

Álló Géza: Az Elliott 803B számítógép

A Neumann-elvű Elliott 803B szószervezésű, soros működésű számítógép *központi egységeit* 4 szekrényben helyezték el:

- 1. szekrény: központi vezérmű (utasításregiszter, címaritmetika, műveletek végrehajtó áramkörei, periférikus eszközök csatoló és vezérlő áramkörei).
- 2. szekrény: fixpontos és lebegőpontos számológép (akkumulátor- és q-regiszter, kitevő-aritmetika, műveleti és léptető áramkörök);
- 3. és 4. szekrény: egy-egy 4 Kszó méretű főtárblokk (39 bit/szó + 1 paritásbit, főtár-regiszter, író/olvasó áramkörök);¹
- Vezérlő konzol egy külön asztalon.²

A periféria egységei

- mágnesszalag vezérmű szekrény, (szalag-regiszter, szalagmozgatás vezérlés, olvasó / író áramkörök),
- 2 mágnesszalag-meghajtó szekrény (író / olvasó fejek, szalagmozgató áramkörök; a meghajtókat programból lehetett kiválasztani, de számozásukat kézzel kellett beállítani.);
- lyukszalag-asztal: két-két 5/8 sávú Olivetti lyukszalag olvasó, illetve Teletype lyukszalag lyukasztó (az aktuális egységet kézzel lehetett kijelölni);
- sornyomtató.

A programok, illetve az adatok bevitelére szolgáló lyukszalagokat off-line módon, Creed illetve Siemens telexgépeken lyukasztották az operátor-lányok, emiatt csak 5 sávú lyukszalagokat használtak.³ Az első években a gép az eredményeket is lyukszalagra ta, amit az említett hölgyek ugyan jobban olvastak, mint Kyklops a nyáját, de a „képzetlen” programozók kedvéért azért telexgépeken papírra is kiírtatták.



A vezérlőpult...

A sornyomtató 1965-ben érkezett, azzal a kis szépséghibával, hogy a betűhengeren az

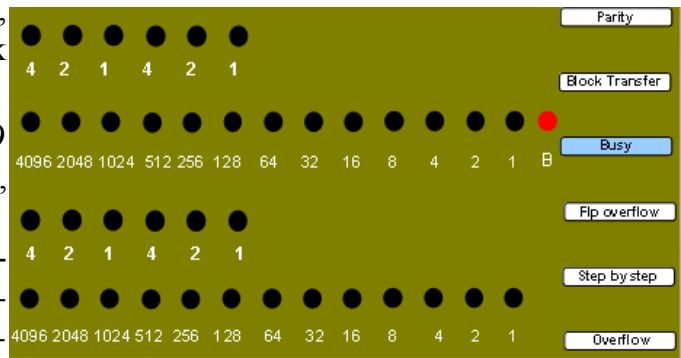
1 A paritásbit az adott gépi szó 1-bitjeinek számát páratlanra egészítette ki. Az ellenőrző bitet a gép adat beírásakor automatikusan generálta, kiolvasáskor pedig ellenőrizte; a felhasználók csak akkor szereztek tudomást létezéséről, amikor paritáshiba hiba miatt a gép „Parity” hibajelzéssel leállt, és minden addigi nem mentett futtatási eredmény elveszett; amitől a programozók enyhe agyvérzést kaptak, és jöhettek a műszakiak.

2 Az elektronika működéséhez szükséges egyenfeszültséget egy 12 V-os csepptöltésű akkumulátor szolgáltatta, amely a váltóáramú hálózati feszültség kimaradása esetén körülbelül 1 órányi folyamatos üzemidőt biztosított. A töltőfeszültséget egy toroid-transzformátorra kötött stabilizátor egység tartotta $\pm 3\%$ tűréshatáron belül.

3 Érdekességképpen: egy házi versenyen a csúcstartó 113 karakter/perc leütéssel, mindössze 2 hibával győzött!

írásjeleken kívül csak az angol ábécé nagybetűi szerepeltek. Mivel ezután már a megkötenő szerződések űrlapja is sornymatón készült, ügyelni kellett, hogy ne tévesszük össze a török báját a torokbajjal.⁴

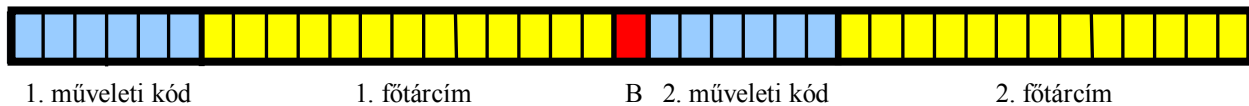
A vezérlőpult gombsorai leképezték a 39 bites gépi szót, amelyben vagy két gépi utasítás, vagy egy 39 bites adat lehetett. A gépi utasítás műveleti kódját két 3-bites oktális számként, a főtárcímet decimálisan értelmezett 13 bites bináris számként kellett „benyomkodni”.



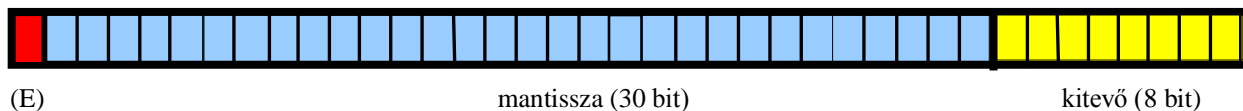
... és égi mása.

A gépi kód

A számítógép 6 μ s-os órajelekkel működött, és a 19 bites gépi utasításokat általában két, egyenként 48 ciklusból álló szóidő (288 μ s) alatt dolgozta fel: az elsőben a 6 bites műveleti kód alapján előkészítette a végrehajtást, a másodikban az akkumulátor-regiszterben tárolt egyik (egytényezős műveletek esetében egyetlen) és a 13 bites címről kiolvasott másik tényezővel együtt elvégezte a műveletet. Az eredmény az akkumulátorban keletkezett, ezt szükség esetén külön utasítással kellett visszaírni a megfelelő főtár-rekeszbe.



A konstruktőrök zseniális újítása volt az ábrán piros színnel jelzett B-bit: ennek 1-értéke esetén a vezérmű hozzáadta az aktuális első utasítást a második szófélben levőhöz, és az összegnek megfelelő utasítást hajtotta végre. A gyakorlatban ilyenkor az első utasítás 0 („do nothing”) szokott lenni, így a második utasításnak csak a címrésze módosult: ez volt az indexelés őse!



A számológép 39 bites fixpontos és lebegőpontos, 2-komplementes ábrázolású bináris számokkal tudott műveleteket végezni. Az előjelet mindenkor a legnagyobb helyiértékű (bal szélső) bit 1 értéke jelképezte, ez után állt a szimbolikus bináris pont, majd az adatbitek következtek, fixpontos törteként $2^{-1} \div 2^{-38}$, lebegőpontos ábrázolásban pedig a mantissza bitek $2^{-1} \div 2^{-30}$ helyiértékig, az utóbbiakhoz csatlakozott a 8 bites, fixpontos kitevő. (Ebben a rendszerben a 0 értékű adat pozitívnak számított, vagyis előjele is 0 volt.)

A fixpontos műveletekben a bináris pont helyzete nem változott, vagyis az ábrázolható adattartomány $-1 \div 1-2^{-38}$ között helyezkedett el; egész számokkal úgy lehetett számolni, hogy az eredményeket kiírás előtt meg kellett szorozni 2^{38} (=274 877 906 944) -nal.

A lebegőpontos műveletekben a bináris pont „csúszkált”, mert a mantissza automatikusan megszorozódott 2-nek a kitevőrész által meghatározott hatványával. A kitevőt a gép előjeles egész számként kezelte, értéke $-128 \div 127$ lehetett, így az ábrázolható lebegőpontos számtartomány $-2^{128} \div 2^{127} - 2^{97}$ ($\sim -2,9 \cdot 10^{-39} \div 1,7 \cdot 10^{38}$) közé esett. Ennél kisebb eredményt a számológép automatikusan nullázott, nagyobb eredmény termelésekor viszont a vezérlőpulton a „Flip overflow” jelzőlámpa kellemetlen fénye jelezte, hogy a regiszter rövidnek bizonyult a tárolására, és a gép

4 A későbbiekben lecserélt betűhenger már a teljes magyar abc nagybetűit tartalmazta.

leállt. (Volt még egy kis lebegőpontos trükk is: ha az eredmény-mantissza 0 lett, a gép a kitevőt is automatikusan nullázta, így a „lebegőpontos 0” ugyanúgy nézett ki, mint a fixpontos.)

Míg az ugró utasításokat a gép 1 szóidő alatt végrehajtotta, a fixpontos „rövid” műveletekhez 2, a lebegőpontos műveletekhez 3, az úgynevezett „hosszú” műveletekhez (szorzás, osztás, léptetések) pedig legfeljebb 40 szóidőre volt szükség; vagyis a gép műveleti sebessége másodpercenként 1700 fixpontos, illetve 1130 lebegőpontos összeadás, de csak 89 fixpontos osztás volt.

A lyukszalag-olvasót, illetve a lyukasztót a gép igen egyszerűen kezelte:

- a beolvasó utasítás (71 0) hatására az akkumulátor 5 legkisebb helyiértékű bitjére beírta az aktuális (1 / 2) olvasóban levő lyukszalagról a soron következő karakternek megfelelő ötbités értéket (lyukhely \rightarrow 0, lyuk \rightarrow 1);
- a kiíró utasítás (74 0) hatására pedig kilyukasztotta az aktuális (1 / 2) lyukszalag-lyukasztón az akkumulátor 5 legkisebb helyiértékű helyén álló számnak megfelelő karaktert.

Mivel 5 bittel csak 31-féle karakter ábrázolható (az „üres”=00000 karaktert az olvasók egyszerűen átugrották), így az angol ábécé 26 betűje, a számjegyek és az egyéb írásjelek ábrázolása végett – a szóköz, a kocsivissza és a soremelés billentyűn kívül – még szükség volt egy „számváltó” illetve „betűváltó” billentyűre is; az utóbbi kettő kódja jelezte, hogy az utánuk következő karaktereket numerikusan vagy szóvegesen kell-e értelmezni.

Az egyszerű kezelés ára az volt, hogy mindkét irányú karakter-átvitel lefoglalta a központi vezérművet, amit a „Busy” jelzőlámpa fénye jelzett a vezérlőpulton, ez addig világított, amíg a megkezdett átvitel be nem fejeződött. (Például, ha az olvasóban vagy a lyukasztóban nem volt lyukszalag, a pótlásáig.)

A „BIOS”

A natúr gép önmagában jószerivel használhatatlan, a programokat és az adatokat karakterenként beolvasni lyukszalagról, illetve az eredményeket lebontani karakterekre és kiírni lyukszalagra kézi műveletekkel (gyakorlatilag) lehetetlen lett volna, ehhez megfelelő programokra volt szükség, amelyeket előzőleg természetesen valamilyen módon be kellett tölteni a főtárba. A kezdeti feltöltési folyamatot („inicializálás”) manapság a beégetett BIOS programok intézik, a korabeli „BIOS”-t viszont egy szellemes konstrukció pótolta: a főtár 0-3 című első négy tárrekesze helyén egy fixen behuzalozott („beégetett”) behúzó-program („bootstrap”) kapott helyet, amellyel egy bináris lyukszalagot lehetett beolvasni a főtár végére; az így betöltött programocskával aztán bármilyen másik programot el lehetett indítani.

Érdeemes egy kis időt szánni rá, olyan ötletes! (Mindен gépi szó két utasításból és a közepén elhelyezkedő B-bitből áll; az utasítások első két oktális számjegye a műveleti kód, a harmadik a decimális főtárcím.). Íme:

Főtárcím U1 B U2

0: 26 4 0 06 0

- 1) A 4 című tárrekesz törlése; eredmény: $\langle 4 \rangle = 0$;
- 2) Akkumulátor-regiszter törlése; eredmény: $\langle A \rangle = 0$.

1: 22 4 1 16 3

- 1) Felszámlálás a 4 című tárrekeszben: $\langle 4 \rangle = \langle 4 \rangle + 1 = 1$;
- 2) $\langle A \rangle$ tárolása a $\langle 4 \rangle$ -gyel megnövelt címen és az akkumulátor törlése (16 4); eredmény: $\langle 4 \rangle = \langle A \rangle = 0$.

2: 71 0 0 55 5

- 1) Egy karakter beolvasása az 1. lyukszalag-olvasóról az akkumulátor 5 legkisebb helyiértékű bitjére;
- 2) $\langle A \rangle$ léptetése balra 5 hellyel.

3: 43 1 0 40 2

- 1) Feltételes ugrás az adott (itt 1) címre túlsordulás esetén, a túlsordulás jelölő törlése;
- 2) Feltétlen ugrás az adott (itt 2) címre.

És most lássuk, mi történt ha az operátor beállította a vezérlőpulton 40 0 utasítást (vagyis benyomta a legelső gom-

bot) és lenyomta az „Obey” (engedelmeskedj) parancsgombot:

A vezérlés a 0 címre kerül, és végrehajtódik a „beégetett” program első 4 utasítása.

Első kis ciklus: a gép beolvassa a lyukszalagról a következő karaktert és 5 hellyel balra lépteti, beolvas, balra léptet sít., visszaugrik az amíg túlsordulás nem lép fel; ez a 8. karakter beolvasása után következik be, amikor az első beolvasott karakter (kötelezően 1 értékű) első bitje „kicsordult” az akkumulátorból. A példa kedvéért tegyük fel, hogy ekkor az akkumulátorban 8156 van (10000 00000 00000 00000 00001 1111 1101 1100), vagyis betöltendő programunkat a 8160 címtől kezdve akarjuk elhelyezni.

Első nagy ciklus: felszámlálás a 4-es tárrekeszben, majd a B-bit hatására ennek tartalma hozzáadódik a második utasításhoz és a gép végrehajtja a 16 4 utasítást, vagyis beírja a az akkumulátor tartalmát a 4 címre:
<4> = 8156, egyszersmind nullázza az akkumulátort <A> = 0;

Ezután felváltva 31 kis és nagy ciklus következik:

minden kis ciklusban betöltődik az akkumulátorba leendő beolvasó programunk soron következő utasításpárja, amely – a 4-es tárrekesz felszámlálása (8157, 8158...) és a második szófélben levő utasítás címrészének módosítása révén – a 16 8160, 16 8161, ..., 16 8191 utasítással rendre tárolódik a főtár megfelelő címén.

Négy üres utasítás (40 0 0 00 0) beolvasása és formális tárolása a 0 – 3 tárrekeszben.

(Mivel az utasítás címrésze 13 bites, $8191+1=0!$ Az első 4 tárrekesz valójában nem létezik, a behúzó programot pedig nem lehet felülírni!)

Utolsó előtti kis ciklus: utasításmódosító és a betöltött program <kezdőcím>-4 (esetünkben 8156) beolvasása az akkumulátorba; <A> = 22 8156

Utolsó előtti nagy ciklus: <A> beírása a 4-es tárrekeszbe; <4> = 22 8156;

Utolsó kis ciklus: 1 üres utasítás beolvasása, <A> = 0;

Utolsó nagy ciklus: felszámlálás a 4-es tárrekeszben és a $16\ 3 + 22\ 8157 = 40\ 8160$ utasítás végrehajtása, vagyis a vezérlés átadása a beolvasott program kezdőcímére, majd ennek végrehajtása.
(Ne feledjük, hogy az utasításkód oktális számrendszerben értendő, vagyis $16 + 22 = 40!$)

A sornyomtató és a mágnesszalagok kezelése

A másik két külső egységet nem lehetett olyan egyszerűen használni, mint a lyukszalagosakat, mert ezek nem karakterekkel, hanem adatblokkokkal dolgoztak. Ezért működtetésük két egymás utáni utasítással történt:

- a 76 <kód> utasításban a sornyomtató, illetve a mágnesszalag kódjának megadásával kellett inicializálni a megfelelő vezérlőegységet;
- a 77 <főtárcím> utasítás pedig a kiírandó, illetve a beolvasandó adatblokk főtárbeli kezdőcímét határozta meg.

A sornyomtató egy utasításra egy sort (128 karaktert) nyomtatott ki. Mivel a karaktereket 6 biten kellett megadni –, jelezve, hogy az 5 bites kód számként vagy betűként nyomtatható-e –, egy gépi szóba csak 6 karaktert fért, ezért a nyomtató a megadott címtől még 21 gépi szó tartalmát nyomtatta ki soronként.

A mágnesszalag-vezérmű egy utasításra egy adatblokkot írt fel, illetve olvasott vissza a kijelölt meghajtón levő szalagtekercsre/-ről. Egy blokk 64 gépi szóból állt, és egy szalagtekercsre 256 blokk, vagyis összesen 16 Kszó (16384 gépi szó) fért rá. A szalagblokkok címezhetők voltak, a kívánt című szalagblokk pozicionálásra külön utasításpár szolgált. A beolvasásnál ügyelni kellett, hogy legyen elegendő szabad tárrekesz a megadott cím után, különben felülíródhettek értékes adatok, netán programsorok.

A 76 kódú műveletet 77 kódúnak kellett követnie és fordítva: a 77 kódú előtt 76 kódúnak kellett állnia, ellenkező esetben a gép „Block transfer” hibajelzéssel leállt. Ha az adatátvitel valamilyen okból elakadt (például kifogyott a papír a nyomtatóból vagy nem létező szalagcímre pozicionáltunk), a gép ugyancsak leállt, ezt a „Busy” lámpa fénye jelezte.

Programozás

A gépet gépi kódban, illetve Autokód vagy ALGOL (!) nyelven lehetett programozni.

Az Autokód tulajdonképpen egy emberközeli gépi kód nyelv volt, vagyis a programot gépi utasításonként kellett megírni, lényeges könnyítésekkel:

- a decimális címkódok helyén (előre deklarált) alfanumerikus változónevek állhattak;
- a 4 alapl műveletet a matematikában szokásos műveleti jelekkel lehetett megadni, további egyszerűbb műveletek megadására az utasítás angol nevéből képzett „mnemonikok” szolgáltak;
- összetettebb műveletek végzésére a géppel együtt szállított szubrutinkönyvtár szolgált.

(Miniatúr mintapélda:

integer I, J, K	3 egész típusú változó deklarálása;
read I	olvassd be az 1. lyukszalag olvasóról a soron következő (decimálisan írt) egész számot, váltsd át binárisra és tárold az I változóban;
read J	ugyanígy a J változóban;
K = I+J	az I és a J változó összegét tárold a K változóban;
punch I K	írd ki decimális formában a K változó tartalmát az 1. lyukszalag lyukasztón.)

A későbbiekben a számítóközpontban még számos újabb szubrutint is kifejlesztettek, az előforduló bonyolultabb feladatok megoldására.

A Graphomat

A Számítóközpont 1966-ban kapott egy ZUSE gyártmányú, lyukszalag-vezérlésű rajzológépet, amelynek rajztolla – a 8 irányt ismerő korabeli plotterektől eltérően – az A2 méretű rajztábla 1/16 mm (0,0625 mm) osztású (elméleti) rácshálójának bármely pontját el tudta érni. A gép „tolltartójába” egyszerre 4 toll fért, s mivel bármelyik lehetett rajzolópozícióban, és – a rajzolás ideiglenes felfüggesztése mellett – menet közben bármelyik tollat ki lehetett cserélni a rendelkezésre álló készlet bármelyik elemével, egy rajzot 4-féle színnel (fekete, piros, zöld, kék) és 6-féle vonalvastagsággal (0.2, 0.5, 1, 1.5, 2, 3 mm) lehetett megrajzolni.

A rajzológép tulajdonképpen két pontot tudott egyenessel összekötni, 1/16 mm-es lépésekben, vagyis vonalas rajzok készítésére volt alkalmas. Egy rajzolóutasítás 3 karakterből állt, és azt határozta meg, hogy a gép egyidejűleg hányat lépjen x és y irányban, illetve ezt a műveletet hányszor (n) ismételje. (A lehetséges értékkészletek: $-15 \leq x, y \leq 15$, illetve $1 \leq n \leq 15$.) További utasítások szolgáltak az aktuális rajztoll kijelölésére (tollváltáskor automatikusan ez állt rá az aktuális pontra), az aktuális toll lesüllyesztésére (rajzoló pozícióba), illetve felemelésére, valamint a rajzolás felfüggesztésére, illetve folytatására.

A Graphomat vezérlésére Álló Géza dolgozta ki a számítógépre a Grafokód nevű programnyelvet, amely egy rajz leírása alapján elkészítette a megrajzolásához szükséges vezérlő lyukszalagot. A nyelv fontosabb utasításai voltak: két tetszőleges pont összekötése különböző típusú (folytonos, szaggatott, pontozott stb.) egyenes vonallal, adott középpont körül tetszőleges sugarú és -középponti szögű körív rajzolása, görbe vonal illesztése adott ponthalmazra, valamint tetszőleges írásirányú, dőlésszögű és betűméretű szöveg kirajzolása, illetve automatikus beillesztése tetszőleges paralelogrammába, a teljes magyar és görög ábécé kis és nagybetűivel. Lehetőség volt továbbá ciklusok és szubrutinok készítésére is.