

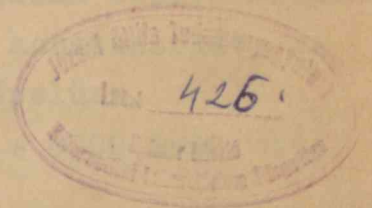
BEINDITÓPROGRAMOK  
AZ M-3 GÉPHEZ

FIDRICH ILONA

szakdolgozata alkalmazott matematikából  
az

1958/59.  
tanévben.

Dr. Kalmár László  
Magyarika



SZTE Klebelsberg Könyvtár



J001102444

## BEINDITÓPROGRAMOK AZ M-3 GÉPHEZ.

A beindítóprogramok hatására a gép leolvassa a szalagon lévő utasításokat, és megfelelően átcimezve elhelyezi a memóriába. Közben végrehajtja a vezérlőkombinációk által előírt műveleteket.

### I. változat:

A beindítóprogram 61 utasításból áll. Az utasításokat a  $t + 10$ -tól  $t + 104$ -ig terjedő memóriapozíciókba helyezzük el  $t$  rögzítése után abszolút címekkel.  $t$  jelenti annak a memóriacellának a címét, ahonnan kezdve a paraméterek találhatóak.

A program felhasznál 9 szabványkonstanst. Ezek helye még nem véglegesen tisztázott, címeiket ideiglenesen görög kisbetűk jelölik. (A szabványkonstansok helyének rögzítése után természetesen be kell írni a görög betűk helyére a pontos címeket.) Szerepel még 4 speciális konstans, amelyek a program utasításai után találhatóak. Használunk ezeken kívül az utasítások végrehajtása során módosuló konstansokat, amelyeket szintén a beindítóprogram után gépelünk.

Feltételezzük még, hogy a 0 helyen a + 0000 0000 utasítás van.

A konstansok elhelyezésének sorrendjénél lényegesen kihasználja a beindítóprogram, hogy  $\langle 2^{-19} \rangle = \langle 2^{-7} \rangle + 1$  és  $\langle 7777 \cdot 2^{-30} \rangle = \langle 7777 \cdot 2^{-18} \rangle + 1$ .

Két memóriacellát munkarekeszként használunk: a  $t + 115$ . címre helyeződik a módosítószó, és helyes működés esetén végrehajtásra nem kerül. A  $t + 116$ . memóriarekeszbe a változatlan elhelyezés jelzése kerül. A beindítóprogram behívása előtt gondoskodni kell ennek a cellának a kiürítéséről. Az utasításokat a gép a  $t + 100$ . pozícióba viszi be, ahol egyes vezérlőkombinációk végrehajtásra is kerülnek.



Az EDSAC-gép 22. utasításának szerepét, a memóriába való átküldést, ennél a beindítóprogramnál a  $t + 56$ . utasítás vette át, mely minden utasítást a  $t + 100$ . címről helyez a memória soronkövetkező rekeszébe.

### Átcimzés:

A gép csak akkor működik helyesen, ha a beindítórutin hatására elsőként leolvasott szó módosítószó.

Egy módosítószó a következő alakú:

$$\pm r_{11}r_{12} \dots r_{51}r_{52},$$

ahol  $r_{ij}$  ( $i = 1, \dots, 5; j = 1, 2$ ) egy oktális számjegy.  $t + r_{ij}$  jelenti azon memóriarekesz címét, ahol a módosítószó után  $i$ -ediként következő, módosítással elhelyezendő utasítás  $j$ -edik címét módosító paraméter van.

A gép módosítással elhelyezendőnek (röviden módosítandónak) tekint minden olyan pozitív szót, amelyre nem vonatkozik az előtte lévő 1. vezérlőkombináció.

$r_{ij}$  a memóriában 3 bit helyet foglal el, tehát egy módosítószó egy egész memóriarekeszt kitölt.

3 bit viszont csak 8 féle információt tárolhat, így a relatív címzésnek 8 féle alapja lehetne.

A beindítóprogram I. változatában a relatív címzés alapja 16 féle lehet. Ez ugy érhető el, hogy egy memóriarekeszben két paramétert tárolunk:  $2^{-18}$  és  $2^{-30}$  nagyságrendben. Hogy a  $t + r_{ij}$ -edik címen lévő két paraméterből álló utasítás első vagy második címrésszel kell -e módosítani az  $i$ -edik utasítás  $j$ -edik címrésszét, azt az  $i$ -edik utasítás  $j$ -edik címrésszének első bitjén lévő bináris jegy dönti el a következőképpen:

Első cím rész módosításánál:

Ha az  $i$ -edik utasítás előlről számított 7. (tehát  $2^{-7}$ -nek megfelelő helyiértékű) bináris jegye 0, akkor a  $t + r_{11}$ . memóriarekesz tartalmának első cím részén található paraméterrel módosítva kerül az  $i$ -edik utasítás vissza a  $t + 100$ . rekeszbe.

Ha a 7. bináris jegy 1, akkor a második cím részen elhelyezett paraméterrel módosul az első cím rész.



Második címresz módosítása:

Ha előlről számítva a 19. (tehát a  $2^{-19}$ -nek megfelelő helyiértékű) bináris jegy 0, akkor a második, ha 1, akkor az első címreszen lévő paraméterrel módosul a második címresz, és így kerül a memória soronkövetkező rekeszébe.

Összefoglalva: Ha a helyes címreszmódosításhoz az elhelyezett paraméter nagyságrendjének megváltoztatása szükséges, akkor a címreszt meg kell növelni 4000-rel.

A gép helyesen működik akkor is, ha a módosítószó után, még az öt módosítandó utasítás előtt, vagy közben vezérlőkombinációk is vannak, a lényeges csak az, hogy az 5. módosított utasítás után rögtön a következő 5 módosítandó utasítás módosító szava kerüljön leolvasásra.

Ha a módosító szó után következő  $i$ . módosítandó utasítás  $j$ . címresztét nem kell módosítani,  $r_{ij} = 0$ , és mivel a  $t$ . memóriarekesz első címresztén van a 0, ezért  $j = 1$  esetén az  $i$ . utasítás  $2^{-7}$  helyértékén 0-t kell írni (azaz az első címreszt változatlanul kell hagyni),  $j = 2$  esetén pedig az  $i$ . utasítás  $2^{-19}$  helyértékére 1-et kell gépelni (azaz a második címreszt növelni kell 4000-rel).

A címmódosítás menete:

A módosítószót a  $t + 10$ . utasítás hatására leolvassa a gép és elhelyezi a  $t + 115$ . címre.

A  $t + 11$ ,  $t + 12$ . utasítások beállítják a végrehajtás során módosuló konstansok kezdeti értékeit, a  $t + 13$ . pedig a  $t + 25$ . utasítás kezdeti értékét.

A  $t + 14$ . utasítás hatására a következő utasítás a szalagról bekerül a  $t + 100$ . címre.

Ha nem változatlanul kell elhelyezni (azaz nem esik 1. vezérlőkombináció hatása alá) és nem vezérlőkombináció, akkor a  $t + 24$ . utasítással elkezdődik az átcimzés. Leválasztja a gép a módosítószó első 3 bitjét, amelyik az első címreszt módosító paraméter címét adja meg. A  $t + 27$ ,  $t + 28$ . utasításokkal megnézi a gép a  $t + 100$ . címen lévő, átcimzésre váró utasítás 7. bitjét, amely a paraméter címresztét adja meg az előzőleg leirt módon.

Az első cimrész módosítása után  $t + 36$ . utasítás visszahelyezi a módosítandó utasítást a  $t + 100$ . címre.

A végrehajtott változó utasítások és a végrehajtás során módosuló konstansok megfelelő változtatása után a gép hasonlóan végrehajtja a második cimrész módosítását, s az így átalakított utasítást elhelyezi a memória soronkövetkező rekeszébe. Ezután az előbb végrehajtott változó utasítások ismét módosulnak, s folytatódik a működés a következő utasítás leolvasásával a  $t + 14$ . utasítás hatására mindaddig, míg a módosító szó utolsó 3 bitjét is fel nem használta a gép. Ekkor viszont a  $t + 10$ . utasításnak adódik át a vezérlés, tehát a helyes működéshez szükséges, hogy az 5. módosított utasítás után a következő 5 módosítandó utasítás módosítószava következék.

#### Vezérlőkombinációk:

Az I. változatnál a vezérlőkombinációk negatív utasítások. A vezérlőkombinációk nem helyeződnek el a memóriába, tehát a beindítóprogram használata esetén negatív szót nem lehetne bevinni.

Főleg ennek a megszorításnak a kiküszöbölésére szolgál az 1. vezérlőkombináció:

—— 00 <a\*\*>  $t + 116$ .

Ez az utasítás a leolvasás során bekerül a  $t + 100$ . memóriarekeszbe, s mivel műveleti kódja 00, a  $t + 71$ . címen végrehajtásra is kerül. Hatására az  $a^{**}$  ( $= a \cdot 2^{-30}$ ) elhelyeződik a  $t + 116$ . címre, mivel ott előzőleg 0 volt. A következő szavak leolvasásuk után a  $t + 15$ ,  $t + 16$  utasítások hatására mindaddig változatlanul helyeződnek el, míg a  $t + 116$ . cella ki nem ürül, miközben mindig  $1^{**}$  ( $= 2^{-30}$ ) levonódik a tartalmából. Az 1. vezérlőkombináció hatására tehát "a" számú utasítást változatlanul helyez el a gép a memória soronkövetkező rekeszeibe. Így lehetővé válik a negatív szavak bevitele is. Az 1. vezérlőkombináció alkalmazása célszerű akkor is, ha legalább 5 utasi-



tást módosítás nélkül kell elhelyezni. Időtakarékosság szempontjából még kevesebb változatlanul elhelyezendő utasítás elé is jobb ezt gépelni, mint a 00...0 módosítószót.

A 2. vezérlőkombináció alakja:

— 05 t + 100  $\beta$

A beindítórutin végrehajtása során ez az utasítás megváltoztatott előjellel a t + 74. utasítás hatására átkerül a t + 56. címre. Hatása tehát nyilván az, hogy a gép az elhelyezést nem a következő, hanem a  $\beta$ . memóriacellánál folytatja.

A 3. vezérlőkombináció:

— 10 0000  $\beta$

Ilyen utasítás leolvasása után a gép az elhelyezést a  $\beta$ . címen tárolt címtől folytatja. A helyes működéshez szükséges, hogy  $\beta$ . című memóriacella tartalma 00 0000  $\delta$  alakú legyen, különben az átküldő utasítás első címrésze (esetleg műveleti jele is) módosul. A beindítóprogram használata esetén ugyanis a 3. vezérlőkombináció a t + 100. címre kerül. A gép ott végre is hajtja, így megjelenik a B. regiszterben a  $\beta$ . cím tartalma. Ehhez adódik hozzá az A műveleti jel kódja és a t + 100. első címrész. Az így összeállított utasítás a t + 56. címre kerül és átadódik a vezérlés a t + 14. utasításnak, azaz a gép folytatja tovább az utasítások leolvasását, de most már az új átküldő utasítás mellett.

A 4. vezérlőkombináció alakja:

— 24 t + 14  $\beta$

Ez az utasítás a t + 100. címen végrehajtásra kerül, miután a t + 77. utasítás hatására előállt a B. regiszterben a soronkövetkező m. memóriapozíció címe  $2^{-30}$  nagyságrendben. Ez a cím tehát elhelyeződik a  $\beta$ . memóriarekeszbe 00 0000 m alakban. A vezérlés átadódik a 14. utasításnak, hogy a gép folytassa tovább a szalagról való leolvasást.



Az 5. vezérlőkombináció:

— 74 —  $\beta$

A beindítóprogram végrehajtása során átadódik a vezérlés a  $t + 100$ . memóriarekeszben lévő 5. vezérlőkombinációnak. Ez feltétlen vezérléstátadást jelent a  $\beta$ . címen lévő utasításnak. Így a gép abbahagyja a szalagról való leolvasást és elkezdi az utasítások végrehajtását a  $\beta$ . címtől.

A beindítóprogram, amíg bent van a memóriában, egy  $U_2$  —  $t + 10$  utasítással bármikor felhívható.

$t + 10$	07	-	$t + 115$
$t + 11$	07	7	$t + 112$
$t + 12$	05	8	$t + 114$
$t + 13$	05	$t + 107$	$t + 25$
$t + 14$	07	-	$t + 100$
$t + 15$	01	6	$t + 116$
$t + 16$	04	$t + 17$	$t + 56$
$t + 17$	00	6	$t + 118$
$t + 20$	15	0000	$t + 100$
$t + 21$	04	$t + 52$	$t + 23$
$t + 22$	06	7	$t + 25$
$t + 23$	02	6	$t + 114$
$t + 24$	15	$t + 113$	$t + 113$
$t + 25$	[		
$t + 26$	14	$t + 111$	$t + 30$
$t + 27$	16	$t + 100$	$t + 207$
$t + 30$	31	$t + 106$	-
$t + 31$	04	$t + 54$	$t + 30$
$t + 32$	00	$t + 112$	$t + 22$
$t + 33$	00	5	$t + 20$
$t + 34$	[		
$t + 35$	14	000	2
$t + 38$	20	$t + 100$	$t + 100$
$t + 37$	00	-	$t + 113$
$t + 40$	00	7	$t + 35$
$t + 41$	15	$t + 107$	$t + 2$
$t + 42$	04	$t + 106$	-
$t + 43$	04	$t + 54$	$t + 20$
$t + 44$	00	8	$t + 111$

Utasításrendszer:

t	+	00	0000	a	} relativ címzés alapsjai
t + 1	+	00	b	c	
t + 2	+	00	d	e	
t + 3	+	00	f	g	
t + 4	+	00	h	i	
t + 5	+	00	j	k	
t + 6	+	00	l	m	
t + 7	+	00	n	o	
t + 10	Be	07	-	t + 115	
t + 11	A'	05	η	t + 113	
t + 12	A'	05	ε	t + 114	
t + 13	A'	05	t + 107	t + 25	
t + 14	Be	07	-	t + 100	
t + 15	-	01	ε	t + 116	
t + 16	Fu	34	t + 17	t + 56	
t + 17	+	00	ε	t + 116	
t + 20	∧,	16	0000	t + 100	
t + 21	Fu	34	t + 62	t + 23	
t + 22	∧	06	η	t + 25	
t + 23	:	02	α	t + 114	
t + 24	∧,	16	t + 115	t + 113	
t + 25	[				]
t + 26	↓+	20	t + 111	t + 34	
t + 27	(∧,	16	t + 100	t + 105)	
t + 30	↓-,	31	t + 106	-	
t + 31	Fu	34	t + 34	t + 32	
t + 32	+	00	t + 112	t + 35	
t + 33	(+	00	δ	t + 34)	
t + 34	[				]
t + 35	(↓+,	30	0000	-	)
t + 36	↓+	20	t + 100	t + 100	
t + 37	x	03	α	t + 113	
t + 40	∧	06	η	t + 35	
t + 41	∧,	16	t + 27	ψ + 1	
t + 42	↓-,	31	t + 106	-	
t + 43	Fu	34	t + 44	t + 52	
t + 44	+	00	δ	t + 111	

t + 45	+	00	$\epsilon$	t + 27
t + 46	+	00	$\beta$	t + 33
t + 47	+	00	$\beta$	t + 112
t + 50	-,	11	$\alpha$	t + 114
t + 51	Fu	34	t + 23	t + 22
t + 52	-	01	$\delta$	t + 111
t + 53	-	01	$\epsilon$	t + 27
t + 54	-	01	$\beta$	t + 33
t + 55	-	01	$\beta$	t + 112
t + 56	( $\hat{A}$	05	t + 100	m ( )
t + 57	+	00	$\epsilon$	t + 56
t + 60	-,	11	t + 107	t + 25
t + 61	Fu	34	t + 10	t + 14
t + 62	$\wedge$ ,	16	t + 100	$\eta$ ( )
t + 63	$\downarrow$ -,	31	$\alpha$	- ( )
t + 64	Fu	34	t + 65	t + 75
t + 65	$\wedge$ ,	16	t + 100	$\varphi$ ( )
t + 66	$\downarrow$ -,	31	$\beta$	- ( )
t + 67	Fu	34	t + 70	t + 73
t + 70	$\hat{A}$	05	t + 100	t + 71
t + 71	[			]
t + 72	U2	74	-	t + 14
t + 73	-,	11	t + 100	0000
t + 74	U1	24	t + 14	t + 56
t + 75	$\downarrow$ -,	31	$\alpha$	-
t + 76	Fu	34	t + 100	t + 103
t + 77	$\wedge$ ,	16	t + 56	$\psi + 1$
t + 100	[			]
t + 101	$\downarrow$ +,	30	t + 110	-
t + 102	U1	24	t + 14	t + 56
t + 103	$\downarrow$ -,	31	$\alpha$	-
t + 104	Fu	34	t + 77	t + 100
t + 105	+	00	4000	0000 (= $2^{-7}$ )
t + 106	+	00	0000	4000 (= $2^{-19}$ )
t + 107	$\downarrow \times$ ,	33	t + 114	0000
t + 110	$\hat{A}$	05	t + 100	0000
t + 111	( $\wedge$ ,	16	$\psi$	t )
t + 112	(:	02	$\gamma$	0000 )



t + 113	leválasztó konstans			
t + 114	átszorzó konstans			
t + 115	módosítószó			
t + 116	(+ 00 0000 0000)			

Felhasznált szabvány-konstansok:

(0)	=	+	00	0000	0000	
(α)	=	+	10	0000	0000	(= 2 <sup>-3</sup> )
(β)	=	-	01	0000	0000	(= 2 <sup>-6</sup> )
(γ)	=	+	00	0100	0000	(= 2 <sup>-12</sup> )
(δ)	=	+	00	0001	0000	(= 2 <sup>-18</sup> )
(ε)	=	+	00	0000	0001	(= 2 <sup>-30</sup> )
(η)	=	√ + ,	70	0000	0000	(= 7·2 <sup>-3</sup> )
(φ)	=	Be	07	0000	0000	(= 7·2 <sup>-6</sup> )
(ψ)	=	+	00	7777	0000	(= 7777·2 <sup>-18</sup> )
(ψ+1)	=	+	00	0000	7777	(= 7777·2 <sup>-30</sup> )

Példa a beindítóprogram alkalmazására.

Két vektor skaláris szorzatának kiszámítására szolgáló nyitott szubrutin.

Legyenek  $\vec{A}(a_1, a_2, \dots, a_n)$  és  $\vec{B}(b_1, b_2, \dots, b_n)$  legfeljebb 63 dimenziós vektorok. Tegyük fel, hogy az  $\vec{A}$  első komponense az x., B első komponense az y. memóriarekeszben található. Az  $\vec{A}$  komponenseit tartalmazó memóriacellák címei alkotnak h,  $\vec{B}$  komponenseit tartalmazó memóriarekeszek címei pedig k differenciájú számtani sorozatot. A kiszámított  $\vec{A} \cdot \vec{B} = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$  szorzat kerüljön a z. memóriacellába.

Feltételezzük, hogy

(o)	=	+	00	0000	0000	
(p)	=	-	01	0000	0000	(= $2^{-6}$ )
(κ)	=	M	77	0000	0000	(= $77 \cdot 2^{-6}$ )
(λ)	=	+	00	0000	0003	(= $3 \cdot 2^{-30}$ )

A feladat megoldásához a következő utasításokat kell a szalagra gépelni a beindítóprogram I. változatának használata mellett:

x	03	0000	0000	- módosítószó
-	24	t + 14	t	} paraméterek elhelyezése
-	05	t + 100	t + 1	
-	00	λ	t + 116	
	n	x	y	
+	00	h	k	
+	00	0000	z	
-	10	0000	t	
Á	05	0000	0000	} módosítandó utasítások
Λ,	16		t + 4001	
U1	24	4003	0011	
-	01		0011	
Fu	34	4012	0005	
-,	11	3300	0000	- módosítószó

X,	13	0000	0000	} módosítandó utasítások
↓+	20	4000	0000	
+	00	t + 2	0005	
U2	74	0000	0003	
M	77	0000	0000	

A rutin feltételezi, hogy az előzőleg elhelyezett utasítás ötödikként módosított utasítás volt. Ez mindig elérhető. Ennek biztosítására az egyik mód az, hogy bizonyos számú 0-val módosítandó utasítást nem módosító szó segítségével hagyunk változatlanul, hanem eléjük 1. vezérlőkombinációt gépelünk (mint a fenti példában). Másik mód az ötös csoportosítás biztosítására, hogy kipótoljuk az utasításokat olyanokkal, amelyek a gép működésében nem okoznak zavart. Ilyenek lehetnek pl. + 00 0000 0000, vagy ↓+, 30 0000 0000. Ezekből legfeljebb 4 kell, de az általuk elfoglalt memóriarekesz felhasználható munkapozícióként vagy konstansok tárolására.

Ha a rutint megelőző utolsó utasítás az s - 1. címre került, akkor a - 24 t + 14 t vezérlőkombináció hatására az s elhelyeződik a t. címre.

A - 05 t + 100 t + 1 vezérlőkombináció azt eredményezi, hogy a paraméterszavak a t + 1., t + 2., t + 3. memóriarekeszbe kerülnek. A - 10 0000 t vezérlőkombináció hatására, mivel a t. memóriarekeszben az s van, a következő utasításokat, tehát rutin utasításait az s. cellától helyezi el a gép.

Az elhelyezett szavak a következő alakban kerülnek a memóriába:

t	+	00	0000	s
t + 1		n	x	y
t + 2	+	00	h	k
t + 3	+	00	0000	z



s	Á	05	0000	z
s + 1	Λ,	16	κ	b + 1
s + 2	u1	24	s + 3	s + 11
s + 3	-	01	β	s + 11
s + 4	Fu	34	s + 12	s + 5
s + 5	x,	13	x	y
s + 6	↓+	20	z	z
s + 7	+	00	t + 2	s + 5
s + 10	u2	74	-	s + 3
s + 11	M	77	0000	0000

A végrehajtás során az s. utasítás kiüríti az eredmény helyét. Erre azért van szükség, mert a z. rekeszt egyúttal munkapozícióknak is használjuk: ennek tartalmához adjuk hozzá az egyes komponensszorzatokat.

Az s + 1. utasítás leválasztja a t + 1. címen tárolt paraméterszóból a dimenziószámot. Az s + 2. utasítás elhelyezi az s + 11. memóriapozícióba,  $2^{-6}$  nagyságrendben. Ez a memóriacella számlálórekeszként szolgál.

Ha  $n \geq 1$ , az s + 5. utasítás elvégzi az első komponensek szorzását. A szorzat az s + 6. utasítás hatására átkerül z. címre.

Az s + 7. utasítás módosítja az s + 5. utasítást. Ezután feltétlenül átadódik a vezérlés az s + 3. utasításnak, amely csökkenti a számlálórekesz tartalmát. Ezzel kezdődik a ciklus előlről. A ciklusból akkor ugrik ki a gép, ha az n. komponensek szorzatát is hozzáadta a z. cím tartalmához, azaz, ha elkészült az eredmény. Az s + 4. utasítás adja át a vezérlést a szubrutin után következő első utasításnak.

Elvé ugyanaz, mint az I. változat által megvalósított átváltás első lépésének.

A helyes működés előfeltétele itt is az, hogy a beindítórutin hatására leolvasott első utasítás módosítóeső legyen, melynek alakja most is:

$$z = \frac{z}{2^6}$$

## II. változat.

Az I. változat cimmódosításának elvéből következik, hogy a beindítóprogram használata esetén csak 4000-ig terjedhet a használt memoriacellák száma. (Oktális rendszerben.) Ezt a korlátozást megszünteti a beindítóprogram II. változata.

A II. változat 48 utasításból áll. Az utasításokat a  $t + 10$ -tól a  $t + 67$ -ig terjedő memoriacellákba helyezzük el. Itt  $t$  ugyanugy, mint az I. változatnál a paraméterek számára fenntartott első memoriapozíció címét jelenti.  $t$  rögzítése után a  $t$ -től függő utasításokat megfelelően átalakítva, abszolút címekkel visszük be a beindítórutint a memóriába.

A programban felhasználunk 7 speciális konstanst, amelyek a program utasításai után,  $t + 70$ -tól nyertek elhelyezést. Szükség volt 7 szabványkonstansra, melyek címét ideiglenesen ugyanugy, mint az I. változatnál, görög kisbetűkkel jelöltük.

Az utasítások végrehajtása során módosuló konstansok a  $t + 100$ . és a  $t + 101$ . rekeszben találhatók.

Két munkarekeszt használunk:  $t + 77$ . címre olvassa le a gép a módosítószót. A  $t + 102$ . cella kiürítéséről a beindítóprogram behívása előtt gondoskodni kell. Ez a munkapozíció szolgál a változatlan elhelyezés jelzésére.

Helyes működés mellett a munkapozíciókban lévő utasítások végrehajtásra nem kerülnek.

A módosítószavak kivételével a gép minden utasítást a  $t + 63$ . címre helyez el. Az ide került utasítások vizsgálata és módosítása után a nem-vezérlőkombinációkat a  $t + 41$ . utasítás viszi be a memória megfelelő rekeszeibe.

## Átcimzés.

Elve ugyanaz, mint az I. változat által megvalósított átcimzés első lépésének.

A helyes működés előfeltétele itt is az, hogy a beindítórutin hatására leolvasott első utasítás módosítószó legyen, melynek alakja most is:

$$\pm r_{11} r_{12} \dots r_{51} r_{52}$$



$r_{ij}$  ( $i = 1, \dots, 5; j = 1, 2$ ) jelentése és alakja az I. változat leírásában szereplő.

A II. változat nem veszi figyelembe az átcimzésnél az utasítás előlről számított 7. és 19. bináris jegyének értékét.

Ebből következik, hogy:

- 1.) A relativ címzés alapja csak 8-féle lehet. Ezeket  $t$ ,  $t + 1, \dots, t + 7$ . memoriacellákba helyezzük el,  $2^{-30}$  nagyságrendben. A cellák többi bitjén bármi lehet, (pl.: címreszmódosításra fel nem használandó egyéb paraméterek) ugyanis a beindítóprogram csak az utolsó 12 bitet veszi figyelembe átcimzésnél.
- 2.) Az I. változatnál a felhasználható memoriacellák számára adódott korlátozás megszűnt.

A címmodosítás menete:

A  $t + 10$ . utasítás hatására a gép elolvassa a módosítószót, s elhelyezi a  $t + 77$ . címre.

Az utasítások kezdeti értékeinek beállítása után a módosítandó utasítások átcimzését a  $t + 23$ .-tól  $t + 31$ -ig elhelyezett utasítások végzik el. 5. utasítás első és második címreszének módosítása után a módosítószó "elfogy". A helyes működéshez az 5. módosítandó utasítás után rögtön módosítószót kell gépelnünk.

### Vezérlőkombinációk:

1. vezérlőkombináció:

— 00  $\langle a^* \rangle$   $t + 102$

Ilyen vezérlőkombináció után következő utasításokból "a" számú változatlanul helyez el a gép.

2. — 05  $t + 63$   $\beta$

Ennek hatására a gép az elhelyezést  $\beta$ . címtől folytatja.

3. — 10 0000  $\beta$

Ilyen utasítás után az elhelyezést ( $\beta$ ). címtől folytatja a gép.



4. vezérlőkombináció:

$$\text{--- } 24 \quad t + 14 \quad \beta$$

Hatására a soronkövetkező memoriapozíció címe elhelyeződik a  $\beta$ . memoriarekeszbe.

5. vezérlőkombináció:

$$\text{--- } 74 \quad \text{--- } \beta$$

után a gép elkezd az utasítások végrehajtását a  $\beta$ . címtől.

Egyébként a vezérlőkombinációk hatására a gép ugyanugy működik, mint az I. változat megfelelő vezérlőkombinációi esetén.

Ez a beindítóprogram is bármikor felhívható egy  $u_2 \text{ --- } t + 10$  utasítással, ha bennevan a memoriában.

Utasításrendszer

t	+	00	0000	0000	} relativ cimzés alapjai
t + 1	+	00	0000	a	
t + 2	+	00	0000	b	
t + 3	+	00	0000	c	
t + 4	+	00	0000	d	
t + 5	+	00	0000	e	
t + 6	+	00	0000	f	
t + 7	+	00	0000	g	
t + 10	Be	07	-	t + 77	
t + 11	Á	05	η	t + 100	
t + 12	Á	05	ε	t + 101	
t + 13	Á	05	t + 70	t + 25	
t + 14	Be	07	-	t + 63	
t + 15	-	01	ε	t + 102	
t + 16	Fu	34	t + 17	t + 41	
t + 17	+	00	ε	t + 102	
t + 20	Λ,	16	0000	t + 63	
t + 21	Fu	34	t + 45	t + 23	
t + 22	Λ	06	η	t + 25	
t + 23	:	02	α	t + 101	
t + 24	Λ,	16	t + 77	t + 100	
t + 25	[				]
t + 26	↓+	20	t + 73	t + 27	
t + 27	[				]
t + 30	(↓:,	32	γ	-	)
t + 31	↓+	20	t + 63	t + 63	
t + 32	x	03	α	t + 100	
t + 33	-	01	t + 71	t + 30	
t + 34	↓-,	31	t + 75	-	
t + 35	Fu	34	t + 40	t + 36	
t + 36	-,	11	α	t + 101	
t + 37	Fu	34	t + 24	t + 23	
t + 40	+	00	t + 71	t + 30	

t + 41	(Á	05	t + 63	m)
t + 42	+	00	ε	t + 41
t + 43	-,	11	t + 76	t + 100
t + 44	Fu	34	t + 10	t + 14
t + 45	^,	16	t + 63	η
t + 46	↓-,	31	α	-
t + 47	Fu	34	t + 50	t + 60
t + 50	^,	16	t + 63	φ
t + 51	↓-,	31	β	-
t + 52	Fu	34	t + 53	t + 56
t + 53	Á	05	t + 63	t + 54
t + 54	[			]
t + 55	u2	74	=	t + 14
t + 56	-,	11	t + 63	0000
t + 57	u1	24	t + 14	t + 41
t + 60	↓-,	31	α	-
t + 61	Fu	34	t + 63	t + 66
t + 62	^,	16	t + 41	ψ + 1
t + 63	[			]
t + 64	+,	30	t + 74	=
t + 65	u1	24	t + 14	t + 41
t + 66	↓-,	31	α	=
t + 67	Fu	34	t + 62	t + 63
t + 70	↓x,	33	t + 101	=
t + 71	:	02	γ	0000
t + 72	↓+,	30	0000	0000
t + 73	^,	16	ψ + 1	0000
t + 74	Á	05	t + 63	0000
t + 75	↓+,	30	0000	0000
t + 76	+	00	0000	0007
t + 77	módosítószó			
t + 100	leválasztó konstans			
t + 101	átszorzó konstans			
t + 102	(+	00	0000	0000)

(=  $3 \cdot 2^{-3}$ )  
(=  $7 \cdot 2^{-30}$ )





Felhasznált szabvány-konstansok:

(0)	=	+	00	0000	0000	
( $\alpha$ )	=	+,	10	0000	0000	(= $2^{-3}$ )
( $\beta$ )	=	-	01	0000	0000	(= $2^{-6}$ )
( $\gamma$ )	=	+	00	0100	0000	(= $2^{-12}$ )
( $\epsilon$ )	=	+	00	0000	0001	(= $2^{-30}$ )
( $\eta$ )	=	$\downarrow + $ ,	70	0000	0000	(= $7 \cdot 2^{-3}$ )
( $\varphi$ )	=	Be	07	0000	0000	(= $7 \cdot 2^{-6}$ )
(+1)	=	+	00	0000	7777	(= $7777 \cdot 2^{-30}$ )

### III. változat.

Ez a beindítóprogram nem korlátozza se a paraméterek, se a felhasználható memoriacellák számát.

47 utasításból áll, amelyek a  $t$ -től a  $t + 56$ -ig terjedő cellákban vannak elhelyezve. Az első utasítás címének,  $t$ -nek rögzítése után megfelelően átcimezve, abszolút címekkel kell bevinni a memóriába.

A program 3 speciális konstanst használ fel, amelyeket a  $t + 60.$ ,  $t + 61.$  és  $t + 62.$  címeken helyeztünk el. Szerepel 7 szabvány-konstans, melyek címét ideiglenesen itt is a görög ABC kisbetűi jelölik.

A  $t + 57.$  cellát munkapozícióként használja a beindítóprogram a változatlan elhelyezés jelzésére, melynek kiürítéséről a program behívása előtt gondoskodni kell.

A program mindig a  $t + 21.$  címre helyezi el a leolvasott utasítást, innen vizsgálat és átcimzés után a  $t + 12.$  utasítás helyezi el a memoria soronkövetkező rekeszébe.

### Vezérlőkombinációk:

Ez a beindítóprogram a 77 műveleti kódu pozitív utasításokat használja vezérlőkombinációknak. Az utasítás első címrésze határozza meg, hogy milyen vezérlőkombinációról van szó, a második címrésze viszont valamilyen adatot ad meg.

Az 1. vezérlőkombináció:

+ 77 0 <a\*\*>

Ennek hatására a  $t + 21.$  és  $t + 22.$  utasítás végrehajtása során elhelyeződik a  $t + 57.$  címre az  $a^{**}$ , s átadódik vezérlés a  $t.$  utasításnak. Ez azt eredményezi, hogy a következő utasítások közül "a" számú változatlanul kerül a memoria megfelelő rekeszeibe. Így lehetővé válik mind a 77-tel kezdődő, mind a negatív utasítások elhelyezése.

2. vezérlőkombináció:

$$+ 77 \quad 1 \quad \beta$$

Ezen vezérlőkombináció leolvasása után a  $t + 26.$  és  $t + 27.$  utasítások végrehajtásra kerülnek, melyek hatására a  $t + 12.$  címen lévő átküldő utasítás második címrésze  $\beta.$  lesz. A vezérlés átadódik a  $t.$  utasításnak, s a következő utasításokat ma a  $\beta.$  címtől helyezi el a gép.

3. vezérlőkombináció:

$$+ 77 \quad 2 \quad \beta$$

Ennek hatására a  $t + 32$ -től  $t + 35$ -ig terjedő utasításokat hajtja végre a gép. Így a  $t + 12.$  utasítás a  $t + 21$  ( $\beta$ )-ra változik, tehát a következő utasítást a gép már a  $\beta.$  címen tárolt címtől folytatja.

4. vezérlőkombináció:

$$+ 77 \quad 3 \quad \beta$$

Ezen utasítás leolvasása után a  $t + 40.$  és  $t + 41.$  utasítás hatására átadódik a vezérlés a  $\beta.$  címen lévő utasításnak.

5. vezérlőkombináció:

$$+ 77 \quad 4 \quad \beta$$

Ennek a vezérlőkombinációnak a hatására a gép a  $t + 44$ -től  $t + 47$ -ig terjedő utasításokat hajtja végre. Ennek eredményeképpen a  $t + 12.$  utasítás második címrésze, mely a soronkövetkező utasítás címét adja meg, elhelyeződik a  $\beta.$  memoriapozícióba  $2^{-30}$  nagyságrendben, s a gép folytatja az utasítások leolvasását.

Átcimzés:

A módosítószó alakja:

$$+ 77 \quad \alpha \quad \beta \quad (\alpha \geq 5)$$

Ezt, mivel 77 a műveleti kódja, a gép vezérlőkombinációként kezeli. Hatására a  $t + 50.$ -től a  $t + 56.$ -ig terjedő utasítások előállítják a  $+$  ( $\alpha$ ) ( $\beta$ ) alakú utasítást. Ez elhelyeződik a



t + 54. memoriacellába, s a gép folytatja az utasítások leolvasását. A módosítandó utasítás előjele negatív, alakja

$$= \Gamma \quad \mu \quad \nu$$

Ha a gép ilyen utasítást talál, mely nincs 770 <a\*\*> alakú utasítás hatása alatt, akkor kivonja belőle a t + 54. cella tartalmát. A memóriába a következő utasítás helyeződik el:

$$= \Gamma \mu+(\alpha) \quad \nu+(\beta)$$

A beindítóprogram behívása U2-t utasítással történhet.

Utasításrendszer

t	Be	07	-	t + 21
t + 1	-	01	ε	t + 57
t + 2	Fu	34	t + 3	t + 12
t + 3	+	00	ε	t + 57
t + 4	∧,	16	t + 21	κ
t + 5	↓-,	31	κ	-
t + 6	Fu	34	t + 7	t + 15
t + 7	∧,	16	0000	t + 21
t + 10	Fu	34	t + 11	t + 12
t + 11	-	01	t + 54	t + 21
t + 12	(Á	05	t + 21	m)
t + 13	+	00	ε	12+t
t + 14	u2	74	-	t
t + 15	-	01	κ	t + 21
t + 16	∧,	16	t + 21	ψ
t + 17	↓-	21	δ	t + 51
t + 20	Fu	34	t + 21	t + 23
t + 21	[			]
t + 22	U1	24	t	t + 57
t + 23	∧	06	ψ + 1	t + 21
t + 24	-,	11	δ	t + 51
t + 25	Fu	34	t + 26	t + 30
t + 26	+,	10	t + 60	t + 21
t + 27	U1	24	t	t + 12
t + 30	↓-,	31	δ	-
t + 31	Fu	34	t + 32	t + 36
t + 32	Á	05	t + 21	t + 33
t + 33	[			]
t + 34	↓+	20	t + 60	t + 12
t + 35	u2	74	-	t
t + 36	↓-,	31	δ	-
t + 37	Fu	34	t + 40	t + 42



t + 40	+	00	t + 62	t + 21
t + 41	U2	74	-	t + 21
t + 42	↓-,	31	δ	-
t + 43	Fu	34	t + 44	t + 50
t + 44	+,	10	t + 61	t + 21
t + 45	u1	24	t + 46	t + 47
t + 46	^,	16	ψ + 1	t + 12
t + 47	[			]
t + 50	+	00	δ	t + 51
t + 51	[			]
t + 52	↓:	22	γ	t + 51
t + 53	Á	05	t + 21	t + 54
t + 54	[			]
t + 55	↓+	20	t + 51	t + 54
t + 56	u2	74	=	t
t + 57	változatlan elhelyezés jelzése.			
t + 60	Á	05	t + 21	0000
t + 61	u1	24	t	0000
t + 62	u2	74	-	0000

Felhasznált szabvány-konstansok:

(0)	=	+	00	0000	0000	
(f)	=	+	00	0100	0000	(= 2 <sup>-12</sup> )
(δ)	=	+	00	0001	0000	(= 2 <sup>-18</sup> )
(ε)	=	+	00	0000	0001	(= 2 <sup>-30</sup> )
(ψ)	=	+	00	7777	0000	(= 7777 · 2 <sup>-18</sup> )
(ψ+1)	=	+	00	0000	7777	(= 7777 · 2 <sup>-30</sup> )
(u)	=	M	77	0000	0000	(= 77 · 2 <sup>-6</sup> )







EX LIBRIS

KALMÁR  
LÁSZLO

*Kalmár László*