

tanulmányok

112 / 1980

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADEμία
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

TÖBBPROCESSZOROS, OSZTOTT INTELLIGENCIAJÚ GRAFIKUS RENDSZEREK
TERVEZÉSI ÉS MEGVALÓSÍTÁSI KÉRDÉSEI

Kandidátusi értekezés

Irta:

Verebély Pál

Tanulmányok 112/1980.

A kiadásért felel:
DR VAMOS TIBOR

ISBN 963 311 109 9
ISSN 0324-2951

Készült a
SZÁMOK Reprográfiai Üzemében 80/204

TARTALOMJEGYZÉK

I.	Bevezetés. A disszertáció felépítése:	5
II.	Célkitűzések	7
III.	Történelem: azaz a GD'80 család helye a világban és a hazai grafikában	11
IV.	Előtanulmányok többprocesszoros grafikus rendszerek létrehozására	23
V.	Az elvégzett munka, azaz a GD'80 rendszertechnikájának ismertetése	27
VI.	Új tudományos eredmények	58
VII.	Az ismertetett tudományos eredmények realizációja és referenciái	66
VIII.	Az értekezés témaköréből készült publikációk jegyzéke	68
IX.	Irodalomjegyzék	73
X.	Függelék /ábrák gyűjteménye/	97

I. BEVEZETÉS. A disszertáció felépítése.

A konvencionális megoldásokban egy többfeladatú nagyteljesítményű kisszámítógép oldotta meg az interaktív grafikus display-rendszerek különböző funkcióit: a grafikus bemeneti eszközök kezelését, a felhasználói programok futtatását, a képmegjelenítő programnak egy magasabb szintű geometriai leírásból vagy matematikai modellből történő előállítását, a kép transzformációját és a kommunikációt más számítógépekkel. A real-time igények kielégítésére a vezérlő kisszámítógép sebességét növelték, például kiegészítő speciális hardware elemek beépítésével.

Az értekezés egy ugyanerre a célra szolgáló, merőben eltérő, új megoldású grafikus berendezés tervezését és megvalósítását írja le. A funkciók - a rendszer intelligenciája - a több egymással kommunikáló mikroprocesszorból, a hozzájuk rendelt saját és közös memóriákból, a perifériákból és a megjelenítő egységekből álló rendszer egyes elemei között oszlanak meg.

A disszertáció a következő főbb fejezetekből áll:

Az első fejezet a témát indokolja és ismerteti a disszertáció felépítését.

A második fejezet ismerteti az elvégzett munka célkitűzéseit, a harmadik pedig a GD'80 helyét jelöli ki a hazai és a nemzetközi grafikában.

A negyedik fejezet a témában folytatott előkészítő munkákat írja le.

Az ötödik fejezetben ismertetem az elvégzett munkát, a GD'80 rendszertechnikai kialakítását.

A hatodik fejezet az értekezés új tudományos eredményeit,
a hetedik fejezet a GD'80-ra vonatkozó referenciákat foglal-
ja össze.

A nyolcadik fejezet az értekezés témakörében készült publi-
kációk listáját,
a kilencedik fejezet pedig az irodalomjegyzéket tartalmazza.

II. CÉLKITŰZÉSEK

A GD'80 projekt körvonalait, célkitűzéseit - a többprocesszoros rendszerek és grafikus berendezések területén végzett előzetes tanulmányok és előkészületek után - 1976 decemberében az MTA SzTAKI Gépipari Automatizálási Főosztály keretében rögzítettük. 1977 kezdete óta ez a projekt az akkor megalakult Számítógépes Grafika Osztály fő témája. A célkitűzéseket az alábbi pontokban foglaltam össze /bár ezek között nyilván átfedések is vannak/.

1. Teljesítőképesség, szolgáltatások, megbízhatóság és árfekvés szempontjából a GD'80 rendszereknek a piacokon tervezett megjelenésük időpontjában - 1980-ban - Nyugaton /USA-ban is/ és Keleten egyaránt versenyképeseknek kell lenniük.
2. A GD'80 interaktív vektorrajzoló grafikus displaycsalád legyen. Ez azt jelenti, hogy a széleskörű alkalmazási igényeket mind grafikus, mind adatfeldolgozási szempontból egyaránt ki tudja elégíteni.
3. A display család egyes tagjai közös moduláris architektúrával rendelkezzenek, azaz egy adott elemkészletből összeállítható legyen a legegyszerűbb és a legnagyobb teljesítőképességű családtag is, még olyan áron is, hogy esetleg egyes családtagokat a közös architektúránál célszerűbb megoldásokkal is kialakíthatnánk.
Az egyes családtagok a másiktól hozzáadással, illetve elvétellel legyenek nyerhetők, azaz a rendszer modulárisan bővíthető is legyen.
4. A GD'80 elemkészletből legyenek kialakíthatók külső számítógéphez csatolt és önálló rendszerek is, az utóbbiakra helyezve a nagyobb hangsúlyt.

5. Az intelligencia legyen dedikált mikroprocesszorokra szétosztva a grafikus display rendszereknél szokásos többfeladatú vezérlő miniszámítógép helyett, amely a periférikezelést, a display listának egy magasszintű - felhasználói - képleírásból vagy matematikai modellből történő előállítását, a transzformációkat, a felhasználói programok futtatását és a kommunikációt végezte.
6. A mikroprogramozást alkalmazzuk általános és speciális utasításrendszerek megvalósítására, valamint geometriai műveletek nagysebességű elvégzésére.
7. Nagy-és kisméretű, színes és egyszínű képernyők legyenek alkalmazhatók.
8. Az egyes családtagokhoz grafikus bemeneti eszközök és konvencionális perifériák széles választéka legyen csatolható.
9. Önálló alkalmazások esetén a háttértárolók kapcsolódási lehetősége legyen biztosított.
10. Hardware és software kompatibilitás álljon fenn
 - 10.a. Interface kérdésekben /RS232-C, CCITT V24, ESZR/IBM, MSZR/DEC, HP-IB/IEEE 488, stb./
 - 10.b. Adatátviteli protokollokban /BSC, HDLC, SDLC, DDCMP, stb./
 - 10.c. Grafikus csomagokban /GPGS, GINO, GTU, CORE/
 - 10.d. Teljes rendszerek kialakításában /display rendszerek emulációja/

11. A GD'80 család legyen kiterjeszthető más technikájú, például tárolócsöves, rászteres, plazma, félvezetős, stb. kijelzőkre is.
12. A felhasznált elemválaszték /integrált áramkörök, csatlakozók, stb./ lehetőleg minél kevesebb gyártóra korlátozódjék és kövesse a szocialista országok hosszútávú gyártási terveiben szereplő listákat. Előnyben részesítendőek a több gyártóval /second source/ rendelkező és a már elterjedt típusok.
13. A GD'80 család modularitása és kompatibilitása vonatkozzék a mechanikai kialakításra is, különös tekintettel a bővíthetőségre és a gyárthatóságra.

A célkitűzések részletezését és pontosítását a fejlesztési munka során végeztük el.

A GD'80 grafikus displaycsalád kifejlesztése az MTA SzTAKI Számítógépes Grafika Osztályának közös munkája volt. Ezen belül az én munkám a teljes hardware architektúra és az egyes konfigurációk kialakítása volt.

III. TÖRTÉNELEM: azaz a GD'80 család helye a világban és a hazai grafikában.

A grafikus display-ek az ember és a számítógép közötti kommunikáció legfejlettebb eszközei, amelyek egyrészt a számítógép közlendőit grafikus formában jelenítik meg a képernyőn, másrészt elfogadják, feldolgozzák azokat az akciókat, amelyeket az ember az interaktív bemeneti eszközök segítségével kezdeményez.

A grafikus display-ek sok szempontból csoportosíthatók. Ezek közül csak a legfontosabbakat emelem ki.

1. A megjelenítés módja szerint alapvetően két típuscsoportba sorolhatjuk a grafikus display-eket /1. ábra/. Ezek közül az egyik a vonalarajzolós display-ek kategóriája, amelynek fő jellemzője a random rajzolás/ random rajzolás: a sugár bármely időben a képernyő tetszőleges pontjára mozgatható/. A másik kategóriába a raszteres elvű megjelenítők tartoznak, amelyek a képet sorokra, a sorokat pedig pontokra bontva ábrázolják. Hasonló felosztás tehető egyébként rajzgépek esetén is: vonalarajzolós rajzgépek, illetve printer-plotterek / printer plotterek: olyan rajzgépek, melyek egyszerre egy rásztersornyi információt rajzolnak ki; a kirajzolt kép az ilyen rásztersorok összessége/. Ez utóbbi kategória felbontóképessége a technika mai állása mellett körülbelül négyszer kisebb, mint a vonalarajzolós típusoké.

Mindkét kategória további csoportokra bontható. A vonalarajzolós display-ek frissítő és tárolócsöves változatokra oszthatók. Az előbbi esetében a képgenerálást állandóan ismételni kell, hogy állóképet kapjunk, az utóbbinál viszont elegendő egyszer felrajzolni a képet. A frissítő típusú rendszerek

előnye az interaktivitás magas foka, a gyors képmódosítás, lehetősége, hátránya az állandó képgenerálás, az ehhez szükséges memória foglалás és a véges megjeleníthetőség, amelyet a kép villódzása korlátoz. A tárolócsöves rendszerek előnye a villódzásmentes, szinte tetszőleges bonyolultságú kép megjelenítési lehetősége és a csekély memória foglалás, hátránya a csökkent interaktivitás /például a fénytoll használata lehetetlen, vagy legalábbis nagyon korlátozott/ és a képmódosítás körülményes volta. A legutóbbi időkben - főleg a Tektronix cég révén - eredmények születtek a két vonalrajzolós rendszer előnyös tulajdonságainak ötvözésére: a tárolócsöves display-ek részleges törlési lehetősége, "write thru" üzemmódja, több foszforréteg alkalmazása, stb. A raszteres megjelenítők alcsoportjai abban különböznek egymástól, hogy mi az egyes képpontok hordozója: a video jel által vezérelt felvillanás a katódsugárcsővön, plazmakisülés a plazmapanelen, vagy a világítódioda által kibocsátott fény. A GD'80 család fejlesztésének első ütemében a vonalrajzolós, frissítő típusok kialakítása volt a célunk, minthogy a tervbe vett alkalmazások többsége erősen interaktív, a tárolócsövek és a raszteres kijelzőkhöz szükséges nagymennyiségű memória beszerzése nehézkes. A memóriaárak drasztikus csökkenése és a nagy felbontóképességű raszteres monitorok megjelenése természetesen rövid időn belül szükségessé teszi a GD'80 raszteres kijelzőt felhasználó változatainak kialakítását is.

A megjelenítés módjához tartozik a színes kijelzés lehetősége. A vonalrajzolós grafikus display-eknél a színes kijelzést több - /általában két/ - foszforréteggel ellátott /penetrációs/ katódsugárcsővekkel oldják meg. A különböző színeket az anódfeszültség átkapcsolásával és ezáltal különböző

foszforrétegek gerjesztésével érik el. Több szín is kikeverhető közbenső feszültségek beállításával; például, ha a két alapszín: piros és zöld, a keverékszínnek: sárga és narancssárga. Természetesen nagy problémát jelent a nagyfeszültség gyors és pontos átkapcsolása, valamint az eltérítő, video- és fókuszerosítók szimultán kompenzálása.

2. A megjelenítés minőségének összehasonlítására sok technikai paraméter megadása szokásos [47, 50, 38, 28, 112, 124, 161, 166, 167]. Ezek közül a legfontosabbak a következők:

2.1. Képernyőméret: kerek, illetve négyszögletes kijelzők szokásosak, az előbbinél az átmérőt, az utóbbinál a két irányú méretet kell megadni. Különböző alkalmazásoknál a kép részletgazdagsága különböző szerepet játszhat, így a 100x100 mm-től a 650 mm-es átmérőig a kijelzők széles választéka található ma a világpiacon.

2.2. A vektorgenerátor /amely egyenesszakaszok megjelenítésére alkalmas/ jellemzői:

2.2.1. Felbontóképesség: ez a számpár azt mutatja, hogy az egyes főirányokban hány egymástól megkülönböztethető pont jeleníthető meg a képernyőn. A maximális érték hosszú ideig az 1024x1024 mátrix volt, ma már a vezető amerikai típusok többségében a 2048x2048, illetve a 4096x4096 a katalógusadat.

- 2.2.2. Generálási sebesség: ezt a jellemzőt vagy a rajzolási sebesség /cm/mikroszek/, vagy a maximális hosszúságú vektor generálási idejével kell megadni. A világ élvonalában ma ez a szám 30 mikroszek /max. vektor, 12 bit/.
- 2.2.3. Intenzitási szintek: ez a szám a képernyőn megkülönböztethető fényességi szinteket jellemzi /4-16 szint a tipikus érték/.
- 2.2.4. Vonalféleségek: a megjeleníthető vonaltípusok /folytonos, szaggatott, pontozott, stb./ számát jelenti /4 jellemző a világpiacon/.
- 2.2.5. Magasabbrendű görbék generálása: kör, kúpszeletek, stb., generálását lehetséges hardware úton megoldani, de viszonylag kevés az ilyen megoldás, főleg azért, mert egyrészt a transzformációk egyenesekről magasabbrendű görbékre nem terjeszthetők ki kellő általánossággal, másrészt az alkalmazások mintegy 90 %-ában kielégítő a vektorok, illetve vektorokból összerakott görbék generálása.
- 2.2.6. Transzformációknak nevezzük a kép/vagy annak egy része/ eltolását, elforgatását, léptékváltoztatását, kivágás készítését és a perspektív ábrázolásához szükséges manipulációkat. Szokásos megoldások: komplex hardware és/vagy grafikus szubrutincsomagok.
- 2.3. A karaktergenerátor /betűk, számok, szimbólumok és speciális jelek megjelenítésére szolgál/ jellemzői:
- 2.3.1. A karaktergenerálási mód jellemzi azt az elvet, amelynek segítségével a karaktert megrajzoljuk. A display-ek döntő többségében pont-mátrixból,

rövid vektorokból, illetve szinusz-koszinuszokból összeállított karakterképek használatosak.

2.3.2. Karaktergenerálási idő: a csúcsteljesítmény 4 mikroszek/karakter.

2.3.3. Karakterek választéka: 128 ASCII karakter a szokásos és további opciókkal bővíthetők a világszerte display-ei.

2.3.4. Karaktertranszformációk: általában a vektorok transzformációjától függetlenül zajlanak. Írásirány, forgatás, sűrűség, döntés, stb. a jellemző lehetőségek.

A megjelenítés minősége szempontjából a GD'80 család minden fő egységből két típus beépítését teszi lehetővé: az egyik a világszerte csúcsteljesítményt nyújtja, a másik pedig a közepes teljesítményű variáns.

3. A grafikus display-rendszerek számítástechnikai háttere, a jellemző konfigurációk szempontjából a grafikus display-ek ciklus fejlődést mutatnak [125]. A ciklikusságot az adja, hogy a tervezők az adott időpontban melyik fő egység: a memória, vagy a processzálást végző egység költségét akarták csökkenteni /ez elsősorban az adott helyzet árviszonyaitól függött./ Az első grafikus display-ek az akkori nagyszámítógéphez kapcsolódtak, a megjelenítendő kép leírását a nagy gép memóriájában tárolták /2. ábra/ [168]. Később az átviteli csatorna sebességigényének csökkentésére a display mellé kihelyeztek egy saját, lokális memóriát /3. ábra/ [18].

A következő lépés az volt, hogy egy speciális célhardware vezérlőegységet vagy egy miniszámítógépet is a terminálba építettek, hogy egyrészt tovább csökkentsék az adatátviteli sebességet /például magasabbszintű, tömörebb leírások - nem közvetlen display kód - átküldését téve lehetővé/, másrészt az interaktív grafikus bemeneti eszközök közvetlen, fizikai kezelésének real-time gondját levegyék a nagy gép válláról /4. ábra/ [27, 53, 54, 69]. Ezek után a miniszámítógépen már a felhasználói program vagy annak részei futtatására is lehetőség nyílt. A következőkben a transzformációk és egyéb grafikus manipulációk elvégzésére a memória és a display megjelenítő közé egy ún. display processzort, speciális nagysebességű egységet /5. ábra/ építettek be [19, 59, 62, 63, 87, 113, 126, 141, 162, 163, 190, 194]. Ennek legfejlettebb változata a 6. ábrán látható, amelyen a display processzor és a megjelenítő közé egy újabb memória buffer kerül. Ezt a szakirodalomban bufferelt display processzornak nevezik [2, 64, 191] /buffer: közben-ső tároló/. Ennek a megoldásnak az az előnye, hogy a képmegjelenítés adatátviteli sebességénél 3-5-ször kisebb át-bocsájtóképességű display processzorral is real-time képmanipulációk érhetőek el/ real-time képmanipulációk: a képváltoztatás olyan sebességű elvégzése, hogy az a display kezelője számára érzékszerveinek reakcióidején belül történjék/. Egy további megoldás az, amikor a minigép és a display processzor egyes funkciói több processzor, vagy mikroprocesszor között oszthatók meg. /7. ábra/.

Ennél a megoldásnál az adatáramlás iránya kevésbé írható le egyenesirányú nyilakkal mint a 2.-6. ábrákon /hozzátéve, hogy azokban az esetekben is csak nagyvonalakban igazak a jelfolyamok/: az adatok a közös memórián keresztül "cserélnek gazdát /processzort/".

A GD'71 grafikus display-t intézetünkben 1971-ben fejlesztettük ki. Kategóriáját tekintve vonalrajzolós, képfriessítő típusú rendszer, amely a 4. ábrán látható elveket követi. A GD'71 első kiviteli alakjában a 3. ábrán látható rendszerhez volt hasonló. Ilyen változat a közelmúltban is működött a MAVEMI Gier gépével. A GD'71 display-eket több miniszámítógéphez csatlakoztattuk: CII 10010 /VT 1010B/, TPA 70, EC 1010 /VT 1010/, KRS 4200/4201. A 8. ábrán látható TPA 70-es változat alkalmazása volt a legszélesebb körű a számítógéppel segített tervezés különböző területein /nyomtatott áramköri lapok tervezése és ellenőrzése, integrált áramkörök maszkjainak tervezése, kivágó szerszámok tervezése, numerikus vezérlésű szerszámgépek programjainak ellenőrzése, stb./. A rendszernek a további kisszámítógépekhez való csatlakoztatásában problémák merültek fel: egyrészt ezen gépek közvetlen meóriáhozáférésének nehézségei, másrészt a software-újraírás miatt /nem volt hordozható, gépfüggetlen/.

Az utóbbi évek során a félvezető gyártás területén, főként a memória és mikroprocesszor technikában történt robbanásszerű változások lehetővé tették, hogy a GD'80 rendszerrel a 7. ábra szerinti osztott intelligenciájú rendszert valósítsuk meg.

4. Egyéb hazai és más szocialista országokból származó grafikus display rendszerek sorában a vonalrajzolós display-rendszerek közül a GD'71 és a GD'80 család egyaránt egyedülálló saját fejlesztésű rendszernek tekinthető. Ugyanis az ESZR-ben elfogadott EC 7064 típus egy több mint tízéves amerikai típus másolata /IBM2250 Model 2/. A szovjet EPG-400 /amely az MSZR első sorozatának elfogadott grafikus eszköze/ és a VIDEOTON által kidolgozott VGDS a

viszonylag kisteljesítményű Digital Equipment GT40 /VT11-PDP11/ rendszerrel mutat azonos tulajdonságokat.

A raszteres technikájú grafikus eszközök területén elsősorban testvérintézetünk, az MTA KFKI MSzKI játszott úttörő szerepet. Három eddig kifejlesztett típusa közül az első egy viszonylag kis felbontóképességű /256x256/, minden egyes képponthez a vezérlő TPA-i memóriájában egy bit információt hozzárendelő display. További két rendszerük már mikroprocesszor bázisú, a kép "karaktermezőkre" van felbontva, amely karaktermezőkben megjelenő kép /pixel/ programozható. A PE-12F típus egy színű, az NE 657 pedig színes kijelzőt használ.

E típusok elsősorban ipari folyamat- és termelésirányítás kijelző berendezéseiként kerülnek felhasználásra. Az SzKI-ban kifejlesztett CDP-3 típusu berendezés elveiben a KFKI "pont=bit" típusával rokon, azonban színes kijelzőt használ és vezérlőgépe mikroprocesszor bázisú /MO51/. Információink szerint a VIDEOTON által kifejlesztett új alfanumerikus display családban is lesz kvázigrafikus családtag /a KFKI PE 12F, illetve NE657 típusokkal megegyező technikai paraméterekkel és célokkal/, hasonlóképpen, mint a Hewlett Packard 264X sorozatában. A GD'71 a maga korában és a GD'80 ma az egyetlen szocialista fejlesztésű grafikus eszköz, amely az interaktív számítógéppel segített tervezés világszinvonalának megfelelő architektúrával és paraméterekkel rendelkezik. Ezt a tényt azért hangsúlyozom, mert az ilyen kategóriájú nyugati display-ek a katonai és a gyorsabb technikai fejlődést lehetővé tevő, számítógéppel segített tervezési területen történő alkalmazási lehetőségük miatt embargó alatt állnak, közvetlen beszerzésük nem lehetséges. Ugyanakkor viszont a GD'80 technikai jellemzői tőkés piacon is versenyképes terméké teszik ezt a berendezést.

5. A számítógépek architektúrájának és az áramköri technológiának a fejlődése napjainkban ugrásszerűen meggyorsult. A konvencionális felépítésű számítógépek mellett egyre inkább teret nyernek az új architektúrájú megoldások, különös tekintettel a párhuzamos műveletvégrehajtásra [4, 24, 25, 43, 91, 99, 102, 129, 176]. A multiprocesszoros /több műveletvégző egységet tartalmazó/ rendszerek konvencionális processzorokból /pl. miniszámítógépekből/ épülnek fel és bizonyos feladatok végrehajtása során sokszor elérik, vagy túlhaladják a nagyszámítógépek teljesítőképességét. A multiprocesszoros rendszerek kialakításához különböző módszereket és technikákat fejlesztettek ki [1, 22, 39, 48, 61, 65, 93, 120, 205]. A mikroprocesszorok megjelenése a többprocesszoros rendszerek fejlesztését is forradalmasította, éppen a processzor árak rohamos csökkenése révén. A több mikroprocesszort tartalmazó rendszerekkel foglalkozó irodalom egy része az ilyen rendszerek elvi felépítésén és jövőbeli esetleges szerepének megítélésén általában nem megy túl [11, 15, 77, 83, 144, 145, 159, 170, 199, 201, 204]. Ismeretes azonban, néhány már megvalósított alkalmazás is [16, 58, 101, 121, 178].

Az egymáshoz szorosan, nagysebességű csatornák segítségével /20-100 Mbit/sec/ kapcsolódó multiprocesszoros rendszerek mellett megjelentek az ugynevezett helyi adatátviteli hálózatok [8, 9, 10, 30, 74, 86, 117, 142, 200], amelyek egy nagyságrenddel lassabb kommunikációt tesznek lehetővé az egyes processzorok között /0,1-10 Mbit/sec/. Ugyanakkor a processzorok térbeli elhelyezkedése kevésbé kötött /néhány száz méteres távolságok hidalhatók át/.

A helyi adatátviteli hálózatok fejlesztését is nagymértékben segíti a technológiai fejlődés. Ez részben a nagy integráltsági fokú, nagysebességű kommunikációs áramkörök megjelenésére vezethető vissza /ez utóbbiak több esetben a mikroprocesszoroknál 3-5-ször bonyolultabbak/. A másik technológiai tényező az adatátviteli sínek fejlődéséből adódik, különös tekintettel a koax-kábelek, valamint legújabbban a fényvezető technika elterjedésére. A többprocesszoros módszerek elterjedése mellett a mai számítástechnika fejlődésének egyik jellemzője a mikroprogramozás ismételt térhódítása. A mikroprogramozás maga már jóval régebbi keletű, mint a többprocesszoros rendszerek elmélete és gyakorlata. Az elmúlt években azonban számos új eredmény született ezen a téren. Ezek közül az egyik legfontosabb az írható mikroprogramtárak /Writable Control Store/ kifejlesztése volt [35, 81, 164, 198]. Ez lehetővé tette a mikroprogram szintű diagnosztika kiterjesztését, bizonyos nagysebességigényű feladatoknak a mikroprogramozás szintjén való megoldását, valamint legfőképpen a felhasználói mikroprogramozást /user microprogramming/. Ez utóbbi biztosítja azt, hogy a felhasználó maga alakítsa ki utasításrendszerét és ezt programfutás közben is módosíthassa [76, 82, 118, 151, 158, 174, 174, 182, 184]. A felhasználói mikroprogramozásnak kialakultak a software és hardware fejlesztési eszközei is [21, 153, 179, 192, 193]. A mikroprogramozás fejlődésére szintén nagy hatással volt az új áramkörök megjelenése. Ez a mikroprogramozható mikroprocesszorok [153, 195] és főként a bit-szelet /bit-slice/ elemcsaládok térhódításával magyarázható [5, 6, 6, 12, 13, 14, 29, 41, 45, 111, 114, 115, 130, 146, 147, 169, 183, 185, 196, 197].

A különböző bit-slice elemcsaládok lehetővé teszik teljes mikroprogramozható processzorok felépítését. Ez attól függetlenül igaz, hogy a megcélzott processzor általános célu vagy pedig speciális rendeltetésű.

A fentiekben ismertetett fő motívumok mellett néhány további, kisebb jelentőségű eredmény is hatást gyakorolt a GD'80 architektúrájának kialakítására. A mini számítógépekben is megjelentek a perifériakezelő /mikro/processzorok [57, 137] és a külső aritmetikai processzorok [52, 136]. Az áramkörök választékában ma már megtalálhatók a nagysebességű, aritmetikai orientált elemek, különös tekintettel a párhuzamos szorzóművekre [33, 132, 171, 186, 203]. Az aritmetika orientált processzorokból felépültek a nagymennyiségű adat nagysebességű kezelésére szolgáló array-processzorok /number cruncherek/, amelyek általában valamilyen nagy-, vagy óriásgép perifériájaként kerülnek alkalmazásra.

IV. ELŐTANULMÁNYOK TÖBBPROCESSZOROS GRAFIKUS RENDSZEREK LÉTREHOZÁSÁRA

Eddigi munkám során több feladat megkivánta processzorok egymáshoz rendelését, összekapcsolását. 1973 óta a KFKI által kifejlesztett TPA'70 kisszámítógép volt az a rendszer, amelyre fejlesztéseket végeztem. Ugyanis architektúrája és utasításrendszere következtében a TPA'70 rendszer különböző feladatokra egyszerűen adaptálható. Megjelenése pillanatában ez a kisszámítógép meglehetősen szegényes kiépítettségű volt. Számos periféria illesztőegység készült ehhez a rendszerhez. Közülük azonban csak a többprocesszoros rendszerek megvalósítását lehetővé tevő elemeket ismertetem.

Ezek közül a legegyszerűbb egy BSI rendszerű, byte-os átvitelt nagyobb /300 m/ távolsáig biztosító interface. Ezt kisszámítógépeknek intézetben belüli összekötésére, DNC szerzőszámgepvezérlési rendszerekben az MTC elnevezésű szerzőszámgep vezérlőegységek és a központi kisszámítógép összekapcsolására használtuk. Az átvitel programozott úton történik, azaz minden egyes byte kivételéhez, illetve fogadásához egy megszakítási alprogramnak kell lezajlania. Ennek a rendszernek a sebessége kb. 10 kbyte/sec.

A TPA'70 buszán a GD'71 grafikus display vezérlőegysége egy másik processzor, amely speciális képi utasításrendszert hajt végre végtelen ciklusban, így álló kép megjelenítését biztosítja a képernyőn. A képi programot a vezérlőegység a kisszámítógép memóriájából közvetlen memóriáhozáféréssel olvassa ki, így a központi egységben és a vezérlőegységben futó programok egymással párhuzamosan működhetnek. A programok közötti kommunikációt megszakítások, illetve a vezérlőegység státuszának megváltoztatása biztosítják.

Hasonló szerepe van a diszk vezérlőegységnek is. Ennek az intelligenciája azonban lényegesen kisebb. Parancsokat hajt végre, adatokat mozgat a memória és a háttértároló kiválasztott területei között, ugyancsak közvetlen memória hozzáféréssel, a központi egység /parancs adásán és a végrehajtás nyugtázásán kívüli/ beavatkozása nélkül.

Több TPA'70 busza közötti közvetlen kapcsolat kiépítésére terveztük meg a buszkapcsoló áramkört /9. ábra/. Ez a rendszer képes programozott úton 16 kismámítógép egyikére kapcsolni egy közös busz perifériáit és/vagy memóriaegységeit. Felhasználása elsősorban osztott intelligenciájú és nagy megbízhatóságú rendszereknél lényeges. Erre mutat példát a 10. ábra, amelyen egy CAM/CAD rendszer számítógépei osztják meg két buszkapcsoló áramkör segítségével a háttértárolót. Esetünkben ez egy mágneslemezegység. Ez az adatbázisok cseréjét teszi lehetővé, például termelésirányítási célokra. A szerszámgépvezérlő egységek real-time kiszolgáló perifériáinak megosztása arra szolgál, hogy a CAM gép meghibásodása esetén a CAD gép átvehesse a real-time feladatot, a szerszámgépek vezérlését.

Egyéves USA-beli tartózkodásom idején a Brown University grafikus rendszerének miniszámítógépéhez /Digital Scientific META4/ terveztem, építettem és programoztam egy mikroprogramozott lebegőpontos processzort, amely az IBM 360 megfelelő utasításait hajtotta végre. /11. ábra/. Ez az egység a fixpontos központi egység mikroprogramja számára volt elérhető. A lebegőpontos műveletek adatszavait, valamint az eredményeket a mikroprogram adta át a lebegőpontos processzornak, illetve továbbította onnan a memóriába. Így az egység, hasonlóan a PDP11/45 központi egység - lebegőpontos processzor kapcsolatához, lényegében a META4 CPU aritmetikai kiegészítésének tekinthető.

A TPA '70 rendszerre hasonló célú lebegőpontos egységet terveztem és valósítottam meg munkatársaimmal együtt. Minthogy azonban a TPA '70 központi egysége nem mikroprogramozott és abba belenyúlunk nem lehetett, a lebegőpontos processzort /FPU/ közvetlenül a buszra illeszkedő, önálló processzorként kellett kialakítanunk. A lebegőpontos processzor indítását a központi egységnek szóló illegális utasítások nyomán fellépő megszakítási alprogramok, leállítását pedig a lebegőpontos program számára nem értelmezett utasítások végrehajtásával érjük el. A két processzor bevárhatja egymást /WAIT-mód/, vagy egymással párhuzamosan is futhat. Ez utóbbi elősegítésére a lebegőpontos utasításokon kívül további vezérlésadó utasításokat is mikroprogramoztunk. Mivel a két processzor interakciója mindig megszakításkérések kezelését is magában foglalja /ami időigényes feladat/, célszerű a programokat úgy szervezni, hogy a lebegőpontos processzor hosszabb programrészeket egyvégtében hajtson végre és komplexebb műveleteket is végezzon. Így például ez az egység lebegőpontos szög- és logaritmus-függvényeket is kifejt - a mikroprogram szintjén.

A TPA '70 rendszerhez egy további TPA '70 - MIPROBUS WINDOW nevű egységet terveztem és valósítottam meg, amely alkalmas standard 8 bites mikroprocesszorok belső adatátviteli sínje és a kiszámítógép busza közötti közvetlen memória - memória és megszakítási kapcsolatok létrehozására. A berendezés memóriablakot biztosít a két busz között, azaz lehetővé teszi a memóriák közös használatát. A mikroprocesszor leveszi a fizikai periféria kezelés gondját a kiszámítógép vállairól és közvetlenül a memóriából veszi, illetve ide rakja a perifériák adatblokkjait. További lehetőségei elsősorban a mikroprocesszorok hardware és software fejlesztését segíti elő a mikroprocesszor programok közvetlen betöltése, tetszőleges címen való megállítása, nyomkövető futtatása útján.

A GD'80 család két alapvető processzortípusa hasonló architektúrával rendelkezik, mint a TPA'70 lebegőpontos processzor és a TPA'70 MIPROBUS WINDOW. E két berendezés tapasztalatai nagymértékben befolyásolták a GD'80 rendszertechnikai kialakítását.

V. AZ ELVÉGZETT MUNKA, AZAZ A GD'80 RENDSZERTECHNIKÁJÁNAK ISMERTETÉSE

A GD'80 család rendszertechnikai kialakításánál az egyes grafikus funkciók szétválasztása és a mikroprocesszorok adott célra történő dedikálása volt a fő cél. A képmegjelenítésre a Display Control Unit /DCU/, az interaktív grafikus perifériák kezelésére a Graphic Peripheral Controller /GPC/, a kommunikációs feladatok ellátására a Host Interface /HIF/, a képi program előállítására a Display Processing Unit /DPU/, a geometriai transzformációk elvégzésére a Transformation Processing Unit /TPU/ szolgál. Ez utóbbi kettő a felhasználói programok futtatása során a konvencionális miniszámítógép és annak lebegőpontos aritmetikai kiegészítése szerepét játssza.

1. Buszrendszer és közös memória

A GD'80 család általános architektúráját a 14. ábrán mutatom be. A rendszer két általános célú adatátviteli sín köré épül /U1 Bus, U2 Bus/. Mindkét busz a DEC PDP11 UNIBUS-éhoz [55] hasonló MASTER-SLAVE protokoll és előzetes kiválasztás alapján 16 bites szavak vagy byte-ok átvitelét teszi lehetővé. A buszok címezési tartománya 18 bit, azaz 256 kbyte /128 kszó/. A két busz teljesen identikus. Az U2 feladata azonban kizárólag a közös memória és a DCU közötti nagysebességű - burst - átvitek biztosítása, az U1 busz tehermentesítése. Az U1 busz pedig az egyes mikroprocesszorok és a közös memória közötti adatátvitek lebonyolítására szolgál. A két busz adatforgalmát egyszerű buszvezérlő egységek /Bus Controller 1 és 2/ vezérlik. Ezek alapvetően abban különböznek a hasonló célú vezérlőegységektől /PDP 11, TPA'70, stb./, hogy nem kötődnek szorosan egy központi egységhez, hanem csak a memóriáhozáférési kérelmek prioritási sorrend szerint való kielégítését te-

szik lehetővé. /A PDP 11 UNIBUS vezérlőegysége például ezt a prioritási sorrendet felborítva lehetővé teszi, hogy a hozzá kapcsolódó központi egység bármely időpontban elfoglalhassa a buszt/. Az U1 buszon levő minden egyes mikroprocesszor "Slave processzor" szerepét tölti be, nincs igazi "Master". Buszforgalmi hiba esetén a buszvezérlő egységek az úgynevezett "supervisor - processzor" -nak adnak át hibaüzenetet. Ez a processzor a rendszerben a GPC. Azért esett erre a választás, mert csak egy standard, nem opcionális, fixprogramú - nem mikroprogramozható - processzor láthatja el ezt a szerepet, tekintettel arra, hogy a bekapcsolási inicializálást is ennek az egységnek kell ellátnia. Egy - esetleg a felhasználó által is - mikroprogramozható egység írható mikroprogramtárjának tartalma a bekapcsolás után viszont nem meghatározott. A buszvezérlő egységek alkalmasak továbbá a megszakításkérések kezelésére is. Ez a lehetőség azonban nem a "slave-processzorok"-tól a "master processzor" felé irányuló kéréseket szolgálja ki /lévén, hogy az utóbbi nincs a rendszerben/, hanem a processzorok egymás közötti megszakításkéréseit továbbítja. Ennek a mechanizmusa megegyezik a normál, vektoros megszakításkérés mechanizmusával egészen a vektornak az adatbuszra történi kitételéig. Ezt a kitételt az egyik, "küldő" - vagy "source" - processzor végzi. A megszakítási vektort buszrendszerű gépeknél a busz vezérlőegységhez kapcsolódó központi egység memóriacímként értelmezi: erről a helyről veszi elő a megszakítási alprogram kezdőcímét és státuszát a futó /az adott pillanatban megszakadó/ program fő jellemzőinek /utasításszámláló, státuszregiszter, esetleg akumulátorok/ elmentése után. A GD'80 busz vezérlőegység a megszakítási vektor megfelelő bitjei alapján egy perifériális címet képez. Ez a cím a "címezett" vagy "destination" mikroprocesszornak egy bevetési regiszterét jelöli ki. A busz vezérlőegység a megszakítási ciklust egy írási ciklussá alakítja át, amelyben a cím

a busz vezérlőegység által kialakított cím, az adat pedig a megszakítási vektor. Az írási ciklus során a vektor beíródik a "destination" mikroprocesszor megcímzett bemeneti regiszterébe és a mikroprocesszor saját, belső buszán megszakításkérést okoz. Mivel az U1 buszon átadott megszakítási vektor a "source" mikroprocesszor kódját is tartalmazza, a "destination" mikroprocesszor megszakítási alprogramja értesül arról, hogy melyik mikroprocesszor volt a megszakításkérés forrása. A megszakításkérést végül a destination processzor nyugtázza oly módon, hogy az írási ciklusra válaszol. Ez a válaszjel megszünteti a "source" processzor és a buszvezérlő egység szolgálati cím- és adatjeleit.

A processzorok közötti megszakítások csak jelzőbitek cseréjére szolgálnak. A tényleges adatcsere a közös memórián keresztül történik [42]. A közös memória egy kettős hozzáférésű félvezető, írható-olvasható tároló, amely lehetővé teszi az U1, illetve az U2 busz felőli szimultán adatátvitelleket, ha különböző memóriablokkokat címezünk meg. Azonos blokkba irányuló átviteli igényeket szekvenciálisan elégíti ki a tároló vezérlőegysége. A közös memória egyszeres hozzáférésű és csak olvasható tárolókkal egészíthető ki. A közös memória alapvetően három feladatot lát el. Ebben helyezkedik el a megjelenítendő kép leírása /amelyet a DCU olvas ki, értelmez és juttat a megjelenítést végző grafikus generátorokhoz/. Itt tartják programjukat, adatmezőiket, vagy ezeknek részeit az egyes processzorok. Végül, de nem utolsósorban, ez szolgál "postaládaként" - adatbufferként - a mikroprocesszorok egymás közötti kommunikációjában.

A többprocesszoros rendszerekben az egyik központi feladat a közös memória felosztása [104]. Ugyanis meg kell akadályozni, hogy ugyanazon memóriaterületet több processzor egyidejűleg használja [56]. Ezt kölcsönös kizárásnak /mutual exclusion/ nevezik és több megoldása ismert. A probléma tisztán software eszközökkel is megoldható, de idő- és helyigények miatt ez az út körülményes. Drasztikus /és real-time feladatok-

ban általában nem megengedhető/ megoldás a közös busz lezárása közös memóriával való adatátviteli idejére /így működik például az INTEL MULTIBUS rendszer [101] /. Kijelölhetünk egy processzort, vagy mikroprocesszort a memória elosztására. Ez a módszer kétségkívül a legáltalánosabb megoldást nyújtja, de sebessége sokszor nem kielégítő, ugyanakkor költséges is /így működik az MTA SzTAKI Hálózati Főosztályán kifejlesztett MICROTERM elnevezésű többprocesszoros rendszer/. A kölcsönös kizárási probléma megoldására egy új, nagyon egyszerű megoldást alkalmaztam. Ennek a lényege egy speciális tároló, amely a közös memória minden egyes egységéhez /pl. lapjához/ vagy a közös perifériák mindegyikéhez legalább egy jelzőbitet rendel /ez a hozzárendelés software úton történik/. A jelzőbit 0 állapota azt jelenti, hogy az adott objektum szabadon használható, 1 pedig a foglaltságára utal.

Tegyük fel, hogy "A" és "B" processzorok egyszerre kívánják használni a szóbanforgó objektumot. A prioritási sorrendnek megfelelően először "A" jut szóhoz és kiolvassa az objektumhoz tartozó jelzőbitet, amelyet 0-nak - szabadnak talál. A kiolvasás hatására a jelzőbit a speciális tárolóban 1-be íródik- azaz már foglaltat fog jelezni a következő adatátviteli ciklusban szóhoz jutó és ugyancsak ezt az objektumot használni kívánó "B" processzornak. "A" - dolga végeztével - 0-t írva a jelzőbitbe, törli a foglaltságot. A fenti megoldás olyanformán is kivitelezhető, hogy az olvasás - vizsgálat - visszaírás időtartalmára a szóhoz jutott processzor lefoglalja a közös buszt. Ez az idő azonban elérheti a néhányszor 10 mikroszekundumot is, emellett a processzor és a busz között különleges kialakítású interfacet igényel. E hátrányai mellett véleményem szerint eltörpül az az előny, hogy különleges tárolót nem igényel.

2. Munkamegosztás az egyes mikroprocesszorok között

2.1. A Graphic Peripheral Controller egy 8 bites standard mikroprocesszor, amely a grafikus bemeneti eszközöket, a konvencionális perifériákat és a lassú háttértárolókat kezeli. Ezenkívül kiegészíthető memóriabővítéssel és egy aritmetikai processzorral is /APU/. A grafikus bemeneti eszközök a Graphic Peripheral Interface Bus-ra /GPIB/ csatlakoznak, ami nem más mint egy dekódolt, illesztett 8 bites adatátviteli sín. A grafikus perifériák koordinátákat és/vagy kódokat adnak a GPC-nek. A periféria választék alfanumerikus és funkcionális tasztatúrára, pozicionáló gömbre, botkormányra, értékadó potenciométerekre, kis méretű digitalizálóra /tablet/ és rajzgépre, valamint további eszközökre terjed ki. A konvencionális perifériák és háttértárolók közvetlenül a mikrogép belső buszára illeszkednek /lyukszalag és lyukkártya I/O, sornyomtató, stb. illetve mágnesszalag, floppy diszk stb./

2.2. A Host Interface a GPC-vel azonos felépítésű mikrogép, amely aszinkron vagy szinkron kommunikációs interface segítségével bonyolítja a külső számítógépekkel történő adatátvitelt. Speciális kommunikációt tesz lehetővé MSZR /DEC PDP 11/ gépekkel az U-Bus WINDOW illetve az ESZR /IBM/ sorozatu gépek multiplexer, illetve szelektor csatornáira kapcsolódó csatornaadapter. Mindkét megoldás egy-egy speciális HIF, amelyet nem 8 bites mikroprocesszor, hanem mikroprogramozható elemek valósítanak meg.

Mindkét processzor lényegében fizikai periféria kezelést végez, az adatokat a közös memóriába teszi, illetve onnan veszi ki. Lényegét tekintve tehát mindkét mikroprocesszor egy magasabb szintről hívható fizikai drivert valósít meg, amely a GPC esetén a grafikus és konvencionális perifériákat hajtja meg, az adatokat rendezi és konvertálja. HIF esetén ez a driver az adatátviteli protokollt, amely fő-

leg szinkron kommunikáció esetén jelent nagy terhelést, továbbá a karakterszűréseket, és a magasabbszintű adatátviteli algoritmusokat foglalja magában.

- 2.3. A Display Control Unit egy 16 bites speciális processzor, amely a közös memóriából végtelen ciklusban olvassa ki a megjelenítendő kép leírását. A DCU belső buszára csatlakoznak a képmegjelenítő egységek, az úgynevezett grafikus generátorok: a vektorgenerátor, a karakter- és szimbólumgenerátor, valamint az intenzitás- és módosító /vonalféleség/- generátor. A DCU a közös memóriából kiolvasott utasításokat és adatokat értelmezi és azokat további feldolgozásra /képmegjelenítésre / a grafikus generátorokhoz továbbítja. Valamennyi generátorból két variáns készült: egy standard, közepes- és egy opcionális, nagyteljesítményű változat. Ezek generálási elveikben is különböznek egymástól. A közepes teljesítményű vektorgenerátor egy digitális pályaszámító egységből /interpolátorból/ és az ahhoz csatlakozó digital-analóg átalakítók- ból áll. Ennek a generátornak a sebességét és felbontóképességét gyakorlatilag a digitális áramkörök korlátozzák /egy elemi lépés megtétele 50 nanoszekundum alatt történik/. 10 bites felbontás /1024x1024 címezhető képpont/ esetén ez egy 100 mikroszekundumos generálási idejű maximális hosszúságú átlós vektornak felel meg. Ennél nagyobb sebesség /30 mikroszekundum a maximális vektorhossz mellett/ és felbontóképesség /12 bit: 4096x4096 címezhető képpont/ volt a célkitűzés. Ez a digitális generálási elv fenntartásával már nem érhető el, mivel hozzávetőleg 240 MHz-es órajelet igényelne, amely a hozzáférhető eszközökkel nem valósítható meg. A nagyteljesítményű vektorgenerátor analóg elven működik: a végpont abszolút és relatív koordinátáit digitál-analóg átalakítók segítségével feszültségekké változtatja, amelyek

integrátorokat vezérelnek. Az integrátorok a kívánt egyenesszakaszt változó időállandók mellett több lépésben közelítik meg. A változó sebesség miatt a képernyőn az intenzitás kompenzálása szükséges.

A vektorgenerátor elsősorban vektorok megjelenítésére szolgál, azonban kiegészíthető oly módon, hogy alkalmas legyen pontraszteres képrészek generálására is. Ilyenkor a képleírás annak a téglalapnak a koordinátáit, amelyen belül a pontraszteres kép megjelenik, valamint a közvetlen videoinformációt tartalmazza. A DCU vízszintes vektorokat rajzoltat a vektorgenerátorral /a sorok között egyet léptet függőlegesen/ és a memóriából kiolvassa a hozzá szükséges videoinformációt. Ilyen esetben nagytömegű adatot kell gyorsan átvinni a DCU és a közös memóriák között, hiszen ha minden egyes lépéshez csak egy bit /0:kioltva, 1:felvillan/ tartozik, akkor is 16 lépésenként /800 nsec a digitális vektorgenerátornál/ egy új 16 bites szóra van szükség. A kiolvasott videoinformáció és a vektorgenerátor lépéseit szinkronizálni kell, ezért a raszteres képek megjelenítésére csak az állandó sebességgel rajzoló, digitális elvű vektorgenerátor alkalmas. A számítógéppel segített tervezés területén az alkalmazások többségében eleendő karakterek és vektorok, illetve vektorokból összetett képelemek megjelenítése. Ezért a GD'80 rendszerek nem tartalmaznak kör-ív generátort, mint a GD'71. A kör egyébként is csak kétdimenziós esetekben tekinthető általánosan elfogadható megoldásnak, hiszen a kör három dimenzióban való transzformációja /pl. forgatása/ útján keletkező görbék kúpszeletek, nem pedig körök. A DCU ezzel szemben, minthogy nagyteljesítményű mikroprogramozott aritmetikai egységet is tartalmaz, oly módon mikroprogramozható, hogy a képleírásban levő függvényparaméterek alapján kúp-

szeleteket és trigonometrikus függvényeket közelítő egyenes szakaszokat állítson elő és ezeket a vektorgenerátorhoz továbbítsa. Ily módon nincs szükség a közelítő vektoroknak a közös memóriában való tárolására. A közelítési algoritmusokra többféle megoldás lehetséges: inkrementális interpoláció, metszéspontok kiszámítása /clipping/ tengelyekkel párhuzamos egyenesekkel, stb.

A karakter- és szimbólumgenerátor esetén hasonlóképpen két eltérő megoldás alakult ki.

A standard karaktergenerátor 7x9 pontmátrixban ábrázolja a karaktert vagy szimbólumot /15. ábra/: valamennyi pontot végig kell járni és a megfelelő pontokban állóhelyzetben fel kell villantani a sugarat. Ez az elv is tisztán digitális megoldást tesz lehetővé szintén egy digitál-analóg átalakítópárral a végén. A digitális elektronika sebessége és a digitál-analóg átalakítók beállítási ideje az ilyen elvű generátor karakterrajzolási sebességét kb. 17 mikroszekundumra korlátozza. A nagyteljesítményű karaktergenerátor ezzel szemben szakaszokból rakja össze a rajzolandó karakter képét. A szakaszokat programozható analóg integrátorok állítják elő.

A szövegmegjelenítés fontossága miatt mindkét karakter- és szimbólumgenerátor egy sor programozható extra tulajdonsággal rendelkezik. A 128 standard ASCII karakteren kívül további 3x128 fix és 128 a felhasználó által run-time átprogramozható szimbólum megjelenítését teszik lehetővé. Az előbbieket "mikro"-programjai csak olvasható tárolókban helyezkednek el, az utóbbiak mikroprogramjai írható-olvasható tárolókba tölthetők. Ezenkívül az írásirány, az írássűrűség, a 75°-os dőlésszög, karakterforgatás és a karaktermagyság is programozható. A fenti tu-

lajdonságok a rajzos formában megjelenített információ feliratozása mellett nagyteljesítményű önálló szövegszerkesztő funkciók megvalósítását is lehetővé teszik.

Az intenzitás- és módosítógenerátor lehetőséget biztosít arra is, hogy különböző intenzitású szakaszok és karakterek mellett a videojel szaggatása révén a vektorokhoz különböző vonaltípusokat /folytonos, szaggatott, pontozott, eredményvonal/ rendeljünk.

A DCU belső buszára csatlakozik a fénytoll is, az egyetlen interaktív grafikus eszköz, amely real-time kiszolgálást kíván és amely szorosan kapcsolódik a képmegjelenítéshez. Ugyanis az éppen megjelenített képelem teljes információtartalmához hozzá kell férni, ha az a fénytoll látómezejében van. A fénytoll alapjában véve képelemek azonosítására szolgál és minden egyéb funkció /fénykereszt-követés, rajzolás, törlés, stb./ erre vezethető vissza. A fentiek miatt a fénytoll nem csatlakozhat - a többi grafikus eszközhöz hasonlóan - a GPC-hez.

A DCU grafikus /rajzoló/utasításokon kívül számítógépszerű tulajdonságokkal is rendelkezik/pl. címzésmódok, regiszterkezelő, vezérlésátadó és ciklusszervező utasítások, stb./, amelyek lehetővé teszik, hogy a DCU struktúrált képleírást értelmezzen.

A DCU mikroprogramozott volta lehetővé teszi más display utasításrendszerek értelmezését, konzolok emulációját is.

A grafikus generátorok analóg jeleket állítanak elő. A vektorgenerátor kimenő jelei a nagyjelű eltérítő erősítőn keresztül a nagyjelű eltérítő tekercsre, a karakter- és

szimbólumgenerátoréi a kisjelű eltérítő erősítőn keresztül a kisjelű eltérítő tekercsre jutnak. A két eltérítő-rendszer jelei a mágneses térben szuperponálódnak: ezt a kettős rendszert a két erősítő eltérő sávszélessége és "eltérítőképesége" - azaz áramleadása - indokolja. Az intenzitás- és módosítógenerátor jele a videoerősítő segítségével jut a katódsugarcsövek elektródáira.

A színes kijelzés az értekezés II. fejezetének 1. pontjában leírtaknak felel meg. Az anódfeszültség kapcsolása 25 mikroszekundum alatt zajlik le. Ezen idő alatt kell megvalósítani az intenzitás- és fókuszáramkörök kompenzációját is. A szín információ a képleírásban ugyanolyan szerepet tölt be, mint az intenzitás vagy a vonalféleség, azaz az eltéréssel, hogy figyelembe kell venni a minden egyes színváltáshoz szükséges időt. Ezért célszerű a színváltások számának minimalizálása, azaz az azonos színinformációjú képelemek blokkokba szervezése.

Az egyszínű és a színes kijelzők is kétféle méretben állnak rendelkezésre. Az E /EC/ változat 20x25 centiméteres, az S/SC/ változat pedig 50 centiméteres kör alakú képernyővel rendelkezik.

- 2.4. A Display Processing Unit a DCU-val azonos felépítésű, de mikroprogramjai és perifériái eltérőek. A DPU-ban miniszámítógépeket emuláló mikroprogramok futnak /PDP11/40,45 TPA'70/25, stb/. Ezek lehetővé teszik például a GD'71 - - TPA'70 rendszerre megírt általános és grafikus, alap- és felhasználói software rendszerek változtatás nélkül való futtatását. Ez az emulációs megoldás a hordozható, eszköz-

független, ezért általában kevésbé effektív programozási módszerek jó alternatívája.

A DPU belső buszára kapcsolódik a gyors háttértárolók /általában "cartridge" vagy "pack" diszk/ vezérlőegysége, amely igen egyszerű áramkör. A háttértárolóra /vagy-ról/ történő adatátvitel idejére a DPU mikroprogramok és az adatutak az U1 busz felé a tárolóvezérlő rendelkezésére állnak /azaz a DPU mint számítógép erre az időre leáll/. Ez megtehető minden olyan esetben, amikor az átvitel kezelése "WAIT-módú", tehát az átvitel befejeződéséig a processzor várakozik, más akciót nem kezdeményez. Ez igaz például a TPA'70-re kifejlesztett egyfelhasználós operációs rendszerre, a DOST-ra, amelyet a GD'80-on standard rendszerként kívánunk alkalmazni.

- 2.5. A Transformation Processing Unit szintén a DPU belső buszára kapcsolódik és gyakorlatilag annak mikroprogramozott aritmetikai kiegészítése. Fő célja a lebegőpontos műveletek nagysebességű elvégzése. A TPU szóhosszúsága 48 bit. A DPU-TPU páros alkalmas arra is, hogy a magasszintű geometriai leírást, a matematikai modellt vagy a grafikus protokollt /ez utóbbi az a magasszintű formátum, amelyen a GD'80 grafikus terminál a környezetével - például egy nagyszámítógéppel érintkezik/ kifejtse, ebből a DCU által értelmezhető képleírást állítson elő és azt a közös memóriában elhelyezze. A DPU szerepe itt az adatstrukturák közötti konverzió, listakezelés. A transzformációkkal kapcsolatos műveleteket pedig a TPU végzi el /transzformációs mátrixok szorzása, mátrixvektor szorzás, képkivágás - ablaktechnika-, perspektív transzformációk két és három dimenzióban, homogén vagy lebegőpontos koordinátákkal, stb./.

3. Processzor típusok és ezek struktúrája

A GD'80 családban két alapvető típusú mikroprocesszort alkalmaztam. A Host Interface és a Graphic Peripheral Controller standard, nyolcbites, fix utasításrendszert végrehajtó mikroprocesszorok /INTEL 8080 vagy Z-80/. A Display Control Unit, a Display Processing Unit és a Transformation Processing Unit bit-slice elemeket tartalmaz, az utasítások végrehajtása mikroprogramozott. Az utóbbi processzorok adathossza 16 bit, illetve a TPU esetében 48 bit. A mikroprogramozás előtérbe helyezését a speciális utasításrendszerek végrehajtása, meglévő display konzolok és processzorok emulációja, valamint a real-time grafikus igények kielégítése indokolja.

3.1. A 8 bites, standard mikroprocesszorok általános blokkvázlatát a 16. ábrán mutatom be. A mikroprocesszor belső busza /MIPROBUS/ felett elhelyezett elemek standard, az alatta levők pedig opcionális egységek. A standard rész mechanikailag is egy egység: egy nyomtatott kártyán helyezkedik el, az opciók szintén kártyaszintűek. A standard rész a CPU-t, 1 kbyte RAM-ot, 6 kbyte ROM-ot, a DMA Controllert, az Interrupt Controllert, a Real Time Clockot, két 8-bites párhuzamos interfacet /egy input, egy output/, a nyolc interaktív grafikus periféria kiszolgálására alkalmas buszvezérlőt, /Graphic Peripheral Interface Bus/, a soros szinkron, aszinkron /CCITT V24/EIA RS232-C/ illesztő egységet /ez utóbbi kettő közül egy adott konfigurációban csak az egyik lehet jelen/ és a MIPROBUS WINDOW nevű interface egységet tartalmazza. Ez utóbbi teremt kapcsolatot a MIPROBUS és az U1 busz között oly módon, hogy a MIPROBUS-on a címtartomány felső felébe irányuló ciklusok hatására memória hozzáférés-kérés keletkezik az U1 buszon. Az U1 busz 18 bites címe a MIPROBUS legalacsonyabb helyértékű 15 bitjéből és egy, a MIPROBUS felől perifériaként címezhető 3 bites regiszter tartalmából tevődik

össze. Az U1 buszon történő adatátvitel idejére a MIPROBUS-on az átvitelt kezdeményező CPU vagy DMA-egység várakozik. Minthogy a MIPROBUS WINDOW egyirányú kapcsolatot teremt, az U1 busz felől a MIPROBUS címek nem érhetők el.

A standard 8 bites mikroprocesszorok memóriatérképét a 17. ábra mutatja. Az első 16 kbyte csak olvasható memória /ROM/, amelyből az első 6 kbyte standard. A második 16 kbyte írható-olvasható memória /RAM/, kivéve az utolsó 256 byte-ot, ahol a perifériák státusz- és adatregiszterei címezhetők. Az írható-olvasható memóriákból 1 kbyte standard. A második 32 kbyte a MIPROBUS WINDOW területe.

Az opcionális bővítőegységeknek különböző felhasználásokban mindig csak egy adott részhalmaza csatolható a mikroprocesszorhoz, minthogy a mikroprocesszor korlátozott teljesítőképességű, ezért nincs értelme valamennyi egység egyszerre való jelenlétének. A bővítések közül a memória kiegészítő elemek terjesztik ki a használható tároló területet: 14 kbyte RAM és 40 kbyte ROM. Ez utóbbi nyilván ellentmond a címtérképen rendelkezésre álló 10 kbyte-os területnek. Ebből a 40 kbyte-ból ténylegesen 10 kbyte címezhető egyszerre: hogy a négy 10 kbyte-os egység közül éppen melyik, ezt a mikroprocesszor alapsoftware egy perifériális regiszter segítségével választhatja ki. A memória úgy működik, mint egy fizikailag közel nulla hozzáférésű háttértároló: a perifériális regiszter átírásával azonnal tartalmat "vált" /erre utal a neve is: BACKPROM/.

Az opcionális periféria illesztőegységek a következők:

ASTRO: nagysebességű soros szinkron/aszinkron modem interface /CCITT/EIA/ + ROM

SIO8:	8xsoros szinkron/aszinkron modem interface /multiplexer /CCITT/EIA/
PIO4:	4x8 bites párhuzamos interface
FD:	floppy diszk vezérlő
EC-5017:	mágnesszalag vezérlő
HP-IB:	Hewlett-Packard Interface Bus /IEEE 488/ ve- zérlő
2652:	bitorientált protokollokat megvalósító soros szinkron interface modem /CCITT/EIA/ és koaxiális kábel kimenettel.

A standard mikroprocesszorok ezen kívül lebegőpontos művele-
tek gyors végrehajtását lehetővé tevő Arithmetic Processing
Unit-tal /APU/ bővíthető ki. Ez főleg olyan konfigurációk-
ban jut szerephez, amelyekben nincs DPU, TPU és mégis helyi
intelligenciát kívánunk adni a rendszernek.

3.2. A mikroprogramozott processzorok /DCU, DPU, TPU/ hasonló
felépítésűek. Vezérlőegységük /Control Path/ azonos. Az
adatút /Data Path/ a TPU-nál eltér a másik kettőtől /18.
ábra/. A választott architektúra a vezérlőegységében hori-
zontális mikroprogramozást használ /19, 19a. ábra/. A mikro-
utasításokat a Control Store tárolja. Ennek 12 bites cím-
busza van, azaz 4096 szó címezhető. A mikroutasítások hossza
processzorfüggő /DCU: 48 bit, DPU: 52 bit, TPU: 64 bit/. A
Control Store kimenetei a Mikroutasítás Regiszter bemenetei-
re csatlakoznak. Ez az úgynevezett "pipeline regiszter" le-
hetővé teszi, hogy az adott mikroutasítás végrehajtása /amely-
nek bitjei éppen a regiszterben vannak/ és a következő mikro-
utasítás előkészítése /a regiszter bemenetén/ egymással pár-
huzamosan történjék, azaz a két idő ne adódjék össze, hanem
közülük a hosszabb határozza meg a mikrociklus időt. A követ-
kező mikroutasítás címét a Sequencer Vezérlés ellenőrzése
alatt a Mikroprogram Sequencer állítja elő, mégpedig a belső
stack, a belső mikroprogram számláló, egy szintén belső re-

giszter tartalmából, vagy a mikroprocesszor belső buszára kikapuzott adatszóból. Ezt a mikroprogram címet módosítják az OR bemenetek, amelyeket szintén a Sequencer Vezérlés befolyásol. A legalsó helyiértékű 8 OR-bemenetre csatlakoznak a 4-1 OR-multiplexerek. Ezek a belső busz alsó, illetve felső helyiértékű nyolc bitjét, vagy a mikro-interruptokhoz tartozó kezdőcímet előállító MIT /Micro-interrupt/PROM kimeneteit, vagy a belső buszon összesen 48-féle 16 bites kombinációt figyelő és ezekre egyenként változó mikroprogram címet előállító OR-FPLA kimeneteit kapuzzák. A felső helyiértékű 4 bitre mindig 0 kapuzódik, azaz az OR-bemenet ezeket a biteket nem módosítja. A Sequencer Vezérlés gyakorlatilag a MID /Microinstruction Decode/ PROM-ból áll. Ennek a bemenetei közvetlenül a Mikroutasítás Regiszterhez kapcsolódnak. Ez alól kivétel a Condition jel, amelyet az éppen végrehajtás alatt álló mikroutasítás ellenőrzése alatt a belső buszra kikapuzott adatszóból egy 16-1 multiplexer választ ki és egy EX-OR /kizáró VAGY/ kapu invertál /vagy sem/.

A Control Store standard módon csak olvasható memóriából épül fel. Opcionálisan bővíthető írható résszel is /Writable Control Store/. Ez az U1 busz felől közösleges memóriaként kezelhető. A beírást a WCS Control Logic segítségével úgy oldottam meg, hogy a beírás időtartamára nincs szükség a mikroprogram futásának megállítására /run-time WCS átírás/. A WCS jelentősége a mikroprogramok és a hardware tesztelése és kipróbálása, valamint a rendszer dinamikus átprogramozása, rekonfigurációja, az utasításrendszer lecserélése szempontjából igen nagy. /Ezért legszigorúbb embargó alatt áll/.

A mikroprogramozható processzorok belső órajeleit a programozható Clock Generator állítja elő - két módban. Az úgynevezett "Normál Mód" jellemzője, hogy az óraimpulzusokat egy szabadon futó kvarc oszcillátor vezérli. "Maintenance mód"-ban ezeket az órajeleket vagy bármely kombinációjukat egyenként, az U1 busz felől a Maintenance Logic segítségével egy regiszter adott bitjeire való beírás útján lehet kiadatni.

Az adatút a belső 16 bites busz köré épül. Ezen busz segítségével lehet az adatokat az egyes kisebb egységek között mozgatni. Az U1 Bus Transceivers révén folytathatók adatátvitel az U1 Buson. A végrehajtandó /makró/ utasítást az Instruction Register tárolja. Ennek a kimenetei közvetlenül vagy az Instruction Decode FPLA-kon keresztül is kikapuzhatók a belső buszra. Az utóbbi megoldás szolgál arra, hogy a FPLA-k segítségével mikroprogram kezdőcímekeket állítsunk elő, és ezeket a belső busz segítségével a Microprogram Sequencerbe juttassuk. Egy további "Register Select FPLA" szolgál a /makro/ utasításban levő regiszter hivatkozások és az ALU regiszter - címeke egymáshoz rendelésére.

Az aritmetikai egység /ALU/ a hozzátartozó regiszter készlettel együtt a belső buszra kapcsolódik. A bemeneti adatbyte-ok egy multiplexer segítségével megcserélhetőek /swapping/. Az aritmetikai egység vezérlő kódjait közvetlenül a Mikroutasítás Regiszterből kapja. A belső buszra tárolók is kapcsolódnak: a Register File közvetlenül a mikroutasításszóból címezhető, a Scratchpad RAM-nak viszont saját, az adatátvitel előtt töltendő, mikroutasításból növelhető, vagy csökkenthető címregisztere /MAR/ van.

Az a megoldás, hogy a Mikroutasítás Regiszter tartalma - 16 bitenként - szintén kikapuzható a belső buszra, lehetővé teszi adatok továbbítását a Control Path-ból a Data Path-ba. Ez akkor célszerű, ha a mikroutasításban leírt közvetlen /immediate/ adatot fel akarjuk használni a Data Path valamelyik regiszterében.

A belső busz szolgál minden további egységgel való kapcsolat megteremtésére. Így ide kapcsolódnak a grafikus generátorok adat,- parancs- és státuszregiszterei /DCU/, a cartridge diszk vezérlő regiszterei /DPU/, stb.

A TPU Data Path-ra merőben más megoldást dolgoztam ki. Ezt az architektúrát a 20. ábrán mutatom be. A TPU egy 48 bites processzor, amely három egyenként 16 bites egységből épül fel. Ezek az egységek a mikroprogram szintjén összerendelhetők vagy külön egységekként kezelhetők, azaz az architektúra dinamikusan változtatható. Ennek megfelelően a TPU-n belül három 16 bites ALU Bus található /0,1,2/ amelyek Transceiverek /0,1,2/ segítségével a TPU Global Bus-ról leválaszthatók.

A Transceiverek nyitásáról és annak irányáról a mikroprogram gondoskodik. A TPU Global Bus-ra a DCU és DPU belső buszához hasonlóan - kapcsolódik az Instruction Register /FPLA-ival együtt/, és a Mikroutasítás Regiszter. A TPU nem az U1 Bus-ra kapcsolódik, hanem a DPU-hoz. A kapcsolatteremtés eszköze két 256 szavas First-in-First-Out /FIFO/ tároló, amelyeknek jelzőbitjei /FIFO Empty, FIFO Full/ mikrointerruptokat okoznak a megfelelő /DPU, TPU/ belső buszokon. A FIFO-k beépítését a két processzor közötti sebesség ingadozások és terhelés változások indokolják.

A közvetlen, halasztást /sorbanállást/ nem tűrő parancs- és státuszinformációk két kettős hozzáférésű memória segítségével kerülnek egyik processzorról a másikra.

Az ALU buszok mindegyikéhez egy 16 bites ALU /a regisztereivel együtt/ és egy 256 szavas Scratchpad RAM /a címregiszterével - MAR - együtt/ tartozik. Ha az ALU-kat többszörös szóhosszúságúra mikroprogramozzuk /azaz a mikroprogram szintjén fűzzük össze/, az átvitel és az eltolás /shift/ bitek összekapcsolásáról a mikroprogram által vezérelt multiplexerek gondoskodnak /ez utóbbiak nincsenek feltüntetve az ábrán/. Minden ALU- Bus-on, ALU-ban és Scratchpad RAM-ban párhuzamosan ugyanaz a művelet hajtódik végre. A mikroutasításban a végrehajtás érvényesítése külön tiltható és egy helyi feltétel Local Condition/ teljesüléséhez köthető. A Local Condition képzése ugyanugy történik, mint a Control Path esetében a Condition-é. A Condition-t egyébként a TPU esetén Global Condition-nek neveztem. A fentiek alapján feltételes mikroutasításátlépések realizálhatók.

A TPU egyik célja geometriai transzformációk nagysebességű, real-time elvégzése. A geometriai transzformációk /eltolás, forgatás, léptékváltoztatás/ általános kezelésére a vektorok 4x1-es homogén koordinátákban való leírása /X,Y,Z és lépték/ és a 4x4-es transzformációs mátrix használata a legalkalmasabb. Transzformált vektorok az eredeti vektor és a transzformációs mátrix szorzásával nyerhetők. Új transzformációk előállítására - a transzformációk konkatenációjára - az eredeti transzformációs mátrixot egy másik, azonos dimenzióju mátrixszal kell megszorozni. A fentiekből látható, hogy a geometriai transzformációk vektor- mátrix-, illetve mátrix- mátrixszorzásokkal írhatók le. Egy új képelem koordinátáinak előállításához a

a geometriai transzformáció után még további lépések is szükségesek: homogén koordinátákból három dimenziós koordinátákká való átalakítás, a képernyő koordinátarendszerén kívül eső vektorszakaszok vágása /clipping/, lát-hatósági vizsgálatok, perspektív transzformációk a két dimenziós koordinátapár előállítására. Erre a speciális célra fejlesztettem ki a Matrix Multiplier nevű egységet, amely az ALU1 és ALU2 Bus-hoz is kapcsolódik. Ennek a részletes blokkvázlata látható a 21. ábrán. A nagysűrűségű és nagysebességű csak olvasható memóriák megjelenése tette lehetővé, hogy szorzótáblákkal, párhuzamosan oldjuk meg a szorzást az n-lépéses összeadás és eltolás helyett /n a szorzó bitjeinek száma/ [171, 203]. A Matrix Multiplier szorzóműve a fenti elemekből épül fel: egy megfelelő tartalmú 256x8 ROM éppen egy 4x4-bites szorzónak fogható fel. A ROM címvezetékeinek két csoportja a szorzó és a szorzandó, az adatkimenet pedig a dupla-pontosságú eredmény. A 4x4-bites szorzókat a Wallace-algoritmust megvalósító 512x4 bites ROM-ok /Wallace tree/ és összeadók segítségével lehet nagyobb bitszámokra kiterjeszteni. Ez esetünkben 16x16 bit, végeredmény 32 bit, azaz dupla-pontosságú. Megjegyzendő, hogy redukciós eljárásaként a Booth-algoritmus is használható [88, 116, 186]. Fel kell hívni a figyelmet arra, hogy a szorzómű kimenetei közbelső regiszter nélkül csatlakoznak az ALU1 és ALU2 Bus-ra. A szorzó és a szorzandó beállítás után 150 nano-szekundum múlva a szorzómű kimenetén előáll a 32 bites eredmény. Ez az idő rövidebb, mint a mikroprogramozott processzorok ciklusideje, azaz egy szorzást éppúgy egyetlen mikroutasítás alatt lehet végrehajtani, mint egyszerű aritmetikai, logikai, vagy adatátviteli műveleteket.

A szorzómű szorzandó bemenetére egy 16 bites Bemeneti Regiszter kapcsolódik, amely az ALU1 Bus felől tölthető. A szorzó bemenetre egy 16x16 bites regiszter file, a Matrix Store csatlakozik. Ez tartalmazza a 4x4 elemű, homogén koordinátákra értelmezett transzformációs mátrixot. A Matrix Store címzését a Matrix Address Counter és Mask Register végzi. Ezt a mátrix minden egyes elemének betöltése előtt be kell állítani /az ALU2 Bus felől tölthető/. A Matrix Store az ALU2Bus-on érhető el: tölthető és tartalma vissza is olvasható. A szorzómű kimenetének az ALU1, illetve ALU2 Bus felől történő kiolvasása a Matrix Address Counter tartalmát eggyel növeli /inkrementálja/. Ha egy szorzandó vektor adott elemét a Bemeneti Regiszterbe töltjük, négy egymást követő kiolvasással négy részletszorzatot kapunk meg. Ezt a vektor további elemeivel háromszor megismételve majd a részletszorzatokat összegezve előáll az új transzformált vektor, duplapontosságú formában. A szorzás tiszta ideje - a mátrix feltöltését leszámítva - 30 mikrociklus, azaz 6 mikroszekundum alatt zajlik le. Ennél a műveletnél igen jól kihasználható az ALU1 és ALU2 Bus struktúrájának mikroutasítás szintű változtatása. A transzformációs mátrix illetve a vektor betöltése külön, tehát 16 bites formátumban, a kiolvasás és a részletszorzatokból az új transzformált vektor előállítására viszont duplapontosságú formátumban, azaz az ALU1 és ALU2 Bus egy 32 bites egységként való kezelése útján történik.

A fentiekben nem esett szó a Mask Registerről. Ennek akkor van szerepe, ha kisebb dimenziójú mátrixszal dolgozunk /3x3, 2x2/. A Mask Register 0 értékű bitjei a

Matrix Store megfelelő cimbitjeit nullává teszik, azaz a cím, az inkrementálás során a Matrix Store-nak nem mind a 16 elemét járja végig ciklusban, hanem kilencet vagy négyet.

A transzformációk kifejtése a DPU-TPU kettős segítségével úgy történik, hogy a DPU értelmezi a magasabb szintű geometriai leírást, matematikai modellt vagy grafikus protokollt. Az ebben található transzformációs mátrixot /amelyek elemeit például a grafikus bemeneti eszközök segítségével módosíthatjuk/ a FIFOIN segítségével a TPU-hoz küldi át, ahol a TPU mikroprogram segítségével a Matrix Store-ba töltődik. A magasabbszintű leírásban ezek után vektorok listája következhet, amelyekre sorban alkalmazni kell a megadott transzformációkat, azaz meg kell szorozni valamennyit a korábban megadott transzformációs mátrixszal. Ezeket a FIFOIN segítségével sorban átveszi a TPU, majd elvégzi a szorzásokat és előállítja az új transzformált vektorokat. A transzformáció után következő 3D-re való konvertálást, clippinget és perspektív transzformációkat - 2D vektorokká - itt nem részletezem. Megjegyzem azonban, hogy például a 3D-re való áttéréskor, azaz a három koordinátának a skálafaktorral való osztásakor, és a 3D clipping esetében is az egymással - időben - párhuzamosan működő ALU Bus-ok nagymértékben kihasználhatók.

4. Az egyes processzoroknak az egymáshoz való kapcsolódásában az alábbiakban összefoglalt multiprocesszoros módszereket használtam fel.

4.1. Közös busz adatok átvitelére és processzorok közötti megszakításkérésekre: az U1 Bus szolgál a processzorok és a közös memória közötti adatátvitelre. Az adatoknak a memóriába való letétele, illetve onnan való kiolvasása után a

processzorok megszakításkérésekkel értesítik egymást arról, hogy az adatátvitel megtörtént.

- 4.2. Közös memória kölcsönös kizárással: ez a magja a processzorok közötti adatátvitelnek. Memóriaterületet minden processzor "igényel" és azt a kölcsönös kizárást egyszerű elven megvalósító flag-tároló segítségével "szerzi" meg.
- 4.3. Kettős hozzáférésű memóriák és FIFO-k: az előbbi megoldásúak a közös memóriák, amelyek az U1 és U2 Bus felől szimultán hozzáféréseket engednek meg, valamint a DPU és TPU belső buszai közötti közvetlen /azonnali/ adatcserét lehetővé tevő tárolók. First-In-First-Out tárolót szintén a DPU és TPU közötti kommunikációra használtam listák, adatsorok cseréjére és a két processzor közötti sebességigadozások kiegyenlítésére.
- 4.4. MIPROBUS WINDOW: a standard 8 bites mikroprocesszorok memóriatartományának második felébe irányuló ciklusok átkerülnek az U1 Bus-ra, lehetővé téve a közös memória és esetleges közös perifériák saját memóriaként való kezelését, elérését. Hasonló elvű az U -Bus WINDOW, amely az U1 Bus és MSZR/DEC gépek UNIBUS-a között teremt kapcsolatot.
- 4.5. Mikroprogramozott multiprocesssing: ez a processzor architektúrájának megváltoztatása, a mikroprocesszorok párhuzamos működtetése a mikroprogram felügyelete alatt. Ezt a módszert alkalmaztam a TPU esetében a három 16 bites mikroprocesszorból álló aritmetikai egység kialakításában.

A GD'80 család kialakításánál az egyes funkciók hatékony megvalósítása különböző struktúrájú processzorok alkalmazását követelte meg. Ezek adatcseréjének, kommunikációjának biztosítása tette szükségessé a különféle

módszerek alkalmazását.

5. Rendszerfejlesztési eszközök

Ebben a fejezetben csak vázlatosan érintem a GD'80 család fejlesztéséhez kialakított eszközöket.

- 5.1. Csak olvasható memóriák kezelésére /programozás, ellenőrzés, töltés/ egy software-hardware rendszert alakítottunk ki, amelynek alapja a KFKI által kifejlesztett TPA 70/25 küsszámítógép egy cserélhető lemezes háttértárolóval. Ezen futnak - a diszkes operációs rendszer /DOST/ alatt és annak minden lehetőséget /editor, file-kezelés/ kihasználva - azok a programok, amelyek a csak olvasható memóriák on-line programozó és ellenőrző berendezését /PROM programozó/ valamint a beégetendő file-okat kezelik, javításukat, módosításukat lehetővé teszik.
- 5.2. A GD'80 valamennyi processzorára cross-assemblereket, cross-mikroassemblereket és cross-szimulátorokat alakítottunk ki. Ezekre a processzorokra a magasszintű BASIC, FORTRAN és GESAL rendszerprogramozói nyelv kódgenerátorai is elkészültek. Valamennyi program /az 5.1. pont szerinti rendszerhez hasonlóan/ a TPA'70/25 DOST alatt fut. Mivel a GD'80 DPU rendelkezik TPA'70/25 emulátorral, a fentiekben leírt valamennyi program futhat a GD'80 konfigurációkon /ahol van DPU/ is.
- 5.3. Valamennyi rendszerelemre /processzorok, memóriák, perifériák, megjelenítő egység/ tesztprogramokat készítettünk.
- 5.4. A hardware és firmware diagnosztika és mikroprogramfejlesztés céljára karbantartási maintenance programokat hoztunk létre. Ez utóbbiak többek között a mikroprogramozott pro-

cesszorok mikroutasításonkénti megállással és közbenső regiszterellenőrzésekkel való futtatását teszik lehetővé.

Az 5.3. és 5.4. programok a GD'80 GPC és DPU processzorain futnak.

6. A GD'80 rendszerelemkből kialakított néhány tipikus konfiguráció.

Ebben a fejezetben a GD'80 család öt báziskonfigurációját mutatom be. Ezenkívül kitérek néhány olyan konfigurációra is, amelyek GD'80 rendszerelemekből épülnek fel, de nem grafikus display-ek.

A GD'80 báziskonfigurációk alapjában véve a processzálási képességben különböznek egymástól. A processzorok valamelyikének hiányában a hozzárendelt funkció hiányozhat vagy egy másik processzor veheti át a szerepét egy alacsonyabb szinten /például csökkentett sebesség/. A GPC és a DCU valamennyi grafikus display konfigurációnak standard része, a HIF, DPU és TPU pedig opcionális egységek. Adott báziskonfiguráció választása több paramétert még nem definiál /kijelző nagysága, kijelzés minősége, memóriaméret, grafikus és konvencionális perifériák és háttértárak választéka, stb./. Ezért az öt báziskonfiguráció paramétereit jellemző példának és nem egyedül üdvöztető, kizárólagos megoldásnak kell tekinteni.

6.1. GD'80 BT/E /Basic Terminal, kis egyszínű kijelzővel; /22. ábra//

Ez egyszerű grafikus terminál, amely szinkron, vagy aszinkron modemen /vagy közvetlenül, nullmodemen/ keresztül kapcsolódik egy külső számítógéphez /Host-hoz/.

A GPC feladata ebben a konfigurációban a következő:

- kezeli az aszinkron kommunikációs vonalat,
- a vonalon vett adatokat - konverzió nélkül - elhelyezi a közös memóriában,
- elindítja és megállítja a DCU-t /a képgenerálást/,
- kezeli az alfanumerikus tasztaturát és az arról, valamint a DCU-n keresztül a fénytollról jövő operátor beavatkozásokról jelentéseket küld a kommunikációs vonalon keresztül a Host-nak.

A képleírás előállítás, a felhasználói program futása, a perifériák logikai kezelése a Host számítógépen történik. A BT konfiguráció software-je az ugynevezett BT executive, amely a fentiekben részletezett funkciókkal rendelkezik. /Ez a konfiguráció a Tektronix terminálok, valamint a DEC GT40, 41, 42 típusok kategóriájának felel meg/.

6.2. GD'80 GC/E /Graphic Computer, kis egyszínű kijelzővel;
/23. ábra//

Asztali számítógép grafikus lehetőségekkel - így lehet jellemezni ezt a konfigurációt. A GPC-n egy floppy diszk bázisú egyfelhasználós mini-operációs rendszer fut, amely a programfejlesztést és - futtatást is támogatja. A rendszer BASIC-ben programozható. Az "MGSS80" grafikus vonalrajzoló szubrutincsomag a BASIC-ből használható. Ez a konfiguráció - megfelelő perifériákkal kiegészítve - egyszerűbb alkalmazásokra is használható /pl. NC programozói állomás/.

6.3. GD'80_IT/S /Intelligent Terminal , nagy egyszínű ki-
jelzővel /24. ábra//

Ez a konfiguráció egy nagy ESZR/IBM számítógép - HOST -
/pl. R-40, R-55/ "közeli" terminálja, amely a Channel
Adapter segítségével közvetlenül a multiplexer csator-
nára csatlakozik. A terminál intelligenciája abban rej-
lik, hogy komplex grafikus parancsok adhatók számára és az
operátori interakció útján létrejövő megszakításkérések
előfeldolgozás után kerülnek a nagygépre. Az előző fela-
datot a DPU, az utóbbit a GPC látja el. A felhasználói
program a nagyszámítógépen fut, célszerűen valamely ma-
gasszintű nyelven megírva /FORTRAN, PL/I/. A "HOST
GSS80" grafikus szubrutincsomag a magasszintű nyelvből
közvetlenül hívható. Problémát jelent a grafikus peri-
féria beillesztése az operációs rendszerbe, mivel az
IBM Graphic Access Method /GAM/ az ESZR-ben nincs rend-
szeresítve.

Ez a konfiguráció alkalmas az IBM 2250 /EC 7064 grafikus
terminál emulációjára.

6.4. GD'80_AGS/SC /Autonomous Graphic System, nagy színes ki-
jelzővel; /25. ábra//

Ez önálló grafikus tervezőrendszer. Megnövelt közös me-
móriakapacitása /64 kbyte/, háttértárolói /cartridge
diszkés mágnesszalag/, perifériái és nem utolsósorban a
DPU/TPU processzorkettős lehetővé teszik a felhasználói
problémák megoldását - kizárólag helyi eszközökkel. A
DPU-n, amelyben a TPA¹70/25 kibővített utasításrendszerét
emuláltuk, a DOST egyfelhasználós operációs rendszer fut.

A grafikus rendszer magas szintű nyelveken /BASIC, FORTRAN, GESAL rendszerprogramozói nyelv/ vagy assemblerben programozható. Ez utóbbira általában csak akkor kerül sor, ha a magas szintű nyelvek futási időben vagy helyfoglalásban már nem nyújtanak megfelelő megoldást. /"Drop down to assembly level"/. A képek előállítására, transzformációjára és grafikus perifériakezelésre valamennyi nyelv a "GSS80" grafikus szubrutincsomagot használja. A "GSS80" részben a GPC-n, részben a DPU/TPU processzorkettősön fut /az utóbbin részben mikrokódban/. Itt is hasonló a helyzet, mint a magas szintű nyelvek és az assembler esetében. Ha a "GSS80" szubrutincsomag valamely oknál fogva nem ad kielégítő megoldást, az "igényes" felhasználó számára lehetőség van az operációs rendszer programfejlesztési eszközei segítségével /beleértve akár a DPU/TPU processzorok mikroprogramozását is/ egy új, saját szubrutincsomag elkészítésére.

6.5. GD'80_SGS/SC /Satellite Graphic System, nagy színes kijelzővel /25. ábra//

A szatellit konfiguráció hardware-je csak abban különbözik az AGS konfigurációtól, hogy van kommunikációs lehetősége egy Host felé /az ábrán szaggatott vonallal jelöltem/. Ez lehetővé teszi, hogy hosszadalmas, számítógépigényes feladatok elvégzésére időközönként a nagygépet is igénybe vegyünk. Így munkamegosztás jön létre a GD'80 szatellit és a nagygép között. Az AGS software kommunikációs programokkal, batch terminál emulátorokkal /IBM 2780, 3780, CDC 734, UT 200 UT200/ egészül ki, különös tekintettel olyan nagygépekre, amelyeken nincs megoldva az interaktív terminálkezelés /ebben az esetben a szatelliton elő kell készíteni egy "grafikus job"-ot, felküldeni a nagygépre, lefuttatni, majd az eredményt visszahozni a grafikus konfigurációra és

értelmezni. Ezt az eljárást "ciklikus batch"-nak neveztük el/.

Ha a GD'80 szatellit és a nagy gép igazi interaktív kapcsolatban állnak egymással, akkor egy közös nyelven érintkezhetnek. Ezt grafikus protokollnak nevezik. Ez egyfajta magasabbszintű leírás, amely azonban általában nem egyezik meg a felhasználói program matematikai modelljével. Ilyen esetben a kommunikációs vonal mindkét végén /a nagy gépen és a szatelliten is / szükség van olyan programokra, amelyek a grafikus protokollt előállítják, értelmezik. Ez utóbbit a DPU/TPU mikroprogramok is támogathatják.

A következő három konfiguráció nem grafikus display, hanem a GD'80 elemekből felépített általános számítástechnikai berendezés.

6.6. GD'80_KC /Koncentrator /26. ábra//

Ez a maximálisan nyolc alfanumerikus display-t kezelő konfiguráció két standard 8 bites mikroprocesszorból és egy közös memóriából áll. A HIF a kommunikációt bonyolítja egy adatátviteli hálózaton keresztül, a GPC pedig az alfanumerikus display-eket kezeli, adataikat blokkokba rendezi, stb. A közös memória a két processzor közötti közbenső adattárolásra szolgál.

6.7. GD'80_TS /Time Sharing /27. ábra//

Ezt a rendszert többkonzolos programfejlesztési konfigurációként jellemezhetjük, amelyben négy alfanumerikus display-en egymással párhuzamos munka folyhat. Minden felhasználó egy-egy floppy diszk segítségével "lép be" a rendszerbe.

Az alfanumerikus display-eket a GPC kezeli, a floppy diszkeket a HIF. Ezen a két processzoron egy reentráns^x editor fut, amely lehetővé teszi a file-ok rendezését, módosítását, stb. a floppy diszkeken. A háttérben álló DPU-t /TPA'70/25-nek vagy PDP 11-nek mikroprogramozva / az egyes konzolok felől assemblálásra compilálásra, programfuttatásra, nyomkövetésre lehet felkérni.

Más rendszerekhez való kapcsolatteremtés céljából a HIF soros kommunikációs interface-szel, a DPU pedig cartridge diszkkal bővíthető.

6.8. GD'80DP Data Processing, /28. ábra/

Ebben a konfigurációban két standard 8 bites mikroprocesszort dedikáltunk periféria kezelésre. A HIF szinkron kommunikációt bonyolít egy adatátviteli hálózatban, a GPC pedig a konvencionális perifériákat kezeli /alfanumerikus display, sornyomtató és a lyukszalag I/O-t helyettesítő floppy diszk/. DPU1 a cartridge diszk fizikai és logikai szintű kezelését látja el / a logikai kezelésbe beleértve a magasszintű adatbázis kezelést is/. A DPU2 és a TPU egy nagyteljesítményű miniszámítógépet és annak a lebegőpontos aritmetikáját emulálja. A közös memória a processzorok közötti adatátviteli bufferként, valamint a DPU1 és a DPU2 programmezejeként szolgál.

Az így kialakított rendszer tulajdonképpen egy általános célú nagy teljesítőképességű számítógép, amely adatbázis-kezelésre is alkalmas. Nagyobb méretű adatbázis esetén a cartridge diszk helyett a jóval nagyobb kapacitású diszk pack alkalmazása kerül előtérbe.

x: reentráns:önmaga megváltoztatása nélküli, többszöri belépésre képes.

A fentiekben bemutatottakon túl további speciális konfigurációk is felépíthetők a GD'80 elemek felhasználásával /pl. több busszal, minden buszon négy nagyteljesítményű mikroprogramozott processzorral, a buszok között kettős hozzáférésű memóriákkal egy úgynevezett "number-cruncher" építhető fel/. Ezeket azonban itt nem ismertetem, hanem egy GD'80 konfigurációkból összeállított, légiforgalmi helyzeteket szimuláló, a földi irányítószemélyzet kiképzését lehetővé tevő oktató rendszer architektúráját mutatom be /29. ábra/. A GD'80 konfigurációk közötti kapcsolatot - a multiprocessing egy további, hierarchikus módszerével - egy nagysebességű koaxiális kábel valósítja meg. Ennek a maximális adatai a következők: a hossza 1000 méter, az adatátviteli sebesség 2 millió bit másodpercenként, az állomások száma 256. A koaxiális kábelhez, amelyet COBUS-nak nevezünk, adatátviteli szempontból egyenrangú állomások csatlakoznak. A nagy átviteli sebesség miatt az illesztőegységek közvetlen memóriahozzáféréssel csatlakoznak a HIF kommunikációs processzorokhoz. A választott vonali protokoll bitorientált /bit stuffing/, blokkos csomagkapcsolt /packet-switching/ adatátvitelt tesz lehetővé a COBUS tetszőleges két állomása között. Az adott célalkalmazásban az állomások száma 9, ezek egymástól mért maximális távolsága 50 méter. Természetesen az adott adatátviteli feladat egy központi gép körül csillagban elhelyezett állomásokkal /pont-pont adatátvitellel és nem buszszerű megoldással is megoldható. Ez a vonali interface-k számának megnövelésével és a központi számítógép processzási képessége jó részének az adatátvitel lebonyolítására történő lekötésével érhető el.

Az oktatóberendezés három típusú GD'80 konfigurációt tartalmaz:

- a./ Alfanyumerikus munkahelyek /pilóták számára/ koncentrátorai /GD'80 KC öt alfanyumerikus display-vel/.
- b./ Grafikus munkahelyek/az oktató és a kiképzendő irányítók számára//GD'80 BT/SSC egyszínű és színes változatai/
- c./ Számítógépes munkahelyek /a légiforgalom szimulálására és a légi helyzetek előkészítésére/ GD'80 DP.

A fenti rendszer teljes kiépítésében összesen három típusú állomást használ 5 féle célra. Ezen három állomástípus között is vannak átfedések a felhasznált hardware és software modulok terén. Ezáltal bizonyos állomások kiesésével, meghibásodásával a rendszer működőképes marad, a meghibásodott állomás funkcióit más állomások vehetik át, esetleg bizonyos funkciók korlátozása árán. A pilóta munkahely koncentrátorok egyikének kiesése esetén öt további pilóta munkahely marad a rendszerben, az oktatói munkahely szerepét átveheti az egyik irányítói munkahely, a szimulátor kiesésével pedig az előkészítő munkahelyre kerülnek a szimulációs programok. Az így kialakított rendszer kedvező megbízhatósági paraméterekkel rendelkezik.

VI. ÚJ TUDOMÁNYOS EREDMÉNYEK:

Eredményeim közül a legfontosabbak a következők:

1. Kidolgoztam egy új általános grafikus display architektúrát, amely a szokásos többfeladatú mini számítógép helyett az egyes funkciók, processzorok közötti optimális szétosztását teszi lehetővé. Az architektúra modulokon alapul. A főbb modulok a következők: buszok, processzorok /standard 8 bites mikroprocesszorok perifériakezelésre, valamint horizontálisan mikroprogramozott célprocesszorok/, tárolók /ROM, RAM/, periféria csatolóegységek és perifériák, háttértárolók, kijelző egységek /képelem generátorok, eltérítő rendszerek, képcső/.

A modulkészlet felhasználásával különböző intelligenciájú grafikus rendszerek állíthatók elő: egyszerű terminál, grafikus számítógép, intelligens terminál, autonóm grafikus rendszer, grafikus szatellit.

A konfigurációk az egyes processzorok jelenlétével vagy hiányával jellemezhetők. Ha az adott processzor hiányzik, a hozzárendelt funkció vagy hiányzik, vagy egy másik processzor veszi át azt, esetleg alacsonyabb hatásfokkal. A kialakított architektúra a grafikus display-ek családjának kialakítását teszi lehetővé. Ezen konfigurációk alkalmasak a számítógéppel segített tervezési /CAD/ feladatok széles körében való alkalmazásra, esetlegesen ilyen feladatok teljes lefedésére. A rendszer egyes családtagjai alkalmasak egyéb speciális grafikus igények kielégítésére is /lokátorrendszerek szintetikus indikátora, légiforgalom irányítás, oktatás, stb./.

A kidolgozott architektúra és a modulkészlet alkalmas más általános vagy speciális számítástechnikai berendezés kialakítására is /time sharing kisszámítógép, terminálkoncentrátor/.

2. Az elterjedt általános buszok /pl. PDP11 UNIBUS, NOVA BUS, R10 MINIBUS, PDP8 OMNIBUS, TPA'70 MIOBUS/ több processzorra történő kiterjesztését dolgoztam ki, amely a szokásos Master-Slave adatátviteli ciklusok mellett a processzorok közötti megszakítások kezelését is lehetővé teszi. A kialakított buszrendszer vezérlőegysége egyszerű autonóm prioritáslogika, amely nem kapcsolódik CPU-hoz, mint a PDP11 vagy a TPA'70 esetében. A buszhoz kapcsolódó egységek "Slave-processzorok", amelyek adatátvitelre és megszakításkérésre igénylik, kapják meg a busz használati jogát. A megszakításkéréseket a megszakítási vektor tartalma alapján a busz vezérlőegység adatátviteli ciklusokká alakítja át és megszakítási információt küld annak a processzornak, amelynek a megszakításkérés szól.

3. A heterogén processzorok közötti kommunikáció optimális megszervezésére különböző multiprocessing módszerek kombinációját alakítottam ki. A kommunikáció alapvető eszközei a közös buszok, amelyek közül az U2 Bus-t a képfelfrissítésre, nagymenyiségű adat-burst-ben történő átvitelére használtam. Az U1 Bus általános adatcserét tesz lehetővé. A közös memória, amelynek a processzorok közötti felosztására új egyszerűsített algoritmust realizáló hardware/software megoldást dolgoztam ki, a képleírás tárolására, az egyes processzorok közötti adatátviteli pufferként, valamint adatok és programok tárolására szolgál. A processzorok közötti adatcsere a közös memórián keresztül történik, a megszakításkérések csak eseményekről szóló jelentéseket hordoznak /például, hogy egy processzor átadta vagy átvette az adatokat a közös memórián keresztül, software hibajelzések, stb./.

A közös memória kettős hozzáférésű, azaz szimultán adatátvitelt tesz lehetővé az U1 és U2 Buson. A standard 8 bites mikroprocesszorok és az általános busz közötti kapcsolat megteremtésére memóriaablak technikát /MIPROBUS WINDOW/ dolgoztam ki.

Ez lehetővé teszi, hogy a mikroprocesszor bizonyos memória területére történő címzések az általános buszon adatátviteli ciklusokká alakuljanak át. Ez a megoldás azt eredményezi, hogy a közös memória a standard 8 bites mikroprocesszorok számára éppúgy kezelhető, mint a saját memóriájuk /időben természetesen van különbség/.

A mikroprogramozható DPU és TPU processzorok közötti szoros együttműködés megvalósítására, a processzorok közötti pillanatnyi sebességkülönbség kiegyenlítésére First-In-First-Out tárolókat, a közvetlen érvényre jutó státuszinformációk átadására kettős hozzáférésű memóriákat alkalmaztam.

4. Horizontálisan mikroprogramozott processzorok családját dolgoztam ki /TPA'70 FPU, GD'80-DCU,-DPU,-TPU/. Ezek valamennyien azonos felépítésű mikroprogram vezérlőegységgel /control path/ rendelkeznek. Az adatkezelő részek /data path/ felépítése viszont esetenként változó is lehet. A mikroprogramozott vezérlőegységhez írható mikroprogramtárat /WCS: Writable Control Sore/ terveztem, amely igen hasznosnak bizonyult a hardware és firmware diagnosztikában. Jelentősége azonban elsősorban a processzorok run-time átmikroprogramozhatóságában és az ugynevezett "user microprogramming"-ben rejlik. Ezen opció segítségével a felhasználó maga alakíthatja ki - mikroprogramozhatja - /makro/ utasításrendszerét, amely az adott feladatra az optimális megoldást nyújtja.

A horizontálisan mikroprogramozott processzorok számára olyan általános architektúrát terveztem és valósítottam meg, amely lehetővé teszi tetszőleges utasításrendszerek emulációját vagy speciális, felhasználó orientált utasításrendszerek kidolgozását. Az emulációval a felhasználói programok hordozhatóságára, a futási időben és memória helyfoglalásában kevésbé effektív magasszintű nyelvekkel szemben olyan alternatívát nyújtok, amely az adott utasításrendszert /amelyben a felhasználói program íródott/ teszi hordozhatóvá.

A mikroprogramozott processzorok kialakítására miniszámítógépre alapozott mikroprogramfejlesztési rendszer szolgál, Ez magában foglalja a mikroprogram cross-assemblereket, -szimulátorokat és a PROM-programozó hardware és software eszközöket. Valamennyi mikroprogram fejlesztő software eszköz a miniszámítógép diszkes operációs rendszere alatt fut és annak valamennyi lehetőségét /editálás, file-kezelés/ kihasználja.

5. A mikroprogram szintjén változatható aritmetika orientált processzorstruktúrát /GD'80-TPU/ dolgoztam ki. Ez azt jelenti, hogy a processzor belső architektúrája mikroprogramozható. Az egyes részprocesszorok egymástól függetlenül dolgozhatnak, vagy tetszőlegesen összekonfigurálhatók. A párhuzamos működést olyan esetekben használtam, amelyekben azonos aritmetikai műveleteket kell /vagy lehet/ elvégezni egymással párhuzamosan különböző operandusokkal /például egy homogén koordinátákkal megadott vektor transzformációja három dimenzióssá, ahol az x, y, z koordinátákat a skálafaktorral kell elosztani/. A csoportos, összefűzött működést a többszörös szóhosszal való műveletvégzés egyszerűsítésére dolgoztam ki.

Az aritmetika orientált processzor architektúráját elsősorban a háromdimenziós /homogén koordinátákban megadott/ vektorokkal és transzformációs mátrixokkal való egyszerű és gyors /real-time/ műveletvégzés céljának megfelelően dolgoztam ki. Ezzel összhangban egy kiegészítő Matrix Multiplier egységet is kialakítottam, amely lehetővé teszi 4×1 vektorok és 4×4 transzformációs mátrixok szorzását, illetve új transzformációs mátrixok létrehozását /a transzformációs mátrixok szorzását, illetve új transzformációs mátrixok létrehozását/. Az egyszeres pontosságú szorzó és szorzandó szorzata a mikroprogram ciklusideje alatt előáll /dupla pontosságú formátumban/, és két részprocesszor összefűzésével még az adott ciklusban eltárolható. A transzformációs

mátrix tárolására a Matrix Multiplier egyik bemenetére mátrixtárolót terveztem /szokásos ugyanis, hogy egy adott képrészletre ugyanazt a transzformációt alkalmazzuk, azaz egy sor vektort ugyanazzal a transzformációs mátrixszal szorozunk/. E mátrixtároló cím- és maszkregisztere lehetővé teszi 2x2, 3x3 és 4x4-es mátrixok megfelelő dimenzióju vektorokkal való egyszerű, automatizált szorzását.

A mikroprogramozás szintjén a transzformációs mátrixszal való szorzáson kívül további grafikus manipulációkat is lehetővé tevő megoldásokat dolgoztam ki. Ezek a manipulációk a következők: a transzformált, duplapontosságú homogén koordinátás vektor átalakítása háromdimenziós vektorrá, hárctdimenziós kivágás /clipping/, perspektív és láthatósági transzformációk már megjeleníthető két dimenziós vektorokká való átalakítás céljából. Algoritmusokat terveztem, amelyek lehetővé teszik, hogy a bemeneti adatokat lebegőpontos formátumban is meg lehessen adni és a TPU ezekkel is képes legyen dolgozni: akár, mint transzformációs processzor a fentiekben leírtaknak megfelelően, csak lebegőpontos számábrázolással, akár mint egyszerű lebegőpontos műveleteket /összeadás, kivonás, szorzás, osztás/ végrehajtó processzor. Hardware és firmware interfacet valósítottam meg a GD'80-DPU és -TPU processzorok között. Ez lehetővé teszi, hogy a processzor-pár különböző feladatok elvégzésére, különböző módokban legyen mikroprogramozható. Alapvetően a DPU mint egy 16 bites miniszámítógép, a TPU pedig, mint annak aritmetikai kiegészítése. A lebegőpontos vagy trigonometrikus utasításokat például a DPU nem értelmezi, hanem átadja végrehajtásra a TPU-nak. Az adott művelet végrehajtását a DPU vagy bevárja, /ekkor az utasítás végrehajtása WAIT módu/ vagy nem várja be. Az utóbbi esetben gondoskodni kell arról, hogy a DPU-nak szóló további utasítások ne számitsanak a TPU által kiszámí-

tott eredményre, vagy amelyik utasítás számít rá, ott a DPU várja be a TPU-t.

Hasonló működést terveztem magasszintű geometriai leírások értelmezésére is. A DPU mint listaprocesszor dolgozik, azaz adatstruktúrát értelmez és képleírássá konvertál. A bevárásai feltételek azonosak, mint a lebegőpontos műveletek esetén. A két processzor közötti pillanatnyi sebességkülönbség kiegyenlítésére az interface-be a két processzor mikroprogramjából kezelhető First-In-First-Out tárolókat terveztem és építettem be. Ezek segítségével a két processzor közötti szinkronizáció egyszerűen megoldható.

6. A mikroprogramozott processzorok speciális üzemmódban való működtetését lehetővé tevő opcionális egységeket alkíttam ki. Ezzel az úgy nevezett "Maintenance módú" működtetéssel lehetővé tettem a processzorok - külső program felügyelete alatt - mikroprogram lépésenként való futtatását, valamint az egyes mikroutasítások között valamennyi belső regiszter és memória tartalmának ellenőrzése céljából való kiolvasását is. A maintenance logika lehetővé teszi a mikroprogramozott processzorok egyes részmoduljainak külön vizsgálatát is azáltal, hogy az egyes részmodulokat lépésenkénti órajellel külön-külön is léptethetjük nem csak közösen.

A lépésenkénti futtatásra, a futás közbeni tesztelésre a standard, fix programu /nem mikroprogramozható/ Graphic Peripheral Controller /GPC/ szolgál. Ez a processzor látja el a GD'80 rendszerekben az ugynevezett "Supervisor Processor" szerepét. A fentiekben leírtakon kívül ez az egység tölti fel bekapcsolás után az írható mikroprogramtárakat. Az általános buszokon /U1, U2/ fellépő buszforgalmi hibák esetén a buszvezérlő egységek /BC1, BC2/ ennek küldenek hibaüzenetet /itt okoznak megszakításkéré-

seket/, és ez az egység gondoskodik a buszhibák megszüntetéséről, a buszok inicializálásáról is.

7. Koaxiális kábelre alapított nagysebességű kommunikációs rendszert terveztem, amely éppúgy alkalmas félduplex pont-pont összekötés megvalósítására, mint a rendszer buszszerű kialakítására. A megvalósított vonali protokoll a kábelhez kapcsolódó tetszőleges két állomás közötti bitorientált, csomagkapcsolt kommunikációt tesz lehetővé.

A koaxiális kábel alapu kommunikációs busz segítségével összekapcsolt GD'80 konfigurációk egy hierarchikus multiprocesszoros rendszert képeznek: a GD'80 konfigurációk önmagukban többprocesszoros rendszerek /ezek valósítják meg a hierarchia első fokozatát/. A hierarchia második lépcsője - a viszonylag nagy sebesség /2 Mbit/sec/ellenére - lazább kapcsolatot jelent az egyes konfigurációk között, mint az első lépcső /32 Mbit/sec/. Ezt tükrözi az előbbi sebességadatokon kívül az egyes lépcsőkben a processzorok, illetve konfigurációk közötti távolság: a GD'80 konfigurációkon belül a maximális buszokon 10 méter, a koaxiális kábelnél kb. 1000 méter.

8. Az alfanumerikus és grafikus megjelenítés két alapvető módjának - a vektoros és a raszteres elvnek - kombinált alkalmazását lehetővé tevő grafikus képmegjelenítő modulokat /generátorokat/ dolgoztam ki. Ezek lehetővé teszik vektoros információ és közvetlen video információ kevert megjelenítését /például egy TV képrészlet "bekeverhető" a vonalas képbe/. Erre a célra szolgál a vektorgenerátor /digitális interpolátor/ kiegészítése, amely lehetővé teszi, hogy megfelelő vízszintes /letapogató/ vektorok ge-

nerálása mellett a közös memóriából kiolvasott közvetlen videoinformáció jelenjék meg a képernyőn.

A kevert megjelenítés egyesíti a két megjelenítési mód előnyeit különösen olyan konfigurációkban, ahol a grafikus rögzítő/hard-copy/eszköz is rászteres elven működik /például printer-plotter/.

Az alfanumerikus megjelenítés lehetőségeinek kibővítésére a felhasználó által futás közben /run-time/ is programozható karakter és szimbólumgenerátort javasoltam. Ez lehetővé teszi a standard karakter- és jelkészlet mellett speciális karakterek és szimbólumok /pl. markerek, és diagramok egyes pontjainak kijelölésére szolgáló centráliszimetrikus szimbólumok/ megjelenítését is.

A vektorok és karakterek megjelenítése mellett lehetővé tettem magasabbrendű függvények megjelenítését is. A szokásos, bonyolult, speciális /random/ hardwaret igénylő megoldások vagy hely-és időigényes software interpolációk helyett a magasabbrendű görbék vektorokkal történő közelítését javasoltam. A közelítő szakaszokat a Display Control Unit mikroprogramja állítja elő és a vektorgenerátor mint bármely normál vektort generálja le azokat. A közelítő szakaszokra bontáshoz különböző algoritmusokat javasoltam.

VII. AZ ISMERTETETT TUDOMÁNYOS EREDMÉNYEK REALIZÁCIÓJA ÉS REFERENCIÁI

A GD'80 család realizációja egy sor létező berendezés, amelyeknek a felépítését az V. fejezetben ismertettem. Ezek többsége világviszonylatban is újdonságnak tekinthető. A következő eredmények biztosítják a világszinvonal elérését.

- nagyfelbontású grafikus megjelenítés, nagyteljesítményű grafikus display
- user mikroprogramozás, írható control store /WCS/, mikroprogram szinten változtatható processzor struktúra, mikroprogram és mikroprocesszor programfejlesztési rendszer
- multiprocessing módszerek használata a GD'80 konfigurációkon belüli processzorok közötti együttműködésre
- nagysebességű helyi adatátviteli hálózat, mint a multiprocessing újabb hierarchikus lépcsőfoka - az egyes GD'80 konfigurációk közötti együttműködésre

A GD'80 konfigurációk kis sorozatban az MTA SzTAKI-ban kerülnek kísérleti gyártásra. A gyártást a Híradástechnika Szövetkezet veszi át.

A GD'80 konfigurációkkal és ezek hierarchikus kapcsolatával légiforgalmi szimulációs oktató rendszert realizáló berendezést az LRI megrendelésére, a Minisztertanács 3299/77. számú rendelete alapján az MTA SzTAKI Számítógépes Grafika Osztályán fejlesztjük és építjük meg.

Az NDK szerszámgépiparának vezető intézményei/Uniformtechnik Erfurt, Forschungsinstitut Karl-Marx Stadt, TU Dresden, Schiffbau Rostock, Universität Rostock, TH Karl-Marx Stadt/ és az MTA

SzTAKI között létrejött hosszútávú együttműködés keretében az NDK szerszámgépipara a GD'80 konfigurációkat standard tervezői munkahelyként kívánja használni.

A KGST számítástechnikai együttműködési szervei közül az MSzR második sora grafikus eszközeinek nomenklaturájában /amelyet a KGM és a SzU Munkaügyi Minisztérium közötti együttműködés keretében az MSzR szovjet bázisintézményével, az INEUM-mal közösen alakítottunk ki/ helyet kaptak a GD'80 grafikus display-ek és terminálkonfigurációk.

Az ESZR keretében - a távadatfeldolgozási témakör részeként - a GD'80 terminál és szatellit konfigurációk ESZR hálózati terminálként fognak üzemelni.

Az AMT távlati terveiben a GD'80 konfigurációk standard interaktív munkahelyként szerepelnek.

A GD'80 konfigurációk iránt a piaci kereslet is igen nagy. Első kiállításon való megjelenése után fél évvel a megrendelések száma elérte az ötvenet /ezt természetesen már nem az MTA SzTAKI, hanem a HT gyártja le/.

A szocialista megrendeléseken kívül több kisebb-nagyobb tőkés cég is érdeklődik a GD'80 rendszerek iránt. Az ARISTO például a HP és Tektronix display-eket kívánja velük kiváltani. Mások viszont a közös értékesítésben, esetleges továbbfejlesztésben, rendszerek létrehozásában látnak fantáziát /Productivity International, INC., Integrated Computer Aids of Norway A/S/. A CAD/CAM Digest folyóiratban ismertető cikk jelent meg a GD'80-ról. Mindezek az érdeklődések anélkül nyilvánultak meg, hogy a GD'80 konfiguráció tőkés kiállításon megjelent volna, vagy külkereskedelmi szerveink bármilyen piackutatást végeztek volna.

VIII. AZ ÉRTEKEZÉS TÉMAKÖRÉBŐL KÉSZÜLT PUBLIKÁCIÓK JEGYZÉKE

- 1./ Forgács T., Verebély P.: "GD71/T grafikus display és bemeneti eszközök: hardware programozási kézikönyv"
MTA SzTAKI, 1974.
- 2./ Forgács T., Verebély P.: "CDC/DRI disc interface hardware programozási kézikönyv"
MTA SzTAKI, 1974.
- 3./ Verebély P.: "Többprocesszoros rendszerek kialakítási lehetőségei buszarchitektúrával rendelkező kisszámítógépek esetén"
MTA SzTAKI belső anyag 1975.
- 4./ Verebély P.: "MIOBUS Repeater és Interprocessor Busswitch a TPA 7025-re"
MTA SzTAKI belső anyag 1975.
- 5./ Verebély P.: "Másodfokú függvénygenerátor és forgatóberendezés"
MTA SzTAKI belső anyag 1974.
- 6./ Verebély P.: "Három dimenziós függvénygenerátor és térbeli forgatóberendezés"
MTA SzTAKI belső anyag 1975.
- 7./ P. Verebély: "META 4A Floating Point Processor"
Brown University, 1976.
- 8./ P. Verebély: "Floating Point Processor for BUGS. Principles of Operation"
Brown University, 1976.
- 9./ Verebély P., Forgács T., Janssen M., Rózsa Gy., Kramer G., Gallai I., Lábadi A., Szántó Gy., Gerhardt G., Hermann Gy., Forján Cs., Darvas P.: "Vezérlőegység grafikus megjelenítő"

egységekhez"

OTH, MA-2643, Szolgálati szabadalom, 1977.

- 10./ Verebélly P.: "MIPROBUS window" Hardware programozási leírás
MTA SzTAKI 1977.
- 11./ Verebélly P.: "TPA70 Memory Management"
Hardware programozási leírás
MTA SzTAKI, 1977.
- 12./ Janssen M., Verebélly P.: "FPU mikroprogramozása"
Hardware programozási leírás
MTA SzTAKI, 1977.
- 13./ P. Verebélly: "Interaktive Konstruktion im Maschinenbau"
Symposium der Chemianlagebau, Leipzig, 1976.
- 14./ Szántó Gy., Krammer G., Verebélly P.: "A GD80 grafikus
display család rendszertervének vázlata".
MTA SzTAKI belső anyag 1976.
- 15./ Verebélly P.: "A GD80 specifikációjának összefoglalása"
MTA SzTAKI belső anyag 1977.
- 16./ P. Verebélly: "Gerätechnik der GD80 Familie"
Conference on Computer Graphics'78, Smolenice
1978.
- 17./ J. Hatvany, P. Verebélly, G. Krammer, I. Gallai: "A Case
for Advanced Graphics Family Planning: GD80"
MTA SzTAKI, gyártmányismertető, 1978.
- 18./ Lőrincze G., Salamon M., Gallai I., Szántó Gy., Janssen M.,
Verebélly P.: "A mikroprogramozás modern eszközei."
Mérnök Továbbképző Intézet 136. előadássorozat,
1978.

- 19./ Verebélly P.: "A HIF és GPC mikroprocesszorok. Processzorok kommunikációja az UNIBUS-on"
MTA SzTAKI belső anyag 1978.
- 20./ P. Verebélly: "HIF /Host Interface/ and GPC /Graphic Peripheral Controller/ microprocessors: a preliminary description"
MTA SzTAKI belső anyag 1978.
- 21./ Verebélly P.: "GD80 Periféria Interface Busz"
MTA SzTAKI, specifikáció, 1978.
- 22./ Verebélly P.: "Transformációs processzor"
MTA SzTAKI belső anyag 1978.
- 23./ Verebélly P., Szántó Gy.: "DCU/DPU mikroprogramozása"
MTA SzTAKI belső anyag 1978.
- 24./ Verebélly P.: "A GD80 rendszertechnikája"
Előadás a Neuman János Számítástudományi Társaság ülésén, 1978.
- 25./ Verebélly P., Forgács T., Janssen M., Rózsa Gy., Krammer G., Gallai I., Lábadi A., Szántó Gy., Gerhardt G., Hermann Gy., Fórján Cs., Darvas P.: "Berendezés legalább egy mágneslemezegység adatfeldolgozó rendszerhez való csatlakoztatására"
OTH, MA-2746 szolgálati szabadalom, 1978.
- 26./ Krammer G., Gallai I., Darvas P., Fórján Cs., Janssen M., Szántó Gy., Verebélly P., Rózsa Gy., Bálint Z., Perjés M., Hatvany J.: "Többprocesszoros adatfeldolgozó rendszer"
OTH, MA-2983 szolgálati szabadalmi bejelentés, 1978.

- 27./ Gallai I., Janssen M., Szántó Gy., Verebély P., Rózsa Gy.,
Bálint Z.: "Általános célú mikroprogramozott adatfeldolgozó egység"
OTH, MA-3047 szolgálati szabadalmi bejelentés,
1978.
- 28./ Fórján Cs., Verebély P., Rózsa Gy.: Mikroprogramozott vezérlésű digitális görbe- és pozíció generáló egys.
OTH, MA szolgálati szabadalmi bejelentés,
1979.
- 29./ P. Verebély: "Combination of Multiprocessing Methods in a Multiprocessor Graphic System"
EUROMICRO'79, Göteborg, 1979
- 30./ P. Verebély: "The GD80 Graphic Display Family"
SID Proceedings, 1979.
- 31./ J. Hatvany, P. Verebély: "Distributed Intelligence in the GD80 Graphic Display System"
SID Symposium, Chicago, 1979.
- 32./ P. Verebély: "GD80, a True Multiprocessor System for Computer Graphics"
IAP'79, Budapest, 1979.
- 33./ I. Gallai, M. Janssen, Gy. Szántó, P. Verebély: "A Family of Microprogrammed Processors"
IAP'79, Budapest, 1979.
- 34./ P. Darvas, A. Lábadi, P. Verebély: "A Peripheral Controller Microprocessor for the PDP11 Series Minicomputers"
IAP'79, Budapest, 1979.
- 35./ Reé E., Bozsó T., Ábrahám I., Ladányi Gy., Verebély P.:
"Számítógépek adatátviteli sínjeit átkapcsoló egység"
OTH, MA- szolgálati szabadalmi bej., 1979.

- 36./ Darvas P., Verebélly P.: "Oktatóberendezés légiforgalmi irányítók kiképzésére"
Műszaki ajánlat, MTA SzTAKI, 1978.
- 37./ Darvas P., Verebélly P.: "COBUS: a local packet-switching network for the NEWMANISM"
MTA SzTAKI belső anyag 1967.
- 38./ Darvas P., Verebélly P.: "Nagysebességű helyi adatátviteli hálózat"
OTH, MA- szolgálati szabadalmi bejelentés,
1979.
- 39./ Verebélly P.: "Multi-mikroprocesszoros rendszerek fejlesztése"
MATE "Mikroprocesszoros rendszerek fejlesztésének eszközei és módszerei" a tanfolyam előadása,
1979. március
- 40./ Burgolits L., Payper G., Verebélly P.: "Tárolócsöves ábramegjelenítő"
Mérés és Automatika 1972, szeptember
- 41./ Verebélly P.: "GD'71 grafikus display"
Számítástechnika 1973/5.
- 42./ Vörös G.J., Lábadi A., Verebélly P.: "Feedback BRM for Control Purposes" MIMI'73, Zürich, 1973.
- 43./ Vörös G. J., Verebélly P.: "Eljárás és berendezés digitális növekményes függvénygenerálásra parahisztémás integrátorokkal"
OTH MA-2333 Szolgálati szabadalom
- 44./ Verebélly P.: "GD'80 System Components Manual"
1979. MTA SzTAKI

IX. IRODALOMJEGYZÉK

- 1./ G.A. Anderson, E.D. Jensen: "Computer Interconnection Structures",
ACM Computer Surveys, Vol. 7. No 3
/1975/ pp. 197-213
- 2./ "Adage GP/400 User'S Reference Manual" ADAGE Inc.,
Boston, Mass. 1975
- 3./ G. Adams, T. Rolander: "Design Motivations for Multiple Minicomputer Systems",
Computer Design, March, 1978, ppl 81-89
- 4./ P. Anagostopoulos, G. Sockret, G.M. Stabler, A. van Dam, M. Michel: "Computer architecture and instruction at design"
National Computer Conference, 1973
- 5./ T. Adams, S.M. Smith: "How bit-slice families compare: Part 1, evaluating processor elements"
Electronics, August 3, 1978, pp. 91-98
- 6./ T. Adams, S.M. Smith: "How bit-slice families compare: Part 2, sizing up the microcontrollers"
Electronics, August 17, 1978, pp. 96-102
- 7./ T. Adams, S.M. Smith: "Post script on bit slice families: microcontrollers serve many needs"
Electronics, August 31, 1978, pp. 138-139
- 8./ J. Acree, A. Lynch: "Ring network architecture supports distributed processing"
Data Communications, March/April 1976,
pp. 51-55
- 9./ N. Abramson: "The Aloha system"
AFIPS Conf. Proc. Vol. 37, 1970 FJCC,
pp. 281-285

- 10./ R.L. Askenhurst, R.M. Vonderohe: "A hierarchical network"
Datamation, Feb., 1975, pp. 40-44
- 11./ R.G. Arnold, E.W. Page: "A hierarchical restructurable
multi-microprocessor architecture"
3rd Annual Symposium on Computer Architecture,
1976, pp. 40-45
- 12./ M. Andrews: "A bit slice architecture for microprogrammable
machines"
9th Annual Workshop on Microprogramming,
1976, pp. 5-8
- 13./ "Am 2900 Bipolar Microprocessor Family"
Advanced Micro Devices Inc., 1976
- 14./ "Am 2903 Bipolar Microprocessor Slice, Am 2910 Micro-
program Controller"
Advanced Micro Devices Inc., 1978
- 15./ B.R. Borgerson: "The Viability of Multimicroprocessor
Systems"
Computer, January 1976, pp. 26-30
- 16./ C.G. Bell, W.A. Wulf: "C.mmp - A Multimicroprocessor"
AFIPS Proceedings FJCC, 1972, Vol. 41, Part II.
pp. 765-777
- 17./ G.H. Barnes: "The ILLIAC IV. Computer"
IEEE Transaction on Computers /August 1968/
pp. 746-757
- 18./ N.A.Ball, H.Q. Foster, W.H. Long, I.E. Sutherland,
R.L. Wigington: "A Shared Memory Computer Display System"
IEEE Transaction on Electronic Computers,
EC-15, 5, 751, October 1966

- 19./ "The Brown University Graphic System"
Brown University, Providence, R.I. Febr. 1972
- 20./ J.D. Bagley: "Microprogrammable Virtual Machines"
Computer, February 1976
- 21./ P. Breeden: "A Control Storage Simulator for Debugging
Microprograms"
Computer Design, April 1974
- 22./ Baer: "Multiprocessing Systems"
IEEE Transactions on Computers, Dec. 1976,
pp. 1271-1277
- 23./ S. Buchwald: "Microcode Increases Minicomputer Processing
Capability"
Computer Design, October 1977, pp. 91-99
- 24./ J.F. Bartram, R.R. Ramseyer, J.M. Heines: "Fifth
Generation Digital Sonar Signal Processing"
EASCON'76, pp. 91-A-91-G
- 25./ P.P. Budnik, Jr.: "Techniques for Parallel Computer
Design"
University of Illinois, 1975
- 26./ I.W. Cotton: "Network graphic attention handling"
ON-LINE 72 Conference Proceedings pp. 465-490
27. / I.W. Cotton: "Some trade-offs in the design of mini-
computer based graphic systems"
Tech. Note, National Bureau of Standards
28. / I.W. Cotton: "Cost-benefit analysis of computer graphics
systems"
Tech. Note 826, National Bureau of Standards,
April 1974.

- 29 ./ D.V. Collins, E.R. Garren, L.L. Lazar: "Bit-Slice
Microprocessors, PLA's and Microprogramming"
Integrated Computer Systems, Inc. Course Notes
187, 1976
- 30 ./ R.C. Chen, P.G. Jessel, R.A. Patterson: "MININET: A Micro-
processor-Controlled Mininetwork"
Proceedings of the IEEE, Vol. 64. No. 6. June
1976
- 31 ./ W.A. Curtin: "Multiple Computer Systems"
Advances in Computers, Vol. 4. F L Alt and
M. Rubinoff Eds New York Academic Press 1973
- 32 ./ G.F. Casaglia: "Nanoprogramming rs. Microprogrammong"
Computer, January 1976
- 33 ./ G. Cavanaugh: "Fast Multipliers using TTL Reed-Only
Memories"
Texas Instuments Application Report, CA-172
- 34 ./ R. Capece, N. Mokhoff: "Technology Update"
Electronics, October 26, 1978, pp. 110-161
- 35 ./ F.F. Coury: "Microprogramming and Writable Control Store"
HP Journal, July 1972, pp. 16-20
- 36 ./ D. Cohen, M.T. Liu: "Emulation of Computer Networks by
Microprogrammable Microcomputers"
Seventh Annual Workshop on Microprogramming
1974, pp. 159-167
- 37 ./ R.G. Cooper: "Micromodules: Microprogrammable Building
Blocks for Hardware Development"
First Annual Symposium on Computer Architecture
1973, pp. 221-237

- 38 ./ E.D. Carlton: "Graphics terminal requirements for the 1970's"
Computer, Vol. 9., No. 8, Aug. 1976. pp. 37-45
- 39 ./ J.W. Conway: "Approach to unified bus architecture sidesteps inherent drawbacks"
Computer Design, Jan. 1977, pp. 71-76
- 40 ./ "Interactive display system supports distributed processing and provides powerful graphics capabilities"
Computer Design, Dec. 1976, pp. 28-29
- 41 ./ J. Clymer: "Use 4-bit slices to design powerful micro-programmed processors"
Electronic Design, 10 May 1977, pp. 62-71
- 42 ./ O. Caprani, U.H. Jensen, V. Ougaard: "Microprocessors connected to a common memory" Microprocessing and Microprogramming, 1973, pp. 175-181
- 43 ./ T.C. Chen: "Parallelism, Pipelining and Computer Systems"
AFIPS Proceedings SJCC, 1971, pp. 365-372
- 44 ./ J.J. Capowski: "The Matrix Transform Processor"
IEEE Transactions on Computers, Vol. C-25, No.7 pp. 703-712, July 1976
- 45 ./ J.C. Conway: "Hardware Approaches to Microprogramming with Bipolar Microprocessors"
Computer Design, August 1978, pp. 83-91
- 46./ P.M. Davies: "Readings in microprogramming"
IBM Systems Journal, Vol. 11, 1972
pp. 16-40
- 47./ S. Davis: "Computer Data Displays"
Prentice-Hall, Englewood Cliffs, 1969

- 48./ R.L. Davis, S. Zucker: "Structure of a multiprocessor using microprogrammable building blocks"
Proc. of National Aerospace Electronics Conf.
Dayton, pp. 186-200, May 1971
- 49./ R.A. DiCurcio: "Computerized Graphics - Present and Future"
Automation, May 1973, pp. 48-52
- 50./ "All About Graphic Display Devices"
DATAPRO Research Corp., 1978
- 51./ "META 4 Computer System Microprogramming Reference Manual"
Digital Scientific Corporation, 1971
- 52./ "8020 Floating Point Processor Unit"
Data General Corporation, 1972
- 53./ "DEC Graphic 11: VS 60 Reference Manual"
Digital Equipment Corp., Maynard, Mass. 1976
- 54./ "GT40 Graphic Display System"
Digital Equipment Corp., Maynard, Mass. 1972
- 55./ "Peripherals and Interface Handbook"
Digital Equipment Corp., Maynard, Mass. 1976
- 56./ E.W. Dijkstra: "Solution of a Problem in Concurrent Programming Control"
Communications of the ACM, Sept. 1965, pp. 369
- 57./ "Microprocessors take over execution and overhead of full- and half-duplex protocols"
Data Communications, July/August 1976, pp.55-56
- 58./ "Designing a multi- μ P-based computer family"
Digital Design, Dec. 1976, pp. 18-20

- 59./ J.F. Eastman, D.R. Wooten: "A general purpose, expandable processor for real-time computer graphics"
Computers and Graphics, Vol. 1, pp. 73-77
- 60./ Electronics review: "Computer alters its architecture fast via new control"
Electronics, August 8, 1974, pp. 39-40
- 61./ P.H. Enslow: "Multiprocessors and Parallel Processing"
John Wiley and Sons, New York, 1974
- 62./ "Line Drawing System Model 1: System Reference Manual"
Evans and Sutherland Computer Corporation,
Salt Lake City, 1970
- 63./ "Line Drawing System Model 2: System Reference Manual"
Evans and Sutherland Computer Corporation,
Salt Lake City, 1971
- 64./ "The Picture System"
Evans and Sutherland Computer Corporation,
Salt Lake City, 1973
- 65./ P.H. Enslow: "Multiprocessor organization - a survey"
Comput. Surv., March 1977, pp. 103-129
- 66./ J.D. Foley: "A tutorial on satellite graphics systems"
Computer, Vol. 9, No. 8, pp. 14-21, Aug. 1976
- 67./ J.D. Foley: "An approach to the optimum design of computer graphics systems"
Communications of the ACM 14, 6 pp. 380-390
- 68./ M.J. Flynn: "A Multiple Instruction Stream Processor with Shared Resources"
Proceedings of Conference on Parallel Processing, Monterey, CA. 1968, Spartan Press, 1970, pp. 251-286

- 69./ H. Fleischer, L.I. Maissel: "An introduction to array logic"
IBM J. Research and Development, Vol. 19.
pp. 98-109, March 1975
- 70./ W.D. Farmer, E.E. Newhall: "An experimental distributed switching system to handle to handle bursty Computer Traffic"
1st Symp. in the Optimization of Data Communications Systems, 1969
- 71./ "AP-120B Array Transform Processor. Processor Handbook"
Floating Point Systems 1975
- 72./ "AP-120B Array Transform Processor. Software development package manuals"
Floating Point Systems 1975
- 73./ S.M. Fuller, V.R. Lesser, C.G. Bell, Ch. Kaman:
"Microprogramming and its relationship to emulation and technology"
7th Annual Workshop on Microprogramming, 1974
- 74./ D.J. Farber: "A ring network"
Datamation, Feb. 1975, pp. 44-46
- 75./ M.J. Flynn, R.F. Rosin: "Microprogramming: An Introduction and a Viewpoint"
IEEE Transactions, Vol. C-20, No. 7, July 1971
- 76./ R. Frankenberg: "User microprogramming"
Mini-Micro Systems, July 1977, pp. 46-48
- 77./ S.H. Fuller, L. Raskin, P.I. Rubinfe, P.J. Sindhu, R.J. Swan: "Multi-microprocessors: an overview and working example"
Proc. IEEE, Feb. 1978, pp. 216-228

- 78./ M. Gonauser: "A Distributed Graphics System"
CAD'78, Brighton, UK, March 1978, pp. 557-560
- 79./ M. Gonauser, E. Hörbst, B. Will: "Distributed Intelligence
in an interactive graphic system"
Interactive Design Systems Conference, 1977
- 80./ M. Gonauser, E. Hörbst, B. Will: "The Interactive Graphic
System GMB 300"
Interactive Design Systems Conference, 1977
- 81./ A. Grasselli: "The Design of Program Modifiable Micro-
-Programmed Control Units"
IRE Transactions on Electronic Computers EC-11,
No. 6, pp. 334-339, 1962
- 82./ P. Gordon, S. Stallard: "Microprogrammed CPU Architecture
Offers User-Alterable Minicomputer Performance"
Computer Design, June 1978, pp. 91-100
- 83./ J.D. Grimes: "Distributed Processing Concepts Using
Microprocessors"
14th IEEE Computer Society International
Conference, 1977, pp. 140-144
- 84./ H.R. Houston: "Large Scale interactive graphics, systems
requirements"
ON-LINE 72 Conference Proceedings, pp. 647-658
- 85./ S.S. Huston: "Microprogramming: Principles and Practice"
Prentice Hall, Inc., 1970
- 86./ I.N. Hooton, R.C. Barnes: "A Standard Data Highway for
On-Line Computer Applications"
Proceedings of the 1968 Fall Joint Computer
Conference, Vol. II., pp. 1077-1087

- 87./ R. Hostovsky: "Design of a Display Processing Unit in a Multiterminal Environment"
University of Illinois Report 343, July 1969
- 88./ V.C. Hagamacher, J. Gavilan: "High Speed Multiplier/Divider Iterative Arrays"
1973 Sagamore Computer Conference on Parallel Processing, pp. 91-100
- 89./ K.S. Hojberg: "One-Step Programmable Arbiters for Multi-processors"
Computer Design, April 1978
- 90./ L. Higbil: "Applications of Vector Processing"
Computer Design, April 1978, pp. 139-145
- 91./ L.C. Hobbs, D.J. Theis: "Parallel Processor Systems, Technologies and Applications"
Spartan Books, 1970
- 92./ J. Hyde: "Multiprocessing"
Microprocessors, Aug. 1977, pp. 385-388
- 93./ J.A. Harris, D.R. Smith: "Hierarchical multiprocessor organization"
4th Annual Symposium on Computer Architecture, 1977, pp. 41-48
- 94./ S. Hoener, W. Rochder: "Efficiency of a multi-microprocessor system with time shared busses"
Microprocessing and Microprogramming, 1977, pp. 35-42
- 95./ H. Hiraishi, K. Kawakubo, S. Yajima: "The design of a microprogrammed graphic display processor with pipeline organization"
Microprocessing and Microprogramming, 1977, pp. 66-73

- 96./ "IMLAC PDS-1 Users Manual"
IMLAC Corporation, Waltham, Mass. 1969
- 97./ "Multiprocessor Systems"
INFOTECH State of the Art Report
- 98./ R.N. Ibett: "The MU5 instruction pipeline"
The Computer Journal, Vol. 15, No. 1,
pp. 42-50, Jan. 1972
- 99./ J.P. Ihnat, T.G. Rauscher, B.P. Shay, H.H. Smith,
W.R. Smith: "The use of two levels of parallelism to
implement an efficient programmable signal
processing computer"
1973 Sagamore Conference on Parallel Processing
pp. 113-119
- 100./ R. Jaeger: "Microprogramming: A General Design Tool"
Computer Design, August 1974
- 101./ "INTEL MULTIBUS Interfacing"
AP-28, INTEL Corp., Santa Clara, 1977
- 102./ E.C. Joseph: "Innovations in Heterogeneous and Homogeneous
Distributed - Function Architectures"
Computer, March 1974
- 103./ E.C. Joseph: "Distributed processing architecture - past,
present and future trends"
Distributed Systems: international state of the
art report, 1976, pp. 319-347
- 104./ H. Jackson: "Multiprocessing: Access of common memory"
Microprocessing and Microprogramming, 1977
pp. 158-167

- 105./ H. Kerr: "Microprocessors in Graphic Display Systems"
EE/Systems Engineering Today, Vol. 32, No. 12,
pp. 70-72 /December 1973/
- 106./ E.Y. Lynn, J.D. Schoeffler, C.W. Rose: "Distributed Micro-
computer Data Acquisition"
Instrumentation Technology, January 1975,
pp. 55-61
- 107./ M.H. Lewin: "An Introduction to Computer Graphic Terminals"
Proceedings IEEE, 1544, September 1967,
- 108./ R.C. Larkin: "A minicomputer multiprocessing system"
Proc. of Computer Designers Conference
Anaheim, Calif., Jan. 1971, pp. 231-235
- 109./ B.H. Liebowitz: "Multiple Processor Minicomputer Systems -
Part 1: Design Concepts"
Computer Design, October 1978, pp. 87-95
- 110./ B.H. Liebowitz, J.H. Carton: "Tutorial: Distributed
Processing"
IEEE Computer Society, Sept. 1977, Chap. 8
- 111./ S.Y. Lau: "Bit-slice microprogramming saves software
compatibility"
EDN, 5 March 1978, pp. 42-46, 20 March 1978,
pp. 68-74
- 112./ S.W. Miller: "Display Requirements for Future Man-Machine
Systems"
IEEE Transaction on Electron Devices , Vol. ED-18
No. 9, pp. 616-628
- 113./ T.H. Myer, I.E. Sutherland: "On the design of display
processors"
Communications of the ACM, Vol. 11, No. 6
/June 1968/ pp. 410-414

- 114./ "M10800: The High Performance LSI Processor Family"
MOTOROLA Semiconductors Inc., 1977
- 115./ J.R. Mick, J. Brick: "Microprogramming Handbook"
Advanced Micro Devices, Inc. 1976
- 116./ J.R. Mick, R. Levy, J. Springer: "Digital Signal Processing
Handbook"
Advanced Micro Devices, Inc., 1976
- 117./ R.M. Metcalfe, D.R. Boggs: "Ethernet: Distributed Packet
Switching for Local Computer Networks"
Communications of the ACM, July 1976, Volume 19
Number 7, pp. 395-404
- 118./ W.G. Matheson: "User Microprogrammability in the HP-21MX
Minicomputer"
Seventh Annual Workshop on Microprogramming,
Palo Alto, 1974, pp. 168-177
- 119./ C. Machover, M. Neighbors, Ch. Stuart: "Graphics displays"
IEEE Spectrum, August 1977, pp. 24-32
- 120./ C. de Moncuit: "Interprocessor communication with
decentralized bus"
EUROMICRO Newsletters, July 1977, pp. 9-13
- 121./ R.W. Macmillan: "A method of intermodule communication in
a multimicrocomputer network"
Microprocessing and Microprogramming, 1977,
pp. 202-209
- 122./ W.M. Newman: "Raster scan graphics in CAD"
IFIP Working Conference CAD Systems, 1976
pp. 369-390

- 123./ W.M. Newman: "Display Procedures"
Communications of the ACM, 14, 10, 651,
October 1971
- 124./ W.M. Newman, R.F. Sproull: "Principles of Interactive
Computer Graphics"
McGraw-Hill, 1974
- 125./ W.M. Newman: "Trends in Graphic Display Design"
IEEE Transactions on Computers
Vol. C-25, No. 12, pp. 1321-1325, Dec. 1976
- 126./ W.M. Newman R.F. Sproull: "An Approach to Graphics
System Design"
IEEE Proc. Vol. 62, pp. 471-483
- 127./ "QM-1 Hardware Level User's Manual"
Nanodata Corporation, 1972
- 128./ R.W. Nowlin, D. Gustafson: "A Microprogrammed Machine
Architecture for Efficient Matrix Multiplication"
9th Annual Workshop on Microprogramming 1976,
pp. 56-61
- 129./ R.N. Nilsen: "Distributed-Function Computer Architecture"
Computer, March 1974
- 130./ J. Nemec, S.Y. Lau: "Bipolar μ P's: an introduction to
architecture and applications"
EDN, Sept. 20 1977, pp. 63-67
- 131./ S.M. Ornstein, W.R. Crowther, M.F. Kralej, R.D. Bressler,
A. Michel, F.E. Heart: "Pluribus: - a reliable multi-
processor"
National Computer Conference 1975
pp. 551-559

- 132./ B. Parasuraman: "Hardware Multiplication Techniques for Microprocessor Systems"
Computer Design, April 1977, pp. 75-82
- 133./ B. Parasuraman: "High Performance Microprocessor Architectures"
Proceedings of the IEEE, Vol. 64, No. 6,
June 1976, pp. 851-859
- 134./ B. Parasuraman: "Pipelined architectures for microprocessors"
COMPCON Fall 74, Digest of Papers,
pp. 225-228, Sept. 1974
- 135./ "MIPROC-16 System Reference Manual"
Plessey Microsystems 1976
- 136./ R.O. Parker, J.H. Kroeger: "Algorithm Details for the AM 9511 Arithmetic Processing Unit"
Advanced Micro Devices, Inc., 1978
- 137./ D. Philips, A. Goodman: "Slave microcomputer lightens main microprocessor load"
Electronics, July 7, 1977. pp. 109-112
- 138./ J. Pathak: "Time-Sharing Offers an Alternative to Multiprocessor Systems"
Computer Design, July 1977, pp. 95-104
- 139./ N.R. Powell, J.M. Irwin: "Integrated Functional Processors in Signal Processing"
EASCON'76, pp. 92-A - 92-D
- 140./ J.L. Posdamer: "Concurrent Processing for Computer Graphics"
Computer and Graphics, Vol. 2, pp. 259-263

- 141./ C.B. Parker: "High-speed interactive graphical display system"
Proc. Inst. Electr. Eng., Vol. 123, No. 12,
Dec. 1976, pp. 1293-1298
- 142./ J.T. Quatse, P. Gaulene, D. Dodge: "The external access network of a modular computer system"
Spring Joint Computer Conference, 1972
pp. 783-780
- 143./ J. Raymond, D.K. Banerji: "Using a microprocessor in an intelligent graphics terminal"
Computer, April 1976, pp. 18-25
- 144./ V.K. Ravindran: "Characterization of Multiple Microprocessor Networks"
IEEE Computer Society International Conference
Digest of Papers /1973/, pp. 133-137
- 145./ G. Reyling, Jr.: "Performance and Control of Multiple Microprocessor Systems"
Computer Design, Vol. 13, No. 3, /March 1974/
pp. 81-86
- 146./ J. Rattner, J-C. Cornet, M.E. Hoff, Jr.: "Bipolar LSI computing elements usher in new era of digital design"
Electronics, September 5, 1974, pp. 89-96
- 147./ G. Reyling, Jr.: "Considerations in Choosing a Microprogrammable Bit-Slice Architecture"
Computer, July 1974, pp. 26-29
- 148./ K. Rothmuller: "Task Partitioning in Programmable Logic Systems"
Computer, January 1976, pp. 19-24

- 149./ J. Rattner: "Building block microprocessors"
COMPCON Spring 75, Digest of Papers, Feb. 1975
pp. 79-82
- 150./ R.F. Rosin: "Contemporary Concepts of Microprogramming
and Emulation"
Computing Surveys, Vol. 1, No. 4, Dec. 1969
pp. 197-212
- 151./ C.V. Ramamoorthy, M. Tsuchiya: "A Study of User-micro-
programmable Computers"
AFIPS Conf. Proc. /SYCC/ Vol. 36, pp. 165-181
1970
- 152./ W.B. Riley: "Minicomputer Networks - A Challenge to
Maxicomputers?"
Electronics, March 29, 1971, pp. 56-62
- 153./ "RAYASM: General Purpose Microcode Assembler"
Raytheon Corp., Semiconductor Division 1976
- 154./ R.R. Ramseyer: "Multi-Micro Processor Implementation of
General Purpose Mainframe CPU Systems"
MS Thesis, University of Pennsylvania, 1976
- 155./ K. Rallapalli: "Anatomy of a FIFO"
Progress, May/June 1977, pp. 10-12
- 156./ S.F. Reddaway: "DAP - A Distributed Array Processor"
First Annual Symposium on Computer Architecture
1973, pp. 61-65
- 157./ G.A. Rose: "Intergraphic: a microprogrammed graphical
interface computer"
IEEE Transactions on Electronic Computers
EC-16, No. 6. pp. 773-784

- 158./ G.E. Rossmann, M.J. Flynn, R.M. McClure, N.D. Wheeler:
"The Technical Significance of User Micro-
programmable Systems"
Microprogramming - a Tutorial on the Queen
Mary, pp. 249-293, 1975
- 159./ P.M. Rusto: "An interface for multi-microcomputer
systems"
13th IEEE Computer Society International
Conference, 1976, pp. 277-282
- 160./ H. Rothlisberger: "A standard for multiprocessor
architecture"
Microprocessing and Microprogramming, 1977,
pp. 23-24
- 161./ S. Sherr: "Display parameters"
Proc. SID, Vol. 10, Winter 1969, pp. 57-61
- 162./ I.E. Sutherland: "A Head Mounted Three Dimensional
Display"
Proc. of the Fall Joint Conference, 1968
pp. 757-764
- 163./ R.F. Sproull, I.E. Sutherland: "Clipping Divider"
Proceedings of the Fall Joint Conference, 1968
pp. 765-776
- 164./ G.W. Schulz: "Designing Optimized Microprogrammed
Control Sections for Microprocessors"
Computer Design, Vol. 13, No. 4, /April 1974/
pp. 119-124
- 165./ L.E. Shar, E.S. Davidson: "A Multiminiprocessor System
Implemented Through Pipelining"
Computer, February 1974, pp. 42-51

- 184./ R.T. Thomas: "Computer organization for allowing dynamic user microprogramming"
SIGMICRO Newsletter 4, 2, July 1973 pp. 28-42
- 185./ E.A. Torrero: "Bipolar bit slice μ P's shrink the size and cost of minis and controllers"
Electronic Design, May 10, 1976
- 186./ "MPY 16 A, B 16 x 16 Parallel Multiplier. Preliminary Information"
TRW 1978
- 187./ H.D. Toong: "MICROSTAR: a microprocessor controlled distributed minicomputer network"
14th IEEE Computer Society International Conference 1977, pp. 320-324
- 188./ A. van Dam: "Introduction to Microprocessors - an Outline"
Course Notes, Computer Science 106,
Brown University, 1976
- 189./ A. van Dam, G.M. Stabler, R.J. Harrington: "Intelligent Satellites for Interactive Graphics"
IEEE Proc. Vol. 62, pp. 483-492
- 190./ A. van Dam: "Microprogramming for Computer Graphics"
Annual JEAS Conf. Pisa, Italy, Sept. 1971
- 191./ "Vector General 3400 Reference Manual"
Vector General Corporation, Woodland Hill
California, 1975
- 192./ C. Vickery: "Software Aids for Microprogram Development"
The Seventh Annual Workshop on Microprogramming
Palo Alto, 1973, pp. 208-211

- 193./ R. Vahlstrom, H. Malone: "Evolution of Microprogram used
Input/Output Processing in One Processor Family"
Computer Design, January 1976, pp. 98-109
- 194./ H.H. Webber: "The Super Integral Microprogrammed
Arithmetic Logic Expeditor /SIMALE/"
SIGMICRO, Vol. 3, No. 4, 1973
- 195./ "MCP 1600 Microprocessor Product Description"
Western Digital Corporation, 1975
- 196./ D.C. Wyland: "Design your own minicomputer ..."
Electronic Design, Sept. 27, 1975
- 197./ D.C. Wyland: "Increase microcomputer efficiency ..."
Electronic Design, Nov. 8, 1975
- 198./ M.V. Wilkes: "The use of writable control memory in a
multiprogramming environment"
5th Annual Workshop on Microprogramming, 1972
pp. 62-65
- 199./ J.E. Wirshing: "Computer of the 1980's - Is It a Network
of Microcomputers?"
Digest of Papers, COMPCOM Fall, 1975, pp. 23-26
- 200./ W. Wulf, R. Levin: "A Local Network"
Datamation, Feb., 1975, pp. 47-50
- 201./ A.J. Weissberger: "Analysis of Multiple - Microprocessor
System Architectures"
Computer Design, June 1977, pp. 151-163
- 202./ W.R. Wittmayer: "Array Processor Provides High Throughput
Rates"
Computer Design, March 1978, pp. 93-100

- 203./ S. Waser: "State-of-the-Art in High Speed Arithmetic Integrated Circuits"
Computer Design, July 1978, pp. 67-75
- 204./ J. Wirsching: "Micro Networks"
Digital Design, February 1976
- 205./ L.C. Widdoes, Jr.: "Architectural considerations for general purpose multiprocessors"
13th IEEE Computer Society International Conference, 1976, pp. 251-254
- 206./ S. Zucker, R.L. Davis, C.M. Campbell: "A building block approach to multiprocessing"
Spring Joint Computer Conference, 1972,
pp. 685-703

X. FÜGGELÉK /ábrák gyűjteménye/

1. ábra: A grafikus displayek felosztása a megjelenítés módja szerint
2. ábra: Host számítógéphez közvetlenül csatolt grafikus display
3. ábra: Host számítógéphez közvetlenül csatolt grafikus display lokális memóriával
4. ábra: Hosthoz mini számítógép segítségével kapcsolódó grafikus display
5. ábra: Hosthoz miniszámítógép segítségével kapcsolódó grafikus display, display processzorral.
6. ábra: Hosthoz miniszámítógép segítségével kapcsolódó grafikus display, bufferelt display processzorral
7. ábra: Elosztott intelligenciájú grafikus display
8. ábra: GD'71-TPA'70 grafikus display rendszer
9. ábra: TPA'70 Bus Switch
10. ábra: CAD/CAM duál processzoros konfiguráció
11. ábra: Digital Scientific META4-Floating Point Processor

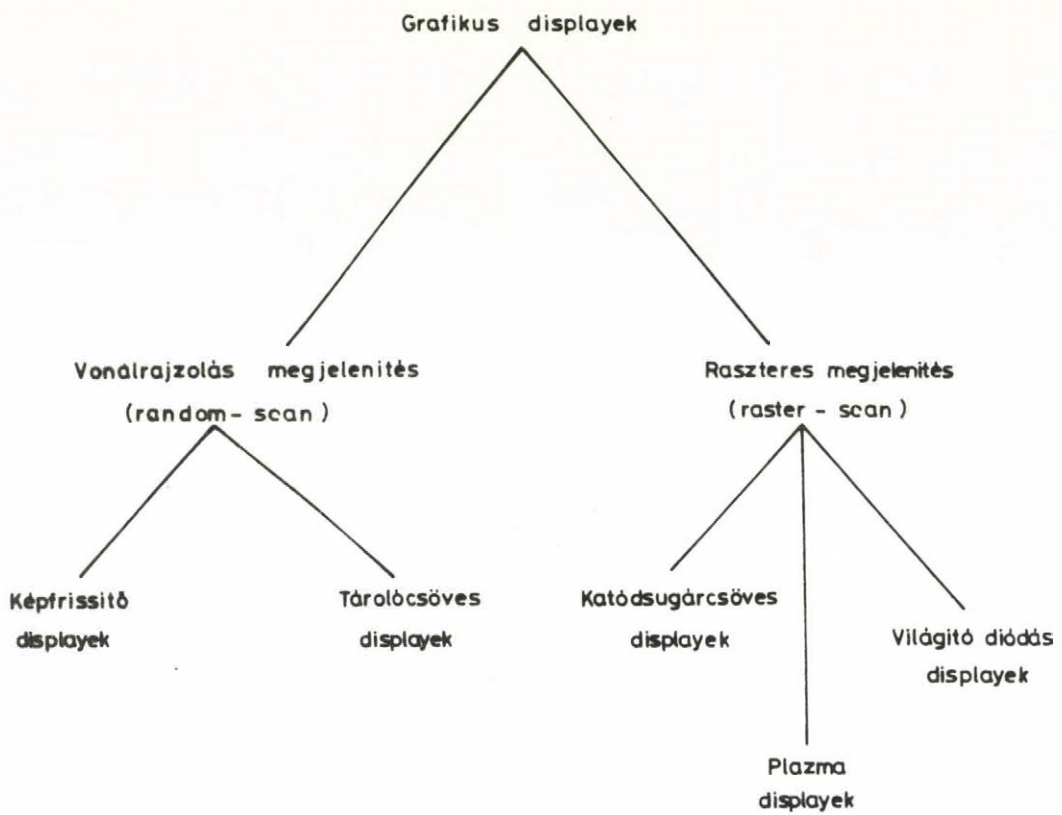
12. ábra: TPA'70-FPU
13. ábra: TPA'70 MIPROBUS WINDOW
14. ábra: A GD'80 általános felépítése
15. ábra: Vektoros és raszteres letapogatású karakter-
generátor
16. ábra: A standard 8 bites mikroprocesszorok blokkvázlata
17. ábra: Standard 8 bites mikroprocesszorok memória tér-
képe
18. ábra: Mikroprogramozott processzorok
19. ábra: Mikroprogramozott vezérlés
- 19a. ábra: Microprogram sequencer
20. ábra: Transzformációs processzor
21. ábra: Matrix Multiplier
22. ábra: GD'80 Basic Terminal
23. ábra: Graphic Computer
24. ábra: GD'80 IT
25. ábra: GD'80 AGS, SGS
26. ábra: GD'80 KC alfanumerikus display koncentrátor

27. ábra: GD'80 TS

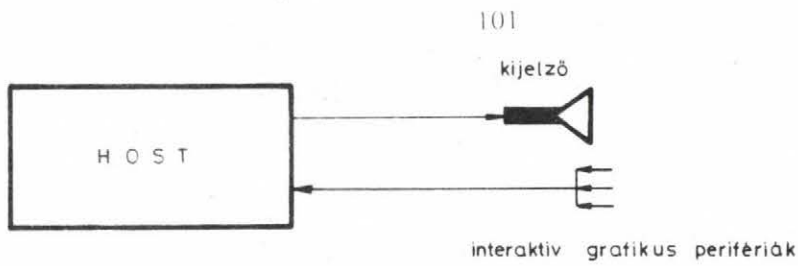
28. ábra: GD'80 DP

29. ábra: Oktatóberendezés légiforgalomirányítók kiképzéséhez

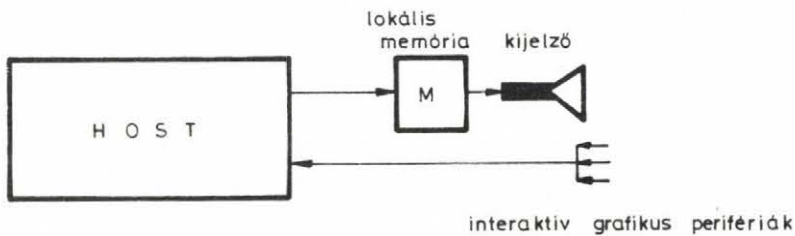




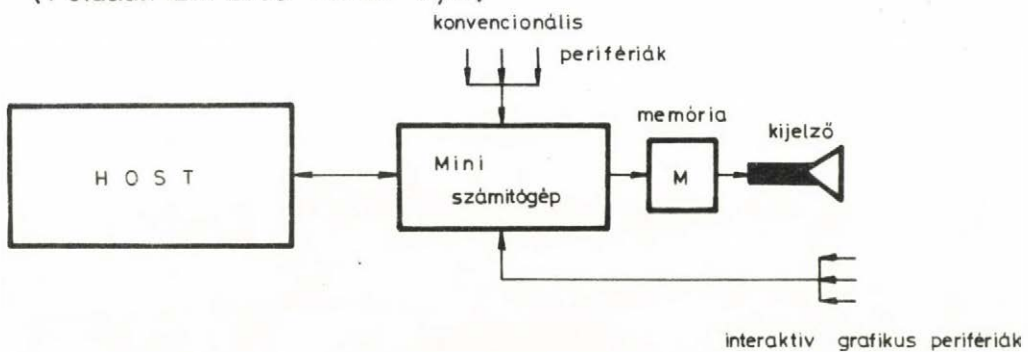
1. ábra: A grafikus displayek felosztása a megjelenítés módja szerint



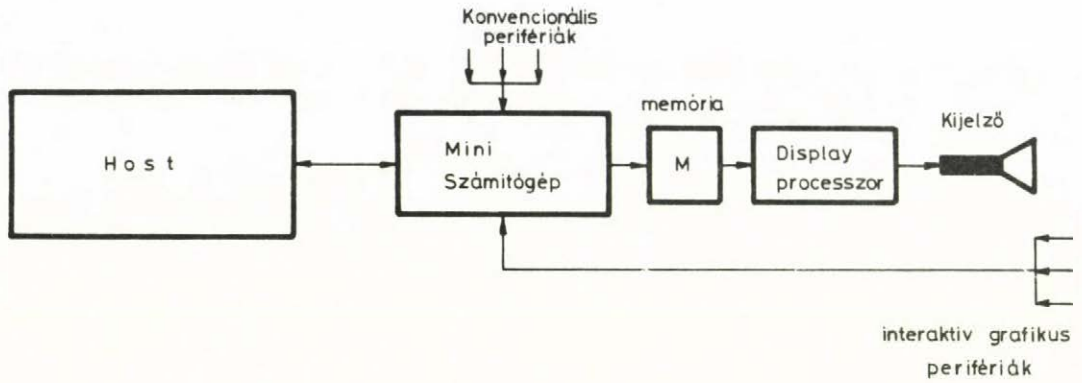
2. ábra : Host számítógéphez közvetlenül
csatolt grafikus display
(Például: TX2 - Sketchpad rendszer)



3. ábra: Host számítógéphez közvetlenül
csatolt grafikus display lokális memóriával
(Például: IBM 2250 Model 2,3)



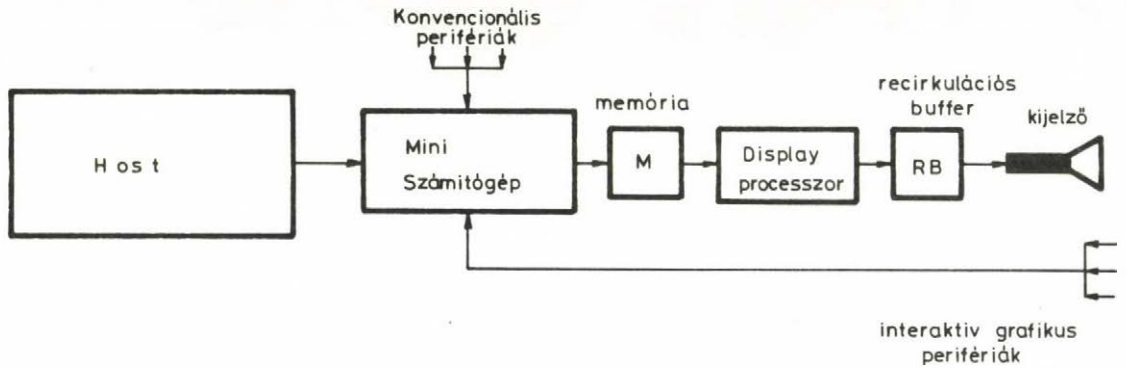
4. ábra : Hosthoz mini számítógép segítségével
kapcsolódó grafikus display
(Például: IBM 2250/ 1130 , Vector General 3-PDP11, stb)



5. ábra: Hozhoz miniszámítógép segítségével

kapcsolódó grafikus display, display processzorral

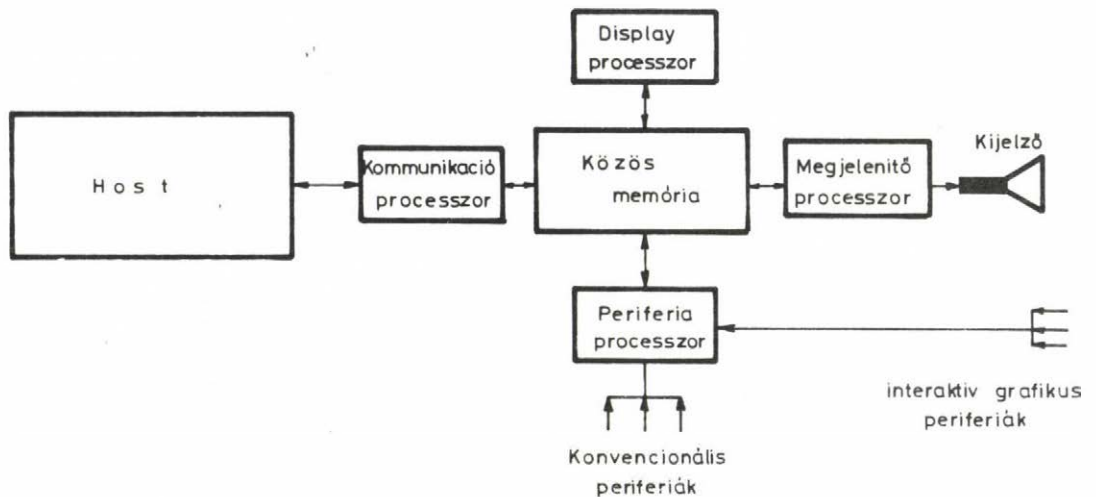
(Például: Brown University Graphic System,
Evans and Sutherland Line Drawing System 1,2)



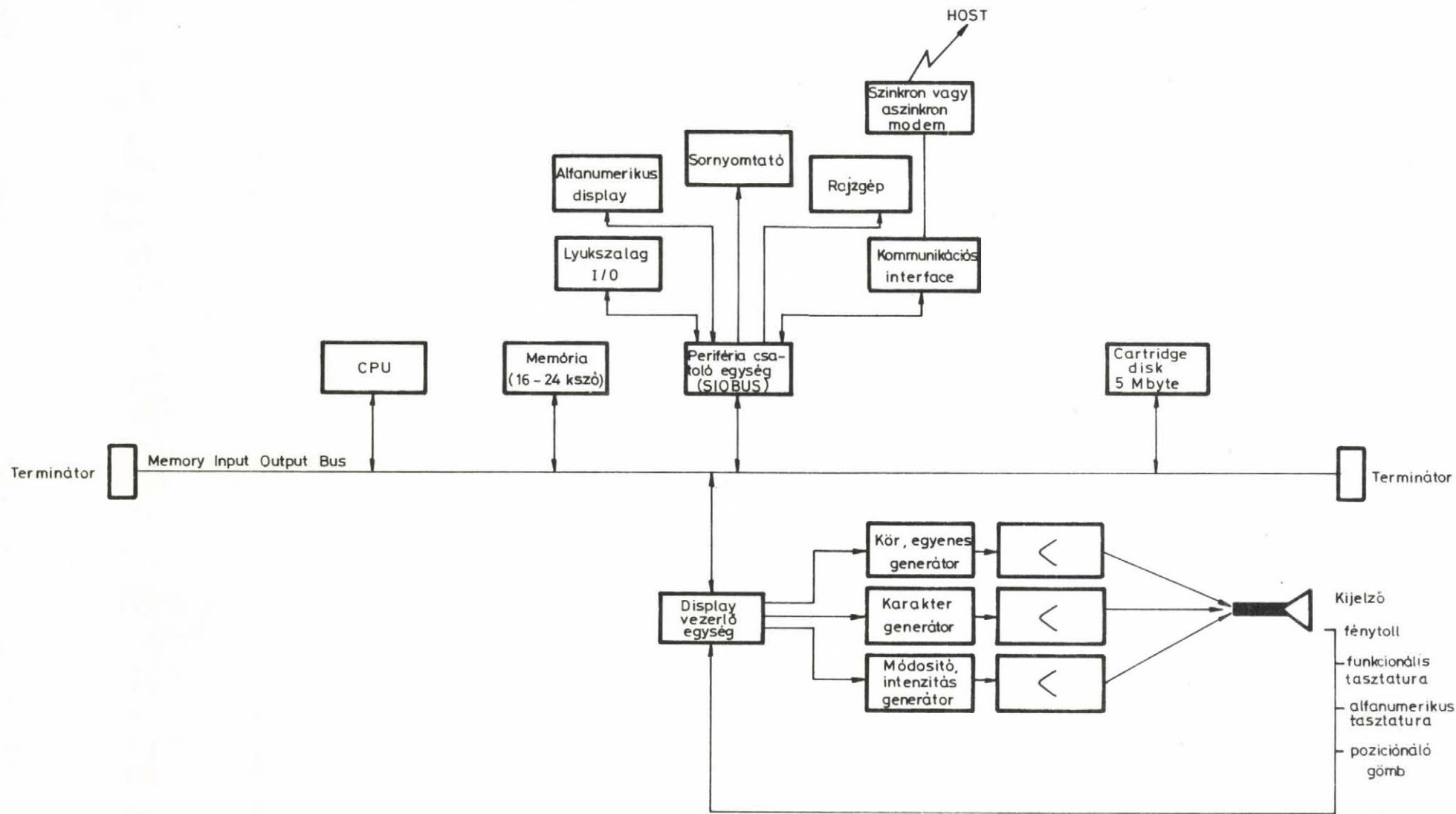
6. ábra: Hozhoz miniszámítógép segítségével

kapcsolódó grafikus display, bufferelt display processzorral

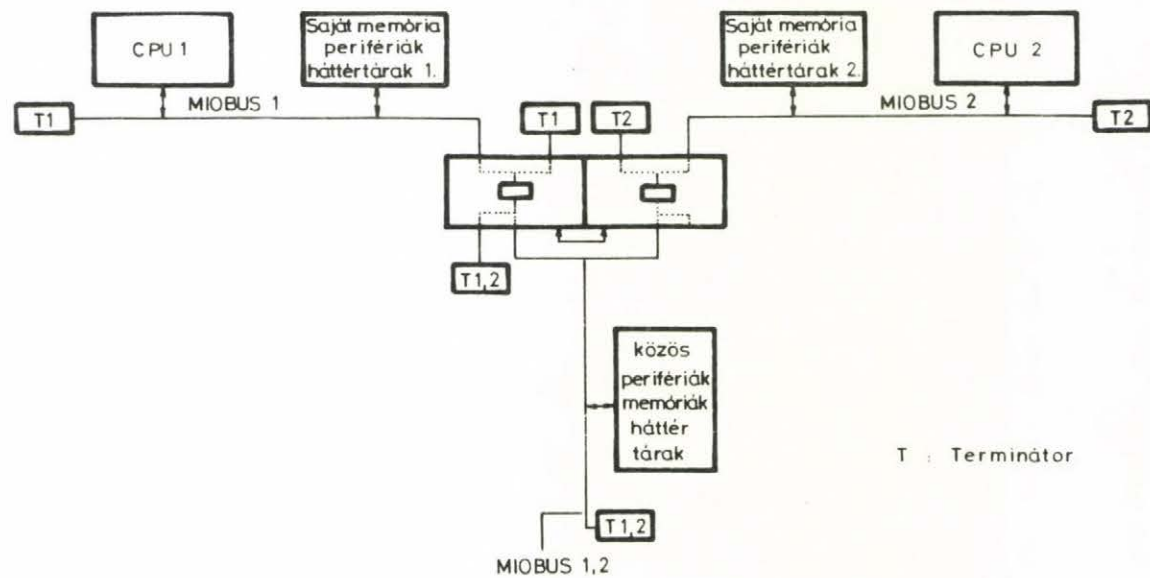
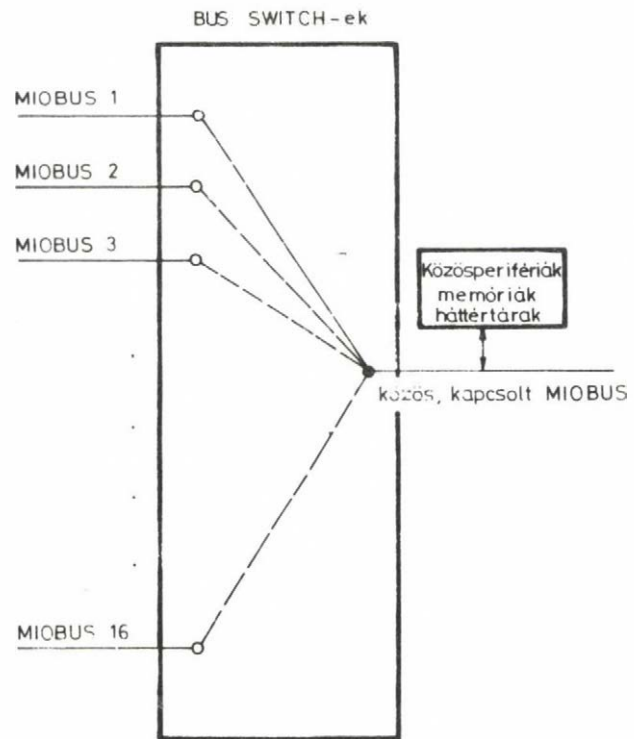
(Például: Vector General 3400 - PDP 11)



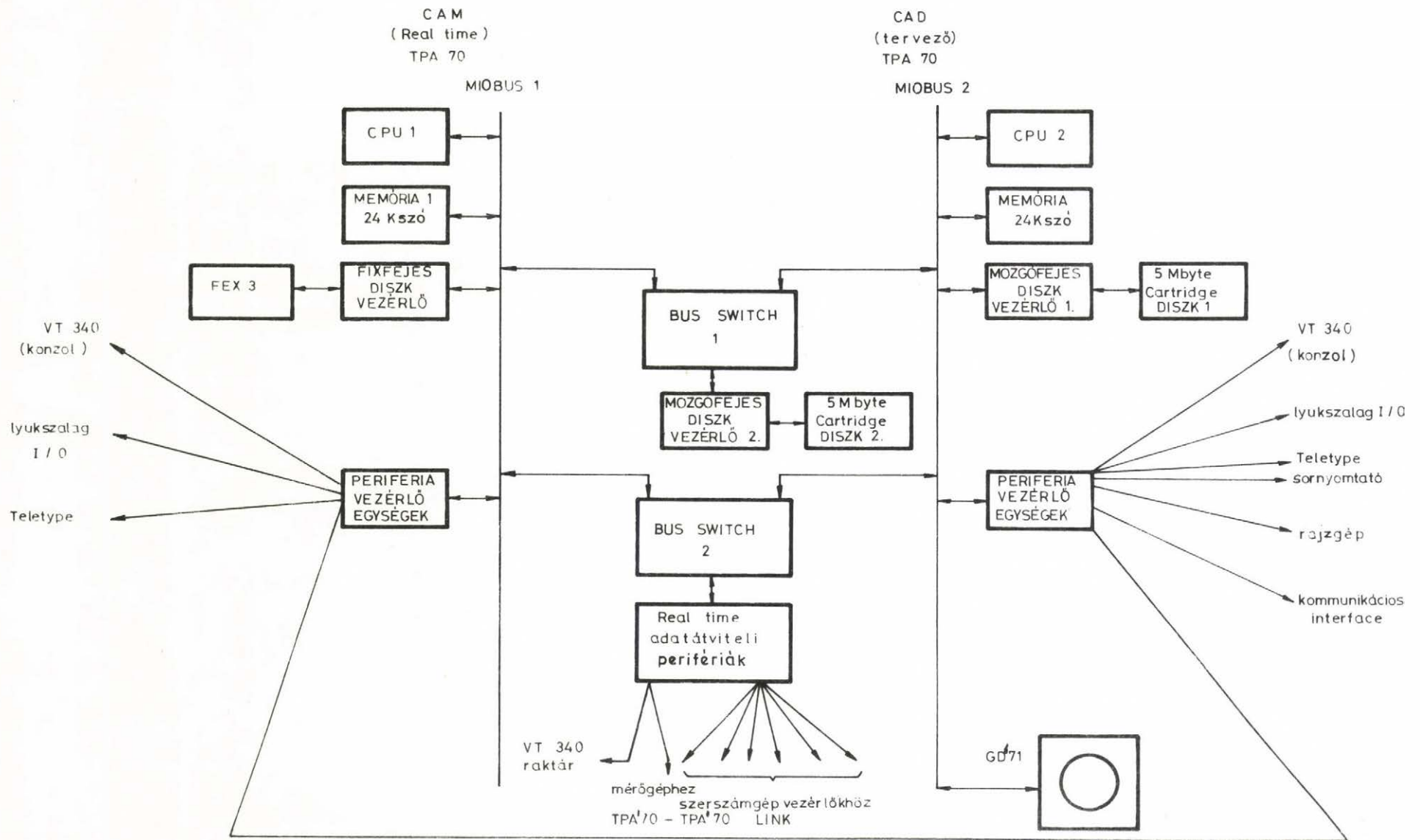
7. ábra: Elosztott intelligenciájú grafikus display
(GD⁸⁰)



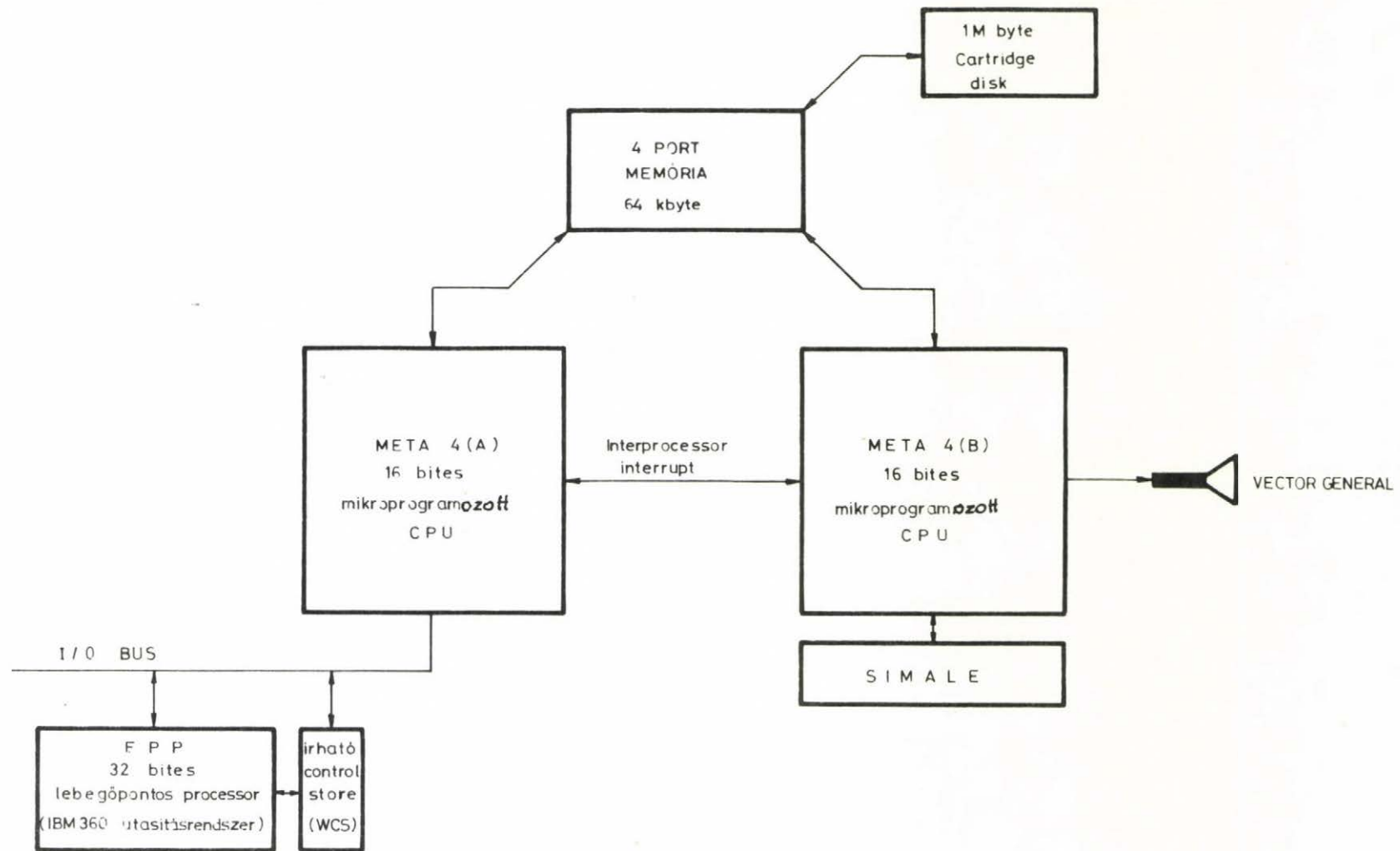
8. ábra: GD^a71 - TPA^a70 grafikus display rendszer



9. ábra: TPA'70 Bus Switch



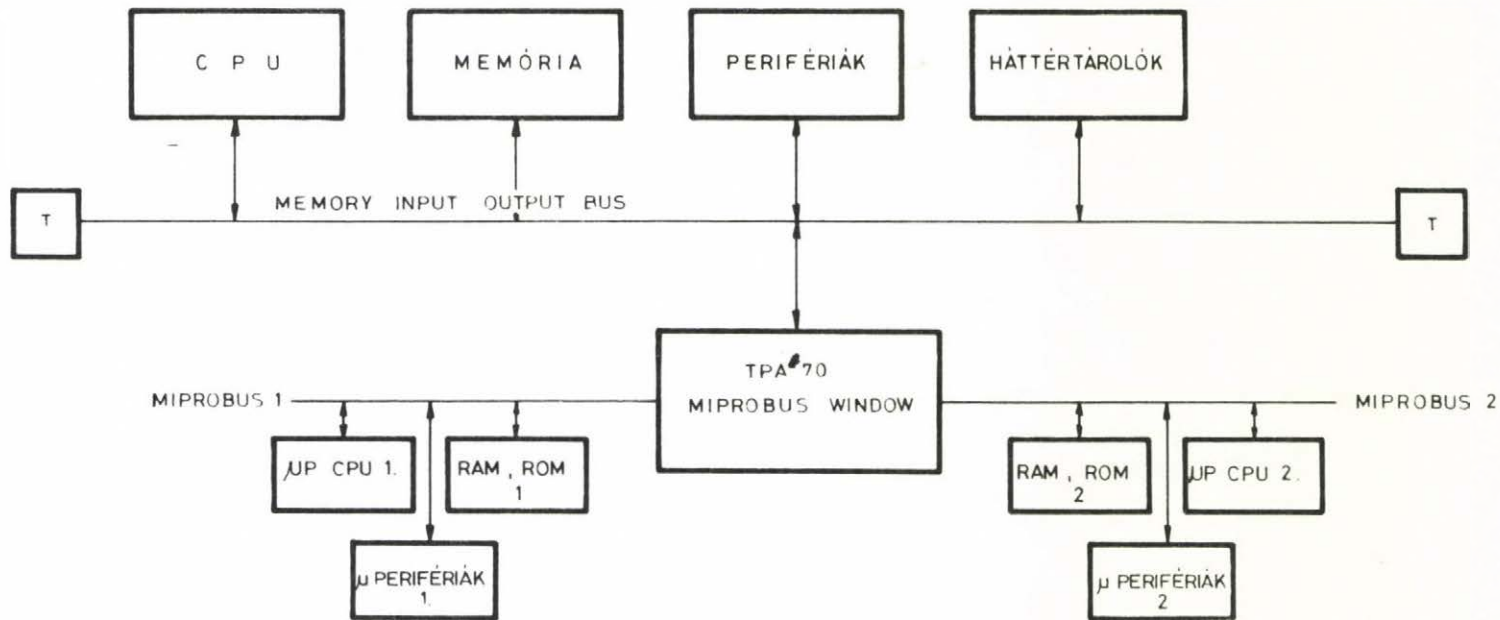
10. ábra: CAD/CAM dual processzoros konfiguráció



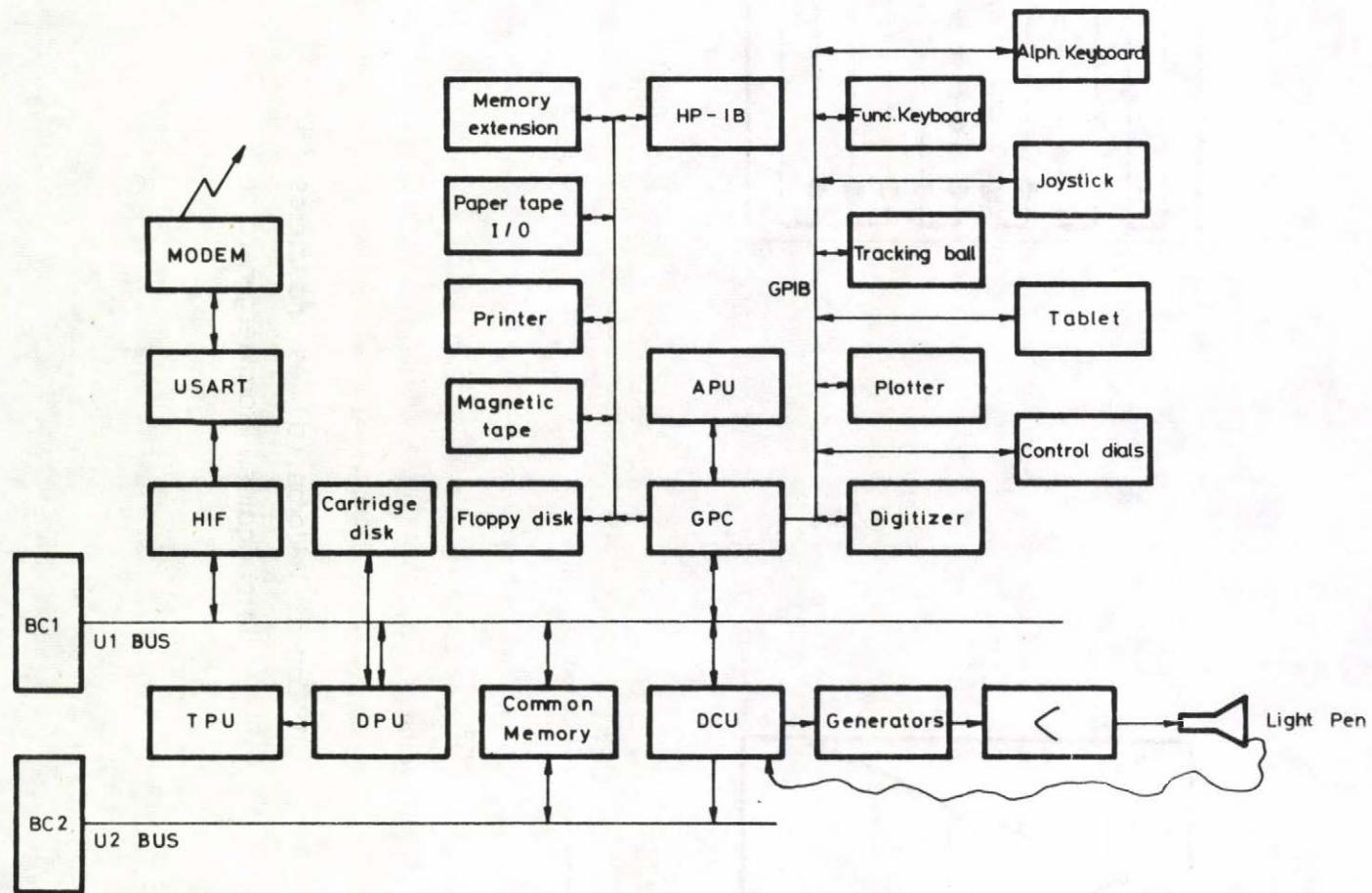
11. ábra: Digital Scientific META 4 - Floating Point Processor
(Brown University Graphic System)



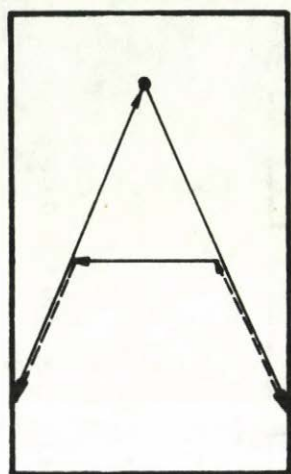
12. ábra: TPA'70 - FPU (lebegőpontos processor)



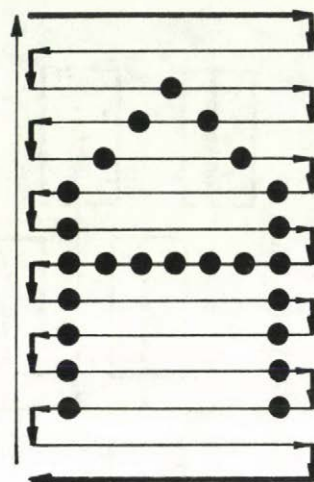
13. ábra : TPA'70 MICROBUS WINDOW



14. ábra: A GD'80 általános felépítése



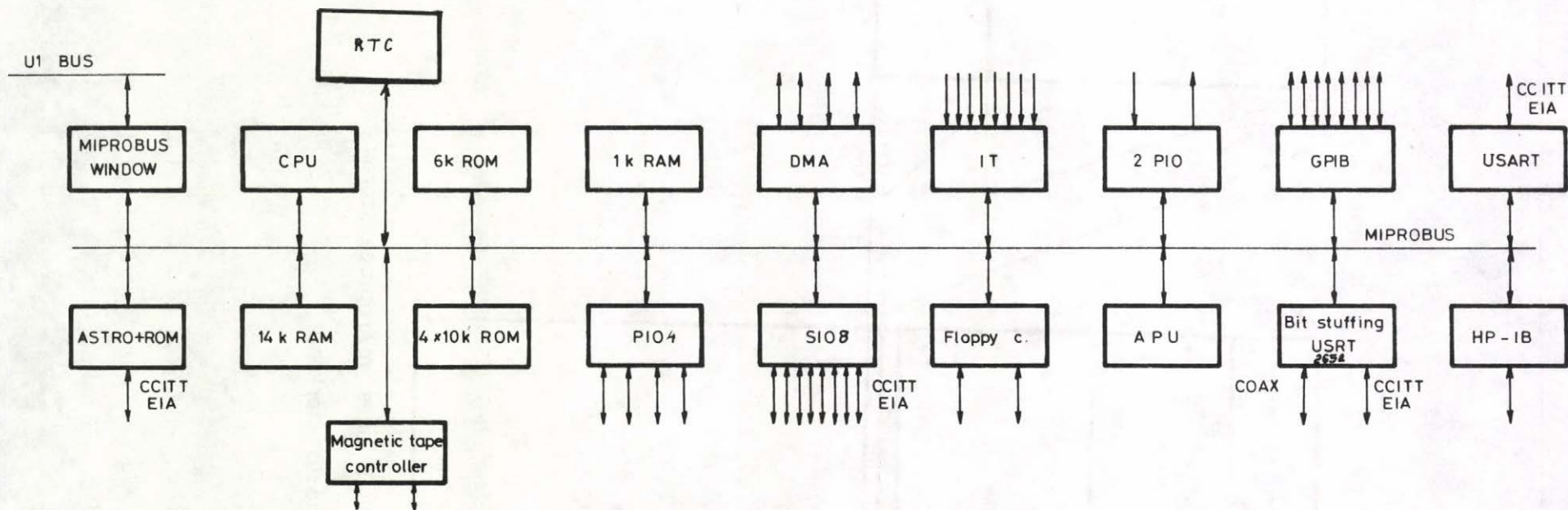
(a)



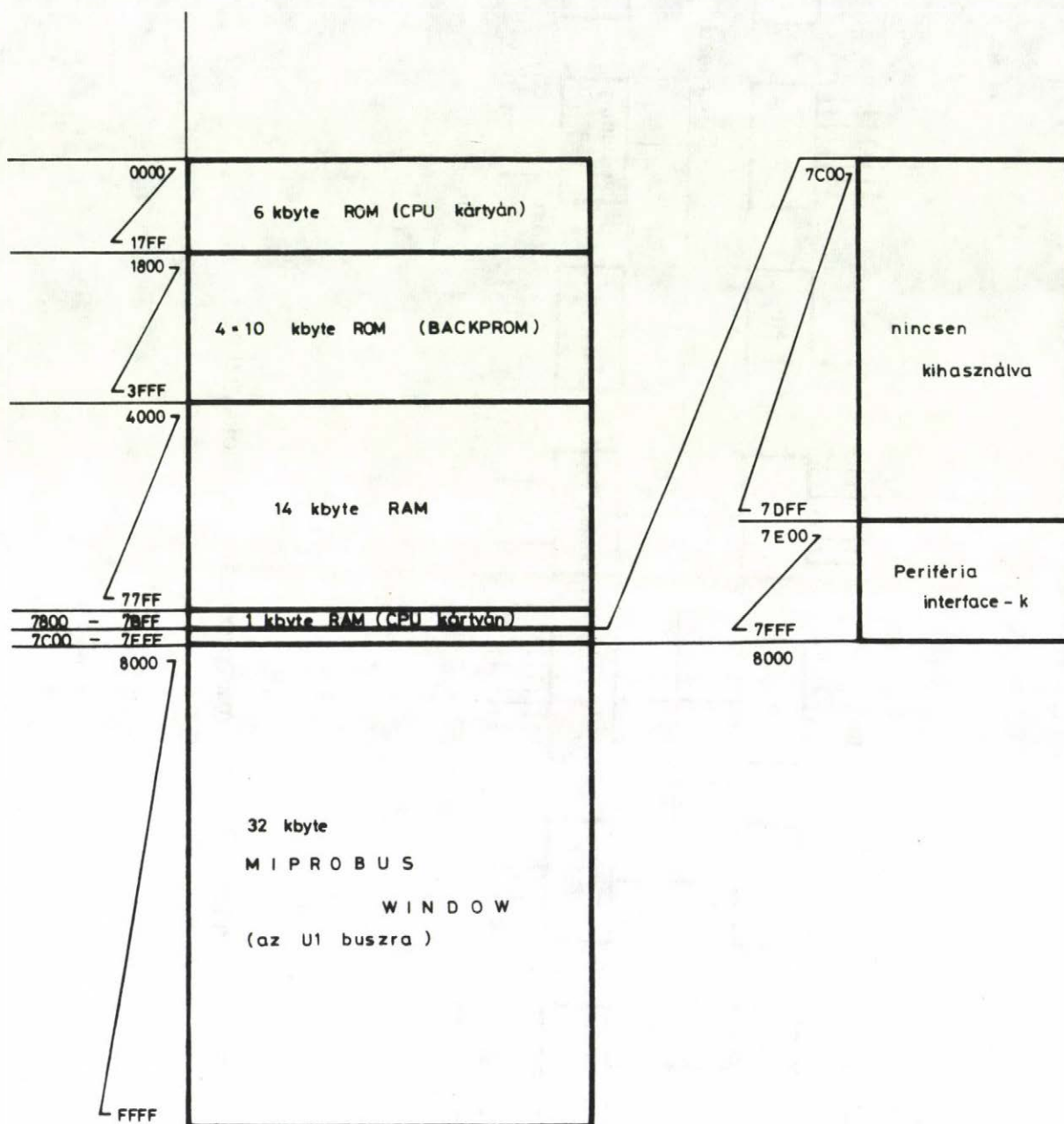
(b)

—————→ kivilágított
- - - - -→ kioltott
sugárral
rajzolt
vektor

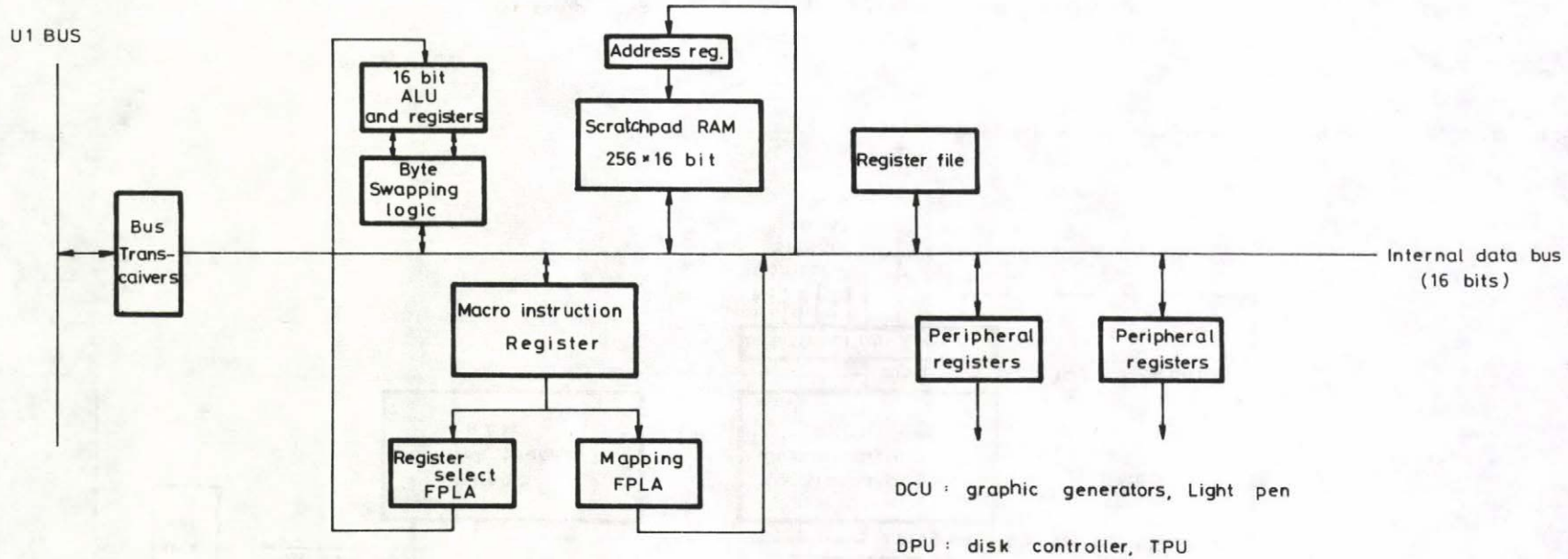
15. ábra: Vektoros (a) és raszteres (b)
letapogatású karaktergenerátor



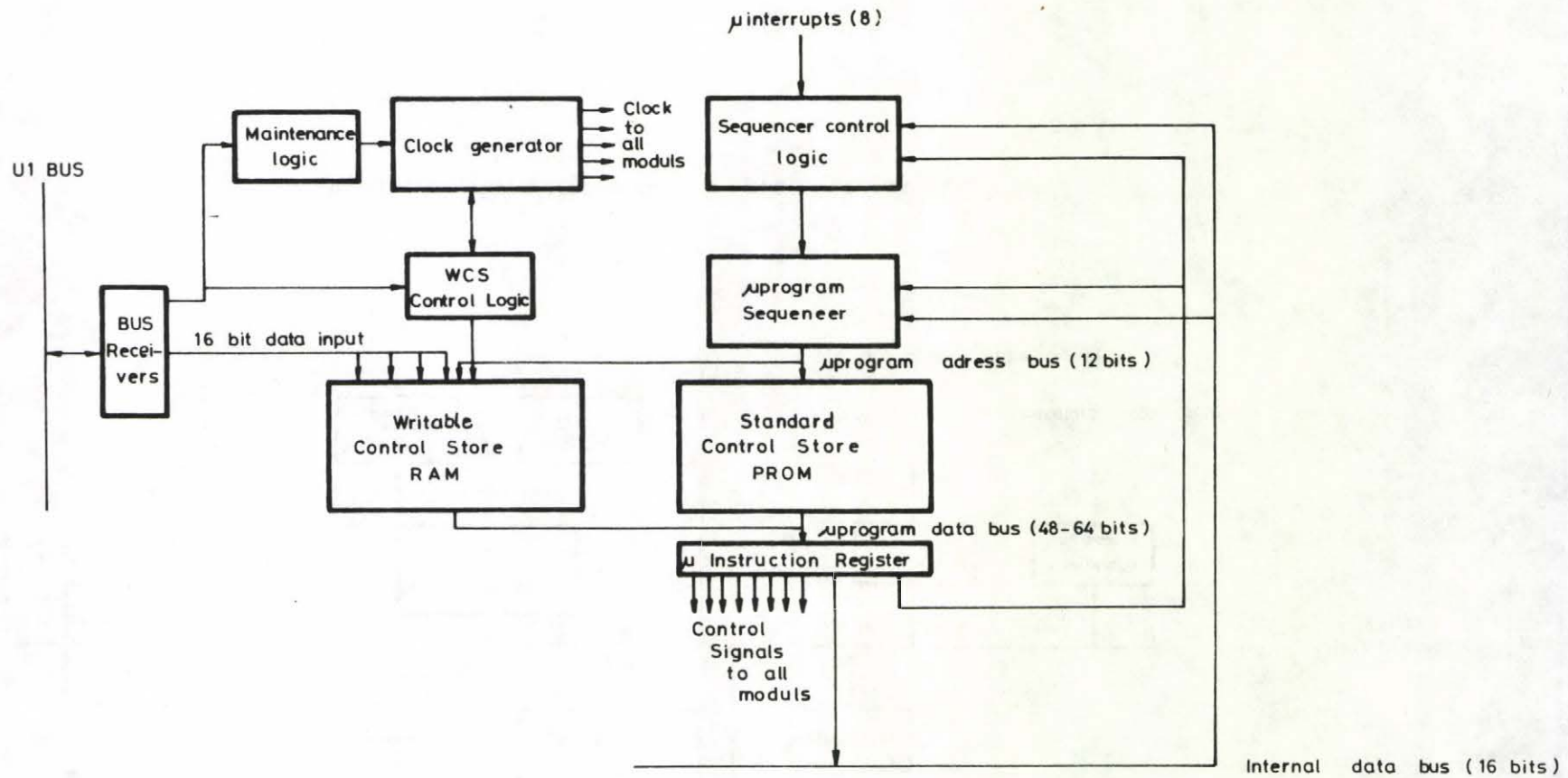
16. ábra.: A standard 8 bites mikroprocesszorok blokkvázlata



17. ábra : Standard 8 bites mikroprocesszorok
memória térképe

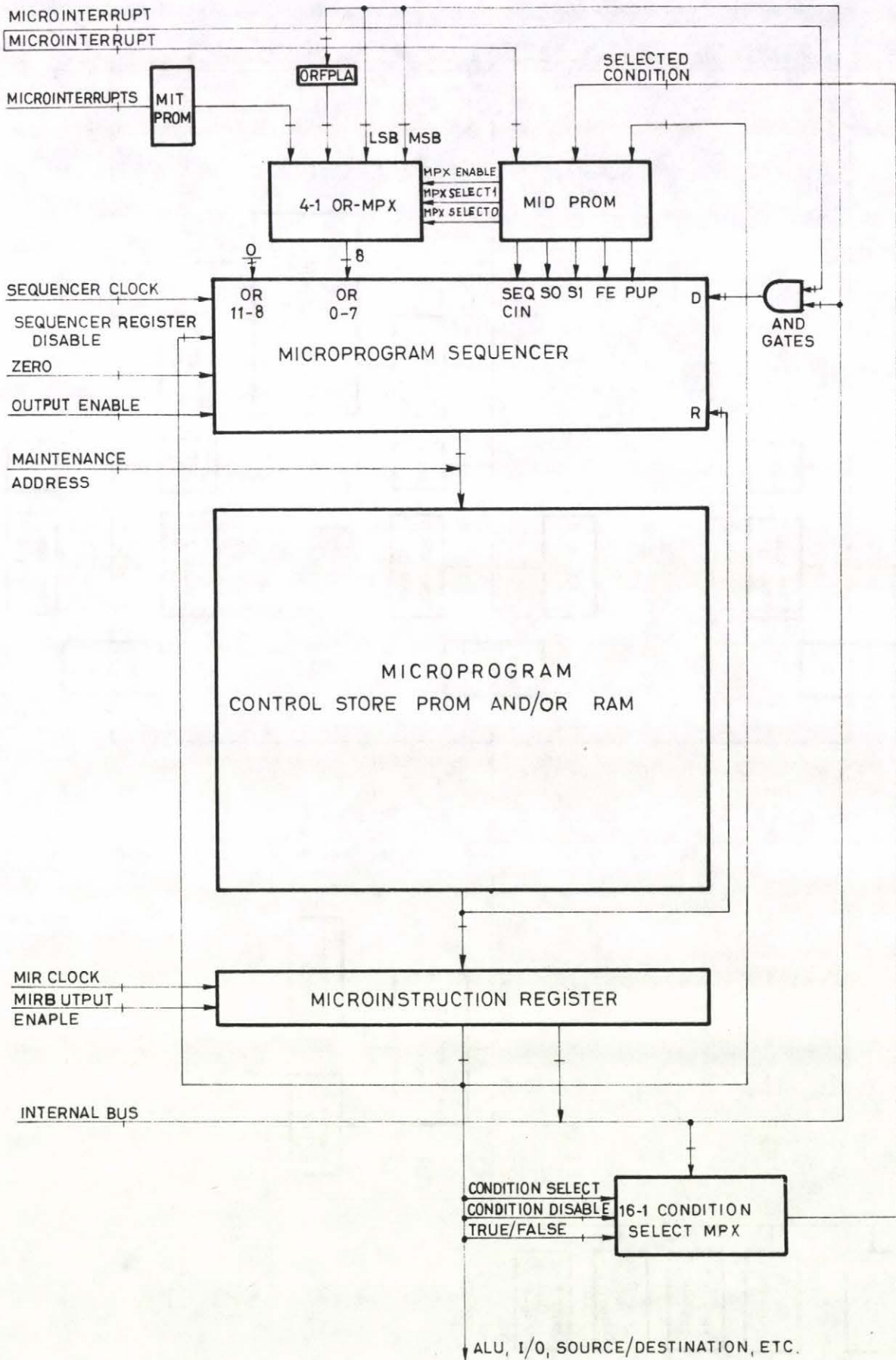


18. ábra: Mikroprogramozott processzorok (DCU , DPU)

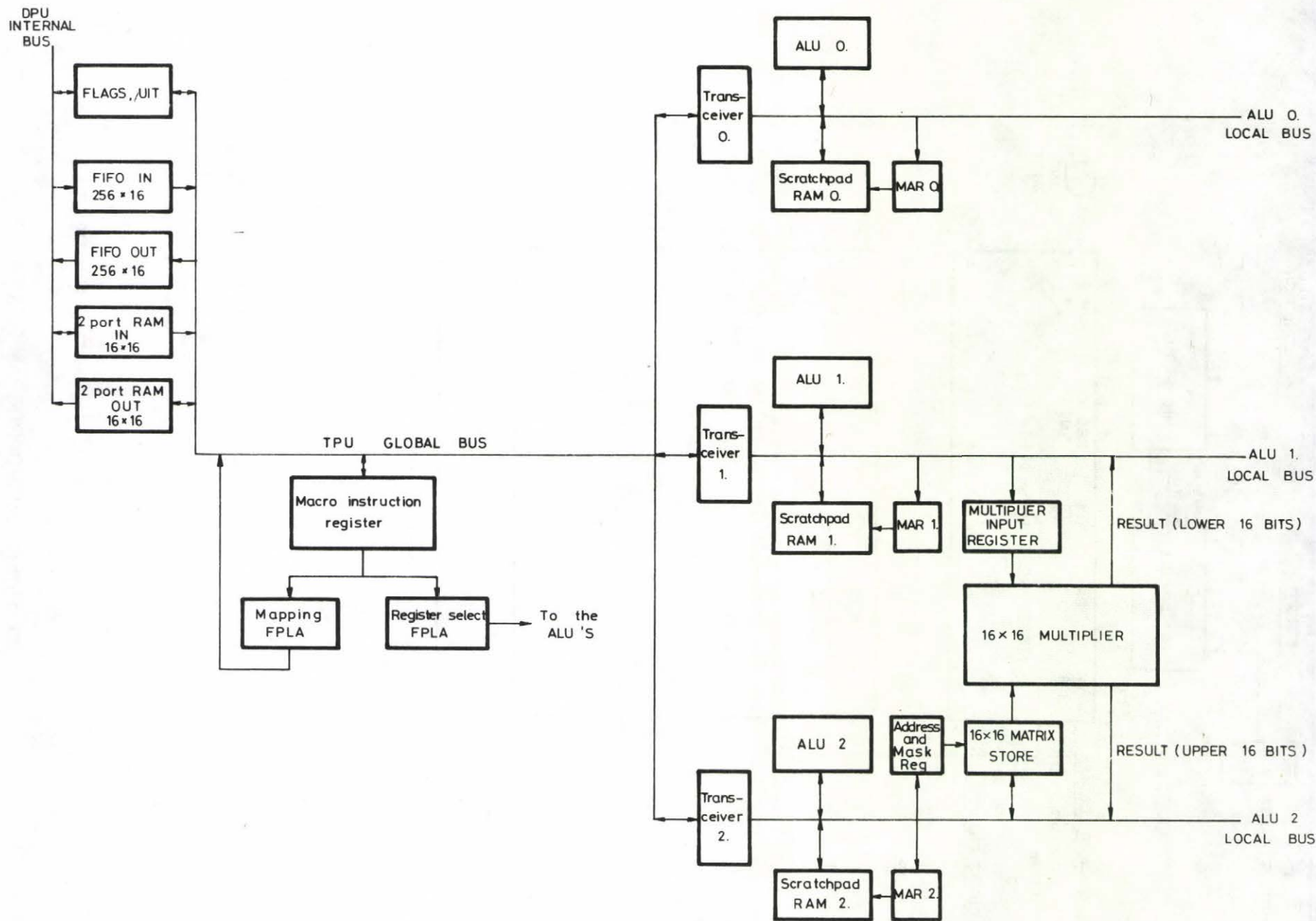


114

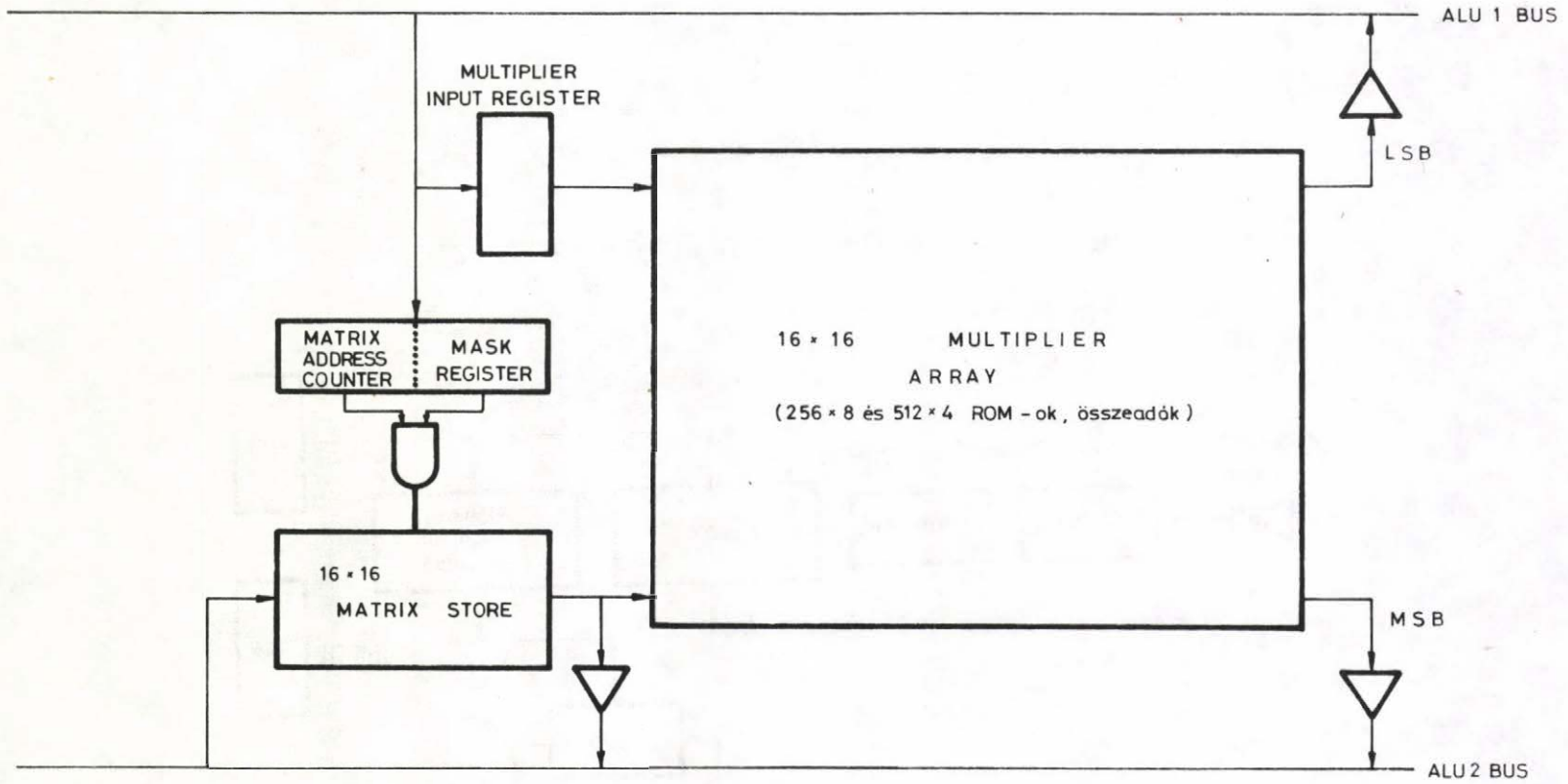
19. ábra: Mikroprogramozott vezérlés



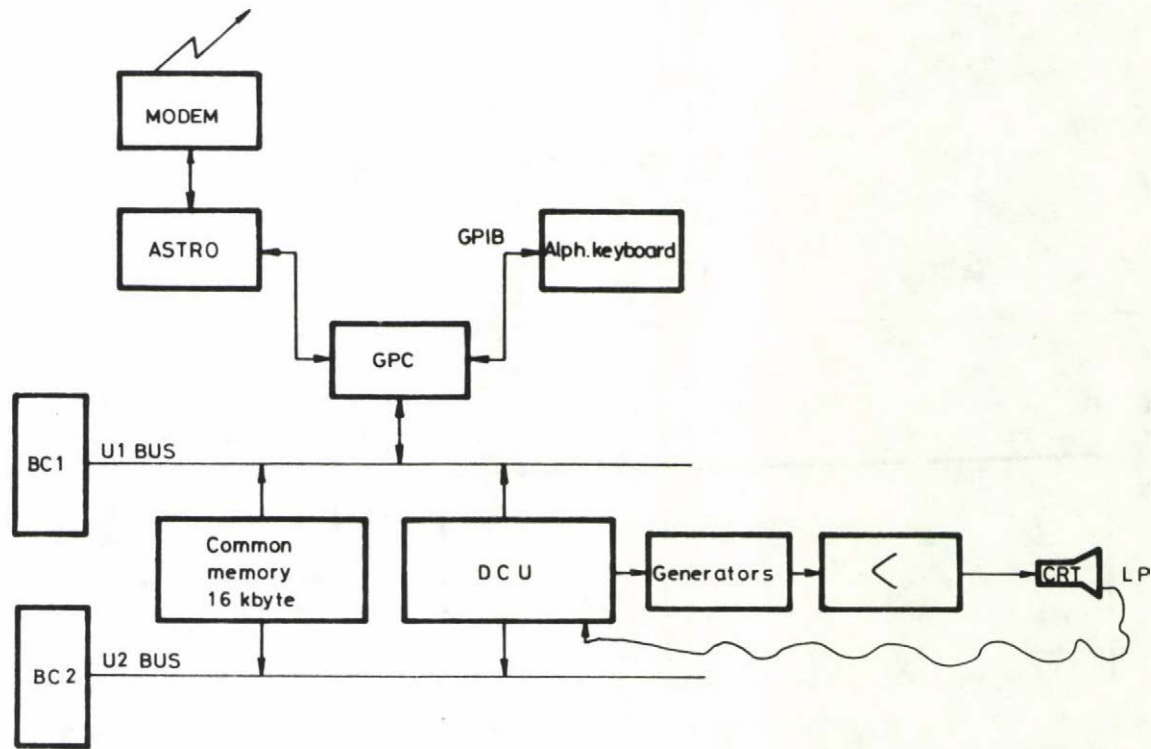
19. a. ábra. MICROPROGRAM SEQUENCER



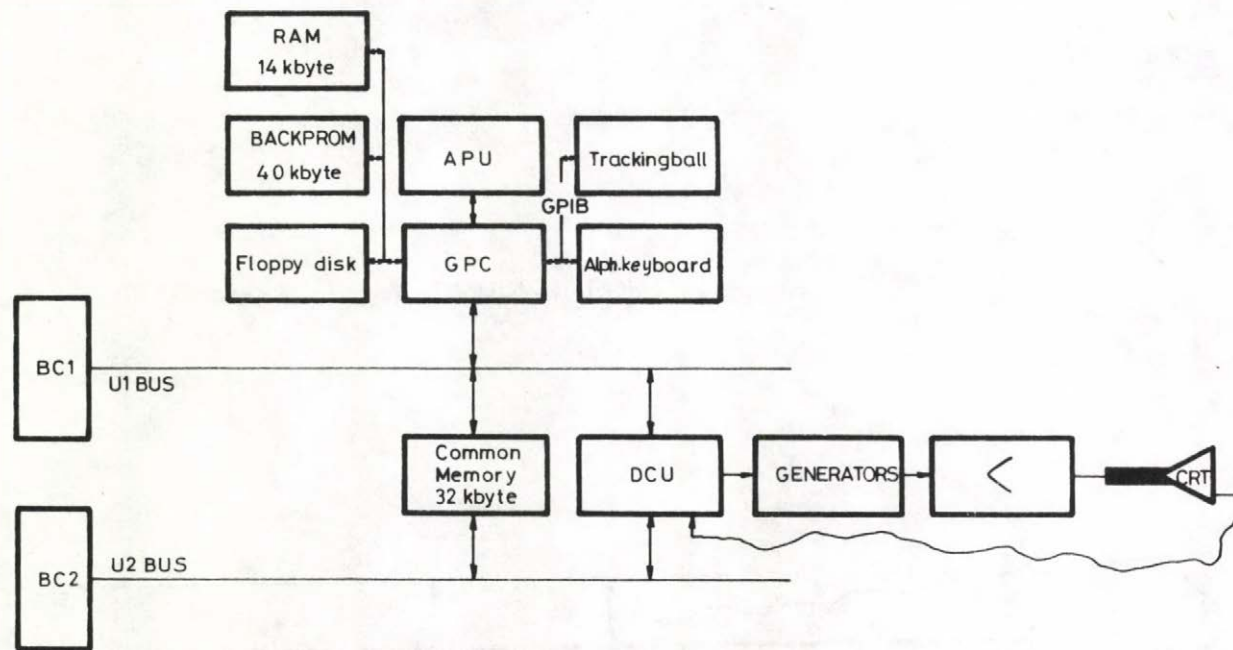
20. ábra.: Transzformációs processzor



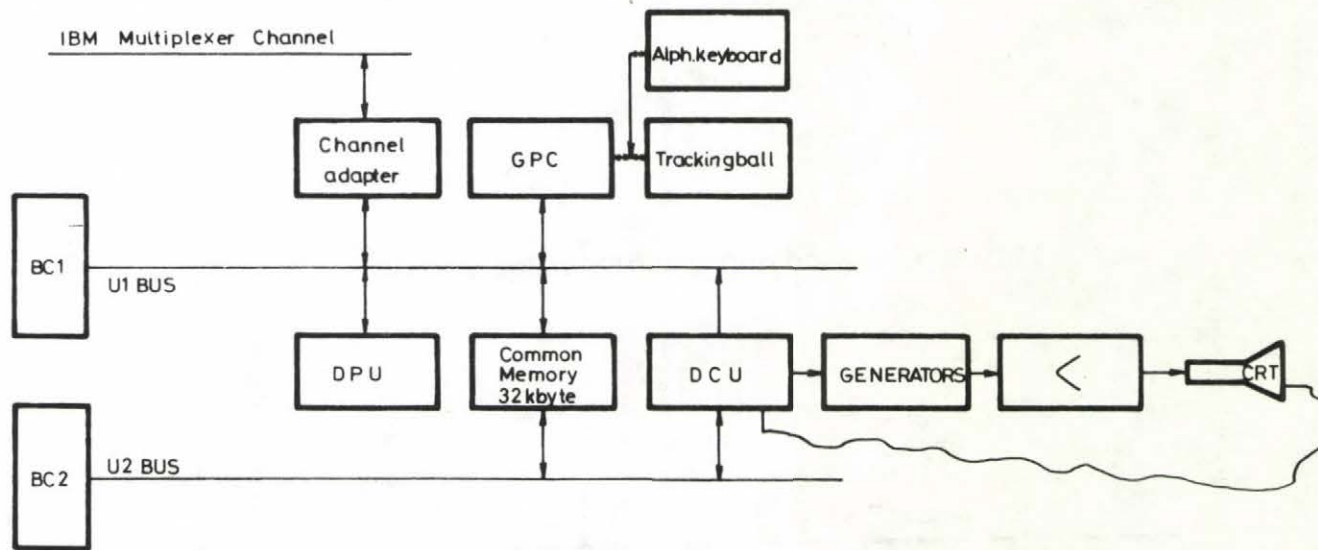
21. ábra: Matrix Multiplier



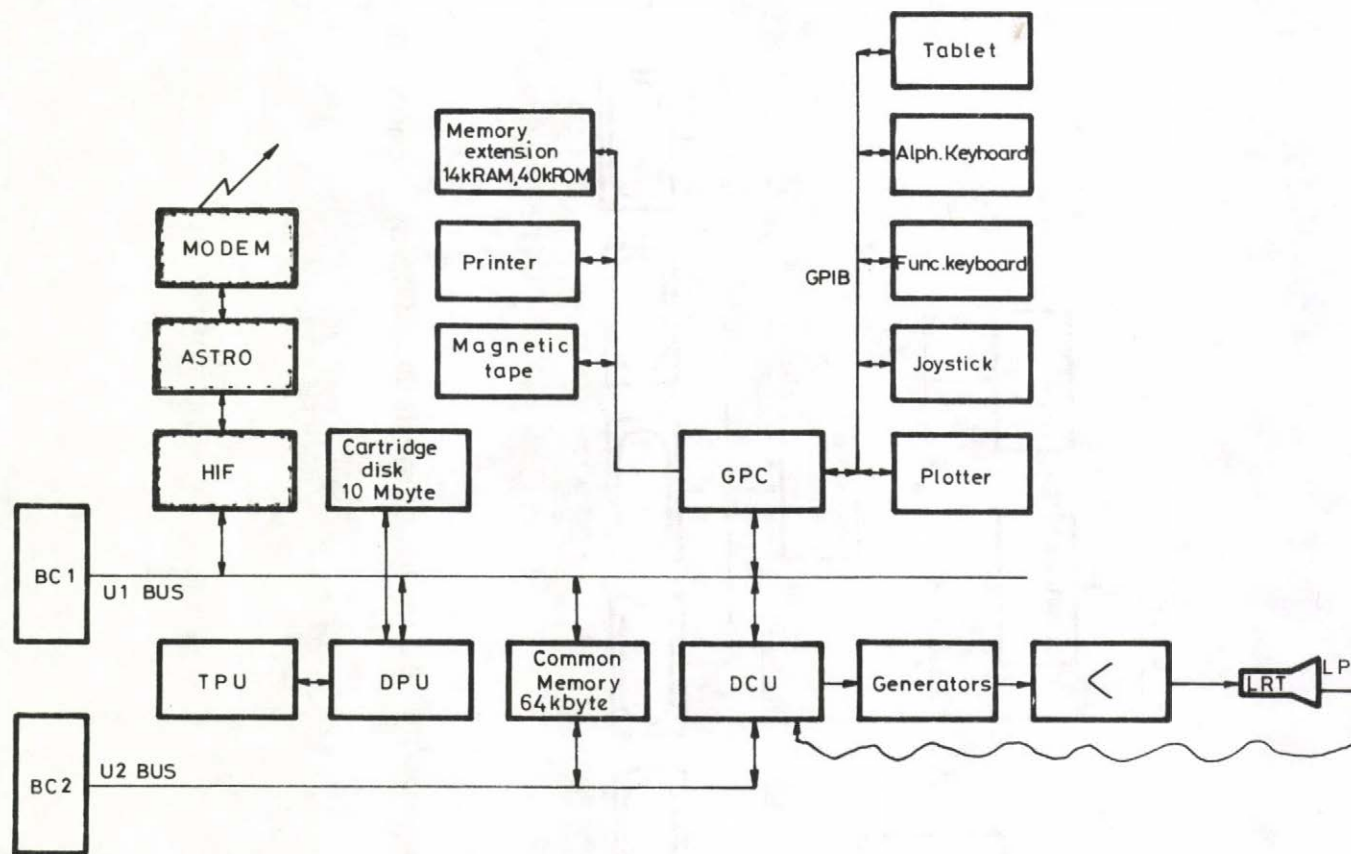
22. ábra : GD'80 Basic Terminal (GD 80 BT)



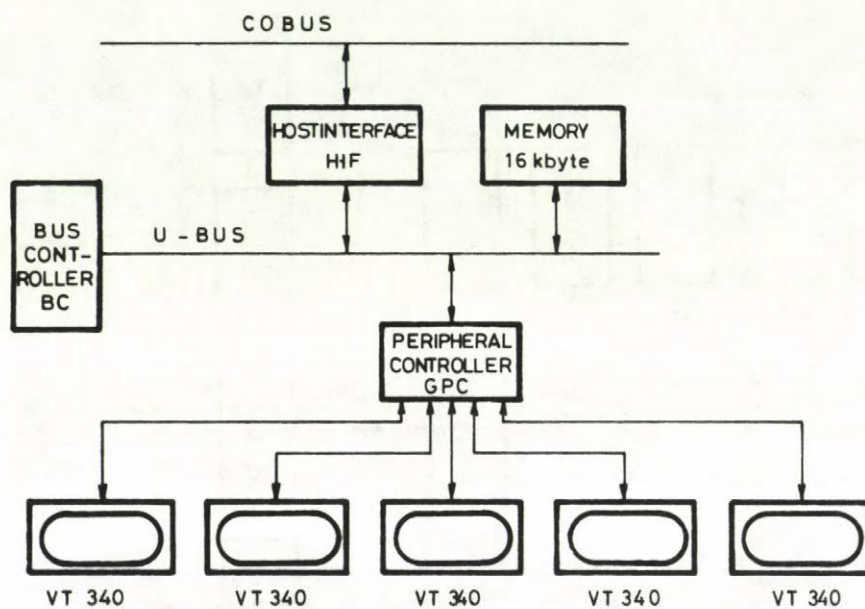
23. ábra.: Graphic Computer (GD80GC)



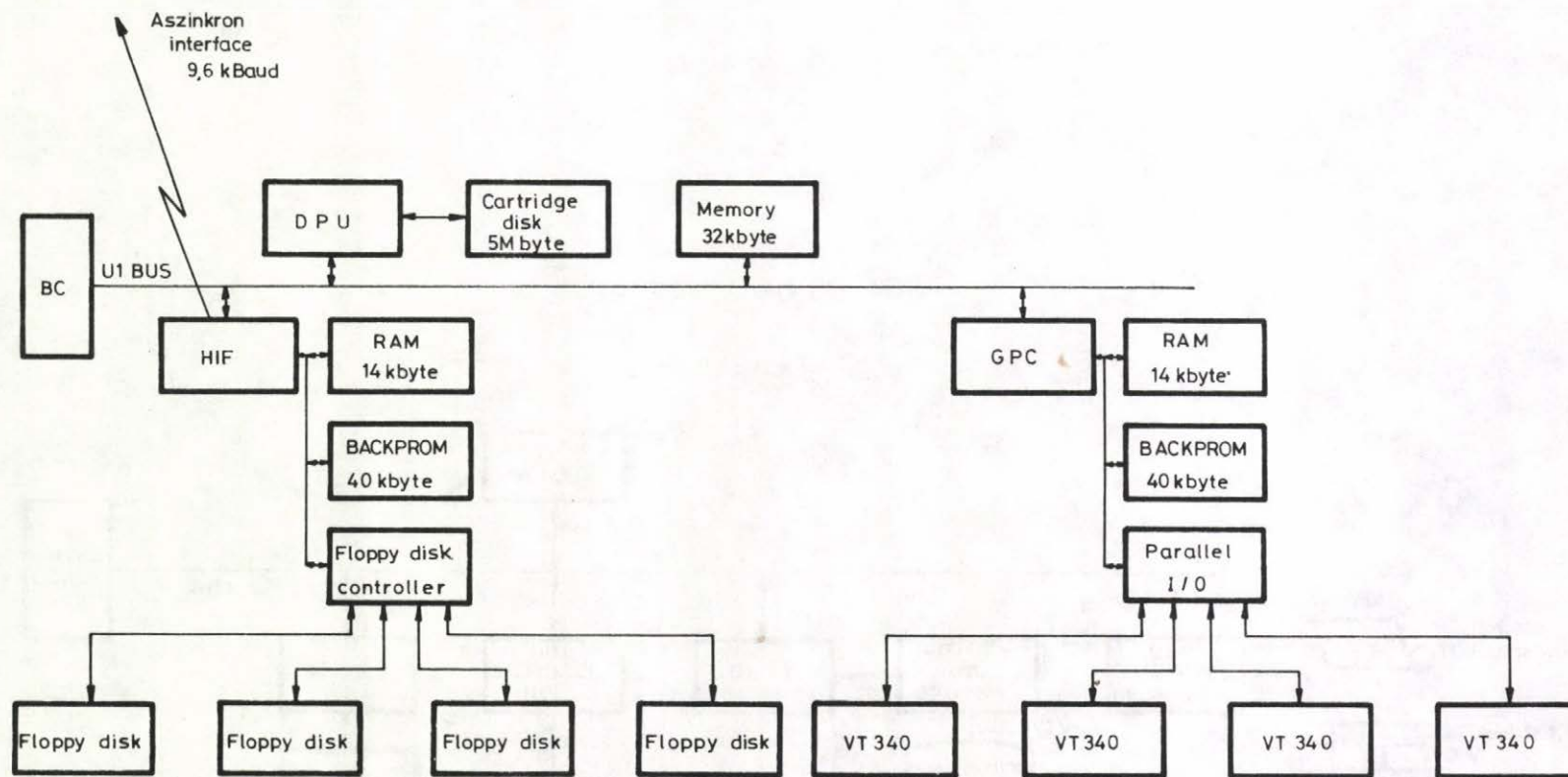
24. ábra.: GD80 IT



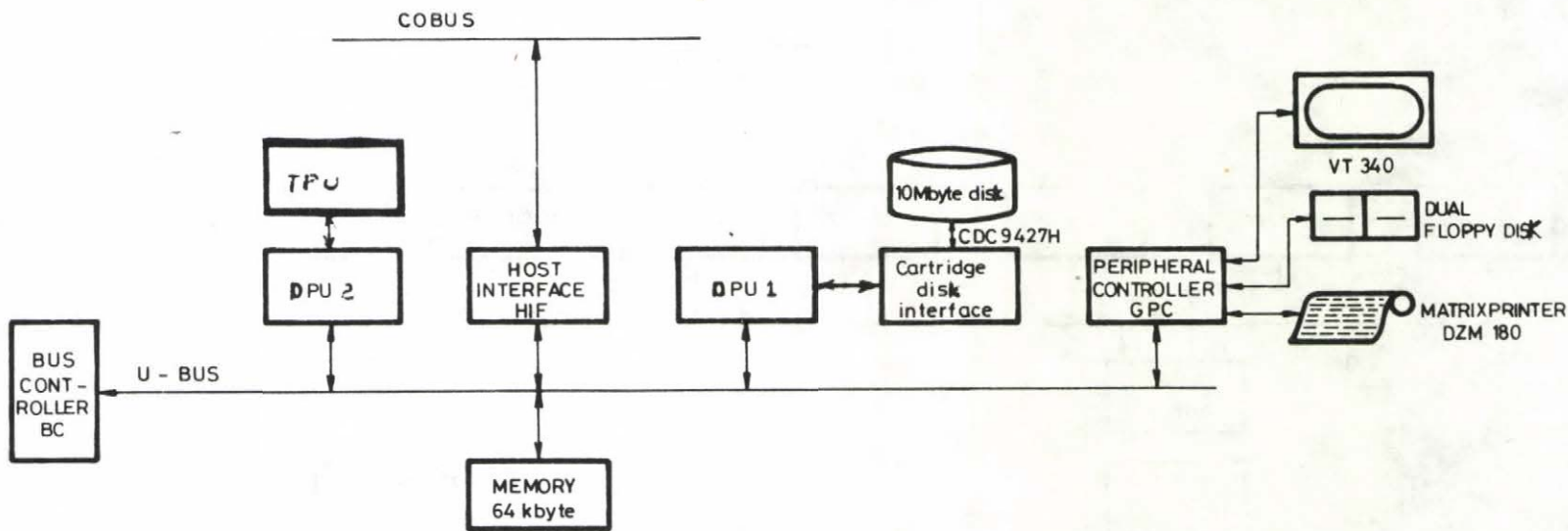
25. ábra: GD'80 AGS . SGS



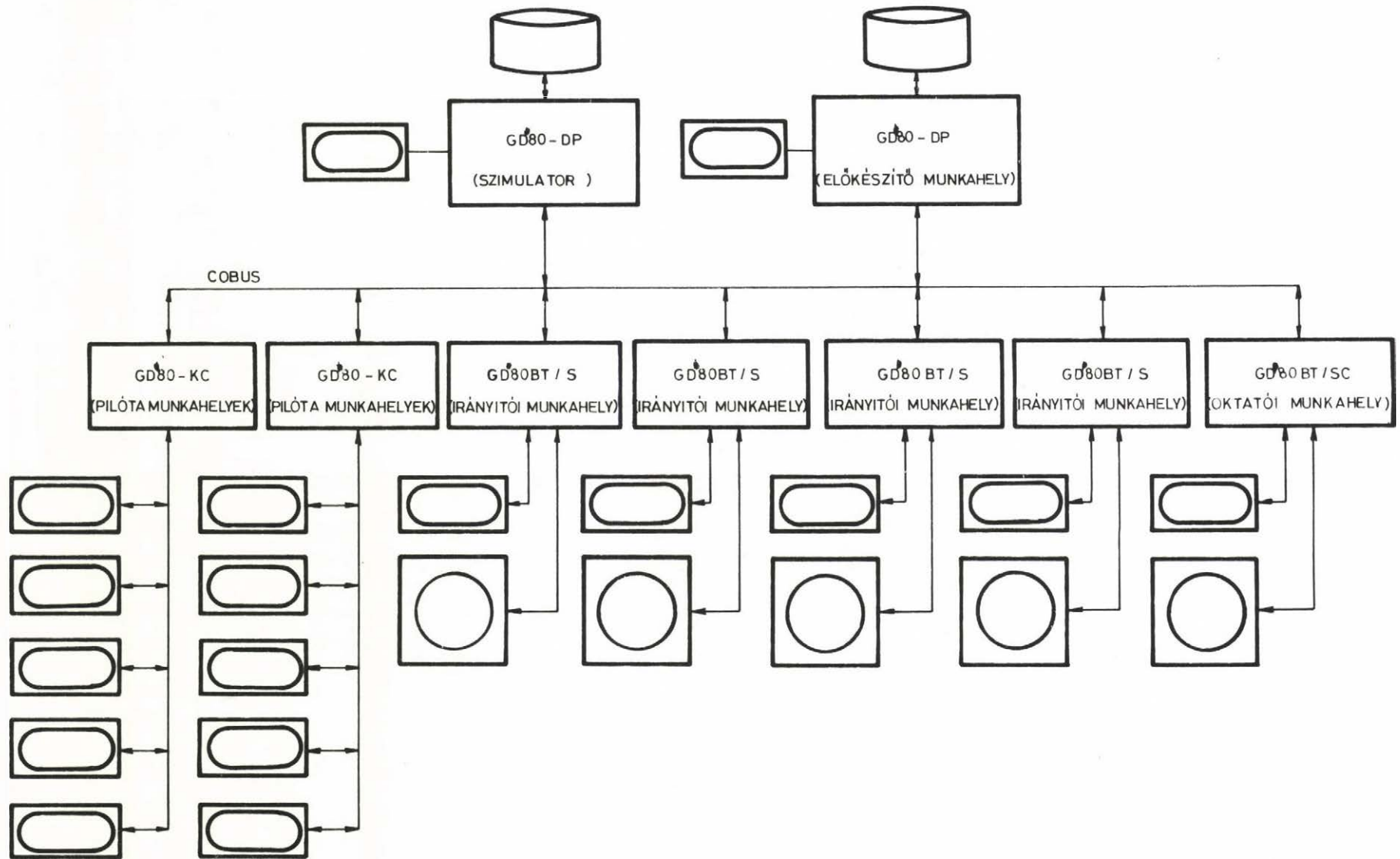
26. ábra: GD'80 KC alfanumerikus display koncentrátor



27. ábra: GD'80 TS



28. ábra: GD'80 DP



29. ábra: Oktatóberendezés légiforgalomirányítók kiképzéséhez

- 99/1979 Ivics József: KGST Riga
- 100/1979 Téli iskola
- 1980-ban jelentek meg:
- 101/1980 Gerencsér László - Hangos Katalin:
Diszkrét lineáris sztochasztikus rendszerek
önhangoló szabályozása
- 102/1980 Pásztorné Varga Katalin: Rekurzív eljárás
- 103/1980 Gerencsér Piroska - Szép Endre - Zilahy Ferenc
Marton Zsolt: Robotmegfogók adaptivitása I.
- 104/1980 Knuth Előd - Radó Péter - Tóth Árpád:
Az SDLA előzetes ismertetése
- 105/1980 E. Knuth - P. Radó - Á. Tóth:
Preliminary description of SDLA
- 106/1980 Prékopa András: Sztochasztikus programozási
modellek és alkalmazásuk
- 107/1980 Kelle Péter: Megbízhatósági készletmodellek és
alkalmazásuk
- 108/1980 Almásy Gedeon: Mérlegegyenletek és mérési
hibák
- 109/1980 Békéssy A. - Demetrovics J. - Gyepesi Gy.:
Relációs adatbázis logikai szintű vizsgálata
funkcionális függőségek szempontjából
- 110/1980 Gaál A. - Soltész J. - Ruda M. - Ratkó I.:
Tanulmányok a statisztikai adatfeldolgozásról
- 111/1980 Benedikt Szvetlána: Nem ismételheto döntéshozatal
analizise kockázattal járó esetekben

