

# MANUAL

MONS--GENS

ALPHA Ltd. @ 1991



## IMPORTANT !

Editura "TM" pune la dispozitia tuturor celor interesati întreaga gamă de manuale în limba română pentru calculatoare compatibile ZX Spectrum (TIM S, TIM S Plus, COBRA, HC 85, CIP, Jet) editate de firma "ALPHA Ltd" S.R.L. :

- 1.01 Limbajul BASIC pe înțelesul tuturor în 12 lectii
- 1.02 Documentatie GENS și MONS (Asamblor-dezasamblor)
- 1.03 Documentatie limbaj FORTH
- 1.04 Documentatie BETA BASIC 3.1 (Extensie BASIC)
- 1.05 Documentatie BETA BASIC 3.1 (Rezumat)
- 1.06 Documentatie compilator FORTRAN 77-8
- 1.07 Documentatie editor de texte TASHWORD
- 1.08 Documentatie compilator BLAST
- 1.09 Documentatie compilator PASCAL HP4TM (Rezumat)
- 1.10 Documentatie limbaj C
- 1.11 Memento timing cod mașină Z80
- 1.12 Documentatie MEGA BASIC (Extensie BASIC)
- 1.13 Documentatie VU-CALC
- 1.14 Manual BASIC avansati - conținând și referiri la COBRA
- 1.15 Documentatie compilator COLT
- 1.16 Documentatie MASTER - FILE (sistem gestiune afaceri)
- 1.17 Documentatie limbaj microPROLOG
- 1.18 Documentatie limbaj PASCAL HP4TM
- 1.19 Documentatie sistem operare CP/M cu referire la calculatorul COBRA
- 1.20 Manual ROM SPECTRUM complet dezasmblat
- 1.21 Documentatie LASER GENIUS (pachet programe pentru lucrul în cod mașină)
- 1.22 Cum să obținem cât mai mult de la calculatorul nostru o carte cu programe și trucuri atât pentru începători cât și pentru avansati, în două variante:
  - a) Numai cartea, cu o parte din figuri în text
  - b) Cartea și o casetă demonstrativă, cu toate programele și figurile introduse
- 1.23 Construiți singuri 20 de montaje electronice interfașabile cu microcalculatorul DVE

GENS ASSEMBLER (©) 1983 HISOFT  
 =====

Adaptare Aurel Gontean  
 Introdus Adriana Gontean

**NOTA TRADUCATORULUI:** Desi cea mai raspindita varianta in circulatie este GENSSM, traducerea s-a facut dupa documentatia unei variante mai vechi, fiind singura disponibila. In sectiunea comenzilor pentru microdrive s-au folosit traduceri din revista MQJ MICRO si observatii bazate pe experienta traducatorului.

GENS este un asamblor Z80 puternic, usor de utilizat, foarte apropiat de asamblorul Zilog standard. Are o lungime de 7 k (GENSSM are 9 k), odata relocat ocupand ceva mai putin si foloseste o stiva proprie. GENS contine propriul sau editor care plaseaza fisierul text (sursa) imediat dupa zona de memorie pe care o ocupa, urmat de tabela de simboluri. Se recomanda incarcarea asamblorului in partea inferioara a memoriei.

1. INTRODUCERE  
 =====

Se incarca cu LOAD "" CODE xxxxx  
 Se lanseaza cu RANDOMIZE USR xxxxx (prima data)  
 si relocat RANDOMIZE USR xxxxx+56 (pornire 'rece')  
 RANDOMIZE USR xxxxx+64 (pornire 'calda')

La inceput apare mesajul: "Buffer size?:". Se introduce un numar intre 0 si 9 inclusiv, urmat de ENTER sau numai ENTER pentru valoarea implicita. Numarul introdus reprezinta factorul de multiplicare a 256 octeti. Daca se doreste minimizarea spatiului ocupat de GENS si spatiul sau de lucru, (si nu se ia in considerare folosirea eficienta a optiunii INCLUDE), se poate apasa 0, asigurandu-se astfel cel mai mic buffer posibil (64 octeti).

In continuare apare simbolul ">" care indica intrarea in monitor.

<p>ATENTIE! GENS DEZACTIVEAZA INTRERUPERILE          SI MODIFICA VALOAREA REGISTRULUI IY.</p>
---

## 2.DETALII

\*\*\*\*\*

La apelarea asamblorului (comanda A) acesta intreaba la inceput "Table size?:". Raspunsul este un numar zecimal care reprezinta cantitatea de memorie ce va fi alocata pentru tabela de simboluri. Valoarea implicita (obtinuta prin apasarea lui ENTER) se apreciaza de asamblor functie de lungimea textului si este in general perfect acceptabila. Folosind optiunea INCLUDE trebuie specificata o tabela de simboluri mai mare decat cea implicita, asamblorul putind anticipa marimea fisierului text care va fi inclus.

Urmeaza cererea optiunilor cu mesajul "Options:"

### OPTIUNI:

- 1 - produce listarea tabelii de simboluri la sfirsitul celei de-a doua treceri a asamblorului.
- 2 - nu genereaza cod obiect.
- 4 - nu listeaza programul asamblat
- 8 - listeaza programul asamblat la imprimanta.
- 16 - pune codul obiect (daca este generat) dupa tabela de simboluri. Contorul de locatii este controlat de directiva ORG deci codul obiect poate fi plasat intr-o portiune de memorie desi ar rula normal doar altundeva.
- 32 - nu mai verifica unde se plaseaza codul obiect (util pentru asamblari rapide).

Daca s-a utilizat optiunea 16, directiva ENT nu mai are efect. Adresa de inceput a codului obiect se poate afla folosind comanda "X" pentru a gasi sfirsitul textului la valoarea obtinuta adaugandu-se 2. Asamblarea are loc in doua treceri.

La prima trecere, GENS cauta erorile si compileaza tabela de simboluri. La a doua trecere se genereaza codul obiect, mai putin daca s-a folosit optiunea 2. La prima trecere nu se afiseaza nimic pe ecran sau pe imprimanta decat daca se detecteaza o eroare. In acest caz se afiseaza numarul liniei care contine eroarea impreuna cu un cod al erorii (vezi ANEXA), asamblarea se opreste, si se poate apasa "E" pentru intoarcerea la editor sau orice alta tasta pentru a continua asamblarea. La sfirsitul primei treceri apare mesajul "Pass 1 errors: nn", si in cazul in care sint erori nu se trece mai departe. Poate aparea si mesajul "WARNING label absent" pentru fiecare eticheta care lipseste. Dupa a doua trecere se genereaza listingul asamblarii (mai putin daca s-a utilizat optiunea 4).

Listingul rezultat in urma asamblarii este in general de forma :

nr. de coloana

1	6	15	21	26	linia 1	
			21	24	26	linia 2
C000	210100	25	etich			
			ld HL, 1			

Primul cimp al liniei reprezinta valoarea contorului de locatii la inceputul executiei liniei respective (in afara de cazul in care mnemonicul din respectiva linie este un pseudo-mnemonic ORG, EQU, ENT) in care caz cimpul respectiv va reprezenta valoarea din cimpul operandului din instructiune. In general valoarea se afiseaza in hexazecimal, dar se poate si in zecimal prin folosirea comenzii "\*D+" a asamblorului.

Urmatatorul cimp din coloana 6 avind pina la 8 caractere lungime (deci 4 octeti) este codul obiect produs de instructiunea curenta (atentie la comanda asamblorului "\*C").

Urmeaza numarul de linie, un numar intreg cuprins intre 1 si 32767 inclusiv.

Coloanele 21-26 din prima linie contin primele 6 caractere a oricarei etichete definite in aceasta linie.

Dupa fiecare eticheta urmeaza o noua linie. Pe aceasta linie mnemonicul este afisat intre coloanele 21-24.

Urmeaza cimpul operandului din coloana 26 a acestei linii si continutul care trebuie inserat la sfirsitul liniei generind noi linii cind este necesar.

Formatul de mai sus asigura lizibilitatea listingului asamblarii pe un ecran ingust ca al SPECTRUM-ului fara a fi necesara definirea unui nou set de caractere (ceea ce ar duce la ocuparea unui spatiu nejustificat de mare de GENS si la imposibilitatea folosirii rutinelor din ROM).

Comanda "\*C" data asamblorului produce o linie de asamblare mai scurta prin faptul ca omite cele 9 caractere reprezentind codul obiect al liniei astfel incit majoritatea liniilor asamblate incapand intr-o singuralinie. Se poate modifica impartirea liniei, ... POKE-ind 3 locatii din GENS:

START + 51 - numarul de caractere continute intr-o linie

START + 52 - coloana de la care incepe fiecare linie pe ecran

START + 53 - cite caractere din reminder-ul liniei se afiseaza pe fiecare ecran dupa prima linie.

**Exemplu.** Sa presupunem ca dorim ca prima linie a oricarei instructiuni asamblate sa contina 20 de caractere (fara cimpul etichetei) si fiecare linie ce urmeaza sa inceapa in coloana 1 umplind intreaga linie. Presupunind ca GENS este incarcat de la adresa 24064 se reda controlul interpretorului BASIC si se tasteaza:

```
POKE 24115,20
POKE 24116,1
POKE 24117,31
```

Modificarile se pot aplica doar daca nu s-a folosit comanda "C".

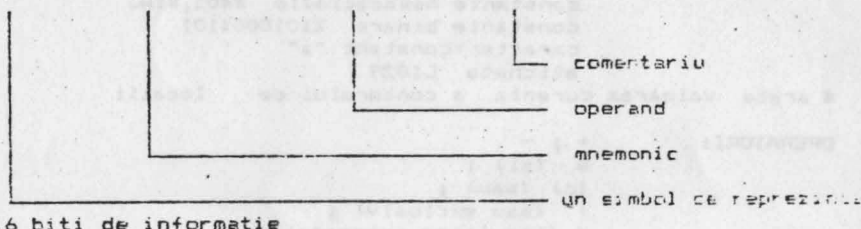
Listingul asamblarii se poate intrerupe cu "CR+SPACE". Apasind "E" se trece in editor sau se continua apasind orice alta tasta.

Singurele erori ce pot aparea la a doua trecere sint "ERROR 10" si "BAD ORG". Eroarea 10 nu este fatala, asamblarea putind fi continuata. La sfirsit va apare "PASS 2 ERRORS:nn" si mesajul de atentionare pentru etichetele inexistente si "Table used xxxxx from yyyy". Daca s-a folosit directiva ENT in mod corespunzator, va apare mesajul "Execute nnnnn" care reprezinta locul de unde programul se poate executa cu comanda "R". Daca s-a specificat optiunea 1 se afiseaza si o lista alfabetica a etichetelor folosite si a valorilor lor asociate. Numarul etichetelor afisate se poate schimba prin "POKE START+50, valoare\_dorita". In continuare controlul revine editorului.

### 3. FORMATUL INSTRUCIUNII

=====

art            LD            HL,label            ; pick up "label"



Daca o eticheta este asociata unei valori mai mari ca 255 si este apoi utilizata intr-un context in care s-ar trebui sa aiba 8 biti, apare "ERROR 10" la a doua trecere. Pentru etichete sint regale caracterele:

Literele (A-Z ; a-z)

Cifrele (0-9)

Semnele (@ - \_ etc) cu mentiunea ca o eticheta trebuie sa inceapa cu o litera.

#### 4. CONTORUL DE LOCATII

Asamblorul mentine contorul de locatii astfel incit unui simbol din cimpul etichetei sa-i fie asociata o adresa si apoi sa fie introdusa in tabela de simboluri. Acest contor de locatii poate fi initializat la orice valoare conform directivei ORG. Simbolul \$ poate fi folosit pentru a se referi la valoarea curenta a contorului de locatii (Ex.: LD HL,\$+5).

#### 5. TABELA DE SIMBOLURI

Cind o eticheta este intilnita pentru prima data, ea se introduce intr-un tabel impreuna cu doi indicatori ce-i arata valoarea asociata.....etc. Acest tip de tabela se numeste "Binary Tree Symbol Table". Lungimea unei noi intrari este de 8...13 octeti functie de lungimea simbolului.

#### 6. EXPRESII

O expresie este un operand constituit dintr-un singur termen sau o combinatie de termeni separati cu un separator.

##### Exemple:

TERMENI:                   constante zecimale 1029  
                           constante hexazecimale #405,#1A3  
                           constante binare %1010001101  
                           caracter constant "a"  
                           eticheta L1029

\$ arata valoarea curenta a contorului de locatii

OPERATORI:               + ; -  
                           & (si) ;  
                           (c) (sau) ;  
                           ! (sau exclusiv) ;  
                           \* (inmultire intreaga) ;  
                           / (impartire intreaga) ;  
                           ? (MOD)       [ a?b=a-(a/b)\*b ]

La citire se iau cei mai putin semnificativi 16 biti. Expresiile se evalueaza strict de la stinga la dreapta. O expresie intre paranteze inseamna o referire la continutul dintre paranteze. Domeniul valorilor relative este -128, +127 de la instructiunea de dupa valoarea din cimpul operandului. Se poate folosi \$ pentru domeniul -126 , 129.

Exemple:               #5000-label  
                           %10010!%1011  
                           "A"+128



"Y"-";"+7  
\$ - label+8

Se pot insera spatii intre termeni si operatori. Daca apar valori nepermise se semnaleaza ERROR 15 (depasire la inmultire) sau ERROR 14 (impartire cu zero). In alte cazuri depasirea se ignora.

## 7. DIRECTIVELE ASAMBLORULUI

ORG expresie  
 EQU expresie: trebuie precedat de o eticheta careia ii da valoarea expresiei.  
 DEFB expresie,..... : 8 biti  
 DEFW expresie,..... : 16 biti (LSB este primul)  
 DEFS expresie : creste valoarea contorului de locatii cu valoarea expresiei (pentru a rezerva loc in memorie)  
 DEFM "s" : defineste continutul a n octeti de memorie egal cu reprezentarea ASCII a sirului s unde n este lungimea sirului ce poate fi teoretic cuprinsa intre 1-255 inclusiv desi practic este limitata de lungimea maxima a liniei ce poate fi introdusa din editor. Primul caracter din cimpul operandului (in cazul nostru ~~~) este considerat ca separator si lungimea este data intre doi separatori. Caracterul "sfirsit de linie" actioneaza ca sfirsit de sir.  
 ENT expresie : stabileste adresa de executie a codului obiect asamblat la valoarea expresiei. Se foloseste impreuna cu comanda "R" a editorului si nu are valoare implicita.

## 8. PSEUDOMNEMONICE CONDITIONATE

Acestea permit programatorului sa includa sau nu anumite sectiuni ale textului in procesul asamblarii.

IF expresie - evalueaza expresia si daca rezultatul este 0 asamblarea se opreste pina cind se intilnesc ELSE sau END. Pentru o valoare diferita de 0 se continua asamblarea.

ELSE - daca asamblarea este pornita (ON) inainte de ELSE ea se opreste (OFF) si viceversa.

END - opreste asamblarea.

## 9. COMENZILE ASAMBLORULUI

Spre deosebire de directive, comenzile nu au efect asupra codului obiect si sint linii ale textului sursa care incep cu "\*".

\*E - (EJECT) se trimite la ecran sau imprimanta 3 linii

albe. Este utila in separarea modulelor.

\*Hs - face ca sirul s sa fie luat ca "heading", tiparit dupa fiecare EJECT (\*E). \*H face automat si \*E.

\*S - determina oprirea listarii la aceasta linie. Listarea reincepe apasind orice tasta. Comanda este utila la citirea adreselor in mijlocul unui listing si este recunoscuta si dupa comanda "\*L-" neopriind tiparirea.

\*L- - face ca listarea si tiparirea sa se opreasca dupa aceasta linie.

\*L+ - listarea si tiparirea reincep dupa aceasta linie.

\*C- - scurteaza listingul asamblarii dupa linia urmatoare, nemaiafisind codul obiect generat.

\*C+ - revine la listarea completa a asamblarii.

\*F -(File name)- permite asamblarea textului de pe banda. Fisierul text este introdus intr-un buffer pe blocuri si asamblat acolo. Astfel sint posibile coduri obiect lungi deoarece textul odata asamblat nu mai ocupa mult spatiu in memorie. "File name" are pina la 10 caractere si trebuie precedat de un spatiu. Daca nu se precizeaza "File name" se incarca primul gasit. Acest text trebuia pus pe banda cu comanda "T" si nu cu comanda "P". Lungimea blocului din buffer se ia in multiplii de 256 octeti. Valoarea implicita este  $4 \times 256 = 1024$  octeti si trebuie specificat acelaasi "Buffer size" (marime a zonei tampon) la incarcare ca si la salvare. Comanda "F" actioneaza la ambele treceri.

## 10. EDITORUL

---

Este transparent pentru utilizator si comprima spatiile. Are urmatoarele functii:

ENTER - ENTER-ul de pe Spectrum  
 CC - CS+1 - renunta la intrare  
 CH - CS+0 - sterge inapoi  
 CI - CS+8 - avanseaza la urmatorul TAB  
 CX - CS+5 - renunta la linia introdusa

La cererea editorului semnalata cu ">" se poate raspunde cu o comanda de urmatoarea structura:

C N1,N2,S1,S2 si/sau ENTER

unde C reprezinta comanda de executat  
 N1,N2 reprezinta numere intre 1 si 32767  
 S1,S2 reprezinta siruri

## 11. COMENZILE EDITORULUI

---

### 11.1 Inserare de text

Textul poate sa fie inserat in fisierul sursa introducind un

numar de linie, un spatiu si apoi textul dorit, sau prin folosirea comenzii "I". Daca se scrie numai un numar de linie si apoi se apasa ENTER textul din linia respectiva se sterge. Ori de cite ori se introduce text se pot folosi functiile de control CX (sterge de la inceputul liniei), CI (trece la urmatorul TAB) si CC (intoarcere la bucla de comanda). Tasta DELETE va produce o stergere inapoi (nu inainte de inceputul liniei de text). Textul se introduce intr-un buffer intern al lui GENS, iar daca acesta se umple nu se mai poate introduce alt text si trebuie folosite functiile CH sau CX pentru a face loc in buffer. In timpul inserarii textului editorul detecteaza daca sfirsitul textului se apropie de virful RAM-ului, caz in care afiseaza mesajul "BAD MEMORY". Acest mesaj indica faptul ca nu se mai poate insera text si fisierul sursa sau cel putin o parte din el trebuie salvata pe banda pentru o redare ulterioara.

**Comanda I n.m.** Folosirea acestei comenzi determina intrarea in modul automat de inserare cu incepere de la linia n si incrementul m. Pentru a iesi din acest mod se foloseste functia CC (EDIT). Daca se specifica un numar de linie existent, aceasta va fi stearsa. Daca se depaseste 32767 modul de inserare automata se opreste. Daca introducind textul se ajunge la capatul ecranului cu o linie fara a introduce toate cele 64 caractere (marimea bufferului), ecranul va intra in mod defilare si se poate continua.

### 11.2 Listarea textului.

Se face cu comanda "L", iar modificarea numarului de linii ce se listeaza odata se poate face cu comanda "K".

**Comanda L n.m.** Listeaza textul de la linia n la linia m. Valoarea implicita a lui n este intotdeauna 1, iar a lui m este intotdeauna 32767 si nu valurile din comenzile precedente. Pentru a lista intreg textul se introduce "L" fara argumente. Linile se formeaza cu o margine la stanga. Tabularea liniei este automata, realizind o separare clara a diferitelor cimpuri. Numarul de linii listate dintr-o data pe ecran poate fi controlat cu ajutorul comenzii "K".

**Comanda K n.** Aceasta comanda stabileste numarul de linii care se afiseaza (sau listeaza) deoata inainte de pauza. Valoarea se inregistreaza in (n MOD 256). De exemplu K5 produce listarea a 5 linii deodata.

### 11.3 Editarea textului

O data ce textul a fost creat, va urma inevitabil nevoia de a edita o parte din el. Se pot folosi urmatoarele comenzi:

**Comanda D n.m.** Toate liniile de la n la m inclusiv sunt sterse din fisierul text. Daca n < m sau se specifica mai putin de doua argumente nu se executa nimic. (FALS! deoarece comanda se executa si in acest caz avind ca efect pur si simplu aparitia

unei noi linii cu numărul  $m$  și cu conținutul liniei  $n$  (Nota traducătorului)).

**Comanda N n,m.** Folosirea comenzii "N" face ca fișierul text să fie renumărat de la linia  $n$  cu incrementul  $m$ . Trebuie specificate atât  $n$  cât și  $m$ .

**Comanda F n,m,f.** Textul dintre liniile  $n$  și  $m$  este căutat după sirul  $f$ . Dacă acest sir este găsit, linia în care este găsit se va edita. În modul de editare se poate căuta următoarea apariție sau se poate în prealabil modifica textul și apoi să se treacă la următoarea apariție. Comanda își păstrează parametrii și pentru repetarea ei este suficient să se apese din nou F.

**Comanda E n.** Editează linia cu numărul  $n$ . Dacă  $n$  nu există, nu se execută nimic. În caz contrar linia se copiază într-un buffer și aici se poate lucra la ea, linia originală rămânând în tot acest timp neschimbată.

**Comanda M n,m.** Generează linia  $n$ , ce va avea conținutul liniei  $m$ .

### 11.3.1 Subcomenzi

**SPACE** - incrementează "text pointerul" (indicatorul de text) cu o unitate. Nu se poate depăși ultimul caracter.

**DELETE** - decrementează "text pointerul" cu o unitate, mergând înapoi pe linie. Nu se poate reveni înaintea ultimului caracter.

**CS+8** - salt înainte până la următorul TAB

**ENTER** - oprește editarea menținând toate modificările făcute.

**Q** - părăsește editarea ignorând toate modificările făcute.

**R** - reîncarcă bufferul cu text, ignorând modificările făcute.

**L** - listează restul liniei care se editează (de exemplu comentariul). Se rămâne în modul de editare cu "text pointerul" repositionat la începutul liniei.

**K** - șterge caracterul de pe poziția curentă a text pointerului.

**Z** - șterge toate caracterele de la (inclusiv) poziția curentă a text pointerului până la sfârșitul liniei.

**F** - găsește următoarea apariție a sirului definit cu comanda "F".

**S** - substituie sirul definit cu comanda "F".

**I** - insereaza caractere incepind cu pozitia curenta a text pointerului. Se ramine in acest mod pina la apasarea tastei ENTER cind se revine in modul de baza de editare cu text pointerul positionat pe ultimul caracter inserat.

**X** - avanseaza text pointerul la sfirsitul liniei si trece in modul de inserare.

**C** - permite rescrierea caracterului de pe pozitia curenta a text pointerului si apoi avansarea acestuia. Se ramine in acest mod pina la apasarea tastei ENTER cind se revine in modul de editare cu text pointerul positionat dupa ultimul caracter modificat.

## 12. COMENZILE CASETFONULUI

=====

**Comanda P n,m,s.** Domeniul dintre liniile n si m (inclusiv) e salvat pe banda cu numele de fisier specificat de sirul s. Argumentele isi pastreaza valoarea data de comanda precedenta.

**Comanda G,s.** Se cauta pe banda un fisier cu numele s. Cind acesta e gasit, se incarca la sfirsitul textului curent. Daca nu se specifica nici un nume se va incarca primul fisier de pe banda. Dupa ce se da comanda G apare mesajul "Start tape..."; se apasa PLAY la casetofon si incepe cautarea fisierului de pe banda. Pot sa apara mesajele "Using filename" sau "Find filename". Daca exista deja un text in memorie, cel de pe banda se va adauga la acesta.

**Comanda I n,m,s.** Aceasta comanda trimite blocul de text dintre liniile n si m pe banda intr-un format adecvat includerii ulterioare prin comanda asamblorului "\*F". Fila este salvata cu numele de fisier s. Trimiterea incepe imediat dupa apasarea tastei ENTER. Aceasta comanda nu se poate folosi ca inlocuitor al comenzii P.

### 13. COMENZILE MICRODRIVE-ULUI

Se pot utiliza exact aceleasi comenzi ca si la casetofon, cu deosebirea ca trebuie specificat si numarul microdrive-ului, iar numele fisierului este obligatoriu si la comanda G. Formatul comenzilor este:

**P n,m,nr1s                      G,,nr1s                      .T n,m,nr1s**

unde nr reprezinta numarul microdrive-ului. Toate celelalte observatii de la comenzile casetofonului ramin valabile, mai putin cele de la mesajele ce pot sa apara. In plus poate apare mesajul "File not found".

**NOTA:** Salvarea pe microdrive se face ca si bloc de date si ca pe banda unde se salveaza ca si cod masina. Se pot face inregistrari cu acelasi nume pe acelasi cartrige, dar se pare ca in acest caz inregistrarea veche se pierde. De asemenea daca se salveaza o rutina care are mici modificari fata de care exista deja pe cartrige, se pare ca sint salvate doar aceste modificari, iar numele sub care s-a salvat aceasta fila nu apare la comanda "CAT", desi in unele cazuri fila poate fi reincarcata cu comanda G. Formatul in care se scrie pe cartrige poate fi vizualizat din BASIC cu comanda 'MOVE "m";nr;"s" TO#2" (spre exemplu). Probabil ca si comanda "\*F" a asamblorului trebuie modificata in "\*F:nr". Consideram ca in aceasta sectiune aportul constructiv al cititorilor ar fi foarte bine venit.

### 14. ASAMBLAREA SI RULAREA DIN EDITOR

**Comanda A.** Asamblarea textului incepind de la prima linie.

**Comanda R.** Daca sursa a fost corect asamblata, fara erori si adresa de executie specificata in directiva ENT atunci comanda executa programul obiect. Daca programul obiect contine o instructiune de tip RET la sfirsitul executiei se poate face intoarcerea in editor atit timp cit nu s-a modificat pozitia stivei.

### 15. ALTE COMENZI

**Comanda B.** Reda controlul sistemului de operare. Reintrarea se face de regula 'cald', dar poate fi si rece.

**Comanda C.** Permite convertirea fisierelor text produse de GENS1 la forma comprimata a lui GENS3. Se incarca fisierul cu GENS1, se comprima si se salveaza cu T. Executia acestei comenzi este lunga si comanda nu are argumente.

**Comanda S.,d.** Permite schimbarea separatorului argumentelor intr-o linie de comanda. Separatorul nu poate fi spatiu si este implicit ";".

**Comanda V.** Afiseaza Ni,N2,S1,S2 cu valorile lor curente.

**Comanda W n,m.** Sectiunea de text dintre n si m este listata la imprimanta. Listarea se face conform comenzii K, iar listarea se reactiveaza dupa apasarea oricarei taste.

**Comanda X.** Afiseaza in zecimal adresele de inceput si sfirsit ale fisierului text.

## 16. CODUL ERORILOR

---

- 1 - eroare in contextul liniei
- 2 - mnemonic necunoscut
- 3 - instructiune formulata gresit
- 4 - simbol multidefinit
- 5 - linia contine caracter ilegal
- 6 - operand ilegal
- 7 - un simbol este cuvint rezervat
- 8 - mitchmetch in registre
- 9 - prea multe registre
- 10 - expresie > 8 biti
- 11 - instructiuni JP (IX+n), JP (IY+n) ilegale
- 12 - eroare in formarea unei directive
- 13 - referinta ilegala (EQU cu un simbol inexistent)
- 14 - impartire cu 0
- 15 - depasire la inmultire

Bad ORG - directiva ORG ar duce la stricarea fisierului text sau a tabelii de simboluri.

Out of table space (No table space) - nu s-a alocat suficienta memorie pentru tabela de simboluri.

Bad memory - nu mai este loc pentru text.

## 17. CUVINTE REZERVATE

---

A, B, C, D, E, H, L, I, R, \$, AF, AF', BC, DE, HL, IX, IY, SP, NC, +Z, NZ, M, P, PE, PO.

## MONS MONITOR-DESASSEMBLER (©) 1983 HISOFT

---

Adaptare Aurel Gontean  
Introdus Adriana Gontean

MONS este furnizat într-o forma relocabilă. Se încarcă pur și simplu la adresa de la care se dorește executarea și se lansează în execuție. Dacă se dorește relansarea lui (din BASIC), atunci se execută de la o adresă cu 29 (zecimal) mai mare decât adresa originală. MONS are 5K lungime odată ce a fost relocat, dar consumă aproape 6K pentru încărcare din cauza "tabloului de relocatari" a adreselor care urmează după codul principal. MONS conține propria sa stivă, fiind un program de sine statator. La intrarea în MONS, adresele sunt afișate prin "PANOUL FRONTAL" în hexazecimal.

Aceste adrese se pot converti în zecimal cu comanda SS+3. Cu toate acestea, adresele trebuie introduse întotdeauna în hexazecimal. Comenzile se introduc de la claviatura în urma cursorului ">" sub forma de litere mari sau mici. Unele comenzi a căror efect poate fi dezastruos dacă sunt utilizate greșit, necesită apăsarea lui SS împreună cu litera de comandă. Comenzile se execută imediat și nu este nevoie de încheierea lor cu ENTER.

Unele comenzi necesită introducerea de numere hexazecimale.

Dacă terminatul este semnul "-" atunci se introduce forma negativă în complementul lui 2. Dacă se introduce mai mult de 4 caractere, atunci doar ultimele 4 se iau în considerare. În orice moment se poate întoarce la interpretorul BASIC prin apăsarea lui CS+1.

**NOTA IMPORTANTA:** MONS dezactivează întreruperile; utilizatorul trebuie să se asigure că întreruperile nu sunt validate în timpul lucrului cu MONS.

### 1. COMENZI

---

SS+3 comută baza de numeratie a adreselor afișate (hexazecimal). Aceasta comandă afectează toate adresele afișate de MONS, inclusiv pe cele generate în cursul dezasmblării, dar nu schimbă afișarea conținutului memoriei care se face întotdeauna în hexazecimal.

SS+4 (sau \$) afișează o pagină dezasmblată începând de la adresa conținută în "MEMORY POINTER". La a 2-a apăsare se face reîntoarcerea la "PANOUL FRONTAL" și orice altă tastă continuă cu următoarea pagină de dezasmblat.



ENTER incrementeaza "MEMORY POINTER" cu 1, astfel incit cei 24 de octeti afisati sint centrati in jurul unei adrese mai mari cu 1 decit cea precedenta.

CS+7       decrementeaza "MEMORY POINTER" cu 1  
 CS+5       decrementeaza "MEMORY POINTER" cu 3  
 CS+8       incrementeaza "MEMORY POINTER" cu 3

, modifica "MEMORY POINTER" [MP] astfel incit sa contina adresa curenta a stivei (indicata de SP). Aceasta comanda este utila cind se doreste o privire de ansamblu asupra adreselor de intoarcere a rutinelor chematoare.

"G"       cauta in memorie un anumit sir (GET). Se afiseaza ";", si se introduce primul octet care se cauta urmat de ENTER si tot asa in continuare pina cind se defineste intreg sirul.

"H"       converteste un numar zecimal in forma sa echivalenta hexazecimala.

"I" (copie inteligenta) este folosit pentru copierea unui bloc de memorie de la o locatie la alta. Este inteligenta in sensul ca blocul de memorie poate fi copiat in locatii unde s-ar suprapune peste locatiile precedente. "I" raspunde intrebind "FIRST:", "LAST:", care sint adresele (inclusiv) de inceput si sfirsit ale blocului care trebuie copiat si apoi "TO:" care reprezinta adresa la care blocul trebuie copiat.

Daca adresa de start este mai mare decit adresa de sfirsit, comanda nu se executa. Numerele se introduc in forma hexazecimala.

"J"       executa codul de la adresa specificata. Comanda raspunde cu ";", asteptind un numar in hexazecimal care odata introdus, duce la resetarea stivei interne, ecranul este sters si executia transferata la adresa specificata. Daca se doreste reintoarcerea la "PANOUL FRONTAL" dupa executarea codului, se alege un punct de intrerupere cu comanda "W" in locul la care se doreste intoarcerea la display.

!!! "J" modifica registrele Z80 inainte de a executa codul. Daca se doreste executarea codului cu anumite valori ale registrelor, se executa "SS+K" de mai jos.

"SS+K"   continua executia de la adresa curenta a lui PC. Aceasta comanda s-ar utiliza cel mai frecvent impreuna cu "W".

"L"       tabelleaza sau listeaza un bloc de memorie incepind de la adresa curenta continuata in MP.

Comanda sterge ecranul si afiseaza reprezentarea hexazecimala si echivalentul ASCII a 80 de octeti de memorie incepind de la valoarea curenta a MP. Adresele se afiseaza in hexazecimal sau zecimal in functie de starea lui "SS+3".

Display-ul este format din 20 de rinduri a 4 octeti/rind si codurile ASCII la sfirsitul rindului.

Pentru aceasta, orice valoare peste 127 este decrementata cu 128 si orice valoare intru 0 si 31 este afisata ca ".".

La sfirsitul paginii se poate face reintoarcerea la "PANOUL

**FRONTAL**" cu **"CS+5"** sau continua cu orice alta tasta.

**"M"** seteaza MP pe adresa specificata. Raspunde cu **;** si asteapta un numar in hexazecimal. MP este actualizat cu **?** esa introdusa si afisarea **"PANDULUI FRONTAL"** se modifica corespunzator.

**"N"** gaseste urmatoarea aparitie a sirului specificat de comanda **"G"**. (Comanda **"G"** permite definirea unui sir si cautata prima sa aparitie.) Cautarea incepe de la MP si actualizeaza ecranul la aparitia sirului cautat.

**"O"** merge la un deplasament relativ. Comanda ia octet-ul adresat de MP si il trateaza ca un deplasament relativ, actualizand ecranul corespunzator. Aceasta comanda ne permite sa vedem unde face saltul o instructiune JR. De retinut ca deplasari relative mai mari decit #7F (127) sint tratate ca negative de microprocesorul Z80, fapt de care comanda **"O"** tine cont (vezi comanda **"U"**).

**"P"** umple memoria dintre limitele specificate cu un octet specificat.

Intreaba **"FIRST:"**, **"LAST:"** si **"WITH:"**. Se introduc numerele in forma hexazecimala ; adresa de inceput si de sfirsit a blocului de memorie care trebuie umplut si octetul cu care se doreste umplerea blocului de memorie.

**"Q"** comuta setul de registre. La intrarea in **"PANDUL FRONTAL"**, setul de registre afisat este cel standard (AF, HL, DE, BC).

Folosirea lui **Q** va afisa setul alternativ de registre (AF', HL', DE', BC') care se poate distinge de cel standard prin semnul **'** dupa numele registrului.

Daca **"Q"** este folosit cind se afiseaza registrul alternativ, se trece la cel standard.

**"SS+T"** stabileste un punct de intrerupere dupa instructiunea curerta si continua executia. Este folosita, in general, atunci cind, executind pas cu pas o secventa de program si ajungind la un CALL, dorim ca subrutina sa fie executata rapid.

**"T"** dezassembleaza o portiune de cod, optional la imprimanta. Prima data intreaba **"FIRST:"** si **"LAST:"** adresele codului pentru care se doreste dezasamblarea in hexazecimal. Daca adresa de inceput este mai mare decit adresa de sfirsit, comanda nu se executa. Dupa introducerea acestor adrese se intreaba **"PRINTER ?"**. Se raspunde cu **Y** (litera mare) daca se doreste ca dezasamblarea sa fie directionata la canalul de imprimanta sau cu orice altceva daca iesirea se face pe ecran; urmeaza intrebarea **"TEXT ?"** pentru a introduce in hexazecimal adresa de start a fisierului text pe care il produce dezasamblarea. Daca nu se doreste generarea unui fisier text, se apasa **ENTER** la aceasta intrebare. Fisierul se produce intr-o forma in care poate fi utilizata de **GENS3**. Daca se doreste utilizarea textului cu **GENS3**, trebuie generat la sau mutat la prima adresa data de comanda editorului **"X"**, deoarece-

ce aceasta este adresa de start a "fisierului text" asteptat de GENS3. Trebuie specificata si adresa de sfirsit a textului. Aceasta se face luind adresa "End of text" data de dezasambloare si punind-o in locatia TEXTEND a lui GENS3 (xxxxx+54, de exemplu, la GENS3M21- unde xxxxx=adresa de incarcare). Se intra apoi in GENS "la cald", pentru a pastra textul. Daca generind fisierul text, acesta s-ar scrie peste MONS, dezasambloarea este abandonata - se apasa orice tasta pentru revenirea la "PANOUL FRONTAL".

Daca se specifica o adresa pentru fisierul text, se intreaba in continuare "Workspace" - adresa spatiului de lucru care ar trebui sa fie inceputul locului gol al memoriei care este folosit pentru o tabela de simboluri pentru orice eticheta generata in procesul dezasambloarii (cantitatea de memorie necesara este de 2 octeti pentru orice eticheta generata). Daca se apasa ENTER, adresa subinteleasa este #6000 in hexazecimal.

Apoi se intreaba in mod repetat "FIRST:" si "LAST:" adresele (inclusive) pentru blocurile de date care la dezasambloare vor fi interpretate ca DEFB si nu ca si instructiuni Z80.

Daca valoarea octetului de date este intre 32 si 127 (#20 si #7F) inclusiv, atunci se va interpreta respectivul octet ca si un caracter ASCII. Daca s-au terminat specificarile sau daca nu se specifica nici un domeniu, se apasa ENTER pentru ambele intrebari.

Comanda "T" foloseste domeniul de la sfirsitul lui MONS pentru a imagazina adresele domeniilor de date, astfel incit se pot alege atatea domenii de date cita memorie exista disponibila; fiecare domeniu de date necesita 4 octeti pentru inmagazinare. De remarcat ca aceasta comanda distruge punctele de intrerupere (comanda "W").

In acest moment ecranul va fi sters. Daca s-a cerut crearea fisierului text, va urma o scurta intirziere (dependenta de marimea sectiunii de memorie care trebuie dezasambolata), cauzata de construirea tabelului de simboluri; aceasta odata facuta, listingul dezasamblat va apare pe ecran sau la imprimanta - listarea poate fi intrerupta cu tastele ENTER sau SPACE, respectiv CS+5 pentru intoarcerea la "PANOUL FRONTAL" sau orice alta tasta (mai putin CS+1) pentru a continua dezasambloarea. Daca un "opcode" invalid este intilnit, se dezasambleaza ca un NOP si este urmat de un "\*" dupa "opcode"-ul din listing.

La sfirsitul dezasambloarii ecranul va astepta si daca s-a cerut producerea unui fisier text, se afiseaza mesajul "End of text xxxxx", unde xxxxx este adresa in hexazecimal sau zecimal care trebuie POKE-ata in GENS3 la locatia TEXTEND pentru ca asamblorul sa poata "culege" adresa fisierului text la un start cald. Cind s-a incheiat se apasa o tasta pentru intoarcere la "PANOUL FRONTAL" sau CS+1, cu care se revine in BASIC.

Etichetele sint generate (cind sint relevante) in forma Lxxxxx unde xxxxx este adresa absoluta in hexazecimal a etichetei, doar daca respectiva adresa se afla in limitele dezasambloarii. Daca adresa se afla in afara acestui domeniu, nu se genereaza nici o eticheta ci se da pur si simplu adresa zecimala sau hexazecimala. Daca o anume adresa s-a referit la o instructiune in dezasambloare, atunci eticheta sa va apare in cimpul etichetei (inainte de mnemonic) doar daca listingul este directionat la un fisier text.

"U" folosit impreuna cu comanda "O".

Dupa cum s-a precizat, comanda "O" actualizeaza ecranul corespunzator unui deplasament relativ (de regula pentru a arata efectul unei instructiuni de tip JR sau DJNZ). "U" este folosit pentru a actualiza ecranul cu valorile dinaintea executarii comenzii "O" (a ultimei comenzi "O").

"V" folosit impreuna cu comanda "X".

"V" este similara comenzii "U", cu deosebirea ca actualizeaza ecranul acolo unde era inainte de executia ultimei comenzi "X".

"W" determina un breakpoint (punct de intrerupere) la adresa MP. Un punct de intrerupere din punct de vedere al lui MONS - este o instructiune de tip CALL a unei subrutine care afiseaza PANoul FRONTAL permitind programatorului sa opreasca executia programului, sa inspecteze registrele, flag-urile (indicatorii de conditie) si orice adrese relevante. Cei trei octeti ai instructiunii CALL sint inlocuiti cu cei originali indata dupa executia opririi.

MONS foloseste ultimele locatii de memorie pentru aceasta actiune si deci se pot alege atitea puncte de intrerupere, cita memorie disponibila exista. Deoarece comanda T utilizeaza de asemenea aceasta zona de memorie, toate punctele de intrerupere vor fi pierdute dupa o comanda T. Fiecare punct de intrerupere necesita 5 octeti pentru memorare.

**ATENTIE !** Locul unui punct de intrerupere trebuie bine ales pentru a nu patrunde in corpul unei instructiuni pe mai multi octeti, caz in care evident nu se va ajunge la punctul de intrerupere scontat.

"X" folosita pentru a actualiza MP cu destinatia unei instructiuni de tip CALL sau JP de tip absolut. "X" ia o adresa (pe 16 biti) specificata de la MP si MP+1 si apoi actualizeaza ecranul astfel incit sa fie centrat in jurul acestei adrese. De retinut ca primul octet este cel mai putin semnificativ (Vezi comanda "V").

"Y" introduce ASCII de la MP. "Y" da o noua linie in care se pot introduce caractere ASCII direct de la tastatura. Acestea se introduc in forma hexazecimala echivalenta in memorie incepind de la adresa curenta a lui MP. Sirul de caractere trebuie terminat cu CS+5, iar DELETE (CS+0) poate fi folosit pentru a sterge caracterele din sir. Cind s-a terminat introducerea caracterelor si a lui CS+5, ecranul se actualizeaza astfel incit MP este positionat imediat dupa sfirsitul sirului in memorie.

"SS+Z" executie pas cu pas. Inainte de folosirea acestei comenzi atat PC cit si MP trebuie stabilite la adresa instructiunii care se doreste executata. Comanda executa instructiunea curenta si actualizeaza "PANoul FRONTAL" pentru a reflecta schimbarile determinate de executia instructiunii. Se poate merge pas cu pas in RAM si in ROM dar aceasta doar daca intreruperile nu sint nicaieri activate.

"\*" (SS+P). Aceasta comanda are exact acelasi efect ca si "L" (List), cu deosebirea ca iesirea se face la canalul imprimantei in loc de er-an. De retinut ca la sfirsitul paginii se apasa CS+5 pentru intoarcerea la "PANOUL FRONTAL" sau orice alta tasta pentru urmatoarea pagina (mai put'in CS+1).

## 2. MODIFICAREA MEMORIEI

=====

Continutul adresei dat de MP poate fi modificat prin introducerea numarului hexazecimal urmat de un terminator. Daca terminatorul nu este valid, comanda nu se executa.

## 3. MODIFICAREA REGISTRELOR

=====

Daca un numar hexazecimal este introdus ca raspuns la prompterul ">" si este terminat cu ".", atunci numarul specificat va fi introdus in registrul Z80 adresat curent de sageata "->". Introducand "." ca terminator al numarului hexazecimal, se va modifica PC. Daca se indica doar "." (deci nu pe post de terminator) pointerul se va roti circular de la PC la AF. Nu este posibila adresarea (deci nici schimbarea) indicatorului de stiva SP sau a registrului IR. "." poate fi folosit pentru modificarea setului alternativ de registre, daca acesta este afisat. Se va folosi comanda "Q" pentru a comuta setul de registre,

## 4. PANOUL FRONTAL

=====

Primele 9 linii ale ecranului contin registrele Z80, numele lor, valoarea lor actuala si continutul a 7 locatii de memorie incepind de la adresa specificata de registru. Registrul F al indicatorilor de conditie este afisat decodificat.

Pointerul de registru "->" indica registrul adresat curent. Dei 24 de octeti afisati in partea de JOS sint organizati ca adrese urmate de continutul lor, centrate in jurul valorii marcate >...<.

NOTA FINALA: S-au depus toate eforturile ca materialul prezentat sa fie clar si corect. In cazul in care s-au strecurat erori de traducere sau de formulare, va rugam sa luati legatura cu ALPHA Ltd. Va asteptam! (Tel. 961/12936)



## GENS ASSEMBLER (®) 1983 HISOFT

1.	INTRODUCERE . . . . .	2
2.	DETALII . . . . .	3
3.	FORMATUL INSTRUCTIUNII. . . . .	5
4.	CONTORUL DE LOCATII . . . . .	6
5.	TABELA DE SIMBOLURI . . . . .	6
6.	EXPRESII. . . . .	6
7.	DIRECTIVELE ASAMBLORULUI. . . . .	7
8.	PSEUDOMNEMONICE CONDITIONATE. . . . .	7
9.	COMENZILE ASAMBLORULUI. . . . .	7
10.	EDITORUL. . . . .	8
11.	COMENZILE EDITORULUI. . . . .	8
	11.1 Inserare de text. . . . .	8
	11.2 Listarea textului . . . . .	9
	11.3 Editarea textului . . . . .	9
	11.3.1 Subcomenzi . . . . .	10
12.	COMENZILE CASETOFONULUI . . . . .	11
13.	COMENZILE MICRODRIVE-ULUI . . . . .	11
14.	ASAMBLAREA SI RULAREA DIN EDITOR. . . . .	12
15.	ALTE COMENZI. . . . .	12
16.	CODUL ERORILOR. . . . .	13
17.	CUVINTE REZERVATE . . . . .	13

## MONS MONITOR-DESASSEMBLER (®) 1983 HISOFT

1.	COMENZI . . . . .	14
2.	MODIFICAREA MEMORIEI. . . . .	19
3.	MODIFICAREA REGISTRELOR . . . . .	19
4.	PANoul FRONTAL. . . . .	17

