jelforgalom 22
 - 2.2. A CAMAC rendszer részletes ismertetése 23
 - 2.2.1. A szabványokat tartalmazó kiadványok áttekintése 23
 - 2.2.2. CAMAC mechanika 26
 - 2.2.3. CAMAC ADATÚT 31
 - 2.2.4. CAMAC CSOPORT, CSOPORTÚT 39
 - 2.2.5. Soros CAMAC, SOROS FŐÚT 44
- 3. CAMAC vezérlők 55
 - 3.1. Egyszerű ADATÚT és CSOPORTÚT vezérlők 56
 - 3.1.1. Egyszerű programozott vezérlők 57
 - 3.1.2. Autonóm működésű egyszerű vezérlők 62
 - 3.1.3. A1 típusú keretvezérlő 65
 - 3.1.4. Figyelj-rám (LAM) kezelés 69
 - 3.1.5. Több-csoportutas rendszerek. Rendszervezérlők 73
 - 3.1.6. Vezérlőkkel együttműködő kiegészítő egységek 74

3.2.	Soros vezérlők	75
3.3.	Összetett vezérlők	77
3.3.1.	Önálló CAMAC processzorok	78
3.3.2.	Számítógéphez kapcsolódó CAMAC processzorok	79
3.3.3.	Általános vezérlőprocesszorok	80
3.4.	Speciális vezérlők	84
4.	Jellegzetes CAMAC modulok	87
4.1.	A modulok általános felépítése	87
4.2.	Általános célú és funkcionális modulok	93
4.2.1.	Digitális bemeneti modulok	94
4.2.2.	Digitális kimeneti modulok	99
4.2.3.	Periféria- és műszerezésvezérlő modulok	101
4.2.4.	Digitális kezelő- és feldolgozó modulok	105
4.2.5.	Analog modulok (Bemeneti analog modulok — Kimeneti analog modulok — Analogjel-kezelő modulok)	107
4.3.	Modulcsaládok	118
4.3.1.	Moduláris oszcilloszkópos megjelenítő rendszer	118
4.3.2.	Ipari analog mérőrendszer	119
5.	CAMAC software	123
5.1.	Az IML nyelv	124
5.1.1.	Deklarációk	125
5.1.2.	Utasítások	129
5.1.3.	Példaprogram IML nyelven	136
5.1.4.	Követelmények a makrofordítóval szemben	137
5.2.	CAMAC programozás egyszerű assembler szinten	141
5.3.	CAMAC programozás magasabb szintű nyelven	142
5.3.1.	BASIC nyelvű implementáció	143
5.3.2.	FORTTRAN-ba épített CAMAC szubrutinok	145
5.3.3.	CAMAC programozás problémaorientáltan	146
6.	Alkalmazások	149
6.1.	Nukleáris alkalmazások	149
6.2.	Fizikai, kémiai analitikai mérések	154
6.3.	Orvosi-biológiai rendszerek	158
6.4.	Ipari alkalmazások	160
6.5.	Egyéb alkalmazások	164
7.	Függelék	169
7.1.	CAMAC utasítások és jelek táblázatai	169
7.2.	CAMAC műszerezés és software decimális osztályozása	176

Előszó

A könyv CAMAC real-time perifériákról, rendszerekről szól. Ebben a tárgykörben úttörő jellegű, hiszen összefoglaló mű még nem jelent meg ebből sem hazánkban, sem külföldön. Sőt a real-time rendszerekről ilyen — a mechanikájától a software rendszeréig — átfogó képet adó könyv tudomásunk szerint egyáltalán nem jelent meg. Ennek két fő oka van: a CAMAC rendszer előtt nem volt általános real-time periféria-rendszer, hanem csak egyes gyártó cégek egyedi, helyi jelentőségű megoldásai léteztek, és így ezekről inkább gépkönyvszerű leírások születtek. A másik ok az, hogy a tárgykör nagyon újszerű, eddig inkább cikkekben jelentek meg leírások az egyes részekről. Bár a könyv tárgyának súlypontja a jelenlegi megoldások és alkalmazások ismertetése a különböző, többnyire még kevésbé publikált kutatási irányokra támaszkodva, megpróbáljuk a CAMAC jövőjét is vázolni.

A tárgyalási módot a tárgykör újdonságának és az általunk legfontosabbnak ítélt igényeknek kielégítése szempontja szerint választottuk. Ez a könyv nem tankönyv. A tankönyvek egyik alapvető vonása, hogy leszűródött, időt álló igazságokat tárgyalnak oly módon, hogy a szerző határozott véleményt nyilvánít még vitás kérdésekben is. E tárgykör viszont olyan gyorsan fejlődött az elmúlt években, hogy csak átfogó képet tudunk adni a CAMAC real-time rendszerek jelenlegi helyzetéről, a problémákat úgy tárgyalva, hogy többféle véleményt is ismertetünk, hangsúlyozva természetesen az általunk helyesnek tartott irányokat. Így nem sztatikusan, hanem dinamikusan, időbeni fejlődésében mutatjuk be a tárgykört. E mellett igen sok, tényleges megoldást bemutató példát közlünk, hogy ezáltal is bővüljön az Olvasónak a tárgykörrel kapcsolatos szemlélete.

A tárgyalás módjának másik jellegzetessége, hogy összefüggéseiben, más témakörökkel összekapcsolva igyekszünk tárgyalni a kérdéseket. E tárgykör igen sok önálló tudományággal van kapcsolatban, igyekeztünk e területekbe annyira behatolni, amennyire az összefüggő kép kialakulásához szükséges volt. Ilyen területek például a korszerű rendszertechnika, real-time programozási nyelvek stb.

Kiknek szól ez a könyv és milyen célra szántuk? Elsősorban a számítástechnikai eszközöket alkalmazó, automatikus mérőrendszereket, folyamatszabályozási rendszereket tervező rendszertervezőknek. A könyvet úgy építettük fel, hogy a rendszertervezőnek két szinten legyen hasznos: egyrészt akkor, amikor valamely feladathoz magát a CAMAC real-time rendszert kiválasztja és mérlegeli annak előnyeit és hátrányait, másrészt, amikor a kiválasztása után a rendszert konkrétan megtervezi. Ezért az első fejezet, majd a további fejezetek első részei általános összefoglalást adnak a nagyvonalakban való tájékozódásra. A sok valódi megoldás tárgyalása segítséget nyújt a részletes rendszertervezéshez, továbbá a kiegészítő modulok tervezőinek, software készítőinek, és ilyen értelemben kézikönyv szerű is. Nem szántuk ezt a könyvet viszont a hivatásszerűen CAMAC modulokat tervezők, ill. software készítők kézikönyvének. Ehhez két tényező hiányzik: egyrészt nem tartalmazza a CAMAC szabványok egészen részletes, pontos, szabványszerű leírását, mivel a könyvet nem a szabványok helyett kell használni. Másrészt nem célja, és terjedelme miatt nem is tudja, helyettesíteni az áramköri tervezéshez szükséges korszerű és klasszikus elektronikai könyveket (analóg és digitális áramkörök tervezése), ill. a programozáshoz szükséges software könyveket (assemblerok, operációs rendszerek stb.).

A könyv megírásakor egyik legnagyobb gondot a magyar szaknyelv nem eléggé kialakult, nem eléggé egységes volta jelentette. Igyekeztünk a szakszavaknak megfelelő magyar kifejezést találni, ill. ha már volt ilyen, terjeszteni. A fogalmak bevezetésénél általában az angol kifejezést is közöltük.

Végül köszönetet mondunk azoknak, akik a könyv megszületését elősegítették. Igen sok segítséget kaptunk elsősorban az ESONE munkacsoportjainak tagjaitól, így I. N. Hooton-tól, H. Klessmann-tól, H. Halling-tól és A. Lewis-tól, akiknek ezért köszönetet mondunk. A munkacsoportok szakmai vitái, belső tanulmányai sok kérdés tisztázását segítették elő. A könyv elkészülésében igen fontos szerepe volt az MTA Központi Fizikai Kutató Intézet vezetőinek és munkatársainak, mivel lehetőségeket biztosítottak és hatékonyan támogatták a CAMAC rendszerekkel kapcsolatos munkát. Ezért elsősorban Sándory Mihály kandidátus, igazgatót illeti köszönet. Munkatársaink közül szeretnénk kiemelni a CAMAC csoport tagjait és közülük is Somlai Lászlót, aki alkotó módon segítette elő a rendszerek megvalósulását. Sok segítséget kaptunk továbbá a hazai és külföldi, elsősorban NDK és szovjet intézmények munkatársaitól, akik a náluk működő CAMAC rendszereinkkel kapcsolatos tapasztalataikat rendelkezésünkre bocsátották.

Budapest, 1975. április

A Szerzők

1.

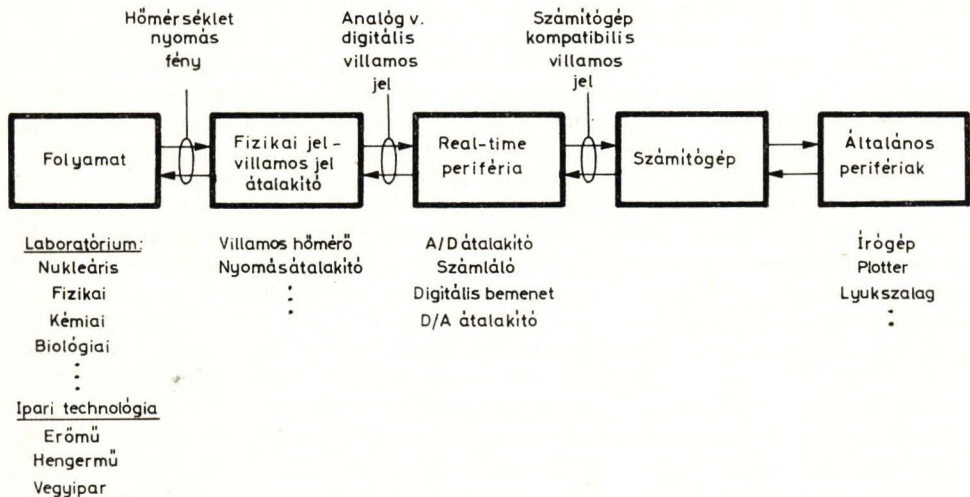
A CAMAC rendszer helye és szerepe

1.1.

A real-time perifériák szerepe, a CAMAC rendszer kialakulása

Az élet különböző területein egyre gyakrabban fordul elő, hogy mérési adatokat közvetlenül digitális számítógépbe kell táplálni, ill. megfordítva, a digitális számítógép adatokat szolgáltat valamely folyamatnak. Az adatok számítógéppel való feldolgozása azért is válik egyre gyakoribbá, mert a korszerű laboratóriumokban, ipari üzemekben mind több adatot gyűjtenek össze, és ezt a hatalmas adattömeget emberi erővel már nemigen lehet feldolgozni, ill. a feldolgozás olyan lassú, hogy közben az adat már el is évülhet. A mérendő folyamatok gyorsasága, a folyamatidentifikációhoz szükséges nagy mennyiségű adat gyors feldolgozása pedig már elképzelhetetlen számítógép nélkül. Ugyanakkor a számítástechnikai eszközök ára világszerte csökken, így egyre több helyen fizetődik ki alkalmazásuk és válik lehetővé használatuk.

A laboratóriumok, ipari üzemek mérési adatai lehetnek: hőmérséklet, nyomás, távolság, idő, frekvencia, darabszám, vastagság stb. Ezeket valamilyen átalakítóval villamos mennyiséggé alakítják át (1.1. ábra). A villamos mennyiség áram- vagy feszült-



1.1. ábra
Real-time perifériák helye

ségnagyság, esetleg impulzusszám. A mérésből származó jelek másik csoportja valamilyen fizikai helyzetet jelez: szelepvégállást, ki-, bekapcsolt állapotot stb. Ugyanígy a számítógépből jövő és a laboratóriumi, ill. ipari folyamatba beavatkozó villamos jelet szintén át kell alakítani fizikai elmozdulássá, fényvé vagy egyéb fizikai mennyiséggé. Az *átalakítók* tehát analóg vagy digitális jelből fizikai mennyiségeket állítanak elő és fordítva.

A számítógépek nem rendelkeznek olyan bemenetekkel, amelyek az átalakítóból érkező analóg, ill. digitális jeleket közvetlenül fogadni tudnák. E célra fejlesztették ki a *real-time perifériákat*, amelyek a bemenetükre érkező analóg vagy digitális villamos jeleket, adatokat a számítógép által érthető formájú villamos jelekké alakítják át, és ezeket az adatokat akár a számítógép kérésére át tudják adni a gépnek, akár pedig meg tudják szakítani a számítógép működésének normális menetét és így jelzik a számítógépnek, hogy az foglalkozzon velük. A real-time perifériák másik csoportja a számítógép által kiadott adatokat alakítja át olyan analóg vagy digitális villamos jelekké, amelyeket a beavatkozó szervek már megértenek. Így például tipikus real-time perifériák az analóg—digitális átalakítók, digitális bemenetek, ill. digitális—analóg átalakítók és digitális kimeneti perifériák.

A számítógépgyártó cégek hamarosan felismerték a real-time perifériák jelentőségét és gyártmányaik közé fölvetették ezeket is. Mivel azonban elsősorban számítógépgyártók maradtak, igyekeztek általános real-time perifériakészletet kifejleszteni, amelyeket az átalakítókhoz sok esetben speciális egységekkel lehetett illeszteni. A real-time perifériarendszerek jelentősége a számítógépes folyamatirányítás lehetőségeinek szélesedésével tovább nőtt.

A felhasználók több ok miatt a számítógép típusától független real-time perifériák kialakítását sürgették. Az egyik ok az volt, hogy az általános tapasztalat szerint a mérésautomatizálásban és folyamatszabályozásban az átalakítók erkölcsi élettartama mostanában húsz év, a real-time perifériáké tíz év, míg a számítógépeké öt év. Ily módon nagyon gazdaságtalan egy számítógéptípushoz kötött rendszer, mivel élettartama, ha független a számítógéptől, kétszerese a számítógéptől függő perifériához képest. A másik ok az, hogy sokkal közelebb lehet kerülni az átalakítóhoz: a különböző típusú átalakítók számára érdemes közvetlenül illeszkedő real-time perifériát kialakítani, amely ezután bármely számítógéphez csatlakozhat. A harmadik ok pedig az, hogyha független a real-time perifériarendszer, és azt több cég gyártja, akkor a felhasználó nem függ egyetlen cégtől, hanem választhat több cég ajánlata között. Számítógéptől független perifériarendszernek viszont csak akkor van gyakorlati jelentősége, ha az valamilyen szabvány szerint készül, és modulárisan, elemekből összerakhatóan bővíthető, ill. változtatható az igényeknek megfelelően.

A fizikai folyamatok számítógépes mérése a nukleáris technikában terjedt el legelőször és itt gyűlt össze a legtöbb tapasztalat is. Ezért először a nukleáris mérésekkel és ilyen folyamatok irányításával foglalkozók körében merült fel a real-time perifériák szabványosításának kérdése. A nagy európai laboratóriumokban összegyűlt tapasztalatok alapján szabványosították nemzetközileg a real-time perifériakészlet mechanikai méreteit, villamos jeleit, bizonyos software megoldásokat: ezt nevezték el CAMAC szabványnak, ill. az ezen szabványok szerint összeállított rendszereket CAMAC rendszereknek. A szabványosító bizottság, az ún. ESONE Bizottság (European Standard on Nuclear Electronics) tagja lehet bármely laboratórium vagy intézet, amely elfogadja a szabályokat. A Bizottság évente összejön, ekkor megvitatják az egyes részfeladatokkal foglalkozó munkacsoportok eredményeit, figyelemmel kísérik a régi szabványokat, újabb problémákra újabb szabványokat dolgoznak ki. Az első szabvány 1968-ban készült el, 1972-ben, majd 1974—75-ben jelentek meg jelentősebb újdonsá-

gok. A szabványt 1972-ben az Egyesült Államok is átvette, azóta az európai és az USA-beli munkacsoportok igen szorosan együttműködnek. Az USA-ban a nukleáris mérés-technika céljára még a számítógépes mérések megjelenése előtt kialakították a NIM (Nuclear Instrumentation Modules) rendszert. Számítógépes mérések céljára a NIM Bizottság elfogadta a CAMAC szabványokat.

1.2.

Általános fejlődési irányok a mérésautomatizálással és folyamat-szabályozással kapcsolatban

A CAMAC szabványok kezdeti kialakulása óta a mérési adatgyűjtésben és folyamat-szabályozásban négy szempont miatt történt igen nagy arányú fejlődés:

- az alkalmazások területe nagymértékben bővült;
- a technológiai lehetőségek megnöttek;
- a rendszerszemlélet megváltozott;
- a szabványosítási törekvés erősödött.

Az *alkalmazási területek* kiszélesedtek. Laboratóriumi téren a bonyolult nukleáris fizikai mérések mellett megnőtt a kémiai, biológiai mérések automatizálása, az orvos-technikai alkalmazások jelentősége. Ezeknél az adatokat szolgáltató, ill. a beavatkozó műszerek, berendezések eltérnek a fizikai mérések során eddig használtaktól. Tömeg- és egyéb spektrométerek, különböző kromatográfok, mintaváltók stb. számítógéphez való illesztését kellett megoldani; az adatok feldolgozása, kezelése viszont nagyon hasonló a korábbiakhoz. Az ipari alkalmazások köre is jelentősen kiszélesedett. Egyrészt kisebb, komplex, programozható mérőműszereket használnak egyre kiterjedtebben az ipari üzemekben. Ilyenek például a motorgyártás során használt bemérő próbapadok, elektronikus készülékeket bemérő komplex automaták stb. Másrészt egyre több nagy kiterjedésű rendszert telepítenek, amelyek nagy területen elterülő ipari komplexumok különböző pontjain mérési adatokat gyűjtenek, ellenőrzést, naplózást és esetleg folyamatszabályozást végeznek. Ilyen komplexumok például az erőművek, hengerművek, kohók, folytonos technológiájú üzemek, gáz- és kőolaj-távvezetékek, ill. -tárolók és -elosztók.

A *technológiai lehetőségek* nagymértékben megnöttek a CAMAC kezdeti elterjedése óta. A CAMAC rendszer elsősorban TTL áramköri logikán alapszik: kezdetben főleg kis és közepes integráltságú áramkörökkel valósították meg a modulokat. A technológiai lehetőségek legfőbb változása a nagy integráltságú áramkörök megjelenése, mikroprocesszorok, félvezető tárolók tri-state logika, és a gyorsabb áramköri elemek terjedése.

A *rendszerszemlélet* is változott az elmúlt néhány évben. Régebben az erőforrásokat (feldolgozó, tárolóegységek) igyekeztek központosítani, mert így jobb hatásfokkal lehetett ezeket kihasználni. Újabban az osztott intelligencia fogalma egyre nagyobb tért hódít: ez azt jelenti, hogy minél több feladatot bíznak a központtól távolabb levő, helyi erőforrásokra, processzorokra, memóriákra és csak a valóban szükséges adatokat továbbítják a központi számítógépbe. Ily módon a feladatok jobban szét vannak osztva, áttekinthetőbbek, könnyebben módosíthatók, az egész rendszer megbízhatóbb, és ugyanakkor hatékonysága is nő. A hardware költségek látszólag nagyobbak, azonban ha a kábelezési, helyi szerelési költségeket is figyelembe vesszük, akkor az állomások megnövekedett processzor, tároló stb. költségeit a kábel- és szerelési költségek általában ellensúlyozzák.

A szabványosítási törekvés az utóbbi időben egyre erőteljesebb. Az eddigi mérési és folyamatszabályozási tapasztalatok megmutatták, hogy igen sok energiavész kárba szabványosítás hiányában. A hardware alkatrészek ára állandóan csökken, a mérnöki, tervezési költségek nőnek: az egyedi, „gazdaságosan” kialakított megoldások nagyon drágáknak bizonyultak.

Az IEC (International Electrotechnical Commission) keretében több Bizottság (Technical Committee) és munkacsoport foglalkozik különböző műszerezési és számítógépes interface-ekkel. A TC 45, a nukleáris műszerezéssel foglalkozó bizottság elfogadta a CAMAC szabványokat. A TC 66, az általános műszerezési bizottság eddig csak mérési összeállításban szereplő, kis számú mérőműszert összekötő szabványt dolgozott ki a HP cég interface-e alapján. A TC 65-ben, az ipari folyamatszabályozási bizottságban 1974-ben alakult új munkacsoport az interface-ek szabványosítására. A TC 57, az energiarendszerek távközlésével és elosztásával foglalkozó bizottság is nemrégben kezdett foglalkozni a szabványosítási kérdésekkel.

Az ISA (Instrument Society of America) és az IFIP (International Federation of Information Processing) védnökségével alakult meg 1973 végén az Ipari Számítógép Rendszerek Nemzetközi Purdue Munkacsoportja. (International Purdue Workshop on Industrial Computer Systems). Ennek a munkacsoportnak az a célja, hogy a számítógépes ipari alkalmazások szabványosítását előmozdítsa. Nyolc technikai bizottsága van:

- FORTRAN,
- megbízhatóság,
- hosszútávú folyamatszabályozási nyelv,
- interface és adatátvitel,
- problémaorientált nyelvek,
- ember—gép kapcsolat,
- megbízhatósági és biztonsági kérdések,
- glosszárium.

1.3.

CAMAC az általános fejlődés tükrében

Az, hogy valamely rendszer mi módon tudja összhangba hozni a nagy tapasztalatokat jelentő jelenlegi megoldásokat és tud alkalmazkodni a jövő fejlődéséhez, két tényezőről múlik. Az egyik az, hogy a rendszer alapjai mennyiben tartalmazzák a jövő megoldások csíráit, a másik pedig, hogy a rendszert alkalmazók, továbbfejlesztők milyen mértékben tudják a jövő igényeit figyelembe venni, ahhoz alkalmazkodni, azokkal kiegészíteni vagy beépíteni a rendszer alapjaiba. A két tényező természetesen erősen összefügg egymással.

1.3.1.

CAMAC és a korszerű hardware-software rendszerek

A real-time számítógépes rendszerek keletkezésük óta nagyon érdekes fejlődésen mentek át. A kezdeti egyedi megoldások tapasztalatai alapján általános megoldások alakultak ki: általános real-time perifériarendszer, általános vezérlők, általános magasabb rendű real-time és operációs rendszer (1.2. ábra). Az általános megoldások alkal-

mazásával magas szinten készültek az egyedi megoldások. Ez az irányzat még igen hosszú ideig sok területen megmaradt. Emellett most kezd kialakulni az a rendszer-technikai felfogás, hogy szükség van egy általános keretre, amely azonban funkcionális programokkal, vezérlőkkel, modulokkal bővíthető, kiegészíthető. Az általános keret egyrészt biztosítja az egyes elemek egymáshoz való kapcsolatát, másrészt megkönnyíti az egyes funkcionális elemek elkészítését. A real-time számítógépes rendszerek alkat-elemeinek változását mutatja, a fejlődési iránynak megfelelően az 1.3. ábra.



1.2. ábra
Real-time számítógépes rendszerek fejlődési iránya



1.3. ábra
Real-time számítógépes rendszerek alkat-elemeinek változása a fejlődési iránynak megfelelően

A CAMAC rendszer a fejlődés második fázisában alakult ki azzal a fő célkitűzéssel, hogy az általános magasrendű megoldásokat támogassa. Azonban már kezdettől fogva figyelemmel kísérték az igényeket, amelyek a harmadik fázis irányában mutattak, és az igények, ill. a technológiai lehetőségek fejlődésével több területen készültek kísérleti rendszerek: funkcionális, autonóm vezérlők, a mérési átalakítókhoz illeszkedő funkcionális modulok. Ezek a vizsgálódások egyértelműen azt mutatták ki, hogy a CAMAC rendszer által meghatározott keretek igen alkalmasak arra, hogy egy funkcionális modulokból felépülő rendszer általános kerete legyen.

1.3.2.

CAMAC az újabb alkalmazási területeken

Amikor a számítógépes mérési és szabályozási rendszerek új területeken is megjelennek, igen sok helyen kezdték el a CAMAC perifériarendszer alkalmazását. Az 1.4. ábra összefoglalja, milyen területeken készültek CAMAC rendszerek. A fizikai és kémiai laboratóriumokban az eredmények egyértelműen pozitívak: az eredeti szabványok elegendőek a rendszer kialakításához és a meglévő modulválasztékot sem nagyon kell kiegészíteni újabb modulokkal. Az orvosi-egészségügyi alkalmazásokban a rendszerek kipróbálása már nagyobb feladat. Egyrészt ezen a téren szigorú életvédelmi,

Fizikai laboratórium:	nukleáris elektronikus bemérő aerodinamikai csillagászati izotóp oktatás
Kémiai laboratórium:	kromatográfia spektrométer spektroszkópia
Orvosi-egészségügyi:	intenzív ápolás nukleáris medicina EKG, EEG klinikai laboratórium
Ipari mérés, szabályozás:	elosztó hálózatok mérőautomaták próbabadok vastagságszabályozás erőművi adatgyűjtés, ellenőrzés hengermű gyorsító szabályozás

1.4. ábra

A CAMAC rendszerek főbb alkalmazási területei

megbízhatósági szabványokat is ki kell elégíteni, másrészt pedig csak viszonylag nagyobb rendszerek kialakítása gazdaságos, így ezek üzembe állítása hosszabb ideig tart. Az eddigi eredmények alapján valószínű, hogy ezen a téren is a CAMAC rendszerek lesznek a legelterjedtebbek.

Az ipari alkalmazások kérdése a legnehezebb. Az ipari környezet mindenféle szempontból sokkal kellemetlenebb, mint a laboratóriumi vagy akár az orvosi. A mechanikus és egyéb környezeti behatások (hőmérséklet, nedvesség, elektromos zajok stb.), a kezelőszemélyzettel szembeni elvárások, a megbízhatóság fontossága mind arra mutatnak, hogy nagyobb körültekintésre van szükség. A több, kisebb ipari alkalmazásokon túlmenően néhány cég az Egyesült Államokban többéves mód-

szerez vizsgálatot végzett a CAMAC ipari bevezetésével kapcsolatban [3], [4]. Az eredmények a következők:

1. A CAMAC szabványok alkalmasak arra, hogy ipari real-time perifériák készítéséhez alapul szolgáljanak.

2. Az eddigi modulkészítési gyakorlathoz képest néhány további részletkérdést szabványosítani kell: ezek elsősorban a csatlakozókkal, ill. a földelési rendszerrel kapcsolatosak [1], [2].

A kezdeti vizsgálatok után 1974-ben nagyarányú kísérleti alkalmazásra került sor, egy hengermű automatizálása kapcsán. Ebben a rendszerben kétszáz körüli modul működik együtt. A rendszer végleges értékelése 1975-ben várható.

1.3.3.

Szervezeti változások

Mivel az utóbbi években a CAMAC alkalmazások jelentősen túlnőttek a nukleáris alkalmazásokon, mind Európában, mind az Egyesült Államokban újabb szervezetek alakultak a CAMAC alkalmazások bővítésére. E célból 1974-ben megalakult az Európai CAMAC Szövetség (European CAMAC Association — ECA), amely független, de szorosan együttműködik az ESONE-val. Három fő új alkalmazási területet tartanak fontosnak: ipari, orvosi és közszolgáltatási alkalmazásokat. Ezekre a területekre a jelenlegi CAMAC szabványokat elfogadva újabb ajánlásokat, esetleg szabványokat dolgoznak ki. Jelenleg az ECA három munkacsoporttal működik: ipari alkalmazások, orvosi alkalmazások, publikációs tevékenység.

Az Egyesült Államokban létrejött a CAMAC Ipari Alkalmazási Csoport (CAMAC Industrial Applications Group — CIAG), amely az ipari alkalmazásokkal kapcsolatos problémákkal foglalkozik, természetesen együttműködve az európai csoportokkal.

Mind az ESONE, mind pedig az újabb CAMAC szervezetek személyes és hivatalos kapcsolatban vannak a többi szabványosító szervekkel (IEC, Purdue Munkacsoport).

1.4.

CAMAC a KGST-ben

Az ESONE Bizottságnak több KGST-ország laboratóriuma, ill. intézete is tagja. Ezek a következők:

- Egyesített Atomkutató Intézet, *Dubna* (Szovjetunió),
- Központi Elektrofizikai Intézet, *Berlin* (DDR),
- Badan Jadrowich Intézet, *Swierk* (Lengyelország),
- Atomfizikai Intézet, *Bukarest* (Románia),
- Központi Fizikai Kutató Intézet, *Budapest*.

Ezekben az országokban — elsősorban nukleáris és egyéb laboratóriumi célra — körülbelül 150 különböző modult fejlesztettek ki és gyártanak. 1974 végéig kb. 50 CAMAC rendszert állítottak üzembe.

Az eredeti CAMAC szabványok mellett a Szovjetunióban, a CAMAC-tól lényegében csak mechanikai méreteiben különböző, VEKTOR rendszert dolgozták ki. A KGST atomenergia békés felhasználásával foglalkozó állandó bizottságának (AÁB) nukleáris

műszeripari csoportja mind az eredeti CAMAC mechanikára, mind pedig a VEKTOR mechanikára kidolgozott ajánlásokat.

A nukleáristól eltérő alkalmazási területek főleg Lengyelországban, az NDK-ban és Magyarországon terjednek elsősorban. A szocialista országok 1974-ben alakított Mini Számítástechnikai Rendszere a real-time perifériák szabványosítását is kitűzte céljául. E szabványok kidolgozásakor különösen figyelembe veszik a CAMAC rendszereket.

Irodalom az 1. fejezethez

- [1] ALCOA Engineering Specifications for CAMAC; Aluminium Company of America, 1974.
- [2] Analysis of CAMAC Grounding and Shielding Practices; Working paper of the NIM—Dataway Working Group.
- [3] Lyon, W. T.: An Evaluation of CAMAC Equipment in an Industrial Environment, Proc. of the First Int. Symp. on CAMAC, p. 211. 1973.
- [4] Lyon, W. T.: Let's Standardize Our Computer Problems. IEEE Industrial Applications Society Annual Meeting, Oct. 10., 1973.

2.

Általános elvek, CAMAC előírások

2.1.

A CAMAC rendszer főbb jellemzői

2.1.1.

CAMAC egységek

A CAMAC rendszer legjellegzetesebb része a keret, a keretbe bedugható modulokkal, keretvezérlővel.

CAMAC modul

A rendszer legkisebb egysége a CAMAC modul. Egy-egy modul valamilyen önálló real-time funkciót valósít meg. A modulok lehetnek: soros, párhuzamos, kimeneti, bemeneti regiszterek, megjelenítő egységek, analóg—digitális és digitális—analóg átalakítók, órajel-, kapujel-generátorok, multiplexerek stb. Szabványosítva vannak a modulok mechanikai méretei és elektromos csatlakoztatása. A modulok magassága és hossza állandó, szélessége előírt egységekben növelhető (2.1. ábra).

CAMAC keret, keretvezérlő

A keret a modulok befogadására alkalmas váz, szélessége a 19"-os, nemzetközileg elfogadott rack rendszerhez igazodik. Egy keretben egymás mellett 25 db egységnyi széles modult lehet elhelyezni. A keret jobb oldalán a kettő vagy három egység szélességű modulként kialakított keretvezérlőt találjuk (lásd 2.2. ábra). A keretvezérlő bonyolítja az egy keretben levő modulok jelforgalmát a keret közepén, a modulok mögött végig futó 86-vezetékes szinkron buszon, az ADATÚT-on keresztül, amelyhez a modulok behelyezésükkel elektromosan csatlakoznak. A keretvezérlő lehet önálló működésű (autonom), csatlakozhat számítógéphez vagy további keretek keretvezérlőjéhez.

2.1.2.

CAMAC rendszerek, FŐUTAK (Highway-k)

Egykeretes CAMAC rendszerek

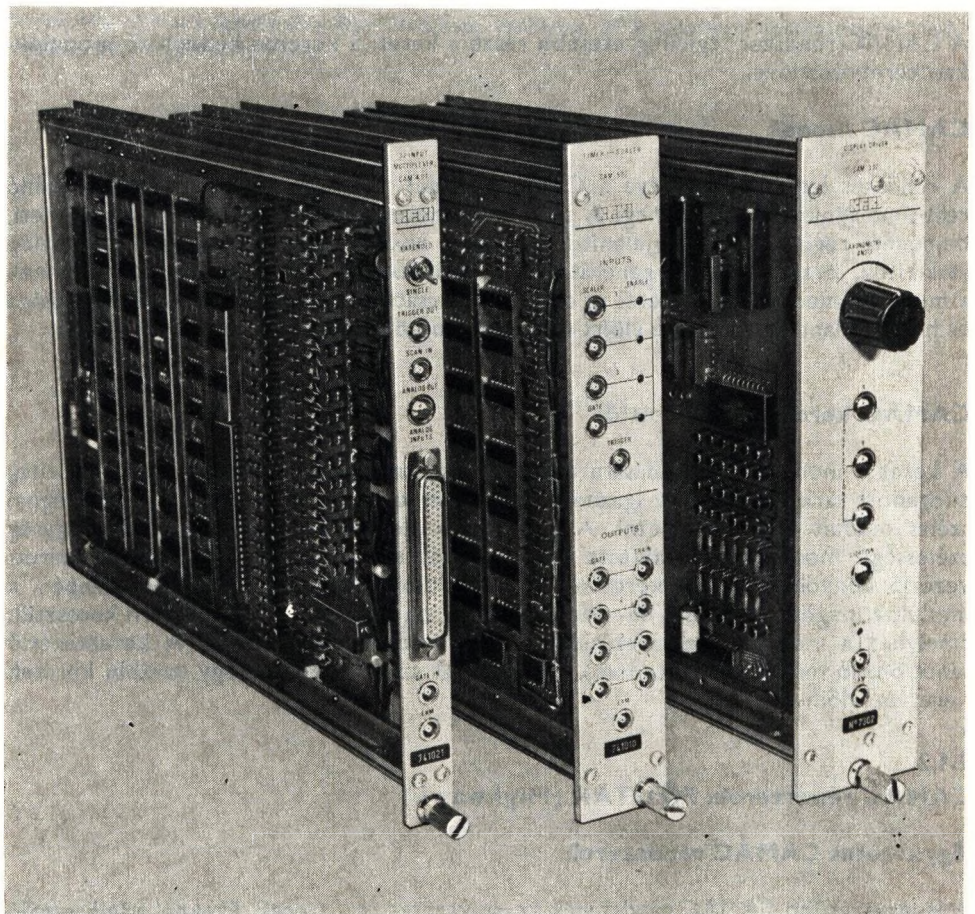
A legegyszerűbb CAMAC rendszerek az egykeretes rendszerek. Ezeknél a felhasználó által igényelt funkciókat megvalósító CAMAC modulok elférnek egyetlen keretben. A keret vezérlését, így egyben az egykeretes CAMAC rendszer vezérlését, a keret-

vezérlő önállóan vagy számítógéphez csatlakozva végzi. Az önálló működésű keretvezérlő egyszerűbb esetben, ahol a feladat végrehajtásának egyes lépései előre meghatározhatók (pl. ciklikus lekérdezés), a soronlevő utasítást valamilyen fix tárból veszi [8], és nem igényel számítógépes csatlakozást.

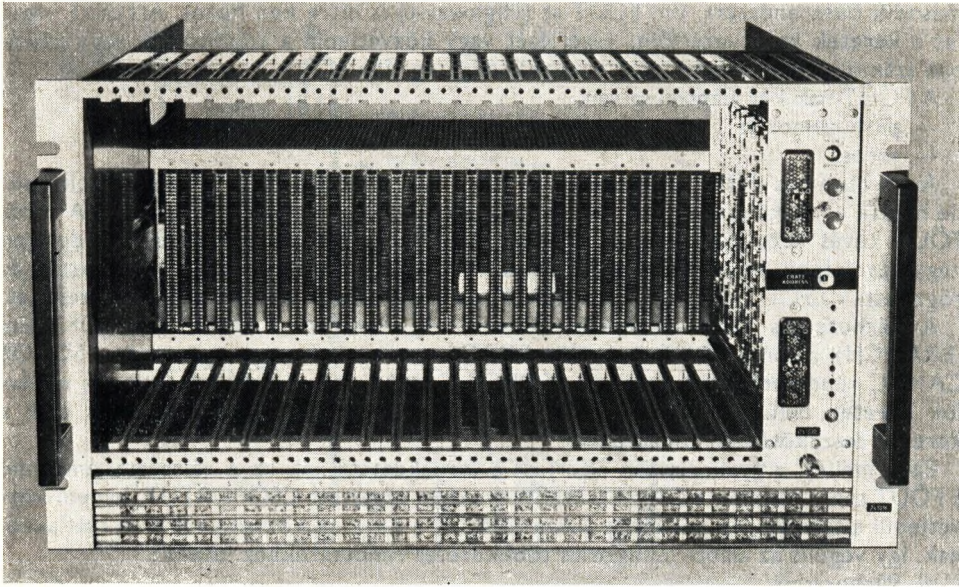
Ha a feladat bonyolultabb és programvezérlésre van szükség, az egykeretes rendszert vagy olyan keretvezérlővel kell ellátni, amely önmagában „intelligens”, rendelkezik valamilyen CPU-val és operatív memóriával [9], vagy olyan keretvezérlővel, amely kapcsolatot teremt valamilyen kisgép felé. Ez utóbbi esetben a keretvezérlő illesztőegységként szerepel a CAMAC ADATÚT és a szóbanforgó kisgép perifériás busza között. Az irodalomban az ilyen vezérlőket gépreszabott (dedicated) vagy — mivel ezt külön szabványok nem rögzítik — U típusú (undefined) vezérlőnek nevezik.

Szinte valamennyi ismertebb kisgéphez készült ilyen U típusú CAMAC keretvezérlő [7]. A 2.3. ábrán az egykeretes rendszer két változatát látjuk.

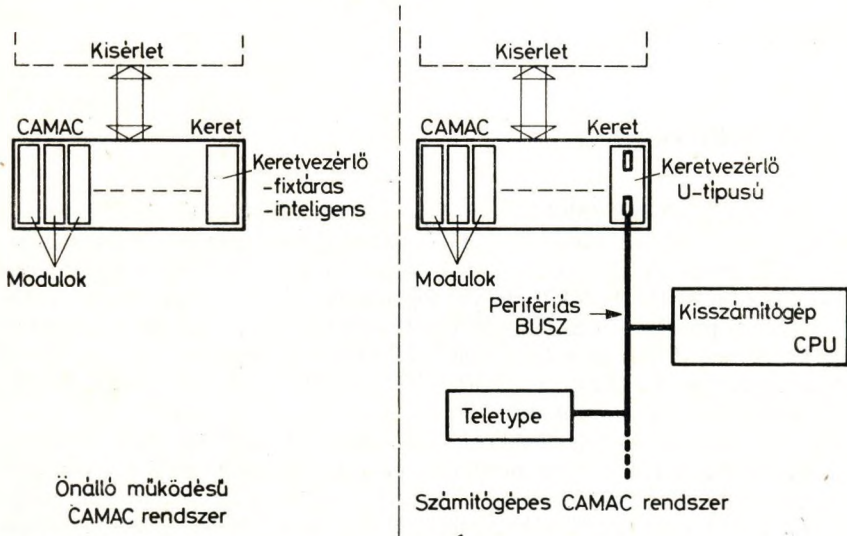
Az egykeretes rendszerek jelentősége igen nagy, az eddig megvalósított CAMAC rendszerek 2/3-a ilyen egykeretes felépítésű.



2.1. ábra
CAMAC modulok



2.2. ábra
CAMAC keret keretvezérlővel



2.3. ábra
Egykeretes CAMAC rendszerek

Többkeretes CAMAC rendszerek

Ha a felhasználó által igényelt funkciókat megvalósító modulok: a felhasználói modulok száma nagy, és így már nem helyezhetők el egyetlen keretben, a rendszert bővíteni kell újabb keretek bevonásával. A keretek csatlakoztatására ebben az esetben egy

második buszrendszert, ún. FŐÚT-at (Highway-t) is létre kell hozni. A FŐÚT végig fut a keretek keretvezérlőin, vezérlését vagy közvetlenül a számítógép, vagy általában számítógéppel összekapcsolt meghajtóegység látja el.

A FŐÚT-nak két alaptípusa van.

- párhuzamos és
- soros.

A párhuzamos FŐÚT sokvezetékes, nagy információáramlási sebességet biztosít, de hossza — elektromos zavar és gazdasági okokból is — erősen korlátozott. A soros FŐÚT kevés vezeték, esetleg egyetlen érpárt használ, így lassabb ugyan, de igen nagy távolságok áthidalására is alkalmas. A CAMAC szabványokban kidolgozták és rögzítették a FŐÚT mindkét változatát, ill. az ezekre épülő többkeretes rendszereket.

A párhuzamos FŐÚT-ra épülő többkeretes rendszert CAMAC CSOPORT-nak (BRANCH), a soros FŐÚT-ra épült SOROS szervezésű, vagy egyszerűen SOROS CAMAC rendszernek nevezték el. (SOROS CAMAC-nál a keret érintetlen, a modulok kereten belüli jelforgalma továbbra is párhuzamos, a soros megjelölés csak a keretek összekötésére vonatkozik!)

Ezek mellett a gyakorlatban elterjedt a csatlakoztatásnak az a változata, amelynél a FŐÚT szerepét nem a CAMAC szabványokban rögzített CSOPORTÚT, hanem közvetlenül a számítógép perifériás busza tölti be, a keretek közvetlenül erre csatlakoznak. Így végülis az alábbi jellegzetes többkeretes rendszerekhez jutunk:

a) Párhuzamos:

- szabványos CAMAC CSOPORT (BRANCH).
- EGYEDI, gépre szabott CSOPORT.

b) Soros:

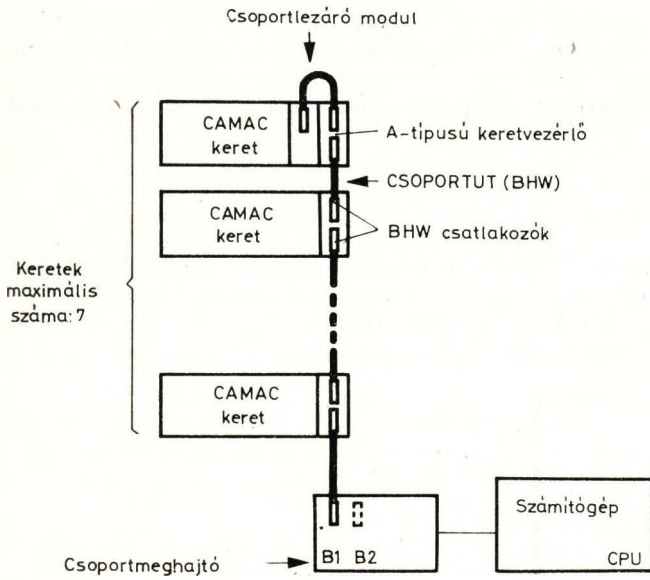
- SOROS CAMAC rendszer.

Párhuzamos CAMAC rendszer

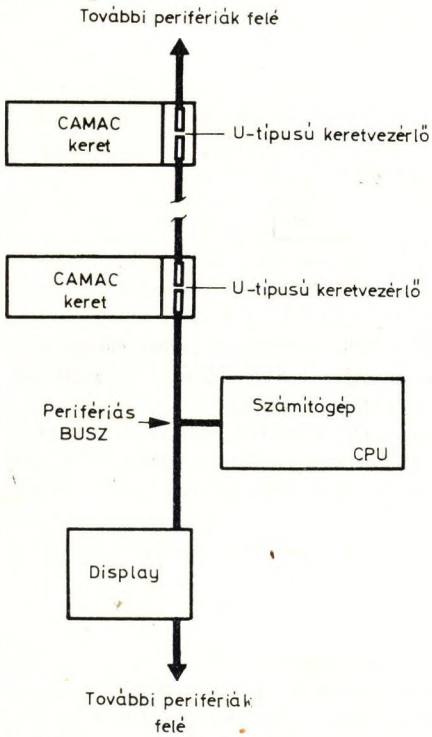
Szabványos CAMAC CSOPORT (BRANCH): Egy CAMAC BRANCH maximálisan 7 keretet tartalmazhat. Minden keret az ún. A típusú keretvezérlővel van ellátva, amelyen két 132-pontos csatlakozót találunk a párhuzamos FŐÚT, a 66-érpáros CSOPORTÚT (BRANCH HIGHWAY, BHW) fogadására, ill. továbbadására. A CSOPORTÚT kiindulási pontja a CSOPORTMEGHAJTÓ-nál van, megszakítás nélkül végighalad a keretvezérlőkön és a CSOPORTLEZÁRÓ-ban végződik. (l. 2.4. ábra). A rendszer bővíthető azáltal, hogy a CSOPORTMEGHAJTÓ további CSOPORTOK (B2, ...) vezérlésére is alkalmassá tehető.

Egyedi, gépre szabott többkeretes CAMAC rendszerek: A CAMAC keretek ebben az esetben közvetlenül a számítógép perifériás buszára, ki-, ill. bemeneti vonalára csatlakoznak, ugyanúgy mint a gép egyéb perifériái (pl. display, lyukkártya-, lyukszalag-olvasó, l. 2.5. ábra). Az alkalmazott keretvezérlő U típusú, a CAMAC szabványokban nincs rögzítve, így a keretek maximális száma nincs előírva. Ez az adott kisgép perifériakezelő adottságaitól, a keretvezérlő tervezőjétől függ. A rendszer előnye, hogy elmarad egy külön CAMAC meghajtóegység, így — különösen kevés keret esetén — olcsó. Hátránya, hogy a rendszer felhasználója elesik a szabványos többkeretes rendszerek nyújtotta többlet lehetőségektől.

A szabványos többkeretes rendszerek egy magasabb fokú CAMAC szervezési formát jelentenek, csak az ilyen rendszerben végrehajtható utasításokkal, műveletekkel stb., míg az egyedi rendszerben a felhasználónak egyszerűen csak több kerete



2.4. ábra
CAMAC CSOPORT

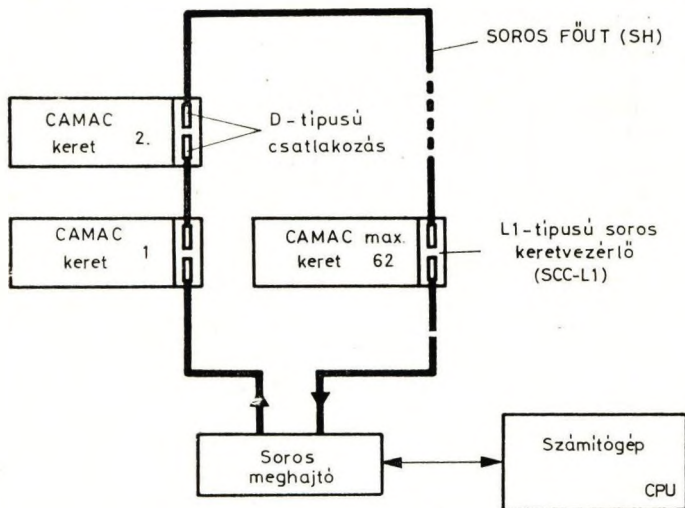


2.5. ábra
Egyedi, gépre szabott többkeretes CAMAC rendszer

van. A CAMAC kompatibilitás csak a keretre terjed ki, a rendszerre már nem. Ilyen megoldás mégis előnyösen használható pl. abban az esetben, ha a gépre több, egymástól lényegében független, egyszerűbb mérést kell kapcsolni, és így a gép szempontjából a CAMAC keretek rendszerként nem jelentkeznek, csak mint az egyes mérések külön keretei.

SOROS CAMAC rendszer

A SOROS összekötésű többkeretes CAMAC rendszerben a keretek a SOROS FŐÚT-ra csatlakoznak. A bit-soros vagy byte-soros FŐÚT egyirányú hurkot képez, melynek kiindulási pontja a számítógépnél levő SOROSMEGHAJTÓ, végighalad a soros keretvezérlőkön és visszatér a SOROSMEGHAJTÓ-hoz (2.6. ábra).



2.6. ábra
SOROS CAMAC rendszer

A soros hurokra egyidejűleg maximálisan 62 CAMAC keretet lehet rákapcsolni. A szabványosított soros keretvezérlő L1 típusú, amelyen a SOROS FŐÚT fogadására, ill. meghajtására egy-egy 25 pólusú csatlakozót találunk (D-port).

Ha sebességben lassabb is a CSOPORT szervezésnél, a SOROS rendszer előnyösen alkalmazható nagy területen elosztva elhelyezett keretek összekapcsolására. A soros hurok egyes szakaszait bármilyen távadatátviteli vonallal (távbeszélő, rádió, mikrohullám stb.) is megvalósíthatjuk.

2.1.3.

Üzenetformátumok, jelforgalom

A CAMAC egységek, modulok (hardware) működtetését a különböző buszokon, főúton közlekedő CAMAC utasítások, üzenetek végzik. Egy összetett rendszerben meg kell adni, hogy melyik keretet (C) kívánjuk megszólítani. A kereten belül ki kell jelölni az állomáshelyet (N), ahová a modult bedugtuk és meg kell adni, hogy a

modul mely részével (A) akarunk kapcsolatba lépni, és ott milyen típusú funkciót (F), műveletet (olvasás, írás, vizsgálat, vezérlés stb.) akarunk végrehajtani. Az üzenet adatrészének (R/W-olvasás/írás) szóhossza 24 bit. Tartalmazza az üzenet állapot-információt, mint amilyen a nyugtázójel (X), ill. válaszjel (Q). Ha egy CAMAC modul fel akarja hívni magára a figyelmet (Look-at-Me LAM), L jelet generál. A modulok L jeleit a keretvezérlők egyesített igénykérés (D) formájában továbbítják.

Vannak olyan közös, címfüggetlen vezérlőjelek, amelyek adott műveletet, hatást az egész rendszerben egyszerre hajtanak végre, ill. váltanak ki (Z — alaphelyzet, C — törlés, I — tiltás). Tehát egy teljes CAMAC üzenet — kb. 50 bit — összetevői:

CÍM	FUNKCIÓ	ADAT	ÁLLAPOT	VEZÉRLÉS
C, N, A	F	R/W	X, Q, L, D	Z, C, I

Az ADATÚT-on és a CSOPORTÚT-on lehetőség van a teljes CAMAC üzenet egyidejű, párhuzamos, egy CAMAC ciklus alatti továbbítására a válasz, ill. nyugtázójel átvételével együtt. Ugyanez a SOROS FŐÚT-on 13 byte-időt igényel.

Sebességek:

ADATÚT	1 μ s ciklusidő (szinkron időzítés),
CSOPORTÚT	1,6...2 μ s ciklusidő (hand-shake időzítés),
SOROS FŐÚT	5 MHz-es byte frekvencia esetén egy teljes CAMAC üzenet (13 byte-idő) kb. 3 μ s.

(A CSOPORTÚT megadott ciklusideje magában foglalja az ADATÚT ciklusidejét, de nem tartalmazza a CSOPORTÚT-kábel hosszának megfelelő késleltetést. A soros rendszer ilyen sebességű üzemeltetése csak külön e célra kiépített kábellel lehetséges. Ha soros hurok egyes részei hagyományos távadatátviteli vonalat stb. vesznek igénybe, a megengedett sebességet ezek szabják meg, és ez nagyságrendekkel a megadott érték alatt lehet.)

A CAMAC üzenet nagy bitszáma miatt a CAMAC buszok nagy működési sebességét nehéz kihasználni, mert a csatlakozó számítógépnek egyszerű CAMAC vezérlők esetén 2...4 számítógépes utasítására van szüksége egy CAMAC üzenet összeállításához. Ha ehhez hozzávesszük a válasz értelmezését, az adattárolást, a rendszer működési sebessége nagyságrenddel csökken, 1...3 μ s helyett 10...30 μ s-os eredő ciklusidő adódik.

2.2.

CAMAC rendszer részletes ismertetése

2.2.1.

A szabványokat tartalmazó kiadványok áttekintése

A CAMAC rendszer részletes specifikációját, ill. az ehhez kapcsolódó kiegészítő, értelmező anyagokat a következő kiadványokban találjuk [1]...[6], [10].

EUR 4100 e (EURATOM, Ispra 1972.)

„CAMAC — Moduláris adatkezelő műszerezési rendszer”.

(„CAMAC — A Modular Instrumentation System for Data Handling”.)

A kiadványokban rögzítésre kerülnek:

- A CAMAC rendszer alapvető tulajdonságai.
- A CAMAC mechanika (keret, modul), csatlakozók.
- Az ADATÚT (DATAWAY, DW) vonalai, utasításrendszere, jelszintjei, időzítése.
- A tápegység paraméterei.

A kiadványnak volt egy korábbi, 1969-es kiadása is. Az új kiadásban található módosításokat az első CAMAC rendszerekkel szerzett tapasztalatok alapján a nagyobb megbízhatóság, a korábban definiálatlan paraméterek pontosítása érdekében kellett bevezetni. Az 1972-es új kiadás ezeket az eltéréseket külön jelöli.

1975-ben ezt a szabványt az IEEE is elfogadta és a következő címen adta ki: IEEE Std 583-1975 „IEEE Standard Modular Instrumentation on Digital Interface System (CAMAC)”.

EUR 4600 e (EURATOM, Ispra 1972.)

„CAMAC — Többkeretes rendszerek szervezése”.

(„CAMAC — Organisation of Multi-Crate Systems”.)

A kiadvány definiálja a többkeretes CAMAC rendszer felépítését, ezen belül:

- A CAMAC CSOPORT (BRANCH) fogalmát.
- A CSOPORTÚT (BRANCH HIGHWAY, BHW) vonalait, utasításrendszerét, időzítését, jelszintjeit.
- A CSOPORTÚT csatlakoztatási pontjait.
- Az „A” típusú CAMAC keretvezérlőt (CCA, CCA—1).

TID—25877 (US. AEC—NIM kiadvány, 1972.)

„Kiegészítés a CAMAC műszerezési rendszerspecifikációkhoz”.

(„Supplement to CAMAC Instrumentation System Specifications”.)

A kiadvány a CAMAC rendszert leíró két alapidokumentum, az EUR 4100 e és EUR 4600 e értelmezésére, kiegészítésére, pontosítására készült és útmutatást ad egyes feladatok áramköri és konstrukciós problémáinak megoldására.

Témakörök:

- Kompatibilitási kérdések.
- Sokpontos csatlakozók pontkiosztása.
- LAM megvalósítási példák.
- A CSOPORTÚT (BHW) hosszának korlátai.
- Tipikus CAMAC tápegység.
- A CAMAC specifikáció 1972-es módosításának elemzése.

Ez a kiadvány a 6-os számú CAMAC BULLETIN mellékleteként is megjelent.

EUR 5100 e (EURATOM, 1974.)

„CAMAC — Analóg jelek amplitúdóspecifikációja”.
(CAMAC — Specification of Amplitude Analogue Signals”).)

A szabvány CAMAC modulok által generált vagy fogadott analóg jelekre vonatkozik. Meghatározza az analóg jelek osztályait, majd ennek megfelelően

- kimeneti paramétereiket,
- bemeneti paramétereiket,
- csatlakozókat.

ír elő.

Jelenleg kidolgozás alatt van az ipari felhasználású modulok, ill. laboratóriumi szimmetrikus jelek csatlakozási paramétereinek szabványa.

TID—26488 (US. AEC—NIM kiadvány, 1973.)

„CAMAC — Soros rendszer szervezése”.
(„CAMAC — Serial System Organization”).)

Ebben az anyagban a többkeretes CAMAC rendszer soros összekapcsolásának szabályait találjuk, ezen belül:

- A soros CAMAC topológiáját.
- Üzenetformátumait.
- Jelátviteli, időzítési feltételeit.
- L igény kezelését.
- Az L1 típusú soros keretvezérlő (SCC—L1) leírását.

Az anyag még nem tekinthető véglegesnek, az ESONE bizottság véleményeltérését az ESONE (SH) 02, 03 jelű kiadványaiban foglalta össze. A végleges anyag 1975. elejére várható.

ESONE (IML) 01 (ESONE kiadvány, 1974. október)

CAMAC — Az IML definíciója.
(„CAMAC — The Definition of IML”).)

Az IML a CAMAC rendszerek legalacsonyabb szintű szabványosított programozási nyelve, a CAMAC egységek megszólítását, egységes kezelését biztosítja. Az IML nem önálló nyelv, valamilyen vendéglátó nyelvbe, a vezérlő számítógépen futó magasabb szintű nyelvbe (pl. Assembler, FORTRAN, BASIC) kell beépíteni.

A szabvány főbb fejezetei:

- Az IML nyelv általános leírása.
- Alapvető definíciók (konstansok, változók, array-k stb.).
- Címparaméterek.
- Működési módok.
- Állítás-típusok (Statementek).

Függelékben az: IML—M1 makró szintakszisa.

Magasabb szintű CAMAC nyelv kidolgozása is elkezdődött és az első elképzelések a CAMAC BULLETIN 6-os számának mellékletében jelentek meg. Az ezzel kapcsolatos tevékenység azonban jelenleg csaknem teljesen abbamaradt, mivel a nemzetközi szinten megindult real-time nyelv egységesítési munkák igen jelentősek és egy ettől eltérő CAMAC nyelv kidolgozása nem látszik célszerűnek.

KGST szabványtervezetek

Az atomenergia békés célú felhasználásával foglalkozó állandó bizottság (AÁB) nukleáris műszeripari csoportja dolgozta ki a CAMAC rendszernek pontosan megfelelő, ill. ettől lényegében csak mechanikai méreteiben különböző szovjet VEKTOR rendszerre vonatkozó KGST szabványok tervezetét. Ezek a következők:

- 160—1—73: CAMAC ADATÚT, tápegység,
- 160—2—73: CAMAC mechanika,
- 160—3—73: VEKTOR mechanika („CSERESZNYE”).

A további CAMAC szabványok (többkeretes rendszer, software) KGST megfelelőjének kidolgozása folyamatban van.

2.2.2.

CAMAC mechanika

A CAMAC rendszer mechanikai felépítésére vonatkozó szabvány (EUR 4100 e) két alapegységet határoz meg, a CAMAC keretet és a CAMAC modult.

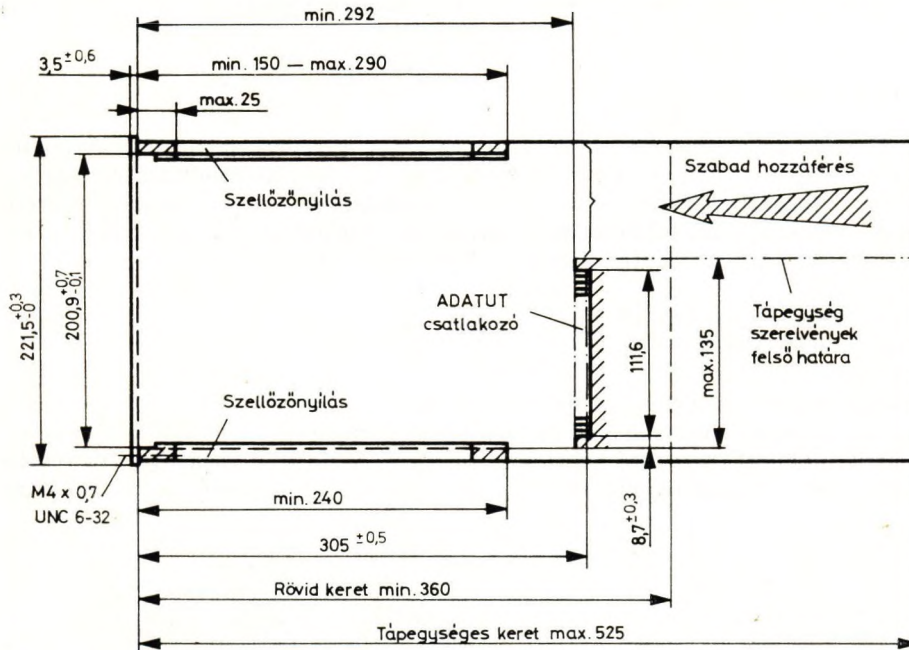
CAMAC keret

A CAMAC keret a rendszer legjellegzetesebb egysége, a modulokat befogadó, rögzítő váz, kialakítását tekintve egy 19" széles, 5 U (kerekítve 222 mm) magas (1 U=44,45 mm), 525 mm mély fiók. A fiók nyitott eleje felől lehet a dugaszolható egységként kiképzett CAMAC modulokat a fiókba, keretbe behelyezni. A modulok helyzete függőleges, a megvezetést a keretben alul és felül 17,2 mm-es osztással elhelyezkedő vezetősínek végzik. Egy keretben egymás mellett 25 db egységnyi (17,2 mm) széles CAMAC modul lehet rögzíteni. A keretben a modulhelyek előlről nézve, balról jobbra haladva 1-től 25-ig számozva vannak. A modulhelyeket állomásoknak, a modulhelyek sorszámát ÁLLOMÁSSZÁM (N)-nak nevezik. A keret elején, az alsó keresztlécen menetes furatokat találunk (M4), ahol a behelyezett modulok csavarral rögzíthetők. A keret külső méretei úgy vannak megválasztva, hogy a nemzetközileg elfogadott 19"-os állványrendszerbe illeszkedjenek, eleget téve az IEC idevonatkozó előírásának is.

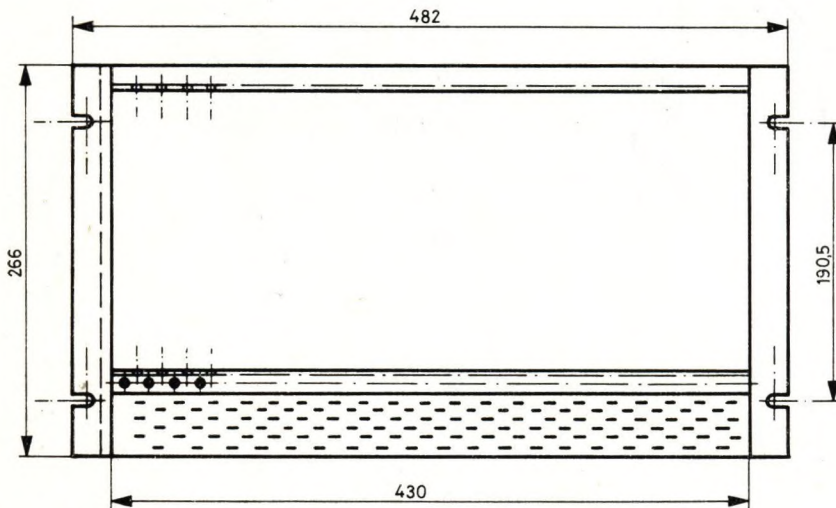
A keretben kerül rögzítésre a modulok adatforgalmát, vezérlését biztosító ADATÚT (DATAWAY, DW), és a modulok tápfeszültség-ellátását biztosító CAMAC tápegység. Az ADATÚT az állomásoként elhelyezett 86 (2×43) pontos, lényegében buszhuzalozással ellátott csatlakozósort jelent. A csatlakozósor (ADATÚT) a modulok mögött fut végig az egyes állomásokon, olyan mélységben, hogy a modul hátoldalán kiálló, általában a modul nyomtatott áramköri kártyájából kialakított direkt csatlakozó a modul behelyezése, csavaros behúzatása után az aktuális állomáshoz tartozó ADATÚT csatlakozót biztonsággal elérje. A CAMAC tápegység a keretben az ADATÚT mögötti térben helyezkedik el, az ADATÚT és a keret hátsó síkja között. A tápegység magasságát korlátozza, hogy a modulok hátoldalának az ADATÚT feletti részéhez vezetett kábelek részére szabad teret kell biztosítani, így a tápegység magassága nem emelkedhet az ADATÚT szerelvényszintje fölé (2.7. ábra).

Egy keretben ventiláció nélkül maximálisan megengedett disszipáció 200 W, azaz állomásoként 8 W. A keret vezetősín-rendszerét és a modulok alsó, felső profiljait

úgy kell kiképezni, hogy a termelődő meleg elvezetésére állomáshelyenként 15 cm^2 szabad légáramlási felület legyen biztosítva. Ha az összdisszipáció a 200 W -ot meghaladja, ventilátoros hűtést kell alkalmazni. Ez esetben a keret alul kiegészül egy 1 U magas ventilációs térrel és a keret teljes magassága 6 U lesz (l. 2.8. ábra).



2.7. ábra
CAMAC keret oldalnézete (metszet)



2.8. ábra
Ventilátoros CAMAC keret előlnézete

A CAMAC keret alkalmas az US. AEC—NIM bizottsága által korábban kidolgozott NIM modulok befogadására is. Ennek osztása 34,4 mm, így egy NIM modul két CAMAC állomáshelyet igényel. Tekintettel arra, hogy a NIM modul rövidebb, az ADATÚT csatlakozó és a NIM modul eltérő hátsó csatlakozója közti csatlakoztatás egy közbe iktatott NIM/CAMAC adapterrel megoldható. A NIM modulok rögzítésére az M4-es furatok mellett külön UNC 6—32 típusú furatokat lehet elhelyezni a CAMAC keret alsó és felső keresztmerezítőiben. A CAMAC tápegység opcióként a NIM feszültségeket is tartalmazhatja, így a NIM modulok tápfeszültség-ellátása is biztosítható.

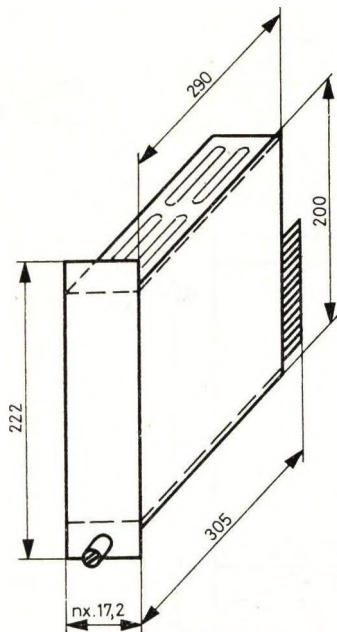
CAMAC modul

A CAMAC modul dugaszolható (Plug-in) egységként szabványosították. A modul magassága: 222 mm, hossza: 305 mm, szélessége a modul bonyolultságától függően változhat, a legkeskenyebb modul 17,2 mm. A szélesebb modul szélességi mérete 17,2 mm egész számú (n) többszöröse lehet, ennek megfelelően beszélünk két egység széles, három egység széles stb. modulról. A modul részei:

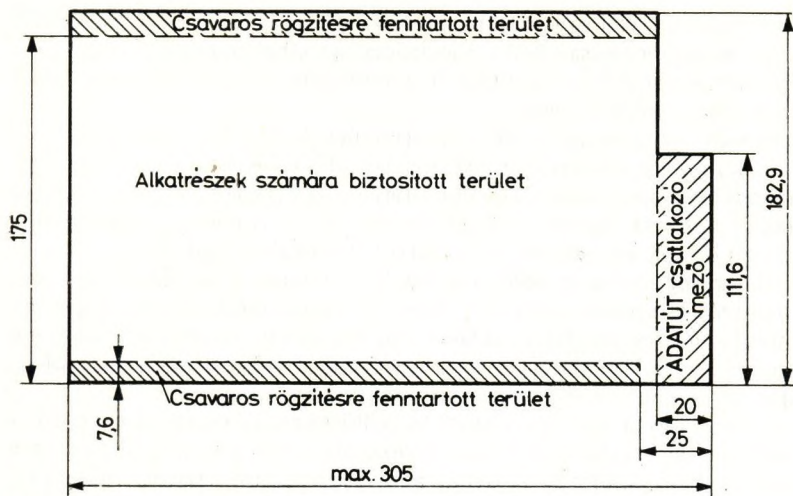
- előlap,
- alsó—felső merevítőprofil,
- nyomtatott áramkörti kártya,
- hátlap.

Egy modul főbb méreteit a 2.9. ábrán mutatjuk be.

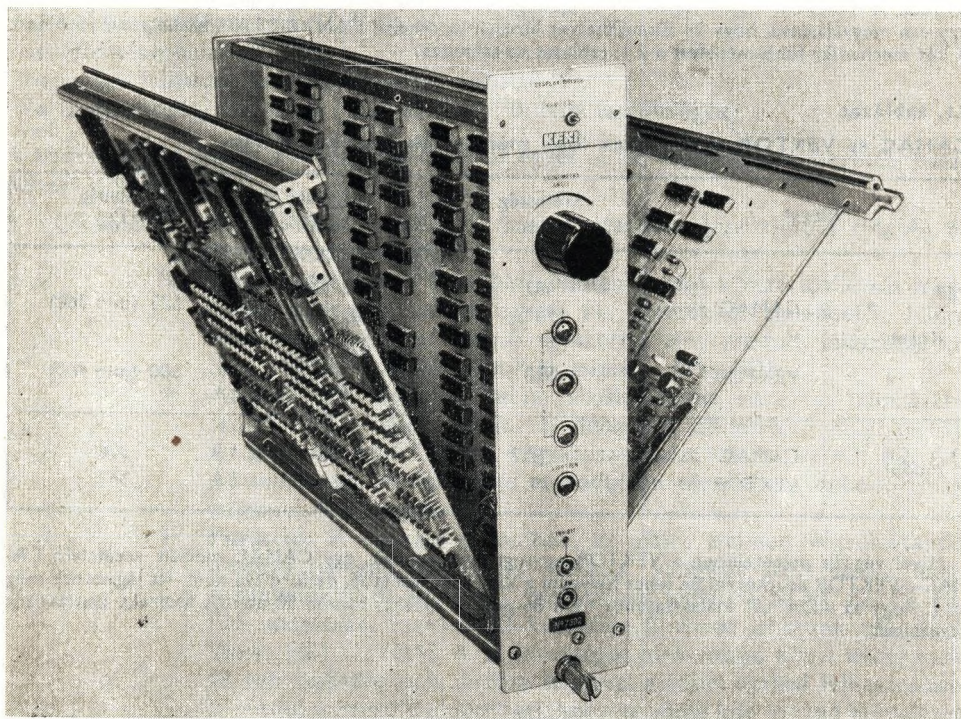
A modul *előlapján* helyezkednek el a modul kezelőszervei, a modult a külvilággal vagy másik modullal összekötő kábelek csatlakozói. A modul előlapjának alsó részén kell elhelyezni a modul rögzítőcsavarját. Ha a modul több egység széles, akkor az



2.9. ábra
Egy n egység széles
CAMAC modul főbb méretei



2.10. ábra
Nyomtatott áramköri kártya



2.11. ábra
A modul kártyáinak széthajtása

ADATÚT-ba csatlakozó kártyával egy vonalba, ha több kártya is csatlakozik az ADATÚT-ba, akkor ezekre szimmetrikusan kell a rögzítőcsavart elhelyezni.

Az alsó és felső merevítőprofilhoz rögzítjük a nyomtatott kártyát, ill. ezek kiálló pereme biztosítja a modul megvezetését.

A nyomtatott áramköri kártyán, melynek ajánlott méretét a 2.10. ábrán láthatjuk, helyezkednek el a különféle elektromos alkatrészek, IC-k. A kártyának általában integrált része a modul hátlapján zászló szerűen túlnyúló csatlakozó felülete, a kártya két oldalán kialakított 2×43 pontos érintkezőmezővel. A szabvány megengedi a megfelelően felszerelt 2×43 pontos indirekt csatlakozó alkalmazását is.

Ha a modul több egység széles és több kártyát is tartalmaz, a kártyákat úgy kell rögzíteni, hogy bemérés és szerviz célra a kártyák egymástól széthajthatók legyenek (2.11. ábra). A modul hátlapján megfelelő ablakot kell biztosítani a modulból hátrafelé kinyúló érintkezősáv részére. Az ADATÚT által nem fedett felső részen csatlakozókat, speciális kezelőszerveket lehet elhelyezni.

A modul két oldalán vagy a kártyák között is, különösen az elektromos zavarójelekre érzékenyebb analóg modulok esetén, árnyékolólemez is szerelhető. A készre szerelt modul alsó és felső felületén elosztva annyszor 15 cm^2 szabad légáramlási keresztmetszetet kell biztosítani, ahány egység széles a modul.

A CAMAC rendszerhez hasonló felépítésű a VEKTOR rendszer, melynek rendszertechnikai felépítése kis eltérésekkel megegyezik a CAMAC rendszerrel, mechanikája: a CSERESZNYE viszont a strukturális hasonlóság ellenére jelentős eltérést mutat. Az eltérések alapja, hogy a VEKTOR mechanikát metrikus rendszerben tervezték, így a vízszintes osztás $17,2 \text{ mm}$ helyett 20 mm , a modulok szélessége tehát 20 mm egész számú többszöröse lehet. A függőleges egység, az $1U=40 \text{ mm}$ a $44,45 \text{ mm}$ helyett. A VEKTOR keret legkisebb magassága ventilátor nélküli keretnél $6U$, szélessége pedig úgy van megválasztva, hogy 24 állomáshelyet biztosít az eredeti CAMAC 25 állomáshelyével szemben. A két mechanika főbb méreteit a 2.1. táblázat tartalmazza.

2.1. táblázat

CAMAC és VEKTOR mechanikák főbb méreteinek összehasonlítása

Méret		Szélesség mm	Magasság mm	Mélység mm
Keret	CAMAC	482 (19")	222 (266) 5U (6U)	525 (min 360)
	VEKTOR	520	240 (280) 6U (7U)	600 (min 400)
Modul	CAMAC	$n \cdot 17,2$	222	306
	VEKTOR	$n \cdot 20$	240	342

Mivel végülis méreteikben a VEKTOR egységek nagyobbak, egy CAMAC modult megfelelő CAMAC—VEKTOR adapterrel be lehet helyezni a VEKTOR keretbe, de fordítva nem. Itt jegyezzük még meg, hogy az ADATÚT kialakításához nem 86 -pontos direkt, hanem 90 -pontos indirekt csatlakozót használnak.

2.2.3.

CAMAC ADATÚT

A keretben levő modulok, felhasználói és vezérlő modulok információcseréje, adatforgalma a keret közepén végigfutó ADATÚT-on megy végbe. Az ADATÚT a keretben állomásonként elhelyezett 86-pontos (2×43) csatlakozó megfelelő összekötésével kialakított sokvezetékes busz. A keret 1...24 állomásain levő csatlakozók bekötése azonos, ezek a normál állomások, a 25-ik állomás csatlakozójának bekötése ettől eltérő, ez a vezérlőállomás. Ha keskenyebb keretet alkalmaznak, akkor is a jobb szélső állomást kell vezérlőállomásként bekötni és a többi normál állomásnak.

A jelvezetékek nagy része, valamint minden tápfeszültségvezeték buszvonala, ami úgy keletkezik, hogy az állomásokon levő csatlakozók azonos jelű érintkezőit összekötjük, multiplikáljuk. Ettől eltér az adatvezetékek (R, W) busza, ami csak a normál állomásokat köti össze, de a vezérlőállomás megfelelő pontjaira már nincs átkötve. A vezérlőállomás ezen pontjaira egyedi vezetékek csatlakoznak.

Az egyedi vezetékek — normál állomásonként kettő — a többi normál állomást kikerülve közvetlenül a vezérlőállomáson levő csatlakozó megfelelő pontjaira vannak kötve. Az ADATÚT továbbá tartalmaz szabad felhasználású pontokat, buszvonalatokat.

Az ADATÚT vonalait, jeleit az alábbi hét csoportra oszthatjuk:

- utasításvonalak (N, A, F).
- adatvonalak (R, W).
- állapotvonalak (L, B, Q, X).
- közös vezérlővonalak (Z, I, C).
- időzítésvonalak (S1, S2).
- szabad felhasználású, segédpontok, vonalak (P).
- tápfeszültség-vezetékek (± 24 V, ± 6 V, 0, NIM feszültségek).

A továbbiakban ebben a csoportosításban az egyes jelek, vonalak értelmezését, használatát ismertetjük.

UTASÍTÁS Egy CAMAC ADATÚT üzenet utasítás része címből (N, A) és funkciókódból áll (F).

N **Állomásszám.** A keretben minden modul a vezérlőállomás megfelelő pontjáról induló egyedi N_i vezetékkel címezhető. Nincs korlátozva, hogy a vezérlő egyszerre hány modult választhat ki, azaz címezhet. Az N vezetékek jelölése: $N1...N25$.

A **Alcím.** Egy modulon belüli, elkülöníthető alszektor címzésére szolgál. Az alcímek száma 16, decimális jelölésük: $A(0)...A(15)$. Az alcím az ADATÚT-on négy buszvonalat köt le: az A1, A2, A4, A8 vonalat. A dekódolás a modulban történik, a dekódolásnak teljesnek kell lennie.

F **Funkció.** A címzett modulban, az előírt alcímen végrehajtandó utasítás jellegét (pl. törlés, olvasás, vizsgálat stb.) szabja meg. A funkciókód öt ADATÚT buszvonalat foglal le, az F1, F2, F4, F8, F16 vonalat. Ezekkel 32 különféle funkciót írhatunk elő, jelölésük: $F(0), \dots, F(31)$. A 32 lehetséges funkciókód közül eddig csak 18 van szabványos funkcióra lekötve, a többi szabad felhasználású (a modul tervezője eldöntheti, hogy az adott kód milyen műveletet váltson ki), vagy fenntartott, amelyeket az ESONE bizottság később definiálhat.

A funkciókódokat feloszthatjuk aszerint is, hogy együtt járnak-e adatforgalommal (pl. olvasás) vagy nem (pl. vezérlés). A teljes funkciókód-választékot a Függelék 7.1. táblázata tartalmazza. A funkciókód dekódolása a modulban történik.

ADAT A CAMAC ADATÚT adatforgalmának szóhossza 24 bit, de természetesen ennél rövidebb szavakat is átvihetünk.

R **Olvasó** vonalak. Az adatot a modulból az R-vonalakon keresztül vihetjük át pl. a vezérlőbe, az ADATÚT-on 24 vonalat kötnek le, jelölésük: R1, R2, ... R24.

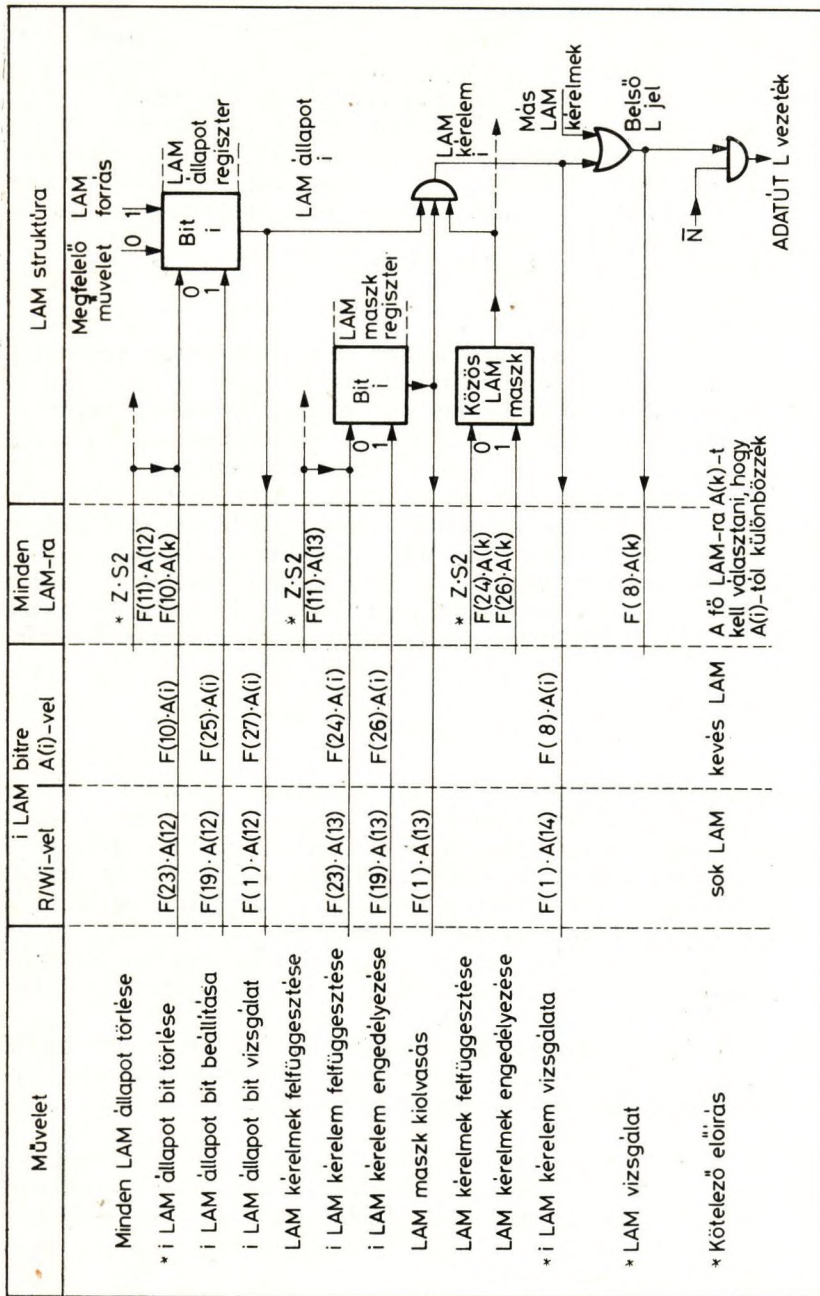
W **Író** vonalak. A vezérlőből a modulba ezeken a vonalakon keresztül vihetjük át az adatot, az ADATÚT-on 24 vonalat kötnek le, jelölésük: W1, W2, ... W24.

Az R és W vonalakon úgy kell adatot továbbítani, hogy a nagyobb index nagyobb helyértéknek feleljen meg.

ÁLLAPOT Az állapotinformációt az L, B, Q és X vonalak hordozzák.

L **LAM** (Look at Me: Figyelj rám). Egyedi vezeték. Minden normál állomás adott pontjáról közvetlenül a vezérlőállomás megfelelő pontjára kötött Li vezetéken jelezheti a modul, hogy figyelmet kér. A figyelem felhívás oka — akár a modulon belül is — többféle lehet, például, ha egy A/D átalakító modul végrehajtotta a konverziót, jelentkezik, hogy a számítógép vegye át a mérési adatot, egy számlálómodul jelezheti, hogy valamelyik regisztere túlcordult stb. (A LAM-jel hasonló a számítógép perifériás használatokor programmegszakítást kérő jelzőbithez, az *interrupt flag*-hez). A LAM megjelenése a vezérlőtől, programtól függően kiválthat: programmegszakítást, utasításátlépést vagy autonóm adatátvitelt. Ez utóbbi esetben az adatot közvetlenül a (vezérlő) számítógép tárolójába visszük át anélkül, hogy a futó program menete megváltozna. A LAM kezeléssel, generálással kapcsolatos kötelező és ajánlott előírásokat a 2.12. ábra foglalja össze. A figyelem felhívás kérés a LAM forrásból jövő jelre indul, amely beír egy ún. LAM ÁLLAPOT REGISZTER bitet. Ennek kimenő jele a LAM ÁLLAPOT jel csak úgy jut tovább, ha a hozzárendelt maszkbít ezt engedélyezi. Az engedélyezett LAM-ok, LAM KÉRELMEK logikai VAGY kapcsolata hozza létre a belső L jelet, amely \bar{N} -nel kapuzva jut ki az ADATÚT megfelelő L vezetékére. Ha egy modul L jelet hozott létre, nem törölheti a LAM állapotot, míg erre megfelelő utasítást (F(10), F(23) a vonatkozó alcímmel...) nem kap. Az L jel fennállása alatt lehetőséget kell biztosítani a tesztelésre (F(8) A(i) ill. A(k)). Sok LAM esetén célszerű a LAM információ szavas kezelése. Erre az esetre a LAM regiszterekhez a szabvány az A(12), A(13), A(14) alcímeiket rendeli. A közös vezérlőjelek közül a Z alaphelyzet jel a LAM állapot biteket törli, a maszkbiteket felfüggesztés állásba hozza. A LAM kezeléssel, generálással részletesen a 3.1.4. és 4.1. szakaszokban foglalkozunk.

B **Foglaltság.** Azt jelzi, hogy az ADATÚTON valamilyen művelet van folyamatban, valamilyen ADATÚT ciklus zajlik, vagyis az ADATÚT foglalt és újabb ciklus nem kezdhető el, amíg a B-jel meg nem szűnik. (Buszvonala.)



2.12. ábra
LAM kezelési előírások

Q **Válasz.** A válaszjellel a vezérlő a modul valamilyen kiválasztott részének állapotáról vagy az adatforgalom helyzetéről kaphat információt. L-jel vizsgálata során pl. az F(8) A(i) utasításra a Q vonalon jelentkező logikai 1 jelzi, hogy a modul L-jelet generált. Állapot vizsgálat során pl. az F(27) A(k) utasításra kapott Q válasz megadja az A(k) alcímű fokozat állapotát. Adatforgalomban a Q jel lehet hitelesítőjel oly módon, hogy az adatokkal együtt 25-ik bitként küldött Q jel azt jelezheti, hogy az adat érvényes-e vagy nem. Blokkátvitelnél az adatok mellett küldött Q jel a blokkátvitel módját jelzi. Van

- címletapogató (Address scan);
- ismétléses (Repeat);
- megállítós (Stop)

típusú blokkátvitel. *Címletapogató* során modulokban levő regisztereket keresünk fel sorjában oly módon, hogy N1, A(0) helyről indulunk. Ha az előírt írás vagy olvasás műveletet elvégeztük, megnöveljük az alcímet A(1) és ott is elvégezzük az előbbi műveletet, majd ismét megnöveljük az alcímet és elvégezzük a műveletet. Az első olyan esetben, amikor az írás vagy olvasás műveletnél Q=0 válasz érkezik, az állomászámot növeljük meg és az alcímet visszaállítjuk nullára (N2, A(0)). A kiolvasást, beírást növekvő alcímekkel folytatjuk, amíg ismét Q=0 válasz jelzi, hogy ebben a modulban sincs már további regiszter, át kell térni a következő állomásra. A Q válasz jelentése címletapogató üzemmód esetén:

Q=1 → Alcím + 1

Q=0 → Alcím = 0, Állomászám + 1.

Ismétléses üzemmódban egy adott regisztert olvasunk többször egymás után. Minden esetben, ha a regiszter Q=1 választ ad, az írás vagy olvasás érvényes volt. Ellenkező esetben Q=0-nál a műveletet meg kell ismételni, mindaddig, amíg a Q=1-et elérjük. *Megállítós üzemmódban* a modul a Q=0 válasszal az átvindó blokk végét jelzi. Az átvitelt mindaddig kell folytatni, amíg a modul a Q=1 választ adja, mert ez azt jelenti, hogy még blokkon belül vagyunk. Ha Q=0 válasz érkezik, azt jelenti, hogy a modul a blokkot lezárta (l. még 4.1. szakaszt).

X **Utasítás elfogadva.** Nyugtázó jel, minden megcímzett modul köteles X jelet küldeni a vezérlő felé, ha a kapott utasítást megértette és képes azt végrehajtani. (Buszvonala.)

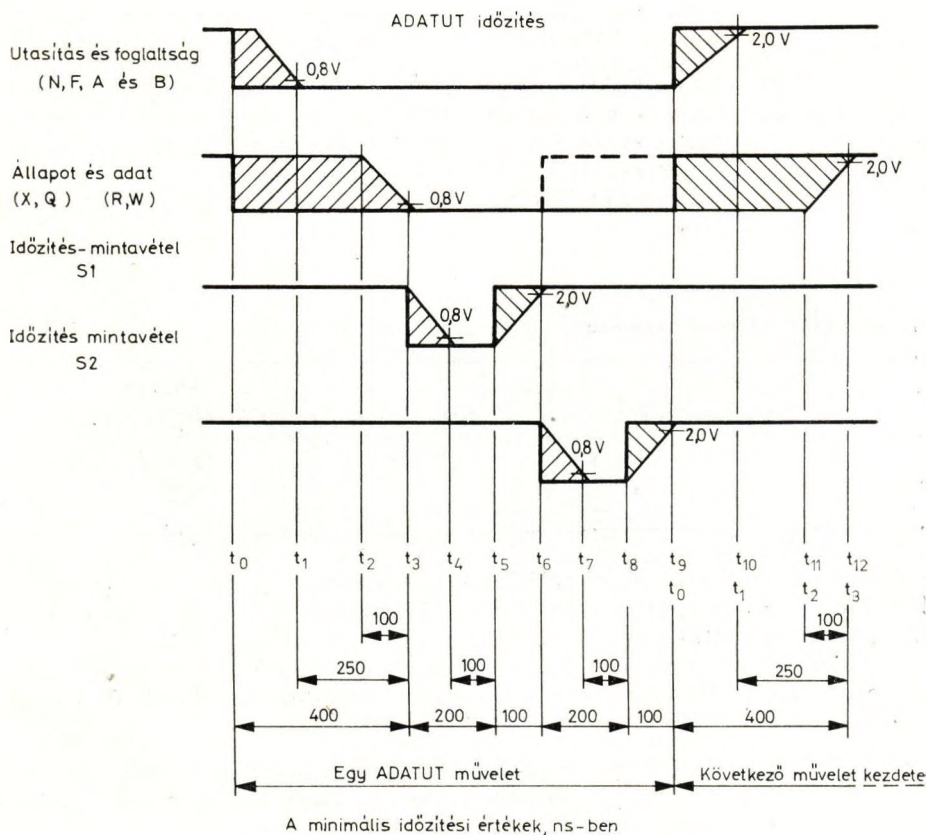
KÖZÖS
VEZÉRLÉS.

A közös vezérlőjelek cím nélküliek, azaz minden modulban egyszerre fejtik ki hatásukat. A Z és C jel az ADATÚT ciklussal jár (B, S2), az I független az ADATÚT műveletektől, bármikor megjelenhet, bármeddig fenntartható, ill. bármikor megszüntethető. (Mindhárom buszvonala.)

Z **Alaphelyzet.** Prioritása van minden más jel felett. Hatására minden modul vezérlőregisztere meghatározott kiinduló helyzetet vesz fel, a LAM jelentkezéseket törli, és felfüggeszti az L jel kiadásának lehetőségét (A Z jellel együtt I-t is ki kell adni.)

- C **Törlés.** Ez a jel töröl minden adatregisztert, bistabilt, amihez a modulokon belül kötve van, de érintetlenül hagyja pl. az üzemmódvezérlő regiszterek állapotát. Így egy összeállított mérést pl. C után változatlan, a program által beállított üzemmódban folytathatjuk, de üres adatregiszterekkel.
- I **Tiltás.** Az I tiltójel megjelenése megbénít minden modulaktivitást, amit a modulok tervezői célszerűnek tartottak. Pl. számlálóregiszter leáll, multiplexer nem lép tovább stb. Rendszerindításra használható oly módon, hogy programmal rendre előkészítjük a modulokat, a mérésakor megkívánt üzemmódokat, paramétereket beállítjuk és végül utolsó lépésként az I-tiltást megszüntetjük. Erre valamennyi modul egyszerre aktivizálódik, működésbe kezd.

IDŐZÍTÉS CAMAC ADATÚT ciklus időzítési viszonyait láthatjuk a 2.13. ábrán. A ciklus jellegzetes jelei a B foglaltságjel, ami jelzi a ciklus tartamát, és a két mintavévi jel. A ciklus hossza 1 μ s.



2.13. ábra
CAMAC ADATÚT művelet időzítése

S1, S2 **Mintavétel.** Az ADATÚT ciklusban a két mintavevő jel, S1, ill. S2 végzi egy-egy kijelölt, előkészített művelet végrehajtását. Pl. S1 időpillanatban írjuk be a vezérlőbe a kiadott utasítás hatására az R vezetékeken megjelenő adatot, S1-gyel léptetünk egy regisztert, indítunk egy konverziót stb. A 2.13-as ábrán látható, hogy egy ciklus indításakor a B jellel egyidőben kell kiadni az utasítás-(NAF) jeleket. Adatforgalommal járó utasításnál az adatot, valamint az X és Q állapotjeleket legkésőbb S1 megjelenése előtt (t_3) kell a vonalra juttatni, hisz ezekből majd S1-gyel veszünk mintát. Az S1 jel alatt az R, W, X, Q adatok nem változhatnak meg, S2 alatt viszont igen. S2-t használhatjuk pl. regiszterek törlésére.

SEGÉD-ÖSZ-
SZEKÖTÉSEK.

P1, P2

P3...P7

Minden normál állomáson 5 érintkező (P1...P5), a vezérlő-állomáson 7 érintkező (P1...P7) specifikálatlan felhasználásra szabad. **Szabad buszvezetékek.** Minden normál állomáson a P1, ill a P2 pontokat összekötve, két szabad felhasználású buszt kell kialakítani, **Segédérintkezők.** A normál állomáson P3, P4, P5 és a vezérlő-állomás P1...P7 érintkezői nincsenek ADATÚT huzalozással ellátva, a felhasználó rendelkezésére állnak mint szabad felhasználású segédérintkezők. Ezekre a pontokra egy-egy modulból kihozhatjuk külső felhasználásra pl. az I, C, N vagy L jelet, vagy a földvezetéket.

TÁP-
FESZÜLTÉG.

Az ADATÚT-on 6 vezeték a kötelező CAMAC tápfeszültségek számára, 6 vezeték a NIM tápfeszültségek számára szabványosítottak és van két fenntartott tápfeszültség-vezeték, az Y1 és Y2, amelyekről az ESONE bizottság később dönt. A tápfeszültség-vezetékek terhelhetőségét a 2.2. táblázatban adtuk meg.

2.2. táblázat

Az ADATÚT tápfeszültség-vezetékeinek terhelhetősége

Feszültség, V	Kötelező CAMAC tápfeszültségek				Kiegészítő (NIM) tápfeszültségek			
	+24	+6	-6	-24	+200	+12	-12	~117
Áram, A	6	25	25	6	0,1	2	2	0,5

Az ADATÚT jeleket a Függelék 7.2. táblázatában foglaltuk össze; a normál állomás és a vezérlőállomás csatlakozóinak érintkezőkiosztását, bekötését pedig a Függelék 7.3. táblázatában adtuk meg.

Az ADATÚT buszvonalaik kialakítása huzalozott VAGY kapuval történik. A jel-szinteket a 2.3. táblázat tartalmazza. A terhelési viszonyokat a következő pontokban foglaltuk össze:

- A modulból a vezérlő felé menő jeleknek 10 egység terhelést kell meghajtaniok, amelyek közül 6 a vezérlőben szintfelhúzásra (pull-up), 4 pedig a fogadó áramkörök meghajtására szolgál.
- A vezérlőből a modul felé menő jeleknek $25 - n + 6$ egység terhelést kell meghajtaniok. Ebből 6 egység terhelés a vezérlőben szintfelhúzásra szolgál. Az n a vezérlő

által elfoglalt állomások számát jelenti. Fontos megjegyezni, hogy a modulok a beérkező jeleket állomáshelyenként max. 1 egység terheléssel (1,6 mA-rel) terhelhetik.

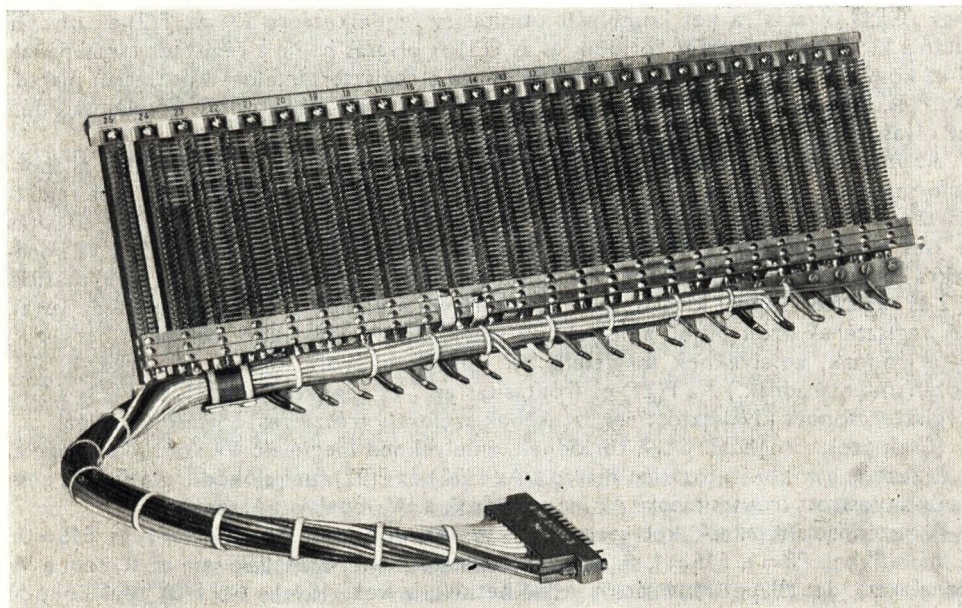
- Az előző pontban előírt követelménytől eltér a mintavevő jelek meghajtása, ami nagyobb teljesítményű, ill. az Állomásszám- (N) vezeték, aminél csak 10 egység terhelés meghajtását várjuk el úgy, hogy 6-ot szokás szerint szintfelhúzásra használunk.
- A vonalmeghajtó fokozatok visszaramának maximális értékét állomáshelyenként 100 μ A-re korlátozták.

2.3. táblázat

Jelszintek az ADATÚT-on

Logikai 1	Bemenet Kimenet	0...+0,8 V 0...+0,5 V
Logikai 0	Bemenet Kimenet	+2 ... +5,5 V +3,5... +5,5 V

Az ADATÚT kivitelezése legtöbb esetben tekert huzalkötéssel készül, elég gyakori a többrétegű alapelemez (MOTHER BOARD-ot) alkalmazó megoldás, de előfordul forrasztott kötésű ADATÚT is, vannak továbbá vegyes technológiát alkalmazó megoldások, amelyeknél a jelvezetékek tekert huzalkötéssel, a tápfeszültségek csatlakozásai forrasztással, sinezéssel készülnek. Egy ilyen vegyes technológiájú ADATÚT szerelvényt láthatunk a 2.14-es ábrán.



2.14. ábra
Vegyes technológiájú ADATÚT szerelvény

ADATÚT utasítások

Az utasítás összetevődik az N, A, F vezetékekre adott jelekből. Minden ADATÚT utasításműveletnél a keretvezérlő generálja a megfelelő utasítást, amelyet kísér a B foglaltságjel és a két mintavevő jel. Az utasítások egy része az R, W vonalakon adatáramlást hoz létre. A modul, ha felismeri az utasítást, X jelet generál és a Q vezetéken egy bit állapotinformációt is közölhet.

Az utasításokat regiszterekre vonatkoztatjuk, ahol a regiszterek fogalmát általános értelemben használjuk. Regiszter ebben az esetben minden címezhető adatforrás vagy -nyelő, függetlenül attól, hogy tényleges adattároló elemeket tartalmaz-e.

A regisztereket két nagy csoportra osztjuk: 1-es regisztercsoportra és 2-es regisztercsoportra. (Group 1, Group 2 regiszterek.) Jelölésük: GP1, GP2. Mind a két regisztercsoportban 16 regisztert tudunk megkülönböztetni, A alcímekkel. Az F kódok nagy része — írás, olvasás, törlés — mindkét regisztercsoportra szabványosítva van, vagyis azonos típusú műveletet mindkét regisztercsoport bármelyik A(i) alcímű regiszterén el tudunk végezni. GP1 regiszterek általában az adatregiszterek, GP2 regiszterek pedig általában a vezérlő- (üzemmód, paraméter stb.) regiszterek.

Az utasítások jellegét az F kódok szabják meg, így az utasításokat ezek alapján csoportosíthatjuk. Vannak:

- olvasóutasítások: F(0)...F(7),
- íróutasítások: F(16)...F(23),
- vezérlőutasítások: F(8)...F(15), F(24)...F(31).

Az *olvasóutasítások* adatot és állapotinformációt visznek a modulból a vezérlőbe. Az F(0) funkciókódú utasítás az 1-es regisztercsoport A(i) alcímmel kiválasztott regiszterének, az F(1) utasítás pedig a 2-es regisztercsoport kiválasztott regiszterének olvasását végzi.

Olvasás után a regiszterek tartalma változatlan marad. Az 1-es csoport regisztereinek (GP1) olvasására két kombinált utasítás is rendelkezésre áll: az F(2)-es olvasás után a kiolvasott regiszter törlődik és az F(3)-as olvasás pedig a regiszter tartalmának 1-es komplementjét viszi át a vezérlőbe, maga a regisztertartalom változatlan marad. Az F(4), F(6) nem szabványos, az F(5), F(7) fenntartott olvasóutasítás. Értelemszerűen az olvasóutasítások használják az R vonalakat.

Az *íróutasítások* a vezérlőből adatot visznek át a modulba és egyidejűleg állapotinformációt a modulból a vezérlőbe. Az F(16) funkciókódú íróutasítás az 1-es regisztercsoport A(i) alcímmel kiválasztott regiszterének tartalmát, az F(17)-es utasítás a 2-es regisztercsoport kiválasztott regiszterének tartalmát a vezérlőből a W vonalakra helyezett adattal megegyezően állítja be. Az F(18) utasítás az 1-es, az F(19) utasítás a 2-es regisztercsoport kiválasztott regiszterének szelektív (részleges) beírását végzi. (A regiszterek azon bitjei, amelyeknek megfelelő W vonalakon logikai „1” érkezik, 1-be íródnak, amelyeknek megfelelő W vonalakon pedig logikai „0” érkezik, változatlanul maradnak.) Az F(21)-es íróutasítás az 1-es regisztercsoport, az F(23) a 2-es regisztercsoport kiválasztott regiszterének szelektív (részleges) törlésére szolgál.

(Csak azok a regiszterbitek törlődnek, amelyeknek megfelelő W vonalakon logikai 1 érkezett, a többi változatlan marad.) Az F(20) és F(22) funkciókódú utasítások nem szabványosított íróutasítások. Az íróutasítások a W vonalakat használják.

A *vezérlőutasításoknak* két csoportja van. Az első csoportban $F8=1$ és $F16=0$, a másodikban $F8=1$, $F16=1$. A vezérlőutasítások nem használják sem az R, sem a W vonalakat, de állapotinformációt közölhetnek a vezérlővel. Az F(8) funkciókódú utasítást az L jel vagy LAM kérelem vizsgálatára kell használni. A modul az F(8) vizsgálatra a Q vonalon ad választ, de nem törölheti a LAM állapotot. Erre külön utasítás

az F(10) szolgál, megfelelő alcímmel. Regiszterek törlésére az 1-es csoportban az F(9)-et, a 2-es csoportban az F(11)-et szabványosították. A második vezérlőutasítás-csoportban az F(24) funkciókódú utasítás felfüggeszti, az F(26) utasítás engedélyezi a modul valamilyen akcióját, működését (pl. LAM kérelem, számlálás, konverzió-indítás stb.). Az F(25) funkciókódú utasítás valamilyen akció, működés kiváltására, végrehajtására, indítására, leállítására szolgál. Tipikus végrehajtó utasítás a multiplexerleptetés, konverzióindítás, számlálótartalom-növelés. Az F(27) funkciókódú utasítással a modul valamilyen kiválasztott részének állapotát (Status) vizsgálhatjuk. A választ a Q vonalon kapjuk. Ezt az utasítást kell használni a LAM állapot vizsgálatára, de használhatjuk akármilyen más állapot jellegű információ vizsgálatára, így például foglaltság, várakozás, tiltás stb. vizsgálatára. A többi vezérlőutasítás közül az F(12), F(14), ill. az F(28), F(30) nem szabványosított, szabad felhasználású, az F(13), F(15), ill. az F(29), F(31) fenntartott vezérlőutasítások. A fenntartott utasításokról az ESONE bizottság később dönthet, hogy milyen további szabványos funkcióra kívánja alkalmazni, tartalékul szolgálnak a szabványos funkciók bővítéséhez (l. Függelék 7.1. táblázatát)

2.2.4.

CAMAC CSOPORT, CSOPORTÚT

A CSOPORT szervezésű többkeretes CAMAC rendszerekben a keretek a CSOPORT-ÚT-ra csatlakoznak. A CSOPORTÚT a csoportmeghajtótól indul ki, megszakítás nélkül végigfut, a keretek keretvezérlőin és a csoportlezárónál végződik. A keretvezérlők a CSOPORTÚT fogadására és továbbadására 2 db 132-pontos csatlakozóval vannak felszerelve. A csoportmeghajtó a csoportműveletek alatt maximálisan hét kerettel tud üzenetet váltani, de egy csoportmeghajtó akár több csoportot is vezérelhet.

A keretvezérlők normál állapotban élő (on-line) kapcsolatban vannak a CSOPORT-ÚT-tal, de ezen felül rendelkeznek olyan lehetőséggel is, hogy lekapcsolhatók, off-line állapotba tehetők. Ilyen lekapcsolt állásban, még ha fizikailag rajta is maradnak a CSOPORTÚT-on, annak jelforgalmát nem zavarják. Egy keret a CSOPORTÚT jelforgalmát akkor sem zavarhatja, ha tápfeszültségét kikapcsolják. A csoportmeghajtó mindenkor meg tudja állapítani, hogy melyek az élő keretek.

A csoportműködésnek két tipikus művelete van, az utasításművelet és az igénykezelő eljárás. Az utasításművelet alkalmas — egyszerre akár több keretben is — a már az ADATÚT-nál is megismert írás, olvasás, vezérlés típusú műveletek elvégzésére.

A csoport igénykezelő eljárása lehet egyszintű, ami tulajdonképpen csak jelzi, ill. továbbítja a csoportban valahol keletkezett LAM jelet (Branch Demand), de lehet többszintű is, ami lehetőséget ad a csoportmeghajtónak 24 különféle igény (Demand) azonosítására (osztályozott L művelet). A CSOPORTÚT adatvonalain az információ mindkét irányba továbbítható, így az író—olvasó vezetékek ugyanazok. A CSOPORT-ÚT-on a közös vezérlőjelek közül csak az alaphelyzet jelet találjuk.

UTASÍTÁS. Az utasítás itt is, mint az ADATÚT-nál, címből és funkciókódból áll. Az eltérés egyrészt abból adódik, hogy a címet ki kell egészíteni a keretcímmel, másrészt abból, hogy a CSOPORTÚT-on az állomátszámot bináris kódban továbbítjuk.

BCR Keretcím. A keretek címezésére a CSOPORTÚT-on 7 vezeték (érpár) van fenntartva. A keretcím-vezetékek jelölése: BCR1, BCR2...BCR7. Minden keretet egy külön keretcím-vezetékre kell

- kötni. A szabvány nem köti meg, hogy melyik keret melyik keret-cím-vezetékre legyen kötve. A csoportmehajtó egyszerre több keretet is megcímezhet, így adott utasítást egyszerre több keretben is végre lehet hajtani.
- BN **Állomásszám.** Egy kereten belül a modul helyének meghatározására, az állomásszám megadására 5-bites bináris kód szolgál. Az állomásszám a CSOPORTÚT-on 5 buszvonlat (érpárt) köt le, jelölésük: BN1, BN2, BN4, BN8, BN16. Az állomásszám dekódolása a keretvezérlőben történik. Mivel a keretvezérlő legalább egy normál állomást is lefoglal, csak 23 állomás címezését kell biztosítani, az 5-bites kód pedig 32-t tesz lehetővé. A fennmaradó keretcímekeket $N(24) \dots N(32)$ a keretvezérlők hasznosítják. (L. 3. fejezet.)
- BA **Alcím.** Használata megegyezik az ADATÚT azonos nevű vonalaival. A CSOPORTÚT-on a négy alcímvezeték jelölése: BA1, BA2, BA4, BA8. A keretvezérlő ezen a négy vezetéken jövő jelet egyszerűen továbbadja az ADATÚT-ra.
- BF **Funkció.** Szerepe azonos az ADATÚT megfelelő vonalaival. A CSOPORTÚT-on az öt funkciókódvonal jelölése: BF1, BF2, BF4, BF8, BF16. A keretvezérlő a BF vonalakon érkező funkciókódot változtatás nélkül továbbadja az ADATÚT F vonalaira.
- ADAT A CSOPORTÚT szóhossza megegyezik az ADATÚT szóhosszával, ez is 24 bit.
- BRW **Olvasó és író** vonalak. Az ADATÚT-tól eltérően a CSOPORTÚT-on írásra és olvasásra ugyanazt a 24 vonalat használjuk, tehát ezeken a vonalakon a jelforgalom kétirányú. Az adatáramlás irányát az F kód szabja meg. Írás típusú utasításoknál a csoportmehajtótól a keretek felé, olvasás típusú utasításoknál a kerektől a csoportmehajtó felé történik az adatátvitel, vezérlőutasításoknál pedig nincs adatforgalom. A BRW vonalakat használjuk még igénykezelés során (l. később) a LAM szó továbbbítésére. Az olvasó- és írvonalak jelölése: BRW1...BRW24.
- ÁLLAPOT A állapotinformációt a BQ és BX jelek hordozzák.
- BQ **Válasz.** Jelentése, szerepe azonos az ADATÚT Q jelével, és a keretvezérlő minden ADATÚT műveletnél keletkező Q jelet tovább kell, hogy adja a CSOPORTÚT BQ vezetékére ($BQ=Q$). Ha olyan műveletet hajt végre a keretvezérlő, ami csak a keretvezérlőre vonatkozik, pl. a keretvezérlő valamely állapotát teszteljük, neki kell a Q választ generálnia. A BQ vezetéken levő jel az összes keretek válaszának VAGY kombinációja.
- BX **Utasítás elfogadva.** Jelentése, szerepe azonos az ADATÚT X jelével. Egy keretnek, ha ADATÚT műveletet hajt végre, a keletkező X jelet továbbítani kell a BX vezetékre ($BX=X$). Minden más utasításműveletnél $BX=1$ -et kell generálnia a keretvezérlőnek, ha elfogadta az utasítást és $BX=0$ -t, ha nem. A BX jel az összes keretek jeleinek VAGY kapcsolatából áll elő.
- IDŐZÍTÉS A CSOPORTÚT időzítése kézfogás (hand-shake) típusú. A kapcsolatot csoportmehajtó BTA és keretvezérlők BTBi jelei tartják.
- BTA, BTB **Időzítő jelek.** A műveletek során a csoportmehajtó kezdeményez a BTA jel kiadásával, a BTBi jelet a megcímezett keretek keret-

vezérlői a BCRi címmel összhangban válaszul küldik a csoportmeghajtónak, egyedileg. Egy csoportban egyszerre maximálisan 7 keret lehet, így BTB vezetékéből is 7 van, jelölésük: BTB1, BTB2...BTB7. A BTB jelek alaphelyzete az összes többi CSOPORTÚT jellel ellentétben logikai 1. Ez ad lehetőséget a csoportmeghajtónak, hogy megállapítsa, melyek az üzemben levő (on-line) keretek a csoportban. A csoportműveletek végrehajtása 4 fázisra bontható, ezekre a jelek ismertetése után visszatérünk.

IGÉNY
(KÉRÉS)
KEZELÉS.
BD

A CSOPORT bármelyik egységében keletkező L jel valamilyen kiszolgálást kér. A CSOPORTÚT-on két kiszolgáló, igénykezelési lehetőség van, az egyik a BD, a másik a BG jellel kapcsolatos. **CSOPORT IGÉNY** (BRANCH DEMAND). A keretek a modulokból beérkező L jelek valamilyen logikai kapcsolatából, a leg-egyszerűbb esetben VAGY kapcsolatából IGÉNY (DEMAND, D) jelet állítanak elő. A CSOPORT IGÉNY a BD vonalon az egyes keretek D jeleinek VAGY kapcsolatából alakul ki. (A BD vonal busz-vonal, és ezt bármelyik keret kijelölheti.) A BD jel hozza a csoportmeghajtó tudomására, hogy valamilyen kiszolgálási igény van a CSOPORT-ban. A BD jel bármikor megjelenhet.

BG **Osztályozott L kérés** (GRADED—L REQUEST). Az osztályozott L műveletet a csoportmeghajtó a BG jel kiadásával kezdi, az összes élő (on-line) keret címzése mellett. Minden címzett keret egy 24-bites osztályozott L szót helyez a BRW vonalakra. Ezeknek a 24-bites osztályozott L szavaknak a bitenkénti VAGY kombinációja jut el a csoportmeghajtóhoz. Az osztályozást a keretekben úgy kell elvégezni, hogy az eredő 24-bites szóban a csoport L jelei, ill. ezek valamilyen logikai kombinációi prioritási sorrendben jelenjenek meg. (A LAM kezeléssel részletesen a 3. fejezetben foglalkozunk.)

KÖZÖS
VEZÉRLÉS.

A CSOPORTÚT jelei között csak a Z-nek megfelelő BZ adatút jelet találjuk, C és I hiányzik.

BZ

Csoport alaphelyzet. Szerepe, jelentése megegyezik az ADAÚT Z jelével. BZ megjelenésére minden élő (on-line) keretnek Z jelet kell generálni, amit természetesen a kötelező B, S2 és I jel kíséri az ADATÚT-on. A BZ jelnek minden más jel felett prioritása van. Azért, hogy a zavarjelektől megkülönböztethető legyen, BZ minimális szélessége 10 µs-ra van előírva.

FENN-
TARTOTT
VEZETÉKEK.

BV

A CSOPORTÚT-on 7 fenntartott vezeték (érpár) van.

Jelölésük: BV1...BV7. A későbbi követelményeknek megfelelően ezek felhasználásáról az ESONE bizottság BV7, BV6... sorrendben dönt.

ÁRNYÉKOLÓ
VEZETÉK.

BSC

A CSOPORTÚT-on egy érpár — általában párhuzamosan kötve — a CSOPORTÚT kábel árnyékolásának továbbítására szolgál. Jelölése: BSC.

Ezt a vezetékét a csoportlezáróban kell földelni. A többi egység (vezérlő), amin a CSOPORTÚT végighalad, csak lehetőséget tartalmaz az árnyékolás földelésére, ha szükséges.

A CSOPORTÚT jeleit a 1. Függelék IV. táblázatában foglaltuk össze. A szabványos Hughes WSS 0132 típusú CSOPORTÚT csatlakozó, érintkező kiosztását pedig az 1. Függelék V. táblázata tartalmazza.

A CSOPORTÚT buszvonalai 66 sodrott érpárból állnak. Az érpárok meghajtása aszimmetrikus (jel—föld). A vonalakat legalább az egyik végén hullámellenállással kell lezárni. Tipikus lezáró ellenállás 70 és 100 Ω közé esik. A lezárás biztosítja a zavaró reflexiók elkerülését és a huzalozott VAGY kapus vonal kihúzó áramát. A CSOPORTÚT-on levő jelszinteket a 2.4. táblázatban foglaltuk össze.

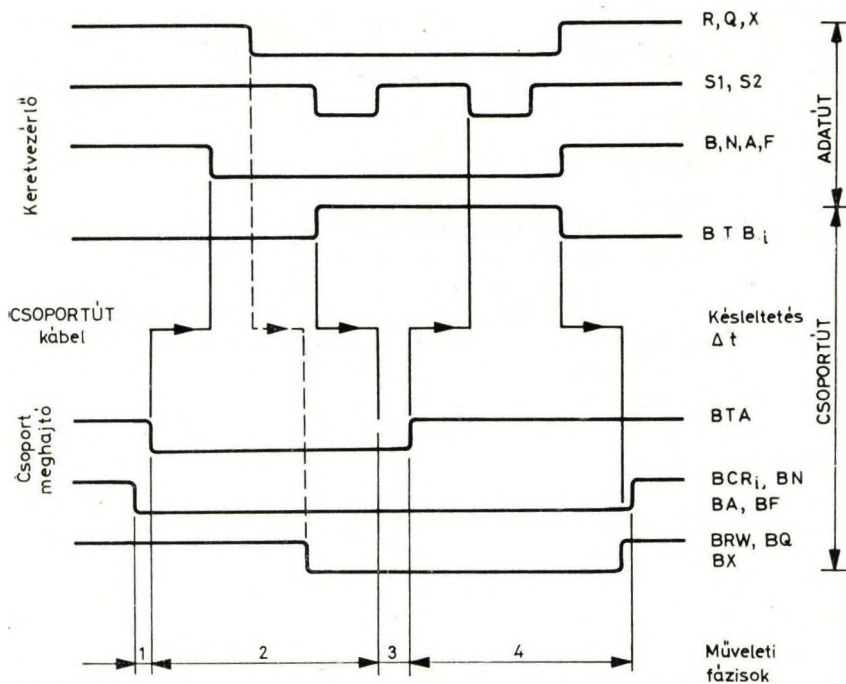
A vonalra kapcsolódó vevőáramkörök (vonalanként maximum 8) célszerű, ha a vonalat $\pm 0,3$ mA-nél többel nem terhelik, de semmiképpen nem haladhatják meg az

2.4. táblázat

Jelszintek a CSOPORTÚT-on

Logikai 1	Bemenet Kimenet	0 V...1,2 V 0 V...0,5 V
Logikai 0	Bemenet Kimenet	+2,4 V...+5,5 V +4,1 V*...+4,5 V

* javasolt érték.



2.15. ábra
CSOPORTÚT művelet (olvasás)

1,6 mA-es terhelést. A meghajtó áramköröknek a viszonylag kis kihúzó ellenállás miatt legalább 127 mA áramnyelő kapacitással kell rendelkezniük.

A CSOPORTÚT hossza az adott jelszint, zaj és terhelési viszonyokat figyelembevéve 20...30 m-nél hosszabb nem lehet [2.10].

A CSOPORTÚT műveleteket két nagy csoportra osztjuk, vannak utasításműveletek és osztályozott L művelet. Az utasításműveletek lehetnek: olvasás, írás, vezérlés típusú műveletek. A CSOPORTÚT műveleteket minden esetben a csoportidőzítő jelek vezérlik (BTA, BTB1...BTB7). Egy művelet jellegzetesen négy fázisra bontható (2.15. ábra):

1. fázis: A csoportmeghajtó előkészíti a műveletet, kihelyezi a CSOPORTÚT vonalakra az utasítást (BCRi, BN, BA, BF) és kis késleltetés után kiadja a 2. fázist indító BTA=1 jelet.
2. fázis: BTA=1 jelre a megcímzett keretvezérlő (BCRi) Δt késleltetés után ADATÚT műveletet indít, kihelyezi az ADATÚT-ra az N, A, F jeleket és elindítja az ADATÚT ciklust (B). A példaképpeni utasítás olvasás, így a címzett modul az R, Q, X vezetésekre adja a kért adatot, amit a vezérlő tovább ad a CSOPORTÚT BRW, BQ, BX vonalaira. Amikor ez megtörtént, legkorábban az ADATÚT ciklus t_3 időpillanatában (2.13. ábra), kiadja a BTBi=0 jelet, ami a 3. fázist indítja.
3. fázis: A csoportmeghajtó Δt késleltetés után érzékeli a BTBi jel $1 \rightarrow 0$ átmenetet és kis belső késleltetéssel átveszi a BRW, BQ, BX vonalakon levő információt. Amikor ez megtörtént, megszünteti a BTA jelet, de az utasítást még kint hagyja a vonalakon.
4. fázis: A keretvezérlő a BTA jel $1 \rightarrow 0$ átmenetéből tudja, hogy a csoportmeghajtó az adatot átvette és így az ADATÚT ciklust, amit S2 előtt felfüggesztett, be lehet fejezni. (Az ADATÚT ciklust S2 előtt azért kell felfüggeszteni, mert pl. egy F(2) típusú utasítás S2-re törli az adatot, és így a csoportmeghajtó nem tudná átvenni.) Az ADATÚT ciklus végén megszűnnek a B, R, Q, X jelek és a keretvezérlő visszaállítja BTBi=1 jelet, mire a csoportmeghajtó is leveszi a CSOPORTÚT vonalokról az utasítást. (A BRW, BQ, BY jelek a megfelelő ADATÚT jelek után, a kábel késleltetésnek megfelelő idő múlva automatikusan megszűnnek.) Ezzel a CSOPORT művelet lezárul.

Az írásművelet hasonlóképpen zajlik le, azzal a különbséggel, hogy a BRW vonalakra az adatot a csoportmeghajtó generálja és tartja fenn az utasításjelekkel azonos ideig.

A vezérlésművelet a fenti, olvasás típusú művelethez hasonlít, azzal a különbséggel, hogy a BRW vonalakon nincs adatforgalom, csak a BQ, BX vonalakon.

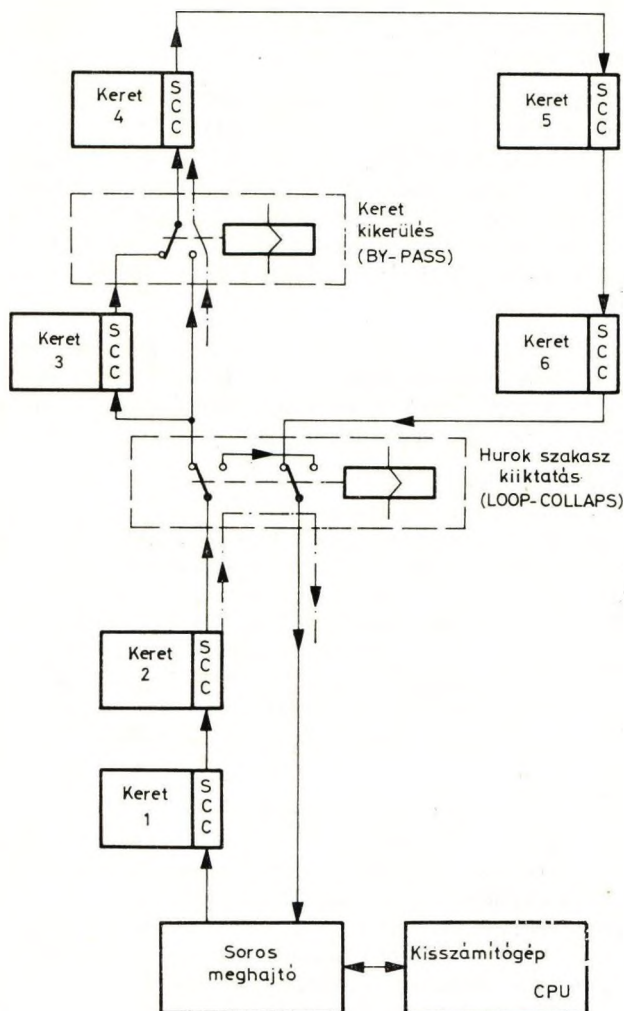
A műveletek mindegyikében egyszerre több élő keret is címezhető, ez esetben minden fázis indítása előtt az összes címzett keret választát meg kell várni. (Off-line keret nem címezhető, mert a válaszjel hiányában a működés megakadhat.)

Az osztályozott L művelet megegyezik egy többkeretes olvasásművelettel, azzal a különbséggel, hogy az utasításvonalak (BN, BA, BF) helyett a BG vonalat kell kijelölni a művelet tartamára, és minden élő (on-line) keretet címezni kell. Az osztályozott L műveletet általában a BD jel váltja ki, de ettől függetlenül is bármikor indítható.

2.2.5.

Soros CAMAC, SOROS FŐÚT

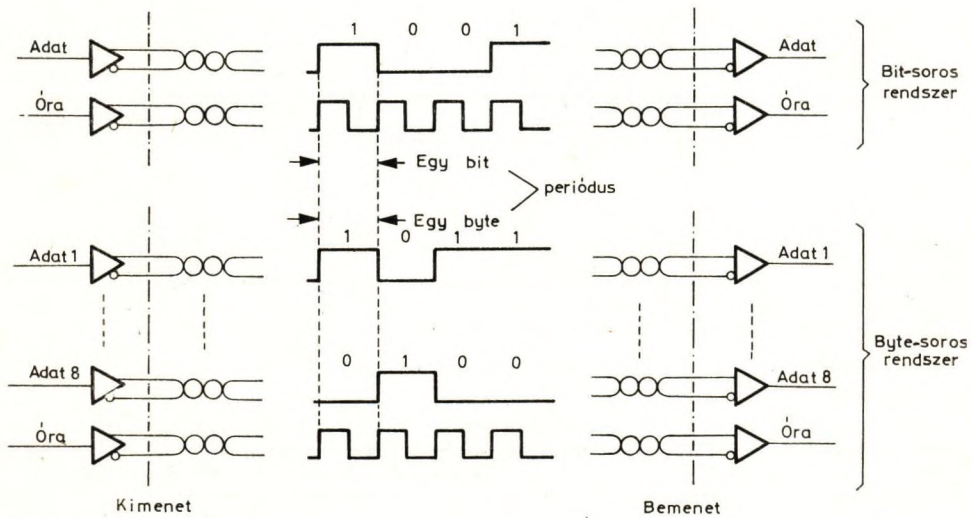
Ha egy rendszert alkotó keretek a vezérlő számítógéptől és egymástól is nagy távolságra helyezkednek el, a sokvezetékes CSOPORTÚT alkalmazása komoly nehézségeket vet fel. Ilyen esetekben a soros CAMAC alkalmazása a célszerűbb. Soros CAMAC-nál a keretek, modulok változatlanok, csak a keretek összekapcsolása történik az ún. SOROS FŐÚT-tal. A SOROS FŐÚT hossza nincs korlátozva, kiindul a számítógépnél levő soros meghajtótól, végighalad a keretek soros keretvezérlőin és a soros meghajtónál végződik. Így tehát a SOROS FŐÚT egy hurkot képez, a hurokban az információ-



2.16. ábra
SOROS FŐÚT folyamatosságának biztosítása
 nem működő keret kikerülésével és
 meghibásodott hurokszakasz kiiktatásával

áramlás egyirányú. A hurok magába foglalhat külön e célra fektetett kábelt, távbeszélő-vonalat, rádiókapcsolatot, mikrohullámú csatornát stb. Az átviteli vonalak, vonal-szakaszok tulajdonságainak megfelelően beiktathatunk modemeket, alkalmazhatunk hibajavító eljárásokat. A zárt hurok folytonossága nem szakadhat meg. Ha egy keret meghibásodott vagy lekapcsolt (off-line) állapotban van, vagy nem kap tápfeszültséget, lehetőséget kell biztosítani a keret kikerülésére (by-pass). Ha a hurok valamely szakasza hibásodott meg, a SOROS FŐÚT elmenő és visszatérő ágát a hibás szakasz előtt össze kell kötni, hogy a hibás szakasz kiiktatható legyen (loop-collaps). A 2.16. ábrán egy hatkeretes rendszer esetén a 3-as keret kikerülését, és a 3, 4, 5, 6 keretekhez tartó hurokszakasz kiiktatását mutattuk be. A soros keretvezérlő által vett és továbbított információ nem egyezik meg minden esetben, így a bemeneti és kimeneti csatlakozó nincs átkötve, mint a CSOPORTÚT-nál. A két csatlakozó között aktív elektronika van, ezek tápfeszültséghiánya a hurok megszakadását okozná, ezért ilyen esetben pl. relével kell a keretet kikerülni. (A kikerülést, hibás hurokszakaszok kiiktatását végző relék vezérlése a soros keretvezérlők ki-, ill. bemeneti csatlakozójának megfelelő pontjáról történik.)

A SOROS FŐÚT-ra maximálisan 62 keret köthető. A keretek keretvezérlőin és a soros meghajtón szabványos bekötésű kimeneti és bemeneti csatlakozó van elhelyezve a SOROS FŐÚT meghajtására és fogadására (D-port). E szabványos csatlakozók felhasználásával és külön a SOROS FŐÚT számára fektetett egyedi kábellel létrehozott



2.17. ábra
Bitsoros és byte-soros rendszerek

hurok jelforgalma lehet bit-soros vagy byte-soros. Az információ továbbítása bit-soros esetben 2, byte-soros esetben 9 sodrott érpáron valósítható meg. Mindkét esetben egy sodrott vezeték végzi az órajel továbbítását (2.17. ábra). Az órajel megengedett maximális frekvenciája 5 MHz.

A SOROS FŐÚT egyes szakaszait, mint láttuk, általános célú távadatátviteli csatornák is képezhetik (távbeszélővonal-modem stb.). Ilyen esetekben a soros CAMAC szabványos D típusú csatlakozója és az átviteli vonal közé egy adapter kerül, amely

2.5. táblázat

SOROS FŐÚT üzenetek

a) Utasításüzenet

	8	7	6	5	4	3	2	1	← bit byte ↓ 1
P1	0		SC32	—	—	—	—	SC1	
P2	0		M2 0	M1 0	SA8	—	—	SA1	2
P3	0		1*	SF16	—	—	—	SF1	3
P4	0		1*	SN16	—	—	—	SN1	4
P5	0		SW24	—	—	—	—	SW19	5
P6	0		SW18	—	—	—	—	SW13	6
P7	0		SW12	—	—	—	—	SW7	7
P8	0		SW6	—	—	—	—	SW1	8
PΣ	0		Σ6	—	—	—	—	Σ1	← összegbyte

Válasz

	1	0	1	1	1	1	1	1	← szünetbyte
	1	1	1	0	0	0	0	0	← záróbyte

 ↑
határolóbit

* = fenntartott bit.

b) Válaszüzenet

	8	7	6	5	4	3	2	1	← bit byte ↓ 1
P1	0		SC32	—	—	—	—	SC1	
P2	0		M2 0	M1 1	D ERR	SQ	SX	ERR	2
P3	0		SR24	—	—	—	—	SR19	3
P4	0		SR18	—	—	—	—	SR13	4
P5	0		SR12	—	—	—	—	SR7	5
P6	0		SR6	—	—	—	—	SR1	6
PΣ	1		Σ6	—	—	—	—	Σ1	← záróösszeg byte

 ↑
határolóbit

c) Igény- (DEMAND) üzenet

	8	7	6	5	4	3	2	1	← bit byte ↓ 1
P1	0	SC32		--		-		-	SC1
P2	0	M2 1	SGL5		-		-		SGL1
PΣ	1	Σ6		-		-		-	Σ1

↑
határolóbit

← záróösszeg
byte

P1, P2...P8 és PΣ keresztirányú paritás
Σ1, Σ2...Σ6 hosszirányú paritás

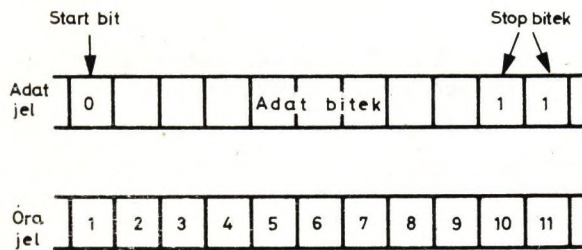
M1, M2 üzenetazonosító
M1=0, M2=0 utasítás
M1=1, M2=0 válasz
M2=1 igény

lefordítja a D típusú csatlakozást az adott vonalnak megfelelő, előre nem rögzíthető (undefined, U-típusú) csatlakozásra (3.20. ábra). (Ezekkel a kérdésekkel a TID—26488 leírás A3. fejezete foglalkozik részletesen.)

A soros CAMAC szabvány még nincs minden részletében elfogadva. Az ESONE ADATÚT munkacsoportja dolgozik néhány nyitott kérdés (pl. több keret egyidejű címzése stb.) lezárásán és már eddig is kiadott az eredetileg az AEC NIM bizottság által készített leírás pontosítására, kiegészítésére kisebb módosítására egy ESONE (SH)02 jelű füzetet. Nem régen alakult meg továbbá az ún. „Soros rendszer kompatibilitási alcsoport” (SSCSG), amely feladatának tekinti az egyéb soros rendszerek, például a Purdue Europe kialakuló soros interface rendszerének és soros CAMAC rendszernek a közelítését. Ez a munka esetleg további módosításokat eredményez, hogy egyfajta minimális kompatibilitás biztosítható legyen.

Üzenetformátumok a SOROS FŐÚT-on:

A SOROS FŐÚT-on közlekedő üzenetek alapegysége a 8-bites byte. Byte-soros esetben minden óraütemben egy-egy byte-ot továbbíthatunk, míg bitsoros esetben egy byte továbbítása 11 óraütemet igényel. Itt ugyanis, mint ezt a 2.18. ábrán is láthatjuk,



2.18. ábra
Bitsoros átvitel formátuma

egy byte kiegészül egy start bittel és két stop bittel. Az egymás után küldött byte-okból végülis három különféle üzenet áll elő, az UTASÍTÁS, a VÁLASZ és az IGÉNY üzenet. A 2.5. táblázat ezt a három üzenetformát adja meg. Az üzenetekben jellegzetes információs mezőket találunk.

Ezek a következők:

- UTASÍTÁS-MEZŐ.** Tartalmazza a cím- és funkciókód-információkat.
- SC **Keretcím.** A SOROS FŐÚT-ra kapcsolt kereteket 6-bites bináris kóddal címezik. Az egyes bitek jelölése SC1, SC2, SC4...SC32. A nullás keretcím a soros meghajtó részére van fenntartva, a 63-as cím nem engedélyezett, így maximálisan 62 keretet lehet címezni.
- SN **Állomásszám.** 5-bites kód, jelentése megegyezik a CSOPORTÚT BN jelével. Az állomásszám bitek jelölése az utasításmezőben: SN1, SN2...SN16.
- SA **Alcím.** 4-bites kód, jelentése megegyezik az ADATÚT A jeleivel. Az alcím bitek jelölése: SA1, SA2, SA4, SA8.
- SF **Funkció.** 5-bites kód, jelentése megegyezik az ADATÚT F jeleivel. A funkciókód egyes bitjeinek jelölése: SF1, SF2, ... SF16.
- M1, M2 **Üzenetazonosító.** A három lehetséges üzenetformát jelöli. Ha mindkét bit 0, utasításüzenetről, ha M1=1, M2=0 válaszüzenetről, ha M2=1 (és M1 bármilyen értékű) igényüzenetről van szó.
- ÍRÁSMEZŐ.**
- SW **Írásbitek.** A soros meghajtóból a keretek felé továbbított adatot tartalmazzák, jelölésük: SW1...SW24. Az utasításüzenetbe helyezhetők el.
- OLVASÁS-MEZŐ.**
- SR **Olvasás bitek.** A keretektől a soros meghajtóhoz továbbított adatot az SR1...SR24 bitek tartalmazzák, a válaszüzenetbe helyezhetők el.
- VÁLASZMEZŐ.**
- SQ **Q-válasz.** 1-bites információ, jelentése megegyezik az ADATÚT Q válaszjelével.
- SX **X-válasz.** Az ADAÚT X jelének (utasítás elfogadva) felel meg. Mindkét bitet (SQ, SX) a válaszüzenet tartalmazza.
- IGÉNYMEZŐ.**
- SGL **LAM információs bitek.** A LAM forrás közelebbi megjelölésére szolgálnak az igényüzenetben. Néhány jellegzetes esetre külön is felhívjuk a figyelmet.
- Ha az SGL mezőt nem töltjük ki, az igényüzenet fejlécéből a soros keretvezérlő csak azt tudja megállapítani, hogy melyik keretben keletkezett LAM jel.
 - Az 5-bites SGL mezőben a keretvezérlő megadhatja azt az állomásszámot, amelyik állomásról az L jel érkezett. Így a soros meghajtónak nem kell külön azonosítania, hogy melyik modult kér kiszolgálást.
 - Az SGL bitek hordozhatnak valamilyen osztályozott LAM információt, ahol az egyes bitek az L1...L24 vezetékek jeleinek logikai kapcsolatából alakul ki. (Például: SGL1=L2; SGL2=L1+L3+L4; ... s í. t.)

• A csupa 1 SGL mező azt jelöli, hogy a keretben kiszolgáltatlan igény van (Hung-Demand).

A LAM információs bitek jelölése: SGL1, ...SGL5.

HIBAMEZŐ. A SOROS FŐÚT üzenetforgalma megnövelt hibavédelemmel van ellátva. Ebben a mezőben olyan hibajelzéseket találunk, amelyek az ADATÚT és CSOPORTÚT jelei közt nem fordultak elő.

P Paritásbit. Minden byte legnagyobb helyértékű bitje paritásbit, amely a többi 7 bitet páratlanra egészíti ki. Ezen „kereszt irányú” paritásbiteknek jelölése: P1...P8, ill. az üzenetek összeg byte-jában $P\Sigma$.

Σ MOD—2 összeg. Hosszirányú páros paritásbitek, amelyek úgy keletkeznek, hogy egy üzenethez tartozó byte-ok azonos helyértékű bitjeinek modulo—2 összegét képezzük, az első hat helyértéken. Jelölésük a hat helyértéknek megfelelően: $\Sigma 1, \Sigma 2, \dots \Sigma 6$.

A P és Σ paritásbitekkel mindhárom üzenetfajtát el kell látni.

ERR Hibabit. A válaszüzenet ERR bitjével a keretvezérlő azt jelzi a csoportmeghajtónak, hogy a kapott utasításban valamilyen hibát észlelt. Ilyen esetben egyidejűleg $SQ=0, SX=0$ választ kell adni, és az SR mezőt sem kell továbbítani.

DERR Késleltetett hibabit. Ezzel a bittel azt jelzi a keretvezérlő, hogy az aktuális utasítást megelőző utasítást hibásnak találta-e.

Tehát egy utasításban észlelt hibát az utasításhoz tartozó válaszüzenet ERR bitjében közvetlenül kell jelezni, majd a következő utasításhoz tartozó válaszüzenet DERR bitjében a hibajelzést meg kell ismételni.

Az eddig tárgyalt információs mezőket szinkronizáló bitek és byte-ok egészítik ki. Az üzenetek szinkronizálását a 7-es bit végzi. A 7-es bit minden üzenet tartama alatt 0, az üzenet utolsó byte-jában lesz 1 (határolóbit). Szinkronizáló byte-ok:

Szünébyte. A soros meghajtó generálja, miközben egy megcímzett keret választ várja, ill. fogadja. Tartalma: 1 0 1 1 1 1 1 1.

Várakozásbyte. A keretvezérlők és a sorosmeghajtó generálja az üzenetek közti időben. Tartalma: 1 1 1 0 0 0 0 0.

Összegbyte. Az utasításüzenet első, információt hordozó részének utolsó byte-ja. Az első hat bit hosszirányú paritásbit ($\Sigma 1 \dots \Sigma 6$), a hetedik 0 (határolóbit), a nyolcadik a keresztirányú paritásbit ($P\Sigma$).

Záróbyte. A soros meghajtó generálja az utasításüzenet végén, tartalma megegyezik a várakozásbyte tartalmával.

Záró-összegbyte. A válasz és igény típusú üzenetben az utolsó byte, a keretvezérlő generálja (ENDSUM Byte), tartalmát meghatározzák a $\Sigma 1 \dots \Sigma 6$ összegbitek, a határolóbit 1-es értéke és $P\Sigma$ paritásbit.

Az ismertetett információs mezők és szinkronbitek, ill. byte-ok felhasználásával a három üzenettípus a következőképpen állítható össze (2.5. táblázat):

Utasításüzenet

Az utasításüzenet első byte-ja a keretcímet (SC) tartalmazza, a második byte az utasításazonosító mezőt, összhangban az utasítással $M1=0$ és $M2=0$ értékkel, valamint az alcímet (SA). A harmadik byte-ban a funkciókód (SF), a negyedikben az állomászám található, mindkettőben a 6-os bit 1-es értékű (fenntartott bit). Ha az utasítás olvasás

vagy vezérlés típusú, az utasítás következő byte-ja a hosszirányú paritásbiteket (Σ) tartalmazó összegbyte, ha írás típusú, az összegbyte csak az írásmező (SW) után következik. A soros meghajtó az összegbyte után szünetbyte-okat generál, hogy a megcímzett keretvezérlőnek legyen ideje elküldeni a válaszüzenetet a meghajtóhoz. Az utasításüzenet utolsó byte-ja a záróbyte, amelyben a határolóbit logikai „1” értékű. Minden byte legnagyobb helyértékű bitjébe P paritásbit aktuális értéke kerül.

Válaszúzenet

A keretvezérlő, amely felismerte az utasítás fejlécében a saját címét, ADATÚT ciklust generál és válaszüzenetet köteles adni. A válaszúzenet első byte-ja a választ adó keret címét tartalmazza (SC), a második byte az utasításazonosítót $M1=1$, $M2=0$ értékkel, továbbá a válasz- (SQ, SX) és hiba- (ERR, DERR) mezőket. Írás és vezérlés típusú műveletnél a válaszúzenet következő byte-ja, a harmadik byte a záróösszegbyte.

Olvasás típusú műveletnél a záróösszegbyte csak a 4 byte-ból álló olvasásmező után következik. A záróösszegbyte a válaszúzenet végét jelzi, ebben vesz fel a határolóbit először 1-es értéket. A legnagyobb helyértékű bitekben itt is a keresztirányú paritásbiteket (P) találjuk.

Igényúzenet

Az igényúzenet biztosít lehetőséget a keretvezérlőnek arra, hogy a meghajtónak jelezze, a keretben LAM jel keletkezett. Az igényúzenet első byte-ja tartalmazza az igényúzenetet generáló keret címét, a második byte az üzenetazonosítót (ez esetben $M2=1$ -et), továbbá az igénymezőt (SGL). Az igényúzenet harmadik byte-ja, a záróösszeg byte egyben az üzenet végét is jelenti. A határolóbit az első két byte-ban logikai 0, az utolsóban logikai 1. A legnagyobb helyértékű bitben a paritásbitek ($P1$, $P2$, $P\Sigma$) találhatóak.

A SOROS FŐÚT működése

A SOROS FŐÚT-on végrehajtható műveletek két csoportra oszthatók. Vannak utasításműveletek, amelyek mindig egy utasításüzenet—válaszúzenet ciklusból állnak és vannak igénybejelentő (kiszolgálást kérő) műveletek, amelyek egyetlen igényúzenet továbbítását jelentik valamely keretvezérlőtől a soros meghajtó felé.

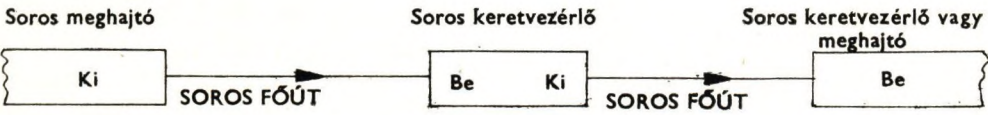
Utasításművelet. Az utasításműveleteknek a funkciókódtól (SF) függően három alaptípusa van: írás, olvasás és vezérlés. Példaképpen egy olvasás típusú utasításműveletet mutatunk be. A művelet lefutását a 2.6. táblázat segítségével követhetjük. A műveletet a soros meghajtó kezdeményezi azáltal, hogy kimenő csatlakozóján egy utasításüzenetet indít. Az utasításüzenet első byte-ja, a keretcím eljut az első keretvezérlőhöz. A keretvezérlő megvizsgálja a címet, és ha úgy találja, hogy öt címeztek és a $P1$ paritás is rendben volt, az első byte-ot még továbbadja a soros hurok többi kerete felé, de az utasításüzenet többi byte-ja helyett záróbyte-ot, majd várakozásbyte-okat továbbít. Tehát egy címzett keret soros keretvezérlője bemeneti csatlakozásán veszi az utasítást, miközben a kimeneti csatlakozóján várakozásbyte-okat küld a további keretek felé. Ha a keretvezérlő úgy találja, hogy nem öt címeztek, egyszerűen tovább adja a bemenetére érkező byte-okat.

Példánkban az első keretvezérlőt címeztük. Így ez a címét még megismétli, de az olvasásutasítás többi 4 byte-ja helyett 1 záró- és 3 várakozásbyte-ot ad tovább. A soros meghajtó a továbbiakban, míg a keretvezérlőtől a válasz meg nem érkezik, szünetbyte-okat ad. A keretvezérlő pedig megvizsgálja a kapott utasítást, nem hibás-e, majd ADATÚT ciklust indít. A ciklus végén előáll a kért információ, és miközben bemenetére érkeznek a szünetbyte-ok, megkezd a kimenő csatlakozóján a válaszüzenet kibocsátását. A soros hurok többi vezérlője, mivel a cím nem nekik szól, az üzenetet „átereszti” és az végülis eljut a soros meghajtó bemenetére.

(Ugyanígy végigfut az utasításüzenet fejléce, a keretcímbyte, valamint a záróbyte. A várakozásbyte-ok csak akkor jutnak el a soros meghajtó bemenetére, ha a közbeeső kereteknek nincs igényüzene.) A soros meghajtó tehát kimenetén szünet byte-okat

2.6. táblázat

SOROS FŐÚT utasítás—válasz ciklusa olvasás típusú művelet esetén



P1	0	SC32		-		-		-		-		-		SC1
P2	0	M2 0	M1 0	SA8		-		-		-		-		SA1
P3	0	1*	SF16		-		-		-		-		-	SF1
P4	0	1*	SN16		-		-		-		-		-	SN1
PΣ	0	Σ6		-		-		-		-		-		Σ1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

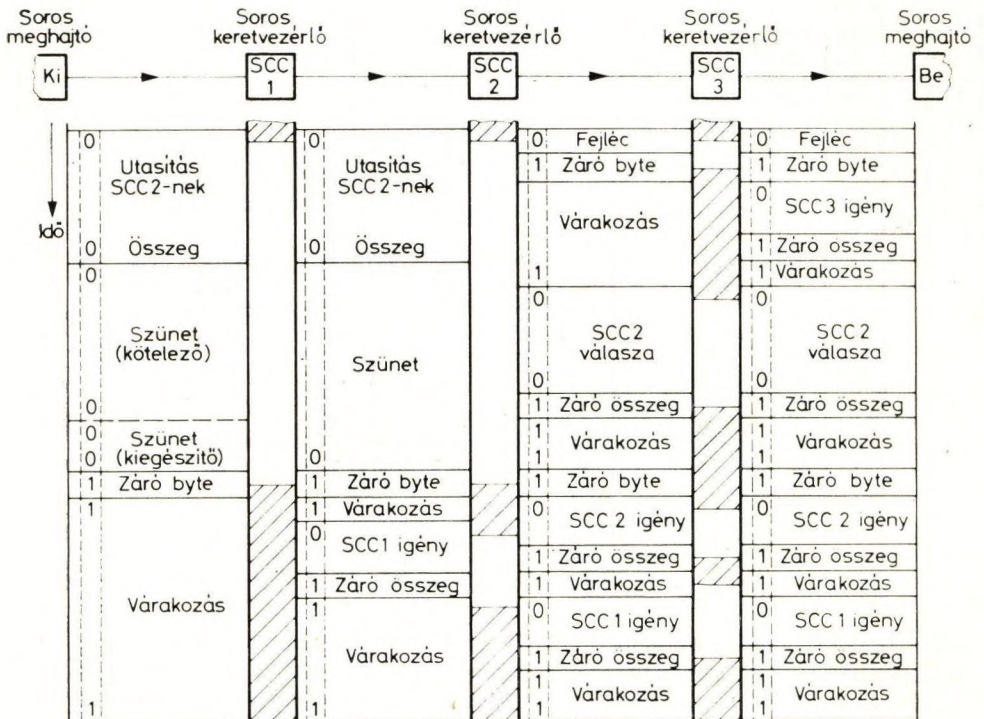
P1	0	SC32		-		-		-		-		-		SC1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
P1	0	SC32		-		-		-		-		-		SC1
P2	0	M2 0	M1 1	D ERR	SQ	SX	ERR							
P3	0	SR24		-		-		-		-		-		SR19
P4	0	SR18		-		-		-		-		-		SR13
P5	0	SR12		-		-		-		-		-		SR7
P6	0	SR6		-		-		-		-		-		SR1
PΣ	1	Σ6		-		-		-		-		-		Σ1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

küld a soros főútra, bemenetén pedig veszi a válaszüzenetet. A válaszüzenet 7-ik, záró-összeg byte-jának érzékelése után a meghajtó kiadja az utasításüzenet záróbyte-ját, és ezzel lezárja az utasítás—válasz ciklust, jelen esetben az olvasásműveletet. Vezérlés vagy írás típusú utasításműveletnél a válasz üzenet rövidebb, nem tartalmazza az olvasásművelet (SR): írásműveletnél viszont az utasításüzenet bővül az írásművelet (SW). Ezekről az eltérésektől eltekintve a vezérlés- és írásműveletek a példában leírt olvasáshoz hasonló módon mennek végbe.

Igénybejelentő (kiszolgálást kérő) művelet. Ez a művelet igényüzenet továbbítását jelenti a keretvezérlőtől, ahol a LAM jel keletkezett, a soros meghajtóhoz. Mivel a soros főúton jelforgalom van, úgy kell a 3 byte-os igényüzenetet a jelforgalomba beiktatni, hogy a többi üzenettel ne ütközzön. Ezért egy keretnek, amelyik igényüzenetet akar továbbítani, figyelnie kell az információáramlást, és igényüzenet továbbítását csak a főúton haladó üzenetek záróbyte-ja után, vagy üzenetek közt továbbított várakozásbyte-ok észlelése esetén szabad kezdeményezni. Egy igényüzenet továbbításához 3 byte-időre van szükség.

Előfordulhat viszont, hogy ezalatt a SOROS FŐÚT ismét foglalttá válik. Ezért a keretvezérlőnek bekapcsolható 3 byte késleltetéssel kell rendelkeznie. Így, ha a keretvezérlő egy utasítás végén elkezd az igényüzenet kibocsátását és egyidejűleg bekapcsolja a 3-byte-os késleltetést, az igényüzenetet biztosan be tudja fejezni, hisz a következő, főútról jövő byte a késleltetés miatt csak 3 byte-idő múlva „érkezik meg”. (A beiktatott késleltetést a keretvezérlő a későbbiekben akkor kapcsolhatja ki, ha az utolsó kiadott byte záróbyte és a késleltetőben mindhárom byte várakozásbyte.)

A 2.19-es ábrán háromkeretes soros CAMAC rendszer SOROS FŐÚT-jának információáramlási diagramját mutatjuk be, amely tartalmaz a 2-es keretnek címzett utasításműveletet, és mindhárom kerettől származó igényüzenetet. (A 2-es keret az igény üzenetének generálásakor 3 byte késleltetést alkalmazott.)



A vonalkázott területeknek megfelelő időben van engedélyezve igénybejelentés és válaszadás

2.19. ábra
Háromkeretes SOROS CAMAC rendszer információáramlási diagramja

A SOROS FŐÚT szabványos, D típusú csatlakoztatásának elektromos jelei sodrott érpáron közlekednek. A jelek földszimmetrikusak, a meghajtó differenciális áram generátor, a vevő differenciál-komparátor. A hasznos jel a generátor kimenetén kb. 0,5 V, a közös fázisú zavarjel maximuma ± 3 V. A vonal hullámellenállása kb. 100 Ω . A D típusú csatlakozók bekötését a 2.7-es táblázatban adtuk meg.

2.7. táblázat

A SOROS FŐÚT fogadására, ill. továbbítására szabványosított csatlakozók bekötése

Bemeneti csatlakozó		Kimeneti csatlakozó	
Érintkező	Jel	Érintkező	Jel
1—2.	bit-soros bemenet	1—2.	bit-soros kiment
3—4.	1. bit	3—4.	1. bit
5—6.	2. bit	5—6.	2. bit
7—8.	3. bit	7—8.	3. bit
9—10.	4. bit	9—10.	4. bit
11—12.	5. bit	11—12.	5. bit
13—14.	6. bit	13—14.	6. bit
15—16.	7. bit	15—16.	7. bit
17—18.	8. bit	17—18.	8. bit
19—20.	Fenntartott	19—20.	Fenntartott
21—22.	Órajel	21—22.	Órajel
23.	Kikerülés vezérlés	23.	Kikerülés vezérlés
24.	Kiiktatás vezérlés	24.	Kiiktatás vezérlés
25.	Áramköri föld	25.	Áramköri föld

Érintkezőpároknál a páratlan szám a jelvezeték, a páros szám a visszatérő vezeték jelöli.
A csatlakozók típusa: CANNON DBC-25P

Irodalom a 2. fejezethez

- [1] Addenda and Errata to CAMAC Serial System Organization. ESONE (SH) 02; ESONE (SH) 03 1974.
- [2] CAMAC — A Modular Instrumentation System for Data Handling — Description and Specifications. EUR 4100e; TID 25875. 1972. (1969).
- [3] CAMAC—Organization of Multicrate Systems. EUR 4600e; TID 25876. 1972.
- [4] CAMAC—Serial System Organization — A Description. ESONE (SH) 01; TID 26488. 1973.
- [5] CAMAC — Specification of Amplitude Analogue Signals within a 50 ohm System. EUR 5100e; TID 26614. 1974. (1972).
- [6] CAMAC — The Definition of IML. A Language for use in CAMAC Systems. ESONE (IML) 01; TID 26615. 1974/1975.
- [7] Costrell, L.: "Highways for CAMAC Systems — A Brief Introduction. IEEE Trans. on Nuclear Science. Vol. NS—21, No 1, Febr. 1974.
- [8] Programmed Dataway Controller Type 7025 and Program Plugboard Type 0361—2. NUCLEAR ENTERPRISES Ltd., Beenham, Reading RG7 5PR, England, 1971.
- [9] Schöberl, E.: CMC 8080: CAMAC Crate Controller with Microcomputer INTEL 8080. Second Symposium on CAMAC, Brussels, 1975.
- [10] Supplementary Information on CAMAC Instrumentation System. CAMAC Balletin No 6. TID 25877, 1973.

3.

CAMAC vezérlők

Az előző fejezetben áttekintettük a CAMAC rendszerek különböző jelvezetési útjait: az ADATÚT-at, a CSOPORTÚT-at és a SOROS FŐÚT-at. Ezen utak jelforgalmát vezérlik a különböző vezérlőegységek: a keretvezérlő (Crate Controller), csoportvezérlő (Branch Driver) és a soros vezérlő (Serial Controller).

A CAMAC szabványok előírják az adatforgalom jeleinek paramétereit, a mechanikai méreteket, tűréseket stb., de nem tartalmaznak megkötéseket arra vonatkozóan, hogy a modulokat vezérlő egységek, a különböző keret-, ill. csoportvezérlők mennyire bonyolultak, vagy milyen arány van a huzalozott és programozható funkciók között. A feladatoknak és a felhasználható alkatrészeknek megfelelően igen sokféle vezérlőt készítettek. Legáltalánosabban elterjedtek az egy-egy számítógéphez kapcsolódó vezérlők, kisebb feladatokra azonban már régóta használnak fix, kézzel változtatható programú önálló vezérlőket.

Az integrált áramkörök fejlődése és a mikroprogramozási technika elterjedése újabb lendületet adott a vezérlők tervezőinek: egyre több intelligenciát kezdenek beépíteni a bonyolultabb vezérlőkbe. A mérésadatgyűjtés, folyamatszabályozás korszerű rendszertechnikai elképzelései is ebbe az irányba mutatnak: a központi számítógéptől távolabb levő, alacsonyabb szintek is minél nagyobb önállósággal rendelkezzenek. Erre két okból is szükség van: egyrészt így mód nyílik az adatok redukciójára, másrészt pedig a központi szerverek hibája esetén is még működőképes a helyi rendszer. Az újabb alkatrészek, ill. adta lehetőségek alapján néhány helyen különleges célokra egyedi vezérlőket kezdtek tervezni, amelyek így gazdaságosabban és jobban el tudják látni a speciális feladatokat.

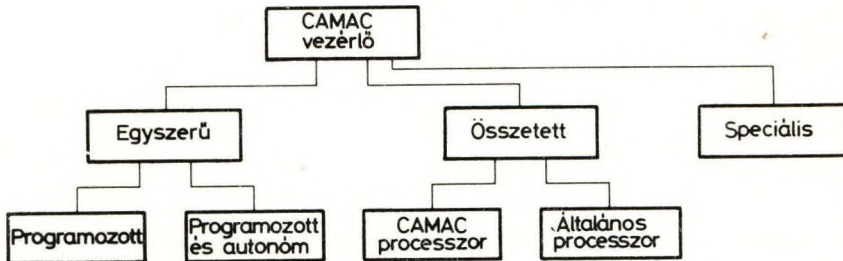
A vezérlőket igen sokféleképp lehet csoportosítani. Az áttekinthetőség miatt elsősorban *funkcionális specifikáció* szerint tárgyaljuk ezeket és alkalmanként mutatunk rá a hasonló funkciójú, de különböző megvalósítású vezérlők közti eltérésekre.

A CAMAC vezérlők funkcionális áttekintését a 3.1. ábra mutatja be.

Egyszerű vezérlőknek nevezzük azokat a vezérlőket, amelyeknél:

1. közvetlenül számítógép irányítja az adatátvitelt, ill. a CAMAC modulok vezérlését;
2. egy-egy számítógép-utasítás (esetleg utasítássor) csak egy-egy CAMAC funkciót hajt végre egyszer (programozott vezérlő) vagy többször (autonóm működésű, egyszerű vezérlő).

A programozott, egyszerű vezérlő esetén egy-egy CAMAC utasítást a számítógép meghatározott utasításaival kell előkészíteni és közvetlen gépi utasítással (amely egyben előkészítő utasítás is lehet) indítani. Az autonóm működésű, egyszerű vezérlők esetén valamilyen — programozott módon előkészített — CAMAC utasítást ismét a vezérlő (pl. előírt számsor vagy sikeres Q válasz) és az adatforgalmat a számítógép autonóm csatornáján bonyolítja. Mivel a CAMAC utasítássor előkészítése a programozott csatornán történik, ezért autonóm vezérlők önállóan nem léteznek.



3.1. ábra
CAMAC vezérlők áttekintése

Az összetett vezérlők több különböző CAMAC utasításból álló utasítássort tudnak végrehajtani vagy önállóan, számítógéptől függetlenül, vagy pedig egy központi számítógép programjával összhangban. Az összetett vezérlők önálló utasításkészlettel rendelkeznek. Ez az utasításkészlet vagy csak CAMAC utasításokat tartalmaz (esetleg néhány egyéb utasítással kiegészítve), vagy általános számítógép utasításkészletet. Az összetett vezérlőt első esetben CAMAC processzornak, az utóbbiban általános processzornak nevezzük.

A speciális vezérlők egyedi feladatok megoldására készültek, és olyan utasításokat, ill. hardware elemeket tartalmaznak, amelyek beépítésekor gazdaságossági szempontokat, esetleg egyéb specifikációs megkötöttségeket (pl. sebesség) vettek figyelembe.

A felsorolt vezérlőegységek készülhetnek akár úgy, hogy a CSOPORTÚT-at, akár úgy, hogy az ADATÚT-at vezérlik. Ha a CSOPORTÚT-at vezérlik, akkor a CSOPORTÚT és az ADATÚT illesztésére minden keretben ún. A1 típusú keretvezérlőt (l. 3.1.2. pont) kell alkalmazni. Az ADATÚT-at közvetlenül a számítógép ki-, ill. bemeneti csatornájával összekötő vezérlőt U típusú, gépreszabott kombinált keretvezérlőnek nevezik.

3.1. Egyszerű vezérlők

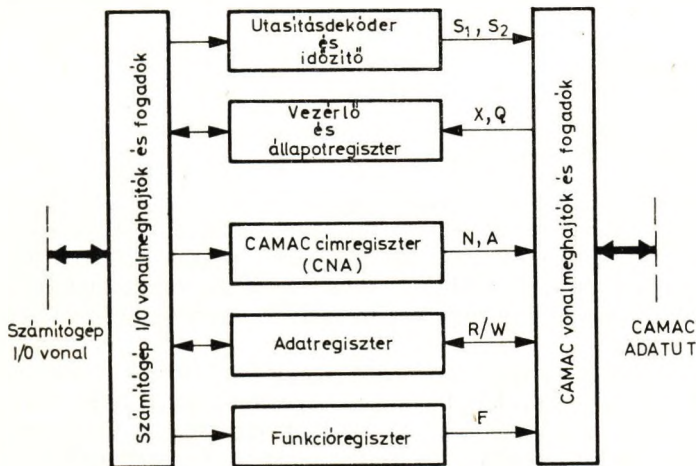
Az egyszerű vezérlő lényegében a számítógép saját ki-, ill. bemeneti buszrendszerét illeszti a CAMAC szabványú CSOPORTÚT, ADATÚT vagy SOROS FŐÚT-hoz. Egyszerű programozott vezérlő esetében a számítógépnek csak a programozott csatornához kapcsolódik, autonóm működésű esetben a számítógép autonóm (közvetlen tárolóelérésű) csatornához is.

3.1.1.

Egyszerű programozott vezérlők

Az egyszerű programozott vezérlők esetén egy számítógép-utasítással (vagy utasítás-sorral) egy CAMAC utasítás — tehát pl. egy adatszó átvitele — hajtódik végre. E vezérlők funkciója ezért nagyon egyszerű: időzítési, elektromos jelszintbeli és logikai illesztése a számítógép saját és a CAMAC szabványú buszrendszerei között.

A 3.2. ábrán egy egyszerű programozott keretvezérlő általános vázlatát mutatjuk be. A programozott csoportvezérlő vázlata majdnem teljesen megegyezik a keretvezérlőével, csak a CAMAC oldali vonalmeghajtók és fogadók az ADATÚT helyett a CSOPORTÚT-nak megfelelő jeleket kezelik.



3.2. ábra
Programozott keretvezérlő általános felépítése

Az elektromos jelszintek illesztését a számítógép és az ADATÚT (CSOPORTÚT) felőli oldalon levő vonalmeghajtók és fogadók végzik. Megoldási módjukat a számítógép-buszvonal belső szabványai, ill. a CAMAC szabványok rögzítik.

Az időzítések összehangolására gyakran kell időzítő áramkört építeni a keretvezérlőbe. Általában valamely vonalon levő jelek időzítése elvileg kétféle lehet:

a) Az adóoldal felől küldött, rögzített időtartamú és időközű mintavevő impulzusok jelentik, hogy valamilyen érvényes információ van a vonalon.

b) A „kézfogás” (hand-shaking) elvű időzítés esetén az adó által kiadott jel addig marad a vonalon, amíg a vevő nyugtázójele vissza nem jut az adóba.

Az ADATÚT az első elven működik, a CSOPORTÚT a másodikon. A különböző számítógép-típusok is vagy az egyik, vagy a másik megoldást használják. Ha azonos típusú a számítógép és a CAMAC út időzítési rendszere (pl. PDP-11 számítógép és CAMAC CSOPORTÚT), az időzítések összehangolása nagyon egyszerű.

A logikai illesztés a számítógép és a CAMAC rendszer között általában a következők miatt szükséges:

- különböző a szóhosszúság;
- különböző a címzési mód;
- különböző az utasításkód.

A CAMAC szabványok szerint az adatszó 24-bites, a cím (C, N, A) 12-bites, a funkció 5-bites. A számítógépek — főleg a kisgépek — ki-, ill. bemeneti rendszerének szóhosszúságai általában rövidebbek; az adatszó pl. 8-, 12- vagy gyakran 16-bites. A címezési lehetőségek sem engednek meg néhány bitnél többet a real-time perifériák céljára. E nehézségeket úgy küszöbölik ki, hogy pufferregisztereket helyeznek el a vezérlőben a CAMAC cím, a funkció és az adat időleges tárolására. A CAMAC cím-regiszter 12-bites: a keretet (C) 3 bit, az állomást (N) a kereten belül 5 bit és az állomáshelyre helyezett modulban levő alcímet (A) 4 bit határozza meg. A funkció-regiszter 5-bites. Az adatregiszter — amennyiben az ADATÚT teljes szélességét ki akarjuk használni — 24-bites. E pufferregisztereket kell a számítógép ki-, ill. bemeneti utasításaival betölteni, majd a CAMAC rendszer felé indítójelet kiadni. A számítógépnek ily módon írás típusú CAMAC utasítás végzéséhez a következő utasításokat kell kiadnia:

- CAMAC cím beírás,
- funkcióregiszter beírás,
- adatregiszter beírás,
- CAMAC ciklus (pl. ADATÚT ciklus) indítás.

A tényleges gépi utasítások száma természetesen függ a számítógép ki-, ill. bemeneti rendszerének szóhosszától. Így pl. esetleg össze lehet vonni a cím- és funkcióbeírást egy utasítássá, ugyanakkor viszont a 24-bites adatregisztert általában csak két utasítással lehet beírni vagy kiolvasni. A CAMAC ciklus legtöbbször automatikusan indul az adatregiszter (írás) vagy funkcióregiszter (olvasás és vezérlőutasítás) beírásával.

A CAMAC ciklus folyamán a keretvezérlő állapotregiszterébe beíródik a megcímzett modulból származó X és Q válasz, továbbá az esetleges hibát jelző bitek. Az X válasz azt jelzi, hogy a modul „megértette-e” az utasítást vagy sem, a Q válaszjel pedig a modul leírásában definiált üzenet. Ilyen üzenet lehet például az, hogy $Q=1$ válasz esetében az átadott adat érvényes, $Q=0$ esetében érvénytelen. Az X válasz hiánya tehát minden esetben hibát jelez, ezért gyakran a számítógép megszakításkérő vezetékére továbbítják megszakítást kérő jelként. A Q választ általában a program-elágazáshoz is használják. Igen sok esetben a CAMAC modult vezérlő utasítássor végén a számítógépprogram megvizsgálja a Q állapotregisztert és a tartalmától függően különböző helyen folytatja a programot. Például, ha a $Q=1$ válasz érvényes adatátvitelt jelentett, akkor ilyenkor tárolja a beolvasott adatot a tárolóba, ha $Q=0$, akkor újra kezdeményezi a beolvasást. Q vizsgálattal kiegészítve egy írás típusú CAMAC utasításhoz a következők szükségesek:

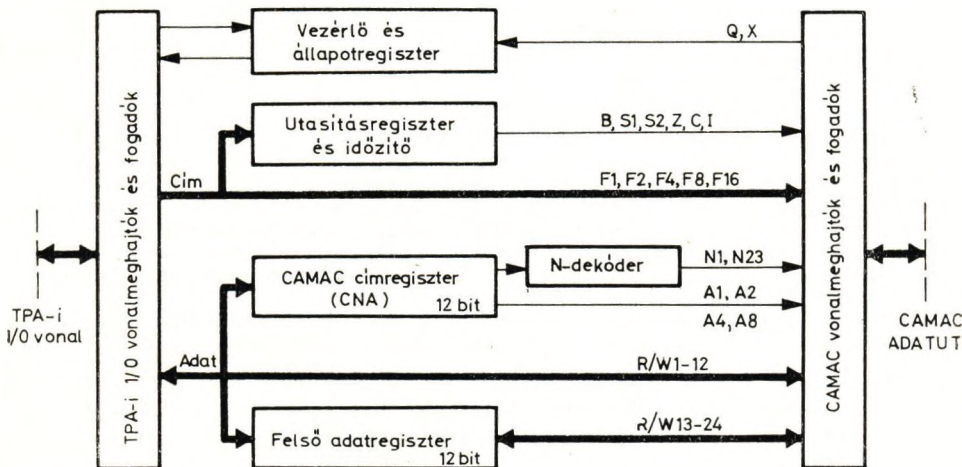
- CAMAC cím beírás;
- funkcióregiszter beírás;
- adatregiszter beírás;
- CAMAC ciklus indítás;
- Q vizsgálat.

A jelenlegi kisszámítógépekkel a fenti utasítássor végrehajtása 10...20 μ s körül van és 4...8 utasításból áll. A számítógép perifériás készletéből csak a pufferregiszterek foglalnak le egy-egy címet, ezért 4...5 perifériás cím elegendő lehet a teljes CAMAC rendszer kiszolgálására.

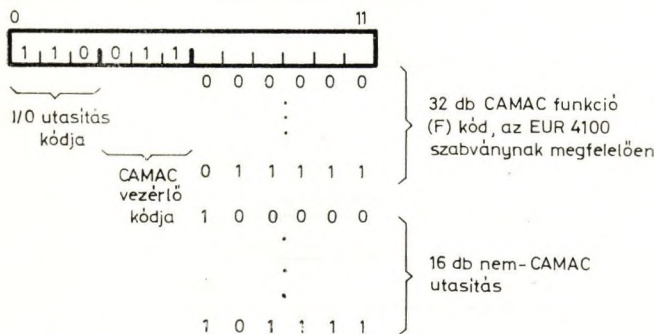
Gyorsabban és gazdaságosabban működő vezérlőket készítettek úgy, hogy nemcsak 4—5-perifériás címet használtak fel a számítógép címkészletéből, hanem ennél jóval többet: 6...10 bitnek megfelelő perifériás címet. Ezek közül 2—3 bit jelenti a

tényleges címet, míg a többi bitet a CAMAC cím vagy funkció közvetítésére lehet használni.

Ilyen megoldású vezérlő a PDP 8-hoz hasonló TPA-i kisszámítógép programozott keretvezérlője. A TPA-i perifériáinak címzésére 9 bit áll rendelkezésre. Egy programozott adatátvitellel a 12 bites akkumulátor tartalma vihető át a perifériába, ill. a perifériából egy 12-bites szó írható be az akkumulátorba. A TPA-i — miután a címet és az adatot tartalmazó akkumulátor jeleit kiadta a perifériás vonalra — indítójelet küld a perifériának. A periféria a TPA-i két vezérlővonalához fér hozzá adatátvitel közben: ezekkel törölheti az akkumulátort és utasításátlépést okozhat a TPA-i-ben. A TPA-i keretvezérlő egyszerűsített blokkvázlatát a 3.3. ábrán láthatjuk.



3.3. ábra
TPA-i keretvezérlő egyszerűsített blokkvázlata



3.4. ábra
A TPA-i számítógépnek a CAMAC keretvezérlőre vonatkozó utasításai

A megoldás lényege az, hogy a TPA-i perifériás címei közül 48-at használnak fel. A keretvezérlőnek adható utasítások két csoportra oszthatók: CAMAC utasításokra és nem CAMAC utasításokra (3.4. ábra).

A CAMAC utasítások esetén a TPA-i periféria címe tartalmazza a funkció- (F) kódot. Mivel az EUR 4100 szabványnak megfelelően 32 különböző F kód lehet, a CAMAC utasítások 32 perifériacímet foglalnak le. (Az F kódra vonatkozó szabványokat a 2. fejezetben, ill. a függelék 7.1. táblázatában talál-

juk meg részletesen.) Egy-egy CAMAC típusú TPA-i utasítás végrehajtása során a következők történnek:

1. A CAMAC címregiszter tartalma (N, A1—A8) kijut az ADATÚT-ra.
2. A TPA-i periféria címében levő F kód az ADATÚT-ra kapuzódik.
3. A CAMAC (ADATÚT) ciklus (B, S1 és S2 jelek) lezajlik.
4. X=0 válasz esetén programmegszakítás történik a TPA-i-ben.
5. Q=1 válasz esetén utasításátlépésre kerül sor a TPA-i-ben.

A nem CAMAC utasítások nem indítanak CAMAC ciklust, hanem lehetővé teszik, hogy a TPA-i beírjon és kiolvasson a keretvezérlő regisztereibe. Ugyancsak ezekkel az utasításokkal lehet kezdeményezni a CAMAC rendszer speciális vezérlőjeleit, amelyek a következők: I=Tiltás, C=Törlés és Z=Alaphelyzet.

Bár a TPA-i adatvonala 12-bitese, a változt keretvezérlővel három TPA-i perifériás utasítással beírható egy 24-bitese szó valamely CAMAC modulba:

- CAMAC címregiszter beírás;
- felső adatregiszter beírás;
- Írás CAMAC utasítás az adatszó alsó, kis helyértékű részével.

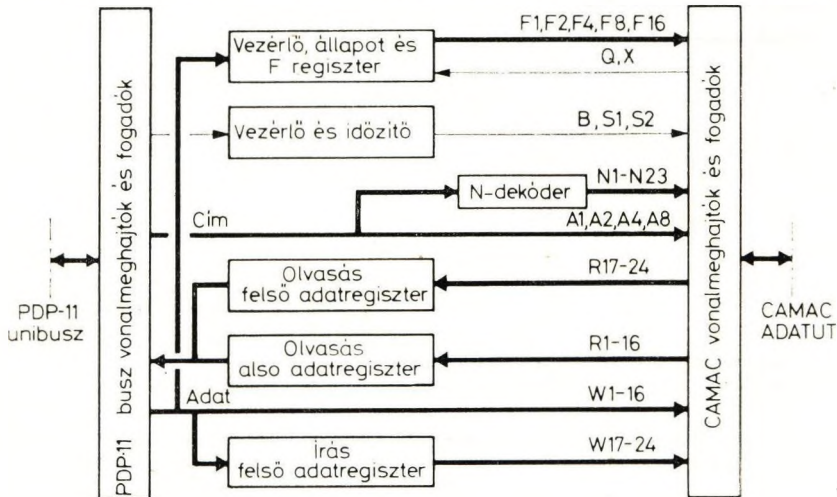
Ha a CAMAC cím változtatlan és a CAMAC modulba csak 12-bitese szót kívánunk beírni, akkor egy TPA-i utasításra van szükség.

Olvasás típusú CAMAC utasítás esetén a helyzet még egyszerűbb. Ilyenkor ugyanis nem kell a TPA-i-ből a CAMAC rendszer felé adatot átvinni, elég csak a címet és a funkciót. Az Olvasás típusú utasítások első lépése ezért azzal bővül, hogy a TPA-i akkumulátorában ilyenkor adatként tárolt CNA cím beíródik a keretvezérlő CNA regiszterébe. Ezután a CNA regisztertartalma kijut az ADATÚT-ra, stb. A CAMAC modulból olvasott adat felső 12 bitje a felső adatregiszterbe kerül, alsó 12 bitje közvetlenül a TPA-i akkumulátorába.

Megjegyezzük, hogy a változt TPA-i keretvezérlőnek itt csak az adatforgalmát mutattuk be. A LAM (megszakítás) kérést és néhány egyéb tulajdonságot a későbbiekben fogjuk tárgyalni.

Olyan kisszámítógépek esetén, amelyek perifériás címkészlete igen nagy, még jobban használhatók a sok cím adta lehetőségek. Ilyen gépcsalád a PDP—11, amely 12 bitet tart fenn a perifériák címzésére, sőt, a tárolóbővítés rovására még ennél is több perifériás cím használható.

Példaként bemutatunk egy PDP—11 keretvezérlő megoldást. A 3.5. ábrán a keretvezérlő egyszerűsített, csak az adatforgalmat tartalmazó blokkvázlata látható. A megoldás lényege, hogy a CAMAC cím közvetlenül, átmeneti tárolás nélkül jut ki az ADATÚT-ra, mivel a PDP—11 perifériacíme tartalmazza a CAMAC címet. Ily módon a keretvezérlő 9 bitet (512 perifériacímet) használ fel az UNIBUSZ



3.5. ábra
PDP—11 keretvezérlő egyszerűsített blokkvázlata

címekből. A funkció- (F) kódot a vezérlő- és állapotregiszter tartalmazza, amit külön utasítással lehet beírni. A PDP—11 és a CAMAC rendszer szóhossz-különbségének áthidalására íráskor és olvasáskor is egy-egy 8-bites felsőadatregisztert alkalmazunk. Olvasáskor az adatszó alsó része is ideiglenesen tárolódik azért, hogy az F(2) funkciójú (Olvasás és Törlés) utasításnál a CAMAC regiszterből kiolvasott adat kellő ideig a PDP—11 bemenetén legyen. A vázolt megoldású keretvezérlővel 16-bites szavak írása két PDP—11 utasítással történik. (Vezérlő- és állapotregiszter betöltés, Adatbeírás.) Az adatátvitel nélküli (vezérlő) CAMAC utasítások végrehajtásához egy PDP—11 utasítás szükséges; ilyenkor ugyanis az adatszó tartalmazza a F funkciókódot, amely közvetlenül átjut a CAMAC ADATÚT-ra.

A bemutatott keretvezérlőkön kívül igen sokféle különböző kialakítású vezérlőt készítettek. A következőkben néhány olyan érdekes megoldást mutatunk be, amelyek valamilyen speciális alkalmazási területen előnyt jelenthetnek.

Az egyik megoldás [8] a CAMAC cím mellé két további címmódosító bitet rendel. Ennél a keretvezérlőnél a CAMAC címet a vezérlőben levő címregiszter tartalmazza. A címmódosító bitek állapotától függően a CAMAC cím a ciklus végén automatikusan eggyel nő (autoindexelés). Lehet növelni az alcímet (A) vagy az állomásszámot (N), esetleg a Q választól függően. Ez a tulajdonság akkor kedvező, ha egymás után következő regiszterekből akarunk adatokat átvinni.

Más érdekes megoldás az ADATÚT ciklus gyorsítása. Sok CAMAC művelet csak az S1 vezérlőjelet használja fel, az S2 jelet nem. Van olyan keretvezérlő [11], amelynél a vezérlőregiszter tartalmától függően az S2 vezérlőjel elhagyható.

Az előbbi két, részletesebben bemutatott keretvezérlő példa egyikében a funkciót, a másikban a CAMAC címet, közvetlenül a számítógép perifériacíme tartalmazta. Kidolgoztak olyan megoldást [5], amely mindkét változatot a felhasználó rendelkezésére bocsátja: a vezérlőben levő átkötés helyzetével választhatunk a kétféle lehetőség közül.

Több keretvezérlő rendelkezik azokkal a tulajdonságokkal, amelyeket az EUR 4600 szabvány ír elő az A1 típusú keretvezérlők számára. Ezeket a tulajdonságokat részletesebben a 3.1.3. pontban tárgyaljuk.

Az egyszerű programozott vezérlőkkel kapcsolatban felhívjuk a figyelmet néhány problémára. Mint láttuk, egy-egy CAMAC utasítás általában előkészítő (cím-, adatstb. regiszterek betöltése), ill. befejező (felső adatszó átvétele, Q vizsgálat) műveletekkel jár együtt. Az ezeket megvalósító gépi utasítássor végrehajtásának megszakításakor gondoskodni kell a vezérlőben levő regiszterek tartalmának mentéséről, ill. újrabebiztosításáról, ha a megszakító program szintén használja a CAMAC rendszert. Kielégítő megoldás az is, ha az összetartozó CAMAC-kal foglalkozó utasítássor megszakíthatóságát letiltjuk.

További problémák az időzítésekből adódnak. A CSOPORTÚT adatforgalma — mint láttuk — kézfogas rendszerű. Így tehát a vonal hosszától függ egy-egy CAMAC utasítás végrehajtásának ideje. Ez az idő néhány μ s is lehet. Ha a számítógéphez olyan gyors periféria is kötve van, amelynél az adatátvitel-kérés várakozási ideje kritikus (pl. mágneslemez), az átvitelben hiba keletkezhet. Ha ui. már elkezdődött a CAMAC utasítás végrehajtása, a gyors perifériát a főgép általában nem tudja kiszolgálni. A probléma megoldására több lehetőség van. Legegyszerűbb, de igen rossz hatásfokú ha a gyors periféria működése alatt a CAMAC utasítások kiadását letiltjuk. Más megoldás szerint a CAMAC vezérlő olyan kialakítású, hogy a CAMAC utasítások kiadásakor a vezérlő nem várja meg a választ a CAMAC rendszertől, hanem befejezi az utasítást; a CAMAC ciklus befejezésekor pedig jelzőbittel jelzi a számítógépnek, hogy befejezte a működést.

3.1.2.

Autonóm működésű (autonóm adatátvitelt használó) egyszerű vezérlők

Az előző pontban láttuk, hogy egyszerű programozott vezérlővel egy-egy szó átvitele a CAMAC rendszer és a számítógép között 5...20 μ s-ig tart. Ez a sebesség — a gyakorlati tapasztalatok szerint — a legtöbb esetben elég. Az Egyesült Államokban végzett vizsgálatok szerint [13] a CAMAC rendszer sebessége — az ipari alkalmazások szempontjából — még a mai átlagnál körülbelül egy nagyságrenddel gyorsabb számítógép kiszolgálására is megfelelő. A laboratóriumi alkalmazások nagyobb részénél is megfelel a programozott vezérlő sebessége. Néhány alkalmazási területen azonban szükség van nagyobb sebességre, ill. arra, hogy a számítógép idejét ne kösse le a CAMAC rendszer mozgatása, s adatok átvitele ilyen hosszú ideig.

A következőkben két példát mutatunk be. Az elsőnél az adatátvitel nagy sebessége, a másodiknál a számítógép idejének tehermentesítése indokolja az autonóm működésű vezérlő használatát.

Nukleáris mérés technikában elég gyakori a számlálási veszteségek, ill. a mérési eredmények torzulásának (holt idő) csökkentése miatt, hogy egy analóg—digitális konvertert vagy több számlálót igen gyorsan kell kiolvasni. Itt szükség lehet 1 μ s körüli adatátviteli időkre, amit csak autonóm működésű vezérlővel lehet elérni.

Néhány egyszerű rendszerben a CAMAC keretben helyezkedik el egy oszcilloszkópcsőves megjelenítő, amely önálló tárolót nem tartalmaz, a felfrissítés a számítógép tárolójából történik. Ezeknél a programozott átvitel azért nehézkes, mert a megjelenítés mellett a gép még valamilyen főfeladatot is ellát. Ezt a főfeladatot kell időről időre megszakítani és megszakítási programot kezdeményezni. Ilyen rendszerekben a közvetlen tárolóelérésű autonóm vezérlők jelentősen csökkentik a megszakításnál szükséges adminisztrálással kapcsolatos számítógépidőt.

Az autonóm adatátvitelt használó vezérlő funkcionálisan hasonlóan működik, mint a számítógép autonóm adatátviteli csatornájára kapcsolt többi, szokásos perifériális berendezés. Ilyen vezérlő vázlatra látható a 3.6. ábrán. A működés három részből áll:

- az átvitelhez szükséges paraméterek és adatok betöltése a vezérlőbe;
- az autonóm átvitel;
- befejező műveletek.

Az átvitelhez szükséges adatok a szokásos berendezéseknél a memóriacím és az átvendő szavak száma. A memóriacím a memóriának arra a szavára mutat, ahová vagy ahonnan az első adatot át kell vinni. A memóriacímet a vezérlőegység általában számlálóregiszterben tárolja, ugyanis minden átvitt adat után ez a cím automatikusan eggyel nő. Így az átvitel során a főgép memóriájában egy folyamatos mezőbe kerülnek az átvendő adatok. Az átvendő szavak száma szintén számlálóregiszterbe kerül (szószámregiszter). E regiszter tartalma minden átvitt adat után eggyel csökken. Amikor az utolsó adatátvitel is befejeződött, a regiszter tartalma zérus, és a vezérlő jelzést kap a befejezésről. Ekkor kezdődhetnek el a befejező műveletek, amit általában a számítógépprogram végez el. Ilyen befejező művelet pl. az átvitel helyességének megállapítása, állapotregisztertartalom vizsgálat stb.

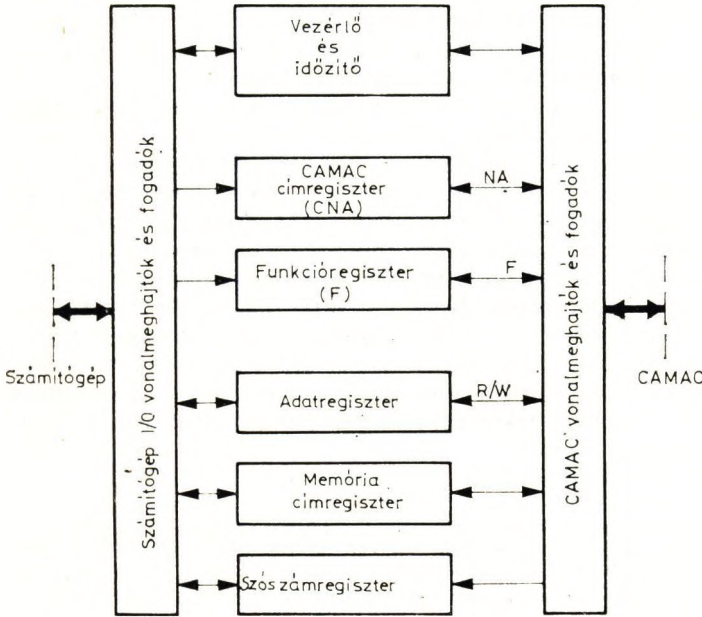
A CAMAC autonóm vezérlő a fentiekén kívül CAMAC címregisztert (CNA) és funkcióregisztert (F) is tartalmaz, ezeket is be kell írni az átvitel előkészítésekor. A funkcióregiszter tartalma a műveletek során nem változik. A CNA regiszter tartalma viszont az autonóm működés közben változhat.

Az autonóm vezérlő a CAMAC cím szempontjából alapvetően kétféle módon működhet:

- állandó címmel, ekkor ugyanarról a CAMAC címről (címre) történik az átvitel egymásután következő tárolócímekre (címekről). Így működik például egy lyukszalagolvasót illesztő modul. Ezt egyeszközű blokkátvitelnek is nevezik.

- Változó címmel, amikor egymás után következő alcímekről (címeke), ill. állomáshelyekről (helyekre) történik az átvitel, egymás után következő tároló-címekekre (címekekről). Így lehet kiolvasni például egymás mellett levő számlálómodulok tartalmát. Ennek az üzemmódnak az elnevezése többszörös átvitel.

Az egy eszközzel, ill. CAMAC címmel végrehajtott blokkátvitel során a CNA regiszter tartalma változatlan, több eszközre vonatkozó utasítás során viszont változik. Ilyenkor általában a szabványokban meghatározott címletapogatás (Address Scan)



3.6. ábra
Autonóm adatátvitelt használó vezérlő funkcionális vázlata

üzemmódot alkalmazták. Ez azt jelenti, hogy a vezérlő minden átvitel után megvizsgálja a Q választ. Ha $Q=1$, az átvitelt érvényesnek fogadja el, és az alcímet (A) megnöveli eggyel. Ha $Q=0$ választ kap, az átvitelt érvénytelennek tekinti (pl. nem tárolja be a memóriába), az állomászámot (N) növeli eggyel és az alcímet (A) nullába állítja. Ezzel az automatikus címnöveléssel egymás mellett levő CAMAC regiszterek tartalmát lehet autonóm módon átvinni a számítógépmemóriába.

A több eszközre vonatkozó művelet nemcsak kiolvasás vagy beírás jellegű lehet, hanem vezérlő művelet is. Ilyenkor természetesen a memóriacím-regisztert nem kell betölteni, mivel nincs adatátvitel a memória és a CAMAC perifériák között. Vezérlő jellegű művelet például regiszterek törlése, alapállapotba hozás, stb.

A vezérlő előkészítésekor a regiszterek betöltését többféleképpen lehet megvalósítani. Autonóm keretvezérlők esetében a leggyakoribb megoldás az, hogy a regiszterek betöltését a CAMAC ADATÚT-on keresztül végezzük. Ilyenkor az autonóm vezérlőben levő regisztereknek CAMAC címük van (a modulon belül A alcímük), és a programozott vezérlőn keresztül programozott CAMAC utasításokkal lehet a vezérlőben levő regisztereket betölteni. E megoldásnak az az előnye, hogy több autonóm vezérlő is kapcsolódhat ugyanarra az ADATÚT-ra, ill. ugyanahhoz a számítógéphez,

címük az állomásszámban (N) különböztethető meg. Másik megoldás szerint a vezérlőben levő regiszterek a számítógép I/O buszrendszer címzési módjának megfelelően közvetlen perifériacímeket kapnak, és közvetlen számítógép I/O művelet végzi e regiszterek betöltését.

Az előkészítő regiszterbetöltések és valamilyen indítójel kiadása után megkezdődik a tényleges autonóm adatátvitel, ill. autonóm működés.

A gyakorlati megvalósítások általában a vázolt elveket követik, ezekből az igényeknek megfelelően kiválasztott üzemmódokat építik be a vezérlőkbe. Néhány megoldásnál a programozott és az autonóm működésű egyszerű vezérlőt egyetlen elválaszthatatlan mechanikai egységbe építik. Gyakrabban azonban a moduláris felépítés: a programozott, az autonóm és esetleg a — később részletesen tárgyalandó — LAM kezelést külön-külön modulokban valósítják meg, természetesen összehangolt működés szerint.

Az előbbieken említett üzemmódokon kívül szinte minden autonóm vezérlő működhet inkrementáló (+1) üzemmódban is. Ez az üzemmód azt jelenti, hogy a CAMAC modulból kinyomtatott adat nem kerül át közvetlenül a számítógépmemóriába, hanem az adatot memóriacímként használjuk, és ezen a címen levő számhoz adunk hozzá egyet. Néhány kisszámítógép (pl. PDP-8, TPA-i, Mitra 15) olyan felépítésű, hogy ez az üzemmód könnyen alkalmazható: a számítógépnek adott speciális vezérlőjel hatására automatikusan ez az inkrementáló üzemmód kerül végrehajtásra. Más gépek esetében erre közvetlenül nincs lehetőség, ilyenkor a vezérlőnek kell előbb kiolvasnia az adott címen levő tartalmat a memóriából, megnövelni, majd visszaírni.

Nem ennyire általános, de több vezérlőben megtalálható a memóriatartalomhoz való hozzáadás. Ez az egyszerű memóriába való beírás művelet kissé bonyolultabb esete: az eredeti memóriatartalom ilyenkor nemvész el, hanem ahhoz adódik hozzá a CAMAC rendszerből kiolvasott adat.

Példaképpen tekintsünk át néhány autonóm keretvezérlőt.

A SEN cég a NOVA gépekhez készített autonóm keretvezérlőt. Ez egybe van építve a programozott vezérlővel. Az átvitel szempontjából két fő üzemmóda van, ezek átkötéssel választhatók; az egyik egyszerű kétirányú adatforgalom, a másik kimenő adatforgalom és memória inkrementáló üzemmód. CAMAC cím szempontjából is kétféle üzemmóda van: CAMAC cím változatlan, ill. a CAMAC cím letapogatással (Address Scan Mode) növekszik. E két üzemmód indítás előtt programmal választható. Az egyes adatok átvitelét a CAMAC modulok kezdeményezhetik; a kezdeményező jel az előlapon levő csatlakozón keresztül jut be a vezérlőbe.

Egy másik megoldás [10] szerint a PDP-11 számítógéphez programozott vezérlő, autonóm vezérlő és LAM kezelő modul különálló, összehangolt családokat alkot. Három üzemmóda van: állandó CAMAC címről blokkátvitel, memória inkrementálás és memóriához hozzáadás. Az átvitelt LAM jel kezdeményezi. Ez a megoldás elég széles körben használatos, mivel a CAMAC címnövelés üzemmódot — az összes alkalmazási területet figyelembevéve — viszonylag ritkán lehet kihasználni.

További megoldás [12] elsősorban gyors nukleáris mérés technikai célokra készült, PDP-8/E típusú számítógéphez. Az autonóm vezérlő egybe van építve a programozott keretvezérlővel. A vezérlő címletapogatással működik, állandó című blokkátvitel üzemmóda nincs. Ezen kívül memória inkrementálás és memóriához hozzáadás üzemmódjai vannak. Az átvitelt a LAM jel vezérli, átkötéssel választható ki, hogy melyik LAM jel indítja az autonóm átvitelt.

Az autonóm keretvezérlők — még akkor is, ha a programozott vezérlővel közös mechanikai egységet alkotnak — a programozott vezérlőkkel időosztásos alapon egy időben tudnak működni. Ez azt jelenti, hogy minden regiszterük független. Probléma az ADATÚT használatokor merül fel: biztosítani kell, hogy az ADATÚT-at egyszerre csak az autonóm működésű, vagy a programozott vezérlő használja. E problémát háromféle módon lehet megoldani. Prioritás áramköröket építenek a programozott és autonóm működésű vezérlőbe és a két vezérlő vezérlőjelekkel (foglalt, szabad) áll kapcsolatban; ezek az áramkörök határozzák meg, hogy az ADATÚT-at pillanatnyilag melyik vezérlő foglalja le. Más megoldás az ADATÚT Foglalt (Busy) jelét használja

fel annak megállapítására, hogy szabad-e igénybevenni az ADATÚT-at. Ennél a megoldásnál nagyon gondosan vigyázni kell a szinkronizáló, reteszelő áramkörök időzítésének helyességére. A legegyszerűbb az a megoldás, amikor a számítógépre bízunk a két adatforgalom összekeveredésének kiküszöbölését, ez azonban nem minden számítógép esetében valósítható meg. A PDP-11 és a TPA-70 számítógépekben ugyanazon a buszrendszeren folyik a programozott és az autonóm átvitel, a számítógép vezérlőáramkörei biztosítják, hogy egyszerre csak egy egység vezérelhesse a buszon folyó adatátvitelt. Így, mivel az ADATÚT-on levő programozott, ill. autonóm adatforgalomhoz mindig tartozik a számítógép közös I/O buszrendszerén levő adatforgalom, ha az időzítéseket jól választjuk meg, a számítógépbe épített buszvezérlés az ADATÚT vezérlését is megoldja. A TPA-70-nel kapcsolatban ez a megoldás még azzal az előnnyel is jár, hogy mivel az I/O buszrendszeren levő pillanatnyi buszvezérlő prioritása programmal módosítható, az autonóm átvitel prioritása is változtatható.

Az eddigiekben a példákat, megoldásokat főleg a keretvezérlőkre vonatkoztattuk. A csoportvezérlőknél a megoldások teljesen hasonlóak, az eltérések csak az ADATÚT, ill. CSOPORTÚT szabványok közötti különbségekből adódnak. Mechanikai kivitelét tekintve a CSOPORTÚT vezérlőket általában nem CAMAC keretben, hanem különálló dobozban, vagy magában a számítógépben helyezik el, kivéve azokat a rendszereket, amelyek ún. rendszerkeretet alkalmaznak. A rendszerkereteket részletesebben a 3.1.5. pontban tárgyaljuk.

3.1.3.

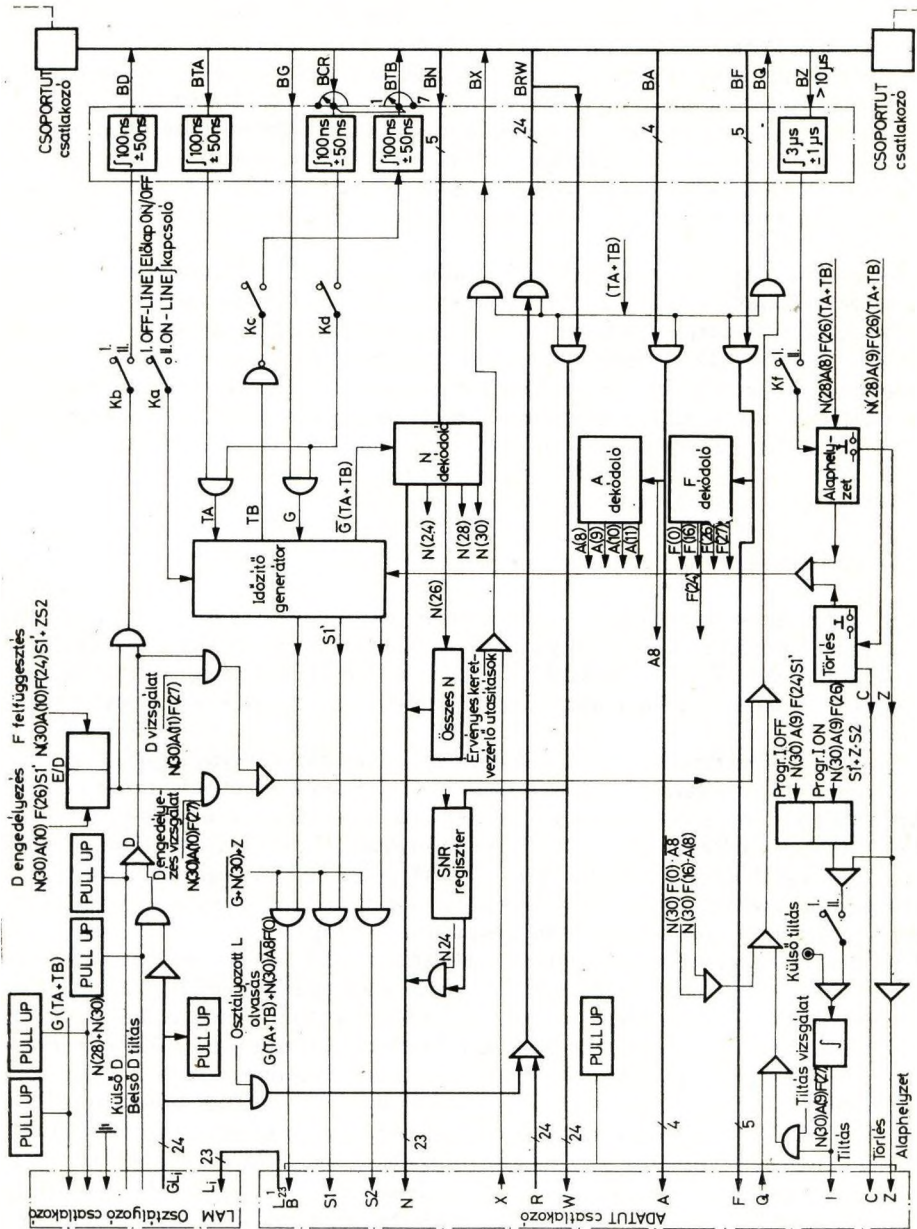
A1 típusú keretvezérlő

A 2. fejezetben láttuk, hogy az EUR 4100 közleményben szabványosították az ADATÚT jeleit, nagyobb rendszerek esetére pedig az EUR 4600 — egy lehetséges megoldásként — a CSOPORTÚT jeleit. Az ADATÚT-at és a CSOPORTÚT-at egymáshoz illesztő keretvezérlőt szintén szabványosították (EUR 4600), ezt nevezik A1 típusú keretvezérlőnek.

Az A1 típusú keretvezérlő tulajdonságainak egy része kézenfekvő: logikai, időzítési és elektromos jelszint illesztése a kétféle busz között. Ezen kívül azonban még néhány további tulajdonságát is szabványosították, amelyek a keretben levő modulok címezésével és LAM kezelésével kapcsolatosak. Szabványosak az A1 típusú vezérlő kezelőszervei is. A továbbiakban ezekkel a tulajdonságokkal is részletesen foglalkozunk, ugyanis igen sok keretvezérlő (pl. az előző fejezetekben ismertetettek nagy része is) átvette az A1 típusú vezérlőre szabványosított tulajdonságokat.

Az A1 típusú keretvezérlő részletes blokkvázlata a 3.7. ábrán látható. A keretvezérlők a CSOPORTÚT-on levő CAMAC címből a keret címét (BCRi) hét vonalon, dekódolva kapják. A vezérlőben van egy hétállású kapcsoló, amellyel beállítható a vezérlő, ill. a hozzátartozó keret címe. Ha a beállított keretcím és a BCRi jel megegyezik, a keretvezérlő aktívvá válik. Mivel a keretcím dekódolva érkezik, a csoportmehajtó több keretvezérlőnek is adhat egyidőben utasítást. Az **állomászám** (BN1, 2, 4, 8, 16) a CSOPORTÚT-on bináris kódban érkezik és a keretvezérlő dekódolja. A dekódolt állomáscímek közül az 1-től 23-ig terjedő címek az ADATÚT N (1)...N(23) vezetékeken keresztül a keret normál állomáshelyeit címezik meg. N(24) és N(26) a normál állomáshelyekre vonatkozó speciális címek. A keretvezérlő tartalmaz egy 23-bites, ún. állomászám-regiszttert (Station Number Register). Ezt a regiszttert szabványosított utasítással: N(30) A(8) F(16) lehet beírni a BRW1...BRW23 vonalakon

keresztül. Minden állomáshoz tartozik ebben a regiszterben egy bit. Amikor a keretvezérlő N(24) állomászám-címet vesz észre, akkor az összes, az állomászám-regiszter által kijelölt állomáshelyet egyszerre megcímezi. Ily módon több előre beállított című modulnak egyszerre lehet ugyanazt az utasítást kiadni. Az N(26) cím az állomászám-



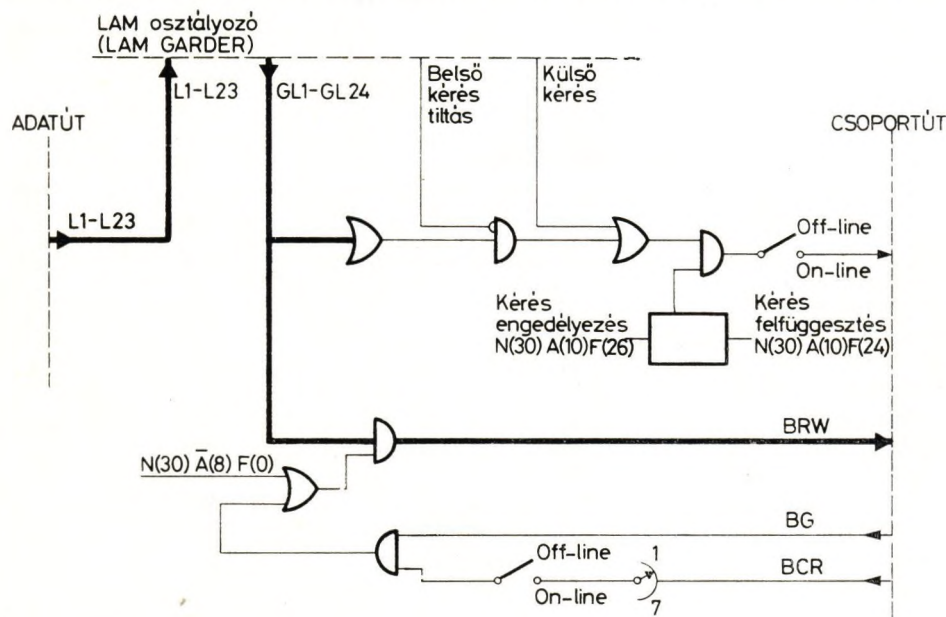
3.7. ábra
A1 típusú keretvezérlő részletes blokkvázlata

regisztertől függetlenül az összes állomáshelyet egyszerre megcímezi. Az N(28) és az N(30) címek a keretvezérlőre vonatkoznak (az állomászám-regisztert pl. N(30) címmel írjuk be), N(0), N(25), N(29) és N(31) későbbi felhasználásra vannak fenntartva.

A binárisba kódolt alcímet (BA) és funkciót (BF) a vezérlő megfelelő jelszintmódosítással, logikai változtatás nélkül továbbítja az ADATÚT felé. Közben azonban saját maga számára dekódolja is, mivel az N(28) és N(30) állomáscímeken a vezérlőnek is lehet utasításokat adni.

A keretvezérlő állítja elő az ADATÚT-on a közös vezérlőjeleket (Common Control Signals): az *Alaphelyzet* (Z), a *Törlés* (C) és a *Tiltás* (I) jeleket. Az *alaphelyzet* (Z) jelet az N(28) A(8) F(26) utasítás vagy a CSOPORTÚT-on levő BZ jel kelt. *Törlés* jel az N(28) A(9) F(26) utasítás hatására keletkezik. Mindkét jelet — a keretvezérlő off-line állapotában — nyomógombbal is kezdeményezni lehet. A *Tiltás* jelet (Inhibit) az N(30) A(9) F(26) utasítás állítja be és az N(30) A(9) F(24) utasítás szünteti meg. A *Tiltás* jel még az *Alaphelyzet* (Z) jel megjelenésekor is automatikusan kijut az ADATÚT-ra.

Az A1 típusú keretvezérlő egyik legfontosabb feladata a LAM jelek kezelése. A LAM jelek logikai útját a keretvezérlőben a 3.8. ábra mutatja. Az ADATÚT-ról —



3.8. ábra
LAM kezelés az A1 típusú keretvezérlőben

mint azt a 2. fejezetben az ADATÚT jelszabványoknál láttuk — minden állomáshelyet külön-külön LAM (L) vezeték köt össze a keretvezérlővel. Az A1 típusú keretvezérlő önmagában viszonylag nagyon egyszerű LAM kezelés végzésére alkalmas. Lehetőség van viszont arra, hogy ún. LAM osztályozó (LAM Grader) modult helyezünk a keretbe, amely tetszőleges bonyolultságú és tulajdonságú lehet (ezekkel részletesebben a következő, 3.1.4. pontban foglalkozunk). Az A1 típusú keretvezérlőn szabványosított csatlakozó van elhelyezve, amely kapcsolatot létesít az esetleges LAM osztályozóval.

A LAM jelek a keretvezérlőből — módosítások nélkül — kijutnak erre a csatlakozóra. A LAM osztályozóból az osztályozott LAM jelek (GL—Graded L) visszakerülnek a keretvezérlőbe. Ha nincs LAM osztályozó, a csatlakozó ellendarabján egyszerű átkötéssel lehet visszavezetni a LAM jeleket. A LAM osztályozóból érkező GL jelek a keretvezérlőben levő VAGY kapura jutnak, majd — ha a keretvezérlőben levő kérésengedélyező (Demand Enable) áramkör engedélyezi — a LAM jelek VAGY kapcsolata CSOPORTÚT Kérés (Branch Demand — BD) néven kijut a CSOPORTÚT-ra. Megjegyezzük, hogy míg az ADATÚT-on a LAM, L (Figyelj-Rám) elnevezés szokásos, a CSOPORTÚT-on a Kérés (Demand, D) szót használják. Az előbb említett VAGY kapu után levő további kapuk lehetővé teszik, hogy a LAM osztályozó letiltsa a jel útját, ill. további, kérést indítványozó jelet adjon a D jelhez.

A LAM kezeléssel kapcsolatos az Osztályozott L kérelem (Graded-L Request—BG). Ez a kérelem a CSOPORTÚT-on megjelenő, a csoportvezérlő által kiadott olyan utasítás, amellyel a csoportvezérlő az összes on-line keretből származó Osztályozott L jelek együttesét kéri. Ennek eredményeképpen az összes on-line keret GL1 jelének VAGY kapcsolata kerül a CSOPORTÚT Olvasás/Írás BRW1 vezetékére, a GL2 jelek VAGY kapcsolata a BRW2 vezetékre stb. Az A1 típusú keretvezérlő ennek megfelelően úgy van kialakítva, hogy a CSOPORTÚT-ról érkező Osztályozott L kérelem (BG) jel hatására a LAM osztályozóból kapott GL1...GL24 jeleket kikapuzza a BRW Olvasás/Írás vezetékekre. Az A1 típusú keretvezérlő szabványos utasításait a 3.1. táblázat foglalja össze.

3.1. táblázat

Az A1 típusú keretvezérlő szabványos utasításai

Művelet	Utasítás			Válasz
	N	A	F	
Alaphelyzet (Z)	28	8	26	BQ=0
Törlés (C)	28	9	26	BQ=0
GL olvasás	30	0—7	0	BQ=1
Állomászám regiszter (SNR) beírás	30	8	16	BQ=1
ADATÚT tiltás (I) megszüntetés	30	9	24	BQ=0
ADATÚT tiltás (I) beállítás	30	9	26	BQ=0
ADATÚT tiltás (I) vizsgálat	30	9	27	BQ=1, ha I=1
CSOPORTÚT kérés (BD) letiltás	30	10	24	BQ=0
CSOPORTÚT kérés (BD) engedélyezés	30	10	26	BQ=0
CSOPORTÚT kérés engedélyezés vizsgálat	30	10	27	BQ=1, ha BD engedélyezve
CSOPORTÚT kérés vizsgálat	30	11	27	BQ=1, ha van BD

Az A1 típusú keretvezérlő további feladata az ADATÚT és CSOPORTÚT időzítésének összehangolása. A 2. fejezetben részletesen tárgyaltuk az ADATÚT és a CSOPORTÚT műveletek időzítései közötti összefüggéseket (l. 2.15. ábra). Ebből látható, hogy a két jelvezeték időzítései kivárik egymást, és így az ADATÚT és CSOPORTÚT adatvezetékei közvetlen kapcsolatba kerülnek. Ezért a keretvezérlő nem tárolja

átmenetileg az adatvonalak tartalmát, csak közvetlenül — az időzítéseknek megfelelően — kapuzza.

Az A1 típusú keretvezérlőben az előlapon levő kezelőszerveket és csatlakozókat is szabványosították. Két szabványos csatlakozó a CSOPORTÚT-hoz kapcsolódik. Az éppen beállított keretcímet (C) könnyen leolvasható, de nehezen változtatható módon kell az előlapon elhelyezni. Az on-line — off-line állapot választását az előlapon kapcsolóval kell biztosítani. *Külső Tiltás* (I) jel számára szabványos csatlakozó szolgál. Két nyomógomb is szabványosítva van: az *Alaphelyzet* (Z) és a *Törlés* (C) nyomógomb.

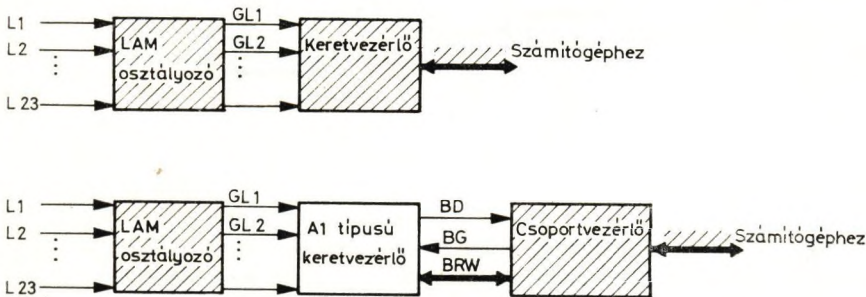
3.1.4.

LAM kezelés

Ebben a fejezetben az ADATÚT és CSOPORTÚT szintjén foglalkozunk LAM kezeléssel, tehát addig a mélységig, amíg a LAM-ot kérő modul (állomáshely) azonosítása tart. Azt az esetet, amikor egy modulon belül több LAM forrás van, tehát több okból kérhet a modul LAM kezelést a 4. fejezetben tárgyaljuk.

A CAMAC szabvány igen tág teret hagy a LAM kezelés különböző megoldási módjainak. A szabvány szerint az egy keretben levő összes állomáshely (modul) LAM jele külön-külön bejut a keretvezérlőbe. A keretvezérlő bonyolultságától vagy az alkalmazástól függően igen sokféleképpen lehet a LAM jellel foglalkozni: lehet egyszerűbb vagy bonyolultabb módon megszakításkérésként a számítógépbe továbbítani, de elindíthat valamilyen önálló műveletet is a keretvezérlőben (pl. autonom adatátvitelt). A CSOPORTÚT LAM kezelésére vonatkozó szabványt a 2. fejezetben, ill. az A1 típusú keretvezérlő leírásakor ismertettük. A szabvány itt lényegében csak azt írja elő, hogy a keretben levő LAM jelek VAGY kapcsolata jut a CSOPORTÚT-ra (BD jel), továbbá meghatározza az *Osztályozott L kérelem* CSOPORTÚT műveletet. Az alkalmazónak itt két helyen is szabad keze van: a keretvezérlőhöz kapcsolható LAM-osztályozó felépítése nincs rögzítve, valamint a csoportvezérlő, ill. a keretvezérlő LAM felhasználási módja sem szabványokban meghatározott (3.9. ábra).

A *legegyszerűbb* esetben a keretvezérlőhöz nem csatlakozik LAM osztályozó, e helyett az állomáshelyekről érkező LAM jelek egyszerűen át vannak kötve a GL vezetékekre. A keretvezérlő a GL jelek VAGY kapcsolatát képezi, és az engedélyezés-felfüggesztés vezérlő bistabil multivibrátor állapotától függően tovább engedi a szá-



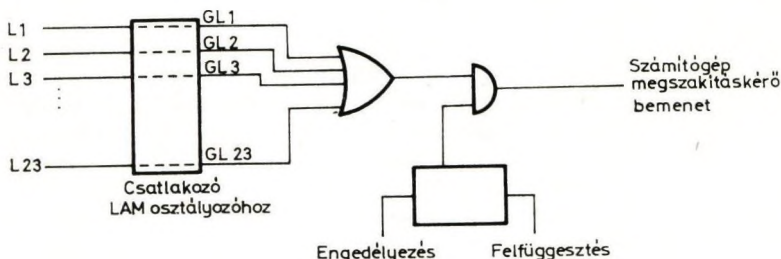
A ferdén vonalazott részek nem szabványosítottak!

3.9. ábra

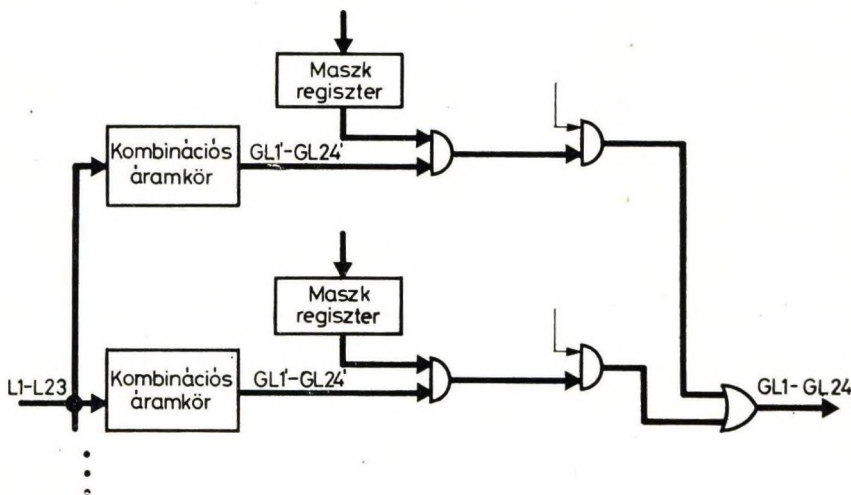
LAM kezelés szabványosított és szabadon megválasztható egységei

mítógép megszakítást kérő bemenetére. A 3.10. ábrán látható egy ilyen legegyszerűbb LAM kezelés egy keret esetére. Ha a keretvezérlő utasításai között nincs *Osztályozott L* kérelem, akkor a LAM-ot kérő modult csak hosszadalmas program segítségével lehet azonosítani: a keretbe helyezett moduloknak egymás után TEST LAM [F(8) A(0)] utasítást kell kiadni és ha $Q=1$ válasz érkezik, akkor ez a modul kért LAM-ot.

Kissé egyszerűsödik a LAM kezelés, ha a keretvezérlőnek az *A1 típusú keretvezérlő LAM kezelési lehetőségei* vannak (3.8. ábra). Ebben az esetben ugyanis az F(0) A(8) N(30) szabványosított utasítás hatására az Olvasás-adatvonalakon megjelennek a GL jelek, amelyek — ha a LAM osztályozót egyszerű átkötés helyettesíti — megegyeznek az egyes állomáshelyekről (modulokból) érkező LAM jelekkel. Így a számítógép egyetlen utasítással be tudja olvasni a keretben levő összes modul LAM állapotát.



3.10. ábra
Legegyszerűbb LAM kezelés



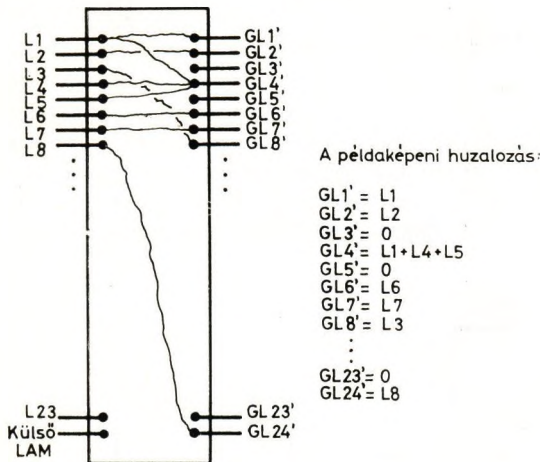
3.11. ábra
LAM osztályozók elvi vázlata

Mind az egy-, mind a többkeretes rendszerek LAM kezelését igen hajlékonnyá teszik a különböző típusú *LAM osztályozók* (LAM Grader). A gyakorlatban megvalósított LAM osztályozók elvi vázlatát a 3.11. ábrán láthatjuk.

A modulokból a keretvezérlő csatlakozóján keresztül érkező LAM jelek (L1... L23) *kombinációs áramkörre* jutnak. Ez igen sok féle lehet és itt csak a leggyakoribb megoldásokat tárgyaljuk. A legegyszerűbb esetben ez az áramkör azt teszi lehetővé,

hogy a bemenő L jeleket tetszőleges kimeneti pontra kössük, egyszerű huzalozott átkötéssel. A leggyakoribb megoldás ezen túlmenően azt is lehetővé teszi, hogy — az átkötés mellett — tetszőleges számú bemenő jelet huzalozott VAGY kapcsolat segítségével egy-egy kimeneti pontra lehessen kötni. Ezt a megoldást mutatja a 3.12. ábrán példaképpen bemutatott összehuzalozás.

Igen meggyorsítja a LAM-ot okozó modul megkeresését a *prioritásos kódoló áramkör*. Ez az áramkör lehetővé teszi, hogy a bemeneten levő legmagasabb helyi értékű logikai



3.12. ábra
LAM osztályozó egyszerű kombinációs áramkör

Bemenet									Kimenet			
1	2	3	4	5	6	7	8	9	1	2	3	4
0	0	0	0	0	0	0	0	0	0	0	0	0
x	x	x	x	x	x	x	x	1	1	0	0	1
x	x	x	x	x	x	x	x	1	0	1	0	0
x	x	x	x	x	x	x	1	0	0	1	1	1
x	x	x	x	x	1	0	0	0	0	1	1	0
x	x	x	x	1	0	0	0	0	0	1	0	1
x	x	x	1	0	0	0	0	0	0	1	0	0
x	x	1	0	0	0	0	0	0	0	0	1	1
x	1	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	1

3.13. ábra
Prioritásos kódoló áramkör igazságtáblája

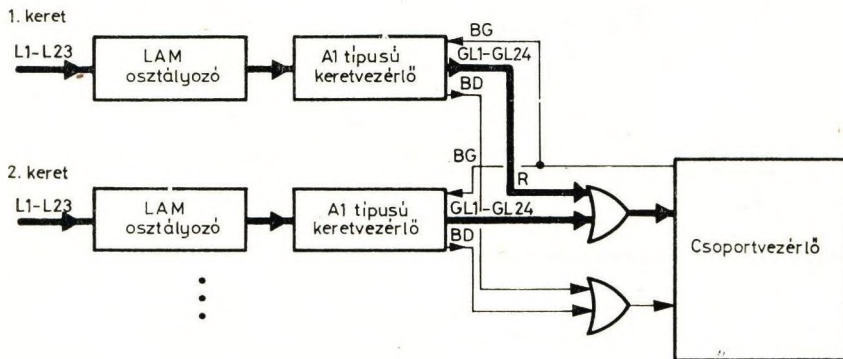
1 állapot binárisba kódolt formája jelenjék meg a kimeneten. Igazság táblája — tíz bemeneti ponttal — a 3.13. ábrán látható. Ilyen áramkör MSI integrált formában a kereskedelemben kapható (pl. TEXAS SN 74147 típus). Ha az egyszerű átkötés után ilyen prioritásos kódoló áramkört teszünk, akkor az osztályozott LAM kérés utasítás hatására a vezérlőben, ill. a számítógépben megjelenik a legnagyobb prioritású LAM

forrás binárisba kódolt formában. A prioritásokat huzalozott átkötéssel lehet beállítani.

A LAM osztályozók további szokásos tulajdonsága, hogy a LAM jeleket bitenként lehet maszkolni. E célra egy 24-bites maszkregisztert helyeznek az osztályozóba, és ennek egyes bitjeivel maszkolják az L jeleket. A maszkregiszter szokásos CAMAC utasítással beírható, esetleg kiolvasható.

Egy-egy LAM osztályozóban több kombinációs és maszkolóáramkör-ágot is el lehet helyezni, és a számítógép-utasítás választhatja ki, hogy melyik ág működik egy adott időben. Az osztályozó azt is lehetővé teszi, hogy külső csatlakozón érkező LAM jelet is hozzákeverjen a modulokból érkező L jelekhez.

Többkeretes rendszerekben ügyelni kell arra, hogy az egyes keretekből érkező LAM jeleket meg lehessen különböztetni. Amikor ugyanis a csoportvezérlő *Osztályozott L* kérelem (BG) utasítást ad a kereteknek, az összes on-line keretvezérlő kiadja az Írás/Olvásás (BRW) vonalakra a GL1...GL24 jeleit. Ezeknek a jeleknek logikai VAGY kapcsolata kerül be a csoportvezérlőbe, tehát az összes GL1 jel VAGY kapcsolata, GL2 jel VAGY kapcsolata stb. (3.14. ábra). Ezért nagyon fontos, hogy az egyes keretek



3.14. ábra
Többkeretes rendszer LAM jeleinek kapcsolata

LAM osztályozóin az átkötéseket, a GLi vonalak kiosztását az egész rendszerrel összhangban alakítsuk ki. Az egyes keretekhez tartozó LAM osztályozók címe természetesen különböző (a keretcím (C) egyedi), ezért külön-külön is kiolvashatók az egyes keretekből keletkező GLi jelek.

A keret-, ill. csoportvezérlő LAM kezelése szintén nincs megkötve, így teljes mértékben ki tudják használni a kapcsolódó számítógép megszakítási lehetőségeit. A keretvezérlő például a GL jelekből igen könnyen tud kódolt formát előállítani, amit aztán a számítógépnek megszakításvektorként tud továbbítani. Ugyancsak lehetőség van arra, hogy többszintű megszakításrendszer tetszőleges szintjére tetszőleges GL jeleket kapcsoljunk, és így például külön megszakítási vonalra kerülhetnek az üzemszerű működés LAM jelei és a hiba jellegű LAM jelek. Szokásos megoldás az is, hogy egy vagy néhány GL jelet nem továbbít a vezérlő megszakításjelként a számítógépbe, hanem autonóm adatátviteli ciklust indít el.

A csoportvezérlők LAM kezelése — a keretvezérlőkhöz hasonlóan — igen rugalmasan tud kapcsolódni a számítógép megszakítási rendszeréhez. A keretvezérlő esetében szinte az összes LAM kezelési tulajdonság sztatikus, kombinációs áramkörök-

kel megvalósítható; csoportvezérlők esetén a helyzet bonyolultabb, mivel a csoport LAM jel (BD) megjelenésekor a csoportvezérlőben nem áll rendelkezésre a teljes LAM minta. Igen sok csoportvezérlő a BD jel hatására automatikusan osztályozott LAM kérelem (BG) utasítást ad a kereteknek, beolvassa a vezérlőbe az osztályozott GL1... GL24 jeleket és csak ezután kér megszakítást a számítógéptől. Ekkor azonban már kész megszakításvektort tud szolgáltatni a számítógépnek, ill. igénybe veheti a többszintű megszakítás lehetőségét.

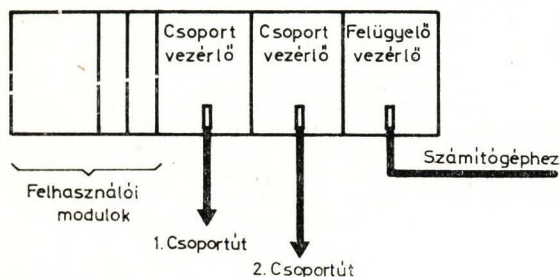
Az összetett vezérlők (l. a 3.2. szakaszt) LAM kezelése az eddigiektől eltérő lehet. A LAM jelek ugyanis nem feltétlenül számítógépes kapcsolatot kezdeményeznek, hanem igen gyakran önálló műveletsorozatot indítanak el.

3.1.5.

Több-csoportutas rendszerek. Rendszervezérlők

Néhány igen bonyolult CAMAC rendszerben az alkalmazott modulok nem fértek el a CSOPORTÚT-ra szabványosított maximálisan hét keretben, ezért vált szükségessé a több-csoportutas rendszerek kidolgozása. Ezekre a rendszerekre eddig még nincs kialakult szabvány. A CAMAC software ugyan már figyelembe veszi a több CSOPORTÚT címzésének lehetőségeit, az eddig megvalósított hardware kialakítások azonban különbözőek.

A gyakorlatban használt több-csoportutas rendszerek igyekeznek a meglévő CAMAC szabványokból minél többet átvenni. Így alakult ki a mechanikusan teljesen a szokásos kerettel megegyező kivitelű, ún. vezérkeret. A vezérkeretet az jellemzi, hogy a

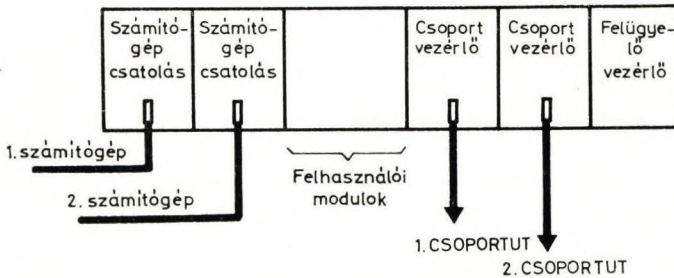


3.15. ábra
Rendszerkeret egy utasításforrással

benne levő egységek (modulok) kapcsolatot teremtenek egy-nél több CSOPORTÚT és egy vagy esetleg több utasításforrás (pl. számítógépes kapcsolat) között.

A megvalósítás szempontjából egyszerűbb, ha egy utasításforrás van (3.15. ábra). Ez az utasításforrás általában számítógépes csatlakozás, lényegében megfelel az eddigiekben tárgyalt keretvezérlő modulnak. Ezek a megoldások átveszik az ADATÚT villamos és logikai jelszabványait, esetleg egy-két jellel kibővítve. A rendszerkeretben felhasználói modulok is lehetnek, amelyek vezérlése, adatforgalma megegyezik a szabványos ADATÚT-ra előírtakkal. Ha a felügyelővezérlő csoportvezérlő modul címez meg, akkor kapcsolat létesül az utasításforrás (számítógép) és a megcímezett CSOPORTÚT, ill. CSOPORTÚT-hoz kapcsolódó további felhasználói keretek között.

Sokkal bonyolultabb, ha több utasításforrás kapcsolódhat a rendszerkerethez (3.16. ábra). Két probléma merül fel: egyrészt biztosítani kell azt, hogy az „Adatút”*-at egyszerre csak egy utasításforrás vezérelje. A másik probléma abból adódik, hogy az „Adatút” vezérlésére egy állomáshely van speciálisan kiképezve, csak ez van összekötve az egyes állomáshelyek LAM (L) és cím- (N) vezetékével. A felügyelővezérlő feladata ennek megfelelően kettős: egyrészt irányítja az „Adatút” forgalmát, másrészt pedig közvetít (átköt, esetleg átkódol) az utasításforrás és a felhasználói modulok, ill. csoportvezérlők között.



3.16. ábra
Rendszerkeret több utasításforrással

Az egyik megvalósított megoldásnál [16] 4 utasításforrás és 7 csoportvezérlő helyezhető el a rendszerkeretben. A fennmaradó helyet normál felhasználói modulok foglalhatják el. A rendszerkeret vázlata megegyezik a 3.16. ábrán bemutatottal.

Az „Adatút” mellett további 4 vezeték van. Ezek az utasításforrásokat kötik össze a felügyelővezérlővel, és ezeken keresztül kérik az utasításforrásokat, hogy az „Adatutat” vezérelhessék. Amikor az „Adatút” vezérlését egy utasításforrás a felügyelővezérlőtől megkapta, akkor az Írás vonalakra kiadja a hívni kívánt cím B és N (CSOPORTÚT és állomászám) értékét, míg az F és A jeleket az „Adatút” F és A vezetékére adja. A felügyelő vezérlő megvizsgálja B értékét. Ha $B=0$, ez azt jelenti, hogy az utasításforrás a rendszerkeretben levő valamely normál állomást kívánja elérni. A felügyelővezérlő ekkor megőrzi N értékét, dekódolja és kiadja a megfelelő N vonalra, B jelet ad az „Adatút”-ra és összeköti az „Adatút” Írás—Olvasás vezetékét. Ezután az utasításforrás közvetlen kapcsolatba kerül a megcímezett modullal. Ha az utasításforrás olyan címet küld a felügyelővezérlőnek, ahol $B \neq 0$, ez azt jelenti, hogy valamelyik csoportvezérlőt akarja vezérelni. A felügyelővezérlő $B \neq 0$ észlelésekor gondoskodik arról, hogy az utasításforrás a kiválasztott csoportvezérlővel összekötődjék.

3.1.6.

Vezérlőkkel együttműködő kiegészítő egységek

A teljesség igénye nélkül megemlítnék néhány olyan egységet, amelyek valamilyen módon kapcsolatban vannak a vezérlőegységekkel, és a teljes rendszerre vonatkoznak.

Ha a CAMAC rendszer CSOPORTÚT-as megoldást használ, a szabvány előírja, hogy a CSOPORTÚT-at, a csoportvezérlőtől legtávolabbi pontján, hullámenállásával le kell zárni. E célra csoportút-lezáró modul (Branch Terminator) szoktak használni, amelyet a CSOPORTÚT-on levő utolsó A1 típusú keretvezérlő mellé helyeznek el. Igényesebb megoldások a lezáró modul előlapján lámpákkal kijelzik a CSOPORTÚT egyes jeleinek logikai állapotát.

* Az „Adatút” elnevezést azért használjuk idézőjellel, mert csak fizikailag, huzalozás szempontjából egyezik meg a szabványos ADATÚT-tal, a vezeték használatára szempontjából nem.

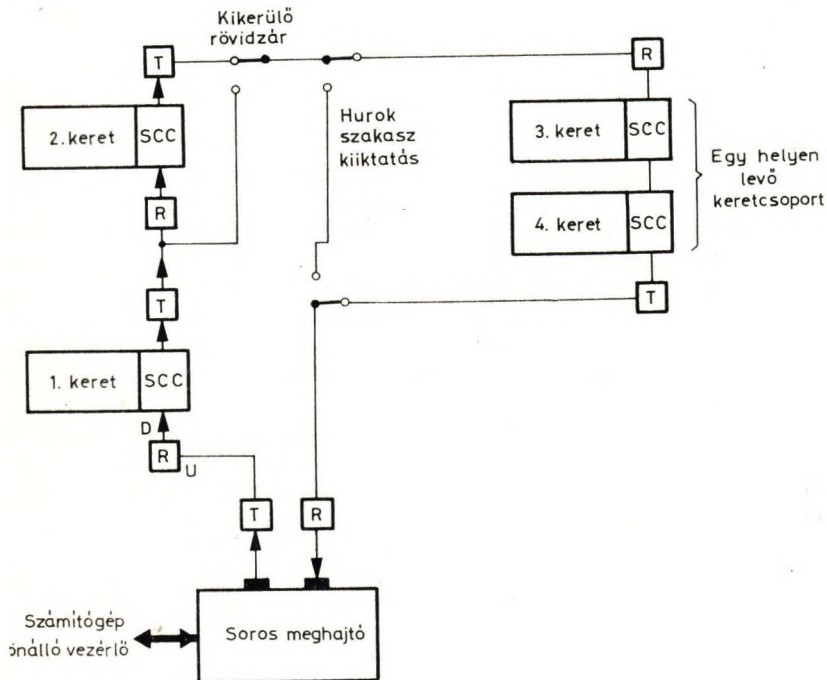
Ismeretes, hogy a CSOPORTÚT egyes szakaszainak hossza nem lehet lényegesen hosszabb, mint 20 méter. Ha ennél nagyobb távolságra kell vezetni a CSOPORTÚT-at, akkor valamilyen adó-vevő áramköröket kell beiktatni. E célra fejlesztettek ki *differentiális csoportút-kiterjesztő* modulokat. Ezek a modulok integrált adó-vevőket (pl. TEXAS SN 75110 N és SN 75107 N- típusok) használnak.

3.2.

Soros vezérlők

A CAMAC rendszer különböző vonalainak áttekintésekor megismerkedtünk a SOROS FŐÚT-tal. Láttuk, hogy az átvitel bitsoros vagy byte-soros lehet és maximálisan 62 keret kapcsolható a vonalba. A SOROS FŐÚT szabványosítása csak napjainkban készült el, különböző előzetes soros rendszerek készültek az egyes tulajdonságok vizsgálatára, azonban az üzemszerűen működőket csak 1974-ben valósították meg. A soros vezérlőkkel ezért csak vázlatosan foglalkozunk.

A SOROS FŐÚT vázlatát a 3.17. ábrán láthatjuk. A soros meghajtóegység igen különböző lehet: legegyszerűbben egy telexgép-kimenet, amely 8-bites kódokat tud egy soros vonalra kiadni. A SOROS FŐÚT szabályainak megfelelően a 8-bites kódokat (amelyek a címet, adatot, állapotot és egyéb információt tartalmazzák) akár a meghajtóhoz csatlakozó számítógép, akár önálló vezérlő programozottan össze tudja állítani, ill. a vett kódokat értelmezni tudja. A vonalon az átvitel módjának megfelelő adó (Transmitter—T) és vevő (Receiver—R) áramkörök vannak. Az átvitel alatt a

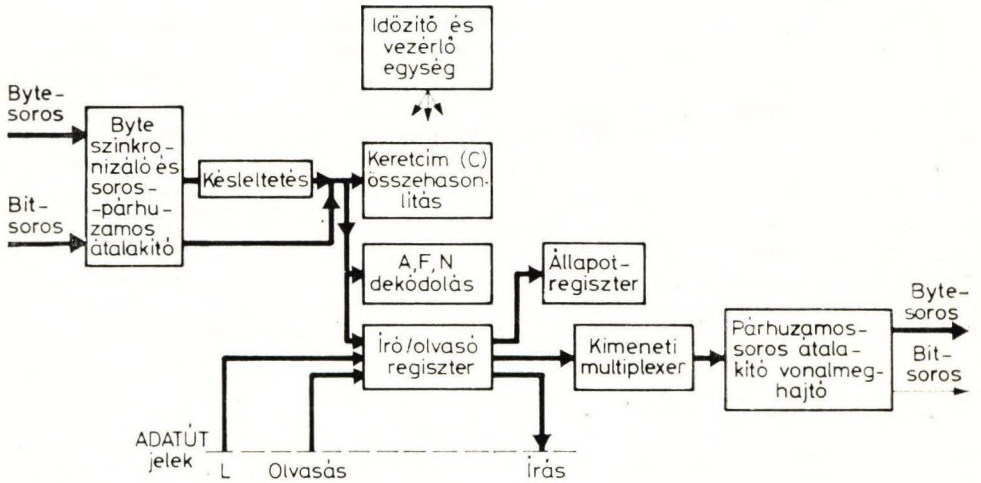


3.17. ábra
SOROS FŐÚT vázlat

jelek szabványban nem rögzítettek (Undefined—U), a vevő után viszont már meghatározottak (Defined—D). E meghatározott jeleket fogadja és adja tovább a soros keretvezérlő (Serial Crate Controller—SCC).

A vonalnak két szabványosított tulajdonsága van: egyrészt egy-egy keretet ki lehet kerülni anélkül, hogy a SOROS FŐÚT folyamatossága megszakadna, másrészt pedig egy-egy keret után a vonalat a soros meghajtó felé rövidre lehet zárni, azaz a további hurokszakaszt kiiktatható. Ennek főleg mérési, hibakeresési jelentősége van.

A soros keretvezérlő (SCC) vázlata a 3.18. ábrán látható. A soros vonali átvitel szabályai (I. 2. fejezet) szerint kell az adatokat fogadni és értelmezni, ezért logikailag



3.18. ábra
Soros keretvezérlő (SCC)

sokkal bonyolultabb és több feladata van, mint az egyszerű keretvezérlőnek. A bejövő — soros vagy párhuzamos — jelsorozatot egymás után következő byte-okká alakítja át az első blokk. Ezután a byte-ok vagy közvetlenül tovább jutnak az SCC különböző vizsgálóáramköreire, vagy egy 3-byte-os késleltetés van közébeiktatva. Erre azért van szükség, hogy az SCC LAM kéréseket tudjon közébeiktatni anélkül, hogy a vonalon levő üzenet megsérülne. Az SCC-be jutott üzenet, ha nem erre a keretvezérlőre vonatkozik, tovább jut a kimenetre. Ha erre a keretre vonatkozott, a vezérlő az egymás után következő byte-okból rekonstruálja a teljes üzenetet, elvégzi az ADATÚT ciklust és válaszüzenetet ad ki a kimenő vonalra.

Az üzenetek közötti időben a keretvezérlőnek módja van LAM jeleket kiadni a soros meghajtó felé. Mivel a soros keretvezérlőnek viszonylag sok „intelligenciá”-val kell rendelkeznie, néhány gyakorlati megvalósítás során funkcióit összekötötték az önállóan is működő összetett vezérlő funkcióival [9]. Ha azonban a keretvezérlő lehetőségei közül nem mindegyiket valósítjuk meg (pl. kihagyjuk a LAM kérést), akkor viszonylag egyszerű vezérlők is készíthetők. Kereskedelmi forgalomban főleg az egyszerű, ún. SCC—L1 típusú soros vezérlők vannak [6].

3.3.

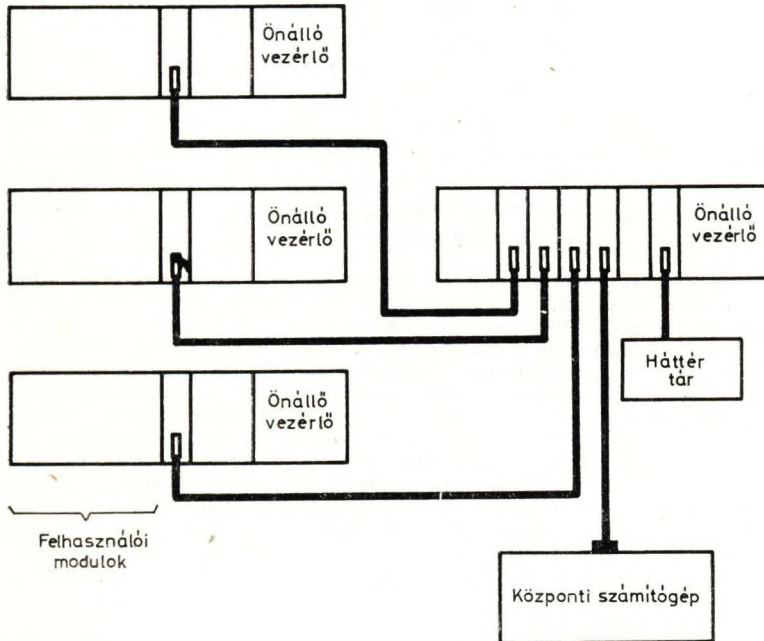
Összetett vezérlők

Az összetett vezérlőknek az eddigiekben bemutatottakhoz képest sokkal nagyobb az önállóságuk. Önállóan, számítógéptől függetlenül tudnak több utasításból álló utasítássorokat végrehajtani, saját utasításkészletük alapján.

A saját utasításkészletnek igen sok előnye van [7]. Egy CAMAC orientált önálló vezérlő szóhossza célszerűen legalább 24 bit, így ki tudja használni a teljes ADATÚT szélességet és nincs szükség többszöri átvitelre. A CAMAC cím és funkció a ki-, ill. bemeneti utasítás része lehet, ezért egyetlen utasítással beírható vagy kiolvasható egy tetszőleges CAMAC regiszter a vezérlő regiszterébe. A programozott vezérlők-nél láttuk, hogy egy CAMAC utasítás végrehajtására 2...4 számítógép-utasítás szükséges, és gondoskodni kell arról, hogy programmegszakítás ne tudja szétválasztani ezeket az utasításokat. Az ideális CAMAC processzor (önálló vezérlő) egy számítógép-utasítással tud egy CAMAC utasítást elvégezni és nem zavarja meg a programmegszakítás.

A saját utasításkészletben történő programozáskor célszerű — a számítógépekhez hasonlóan — valamilyen *programozási nyelvet* használni [17]. Ez a nyelv általában assembler szintű. Ha az összetett vezérlő programját PROM tartalmazza, akkor az assembler a fordítás eredményeképpen nemcsak listát, hanem például PROM programozógységet vezérlő lyukszalagot is készít.

Az összetett, önálló vezérlők új rendszertechnikai lehetőségeket adnak a CAMAC rendszereknek. Egyrészt lehetőség nyílik arra, hogy egyetlen keret tartalmazza a mérő-, vezérlő-, feldolgozóegységeket, és így kisebb, olcsóbb rendszereket lehet össze-



3.19. ábra
Osztott intelligenciájú CAMAC rendszer

állítani. Ez főleg egyszerűbb mérések (pl. kémiai-analitikai) automatikus vezérlésére, mérésadatgyűjtésre, mérőautomaták készítéséhez előnyös. A másik igen fontos rendszertechnikai szempont a bonyolult, nagyobb rendszerek kiépítése során kerül előtérbe: az önálló vezérlők segítségével a nagy rendszerekben levő „intelligenciát” szét lehet osztani a rendszert alkotó keretekbe vagy keretcsoporthoz. Az egyes intelligens keretek például a keretekben levő adatátviteli modulok útján kapcsolhatók össze egy központi számítógéppel (3.19. ábra).

Vegyünk példaképpen egy olyan rendszert, amelyben az egyes keretek önállóan végzik az adatgyűjtés, érvényesség, határértékellenőrzés, adatredukálás műveleteit. Ha egy összesített adattömb elkészül, a mérőkeret az adatokat átadja a számítógép mellett levő keretnek, amely az adatokat vagy háttértárolóba helyezi el, vagy átadja a számítógépnek további feldolgozásra. A bonyolultabb számítási igényű munkákat (pl. hatásfokszámítás) a számítógép végzi el. Ha a központi számítógép meghibásodik, a helyi adatgyűjtés és primer ellenőrzés változatlanul tovább folytatódik, sőt az adatok háttértárolóba gyűjtése is zavartalan.

Az összetett vezérlőket funkciójuk szerint több csoportra lehet osztani:

- önálló CAMAC processzorok;
- számítógéphez kapcsolódó CAMAC processzorok;
- önálló vagy számítógéphez kapcsolódó általános processzorok.

3.3.1.

Önálló CAMAC processzorok

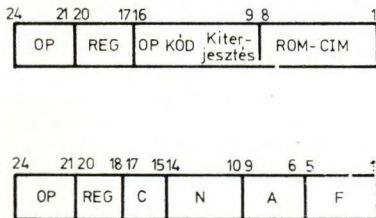
Az önálló CAMAC processzoroknak három fő jellemzőjük van: nem kapcsolódnak számítógéphez, utasításrendszerük CAMAC irányítottaságú és általában nem rendelkeznek olvasható-írható tárolóval (RAM). A CAMAC orientáltság azt jelenti, hogy az utasításkészlet nem a szokásos, általános számítógép-utasításkészlet, hanem főleg a CAMAC modulok vezérlésére, adatmozgatására szorítkozik.

Háromfajta utasítást feltétlenül tartalmaznia kell egy ilyen készletnek: CAMAC utasítást (CNAF), feltételes ugrást és egyesével számlálást. A feltételes ugrás és számlálás utasításokra azért van szükség, hogy ciklusokat lehessen képezni. Egy ilyen önálló vezérlő legegyszerűbb feladata az, hogy adott számú CAMAC regisztert olvasson ki, tároljon be egy tárolóba, esetleg írógépen írja ki az adatokat. Önálló CAMAC processzorokra két példát mutatunk be: egy régebbi, több helyen használt megoldást és egy új, korszerű kivitelű berendezést.

Az első példában a programozható keretvezérlő (Programmed Dataway Controller) [18] három egység széles. Szóhossza 24 bit. Hozzákapcsolható egy 256-szavas csak olvasható tároló (ROM), amely a programot tartalmazza. Az eredeti megoldás szerinti kivitel dugaszolható, diódás mátrix. Adatok számára nincs külön tárolási lehetőség, csak a modulokban. A vezérlő tartalmaz egy utasításcím-regisztert, amely az éppen folyó utasítás 8-bites címét tartalmazza, egy 8-bites utasításszámlálót és egy 24-bites akkumulátort. A vezérlő utasításkészlete 16 csoportba osztható. Az utasítás tartalmazhatja közvetlenül az operandust, amely az akkumulátorba vagy egy CAMAC címre írható be. CAMAC utasítások (CNAF) végrehajthatók egyetlen utasítással. Adatolvasásnál ugyanez az utasítás még azt is lehetővé teszi, hogy az akkumulátorhoz hozzáadódjék vagy kivonódjék az adat. Feltételes utasítás-átlépés és feltételes ugrás hajthatók végre. Néhány speciális utasítás után szünet következik, ameddig egy megszakításkérés nem érkezik. Feltételes szünet utasítás hatására a gép addig ismétli az utasítást, amíg a feltételbitekben meghatározott feltétel be nem következik. Ugrás és szubrutinra ugrás utasítások, valamint az akkumulátorra vonatkozó egyszerű operációk egysítesítik ki az utasításkészletet. Később kiegészítették a keretvezérlőt egy tárolóvezérlő egységgel, amelynek segítségével 2730-szavas 24-bites ferritgyűrűs tároló köthető a rendszerhez.

Másik példaként önálló csoportvezérlőt [11] mutatunk be. Ez a modul négy egységnyi helyet foglal el a keretben. Az utasítások max. 256-szavas, csak-kiolvasható tárolóban vannak. Egy utasítás végrehajtási ideje $4 \mu\text{s}$ körüli. A csoportvezérlőnek 4 címezhető regisztere van: adatregiszter (24 bit), alcímregiszter (A) (4 bit), állomásszám-regiszter (SNR) (5 bit), indexregiszter (8 bit). Ezen kívül

a működéshez szükséges címregiszterrel, utasításregiszterrel és programszámlálóval rendelkezik. Írható—olvasható tárolója nincs. Az utasításszó formátuma alapvetően kétféle (3.20. ábra): az első típusba tartoznak az ugrás- és regiszterbetöltő utasítások, a másodikba a vezérlő és CAMAC utasítások. Az ugrásutasítások feltétel nélküli és feltételes ugrás típusúak. Feltételes ugráskor a 18...20 bitek által kijelölt regiszter eggyel nő; ha eléri a 8...16 bitek által meghatározott számot, a program elugrik az 1...8 bitek által adott címre. Az ugrásutasítások közé tartozik a szubrutinra ugrás és visszatérés. A regiszterbetöltő utasítások közé tartoznak a közvetlen operandussal vagy ROM tárolóból történő



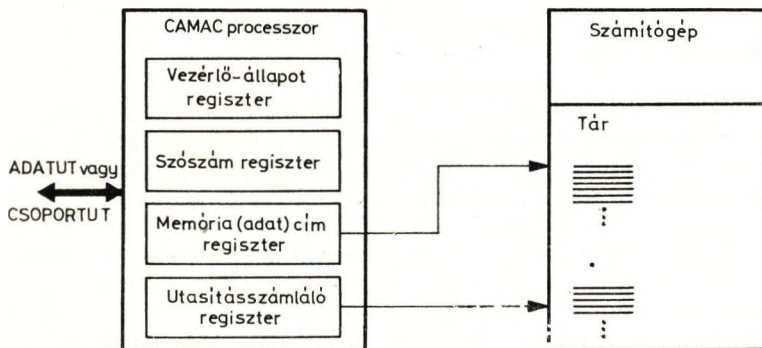
3.20. ábra
Önálló csoportvezérlő utasításformátuma

betöltések. Az operandus címzése indexelt is lehet. A CAMAC utasítások Q vizsgálatával vagy anélkül kerülnek végrehajtásra. Ha Q vizsgálat van, Q=1-nél utasításátlépés keletkezik. Különleges utasításokkal lehet megjelölni a program végét vagy várakozást valamilyen esemény bekövetkezésére. Ez lehet BD jel, külső jel vagy nyomógombos indítás.

3.3.2. Számítógéphez kapcsolódó CAMAC processzorok

Ebbe a csoportba tartozó processzorok lényegében hasonlóak a korszerű számítógépek csatornavezérlő rendszeréhez, ill. ki-, ill. bemeneti processzoraihoz [19]. Más oldalról ezeket a vezérlőket úgy is tekinthetjük, mint az autonóm adatátvitel általánosítását, kiterjesztését: egy átvitelkérés során nemcsak egyetlen, rögzített utasítást képes végrehajtani, hanem egy korlátozott utasításkészletből álló utasítás-sort. Az utasításkészletre ugyanaz jellemző, mint az önálló CAMAC processzorok esetében.

A működést vázlatosan a következőkben ismertetjük (3.21. ábra). A számítógép programozott utasításokkal az önálló működéshez szükséges paramétereket elő-

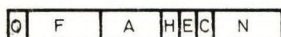


3.21. ábra
Számítógéphez kapcsolódó CAMAC processzor vázlata

készíti. Ezek a paraméterek az autonóm adatátvitel paraméterein (szószám, memória-cím) kívül: az utasításszámláló és a vezérlőállapot-regiszter kezdő tartalma. Ezután vagy a számítógép külön utasítással, vagy a CAMAC rendszer LAM jellel kezdeményezi az egyes autonóm működéseket. Az autonóm működés indításakor az utasításszámlálóban levő cím alapján a vezérlő az utasításokat tartalmazó tárolóból kikeresi a megfelelő utasítást, végrehajtja, veszi a következőt stb. Az utasításokat tartalmazó tároló általában a számítógép saját operatív tárolója vagy mikroprogramtára. Az utasítássor végéig vagy a szószámregiszter kiürülése, vagy speciális utasítás jelzi. A működés végén a vezérlőprogram megszakításkérésrel jelentkezik a számítógép felé.

A példaként bemutatott vezérlő PDP—15 számítógéphez csatlakozik és két keret tud vezérelni [1]. Két részből áll: programozott vezérlő és önálló CAMAC processzor. A két rész működése egymástól független, kivéve azt, hogy az önálló processzorba a kezdeti paraméterbetöltéseket a programozott vezérlő végzi. A vezérlőnek az az érdekessége, hogy összesen 4 programot tud egyszerre kezelni, ezek közül — időosztásos alapon — egyszerre két program tud működni. Ennek megfelelően a paramétereket tartalmazó regiszterek mind kétszer fordulnak elő. A két program prioritása változtatható.

A processzor utasításformátuma a 3.22. ábrán látható. A szóhossz — a PDP—15-höz illeszkedően — 18 bit. A formátum a CAMAC utasításnak (CNAF) felel meg, kiegészítve 3 bittel (C 1-bites, két keret kiválasztására ad lehetőséget).



Q : Q válasz figyelés

H : 24 bites szóátvitel

E : utasítássor vége (Exit)

3.22. ábra
CAMAC processzor
utasításformátuma

Néhány F kódnak speciális jelentése van: az F(12) lapon belüli feltétel nélkül ugrást hoz létre, az F(4) és F(6) a memória címtartalmát egyesével számláltatja. A Q választól függően utasításátlépés történik. A vezérlő belső regisztereinek CAMAC címe van, így CAMAC utasításokkal lehet tartalmukkal műveletet végeztetni.

Amikor egy CAMAC modul meghatározott LAM jellel indítani akarja a CAMAC processzort és éppen nincs átvitel, a vezérlő a csatornaszámnak megfelelő tárolóhelyről kiolvassa az illető csatornához tartozó utasításszámlálót. Ennek alapján kiolvassa a végrehajtandó utasítást, az utasításszámlálót pedig eggyel megnövelve visszairja. Ezután megvizsgálja az F8 és H biteket és az eredménytől függően 0, 1 vagy 2 adatátviteli ciklust indít az F16 értékétől függő irányban. A szószám és a memória adatcím automatikusan növekszik minden átvitel után.

3.3.3.

Általános vezérlőprocesszorok

Az újabb fejlődés a *kisszámítógépek* terén arra mutat, hogy egyre bonyolultabbá válnak. Ennek oka kettős: egyrészt megnöttek a technológiai lehetőségek (MSI és LSI integrált áramkörök), másrészt a rendszertechnikai fejlődés a szétosztott intelligencia irányába, tehát az összetettebb, okosabb kisgépek irányába mutat.

A mérési adatgyűjtés automatizálásban — a CAMAC rendszerektől függetlenül — megjelentek az *intelligens, programozható vezérlők*, amelyek nem feltétlenül kapcsolódnak további számítógéphez, utasításkészletük viszont univerzális. Ilyenek például a Quindar cég ipari és kereskedelmi adatgyűjtő- és vezérlőrendszere, a Process Control Systems 2000-es sorozata, a különböző programozható kalkulátorokkal vezérelt rendszerek stb. Ezekben a vezérlőkben egyre inkább alkalmaznak mikroprocesszorokat, amelyek általános számítógép-utasításrendszerrel rendelkeznek.

A CAMAC rendszerek tervezői — mint az előzőekben láttuk — már régebben készítettek ilyen vezérlőket, és ez a folyamat — az újabb technológiai lehetőségeknek és rendszertechnikai igényeknek megfelelően — meggyorsult.

Az *önálló processzor* tulajdonságokkal rendelkező CAMAC vezérlők főbb jellemzői a következők:

- utasításkészletük teljes kishámítógép utasításkészlet;
- írható-olvasható tárolója és csak-olvasható tárolója van;
- a ki-, ill. bemeneti buszvonalaival megegyeznek az ADATÚT-tal vagy CSOPORT-ÚT-tal;
- önállóan, számítógép nélkül is képesek működni.

A *teljes utasításkészlet* azt jelenti, hogy egyszerű aritmetikai, logikai utasításai is vannak, tehát a processzor mindazon funkciók teljesítésére képes, amire általában a kishámítógépek. Az utasításformátum kialakításánál nehézséget jelent a kötött szóhossz. A CAMAC rendszerekhez a hosszú szóhosszúság illeszkedik jobban: a szóhossz legyen legalább 24 bit. Egy példaképeni önálló vezérlőprocesszor ennek az elvnek szem előtt tartásával készült 32 bittel [14]. Ugyanakkor viszont egyrészt a jelenleg szokásos kishámítógépekhez való illeszkedés miatt 16-bites szóhossz lenne kedvező, másrészt a jelenleg forgalomban levő mikroprocesszorok is csak 8...16-bites szóhosszal készülnek. Ezek miatt nem alakult ki még egységes szóhossz-, ill. utasításformátum.

Az önálló processzoroknak *írható-olvasható és csak-olvasható tárolójuk* van, az általános számítógépekhez hasonlóan.

A program utasításainak és az adatoknak a tárolóban való elhelyezése, a tároló helykiosztása többféle: van olyan megoldás, amelynél az utasítások és az adatok tetőszólegesen helyezkednek el a tárolóban, akár egymással keveredve. Ez a hagyományos elvű számítógépek megoldása. Sok esetben célszerűbb a változtatás nélküli programokat, esetleg adatokat különválasztani a változtatható adatoktól, mivel így az előzőeket csak-olvasható tárolóban is lehet tárolni; a processzorok más része ezt a megoldást követi.

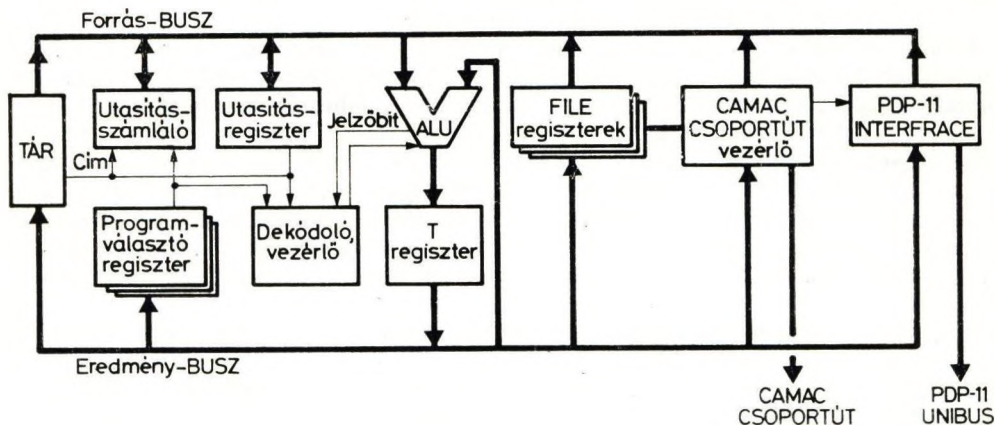
Az általános vezérlőprocesszorok *önállóan, számítógép nélkül* is működnek. A gyakorlatban megvalósított általános processzorok ebből a szempontból igen nagy változatosságot mutatnak. Egy részük számítógéphez is kapcsolódhat, azonban önállóan is képesek működni. Érdekességként viszont megemlíjtük, hogy a későbbiekben példaként bemutatandó vezérlő, amelyet mind önálló, mind számítógéppel összekötött üzemre terveztek, a legtöbb alkalmazásban önállóan működik. Az általános vezérlőprocesszorok másik része az önálló működés mellett soros keretvezérlőként is alkalmazható, ill. egyszerű soros keretvezérlővel való együttműködésre alkalmas. A soros átvitel üzenetformátumainak kezelése legnagyobb részt egymás után következő logikai lépésekből áll, amelyeket — megfelelően programozva — egy általános processzor jól el tud végezni. Az adatok előfeldolgozását pedig önálló üzemmódban látja el a processzor. Vannak végül olyan autonóm vezérlők, amelyeknél nincs is meg a lehetőség a közvetlen számítógépes kapcsolatra, csak külön modul segítségével.

Általános vezérlőprocesszorok alkalmazásakor a szokásos mérő, kimeneti, bemeneti, felhasználói modulokon kívül néhány további modul is szükséges. Így például különböző típusú tárolómodulokat, továbbá az általános perifériákat illesztő CAMAC modulokat kell alkalmazni. Ez azonban nem teljesen új feladat, mivel már korábbi, számítógéppel vezérelt CAMAC rendszereket is készítettek olyan rendszertechnikai megoldással, hogy az összes szokásos általános perifériás egységet a CAMAC kereten keresztül illesztették a rendszerbe [15].

Két, gyakorlatban megvalósított általános vezérlőprocesszort mutatunk be példaként.

Az egyik vezérlőprocesszor [2], [4] PDP—11-hez kapcsolódóan vagy teljesen önállóan képes működni, és CAMAC CSOPORTÚT-at vezérel. A processzor három fő részre tagolódik: az önálló processzorra, a CAMAC csoportvezérlőre és a PDP—11 interface-re (3.23. ábra). A regiszterek, a tároló és az aritmetikai—logikai egység (ALU) két buszhoz, a forrás- és az eredménybuszhoz kapcsolódnak.

Az *önálló működés* regiszterei két csoportba sorolhatók: a processzor egyszerre 8 programot tud időosztásos alapon kezelni, ennek megfelelően a regiszterek egyik csoportjából, az ún. fileregiszterekből 8 készlet van beépítve. A regiszterek másik csoportját mind a 8 program közösen használja. A fileregiszterek csoportjában minden programnak 2×7 darab 16-bites regisztere van. Ezek közül



3.23. ábra
Általános vezérlőprocesszor blokkvázlata

Operáció	Vezérlő	Forrás	Eredmény
----------	---------	--------	----------

Normál utasítás

Operáció	Feltétel	Cím
----------	----------	-----

Feltételes ugrás utasítás

Operáció	Cím
----------	-----

Tárolóreferenciás utasítás

Operáció	Mód és irány	Lépések száma
----------	--------------	---------------

Léptető utasítás

3.24. ábra
Általános vezérlőprocesszor utasításformátumai

a vezérlőregiszternek speciális hardware funkciója is van, a többi programozható forrásként és eredményként egyaránt. A közös regiszterek közé tartozik az utasításszámláló, utasításregiszter, a T jelű aritmetikai regiszter és a programválasztó, programbejelentkező regiszterek. Ez utóbbiak őrzik az egyes programok futásra való bejelentkezését és jelölik ki az éppen futó programot.

A forrás- és eredménybuszokra kapcsolódik az írható—olvasható tároló. Ez minimálisan 256, maximálisan 4096 szóból áll és tartalmazza mind a programot, mind pedig az adatokat.

Ugyancsak erre a buszra kapcsolódnak a CAMAC csoportvezérlő és a PDP—11 interfaceregiszterei. A csoportvezérlőnek 3 regisztere van: CAMAC címregiszter, amely a CNAF kódokat tartalmazza (16 bit, F8-at az utasításregiszter őrzi), 24-bites adatregiszter és 24-bites osztályozott LAM (GL) regiszter.

A processzor szóhossza 16 bit, utasításformátuma a 3.24. ábrán látható. Az utasítások négy nagy csoportba oszthatók: normál, feltételes ugrás, tárolóreferenciás és léptetőutasítások. A *normál utasítások* valójában két, egymás után következő műveletet jelentenek: először az operációban meghatározott műveletet, majd az ún. vezérlőfunkciót hajtja végre a processzor. Az operációban meghatározott egyoperandusú műveletek operandusa a forrásregiszterben, a kétoperandusú műveletek egyik operandusa a forrásregiszterben, másik a T aritmetikai regiszterben van. Az eredmény mindig a kijelölt eredményregiszterbe kerül, valamint két operandus esetén a T regiszterbe. A normál utasítások készlete a következő: átírás, inkrementálás, dekrementálás, kettes komplementű összeadás, kivonás, inkluzív és exkluzív VAGY, logikai ÉS, aritmetikai (T) regiszter átírás. A normál utasítások második fázisaként végrehajtott vezérlőművelet feladata hármas: e műveletekkel lehet egyrészt a CAMAC utasításokat kiadni, másrészt ezekkel lehet a kapcsolódó számítógép memóriáját közvetlenül elérni. Három CAMAC vezérlőutasítás van: normál (F8=0) CAMAC művelet, vezérlő (F8=1) CAMAC művelet és alaphelyzetbe hozó (BZ) művelet. A vezérlőutasítások harmadik csoportja befejező műveleteket, prioritásváltást, programváltást stb. végez. A normál utasításokban levő forrás- és eredményregiszter néhány kivételtől eltekintve bármely fileregiszter, CAMAC csoportvezérlő vagy PDP-11 interfaceregiszter lehet. A tároló is lehet forrás, vagy eredményhely, azonban ilyenkor a normál utasítás vezérlési művelete elmarad.

A *feltételes ugrás* utasítások a FELTÉTEL mezőben meghatározott vizsgálat eredményétől függően az utasításslámlóba a CÍM mezőben levő számot helyezik.

A *tárolóreferenciás utasítások* közül a *Tárolás* és *Kioltvasás* utasítások a CÍM-mel megcímzett tárolószó és a T aritmetikai regiszter közötti átvitelt teszik lehetővé. Ezen kívül feltétel nélküli ugrás utasítások tartoznak ebbe a csoportba. A *léptetőutasítások* a kisgépeknél szokásos típusúak.

A példaképpeni vezérlőprocesszor a felsorolt, önálló műveleteken túlmenően *számítógéphez is tud kapcsolódni*. A számítógép programozott utasításokkal lépésenként tudja vezérelni a processzort. Ekkor a működés olyan, mintha a processzor egylépéses üzemmódba lenne átkapcsolva, és a számítógépben levő program „kapcsolgatná” a processzor kezelőszerveit és lépésenként hajtáná végre a programot. Ez az üzemmód a kezdeti programbetöltéskor használható előnyösen, valamint programok, rendszerek kipróbálásakor. A másik számítógépes kapcsolat a processzor önálló működésekor válik lehetővé. Ilyenkor ugyanis a processzorban levő utasításkészlet segítségével (normál utasítások vezérlőműveletei) a vezérlőprocesszor közvetlenül el tudja érni a PDP-11 autonóm csatornáján keresztül a számítógép memóriáját.

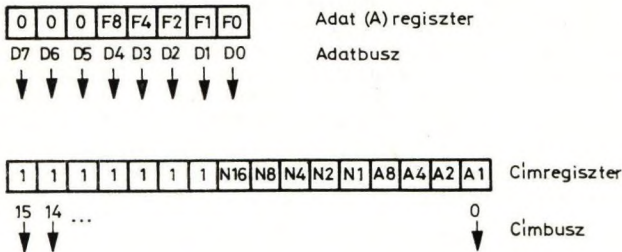
A másik megoldáshoz Intel 8080 mikroprocesszort használnak a vezérlőben. A kezdeti mikroprocesszoros vezérlők inkább kísérleti célokra készültek, a 8080 megjelenése után sok helyen kezdték alkalmazni a most már kereskedelmi forgalomba kerülő vezérlőprocesszorokba.

A példaképpeni Intel 8080 mikroprocesszorral készült rendszer [3] közvetlenül egy keretet tud vezérelni. A keretbe helyezett csoportmeghajtó segítségével lehet a vezérlést több keretre kiterjeszteni. Az alap vezérlőmodul két egység széles; tartalmazza a 8080 köré épített általános processzort, 4k tárolót, az ADATÚT programozott vezérlését ellátó áramköröket és egy UART-tal készült soros periféria (pl. írógép) illesztőt. Az almodul önállóan is sok célra használható, kiegészíthető azonban a következő további modulokkal: tárolómodulok (a 8080-hoz kapcsolható közvetlenül címezhető PROM vagy RAM tároló max. 64 kbyte lehet), csoportmeghajtó modul (ha egynél több keretet kell vezérelni), autonóm működést megvalósító modul és a kezelőszerveket, kijelzéseket tartalmazó vezérlő—kijelző előlap. Maga a vezérlőmodul csak igen kevés kezelőszervet tartalmaz. Ezek közül legfontosabbak a megállításkapcsoló és a Reset és Task nyomógombok. A 8080 lehetővé teszi, hogy külső jellel vagy nyomógombbal 8 különböző címről induljon a program. Ezek közül az egyik (Reset) alaphelyzetbe hozza a processzort és a 0. címről indítja a programot. A vezérlőprocesszor tárolójában egy kis operációs rendszer van, amely a program kipróbálásához, javításához szükséges kis programokat tartalmazza. A Task nyomógombok hívják az operációs rendszer megfelelő részeit, ill. az írógép párbeszédet lefolytató programrészt, és a próbák, javítások írógépen keresztül végezhetők el.

A vezérlőprocesszor három 8-bites adatregisztert és egy állapot-vezérlő regisztert tartalmaz. E regisztereknek perifériacímei vannak, a 8080 ki-, ill. bemeneti utasításaival lehet őket betölteni, ill. kioltvasni. A CAMAC címeket (N, A) tárolócímként kezeli a processzor. Ez azt jelenti, hogy a tároló utolsó 512 helye nem használható tárolóként, hanem ezek a címek tartalmazzák a CAMAC címeket. A 8080 mikroprocesszor tárolóreferenciás utasításnál 16-bites címvezetéken adja ki a címet és 8-bites adatvezetéken az adatot. A 16-bites cím első 7 bitje jelzi, hogy CAMAC címről van szó, a többi az N és A értékeket tartalmazza (3.25. ábra). Az adatvonalon a funkciókód (F(0)...F(31)) kerül kiadásra. Adatátvitelt végző utasításnál a CAMAC modulokba átkerülő, ill. onnan érkező adatok a

3 × 8 bites adatregiszterbe kerülnek. Ha a vezérlő a címbitekből megállapította, hogy CAMAC utasításról van szó, CAMAC ciklust indít el.

A mikroprocesszoros vezérlő adatátvitel nélküli CAMAC utasítást egy (előkészítéssel együtt két) 8080 utasítással hajt végre. Olvasáskor egy CAMAC utasítás után három perifériás utasítással ki kell olvasni az adatregisztert (előkészítéssel együtt 24-bites, szó esetében összesen 8 utasítás). Íráskor három perifériás utasítással be kell írni az adatregisztert, majd CAMAC utasítással az adatregiszter tartalma kijut a megfelelő CAMAC modulba (előkészítéssel együtt összesen nyolc utasítás). Egy teljes CAMAC utasítás végrehajtásának időtartama, előkészítő utasításokkal együtt 10...50 μs körül van, az utasítás típusától és az átvendő adatszó hosszától függően.



3.25. ábra
Intel 8080 mikroprocesszor adat- és címbuszának jele.
CAMAC utasítás esetén

3.4.

Speciális vezérlők

Az eddigiekben bemutatott, különböző típusú vezérlőkkel tulajdonképpen szinte minden CAMAC vezérlési probléma megoldható. Néhány speciális alkalmazás esetén azonban felmerülhet olyan szempont, amelynek kielégítése egy — külön a speciális feladatra készített — testre szabott vezérlővel jobban megoldható, mint egy általános vezérlővel. Példaként röviden bemutatunk két speciális vezérlőt.

CAMAC berendezések (modulok, vezérlők) mérésére, ellenőrzésére készítettek olcsó, írógéppel vezérelt önálló csoportvezérlőt [20]. (Régebben e célra kézi kapcsolókból és kijelző lámpákból álló bemérő vezérlőket használtak.) A csoportvezérlőben a CAMAC címet (C, N, A, F) és külön az írás és olvasás adatot tartalmazó regiszterek vannak. Ezeket a regisztereket úgy lehet betölteni, hogy az írógépen leütjük a regiszter betűjelét, majd utána írjuk a beírandó tartalmat. Pontosvessző beütése CAMAC ciklust indít a regiszterek pillanatnyi tartalmával. Így például a

C1, N2, A0, F16, W7310;

begépelte sor hatására az 1. keret 2. állomáshelyén a 0. alcímre beíródik a 7310 tartalom. Ha ezután a következőt gépeljük:

F0;

akkor ugyanez a tartalom megjelenik az írógépen. A Q és X válasz értéke minden utasítás után megjelenik az írógépen. E keretvezérlőnek a felsoroltakon kívül még további igen érdekes tulajdonságai is vannak (például osztályozott LAM kezelés stb.).

Egy másik, PDP—8-hoz kapcsolódó speciális keretvezérlőt sebességi problémák miatt készítettek. Részecskefizikában igen nagy adattömeget kell gyorsan átolvasni CAMAC regiszterekből a feldolgozó számítógépbe. Az adattömeg érkezése statisztikus, két egymás után érkező adat közti idő igen rövid lehet. Az e feladat megoldására tervezett speciális vezérlőbe ezért egy 256-szavas puffertárat építettek be. A CAMAC modulokban levő adatok kiolvasása Address Scan vagy Stop üzemmóddal ebben a puffertárban történik, majd következő fázisban a puffertár kiürül a PDP—8 autonóm csatornáján keresztül.

Irodalom a 3. fejezethez

- [1] *Bercaw, R. W.—Fessler, T. E.—Arnold, I. M.*: A Programmable Computer Interface for CAMAC. NASA Technical Note, NASA TN D—7148, 1973.
- [2] *Bevington, P. R.*: A Programmers Guide to the Microprogrammed Branch Driver for the PDP—11. LA—5028—MS Report, 1972. Los Alamos, Scientific Lab.
- [3] *Biri J. és mások*: A KFKI-ban kifejlesztett intelligens keretvezérlő (ICC) ismertetése. VIII. International Symposium on Nuclear Electronics; Dubna, 1975.
- [4] *Biswell, L. R.—Rajala, R. E.*: A Microprogrammed Branch Driver (MBD) for a PDP—11 Computer. LA—4916—MS Report, 1972. Los Alamos Scientific. Lab.
- [5] CAMAC—PDP—11 Data and Control Modules, type PTI—11 C. GEC—Elliott Process Automation Ltd., Lewisham, London.
- [6] CAMAC Type L—1 Crate Controller, Model 3950 Kinetic Systems Corporation, Mary Knoll Drive, Lockport, Ill. USA.
- [7] *Cohn, C. E.*: Considerations in the Design of CAMAC—Oriented Processors. CAMAC Bulletin, No 6., p. 30., 1973.
- [8] Crate Controller for NOVA, type CC 2023. SEN Electronique 1211 Geneve 13...31, Av. Ernest—Pictet.
- [9] *Halling, H.*: CAMAC Serial System with Programmable Crate Controller. IEEE Trans. on Nucl. Sci. NS—21. No 1., p. 886., 1974.
- [10] *Halling, H.—Zwoll, K.—Müller, K. D.*: A Versatile PDP—11 CAMAC Crate Controller for Nuclear Data Acquisition and Processing. CAMAC Bulletin, No 2., p. 17., 1971.
- [11] *Heldam, G.*: Ein CAMAC Branch Kontroller (PROM—BDC) für programmierbare Festwertspeicher. HMI—B 128 Report, 1973. Hahn Meitner Institut für Kernforschung, Berlin.
- [12] *Hellmann, G.—Ottes, J. G.*: An Efficient CAMAC Single—Crate Controller for PDP—8/E. CAMAC Bulletin, No 7., p. 15., 1973.
- [13] *Lyon, W. T.*: An Evaluation of CAMAC Equipment in an Industrial Environment. Proc. First Int. Symp. on. CAMAC, p. 211., 1974.
- [14] *Oakes, A. E.*: ALGEN—A Microprogrammable CAMAC Branch Driver/Controller. IEEE Trans. on Nucl. Sci. NS—19., No. 1., p. 696., 1972.
- [15] *Peatfield, A. C.—Spurling, K.—Zacharov, B.*: CAMAC as a Computer Peripheral Interface System. IEEE Trans. on Nucl. Sci. NS—21, No 1., p. 867., 1974.
- [16] *Toy, N. V.—Drury, D. M.—Smith, K. R. E.*: A Modular Method of Multiplexing Program Sources to Branch Drivers in CAMAC Systems. CAMAC Bulletin, No 7., p. 21., 1973.
- [17] *Ward, L. D.—Barnes, R. C. M.*: Developments in Hardware and Software for the 7025 Programmed Dataway Controller. CAMAC Bulletin, No 8., p. 15., 1973.
- [18] *Ward, L. D.—Mitchell, G. S. L.—Richards, J. M.*: A Programmed Controller in the CAMAC System. AERE—R6334 Report. Harwell, 1970.
- [19] *Whitehead, N. P.*: A Parallel Processing Computer for CAMAC—Computer Systems. AERE—R6817 Report, Harwell, 1971.
- [20] *Kollbach, D.*: A Teletype-Controlled CAMAC Branch Driver. CAMAC Bulletin, No 9., p. 10., 1974

4.

Jellegzetes CAMAC modulok

A 2. fejezet, a szabványok, jelutak leírása a CAMAC rendszer keretét, formáját írta le. A jelenlegi modulokról szóló rész vázolja a tulajdonképpeni lényeget, tartalmat: a felhasználói modulok végzik el a tényleges feladatot, ezek kapcsolják össze, illesztik a külső világhoz (fizikai, kémiai stb. méréshez, technológiai folyamathoz) a számítógéprendszert.

E modulok általános felépítésére szintén vannak szabványok, ajánlások, ill. szokások, ezekkel foglalkozunk a fejezet első részében. A modulok tartalmára nézve, tehát arra, hogy mi a funkciójuk, milyen feladatot látnak el, nincs előírás. Az alkalmazások során a gyakorlatban igen sok célra terveztek modulokat. A különböző típusú, gyártmányú modulok száma 1974 végén 1000 körül volt, kb. 60 különböző cég gyártásában. Mivel ma már a legtöbb alkalmazáshoz rendelkezésre áll a megfelelő modul, a típusválaszték növekedési üteme csökkent, bár ma is készülnek új modulok az alkalmazási területek bővülése, ill. a technológiai lehetőségek változása miatt.

Azért, hogy a rendszertervezők könnyebben eligazodhassanak az igen széles modulválasztékban, nemzetközi, állandóan felrészített katalógus áll a felhasználók rendelkezésére, a CAMAC Bulletin c. folyóiratban. E katalógus a CAMAC műszerezés és *software* decimális osztályozása alapján készül (l. Függelék 2. fejezet [4], [5], [16]). Ez az osztályozás a hardware elemek közül rendszerezi a felhasználói adatmodulokat, vezérlőket, vizsgáló és ellenőrző berendezéseket, továbbá a kereteket, tápegységeket, mechanikai tartozékokat. A fejezet második részében ennek alapján áttekintést adunk a felhasználói modulokról.

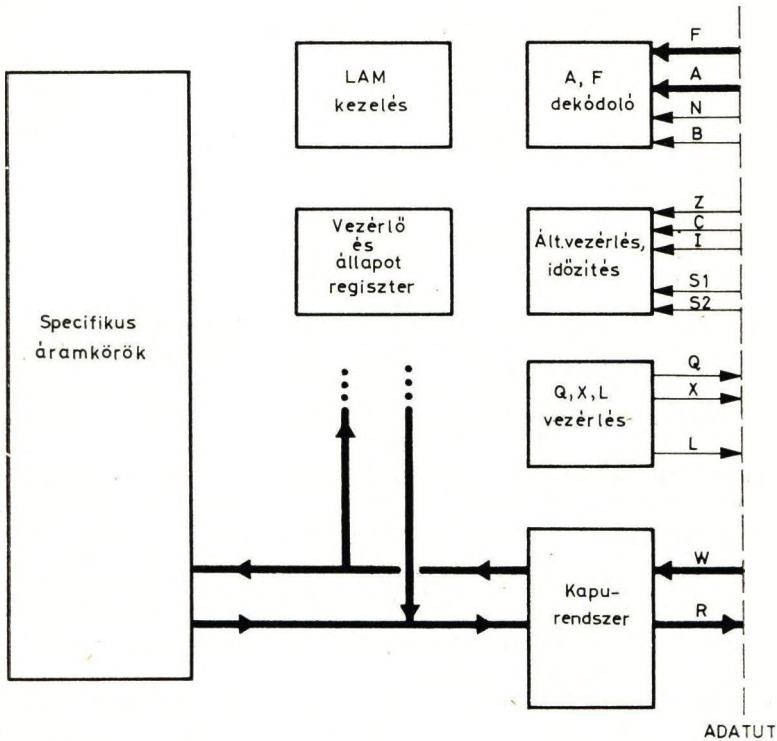
4.1.

A modulok általános felépítése

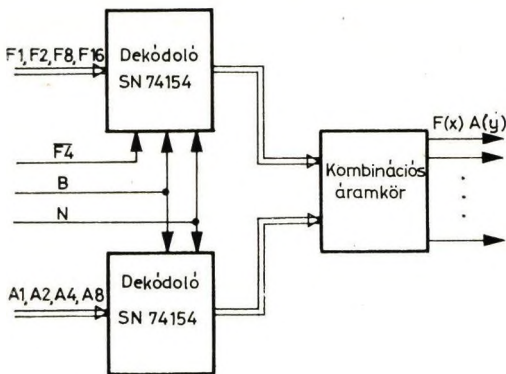
A CAMAC modulok belső felépítése általában három, jól széttagolható részre osztható (4.1. ábra):

1. az ADATÚT-hoz kapcsolódó áramkörök;
2. LAM kezelés- és vezérlő állapotregiszter;
3. specifikus áramkörök.

Az ADATÚT-hoz kapcsolódó áramkörök közös feladata az ADATÚT jeleinek fogadása, ill. az ADATÚT-ra kijutó jelek illesztése. Logikailag négyféle feladatot látnak el: binárisba kódolt alcím (A) és funkció (F) dekódolása, az általános vezérlő- és időzítő-jelek fogadása és továbbítása, valamint az olvasó—író adatvezetékek kapuzása. A szabvány előírja, hogy az A és F dekódolás teljes legyen. Ez azt jelenti, hogy minden dekódolt



4.1. ábra
CAMAC modul általános felépítése (belső vezérlőjelek nélkül)

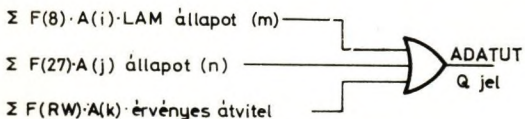


4.2. ábra
A—F dekódolás felépítése

A, ill. F jelhez csak egyetlen egy bemenő A, ill. F kódkombináció tartozik. A ma használatos áramkörökkel példaképpen a leggyakoribb megoldást a 4.2. ábrán láthatjuk. Egy-egy integrált áramkörrel külön-külön dekódolják az A1, A2, A4 és A8, valamint F1, F2, F8 és F16 jeleket, azután kapuzzák a B és N jelekkel, majd egy további kombinációs áramkörben összeállítják a modul által használt $F(x) A(y)$ funkció—alcím kombinációkat. Ezek az $F(x) A(y)$ jelek végzik el a modulban szükséges további kapuzásokat. A példában használt integrált áramkör csak 16 kombinációt tud fogadni, ez az 5-bites F kód dekódolásra kevés. A legtöbb esetben azonban a modul csak a 16 szabványosított F kódot használja, ezeknél $F_4=0$. A példakénti megoldás ennek megfelelően csak az F1, F2, F8 és F16 jeleket dekódolja, az F4 jellel pedig kapuzza a dekódolt jelet.

A Q és X jelek összeállítása igen egyszerű feladat. Q választ három fő ok miatt küld a modul az ADATÚT-ra:

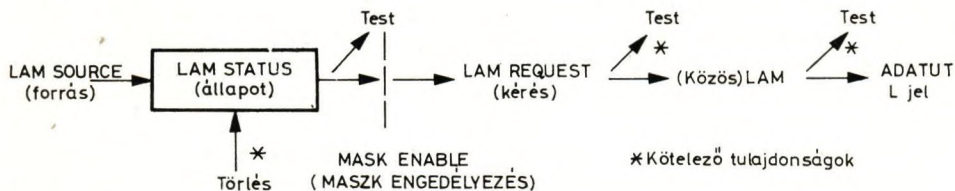
1. a LAM állapotot vizsgáló $F(8) A(i)$ utasítás eredményeképpen;
2. valamely egyéb állapotot vizsgáló $F(27) A(i)$ utasítás hatására;
3. az 1. csoportú adatregiszterek írásakor és olvasásakor, ha érvényes volt az adatátvitel (ez nem szabványos, de gyakran szokásos). Ezeknek megfelelően e három ok VAGY kapcsolata kerül ki a Q vonalra (4.3. ábra). Az X jel létrehozása az ADATÚT-



4.3. ábra
ADATÚT Q és X jelének összeállítása

on kötelező; összeállítása egyszerű. Általában az A—F dekódoló és kombinációs áramkör kimenő $F(x) A(y)$ jeleit VAGY kapura vezetik, és ez közvetlenül adja az ADATÚT-ra kerülő X jelet (4.3. ábra). Néhány modulnál, ha szükséges, bizonyos $F(x) A(y)$ kombinációkat további külső vagy belső feltétellel ÉS kapcsolatba hoznak és az kerül a többi F—A kombinációval VAGY kapcsolatban az ADATÚT-ra.

A modulon belüli LAM kezelés már jóval bonyolultabb lehet. A modul speciális részéből érkezik a LAM forrásjel (LAM source), amely például egy számláló túlsordulását vagy konverzió befejeztét jelzi (4.4. ábra). A LAM forrás bebillenti a LAM állapot



4.4. ábra
CAMAC modul LAM kezelésének blokkvázlata

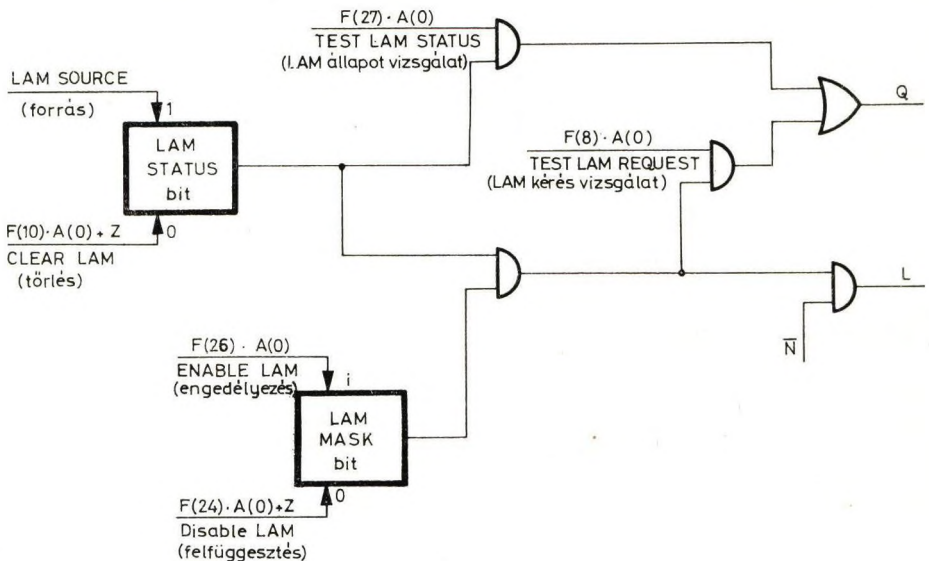
(LAM status) bistabilt. A LAM állapot bistabil kimenetét kapuzni lehet maszkbittel vagy engedélyezés-tiltás kapuval. A kapuzott LAM neve LAM kérés (LAM request). Ha több LAM forrás van, akkor ezután kerülnek a LAM kérések VAGY kapcsolatba: ez a közös LAM vagy röviden L jel, mely kijut az ADATÚT-ra. Utasításokkal vizsgálni lehet a LAM állapot, LAM kérés és közös L jelek állapotát.

A modulok nem kötelesek LAM jelet létrehozni. Az EUR 4100 szabvány csak azt írja elő, hogy ha valamelyik modul L jelet generál, akkor lehessen

- a LAM állapot bistabilt törölni;
- a közös L jel állapotát vizsgálni;
- az egyedi LAM kérés állapotát vizsgálni.

Az *Alapállapot* (Z) jel hatására a LAM állapotregiszternek törölnődnie kell, a maszkoknak, ill. a LAM engedélyezésnek tiltania kell a LAM jelet. A 4.4. ábrán a kötelező tulajdonságokat csillaggal jelöltük.

A 4.5. ábrán *egyetlen LAM forrással* rendelkező LAM kezelő áramkört mutatunk be. Az ábrán példaképpen A(0) alcímet választottunk. Ez nem kötelező, de szokásos megoldás. A funkciókódokat a szabvány kötelezően írja elő. A példaképpeni megoldás



4.5. ábra
Példa LAM kezelésre egy forrás esetében

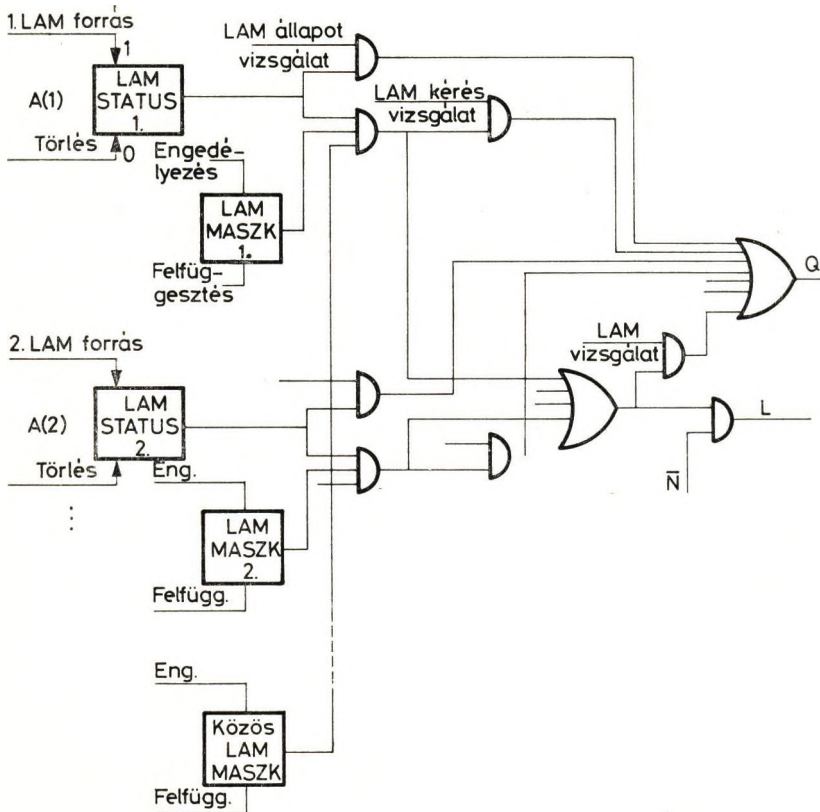
a kötelező szabványnál annyival nyújt többet, hogy a LAM állapotot is lehet vizsgálni. Az ábrával kapcsolatban még megjegyezzük, hogy a LAM állapot törlésére az itt feltüntetett megoldásokon kívül esetleg továbbiak is célszerűek: a modullal kapcsolatos bizonyos utasítások, például egy A–D konverter eredményének kiolvasása szintén törölheti a LAM állapotot.

Egy modulban természetesen *több LAM forrás* is lehet, ekkor a fenti alapelveket értelemszerűen kell alkalmazni. Minden forráshoz tartozik egy-egy LAM állapot bistabil. Ezek kimenete külön-külön engedélyezhető, tiltható, ill. maszkolható. Ezen kívül egy közös LAM engedélyezés-tiltás is van. A közös L jel kerül ki azután

az ADATÚT-ra. Ezen elvek gyakorlati megvalósítására az ajánlás két lehetőséget ad:

1. az alcímmel (A) való LAM kezelést;
2. közös regiszterekben bit pozícióval való LAM kezelést.

Az *alcímmel való LAM kezelés* tulajdonképpen az egy LAM forrásos megoldás egyszerű bővítése a többszörösítésre, a LAM források számának megfelelően. Minden LAM forráshoz tartozik egy-egy alcím, és az illető LAM úthoz tartozó minden utasítás ezzel az alcímmel kerül végrehajtásra. A közös részekhez szintén tartozik egy további alcím. Példamegoldást a 4.6. ábrán láthatunk. Az alcímmel való LAM kezeléskor csak az F kódok vannak szabványosítva, az alcímek nem, még a közös részekhez tartozó

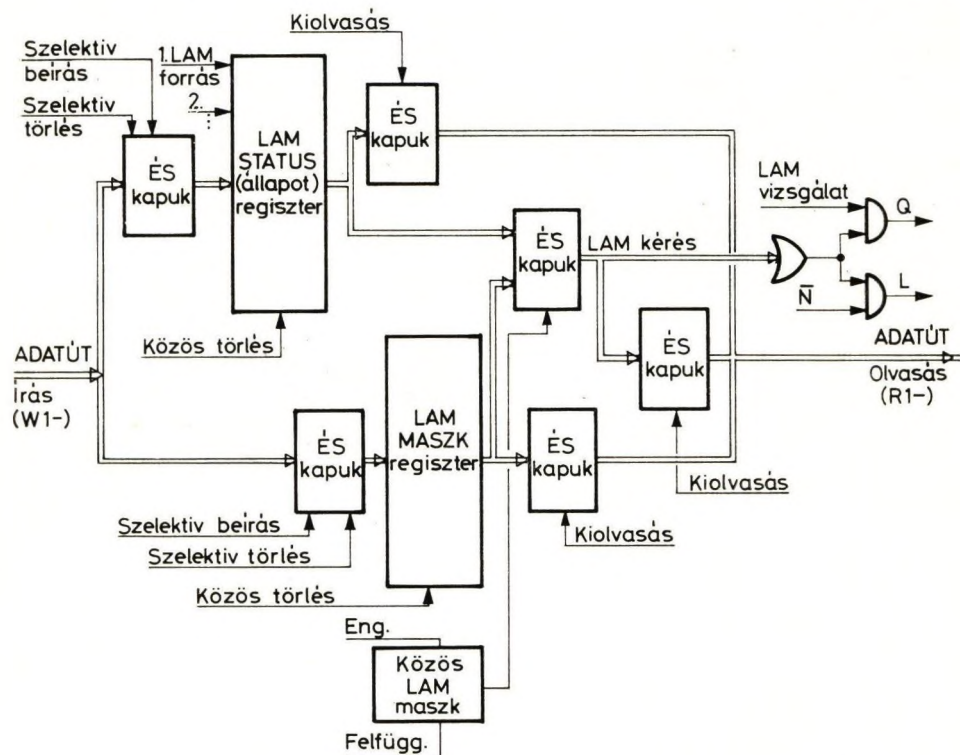


Egyedi LAM forrás	Egyedi LAM állapot regiszter	Egyedi + közös LAM maszk regiszter	Egyedi LAM kérés vizsgálat	Közös LAM vizsgálat
Törlés: $F(10)A(i)+Z$ Vizsgálat: $F(27)A(i)$	Engedélyezés: $F(26)A(i)$ Felfüggesztés: $F(24)A(i)+Z$		Vizsgálat: $F(8)A(i)$	Vizsgálat: $F(8)A(k)$

4.6. ábra
Példa az Alcímmel (A) való LAM kezelésre

alcím sem. A közös részek alcíme szokás szerint A(0) vagy A(15). Az alcímmel való LAM kezelést akkor célszerű használni, ha kevés LAM forrás van.

Sok LAM forrás esetében célszerű a LAM állapot és maszk bistabilokat egy-egy közös regiszterre összefogni. A közös regiszterekben egy-egy LAM forrásnak egy-egy bit felel meg. Az egyedi LAM törlések, engedélyezések és vizsgálatok helyett a megfelelő regisztereket kell beírni, törölni, ill. kiolvasni. A LAM állapotregiszter alcíme kötelezően A(12), a maszkregiszteré A(13), és a 2. csoportba tartoznak. A 4.7. ábrán látható példamegoldás esetében mindkét regiszter szelektíven beírható, szelektíven törölhető és kiolvasható. Ezeknél a műveleteknél természetesen az ADATÚT Olvasó—Író adatvezetékein jut át az információ. Mindkét regiszter ezeken felül még közösen



<u>LAM forrás</u>	<u>LAM állapot regiszter</u>	<u>LAM maszk regiszter</u>	<u>Közös LAM maszk</u>	<u>LAM kérés A(14)</u>	<u>Közös LAM</u>
	A(12)	A(13)			
Szelektív beírás	F(19) A(12)	Szelektív beírás	Engedélyezés	Kiolvasás	Vizsgálat
		F(19) A(13)	F(26) A(k)	F(1) A(14)	F(8) A(k)
Szelektív törlés	F(23) A(12)	Szelektív törlés	Feltüggesztés		
		F(23) A(13)	F(24) A(k)-Z		
Közös törlés	Közös törlés	Közös törlés			
	F(11) A(12)+Z	F(11) A(13)+Z			
Kiolvasás	Kiolvasás	Kiolvasás			
	F(1) A(12)	F(1) A(13)			

4.7. ábra

Példa közös regiszterekben bit-pozícióval való LAM kezelésre

is törölhető. A maszkolt LAM kéréseket szintén regiszternek tekinthetjük. Ennek alcíme A(14) és szintén kiolvasható. További közös utasítások a közös LAM engedélyezés-tiltás és a közös LAM vizsgálat. Ezek az utasítások már az alcímmel való LAM kezeléshez hasonlóan működnek. Nincs szabványos alcímük, de célszerűen az A(0) vagy A(15) alcímet veszik igénybe. Az általános kötelező előírások itt is érvényesek. A LAM állapot bistabil törlésének megfelel a szelektív törlés utasítás, az egyedi LAM kérés állapotvizsgálatának pedig LAM kérés kiolvasás: ezek tehát kötelezőek.

Az ADATÚT-ra kijutó *L jelet kapuzni* kell. A szabvány előírja, hogy ha egy modul, amely L jelet ad ki egy olyan utasítást kap, amely megszünteti ezt az L jelet, akkor a modulnak még az S1 mintavevőjel előtt le kell tiltania a kimenő L jelet vagy a megfelelő LAM kérést. Ezt a követelményt a legegyszerűbben úgy lehet teljesíteni, hogy az L jelet az \bar{N} jellel kapuzzuk. Ily módon ugyanis megszűnik az L jel, ha a modul meg van címezve.

Az előbbieken leírt általános tulajdonságok valamennyi modulban megtalálhatók. Ezért tervezési, kísérleti célokra olyan modulok is készültek, amelyekbe ezek az általános jellemzőket megvalósító áramkörök be vannak építve, míg a modulban levő kártya további része üres, mellyel a tervező, ha szükséges, gyorsan meg tudja valósítani a modul speciális tulajdonságait [7].

Az általános tulajdonságokon felül bizonyos alkalmazási területeken vagy egy-egy laboratóriumon, cégen belül esetleg további, *belső ajánlásokat* is tesznek a modulok felépítésére nézve. Példaképp bemutatjuk a CERN-ben belső ajánlásként használt ún. modulmeghatározó (Module Descriptor) jellemzőit [17]. A modulmeghatározó a moduloknak rendszertechnikai jellemzőit tartalmazza egy regiszterben, amelynek tartalmát CAMAC utasítással ki lehet olvasni. A példaképpeni modulmeghatározó regiszter 16-bites, az alsó 12 bit tartalmazza a modul decimális osztályozási számát, a 12...15. bitek pedig a modulnak Q válasszal kapcsolatos tulajdonságait: címletapogatásos, ismétléses vagy megállásos üzemmódban működik-e a modul. Amennyiben van modulmeghatározó-regiszter a modulban, akkor a szabvány szerint F(1) A(15) utasítással kell kiolvasni. A modulmeghatározó az egyedi konfigurációhoz tartozó egyedi software-rendszer összeállítását teszi gépesíthetővé: egy összeállító program végigtapogatja az egyes állomáshelyeket, és a válaszok alapján összeállítja a szükséges kezelőprogramokat.

4.2.

Általános célú és funkcionális modulok

A következőkben a fontosabb általános célú modulokról adunk rövid áttekintést. Célunk nem az, hogy részletes tervezési módszereket írjunk le, hiszen ez az általános elektronikai szakkönyvek feladata. Inkább katalógus szerűen összefoglaljuk a modulok mai fő jellemzőit, esetleg a működési elveket, segítséget nyújtva a rendszertervezőnek.

A modulok itt következők rendszerzése bizonyos fokig önkényes, mivel egy-egy modul több címszó alá is tartozhat. Egy programmal beállítható számú impulzust kiadó generátor például a digitális soros kimeneti modulok családjába tartozik, de használható léptetőmotorok adott számú lépéssel való továbbítására is, és mint ilyen a periféria-műszerezésvezérlés kategóriába is sorolható. A továbbiakban anyagunkat a CAMAC decimális osztályozás sorrendje szerint tárgyaljuk.

A decimális osztályozásnál általánosabb felosztás két nagy csoportba osztja a modulokat: általános célú és funkcionális modulokra. Az *általános célú modulok* viszonylag

egyszerűbb felépítésűek és nincs meghatározva, hogy a méréshez, folyamathoz milyen átalakítón keresztül csatlakoznak. A hagyományos értelemben vett real-time perifériák ebbe a csoportba tartoznak. CAMAC rendszerekben kezdenek terjedni a *funkcionális modulok*, amelyek az átalakító speciális típusához közvetlenül illeszkednek, és mind a logikai, mind az elektromos jelszintillesztést egyszerűbbé teszik [20]. Mivel a funkcionális modul a speciális átalakító szempontjai szerint van kialakítva, egyrészt a csatlakozásnál már nem szükséges további illesztő áramkör, másrészt pedig a programozása egyszerűbb: a funkcionális modul hardware úton a modulban valósít meg olyan funkciókat, amelyeket egyébként egy általános modullal a vezérlő vagy számítógép programozottan, jóval hosszabb idő alatt tudna csak végrehajtani.

Funkcionális modulra mutatunk be példát [13]. Egy nyomáskülönbséget mérő átalakítónál a nyomáskülönbség mérőszámát az adja meg, hogy egy 15 s-os időtartam alatt milyen hosszú ideig van zárva egy érintkező. Általános modullal, programozottan a feladat úgy oldható meg, hogy megfelelő szinkronizálás után a felbontási idő által meghatározott sűrűséggel a program beolvassa az érintkező állapotát, és ezekből az adatokból határozza meg a nyomást. Megfelelően kialakított funkcionális modul hardware úton az érintkező zárt állapotának időtartamát binárisba kódolva továbbítja a számítógépnek. Ily módon például az 50 ms-os letapogatási idő helyett elég minden 15. másodpercben egyszer a modulhoz fordulni.

Bár a modulok tárgyalása során nem tárgyaljuk külön a funkcionális modulokat, a példák között — a decimális felosztású tárgyalásmód megfelelő helyén — ilyeneket is ismertetünk.

4.2.1.

Digitális bemeneti modulok

A digitális bemeneti moduloknál az információ alapvetően háromféleképpen juthat a modulba:

- soros, számláló bemeneten;
- párhuzamos bemeneten;
- soros, léptetéssel működő bemeneten keresztül.

Számlálók

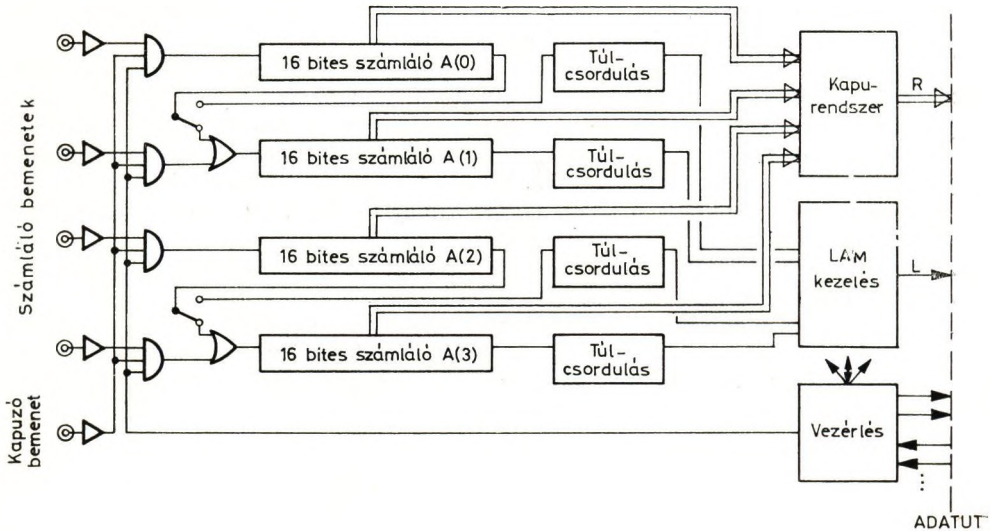
A modulok egyik nagy csoportját a számlálók képezik. A decimális osztályozás négy alcsoportba osztja ezeket a modulokat:

- egyszerű bináris számlálóregiszterek;
- egyszerű dekadikus számlálóregiszterek;
- preszet bináris számlálóregiszterek;
- preszet dekadikus számlálóregiszterek.

A számlálómodulok általában egységnyi szélesek, egy-egy modulban egy, kettő vagy négy számláló van. A számlálók hossza 16 vagy 24 bit, dekadikus esetben 4 vagy 6 dekád. A kereskedelemben forgalomban levő leggyakoribb típus 4×16 bites. Számlálási sebességük 15...200 MHz körül van, a tipikus sebesség 25...50 MHz. Bemenetük általában TTL szintű. A számlálóbemenetet külső jellel vagy *Adatút* utasítással kapuzni lehet. A több számlálót tartalmazó moduloknál a kapuzás a legtöbb esetben közös az összes számlálóra nézve. Az egyes számlálók túlsordulása LAM jelet generál. Több

számlálót tartalmazó modulok esetében igen sokszor két számlálót egymás után lehet kapcsolni és így kétszeres hosszúságú számlálóként használni.

A 4.8. ábrán látható példaként egy számlálómodul vázlata és utasításkészlete [19]. A modul négy különálló bináris, 16-bites számlálót tartalmaz, a túlsordulások LAM jelet keltenek.



Utasítások:	F(0) A(k)	olvasás
	F(2) A(k)	olvasás és törlés
	F(9) A(k)	regiszter és LAM állapotbit törlés
	F(10) A(k)	LAM állapotbit törlés
	F(25) A(k)	+1 hozzászámolás
	F(1) A(12)	LAM állapotregiszter olvasás
	F(8) A(12)	LAM vizsgálat
	F(24) A(12)	LAM felfüggesztés
	F(26) A(12)	LAM engedélyezés
		k=0, 1, 2, 3

4.8. ábra
Számlálómodul vázlata és utasításai

Digitális párhuzamos bemeneti modulok

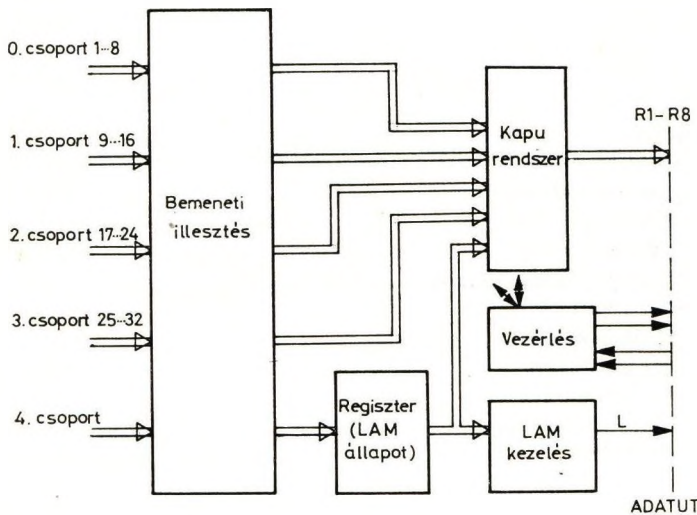
A digitális párhuzamos bemeneti modulok öt alcsoportba osztályozhatók:

- bemeneti kapuk (puffer nélkül);
- bemeneti regiszterek (pufferregiszterrel);
- jelátalakítást végző bemeneti regiszterek;
- kézi bemeneti modulok;
- egyéb párhuzamos bemeneti modulok.

A legjobban elterjedtek a *puffer nélküli bemeneti kapuáramkörök*. Egy-egy, általában egységnyi széles modulban 16...256 bemeneti kapu van; a vezérlőnek, ill. a számítógépnek 8-, 16- vagy 24-bites szavanként olvasható be a kapuk bemenetén levő információ. A leggyakoribb a 2×16 vagy 2×24 bites kapuáramkörrendszer.

A kapuáramkörrendszerek különböző bemenő jelszintekre készültek. Gyakori a TTL szintű bemenet, de újabban terjednek az egyenáramon leválasztott bemenetek jelfogós, ill. optoelektronikus leválasztással. Több modulban a gyors zavarjelek elleni védekezés miatt a bemenő jeleket integrálják.

A bemeneti kapuáramkörrendszerek szinte minden esetben olyan, logikai jelek fogadására, ill. adására szolgáló áramkörökkel vannak kiegészítve, amelyek megkönnyítik a külső eszköz és a számítógépprogram szinkronizálását, időbeni megfelelő kapcsolattartását. A leggyakoribb esetben külön digitális bemenetre vagy bemenetekre adott külső jellel LAM-ot lehet kelteni. Ezt célszerűen úgy alkalmazzák, hogy amikor a bemeneti kapuáramkörre a külső egység a megfelelő jeleket ráadja, a LAM jelet generáló bemenetre is jelet ad és ezzel jelzi a vezérlőnek, ill. a számítógépnek, hogy az vegye át a bemeneten levő adatot. E módszer továbbfinomított változatánál a kapurendszeren levő adatokat beolvasó ADATÚT utasítás akkor ad $Q=1$ választ, ha a LAM jel fennáll. Ezáltal a beolvasó utasítás Q válaszában vizsgálatával el lehet dönteni, hogy érvényes adat került-e beolvasásra. Néhány modulban megoldották azt is, hogy a számítógép a külső egységet értesíteni tudja arról, hogy új adat kerülhet a kapuáramkörök bemenetére.



Utasítások: F(0) A(k)	k. csoport olvasás
F(8) A(0)	LAM vizsgálat
F(10) A(0)	LAM állapottörlesztés
F(24) A(0)	4. csoport 1—4. bitek LAM felfüggesztése
F(24) A(1)	4. csoport 5—8. bitek LAM felfüggesztése
F(26) A(0)	4. csoport 1—4. bitek LAM engedélyezése
F(26) A(1)	4. csoport 5—8. bitek LAM engedélyezése
	$k=0, 1, 2, 3, 4$

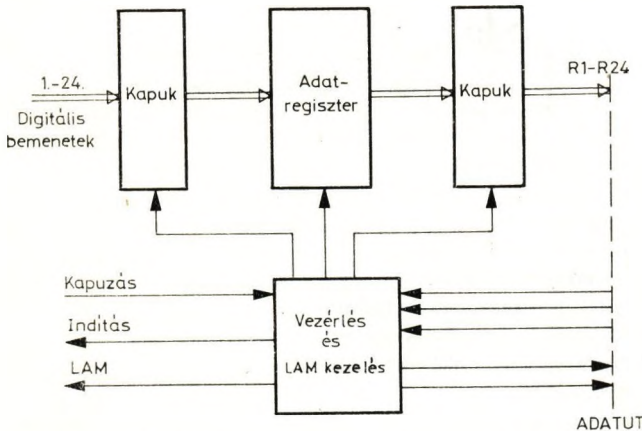
4.9. ábra
Kapuáramkör-rendszer modul vázlata és utasításai

A 4.9. ábrán bemeneti kapurendszer modul vázlata és utasításkészlete látható [9]. A modul 32 sztatikus digitális információbitt tud a vezérlőegységnek, ill. a számítógépnek továbbítani, 8-bites csoportokban. Ezen felül további 8-bites bemenete van. Az ezen érkező információ a pufferregiszterbe kerül és LAM jelet kelthet az ADATÚT-on. A modul egységnyi széles. Kártyája két részből áll: a bemeneti illesztőrészből és a CAMAC ADATÚT-hoz csatlakozó vezérlésből, kapurendszerből. A bemeneti illesztés — az alkalmazói körülményeknek megfelelően — négyféle lehet:

1. TTL jelek fogadására alkalmas bemenet.
2. 24 V-os feszültségű, ún. magas szintű logikai jelek (High level logic) fogadására kiképezett bemenet.
3. Opto csatolással felépített, potenciálfüggetlen bemenet.
4. Külső mintavevőjellel bekapuzott, puffertárolóval ellátott bemenet.

Elég gyakori a *pufferregiszterrel rendelkező, párhuzamos bemeneti* modulok használata is. Egységnyi széles kártyán egy- vagy kétszer 16- vagy 24-bites bemenetet valósítanak meg. A bemenetek és a külső eszközzel való logikai kapcsolat hasonló a bemeneti kapurendszerekéhez. A külső információt a regiszterbe általában vagy külső kapuzójellel, vagy CAMAC utasítással lehet bekapuzni.

A 4.10. ábra 24-bites bemeneti pufferregisztert mutat be [18]. A modulnak az az érdekessége, hogy többféleképpen lehet vezérelni a regiszter betöltését a külső bemenetek felől. A különböző vezérlési módokat a kétbites vezérlőregiszterben lehet beállítani. A legegyszerűbb az, amikor ADATÚT utasítás kapuzza az adatokat a regiszterbe, majd külön, további utasítással lehet kiolvasni a regisztert az ADATÚT-on keresztül a számítógépbe. A második esetben külső kapuzójel hatására íródik be a regiszter. A beírással egyidőben LAM jel keletkezik. Utasítással lehet kiolvasni a regiszter tartalmát, amely egyben törli is a LAM állapotregisztert. A LAM jel ki van vezetve az előlapra is, így a külső egység értesül arról, hogy mikor vette át az adatot a számítógép. Ezen felül a számítógép ilyenkor az F(25) utasítással az INDÍTÁS csatlakozóra jelet ad, amely jelezheti a külső egységnek, hogy további

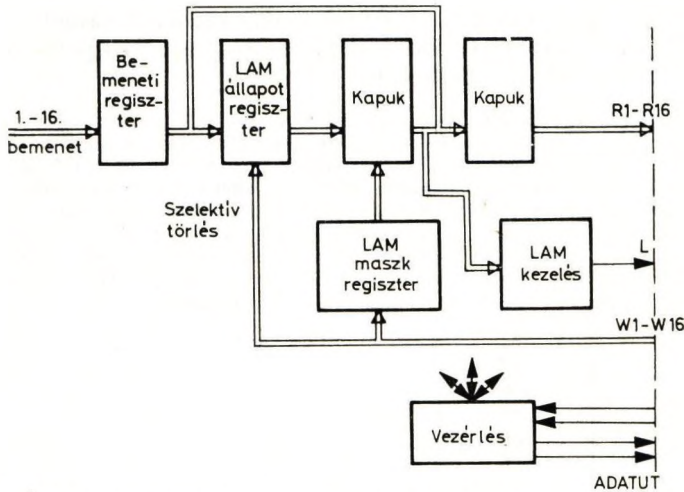


Utasítások: F(0) A(0)	regiszter olvasás és LAM törlés
F(9) A(0)	regiszter törlés
F(25)	digitális bemenetek kapuzása a regiszterbe vagy indítás kifelé
F(8) A(0)	LAM vizsgálat
F(10) A(0)	LAM állapotörlesztés
F(24) A(0)	LAM felfüggesztés
F(26) A(0)	LAM engedélyezés
F(27) A(0)	LAM forrásvizsgálat
F(17) A(11)	vezérlőregiszter beírás

4.10. ábra
Bemeneti regiszter modul vázlata és utasításai

adatot adjon. A harmadik működési mód esetén a regiszterkiolvasás utasítás előbb bekapuzza a digitális bemeneteket az adatregiszterbe, majd kiolvassa az adatregisztert a számítógépbe. Ilyenkor sem keletkezik LAM jel, az F(25) utasítás pedig hatástalan. A negyedik működési mód az első kettő kombinációja: amíg a kapuzójel a modul bemenetén van, addig a modul a második mód szerint működik. A kapuzójel végén viszont az adatregiszter betöltődik és LAM jel keletkezik.

A jelátalakítást végző bemeneti regiszterek közé elsősorban a nukleáris technikában használt gyors koincidenca, gyors mintavevő modulok tartoznak. Logikai felépítésük hasonló az előzőekben tárgyalt párhuzamos bemeneti regiszterekéhez, azonban áramköri felépítésük — főleg a nagyobb sebességigények miatt — sokkal bonyolultabb azoknál. A modulok funkciója az, hogy gyors, külső mintavevőjelre bekapuzzák



Utasítások: F(1) A(0)	bemeneti regiszterolvasás
F(1) A(14)	LAM kérés olvasás
F(8) A(0)	LAM vizsgálat
F(11) A(12)	LAM állapotregiszter törlés
F(11) A(13)	LAM maszkregiszter törlés
F(19) A(13)	LAM maszkregiszter szelektív beírás
F(23) A(12)	LAM állapotregiszter szelektív törlés
F(23) A(13)	LAM maszkregiszter szelektív törlés
F(24) A(0)	LAM felfüggesztés
F(26) A(0)	LAM engedélyezés

4.11. ábra
Megszakító regiszter vázlata és utasításai

a párhuzamos bemeneteken levő digitális jeleket. A bemenő jelek a ns-os tartományba esnek, jelszintjük általában néhány száz mV nagyságrendű.

Laboratóriumi mérések során vagy automata mérőberendezéseknél a paraméterek beállítására gyakran használnak kézi kapcsolósávot tartalmazó modulokat. Ezek a modulok vagy kétállású, vagy tízpozíciójú kapcsolókat és gyakran nyomógombot tartalmaznak. A kapcsolóban beállított értékeket ADATÚT utasítással lehet beolvasni, a nyomógombok LAM jelet keltenek. Újabbán a félvezető RAM tárolók felhasználásával több, általában 16-szavas kézi paramétertárolókat készítenek, amelyekbe a para-

méterek értékeit kézzel, egymás után lehet beírni, és az előlapon levő kijelzőkön ellenőrizni; a számítógép az ADATÚT-on keresztül megcímezi és kiolvassa a megfelelő paramétereket.

A párhuzamos bemeneti regiszterek egyik különleges fajtája a *megszakításkérő* vagy *LAM állapotregiszterek*. E regiszterek abban különböznek az egyszerű, párhuzamos bemenetektől, hogy ha bármelyik bemenetére jel érkezik, akkor LAM jelet generálnak; ezen kívül LAM kezelése általában kifinomult: LAM maszkregiszterrel, esetleg programozható LAM beállítással és többféle vizsgálati utasítással rendelkezik.

A 4.11. ábra megszakításkérő regisztert mutat be [12]. A modulnak 16 bemenete van, amelyek egy bemeneti kapuzható (latch) regiszterre jutnak. Ha bármelyik bemenet 0-ból 1-be változik, a LAM állapotregiszter megfelelő bitje 1-be kerül. A LAM állapotregiszter kimeneteit a LAM maszkregiszter egyes bitjei kapuzzák; ezután VAGY kapcsolatba kerülve és az általános LAM engedélyezés-felfüggesztés kapun keresztül jutva LAM jelet generálnak az ADATÚT-on.

4.2.2.

Digitális kimeneti modulok

A digitális kimeneti modulok a következő két nagyobb csoportba oszthatók:

- soros kimeneti modulok;
- párhuzamos kimeneti regiszterek.

Soros kimeneti modulok

A soros kimeneti modulok családjába órajelgenerátorok, időzítőegységek és programozott impulzusgenerátorok tartoznak.

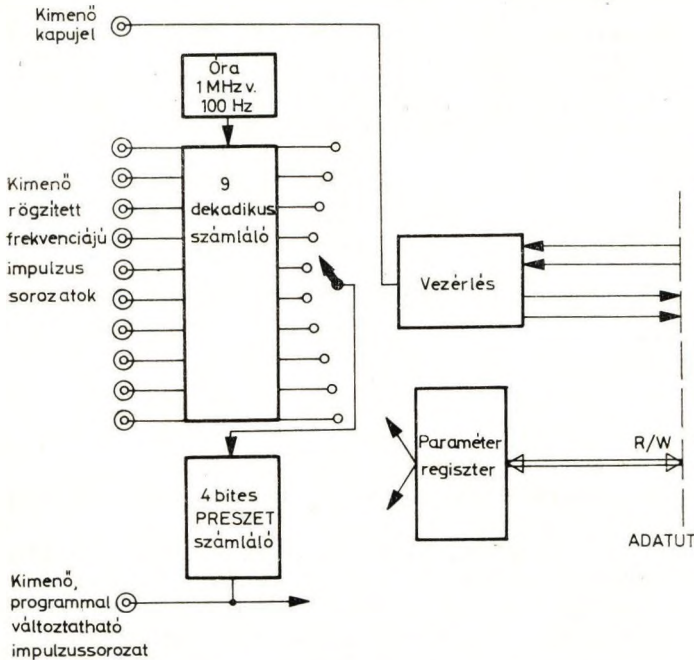
A leggyakrabban használt *órajelgenerátorok* kvarcvezérelt generátort és dekadikus leosztókat tartalmaznak. Frekvenciartományuk, azaz a dekadikus leosztott és az előlapra kivezetett impulzussorozat frekvenciája 1 Hz és 10 MHz között van; különleges modulok esetén 100 MHz is lehet. A kimenet általában TTL szintű. Programozhatóságuk nagyon különböző. A legegyszerűbb órajelgenerátorok egyáltalán nem programozhatók, egyetlen kapcsolatuk a CAMAC ADATÚT-tal az, hogy valamelyik meghatározott leosztású impulzussorozat LAM jeleket generál, például 1, 10 vagy 100 Hz frekvenciával. Kissé fejlettebb órajelgenerátorok már CAMAC utasítással kapuzhatók. Vannak bonyolultabbak, amelyeknél a kimenő impulzussorozat frekvenciáját is programmal lehet beállítani.

Példaként bemutatjuk egy bonyolultabb órajelgenerátor modul blokkvázlatát és utasításait (4.12. ábra) [6]. Az 1 MHz-es órajelgenerátor jelei kilenc egymás után kötött dekadikus számlálóra (osztóra) jutnak; az egyes osztók kimenetei az előlapra is ki vannak vezetve. Programmal beállíthatóan valamelyik dekadikus osztó kimenete egy további 4-bites bináris preszet számlálóra jut, amelynek leosztása a preszetérték beállításával szintén programozható. Ennek kimenete van kivezetve az előlapra mint programmal beállítható frekvenciájú impulzussorozat. CAMAC utasítással lehet beírni a paraméterregisztert, amely az impulzussorozat leosztásának mértékét tartalmazza, és ennek megfelelően vezérli a dekadikus számláló kimeneteket, ill. a preszet számlálóba beírandó preszetszámot. Az órajelgenerátor kétféle üzemmódban használható: vagy folyamatosan bocsátja ki az impulzussorozatot, vagy pedig szakaszosan. Szakaszos működés esetén a preszet számláló túlsordulására az órajelgenerátor leáll és LAM jelet generál. Ezután programmal lehet újra indítani az órajelgenerátort. Amíg az órajelgenerátor működik, egy kimeneti csatlakozón kimenő kapujelet ad a modul. Ily módon akár előre beállított impulzusszámot, akár időtartamot kaphatunk az órajelgenerátor modulból.

Készítettek olyan órajelgenerátor modulokat is, amelyekben a kvarcgenerátor által meghajtott impulzusosztó számlálók a másodperc, perc, óra, napnak megfelelő leoszt-

tást végzik, és a számlálóregiszterek kiolvasásakor közvetlenül a BCD kódú valós időt adják.

A programozott impulzusgenerátorok a program által meghatározott számú impulzust adnak ki, előre rögzített frekvenciával és impulzusalakokkal. Impulzusgenerátor által meghajtott preszet számlálót tartalmaznak; a preszetszám beállítása határozza meg a kiadandó impulzusok számát. A generátor külső vagy programozott utasításra indul és a számláló túlcordulásakor áll meg.



Utasítások: F(0) A(11)	paraméter regiszter olvasás
F(1) A(11)	vezérlő állapotregiszter olvasás
F(8) A(15)	LAM vizsgálat
F(9)	számlálók törlése
F(10) A(15)	LAM állapottörlés
F(16) A(11)	paraméterregiszter beírás
F(17) A(11)	vezérlőregiszter beírás
F(24) A(11)	órajel felfüggesztés
F(24) A(15)	LAM felfüggesztés
F(26) A(11)	órajel engedélyezés
F(26) A(15)	LAM engedélyezés
F(27) A(11)	órajel engedélyezés vizsgálat
F(27) A(15)	LAM forrás vizsgálat

4.12. ábra
Órajelgenerátor vázlata és utasításai

Párhuzamos kimeneti regiszterek

Ezek a modulok pufferregisztert vagy regisztereket tartalmaznak, amelyeket az ADATÚT-ról utasítással lehet betölteni és kimenetük a modul előlapján levő csatlakozóra van kivezetve. 2×16 , 2×24 vagy 4×8 bitesek. Különböző célokra különböző kimenettel ellátott modulok készülnek. Ezek közül a leggyakoribbak a TTL, a nyitott kollektoros, a magas szintű (12 vagy 24 V) és a leválasztott potenciálú jelfogó-érintkezős vagy opto-csatolású kimenetek. Több célra használható konstrukciók esetében a logikai részt és a regisztereket tartalmazó közös kártyához különböző végfokokat tartalmazó, cserélhető további kártya csatlakozik, természetesen egy CAMAC modulba összeépítve. Az egyszerűbb modulok utasításrendszere egy-két utasításból áll: ezekkel a regisztereket csak betölteni és esetleg törölni lehet. Sok esetben LAM jelet nem is hoz létre a modul. Bonyolultabb utasításrendszereknél a regiszterek tartalmát ki is lehet olvasni. Néhány modul a külső eszközzel való kapcsolat szinkronizálását is lehetővé teszi: ezeknél a modul kimenő vezérlőjellel jelezni tudja a külső perifériának, hogy új adatot írt be a regiszterbe, a külső periféria pedig LAM jelet tud keltetni a modulban, értesítve a számítógépet, ill. a programot arról, hogy átvette az adatot.

4.2.3.

Periféria- és műszerezésvezérlő modulok

A periféria- és műszerezésvezérlő modulok a következő főbb csoportokba sorolhatók:

- általános ki-, ill. bemeneti regiszterek;
- szokásos számítógép-perifériavezérlő modulok, digitális adatkezeléssel;
- kijelző, rajzoló perifériák;
- műszerezésvezérlő-, csatoló modulok;
- egyéb digitális eszköz vezérlő modulok.

Az általános ki-, ill. bemeneti regiszterek fő célja az, hogy nem-CAMAC rendszerű külső eszközöket illesszenek az ADATÚT-hoz. Ilyen eszközök például a konverterek, a jelfogók, a számlálók, esetleg a lyukszalagolvasók stb. Azért, hogy ne csak adatokat tudjanak beolvasni, hanem például a külső eszközt irányító vezérlőregisztert is be lehessen írni, kétirányú adatforgalomra készülnek. Rendelkeznek továbbá a szinkronizáláshoz, adatforgalom időbeni vezérléséhez szükséges ki-, ill. bemenő vezérlőjelekkel. Az általános ki-, ill. bemeneti regiszterek hátránya, hogy viszonylag sok felesleges elemet tartalmaznak. A gyakrabban használt *hagyományos számítógép-perifériák illesztésére* ezért speciális funkcionális modulokat fejlesztettek ki. Ilyen illesztőmodulok készültek írógépek, kalkulátorok, sornyomatók, lyukszalagolvasók és -lyukasztók, mágnesszalagos egységek stb. CAMAC rendszerhez való csatlakozására. Választékuk állandóan bővül az újabb perifériák megjelenésével. A számítógép-perifériák illesztésére szolgáló modulok jelentősége erősen nőtt az utóbbi időben, mivel sok helyen a hagyományos perifériakészlet ezeken keresztül kapcsolódik a kis-számítógéphez, ill. intelligens vezérlőprocesszorokhoz.

Az általános számítógép-perifériák egy nagy csoportja a különböző *digitális és analóg kijelzők, a megjelenítő oszcilloszkópok és kirajzoló*k. A digitális kijelzők olyan kimeneti regiszterek, amelyek tartalma az előlapon levő lámpákon megjelenik. Vannak bináris és decimális kijelzők. Az oszcilloszkópos megjelenítők igen széles

választékúak. A legegyszerűbbek kereskedelmi forgalomban kapható tárolócsöves oszcilloszkópot (pl. Tektronix 601, 611 stb.) vezérelnek.

Igen gyakori, hogy a vezérlő két regisztert és a hozzájuk kapcsolódó digitális—analóg konvertert tartalmazza és egyszerre egy pontot képes megjeleníteni, ill. egy pont koordinátáit tárolni. CAMAC utasításokkal be kell tölteni az X, ill. Y koordinátákat tároló regisztert és indítóimpulzussal kell vezérelni a pont felvillantását. E vezérlőmodul kimenete bármilyen oszcilloszkópot képes vezérelni. Az egységes mechanikai kivétel miatt azonban gyakran CAMAC keretbe helyezhető oszcilloszkóppal együtt használják. Ez az egyszerű megjelenítő azonban az ADATÚT-at nagyon hosszú időre leköti, mivel egy-egy pont felvillantását a villogásmentes ábrázolás miatt másodpercenként hússzor kell megismételni. Kis könnyebbséget jelent az, ha az X tengelyen egymás után következő értékeket kell megjeleníteni, és minden X értékhez egy Y érték tartozik. Ez az eset igen gyakori spektrumfelvételeknél. Ilyenkor elég egy-egy ponthoz csak az Y értéket átírni a megjelenítő regiszterébe, az X regiszter tartalmát pedig minden pont felvillantása után automatikusan növelni lehet eggyel. Sokszor ez sem elég gyors, ezért programozott vezérlő helyett autonóm vezérlőt is használnak a megjelenítőkhöz.

Ezek az egyszerű megjelenítőkön kívül moduláris, bonyolult megjelenítőket is alkalmaznak CAMAC rendszerekben. Ezeket részletesebben a 4.3. szakaszban, a modulcsaládok tárgyalásakor ismertetjük.

Az oszcilloszkópot vezérlő modulokhoz nagyon hasonló az X—Y írókat, analóg kirajzolókat vezérlő modulok. Szintén két regisztert és két digitális—analóg konvertert tartalmaznak, de a vezérlés sebessége jóval lassabb, mint az oszcilloszkópos megjelenítőknél.

A 4.13. ábrán egy elég bonyolult X—Y rajzoló vezérlőmoduljának vázlatát és utasításrendszerét mutatjuk be. A modul 10—10 bites X és Y regisztert tartalmaz, ezeket az ADATÚT-ról lehet beírni. Háromféle — kapcsolóval állítható — üzemmódja van:

1. az X és Y regiszterek tartalmát külön-külön fel kell tölteni programmal, és a két adatnak megfelelő analóg jelek kijutnak a rajzolóra;

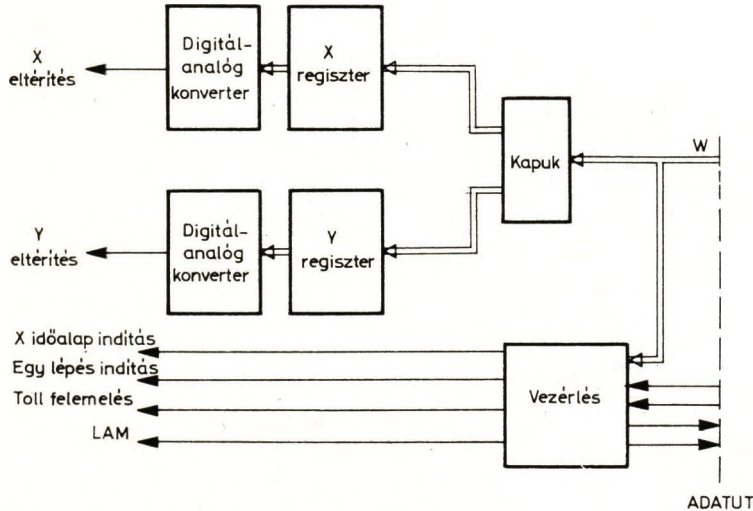
2. az Y regisztert be kell írni az ADATÚT-ról, az X regiszter tartalma 1-gyel nő, mindkét regiszter tartalmának megfelelő analóg jel a rajzolóba jut;

3. az X regiszter tartalmát írja csak be a program, és csak az ennek megfelelő analóg jel jut ki a rajzolóra, az X regiszter tartalmának megfelelő mozgást a rajzolómechanizmus automatikusan végzi. A rajzolótol mozgatása is programmal vezérelhető. A vezérlőjelek időzítése figyelembe veszi a tollmozgatás viszonylag lassú sebességét. Az X regiszter túlcsoordulása LAM jelet kelt.

A *műszereket csatoló modulok* igen sok félék, egyrészt különböző fizikai mérőberendezések, másrészt beavatkozó szervek csatolására szolgálnak. Igen sok — kereskedelmi forgalomban levő — amplitúdó analízátor illesztésére készültek modulok. A műszereket illesztő modulok közé lehet sorolni a viszonylag ritkábban használt fizikai mérőberendezések speciális moduljait is, mint például proporcionális kamra, szikrakamra, plumbikon, stb. illesztőmodulok. Ugyancsak ebbe a csoportba sorolhatók az átalakítókhoz (pl. különleges nyomás) közvetlenül kapcsolódó funkcionális modulok. A műszereket vezérlő modulok közül leggyakoribbak a különböző léptetőmotor-meghajtó modulok. Ezek választéka igen széles. A legegyszerűbbek lényegében egy vagy két preszet számlálót tartalmaznak, amelyeket belső vagy külső órajelgenerátor számlálót, és a számláló tartalmának megfelelő impulzusszám (vagy annak komplemente) jut ki a léptetőmotorra. A bonyolultabb vezérlők esetében az impulzusekvencia, azaz a léptetés sebessége automatikusan változik attól függően, hogy hány lépés van még hátra. Vannak olyan vezérlőmodulok is, amelyeknél az abszolút elfordulást kell programmal megadni, és a modul ebből és a visszakapott pozíciókódból ön maga állítja elő a szükséges lépésszámot. A nagyobb teljesítményű léptető-

motorokat meghajtó végfokokat külön — nem-CAMAC szabványú mechanikában — szokták elhelyezni. A léptetőmotorokat meghajtó logikák általában alkalmasak a digitális, inkrementális rajzolók vezérlésére is. Ebbe a családba tartoznak még a tápegységvezérlő modulok is, egyenáramú motor meghajtó modulok, stb.

Az egyéb digitális modulok csoportjának legjelentősebb tagjai különböző adatátviteli vonalakhoz, ill. jelforgalmi előírásokhoz illesztik a CAMAC ADATÚT-at. E modulok



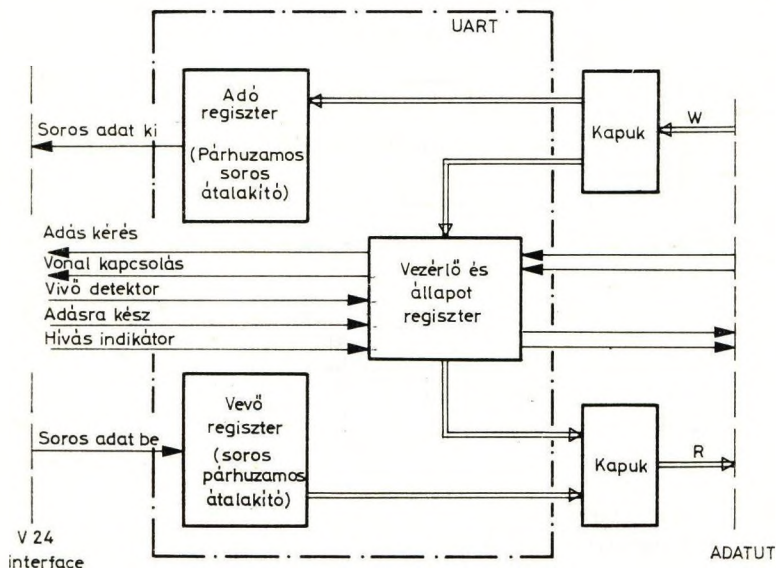
- Utasítások:
- | | |
|-------------|--|
| F(9) A(0) | X és Y regisztertörlés |
| F(16) A(0) | Y regiszter beírás |
| F(16) A(1) | X regiszter beírás |
| F(17) A(0) | Y regiszter beírás és toll leengedés |
| F(17) A(11) | vezérlőregiszter beírás (folyamatos-pontozott vonal) |
| F(24) A(11) | Toll leengedés felfüggesztés |
| F(25) A(1) | X regiszter +1 |
| F(25) A(2) | előkészület, toll felemelés |
| F(25) A(11) | toll leengedés |
| F(8) A(0) | LAM vizsgálat |
| F(10) A(0) | LAM állapot törlése |
| F(24) A(0) | LAM felfüggesztés |
| F(26) A(0) | LAM engedélyezés |
| F(27) A(0) | LAM forrás vizsgálat |
| F(27) A(11) | foglaltság vizsgálat |

4.13. ábra
Rajzolóvezérlő modul vázlata és utasításai

jelentősége is erősen nő, amit az is mutat, hogy újabban egyre több ilyen modult alkalmaznak. A legtöbb ilyen újabb tervezésű modul a V24 CCITT szabványnak felel meg, készülnek azonban szinkron átvitelű és egyedi megoldású adatátviteli modulok is.

V24 CCITT szabványnak megfelelő soros adatátviteli modul blokkvázlatát és utasításait a 4.14. ábra mutatja. Az adatátvivő csatornával a forgalmat az UART (Universal Asynchronous Receiver-Transmitter) elnevezésű MOS—LSI integrált áramkör bonyolítja le. Ez tartalmazza az adó- és vevő-regisztereket, elvégzi a soros—párhuzamos átalakításokat és tartalmazza a vezérlő—állapotregisztert

is. Adáskor programozott utasítással be lehet tölteni az adó-pufferregisztert, majd megindul a regiszter tartalmának kiléptetése a soros vonalra. Amikor a regiszter új adatot képes fogadni, LAM jelet ad a számítógépnek és állapotát az állapotregiszterben is jelzi. A vevőoldali működés azzal kezdődik, hogy az UART soros bemenete start-polaritást érzékel. Az érkező soros karakter bekerül (léptetődik) a vevőregiszterbe, majd amikor ez befejeződött, LAM jelet generál és az állapotregiszter megfelelő jelzőbitjét is beállítja. Ezután a program be tudja olvasni a vett karaktert. Megjegyezzük, hogy az UART-ban két adó- és két vevőregiszter van, amelyek felváltva működnek, így amíg az egyikből ki- vagy beléptetés történik, a másik már tölthető vagy üríthető a gépből. A vonal- vagy a külső



Utasítások: F(0) A(1)	vevő-regiszter olvasás, LAM állapot törlése
F(16) A(1)	adó-regiszter beírás, adás indítás, adás végén LAM
F(1) A(11)	állapotregiszter olvasás
F(17) A(11)	vezérlőregiszter beírás
F(8) A(0)	LAM vizsgálat
F(10) A(0)	LAM állapot törlése
F(24) A(0)	LAM felfüggesztés
F(26) A(0)	LAM engedélyezés

4.14. ábra

Soros adatátviteli csatoló modul blokkvázlata és utasításai

egység felől érkező vezérlőjelek (*Vívődetektor*, *Adásra kész*, *Hívás indikátor*) megváltozása szintén LAM-ot generál. E soros adatátviteli csatolómodul többféle adatátvivő közegezhöz csatlakozhat: nem nagy távolságban levő, V24-es interface-szel rendelkező Teletype-ok, alfanumerikus megjelenítő egységek közvetlenül a modulra csatlakozhatnak, másrészt postai szabványoknak megfelelő modemeken keresztül távbeszélővonalas átvívő hálózathoz is csatlakozhat. E különböző csatolási módoknak megfelelően a kimeneti adó-vevő áramkörök a modulon belül levő kisebb kártya cseréjével változtathatók.

4.2.4.

Digitális kezelő- és feldolgozó modulok

A digitális kezelő- és feldolgozó modulok:

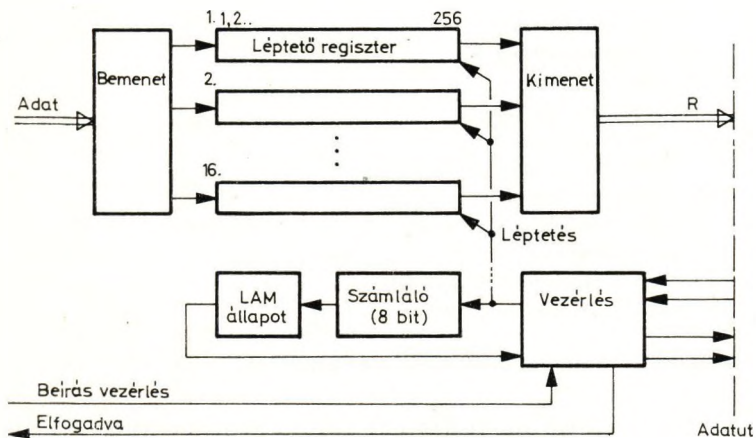
- a kapuáramkör rendszerek;
- a digitális szintkonverterek;
- a kódkonverterek;
- a tárolók és
- a logikai és aritmetikai feldolgozó modulok.

A kapuáramkör rendszereket megvalósító modulok funkciója az, hogy a kívülről érkező és az előlapcsatlakozókon újra kimenő impulzussorozatot programozott utasítással kapuzni lehet. Egy-egy modulba általában 6...8 ilyen kapuáramkört építenek be, a kapuzó feltételt pedig egy maszkregiszter adja.

A kódkonverterek leggyakrabban bináris és valamilyen más (pl. BCD) kód között végeznek konverziót.

Igen fontosak és egyre nagyobb jelentőségűek a tárolómodulok. Két nagyobb felhasználási területük van: egyrészt lökészerűen érkező mérési eredmények puffertárolására használhatók, másrészt pedig általános adat- vagy programtárolás céljára. A puffertárolóként használt tárolómodulokban a szavakat nem lehet egyedileg címezni, hanem csak szekvenciálisan, egymás után következően lehet elérni.

A 4.15. ábrán 256-szavas 16-bites puffertároló látható [14]. A tároló 16 db sztatikus MOS léptető-regiszterből áll. A kívülről érkező adatszavak egymás után következő helyekre kerülnek. A betöltött szavak, azaz a léptetőimpulzusok számát egy 8-bites számláló számolja, *túlsorduláskor* LAM jel keletkezik. Ezután programozott utasításokkal kiüríthető a tároló; minden szó kiolvasása után a cím automatikusan tovább lép.

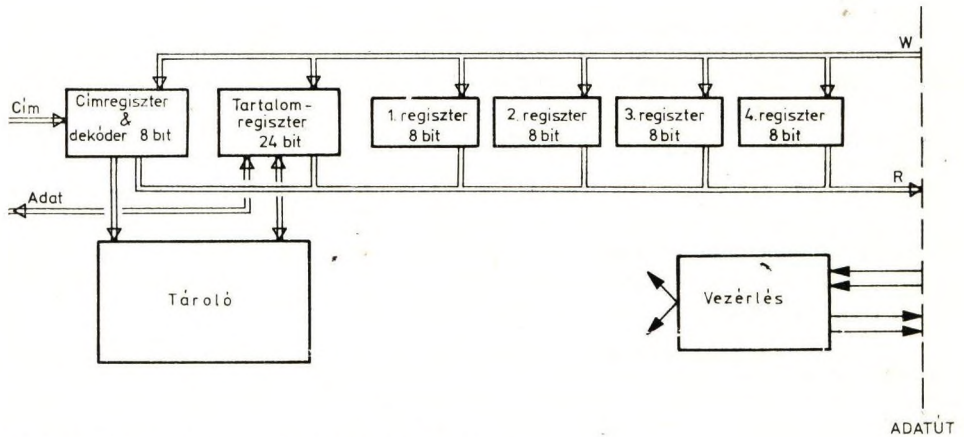


Utasítások: F(0) A(0)	tárolószó kiolvasás és címnövelés
F(8) A(0)	LAM vizsgálat
F(24) A(0)	LAM felfüggesztés
F(26) A(0)	LAM engedélyezés

4.15. ábra

256-szavas 16-bites puffertároló modul blokkvázlata és utasításai

Az általános adat- és programtároló hasonló felépítésű és szervezésű, mint az általános célú számítógépek operatív tárolója: tartalma véletlenszerűen elérhető és írható—olvasható. A tárolóhoz sok esetben mind az ADATÚT felől, mind előlapi csatlakozó felől hozzá lehet férni. Nehézséget jelent a címzés, mivel csak 4 alcímbit van. A tároló szavainak címzésére egy elterjedt mód az, hogy a címet külön utasítással előbb be kell írni a modulban levő címregiszterbe, és egy következő utasítás viszi át a tartalmat; ez azonban a tényleges elérési időt megkétszerezi. Az általános tárolót is a leggyakrabban integrált áramkörökből építik fel, bár vannak ferritgyűrűkből felépített tárolómodulok is.



Utasítások: F(0) A(0)	1. regiszter kiolvasás
F(16) A(0)	1. regiszter beírás
F(0) A(1)	2. regiszter kiolvasás
F(16) A(1)	2. regiszter beírás
F(0) A(2)	3. regiszter kiolvasás
F(16) A(2)	3. regiszter beírás
F(0) A(3)	4. regiszter kiolvasás
F(16) A(3)	4. regiszter beírás
F(0) A(4)	tárolószó kiolvasás
F(1) A(4)	címregiszter kiolvasás
F(16) A(4)	tárolószó beírás
F(17) A(4)	címregiszter beírás
F(24) A(4)	külső tárolóelérés felfüggesztés
F(26) A(4)	külső tárolóelérés engedélyezés

4.16. ábra

256-szavas 24-bites általános tároló modul blokkvázlata és utasításai

Egy 256-szavas, 24-bites véletlen elérésű tárolót mutat be a 4.16. ábra [23]. A tároló integrált áramkörökből épül fel, gondoskodtak azonban arról, hogy hálózatkimaradás esetén egy kisteljesítményű pótáramforrásról kapjon feszültséget, és a tartalom így megőrződik. A 256 — 24-bites — tárolószót mind az előlapon keresztül, mind ADATÚT utasítással el lehet érni. A címzés úgy történik, hogy külön utasítással egy címregisztert be kell tölteni, és ezután lehet a tárolóhoz hozzáférni. Magán a tárolón kívül 4 közvetlenül, alcímmel címezhető 8-bites regiszter is van a modulban.

A tárolómodulok között nemcsak írható—olvasható (RAM) tárolók vannak, hanem különböző, csak-kiolvasható típusúak (ROM, PROM stb.) is. Ezekben paramétereiket,

programokat, stb. tárolnak. Példaképpen említjük, hogy érdekes szerep jut az önálló vezérlővel felépített rendszerekben a csak-kiolvasható tárolónak: ui. a kezdeti betöltő (bootstrap) program ezekben tárolódik.

A logikai és aritmetikai feldolgozó modulok jelentősége az önálló vezérlők megjelenésével megnőtt. A CAMAC rendszerben a lehetőségek adottak, akár egy többprocesszoros vezérlő számítógép összeállítására is. Ilyen irányú törekvések vannak, azonban ez a szervezés még nem terjedt el szélesebb körben, bár pl. egy lebegőpontos aritmetikai egységet csatoló modul már kereskedelmi forgalomban kapható.

Speciális célokra is készülnek digitális adatkezelő modulok. Erre példa a következő. Digitális adatátvitelnél gyakran alkalmaznak digitális csatolt szűrőt, amely az átvívóvonallal érkező jelsorozatot keresztkorrelációba hozza egy adott jelsorozattal [10]. Ez a művelet szorzásokból és átlagolásokból áll. Bár a feladat programmal kényelmesen megoldható, gépidő- és tárolókapacitás-megtakarítás miatt egy speciális CAMAC modult készítettek, amely a műveleteket hardware úton gyorsan elvégzi, és az eredményt az ADATÚT felé továbbítja.

4.2.5.

Analóg modulok

Az analóg modulokat három nagy csoportba oszthatjuk:

- bemeneti analóg modulok;
- kimeneti analóg modulok;
- analóg jel-kezelő modulok.

Bemeneti analóg modulok

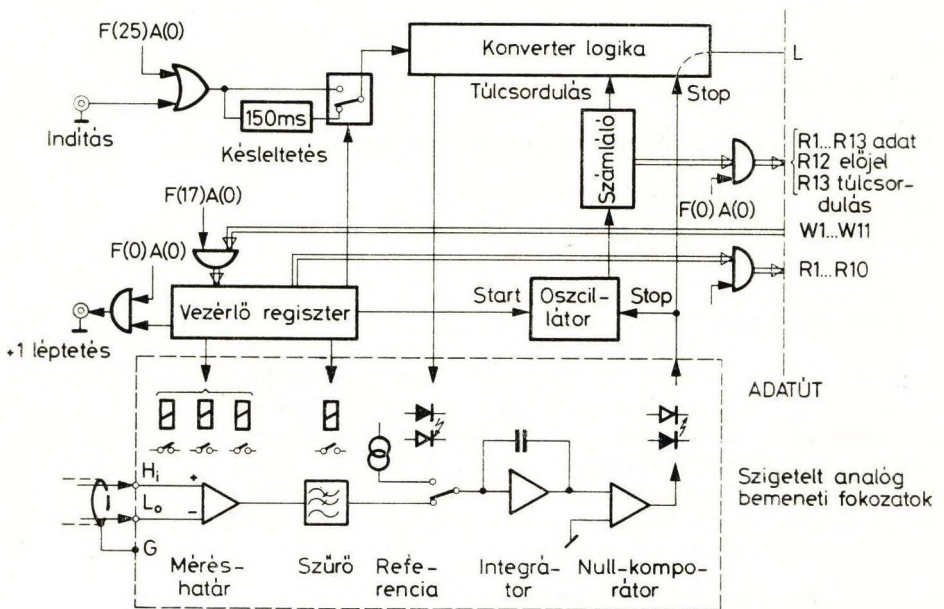
A bemeneti analóg modulok (30 különféle típus) feladata, hogy valamilyen analóg formában érkező információt a számítógépnek továbbítsanak a gép számára érthető, digitális formában. A modul tehát hozzárendel az analóg mennyiséghez egy számértéket, analóg—digitális konverziót végez. A bemenő analóg jel leggyakrabban: egyenfeszültség vagy feszültségimpulzus, ritkábban áram, töltés vagy időtartam. Ennek megfelelően a különféle bemeneti analóg modulok, azaz analóg—digitális konverterek a következő feladatra alkalmasak:

- egyenfeszültség (áram) mintavételes mérésére;
- impulzuscsúcsérték mérésére;
- időmérésre.

Az egyenfeszültség mintavételes mérésére szolgáló modulok családja a legnépesebb, az összes bemeneti modulnak kb. 70%-a esik ebbe a csoportba. Ezen belül a többség 10 vagy 12 bit felbontású, de vannak 7-, 8-bites és 14-, 15-bites A/D konverterek is. A konverziós elvet tekintve a kisebb bitszámú (8...12-bites) konverterek a sorozatos közelítés (szukcesszív approximáció) elvét alkalmazzák bináris súlyozással, a nagyobb bitszámú (10...15-bites) konverterek a „kétfűrész” (dual slope) elvet vagy amplitúdófrekvencia-konverzió elvét alkalmazzák, tehát lényegében „számláló” típusúak. A sorozatos közelítésű A/D konverterek gyorsak, konverziós idejük a bitszámtól függően 5...30 μ s. A számláló típusúak nagy része CAMAC modulba épített integráló típusú digitális voltmérő, amelyeknél a konverziós idő a hálózati zavarok kiszűrésére érdekében legalább 20 ms vagy ennek egész számú többszöröse.

A leggyorsabb A/D konverter egy egységnyi széles modulba épített, 7-bites konverter [22], amelynek a konverziós ideje 0,2 μ s. Természetesen a modulból ilyen sebességgel nem lehet az adatokat a CAMAC ADATÚT-on keresztül kiolvasni, ezért beépített 1024-szavas MOS félvezető puffertárat is tartalmaz. A legnagyobb felbontású modul 15-bites, integráló típusú A/D konverter, konverziós ideje: 200 ms [2].

A konverterek bemeneti feszültségtartománya az igényekhez igazodik, csak annyiban követi az EUR 5100 előírásait*, amennyiben általában ± 5 V-os mérési tartomány is van, de emellett megtalálható ± 10 V-os tartomány is. Integráló típusú konverterek esetén pedig a kis feszültségű méréshatár általában 20...100 mV közötti. Gyakori, hogy egyetlen modulban több egymástól független A/D átalakítót valósítanak meg. Tipikus a két 10-bites konvertert tartalmazó modul, de van 2×12 -bites és 8×8 -bites is. Ezen kívül készítenek átalakítót analóg multiplexerrel egybeépítve. A multiplexert is tartalmazó átalakítóknál a bemenetek száma szimmetrikus jelnél 8, aszimmetrikus jelnél 16; ez azonban megfelelő multiplexer modullal még tovább bővíthető.



- Utasítások: F(0) A(0) adatolvasás; +1 generálás, ha engedélyezve van
 F(1) A(0) vezérlőregiszter olvasás
 F(17) A(0) vezérlőregiszter beírás
 F(25) A(0) konverzió indítás
 F(8) A(0) LAM vizsgálat
 F(10) A(0) LAM állapot törlése
 F(27) A(0) foglaltságvizsgálat

4.17. ábra
Integráló típusú analóg—digitális átalakító blokkvázlata és utasításkészlete [11]

* A ESONE EUR 5100e/72 kiadványban illetetlen analóg jelekre egypolarítású jelnél 0...+5 V-os, kétpolarítású jelnél ± 5 V-os működési tartományt írnak elő. A jelet generáló modul kimenő ellenállása 50 Ω , a jelet fogadó modul bemeneti ellenállása 5 k Ω -nál nagyobb kell, hogy legyen. Koaxiális kábelek csatlakoztatására LEMO 00250 típuscsatlakozót kell alkalmazni.

Példaként egy 12-bites — 11 bit+előjel — integráló típusú analóg—digitális konverter modul mutatunk be (4.17. ábra). A konverter bemenete galvanikusan leválasztott, szigetelt, amit úgy valószínűleg meg, hogy a teljes analóg lánc lebeg. Az analóg lánc egyes fokozatait potenciálisan független elemekkel vezérlik. Így a méréshatár-átkapcsolás és a szűrő ki/bekapcsolás jelfogóval, a gyors vezérlést igénylő referencijel átkapcsolás és a lánc végéről az időtranszformált jel továbbítása opto-csatolt eszközökkel valósul meg. A konverter főbb állapotait a vezérlőregiszterrel lehet beállítani. E regiszter egyes bitjeit az ADATÚT írásvonalain keresztül lehet beírni, az alábbiak szerint:

W1...W3	méréshatár 10 V ÷ 20 mV között 8 fokozatban (a méréshatár-váltás az első erősítő-fokozat erősítési tényezőjének jelfogás-átkapcsolásával történik. Az átkapcsolás után a konverzióindítás 3 ms-mal van késleltetve).
W4, W5	szűrő ki/bekapcsolás
W6, W7	LAM engedélyezés/felfüggesztés.
W8, W9	a konverterrel esetleg összekapcsolt multiplexer léptetésére szolgáló „+1” kimenő jel engedélyezés/felfüggesztés.
W10, W11	konverziókésleltetés (150 ms) bekapcsolás/kiiktatás.

A modul konverziós ideje 40 ms (20 ms integrálás, 20 ms kisütés), mechanikai felépítését tekintve 2 egységnyi széles, 1 digitális kártyát tartalmaz, ez csatlakozik az ADATÚT-hoz. A lebegő analóg rész egy külön kártyán van felépítve és árnyékolódobozba szerelve. Ez a doboz a digitális kártya első üresen hagyott része fölött helyezkedik el.

Impulzuscsúcsérték mérésére szolgáló modulok. A nukleáris mérések egyik legtipikusabb feladata impulzuscsúcsértékek meghatározása, ami pl. lehetőséget biztosít részecskék energia szerinti osztályozására. Ez ma is a sokcsatornás analízátorok egyik alapüzemmódja. Az egyszerű sokcsatornás analízátorok mellett kialakultak a többparaméteres, asszociatív, tároltprogramú analízátorok, jelenleg a CAMAC számítógépes analízátorok, de a bemenet továbbra is gyakran valamilyen impulzuscsúcskonverter. (Ritkábban időkonverter.) Annak ellenére, hogy a CAMAC rendszert eredetileg nukleáris központok munkatársai dolgozták ki és ott került alkalmazásra is, rendkívül kevés, mindössze 3 különböző típusú CAMAC modulba épített, impulzuscsúcsmérésre alkalmas A/D konverter terjedt el szélesebb körben. A jó differenciális linearitás érdekében mindegyik a Wilkinson-típusú amplitúdó—idő konverziós elvet alkalmazza. A felbontások: 1 k, 4 k, ill. 16 k. A konverzió sebességére jellemző órfrekvenciák: 80, 100, ill. 200 MHz. A konverziós idő, azaz a holtidő csökkentésére módosított Wilkinson-elvet, a kétmeredekségű — dual ramp — kisütést alkalmazó CAMAC konverterek is készültek. NIM modulban valósították meg, de adapterrel CAMAC rendszerben is felhasználható, a Gatti által kidolgozott csúszkálós-kálás, sorozatos közelítési elvet alkalmazó impulzuscsúcsmérő konverter, amelynek műszaki paraméterei igen jók: 8 k felbontás, 4 μ s holtidő, $\pm 0,2\%$ -os diff. linearitás [24].

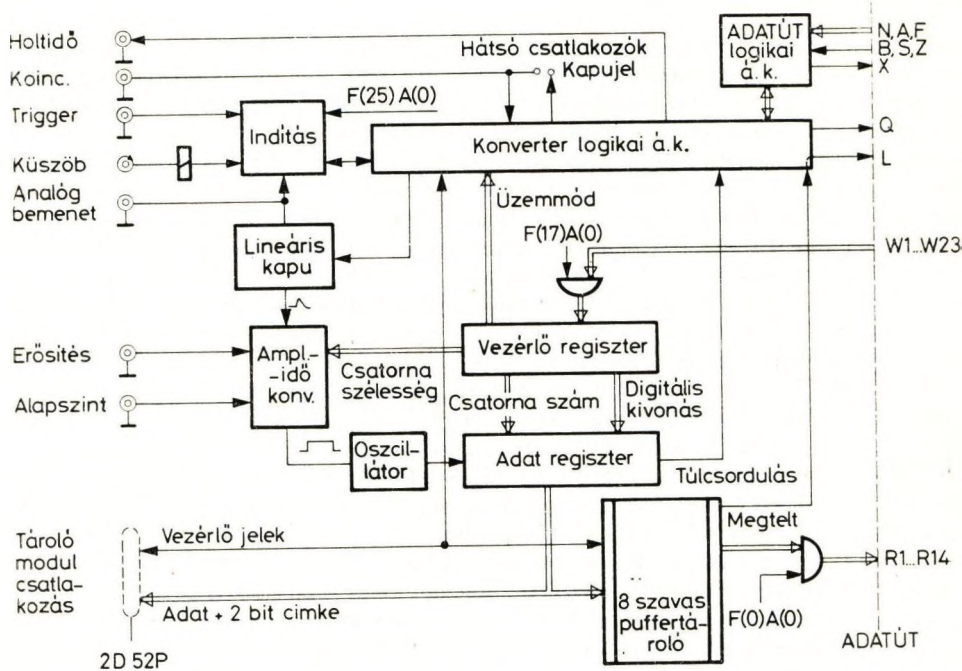
A CAMAC impulzuscsúcs-konverterek felépítése sokban hasonlít a sokcsatornás analízátorok konvertereihez, az ADATÚT logikán kívül azzal a bővítéssel, hogy mindegyik tartalmaz csatlakozást az ADATÚT megkerülésével valamilyen puffertárolóhoz. Így nem kell minden egyes adattal közvetlenül a számítógéphez fordulni, hanem csak akkor, ha magában a konverterben megépített, vagy külön CAMAC modulként kialakított puffertároló megtelt, amikor az adatokat blokkba szervezve („derandomized” módon) lehet átvinni a gépbe. Ez pedig az átlagos holtidő jelentős csökkenését eredményezi.

A 4.18. ábrán egy 12-bites, impulzuscsúcsérték mérésre alkalmas A/D átalakító modul mutatunk be. A modul bemenete aszimmetrikus (Single-ended), átköthetően 0...+5 V-os feszültségimpulzusok, vagy üzemmódváltással ± 5 V-os bipoláris jel mintavételes mérésére alkalmas. Mintavételes üzemben külső jel (TRIGGER) vagy ADATÚT utasítás kezdeményezheti a konverziót. Impulzusmódban a konverzió indítása a vezérlőregiszter által beállítható különféle koincidenciafeltételektől, az előlapon potenciométerrel állítható küszöbszinttől, továbbá mindkét üzemmódban a konverterlogikából jövő engedélyező—tiltó jeltől függ. A modul a foglaltságát holtidőjel kiadásával jelzi és kapuület generál minden, az indítási feltételeknek megfelelő jel beérkezésekor. (E kapuület és egy másik ugyanilyen modul koincidenciabemenetének keresztbe kapcsolásával kétparaméteres mérést végezhetünk.)

Az átalakítás Wilkinson-típusú, az átalakítási karakterisztika stabilizálására a modul két bemenettel látták el. Az erősítés bemeneten beadott analóg jellel az átalakítási karakterisztika meredekségét, az alapszint bemeneten beadott analóg jellel pedig a konverziós karakterisztika nullátmeneteit lehet kismértékben állítani. A modul üzemmódját, csatornaparamétereit a vezérlőregiszterbe beírt szóval lehet meghatározni. A vezérlőregiszter egyes bitjeinek jelentése a következő:

- W1...W4 csatornaszám-kiválasztás 256-tól 4096-ig bináris lépésekben.
- W5...W14 digitális kivonás állítása 4 csatornánként 0-tól 4096-ig.
- W15 üzemmód-beállítás, impulzuscsúcs vagy mintavételes mérés.
- W16 kiolvasás módja: ADATÚT-on vagy előlapon.
- W17...W21 konverziós tényező beállítása 1,25 mV-tól 20 mV-ig bináris lépésekben.
- W22...W23 koincidencia-üzemmódok kiválasztása.

A modul 80 MHz-es órát tartalmaz, konverziós ideje amplitúdótól, csatornaszámtól függően 5...50 μ s. A mérési eredmények az adatregiszterből egy 8-szavas puffertárolóba íródnak be és a modul csak akkor jelez a számítógépnek LAM-jellel, ha a puffertároló megtelt. Az olvasás a puffertárolóból történik; ha kiürült a tároló, az olvasásutasításra Q=0 választ kapunk. Egy másik üzemmódban (W16)



Utasítások: F(0) A(0)

a 8-szavas adatpuffer kiolvasása (minden olvasás, ha van adat a puffertárolóban, Q=1 választ ad, ha elfogyott, Q=0 válasz keletkezik)

F(17) A(0)

vezérlőregiszter beírás

F(25) A(0)

konverzió indítás

F(8) A(15)

LAM vizsgálat

F(10) A(15)

LAM állapot törlése

F(24) A(15)

LAM felfüggesztés

F(26) A(15)

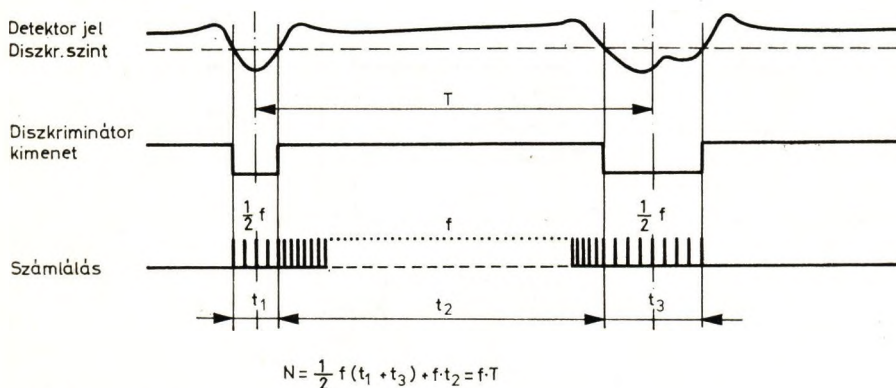
LAM engedélyezés

4.18. ábra

Impulzuscsúcsérték mérésére alkalmas analóg-digitális konverter modul blokkvázlata és utasításkészlete [1]

az adatok az adatregiszterből az előlapon levő 52-pontos csatlakozón keresztül egy külön megépített 2048-szavas puffertároló modulba kerülnek. A konverter egyetlen kártyán, egy egységnyi széles CAMAC modulban készült.

Az időmérésre szolgáló modulok (ebből kb. 10 típus van) jellegzetesen két csoportra oszlanak. Az egyik csoportba tartozó modulok magnetostrikciós kiolvasóvonalakkal ellátott szikrakamrákhoz készültek, az események koordinátáinak meghatározására. A modul ebben az esetben 4 db 16-bites számlálót tartalmaz, amelyek közös jelre kezdenek egy nagyfrekvenciás óra (max. 50...70 MHz) impulzusainak számlálásába; az indítójel utáni első esemény, szikraimpulzus leállítja az első 16-bites számlálót, a második leállítja a második számlálót, stb. A számlálók indításakor, leállításakor közép-vonal-azonosító eljárást alkalmaznak, melynek lényege az, hogy az indítójel és leállítójel tartama alatt a számlálás $1/2$ -es frekvenciával történik, így eredőben a leszámított órajelek száma az indító- és leállítójel közép-vonala között eltelt idővel arányos (4.19. ábra).



4.19. ábra
Idő—digitális átalakítók közép-vonal-azonosító eljárása

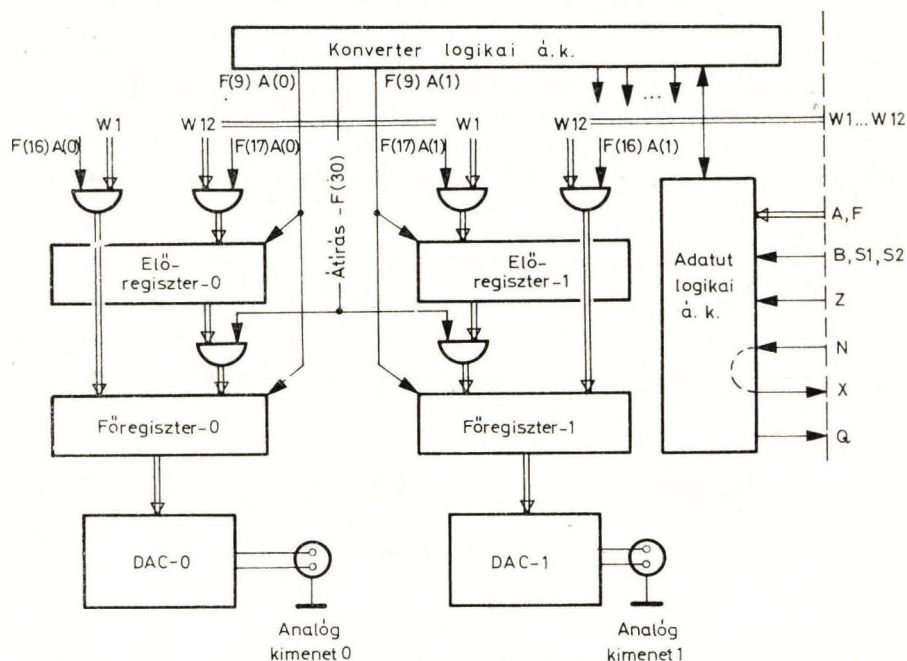
A második csoportba tartozó modulok rövid idők mérésére alkalmasak, ahol a teljes skála 100...500 ns. A mérési pontosság 8...10 bit, így a csatornaszélességre $0,2 \div 1$ ns adódik. A konverterek valamilyen ismert időtranszformációs (idő—amplitúdó—idő, noniusz...) elvet alkalmaznak. Egy egységnyi széles modulban 4...8 átalakító van.

Az elmondottakon kívül időmérésre általános célú digitális modulokat is alkalmaznak. Különösen biológiai, orvosi mérésekben elterjedt az az időmérési eljárás, hogy a mérendő időtartam alatt egy megfelelően kiválasztott (programozott) frekvenciájú óramodul kimenőjelét számláltatják egy soros számlálóval [3].

Kimeneti analóg modulok

A kimeneti analóg modulok (kb. 75 alaptípus) csoportjába olyan modulok tartoznak, amelyek a számítógépből érkező számértéknek megfelelő analóg jelet állítanak elő és ezt külső eszközöknek továbbítják, tehát digitális—analóg konverziót végeznek. A kimenő jel tipikusan aszimmetrikus, feszültség a ± 5 , ± 10 vagy $0...+5$ V és $0...+10$ V tartományban, 5...20 mA-es terhelhetőséggel, de előfordul áramgenerátoros kimenet ($0...2$ mA, ill. ± 20 mA) és differenciális kimenet is. A modulok felbontása 8-, 10-, ill.

12-bites, a leggyakoribb a 10-bites. Egy egységnyi széles kimeneti modulban általában több független digitális—analóg konvertert, DAC-t helyeznek el, a kisebb pontosságú 8-bites DAC-ból 8-at, a 10-bitesből 2-t, 4-et vagy 8-at, a 12-bitesből max. 2-t. A digitális—analóg átalakítóknak szinte kizárólag a bináris áram- vagy feszültség-súlyozásos elvet alkalmazzák, a konverzió sebessége 2...10 μ s, amit legtöbb esetben a végfokerősítő maximális feszültségemelkedési üteme (slew-rate-je) szab meg.



- Utasítások:
- | | |
|------------|--|
| F(9) A(0) | 0-ás elő- és főregiszter törlés |
| F(9) A(1) | 1-es elő- és főregiszter törlés |
| F(16) A(0) | 0-ás főregiszter beírás |
| F(16) A(1) | 1-es főregiszter beírás |
| F(17) A(0) | 0-ás előregiszter beírás |
| F(17) A(1) | 1-es előregiszter beírás |
| F(30) | átírás mindkét előregiszterből a főregiszterbe |

4.20. ábra
2x12 bites D/A átalakító bloksémája és utasításkészlete [8]

A 4.20. ábrán egy 2x12 bites kimeneti analóg modul mutatunk be. A modulba két egymástól független digitális—analóg átalakítót építettek. A modul érdekessége, hogy a D/A átalakítókat kettős pufferral látták el. A kimenő analóg jel aktuális értékét a főregiszterben levő adat határozza meg, miközben az előregiszterben levő adatot szabadon változtathatjuk, így lehetőség van arra, hogy a következő adatot több modulban előre előkészítsük és egy későbbi utasítással — F(30) — egyszerre átírjuk a FŐREGISZTER-be és így az új értéket minden DAC-nál egyszerre állítsuk be. A modul kimenő feszültsége mindkét kimeneten ± 5 V vagy ± 10 V (átkötéssel választható), terhelhetősége 20 mA, rövidzárvédett, beállási ideje 2 μ s.

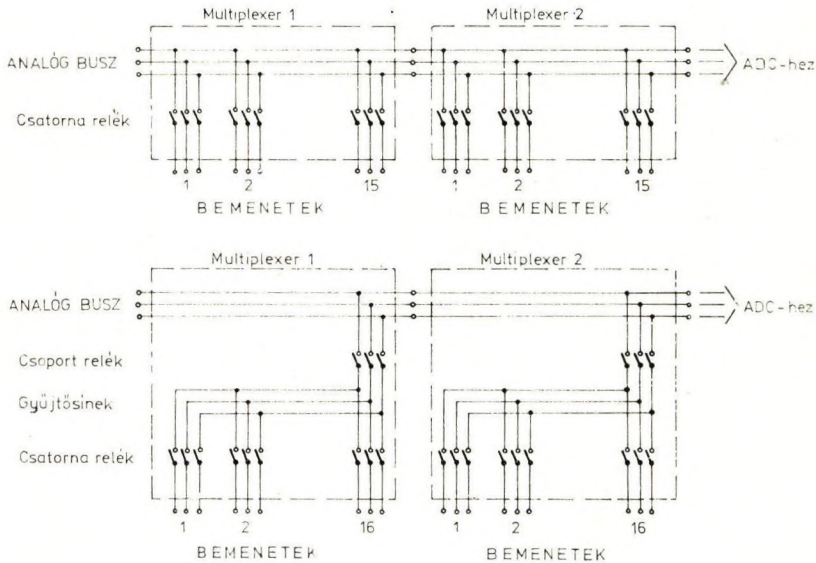
Analóg jel-kezelő modulok

Az analóg jel-kezelő modulok meghagyják a jelet analóg formában, megváltoztathatják viszont a jel útját, nagyságát, frekvenciamenetét, töltés—feszültség, áram—feszültség átalakítást végezhetnek stb., eközben esetleg ADATÚT vezérlést igényelnek. A modulok decimális osztályozása szerint az alábbi három csoportot különböztetjük meg:

- analóg jel-kezelő modulok I. csoportja (multiplexerek);
- analóg jel-kezelő modulok II. csoportja (erősítők, kapuáramkörök, stb.);
- ADATÚT vezérlés nélküli modulok.

Az analóg jel-kezelő modulok I. csoportjába az analóg multiplexerek tartoznak. A multiplexerek feladata, hogy több analóg jel közül egy tetszés szerintit, vagy az analóg jeleket sorjában, egyiket a másik után egy közös kimeneti pontra kapcsoljanak. A mérendő analóg jelek száma pl. elég nagy lehet, de nem lenne kifizetődő, meg nem is indokolt, hogy minden jel mérésére egy külön analóg bemenő modult, analóg—digitális átalakítót építsünk a rendszerbe. Több analóg jel mérését végezhetjük oly módon, hogy az analóg bemenő jeleket egy multiplexer bemeneteire kötjük és ennek kimenetét, amelyen tehát akármelyik kiválasztott analóg jel megjelenhet, az ADC bemenetére kötjük. A multiplexerek fontosságát mutatja, hogy az eddig elkészült jelkezelő modulok több, mint 70%-át a különféle multiplexerek teszik ki. A multiplexereket osztályozhatjuk csatornaszám (bemenetek stáma) és a csatornánként kapcsolt vezeték száma szerint. Leggyakoribb a 15-, 16-csatornás multiplexer, többször előfordul még a 32-csatornás és egy-egy esetben a 8-, ill. 64-csatornás multiplexer is. A csatornánként átkapcsolt vezeték számát a mérendő jel szabja meg. Az aszimmetrikus jelek átkapcsolása egyvezetékes multiplexert, a szimmetrikus jelek átkapcsolása kétvezetékes, a lebegő rendszerekhez a két jelvezeték (Hi, Lo) mellett a mérő árnyékolás (guard) átkapcsolására is szükség van egy vezetékre, így ebben a rendszerben 3-vezetékes multiplexert kell alkalmazni. Ez utóbbi típus különösen ipari rendszerek esetén elterjedt. Ugyanitt alkalmaznak még 4-vezetékes multiplexereket is, speciális jelforrások (pl. hidak) átkapcsolására. Az átkapcsolható jel nagyságát, a kapcsolás sebességét a lineáris kapuáramkörként használt kapcsolóelem szabja meg. Mai gyakorlatban FET-es vagy reed-relés kapcsolókat használnak. A FET-es kapcsolókat elsősorban az 1- és 2-vezetékes multiplexerekhez, míg a reed-relét a 3- és 4-vezetékes rendszerekhez alkalmazzák. Az átkapcsolás feszültséghibája 1-vezetékes multiplexerek esetén a közös földön levő zavarok miatt 1 mV körül van, a 2-vezetékes, differenciál FET-es multiplexerek esetén kb. 50...100 μ V, relés multiplexerek esetén, speciális relék alkalmazásával (kis kontakt potenciál, drift) elérheti a 10 μ V-ot. Az átkapcsolás sebessége FET-es multiplexer esetén kb. 10 μ s, relés multiplexer esetén 2...3 ms. A multiplexerek általában bővíthetők, ha tehát több bemenetre van szükség, mint amit egy modul át tud kapcsolni, több multiplexer modult alkalmazhatnak, amelyek megfelelően összekötve egy közös kimenetet adnak. A 4.21. ábrán bővített 3-vezetékes multiplexerek analógjel-útját mutatjuk be. Ha az ábra felső részén levő 15-bemenetű multiplexerek csatornareléi közül valamelyiket bekapcsoljuk, az adott bemeneten levő analóg jelet közvetlenül rákapcsoljuk a modulok analóg kimeneteinek multiplálásával keletkező ANALÓG BUSZ-ra, amely végül az ADC-hez csatlakozik. Az ábra alsó részén levő multiplexereknél a csatornarelé az analóg jelet csak egy közbülső gyűjtő sínre kapcsolja és onnan csak úgy jut ki az ANALÓG BUSZ-ra, ha a csoportrelét is bekapcsoljuk. A csoportrelé alkalmazása kettős leválasztást biztosít, így a kiiktatott bemenetek zavaró hatása lényegesen csökken. A csatornarelék szórt

kapacitása nem közvetlenül terheli az ANALÓG BUSZ-t, hanem csak a kikapcsolt csoportrelé szórt kapacitásán leosztva, vagyis a jel útjába még egy nyitott kontaktus kerül. A csoportrelé alkalmazása lehetőséget ad továbbá arra, hogy valamelyik csatornarelé meghibásodása esetén az adott multiplexer modult kiiktassuk, leválasszuk az ANALÓG BUSZ-ról és a rendszer többi részének működését biztosítani tudjuk. Az ábrán nem tüntettük fel a modulok digitális részét (ADATÚT logika, multiplexer



4.21. ábra

Bővített 3-vezetékes relés multiplexerek analóg jelűtja csoportrelé nélkül és csoportrelével

kijelölő logika, stb.), ami az ábrázolt analóg jel-utaktól galvanikusan független és a reed-relék meghúzó tekercseit vezérli. A gyakorlatban megvalósított relés multiplexerek között találunk olyan megoldást is, amelynél a tulajdonképpeni relés modul nem tartalmaz teljes ADATÚT logikát és a vezérlést részben egy ún. multiplexer vezérlőmodul látja el, amelyből keretenként egyet kell biztosítani [15]. Előnye, hogy sok bemenet esetén a rendszer kialakításához nagy számban igényelt modul típus ára jelentősen csökkenthető.

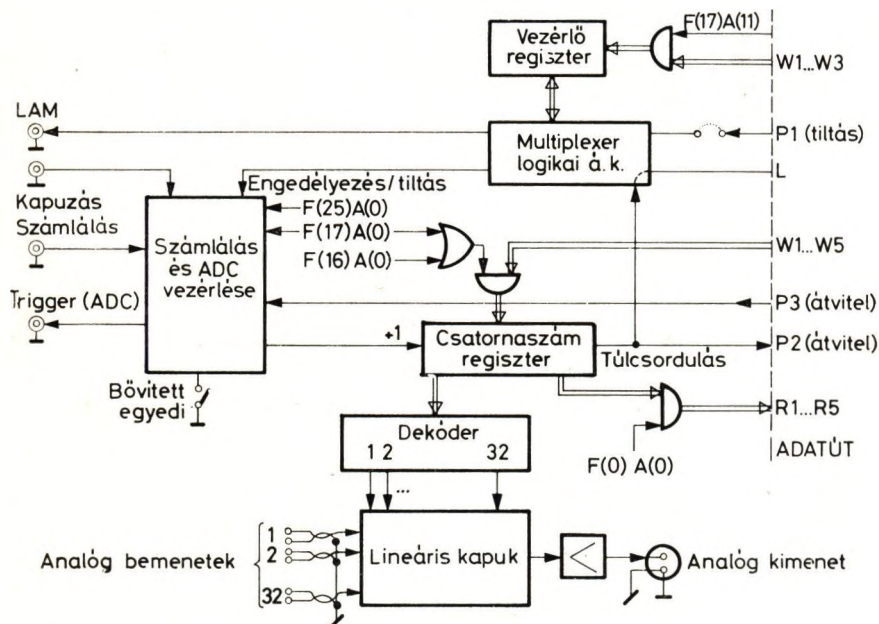
Multiplexerek működésére bemutatjuk a 4.22. ábrán látható, 32-bemenetű modult. Azt, hogy adott pillanatban melyik bemenetet kapcsoljuk a kimenethez, a csatornaszám regiszterben levő adat határozza meg a regiszterhez kapcsolódó dekóderen keresztül. A regiszter az ADATÚT felől párhuzamosan beírható, kiolvasható, külső jellel vagy ADATÚT utasítással számláltható. Így a bemenetek bármelyikét kiválaszthatjuk, vagy sorjában felkereshetjük a bemeneteket. Ez utóbbi esetben egy kezdőcímet is megadhatunk az F(16) A(0) utasítással. Ha a multiplexer a párhuzamos beírás: F(17) A(0), vagy továbbléptetés: F(25) A(0), ill. SZÁMLÁLÁS jel után a kívánt csatornát kiválasztotta és az analóg kimenet jel tranziensei lecsengtek, az analóg-digitális konverter indítására *Trigger* jelet ad ki. A csatornaszám regiszter számlálását, továbbléptetését külső kapujellel *kapuzás* vagy az ADATÚT P1 pontján beadott szinttel letilthatjuk. Ha pl. a P1 pontra az ADC-ben rákötjük a holtidőjelet, a P1 buszvonál a multiplexerben letiltja a multiplexer továbbléptetését mindaddig, amíg a konvertert be nem fejezte és belőle az adatot el nem vittük. A csatornaszám regiszter túlcsoordulása LAM jelet és bővített működéshez (P2, P3) átviteljelet generál. A modul üzemmódjait a vezérlő regiszter

határozza meg. Az egyes bitek jelentése:

- W1 törlés (C) engedélyezés/felfüggesztés.
- W2 tiltás (I) engedélyezés/felfüggesztés.
- W3 külső számlálás engedélyezés/felfüggesztés.

A modulban lineáris kapuáramkörként FET-es kapcsolók kerültek alkalmazásra. A csatornaváltás beállási ideje max. 10 μ s, feszültségtartománya ± 5 V, a multiplexer 1-vezetékes ugyan, de a vázfeldeleltől független közös mérő (clean) földelésvezetékét is fogadja, ill. kiadja. Mechanikai felépítését tekintve egységnyi széles.

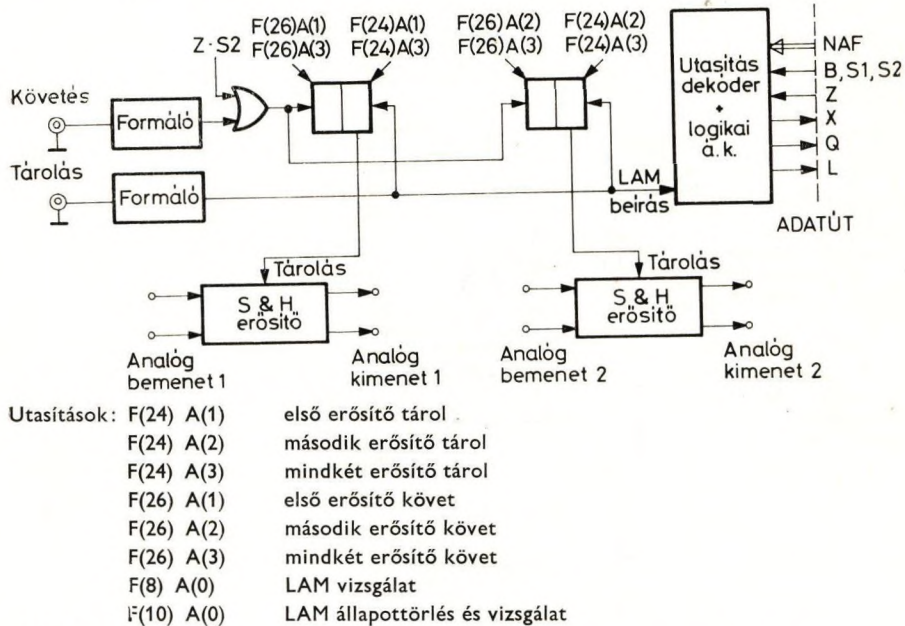
Az *analóg jel-kezelő modulok második csoportjába* sorolunk a multiplexerek kivételével lényegében minden további jelkezelő modult, amely ADATÚT vezérléssel rendelkezik. Ilyen modulok pl.: lineáris kapuáramkörök, programozható erősítők, osztók, diszkriminátorok stb. Annak ellenére, hogy sok féle rendeltetésű modul tartozik



- | | |
|-----------------------|---|
| Utasítások: F(0) A(0) | csatornaszám-regiszter kiolvasás |
| F(9) A(0) | csatornaszám-regiszter törlés |
| F(16) A(0) | csatornaszám-regiszter beírás és trigger impulzus generálás |
| F(17) A(0) | csatornaszám-regiszter beírás |
| F(17) A(11) | vezérlőregiszter beírás |
| F(24) A(11) | külső számlálás tiltás |
| F(25) A(0) | csatornaszám-regiszter tartalom+1 és trigger impulzus generálás |
| F(26) A(11) | külső számlálás engedélyezés |
| F(8) A(0) | LAM vizsgálat |
| F(10) A(0) | LAM törlés |
| F(24) A(0) | LAM felfüggesztés |
| F(26) A(0) | LAM engedélyezés |
| F(27) A(0) | LAM állapot (túlcsordulás) vizsgálat |

4.22. ábra
32-bemenetű multiplexer blokkvázlata és utasításkészlete [21]

ebbe a csoportba, eddig összesen alig 10-féle modul készült, amelyekből bemutatunk néhányat. Tervezték analóg jel leosztására szolgáló modult. A modul csillapítása 0 és 60 dB között, 1 dB-es lépésekben állítható. A beállítás számkerékkel vagy ADATÚT utasítással történik. Az osztóellenállások átkapcsolását jelfogók végzik, az átkapcsolási idő 20 ms. Készítettek programozható erősítőt, amelynek az erősítését 1...300-ig 6 fokozatban lehet állítani. A beállítás az előlapon levő fokozatkapcsolóval vagy ADATÚT utasítással történik, az átkapcsolást FET-es kapcsolók végzik. Egy új erősítésértékre való beállítás ideje az erősítési fokozattól függ, kis erősítési fokozatnál kb. 10 μ s, a maximális erősítésnél közel 300 μ s. Az erősítő bemenete szimmetrikus, lebegtetett, de nem galvanikusan leválasztott, kimenete aszimmetrikus. Megvalósítottak egy modulon belül erősítő/leosztó egységet, a modul erősítése/csillapítása 0... \pm 60 dB között 6 fokozatban állítható. A modulnak van 2-csatornás változata is.



4.23. ábra

Mintavevő és tároló- (S & H) erősítő blokkvázlata és utasításkészlete [8]

A 4.23. ábrán 2-csatornás mintavevő és tárolómodult mutatunk be. A modulban két független mintavevő és tároló- (sample and hold) erősítőt találunk, amelyek alaphelyzetben (Z.S2-re) vagy a követés bemeneten adott jelre másolják a bemenő jelet. A tárolás bemenetre adott jelre mindkét erősítő átvált tárolás üzemmódra és kimenetén őrzi a bemenő jel aktuális értékét. A tárolás bemenetre adott jelre a modul egyben LAM jelet is generál. ADATÚT vezérléssel, F(26) és megfelelő alcím-kombinációjával az erősítőket külön-külön vagy egyszerre követésre, F(24) és megfelelő alcím-kombinációjával pedig tárolásra állíthatjuk. Feszültségtartomány: \pm 10 V, tárolási pontosság: 50 μ V/ms, beállítás követésre az átváltási időt is beleértve: 5 μ s.

Az ADATÚT vezérlés nélküli modulok az ADATÚT-ból csak a tápfeszültséget kapják, de azzal nincsenek logikai kapcsolatban. Tulajdonképpen CAMAC modul mechanikába épített, önálló működésű funkcionális egységek. Ilyenek például:

- az impulzusnyújtó (stretcher).
- a spektroszkópiai erősítő.

- az összeadó, invertáló erősítő.
- a ns-os késleltető.
- a ns-os idő—amplitúdó-konverter.
- az amplitúdó—frekvencia-konverter.
- a referenciaszűrés generátor (egyenfeszűrés).
- a hiteles amplitúdójú impulzusgenerátor (FET vagy higany-relés kapcsolóval).
- a nagyfeszűrésű tápegység, detektorok számára.

Régebben ilyen jellegű modulokat szinte kizárólag a NIM szabvány szerint készítették. A CAMAC elterjedésével viszont egyre többet választhatóan NIM vagy CAMAC kivitelben készítenek. Egy-egy rendszer összeállításakor igen előnyös lehet, ha ezeket a modulokat is be lehet tenni a CAMAC keretbe közvetlenül adapter nélkül, vagy nem kell egy külön NIM keretet (ún. NIM—BIN-t) is alkalmazni, ahová viszont csak NIM modulok tehetőek, de CAMAC modulok egyáltalán nem.

Végül összefoglalásként arra mutatunk be példákat, hogy több bemenet esetén a feldolgozandó feszűrésztartomány, pontossági követelmény és holtidő (sebesség) figyelembevételével hogyan lehet *analóg mérőláncokat* felépíteni (4.24. ábra). A könyv-

Mérési tartományok	Analóg mérőlánc	Holtidő
$\pm 5\text{ V}$		40 μs
$\pm 0,5 \dots \pm 5\text{ V}$		60 μs
$\pm 50\text{ mV} \dots \pm 5\text{ V}$		5 ms
$\pm 50\text{ mV} \dots \pm 5\text{ V}$		25 ms

4.24. ábra
Analóg mérőláncok tulajdonságai

nyebb összehasonlítás miatt a példákban a felbontóképesség mindenhol 12 bit. A leggyorsabb mérést úgy érjük el, hogy egy gyors, például approximációs típusú A/D konverter elé egy 1-vezetékes FET-es multiplexert kapcsolunk. A közös földvezetéken jelentkező mV-os zavarjelek miatt 12-bites felbontáshoz 4...5 V-os bemenő jelet kell a multiplexer bemenetére kapcsolni. Ez a feszűrés egyben az A/D konvertert is kivezérli, így a multiplexert közvetlenül ráköthetjük az A/D konverter bemenetére. Az összeállítás hátránya, hogy a bemenő jeleket csatornánként kell erősíteni, azaz 5 V-os szintre hozni. Ha differenciál FET-es multiplexert használunk, a kisebb multiplexer-zajszint (0,1 mV) már megenged egy kb. tízszeres erősítés alkalmazását a multiplexer és a ADC között. Így a bemenő jel-szint $\pm 0,5\text{ V}$ -ra csökken. Ha az erő-

sítfő kézi beállítású, akkor az összes bemenő jelnek azonos, pl. $\pm 0,5$ V-os tartományban kell lennie, ha programozható, akkor $\pm 0,5$ V és ± 5 V között bárhol lehet, de amikor az egyes csatornákat felkeressük, az erősítés megfelelő értékét is be kell állítani (1...10 között). Ez viszont a holtidő kb. 20 μ s-os növekedését eredményezi.

A bemenő jel-szintet már csak relés multiplexer alkalmazásával lehet tovább csökkenteni. Különleges reed-relék és árnyékolással védett (guard) szimmetrikus jelvezetés mellett (10 μ V zaj) 12 bit felbontáshoz már ± 50 mV-os jel is elég. Az 50 mV-os jel közvetlenül képes kivezélni egy integráló típusú A/D konvertert, amelyet általában ellátnak ilyen kis érzékenységű bemenettel, ill. az érzékenységük ilyen kis értékre is beállítható programmal. Ez az összeállítás igen gazdaságos, széles mérési tartományt biztosít, a bemenő jelek ± 50 mV és ± 5 V között bárhol lehetnek, viszont a holtideje a legnagyobb, legalább 25 ms. A mérési időt csökkenthetjük, ha megtartjuk a 12-bites approximációs konvertert és a multiplexer és az A/D konverter közé ismét programozható erősítőt kapcsolunk. Ennek az erősítést max. százszorosra lehet állítani, ha azt akarjuk, hogy a zajok a legkisebb bitet ne haladják meg. Ennek a rendszernek a holtidejét lényegében a relék beállási ideje szabja meg. Ilyenkor természetesen elesünk az integráló A/D konverterek 50 Hz-es zajnyomásától, mivel a bemenő jel pillanatértékét kódoljuk és nem az adott időszakra vett átlagát. Ebben az esetben galvanikus leválasztást is a programozható erősítőben kell megoldani. Az integráló A/D konverterek a leválasztást általában közvetlenül biztosítják, mivel lebegtetett, leválasztott bemenettel készülnek; az approximációs típusúak, azonban többnyire aszimmetrikus, nagyszintű bemenettel rendelkeznek.

4.3.

Modulcsaládok

Az eddig tárgyalt modulok általában egy, ritkábban két, ill. három állomáshelyet foglalnak el, és mechanikusan és elektromosan egységet alkotnak. Bonyolultabb és a pillanatnyi igényekhez alkalmazkodóan bővíthető funkciókat ellátó feladatokra modulcsaládokat kell kialakítani. Ezek több külön álló modulból épülnek fel és már nem mechanikusan, hanem funkcionálisan képeznek egységet. Egy-egy feladat megoldására a modulcsaládból természetesen az éppen szükséges modulok kerülnek alkalmazásra.

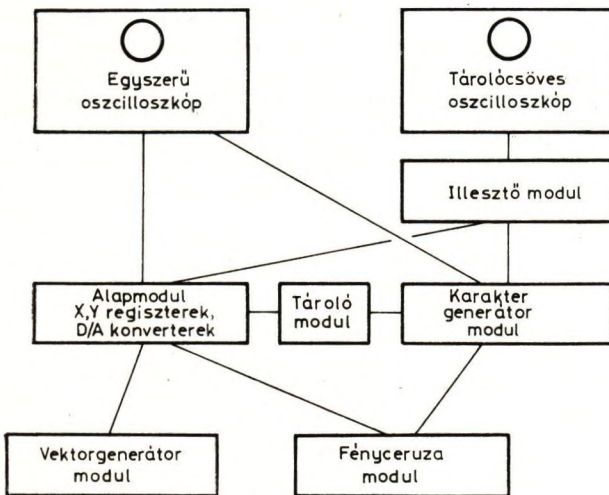
Részletesebben két igen eltérő alkalmazású modulcsaládot mutatunk be: egy moduláris oszcilloszkópos megjelenítő rendszert és egy ipari analóg mérőlánc családot.

4.3.1.

Moduláris oszcilloszkópos megjelenítő rendszer

Egy oszcilloszkópos megjelenítő modulcsalád 5–6 különböző modulból áll (4.25. ábra). *Alapmodulja* magában foglalja az egy ábrázolandó pont koordinátáit tartalmazó X és Y regisztereket és az ezekhez kapcsolódó digitális–analóg átalakítókat. A regiszterek az ADATÚT-ról betölthetők, a D/A konverterek kimenetei közvetlenül meg tudnak hajtani egy egyszerű oszcilloszkópot, vagy egy további *illesztőmodulon* keresztül egy tárolócsöves oszcilloszkópot. Az alapmodul áramkörti készletével voltaképpen mindenféle ábra megjeleníthető, ehhez azonban igen sok utasítás végrehajtására és ezzel sok gépidőre van szükség. Ezért egy további kiegészítő modul *karaktergenerátort*

tartalmaz, amelynek segítségével az adott karakterkészlettel megvalósítható szöveg igen egyszerűen előállítható. A karaktergenerátor különböző megoldásokkal illesz-
kedhet a rendszerbe: lehet önálló is, vagy pedig az alapmodul kiegészítése (az ábra
önállóan is működni képes karaktergenerátort mutat). Grafikus ábrák egyszerűbb és
gyorsabb rajzolására az alapmodul még vektor- és esetleg *körgenerátorral* is kiegészít-
hető. Az ábrázolandó egyenes kezdőpontjának koordinátáit be kell tölteni az alap-
modul X , Y regisztereibe, az X és Y növekményeket a vektorgenerátor ΔX és ΔY
regisztereibe, majd az így meghatározott vonal automatikusan kirajzolódik. Az alap-
modult és a vektorgenerátort külön kábellel kötik össze. Egy gyakori megoldás
szerint ez teljesen digitális: a vektorgenerátor az egyenes pontjainak megfelelő X
és Y digitális értékeket generál és azokat adja át az alapmodulnak. Az alfanumerikus
és a grafikus ábrázolás egyszerre is megvalósítható, ha alkalmazzuk mind a karakter-
generátort, mind az alap-, mind a vektorgenerátor modulokat. A modulcsalád tar-
talmaz még fényceruzát is: ez egyaránt használható az alfanumerikus és a rajzoló
modulokkal.



4.25. ábra
Oscilloszkópos megjelenítő modulcsalád

A megjelenített képtartalom adatai általában a főgép vagy a CAMAC vezérlő-
tárolójában vannak. Az egyszerű oszcilloszkópnál a kép állandó felrészítése még
autonóm adatátvitel esetében is igen sok gép-, ill. ADATÚT időt vesz igénybe.
Ezért a megjelenítő családot *tárolóval* is ki lehet egészíteni, amelybe egyszer kell
beírni a képtartalmat, és ezután a felrészítés ebből a tárolóból történik.

4.3.2. Ipari analóg mérőrendszer

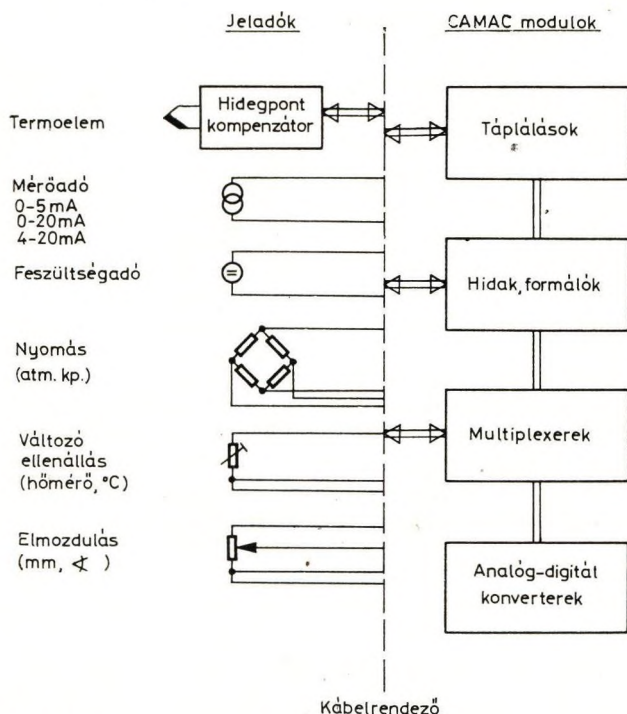
Az ipari analóg mérőrendszer moduljai nemcsak funkcionális, hanem konstrukciós,
mechanikai szempontból is egységes rendszert kell, hogy alkossanak. Előjáróban
az egyik legfontosabb, egységesen megvalósítandó feladatot, az elektromos zajok
elleni védekezést emeljük ki. Ennek — többek között — egyik módszere az, hogy

a teljes rendszert, az érzékelőtől a kábeleken, tápfeszültségforrásokon, multiplexeren, konverteren keresztül egységes, megfelelően földelt árnyékoló védő- (guard) „zsák”-kal veszik körbe: e „zsák” kialakítása csak összehangolt villamos és mechanikai konstrukcióval oldható meg kielégítően.

Az ipari rendszereknek általában a következő tulajdonságokkal kell rendelkezniük, ill. a következőket kell biztosítaniuk:

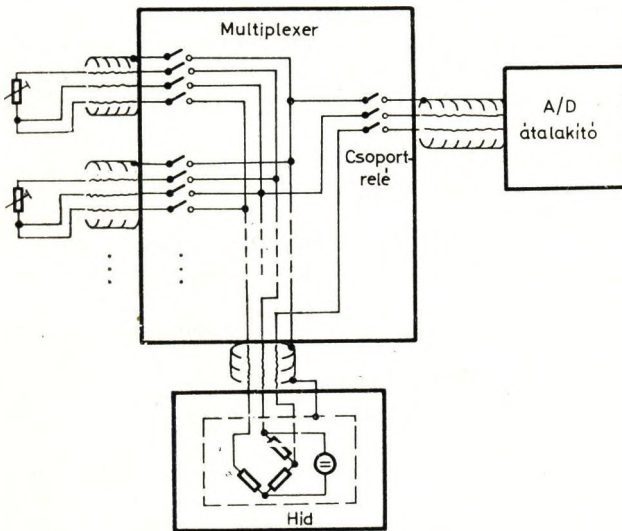
- Megbízható javított nagy háttérzajok esetében is.
- Lebegő, potenciálfüggetlen bemenetek.
- Különböző típusú mérőérzékelők (hőmérők, nyúlásmérőbéllyegek stb.) táplálása.
- Robbanásbiztos elektromos leválasztás.
- Egyszerű, gazdaságos bővíthetőség.
- Viszonylag nem túl nagy sebesség.
- Mechanikai igénybevételt jól tűrő csatlakozások, kábelezések.

A modulcsalád *funkcionális felépítése* olyan, hogy az ipari analóg méréseknél szokásos mindenféle típusú mérő érzékelőhöz a család valamelyik tagjával csatlakozni lehet, sőt, a szükséges táplálások is biztosítva vannak. A 4.26. ábra röviden vázolja a rendszerben használható legfontosabb jeladók és a családhoz tartozó modulok főbb típusait. A családhoz tartozó A/D konverter integráló típusú digitális voltmérő (4.2.5. pont), amelynek vázlata a 4.17. ábrán látható. Többféle analóg multiplexer kapcsolódhat a konverterhez. Mivel a rendszer lebegőpotenciálú és árnyékoló véde-



4.26. ábra
Ipari analóg modulcsalád

lemmel (guard) van ellátva, a legegyszerűbb multiplexernek is három vezetékét kell átkapcsolnia: a két jelvezetékét és a védőárnyékolást. Az átkapcsolást ezért reed-relek valósítják meg. A multiplexer bemenő vezetékén levő jeleket igen sok esetben egyedileg kell illeszteni, ezért minden bemenethez tartozik egy kis kártya, amely különböző szűrőket, osztóellenállást vagy hidat (esetleg egyszerű átkötést) tartalmaz a bemenő jelnek megfelelően. Egy modul 16 mérőhely bemenő jeleit tudja kezelni; az analóg—digitális konverterre azonban sok multiplexer modul jelei csatolhatók. A multiplexerek jelfogóinak gerjesztőáramát közös multiplexer vezérlőmodul szolgáltatja. A modulcsalád 4 vezeték átkapcsolását végző multiplexert is tartalmaz: ilyenkor a táplálás is átkapcsolható. A modulcsalád további elemei a hidak. Ezek vagy a multiplexerbe építve csatornánként egyedileg vannak megvalósítva, vagy pedig önálló modulként, saját tápfeszültségforrással együtt. Az általános tervezési szempontok természetesen ebben is érvényesülnek: a híd tápegységével együtt potenciálfüggetlen, lebeg és körbeveszi az árnyékolóvédelem (guard). Egy modulban nyolc egymástól is független híd helyezkedik el. A 4.27. ábrán vázoltuk a



4.27. ábra
Pt hőmérők mérési összeállítása 4-érintkezős multiplexerrel és közös hiddal

modulcsalád egy olyan összeállítását, amelyben Pt ellenálláshőmérők egy közös, önálló modulban levő hídhoz kapcsolódnak 4-érintkezős multiplexeren keresztül. A családnak még szintén CAMAC modulba épített tápegységek tartoznak, amelyek a multiplexeren levő hidak, potenciométer távadó áramkörök, erőmérőcellák stb. táplálására alkalmasak. A tápegység szintén lebegő típusú és árnyékolóvédelemmel rendelkezik; egy modulba két, egymástól független tápegység van beépítve. A rendszer kiegészíti a hőelemekhez tartozó hidegpontkompenzátor. Ez nem CAMAC mechanikában készült, mivel célszerűen a hőelem közelében van elhelyezve. Táplálását viszont már a CAMAC keretben elhelyezhető tápegységmodul végezheti.

A modulcsaládhoz tartozó csatlakozók, kábelek is megfelelnek az ipari követelményeknek. E kábelek méretei nagyobbak a laboratóriumokban használt szokásos kábeleknél; a csatlakozók is ennek megfelelően erősebbek: a példaképpeni modulcsalád AMP 201 356—3 és 925 173—2 típusú előlapi csatlakozókkal van felszerelve.

Irodalom a 4. fejezethez

- [1] Amplitude to Digital Converter, Type: NE 9060 NUCLEAR ENTERPRISES Ltd., Beenham, Reading RG7 5PR. England. Gépkönyv, 1975.
- [2] Analog—Digital—Umsetzer nach Doppelrampenprinzip. Type: ADC 15 D0200—1021. DORNIER GmbH., 799 Friedrichshafen, Pf. 317., 1973.
- [3] *Biri, J.—Blasovszky, N. S.*: Real-Time Data Collection and Processing in Biological Laboratories. 6th World Congress of Cybernetic Medicine, Napoli, 1972.
- [4] *Bossei, I. B.—Klessmann, H.*: A Standard Format for CAMAC Device Specifications. CAMAC Bulletin, No 7., p. 31., 1973.
- [5] *Christensen, P. et. al.*: Proposed Classification of Software for CAMAC. CAMAC Bulletin, No 9., p. 25., 1974.
- [6] Clock Pulse Generator, type CAM 5.01. MTA—KFKI 1525 Budapest, Pf. 49.
- [7] *Dhawan, S.*: YALE—NAL CAMAC System. IEEE Trans. Nucl. Sci. Ns—19. No., 1 p. 689., 1972.
- [8] Digital—Analog—Umsetzer (M09005—A0144) und Sample-und-Hold Verstärker (M09005—A0136) SIEMENS. Messwertverarbeitungssysteme, Katalog MS1, Teil 7, 1973.
- [9] Digitales Eingangsregister, type DO 200—2001 DORNIER GmbH, 799 Friedrichshafen, Pf. 317.
- [10] *Hawkins, P.*: A Fast Digital Multiplier for CAMAC. CAMAC Bulletin, No 11., p. 13., 1974.
- [11] Integrating, Dual-slope ADC, Type: 1241. BORER ELECTRONICS AG., CH—4500 Solothurn 2, 1974.
- [12] Interrupt Request Register, type CAM 2.09. MTA—KFKI, 1525 Budapest, Pf. 49.
- [13] *Lyon, W. T.*: An Evaluation of CAMAC Equipment in an Industrial Environment. Proc. of the First Int. Symp. on CAMAC, 1973., p. 211.
- [14] Memory Buffer, type JMT 20. SAIP—SCHLUMBERGER, Bp., 47—92220 Bagnaux, France.
- [15] Multiplexer Control Unit, Type: CAM 4.08—1 and 16—Channel Multiplexer, type: CAM 4.08—21/22. MTA-KFKI, 1525 Budapest, Pf. 49, 1974.
- [16] *Nicolaysen, O. PH.*: Decimal Classification of CAMAC Instrumentation. CAMAC Bulletin, No 7., p. 33., 1973.
- [17] *Nicolaysen, O. PH.*: Module Descriptor. CAMAC Bulletin, No 8., p. 26., 1973.
- [18] Parallel Input Register, type CAM 2.05. MTA—KFKI. 1525 Budapest, Pf. 49.
- [19] Quad-Scaler, type CAM 2.02. MTA—KFKI 1525 Budapest, Pf. 49.
- [20] *Zobrist, D. W.*: Real-Time Applications of CAMAC in Industrial Process Control. IEEE Tutorial Course on CAMAC, New-York, 1974.
- [21] 32-Input Multiplexer, type: CAM 4.07. MTA—KFKI. 1525 Budapest, Pf. 49., 1974.
- [22] 200 nsec Analogue-to-Digital Converter and Store. Type: 95/8190—2/6. CULHAM LABORATORY, Abingdon, Berks., Culham. UK. 1973.
- [23] 256 Word Store, type CS0015. NUCLEAR ENTERPRISES Ltd., Beenham, Reading RG7 5PR, England.
- [24] 8192—Channel ADC, MOD:8215. LABEN divisione della MONTEDEL S. p. A., 20133 Milano, Via E. Bassini 15., 1974.

5.

CAMAC software

A CAMAC rendszerek terjedése során minden felhasználónál kialakult az igény arra, hogy a programozás könnyebbé váljék. Ezért a felhasználó által használt alacsonyabb-magasabb szintű nyelveket mindenfelé kibővítették a CAMAC modulok vezérlését megkönnyítő kiegészítésekkel. Ily módon elterjedtek assembler szintű CAMAC vezérlők, makrók, FOCAL és BASIC kiegészítések, FORTRAN szubrutin-rendszerek CAMAC vezérlésre s í. t. Ezek a software kiegészítések különböző számítógépekre és különböző CAMAC vezérlőkre készültek és általában a magasabb rendű nyelvbe való beépítésük, a hívás módja is különböző volt.

Hamarosan egyre égetőbbé vált ezen egymástól különböző programrendszerek egységesítése. Az volt a törekvés, hogy egy-egy magasabb rendű nyelven ugyanúgy, azonos módon lehessen a CAMAC rendszert kezelni, adatokat átvinni és vezérelni, függetlenül attól, hogy milyen számítógépen és vezérlőn keresztül valósul meg a program. Ezért néhány további szabvánnyal egészítették ki a hardware rendszert. Ezek a LAM vezérlésre, különböző vezérlő-, maszk- és állapotregiszterek címeire vonatkoznak, és megtalálhatók már az EUR 4100 szabvány 1972. évi kiadásában.

Két példával mutatjuk be az egységesítés szükségességét. A modulgyártó cégek egyre bonyolultabb modulokat, modulcsaládokat gyártanak. Például az egyik moduláris, oszcilloszkópos megjelenítő modulcsalád 4 modulból áll és a modulok vezérlése 52 db különböző utasítással történik. A modulokat gyártó cég a leírásban megadja ezeket a kódokat, de semmilyen formában nem utal arra, hogy az általában szokásos egyszerűbb funkciókat hogyan kell célszerűen programozni. Mivel a modulokat tetszőleges számítógéppel és vezérlővel lehet használni, nem is várható el a cégtől, hogy programokkal együtt szállítsa a modulokat. Az egységes IML nyelv elterjedésével viszont a modulgyártó cégektől megkövetelhető, hogy bonyolultabb moduljaikkal megfelelő alapsoftware-t szállítsanak IML nyelven, például az említett megjelenítő modulcsaládhoz egyenest, kört stb. rajzoló rutinokat.

Hasonló a helyzet a modulok tesztprogramjaival. Egységes nyelv hiányában nem terjedt el általánosan, hogy a gyártók tesztprogramokat is adnak a modulokhoz. Egységes IML vagy CAMAC-kal kiegészített BASIC nyelv szabványosítása esetén ezen a nyelven írt tesztprogramokkal kell a modulokat tesztelni.

Az egységesítési törekvéseknél a nagyobb problémát az jelentette, hogy ki kellett választani, melyik magasabb rendű nyelvbe vagy nyelvekbe épüljenek be a CAMAC szabványú hívások. Jelenleg ugyanis még nem alakult ki sem Európában, sem az USA-ban egységes, magasabb rendű real-time nyelv, és kialakulása néhány éven belül nem is várható. Nem látszik célszerűnek az sem, hogy egy, az egyéb területeken folyó real-time programozási erőfeszítésektől független új, önálló CAMAC nyelv alakuljon ki. Ezért a jelenlegi software szabványosítási munkák során egy viszonylag

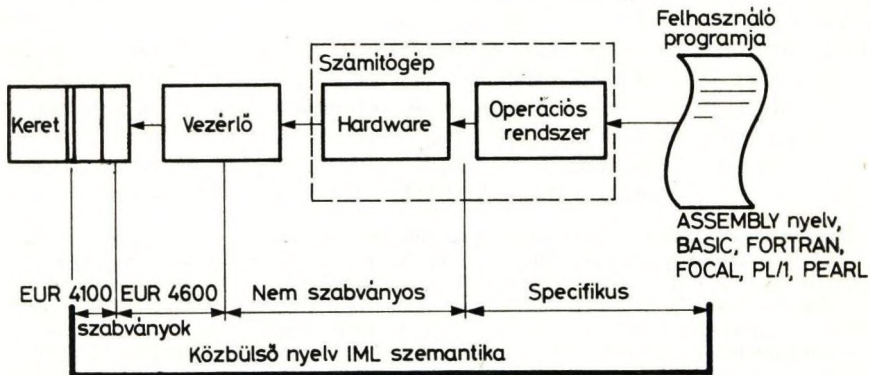
alacsony szintű, ún. *közbülső nyelvet* (*Intermediate Language*® – IML) dolgoztak ki. Ennek definícióját, szemantikáját fogadták el szabványosnak és ezt implementálták különböző formában különböző nyelvekre. Így például rögzítésre került az IML-nek egy olyan szintaxisa, amely makrofordítókkal használható. Ennek tipikus felhasználása az, hogy az IML utasítások egy assembly nyelven írt programba vannak beágyazva, és az assembler program makrolehetőségei fejtik ki, építik be a tárgyprogramba az IML utasításoknak megfelelő gépi kódot. Másik példa az IML implementálásra FORTRAN szubrutinok szabványosítása, ill. a CAMAC modulokat hívó szubrutinok paramétereinek definiálása az IML szemantikának megfelelően.

A továbbiakban részletesen foglalkozunk az IML nyelvvel, majd ismertetjük az eddigi főbb, különböző CAMAC programozási módszereket, eszközöket. Ezek a megvalósított rendszerek még nem, vagy csak kevéssé veszik figyelembe az IML szemantika előírásait, mivel ezek csak jelenleg (1974 őszén) rögzítődtek.

5.1.

Az IML nyelv

A CAMAC számítógépes rendszer határozott *hierarchikus felépítésű* (5.1. ábra). A struktúra legeljén található az egyes CAMAC modulok, ezek csoportja alkotja a keretet, a keretek összessége a csoportot. A hardware rendszer definiálása után a struktúrában a következő fokozat a CAMAC hardware-rel való software kapcsolat. Az IML a CAMAC hardware-rel való kapcsolat legelső szabványosított szintje.



5.1. ábra
Software helye a CAMAC rendszerben

Az IML nyelvet valamely szokásos nyelvvel együtt, abba beleépítve kell használni. Az IML nyelv *nem önálló, hanem a „vendéglátó” programrendszer kiegészítése* olyan utasításokkal, amelyek lehetővé teszik a CAMAC hardware minden lehetőségének, tulajdonságának kihasználását. A nyelv szemantikáját definiálták alapvetően. A szintaktikus formák viszont — az implementálás megkönnyítése miatt — változnak a „vendéglátó” rendszertől függően. Három fő irányban dolgozzák ki a szintaktikus szabályokat: makro, transzlátor és szubrutin—hívás szintaxisra. A „vendéglátó” programrendszer az operációs rendszert, a futás közbeni vezérlőprogramot és azt a nyelvet tartalmazza, amelyen az adatkezelő programot írták.

Az IML nyelv — mint valamennyi nyelv — állítások* (statement) sorozatából áll. Az állítások kétféle típusúak: utasítás jellegű állítások (action statement) — továbbiakban utasítások — és deklarációk. A deklarációknak az a célja, hogy valamely azonosítóhoz tartozó összes tulajdonságot egyetlen állítással lehessen specifikálni. Bizonyos implementációknál a deklaráció teszi lehetővé, hogy a programrendszer megfelelő számú memóriahelyet biztosítson a változó számára a program futásakor. Az utasítás jellegű állítások a CAMAC hardware-nek adnak utasításokat. Ezek vagy a modulra vonatkoznak (modul action statement), vagy a LAM kérésekkel kapcsolatosak, vagy pedig az egész rendszerre (keret-, ill. rendszervezér) vonatkoznak.

Az IML szemantikájának leírását az [6] irodalom közli. Hét fejezetből áll, ezek a következők:

1. A leírás értelmezése.
2. Az IML vázlatos leírása.
3. Alapdefiníciók.
4. Címparaméterek.
5. Működési módok.
6. Állítások.
7. Referenciák.

A továbbiakban nem részletezzük az IML nyelv valamennyi részletének definícióját, hanem a makroszintakszisból vett példákkal illusztrálva ismertetjük főbb tulajdonságait.

5.1.1.

Deklarációk

A deklarációk rögzítik valamely azonosítóhoz tartozó tulajdonságokat, jellemzőket. Igen sok féle tulajdonságot lehet rendelni a különböző azonosítókhoz, ezeket a szemantika részletesen felsorolja.

A makroszintakszisban a következőket lehet deklarálni:

- külső referenciát (eszközcímet),
- belső referenciát (tárolócímet),
- csatornanevet és
- L (LAM) nevet.

Külső referencia deklarációja

Külső referenciával a legegyszerűbb esetben közvetlen hardware címet (CAMAC címet) határozzunk meg. Ez négy paramétert tartalmaz: csoportcímet (B), keretcímet (C), állomásszámot (N) és alcímet (A).

A deklaráció a következő formájú:

LOCD TIMER, H, <1, 2, 3, 0>

LOCD jelzi azt, hogy lokális eszköz- (Device) deklaráció az állítás. TIMER az azonosító, a későbbi utasításokban ezen a néven fogunk hivatkozni az 1. csoportút 2. keretének

* A magyar software irodalom a statement kifejezésre általában az utasítás szót használja. Mivel azonban ez az instruction-nal könnyen összetéveszthető, ezért itt az állítás kifejezést használjuk.

3. állomáshelyén levő modul 0. alcímű regiszterére. Az azonosító után levő H betű jelzi, hogy ez a deklaráció közvetlenül hardware-re vonatkozik. Ez azt is jelenti, hogy az eszköz címe futás közben nem változtatható.

Ha a hardware eszköz címét futás közben változtatni akarjuk, akkor közvetlen cím helyett *indirekt* módon is deklaráálhatjuk. Ilyenkor az azonosító a hardware eszköz címét (BCNA-t) tartalmazó tárolóhelyre mutat, ezért eszközmutatónak (Device Pointer) nevezik. Ennél a deklarációnál az azonosító által definiált tárolóhely kezdeti értékét is meg lehet adni. Példaként a következő deklarációt mutatjuk be:

LOCD TIMER, P, ⟨1, 2, 3, 0⟩

E deklarációnál P jelzi azt, hogy indirekt hardware eszközre vonatkozik a deklaráció, TIMER annak a memóriahelynek a neve, amely a BCNA címét tartalmazza. TIMER kezdőértéke 1, 2, 3, 0, ezt azonban a program futása közben változtatni lehet.

Nemcsak egyes CAMAC címeket, hanem ezekből álló tömböket is lehet deklaráálni. A CAMAC tömb (Given CAMAC Array) deklarálása például a következő:

LOCD REGS, G, 3, ⟨1, 2, 7, 0⟩, ⟨1, 3, 6, 1⟩, ⟨1, 2, 2, 0⟩

A tömb neve REGS, a G jelzi azt, hogy előre adott tömbről van szó; a tömb 3 CAMAC címből áll, ezeket tartalmazzák a ⟨⟩ zárójelek.

Sűrített formában is megadható egy CAMAC címekből álló tömb: B, C, N, A értékek kezdő, befejező és növekmény értékeivel. Ezt nevezik *számított (Calculated) tömbnek*. Példa számított CAMAC címtömbre:

LOCD REGS, C, 60, ⟨1, 3, 2, 0⟩, ⟨1, 5, 10, 3⟩ ⟨0, 1, 2, 1⟩

A REGS nevű tömb 60 hardware címből áll; az első cím: B=1, C=3, N=2 és A=0. A 2.⟨⟩-ben levő címcsoport az utolsó címet tartalmazza, az utolsó zárójel pedig a növekményt. Így a teljes tömb 4 alcímből áll (0...3-ig egyesével növekedve), 5 állomáshelyből (2...10-ig kettesével növekedve) és 3 keretből (3...5-ig egyesével növekedve).

Az adott, ill. a számított tömbökre az utasításállítások kétféleképpen hivatkozhatnak. Vagy a teljes tömbre vonatkozik egy utasítás, vagy a tömb egyes elemére. Ha a tömb egyes elemeire vonatkozik az utasítás, akkor a kívánt elem sorszámát indexben kell a hivatkozáskor megadni. Ilyenkor a deklarációnál a GS, ill. a CS jelet kell használni (S: subscript).

Az EUR 4100-ban definiált címletapogatásnak felel meg egy további CAMAC tömb deklarálás: a Q *letapogatási tömb*. A deklaráció formája példaképpen a következő:

LOCD REGS, Q, ⟨2, 4, 1, 0⟩, ⟨2, 6, 5, 7⟩

Q jelenti a Q letapogatási tömböt, az első BCNA érték 2, 4, 1, 0; az utolsó 2, 6, 5, 7. A címtartomány a Q választól függően az EUR 4100-ban definiált módon végigsöpri az első értéktől az utolsóig a teljes tartományt. Az egyes hardware elemek száma előre nincs meghatározva (a Q választóktól függ!).

Belső referencia deklarálása

Az IML-ben használt belső referenciákat (tárolóreferenciákat) is deklaráálni kell a makro assembler számára. Ezek lehetnek konstansok (K), változók (V — Variable),

tömbök (A — Array) és pufferek* (B — buffer). A

LOCM PERIOD, K, 16, 6000

deklaráció értelme a következő: LOCM jelenti a belső (memória) referenciát, PERIOD az azonosító neve, K jelenti azt, hogy konstans, 16 jelöli a szóhosszat (16 bit), 6000 a konstans értéke.

Változóra példa a következő deklaráció:

LOCM TEMP, V, 16

Ennél TEMP a változó neve, V jelenti azt, hogy változó, 16 jelöli a szóhosszat.

A következő példa tömböt deklarál:

LOCM EVENTS, A, 24, 60

Az EVENTS nevű tömb 60 darab 24-bites szóból áll.

A változó lehet logikai változó is. Erre példa:

LOCM FLAG1, VL

Ha a tömbnek egyes elemeire külön is hivatkozunk az utasításállításokban, akkor ezt a deklarációban is jelezni kell:

LOCM EVENTS, AS, 24, 60

Az EVENTS nevű tömb 60 darab 24-bites szóból áll, és az utasításokban indexszel lehet a tömb egyes elemeire hivatkozni.

A puffer abban különbözik a tömbtől, hogy ha egymás után következő utasításokkal hivatkozunk a deklarált pufferre, akkor ezek a hivatkozások automatikusan a puffer egymás után következő elemeire vonatkoznak. A puffer egyes elemeit viszont indexszel nem lehet elérni. Példa a puffer deklarálására:

LOCM BUF3, B, BUFF, 16, 128, FULL

A BUF3 nevű pufferdeklarálás (B jelenti, hogy puffert deklaráltunk) a BUFF nevű adattömbön legyen értelmezve. Az adatok 16 bit hosszúak, a puffer mérete 128 szó. Ha különleges helyzet áll elő, ugorjon a program a FULL címkejű utasításra. („Különleges helyzet” definíciója az implementálástól függ. Lehet például puffer üres vagy tele, esetleg hibaállapot.) Ugyanazt az adattömböt többször is deklarálhatjuk, különböző programelágaztatással. Például:

LOCM BUF4, B, BUFF, 16, 128, ERROR

Ez ugyanarra a BUFF adattömbre vonatkozik, mint az előző deklaráció, de ha erre hivatkozunk az utasításban, akkor a „különleges helyzet” elérésekor az ERROR nevű címkére ugrik a program.

Csatorna deklarálása

Ha a „vendéglátó” rendszer rendelkezik perifériás ki/bemenő csatornákkal, akkor ezeket a csatornákat is lehet deklarálni. Példaképpen deklaráljuk a következőt:

LOCC OUT2, 30, TALLY2, STAT2

* A magyar hardware szakirodalom a puffer, a software szakirodalom a buffer kifejezést használja szívesebben. E leírásban az egységes tárgyalásmód miatt a software leírásnál is a puffer szót használjuk.

LOCC jelöli, hogy csatorna deklarációjáról (Channel) van szó, OUT2 a csatornaparamétereket tartalmazó blokk neve, 30-szor kell a csatornán az utasítást végrehajtani, TALLY2 nevű változó tartalmazza azt, hogy eddig hány átvitel zajlott le, STAT2 tartalmazza e csatorna állapotbitjeit.

LAM deklaráció

A 3.1.4. pontban láttuk, hogy a LAM-ot kérő modul azonosítása a leggyorsabban a GL — osztályozott LAM kérelem — utasítással történik, ilyenkor a GL1—GL24 bitekből álló minta minden bitjéhez egy állomáshely, ill. modul tartozik. A modulon belüli LAM forrás azonosítására két módszer van (l. 4.1. szakaszt): vagy alcím (A) szerint lehet azonosítani, vagy pedig a LAM állapotregiszter bitjeinek állapota alapján. Az IML nyelvben ennek megfelelően kétféle deklarációval lehet egy LAM forrást azonosítani:

LOCL READY, SUB, <1, 2, 4, 7>, 5

READY a neve az 1. csoport 2. keret 4. állomáshelyén levő 7. alcímen elérhető és a GL=5 osztályozott LAM helyen jelentkező kérésnek. SUB jelenti azt, hogy ez a LAM alcímmel (Subaddress) azonosítható.

A következő példában a BIT megjelölés jelenti azt, hogy a LAM állapotregiszter bitjeivel azonosítható az L kérelem.

LOCL NEXT, BIT, 6 <1, 1, 10>, 8

A deklaráció jelentése: NEXT a neve a 8. GL helyen jelentkező, 1. csoport 1. keret 10. állomáshelyén levő modul LAM állapotregiszterének 6. bitjén elérhető LAM kérésnek.

Lokális és globális deklarációk

Az eddigi példák olyan deklarációkat mutattak be, ahol egy programszegmensen belül történt a deklarálás és a deklarált azonosítóra való hivatkozás. Ezért ezeket helyi (local) deklarációknak nevezik. Az IML lehetővé teszi azonban azt is, hogy különböző programszegmensekben történjen a deklarálás és a rá való hivatkozás, azaz a deklarációban definiált azonosítónak valamely utasításban való felhasználása. Ha a deklarált azonosítót nemcsak a saját programszegmensében akarjuk használni, akkor nem lokálisan, hanem globálisan kell deklarálni. Ennek neve: *export deklaráció*. A lokális és az export deklaráció közötti különbség csak a definiáló makro nevében van, a paraméterek megegyeznek. Így pl. a külső referencia deklarálásakor LOCD helyett EXPD-t kell írni stb.

Az *import deklaráció* a különállóan fordított programban definiált azonosítókat deklarálja. Az import deklarációban csak az éppen fordítás alatt levő program sikeres fordításához szükséges tulajdonságokat kell megismételni. Az import deklaráció jele például külső referencia deklarálásakor IMPD.

5.1.2.

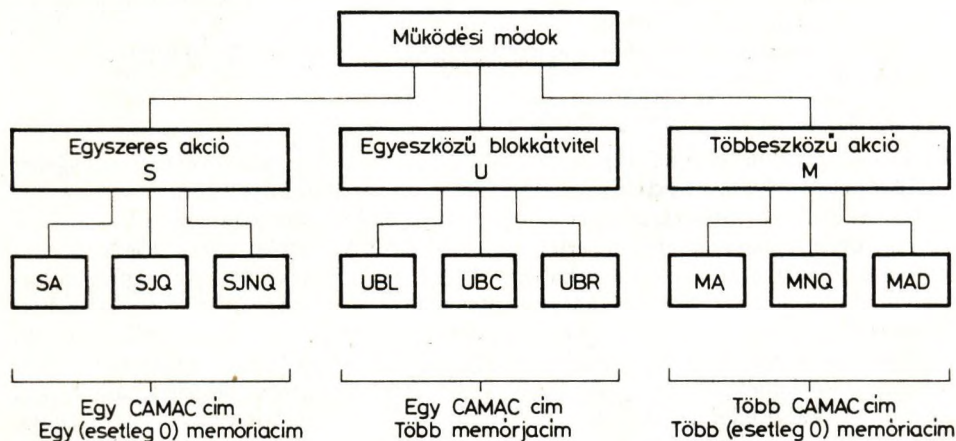
Utasítások

A CAMAC hardware-nek adható utasítások három csoportra oszthatók:

- modulutasítások,
- LAM vezérlő-utasítások és
- rendszerre vonatkozó utasítások.

Modulra vonatkozó utasítások

A modulokat vezérlő utasítások az egyes modulokban levő alcímekre vonatkoznak. Ez az általános módja a CAMAC regiszterekkel való műveleteknek. Az utasítás általában tartalmazza a működés módját, külső referenciát (CAMAC eszközcímet), belső referenciát (tárolócímet), csatornaparamétereket és esetleges ugrási vagy visszatérési címet.



5.2. ábra
Működési módok osztályozása az IML nyelv szerint

A modulokra vonatkozó utasításoknak 3 működési módja van (5.2. ábra):

- egyszeres akció (single action: S);
- egyeszközü blokkátvitel (uni-device block transfer: UB);
- többeszközü akció (multi-device action: M).

Az *egyszeres akció* az IML-ben definiált legegyszerűbb utasítás. Egyetlen ADATÚT műveletet indít és írás—olvasás funkció esetében egyetlen szót visz át. A makroszintaxis feltételezi, hogy az egyszeres akciójú utasítás befejeződik, mielőtt a program következő utasítása elkezdné. Az egyszeres akciójú utasítás a funkciókódot, egy CAMAC címet és adatátvitel típusú műveletnél egy tárolócímet tartalmaz:

SA RD1, TIMER, TEMP

SA jelöli az egyszeres akciót (Single Action), RD1 a funkciót, TIMER a CAMAC címet és TEMP a tárolócímet. TIMER és TEMP azonosítókat már előzőleg külső, ill. belső

referenciával deklarálni kellett. A funkció helyére vagy a megfelelő F0, F1* stb. kódot, vagy szabványos mnemokódot (például F0-nak RD1 felel meg) kell írni. Az utasítás jelentése: olvasd ki a TIMER nevű CAMAC regiszter tartalmát a TEMP nevű memóriahelyre.

Ha nincs adatátvitel, a tárolóreferencia hiányzik. Például az

SA F9 TIMER

utasítás azt jelenti, hogy a TIMER nevű CAMAC regiszteren az F9 szabványos utasítást kell végrehajtani, azaz törölni kell.

Az egyszeres akciójú utasításoknál lehetőség van arra, hogy a modulból érkező Q válasz hatására a program elágazzon. Ilyenkor az ugrási címet is meg kell adni. A következő példában a program akkor ugrik el, ha $Q = 1$.

SJQ RD1, ADC, TEMP, LIN

SJQ jelöli a Q feltételre való ugrással kombinált egyszeres akciójú műveletet (Single Jump on Q), a továbbiak pedig azt jelentik, hogy ki kell olvasni (RD1) az ADC nevű CAMAC regiszter tartalmát a TEMP nevű tárolóhelyre, és ha a művelet végén Q értéke 1, akkor el kell ugrani a LIN nevű programrészre. Ha Q értéke 0, a program a következő utasítással folytatódik.

Egy további utasítással azt lehet meghatározni, hogy a program $Q = 0$ válasz esetében ugorjék el. Erre példa

SJNQ RD1, ADC, TEMP, AGAIN

Ebben az esetben, ha az ADC regiszter kiolvasásakor $Q = 0$ válasz érkezett, a program az AGAIN című utasításnál, $Q = 1$ esetében folyamatosan folytatódik.

A következő nagy utasításcsoport az egyszerű *blokkátvitel* kizárólag adatátvitelre (írás vagy olvasás) használható, egyetlen rögzített CAMAC cím és a számítógép tárolójában levő egymás után következő helyek között. Ha a felhasználónak megfelelő csatornavezérlője van, akkor a blokkátvitel utasítás csatornaazonosítót is tartalmazhat. A csatornaazonosító értelmezése attól a rendszertől függ, amelyekben alkalmazzák.

A blokkátvitelnél nincs kikötve, hogy befejeződjék, mielőtt a program következő utasítása elkezdődne. Ha a „vendéglátó” rendszernek van multi-tasking tulajdonsága, akkor a blokkátvitellel egyidőben a program további utasításait is folytatni tudja.

A blokkátvitel három üzemszerű ok következtében fejeződhet be:

- a szószám (ismétlési szám) lejárt,
- a fogadóblokk tele van
- az adóblokk kiürült.

Ezen kívül természetesen hiba esetében is leállhat az átvitel. A CAMAC modul az állapotát minden átvitel után a Q válasszal tudja jelezni: minden érvényes átvitelnél $Q = 1$. Ha a modulnak már nincs több adata vagy nem tud több adatot átvenni, akkor $Q = 0$ választ ad.

A blokkátvitel lehet aszinkron vagy szinkron. Az *aszinkron átvitel* azt jelenti, hogy minden egyes átvitelt a modulból jövő jel kezdeményez. Ez a jel az átvitel során törölődik. A modulból érkező, átvitelt kezdeményező jel érkezik a szokásos L vezetéken keresztül LAM jel formájában, de jöhet speciális vezetéken is. Az aszinkron átvitel jele: UBL. Ez a leggyakoribb, leginkább ajánlott blokkátvitel. Példaképpen utasítás a következő:

UBL WT1, TAPE, OUTBUF, READY, CHAN5

* A funkciókód értékét általában F és utána zárójelben a kód száma (pl. F(5)) jelöli CAMAC leírásokban. Az IML nyelv makroszintakszisa viszont nem használ zárójelet a funkciókód jelölésénél.

UBL jelenti azt, hogy LAM-mal kezdeményezett egyeszközű blokkátvitel van, WT1 az 1. csoportú írás (F16) mnemokódja, TAPE a külső eszköz (CAMAC regiszter) címe, OUTBUF a memóriatömb címe, READY a LAM jel azonosítója, CHAN5 csatornaazonosító. Ezeket az azonosítókat természetesen előzőleg deklarálni kellett.

A szinkron átvitelt a vezérlő kezdeményezi. Ennél feltételezzük, hogy az adat a modulban igen gyorsan rendelkezésre áll, és így az átvitel ütemét a vezérlő szabja meg. Jele: UBC. Példa szinkron átvitelre:

UBC WT1, TAPE, OUTBUF, CHAN5

Látható, hogy az utasítás formája megegyezik az aszinkronéval, csak hiányzik a LAM megjelölése.

A szinkron átvitel másik fajtája az „ismétlés, amíg nincs Q”. Jele: UBR. Megfelel az EUR 4100-ban leírt ismétlés üzemmódnak (Repeat Mode).

A többeszközű akciók tulajdonképpen egyetlen CAMAC funkciókóddal végrehajtott, egyszeres akciók sorozata. Sok CAMAC címet tartalmaznak, ill. adatátvitelkor sok tárolócímet is, melyek teljesen megfelelnek egymásnak. Ugyanúgy, mint a blokkátvitelnél, megvalósíthatók akár hardware, akár software módon. A többeszközű akció-utasítás tartalmaz ismétlési számot, csatornaszámot, állapotváltozót és minden pillanatban az addig végrehajtott műveletek (átvitelek) számát.

A legegyszerűbb többeszközű akció az MA- (Multiple-Device Action)-val jelölt mód. Adatátviteli és működtető utasításokkal használható. Ha adatátviteli utasítást tartalmaz, akkor CAMAC regiszterek és a számítógép tárolója között folyik le az átvitel úgy, hogy minden CAMAC regiszterbe vagy regiszterből egy szó kerül átvitelre.

Példa többeszközű akciójú utasításra:

MA RD2, REGS, INBUF, CHAN2

Az utasítás jelentése: RD2 funkcióval (RD2=F1: kettes csoportú regiszter olvasás) ki kell olvasni a REGS néven deklarált CAMAC regiszterekből álló tömböt az INBUF néven deklarált memóriatömbbe a CHAN2 néven definiált csatornán keresztül.

A következő többeszközű utasítás modulok állapotának a Q válasszal való gyors vizsgálatára szolgál. A végrehajtás során az utasítás végigsöpör az egymásután következő CAMAC címeken egészen addig, amíg Q=1 nem érkezik. A vizsgálat vagy a LAM állapot (TLM: Test LAM), vagy egy speciálisan definiált tetszőleges állapotra (TST: Test Status) vonatkozik. Példaképeni utasítás a következő

MNQ TLM, REGS, CHAN2

Az utasítás azt írja le, hogy a REGS-ben deklarált CAMAC regisztertömböt kell növekvő sorrendben LAM lekérdező Q vizsgálattal végigvizsgálni egészen addig, amíg Q=1 válasz nem érkezik. A végrehajtást a CHAN2-ben definiált csatornán kell végezni.

Az EUR 4100 CAMAC specifikáció tartalmazza a címletapogatási üzemmódot (Address Scan). Ennek programozását könnyíti meg a MAD jelű utasítás. Ezt az utasítást természetesen csak akkor lehet használni, ha az egyes modulok hardware-je a címletapogatási üzemmódnak megfelelő Q választ adja. A következő utasítás a MAD módra példa:

MAD RD2, REGS, INBUF, CHAN2

5.1. táblázat

Modulra vonatkozó utasítások összefoglalása

	Mód	Külső referencia	Belső referencia	Csatorna- paraméter	Ugrási vagy visszatérési cím	
a) Egyszeres akciók						
	Olvasás	SA, SJQ, SJNQ	H, P, G, C	V, A, B	nincs	SJQ és SJNQ-nál
	Írás	SA, SJQ, SJNQ	H, P, G, C	K, V, A, B	nincs	SJQ és SJNQ-nál
	Működtetés	SA, SJQ, SJNQ	H, P, G, C	nincs	nincs	SJQ és SJNQ-nál
Állapot-olvasás	SA	H, P, G, C	VL	nincs	nincs	
b) Egyszeres akciók blokkátvitel						
	Olvasás, írás	UBL, UBC, UBR	H, P, G, C	A, B	van	opcionális
c) Többszörös akciók						
	Olvasás, írás	MA	G, C	A, B	van	opcionális
	Működtetés	MA, MNQ	G, C	nincs	van	opcionális
	Olvasás, írás	MAD	Q	A, B	van	opcionális
Működtetés	MAD	Q	nincs	van	opcionális	

Az ebben a példában szereplő REGS regisztercsoportot természetesen Q letapogatósi tömbbel kell deklarálni.

Az 5.1. táblázat összefoglalja a modulokra vonatkozó utasításokat. A táblázatban a külső, ill. belső referenciaként szereplő rövidítéseket a deklarációk leírásánál adtuk meg!

LAM kezelő utasítások

Azért, hogy a modulokban levő LAM kezelés programozása egységes legyen, külön utasításcsoportba vannak összefogva a modulokon belüli LAM-mal kapcsolatos tevékenységek. Ugyanezeket a működési módokat a modulokra vonatkozó egyszeres akciójú utasításokkal is el lehet érni. Mivel azonban néhány „vendéglátó” rendszerben a LAM kezeléssel kapcsolatos megszakításkérő tevékenységet a normális ki/bemenő utasításoktól különválasztva kezelik, és ezekkel a futás közbeni vezérlőprogram foglalkozik, célszerűnek látszott a LAM-mal kapcsolatos utasításokat különválasztani.

A LAM-mal kapcsolatos utasítások engedélyezésre, felfüggesztésre, törlésre, L-től függő utasításátlépésre és különböző állapotregiszterek kiolvasására vonatkoznak.

A LAM deklarációnál (5.1.1. pont) láttuk, hogy a LAM-ot kérő modul azonosítására az osztályozott LAM kérelem (GL) használható, míg a modulon belüli azonosításra két hardware lehetőség is van. Ennek megfelelően vagy alcímmel (SUB), vagy a LAM regiszter megfelelő bitpozíciói (BIT) szerint kell egy LAM nevet deklarálni. Bemutatunk néhány példát LAM kezelő utasításra.

ENL NEXT

Jelentése a következő: a NEXT azonosítóval deklarált LAM engedélyezése (Enable). További példa:

IFL NEXT, CHANGE

Jelentése: Ha a NEXT azonosítóval deklarált LAM=1, akkor ugorjon a program a CHANGE címkére.

Az 5.2. táblázatban összefoglaltuk a LAM kezelő utasításokat. A megfelelő paramétereket a deklaráció részben definiálni kell. A LAM nevek deklarációjától függően a futás során vagy a SUB, vagy a BIT oszlopban feltüntetett szabványos CAMAC funkciók kerülnek végrehajtásra. A modulok hardware tulajdonsága így csak a deklarációs részben tükröződik.

Rendszerre vonatkozó utasítások

A rendszerre vonatkozó utasítások a keret-, csoport-, ill. rendszervezérlőkre vonatkoznak. Tulajdonképpen ezeket is lehetne modulra vonatkozó utasításokkal helyettesíteni, azonban ha külön csoportban szerepelnek, akkor a programok kevésbé implementációfüggőek. A rendszerre vonatkozó utasítások mindig egyszeres akciók.

Ezek az utasítások az alaphelyzetbe hozással, törléssel, tiltással, LAM kérés kezelésével, X hiba kezeléssel, Q vizsgálattal kapcsolatosak részint keret, részint rendszer (csoport) szinten. A keretre vonatkozó utasítások az A1 típusú szabványos keretvezérlőt tekintik elsősorban hardware alapnak.

Példák rendszerutasításokra:

CZ <1, 2>
ENBD 1

5.2. táblázat

Modulra vonatkozó LAM kezelő utasítások összefoglalása

MNEMO-kód	Működés	Fünció		Paraméterek
		SUB	BIT	
ENL	LAM engedélyezés	F(26)	F(19) A(13)	LAM név
DISL	LAM felfüggesztés	F(24)	F(23) A(13)	LAM név
CLRL	LAM törlés	F(10)	F(23) A(12)	LAM név
IFL	Ugrás, ha L = 1	F(8)	A(14) bitpozíció teszt	LAM név, címke
IFNL	Ugrás, ha L = 0	F(8)	A(14) bitpozíció teszt	LAM név, címke
IFS	Ugrás, ha LAMSTAT = 1	F(27)	A(12) bitpozíció teszt	LAM név, címke
IFNS	Ugrás, ha LAMSTAT = 0	F(27)	A(12) bitpozíció teszt	LAM név, címke
RL	LAM kiolvasás	F(8)	A(14) bitpozíció teszt	LAM név, belső referencia
RLS	LAMSTAT regiszter kiolvasás	—	F(1) A(12)	külső referencia,
RLR	LAMREQ regiszter kiolvasás	—	F(1) A(14)	belső referencia
RLM	LAM MASK regiszter kiolvasás	—	F(1) A(13)	külső referencia,
WLM	LAM MASK regiszter kiolvasás	—	F(?) A(13)	belső referencia
				külső referencia,
				belső referencia

Az első utasítás jelentése: alphelyzetbe állítás a Z buszvezetékkel az 1. csoport 2. keretében.

A második utasítás: az 1. csoportban a rendszer LAM kérések engedélyezése.

Az 5.3. táblázat a keretre vonatkozó utasításokat, az 5.4. táblázat a csoportra vonatkozó utasításokat foglalja össze.

5.3. táblázat

Keretre vonatkozó utasítások

MNEMO-kód	Működés	Paraméterek
CZ	Keret iniciálás (Z buszvezeték)	B, C cím
CC	Keret törlés (C buszvezeték)	B, C cím
WSN	Állomászám-regiszter (SNR) beírás	BC cím, belső referencia
SETCI	Tiltás busz 1-be állítása	BC cím
CLRCI	Tiltás busz törlése	BC cím
ENCD	Keret LAM engedélyezés	BC cím
DISCD	Keret LAM felfüggesztés	BC cím
IFCI	Ugrás, ha keret tiltás = 1	BC cím, címke
IFNCI	Ugrás, ha keret tiltás = 0	BC cím, címke
IFCD	Ugrás, ha keret LAM engedélyezve	BC cím, címke
IFNCD	Ugrás, ha keret LAM felfüggesztve	BC cím, címke
IFGL	Ugrás, ha keret LAM kérés van	BC cím, címke
IFNGL	Ugrás, ha keret LAM kérés nincs	BC cím, címke
RCGL	Keret LAM kérés minta olvasás	BC cím, belső referencia
RCS	Keret állapotszó olvasás	BC cím, belső referencia

5.4. táblázat

Csoportvezérlőre vonatkozó utasítások

MNEMO-kód	Működés	Paraméter
BZ	Csoportiniciálás (BZ)	B cím
ENBD	Rendszer (csoport) kérés engedélyezése	B cím
DISBD	Rendszer (csoport) kérés felfüggesztés	B cím
ENBX	X-hiba detektálás engedélyezés	B cím
DISBX	X-hiba detektálás tiltás	B cím
RBGL	Rendszer kérés minta olvasás (GL minta)	B cím, belső referencia
RBQ	Q állapot beolvasása	B cím, VL
IFBQ	Ugrás, ha Q = 1	B cím, címke
IFB NQ	Ugrás, ha Q = 0	B cím, címke
RBCR	Csoportvezérlő regiszter tartalmának kiolvasása	B cím, külső referencia, belső referencia
WBCR	Csoportvezérlő regiszterbe írás	B cím, külső referencia, belső referencia

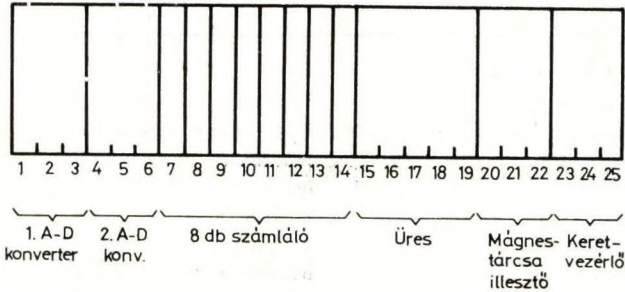
5.1.3.

Példaprogram IML nyelven

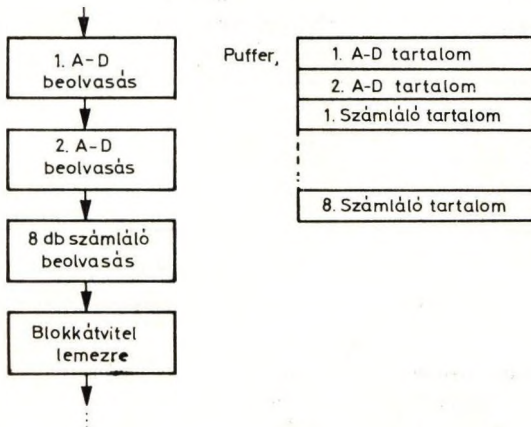
A következőkben bemutatunk egy IML nyelven írt programrészletet.

A program által kezelt hardware egységek a következők (5.3. ábra):

- 2 db analóg—digitális átalakító,
- 8 db számláló,
- 1 db mágneslemez-vezérlő és mágneslemez.



5.3. ábra
IML példaprogram hardware moduljai



5.4. ábra
IML példaprogram vázlata és puffertároló beosztása

A programrészlet órajel vagy külső interrupt (esetleg LAM jel) hatására indul, folyamatábrája az 5.4. ábrán látható. A program minden indítás után indítja és beolvassa az 1., majd a 2. analóg—digitális átalakító által mért értéket, ezután a 8 egymás mellett levő számláló tartalmát olvassa be. A beolvasást folyamatosan egy PUFFER címkéjű tárolótömbbe végzi. Amikor a tárolótömb tele van, tartalmát blokkátvitellel mágneslemez tárolóba viszi át. Az itt következő programrészlet nem tartalmazza a csatornák előkészítését, a hibarutinokat és a különleges vizsgálatokat, mint például állapotvizsgálat!

/KÜLSŐ DEKLARÁCIÓK

LOCD AD1, H, <1,1,3,0>; /1. A—D átalakító

LOCD AD2, H, <1,1,6,0>; /2. A—D átalakító

LOCD SZÁMLÁLÓK, Q, <1,1,7,0> <1,1,14,0>;

/számlálók deklarációja Q letapogatási mezőben

LOCD DISZK, H, <1,1,22,0>;

/CSATORNA DEKLARÁCIÓK

LOCC CHAN1, 8, COUNTER1, CSTAT1; /Q letapogatást vezérlő csatorna

LOCC CHAN2, 10, COUNTER2, CSTAT2; /Diszket vezérlő csatorna

/BELSŐ DEKLARÁCIÓK

LOCM PUF1, B, PUFFER, 16, 10, TELE; /PUF1 néven töltve a puffert, ha tele van, TELE címkére ugrik a program

... /A PROGRAM UTASÍTÁSAI

START1, SA F25, AD1; /Az 1. A—D átalakító mérés indítása

SA RD2, AD1, PUF1; /A mért eredmény beolvasása F1 funkcióval

SA F25, AD2; /A 2. A—D átalakító mérés indítása

SA RD2, AD2, PUF1; /A mért eredmény beolvasása F1 funkcióval

MAD RD1, SZÁMLÁLÓK, PUF1, CHAN1; /A számlálók tartalmának beolvasása F0 funkcióval

...

TELE, UBC WT1, DISZK, PUF1, CHAN2; /A teljes blokk kiírása diszkre

5.1.4.

Követelmények a makrofordítóval szemben

Amint az előző példában láttuk, az IML nyelv használata a programozó számára igen kényelmes és a gépi programmal azonos hatásfokkal futó programot kapunk. A teljes IML nyelv makrókkal való megvalósítása azonban igen magas szintű makrotranszlátorral, ill. makroprocesszorral lehetséges. A fordítás elég nagy memóriahely- és időigényes. (Az egyik megvalósított PDP—11 implementációnál 12—16k ferrittárolót és diszk háttértárat használnak.)

A továbbiakban bemutatjuk a makro nyelvű programozás lényegét (további irodalom [3]-ben található), majd az IML-ből vett példa programrészlettel szemléltetjük az általános elveket. A jelölésmódban a PDP—11 makro nyelvének konvencióit használjuk.

Makroprogramozás

Az assembly szintű programozásnál gyakran kényelmes, ha valamely ismétlődő utasítás-sorozatot egyetlen állítással, utasítással generálhatunk. Ahhoz, hogy ezt meg tudjuk tenni, az utasítás-sorozatot előbb látszatparaméterekkel (formális paraméterekkel) kell definiálni. Ezt nevezzük *makrodefiniálásnak*. Ha egyszer ily módon definiáltunk egy makrót, a programban levő egyetlen utasítással a *makrót* a neve segítségével *előhívhatjuk*, és ez a hívás az előre definiált utasítás-sorozatot, azaz a makrokifejtést a programba beépíti. Amikor a makrót előhívjuk, akkor természetesen a valódi paramétereket (más néven aktuális paraméter) is meg kell adni, és a programba ténylegesen beépülő utasítás-sorozat a látszatparaméterek helyett ezeket a valódi paramétereket fogja tartalmazni. Hangsúlyozzuk, hogy a makróknak a *programba való beépítése fordításkor* történik, a program futásakor már nincs nyoma a makrók használatának.

A makrók használata kicsit hasonló a változók szimbólumokkal való jelöléséhez. Fordításkor az assembler minden szimbólumot a hozzá tartozó értékével együtt elhelyez egy szimbólumtáblába, és amikor az illető szimbólum a programban előfordul, a táblából kikeresi a hozzá tartozó értéket és azt helyettesíti a programba. Ha egy makrót definiálunk, az assembler a makro névvel együtt a hozzá tartozó utasítás-

sorozatát elhelyezi egy makrotáblába. Amikor a makro név a programban előfordul (azaz amikor a makrót hívjuk), a fordítóprogram a makrotáblából kikeresi a hozzá tartozó utasítást, és azt helyezi el a programban.

Fejlettebb makrofordítóknál a makro definiálással meghatározott, és így a makrotáblába bekerülő utasítás nem csak a program futása közben végrehajtandó utasításokat tartalmazhatja, hanem a fordítóprogramnak szóló direktívákat, *fordításkor* végrehajtandó utasításokat is.

Ezek közül a leggyakoribb a *feltételes fordítás*. Legyen például egy utasítás a következő:

```
.Y=0  
.IF EQ, A-110  
.EXT=1  
.  
.  
.  
.ENDC
```

Az *.IF...*-től az *.ENDC*-ig terjedő utasítást az assembler csak akkor dolgozza fel, ha az *IF*-ben jelölt feltétel igaz, azaz a példában *A=110*-zel.

Makrodefiniálásra példa a következő:

```
.MACRO LOCD, MAC, A, B, C  
.  
.  
.  
.ENDM
```

A definíció mindig *.MACRO*-val kezdődik és *.ENDM*-mel végződik. (Más assemblerek más jelölésmódot is alkalmazhatnak!) A példában *LOCD* a makro neve, a formális paraméterek pedig *MAC, A, B, C*. A makro hívása a programban a nevével történik. Például:

```
LOCD AD1, H, <1, 1, 3, 0>
```

Híváskor a fordító a tényleges paramétereket helyettesíti be a programba, tehát a példában *MAC* helyére *AD1*-et, „*A*” helyére „*H*”-t, „*B*” helyére *<1, 1, 3, 0>*-t. Figyeljünk meg a példában két érdekességet: a „*C*” formális paraméterhez a híváskor nem tartozik aktuális paraméter; a csúcsos zárójellel összefogott paraméterek egyetlen formális paraméterhez tartoznak.

A makrodefiniálás is a fordítóprogramnak szóló direktíva. Ezért egy *makróban* definiálni lehet egy másik *makrót*. Például gondoljuk végig a következő makrodefiniációt:

```
.MACRO LOCD, MAC, A, B, C  
.MACRO MAC  
.Y=0  
.IF EQ, A-110  
.  
.  
.  
.ENDM  
.ENDM
```

Amikor a fenti makrót a programban definiáltuk, akkor ez az utasítássor LOCD címen bekerült a fordítóprogram makrotáblájába.

A program további részében hívjuk a makrót aktuális paramétereivel:

```
LOCD AD1, H, <1, 1, 3, 0>
```

Ennek hatására még fordításkor a LOCD néven definiált utasítássor előhívódik. Ennek első utasítása (.MACRO MAC) a fordítóprogramnak szóló direktíva, amely egy további makrót definiál. Ennek neve — az aktuális „AD1” paramétert a formális „MAC” paraméter helyére helyettesítve: AD1. AD1 néven tehát a makrotáblába újabb makro kerül, ahol viszont az előző A és B formális paraméterek helyébe a H, <1, 1, 3, 0> aktuális paraméterek kerülnek.

A LOCD makrónak az a további érdekessége, hogy a futásra kész lefordított programban nem szerepel neki megfelelő gépi utasítás, ugyanis a makro csak a fordítóprogramnak szóló direktívákat tartalmazott.

IML megvalósítása makrók formájában

Az IML nyelv makro változatának használatához ezek szerint két feltétel szükséges:

1. Magas szintű makrotulajdonságokkal rendelkező fordítóprogram (assembler vagy makro processzor).

2. Az IML nyelv állításainak megfelelő makrók definiálása és könyvtárba gyűjtése az adott számítógép és CAMAC vezérlőutasításainak figyelembevételével.

Ha ezek a feltételek megvannak, az IML makroszintaxisának megfelelő utasításokat írhatunk az assembly nyelvű utasítások közé. Ezek voltaképpen makrohívások; a hívott makrók az előre elkészített makrokönyvtárból kerülnek végrehajtásra.

Az 5.5. táblázat részleteket mutat be egy PDP—11-re megvalósított, az IML állításait makroként definiáló programból. A táblázat alján a definiált makrókat használó példaprogram látható.

A külső referenciát deklaráló LOCD nevű makro definíciója látható a program felső részében. Ez a makro nem tartalmaz a futás közben végrehajtandó utasításokat, hanem egy további makrót definiál. A példaprogram LOCD állítása után hívja ezt a makrót, és AD1 néven a H és <1,1,3,0> paraméterekkel ellátva egy újabb makrót helyez el a makrotáblába.

A MAC nevű makro elején feltételes utasítást látunk. Ha „A” értéke a H betűnek felel meg (110), akkor az EXT nevű változó értéke legyen 1 és kerüljön végrehajtásra a BCNA nevű makro. (Ez állítja össze a CAMAC címet a vezérlő számára.) Ebben az esetben egyetlen CAMAC cím a külső referencia.

A táblázatban továbbá részletet látunk az SA egyszeres akciót jelentő IML utasítást megvalósító makrodefinícióból. A makro első utasítása az EXT látszólagos paraméterrel mint névvel hívott makro. Ennek a paraméternek a helyén a LOCD deklarációval meghatározott külső paraméter áll: a példaprogramban AD1. A LOCD makro — mint az előbb láttuk — egy AD1 nevű makrót generált: ezt az AD1 makrót hívja az SA makro első utasítása. Ez a makro összeállítja a tényleges CAMAC címet, a vezérlőnek szükséges formátumnak megfelelően. Ezután feltételes utasítás következik: mivel egyszeres akcióról van szó, a program megvizsgálja, hogy valóban csak egy CAMAC címet deklarálunk-e. Mindezek a vizsgálatok, címösszeállítás a fordításkor történik. A futó programba csak a MOVE... utasítás kerül, amely a CAMAC vezérlő megfelelő regiszterébe betölti az utasítást és indítja a CAMAC ciklust. (Más vezérlőknél természetesen több MOVE-ra is szükség lehet, ha több segédregisztert kell a vezérlőbe tölteni.)

A makro translátorral szembeni követelmények röviden a következők:

1. A makrodefiníciókat lehessen egymásba helyezni, azaz lehessen egy makrót definiálni egy másik makro belsejében.

2. A makrohívásokat lehessen egymásba helyezni, azaz lehessen egy makrót hívni egy másik makro belsejében.

3. Egy makro név lehessen egy makroutasításban megadott paraméter.

4. A fordítás lehessen feltételes egy makroparaméter numerikus értékétől függően.
5. Ugyanazt az azonosítót lehessen használni makro névként és változó, ill. címke nevéként. A fordítónak meg kell tudnia különböztetni őket, például úgy, hogy az operátor és az operandus neveket külön szimbólumtáblában tartja, vagy úgy, hogy a makrót „zárójelbe teszi”, azaz lehetővé teszi, hogy a makrón belüli neveket megváltoztassuk vagy pedig úgy, hogy megvan az a lehetőség, hogy egy adott azonosító újradefiniálását megtiltsuk vagy engedélyezzük.
6. A transzlátor engedje meg, hogy a makrót a formális definícióban adottnál kevesebb paraméterrel hívhassuk.
7. A makrót a nevének leírásával lehessen hívni. A nevet egy vagy több betűköz vagy tabulátor kövesse, ezután következzen a paraméterlista (ha van ilyen), amelyben a paramétereket vesszők választják el egymástól.
8. A felhasználó definiálhasson legalább 5 karakterből álló azonosítókat.

5.2.

CAMAC programozás egyszerű assembler szinten

Az előzőekben láttuk, hogy a teljes IML nyelv makrókkal való megvalósítása elég bonyolult makro assemblert vagy makroprocesszort igényel, amellyel viszonylag kevés kisszámítógép rendelkezik.

A gépi nyelven való CAMAC programozás bonyolultsága miatt viszont már régóta készítenek egyszerű szubrutin- vagy makrokészleteket az egyszerűbb IML utasítások könnyű programozhatósága érdekében.

A legegyszerűbb megoldás szerint az assembler szintű utasításkészletet kiegészítik állandó makrohívó utasításokkal, amelyek a CAMAC perifériakészletet mozgatják [1]. Az előre definiált makrók be vannak építve a módosított assembler-be, a programozónak nem áll módjában, hogy újabb makrókat definiálhasson. Bonyolultabb megoldásnál a programozó bővítheti is makrokészletét újabb definíciókkal [2].

Példaképpen tekintsünk át egy igen egyszerű, de jól használható assembler kiegészítést.

A TPA-i kisszámítógép assembler-e nem rendelkezik korszerű makrolehetőségekkel, ezért a SLANG assembler-t egészítették ki a CAMAC modulok mozgatását végző makroutasításokkal: ennek neve CAMAS. Ezek a makrók be vannak építve az assembler-be, így a programozó nem tud újabbakat definiálni. A makrók két nagy csoportba oszthatók: CAMAC címeket lehet deklarálni és a CAMAC hardware-nek adható mozgató, adatátvivő utasítások.

Az IML-be leírt *deklarációk* közül ez az egyszerű változat a legfontosabbat, a *külső referenciát* (CAMAC cím) valósítja meg. Mivel az akcióutasítások között csak az egyszeres akciók szerepelnek, ezért a CAMAC címekből álló tömbök deklarációja elmarad. Az egyszerűség kedvéért csak a leggyakrabban használt, indirekt, pointer-rel használt deklaráció (P típusú) használható. Formája:

AZONOSÍTÓ: C N A

ill. példaképpeni értékkel helyettesítve:

TIMER: 2 3 0

Ez azt jelenti, hogy a TIMER nevű azonosító tartalmazni fogja a 2. keret 3. állomáshely 0. alcímét olyan formátumban előkészítve, ahogy azt az utasításmakrók felhasználni tudják. A TIMER nevű memóriaelem tartalmát a program futása közben assembler utasításokkal meg lehet változtatni. Programszerkezetiileg ez a megoldás abban különbözik az IML-ben leírttól, hogy míg ott a deklaráció makrók újabb makrókat hoztak létre a makrotáblában, ennél az egyszerű megoldásnál — makrotábla hiányában — az assembler-ben egyébként is meglévő szimbólumtáblába kerülnek a deklarált azonosítók. Azonos viszont mindkét megoldás esetében az, hogy egyik sem hoz létre olyan utasításokat,

amelyek a futó programba is bekerülnek. A külső referencia deklarálás ennél az egyszerű megoldásnál is ellenőrzi, hogy a CAMAC címek az érvényességi tartományon belül vannak-e. A fordítás végén megjelenő hibalista ezeket a hibákat is tartalmazza.

A hardware-nek adható mozgó, adatátvivő utasítások közül az egyszeres *akció* van megvalósítva. Formája az olvasás, írás és vezérlés típusú utasításoknak megfelelően:

& READ, D Q (Fn, EXT, INT)
& WRITE, D Q (Fn, EXT, INT)
& CTRL, Q (Fn, EXT)

D jelöli azt, hogy 24-bit-es szót kell átvinni; ha nem szerepel, akkor a számítógépnek megfelelő 12 bit kerül átvitelre.

Q jelöli a Q vizsgálatlól függő utasításátlépést. Ha szerepel Q a paraméterek között és a CAMAC rendszerből érkező Q válasz 1, akkor a soron következő utasítást a program átugorja.

Fn a megfelelő funkciókódot tartalmazza. Olvasás esetében természetesen csak F0—F7 kódok érvényesek. Az EXT paraméter a CAMAC címet tartalmazó memóriarekesz azonosítója, amelyet például a fenti, külső deklarációval adtunk meg. INT az átvendő adatot tartalmazó tárolórekesz címe.

5.3.

CAMAC programozás magasabb szintű nyelven

Mivel a CAMAC egységek nem különböznek alapvetően a többi real-time perifériáktól, a modulokat vezérlő vagy meghajtó (driver) programok tetszőleges real-time nyelvbe beilleszthetők. A CAMAC hardware rendszer géptől független, ezért célszerűnek látszik, hogy a géptől és gyártó cégtől független, nemzetközileg elfogadott real-time nyelvekbe kerüljenek beépítésre a CAMAC-ot kezelő programrészek.

A probléma az, hogy egységes, szélesebb körben elfogadott real-time nyelvek 1974. végéig nem alakult ki sem általános gyakorlat, sem szabvány, és nincs is remény arra, hogy egy-két éven belül ilyen szabvány kialakuljon.

A nemzetközi szabványosító munka 1968. óta folyik Prof. T. J. Williams vezetésével a „Purdue Workshop on Standardization of Industrial Computer Languages” elnevezésű munkacsoportban. 1973-ig nagyon kevés előrehaladás történt. 1973. novemberben újjászerveződött, kibővült a munkacsoport, és meggyorsult a munka. Vezetője továbbra is Prof. Williams, neve „International Purdue Workshop on Industrial Computer Systems”. 8 bizottsággal működik; ezek a nyelveken túlmenő problémákkal is foglalkoznak, mint például biztonsági, titkosítási kérdések stb. Real-time nyelvekkel a következő bizottságok foglalkoznak: FORTRAN, BASIC, Long Term Procedural Language (LTPL), Problem Oriented Language. A legtöbb bizottságnak van USA-beli, európai és japán ága.

Az európai LTPL bizottság a hosszútávon is használható nyelvvel foglalkozik. Jelenleg a kiindulásként szóba jöhető nyelvek összehasonlítását végzi. Ezek a következők (a származási országot feltüntetéseivel):

- CORAL 66,RTL/1,RTL/2 (Anglia).
- LAI, PROCOL (Franciaország).
- PAS/1, PEARL (NSZK).

Az USA-beli bizottság eddig megvizsgálta a következő nyelveket: RTL/1, LAI, SINTRAN, JOVIAL, SLP/76, ALGOL 60, PL/1, „Case RTL”, LISP, SNOBOL, GPSS, SIMSCRIPT.

A magasabb szintű programnyelven való CAMAC programozás igénye két, eléggé szétválasztható területen merült fel: az első a viszonylag egyszerű, nem nagy sebességet igénylő alkalmazások, a második a bonyolult, a korszerű real-time technika finomságait, sebességét kihasználó rendszerek.

Az egyszerű alkalmazások közül a legáltalánosabb a modulok, rendszerek tesztelése. Ezeknél sebességi probléma rendszerint nincs, kedvező viszont, ha a kezelő a programmal és a hardware rendszerrel egyszerű módon „beszélgetni” tud, tehát a nyelv interaktív. A programozási nyelv leggyakrabban a BASIC valamelyik változata. Régebb-

ben FOCAL-t is használtak [4]. Sok helyen az elsősorban tesztelésre szánt egyszerű programnyelveket használják mérések vezérlésére, mérési adatgyűjtésre is.

Nagyobb rendszerek programozására CAMAC, ill. real-time lehetőségekkel kiegészített FORTRAN-t, ill. BASIC-et használnak. A futásiidő rövidítése miatt ezek a BASIC fordítóprogramok nem interpretáló típusúak, hanem a forrásnyelvű programot először lefordítják és ezután futtatják.

5.3.1.

BASIC nyelvű implementáció

A BASIC nyelv — megfelelő kiegészítésekkel — igen alkalmas folyamatszabályozási célokra [8]. Két fő előnye más nyelvekkel szemben az, hogy könnyen megtanulható és kevesebb tárolót igényel. (Például az egyik megvalósításban FORTRAN program 8K tárolója helyett 3,2 k tároló.) Folyamatszabályozási célokra való alkalmasságát, a szükséges kiegészítéseket jelenleg (1974 vége) az USA szabványügyi hivatala tanulmányozza. Addig, amíg végleges eredmény erről nem születik, CAMAC software szabványokat sem dolgoznak ki a BASIC magas szintű folyamatszabályozási alkalmazásaira.

Az egyszerű, elsősorban tesztelési célt szolgáló BASIC kiegészítésre már készült *előzetes ajánlás*. Ennek lényege az, hogy a BASIC nyelv a CAMAC perifériákkal *assembly nyelv szintű szubrutinokkal* tart kapcsolatot. A BASIC nyelvet a CALL utasítással egészítik ki, amely megfelelő assembly nyelven írt rutinokkal elvégzi a szükséges adatátvitelt vagy CAMAC vezérlést.

Az ajánlás szerint a csoport- és keretcímeket a BASIC-ben szokásos módon, előre meg kell adni a rendszernek, például:

```
INPUT B
INPUT C
```

utasításokkal a vezérlő írógépen keresztül vagy valamilyen más módon. Az ajánlott utasításokat az 5.6. táblázat tartalmazza. A CR és a CW nevű rutinok az olvasást, ill

5.6. táblázat

BASIC nyelv kiegészítése CAMAC-ot vezérlő utasításokkal

Modulra vonatkozó utasítások:

```
CALL CR (F, N, A, D)
CALL CW (F, N, A, D)
CALL CO (F, N, A)
```

A Q és X válasz az utasítás végén automatikusan beállítja a Q és X nevű változókat!

Keretre vonatkozó utasítások:

```
CALL Z
CALL C
CALL I
CALL P
```

írást vezérlik. Az F, N, A a CAMAC modul funkció-, állomáshely és alcím-kódja, D a tárolóhelyet jelöli. A CO nevű rutin CAMAC vezérlést lát el (CAMAC Operation), ennél adatátvitel nem történik. A Q és X vizsgálatra nincs külön utasítás. Az előző

rutinok ugyanis a rutin végén automatikusan beállítják a Q és X nevű változókat a CAMAC hardware-ből kapott értékek megfelelően. Ezt azután a BASIC nyelv utasításaival lehet már kezelni.

A Z, C, I és P nevű rutinok a keretre vonatkozó általános vezérlőutasítások. Z a keretet alaphelyzetbe hozza, C törli, I (Inhibit) bekapcsolja a Tiltás jelet és P (Permit) kikapcsolja.

A LAM kezelés lehetősége opcionális. Legegyszerűbb esetben minden LAM kérésre a program egy előre adott sorszámú utasításra ugrik, például a 90 000 címkéjű utasításra. Az ezen a helyen levő program végzi el az esetleges LAM forrás keresést. Bonyolultabb megoldásnál minden Osztályozott LAM (GL) számnak megfelel egy-egy utasításcímke.

Példaképpen tekintsük át a következő feladat BASIC nyelvű programját! 10 db két egységnyi széles, 4—4 számlálót tartalmazó modul számlálóregisztereinek tartalmát kell kiolvasni, az eredményeket nyomtassuk ki írógépen. Közben adjuk össze az egyes számlálók tartalmát és a kiolvasás végén nyomtassuk ki ezt az összeget!

```
5 INPUT B
7 INPUT C
10 LET S=0
20 FOR N=1 TO 20 STEP 2
30 FOR A=1 TO 4
40 CALL CR (0, N, A, D)
50 IF Q <> 1 THEN 80
60 PRINT „SCALER”, N, „”, „”, A, „=”, D
70 LET S=S+D
80 NEXT A
90 NEXT N
100 PRINT „TOTAL=”, S
```

A fenti megoldás két szempont miatt alakult ki. Egyrészt a CAMAC-tól függetlenül perifériakezelő kiegészítések előszeretettel használják ezt a megoldást. A másik ok az, hogy ennek az implementációja a legegyszerűbb.

A használatban levő BASIC programrendszerek igen különbözőek, így nem mindegyik rendelkezik a fenti CALL hívással. Ezek számára vagy át kell alakítani a BASIC-et, vagy pedig egy preprocesszor programot kell készíteni. A preprocesszornak az a feladata, hogy a CAMAC-kal kiegészített, ajánlott BASIC nyelven írt programot átalakítsa, átírja olyan formájúvá, amelyet az adott BASIC már megért.

A fenti megoldáson kívül még sokféleképpen egészítették ki a BASIC nyelvet. Ezek a kiegészítések 4 csoportba oszthatók:

- új változók bevezetése;
- új funkciók bevezetése;
- új állítások, utasítástípusok bevezetése;
- új szubrutinok bevezetése.

Mindegyik megoldásra bemutatunk egy-egy példát:

1. LET %SWR=STA (1, 1, 20, 0); az új típusú változó itt CAMAC cím. Jele: %.
2. LET X=CCR (0, 0); az új funkció: CCR, és a 0, 0 című modul tartalmát jelenti.
3. RD1 SWREG, X; az új utasítás ebben a példában RD1, azaz kiolvasás.
4. Új szubrutin bevezetésére részletes példát már előbb láttunk.

5.3.2.

FORTRAN-ba épített szubrutinok

Az egységes, jól használható real-time nyelv hiánya miatt már a hatvanas évek végén elkészítették a FORTRAN nyelv kibővítését, hogy használhatóvá tegyék folyamat-szabályozási célokra. 1972-ben ISA (Instrument Society of America) szabvány is (ISA—S61.1 1972) készült a kibővített FORTRAN-ról. A bővítés három irányú: a vezérlőprogrammal (executive) való kapcsolatot időzítő, feladatot ismétlődően indító stb. utasításokkal bővítették, kialakították a folyamatperifériákat mozgó utasítás-készletet és az általános utasításkészletet bővítették bit és string műveletekkel.

Több helyen, főleg az Egyesült Államokban, CAMAC utasításokkal egészítették ki az ISA által szabványosított FORTRAN nyelvet. Ezek a CAMAC kiegészítések azonban még nem váltak ajánlásokká vagy szabványokká.

A kiegészítést mindenütt szubrutin formájában illesztették be a FORTRAN-ba. Az egyes megvalósítások abban különböztek, hogy hány és milyen szubrutint használnak, és hogy a paramétereket hogyan adják meg.

A legjobban elterjedt FORTRAN kiegészítést a Los Alamos-i laboratóriumban készítették [7]. Ennek a megoldásnak az a lényege, hogy igen összetett szubrutint használ, nagyon sok paraméterrel.

Példaképpen bemutatunk néhány szubrutint. A CMCBSC nevű szubrutin feladata egy CAMAC funkció végrehajtása egy CAMAC címen, egy vagy több alkalommal. Hívása

```
CALL CMCBSC (F, B, C, N, AD, LN, DATA, Q, ERRORA)
```

utasítással történik. A paraméterek jelentése a következő:

F	CAMAC funkciókód
B	csoportszám
C	keretszám
N	állomászám
AD	alcím
LN	ismétlési szám és szóhossz
DATA	tárolócím
Q	Q válasz
ERRORA	hibavektor.

A CMCLUP nevű szubrutin egy CAMAC funkciót hajt végre egy CAMAC címekből álló hierarchikus mezőn, a CAMAC címekből álló mezőből azonban a Q választól függően bizonyos részeket kihagy. Hívása:

```
CALL CMCLUP (F, BA, LB, CA, LC, NA, LKN, ADA, LAD, LN, DATA, Q, QON,  
ERRORA, NEX);
```

A paraméterek közül a BA-tól LAD-ig levők a CAMAC címmező kezdetét és hosszát adják meg. LN, DATA, Q és ERRORA megegyeznek az előző szubrutinnál használt paraméterekkel. QON jelzi azt, hogy Q-tól függően kell-e CAMAC címet átugrani. NEX változó tárolja azt, hogy hányszor került végrehajtásra a kívánt funkció.

A példából is látható, hogy egyszerű esetekben is (az első példa a legegyszerűbb szubrutin!) igen sok paraméterrel kell dolgozni. Egy másik megoldás [5] azt javasolja, hogy egyszerűbb szubrutinokat kell használni.

Egy egyszerűbb FORTRAN szubrutinra példa a következő:

```
CALL READ (F, EXT, INT, COND)
```

Az F, EXT és INT paraméterek funkciókódot, CAMAC címet és memóriacímet tartalmaznak, a COND a feltételeket: például utasításátlépés a Q választól függően.

5.3.3.

CAMAC programozás problémaorientáltan

Az eddig leírt software rendszerek közös tulajdonsága az, hogy a programozónak nem kell ismernie az alkalmazott gép alacsony szintű nyelvét, sem a CAMAC rendszert illesztő vezérlőegységet. Ismernie kell viszont a CAMAC rendszerben levő modulok helyét a rendszerben, továbbá a modulok F funkciókódjait. E programrendszerek előnye az, hogy a felhasználó a modulokban adott minden lehetőséget ki tud használni, hátránya viszont, hogy a leggyakoribb, általános alkalmazásoknál is ismerni kell a hardware modulokat.

E hátrány kiküszöbölésére több helyen készültek olyan programrendszerek, amelyekben a CAMAC modulok programozása nem CAMAC, tehát ebben a vonatkozásban gépi szinten, hanem a feladat, a probléma felől megalapozva lehetséges.

Problémaorientált CAMAC programozásnál a rendszertervező határozza meg a feladathoz szükséges hardware modulokat és ezek rögzített elhelyezését a keretekben, természetesen modulárisan, ésszerű bővítési lehetőségeket figyelembe véve. Hasonló természetű feladatokhoz viszonylag könnyen lehet „szabványos” CAMAC elrendezést létrehozni: például összetartozó ipari analóg—digitális konverter—multiplexer, digitális bemenet—kimenet összeállításokat. A rendszertervező ezután egy-egy ilyen „szabványos” hardware elrendezéshez elkészíti a software handler-ek-et, azaz a hardware modulokat meghajtó programokat. Ezek a handler-ek valamely problémaorientált magasabb rendű nyelvbe épülnek be és a legtöbbször assembly szintű nyelven készülnek, így futási idejük rövid.

Az alkalmazói programok készítésekor a programozónak két feladata van. Egyrészt pontosan meghatározza a konfigurációt, azaz a rendszertervező által „szabványosított” konfigurációból kijelöli, hogy mely modulokat (vagy méginkább milyen funkciókat) használ és milyen mennyiségben. Ezt a meghatározást valamilyen deklaráció formájában rögzíti. Másrészt a program végrehajtandó utasításai során egyszerű rövidítéssel küldi ki vagy kéri be a szükséges adatokat.

A következőkben problémaorientált CAMAC periféria programozásra két példát mutatunk be: mindkét példa a DEC cég PDP típusú számítógépeire kidolgozott INDAC nyelvhez kapcsolódik.

Első példa: jelzőlámpák meghajtását végző modul vezérlése. Az alkalmazói programot készítő a deklarációban meg kell, hogy adja a periféria típusát (DGL: Digital Lamp), a meghajtandó lámpák számát és azt a tárolótömböt, amelynek tartalma határozza meg a lámpák bitképét:

```
.EQUIPMENT
*DGL
  CHAN (1,4) IFLOW
  CHAN (5,8) IERROR
```

A deklaráció jelentése: a DGL nevű eszköz 1...4. csatornája (egy csatornához 24 lámpa tartozik) az IFLOW nevű tárolótömbből, az 5...8. csatornája az IERROR nevű tárolótömbből kapja az adatait.

A végrehajtandó programrészben a működtető utasítás:

```
SEND (DGL) IFLOW
```

iii.

```
SEND (DGL) IERROR
```

Ezek a z utasítások az IFLOW, iii. IERROR tartalmát megjelenítik a lámpákon. Indexeléssel külön-külön is ki lehet gyújtani a lámpákat:

```
SEND (DGL) IERROR2
```

azt jelenti, hogy az IERROR tömb 2. tagjának tartalmát ki kell küldeni a 6. csatorna lámpáira.

Második példa: 20 db 24-bites logikai szó állapotát (pl. jelfogóérintkezők) kell beolvasni. A bejövő vezetékeken rövid idejű zavarjelek lehetnek, ezért több egymás után beolvasott értékből kell a valódi értéket megállapítani. Az alkalmazói program deklarációs része:

```
.EQUIPMENT
*DGC
CHAN (1, 20) ID
```

A bemenő periféria neve DGC (Digital Contact), a beolvasott értéket az ID-vel definiált tárolótömbbe kell helyezni. A

```
GET DGC ID
```

utasítás hatására megtörténik a 480 db digitális jel állapotának beolvasása, simítással. (A simítás algoritmus: ha a jelenlegi beolvasott minta C és az előző két beolvasott minta A és B, akkor $ID = AB + AC + BC$, bitenkénti művelettel értelmezve.) A példából látható, hogy a rendszertervező itt nemcsak a hardware-t „szabványosította”, hanem a simítás algoritmusát is!

Irodalom az 5. fejezethez

- [1] Almási L.—Buday L.—Görög P.: CAMAC real-time perifériák software problémái. Számítógéptechnika '74 konferencia előadásai 500...507. old., 1974.
- [2] Golding, F. R. et. al.: CAMACRO—An Aid To CAMAC Interface Programming. CAMAC Bulletin, No 5., p. 29...30., 1972.
- [3] Macro Processors and Techniques for Portable Software. J. Wiley, 1974.
- [4] May, F. et. al.: Focal Overlay for CAMAC Data and Command Handling. CAMAC Bulletin, No 1., p. 18., 1971.
- [5] Michelson, J.—Halling, H.: Procedure Calls — A Pragmatic Approach. Proc. of the First Int. Symp. on CAMAC, 1973., p. 63.
- [6] The Definition of IML. Oct. 1974. ESONE Committee ESONE/IML/01.
- [7] Thomas, R. F.: Specifications For Standard CAMAC Subroutines CAMAC Bulletin, No 6., p. 23., 1973.
- [8] Worster, L. D.: BASIC As a Process Control and Monitoring Computer Language. Douglas United Nuclear, Inc., Richland, Was. DUN—SA—185. 1972.

6.

ALKALMAZÁSOK

Amint az 1. fejezetben már láttuk, a CAMAC rendszereket először a nukleáris laboratóriumokban kezdték el használni a hatvanas évek második felében. A hetvenes években terjedtek el az újabb alkalmazási területeken, kémiai, fizikai, orvosi-biológiai laboratóriumokban és ipari feladatok megoldására. Az alkalmazási területeket elég nehéz szétválasztani: ugyanazt a feladatot ellátó és esetleg még ugyanolyan körülmények között működő rendszereket sokszor csak az elnevezés sorol különböző területekre. Így például egy gyorsító vagy ciklotron működésének vezérlése valójában ipari feladat, ipari környezettel (nagy kiterjedés, fokozott megbízhatóság, nagy környezeti zajszint stb.), azonban általában a nukleáris feladatok között tartják számon. Az alkalmazások csoportosítása a tárgyalás során ezért önkényes. A következő főbb csoportokban tárgyaljuk az alkalmazási területeket: nukleáris, fizikai- és kémiai laboratórium, orvosi-biológiai alkalmazások, ipari alkalmazások, egyéb és különleges alkalmazások.

6.1.

Nukleáris alkalmazások

Az alkalmazások közül a CAMAC eredetének megfelelően legszélesebb körű a nukleáris alkalmazás. A CAMAC a nukleáris mérés, vezérlés szinte minden területén elterjedt. Az igen gyorsan változó igényekhez könnyen igazodó CAMAC a nukleáris rendszerek telepítésénél jelentős, átlagosan kb. 30%-os anyagi megtakarítással és még ennél is nagyobb, kb. 60%-os időmegtakarítással jár [30]. A legtöbb egyszerű rendszer a „polcra elvett” modulokból azonnal összeállítható. Minden nagyobb kutatóközpont rendelkezik olyan CAMAC modulbankkal, ahonnan egy-egy rendszer összeállításához szükséges egységek kivehetők. Ezen túlmenően szinte minden kisgéphez megvan a CAMAC illesztőegység (keretvezérlő, csoportmehajtó) és valamilyen felhasználó-orientált CAMAC programkészlet (CAMAC makrókkal bővített assembler, CAMAC utasításokkal bővített FORTRAN, BASIC), így az aktuális program megírása sem vesz sok időt igénybe. Nagy rendszereknél — például gyorsító vezérlésnél, amelynek a méretei egy ipari rendszert gyakran meghaladnak — több hónapra, egy-két évre tehető az időnyereség. Ez egyrészt abból adódik, hogy külön-külön egyik kisgépre sincs olyan gazdag real-time perifériakészlet, mint CAMAC modulokból. Még a leg-

jobban ellátott kisgépre sincs 30...50-nél több, különféle real-time egység, míg CAMAC-nál, mint láttuk, kb. 1000 modultípust találunk, így szinte minden elképzelhető real-time funkciót közvetlenül, készen meg lehet vásárolni. Másrészt bonyolult rendszernél a hierarchia különböző szintjein a legmegfelelőbb géptípust lehet alkalmazni, és CAMAC-on keresztül ezek információcsereje is megoldható, így nem kell költséges gép—gép csatolóegységeket létrehozni.

A továbbiakban a teljesség igénye nélkül röviden a főbb nukleáris ágazatok jellegzetes CAMAC alkalmazásait foglaljuk össze, majd néhány tipikus példát részletesen is megvizsgálunk.

Nagyenergiás fizika. Proporcionális és magnetostrikiós huzalkamrák kiolvasó rendszere [1], [4], szikrakamrák felvételeinek automatizált kiértékelő rendszere.

Részecske- és magfizika. Egy- és többparaméteres amplitúdó-, időanalizátorok [23], [34], [26]. Diffraktométerek [6], spektrométerek [33], [19] mérésadatgyűjtő vezérlő-rendszere.

Aktivációs analízis. Nagy felbontóképességű spektrumanalizáló és kiértékelő rendszer, mintaváltó vezérléssel.

Reaktortechnika. Multiszámlálás típusú mérések, reaktor szimuláció hybrid rendszerrel, ahol az analóg gép és a digitális számítógép között CAMAC real-time rendszer teremt kapcsolatot [32], erőművi reaktorok sugárvédelmi rendszere.

Gyorsítóvezérlés. Igen elterjedt alkalmazás, vannak teljesen CAMAC-ra épített gyorsító-vezérlő rendszerek (CERN, Super Proton Synchrotron rendszere [7]) és vannak, amelyeknél csak a mérésadatgyűjtő rendszer és gyorsító egyes részeinek figyelése, vezérlése épült CAMAC rendszerben (USA, National Accelerator Laboratory: nyaláb monitor, booster vezérlés, mérésadatgyűjtés [14], [18], [2]. A rendszerben összesen kb. 200 keret, 145 soros keretvezérlő és mintegy 2000 modul található). Mint különlegességet említjük meg, hogy a TRIUMF laboratórium (USA) gyorsító-vezérlő rendszerében fénycsatolt soros CAMAC-ot alkalmaz [15] 300 kV potenciálon levő ionforrás vezérlésére. (Ez a rendszer kb. 30 keretet tartalmaz, 6 CAMAC CSO-PORT-ban.)

Számítógép-összeköttetések. A nukleáris mérésekben résztvevő számítógépek rendelkeznek CAMAC real-time egységekkel, így CAMAC—CAMAC összekapcsolással gépeket lehet összekapcsolni és akár egész intézetre kiterjedő géphálózatot lehet létrehozni. Ezen a téren úttörő munkát végzett a Daresbury Laboratórium [28]. A CAMAC-ra épülő géphálózatokra a 6.4. szakaszban még visszatérünk.

A nukleáris mérések szemléltetésére két példát mutatunk be, egy sokcsatornás analízátor- és háromtengelyű spektrométer-rendszert.

Sokcsatornás analízátor-rendszerek

A számítógéppel és CAMAC real-time perifériákkal megvalósított sokcsatornás amplitúdóanalizátorokat közelebről a SEN ELEKTRONIK 6000-es spektrumanalizáló rendszere [34] és a KFKI TPA-i — CAMAC amplitúdóanalizáló rendszere [13] alapján ismertetjük.

A feladat nukleáris, többnyire félvezető detektorból származó elektromos impulzusok amplitúdó szerinti osztályozása, majd a keletkező spektrumok kiértékelése. Elvégezhető mérések:

- egyparaméteres amplitúdóanalízis;
- kétparaméteres amplitúdóanalízis;
- multispektrum analízis.

Egyparaméteres amplitúdóanalízisnél egyetlen detektorból származó jeleket kell mérni és amplitúdó szerint válogatni meghatározott élő vagy teljes ideig, előírt csatornatartalomig. Kétparaméteres mérésnél két alest van. Egyik üzemmódban a két detektorból érkező jelet egymástól függetlenül mérjük, mintha két egymástól független egyparaméteres mérésünk futna egymással „párhuzamosan”. A másik



6.1. ábra
Sokcsatornás analízátor rendszer (TPA-i—CAMAC)

üzem módban a két detektor jelei közül csak azokat a párokat mérjük, amelyek a mérőrendszer bemenetére egyszerre (koincidenzában) érkeznek. Ebben az esetben mérés alatt vagy közvetlenül az adatpárokat tároljuk, vagy pedig a két paraméter által meghatározott mezőben az adatpárok előfordulási gyakoriságát regisztráljuk a hely függvényében (többdimenziós spektrum).

Multispektrum mérésénél az egyparaméteres spektrum időfüggését határozzuk meg oly módon, hogy egyparaméteres amplitúdóspektrumot mérünk előírt ideig, majd megismételjük a mérést többször egymás után, ugyanennyi ideig mérve, és a keletkező spektrumokat rendre eltároljuk a gép adatmezejének alszektoraiba.

Az analízátort képező hardware a két megoldásnál hasonló felépítésű, példaképpen a 6.1. ábrán a KFKI rendszer fotóját mutatjuk be. A felső szekrényben alul helyezkedik el a TPA-i számítógép, fölötte az oszcilloszkópos CAMAC megjelenítő, majd legfelül az aktív CAMAC keret. Az alsó, kisebb szekrényben a tárolóbővítés található.

A gépek (TPA-i, NOVA 2) 16K operatív tárolóval, bővített aritmetikával, real-time órával rendelkeznek, konzolirégékük: ASR—33, normál perifériák közül találunk lyukszalag-perifériákat, a SEN rendszernél kazettás mágnesszalagos egységet. (Természetesen mindkét rendszer tovább bővíthető hagyományos nagyperifériákkal: mágnesszalag, tárcsa, sornyomatató stb.).

A CAMAC keretbe az alábbi modulok kerülnek:

- 1 vagy 2 db analóg—digitális átalakító,
- 1 db időzítőszámoló;
- 1 db X—Y rajzolóvezérlő;
- oszcilloszkópos megjelenítő egységek, vezérlők.

Az analízátorok programkészlete mérésadatgyűjtő és kiértékelő programokat tartalmaz. A mérés indításánál, a mérésadatgyűjtő programban megadhatjuk a kívánt felbontást: 256 csatornától 4096-ig, a tárolásra kerülő spektrumhoz szükséges tárolószektor hosszát: 64-től 4096 csatornáig bináris lépésekben. A kettő nem feltétlenül egyezik meg, egy viszonylag nagy felbontású spektrumnak bármelyik kiválasztott részét tárolhatjuk. A kiválasztásra alkalmazhatunk digitális kivonást (0-tól 2048-ig beállítható) vagy digitális ablakot. Beállíthatjuk a mérési időt (élő vagy teljes időt) max. 2^{24} s-ig 1 s-os lépésekben, vagy a csatornatartalmat ugyancsak 2^{24} impulzusig. A mérés indítása után a megjelenítő egységen vizuálisan követhetjük a spektrum alakulását, kiírhatjuk fényponttal a megjelölt csatornák aktuális tartalmát, megkérdezhetjük a még hátra levő mérési időt, törölhetjük az adatmezőt. (A SEN megoldásában egyes gyakori műveletek, pl. indítás, törlés, megállítást nemcsak programmal, hanem nyomógombbal is kezdeményezhetők.)

A kiértékelő program a kész spektrumon végez műveleteket. Az adatmező, ahol a spektrumot tároljuk, osztható, a legkisebb megjeleníthető szektora 64, a legnagyobb 2048 csatorna. A megjelenítő egységen egyszerre két szektor rajzolható fel, amelyek között műveleteket végezhetünk (szektorok összeadása, kivonása, cseréje). A spektrumon két fénypont helyezhető el. A fényponttal megjelölt csatornák kiírathatók, a fénypontok közötti spektrumszakasz optimális méretre (X és Y irányban egyaránt) kinagyítható, integrálható. Meghatározható az egyes csúcsok helye, területe (háttérlevonással), félértékisége, energiafelbontása. A spektrumon végezhetünk energia- és amplitúdó-normálási műveletet, simítást és természetesen törölhetjük az adatmezőt (8k).

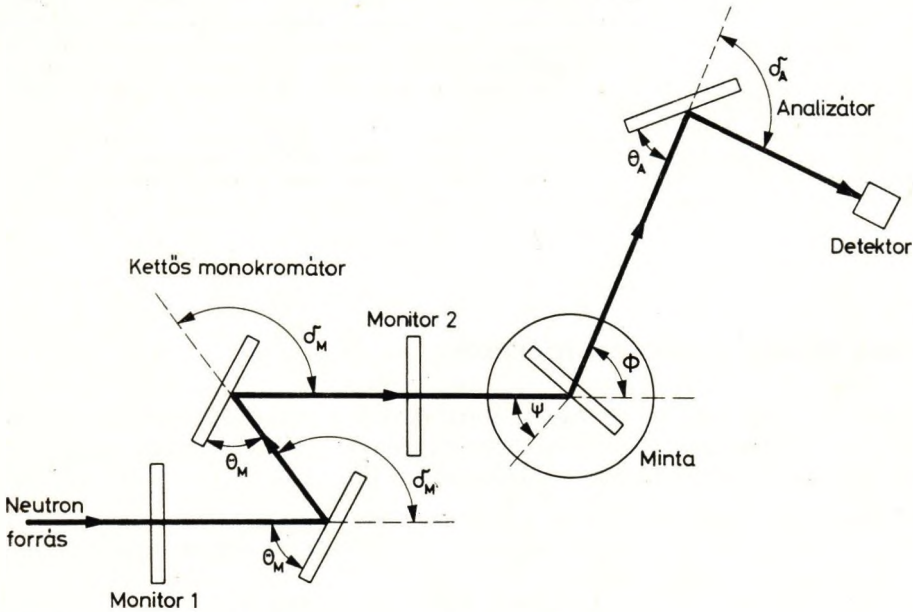
A programok a TPA-i gépre assembler-ben, a NOVA2-re a gép real-time operációs rendszerének (RTOS) felhasználásával készültek.

Háromtengelyű spektrométerek

Az atomok statikus és dinamikus viselkedésének vizsgálatára elterjedten alkalmazzák a háromtengelyű neutron spektrométereket [6, 9]. A spektrométer három főbb részből áll (6.2. ábra):

- monokromátor-rendszer;
- mintatartó;
- analízáló rendszer.

A monokromátor a reaktorból jövő különböző neutronok közül kiválaszt, kiszűr egy meghatározott energiájú nyalábot. A neutronnyaláb energiáját az állítható tükrök (kristályok) θ_M szögével lehet kiválasztani. A példaképpeni esetben a szűk energiasávot kettős monokromátor alkalmazásával érik el. A monoenergetikus neutronnyaláb ütközik az állítható (Ψ) szög alatt elhelyezett mintán és szóródik. Az analízáló rendszer azt nézi meg, hogy a mintából egy bizonyos szög alatt (φ) kilépő neutronoknak milyen az energiaeioszlása. Az analízálást egy további forgatható tükör (egykristály) végzi, amelyet a neutronok útjába tesznek valamilyen állítható beesési szöggel (θ_A) és a kristállyal szemben elhelyezett detektorban számlálják a beérkező, szelektált



6.2. ábra
A neutron útja háromtengelyű spektrométerben

energiájú neutronokat. A mérés a beeső neutronnyaláb útjába tett monitor detektorból (M2) érkező impulzusok előre beállított értékéig tart. Ha a monitor detektor ezt a beállított értéket elérte, az analízáló rendszer detektorát ki kell olvasni, és új szöghelyzetben kell folytatni a mérést. A többszörös verődés miatt a neutronhozam az analízáló detektorban igen kicsi, ezért egy teljes mérés igen hosszú, esetleg több hetes is lehet. A mérés alatt az új szöghelyzetek kiszámítása, az erre való optimális sebességű átállás vezérlése, a mérési feltételek (pl. mintahőmérséklet stb.) folyamatos ellenőrzése mind a számítógépes CAMAC-os rendszer feladata.

Az egyes tengelyek adott szögre történő beállítását léptetőmotorok végzik, a tengely aktuális szögét a tengelyre szerelt optikai szögtávadó (10 000 impulzus/fordulat) jelzi vissza. A beállított tengely helyzetét mágneses fékek rögzítik.

A számítógépes real-time rendszernek összesen 5 léptetőmotort kell vezérelnie, le kell olvasnia 5 optikai szögtávadót, figyelnie kell 12 véghelyzet-kapcsolót, vezérelnie kell 4 mágneses féket, számlálnia kell a 2 monitor detektorból és az esemény detektorból jövő jeleket.

A rendszert vezérlő számítógép PDP 8/I típusú DEC kisgép, 8K operatív tárolóval, katódugár-csöves megjelenítővel, konzolkiírógéppel és közvetlen csatlakozással egy SIEMENS 305-ös számítógéphez. CAMAC perifériák:

- U típusú keretvezérlő;
- 5 db léptetőmotor-vezérlő modul;
- 2 db megszakítást kérő modul (8-bemenetű);
- 2 db 24-bites számláló;
- 2 db 24-bites előre beállítható számláló;
- 1 db óra—impulzus generátor;
- 1 db digitális voltmérő;
- 1 db decimális kijelző.

A hardware összeállításból ki kell emelni a külön e célra kifejlesztett, 2 egységnyi széles léptetőmotor-vezérlő modult, amely egy léptetőmotorral és optikai távadóval kapcsolatos összes feladatot ellátja. (A modulba épített oszcillátor impulzusai fénycsatolt teljesítményfokozaton keresztül vezérlik a léptetőmotort, a mozgás irányának megfelelően az optikai távadó jelei töltenek vagy ürítenek egy számlálót. Ha ennek tartalma a modulban levő névérték-regiszterben tárolt értéket eléri, működésbe lép a mágneses fék és LAM jel keletkezik.)

A spektrométert működtető programot assembler-ben írták. A programozás megkönnyítésére a CAMAC üzenetek kezelésére rövid assembler szubrutinokat (CATR) készítettek. A rendszer üzembehelyezésénél felhasználtak egy CAMAC adat és igénykérelem kezelésére alkalmas FOCAL overlay (FOCADAT) programot, amelyet később interrupt kezelésére is alkalmassá tettek (FOCALINT) [21].

6.2.

Fizikai, kémiai analitikai mérések

Az analitikai berendezések az anyag, minta valamilyen specifikus tulajdonságának, szerkezeti felépítésének, állapotának, összetételének meghatározására, vizsgálatára alkalmasak. Tipikus berendezések a következők (6.20):

- Tömegspektrométerek (szerves, szervetlen).
- Spektrofotométerek (infravörös, ultraibolya, látható, Raman, emissziós stb.).
- Kromatográfok (gáz, folyadék).
- NMR (folyamatos és impulzusüzemű), ESR spektrométerek.

A berendezések nagy része egyetlen analóg jelet ad ki, amelyet mintavételes rögzíteni kell és a görbét ki kell értékelni. A görbe felvételével kapcsolatos fontosabb paramétereket röviden az alábbiakban foglalhatjuk össze:

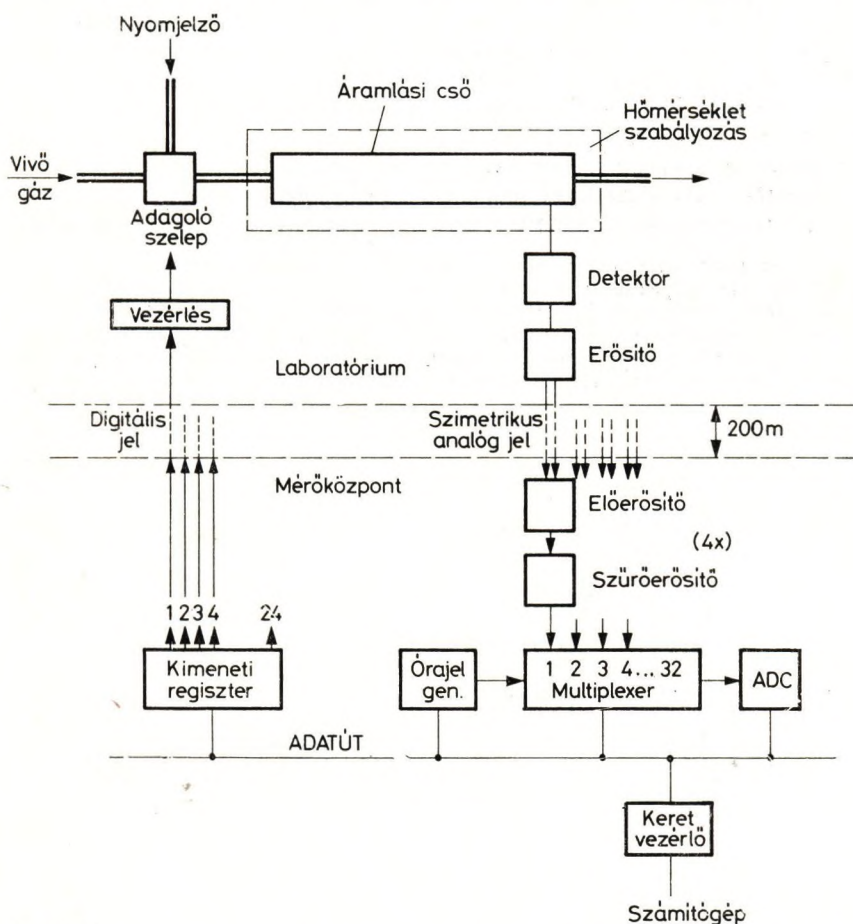
- dinamikus tartomány, felbontás;
- mintavételi frekvencia (sebesség);
- keletkező adatmennyiség (on-line redukció);
- vezérlés.

Gáz—folyadék kromatográfoknál, egyes tömegspektrométereknél a kiadott jel dinamikája gyakran eléri az $1:10^6$ -t, ami legalább két nagyságrenddel meghaladja a CAMAC és egyéb szokásos real-time analóg—digitális átalakítók dinamikus tartományát. A tartomány növelését vagy úgy érik el, hogy az A/D átalakító elé programozható erősítőt tesznek, vagy úgy érik el, hogy a jelet párhuzamosan több, fix erősítésű ($1\times$, $10\times$, $100\times$, ...) erősítőbe vezetik és ezek kimenetét multiplexeren keresztül kapcsolják az analóg—digitális átalakító bemenetére. Azt, hogy a mérés melyik erősítő kimenetén végezze, ill. milyen erősítésfokozatot kapcsoljanak be, az éppen mért amplitúdó nagyságából lehet eldönteni.

NMR, ESR spektrométereknél, egyes spektrofotométereknél (IR, Raman) gyakran jelentkező probléma, hogy a mérőberendezés felbontása elégséges ugyan, de a mérendő jel az elektromos zajok határán van. Ilyen esetekben a sorozatos mérés és valamilyen jel/zaj javító eljárás (átlagolás, ismételt összeadás) alkalmazása válik szükségessé.

A mintavételi frekvenciák két szélső határát egyfelől a nagy felbontóképességű tömegspektrométerek és az impulzusüzemű NMR spektrométerek, másfelől az igen lassú folyadék-kromatográfok jelentik. Az első csoportban a mintavételi frekvencia közel 50 kHz, és a keletkező adattömeg eléri a 10^6 szót. Itt tehát mérés közbeni, gyors adatredukciónak is kell végezni, hogy csak az értékes adatokat tároljuk. A második csoportban az igen lassú folyamat miatt (kb. 1 Hz-es mintavétel) igen nehéz feladat az effektus — pl. kromatográfoknál a hosszan elnyúló csúcsok leszálló ága — és az elektromos alapvonal-elcsúszások szétválasztása.

A vezérlési funkció általában a spektrumfelvétel indítására, megállítására, a mérési pontok kijelölésére, mintaváltásra terjed ki, de vannak pl. olyan tömegspektrométeres, NMR spektrométeres mérések, ahol nem kell végigvárni a teljes

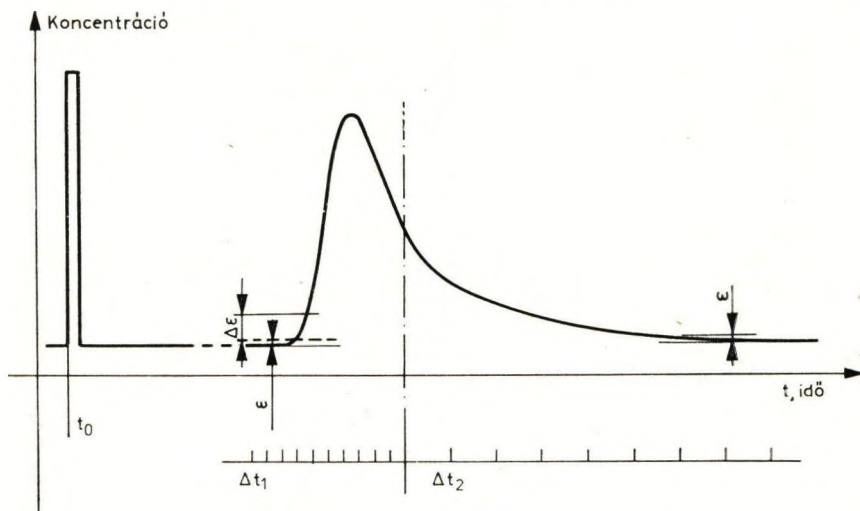


6.3. ábra
Anyagszállítási paraméterek meghatározására szolgáló mérőrendszer blokkvázlata

spektrumot, hanem a mágnes tér közvetlen beállításával (számítógéppel vezérelt digitális—analóg konverter felhasználásával) mindjárt az érdekes spektrumszakaszt választhatjuk ki. Vannak továbbá olyan összetett rendszerek: gázkromatográf-tömegspektrométer, kaskád kromatográf [10], ahol a vezérlési feladat sokkal bonyolultabb és az egyidejűleg mért paraméterek száma is több.

Korábban az ilyen jellegű berendezések kimenete valamilyen kirajzoló által készített diagram vagy filmfelvétel volt, később célhardware-ket építettek, amelyek bizonyos előre rögzített feltételek mellett automatikus spektrumfelvételt és kiértékelést végeztek. Jelenleg a számítástechnikai eszközök árának csökkenése lehetővé teszi, hogy a nagyobb, költségesebb mérőberendezésekbe mikro, mini számítógépet építsenek a vezérlési és adatredukciós feladatok ellátására. Az előkészített, redukált adatok végső kiértékelése azután valamilyen nagygépen történhet. Kisebb, olcsóbb berendezéseknél viszont célszerű, ha ezeket valamilyen szabványos kimenettel, bemenettel látjuk el és több ilyen berendezést kötünk egy közös számítógéphez. Találunk irodalmi példát [16]-ban, ahol a MODULAR ONE számítógéppel két NMR és két kromatográf kiszolgálását oldották meg 14 CAMAC egység felhasználásával (MULTILAB). A rendszer egykeretes, de lehetőség van egy második keret és második gép csatolására, ha a sebességi követelmények ezt indokolják. Egy másik rendszer PDP-8/1 gépet használ (LABCOM, [9]) 12 K-s operatív tárolóval, 32 K-s lemeztárolóval. A gépre jelenleg 3 készülék van kötve, két spektrofotométer (UV és IR) és egy ESR spektrométer. A real-time felügyelő program kezeli a programmegszakításokat, prioritási sorrend szerint osztályozza a programokat, vezérli a diszkes átvitelt. A különféle berendezésekhez megfelelő adatfeldolgozó programok vannak, például ESR spektrométerhez: ismételt alapvonal-visszaállító és integráló program.

Erre az alkalmazási körre példaként bemutatjuk egy gázkromatográfiás méréshez hasonló gázáramlási kísérlet [22] számítógépes automatizálását (6.3. ábra). A kísérletet áramlási csőben végezzük, amelynek hőmérséklete szabályozható, és amelyben valamilyen töltőanyag van. A csőn egyenesen áramlik át a vivőgáz mindaddig, amíg a stationárius állapot az adagolószелеp átkapcsolásával meg nem zavarjuk. Az adagolószелеp átkapcsolásával (mágneses vagy pneumatikus) rövid ideig tartó koncentrációugrást hozunk létre (Dirac-nyomjelző impulzus). Az ugrás a vivő gázzal együtt végighalad



6.4. ábra
Koncentráció—idő függvény

az áramköri csövön, közben a töltőanyagtól függően megváltoztatja az alakját, kiszélesedik, elnyúlik (6.4. ábra). Az áramlási cső végén elhelyezett detektorral ezt a koncentrációeloszlás-görbét tudjuk meghatározni. Az eloszlásgörbe momentumaiból (μ_0 : terület, μ_1 : súlypont, μ_2 : variancia, μ_3 : aszimmetria) pedig ki lehet számítani az anyagszállítási állandókat (tengely irányú diszperziós állandó, abszorbeáló és katalizáló anyagok diffúziós állandói stb.). A kísérlet számítógépes automatizálása a következőképpen van megoldva. A számítógép CAMAC kimenő regiszteren keresztül elindítja a mérési folyamatot azáltal, hogy a kimenő regiszter megfelelő bite működteti az adagolószелеpet és így Dirac-impulzust hoz létre az áramlási cső bemenetén.

Az áramlási cső végén levő detektor jele erősítőken keresztül egy multiplexer bemenetére kerül. A multiplexer a jelet analóg—digitális átalakító modulhoz továbbítja, amely azt mintavételelesen mér. A mérési ütemet óragenerátor modul adja. A mérés addig tart, amíg a görbe a mérés előtti alapvonalat ε hibával megközelíti. A mérés után a számítógép kiszámítja a momentumokat, kiírja a végeredményeket és ellenőrző adatokat.

A számítógépre egyidejűleg négy ilyen mérés köthető. Ezek közül kettő egy laboratóriumban, kettő egy technikumban van és mindkét hely kb. 200 méterre van a mérőközponttól, ahol a számítógép és a CAMAC egységek elhelyezkednek. A mérőközpontban a készülékeket kiszolgáló teljes hardwारे tartalmazza a számítógépet (TPA—1001) 12 K-s operatív tárolóval, bővített aritmetikával, lyuk-szalag-perifériákkal, továbbá az alábbi CAMAC egységeket:

- 2 db CAMAC keret tápegységgel,
- 4 db előerősítő,
- 4 db szűrőerősítő,
- 1 db multiplexer,
- 1 db analóg—digitális átalakító,
- 1 db órajelgenerátor,
- 1 db kimeneti regiszter,
- 1 db keretvezérlő (U típusú).

A rendszerben alkalmazásra került ezen kívül minden készüléknél közvetlenül elhelyezett vezérlő-elektronika az adagolószелеp átkapcsolására és szimmetrikus kimenetelű erősítő a detektorjel nagy távolságú továbbításához. (Az analóg jelet páronként sodrott, közös külső árnyékolású kábelben vezetik.)

A CAMAC modulok a központban két keretben helyezkednek el. Az egyik keretben detektorjelet fogadó, szimmetrikus bemenetű, nagy közösfázisú zavarvédelmet biztosító előerősítők és az ezekhez csatlakozó szűrőerősítők találhatóak, ez esetben fixen beállított erősítéssel, így ebben a keretben nincs keretvezérlő. A másik keret tartalmazza a többi CAMAC modult a keretvezérlővel.

A rendszert vezérlő programot assembler-ben írták, helyfoglalása: 4 K-szó. A további 8 K tárolókapacitás adatmezőnek van fenntartva a 4 készülékhez tartozó koncentrációeloszlás-görbe tárolására. Az adatmező felosztása: 4×2 K. A program a 4 készülék párhuzamos, időben független működését teszi lehetővé, ami azt jelenti, hogy a készülék adott időben a program által nyilvántartott állapotok (klkapsolt állapot, számítás, mérés) bármelyikében lehet. A program három főbb részre bontható. Bevezető részből, mérésadatgyűjtő, vezérlőprogramból és kiértékelő, adatkiadó programból áll. A bevezető részben a program megnézi, hogy van-e mérésre kész berendezés, és ha van ilyen, írógépben bekéri a készülékre vonatkozó paramétereket:

- a készülék számát;
- a mérési összeállítás ismertetőszámát;
- a mintavételi frekvencia adatait;
- a mérés ismétlési számát.

A mérésgyűjtő, vezérlőprogram:

- meghatározza az aktuális alapvonalat;
- az adagolószелеp állításával kiváltja a Dirac-impulzust;
- megállapítja az eloszlásgörbe kezdetét;
- a megfelelő szektorban tárolja a koncentrációértékeket;
- meghatározza a koncentrációgörbe végét;
- visszaállítja az adagolószелеpet;
- számolásra jelentkezik.

A kiértékelő programrész meghatározza a momentumokat és a csúcsra vonatkozó néhány jellemző adat (alapvonal, maximum, első és utolsó feldolgozott mérési adat) kíséretében kiírja az írógépben.

A kiértékelő és a bevezető program mind a négy készülékre közös. Ezeket bármelyik készülék igénybe veheti, de egyidejűleg csak az egyik. A mérésadatgyűjtő program viszont minden készülékre külön-külön biztosítva van.

A görbe felvételénél — a tárolóhellyel való takarékoskodás miatt — a görbe nem kerül tárolásra a kiváló impulzus és az eloszlásgörbe megjelenése közötti holtidőben. Itt lényegében csak az alapvonalat mérhetnénk. Azért, hogy az időskálát ne veszítsük el, tárolásra kerül viszont itt is a mérések száma. Ha a mérendő görbe az alapvonal ingadozását figyelembevevő ε értékektől legalább egy előírt további $\Delta\varepsilon$ értékkel eltér, megkezdődik a koncentrációértékek tárolása. Ugyancsak tárolóhely takarékoság érdekében, ha a görbe a maximumán túljutott, a mintavételi frekvenciát átkapcsolják egy alacsonyabb értékre. A leggyorsabb eloszlásgörbe 2 s-os, a leglassabb kb. 750 s-os. (Ez az adat magában foglalja a holtidőt és a görbe felfutási, lecsengési idejét.)

6.3.

Orvosi-biológiai rendszerek

Az orvosi-biológiai alkalmazások jelentősége az utóbbi időben megnőtt. Mostanában készülnek el — sok helyen még kísérleti állapotban — az első számítógépes mérőrendszerek.

A számítógépes orvosi-biológiai rendszerek három nagy csoportba oszthatók:

1. Orvosi-klinikai rendszerek.
2. Kísérleti biológiai rendszerek.
3. Egyéb (például pszichológiai) rendszerek.

Az *orvosi-klinikai* rendszerek közvetlenül a humán vizsgálatokkal, diagnosztizálással, gyógyítással kapcsolatosak. A főbb területek a következők: intenzív ápolás, nukleáris medicina, EKG, EEG stb. és klinikai laboratórium.

Az *intenzív ápolási rendszer* feladata az, hogy az olyan kritikus állapotban levő beteget ellenőrizi, akinek állapota rövid idő alatt végzetessé válhat. Hosszú ideig folyamatosan mérik a beteg különböző életfunkcióinak jellemzőit (vérnyomás, pulzusszám, lélegzéssűrűség, hőmérséklet stb.) és a mért értékeken határérték- és trendszámításokat végeznek. A rendszer célja az, hogy a kritikus helyzetet gyorsan felismerje és a kezelőszemélyzetnek alarmjeleket küldjön. Mivel a mérőérzékelők közvetlen kapcsolatban vannak a beteggel, nagyon szigorú életvédelmi szabályokat kell megtartani a villamos balesetek elkerülése miatt. Ezért CAMAC-kal még csak kísérleti intenzív ápolási rendszerek készültek.

A *nukleáris medicina* leggyakoribb módszere az, hogy valamilyen radioaktív izotópot visznek be a szervezetbe, és ennek sugárzását mérik. Ily módon a sugárzó izotóp térbeli és időbeli eloszlása képet ad a betegségről, ill. a gyógyszerhatásról.

A mérések egyrészt statikusak, azaz a szervezet egyes pontjaihoz helyezett sugárzás-mérő detektorok rögzítve vannak, ezekhez számlálók tartoznak, és a számítógép e számlálók értékét olvassa be. Ilyen megoldást alkalmaznak például szívkatéter laboratóriumban, amikor a szív meghatározott kamrájába helyezett katéteren keresztül injektálnak izotópot a vérbe, és mérik az izotóp haladását. A mérések másik része dinamikus: ilyenkor vagy a detektort, vagy a radioaktív tárgyat mozgatják egy síkban, és mozgatás közben egy-egy diszkrét helyen mérik az intenzitás értékét. Evvel a módszerrel az izotóp eloszlásának térképét lehet megkapni. A nukleáris orvosi mérések igen nagy része CAMAC rendszerben készül. Példaképpen egy dinamikus, letapogatási elven működő rendszert ismertetünk. Megjegyezzük, hogy a rendszer elemeiből más összekapcsolással további mérési összeállításokat is létrehozhatunk.

A példaképpeni mérőrendszer kis radioaktív izotóppal kezelt szervezetek mintavételes letapogatású feltérképezésére készült [5]. Ha kis szervezetet kell letapogatni, egyszerűbb a detektort rögzített helyzetben tartani és a tárgyat mozgatni, ezért választották itt ezt a megoldást. A radioaktivitást mérő detektorból származó impulzusok — megfelelő erősítés után — a CAMAC számlálóba jutnak. A tárgyasztal mozgását X és Y irányban a számítógép vezérli: a mozgó léptetőmotorok egy kettős, léptetőmotor-meghajtó modulhoz kapcsolódnak. A hardware rendszer egy PDP—8 típusú számító-

géből áll, amelyből a lyukszalag-perifériákon kívül sornyomtató is tartozik az adatok, „térképek” kiolvasására. A CAMAC rendszer egykeretes, programozott vezérlőn keresztül kapcsolódik a számítógéphez. A paramétereket írógépén lehet beadni, a számítógép ezután megfelelő lépésekben automatikusan mozgatja a tárgyasztalt, és végzi el az impulzuszámláló méréseket. A teljes minta letapogatása után a sornyomtatón kiadja a radioaktivitás térképét a mért tárgyban.

Az említett radioizotópos méréseken túlmenően még igen sokféle sugárzásos méréstechnikát alkalmaznak: újabban például lassú és gyors Anger-kamérákat vezérelnek számítógéppel. Ezeknél igen nagy a csúcsadatsűrűség, ezért az analóg—digitális konverter után egy CAMAC rendszerben megépített FIFO pufferregiszter van, amely a csúcsterhelések alatt az adatokat tárolja.

Az orvosi-klinikai rendszerek következő nagy csoportja az élő szervezet különböző helyeiről nyert közvetlen vagy közvetett *elektromos jelek felvételével*, vizsgálatával foglalkozik. Ezek közül a legfontosabban az EKG (elektrokardiográf), EEG (elektroencefalográf), EOG (szemmozgás-regisztrálás), EMG (izomaktivitás-regisztrálás), ESG (lélegzés-regisztrálás). Ezek a berendezések általában önálló, számítógéptől független eszközök, amelyek papírszalagon jelenítik meg a jel időbeni változásait. Ezeket a papírszalagokat elemzi ezután a diagnosztizáló orvos különböző jellemzők szerint: jelalak, frekvencia, jel nagyság, a jelek differenciálhányadosai s. í. t. Az elemzés másik módja az, hogy összefüggéseket keresnek a különböző helyekről nyert elektromos jellemzők között. A jelek egyedi, ill. egymással összefüggő vizsgálatát megkönnyíti a számítógép használata. A számítógép vagy közvetlenül kapcsolódik a mérőeszközhöz, vagy közvetve, például mágnesszalagos adathordozó útján. Ilyenkor az EKG vagy EEG stb. berendezés kimenete analóg mágnesszalagos rögzítőkhöz van kötve. A folyamatos mérési adatfelvétel után a mágnesszalagon levő mérési eredményeket analóg—digitális konverteren keresztül betáplálják a számítógépbe, ahol megtörténik a feldolgozás. E célra gyakran használnak CAMAC perifériákat.

Példaképpen bemutatunk egy összetett alvászanalizáló-mérő összeállítást [12]. A vizsgált személyről alvás közben EEG, EOG, EMG, ESC és EKG görbéket vesznek fel és 7-csatornás analóg mágnesszalagos tárolón rögzítik. Ezután a mágnesszalagot gyorsított ütemben (16 x sebességgel) visszajátsszák CAMAC modulokon keresztül a számítógépbe. A jeleket feldolgozó analógmodulok külön EEG célra készült funkcionális modulok, amelyek vagy jellegzetes részeket tudnak kiválasztani az analóg jelekből, vagy pedig digitális jelekké alakítják azokat. Az EEG jel útja a funkcionális CAMAC modulban a következő: a bemenő jel sávátresztőre jut, majd egyenirányítódik, ezután kissé integrálódik, majd amplitúdódiszkriminátorra jut. A diszkriminátor szintjét számítógép- (CAMAC) utasítással lehet beállítani, párhuzamos kimenő regiszteren keresztül. A diszkriminátor állapota a bemenő regiszteren keresztül olvasható be a számítógépbe. Az EEG-jel több, különböző frekvenciatartományra beállított sávátresztőre, ill. analógjel-feldolgozó csatornára jut. Az EEG görbéket további két diszkriminátor modul vizsgálja: az egyik a jelek gyors változását figyeli, a másik pedig azokat a jeleket észleli, amelyek adott időn keresztül túlhaladnak egy beállított amplitúdóértéket. Hasonló diszkriminátorra jutnak a szemmozgás EOG jelei is. A lélegzési, izomaktivitási és EKG jelek megfelelő diszkriminálás után számláló bemenetekre jutnak.

A hardware rendszer egy NOVA 1200 kisszámítógépből áll, 4 K memóriával. Írógép- és lyukszalag-perifériák mellett rajzológéppel van kiegészítve. A CAMAC modulok egy keretben helyezkednek el és programozott keretvezérlővel csatlakoznak a géphez.

Az alkalmazott CAMAC modulok a következők:

- 2 db funkcionális szűrőmodul, egyenként 6 különböző frekvenciasávval;
- 2 db párhuzamosan kimenő regiszter a diszkriminátor beállítására;
- 2 db funkcionális diszkriminátor modul;
- 1 db párhuzamos bemenő regiszter;
- 1 db négyes számlálómodul;
- 1 db időzítőmodul.

A mérő programrendszer 160 ms-onként beolvassa a diszkriminátorok IGEN—NEM értékeit, majd egy-egy diszkriminátorra nézve egyperces időtartamra összegzi az IGEN értékeket; ez az érték 0 és 375 közötti szám. Az EKG és ESG jeleket számláló számlálók értékét percenként olvassa be, az

EMG-t percenként 15-ször. A program a beolvasott értékekből különféle számításokat végez; a frekvenciákból, ill. meredekségekből alvás-típusokat határoz meg stb.

A klinikai laboratóriumok — az orvosi ellátás fejlődésével — egyre több rutinvizsgálatot végeznek. Ezek a vizsgálatok különböző, az emberi testből származó anyagok analitikus vizsgálatait (vér, vizelet stb.), amelyek igen közel állnak a kémiai laboratóriumi vizsgálatokhoz.

Példaképpen egy automatikus vércukormeghatározást végző rendszert ismertetünk [31]. A preparált vérminták egy 40 mérendő mintát és 6 referenciámintát tartalmazó mintaváltóban helyezkednek el. Először a 6 ismert vércukortartalom-mintát mérik, majd az ismeretlen összetételüket. A mérés flow-rendszerű kémiai automatában történik, a kémiai reagensbe beleszívják a vizsgálati anyagot: az anyag fényelnyelő képessége függ a vércukor mennyiségétől. Fotométer méri az anyagon átjutott fény mennyiségét, a fotométer kimenő árama jut az analógdigitális konverterre. Az A/D konverterből 1 másodpercenként mintát vesznek, minden vérmintából körülbelül nyolcvanszor. Ezalatt kirajzolódik egy amplitúdócsúcs. A görbe maximuma jellemző a vércukortartalomra. Hasonló elvű a vizelet nitrogéntartalmának mérése is. Egy A/D konverterhez multiplexeren keresztül több mérés csatlakoztatható. Lényegében ugyanez a számítógépes berendezés használható aminoszerek vizsgálatára: ekkor a görbe csúshelyezete és területe jellemzi a mérendő anyagmennyiséget, itt azonban alaphelyzet-korrekciót is kell végezni.

A hardware konfiguráció TPA-i kiszámítógép 8 K tárolóval, írógéppel és lyukszalag-perifériákkal. A CAMAC modulok egy keretben helyezkednek el, programozott vezérlő csatolja őket a számítógéphez. A CAMAC modulok a következők:

- 32-csatornás multiplexer;
- mintavételes analóg—digitális konverter;
- időzítőmodul.

A programrendszer a mérés elvégzése után kiszámítja az etalonok alapján a vércukor stb. mennyiségeket.

A kísérleti biológia számítógéppel segített mérései közül gyakoriak a spontán vagy kiváltott aktivitásmérések. A mérés lényege itt az, hogy valamilyen sejt vagy sejt-populáció elektromos aktivitását regisztrálják különböző mérőszondákkal. Az aktivitás lehet spontán vagy létrejöhet külső, például elektromos ingerlés hatására is. Ez utóbbi módszerrel mérhető például valamely idegben az inger hatására keletkező ingerület terjedése. A mérőszondában a mérés számára lényeges, „hasznos” jelek igen nagy egyéb, a mérés szempontjából „háttérzaj”-ban vannak elrejtve, ebből kell különféle számítástechnikai módszerekkel a hasznos jelet kiválasztani. A felvett adatokon időbeni eloszlást, egymásra hatást, korrelációt stb. számolnak. Mivel ezek a mérési összeállítások gyakran módosulnak, a CAMAC rendszer modularitását, könnyű változtathatóságát jól ki lehet használni: egyre gyakoribb, hogy kísérleti biológiai célokra CAMAC rendszereket használnak.

A felsorolt orvos-biológiai területeken kívül még több e területhez közel álló, esetleg ugyanezeket a módszereket használó alkalmazásban vannak CAMAC rendszerek, így elsősorban a pszichológia, pszichofiziológia, munkaalkalmasság stb. területeken.

6.4.

Ipari alkalmazások

Az ipari alkalmazások — a mérő-szabályozó rendszer elhelyezése, szervezése szempontjából — két nagy csoportra oszthatók. Az elsőbe azok a rendszerek tartoznak, amelyeknél a számítógép és a real-time perifériák egymáshoz közel, egy helységben helyezkednek el és a mérőátalakítóból érkező, ill. a folyamattal kapcsolatos jelek ere-

deti formájukban érkeznek ebbe a helységbe. Ez az elrendezés kis mérőautomaták vagy kis területen levő mérőadók és beavatkozó szervek esetében nagyon kényelmes. Megfelelő kábelezési, jelvezetési óvatossággal még egy üzemsarnok vagy erőművi blokk területén levő mérési és beavatkozó pontok jelei is lényegében eredeti formájukban a számítógépes mérőrendszerhez vezethetők. Ezekben a rendszerekben a CAMAC ADATÚT, ill. CSOPORTÚT használható.

Ha a mérési és beavatkozó pontok nagyobb területen helyezkednek el, akkor már nemigen lehet, ill. nem célszerű a primer műszerezés jeleit közvetlenül, eredeti állapotukban a központi számítógépterembe vezetni. Így például a különböző távvezetékek, közműhálózat stb. adatainak gyűjtése és vezérlése, már inkább a távméréssel kapcsolatos megoldásokat igénylik. A real-time perifériák a hálózat egy-egy távoli, célszerűen kiválasztott pontján vannak elhelyezve, ezek a primer műszerezés jeleit átalakítják adatátvitelre alkalmas, kódolt digitális formába, és ezek jutnak a központba. Ezeknél a rendszereknél a CAMAC SOROS FŐÚT-at lehet használni, vagy pedig a központban levő CAMAC keretben elhelyezett adatátviteli modulok segítségével csillagrendszer lehet kialakítani.

Az ipari jellegű folyamatokból érkező analóg jelek általában a következők lehetnek: mérőadók, mérőátalakítók jelei (egyenfeszültség vagy egyenáram kimenettel), ellenállástávadók, ellenálláshőmérők, hőelemek jelei stb. A digitális jelek bináris formában jelennek meg. Ilyenek például csapok, szelepek állapotának jelei, motorok bekapcsolt állapota, tartály üres vagy tele állapota stb. Ugyancsak digitális formában jelennek meg szögávadók jelei, kapcsolók, nyomógombok, különböző berendezések mért adatai. Olyan adók is vannak, amelyeknél az impulzusszám az információhordozó (pl. teljesítménymérő, áramlásmérő, fordulatszám-mérő stb.). A beavatkozó, kimenő oldalon a jelek hasonlóak lehetnek: analóg jelek, digitális jelek, impulzusszám. Ezek az ipari feladatok általános célú CAMAC modulokkal könnyen megvalósíthatók, esetenként azonban szükség lehet egy-egy speciális jelátalakítóra (pl. jelszint-átalakító), vagy csatoló elemre. Ezért célszerűbb funkcionális CAMAC modulokat használni, amelyek közvetlenül, kényelmesen illeszkednek egy-egy jeladóhoz. Ilyen funkcionális modulcsaládot analóg bemenő jelek fogadására a 4.3.2. pontban írtunk le.

Mérési adatgyűjtést és folyamatellenőrzést végző rendszerek programjai sok esetben nagyon hasonló feladatokat látnak el. Az analóg bemeneteket adott ciklusidővel ciklikusan lekérdezik, a beérkezett adatokon hihetőség-vizsgálatot végeznek, lineárizálják, esetleg korrigálják és különféle típusú határérték-túllépéseket vizsgálnak. A digitális bemeneteket vagy szinten ciklikusan lekérdezik és összehasonlíttják adott vagy előző értékkel, vagy pedig a veszélyállapot jelző digitális bemenetek programmegszakítást okoznak. A beérkezett analóg és digitális adatokon különféle számításokat végeznek (például hatásfokszámítás), majd a redukált, számított adatok különböző naplókön, tablókon, analóg regisztrálókon megjelenítésre kerülnek. Folyamat-szabályozásnál a fentiekén túlmenően a szabályozási algoritmusnak megfelelő számításokat is elvégzi a számítógép, és az eredményektől függően különböző digitális vagy analóg beavatkozó szerveken keresztül vezérli a folyamatot. A központi egység programjai — a CAMAC perifériákat mozgató kezelőprogramoktól eltekintve — általában függetlenek a real-time perifériarendszertől, bár újabb törekvések szerint például az analóg bemenetekkel foglalkozó adatredukáló és ellenőrző programrészeket rábízják a CAMAC rendszerben levő összetett vezérlőre, és csak az ellenőrzött, sűrített adatok kerülnek át a központi számítógépbe.

Felsorolunk néhány ipari vonatkozású, CAMAC-kal megvalósított alkalmazást:

- szinkrongenerátor on-line vezérlése;
- vízmű és szivattyúállomás mérési adatgyűjtése és vezérlése;

- műanyagfólia-gyártáskor folyamatos vastagságmérés és ellenőrzés;
- azbesztcementlap-gyártás automatizálása;
- hőerőmű adatgyűjtése és ellenőrzése;
- több munkahelyű elektromos bemérőállomás;
- mozgó (gépkocsira szerelt) laboratórium ipari technológiai folyamatok (például üvegyipar) vizsgálatára;
- belsőégésű motor próbapad-vezérlés;
- szerszámgépvezérlés;
- hengermű-automatizálás;
- radioaktív minták besugárzásának és szállításának automatizálása.

Példaképpen részletesebben ismertetünk három ipari alkalmazási megoldást: egy műanyagfólia-gyártó gépsor automatizálást, egy vízmű- és szivattyúállomás mérő—ellenőrző rendszerét és egy erőművi blokk mérő—ellenőrző rendszerét.

Egy számítógéppel vezérelt CAMAC rendszer folyamatosan méri *műanyagfólia-gyártó gépről lekerülő fóliák vastagságát* [25]. Jelenleg öt gépről lekerülő fóliát méri, maximális kiépítésben tizenöt géphez tud kapcsolódni.

A vastagságmérés radioaktív sugárzással történik. A sugárzás elnyelésének mértéke jellemzi a vastagságot. Minden fóliagyártó gépen van egy sugárforrás és egy radioaktív sugárzást mérő fej, azaz egy sugárzásdetektor, amely 0...10 V kimeneti feszültséget ad ki. A mérőfej mozgatható a fólia szélességének irányában, így a teljes szélességben végigmérhető a vastagság. Minden szélességben 100 ponton történik a mérés. Egy-egy tekercset hosszirányban 5 helyen mérnek meg.

A fóliagyártó géptől a következő jelek érkeznek be a CAMAC rendszerbe:

- a sugárzásdetektor (vastagságmérő) analóg jele;
- a fej mozgását jelző jel;
- fólia-tekercsváltást jelző jel.

A CAMAC rendszer a fóliagyártó gép felé a következő jeleket küldi:

- a fejmozgatást indító jel;
- a fejmozgatást felszabadító jel (kézi mozgatáshoz);
- vészjel, ha a mért és az írógépen leadott szükséges érték közötti eltérés az előírtnál nagyobb;
- „mérés folyamatban” jel.

A hardware rendszer a következő elemekből áll. A számítógép PDP 11/05 típusú, 8 K tárolóval, bővített aritmetikával ASR 33 írógéppel és LOGABAX nyomtatóval. A CAMAC modulok egy keretben helyezkednek el és egyszerű, programozott vezérlővel kapcsolódnak a számítógéphez. A keret a következő CAMAC modulokat tartalmazza:

- 1 db kvarcóra;
- 1 db 16 bemenetű analóg multiplexer;
- 1 db analóg—digitális konverter;
- 1 db számláló;
- 1 db 24-bites párhuzamos bemenő regiszter;
- 2 db 16-bites reed-relés párhuzamos kimenő regiszter;
- 3 db kettős digitális—analóg konverter, regisztrálók (X—Y írók) vezérlésére.

Az analóg multiplexer, A/D konverter és számláló összetartozó modulcsaládot alkot a bemenő analóg jelek vezérlésére. A többi modul általános célú.

A számítógép a vastagságellenőrzés és határérték-vizsgálat mellett különböző számításokat, naplózásokat is végez, így például a naplóba kerülnek az egyes fóliatekercsek főbb adatai: dátum, gépszám, tekercsszám, tekercshossz, keresztmetszet, elméleti számított súly, felületi határfok. A programozási rendszer alapja egy egyszerű real-time monitorprogram, amely különböző task-okat vezérel prioritásuknak megfelelően. A fontosabb task-ok a következők: a dialógus a kezelővel, fejmozgatásindítás, mérés (A/D konverzió) indítás és mérés, középértékszámítás, számítások és eredménykiadás.

Egy másik, körülbelül hasonló nagyságú, megvalósított rendszer azbeszt cementlap gyártósor mellett működik. E rendszer a vastagságmérés után nemcsak ellenőriz és naplóz, hanem egy szabályozó algoritmus kiszámításával a nyersanyagfordozó motorra is visszahat és az előírt lapvastagságot szabályozza [20].

A következő példában egy víztisztító és szivattyúmű mérő- és adatgyűjtő rendszerét mutatjuk be [24]. A víztisztító és szivattyúmű különböző pontjain vízminőséget és vízmennyiséget kell mérni.

A vízminőséget a következő mérésekből állapítják meg: vízzavarosság mérés (fényáteresztő képesség), hőmérséklet, lebegő anyagok százalékos aránya, elnyelt oxigén százalékos aránya, oxidálhatóság (KMNO₄ elnyelés), különböző lebegő sók koncentrációja, csírák koncentrációjának mérése. Ez utóbbi kivételével a többi mérést a vízkezelés különböző fázisaiban folyamatosan mérik. A vízmennyiség-mérésnél két határértéket figyelnek: a vízmű maximális kapacitásának megfelelő mennyiséget és egy — a csőrendszerrel függő — minimális értéket. (Erre azért van szükség, mert ha ez alá csökken a termelt vízmennyiség, különböző organizmusok szaporodhatnak el a csőrendszerben.) A vízműkapacitás 1500 l/s.

A vízkezelési technológia különböző fázisai a következők: nyersvíz-szivattyúzás, sterilizálás, flokulálás, mechanikus szűrés, pH szabályozás, ózonizálás, maradék ózon eltávolítás, klórozás. A víztulajdonság-méréseket a technológia különböző fázisaiban, valamint a vízmű kimenő csővezetékén mérik. Ezen kívül víznyomásesést is mérnek a hálózat különböző pontjain.

A technológia berendezésből a CAMAC rendszer a következő jeleket kapja: 50 db analóg jelet, 0...20 mA-es távadókról 7000 db digitális jelet.

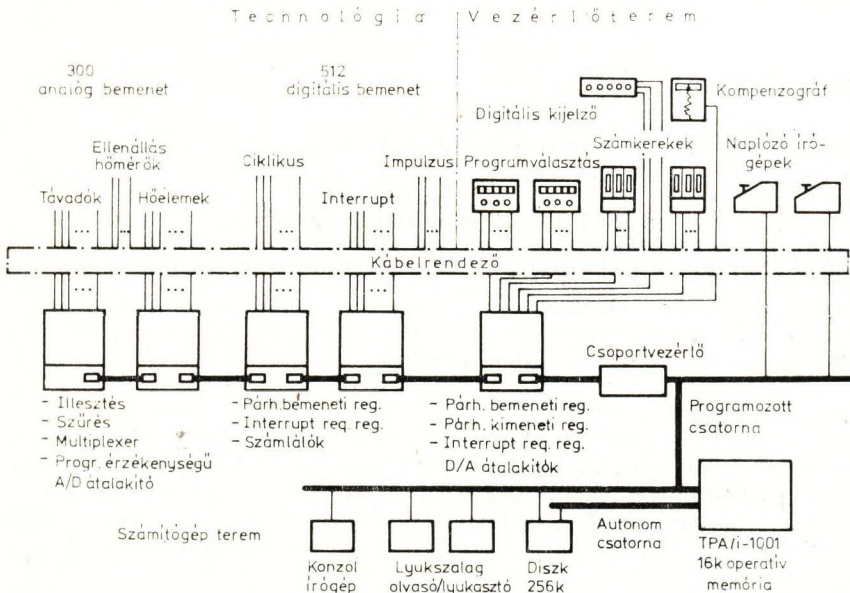
A technológiai berendezéshez 60 db digitális kimenő jelet küld a mérő-ellenőrző rendszer.

A hardware rendszer elemei a következők: a számítógép NOVA 1200 típusú, közvetlen perifériás egységei a következők: fixfejes mágneseslemez háttértár, papírszalagolvasó és írógép. A CAMAC rendszer két keretben helyezkedik el; ezt egyszerű programozott vezérlő illeszti a géphez. A rendszerben a következő típusú modulok vannak:

- 12-bemenetű analóg multiplexerek;
- kettős digitális voltmérő (analóg—digitális konverter);
- kettős, 16-bites párhuzamos kimenő regiszterek;
- kettős digitális—analóg átalakítók;
- 64-bites megszakításmérő bemenetek.

A számítógép a vízmű különböző pontjainak állapotát folyamatosan ellenőrzi és naplózza. A rendszerben levő hiba esetében a post-mortem naplóból következtetni lehet a hiba kifejlődésére. A gép ezen kívül az összegyűjtött adatokat analizálja, és ennek alapján jelzéseket ad megelőző karbantartásra, optimális működtetési feltételek kialakítására, szimulációs modellek felállítására. A programozási rendszer itt is egyszerű real-time monitor köré épül fel.

A következő példa egy nagyobb CAMAC rendszert mutat be. A feladat egy hőerőműben megvalósított mérő-, adatgyűjtő, ellenőrző rendszer [6.32]. A hőerőmű minden 215 MW-os blokkjának önálló számítógépes rendszere van.



6.5. ábra
Hőerőművi mérésadatgyűjtő rendszer

A blokk különböző technológiai pontjain kell hőmérsékletet, nyomást, elmozdulást, tengelyrezgést stb. mérni. Az erőművi technológia felől a következő jelek érkeznek a CAMAC rendszerbe (6.5. ábra):

- 300 db analóg jel (távadók, ellenállás hőmérők, hőelemek stb.);
- 512 db kétállapotú digitális jel (szelepállások helyzete stb.);
- 32 db megszakításkérő jel (vészhelyzetek);
- impulzussorozat (pl. teljesítménymérőből);
- nyomógombok és számkerekek jelei a vezénylőtermi pultról.

A CAMAC rendszerből a vezénylőterembe analóg kirajzolók és digitális kijelzők vezérlőjelei jutnak. A hardware rendszer TPA-i kiszámítógépet tartalmaz, 16 Kszó operatív tárolóval, fixfejes mágneslemez háttértárral, papírszalag-perifériákkal és 4 db írógéppel (konzol-, ill. naplózó írógépek).

A CAMAC rendszer 5 keretben helyezkedik el. Egyszerű, programozott csoportvezérlővel kapcsolódik a számítógéphez.

Az alkalmazott modulok a következő típusúak:

- 16-csatornás analóg multiplexer;
- multiplexer vezérlő;
- integráló analóg—digitális konverter;
- híd és hídtápegység;
- 24-bites párhuzamos bemenő regiszter;
- 16-bites megszakításkérő bemenet;
- 24-bites párhuzamos kimenő regiszter;
- digitális—analóg átalakító analóg regisztráló meghajtására;
- LAM osztályozó.

Az erőművi technológiához az összes kapcsolat potenciálfüggetlen. Az analóg bemenő modulok összehangolt családot alkotnak (l. 4.3.2. pont).

A programrendszer különböző ciklusidőkkel vezérli a méréseket, számításokat, vizsgálatokat végez a mért értékeken. Határértéktúllépés és veszélyhelyzet esetében riasztja a kezelőt. Ezen kívül többféle szintű naplózási, kijelző feladatokat lát el.

A software rendszer magas szintű — a real-time perifériák mozgatói utasításaival kiegészített — BASIC nyelven készült és mágnes tárcsás real-time operációs rendszerben fut.

6.5.

Egyéb alkalmazások

Az eddig tárgyalt nagyobb alkalmazási területeken kívül igen szerteágazó feladatokra használják még a CAMAC rendszereket, így:

- vízalatti hangfelvétel-analízis (halrajok vonulásának figyelése);
- meteorológiai adatgyűjtő rendszer;
- repülőgéphez épített γ scintillométer;
- levegőtisztaság-ellenőrző rendszer;
- csillagászati mérések automatizálása;
- számítógépes hálózatok, adatátviteli rendszerek.

E két utóbbi alkalmazását részletesebben ismertetjük.

A *csillagászati mérések* nagyon hasonlítanak a fizikai mérésekhez annyiban, hogy a számítógépes mérő összeállítás, ill. maga a mérés változó, és olyan rendszert célszerű kialakítani, amely alkalmas arra, hogy mind a hardware, mind a software könnyen és gyorsan átváltható legyen az újabb mérési igényeknek megfelelően. Két mérést írunk le kissé részletesebben, ezek a következők: nagysebességű fotometria és a scanning polariméter [27].

A *nagysebességű fotometria* célja az, hogy a csillagok fényének periodikus fluktuációját meghatározza. Ezt a mérési módot használják például plazmarezgések mérésére vagy pulzások frekvenciaváltozásának

mérésére. A szokásos rezgési idő néhány tized másodperc és néhány másodperc között változik.

A csillag fénye a teleszkópba épített fotométerre jut. A fotométer kimenetén a fényárammal arányos frekvenciájú impulzussorozat jelenik meg, ez jut be a CAMAC rendszerbe.

A mérőrendszer egykeretes, a TPA-i kisszámítógéphez programozott egyszerű vezérlőn keresztül kapcsolódik és a következő modulokat tartalmazza:

- 24-bites számláló;
- időzítőóra;
- megjelenítő vezérlő és megjelenítő;
- párhuzamos digitális bemenő modul az abszolút időjelek fogadására.

A mérés menete a következő. A feltételezett periódusidőt a felbontásnak megfelelően felosztják, általában 50 részre. Egy-egy részdőtartamig a számláló számolja a fotométerből jövő impulzussorozatot, majd a részdőtartam végén egy tárolóhelyre tárolja. Ezután a következő időtartam impulzusainak számlálása történik stb. Amikor a feltételezett periódusidő befejeződött, újra az első részdőnek megfelelő tárolóhely következik: ennek előző tartalmához adódik hozzá az új számlálóérték. A mérés ily módon ciklikusan folytatódik, a tároló tartalma közben a megjelenítőben figyelhető meg. Ha a feltételezett periódusidő megegyezik (ill. többszöröse) a tényleges idővel, szignifikáns csúcs alakul ki. Ha a csúcs mérés közben csúszik, a periódusidőt — például az írógépen keresztül vezérelve — módosítani kell. Mérés előtt különböző hitelesítéseket végez a rendszer, mérés végén az eredmény kinyomtatódik.

Az előző méréshez kissé hasonló, de annál jóval bonyolultabb a *scanning polarimetria*. A lényeg itt is a csillag fényerőváltozásának a mérése, azonban a változás most nem időbeni, hanem a csillag átmérője mentén egyenes vonalon térbeni. A csillag fénye egy diafragmán keresztül kerül fotométerre, a diafragmát egy léptetőmotor mozgatja egy egyenes mentén. Ennél a mérésnél két fotométert alkalmaznak, mivel itt nem közvetlen, hanem lineárisan polarizált fény két komponense kerül egy-egy fotométerre. A diafragma egy-egy helyzetében adott ideig számlálják a két fotométerről érkező impulzussorozatot egy-egy számlálóba, majd tárolják a gép tárolójába. Ezután tovább mozdul a diafragma, új számlálás kezdődik stb.

A számítógépes összedíltás az előzőhöz hasonló, csak több CAMAC modult igényel. Az alkalmazott modulok a következők:

- 2 db 24-bites számláló;
- időzítőóra;
- megjelenítő vezérlő és megjelenítő;
- párhuzamos digitális bemenő modul az abszolút időjelek fogadására;
- 2 db léptetőmotor-meghajtó modul.

A számítógép mérés közben és után különböző fényáramadatokat számol, majd az eredményeket naplószerűen kinyomtatja.

Kisebb (egy nagy gép köré felépített) számítógéphálózatokban is használnak CAMAC rendszereket.

Példaképpen egy IBM 370/165 köré telepített hálózatot ismertetünk [29], [8], amely egy nagy fizikai kutatóintézet számítási és mérési automatizálási feladatait látja el. A főgép 2 Mbyte ferrittárolóval, 16 Kbyte-os gyors puffertárolóval, fixfejes nagy diszkkal, mágnesszalaggal és konvencionális perifériákkal rendelkezik.

A főgép 12 külső, gyors perifériás csatornája egy IBM 1802 gépet használ *front-end* processzorként. Az 1802-hez csatlakozó 12 külső, gyors perifériás csatornán egy-egy kisszámítógép van. Ezek a kisszámítógépek CAMAC perifériákkal vannak ellátva; az 1802 gyors csatornájához is egy speciális CAMAC modul csatolja a helyi rendszereket.

A helyi rendszerek különfélék lehetnek és a kutatóintézet igényei szerint változtathatók. Egy ilyen helyi rendszer például 40 alfanumerikus megjelenítőt tartalmaz, amelyek a főgép időosztásos opciója (TSO) alatt működnek. Ez a rendszer a nagy gép mellett van, és a CAMAC lényegében a 40 megjelenítő multiplikálási, üzenetösszerakási, koncentrálsai stb. feladatait látja el. Minden megjelenítőhöz tartozik egy egységnyi széles csatolómodul, amely egyben a távolabb (más épületben) levő megjelenítőhöz vezető kábeleket is meghajtja. A helyi rendszer vezérlő kisgépe PDP 11/05 típusú. A CAMAC vezérlő egyszerű, programozott és autonóm adatátviteli tulajdonságokkal rendelkezik. Az autonóm adatátviteli csatorna a nagy géppel való kapcsolatot biztosító speciális modullal működik együtt.

Egy további helyi rendszertípust *batch-terminal* céljára építettek ki. Ebben sornyomtató, kártyaolvasó és megjelenítő kapcsolódik CAMAC modulon keresztül egy egykeretes, PDP—11/05-tel vezérelt rendszerbe. A nagygéppel való kapcsolatot itt általában modemem keresztül valósítják meg. E célra 4800 baud-os, szintén CAMAC-ban megépített szinkron modemet használnak.

Elkészültek mikrohullámú átvivő csatornához illeszkedő modulok is: ezek segítségével a helyi rendszerek akár nagy távolságokban is lehetnek, és igen gyorsan lehet velük kommunikálni.

A fizikai kutatóintézet igényeinek megfelelően gyakoriak a kísérletekhez, mérésekhez közvetlenül kapcsolódó helyi rendszerek, amelyek teljes mértékben ki tudják használni a modulválaszték változóságát.

A számítástechnikai igények az élet egyre nagyobb területein jelentkeznek; így egyre nő az a részterület is, amelyben a CAMAC rendszerek alkalmazhatók. A CAMAC rendszer tulajdonképpen egy olyan eszközt jelent, amit a különböző szakterületek művelőinek a saját területükön kell alkalmazniuk a saját területük igényeiből kiindulva; a számítógépes szakembereknek pedig arra kell törekedniük, hogy ez az eszköz minél kényelmesebben és célszerűbben legyen kezelhető, és minél olcsóbban elérhető.

Irodalom a 6. fejezethez

- [1] Bal, F. et. al.: Chamber Read-out for MWPC. CERN—NP CAMAC Note 47—00, 1973.
- [2] Barsotti, E. J. et. al.: Operational Aspects of a Serial CAMAC Control System. IEEE Proceedings of the 1973. Nuclear Science Symposium, San Francisco, 1973. nov. 14...16.
- [3] Biri J. és mások: KFKI—CAMAC rendszerek nemnukleáris alkalmazásokban. Mérés és Automatika, 1974. XXII. 7. szám, 257. oldal.
- [4] Bertolucci, B.: CAMAC Magnetostrictive Readout System Using Shottky Memories. IEEE. Trans. Nucl. Sci., NS—20, No 1., 1973.
- [5] Brill, A. B. et. al.: CAMAC Applications in Nuclear Medicine at Vanderbilt: Present Status and Future Plans. IEEE Transactions on Nuclear Science NS—21, No 1., p. 892., 1974.
- [6] Burley, A. C. et. al.: A CAMAC System for Control of a Diffraktometer CAMAC Bulletin, No 3., 12. oldal., 1972.
- [7] Crowley, M. C.—Milling et. al.: The CERN SPS—Multicomputer and CAMAC Real-Time Control System. First Symposium on CAMAC, Luxemburg, 1973. Proceedings, p. 121.
- [8] Daniels, T. et. al.: The Daresbury On-Line System. Preprint of the Daresbury Nuclear Physics Laboratory, DN.PL/P 157, 1973.
- [9] Deane, A. M. et. al.: A CAMAC—Based Data-Processing System, LBCOM. CAMAC Bulletin, No 5., 1972.
- [10] Edelman, G. M.—Gall, W. E.: Cascade Chromatography and Automated Multidimensional Fractionation. Proc. Nat. Acad. Sci. USA Vol 68, No 7, 1971.
- [11] Farren, J.—Webster, R. K. (Harwell A. E. R. E.): Automation of Laboratory Instruments—A Review. First Symposium on CAMAC, Luxemburg, 1973. Proceedings, p. 109.
- [12] Gaillard, J. M.—Tissot, R.: Principles of Automatic Analysis of Sleep Records with a Hybrid System. Computers and Biomedical Research, 6, p. 1...13., 1973.
- [13] Görög P.—Somlai L.—Láng I.: Energiaeloszlás-analízis és spektrumkiértékelő programrendszer (EDMAP, SKP) CAMAC—TPA-i összeállításához. MTA—KFKI, MSZKI Programkönyvtár, 1974.
- [14] Hepinstall, L. J. et. al.: CAMAC Experimental Beam Line Control System. IEEE Trans. Nucl. Sci., NS—20, No 3., 1973.
- [15] Heywood, D. R.—Ozzard, B.: An Optically—Coupled Serial CAMAC System. IEEE Proceedings of the 1973. Nuclear Science Symposium, San Francisco, 1973. nov. 14...16.
- [16] Hills, P. M.: MULTILAB—A Laboratory Automation System Based MODUS 4 and CAMAC. CAMAC Bulletin, No 9., 1974.
- [17] Howells, M. R. et. al.: A CAMAC System for Computer Control of Spectrometers. CAMAC Bulletin, No 6, 3. oldal, 1973.
- [18] Klaisner, L. A.—Stephenson, J. M.: An Accelerator Instrumentation and Control System Using CAMAC. IEEE Trans. Nucl. Sci., NS—18, No 1., 1971.
- [19] Kroó N. és mások: Háromtengelyű neutronspektrométer CAMAC-egységekkel történő automatizálása. MTA—KFKI Rendszerterv. 1974.

- [20] *Liebl, P.*: System zur industriellen Prozesssteuerung des Herstellvorgangs von Asbest—Zement Platten mit Kleinrechnern und CAMAC im On-Line Betrieb. Proc. of the First Int. Symp. on CAMAC, p. 237., 1973.
- [21] *May, F. et. al.*: FOCAL Controls CAMAC. SGAE Berichte No 2047, EL—19/72, 1972.
- [22] *Roethe, K. P.—Gelbin, D.—Wolf, H. J.*: Einsatz des Prozessrechners TPA 1001 bei der verfahrenstechnischen Forschung. Zentralinstitut für Physikalische Chemie, Berlin—Adlershof.
- [23] *Sanghera, D. (EKCO)*: Multi-Channel Analyser System in CAMAC. CAMAC Bulletin, No 3., p. 9., 1972.
- [24] *Schindler, A.*: Computer CAMAC Based Control System for a 1500 l/s Waterpump Station. Proc. of the First Int. Symp. on CAMAC, p. 227., 1973.
- [25] *Servent, J. M.*: Controle d'épaisseur dans une installation industrielle. Proc. of the First Int. Symp. on CAMAC, p. 233., 1973.
- [26] *Servent, J. M.*: Nuclear Spectrometry. (SAIP—Schlumberger). CAMAC Bulletin, No 4., p. 15., 1972.
- [27] *Virághalmi G.*: Csillagászati mérőrendszer. MTA Csillagvizsgáló Intézet, Piskésetető (megjelenés alatt).
- [28] *Zaharov, B.*: CAMAC as a Processor Interface in Computer Networks. First Symposium on CAMAC, Luxemburg, 1973. Proceeding, p. 197.
- [29] *Zaharov, B.*: Integrated Computer and Data-Processing Networks. Preprint of the Daresbury Nuclear Physics Laboratory, DNPL/P 155, 1973.
- [30] *Zander, K.*: Status and Future of CAMAC in Western Europe. First Symposium on CAMAC, Luxemburg, 1973. Proceedings, p. 31.
- [31] Zentralinstitut für Ernährung 1505 Bergholz—Rehbrücke—Potsdam Arthur—Schennert—Allee 115—116.
- [32] *Zobor E.—Sándor Gy.*: TPA—CAMAC—MEDA Hybrid rendszer. Személyes közlés, 1975.
- [33] *Zwoll, K. et. al.*: A Computer Controlled Triple Axis Neutron Spectrometer with CAMAC Instrumentation KFA—Jülich kiadvány (Jül—774—ZE—FF), 1971.
- [34] 6000 Series Spectrum Analyzers SEN Elektronik GmbH, D2000 WEDEL/HMB. Gyártmányismerető leírás, 1975.

7.

Függelék

7.1.

CAMAC utasítások és jelek táblázatai



7.1. táblázat

CAMAC funkciókódok összefoglalása

Funkciókód	Funkció	R/W használat	F16 F8 F4 F2 F1 vonalak	IML kód
F(0)	GP1 olvasás	R	0 0 0 0 0	RD1
F(1)	GP2 olvasás	R	0 0 0 0 1	RD2
F(2)	GP1 olvasás és törlés	R	0 0 0 1 0	RC1
F(3)	GP1 komplement olvasás	R	0 0 0 1 1	RCM
F(4)	nem szabványosított	R	0 0 1 0 0	—
F(5)	fenntartott	R	0 0 1 0 1	—
F(6)	nem szabványosított	R	0 0 1 1 0	—
F(7)	fenntartott	R	0 0 1 1 1	—
F(8)	LAM vizsgálat	—	0 1 0 0 0	TLM
F(9)	GP1 törlés	—	0 1 0 0 1	CL1
F(10)	LAM törlés	—	0 1 0 1 0	CLM
F(11)	GP2 törlés	—	0 1 0 1 1	CL2
F(12)	nem szabványosított	—	0 1 1 0 0	—
F(13)	fenntartott	—	0 1 1 0 1	—
F(14)	nem szabványosított	—	0 1 1 1 0	—
F(15)	fenntartott	—	0 1 1 1 1	—
F(16)	GP1 felülírás	W	1 0 0 0 0	WT1
F(17)	GP2 felülírás	W	1 0 0 0 1	WT2
F(18)	GP1 szelektív beírás	W	1 0 0 1 0	SS1
F(19)	GP2 szelektív beírás	W	1 0 0 1 1	SS2
F(20)	nem szabványosított	W	1 0 1 0 0	—
F(21)	GP1 szelektív törlés	W	1 0 1 0 1	SC1
F(22)	nem szabványos	W	1 0 1 1 0	—
F(23)	GP2 szelektív törlés	W	1 0 1 1 1	SC2
F(24)	felfüggesztés	—	1 1 0 0 0	DIS
F(25)	végrehajtás	—	1 1 0 0 1	XEQ
F(26)	engedélyezés	—	1 1 0 1 0	ENB
F(27)	állapotvizsgálat	—	1 1 0 1 1	TST
F(28)	nem szabványosított	—	1 1 1 0 0	—
F(29)	fenntartott	—	1 1 1 0 1	—
F(30)	nem szabványosított	—	1 1 1 1 0	—
F(31)	fenntartott	—	1 1 1 1 1	—

GP1: 1-es regisztercsoport

GP2: 2-es regisztercsoport

7.2. táblázat

CAMAC ADATÚT jelei és vezetékeinek száma

	Jelölés	Normál- állomáson	Vezérlő- állomáson	Név
Utasítás	N, ill. N1...N24 A1, A2, ..., A8 F1, F2, ..., F16	1 4 5	24 4 5	állomásszám* alcím funkció
Adat	R1, R2, ..., R24 W1, W2, ..., W24	24 24	— —	olvasóvonalak íróvonalak
Állapot	L, ill. L1 ... L24 B Q X	1 1 1 1	24 1 1 1	LAM* foglaltság válasz utasítás elfogadva
Közös vezérlés	Z C I	1 1 1	1 1 1	alaphelyzet törlés tiltás
Időzítés	S1 S2	1 1	1 1	mintavétel mintavétel
Segéd összekötések	P1, P2 P3...P5, ill. P1...P7	2 3	— 7	szabad buszvezeték segédérintkezők**
Tápfeszültségek	+24 V +6 V -6 V -24 V 0 V	1 1 1 1 2	1 1 1 1 2	} kötelező CAMAC tápfeszültségek } közös föld
Tápfeszültségek	+200 V +12 V -12 V ~117 V (fázis) ~117 V (nulla) E Y1, Y2	1 1 1 1 1 1 2	1 1 1 1 1 1 2	} kiegészítő (NIM) tápfeszültségek } tiszta föld (analog) fenntartott
Vezetékek száma összesen		86	86	

* Egyedi vezetékek

** Szabad érintkezőpontok (ADATÚT huzalozással nincsenek ellátva)

7.3. táblázat

2×43 pontos ADATÚT csatlakozó érintkező kiosztása a normál állomáson és a vezérlőállomáson

Normál állomás		Vezérlőállomás	
P1	B	P1	B
P2	F16	P2	F16
P3	F8	P3	F8
P4	F4	P4	F4
P5	F2	P5	F2
X	F1	X	F1
I	A8	I	A8
C	A4	C	A4
N	A2	P6	A2
L	A1	P7	A1
S1	Z	S1	Z
S2	Q	S2	Q
W24	W23	L24	N24
W22	W21	L23	N23
W20	W19	L22	N22
W18	W17	L21	N21
W16	W15	L20	N20
W14	W13	L19	N19
W12	W11	L18	N18
W10	W9	L17	N17
W8	W7	L16	N16
W6	W5	L15	N15
W4	W3	L14	N14
W2	W1	L13	N13
R24	R23	L12	N12
R22	R21	L11	N11
R20	R19	L10	N10
R18	R17	L9	N9
R16	R15	L8	N8
R14	R13	L7	N7
R12	R11	L6	N6
R10	R9	L5	N5
R8	R7	L4	N4
R6	R5	L3	N3
R4	R3	L2	N2
R2	R1	L1	N1
-12 V	-24 V	-12 V	-24 V
+200 V	-6 V	+200 V	-6 V
~117 V (F)	~117 V (N)	~117 V (F)	~117 V (N)
Y1	E	Y1	E
+12 V	+24 V	+12 V	+24 V
Y2	+6 V	Y2	+6 V
0 V	0 V	0 V	0 V

Az érintkezőkiosztást a keret elejétől nézve adtuk meg.

7.4. táblázat

A CAMAC CSOPORTÚT jeleinek összefoglalása

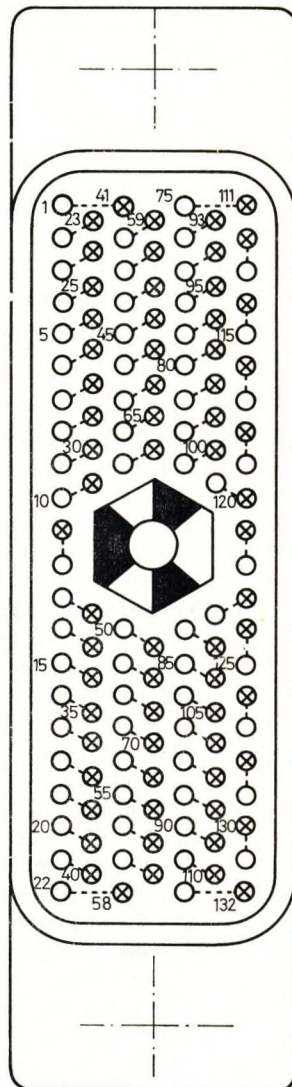
	Jelölés	Buszvonalak száma	Név
Utastítás	BCR1, BCR2, ... BCR7	7	keretcím
	BN1, BN2, BN4, BN8, BN16	5	állomásszám
	BA1, BA2, BA4, BA8	4	alcím
	BF1, BF2, BF4, BF8, BF16	5	funkció
Adat	BRW1, BRW2 ... BRW24	24	olvasó/író vonalak
Állapot	BQ	1	válasz
	BX	1	utastítás elfogadva
Időzítés	BTA	1	„kézfogás”
	BTB1, BTB2, ... BTB7	7	(HAND-SHAKE) jelek
Igénykezelés	BD	1	csoportigény osztályozott LAM kérelem
	BG	1	
Közös vezérlés	BZ	1	alaphelyzet
Fenntartott	BV1, BV2, ... BV7	7	felhasználásukról az ESONE bizottság később dönt
Egyéb	BSC	1	kábelárnyékolás
Buszvonalak száma összesen		66	

7.5. táblázat

A CSOPORTÚT jeleinek kontaktus-elrendezése

(Az ábrán rögzített CSOPORTÚT csatlakozó — Hughes WSS 0132 — előlnézeti képe látható)

Buszvonala	Jel—föld kontaktus	Buszvonala	Jel—föld kontaktus
BCR1	32—13	BRW1	93—76
BCR2	33—14	BRW2	94—77
BCR3	34—15	BRW3	95—78
BCR4	35—16	BRW4	96—79
BCR5	67—50	BRW5	97—80
BCR6	68—51	BRW6	98—81
BCR7	69—52	BRW7	99—82
BN1	36—17	BRW8	100—83
BN2	37—18	BRW9	103—84
BN4	38—19	BRW10	104—85
BN8	39—20	BRW11	105—86
BN16	40—21	BRW12	106—87
BA1	41—1	BRW13	107—88
BA2	23—2	BRW14	108—89
BA4	24—3	BRW15	109—90
BA8	25—4	BRW16	110—91
BF1	70—53	BRW17	112—113
BF2	71—54	BRW18	114—115
BF4	72—55	BRW19	116—117
BF8	73—56	BRW20	118—119
BF16	74—57	BRW21	124—125
BQ	61—44	BRW22	126—127
BTA	63—46	BRW23	128—129
BTB1	31—10	BRW24	130—131
BTB2	11—12	BV1	26—5
BTB3	58—22	BV2	27—6
BTB4	132—92	BV3	28—7
BTB5	123—102	BV4	29—8
BTB6	120—101	BV5	30—9
BTB7	121—122	BV6	64—47
BD	60—43	BV7	65—48
BG	59—42	BX	66—49
BZ	62—45	BSC	111—75



7.2.

CAMAC műszerezés és software decimális osztályozása

- .0 Általánosságok, módszerek, dokumentáció, stb.
 - .1 Adatmodulok
 - .2 Vezérlők (keret, csoport, stb.)
 - .3 Vizsgáló, ellenőrző készülékek
 - .4 Keretek, tápegység, tartozékok
 - .5 Software
- .1 Adatmodulok
- .11 Digitális soros bemeneti modulok (számlálók)
 - .111 Egyszerű bináris számlálók
 - .112 Egyszerű dekadikus számlálók
 - .113 Preszet bináris számlálók
 - .114 Preszet dekadikus számlálók
 - .117 Egyéb soros bemeneti modulok
 - .12 Digitális párhuzamos bemeneti modulok
 - .121 Bemeneti kapuáramkör rendszerek
 - .122 Bemeneti regiszterek
 - .123 Jelátalakítást végző bemeneti regiszterek (koincidenciák, mintaadó generátorok stb.)
 - .124 Kézi bemeneti modulok (kapcsolóregiszterek, paraméteregység stb.)
 - .127 Egyéb párhuzamos bemeneti modulok (interrupt-, LAM- állapotregiszterek).
 - .13 Digitális kimeneti modulok
 - .131 Soros kimeneti modulok (órák, időzítők, impulzusgenerátorok)
 - .132 Párhuzamos kimeneti regiszterek (TTL, HTL, NIM stb.)
 - .133 Párhuzamos kimeneti meghajtók (szabadkollektoros, jelfogós stb.)
 - .14 Digitális ki-, ill. bemeneti egységek, perifériás és műszerezési interface-ek (pl. nyomtató, szalag, digitális voltmérő, léptetőmotor stb.)
 - .141 Soros általános célú ki-, ill. bemeneti regiszterek
 - .142 Párhuzamos ki-, ill. bemeneti regiszterek
 - .143 Perifériás interface modulok (teletype, lyukszalag, stb.)
 - .144 Megjelenítő és rajzoló interface-ek
 - .145 Műszerek interface moduljai (digitális voltmérő, léptetőmotor-meghajtó, tápfeszültség-vezérlő, stb.)
 - .147 Egyéb digitális ki-, ill. bemeneti egységek (beleértve az adatátviteli csatolómodulokat is)

- .15 Digitális adatkezelő és feldolgozó modulok (kapuáramkörök, digitális szint- és kód konverterek, pufferek, késleltetők, aritmetikai processzorok)
 - .151 Kapuáramkörök
 - .152 Digitális szintkonverterek
 - .154 Pufferek, tárolók
 - .155 Logikai és aritmetikai feldolgozó modulok
 - .157 Egyéb digitális kezelő- és feldolgozó modulok
- .16 Analóg modulok (ADC, DAC multiplexerek, erősítők, lineáris kapuáramkörök, diszkriminátorok, stb.)
 - .161 Analóg bemeneti modulok (DC és impulzus analóg—digitális átalakítók, idő—digitális átalakítók, stb.)
 - .162 Analóg kimeneti modulok (digitális—analóg átalakítók)
 - .163 Analóg ki-, ill. bemeneti modulok
 - .164 Analóg kezelő- és feldolgozó modulok I. (multiplexerek)
 - .165 Analóg kezelő- és feldolgozó modulok II. (lineáris kapuáramkörök, diszkriminátorok)
 - .167 Egyéb analóg modulok
- .17 Egyéb digitális, analóg vagy mindkétféle modulok, vegyes analóg digitális, nem ADATÚT-hoz kapcsolódó modulok, stb.

2 Vezérlők

- .21 Interface-ek, meghajtók, vezérlők a párhuzamos 4600-as CSOPORTÚT-hoz vagy egyéb többkeretes párhuzamos buszhoz, egykeretes rendszerekhez, autonom rendszerek
 - .211 Interface-ek, vezérlők, meghajtók többkeretes rendszerekhez I. (4600-as CSOPORTÚT kompatibilis)
 - .212 Interface-ek, vezérlők, meghajtók többkeretes rendszerekhez II. (egyéb párhuzamos utakhoz)
 - .213 Interface-ek, vezérlők, meghajtók egykeretes rendszerekhez (4100 ADATÚT kompatibilis)
 - .214 Autonom működésű vezérlők
 - .217 Egyéb párhuzamos működésű interface-ek, vezérlők, meghajtók
- .22 Interface-ek, meghajtók, vezérlők soros úthoz
- .23 A 4600-as CSOPORTÚT-hoz vagy egyéb párhuzamos úthoz kapcsolódó egységek
 - .231 Keretvezérlők (A1 típusú)
 - .232 LAM osztályozók
 - .233 Lezáró egységek (egyszerűek vagy kijelzővel ellátottak)
 - .234 CSOPORTÚT meghosszabbítók, buszmeghosszabbítók
 - .237 Egyéb CSOPORTÚT- vagy más úthoz kapcsolódó egységek
- .27 Egyéb rendszervezérlők, rendszerek közötti kapcsolatot vezérlők, stb.

.3 Vizsgáló, ellenőrző készülékek

- .31 Rendszerellenőrző egységek
 - .311 Számítógép-szimulátorok
- .32 CSOPORTÚT-tal kapcsolatos vizsgáló, ellenőrző, vezérlő készülékek és megjelenítők
 - .321 CSOPORTÚT ellenőrző vezérlők (kézi, programozott)
 - .322 CSOPORTÚT kijelzők

- .33 ADATÚT-tal kapcsolatos vizsgáló, ellenőrző, vezérlő készülékek és megjelenítők
 - .331 ADATÚT ellenőrzők, vezérlők (kézi, programozott)
 - .332 ADATÚT kijelzők
 - .34 Modulokkal kapcsolatos vizsgáló készülékek (modulmeghosszabbítók)
 - .37 Egyéb CAMAC-kal kapcsolatos vizsgáló berendezés
- .4 Keretek, tápegységek, alkatrészek, tartozékok
- .41 Keretek és hozzátartozó elemek, tartozékok
 - .411 Keretek ADATÚT-tal és táplálással
 - .412 Keretek ADATÚT-tal, táplálás nélkül
 - .413 Keretek ADATÚT nélkül, táplálással
 - .417 Üres keretek és egyéb elemek és tartozékok (pl. ventilátoregységek)
 - .42 Tápegységek és hozzátartozó elemek, tartozékok
 - .421 Többkeretes tápegységek
 - .422 Egykeretes tápegységek
 - .427 Üres tápegység sasszék, tartozékok, elemek
 - .43 Ajánlott és szabványos alkatrészek, tartozékok (CSOPORTÚT kábelek, csatlakozók, üres modulok, stb.)
 - .431 CSOPORTÚT-tal kapcsolatos elemek (kábelek, csatlakozók, stb.)
 - .432 ADATÚT-tal kapcsolatos elemek (Nyomtatott vagy egyéb módon huza-lozott ADATÚT csatlakozók, stb.)
 - .433 Modulal kapcsolatos elemek (üres modulok, átköthető modulok, stb.)
 - .437 Egyéb ajánlott és szabványos alkatrészek, tartozékok
- .5 Software
- .50 Alapvető koncepciók, általános témák
 - .500 Általános leírások, dokumentáció
 - .501 Programnyelvek
 - .502 Algoritmusok
 - .51 Felhasználói programok I. (teljes rendszer háttérrel)
 - .52 Felhasználói programok II. (specifikus futási programok)
 - .53 Felhasználói programok III. (szubrutinok, hardware programok)
 - .54 Támogató software I. (transzlátorok)
 - .541 Assembler-ek (makróval vagy anélkül)
 - .542 Cross-Assembler-ek
 - .543 Fordítóprogramok
 - .544 Interpreter-ek
 - .55 Támogató software II.
 - .551 Betöltő programok
 - .552 Összeállító programok (Linking programs)
 - .553 Kiszolgáló programok (Utility routines)
 - .57 Egyéb segédprogramok
 - .571 Editor-ok
 - .572 Hibakezelő rutinok (Debugging routines)
 - .573 Ellenőrző programok (Test routines)

PROGRAMOZÁS ÉS ELEKTRONIKA

sorozat

SZERK.: DR. TARNAY KÁLMÁN

A számítástechnikai kultúra fejlesztése érdekében a Műszaki Könyvkiadó indította a Programozás és elektronika című sorozatot. E sorozat célja az elektronika különböző jellegzetes számítási, tervezési és dokumentálási feladatainak tárgyalása, a korszerű programozási módszerek ismertetése. Az egyes módszerek alkalmazását — rövid elvi ismertetés után — gyakorlati feladatok megoldásán keresztül is bemutatja. Az egyes kötetek Függelékben közlik a gyakorlatban kipróbált programokat. Ezzel a sorozat egy olyan programkönyvtárat biztosít, melyet minden bizonnyal jól használhat a termelés és kutatás területén dolgozó elektronikai, híradástechnikai, műszeripari és számítástechnikai szakember, valamint az egyetemen, főiskolán az oktató és a hallgató egyaránt.

A sorozat eddig megjelent kötetei:

Székely—Tarnay: A programozás alapjai

Székely—Benkőné: Karakterisztikák — diagramok — nomogramok

Herendi—Tarnay: Egyenáramú hálózatok analízise

Ambrózy—Jávor: Mérésadatok kiértékelése

Előkészületben levő kötetei:

Csáki: Lineáris szabályozási rendszerek analízise

Csáki: Lineáris szabályozási rendszerek szintézise

Benkő—Jávor: Diszkrét rendszerek szimulációja

Zombory—Koltay: Elektromágneses terek

Mészáros: Folyamatidentifikáció

Műszaki vezető: Hegedűs Ernő — Műszaki szerkesztő: Molnár József
Borító és kötéstervező: Killer Marcella
A könyv ábráit rajzolta: Olgyay Géza
A könyv formátuma: B/5 — Ívterjedelem: 15,75 (A5)
Ábrák száma: 83 — Példányszám: 4000
Papír minősége: 100 g. Delta — Betűcsalád és -méret: gm/gm Gill monó
Azonossági szám: 60 689 — MŰ: 2324—k—7679
Készült az MSZ 5601—59 és 5602—55 szerint
75-3687 — Szegedi Nyomda

**CSÁKÁNY ANTAL—
DR. VAJDA FERENC
MIKROSZÁMÍTÓGÉPEK**

A könyv a számítástechnikában ma már igen elterjedten alkalmazott mikroszámítógépeket ismerteti. A mikroszámítógépek és az azokat alkotó mikroprocesszorok fogalmának meghatározása után tárgyalja a mikroszámítógépek gyártásának alapját képező technológiai eljárásokat, majd bemutatja a jellegzetes mikroszámítógép elemeket. Ezt követi a különböző típusú — mikroprocesszor elemekből felépített — mikroszámítógépek ismertetése. Az utolsó fejezetben alkalmazási példák találhatók.

A számítógéptervezők és alkalmazók, rendszertervezők, digitális technikával és szabályozástechnikával foglalkozó szakemberek számára készült könyvből a mikroszámítógépek iránt általánosan érdeklődők is sok hasznos ismeretet kaphatnak.

ÁRA: 45,— FT