

A. PETRESCU : coordonator

GH. RIZESCU

F. IACOB

T. ILIN

E. DECISOV

C. NOVĂCESCU

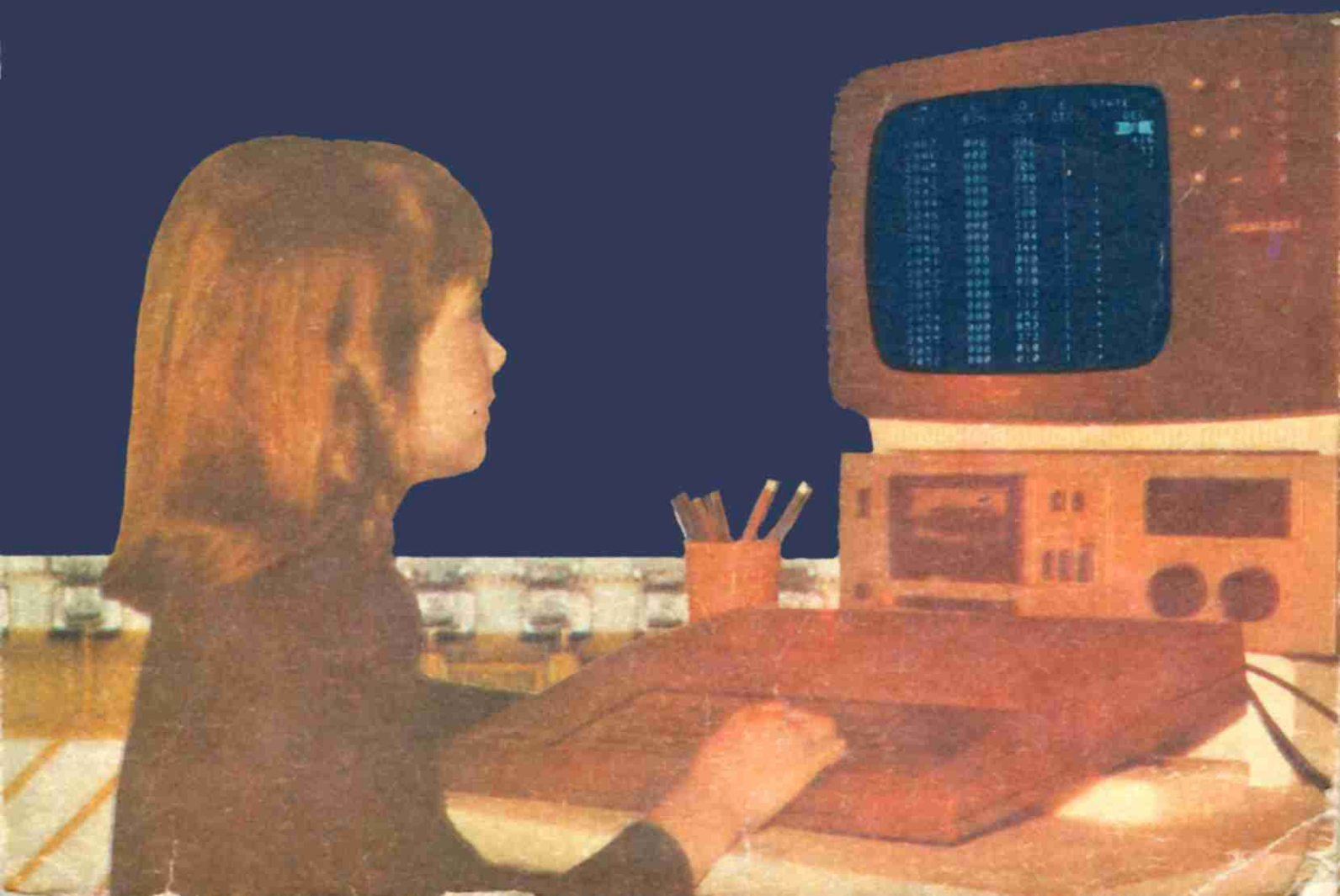
F. BAR

R. BERINDEANU

D. PANESCU

Volumul 1

TOTUL DESPRE ... CALCULATORUL PERSONAL aMIC



681
778

Prof. dr. ing. **Adrian Petrescu**

Prof. emerit **Gheorghe Rizescu**

Ing. asistent **Francisc Iacob**

Ing. **Tiberiu Ilin**

Ing. **Eduard Decsoy**

Ing. **Constantin Novăcescu**

Ing. **Florian Bar**

Ing. **Radu Berindeanu**

Ing. **Dumitru Pănescu**

**Totul despre ...
calculatorul
personal
aMIC**

Volumul 1

Coordonare : Prof. dr. ing. **Adrian Petrescu**

Prefață : Dr. ing. **Vasile Baltac**



Editura Tehnică
București, 1985

Colectivul de elaborare al cărții cuprinde specialiști de la Institutul Politehnic București, Liceul „Dimitrie Cantemir”, București, Întreprinderea de Memorii Electronice Timișoara, ITCI — Timișoara și „Electrotimis” Timișoara.

Contribuția autorilor este următoarea :

A. Petrescu : coordonarea, cap : 1 (p), 2 (p), 3 (p), 4, 7 (p), anexa 2
Gh. Rizescu : cap : 1 (p), 7 (p)
F. Iacob : cap : 3 (p), 5 (p), anexa 1
T. Ilin : cap : 3 (p), 5 (p), 6 (p)
E. Deesov : cap : 3 (p), 5 (p), 6 (p)
C. Novăcescu : cap : 2 (p), 3 (p)
F. Bar : cap : 2 (p), 3 (p)
R. Berindeanu : cap : 5 (p), 6 (p)
D. Pănescu : cap : 5 (p)

Recenzie : Dr. Ing. ADRIAN DAVIDOVICIU

Redactor : Ing. PAUL ZAMFIRESCU

BIBLIOTECA JUDEȚEANĂ BRASOV



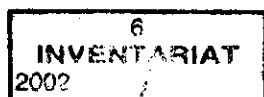
R05290710812

Culegerea și paginarea realizată de o echipă coordonată de EDUARD GIESSER

Coperta : Arh. SILVIA MIRȚU
Desene : LAURENȚIU ILIESCU
Tehnoredactor : ELLY GORUN

Bun de tipar 10 dec. 1985 Coli tipar, 17,5
C.Z. 681.142

Tiparul a fost executat sub comanda nr. 110
la Întreprinderea Poligrafică „Banat”
Timișoara, Calea Aradului nr. 1.
Republică Socialistă România.



476-930

Prefață

Ce este calculatorul individual (personal)? Cum să definim această familie de echipamente și programe? Ce caracteristici comune are cu alte produse ale tehnicii de calcul și prin ce se diferențiază? Cum va evolua?

Fără să adoptăm o definiție extinsă, vom spune simplu că este portabil, are o structură cu unul sau mai multe microprocesoare, o tastatură și un afișaj de tip TV, o memorie externă magnetică și unul sau mai multe limbaje de programare de nivel înalt, avînd un preț foarte accesibil. Toate resursele microsistemului sînt la dispoziția operatorului-programator pentru utilizare individuală interactivă. Deseori se folosește și denumirea de calculatoare personale, dar cu înțelesul de mai sus.

Însă orice definiție s-ar considera, aceasta ar trebui modificată mereu pentru a ține seama de evoluția performanțelor, tehnologiilor, funcțiunilor noi înglobate, interfețelor om-mașină tot mai naturale și prietenoase și facilităților de interconectare în rețele locale și acces la baze mari de date.

Japonezii au anunțat deja calculatoare personale de generația a cincea, suport pentru sistemele expert evaluate.

O serie de specialiști le numesc instrumente de lucru ale viitorului, dar apreciem că au devenit instrumente ale prezentului și prietene ale omului. Ale omului inginer, medic, proiectant, tehnolog, fizician, chimist, matematician, economist, muncitor, agricultor, profesor, elev, om de artă etc.

Și dacă în acest final de veac, locul îngust al dezvoltării mondiale pare a fi educația, atunci ce speruri uriașe, rezerve ale dezvoltării societăților, se pot obține prin accelerarea proceselor educaționale, de instruire asistată de calculatoare!

În mai multe țări, printre care U.R.S.S., S.U.A., Franța, Japonia, R.P.B., R.D.G. și Anglia există preocupări intense și chiar programe naționale privind introducerea calculatoarelor individuale la locurile de muncă, la domiciliu, în sfera educației și învățămîntului.

Astfel, în școlile și universitățile din S.U.A. erau instalate în 1984 peste un milion de calculatoare individuale, estimîndu-se o creștere de circa 4 ori pînă în anul 1986.

Cîteva dintre cele mai reprezentative calculatoare individuale, clasificate după performanțe, preț și loc de utilizare, sînt:

- familiale: modelele Sinclair;*
- educaționale: modelul Sinclair ZX 81 Spectrum în Anglia, Apple II în școlile americane, Praveț în R.P.B., Agat și Iskra în U.R.S.S., IBM PC în universitățile americane;*

— profesionale : modelul ISKRA 250 în U.R.S.S., modelele IBM PC, PC XT, PC AT cu microprocesoare evolute de 16 biți în S.U.A. și unele modele noi cu microprocesoare de 32 biți.

Modelele Sinclair au o memorie internă standard de 16—48 koct. și utilizează ca memorie externă minicaseta, iar celelalte tipuri de mai sus au memorii interne uzuale în plaja 64—1 000 koct. și memorii externe cu disc magnetic flexibil sau disc Winchester, fiind folosite frecvent sistemele de operare CP/M, MS/DOS și XENIX (o versiune UNIX pentru micro). Cele mai răspândite limbaje de programare sînt BASIC, LOGO, PASCAL, FORTRAN și FORTH.

Lansarea în urmă cu trei ani în producție de mare serie a modelului IBM PC, a generat preluarea de către IBM a rolului de lider și în acest segment al tehnicii de calcul, obținind o pondere de 30% din piața mondială, la care se adaugă un sector aproximativ egal al firmelor cu produse compatibile IBM PC.

Astfel, în 1984 lista programelor aplicative pentru IBM PC ajunsese la peste 11 000 de titluri, circa 700 de aplicații fiind din domeniul matematicii. De altfel, un matematician român scria recent că tendința principală în matematica de azi este „informaticizarea” (algoritmizarea, discretizarea, apelul la calculator), pentru a-l cita pe Gheorghe Păun.

Organizarea de către IBM a unor capacități de producție cu grad ridicat de automatizare, pentru cîteva milioane de calculatoare personale anual, a permis nu numai reducerea spectaculoasă a ciclului de producție, dar și o reducere dramatică a costurilor de fabricație, antrenînd dezvoltarea în continuare a unei largi industrii orizontale.

În acest fel, calculatorul individual a devenit purtător al undei de progres tehnic și inovare tehnologică, iar prin difuzarea sa în masă, accesibilitate și prin pătrunderea în toate sferile activității umane, permite implementarea conceptului de informatică distribuită, accelerînd trecerea spre viitoarele societăți informatizate.

Iată de ce apreciem că se justifică pe deplin acțiunea de a se organiza trecerea chiar din acest an la asimilarea în producție de serie a unei familii de calculatoare individuale atât la Fabrica de memorii electronice și componente pentru tehnica de calcul din Timișoara, cît și la alte întreprinderi de profil din București, acoperind în bună măsură tendințele prezentate mai sus și cerințele economiei naționale.

Se au în vedere atât modelele originale, competitive ca performanțe, aMIC și PRAE*, cît și alte modele de 8 și 16 biți compatibile cu cele mai răspândite tipuri pe plan mondial (HC-85 și FELIX PC).

Calculatorul aMIC este conceput de un colectiv de cercetare condus de prof. dr. ing. Adrian Petrescu, binecunoscut în țara noastră pentru o serie de inițiatve și realizări în domeniul microcalculatoarelor. Avînd un design modern, cu o tehnologie fiabilă și pachete extinse de programe aplicative puse la punct prin colaborarea specialiștilor din ITCI și FMECTC Timișoara, modelul aMIC la fel ca și PRAE, este îndrăgit de copiii, elevii și studenții care au avut posibilitatea „să se împrietenească” cu calculatoarele individuale românești, atât cu ocazia taberelor

*) PRAE reprezintă un ”în ceput” (vezi latinescul citit pre) care generează o familie de modele bazate pe inițiativa colectivului Filialei din Cluj-Napoca a Institutului de cercetare științifică și inginerie tehnologică pentru tehnica de calcul și informatică — București (ITCI).

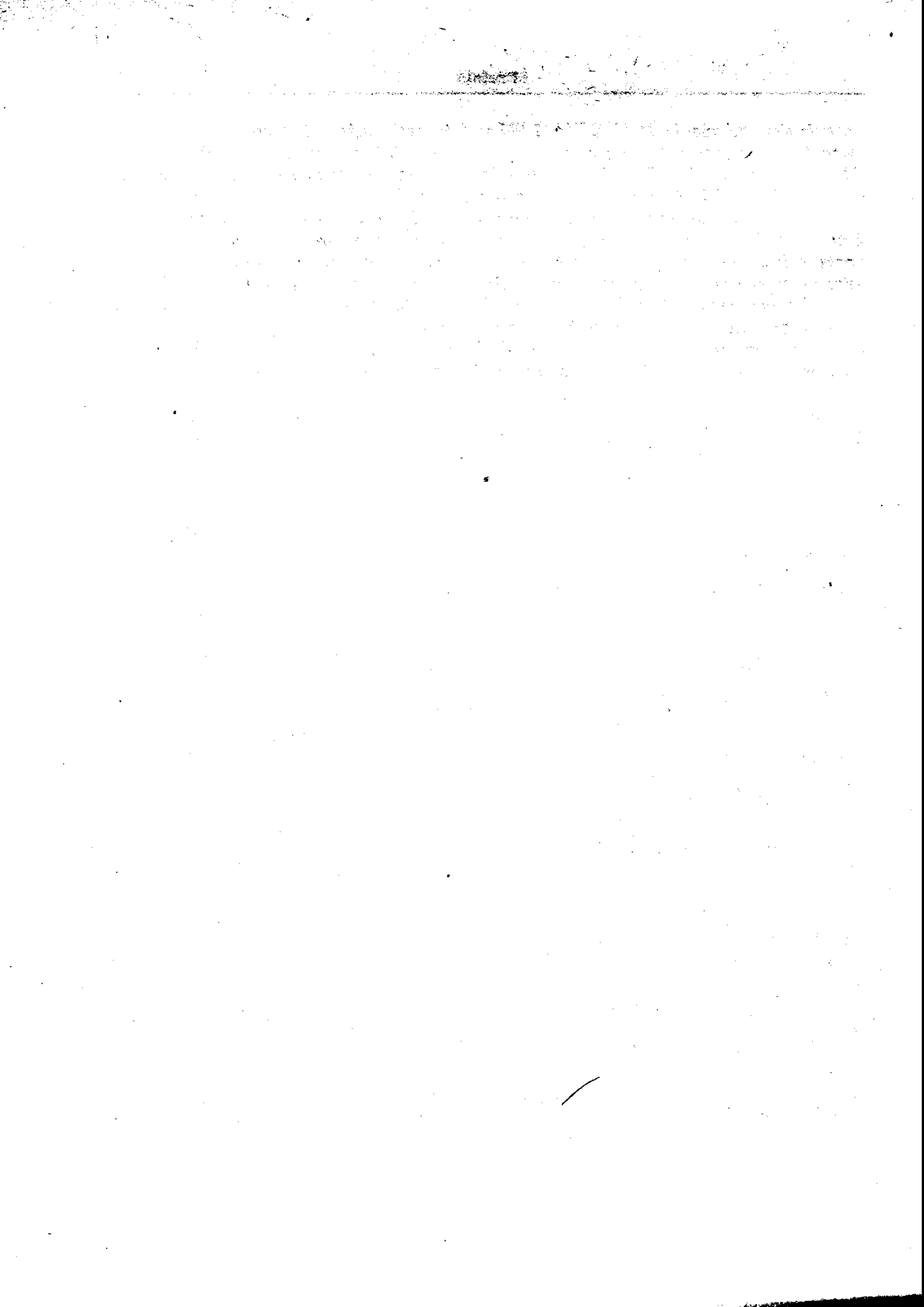
de instruire organizate de ITCI în 1985 și a taberei inițiate de Catedra de calculatoare din IPB care a avut loc recent, cele mai multe tabere beneficiind și de sprijinul CNOP și CC al U.T.G., cît și cu prilejul organizării cercurilor de copii, elevi și studenți de la ITCI și Catedra de calculatoare din IPB.

Colectivul Catedrei de calculatoare din IPB, oferă împreună cu specialiștii din ITCI și întreprinderile de profil și cu această ocazie, un bun exemplu de integrare a învățămîntului superior cu cercetarea și producția, mai ales în domeniul microcalculatoarelor și terminalelor inteligente, dar și în alte domenii de vîrf.

În acest context, subliniez receptivitatea față de nou a Editurii Tehnice, considerînd inițiativa de a publica această carte despre calculatorul individual aMIC extrem de valoroasă, lucrarea fiind așteptată cu un viu interes de un cerc larg de cititori, viitori utilizatori ai calculatoarelor personale fabricate în țară.

Dr. ing. VASILE BALTAO

15 decembrie 1985



Cuvînt înainte

Gabaritele reduse, preşurile relativ mici, fiabilitatea ridicată, simplitatea exploatării, au făcut ca sistemele de tip microcalculator personal (în continuare se va folosi termenul de calculator personal) să devină un mijloc de tehnică de calcul de masă, cu aplicaţii în cele mai multe domenii ale activităţii sociale: ştiinţă, producţie, învăţămînt, medicină, agricultură etc.

Larga utilizare a calculatoarelor personale permite creşterea eficienţei şi exactităţii activităţilor ştiinţifice şi financiar-contabile, sporeşte eficienţa lucrărilor de cercetare şi proiectare, asigură un înalt nivel tehnic al producţiei.

Calităţile tehnice şi de exploatare ale calculatoarelor personale au creat premisele creşterii volumului producţiei şi al vânzărilor acestor echipamente. Actualmente în întreaga lume peste 300 de firme produc circa 700—800 tipuri de calculatoare personale. Numai în anul 1983 au fost produse circa 5,7 milioane bucăţi. Numărul lor în S.U.A., în anul 1983, a fost aproximativ 11 milioane bucăţi, considerîndu-se că, spre sfîrşitul secolului, acesta va creşte cu un ordin de mărime.

O latură a eficienţei calculatoarelor personale se referă la faptul că o bună parte din categoria celor personal-profesionale şi respectiv-familiale este achiziţionată de persoane particulare care urmăresc creşterea eficienţei şi a nivelului ştiinţific al activităţilor desfăşurate de ele în ştiinţă, tehnică, medicină, învăţămînt etc.

Un asemenea echipament de tehnică de calcul trebuie să posede o fiabilitate foarte ridicată, care se obţine printr-un înalt nivel tehnologic, prin folosirea tehnicilor de proiectare şi asamblare asistate de calculator, printr-un software puternic şi prietenos, orientat către utilizatorii neprofionişti în domeniul programării.

Utilizarea cu succes a calculatoarelor personale impune o modificare substanţială a conceptelor stabilite în ultimii 30 de ani, în legătură cu tehnologia programării, dimensiunile, structura, complexitatea şi calitatea service-ului echipamentelor de tehnică de calcul.

Pentru a da un puternic impuls dezvoltării forţelor de producţie sînt necesare măsuri ferme în vederea răspîndirii în masă a cunoştinţelor privind utilizarea calculatoarelor personale, producerea lor în cantităţi mari, la costuri accesibile.

Se apreciază ca rămînerea în urmă a oricărei ţări industrializate în privinţa introducerii calculatoarelor personale, în principalele domenii economico-sociale, va necesita în următoarea decadă eforturi materiale foarte mari pentru a depăşi consecinţele unei asemenea situaţii. Nivelul scăzut al productivităţii

muncii în sfera activităților legate de informatică va constitui o problemă avînd aceleași dimensiuni ca și cea a naștinței de carte de la începutul secolului nostru.

*
* *

În cadrul Catedrei de calculatoare din Institutul Politehnic București, încă din anul 1976 a fost realizat un microcalculator bazat pe microprocesorul 8080, microcalculator care a purtat numele MC-80 și care a constituit punctul de plecare pentru FELIX-M18.

Sub forma inițială, MC-80 era prevăzut cu o memorie REPR0M de 16 Ko. și o memorie RAM de 16 Ko. În memoria REPR0M se afla un monitor simplu, cu ajutorul căruia se putea citi de la un lector de bandă perforată, sau de la un casetofon, un interpretor pentru limbajul BASIC. Ca dispozitive de dialog cu operatorul s-au folosit un display și un teletype.

Realizarea în țara noastră a microprocesoarelor 8080 și Z80, a memoriilor RAM dinamice de 16 Ko., a permis, în anii 1982—1983 proiectarea și execuția unor microcalculatoare de laborator, folosind ca dispozitiv de afișare un televizor alb/negru comercial, iar ca dispozitiv de intrare o tastatură alfanumerică simplă. Prevăzute cu o memorie EPROM de 16 Ko. și o memorie RAM de 16—48 Ko., aceste microcalculatoare dispuneau de monitoare puternice, de asamblare, editoare de texte și interpretoare pentru limbajul BASIC. Stocarea programelor se realiza cu ajutorul unui casetofon comercial.

Asigurarea accesului din exterior la magistrala internă de date, adrese și comenzi, a permis conectarea unor echipamente periferice nestandard, în cadrul unor lucrări de laborator.

Au fost realizate numeroase modele, în variante bazate pe microprocesoarele 8080/Z80 și pe memoriile statice 2114/memoriile dinamice 4416. Două dintre aceste modele au fost prezentate, în anul 1983, conducerii Consiliului Național pentru Știință și Tehnologie, care, apreciind utilitatea unor asemenea echipamente ieftine de tehnică de calcul, a recomandat introducerea lor în fabricație.

Cu sprijinul tovarășului dr. ing. V. Baltac, Secretar de Stat în Ministerul Industriei Construcțiilor de Mașini, proiectul a fost preluat de Întreprinderea de memorii electronice, care, împreună cu Institutul pentru Tehnică de Calcul — Timișoara, au avut în continuare o importantă contribuție în ceea ce privește adaptarea proiectului și implementarea lui într-o tehnologie adecvată, cum și în privința dezvoltării pachetelor de programe de sistem și aplicații.

Produsul respectiv a primit denumirea de aMIC, în ideea că el va reprezenta un adevărat „prieten“ al proiectanților, cercetătorilor științifici, profesurilor, studenților, elevilor și al altor categorii de oameni ai muncii, în activitățile lor curente.

Microcalculatorul aMIC poate fi folosit atât pentru calcule tehnico-științifice, cît și pentru conducerea unor procese tehnologice de complexitate redusă.

Ideea care a stat la baza proiectului a fost aceea a unui produs de tehnică de calcul ieftin, cu performanțe superioare, folosind cu precădere componente și echipamente electronice (televizor alb/negru, casetofon) din producția curentă a întreprinderilor noastre.

Fiind un calculator programabil atât în limbaj de asamblare cât și în limbaj de nivel înalt (BASIC), el poate fi folosit în echipamente complexe, sub forma unui calculator pe o singură plachetă, pierzîndu-și astfel identitatea.

Pe baza acestui calculator, specialiștii de la ITC — Timișoara și IPB au realizat numeroase instalații complexe, dintre care unele sînt prezentate în această lucrare. De asemenea, trebuie subliniată (ca și în carte) utilizarea lui pentru conducerea unui minirobot în cadrul Întreprinderii Electrotimiș.

Încă de la început au fost sesizate posibilitățile acestui calculator personal în procesul de învățămînt. Pe baza bogatei experiențe privind organizarea, începînd din anii 1974—1978, a laboratorului de matematici (cu aplicații în tehnica de calcul), la Liceul „Dimitrie Cantemir“, din capitală, sub conducerea profesorului emerit Gh. Rîzescu *) **) în anii școlari 1983—1984 și 1984—1985, au fost organizate grupe de elevi pentru studiul bazelor aritmetice și logice ale calculatoarelor, avînd în vedere perspectiva introducerii în fabricație a calculatoarelor personale în țara noastră. Au fost, de asemenea, elaborate pachete de programe pe calculatorul aMIC, pentru asistarea predării unor capitole de matematici din programa claselor IX—X, din liceu.

Rezultatele obținute au fost comunicate la sesiunile științifice și consfăturile pe sector, municipiu și țară ale profesorilor de specialitate, ca, de altfel, și în cadrul altor acțiuni. De asemenea, pe linia manifestărilor științifice ale elevilor, la nivel de municipiu și țară au fost făcute comunicări, care s-au bucurat de o bună apreciere. ***)

Colaborarea între Institutul Politehnic București, Catedra de calculatoare și Liceul „Dimitrie Cantemir“ se desfășoară în baza unui protocol care vizează folosirea experimentală în învățămîntul liceal a calculatoarelor electronice.

Trebuie subliniate, de asemenea, acțiunile privind organizarea unor tabere de instruire în domeniul calculatoarelor, pentru elevi și studenți, la inițiativa și cu sprijinul Uniunii Tineretului Comunist și al Consiliului Național al Pionierilor. Asemenea tabere, cu rezultate excelente, au funcționat în anul 1985 la Brașov și Cîmpulung Muscel. Ele au fost organizate cu bază materială și instructori de la Institutul pentru Tehnică de Calcul București, Institutul Politehnic București și Întreprinderea de calculatoare electronice. ITC — București a organizat un laborator dotat cu calculatoare personale aMIC și Prae în care săptămînal sînt instruite grupuri de elevi de la diverse școli din capitală.

*) Acad. N. Teodorescu, Prof. emerit Gh. Rîzescu, ș.a.

Laboratorul de matematică

Organizarea laboratorului și recomandări privind desfășurarea lucrărilor practice. EDP. 1974.

**) Prof. emerit Gh. Rîzescu.

Îndrumător.

Laborator școlar de matematică. Teme și fișe experimentale. 421 pag. Ministerul Industriei Construcțiilor de Mașini, 1978.

***) I. Petrescu. Programe în BASIC pe microcalculatorul aMIC, privind unele capitole de matematici din materia clasei a IX-a. Comunicare la sesiunea pe țară a cercurilor științifice ale elevilor. Pitești, 1984.

I. Petrescu. Biblioteca de programe în BASIC, pe calculatorul HC-85, pentru unele capitole de matematici din materia clasei a X-a. Comunicare la sesiunea pe municipiu a cercurilor științifice ale elevilor. București, mai 1985.

Desigur, realizarea unui microcalculator nu ridică probleme deosebite pentru industria noastră. Adevăratele probleme sînt legate de obținerea unei fiabilități ridicate a produsului, de prevederea unor posibilități de depanare rapidă și de asigurarea unui software de sistem și aplicații cît mai bogat, „prietenos“ orientat către cele mai largi categorii de utilizatori.

Din acest punct de vedere nu trebuie să se considere că microcalculatorul aMIC este un produs „înghețat“. El este într-o continuă evoluție, atît sub aspectul hardware-lui, cît și sub cel al software-lui. Astfel, se conectează noi echipamente periferice, se realizează noi aplicații, se implementează noi tipuri de limbaje (Forth de exemplu), se încearcă compatibilizarea cu limbaje BASIC de pe alte calculatoare personale. La Institutul Politehnic „Traian Vuia“, din Timișoara s-a realizat experimental, prin unele modificări hardware, pornind de la aMIC, un echipament de calcul „Spectim“, compatibil — în cea mai mare măsură — cu limbajul BASIC-Sinclair Spectrum.

În contextul apariției altor calculatoare personale din aceeași clasă (HC-85, Prae, DEGA-209 etc.) sau din clase superioare (FELIX-AP, cu microprocesor 6502 și disc flexibil; FELIX PC, cu microprocesorul 8086/8088 și disc flexibil), aMIC nu-și pierde actualitatea, avînd în vedere costul său scăzut, existența unei importante baze de programe de sistem și aplicații, fiabilitatea lui ridicată și realizarea lui cu componente produse exclusiv în țară.

Lucrarea de față are la bază experiența specialiștilor de la Institutul Politehnic București, Institutul pentru Tehnică de Calcul — Timișoara, Întreprinderea de memorii — Timișoara, Întreprinderea Electrotimiș, Liceul „Dimitrie Cantemir“ București. Autorii mulțumesc Editurii Tehnice și, în mod deosebit, redactorului de specialitate, ing. Paul Zamfirescu pentru efortul depus în privința orientării spre aplicații, pentru structurarea și actualizarea lucrării.

Cuprins

(VOLUMUL 1)

Prefață	5
Curvint înainte	9
Cuprins volumul 1 și volumul 2	13
Capitolul 1. Clase de microcalculatoare personale și personal-profesionale.	21
1.1. Calculatoare de buzunar programabile	21
1.2. Microcalculatoare personale (individuale)	22
1.3. Microcalculatoare personal-profesionale	23
1.3.1. Microcalculatorul profesional CUB	24
1.3.2. Terminalul de pregătire a datelor TPD	24
1.3.3. Microcalculatorul personal-profesional FELIX-PC	25
1.4. Caracteristici tehnice și comerciale ale unor calculatoare de buzunar, calculatoare personale și personal-profesionale străine	27
Capitolul 2. Prezentarea generală a microcalculatorului aMIC.	31
2.1. Componente și scheme bloc	31
2.2. Software de bază. (monitoare, asamblor, interpretor BASIC)	34
2.2.1. Monitorul aMIC V0.1 (sumar, în extenso în 5.1)	35
2.2.2. Monitorul aMIC V0.2 (sumar, în extenso în 5.2)	35
2.2.3. Monitorul Z80-V0.0 (sumar, în extenso în 5.3)	36
2.2.4. Monitorul DEST. (sumar, în extenso în Cap. 6)	37
2.2.5. Monitor-Asamblor-Text-Editor (MATE). (sumar, în extenso în Cap. 7)	37
2.2.6. Interpretorul pentru limbajul BASIC	38
2.2.7. BASIC-memento (în extenso în Cap. 9, din vol. 2)	38
2.3. Configurații disponibile la desfacere	43

Capitolul 3. Structura și funcționarea microcalculatorului aMIC	45
3.1. <i>Generalități</i>	45
3.2. <i>Unitatea centrală de prelucrare</i>	49
3.3. <i>Memoria RAM</i>	52
3.4. <i>Memoria EPROM</i>	60
3.5. <i>Interfața cu tastatură</i>	61
3.6. <i>Interfața cu televizorul</i>	68
3.7. <i>Interfața de comunicație serială</i>	72
3.8. <i>Interfața pentru casetofonul audio</i>	75
3.9. <i>Sursa de alimentare</i>	79
Capitolul 4. Microprocesorul Z80. Interfețele programabile	80
4.1. <i>Generalități</i>	80
4.2. <i>Structura internă</i>	81
4.3. <i>Terminalele microprocesorului Z80 și semnalele asociate</i> ...	84
4.4. <i>Sincronizarea și execuția instrucțiunilor microprocesorului Z80</i>	86
4.5. <i>Înteruperile externe</i>	91
4.6. <i>Starea HALT</i>	94
4.7. <i>Instrucțiunile microprocesorului Z80</i>	95
4.8. <i>Interfața paralelă programabilă PIO</i>	112
4.9. <i>Interfața serială programabilă SIO</i>	120
4.10. <i>Circuitul contor-temporizator CTC</i>	138
Capitolul 5. Monitoarele V0.1, V0.2, Z80-V0.0	138
5.1. <i>Monitorul V0.1</i>	138
5.1.1. <i>Prezentare generală</i>	138
5.1.2. <i>Comenzile monitorului</i>	142
5.1.3. <i>Exemple de utilizare</i>	145
5.2. <i>Monitorul MON, aMIC V0.2</i>	147
5.2.1. <i>Prezentare generală</i>	147
5.2.2. <i>Comenzile monitorului V0.2</i>	148
5.2.3. <i>Funcțiile utilizator, descriere și utilizare, funcții standard-STD și funcții nestandard-NST</i>	157
5.2.4. <i>Structura zonelor de lucru utilizate de monitorul V0.2</i> ...	162
<i>Zona de memorie EPROM</i>	162
<i>Zona de memorie RAM</i>	164
5.2.5. <i>Modul de utilizare a monitorului V0.2</i>	166

5.3. Monitorul Z80-V0.0	167
5.3.1. Prezentare generală	167
5.3.2. Comenzile monitorului	168
5.3.3. Legătura monitor-utilizator	171
5.3.4. Exemple de utilizare	172
Capitolul 6. Monitorul DEST	174
6.1. Introducere	174
6.2. Comenzile monitorului	175
6.2.1. Comanda A (assembly source program)	175
— Definierea termenilor	176
— Sintaxa limbajului de asamblare acceptat de asamblorul ASR-Z80	176
— Directivele admise de asamblorul ASR-Z80	177
— Evaluarea expresiilor din câmpul de argument	181
— Modul de utilizare a asamblorului ASR-Z80	182
— Lista erorilor emise de asamblorul ASR-Z80	184
— Restricții de utilizare a asamblorului ASR-Z80	185
6.2.2. Comanda E (edit source program)	185
— Definierea termenilor	186
— Comenzile acceptate de editorul de texte EDR-Z80	186
— Restricții ale editorului de texte EDR-Z80	189
6.2.3. Comanda P (list disassembled code)	189
6.2.4. Comanda T (trace flow of execution)	190
6.2.5. Comanda Q (relocate and link object modules)	191
— Restricții ale editorului de legături LRR-Z80	193
Capitolul 7. Sistemul de operare rezident MATE (Monitor-Asamblor-Text-Editor)	194
7.1. Generalități	194
7.2. Comenzile modului monitor	194
7.3. Formatul comenzilor modului monitor	195
7.4. Editorul de fișiere	196
7.5. Asamblorul	197
7.5.1. Instrucțiunile limbajului de asamblare	197
7.5.2. Nume simbolice	198

Cuprins

(VOLUMUL 2)

Capitolul 8. Cuplări de echipamente periferice, interconectări și aplicații ale microcalculatorului aMIC	7
8.1. Cuplarea unor LED-uri și comutatoare	7
8.2. Cuplarea unui convertor numeric analogic	8
8.3. Interconectarea cu microcalculatoarele <i>FELIX M18</i>	11
8.4. Cuplarea unui <i>JOYSTICK</i>	12
8.5. Cuplarea convertor analog/numeric	19
8.6. Simularea unui circuit logic	21
8.7. Cuplarea la microcalculator a unei minitimprimante <i>MIM40</i> ...	29
8.8. Cuplarea microcalculatorului cu un programator de EPROM	30
8.9. Cuplarea cu un terminal <i>DAF 2010</i>	30
8.10. <i>Interfața cu un minirobot</i>	37
8.11. <i>Echipament de testare pentru micro sisteme orientate pe magistrală</i>	39
8.12. <i>aMIC-ul în unități de deservire pentru mașini unelte</i>	39
8.13. Sistem de înregistrare/redare a parametrilor semicontinui de proces	41
8.14. Microcalculator (de laborator) pentru prelucrarea datelor provenite din analiza cromatografică	42
Capitolul 9. Limbajul BASIC, pentru microcalculatorul personal aMIC. Manual practic	44
9.1. <i>Introducere</i>	44
9.2. <i>Elementele limbajului BASIC</i>	47
9.2.1. <i>Constante</i>	47
9.2.2. <i>Variabile</i>	49
9.2.3. <i>Operatori</i>	49
9.2.4. <i>Funcții</i>	49
9.2.5. <i>Expresii</i>	51
9.2.6. <i>Instrucțiuni și monezi</i>	52
9.2.7. <i>Exerciții</i>	52
9.3. <i>Comenzile și modul de utilizare</i>	54
9.3.1. <i>Lansarea în execuție a interpretorului BASIC</i>	54
9.3.2. <i>Editarea programului</i>	55
9.3.3. <i>Listarea și salvarea pe casetă a unui program</i>	55
9.3.4. <i>Citirea unui program</i>	56

9.3.5. Execuția unui program	56
9.3.6. Ștergerea unui program din memorie	57
9.3.7. Exerciții	57
9.4. Instrucțiunile limbajului BASIC	58
9.4.1. Exemplu de program	58
9.4.2. Comentarea unui program	59
9.4.3. Terminarea unui program	59
9.4.4. Instrucțiunea de atribuire — LET	59
9.4.5. Utilizarea variabilelor indexate — DIM	60
9.4.6. Exerciții	61
9.4.7. Instrucțiuni de intrare/ieșire	62
9.4.8. Exerciții	67
9.4.9. Instrucțiuni de control (transfer necondiționat, condiționat și ciclare)	69
9.4.10. Exerciții	73
9.4.11. Utilizarea subrutinelor	74
9.4.12. Exerciții	77
9.4.13. Instrucțiuni de calcul cu matrici	83
9.4.14. Instrucțiuni de prelucrare grafică	89
9.5. Mesajele de eroare ale interpretorului BASIC	96
Capitolul 10. Microcalculatorul aMIC în matematicile elementare și statistică	98
10.1. Rezolvarea ecuației de gradul II	99
10.2. Rezolvarea inecuației $A \cdot X^2 \pm B \cdot X + C < 0$	100
10.3. Rezolvarea unui sistem (Cramer) de 5 ecuații cu 5 necunoscute	101
10.4. Afișarea unui șir finit de numere prime	102
10.5. Verificarea dacă un număr dat este prin sau nu	103
10.6. Descompunerea unui număr în factori primi	105
10.7. Determinarea celui mai mare divizor comun	106
10.7. Determinarea celui mai mare divizor comun	106
10.8. Simplificarea unei fracții	106
10.9. Calculul aproximativ al factorialului unui număr	106
10.10. Permutări, aranjamente, combinații	107
10.11. Ordonarea unui șir de numere	108
10.12. Calculul sumei celor mai mari m numere dintr-un șir de n numere date	108
10.13. Calculul valorii medii ponderate a unei variabile aleatoare	109
10.14. Calculul valorii medii și abaterii unei variabile aleatoare	109
10.15. Tabela valorilor unei funcții definite pe intervale	110
10.16. Calculul volumului butoiului	110
10.17. Calculul volumului și suprafeței torului	111
10.18. Calculul perimetrului și suprafeței unui triunghi	111
10.19. Calculul celui de-al N -lea număr din șirul lui Fibonacci	112
10.20. Calculul aproximativ al rădăcinii $\sqrt[n]{X} = Z, n > 2, X > 0$	113
Capitolul 11. Microcalculatorul aMIC în economie și tehnică	114
11.1. Antecalculația de preț pentru un produs	114
11.2. Calculul primei acordate după grupa de vechime	115

11.3. Determinarea beneficiului pentru o structură de fabricație de produse dată	115
11.4. Determinarea drumului minim între două noduri ale unui graj dat	119
11.5. Gestiunea unui stoc de magazie de tehnică dentară	120
11.6. Balanța de verificare debit-credit	123
11.7. Transformarea stea-triunghi și invers	124
11.8. Dimensionarea liniilor de alimentare în curent continuu	125
11.9. Determinarea greutateii materialelor	126
11.10. Dimensionarea grinzilor de beton armat	127
11.11. Calculul secțiunii elementelor de construcție	127
11.12. Determinarea momentelor de incastrare perfectă ale unei grinzi de beton armat	128
11.13. Optimizarea consumului de îngrășăminte chimice în agricultură	129
11.14. Calculul volumului rezervorului de compensație pentru rețeaua de apă potabilă	131
11.15. Studiul unui filtru „trece — jos”	131
11.16. Calculul salinității unui canal de ecluză	132
11.17. Calculul hidraulic al ecluzelor	134

Capitolul 12. Microcalculatorul aMIC în învățămînt 138

12.1. Modalități de integrare a calculatorului în procesul de predare-învățare, rolul și locul acestuia în asistarea procesului de învățămînt	138
12.2. Modele de lecții sau secvențe ale acestora pentru instruirea asistată de calculator (IAC) în predarea matematicii în liceu	142
12.3. Programe utile în procesul de învățămînt	149
12.4. Program pentru trasarea cercului trigonometric	149
12.5. Program pentru vizualizarea pozițiilor unor drepte care trec prin originea axelor de coordonate	150
12.6. Graficul funcției de gradul doi	150
12.7. Graficul funcției de gradul n	151
12.8. Graficul funcției logaritmice	152
12.9. Program pentru studiat aruncării corpurilor sub un unghi dat	153
12.10. Calculul punctului de intersecție a două drepte	154
12.11. Calculul punctelor de intersecție a două cercuri	156
12.12. Calculul tangentelor dintr-un punct la un cerc	158
12.13. Calcule cu polinoame	161
12.14. Rezolvarea ecuațiilor algebrice prin metoda Bairstow	161
12.15. Metoda celor mai mici pătrate	164
12.16. Transformata Fourier rapidă	168
12.17. Simularea salturilor unei mingi	171
12.18. Exerciții de despărțire a cuvintelor în silabe	173
12.19. Verificarea cunoștințelor de geografie	174
12.20. Verificarea cunoștințelor unui grup de candidați	175
12.21. Ordonarea candidaților după medțiile obținute	176

Capitolul 13. Microcalculatorul aMIC in grafică, jocuri aplicații diverse	178
13.1. <i>Trasarea strofoidei</i>	178
13.2. <i>Trasarea cicloidei</i>	179
13.3. <i>Trasarea epicycloidei</i>	179
13.4. <i>Trasarea melcului lui Pascal</i>	180
13.5. <i>Trasarea cercului circumscris unui triunghi</i>	181
13.6. <i>Grafiicul funcției polinomiale</i>	183
13.7. <i>Suma grafică a mai multor vectori</i>	184
13.8. <i>Mișcarea unui punct material într-un câmp gravitațional</i> ...	185
13.9. <i>Generarea și modificarea unei figuri</i>	186
13.10. <i>Generarea de figuri tridimensionale conform legilor perspectivei</i>	186
13.11. <i>Trasare de labirint</i>	193
13.12. <i>Mastermind</i>	194
13.13. <i>Vânătoare de vulpi</i>	195
13.14. <i>Verificarea vitezei de reacție</i>	196
13.15. <i>Perspico</i>	196
13.16. <i>Cursa cu obstacole</i>	197
13.17. <i>Tragerea la fiță</i>	198
13.18. <i>Ecranul magic</i>	199
13.19. <i>Nim</i>	200
13.20. <i>Turnurile din Hanoi</i>	202
13.21. <i>Jocul cu trei grămezi</i>	204
13.22. <i>Ruletă</i>	205
13.23. <i>Trasarea bioritmului</i>	207
13.24. <i>Dicționar de sinonime</i>	209
13.25. <i>Ordonarea unui set de informații</i>	210
13.26. <i>Decodificarea numerelor romane</i>	210
13.27. <i>Mira de control aMIC</i>	211
13.28. <i>Ceas electronic</i>	212
13.29. <i>Anagrame</i>	213
13.30. <i>Bugetul cheltuielilor zilnice într-o familie</i>	214
13.31. <i>Microfișier</i>	214
13.32. <i>Universul lui Conway</i>	216
13.33. <i>Pătratul magic</i>	218
Capitolul 14. Testarea resurselor hardware și a interpretorului BASIC	221
14.1. <i>Prezentarea generală a setului de programe de test</i>	221
14.2. <i>Comanda E — testarea zonei de memorie EPROM</i>	223
14.3. <i>Comanda K — testarea preluării de caractere de la tastatură</i>	224
14.4. <i>Comanda D — testarea afișării pe ecran</i>	224
14.5. <i>Comanda R — testarea zonei de memorie RAM</i>	226
14.5.1. <i>Descrierea modurilor de lucru ale programului</i>	226
14.5.2. <i>Organizarea testului RAM</i>	230
14.5.3. <i>Modul de testare a erorilor</i>	232
14.6. <i>Testarea transferului de informații dinspre/spre casetofon</i> ...	234
14.7. <i>Procedura de test a interpretorului BASIC</i>	235
Anexa 3. Colecție de programe pentru rezolvarea unor probleme de matematică din materia claselor a IX-a și a X-a	245

Clase de microcalculatoare personale și personal-profesionale

Progresele înregistrate în domeniul tehnologiei circuitelor integrate pe scară largă și foarte largă au permis realizarea unei game de mijloace de tehnică de calcul, bazate pe microprocesoare, extrem de diversificate în privința performanțelor și a costurilor.

Cunoscute sub numele generic de microcalculatoare, ele pot fi împărțite în prezent în mai multe grupe, în funcție de performanțe, caracteristici tehnice, utilizări, costuri etc.

1.1. Calculatoare de buzunar programabile

Calculatoarele de buzunar programabile în limbaje puțin evaluate (limbaj-mașină) se plasează la nivelul inferior al gamei, fiind capabile să execute programe cu un număr relativ mic de instrucțiuni sau pași. Ele sînt construite pe baza unor circuite specializate, integrate pe scară medie sau scară largă, dispun de o tastatură miniaturizată și de un ecran de afișare, prevăzut cu diode luminescente sau cu cristale lichide. Pînă la începutul acestui deceniu ele erau cunoscute sub numele de calculatoare de buzunar, avînd o largă răspîndire și fiind utilizate în special pentru calcule tehnico-științifice.

După tipul de limbaj-mașină folosit, aceste calculatoare se pot plasa în două mari categorii:

- calculatoare care utilizează un limbaj-mașină corespunzător notației poloneze inverse, bazate pe o unitate aritmetică cu organizare de tip știvă;
- calculatoare care se programează într-un limbaj de tip algebric.

Din prima categorie fac parte calculatoarele: CE 109 M (produs la centrul de Cercetări de Automatică București), HP41, HP67, HP97 (produse de firma Hewlett Packard) etc.

În cea de-a doua categorie se plasează calculatoarele TI58, TI59 (produse de Texas Instruments) și altele.

Intrucît aceste calculatoare nu pot fi utilizate pentru prelucrarea informației alfanumerice, ele mai poartă numele de mașini de calculat programabile.

Următorul nivel este cel al *calculatoarelor de buzunar (programabile într-un limbaj conversațional de nivel înalt, de regulă, BASIC.*

Avînd dimensiuni extrem de reduse, un format plat și dispunînd de o sursă de alimentare autonomă (acumulator, baterie) miniaturizată, ele întrunesc toate

calitățile cerute unor calculatoare de buzunar. Pentru afișarea caracterelor alfanumerice, cu ajutorul cărora se pot reprezenta linii de program, date, mesaje etc., se folosește un ecran cu cristale lichide de tip matricial.

Instrucțiunile și datele sînt introduse de la o tastatură alfanumerică miniaturizată, la care unele taste pot avea și o semnificație funcțională, fiind asociate cu comenzi specifice limbajului BASIC.

Capacitatea de reprezentare pe ecran este limitată la o fereastră constînd din 14—30 caractere alfanumerice, dintr-o linie de 60—80 asemenea caractere. Ecranul poate fi utilizat și în modul grafic, în unele cazuri cu posibilitate de control la nivel de punct.

În funcție de capacitatea memoriei (RAM) alocate, utilizatorului (4—10Ko) ele acceptă de la 1000, pînă la 65000 linii de program scrise în BASIC. Memoria cu conținut permanent (PROM) stochează interpretorul pentru limbajul BASIC, care dispune și de facilitate de editare.

Ca extensii pentru aceste calculatoare, în unele cazuri sînt prevăzute: interfață pentru casetofon/magnetofon, interfață pentru miniimprimantă, interfață RS-232C — pentru comunicații seriale etc.

Dintre aceste calculatoare de buzunar se pot menționa: SHARP PC 1251, CASIO FX 802P, TANDY TRS80 PC2 etc.

1.2. Microcalculatoare personale (individuale)

O primă subclasă este cea a *microcalculatoarelor portabile* avînd dimensiuni de circa 30×20×5 cm și o greutate variînd între 0,5—2 kg. Ele dispun de un ecran de afișare matricial, cu cristale lichide, de dimensiuni relativ mari, ceea ce permite afișarea unui număr mai mare de linii decît în cazul calculatoarelor de buzunar. De asemenea, tastatura folosită are dimensiunile unei tastaturi standard, ceea ce oferă posibilitatea lucrului cu ambele mîini.

Aceste microcalculatoare sînt programabile în limbajul BASIC și dispun de un interpretor stocat în memoria cu conținut permanent.

Sînt prevăzute cu alimentare autonomă sau de la rețea. Ele mai pot fi conectate la miniimprimantă și la un televizor obișnuit alb-negru sau color.

Pot fi utilizate în timpul deplasărilor, în aplicații de prelucrări de texte, bloc-notes, carnet de adrese etc.

Ca exemple de microcalculatoare portabile se pot da: SANYO TPC 8300, TEXAS INSTRUMENTS CC 40, CANON X07, CASIO FP 200, TANDY TRS80 MODEL 100.

O altă subclasă cu utilizări caracteristice o reprezintă cea a *microcalculatoarelor familiale*. Ele posedă o tastatură normală și folosesc pentru vizualizare un televizor alb-negru sau color, iar pentru stocarea externă a programelor caseta magnetică.

Aceste microcalculatoare dispun de o memorie internă de capacitate relativ mare (64 Kocteti), de o gamă largă de periferice incluzînd: miniimprimantă, casetofon, microcasetofon, manete pentru jocuri, difuzor etc. și se alimentează de la rețea.

Calculatoarele familiale sînt prevăzute cu un software destul de puternic, constînd din monitoare, editoare interpretoare pentru BASIC, compilatoare pentru o serie de limbaje evolute : PASCAL, FORTH, MICROPROLOG etc.

Utilizarea casetofonului comercial pentru introducerea și stocarea programelor prezintă unele inconveniente, datorită manierei secvențiale de lucru a acestui dispozitiv.

Aplicațiile acoperă o paletă foarte largă : învățămînt, proiectare, gestiune, supravegherea unor procese, comenzi secvențiale, jocuri etc.

Clasa mare din care fac parte aceste categorii de calculatoare (microcalculatoare) este cunoscută sub denumirea de clasa calculatoarelor personale sau individuale.

În țara noastră s-au realizat mai multe tipuri de asemenea calculatoare personale : aMIC, FELIX-Student, HC-85, Prae și DEGA-209. Pînă la data elaborării acestui text numai microcalculatorul aMIC fusese omologat și introdus în producția de serie, ceea ce explică și realizarea acestei lucrări.

Dintre microcalculatoarele personale realizate peste hotare se pot aminti : ZX81, SINCLAIR-SPECTRUM, ORIC1, DRAGON32, MULTITECH MPF, LASER 200, JUPITER AGE etc.

1.3. Microcalculatoare personal-profesionale

Microcalculatoarele profesionale-personale se plasează la nivelul cel mai înalt sub aspectul performanțelor și al costului. Realizate în formatul „desk-top” ele constau dintr-o tastatură, o unitate centrală, un monitor video (alb-negru sau color), unul sau mai multe unități de discuri flexibile pentru stocarea fișierelor, o imprimantă și eventual alte echipamente periferice nestandard. Ele sînt echipate cu microprocesoare orientate pe 8 sau 16 biți.

Avînd un caracter profesional ele se folosesc ca sisteme universale sau sisteme „la cheie” orientate pe aplicații specifice.

În țara noastră se produc în mod curent sisteme din această categorie : FELIX M118*, CUB. Terminalul pentru pregătirea datelor TPD, și FELIX-PC (recent introdus în fabricație).

Dintre sistemele din această categorie produse în alte țări se pot menționa printre altele : Apple II, COMMODORE SX 64, TANDY 4, ALPHATRONIC PC-TRIUMPH ADLER, EPSON aX10, XEROX 820-II, KAYPRO 10, MACINTOSH, LISA, LILITH, IBM-PC (mai multe variante), ADVANCE 86, ZENITH Z 150 și Z 16, HITACHI 16000, CORONA PC, DECISION V, TELE-VIDEO PC, AXEL 20, CANON AS-100, EAGLE SPIRIT, PAP TOSHIBA, TI PC, RAINBOW 100 etc.

Ultimele tipuri folosesc microprocesoare evolute INTEL 8086, 8088 sau MOTOROLA 68000. Cele care se bazează pe microprocesoarele 8086/8088 s-au aliniat în general la sistemul IBM-PC, sub aspectul compatibilității software.

* FELIX M118 a fost prezentat în lucrarea „Microcalculatoarele FELIX M18, M18 B și M118” (vol. 1 și vol. 2) E.T., 1984. Autori : A. Petrescu și colectiv IPB, ICE ș.a.

Limbajele evoluat e : BASIC, PASCAL, MODULA, PROLOG, C, FORTH și altele sînt implementate sub sistemele de operare destul de răspîndite CP/M, MSDOS etc.

În continuare se vor prezenta cîteva caracteristici ale unor microcalculatoare profesionale realizate în țara noastră.

1.3.1. Microcalculatorul profesional CUB (Calculatorul Universal de Birou), produs la Întreprinderea de Calculatoare Electronice, este constituit dintr-o unitate centrală realizată pe o singură plachetă, un monitor alfanumeric, o tastatură convențională și una sau două unități de discuri flexibile — simplă densitate.

Unitatea centrală se bazează pe microprocesorul 8080 și pe circuitele din familia acestuia. Memoria are o capacitate maximă de 64 Ko, dintre care 2—16 Ko sînt folosiți pentru monitor și programe de autotestare.

Dispozitivul de afișare asigură 24 de linii a cîte 80 caractere alfanumerice pe fiecare linie. Caracterele mari și mici sînt realizate printr-o matrice de 5x7 puncte. Caracterele pot fi afișate în video normal sau video invers și/sau cu posibilitatea de modificare a intensității.

Tastatura alfanumerică de tip QWERTY dispune de 78 taste, dintre care unele sînt asociate anumitor funcțiuni.

Memoria externă este asigurată prin una sau două unități de discuri flexibile de 5"/8", cu o capacitate de memorare de 512/1024 Ko în variantă dublă față — densitate simplă.

Opțional, microcalculatorul poate fi prevăzut cu o imprimantă matricială cu 132 coloane și cu o viteză de imprimare de 150 caractere pe secundă.

Microcalculatorul CUB este exploatat sub sistemul de operare CP/M, monoutilizator-monotask. Sub acest sistem de operare sînt implementate limbajele BASIC, PASCAL, COBOL etc. Sistemul își găsește numeroase aplicații în birotică, proiectare asistată de calculator, gestiune, învățămînt etc.

1.3.2. Terminalul de pregătire a datelor TPD, fabricat la Întreprinderea de Echipamente Periferice FEPEP, poate fi utilizat atît ca terminal inteligent cuplat la un minicalculator, cît și ca microcalculator independent. Ca structură hardware, terminalul TPD este construit cu circuite din familia 8080, dar ulterior au fost dezvoltate și alte variante constructive.

În varianta inițială TPD dispune de : o unitate centrală cu 8080 (funcționînd la frecvența de 1,8 MHz), un controlor de întreruperi 8259, canal de acces direct la memorie 8257, un controlor de ecran 8275, un controlor de disc 8271, un controlor de transmisie serială 8251, o interfață paralelă 8255 și un ceas numărător 8253.

Memoria RAM are o capacitate minimă de 32 Ko și maximă de 64 Ko. De asemenea, folosește o memorie REPR0M de 2 Ko, care conține un încărcător de sistem și un mic monitor de depanare.

Ulterior s-a înlocuit controlorul de ecran 8275 cu o schemă ce asigură și posibilitatea de utilizare în mod grafic a ecranului, cu o rezoluție de 512 x 288

puncte. În acest scop, terminalul este dotat și cu o memorie de ecran de 32 Ko, separată de memoria de program (de 64 Ko).

O altă variantă utilizează controlorul de disc de dublă densitate 8272 în locul lui 8271.

Ultima variantă a terminalului TPD utilizează un microprocesor Z80 și este realizată tehnologic pe o singură placă, iar consola ecran este de tip monitor TV.

La terminalul TPD se pot cupla mai multe tipuri de imprimante (pe interfața paralelă), cititor de cartele, ploter, unitate de bandă magnetică și linii de transmisie pe legătură serială.

Din punct de vedere software, pe TPD se pot utiliza două sisteme de operare: un sistem original FEPER și sistemul CP/M. Sistemul de operare CP/M-TPD este perfect compatibil cu CP/M-M118, putând fi utilizate toate programele existente sub CP/M. Limbajele utilizate pe TPD sub CP/M sînt: limbaj de asamblare, FORTRAN, C, BASIC, COBOL.

Pentru aplicații grafice există o bibliotecă de rutine grafice.

1.3.3. Microcalculatorul profesional-personal FELIX PC — este un nou tip de microcalculator personal-profesional bazat pe microprocesoare din generația a III-a, cu un grad de integrare tehnologică ridicat, structură compactă și un sistem de programe ce acoperă o gamă largă de aplicații.

Microsistemul este destinat utilizării individuale în aplicații profesionale de dezvoltare a programelor de bază și aplicații sau ca sistem dedicat funcțional, în aplicații specializate de complexitate ridicată.

FELIX PC are o structură compactă, cu posibilități de extensie în vederea alcătuirii unor configurații adecvate. Este alcătuit din modulul de bază și module de extensie.

Modulul de bază constituie un calculator pe o plachetă și conține următoarele resurse:

- unitate de prelucrare bazată pe microprocesoarele 8088/8086 și 8087;
- memorie RAM de 256 Ko, organizată pe 8 sau 16 biți;
- memoria EPROM de 8—64 Ko, organizată pe 8 sau 16 biți;
- cuplor pentru discuri flexibile de 8" sau 5 1/4";
- interfețe pentru:
 - tastatură;
 - casetă magnetică (audio);
 - imprimantă (serială);
 - comunicație asincronă/sincronă;
- ceas de timp real;
- numărătoare programabile;
- sistem de întreruperi;
- canale de acces direct la memorie;
- conectori pentru module de extensie.

Resursele hardware cuprinse în modulul de bază asigură funcțiile necesare utilizării ca sistem de dezvoltare universal, într-o configurație redusă, care include: discuri flexibile, imprimantă serială, tastatură, terminal alfanumeric/grafic conectat serial.

Pentru a permite o mai mare flexibilitate, modulul de bază conține 8 conectori care asigură conectarea la magistrala sistemului a unor module de ex-

tensie. În configurația standard FELIX PC include ca modul de extensie adaptorul pentru terminal grafic color cu următoarele caracteristici:

- funcționare în mod alfanumeric;
- funcționare în mod grafic.

În mod alfanumeric se asigură următoarele regimuri de funcționare:

- 25 rânduri a 40 de caractere fiecare;
- 25 rânduri a 80 de caractere fiecare.

Fiecare caracter este afișat în funcție de atributele asociate astfel:

- alb/negru;
- video direct/invers;
- intensitate mărită;
- clipire („blinking”);
- color, stabilindu-se culoarea fondului și a caracterului.

Generatorul de caractere utilizează două seturi de caractere înscrise în ROM, reprezentând setul standard ASCII și o serie de semne speciale pentru utilizarea în regim semigrafic.

În mod grafic sînt implementate următoarele regimuri de funcționare:

- rezoluție mică — 320 × 200 puncte;
- rezoluție medie — 640 × 200 puncte;
- rezoluție mare — 640 × 400 puncte.

Adaptorul pentru terminal grafic este prevăzut cu ieșire pentru cuplare la:

- monitor color cu intrări RGBI;
- monitor alb-negru/color cu intrare video complex;
- televizor alb-negru/color cu intrare prin antenă (cu modulator atașat).

Adaptorul este proiectat și implementat pe principiul „bit mapped display”.

Memoria de reimprospătare a ecranului este organizată ca o memorie biport și este plasată în spațiul de adresare al microprocesorului, oferind astfel facilități ridicate de prelucrare grafică. Corespondența biților din memoria de reimprospătare cu punctele de pe ecran este flexibilă și se alege în funcție de modul și regimul de lucru. Adaptorul pentru terminal grafic include și cuplorul pentru creion optic.

Pentru mărirea disponibilităților sistemului sînt în lucru următoarele module de extensie:

- interfața pentru imprimanta paralelă;
- interfața pentru I/E analogice (8 canale intrare și 4 canale de ieșire);
- interfața pentru I/E numerice;
- interfață specializată pentru aplicații medicale (termografie).

De asemenea, se are în vedere proiectarea unor noi module de extensie:

- cuplor pentru disc Winchester;
- modul specializat pentru culegerea și prelucrarea de semnale EKG;
- extensie pentru analiza și sinteză de voce;
- cuplor de rețea locală.

Sistemul de programe de bază și aplicații implementat pe FELIX PC are la bază sistemele de operare PC-DOS și MS-DOS și include:

- utilitățile sistemului de operare pentru interfața cu operatorul, gestiunea și întreținerea fișierelor, funcții de bază accesibile prin program, programe de test etc.;

- facilități de execuție și depanare a programelor ;
- translațoare pentru programe în limbaj de asamblare și în limbaj BASIC ;
- interpretor de BASIC cu extensii pentru prelucrări grafice ;
- mediu de dezvoltare a programelor în MODULA 2 ;
- mediu de dezvoltare a programelor în UCSD-PASCAL ;
- programe de aplicații pentru :
 - prelucrări grafice ;
 - editarea și prelucrarea textelor ;
 - baze de date ;
 - culegerea și validarea datelor ;
 - aplicații economice.

Sistemele de operare PC-DOS și MS-DOS sînt compatibile între ele și sînt principal asemănătoare cu CP/M.

Sistemul FELIX PC este introdus în fabricație la întreprinderea de calculatoare electronice București. Datorită soluțiilor tehnologice ce vizează implementarea sistemului, este de așteptat ca fiabilitatea acestuia să fie ridicată, constituind o alternativă pentru diverse aplicații industriale. Este în curs de elaborare o astfel de aplicație pentru conducerea roboților industriali.

Compatibilitatea cu microsistemele similare cu o largă răspîndire cum ar fi: IBM PC, SANYO 550, OLIVETTI M24, CORONA etc. oferă o mare disponibilitate de software.

1.4. Caracteristicile tehnice și comerciale ale unor calculatoare de buzunar, calculatoare personale și calculatoare personal-profesionale străine

Calculatoare de buzunar programabile în limbaj de nivel înalt.

SHARP PC 1251

Caracteristici generale :

- dimensiuni : 13,5 × 7 × 0,9 cm
- greutate : 115 g.
- alimentare : două baterii de 1,5 V, cu Litriu sau de la rețea, pentru extensii imprimantă, casetofon etc.).

Memoria :

- cu conținut variabil (nevolatil) : 4,2 Ko,
- cu conținut permanent : 24 Ko.

Afișare :

- cristale lichide,
- o linie cu 24 de caractere,
- opt indicatori.

Tastatura :

- miniaturizată,
- organizare : QWERTY-majuscule;

— 18 taste alfabetic programabile în modul RESERVE, accesibile prin SHIFT,

— bloc numeric.

Limbaj :

- BASIC,
- editor performant,
- linii program : 1—999, cu 79 semne pe linie,
- variabile : numerice (nume A)—precizie : 7 cifre zecimale ; șiruri de caractere (nume AS)—lungime 7 caractere ; tablo-uri (nume a,AS)—dimensiune : 2,
- mesaje de eroare : 9.

Observație : poate fi utilizat drept calculator de buzunar științific pentru calcule obișnuite.

Extensii :

- imprimantă ;
- casetofon.

Cost : 1400 FF în configurația de bază.

TANDY TRS 80 PCE

Caracteristici generale :

- dimensiuni : 19,5 × 8,6 × 2,5 cm ;
- greutate : 375 g ;
- alimentare : patru baterii de 1,5 V, adaptor de rețea pentru extensii (imprimantă, casetofon).

Memoria :

- cu conținut variabil : 2,6 Ko, extensibilă la 10 Kocteți ;
- cu conținut permanent : 16 Ko.

Afișare :

- cristale lichide ;
- imprimanta cu 4 culori ;
- o linie cu 26 de caractere ;
- 14 indicatori ;
- grafice : 7 × 156 puncte pe ecran, 216 × 4096 puncte pe imprimantă ; texte și grafice mixabile.

Tastatura :

- miniaturizată ;
- organizare : QWERTY-majuscule, minuscule ;
- 10 taste alfabetice funcționale pentru instrucțiuni BASIC ;
- 18 funcții programabile pe 6 taste ;
- bloc numeric ;
- caracterele grafice se pot defini pe întregul ecran.

Limbaaj :

- BASIC ;
- editor performant ;
- linii de program : 1—65000, cu 8 caractere pe linie ;
- variabile : numerice (nume A1)—precizie : 10 cifre zecimale, șiruri de caractere (nume A1 \$)—lungime : 80 de semne, tablouri (nume A1, A1 \$)—dimensiuni ; 2 ;
- mesaje de eroare ; 40 (codificate).

Observație : poate fi utilizat drept calculator de buzunar științific, pentru calcule obișnuite.

Extensii :

- imprimanta cu 4 culori ;
- casetofon,
- interfață serială RS 232 C.

Cost : 1800 FF. în configurația de bază.

Calculatoare portabile.

TEXAS INSTRUMENTS CC 40

Caracteristici generale :

- dimensiuni : 24—14,5 × 2,4 cm ;
- greutate : 600 g ;

— alimentare : patru baterii de 1,5 V, adaptor rețea.

Memorie :

- cu conținut variabil : 6—22 Ko ;
- cu conținut permanent : 34 Ko.

Afișare :

- cristale lichide ;
- o linie cu 31 de caractere ;
- 18 indicatori, dintre care 6 sînt controlați de utilizator.

Tastatura :

- normală ;
- organizare : QWERTY-majuscule și minuscule,
- 30 taste pentru instrucțiuni BASIC ;
- bloc numeric cu taste programabile ;
- alfabet japonez, caractere grecești, 7 caractere pot fi definite de utilizator.

Limbaaj :

- BASIC ;
- editor performant ;
- linii de program : 1-32766, cu 80 caractere pe linie ;
- variabile : numerice (nume : AB...N)—lungime : 255 caractere ; tablouri (nume : AB...N)—dimensiuni ; 3,
- mesaje de eroare : 75 în clar, 29 codificate,
- alte limbaje : asamblor integrat, Pascal.

Extensii :

- imprimanta cu 4 culori ;
- cititor de cartuș magnetic ;
- interfață serială RS 232 C ;
- interfață paralelă ;
- interfață video.

Cost : 2750 FF.

TANDY TRS MODEL 100

Caracteristici generale :

- dimensiuni : 30 × 21,5 × 5 cm ;
- greutate : 1,36 kg ;
- alimentare : patru baterii de 1,5 V, acumulator Cd-Ni, adaptor de rețea.

Memoria :

- cu conținut variabil : 8—32 Ko,
- cu conținut permanent : 32 Ko.

Afișare :

- 8 linii cu 40 caractere ;
- grafica : 240 × 64 puncte.

Tastatura :

- normală ;
- organizare : QWERTY-majuscule și minuscule ;

— 8 taste funcționale pentru software-ul integrat, redefinibile în BASIC,
— bloc numeric integrat, caractere grafice.

Limbaaj :

— BASIC,
— editor performant,
— linii de program : 1-65529, cu maximum 255 caractere pe linie ;
— variabile : numerice—simplă precizie : +32767/-32768, numerice—dublă precizie : 14 cifre zecimale, șiruri de caractere—lungime 255 caractere, tablouri de dimensiuni nelimitate ;
— mesaje de eroare : 32 codificate ;
— software integrat : prelucrare de texte, agendă, carnet de adrese, teleprelucrare.

Extensii :

— casetofon ;
— interfață pentru imprimanta Centronics, RS 232 C, modem și cititor de cod de bare.

Cost : 5995 FF.

Calculatoare familiale.

SINCLAIR SPECTRUM.

Caracteristici generale :

— dimensiuni : 23,3 × 14,4 × 3 cm ;
— greutate : neprecizată ;
— alimentare : adaptor de rețea.

Memoria :

— cu conținut variabil : 16—48 Ko, din care : 8 — 40 Ko sînt disponibili pentru utilizator ;

— cu conținut permanent : 16 Ko.

Afișare :

— televizor alb/negru sau color : PAL, Peritel sau SECAM, cu intrare prin antenă ;
— opt culori la alegere pentru chenar, „hirtie” și „cerneală” ;

— 22 de linii, cu 32 de caractere pe linie plus o zonă de lucru la baza ecranului extensibilă la 22 de linii ;

— grafică : 256 × 192 puncte (texte și grafice mixabile) ;

— video-invers, două niveluri de luminozitate, superpoziție, afișare intermitentă.

Tastatura :

— normală ;
— organizare : QWERTY-majuscule, minuscule ;
— instrucțiunile BASIC sînt asociate cu taste unice, modul de acces la

taste determină automat poziția cursorului pe linie ;

— 16 caractere grafice și 21 caractere definibile de către utilizator.

Limbaaj :

— BASIC ;
— editor extrem de performant ;
— linii de program : 1-9999, cu 704 caractere pe linie ;
— variabile : numerice (nume fără restricții), precizie : 9-10 cifre zecimale, șiruri de caractere (nume : A \$)—lungime nelimitată, tablouri (nume : A \$)—dimensiuni nelimitate.

— mesaje de eroare : 29 în clar ;
— alte limbaaje (pe caseta magnetică) : asamblor/dezasamblor, Pascal, Forth, Microprolog.

Extensii :

— magistrală externă cu linii de date, adrese și comenzi ;
— interfața serială : RS232 C și Centronics ;

— memorie de masă (Microdrive-100 Ko) ;

— imprimantă termică.

Cost : 1480-2325 FF.

ORIC 1.

Caracteristici generale :

— dimensiuni : 28 × 17,5 × 5,2 cm ;
— greutate : 1,1 kg ;
— alimentare : adaptor de rețea.

Memoria :

— cu conținut variabil : 16—48 Ko, din care, la capacitatea maximă de 48 Ko, pentru utilizator sînt disponibili 47 Ko, în modul text și 39 Ko în modul cu rezoluție ridicată ;

— cu conținut permanent : 16 Ko.

Afișare :

— televizor alb/negru sau color : PAL, Peritel, SECAM ;
— 8 culori la alegere pentru cadru și „hirtie” ;

— 27 de linii cu 38 caractere pe linie ;

— grafică : 39 × 27 puncte (rezoluție redusă), 240 × 200 și 3 linii de text (rezoluție ridicată), grafice și texte miscibile ;
— video-invers, afișare intermitentă, linii duble.

Tastatura :

— normală,
— organizare : QWERTY-majuscule, minuscule ;
— 80 caractere grafice, care pot fi definite de utilizator.

Limbaj :

- BASIC ;
- editor cu posibilități modeste ;
- linii de program : 1-64000, cu 78 caractere pe linie ;
- variabile : numerice (nume : A1) — precizie : 9 cifre zecimale, șiruri de caractere (nume : A1 \$) — lungime neprecizată, tablouri (nume A1, A1 \$) — dimensiuni nelimitate ;

- mesaje de eroare : 20 necodificate ;
- alte limbaje (pe caseta magnetică) : asamblor/dezasamblor, Forth.

Extensii :

- magistrala externă cu linii de date, adrese și comenzi ;
- interfața Centronics încorporată ;
- imprimantă ;
- microdisc.

Cost : 2000—2500 FF, în funcție de configurație.

Calculatoare personal-profesionale.

IBM PC jr.

Caracteristici generale :

- dimensiuni : 35 × 29 × 9,65 cm (unitatea de bază) și 34,29 × 16,76 × 2,5 cm (tastatura) ;

- greutate : 4,2 kg ;
- alimentare : de la rețea, unitatea centrală și celelalte periferice, cu excepția tastaturii, care se alimentează de la baterii, nefiind conectată prin cablu cu unitatea centrală.

Memoria :

- cu conținut variabil : 64 Ko ;
- cu conținut permanent : 64 Ko, extensibilă până la 128 Kocteți ;

Microprocesor :

- Intel 8088.

Afișare :

- televizor color sau monitor RGB ;
- 16 culori la rezoluția : 320 × 200 puncte ;

- 4 culori la rezoluția : 640 × 200 puncte ;

Tastatura :

- normală, cu taste separate nemarcate (marcarea se face pe spațiile dintre taste, cu marcaj șanjabil) ;

- alimentare la baterii ;

- fără legături fizice cu unitatea centrală.

Software :

- sistem de operare : PC-DOS 2.10 ;
- limbaje : BASIC- în cartuș ROM, LOGO etc ;

- programe de aplicații : Home Word (pentru prelucrări de texte), Word Star.

Interfețe și periferice standard :

- interfață serială RS-232 C ;
- interfața video (40 coloane) pentru monitor RGB sau receptor TV, cu modulator pentru semnal video-complex ;

- generator de semnale acustice.

Extensii :

- unitate de disc flexibil 5 1/4", dublă față, dublă densitate (360 Ko/disc) ;

- memorie RAM, 64 Ko pentru opțiunea video-80 coloane ;

- adaptor pentru imprimantă paralelă ;

- modem : 300 biți/s.

Cost : \$599—\$999 în funcție de configurație.

MACINTOSH.

Caracteristici generale :

- dimensiuni : 34,30 × 24,64 × 27,70 cm (unitatea de bază constind din : display, unitatea centrală și unitatea de disc flexibil) și 6,6 × 33,53 × 14,73 cm — tastatura ;

- greutate : 8,5 kg ;

- alimentare de la rețea.

Memorie :

- cu conținut variabil : 128 Ko ;

- cu conținut permanent : 64 Ko.

Microprocesor :

- Motorola 68000.

Afișare :

- monitor încorporat cu diagonală de 22,85 cm ;

- rezoluție : 512 × 342 puncte,

- control la nivel de bit.

Tastatura :

- normală ;

- organizare : QWERTY, standard ;

- detașabilă.

Software :

- sistem de operare : FINDER ;

- limbaje : Mac FORTH, Microsoft

BASIC ;

- programe de aplicații : Mac Write, Mac Paint, Multiplan.

Interfețe și periferice standard :

- indicator de tip „mouse” ;

- unitate de disc flexibil-3,5", capacitate : 400 Kocteți ;

- generator de semnale acustice ;

- două interfețe seriale RS-422 A ;

- interfață pentru o unitate suplimentară de disc flexibil ;

- magistrală serială sincronă pentru tastatură.

Extensii :

- imprimantă matricială ;

- tastatura numerică ;

- unități de disc flexibil.

Cost : \$ 2495.

Prezentarea generală a microcalculatorului „aMIC”

2.1. Componente și scheme bloc

Microcalculatorul „aMIC” (fig. 2.1) face parte din categoria microcalculatoarelor personale (individuale), destinate acoperirii unei largi game de aplicații, în condițiile unor performanțe superioare și al unui cost relativ scăzut.

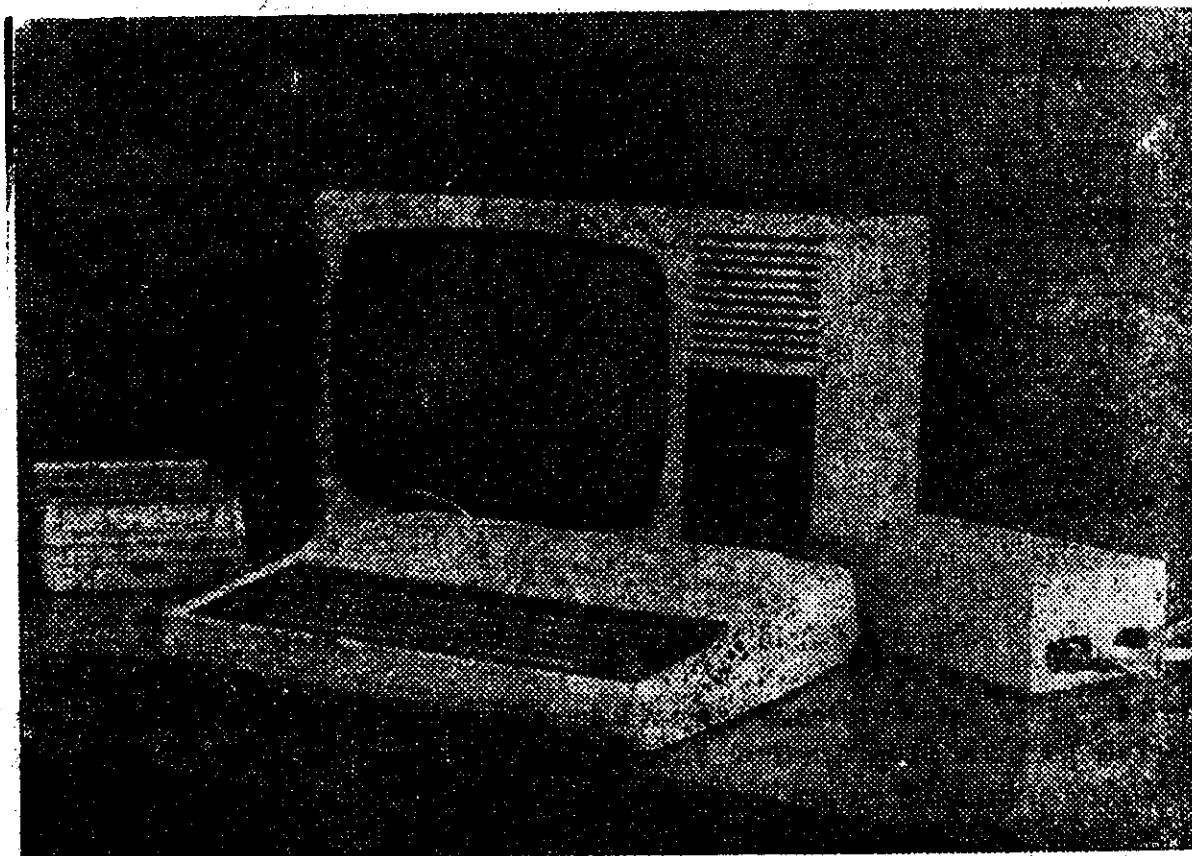


Fig. 2.1. Microcalculatorul „aMIC” (foto).

La proiectarea și realizarea sa tehnologică s-au avut în vedere o serie de factori, privind folosirea cu precădere a circuitelor integrate produse în țara noastră și a unor echipamente periferice din gama bunurilor de larg consum :

televizorul alb-negru * și casetofonul audio. De asemenea, s-a urmărit ca acest produs să reprezinte un sistem *deschis* sub aspectul hardware-ului, software-ului și al aplicațiilor. Acesta permite cuplarea unor periferice destinate creșterii performanțelor și lărgirii gamei aplicațiilor: disc flexibil, înregistrator X—Y, imprimantă, cuplor de proces etc.

Dezvoltările software se referă la extinderea și perfecționarea monitoarelor, asamblorilor, interpretoarelor și compilatoarelor de limbaje universale și specializate de nivel înalt.

Sistemul „aMIC” este organizat (Fig. 2.2) în jurul unei magistrale, care conține liniile de date, adrese, comenzi și alimentare. Aceste linii sînt disponibile la un conector extern, cu 50 de contacte, ceea ce permite cuplarea unor periferice evaluate cu acces direct la memorie (unitate de disc flexibil) sau a unor periferice nestandard. Semnalele sînt descrise în capitolul 3. Această magistrală

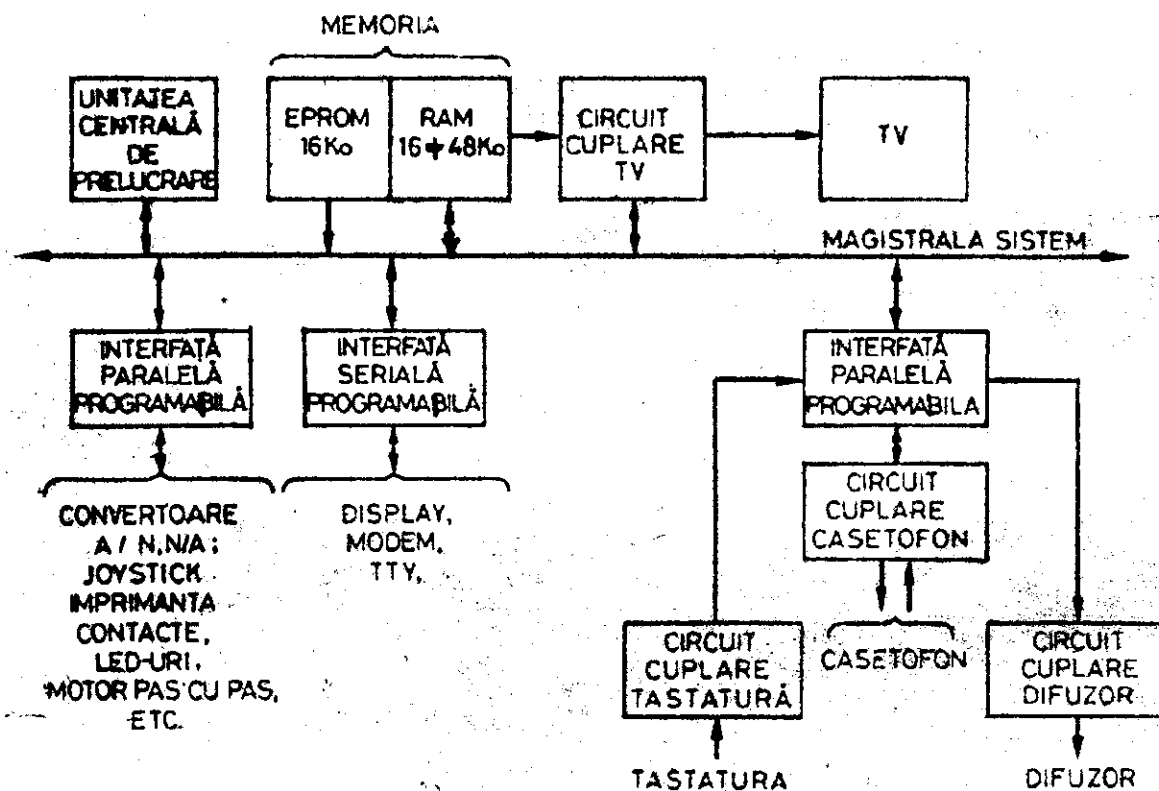


Fig. 2.2. Organizarea microcalculatorului „aMIC”.

asigură legătura între unitatea centrală de prelucrare, memoriile EPROM-RAM, interfețele paralele programabile, interfața serială programabilă și cuplorul TV.

Unitatea centrală de prelucrare se bazează pe microprocesorul Z80, funcționând la frecvență de 2,5 MHz.

Memoria EPROM, care conține monitorul și interpretorul limbajului BASIC sau monitorul, asamblorul și editorul de texte, are o capacitate de 16 Ko și folosește circuitele 2716.

Memoria RAM, destinată programelor de aplicații, este realizată cu circuite dinamice 4116 și asigură o capacitate maximă de 48 Ko.

*) Varianta color se află în curs de asimilare de către industrie.

*Interfața paralelă programabilă ** are un caracter opțional și este realizată cu un circuit 8255. Ea se folosește pentru conectarea unor echipamente convenționale sau a unor echipamente nestandard. Astfel, se pot menționa : convertorul A/N-N/A, Joy-stick-ul, imprimanta, contacte, LED-uri, circuite de comandă a unui motor pas cu pas etc.

*Interfața serială programabilă **, este opțională și se bazează pe circuitul 8251. Ea este utilizată pentru cuplarea unor echipamente cu transmisie serială : display, MODEM, TTY, eventual alt calculator prevăzut cu interfață serială.

Cuplorul TV asigură generarea semnalului video complex modulat, pe baza conținutului memoriei de ecran, cu o capacitate de 8 Ko., care face parte tot din memoria RAM a sistemului.

Pentru introducerea comenzilor, instrucțiunilor și a datelor în sistem se folosește o *tastatură elastică, ultraplă, cu martor sonor (difuzor), dispunând de 59 de taste*. În principal organizarea tastaturii corespunde convenției QWERTY



Fig. 2.3. Tastatura microcalculatorului „aMIC“ (foto).

pentru caracterele alfanumerice (fig. 2.3). A fost prevăzut un set de 16 caractere semigrafice, care pot fi afișate în video normal sau video invers ca și celelalte caractere alfanumerice. Introducerea caracterelor prezente în colțul stînga sus pe fiecare tastă se realizează acționînd simultan Tasta SHIFT și Tasta cu codul

* Pentru programarea interfețelor paralele și seriale, în vederea conectării diverselor echipamente, sînt prezentate în cap. 8 o serie de exemple. De asemenea, se poate consulta lucrarea : Microcalculatoarele FELIX M18, M18B, M118, vol. I, Editura Tehnică, 1984, autori : A. Petrescu și colectiv.

dorit. În figura 2.4 se prezintă caracterele semigrafice și codificarea lor hexazecimală. Trecerea la afișarea video-invers se asigură prin acționarea simultană a tastelor CTRL și E. Tasta RESET generează condiția de inițializare a sistemului, trecerea sub controlul programului de sistem numit „monitor” și afișarea în video normal. Tasta INT permite generarea unor întreruperi de la tastatură, care pot fi tratate prin programe speciale.

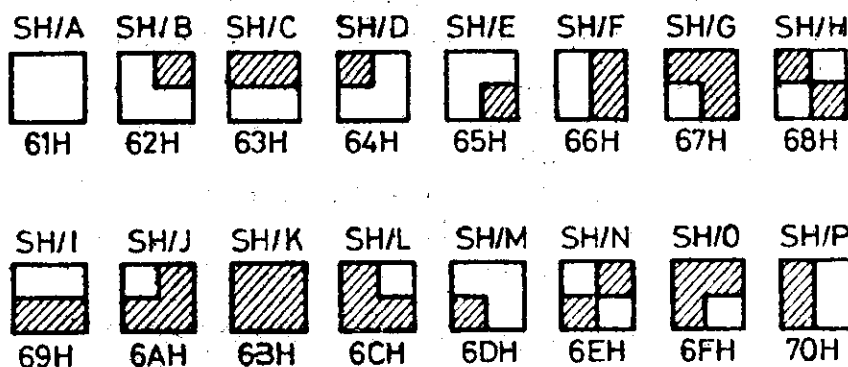


Fig. 2.4. Caracterele semigrafice.

Afișarea informației alfanumerice, semigrafice și grafice este asigurată cu ajutorul unui televizor comercial alb/negru. Pentru reprezentări grafice rezoluția ecranului este de 256×256 puncte. În regimul alfanumeric se afișează 32 de rânduri, a câte 30 caractere pe rând*. Generatorul de caractere programat permite afișarea setului standard de 64 caractere ASCII și a setului de caractere semigrafice menționate mai sus. La cerere, setul de caractere poate fi modificat.

Cuplarea televizorului la calculator se realizează cu ajutorul unui cablu coaxial, prin intrarea de antenă, modulatorul fiind acordat în banda II VHF, canalele 6—12.

Stocarea programelor elaborate în cod mașină, limbaj de asamblare sau BASIC se face pe casetă magnetică obișnuită, folosind un casetofon comercial. Viteza de transfer a informației este de circa 1 600 bauds, ceea ce permite încărcarea sau stocarea unor programe relativ lungi într-un interval de timp suficient de mic. Deși s-au luat măsuri speciale pentru amplificarea semnalelor, se impune stabilirea unui volum optim al semnalului la casetofon, atât la redare, cât și la înregistrare. Se va căuta ca, pe cât este posibil, să se folosească mufe separate pentru conectarea la casetofon în cazul citirii, respectiv al scrierii (în cazul în care nu se folosește casetofonul furnizat de către producătorul sistemului de calcul).

2.2. Software de bază : monitoare, asamblor, interpretor BASIC

Spre deosebire de alte sisteme de calcul individuale din aceeași clasă, la care utilizatorul operează direct cu o „mașină BASIC”, microsistemul „aMIC” dispune de un Monitor, rezident în memoria EPROM, care asigură interpretarea și execuția comenzilor introduse de la tastatură.

* În cadrul versiunii V.01 a monitorului „aMIC”.

Monitorul este constituit dintr-o colecție de rutine, care pot fi apelate, atât de la tastatură, cât și de programele scrise de către utilizator. Intrarea în Monitor se realizează automat, la aplicarea tensiunii de alimentare sau pe parcursul utilizării calculatorului, acționând tasta RESET. Când sistemul se află sub controlul Monitorului, pe ecran se afișează, în colțul stînga sus mesajul AMIC. Pe rîndul următor, sub mesajul AMIC, apare un punct urmat de cursor, care este reprezentat sub forma unei linii cu afișare intermitentă. Aceasta indică poziția pe ecran la care se va înscrie următorul caracter introdus de la tastatură. În continuare Monitorul așteaptă comenzi. *Pînă în prezent au fost scrise trei versiuni ale Monitorului „aMIC”. Versiunea restrînsă V0.1 ocupă 2 Ko de memorie. Versiunea extinsă V0.2 dispune de facilități suplimentare și ocupă 2,5 Ko de memorie. Monitorul, care are înglobate un asamblor și un editor de fișiere create în memorie (MATE), ocupă 6 Ko de memorie.*

Avînd în vedere posibilitatea reprogramării memoriilor EPROM, cât și faptul că acestea sînt plasate pe socluri în calculator este posibilă scrierea unor monitoare orientate pe aplicații specifice. În cazul unor aplicații dedicate, chiar programul utilizatorului poate fi înscris în EPROM, folosindu-se numai 16 Ko de memorie RAM pentru : afișare pe ecran (8 Ko) și manipularea variabilelor (8 Ko).

2.2.1. Monitorul „aMIC” V0.1 * (sumar, in extenso în § 5.1) are următoarele comenzi :

- D — afișarea pe ecran a conținutului unei zone de memorie,
- F — încărcarea unei zone de memorie cu o constantă,
- M — deplasarea conținutului unei zone de memorie în altă zonă de memorie,
- C — modificarea registrelor interne ale utilizatorului,
- X — afișarea registrelor interne ale utilizatorului,
- S — afișarea și modificarea conținutului unor locații de memorie,
- G — lansarea în execuție a unui program obiect aflat în memorie,
- K — salvarea unui fișier din memorie, pe casetă magnetică,
- L — citirea în memorie a unui fișier de pe caseta magnetică,
- B — lansarea în execuție a interpretorului limbajului BASIC.

Unele dintre aceste comenzi necesită parametri numerici reprezentând adrese (patru cifre hexazecimale) sau constante (două cifre hexazecimale).

Monitorul „aMIC” versiunea 0.1 ocupă 2 Ko în memoria EPROM, fiind plasat la adresele 0000H-07FFH. El este descris pe larg în capitolul 5 al lucrării.

Comenzile de mai sus asigură introducerea unor programe în cod obiect, depanarea lor și lansarea în execuție. În acest mod pot fi controlate deosebit de eficient toate resursele hardware ale calculatorului în scopul depanării și elaborării unor programe de aplicații extrem de performante.

2.2.2. Monitorul „aMIC” V0.2 (sumar, in extenso, în § 5.2) constituie o versiune extinsă față de V0.1, oferind o viteză mai mare de execuție a rutinelor sale și o condensare a codului, datorită utilizării întregului set de instrucțiuni ale microprocesorului Z80.

* Este scris în subsetul de instrucțiuni al microprocesorului Z80 compatibil direct, de jos în sus, cu setul de instrucțiuni al microprocesorului 8080.

Această versiune se caracterizează prin următoarele :

- modificarea definiției caracterelor, ceea ce permite afișarea a 40 caractere pe rând ;
- atribuirea de nume fișierelor pe casetă magnetică pentru a efectua operații de citire, scriere și verificare a fișierelor pe baza numelui asociat ;
- introducerea funcțiilor utilizator pentru manipularea facilă a rutinelor din Monitor, care gestionează perifericele sistemului ; funcțiile utilizator sînt standardizate conform sistemului de operare CP/M V2.2, ceea ce permite execuția pe calculatorul „aMIC“ a unor programe dezvoltate pe alte sisteme sub CP/M ;
- implementarea unor noi comenzi privind scrierea și citirea unor fișiere în format hexa la interfața serială.

Spațiul ocupat în memoria EPROM de acest Monitor depinde de numărul funcțiilor utilizator implementate. Versiunea V0.2 ocupă circa 2,5 Ko în memoria EPROM, începînd cu adresa 0000H. Spațiul de la sfîrșitul Monitorului pînă la 0FFFH este destinat dezvoltărilor ulterioare. Programele utilizatorului rezidente în EPROM pot ocupa 12 Ko începînd cu adresa 1000H.

Monitorul „aMIC“ V0.2 are următoarele comenzi :

- D — afișarea pe ecran a conținutului unei zone de memorie,
- F — încărcarea unei zone de memorie cu o constantă,
- M — deplasarea conținutului unei zone de memorie în alte zone de memorie,
- X — examinarea și modificarea registrelor interne ale microprocesorului Z80,
- S — afișarea și modificarea conținutului unor locații de memorie,
- G — lansarea în execuție a unui program obiect aflat în memorie,
- C — comparare a conținutului a două zone de memorie,
- K — salvarea unui fișier din memorie pe caseta magnetică ;
- L — citirea în memorie a unui fișier de pe caseta magnetică ;
- N — afișarea conținutului antetului de fișier de pe caseta magnetică ;
- R — citirea unui bloc de date în format hexa de la interfața serială ;
- V — compararea conținutului unei zone de memorie cu conținutul unui fișier de pe caseta magnetică ;
- W — scrierea unui bloc de date în format hexa la interfața serială.

Unele comenzi necesită parametri sub forma unor adrese sau constante reprezentate în coduri hexazecimale.

2.2.3. Monitorul Z80-V0.0 (sumar, in extenso in § 5.3) reprezintă o versiune de monitor scrisă cu instrucțiunile specifice microprocesorului Z80 și asigură următoarele funcțiuni :

- afișarea/modificarea unor zone de memorie RAM ;
- afișarea conținutului registrelor microprocesorului ;
- lansarea în execuție a programelor ;
- posibilitatea lucrului cu întreruperi software în faza de depanare a programelor ;
- salvarea unor zone de memorie sub forma de fișiere pe caseta magnetică ;
- încărcarea de fișiere de pe casetă în memoria RAM ;

Spațiul de memorie EPROM ocupat de acest monitor este de cca 3 Ko

Monitorul Z80-V0.0 are următoarele comenzi :

- I — inserare șir octeți ;
- V — vizualizare conținut zonă memorie demarcată prin adresa inferioară și superioară ;
- G — lansare în execuție program ;

- F — umplere zonă memorie cu o constantă ;
- M — deplasare zona memorie ;
- Y — comparare zona memorie ;
- S, D — suma, diferența ;
- R — inițializare mod de lucru cu întreruperile programabile ;
- B — programare întrerupere la o adresă dată ;
- C — relansare program întrerupt ;
- T — trasare program ;
- D — dezactivare întreruperi ;
- X — afișare conținut registre ;
- K — salvare zona memorie pe caseta ;
- A — listare antete fișier ;
- L Q Z — încărcare fișier de pe caseta la diverse adrese

2.2.4. Monitorul DEST (sumar, in extenso in Cap. 6). DEST (monitor Dezvoltare Software și Testare) reprezintă un monitor de dezvoltare software și testare pentru sisteme care folosesc microprocesorul Z80.

Monitorul oferă următoarele posibilități de lucru :

- crearea și modificarea fișierelor sursă în limbaj de asamblare,
- asamblarea de fișiere sursă și crearea de module obiect relocabile sau absolute,
- editarea și legarea mai multor module obiect relocabile într-un singur modul, acesta devenind modul obiect absolut,
- dezasamblarea codului obiect din orice zonă de memorie, listarea sursei și memorarea sub forma de fișier pentru prelucrări ulterioare,
- execuția pas cu pas a programului,
- încărcarea datelor/salvarea datelor de la/pe caseta magnetică.

Facilitățile enumerate mai sus asigură realizarea cerințelor necesare unui sistem de dezvoltare pentru software.

Configurația minimă pentru testare-depanare necesită 8 Kocteți de memorie EPROM și 16 Ko. de memorie RAM, iar configurația necesară dezvoltării de aplicații necesită 16 Ko. EPROM și 16—48 Ko. RAM, împreună cu perifericele : casetofon și miniimprimantă.

2.2.5. Monitor—Asamblor—Text Editor (MATE) poate fi considerat un sistem de operare de capacitate și posibilități limitate, rezident din memoria EPROM. El asigură editarea, asamblarea, depanarea și execuția unor programe sursă, scrise în limbajul de asamblare al microprocesorului 8080. Programele sînt tratate ca fișiere create în memorie, cărora li se atribuie câte un nume. În cazul în care sînt mai multe fișiere în memorie, fișierul cu care se lucrează poartă numele de fișier curent.

Fișierele sînt organizate pe linii, fiecare linie fiind identificată printr-un număr N ($0000 \leq N \leq 9999$ în zecimal).

Editorul permite încărcarea informațiilor structurate pe linii în fișiere și modificarea conținutului liniilor. O linie poate conține cel mult 80 de caractere.

Asamblorul permite generarea codului obiect pentru programele editate sub formă de fișiere. Fișierul obiect astfel creat poate fi lansat în execuție. Asamblorul manipulează constante zecimale, hexazecimale, expresii, pseudoinstrucțiuni etc. El oferă o serie de mesaje de eroare.

Fișierele sursă sau obiect din memorie pot fi salvate pe casetă magnetică sau pot fi restaurate în memorie prin citirea lor de pe caseta magnetică.

Comenzile Monitorului MATE sînt :

ASSM	— assemblează un program sursă,
BREK	— poziționează sau șterge puncte de întrerupere (suspendare) în programul care se va executa,
CTRL-X	— abandonează linia curentă,
DELT	— șterge linii dintr-un fișier,
DUMP	— afișează conținutul memoriei,
ENTR	— introduce date în memorie,
EXEC	— lansează în execuție un program,
FILE	— creează, distruge, activează un fișier sau afișează informații referitoare la un fișier,
LIST	— listează conținutul unui fișier,
LOAD	— citește în memorie un fișier de pe caseta magnetică,
PAGE	— deplasează o pagină (zonă) de memorie,
PROC	— relansează în execuție un program oprit într-un punct de întrerupere (suspendare),
SAVE	— încarcă pe casetă un fișier din memorie,
YYYY	— cheamă editorul de fișiere ($0 \leq Y \leq 9$).

Modulul monitor posedă un singur mesaj de eroare (...WHAT?), care indică o comandă eronată sau folosirea incorectă a parametrilor unei comenzi.

MATE este descris pe larg în capitolul 7 al lucrării.

2.2.6. Interpretorul pentru limbajul BASIC a fost implementat pînă în prezent în două versiuni *. Prima variantă constituie un subset al celei de-a doua în sensul că nu dispune de instrucțiuni referitoare la matrici, prelucrare grafică și operația CALL. Versiunea redusă este realizată ca un interpretor care ocupă 8 Ko de memorie EPROM, în timp ce versiunea extinsă ocupă 14 Ko de memorie.

Interpretoarele BASIC implementate nu utilizează o formă intermediară a programului, începînd de fiecare dată execuția de la forma sursă. Ca urmare a execuției programului, utilizatorul nu va dispune de codul obiect al programului, ci de rezultatele execuției acestuia.

Interpretorul BASIC este stocat în memoria EPROM începînd de la adresa fixă 0800H. Lansarea sa în execuție, din Monitor, se recunoaște prin afișarea pe ecran a mesajului READY, ceea ce indică faptul că sistemul așteaptă comenzi sau instrucțiuni de la utilizator.

Pentru editarea programelor au fost introduse facilități de corecție a unei linii în timpul introducerii sale de la tastatură sau de editare a programului deja introdus, prin ștergerea sau înlocuirea unor linii.

În vederea evaluării rapide a limbajului BASIC extins, în continuare este prezentat sub forma unui memento.

2.2.7. Limbajul BASIC — memento (în cap. 9, din vol. 2, în extenso). Numerele sînt considerate reale și reprezentate în formatul cu virgulă mobilă avînd 6—7 cifre semnificative. Toate variabilele numerice sînt reale. Numele variabilelor simple este format dintr-o literă sau o literă și o cifră, iar al tablourilor (care pot avea una sau două dimensiuni) dintr-o literă. Indicii tablourilor sînt cuprinși între 1 și 254.

* În curs de implementare se află noi versiuni de BASIC, care urmăresc compatibilizarea cu versiunile instalate pe alte microcalculatoare individuale de largă răspîndire sau pentru aplicații specifice de supraveghere și conducere a proceselor industriale.

Numele unui șir constă dintr-o literă urmată de semnul \$. Se pot utiliza tablouri de șiruri, toate șirurile componente avînd aceeași dimensiune, specificată în instrucțiunea DIM.

Pot fi utilizate subșiruri, specificarea unui subșir realizîndu-se cu notația (e1TOe2), atașată numelui variabilei șir, unde e1, e2 sînt expresii ale căror valori reprezintă poziția primului și, respectiv, a ultimului caracter al subșirului, din șirul dat.

Expresiile e1 și/sau e2 pot să lipsească. În acest caz se vor lua implicit primul caracter și respectiv ultimul al șirului.

Funcții

Sintaxă	Rezultat
ABS(X)	Valoarea absolută.
ATN(X)	Arctangentă din X (X în radiani).
CHR\$(X)	Caracterul al cărui cod este X.
COS(X)	Cosinus din X (X în radiani).
EXP(X)	e^x
EE	Constanta e (baza logaritmilor naturali).
GET(X)	Valoarea citită de la portul X ($0 \leq X \leq 255$).
INKEY\$	Caracterul introdus de la tastatură sau șirul vid, dacă nu s-a acționat nici o tastă.
INT(X)	Partea întregă din X.
LEN(X\$)	Lungimea șirului X\$.
LOG(X)	Logaritmul natural din X.
PI	Constanta π (3.14159265...).
POW(X)	Se utilizează numai în membrul stîng al instrucțiunii de atribuire. Transmite la portul X, ($0 \leq X \leq 255$), valoarea expresiei din membrul drept convertită în întreg pe un octet (eventual prin trunchiere).
RND(X)	Generează un număr aleator în intervalul (0,1).
SGN(X)	Signum: - 1 pentru $X < 0$, 0 pentru $X = 0$ și 1 pentru $X > 0$.
SIN(X)	Sinus din X (X în radiani).
SQR(X)	Rădăcina pătrată din X.
STR\$(X)	Șirul de caractere care ar fi afișat, dacă X ar fi tipărit cu PRINT.
VAL(X\$)	Evaluează șirul X\$, privit ca o expresie numerică.
AT(X, Y)	Se utilizează în instrucțiunea PRINT pentru a indica linia X și coloana Y, în care se dorește să se tipărească ($1 \leq X \leq 32$), ($1 \leq Y \leq 30$).
X\$(XTOY)	Subșirul format din caracterele X pînă la Y, din șirul X\$. Dacă X sau Y lipsesc, se consideră că subșirul începe cu primul caracter și respectiv se termină cu ultimul caracter din X\$.
CON	Inițializează o matrice cu valoarea 1.
IDN	Inițializează o matrice cu valoarea 1 pe diagonala principală (sau cvasidiagonală) și zero în rest.
INV(A)	Inversa matricei A.
TRN(A)	Transpusa matricei A.
ZER	Inițializează o matrice cu valoarea zero.

Operatori

- Scădere (binar sau unar).
- ⊕ Adunarea (inclusiv pentru matrici).
- * Înmulțire (inclusiv pentru matrici).
- / Împărțire.
- ↑ Ridicare la putere.

$\left. \begin{array}{l} =, >, < \\ = <, < = \\ = >, > = \\ < >, < > \end{array} \right\}$	<p>Operatori relaționali ce pot fi utilizați în instrucțiunea IF. Mărimile comparate trebuie să fie de același tip : numeric sau șir</p>
--	--

Comenzi

GOTO	n	Execută programul începând de la linia n.
LIST	m, n	Afișează instrucțiunile programului cu numerele de linie cuprinse între m și n. În cazul absenței parametrilor se listează programul în întregime.
LOAD		Încarcă de pe casetă în memoria internă un program împreună cu variabilele utilizate.
RUN	n	Inițializează variabilele programului și lansează execuția începând cu linia n (sau în absența parametrului n se începe cu prima linie).
SAVE		Depune programul împreună cu variabilele utilizate pe casetă.
SCR		Șterge programul din memorie.

Instrucțiuni

CALL(N,X,Y,...)	Apelează subrutina în limbaj mașină (Z80) cu numărul N. ($0 \leq N \leq 254$). X, Y sînt parametri utilizați de sub rutină.
DATA C1, C2, ...	Definește constante numerice sau șiruri.
DIM A(m, n), ...	Definește tablouri de variabile numerice.
DIM B\$(m,n) ...	Definește tablouri de șiruri și le inițializează cu zerouri, respectiv cu spații.
DRAW X,Y	Desenează o linie din punctul grafic curent, în punctul de coordonate X, Y.
END	Oprește execuția programului. Este ultima instrucțiune din program.
FOR I=XTOY FOR I=XTOY STEP Z	Instrucțiuni de ciclare. I este variabila de control, X valoarea inițială, Y valoarea finală și Z pasul (X, Y, Z valori numerice reale).
NEXT I	
GOTO n	Salt la execuția instrucțiunii n. Este singura instrucțiune ce poate fi utilizată și sub formă de comandă.
GOSUB n	Salt la execuția subrutinei care începe la linia n.
RETURN	Instrucțiune utilizată pentru revenirea din subrutina.
IF X>Y THEN n IF X\$≤Y\$ THEN n	Dacă relația dintre cele două mărimi este adevărată, se execută instrucțiunea de la linia n, altfel se continuă cu instrucțiunea următoare lui IF.
INIT P	
INPUT X,X\$, ...	Șterge ecranul și eventual îl comută în alt mod de lucru (defilare/pagină).
LET X=expresie X=expresie X\$=șir	Citește de la tastatură valori pentru variabilele specificate.
ON X GOSUB n1, n2, ON X GOTO n1, n2, ...	Instrucțiunea de atribuire. Asociază unei variabile o valoare.
MOVE X,Y	Se evaluează X (care poate fi expresie) și i se calculează partea întreagă, $n = \text{INT}(X)$. Se trece apoi la execuția instrucțiunii cu numărul (eticheta) nk. Dacă k este mai mare decît numărul de etichete specificate, atunci nu se execută saltul.
PLOT X,Y	Punctul grafic va avea coordonatele X, Y. Nu se afișează nimic. Pe ecran pot fi afișate 256×256 puncte grafice.
PRINT X,X\$, ...	„Aprinde” pătrățelul de coordonate X, Y ($0 \leq X \leq 63$), ($0 \leq Y \leq 63$). Un pătrățel are 16 puncte grafice.
	Afișează valorile expresiilor numerice sau șir, specificate în instrucțiune. Tratează separatorii : „,“ ;” și AT(X, Y).

READ X,X\$,	Citește valori pentru variabilele specificate. Valorile sînt luate din instrucțiunile DATA, din program.
REM	Permite introducerea de comentarii într-un program.
RESTORE	Instrucțiune utilizată în conjuncție cu READ și DATA, pentru a permite recitirea constantelor din instrucțiunile DATA.
ROTATE U	Permite rotația cu unghiul U (în radiani) a vectorilor generați cu RDRAW sau a poziționărilor realizate cu RMOVE
RMOVE X,Y	Punctul grafic va fi deplasat cu X pe orizontală și Y pe verticală față de poziția curentă. Nu afișează nimic.
RDRAW X,Y	Generează un vector din punctul curent, pînă în punctul de coordonate X,Y relative la punctul curent.
STOP	Oprește execuția programului.
SCALE X,Y	Permite definirea scării de reprezentare grafică pe orizontală și verticală.
UNPLOT X,Y	Șterge pătrățelul de coordonate X,Y. ($0 \leq X \leq 63$), ($0 \leq Y \leq 63$).
VIEWPORT X1,X2, Y1,Y2	Definește zona din ecran pe care va avea loc afișarea grafică (spațiul fizic).
WINDOW X1,X2, Y1,Y 2	Definește limitele între care pot varia coordonatele punctelor ce vor avea imagine pe ecran (spațiu utilizator).
MAT INPUT A,B, ... MAT READ A,B, ... MAT PRINT A,B, ...	} Instrucțiuni care permit citirea și scrierea tablourilor numerice fără specificarea individuală a elementelor componente.

2.3. Configurații disponibile la desfacere

Sistemul de calcul „aMIC“ poate fi livrat în diverse configurații funcționale, impuse de tipurile aplicațiilor avute în vedere.

În cazul limită inferior se poate folosi numai placheta cu cablaj imprimat, avînd implantate circuitele necesare pentru a realiza structura de resurse hardware solicitate într-o aplicație dată. Astfel, introdusă într-un echipament mai complex, placheta de bază își pierde identitatea.

Intr-o configurație extinsă sistemul este livrabil actualmente cu următoarele componente :

- Microcalculator „aMIC“ (cu mufe; TV; CAS; Alimentare).
- Memorie fixă 16 Ko EPROM : monitor, interpretor BASIC extins.
- Memorie utilizator 48 Ko. RAM.
- Televizor (TV), cablu de legătură cu microcalculatorul.
- Casetofon (CAS), cablu de legătură cu microcalculatorul.
- Imprimantă, conector periferic pentru imprimantă, cablu de legătură cu microcalculatorul.
- Sursa de alimentare, cablu de legătură cu microcalculatorul.
- Conector interfață serială.
- Conector legături externe.

Pentru a veni în sprijinul celor care solicită microsiseme „aMIC“ în diverse variante, se prezintă în continuare codificarea resurselor hardware.

84039-1.0. Microcalculator individual, compus din următoarele subansamble :

— **84039-PE-1.0.** Placheta echipată „aMIC“, constând din circuitul imprimat 84039PE1.1 pe care se implantează componentele electronice și cablurile spre mufele de conexiuni și tastatură.

Modulele funcționale existente pe placheta care reprezintă un „microcalculator pe o singură plachetă“ sînt următoarele :

— unitate centrală de prelucrare cu microprocesor Z80,
 — memorie RAM, cu circuite dinamice tip 4116,
 — memorie EPROM, cu circuite tip 2716,
 — interfața paralelă programabilă bazată pe circuitul tip 8255, care asigură următoarele funcțiuni :

- interfața cu tastatura,
- interfața cu casetofonul audio,
- generarea semnalului video complex,
- generarea semnalului pentru amplificatorul audio și difuzor,
- interfața cu receptorul TV,
- interfața de comunicație serială, realizată cu circuitul 8251,
- interfața cu miniprintanta, realizată cu circuitul tip 8255.

Pe o plachetă cu conectori, dispusă în partea posterioară a carcasei microcalculatorului, se fixează cablurile de legătură cu diversele periferice :

- **84039-S** pentru mufa de alimentare a sursei,
- **84039-C** pentru casetofon.
- **84039-T** pentru televizor,
- **84039-E** pentru magistrala externă,
- **84039-M** pentru miniprintantă,
- **84039-O** pentru interfața serială.

În figura 2.5 se prezintă forma, dimensiunile și elementele microcalculatorului „aMIC“, exceptînd sursa, perifericele și cablurile de legătură.

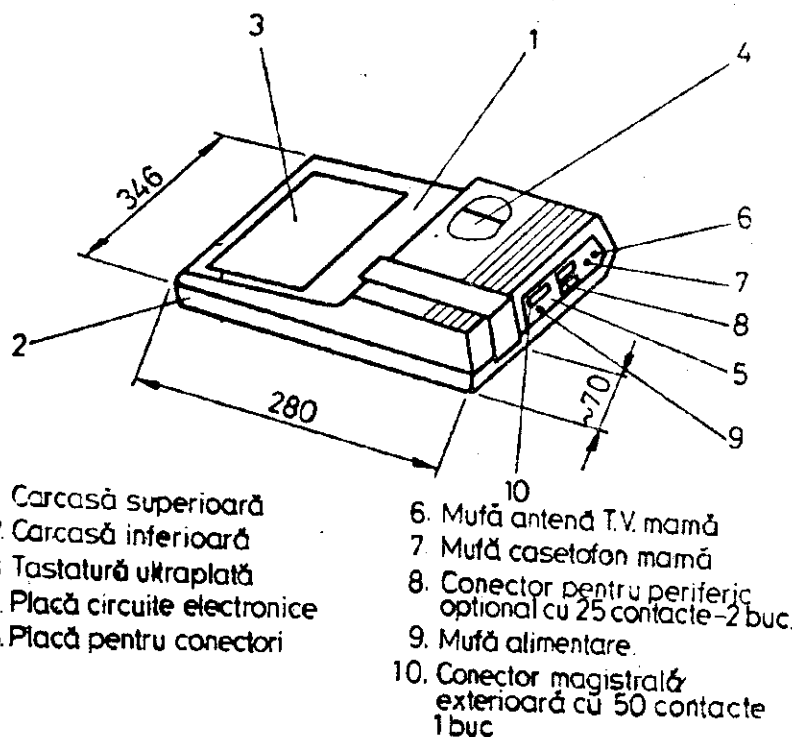


Fig. 2.5. Forma și dimensiunile microcalculatorului „aMIC“.

- . 84039-2.0. Sursa de alimentare externă, care furnizează tensiunile de alimentare de $\pm 5\text{ V}$, $\pm 12\text{ V}$.
- . 84039-3.0. Casetofon audio (cu cablu de legătură).
- . 84039-4.0. Receptor TV alb-negru (cu cablu de legătură).
- . 84039-5.0. Miniimprimantă.

EXEMPLUL 1=84039-A M I I O C O C

EXEMPLUL 2=84039-A 0 O I O O O C

EXEMPLUL 3=84039-A .0.000.00C

84039 0.0.000 000 ← CODIFICARE

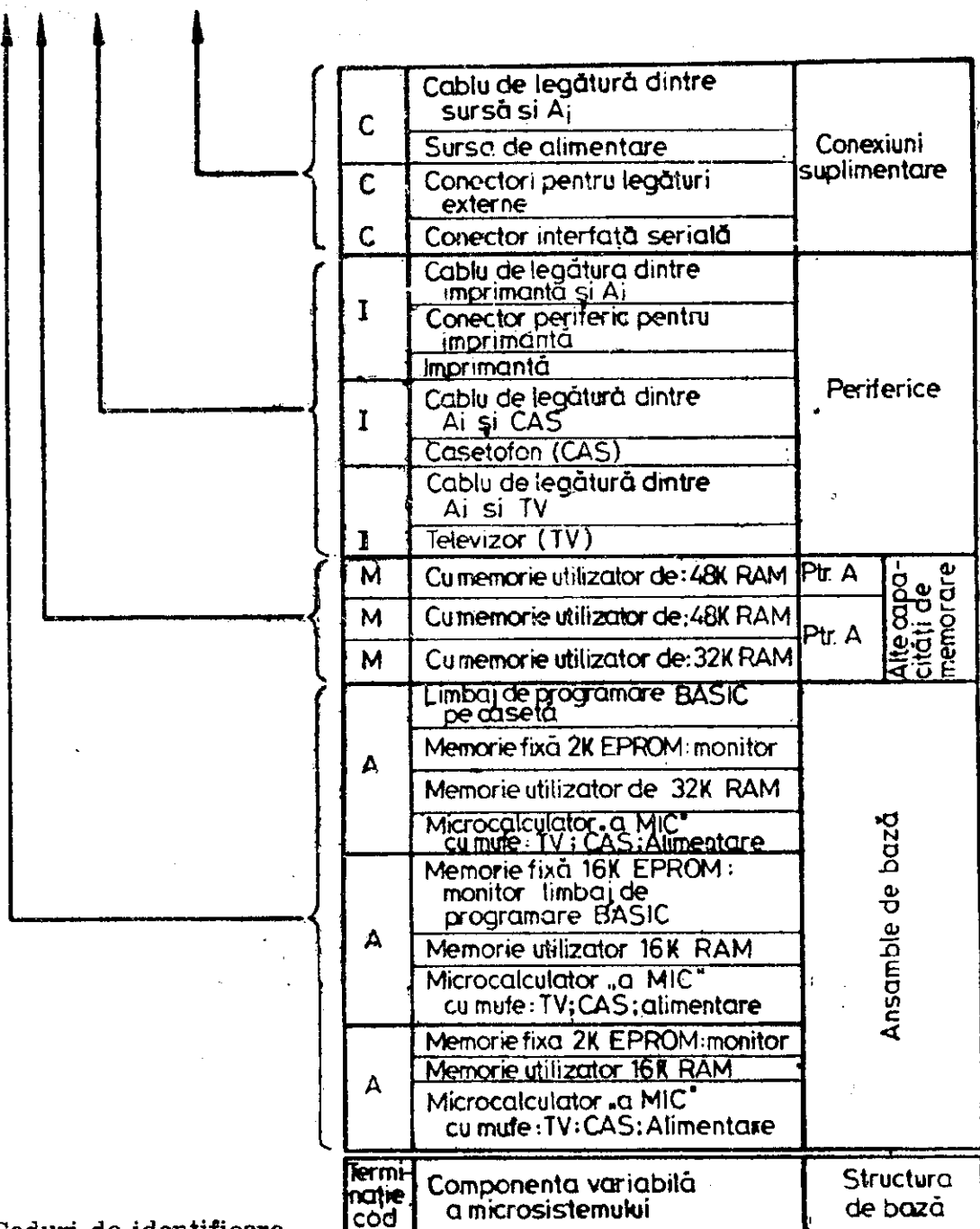


Fig. 2.6. Coduri de identificare.

În figura 2.6 se prezintă modul de codificare a configurațiilor solicitate de utilizatori pentru diverse aplicații.

Cea mai redusă configurație livrabilă are codificarea 84039-A1.0,0,000,000. Ea este utilizată cu casetofon, televizor și sursă furnizate de către beneficiar, cu programe livrate la cererea acestuia.

Configurația de bază apreciată ca uzuală cuprinde :

- microcalculator 84039-1.0 (A2 sau A3 fig. 2.6),
- sursa de alimentare 84039-2.0, cu cablu 84039-S,
- casetofon audio 84039-3.0, cu cablu 84039-C,
- receptor TV alb/negru 84039-4.0, cu cablu 84039 T.

În cazul în care beneficiarul dispune de receptor TV și/sau casetofon și/sau sursă de alimentare, ansamblele respective nu se vor mai livra.

Structura și funcționarea microcalculatorului „aMIC”

3.1. Generalități

Microcalculatorul personal se prezintă sub forma unui sistem pe o singură plachetă, la care se conectează următoarele echipamente :

- tastatură pentru introducerea comenzilor și datelor ;
- televizor pentru afișarea informațiilor ;
- casetofon audio pentru salvarea programelor din memoria internă și refacerea ulterioară a acestora ;

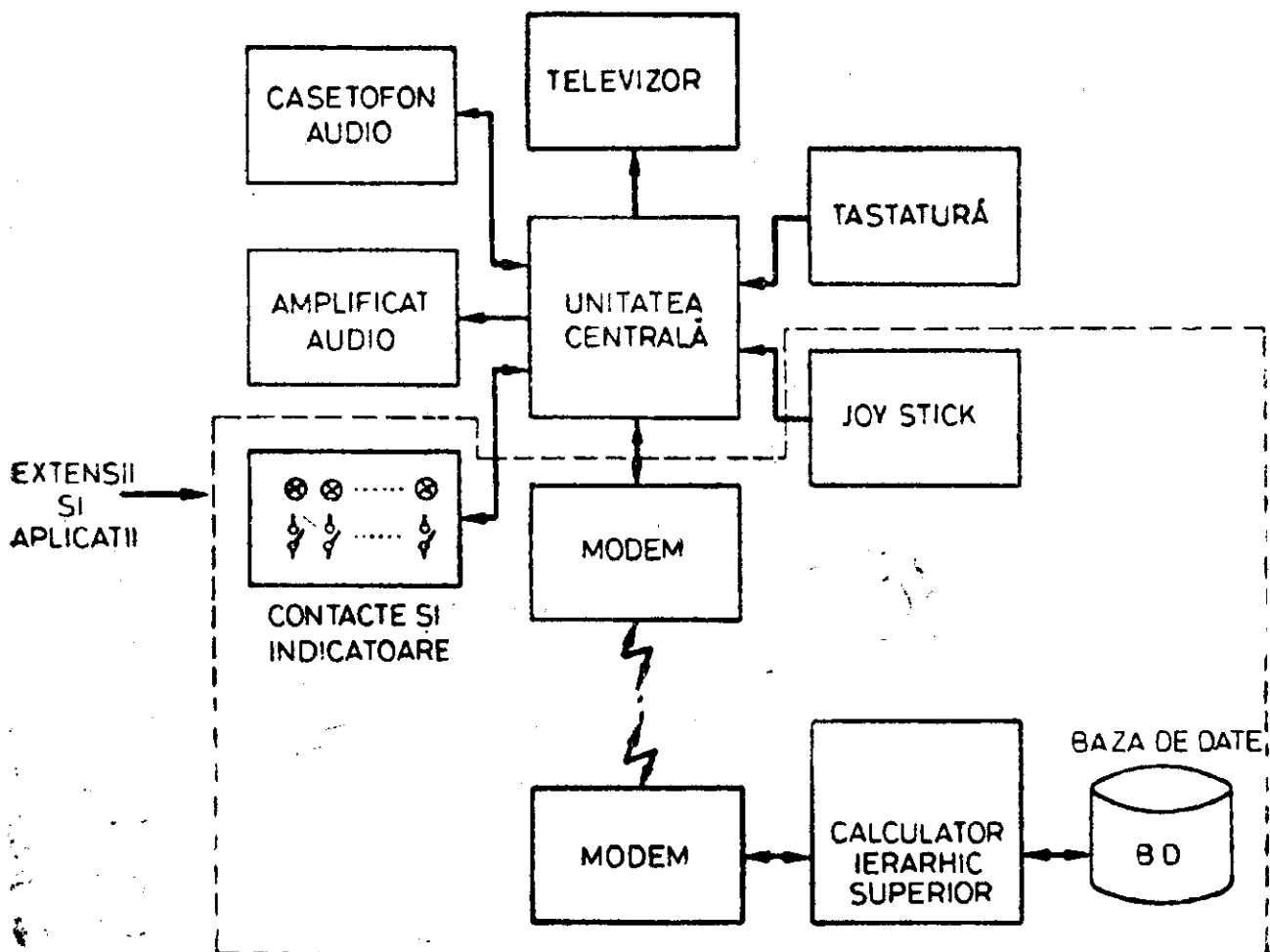


Fig. 3.1. Structura microcalculatorului personal.

- modem pentru transmiterea/recepționarea datelor pe linie telefonică ;
- joystick, dispozitiv pentru interacționarea directă între utilizator și ecranul televizorului în modul de lucru grafic ;
- amplificator audio și difuzor pentru diverse aplicații acustice.

Structura microcalculatorului personal este prezentată în figura 3.1. Unitatea centrală cuprinde microprocesorul, memoria internă și circuitele de interfață, la care se conectează echipamentele periferice. Prin intermediul unor porturi de intrare/ieșire microcalculatorul personal poate să controleze un proces simplu. În figura 3.1 această posibilitate s-a reprezentat printr-un dispozitiv cu LED-uri și comutatoare. Sistemul poate să citească nivele logice (starea unor contacte) și să comande dispozitive numerice (LED-uri).

Placheta cu unitatea centrală împreună cu tastatura se află introduse într-o carcasă. Utilizatorul are acces la claviatură și butoanele pentru întrerupere și reset (inițializare). De asemenea, s-au prevăzut mufe pentru semnalul video complex, semnalul video modulat, înregistrare/redare casetofon audio și conectori pentru comunicație serială și porturi de intrare/ieșire. Circuitul imprimat al unității centrale grupează liniile magistralei sistemului pentru furnizarea în exterior a acestora și conectarea unei extensii de memorie sau interfațarea unor echipamente periferice. Semnalele de la conectorul de magistrală de pe placa de circuit imprimat, precum și pinii, sînt dați în lista următoare :

Nr. pin	Semnal	Nr. pin	Semnal	Nr. pin	Semnal	Nr. pin	Semnal
1	<u>GND</u>	11	<u>BUSACKB</u>	21	AB8	31	DB2
2	<u>X\emptysetB</u>	12	<u>INT</u>	22	AB9	32	DB3
3	<u>RESET</u>	13	AB \emptyset	23	AB1 \emptyset	33	DB4
4	<u>WAIT1</u>	14	AB1	24	AB11	34	DB5
5	<u>MIB</u>	15	AB2	25	AB12	35	DB6
6	<u>WRB</u>	16	AB3	26	AB13	36	DB7
7	<u>RDB</u>	17	AB4	27	AB14	37	<u>RFSH</u>
8	<u>IOREQB</u>	18	AB5	28	AB15	38	+5V
9	<u>MREQB</u>	19	AB6	29	DB \emptyset	39	-5V
10	<u>BUSREQ</u>	20	AB7	30	DB1	40	+12V

De asemenea, pe circuitul imprimat se află un alt conector care furnizează semnalele pentru modulatorul TV (informație și sincronizare), difuzor audio, casetofon audio și semnalele pentru selecția circuitelor de comunicație serială și interfață paralelă (aceste circuite sînt dispuse în exterior, pe o placă suplimentară). Sînt disponibile și o serie de semnale neutilizate de la circuitul 8255, care pot fi folosite extern. Acest conector și pinii sînt listați în continuare :

Nr. pin	Semnal	Nr. pin	Semnal	Nr. pin	Semnal	Nr. pin	Semnal
1	PB \emptyset	6	PC3	11	<u>SLC</u>	16	-5V
2	PB1	7	PC4	12	<u>USART</u>	17	+12V
3	PB2	8	PC6	13	<u>PPI2</u>		
4	PB3	9	PC7	14	+5V		
5	PB4	10	<u>INF</u>	15	GND		

Tot pe circuitul imprimat se află o zonă universală liberă, la dispoziția utilizatorului, pentru eventuale modificări sau pentru introducerea unor circuite suplimentare.

Schima bloc a microcalculatorului personal este prezentată în figura 3.2.

Structura este modulară și se compune din :

- unitatea centrală de prelucrare;
- memoria EPROM;
- memoria RAM;
- logica de afișare la televizor;
- interfața periferică programabilă;
- interfața de comunicație serială.

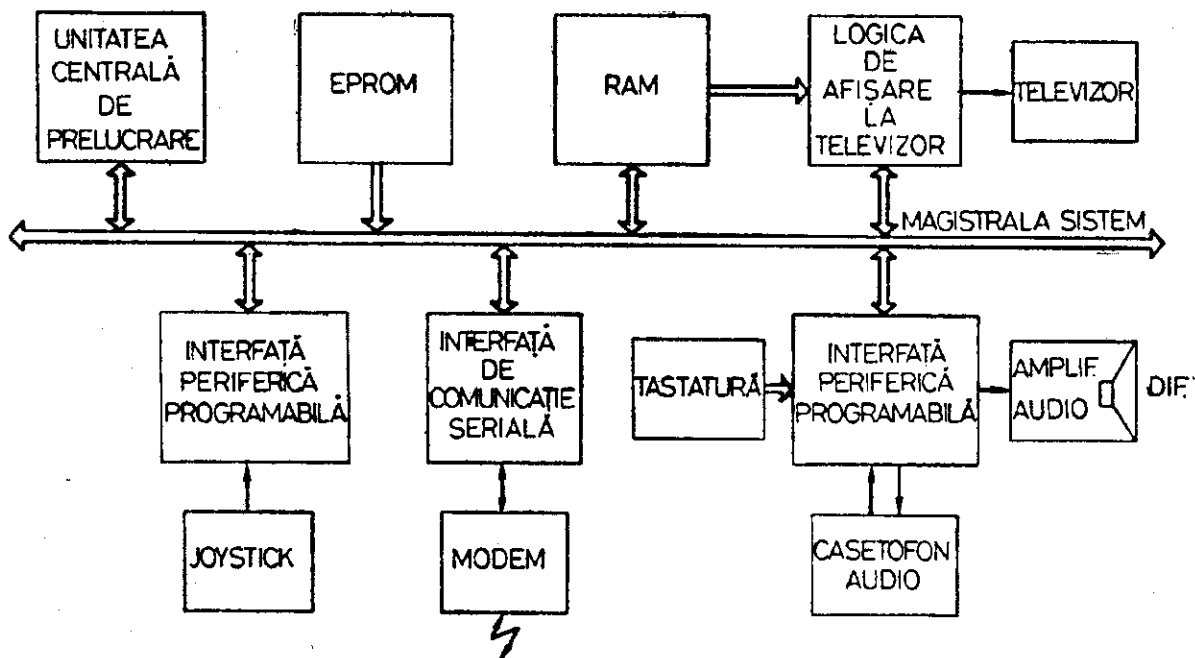


Fig. 3.2. Schema bloc a microcalculatorului personal.

Toate aceste module sînt conectate la o magistrală unică care conține :

- 16 linii de adrese AB₀—AB₁₅ ;
- 8 linii de date DB₀—DB₇ ;
- 8 linii de comenzi : MREQB, IOREQB, RDB, WRB, WAIT, INT, NMI, M1 ;
- 5 linii de alimentare : +5 V, -5 V, +12 V, -12 V, masă.

Unitatea centrală de prelucrare (UCP) este singurul modul master din sistem, deținînd în permanență controlul magistralei. Modulul UCP este construit pe baza microprocesorului Z80. Poate adresa direct 64 Kcuvinte de memorie și 256 de porturi de intrare/ieșire.

Memoria EPROM este realizată cu circuite 2716, de 2 Ko, realizîndu-se o capacitate maximă de 16 Kocteți (8 cipuri). Conține sistemul de operare rezident ; monitorul și interpretorul BASIC. Zona de memorie ocupată de EPROM este cuprinsă între adresele 0000H-3FFFH.

Memoria RAM este realizată cu circuite dinamice 4116, de 16 Kbiți, realizîndu-se o capacitate maximă de 48 Ko (24 cipuri). Zona ocupată de RAM este cuprinsă între adresele 4000H-FFFFH. Există trei module distincte, fiecare de cîte 16 Ko, primul între adresele 4000H-7FFFH, al doilea între 8000H-

BFFFH, iar al treilea între C000H-FFFFH. Memoria video (memoria ecran) este inclusă în primul modul, între adresele 4000H-5FFFH și are capacitatea de 8 Ko.

Televizorul este un terminal grafic cu rezoluția ecranului de 256×256 de puncte. Există o corespondență biunivocă între biții din memoria de imagine și punctele de pe ecran. Utilizatorul avînd acces la această memorie poate programa oricare din puncte să fie aprins sau stins. În regim alfanumeric se pot afișa 32 de rînduri a cîte 30 de caractere, generatorul de caractere fiind inclus în monitorul microcalculatorului personal.

4000	4001	4002	4003	4004	----	----	401D	401E	401F
4020	4021	4022	4023	----	----	----	403E	403F	
4040	4041	4042	----	----	----	----	405E	405F	
4060	4061	4062	----	----	----	----	407E	407F	
4080	4081	----	----	----	----	----	----	409F	
----	----	----	----	----	----	----	----	----	----
5F80	5F81	----	----	----	----	----	5F9F		
5FA0	5FA1	5FA2	----	----	----	----	5FBE	5FBF	
5FC0	5FC1	5FC2	5FC3	----	----	5FDD	5FDE	5FDF	
5FE0	5FE1	5FE2	5FE3	----	----	5FFD	5FFE	5FFF	

Fig. 3.3. Memoria ecran.

Corespondența între adresele trimise de microprocesor și octeții din memoria ecran este prezentată în figura 3.3. Bitul 7 din octetul de informație se afișează în stînga, iar bitul 0 în dreapta. De asemenea, un bit egal cu 0 din memorie înseamnă punct aprins pe ecran, iar bit egal cu 1, punct stins.

Logica de afișare la televizor realizează citirea permanentă a memoriei ecran, serializează informația, amestecă semnalele de sincrolinii, sincrocadre și stingere și trimite semnalul sincrocomplex la televizor.

Interfața periferică programabilă * folosește un circuit 8255 care realizează mai multe funcții :

- interfață pentru tastatură ;
- interfață pentru casetofon ;
- generator de semnal pentru amplificatorul audio ;
- generator de semnal pentru video invers.

* Se află tratate în capitolul 2 al lucrării „Microcalculatoarele FELIX M18, M18B, M118”. Ed. Tehnică — 1984, autori : A. Petrescu și colectiv.

*Microcalculatorul personal posedă o a doua interfață periferică programabilă (un al doilea circuit 8255) ** cu ajutorul căreia se poate controla un proces simplu, sau se pot cupla diverse echipamente: joy-stick, convertor analog/numeric, convertor numeric/analogic etc.

*Interfața de comunicație serială ** este realizată cu circuitul 8251 și permite cuplarea sistemului la un alt calculator, direct sau prin modem și linie telefonică. Viteza de transmisie/recepție a datelor este selectabilă între valorile 300 Baud, 600 Baud și 1200 Baud.

Adresele porturilor de intrare/ieșire sînt următoarele:

- 00H: intrare/ieșire date pentru interfața de comunicație serială (8251);
- 01H: comenzi/stări pentru 8251;
- 20H: portul A din circuitul 8255;
- 21H: portul B din circuitul 8255;
- 22H: portul C din circuitul 8255;
- 23H: portul de comandă din circuitul 8255;
- 40H: portul A din al doilea circuit 8255;
- 41H: portul B din al doilea circuit 8255;
- 42H: portul C din al doilea circuit 8255;
- 43H: portul de comandă din al doilea circuit 8255.

3.2. Unitatea centrală de prelucrare

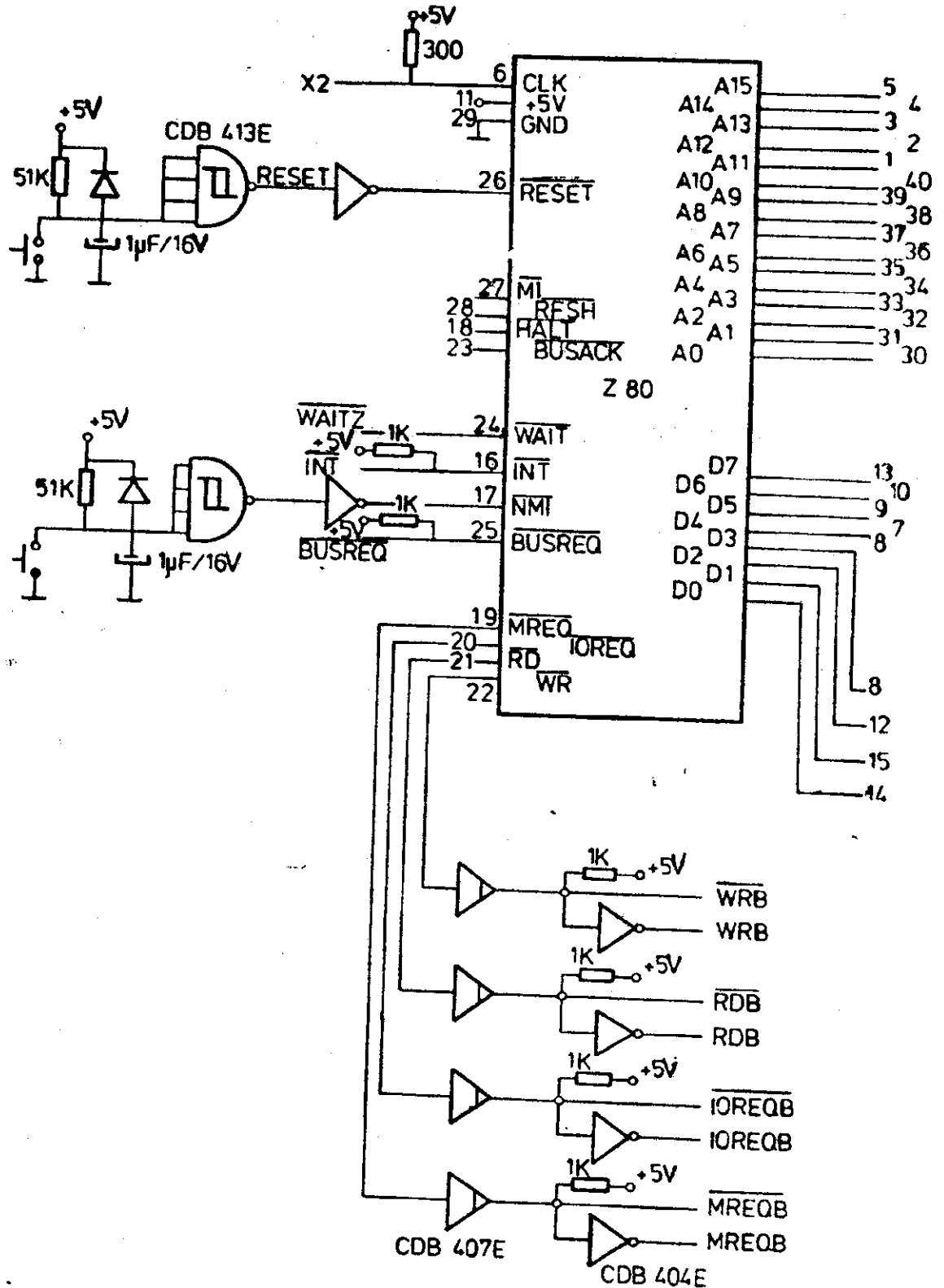
Unitatea centrală de prelucrare (fig. 3.4) se bazează pe un microprocesor Z80, la care se adaugă o serie de circuite logice pentru conectarea la magistrala sistemului. Pentru buna funcționare a microprocesorului, pe lângă tensiunea de alimentare de +5 V și masă, trebuie să i se furnizeze un semnal de ceas cu frecvența maximă de 2,5 MHz. Acesta este semnalul X2, preluat de la sincrogenerator, avînd perioada de 400 ns, deci exact frecvența de 2,5 MHz.

Un buton cu revenire, de pe carcasa microcalculatorului personal, acționat de utilizator, poate furniza impulsuri negative singulare de resetare a microprocesorului. Semnalul de la comutator este conectat pe intrarea RESET a lui Z80 prin intermediul unui circuit 7413 (trigger Schmidt). Un al doilea comutator cu revenire, aflat de asemenea pe carcasă, este conectat pe intrarea de întrerupere nemascabilă NMI. Această întrerupere nu poate fi dezactivată prin program de către utilizator și de aceea este acceptată oricînd de către microprocesor. Activarea intrării NMI, printr-un impuls negativ are ca efect un salt în program la adresa 0066H, unde se află subrutina de tratare a întreruperii.

Pentru a realiza sincronizarea vitezei microprocesorului cu cea a memoriei interne se utilizează o logică simplă pentru generarea semnalului WAIT, de trecere în starea de așteptare. Această logică este descrisă în paragraful 3.3 și urmărește suspendarea activității microprocesorului în timpul execuției unei instrucțiuni cu referire la memoria cu acces aleator (RAM), pînă cînd citirea sau scrierea este permisă.

* Se află tratate în capitolul 2 al lucrării „Microcalculatoarele FELIX M18, M18B, M118”. Ed. Tehnică—1984, autori A. Petrescu și colectiv.

Celelalte două intrări de comandă BUSREQ, cerere de magistrală și INT, cerere de întrerupere cu posibilități de mascare prin program, sînt dezactivate în actuala configurație a unității centrale de prelucrare, fiind conectate prin intermediul unei rezistențe de $1K\Omega$ la tensiunea de $+5V$ (nivel logic ridicat).



Ieșirile microprocesorului Z80 au un fan-out (sarcină totală) scăzut, ceea ce necesită utilizarea unor circuite tampon. Astfel, tensiunea furnizată de o ieșire în starea 0 logic este $V_{OL}=0,4\text{ V}$ (valoarea maximă, prevăzută în catalog) la un curent $I_{OL}=1,8\text{ mA}$, iar în starea 1 logic este $V_{OH}=2,4\text{ V}$ (valoare minimă) la un curent $I_{OH}=250\text{ }\mu\text{A}$. Bufferarea semnalelor de adresă A0-A15

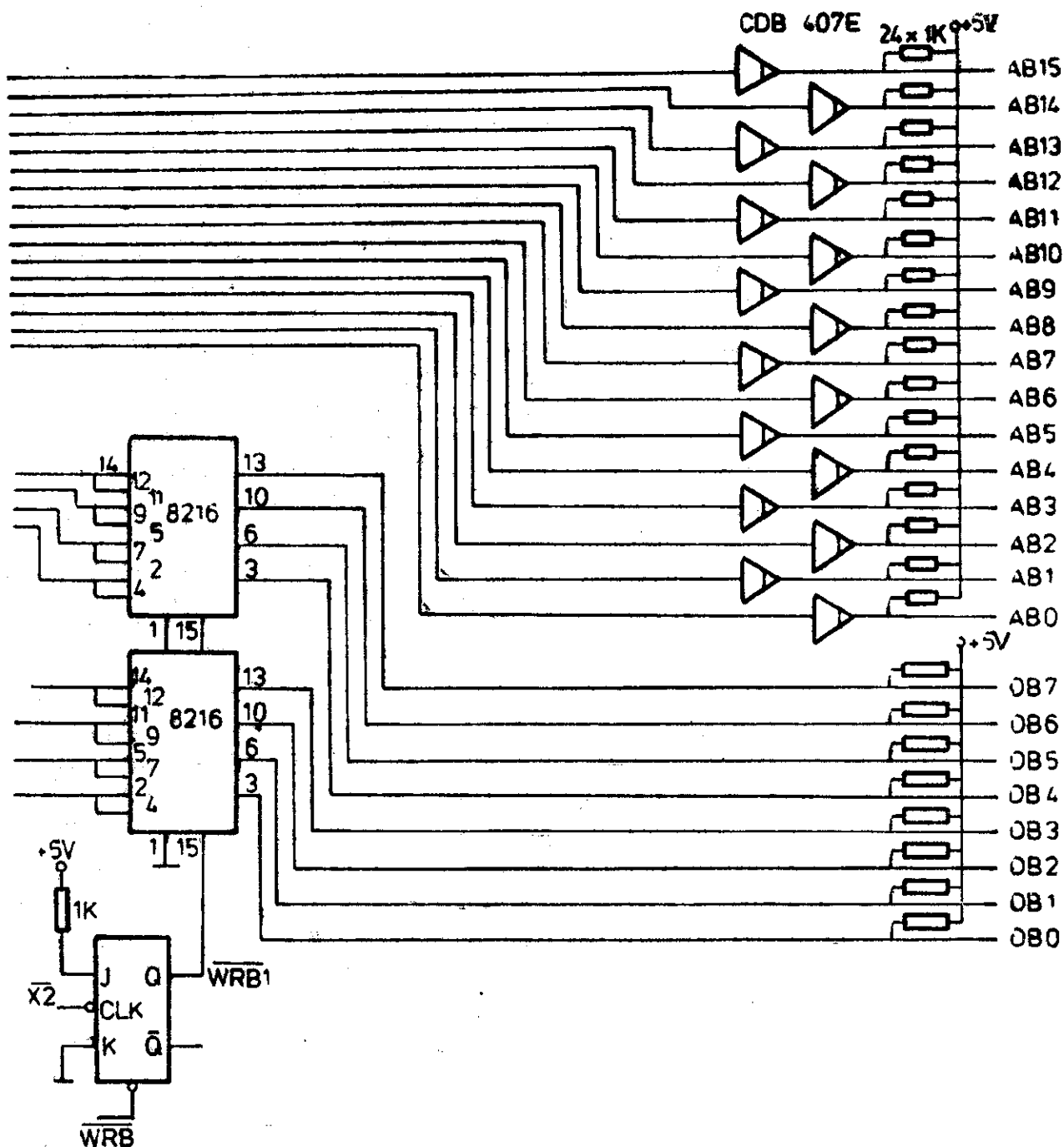


Fig. 3.4. Unitatea centrală de prelucrare.

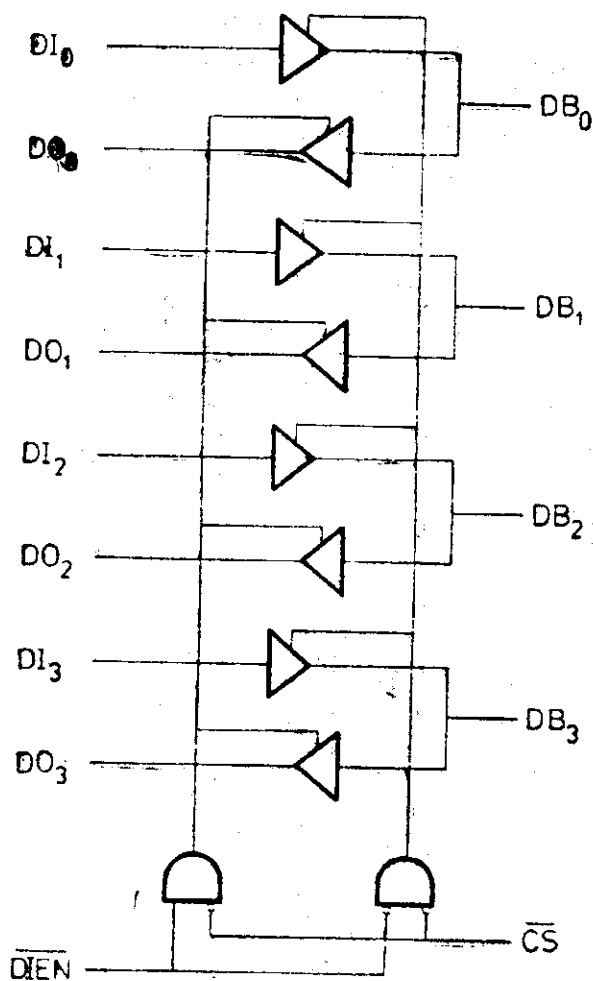


Fig. 3.5. Circuitul 8216.

Pentru a stabili sensul de transfer prin circuitul 8216 există două intrări de comandă. Intrarea \overline{CS} selectează circuitul: cît timp se află la nivel logic ridicat ieșirile tuturor bufferelor se află în stare de mare impedanță. Cînd \overline{CS} se află pe 0 logic, circuitul este selectat și sensul este determinat de intrarea \overline{DIEN} . Dacă \overline{DIEN} este pe 0 logic, sensul este de la DI la DB, iar dacă este pe 1 logic, sensul este de la DB la DO. În schema din figura 3.5 cele două circuite au intrările CS legate la masă (sînt selectate în permanență) iar pe intrările \overline{DIEN} se furnizează semnalul $\overline{WRB1}$, care devine 0 logic pentru scriere, din sensul DI la DB și 1 logic pentru citire, deci sensul DB la DO.

3.3. Memoria RAM

Memoria RAM a sistemului este construită cu circuite dinamice 4116 și are capacitatea minimă de 16 Ko. Prin implantarea de circuite chiar pe placa unității centrale, capacitatea se poate extinde la 48 Ko.

Circuitul 4116 este un circuit de memorie dinamică cu acces aleator cu capacitatea de 16 Kbiți, formatul 16384×1 , realizat în tehnologia MOS canal

și a semnalelor de comandă \overline{MREQ} , cerere de acces la memorie, \overline{IORQ} , cerere de intrare/ieșire, \overline{RD} , citire din memorie sau port de intrare și \overline{WR} , scriere în memorie sau port de ieșire, s-a făcut cu porți neinversoare cu colector în gol CDB 407E.

Pentru interfațarea liniilor bidirecționale de date ale microprocesorului, D0-D7 s-au utilizat două circuite 8216.

Schema logică a acestui circuit este prezentată în figura 3.5. Fiecare linie bidirecțională constă din două buffere cu 3 stări, la care ieșirea unuia și intrarea celuilalt sînt conectate împreună (DB). Cele patru linii DB0-DB3 sînt utilizate pentru a interfața diferite componente, cum sînt memorii, echipamente de intrare/ieșire. Celelalte intrări și ieșiri ale bufferelor din circuitul 8216 sînt lăsate libere, constituind liniile DI0-DI3 și DO0-DO3, aceasta pentru a conferi maximum de flexibilitate. Pentru a interfața magistrala de date a microprocesorului, aceste linii au fost însă conectate împreună și legate la pini de date ai lui Z80.

N, destinat utilizării în sisteme cu cerințe mari de memorie, viteză sporită, putere disipată mică și cost scăzut. Caracteristicile principale sînt :

- capsulă standard cu 16 pini ;
- tensiuni de alimentare : +5V, -5V, +12V și masă ;
- timp de acces 150ns/200ns/250ns, în funcție de tipul circuitului 4116-2/3/4 ;
- ciclul memoriei 320ns/375ns/410ns pentru 4116-2/3/4 ;
- consum scăzut de energie 462 mW (activ)/200 mW (inactiv) ;
- 128 cicluri de refresh la interval de 2 ms.

Pentru a adresa 16384 de locații de memorie sînt necesari 14 biți de adresă, care se multiplexează în raport 2 : 1. Astfel pentru referirea la o celulă din circuitul de memorie se trimite adresa de rînd (7 biți) cu activarea semnalului de strob a adresei de rînd, $\overline{\text{RAS}}$, apoi se trimite adresa de coloană (7 biți) cu activarea semnalului de strob a adresei de coloană, $\overline{\text{CAS}}$. Conexiunile externe ale circuitului 4116 sînt prezentate în figura 3.6.

În figura 3.7 se prezintă schema memoriei construită cu circuite 4116. Fiecare modul este format din 8 cipuri, realizîndu-se în acest fel o capacitate de 16 Ko. Intrările de adrese pentru toate cipurile sînt legate împreună la liniile de adrese AM0-AM6 . De asemenea semnalele de strob pentru adresa de rînd, $\overline{\text{RAS}}$ și de scriere $\overline{\text{W}}$ sînt comune la toate circuitele de memorie. Diferă numai semnalele de strob pentru adresa de coloană. Astfel la modulul 0, acest semnal este $\overline{\text{CAS0}}$, la modulul 1, $\overline{\text{CAS1}}$, iar la modulul 2, $\overline{\text{CAS2}}$.

Datele de ieșire ale memoriei RAM, DO0-DO7 , se încarcă într-un registru, construit cu două circuite CDB 495E. Circuitul CDB 495E poate funcționa în două moduri : deplasare și încărcare paralelă, prin utilizarea a două intrări de tact $\overline{\text{CP1}}$ și $\overline{\text{CP2}}$. Selecția modului de funcționare se face prin intrarea S : un nivel logic ridicat activează intrarea $\overline{\text{CP2}}$ (încărcare paralelă), iar un nivel logic coborît activează intrarea $\overline{\text{CP1}}$ (deplasare). Registrul de date de ieșire al memoriei RAM, funcționează numai în regim de încărcare paralelă, pentru aceasta intrarea de selecție a modului de lucru este conectat prin intermediul unei rezistențe de 1K Ω la +5 V. Încărcarea datelor de ieșire DO0-DO7 se face utilizînd semnalul X3, furnizat de sincrogenerator. În continuare, aceste date sînt preluate pe frontul negativ al semnalului STB într-un al doilea nivel de registre CDB 495E, care funcționează de asemenea numai în regim de încărcare paralelă. De aici datele ajung pe magistrala de date a sistemului, DB0-DB7 , printr-un tampon construit cu porți SI-NU cu eolctor în gol CDB 403E, care este activat dacă există cerere de acces la memorie ($\text{MREQ}=1$) și accesul este pentru citire ($\text{RDB}=1$).

Intrările de date ale-circuitelor de memorie sînt conectate la magistrala de date a sistemului DB0-DB7 , printr-un nivel de inversoare CDB 404E. Acest

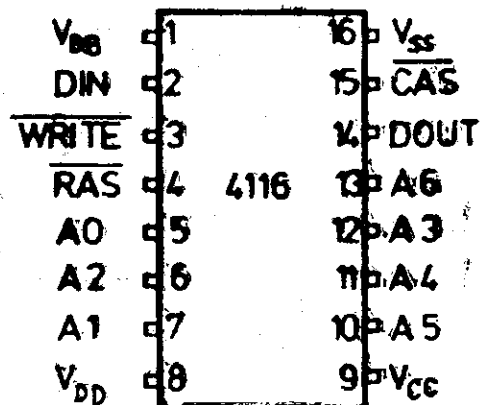


Fig. 3.6. Conexiunile externe ale circuitului 4116

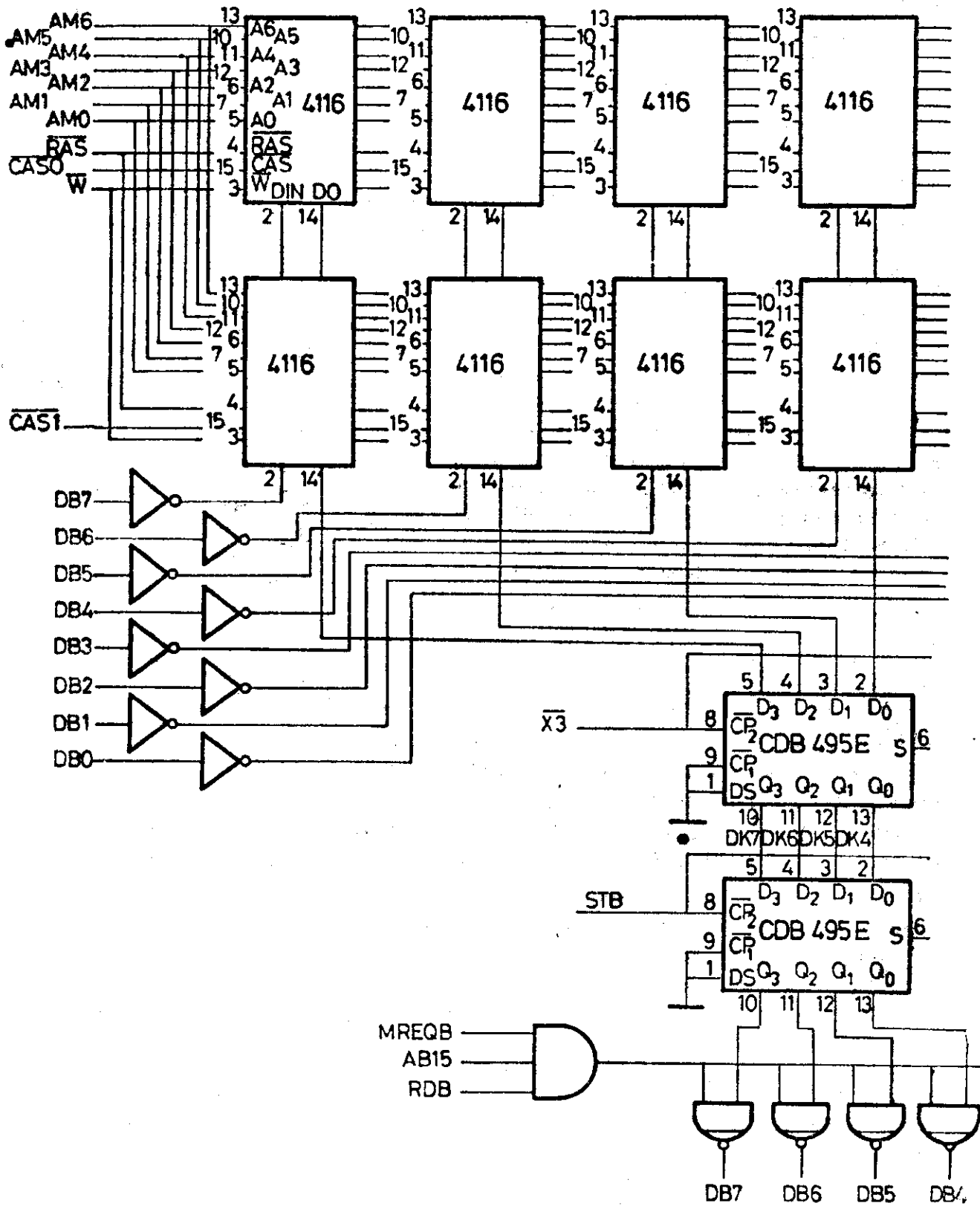
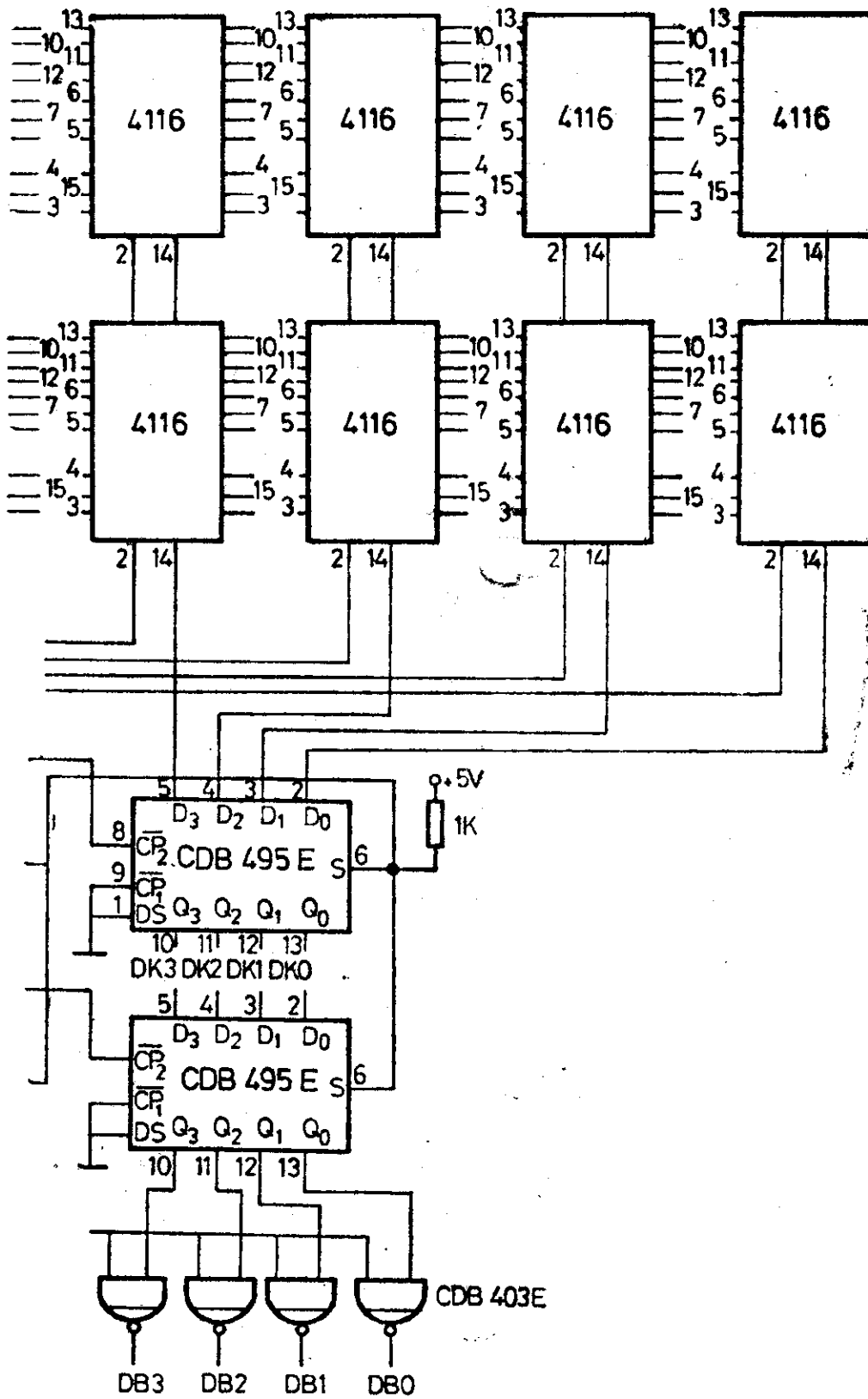


Fig. 3.7. Memoria RAM



(modulele 0 și 1 : 32 Kocteți).

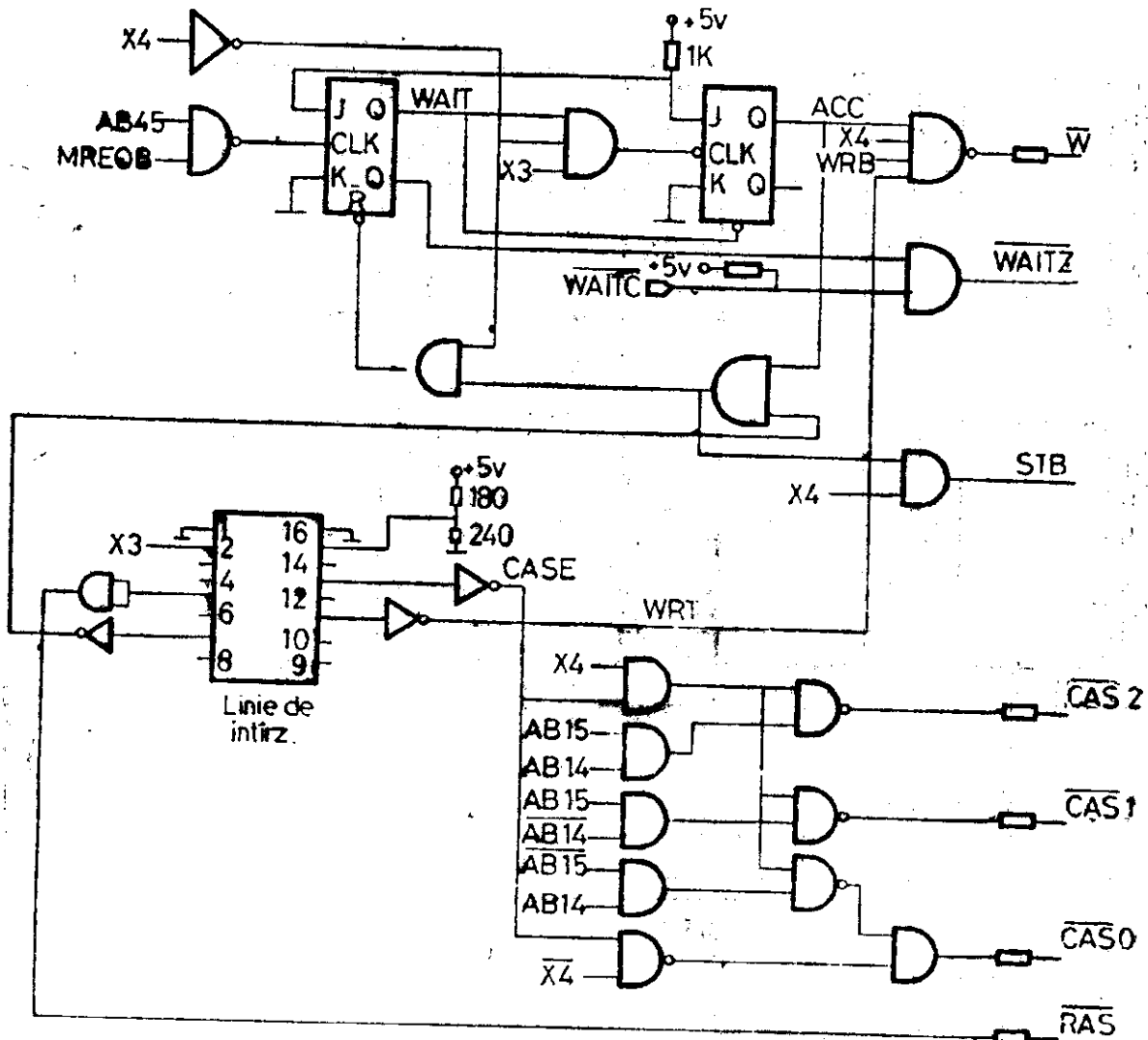


Fig. 3.8. Logica de comandă a memoriei RAM.

lucru este necesar deoarece la citire datele ajung din memorie pe magistrală, negate, datorită tampoanelor CDB 403E, deci la înscrisere se vor complementa.

Logica de comandă a memoriei RAM este prezentată în figura 3.8. Ciclul de memorie cuprinde două accese: un acces pentru citirea datelor din zona ecran, între adresele 4000H-5FFFH, în vederea afișării la televizor și un acces la oricare locație din RAM, efectuat de microprocesor pentru citirea/înscrisura datelor din/în memorie. Deoarece rezoluția ecranului este de 256×256 puncte, distanța în timp între două puncte succesive de pe o aceeași linie TV este de 200 ns. Din memoria RAM se citesc 8 biți simultan, deci 8 puncte succesive ceea ce permite ca un ciclu să dureze 1,6 μ s. Diagrama în timp a semnalelor furnizate de logica de comandă a memoriei este prezentată în figura 3.9.

Ciclul este definit de semnalul X4, cu perioada de 1,6 μ s (paragraful 3.6). Timp de 800 ns, cît semnalul X4 este pe nivel logic coborît, se face acces la memorie pentru citire în vederea afișării la televizor, iar în următoarele 800 ns, cît semnalul X4 este pe nivel logic ridicat, se face acces din partea

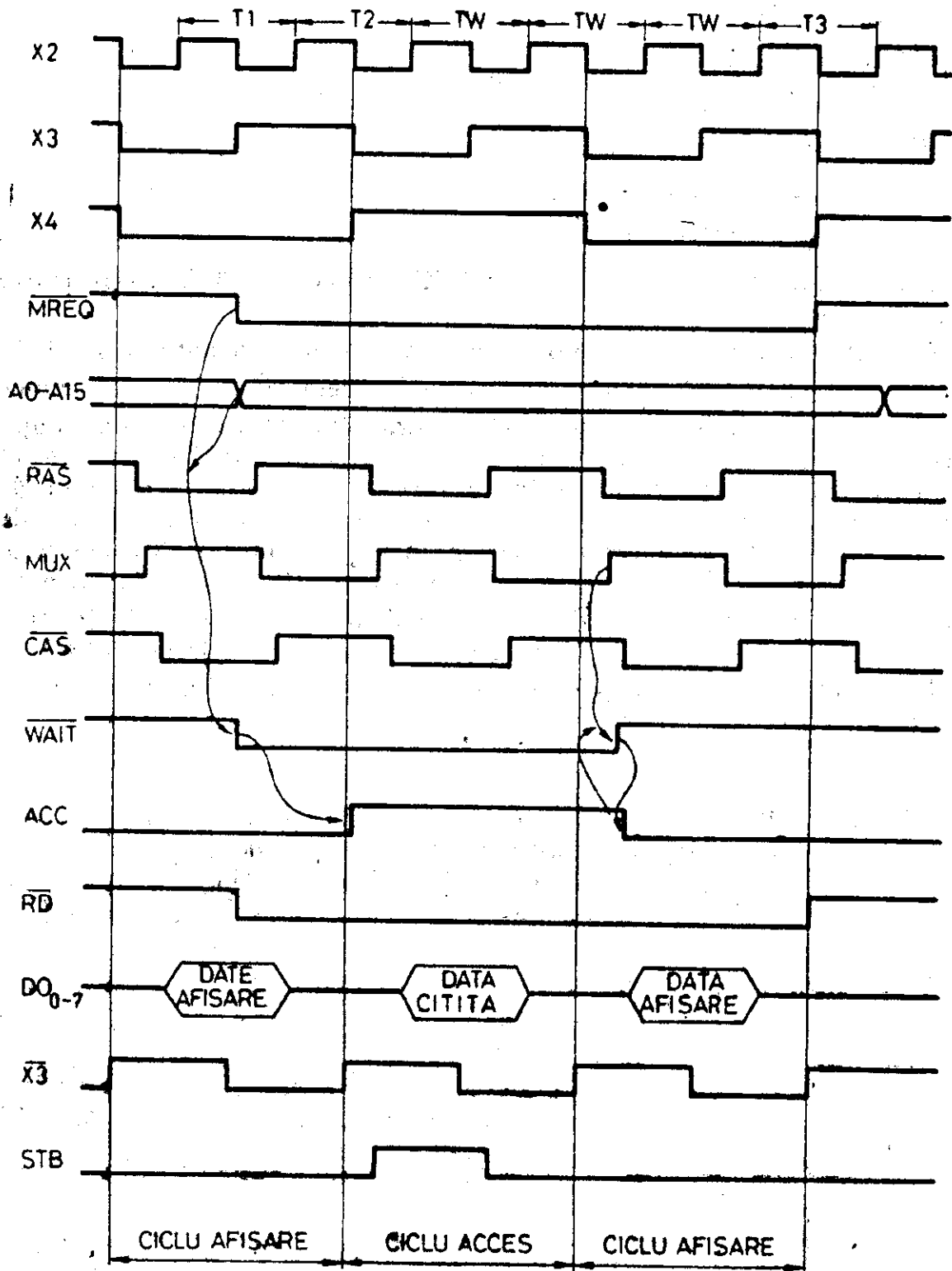


Fig. 3.9. Diagrama in timp a semnalelor furnizate de logica de comandă a memoriei RAM.

microprocesorului. Ecuțiile logice furnizate de schema de comandă a modului RAM sînt următoarele :

$$\overline{\text{RAS}} = \text{X3D}$$

$$\text{CASE} = \overline{\text{X3DD}}$$

$$\overline{\text{CAS0}} = \overline{\text{CASE} \cdot (\text{X4} + \text{X4} \cdot \text{AB14} \cdot \overline{\text{AB15}})}$$

$$\overline{\text{CAS1}} = \overline{\text{CASE} \cdot \text{X4} \cdot \overline{\text{AB14}} \cdot \text{AB15}}$$

$$\overline{\text{CAS2}} = \overline{\text{CASE} \cdot \text{X4} \cdot \text{AB14} \cdot \overline{\text{AB15}}}$$

$$\overline{\text{W}} = \overline{\text{ACC} \cdot \text{CASE} \cdot \text{X4} \cdot \overline{\text{WRB}}}$$

unde X3D și X3DD reprezintă semnalul X3 respectiv $\overline{\text{X3}}$ întîrziat printr-o linie de întîrziere.

Accesul la o locație din RAM se face prin trimiterea adresei de rînd cu semnalul de strob RAS la toate cipurile, apoi adresa de coloană cu semnalul de strob CAS, numai la modulul selectat. Semnalul RAS este activ timp de 400 ns, la fiecare semi-ciclu al memoriei, iar semnalul CASE se activează cu aproximativ 100 ns mai tîrziu. Strobarea adresei de coloană la modulul de memorie se face numai dac̃ă semiciclu curent este de afișare (X4=0) sau dac̃ă semiciclu curent este de acces din partea microprocesorului (X4=1) și biții de adresă sînt AB14=1 și AB15=0 (referire la zona 4000H-7FFFH). Semnalul CAS1 se activează dac̃ă semiciclu curent este de acces (X4=1) și biții cei mai semnificativi sînt AB14=0 și AB15=1 (referire la zona 8000H-BFFFH).

Încărcarea registrului de date de ieșire se face pe frontul negativ al semnalului LOAD, cu ecuația logică

$$\text{LOAD} = \overline{\text{X3}}$$

În figura 3.10 este prezentat blocul de multiplexare a adreselor pentru memorie. Memoria RAM poate fi adresată fie de logica de afișare la televizor, fie de microprocesor. Deci există două seturi de adrese de cîte 14 biți fiecare :

– X5-X9, Y0-Y7 adresa furnizată de sincrogenerator, valabilă în timpul accesului pentru afișare (X4=0). Bitul cel mai semnificativ de adresă este legat la masă, căci zona ecran se află între adresele 4000H-5FFFH ;

– AB0-AB13 adresa furnizată de pe magistrala de adresă a sistemului și care este valabilă în timpul accesului microprocesorului.

Multiplexorul este implementat cu circuite CDB 4153E. Se realizează o multiplexare în raport de 4 : 1, obținîndu-se 7 linii de adresă AM0-AM6, care merg direct la intrările de adresă ale circuitelor de memorie.

Selecția este realizată cu semnalele MUX (X3 întîrziat) și X4.

În implementarea acestor ecuații s-a ținut seama de faptul că la activarea semnalelor de strob pentru adresa de rînd RAS și de coloană CAS adresele corespunzătoare trebuie să fie deja stabile pe magistrala de adrese a memoriei RAM, AM0-AM6.

Schema de comandă a memoriei conține o logică de arbitrare a conflictului între accesul din partea microprocesorului și afișare. În acest scop se

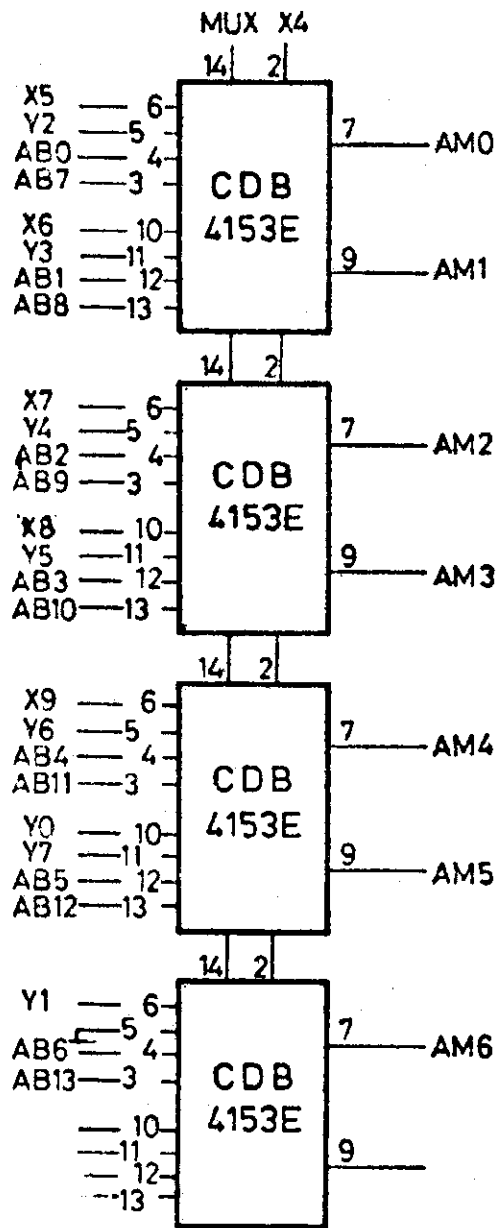


Fig. 3.10. Blocul de multiplexare adrese.

utilizează o schemă secvențială compusă din bistabilii WAIT și ACC, împreună cu câteva porți aferente. Un eventual conflict se poate datora faptului că cererea de acces la memorie a microprocesorului \overline{MREQ} poate să apară în orice moment de timp, inclusiv în timpul semiciclului de afișare. De aceea, dacă se face acces la memoria RAM în zona 4000H-FFFFH, activarea semnalului \overline{MREQ} poziționează bistabilul WAIT în 1 logic. Ecuațiile de excitație pentru acest bistabil sînt :

$$J/WAIT = 1$$

$$K/WAIT = 0$$

$$CLK/WAIT = \overline{MREQ(AB15 + AB14)}$$

$$R/WAIT = \overline{ACC.MUX.X4}$$

În acest fel microprocesorul este trecut în starea de așteptare pînă ce accesul la RAM este permis. Dacă bistabilul WAIT este poziționat în 1 logic, la sfîrșitul semiciclului de afișare se poziționează bistabilul ACC în 1, semnificînd că în semiciclul următor, se va face un acces pentru citire sau scriere. Ecuațiile de excitație pentru acest bistabil sînt :

$$J/ACC = 1$$

$$K/ACC = \bar{0}$$

$$CLK/ACC = WAIT \cdot \bar{X4} \cdot X3$$

$$R/ACC = WAIT$$

După efectuarea operației de citire sau scriere, la sfîrșitul semiciclului de acces, bistabilul WAIT este șters, trecerea în 0 a acestuia efectuînd și resetarea bistabilului ACC. În acest fel micșorarea vitezei de lucru a microprocesorului, prin trecerea sa în WAIT la accesele la memoria RAM, este neglijabilă.

Reîmprospătarea informației în circuitele de memorie dinamică se face automat prin citirea pentru afișare. Pentru executarea unui refresh este suficient să se furnizeze numai adresa de rînd, cu activarea semnalului \bar{RAS} , în acest fel se reîmprospătează informația de pe întregul rînd selectat. Memoria ecran este organizată în așa fel încît informația corespunzătoare la grupuri succesive de 8 puncte de pe aceeași linie TV se află pe rînduri succesive din modulul 0. Prin afișarea unei linii TV se face adresarea la 32 de rînduri succesive din ambele module de RAM, căci se furnizează adresa de rînd și se activează semnalul \bar{RAS} pentru întreaga memorie. Deci baleierea întregii memorii se face într-un interval de timp dat de formula :

$$T = \frac{n_1}{n_2} \cdot t = \frac{128}{32} \cdot 0,064 = 0,256 \text{ ms} < 2 \text{ ms}$$

unde n_1 este numărul de rînduri în cipul 4116 ;

n_2 = numărul de rînduri baleiate la afișarea unei linii TV ;

t = durata totală a unei linii TV (în ms).

Reîmprospătarea informației se face corect, perioada fiind mai mică decît perioada maximă de reîmprospătare de 2 ms, prevăzută în catalog.

3.4. Memoria EPROM

Modulul de memorie EPROM constituie suportul fizic al sistemului de operare. Este construit cu circuite 2716, de 2 Ko, formatul 2048 × 8, avînd capacitatea maximă 16 Ko.

În funcție de sistemul de operare rezident, există mai multe variante dimensionale ale modulului EPROM :

- Monitor (2 Ko) și interpretor BASIC simplu (8 Ko) ;
- Monitor (2 Ko) și interpretor BASIC extins (14 Ko) ;
- MATE-Monitor, Asamblor, Editor de Texte (6 Ko).

Deci pentru prima variantă sînt necesare 5 cipuri (10 Ko) în varianta a doua (variantă maximă) 8 cipuri (16 Ko), iar în varianta a treia 3 cipuri (6 Ko).

Modulul EPROM începe de la adresa 0000H și se întinde în varianta maximă pînă la 3FFFH.

În figura 3.11 se prezintă configurația pinilor pentru circuitele INTEL 2716 (a) și TMS 2716 (b). Caracteristicile principale sînt :

- capsulă standard cu 24 de pini ;
- capacitatea de memorie 2048×8 biți ;
- timp de acces 450ns ;
- puterea maximă disipată 500mW ;
- intrările și ieșirile compatibile TTL ;
- ieșirile sînt 3- state ;
- tensiuni de alimentare : $\pm 5V$ și masă pentru INTEL 2716, respectiv $\pm 5V$, $\pm 12V$, $-5V$ și masă pentru TMS 2716.

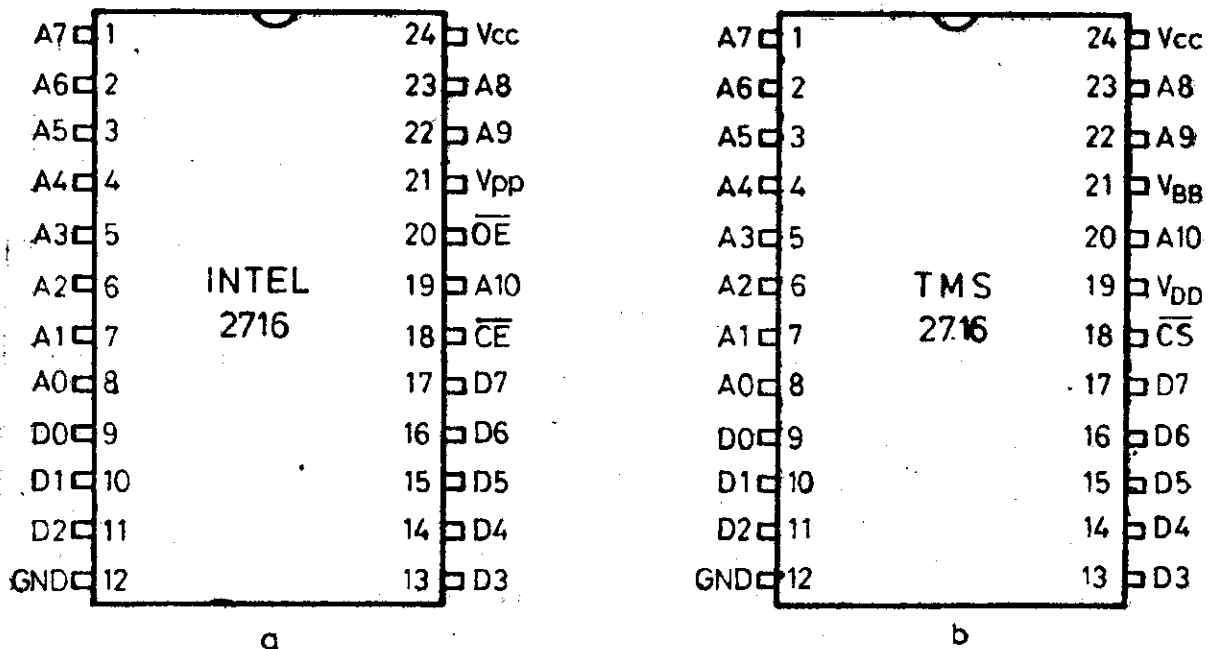


Fig. 3.11. Configurația pinilor la circuitele INTEL 2716 (a) și TMS 2716 (b).

Schema modului de memorie EPROM este prezentată în figura 3.12. Decodificatorul CDB 442E realizează selecția circuitului adresat, decodificînd biții de adresă AB15-AB11 de pe magistrală. Ieșirea circuitului selectat este activată cu semnalul MREQB.RDB.

3.5. Interfața cu tastatura

Interfața periferică programabilă, realizată cu circuitul 8255 (1), îndeplinește următoarele funcții în cadrul sistemului :

- interfațează tastatura ;
- interfațează casetofonul audio ;

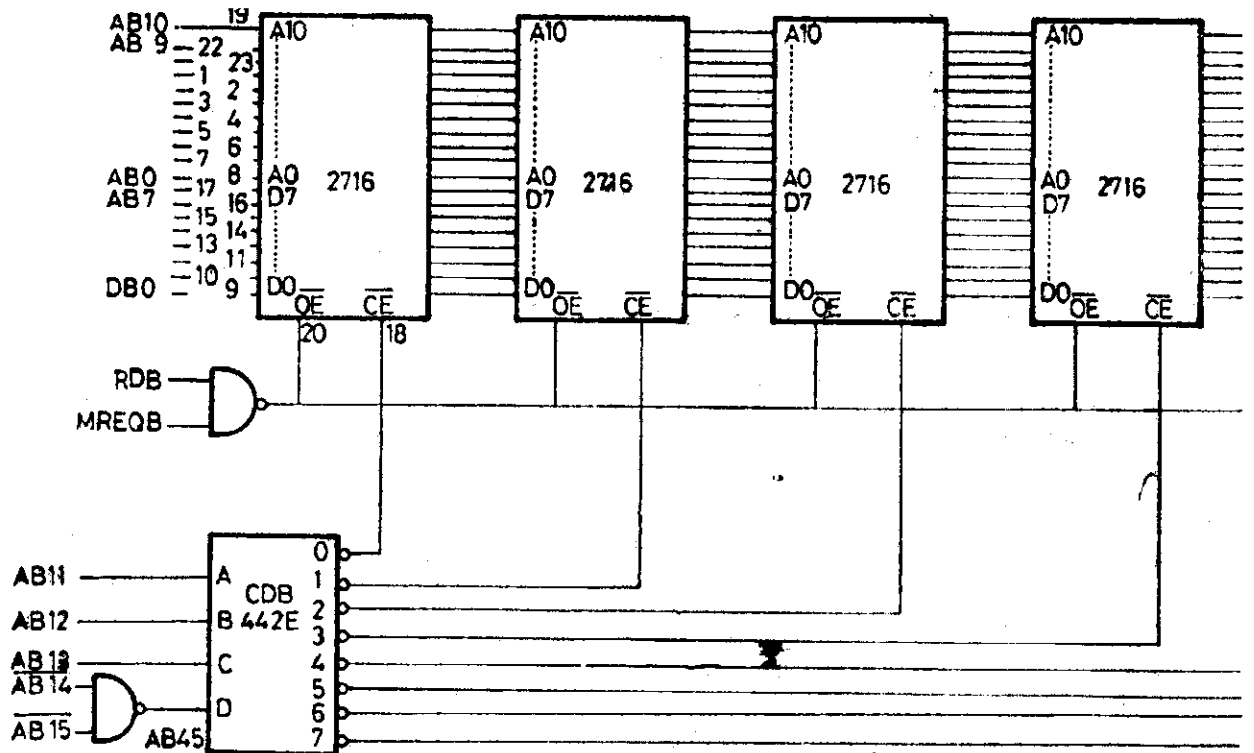


Fig. 3.12. Schema

- furnizează semnal pentru video invers ;
- generează semnal pentru un amplificator audio și difuzor în vederea unor aplicații acustice.

Circuitul 8255 este un dispozitiv de I/E programabil, de uz general, având 24 de pini de intrare/ieșire care se pot programa individual în două grupuri de câte 12 pini și se pot utiliza în 3 moduri generale de operare. În figura 3.13 se prezintă configurația pinilor (a) și schema bloc internă a circuitului (b).

\overline{CS} (Chip Select), activ pe nivel logic coborât, permite comunicația între circuitul 8255 și microprocesor.

\overline{RD} (Read) permite transmiterea de date sau informații de stare de la 8255 către microprocesor.

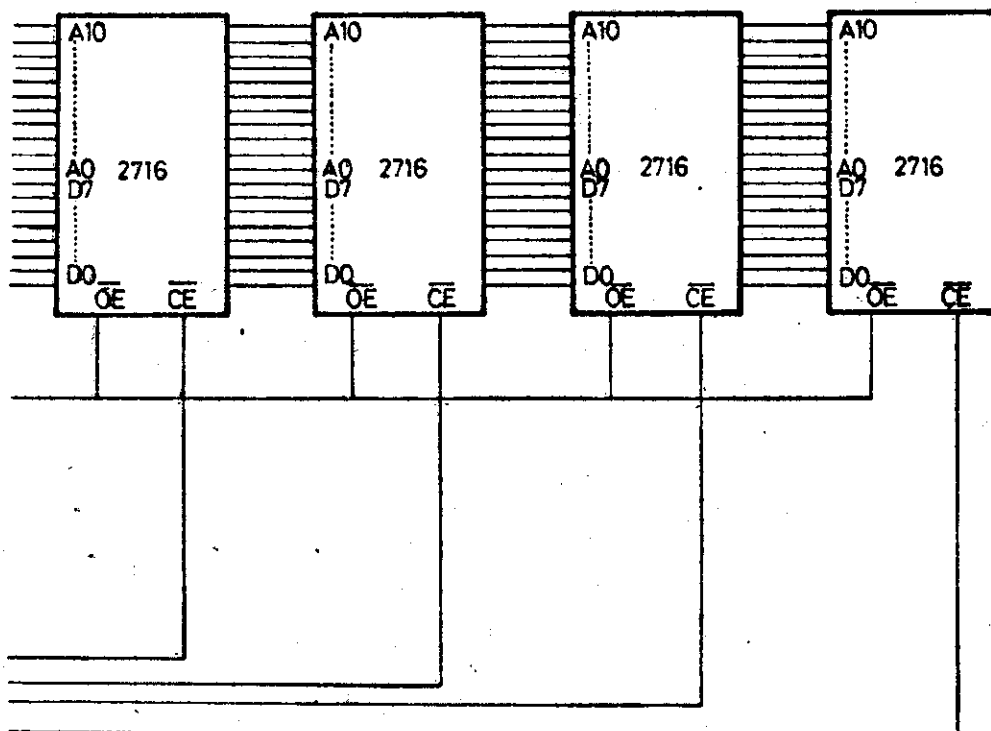
\overline{WR} (Write) este semnalul de înscriere în 8255 a unor cuvinte de control sau date.

A_0, A_1 , împreună cu semnalele \overline{RD} și \overline{WR} , selectează unul din cele 3 porturi de intrare/ieșire sau registrul cuvântului de control. În mod normal aceste intrări se conectează la magistrala de adrese, la biții cei mai puțin semnificativi. În figura 3.14 se prezintă operațiile de bază executate de circuitul 8255.

RESET, activ pe nivel logic ridicat, șterge toate registrele interne, inclusiv registrul cuvântului de control, iar toate porturile (A, B și C) sînt trecute în modul intrare.

D7-D0 se conectează la magistrala bidirecțională de date și permite transferul datelor, stărilor și cuvântului de control.

PA7-PA0, PB7-PB0 și PC7-PC0 reprezintă cele 3 porturi de intrare/ieșire care se pot programa de către utilizator.



modulului EPROM.

În schema internă din figura 3.13 (b) sînt reprezentate următoarele blocuri :

- logica pentru controlul scrierii/citirii ;
- bufferul de date ;
- blocurile de control pentru grupul A și grupul B ;
- grupul A constituit din portul A și jumătatea mai semnificativă a portului C ;
- grupul B constituit din portul B și jumătatea mai puțin semnificativă a portului C.

Logica pentru controlul scrierii/citirii are rolul de a gestiona toate transferurile interne sau externe de date, comenzi sau stări. Acest bloc acceptă semnale de pe magistrala sistemului și furnizează comenzi pentru ambele blocuri de control de grup.

Bufferul de date, bidirecțional, cu 3 stări, interfațează circuitul 8255 la magistrala de date. Datele, cuvintele de control și informațiile de stare sînt transmise sau recepționate de către buffer prin executarea unor instrucțiuni IN sau OUT.

Configurația funcțională a fiecărui port este programată prin software. Cuvîntul de control transmis de microprocesor la 8255, conține informații care inițializează configurația circuitului. Fiecare din blocurile de control pentru grupul A și grupul B acceptă comenzi de la logica de control a scrierii/citirii, prin magistrala internă de date și emite comenzi proprii către porturile asociate

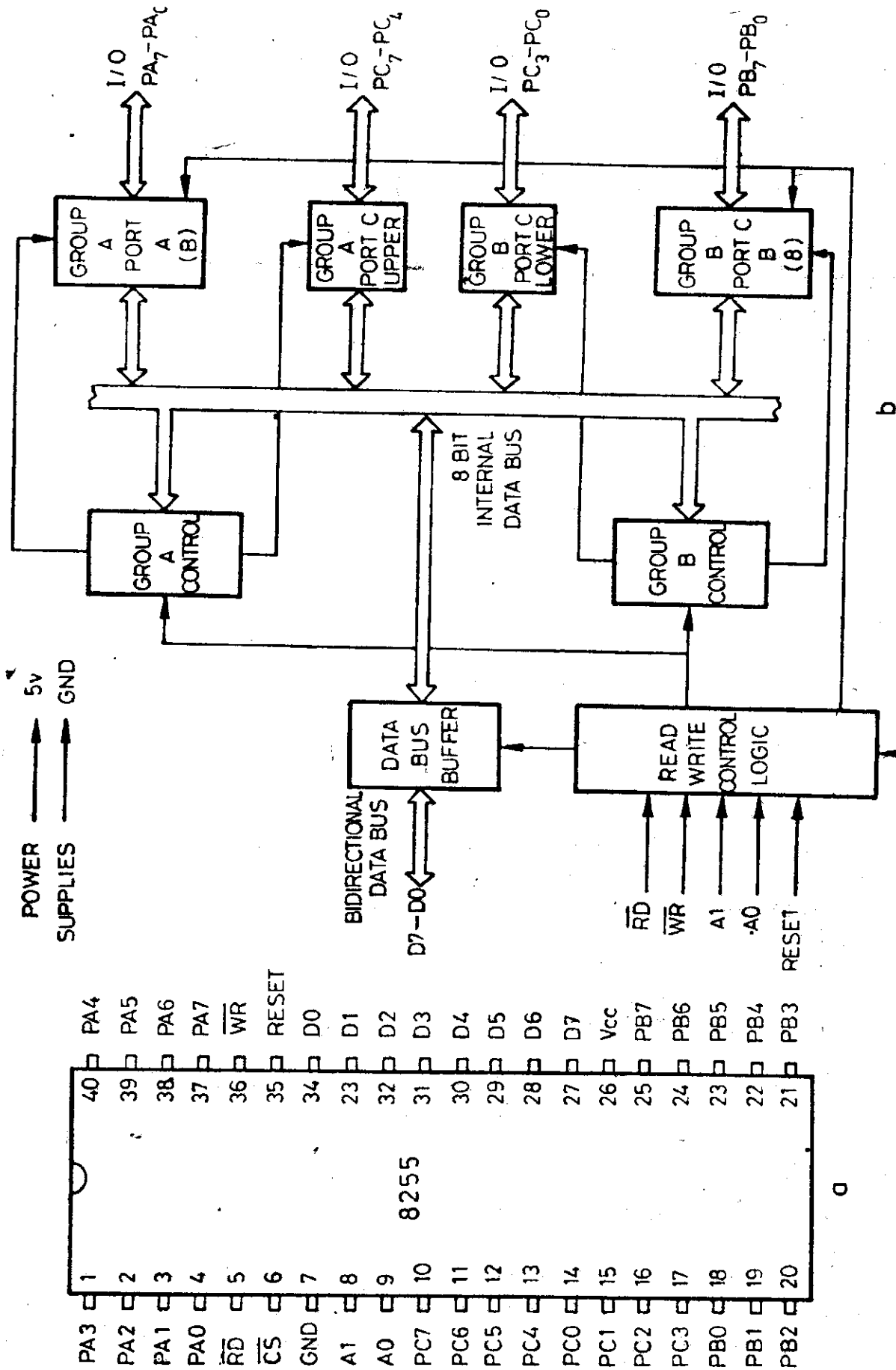


Fig. 3.13. Configurația pinilor (a) și schema bloc internă a circuitului 8255 (b).

A1	A0	\overline{RD}	\overline{WR}	\overline{CS}	OPERAȚII DE INTRARE
0	0	0	1	0	PORT A → MAG. DATE
0	1	0	1	0	PORT B → MAG. DATE
1	0	0	1	0	PORT C → MAG. DATE
					OPERAȚII DE IEȘIRE
0	0	1	0	0	MAG. DATE → PORT A
0	1	1	0	0	MAG. DATE → PORT B
1	0	1	0	0	MAG. DATE → PORT C
1	1	1	0	0	MAG. DATE → PORT CONTROL
					DEZACTIVARE
X	X	X	X	1	MAG. DATE → 3 - STATE
1	1	0	1	0	ILEGAL
X	X	1	1	0	MAG. DATE → 3 - STATE

Fig. 3.14. Operațiile de bază ale circuitului 8255.

Există trei moduri de operare de bază

- modul 0 : intrare/ieșire de bază ;
- modul 1 : intrare/ieșire strobată ;
- modul 2 : magistrală bidirecțională.

Se pot defini separat modurile de lucru pentru portul A și portul B, însă portul C este divizat în două, fiecare din cele două jumătăți funcționând în modul portului de care aparține (A sau B)

In modul 0 (intrare/ieșire de bază) fiecare din cele 3 porturi funcționează pentru intrare sau pentru ieșire, datele fiind citite din, sau înscrise în oricare din porturi.

Modul 1 (intrare/ieșire strobată) permite transferul de date cu un port specificat în conjuncție cu semnale de strob sau de protocol. Porturile A și B utilizează liniile portului C pentru a genera sau accepta aceste semnale.

Modul 2 (magistrală bidirecțională) permite comunicația cu un dispozitiv periferic printr-o magistrală cu 8 linii, în ambele sensuri, recepție/transmisie de date, utilizând portul A. Semnalele de protocol sînt furnizate pe 5 linii ale portului C.

În cadrul microcalculatorului personal, circuitul 8255 este programat din monitor în modul 0 de lucru, cuvîntul de control transmis fiind 92H. În acest fel portul C este programat pentru ieșire iar porturile A și B pentru intrare. Adresele pentru aceste porturi sînt :

- 20H — port A ;
- 21H — port B ;
- 22H — port C ;
- 23H — registrul cuvîntului de control.

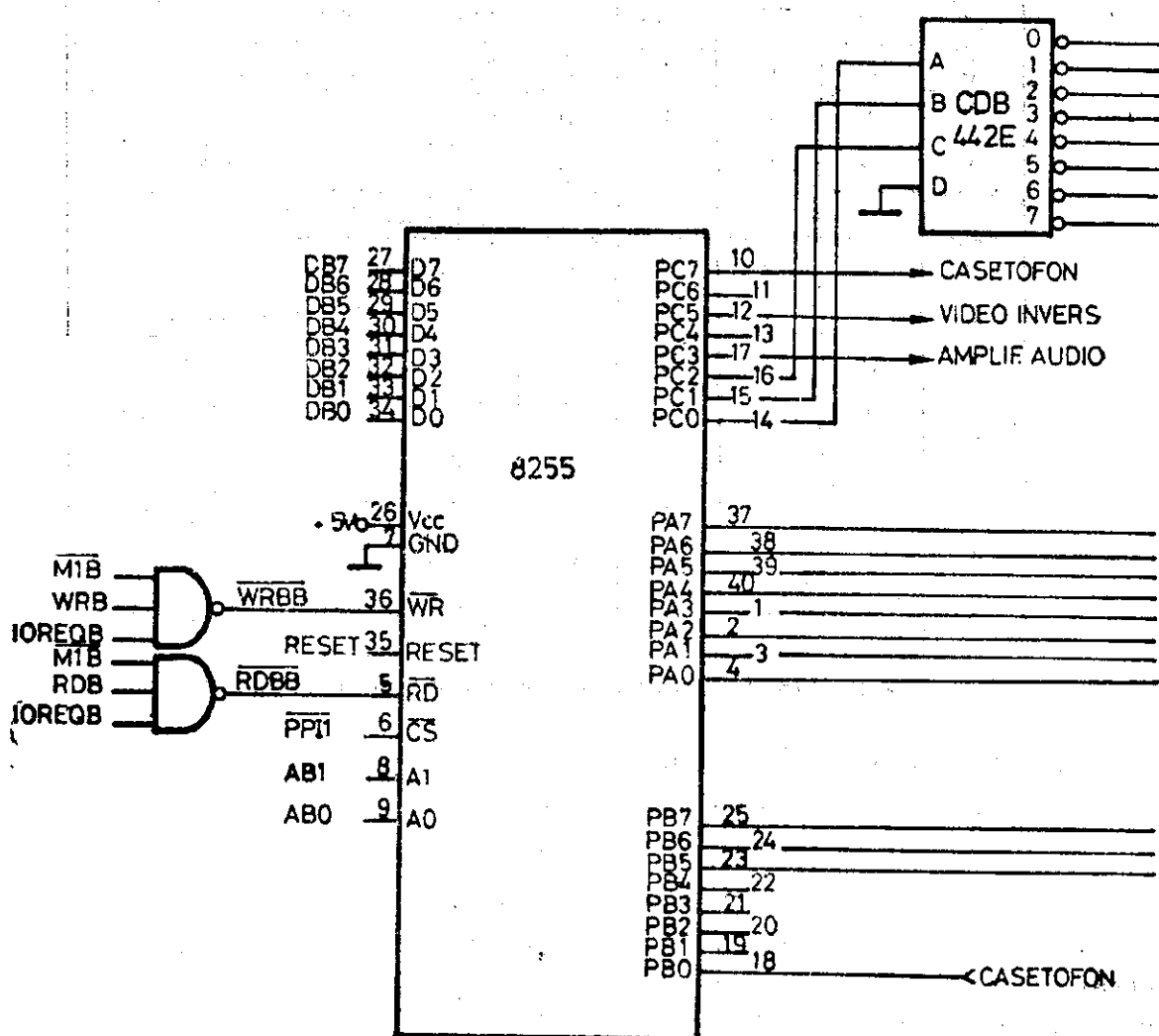
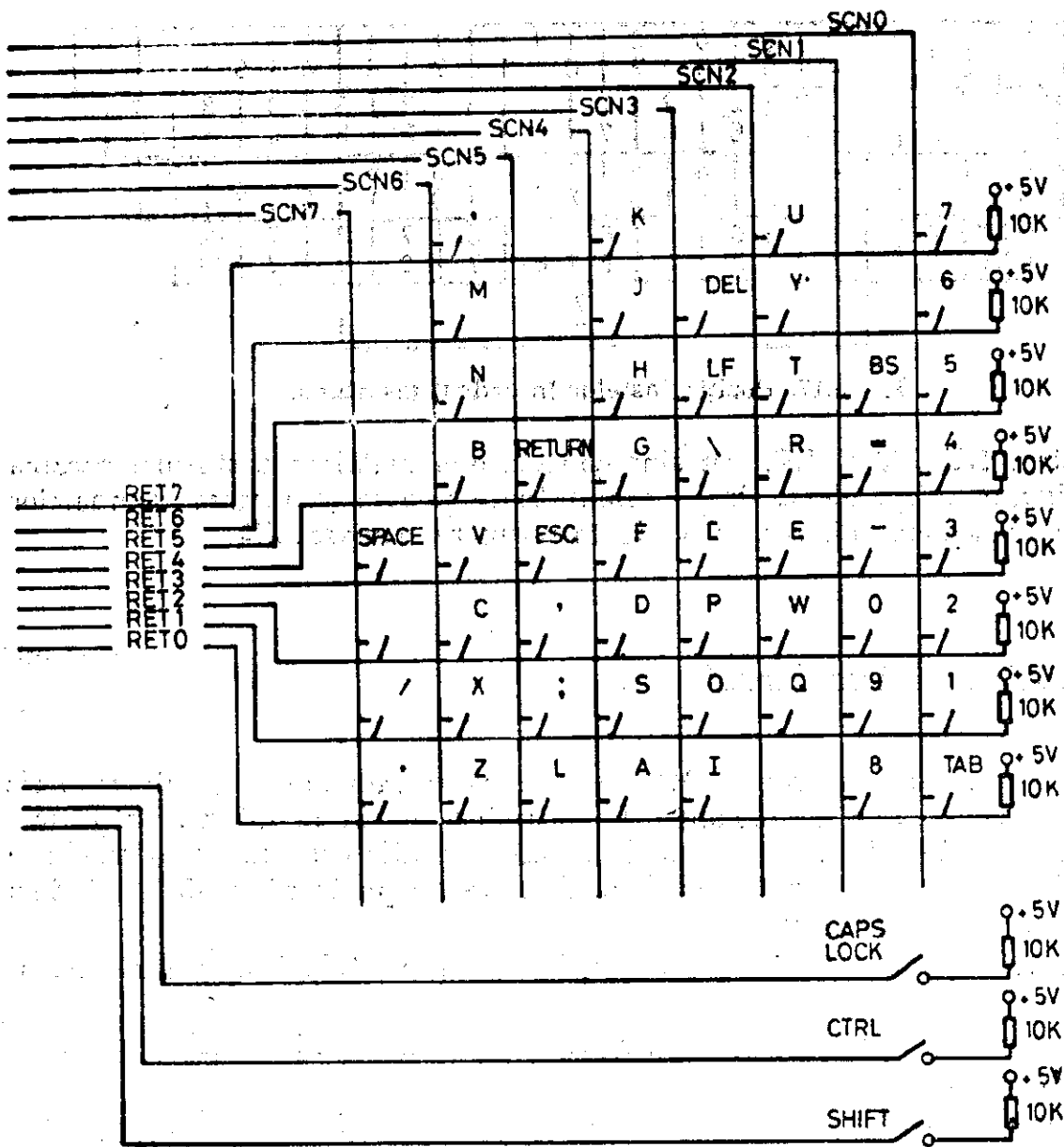


Fig. 3.15. Schema electrică a tastaturii.

Tastatura interfațată la acest sistem este un dispozitiv simplu format dintr-o matrice 8×8 de întrerupătoare, așezate pe 8 linii de scanare și 8 linii de revenire. Schema este prezentată în figura 3.15. Scanarea tastaturii se face prin circuitul 8255, liniile PC2—PC0, cei trei biți fiind decodificați la 8 printr-un circuit CDB442E. În acest fel, la un moment dat, o singură linie de scanare se află la 0 logic, celelalte fiind la 1 logic. Liniile de revenire se află în mod normal la nivel logic ridicat, dar la apăsarea unei taste, se produce contact electric între linia de scanare și linia de revenire pe care se află tasta. Astfel, linia de revenire corespunzătoare trece la nivel logic coborât. Utilizatorul are posibilitatea să citească cele 8 linii de revenire conectate la portul A al circuitului 8255. Cunoșcând poziția tastei apăsate (codul liniei de scanare a fost transmis în portul C, iar liniile de return au fost citite în portul A) se determină codul ASCII al acesteia, în monitor, prin căutare într-o tabelă de coduri.



Separat, se citesc direct în portul B, liniile PB7, PB6 și PB5 (tastele CAPS LOCK CONTROL și SHIFT).
 Organizarea tastaturii se prezintă în figura 3.16, iar poziția tastelor în cadrul matricii, în figura 3.17.

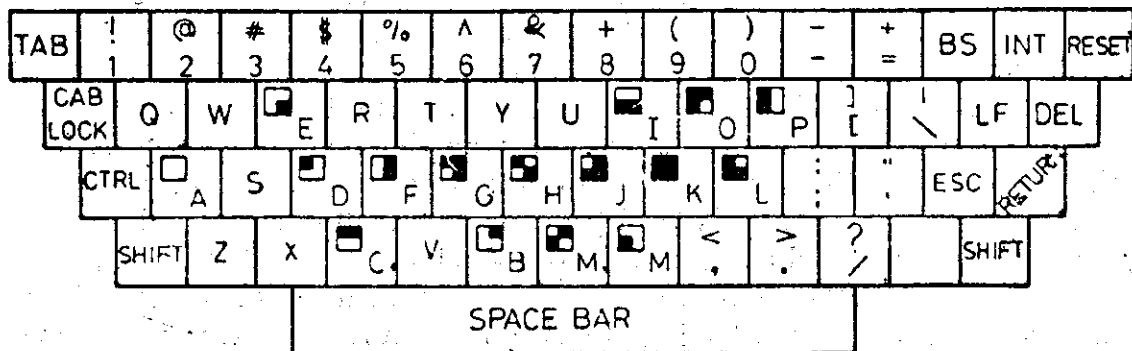


Fig. 3.16. Organizarea tastaturii.

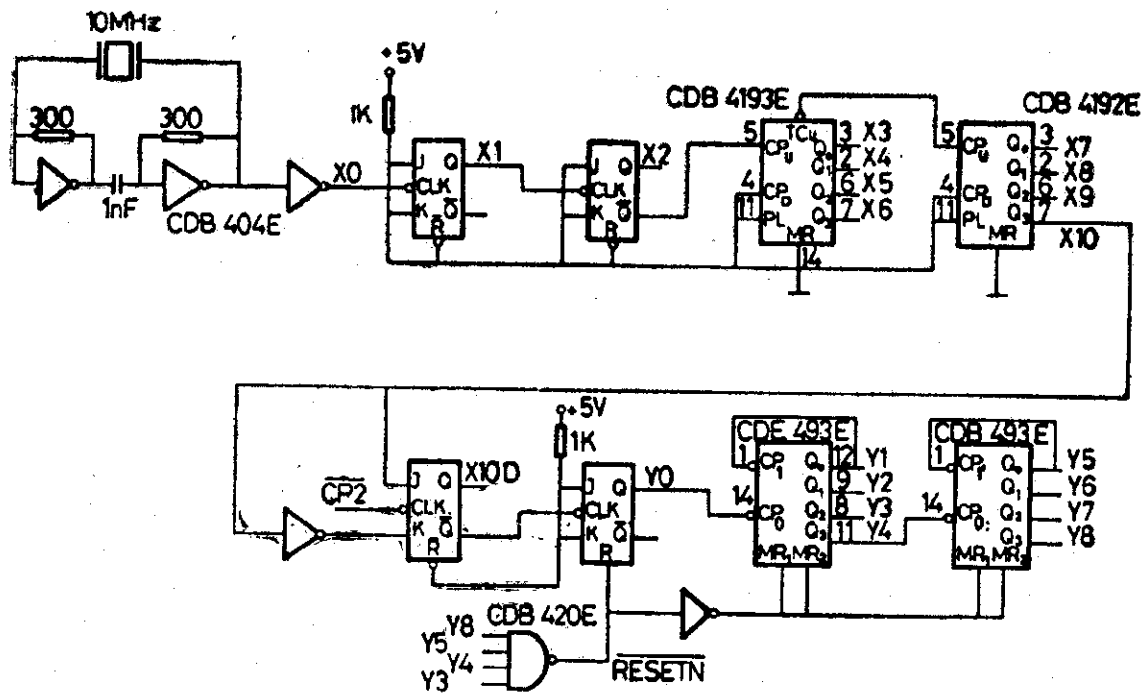


Fig. 3.18. Sincrogeneratorul.

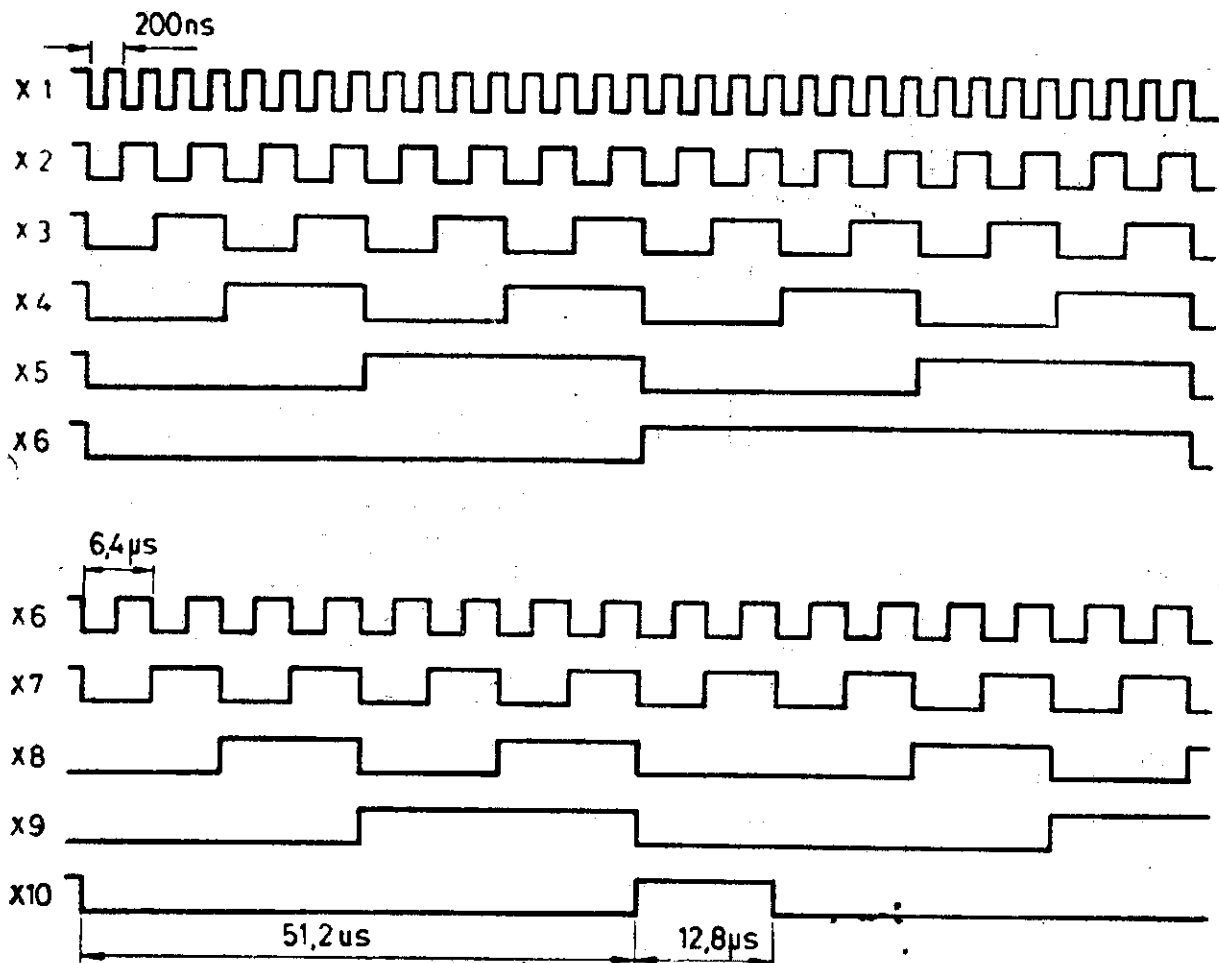


Fig. 3.19. Diagrama de semnale generate de primul bloc de numărătoare.

Al doilea bloc primește ca semnal de tact $\overline{X10D}$, care este de fapt $\overline{X10}$ întârziat cu 800 ns, cu ajutorul unui bistabil JK. Această întârziere este necesară pentru ca la începutul unei linii TV, informația corespunzătoare primelor 8 puncte să fie deja pregătită în registrul de serializare. Al doilea bloc este realizat de asemenea cu circuite CDB 473E și CDB 493E și reprezintă de fapt un numărator modulo 312, care furnizează adresele pe verticală ale punctului curent de pe ecran. Ecuația logică a semnalului de resetare a numărătoarelor este:

$$\overline{\text{RESETN}} = \overline{Y8 \cdot Y5 \cdot Y4 \cdot Y3}$$

În figura 3.20 se prezintă circuitele pentru serializarea informației, generarea semnalelor de sincronizare pe orizontală și verticală și mixarea acestora în semnalul sincrocomplex. Datele citite din memoria ecran în timpul semiciclului de afișare sunt preluate din registrul de date de ieșire și încărcate paralel în alte două circuite CDB 495E pe frontul negativ al semnalului de ceas $\overline{CP2}$, semnalul de mod S fiind pe nivel logic ridicat (încărcare paralelă). Apoi, după trecerea semnalului S pe nivel logic coborât (deplasare) cu ajutorul a 7 impulsuri succesive furnizate pe intrarea de tact $\overline{CP1}$, ale celor două registre, se scot succesiv, la ieșirea Q3 a circuitului corespunzător semi-octetului mai

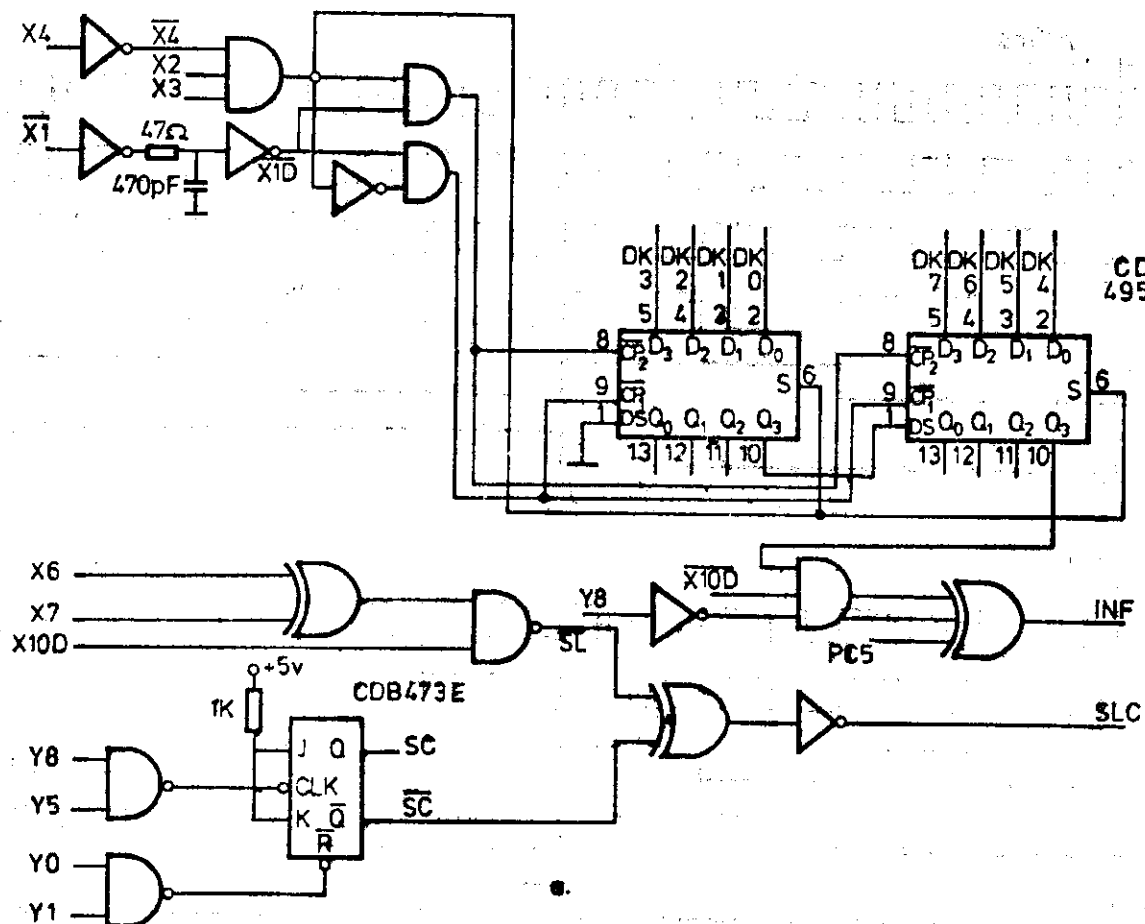


Fig. 3.20. Circuitele pentru afișare la televizor :

a) circuitele video ;

semnificativ toți cei 8 biți obținuți la o citire din memoria ecran. Ecuțiile logice ale semnalelor aplicate celor două registre CDB 495E, sînt :

$$S = X2.X3.\overline{X4}$$

$$\overline{CP1} = \overline{X1D} . (X2 X3 \overline{X4})$$

$$\overline{CP2} = \overline{X1D} . X2 X3 \overline{X4}$$

unde $\overline{X1D}$ este $\overline{X1}$ întrziat.

Informația de date serializată este mixată cu semnalele de stingere pe orizontală ($\overline{X10D}$), stingere pe verticală ($\overline{Y8}$) și bitul 5 al portului G din circuitul 8255 prin care se comandă video invers pentru întregul ecran.

Semnalul de sincronizare linii \overline{SL} se obține dintr-o schemă combinațională cu următoarea ecuație logică :

$$\overline{SL} = \overline{X10D} . (X6 \oplus X7)$$

Semnalul de sincronizare cadre \overline{SC} se obține de pe ieșirea \overline{Q} a unui bistabil ale cărui semnale de excitație au următoarele ecuații :

$$J/SC=1$$

$$K/SC=1$$

$$CLK/SC = \overline{Y5} . Y8$$

$$R/SC = \overline{Y0} . Y1$$

Semnalele de sincro linii și sincrocadre sînt mixate cu informație obținându-se semnalul sincrocomplex care este trimis la televizor direct sau printr-un modulator pentru intrare prin antenă, acordat în banda VHF, canalele 6-12.

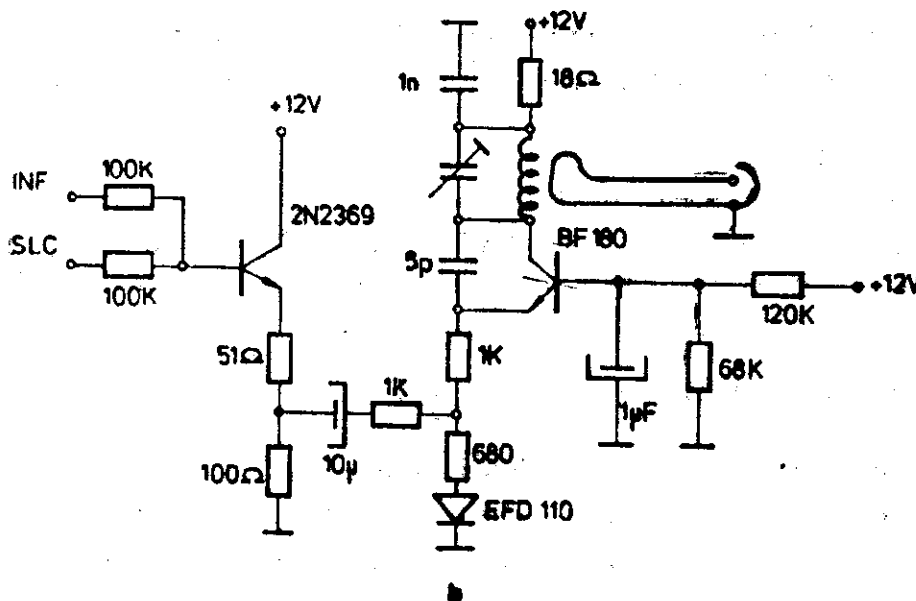


Fig. 3.20 : b) modulatorul.

3.7. Interfața de comunicație serială

Calculatorul personal are posibilitatea să comunice cu un alt calculator, direct sau prin modem și linie telefonică, datorită unei interfețe de comunicație seriale, realizată cu circuitul 8251. Viteza de transmisie/recepție este selectabilă de pe placă printr-un jumper între valorile 1200, 600 și 300 Baud.

Circuitul 8251 este un transmițător/receptor sincron-asincron universal (USART-Universal Synchronous/Asynchronous Receiver/Transmitter), destinat pentru comunicații de date între microcalculatoare. Acest circuit poate fi programat de unitatea centrală de prelucrare să lucreze utilizând orice tehnică uzuală de transmisie de date. USART acceptă caractere de la UCP, în format paralel și apoi le convertește în șiruri seriale continue pentru transmisie. Simultan poate să recepționeze șiruri de date seriale pe care le transformă în format paralel pentru a fi preluate de UCP. USART semnalizează unității centrale de prelucrare dacă poate accepta un nou caracter pentru transmisie sau dacă a recepționat unul. În orice moment se poate citi starea circuitului.

În figura 3.21 se prezintă configurația pinilor circuitului 8251 (a) și structura sa internă (b):

— RESET, activ pe nivel logic ridicat, forțează circuitul în stare inactivă, stare în care rămâne până la înscrierea cuvintelor de control care îi definesc funcționarea.

— CLK (Clock) reprezintă ceasul intern care determină temporizarea circuitului. Este necesar ca frecvența acestui semnal să fie mai mare decât de 30 de ori ceasul de recepție sau transmisie pentru modul sincron și de 4,5 ori pentru modul asincron.

— \overline{WR} (Write), intrare activă pe nivel logic coborât, indică că unitatea centrală de prelucrare înscrie date sau cuvinte de control în circuitul 8251.

— \overline{RD} (Read) indică faptul că UCP citește date sau informații de stare din USART.

— C/D (Control/Data) indică împreună cu intrările \overline{WR} și \overline{RD} dacă octetul de pe magistrala de date este caracter, cuvânt de control sau stare.

— CS (Chip Select), intrare activă pe nivel logic coborât, permite selecția circuitului 8251. Interpretarea ultimelor patru semnale de comandă este dată mai jos:

C/D	RD	WR	CS	Sensul transferului
0	0	1	0	8251 → magistrala de date
0	1	0	0	magistrala de date → 8251
1	0	1	0	stare → magistrala de date
1	1	0	0	magistrala de date → contro
X	1	1	0	dezactivat
X	X	X	1	dezactivat

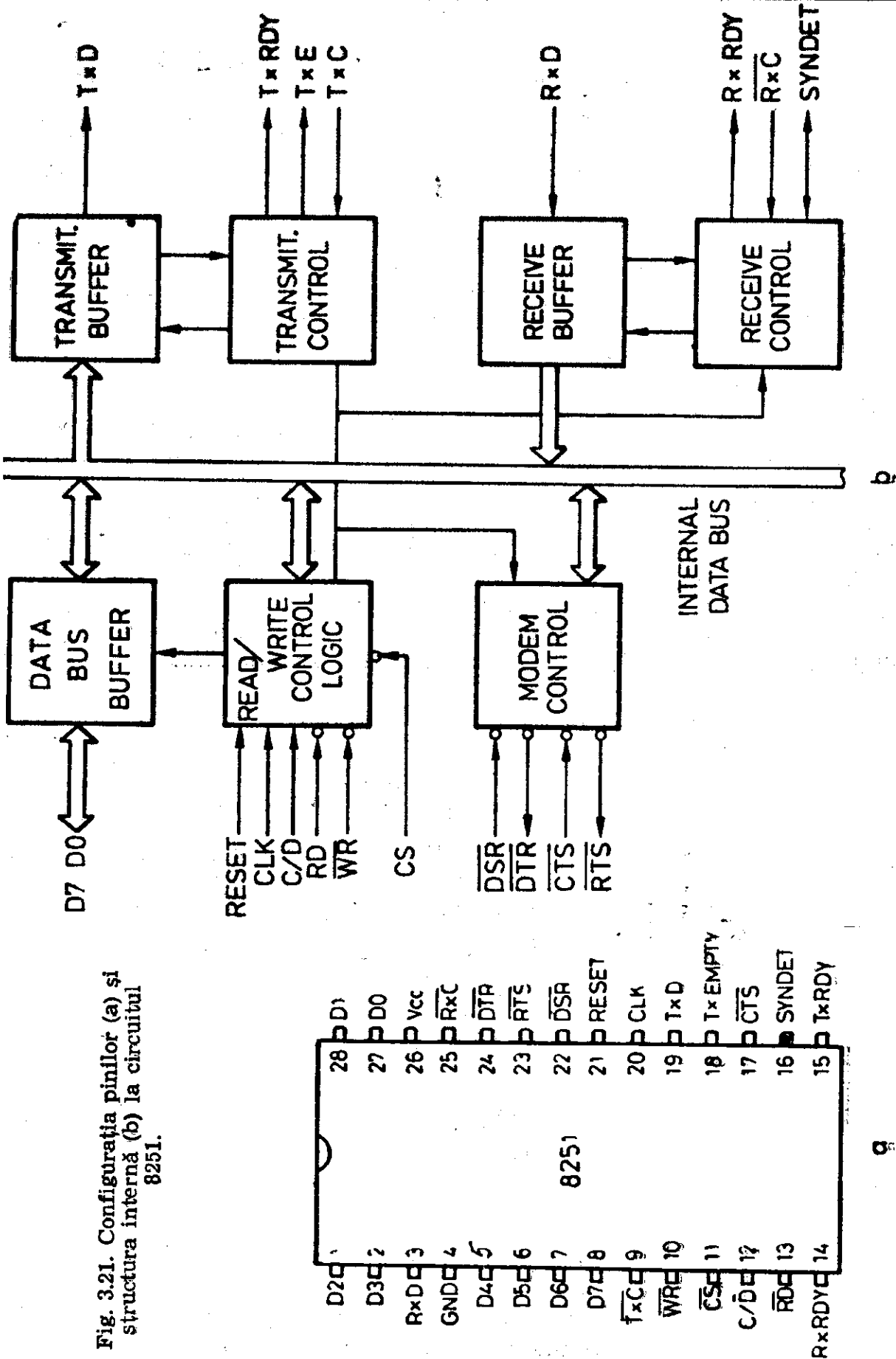


Fig. 3.21. Configurația pinilor (a) și structura internă (b) la circuitul 8251.

— $\overline{\text{DSR}}$ (Data Set Ready) este o intrare de uz general indicând modem pregătit. Se poate testa de către unitatea centrală de prelucrare printr-o operație de citire a stării.

— $\overline{\text{DTR}}$ (Data Terminal Ready) este o ieșire de uz general indicând circuit 8251 pregătit. Se poate activa prin program în cadrul cuvântului instrucțiunii de comandă.

— $\overline{\text{RTS}}$ (Request To Send) este de asemenea o ieșire de uz general. La fel ca și $\overline{\text{DTR}}$, se poate activa prin program.

— $\overline{\text{CTS}}$ (Clear to Send), intrare activă pe nivel logic coborât, permite circuitului 8251 să transmită date serial dacă bitul TxE din cuvântul de comandă este setat la 1 logic.

— TxRDY (Transmitter Ready) este o ieșire care anunță UCP că transmițătorul este gata să accepte un caracter.

— TxE (Transmitter Empty), ieșire activă pe nivel logic ridicat, indică unității centrale de prelucrare că circuitul 8251 nu are caractere de transmis.

— $\overline{\text{TxC}}$ (Transmitter Clock) reprezintă intrarea de ceas care determină frecvența caracterelor la transmisie. În modul de lucru sincron frecvența ceasului este egală cu frecvența de transmisie. În modul de lucru asincron frecvența ceasului poate fi 1x, 16x sau 64x rata de transfer, factorul de multiplicare fiind selectat de 2 biți din cuvântul de instrucțiune.

— RxRDY (Receiver Ready), ieșire activă pe nivel logic ridicat, anunță unitatea centrală de prelucrare că circuitul 8251 are un caracter recepționat.

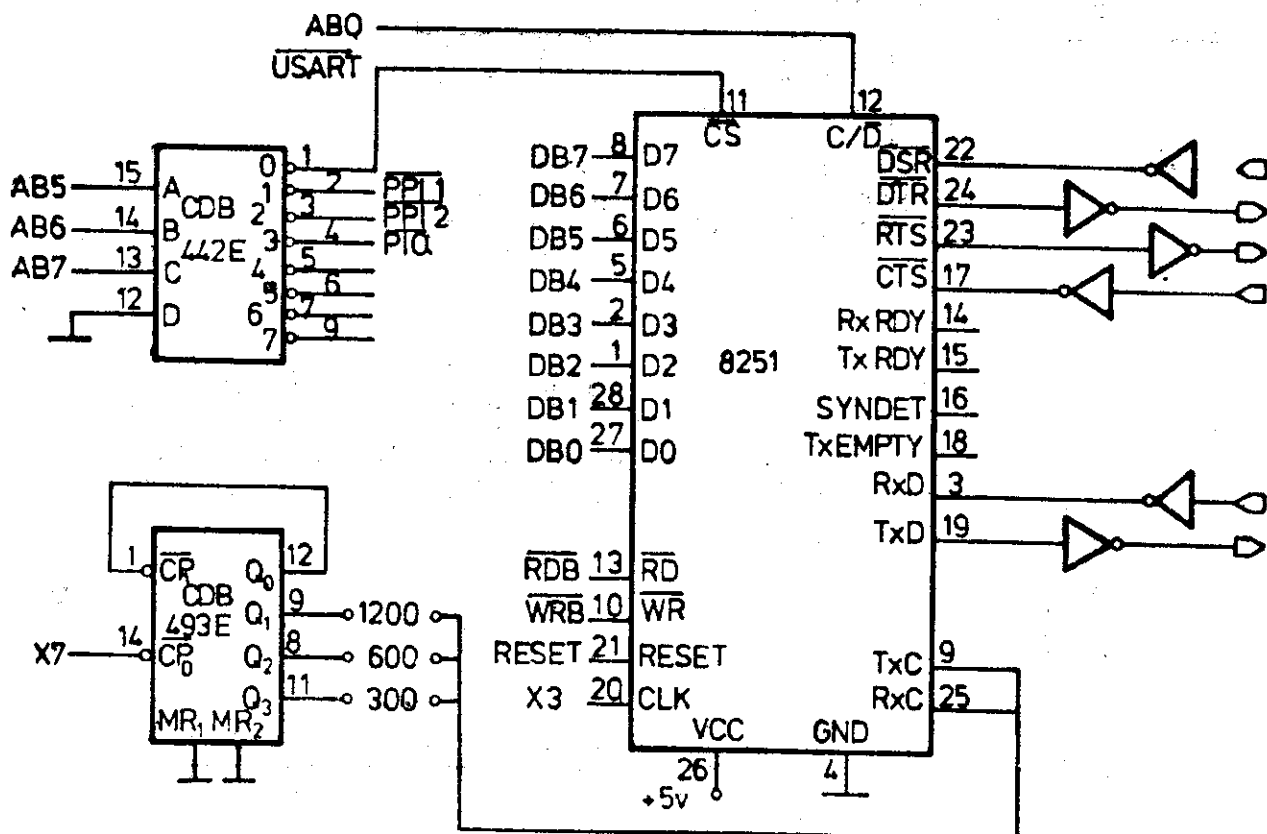


Fig. 3.22. Interfața de comunicație serială.

— $\overline{\text{RxC}}$ (Receiver Clock) este ceasul care controlează frecvența datelor la recepție. Se utilizează la fel cu semnalul $\overline{\text{TxC}}$.

— SYNDET (SYNC Detect) este folosit numai în modul de lucru sincron și se poate utiliza fie ca intrare, fie ca ieșire, funcție de modul de realizare a sincronizării (intern sau extern).

Schema interfeței de comunicație serială se prezintă în figura 3.22. Pe intrarea CLK a circuitului 8251 este conectat semnalul X3, cu frecvența 1,25 MHz, furnizat de sincrogenerator. Ca ceas pentru recepție/transmisie (intrările $\overline{\text{RxC}}$ și $\overline{\text{TxC}}$ sînt legate împreună) se utilizează un semnal selectabil printr-un jumper dintr-un set de semnale furnizate de un bloc de divizare cu 16 a lui X7. Se obțin frecvențele 1200×16 , 600×16 , 300×16 .

Intrarea C/D a circuitului 8251 este conectată la linia AB0 a magistralei de adrese. În acest fel cu adrese pare se selectează date, iar cu adrese impare se selectează comenzi/stări. De asemenea, intrarea CS este conectată la semnalul $\overline{\text{USART}}$, furnizat de decodificatorul adreselor porturilor de intrare/ieșire.

Semnalele de transmisie/recepție trec prin circuitele 1488/1489, care sînt driveri pentru standardul CCITT V 24.

3.8. Interfața pentru casetofonul audio

În scopul salvării programelor din memoria RAM a calculatorului personal se utilizează un casetofon audio obișnuit. Transferul datelor se poate face în ambele sensuri, viteza fiind de peste 16000 biți/s și se asigură o fiabilitate ridicată. Astfel se pot memora octeți pe casetă magnetică în fișiere cu următoarea structură :

- adresa de început a zonei care se transferă, pe 2 octeți în ordinea octet inferior, octet superior ;
- contorul de octeți, pe 2 octeți, de asemenea în ordinea octet inferior, octet superior ;
- datele ;
- suma de control ciclic pe un octet.

Un asemenea fișier înregistrat pe casetă se poate citi în memoria RAM a calculatorului personal, în acest fel se pot reface programe.

Schema interfeței pentru casetofonul audio este prezentată în figura 3.23. Este utilizat același circuit 8255 cu cel folosit pentru interfațarea tastaturii, difuzorului și ecranului.

În figura 3.24 este prezentată diagrama semnalului furnizat de interfața pentru casetofon la înregistrare (salvarea unei zone de memorie pe casetă). Timp de 10 secunde sînt generate impulsuri cu perioada de 1,19 ms și factor de umplere 1/2. Aceste impulsuri, care constituie preambulul, sînt necesare pentru poziționarea capului de citire la redare (citire) și pentru realizarea sincronizării.

După preambul urmează un singur impuls cu perioadă 0,4 ms care reprezintă impulsul de start și marchează începutul informației utile.

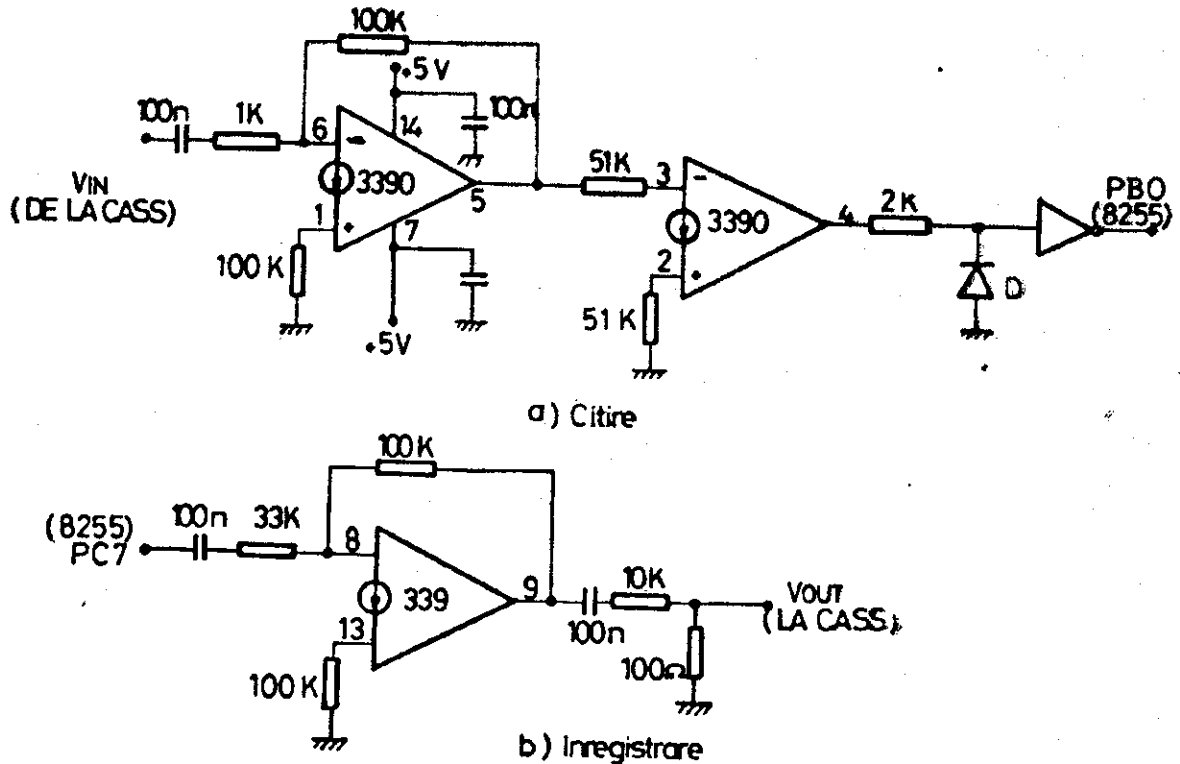


Fig. 3.23. Interfața pentru casetofonul audio.

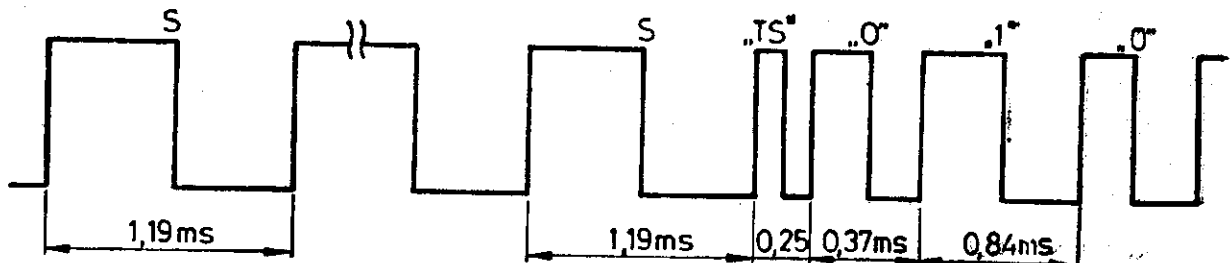


Fig. 3.24 Semnalul pentru casetofonul audio.

Tehnica de înregistrare pe casetă magnetică este următoarea: un impuls cu perioada $0,84$ ms pentru un bit 1 logic și un impuls cu perioada $0,37$ ms pentru un bit 0 logic. Toate impulsurile au factorul de umplere $1/2$ și sînt generate prin software. Astfel, fiecare octet, care urmează să fie salvat pe casetă, este serializat (primul fiind bitul 7) și prin portul PC, bitul 7 al circuitului 8255, sînt scoase impulsuri de $0,37/0,84$ ms pentru biți 0/1.

Realizată practic, această metodă de înregistrare a dat rezultate bune, obținindu-se o rată de transfer medie de 1600 biți/s, cu o densitate ridicată conferind, pentru o casetă de 60 minute, o capacitate de pînă la 950 Ko.

Atît modularea în durată a semnalului, care se înregistrează, cit și decodificarea informației recepționate de pe casetă sînt realizate prin software. La înregistrare, în funcție de tipul informației, ieșirea PC7 a interfeței programabile 8255 este ținută pe zero respectiv pe unu logic, pe durate bine stabilite.

Temporizările necesare sînt realizate ciclînd de un anumit număr de ori bucla următoare, care durează $12 \mu\text{s}$:

B1: IN PORTB
ANI 01
DCR B
JNZ B1:

Înainte de intrarea în buclă, registrul B va conține numărul 48, pentru jumătatea de semnal de sincronizare (S), 10 — în cazul semnalului de terminare a sincronizării (TS), 15 — și respectiv 34 — corespunzător biților de „0” și „1” logic din informație.

Înainte de livrarea informației spre casetă, se înregistrează un tren de impulsuri de sincronizare cu frecvență de cca. 0,8 KHz, avînd factorul de umplere de 0,5 sau un antet de recunoaștere a fișierului. Această secvență este necesară pentru reglarea nivelului de înregistrare, astfel încît, la sosirea datelor, înregistrarea să fie sigură. Informația este serializată tot software, prelucrarea ei prin hardware fiind minimă (o divizare cu 100, livrîndu-se spre casetofon la un nivel de ordinul a 50 mV).

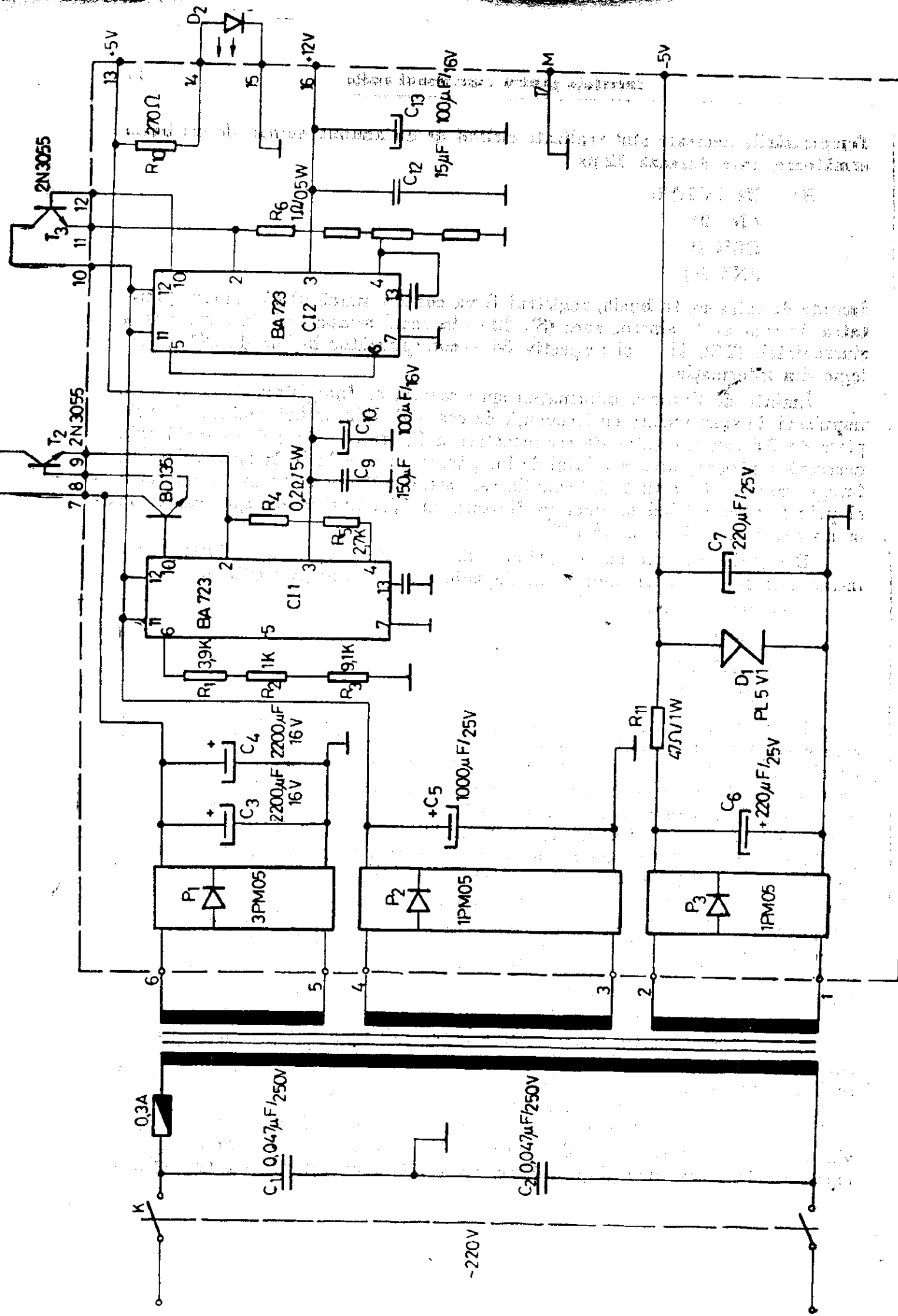
Decodificarea informației primite de pe casetă se face software, prin intrarea într-o buclă în momentul depistării unei tranziții pozitive.

R2: IN PORTB
INR B
ANI 01
JNZ B2

Deoarece la intrarea în buclă registrul B era nul, la ieșirea din buclă, care corespunde frontului negativ al semnalului, registrul B conține numărul de cicluri efectuate (un ciclu durînd tot $12 \mu\text{s}$), constituind astfel un criteriu de separare al semnalelor. În faza de sincronizare separarea se face între semnalele S și TS, prin compararea cu media aritmetică a numerelor de cicluri corespunzătoare lor, adică 29. În momentul în care semnalul a durat pe „1” un timp inferior la 29 cicluri, se consideră că informația ulterioară este formată de date. În cazul înregistrărilor cu antet de recunoaștere, datele vor fi preluate din momentul recunoașterii codului de fișier. În continuare defalcarea se face între biții de „1” și „0”, prin compararea registrului B cu 24. Pentru unele tipuri de casetofon, care au tendința să desimetrizeze semnalul redat, astfel încît factorul de umplere a semnalului ajuns în procesor este mai mic decît 50%, se compară cu o valoare mai mică.

Deoarece semnalul redat de pe casetă este desimetrizat diferit în funcție de volumul de redare (în cazul casetofonului fără ieșire standardizată), factorul de umplere al semnalului de decodificat variază în limite foarte largi cu reglarea volumului, compromițînd programul încărcat („1” poate fi luat „0” pentru un factor de umplere mult micșorat sau invers).

De aceea, pe porțiunea de sincronizare, la redarea de pe casetă, s-a prevăzut un control software al duratei pe „1” a semnalului de sincronizare care, dacă este în afara unei plaje admise, va avertiza operatorul pentru reglarea



volumului. Acest control îmbunătățește precizia de încărcare și avertizează de la început asupra unei nepotriviri a volumului, nefiind nevoie de încărcarea pînă la capăt a programului. De asemenea, în cazul în care viteza de rulare a benzii diferă constant, de aceea de la înregistrare, pe aceeași porțiune, de sincronizare, se poate face o reglare software automată a limitei de separare între două semnale de informație, permițînd încărcarea aceluiași program de pe casetofoane ale căror viteză de rulare a fost modificată.

La citirea unui fișier de pe casetofon se poziționează banda magnetică pe preambul. Se introduce de la tastatură comanda pentru refacerea fișierului în memorie, se pornește casetofonul în regim de redare și apoi se apasă tasta RETURN pentru executarea comenzii. Dacă înregistrarea are antet de recunoaștere, se poziționează banda înaintea fișierului cerut, se lansează comanda de încărcare și se comandă derularea benzi. Fișierul va fi recunoscut și încărcat conform codului specificat.

Dacă citirea unui fișier de pe casetă s-a executat corect (s-a verificat suma ciclică), atunci se afișează pe ecranul televizorului adresa de încărcare în memorie și numărul de octeți, în hexazecimal. În cazul apariției unei erori se afișează un semn de întrebare (?) și controlul revine monitorului. Dacă informația de pe casetă nu s-a alterat, se reia citirea.

3.9. Sursa de alimentare

Microcalculatorul aMIC are o sursă de alimentare externă care asigură tensiunile $\pm 5V$, $+12V$ necesare unei bune funcționări. Schema sursei de alimentare este prezentată în figura 3.25. Sursa este realizată cu stabilizatoare de tensiune integrate $\beta A723$ într-o configurație de stabilizator de tensiune pozitivă cu tranzistori NPN de tipul 2N3055, asigurînd stabilizare pe sarcină de 15 mV pentru $\Delta I_0 = 1\text{ A}$.

Tensiunea de $-5V$ după redresare cu o punte de tipul 1PM05 este stabilizată cu o diodă zener PL5V1.

Microprocesorul Z80. Interfețele programabile

4.1. Generalități

Microprocesorul Z80 este realizat în tehnologia NMOS, pe un circuit cu 40 de terminale. Față de microprocesorul 8080 prezintă o serie de perfecționări ca hardware și software.

Perfecționările hardware se referă la: utilizarea unei singure surse de alimentare, de 5V; încorporarea logicii generatorului de tact, care va necesita din exterior un semnal de ceas monofazic; prezența logicii pentru generarea unui semnal de reîmprospătare, necesar memoriilor dinamice; modificarea semnificației semnalelor de comandă pentru citire/scriere în sensul că se generează semnalele de citire și scriere, care se pot corela cu semnalele specificând o operație cu memoria sau de I/E; cererea externă de acces direct la memorie va conduce la intrarea magistralelor de date și adrese în starea de mare impedanță, la început de ciclu mașină; prezența unei linii de cerere de întrerupere nemascabilă, utilă în cazul tratării întreruperilor provocate de căderea tensiunii de alimentare.

În legătură cu software-ul se pot menționa următoarele: extinderea setului de instrucțiuni de la 78 la 158, menținându-se compatibilitatea la nivelul codului obiect, cu instrucțiunile microprocesorului 8080; duplicarea registrelor generale standard și a indicatorilor de condiții, ceea ce permite tratarea facilă a întreruperilor pe un singur nivel, prin simpla comutare pe setul suplimentar de registre, fără a se mai utiliza stiva organizată în memorie; adăugarea modulelor de adresare indexată, prin folosirea a două registre index; posibilitatea logicii externe de a răspunde la o recunoaștere a unei cereri de întrerupere prin forțarea unei instrucțiuni de tip chemare de subrutină, operație facilitată de existența unui registru al vectorului de întreruperi; existența unor instrucțiuni care permit transferul unor blocuri de informații, organizate în celule adiacente de memorie, în alte zone de memorie sau la un port de I/E; facilități de execuție a unor comparații pe blocuri; adăugarea unor instrucțiuni care testează sau modifică biți individuali în registre sau memorie.

Microprocesorul Z80 cuprinde în familia sa mai multe circuite, care oferă posibilitatea realizării unor sisteme cu un număr relativ mic de circuite. Trebuie menționat faptul că, în general, pot fi folosite și unele circuite din familia microprocesorului 8080. Dintre acestea se pot menționa: interfața paralelă programabilă 8255, interfața serială sincronă/asincronă 8251, etc.

Circuitele mai cunoscute din familia microprocesorului Z80 sînt următoarele :

Z8420 PIO — unitate de control pentru intrări/ieșiri paralele, conținînd două porturi de cîte opt biți, cu logică de dialog, generare de întrerupere și posibil tate de operare la nivel de octet sau de bit.

Z8440 SIO — unitate de control pentru intrări/ieșiri seriale, în modurile sincron și asincron, cu facilitățile necesare dialogului și verificărilor corectitudinii efectuării operațiilor.

Z8470 DART — unitate de control pentru intrări/ieșiri seriale în modul asincron, cu două canale distincte.

Z8430 CPG — unitate contor/periodizator cu patru contori programabili individual.

Z8410 DMA — unitate de acces direct la memorie, cu o rată de transfer de 2Mbiți, permițînd transferul datelor și/sau căutarea datelor.

4.2. Structura internă

Schema bloc a microprocesorului este prezentată în figura 4.1. În mare ea constă din : registrele generale, unitatea aritmetică-logică, registrul instrucțiunii, decodificatorul de instrucțiuni, unitatea de comandă și sincronizare, logica și circuitele tampon pentru adrese, interfață pentru magistrala de date

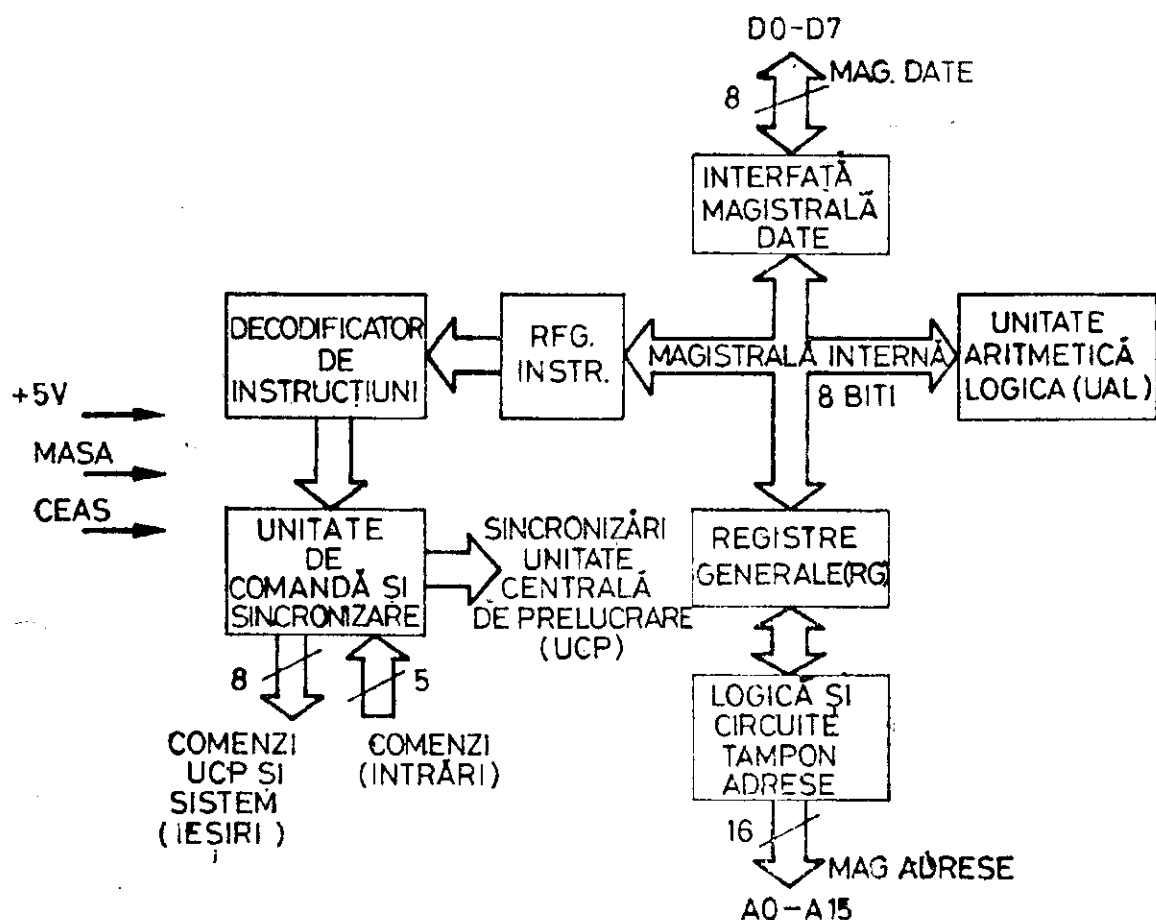


Fig. 4.1. Schema bloc a microprocesorului Z80

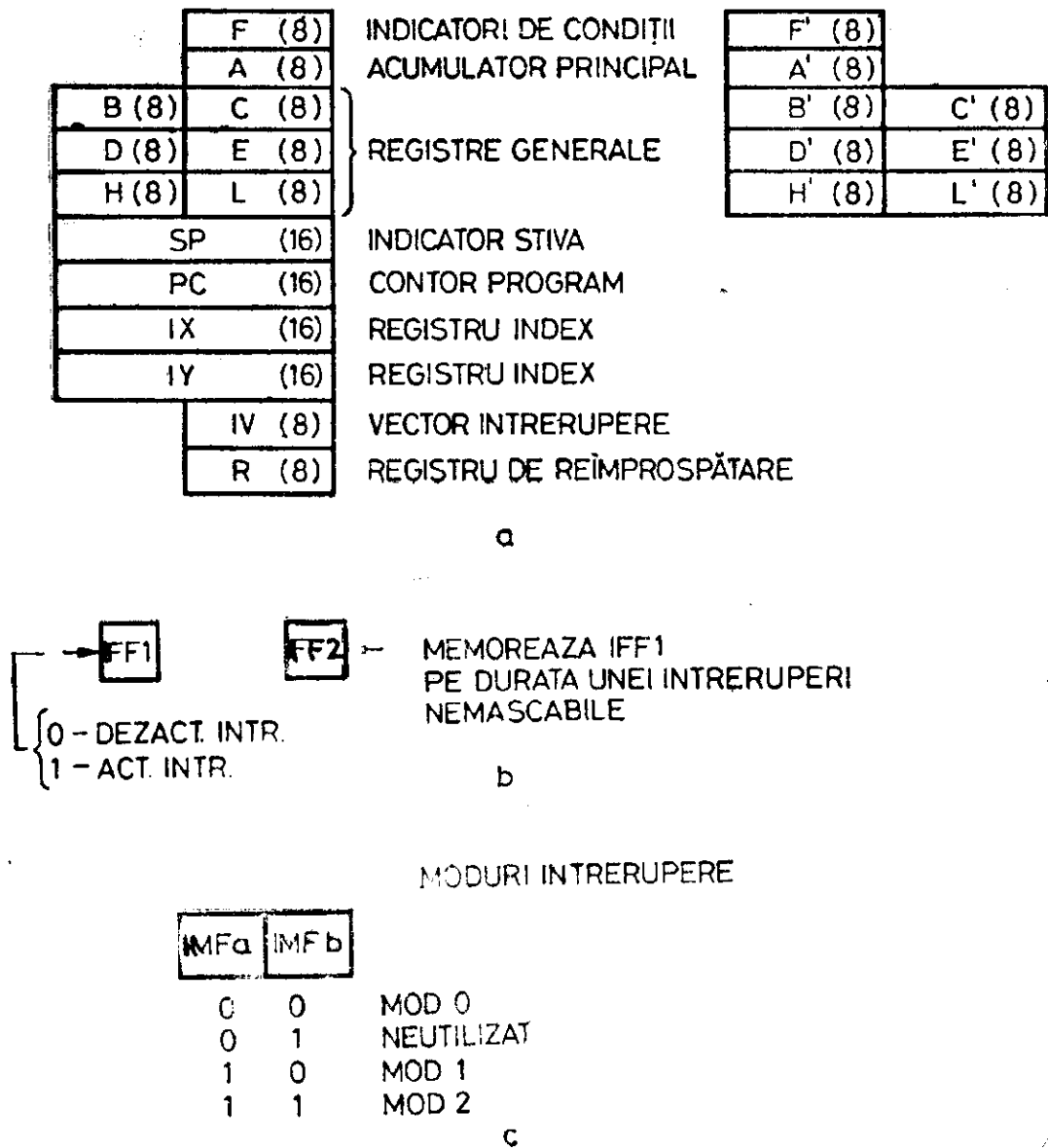


Fig. 4.2. Structura microprocesorului Z80 la nivel de registre :
 a) seturile de registre principale ; b) bistabilele stării intreruperii mascabile ; c) bistabilii modurilor de manipulare a intreruperii mascabile.

Din punctul de vedere al programatorului structura internă a microprocesorului Z80 apare ca în figura 4.2.

Microprocesorul Z80 dispune de două seturi de registre, având duplicate pentru registrele F, A, B, C, D, E, H, L. Trecerea de la un set de registre la altul se realizează prin instrucțiunea EXCHANGE (EX AF, AF').

Registrele acumulator (A,A') și registrele indicatoare de condiții (F,F').

Registrele acumulator sînt organizate pe 8 biți și au asociate registrele indicatoare de condiții.

Structura cuvintelor în registrele de condiții este dată în figura 4.3.

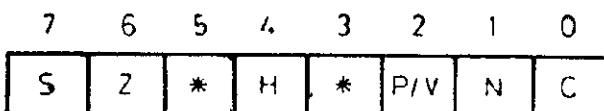


Fig. 4.3. Indicatorii de condiții.

Indicatorii de condiții sînt poziționați automat, ca urmare a operațiilor efectuate în UAL și pot fi testați prin instrucțiuni de transfer condiționat, în vederea efectuării unor transferuri ale comenzii, în program.

Semnificația lor este următoarea :

C — *transport*, se poziționează în 1 ca urmare a apariției unui transport în afara rangului de semn.

Z — *rezultat zero*, se poziționează în 1 la înregistrarea unui rezultat egal cu zero.

S — *semnul*, se poziționează în conformitate cu semnul rezultatului : 0 — pentru rezultat pozitiv sau zero și 1 — pentru rezultat negativ.

P/V — *paritate/depășire*, indică paritatea rezultatului în acumulator, în cazul operațiilor logice sau depășirea aritmetică, în cazul operațiilor cu numere reprezentate în complementul față de doi.

H — *transport auxiliar*, se poziționează în unu ca urmare a apariției unui transport/imprumut spre/de la bitul patru al acumulatorului.

N — *indicator de adunare/scădere*, specifică tipul instrucțiunii executate înaintea operației de corecție, la operarea în binar zecimal.

Registrele B-L, B'-L' pot fi folosite individual, ca registre de 8 biți sau asamblate în perechi B-C, D-E, H-L și B'-C', D'-E', H'-L' ca registre de 16 biți. Seturile de registre se pot selecta prin instrucțiunea EXX.

Registrul contorului programului PC, are 16 biți și indică adresa instrucțiunii următoare, în timpul execuției instrucțiunii curente.

Indicatorul de adrese al stivei SP, are 16 biți și conține adresa celulei din vârful stivei, care este organizată în memorie.

Registrele index IX și IY au câte 16 biți fiecare și păstrează adresa de indexare ; avînd o lungime de 16 biți ele se comportă, în raport cu adresele de 8 biți, furnizate de către UCP, ca niște registre de bază.

Registrul I, cu o lungime de 8 biți, permite adresarea indirectă a unei locații de memorie în urma unei cereri de întrerupere. Perifericul furnizează cei 8 biți mai puțin semnificativi, în timp ce primii 8 biți mai semnificativi sînt stocați în I. Astfel, rutinele de întrerupere pot fi lansate prin instrucțiuni de tip chemare de subrutine și pot fi plasate în orice zonă de memorie.

Registrul R este folosit pentru reînprospătarea memoriei dinamice. El este incrementat după fiecare ciclu de citire, conținutul său fiind transmis pe liniile de adresă A0-A6, simultan cu semnalul de comandă a reînprospătării în timpul execuției instrucțiunii, de către procesor. Această operație este transparentă pentru utilizator.

Microprocesorul Z80 acceptă două semnale de întrerupere : *nemascabile și mascabile*.

Bistabilele IFF1 și IFF2 specifică starea sistemului de întrerupere al microprocesorului, pentru întreruperile mascabile. Conținutul lui IFF1 indică activarea (IFF1=1) sau dezactivarea (IFF1=0) a sistemului pentru întreruperile mascabile. IFF2 va memora conținutul lui IFF1 pe durata servirii unei întreruperi nemascabile.

Bistabilele IMF_a, IMF_b specifică modurile programate pentru răspunsul la întreruperile mascabile : Modul 0-IMF_a, IMF_b=00 ; Modul 1-IMF_a, IMF_b=10 ; Modul 2-IMF_a, IMF_b=11.

4.3. Terminalele microprocesorului Z80 și semnalele asociate

În figura 4.4 sînt prezentate terminalele și semnalele corespunzătoare pentru microprocesorul Z80.

A0 ÷ A15 sînt liniile semnalelor de adrese, reprezentînd ieșiri cu trei stări, active pe nivel ridicat. Adresele sînt folosite pentru accesul la memorie (pînă

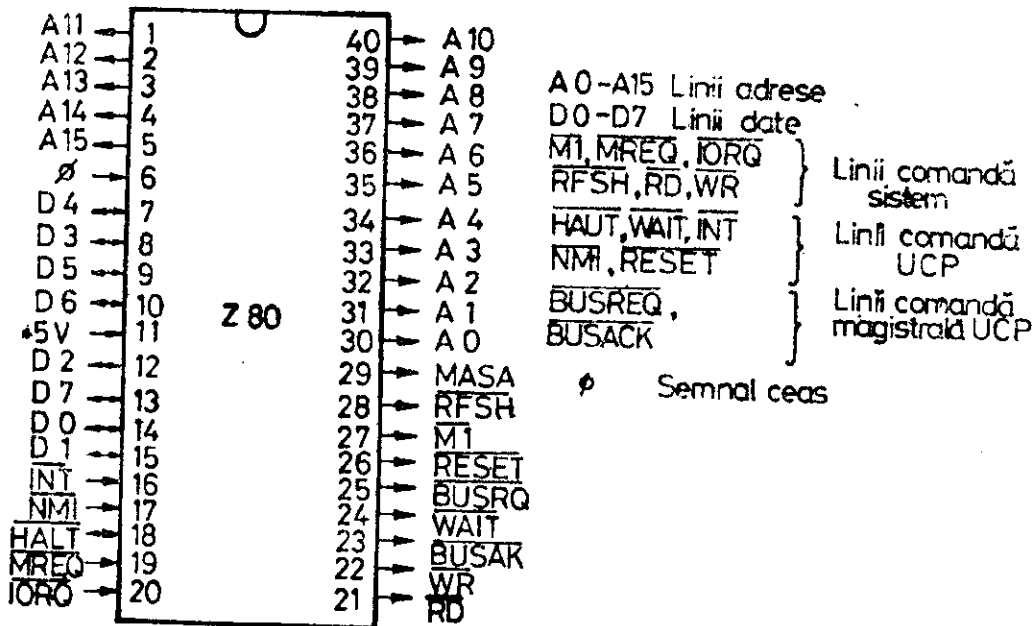


Fig. 4.4. Terminalele microprocesorului Z80.

a 64 Ko) și la porturile de intrare/ieșire. În ultimul caz se folosesc numai rangurile A0-A7 pentru a selecta unul din cele 256 porturi de I/E. A0 constituie bitul cel mai puțin semnificativ. Pe durata ciclului de reîmprospătare a memoriei, biții A0-A6 conțin adresa de reîmprospătare.

D0 ÷ D7 reprezintă liniile semnalelor de date care sînt transferate între microprocesor și memorie sau între microprocesor și porturile de intrare/ieșire. Semnalele sînt active pe nivel ridicat. Circuitele tampon ale microprocesorului, care comandă terminalele corespunzătoare acestor semnale, funcționează bidirecțional.

M1 reprezintă o linie de ieșire activă pe nivel coborît indicînd faptul că în ciclul mașină curent se citește pe magistrala de date un octet care constituie un cod de operație. El este activ și pe durata ciclului cînd se citește al doilea octet, reprezentînd un cod de operație pentru instrucțiunile cu doi octeți afectați codului operației. Aceste coduri, de doi octeți, încep întotdeauna cu : CB, DD, ED, FD în hexazecimal. M1 este activ simultan cu IORQ, pentru a indica un ciclu de recunoaștere a unei cereri de întrerupere, pe durata căruia se forțează din exterior, pe magistrala de date, un vector asociat cu rutina de tratare a întreprerii.

MREQ reprezintă o ieșire cu trei stări, activă pe nivel coborît specificînd faptul că la terminalele A0-A15 este prezentă adresa unei celule de memorie, în vederea unei operații de scriere/citire, cu memoria.

$\overline{\text{IOREQ}}$ constituie o ieșire cu trei stări, activă pe nivel coborât, indicând prezența unei adrese de port de I/E, pe biții A0-A7. Acest semnal este activ simultan cu $\overline{\text{M1}}$, în cazul în care se recunoaște o cerere de întrerupere, indicând faptul că un vector de răspuns, din partea echipamentului care a cerut întreruperea, poate fi plasat pe liniile D0-D7. Operațiile de I/E nu apar niciodată în cadrul ciclului M1, de citire a codului de operație a unei instrucțiuni.

$\overline{\text{RD}}$ este o ieșire cu trei stări, activă pe nivelul coborât, indicând faptul că procesorul solicită date de la memorie sau de la un port de intrare. Memoria sau portul adresate trebuie să forțeze data pe liniile D0-D7.

$\overline{\text{WR}}$ constituie o ieșire cu trei stări, activă pe nivel coborât, specificând prezența datelor furnizate de procesor pe liniile D0-D7 care pot fi înscrise în memorie sau la un port de ieșire.

$\overline{\text{RFSH}}$ ieșire, activă pe nivel coborât, specificând prezența la terminalele A0-A6 a adresei de reîmprospătare pentru memoria dinamică. Operația de reîmprospătare se realizează folosind semnalul curent $\overline{\text{MREQ}}$.

$\overline{\text{HALT}}$ ieșire activă pe nivel coborât, indicând faptul că procesorul a executat instrucțiunea $\overline{\text{HALT}}$ și așteaptă o întrerupere, nemascabilă sau o întrerupere mascabilă (dacă sistemul de întreruperi este activat), pentru a ieși din această stare. În $\overline{\text{HALT}}$ microprocesorul execută operații NOP (corespunzătoare instrucțiunilor neoperaționale) pentru a realiza reîmprospătarea memoriei.

$\overline{\text{WAIT}}$ intrare activă pe nivel coborât, care specifică procesorului că memoria sau portul adresate nu sînt pregătite pentru transferul de date, permițînd astfel, sincronizarea cu procesorul a unor memorii sau echipamente de I/E lente. Procesorul se menține în starea $\overline{\text{WAIT}}$ pe durata cît semnalul $\overline{\text{WAIT}}$ este activ.

$\overline{\text{INT}}$ intrare activă pe nivel coborât, reprezentînd o cerere de întrerupere solicitată de un echipament de I/E. Cererea va fi acceptată la sfîrșitul instrucțiunii curente, dacă bistabilul IFF1 este poziționat în unu și dacă semnalul $\overline{\text{BUSREQ}}$ nu este activ. La acceptarea cererii de întrerupere, procesorul va emite în ciclul mașină următor un semnal $\overline{\text{IORQ}}$ simultan cu semnalul $\overline{\text{M1}}$. În funcție de poziționarea bistabililor IMF_a, IMF_b, procesorul poate răspunde în trei moduri diferite la cererile de întrerupere mascabile.

$\overline{\text{NMI}}$ intrare activă pe front negativ, constituind cererea de întrerupere nemascabilă cu prioritate mai mare decît cererea $\overline{\text{INT}}$. Independent de starea bistabililor IFF1, IFF2, cererea $\overline{\text{NMI}}$ este recunoscută la sfîrșitul instrucțiunii curente, forțînd procesorul, după salvarea conținutului controlului programului în stivă, să execute instrucțiunea de la locația 0066H. Ciclurile continue $\overline{\text{WAIT}}$ vor face ca instrucțiunea curentă să nu se termine, astfel încît semnalul $\overline{\text{BUSRQ}}$ poate avea prioritate față de $\overline{\text{NMI}}$.

$\overline{\text{RESET}}$ intrare activă pe nivel coborât, care forțează în zero conținutul contorului programului și inițializează procesorul. Inițializarea are ca efect:

- dezactivarea bistabilului IFF1,
- anularea conținutului registrului IV
- anularea conținutului registrului R,
- stabilirea Modulului 0 pentru întreruperile mascabile.

Pe durata intervalului $\overline{\text{RESET}}$, liniile de adrese și de date trec în starea de mare impedanță, iar ieșirile reprezentând semnale de comandă devin inactive.

$\overline{\text{BUSRQ}}$ intrare activă pe nivel coborât, prin care se solicită din partea unui dispozitiv extern controlul asupra liniilor de adrese, date și comenzi, care trec în starea de mare impedanță.

La recepționarea semnalului $\overline{\text{BURSQ}}$ procesorul va trece liniile menționate mai sus în starea de mare impedanță, la terminarea ciclului mașină curent.

$\overline{\text{BUSAk}}$ ieșire activă pe nivel coborât, care indică unui dispozitiv extern trecerea liniilor de adrese, date și a unora din liniile de comenzi în starea de mare impedanță, care pot fi astfel controlate de către dispozitivul în cauză.

Φ semnal de ceas monofazic, cu frecvență maximă * de 6 MHz. Este generat din exterior.

4.4. Sincronizarea și execuția instrucțiunilor microprocesorului Z80

Instrucțiunile microprocesorului se desfășoară pe unul pînă la șase cicluri mașină (MC1 ÷ MC6). Fiecare ciclu mașină poate fi constituit din trei sau patru perioade de ceas (T1 ÷ T4), fiind posibilă inserarea unor perioade suplimentare de ceas (TW) între perioadele T2 și T3 (fig. 4.5).

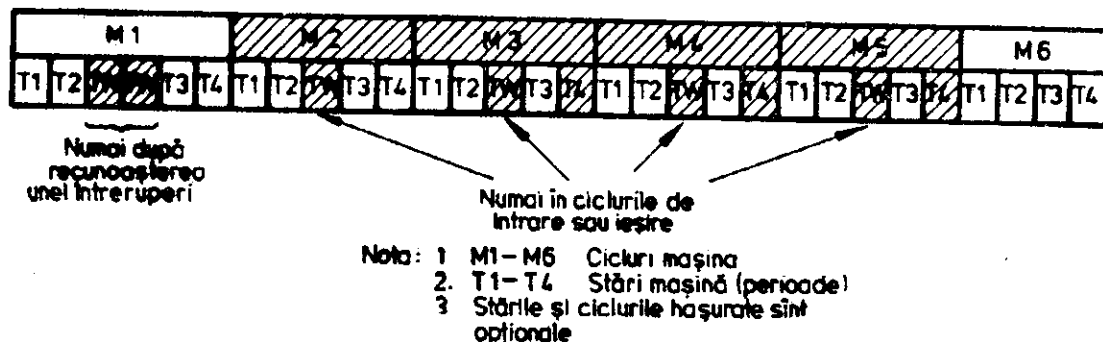


Fig. 4.5. Cicluri ale instrucțiunilor.

Pe durata unui ciclu mașină se execută o serie de operații specifice, care permit evidențierea a șapte cicluri mașină diferite :

- ciclul de citire a codului operației instrucțiunii (M1),
- ciclul de citire/scriere de la/in memorie,
- ciclul de I/E,
- ciclul de cerere/acceptare magistrală,
- ciclul de cerere/acceptare întrerupere mascabilă,
- ciclul de cerere/acceptare întrerupere nemascabilă,
- ciclul de ieșire din instrucțiunea HLT.

*) Z80 (f. max.=2,5 MHz) ; Z80A (f. max=4MHz) ; Z80B (f. max.=6MHz)

Durata unei perioade de ceas T_i este dată de frecvența maximă a semnalului de ceas, pentru microprocesorul Z80 cu care lucrează. De exemplu, frecvența maximă este de 4MHz conduce la o perioadă cu durata de 250 ns.

În figura 4.6 se prezintă ciclul mașină M1. Ciclul M1 este identificat prin activarea semnalului $\overline{M1}$, pe duratele perioadelor T1 și T2. Conținutul contorului programului, reprezentînd adresa instrucțiunii curente, este prezent pe

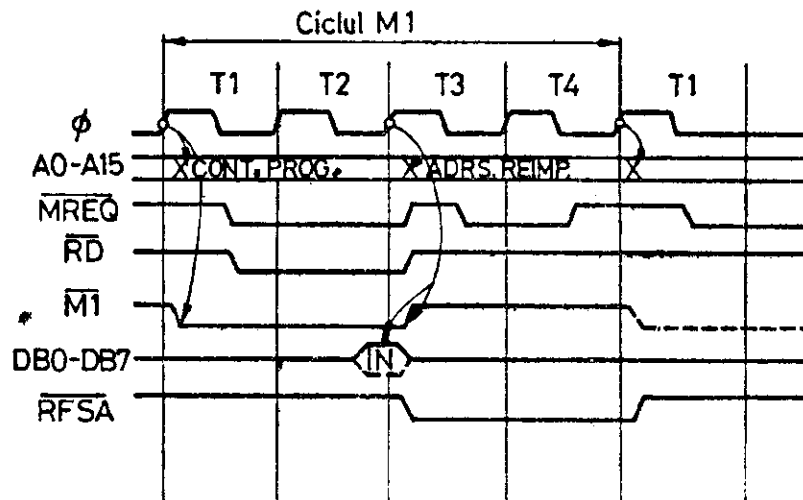


Fig. 4.6. Ciclu M1.

liniile de adrese $A_0 \div A_{15}$, tot pe durata T1-T2. Semnalele \overline{MREQ} și \overline{RD} devin active la jumătatea perioadei T1 și rămîn în această stare pînă la începutul perioadei T3.

Deoarece pe frontul căzător al semnalului Φ în T2, terminalul WAIT se afla la un nivel ridicat, nu se va intra, după perioada T2, într-o perioadă de așteptare TW.

Datele sînt citite de către procesor, de pe magistrala de date ($DB_0 \div DB_7$), pe frontul crescător al semnalului de ceas, în T3. Perioadele T3, T4 sînt folosite pentru operații interne în microprocesor și pentru reîmprospătarea memoriei, pe liniile $A_0 \div A_{15}$ fiind prezentă adresa de reîmprospătare. Semnalul \overline{MREQ} devine activ în a doua jumătate a perioadei T3 și rămîne activ pe durata activă a perioadei T4. De asemenea, semnalul \overline{RFSH} este activ pe durata perioadelor T3 și T4.

Operația de citire din memorie

Între ciclul de citire din memorie a codului operației și ciclul de citire a unei date sînt cîteva diferențe care trebuie menționate. Astfel, un ciclu M1 are patru perioade, în timp ce un ciclu de citire a unei date are numai trei perioade. În primul caz data furnizată de memorie este strobată pe frontul anterior al semnalului de ceas T3; în cazul al doilea strobarea se face pe frontul căzător al semnalului T3 (fig. 4.7).

Trebuie amintit că pe durata ciclurilor M1 semnalul $\overline{M1}$ este activ pe nivel coborît, în cadrul primelor două perioade de ceas T1 și T2.

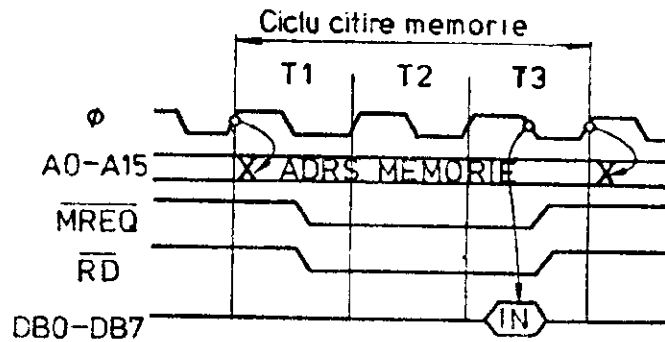


Fig. 4.7. Ciclu de citire a unei date din memorie.

Operația de scriere în memorie

Această operație se aseamănă cu operația precedentă cu observația că \overline{WR} este adus în starea activă, pe nivel coborât, în a doua jumătate a perioadei T2 și în prima jumătate a perioadei T3. Datele furnizate de microprocesor fiind stabile în acest interval, \overline{WR} poate fi folosit ca strob de către memorie pentru înscrierea datelor în ea (fig. 4.8).

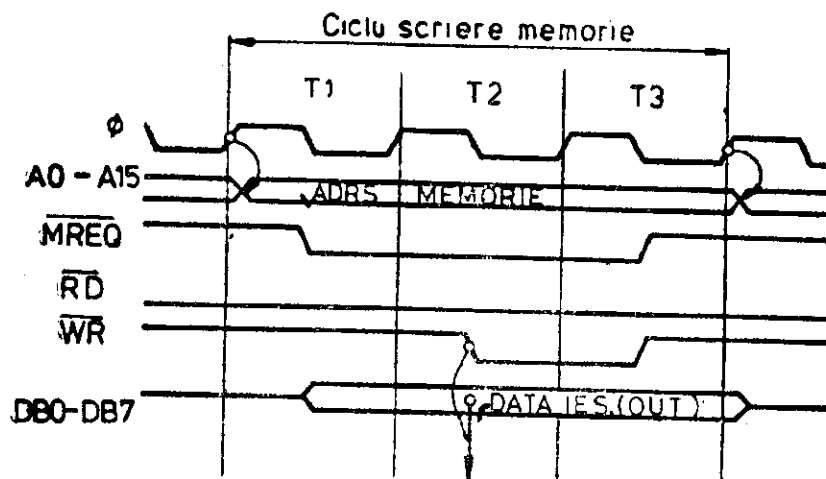


Fig. 4.8. Ciclu de scriere a unei date în memorie.

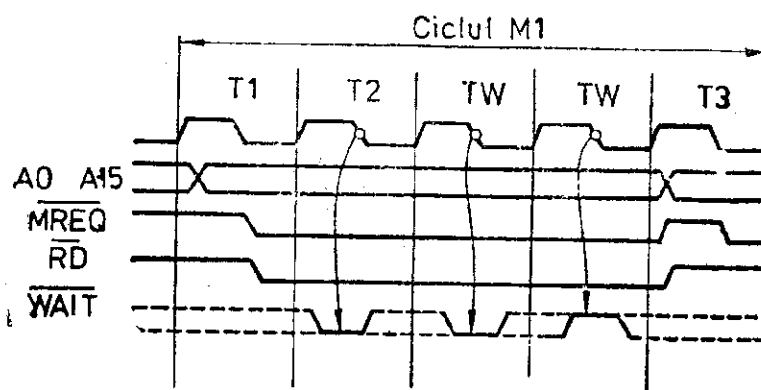


Fig. 4.9. Intrarea în starea WAIT.

Starea WAIT

Starea WAIT apare la microprocesorul Z80 între perioadele T2 și T3, atunci când logica externă sau memoria nu pot opera la viteza microprocesorului. Semnalul de la terminalul $\overline{\text{WAIT}}$, furnizat din exterior este testat pe frontul căzător al semnalului de ceas, în perioada T2 (fig. 4.9). Dacă semnalul $\overline{\text{WAIT}}$ este la nivel coborât, în timpul acestui test, automat se va introduce o perioadă de ceas TW, între T2 și T3. În cazul în care $\overline{\text{WAIT}}$ capătă un nivel ridicat, testul pe frontul căzător al semnalului de ceas în TW, va conduce la inițierea stării T3, în perioada următoare de ceas. Microprocesorul Z80 nu va furniza în exterior un semnal care indică intrarea în starea WAIT.

Operațiile de intrare/ieșire

Ciclurile de execuție pentru instrucțiunile de intrare/ieșire au inserată automat câte o perioadă de ceas TW, între perioadele T2 și T3, pentru a permite adaptarea microprocesorului la ritmul de lucru al logicii de I/E.

Echipamentele de I/E pot fi selectate ca și celulele de memorie, în spațiu de adresare al acesteia. În cazul în care nu se vor insera stări TW, logica respectivă trebuie să funcționeze la viteza microprocesorului.

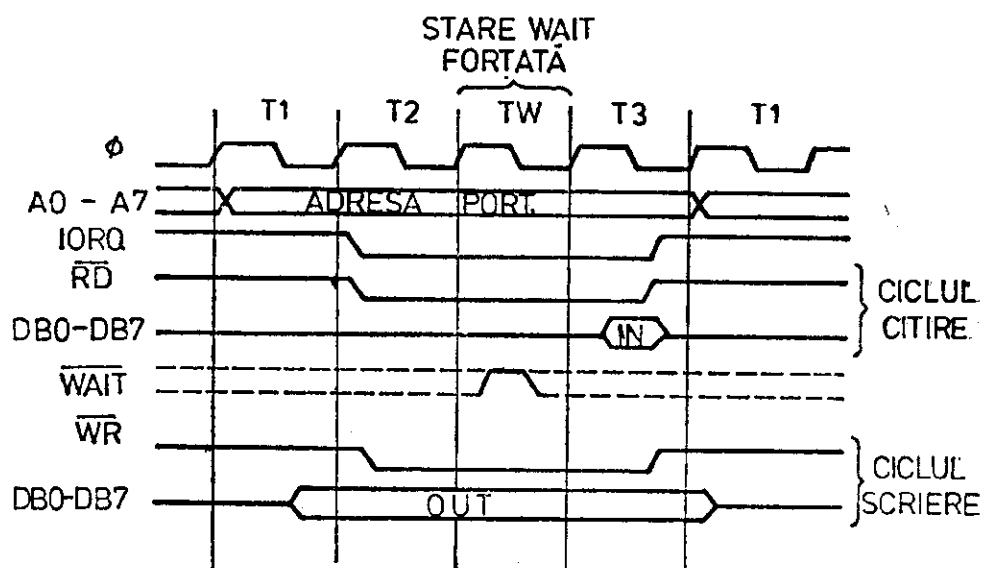


Fig. 4.10. Cicluri de execuție pentru I/E, fără inserție de stări WAIT.

În figura 4.11 se prezintă ciclurile de I/E fără inserție de stări TW, iar în figura 4.12 cicluri de I/E cu inserție de stări TW.

Se constată că adresa postului de I/E este prezentă pe liniile A0-A7 pe toată durata ciclului. Semnalele $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ sau $\overline{\text{WR}}$ sînt active pe duratele perioadelor T2, TW și T3 pînă la frontul căzător al semnalului de ceas din această ultimă perioadă. Datele de intrare sînt strobate pe frontul căzător al semnalului de ceas din perioada T3. Datele de ieșire sînt stabile începînd cu frontul căzător al semnalului de ceas, în T1, pe toată durata ciclului de execuție.

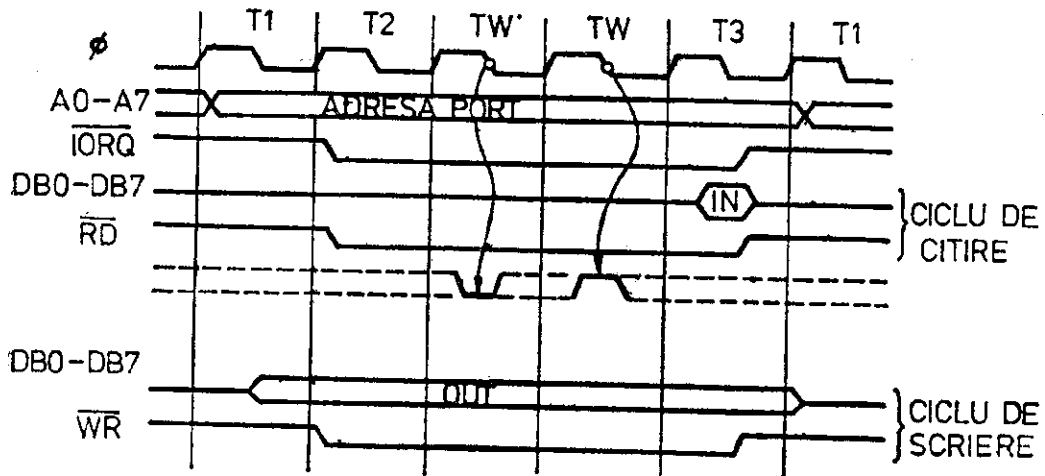


Fig. 4.11. Cicluri de execuție pentru I/E, cu inserție de stări suplimentare WAIT.

Cererile de magistrală

Semnalul extern aplicat la terminalul $\overline{\text{BUSRQ}}$ este testat pe frontul crescător al semnalului de ceas, în ultima perioadă a fiecărui ciclu mașină. Dacă acest semnal este activ, pe nivel coborât, atunci toate terminalele de adrese, date și comenzi, comandate cu circuite tampon, intră în starea de mare impedanță. Această situație este semnalizată de către microprocesor prin activarea semnalului $\overline{\text{BUSAK}}$, pe nivel coborât. Semnalul $\overline{\text{BUSRQ}}$ este testat pe frontul crescător al semnalului de ceas, în cadrul fiecărei perioade. La dezactivarea lui $\overline{\text{BUSRQ}}$, microprocesorul va comanda ieșirea din starea de mare impedanță a liniilor de adrese, date și comenzi, la începutul următoarei perioade de ceas.

În cazul în care se folosește pentru reîmprospătarea memoriei semnalul furnizat de microprocesorul Z80, pe duratele unor perioade mai lungi în cadrul cărora liniile menționate mai sus sînt în stare de mare impedanță, o logică externă trebuie să furnizeze memoriei comanda de reîmprospătare și adresele

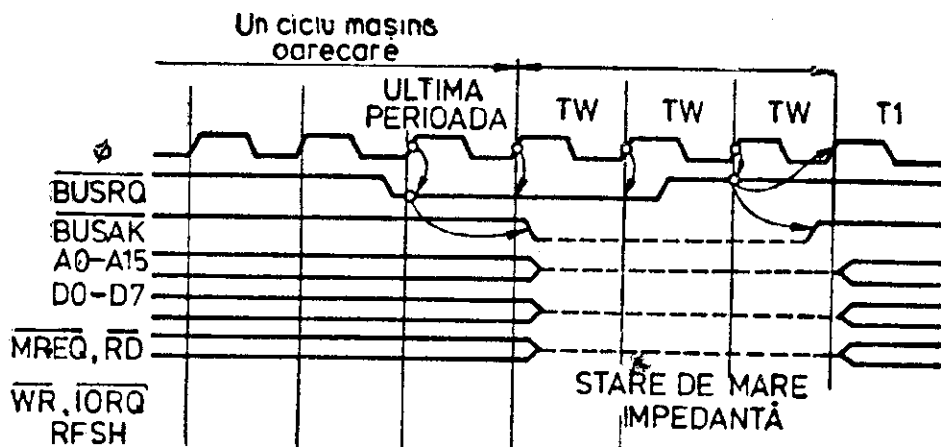


Fig. 4.12. Diagrama temporală pentru cereri/acceptare de magistrală.

asociate. Pentru a evita această situație, se impune ca accesul direct la memorie să se execute pe perioade de scurtă durată.

Figura 4.12 conține diagramele temporale pentru cererile de magistrale.

4.5. Întreruperile externe

Microprocesorul Z80 posedă două intrări pentru semnalele de întrerupere: $\overline{\text{NMI}}$ și $\overline{\text{INT}}$. Cererea de întrerupere nemascabilă $\overline{\text{NMI}}$ are prioritate față de cererea de întrerupere mascabilă $\overline{\text{INT}}$. Semnalul de cerere de întrerupere $\overline{\text{INT}}$ este testat de către unitatea centrală pe frontul crescător al ultimei perioade de ceas, în cadrul ultimului ciclu de execuție a fiecărei instrucțiuni. Cererea de întrerupere $\overline{\text{INT}}$ nu va fi luată în considerație, dacă întreruperile au fost dezactivate prin program sau dacă semnalul $\overline{\text{BUSRQ}}$ este activ pe nivel coborât. Astfel, cererile de acces la magistrală vor avea prioritate față de cererile de întrerupere mascabile.

Recunoașterea unei întreruperi $\overline{\text{INT}}$ este realizată prin generarea unor semnale active $\overline{\text{MI}}$ și $\overline{\text{IORQ}}$. Acestea apar în cadrul unui ciclu mașină special, de întrerupere, care conține două perioade TW inserate între perioadele T2 și T3, ceea ce va permite logicii externe să plaseze un vector de întrerupere pe magistrala de date.

Vectorul de întrerupere poate capăta o formă care depinde de modul selectat prin program.

Modul 0 va interpreta vectorul de întrerupere ca un cod obiect de un octet, care va forța contorul programului la una din adresele următoare 0000H, 0008H, 0010H, 0018H, 0020H, 0028H, 0030H, 0038H. Codul obiect al acestei instrucțiuni este 11XXX111, unde XXX ia valori cuprinse între 000 și 111, corespunzător locațiilor menționate mai sus. Codul sursă este acela al instrucțiunii RSTN. (Restart N), N fiind codul octal pentru biții XXX, amintiți mai înainte.

Modul 1 presupune în mod automat că prima instrucțiune, care se va executa după răspunsul la cererea de întrerupere, va fi o instrucțiune RST7, care va forța execuția programului de la adresa 0038H. În acest caz nu va mai fi necesară forțarea unei instrucțiuni din exterior.

Modul 2 a fost proiectat pentru a utiliza mai eficient posibilitățile microprocesorului Z80 și ale circuitelor din familia acestuia. Echipamentul periferic, care solicită întreruperea, selectează adresa de start a rutinei de tratare a întreruperii. Aceasta se realizează prin plasarea unui vector de adresă, de opt biți, pe magistrala de date, pe durata ciclului de recunoaștere a întreruperii. Octetul de ordin superior este furnizat de registrul I. Prin aceasta rutinele pot fi plasate la orice locație în memorie. Deoarece echipamentul furnizează octetul inferior al unui vector cu doi octeți, bitul A0 trebuie să fie zero.

Dacă două stări WAIT nu sînt suficiente pentru ca logica externă să poată arbitra prioritățile cererilor de întrerupere, pentru a plasa vectorul necesar de întrerupere, se pot insera stări WAIT adiționale.

În figura 4.13 se prezintă diagrama semnalelor pentru cazul răspunsului la o întrerupere externă.

Întreruperile nemascabile nu pot fi dezactivate prin program, fiind acceptate de microprocesor în orice moment. Ele sînt asociate cu evenimente cu cea mai mare prioritate, cum ar fi căderea tensiunii de alimentare. După recu-

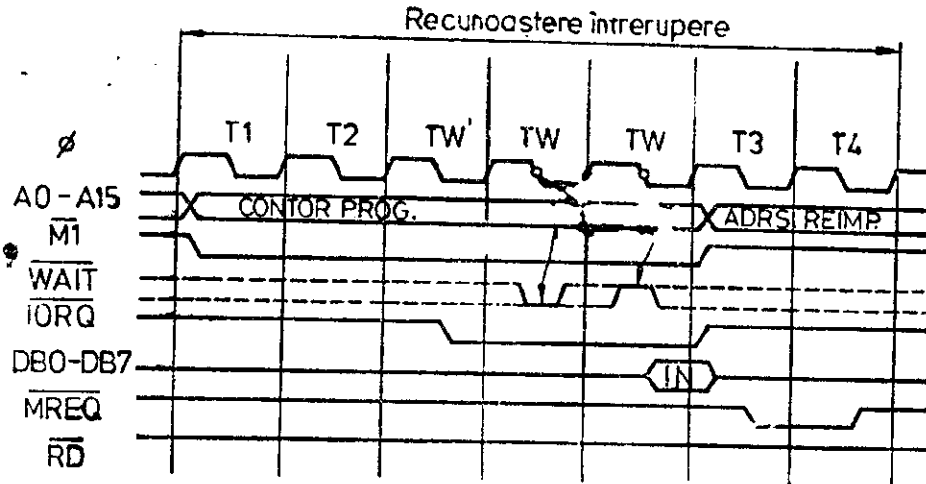


Fig. 4.13. Diagrama răspunsului la o cerere externă de întrerupere mascabilă cu inserția unei stări TW suplimentare.

noașterea unei întreruperi $\overline{\text{NMI}}$ (dacă $\overline{\text{BUSREQ}}$ nu este activ), microprocesorul va efectua transferul la locația 0066H . Pe baza conținutului acestei locații se intră în rutina de tratare a întreruperii. Întreruperea nemascabilă operează

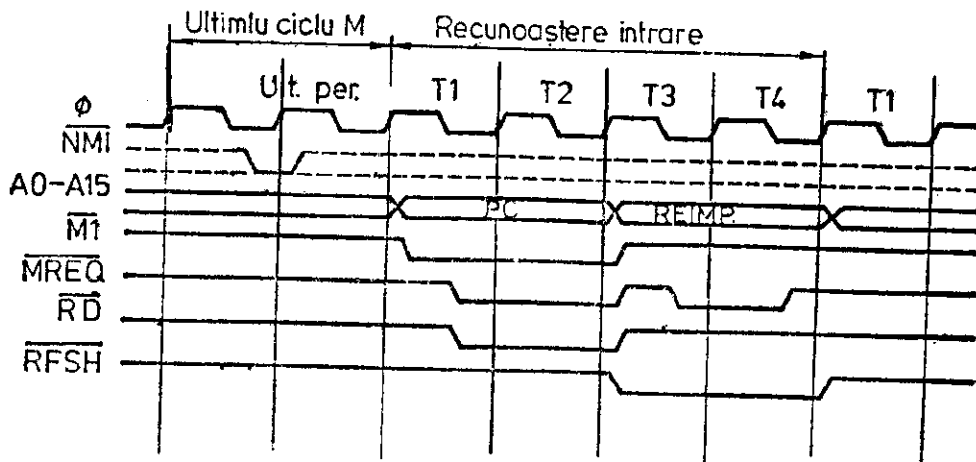
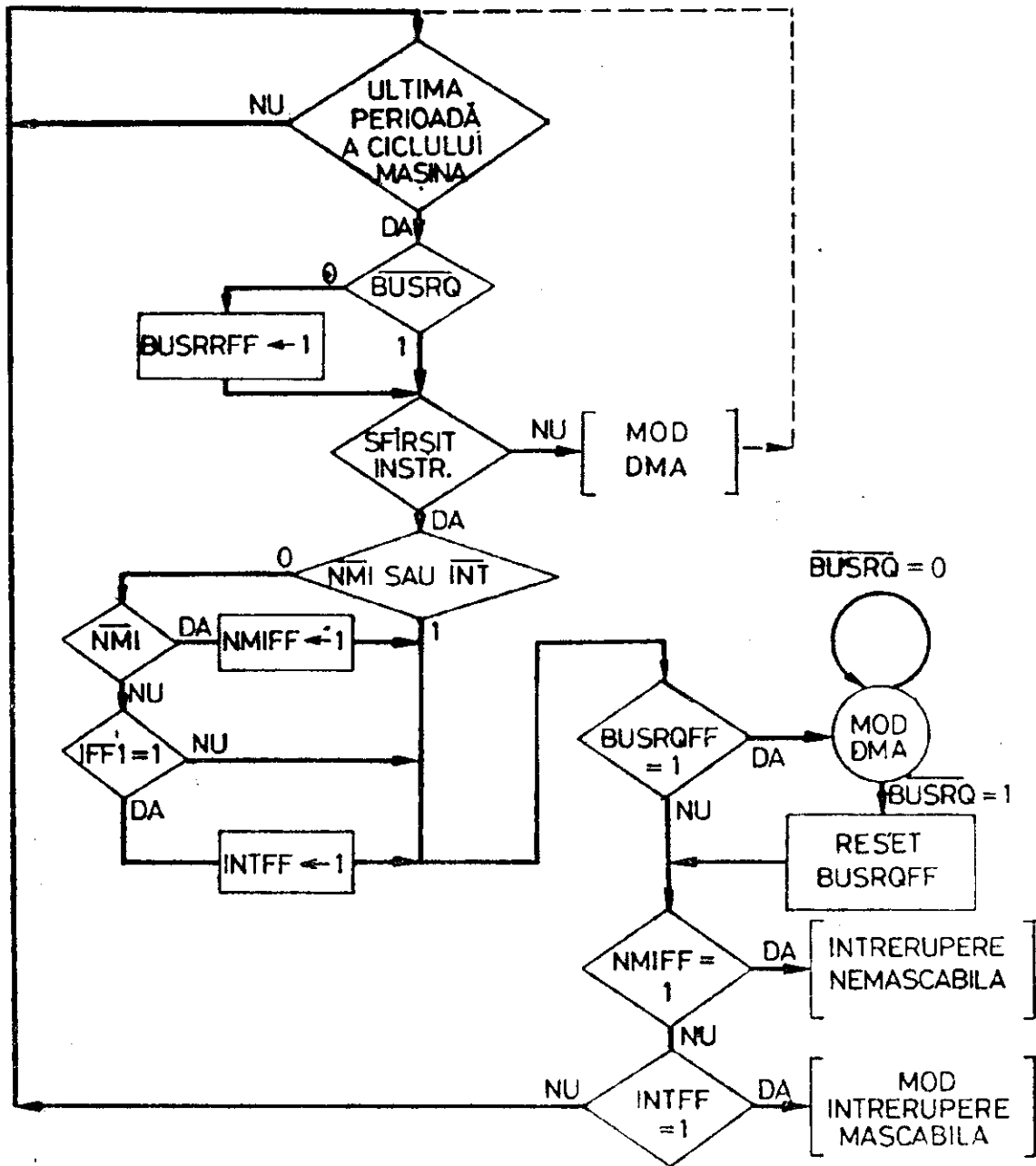


Fig. 4.14. Diagrama răspunsului la o cerere de întrerupere nemascabilă.

numai în Modul 1. În figura 4.14 se prezintă diagramele de semnale în cazul răspunsului la o întrerupere nemascabilă.

Interacțiunea între cererile $\overline{\text{INT}}$, $\overline{\text{NMI}}$ și $\overline{\text{BUSREQ}}$ este prezentată în figura 4.15.



NOTA:

1. \overline{BUSRQ} este testat la sfîrșitul fiecărui ciclu masină.
2. \overline{INT} , \overline{NMI} sînt testate în ultima perioadă a ultimului ciclu masină al instrucțiunii.
3. Pe durata cedării magistralei ($\overline{BUSAK} = 0$) nu se răspunde la cererile \overline{INT} , \overline{NMI} .
4. Ordinea priorităților este: \overline{BUSRQ} , \overline{NMI} , \overline{INT} .
5. $BUSRQFF$, $NMIFF$, $INTFF$: bistabile în care se memorează prezența cererilor corespunzătoare.

Fig. 4.15. Interacțiunea între \overline{INT} , \overline{NMI} , \overline{BUSRQ} .

4.6. Starea HALT

Execuția unei instrucțiuni HALT, de către microprocesorul Z80, se realizează prin operarea unei secvențe de instrucțiuni NOP (neoperaționale), pînă la apariția unei cereri de întrerupere.

Semnalele $\overline{\text{NMI}}$ și $\overline{\text{INT}}$ sînt testate la fiecare perioadă T4, pe durata frontului crescător al semnalului de ceas, în cadrul fiecărui ciclu al instrucțiunii NOP.

În figura 4.16 se prezintă diagramele de semnale pentru instrucțiunea HALT.

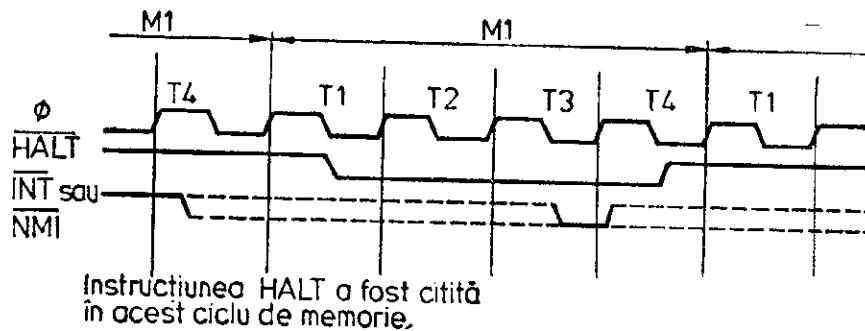


Fig. 4.16. Diagrama de semnale pentru instrucțiunea HALT.

În figura 4.17 se prezintă ciclul RESET. Semnalul $\overline{\text{RESET}}$ trebuie să fie activ pe durata a cel puțin trei perioade de ceas, pentru ca efectul său asupra UCP să fie cel așteptat.

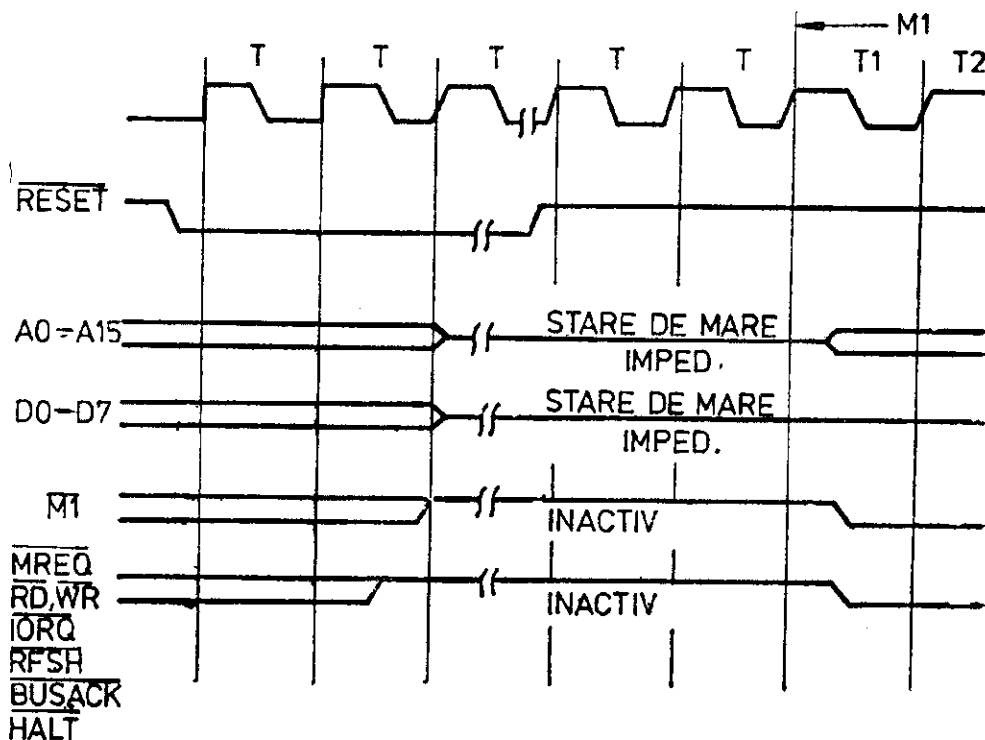


Fig. 4.17. Diagrama de semnale pentru ciclul RESET.

Liniile de adrese și de date trec în starea de mare impedanță, iar liniile de comandă devin inactive, pe durata ciclului RESET. După dezactivarea semnalului RESET, se așteaptă două perioade T, pentru ca UCP să intre într-un ciclu normal M1. Deoarece conținutul lui CP este anulat, procesorul va citi instrucțiunea de la adresa 0000H din memorie.

4.7. Instrucțiunile microprocesorului Z80

O instrucțiune este reprezentată de un set de caractere care definesc o operație și eventual alte informații necesare calculatorului pentru a executa instrucțiunea dată.

Operațiile pe care le poate executa microprocesorul Z80 pot fi grupate astfel:

- transferuri de date,
- operații aritmetice și logice,
- deplasări și rotații,
- manipulări de biți,
- transfer al comenzii, chemări de subrutine, reveniri din subrutine,
- operații de I/E și de comandă a microprocesorului.

În cadrul repertoriului de instrucțiuni ale microprocesorului Z80 se întâlnesc instrucțiuni pe unul, doi, trei sau patru octeți. Numărul de octeți ai unei instrucțiuni este legat de complexitatea instrucțiunii și informația pe care o necesită. Repertoriul de instrucțiuni al microprocesorului Z80 conține ca subset repertoriul de microinstrucțiuni al microprocesorului 8080. Astfel, programele scrise pentru 8080 sînt direct executabile de către microprocesorul Z80. Compatibilitatea este asigurată la nivel de cod mașină și nu la nivelul codului sursă în limbaj de asamblare.

Formatul instrucțiunilor cuprinde: codul de operație, pe unul, doi sau trei octeți; data, pe unul sau doi octeți; deplasarea, pe un octet; codul echipamentului, pe un octet; adresa absolută sub forma a doi octeți succesivi-octetul inferior și octetul superior, etc.

Codul de operație specifică funcția pe care o execută instrucțiunea.

Data constituie o informație binară avînd opt ranguri care reprezintă un operand, pentru operațiile aritmetice/logice, de memorare, de I/E, etc. Ea poate reprezenta un cod zecimal codificat binar sau un cod ASCII.

Codul echipamentului identifică numărul portului de I/E cu care se face schimbul de informație. Acesta are valori zecimale cuprinse între 0 și 255.

Adresa unei celule de memorie este constituită din doi octeți, întrucît microprocesorul Z80 poate adresa direct 65536 octeți de memorie. Instrucțiunea cuprinde, după codul operației, octetul inferior mai puțin semnificativ și apoi octetul superior mai semnificativ al adresei.

Deplasarea constituie informația de un octet care se adună la conținutul unuia din cele două registre index IX, IY, pentru a forma adresa unei celule de memorie.

Deplasarea se reprezintă în complementul față de doi, luând valori pozitive și negative cuprinse între (+127 și -128).

În continuare sînt date formatele instrucțiunilor microprocesorului Z80. Instrucțiunile pe un octet conțin numai codul operației :

COD OPERAȚIE

Instrucțiunile pe doi octeți au patru formate :

COD OPERAȚIE
COD OPERAȚIE

COD OPERAȚIE
DATA

COD OPERAȚIE
COD ECHIPAMENT

COD OPERAȚIE
DEPLASARE

Instrucțiunile pe trei octeți au trei formate diferite

COD OPERAȚIE
DATA
DATA

COD OPERAȚIE
ADRESA OCT. INF.
ADRESA OCT. SUP.

COD OPERAȚIE
COD OPERAȚIE
DEPLASARE

Instrucțiunile pe patru octeți au următoarele formate :

COD OPERAȚIE
COD OPERAȚIE
DATA
DATA

COD OPERAȚIE
COD OPERAȚIE
ADRESA OCT. INF.
ADRESA OCT. SUP.

COD OPERAȚIE
COD OPERAȚIE
DEPLASARE
DATA

COD OPERAȚIE
COD OPERAȚIE
DEPLASARE
COD OPERAȚIE

Ultimele două tipuri de instrucțiuni, pe patru octeți sînt destul de complicate.

Moduri de adresare

Repertoriul de instrucțiuni al microprocesorului Z80 conține zece moduri de adresare a operanzilor, ceea ce îi conferă superioritate și flexibilitate sporite în raport cu microprocesorul 8080.

Adresare la registre

Operanzii se găsesc în registrele generale. Acestea se codifică cu oște trei biți și pot reprezenta registre sursă de operanzi și registru destinație, pentru rezultat.

Codificarea registrelor :

<i>Registru</i>	<i>Sursa sau Destinația SSS sau DDD</i>	<i>Registru</i>	<i>Sursa sau Destinația SSS sau DDD</i>
B	000	H	100
C	001	L	101
D	010	(H,L)	110
E	011	A	111

(HL) specifică faptul că sursa/destinația reprezintă o celulă de memorie a cărei adresă se găsește în perechea HL.

De exemplu instrucțiunea „Încarcă (LD) registrul destinație (DDD) cu conținutul registrului sursă (SSS)” are codul binar :

01DDDSSS

în care 01 — reprezintă codul operație.

În limbaj de asamblare instrucțiunile cu adresare la registru pot avea următoarele aspecte :

- LD A,C încarcă registrul B cu conținutul lui C,
- LD A,H încarcă acumulatorul cu conținutul lui H,
- LD (HL),A încarcă celula de memorie, a cărei adresă este dată în perechea de registre H,L cu conținutul lui A.

Adresare imediată

În acest caz instrucțiunea pe mai mulți octeți conține data asupra căreia se operează. De exemplu, instrucțiunea încarcă acumulatorul cu constanta hexazecimală 05H are doi octeți. Primul octet specifică codul operației, iar al doilea conține constanta 05H.

LD A, 05H

Exprimată în binar instrucțiunea va avea aspectul :

octet 1 : 00DDD110 ; DDD pentru A este 111

octet 2 : 00000101 ; reprezintă 05H

Adresare imediată extinsă

Instrucțiunea conține doi octeți de date, după codul operației, care vor fi folosiți în conjuncție cu o pereche de registre. Perechea de registre este codificată cu doi biți notați cu rp, în instrucțiune :

<i>Perechea de registre rp</i>	<i>Codul binar</i>	<i>Perechea de registre rp</i>	<i>Codul binar</i>
BC	00	HL	10
DE	01	SP	11

De exemplu o instrucțiune de încărcare a perechii de registre rp cu conținuturile octeților doi și trei din instrucțiune are aspectul următor :

LD rp , <B3> <B2>

unde <B3> <B2> specifică amplasarea octeților. Astfel, conținutul octetului B3 se va plasa în registrul mai puțin semnificativ al perechii, iar conținutul octetului B2 în registrul mai semnificativ. În cod mașină vom avea formatul :

octet 1	00rp0001
octet 2	<B2>
octet 3	<B3>

Adresare indirectă prin registre

Instrucțiunea folosește o pereche de registre pentru a indica adresa unei celule de memorie care conține un operand. Pentru a arăta că perechea de registre constituie un indicator pentru o celulă de memorie, numele perechii se plasează între paranteze :

LD A, (H, L) indică încărcarea lui A cu conținutul celulei de memorie specificată de perechea H, L.

În unele cazuri adresarea indirectă specifică doi octeți asupra cărora se efectuează o operație. De exemplu, încărcarea perechii BC cu conținutul primului octet din stivă specificat de (SP) și cu conținutul celui de-al doilea octet din stivă specificat de (SP+2) se exprimă astfel :

POP BC întâi se încarcă registrul C și apoi registrul B.

Adresare extinsă

O instrucțiune care utilizează adresarea extinsă conține în ultimii doi octeți o adresă de 16 biți. Această adresă se folosește ca indicator al unei celule de memorie care conține un operand sau ca adresă la care se face transferul programului printr-o instrucțiune de salt (JP). Cei doi octeți sînt specificați prin notația nn. Pentru o instrucțiune de încărcare a acumulatorului se va folosi notația :

LD (nn), A — unde (nn) are forma, de exemplu, (1310 H).

O instrucțiunea de transfer al comenzii va avea aspectul :

JP nn — unde nn are forma, de exemplu, 1310 H

Adresare prin pagina zero modificată

Z80 posedă opt instrucțiuni care folosesc acest mod de adresare pentru a face transferul comenzii programului la o anumită subrutină. Aceste instrucțiuni poartă numele de „restart” și au codul operației RST xxH, unde xx poate fi : 00, 08, 10, 18, 20, 28, 30, 38 în hexazecimal. Octetul cel mai puțin semnificativ al adresei de salt îl va constitui xx, în timp ce octetul mai semnificativ al adresei va fi 00H (pagina zero). Astfel, instrucțiunea :

RST 20H

va efectua transferul comenzii la adresa 0020H

Instrucțiunile RST xx au o lungime de un octet.

Adresare implicită

Unele instrucțiuni folosesc în mod implicit unul din registre. Astfel, instrucțiunile aritmetice și logice utilizează acumulatorul ca sursă de operand și ca destinație, pentru rezultat.

De exemplu, instrucțiunea :

ADD A,C

specifică adunarea conținutului acumulatorului A cu cel al registrului C și plasarea rezultatului în A.

Adresare la biți

O serie de instrucțiuni asigură adresarea la un bit specificat, într-un registru sau într-o celulă de memorie, pentru a-l poziționa în unu (SET) sau zero (RESET). Bitul va fi specificat cu un cod format din trei ranguri binare. În cadrul cuvântului de un octet, al cărui bit specificat se modifică, numărarea bitilor se face în sens crescător, de la bitul cel mai puțin semnificativ, aflat la dreapta, la bitul cel mai semnificativ, aflat la stânga. De exemplu, dacă registrul C va avea conținutul :

poziție bit 7 6 5 4 3 2 1 0

conținut C 1 0 1 0 1 0 0 0

după execuția instrucțiunii :

SET 4,C

conținutul său se va modifica astfel :

poziție bit 7 6 5 4 3 2 1 0

conținut C 1 0 1 1 1 0 0 0

Adresare indexată

Z80 posedă două registre index de câte 16 biți IX, IY ale căror conținuturi se adună cu conținutul octetului deplasare, pentru a forma o adresă de celulă de memorie în care se află un operand. Deplasarea constituie un octet aflat după codul de operație, din instrucțiune. Ea poate avea valori pozitive sau negative, fiind reprezentată ca un număr exprimat în complementul față de doi. De exemplu, instrucțiunile de încărcare a acumulatorului cu conținutul unei celule de memorie, a cărei adresă este cu 127 unități mai mare sau cu 128 unități mai mică decât cea specificată în IX are următorul aspect :

LD A,(IX+7FH) ; 7FH reprezintă 127 (10)

LD A,(IX+80H) ; 80H reprezintă -128 (10)

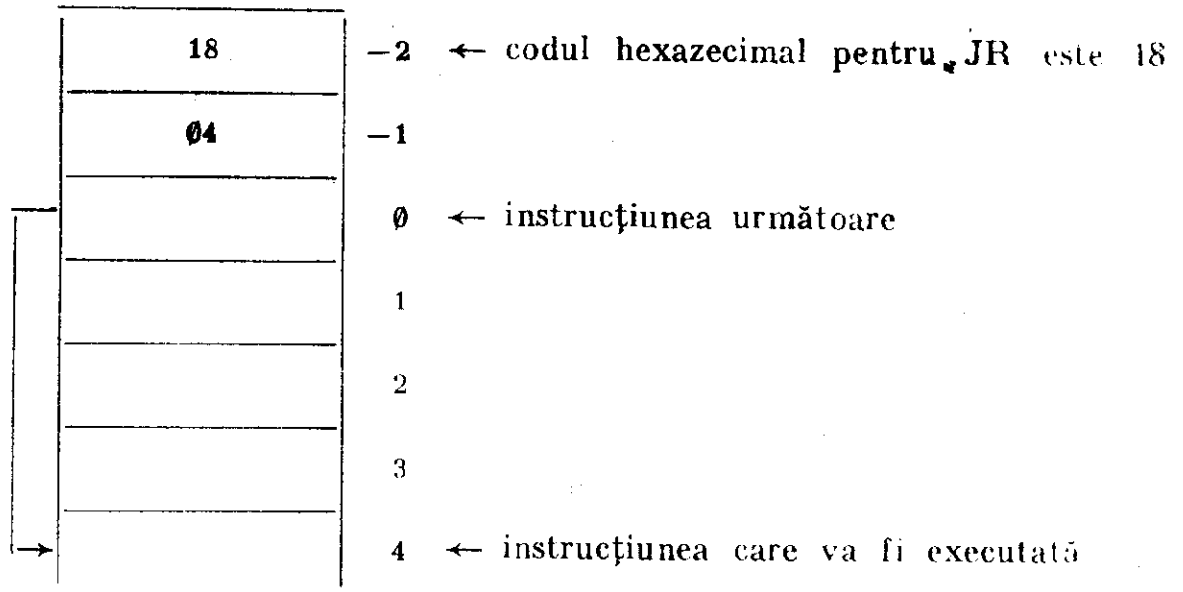
Adresarea indexată este extrem de utilă pentru accesul în tabelele de date, organizate în memorie. Registrele IX și IY se încarcă cu adresele de start ale tabelelor. Referirile în tabele se vor face relativ la aceste adrese.

Adresare relativă

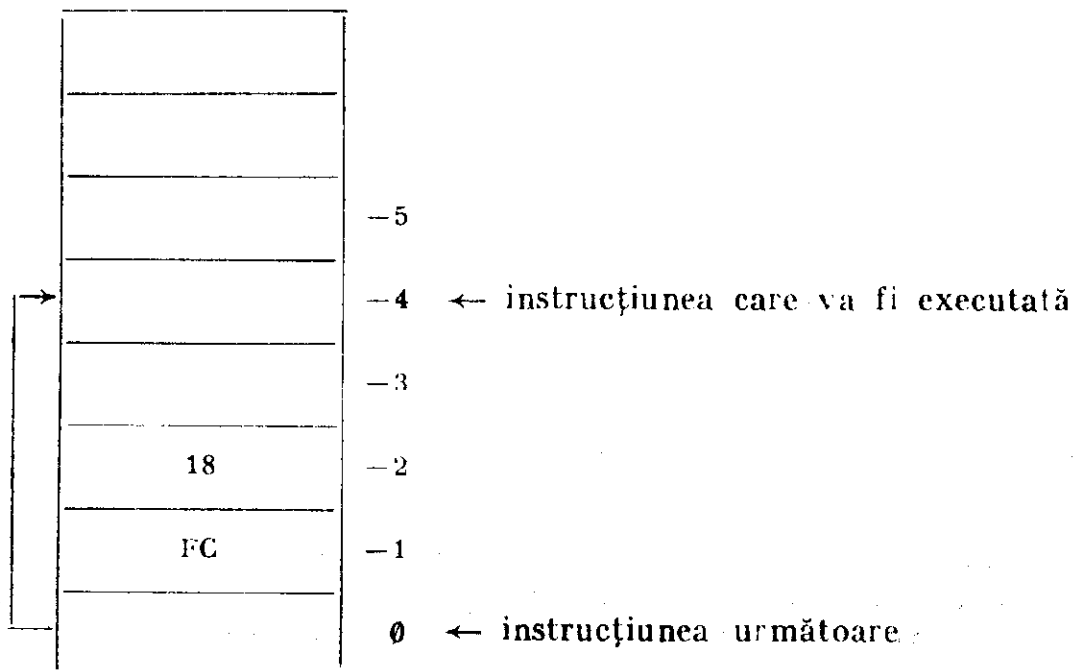
Aceasta reprezintă un mod specializat, care este folosit numai de instrucțiunile de transfer al comenzii numite transferuri relative ale comenzii (JR). Primul octet, după codul operației, în instrucțiune, reprezintă o deplasare pozitivă sau negativă față de o adresă, care este egală cu adresa instrucțiunii următoare din program. Adresarea relativă se face în limitele +127 și -128 față de adresa instrucțiunii care urmează după instrucțiunea de transfer al comenzii.

Exemple :

JR 04H are următorul efect :



JR FC H are următorul efect :



Modul relativ de adresare permite scrierea de programe relocabile, independente de locul de plasare în memorie.

Transferul relativ al comenzii necesită numai doi octeți de memorie, față de instrucțiunile de transfer absolut al comenzii, care necesită trei octeți.

Repertoriul de instrucțiuni

Instrucțiunile microprocesorului Z80 pot fi organizate în următoarele grupuri :

- instrucțiuni de încărcare pe 8 biți ;
- instrucțiuni de încărcare pe 16 biți ;
- instrucțiuni de schimb, transfer de blocuri și căutare ,
- instrucțiuni aritmetice și logice pe 8 biți ;
- instrucțiuni universale și de comandă a UCP ;
- instrucțiuni aritmetice pe 16 biți ;
- instrucțiuni de rotire și deplasare ;
- instrucțiuni de poziționare în unu, în zero și de testare la nivel de bit ;
- instrucțiuni de transfer al comenzii ;
- instrucțiuni de chemare și revenire din subrutină ;
- instrucțiuni de intrare/ieșire.

În continuare ele se prezintă într-o manieră sistematizată în tabelul 4.1, care conține o serie de informații :

- mnemonică,
- operația,
- indicatorii,
- codul de operație,
- numărul de octeți din instrucțiune,
- numărul de cicluri ale instrucțiunii,
- numărul de perioade ale instrucțiunii.

În cadrul tabelii* s-au folosit următoarele notații :

- ↑ — indicatorul este afectat conform rezultatului operației,
- — indicatorul nu este modificat de operație,
- 0 — indicatorul este forțat în zero,
- 1 — indicatorul este forțat în unu,
- X — indicatorul este indiferent,
- V — indicatorul P/V este poziționat în conformitate cu depășirea rezultatului operației,
- P — indicatorul P/V este poziționat în conformitate cu paritatea rezultatului,
- r — unul din registrele UCP : A,B,C,D,E,H,L
- s — o locație de 8 biți pentru toate modurile de adresare permise de acea instrucțiune,
- ss — o locație de 16 biți pentru toate modurile de adresare permise de acea instrucțiune,
- II — unul din registrele index X,IY,
- R — conținutul de reimprespățare,
- n — un octet cu valoarea cuprinsă în gama 0-255
- nn — doi octeți cu valoarea cuprinsă în gama 0-65535.

Tabelul 4.1.1.

GRUPUL INSTRUCȚIUNILOR DE ÎNCARCARE PE 8 BIȚI

Mnemonică	Operația	Indicatorii								Cod Op			Nr. octeții	Nr. cicluri M	Nr. perioade T	Comentarii	
		S	Z	H	P/V	N	C	76	543	210	Hex						
LDr, s	$r \leftarrow s$	•	•	X	•	X	•	•	•	01	r	s		1	1	4	r, s Reg
LDr, n	$r \leftarrow n$	•	•	X	•	X	•	•	•	00	r	100		2	2	7	000 B
LDr, (HL)	$r \leftarrow (HL)$	•	•	X	•	X	•	•	•	01	r	110		1	2		001 C
LDr, (IX+d)	$r \leftarrow (IX+d)$	•	•	X	•	X	•	•	•	11	011	101	DD	3	5	19	010 D
										01	r	110					011 E
										-	d	-					100 H
LDr, (IY+d)	$r \leftarrow (IY+d)$	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19	101 L
										01	r	110					111 A
										-	d	-					
LD(HL), r	$(HL) \leftarrow r$	•	•	X	•	X	•	•	•	01	110	r		1	2	7	
LD(IX+d), r	$(IX+d) \leftarrow r$	•	•	X	•	X	•	•	•	11	011	101	DD	3	5	19	
										01	110	r					
										-	d	-					
LD(IY+d), r	$(IY+d) \leftarrow r$	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19	
										01	110	r					
										-	d	-					
LD(HL), n	$(HL) \leftarrow n$	•	•	X	•	X	•	•	•	00	110	110	36	2	3	10	
LD(IX+d), n	$(IX+d) \leftarrow n$	•	•	X	•	X	•	•	•	11	011	101	DD	4	5	19	
										00	110	110	36				
										-	d	-					
										-	n	-					
LD(IY+d), n	$(IY+d) \leftarrow n$	•	•	X	•	X	•	•	•	11	111	101	FD	4	5	19	
										00	110	110	36				
										-	d	-					
										-	n	-					
LDA, (BC)	$A \leftarrow (BC)$	•	•	X	•	X	•	•	•	00	001	010	DA	1	2	7	
LDA, (DE)	$A \leftarrow (DE)$	•	•	X	•	X	•	•	•	00	011	010	1A	1	2	7	
LDA, (nn)	$A \leftarrow (nn)$	•	•	X	•	X	•	•	•	00	111	010	3A	3	4	13	
										-	n	-					
										-	n	-					
LD(BC), A	$(BC) \leftarrow A$	•	•	X	•	X	•	•	•	00	000	010	02	1	2	7	
LD(DE), A	$(DE) \leftarrow A$	•	•	X	•	X	•	•	•	00	010	010	12	1	2	7	
LD(nn), A	$(nn) \leftarrow A$	•	•	X	•	X	•	•	•	00	110	010	32	3	4	13	
										-	n	-					
										-	n	-					
LD A, I	$A \leftarrow I$	•	•	X	0	X	IFF	0	•	11	101	101	ED	2	2	9	
										01	010	111	57				
LD A, R	$A \leftarrow R$	•	•	X	0	X	IFF	0	•	11	101	101	ED		2	9	
										01	011	111	5F				
LD I, A	$I \leftarrow A$	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	9	
										01	000	111	47				
LD R, A	$R \leftarrow A$	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	9	
										01	001	111	4F				

Notă : r, s oricare dintre registrele A, B, C, D, E, H, L

(IFF) este copiat în indicatorul P/V

• = indicator neafectat, 0 = indicator zero, X = indicator necunoscut

• = indicator afectat conform rezultatului operației

Tabelul 4.1.2.

GRUPUL INSTRUCȚIUNILOR DE ÎNCĂRCARE PE 16 BIȚI

Mnemonică	Operația	Indicatorii								Cod Op.		Nr. octeții	Nr. cicluri M	Nr. perioade T	Comentarii			
		S	Z	H	P/V	N	C	76	543	210	Hex							
LD dd,nn	dd ← nn	•	•	X	•	X	•	•	•	•	00	dd	001	3	3	10	dd Perechea 00 BC 01 DE 10 HL 11 SP	
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	•	11	011	101	DD	4	4	14	
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	•	11	111	101	FD	4	4	14	
LD HL(nn)	H ← (nn+1) L ← (nn)	•	•	X	•	X	•	•	•	•	00	101	010	2A	3	5	16	
LD dd,(nn)	dd _H ← (nn+1) dd _L ← (nn)	•	•	X	•	X	•	•	•	•	11	101	101	ED	4	6	20	
LD IX,(nn)	IX _H ← (nn+1) IX _L ← (nn)	•	•	X	•	X	•	•	•	•	11	011	101	DD	4	6	20	
LD IY,(nn)	IY _H ← (nn+1) IY _L ← (nn)	•	•	X	•	X	•	•	•	•	11	111	101	FD	4	6	20	
LD (nn)HL	(nn+1) ← H (nn) ← L	•	•	X	•	X	•	•	•	•	00	100	010	22	3	5	16	
LD (nn)dd	(nn+1) ← dd _H (nn) ← dd _L	•	•	X	•	X	•	•	•	•	11	101	101	ED	4	6	20	
LD (nn)IX	(nn+1) ← IX _H (nn) ← IX _L	•	•	X	•	X	•	•	•	•	11	011	101	DD	4	6	20	
LD (nn)IY	(nn+1) ← IY _H (nn) ← IY _L	•	•	X	•	X	•	•	•	•	11	111	101	FD	4	6	20	
LD SP,HL	SP ← HL	•	•	X	•	X	•	•	•	•	11	111	001	F9	1	1	6	
LD SP,IX	SP ← IX	•	•	X	•	X	•	•	•	•	11	011	101	DD	2	2	10	
LD SP,IY	SP ← IY	•	•	X	•	X	•	•	•	•	11	111	001	F9	2	2	10	
PUSH qq	(SP-2) ← qq _L (SP-1) ← qq _H	•	•	X	•	X	•	•	•	•	11	qq	101	F9	1	3	11	qq Perechea 00 BC 01 DE 10 HL 11 AF
PUSH IX	(SP-2) ← IX _L (SP-1) ← IX _H	•	•	X	•	X	•	•	•	•	11	011	101	DD	2	4	15	
PUSH IY	(SP-2) ← IY _L (SP-1) ← IY _H	•	•	X	•	X	•	•	•	•	11	111	101	FD	2	4	15	
POP qq	qq _H ← (SP+1) qq _L ← (SP)	•	•	X	•	X	•	•	•	•	11	qq	001	F9	1	3	10	
POP IX	IX _H ← (SP+1) IX _L ← (SP)	•	•	X	•	X	•	•	•	•	11	011	101	DD	2	4	14	
POP IY	IY _H ← (SP+1) IY _L ← (SP)	•	•	X	•	X	•	•	•	•	11	111	101	FD	2	4	14	

Notă: dd oricare dintre registrele : BC, DE, HL, SP
 qq oricare dintre registrele ; AF, BC, DE, HL
 (PER)_H, (PER)_L se referă la octeții superior (H) și inferior (L)
 ai perechii de registre

Tabelul 4.1.3.

GRUPUL INSTRUCȚIUNILOR DE SCHIMB, TRANSFER DE BLOCURI ȘI CĂUTARE

Mnemonică	Operatie	Indicatorii								Cod.Op.				Nr. octeți	Nr. cicluri M	Nr. perioade T	Comentarii		
		S	Z	H	P	V	N	C	76	543210	Hex								
EX DE, HL	DE → HL	•	•	X	•	X	•	•	•	•	•	11	101	011	EB	1	1	4	Schimb de tablou de registre (principal și auxiliar)
EX AF, AF	AF → AF	•	•	X	•	X	•	•	•	•	•	00	001	000	D8	1	1	4	
EXX	(BC → BC) (DE → DE) (HL → HL)	•	•	X	•	X	•	•	•	•	•	11	011	001	D9	1	1	4	
LX(SP), HL	H ← (SP) · 1 L ← (SP)	•	•	X	•	X	•	•	•	•	•	11	100	011	E3	1	5	19	
EX(SP), IX	IX ← (SP) · 1	•	•	X	•	X	•	•	•	•	•	11	011	101	DD	2	6	23	
EX(SP), IY	IY ← (SP) · 1 IV ← (SP)	•	•	X	•	X	•	•	•	•	•	11	100	011	E3	2	6	23	
LDI	(DE) ← (HL)	•	•	X	0	X	1	0	•	•	•	11	101	101	ED	2	4	16	Încarcă (HL) în (DE) incrementează indicatorii și decrementează contorul de octeți (BC)
	DE → DE - 1	•	•	X	0	X	0	0	•	•	•	11	101	101	ED	2	4	16	
	HL → HL - 1	•	•	X	0	X	0	0	•	•	•	10	100	000	AD	2	4	16	
	BC → BC - 1	•	•	X	0	X	0	0	•	•	•	10	100	000	AD	2	4	16	
LDIR	(DE) ← (HL)	•	•	X	0	X	0	0	•	•	•	11	101	101	ED	2	5	21	Dacă BC ≠ 0 Dacă BC = 0
	DE → DE - 1 HL → HL - 1 BC → BC - 1	•	•	X	0	X	0	0	•	•	•	10	110	000	BD	2	4	16	
LDD	Repetă până când BC = 0																		
	(DE) ← (HL)	•	•	X	0	X	1	0	•	•	•	11	101	101	ED	2	4	16	
LDDR	DE → DE - 1	•	•	X	0	X	0	0	•	•	•	11	101	101	ED	2	5	21	Dacă BC ≠ 0 Dacă BC = 0
	HL → HL - 1 BC → BC - 1	•	•	X	0	X	0	0	•	•	•	10	111	000	B8	2	4	18	
CPIR	Repetă până când BC = 0																		Dacă BC ≠ 0 și A ≠ (HL) Dacă BC = 0 și A = (HL)
	A ← (HL)	1	1	X	1	X	1	1	•	•	•	11	101	101	ED	2	4	16	
CPD	HL → HL - 1	•	•	X	1	X	1	1	•	•	•	11	101	101	ED	2	5	21	Dacă BC ≠ 0 și A ≠ (HL) Dacă BC = 0 și A = (HL)
	BC → BC - 1	•	•	X	1	X	1	1	•	•	•	10	110	001	B1	2	4	16	
CPDR	Repetă până când A = (HL) și BC = 0																		
	A ← (HL)	1	1	X	1	X	1	1	•	•	•	11	101	101	ED	2	4	16	
CPDR	HL → HL - 1	•	•	X	1	X	1	1	•	•	•	11	101	101	ED	2	5	21	Dacă BC ≠ 0 și A ≠ (HL) Dacă BC = 0 și A = (HL)
	BC → BC	•	•	X	1	X	1	1	•	•	•	10	111	001	B9	2	4	16	

Notă: ① Indicatorul P/V este zero dacă BC - 1 = 0, altfel P/V = 1
 ② Indicatorul Z este unu dacă A = (HL), altfel Z = 0

Tabelul 4.1.4.

GRUPUL INSTRUCȚIUNILOR ARITMETICE ȘI LOGICE PE 8 BIȚI

Mnemonică	Operatie	Indicatorii								Cod Op				Nr. octeții	Nr. cicluri M	Nr. perioade T	Comentarii
		S	Z	H	P	V	N	C	7	6	5	4	3				
ADD A,r	A ← A + r			X		X	V	0		10	000	r	Hex	1	1	4	r Reg
ADD A,n	A ← A + n			X		X	V	0		11	000	110		2	2	7	000 B 001 C 010 D 011 E
ADD A,(HL)	A ← A + (HL)			X		X	V	0		10	000	110		1	2	7	100 H 101 L 111 A
ADD A,(IX+d)	A ← A + (IX+d)			X		X	V	0		11	011	101	DD	3	5	19	
ADD A,(IY+d)	A ← A + (IY+d)			X		X	V	0		11	111	101	FD	3	5	19	
ADC A,s	A ← A + s + CY			X		X	V	0		00	000	100					s este oricare r,n (HL),(IX+d),(IY+d) ca în instrucțiunea ADD
SUB s	A ← A - s			X		X	V	1		01	000	100					Biții indicați înlocuiesc 000 în instr ADD de mai sus
SBC A,s	A ← A - s - CY			X		X	V	1		01	011	100					
AND s	A ← A ∧ s			X		1	X	P	0	0	0	0					
OR s	A ← A ∨ s			X		0	X	P	0	0	0	0					
XOR s	A ← A ⊕ s			X		0	X	P	0	0	0	0					
CP s	A ← s			X		X	V	1		11	111	101					
INC r	r ← r + 1			X		X	V	0	•	00	r	100		1	1	4	
INC (HL)	(HL) ← (HL) + 1			X		X	V	0	•	00	110	100		1	3	11	
INC (IX+d)	(IX+d) ← (IX+d) + 1			X		X	V	0	•	11	011	101	DD	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d) + 1			X		X	V	0	•	11	111	101	FD	3	6	23	
DEC s	s ← s - 1			X		X	V	1	•	-	d	-					s este oricare r,(HL),(IX+d),(IY+d) ca la INC DEC are același format și stări ca INC înlocuiește 100 cu 101 în cod de Op

Nota . Simbolul V în coloana indicatorului P/V specifică prezența depășirii
 Simbolul P în coloana indicatorului P/V specifică prezența parității
 V=1 înseamnă depășire, V=0 înseamnă lipsa depășirii
 P=1 înseamnă paritate pară a rezultatului, P=0 înseamnă paritate impară a rezultatului.

Tabelul 4.1.5

GRUPUL INSTRUCȚIUNILOR UNIVERSALE ȘI DE COMANDĂ A UCP

Mnemonică	Operație	Indicatorii							Cod Op		Nr octeții	Nr cicluri M	Nr perioade T	Comentarii	
		S	Z	H	P/V	N	C	76543210	Hex						
DAA	Converteste conținutul ac în BCD împachetat după adunare sau scădere a numerelor BCD împachetate	↓	↓	X	↓	X	P	-	1	00 100 111	27	1	1	4	Ajustare zecimală a acumulatorului
CPL	$A \leftarrow \bar{A}$	•	•	X	1	X	•	1	•	00 101 111	2F	1	1	4	Complementează ac (Compl față de 1)
NEG	$A \leftarrow \bar{A} + 1$	↓	↓	X	↓	X	V	1	↓	11 101 101 01 000 100	ED 44	2	2	8	Neagă ac (Compl față de 2)
CCF	$CY \leftarrow \bar{CY}$	•	•	X	X	X	•	0	↓	00 111 111	3F	1	1	4	Compl ind de transport
SCF	$CY \leftarrow 1$	•	•	X	0	X	•	0	↓	00 110 111	37	1	1	4	Indic de transp
NOP	Neoperational	•	•	X	•	X	•	•	•	00 000 00	00	1	1	4	1
HALT	UCP stop	•	•	X	•	X	•	•	•	01 110 110	76	1	1	4	
DI	$IFF \leftarrow 0$	•	•	X	•	X	•	•	•	11 110 011	F3	1	1	4	
EI	$IFF \leftarrow 1$	•	•	X	•	X	•	•	•	11 111 011	F3	1	1	4	
IM 0	Stăp mod intr 0	•	•	X	•	X	•	•	•	11 101 101	ED	2	2	8	
IM 1	Stăp mod intr 1	•	•	X	•	X	•	•	•	01 000 110 11 101 101	46 ED	2	2	8	
IM 2	Stăp mod intr 2	•	•	X	•	X	•	•	•	01 010 110 11 101 101 01 011 110	56 ED 5E	2	2	8	

Nota IFF specifică bistabilul de activare a intreruperilor
CY specifică histabilul de transport
Intreruperile nu sînt testate la sfîrșitul instrucțiunilor
DI și EI

Tabelul 4.1.6.

GRUPUL INSTRUCȚIUNILOR ARITMETICE PE 16 BITI

Mnemonică	Operație	Indicatorii							Cod Op.		Nr octeții	Nr cicluri M	Nr perioade T	Comentarii	
		S	Z	H	P/V	N	C	76543210	Hex						
ADD HL, ss	$HL \leftarrow HL + ss$	•	•	X	X	X	•	0	↓	00 ss1 001		1	3	11	ss Reg
ADC HL, ss	$HL \leftarrow HL + ss + CY$	↓	↓	X	X	X	V	0	↓	11 101 101 01 ss1 010	ED	2	4	15	01 DE 10 HL 11 SP
SBC HL, ss	$HL \leftarrow HL - ss - CY$	↓	↓	X	X	X	V	1	↓	11 101 101 01 ss0 010	ED	2	4	15	
ADD IX, pp	$IX \leftarrow IX + pp$	•	•	X	X	X	•	0	↓	01 011 101 00 pp1 001	DD	2	4	15	pp Reg 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	$IY \leftarrow IY + rr$	•	•	X	X	X	•	0	↓	11 111 101 00 rr1 001	FD	2	4	15	rr Reg 00 BC 01 DE 10 IY 11 SP
INC ss	$ss \leftarrow ss + 1$	•	•	X	•	X	•	•	•	00 ss0 011		1	1	6	
INC IX	$IX \leftarrow IX + 1$	•	•	X	•	X	•	•	•	01 011 101 00 100 011	DD 23	2	2	10	
INC IY	$IY \leftarrow IY + 1$	•	•	X	•	X	•	•	•	11 111 101 00 100 011	FD 23	2	2	10	
DEC ss	$ss \leftarrow ss - 1$	•	•	X	•	X	•	•	•	00 ss1 011		1	1	6	
DEC IX	$IX \leftarrow IX - 1$	•	•	X	•	X	•	•	•	11 011 101	DD	2	2	10	
DEC IY	$IY \leftarrow IY - 1$	•	•	X	•	X	•	•	•	00 101 011 11 111 101 00 101 011	2B FD 2B	2	2	10	

Notă: ss oricare din perechile BC, DE, HC, SP
pp oricare din perechile de registre BC, DE, IX, SP
rr oricare din perechile de registre BC, DE, IY, SP

Wabelul 4.1.7.

GRUPUL INSTRUCȚIUNILOR DE ROTIRE ȘI DEPLASARE

Mnemonica	Operația	Indicatorii							Cod.Op.			Nr. octeți	Nr. cicluri M	Nr. perioade T	Comentarii		
		S	Z	H	V	N	C	76	543	210	Hex						
RLCA		•	•	X	0	X	•	0	1	00	000	111	07	1	1	4	Roteste stînga circular acumulatorul
RLA		•	•	X	0	X	•	0	1	00	010	111	17	1	1	4	Roteste acumulatorul stînga
RRCA		•	•	X	0	X	•	0	1	00	001	111	0F	1	1	4	Roteste dreapta circular acumulatorul
RRA		•	•	X	0	X	•	0	1	00	011	111	1F	1	1	4	Roteste dreapta acumulatorul
RLC		↑	↑	X	0	X	P	0	1	11	001	011	CB	2	2	8	Roteste stînga circular registrul r
RLC(HL)		↑	↑	X	0	X	P	0	1	11	001	011	CB	2	4	15	Reg 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC(IX+d)		↑	↑	X	0	X	P	0	1	11	011	101	DD	4	6	23	
		↑	↑	X	0	X	P	0	1	11	001	011	CB				
		↑	↑	X	0	X	P	0	1	00	000	110					
RLC(IY+d)	↑	↑	X	0	X	P	0	1	11	111	101	FD	4	6	23		
		↑	↑	X	0	X	P	0	1	11	001	011	CB				
		↑	↑	X	0	X	P	0	1	00	000	110					
RLs		↑	↑	X	0	X	P	0	1	0	10						Formatul și stările sînt arătate pentru instr. RLC. Pentru a forma un cod de Op se înlocuiește 000 cu codul arătat
RRCs		↑	↑	X	0	X	P	0	1	00							
RRs		↑	↑	X	0	X	P	0	1	01							
SLAs		↑	↑	X	0	X	P	0	1	10							
SRA s		↑	↑	X	0	X	P	0	1	10							
SRLs		↑	↑	X	0	X	P	0	1	11							
RLD		↑	↑	X	0	X	P	0	•	11	101	101	ED	2	5	18	Roteste cifra la stînga și la dreapta între acumulator și locație (HL)
										01	101	111	6F				
RRD		↑	↑	X	0	X	P	0	•	11	101	101	ED	2	5	18	Continutul jumătății superioare a acumulatorului este neafectată.
										01	100	111	67				

Tabloul 4.1.8

GRUPUL INSTRUCIUNILOR DE POZITIONARE ÎN UNU
ÎN ZERO ȘI TESTARE LA NIVEL DE BIT

Mnemonică	Operația	Indicatorii								Cod Op.			Nr. octeți	Nr. cicluri M	Nr. perioade T	Comentarii		
		S	Z	H	P/V	N	C	76	543	210	Hex	Comentarii				Reg		
BIT b,r	$Z \leftarrow \overline{r}_b$	X	X	1	X	X	0	•	11	001	011	CB	2	2	8	r	Reg	
BIT b,(HL)	$Z \leftarrow \overline{(HL)}_b$	X	X	1	X	X	0	•	01	b	r	CB	2	3	12	000	B	
BIT b,(IX+d)	$Z \leftarrow \overline{(IX+d)}_b$	X	X	1	X	X	0	•	01	b	110	DD	4	5	20	010	D	
									11	001	011					011	100	H
BIT (IY+d)	$Z \leftarrow \overline{(IY+d)}_b$	X	X	1	X	X	0	•	—	d	—	FD	4	5	20	101	L	
									01	b	110					111	A	
RES b,s	$s \leftarrow 0$ $s \equiv r, (HL), (IX+d), (IY+d)$	X	X	1	X	X	0	•	11	111	101	FD	4	6	23	b	Bit Testat	
									11	001	011					000	0	
SET b,r	$r \leftarrow 1$	•	•	X	•	X	•	•	•	11	001	011	CB	2	2	8	001	1
																	010	2
SET b,(HL)	$(HL) \leftarrow 1$	•	•	X	•	X	•	•	•	11	b	r	CB	2	4	15	011	3
																	110	4
SET b,(IX+d)	$(IX+d) \leftarrow 1$	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	23	110	5
																	110	6
SET b,(IY+d)	$(IY+d) \leftarrow 1$	•	•	X	•	X	•	•	•	11	b	110	FD	4	6	23	111	7
																	110	7
RES b,s	$s \leftarrow 0$ $s \equiv r, (HL), (IX+d), (IY+d)$	•	•	X	•	X	•	•	•	10	b	110	CB	4	6	23	111	7
																	110	7

Pentru a forma un nou cod de op. se înlocuiește $\boxed{11}$ de la SET b,s cu $\boxed{10}$ indicatorii și perioadele sînt identice cu cele pentru instrucionarea SET

Tabelul 4.1.9

GRUPUL INSTRUCȚIUNILOR DE TRANSFER AL COMENZII

Mnemonică	Operația	Indicatorii								Cod Op				Nr octeți	Nr cicluri M	Nr perioade T	Comentarii
		S	Z		H	P/V	N	C	76	543	210						
JP nn	PC ← nn	•	•	X	•	X	•	•	•	11 000 011	C3	3	3	10			
										- n -							
										- n -							
JF cc, n	Dacă cond cc este adevărată PC ← nn altfel continuă	•	•	X	•	X	•	•	•	11 cc 010		3	3	10	cc - Condiția 000 NZ nu este zero 001 Z zero 010 NC transport lipsă 011 transport 100 paritate impară 101 paritate pară 110 semn pozitiv 111 semn negativ		
										- n -							
										- n -							
JR	PC ← PC + e	•	•	X	•	X	•	•	•	00 011 000	18	2	3	12			
										- e-2 -							
JR C, e	C=0, continuă C=1, PC ← PC + e	•	•	X	•	X	•	•	•	00 111 000	38	2	2	7	Pt. cond. neindeplinită		
										- e-2 -		2	3	12	Pt. cond. neindeplinită		
JR NC, e	C=1, continuă C=0, PC ← PC + e	•	•	X	•	X	•	•	•	00 110 000	30	2	2	7	Pt. cond. neindeplinită		
										- e-2 -		2	3	12	Pt. cond. neindeplinită		
JR Z, e	Z=0, continuă Z=1, PC ← PC + e	•	•	X	•	X	•	•	•	00 101 000	28	2	2	7	Pt. cond. neindeplinită		
										- e-2 -		2	3	12	Pt. cond. neindeplinită		
JR NZ, e	Z=1, continuă Z=0, PC ← PC + e	•	•	X	•	X	•	•	•	00 100 000	20	2	2	7	Pt. cond. neindeplinită		
										- e-2 -		2	3	12	Pt. cond. neindeplinită		
JR(HL)	PC ← HL	•	•	X	•	X	•	•	•	11 101 001	E9	1	1	4			
JR(IX)	PC ← IX	•	•	X	•	X	•	•	•	11 011 101	DD	2	2	8			
										11 101 001	E9						
JR(IV)	PC ← IV	•	•	X	•	X	•	•	•	11 111 101	FD	2	2	8			
										11 101 001	E9						
DJNZ, e	B ← B-1 B=0, continuă B ≠ 0 PC ← PC + e	•	•	X	•	X	•	•	•	00 010 000	10	2	2	8	B=0		
										- e-2 -		2	3	13	B ≠ 0		

Nota e reprezintă extensia în modul de adresare relativă e este un număr cu semn în complementul față de doi în gama <-128 127> e-2 în codul de operație asigură o adresă efectivă PC + e, deoarece PC este incrementat cu 2 înaintea adunării cu e

Tabelul 4.1.10.

GRUPUL INSTRUCȚIUNILOR DE CHEMARE ȘI REVENIRE DIN SUBRUTINĂ																	
Mnemonică	Operație	Indicatorii							Cod Op			Nr. octeți	Nr. cicluri M	Nr. stații T	Comentarii		
		S	Z	H	P	V	N	C	76	543	210					Hex	
CALL nn	(SP 1) ← PC _L (SP 2) ← PC _L PC ← nn	•	•	X	•	X	•	•	•	11	001	101	CD	3	5	17	
CALL cc, nn	Dacă cond. cc este falsă, continuă altfel la fel ca la CALL nn	•	•	X	•	X	•	•	•	11	cc	100		3	3	10	Dacă cc este fals
										-	n	-					
RET	PC _L ← (SP) PC _H ← (SP+1)	•	•	X	•	X	•	•	•	11	001	101	C9	1	3	10	
RET cc	Dacă cond. cc este falsă, continua, altfel la fel ca la RET	•	•	X	•	X	•	•	•	11	cc	000		1	1	5	Dacă cc este fals
RETI	Revenire din intrerupere	•	•	X	•	X	•	•	•	11	101	101	ED	2	4	14	000 NZ nu este zero
										01	001	101					
RETN1	Revenire din intrerupere nemascabilă	•	•	X	•	X	•	•	•	11	101	101	ED	2	4	14	010 NC transport lipsă
										01	000	101					011 C transport
RST p	(SP 1) ← PC _H (SP 2) ← PC _L PC _H ← 0 PC _L ← p	•	•	X	•	X	•	•	•	11	t	111		1	3	11	100 PO prioritate impară
																	101 PE prioritate pară
																	110 P semn pozitiv
																	111 M semn negativ
																	t
																	p
																	000 00H
																	001 08H
																	010 10H
																	011 18H
																	100 20H

Notă : RETN realizează încărcarea IFF₂ ← IFF₁

Tabelul 4.1.11.

GRUPUL INSTRUCȚIUNILOR DE INTRARE/IEȘIRE

Mnemonică	Operație	Indicatorii							Cod Op.		Nr. octeți	Nr. cicluri M	Nr. perioade T	Comentarii	
		S	Z	H	P/V	N	C	76	543210	Hex					
IN A ₇ (n)	A ← (n)	•	•	X	•	X	•	•	•	11 011 011	DB	2	3	11	n în A ₀ ~A ₇ Acc în A ₈ ~A ₁₅
IN r.(C)	r ← (C) Dacă r=110 va fi afectat numai indi catorul	↑	↓	X	↑	X	P	0	•	11 101 101 01 r 000	ED	2	3	12	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅
INI	(HL) ← (C) B ← B-1 HL ← HL-1	X	↑	X	X	X	X	1	X	11 101 101 10 100 010	ED A	2	4	16	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅
INI R	(HL) ← (C) B ← B-1 HL ← HL-1 Repetă până când B=0	X	1	X	X	X	X	1	X	11 101 101 10 110 010	ED B2	2	5 (pt B≠0) 4 (pt B=0)	21 16	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅
IND	(HL) ← (C) B ← B-1 HL ← HL-1	X	↑	X	X	X	X	1	X	11 101 101 10 101 010	ED AA	2	4	16	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅
INDR	(HL) ← (C) B ← B-1 HL ← HL-1 Repetă până când B=0	X	1	X	X	X	X	1	X	11 101 101 10 111 010	ED BA	2	5 (pt B≠0) 4 (pt B=0)	21 16	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅
OUT(n),A	(n) ← A	•	•	X	•	X	•	•	•	11 010 011	D3	2	3	11	n în A ₀ ~A ₇ Acc în A ₈ ~A ₁₅
OUT(C)	(C) ← r	•	•	X	•	X	•	•	•	11 101 101 01 r 000	ED	2	3	12	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅
OUTI	(C) ← (HL) B ← B-1 HL ← HL-1	X	↑	X	X	X	X	1	X	11 101 101 10 100 011	ED A3	2	4	16	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅
OTIR	(C) ← (HL) B ← B-1 HL ← HL+1 Repetă până când B=0	X	↑	X	X	X	X	1	X	11 101 101 10 110 011	ED B	2	5 (pt B≠0) 4 (pt B=0)	21 16	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅
OUTD	(C) ← (HL) B ← B-1 HL ← HL-1	X	↑	X	X	X	X	1	X	11 101 101 10 101 011	ED AB	2	4	16	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅
OTDR	(C) ← (HL) B ← B-1 HL ← HL-1 Repetă până când B=0	X	↑	X	X	X	X	1	X	11 101 101 10 111 011	ED BB	2	5 (pt B≠0) 4 (pt B=0)	21 16	C în A ₀ ~A ₇ B în A ₈ ~A ₁₅

Nota: ① Dacă rezultatul lui B=1 este zero indicatorul Z este poziționat în unu, în caz contrar este poziționat în zero

4.8. Interfața paralelă programabilă PIO

Interfața paralelă de I/E(PIO) este destinată cuplării microsistemelor cu echipamentele periferice de tip paralel: imprimante, perforatoare de bandă, tastaturi, etc. PIO este prevăzută cu două porturi paralele de câte opt biți și cu o unitate de comandă corespunzătoare.

Din punct de vedere constructiv ea este realizată în tehnologia NMOS, pe o pastilă cu 40 terminale, necesitând o singură sursă de alimentare (+5V) și cu semnal monofazic de ceas \emptyset , furnizat de către microprocesor.

Pentru manipularea perifericelor rapide se asigură un dialog prin întreruperi.

Cele două porturi de intrare/ieșire, notate cu A și B, pot fi programate ca porturi de intrare sau ca porturi de ieșire, la nivel de octet sau de bit. Portul A poate fi programat pentru a lucra bidirecțional. În funcție de indicatorii de stare ai echipamentelor periferice, se pot genera întreruperi programabile.

Pentru a simplifica logica externă de întreruperi, interfața are posibilitatea înlănțuirii facilităților oferite de circuitele de întrerupere prioritară, în vederea generării automate a vectorului corespunzător de întrerupere.

Schema bloc a interfeței programabile este dată în figura 4.18. Ea constă din: interfața cu magistrala unității centrale de prelucrare (UCP), logica internă de comandă, logica portului A, logica portului B și logica de comandă a întreruperilor. În general portul A poate fi folosit pentru transfer de date (bidirecțional), iar portul B pentru comenzi și controlul stărilor.

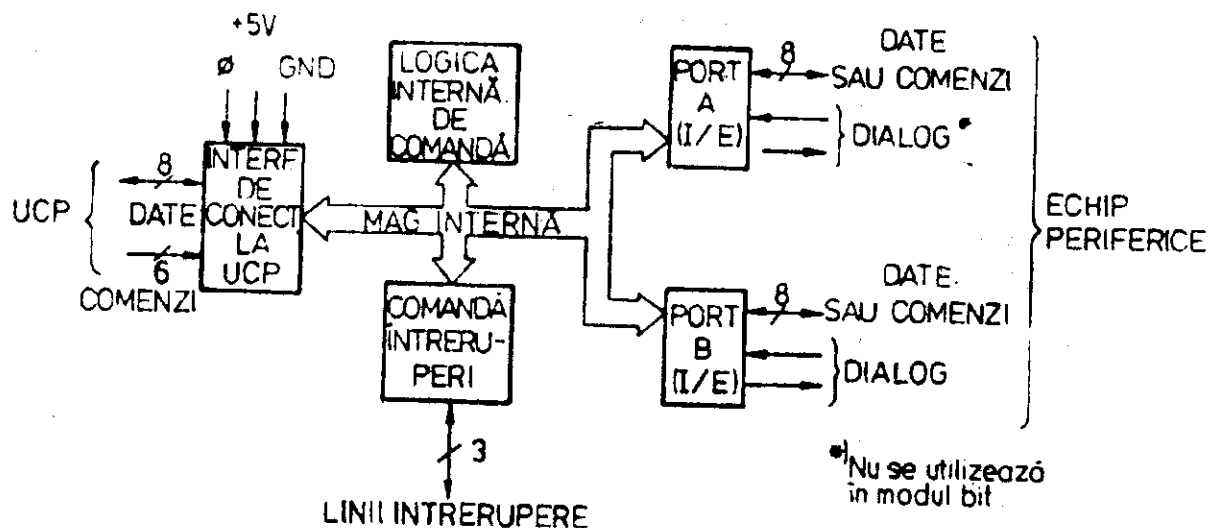


Fig. 4.18. Schema bloc a interfeței PIO.

În figura 4.19 se prezintă schema bloc a unui port de I/E. Ea constă dintr-un set de șase registre și logica de comandă a dialogului. Sînt prezente următoarele registre: registrul de intrare (8 biți), registrul de ieșire (8 biți), registrul de comandă a modului (2 biți), registrul mască (8 biți), registrul de selecție pentru I/E (8 biți) și registrul de comandă a măștii (2 biți). Ultimele

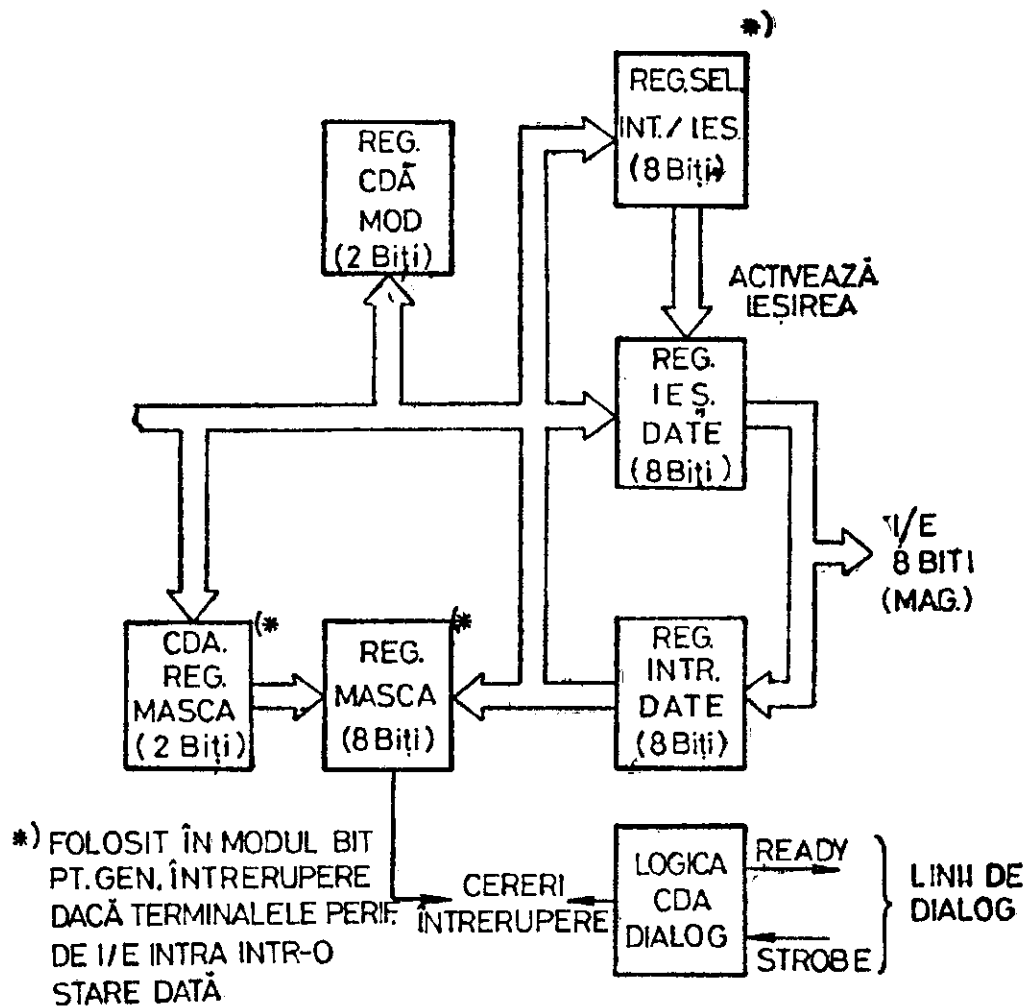


Fig. 4.19. Schema bloc a unui port de I/E.

trei registre se folosesc numai în cazul cînd portul a fost programat pentru a opera în modul bit.

Descrierea registrelor

Registrul de comandă a modului (2 biți) este încărcat de către UCP pentru a selecta unul din modurile de operare: intrare octet, ieșire octet, modul bidirecțional-octet, modul bit.

Registrul de ieșire a datelor (8 biți) asigură transferul datelor de la UCP, la echipamentul periferic.

Registrul de intrare a datelor (8 biți) primește datele de la periferic, pentru a le transmite la UCP.

Registrul de comandă a măștii (2 biți) este încărcat de către UCP, pentru a specifica starea activă (nivel coborît/ridicat) a oricărui terminal, al echipamentului periferic manipulat și dacă trebuie să se genereze o întrerupere, cînd toate terminalele nemascate sînt active sau cînd unul din terminalele nemascate este activ.

Registrul mască (8 biți) este încărcat de către UCP, pentru a specifica terminalele echipamentului periferic care trebuie urmărite pentru o anumită condiție dată de stare.

Registrul de selecție pentru I/E (8 biți) este încărcat de către UCP pentru a stabili modul de lucru (intrare sau ieșire) pentru fiecare terminal, atunci când programarea s-a făcut pentru modul bit.

Descrierea terminalelor interfeței PIO.

În figura 4.20 se prezintă terminalele interfeței PIO. Ele au următoarele semnificații :

D7 + D0. Magistrala de date bidirecțională, de legătură cu UCP, poate intra în starea de mare impedanță.

SEL.PB/P_A. Selecția porturilor B sau A (intrare activă pe nivel ridicat).

SEL.CTRL/DATA. Selecție comanda sau date (intrare activă pe nivel ridicat).

CE. Activează circuitul (intrare activă pe nivel coborât).

∅. Semnal de ceas (intrare).

M1. Semnal pentru ciclul mașină M1, furnizat de către UCP (activ pe nivel coborât).

IORQ. Cerere de I/E de la UCP (intrare activă pe nivel coborât).

RD. Semnal de ciclu de citire, furnizat de către UCP (intrare activă pe nivel coborât).

IN.EN.IN (IEI). Semnal de intrare care specifică întreruperea activă (activ pe nivel ridicat).

IN.EN.OUT (IEO). Semnal de ieșire care specifică întreruperea activă (activ pe nivel ridicat). IEI și IEO formează o conexiune în lanț pentru comanda prioritară a întreruperii.

INT. Cerere de întrerupere (ieșire cu colector în gol) activă pe nivel coborât.

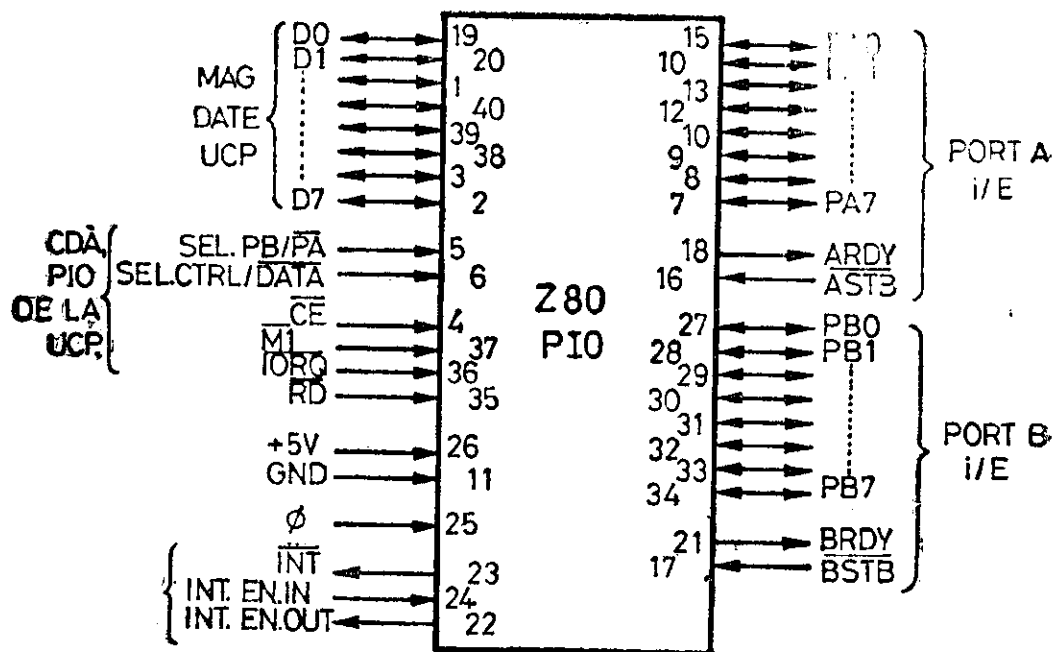


Fig. 4.20. Terminalele interfeței PIO.

PA0-PA7. Magistrala bidirecțională a portului A (poate intra în starea de cerere impedanță).

ASTB. Impuls de strob pentru portul A, de la echipamentul periferic (activ pe nivel coborât).

ARDY. Semnal care specifică faptul că registrul A este pregătit (ieșire, activă pe nivel ridicat).

PB0-PB7. Magistrala bidirecțională a portului B (poate intra în starea de mare impedanță).

BSTB. Impuls de strob pentru portul B, de la echipamentul periferic (activ pe nivel coborât).

BRDY. Semnal care specifică faptul că registrul B este pregătit (ieșire activă pe nivel ridicat).

Modurile de lucru ale interfeței PIO

Modul 0, de ieșire. Un ciclu de ieșire este amorsat prin execuția unei instrucțiuni de ieșire, de către UCP. Semnalul \overline{WR} furnizat de către UCP forțează datele pe magistrala de date, în portul de ieșire selectat. Impulsul de scriere poziționează indicatorul READY (fig. 4.21) după frontul căzător al semnalului Φ , indicând disponibilitatea informației. Linia READY rămâne activă până la recepționarea frontului pozitiv al semnalului STROB, care specifică prelucrarea datei de către periferic. Frontul pozitiv al semnalului STROB generează o întrerupere \overline{INT} , dacă bistabilul de activare a întreruperilor a fost poziționat în unu și dacă echipamentul în cauză are cea mai mare prioritate.

Modul 1, de intrare. Data este încărcată în portul de intrare selectat atunci când semnalul STROB trece pe un nivel coborât (4.22). Următorul front

Fig. 4.21. Sincronizarea pentru modul 0 (ieșire).

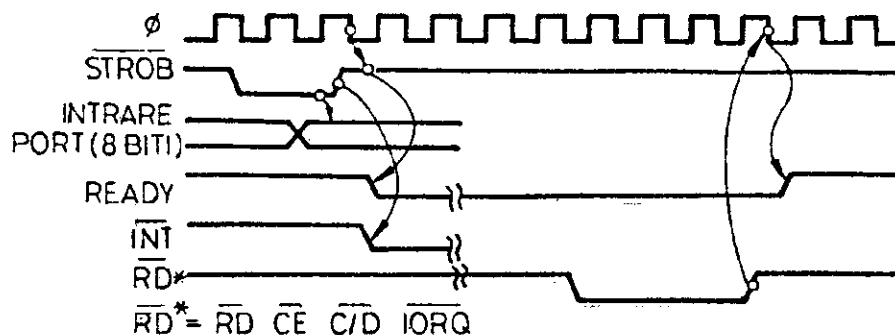
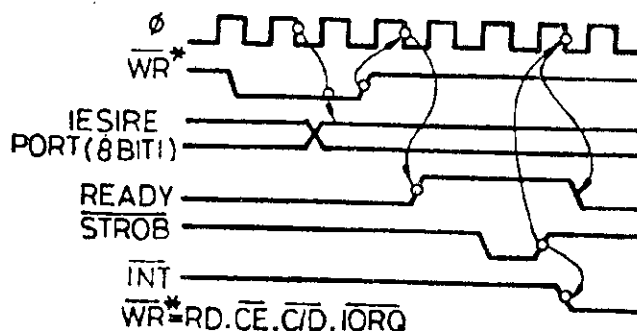


Fig. 4.22. Sincronizare pentru modul I (intrare).

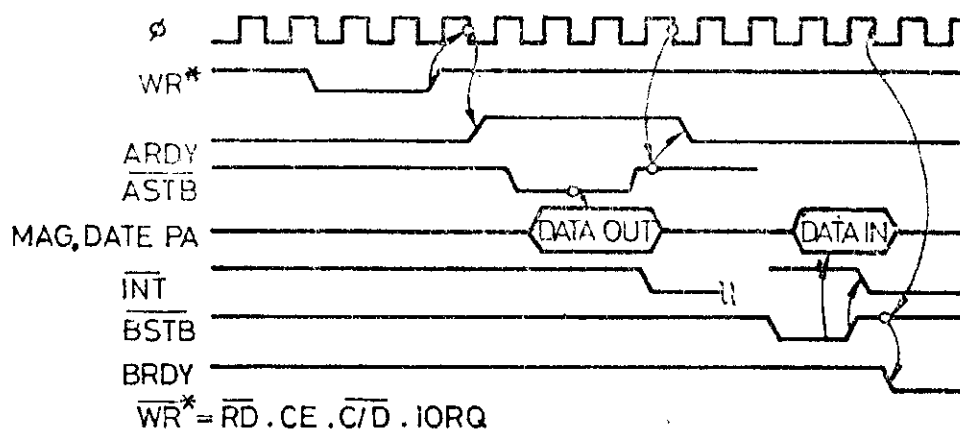


Fig. 4.23. Sincronizare pentru modul bidirecțional.

creșcător al semnalului STROB activează \overline{INT} , dacă bistabilul de activare a întreruperilor a fost poziționat în unu și dacă echipamentul în cauză are prioritatea cea mai mare. Următorul front căzător al lui ϕ aduce semnalul **READY** în stare inactivă, specificând faptul că portul de intrare conține informație și nu mai poate fi încărcat cu o altă informație până la citirea celei existente, de către UCP. După preluarea datei de către UCP, frontul pozitiv al lui \overline{RD} va activa **READY**, la următorul front negativ al semnalului de ceas ϕ . Astfel, o nouă informație poate fi înscrisă în PIO.

Modul bidirecțional. Acesta reprezintă o combinație a modurilor 0 și 1, folosind toate cele patru linii de dialog și cele opt linii de I/E, ale portului A. Portul B va fi programat în modul bit (fig. 4.23).

Liniile de dialog ale portului A se folosesc pentru ieșirea comenzii, iar cele ale portului B, pentru intrarea comenzii. Data se poate extrage din portul A, numai când semnalul \overline{ASTB} este pe nivel coborât. Frontul crescător al acestui strob poate fi folosit pentru a forța date în echipamentul periferic.

Modul bit. Acest mod nu folosește semnalele de dialog. O operație normală de scriere poate avea loc în orice moment.

La scriere, informația va fi forțată în registrele de ieșire, cu aceeași sincronizare ca și modul de ieșire.

Când are loc citirea PIO, fig. 4.24, datele transferate spre UCP vor consta din liniile de date, care au fost asignate ca ieșiri, ale portului de ieșire, și

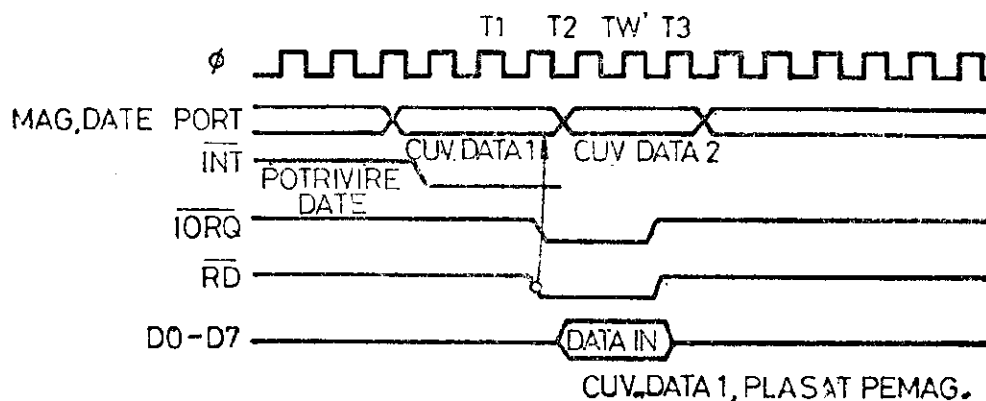


Fig. 4.24. Sincronizare pentru modul bit (intrare).

liniile de date, care au fost asignate ca intrări, ale portului de intrare. Registrul de intrare va conține informațiile, care au fost prezente imediat înaintea frontului căzător al semnalului RD. O întrerupere va fi generată dacă sînt activate întreruperile de la port și dacă informațiile de pe liniile de date ale portului satisfac ecuațiile logice, definite de registrul mască și registrul de comandă a măștii.

Recunoașterea unei întreruperi

Pe durata ciclului M1, (fig. 4.25) interfețele PIO sînt blocate în ceea ce privește modificarea stării de activare a întreruperii. Astfel, semnalul $\overline{\text{INT}}$

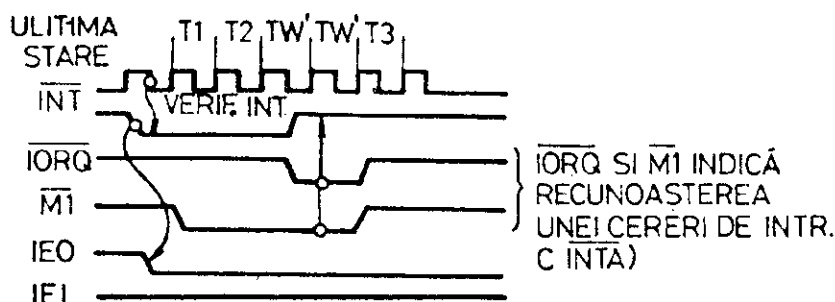


Fig. 4.25. Recunoașterea unei întreruperi.

poate parcurge succesiv întregul lanț. Perifericul cu liniile IEI la nivel ridicat și IEO la nivel coborît, pe durata $\overline{\text{IORQ}} \cdot \overline{\text{M1}}$ vor plasa un vector de întrerupere preprogramat, pe liniile de date. IEO este menținut la nivel coborît pînă la execuția, de către UCP, a unei instrucțiuni RETI (de revenire din întrerupere), în timp ce IEI este la nivel ridicat. În acest scop instrucțiunea RETI, de doi octeți, este decodificată intern, de către PIO.

Revenirea dintr-un ciclu de întrerupere

Dacă PIO are o cerere de întrerupere nerezolvată sau nu este în curs de servire, atunci $\text{IEO} = \text{IEI}$. În cazul cînd o întrerupere este în curs de servire (adică a efectuat o cerere de întrerupere și a primit un răspuns de acceptare), atunci IEO este la nivel coborît, inhibînd interfețele cu prioritate mai mică de a cere întreruperi. Dacă are o cerere de întrerupere lansată, fără să fie încă recunoscută, IEO va fi la nivel coborît pînă la decodificarea octetului „ED”, de pe liniile DO-D7 (fig. 4.26) care reprezintă primul octet, al unui cod de doi octeți. În acest caz, IEO va trece pe nivel ridicat din nou pînă la recepționarea următorului octet „4D” după care va trece pe nivel coborît.

Acest cod (ED 4D) constituie instrucțiunea RETI.

După codificarea codului de operație „ED”, numai interfața, care a cerut întreruperea și este curent servită, va avea IEI la nivel ridicat și IEO la nivel coborît. Această interfață are prioritatea cea mai mare în lanțul de întreruperi, care a primit semnalul de recunoaștere. Toate celelalte periferice au $\text{IEI} = \text{IEO}$. Dacă următorul cod de operație este „4D” interfața a cărei cerere de întrerupere a fost tratată va anula condiția întrerupere în curs de servire.

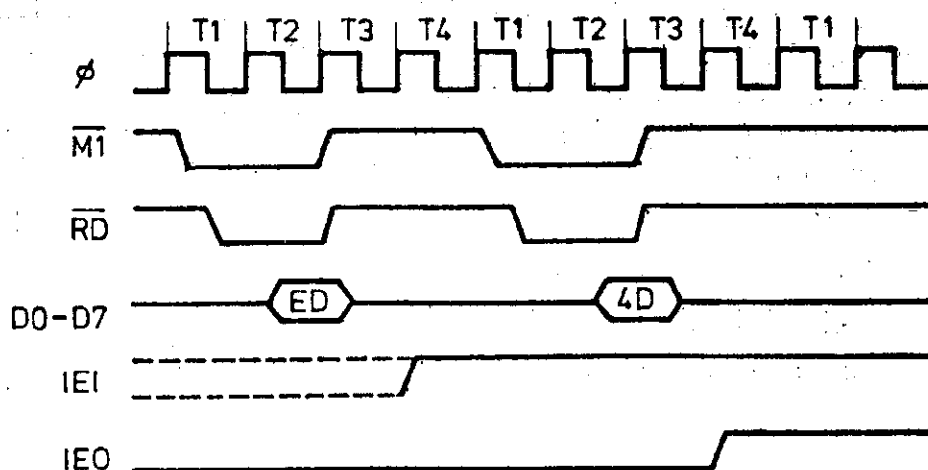
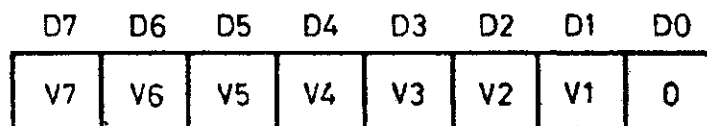


Fig. 4.26. Ciclul de revenire din întrerupere.

Încărcarea vectorului de întrerupere

UCP echipată cu Z80 necesită un vector de întrerupere, de 8 biți, din partea interfeței, care solicită întreruperea. Acest vector este folosit de către UCP, pentru a forma adresa subrutinei de tratare a întreruperii, pentru acel port. Echipamentul cu prioritatea cea mai mare va plasa magistrala D0-D7 vectorul respectiv, pe durata ciclului de recunoaștere a cererii de întrerupere. Vectorul de întrerupere este încărcat în PIO, prin scrierea în portul dorit, a unui cuvânt cu formatul din figura 4.27.



Semnifică faptul că
acest cuvânt de comandă este
un vector de întrerupere

Fig. 4.27. Încărcarea vectorului de întrerupere.

Selectarea modului de operare

La selectarea modului de operare, registrul de comandă (de doi biți) este forțat la una din cele patru valori posibile (fig. 4.28).

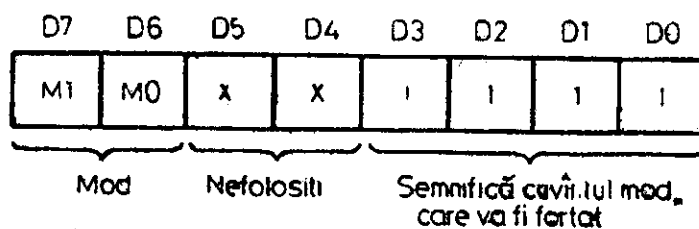


Fig. 4.28. Selecția modului de operare.

Biții D7-D6 (M1,M0) vor fi poziționați astfel :

M1	M0	MOD
0	0	ieșire,
0	1	intrare,
1	0	bidirecțional,
1	1	bit.

Biții D3-D0 vor fi forțați în unu, pentru a specifica selecția modului, în timp ce biții D5 și D4 sînt nefolosiți.

Modul 0 arată că informația trebuie să fie transmisă de la UCP, la echipamentul periferic.

Modul 1 specifică preluarea informației de la echipamentul periferic, către UCP.

Modul 2 se referă la o operație de citire sau scriere al un echipament periferic.

Modul 3 se folosește pentru aplicațiile în care se generează semnalele de comandă sau se monitorizează stări. După selectare, următorul cuvînt de comandă va poziționa registrul de I/E, bit cu bit, pentru a specifica liniile folosite ca intrări și liniile folosite ca ieșiri.

Astfel, I/E=1 poziționează linia respectivă ca intrare, în timp ce I/E=0, a poziționează ca ieșire.

Comanda întreruperilor

Comanda întreruperilor se realizează prin cuvîntul cu structura dată în figura 4.29, în care biții D7-D0 au următoarele semnificații :

- Bitul 7=1 activează logica de întrerupere, permițînd generarea unei întreruperi.
- Bitul 7=0 dezactivează logica de întrerupere, inhibînd generarea unei întreruperi.
- Biții 6, 5, 4 sînt folosiți în modul bit, în cadrul operațiilor de întrerupere, în caz contrar sînt neglijați.
- Biții 3, 2, 1, 0 specifică faptul că este vorba de un cuvînt de comandă pentru întrerupere.

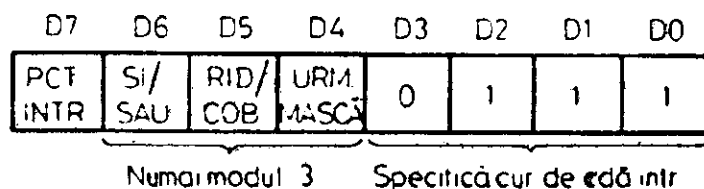
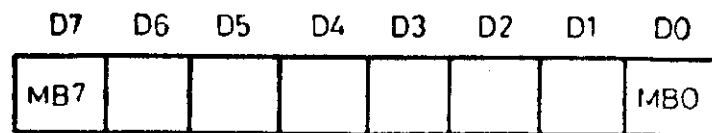


Fig. 4.29. Cuvîntul de comandă a întreruperilor.

Dacă urmează un cuvînt mască, bitul D4 din figura 4.30 trebuie să fie unu, iar următorul cuvînt înscris în port trebuie să fie masca, cu configurația indicată în figura 4.30.



Pentru generarea intreruperii vor fi monitorizati
bitii pentru care $MB_i = 0$

Fig. 4.30. Cuvintul mască.

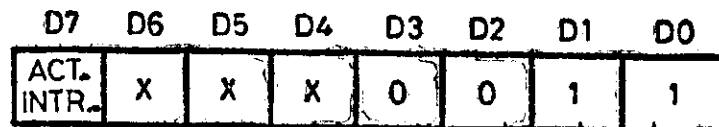


Fig. 4.31. Cuvintul de activare/dezactivare a
logicii de intrerupere.

Bistabilul de activare a logicii de intrerupere a portului poate fi poziționat în unu, fără a modifica restul cuvintului de comandă a intreruperilor, folosind un cuvint de comandă cu structura din figura 4.31.

4.9. Interfața serială programabilă SIO

Interfața SIO, prevăzută cu două canale, este destinată aplicațiilor de transmisie serială a datelor, folosind microcalculatoare.

SIO are în principal funcția de convertor/unitate de comandă, pentru transformarea datelor de la forma serială, la forma paralelă și invers. Ea este capabilă să manipuleze formate asincrone, protocoale sincrone orientate pe octet (BSC-IBM)*, ca și protocoalele sincrone orientate pe bit (HDLC și SDLC)**. De asemenea, în alte aplicații, privind conectarea unor echipamente seriale (casetă magnetică, etc.), SIO poate fi utilizat pentru asigurarea protocolului necesar. Pentru verificarea corectitudinii datelor la recepție/transmisie, SIO este prevăzută cu facilități de generare și verificare a codurilor (CRC***). Interfața poate fi cuplată la canalele de comunicații telefonice/telegrafice, folosind echipamente de tip modem, pentru care posedă semnalele de comandă necesare.

Structura

Interfața SIO este realizată în tehnologia NMOS, pe o pastilă cu 40 terminale. Necesită o singură sursă de alimentare de +5V și un singur semnal de ceas, cu amplitudinea de 5 V. Toate intrările și ieșirile sînt compatibile TTL.

Structura interfeței, la nivel de schemă bloc, este dată în figura 4.32. Se constată că cele două canale A, E pot funcționa independent unul de celălalt, fiind prevăzute cu registrele și logica necesare conversiei serial/paralel și paralel/serial, a datelor. Sînt prevăzute, de asemenea: interfața cu magistrala UCP, logica de comandă internă și logica de comandă a intreruperilor.

*) Binary-Synchronous Communications.

**) High Level Data Link Control și Synchronous Data Link Control.

**) Cyclic Redundancy Chekword.

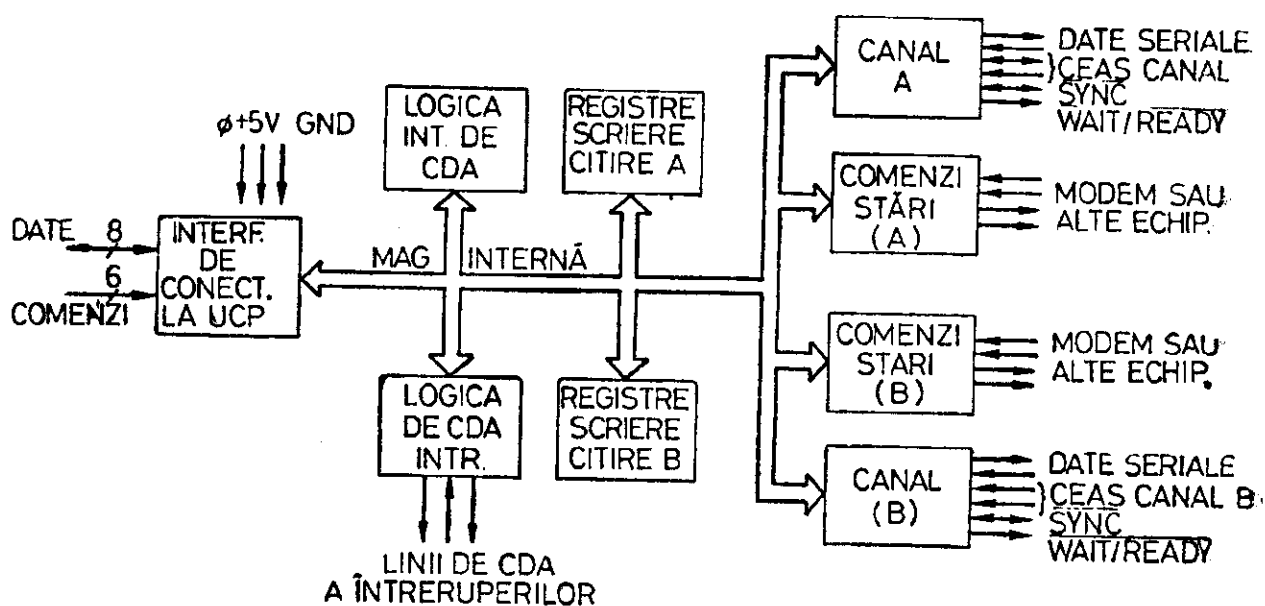


Fig. 4.32. Schema bloc a interfeței SIO.

Cele două canale lucrează în regim duplex asigurând, în modurile sincron și isosincron, viteza de transmisie de $0 \div 500$ K biți/s, la o frecvență a ceasului sistemului de 2,5 MHz și de $0 - 800$ K biți/s, la o frecvență a ceasului sistemului de 4 MHz.

În modul de lucru asincron se pot manipula caractere având 5, 6, 7 sau 8 biți, cu 1, $1\frac{1}{2}$ sau 2 biți de stop. Paritatea poate fi: pară, impară, absentă. S-a prevăzut posibilitatea detecției erorilor de paritate, depășire și cadrare.

Schema de întreruperi poate fi organizată sub forma serial-înlanțuită fără a mai fi necesară o logică externă pentru forțarea vectorului de întrerupere.

Vectorul de întrerupere forțat automat poate fi programat de către utilizator în mod corespunzător.

Circuitul dispune de facilități de manipulare a erorilor folosind coduri ciclice redundante: CRC-16 sau CRC-CCITT, pentru verificări de cadre de blocuri.

Descrierea terminalelor (fig. 4.33)

D7 ÷ D0 — *Magistrala de date a sistemului* (bidirecțională, cu posibilitatea de a intra în starea de mare impedanță). Pe această magistrală se transferă date și comenzi între UCP și SIO.

B/ \bar{A} — *Selecția canalului A sau B*. Semnalul de nivel ridicat selectează canalul B. Canalul selectat va fi folosit pentru transferul datelor cu UCP. Adesea pentru selecție se folosește bitul A0 al magistralei de adrese a UCP.

C/ \bar{D} — *Selecție comanda sau date*. Semnalul pe nivel ridicat selectează comanda.

În acest mod se definește tipul informației care se vehiculează între UCP și SIO. Pe durata unei scrieri în SIO, dacă această intrare este pe nivel ridicat, informația transmisă de UCP către canal este interpretată ca o comandă.

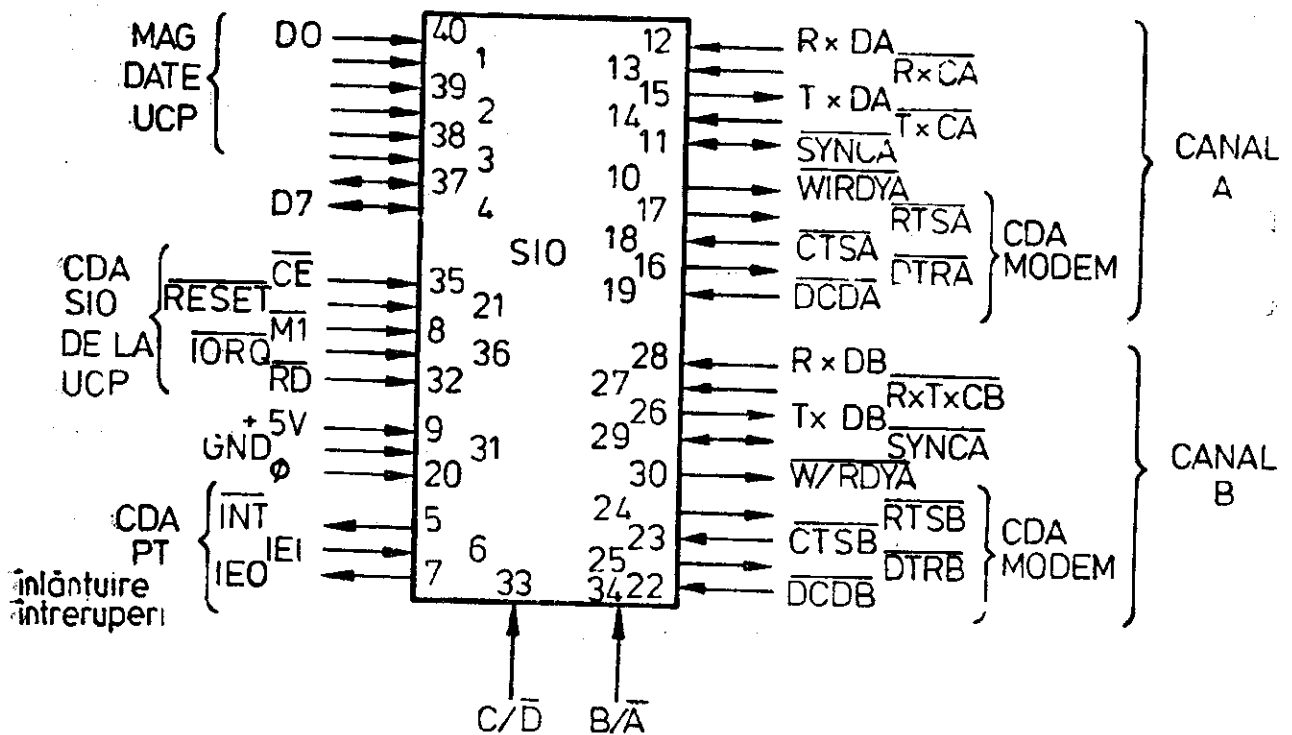


Fig. 4.33. Terminalele interfeței SIO.

În caz contrar, este interpretată ca informație. Pentru această funcțiune se folosește adesea bitul A1, al magistralei de adrese a UCP.

\overline{CE} — Activare circuit (activ pe nivel coborît). Semnalul \overline{CE} activ determină SIO să accepte comenzi sau date de la UCP, pe durata unui ciclu de scriere, sau să transmită date, pe durata unui ciclu de citire.

Φ — Ceasul sistemului. Asigură sincronizarea semnalelor interne în SIO.

$\overline{M1}$. — Ciclul mașină $\overline{M1}$ (activ pe nivel coborît). Când $\overline{M1}$ este activ simultan cu \overline{IORQ} , SIO va interpreta această situație ca o recunoaștere a cererii de întrerupere din partea echipamentului cu prioritatea cea mai mare, dacă ea reprezintă acest echipament.

\overline{IORQ} — Cerere de I/E, intrare furnizată de UCP folosită în conjuncție cu B/\overline{A} , C/\overline{D} , \overline{CE} și \overline{RD} pentru a transforma comenzi și date între UCP și SIO. Când \overline{CE} , \overline{RD} și \overline{IORQ} sînt active, canalul selectat de B/\overline{A} transferă date către UCP. Când \overline{CE} și \overline{IORQ} sînt active, dar \overline{RD} este inactiv, canalul selectat de B/\overline{A} primește informație de la UCP sub formă de date sau comenzi, după cum este specificat de semnalul C/\overline{D} . Când \overline{IORQ} și $\overline{M1}$ sînt simultan active, UCP recunoaște o cerere de întrerupere, iar SIO va plasa automat vectorul său de întrerupere pe magistrala de date a UCP, dacă reprezintă echipamentul cu prioritatea cea mai mare, care solicită întreruperea.

\overline{RD} — Semnal corespunzător unui ciclu de citire. Este emis de UCP, pentru a specifica o operație de citire din memorie sau de la un port de intrare. Pentru a transfera date de la SIO către UCP se folosește în conjuncție cu semnalele B/\overline{A} , \overline{CE} și \overline{IORQ} .

RESET — *Semnal de inițializare*, activ pe nivel coborît. Semnalul de intrare **RESET** dezactivează receptorii și emițătorii din SIO, forțează în zero ieșirile TxDA și TxDB, aduce semnalele de comandă pentru modemuri la nivel ridicat și dezactivează toate întreruperile. Registrele de comandă trebuie să fie reinițializate după ce SIO a primit semnalul **RESET**, înainte ca datele să fie transmise sau recepționate.

IEI — *Semnal de intrare*, care specifică întrerupere activă (activ pe nivel ridicat). Semnalul este folosit împreună cu IEO pentru a realiza o schemă de priorități înlănțuite, atunci când există mai multe echipamente, care lucrează în întreruperi.

Un nivel ridicat specifică faptul că nu există un alt echipament cu o prioritate mai mare în situația de a fi cerut o întrerupere și care este în mod curent tratată de către UCP.

IEO — *Semnal de ieșire* care specifică întrerupere activă. Este activ pe nivel ridicat. IEO este la nivel ridicat, dacă IEI este la nivel ridicat și dacă UCP nu tratează o cerere de întrerupere furnizată de acest SIO. Acest semnal blochează echipamentele cu prioritate mai mică de a cere întreruperi, în timp ce un echipament cu prioritate mai mare este servit, prin rutina lui specifică, de către UCP.

INT — *Cerere de întrerupere* (ieșire, cu colectorul în gol, activă pe nivel coborît). Când SIO solicită o întrerupere forțează **INT** la nivel coborît.

W/RDYA, W/RDYB — (Wait/Ready A, Wait/Ready B). Acestea reprezintă ieșiri cu colectorul în gol, când sînt programate pentru funcția Wait, și sînt comandate la nivel ridicat sau coborît, când sînt programate pentru funcția Ready. Aceste ieșiri cu rol dublu pot fi programate ca linii Ready, pentru unitatea de comandă DMA sau ca linii Wait, pentru sincronizarea UCP cu debitul de date al SIO. Starea de inițializare corespunde ieșirii în gol.

CTSA, CTSB — (Clear to Send). *Intrări active pe nivel coborît*. Când sînt programate pentru autoactivare, un semnal coborît pe aceste intrări activează emițătorul respectiv. În cazul în care nu sînt programate pentru autoactivare, ele pot fi folosite ca intrări universale. Ele sînt prevăzute cu trigger-e Schmitt pentru semnale cu fronturi lente. Aceste semnale vor întrerupe UCP pe ambele tranziții ale nivelurilor logice.

DCDA, DCDB — (Data Carrier Detect). *Intrări active pe nivel coborît*. Aceste intrări au rolul de activare a receptorului, în cazul în care SIO a fost programat pentru autoactivare. În caz contrar, ele se pot folosi ca intrări de uz general. Pentru a reacționa la fronturi lente sînt prevăzute cu trigger-e Schmitt. Impulsurile obținute pe ambele tranziții ale fronturilor generează întreruperi către UCP.

RxDA, RxDB. *Intrări pentru recepția datelor*, active pe nivel ridicat.

TxDA, TxDB. *Ieșiri pentru transmisia datelor*, active pe nivel ridicat.

RxCA, RxCB. *Intrări pentru orologiile de recepție*. Datele recepționate sînt testate pe frontul crescător al lui **RxC**, cu o frecvență de 1, 16, 32 sau 64 ori mai mare decît viteza de transmisie în modurile asincrone. Sînt prevăzute cu trigger-e Schmitt.

$\overline{\text{TxCA}}$, $\overline{\text{TxCB}}$ — Intrări pentru orologiile de transmisie, active pe nivel coborât. Informațiile pe liniile de date se modifică pe frontul căzător al semnalului TxC . În modurile asincrone frecvențele orologiilor de transmisie și recepție trebuie să fie aceleași (1, 16, 32 sau 64 ori mai mare decât frecvența cu care sînt recepționate datele). Intrările sînt prevăzute cu trigger-e Schmitt.

$\overline{\text{RTSA}}$, $\overline{\text{RTSB}}$ (Request to Send). Ieșiri active pe nivel coborât. Cînd în registrul intern W5, bitul D1 (RTS) este poziționat în unu, ieșirea $\overline{\text{RTS}}$ trece pe un nivel coborât. În modul asincron, cînd același bit D1 este forțat în zero, ieșirea trece pe nivel ridicat, dacă emițătorul este vid. În modul sincron $\overline{\text{RTS}}$ urmărește starea bitului D1 (RTS) din registrul intern W5.

$\overline{\text{DTRA}}$, $\overline{\text{DTRB}}$ (Data Terminal Ready). Ieșiri active pe nivel coborât. Aceste ieșiri urmăresc starea programată a bitului DTR (D7) din registru intern W5.

$\overline{\text{SYNCA}}$, $\overline{\text{SYNCB}}$. Sincronizări. Intrări/ieșiri active pe nivel coborât. În modul asincron de recepție, ele reprezintă intrări similare cu $\overline{\text{CTS}}$ și $\overline{\text{DCD}}$. În acest mod, tranzițiile pe aceste linii afectează starea biților Sync/Hunt, din registrul de recepție R0. În modul de sincronizare externă, aceste linii se folosesc ca intrări. După ce s-a detectat caracterul de sincronizare, logica externă trebuie să aștepte, pentru a activa intrarea $\overline{\text{SYNC}}$, un interval de timp corespunzător la două cicluri de recepție. După ce $\overline{\text{SYNC}}$ a fost forțat la nivel coborât, el se va menține la acest nivel pînă cînd UCP informează logica externă că s-a pierdut sincronizarea sau că va începe un nou mesaj. Asamblarea caracterelor începe pe frontul crescător al lui $\overline{\text{RxC}}$, care precede frontul căzător al semnalului $\overline{\text{SYNC}}$, în modul de Sincronizare externă.

În cazul modului de sincronizare internă, terminalele $\overline{\text{SYNCA}}$, $\overline{\text{SYNCB}}$ funcționează ca ieșiri, care sînt active pe durata aceluși ciclu al ceasului de recepție ($\overline{\text{RxC}}$), în care sînt recunoscute caracterele sync. Condițiile sync nu sînt forțate în bistabile, astfel că, aceste ieșiri sînt active de fiecare dată, cînd se recunoaște un caracter sync.

Variante ale interfeței SIO

Restricția referitoare la cele 40 terminale ale capsulei face imposibil accesul din exterior la ceasul pentru recepție, ceasul pentru transmisie, DTR și SYNC simultan, pentru ambele canale. De aceea, canalul B va sacrifica un semnal sau va reuni pe același terminal două semnale. Astfel, sînt oferite trei variante :

- SIO/0 are toate cele patru semnale, cu observația că $\overline{\text{TxCB}}$ și $\overline{\text{RxCB}}$ sînt grupate pe același terminal,
- SIO/1 sacrifică $\overline{\text{DTRB}}$ și menține $\overline{\text{TxCB}}$, $\overline{\text{RxCB}}$ și $\overline{\text{SYNCB}}$,
- SIO/2 sacrifică $\overline{\text{SYNCB}}$ și menține $\overline{\text{TxCB}}$, $\overline{\text{RxCB}}$ și $\overline{\text{DTRB}}$

Arhitectura SIO

Structura internă a SIO include interfața cu UCP, logica internă de comandă și logica de întrerupere, precum și cele două canale duplex. Fiecare

canal conține registre de scriere și citire și logica pentru comenzi și stări, care asigură interfața cu modemurile sau alte echipamente externe.

Registrele de citire și scriere constau din cinci registre de comandă, de câte 8 biți, două registre pentru caracterele de sincronizare și două registre de stare. Vectorul de întrerupere este înscris într-un registru suplimentar de 8 biți (WR2-registrul de scriere 2) din canalul B. Registrele pentru cele două canale sînt marcate după cum urmează :

WR + 0WR7 — registrele de scriere 0 ÷ 7,

RR + 0RR2 — registrele de citire 0 ÷ 2.

Funcțiunile registrelor sînt date mai jos :

- RR 0 — conține starea tamponului de emisie/recepție, starea întreruperii și stări externe ;
- RR1 — conține starea condițiilor speciale de recepție ;
- RR2 — memorează vectorul modificat de întrerupere (numai canalul B) ;
- WR 0 — stochează indicatorii registrelor, inițializarea CRC, comenzile de inițializare pentru diferite moduri ;
- WR1 — definește întreruperea de Emisie/Recepție și modul de transfer al datelor ;
- WR2 — conține vectorul de întrerupere (numai canalul B) ;
- WR3 — stochează parametrii de recepție și comandă ;
- WR4 — memorează diverși parametrii de recepție și comandă ;
- WR5 — memorează parametrii de emisie și comandă ;
- WR6 — conține caracterul Sync sau cîmpul de adresă SDLC ;
- WR7 — conține caracterul Sync sau semaforul SDLC.

Logica pentru ambele canale asigură formatele, sincronizarea și validarea datelor transferate către și de la interfața canalului. Intrările de comanda ale modemului CTS și DCD sînt monitorizate de o logică discretă de comandă, sub controlul programului.

Pentru cazul întreruperilor vectorizate cu forțare automată, logica de comandă determină care canal și care dispozitiv, în cadrul canalului respectiv, are cea mai mare prioritate. Prioritatea cea mai mare o are canalul A, iar în cadrul canalului Recepția, Transmisia și întreruperile Externe/Stare au prioritățile în ordine descrescîndă.

Ambele canale sînt prevăzute cu registre identice la recepție și transmisie.

Recepția este asigurată printr-un tampon de trei registre de câte 8 biți organizate sub forma primul intrat-primul ieșit (FIFO) și de un registru de deplasare-receptor. Aceasta permite crearea unui interval de timp suplimentar pentru ca UCP să trateze o întrerupere la sosirea unui bloc de date. Datele recepționate pot fi transferate prin lanțul de date sau lanțul de verificare CRC, în funcție de modul selectat, iar în modul asincron și de lungimea caracterului.

Emisia este asigurată cu ajutorul unui registru de date, de 8 biți, care se încarcă de la magistrală internă și de un registru de deplasare emițător, de 20 de biți, care poate fi încărcat din tampoanele (W6, W7) ale caracterelor de sincronizare sau de la registrul de date.

SIO poate fi examinat ca interfață specializată pentru transmisii seriale, în cadrul familiei de circuite ale microprocesorului Z80 sau ca dispozitiv de co-

municații, care emite și recepționează date sub formă serială, corespunzător anumitor protocoali.

În primul caz SIO utilizează liniile de date, adrese și comenzi ale microprocesorului Z80 și se încadrează în structura sistemului său de întreruperi.

Pentru transferul datelor, al stărilor și comenzilor către/dela UCP, SIO poate folosi metodele: interogare, întreruperi (vectorizate sau nevectorizate) și transferul în blocuri. Acesta din urmă se poate realiza sub controlul UCP sau al circuitului de acces direct la memorie (DMA).

Interogarea se referă la examinarea stărilor conținute în registrele RR0, pentru fiecare canal. Registrele de stare RR0 și RR1 sînt actualizate cu ocazia efectuării fiecărei funcții în SIO. Pentru aceasta, modurile de întrerupere ale SIO trebuie să fie dezactivate.

Biții de stare din RR0 servesc ca o recunoaștere a cererii de interogare. Biții D0 și D2, din RR0, specifică necesitatea unui transfer de date. Același registru conține indicații privind erorile sau alte condiții speciale de stare. Nu este necesară citirea din RR1 a stării corespunzătoare condiției speciale de recepție, deoarece biții de stare din RR1 trebuie să fie însoțiți de starea de disponibilitate a unui caracter (data în RR0).

Întreruperile în SIO sînt organizate într-o manieră care permite un răspuns rapid, în timp real. Registrele WR2 și RR2, din canalul B, conțin vectorul de întrerupere necesar stabilirii adresei de start, a rutinei de tratare. Vectorul de întrerupere din RR2 poate fi modificat, prin program, pentru a putea specifica direct una din cele opt rutine de tratare a întreruperilor. Prin poziționarea în unu a bitului D2, din WR1, vectorul de întrerupere din WR2 poate fi modificat în conformitate cu prioritățile atribuite diferitelor condiții de întrerupere.

Principalele surse de întrerupere se referă la: emisie, recepție și stări/externe.

Fiecare sursă de întrerupere este activată sub controlul programului.

La activarea întreruperii pentru emisie, UCP va fi întrerupt cînd timpul de emisie devine vid.

În cazul activării întreruperii la recepție, UCP poate fi întrerupt în următoarele situații:

- Întrerupere la primul caracter recepționat,
- Întrerupere după recepționarea tuturor caracterelor,
- Întrerupere la condiții speciale de recepție (în modul caracter sau mesaj).

Întreruperile referitoare la stări/externe sînt asociate cu tranzițiile semnalelor CTS, DCD și SYNC și de unele condiții de eroare.

Transferurile de date în blocuri, în conjuncție cu UCP sau DMA, sînt realizate folosind semnalele WAIT/READY, în asociație cu biții W/R, din registru WR1. Ieșirea WAIT/READY poate fi definită sub controlul programului ca linie WAIT, pentru UCP (în modul transfer de bloc), sau ca linie READY, pentru DMA (în modul transfer de bloc). Pentru UCP, ieșirea WAIT

indică faptul că SIO nu este pregătit pentru transfer, solicitând UCP să-și extindă ciclul de I/E. Pentru unitatea de comandă DMA, ieșirea READY specifică faptul că SIO este pregătit pentru a transfera date de la la memorie.

Ca dispozitiv pentru recepția/emisia serială a datelor, SIO asigură două canale independente, care pot lucra în modul duplex. Ele pot fi programate să lucreze în modurile asincrone, sincron și SDLC (HDLC).

În continuare se vor trata pe scurt numai modurile asincrone*. SIO poate manipula caractere de 5—8 biți prevăzute opțional cu bit de paritate (pară/impară) și cu biți de start, stop (1, 1^{1/2}, 2).

Emisia poate fi întreruptă în orice moment. La recepție, UCP este întrerupt numai la începutul și sfârșitul caracterului.

Erorile de cadru sau de depășire detectate sînt memorate împreună cu caracterul în cadrul căruia au apărut. Erorile de cadru apar ca urmare a adăugării unui interval de timp de 1^{1/2} bit, la punctul la care începe căutarea pentru bitul de start al unui nou caracter.

Programarea SIO se realizează printr-o serie de comenzi care inițializează modul de bază de operare și apoi alte comenzi care stabilesc condițiile în cadrul modului selectat. De exemplu, în modul asincron se stabilesc mai întii: lungimea caracterului, frecvența orologiului, numărul biților de stop, condiția de paritate, modul de întrerupere și în final se activează emițătorul sau receptorul. Parametrii pentru registrul WR4 vor fi transmiși înaintea altor parametri, de către rutina de inițializare.

Ambele canale conțin registre de comandă, care trebuie să fie programate separat înainte de a se începe alte operații. Pentru aceasta UCP va folosi intrările C/D, B/A ale SIO.

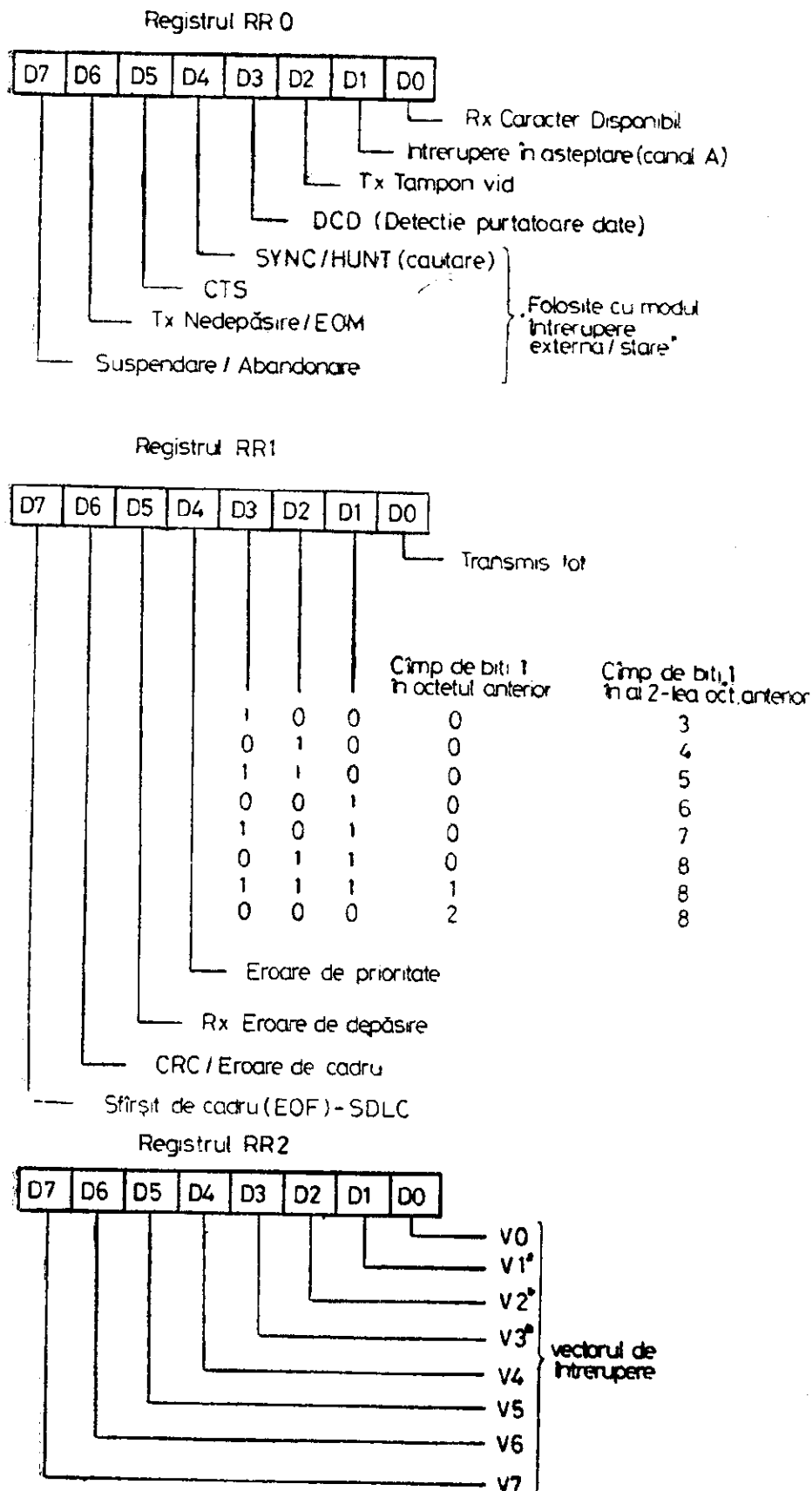
Registrele de citire. SIO conține trei registre RR0-RR2, care pot fi citite pentru a obține informația de stare pentru fiecare canal (fig. 4.34). Informațiile de stare includ condițiile de eroare, vectorul de întrerupere și semnalele standard ale interfeței de comunicație.

Pentru a citi conținutul unui registru selectat, diferit de RR0, este necesar mai întii să se scrie în WR0 un octet indicator în același mod ca în cazul operației de scriere într-un registru. În continuare, executînd o instrucțiune de intrare, conținutul registrului adresat poate fi citit.

Registrele de scriere. Ele sînt în număr de opt, pentru fiecare canal, și pot fi programate separat. Cu excepția lui WR0, programarea registrelor de scriere necesită doi octeți. Primul octet conține trei biți (D0—D2), care indică registrul selectat, iar al doilea va reprezenta cuvîntul de comandă propriu-zis.

Registrul WR0 constituie un caz special prin aceea că toate comenzile de bază (CMD0—CMD2) pot fi asigurate printr-un singur octet. Semnalul Reset (intern/extern) inițializează indicatorul D0—D2 la WR0.

*) A se vedea: Z80-SIO Technical Manual, ZILOG Corp. 1980.



Pot avea un caracter variabil dacă s-a programat "starea afectează vectorul"

Fig. 4.34. Registrele de citire RRO-RR2.

În figura 4.35 a, b, c, d, sînt date modalitățile de manipulare ale registrelor WR0—WR7 și semnificațiile lor la nivel de biți.

Sincronizarea SIO

În legătură cu sincronizarea SIO vor fi examinate ciclurile de citire, scriere, recunoaștere, întrerupere și revenire din întrerupere.

Ciclul de citire, generat de execuția unei instrucțiuni de intrare pentru citirea datelor sau a stărilor din SIO, se caracterizează prin semnalele date în figura 4.36.

Ciclul de scriere, generat de execuția unei instrucțiuni de ieșire pentru înscrierea în SIO a datelor sau comenzilor, este ilustrat în figura 4.37.

Ciclul de recunoaștere a unei întreruperi este prezentat în figura 4.38. După recepționarea unui semnal de întrerupere \overline{INT} , UCP va trimite ca răspuns semnalele $\overline{M1}$ și \overline{IORQ} . Circuitele înlănțuite de întrerupere vor determina cererea activă cu prioritatea cea mai mare, din cadrul lanțului. Intrarea IEI a perifericului cu prioritatea cea mai mare este la nivel ridicat. Perifericele ce nu au o întrerupere care așteaptă să fie tratată sau o întrerupere în curs de servire vor avea semnalele $IEO=IEI$. Perifericul care are o întrerupere ce așteaptă să fie tratată sau în curs de tratare forțează ieșirea IEO la nivel coborît.

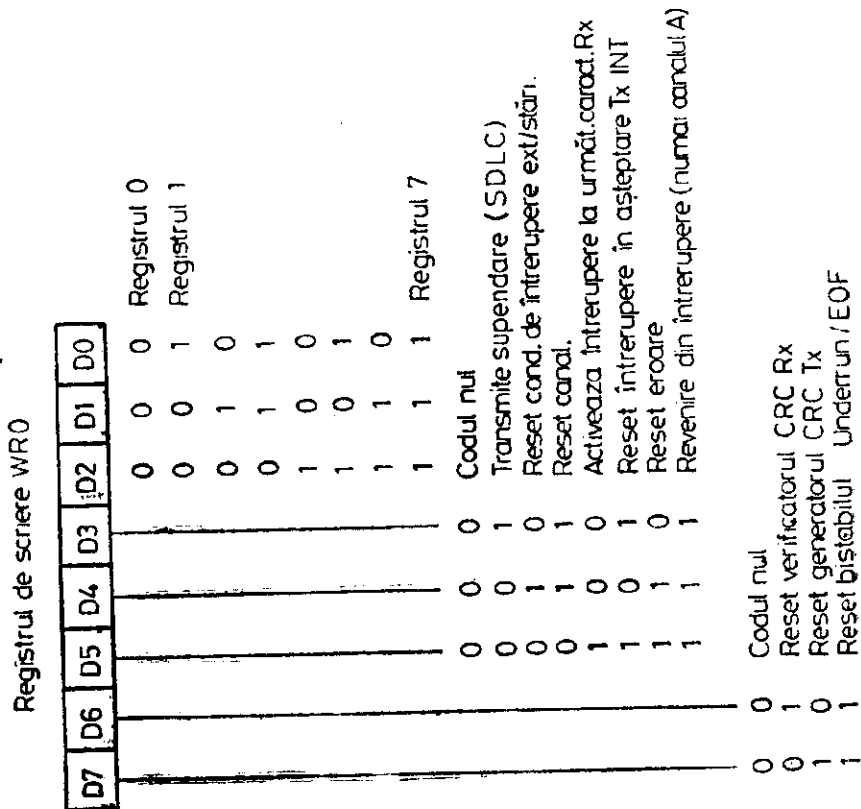
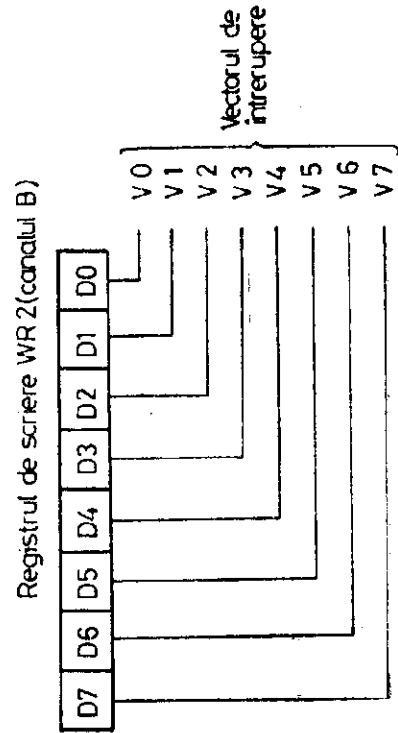
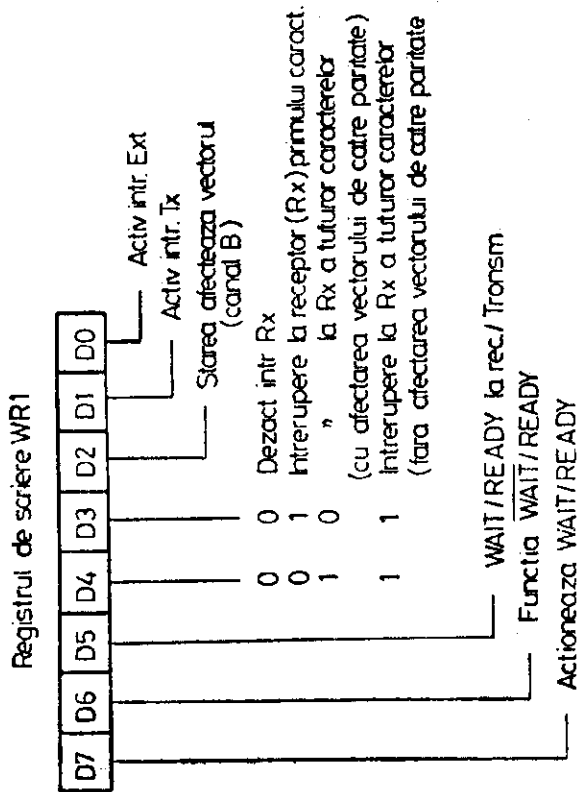
Pentru a asigura condiții stabile în cadrul lanțului de întrerupere, se vor bloca, pe durata semnalului $\overline{M1}$, toate modificările asociate cu apariția unor eventuale noi cereri de întrerupere. Cînd \overline{IORQ} este la nivel coborît, elementul care solicită întreruperea și are prioritatea cea mai mare (acela cu IEI la nivel ridicat) plasează vectorul său de întrerupere pe magistrala de date și își activează bistabilul intern, care specifică condiția de întrerupere în curs de servire.

Ciclul de revenire din întrerupere este ilustrat prin diagrama de semnale din figura 4.39. La sfîrșitul unei rutine de tratare a întreruperii, UCP forțează în mod normal o instrucțiune RETI, constînd din doi octeți (ED-4D).

RETI dezactivează bistabilul care specifică condiția de întrerupere în curs de tratare, pentru elementul a cărui cerere de întrerupere a fost tratată. În acest scop, pe lanțul de priorități vor avea loc mai multe operații. Astfel, pe lanțul de prioritate se poate detecta o cerere de întrerupere în așteptarea tratării, fără a se putea face o distincție între o cerere de întrerupere în curs de tratare și o cerere de întrerupere cu prioritate mai mare, care încă nu a fost recunoscută de către UCP. La decodificarea octetului ED, lanțul va suferi o modificare în sensul că IEO este forțat la nivel ridicat pentru oricare întrerupere ce nu a fost recunoscută. Astfel, lanțul identifică elementul cu cererea de întrerupere în curs de tratare ca pe acela cu intrarea IEI la nivel ridicat și ieșirea IEO la nivel coborît. Dacă următorul octet este 4D, bistabilul de întrerupere în curs de servire se dezactivează.

Numărul de echipamente care pot fi plasate în lanțul de prioritate (prin interfețele de tip SIO, PIO etc.) sînt limitate de timpul necesar parcurgerii semnalelor de activare/inhibare pe lanțul respectiv. Se pot folosi tehnicile bazate pe utilizarea unui circuit de transport anticipat sau pe extinderea ciclului de recunoaștere a unei întreruperi.

În figura 4.40 se prezintă imbricarea cererilor de întrerupere în cadrul unei organizări cu prioritate înlănțuită.



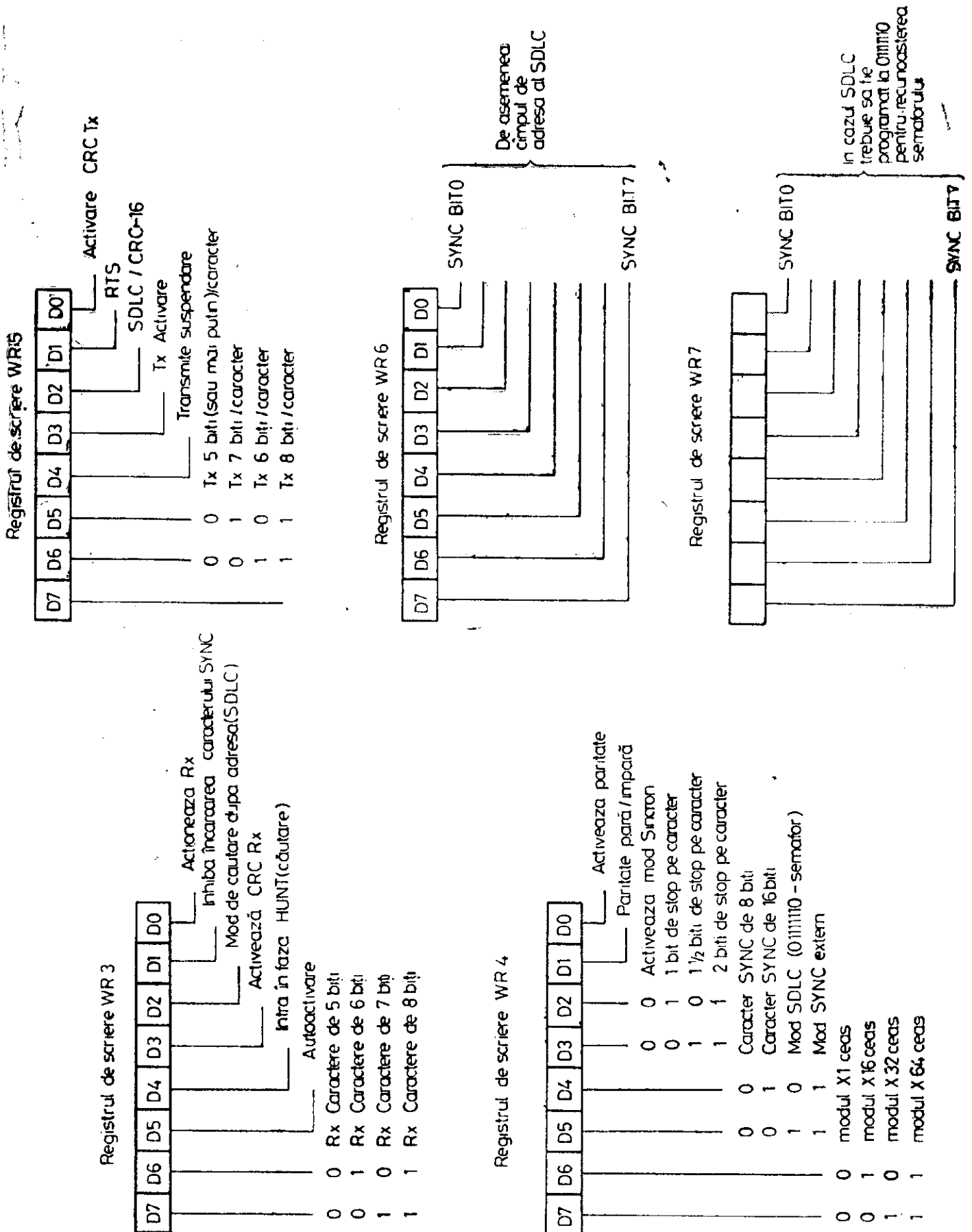


Fig. 4.35. Registrele de scriere WR0-WR7.

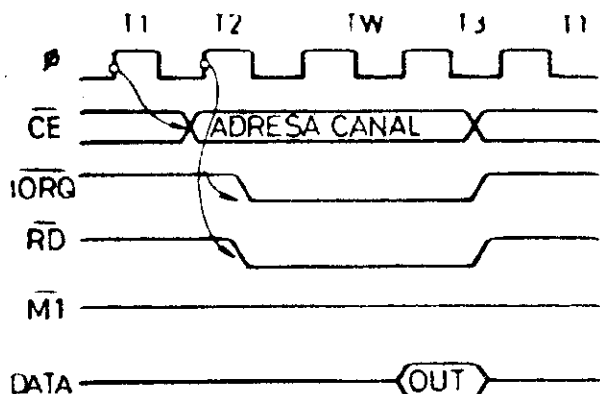


Fig. 4.36. Ciclul de citire.

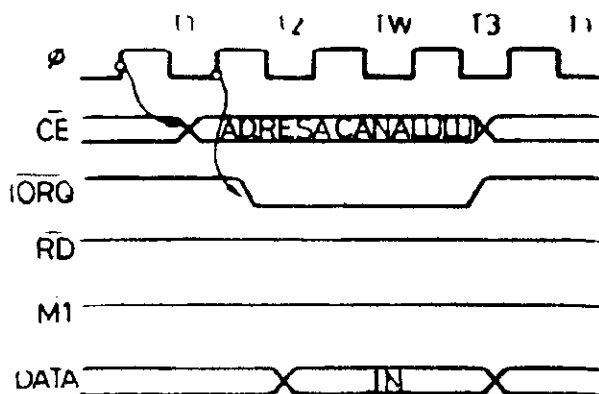


Fig. 4.37. Ciclul de scriere.

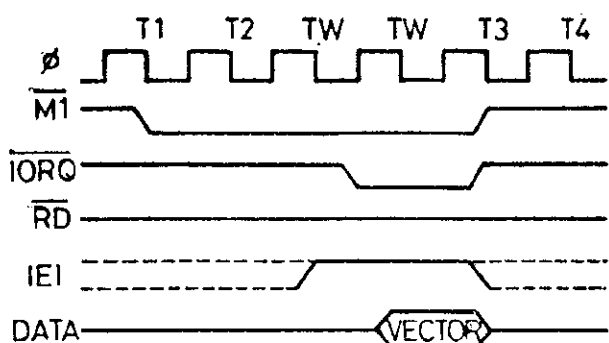


Fig. 4.38. Ciclul de recunoaștere a unei întreruperi.

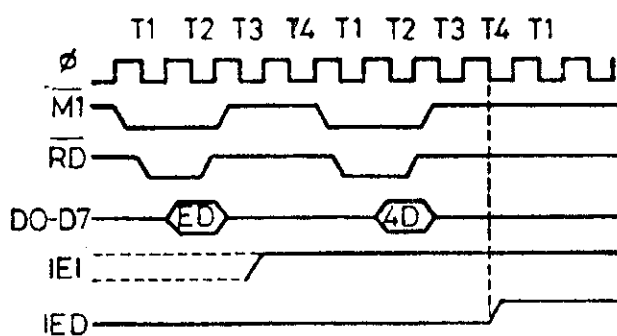
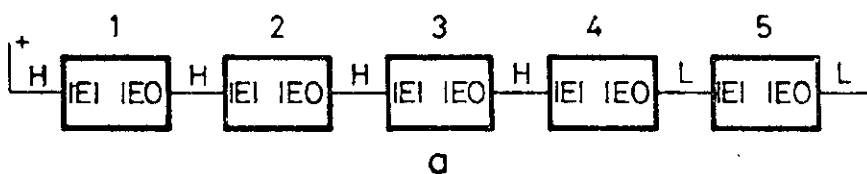
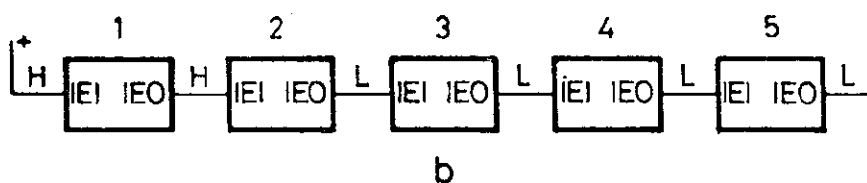


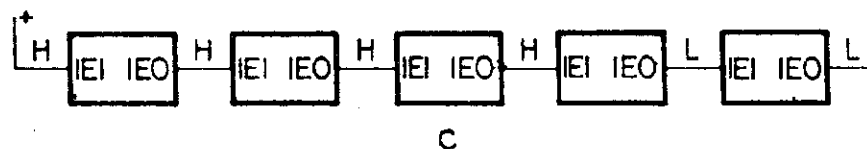
Fig. 4.39. Ciclul de revenire din întrerupere.



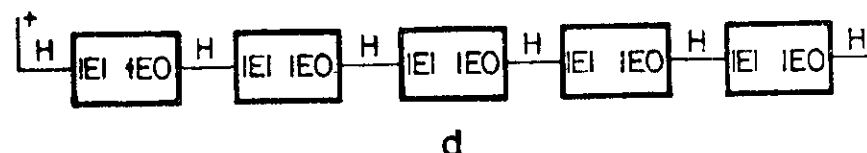
a



b



c



d

H = nivel ridicat
L = nivel coborât

Fig. 4.40. Imbricarea cererilor de întrerupere.

a) întrerupere în curs de servire la elementul 4; b) întrerupere recunoscută și în curs de servire la elementul 2 și suspendarea servirii întreruperii cerute la elementul 4; c) revenire la tratarea cererii de întrerupere cerute de elementul 4; d) absența cererilor de tratare a întreruperilor.

4.10. Circuitul Contor-temporizator-CTC

Circuitul Contor-temporizator (CTC) reprezintă un dispozitiv programabil cu patru canale, care asigură funcțiunile de contorizare și temporizare pentru unitatea centrală de prelucrare Z80. Sub controlul programat al UCP, circuitul CTC poate fi configurat la nivel de canal independent pentru a lucra în diverse moduri.

CTC este realizat în tehnologia NMOS, pe o pastilă cu 28 terminale, cu o singură sursă de alimentare de +5 V și cu un ceas monofazic cu amplitudinea de +5 V. Cele patru canale se pot programa independent pentru a funcționa ca numărătoare pe 8 biți sau în calitate de canale de temporizare pe 16 biți.

Schema bloc a CTC este dată în figura 4.41 și constă din elementele de cuplare cu magistrala UCP, logica internă de comandă, patru canale numărătoare și logica de comandă a întreruperilor. Fiecare canal posedă un vector de întrerupere propriu, prioritatea cea mai mare avînd-o canalul zero.

Schema canalului cuprinde două registre, două numărătoare și logica de comandă (fig. 4.42). Unul din registre este folosit pentru a memora o constantă

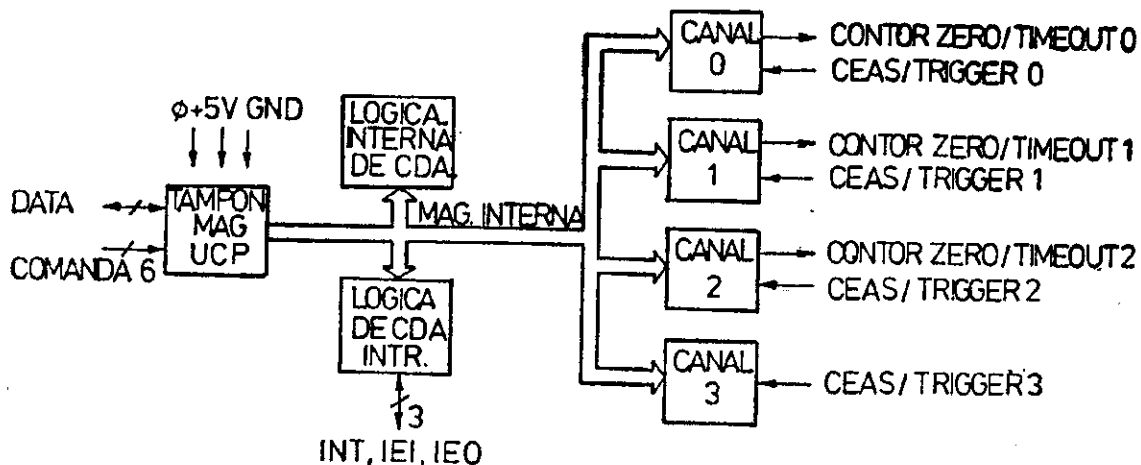


Fig. 4.41. Schema bloc a C.T.C.

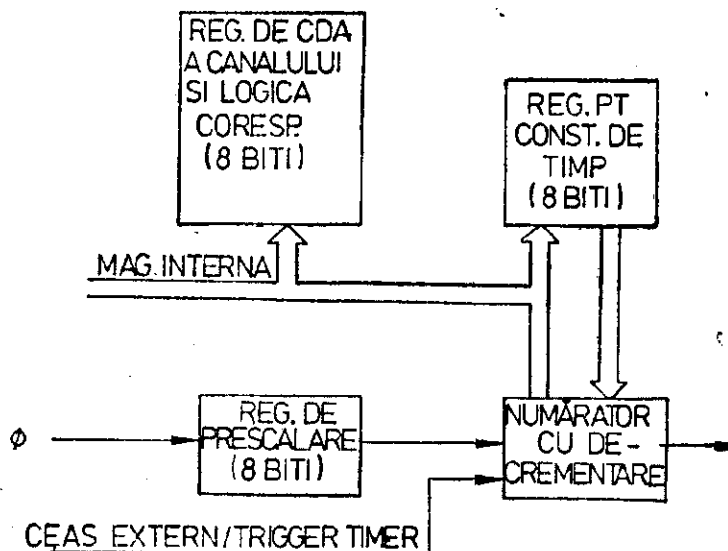


Fig. 4.42. Schema bloc a unui canal.

de timp, de opt biți, iar celălalt asigură comanda canalului. Unul din numărătoare este utilizat în regim de decrementare, cu posibilitatea de citire a conținutului, în timp ce al doilea numărător, de opt biți, se folosește pentru prescalare, divizând frecvența ceasului, fie cu 16, fie cu 256, conform programării.

Registrul care memorează constanta de timp (8 biți) este încărcat de UCP pentru a inițializa și reîncărca numărătorul cu decrementare.

Registrul de comandă a canalului (8 biți) este încărcat de UCP, pentru a selecta modul și condițiile de funcționare ale canalului.

Numărătorul cu decrementare (8 biți) este încărcat, cu conținutul registrului pentru constanta de timp, sub controlul programului, automat, la trecerea prin zero. Conținutul său poate fi citit în orice moment de către UCP. Numărătorul este decrementat de către registrul de prescalare, în modul „timer”, și de către CEAS/TRIG, în modul numărător (contor).

Numărătorul de prescalare (8 biți) divizează ceasul sistemului cu 16 sau 256, pentru comanda numărătorului cu decrementare. Este folosit în modul „timer”.

Descrierea funcțiilor terminalelor CTC (fig. 4.43).

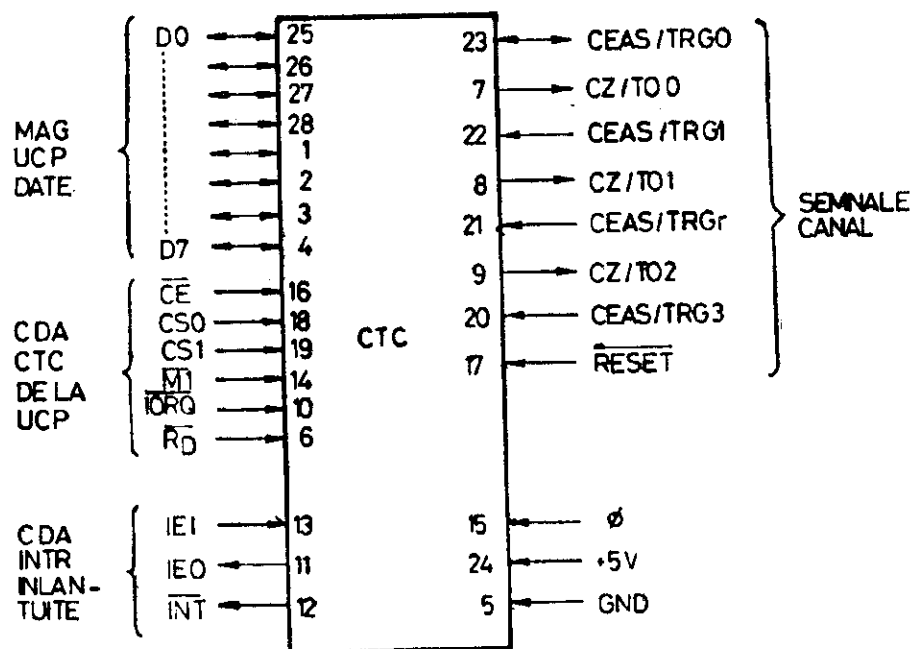


Fig. 4.43. Terminalele C.T.C.

CEAS/TRG0...CEAS/TRG3 : intrări de ceas extern sau de trigger de temporizare.

CZ/TO0...CZ/TO2 : ieșiri active pe nivel ridicat, specifică trecerea prin zero a numărătorului sau „timeout-ul.”

CS0...CS1 : intrări active pe nivel ridicat pentru selecția canalului.

D0—D7 : intrări/ieșiri pentru magistrala de date a UCP, active pe nivel ridicat, cu posibilitatea intrării în starea de mare impedanță.

CE intrare de activare a circuitului (activă pe nivel coborât).

Φ intrare de ceas.

$\overline{M1}$ intrare care specifică ciclul mașină M1, activă pe nivel coborât.

\overline{IORQ} intrare care specifică o cerere de I/E din partea UCP, activ pe nivel coborât.

\overline{RD} intrare activă pe nivel coborât, specifică o cerere de citire din partea UCP.

\overline{IEI} intrare de activare a întreruperilor (activă pe nivel ridicat).

\overline{IEO} ieșire de activare a întreruperilor (activă pe nivel ridicat); împreună cu \overline{IEI} formează un lanț, pentru comanda prioritară a întreruperilor.

\overline{INT} ieșire activă pe nivel coborât reprezentând cererea de întrerupere, furnizată de un tranzistor cu colectorul neconectat.

\overline{RESET} intrare activă pe nivel coborât, blochează numărarea în toate canalele, dezactivează circuitele de întrerupere la nivelul canalelor. Pe durata perioadei de inițializare (\overline{RESET}) ieșirile CZ/TO0-2 și \overline{INT} devin inactive; \overline{IEO} reflectă starea lui \overline{IEI} și circuitele de ieșire ale magistralei de date trec în starea de mare impedanță.

Funcționarea CTC are loc în cadrul mai multor cicluri de lucru : de scriere, de citire, de recunoaștere a unei întreruperi, de revenire din întrerupere etc.

În cadrul ciclului de scriere, în CTC se încarcă : cuvântul de comandă a canalului, constanta de timp și vectorul de întrerupere.

Ciclul de citire este folosit pentru a aduce în UCP conținutul numărătorului cu decrementare, în modul de lucru numărător. Valoarea citită corespunde numărului de fronturi crescătoare ale semnalului extern de ceas, până la începutul perioadei T2, în cadrul execuției operației de citire.

Ciclurile de recunoaștere a întreruperii și revenirii din întrerupere sînt asemănătoare cu cele ale circuitelor PIO și SIO. Întreruperile la nivelurile canalelor sînt înlănțuite, cu prioritatea cea mai mare acordată canalului zero.

Numărarea și temporizarea

În modul numărător fronturile crescătoare sau căzătoare ale ceasului decrementează numărătorul. Frontul este detectat asincron și trebuie să corespundă, ca durată minimă, cu durata impulsului de ceas. Numărătorul este sincron cu Φ , ceea ce impune un interval de timp prestabilit pentru a începe decrementarea numărătorului pe frontul crescător al următorului semnal de ceas.

În modul „timer” numărătorul de prescalare trebuie să fie activat de fronturile crescător sau căzător ale intrării TRG. Frontul este detectat asincron și trebuie să aibă o durată minimă egală, cel puțin, cu durata impulsului TRG.

Dacă temporizarea trebuie să înceapă la următorul front crescător al semnalului Φ , trebuie să se aloce un timp de stabilire corespunzător. Numărătorul de prescalare funcționează pe fronturile crescătoare ale lui Φ .

Programarea CTC

Seleționarea modului de operare impune folosirea unui cuvânt de comandă cu bitul D0 egal cu unu, pentru a specifica încărcarea registrului de comandă

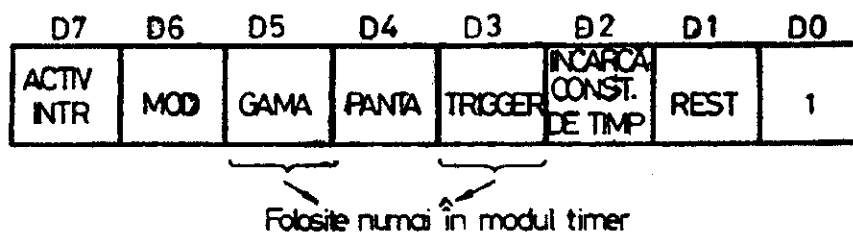


Fig. 4.44. Structura cuvîntului de selecție a modului de operare a C.T.C.

al canalului (fig. 4.44). Semnificația biților din cuvîntul de comandă este dată mai jos :

$D7=0$ — dezactivează întreprerile canalului respectiv.

$D7=1$ — activează întreprerile, pentru a fi generate cînd număratoru este decrementat la zero.

$D6=0$ — stabilește modul „timer“, numărătorul fiind decrementat de către numărătorul de prescalare.

Perioada numărătorului este dată de formula :

$$T = t_c \cdot P \cdot CT$$

unde :

t_c este perioada ceasului sistemului,

P — factor de prescalare (16 sau 256),

CT — constata de timp, de 8 biți, (nmax. 256).

$D6=1$ — stabilește modul numărător, decrementarea fiind realizată de ceasul extern, fără a se utiliza prescalarea.

$D5=0$ — numai în modul „timer“ ceasul sistemului Φ este divizat cu 16, în numărătorul de prescalare.

$D5=1$ — numai în modul „timer“, ceasul sistemului Φ este divizat cu 256, în numărătorul de prescalare.

$D4=0$ — în modul „timer“ frontul negativ al semnalului trigger amorsează operarea ; în modul numărare contorul este decrementat pe fronturile negative.

$D4=1$ — în modul „timer“ frontul pozitiv al semnalului trigger amorsează operarea ; în modul numărare contorul este decrementat pe fronturile pozitive.

$D3=0$ — numai în modul „timer“, „timer“-ul începe operarea pe frontul crescător al perioadei $T2$, în ciclul mașină, care urmează după încărcarea constantei de timp.

$D3=1$ — numai în modul „timer“, un „trigger“ extern este validat pentru amorsarea operării „timer-ului“, după frontul crescător al perioadei $T2$, din ciclul mașină, care urmează după încărcarea constantei de timp. Numărătorul de prescalare este decrementat cu două cicluri de ceas mai tîrziu, dacă timpul de stabilire este ales corespunzător, în caz contrar decrementarea începe cu trei cicluri mai tîrziu.

$D2=0$ — după cuvîntul de comandă al canalului nu va urma constanta de timp. Pentru a iniția funcționarea canalului, trebuie să fie înscrisă o constantă de timp egală cu unu.

$D2=1$ — constanta de timp pentru numărătorul cu decrementare va fi reprezentată de următorul cuvînt înscris în canalul selectat. Dacă în timpul operării

canalului se înscrie o nouă constantă de timp, acțiunea curentă se va continua până la finalizare, după care noua valoare a constantei de timp va fi înscrisă în contorul cu decrementare.

$D1=0$ — canalul continuă numărarea.

$D1=1$ — operația se blochează. Dacă $D2=1$, canalul va termina operarea după încărcarea unei constante de timp, în caz contrar se va încărca un nou cuvânt de comandă.

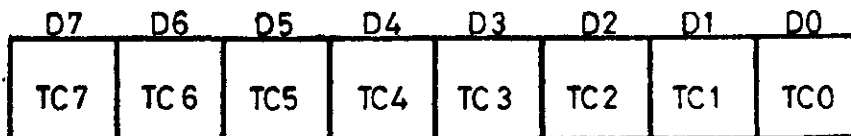


Fig. 4.45. Structura cuvântului constanta de timp.

Încărcarea constantei de timp (fig. 4.45) în registrul corespunzător al canalului se realizează după ce a avut loc încărcarea cuvântului de comandă cu bitul doi poziționat în unu. O constantă de timp egală cu 156 corespunde unui cuvânt cu toți biții egali cu zero.

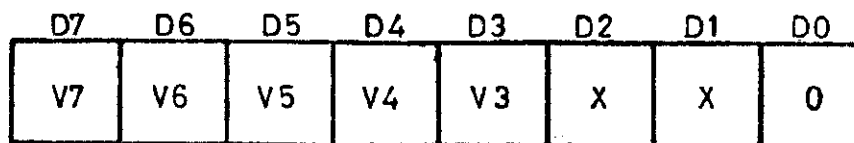


Fig. 4.46. Starea cuvântului vector de întrerupere.

Încărcarea vectorului de întrerupere se realizează prin înscrierea în canalul zero, a cuvântului corespunzător (fig. 4.46). Bitul $D0$ va fi egal cu zero, biții $D7-D3$ conțin vectorul de întrerupere, $D2$ și $D1$ nu sînt utilizați. Cînd CTC răspunde la o recunoaștere de întrerupere biții $D2$ și $D1$ conțin codul binar al canalului cu prioritatea cea mai mare, care a solicitat întreruperea, iar $D0$ este egal cu zero deoarece adresa unei rutine de tratare a întreruperii începe cu un octet par.

5.1. Monitorul V0.1. *)

5.1.1. **Prezentare generală.** La pornirea calculatorului personal este lansat automat în execuție un program de bază, Monitorul, care deține controlul sistemului, permițând utilizatorului introducerea de comenzi de la tastatură. Dacă se reprezintă ansamblul hardware-software al microcalculatorului, sub forma unor cercuri concentrice (fig. 5.1), atunci în cercul din centru se află mașina de bază (hardware), apoi urmează pe primul nivel monitorul. Pe nivelurile următoare se află interpretorul de BASIC și programele de aplicații.

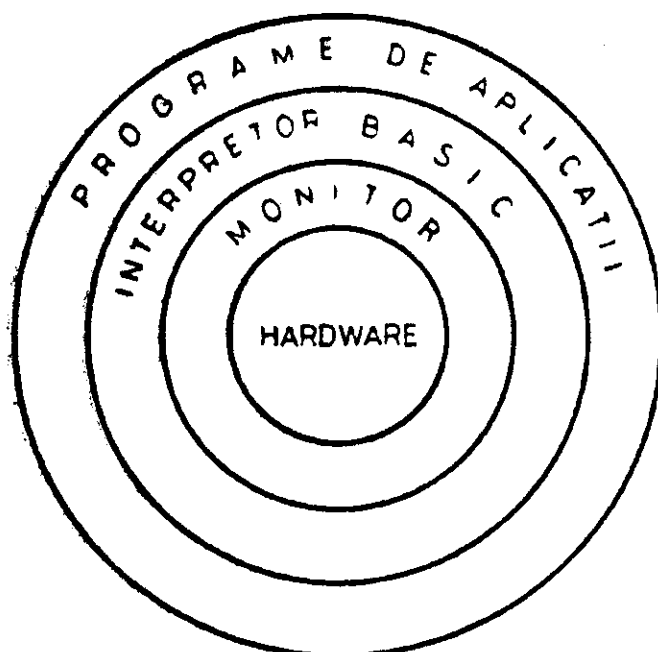


Fig. 5.1. Ierarhia sistemului.

Monitorul asigură primul nivel, cel mai de jos, de interfață cu utilizatorul. De asemenea, asigură interfața mașinii de bază cu interpretorul de BASIC. Monitorul conține subrutinele de intrare/ieșire, pentru echipamentele periferice

*) Exemplele de programe sînt date cu instrucțiuni din repertoriul microprocesorului 8080, (repertoriul este dat în § 7.6) folosindu-se mnemonicele corespunzătoare.

interfațate la microcalculatorul personal: televizor, tastatură și casetofon audio. Utilizarea acestor subrutine simplifică foarte mult transferurile de intrare/ieșire din programele utilizator.

La lansarea în execuție a monitorului, ecranul este șters, sînt inițializate registrele interne de lucru ale utilizatorului și se afișează în partea de sus stînga, pe primul rînd alfanumeric, mesajul 'AMIC', care reprezintă numele acestui program de bază. Pe rîndul următor se afișează caracterul '.', indicînd faptul că sistemul așteaptă introducerea unei comenzi. Comenzile de monitor realizează: afișarea și modificarea unor zone de memorie, lansarea în execuție a programelor, afișarea și modificarea registrelor interne ale microprocesorului și lucrul cu casetofonul audio. Comenzile implementate în această versiune de monitor sînt date în continuare:

- B (BASIC): lansează în execuție interpretorul de BASIC;
- C (Change): modifică registrele interne ale utilizatorului;
- D (Display): afișează pe ecranul televizorului conținutul unei zone de memorie;
- F (Fill): umple o zonă de memorie cu o constantă;
- G (Go): lansează în execuție un program din memoria sistemului;
- K (Cassette): salvează un fișier pe casetă magnetică;
- L (Load): citește un fișier de pe casetă în memorie;
- M (Move): mută o zonă de memorie;
- S (Substitute): afișează și modifică locații din memorie;
- X (Examine): afișează conținutul registrelor interne ale utilizatorului.

Unele dintre aceste comenzi necesită parametri numerici (adrese sau constante pe un octet). Fiecare parametru numeric de tip adresă se introduce de la consolă prin patru cifre hexazecimale și, de asemenea, fiecare parametru de tip constantă pe un octet, se introduce prin două cifre hexazecimale.

Aceste comenzi reprezintă un minim necesar pentru introducerea de programe în cod obiect, pentru lansarea și depanarea acestora. Monitorul aMIG versiunea 1 ocupă 2 Ko de memorie EPROM, între adresele 0000H—07FFH. El se găsește într-un circuit 2716. În cei 2 Ko se găsește și generatorul de caractere, sub forma unei tablele ce cuprinde, pentru fiecare caracter afișabil, cîte un set de șase octeți. Generatorul este implementat pentru:

- 26 litere, de la A la Z;
- 10 cifre, de la 0 la 9;
- 28 caractere speciale.

Structura unui caracter se bazează pe o matrice de 8×8 puncte (8 linii și 8 coloane), din care zona utilă este de 5×6 puncte. Prima coloană și ultimele două din matricea de bază reprezintă separatori de caractere, iar prima și ultima linie sînt folosite ca separatori de rînduri. În figura 5.2 se prezintă un exemplu pentru litera A. Cei 6 octeți utilizați pentru generarea acestui caracter sînt:

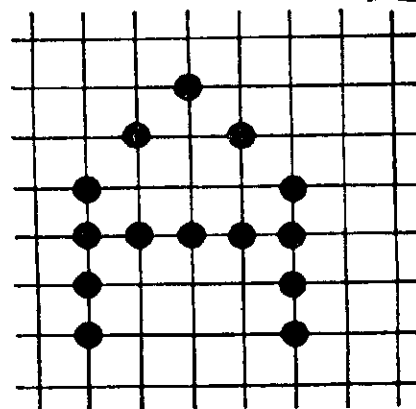


Fig. 5.2. Generarea caracterului A.

10H, 28H, 44H, 7CH, 44H, 44H (în ordinea liniilor TV). Punctul aprins s-a reprezentat prin 1, dar înainte de înscriere în memoria ecran, datele citite din tabelul generatorului sînt complementate.

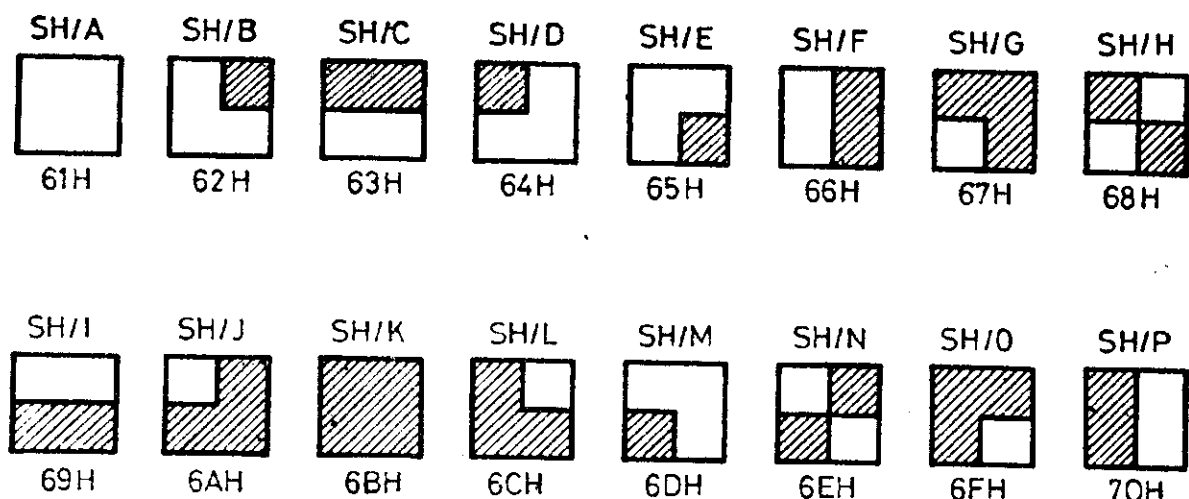


Fig. 5.3. Caracterele semigrafice.

Monitorul permite, de asemenea, afișarea unui set de 16 caractere semi-grafice. Codurile ASCII ale acestor caractere sînt cuprinse între 61H și 70H, iar prezentarea lor este făcută în figura 5.3. Un caracter semigrafic se bazează, de asemenea, pe o matrice de 8×8 puncte, iar dimensiunile unui pixel elementar sînt de 4×4 puncte.

Comenzile monitorului se introduc de la tastatură într-un tampon de intrare. Lungimea maximă a tamponului este de 17 caractere, luînd în considerare comanda, cu cele mai multe caractere, care este M (Move). Toate comenzile se încheie cu RETURN (codul ASCII 0DH). Numai după introducerea acestui caracter tamponul de comandă este interpretat de monitor.

Înainte de încheierea comenzii, prin introducerea caracterului RETURN, tamponul de intrare se poate corecta cu ajutorul tastei DEL (Delete). O apăsare a acestei taste produce ștergerea ultimului caracter introdus în tampon, iar modificarea apare pe ecran.

Monitorul prezintă facilitatea de afișare în video invers, la nivel de caracter, prin introducerea de la tastatură a caracterului CTRL/E (05H).

De asemenea, monitorul conține o serie de subrutine de I/E care pot fi utile programatorului. Aceste subrutine împreună cu adresele de început sînt date în continuare:

— CIN (Console input-07FDH), citește un caracter de la tastatură și îl furnizează în registrul A.

— COUT (Console output-07FAH), trimite la display caracterul conținut de registrul C (codul ASCII) și îl afișează în poziția curentă a cursorului pe ecran.

— KIN (cassette input — 07F7H), citește de pe casetă un fișier, în memoria microcalculatorului, la adresa de la care a fost salvat.

— KOUT (cassette output-07F4H), înscrie pe casetă un fișier din memoria calculatorului (image de memorie). Parametrii de intrare sînt: perechea de registre H, L, care reprezintă adresa de început a zonei de memorie, ce se salvează pe casetă, și perechea de registre D, E, care reprezintă numărul total de octeți.

Pentru citirea unui caracter de la consolă, se utilizează următoarea secvență :

```
CIN EQU 7FDH ; subrutina „Console Input“ din monitor
```

```
·
```

```
CALL CIN ; recepționează caracterul în registrul A
```

Pentru înscrierea pe ecran a unui caracter, în poziția curentă a cursorului, codul ASCII al caracterului de tipărit este încărcat în registrul C și se apelează subrutina „Console Output“. Poziția cursorului este incrementată.

```
COUT EQU 7FAH ; subrutina Console Output din monitor
```

```
·
```

```
MVI C,41H ; exemplu pentru caracterul A
```

```
CALL COUT ; trimite la display
```

```
·
```

Pentru salvarea pe casetă magnetică a unui program obiect din memoria microcalculatorului, se încarcă în perechea de registre H și L adresa de început a zonei, iar în perechea D și E lungimea (numărul de octeți). Apoi se apelează subrutina „Cassette output“. Înainte de executarea subrutinei casetofonului trebuie să fie pornit în modul înregistrare.

```
KOUT EQU 7F4H ; subrutina „Cassette output“ din monitor
```

```
·
```

```
LXI H,0B000H ; exemplu pentru salvarea a 256 de octeți
```

```
LXI D,100H ; de la adresa de memorie 0B000H
```

```
CALL KOUT ;
```

Pentru citirea unui fișier de pe casetă în memoria RAM se apelează subrutina „Cassette input.“ La intrarea în subrutină casetofonul trebuie să fie pornit în modul redare, iar capul de citire să se găsească poziționat pe preambulul fișierului.

```
KIN EQU 7F7H ; subrutina „Cassette input“ din monitor
```

```
·
```

```
CALL KIN ; citește fișier de pe casetă
```

La revenirea din subrutină, în patru locații fixe de memorie, se găsesc următoarele informații în legătură cu fișierul citit :

6023H : octetul inferior al contorului (lungimea fișierului) ;

6024H : octetul superior al contorului ;

6025H : octetul inferior al adresei de încărcare a fișierului ;

6026H : octetul superior al adresei de încărcare.

De asemenea, există o serie de locații de memorie aflate la adrese fixe conținând informații care pot fi utile unui program de aplicații :

6000H : numărul rândului alfanumeric, în care se află poziționat cursorul pe ecran. Valoarea este cuprinsă între 00H și 1FH (00H corespunde primului rând de caractere, iar 1FH corespunde la al 32-lea rând).

6001H : numărul coloanei în care se află poziționat cursorul pe ecran. Valoarea este cuprinsă între 00H și 1DH (00H corespunde primei coloane, iar 1DH corespunde la a 30-a coloană). Aceste două locații de memorie indică poziția pe ecran în care se va înscrie un caracter apelând subrutina COUT.

6002H : modul de afișare la televizor. Dacă această locație conține valoarea 00H modul este defilare, iar dacă locația conține o valoare diferită de 0, modul este pagină.

6003H : video normal/invers la nivelul întregului ecran : 00H pentru video normal și FFH pentru video invers. Comutarea de pe un mod pe celălalt se mai poate face înscriind în portul C al circuitului 8255 (adresa de I/E : 22H) un octet având în bitul 5 valoarea 1 logic pentru video invers și 0 logic pentru video normal.

6004H : video normal/invers la nivel de caracter : 00H pentru video normal și FFH pentru video invers. Comutarea polarității se poate face și prin apelarea subrutinei COUT, având în registrul C valoarea 05H (CTRL/E).

5.1.2. Comenzile monitorului

Comanda B (BASIC)

Format :

• B <return>

Comanda B lansează în execuție interpretorul de BASIC aflat în memoria EPROM în zona 0800H-27FFH (varianta simplă de 8 Kocteți), respectiv 0800H-3FFFH (varianta complexă de 14 Kocteți). Această comandă este echivalentă cu o comandă G având ca parametru adresa 0800H.

Comanda C (Change)

Format :

• C <return>

Comanda C oferă posibilitatea utilizatorului să modifice registrele interne. Modificarea registrelor se efectuează în ordinea A, F, B, C, D, E, H, L, SP (octetul mai semnificativ), SP (octetul mai puțin semnificativ), PC (octetul mai semnificativ), PC (octetul mai puțin semnificativ).

După introducerea comenzii se afișează conținutul primului registru (A), urmat de liniuță și la fel ca la comanda S (Substitute) utilizatorul are posibili-

tatea să modifice conținutul registrului prin introducerea noii valori sau introducând „blanc” se trece la registrul următor. Se modifică în acest fel maximum 12 octeți. Comanda se încheie cu <return>.

Comanda D (Display)

Format :

- D <adr1>, <adr2> <return>

Comanda D afișează pe ecranul televizorului conținutul unei zone de memorie cuprinsă între adresele <adr1> și <adr2>. Pe fiecare rând alfanumeric se afișează opt octeți, fiecare octet reprezentat prin două cifre hexazecimale. La începutul rândului se afișează adresa primului octet din grup, aceasta fiind întotdeauna multiplu de 8.

Comanda F (Fill)

Format :

- F <adr1>, <adr2>, <const> <return>

Comanda F umple o zonă de memorie RAM, cuprinsă între adresele <adr1> și <adr2>, cu o constantă <const>, reprezentată pe un octet.

Comanda G (Go)

Format :

- G <adr1>, [<adr2>] <return>

Comanda G lansează în execuție un program utilizator aflat în memoria microcalculatorului și are două forme.

Dacă se introduce un parametru, care este o adresă, atunci în contorul de program este încărcată această valoare. În acest fel se poate lansa în execuție un program aflat oriunde în memoria calculatorului personal.

Dacă se introduc doi parametri, despărțiți prin virgulă, atunci primul parametru reprezintă adresa care se va încărca în contorul programului, deci adresa de lansare în execuție, iar al doilea parametru reprezintă adresa punctului de întrerupere. Prin executarea unei comenzi G cu punct de întrerupere, la adresa indicată de al doilea parametru numeric (adresa punctului de întrerupere) se salvează octetul din program, înlocuindu-se cu valoarea CFH (codul instrucțiunii RST 1), apoi valorile registrelor utilizator A, F, B, C, D, E, H, L, SP sînt încărcate în registrele fizice ale microprocesorului, iar în contorul de program se încarcă primul parametru numeric al comenzii (adresa de lansare a programului utilizator).

În momentul în care execuția programului utilizator ajunge în punctul de întrerupere, este decodificată instrucțiunea RST 1 și se face un apel de subrutină cu adresa 0008H, unde se găsește secvența de tratare a punctului de întrerupere. Această secvență salvează starea registrelor microprocesorului în zona registrelor utilizator (sînt 12 octeți de RAM) și reface octetul inițial din program utilizator, de la adresa punctului de întrerupere.

Această facilitate permite rularea controlată a unui program aflat în RAM, eventual pe secvențe scurte, în scopul depanării și punerii la punct. După fie-

care secvență utilizatorul are posibilitatea să vizualizeze registrele și zone de memorie sau să modifice starea programului.

Comanda K (Cassette)

Format :

- K <adr1>, <adr2> <return>

Comanda K salvează pe casetă magnetică o zonă de memorie cuprinsă între adresele <adr1> și <adr2>, unde se află un program în cod obiect. Utilizatorul înregistrează la începutul fișierului, numele programului și eventual câteva explicații, utilizând microfonul casetofonului. Apoi se lansează comanda, observându-se pe ecranul televizorului o desincronizare a imaginii. Aproximativ în primele 10 secunde are loc înscrierea preambulului, după care se salvează informația utilă : adresa de început a zonei de memorie, contorul, octeții de informație și suma ciclică. Salvarea pe casetă magnetică a unei zone de 1 Ko din memoria RAM durează aproximativ 4,5 secunde, acest timp depinzând de raportul dintre numărul de biți 1 logic și numărul de biți 0 logic.

Comanda L (Load)

Format :

- L <return>

Comanda L citește un fișier de pe casetă în memoria microcalculatorului personal. Încărcarea programului se face la adresa de unde acesta a fost salvat prin comanda K.

Pentru realizarea operației de citire se poziționează caseta pe începutul de fișier (preambul) se introduce comanda de la tastatură, se pornește casetofonul în regim de redare și imediat se apasă pe tasta <return>, având grijă ca în momentul apăsării capul de citire să nu treacă de zona de început de fișier.

Dacă citirea întregului fișier s-a făcut corect atunci se afișează la display adresa de început și lungimea programului încărcat. În cazul apariției unei erori, detectată prin faptul că suma de control nu se verifică, se afișează mesajul '?', caz în care fișierul trebuie citit din nou.

Comanda M (Move)

Format :

- M <adr1>, <adr2>, <adr3> <return>

Comanda M mută o zonă de memorie RAM/EPROM într-o altă zonă de memorie RAM. Se introduc trei parametri, care au următoarele semnificații :

- <adr1> : adresa de început a zonei sursă ;
- <adr2> : adresa de sfârșit a zonei sursă ;
- <adr3> : adresa de început a zonei destinație.

Comanda se încheie prin apăsarea tastei <return>. Operația are loc fără modificarea zonei sursă.

Comanda S (Substitute)

Format :

- S <adr> <return>

Comanda S afișează și permite modificarea locațiilor dintr-o zonă de memorie RAM cu adresa de început <adr>. Conținutul fiecărei locații este afișat la display sub forma a două cifre hexazecimale (un octet) urmate de liniuță. Utilizatorul are posibilitatea să modifice locația curentă prin introducerea noii valori urmată de blanc (spațiu), pentru afișarea locației următoare, sau direct blanc, caz în care locația curentă rămâne nemodificată. La modificarea unei locații, comanda S permite introducerea noii valori nu numai cu două cifre hexazecimale, dar și cu o singură cifră, caz în care se consideră ca cifră mai puțin semnificativă, cifra mai semnificativă fiind 0 sau cu mai multe cifre, caz în care se iau în considerare numai ultimele două cifre introduse.

Comanda X (Examine)

Format :

- X <return>

Comanda X afișează pe ecranul televizorului conținutul registrelor utilizator sub forma :

AF	BC	DE	HL	SP	PC
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

unde :

- A reprezintă registrul acumulator,
- F reprezintă registrul indicatorilor de condiție,
- B, C, D, E, H, L, sînt registrele generale de lucru,
- SP este indicatorul vârfului stivei,
- PC este contorul de program,
- X reprezintă o cifră hexazecimală,

5.1.3. Exemple de utilizare

Exemplul 1 : realizarea unei anumite configurații a zonei de memorie RAM cuprinsă între adresele A100H și A11FH, astfel încît fiecare locație să conțină octetul inferior de adresă al locației respective. În acest exemplu s-a presupus că inițial în memorie există o configurație oarecare de octeți :

. DA100, A11F <return>

```
A100 00 15 2C FF 00 3C 27 11
A108 15 21 A4 32 22 1A CC 3E
A110 54 52 AA 55 01 44 32 7C
A118 24 80 96 EF 25 30 4C D2
```

. SA100 <return>

```
00-15-1 2C-2 FF-3 00-4 3C-5 27-6 11-7 15-8 21-9 A4-A 32-B
22-C 1A-D CC-E 3E-F 54-10 52-11 AA-12 55-13 01-14 44-15 32-16
7C-17 24-18 80-19 96-1A EF-1B 25-1C 30-1D 4C-1E D2-1F <return>
```

. DA100, A11F <return>

```
A100 00 01 02 03 04 05 06 07
A108 08 09 0A 0B 0C 0D 0E 0F
A110 10 11 12 13 14 15 16 17
A118 18 19 1A 1B 1C 1D 1E 1F
```


Exemplul 2 : modificarea registrelor utilizator B, C, H și SP.

. X <return>

```

AF  BC  DE  HL  SP  PC
0000 0000 0000 0000 A100 0000

```

. C <return>

```

00-00-00-FF 00-20 00-00-00-A100-00-B0 <return>

```

. X <return>

```

AF  BC  DE] HL  SP  [PC
0000 FF20 0000 A100]B000 0000

```

Exemplul 3 : umplerea zonei de memorie RAM cuprinsă între adresele B200H și BFFFH cu valoarea constantă C7H.

. FB200, BFFF, C7 <return>

Această operație se poate executa și prin comenzile S și M :

. SB200 <return>

21-C7 <return>

. MB200, BFFE, B201 <return>

Exemplul 4 : introducerea în memoria calculatorului la adresa A100H a unui program obiect și lansarea lui în execuție.

Programul care va fi introdus este următorul :

```

          ORG  A100H
START:   MVI  A, 20H      ; A=codul ASCII pentru blank
BUCLA:   MOV  C, A       ; C=codul ASCII pentru caracterul de tipărit
          CALL COUT      ; se trimite la consolă
          INR  A         ; se trece la caracterul următor
          CPI  60H       ; s-au terminat caracterele ?
          JNZ  BUCLA     ; dacă nu, reluare
          JMP  START     ; reluare program
COUT     EQU  7FAH
          END

```

Acest program realizează afișarea continuă la display a caracterelor având codurile cuprinse între 20H și 5FH. Programul obiect care trebuie introdus, împreună cu adresele absolute de memorie sînt listate în continuare :

```

A100 3E 20
A102 4F
A103 CD FA 0
A106 3C
A107 FE 60
A109 C2 02 A1
A10C C3 00 A1

```

Sînt utilizate următoarele comenzi de monitor :

. DA100, A10E <return>

```

A100 C1 3E 27 4A FF 23 15 06
A108 8A 42 15 FF FF FF FF

```

. SA100 <return>

```

C1-3E 3E-20 27-4F 4A-CD FF-FA 23-07 15-3C 06-FE 8A-60
42-C2 15-02 FF-A1 FF-C3 FF-00 FF-A1 <return>

```

```
. DA100, A10E <return>
A100 3E 30 4F CD FA 07 3C FE
A108 60 C2 02 A1 C3 00 A1
. GA100 <return>
```

Exemplul 5 : considerînd programul în cod obiect, de la exemplul 4, aflat în memorie la adresa A100H, se salvează pe casetă magnetică, iar după un timp programul este reincărcat și lansat în execuție.

```
. DA100, A10E <return>
A100 3E 20 4F CD FA 07 3C FE
A108 60 C2 02 A1 C3 00 A1
. KA100, A10E <return>
. . . . .
. L <return>
A100
000F
. DA100. A10E!<return>
A100 3E 20 4F CD FA 07 3C FE
A108 60 C2 02 A1 C3 00 A1
. GA100 <return>
```

5.2. Monitorul MON. aMIC V0.2

5.2.1. **Prezentare generală.** Monitorul MON.aMIC V 0.2 reprezintă o versiune extinsă a monitorului aMIC prezentată în §.5.1 asigurînd accesul utilizatorilor la resursele microcalculatorului aMIC.

Caracteristicile principale ale monitorului MON.aMIC V 0.2 sînt următoarele :

- utilizarea instrucțiunilor specifice microprocesorului Z80, cu care este echipat microcalculatorul aMIC, în scopul creșterii vitezei de execuție a rutinelor monitor și condensării codului obiect al acestora ;

- adăugarea de comenzi noi privitoare la citire/scriere de fișiere în format hexa pe interfața serială ;

- posibilitatea de a atribui un nume fișierelor pe caseta magnetică și de a efectua operațiile de citire, scriere și verificare a fișierelor pe baza numelui atribuit ;

- modificarea matricilor de definire a caracterelor, mărind la 40 numărul de caractere afișabile pe un rînd al ecranului TV ;

- introducerea noțiunii de „funcție utilizator“, pentru a permite accesul facil la rutinele monitor de gestionare a perifericelor atașate microcalculatorului aMIC ;

- standardizarea funcțiilor utilizator la nivelul sistemului de operare CP/M V2.2 în scopul portabilizării pe aMIC a programelor dezvoltate sub CP/M pe alte sisteme cu microprocesor.

Perifericele atașate microcalculatorului aMIC sînt privite ca echipamente de tip logic, ce îndeplinesc următoarele funcțiuni :

— consolă — asigură dialogul operatorului cu sistemul, fiindu-i asignată tastatura (intrare date) și ecranul TV (ieșire date) ;

— cititor — asigură intrarea datelor de pe interfața serială ;

— perforator — asigură ieșirea datelor pe interfața serială ;

— listare — asigură afișarea datelor pe miniimprimantă.

În afara acestor echipamente de tip logic, funcțiile utilizator mai permit gestionarea următoarelor periferice :

— casetofon audio — asigură stocarea pe casete magnetice a informație conținută în memoria microcalculatorului ;

— ecran TV în mod grafic — asigură aprinderea, stingerea și testarea stării unui pixel pe ecran ;

— difuzor — asigură generarea de sunete de durată și frecvență programabilă.

Menționăm că asignarea perifericelor fizice la echipamentele logice este rigidă, cu excepția interfeței seriale, care poate prelua funcția de consolă, în vederea utilizării echipamentelor de tip display în locul tastaturii elastice și ecranului TV.

Spațiul de memorie EPROM ocupat de MON.aMIC V 0.2 variază în numărul de funcții utilizator implementate, versiunea V 0.2 avînd aproximativ 2,5 Ko lungime. Deoarece se utilizează circuite 2716 de 2 K octeți pentru memoria EPROM, spațiul pînă la adresa OFFFH este menținut ca rezervă pentru dezvoltări ulterioare ale monitorului, programele aplicative putînd fi implementate în memoria EPROM începînd cu adresa 1000H. Lungimea maximă a unui program aplicativ aflat în meoria EPROM nu poate depăși 12 Ko.

5.2.2. Comenzile monitorului MON. AMIC V0.2 Setul de comenzi puse la dispoziția utilizatorilor de către monitorul MON.AMIC V 0.2 este :

- | | | |
|----------|----------------------|--|
| C | — (compare) : | comparare conținut zone de memorie ; |
| D | — (display memory) : | afișarea la consolă a conținutului unei zone de memorie ; |
| F | — (fill memory) : | umplerea unei zone de memorie cu o constantă ; |
| G | — (go) : | lansarea în execuție a unui program cu sau fără punct de întrerupere ; |
| K | — (casette) : | salvarea unei zone de memorie ca fișier pe casetă magnetică ; |
| L | — (load) : | încărcarea în memorie a unui fișier de pe casetă magnetică ; |
| M | — (move memory) : | transferarea conținutului unei zone de memorie ; |
| N | — (name) : | afișare conținut antet de fișier pe casetă magnetică ; |
| R | — (read) : | citirea unui bloc de date în format hexa de la interfața serială ; |
| S | — (substitute) : | afișarea și modificarea conținutului unei zone de memorie ; |

- V** — (verify) : compararea conținutului unei zone de memorie cu conținutul unui fișier pe casetă magnetică ;
- W** — (write) : scrierea unui bloc de date în format hexa la interfața serială ;
- X** — (examine) : examinarea și modificarea conținutului registrelor microprocesorului Z80.

Parametrii solicitați de o parte dintre comenzi sînt adrese sau constante, fiind introduși de la consolă prin maximum 4 cifre hexazecimale ; de ex. adresa 1A poate fi exprimată sub forma : 1A ; 01A ; 001A ; toate cele trei forme fiind valide.

Comanda C — (compare memory)

Formatul comenzii este :

C<adinf>, <adsup>, <adcomp><CR>

unde :

<adinf> — adresa inferioară a zonei de memorie de comparat ;

<adsup> — adresa superioară a zonei de memorie de comparat ;

<adcomp> — adresa inferioară a zonei de memorie cu care se face compararea.

Compararea se execută începînd de la adresele <adinf> și <adcomp>, pînă la atingerea adresei <adsup>. Dacă nu există diferențe, se revine în starea de așteptare comandă de la operator, prin afișarea prompterului ".".

Dacă există diferențe, afișarea lor se execută sub forma :

XXXX YY ZZ

In care :

XXXX — adresa de memorie din cadrul zonei de comparat

YY — conținutul octetului de la adresa XXXX

ZZ — conținutul octetului de la adresa <adcomp> + (XXXX - <adinf>), din zona cu care se execută compararea.

Afișarea diferențelor continuă pînă la atingerea adresei <adsup> ; dacă se dorește întreruperea afișării, se apasă pe tasta INT.

Comanda D — (display memory)

Formatul comenzii este :

D<adinf>, <adsup><CR>

unde :

<adinf> — adresa inferioară a zonei de afișat ;

<adsup> — adresa superioară a zonei de afișat.

Afișarea se face în format hexazecimal și format ASCII, cîte 8 octeți pe linie pentru fiecare format. La începutul liniei în format hexazecimal se afișează și adresa de memorie a primului octet din linie ; adresa este întotdeauna

una multiplu de 8. Adresa inferioară dată în comandă este rotunjită la primul multiplu de 8 inferior.

Formatul ASCII este util în vizualizarea unor zone de memorie care conțin mesaje, texte sursă etc. Se afișează numai caracterele având codul cuprins între 20H și 60H, restul codurilor fiind înlocuite cu caracterul “.”

Comanda F — (fill memory)

Formatul comenzii este :

F<adinf>, <adsup>, <const> <CR>

unde :

<adinf> — adresa inferioară a zonei de memorie

<adsup> — adresa superioară a zonei de memorie

<const> — valoarea hexa cu care se umple zona specificată

Comanda F servește la umplerea unei zone de memorie RAM cu o constantă.

Comanda G — (go)

Formatul comenzii este :

G[<adlans>] [,<adbrk>]<CR>

unde :

<adlans> — adresa de lansare în execuție a programului utilizator

<adbrk> — adresa punctului de întrerupere a execuției programului.

Comanda G servește la lansarea și urmărirea execuției unui program utilizator.

Dacă nu se introduce nici un parametru (forma G<CR>), se reia execuția programului din punctul în care a fost întrerupt anterior.

Dacă se introduce numai adresa de lansare (forma G<adlans><CR>), se predă controlul programului utilizator începând cu adresa <adlans>.

Dacă se introduce numai adresa punctului de întrerupere (forma G,<adbrk><CR>), se reia execuția programului din punctul unde a fost întrerupt ultima dată, și se continuă pînă la atingerea adresei punctului de întrerupere, moment în care se redă controlul monitorului.

În toate cazurile, înainte de lansarea în execuție a programului utilizator, se reface contextul său (registre și indicatori), permițându-se astfel atît execuția programului fără a fi perturbată de întreruperi, cît și modificarea dorită a contextului de lucru între două lansări succesive.

Tratarea întreruperii software, prin specificarea adresei punctului de întrerupere, se realizează prin introducerea la această adresă a codului corespunzător instrucțiunii RST 38H, a cărei secvență de tratare se află la adresa 0038H. La detectarea codului de RST 38H, se afișează contextul de lucru al programului întrerupt, analog ca la funcția X (vezi 5.2.14), apoi se predă controlul monitorului. Menționăm că atît întreruperea software, cît și întreruperea nemascabilă dată de tasta INT au aceeași tratare, fapt care permite întreruperea unui program prin tasta INT și reluarea execuției prin comanda G<CR>.

Comanda K — (write file on cassette).

Formatul comenzii este :

K<adinf>,<adsup>,<nume><CR>

unde :

<adinf> — adresa inferioară a zonei de memorie de salvat

<adsup> — adresa superioară a zonei de memorie de salvat

<nume> — numele atribuit fișierului pe casetă (max. 4 cifre hexazecimale).

Comanda K salvează pe caseta magnetică o zonă de memorie cuprinsă între adresele adinf și adsup. Se emite mai întâi mesajul :

START CASS., THEN TYPE(CR) :

Monitorul așteaptă apăsarea tastei CR (RETURN), apoi efectuează scrierea informației pe casetă. Operatorul va poziționa înainte de apăsarea tastei CR potențiometrul de volum al casetofonului, în ultima treime a cursei (mai mare decât volumul mediu și mai mic decât volumul maxim) astfel încât la înregistrare, indicatorul de nivel al casetofonului să oscileze doar în porțiunea roșie (nu la limita dintre porțiunea verde și roșie, cum se procedează la înregistrări muzicale obișnuite).

Transferul informației spre casetă se efectuează cu o viteză medie de 1500 baud/s ; cu alte cuvinte, fiecare Ko de memorie necesită aprox. 5 secunde pentru transfer.

Structura informației corespunzătoare unui fișier este următoarea :

- antet de sincronizare
- antet de fișier
- zona de date

Antetul de sincronizare conține 2 tipuri de informații :

— o succesiune de biți astfel aleasă încât să permită detectarea automată prin program a unui început de fișier.

Se elimină astfel necesitatea poziționării manuale a casetei pe începutul unui fișier la operațiile de citire antet, citire fișier sau verificare fișier.

— o succesiune de biți astfel aleasă încât să permită calculul valorii medii de prag care permite diferențierea impulsurilor corespunzătoare valorii de 0 logic de cele corespunzătoare valorii de 1 logic. Astfel devine posibilă auto-reglarea valorii medii de prag în cadrul operațiilor de citire casetă. (v. comanda L)

Antetul de fișier urmează antetului de sincronizare, și conține pe 7 octeți următoarele informații :

- numele fișierului (2 octeți)
- adresa de început a zonei de memorie transferate (2 octeți).
- adresa de sfârșit a zonei de memorie transferate (2 octeți)
- suma de control pentru cei 6 octeți precedenți (1 octet)

Zona de date conține șirul de octeți aflați în spațiul de memorie <adinf>—<adsup>. Fiecare octet este codificat ca o succesiune de 8 impulsuri, un impuls pentru 1 logic avînd durată dublă față de un impuls pentru

0 logic. Zona de date este urmată de 1 octet ce conține suma de control a datelor transferate, necesară verificării datelor la operațiile de citire,

Comanda L — (load file from cassette)

Formatul comenzii este :

L[<nume>] [,<adinf>]<CR>

unde :

<nume> — numele fișierului de încărcat (max. 4 cifre hexazecimale)

<adinf> — adresa inferioară a zonei de memorie de la care începând se va încărca fișierul

Comanda L citește un fișier pe casetă magnetică în memoria microcalculatorului. Distingem 4 moduri de încărcare a unui fișier, în funcție de sintaxa comenzii L :

— L<CR>: încarcă primul fișier identificat după pornirea casetofonului, la adresa la care acesta a fost salvat prin comanda K.

— L,<adinf><CR>: încarcă primul fișier identificat după pornirea casetofonului, la adresa <adinf>.

— L<nume><CR>: încarcă fișierul cu numele <nume> la adresa la care acesta a fost salvat prin comanda K.

— L<nume>,<adinf><CR>: încarcă fișierul cu numele <nume> la adresa <adinf>.

Înainte de a începe operația de identificare fișier, monitorul emite mesajul :

START CASS.,THEN TYPE(CR) :

Spre deosebire de comanda K, apăsarea tastei CR (RETURN) poate fi efectuată în orice moment, indiferent dacă casetofonul este sau nu pornit; operația de identificare a antetului de sincronizare va rămâne în buclă de așteptare pînă la pornirea casetofonului. De asemenea, poziția benzii magnetice în fața capului de redare poate fi oarecare (în zona neînregistrată dintre fișiere sau în mijlocul oricărui fișier), operația de încărcare fișier devenind activă numai după identificarea unui antet de sincronizare.

După identificarea antetului de sincronizare, monitorul citește antetul de fișier, compară datele din antet cu cele specificate în comanda L, și efectuează, dacă e cazul, încărcarea fișierului în memorie.

După citirea antetului de fișier, sau după terminarea operației de încărcare, se emite la consolă mesajul :

EE VV NNNN XXXX YYYY}

În care :

EE — cod de retur

VV — valoare medie de prag

NNNN — nume fișier

XXXX — adresa inferioară a zonei de memorie salvate

YYYY — adresa superioară a zonei de memorie salvate

Codul de retur specifică modul de desfășurare a operației de citire fișier. Valorile și semnificațiile respective sînt:

- 00 — operația terminată normal:
- 01 — numele fișierului citit nu coincide cu numele fișierului specificat în comanda L<nume> sau L<nume>, <adinf>.
- 02 — eroare sumă de control antet de fișier; în acest caz, informațiile din cîmpurile NNNN, XXXX și YYYY nu au nici o semnificație. Aceasta este singura eroare de tip „fatal“, indicînd faptul că informațiile din porțiunea respectivă de casetă sînt incomprehensibile.
- 03 — eroare sumă de control zonă de date; în acest caz informațiile citite în memorie pot fi eronate.
Nu se exclude posibilitatea ca informațiile citite să fie totuși corecte, eroarea provenind din citirea eronată a însuși sumei de control de pe casetă. Se impune verificarea de către utilizator a informației citite.
- 04 — neconcordanță între informația din fișier și informația din memorie (acest cod apare numai la execuția comenzii de verificare V).

Dacă codul de retur este diferit de zero, comanda L rămîne activă în continuare, emițînd mesajul:

NEXT ?

Utilizatorul are posibilitatea să răspundă CR(RETURN) ceea ce are ca efect relansarea întregului ciclu de identificare fișier-citire antet-citire date pentru porțiunea următoare de casetă. Această căutare este utilă pentru comanda de citire fișier cu nume dat, permițînd totodată și vizualizarea antetelor tuturor fișierelor aflate înaintea fișierului căutat.

După emiterea mesajului cu cod de retur diferit de zero, casetofonul poate fi oprit, re poziționat și repornit; funcția de citire rămîne activă în continuare. Dacă codul de retur este zero, mesajul NEXT ? nu [mai este emis, și se revine în starea de așteptare comenzi monitor.

Dacă la mesajul NEXT ? se răspunde cu orice alt caracter diferit de CR, execuția comenzii de citire se termină, monitorul reintrînd în starea de așteptare comenzi.

Valoarea medie de prag este cuprinsă în mod normal în domeniul 18H-1DH. Cu cît valoarea este mai mică, cu atît nivelul de redare al casetofonului este mai mare, și reciproc. Cunoscînd acest fapt, este indicată lansarea unei operații de citire de tip:

LFFFF<CR> (presupunînd că nu s-a înregistrat nici un fișier cu numele FFFF pe casetă), pentru casetele a căror conținut nu se cunoaște apriori; urmărind în paralel fondul sonor și mesajele emise pe ecranul TV, distingem următoarele situații:

— la trecerea de la o porțiune neînregistrată la una înregistrată nu apare nici un mesaj: în acest caz, se va regla potențiometrul de volum în jurul poziției medii, se va re poziționa manual casetofonul, pe porțiunea neînregistrată și se va relua operația de redare, fără nici o intervenție la tastatură.

Dacă după mai multe reluări ale aceleiași porțiuni de bandă nu se obține nici un mesaj pe ecran, informația din porțiunea respectivă nu prezintă o structură standard de fișier și în consecință nu poate fi tratată de comanda de citire.

— la trecerea de la o porțiune neînregistrată la una înregistrată apare un mesaj cu cod de retur 02; în acest caz s-a „prins” o porțiune cu informație aleatoare, confundabilă cu un antet de sincronizare. Se va lăsa casetofonul în funcțiune, pînă la întîlnirea unui nou fișier.

— la trecerea de la o porțiune neînregistrată la una înregistrată apare un mesaj cu cod de retur 01; reglajul de volum este corect.

Dacă valoarea medie de prag este mică (18—19H) se va micșora nivelul de redare din potențiometrul de volum, se va re poziționa manual caseta pe zona neînregistrată ce precede fișierul și se va porni din nou casetofonul. Pentru valori medii de prag mari (1C-1DH) se va proceda invers, mărind nivelul de redare. Prin 2—3 treceri manuale succesive scurte ale începutului de fișier prin fața capului de redare se va stabili poziția optimă la redare a potențiometrului de volum, căutîndu-se obținerea unei valori medii de prag de 1AH-1BH. Menționăm că pe parcursul întregii operații de reglaj amintite se va răspunde cu CR(RETURN) la mesajul NEXT? pentru a rămîne în cadrul comenzii de citire fișier.

Autoreglarea valorii medii de prag se execută separat de către comanda L pentru fiecare fișier identificat; în general variațiile pe parcursul unei casete întregi nu trebuie să depășească ± 1 față de valoarea stabilită prin reglaj manual la începutul casetei.

În caz contrar casetofonul prezintă variații ale vitezei de antrenare în funcție de cantitatea de bandă magnetică de pe rola debitoare.

Pentru a elimina pe cît posibil riscul imposibilității citirii unui fișier salvat pe casetă, se recomandă utilizarea aceluiași casetofon atît la înregistrare cît și la redare.

Codificarea informației solicită un spectru de frecvență care depășește 2KHz, recomandîndu-se utilizarea unor capete de înregistrare/redare noi (uzura acestora reduce nivelul de redare pentru frecvențe ridicate).

Comanda M — (move memory)

Formatul comenzii este:

M<adinf>, <adsup>, <addest> <CR>

unde:

<adinf> — adresa inferioară a zonei de memorie de transferat

<adsup> — adresa superioară a zonei de memorie de transferat

<addest> — adresa de destinație.

Comanda M transferă o zonă de memorie cuprinsă între adresele <adinf> și <adsup> într-o zonă a cărei adresă inferioară este dată de <addest>.

Comanda N — (name).

Formatul comenzii este:

N<CR>

Comanda N permite afișarea antetului primului fișier întâlnit după pornirea casetofonului. Modul de operare al comenzii N este analog cu cel al comenzii L, singura deosebire fiind terminarea execuției comenzii, după citirea antetului de fișier (nu se mai citește zona de date). Codul de retur emis poate avea valorile 00 (terminare normală) sau 02 (eroare sumă de control antet fișier), caz în care se emite mesajul :

NEXT ?

și se așteaptă răspunsul operatorului. Apăsarea tastei CR lasă activă comanda în continuare ; orice alt caracter introdus va termina execuția ei.

Reglarea volumului la redare poate fi făcută analog ca la comanda L ; singura deosebire constă în terminarea automată a comenzii la identificarea corectă a unui antet de fișier.

Comanda R — (read)

Formatul comenzii este următorul :

R[<addep>]<CR>

unde :

<addep> — adresa de deplasare cu care se translatează adresele de încărcare ale fișierului hexa.

Comanda R citește un fișier hexa de la interfața serială, și îl încarcă în memorie. Încărcarea se face fie la adresele specificate de fișierul hexa (forma R<CR>), fie la adrese rezultate ca sumă între <addep> și adresele specificate în fișier. Fiecare înregistrare este controlată prin compararea sumei de control calculate cu cea înscrisă în fișier. În caz de eroare se afișează mesajul :

READ ERR. și se termină execuția comenzii.

Încărcarea normală se termină la identificarea unei înregistrări „sfârșit de fișier“ ; indiferent de conținutul acestei înregistrări, comanda este redată monitorului.

Comanda S — (substitute)

Formatul comenzii este :

S<adr><CR>

unde :

<adr> — adresa primului octet din memorie de la care începând se execută substituirea.

Comanda S permite modificarea conținutului memoriei RAM, octet cu octet, începând de la o adresă specificată. Modificarea se execută astfel :

- monitorul afișează adresa octetului și valoarea sa și rămîne în așteptare,
- dacă se introduc două cifre hexazecimale, conținutul adresei afișate se înlocuiește cu valoarea dată, și se trece la tratarea următorului octet,
- dacă se introduce <CR>, conținutul adresei rămîne nemodificat, și se trece la tratarea următorului octet,

— dacă se introduce caracterul "•", se termină execuția comenzii.
 Introducerea unei cifre hexazecimale eronate are ca efect reluarea afișării adresei și a valorii octetului de modificat. Singurul mod de terminare al comenzii este introducerea caracterului "•"

Comanda V_a — (verify_file)

Formatul comenzii este :

V<nume>,<adinf><CR>

unde :

<nume> — numele fișierului de verificat

<adinf> — adresa inferioară a zonei de memorie de comparat

Comanda V este deosebit de utilă, permițând verificarea modului în care s-a efectuat salvarea unui fișier pe casetă magnetică, fără a fi necesară încărcarea acestuia în memorie.

Compararea se execută între zona de date a fișierului <nume>, și zona de memorie care începe la adresa <adinf>. Compararea se execută pînă la epuizarea citirii fișierului (deci pe lungimea corespunzătoare fișierului salvat).

Codul de retur 00 semnifică egalitatea fișierului cu zona de memorie specificată. Codul de retur 04 indică detectarea unei inegalități; semnificația cîmpurilor din mesajul de eroare se deosebește în acest caz de cazurile anterioare :

XXXX — adresa unde s-a detectat inegalitatea

YYYY — primul octet reprezintă valoarea octetului de la adresa XXXX, iar al doilea, valoarea citită din fișier.

Se recomandă utilizarea comenzii V pentru toate fișierele mai importante din punctul de vedere al conținutului acestora.

Comanda W — (write)

Formatul comenzii este :

W<adinf>,<adsup><CR>

unde :

<adinf> — adresa inferioară a zonei de memorie

<adsup> — adresa superioară a zonei de memorie ce va constitui conținutul fișierului hexa.

Comanda W efectuează transformarea conținutului zonei de memorie specificate într-un bloc de date de format hexa, avînd înregistrări cu lungimea fixă de 16 octeți. Înregistrările sînt emise octet cu octet pe interfața serială. Sfîrșitul fișierului hexa este marcat de o înregistrare de tip „sfîrșit fișier” cu lungimea zero.

Comanda X — (examine)

Formatul comenzii este :

X<CR>

Comanda X permite vizualizarea și modificarea contextului de lucru (registre și indicatori) a unui program întrerupt prin tasta INT sau prin comanda G cu punct de întrerupere. Registrele sînt afișate astfel :

```
IX  YYYY
IY  YYYY
HL  YYYY
DE  YYYY
BC  YYYY
AF  YYYY
PC  YYYY
```

Valoarea YYYY reprezintă conținutul fiecărui registru index sau pereche de registre, în ordinea : octet superior, octet inferior.

Afișajul se execută linie cu linie ; după fiecare conținut afișat, se așteaptă răspunsul operatorului. Dacă se introduce <CR>, conținutul rămîne nemodificat, și se trece la perechea de registre următoare. Dacă se introduc 2 cifre hexazecimale, se modifică conținutul octetului superior al registrelor IX, IY, SP, PC sau registrele H, D, B, A. Dacă se introduc 4 cifre hexazecimale, se modifică conținutul perechii de registre index specificate. Modificarea octetului inferior este posibilă prin introducerea a 2 cifre hexazecimale identice u cele ale octetului superior afișat, urmate apoi de 2 cifre ce reprezintă valoarea dorită pentru octetul inferior.

Comanda X tratează întotdeauna setul complet de registre, terminîndu-se după afișarea registrului PC.

5.2.3. Funcții utilizator — descriere și utilizare (Funcții standard — STD și nestandard — NST). Funcțiile utilizator reprezintă interfața dintre rutinele de bază ale monitorului MON.AMIC și programele utilizator. Rolul lor constă în crearea unei interfețe standard între programele de aplicație și monitor, simplificînd considerabil scrierea și punerea la punct a acestora.

Apelul oricărei funcții utilizator se execută astfel :

- se încarcă registrul C cu numărul (identificatorul) funcției
- se încarcă (dacă e cazul) registrele D și E cu parametrii de apel
- se execută instrucțiunea CALL 0005H.

Monitorul MON.aMIC posedă la adresa 0005H o secvență de decodificare a apelului și de bransare la rutina de tratare specifică funcției apelante. După executarea rutinei de tratare, controlul este redat utilizatorului la instrucțiunea ce urmează secvenței de apel.

Funcțiile de utilizator rezolvă interfața cu toate echipamentele periferice cu care este dotat microcalculatorul aMIC, degrevînd programatorii aplicațiilor de detaliile hardware specifice fiecărui periferic în parte.

Definirea funcțiilor utilizator s-a făcut conform standardului acceptat de sistemul de operare CP/M V2.2 ; toate funcțiile de lucru cu perifericele logice (consolă, cititor, perforator, listare) sînt identice cu cele ale sistemului GP/M. Se asigură astfel posibilitatea dezvoltării și testării programelor de aplicație pe sisteme de dezvoltare ce oferă facilități evaluate de punere la punct a programelor, codul obiect obținut în final putînd fi apoi transferat prin interfața serială pe microcalculatorul aMIC, fără nici un fel de modificare.

Singurele restricții impuse la scrierea programelor sub CP/M sînt următoarele :

a) absența funcțiilor de lucru cu discul flexibil, acesta nefăcînd parte din configurația de periferice a microcalculatorului aMIC.

b) codul obiect al programului de aplicație va trebui să înceapă la adresa 6100H și să nu utilizeze spațiul de memorie 6000H-60FFH decît eventual pentru citiri. Zona de memorie 6000H-60FFH constituie zona de variabile monitor, distrugerea informațiilor de aici putînd avea consecințe imprevizibile.

O dată transferat pe aMIC, codul obiect al aplicației poate fi salvat sub formă de fișier pe casetă magnetică și utilizat ori de cîte ori este necesar.

Dăm în continuare descrierea funcțiilor utilizator implementate în versiunea V 0.2 a monitorului MON.AMIC. Notăția [STD] specifică faptul că funcția este standard CP/M, iar [NST] specifică faptul că funcția este proprie numai monitorului MON.AMIC.

RESET — inițializare sistem [STD]

Apel

C=00H

Efect : inițializarea monitorului, prin executarea unui salt la adresa 0000H. Ecranul este șters, variabilele monitor cuprinse în zona 6000H-60FFH sînt puse toate pe 00H, apoi în stînga sus a ecranului apare mesajul : MON.AMIC V 0.2. Monitorul afișează prompterul "." și așteaptă introducerea de comenzi.

CONIN — "console input" : citire caracter de la consolă [STD]

Apel :

C=01H

Retur :

A=caracter ASCII

Efect : citirea de la consolă a unui caracter și livrarea sa în acumulator. Nu se execută și afișarea caracterului pe ecranul consolei. O serie de caractere de control au semnificații speciale :

- CTRL-C : inițializare sistem
- CTRL-D : inversare mod afișare pe ecran (video invers/normal)
- CTRL-E : inversare mod afișare caractere pe ecran (video invers/normal)
- CTRL-F : inversare regim afișare (scroll/pagină)
- CTRL-P : cuplare/decuplare miniimprimantă pe post de hardcopy al ecranului TV
- CTRL-U : asignarea interfeței seriale drept consolă ; din acest moment, dialogul cu monitorul se poartă prin intermediul interfeței seriale, pînă la primirea pe interfață a unui nou CTRL-U, care reasignează tastatura elastică și ecranul TV la consolă.
- RUBOUT/DEL : ștergerea ultimului caracter introdus și mutarea cursorului înapoi cu o poziție.

Returul din funcția CONIN nu se execută decât după introducerea unui caracter la consolă.

CONOUT — „console output“ : scriere caracter la consolă [STD]

Apel :

C=02H

E=caracter ASCII

Efect : scrierea unui caracter pe ecranul TV în poziția marcată de linia și coloana curentă, sau emiterea caracterului pe interfața serială, dacă aceasta a fost asignată drept consolă.

RIN — „reader input“ : citire caracter de la interfața serială [STD]

Apel :

C=03H

Retur :

A=caracter ASCII

Efect : se citește un octet de la interfața serială și se livrează în acumulator. Nu se predă controlul apelantului decât după citirea caracterului.

POUT — „punch output“ : scriere caracter la interfața serială [STD]

Apel :

C=04H

E=caracter ASCII

Efect : se emite octetul specificat la interfața serială.

LOUT — „List output“ : listare caracter la miniimprimantă [STD]

Apel :

C=05H

E=caracter ASCII

Efect : caracterul specificat se trimite spre imprimare la miniimprimanta atașată microcalculatorului.

INITV — inițializare ecran TV [NST]

Apel :

C=06H

Efect : ștergerea ecranului TV

CASSDR — „audio cassette driver“ : driver tratare operații de intrare/ieșire cu caseta [NST]

Apel :

C=07H

DE=adresă zonă descriptor fișier

Efect : se efectuează operația de intrare/ieșire specificată în descriptorul de fișier. Acesta are următoarea structură :

- oct. 0-1 : numele fișierului (octet inferior, octet superior)
- oct. 2-3 : adresa inferioară de memorie (octet inferior, octet superior)
- oct. 4-5 : adresa superioară de memorie (octet inferior, octet superior)
- oct. 6 : cod operație, cu următoarea structură :
 - c.b.7=1 — scriere fișier cu numele dat în oct. 0-1, și spațiul de [adrese delimitat de conținutul octeților 2-3 și 4-5. (comanda K)
 - c.b.6=1 — citire prim fișier la adresa la care a fost salvat prin comanda de scriere ;
oct. 0-5 pot avea orice conținut (comanda L<CR>)
 - c.b.5=1 — citire prim fișier la adresa dată în oct. 2-3 ; oct. 0-1 și 4-5 pot avea orice conținut. (comanda L,<adinf>CR>)
 - c.b.4=1 — citire fișier cu numele dat în oct. 0-1 la adresa specificată de conținutul octeților 2-3 ; oct. 4-5 pot avea orice conținut. (comanda L<nume>,<adinf><CR>)
 - c.b.3=1 — citire fișier cu numele dat în oct. 0-1, la adresa de la care a fost salvat prin comanda de scriere ; oct. 2-5 pot avea orice conținut. (comanda L<nume><CR>)
 - c.b.2=1 — verificare fișier cu numele dat în oct. 0-1, prin comparare cu zona de memorie a cărei adresă de început se află în oct.2-3 ; oct. 4-5 pot avea orice conținut. (comanda V<nume>,<adinf><CR>)
 - c.b.1=1 — citire antet de fișier ; oct. 0-5 pot avea orice conținut (comanda N)
 - c.b.0=1 — nu se afișează mesajul cu codul de retur, valoarea medie de prag și antetul de fișier pe ecranul TV.
 - c.b.0=0 — se afișează mesajul cu codul de retur, valoarea medie de prag și antetul de fișier pe ecranul TV.

Funcția CASSDR permite programelor utilitare efectuarea operațiilor de intrare/ieșire cu caseta, modificând dinamic doar conținutul descriptorului de fișier. La revenirea după apelul funcției CASSDR, registrul A conține codul de retur, cu semnificația dată în paragraful 5.2.1 la tratarea comenzii L. Funcție de semnificația codului de retur, programul utilitar poate iniția un dialog cu operatorul microcalculatorului, vizînd emiterea de mesaje de oprire case-tofon, rebobinarea manuală pe început de fișier etc., ajungîndu-se la o tratare „semiautomată“ a operațiilor de intrare/ieșire pe caseta magnetică. Acest mod de tratare nu s-a realizat în cadrul monitorului, din cauza restricțiilor de spațiu de memorie EPROM impuse.

Conținutul descriptorului de fișier nu este modificat de către funcția CASSDR, cu excepția codului de retur 04 (eroare de comparare la operația de verificare fișier). În acest caz, oct. 2-3 conțin adresa din memorie unde s-a detectat inegalitatea, iar oct. 4 și 5 conțin valoarea octetului de la adresa din memorie, respectiv din fișierul de verificat.

BEEP — emitere sunet în difuzor [NST]

Apel :

C=08H

D=număr impulsuri de emis (durată sunet)

E=frecvență sunet

Efect : emiterea unui sunet a cărui durată este proporțională cu valoarea din D, și a cărui frecvență este invers proporțională cu valoarea din E.

WSTRIN — „write string“ : scriere șir caractere la consolă [STD]

Apel :

C=09H

DE₂= adresă șir caractere

Efect : scrierea la consolă a șirului de caractere aflat la adresa dată de perechea DE. Șirul se va termina cu caracterul „\$“ sau 00H.

RSTRIN — „read string“ : citire și editare buffer consolă [STD]

Apel :

C=0AH

DE₂= adresă buffer

Retur : buffer completat

Efect : citirea și editarea bufferului definit de perechea DE. Primul octet al bufferului va fi inițializat de utilizator cu numărul maxim de caractere admise la citire. Al doilea octet va fi completat de funcția RSTRIN, la terminarea editării, și va indica numărul total de caractere introduse în buffer.

Caracterele de control ale editării sînt următoarele :

CTRL — I : tabulare 4 caractere

CTRL — X : anularea întregului buffer și reluarea editării

RUBOUT/DEL : ștergerea ultimului caracter introdus

CR : terminarea editării

LF : terminarea editării

Menționăm că toate caracterele de control nu incrementează contorul de caractere citite. Astfel, dacă primul caracter introdus de operator după apelul funcției RSTRIN este CR, contorul de caractere citite de la adresa „buffer+1“ va fi nul.

CSTS — „console status“ : obținere stare consolă [STD]

Apel :

C=0BH

Retur :

A = stare consolă

Efect : verifică dacă s-a introdus un caracter de la consolă. În caz afirmativ, acumulatorul va conține valoarea 0FFH. Dacă nu s-a introdus nici un caracter, acumulatorul va conține 00H

PGRAF — "graphie primitives" : primitive grafice pentru ecranul TV
[NST]

Apel :

C=0CH

D= număr linie TV (0-255)

E= număr coloană T (0-255)

B= cod operație

Efect : aprinderea, stingerea sau testarea stării unui pixel de coordonate X, Y (X=val. din reg. E ; Y=val. din reg. D) ; ecranul TV este considerat ca o matrice de coordonate X, Y, colțul din stînga-sus avînd coordonatele (0, 0)

Codul operației din reg. B poate avea valoarea :

00— testare stare pixel de coordonate X, Y ; dacă pixelul este aprins, la retur din funcție se încarcă în acumulator valoarea 00H ; în caz contrar, acumulatorul va conține o valoare diferită de zero

01— aprindere pixel de coordonate X, Y ; aprinderea are loc indiferent de modul de afișare (video normal sau invers)

02— stingere pixel de coordonate X, Y ; stingerea are loc indiferent de modul de afișare (video normal sau invers)

5.2.4. Structura zonelor de lucru utilizate de monitorul MON.AMIC V0.2.
Zona de memorie EPROM. În memoria EPROM se află o serie de tabele cu conținut nemodificabil, necesare rutinelor monitorului. Dăm în continuare structura tabelelor mai importante :

Tabelul FNCHR — „Function Characters“

Conține pe cîte un octet caracterul asociat fiecărei comenzi acceptate de monitorul MON.AMIC. Tabelul se definește astfel :

FNCHR : DB 'CDFGKLMNVRSWX'

Acest tabel este utilizat de rutina de identificare comenzi pentru compararea caracterului introdus de operator cu setul de comenzi admise de monitor.

Tabelul FNCADD — "Function Addresses"

Este paralel cu FNCHR, existînd o corespondență biunivocă între o intrare din FNCHR și o intrare din FNCADD. Fiecare intrare din FNCADD

conține adresa de tratare a comenzii aflată în intrarea corespunzătoare din FNCHR. Tabelul se definește astfel:

FNCADD: DW CCMPAP ; comparare zone memorie
 DW DISP ; display memorie
 DW FILL ; umplere memorie cu o constantă
 DW GO ; lansare program în execuție
 DW STORIE ; scriere fișier pe casetă magnetică
 DW LOAD ; încărcare fișier pe casetă magnetică
 DW MOWE ; mutare zonă de memorie
 DW NAME ; atribuire nume fișier curent
 DW VERIF ; verificare fișier
 DW READ ; citire fișier hexa
 DW SUBST ; substituie conținut memorie
 DW WRITE ; scriere fișier hexa
 DW EXAM ; examinare și modificare registre

Tabelul FNCADD este utilizat în cadrul rutinei de decodificare comenzi monitor, pentru branșarea la rutinele specifice de tratare, în funcție de comanda identificată pe baza tabelului FNCHR.

Tabelul CPMADD — "CP/M Function Addresses"

Tabelul CPMADD conține adresele de tratare a funcțiilor utilizator. Tabelul se definește astfel:

CPMADD: DW RESET ; reset sistem
 DW CONIN ; intrare consolă
 DW CONOUT ; ieșire consolă
 DW RIN ; intrare cititor
 DW POUT ; ieșire perforator
 DW LOUT ; ieșire listare
 DW INITV ; inițializare ecran TV
 DW CASSDR ; driver I/E casetă magnetică
 DW BEEP ; emiterie sunet în difuzor
 DW WSTRIN ; scriere șir caractere la consolă
 DW RSTRIN ; editare buffer consolă
 DW CSTS ; obținere stare consolă
 DW PGRAF ; primitive grafice

Acest tabel este utilizat de rutina de decodificare a apelurilor funcțiilor utilizator. Codul funcției, introdus în registrul C, reprezintă indexul în CPMADD, pentru selectarea intrării dorite.

Tabelul GENCAR — "Generate Characters"

Tabelul GENCAR conține definițiile setului de caractere afișabile pe ecranul TV de către driverul de ecran. Un caracter ocupă un spațiu de 6 * 8 pixeli, din care caracterul propriu-zis este definit de 4 * 6 pixeli, restul spațiului din matrice (două coloane și două linii de pixeli) constituind separatorii dintre caractere și rândurile de caractere. Fiecărui caracter îi corespunde o

intrare în GENCAR cu lungime de 3 octeți. Intrarea respectivă conține pe 24 de biți matricea caracterului, fiecare 4 biți constituind o linie de pixeli. Biții poziționați pe 1 indică setarea pixelului respectiv. Afișajul unui caracter se realizează cu ajutorul funcției PGRAF, rolul driverului de ecran constând în determinarea coordonatelor absolute necesare fiecărui apel al funcției. Determinarea coordonatelor pixelilor se face pe baza coordonatelor caracterului de afișat (rînd/coloană curentă) și a celor 6 linii de pixeli ce definesc matricea caracterului.

Tabelul GENCAR conține definițiile pentru caracterele ASCII cuprinse în domeniul 20H (blanc) —5FH (cursor), ordonate crescător :

GENCAR :	DB	0,	0,	0		; blanc
	DB	22H,	22H,	02H		; !
	DB	55H,	0,	0,		; "
	DB	66H,	0FFH,	66H		; #
	DB	27H,	0A6H,	0B6H		; \$
	DB	09H,	24H,	90H		; %
	.					
	.					
	.					
	DB	69H,	0BDH,	96H		; 0
	DB	26H,	0A2H,	27H		; 1
	.					
	.					
	.					
	DB	69H,	9FH,	99H		; A
	DB	0E9H,	0E9H,	9EH		; B
	.					
	.					
	.					
	DB	0,	0,	0FFH		; cursor

Modificarea setului de caractere nu este posibilă prin program, dat fiind faptul că generatorul de caractere se află în memoria EPROM, iar driverul de ecran nu îl recopiază în memoria RAM. Afișarea de caractere diferite de standardul implementat în GENCAR este posibilă (tot cu ajutorul funcției PGRAF), dar procedura este destul de laborioasă, necesitînd scrierea de rutine în limbaj de asamblare.

Zona de memorie RAM

În memoria RAM, implantată începînd cu adresa 4000H, se află două zone de lucru principale :

- memoria ecran, în spațiul de adrese 4000H-5FFFH
- zona de variabile monitor, în spațiul de adrese 6000H-60FFH

Începînd cu adresa 6100H, memoria RAM este disponibilă pentru programele utilizator.

Memoria ecran păstrează în corespondență biunivocă imaginea ecranului TV, considerat ca o matrice de 256 * 256 de pixeli. Adresa 4000H corespunde pixelului de coordonate (0, 0), aflat în colțul din stînga sus al ecranului.

Ecranul este compus din 256 linii TV, fiecare linie TV putînd afișa 256 de pixeli. Rezultă că unei linii TV îi corespunde o zonă de 32 de octeți din memoria ecran, adresarea unei linii TV făcîndu-se în increment de 32 relativ la adresa de început a memoriei ecran. Calculul adresei corespunzătoare unui pixel de coordonate (X, Y) se efectuează după formula :

$$ADROCT = 4000 + Y * 32 + X / 8$$

Adresa ADROCT specifică octetul care conține pixelul de coordonate X, Y ; restul împărțirii X/8 specifică bitul din octet corespunzător pixelului (bitului c. m.p.s. din octet îi corespunde restul 7).

Zona de variabile monitor, rezervată în continuarea memoriei ecran, ocupă un spațiu de 256 octeți ; structura zonei este următoarea :

RND(6000H) — numărul rîndului curent utilizat de driverul de ecran pentru afișarea de mesaje. Valoarea conținută variază între 0 (primul rînd de sus) și 31 (ultimul rînd de jos).

COL(6001H) — numărul coloanei curente utilizat de driverul de ecran pentru afișare de mesaje. Valoarea conținută variază între 0 (primul caracter din stînga în cadrul unui rînd) și 39 (ultimul caracter din dreapta în cadrul unui rînd)

AFMOD(6002H) — indicator mod de afișare ; dacă AFMOD=00H, afișajul se execută în mod SCROLL, iar dacă AFMOD=0FFH, afișajul se execută în mod PAGE. Setarea/resetarea indicatorului se execută de funcția CONIN la recepționarea codului CTRL-F (06H)

ECRINV(6003H) — indicator stare ecran ; dacă ECRINV=00H, conținutul întregului ecran se afișează în video normal, iar dacă ECRINV=0FFH, conținutul întregului ecran se afișează în video invers. Setarea/resetarea indicatorului se execută de funcția CONIN la recepționarea codului CTRL-D(04H).

CARINV(6004H) — indicator mod afișare caractere ; dacă CARINV=00H, orice caracter emis cu funcția CONOUT va fi afișat în video normal, iar dacă CARINV=0FFH, afișarea caracterelor se execută în video invers. Afișarea caracterelor este independentă de starea indicatorului ECRINV. Setarea/resetarea indicatorului CARINV se execută de funcția CONIN la recepționarea codului CTRL-E (05H).

IOBYTE(6005H) — indicator asignare echipamente fizice la echipamente logice :

- C.b. 0 (asignare consolă) :
 - 0 : ecran TV și tastatura elastică
 - 1 : interfața serială
- C.b. 7 (asignare imprimantă) :
 - 0 : nu se cere imprimare în paralel cu scrierea la consolă
 - 1 : se cere imprimare în paralel cu scrierea la consolă.

Setarea/resetarea c.b.0. este efectuată de funcția CONIN la recepționarea codului CTRL-U(15H) ; setarea/resetarea c.b.7 este efectuată de funcția CONIN la recepționarea codului CTRL-P(10H).

La inițializarea sistemului (RESET), octetul IOBYTE conține valoarea 00H.

Zona de memorie 6006H-602BH este utilizată de monitor pentru recepție mesaje operator și salvări de registre.

USERIX(602CH) — USERPC(603AH) — conține contextul de lucru al programelor utilizator după o întrerupere; ordinea depunerii conținutului registrelor este următoarea: IX, IY, HL, DE, BC, AF, SP, PC. Această zonă este afișată pe ecran la orice întrerupere de program, și poate fi modificată înainte de relansarea programului prin comanda X.

ADINF(6044H), ADSUP(6046H), ADMOV(6048H) — zonă ce conține adresa inferioară, superioară și de depunere rezultată în urma decodificării parametrilor comenzilor D, F, G, K, L, M, R, S, V, W

ZONCAR (604AH) — zonă cu lungimea de 8 octeți, utilizată de driverul de ecran pentru expandarea matricilor caracterelor într-o formă utilizabilă de funcția PGRAF. Fiecare linie de 4 pixeli din matricea de caractere este transformată într-un șir de 6 biți, primul și ultimul bit din șir constituind separatorii de caractere. Primul și ultimul octet din ZONCAR au valoarea 0FFH, generând separatorii de rând (câte o linie de pixeli „stinsă“).

Zona de memorie 6052H—60FFH conține alte variabile de lucru, fără semnificație pentru utilizatori. Pointerul stivei monitorului este inițializat cu valoarea 60FFH.

Programele utilizator se încarcă în memoria RAM începând cu adresa 6100H; limita superioară a memoriei RAM depinde de opțiunea de livrare a microcalculatorului AMIC, valoarea cea mai redusă fiind 7FFFH (16 Ko RAM utilizator, din care primii 8 Ko sînt ocupați întotdeauna de memoria ecran). Versiunile cu spațiu de memorie RAM extins cuprind 32 Ko sau 48 Ko memorie utilizator.

5.2.5. Modul de utilizare a monitorului MON.AMIC V0.2. După punerea sub tensiune și apăsarea tastei RESET, monitorul efectuează inițializarea zonei de variabile, ștergerea ecranului TV și afișarea mesajului:

MON.AMIC V0.2]

Se emite pe rîndul următor prompterul "." indicînd așteptarea unei comenzi din partea operatorului. Orice comandă introdusă este verificată din punct de vedere al:

- corectitudinii numelui (să aparțină setului de comenzi admis).
- corectitudinii valorilor parametrilor (maxim 4 cifre hexazecimale).
- corectitudinii numărului de parametri introduși (să fie în concordanță cu cel solicitat de comanda în cauză).

Orice eroare are ca efect afișarea caracterului "?" urmat de prompter; operatorul poate reintroduce din nou comanda dorită.

Generarea secvențelor scurte de programe utilizator poate fi efectuată cu ajutorul comenzii S (substitute) în zona de memorie utilizator. Verificarea

corectitudinii datelor introduse se efectuează cu comanda D (display). Pentru programe cu lungimi ce depășesc câteva sute de octeți, utilizarea comenzii S devine anevoioasă, crescând riscul erorilor la introducere. În acest caz, se vor elabora programele pe un alt sistem de dezvoltare și transfera apoi codul obiect prin interfața serie pe aMIC, cu comanda R (read).

Verificarea programelor introduse sau generate în memorie se va efectua cu ajutorul comenzii G (go) cu punct de întrerupere. Se vor efectua verificări din aproape în aproape, pe porțiuni cât mai scurte de program. La fiecare întrerupere, se va verifica conținutul registrelor; dacă se detectează o neconcordanță față de valorile presupuse prin program, se va restringe spațiul de testare, pentru a localiza cât mai rapid erorile. Dacă programul lansat în execuție ciclează (rămâne în buclă infinită), se va apăsa tasta INT; ca urmare, se va întrerupe necondiționat execuția programului și se va afișa contextul de lucru. Se reia procedura de verificare, modificând după necesitate conținutul acestora înainte de o nouă relansare a programului.

Odată efectuată punerea la punct a unui program acesta va fi salvat pe casetă magnetică utilizând comanda K (casette), corectitudinea salvării testându-se cu comanda V (verify). Se recomandă elaborarea modulară a programelor testându-se segmente (porțiuni din program, subrutine etc.) care efectuează câte o singură funcție logică din p.v. al aplicației căreia îi este destinat programul. Fiecare modul va fi salvat separat într-un fișier pe casetă magnetică, simplificându-se astfel operațiile de punere la punct în ansamblu al programului. Pentru testarea unui nou modul, se vor încărca în memorie numai acele fișiere care sînt legate de modulul de testat; odată terminată punerea la punct a modulului, acesta va fi salvat pe casetă magnetică, reluîndu-se procedura de mai sus pînă la elaborarea completă a programului. Versiunile finale ale programului vor fi salvate, sub forma unui singur fișier, pentru a nu prelungi inutil operația de încărcare în memorie.

5.3. Monitorul Z-80 V0.0 — descriere și utilizare

5.3.1. *Prezentare generală.* Monitorul Z-80 V0.0 reprezintă prima versiune de monitor integrată cu interpretorul BASIC scris în instrucțiuni specifice microprocesorului Z-80, fiind implementat pe prima versiune comercializată a microcalculatorului aMIC.

Monitorul asigură interfața hardware-ului cu interpretorul BASIC sau cu alte programe aplicative create de utilizator în memoria RAM. Interfața este asigurată prin subrutine de intrare/ieșire pentru perifericele microcalculatorului: tastatură, ecran TV, casetofon audio.

La lansarea în execuție a monitorului (punere sub tensiune sau apăsare pe tastă „RESET“), pointerul stivei este plasat pe ultima adresă de memorie RAM, ecranul este șters și se afișează mesajul: Z-80 MONITOR; apoi este emis caracterul "*" care indică așteptarea unei comenzi din partea utilizatorului.

Comenzile monitorului asigură :

- afișarea și modificarea unor zone de memorie RAM ;
- afișare conținut registre microprocesor;
- lansarea în execuție a unui program ;
- programarea de întreruperi software în cadrul programelor de testat ;
- salvarea unor zone de memorie pe casetă sub formă de fișiere ;
- încărcarea fișierelor de pe casetă în memorie.

Comenzile monitorului se compun dintr-o singură literă, urmate de unul sau mai multe argumente ; aceste argumente se introduc sub forma de 4 caractere hexazecimale (0-9 și A-F). În unele situații, monitorul corectează erorile de operare la introducerea comenzilor. Operatorul are posibilitatea de a șterge ultimul caracter introdus prin tasta „DEL”.

Setul de comenzi implementat reprezintă un minim necesar pentru introducerea de programe în cod obiect, lansarea, depanarea și stocarea lor pe casetă.

❑ Versiunea actuală de monitor ocupă cca. 3 Koct. de memorie EPROM.

5.3.2. Comenzile monitorului. Vom adopta următoarea notație, utilizată pentru argumentele din cadrul comenzilor :

- xxxx** — adresă de memorie (4 caractere hexazecimale)
- aaaa** — adresă condiționată de relația $aaaa \geq xxxx$
- bbbb** — adresă de memorie
- cccc** — adresă furnizată automat de monitor, ca rezultat al operației $bbbb + (aaaa - xxxx)$
- kkkk** — cod de fișier (4 caractere hexazecimale)
- hh** — constantă hexazecimală pe 1 octet

Listarea comenzilor se va efectua în ordinea de la paragraful ❶ (lucrul cu memoria, lucrul cu programele, lucrul cu caseta).

I — INSERARE

Ixxxx(CR) hhhhhh ... hh (CR)

— inserează un șir de octeți hh începând de la adresa XXXX. Corecția pe șir se face prin apăsarea tastei "DEL". După acționarea celui de-al doilea (CR), monitorul afișează adresa primei locații de memorie ce urmează ultimului octet intr dus.

V — VIZUALIZARE

Vxxxx, aaaa(CR) ;

— se afișează conținutul zonei de memorie [delimitată] de adresele
xxxx — aaaa. ❑ ❑

G — LANSARE PROGRAM

Gxxxx(CR)

— se lansează în execuție programul aflat la adresa de memorie XXXX.

F — UMLERE CU O CONSTANTĂ

Fxxxx, aaaa, hh(CR)

— se umple zona de memorie de la xxxx la aaaa cu constanta hh.

M — MUTARE ZONĂ MEMORIE

Mxxxx, aaaa, bbbb, cccc(CR)

— se mută conținutul zonei de memorie xxxx-aaaa în zona bbbb-cccc.

Y — COMPARARE ZONĂ MEMORIE

Yxxxx, aaaa, bbbb, cccc(CR)

— compară octet cu octet conținutul zonei xxxx-aaaa cu zona bbbb-cccc.

Diferențele sesizate sînt afișate sub forma adreselor din prima zonă. Afișarea poate fi oprită prin tasta "INT", iar monitorul se relansează prin comandă "C".

SD — SUMĂ ȘI DIFERENȚĂ

SDxxxx, bbbb(CR)

— afișează suma $xxxx+bbbb$ și diferența $xxxx-bbbb$ **R — RESET ÎNTRERUPERI SOFTWARE**

R(CR)

— se inițializează monitorul pentru lucrul cu întreruperile programabile; comanda este obligatorie după punerea sub tensiune a microcalculatorului.

B — PROGRAMARE BREAKPOINT

Bxxxx(CR)

— se programează o întrerupere de tip „B” la adresa xxxx. Dacă programul în curs de execuție ajunge la această adresă, se afișează mesajul:

&xxxx — unde xxxx = adresa de întrerupere și se redă controlul monitorului.

C — RELANSARE PROGRAM ÎNTRERUPT

C(CR)

— se relansează ultimul program întrerupt prin comandă „B” anterioară sau apăsare tastă „INT”.

TASTA „INT” — ÎNTRERUPERE NEMASCABILĂ

— acționarea acestei taste se poate face, în orice moment; se oprește execuția programului în curs și se afișează &xxxx unde xxxx = adresa la care s-a produs întreruperea.

Se indică utilizarea tastei „INT” pentru oprirea programelor aflate în buclă infinită.

T — TRASARE

Txxxx(CR)

— se programează o întrerupere de tip „T” la adresa xxxx. Spre deosebire de întreruperea de tip „B”, controlul nu este redat monitorului, ci se afișează:

— &xxxx

— conținut registre microcalculator

Se continuă apoi execuția programului cu instrucțiunea ce urmează după adresa xxxx.

D — DEZACTIVARE ÎNTRERUPERI

D(CR)

— toate întreruperile programabile de tip B sau T sînt dezactivate (nu mai au efect asupra programelor în curs de execuție).

X — AFIȘARE CONȚINUT REGISTRE)

X(CR)

— se afișează conținutul registrelor; acest conținut corespunde ultimei întreruperi „B”, „T” sau „INT”.

K — SALVARE ZONĂ MEMORIE PE CASETĂ

Kxxxx, aaaa, kkkk(CR)

— se pornește manual casetofonul, apoi se introduce comanda „K”: zona de memorie xxxx—aaaa va fi salvată ca un fișier cu numele kkkk.

A — LISTARE ANTETE FIȘIERE.

A(CR)

— se pornește casetofonul și se poziționează caseta pe început de bandă: comanda „A” va afișa toate antetele fișierelor sub forma:

PP HL : xxxx DE : kkkk BC : LLLL

— informațiile sînt:

PP — valoare prag de diferențiere la înregistrarea informației pe casetă; în mod normal PP=18-1A.

xxxx — adresa de încărcare în memorie a fișierului

kkkk — numele fișierului

LLLL — lungimea în octeți a fișierului

— oprirea din comanda „A” se face apăsînd pe tasta „INT”

L — ÎNCĂRCARE FIȘIER

L(CR)

— se încarcă primul fișier (indiferent de numele acestuia) la adresa de la care a fost salvat prin „K”.

Q — ÎNCĂRCARE FIȘIER

Qxxxx(CR)

— încărcare primul fișier (indiferent de numele acestuia) la adresa xxxx.

Z — ÎNCĂRCARE FIȘIER

Zkkkk(CR)

— caută pe casetă fișierul cu numele kkkk și îl încarcă la adresa de la care a fost salvat prin comanda „K”.

Obs: la citirea unei informații incorecte de pe casetă, se afișează mesajul „ERROR”.

5.3.3. Legătura monitor-utilizator. Pentru a utiliza facilitățile oferite de monitor cu privire la gestiunea perifericelor, se vor da în continuare adresele de implantare ale subrutinelor de intrare/ieșire.

CIN — CONSOLE INPUT

ADR=07FD

RUTINA CIN citește un caracter de la tastatură și îl furnizează în registrul A. Rămâne în buclă pînă la apăsarea unei taste. După apăsarea tastei, se emite un semnal sonor scurt, indicînd terminarea rutinei CIN.

COUT — CONSOLE OUTPUT

ADR=07FA

Rutina COUT trimite la display (ecran TV) caracterul conținut în registrul A (cod ASCII) și îl afișează în poziția curentă a cursorului pe ecran

KIN — CASSETTE INPUT

ADR=3C1C

Rutina KIN citește de pe casetă un fișier, în memoria RAM, la adresa de la care a fost salvat. Înainte de apelarea subrutinei, casetofonul trebuie pornit în mod redare.

KOUT — CASSETTE OUTPUT

ADR=3BAE

Rutina KOUT înscrie pe casetă un fișier din memoria microcalculato-
rului. Înainte de apel, se vor încărca registrele astfel:

HL= adresa de început zonă de memorie

BC= număr total octeți de scris

DE= nume fișier.

Casetofonul va fi pornit în prealabil în mod înregistrare, cu volumul poziționat pe nivel mediu. Este de preferat ca înregistrările și redările să fie efectuate pe același casetofon.

De asemenea, există o serie de locații de memorie aflate la adrese fixe, conținînd informații care pot fi utile în aplicații:

- 6000 : număr rînd în care se află poziționat cursorul pe ecranul TV. Valoarea 00 corespunde rîndului de sus, iar 1F rîndului de jos (ultim).
- 6001 : numărul coloanei în care se află poziționat cursorul pe ecranul TV. Valoarea 00 indică coloana cea mai din stînga, iar 1D coloana cea mai din dreapta.
- 6002 : mod afișaj — dacă valoarea este 00, avem mod defilare, dacă avem FF, mod pagină.
- 6003 : video normal/invers la nivelul întregului ecran (00 video normal — FF video invers). Comutarea se poate face și prin înscrierea în portul 22H a valorii 00H pentru video normal și 10H pentru video invers.
- 6004 : video normal/invers la nivel de caracter (00 video normal — FF video invers). Comutarea se poate face și apăsînd tastele CTRL și E sau apelînd rutina COUT cu acumulatorul A=05H.

5.3.4. Exemple de utilizare

● Se dorește afișarea și modificarea unei zone de memorie RAM cuprinsă în spațiul D10A-D10F; zona conține inițial o informație oarecare:

```
VD10A, D10F (CR)
D10A 0F0B 001A 0D03
ID10A (CR) 102030 (CR) D10L
VD10A, D10F (CR)
D10A 1020 301A 0D03
```

● Umplerea zonei de memorie de la 6000 la 7000 cu constanta FE:

```
F6000, 7000, FE (CR)
V6000, 6003 (CR) — vizualizarea primilor 4 octeți
6000 FEFE FEFE
```

● Introducerea unui program în memorie și lansarea lui în execuție:

```
R (CR) — inițializare puncte de break
I6800 (CR) 3E41C DFA0718F9 (CR) 6807
Programul introdus la adresa 6800 este:
```

```
3E41 LD A,41H
C DFA07 CALL 07FAH (rutina COUT)
18F9 JR 3-7
```

Are ca efect afișarea caracterului „A” în mod repetat pe ecran.
Stabilim puncte de întrerupere:

```
B6800 (CR)
T6802 (CR)
```

Lansăm programul în execuție:

```
G6800 (CR); ca urmare a întreruperii „B” se afișează:
& 6800
```

Cu comanda X vizualizăm registrele

```
SP FFFE IX FFFF IY FFFF A' 0D9B
HL 6800 DE 0000 BC 9801 A 0054
HL 0000 DE 0000 BC 0101
```

Primele perechi de registre A, HL, DE, BC reprezintă registrele și indicatorii în condiții curenți; următoarele perechi reprezintă A', HL', DE' BC'.

Reluarea execuției după întrerupere se face cu:

```
C (CR)
```

În acest moment, execuția programului se reia, iar la atingerea adresei 6802 se execută trasarea, prin afișarea informației:

```
& 6802 SP FFFE IX FFFF IY FFFF
A 4126 HL 6800 DE 0010 BC 2793
A 0054 HL 0000 DE 1000 BC 0101
```

Execuția continuă fără oprire, afișându-se caracterul: „A” pe ecran, după apelul rutinei COLT. Oprirea se execută iarăși la adresa 6800, cu afișarea informației de mai sus.

Dezactivarea punctelor de întrerupere din Program se execută cu comanda

```
D (CR)
```

Din acest moment, după lansare, programul nu se va mai opri decât prin apăsarea tastei „INT”.

● Salvarea programului de la adresa 6800 pe casetă :

K6800, 6806 0001 (CR)

Se salvează zona 6800-6806, în fișierul cu nume 0001.
Ștergem acum zona respectivă :

F6800, 6808, 00 (CR)

Vizualizăm zona ștearsă :

V6800, 6806 (CR)

6800 0000 0000 0000 0000

Încărcăm acum fișierul de pe casetă în memorie :

L (CR)

1A HL 6800 DE 0001 BC 0006

Vizualizăm din nou zona ștearsă anterior :

V6800, 6806 (CR)

6800 3E41 CDFA 0718 F900

Observăm încărcarea la adresa 6800 a informației salvate anterior pe casetă.

6.1. Introducere

Monitorul DEST (Monitor DEzvoltare Software și Testare) pentru sisteme cu microprocesor Z80 constituie extensia maximă a monitorului MON.AMIC prezentat în capitolul 5, transformând microcalculatorul aMIC într-un sistem eficient, compact și versatil pentru dezvoltarea și testarea aplicațiilor scrise în limbaj de asamblare Z80.

Comenzile monitorului MON.AMIC reprezintă un subset al comenzilor monitorului DEST. Setul de comenzi caracteristic monitorului DEST oferă utilizatorilor următoarele facilități:

- crearea și modificarea de fișiere sursă în limbaj de asamblare;
- asamblarea fișierelor sursă și crearea de module obiect relocabile sau absolute;
- linkeditarea mai multor module obiect relocabile într-un singur modul și transformarea acestuia în modul obiect absolut (program executabil);
- dezasamblarea codului obiect din orice zonă de memorie, listarea sursei astfel obținute și memorarea ei sub formă de fișier pentru producții ulterioare
- execuția monitorizată pas cu pas a unui program, indiferent de tipul memoriei (EPROM sau RAM) unde se află stocat programul;
- încărcarea datelor de intrare și salvarea datelor de ieșire pe casetă magnetică prin proceduri incluse în cadrul comenzilor, degrevând utilizatorii de gestiunea zonelor de memorie solicitate de fiecare comandă în parte.

Prin implementarea facilităților enumerate mai sus s-a urmărit acoperirea cerințelor de bază necesare unui sistem de dezvoltare software:

- concepția, elaborarea și testarea modulară a aplicațiilor;
- integrarea facilă a unui set de module testate într-o singură aplicație complexă;
- posibilitatea testării imediate a unui modul elaborat;
- ciclu scurt de creare-modificare-testare a fiecărui modul în parte; un fișier sursă creat în memorie sau încărcat de pe casetă magnetică poate fi asamblat, relocat și testat fără pierderea informației din zona de memorie ce conține textul sursă, putându-se opera modificările necesare imediat după detectarea unor anomalii în execuția modulului și relua întregul ciclu de dezvoltare;

— stocarea modulelor obiect ce compun o aplicație, sub formă de fișiere pe casetă magnetică, asigurând disponibilitatea lor în orice moment;

— posibilitatea transmiterii codului obiect al aplicației prin interfața serială spre un programator de memorii EPROM, sau programarea imediată a memoriilor EPROM cu programatorul cuplat la extensia de bus a microcalculatorului.

Monitorul DEST în versiune maximă (avînd toate facilitățile enumerate) ocupă întreg spațiul de memorie EPROM (0—3FFFH) al microcalculatorului aMIC. Spațiul de memorie (0—0FFFH) este ocupat de monitorul MON.AMIC împreună cu driverele de tastatură, ecran, casetă magnetică, interfață serială și miniimprimantă. Spațiul de memorie 1000H—1FFFH este ocupat de funcția de dezasamblare și execuție monitorizată a programelor înscrise în memoria RAM sau EPROM. Restul spațiului de memorie, pînă la adresa 3FFFH, este ocupat de funcțiile de editare fișiere sursă, asamblare și linkeditare module relocabile. Utilizatorul are astfel posibilitatea de a testa o aplicație înscrisă în propriile sale „chip-uri” EPROM 2716 înlocuind orice chip din spațiul de adrese 2000H-3FFFH. Comanda de dezasamblare și execuție monitorizată rămîne funcțională, permițînd testarea aplicației din memoria EPROM „utilizator”.

Configurația minimă pentru testare-depanare necesită în consecință 8 Ko de memorie EPROM și 16 Ko de memorie RAM. Configurația necesară dezvoltării de aplicații cuprinde 16 Ko memorie EPROM și 16—48 Ko memorie RAM, iar ca periferice atașate, un casetofon audio și o miniimprimantă. Prezența acestora din urmă nu este obligatorie, listările fișierelor sursă obținute în urma asamblării, dezasamblării sau editării de test făcîndu-se implicit pe ecranul TV, însă dezvoltarea unor aplicații mai complexe este dificilă fără posibilitatea păstrării unui listing pe hîrtie.

6.2. Comenzile monitorului DEST

Vom prezenta în continuare numai comenzile specifice monitorului DEST, restul comenzilor fiind identice cu cele ale monitorului MON.AMICV0.2, prezentat în cap. 5.

6.2.1. Comanda A—„assembly source program”. Formatul comenzii este următorul:

- A<CR>

Comanda A lansează în execuție asamblorul rezident ASR-Z80, care prelucrează un fișier sursă în limbaj de asamblare Z80, elaborînd listingul asamblării și modulul obiect respectiv.

ASR—Z80 recunoaște toate mnemonicele standard Z80, un set de directive standard, un set de directive pentru generarea de funcții utilizator (v. cap. 5 paragraf 5.3), simboluri globale și referințe externe. Asamblorul poate prelucra un fișier sursă de orice lungime, limitat doar de spațiul de memorie necesar păstrării sursei respective și de numărul total de simboluri din fișier.

Definirea termenilor

FIȘIER SURSĂ — zonă de memorie definită prin adresa ei de început, care conține programul sursă al utilizatorului, scris în limbaj de asamblare Z80. Sfârșitul unui fișier sursă este indicat de primul caracter ne-ASCII (având c.b. 7=1) întâlnit în parcurgerea octet cu octet a zonei de memorie cu adresa de început specificată.

MODUL OBIECT — rezultatul asamblării unui fișier sursă. Pentru un fișier sursă asamblat, se creează un modul obiect, care poate fi de tip absolut sau relocabil.

MODUL OBIECT ABSOLUT — modul obiect reprezentând un program executabil într-un singur spațiu de adrese, definit la crearea modului. Modulele obiect absolute nu pot fi linkeditate împreună cu alte module obiect și nu pot fi executate decât prin încărcarea lor în zona de memorie pentru care au fost definite.

MODUL OBIECT RELOCABIL — modul obiect conținând toate informațiile pentru a putea fi linkeditat împreună cu alte module obiect relocabile și transformat în modul obiect absolut, încărcabil și executabil în orice spațiu de adrese. Modulele obiect relocabile nu sînt programe executabile.

LISTING ASAMBLARE — imaginea fișierului sursă, completată cu conținutul de amplasare în memorie a codului obiect generat și cu codul obiect asociat fiecărei linii sursă; listingul asamblării se afișează la consolă și poate fi imprimat în paralel la miniimprimantă.

SIMBOL GLOBAL — simbol definit într-un fișier sursă (și atașat modului obiect relocabil respectiv) destinat a fi utilizat în oricare alt modul obiect relocabil linkeditat împreună cu modulul ce conține definiția.

REFERINȚA EXTERNĂ — simbol utilizat într-un fișier sursă (și atașat modului obiect relocabil respectiv) și definit obligatoriu ca simbol global în alt fișier sursă. Rezolvarea referințelor externe se face la linkeditarea modulelor obiect relocabile ce conțin definițiile cu cele ce conțin referințele.

LINKEDITARE — transformarea unui număr oarecare de module obiect relocabile într-un singur modul obiect absolut, cu rezolvarea tuturor referințelor externe. Modulul obiect astfel obținut este executabil de la adresa specificată la lansarea operației de linkeditare.

¶ Sintaxa limbajului de asamblare acceptat de ASR-Z80

Un program sursă în limbaj de asamblare constă din linii sursă care pot conține: etichete, mnemonice, argumente ale mnemonicelelor, directive și comentarii. Formatul unei linii sursă cuprinde următoarele cîmpuri:

[<et>] <mnemonică> | <directivă> [<argument>] [<comentariu>]
 <et> — etichetă, ce constă din maxim 6 caractere alfanumerice, primul fiind obligatoriu literă.
 <mnemonică> — mnemonicele standard corespunzătoare setului de instrucțiuni ale microprocesorului Z80.

- <directivă> — mnemonică cu semnificație specială în cadrul procesului de asamblare.
- <argument> — parametrii aferenți mnemoniceii sau directivei, separați prin virgulă.
- <comentariu> — șir de caractere ASCII precedat de caracterul „;” și ignorate la asamblare.

Din cîmpurile enumerate mai sus, numai cîmpul mnemoniceii sau directivei sînt obligatorii. Separatorii cîmpurilor pot fi: blanc(uri) sau caracterul TAB (09H). Eticheta, dacă există, va începe întotdeauna din prima coloană a liniei sursă. Dacă nu există etichetă, prima coloană a liniei sursă va fi obligatoriu blanc sau TAB. Eticheta nu trebuie să fie urmată de caracterul „:”, simplificîndu-se astfel activitatea de introducere a textului sursă de către utilizatori.

Directivele admise de asamblorul ASR-Z80

Deoarece mnemonicele standard și argumentele aferente corespunzătoare setului de instrucțiuni Z80 au fost prezentate anterior, nu mai insistăm asupra lor, specificînd doar faptul că asamblorul ASR-Z80 recunoaște toate cele 74 de mnemonice standard Z80 (LD, INC, DEG, ADD etc.).

Directivele admise de ASR-Z80 se divid în 2 categorii: directive standard (caracteristice majorității asambloarelor) și directive utilizator, specifice numai pentru microcalculatorul aMIC.

Directivele standard sînt următoarele:

Define Byte — definire șir octeți

Sintaxa:

DB expr. 1, expr. 2, ...

Directiva DB generează, începînd de la valoarea curentă a contorului de amplasare cod obiect, un număr de octeți egali cu numărul de expresii din zona de argument, fiecare octet conținînd valoarea evaluată a expresiei corespunzătoare (valoarea nu poate depăși 0FFH). O expresie poate fi și un șir de caractere ASCII încadrate de apostrofuri. În acest caz, fiecare octet va conține valoarea codului ASCII corespunzător caracterelor din cadrul șirului.

Define Word — definire cuvînt

Sintaxa:

DW expr. 1, expr. 2, ...

Directiva DW generează pe cîte 2 octeți, începînd de la valoarea curentă a contorului de amplasare cod obiect, valoarea corespunzătoare fiecărei expresii din zona de argument. Generarea fiecărei valori se face în ordinea: octet mai puțin semnificativ, octet mai semnificativ.

Define Storage — definire spațiu de memorie

Sintaxa:

DS expr.

Directiva DS execută rezervarea, începînd de la valoarea curentă a contorului de amplasare cod obiect, a unui număr de octeți egal cu valoarea

rezultată în urma evaluării expresiei din zona de argument. Nu se admit mai multe expresii în cadrul argumentului.

End Statement — sfârșit program sursă

Sintaxa :

END

Directiva **END** este opțională, semnificând sfârșitul programului sursă. În absența ei, sfârșitul sursei este detectat prin identificarea primului caracter ne-ASCII din zona de memorie ce conține sursa.

Equated Symbol — definire valoare simbol

Sintaxa :

Etichetă **EQU** expr

Directiva **EQU** atribuie simbolului din câmpul etichetei, valoarea calculată a expresiei din câmpul de argument. Nu se admite decât o singură directivă **EQU** pentru un simbol dat. Expresia din câmpul de argument nu poate conține simboluri care nu au fost încă definite pînă la apariția directivei, sau referințe externe.

Set Symbol — setare valoare simbol

Sintaxa :

etichetă **SET** expr

Are același efect ca și directiva **EQU**, cu deosebirea că pentru același simbol se admit oricâte directive **SET**. Expresia din câmpul de argument nu poate conține simboluri nedefinite pînă la apariția directivei **SET**, sau referințe externe.

Entry Point — definire simboluri globale (puncte de intrare în program)

Sintaxa :

ENTRY Simbol 1, Simbol 2, ...

Directiva **ENTRY** se utilizează în scopul marcării simbolurilor din câmpul de argument drept simboluri globale. Toate simbolurile din lista directivei vor fi definite obligatoriu în cadrul programului sursă.

External Symbol — definire referințe externe

Sintaxa :

EXTRN simbol 1, simbol 2, ...

Directiva **EXTRN** se utilizează în scopul marcării simbolurilor din câmpul de argument drept simboluri utilizate în cadrul programului sursă, dar definite în cadrul altui program sursă. Orice simbol din lista directivei **EXTRN** va trebui să apară și în cadrul unei directive **ENTRY**.

Origin of Code — setare valoare contor de amplasare cod obiect

Sintaxa :

ORG expr

Directiva **ORG** modifică valoarea contorului de amplasare a codului obiect conform evaluării expresiei din câmpul de argument. Inițial, valoarea contorului de amplasare este zero pentru modulele obiect relocabile, sau dată de utilizator pentru cele absolute. Dacă valoarea calculată a expresiei este mai mică decât valoarea curentă, se semnalază eroare. Expresia nu poate conține simboluri nedefinite pînă la întîlnirea directivei **ORG**, sau referințe externe.

Phase modify — modificare spațiu de adrese

Sintaxa :

PHASE expr

Directiva **PHASE** modifică valoarea contorului de adrese asociat simbolurilor din programul sursă. Pînă la apariția primei directive **PHASE**, valoarea contorului de amplasare cod obiect coincide cu cea a contorului de adrese simboluri (unui simbol din câmpul de etichetă i se asignează valoarea curentă a contorului de amplasare cod obiect). Directiva **PHASE** lasă nemodificat contorul de amplasare, astfel încît codul obiect se generează în continuare în zona de memorie specificată la lansarea în execuție a asamblorului; valorile care se vor atribui în continuare tuturor simbolurilor din câmpul de etichetă vor fi relative la valoarea calculată a expresiei din câmpul de argument al directivei. Toate simbolurile care urmează după apariția unei directive **PHASE** sînt marcate ca simboluri absolute, valoarea lor rămînd nemodificată în cursul operației de linkeditare a modulului obiect relocabil. Din acest motiv, codul obiect poate fi executat numai în spațiul de adrese definit de directivele **PHASE** din program. Numărul directivelor nu este limitat; următoarea directivă **PHASE** nu ține seama de existența directivei precedente. Expresia din câmpul de argument nu poate conține simboluri nedefinite încă, sau referințe externe.

Directivele utilizator, specifice microcalculatorului AMIC, implementează în cadrul limbajului de asamblare, apelurile funcțiilor utilizator (v. cap. 5, paragraf 5.3: „Funcții utilizator — descriere și utilizare”). Toate directivele utilizator generează secvența de instrucțiuni:

LD C, nr

CALL 5

în care „nr” reprezintă codul de apel al funcției respective.

Reset System — inițializare sistem

Sintaxa :

RESET

Efect : inițializare sistem, prin salt la adresa 0000H

Console Input — citire caracter de la consolă

Sintaxa :

CONIN

Efect : citirea unui caracter de la consolă și returnarea codului caracterului în registrul A.

Console Output — scriere caracter la consolă

Sintaxa :

CONOUT

Efect : scrierea caracterului încărcat în prealabil în registrul E, la consolă.

Reader Input — citire caracter de la interfața serială

Sintaxa :

RIN

Efect : citirea unui caracter de la interfața serială și returnarea codului caracterului în registrul A.

Punch Output — scriere caracter la interfața serială

Sintaxa :

POUT

Efect : codul caracterului, încărcat în prealabil în registrul E, este emis la interfața serială.

List Output — scriere caracter la miniimprimantă

Sintaxa :

LOUT

Efect : codul caracterului, încărcat în prealabil în registrul E, este emis la miniimprimantă.

Init TV Screen — inițializare memorie ecran TV

Sintaxa :

INITV

Efect : umplerea memoriei ecran (4000H-5FFFH) cu valoarea OFFH, ceea ce are ca efect ștergerea ecranului.

Cassette I/O Driver — apel driver casetă audio

Sintaxa :

CASSDR

Efect : apelul driverului de casetă audio, având în prealabil încărcată adresa descriptorului de fișier în perechea de registre DE.

Beep — emiteră sunet în difuzor

Sintaxa :

BEEP

Efect : emiteră sunet în difuzor, având în prealabil încărcată durata în registrul D și frecvența în registrul E.

Write String — scriere șir caractere la consolă

Sintaxa :

WSTRIN

Efect : scrierea la consolă a șirului de caractere a cărui adresă de început s-a încărcat în prealabil în perechea de registre DE.

Read and Edit String — citire și editare buffer, consolă

Sintaxa :

RSTRIN

Efect : citirea și modificarea unui șir de caractere de la consolă în bufferul de memorie a cărui adresă de început s-a încărcat în prealabil în perechea de registre DE.

Console Status — obținere stare consolă

Sintaxa :

CSTS

Efect : verifică dacă s-a introdus un caracter de la consolă și returnează OFFH în registrul A în caz afirmativ.

Graphic Primitives — apel primitive grafice

Sintaxa :

PGRAF

Efect : apelul primitivelor de setare, resetare și testare pixel, având în prealabil încărcate registrele D și E cu coordonatele pixelului și registrul B cu codul operației.

Utilizarea directivelor enumerate mai sus simplifică efortul de programare, eliminând totodată și riscul apelurilor incorecte ale funcțiilor utilizator. Nu s-au introdus și secvențele de încărcare parametri de apel (sub formă de argument în cadrul directivelor), pentru a permite apelul dinamic al funcțiilor utilizator în cadrul subrutinelor.

Evaluarea expresiilor din câmpul de argument

O expresie din câmpul de argument poate conține cel mult doi operanzi. Operanzii acceptați de asamblorul ASR—Z80 pot fi :

- a) nume de registre : A, B, C, D, E, H, L, I, R.
- b) nume de perechi de registre : AF, BC, DE, HL
- c) nume de registre index : IX, IY
- d) pointerul stivei : SP
- e) indicatorii de condiții : NZ, Z, NC, C, PO, PE, P, M.
- f) contorul de program : \$
- g) constante numerice :

— zecimale (opțiune implicită); ex. : 35, 144, 03

— hexazecimale (încep obligatoriu cu o cifră 0—9 și se termină cu litera H); ex. 0AFH, 13EH

— tip caracter (un caracter ASCII inclus între apostrofuri); ex. 'B', 'F'.

Orice constantă numerică trebuie să aibă valoarea cuprinsă în domeniul 0-0FFFFH. Dacă după evaluare un număr depășește 2 octeți, depășirea este ignorată și rezultatul trunchiat pe 16 biți. O constantă tip caracter se convertește la valoarea codului ASCII corespunzător (ex. : 'B'=42H).

h) simboluri care apar în câmpul de etichetă sau în directiva EXTRN. Simbolul „\$” reprezintă întotdeauna valoarea contorului de program înainte de asamblarea instrucțiunii curente.

Operatorii acceptați de ASR—Z80 sînt :

- (operație unară : generare complement față de 2)
- +, — (adunare și scădere)

Operatorii se admit numai pentru grupele f, g și h de operanzi. Se va ține seama de următoarele reguli :

- 1) <constantă> <op> <constantă> = <constantă>
- 2) <constantă> <op> <simbol> = <simbol>
- 3) <simbol> <op> <constantă> = <simbol>
- 4) <simbol> — <simbol> = <constantă>
- 5) <simbol> + <simbol> = operație interzisă
- 6) simbolurile externe nu sînt admise în cadrul expresiilor
- 7) domeniul de valori permis pentru o expresie depinde de contextul în care este utilizată (valoare pe 1 sau 2 octeți); asamblorul generează un mesaj de eroare la depășirea domeniului admis.
- 8) salturile relative admit un domeniu cuprins între —126 și +129 octeți.

Modul de utilizare a asamblorului ASR—Z80]

După lansarea în execuție a asamblorului prin comanda A, se solicită operatorului opțiunile de lucru :

a) tip modul generat

CODE (A/R) : r

- dacă r=A se generează modul obiect absolut
- dacă r=R se generează modul obiect relocabil

b) mod lucru

MODE (I/M/F) : r

— dacă r=I, asamblorul lucrează în mod interpretativ, utilizatorul introducînd programul sursă linie cu linie de la consolă, fiecare linie fiind asamblată pe loc. Se semnalează erorile de sintaxă ; dacă se utilizează simboluri nedefinite încă, cîmpul corespunzător din codul obiect al instrucțiunii generate va avea valoarea 0000H.

— dacă r=M, se solicită adresa de început a programului sursă (max. 4 cifre hexazecimale) :

SRC ADD : XXXX (CR)

— dacă r=F, se solicită numele fișierului ce conține programul sursă :

FN : XXXX (CR)

În acest caz, încărcarea sursei și generarea codului obiect se execută la adrese de memorie cunoscute de asamblor și comunicate utilizatorului după încărcarea fișierului prin mesajul :

S : XXXX C : YYYY

unde XXXX=adresa de încărcare sursă ; YYYY=adresă generare modul obiect.

Pentru modul I sau M, se solicită operatorului adresa de început generare modul obiect :

OBJ. CODE : XXXX (CR)

c) extensie tabelă simboluri

EXT : r

— dacă $r=(CR)$ se utilizează tabela standard, care permite aprox. 300 de simboluri în programul sursă

— dacă $r=(\text{orice alt caracter})$, se utilizează tabela extinsă, care permite aprox. 700 de simboluri în programul sursă.

d) listing asamblare

LST : r

— dacă $r=(CR)$ se va genera listingul asamblării și se va afișa la consolă.

— dacă $r \neq (CR)$ (orice alt caracter), nu se generează listing de asamblare.

Formatul listingului afișat este :

E XXXX CCCCCCCC SS...S

în care :

E — literă indicînd tipul erorii, sau blank dacă linia sursă este corectă

XXXX — contorul curent de amplasare al codului obiect generat

CC... — codul obiect generat

SS... — linia sursă

La terminarea operației de asamblare, se emite mesajul :

NN ERRORS ; OBJ.CODE : XXXX YYYY

în care

XXXX — adresa de început a modulului obiect

YYYY — adresa de sfîrșit a modulului obiect (adresa ultimului octet al modulului)

Reguli de bază :

— dacă s-a cerut generare de modul obiect absolut, acesta este generat relativ la adresa dată în mesajul :

OBJ.CODE : XXXX (opțiunea I sau M)

...C : YYYY (opțiunea F)

Programul este executabil de la adresa **XXXX** sau **YYYY** ; deoarece adresa **YYYY** depinde întotdeauna de lungimea fișierului sursă citit, se impune utilizarea unei directive **PHASE** în programul sursă, specificînd astfel spațiul de adrese în care programul este executabil.

Rezultă că în modul F de lucru, modulul obiect absolut generat la adresa **YYYY** va trebui transferat printr-o comandă de tip M (Move Memory) la adresa specificată în directiva **PHASE** din program. În modul I sau M de lucru nu se va utiliza directiva **PHASE** în program, modulul obiect fiind gata de lansare în execuție de la adresa **XXXX**.

— dacă s-a cerut generare de modul obiect relocabil, codul obiect generat nu este executabil de la adresa **XXXX** sau **YYYY** ; această adresă va fi comuni-

cată linkeditorului ca parametru de intrare (v. paragr. 6.2.5) pentru a genera modulul absolut din cel relocabil.

— utilizatorul va salva pe casetă magnetică modulul obiect printr-o comandă K, pe baza adreselor obținute din mesajul:

NN ERRORS ; OBJ.CODE : XXXX YYYY

Valorile XXXX și YYYY vor constitui parametrii „adresă inferioară” și „adresă superioară” solicitați de comanda K. De remarcat că pentru modulele obiect relocabile, spațiul XXXX—YYYY conține și tabelele necesare relocării și editării de legături.

— dacă s-a cerut generare de modul obiect absolut, la sfârșitul asamblării se va emite mesajul:

UNRESOLVED REF'S

urmat de lista simbolurilor nerezolvate, (dacă există) sub forma :

simbol 1 :

simbol 2 :

.

.

simbol n :

După fiecare nume de simbol, utilizatorul poate răspunde (CR) (simbolul rămîne nerezolvat, cu valoarea 0000H) sau va introduce valoarea absolută (max. 4 cifre hexa) care va rezolva astfel toate referințele la simbolul în cauză. Acest mod de lucru permite punerea la punct rapidă a programelor care apelează subrutine sau alte module de program nedefinite încă în sursa asamblată; referințele nerezolvate pot fi astfel interceptate într-un singur loc („dummy routine”) verificînd în cursul testării doar trecerile prin punctele de apel ale porțiunilor încă nedefinite.

Lista erorilor emise de asamblorul ASB-Z80

Primul caracter dintr-o linie sursă eronată va conține o literă cu semnificația :

- A — argument incompatibil cu tipul instrucțiunii
- B — sintaxă eronată a unei expresii
- C — comandă (mnemonic instrucțiune) eronată
- D — simbol dublu definit
- E — etichetă eronată
- F — tabelă de simboluri plină
- P — dublu registru incompatibil cu tipul instrucțiunii
- R — registru incompatibil cu tipul instrucțiunii
- S — sintaxă eronată a liniei sursă
- T — directivă eronată
- U — simbol nedefinit
- V — valoare operand sau expresie mai mare decît o permite tipul instrucțiunii
- X — utilizare simbol extern în expresii

Eroarea de tip „F” este singura eroare care duce la abandonarea procesului de asamblare. Se emite mesajul:

F ABORT

și se redă controlul monitorului. Utilizatorul poate relansa asamblorul cu opțiunea de „tabelă simboluri extinsă” sau poate diviza programul sursă în mai multe module, care să nu ducă la depășirea numărului maxim admis de simboluri.

Restricții de utilizare a asamblorului ASR-Z80

Versiunea actuală a asamblorului implementată în monitorul DEST prezintă următoarele restricții:

1) în cadrul unei asamblări se tratează un singur program sursă; opțiunea de asamblare a mai multor programe sursă într-un singur modul obiect este destinată dezvoltărilor ulterioare ale asamblorului.

2) nu se efectuează nici un control asupra spațiului de memorie disponibil în cursul generării modului obiect; din acest motiv, se recomandă, la modul de lucru cu opțiunea M, specificarea adresei de început modul obiect după adresa de sfârșit program sursă.

3) zona de lucru în memoria RAM solicitată de asamblor este cuprinsă între 6100H-70FFH cu tabelă standard de simboluri, și 6100H-80FFH cu tabelă de simboluri extinsă. Utilizatorul își va gestiona în consecință zona de memorie.

Cu titlu pur orientativ, se poate aprecia că lungimea maximă a unui program sursă nu va depăși 20—24 Ko, iar cea a unui modul obiect 8—10 Ko. Utilizarea unor programe sursă de dimensiuni mari îngreunează sensibil urmărirea rezultatelor asamblării și punerea la punct a programului obținut. Recomandăm o modularizare ierarhică pe funcții specifice a unei aplicații date, astfel încât lungimea codului obiect corespunzătoare unui modul de program să se situeze în jurul valorii optime de 1—4 Ko.

6.2.2. Comanda E — „edit source program”

Formatul comenzii este:

E <adinf> [, <adsup>] <CR>

în care:

<adinf> — adresa de memorie de la care începând se va genera programul sursă

<adsup> — limita maximă superioară de memorie alocată pentru generarea sursei; în absență, se consideră implicit sfârșitul memoriei RAM.

Comanda E lansează în execuție editorul de text rezident **EDR-Z80**, care permite utilizatorilor crearea și modificarea programelor sursă. Editorul acceptă ca intrare orice fel de text sursă, indiferent de conținut, dar scopul său principal este crearea fișierelor sursă pentru aplicațiile în limbaj de asamblare Z80.

Definirea termenilor

TEXT SURSĂ — zonă de memorie cuprinzând cel puțin o linie sursă, și terminată cu un octet ce conține valoarea OFFH .

LINIE SURSĂ — șir de caractere ASCII cuprins între două caractere (CR) sau un caracter (CR) și OFFH .

BUFFER INTRARE — zonă de memorie tampon pentru introducerea de la consolă a liniilor sursă. Lungimea bufferului de intrare este de 80 de caractere, valoare care limitează superior dimensiunea unei linii sursă.

LINIE CURENTĂ — linia sursă la care se referă orice acțiune a editorului.

Comenzile acceptate de editorul de texte EDR-Z80

După lansarea în execuție, editorul emite mesajul:

NEW : r

— dacă $r = (\text{CR})$, editorul consideră că zona de memorie cu decontul la $\langle \text{adinf} \rangle$ este vidă, și începe generarea unui text sursă nou.

— dacă $r \neq (\text{CR})$ (orice alt caracter), editorul consideră că în respectiva zonă de memorie se află deja un text sursă (încărcat în prealabil de pe casetă magnetică, sau rămas de la o sesiune de lucru anterioară) și verifică dacă primul caracter de la adresa $\langle \text{adinf} \rangle$ este ODH (CR). În caz afirmativ, se calculează lungimea textului deja introdus și se intră în starea de așteptare comenzi. În caz negativ, se emite mesajul:

NO SOURCE

și se predă controlul monitorului.

Starea de așteptare comenzi este marcată de apariția prompterului „>” pe ecran.

Comanda A — „append source text“

Formatul comenzii este:

>XXXX (CR)

în care:

XXXX — numele fișierului (max. 4 cifre hexa) care conține textul sursă de introdus.

Comanda A adaugă la sfârșitul textului sursă din memorie, textul sursă citit din fișierul de pe caseta magnetică, deplasând corespunzător marca de sfârșit sursă (octetul ce conține OFFH).

Comanda B — „point to begin/end of source“

Formatul comenzii este:

>±B(CR)

Comanda +B marchează drept linie curentă, prima linie din textul sursă. Comanda -B marchează drept linie curentă, ultima linie din textul sursă.

Comanda I — „insert lines“

Formatul comenzii este :

>I(CR)

Comanda I permite inserarea de linii sursă de la consolă în textul sursă ; inserarea se face întotdeauna după linia curentă.

În timpul operației de inserare, se afișează în zecimal (0000—9999) numărul liniei respective. Acest număr nu se introduce în textul sursă, fiind utilizat doar pentru regăsirea simplă a informațiilor de către utilizator.

Introducerea unei linii sursă se termină fie la apăsarea tastei (CR), fie la depășirea lungimii maxime de 80 de caractere.

Utilizatorul poate corecta șirul de caractere introdus, înainte de apăsarea tastei (CR), utilizând tastele :

DEL : șterge ultimul caracter introdus în bufferul de intrare. **CTRL-X** : șterge întreaga linie în bufferul de intrare.

La apăsarea tastei (CR), conținutul bufferului de intrare este transferat în textul sursă, utilizatorul nemaiavând acces la linia sursă introdusă decât după ieșirea din comanda de inserare.

Ieșirea din comanda de inserare se face prin apăsarea tastei **CTRL-Z** ; editorul intră din nou în starea de așteptare comenzi.

În timpul inserării, dacă s-a specificat la lansarea editorului o limită superioară maximă admisă pentru zona de text sursă, și dacă această limită a fost atinsă, se emite mesajul :

MEM FULL

Editorul reîntră în starea de așteptare comenzi ; prin comanda **E** se determină mărimea zonei ce conține textul sursă, utilizatorul putând astfel salva pe casetă textul introdus. Menționăm că, în acest caz, ultima linie sursă introdusă de la consolă se pierde.

Comanda K — „kill lines“

Formatul comenzii este :

> ±nK(CR)

în care :

n — număr zecimal, în domeniul 0000—9999

Comanda **nK** șterge **n** linii de după linia curentă (inclusiv linia curentă) ; comanda **-nK** șterge **n** linii dinaintea liniei curente (inclusiv linia curentă).

Comanda T — „type lines“

Formatul comenzii este :

> ±nT(CR)

în care :

n — număr zecimal, în domeniul 0000—9999

Comanda **nT** afișează **n** linii de după linia curentă ; comanda **-nT** afișează **n** linii dinaintea liniei curente.

Atît la comanda **K**, cît și la comanda **T**, dacă **n** nu este specificat, se consideră implicit **n=1** ; dacă **n=0**, nu se execută comanda.

Comanda F — „find character string“

Formatul comenzii este

Fccc...c(CR)

Comanda **F** caută șirul de caractere **ccc...c**, începînd cu linia curentă și pînă la sfîrșitul textului sursă. Căutarea se face linie cu linie, începînd cu primul caracter al fiecărei linii. Numărul maxim admis de caractere din șir este 16. Dacă șirul **n-a** fost găsit, se emite mesajul :

NO STRING

și se reintră în starea de așteptare comenzi.

Dacă șirul a fost găsit, se afișează numărul liniei respective, iar linia în cauză devine linie curentă.

Comanda n : — „point to line number“

Formatul comenzii este :

>n : (CR)

în care :

n — număr zecimal, în domeniul 0000—9999

Comanda are ca efect poziționarea pe linia cu numărul **n** ; acesta devine linia curentă. Dacă s-a atins sfîrșitul textului sursă, fără a detecta numărul de linii dat, se emite mesajul :

NO LINE

și se reintră în starea de așteptare comenzi.

Valoarea **n ≠ 0** inhibă execuția comenzii.

Comanda E — „end of session“

Formatul comenzii este :

>E(CR)

Comanda **E** semnifică sfîrșitul sesiunii de lucru cu editorul de text. Se emite mesajul :

SRC CODE : XXXX YYYY

în care :

XXXX — adresa de început a textului sursă

YYYY — adresa de sfîrșit a textului sursă (adresa markerului de sfîrșit text)

Zona respectivă de memorie poate fi salvată sub formă de fișier pe casetă magnetică, și reîncărcată pentru editări ulterioare sau pentru asamblare cu ASR-Z80.

Restricții ale editorului de texte EDR-Z80

Versiunea actuală a editorului de texte implementată în monitorul DEST prezintă următoarele restricții:

1) editarea unei linii la nivel de caracter se poate executa numai în faza de introducere de la consolă; toate comenzile editorului efectuează prelucrări la nivel de linie, exceptând comanda F.

2) comanda A nu verifică depășirea limitei superioare impuse zonei de text sursă; în scopul evitării unor eventuale distrugerii de zone de memorie, se va utiliza înainte de lansarea editorului o comandă N (v. cap. 5 paragr. 5.2) pentru a determina lungimea fișierului sursă de încărcat.

6.2.3. Comanda P — „list disassembled code“. Formatul comenzii este următorul:

P <adinf>, <adsup>[, <adtxt>] <CR>

în care:

<adinf> — adresa inferioară a zonei de memorie de dezasamblat
 <adsup> — adresa superioară a zonei de memorie de dezasamblat
 <adtxt> — adresa de început a zonei de memorie în care se vor depune liniile sursă rezultate în urma dezasamblării.

Comanda P lansează în execuție dezasamblorul rezident DSR-Z80, care creează un text sursă analog listingului de asamblare, pe baza decodificării codurilor instrucțiunilor aflate în spațiul de adrese limitat de <adinf> și <adsup>. Textul sursă este listat la consolă pe măsură ce se creează; dacă în comandă s-a specificat și adresa <adtxt>, fiecare linie sursă listată va fi memorată succesiv începând cu adresa dată.

Formatul unei linii sursă este:

XXXX CCCCCCCC MMMM AAAAAAAAAA

în care:

XXXX — adresa instrucțiunii decodificate
 CC... — codul obiect al instrucțiunii
 MMMM — mnemonica instrucțiunii
 AA... — zona de argument a instrucțiunii (dacă există).

Codul obiect, adresa instrucțiunii și constantele din zona de argument sînt prezentate ca numere hexazecimale.

Pentru a putea utiliza textul sursă astfel obținut ca fișier de intrare pentru asamblorul ASR-Z80, memorarea fiecărei linii în zona de text se execută astfel:

MMMM AAAAAAAAAA; XXXX CCCCCCCC

Acest format permite inserarea ușoară de linii sursă conținând doar simbolurile din câmpurile de etichetă ce se vor atribui instrucțiunilor apelate prin

salt sau apel cu revenire în secvența de program dezasamblată. După o primă asamblare a textului sursă astfel modificat, compararea zonei de comentariu generată de dezasamblor cu contorul de amplasare generat de asamblor simplifică în mare măsură operația de verificare/modificare a programului sursă.

Linii sursă generate încep cu caracterul (CR) în prima coloană, urmat de un blank. După ultima linie sursă generată, dezasamblorul introduce un octet cu valoarea 0FFH, permițând utilizatorului lansarea editorului de text în vedere a listării și/sau prelucrării sursei obținute.

Deoarece codurile instrucțiunilor microprocesorului Z80 ocupă toate valorile din domeniul 0-0FFH, adresa de început dezasamblare va trebui să specifice un cod de început instrucțiune; în caz contrar, rezultatul dezasamblării este lipsit de sens. Același efect îl are și întâlnirea unei zone de date în cadrul spațiului de memorie de dezasamblat, ceea ce duce la desincronizarea dintre contorul de amplasare real al codurilor de instrucțiuni ce urmează zonei de date, și contorul de avans în memorie al dezasamblorului.

6.2.4. Comanda T — „trace flow of execution”. Formatul comenzii este următorul:

T [**<adinf>**] [**<nbtr>**]₂ **<CR>**

în care:

<adinf> — adresa de început a secvenței de program de trasat

<nbtr> — număr total de instrucțiuni de trasat (max. 4 cifre hexa)

Comanda T permite execuția monitorizată pas cu pas (instrucțiune cu instrucțiune) a unei secvențe de program în cod obiect stocată în memoria RAM sau EPROM.

Pentru fiecare instrucțiune executată și trasată, se afișează la consolă informațiile următoare:

XXXX CCCCCCC MMMM AAAAAAAAAA

Semnificațiile fiecărui câmp sînt identice cu cele din linia sursă listată de comanda de dezasamblare. Adresa XXXX reprezintă adresa absolută din memorie a instrucțiunii executate.

Dacă nu se specifică nici un parametru, se trasează execuția unei singure instrucțiuni, a cărei adresă se găsește în locația USERPC din zona de variabile monitor (v. cap. 5). Conținutul acestei locații este completat de funcția G cu punct de întrerupere și reprezintă adresa instrucțiunii la care s-a oprit execuția programului utilizator prin comanda G.

Dacă se specifică parametrul **<nbtr>** se va trasa numărul dat de instrucțiuni, începînd cu adresa sus-menționată.

Dacă parametrul **<adinf>** este prezent, se va trasa execuția uneia sau mai multor instrucțiuni (funcție de valoarea **<nbtr>**).

Menționăm că în acest caz, adresa **<adinf>** trebuie să specifice un început corect de instrucțiune, altfel rezultatul trasării este lipsit de sens.

După trasarea execuției numărului dat de instrucțiuni, se revine în monitor utilizatorul putînd să examineze și să modifice conținutul registrelor prin

comanda **X**, să afișeze și să modifice conținutul zonelor de memorie prelucrate în secvența trasată prin comenzile **D** și **S** etc.

Un program executabil stocat în memoria RAM poate fi lansat în execuție pînă la o anumită adresă prin comanda **G** cu punct de întrerupere, apoi poate fi executat pas cu pas prin comenzi **T** în secvențele suspectate a fi eronate. Procesul de testare și punere la punct prin utilizarea combinată a comenzilor **G**, **T**, **S** și **X** reprezintă modul cel mai eficient de aducere a unui program la parametrii funcționali proiectați.

Un program executabil stocat în memoria EPROM poate fi lansat în execuție prin comanda **G**, dar nu poate fi întrerupt decît prin apăsarea tastei **INT** (întrerupere nemascabilă). Din acest motiv, trasarea execuției unui astfel de program se va efectua exclusiv cu comanda **T**, prima lansare fiind de forma **T** <adresă>, unde <adresă> reprezintă adresa primei instrucțiuni din secvența de trasat aflată în memoria EPROM. Vor urma apoi comenzi **T(CR)** succesive, neputîndu-se evita porțiuni deja testate prin comenzi de tip **G**. Și în acest caz se pot utiliza comenzile **X**, **S** sau **D** între două lansări succesive ale comenzii de trasare.

6.2.5. Comanda Q — „relocate and link object modules“. Formatul comenzii este următorul :

.Q<adinf> [<adsup>] <CR>

în care :

<adinf> — adresa de la care va fi executabil modulul obiect absolut creat

<adsup> — adresa de început a modulului obiect relocabil.

Comanda **Q** lansează în execuție editorul de legături rezident **LRR-Z80**, care permite crearea unui modul obiect absolut (program executabil), pornind de la unul sau mai multe module obiect relocabile, obținute prin asamblări separate de fișiere sursă.

Se oferă astfel posibilitatea dezvoltării și testării modulelor aplicațiilor : un program de dimensiuni mari poate fi conceput sub forma unui set de module sursă de dimensiuni relativ reduse, testate separat și integrate într-un singur modul obiect după faza de punere la punct a fiecăruia. Ciclul de dezvoltare—testare a unei aplicații date poate fi scurtat astfel în mod semnificativ.

Relocarea modulelor obiect se execută relativ la adresa <adinf> ; codul obiect relocat va fi executabil în spațiul de adrese [adinf—(adinf + sumă lungimi module obiect)].

Dacă parametrul <adsup> este specificat, editorul **LRR-Z80** va efectua numai operația de relocare a unui singur modul obiect, încărcat în prealabil la adresa <adsup>. La sfîrșitul relocării, se emite mesajul :

OBJ.CODE : XXXX\YYYY

în care :

XXXX — adresa de început a modulului obiect absolut generat (coincide cu <adsup>)

YYYY — adresa de sfîrșit a modulului obiect absolut generat.

Pentru a putea fi executat, modulul obiect va fi transferat printr-o comandă de tip M (Move Memory) la adresa pentru care s-a cerut relocarea (adresa <adinf>). Din punctul de vedere al editorului de legături, adresele <adinf> și <adsup> pot fi identice sau diferite; prima semnifică modificarea corespunzătoare a cîmpurilor de adrese pe 16 biți din codul obiect relocabil, iar cea de-a doua semnifică adresa unde se află codul obiect de relocat.

Fiind tratat un singur modul relocabil, după relocare acesta rămîne în același spațiu de memorie, modificîndu-se doar conținutul său. Dacă <adinf> = <adsup>, modulul obiect absolut poate fi executat de la această adresă, fără a mai fi transferat în altă zonă de memorie.

Dacă parametrul <adsup> este absent, editorul de legături va efectua atît operația de relocare, cît și cea de linkeditare pentru un număr oarecare de module obiect relocabile, stocate sub formă de fișiere pe casetă magnetică.

Se efectuează mai întîi operația de încărcare a fișierelor solicitîndu-se numele fiecărui fișier (un fișier reprezentînd un modul obiect relocabil, salvat pe casetă după o operație de asamblare text sursă) prin mesajul:

FN : XXXX(CR)

Încărcarea fișierelor se face începînd de la adresa fixă 8100H; numărul maxim de fișiere încărcabile este limitat doar de spațiul de memorie RAM disponibil.

Pentru fiecare fișier, se solicită numele acestuia; dacă se răspunde (CR), editorul consideră terminată încărcarea de fișiere și începe operația de relocare—linkeditare a modulelor încărcate.

Primul modul va fi relocat relativ la adresa <adinf>; următorul modul va fi relocat relativ la adresa (adinf + lungime) modul precedent.

Se efectuează astfel concatenarea tuturor modulelor relocabile într-un singur modul absolut, executabil în spațiul de adrese [adinf—(adinf + suma lungimilor tuturor modulelor)].

După relocare, se execută operația de linkeditare, rezolvînd referințele externe din cadrul fiecărui modul. Sfîrșitul linkeditării este marcat de apariția mesajului:

OBJ.CODE : XXXX YYYY

în care

XXXX — adresa început modul obiect absolut generat (întotdeauna 8100H)

YYYY — adresă sfîrșit modul obiect absolut generat

În cadrul operației de linkeditare pot apare mesaiele

a) **DOUBLE DEF'S**

Simbol 1

Simbol 2

.

.

.

Simbol n

Mesajul afișează lista simbolurilor globale dublu definite (același nume de simbol apare în două directive ENTRY)

b) UNRESOLVED REF'S

simbol 1

simbol 2

.

.

.

simbol n

Mesajul afișează lista referințelor externe nerezolvate (simbolurile din listă nu apar în nici o directivă ENTRY)

c) F ABORT

Mesajul semnifică umplerea tabelii de simboluri globale și referințe externe. Procesul de linkeditare este abandonat ; se revine în monitor în starea de așteptare comenzi.

Modulul obiect absolut obținut după relocare-linkeditare va fi transferat la adresa <adinf> pentru a putea fi executat. Dacă la lansarea comenzii s-a specificat <adinf>=8100H, transferul nu mai este necesar.

Restricții ale editorului de legături LRR-Z80

Versiunea actuală a editorului de legături LRR-Z80 prezintă următoarele restricții :

1) relocarea modulelor se face la nivel de pagină de memorie (multiplu de 256 de octeți) ; acesta implică :

- adresele <adinf> și <adsup> vor fi multiplu de 256 (100H), în caz contrar, editorul ia în considerare prima adresă multiplu de 256 imediat superioară.
- programele obiect obținute sînt executabile începînd de la adresa multiplu de 256.
- la relocarea mai multor module obiect, între sfîrșitul unui modul și începutul următorului pot apărea spații neutilizate de pînă la 255 octeți. Rezolvarea referințelor externe este efectuată corect, utilizatorul fiind însă avizat că orice cod obiect corespunzător unui modul relocabil debutează în modulul absolut la adresă multiplu de 256.

2) la încărcarea de pe casetă a fișierelor cu module obiect relocabile nu se efectuează verificarea depășirii unei limite superioare de memorie. Utilizatorul își va gestiona în consecință spațiul de memorie RAM disponibil.

Sistemul de operare rezident MATE (Monitor-Asamblor-Text-Editor)

7.1. Generalități

MATE este un sistem software proiectat pentru a funcționa pe microcalculatorul personal „aMIC”. Sistemul conține un modul monitor, un asamblor și un editor de fișiere.

Pentru a folosi sistemul MATE sînt necesari 6 Ko de memorie (0000-17FF). Memoria adițională este necesară pentru a stoca fișierele sursă și obiect ale utilizatorului.

Comunicația între utilizator și sistemul MATE se realizează prin intermediul tastaturii și ecranului televizorului.

Adresa de inițializare a sistemului software MATE este 0000. Pentru a porni sistemul fără a-l inițializa se folosește adresa 0004.

7.2. Comenzile modulului monitor

- CTRL-X** — abandonează linia curentă,
- ENTR** — introduce date în memorie,
- DUMP** — afișează conținutul memoriei,
- FILE** — creează, distruge, activează un fișier sau afișează informații referitoare la un fișier,
- EXEC** — lansează în execuție un program,
- ASSM** — assemblează un program sursă,
- LIST** — listează conținutul unui fișier,
- DELT** — șterge linii dintr-un fișier,
- YYYY** — cheamă editorul de fișier ($0 \leq Y_i \leq 9$),
- PAGE** — translatează o pagină de memorie,
- BREK** — poziționează sau șterge puncte de întrerupere,
- PROC** — relansează în execuție un program oprit la un punct de întrerupere,
- SAVE** — salvează pe casetă magnetică un fișier din memorie,
- LOAD** — citește în memorie un fișier de pe casetă magnetică.

Modulul monitor conține un singur mesaj de eroare (...WHAT? ...), care indică o comandă eronată sau folosirea incorectă a parametrilor unei comenzi.

7.3. Formatul comenzilor modulului monitor

ENTR AAA (CR)

Această comandă este folosită pentru a introduce date (în format hexazecimal) în memorie începând de la adresa AAAA. Introducerea datelor este terminată prin/(CR). Exemplu:

ENTR 1900 (CR)

0A 30 FF F5/(CR)

DUMP AAAA BBBB (CR)

Comanda **DUMP** este folosită pentru a examina conținutul locațiilor de memorie între adresele AAAA și BBBB. Afișarea conținutului memoriei se face în format hexazecimal, fiecare linie afișată conținând pînă la 8 octeți. Dacă parametrul BBBB nu este indicat, numai conținutul locației AAAA va fi afișat.

FILE / NAME / AAAA (CR)

Această comandă este folosită pentru a crea un fișier (în memorie) cu numele NAME și cu adresa de început AAAA. Dacă mai există în sistem un fișier cu numele NAME, modulul monitor va emite mesajul de eroare NO NO.

FILE / NAME / (CR)

În urma acestei comenzi fișierul cu numele NAME devine fișierul „curent”. La orice moment de timp, cel mult unul dintre fișierele create în cadrul sistemului MATE poate fi fișierul „curent”.

FILE / NAME / 0 (CR)

Fișierul cu numele NAME este distrus.

FILE (CR)

Parametrii fișierului „curent” sînt afișați. Parametrii unui fișier sînt adresa sa de început și adresa sa de sfîrșit.

FILES (CR)

Parametrii tuturor fișierelor existente în sistem sînt afișați.

LIST N (CR)

Această comandă este folosită pentru a afișa liniile fișierului „curent” începînd cu linia numărul N. Dacă parametrul N nu este specificat afișarea liniilor începe cu linia de început a fișierului. Afișarea conținutului poate fi oprită introducînd caracterele C|RL-X.

DELT L1 L2 (CR)

Această comandă este folosită pentru a șterge din fișierul „curent” liniile L1 pînă la L2, inclusiv. Dacă parametrul L2 nu este specificat, numai linia L1 este ștersă.

PAGE AAAA BBBB (CR)

Această comandă este folosită pentru a transla conținutul paginii de memorie cu adresa de început AAAA (256 de octeți) în pagina de memorie cu adresa de început BBBB.

BREK AAAA (CR)

Această comandă poziționează un punct de întrerupere (break point) la adresa AAAA. Când execuția programului ajunge la adresa AAAA, punctul de întrerupere este șters, toate registrele sînt salvate și se emite mesajul „AAAA BREAK“. Apoi, controlul este cedat modulului monitor. Registrele sînt salvate în următoarele locații (și deci conținutul lor poate fi examinat și modificat folosind comenzile DUMP și respectiv ENTR):

1000	PSW	1006	SP (low)
1001	A	1007	SP (high)
1002	C	1008	L
1003	B	1009	H
1004	E	100A	PC (low)
1005	D	100B	PC (high)

Restricții :

- (1) Se pot poziționa simultan maximum 8 puncte de întrerupere
- (2) Puncte de întrerupere nu pot fi poziționate între adresele 0000-0040.

Dacă parametrul AAAA nu este specificat, toate punctele de întrerupere deja poziționate sînt șterse.

PROC AAAA (CR)

Această comandă este folosită pentru a relansa în execuție un program de la un punct de întrerupere. Toate registrele sînt refăcute și execuția programului continuă de la locația AAAA. Dacă parametrul AAAA nu este specificat execuția continuă de la adresa conținută în registrul PC.

ASSM AAAA BBBB (CR)

Programul sursă conținut în fișierul „curent“ este asamblat de către asamblorul rezident. Asamblarea se efectuează asignînd adrese în codul obiect începînd cu adresa AAAA. În pasul al doilea, codul obiect este plasat în memorie începînd de la adresa BBBB. Dacă parametrul BBBB nu este specificat, se presupune că BBBB=AAAA. Pe parcursul asamblării se produce un listing complet.

ASSME AAAA BBBB (CR)

Comanda ASSME funcționează analog cu comanda ASSM cu excepția faptului că nu se produce un listing complet, ci se afișează numai liniile programului sursă conținînd erori.

7.4. Editorul de fișiere

Fișierele pe care utilizatorul le poate crea în memoria principală sînt organizate pe linii. Fiecare linie a unui fișier este identificată printr-un număr de linie N, unde $0000 \leq N \leq 9999$ (zecimal). Editorul de fișiere permite încăr-

careia informaţiei (liniilor) în fişiere precum şi modificarea conţinutului (liniilor) fişierelor. Editorul de fişiere acţionează întotdeauna asupra fişierului „curent”.

Pe măsură ce utilizatorul introduce linii de la echipamentul periferic de intrare, editorul plasează liniile respectiv în spaţiul de memorie al fişierului „curent” în ordinea indicată de numerele de linii corespunzătoare. Rearanjarea liniilor în fişiere în ordinea crescătoare a numerelor de linii este efectuată în mod automat de modulul editor. În cazul în care utilizatorul introduce o linie cu numărul N' , unde N' este numărul unei linii deja existente în fişier, noua linie N' va înlocui vechea linie N' .

Editorul nu asignează în mod automat numere de linii. Utilizatorul trebuie să introducă mai întâi numărul liniei urmat apoi de conţinutul liniei respective. Numerele de linie valide trebuie să conţină 4 cifre zecimale. Introducerea unei linii este terminată prin caracterul (CR). O linie poate conţine cel mult 80 de caractere.

7.5. Asamblorul

Asamblorul acţionează asupra fişierului „curent”. Conţinutul liniilor fişierului „curent” este translatat în cod obiect. Al doilea caracter care urmează numărului liniei este considerat drept primul caracter al codului sursă. În consecinţă, caracterul imediat următor numărului liniei trebuie să fie în mod normal un blank. Numerele liniilor nu sînt procesate de către asamblor dar sînt reproduse în listing.

7.5.1. Instrucţiunile limbajului de asamblare. Instrucţiunile limbajului de asamblare sînt fie instrucţiuni maşină ale microcalculatorului „aMIC” sub formă simbolică *, fie pseudoinstrucţiuni. Structura unei instrucţiuni este:

NUME| OPERATIE OPERAND COMENTARIU

Cîmpul numelui, dacă este folosit, începe în poziţia 1 a codului sursă. Simbolul plasat în cîmpul numelui poate conţine oricîte caractere dar numai primele 5 caractere sînt folosite în tabela de simbolii a asamblorului. Numele trebuie să înceapă cu un caracter alfabetic şi nu poate conţine caractere speciale.

Cîmpul operaţiei conţine codul mnemonic al unei instrucţiuni maşină sau codul unei pseudooperaţii.

Cîmpul operandului conţine parametrii aferenţi operaţiei specificate în cîmpul operaţiei. Două argumente sînt separate printr-o virgulă.

Exemplu :

```
0015 ET1 MOV M,B ; COMENTARIU 0030 CALL ET1
0020 ; COMENTARIU 0035 ET2 ADI 8+6-4
0025 JMP ET2 0040 MOV A,B
```

Cîmpurile sînt separate prin unul sau mai multe blankuri.

Cîmpul comentariului este reprodus în listing fără a fi procesat. Liniile de comentariu încep cu caracterul ; în poziţia 1 a codului sursă. Comentariul

* Simboluri mnemonice ale limbajului de asamblare 8080.

instrucțiunilor individuale este precedat de asemenea de caracterul ; (vezi exemplele 0015 și 0020).

7.5.2. Nume simbolice. Pentru a asigna un nume simbolic unei instrucțiuni, se plasează un simbol în câmpul numelui. În caz contrar, utilizatorul plasează două sau mai multe blaturi în urma numărului liniei respective. Dacă un nume este atașat unei instrucțiuni, asamblorul asignează simbolului respectiv valoarea curentă a contorului de locații. Singura excepție la această regulă o constituie pseudoinstrucțiunea EQU. În acest caz, simbolul din câmpul numelui i se asignează valoarea conținută în câmpul operandului. Exemplu :

```
0030 ET EQU 170
```

Simbolii sînt definiți atunci cînd apar în câmpul numelui. Orice simbol definit poate fi utilizat ca argument simbolic în câmpul operandului (vezi exemplele 0015, 0025, 0030, 0035).

În afara simbolilor definiți de către utilizator, asamblorul recunoaște un set de simbolii rezervați, a căror valoare este predeterminată. Acești simbolii nu pot fi utilizați decît în câmpul operandului. Simbolii rezervați sînt (valoarea corespunzătoare este indicată în paranteză) :

A — acumulatorul (7)	H — registrul H (4)
B — registrul B (0)	L — registrul L (5)
C — registrul C (1)	M — memoria (locația indicată de conținutul registrelor H și L) (6)
D — registrul D (2)	P — Program Status Word (6)
E — registrul E (3)	S — Indicatorul stivei (6)

În câmpul operandului mai poate apărea și simbolul special \$ a cărui valoare se modifică pe măsură ce asamblarea programului sursă progresează. Simbolul \$ este întotdeauna echivalent cu valoarea contorului de locații după asamblarea instrucțiunii curente.

Exemple :

```
JMP $ ; implică salt la locația plasată după instrucțiunea
MOV A,B ; curentă, adică la instrucțiunea MOV A,B.
LDA $+5 ; implică încărcarea datei plasate în a cincea locație
DB 0 ; după instrucțiunea curentă. În cazul de față, această
DB 1 ; dată are valoarea 5.
DB 2
DB 3
DB 4
DB 5
```

7.5.3. Adresare simbolică relativă. O locație particulară poate fi referită folosind un simbol definit în program și un deplasament numeric. Exemplu :

```
JMP BEG
JPE BEG+4
CC SUB
CALL $+48
BEG MOV A,B
HLT
MVI C,'B' ; AICI 'B' ESTE CONSTANTA ASCII
INR B
```

Instrucțiunea **JPE BEG+4** se referă la instrucțiunea **INR B. BEG+4** înseamnă adresa **BEG** plus 4 octeți.

7.5.4. Constante. Asamblorul permite utilizatorului să folosească numere pozitive sau negative direct într-o instrucțiune. Numerele respective vor fi considerate drept constante zecimale. Orice număr fără semn este considerat pozitiv. Constante zecimale pot de asemenea fi definite folosind indicatorul **D** în urma valorii numerice respective.

Constante hexazecimale pot fi definite folosind indicatorul **H** în urma unei valori numerice (de exemplu **+10H, 3AH, 10H, 0F4H**). O constantă hexazecimală nu poate începe cu simbolii **A—F**. În acest caz constanta trebuie să fie precedată de cifra **0**.

Constante **ASCII** pot fi definite plasând un apostrof înaintea și în urma caracterelor **ASCII** respective, de exemplu **'C'** sau **'CC'**.

7.5.5. Expresii. O expresie reprezintă o secvență conținând unul sau mai mulți simbolii, constante sau alte expresii separate prin operatorii aritmeticii **+** sau **-**. Exemple :

PAM+3

ISAB-'A'+52

LOOP+32H-5

Expresiile sînt calculate folosind 16 biți, în aritmetica modulo **65536**. Orice valoare în afara acestui domeniu va rezulta într-o eroare de asamblare.

7.5.6. Pseudoinstrucțiuni

ORG

Formatul acestei pseudoinstrucțiuni este :

etichetă **ORG** expresie

unde eticheta este opțională dar în cazul în care este prezentă va fi echivalată cu valoarea expresiei specificate.

END

Pseudoinstrucțiunea **END** indică asamblorului sfîrșitul codului sursă. Ac astă pseudoinstrucțiune este opțională deoarece asamblorul detectează oricum sfîrșitul fișierului din care este preluat codul sursă.

EQU

Formatul acestei pseudoinstrucțiuni este

etichetă **EQU** expresie

unde eticheta reprezintă un simbol căruia i se va asigna valoarea expresiei specificate în cîmpul operandului.

DS

Formatul acestei pseudoinstrucțiuni este :

etichetă **DS** expresie

Pseudoinstrucțiunea DS va forța asamblorul să avanseze contorul de locații cu valoarea expresiei specificate.

DB

Formatul pseudoinstrucțiunii DB este :
etichetă DB expresie

Această pseudoinstrucțiune este folosită pentru a rezerva un octet de memorie. Valoarea expresiei din câmpul operandului va fi introdusă în octetul respectiv.

DW

Această pseudoinstrucțiune este folosită pentru a rezerva doi octeți de memorie. Formatul pseudoinstrucțiunii DW este similar cu cel al pseudoinstrucțiunii DB. Valoarea expresiei specificate va fi plasată în cei doi octeți rezervați.

7.5.7. **Erori de asamblare.** Următorii indicatori sînt folosiți de asamblor pentru a marca detectarea unei erori în cadrul unei instrucțiuni sursă :

- O Cod de operație eronat
- L Etichetă eronată,
- D Etichetă dublu definită,
- M Etichetă absentă,
- V Valoare eronată,
- U Simbol nedefinit,
- S Eroare de sintaxă,
- R Registru eronat,
- A Argument eronat,

7.5.8. **Salvarea programelor.** Sistemul de software MATE conține o comandă pentru salvarea programelor pe casetă magnetică. Sintaxa acestei comenzi este :

SAVE AAAA BBBB (CR)

Utilizînd această comandă, întreaga zonă de memorie cuprinsă între adresele AAAA și BBBB este înscrisă pe suport magnetic. În acest fel se pot salva atît fișiere sursă cît și fișiere obiect.

7.5.9. **Citirea programelor în memorie de pe caseta magnetică.** Această operație se execută utilizînd comanda LOAD cu sintaxa :

LOAD (CR)

7.6. Exemple de folosire a comenzilor MATE

```

ENTR 6800
AB 0C FF 02 4D 91/
DUMP 6800 6805
AB 0C FF 02 4D 91

```



```

0010 ; EXEMPLU DE PROGRAM INTRODUS ÎN FIȘIERUL CURENT
0020 MVI A,0
0030 MVI B,0
0040 MVI A,1
0050 MVI B,2
0060 MVI A,3
0070 MVI B,4
0080 HLT

```

```
DELT 0060 0070
```

```
LIST
```

```

0010 ; EXEMPLU DE PROGRAM INTRODUS ÎN FIȘIERUL CURENT
0020 MVI A,0
0030 MVI B,0
0040 MVI A,1
0050 MVI B,2
0080 HLT

```

```
ASSM 6A00
```

```

6A00          0010 ; EXEMPLU DE PROGRAM INTRODUS ÎN FIȘIERUL
                ; CURENT
6A00 3E00    0020 MVI A,0
6A02 0600    0030 MVI B,0
6A04 3E01    0040 MVI A,1
6A06 0602    0050 MVI B,2
6A08 70      0080 HLT

```

```
BREK 6A04
```

```
BREK 6A08
```

```
EXEC 6A00
```

```
6A04BREAK
```

```
DUMP 6000 600B
```

```
12 00 00 00 DB 02 B0 10 00 1A 04 1A
```

```
PROC
```

```
6A08BREAK
```

```
DUMP 6000 600B
```

```
12 01 00 02 B0 10 00 1A 08 1A
```

```
BREK 6A04
```

```
BREK 6A08
```

```
EXEC 6A00
```

```
6A04BREAK
```

```
DUMP 6000 600B
```

```
12 00 00 00 DB 02 B0 10 00 04 1A
```

```
PROC 6A06
```

```
6A08BREAK
```

```
DUMP 6000 600B
```

```
12 00 00 02 DB 02 B0 10 00 1A 08 1A
```

7.7. Repertoriul de instrucțiuni al microprocesorului 8080

Intrucît MATE dispune de un asamblor pentru repertoriul de instrucțiuni al microprocesorului 8080, în cele ce urmează se va face o scurtă prezentare a acestor instrucțiuni cu mnemonicile acceptate de asamblor și cu efectul lor asupra indicatorilor de condiții. De asemenea, vor fi prezentate în paralel și instrucțiunile corespunzătoare din repertoriul de instrucțiuni al microprocesorului Z80. Versiunea Z80 este indicată în paranteze unghiulare.

Literele A, B, C, D, E, H, L și SP reprezintă notațiile standard pentru registrele microprocesorului 8080. Simbolurile BC, DE și HL specifică perechile de registre corespunzătoare. Următoarele simboluri sînt folosite pentru parametrii generali:

r,r2 — registru de 8 biți din UCP,
 n — o constantă reprezentînd un octet
 nn — o constantă reprezentînd doi octeți

Indicatorii de condiții * au următoarele simboluri și semnificații:

C — transport,
 H — transport/imprumut între cele două tetrade ale rezultatului,
 N — adunare/scădere
 P/O — paritate/depășire
 S — semn
 Z — zero

Pentru mnemonicile Z80, indicatorii celulelor de memorie sau adresele de I/E sînt cuprinse între paranteze:

1. ACI n <ADC A,n>

Adună constanta n cu acumulatorul și cu indicatorul de transport. Rezultatul este plasat în acumulator.

Indicatori afectați: C, H, O, S, Z

Indicatori anulați: N

2. ADC M <ADC A,(HL)>

Adună octetul de memorie indicat de perechea HL cu acumulatorul și cu indicatorul de transport. Rezultatul este plasat în acumulator.

Indicatori afectați: C, H, O, S, Z

Indicatori anulați: N

3. ADC r <ADC A,r>

Adună registrul r cu acumulatorul și cu indicatorul de transport. Rezultatul este plasat în acumulator.

Indicatori afectați: C, H, O, S, Z

Indicatori anulați: N

4. ADD M <ADD A,(HL)>

Adună octetul din memorie indicat de perechea HL cu acumulatorul. Rezultatul rămîne în acumulator.

Indicatori afectați: C, H, O, S, Z

Indicatori anulați: N

5. ADD r <ADD A,r>

Adună registrul r cu acumulatorul. Rezultatul rămîne în acumulator.

Indicatori afectați: C, H, O, S, Z

Indicatori anulați: N

* De reținut că mașina fizică este procesorul Z80, în cazul microcalculatoarelor AMIG.

6. ADI n <ADD A,n>

Adună constanta n la acumulator. Rezultatul este păstrat în acumulatorul A.
Indicatori afectați : C, H, O, S, Z
Indicatori anulați : N

7. ANA N <AND (HL)>

Produsul logic între acumulator și octetul din memorie specificat de perechea HL.
Rezultatul este păstrat în acumulatorul A.

8. ANA r <AND r>

Produsul logic între acumulator și registrul r. Rezultatul este păstrat în acumulator.
Instrucțiunea ANA A se folosește pentru testarea indicatorilor de paritate, semn și rezultat zero, deoarece valoarea conținută în acumulatorul A nu se modifică.

Indicatori afectați : P, S, Z
Indicatori anulați : C, N
Indicatori poziționați în unu : H

9. ANI n <AND n>

Efectuează produsul logic între conținutul acumulatorului și octetul n prezent în instrucțiune. Rezultatul rămâne în acumulatorul A. Instrucțiunea poate fi utilizată pentru a anula selectiv biți din acumulatorul A. Astfel, instrucțiunea ANI FEH va anula bitul 0.

Indicatori afectați : P, S, Z
Indicatori anulați : C, N
Indicatori poziționați în unu : H

10. CALL nn <CALL nn>

Chemare necondiționată de subrutina de la adresa nn. Adresa instrucțiunii următoare este plasată în stivă, de unde este extrasă la revenirea din subrutină.

Indicatori afectați : nici unul

11. CC nn <CALL C, nn>
CM nn <CALL M, nn>
CNC nn <CALL NC, nn>
CNZ nn <CALL NZ, nn>
CP nn <CALL P, nn>
CPE nn <CALL PE, nn>
CPO nn <CALL PO, nn>
CZ nn <CALL Z, nn>

Chemări condiționate de subrutina de la adresa nn. Adresa instrucțiunii următoare este plasată în stivă.

Condițiile sint următoarele :

- C indicatorul de transport poziționat în unu,
M indicatorul de semn poziționat în unu,
NC indicatorul de transport poziționat în zero,
P indicatorul de semn poziționat în zero,
PE indicatorul de paritate poziționat în unu,
PO indicatorul de paritate poziționat în zero,
Z indicatorul de rezultat zero poziționat în unu

12. CMA <CPL>

Complementează acumulatorul (complementul față de unu). Biții egali cu zero au valoarea unu și invers.

Indicatorii afectați : H, N

13. CMC <CCF>

Complementează indicatorul de transport. Pentru a anula indicatorul de transport instrucțiunea CMC se va folosi după instrucțiunea STC.

Indicatori afectați : C

Indicatori poziționați în zero : N

14. CM M <CP (HL)>

Compară octetul din memorie, a cărui adresă este dată de perechea HL, cu acumulatorul. În cazul în care cele două valori sînt egale, se poziționează în unu indicatorul de rezultat zero. Acest indicator este poziționat în unu, dacă conținutul acumulatorului este mai mic decît operandul.

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în unu : N

15. CMP r <CPr>

Compară registrul r cu acumulatorul, care joacă rol de operand implicat. Indicatorul de rezultat zero este poziționat în unu, dacă conținutul acumulatorului este egal cu cel al registrului r. Dacă conținutul acumulatorului este mai mic decît cel al registrului r, indicatorul transportului este poziționat în unu.

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în unu : N

16. CPI n <CP n>

Compară constanta n conținută în instrucțiune cu acumulatorul. Indicatorul de rezultat zero este poziționat în unu, dacă constanta n este egală cu conținutul acumulatorului. Indicatorul de transport este poziționat în unu, dacă conținutul acumulatorului este mai mic decît constanta n.

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în unu : N

17. DAA <DAA>

Ajustare zecimală a acumulatorului.

Această instrucțiune este folosită după adunarea numerelor exprimate în codul binar-zecimal. Z80 efectuează corect această operație, atît pentru adunare, cît și pentru scădere. 8080 dă un rezultat incorect în cazul scăderii.

Indicatori afectați : C, H, O, S, Z

18. DAD B	<ADD HL,BC>
DAD D	<ADD HL,DE>
DAD H	<ADD HL,HL>
DAD SP	<ADD HL,SP>

Adună registrul dublu specificat, la registrul HL. Rezultatul este plasat în HL. Aceasta reprezintă adunarea cu precizie dublă. Indicatorul de transport este poziționat în unu dacă rezultatul depășește capacitatea de reprezentare pe 16 biți (dacă apare depășire). Instrucțiunea DAD H efectuează o deplasare spre stînga a conținutului perechii HL. Instrucțiunea DAD SP permite salvarea unui indicator de stivă.

LXI H,O	; încarcă perechea HL cu zero
DAD SP	; deplasează spre stînga cu un rang SP
SHLD SPVECHI	; plasează în stivă la adresa SPVECHI conținutul lui L și ; la adresa SPVECHI+1 conținutul lui H

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în zero : N

19. DCR M <DEC (HL)>

Decrementează octetul de memorie specificat de registrul HL.

Indicatori afectați : H, O, S, Z

Indicatori poziționați în unu : N

Indicatori neafectați : C

20. DCR r <DEC r>

Decrementează registrul r. În timp ce se execută instrucțiunea JNC buclă, nu mai trebuie decrementat registrul după ce a atins valoarea zero, deoarece indicatorul de transport nu este afectat de această operație.

Indicatori afectați : H, O, S, Z

Indicatori poziționați în unu : N

Indicatori neafectați : C

21. DCX B <DEC BC>
DCX D <DEC DE>
DCX H <DEC HL>
DCX SP <DEC SP>

Decrementează registrul dublu indicat. Într-un ciclu ce conține instrucțiunea JNZ ciclu, nu trebuie să se încerce decrementarea la zero a registrului dublu indicat, deoarece indicatorii nu sînt afectați. Se poate deplasa un octet din registrul dublu în acumulatorul A, pentru a se efectua operația logică OR cu celălalt octet :

REPETA :

MOV A,C
ORA B
JNZ REPETA

Indicatori afectați : nici unul

22. DI <DI>

Dezactivează cererea de întrerupere.

23. EI <EI>

Activează cererea de întrerupere

24. HLT <HALT>

Suspendă funcționarea UCP pînă la apariția unui semnal RESET sau de întrerupere

25. IN n <IN A,(n)>

Transferă în acumulatorul A octetul de la portul cu adresa n.

Indicatori afectați : nici unul

26. INR M <INC (HL)>

Incrementează octetul din memorie cu adresa specificată în perechea HL.

Indicatori afectați : H, O, S, Z

Indicatori poziționați în unu : N

Indicatori neafectați : C

27. INR r] <INC r>

Incrementează registrul r. În timp ce se execută un ciclu ce conține instrucțiunea JNC ciclu, nu trebuie incrementat un registru peste valoarea zero, deoarece indicatorul de transport nu va fi afectat.

Indicatori afectați : H, O, S, Z

Indicatori poziționați în unu : N

Indicatori neafectați : C

28. INX B <INC BC>
INX D <INC DE>
INX H <INC HL>
INX SP <INC SP>

Incrementează registrul dublu specificat.

Indicatori afectați : nici unul

29. JMP nn <JP nn>

Transfer necondiționat al comenzii la adresa nn.

Indicatori afectați : nici unul

30. JG nn <JP C,nn>
 JM nn <JP M,nn>
 JNC nn <JP NC,nn>
 JNZ nn <JP NZ,nn>
 JP nn <JP P,nn>
 JPE nn <JP PE,nn>
 JPO nn <JP PO,nn>
 JZ nn <JP Z,nn>

Transfer condiționat al comenzii la adresa nn unde condițiile sînt următoarele :

C indicatorul de transport poziționat în unu,

M indicatorul de semn poziționat în unu,

NC indicatorul de semn poziționat în zero,

NZ indicatorul de rezultat zero poziționat în zero,

P indicatorul de semn poziționat în zero,

PE indicatorul de paritate poziționat în unu,

PO indicatorul de paritate este poziționat în zero,

Z indicatorul de rezultat zero este poziționat în unu.

31. LDA nn <LD A,(nn)>

Încarcă acumulatorul A cu octetul din memorie de la adresa nn

32. LDAX B <LD A,(BC)>

LDAX D <LD A,(DE)>

Încarcă acumulatorul A cu octetul din memorie, de la adresa specificată de perechea de registre BC sau DE.

33. LHLD nn <LD HL,(nn)>

Încarcă registrul L cu octetul din memorie de la adresa nn, iar registrul H cu octetul de la adresa nn+1.

34. LXI B,nn <LD BC,nn>

LXI D,nn <LD DE,nn>

LXI H,nn <LD HL,nn>

LXI SP,nn <LD SP,nn>

Încarcă perechea de registre specificate cu constanta de 16 biți nn.

35. MOV M,r] <LD (HL),r>

Stochează în memorie octetul din registrul r, la locația specificată ca adresă în perechea de registre HL.

36. MOV r,M <LD r,(HL)>

Încarcă registrul r cu octetul din memorie, de la adresa specificată de perechea HL.

37. MOV r,r2 <LD r,r2>

Transferă conținutul registrului r2 în registrul r.

38. MVI M,n <LD (HL),n>

Transferă operandul imediat n, în memorie la adresa specificată de perechea HL.

39. MVI r,n <LD r,n>

Încarcă registrul r cu operandul imediat n.

40 NOP <NOP>

UCP nu efectuează nici o operație
Indicatori afectați : nici unul

41. ORA M <OR (HL)>

Efectuează suma logică între conținutul acumulatorului A și conținutul celulei de memorie specificată de perechea HL. Rezultatul este plasat în acumulator.

Indicatori afectați : P, S, Z

Indicatori poziționați în zero : C, H, N

42. ORA r <OE r>

Efectuează suma logică între conținutul acumulatorului A și conținutul registrului r. Rezultatul este păstrat în acumulator. Deoarece instrucțiunea ORA A nu modifică conținutul lui A, ea poate fi folosită pentru a testa indicatorii de paritate semn și rezultat zero.

Indicatori afectați : P, S, Z

Indicatori poziționați în zero : C, H, N

43. ORI n <OR n>

Efectuează suma logică între conținutul acumulatorului și operandul imediat n. Rezultatul este plasat în acumulator. Instrucțiunea poate fi folosită pentru poziționarea în unu a unor biți anumiți din acumulator. De exemplu : ORI,40H va poziționa bitul 6 al acumulatorului în unu.

Indicatori afectați : P, S, Z

Indicatori poziționați în zero : C, H, N

44. OUT n <OUT (n),A>

Transferă octetul din acumulatorul A la portul de ieșire cu adresa n.

Indicatori afectați : nici unul

45. PCHL <JP (HL)>

Forțează conținutul perechii HL în conținutul programului PC. Astfel, se asigură un transfer al comenzii la adresa specificată de perechea HL.

Indicatori afectați : nici unul

46. POP B <POP BC>

POP D <POP DE>

POP H <POP HL>

Transferă primii doi octeți din stivă, în perechea de registre specificată : BC, DE, HL. Octetul specificat de SP este transferat în registrul inferior (C, E, L), după care are loc incrementarea indicatorului stivei, SP.

În continuare octetul specificat de SP este transferat în registrul superior (B, D, H), după care SP este din nou incrementat.

Indicatori afectați : nici unul.

47. POP [PSW] <POP [AF]>

Transferă primul octet din stivă, indicat de SP, în registrul indicatorilor de condiții și incrementează indicatorul SP. Transferă în continuare primul octet din stivă în acumulatorul A, apoi incrementează indicatorul SP.

Indicatori afectați : nici unul

48. PUSH B <PUSH BC>

PUSH D <PUSH DE>

[PUSH H] <PUSH [HL]>

Plasează în stivă perechile de registre BC, DE, HL. Indicatorul SP este decrementat și la adresa indicată de el se stochează registrul de rang superior (B, D, H). În continuare SP este din nou decrementat memorându-se, la adresa astfel obținută, registrul de rang inferior din perechea specificată (C, E, L).

Indicatori afectați : nici unul.

49. PUSH PSW <PUSH AF>

Stochează acumulatorul și registrul indicatorilor de condiții în memorie. Se decrementează indicatorul SP și la adresa specificată de el se stochează acumulatorul A. În continuare se decrementează din nou SP, memorându-se, la adresa astfel specificată, registrul indicatorilor de condiții.

Indicatori afectați : nici unul

50. RAL <RLA>

Instrucțiunea rotește spre stînga cu un bit conținutul acumulatorului, prin intercalarea bistabilului indicator de transport între bitul 7 și bitul 0. Bitul 7 se transferă în bistabilul indicator de transport, în timp ce conținutul acestuia este transferat în bitul 0 al acumulatorului.

Indicatori afectați : C

Indicatori poziționați în zero : H, N

51. RAR <RRA>

Instrucțiunea rotește spre dreapta conținutul acumulatorului A, prin intercalarea bistabilului de transport între bitul 0 și bitul 7. Bitul 0 se transferă în bistabilul indicator de transport, în timp ce conținutul anterior al acestuia se transferă în bitul 7.

Indicatori afectați : C

Indicatori poziționați în zero : H, N

52. RET <RET>

Revenire din subrutină. Primii doi octeți din virful stivei sînt plasați în contorul programului PC. Indicatorul SP este incrementat de două ori.

53. RC	<RET C>
RM	<RET M>
RNC	<RET NC>
RNZ	<RET NZ>
RP	<RET P>
RPE	<RET PE>
RPO	<RET PO>
RZ	<RET Z>

Reveniri condiționate din subrutine. Dacă condiția specificată este îndeplinită, conținutul primelor două celule din stivă este transferat în contorul programului. Contorul programului este incrementat de două ori.

Condițiile testate sînt următoarele :

- C indicatorul de transport poziționat în unu,
- M indicatorul de semn poziționat în unu,
- NC indicatorul de transport poziționat în zero,
- NZ indicatorul de rezultat egal cu zero poziționat în zero,
- P indicatorul de semn poziționat în unu,
- PE indicatorul de paritate poziționat în unu,
- PO indicatorul de paritate poziționat în zero,
- Z indicatorul de rezultat zero poziționat în unu,

54. RLC <RLCA>

Instrucțiunea rotește conținutul acumulatorului cu un bit spre stînga. Bitul 7 se transferă, atât în indicatorul de transport, cit și în bitul 0 al acumulatorului.

Indicatori afectați : C

Indicatori poziționați în zero : H, N

55. RRC <RRCA>

Instrucțiunea rotește conținutul acumulatorului cu un bit spre dreapta. Bitul 0 se transferă atât în indicatorul de transport, cit și în bitul 7 al acumulatorului.

Indicatori afectați : C

Indicatori poziționați în zero : H, N

56. RST 0	<RST 00H>
RST 1	<RST 08H>
RST 2	<RST 10H>
RST 3	<RST 18H>
RST 4	<RST 20H>
RST 5	<RST 28H>
RST 6	<RST 30H>
RST 7	<RST 38H>

Instrucțiunile de restart generează chemări de subrutine la adresele 00H, ..., 38H. De exemplu RST 6 va chema adresa 30 hexa.

57. SBB M <SBC A,(HL)>

Din acumulator se scad : octetul din memorie specificat de perechea HL în conținutul indicatorului de transport. Rezultatul este plasat în acumulator.

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în unu : N

58. SBB r <SBC A,r>

Din acumulator se scad : conținutul registrului r și conținutul indicatorului de transport. Rezultatul este plasat în acumulator.

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în unu : N

59. SBI n <SBC A,n>

Din conținutul acumulatorului se scad : operandul imediat n și conținutul indicatorului de transport. Rezultatul rămâne în acumulator.

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în unu : N

60. SHLD nn <LD (nn),HL>

Stochează registrul L în memorie la adresa nn, iar registrul H la adresa nn+1.

61. SPHL <LD SP,HL>

Încarcă indicatorul SP cu conținutul registrului HL. Această instrucțiune se poate folosi pentru a extrage din memorie indicatorul de stivă salvat anterior.

LHLD n
SPHL

62. STA nn] <LD fnn),A>

Stochează acumulatorul în locația de memorie cu adresa nn

63. STAX B <LD (BC),A>

STAX D <LD (DE),A>

Stochează conținutul acumulatorului în celula de memorie a cărei adresă este specificată de perechea de registre BC sau DE.

64. STC <SCF>

Poziționează în unu indicatorul în transport.

Întrucât nu există instrucțiune echivalentă de poziționare în zero a indicatorului, aceasta se poate realiza, fie cu ajutorul instrucțiunii XRA A, fie cu ajutorul perechii de instrucțiuni STC și CMC.

Indicatori poziționați în unu : C

Indicatori poziționați în zero : H, N

65. SUB M <SUB (HL)>

Scade din acumulator octetul din memorie specificat de perechea HL. Rezultatul este plasat în acumulator.

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în unu : N

66. SUB r <SUB r>

Scade din acumulatorul A conținutul registrului r. Rezultatul este plasat în acumulator.

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în unu : N

67. SUI n <SUB n>

Scade operandul imediat n din acumulator. Rezultatul este plasat în acumulator.

Indicatori afectați : C, H, O, S, Z

Indicatori poziționați în unu : N

68. XCHG <EX DE,HL>

Interschimb de conținuturi între perechile DE și HL.

Indicatori afectați : nici unul

69. XRA M <XOR (HL)>

Execută suma modulo doi (SAU-EXCLUSIV) între conținutul acumulatorului și octetul din memorie specificat de perechea HL. Rezultatul este plasat în acumulator.

Indicatori afectați : P, S, Z

Indicatori poziționați în zero : C, H, N

70. XRA r <XOR r>

Execută suma modulo doi între conținutul acumulatorului și conținutul registrului r. Rezultatul este plasat în acumulator.

Instrucțiunea XRA A permite anularea conținutului indicatorilor. Ea se folosește și pentru anularea indicatorului de transport.

Indicatori afectați : P, S, Z

Indicatori poziționați în zero : C, H, Z

71. XRI n <XOR n>

Efectuează suma modulo doi între conținutul acumulatorului și operandul imediat n. Rezultatul este plasat în acumulator.

Indicatori afectați : P, S, Z

Indicatori poziționați în zero : C, H, N

72. XTHL <EX (SP),HL>

Interschimb între conținutul octetului de memorie indicat de SP și conținutul registrului L. De asemenea, interschimb între conținutul octetului de memorie indicat de SP+1 și conținutul registrului H.

Indicatori afectați : nici unul.

LOC OBJ	LINE	SOURCE STATEMENT	LOC OBJ	LINE	SOURCE STATEMENT
0000	1	*****	0067	221CA0	55 ; TRATARE INTRERUPERE NEMASCABILA
0000	2	* A M I C *	006A	F5	56 ; RESTART ADRESA 0066H
0000	3	*****	006B	E1	57 ; LA 66H SE GASESTE OCTET SUPERIOR DE ADRESA
0000	4	MONITOR PENTRU CALCULATORUL PERSONAL AMIC	006C	E1	58 ; PENTRU JMP ET43 00H+NOP
0000	5		006D	F5	59 ; SHLD STVUT+6 ;H,L
0000	6		006E	F1	60 ; PUSH PSW ;PSW
0000	7		006F	E1	61 ; POP H ;PSW
0000	8		0070	D5	62 ; SHLD STVUT
0000	9		0071	E1	63 ; POP H
0000	10		0072	D5	64 ; SHLD STVUT+10 ;PC
0000	11		0073	F5	65 ; LXI H,0
0000	12		0074	39	66 ; DAD SP
0000	13		0075	E1	67 ; SHLD STVUT+8 ;SP
0000	14		0076	D5	68 ; LXI SP,STVUT+6 ;D,E
0000	15		0077	E1	69 ; POP H ;B,C
0000	16		0078	C5	70 ; DCR B ;INVERSEAZA ORDINEA
0000	17		0079	F5	71 ; MOV SP,MONSP
0000	18		007A	F5	72 ; INVHL BUCLA
0000	19		007B	F5	73 ; CALL C,A
0000	20		007C	F5	74 ET43: MOV M,A ;INTRODUCERE IN BUFFER COD TASTA APASATA
0000	21		007D	F5	75 ; MOV M,A ;AFISARE
0000	22		007E	F5	76 ; CALL AFIS
0000	23		007F	F5	77 ; INX H
0000	24		0080	05	78 ; DCR B
0000	25		0081	F5	79 ; MOV SP,STVUT+6 ;ERR
0000	26		0082	F5	80 ; DCR C ;CDA PREA LUNGA
0000	27		0083	F5	81 ; DCR D ;ESTE RETURN
0000	28		0084	F5	82 ; DCR E ;ET42
0000	29		0085	F5	83 ; DCR H ;C,0AH ;TIPARESTE LF
0000	30		0086	F5	84 ; DCR D ;D,CDA
0000	31		0087	F5	85 ; DCR C ;D
0000	32		0088	F5	86 ; DCR B ;D'
0000	33		0089	F5	87 ; DCR A ;D'ISP
0000	34		008A	F5	88 ; DCR H ;S'
0000	35		008B	F5	89 ; DCR C ;SUBST
0000	36		008C	F5	90 ; DCR D ;X'
0000	37		008D	F5	91 ; DCR E ;X'
0000	38		008E	F5	92 ; DCR H ;EXAM
0000	39		008F	F5	93 ; DCR D ;C'
0000	40		0090	F5	94 ; DCR C ;CHNG
0000	41		0091	F5	95 ; DCR B ;M'
0000	42		0092	F5	96 ; DCR A ;MOVE
0000	43		0093	F5	97 ; DCR H ;F'
0000	44		0094	F5	98 ; DCR D ;FILL
0000	45		0095	F5	99 ; DCR C ;G'
0000	46		0096	F5	100 ; DCR B ;GO
0000	47		0097	F5	101 ; DCR A ;L'
0000	48		0098	F5	102 ; DCR H ;LOAD
0000	49		0099	F5	103 ; DCR D ;K'
0000	50		009A	F5	104 ; DCR C ;STORE
0000	51		009B	F5	105 ; DCR B ;B'
0000	52		009C	F5	106 ; DCR A ;BASIC
0000	53		009D	F5	107 ; DCR H ;C,7'
0000	54		009E	F5	108 ; DCR D ;AFIS
0000	55		009F	F5	109 ; DCR C ;COD CDA ANULATA
0000	56		00A0	F5	109 ; DCR B ;BUCLA

 * A M I C *

 MONITOR PENTRU CALCULATORUL PERSONAL AMIC
 ;CUVINT CDA 8255
 ;INSCRIERE IN 8255
 ;REFACE H,L
 ;REFACE PSW
 ;SALVEAZA PC
 ;REFACE OCTET PROGRAM
 ;INVERSEAZA ORDINEA
 ;INITIALIZARE SP MONITOR
 ;STERGE ECRAN
 ;H,L=ADRESA INCEPUT TEXT
 ;B=CONTOR CARACTERE
 ;MESAJ MONITOR
 ;PROMPTER
 ;NR MAX CARACTERE
 ;CITIRE TASTA
 ;ESTE TASTA DELETE?
 ;AFISARE

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
00DB	3A01A0	110	SECVENTA DELETE	0129	114000	165	LXI D,64
00DE	FE01	111	ETA1: LDA COL	012C	19	166	DAD D
00EO	CASCO0	112	CPI I	012D	79	167	MOV A,C
00E3	3D	113	JZ ET42	012E	110800	168	LXI D,8
00E4	3201A0	114	DCR A	0131	E607	169	ANI 7
00E7	23	115	STA 80L	0133	CA3B01	170	E111
00EB	04	116	DCX H	0136	19	171	DAD D
00E9	C35C00	117	INR B	0137	3D	172	DCR A
		118	JMP ET42	0138	C23601	173	JNZ E15
		119		013B	78	174	A,B
		120		013C	0F	175	MOV A,B
		121	SUBROUTINA KEY	0140	23	176	JC ET6
		122		0141	DA4401	177	INX H
		123	SCANEAZA TASTATURA	0141	C33C01	178	JMP ET7
		124	CO-C2 =NR LINIE SCANARE	0144	D821	179	IN PORTB
		125	AI = 1, LINIA I DE RETURN	0146	E640	180	ANI 40H
		126	Z=1/0 DA/NU TASTA APASATA	0148	7E	181	MOV A,M
		127		0149	C24E01	182	JNZ ET8
00EC	3A03A0	128	KEY: LDA EINV	014C	E61F	183	ANI 1FH
00EF	4F8	129	ANI 0F8H	014E	67	184	MOV H,A
00F1	4F	130	MOV C,A	014F	CDEC00	185	CALL KEY
00F2	D322	131	PORTC	0152	C24F01	186	E19
00F4	DB20	132	PORTA	0155	7C	187	MOV A,H
00F6	EEFF	133	XRI OFFH	0156	CD5001	188	BIP
00F8	CO	134	RNZ	0159	C1	189	POP B
00F9	0C	135	INR C	015A	D1	190	POP D
00FA	79	136	MOV A,C	015B	E1	191	POP M
00FB	E607	137	ANI 7	015C	C9	192	RET
00FD	C2F200	138	JNZ KEY1				
0100	C9	139	RET				
		140					
		141					
		142	SUBROUTINA KEYIN				
		143					
		144	CITESTE UN CARACTER DE LA TASTATURA	015D	F5	199	BIP: PUSH B
		145	A=COD ASCII AL TASTEI APASATE	015E	C5	200	PUSH B
		146		015F	0E10	201	MVI C,10H
0101	E5	147	KEYIN: PUSH H	0161	3A03A0	202	LDA EINV
0102	D5	148	PUSH D	0164	F5	203	PUSH PSH
0103	C5	149	PUSH B	0165	47	204	MOV B,A
0104	0600	150	MVI B,0	0166	D322	205	OUT PORTC
0106	CD0005	151	CALL BITW	0168	CD0005	206	CALL BITW
0109	3E5F	152	MVI A,5FH	016B	F1	207	POP PSH
010B	CD7A01	153	CALL WRITE	016C	EE08	208	XRI 8
010E	3E20	154	MVI A,20H	016E	D322	209	OUT PORTC
0110	CD7A01	155	CALL WRITE	0170	CD0005	210	CALL BITW
0113	0400	156	MVI B,0	0173	00	211	DCR C
0115	CD0005	157	CALL BITW	0174	C26101	212	JNZ BIP1
0118	CD0E00	158	CALL KEY	0177	C1	213	ROP B
011B	CA0401	159	JZ ET3	0178	F1	214	POP PSH
011E	47	160	ETA2: M8V B,A	0179	C9	215	RET
011F	210005	161	LXI H,TAB				
0122	5B21	162	IN PORTC				
0123	E620	163	ANI 20H				
0126	C22D01	164	JNZ ET10				

!TEST PB6=CTRL
 !CITESIE COD IN A
 !REVENIRE DACA NON CTRL
 !DACA CTRL,ANULEAZA BITII 6,5
 !SALVARE COD
 !ASTEAPTA ELIBERARE TASTA APASATA

!MARTOR SONDR APASARE TASTA

!NR. IMPULSURI IN DIFUZOR

!RELUARE DACA NU TASTA APASATA
 !SALVARE
 !ABRESA TABELEI DE CODURI
 !TEST BIT 5=SHIFT

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
017A	E5	220	:AFISEAZA UN CARACTER LA CONSOLA	01D5	3E06	275	MVI A,6
017B	D5	221	:NU SE UTILIZEAZA POZITIE CURSOR	01D7	CD0502	276	CALL WR26
017C	C5	222	:AF CARACTERE DE AFISAT	01DA	9A04A0	277	LDA VINV
017D	F561	223	:DIFERENTA DE AFISAT	01DD	2F	278	CMVA
017E	DAE901	224	:61H=CAR SEMIDRAFIC<=70H	01DE	77	279	MOV M,A
017F	3D	225	:	01DF	C1	280	POP B
0180	3D	226	WRITE: PUSH H	01E0	D1	281	POP D
0181	3D	227	PUSH D	01E1	E1	282	POP H
0182	C5	228	PUSH B	01E2	C9	283	RET
0183	F561	229	CFI	01E3	EB	284	:CALCULEAZA ADRESA PENTRU ECRAN
0184	2129A0	230	JC WR27	01E4	D5	285	:D,E=ADRESA IN GENERATORUL DE CARACTERE
0185	3D	231	JR A	01E5	210140	286	D
0186	3D	232	MOV C,A	01E6	210140	287	H,4001H :ADR. PRIMULUI CARACTER
0187	1E02	233	H,TAMP	01E7	110010	288	D,100H :INCREMENT RIND CARACTER
0188	16FF	234	LXI MVI E,2	01E8	3A00A0	289	RND :IN A SE AFLA NR. RIND
0189	E01	235	WR43: I	01E9	B7	290	ORA
0190	16FF	236	D,OFFH	01EA	CAF701	291	JZ WR22
0191	CA9201	237	WR50	01EB	19	292	DAD D
0192	79	238	D,OF0H	01EC	3D	293	DCR A
0193	E602	239	MOV A,C	01ED	C2F201	294	JNZ WR23
0194	CA9E01	240	ANI 2	01EE	3A01A0	295	WR22: LDA COL
0195	3D	241	JZ WR41	01EF	B7	296	ORA A
0196	7A	242	MOV M,A	01F0	CA0302	297	JZ WR24
0197	5F	243	CMVA	01F1	E 23	298	WR25: INX H
0198	0604	244	MOV MVI B,4	01F2	3D	299	DCR A
0199	72	245	WR41: M,D	01F3	CA0302	300	JNZ WR25
019A	23	246	WR44: H	01F4	C2FE01	301	POP P
019B	05	247	INX B	01F5	D1	302	RET
019C	05	248	DCR B	01F6	C9	303	HR26: LXI PUSH
019D	05	249	MOV MVI B,4	01F7	1A	304	LDAX D
019E	23	250	WR44: A,C	01F8	2F	305	CMVA
019F	05	251	RRC	01F9	77	306	MOV M,A
01A0	C29D01	252	RRC	01FA	3D	307	LDA VINV
01A1	CF	253	MOV MVI B,4	01FB	AE	308	M
01A2	4F	254	WR44: A,C	01FC	77	309	MOV M,A
01A3	4F	255	RRC	01FD	09	310	DAD B
01A4	1D	256	DCR E	01FE	13	311	INX D
01A5	28901	257	JNZ WR43	01FF	F1	312	POP P
01A6	2129A0	258	LXI H,TAMP	0200	3D	313	DCR A
01A7	3E08	259	CALL WR30	0201	C9	314	JNZ WR26
01A8	3E08	260	MVI A,8	0202	05	315	RET
01A9	CD0502	261	CALL WR25	0203	1A	316	:
01AA	215304	262	JMP WR22	0204	2F	317	:
01AB	110600	263	LXI H,CARGN	0205	77	318	:SUBROUTINA SCROL
01AC	D620	264	D,6	0206	F5	319	:REALIZEAZA DEFILARE ECRAN CU UN RIND DE CARACTERE
01AD	CAE901	265	SUI 20H	0207	1A	320	:
01AE	19	266	JZ WR20	0208	77	321	:
01AF	3D	267	DAD D	0209	05	322	:
01B0	C2E401	268	DCR A	020A	2F	323	:SCROL: LXI H,4000H
01B1	CBE301	269	JNZ WR21	020B	77	324	D,4100H :ADRESA DESTINATIE
01B2	2F	270	CALL WR30	020C	1A	325	D :ADRESA SURSA
01B3	77	271	LXI B,20H	020D	1A	326	MOV M,A
01B4	09	272	LDA VINV	020E	13	327	INX D
		273	CMVA	020F	23	328	INX H
		274	DAD B	0210	7A	329	MOV A,D

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
0202	BC	440	CMF	0037	23	495	ET61: INX H
0203	CA4F00	441	JZ	0338	C3EC02	496	JMP ET64
0206	23	442	ET60: INX H			497	
0207	7D	443	MOV A,L			498	
0208	E607	444	ANI 7			499	-----
020A	C2C502	445	INZ ET52			500	SUBROUTINA CONVI
020D	CD7904	446	CALL CRLF			501	EGALIZEAZA NIVELUL DE ADINCIME
02E0	C3BD02	447	JMP ET51			502	CONVA SI CONVB
		448				503	
		449				504	CONVI: CALL CONVB
		450	SUBROUTINA SUBST			505	RET
		451				506	
		452	PERMITE AFISAREA SI MODIFICAREA			507	
		453	UNITI SIR DE OCTETI DIN MEMORIE			508	SUBROUTINA EXAM
		454				509	
		455	SUBST: CALL CONVA			510	AFISEAZA REGISTRELE INTERNE UTILIZATOR
02E3	CD2804	456	LDAX D			511	
02E6	1A	457	RETUR			512	EXAM: LDAX D
02E7	FE0D	458	CPI			513	CPI
02E9	C2D300	459	JNZ			514	JNZ
02EC	7E	460	MOV A,M			515	LXI H,TEXT2
02ED	CD5404	461	CALL			516	MVI B,27
02F0	0E2D	462	MVI C,...			517	CALL TEXT
02F2	CD3602	463	CALL			518	CALL CRLF
02F5	1105A0	464	LXI D,CDA			519	MVI E,6
02F8	3E30	465	MVI A,30H			520	H,STVUT
02FA	12	466	STAX D			521	MOV A,M
02FB	13	467	INX D			522	CALL BINASC
02FC	CD0101	468	CALL			523	INX H
02FF	FE0D	469	CPI			524	MOV A,M
0301	CA4F00	470	JZ			525	CALL BINASC
0304	FE20	471	CPI			526	INX H
0306	6A3703	472	ET61			527	DCR E
0309	4F	473	MOV C,A			528	BUCLA
030A	CD3602	474	CALL			529	CALL AF20H
030D	12	475	STAX			530	JMP ET71
030E	CD0101	476	CALL			531	
0311	F5	477	PUSH PSW			532	-----
0312	4F	478	MOV C,A			533	SUBROUTINA CHNG
0313	CD3602	479	CALL			534	
0318	FE20	480	CPI			535	PERMITE AFISAREA SI MODIFICAREA
031B	CA2503	481	JZ			536	REGISTRELOR UTILIZATOR
031D	FE0D	482	RETUR			537	
0320	F1	483	JZ			538	CHNG: LDAX D
0321	13	484	POP PSW			539	CPI
0322	C30D03	485	INX D			540	RETUR
0325	1B	486	JMP			541	ERR
0326	2223A0	487	DCX D			542	LXI H,STVUT
0329	CD3E03	488	CALL			543	JMP ET64
032C	7D	489	SHLD			544	
032D	2A23A0	490	CALL			545	SUBROUTINA MOVE
0330	77	491	MOV A,L			546	
0331	F1	492	LHLD			547	MUTA O ZONA DE MEMORIE IN ALTA ZONA
0332	FE0D	493	MOV M,A			548	
0334	CA4F00	494	POP PSW			549	CONV2: CALL CONVA

LOC	OBJ	LINE	SOURCE STATEMENT	ADR.	INCEPUT ZONA SURSA	LOC	OBJ	LINE	SOURCE STATEMENT
0378	2223A0	550	SHLD	ADR1		03DA	7C	605	MOV A,H
037B	1A	551	LDAX	D		03DB	BA	606	D
037C	FE2C	552	CPI	2CH		03DC	C2D303	607	JNZ FIL1
037E	C28A03	553	JNZ	ER1		03DF	70	608	MOV M,B
0381	13	554	INX	B		03E0	C34F00	609	JMP BUCLA
0382	C02804	555	CALL	CONVA				610	
0385	2225A0	556	SHLD	ADR2	:ADR. SF IRS11 ZONA SURSA			611	
0388	1A	557	LDAX	D				612	SUBROUTINA GO
0389	C9	558	RET					613	
038A	E1	559	POP	H				614	LANSEAZA IN EXECUTIE UN PROGRAM UTILIZATOR
038B	C3D300	560	JMP	ERR				615	
038E	DD7503	561	CALL	CONV2		03E3	CD2804	616	GO: CONVA
0391	FE2C	562	CPI	2CH		03E6	2223A0	617	SHLD ADR1
0393	C2D300	563	JNZ	ERR		03E9	1A	618	LDAX D
0396	13	564	INX	D		03EA	FE0D	619	CPI RETUR
0397	C02804	565	CALL	CONVA		03EC	CA0204	620	JZ ET81
039A	2227A0	566	SHLD	ADR8	:ADR. INCEPUT ZONA DESTINATIE	03EF	FE2C	621	CPI 2CH
039D	1A	567	LDAX	D		03F1	C2D300	622	JNZ ERR
039E	FE0D	568	CPI	RETUR		03F4	13	623	INX D
08A0	C2D300	569	JNZ	ERR		03F5	CD2804	624	CALL CONVA
03A3	2A27A0	570	: TRANSFER ZONA DE MEMORIE			03F8	1A	625	LDAX D
03A6	EB	572	XCHG	ADR3		03F9	FE0D	626	CPI RETUR
03A7	2A23A0	573	LHLD	ADR1	:D,E=ADRESA ZONA DESTINATIE	03FR	C2D300	627	JNZ ERR
03AA	7E	574	MOV	A,M	:H,L=ADRESA ZONA SURSA	03FE	7E	628	MOV A,M
03AB	12	575	STAX	D	:CITESTE DIN ZONA SURSA	03FF	36CF	629	MVI M,OCFH ;RST1
03AC	3A25A0	576	LDI	ADR2	:SORIE IN ZONA DESTINATIE	0401	F5	630	PUSH PSW
03AF	6D	577	CMP	L		0402	CD1E04	631	CALL INVHL ;SALVARE IN STIVA MONITOR OCTET PROGRAM
03B0	C28A03	578	JNZ	ET75		0405	311EAO	632	LXI SP,STUVT ;INVERSARE OCTETI STIVA AFISARE
03B3	3A26A0	579	LDA	ADR2+1		0408	F1	633	POP B
03B6	BC	580	CMP	H		0409	C1	634	POP POP
03B7	CA4F00	581	JZ	BUCLA		040A	D1	635	POP D
03BA	23	582	INX	H		040B	E1	636	POP POP
03B8	13	583	INX	D		040C	E1	637	POP H
03BC	C3AA03	584	JMP	ET77		040D	F9	638	SPHL SPW ;REFACERE REGISTRU UTILIZATOR
03C2	FE2C	592	CPI	2CH		040E	2A23A0	639	LHLD ADR1
03C4	C2D300	593	INX	ERR		0411	E5	640	STUVT+6 ;REFACERE H,L PROGRAM
03C8	03B803	595	CALL	CONV3		0412	2A1CA0	641	LHLD ;LANSARE
03C8	45	596	MOV	B,L		0415	C9	642	RET
03CC	2A25A0	597	LHLD	ADR2				643	
03CF	EB	598	XCHG	ADR1				644	
03D0	2A23A0	599	LHLD	M,B				645	SUBROUTINA INVHL
03D3	70	600	MOV	H				646	
03D4	23	601	INX	A,L				647	INVERSEAZA ORDINEA OCTET LOW SI OCTET HIGH IN STUVT
03D5	7D	602	MOV	E,L				648	
03D6	BB	603	CMP	E				649	INVHL: MVI C,6
03D7	C2D303	604	JNZ	FIL1				650	LXI H,STUVT
								651	MOV A,M
								652	INX H
								653	MOV B,M
								654	MOV M,A
								655	DCX H
								656	MOV M,B
								657	INX H
								658	INX H
								659	DOR C

LOC OBJ	LINE	SOURCE STATEMENT
0424 C21B04	660	JNZ ET72
0427 C9	661	RET
	662	-----
	663	-----
	664	! SUBRUTINA CURVA
	665	-----
	666	! CONVA=CONVERSIE ADRESA
	667	! CONVERSIIE 4 CARACTERE ADRESA
	668	! ADRESA REZULTA IN H,L
	669	! MODIFICA: A, AD, E=D, E+4
	670	-----
0426 210000	671	! CONVA: LXI H,0 ! INITIALIZARE H,L
042B CD3304	672	CALL CONV B ! REZULTAT IN L
042E 65	673	MOV H,L
042F CD3304	674	CALL CONV B
0432 C9	675	RET
	676	-----
	677	-----
	678	! SUBRUTINA CONV B
	679	-----
	680	! CONVB=CONVERSIE BYTE
	681	! 2 CARACTERE ASCII DE LA ADRESA DIN D,E
	682	! L-BYTE
	683	! MODIFICA: A, D,E=D,E+2
	684	-----
0433 1A	685	CONVB: LDAX D
0434 CD7104	686	CALL ASCB IN
0437 FE10	687	CPI 10H
0439 D24F04	688	JNC CON1 ! TEST DACA CIFRA HEXA
043C 07	689	RLC
043D 07	690	RLC
043E 07	691	RLC
043F 07	692	RLC
0440 6F	693	MOV L,A
0441 13	694	INX D
0442 1A	695	LDAX D
0443 CD7104	696	CALL ASCB IN
0446 FE10	697	CPI 10H
0448 D24F04	698	JNC CON1
044B B5	699	ORA L
044C 6F	700	MOV L,A
044D 13	701	INX D
044E C9	702	RET
044F B1	703	CON1: POP D
0450 D1	704	POP D
0451 CD3300	705	JMP ERR
	706	-----
	707	-----
	708	! SUBRUTINA BINASC
	709	-----
	710	! EXECUTA AFISAREA
	711	! MODIFICA B,C
	712	! A=OCTET BINAR
	713	-----
0454 47	714	! BINASC: MOV B,A ! SALVARE OCTET

LOC OBJ	LINE	SOURCE STATEMENT
0455 OF	715	RRC
0456 OF	716	RRC
0457 OF	717	RRC
0458 OF	718	RRC
0459 CD6104	719	CALL B N1
045C 78	720	MOV A,B
045D CD6104	721	CALL B N1
0460 C9	722	RET
0461 E60F	723	B N1: OFH
0463 FE0A	724	CPI 10
0465 DA6A04	725	JC B N2
0468 C607	726	ADI 7
046A C630	727	ADI 7
046C 4F	728	MOV C,A
046D CD3602	729	CALL AFIS
0470 C9	730	RET
	731	-----
	732	-----
	733	! SUBRUTINA ASCB IN
	734	-----
	735	! EXECUTA CONVERSIE ASCII-BINAR
	736	! A=OCTET ASCII
	737	! A0-A3=SEMI-OCTET BINAR
	738	! MODIFICA: A
	739	-----
0471 D630	740	ASCB IN: SUI 30H
0473 FE0A	741	CPI 10
0475 D8	742	RC
0476 D607	743	SUI 7
0478 C9	744	RET
	745	-----
	746	! SUBRUTINA CRLF
	747	-----
	748	! CAP DE RIND - LINIE NOUA
	749	-----
0479 0E0D	750	CRLF: MVI C,0DH
047B CD3602	751	CALL AFIS
047E 0E0A	752	MVI C,0AH
0480 CD3602	753	CALL AFIS
0483 C9	754	RET
	755	-----
	756	! SUBRUTINA TEXT
	757	-----
	758	! TIPARESTE UN TEXT DIN MEMORIE
	759	! H-L ADRESA DE INCEPUT ZONA TEXT
	760	! B = CONTOR DE CARACTERE
	761	-----
0484 4E	762	TEXT: MOV C,M
0485 CD3602	763	CALL AFIS
0488 23	764	INX H
0489 05	765	DCR B
048A C28404	766	JNZ TEXT
048D C9	767	RET
	768	! SUBRUTINA AF20H
	769	! AFISEAZA BLANC LA DISPLAY

! A, D0F0H ! SEPARARA CUARTET L0M
! ADUCE CUARTET IN POZ A0-A3

OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
0E20	770	AF20H: MVI C,20H	04E6	79	025	MOV A,C
CD3602	771	CALL AF15	04E7	2F	826	CMA
C9	772	RET	04E8	3C	827	INR
	773		04E9	3C	828	CALL CKSMO
	774		04E2	CD6605	829	CALL CKSMO
	775	SUBROUTINA STORE	04EC	CD6605	830	RET
	776		04EF	C9	831	
	777	SALVEAZA PE CASETOFON 0 ZONA DE MEMORIE			832	
	778	CUPRINSA INTRE ADRESELE ADR1 SI ADR2			833	SUBROUTINA LOAD
	779				834	
CD7503	780	STORE: CALL CONV2			835	CITESTE UN FISIER DE LA CASETOFON
FE0D	781	RETUR			836	SINTAXA: LACR
C2D300	782	JNZ ERR			837	CITESTE DE PE CASETA IN MEMORIE LA ADRESA
EB	783	XCHG			838	CITITA DE PE CASETA
2A23A0	784	LHLD ADR1			839	
	785	MOV A,L	04F0	1A	840	LOAD: LDAX D
2F	786	CMA	04F1	FE0D	841	RETUR
6F	787	MOV L,A	04F2	C2D300	842	ERR
7C	788	MOV A,H	04F6	CDFC04	843	LTAPE
2F	789	CMA	04F9	C34F00	844	BUCLA
23	790	MOV H,A	04FC	D621	845	JMP IN 21H
19	791	INX H	04FE	47	846	MOV B,A
EB	792	DAD D	04FF	D621	847	IN 21H
2A23A0	793	XCHG EB	0501	A8	848	B
CD8204	794	LHLD ADR1	0502	CAFF04	849	SRII1
C34F00	795	CALL SRIOM	0505	D821	850	IN 21H
D5	796	JMP BUCLA	0507	E601	851	ANI 1
110000	797	PUSH D	0509	C20505	852	JNZ SRII2
	798	LXI B,0	050C	D621	853	IN 21H
0630	799	MVI B,30H	050E	E601	854	ANI 1
CD9E05	800	CALL IMFUL	0510	CA0C05	855	SRII3
	801	INX D	0513	CD305	856	BITR
FE20	802	MOV A,D	0516	3E1D	857	MVI A,10H
C2B604	803	CPI 20H	0518	B8	858	CMP B
060A	804	JNZ FRAMB	0519	DA0C05	859	JC SRII3
CD8E05	805	MVI B,0AH	051C	DE0A	860	C,O
D1	806	CALL IMPUL	051E	CD6E05	861	CKSMI
0E00	807	POP D	0521	67	862	H,A
7C	808	MVI C,0	0522	CD6E05	863	CKSMI
CD6605	809	MOV A,H	0525	4F	864	MOV L,A
7D	810	CALL CKSMO	0529	CD6E05	865	CALL CKSMI
CD6605	811	MOV A,L	0529	57	866	D,A
7A	812	CALL CKSMO	052A	CD6E05	867	CALL CKSMI
7B	813	MOV A,D	052D	5F	868	MVI E,A
2B	814	CALL CKSMO	052E	2225A0	869	SHLD ADR2
23	815	MOV A,E	0531	EB	870	INX XCHG
7E	816	CALL CKSMO	0532	23	871	INX H
23	817	DCX H	0533	2225A0	872	SHLD ADR2-2
7E	818	INX H	0535	2B	873	DCX H
CD6605	819	MOV A,M	0537	EB	874	XCHG
7A	820	CALL CKSMO	0539	2B	875	DCX H
B3	821	MOV A,D	0539	23	876	INX H
1B	822	ORA E	053A	CD6E05	877	CALL CKSMI
C2D804	823	DCX D	053D	77	878	MOV M,A
	824	JNZ TAPE1	053E	7A	879	MOV A,D

!SCRIE PE CASETA SUMA DE CONTROL
!IN COMPLEMENT FATA DE 2

!POSTAMBUL

!D,E=ADRESA SUPERIOARA

!COMPLEMENTUL '2

!D,E=CONTOR
!H,L=ADRESA INFERIOARA

!INITIALIZARE SUMA DE CONTROL
!SCRIE ADRINF SI CONTOR DE OCTETI

!CITESTE OCTET
!SALVEAZA-L PE CASETA

!RETA PINA LA CONTOR NUL

!ACC,B=DURATA IMPULS, CY=1

!SUMA DE CONTROL
!CITESTE ADRINF SI CONTOR DE
!PE CASETA

!PREGATESTE ZONA DE AFISAT

!D,E=ADR SUP -ADR INF
!H,L=ADRESA DE MEMORIE

!CITESTE OCTET

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
059F B3		890	ORA E	0598 D322		935	OUT 22H
0340 1B		881	DCX D	059A CDD005		936	CALL BITH
0541 C23905		882	TAPE2 CKSMI	059D C9		937	RET
0544 CD3E05		883	CALL			938	!CITESTE OCTET DE PE CASETA
0547 CA4F05		884	JZ ERG			939	SRIIN: PUSH D
054A F1		885	POP PSW			940	PUSH B
054B C3D300		886	JMP PSW			941	MVI E,8
054E C9		887	RET ERR			942	XRA A
054F 1602		888	ERG: D,2			943	SRII7: RLC
0551 2126A0		889	LX1 H,ADR2+1			944	MOV D,A
0554 7E		890	MOV A,M			945	SRII4: IN 21H
0555 CD5404		891	CALL BINASC			946	ANI 1
0558 2B		892	DCX H			947	JZ SRII4
0559 7E		893	MOV A,M			948	CALL BITR
055A CD5404		894	CALL BINASC			949	MVI A,18H
055D 2B		895	DCX H			950	CMP B
055E CD7904		896	CALL CRLF			951	JC SRII5
0561 15		897	DCR D			952	XRA A
0562 C25405		898	JNZ LMES			953	JMP SRII6
0565 C9		899	RET			954	SRII5: MVI A,1
		900	!CALCULEAZA SUMA DE CONTROL LA SCRIERE			955	SRII6: ORA D
0566 F5		901	CKSMI: PUSH PSW			956	E
0567 81		902	ADD C			957	JNZ SRII7
0568 4F		903	MOV C,A			958	POP B
0569 F1		904	POP PSW			959	POP D
056A CD7605		905	CALL SRIOT			960	RET
056D C9		906	RET			961	BITR: IN 21H
		907	!CALCULEAZA SUMA DE CONTROL LA CITIRE			962	MOV C,A
056E CD7E05		908	CKSMI: CALL SRIIN			963	MVI B,0
0571 47		909	MOV B,A			964	BITR1: IN B
0572 81		910	ADD C			965	IN C
0573 4F		911	MOV C,A			966	XRA C
0574 78		912	MOV A,B			967	JZ BITR1
0575 C9		913	RET			968	RET
		914	!SCRIE OCTET PE CASETA			969	BITM: IN 21H
0576 D5		915	SRIOT: PUSH D			970	XRA B
0577 1E08		916	MVI E,8			971	DCR B
0579 57		917	SRI03: MOV D,A			972	JNZ BITM
057A E660		918	MOV B0H			973	RET
057C 060E		919	MVI B,0EH			974	!-----
057E CA8305		920	JZ SRI02			975	!TABELA DE SIMBOLI
0581 0622		921	MVI B,22H			976	!-----
0583 CD8E05		922	CALL IMPUL			977	!
0586 7A		923	MOV A,D			978	!TABELA DE CUDURI A&CII PENTRU TASTATURA
0587 07		924	RLC			979	!SCAN/RETURN : 00,01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99
0588 1D		925	DCR E			980	TAB: DB
0589 C27905		926	JNZ SRI03				
058C D1		927	POP PSW				
058E C5		928	RET				
059F 3EFF		929	IMPUL: PUSH B				
0591 E322		930	MVI A,OFFH				
0593 CDD005		931	OUT 22H				
0596 C1		932	CALL BITH				
0597 AF		933	POP B				
		934	XRA A				
						981	0B
							20H, GHEITYUIOPE' 5CH,0AH,7FH,20H
						0506 09	
						0509 3123334	
						050D 35963738	
						05E1 3930203D	
						05E5 08	
						05E6 20	
						05E7 20	
						05E8 20	
						05E9 51574552	
						05ED 54525549	

LOC OBJ	LINE	SOURCE STATEMENT	LOC OBJ	LINE	SOURCE STATEMENT
0644 OD	982	ASDFGHJKL: .27H,1EH,0DH,20H,20H	0644 OD	990	DB 7AH,78H,63H,76H,62H,6EH,6DH,3CH
0645 20			0645 20		
0646 20			0646 20		
0647 20			0647 20		
0648 7A			0648 7A		
0649 78			0649 78		
064A 63			064A 63		
064B 76			064B 76		
064C 62			064C 62		
064D 6E			064D 6E		
064E 6D			064E 6D		
064F 3C			064F 3C		
0650 3E			0650 3E		
0651 3F			0651 3F		
0652 20			0652 20		
0653 00	983	'ZXCVBNM, .', 20H, 20H	0653 00	1001	DB 10H,10H,10H,10H,0,10H ;
0654 00			0654 00		
0655 00			0655 00		
0656 00			0656 00		
0657 00			0657 00		
0658 00			0658 00		
0659 10			0659 10		
065A 10			065A 10		
065B 10			065B 10		
065C 10			065C 10		
065D 00			065D 00		
065E 10			065E 10		
065F 00			065F 00		
0660 28			0660 28		
0661 00			0661 00		
0662 00			0662 00		
0663 00			0663 00		
0664 00			0664 00		
0665 00			0665 00		
0666 28			0666 28		
0667 7C			0667 7C		
0668 28			0668 28		
0669 7C			0669 7C		
066A 28			066A 28		
066B 10			066B 10		
066C 38			066C 38		
066D 50			066D 50		
066E 38			066E 38		
066F 14			066F 14		
0670 38			0670 38		
0671 00			0671 00		
0672 24			0672 24		
0673 00	984	'AMIC'	0673 00	1002	DB 0,28H,0,0,0,0 ;
0674 00	985	02H, '(e#%'.2*()--+', 8, 20H, 20H	0674 00		
0675 00			0675 00		
0676 00			0676 00		
0677 00			0677 00		
0678 00			0678 00		
0679 00			0679 00		
067A 00			067A 00		
067B 00			067B 00		
067C 00			067C 00		
067D 00			067D 00		
067E 00			067E 00		
067F 00			067F 00		
0680 00			0680 00		
0681 00			0681 00		
0682 00			0682 00		
0683 00			0683 00		
0684 00			0684 00		
0685 00			0685 00		
0686 00			0686 00		
0687 00			0687 00		
0688 00			0688 00		
0689 00			0689 00		
068A 00			068A 00		
068B 00			068B 00		
068C 00			068C 00		
068D 00			068D 00		
068E 00			068E 00		
068F 00			068F 00		
0690 00			0690 00		
0691 00			0691 00		
0692 00			0692 00		
0693 00			0693 00		
0694 00			0694 00		
0695 00			0695 00		
0696 00			0696 00		
0697 00			0697 00		
0698 00			0698 00		
0699 00			0699 00		
069A 00			069A 00		
069B 00			069B 00		
069C 00			069C 00		
069D 00			069D 00		
069E 00			069E 00		
069F 00			069F 00		
0690 00	986	20H, 74H, 77H, 65H, 72H, 74H, 79H, 75H	0690 00	1003	DB 0,28H,7CH,28H,7CH,28H ;
0691 00			0691 00		
0692 00			0692 00		
0693 00			0693 00		
0694 00			0694 00		
0695 00			0695 00		
0696 00			0696 00		
0697 00			0697 00		
0698 00			0698 00		
0699 00			0699 00		
069A 00			069A 00		
069B 00			069B 00		
069C 00			069C 00		
069D 00			069D 00		
069E 00			069E 00		
069F 00			069F 00		
0690 00	987	65H, 6FH, 70H, 5DH, 21H, 0AH, 7FH, 20H	0690 00	1004	DB 10H,36H,50H,38H,14H,38H ;
0691 00			0691 00		
0692 00			0692 00		
0693 00			0693 00		
0694 00			0694 00		
0695 00			0695 00		
0696 00			0696 00		
0697 00			0697 00		
0698 00			0698 00		
0699 00			0699 00		
069A 00			069A 00		
069B 00			069B 00		
069C 00			069C 00		
069D 00			069D 00		
069E 00			069E 00		
069F 00			069F 00		
0690 00	988	61H, 73H, 64H, 66H, 67H, 68H, 6AH, 6EH	0690 00	1005	DB 0,24H,6,10H,24H,0 ;
0691 00			0691 00		
0692 00			0692 00		
0693 00			0693 00		
0694 00			0694 00		
0695 00			0695 00		
0696 00			0696 00		
0697 00			0697 00		
0698 00			0698 00		
0699 00			0699 00		
069A 00			069A 00		
069B 00			069B 00		
069C 00			069C 00		
069D 00			069D 00		
069E 00			069E 00		
069F 00			069F 00		
0690 00	989	6CH, 3AH, 22H, 1BH, 0DH, 20H, 20H, 20H	0690 00		
0691 00			0691 00		
0692 00			0692 00		
0693 00			0693 00		
0694 00			0694 00		
0695 00			0695 00		
0696 00			0696 00		
0697 00			0697 00		
0698 00			0698 00		
0699 00			0699 00		
069A 00			069A 00		
069B 00			069B 00		
069C 00			069C 00		
069D 00			069D 00		
069E 00			069E 00		
069F 00			069F 00		

992 ;
 993 ;
 994 ; GENERATORUL DE CARACTERE
 995 ;
 996 ; STRUCTURA CAR ESTE:5*6 PCTE, INTR-O MATRICE DE 8*8
 997 ; PCTELE 1,7,8 DIN OCTET SINT FOLOSITE CA SEPARATORI DE CAR
 998 ; LINIILE 1 SI 8 SINT FOLOSITE CA SEPARATORI DE LINII DE CAR
 999 ;
 1000 CARGN: DB 0,0,0,0,0,0 ;BLANC

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
0673	08			06AA	00		
0674	10			06AB	00		
0675	24			06AC	10		
0676	00	1006	DB 20H,50H,20H,54H,48H,34H ;&	06AD	04	1015	DB 4,4,8,10H,20H,20H ;/
0677	20			06AE	04		
0678	50			06AF	08		
0679	20			06B0	10		
067A	54			06B1	20		
067B	48			06B2	20		
067C	34			06B3	38		
067D	08	1007	DB 8,10H,0,0,0,0	06B4	4C	1016	DB 38H,4CH,54H,54H,64H,38H ;0
067E	10			06B5	54		
067F	00			06B6	54		
0680	00			06B7	64		
0681	00			06B8	38		
0682	00			06B9	18		
0683	20	1008	DB 20H,40H,40H,40H,40H,20H ;(06BA	30	1017	DB 10H,30H,50H,10H,10H,38H ;1
0684	40			06BB	50		
0685	40			06BC	10		
0686	40			06BD	10		
0687	40			06BE	38		
0688	20			06BF	18		
0689	08	1009	DB 8,4,4,4,4,8 ;)	06C0	24	1018	DB 18H,24H,8,10H,20H,3CH ;2
068A	04			06C1	08		
068B	04			06C2	10		
068C	04			06C3	20		
068D	04			06C4	3C		
068E	08			06C5	38		
068F	00	1010	DB 0,10H,54H,38H,54H,10H ;#	06C6	04	1019	DB 38H,4,18H,4,4,38H ;3
0690	10			06C7	18		
0691	54			06C8	04		
0692	38			06C9	04		
0693	54			06CA	38		
0694	10			06CB	0C		
0695	00	1011	DB 0,10H,10H,7CH,10H,10H ;+	06CC	14	1020	DB 0CH,14H,24H,3CH,4,4 ;4
0696	10			06CD	24		
0697	10			06CE	3C		
0698	7C			06CF	04		
0699	10			06D0	04		
069A	10			06D1	3C		
069B	00	1012	DB 0,0,0,0,8,10H ;.	06D2	20	1021	DB 3CH,20H,38H,4,4,38H ;5
069C	00			06D3	38		
069D	00			06D4	04		
069E	00			06D5	04		
069F	08			06D6	38		
06A0	10			06D7	18		
06A1	00	1013	DB 0,0,0,7CH,0,0 ;-	06D8	20	1022	DB 18H,20H,38H,24H,24H,18H ;6
06A2	00			06D9	38		
06A3	00			06DA	24		
06A4	7C			06DB	24		
06A5	00			06DC	18		
06A6	00			06DD	3C		
06A7	00	1014	DB 0,0,0,0,0,10H ;.	06DE	04	1023	DB 3CH,4,8,10H,20H,20H ;7
06A8	00			06DF	08		
06A9	00			06E0	10		

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
06E1	20			0718	3C		
06E2	20			0719	10	1033	DB 10H,28H,44H,7CH,44H,44H,44H 1A
06E3	18	1024	DB 18H,24H,18H,24H,24H,18H,18H 18	071A	28		
06E4	24			071B	44		
06E5	18			071C	7C		
06E6	24			071D	44		
06E7	24			071E	44		
06E8	18			071F	78	1034	DB 78H,44H,78H,44H,44H,78H 1B
06E9	18	1025	DB 18H,24H,1CH,4,4,18H 19	0720	44		
06EA	24			0721	78		
06EB	1C			0722	44		
06EC	04			0723	44		
06ED	04			0724	78		
06EE	18			0725	38	1035	DB 38H,44H,40H,40H,44H,44H,38H 1C
06EF	00	1026	DB 0,10H,0,10H,0,0 11	0726	44		
06F0	10			0727	40		
06F1	00			0728	40		
06F2	10			0729	44		
06F3	00			072A	38		
06F4	00			072B	78	1036	DB 78H,44H,44H,44H,44H,78H 1D
06F5	00	1027	DB 5,10H,0,10H,20H,0 11	072C	44		
06F6	10			072D	44		
06F7	00			072E	44		
06F8	10			072F	44		
06F9	20			0730	78		
06FA	00			0731	7C	1037	DB 7CH,40H,7CH,40H,40H,7CH 1E
06FB	00	1028	DB 3,18H,24H,40H,20H,18H 11	0732	40		
06FC	18			0733	7C		
06FD	20			0734	40		
06FE	40			0735	40		
06FF	20			0736	7C		
0700	18			0737	7C	1038	DB 7CH,40H,7CH,40H,40H,40H 1F
0701	00	1029	DB 0,0,7CH,0,7CH,0 11	0738	40		
0702	00			0739	7C		
0703	7C			073A	40		
0704	00			073B	40		
0705	7C			073C	40		
0706	00			073D	38	1039	DB 38H,44H,40H,5CH,44H,38H 1G
0707	00	1030	DB 3,0H,5,4,0,10H 11	073E	44		
0708	30			073F	40		
0709	08			0740	5C		
070A	04			0741	44		
070B	08			0742	38		
070C	30			0743	44	1040	DB 44H,44H,7CH,44H,44H,44H 1H
070D	18	1031	DB 18H,24H,8,10H,0,10H 11	0744	44		
070E	24			0745	7C		
070F	08			0746	44		
0710	10			0747	44		
0711	00			0748	44		
0712	10			0749	38	1041	DB 38H,10H,10H,10H,10H,38H 1I
0713	38	1032	DB 38H,44H,58H,58H,40H,3CH 18	074A	10		
0714	44			074B	10		
0715	58			074C	10		
0716	58			074D	10		
0717	40			074E	38		

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
074F 3C		1042	DB 3CH, S, S, S, 48H, 30H	0786 40		1052	DB 7CH, 10H, 10H, 10H, 10H, 10H ; T
0750 08				0787 38			
0751 08				0789 04			
0752 08				0789 04			
0753 48				078A 78			
0754 30		1043	DB 48H, 50H, 60H, 50H, 48H, 44H ; K	078B 7C			
0755 48				078C 10			
0756 50				078D 10			
0757 60				078E 10			
0758 50				078F 10			
0759 48				0790 10			
075A 44		1044	DB 40H, 40H, 40H, 40H, 40H, 7CH ; L	0791 44		1053	DB 44H, 44H, 44H, 44H, 44H, 44H, 44H ; U
075B 40				0792 44			
075C 40				0793 44			
075D 40				0794 44			
075E 40				0795 44			
075F 40				0796 38			
0760 7C		1045	DB 44H, 6CH, 54H, 44H, 44H, 44H ; M	0797 44		1054	DB 44H, 44H, 44H, 44H, 44H, 28H, 10H ; V
0761 44				0798 44			
0762 6C				0799 44			
0763 54				079A 44			
0764 44				079B 28			
0765 44				079C 10			
0766 44				079D 44		1055	DB 44H, 44H, 44H, 44H, 54H, 68H, 44H ; W
0767 44		1046	DB 44H, 64H, 54H, 4CH, 44H, 44H ; N	079E 44			
0768 64				079F 44			
0769 54				07A0 54			
076A 4C				07A1 6C			
076B 44				07A2 44			
076C 44				07A3 44			
076D 38		1047	DB 38H, 44H, 44H, 44H, 44H, 38H ; O	07A4 28		1056	DB 44H, 28H, 10H, 10H, 10H, 28H, 44H ; X
076E 44				07A5 10			
076F 44				07A6 10			
0770 44				07A7 28			
0771 44				07A8 44			
0772 28				07A9 44		1057	DB 44H, 28H, 10H, 10H, 10H, 10H ; Y
0773 78		1048	DB 78H, 44H, 78H, 40H, 40H, 40H ; P	07AA 28			
0774 44				07AB 10			
0775 78				07AC 10			
0776 40				07AD 10			
0777 40				07AE 10			
0778 40				07AF 7C		1058	DB 7CH, 4, 18H, 50H, 40H, 7CH ; Z
0779 38		1049	DB 38H, 44H, 44H, 54H, 48H, 34H ; D	07B0 04			
077A 44				07B1 18			
077B 44				07B2 30			
077C 54				07B3 40			
077D 48				07B4 7C			
077E 34				07B5 60		1059	DB 60H, 40H, 40H, 40H, 40H, 60H ; I
077F 78		1050	DB 78H, 44H, 78H, 50H, 48H, 44H ; R	07B6 40			
0780 44				07B7 40			
0781 78				07B8 40			
0782 50				07B9 40			
0783 48				07BA 60			
0784 44				07BB 20		1060	DB 20H, 20H, 10H, 8, 4, 4 ; SLASH LEFT
0785 3C		1051	DB 3CH, 40H, 38H, 4, 4, 78H ; S	07BC 20			

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 27

LOC	OBJ	LINE	SOURCE STATEMENT
07BD	10		
07BE	08		
07BF	04		
07C0	04		
07C1	0C	1061	DB 0CH,4,4,4,4,0CH ;1
07C2	04		
07C3	04		
07C4	04		
07C5	04		
07C6	0C		
07C7	10	1062	DB 10H,28H,44H,0,0,0 ;CARETA
07C8	28		
07C9	44		
07CA	00		
07CB	00		
07CC	00		
07CD	00	1063	DB 0,0,0,0,0,7CH ;BARA JOS
07CE	00		
07CF	00		
07D0	00		
07D1	00		
07D2	7C		
07D3	41204620	1064	TEXT2: DB 'A F B C D E H L SP PC'
07D7	20422043		
07DB	20204420		
07DF	45202048		
07E3	204C2020		
07E7	53502020		
07EB	205043		
		1065	;
		1066	;TABELA DE JUMP-URI LA ADRESE IMPORTANTE DIN MONITOR
		1067	;
07F4		1068	ORG 7F4H
07F4	C3B204	1069	POUT: JMP SRIOM
07F7	C3FC04	1070	RIN: JMP LTAPE
07FA	C33602	1071	COUT: JMP AFIS
07FD	C30101	1072	CIN: JMP KEYIN
		1073	;
		1074	;ECHIVALARI SI REZERVARI DE MEMORIE
		1075	;
0023		1076	CWPPI EQU 23H
8002		1077	ADRIN EQU 8002H ;ADRESA INCEPUT ECRAN
007F		1078	DEL EQU 7FH
000D		1079	RETUR EQU 0DH
0800		1080	BASIC EQU 800H
0020		1081	PORTA EQU 20H
0021		1082	PORTB EQU 21H
0022		1083	PORTC EQU 22H
8000		1084	ADREC EQU 8000H ;ADRESA ECRAN
A100		1085	USESP EQU 0A100H ;STIVA UTILIZATOR
A000		1086	ORG 0A000H
A000		1087	RND: DS 1 ;POINTERI TV
A001		1088	COL: DS 1
A002		1089	AFMOD: DS 1 ;SCROLL/PAGE 00/FF
A003		1090	EINV: DS 1 ;VIDEO NORM/INV ECRAN

LOC	OBJ	LINE	SOURCE STATEMENT
A004		1091	VIINV: DS 1
A005		1092	CDAL: DS 17
A016		1093	STVUT: DS 12
A022		1094	CONT: DS 1
A023		1095	ADR1: DS 2
A025		1096	ADR2: DS 2
A027		1097	ADR3: DS 2
A029		1098	TAMP: DS 8
A031		1099	DS 8
A059		1100	MONSP: DS 1
		1101	END

!VIDEO NORM/INV CAR
!BUFFER COMANDA
!STIVA UTILIZATOR
!CONTOR

!TAMFON CARACTERE SEMIGRAFICE

!STIVA MONITOR

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

ADR1	A 0023	ADR2	A 0025	ADR3	A 0027	ADREC	A 0000	ADRLN	A 0002	AF1	A 0246	AF11	A 0240
AF12	A 029D	AF20H	A 025C	AF20H	A 048E	AF3	A 026B	AFIS	A 0236	AFMOD	A 0002	ASCBIN	A 0471
BASIC	A 0800	BIP	A 0454	BIP	A 015D	BIP1	A 0161	BITR	A 05C3	BITR1	A 05C8	BITW	A 05D0
BN1	A 0461	BUCLA	A 046A	BUCLA	A 004F	CARGN	A 0453	CA	A 0005	CHND	A 0349	CIN	A 07FD
CKSM1	A 056E	CMR	A 0566	CMR	A 0057	COL	A 0001	CON1	A 044F	CONT	A 0022	CONV1	A 033B
CONV2	A 0375	CONVA	A 0428	CONVB	A 0433	COUT	A 07FA	CRLF	A 0479	CMPP1	A 0023	DEL	A 007F
DISP	A 02B1	EINV	A 0003	ER1	A 038A	ERG	A 054F	ERR	A 00D3	ET10	A 012D	ET11	A 013B
ET19	A 02A7	ET2	A 011E	ET3	A 0104	ET41	A 00DB	ET42	A 005C	ET43	A 0088	ET45	A 0033
ET46	A 002E	ET5	A 0136	ET50	A 02D6	ET51	A 02BD	ET52	A 02C5	ET6	A 0144	ET61	A 0337
ET62	A 0325	ET63	A 030D	ET64	A 02EC	ET7	A 013C	ET71	A 0355	ET72	A 041B	ET75	A 03BA
ET77	A 03AA	ET8	A 014E	ET81	A 0402	ET9	A 014F	EXAM	A 033F	FIL1	A 03D3	FILL	A 03BF
GO	A 03E3	IMPUL	A 058E	INITV	A 02A4	INVHL	A 0416	KEY	A 00EC	KEY1	A 00F2	KEYIN	A 0101
LINES	A 0554	LOAD	A 04FO	LYAFE	A 04FC	MONSP	A 0059	MOVE	A 03EE	PORTA	A 0020	PORTB	A 0021
PORTC	A 0022	POUT	A 07F4	PRAMB	A 04B6	RETR	A 000D	RIN	A 07F7	RND	A 0000	SCRI	A 021F
SCR2	A 022C	SCROL	A 0219	SRI11	A 04FF	SRI12	A 0505	SRI13	A 050C	SRI14	A 05A5	SRI15	A 05B9
SRI16	A 05BB	SRI17	A 05A3	SRIIN	A 059E	SRI02	A 0583	SRI03	A 0579	SRJOM	A 04B2	SRJOT	A 0576
START	A 0041	STORE	A 0494	STVUT	A 0016	SUBST	A 02F3	TAB	A 05D8	TAMP	A 0029	TAPE1	A 04DB
TAPE2	A 0539	TAPE4	A 052E	TEXT	A 0484	TEXT1	A 0614	TEXT2	A 07D3	USESP	A 0100	VINV	A 0004
WR20	A 01C9	WR21	A 01C4	WR22	A 01F7	WR23	A 01F2	WR24	A 0203	WR25	A 01FE	WR26	A 0205
WR27	A 01B9	WR30	A 01E3	WR32	A 01DF	WR40	A 0192	WR41	A 019B	WR43	A 0189	WR44	A 019D
WRITE	A 017A												

ASSEMBLY COMPLETE. NO ERRORS

Monitor-Asamblor-Text Editor. (MATE). Listing sursă

ASM80 :F1:MATE2.ASM PRINT(:LP1)

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0 MODULE PAGE 1

LOC	OBJ	LINE	SOURCE STATEMENT
		1 ;	*****
		2 ;	* M A T E *
		3 ;	*****
		4 ;	
		5 ;	MONITOR - ASAMELOR - TEXT EDITOR
		6 ;	PENTRU MICROCALCULATORUL PERSONAL AMIC
		7 ;	
0000		8	ORG 0 ;PROGRAMUL DE MONITOR INCEPE LA ADRESA 0
0000	C31900	9	START: JMP INITA ;SALT LA ADRESA DE START MONITOR
0003	C35900	10	JMP EOR ;RESTART MONITOR
0006	00	11	NOP
0007	00	12	NOP
0008	C3500D	13	JMP BRKP ;RESTART PUNCT DE INTRERUPERE
000B	00	14	NOP
000C	00	15	NOP
000D	C34B0F	16	JMP SCAN ;CITESTE CARACTER DE LA CONSOLA
0010	C37A10	17	JMP AFIS ;AFISEAZA CARACTER LA CONSOLA
0013	C3EF11	18	JMP LTAPE ;CITESTE CARACTER DE LA CASETOFOON
0016	C3A111	19	JMP STAPE ;TRIMITE CARACTER LA CASETOFOON
0019	212460	20	INITA: LXI H,FILEO ;INITIALIZEAZA CU O ZONA DE MEMORIE CE CONTINE
001C	016001	21	LXI B,160H ;INFORMATII DESPRE FISIERELE DIN EVIDENTA SISTEMULUI
001F	AF	22	XRA A ;ZONA INCEPE LA ADRESA FILEO
0020	77	23	INIT2: MOV M,A ;SI ARE LUNGIMEA ADOPERITOARE DE 160H
0021	23	24	INX H
0022	0D	25	DCR C
0023	C22000	26	JNZ INIT2
0026	05	27	DCR B
0027	C22000	28	JNZ INIT2
		29	;INIT. ZONA BREACKPOINT
002A	0618	30	MVI B,NBR*3 ;INITIALIZEAZA CU O ZONA CE CONTINE INFORMATII
002C	210C60	31	LXI H,BRT ;DESPRE BREAKPOINT-URI
002F	77	32	INIT3: MOV M,A ;ZONA INCEPE LA ADRESA BRT
0030	23	33	INX H ;SI ARE LUNGIMEA NBR*3
0031	05	34	DCR B
0032	C22F00	35	JNZ INIT3
		36	;INITV SI INISC
4000		37	BAZTV EQU 4000H
0035	AF	38	ECLER: XRA A ;INITIALIZEAZA POINTERII DE AFISARE
0036	326E61	39	STA COL ;NUMARUL COLOANEI INTRE 0 SI 29
0039	326D61	40	STA LIN ;NUMARUL LINIEI INTRE 0 SI 31
003C	210040	41	LXI H,BAZTV ;STERGE ECRANUL
003F	36FF	42	REIA: MVI M,OFFH ;PRIN INSCRIEREA VALORII FFH
0041	23	43	INX H ;IN INTREAGA MEMORIE VIDEO
0042	7C	44	MOV A,H ;INTRE ADRESELE 4000H-5FFFH
0043	FEA0	45	CPI 0A0H
0045	C23F00	46	JNZ REIA
0023		47	CH55 EQU 23H ;PORTUL DE COMANDA PENTRU 8255
0048	3E92	48	MVI A,92H ;COMINTUL DE COMANDA
004A	D323	49	OUT CH55 ;TRIMITE LA CIRCUITUL 8255
004C	AF	50	XRA A
004D	326F61	51	STA MCAP ;VARIABILA MCAP MEMOREAZA POZITIA TASTEI SHIFT LOCK
0050	217200	52	LXI H,AMON ;AFISEAZA "MATE"
0053	310461	53	LXI SP,AREA+100 ;INITIALIZEAZA INDICATOR DE STIVA
0056	CD5402	54	CALL SCRN

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 2

LOC	OBJ	LINE	SOURCE STATEMENT
		55	;MONITORUL
0059	310461	56	EOR: LXI SP,AREA+100 ;LA INTRAREA IN MONITOR FARA INITIALIZARE
005C	CD4000	57	CALL CRLF ;SE INITIALIZEAZA STIVA
005F	CD7700	58	CALL READ ;CITESTE CARACTERE DE LA TASTATURA PINA LA CR
0062	23	59	INX H ;DACA PRIMUL CARACTER INTRODUS ESTE CIFRA
0063	7E	60	MOV A,M ;SALT LA ADRESA LINE
0064	FE3A	61	CPI '9'+1
0066	DABF04	62	JC LINE
0069	CD4D01	63	CALL VALC ;PREIA PARAMETRII COMENZII
006C	CD0501	64	CALL COMM ;IDENTIFICA COMANDA SI SALT LA ADRESA UNDE SE EXECUTA
006F	C35900	65	JMP EOR ;REIA CICLUL
0072	4D415445	66	AMON: DB 'MATE',13 ;MESAJ MONITOR
0076	00		
		67	:
		68	;
		69	;SUBROUTINA READ
		70	;
		71	;CITESTE PINA LA CR
		72	:
0077	211A61	73	READ: LXI H,IBUF ;ADRESA INCEPUT BUFFER
007A	227460	74	SHLD ADDS
007D	1E02	75	MVI E,2 ;INITIALIZARE CONTOR CARACTERE
007F	CDE000	76	NEXT: CALL INB ;CITESTE CARACTER
0082	78	77	MOV A,B
0083	FE18	78	CPI 24 ;ESTE CTRL/X ?
0085	C28E00	79	JNZ ECR
0088	CD4000	80	CALL CRLF ;DACA DA, RENUNTA LA LINIA CURENTA
008B	C37700	81	JMP READ ;CITESTE UN RIND NOU
008E	FE0D	82	ECR: CPI ASCR ;ESTE CR ?
0090	C2A900	83	JNZ DEL
0093	7D	84	MOV A,L ;DACA DA SI DACA BUFFER GOL
0094	FE1A	85	CPI IBUF AND OFFH ;CITESTE O NOUA LINIE
0096	CA7700	86	JZ READ
0099	360D	87	MVI M,ASCR ;INSCRIE CR IN BUFFER
009B	23	88	INX H
009C	3601	89	MVI M,1 ;SFIRSIT DE LINIE
009E	23	90	INX H
009F	3E6D	91	MVI A,IBUF+83 AND OFFH ;INTRODUCE BLANCURI PINA LA SFIRSITUL
00A1	CD0800	92	CALL CLER ;ZONEI BUFFER
00A4	211961	93	LXI H,IBUF-1 ;CONTOR BUFFER LA ADRESA IBUF-1
00A7	73	94	MOV M,E
00A8	C9	95	RET
00A9	FE7F	96	DEL: CPI 127 ;ESTE DEL ?
00AB	C2BE00	97	JNZ CHAR
00AE	3E1A	98	MVI A,IBUF AND OFFH ;DACA DA SI DACA BUFFER GOL
00B0	BD	99	CMPL ;PREIA URMATORUL CARACTER
00B1	CA7F00	100	JZ NEXT
00B4	2B	101	DCX H ;ACTUALIZEAZA CONTOARE
00B5	1D	102	DCR E
00B6	065F	103	BSPA: MVI B,SFH ;AFISEAZA CURSOR
00B8	CDE700	104	CALL OUTS
00BB	037F00	105	JMP NEXT ;SALT LA PRELUAREA URMATORULUI CARACTER
00BE	FE20	106	CHAR: CPI ' ' ;REJECTEAZA CARACTERE CU COD ASCII
00C0	DA7F00	107	JC NEXT ;MAI MIC DECIT 20H
00C3	FEB8	108	CPI 'Z'+1 ;SAU MAI MARE DECIT 5AH

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 3

LOC	DBJ	LINE	SOURCE STATEMENT
00C5	D27F00	109	JNC NEXT
00C8	47	110	MOV B,A
00C9	CDE700	111	CALL OUT8 ;AFISEAZA CARACTER LA DISPLAY
00CC	70	112	MOV M,B ;INSCRIE CARACTER IN BUFFER
00CD	3E68	113	MVI A,IBUF+81 AND OFFH ;BUFFER PLIN ?
00CF	BD	114	CMP L
00D0	CAB600	115	JZ BSPA
00D3	23	116	INX H ;ACTUALIZEAZA CONTOARE
00D4	1C	117	INR E
00D5	C37F00	118	JMP NEXT ;SALT LA PRELUAREA UNUI NOU CARACTER
		119	;
		120	-----
		121	;SUBROUTINA CLER
		122	-----
		123	;PUNE BLANC INTR-O ZONA DE MEM
		124	;
00D8	BD	125	CLER: CMP L ;PUNE BLANCURI INTRO ZONA DE MEMORIE
00D9	C8	126	RZ ;INCREMENTEAZA H,L
00DA	3620	127	MVI M,' ;PINA CIND L=A
00DC	23	128	INX H
00DD	C3D800	129	JMP CLER
		130	;
		131	-----
		132	;SUBROUTINA IN8
		133	-----
		134	;CITESTE UN CAR
		135	;
00E0	F5	136	IN8: PUSH PSW ;CITESTE CARACTER DE LA TASTATURA
00E1	CD480F	137	CALL SCAN ;ASTEAPTA INTRODUCEREA CARACTERULUI
00E4	47	138	MOV B,A ;FURNIZEAZA CODUL ASCII AL CARACTERULUI
00E5	F1	139	POP PSW ;NU AFECTEAZA REGISTRELE
00E6	C9	140	RET
		141	;
		142	-----
		143	;SUBROUTINA OUT8
		144	-----
		145	;SCRIE CAR. LA TV.
		146	;
00E7	F5	147	OUT8: PUSH PSW ;SCRIE UN CARACTER PE ECRANUL TELEVIZORULUI
00E8	C5	148	PUSH B ;PREIA CODUL ASCII AL CARACTERULUI DIN B
00E9	48	149	MOV C,B ;NU AFECTEAZA REGISTRELE
00EA	CD7A10	150	CALL AFIS
00ED	C1	151	POP B
00EE	F1	152	POP PSW
00EF	C9	153	RET
		154	;
		155	-----
		156	;SUBROUTINA CRLF
		157	-----
		158	;CAP DE RIND , LINIE NOUA
		159	;
00F0	0E0D	160	CRLF: MVI C,CR ;TRECE LA INCEPUTUL URMATOAREI LINII TV
00F2	CD7A10	161	CALL AFIS ;AFECTEAZA REGISTRELE A,C
00F5	0E0A	162	MVI C,LF
00F7	CD7A10	163	CALL AFIS

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 4

LOC	OBJ	LINE	SOURCE STATEMENT
00FA	C9	164	RET
		165	;
		166	-----
		167	;COMANDA EXECUTE
		168	-----
		169	;
00FB	CDFE02	170	EXEC: CALL VCHK ;VERIFICA EXISTENTA PARAMETRILO
00FE	CDF000	171	CALL CRLF ;CAP DE RIND LINIE NOUA
0101	2ABA60	172	LHLD BBUF ;IN H,L ADRESA DE SALT
0104	E9	173	PCHL ;SALT
		174	;IDENTIFICA COMANDA
0105	11A402	175	COMM: LXI D,CTAB ;ADRESA TABELEI DE COMENZI IN D,E
0108	060F	176	MVI B,NCOM ;NUMARUL COMENZII IN REGISTRUL B
010A	3E04	177	MVI A,4 ;LUNGIME COMANDA
010C	329560	178	STA NCHR
010F	CD1601	179	CALL COMS ;IDENTIFICA COMANDA
0112	C26404	180	JNZ WHAT ;COMANDA ERONATA
0115	E9	181	PCHL ;SALT LA ADRESA DE EXECUTIE
		182	;
		183	-----
		184	;SUBROUTINA COMS
		185	-----
		186	;CAUTA UN SIMBOL AFLAT LA ADRESA DATA LA ADRESA ADDS
		187	;AVIND LUNGIMEA DATA LA ADRESA NCHR
		188	;CAUTAREA SE FACE INTR-O TABELA CU ADRESA DE INCEPUT DATA IN D,E
		189	;TABELA CONTINE UN NUMAR DE SIMBOLI PRECIZAT IN REGISTRUL B
		190	;RASPUNSURI: Z=0 SIMBOL NEGASIT
		191	; Z=1 SIMBOL GASIT, H,L CONTIN ADRESA AFLATA IN TABELA SUB SIMBOL
		192	;
0116	2A7460	193	COMS: LHLD ADDS ;ADRESA SIMBOL CAUTAT
0119	3A9560	194	LDA NCHR ;LUNGIME SIMBOL
011C	4F	195	MOV C,A
0110	CD2D01	196	CALL SEAR ;CAUTA SIMBOL LA ADRESA DATA IN D,E
0120	1A	197	LDAX D ;OCTETUL DE DUPA SIMBOL ESTE PUS IN L
0121	6F	198	MOV L,A
0122	13	199	INX D ;URMATORUL OCTET ESTE PUS IN H
0123	1A	200	LDAX D
0124	67	201	MOV H,A
0125	C8	202	RZ ;SIMBOL GASIT
0126	13	203	INX D ;URMATOAREA ADRESA IN TABELA
0127	05	204	DCR B
0128	C21601	205	JNZ COMS ;REIA DACA MAI SINT SIMBOLI IN TABELA
012B	04	206	JNR B ;Z=0
012C	C9	207	RET
		208	-----
		209	;SUBROUTINA SEAR
		210	-----
		211	;COMPARA DOUA SIRURI DE CARACTERE DIN MEMORIE
		212	;INTRARI: D,E=ADRESA DE INCEPUT A SIRULUI CU CARE SE COMPARA
		213	; H,L=ADRESA DE INCEPUT A SIRULUI CARE SE COMPARA
		214	; C=NUMAR DE CARACTERE DIN SIR
		215	;IESIRI: Z=1 SIRURI IDENTICE
		216	; Z=0 SIRURI DIFERITE
		217	; D,E=ADRESA DE DUPA ULTIMUL CARACTER DIN SIRUL CARE SE COMPARA
		218	;

SF0X-10 8090/8085 MACRO ASSEMBLER, V3.0 MODULE PAGE 9

LOC	OBJ	LINE	SOURCE STATEMENT
012D	1A	219	SEAR: LDAX D
012E	BE	220	CMP M
012F	C23901	221	JNZ INCA
0132	23	222	INX H
0133	13	223	INX D
0134	0D	224	DCR C
0135	C22D01	225	JNZ SEAR
0138	C9	226	RET
0139	13	227	INCA: INX D
013A	0D	228	DCR C
013B	C23901	229	JNZ INCA
013E	0C	230	INR C
013F	C9	231	RET
		232	;
		233	-----
		234	;SUBROUTINA ABUF
		235	-----
		236	;STERGE ABUF=12 OCT.
		237	;
0140	AF	238	ZBUF: XRA A ;PUNE 0 IN ZONA DE MEMORIE DE LA ABUF LA ABUF+12
0141	118A60	239	LXI D,ABUF+12 ;LA REVENIRE D,E CONTINE ADRESA ABUF
0144	060C	240	MVI B,12
0146	1B	241	ZBU1: DCX D
0147	12	242	STAX D
0148	05	243	DCR B
0149	C24601	244	JNZ ZBU1
014C	C9	245	RET
		246	;DETERMINAREA PARAMETRILOR COMENZII
014D	CD5401	247	VALC: CALL ETRA ;PREIA PARAMETRII COMENZII
0150	DA6404	248	JC WHAT ;SEMNALIZEAZA EROARE
0153	C9	249	RET
0154	210000	250	ETRA: LXI H,0 ;INITIALIZEAZA BBUF (BUFFER BINAR)
0157	228C60	251	SHLD BBUF+2 ;ABUF (BUFFER ASCII)
015A	227660	252	SHLD FBUF ;FBUF (BUFFER NUME FISIER)
015D	CD4001	253	CALL ZBUF
0160	211961	254	LXI H,IBUF-1
0163	23	255	VAL1: INX H ;PRIMUL CARACTER DIN BUFFER-UL IBUF
0164	7E	256	MOV A,M
0165	FE20	257	CPI
0167	3F	258	CMC ;TEST CR
0168	D0	259	RNC
0169	C26301	260	JNZ VAL1 ;SALT PESTE NUMELE COMENZII
016C	229660	261	SHLD PNTR
016F	CD3A09	262	CALL SBLK ;SALT PESTE BLANCURI
0172	3F	263	CMC ;PRIMUL CARACTER DUPA NUME CDA ESTE CR ?
0173	D0	264	RNC
0174	FE2F	265	CPI
0176	C29E01	266	JNZ VAL5 ;EXISTA NUME FISIER ?
0179	117660	267	LXI D,FBUF ;PREIA NUME FISIER LA ADRESA FBUF
017C	0E05	268	MVI C,NMLEN
017E	23	269	VAL2: INX H
017F	7E	270	MOV A,M
0180	FE2F	271	CPI
0182	C8E01	272	JZ VAL3
0185	0D	273	DCR C

BFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 6

LOC	OBJ	-LINE	SOURCE STATEMENT
0186	FA6404	274	JM WHAT ;SEMNALIZEAZA DACA NUME FISIER PREA LUNG
0189	12	275	STAX D
018A	13	276	INX D
018B	C37E01	277	JMP VAL2
018E	3E20	278	MVI VAL3: A, ' ' ;COMPLETEAZA CU BLANCURI ZONA DIN FBUF
0190	0D	279	VAL4: DCR C ;DE DUPA NUME FISIER
0191	FA9901	280	JM DONE1
0194	12	281	STAX D
0195	13	282	INX D
0196	C39001	283	JMP VAL4
0199	CD4109	284	DONE1: CALL SBL2 ;CAUTA PRIMUL CARACTER DIFERIT DE BLANC
019C	3F	285	CMC ;DE DUPA NUME FISIER
019D	D0	286	RNC
019E	117E60	287	VAL5: LXI D,ABUF ;PARAMETRUL 1 IN ABUF
01A1	CDA20B	288	CALL ALPS
01A4	78	289	MOV A,B
01A5	FE05	290	CPI 5 ;SEMNALIZEAZA DACA PARAMETRU PREA LUNG
01A7	3F	291	CMC
01A8	D8	292	RC
01A9	017E60	293	LXI B,ABUF ;CONVERSIE PARAMETRU IN BINAR
01AC	CDF501	294	CALL AHEX ;SI SEMNALIZEAZA EVENTUALE ERORI
01AF	D8	295	RC
01B0	228A60	296	SHLD BBUF ;PARAMETRU 1 IN BINAR IN BBUF
01B3	217E60	297	LXI H,ABUF ;COMPLETEAZA CU 0 BBUF
01B6	CDC705	298	CALL NORM
01B9	CD3A09	299	CALL SBLK ;CAUTA URMATORUL PARAMETRU
01BC	3F	300	CMC ;TEST CR
01BD	D0	301	RNC
01BE	118260	302	LXI D,ABUF+4 ;PARAMETRUL 2 IN ABUF
01C1	CDA20B	303	CALL ALPS
01C4	78	304	MOV A,B
01C5	FE05	305	CPI 5 ;SEMNALIZEAZA DACA PARAMETRU PREA LUNG
01C7	3F	306	CMC
01C8	D8	307	RC
01C9	018260	308	LXI B,ABUF+4 ;PARAMETRUL 2 IN BINAR IN BBUF
01CC	CDF501	309	CALL AHEX
01CF	D8	310	RC
01D0	228C60	311	SHLD BBUF+2
01D3	218260	312	LXI H,ABUF+4 ;COMPLETEAZA CU 0 PARAMETRUL 2 IN ABUF
01D6	CDC705	313	CALL NORM
01D9	B7	314	ORA A ;CY=0
01DA	C9	315	RET
		316	;CONVERSII
		317	;
		318	;
		319	;SUBROUTINA ADEC
		320	;
		321	;TRANSFORMA UN SIR DE CARACTERE HEXA CE REPREZINTA NUMERE ZECIMALE
		322	;INTR-UN NUMAR BINAR
		323	;INTRARI: B,C=ADRESA SIRULUI DE CARACTERE, TERMINAT CU 0
		324	;IESIRI: H,L=VALOAREA IN BINAR
		325	;
01DB	210000	326	ADEC: LXI H,0
01DE	0A	327	ADE1: LDAX B ;INCARCA UN CHARACTER
01DF	B7	328	ORA A

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 7

LOC	OBJ	LINE	SOURCE STATEMENT
01E0	C8	329	RZ
01E1	54	330	MOV D,H ;RETURN LA PRIMUL 0 INTILNIT
01E2	5D	331	MOV E,L ;TRANSFORMA VALOAREA BINARA CURENTA
01E3	29	332	INAD H ;INTR-O VALOARE DE 10 ORI MAI MARE
01E4	29	333	DAD H
01E5	19	334	DAD H
01E6	29	335	DAD H
01E7	D630	336	SUI 48 ;TRANSFORMA CARACTER HEXA IN BINAR
01E9	FE0A	337	CPI 10 ;SI SEMNALIZEAZA ERGAREA
01EB	3F	338	CMC
01EC	D8	339	RC
01ED	5F	340	MOV E,A ;ADUNA VALOAREA BINARA A NOULUI CARACTER
01EE	1600	341	MVI D,0 ;SI REIA
01F0	19	342	DAD D
01F1	03	343	INX B
01F2	C3LE01	344	JMP ADE1
		345 ;	
		346 ;	
		347 ;SUBROUTINA AHEX	
		348 ;	
		349 ;CONVERSIE SIR DE CARACTERE HEXA IN BINAR	
		350 ;INTRARI: B,C=ADRESA SIRULUI DE CARACTERE TERMINAT CU 0	
		351 ;IESIRI: H,L=CONTINE VALOAREA BINARA A ULTIMELOR 4 CARACTERE DIN SIR	
		352 ;	
		353 ; CY=1 ERGARE	
		354 ; CY=0 SFIRSIT NORMAL	
01F5	210000	355 AHEX:	LXI H,0
01F8	0A	356 AHE1:	LDAX B ;CITESTE CARACTER
01F9	B7		ORA A
01FA	C8	358	RZ
01FB	29	359	DAD H ;SFIRSIT CONVERSIE LA PRIMUL 0
01FC	29	360	DAD H ;DEPLASARE STINGA CU 4 POZITII
01FD	29	361	DAD H
01FE	29	362	DAD H
01FF	CD0C02	363	CALL AHS1 ;CONVERSIE ASCII-BINAR
0202	FE10	364	CPI 10H ;SEMNALIZEAZA DACA ERGARE
0204	3F	365	CMC
0205	D8	366	RC
0206	85	367	ADD L ;INTRODUCE NOUL CARACTER PE ULTIMELE
0207	6F	368	MOV L,A ;4 POZITII IN H,L
0208	03	369	INX B
0209	C3F801	370	JMP AHE1 ;RELUARE
020C	D630	371 AHS1:	SUI 48 ;CONVERSIE HEXA-BINAR PENTRU UN CARACTER
020E	FE0A	372	CPI 10
0210	D8	373	RC
0211	D607	374	SUI 7
0213	C9	375	RET
		376 ;	
		377 ;	
		378 ;SUBROUTINA HOUT	
		379 ;	
		380 ;AFISEAZA UN OCTET BINAR SUB FORMA A DOUA CARACTERE HEXA	
		381 ;INTRARI: A=CARACTER BINAR	
		382 ;MODIFICA: A,B,H,L	
		383 ;	

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 8

LOC	OBJ	LINE	SOURCE STATEMENT
0214	CD6C02	384	HOUT: CALL BINH ;CONVERSIE BINAR HEXA
0217	217460	385	LXI H ,HCON
021A	46	386	CHOT: MOV B,M ;AFISEAZA CARACTER HEXA
021B	CDE700	387	CALL OUTS
021E	23	388	INX H
021F	46	389	MOV B,M ;AFISEAZA CARACTER HEXA
0220	CDE700	390	CALL OUTS
0223	C9	391	RET
0224	CD1402	392	HOTE: CALL HOUT ;APEL HOUT
0227	CD3702	393	CALL BLK1 ;AFISARE BLANC DUPA CELE DOUA CARACTERE HEXA
022A	C9	394	RET
		395	;
		396	-----
		397	;SUBROUTINA DOUT
		398	-----
		399	;CONVERTESTE O VALOARE BINARA IN ASCII ZECIMAL SI O AFISEAZA
		400	;
022B	CD8902	401	DOUT: CALL BIND
022E	CD1702	402	CALL HOUT+3
0231	23	403	INX H
0232	46	404	MOV B,M
0233	CDE700	405	CALL OUTS
0236	C9	406	RET
0237	0620	407	BLK1: MVI B, ' ' ;AFISEAZA BLANC LA TELEVIZOR
0239	CDE700	408	CALL OUTS
023C	C9	409	RET
		410	;
		411	-----
		412	;SUBROUTINA ACHK
		413	-----
		414	;COMPARA ADRESELE DE LA ADRESELE BBUF SI BBUF+2
		415	;IESIRI: CY=0 ADRESE DIFERITE
		416	; CY=1 ADRESE EGALE
		417	;INCREMENTEAZA ADRESA DE LA ADRESA BBUF
		418	;MODIFICA: A,H,L
		419	;
023D	2A8A60	420	ACHK: LHLD BBUF
0240	3A8D60	421	LDA BBUF+3
0243	BC	422	CMP H ;COMPARA ((BBUF+3)) CU ((BBUF+1))
0244	C24F02	423	JNZ ACH1
0247	3A8C60	424	LDA BBUF+2 ;COMPARA ((BBUF+2)) CU ((BBUF))
024A	BD	425	CMP L
024B	C24F02	426	JNZ ACH1
024E	37	427	STC ;CY=1
024F	23	428	ACH1: INX H
0250	228A60	429	SHLD BBUF ;INCREMENTEAZA ADRESA DE LA ADRESA BBUF
0253	C9	430	RET
		431	;
		432	-----
		433	;SUBROUTINA SCRNI
		434	-----
		435	;SCRIE SIR DE CAR PINA LA (CR) LA IV
		436	;INTRARI: H,L=ADRESA DE INCEPUT A TEXTULUI CE URMEAZA SA FIE SCRIS
		437	;IESIRI: SE AFISEAZA TEXTUL LA TELEVIZOR
		438	;DUPA FIECARE CARACTER AFISAT SE TESTEAZA DACA NU S-A APASAT TASTA S (CTRL/S)

*DIX-12 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 9

LOC	OF	LINE	SOURCE STATEMENT
		439	:DACA NU, SE AFISEAZA CARACTERUL URMATOR
		440	:DACA DA, NU SE REIA AFISAREA PINA CIND NU SE AFASA TASTA 0 (CTRL/0)
		441	:MODIFICA: A,B
		442	:
0254	46	443	SCRN: MOV B,M ;PREIA CARACTER
0255	3E0D	444	MVI A,13 ;CR IN REGISTRUL A
0257	88	445	CMF B
0258	C8	446	RZ
0259	CDE700	447	CALL OUTB ;AFISEAZA CARACTER
025C	CDF706	448	CALL CTRLS ;TEST CTRL/S
025F	C26802	449	JNZ TREC
0262	CD0307	450	STAI: CALL CTRL0 ;TEST CTRL/0
0265	C26202	451	JNZ STAI
0268	23	452	TREC: INX H
0269	C35402	453	JMP SCRN ;RELUARE
		454	:
		455	-----
		456	:SUBROUTINA BINH
		457	-----
		458	:EXECUTA CONVERSIE BINAR-HEXA
		459	:INTRARI: A=CARACTERUL BINAR
		460	:IESIRI: LA ADRESA HCON 2 OCTETI CE REPREZINTA CAR BINAR CONVERTIT IN HEXA
		461	:
026C	217460	462	BINH: LXI H,HCON
026F	47	463	MOV B,A ;SALVEAZA CARACTER
0270	1F	464	RAR ;PROTESTE DREAPTA CU PATRU POZITII
0271	1F	465	RAR
0272	1F	466	RAR
0273	1F	467	RAR
0274	CD7F02	468	CALL BINI ;CONVERSIE BINAR-HEXA
0277	77	469	MOV M,A ;SALVEAZA IN MEMORIE
0278	23	470	INX H
0279	78	471	MOV A,B ;REFA CARACTER
027A	CD7F02	472	CALL BINI ;CONVERSIE
027D	77	473	MOV M,A ;SALVEAZA IN MEMORIE
027E	C9	474	RET
027F	E60F	475	BINI: ANI 0FH ;PASTREAZA PRIMII PATRU BITI
0281	C630	476	ADI 48 ;CONVERSIE BINAR HEXA
0283	FE3A	477	CPI 58
0285	D8	478	RC
0286	C607	479	ADI 7
0288	C9	480	RET
		481	:
		482	-----
		483	:SUBROUTINA BIND
		484	-----
		485	:EXECUTA CONVERSIE BINAR ASCII ZECIMAL
		486	:INTRARI: A=CARACTERUL BINAR
		487	:IESIRI: LA ADRESA HCON SE AFLA VALOAREA CONVERTITA
		488	:
0289	217460	489	BIND: LXI H,HCON
028C	0664	490	MVI B,100
028E	CD9A02	491	CALL BIDI
0291	060A	492	MVI B,10
0293	CD9A02	493	CALL BIDI

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 10

LOC	OBJ	LINE	SOURCE STATEMENT
0296	0630	494	ADI '0'
0298	77	495	MOV M,A
0299	09	496	RET
029A	362F	497	BIDI: MVI M, '0'-1 ;CONVERSIE VALOARE BINARA
029C	34	498	INR M ;DIN REGISTRUL A IN ZECIMAL
029D	00	499	SUB B
029E	029C02	500	JNC BIDI+2
02A1	80	501	ADD B
02A2	23	502	INX H
02A3	09	503	RET
		504 ;	
		505 ;-----	
		506 ;TABELA DE COMENZI	
		507 ;-----	
		508 ;	
02A4	44554D50	509	CTAB: DB 'DUMP' ;COMANDA DUMP
02A8	0603	510	DW DUMP ;ADRESA LA CARE SE FACE EXECUTIA COMENZII DUMP
02AA	45584543	511	DB 'EXEC'
02AE	FB00	512	DW EXEC
02B0	454E5452	513	DB 'ENTR'
02B4	8004	514	DW ENTR
02B6	46494045	515	DB 'FILE'
02BA	4803	516	DW FILE
02BC	40495354	517	DB 'LIST'
02C0	0A05	518	DW LIST
02C2	44454054	519	DB 'DELT'
02C6	EE05	520	DW DELL
02C8	4153534D	521	DB 'ASBM'
02CC	6506	522	DW ASBM
02CE	50414745	523	DB 'PAGE'
02D2	3C03	524	DW EPAGE
02D4	43555354	525	DB 'CUST'
02D8	0048	526	DW 4800H
02DA	4252454B	527	DB 'BREAK'
02DE	FF0C	528	DW BREAK
02E0	50524F43	529	DB 'PROC'
02E4	AEDD	530	DW PROC
02E6	53544F52	531	DB 'STOR'
02EA	A111	532	DW STAPE
02FC	404F4144	533	DB 'LOAD'
02F0	EF11	534	DW LTAPE
02F2	52455354	535	DB 'REST'
02F6	1900	536	DW INITA
02FB	43404552	537	DB 'CLEAR'
02FC	3500	538	DW ECLER
		539 ;	
		540 ;-----	
		541 ;SUBROUTINA VCHK	
		542 ;-----	
		543 ;VERIFICA PARAMETRII COMENZII	
		544 ;	
02FE	3A7E60	545	VCHK: LBA ABUF ;TEST PRIMUL CARACTER DE LA ADRESA ABUF
0301	E7	546	ORA A ;DACA ESTE 0 INSEAMNA CA NU EXISTA PARAMETRII
0302	CA6404	547	JZ WHAT ;SI SE SEMNALEAZA ERORARE
0305	09	548	RET

SFDX-18 8020/8065 MACRO ASSEMBLER. V3.0 MCMHF PAG 11

LOC	OBJ	LINE	SOURCE STATEMENT
		549	;
		550	;
		551	:COMANDA DUMP
		552	;
		553	:AFISEAZA O ZONA DE MEMORIE
		554	;
0304	CDFE02	555	DUMP: CALL VCHK :VERIFICA EXISTENȚIA PARAMETRIILOR
0309	CDFO00	556	DUMS: CALL CRLF :CAP DE RIND ÎNȚE NOUA
030C	2ABA60	557	DUMI: LHLD BBUF
030F	7E	558	MOV A,M
0310	CDI2402	559	CALL HOTB :AFISEAZA OCTET
0313	CD3D02	560	CALL ACHK :TEST SFIRȘIT AFISARE
0316	D8	561	RC
0317	CDI706	562	CALL CTRL5 :OPREȘTE ȘI CONTINUA AFISAREA CU AJUTORUL
031A	C22903	563	JNZ TREC1 :SUBROUTINELOR CTRL/5 ȘI CTRL/0
031D	CD0307	564	STALL: CALL CTRL0
0320	C21B03	565	JNZ STALL
0323	7D	566	TREC1: MOV A,L
0324	E607	567	ANI 7
0326	C20C03	568	JNZ DUMI :TEST NUMAR CARACTERE AFISATE
0329	C30903	569	JMP DUMS
		570	;
		571	;
		572	:COMANDA PAGE
		573	;
		574	:MUTA O PAGINA DE DATE
		575	;
032C	CDFC02	576	EPAGE: CALL VCHK :VERIFICA PARAMETRII COMENZII
032F	3A8260	577	LDA ABUF+4
0332	E7	578	ORA A
0333	CA6404	579	JZ WHAT :EROARE DACA NU EXISTA AL DOILEA PARAMETRU
0336	2ABA60	580	LHLD BBUF :PREȚIA ADRESELE ZONELOR DE MEMORIE
0339	FB	581	XCHG
033A	2A8C60	582	LHLD BBUF+2
033D	0600	583	MVI B,0 :CONTOR
033F	1A	584	PAGE: LDAX D :TRANSFERA UN OCTET DINTR-O ZONA ÎN ALTA
0340	77	585	MOV M,A
0341	23	586	INX H :INCREMENTARE ADRESE
0342	19	587	INX D
0343	05	588	DCR B
0344	C23F03	589	JNZ PAGE :REȚINARE
0347	C9	590	RET
		591	;
		592	;
		593	:COMANDA FILE
		594	;
		595	:CREEAZA, ASIGNEAZA SAU AFISEAZA INFORMATII DESPRE FISIERE
		596	;
0346	CDFO00	597	FILE: CALL CRLF :LINIE NOUA, CAP DE RIND
034B	3A7660	598	LDA BBUF :TEST EXISTENȚA NUME FISIER
034F	E7	599	ORA A
034F	CA0303	600	JZ FOOT
0352	CD2204	601	CALL FSEA :CAUTA ÎN TABELA DE FISIERE
0355	FB	602	XCHG
0356	C26D03	603	JNZ TEST1 :SALT DACA FISIER GASIT

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 12

LOC	OBJ	LINE	SOURCE STATEMENT
0359	3A7E60	604	LDA ABUF ;EROARE-SE CERE SA SE FACA CURENT
035C	B7	605	ORA A ;UN FISIER NEGASIT
035D	CA6704	606	JZ WHA1
0360	3A7D60	607	LDA FEF
0363	B7	608	ORA A
0364	C28203	609	JNZ ROOM ;SALT DACA SE POATE CREA UN NOU FISIER
0367	217504	610	LXI H,EMES1 ;EROARE-SE CERE SA SE CREEZE UN NOU FISIER
036A	C36A04	611	JMP MESS ;SI NU MAI ESTE LOC
036D	3A7E60	612	TEST1: LDA ABUF
0370	B7	613	ORA A
0371	CA9503	614	JZ SWAPS
0374	2A8A60	615	LHLD BBUF ;SE DORESTE STERGEREA UNUI FISIER
0377	7C	616	MOV A,H ;CARE A FOST GASIT IN TABELA
0378	B5	617	ORA L
0379	CA9503	618	JZ SWAPS
037C	217A04	619	LXI H,EMES2 ;EROARE-SE DORESTE CREEAREA UNUI FISIER EXISTENT
037F	C36A04	620	JMP MESS
0382	2A7E60	621	ROOM: LHLD FREAD ;MEMOREAZA NUME FISIER LA ADRESA
0385	E5	622	XCHG ;DATA LA ADRESA FREAD
0386	217660	623	LXI H,FBUF
0389	D5	624	PUSH D
038A	0E05	625	MVI C,NMLEN
038C	7E	626	MOV23: MOV A,M
038D	12	627	STAX D
038E	13	628	INX D
038F	0D	629	DCR C
0390	23	630	INX H
0391	C28C03	631	JNZ MOV23
0394	D1	632	POP D
0395	212460	633	SWAPS: LXI H,FILE0 ;NUMELE FISIERULUI CURENT SI PARAMETRII SAI
0398	0E0D	634	MVI C,FELEN ;SE SCHIMBA IN TABELE DE FISIERE
039A	1A	635	SWAP: LDAX D ;CU NUMELE FISIERULUI NOU INTRODUS
039B	46	636	MOV B,M ;SI PARAMETRII SAI
039C	77	637	MOV M,A
039D	78	638	MOV A,B
039E	12	639	STAX D
039F	13	640	INX D
03A0	23	641	INX H
03A1	0D	642	DCR C
03A2	C29A03	643	JNZ SWAP
03A5	3A7E60	644	LDA ABUF ;COMANDA NU ARE DREPT PARAMETRII NICI 0
03A8	B7	645	ORA A ;NICI 0 ADRESA
03A9	CACD03	646	JZ FOOT
03AC	2A8A60	647	LHLD BBUF ;ACTUALIZEAZA BOFF SI EOFP
03AF	222960	648	SHLD BOFF
03B2	222E60	649	SHLD EOFP
03B5	7D	650	MOV A,L ;PARAMETRUL A FOST 0 ?
03B6	B4	651	ORA H
03B7	CABCO3	652	JZ FILES
03BA	3601	653	FILE30: MVI M,1 ;MARCHEAZA SFIRSIT DE FISIER
03BC	AF	654	FILES: XRA A
03BD	322D60	655	STA MAXL ;INITIALIZEAZA MAXL
03C0	C3CD03	656	JMP FOOT
03C3	3A1E61	657	FOOT: LDA IBUF+4 ;COMANDA FILES ?
03C6	FE53	658	CPI 'S' ;DACA DA, C-MAX-IL

FDX-10 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 13

LOC	OBJ	LINE	SOURCE STATEMENT
03C8	0E06	659	MVI C,MAXFIL ;DACA NU, C=0
03CA	CACF03	660	JZ FOUL
03CD	0E01	661	FOUL: MVI C,1
03CF	212460	662	FOUL: LXI H,FILE0 ;SALVEAZA CONTOR
03D2	79	663	MOV A,C
03D3	327D60	664	FINE: STA FOCNT
03D6	E5	665	PUSH H
03D7	110500	666	LXI D,NMLEN ; (H,L)=BOFP
03DA	19	667	DAD D
03DB	7E	668	MOV A,M ;TEST BOFP
03DC	E7	669	ORA A ;BOFP DIFERIT DE ZERO, SALT LA FOOD
03DD	C2ED03	670	JNZ FOOD
03E0	23	671	INX H
03E1	86	672	ADD M
03E2	23	673	INX H
03E3	C2ED03	674	JNZ FOOD
03E6	33	675	INX SP ;ACTUALIZARE
03E7	33	676	INX SP
03E8	23	677	INX H ;(H,L)=MAXL
03E9	23	678	INX H
03EA	C30204	679	JMP FEET
03ED	E1	680	FOOD: POP H ;AFISEAZA NUME FISIER
03EE	0E05	681	MVI C,NMLEN
03F0	46	682	FAST: MOV B,M
03F1	CDE700	683	CALL OUTS
03F4	0D	684	DCR C
03F5	23	685	INX H
03F6	C2F003	686	JNZ FAST
03F9	CD0E04	687	CALL FOOL ;AFISEAZA PARAMETRII FISIERULUI
03FC	CD0E04	688	CALL FOOL
03FF	CDFO00	689	CALL CRLF ;CAP DE RIND LINIE NOUA
0402	110400	690	FEET: LXI D,FELEN-NMLEN-4 ;CAUTA ADRESA URMATORULUI NUME
0405	19	691	DAD D ;DE FISIER DIN TABELA
0406	3A7D60	692	LDA FOCNT
0409	3D	693	DCR A
040A	C2D303	694	JNZ FINE
040D	C9	695	RET
040E	CD3702	696	;SCRIE NUMAR
0411	23	697	FOOL: CALL BLK1 ;AFISEAZA BLANC
0412	7E	698	INX H ;AFISEAZA OCTETUL SUPERIOR
0413	7E	699	MOV A,M
0413	2B	700	DCX H
0414	E5	701	PUSH H
0415	CD1402	702	CALL HOUT
0418	E1	703	POP H
0419	7E	704	MOV A,M
041A	23	705	INX H ;AFISEAZA OCTETUL INFERIOR
041B	23	706	INX H ;SI INCREMENTEAZA H,L CU 2
041C	E5	707	PUSH H
041D	CD2402	708	CALL HOTB
0420	E1	709	POP H
0421	C9	710	RET
		711	;
		712	;-----
		713	;SUBROUTINA FSEA

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 14

LOC	OBJ	LINE	SOURCE STATEMENT
		714	;-----
		715	;CAUTA IN TABELA DE FISIERE FISIERUL DAT IN BBUF
		716	;IESIRI: Z=0 FISIER GASIT, IN H,L ADRESA DIN TABELA UNDE SE AFLA FISIERUL
		717	; Z=1 FISIER NEGASIT
		718	; FEF=0 TABELA PLINA
		719	; FEF DIFERIT DE ZERO , FREAD TINE ADRESA
		720	; UNDE POATE FI PLASAT FISIERUL IN TABELA
		721	;
0422	AF	722	FSEA: XRA A
0423	327D60	723	STA FEF
0426	0606	724	MVI B,MAXFIL
0428	112460	725	LXI D,FILEO ;CAUTA FISIERUL CU NUMELE DAT
042B	217660	726	FSE10: LXI H,FBUF ;IN BUFFERUL FBUF, IN TABELA DE FISIERE
042E	0E05	727	MVI C,NMLEN
0430	CD2D01	728	CALL SEAR
0433	F5	729	PUSH PSW
0434	D5	730	PUSH D
0435	1A	731	LDAX D ;SALT LA FSE20 DACA BOFF DIFERIT DE ZERO
0436	B7	732	ORA A ;ADICA DACA EXISTA FISIERE IN ZONA
0437	C25804	733	JNZ FSE20
043A	13	734	INX D
043B	1A	735	LDAX D
043C	B7	736	ORA A
043D	C25804	737	JNZ FSE20
0440	EB	738	XCHG
0441	11FAFF	739	LXI D,-NMLEN-1 ;FREAD TINE ADR PT O ZONA DE FIS LIBERA
0444	19	740	DAD D
0445	227B60	741	SHLD FREAD
0448	7A	742	MOV A,D ;(FEF) DIFERIT DE ZERO
0449	327D60	743	STA FEF
044C	E1	744	POP H
044D	F1	745	POP PSW
044E	110800	746	FSE15: LXI D,FELEN-NMLEN ;ADRESA URMATOAREI ZONE FISIER
0451	19	747	DAD D
0452	EB	748	XCHG
0453	05	749	DCR B ;TESI SFIRSIT CAUTARE
0454	C8	750	RZ
0455	C32B04	751	JMP FSE10 ;RELUARE
0458	E1	752	FSE20: POP H
0459	F1	753	POP PSW
045A	C24E04	754	JNZ FSE15 ;FISIER NEGASIT
045D	11FBFF	755	LXI D,-NMLEN ;(H,L)=ADRESA FISIER GASIT
0460	19	756	DAD D
0461	7A	757	MOV A,D
0462	B7	758	ORA A ;Z=0
0463	C9	759	RET
		760	;MESAJ EROARE
0464	CDFO00	761	WHAT: CALL CRLF ;CAP DE RIND , LINIE NOUA
0467	217004	762	WHAT: LXI H,EMES ;ADRESA MESAJ
046A	DD5402	763	MESS: CALL SCRN ;SORIE MESAJ
046D	C35900	764	JMP EOR ;SALT IN MONITOR CU INITIALIZARE SP
0470	57484134	765	EMES: DB 'WHAT',13 ;MESAJE DE EROARE
0474	0D		
0475	46554C4C	766	EMES1: DB 'EULL',13
0479	0D		

SFDX-18 8090/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 15

LOC	OBJ	LINE	SOURCE STATEMENT
047A	4E4F204E	767	EMES2: DB 'NO NO .13
047E	4F		
047F	0D		
		768	;
		769	;
		770	;COMANDA ENTR
		771	;
		772	;INTRODUCE DATE IN MEMORIE
		773	;
0480	CDFE02	774	ENTR: CALL VCHK ;VERIFICA EXISTENTA PARAMETRILOR
0483	CD8D04	775	CALL ENTS
0486	DA6404	776	JC WHAT ;SEMNALIZEAZA ERGARE
0489	CDF000	777	CALL CRLF ;CAP DE RIND LINIE NOUA
049C	C9	778	RET
002F		779	EEND EQU ;
048D	CDF000	780	ENTG: CALL CRLF ;CAP DE RIND , LINIE NOUA
0490	CD7700	781	CALL READ ;CITESTE O LINIE
0493	211A61	782	LXI H,IBUF
0496	229660	783	SHI D PNTR
0499	CD4001	784	ENT1: CALL ZBUF ;INITIALIZARE ABUF
049C	CD3A09	785	CALL SRLK ;SALT PESTE BLANCURI
049F	DA8D04	786	JC ENTS ;RELUARE DACA CR
04A2	FE2F	787	CPI EEND ;SFIRSITUL INTRODUCERII
04A4	C8	788	RZ
04A5	CD420B	789	CALL ALPS ;PREIA CARACTERE
04A8	78	790	MOV A,B ;TEST LUNGIME SIR CARACTERE
04A9	FE03	791	CPI 3
04AB	3F	792	CMC
04AC	D8	793	RC
04AD	017E60	794	LXI B,ABUF ;CONVERSIE IN BINAR
04B0	CDF501	795	CALL AHEX
04B3	D8	796	RC ;ERGARE
04B4	7D	797	MOV A,L ;CARACTER BINAR IN A
04B5	2ABA60	798	LHLD BBUF ;PUNE CARACTER IN MEMORIE
04B8	77	799	MOV M,A
04B9	CD4F02	800	CALL ACHI ;TEST SFIRSIT COMANDA
04BC	C39904	801	JMP ENT1 ;RELUARE
		802	;
		803	;
		804	;EDITORUL
		805	;
		806	;
04BF	3A2460	807	LINE: LDA FILE0 ;EXISTA NUME DE FISIER CURENT ?
04C2	B7	808	ORA A
04C3	CA6404	809	JZ WHAT
04C6	0E04	810	MVI C,4 ;TEST NUMAR DE LINIE CORECT
04C8	211961	811	LXI H,IBUF-1 ;SEMNALIZEAZA ORICE ERGARE
04CB	23	812	LICK: INX H
04CC	7E	813	MOV A,M
04CD	FE30	814	CPI '0'
04CF	DA6404	815	JC WHAT
04D2	FE3A	816	CPI '9'+1
04D4	D26404	817	JNC WHAT
04D7	0D	818	DCR C
04D8	C2CB04	819	JNZ LICK

8FDX-18 8080/8085 MACRO ASSEMBLER, V3.0 MODULE PAGE 16

LOC	OBJ	LINE	SOURCE STATEMENT
04DB	227460	820	SHLD ADDS
04DE	113060	821	LXI D,MAXL+3 ;COMPARA NUMARUL LINIEI CURENTE
04E1	CDAC05	822	CALL COMO ;CU NUMARUL ULTIMEI LINII DIN FISIER
04E4	D20405	823	JNC INSR ;SALT DACA LINIA TREBUIE INSERATA
04E7	23	824	INX H
04E8	CD9C05	825	CALL LODM
04EB	213060	826	LXI H,MAXL+3 ;NUMAR LINIE NOUA LA ADRESA MAXL
04EE	CDA405	827	CALL STOM
04F1	111961	828	LXI D,IBUF-1 ;PREIA LINIA NOU INTRODUSA
04F4	2A2B60	829	LHLD EOFP ;LA SFIRSITUL FISIERULUI
04F7	0E01	830	MVI C,1 ;SI ACTUALIZEAZA SP
04F9	CD8A05	831	CALL LMOV
04FC	3601	832	SEOF: MVI M,1
04FE	222B60	833	SHLD EOFP
0501	C35900	834	JMP EOR
0504	CD5C05	835	INSR: CALL FIN1 ;CAUTA LOCUL LINIEI IN FISIER
0507	0E02	836	MVI C,2
0509	CA0D05	837	JZ EQU
050C	0D	838	DCR C
050D	46	839	EQUL: MOV B,M ;NR LINIE NOUA DIFERIT DE NR LINIE VECH
050E	2B	840	DCX H ;NR CARACTERE DIN LINIA VECH IN B
050F	3602	841	MVI M,2 ;INDICATOR DE LINIE
0511	227260	842	SHLD INSP
0514	3A1961	843	LDA IBUF-1 ;NR CARACTERE DIN LINIA NOUA IN A
0517	0D	844	DCR C
0518	CA2205	845	JZ ELT ;NUMERELE CELOR DOUA LINII SINT DIFERITE
051B	90	846	SUB B
051C	CA4505	847	JZ ZERO ;LINIILE AU ACELASI NUMAR
051F	DA3505	848	JC EOT ;SI ACELASI NUMAR DE CARACTERE
0522	2A2B60	849	ELT: LHLD EOFP ;NR CARACTERE DIN LINIA NOUA > NR CAR DIN LINIA VECH
0525	54	850	MOV D,H ;SAU NUMERELE CELOR DOUA LINII NU SINT EGAL
0526	5D	851	MOV E,L
0527	CD8505	852	CALL ADR
052A	222B60	853	SHLD EOFP
052D	0E02	854	MVI C,2
052F	CD9305	855	CALL RMOV ;DEPLASEAZA ULTIMA PARTE DIN FISIER
0532	C34505	856	JMP ZERO
0535	2F	857	EGT: CMA
0536	3C	858	INR A ;NUMERELE CELOR DOUA LINII SINT EGAL
0537	54	859	MOV D,H ;NUMARUL DE CARACTERE DIN LINIA NOUA < NR CARACTERE
0538	5D	860	MOV E,L ;DIN LINIA VECH
0539	CD8505	861	CALL ADR ;SE STERG CARACTERELE IN PLUS
053C	EB	862	XCHG
053D	CD8A05	863	CALL LMOV
0540	3601	864	MVI M,1
0542	222B60	865	SHLD EOFP
0545	2A7260	866	ZERO: LHLD INSP ;PUNE LINIA NOUA IN FISIER SI MARCHEAZA
0548	360D	867	MVI M,ASCR ;SFIRSITUL DE FISIER
054A	23	868	INX H
054B	111961	869	LXI D,IBUF-1
054E	0E01	870	MVI C,1
0550	CD8A05	871	CALL LMOV
0553	C35900	872	JMP EOR
		873	;
		874	;

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 17

LOC	OBJ	LINE	SOURCE STATEMENT
		875	;SUBROUTINA FIND
		876	-----
		877	;CAUTA LINIE IN FIS CURENT
		878	;
0556	218160	879	FIND: LXI H,ABUF+3
0559	227460	880	SHLD ADDS
055C	2A2960	881	FINI: LHLD BOFF ;SALT IN MONITOR DACA BOFF=0
055F	7C	882	MOV A,H ;ADICA FISIER CURENT INEXISTENT
0560	B5	883	ORA L
0561	CA5900	884	JZ EOR
0564	CD7E05	885	FII: CALL E01 ;TEST SFIRSIT FISIER
0567	EB	886	XCHG
0568	2A7460	887	LHLD ADDS ;(H,L)=ADRESA ULTIMULUI OCTET
056B	EB	888	XCHG ;DIN NUMARUL LINIEI CURENTE
056C	3E04	889	MVI A,4
056E	CD8505	890	CALL ADR
0571	CDAC05	891	CALL COMO ;COMPARA NUMARUL LINIEI NOU INTRODUSE
0574	D8	892	RC ;CU NUMARUL LINIEI CURENTE DIN FISIER
0575	C8	893	RZ
0576	7E	894	FII: MOV A,M ;SE TRECE LA O NOUA LINIE IN FISIER
0577	CD8505	895	CALL ADR
057A	C36405	896	JMP FII
		897	;CAUTA EOF
057D	23	898	EOF: INX H ;TESTEAZA SFIRSIT DE FISIER
057E	3E01	899	E01: MVI A,1 ;SI SALT LA EOR DACA S-A GASIT
0580	BE	900	CMP M
0581	C0	901	RNZ
0582	C35900	902	JMP EOR
		903	;ADD O VALOARE LA H SI L
0585	85	904	ADR: ADD L ;ADUNA A LA H,L
0586	6F	905	MOV L,A
0587	D0	906	RNC
0588	24	907	INR H
0589	C9	908	RET
		909	;
		910	-----
		911	;SUBROUTINA LMOV
		912	;
		913	INTRARI: D,E=ADRESA ZONEI CARE SE TRANSFERA
		914	; H,L=ADRESA ZONEI UNDE SE TRANSFERA
		915	; C=CARACTER PINA LA CARE SE TRANSFERA
		916	IESIRI: D,E SI H,L IAU VALDAREA ADRESEI DE DUPEA
		917	; ULTIMUL OCTET TRANSFERAT
		918	;MUTA UN SIR DE CAR
		919	;
058A	1A	920	LMOV: LDAX D ;TRANSFERA O ZONA DE MEMORIE IN ALTA ZONA
058B	13	921	INX D
058C	B9	922	CMP C ;IN C ESTE CARACTERUL DE STOP
058D	C8	923	RZ
058E	77	924	MOV M,A
058F	23	925	INX H
0590	C38A05	926	JMP LMOV
0593	1A	927	RMOV: LDAX D ;ACELASI LUCRU CA LMOV
0594	1B	928	DCX D ;NUMAI CA TRANSFERUL SE FACE PRIN DECREMENTARE CONT
0595	B9	929	CMP C

SFDX-18 8080/8085 MACRO ASSEMBLER. V3.0

MODULE PAGE 18

LOC	OBJ	LINE	SOURCE STATEMENT
0596	C8	930	RZ
0597	77	931	MOV M,A
0598	2B	932	DCX H
0599	C39305	933	JMP RMOV
		934	; INCARCA 4 CAR DIN MEM IN REGISTRE
059C	46	935	LDM: MOV B,M
059D	23	936	INX H
059E	4E	937	MOV C,M
059F	23	938	INX H
05A0	56	939	MOV D,M
05A1	23	940	INX H
05A2	5E	941	MOV E,M
05A3	C9	942	RET
		943	; PUNE 4 CAR DIN REGISTRE IN MEM
05A4	73	944	STOM: MOV M,E
05A5	2B	945	DCX H
05A6	72	946	MOV M,D
05A7	2B	947	DCX H
05A8	71	948	MOV M,C
05A9	2B	949	DCX H
05AA	70	950	MOV M,B
05AB	C9	951	RET
		952	;
		953	;-----
		954	;SUBROUTINA COMO
		955	;-----
		956	;COMPARA 2 SIRURI DE 4 CAR
		957	;INTRARI: D,E SI H,L CONTIN ADRESELE UNDE SE AFLA CELE DOUA SIRURI
		958	;IESIRI: Z=1 SIRURI EGALE
		959	; CY=0 SIRUL INDICAT DE D,E ARE O VALOARE MAI MARE
		960	; SAU EGALA CU SIRUL INDICAT DE H,L
		961	;MODIFICA: B,C
		962	;
05AC	0601	963	COMO: MVI B,1
05AE	0E04	964	MVI C,4
05B0	B7	965	ORA A
05B1	1A	966	CO1: LDAX D
05B2	9E	967	SBB H
05B3	CAB705	968	JZ CO2
05B4	04	969	INR B
05B7	1B	970	CO2: DCX D
05B8	2B	971	DCX H
05B9	0D	972	DCR C
05BA	C2B105	973	JNZ CO1
05BD	05	974	DCR B
05BE	C9	975	RET
05BF	0E04	976	COM1: MVI C,4
05C1	1A	977	LDAX D
05C2	D601	978	SUI 1
05C4	C3B205	979	JMP CO1+1
		980	;NORMALIZEAZA
05C7	CD9C05	981	NORM: CALL LDM
05CA	AF	982	XRA A
05CB	B8	983	CMP B
05CC	C8	984	RZ

BFDX-48 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 19

LOC	OBJ	LINE	SOURCE STATEMENT
05CD	BB	985	NOR1: CMP E
05CE	C4A405	986	CNZ STOM
05D1	C0	987	RNZ
05D2	5A	988	MOV E,D
05D3	51	989	MOV D,C
05D4	48	990	MOV C,B
05D5	0630	991	MVI B,'0'
05D7	C3CD05	992	JMP NOR1
		993	;
		994	;
		995	;COMANDA LIST
		996	;
		997	;AFISEAZA LINII
		998	;
05DA	CDF000	999	LIST: CALL CRLF ;LINIE NOUA , CAP DE RIND
05DB	CD5605	1000	CALL FIND ;CAUTA LINIA INDICATA
05E0	23	1001	LISTO: INX H
05E1	CD5402	1002	CALL SCRNL ;AFISEAZA LINIA
05E4	CDF000	1003	CALL CRLF ;LINIE NOUA
05E7	CD7D05	1004	CALL EOF ;TEST SFIRSIT FISIER
05EA	C2E005	1005	JNZ LISTO ;RELUARE
05ED	C9	1006	RET
		1007	;
		1008	;
		1009	;COMANDA DELETE
		1010	;
		1011	;STERGE LINII DIN FISIER
		1012	;
05EE	CDFE02	1013	DELL: CALL VCHK ;VERIFICA EXISTENTA PARAMETRILOP
05F1	CD5605	1014	CALL FIND ;GASESTE LINIA DE STERS
05F4	227260	1015	SHLD DELP
05F7	218560	1016	LXI H,ABUF+7 ;TEST PARAMETRU 2
05FA	7E	1017	MOV A,M
05FB	B7	1018	ORA A
05FC	C20206	1019	JNZ DEL1
05FF	218140	1020	LXI H,ABUF+3
0602	227460	1021	DEL1: SHLD ADDS
0605	EB	1022	XCHG
0606	218060	1023	LXI H,MAXL+3 ;COMPARA PRIMUL PARAMETRU
0609	8DACC5	1024	CALL COMO ;CU NUMARUL ULTIMEI LINII DIN FISIER
060C	2A7260	1025	LHLD DELP
060F	DA5806	1026	JC NOVR ;SALT DACA ZONA DE STERS ESTE IN INTERIORUL FISIERULUI
0612	222B60	1027	SHLD EOFP ;MARCHEAZA SFIRSIT DE FISIER
0615	3601	1028	MVI M,1
0617	EB	1029	XCHG
0618	2A2960	1030	LHLD BOFP
0618	EB	1031	XCHG
061C	060D	1032	MVI B,13
061E	2B	1033	DCX H
064F	7D	1034	DEL2: MOV A,L
0620	93	1035	SUB E
0621	7C	1036	MOV A,H
0622	9A	1037	SBB D
0623	3E0D	1038	MVI A,ASCR
0625	DA4706	1039	JC DFL4

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 20

LOC	OBJ	LINE	SOURCE STATEMENT
0628	05	1040	DCR B
0629	2B	1041	DCX H
062A	BE	1042	CMP M
062B	C21F06	1043	JNZ DEL2
062E	2B	1044	DCX H
062F	7D	1045	MOV A,L
0630	93	1046	SUB E
0631	7C	1047	MOV A,H
0632	9A	1048	SBB D
0633	DA4806	1049	JC DEL5
0636	BE	1050	CMP M ;SFIRSIT PROVIZORIU DE LINIE
0637	23	1051	INX H
0638	23	1052	INX H
0639	CA3D06	1053	JZ DEL3
063C	23	1054	INX H
063D	CD9C05	1055 DEL3:	CALL LODM ;INCARCA NOUA VALDARE PENTRU MAXL
0640	213060	1056	LXI H,MAXL+3
0643	CDA405	1057	CALL STOM
0646	C9	1058	RET
0647	B8	1059 DEL4:	CMP B
0648	EB	1060 DEL5:	XCHG
0649	C23C06	1061	JNZ DEL3-1
064C	322D60	1062	STA MAXL ;PUNE UN NUMAR MIC IN MAXL
064F	C9	1063	RET
0650	CD6405	1064 NOVR:	CALL FI1 ;GASESTE SFIRSITUL ZONEI DE STERS
0653	CC7605	1065	CZ FI2
0656	EB	1066 NOV1:	XCHG
0657	2A7260	1067	LHLD DELP
065A	0E01	1068	MVI C,1 ;TERMINATOR
065C	CD8A05	1069	CALL LMOV ;COMPACTEAZA FISIERUL
065F	222B60	1070	SHLD EOFP ;ACTUALIZEAZA EOFP
0662	3601	1071	MVI M,1 ;EOF
0664	C9	1072	RET
		1073	;
		1074	;-----
		1075	;ASAMBLOLUL
		1076	;-----
		1077	;
0665	CDFE02	1078 ASSM:	CALL VCHK ;SALT LA EOR DACA CDA NU ARE PARAMETRII
0668	3A8260	1079	LDA ABUF+4 ;SALT LA ASM4 DACA EXISTA SI PARAMETRUL 2
066B	B7	1080	GRA A
066C	C27506	1081	JNZ ASM4
066F	2ABA60	1082	LHLD BBUF ;PARAMETRUL 2 PRIMESTE VALOAREA PARAMETRUL 1
0672	228C60	1083	SHLD BBUF+2
0675	3A1E61	1084 ASM4:	LDA IBUF+4 ;LOCATIA AERR=0 PENTRU COMANDA ASSME
0678	D645	1085	SUI 'E' ;SI DIFERIT DE 0 PENTRU COMANDA ASSM
067A	329E60	1086	STA AERR
067D	AF	1087	XRA A ;INITIALIZARE CU ZERO CONTOR DE ETICHETE
067E	329860	1088	STA NOLA
0681	329460	1089 ASM3:	STA PASI ;LOCATIA PASI=0 LA PAS 1 SI DIFERIT DE 0 LA PAS 2
0684	CDF000	1090	CALL CRLF ;CAP DE RIND, LINIE NOUA
0687	2ASA60	1091	LHLD BBUF ;LOCATIA ASPB CONTINE PC-UL ASAMBLARII
068A	229260	1092	SHLD ASPC ;SI ESTE INITIALIZAT CU VALDAREA PARAMETRULUI 1
068D	2A2960	1093	LHLD BOFP ;LOCATIA APNT PUNCTEAZA LINIA CURENTA DE ASAMELAT
0690	227260	1094	SHLD APNT ;DIN FISIER SI ESTE INITIALIZATA CU ADRESA DE INCEPUT

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 21

LOC	OBJ	LINE	SOURCE STATEMENT
0693	2A7260	1095	ASM1: LHL D APNT ;DE FISIER AFLATA LA BOFF
0696	310461	1096	LXI SP,AREA+100 ;INITIALIZARE SP
0699	7E	1097	MOV A,M
069A	FE01	1098	CPI 1
069C	CA2E09	1099	JZ EASS ;SALT LA EASS LA SFIRSIT DE FISIER
069F	EB	1100	XCHG ;IN D,E ADRESA DE LINIE CURENTA
06A0	13	1101	INX D
06A1	210561	1102	LXI H,OBUF ;OBUF ESTE INITIALIZAT CU BLANCURI
06A4	3E15	1103	MVI A,IBUF-5 AND 0FH
06A6	CDD800	1104	CALL CLER
06A9	0E0D	1105	MVI C,ASCR ;RUTINA LMOV DEPUNE IN IBUF SI IN CEI 5 OCTETI
06AB	CD8A05	1106	CALL LMOV ;DINAINTEA LUI LINIA CURENTA DE PRELUCRAT * CR
06AE	71	1107	MOV M,C
06AF	EB	1108	XCHG ;IN H,L ADRESA DE LINIE CURENTA SALVATA LA APNT
06B0	227260	1109	SHLD APNT
06B3	3A9460	1110	LDA PAS1 ;SALT LA ASM2 LA PAS 2
06B6	B7	1111	DRA A
06B7	C2C006	1112	JNZ ASM2
06BA	CD0907	1113	CALL PAS1 ;PAS 1 DE ASAMBLARE
06BD	C39306	1114	JMP ASM1
06C0	CD0007	1115	ASM2: CALL PAS2 ;PAS 2 DE ASAMBLARE
06C3	210561	1116	LXI H,OBUF ;AFISEAZA LINIA CURENTA ASAMBLATA
06C6	CDCC06	1117	CALL ACUT
06C9	C39306	1118	JMP ASM1
		1119	;SCRIE LISTING DE ASAMBLARE
06CC	3A8E60	1120	ADUT: LDA AERR
06CF	B7	1121	ORA A
06D0	C2D906	1122	JNZ AOU1 ;SALT LA AOU1 PENTRU COMANDA ASSM
06D3	3A0561	1123	AOU2: LDA OBUF ;RETURN DACA NU EXISTA EROARE DE ASAMBLARE
06D6	FE20	1124	CPI "
06D8	C8	1125	RZ
06D9	210561	1126	AOU1: LXI H,OBUF ;SCRIE OBUF PE O LINIE
06DC	AF	1127	XRA A
06DD	46	1128	EAF: MOV B,M
06DE	CDE700	1129	CALL OUT8
06E1	23	1130	INX H
06E2	3C	1131	INR A
06E3	FE10	1132	CPI 16
06E5	C2DD06	1133	JNZ EAF
06E8	CDF000	1134	CALL CRLF ;CAP DE RIND, LINIE NOUA
06EB	0E09	1135	MVI C,TAB ;SCRIE TAB
06ED	CD7A10	1136	CALL AFIS
06F0	CD5402	1137	CALL SCRN ;SCRIE IN CONTINUARE PINA LA CR
06F3	CDF000	1138	CALL CRLF ;CAP DE RIND, LINIE NOUA
06F6	C9	1139	RET
06F7	3EC4	1140	CTRLS: MVI A,0C4H
06F9	322200	1141	COM9: STA PORTC
06FC	3A2000	1142	LDA PORTA
06FF	2F	1143	CMA
0700	D602	1144	SUI 2
0702	C9	1145	RET
0703	3EC2	1146	CTRLQ: MVI A,0C2H
0705	CDF906	1147	CALL COM9
0708	C9	1148	RET
		1149	;PAS 1 DE ASAMBLARE

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 22

LOC	OBJ	LINE	SOURCE STATEMENT
0709	CD4001	1150	PAS1: CALL ZBUF ;INITIALIZEAZA ABUF CU ZERURI
070C	329460	1151	STA PASI ;PASI=0
070F	211A61	1152	LXI H,IBUF ;LOCATIA PNTR CONTINE POINTERUL DE CITIRE DIN IBUF
0712	229660	1153	SHLD PNTR
0715	7E	1154	MOV A,M ;DACA PRIMUL CARACTER DIN LINIE = BLANC,
0716	FE20	1155	CPI ;ATUNCI SE TRECE DIRECT LA PRELUCRARE COD
0718	CA4B07	1156	JZ OPC
0718	FE2A	1157	CPI '*' ;DACA PRIMUL CARACTER DIN LINIE = *,
071D	C8	1158	RZ ;ATUNCI LINIA ESTE DE COMENTARIU SI NU SE ASAMBLEAZA
071E	CD4D0B	1159	CALL SLAB ;AICI INCEPE PRELUCRAREA ETICHETELOR, PRIN RUTINA SLAB
0721	DA0C0B	1160	JC ;CY=1 PENTRU EROARE IN ETICHETA
0724	CAF40C	1161	JZ ERRD ;Z=1 PENTRU ETICHETA MULTIPLU DEFINITA
0727	CD6207	1162	CALL LCHK ;VERIFICA CARACTER DUPA ETICHETA SI LA Z=0 EROARE
072A	C20C0B	1163	JNZ OP5
072D	0E05	1164	MVI C,LLAB ;LLAB=5 LUNGIME ETICHETA
072F	217E60	1165	LXI H,ABUF
0732	7E	1166	MOV A,M ;SE DEPUNE ETICHETA IN TABELA DE SIMBOLI :
0733	12	1167	STAX D ;5 OCTETI CE SPECIFICA NUMELE, URMATI DE 2 OCTETI
0734	13	1168	INX D ;CE INDICA VALOAREA SIMBOLULUI
0735	23	1169	INX H
0736	0D	1170	DCR C
0737	C23207	1171	JNZ MLAB
073A	EB	1172	XCHG ;ADRESA ULTIMEI VALORI DE ETICHETA SE SALVEAZA
073B	229060	1173	SHLD TABA ;LA TABA
073E	3A9360	1174	LDA ASPC+1 ;SE DEPUNE VALOAREA ETICHETEI IN TABELA
0741	77	1175	MOV M,A
0742	23	1176	INX H
0743	8A9260	1177	LDA ASPC
0746	77	1178	MOV M,A
0747	219860	1179	LXI H,NOLA ;SE INCREMENTEAZA CONTORUL DE ETICHETE
074A	34	1180	INR M ;INCEPE PRELUCRARE COD INSTRUCIUNE
074B	CD4001	1181	OPC: CALL ZBUF ;SBLK POZITIONEAZA PNTR PE PRIMUL CARACTER DIFERIT DE
074E	CD3A09	1182	CALL SBLK ;BLANC DIN IBUF
0751	DA330B	1183	JC OERR ;CY=1 DACA CARACTER ESTE CR
0754	CDA20B	1184	CALL ALPS ;ALPS DEPUNE COD IN ABUF SI LASA IN A CAR DUPA COD
0757	FE20	1185	CPI
0759	DA920A	1186	JC OPCD ;SALT DACA CARACTERUL DUPA COD ESTE CR
075C	C2330B	1187	JNZ OERR ;SALT LA EROARE , ADICA CARACTER DIFERIT DE BLANC
075F	C3920A	1188	JMP OPCD ;SALT LA OPCOD , UNDE SE PRELUCREAZA CODUL
0762	2A9660	1189	CAUTA BLANC SAU ; DUPA ETICHETA
0765	7E	1190	LCHK: LHLD PNTR ;SE CITESTE CARACTER DUPA ETICHETA
0766	FE20	1191	MOV A,M
0768	C8	1192	CPI
0768	C8	1193	RZ ;RETURN CU Z=1 LA BLANC
0769	FE3A	1194	CPI '*'
076B	C0	1195	RNZ ;RETURN CU Z=0 DACA NU ESTE BLANC SAU '*'
076C	23	1196	INX H
076D	229660	1197	SHLD PNTR ;REFACERE POINTER
0770	C9	1198	RET
0771	CD3A09	1199	PRELUCREAZA PSEUDO INSTR LA PAS 1
0774	1A	1200	PSU1: CALL SBLK ;SALT PESTE BLANCURI
0775	B7	1201	LDAX D ;IN A NR CE CORESPONDE PSEUDO-INSTR IDENTIFICATE
0776	CAB007	1202	ORA A
0776	CAB007	1203	JZ ORD1 ;DECIDE PSEUDO-INSTR SI SALT LA SECVENTA CORESPUNZATOARE
0779	FABD07	1204	JM DAT1

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 23

LOC	OBJ	LINE	SOURCE STATEMENT
077C	E2A207	1205	JPO EQU1
077F	FE05	1206	CPI 5
0781	DAB507	1207	JC RES1
0784	C22E09	1208	JNZ EASS ;SALT LA EASS LA END
0787	0E02	1209	AC01: MVI C,2 ;DW - INCREMENTEAZA ASPC CU 2
0789	AF	1210	XRA A
078A	C3220B	1211	JMP OCN1
078D	CDC40B	1212	ORG1: CALL ASCN ;ORG
0790	3A0561	1213	LDA OBUF ;REINIT ASPC SI DEPUNE EVENTUALA ETICHETA IN TAB SIMB
0793	FE20	1214	CPI ' ' ;ASCN PRODUCE IN H,L VALOAREA ORIGINII
0795	C0	1215	RNZ
0796	229260	1216	SHLD ASPC
0799	3A1A61	1217	LDA IBUF
079C	FE20	1218	CPI ' ' "
079E	C8	1219	RZ
079F	C3AD07	1220	JMP EQU5
07A2	CDC40B	1221	EQU1: CALL ASCN ;EQU
07A5	3A1A61	1222	LDA IBUF ;DEPUNE VALOAREA ETICHETEI CALCULATA DE ASCN
07A8	FE20	1223	CPI ' ' ;IN TABELA DE SIMBOLI
07AA	CACCC0C	1224	JZ ERRM
07AD	EB	1225	EQU5: XCHG
07AE	2A9060	1226	LHLD TABA ;TABA INDICA PRIMUL OCTET DIN VALOAREA
07B1	72	1227	MOV M,D ;ULTIMULUI SIMBOL IIN TABELA
07B2	23	1228	INX H
07B3	73	1229	MOV M,E
07B4	C9	1230	RET
07B5	CDC40B	1231	RES1: CALL ASCN ;DS
07B8	44	1232	MOV B,H ;INCREMENTEAZA ASPC CU VALOAREA DATA DE ASCN IN H,L
07B9	4D	1233	MOV C,L
07BA	C31A08	1234	JMP RES21
07BD	C32108	1235	DAT1: JMP DAT2A ;DB
		1236	;PAS 2 AL ASAMBLARII
07C0	210761	1237	PAS2: LXI H, OBUF+2 ;DEPUNE IN OBUF VAL DIN ASPC IN FORMAT HEXA
07C3	3A9360	1238	LDA ASPC+1 ;PRIMELE 2 CAR DIN OBUF SINT LASATE PT COD EROARE
07C6	CD6F02	1239	CALL BINH+3 ;EXECUTA CONVERSIA BINAR - HEXA
07C9	23	1240	INX H
07CA	3A9260	1241	LDA ASPC
07CD	CD6F02	1242	CALL BINH+3
07D0	23	1243	INX H
07D1	229E60	1244	SHLD OIND ;CONTINE POINTERUL DE SCRIERE IN OBUF
07D4	CD4001	1245	CALL ZBUF ;ABUF(12) IA VALOAREA 0
07D7	211A61	1246	LXI H, IBUF
07DA	229660	1247	PABL: SHLD PNTR ;POZITIONEAZA POINTER CE CITIRE PNTR DIN IBUF
07DD	7E	1248	MOV A,M
07DE	FE20	1249	CPI ' ' ;CARACTER = BLANC , SALT LA PRELUCRARE COD
07E0	CA4B07	1250	JZ OPC
07E3	FE2A	1251	CPI '*' ;CARACTER = * , RETURN CACI ESTE COMENTARIU
07E5	C8	1252	RZ
07E6	CD4D0B	1253	CALL SLAB ;PRELUCRARE ETICHETE
07E9	DAB50C	1254	JC BRNL ;CY=1 EROARE IN ETICHETE
07EC	CD4D07	1255	CALL LOKK ;CARACTER DUPA ETICHETA
07EF	C2EFC0C	1256	JNZ BRNL ;Z=0 EROARE LA DIFERIT DE BLANC SAU ' ' "
07F2	634B07	1257	JMP OPC ;SALT LA PRELUCRARE COD
		1258	::PRELUCREAZA PSEUDO-INSTR PT PAS 2
07F5	1A	1259	PSU2: LBAX D ;IN A COD PSEUDO-INSTR IDENTIFICATA

SFDX-16 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 24

LOC	OBJ	LINE	SOURCE STATEMENT
07F6	B7	1260	ORA A ;IDENTIFICA PSEUDO-INSTR SI SALT LA SECVENTA CORESP
07F7	CA3908	1261	JZ ORG2
07FA	FA1E08	1262	JM DAT2
07FD	E22708	1263	JPO EQU2
0800	FE05	1264	CPI 5
0802	DA0E08	1265	JC RES2
0805	C22E09	1266	JNZ EASS ;END, ADICA SALT LA EASS
0808	CD0E09	1267	AC02: CALL TYS6 ;DW - TYS6 PREIA VALOAREA
080B	C38707	1268	JMP AC01
080E	CD010B	1269	RES2: CALL ASBL ;DS
0811	44	1270	MOV B,H ;ASBL PREIA OPERANDUL
0812	4D	1271	MOV C,L ;SE INCREMENTEAZA CONTORUL MEMORIEI
0813	2A8C60	1272	LHLD BBUF+2
0816	09	1273	DAD B
0817	228C60	1274	SHLD BBUF+2
081A	AF	1275	RES21: XRA A
081B	C32508	1276	JMP GCN2
081E	CBCD08	1277	DAT2: CALL TYS5 ;DB
0821	AF	1278	DAT2A: XRA A ;TYS5 PREIA OPERANDUL
0822	0E01	1279	MVI C,1
0824	C32208	1280	JMP GCN1
0827	CD010B	1281	EQU2: CALL ASBL ;EQU
082A	EB	1282	BINAD: XCHG ;RUTINA DEPUNE LA OBUF+2 IN HEXA VALOAREA
082B	210761	1283	LXI H, OBUF+2 ;PRELUATA DIN H.L IN BINAR
082E	7A	1284	MOV A,D ;LA REVENIRE IN D,E SE AFLA VALOAREA DIN H.L
082F	CD6F02	1285	CALL BINH+3
0832	23	1286	INX H
0833	7B	1287	MOV A,E
0834	CD6F02	1288	CALL BINH+3
0837	23	1289	INX H
0838	C9	1290	RET
0839	CD010B	1291	ORG2: CALL ASBL ;ORG - ASBL PREIA PARAMETRUL
083C	3A0561	1292	LDA OBUF
083F	FE20	1293	CPI
0841	C0	1294	RNZ ;RETURN LA EROARE
0842	CD2A08	1295	CALL BINAD ;DEPUNE NOUL PC IN OBUF
0845	2A9260	1296	LHLD ASPC ;MODIFICA ASPC
0848	EB	1297	XCHG
0849	229260	1298	SHLD ASPC
084C	7D	1299	MOV A,L ;OBTINE DIFERENTA ORIGINILOR
084D	93	1300	SUB E
084E	5F	1301	MOV E,A
084F	7C	1302	MOV A,H
0850	9A	1303	SBB D
0851	57	1304	MOV D,A
0852	2A8C60	1305	LHLD BBUF+2 ;ADUNA DIFERENTA LA POINTERUL MEMORIEI
0855	19	1306	DAD D
0856	228C60	1307	SHLD BBUF+2
0859	C9	1308	RET
085A	CD1B09	1309	TYP1: CALL ASTO ;PRELUCREAZA TIPUL 1 DE INSTR: DE 1 OCT FARA OP
085D	C9	1310	RET ;ASTO DEPUNE VALOAREA IN MEMORIE
085E	CD010B	1311	TYP2: CALL ASBL ;PRELUCREAZA TIP 2 : STAX, LDAX
0861	C4AEOC	1312	CNZ ERRL ;PREIA VALOARE REGISTRU
0864	7D	1313	MOV A,L
0865	B7	1314	ORA A

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 25

LOC	OBJ	LINE	SOURCE STATEMENT
0866	C8208	1315	JZ TY31
0869	FE02	1316	CPI 2
086B	C4AEOC	1317	CNZ ERRR
086E	C38208	1318	JMP TY31
0871	CDC10B	1319	TYP3: CALL ASBL ;PRELUCREAZA TIP 3: PUSH, POP, INX, DCX, DAD
0874	C4AEOC	1320	CNZ ERRR ;PREIA VALGARE REGISTRU
0877	7D	1321	MOV A,L ;FORMEAZA CODUL INSTRUCIUNII IN A
0878	0F	1322	RRC ;LA REGISTRU INCORECT SPECIFICAT SALT LA ERRR
0879	DCAEOC	1323	CC ERRR
087C	17	1324	RAL
087D	FE08	1325	CPI 8
087F	D4AEOC	1326	CNC ERRR
0882	07	1327	TYP3: RLC
0883	17	1328	RAL
0884	17	1329	RAL
0885	47	1330	TYP2: MOV B,A
0886	1A	1331	LDAX D
0887	80	1332	ADD B
0888	FE76	1333	CPI 118
088A	CCAEOC	1334	CZ ERRR
088D	C35A08	1335	JMP TYP1
0890	CDC10B	1336	TYP4: CALL ASBL ;PRELUCREAZA TIP 4: INSTR CU ACC, INR, DCR, MOV, RST
0893	C4AEOC	1337	CNZ ERRR
0896	7D	1338	MOV A,L ;SALT LA REGISTRU ILEGAL
0897	FE08	1339	CPI 8
0899	D4AEOC	1340	CNC ERRR
089C	1A	1341	LDAX D ;IN A COD INSTRUCIUNE DE BAZA
089D	FE40	1342	CPI 64
089F	CAAE08	1343	JZ TY41 ;SALT PENTRU MOVE
08A2	FEC7	1344	CPI 199
08A4	7D	1345	MOV A,L
08A5	C8208	1346	JZ TY31 ;SALT PENTRU RST
08A8	F8508	1347	JM TY32 ;SALT PENTRU INSTR CU ACCUMULATORUL
08AB	C38208	1348	JMP TY31 ;SALT PENTRU INR, DCR
08AE	29	1349	TYP4: DAD H ;PRELUCREAZA INSTRUCIUNEA MOV
08AF	29	1350	DAD H
08B0	29	1351	DAD H
08B1	85	1352	ADD L
08B2	12	1353	STAX D
08B3	CDEC08	1354	CALL MPNT
08B6	CDC40B	1355	CALL ASCN
08B9	C4AEOC	1356	CNZ ERRR
08BC	7D	1357	MOV A,L
08BD	FE08	1358	CPI 8
08BF	D4AEOC	1359	CNC ERRR
08C2	C38508	1360	JMP TY32
08C5	FE06	1361	TYP5: CPI 6 ;PRELUCREAZA TIP 5: INSTRUCIUNI IMEDIATE
08C7	CCDA08	1362	CZ TY56 ;SALT PENTRU MVI
08CA	CD1B09	1363	CALL ASTO ;DEPUNE COD OBIECT
08CD	CDC10B	1364	TYP5: CALL ASBL ;PREIA ARGUMENTUL IMEDIAT
08D0	3C	1365	INR A
08D1	F802	1366	CPI 2
08D3	D4C70C	1367	CNC ERRR ;SALT LA ARGUMENT INCORECT
08D6	78	1368	MOV A,L
08D7	C35A08	1369	JMP TYP1

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 28

LOC	OBJ	LINE	SOURCE STATEMENT
08DA	CDC10B	1370	TY56: CALL ASBL ;PREIA PRIMUL ARGUMENT LA INSTR CU 2 ARGUMENTE
08DD	C4AE0C	1371	CNZ ERRR
08E0	7D	1372	MOV A,L ;SALT LA REGISTRU ILEGAL
08E1	FE08	1373	CPI 8
08E3	D4AE0C	1374	CNC ERRR
08E6	29	1375	DAD H
08E7	29	1376	DAD H
08E8	29	1377	BAD H
08E9	1A	1378	LDAX D
08EA	85	1379	ADD L
08EB	5F	1380	MOV E,A
08EC	2A9660	1381	MPNT: LHL D PNTR ;DECIDE DACA SINTAXA ESTE CORECTA LA INSTR CU
08EF	7E	1382	MOV A,M ;DOUA ARGUMENTE
08F0	FE2C	1383	CPI ' ' ;PRIN VERIFICAREA VIRGULEI
08F2	23	1384	INX H
08F3	229660	1385	SHL D PNTR
08F6	C2B70C	1386	JNZ ERRS
08F9	7B	1387	MOV A,E
08FA	C9	1388	RET
08FB	FE01	1389	TYP6: CPI 1 ;PRELUCEAZA TIP 6: INSTR PE 3 OCTETI, LXI CAZ SPECIAL
08FD	C20B09	1390	JNZ TY6 ;SALT DACA NU ESTE LXI
0900	CDDA09	1391	CALL TY56 ;PREIA REGISTRU
0903	E609	1392	ANI 8
0905	C4AE0C	1393	CNZ ERRR ;SALT LA REGISTRU ILEGAL
0908	7B	1394	MOV A,E
0909	E6F7	1395	ANI 0F7H
090B	CD1B09	1396	TY6: CALL ASTO ;DEPUNE COD OBIECT
090E	CDC10B	1397	TY56: CALL ASBL ;PREIA OPERAND
0911	7D	1398	MOV A,L
0912	54	1399	MOV D,W
0913	CD1B09	1400	CALL ASTO ;DEPUNE OCTETUL 2
0916	7D	1401	MOV A,D
0917	C35A09	1402	JMP TYP1
091A	C9	1403	RET
		1404	;PUNE CODUL OBIECT DE LA PAS 2
091B	2A8C60	1405	ASMO: LHL BBUF+2 ;DEPUNE IN MEMORIE IMAGINEA OBIECT
091E	77	1406	MOV M,A
091F	23	1407	INX H
0920	228C60	1408	SAL B BBUF+2
0923	2A9E60	1409	LHL D OIND
0926	23	1410	INX H
0927	CD6F02	1411	CALL BINH+3
092A	229E60	1412	SHL D OIND
092D	C9	1413	RET
		1414	;TERMINAREA ASAMBLARII
092E	3A9460	1415	EASS: LDA PAS1 ;SALT LA TERMINARE PAS1 SI PAS2
0931	B7	1416	ORA A
0932	C25900	1417	JNZ EOR
0985	3E01	1418	MVI A,1 ;SALT LA PAS2 LA TERMINARE PAS1
0937	C38106	1419	JMP ASMO
		1420	;CAUTA CAR DIFERIT DE BLANC
093A	2A9660	1421	SBLK: LHL PNTR ;EXPLOREAZA IBUF SI POZITIONEAZA POINTERUL DE CITIRE
093D	7E	1422	SBL1: MOV A,M ;PNTR PE PRIMUL CARACTER DIFERIT DE BLANC
093E	FE20	1423	CPI ' ' ;
0940	00	1424	RNZ ;

SFDX-18 8080/8085_MACRO ASSEMBLER, V3.0

MODULE PAGE 27

LOC	OBJ	LINE	SOURCE STATEMENT
0941	23	1425	SBL2: INX H
0942	229660	1426	SHLD FNIR
0945	038D09	1427	JMP SBL1
0948	217F60	1428	; DETERMINA NATURA SALTULUI
094B	227460	1429	COND: LXI H,ARUF+1 ; DETERMINA CONDITIA LA SALTURI
094E	0602	1430	SHLD ADDS ; CONDITIONATE : JMP, CALL, RET
0950	0D7D0A	1431	MVI B,2
0953	C9	1432	CALL COPC
		1433	RET
0954	4F5247	1434	; TABELA CORESPONDENTE NUME COD
0957	00	1435	OTAR: DB 'ORG',0,0 ; PSEUDO-INSTRUCTIUNI
0958	00		
0959	455155	1436	DB 'EQU',0,1
095C	00		
095D	01		
095E	4442	1437	DB 'DB',0,0,-1
0960	00		
0961	00		
0962	FF		
0963	4453	1438	DB 'DS',0,0,3
0965	00		
0966	00		
0967	03		
0968	4457	1439	DB 'DW',0,0,5
096A	00		
096B	00		
096C	05		
096D	454E44	1440	DB 'END',0,6,0
0970	00		
0971	06		
0972	00		
0973	484C54	1441	DB 'HLT',118 ; TIP 1: INSTRUCTIUNI DE 1 OCTET PE 3 CARACTERE
0974	76		
0977	524C43	1442	DB 'RLC',7
097A	07		
097B	525243	1443	DB 'RRC',15
097E	0F		
097F	52414C	1444	DB 'RAL',23
0982	17		
0983	524152	1445	DB 'RAR',31
0986	1F		
0987	524554	1446	DB 'RET',201
098A	C9		
098B	434D41	1447	DB 'CMA',47
098E	2F		
098F	535443	1448	DB 'STC',55
0992	37		
0993	444141	1449	DB 'DAA',39
0996	27		
0997	434D43	1450	DB 'CMC',63
099A	3F		
099B	4549	1451	DB 'EI',0,251
099D	00		
099E	FB		

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 28

LOC	OBJ	LINE	SOURCE STATEMENT	
099F	4449	1452	DB 'DI',0,243	
09A1	00			
09A2	F3			
09A3	4E4F50	1453	DB 'NOP',0,0	;TIP 1: INSTRUCȚIUNI DE 1 OCTET PE 4 CARACTERE
09A6	00			
09A7	00			
09A8	58434847	1454	DB 'XCHG',235	
09AC	EB			
09AD	5854484C	1455	DB 'XTHL',227	
09B1	E3			
09B2	5350484C	1456	DB 'SFHL',249	
09B6	F9			
09B7	5043484C	1457	DB 'PCHL',233,0	
09BB	E9			
09BC	00			
09BD	53544158	1458	DB 'STAX',2	;TIP 2
09C1	02			
09C2	4C444158	1459	DB 'LDAX',10,0	
09C6	0A			
09C7	00			
09C8	50555348	1460	DB 'PUSH',197	;TIP 3
09CC	CS			
09CD	504F50	1461	DB 'POP',0,193	
09D0	00			
09D1	C1			
09D2	494E58	1462	DB 'INX',0,3	
09D5	00			
09D6	03			
09D7	444358	1463	DB 'DCX',0,11	
09DA	00			
09DB	0B			
09DC	444144	1464	DB 'DAD',0,9,0	
09DF	00			
09E0	09			
09E1	00			
09E2	494E52	1465	DB 'INR',4	;TIP 4
09E3	04			
09E6	444352	1466	DB 'DCR',5	
09E9	05			
09EA	4D4F56	1467	DB 'MOV',64	
09ED	40			
09EE	414444	1468	DB 'ADD',128	
09F1	80			
09F2	414443	1469	DB 'ADD',136	
09F3	88			
09F6	535542	1470	DB 'SUB',144	
09F9	90			
09FA	534242	1471	DB 'SBB',152	
09FD	98			
09FE	414E41	1472	DB 'ANA',160	
0A01	A0			
0A02	585241	1473	DB 'XRA',168	
0A05	A8			
0A06	4F5241	1474	DB 'ORA',176	
0A09	B0			

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 29

LOC	OBJ	LINE	SOURCE STATEMENT		
0A0A	434D50	1475	DB	'CMP',184	
0A0D	B8				
0A0E	525354	1476	DB	'RST',199,0	
0A11	C7				
0A12	00				
0A13	414449	1477	DB	'ADI',199	;TIP 5
0A16	C6				
0A17	414349	1478	DB	'ACI',206	
0A1A	CE				
0A1B	535549	1479	DB	'SUI',214	
0A1E	D6				
0A1F	534249	1480	DB	'SBI',222	
0A22	DE				
0A23	414E49	1481	DB	'ANI',230	
0A26	E6				
0A27	585249	1482	DB	'XRI',238	
0A2A	EE				
0A2B	4F5249	1483	DB	'ORI',246	
0A2E	F6				
0A2F	435049	1484	DB	'CPI',254	
0A32	FE				
0A33	494E	1485	DB	'IN',0,219	
0A35	00				
0A36	DB				
0A37	4F5554	1486	DB	'OUT',211	
0A3A	D3				
0A3B	4D5649	1487	DB	'MVI',6,0	
0A3E	06				
0A3F	00				
0A40	4A4D50	1488	DB	'JMP',0,195	;TIP 6
0A43	00				
0A44	C3				
0A45	43414C4C	1489	DB	'CALL',205	
0A49	CD				
0A4A	4C5B49	1490	DB	'LXI',0,1	
0A4D	00				
0A4E	01				
0A4F	4C4441	1491	DB	'LDA',0,58	
0A52	00				
0A53	3A				
0A54	535441	1492	DB	'STA',0,50	
0A57	00				
0A58	32				
0A59	53484C44	1493	DB	'SHLD',34	
0A5D	22				
0A5E	4C484C44	1494	DB	'LHLD',42,0	
0A62	2A				
0A63	00				
0A64	4E5A	1495	DB	'NZ',0 ;TABELA DE CONDITII	
0A66	00				
0A67	5A	1496	DB	'Z',0,8	
0A68	00				
0A69	08				
0A6A	4E43	1497	DB	'NC',16	
0A6C	10				

SFDX-18 8080/8085 MACRO ASSEMBLER. V3.0

MODULE PAGE 30

LOC	OBJ	LINE	SOURCE STATEMENT
0A6D	43	1498	DB 'C',0,24
0A6E	00		
0A6F	18		
0A70	504F	1499	DB 'P0',32
0A72	20		
0A73	5045	1500	DB 'PE',40
0A75	28		
0A76	50	1501	DB 'P',0,48
0A77	08		
0A78	30		
0A79	40	1502	DB 'M',0,56,0
0A7A	00		
0A7B	38		
0A7C	00		
		1503	;
		1504	-----
		1505	;SUBROUTINA COPC
		1506	-----
		1507	;CAUTA IN OTAB
		1508	;INTRARI: D,E=ADRESA TABELEI
		1509	; B=LUNGIMEA SIRULUI DE CARACTERE DE CAUTAT
		1510	; (ADDS)=ADRESA SIRULUI DE CARACTERE
		1511	;IESIRI: Z=0, NU S-A GASIT
		1512	; Z=1, SIR IDENTIFICAT SI IN A VALOAREA DUPA SIR IDENTIFICAT
		1513	;
0A7D	2A7460	1514	COPC: LHLD ADDS
0A80	1A	1515	LDAX D
0A81	B7	1516	ORA A
0A82	CA8FOA	1517	JZ COP1
0A85	48	1518	MOV C,B
0A86	CD2D01	1519	CALL SEAR
0A89	1A	1520	LDAX D
0A8A	C8	1521	RZ
0A8B	13	1522	INX D
0A8C	C37D0A	1523	JMP COPC
0A8F	3C	1524	COP1: INR A
0A90	13	1525	INX D
0A91	C9	1526	RET
		1527	;
		1528	-----
		1529	;SUBROUTINA OPCD
		1530	-----
		1531	;IDENTIFICA CODUL OPERATIEI PRINTRE CELE DIN OTAB
		1532	;LA PAS 1 INCREMENTEAZA PC ADICA (ASPC)
		1533	;LA PAS 2 LASA IN A VALOAREA DIN OTAB CORESP INSTRUCIUNII IDENTIFICATE
		1534	; SI LANSEAZA SECVENTA CORESP TIPULUI INSTR
		1535	;LOCAL B=NUMAR DE CARACTERE DIN INSTRUCIUNE
		1536	; C=NUMAR DE OCTETI AI INSTRUCIUNII
		1537	; H,L=ADRESA TIPULUI UNDE SE FACE SALTUL
		1538	;MODIFICA PC
		1539	;
0A92	217E60	1540	OPCD: LXI H,ADUF ;ADDS CONTINE ADRESA SIRULUI DE CARACTERE=INSIB
0A95	227460	1541	SHLD ADDS
0A98	115409	1542	LXI D,OTAB
0A9B	0604	1543	MVI B,4

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 31

LOC	OBJ	LINE	SOURCE STATEMENT
0A9D	CD7D0A	1544	CALL COPC
0AA0	CA3B0B	1545	JZ PSEU ;SALT PENTRU PSEUDO-INSTR, 4 CAR, B=4
0AA3	05	1546	DCR B
0AA4	CD7D0A	1547	CALL COPC
0AA7	CAAE0A	1548	JZ OP1 ;SALT LA TIP 1. LA OP1, 3 CARACTERE, B=3
0AAA	04	1549	INR B
0AAB	CD7D0A	1550	CALL COPC
0AAE	215A08	1551	OP1: LXI H,TYP1
0AB1	0E01	1552	OP2: MVI C,1
0AB3	CA0E0B	1553	JZ OCNT ;SALT LA TIP 1, LA OCNT, B=4, C=1
0AB6	CD7D0A	1554	OPC2: CALL COPC
0AB9	215E08	1555	LXI H,TYP2
0ABC	CAB10A	1556	JZ OP2 ;SALT LA OP2 PENTRU STAX, LDAX
0ABF	CD7D0A	1557	CALL COPC
0AC2	217108	1558	LXI H,TYP3
0AC5	CAB10A	1559	JZ OP2 ;SALT LA OP2 PENTRU TIP 3
0AC8	05	1560	DCR B
0AC9	CD7D0A	1561	CALL COPC
0ACC	219008	1562	LXI H,TYP4
0ACF	CAB10A	1563	JZ OP2 ;SALT LA OP2 PENTRU TIP 4, B=3
0AD2	CD7D0A	1564	OPC3: CALL COPC
0AD5	21C508	1565	LXI H,TYP5
0AD8	0E02	1566	MVI C,2
0ADA	CA0E0B	1567	JZ OCNT ;SALT LA OCNT PENTRU TIP 5, INSTR IMEDIATE, C=2
0ADD	04	1568	INR B
0ADE	CD7D0A	1569	CALL COPC
0AE1	CA090B	1570	JZ OP4 ;SALT LA OP4 PENTRU TIP 6, B=4
0AE4	CD4809	1571	CALL COND ;IDENTIFICAREA CONDITIEI DE SALT
0AE7	C2330B	1572	JNZ OERR ;SALT LA EROARE
0AEA	C6C0	1573	ADI 192
0AEC	57	1574	MOV D,A
0AED	0603	1575	MVI B,3
0AEF	3A7E60	1576	LDA ABUF
0AF2	4F	1577	MOV C,A
0AF3	F52	1578	CPI 'R' ;SALT LA OP1 PENTRU RETURN CONDITIONAT
0AF5	7A	1579	MOV A,D
0AF6	CAAE0A	1580	JZ OP1
0AF9	79	1581	MOV A,C ;SALT LA OPAD PENTRU JMP CONDITIONAT
0AFA	14	1582	INR D
0AFB	14	1583	INR D
0AFC	FE4A	1584	CPI 'J'
0AFE	CA080B	1585	JZ OPAD
0B01	FE43	1586	CPI 'C' ;FORMEAZA CALL CONDITIONAT
0B03	C2330B	1587	JNZ OERR
0B06	14	1588	INR D
0B07	14	1589	INR D
0B08	7A	1590	OPAD: MOV A,D ;IN A COD OPERATIE
0B09	21FB08	1591	OP4: LXI H,TYP6 ;IN H,L ADRESA DE SALT
0B0C	0E03	1592	OP5: MVI C,3 ;3 OCTETI
0B0E	329D60	1593	OCNT: STA TEMP ;DEPUNE TEMPORAR CODUL OPERATIEI
0B11	3E7E	1594	MVI A,ABUF AND OFFH
0B13	80	1595	ADD B
0B14	5F	1596	MOV E,A
0B15	3E60	1597	MVI A,ABUF/256
0B17	CE00	1598	ACI 0

BFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 32

LOC	OBJ	LINE	SOURCE STATEMENT
OB19	S7	1599	MOV D, A
OB1A	1A	1600	LDAX D
OB1B	B7	1601	ORA A ;SALT LA OERR LA EROARE
OB1C	C2330B	1602	JNZ OERR ;ADICA DACA DUFA COD INSTR URMEAZA CAR DIFERIT DE 0
OB1F	3A9460	1603	LDA PASI ;IN A INDICATOR DE PAS
OB22	0600	1604	OCN1: MVI B, 0 ;B=0
OB24	EB	1605	XCHG
OB25	2A9260	1606	OCN2: LHLD ASPC ;(ASPC)=(ASPC)+B,C
OB28	09	1607	DAD B ;ADICA INCREMENTEAZA PC CU NR DE OCTETI AI INSTR
OB29	229260	1608	SHLD ASPC ;RESPECTIVE, CONTINUT IN REGISTRUL C (B=0)
OB2C	B7	1609	ORA A
OB2D	C8	1610	RZ ;RETURN LA PAS 1
OB2E	3A9D60	1611	LDA TEMP ;LA PAS 2 SE LANSEAZA SECVENTA CORESPUNZATOARE
OB31	EB	1612	XCHG ;TIPULUI IDENTIFICAT, ADICA TYP1-TYP6
OB32	E9	1613	PCHL ;IN REGISTRUL A SE AFLA CODUL OPERATIEI DE BAZA
OB33	21DA0C	1614	OERR: LXI H, ERRO ;SALT LA ERRO IN CAZ DE EROARE IN CODUL OPERATIEI
OB36	0E03	1615	MVI C, 3 ;ALOCA 3 OCTETI, C=3
OB38	0B1F0B	1616	JMP OCN1-3
OB39	218260	1617	PSEU: LXI H, ABUF+4 ;AICI SE AJUNGE LA PSEUDO-INSTRUCTIUNI
OB3E	7E	1618	MOV A, M
OB3F	B7	1619	ORA A
OB40	C2330B	1620	JNZ OERR ;SALT LA EORR DACA EROARE, CAR DUFA COD DIFERIT DE 0
OB43	3A9460	1621	LDA PASI ;SALT LA PSUI PENTRU PAS 1 SI PSU2 PENTRU PAS 2
OB46	B7	1622	ORA A
OB47	CA7107	1623	JZ PSU1
OB4A	C3F507	1624	JMP PSU2
		1625	;
		1626	;
		1627	;SUBROUTINA SLAB
		1628	;
		1629	;EXECUTA PRELUCRAREA ETICHETELOR
		1630	;SLAB ESTE UTILIZAT IN DUFA SCOPURI:
		1631	; 1) IDENTIFICARE REGISTRU PREDEFINIT IN TABELA RTAB
		1632	; CY=0, Z=1 S-A GASIT, IN H,L O, COD REGISTRU
		1633	; NU S-A GASIT, SE CONSIDERA ETICHETA, SALT LA 2
		1634	; 2) IDENTIFICARE ETICHETA IN TABELA DE LA SYMT
		1635	; NOLA=0, ADICA NU SINT SIMBOLI IN TABELA, CY=0, Z=0
		1636	; NOLA DIFERIT DE 0
		1637	; DACA ETICHETA GASITA, CY=0, Z=1, H,L=VALOAREA ETICHETEI
		1638	; DACA ETICHETA NEGASITA, CY=0, Z=0, D,E=ADRESA SIMBOLULUI URMATOR
		1639	; 3) CY=1 PENTRU SIMBOL ILEGAL
		1640	;
OB4D	FE41	1641	SLAB: CPI 'A'
OB4F	D8	1642	RC ;RETURN LA EROARE CU CY=1
OB50	FE5B	1643	CPI 'Z'+1
OB52	3F	1644	CMC
OB53	D8	1645	RC
OB54	0DA20B	1646	CALL ALPS ;SIMBOL ADUS IN ABUF
OB57	217E60	1647	LXI H, ABUF ;H,L MEMORAT LA ADDS
OB5A	227460	1648	SHLD ADDS
OB5D	05	1649	DCR B
OB5E	C2710B	1650	JNZ SLAI ;SALT LA SIMBOL DIN MAI MULTE CARACTERE
OB61	04	1651	INR B
OB62	118D0B	1652	LXI D, RTAB
OB65	0D7D0A	1653	CALL COPL

LOC	OBJ	OC	OBJ	LOC	OBJ	LINE	SOURCE STATEMENT	LINE	OBJ	SOURCE STATEMENT
B68	C2710B			1654	JNZ SLA1			1692		
B6B	6F			1655	MOV L,A			1693	ALPS:	B,O
B6C	2600			1656	MOV H,O			1694	STAX D	D
B6E	C870B			1657	MOV SLA2			1694	ALP1:	B
B71	3A9860			1658	LDA SLA1:			1695	INR B	A,B
B74	47			1659	MOV B,A			1696	MOV CFI	11
B75	117C61			1660	LXI D,SYMT			1698	RNC	D
B78	B7			1661	ORA A			1699	INX D	H
B79	CA3A0B			1662	JZ SLA3			1700	INX H	PNTR
B7C	3E05			1663	MOV A,LLAB			1701	SHLD MOV	A,M
B7E	329560			1664	STA NCHR			1702	RC	'0'
B81	CD1601			1665	CALL COMG			1703	RC	'0'
B84	4C			1666	MOV C,H			1704	RC	'0'
B85	65			1667	MOV H,L			1705	CPI	'9'+1
B86	69			1668	MOV L,C			1706	JC	ALP1
B87	37			1669	STC SLA2:			1707	RC	'A'
B88	3F			1670	CMC			1708	RC	'Z'+1
B89	C9			1671	RET			1709	CPI	ALP1
B8A	3C			1672	SLA3:			1710	JC	ALP1
B8B	B7			1673	ORA A			1711	RET	
B8C	C9			1674	RET			1712		
B8D	41			1675	TABELA REGISTRE			1713		
B8E	07			1676	RTAB:			1714	SUBROUTINA ASBL	
B8F	42							1715		
B90	00							1716	ASBL ESTE UN ANALIZOR SINTACTIC PENTRU CIMPUL OPERAND	
B91	43							1717	CARE LASA IN H,L VALOAREA OPERANDULUI	
B92	01							1718		
B93	44							1719	ASBL:	CALL
B95	45							1720	ASBL:	LXI H,0
B96	03							1721	ASBL:	SHLD OPRD
B99	4C							1722	ASBL:	INR H
B94	02							1723	ASBL:	SHLD OPR1-1
B95	45							1724	ASBL:	LHLD PNTR
B97	48							1725	ASBL:	DCX H
B98	04							1726	ASBL:	CALL ZBUF
B99	4C							1727	ASBL:	STA SIGN
B9A	05							1728	ASBL:	INX H
B9B	4D							1729	ASBL:	MOV A,M
B9C	06							1730	ASBL:	'++'
B9D	50							1731	ASBL:	JC CEND
B9E	06							1732	ASBL:	CPI
B9F	53							1733	ASBL:	JZ SEID
BA0	06							1734	ASBL:	'++'
BA1	00							1735	ASBL:	CPI
								1736	ASBL:	'++'
								1737	ASBL:	JNZ ASC2
								1738	ASBL:	STA SIGN
								1739	ASBL:	LDA OPR1
								1740	ASBL:	CPI 2
								1741	ASBL:	JZ ERRS
								1742	ASBL:	MOV A,2
								1743	ASBL:	STA OPR1
								1744	ASBL:	MOV NXT2
								1745	ASBL:	CPI
								1746	ASBL:	LDA OPR1

;SALT LA NEIDENTIFICAREA REGISTRULUI: PREDEFINIT
 ;IDENTIFICARE REGISTRE
 ;SALT LA SLA3 DACA TABELA ESTE VIDA
 ;CAUTA IN TABELA
 ;INTERSCHEMIA H CU L
 ;POZITIONARE INDICATORI
 ;A,7,B,0,C,1
 ;D,2,E,3,H,4
 ;L,5,M,6
 ;P,6,S,6,0
 ;SALT LA CR SAU BLANC
 ;SALT LA VIRGULA
 ;SALT LA PLUS
 ;SALT LA MINUS
 ;MEMOREAZA A LA SIGN
 ;SALT LA ERGARE
 ; INCARCA 1 IN OPR1
 ; INCARCA 0 IN OPRD
 ; INCARCA 1 IN OPR1
 ; INCARCA IN H,L PNTR
 ;ABUF(12) IA VALOAREA 0
 ; INCARCA 0 IN SIGN

;SALT LA CR SAU BLANC
 ;SALT LA VIRGULA
 ;SALT LA PLUS
 ;SALT LA MINUS
 ;MEMOREAZA A LA SIGN
 ;SALT LA ERGARE
 ; INCARCA 2 IN OPR1
 ; INCARCA 1 IN OPR1
 ; INCARCA 0 IN OPRD
 ; INCARCA 0 IN SIGN

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
0C05	B7	1747	ORA	0C7A	7AD50C	1802	JC
0C06	CAB70C	1748	A	0C7D	C3C20C	1803	ERRA
0C09	79	1749	ERRS	0C80	3A9C60	1804	ERRU
0C0A	FE24	1750	A,C	0C83	B7	1805	OPRI
0C0C	C2190C	1751	ASC3	0C84	C2B70C	1806	A
0C0F	23	1752	INX	0C87	2A9A60	1807	ERRS
0C10	229660	1753	H	0C8A	7C	1808	OPRD
0C13	2A9260	1754	FNTR	0C8B	119D60	1809	A,H
0C16	C3550C	1755	ASFC	0C8E	B7	1810	D,TEMP
0C19	FE27	1756	AVAL	0C8F	C9	1811	A
0C1B	C2450C	1757	27H			1812	
0C1E	110000	1758	ASC5			1813	
0C21	0E03	1759	LXI			1814	SUBROUTINA NUMS
0C23	23	1760	MVI			1815	
0C24	229660	1761	INX			1816	PREIA IN A O VALOARE NUMERICA FIE HEXA FIE Zecimala
0C27	7E	1762	SHLD			1817	CONVERSIILE SE EXECUTA PRIN ROUTINELE:
0C2B	FE0D	1763	MOV			1818	ADEC : Zecimal - BINAR
0C2A	CAD50C	1764	ASC4			1819	AHEX : HEXA - BINAR
0C2D	FE27	1765	CPI			1820	LA REVENIRE CY=1 INDICA ERGARE
0C2F	C23C0C	1766	JZ			1821	
0C32	23	1767	INX			1822	NUMS:
0C33	229660	1768	SHLD	0C90	CDA20B	1823	CALL
0C36	7E	1769	MOV	0C93	1B	1824	DCX
0C37	FE27	1770	CPI	0C94	1A	1825	LDAX
0C39	C2560C	1771	FNTR	0C95	017E60	1826	LXI
0C3C	0D	1772	ASC4	0C98	FE48	1827	B,ABUF
0C3D	CAD50C	1773	ERRA	0C9A	CAAB0C	1828	'H'
0C40	53	1774	MOV	0C9D	FE44	1829	NUM2
0C41	5F	1775	MOV	0C9F	C2A40C	1830	'D'
0C42	C2230C	1776	ASC4	0CA2	AF	1831	NUM1
0C45	FE30	1777	ERRA	0CA3	12	1832	XRA
0C47	DAD50C	1778	JC	0CA4	CDDE01	1833	STAX
0C4A	FE3A	1779	ERRA	0CA7	C9	1834	CALL
0C4C	D2740C	1780	CPI	0CA8	AF	1835	RET
0C4F	CD900C	1781	ASC5	0CA9	12	1836	XRA
0C52	DAD50C	1782	ERRA	0CAA	CD501	1837	STAX
0C55	EB	1783	JC	0CAD	C9	1838	CALL
0C56	2A9A60	1784	XCHG			1839	AHEX
0C59	AF	1785	LHLD			1840	RET
0C5A	389C60	1786	STA			1841	
0C5D	3A9960	1787	TRA			1842	SUBROUTINA ERRR
0C60	B7	1788	LDA			1843	
0C61	C2680C	1789	ORA			1844	RUTINELE DE ERGARE
0C64	19	1790	FNTR			1845	DEFIN IN OBUI CODUL ERORII
0C65	229A60	1791	DAD			1846	ADRESUNZATUR ERORII IDENTIFICATE
0C68	C3CE0B	1792	SHLD	0CAE	3E52	1847	ERRR:
0C6B	7D	1793	JMP	0CB0	210000	1848	LXI
0C6C	93	1794	MOV	0CB3	320561	1849	STA
0C6D	6F	1795	SUB	0CB6	C9	1849	OBUI
0C6E	7C	1796	MOV	0CB7	3E53	1850	ERRS:
0C6F	9A	1797	MOV	0CB9	320561	1851	STA
0C70	67	1798	SEB	0CBC	210000	1852	OBUI
0C71	C3650C	1799	MOV	0CBF	C38A0C	1853	LXI
0C74	CD4D0B	1800	H,A	0CC2	3E55	1854	H,O
0C77	CAS50C	1801	ASC7	0CC4	C3E90C	1855	H,A
			SLAB	0CC7	3E56	1856	H,A
			CALL				ERRS+2
			JZ				A,V

GFDN-18 8080/8085 MACRO ASSEMBLER, V3.0 MODULE PAGE 37

LOC	OBJ	LINE	SOURCE STATEMENT
OCC9	E3800C	1857	JMP ERRR+2
OCCC	3E4D	1858	ERRR: MVI A, 'M'
OCCE	320561	1859	STA OBUF
OCDI	CD0906	1860	CALL ACUI
OCDA	C9	1861	RET
OCDS	3E41	1862	ERRA: MVI A, 'A'
OCB7	C3B90C	1863	JMP ERRS+2
OCDA	3E4F	1864	ERRO: MVI A, 'O'
OCDC	320561	1865	STA OBUF
OCDF	3A7460	1866	LDA PASI
OCE2	B7	1867	ORA A
OCE3	C8	1868	RZ
OCE4	0E03	1869	MVI C, 3
OCE6	AF	1870	ERO1: XRA A
OCE7	CD1B09	1871	CALL ASTO
OCEA	0D	1872	DCR C
OCEB	C2E60C	1873	JNZ ERO1
OCEE	C9	1874	RET
OCEF	3E4C	1875	ERRL: MVI A, 'L'
OCF1	C3DC0C	1876	JMP ERRO+2
OCF4	3E44	1877	ERRD: MVI A, 'D'
OCF6	320561	1878	STA OBUF
OCF9	CDCC06	1879	CALL ACUI
OCFC	C34B07	1880	JMP OPC
		1881	;
		1882	;
		1883	;COMANDA BREAKPOINT
		1884	;
		1885	;SETEAZA SAU RESETEAZA PUNCTE DE SUSPENDARE A EXECUTIEI
		1886	;
OCFF	3A7E60	1887	BREAK: LDA ABUF ;DACA COMANDA NU ARE PARAMETRII
OD02	B7	1888	ORA A ;SE SARE LA CLRB UNDE SE STERO TOATE BREAKPOINT-URILE
OD03	CA360D	1889	JZ CLRB
OD06	1608	1890	MVI D, NBR ;IN D, E NUMARUL DE BREAKPOINT-URI=6
OD08	210C60	1891	LXI H, BRT ;IN H, L ADRESA TABELEI
OD0B	7E	1892	B1: MOV A, M ;SALT LA B2 DACA S-A GASIT UN LOC LIBER
OD0C	23	1893	INX H ;IN TABELA DE BREAKPOINT-URI
OD0D	46	1894	MOV B, M
OD0E	80	1895	ORA B
OD0F	CA1B0D	1896	JZ B2
OD12	23	1897	INX H ;RETA CAUTAREA
OD13	23	1898	INX H
OD14	15	1899	DCR D
OD15	C20B0D	1900	JNZ B1
OD18	C36404	1901	JMP WHAT ;MESAJ DE EROARE LA TABELA P&INA
OD1B	2B	1902	B2: DCX H
OD1C	EB	1903	XCHG
OD1D	2A8A60	1904	LHLD BBUF ;IN H, L ADRESA DE BREAKPOINT
OD20	EB	1905	XCHG
OD21	7A	1906	MOV A, D ;NU SE POATE PUNE O BREAKPOINT
OD22	B7	1907	ORA A
OD23	C22C0D	1908	JNZ B2
OD24	7B	1909	MOV A, E
OD27	FE0B	1910	CPI 11
OD29	DA6404	1911	JC WHAT

SFOX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 38

LOC	OBJ	LINE	SOURCE STATEMENT
0D2C	72	1912	MOV M,D ;PUNE IN TABELA ADRESA HIGH SI ADRESA LOW
0D2D	23	1913	INX H
0D2E	73	1914	MOV M,E
0D2F	23	1915	INX H
0D30	1A	1916	LDAX D ;SALVEAZA IN TABELA OCTETUL DIN PROGRAM
0D31	77	1917	MOV M,A ;UNDE SE PUNE BREAKPOINT-UL
0D32	3ECF	1918	MVI A,(RST 1) ;IN LOCUL OCTETULUI RESPECTIV
0D34	12	1919	STAX D ;SE PUNE CODUL INSTRUCIUNII RST 1
0D35	C9	1920	RET
		1921	;STERGE BREAKPOINT
0D36	210060	1922	CLRR: LXI H,BRT ;SESTERG TOATE BREAKPOINT-URILE
0D39	0608	1923	MVI B,NBR
0D3R	AF	1924	CLBI: XRA A
0D3C	56	1925	MOV D,M ;CITESTE ADRESA HIGH
0D3D	77	1926	MOV M,A ;SI PUNE 0 IN LOC
0D3E	23	1927	INX H
0D3F	5E	1928	MOV E,M ;CITESTE ADRESA LOW
0D40	77	1929	MOV M,A ;PUNE 0 IN LOC
0D41	23	1930	INX H
0D42	46	1931	MOV B,M
0D43	23	1932	INX H
0D44	7A	1933	MOV A,D ;CONTINUA DACA ADRESA = 0 PRIN CL2
0D45	B3	1934	ORA E
0D46	CA480D	1935	JZ CL2
0D47	78	1936	MOV A,B ;REFA OCTETUL IN PROGRAM LA ADRESA
0D4A	12	1937	STAX D ;CITITA DIN TABELA
0D4B	06	1938	DCR B ;REIA PINA LA NRB=8
0D4C	123E0D	1939	JNZ CLBL
0D4F	C9	1940	RET
		1941	;LA ATINGEREA UNUI BREAKPOINT IN PROGRAM SE EXECUTA INSTRUCIUNEA RST 1
		1942	;CARE REALIZEAZA UN CALL CU ADRESA FIXA 0008H.
		1943	;LA ACEASTA ADRESA SE GASESTE UN JMP BRKP.
		1944	;IN ACEASTA SECVENTA DE PROGRAM SESALVEAZA TOATE REGISTRELE
		1945	;INTR-O ZONA INCEPIND DE LA ADRESA 4000H, ASTFEL:
		1946	; 4000H: INDICATORII
		1947	; 4001H: A
		1948	; 4002H: C
		1949	; 4003H: B
		1950	; 4004H: E
		1951	; 4005H: D
		1952	; 4006H: SP-LOW
		1953	; 4007H: SP-HIGH
		1954	; 4008H: L
		1955	; 4009H: H
		1956	; 400AH: PC-LOW
		1957	; 400BH: PC-HIGH
		1958	;SE STERGE APOI BREAKPOINT-UL SI SE INTRA IN MONITOR
		1959	;UTILIZATORUL PRIMESTE MESAJUL: 'XXXX BREAK'
		1960	;UNDE XXXX ESTE ADRESA IN HEXAZECIMAL
0050	200660	1961	BRKP: SHLD HOLD+8 ;SALVARE H,L
0053	E1	1962	POP H ;SALVARE PC
0054	20	1963	DCX H
0055	200A60	1964	SHLD HOLD+10
0058	F5	1965	PUSH PSW ;SALVARE INDICATORI
0059	E1	1966	POP H

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 39

LOC	OBJ	LINE	SOURCE STATEMENT
005A	220060	1967	CHLD HOLD
005D	210000	1968	LXI H,0
0060	39	1969	DAD SP
0061	310860	1970	LXI SP,HOLD+8
0064	E5	1971	PUSH H ;SALVARE SP
0065	D5	1972	PUSH D ;SALVARE D,E
0066	05	1973	PUSH B ;SALVARE B,C
0067	2F	1974	CMA
0069	310441	1975	LXI SP,AREA+100 ;REFACE SP
006B	2A0A60	1976	LHLD HOLD+10 ;INCARCA PC IN H,L
006E	EB	1977	XCHG
006F	210C40	1978	LXI H,BRT ;SE CAUTA IN TABELA BRT ADRESA
0072	0608	1979	MVI B,NBR ;CORESPUNZATOARE PUNCTULUI DE BREAKPOINT
0074	7E	1980	BL1: MOV A,M
0075	23	1981	INX H
0076	8A	1982	CMP D
0077	C27F00	1983	JNZ BL2
007A	7E	1984	MOV A,M
007B	BB	1985	CMP E
007C	CA8800	1986	JZ BL3
007F	23	1987	BL2: INX H
0080	23	1988	INX H
0081	05	1989	DCR B
0082	CA6404	1990	JZ WHAT
0085	C37400	1991	JMP BL1
0088	23	1992	BL3: INX H ;SE REFACE IN PROGRAMUL UTILIZATOR
0089	7E	1993	MOV A,M ;OCTETUL DE LA ADRESA PUNCTULUI DE BREAKPOINT
008A	12	1994	STAX D
008B	AF	1995	XRA A
008C	2B	1996	DCX H
008D	77	1997	MOV H,A
008E	2B	1998	DCX H
008F	77	1999	MOV M,A
0090	CD0000	2000	CALL CRLF ;AFISEAZA PC-HIGH IN HEXA
0093	3A0960	2001	LDA HOLD+11
0096	CD1402	2002	CALL HOUT
0099	3A0A60	2003	LDA HOLD+10 ;AFISEAZA PC-LOW IN HEXA
009C	CD1402	2004	CALL HOUT
009F	21A900	2005	LXI H,BMES ;AFISEAZA BMES
00A2	CD5462	2006	CALL SCRN
00A5	C35900	2007	JMP EOR
00A8	42324541	2008	BLMES: DB 'BREAK'.13
00AC	4B		
00AD	0B		
		2009	:
		2010	:-----
		2011	:COMANDA PROCFD
		2012	:-----
		2013	:CONTINUAREA EXECUTIEI PROGRAMULUI UTILIZATOR DUPA UN BREAKPOINT
		2014	:
00AE	3A7E60	2015	PROCF: LDA ABUF ;SALT DACA COMANDA NU ARE PARAMETRII
00B1	B7	2016	ORA A
00B2	CA8800	2017	JZ P1
00B5	2A8A60	2018	LHLD BRUF ;PUNE PARAMETRU=ADRESA DE CONTINUARE PE FOZITIA PC
00B8	20A660	2019	SILD HOLD+10

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 40

LOC	OBJ	LINE	SOURCE STATEMENT
ODBB	310060	2020	PI: LXI SP,HOLD ;POZITIONARE SP
ODBE	F1	2021	POP PSW ;REFACERE PSW
ODBF	C1	2022	POP B ;REFACERE B,C
ODCO	D1	2023	POP D ;REFACERE D,E
ODC1	E1	2024	POP H ;REFACERE S,P
ODC2	F9	2025	SPHL
ODC3	2A0A60	2026	LHLD HOLD+10 ;SE PUNE PC-UL IN STIVA CA ADRESA DE REVENIRE DIN CALL
ODC6	E5	2027	PUSH H
ODC7	2A086C	2028	LHLD HOLD+8 ;SE REFACE H,L
ODCA	C9	2029	RET ;SE INTRA IN PROGRAMUL UTILIZATOR
		2030	;GENERATORUL DE CARACTERE
		2031	PUBLIC BAZA
ODCB	38	2032	BAZA: DB 38H,44H,58H,58H,40H,3CH ;B
ODCC	44		
ODCD	58		
ODCE	58		
ODCF	40		
ODD0	3C		
ODD1	10	2033	DB 10H,28H,44H,7CH,44H,44H ;A
ODD2	28		
ODD3	44		
ODD4	7C		
ODD5	44		
ODD6	44		
ODD7	78	2034	DB 78H,44H,78H,44H,44H,78H ;B
ODD8	44		
ODD9	78		
ODDA	44		
ODDB	44		
ODDC	78		
ODDD	38	2035	DB 38H,44H,40H,40H,44H,38H ;C
ODDE	44		
ODDF	40		
ODE0	40		
ODE1	44		
ODE2	38		
ODE3	78	2036	DB 78H,44H,44H,44H,44H,78H ;D
ODE4	44		
ODE5	44		
ODE6	44		
ODE7	44		
ODE8	78		
ODE9	7C	2037	DB 7CH,40H,7CH,40H,40H,7CH ;E
ODEA	40		
ODEB	7C		
ODEC	40		
ODED	40		
ODEE	7C		
ODEF	7C	2038	DB 7CH,40H,7CH,40H,40H,40H ;F
ODF0	40		
ODF1	7C		
ODF2	40		
ODF3	40		
ODF4	40		
ODF5	38	2039	DB 38H,44H,40H,5CH,44H,38H ;G

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
0DF6	44	2040	DB 44H,44H,7CH,44H,44H,44H ;H	0E2D	79	2049	DB 36H,44H,24H,54H,48H,34R ;Q
0DF7	40			0E2E	40		
0DF8	5C			0E2F	40		
0DF9	44			0E30	40		
0DFA	38			0E31	38		
0DFB	44			0E32	44		
0DFC	44			0E33	44		
0DFD	7C			0E34	54		
0DFE	44			0E35	48		
0DFF	44			0E36	34		
0E00	44	2041	DB 36H,10H,10H,10H,10H,36H ;I	0E37	78	2050	DB 76H,44H,76H,50H,48H,44H ;R
0E01	36			0E38	44		
0E02	10			0E39	78		
0E03	10			0E3A	50		
0E04	10			0E3B	48		
0E05	10			0E3C	44		
0E06	38			0E3D	3C		
0E07	2C	2042	DB 3CH,8,8,8,48H,30H ;J	0E3E	40	2051	DB 3CH,40H,36H,4,4,76H ;S
0E08	08			0E3F	38		
0E09	08			0E40	04		
0E0A	08			0E41	04		
0E0B	48			0E42	78		
0E0C	30			0E43	7C	2052	DB 7CH,10H,10H,10H,10H,10H ;T
0E0D	48			0E44	10		
0E0E	50			0E45	10		
0E0F	60			0E46	10		
0E10	50			0E47	10		
0E11	48			0E48	10		
0E12	44			0E49	44	2053	DB 44H,44H,44H,44H,44H,36H ;U
0E13	40	2044	DB 40H,40H,40H,40H,40H,7CH ;L	0E4A	44		
0E14	40			0E4B	44		
0E15	40			0E4C	44		
0E16	40			0E4D	44		
0E17	40			0E4E	38		
0E18	7C			0E4F	44	2054	DB 44H,44H,44H,44H,26H,10H ;V
0E19	44	2045	DB 44H,6CH,54H,44H,44H,44H ;M	0E50	44		
0E1A	6C			0E51	44		
0E1B	54			0E52	44		
0E1C	44			0E53	28		
0E1D	44			0E54	10	2055	DB 44H,44H,44H,44H,54H,6CH,44H ;H
0E1E	44			0E55	44		
0E1F	44	2046	DB 44H,64H,54H,4CH,44H,44H ;N	0E56	44		
0E20	64			0E57	44		
0E21	54			0E58	54		
0E22	4C			0E59	6C		
0E23	44			0E5A	44		
0E24	44			0E5B	44	2056	DB 44H,26H,10H,10H,26H,44H ;X
0E25	3B	2047	DB 36H,44H,44H,44H,44H,36H ;D	0E5C	28		
0E26	44			0E5D	10		
0E27	44			0E5E	10		
0E28	44			0E5F	28		
0E29	44			0E60	44		
0E2A	3B			0E61	44	2057	DB 44H,26H,10H,10H,10H,10H ;Y
0E2B	78	2048	DB 76H,44H,76H,40H,40H,40H ;P	0E62	28		
0E2C	44			0E63	10		

SFDX-10 8080/8085 MACRO ASSEMBLER, V3.0 MODULE PAGE 43 SFDX-1B 8080/8085 MACRO ASSEMBLER, V3.0 MODULE PAGE 44

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ
0E54	10			0E9B	00				
0E55	10			0E9C	00				
0E56	10			0E9D	00				
0E57	7C	2058	DB 7CH,4,18H,30H,40H,7CH 1Z	0E9E	28	2067	DB 0,20H,7CH,20H,7CH,20H 1B		
0E58	04			0E9F	7C				
0E59	18			0EA0	28				
0E5A	30			0EA1	7C				
0E5B	40			0EA2	28				
0E5C	7C			0EA3	10				
0E5D	60	2059	DB 60H,40H,40H,40H,40H,60H 1I	0EA4	38	2068	DB 10H,38H,50H,38H,14H,38H 1B		
0E5E	40			0EA5	50				
0E5F	40			0EA6	38				
0E70	40			0EA7	14				
0E71	40			0EA8	38				
0E72	60	2060	DB 20H,20H,10H,8,4,4 ;SLASH LEFT	0EA9	00	2069	DB 0,24H,8,10H,24H,0 1X		
0E73	20			0EAA	24				
0E74	20			0EAB	08				
0E75	10			0EAC	10				
0E76	08			0EAD	24				
0E77	04			0EAE	00				
0E78	04			0EAF	20	2070	DB 20H,50H,20H,54H,48H,34H 1I		
0E79	0C	2061	DB 0CH,4,4,4,4,0CH 1J	0EB0	50				
0E7A	04			0EB1	20				
0E7B	04			0EB2	54				
0E7C	04			0EB3	48				
0E7D	04			0EB4	34				
0E7E	0C			0EB5	08	2071	DB 8,10H,0,0,0,0 1K		
0E7F	10	2062	DB 10H,28H,44H,0,0,0 1CARETA	0EB6	10				
0E80	28			0EB7	00				
0E81	44			0EB8	00				
0E82	00			0EB9	00				
0E83	00			0EBA	00				
0E84	00			0EBB	20	2072	DB 20H,40H,40H,40H,40H,20H 1L		
0E85	00	2063	DB 0,0,0,0,0,7CH ;BARA JDS	0EBC	40				
0E86	00			0EBD	40				
0E87	00			0EBE	40				
0E88	00			0EBF	40				
0E89	00			0EC0	20				
0E8A	7C			0EC1	08	2073	DB 8,4,4,4,4,8 1M		
0E8B	00	2064	DB 0,0,0,0,0,0 1BLANC	0EC2	04				
0E8C	00			0EC3	04				
0E8D	00			0EC4	04				
0E8E	00			0EC5	04				
0E8F	00			0EC6	08				
0E90	00			0EC7	00	2074	DB 0,10H,54H,38H,54H,10H 1N		
0E91	10	8065	DB 10H,10H,10H,10H,0,10H 1I	0EC8	10				
0E92	10			0EC9	54				
0E93	10			0ECA	38				
0E94	10			0ECB	54				
0E95	00			0ECC	10				
0E96	10			0ECD	00	2075	DB 0,10H,10H,7CH,10H,10H 1O		
0E97	00	2066	DB 0,26H,0,0,0,0 1P	0ECE	10				
0E98	28			0ECF	10				
0E99	00			0ED0	7C				
0E9A	00			0ED1	10				

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0 MODULE PAGE 45 SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0 MODULE PAGE 46

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
OED2 10		2076	DB 0,0,0,0,8,10H	OF09 3C		2085	DB 3CH,20H,38H,4,4,38H
OED3 00				OF0A 20			
OED4 00				OF0B 39			
OED5 00				OF0C 04			
OED6 00				OF0D 04			
OED7 08				OF0E 38			
OED8 10				OF0F 18		2086	DB 18H,20H,38H,24H,24H,18H,16
OED9 00		2077	DB 0,0,0,7CH,0,0	OF10 20			
OEDA 00				OF11 38			
OEDB 00				OF12 24			
OEDC 7C				OF13 24			
OEDD 00				OF14 18			
OEDE 00				OF15 3C		2087	DB 3CH,4,8,10H,20H,20H
OEDF 00		2078	DB 0,0,0,0,0,10H	OF16 04			
OEE0 00				OF17 08			
OEE1 00				OF18 10			
OEE2 00				OF19 20			
OEE3 00				OF1A 20			
OEE4 10				OF1B 18			
OEE5 04		2079	DB 4,4,8,10H,20H,20H	OF1C 24		2088	DB 18H,24H,18H,24H,24H,18H,18
OEE6 04				OF1D 18			
OEE7 08				OF1E 24			
OEE8 10				OF1F 24			
OEE9 20				OF20 18		2089	DB 18H,24H,1CH,4,4,18H
OEEA 20				OF21 18			
OEEB 38		2080	DB 38H,4CH,54H,54H,64H,38H,10	OF22 24			
OEEC 4C				OF23 1C			
OEEE 54				OF24 04			
OEEF 64				OF25 04			
OEF0 38				OF26 18			
OEF1 10		2081	DB 10H,30H,50H,10H,10H,38H,11	OF27 00		2090	DB 0,10H,0,10H,0,0
OEF2 30				OF28 10			
OEF3 50				OF29 00			
OEF4 10				OF2A 10			
OEF5 10				OF2B 00			
OEF6 38				OF2C 00			
OEF7 18		2082	DB 18H,24H,8,10H,20H,3CH,12	OF2D 00		2091	DB 0,10H,0,10H,20H,0
OEF8 24				OF2E 10			
OEF9 08				OF2F 00			
Oefa 10				OF30 10			
OEFB 20				OF31 20			
OEFc 3C				OF32 00			
OEFd 38		2083	DB 38H,4,18H,4,4,38H,13	OF33 00		2092	DB 0,18H,20H,40H,20H,18H,1C
OEFf 04				OF34 18			
OF00 04				OF35 20			
OF01 04				OF36 40			
OF02 38				OF37 20			
OF03 0C		2084	DB 0CH,14H,24H,3CH,4,4,14	OF38 18		2093	DB 0,0,7CH,0,7CH,0
OF04 14				OF39 00			
OF05 24				OF3A 00			
OF06 3C				OF3B 7L			
OF07 04				OF3C 00			
OF08 04				OF3D 7C			
				OF3E 00			
				OF3F 00		2094	DB 0,30H,8,4,8,30H

MFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 47

LOC	OBJ	LINE	SOURCE STATEMENT
0F40	35		
0F41	08		
0F42	04		
0F43	08		
0F44	30		
0F45	18	2095	DB 18H,24H,8,10H,0,10H ;7
0F46	24		
0F47	08		
0F48	10		
0F49	00		
0F4A	10		
		2096	;
		2097	;
		2098	;SUBROUTINA SCAN
		2099	;
		2100	;PREIA IN REG. A UN CARACTER DE LA TASTATURA
		2101	;ASTEAPTA INTRODUCERE CARACTER
		2102	;PORT A ESTE PORTUL DE RETURN
		2103	;PORT B ESTE PORTUL PT SHIFT-LOCK,CTRL,SHIFT
		2104	;PORT C ESTE PORTUL DE SCANARE
		2105	;
0020		2106	PORTA EQU 20H ;PORT A 8255
0021		2107	PORTB EQU 21H ;PORT B 8255
0022		2108	PORTC EQU 22H ;PORT C 8255
0F4B	C5	2109	SCAN: PUSH B ;SALVEAZA REGISTRE
0F4C	D5	2110	PUSH D
0F4D	E5	2111	PUSH H
0F4E	216F61	2112	BR: LXI H,MCAP ;ADR.LOCATIE CE MEMOREAZA POZITIE SHIFT
0F51	DB21	2113	IN PORTB
0F53	2F	2114	CMA
0F54	17	2115	RAL
0F55	47	2116	MOV B,A ;SALVEAZA CTRL SI SHIFT
0F56	D2620F	2117	JNC SHIF ;SALT DACA NU E SHIFT-LOCK
0F59	7E	2118	MOV A,M
0F5A	2F	2119	CMA ;COMPLEMENTEAZA CONTINUTUL LOCATIEI MCAP
0F5B	77	2120	MOV M,A
0F5C	DB21	2121	TEST: IN PORTB ;TEST ELIBERARE SHIFT-LOCK
0F5E	17	2122	RAL
0F5F	D25C0F	2123	JNC TEST ;SALT DACA TASTA APASATA
0F62	3A6F61	2124	SHIF: LDA MCAP
0F65	A7	2125	ANA A
0F66	CA700F	2126	JZ SAL ;SALT LA SHIFT NORMAL
0F69	78	2127	MOV A,B
0F6A	17	2128	RAL
0F6B	17	2129	RAL
0F6C	3F	2130	CMC ;COMPLEMENTEAZA BITUL PT SHIFT
0F6D	1F	2131	RAR
0F6E	1F	2132	RAR
0F6F	47	2133	MOV B,A
0F70	78	2134	SAL: MOV A,B
0F71	327061	2135	STA SHCT ;SALVEAZA CTRL,SHIFT
0F74	0E5F	2136	MVI C,95 ;AFISEAZA CURSOR
0F76	CDD810	2137	CALL SCRUI
0F79	0600	2138	MVI B,0
0F7B	CDBF12	2139	CALL BITW ;ASTEAPTA

3FDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 48

LOC	OBJ	LINE	SOURCE STATEMENT	
OF7E	0E20	2140	MVI C, BLANC	;STERGE CURSOR
OF80	CDD810	2141	CALL SCRUI	
OF83	0600	2142	MVI B, 0	
OF85	CDBF12	2143	CALL BITW	;ASTEAPTA
OF88	3A7661	2144	LDA EINV	;VIDEO INVERS ECRAN
OF8B	E6F8	2145	ANI OF8H	;CONTOR SCANARE
OF8D	47	2146	MOV B, A	
OF8E	CD90F	2147	CONT1: CALL TSTAS	;TEST LINIE DE SCANARE
OF91	2F	2148	CMA	
OF92	A7	2149	ANA A	
OF93	C2A40F	2150	JNZ TASAP	;SALT LA TASTA APASATA
OF96	04	2151	FALS: INR B	
OF97	78	2152	MOV A, B	
OF98	E607	2153	ANI 7	
OF9A	C28E0F	2154	JNZ CONT1	;SALT DACA MAI SINT LINII DE SCANAT
OF9D	AF	2155	XRA A	
OF9E	327061	2156	STA SHCT	;INITIALIZARE
OFA1	C34E0F	2157	JMP BR	;REIA
OFA4	0E07	2158	TASAP: MVI C, 7	;CONTOR RETURN
OFA6	17	2159	CIC1: RAL	
OFA7	DAAE0F	2160	JC CONEX	;SALT DECA S-A GASIT LINIA DE RETURN
OFAA	0D	2161	DCR C	
OFAB	F2A60F	2162	JP CIC1	;REIA
OFAE	CD90F	2163	CONEX: CALL TSTAS	;TEST LINIE DE SCANARE
OFB1	2F	2164	CMA	
OFB2	A7	2165	ANA A	
OFB3	C2AE0F	2166	JNZ CONEX	;ASTEAPTA ELIBERARE TASTA
OFB6	21FF0F	2167	LXI H, SIMB	;ADR TABELA DE SIMBOLI
OFB7	AF	2168	XRA A	
OFBA	78	2169	MOV A, B	;CALCUL DEPLASARE
OFBB	E607	2170	ANI 7	
OFBD	17	2171	RAL	
OFBE	17	2172	RAL	
OFBF	17	2173	RAL	
OFC0	B1	2174	ORA C	
OFC1	4F	2175	MOV C, A	
OFC2	3A7061	2176	LDA SHCT	;CTRL SI SHIFT
OFC5	E640	2177	ANI 40H	;TINE SHIFT
OFC7	B1	2178	ORA C	;DEPLASARE FINALA
OFC8	4F	2179	MOV C, A	
OFC9	0600	2180	MVI B, 0	
OFCB	09	2181	DAD B	;ADRESA ABSOLUTA SIMBOL IN H,L
OFC4	7E	2182	MOV A, M	;SIMBOL IN REG A
OFCD	47	2183	MOV B, A	
OFCE	3A7061	2184	LDA SHCT	
OFD1	17	2185	RAL	
OFD2	D2D90F	2186	JNC NCOR	;SALT DACA NU E NEVOIE DE CORECTIE
OFD5	78	2187	MOV A, B	
OFD6	E63F	2188	ANI 3FH	;CORECTIE SIMBOL
OFD8	47	2189	MOV B, A	
OFD9	78	2190	NCOR: MOV A, B	;SIMBOL CORECTAT IN REG A
OFDA	F5	2191	PUSH PSW	
OFDB	0E10	2192	MVI C, 10H	;MARTOR SONOR TASTA
OFDD	3A7661	2193	BIP1: LDA EINV	
OFEO	47	2194	MOV B, A	

SFDX-18 8080/8085 MACRO ASSEMBLER. V3.0

MODULE PAGE 49

LOC	OBJ	LINE	SOURCE STATEMENT
0FE1	D322	2195	OUT PORTC
0FE3	CDBF12	2196	CALL BITW
0FE6	3A7661	2197	LDA EINV
0FE9	EE08	2198	XRI 8 ;COMPLEMENTEAZA BIT 3
0FEB	D322	2199	OUT PORTC
0FED	CDBF12	2200	CALL BITW
0FF0	0D	2201	DCR C
0FF1	C2DD0F	2202	JNZ BIP1
0FF4	F1	2203	POP PSW ;REFACE REGISTRELE SI INDICATORII DE CONDITII
0FF5	E1	2204	POP H
0FF6	D1	2205	POP D
0FF7	C1	2206	POP B
0FF8	C9	2207	RET
		2208	;TEST LINIE SCANARE
0FF9	78	2209	TSTAS: MOV A,B
0FFA	D322	2210	OUT PORTC
0FFC	DB20	2211	IN PORTA
0FFE	C9	2212	RET
		2213	;TABELA DE SIMBOLI
0FFF	1B	2214	SIMB: DB 1BH, '1234567890-=-\', 8, 20H
1000	31323334		
1004	35363738		
1008	3930203D		
100C	5C		
100D	08		
100E	20		
100F	09515745	2215	DB ' QWERTYUIOP[', 5CH, 0AH, 7FH, 20H
1013	52545955		
1017	494F505B		
101B	5C		
101C	0A		
101D	7F		
101E	20		
101F	41534446	2216	DB 'ASDFGHIJKL ', 27H, 0DH, '
1023	47484A4B		
1027	4C3B		
1029	27		
102A	0D		
102B	20202020		
102F	5A584356	2217	DB 'ZXCVBNM, ,'
1033	424E4D2C		
1037	2E2F2020		
103B	20202020		
103F	1B	2218	DB 1BH, '(0#%*!@()-+\'', 8, 20H
1040	21402324		
1044	255E242A		
1048	28292D2B		
104C	5C		
104D	08		
104E	20		
104F	09	2219	DB 9, 71H, 77H, 65H, 72H, 74H, 79H, 75H
1050	71		
1051	77		
1052	65		
1053	72		

SFDX-18 800078005 MACRO ASSEMBLER, V3.0 MODULE PAGE 50

LOC	OBJ	LINE	SOURCE STATEMENT
1054	74		
1055	79		
1056	75		
1057	69	2220	DB 69H, 6FH, 70H, 5DH, 21H, 0AH, 7FH, 20H
1058	6F		
1059	70		
105A	5D		
105B	21		
105C	0A		
105D	7F		
105E	20		
105F	61	2221	DB 61H, 73H, 64H, 66H, 67H, 68H, 6AH, 6BH
1060	73		
1061	64		
1062	66		
1063	67		
1064	68		
1065	6A		
1066	6B		
1067	6C	2222	DB 6CH, 3AH, 22H, 0DH, 20H, 20H, 20H, 20H
1068	3A		
1069	22		
106A	0D		
106B	20		
106C	20		
106D	20		
106E	20		
106F	7A	2223	DB 7AH, 78H, 63H, 76H, 62H, 6EH, 6DH, 3CH
1070	78		
1071	63		
1072	76		
1073	62		
1074	6E		
1075	6D		
1076	3C		
1077	3E	2224	DB 3EH, 3FH, 20H
1078	3F		
1079	20		
		2225	;
		2226	;-----
		2227	;SUBROUTINA AFIS
		2228	;
		2229	;AFISEAZA CARACTER LA TELEVIZOR
		2230	;SALVEAZA SI REFACI H,L,D,E,B,C
		2231	;CARACTERUL IN REGISTRUL C
		2232	;AFISEAZA CAR INTRE BLANC SI BARA JOS
		2233	;PRODUCE DEFILARE(=CROL)
		2234	;RECUNDASTE LINE PEED SI TAB=4 BLANCURI
		2235	;
107A	E5	2236	AFIS: PUSH H
107B	D5	2237	PUSH B
107C	C5	2238	PUSH B
107D	79	2239	MOV A,C
107E	FE0D	2240	CPI CR ;REG A=0AR DE AFISAT
1080	C28A10	2241	JNZ URM10 ;CR?

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 51

LOC	OBJ	LINE	SOURCE STATEMENT
1083	AF	2242	XRA A ;COL=0
1084	326E61	2243	STA COL
1087	C3D710	2244	JMP REF1
108A	FE20	2245 URM10:	CPI BLANC ;BLANC?
108C	C29A10	2246	JNZ URM7
108F	0E20	2247	MVI C,BLANC ;REG C=CAR DE AFISAT
1091	CDD810	2248	CALL SCRUI ;AFISEAZA BLANC
1094	CD5C11	2249	CALL MODDR ;MUTA POINTERI TV CU O POZITIE LA DREAPTA
1097	C3D710	2250	JMP REF1
109A	FE0A	2251 URM7:	CPI LF ;LINE FEED?
109C	C2BD10	2252	JNZ URM8
109F	216D61	2253	LXI H,LIN ;CITIRE LINIE CURENTA
10A2	7E	2254	MOV A,M ;DE CARACTERE
10A3	3C	2255	INR A
10A4	77	2256	MOV M,A
10A5	FE20	2257	CPI 32 ;A FOST LINIA 32 ?
10A7	C2D710	2258	JNZ REF1
10AA	35	2259	DCR M
10AB	3A7761	2260	LDA AFMOD ;TESTEAZA MODUL DE AFISARE
10AE	B7	2261	ORA A
10AF	CAB710	2262	JZ URM71
10B2	3600	2263	MVI M,0
10B4	C3D710	2264	JMP REF1
10B7	CD7F11	2265 URM71:	CALL SCROL
10BA	C3D710	2266	JMP REF1
10B0	FE05	2267 URM8:	CPI CTRLR ;COD VIDEO INVERS
10BF	C20C10	2268	JNZ URM9
10C2	3A7561	2269	LDA VINV ;00=DIRECT, FF=INVERS
10C3	2F	2270	CMA ;COMPLEMENTEAZA
10C6	327561	2271	STA VINV ;ACTUALIZEAZA
10C9	C3D710	2272	JMP REF1
10CC	FE10	2273 URM9:	CPI 10H ;REJECTEAZA CAR NEIMPRIMABILE
10CE	DAD710	2274	JC REF1 ;COD ASCII<COD BLANC
10D1	CDD810	2275	CALL SCRUI ;AFISEAZA CARACTER IMPRIMABIL
10D4	CD5C11	2276	CALL MODDR ;MODIFICA POINTERI TV CU O
10D7	C1	2277 REF1:	POP B ;POZITIE LA DREAPTA
10D8	D1	2278	POP D
10D9	E1	2279	POP M
10DA	C9	2280	RET
		2281	;
		2282	;
		2283	;SUBROUTINA SCRUI
		2284	;
		2285	;SCRIE CARACTER LA TV DIN REGISTRUL C
		2286	;FORMAT ECRAN = 32 LINII DE 30 DE CARACTERE
		2287	;FORMAT CAR = 5*6 PCTE IN CADRU DE 8*8
		2288	;O LINIE DE CAR = 8 LINII TV
		2289	;TOTAL MEMTV *256 LINII TV * 256 PCTE
		2290	;MEM TV ARE CUV DE 8 BITI,DECI FIECARE OCTET SE TRIMITE
		2291	;IN TRANSE DE CITE 8 BITI
		2292	;PE TV '1' = NEGRU '0' = ALB
		2293	;
6200		2294	LITL EQU 6200H ;TABELA DE CARACTERE MICI ORGANIZATA DE UTILIZATOR
		2295	;
6240		2296	SEMI EQU 6240H ;TAB DE CARACTERE SEMIGRAFICE ORGANIZATA DE UTILIZATOR

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 52

LOC	OBJ	LINE	SOURCE STATEMENT
		2297	
4001		2298	BEGTV EQU BAZTV+1 ;CODURI INTRE 10H - 1FH
10DB 79		2299	SCR10: MOV A,C ;COD ASCII PE 6 BITI
10DC F5		2300	PUSH PSW
10DD FE20		2301	CPI 20H ;10H<=CAR<=20H
10E# DA2911		2302	JC SCR52
10E2 FE60		2303	CPI 60H
10E4 DAEF10		2304	JC SCR50
10E7 D660		2305	SUI 60H
10E9 210062		2306	LXI H,LITL
10EC C3F410		2307	JMP SCR51
10EF E63F		2308	SCR50: ANI 3FH ;STERGE 2 BITI C.M.S.
10F1 21CB0D		2309	LXI H,BAZA ;BAZA GENERATORULUI DE CARACTERE
10F4 010600		2310	SCR51: LXI B,6 ;INCREMENT ADRESA GENERATOR DE CAR
10F7 B7		2311	DRA A ;TEST A=0 ?
10FB CA0011		2312	JZ SCR11
10FB 09		2313	SCR22: DAD B
10FC 3D		2314	DCR A
10FD C2FB10		2315	JNZ SCR22
1100 227161		2316	SCR11: SHLD ADEC
1103 216D61		2317	LXI H,LIN ;CALCULEAZA ADR DIN MEMTV
1106 4A		2318	MOV B,M
1107 216E61		2319	LXI H,COL
110A 4E		2320	MOV C,M
110B 210140		2321	LXI H,BEGTV
110E 09		2322	DAD B
110F F1		2323	POP PSW
1110 FE20		2324	CPI 20H
1112 DA3C11		2325	JC SCR54
1115 112000		2326	LXI D,32
1118 3A7561		2327	LDA VINV
111B 2F		2328	CMA
111C 77		2329	MOV M,A
111D 19		2330	DAD D
111E 0E06		2331	MVI C,6
1120 CD4211		2332	CALL SCR56
1123 3A7561		2333	LDA VINV
1126 2F		2334	CMA
1127 77		2335	MOV M,A ;SEPARATOR DE LINII
1128 C9		2336	RET
1129 214062		2337	SCR52: LXI H,SEMIO
112C 010800		2338	LXI B,8
112F D610		2339	SUI 10H
1131 CA0011		2340	JZ SCR11
1134 09		2341	SCR53: DAD B
1135 3D		2342	DCR A
1136 C23411		2343	JNZ SCR53
1139 C30011		2344	JMP SCR11
113C 0E08		2345	SCR54: MVI C,8
113E CD4211		2346	CALL SCR56
1141 C9		2347	RET
1142 E5		2348	SCR56: PUSH H
1143 2A7161		2349	LHLD ADEC
1146 EB		2350	XCHG
1147 E1		2351	POP H ;D,E=ADRESA GENERATOR DE CARACTERE

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 53

LOC	OBJ	LINE	SOURCE STATEMENT
1148	3A7561	2352	SCR55: LDA VINV
114B	47	2353	MOV B,A
114C	1A	2354	LDAX D
114D	2F	2355	CMA
114E	A8	2356	XRA B
114F	77	2357	MOV M,A
1150	13	2358	INX D
1151	D5	2359	PUSH D
1152	112000	2360	LXI D,32
1155	19	2361	DAD D
1156	D1	2362	POP D
1157	0D	2363	DCR C
1158	C24811	2364	JNZ SCR55
115B	C9	2365	RET
		2366	;MUTA POINTERUL TV CU O POZITIE LA DR.
115C	216E61	2367	MODDR: LXI H,COL
115F	7E	2368	MOV A,M
1160	3C	2369	INR A
1161	77	2370	MOV M,A
1162	FE1E	2371	CPI 30
1164	CO	2372	RNZ
1165	3600	2373	MVI M,0
1167	216D61	2374	LXI H,LIN
116A	7E	2375	MOV A,M
116B	3C	2376	INR A
116C	77	2377	MOV M,A
116D	FE20	2378	CPI 32
116F	CO	2379	RNZ
1170	35	2380	DCR M
1171	3A7761	2381	LDA AFMOD
1174	B7	2382	ORA A
1175	CA7B11	2383	JZ MOD11
1178	3600	2384	MVI M,0
117A	C9	2385	RET
117B	CD7F11	2386	MOD11: CALL SCROL
117E	C9	2387	RET
		2388	;EFFECT:DEFILARE
117F	210040	2389	SCROL: LXI H,4000H
1182	110041	2390	LXI D,4100H
1185	1A	2391	SCR2: LDAX D
1186	77	2392	MOV M,A
1187	23	2393	INX H
1188	13	2394	INX D
1189	7B	2395	MOV A,E
118A	B7	2396	ORA A
118B	C26511	2397	JNZ SCR2
118E	7A	2398	MOV A,D
118F	FE40	2399	CPI 60H
1191	C26511	2400	JNZ SCR2
1194	21003F	2401	LXI H,3F00H
1197	36FF	2402	SCR1: MVI M,OFFH
1199	23	2403	INX H
119A	7C	2404	MOV A,H
119B	FE40	2405	CPI 60H
119D	C29911	2406	JNZ SCR1

WFOX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 54

LOC	OBJ	LINE	SOURCE STATEMENT
11A0	C9	2407	RET
		2408	!RUTINE DE LUCRU CU CASETOFOFONUL
		2409	!
		2410	!-----
		2411	!COMANDA STORE
		2412	!-----
		2413	!SINTAXA: K ADRINF,ADRSUP(CR)
		2414	!SALVEAZA PE CASETA ZONA DE MEMORIE DINTRE ADRINF SI ADRSUP
		2415	!OBS:ACELEASI CA LA CDA DISPLAY
		2416	!
11A1	110000	2417	STAPE: LXI D,0
11A4	0630	2418	PRAMB: MVI B,30H
11A6	CD7D12	2419	CALL IMPUL
11A9	13	2420	INX D
11AA	7A	2421	MOV A,D
11AB	FE20	2422	CPI 20H
11AD	C2A411	2423	JNZ PRAMB
11B0	060A	2424	MVI B,0AH
11B2	CD7D12	2425	CALL IMPUL
11B5	0E00	2426	MVI C,0
11B7	1604	2427	MVI D,4
11B9	212D60	2428	LXI H,MAXL
11BC	7E	2429	ECK: MOV A,H
11BD	CD5212	2430	CALL CKSMO
11C0	23	2431	INX H
11C1	15	2432	DCR D
11C2	C2BC11	2433	JNZ ECK
11C5	2A2B60	2434	LHLD EOFP
11C8	EB	2435	KCHG
11C9	2A2960	2436	LHLD BOFP
11CC	7B	2437	MOV A,E
11CD	95	2438	SUB L
11CE	3F	2439	MOV E,A
11CF	7A	2440	MOV A,D
11D0	9C	2441	SBB H
11D1	57	2442	MOV D,A
11D2	CD5212	2443	CALL CKSMO
11D3	7B	2444	MOV A,E
11D6	CD5212	2445	CALL CKSMO
11D9	2B	2446	DCX H
11DA	23	2447	TAPE1: INX H
11DB	7E	2448	MOV A,H
11DC	CD5212	2449	CALL CKSMO
11DF	7A	2450	MOV A,D
11E0	B3	2451	ORA E
11E1	1B	2452	DCX D
11E2	C2DA11	2453	JNZ TAPE1
11E5	79	2454	MOV A,C
11E6	2F	2455	ORA
11E7	3C	2456	INR A
11E8	CD5212	2457	CALL CKSMO
11EB	CD5212	2458	CALL CKSMO
11EE	C9	2459	RET
		2460	!
		2461	!-----

!INITIALIZARE SUMA DE CONTROL

!NUMARUL DE LINIE MAXIM

!SFIRSIT FISIER

!INCEPUT FISIER

!CITESTE OCTET

!SALVEAZA-L PE CASETA

!REIA PINA LA CONTOR NUL

!SCRIE PE CASETA SUMA DE CONTROL

!IN COMPLEMENT FATA DE 2

SFDX-18 8080/8085 MACRO ASSEMBLER, V3.0

MODULE PAGE 55

LOC	OBJ	LINE	SOURCE STATEMENT
		2462	;COMANDA LOAD
		2463	;-----
		2464	;SINTAXA: L ADR(CR) SAU L(CR)
		2465	;CITESTE DE PE CASETA IN MEMORIE, FIE LA ADRESA 'ADR' LA PRIMA FORMA
		2466	;A CZII SI LA ADRESA CITITA DE PE CASETA LA A DOUA FORMA
		2467	;
11EF	DB21	2468	LTAPE: IN 21H
11F1	47	2469	MOV B,A
11F2	DB21	2470	SRII1: IN 21H
11F4	A8	2471	XRA B
11F5	CAF211	2472	JZ SRII1
11F8	DB21	2473	SRII2: IN 21H
11FA	E601	2474	ANI 1
11FC	C2FB11	2475	JNZ SRII2
11FF	DB21	2476	SRII3: IN 21H
1201	E601	2477	ANI 1
1203	CAFF11	2478	JZ SRII3
1206	CDB212	2479	CALL BITR
1209	3E1D	2480	MVI A,1DH ;ACC,B=DURATA IMPULS, CY=1
120B	B8	2481	CHP B
120C	DAFF11	2482	JC SRII3
120F	OE00	2483	MVI C,0 ;SUMA DE CONTROL
1211	212D60	2484	LXI H,MAXL
1214	1604	2485	MVI D,4
1216	CD5A12	2486	IEK: CALL CKSMI
1219	77	2487	MOV M,A
121A	23	2488	INX H
121B	15	2489	DCR D
121C	C21612	2490	JNZ IEK
121F	2A2960	2491	LHLD BOFP
1222	CD5A12	2492	CALL CKSMI
1225	57	2493	MOV D,A
1226	CD5A12	2494	CALL CKSMI
1229	5F	2495	MOV E,A
122A	2B	2496	DCX H
122B	23	2497	TAPE2: INX H
122C	CD5A12	2498	CALL CKSMI ;CITESTE OCTET
122F	77	2499	MOV M,A
1230	7A	2500	MOV A,D
1231	B3	2501	ORA E
1232	1B	2502	DCX D
1233	C22B12	2503	JNZ TAPE2 ;REIA PINA LA CONTOUR MUL
1236	222B60	2504	SHLD EOFP
1239	CD5A12	2505	CALL CKSMI
123C	C8	2506	RZ
123D	214412	2507	LXI H,ERMES
1240	CD5402	2508	CALL SCRN
1243	C9	2509	RET
1244	20524541	2510	ERMES: DB ' READ ERROR ',0DH
1248	44202045		
124C	52524F52		
1250	20		
1251	0D		
		2511	;CALCULEAZA SUMA DE CONTROL LA SCRIERE
1252	F9	2512	CKSMO: PUSH PSM

LOC	OBJ	LINE	SOURCE STATEMENT	LOC	OBJ	LINE	SOURCE STATEMENT
1253 81		2513	ADD C	12AB ID		2568	DCR E
1254 4F		2514	MOV C,A	12AC C29212		2569	JNZ SRII7
1255 F1		2515	POP PSW	12AF C1		2570	POP B
1256 CD4212		2516	CALL SRIOT	12B0 D1		2571	POP D
1259 C9		2517	RET	12B1 C9		2572	RET
125A ED8D12		2518	:CALCULEAZA SUMA DE CONTROL LA CITIRE	12B2 DB21		2573	IN 21H
125D 47		2519	CKSMI: CALL SRIIN	12B4 4F		2574	MOV C,A
125E 81		2520	MOV B,A	12B5 0600		2575	MVI B,0
125F 4F		2521	ADD C	12B7 04		2576	BITRI: INR B
1260 78		2522	MOV C,A	12B8 A9		2577	IN 21H
1261 C9		2523	MOV A,B	12B9 CAB712		2578	XRA C
		2524	RET	12BE C9		2579	JZ BITRI
1262 D5		2525	:SCRIE OCTET PE CASETA	2580		2580	RET
1263 1E08		2526	SRIOT: PUSH D	2581 BITW:		2581	IN 21H
1265 57		2527	MVI E,8	12C1 A8		2582	XRA B
1266 E890		2528	MOV D,A	12C2 05		2583	DCR B
1268 CA7012		2529	:SALVARE OCTET IN D	12C3 C2BF12		2584	JNZ BITW
126B 0622		2530	ANI 80H	2585		2585	RET
126D 0622		2531	JZ SRI01	2586		2586	:FECHVALARI SI REZERVARI DE MEMORIE
126E 0622		2532	MVI B,22H	2587		2587	:FECHVALARI SI REZERVARI DE MEMORIE
1270 060E		2533	JMP SRI02	2588		2588	:FECHVALARI SI REZERVARI DE MEMORIE
1272 CD7D12		2534	:BIT=0	2589		2589	:FECHVALARI SI REZERVARI DE MEMORIE
1275 7A		2535	SRI02: CALL IMPUL	6000		6000	ORG 6000H
1276 07		2536	MOV A,D	6008		2590	NBR EQU 8
1277 07		2537	RLC	6000		2591	HOLD: DS 12
1278 C26512		2538	DCR E	600C		2592	BRT: DS 3*NBR
127B D1		2539	JNZ SRI03	0006		2593	MAXFIL EQU 6
127C C9		2540	POP D	0000		2594	NMLEN EQU 5
127D C5		2541	RET	6024		2595	FELEN EQU NMLEN*8
127E 3EFF		2542	IMPUL: PUSH B	602B		2596	FILEO: DS 2
1280 D322		2543	MVI A,OFFH	602D		2597	ROFF: DS 2
1282 CD6F12		2544	OUT 22H	6031		2598	EOFF: DS 2
1285 C1		2545	CALL BITW	6072		2600	FILTB: DS 4
1286 AF		2546	POP B	6072		2601	INSP: DS 2
1287 D322		2547	XRA A	6072		2602	DELP EQU INSP
1289 CD6F12		2548	OUT 22H	6074		2603	ASCR EQU 13
128C C9		2549	CALL BITW	6074		2604	HCON: DS 2
		2550	RET	6074		2605	ADDS EQU HCON
128D D5		2551	:CITESTE OCTET DE PE CASETA	6076		2606	FBUF: DS NMLEN
128E C5		2552	SRIIN: PUSH D	607B		2607	FREAD: DS 2
128F 1E08		2553	PUSH B	607D		2608	FEF: DS 1
1291 AF		2554	MVI E,8	607E		2610	ABUF: DS 12
1292 07		2555	XRA A	608A		2611	BBUF: DS 4
1293 57		2556	RLC	608F		2612	SCNT: DS 1
1294 DB21		2557	MOV D,A	000F		2613	DCNT: DS 1
1296 E601		2558	21H IN	6090		2614	NCOM EQU 15
1298 CA9412		2559	:VALCARE DE PRAG	6092		2615	TABA: DS 2
129B CD8212		2560	SRII4: CALL BITR	6094		2616	ASPC: DS 2
129E 3E18		2561	MVI A,18H	6095		2617	PASI: DS 1
12A0 88		2562	CMP B	6096		2618	NCHR: DS 1
12A1 DA812		2563	JC SRII5	6098		2619	PNTR: DS 2
12A4 AF		2564	XRA A	6099		2620	NOLA: DS 1
12A5 C3AA12		2565	JMP SRII6	6099		2621	SIGN: DS 1
12A8 3E01		2566	MVI A,1	609A		2622	OPRD: DS 2
12AA 82		2567	SRII6: ORA D				

SFDX-18 8080/8085 MACRO ASSEMBLER V1.0

MODULE PAGE 59

DEL4	A 0647	DEL5	A 0648	DELL	A 05EE	DELP	A 6072	A 0199	DOUY	A 022B	DUM1	A 030C
DUMP	A 0306	DUMS	A 0309	EAF	A 06DD	EASS	A 092E	A 11BC	ECLER	A 0035	EGR	A 008E
EEND	A 002F	EGT	A 0535	EINV	A 6176	ELT	A 0522	A 0470	EMES1	A 0475	EMES2	A 047A
ENT1	A 0499	ENTR	A 0480	ENTS	A 048D	E01	A 057E	A 057D	EOFP	A 602B	EOR	A 0059
EPAGE	A 032C	EQU1	A 07A2	EQU2	A 0827	ERUL	A 050D	A 07AD	ERMES	A 1244	ERO1	A 0CE6
ERRA	A 0CD5	ERRD	A 0CF4	ERRL	A 0CEF	ERRM	A 00CC	A 0CDA	ERRR	A 0CAE	ERRS	A 0CE7
ERRU	A 0CC2	ERRV	A 0CC7	ETRA	A 0154	EXEC	A 00FB	A 0F96	FAST	A 03F0	FBUF	A 6076
FEET	A 0402	FEF	A 607D	FELEN	A 000D	F11	A 0564	A 0576	FIL30	A 03BA	FIL35	A 03BC
FILE	A 0348	FILE0	A 6024	FILTB	A 6031	FIN1	A 055C	A 0556	FINE	A 03D3	FOCNT	A 607D
FOOD	A 03ED	FOOL	A 040E	FOOT	A 03CD	FOUL	A 03CF	A 042B	FREAD	A 607B	FSE10	A 042B
FSE15	A 044E	FSE20	A 0458	FSEA	A 042D	HOON	A 6074	A 6000	HOTB	A 0224	HOUT	A 0214
IBUF	A 611A	IEK	A 1216	IMPUL	A 127D	IN8	A 00E0	A 0139	INIT2	A 0020	INIT3	A 002F
INITA	A 0019	INSP	A 6072	INSR	A 0504	LCHK	A 0762	A 000A	LICK	A 04CB	LIN	A 016D
LINE	A 04BF	LIST	A 05DA	LISTO	A 05E0	LITLE	A 6200	A 0005	LMOV	A 058A	LODM	A 059C
LTAPE	A 11EF	MAXFIL	A 0006	MAXL	A 602D	MCAP	A 616F	A 046A	MLAB	A 0732	MOD11	A 117B
MODDR	A 115C	MOV23	A 038C	MPNT	A 08EC	NBR	A 0008	A 6095	NCOH	A 000F	NCOR	A 0FD9
NEXT	A 007F	MMLEN	A 0005	NOLA	A 6098	NOR1	A 05CD	A 05C7	NOV1	A 0656	NOVR	A 0650
NUM1	A 0CA4	NUM2	A 0CAB	NUMS	A 0C90	NXT1	A 0BCE	A 08D8	OBUF	A 6105	OCN1	A 0B22
OCN2	A 0B25	OCNT	A 0B0E	OERR	A 0B33	OIND	A 609E	A 0AAE	OP2	A 0AB1	OP4	A 0B09
OP5	A 0B0C	OPAD	A 0B08	OFC	A 074B	OFC2	A 0AE6	A 0AD2	OPCD	A 0A92	OPRD	A 609A
OPRI	A 609C	ORG1	A 078D	ORG2	A 0839	OTAB	A 0954	A 00E7	P1	A 0DBB	PABL	A 07DA
PAG1	A 033F	PAS1	A 0709	PAS2	A 07C0	PASI	A 6094	A 6096	PORTA	A 0020	PORTB	A 0021
PORTC	A 0022	PRAMB	A 11A4	PROC	A 0DAE	PSEU	A 0B3B	A 0771	PSU2	A 07F5	READ	A 0077
REF1	A 10D7	REIA	A 003F	RES1	A 07B5	RES2	A 080E	A 081A	RMOV	A 0593	ROOM	A 0382
RTAB	A 0B8D	SAL	A 0F70	SBL1	A 093D	SBL2	A 0941	A 093A	SCAN	A 0F4B	SCNT	A 608E
SCR1	A 1197	SCR11	A 1100	SCR2	A 1185	SCR22	A 10FB	A 10EF	SCR51	A 10F4	SCR52	A 1129
SCR53	A 1134	SCR54	A 113C	SCR55	A 1148	SCR56	A 1142	A 10DB	SCRN	A 0254	SCROL	A 117F
SEAR	A 012D	SEMIG	A 6240	SEN1	A 0C8A	SEND	A 0C80	A 04FC	SHCT	A 6170	SHIF	A 0F62
SIGN	A 6099	SIMB	A 0FFF	SLA1	A 0B71	SLA2	A 0B97	A 088A	SLAB	A 0B4D	SRI11	A 11F2
SRI12	A 11F8	SRI13	A 11FF	SRI14	A 1294	SRI15	A 12A8	A 12AA	SRI17	A 1292	SRIIN	A 128D
SRI01	A 1270	SRI02	A 1272	SRI03	A 1265	SRIOT	A 1262	A 0C3C	STAI	A 0262	STAI1	A 031D
STAPE	A 11A1	STAR	A 003E	START	A 0000	STOM	A 05A4	A 039A	SWAPS	A 0395	SYMT	A 617C
TAB	A 0009	TABA	A 6090	TAPE1	A 11DA	TAPE2	A 122B	A 0FA4	TEMP	A 609D	TEST	A 0F5C
TEST1	A 036D	TREC	A 0268	TRED1	A 0323	TSTAS	A 0FF9	A 0882	TY32	A 0885	TY41	A 08AE
TY56	A 08DA	TY6	A 090B	TYP1	A 085A	TYP2	A 085E	A 0871	TYP4	A 0890	TYP5	A 08C5
TYP6	A 08FB	TY55	A 08CD	TYS6	A 090E	URM10	A 108A	A 109A	URM71	A 10B7	URMB	A 10BD
URM9	A 10CC	VAL1	A 0163	VAL2	A 017E	VAL3	A 018E	A 0190	VAL5	A 019E	VALC	A 014D
VCHK	A 02FE	VINV	A 6175	WHA1	A 0467	WHAT	A 0464	A 0146	ZBUF	A 0140	ZERG	A 0545

ASSEMBLY COMPLETE. NO ERRORS



VĂ RECOMANDĂM :

Din seria Automatică-Management-Calculatoare (AMC) au apărut în trimestrul III 1985 volumele :

AMC 48, AMC 49, AMC 50, AMC 51, în cuprinsul cărora sînt prezente module și cicluri de foarte mare actualitate, de înaltă calitate și de un interes deosebit, și anume :

– Congresul mondial trienal al Federației Internaționale de Automatizare (IFAC) „O punte între știință și tehnologie”, Budapesta 1984, reprezentat prin plenary, studii de caz și sinteze pentru toate secțiunile (autori străini și români).

– „Societatea informatică” note de lectură după cartea japonezului Masuda, „Resursele informaționale naționale” și „Fenomenul calculatoarelor personale” după sovieticul Gromov.

– „Memento de teleprelucrare”, cu toate informațiile necesare pentru echipamentele și sistemele teletinformatică românești.

– „Minicalculatoarele INDEPENDENT și CORAL”. Manual de utilizare din ciclul SERVICE pentru CALCULATOARE.

– „BASIC pentru începători, cu calculatorul personal”, un manual practic din ciclul „CALCULATOARE PERSONALE ȘI PROGRAMAREA LOR”.

– Ciclul „PROIECTAREA ASISTATĂ DE CALCULATOR” reprezentat prin articole de direcționare în domeniu și articole prezentate la o primă sesiune națională.

– Microcalculatoarele personale românești, Student-HC 80, PRAE (pentru acesta și limbajul său BASIC) și microcalculatorul profesional-personal românesc Felix PC, în prezentări sintetice – în premieră într-o carte.

În trimestrul IV 1985 apar și volumele AMC 52-53-54, cu ciclurile amintite, dar și cu automatizarea flexibilă, roboții, limbajul BASIC pentru WANG VS, ghidul analistului (continuare la AMC 45-46), jocuri de întreprindere ș.a.

Prețul unui volum AMC este de aproximativ 25 lei.

Volumele AMC se găsesc în librării. Informații și la Editura Tehnică, Piața Științei 1. Telefon : 18 06 30 și 17 60 10/2100.



- Ce este aMIC-ul și de ce „Totul despre...”. Chiar... totul?
 - Calculatorul personal (individual) aMIC este primul calculator românesc cu ecran tele separat, destinat utilizării individuale pe scară largă în școlile de toate gradele, în unități de cercetare-proiectare, în gestiuni tehnico-economice curente, în activități de birou și chiar în cluburi și tabere, pentru jocuri distractiv-educative.
 - Autorii volumelor de față sînt cadre didactice și cercetători de la Facultatea de Automatică și Calculatoare din Institutul Politehnic București, care au conceput acest calculator, proiectanți și specialiști din unitățile care îl produc în serie, și anume Institutul pentru Tehnică de Calcul și Informatică – Filiala Timișoara și Fabrica de Memorii – Timișoara, cum și reprezentanți ai utilizatorilor, printre care un profesor emerit de la Liceul Industrial „Dimitrie Cantemir” din București și un elev de la același liceu.
 - Cărțile aMIC (volumul 1) și aMIC (volumul 2) ce apar simultan își propun să constituie manuale de prezentare-utilizare-operare indispensabile tuturor categoriilor de utilizatori.
 - O atenție deosebită este dată limbajului interactiv de programare BASIC-aMIC a cărui învățare este înlesnită de un număr mare de exemple.
- (continuare la vol. 2)

Vol. I și II Lei 42

