

Az EMG 830 aritmetikai rendszere

(Kóta Gábor)

Ennek az anyagnak az elkészítésekor arra törekedtem, hogy leírjam azokat a dolgokat, vagy legalábbis azok jelentősebb részét, amelyeket az EMG 830 aritmetikai rendszeréről és annak fejlesztési munkájáról az utókor számára megőrzésre érdemesnek gondolok. Igyekeztem mindezt megfelelő elméleti és történeti keretben bemutatni. A technikai leírásban összefoglaltam az elméleti alapokat és az alkalmazott algoritmusokat is. Technika-történeti anyagról lévén szó, igyekeztem némi történeti háttérrel is adni. A technikai részletek iránt kevésbé érdeklődő olvasó ezeket a részeket könnyedén átugorhatja, remélhetőleg így is képet kaphat arról, milyen is volt az EMG 830 aritmetikája a hatvanas évek végén. És természetesen nem hiányozhatnak a leírásból a személyes emlékek sem.

Mi is az aritmetikai egység?

Neumann alapvető művében, a „First Draft of a Report on the EDVAC”-ban így fogalmaz „First: Since the device is primarily a computer, it will have to perform the elementary operations of arithmetics most frequently. These are addition, subtraction, multiplication and division: +; -; *; /.. It is therefore reasonable that it should contain specialized organs for just these operations”

Az első számítógépeket valóban számítások elvégzésére készítették, ezért az aritmetikai egység kiemelt jelentőségű volt. Azt is mondhatjuk, hogy az aritmetika **a számítógép számológépe**.

Számos technikatörténeti mű foglalkozik a számrendszerek és a számológépek történetével, az abakusztól a bináris aritmetikáig. Ebből csak a témánk szempontjából legérdekesebbeket emeljük ki. A XVII. században készültek el az első **mechanikus számológépek**, ezek közül a következőket ildomos megnevezni:

- 1623: Wilhelm Schickard: első fogaskerekes számológép
- 1642 Blaise Pascal kis számológépe 8 jegy összeadására és kivonására
- 1673 Gottfried Wilhelm von Leibniz: szorozni és osztani is tudott.

Ezek a gépek helyértékenként egy-egy **tízfogú fogaskereket** alkalmaztak, a kerék állása felelt meg a számjegyeknek. Az összeadás úgy történt, hogy a fogaskereket az összeadónak megfelelő számú foggal **továbbforgatták**, ha a kerék körbefordult, akkor egy büttyök eggyel továbbléptette az eggyel magasabb helyértékű fogaskereket, ezzel megvalósítva az átvitelt.

Az első általános célú elektronikus digitális számítógép, az 1943-46 között kifejlesztett **ENIAC** is ugyanezt az elvet alkalmazta. 10 flip-flop-ból **gyűrűsszámlálót** alakítottak ki, a flip-flop-okból mindig csak egy volt 1-es állapotban. Egy flip-flop-hoz két elektroncső kellett. A hozzáadás **elektromos impulzusokkal** történt, minden impulzus eggyel továbbléptette az 1-es értéket. 20 akkumulátorának mindegyikével lehetett párhuzamosan összeadást vagy kivonást végezni, a szorzásra, osztásra és négyzetgyökvonásra (ez is alapl művelet volt) kijelölt akkumulátorok voltak. A gép 100 kHz-es órajellel működött, 12 jegyre 5000 összeadást tudott másodpercenként (20 órajel). A szorzás legfeljebb 14, az osztás kb. 143 összeadásidő volt. Az

ENIAC 17000-nél több elektroncsövet tartalmazott, szemben a kor számítógépeiben szokásos 4000-6000 elektroncsővel vagy jelfogóval. A projektet az amerikai hadsereg finanszírozta...

Az 1960-as évek második felében a VATUKI feladatuk kapta egy 2 fővonalas 20 vonalas elektronikus telefon-alközpont kifejlesztését a kormányvonal számára. A fiatal tervező jól ismerte a telefonközpontokat, de nem volt tapasztalata a digitális technikában. EMG 4000-es digitális elemekkel dolgozott, az elkészült telefonközpont időnként működött is, de sok hibával. Ezért az EMG-től kértek szaktanácsadást. Néhány év digitális tervezéssel a hátunk mögött mosolyogtunk azon a megoldáson, hogy a tervező az elektromechanikus híváskeresőt egy az egyben leképezte flip-flop-okból álló gyűrűre. Most már megértőbb vagyok. Egy vadonatúj technika szabályai csak utólag magától értetődőek...

A kettes számrendszer

Neumann, aki később kapcsolódott be az ENIAC fejlesztésébe, említett jelentésében a bináris számrendszer mellett teszi le a voksot. A kettes számrendszer gondolata nem új. Már **1679**-ben **Leibniz** kidolgozta a kettes számrendszer és a bináris aritmetika elméletét. Az ENIAC kivételével az első digitális számítógépek (ABC, MARK-I, ZUSE., stb.) is bináris aritmetikát alkalmaztak. Sőt, már az 1930-as években R. Valtat szabadalmaztatott egy bináris számológépet, input és output decimális-bináris átalakítóval. Ezek az átalakítások azonban komplikálttá tették a gépet. A műszaki-tudományos számításokra készített számítógépekre az átalakítások bőségesen megtérültek a bináris aritmetika nagyobb sebessége miatt.

Az ENIAC csapata (Eckert, Mauchly, Goldstine) is az **EDVAC**-nál már kettes számrendszert alkalmazott. A Neumann által ajánlott soros feldolgozással talán kissé túllőttek a célon, amikor bitenként sorosan adtak össze. Az EDVAC 44 bitre 1160 összeadást és kb. 350 szorzást tudott másodpercenként, az osztást programozottan hajtotta végre. Mintegy 6000 elektroncsövet tartalmazott. (A csökkenés nem csak a kettes számrendszernek volt köszönhető, hanem annak is, hogy rövid idejű tárolásra higanyos késleltető művonalat alkalmaztak.)

A néhány évvel később kezdődő **adattfeldolgozási alkalmazásoknál** azonban már nem egészen ez a helyzet. Ahol az adatokon mindössze néhány egyszerű műveletet végeznek el, és a hangsúly a nagy tömegű adat tárolásán és visszakeresésén van, előnyösebb megtartani a tízes számrendszert, az egyes számjegyeket azonban már binárisan kódolni 4 biten. Ez az ún. binárisan kódolt decimális (**BCD**) számbábrázolás.

A **műszaki-tudományos számítások** nem egész, hanem **valós számokkal** dolgoznak. Felmerül a tízes (bináris) vessző elhelyezésének a problémája. A legegyszerűbb és általános megoldás a **vesszőt** (pontot) a gépi szó elején – pontosabban egy előjel-bit után - rögzíteni, és a szó bináris helyértékeihez sorban 2 negatív hatványait rendelni. Ennél az ún. **fixpontos számbábrázolásnál** tehát a sorrend felülről (előlről) lefelé: előjel, az $\frac{1}{2}$, $\frac{1}{4}$, stb. helyérték. A leírható számtartomány így -1 és +1 közé esik (a +1 már nem).

A valós számot tehát normalizálni kell, -1 és 1 közé kell eltolni (mantissza), és tárolni a normalizálás lépéseinek a számát (karakterisztika). Ennek az ún. **lebegőpontos ábrázolásnak** a gépi realizálása azonban meglehetősen bonyolult, ezért az aritmetikai egységek általában csak a rögzített tízesű törtszámokon végeztek műveleteket, a karakterisztika kezelését a

szoftverre bízták. Bár ZUSE gépe már rendelkezett lebegőpontos aritmetikával, azok inkább csak később és a nagyobb gépeken terjedtek el.

Hogyan lehet a **negatív számokat** ábrázolni? A negatív kivonást jelent, a számológépeken a kivonáshoz a fogaskerekeket visszafelé tekerték, az ENIAC-ban a gyűrűs-számlálókat visszafelé léptették. Ugyanezt az elvet lehet alkalmazni a kettes számrendszerben is: a negatív számot úgy ábrázoljuk, hogy az abszolút értéket binárisan kivonjuk nullából. Ekkor a bináris szó elején 1-es (vagy 1-es sorozat) jelenik meg:

-1: 111...111

Ezek után a legfelső bitet előjelnek tekintjük: 0 jelenti a pozitív számot, 1 a negatívot. Ezt az ábrázolást **kettes komplementnek** nevezik, és az az előnye, hogy az összeadásnál és kivonásnál nem kell tekintettel lenni az előjelre (sőt a szorzásnál és osztásnál sem!), csak arra kell figyelni, hogy az eredmény ne csorogjon be az előjel-bitre, azaz ne legyen 1 vagy annál nagyobb, és -1-nél kisebb. Ezt nevezik túlsordulásnak. A kettes komplementet a gépben úgy képezhetjük, hogy bitenként negálunk, alul pedig 1-t hozzáadunk (kezdeti átvitel).

Aritmetikai műveletek

Az aritmetikai műveletek központi eleme az ún. **akkumulátor** regiszter, amely az additív műveletek egyik operandusát tartalmazza. A másik operandust az utasítás jelöli ki a memóriából. Egy összeadáshoz tehát először az egyik operandust az akkumulátorba kell vinni, ezután jön az összeadás művelete, majd az eredmény tárhoz írása. Az eredményen gyakran további műveleteket is végre kell hajtani, ezért gazdaságos, hogy az eredmény nem kerül közvetlenül a tárhoz. A szorzáshoz és osztáshoz egy további regiszter szükséges, de arra nem kell tudni összeadni. Több akkumulátor használata gyorsítja a számításokat, de a hardver így bonyolultabbá válik, ezért ez inkább csak a későbbi gépeken terjedt el.

Az **összeadás** nélkülözhetetlen alpművelet. **Kivonás** művelettel a legtöbb gép rendelkezik, ezt a kivonandó komplementjének (negatívjának) az akkumulátorhoz adásával hajtják végre. De például a hatvanas évek legendás 12 bites kisgépe, a PDP-8 – ez az a gép, amelynek az utasításrendszerét a KFKI átvette az EMG 830-al együtt 1968-ban elkészült TPA 1001-es gépéhez – nem rendelkezett kivonás művelettel, csak komplement-képzéssel. A kivonáshoz először a kivonandót kellett az akkumulátorba vinni, ott a komplementjét képezni, majd hozzáadni a kisebbítendőhöz.

A **szorzást és osztást** nem kell feltétlenül egyetlen gépi utasítással végrehajtani. Költségkímélőbb (és természetesen jóval lassabb) megoldás ezeket a műveleteket összeadás, kivonás és jobbra ill. balra léptetés műveletekkel szoftverben megoldani.

Lebegőpontos és **decimális** aritmetikával is inkább a nagyobb teljesítményű gépek rendelkeztek, többnyire opcionális egységként.

EMG 830 aritmetikai egységek

A modularitás elvének megfelelően az aritmetikai műveletek két modulban kerültek megvalósításra:

- Alaparitmetika modul, amely az alapvető műveleteket, mint az összeadást, kivonást, kétirányú léptetést és bizonyos logikai műveleteket volt képes végrehajtani.
- Kiegészítő modul, amely a szorzás és osztás elvégzését vezérelte

A decimális műveleteket az eredeti tervek szerint két párhuzamos modul valósította volna meg, az első dokumentációk ezt a változatot írják le. Végül helyette egyetlen komplex bináris/decimális aritmetikai egység készült el.

Az egyes modulokat, azok lemezeinek és mikrokardjainak számát, a fejlesztés idejét és a tervezőket a következőkben soroljuk fel:

- A1 21 bites bináris alaparitmetika
4 lemez, 640 mk.
1966-68
- A2 21 bites bináris * /.
3 lemez, 360mk.
1966-69 Sárossy József és Kóta Gábor
- A3 24 bites bináris alaparitmetika.
4 lemez, 600 mk.
1968
- A4 24 bites bináris * /.
3 lemez, 360 mk.
1969 Kóta Gábor
- A5 decimális alaparitmetika – nem készült el
- A6 decimális * / - nem készült el
- A7 bináris/decimális komplex egység.
8 lemez, 1200 mk.
1969-70 Terv: Kóta Gábor, kivitelezés: Békássy Csaba

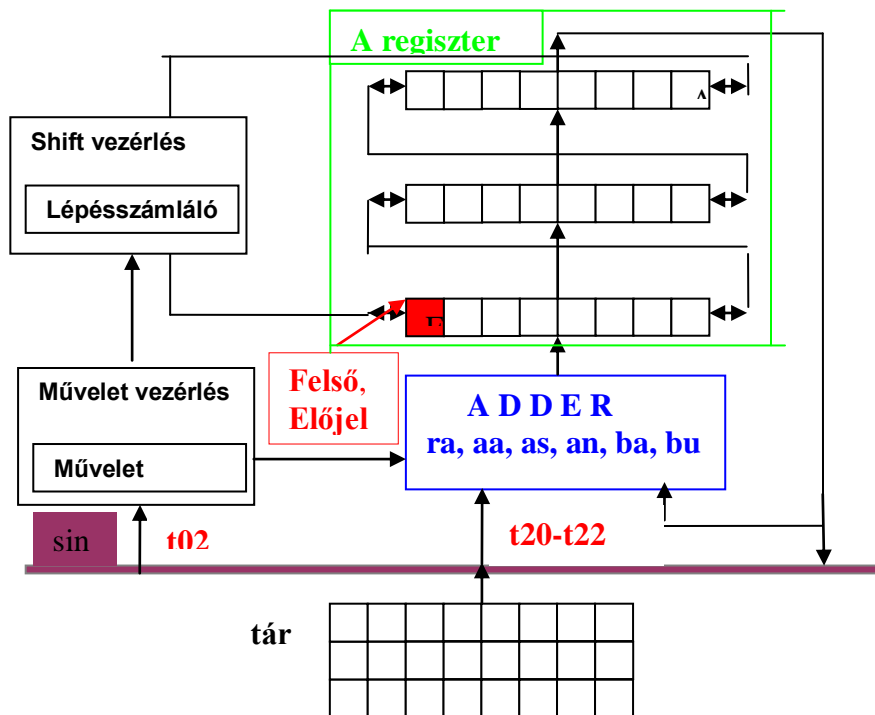
A1 és A3 alaparitmetika modulok

A modulok tartalmazták a A-regisztert (az akkumulátort), amely a művelet egyik operandusát képezte. A másik operandust az utasítás adta meg.

A műveletek:

- A-regiszter töltése (ra), kiírása (wa)
- Additív: + (aa), - (as), op-A(an),
+/- abszolút érték (pa, na)
- Logikai ÉS (ba), ANTIVALENCIA(bu)
- Léptetés A-ban (sh): 1-64, jobbra/balra:
 - logikai (0 jön be)
 - aritmetikai (lefelé előjel marad)
 - ciklikus
 - normalizálás (ws)
- Feltételes ugrások: $A=0$ (jz), $A<>0$ (jn), $A<0$ (jn), $A\geq 0$ (jp) túlszordulás (jo)

Az **A3 modul** sémáját a következő ábra mutatja:



Az additív műveletek 3 lépésben, **8 bitre párhuzamosan** kerülnek végrehajtásra. Az „adder” 8 bites. Átviteli láncá képezte a gép leghosszabb logikai láncát, amely még éppen jól belefért a 4 mikrosec-es ütembe. Az A-regiszter is 3-szor 8 bites felbontású. Szükség van az A-regiszter jobbra és balra léptetésére is, így a keresztirányú léptetéssel együtt 3 irányú léptetést végezhet.

Egy művelet végrehajtása úgy történik, hogy a **t02-es műveleti kód ütemben** az aritmetikai egység beveszi és letárolja a sínról a műveleti kódot. Additív műveleteknél a **t20-t21-t22 adatütemekben** a sínen megjelenő adat és az A-regiszter tartalmával végrehajtja a műveletet. A léptetések a t20-as ütemben történnek, ütemenként egy lépést hajt végre.

Az EMG **830-10** típusnál az adatokban minden 8 bitből 7 bit jelentett valódi adatot, a 8. bit a **paritás** volt. Az adatszó 21 bites volt. Az eredmény mellett annak paritásbitjét is képezni kellett, ennek az optimalizálása alapos logikai tervezést igényelt. Azt használtuk ki, hogy az összeadás átvitelek nélkül helyértékekenként éppen a logikai antivalenciát képezi, átvitel esetén a következő helyértéken pedig éppen fordít azon. Az operandusok paritásbitjéből kiindulva és az átviteleket figyelve sikerült végül az összeadó egységet úgy megvalósítani, hogy az pont elférjen egy lemezen.

Az A1 és A3 modul 4-4 lemezen készült el, amelyek között szerencsés felbontást sikerült elérni, minden lemez egy-egy funkcionális egységet alkotott: A-regiszter, összeadó, léptetés-vezérlés, utasítás-vezérlés.

A2 és A4 kiegészítő modulok

A szorzás és osztás vezérlését végezték el. Ezekben a műveletekben még egy regiszter, az ugyancsak 21+3 ill. 24 bites Q-regiszter is részt vett.

Bináris szorzás

Nézzük először a bináris szorzás **legegyszerűbb algoritmusát!** A decimális szorzásnál megszokott módon kell eljárni, de a részletösszeget nem kell külön képezni, az a szorzó jegy 1 értékénél megegyezik a szorzandóval, 0 értékénél 0. Írjunk fel egy példát, 4 bitre, de a szokásostól eltérően a szorzó legalsó jegyétől kiindulva:

```
0100 * 0101
-----
   0100 1
  0000 0
 0100 1
0000 0
-----
0010100
```

Az eredmény 7 jegyen, $2 * \text{szóhossz} - 1$ hosszúságban képződött. Ezt egy dupla szóban, 8 jegyben helyezhetjük el. Ebben egész értelmezésben lefelé, törtnél felfelé kell az eredményt igazítani.

A gépi megvalósításhoz azt kell mindössze felismernünk, hogy összeadáskor ne a részletösszegeket toljuk el 1-1 helyértékkel, hanem a rész-eredményeket tároló akkumulátort.

Az alábbi ábra jobboldala mutatja a **gépi megvalósítást**. A szorzandó az operandus, a **szorzót** Q-ban helyezzük el, az A akkumulátor 0-ból indul.

Vegyük észre, hogy a dupla hosszúságú **eredmény** tárolásához nem kell újabb regiszter. A léptetések során Q-ból sorban kilépnek a szorzóbiték, a fent megüresedő helyre betehetjük az eredmény bitjét. Az eredményt az A és Q regiszterekben folyamatosan (nevezzük AQ-nak) kapjuk meg. Az EMG 830 a szorzást tört számokra végezte el, egész szám értelmezésben AQ-ban a szorzat kétszerese látható.

```
0100 * 0101   A    Q = szorzó
-----
->            0000 0101 szorzójegy
   0100 1     0000 0010 1   szorzójegy ki
->            0100
   0000 0     0010 0001 0   eredmény be
->            0001 0000 1   szorzójegy ki
   0100 1     0101
->            0010 1000 0
0000 0       törtre kész
-----
0010100
```

A szorzás viszonylag hosszú művelet, végrehajtási ideje a szükséges összeadások számától függ. Annak csökkentésére az EMG 830 egy egyszerűen megvalósítható **gyorsított algoritmust** használt. Eredetét nem ismerem, Reitwiesner írta le egy könyvében. Az algoritmus elve az, hogy az összeadás mellett a kivonást is alkalmazva az összeadások/kivonások száma csökkenthető. Az algoritmus a szorzót átértelmezi egy képzel - 1, 0, 1-es számrendszerbe az 1-es sorozatok és az abban közbeékelten egyetlen 0 jegy felismerésével. 0111 például felfogható 100-1 -ként, 01011 pedig 10-10-1 -ként. Így egy összeadás vagy kivonás után minimum két léptetés következik, az additív műveletek száma maximum a szóhossz fele.

A gépi megvalósítás 3 bitet figyelt. Q-t egy flip-floppal (Q(-1)) megtoldotta, amely 0-ból indul, aztán Q(0) előző értékét tárolja. Ha $Q(0) = Q(-1)$, csak léptetés történik. Ha nem egyenlő, akkor Q(1) értékétől függően A-ban összeadás ($Q[1] = 0$) vagy kivonás ($Q[1]=1$) történik. Ezután két léptetés következik.

Az algoritmust teljesen az A2 ill. A4 modul valósította meg. Ehhez a sín 3 jelét (bs19, bs20, bse) használta fel, amelyeken keresztül jelezte, hogy léptetés, összeadás vagy kivonás következik-e, illetve befejeződött-e a művelet.

Bináris osztás

Az osztás algoritmus a szokásos eljárásból indul ki:

- ha az osztandó/maradék nem kisebb az osztónál, a hányados-jegy 1 és kivonás következik
- ha kisebb, a hányados-jegy 0, nincs kivonás

Induljunk ki az előző szorzás végeredményéből, de adjunk ahhoz 10-t, hogy maradékot is kapjunk! AQ tartalmazza az osztandót, az operandus az osztót. Az osztást fentről lefelé végezzük el, összehasonlításokkal: Ha az osztandó/maradék kisebb az osztónál, a hányados-bit 0, a maradékot felfelé léptetjük. Ha az osztandó/maradék nagyobb az osztónál, vagy egyenlő azzal, akkor a hányados-bit 1, és az osztót kivonjuk a maradékból, ezután a maradékot felfelé (balra) léptetjük. Ezt az eljárást ismételjük, amíg az osztandó végére nem érünk.

```

Osztó (operandus)
0100
A           Q
0010 1010   <: 0, csak léptetés
0101 010    0 >: 1, kivonás
0001 010    -
0010 10     01 <
0101 0      010 >
0001 0      -
0010       0101

```

Itt sem szükséges további regiszter alkalmazása. Az eredményt Q-ban, a maradékot A-ban kapjuk.

Az EMG 830-ban az **összehasonlítás kivonással** történt. Ha a kivonás eredménye negatív, az azt jelentette, hogy a maradék kisebb az osztandónál, tehát a kivonásra nem volt szükség, vissza kellene állítani a maradékot. Erre azonban nincs szükség, a léptetés után ilyenkor a maradékból nem kivonni kell az osztandót, hanem hozzáadni azt. Belátható, hogy ezután ugyanott tartunk, mint a visszaállítási módszernél. (2*maradék-osztandó). Ez az un. **visszaállítás nélküli módszer**.

Az osztás is tört számokra vonatkozik, egész szám értelmezésben AQ-ban az osztandó kétszeresét kell elhelyezni.

A5 és A6 decimális aritmetika

Az eredeti elképzelések szerint a bináris aritmetika mellett párhuzamosan működött volna egy BCD kódolású decimális aritmetika, amely minden bináris aritmetikai műveletnek a decimális párját valósította volna meg.. Ennek a decimális aritmetikának a 2 modulja kapta az A5 ill. A6 nevet. Az első dokumentációk ezt a változatot és az ennek megfelelő decimális utasításokat írták le.

A modulok tervezésekor azonban egyértelművé vált, hogy a kétféle aritmetikában sok a közös rész. Ezért később az a döntés született, hogy egyetlen közös modul legyen a bináris és decimális aritmetikára.

A7 bináris/decimális aritmetika

Bináris kódú számon csak bináris műveletnek van értelme, BCD kódolásún pedig csak decimális műveletnek. Ezért egyetlen utasítást kellett csak megduplázni, az **A-regiszter töltését**. Az ra és rd utasítások egyformán töltötték A-t, de emellett egy jelzőt is állítottak, ami megmutatta, hogy bináris vagy decimális számról van-e szó? Az aritmetikai műveletek ezután ennek a jelzőnek a figyelembe vételével dolgoztak.

Decimális műveletek:

- ra A töltése binárisan
- rd A töltése decimálisan

A típusától függően bináris/decimális:

- aa A-hoz hozzáadás
- as A-ból kivonás
- an $A = \text{operandus} - A$
- am $AQ = \text{operandus} * Q$
- ad $Q = AQ/\text{operandus}$, A=maradék

A túlcsoportolás, decimális/bináris kód, átvitel jelzők egy un. E-regiszterben vannak összefogva.

- re E-regiszter töltése
- we E-regiszter olvasása.

Az így felszabadult műveletei kódok lehetővé tették újabb utasítások bevezetését is. A bájtos kezelés megkönnyítésére készültek olyan utasítások, amelyek lehetővé tették, hogy egy

memóriaszó 3 bájtyának bármelyikét be lehessen tölteni az akkumulátor alsó bájtyába (pr, sr, tr) a másik két bájtt nullázásával, illetve onnan ki lehessen írni egy bájtot a memóriaszó tetszőleges bájtyába (pw, sw, tw), a másik 2 bájtt változatlanul hagyása mellett.

Bájtos műveletek:

- pr tárrekesz alsó bájtyának A-ba írása
- sr tárrekesz középső bájtyának A-ba írása
- tr tárrekesz felső bájtyának A-ba írása
- pw A legalsó bájtyának a tárrekesz alsó bájtyába írása
- sw A legalsó bájtyának a tárrekesz középső bájtyába írása
- tw A legalsó bájtyának a tárrekesz felső bájtyába írása

Decimális számábrázolás

Egy szóban **6 számjegy** helyezkedett el, a legfelső jegy egyúttal az előjelet is tartalmazza: 0-4 között pozitív, 5-9 között negatív számról van szó. Az ábrázolható számtartomány -500000 és +499999 közötti.

A negatív számok ábrázolása a bináriszhoz hasonlóan komplementum képzésével történik, értelemszerűen 1000000 hozzáadásával. Így tekinthetjük akár a számokat előjel nélküli pozitív számoknak is, 0-tól 999999-ig, ugyanaz az összeadás és kivonás alkalmazható rájuk, különbség csak a túlsordulás kezelésében van.

Decimális összeadás és kivonás.

Értelemszerűen alulról felfelé jegyenkénti összeadással, 9 feletti összegnél korrekcióval és átvitelrel valósul meg.

Többszörös szóhosszúságú számok összeadásához alkalmazható az rc utasítás, amely az r-hoz hasonló, de az akkumulátorba az operandus és a carry összegét tölti be.

Decimális szorzás

A bináris szorzáshoz hasonlóan a szorzót a **Q-regiszter**, a szorzandót az operandus tartalmazza, az eredmény **AQ**-ban képződik, a szorzás végrehajtása Q alsó jegyével való szorzással és lefelé léptetéssel történik. Egy decimális jeggyel való szorzás 1-4 között ismételt összeadásokkal, 5-9 között ismételt kivonásokkal és a léptetés utáni korrekcióval (1 hozzáadása) valósul meg.

Decimális osztás

A bináris osztáshoz hasonlóan az osztandó AQ-ban helyezkedik el, az osztó az operandus, a hányados Q-ba, a maradék A-ba kerül, az osztás pedig felfelé léptetéssel és az osztóval való összehasonlítással történik. Az összehasonlítás ismételt összeadásokkal vagy kivonásokkal megy végbe. A művelet iránya olyan, hogy AQ-t zérus felé viszi. Az összegzések A előjelváltásáig folytatódnak, a hányadosjegyet a kivonások száma -1, vagy 10 mínusz az összeadások száma adja. Előjelváltás után egy digittel való felfelé léptetés következik, Q alul beveszi a hányadosjegyet.

Az A7 **8 lemezen** készült el, kb. 1200 mikrokárdot tartalmazott, szemben az A3 és A4 együttesen kb. 1000 mikrokárdjával. Külön decimális modulok esetében is közel 1000 mikrokárddal kellett volna számolni. Az összevonás tehát gazdaságos volt. Bizonyos árat azért kellett fizetni ezért: a nagyobb bonyolultság miatt hosszabb lett a tervezési idő, mint csak decimális modulok esetében lett volna. 8 lemeznél már jelentkeztek a lemezek közötti belső sínzés korlátai is,



Az A7 aritmetika a szegedi Informatika Történeti Múzeumban.

Mintha most került volna le a szerelőszalagról ...
- hála Muszka Dánielnek és Bohus Mihálynak



Az A-regiszter (akkumulátor)

IC-s aritmetika

A teljesség kedvéért megemlítjük, hogy ugyanez az aritmetika elkészült 1972-74 között integrált áramkörös változatban, az EMG 840.ben. Az aritmetika teljesen kompatibilis maradt a 830-éval, az algoritmusok sem változtak. A műveletek már teljes szóhosszon párhuzamosan hajtották végre. Az összeadó egységhez ekkor már MSI áramköröket lehetett használni, 8 tokból párezer forintért elkészült az az egység, amelynek harmada korábban egy teljes lemezt (kb. 160 mikrokard) foglalt el. Az átviteli lánc hossza sem jelentett problémát a carry-gyorsítók alkalmazásával.

Részvétel az aritmetika fejlesztésében

Az aritmetikai egység első terveit **Sárossy József** készítette. Én 1966 szeptemberében léptem be pályakezdőként az EMG-be, és megkaptam ezeket a terveket kivitelezésre.

Megemlítem, hogy teljesen véletlenül tudtam meg, hogy az EMG-ben számítógép fejlesztése folyik. A Budapesti Műszaki Egyetemen kaptunk képzést logikai áramkörös tervezésből, némileg a számítógép alapjairól is, de úgy tudtuk, hogy Magyarországon nincs és nem is várható számítógép-fejlesztés. Túl nagy feladat lenne és gyártásuk nem is lenne gazdaságos, azt a néhány gépet, ami az ország számítási igényeit kielégíti, jobb külföldről behozni.

Sárossy névvel már korábban találkoztam. A BME Lineáris Hálózatok Tanszékén Géher Károly docenstől kaptam diplomatervet. Két témát ajánlott fel. Elsőként egy Toleranciaszámítás című diplomamunkát mutatott, amely két évvel korábban a diplomatervpályázaton első díjat nyert, ezt kellett volna folytatnom. Szerzője – mint mondta – egy igen tehetséges hallgató, Sárossy József volt. Én akkor a másik témát választottam. Sorsomat

azonban nem kerültem el. Fél évvel később az EMG-ben Sárossy aritmetikáját kellett folytatnom. Nagy szerencsémre, teszem hozzá.

Mikor beléptem, Sárossy éppen a négyértékű logikáról tartott előadást egy konferencián Balatonszéplakon. Megérkezéséig is igyekeztem tanulni. Münnich Tóni megkérdezte tőlem, hogy tudok-e algoritmus? De furcsa kiejtése van, gondoltam magamban, aztán válaszoltam, hogy a suliban németet tanultam, de fakultative angolt is. Pár nap múlva pedig megkérdeztem, hogy mit jelent a hardver meg a szoftver, mert olyan sokat emlegetik...

Sárossy valóban igen tehetséges ember volt, emellett lenyűgözően sokoldalú általános és szakmai műveltséggel rendelkezett. A munkanapot német és angol nyelvű szakkönyvek olvasásával kezdte, kedvencei az automata-elméleti könyvek voltak, ezek legjobbait aztán ajánlotta, sőt kölcsön is adta a többieknek. Tudósok számára ez természetesnek tűnhet, de gyári környezetben, reggel 7 órás blokkolás és a rengeteg gyakorlati teendő mellett nem volt az. Aztán nemsokára átvette az áramköri csoport vezetését, ahol csoportjával rövid időn belül kifejlesztette a szilícium alapú áramköri sorozatot a nem kellő megbízhatóságú germánium-tranzisztorok kiváltására. Megemlítem még, hogy szinte verhetetlen volt a kaszinó nevű kártyajátékban...

A tervek szerint az A1 modul 6, az A2 2 lemezen helyezkedett el. A modulok működését Sárossy Algol nyelven is leírta.

Az A1 **kivitelezése egy évig tartott**. A lemezek logikai tervei alapján először a nyomtatott áramköri terveket kellett elkészíteni, az ennek alapján elkészített kliséket ellenőrizni, majd a szerelt lemezeket bemérni. 1967 augusztusára elkészült a 6 lemez, azokat egy modulba szerelték, és elkezdtem a modult a számítógépben gépi utasításokkal tesztelni. Amikor Klatsmányi bejött a számítógép-szobába, Köves Péter csoportvezetőm büszkén mutatta az éledező aritmetikát. A „tanár úr” erre megkérdezte, hogyan működik a szorzás és az osztás vezérlése? Teljesen az A2 végzi el azt, vagy jut belőle az A1-be is? A tervezésnél néhány funkciót kézenfekvő volt az A1-ben realizálni - válaszoltam. Klatsmányi erre azt mondta, hogy ez így nem jó, így nem lehet az A1-ből prototípust csinálni! A szorzás és osztás tesztelése nélkül nincs garancia arra, hogy később nem kell az A1-et is módosítani. A modult újra kell tervezni! Az A1 csak 2 vezérlő jelet figyeljen, amelyek megmondják, hogy összeadást, kivonást, jobbra vagy balra léptetést kell-e végezni.

Elhűlve hallottam a gyors döntést! Eddig mindig sürgették a prototípust, és most kezdjük előlről?

Az **áttervezés** feladatát én kaptam meg. A szorzás és osztás vezérlésében résztvevő logika teljesen az A2-be helyezése mellett volt egy más elképzelésem is. A bemérés során sokat vizsgáltam a paritásbit ellenőrző szerepét, és támadt egy ötletem az ellenőrzés hatékonyságának a növelésére, amely egyúttal a paritásbit képzését is egyszerűsítene. Kissé szorongva és táblázatokkal felszerelve mentem Sárossyhoz, hogy én akkor a paritásképzést is áttervezném, mit szól hozzá? Aggodalmam alaptalan volt. Ő azonnal átlátta a dolgot, és helyeselte a megoldást.

Az áttervezés és az újabb kivitelezés már sokkal gyorsabban ment, mint először, nem csúszott emiatt a prototípus elkészülése. Az aritmetika áttekinthetőbbé és könnyebben kezelhetővé vált, kisebb is lett. Az idő Klatsmányit igazolta. Sárossy emberi kvalitását pedig jellemzi, hogy innen kezdve ő is engem nevezett az Aritmetika tervezőjének, és sosem említette, hogy

az alapokat azért ő rakta le. Pedig mennyivel könnyebb volt az én munkám az ő vállaira állva, mintha nekem kellett volna mindent előlről kezdenem!

Klatsmányi döntési képességével később is szembesültem, amikor eléje került a javaslatom, hogy ne legyen külön decimális aritmetika, hanem inkább egy közös bináris-decimális modul. Megint percekben belül döntött: A5 és A6 visszavonva, legyen egy közös aritmetika A7 néven! Köves Péter, aki akkor már tanácsadóként dolgozott mellette, lelkiismeretesen felsorolta, hogy milyen dokumentumokat érint ez a változtatás, meg már tasakok is nyíltak, amelyekre anyagrendelések is elmentek. (A beszerzések hosszú átfutási ideje miatt az anyagokat sokszor már a logikai tervezés előtt meg kellett rendelni...) Most is magam előtt látom, ahogy Klatsmányi széles mozdulatot tesz a karjával, ezzel mintegy le is seperve ezeket a problémákat az asztalról.

Az EMG 830 tervezői igazi úttörő munkát végeztek. Nem lehetett sok mindent előre látni, nem lehetett mindig elsőre eltalálni a legjobb megoldást. Hajtotta viszont őket a **szakmai maximalizmus**, mindig a legjobb megoldást keresték, és a lehetőségek szerint meg is valósították azt, nem törődve a velejáró többletmunkával. Ebben a mentalitásban is meghatározó szerepet játszott Klatsmányi Árpád. Talán ez a legfőbb útravaló, amit magammal vittem az EMG-ből.