

# Nagy adattömbökkel végzett tudományos számítások lehetőségei kishszámítógépes rendszerekben

Tudományos számításokban gyakran nagy mennyiségű aritmetikai művelet elvégzésére van szükség. Egyes alkalmazásokban másodpercenként több millió lebegőpontos művelet végrehajtása a követelmény (jelfeldolgozás, szeizmikus mérések, időjárás számítások stb.). Általános célú számítógépek közül csak kevés felel meg ennek. Ilyen nagy gépek alkalmazására csak ritkán van mód. Sokszor szükség van arra is, hogy az egész mérőrendszer könnyen mozgatható legyen, így csak kishszámítógépek alkalmazása jöhet szóba.

A kishszámítógépek aritmetikai teljesítőképessége nagyon korlátozott. A lebegőpontos műveleteket általában programmal, esetleg mikroprogrammal végzik el. Olyan alkalmazások céljára, ahol lényeges a gyorsabb aritmetikai műveletvégzés, a gyártók lebegőpontos aritmetikai processzorokat ajánlanak. Nagyobb teljesítményű kishgépes rendszerekben ezek és a központi processzor időben átlapoltan működnek (PDP 11/60—FP 11 E, PDP 11/70—FP 11 C). Az utasítások lehívását, a címszámítást és az operandusok átvitelét a számítógép, a lebegőpontos műveletet pedig a lebegőpontos processzor végzi el. Az egyszerű pontosságú aritmetikai műveletek végrehajtási idejét így 10  $\mu$ s, egyes típusoknál (FP 11 E) 1  $\mu$ s alá csökkentették.

A nagy tömegű aritmetikai művelet elvégzését igénylő számítások jelentős részére jellemző, hogy ugyanazt a műveletet nagy számú adaton nagyrészt egymástól függetlenül kell végrehajtani. A soros működésű számítógépekben ezt programhurokkal oldják meg. Ilyenkor csak az idő kis részében folyik aritmetikai műveletvégzés. Egy PDP 10 számítógépen futott FORTRAN nyelvű beszédfeldolgozó program művelettípusainak időarányai a következők voltak [1]:

keresés	10%,
tárolás	5%,
címszámítás	35%,
ciklusvezérlés	20%,
aritmetikai művelet	30%.

Soros számítógépről lévén szó, látható, hogy csupán az idő 30%-ában történik a probléma megoldása szempontjából hasznos műveletvégzés, a többit az adminisztráció emésztí fel. A CPU-val átlapoltan működő aritmetikai processzor alkalmazásakor az előző

arányokat feltételezve, optimális esetben a végrehajtási idő 30%-kal csökkenhet. Látható, hogy csupán az aritmetikai processzor sebességének növelésével nem csökkenthető jelentősen a végrehajtási idő.

A nagy tömegű aritmetikai művelet elvégzését igénylő tudományos számítások nagy részében (mátrixműveletek, lineáris transzformációk, korrelációs analízis stb.) csak egyszerű vezérlési és címszámítási műveletekre van szükség, így a kishszámítógépek változatos utasításkészlete nem használható jól ki. A vezérlési feladatokat jóval egyszerűbb eszközökkel gyorsabban el lehet végezni. Így születtek meg a vektorprocesszorok, amelyek egy vagy több nagy teljesítményű aritmetikai egység és egyszerű, de gyors vezérlő egység kombinációi.

Az angol nyelvű irodalomban ezeket „array processor”-nak nevezik ugyanúgy, mint az általam később tömbprocesszornak nevezett architektúrát.

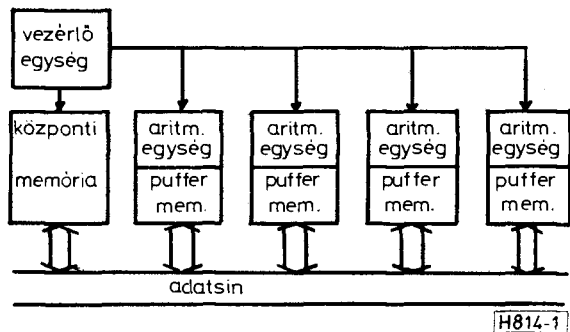
A vektorprocesszor és a kishszámítógép együtt egy kétprocesszoros rendszert alkot. A kishszámítógép vezérli az egész rendszert, irányítja a processzorok egymás közti és a külvilággal folytatott kommunikációját, a vektorprocesszor pedig a nagy tömegű aritmetikai műveletet tartalmazó algoritmusokat hajtja végre.

## Vektorprocesszor architektúrák

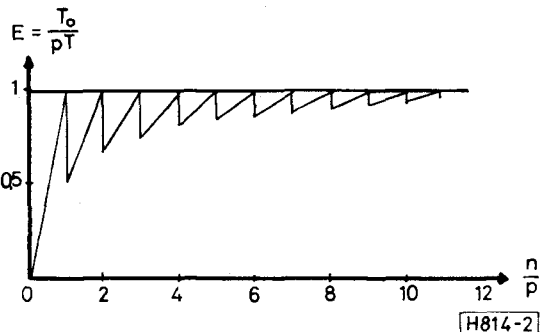
A vektorprocesszorok belső felépítése rendkívül változatos. Megtalálhatjuk bennük mindazokat az architektúrális elemeket, amelyek a párhuzamos adatfeldolgozási rendszerekre jellemzőek. Ezek az elemek rendszerint kombináltan fordulnak elő.

Céлом elsősorban a vektorprocesszorok architektúrális sajátosságainak ismertetése. Ezért nem az adat- és utasítássorozatok viszonyán, hanem a rendszer felépítésén alapuló csoportosítást használtam. Ez természetesen tükrözi az adat- és utasítássorozatok viszonyát is. Meg kell jegyezni, hogy a párhuzamos adatfeldolgozási rendszerek architektúrájára vonatkozó elnevezések sem a hazai, sem az angol nyelvű szakirodalomban nem egységesek.

Tömbprocesszor: sok alkalmazásban valamilyen adattömb elemein kell azonos, egymástól elszigetelt műveleteket végezni. Az ilyen feladatok gyorsan végrehajthatók olyan rendszerekkel, amelyekben egy központi vezérlő egység irányítása alatt több arit-



1. ábra. Tömbprocesszor-architektúra



2. ábra. Tömbprocesszor aritmetikai egységeinek kihasználtsága ( $E$ ) az aritmetikai egységek számának ( $p$ ) és a feldolgozandó adattömb elemei számának ( $n$ ) függvényében.  $T_0$  — egy művelet elvégzésének ideje az aritmetikai egységben,  $T$  —  $n$  művelet elvégzésének ideje  $p$  aritmetikai egységet tartalmazó tömbprocesszorban

metikai egység dolgozik. A központi vezérlő egység utasításait a megcímzett aritmetikai egységek saját memóriájukon hajtják végre. A tömbprocesszor az úgynevezett SIMD rendszerek jellegzetes típusa. Az elnevezés: Single Instruction Multiple Data Flow azt fejezi ki, hogy egyetlen utasítássorozat hatására több adatsorozaton történik műveletvégzés. Ha a műveletek nagy része egymástól független, a közös adatsínen viszonylag kis adatforgalom van. Az aritmetikai egységek egymás közti és a központi memóriával folytatott adatforgalmát a központi vezérlő vagy egy speciálisan erre a célra szolgáló sinvezérlő processzor irányítja (1. ábra).

Lényeges előnye ennek a rendszernek, hogy több azonos egységből áll, könnyen bővíthető, kialakítható az adott feladatnak leginkább megfelelő konfiguráció. Az egy tokba integrált aritmetikai processzorok, Intel 8087, Am 9512 stb. megjelenése lehetővé teszi kis helyigényű rendszerek létrehozását is.

A központi vezérlés megkönnyíti a szinkronizációt, ugyanakkor lehetővé teszi, hogy az aritmetikai egységek viszonylag egyszerűek legyenek.

A rendszer hátrányai két fő okra vezethetők vissza. Az egyik az, hogy az aritmetikai egységek kihasználtsága az aritmetikai egységek és a feldolgozandó adattömb elemei számának függvénye. A 2. ábra mutatja az aritmetikai egységek kihasználtságát abban az ideális esetben, ha egy adattömb elemein ugyanazt a műveletet kell egymástól függetlenül

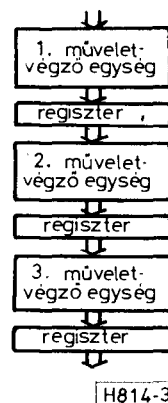
végrehajtani, és a közös adatsín nem növeli a műveletvégzési időt. Látható, hogy a kihasználtság akkor jó, ha a feldolgozandó adattömb elemeinek száma az aritmetikai egységek számának többszöröse, azt alulról közelíti, vagy az adattömb elemeinek száma jóval meghaladja az aritmetikai egységek számát.

A másik tényező, amely korlátozza a tömbprocesszorok alkalmazási körét az, hogy csak az algoritmusok szűk körében biztosítható, hogy nagyszámú aritmetikai egység hosszú ideig saját memóriáján dolgozhasson. Egyszerű vezérlő és adatátviteli rendszer mellett az egységek memóriái közti adatátvitel jelentősen növelik a műveletvégzési időt.

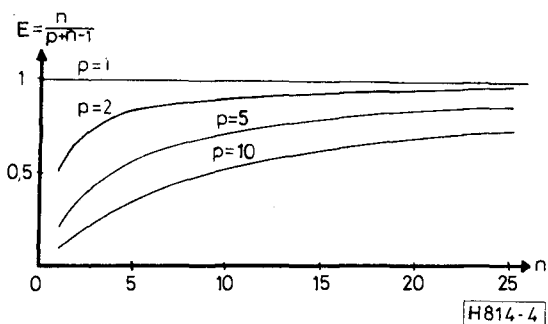
E két korlátozó feltétel miatt csak ritkán alkalmaznak négynél több aritmetikai egységet.

Multiprocesszorok: felépítésük gyakran a tömbprocesszorokéhoz hasonló, működésük azonban lényegesen különbözik azokétól. A MIMD (Multiple Instruction Multiple Data Flow) elvet valósítják meg. Mindegyik egység saját programját hajtja végre. A processzorok egymáshoz képest gyakran aszinkron módon működnek. Egy vagy több operatív memóriát közösen használnak. Sok esetben van saját memóriájuk is, így csökkenthető a memória használatából adódó konfliktusok száma. A különböző egységek nagyobb szeparált részfeladatokat hajtának végre. A processzorokat összekapcsoló rendszer több processzor esetén már igen bonyolult lehet. Ugyancsak nehezzé válik a szinkronizáció és az algoritmusok felosztása egymással csak lazán összefüggő párhuzamosan végrehajtható részfeladatokra. Ezért a vektorprocesszoroknál maximálisan 3–4 processzort alkalmaznak. Gyakori, hogy a konkrét algoritmustól függetlenül különvált feladatokra használnak külön processzorokat (input-output processzor, sinvezérlő stb.). Egy konkrét, de jellegzetes multiprocesszoros rendszert mutat be az 5. ábra.

Pipeline rendszerek: ezek a számítási rendszerek a futószalag elvén működnek (3. ábra). Egy folyamatot részfolyamatokra bontanak, és mindegyiket külön egységgel valósítják meg. Mindegyik egység elvégzi a bejövő adaton a szükséges műveletet, és továbbadja a következőnek. Amint egy egység felszabadult, újabb adat fogadására kész. Így az időegység alatt elvégezhető műveletek számát az ütemidő szabja meg, amely a teljes műveletvégzési időnek töredéke



3. ábra. Pipeline műveletvégző egység



4. ábra. Pipeline műveletvégző egység kihasználtsága ( $E$ ) a vezérlésátadási feltétel kiértékeléséig szükséges lépcsők számának ( $p$ ) és a feldolgozandó adattömb elemei számának függvényében

lehet. A pipeline rendszerek a MISD — Multiple Instruction Single Data Flow — elv reprezentációinak tekinthetők, mivel a pipeline-ban levő adatsorozatra több utasítás is hathat.

A pipeline elv az aritmetikai és a vezérlő egységekben is alkalmazható. Túlságosan sok részfolyamatra osztott pipeline esetén nagyon sok átmeneti tároló szükséges, és hosszadalmassá válik az olyan feltételes vezérlésátadó utasítások végrehajtása, amelyeknél a feltétel kiértékeléséhez szükséges az utoljára végrehajtott művelet eredménye. Ilyenkor csak azután juthat újabb adat a pipeline-ba, miután a kérdéses feltétel kiértékelődött. Feltéve, hogy ez a  $p$ -edik lépcsőben történik meg, és az adattömb  $n$  elemet tartalmaz, a műveletvégző egység kapacitásának ki-

használtsága  $\frac{n}{p+n-1}$  arányban romlik (4. ábra).

Az elvégzendő műveletek általában természetes módon feloszthatók néhány részfolyamatra, és ez nem igényel sem jelentős hardware többletet, sem programozási szemléletváltozást. Ez az oka annak, hogy egyre több berendezésben alkalmaznak néhány lépcsős pipeline-t.

Bit-soros rendszerek: a műveletvégzési idő csökkentésének bevált módszere a párhuzamos műveletvégzés. Ennek ellenére kisebb sebességű vektorprocesszorok esetében jól beváltak olyan rendszerek, amelyekben az aritmetikai műveletek végzése bitenként sorosan történik. Így nagyon egyszerű, olcsó berendezések hozhatók létre.

Mivel egy soros műveletvégző egységben az eredmény első bitje már a művelet első vagy második ciklusában képződik, az egymás után képződő bitek egy másik soros műveletvégző egységben már az előző művelet befejezése előtt felhasználhatók. Ezt csak akkor lehet kihasználni, ha a műveletvégzés viszonylag hosszabb ideig sorosan történik. A ma általánosan használt operatív táruk szavas szervezésűek, ezért a soros műveletvégzésből származó előny kihasználását erősen korlátozzák. Ilyen társzervezés mellett csak úgy hozhatók létre gyors processzorok, ha a műveletvégző egységek viszonylag bonyolultabb műveleteket hajtanak végre anélkül, hogy a szavas szervezésű memóriához fordulnának.

Mivel a lebegőpontos számábrázolás esetén a normalizálás csak a legmagasabb helyi értékű bit ismeretében végezhető el, a rendszer előnyei csak fixpontos számábrázolású vektorprocesszorokban mutatkoznak meg.

#### A műveletvégző egységek szinkronizálása

Minden, gyakorlatban megvalósított vektorprocesszor több, többé-kevésbé független műveletvégző egységet tartalmaz. Aszinkron rendszereknél minden egység maximális sebességgel dolgozhat, időnként azonban várakoznia kell más egységekre. Ha a műveletvégző egységeknek sűrűn kell egymással kapcsolatba lépniük, a szinkronizáció okozta idővesztés nagyobb lehet, mint a maximális sebességből adódó nyereség.

Szinkron működés esetén valamennyi egység a leglassabbnak a sebességével dolgozik, szinkronizálásuk azonban nem okoz gondot.

Egyes architektúrák alapvető eleme a szinkronizmus (pipeline, bit-soros rendszerek), más architektúrák aszinkron működés esetén előnyösek.

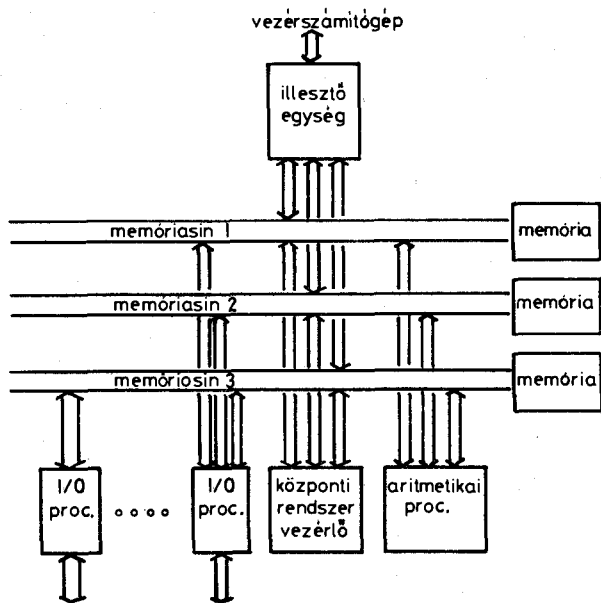
Azt, hogy egy adott célra szinkron vagy aszinkron vezérlés kedvezőbb, a feladat jellege dönti el. Ha több, egymással csak lazán összefüggő, hosszabb részfeladatra bontható az algoritmus, aszinkron rendszer előnyösebb. Ha ez nem tehető meg, és a különböző egységek sebessége nem tér el jelentősen egymástól, akkor a szinkron rendszerek mind programozásukat, mind felépítésüket tekintve egyszerűbbek és gyorsabbak.

#### A vezérszámítógép és a vektorprocesszor kapcsolata

A vezérszámítógép és a vektorprocesszor összekapcsolása többféleképpen lehetséges. A legelterjedtebbek azok a rendszerek, amelyekben a vektorprocesszor a kisszámítógép valamelyik buszára csatlakozik.

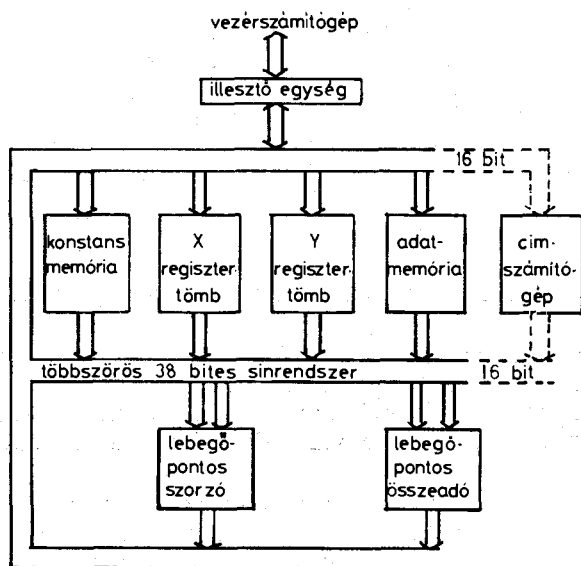
A lazán csatolt vektorprocesszoroknak nagy belső memóriájuk van. Működésük során csak a bemenő és kimenő adattömbök átvitelekor fordulnak a vezérszámítógéphez. Az adatátvitel DMA vagy programozott adatátviteli csatornán keresztül történik. Működés közben csak belső memóriájukat használják, így maximális sebességgel dolgozhatnak. A tekintélyes adatátviteli idők miatt akkor gazdaságosak, ha nagy mennyiségű számítást önállóan hajthatnak végre. Ez esetben a műveletvégzési időkhöz képest a szinkronizálási idők elhanyagolhatók, így a vezérszámítógép felépítése csak az adatátviteli időket befolyásolhatja, a számítási sebességet nem. Nagy előnyük, hogy könnyen illeszthetők a legkülönbözőbb szervezésű és méretű számítógépekhez.

A szorosan csatolt vektorprocesszorok a vezérszámítógéppel integrált egységet képeznek. Kiseb belső memóriájuk van, működésük közben gyakran fordulnak a vezérszámítógép operatív tárához. A legegyszerűbb típusokban csak néhány regiszter található.



H814-5

5. ábra. A MAP vektorprocesszorok felépítése [2]



H614-6

6. ábra. Az AP 120 B vektorprocesszor felépítése [3]

Ha a kiszámítógépek valamelyik gyors belső buszára illesztik ezeket a vektorprocesszorokat, az elveszti univerzalitását, csak egy bizonyos típusú számítógéppel képes együttműködni. Ennek fejében gyorsabban tud hozzáférni a vezérszámítógép operatív memóriájához, és használhatja annak egyéb szolgáltatásait is (cash, memory management).

Bár a félvezető memóriák egyre olcsóbbak, a kiszámítógépek árának ma is jelentős részét képezik. Mivel számításigényes algoritmusokban gyakran igen nagy memóriaterületre van szükség, a szorosan csatolt vektorprocesszorok jóval olcsóbbak a lazán csatoltaknál.

MAP 100, 200, 300: A CSP Inc. vektorprocesszor családjának tagjai, többszoros rendszerek [2] (5. ábra). A processzorok három memóriasínnre csatlakoznak. A központi rendszervezérlő irányítja a rendszer működését. A három memóriasínen és a belső IT rendszeren a többi processzorral, az illesztő egységen és a vezérszámítógép IT rendszerén keresztül pedig a vezérszámítógéppel tartja a kapcsolatot.

Az aritmetikai processzor végzi az aritmetikai műveleteket, az I/O processzor irányítja a környezettel és a saját tömegtároló egységekkel folytatott adatátvitelt. Mindegyik processzor saját programját hajtja végre, a központi rendszervezérlő végzi a szinkronizálást és a többi processzor viszonylag kis (128 szó) programmemóriájának újratöltését.

Egy szorzás és két összeadás végrehajtásának ideje típustól függően 210 ns...1,6  $\mu$ s. A használt számábrázolás az IBM 32 bites, lebegőpontos formátuma.

AP 120 B: a Floating Point Systems vektorprocesszora talán a legnépszerűbb kategóriájában [3] (6. ábra). DMA vagy programozott I/O csatornán keresztül csatlakozik a vezérszámítógéphez. Funkcionálisan osztott műveletvégző egységei lehetővé teszik a sokszorosán átlapoltt műveletvégzést úgy, hogy többé-kevésbé sikerült megőrizni a programozás soros jellegét. A rendszer elemei egy párhuzamos sínrendszeren keresztül kapcsolódnak egymáshoz. Két regisztertömb, egy gyors konstans memória és egy viszonylag lassabb (330 ns hozzáférési idejű) adatmemória biztosítja a szükséges adat sávszélességet.

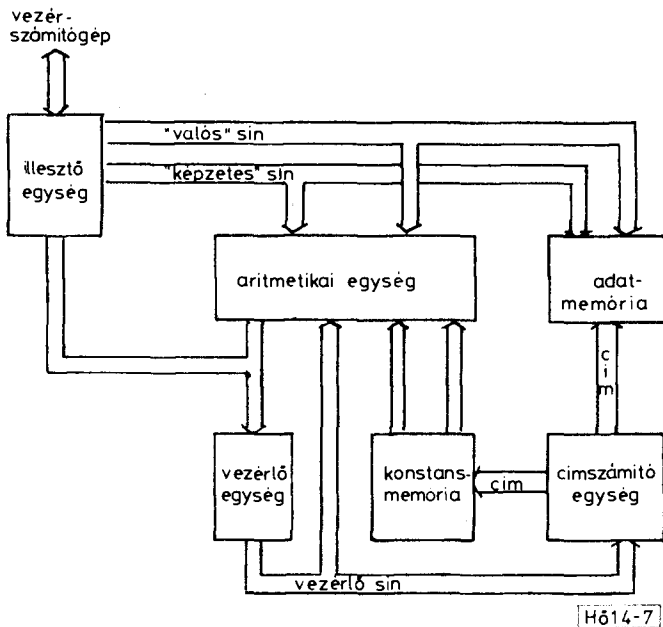
Egy kétlépcsős lebegőpontos összeadó és egy háromlépcsős lebegőpontos szorzó pipeline végzi az aritmetikai műveleteket. Az ütemidő 167 ns, ez egyben az adatmemória kivételével a többi egység ciklusideje is. A címszámítógép számítja ki az operandusok címét, és végzi el az egyéb egész típusú aritmetikai és logikai műveleteket. A vezérlő egység szinkron módon vezérli az összes műveletvégző egységet. Ez 64 bit hosszúságú programszavakkal történik.

Ez a vektorprocesszor 38 bites belső számábrázolást használ, de a cég még hasonló típusokat is ajánl ennél hosszabb adatformátumokkal.

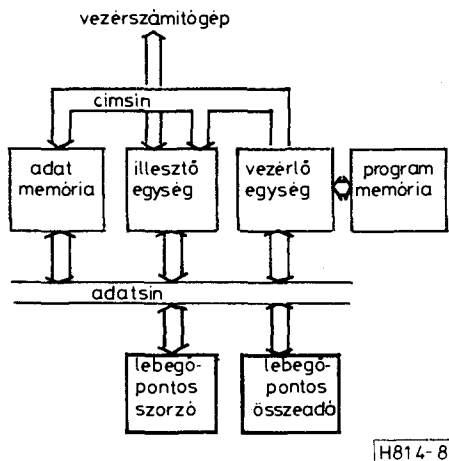
MSP 3: a CDA vektorprocesszora a legkisebbek közé tartozik [4]. Két  $39,8 \times 21,4$  cm-es kártyán helyezkedik el, ára meg sem közelíti az eddig leírt típusokét. Művelti sebességét tekintve méretéhez és árához képest igen előkelő helyet foglal el. Az 1024 pontos komplex Fourier-transzformáció elvégzésének ideje pl. az eddig leírt gépek esetében a következőképpen alakul:

MAP 100	60 ms,
MAP 300	4,5 ms,
AP 120 B	7,5 ms,
MSP 3	13 ms.

Az egyszerű felépítést bit-soros műveletvégzéssel valósították meg. A műveletvégzési sebesség növelése érdekében olyan összetett aritmetikai egységet építettek, amely pl. a Fourier-transzformációban fontos szerepet játszó pillangó műveletet komplex számokkal egyetlen lépésben végzi el.



7. ábra. Az MSP 3 vektorprocesszor felépítése [4]



8. ábra. A tervezett vektorprocesszor felépítése

A bit-soros műveletvégzés előnyeit csak úgy tudták kihasználni, hogy a fixpontos számábrázoláshoz nagyon közel álló ún. blokklebegőpontos számábrázolást használnak. Ez azt jelenti, hogy minden adatbloknak egy karakterisztikája van. Így gyakorlatilag a lebegőpontos számábrázolás minden előnyét elvesztették.

Gyors adatmemóriája 2048 szó kapacitású (egy szó  $2 \times 24$  bit). Külön konstansmemória szolgál a számításokban szereplő konstansok (pl. egységgyökök) tárolására. A címszámító egység végzi mind az adat-, mind a konstansmemória címeinek kiszámítását. Az MSP 3 belső felépítése a 7. ábrán látható.

#### Egy tervezés alatt álló vektorprocesszor

A bevezetőben már említett okok miatt a Központi Fizikai Kutató Intézetben elkezdődött egy lebegőpontos, szorosan csatolt vektorprocesszor kifejlesztése.

aritmetikai műveleti kód	vezérlésátadás kódja	címszámítási műveleti kód	adatátvitel kódjai
--------------------------	----------------------	---------------------------	--------------------

H814-9

9. ábra. A tervezett vektorprocesszor programszavának felépítése

tése. Célunk a TPA 11/440 kisszámítógép alkalmazása tétele nagy aritmetikai számításigényű algoritmusok végrehajtására.

A vektorprocesszor a kisszámítógép gyors, 32 bites rendszerbuszára csatlakozik, így biztosított a gyors hozzáférés a vezérszámítógép egész memóriaterületéhez. A cash használata tovább rövidíti az átlagos hozzáférési időt.

A vektorprocesszor belső számábrázolása azonos a TPA gépek lebegőpontos számábrázolásával. Egyszeres és kétszeres pontosságú műveletek végzésére alkalmas. Belső felépítését a 8. ábra mutatja.

A vezérlő egység szinkron módon irányítja a többi egység működését, végzi a címek kiszámítását, és irányítja az adatáramlást. Így az egységek szinkronizációja egyszerűvé vált.

A lebegőpontos szorzó a szorzást, reciprokképzést és négyzetgyökvonást, az összeadó az összeadást, kivonást, abszolút érték összehasonlítást és a konverziós műveleteket végzi el. Mindkét aritmetikai egység kétlépcsős, pipeline szervezésű. A szorzás és összeadás ütemideje 480 ns, így másodpercenként 4 millió lebegőpontos művelet elvégzésére van lehetőség.

Mind a program, mind az adatmemória ciklusideje 160 ns. Be- és kimeneteiken pipeline regiszterek biztosítják a többi egységgel való átlapolt műveletvégzést. Az adatsín is szinkron működésű, ciklusonként két adatátvitelre van szükség.

A vektorprocesszorok programozása 64 bites programszavakkal történik. Ez egyszerre vezérli valamennyi egység működését (9. ábra).

A vektorprocesszorok az elsők voltak abban a sorban, mely ma már több olyan eszközt tartalmaz, amely alkalmazható a kisszámítógépeket a legkülönbözőbb olyan feladat elvégzésére, amelyhez eddig nagyszámítógép kellett. Egymás után jelennek meg más jellegű feladatok megoldását célzó pl. (asszociatív) processzorok is.

#### I R O D A L O M

- [1] Reddy, D. R.: Some Numerical Problems in Artificial Intelligence: Implications for Complexity and Machine Architecture. Traub, J. ed.: Complexity of Sequential and Parallel Numerical Algorithms. Academic Press, New York, 1973.
- [2] An Introduction to the MAP Series Models 100, 200&300. CSP Inc. Document No. S-02, 1975.
- [3] Ap 120 B Floating Point Array Transform Processor. Floating Point Systems Form 7259 R 1975.
- [4] Micro Signal Processor 3. Computer Design and Applications Inc. Publication No. 50023.