

# OPERATING USER SYSTEM

## CUPRINS

- 1 Introducere
  - 2 Optiuni principale
  - 3 Comenzi OPUS
  - 4 Monitor OPUS
    - 4.1 Utilizare generala
    - 4.2 Dezasamblor OPUS
    - 4.3 Rularea codului masina
    - 4.4 Comenzile monitorului
  - 5 Asamblorul OPUS
    - 5.1 Introducerea si editarea textelor
      - 5.1.1 Formatul general
      - 5.1.2 Constantele
      - 5.1.3 Operatori
      - 5.1.4 Expresiile
      - 5.1.5 Directivele de asamblare
      - 5.1.6 Ecran-editorul
    - 5.2 Comenzile asamblorului
    - 5.3 Stocarea fisierelor
  - 6 BASIC DECODER-ul
  - 7 Utilizarea imprimantei
  - 8 Harta memoriei OPUS
  - 9 Variabilele sistem
  - 10 Subrutinele OPUS utile
    - 10.1 Afisari pe ecran
    - 10.2 Conversii
    - 10.3 Teste
    - 10.4 Citiri de la tastura
    - 10.5 Lucrul cu caseta
    - 10.6 Sunete
    - 10.7 Subrutine speciale
  - 11 Copierul OPUS
- Anexa 1: Codurile rapoartelor de eroare  
Anexa 2: Cuvinte rezervate

## 1 Introducere

OPUS contine un asamblor de mare viteza in doi pasi, proiectat spre a permite reprezentarea simbolica a codului masina, un dezasamblor in doi pasi, capabil sa genereze fisiere sursa si un monitor bazat pe un ecran-editor. Aceasta inseamna ca se pot edita programe in limbaj de asamblare Z-80, modifica texte afisate pe ecran, trece in revista si manipula locatiile de memorie sau se pot face unele calcule hexazecimale.

## 2 OPTIUNILE PRINCIPALE

- 1-Monitorul
- 2-Asamblorul
- 3-Copiere de pe caseta

## 3 COMENZILE OPUS

O comanda consta dintr-o litera, de obicei urmata de pina la 3 parametri. Parametrii numerici constau dintr-o constanta zecimala sau hexazecimala. Introducerea parametrilor numerici modifica parametrii predefiniti pentru comanda. Daca vrem sa modificam al doilea sau al treilea parametru, cei anteriori raminand nenodificati, se introduce o virgula pentru fiecare parametru nemodificat. Ca separator intre constante se utilizeaza blankul. Predefinit, monitorul considera parametrii ca fiind hexazecimali, iar asamblorul ii considera zecimali. Pentru a indica tipul invers de parametru, se prefixeaza parametrii zecimali cu ., iar cei hexazecimali cu #. Parametrii sir de caractere sint reprezentati prin inchiderea sirului intre delimitatori. Acestia pot fi orice caracter cu exceptia: cifre, virgula.

## 4-MONITORUL OPUS

### 4.1 Utilizare generala

Cu aceasta parte a OPUS-ului se poate trece in revista si modifica direct continutul memoriei. Trebuie avut in vedere ca in acest mod parametrii sint hexazecimali, asa incit trebuie avut in vedere ca parametrii zecimali trebuie prefixati cu . . Cu comanda "R" poate fi vizualizat continutul registrelor utilizator la reintoarcerea din executia codului masina.

### 4.2 Dezasamblorul OPUS

Cu comanda "D" urmata de un parametru, continutul memoriei este dezasamblat si afisat pe ecran in mod uzual. Cu ajutorul comenzii "N" se pot crea fisiere sursa pentru asamblorul OPUS. Comanda nu necesita parametri. Acestia vor fi furnizati ulterior la cererea dezasamblorului. OPUS va cere mai intii zona de dezasamblat prin "FIRST:" si "LAST:". Daca se apsa "ENTER", predefinit se considera 0000 si nu se va intimpla nimic. Dupa aceasta se va cere "DISP:" pentru a genera sursa care sa ruleze de la alte adrese decit cea de la care este dezasamblat codul. Se va introduce deplasarea relativa adresei de rulare fata de adresa de la care ruleaza codul. Prin "ENTER" se considera deplasare 0 la fisierul sursa generat. Daca exista un alt fisier si nu se doreste distrugerea lui, trebuie aflat intii sfirsitul lui. Apasind doar "ENTER" nu se va genera fisier sursa.

"WORKSPACE:" necesita o adresa pentru spatiul de lucru al dezasamblorului. Apasind "ENTER" va fi considerata valoarea

#F000. Urmatoarele perechi de "FIRST:" si "LAST:" marcheaza portiunile de memorie ce nu vor fi dezasamblate, considerandu-se a fi zone de memorie adatele. Pot exista mai multe astfel de zone, asa incit pentru iesirea din aceasta bucla trebuie raspuns cu "ENTER" la ambele cereri. Ultima cerere este "LIST?". La raspunsul "Y" un listing al codului dezasamblat va fi afisat pe ecran in timpul dezasamblarii. Raspunsul cu orice alta tasta nu va genera listing. Cind se va termina dezasamblarea, adresa de sfirsit sfisierului sursa generat va fi afisata (utila in concatenarea cu un alt text dezasamblat).

Textul sursa generat poate fi acum manipulat cu asamblorul OPUS dupa cum este scris in cap.5. Etichetele vor fi marcate cu "L" si adresa originala (#E000). Toate salturile absolute sau relative (JF sau JR) si apelurile de subrutina (CALL) vor avea etichete in loc de adresa, considerind ca toate adresale la care se sare sint in interiorul blocului dezasamblat.

#### 4.3 Rularea codului masina

Codul masina poate fi rulat cu comanda "E". codul va fi rulat dupa ce din registrele utilizate se incarca in registrele microprocesoarelor. Reintoarcerea se face cu un simplu RET (#C9). Aceasta inseamna ca nu trebuie modificat continutul stivei (se scoate tot ce se introduce pe stiva). Daca continutul stivei trebuie distrus, reintoarcerea se face cu RSTB (#CF). O posibilitate mai speciala pentru programelor in cod este posibilitatea definirii unui punct de intrerupere la intilnirea caruia executia se va suspenda si va fi afisat continutul registrelor generale ale microprocesorului. Acum se poate proceda in doua moduri :  
cu "Q" se opreste executia si se va transfera controlul monitorului;

cu "C" se va continua executia pina la intoarcerea in monitor.

Daca se foloseste subrutina de tiparire caracter pe ecran RST#10, trebuie ca registrul IY sa contina valoarea #5C3A.

#### 4.4 Comenzile monitorului

AXY=#aritmetica hexa si conversie zecimal-hexa. Ambii parametri sint afisati atat in zecimal cit si in hexazecimal. Se mai afisaza in hexa X+Y si X-Y. Predefinit avem X=0, Y=0.

BX= se va pune punctul de intrerupere la adresa X. X=0 nu pune punct de intrerupere. Predefinit X=0.

CXYZ= copiaza blocul de memorie dintre X si Y de la adresa Z mai departe. Predefinit X=0, Y=0, Z=0.

DXYZ= dezasambleaza codul dintre X si Y cu pauza dupa fiecare Z linie afisata. Predefinit X=0, Y=#FFFF, Z=14.

EX= executa codul de la adresa X. Predefinit X=0.

FXYZ(...)= cauta intre X si Y aparitia secventei (...) cu pauza dupa Z secvente gasite. (...) poate fi :

a) oriciti octeti hexa separati printr-un blank;

b) un sir ASCII inchis intre ghilimele;

c) amestec intre a) si b).

Predefinit X=0, Y=#FFFF, Z=14.

GXYZ= incarca cod de pe caseta, memoreaza la adresa X. Lungimea zonei de cod este Y iar Z octetul de indicator

Predefinit X=0, Y=0, Z=0.

H= citește și interpretează "HEADERE" în format SPECTRUM.

IXYZ= umple memoria (inserează) între adresele X și Y octetul Z. Predefinit X=#4000, Y=#57FF, Z=0 (sterge ecranul).

JXY= deplasamentul unui salt relativ de la X la Y va fi afișat. Predefinit X=0, Y=0.

KXYZ= se aduce borderul la X, fondul la Y și scrisul la Z. Predefinit X=0, Y=0, Z=7.

MX= o pagină de memorie va fi afișată atât în hexa cit și în ASCII cu un cursor începând de la adresa X. Predefinit X=0.

Comenzi de deplasare a cursorului pe ecran:

DELETE-muta cursorul înapoi;

ENTER -muta cursorul înainte;

SS+S -afișează pagina următoare;

SS+W -afișează pagina următoare;

SS+CS -comută cursorul între zona ASCII și zona HEXA.

N=dezasamblor generator de fișiere sursă (descrie în 4.2.)

PXYZ=inregistrează cod pe caseta, blocul de memorie cu lungimea Y și începând de la adresa X va fi salvat pe caseta cu octetul de indicatorul Z. Predefinit X=0, Y=0, Z=#FF.

Q= ieșire în pagina inițială de opțiuni.

R= se afișează registrele utilizator și se pot înlocui

S= comută activat / dezactivat sunetul la tastarea unei taste

UXYZ= controlul este transferat la o rutină utilizator care începe la adresa de variabilă sistem USERAD (#5C5C). Predefinit X,Y,Z=0.

XYZU= schimbă octetul Z cu octetul U între adresele X și Y. Predefinit X,Y,Z,U=0.

YX= fixează indicatorul de stivă care în mod normal 0 începe la #6000. Predefinit X=#5FFD.

ZXY= umple memoria cu octetul #00 între adresele X și Y. Predefinit X=#4000, Y=#57FF (sterge ecranul)

## 5. ASANBLORUL OPUS

### 5.1 Introducerea și editarea textelor

#### 5.1.1 Formatul general

OPUS utilizează setul de caracter ASCII. Pentru a introduce instrucțiuni ale limbajului de asamblare se va introduce mai întâi un număr de linie urmat apoi de textul dorit. Textul poate consta dintr-una sau mai multe instrucțiuni separate prin ":". De obicei instrucțiunile cuprind:

a) o etichetă opțională. Etichetele pot conține litere mari, mici și cifre. Trebuie să înceapă cu o literă; pot avea șase caractere în lungime; nu trebuie să coincidă cu un cuvânt rezervat

b) o instrucțiune în limbaj de asamblare. Instrucțiunile pot fi standard Z-80 sau directive de asamblare. O facilitate este oferită în referirea la indicatorul de paritate/depasire: se poate utiliza JPV, ...(JPNV,...) în loc de JPFE,...(JPOD,...)

c) comentariile pot fi anexate cu ajutorul ";" si sint permise linii continind doar comentarii.

#### 5.1.2. Constantele

Constantele sint recimale ,dar pot fi exprimate in hexa prefixindu-le cu "#". Constantele literale pot fi exprimate prin includerea lor intre ghilimele:"C" de exemplu. Exista o constanta de sistem notata cu \$ care intotdeauna ia valoarea adresei curente la care se asambleaza: DONS \$x=> LOOP DONS LOOP.

#### 5.1.3. Operatorii

Operatorii posibili sint :+ = adunarea; - = scaderea; &=AND logic; !=OR logic. Nu exista prioritatea unui operator fata de altul.Expresiile sint evaluate strict de la stinga la dreapta.

#### 5.1.4. Expresiile

Oriunde este necesara o constanta intr-o instructiune ,o expresie poate fi folosita in loc. Expresiile sint construite din etichete si/sau constante separate de operatori. A se avea in vedere ca expresiile sint calculate la asamblare si nu in timpul rularii codului.

#### 5.1.5 Directivele de asamblare

Urmatoarele pseudoinstructiuni ofera parametrii necesari operarii DPUS-ului sau comanda asamblorului sa stocheze ori ca simboluri ori direct in memorie.

ORGnnnn= indica adresa de la care va fi asamblat in continuare codul .Sint permise mai multe ORG-uri .

DISPnnnn= modifica locul in care este generat codul dar acesta va rula de la adresa indicata de ORG.

ENT= comanda"X" va introduce in executie codul de la ultima directiva ENT din fisierul sursa .

EQU= da o valoare unui simbol (de ex.ONE EQU 1)

DEFSnnnn=insereaza numarul specificat de locatii libere la adresa curenta de asamblare.

DEFBnn.nn,...= insereaza octetii specificati la adresa curenta de asamblare.

DEFWnnnn,...= insereaza cuvintele specificate la adresa curenta de asamblare.

DEFM"... "= insereaza textul ASCII dintre ghilimele.

#### 5.1.6 Ecran-editorul

Cursorul poate fi mutat pe ecran cu ajutorul cursoarelor. Orice caracter poate fi inlocuit direct prin simpla apasare a noului caracter cind cursorul se afla sub el. Pot fi inserate spatiila pozitia curenta a cursorului; dar la fiecare spatiu se pierde un caracter de la sfirsitul liniei. acest lucru se obtine cu CS+4. caracterul sub care se afla cursorul poate fi sters cu CS+3. Dupa modificarea unei linii trebuie apasat ENTER pentru a fi memorata. CS+9 sterge ecranul si pune cursorul in stinga sus. CS+1 strge linia curenta si pune cursorul la inceputul ei. CS+2 muta cursorul la urmatoarea pozitie de tabulare din cadrul liniei curente.

## 5.2 COMENZILE ASAMBLORULUI

Ax= asambleaza fisierul sursa si afiseaza mesajele de eroare. Dupa x erori asamblarea se opreste. Pentru a continua se apasa pe ENTER.Predefinit x=14.

Cxyz= copieaza blocul incepind de la linia x pina la linia y, de

la linia z incolo. O renumerotare cu incrementul va avea loc automat. Predefinit x,y,z=0.

Dxy= sterge toate liniile dintre x si y, daca nu se dau numere de linie nu se va intimpla nimic.

E= executa codul masina produs de ultima asamblare. Executia va incepe de la ultimul ENI intilnit in fisierul sursa, daca nu exista nici-un ENI, se va afisa un mesaj de eroare.

F "string 1 string 2"xyz= se cauta apartiile "string 1" dintre liniile x si y. Daca "string 2" nu este dat, se afiseaza toate apartiile cu pauza dupa z apartii. Daca "string 2" este dat la fiecare apartie a lui "string 1" o portiune din fisierul sursa este afisata cu linia in care sa intilnit scoasa in evidenta. Daca se apasa pe "Y" "string 1" va fi inlocuit cu "string 2". Daca se apasa "Q" nu se mai continua cautarea, apasind orice alta tasta continua cautarea. Predefinit: x=prima linie, y=ultima linie din fisier, z=14, "string..."="".

Ixy=se genereaza automat numere de linie incepind cu x si cu increment y. Predefinit x=10, y=10. Pentru a parasi aceasta comanda se sterge ultimul numar generat si se apasa ENTER.

Kxyz= ca in monitor.

Lxyz= se listeaza sursa dintre liniile x si y cu pauza dupa z linii. Predefinit z=14.

Mxyz= acelasi lucru ca si la comanda "C", dar sterge liniile originale (mota nu copiaza).

nx= creaza un fisier sursa nou la adresa x. Predefinit x=24576=#6000.

Ox= face ca un fisier anterior creat la adresa x sa devina fisier curent. Predefinit x=24576=#6000.

Q= iesire din pagina initiala.

Rxyz= se numereaza liniile de la z incolo, prima valoare fiind x si increment y. Predefinit x=10, y=10, z= prima linie.

Sx= daca nu este dat x se afiseaza adresa de start, adresa de sfirsit si dimensiunea fisierului curent. Altfel se da adresa de start a liniei x.

Tx= label name= daca nu se da al doilea parametru, se afiseaza tabela de simboluri cu pauza dupa x simboluri afisate. Funind un nume de etchieta valoarea ei hexa va fi afisata. Predefinit x=14.

Uxy= la fel ca la monitor.

Xx= se vor tipari toate numerele de linie in care apare o etchieta, pentru fiecare etchieta in parte. Linia in care este definita etchieta este marcata de simbolul "!". Se vor afisa x etchiete inainte de pauza. Predefinit x=6.

### 5.3 STOCAREA FISIERELOR

Fisierele sint salvate pe banda in format compatibil ZX-SPECTRUM si sint memorate ca zona de cod. Daca la incarcarea unui fisier mai exista unul creat in calculator si facut curent se va intimpla ce se intimpla in BASIC cu instructiunea MERGE si in plus se va face o renumerotare cu increment 1. Comenzile de lucru cu caseta sint:

-G string= incarca fisierul cu numele "string" de pe caseta. Daca nu se precizeaza "string" se incarca primul fisier intilnit

-P string= salveaza pe caseta fisierul curent sub numele "string"

daca acesta este precizat. Daca nu, se foloseste ca nume un blank  
La sfirsit se afiseaza lundimea si adresa de start a fisierului  
salvat.

-Vstring= se verifica fisierul de pe caseta avind numele "string"  
Daca "string" nu e precizat se verifica primul fisier intilnit.

#### 6. BASIC DECODER-ul

DFUS contine si un decodor pt. BASIC. (nu e interpretor !). Dupa  
incarcarea la orice adresa a unui fisier BASIC cu comanda monitor  
"Q", se poate rula DECODER-ul BASIC indicind intii adresa de  
start. Intii vor fi afisate lungimea si nr. de linii. Mai departe  
DECODER-ul asteapta apasarea unei taste. apasind "Q" se va parasi  
DECODER-ul. Orice alta tasta va continua decodarea. Codurile  
neinterpretabile sint scoase in evidenta prin mesajul "INVALID!"  
Codurile de control sint incadrate de "\*" si "#". Numerele sint  
afisate mai intii asa cum apar in listing, urmate de valoarea  
reala in hexa si zecimal intre paranteze, asa cum este codificata  
de interpretorul BASIC dupa octetul #OE. Sfirsitul de linie BASIC  
e marcat cu codul de control ENTER si cu o linie continua pe  
ecran. Atentie ca: decodorul nu recunoaste codurile Z-80 din REM

#### 7. UTILIZAREA IMPRIMANTEI

Prin apasarea tastelor SS+Y se poate activa imprimanta.  
Aceasta se indica prin mesajul "LPRINT" pe ecran. Astfel, tot ce  
se afiseaza pe ecran va fi transmis si la imprimanta. Dupa  
terminarea executiei curente, imprimanta va fi dezactivata  
automat. Astfel ca pt. tiparirea urmatoarei comenzi se apasa pe  
SS+Y. Rata de transmisie a interfetei RS232 este fixata in  
variabila sistem #5C36 dupa cum urmeaza: #01C1 pt. 300 Bd; #00E4  
pt. 600 Bd; #006E pt. 1200 Bd; #0035 pt. 2400; #0019 pt. 4800;  
#000C pt. 9600; #0005 pt. 19200 Bd. Predefinit este rata de  
transmisie de 1200 Bd.

#### 8. HARTA MEMORIEI DFUS

DFUS este sistemul de operare rezident in memorie si se afla  
intre locatiile 0 si #3FFFF. Intre #4000 si #5800 se afla  
memoria ecran, iar intre #5800 si #5B00 aflindu-se atributele de  
culoare. Pt a respecta configuratia SPECTRUM variabilele sistem  
sint plasate incepind cu #5C00. Stiva este fixata la #5FFD.  
Fisierul sursa incepe in mod normal la #6000, dar inceputul sau  
poate fi plasat si la alte adrese cu ajutorul comenzilor "N" sau  
"O". Tabela de simboluri incepe de la adresa #FFFF si descreste  
precum stiva, cu 8 octeti pt. fiecare eticheta utilizata.

#### 9. VARIABILELE SISTEM

Variabilele marcate cu "!" pot distruge sistemul de operare,  
iar cele notate cu "\*" poate fi util sa fie modificate.

Addr.	nr.oct.	nume	semnificatie
5C00	11	KSTATE	buf. intrare tastatura (SPECTRUM)
5C0B	1	*LASTK	ultima tasta apasata
5C0C	2	*MAXREP	
5C0E	2	*MINREP	
5C10	1	MAXWAR	const. de timp pt. clipire cursor
5C11	2	KEYRTP	contor de repetari
5C13	1	CRSFLA	indica daca cursorul este aprins sau stins
5C14	1	*CRSWAR	contor de clipire al cursorului
5C15	2	*PRTPOS	poz. de tiparire in mem video(adr)
5C17	1	*KOLON	coloana poz. de tiparire
5C18	1	*ROW	rindul poz. de tiparire
5C19	2	!STKPTR	stiva OPUS-ului
5C1B	1	CMNDR	nr. comenzii in executie
5C1C	2	SYMPTR	indicatorul tabelii de simboluri
5C1E	2	*SOURCE	adr. de start a fis. sursa OPUS
5C20	2	PROEND	adr. de sfirsit a fis sursa
5C22	2	LSTPTR	ind. de listare a fis. sursa
5C24	2	LSTLNG	nr. linii listate inainte de pauza
5C26	2	LUNG	lungimea liniei sursa de introdus
5C28	1	MARK	marcheaza un anumit simbol
5C29	2	ORIGIN	adr. ORG-ului
5C2B	2	DISTA	deplasamentul DIST-ului
5C2D	2	ENTRY	adr. ultimului ENT
5C2F	1	*PARMR	nr. de param. dintr-o comanda
5C30	2	*PARM1	
5C32	2	*PARM2	parametri comenzii in curs de exec.
5C34	2	*PARM3	
5C36	2	*BAUD	rata de transfer pt. imprimanta
5C38	1	*BORDER	culoarea borderului
5C39	1	*PATTR	octet de atribut pt. ecran
5C3A	1	!FLAGS 1	indicatori pt. OPUS
5C3B	1	!FLAGS	indicatori pt. tastatura
5C3C	2	DSTART	adresa de start a dezasamblorului
5C3E	3	CBYTE	memoreaza codul operatiei
5C41	1	MODE	indicatori pt. tastatura
5C42	2	CRTADR	adresa curenta a dezasamblorului
5C44	2	LASTA	ultima adresa a dezasamblorului
5C46	12	*WORKSP	spatiu de lucru
5C52	24	*REGSAV	reg. utilizator (exclusiv I si R)
5C6A	1	!FLAGS 2	alti indicatori
5C6B	2	*SPINH	continutul SP utilizator
5C6D	1	*ISAVE	continutul I utilizator
5C6E	1	*RSAVE	continutul R utilizator
5C6F	2	RERE	adr. de retur din rutina utilizator
5C71	1	DFLAG	indicator pt. dezasamblor
5C72	2	PTXT	ind. de test pt. dezasamblor
5C74	2	PTBUF	ind. in bufferul dezasamblor
5C76	2	POBUF	alti indicatori
5C78	3	*FRAMES	numaratorul de linii (SPECTRUM)
5C7B	2	*USERAD	adresa comenzii utilizator



5C7D	2	LNRTX	nr. de linii pt. textul dezasamblorului
5C7F	42	*COFIE	buffer de lucru OPUS
5CA6	42	*INBUFF	buffer de intrare OPUS

## 10. SUBRUTINE OPUS UTILE

La "input" se vor indica registrele ce contin datele de intrare in subrutina. La "output" se dau datele de iesire si registrele ce le contin. "Destroy" avata ce alte registre sint afectate de calculele din subrutina.

### 10.1 Afisari pe ecran

**PRTOUT #0010=** tipareste codul ASCII din registrul A la pozitia curenta de tiparit (IY=#5C3A)

-input: A=codul ASCII al caracterului de tiparit

-output:-

-destroy:-

**PRTMES #0030=** afiseaza mesajul ce urmeaza dupa RST#30 si care trebuie terminat cu octetul #00. I=: O=: D=:

**PRBLK #00AC=** afiseaza un blank. I=: O=: A=#20; D=:A

**PRTCR #00B0=** tipareste codul de control al saltului de inceput de rind nou. I=: O=: A=#0D; D=:A

**PRTCL #00B5=** sterge linia curenta prin tiparirea codului de control cu codul #07. I=: O=: A=7; D=:A

**PMSHL #00B9=** afiseaza mesajul indicat de HL si care trebuie sa fie terminat cu octetul #00. I=: HL=adresa de inceput a mesajului O=:D=:HL,AF

### 10.2 Conversii

**PRTHL #00C6=** afiseaza continutul registrului HL in hexa ,urmat de un blank. I=:HL; O=: D=:AF

**PRTDE #00C0=** afiseaza continutul memoriei indicate de DE. in hexa, urmat de un blank. I=:DE=indicator spre locatie de memorie (LSB); O=: D=:HL,DE,AF

**PRITA #0BD0=** tipareste continutul reg. A in hexa. I=:A=octet de tiparit; O=: D=:AF

**ABIT #00E3=** tipareste continutul reg. A in reprezentare binara I=:A=octet de tiparit; O=: D=:BC,AF

**CONVO #00F9=** tipareste continutul lui HL in recimal, fara zerouri nesemnificative in fata si terminat cu un blank. I=:HL=numar; O=:D=:C,AF

**CONVE #00FE=** ca si in CONVO dar cu zerouri nesemnificative

**ASCII #00F1=** converteste o cifra hexa in codul ASCII corespunzator I=:A=cifra hexa; O=:A=codul ASCII; D=:AF

**ASCHX #013E=** converteste un nr. codificat ASCII in valoarea sa hexa .Sint luate in considerare "W" si " " .I=:DE=indicatorul inceputului codului ASCII ; O=:HL= nr. convertit; D=:HL,DE,BC,AF

**INAHX #0191=** ca si in ASCII dar pt. nr. de 8 biti. I=:DE=codul de inceput al codului ASCII; O=:A+ nr. convertit D=:DE,BC,AF

**HTEST #009C=** converteste continutul reg. A in cifra hexa corespunzatoare I=:A= codul ASCII al cifrei; O=:A= cifra hexa; CY=1 daca codul nu e al cifrei; D=:AF

### 10.3 Teste

LETTER #0079= verifica daca continutul lui A e majuscula. I=:A= codul ASCII; O=:CY=1 daca testul a esuat; D=:CY

ZAHL #0080= verifica daca continutul lui A este numar. I=:A= codul ASCII; O=:CY=1 daca testul a esuat; D=:CY

ZIFFER #0087= verifica daca contine o litera sau o cifra (exclde simbolurile speciale). I=:A= codul ASCII; O=:CY=1 pt. test esuat; D=:CY

BCHTS #008B= verifica daca continutul lui A este o litera I=: codul ASCII; O=:CY=1 pt. test esuat; D=:CY

ASCTES #046D= verifica daca A contine un caracter ASCII. Daca nu, A este incarcat cu "."=#2E. I=:A= octet hexa; O=:CY= 1 pt. test esuat; D=:AF

### 10.4 Citiri de la tastatura

INCH #03FA= asteapta apasarea unei taste. I=: O=:A= codul ASCII al tastei apasate; D=DE,BC,AF

KPRESS #0426= tastea apasata si ridicata. I=:O=codul ASCII al tastei apasate; D=:DE,BC,AF

KEYBRD #3047= asteapta afisarea unei taste, afisind un cursor clipitor la pozitia curenta de tiparit. I=: O=:A= codul ASCII al tastei apasate; D=HL,DE,BC,AF

### 10.5 Lucrul cu caseta

SAVE #04C2= salveaza un bloc de date pe caseta. I=:A= octetul de indicator; DE=lungimea blocului; IX=adresa de start ;O=:D= toate reg.

LOAD #0556= incarca un bloc de date de pe caseta; (ca mai sus)

### 10.6 Sunete

CLICK #03FA= daca sunetul este comutat ( bitul 2 al #5C36 setat) subrutina trimite un pacanit de taste apasate. I=:D=:O=;

PILPEN #03B5= genereaza sunet (SPECTRUM) I=:DE=durata sunetului in secunde HL=durata frecventei; O=:D=:toate reg.

### 10.7 Subrutine speciale

HILD #01EF= compara DE cu HL. I=:HL;O=:CY=1 daca DE >HL, CY=0 daca DE <HL, Z=1 daca DE=HL; D=:F.

LADALL #0449= incarca parametrii comenzii in registre. I=: O=:BC= PARAM2-PARAM1, DE= PARAM3; D=:HL,DE,BC,F.

WATRE #30ER= da o secunda de pauza. I=:O=:D=:BC,AF.

DELAY #30EE= da intirziere pe const. din BC. I=:BC= durata; O=: D=:BC.

RSTOB #0008= reintoarcerea in subrutine utilizator. CY=1 este utilizat ca indicator de eroare la return. I=:O=:D=;

ERROR #0053= trimite mesajul "ERROR" pe ecran si da un biziit specific de eroare. I=:D=: toate registrele.

HOME #2DF9 = sterge ecranul si pune cursorul stinga sus. I=:O=: D=:HL,DE,BC,AF

## 11.COPIERUL OPUS

Cu ajutorul acestei optiuni OPUS se poate copia orice fisier de pe o caseta pe alta. Acest copier ofera toate facilitatile cunoscute de la oricare alt program de copiat

T=P-program BASIC

N-date numerice in matrice

S-date alfa numerice in matrice

C-zona de cod pt. un alt caracter. Tipul nu este standard

Length- lungimea preconizata

Addr.- adresa de incarcare sau linia de autorulare in cazul unui program BASIC

Bytes- lungimea reala a fisierului

Error- semnaleaza eroare la incarcare cu "\*"

VE- semnaleaza eroare la verificari

Cu tasta BREAK se poate opri modul de incarcare. Daca nu avem niciun fisier incarcat avem doar optiunile :

Q-iesire in pagina initiala OPUS

R-comuta baza de numeratie intre zecimal si hexa

S-comuta pacanitul de tastatura activ-inactiv

N-sterge tot ce a fost incarcat si trece in mod de incarcare

P-fisierele sint incarcate dupa lungimea data in header, pt. a putea incarca fisierele salvate unul dupa altul fara pauza. Nu pot fi incarcate in acest mod headless-uri iar in caz de incarcare sau in lipsa header-ului se intra automat in mod de salvare.

M-se ofera o memorie libera de 48K (-2 octeti) dar se poate incarca un singur fisier, fara header. Erorile de incarcare vor fi semnalate prin biziitul specific de eroare. Dupa incarcare in acest mod avem urmatoarele optiuni :

C-copiaza fisierul incarcat

N-incarca alt fisier

Q-se iese in pagina initiala OPUS

Dupa incarcarea unor fisiere vom avea si optiunile:

A-copie automat toate fisierele cu o pauza de o secunda

C-salveaza fisierul marcat

S-muta cursorul la urmatorul fisier

L-intra in mod de incarcare fara sa stearga ce a fost incarcat

V-verifica fisierul daca nu da eroare scrie "V", daca da scrie "\*"

D-sterge fisierul marcat

R-redefineste fisierul marcat (max. 10 caractere)

### ANEXA 1 CODURILE RAPORTELOR DE EROARE

0-caracter nepermis sau instructiuni incomplete

1-eticheta prea lunga

2-se asteapta ")"

3-eroare de trunchiere sau iesire din gama valorilor permise

4-se asteapta ",."

5-eroare de context

6-simbol nedefinit

7-se asteapta "("

