
MOJ MIKRO Slovenija

KAZALO

OPIS MIKRORACUNALNISKEGA SISTEMA
MOJ MIKRO SLOVENIJA 3

MIKRO SLOVENIJA UPORABNISKI PRIROCNIK 4

OPIS OSNOVNIH UKAZOV 4

VHODNO IZHODNE FUNKCIJE
MONITORJA 8

Obdelava prekinitev 8
Slikovni prikazovalnik 9
Vhod za paralelno tipkovnico 10
Krmilnik za gibke diske 10
Vmesnik za serijsko komunikacijo 10
Ura realnega časa 11
Paralelna vrata 11

Pregled sistemskih programov 11
Podroben opis funkcij 12
Tabela sistemskih spremenljivk 13

KAKO RACUNALNIK DELUJE 15

CENTRALNI PROCESOR 15

Generator urnih impulzov 15
Krmilnik za inicializacijo sistema 15
Ojačevanje sistemskega vodila 15
Druga pomnilna banka 16
Dekodiranje vhodno izhodnega prostora 16

Sinhronizacija prenosa podatkov
pri gibkih diskih 16

SLIKOVNI PRIKAZOVALNIK 17

Generator slikovnih impulzov 17
Pomik slike na zaslonu 17
Naslavljanje slikovnega pomnilnika 17
Generiranje sinhronizacijskih impulzov 17
Nastanek slike 17
Zatemnitev zaslona 18

POMNILNA BANKA 64 K IN PREKLAPLJANJE 18

Osveževanje 18
Preklop pomnilnih bank 18

KRMILNIK ZA GIBKE DISKE, SISTEMSKI PIO
PIO ZA SPLOŠNE NAMENE, CTC IN SIO 18

Krmilnik za gibke diske 18
Podatkovni separator 18
Krmiljenje glave 19
Notica 19

Sistemski PIO 19
PIO za splošne namene in SIO 20
Generator prenosne hitrosti 20

PREKINITVENA STRUKTURA 20

ZAHTEVE ZA NAPAJALNO NAPETOST 21

KAKO NASTAVLJAMO PREVEZE, KAJ POMENIJO KONEKTORJI .. 21

PIO splošnega namena (J5 JB3) 21
CTC (JB2) 22
Slikovni krmilnik (J6 JB7) 22
Kanal SIO A (J4), kanal B (J3) 23
Določanje konfiguracije serijskih kanalov
kanal A (JB4), kanal B (JB5) 23
Paralelna tipkovnica (J2 JB6) 24
Konektor za diskovne pogone (J1) 24
Konektor za priključitev napajanja (TB1) 24

Zaključevanje prekinitvene verige 24

Dodatne nastavitve na MMS 25

Dodajanje pomnilnika RAM/ROM 26
Uporaba funkcionalnega elementa za 8116 26

PRIKLJUCEVANJE 5-PALCNIH DISKOV 26

NAVODILO ZA SESTAVLJANJE 28

RACUNALNIK NE DELUJE - KAJ ZDAJ ? 36

KOSOVNICA SESTAVNIH DELOV 38

MIKRO SLOVENIJA MONITOR ROM PROGRAM ..... PRILOGA 1

ELEKTRICNE SHEME MMS ..... PRILOGA 2

OPIS MIKRORAČUNALNIŠKEGA SISTEMA MOJ MIKRO SLOVENIJA

Računalnik MMS naj bi izpolnil vsaj del potreb vseh tistih ljubiteljev mikroračunalniških sistemov, ki bi radi imeli zelo kakovosten, vendar cenejši izdelek. Pri nas lahko to željo uresničimo le s samogradnjo. Zatakne pa se že na začetku, saj pri nas ne moremo kupiti kakovostno zasnovanega mikroračunalnika, s kakovostno izvedbo tiskanega vezja in vso spremljajočo dokumentacijo, da lahko vsak, ki je veščč spajkanja in ima malo smisla za materialno opremo, sam zgradi mikroračunalnik z lastnostmi profesionalne zasnove. MMS podpira enega najbolj razširjenih mikroračunalniških operacijskih sistemov, CPM*. MMS je mikroračunalnik,ki vas ne bo pustil na cedilu, zavedati pa se morate, da je potrebno kupiti integrirana vezja, podnožja, diskovne pogone, narediti (kupiti) usmernik, tipkovnico in ohišje. Sredstva, ki jih je potrebno vložiti za izdelavo celotnega računalnika, so vsaj 10 X manjša od cene podobnega sistema na domačem tržišču in nekajkrat manjša od podobnega sistema v tujini.

Materialno opremo sistema MMS lahko razdelimo v dva dela:

Osnovni del in dodatki.

Osnovni del sestavljajo :

- pomnilnik 64 K zlogov
- centralni procesor Z80
- krmilnik za gibke diske
- slikovni krmilnik 24 X 80 znakov
- paralelna vrata za priključitev tipkovnice

Dodatki:

- dva popolna serijska kanala
- ura realnega časa
- dvoje paralelnih vhodno-izhodnih vrat
- dodatna pomnilna banka 8 K zlogov RAM ali ROM.

Seveda se da računalnik MMS dograjevati in izpopolnjevati, čeprav pogled na tiskano vezje odkrije, da nimamo na voljo sistemskega vodila. To je možno z implementacijo ortogonalne programske opreme (izkoriščanje obstoječih podnožij za razširitev funkcijskih lastnosti) ali pa z uporabo paralelnih in serijskih kanalov.

MMS ni ceneno uporabniški računalnik. Je računalnik, ki zahteva znanje o materialni in programski opremi, o sestavljanju in spuščanju v pogon. Če ste si pripravljeni pridobiti izkušnje po tej poti, gotovo ne boste razočarani.

---

* je oznaka za poljubno verzijo operacijskega sistema CP/M. Primer: CP/M 2.0 ali CP/M 2.2

MOJ MIKRO SLOVENIJA,
UPORABNISKI PRIROCNIK

Mikro Slovenija je osnovni kontrolni program (v nadaljevanju teksta ga imenujemo monitor) enokartičnega sistema MOJ MIKRO SLOVENIJA. Program se prične izvajati po vsaki vključitvi napajanja, ali po vsakem pritisku na tipko START. Zaseda zgornje 4 K pomnilnega prostora.

Monitor opravlja dve osnovni funkciji:
- je programski vmesnik sistema
- vsebuje vse potrebne podprograme za nadzor vhodno-izhodnih komunikacij.

Programski vmesnik sistema vsebuje ukaze za prikaz in spremembo pomnilnih lokacij, vhodno-izhodnega naslovnega področja, start programa z izbranega naslova in nalagalni krmilnik za delo z diskom.

Program za nadzor vhodno-izhodnih komunikacij je sestavljen iz krmilnika za slikovni prikazovalnik, paralelne tipkovnice, dveh serijskih kanalov, krmilnika za gibke diske in ure realnega časa. Te funkcije monitorja so stalno aktivne, tudi kadar sistem izvaja operacijski sistem CPM*.

V nadaljevanju bomo opisali uporabo monitorskih ukazov, funkcije vdelanih vhodno-izhodnih programskih vmesnikov in to, kako navezati uporabniške programe na monitorske podpograme.

OPIS OSNOVNIH UKAZOV

MIKRO SLOVENIJA stopi v ukazni način delovanja po vsaki vključitvi napajanja ali po pritisku na tipko START. Po pritisku na tipko CR na tipkovnici se na slikovnem prikazovalniku pokaže znak, da je sistem pripravljen:

(....Moj mikro Slovenija ....)

Vsi ukazi so sestavljeni iz ene črke in nič, enega, dveh ali treh šestnajstiških številskih parametrov. Ukaz lahko tipkamo z velikimi ali malimi črkami. Zaključni znak je vselej CR. Napake v ukazni vrstici lahko vedno popravimo (skočimo za mesto nazaj) s pritiskom na tipki ctrl-H ali zbrišemo celotno ukazno vrstico s pritiskom na tipki ctrl-X.

Uporabnik lahko predčasno ustavi izvajanje dolgih sekvenc. To stori s pritiskom na tipko CR in ukaz se bo nehal izvajati. Izpis na prikazovalnik in izvajanje ukazov pa lahko začasno prekinemo s pritiskom na tipki ctrl-S; prav tako sprožimo nadaljnje delo.

Naslednja tabela prikazuje vse monitorske ukaze. Sledi ji podroben opis posameznih ukazov.

ukaz	oblika
d(prikaži)	D/začetek/,/konec/
m(pomnilnik) ..	M/naslov/
t(test)	T/začetek/,/konec/
f(napolni)	F/začetek/,/konec/,/konstanta/
c(kopiraj)	C/začetek izv./,/konec izv./,/začetek ponora/
s(preklopi) ...	S
g(začni)	G/naslov/
r(čitaj)	R/enota/,/sled/,/sektor/
b(naloži)	B
i(vhod)	I/vhodni naslov/
o(izhod)	O/izhodni naslov/,/podatek/

Ukaz za izpis vsebine pomnilnika - D

Ukaz povzroči, da se vsebina pomnilnika izpiše v šestnajstiški obliki, ki ji je pridružena še definicija ASCII. Vsaka izpisana vrstica je naslednje oblike:

AAAA DD DD DD DD .... (16 krat) CCCC .... (16 krat)

AAAA je startna številka vrstice v šestnajstiškem zapisu, DD je vrednost pomnilne lokacije, CC pa ASCII definicija. Vrednosti, ki so manj kot 20H, so prikazane s piko.

Ukaz D je lahko nič, eno ali dvoparametričen. Če določimo dva parametra, bo prikazana vsebina pomnilnika med obema naslovoma. Vpis enega parametra prikaže 256 vrednosti, začenši pri vpisanem naslovu. Če vtipkamo samo D, bo prikazan izpis 256 vrednosti od zadnjega prikazanega naslova naprej.

Ukaz za izpis in spremembo pomnilnih lokacij - M

S tem ukazom lahko opazujemo in spreminjamo posamezne pomnilne lokacije. Ukaz je enoparametričen. Izpis na zaslonu je oblike:

AAA DD -

AAAA je tekoča naslovna lokacija, DD vrednost pomnilne lokacije (oboje seveda v šestnajstiškem zapisu). Po izpisu vsebine lokaci je sistem čaka na vpis naslednjih možnih znakov.

- Pritisk na tipko CR povzroči izpis vsebine naslednje pomnilne lokacije.

- Pritisk na tipko minus (-) povzroči izpis prejšnje pomnilne lokacije.

- Vpis dvocifrnega šestnajstiškega števila povzroči vpis te vrednosti v pomnilno lokacijo.

- Vpis kateregakoli drugega znaka pomeni skok iz ukaza M.

Testiranje pomnilnika - T

S tem ukazom poizkušamo odkriti okvarjen pomnilni element, kratek stik na tiskanem vezju ali kakšen drug stranski učinek, ki se zrcali v slabem delovanju pomnilnika. Testiramo lahko katerokoli področje, razen področja, na katerem je sam monitor (F000H do FFFFH). Vtipkamo vedno dva parametra: začetni in končni naslov. Test bo na zaslon izpisal znak plus (+) za vsak prehod prek definiranega področja, pri katerem ne bo odkril nobene napake. V primeru napak sledi obvestilo:

AAAA DD should=XX

AAAA je naslov okvarjene lokacije, DD je njena vsebina, XX pa vrednost, ki naj bi bila prečitana.

Ukaz za polnjenje pomnilnika - F

Ukaz omogoča, da se v pomnilniški blok poljubne dimenzije vpišejo konstante poljubne vrednosti. Ukaz je vedno triparametričen: začetni in končni naslov ter konstanta. Če sistem pri vpisu odkrije napako, jo javi na podoben način kot pri ukazu T.

Ukaz za kopiranje vsebine - C

Ukaz za kopiranje omogoča kopiranti iz enega v drug pomnilni blok. Ukaz je vedno triparametričen: začetni in končni naslov izvora ter začetni naslov ponora. Tudi, kadar je ukaz tak, sistem preveri, če sta oba bloka po končanem ukazu enaka.

Ukaz za preklop nadzorne enote - S

Ukaz uporabimo, kadar želimo preklopiti prikazovanje in vnos znakov iz trenutno aktivnega načina na drugo možnost. Pri sistemu MOJ MIKRO SLOVENIJA lahko izbiramo med dvema načinoma. Uporabljamo dve različni enoti za komunikacijo človek - stroj. To je lahko zunanji slikovni prikazovalnik (PAKA, KOPA etc.) ali pa vgrajen slikovni prikazovalnik s paralelno tipkovnico. Po inicializaciji sistema (tipka START ali vklop napajanja) je glavni komunikacijski medij tista enota, na kateri prej kot na drugi pritisnemo tipko CR. Z ukazom S preprosto preklapljamo (kolikokrat hočemo) med obema načinoma komunikacije človek-stroj. Seveda moramo pri preklopu na zunanji serijski slikovni terminal upoštevati tudi prenosno hitrost. Po definiciji je izbrana hitrost 300 bitov/s. Če nam ni všeč, jo lahko poljubno spremenimo tako, da vtipkamo ustrezno vrednost (glej tabelo za izbiro komunikacijskih hitrosti) prek ukaza O v izhodni prostor (O OC,/podatek/). Ta prenosna hitrost velja do naslednje inicializacije sistema.

Ukaz za izvajanje programa - G

S pomočjo tega ukaza startamo poljuben program s poljubne pomnilne lokacije. Ukaz je enoparametričen. Vsebuje samo šestnajstiško številko startne lokacije. Monitorski program je zgrajen tako, da ob tem ukazu prične izvajati podprogram (preskok v izvajanje prek ukaza CALL); tako se lahko po končanem programu vedno vrnemo v monitorski program z ukazom RET (seveda,

če nismo popackali sklada ali izgubili naslova, na katerega se moramo vrniti).

Ukaz za čitanje z gibkega diska - R

Ukaz omogoči čitati posamezne sektorje na disku; njihova vsebina se izpiše na zaslon. Določiti moramo tri parametre: številko diskovne enote (0 - 3), številko sledi (0 - 4C) in številko sektorja (1 - 1A). Podprogram izbere ustrezen pogon, premakne glavo na željeno sled in odčita zahtevani sektor. Če je sektor prečitan brez napake, bo izpisan v formatu, kot ga uporablja ukaz D. Pri napaki se bo na zaslonu izpisalo diagnostično obvestilo v naslednji obliki:

disk error XX UAA TBB SC

XX pomeni statusno kodo krmilnika za diskovni pogon, AA je številka pogona, BB je številka sledi in CC številka sektorja. Statusna koda je sestavljena iz osmih bitov, ki pomenijo:

bit	čitaj/piši	premik/vrnitev/izbor
7	ni pripravljen	ni pripravljen
6	pisalna zaščita	neuporabljeno
5	pisalna napaka	neuporabljeno
4	ne najde zapisa	napaka pri premiku
3	crc napaka	crc napaka
2	izgubljen podatek	vrnitev ni možna
1	neuporabljeno	neuporabljeno
0	vedno 1	vedno 0

Bit z najmanjšo utežjo določa, kateri del diagnostičnega sporočila je pomemben. Če je bit 0 enak 1, gre za napako pri čitanju in pisanju, drugače pa za premik, vrnitev ali izbor diska.

Ukaz za nalaganje operacijskega sistema - B

Ta ukaz nam rabi za nalaganje in start izvajanja vsebine enega sektorja na sledi 0 pogona 0. Najbolj naravna uporaba tega ukaza je start operacijskega sistema CPM *, čeprav se ne omejuje samo na to opravilo. Ukaz B prebere vsebino sektorja 1 na sledi 0 in jo naloži na pomnilni segment s začetnim naslovom 80H. Ponavadi je ta program nalagalnik za kak večji program, kakršen je prav gotovo operacijski sistem. Taka dvonivojska nalagalna struktura je precej bolj neodvisna od konkretne aplikacije. Edina zahteva, da je ukaz B izpolnjen, je, da je prvi sektor na sledi 0, pogona 0, namenjen za nalagalnik in da je spodnjih 256 zlogov pomnilnika na voljo pri drugi fazi nalaganja .

Ukaz čitaj vhodni prostor - I

Ukaz uporabimo za čitanje zlogov vhodnega prostora (vhodnih vrat). Deluje zelo podobno kot ukaz M, le da namesto pomnilnega področja posega po vhodnem naslovnem področju. Tudi način vnosa parametrov in pregled posameznih vrednosti je zelo podoben ukazu M.

S tem ukazom je mogoče vpisovanti zloge v izhodni prostor. Ukaz je dvoparametričen: številka izhodnega prostora in vrednost zloga, ki ga vpisujemo. Oba parametra morata biti med 0H in FFH. Po vpisu zloga se ponovno javi monitor.

VHODNO IZHODNE FUNKCIJE MONITORJA

V tem poglavju se bomo seznanili z osnovnimi vhodno-izhodnimi funkcijami glavnega nadzornega krmilnega programa MIKRO SLOVENIJA.

Obdelava prekinitev

Program MIKRO SLOVENIJA uporablja vse dobre lastnosti prekinitvene strukture, ki jo omogoča družina Zilog *. Prekinitve rabijo za delo s tipkovnico, ure realnega časa in krmilnika za gibke diske. Vse potrebne inicializacije in ustrezni podprogrami so že vsebovani v monitorju.

V glavnem je prekinitvena struktura lahko transparentna za večino aplikacijskih programov, ki jih poganjamo z računalnikom. Seveda pa moramo upoštevati nekatere omejitve, da se lahko uporabniški program lepo vključi v obstoječo strukturo prekinitev.

- Ne smemo nenadzorovano uporabljati ukaza DI, ker zaradi tega ne more delovati tipkovnica ali zunanji slikovni terminal in ura realnega časa.
- Ne smemo se dotikati registra I, ker bo sicer prav gotovo izpadel sistem.
- Z80 deluje v prekinitvenem načinu 2, ki mora ostati nedotaknjen.
- Uporabniški programi terjajo primeren pomnilni prostor za sklad. Uporabljati sklad za izvajanje programskih poslastic je nezaželeno.

Inicializacijski program nastavi register I tako, da kaže na prekinitveno tabelo vektorjev, ki je na pomnilnih lokacijah FF00H do FF1FH. Ta tabela vsebuje vse potrebne prekinitvene kazalce za vsa vgrajena integrirana vezja, ki uporabljajo prekinitveno strukturo. Vezja, ki niso vključena v obstoječo strukturo, so: SIO kanal A, GPIO in kanala 0 in 1 elementa CTC. Te elemente lahko preprosto inicilizirамo, ne da bi ovirali delovanje sistema. Zato je moramo pregledati spremenljivke v osnovnem nadzornem programu.

Slikovni prikazovalnik

Materialna oprema mikroračunalnika vsebuje tudi krmilnik za slikovni prikazovalnik 80 X 24 znakov. Pomnilnik za ta prikazovalnik je v drugi pomnilni mapi in ne omejuje osnovnih 64 K pomnilnega prostora. Vsebuje še dodatek za pomik vrstice, narejen z materialno opremo. S programskega stališča pomenita vgrajeni slikovni prikazovalnik in paralelna tipkovnica emulator za tipičen samostojen terminal. Zbirka nadzornih znakov je usklajena s priznanim terminalom ADM3-A firme Lear Sigler.

Opis posameznih funkcij krmilnika za slikovni prikazovalnik

Vsi znaki med 20H in 7FH so prikazani na ekranu. Razpeti so nad matričnim poljem dimenzije 5X7.

Nov vtipkan znak zajema prostor, kjer je bila poprej svetlobna točka, ki se se je pomaknila mesto naprej.

Če se svetlobna točka pokaže tam, kjer je že kak znak, bo ta začel utripati.

Svetlobna točka se samodejno premakne na začetek nove vrstice, če je prejšnja polna.

Vsi znaki med 00 in 1F so ovrednoteni kot krmilni znaki. Samo 14 vpliva na slikovni prikazovalnik.

koda	znak	ime	vpliv na prikazovalnik
00	CRTL	NUL	ne
01	CTRL A	SOH	ne
02	CTRL B	STX	ne
03	CTRL C	ETX	ne
04	CTRL D	EOT	ne
05	CTRL E	ENQ	ne
06	CTRL F	ACK	ne
07	CTRL G	BEL	zvočni signal
08	CTRL H	BS	pomik nazaj
09	CTRL I	HT	horizontalni tabulator
0A	CTRL J	LF	pomik valja
0B	CTRL K	VT	pomik svetlobne točke navzgor
0C	CTRL L	FF	pomik svetlobne točke na desno
0D	CTRL M	CR	zaključni znak
0E	CTRL N	SO	ne
0F	CTRL O	SI	ne
10	CTRL P	DLE	ne
11	CTRL Q	DC1	ne
12	CTRL R	DC2	ne
13	CTRL S	DC3	ne
14	CTRL T	DC4	ne
15	CTRL U	NAK	ne
16	CTRL V	SYN	ne
17	CTRL W	ETB	zbriše zaslon od trenutne svetlobne točke do konca zaslona
18	CTRL X	CAN	zbriše del vrstice od svetlobne točke do konca vrstice
19	CTRL Y	EM	ne
1A	CTRL Z	SUB	zbriše ves zaslon

1B	CTRL [	ESC	ubežna sekvenca. Uporabimo jo za pozicioniranje svetlobne točke. Lahko jo premaknemo, kamor želimo, po naslednjem zaporedju znakov: ESC = številka vrstice številka stolpca. Številka vrstice je določena (0 - 23) Številka stolpca je določena (0 - 79) Obema številoma moramo prišteti še 20H.
1C	CTRL \	FS	ne
1D	CTRL]	GS	ne
1E	CTRL ^	RS	pomakne svetlobno točko v zgornji levi kot in pri tem ostane drugi del zaslona naspremenjen.
1F	CTRL_	VS	naslednji znak bo prikazan na zaslonu ne glede na njegovo vrednost.

Vhod za paralelno tipkovnico

Vmesnik omogoča, da se priključi paralelna, sedembitna tipkovnica. Program vsebuje prekinitveni nadzorni program, ki polni 16-zložni register FIFO za čitanje podatkov. Tako je mogoče vnaprejšnje tipkanje brez izgubljanja znakov. To ne velja, če tipkamo med delovanjem diska, ko lahko izgubimo kak znak s tipkovnice, saj program za disk ovira prekinitveno strukturo. Vsak znak, ki ga vtipkamo, potem ko je FIFO že poln, je izgubljen.

Program za tipkovnico vsebuje tudi programsko alfamerično zaklenitev. To je posebna pomnilna lokacija, ki ima ob inicializaciji vrednost 00H. Ko je s tipkovnice sprejet znak, enak vsebini te pomnilne lokacije, preide vrednost alfamerične zaklenitve v svoje nasprotno stanje. Tak znak seveda ne gre v FIFO. Znakov, po vrednosti manjših od 40 H, alfanumerična zaklenitev ne zajame.

Krmilnik za gibke diske

Računalnik je opremljen s krmilnikom za 8-palčne gibke diske enojne gostote. Srce krmilnika je integrirano vezje 1771 firme Western Digital. Tiskano vezje pa vsebuje elemente, potrebne za priključitev visoko sposobnih krmilnikov iste firme z oznako 279X, s katerimi lahko krmilimo poljubno kombinacijo 8 ali 5-palčnih diskov, ki so eno ali dvostranski, enojne ali dvojne gostote.

Programski paket pa vsebuje vse programe za nadzor krmilnika, ki je uporaben tudi za 279X. Uporabnik lahko sam določi hitrost koračnih motorjev in dolžino sektorja. Rezultat vsake operacije z diskom je preverjen in če pride do napake, se izvrši še en poizkus.

Priključitveni konektor za gibke diske je narejen po vzoru standarda firme Shugart.

Vmesnik za serijsko komunikacijo

Računalnik vsebuje dva serijska kanala, ki sta združena v integriranem vezju Z80-SIO. Oba sta popolnoma neodvisna kanala RS232 in ju lahko uporabimo za priključitev zunanjega terminala, tiskalnika, modema ipd. Ob inicializaciji sistema programska oprema samodejno ugotovi, kateri medij bo za komunikacijo človek - stroj, paralelna tipkovnica ali zunanji terminal (na primer

Kopa, Paka ipd.). V primeru, da uporabimo zunanji terminal, postane serijski kanal B komunikacijska vez v asinhronem načinu delovanja. Programska oprema tudi samodejno izračuna hitrost prenosa podatkov in računalnik se sam prilagodi prenosni hitrosti terminala. Serijski kanal deluje s pomočjo prekinitev.

Če sistem uporablja vgrajen terminal, elementa SIO ni potrebno vgraditi v sistem, programska oprema pa ga inicializira kot kanal za povezavo s tiskalnikom. Tudi v tem primeru je SIO inicializiran za asinhrono komunikacijo, s prenosno hitrostjo 300 bitov/s, prekinitve pa so prepovedane. Program vsebuje nekaj vstopnih točk, s pomočjo katerih lahko krmilimo SIO v tem načinu delovanja. Kakorkoli uporabimo SIO, vedno je inicializiran na 7-bitno besedo, liho parnost in en stop bit.

Ura realnega časa

V računalnik lahko po želji vgradimo tudi časovno vezje Z80-CTC, ki postane osnova za generiranje raznih prekinitev, kot so naprimer razni časovniki, ura realnega časa, ipd. Če je CTC vgrajen, bo programska oprema inicializirala kanala 2 in 3. Kanala 0 in 1 še lahko uporabimo v poljubne namene.

Enosekundna prekinitev iz CTC rabi za izključevanje diskovnih pogonov, če so neaktivni. Po 30 sekundah neaktivnosti diska, lahko motor izključimo. V ta namen uporabimo sponko DAC na napajalnem konektorju. Seveda bo disk ob prvem naslednjem posegu programsko takoj aktiviran.

Paralelna vrata

V računalniku je predviden prostor za dodatno integrirano vezje Z80 - PIO. Osnovna programska oprema ga ne inicializira in nam je v celoti na voljo. Uporabimo ga lahko za priključitev paralelnega printerja, programatorja pomnilnika ROM, EPROM, EEPRPOM, analognih pretvornikov ipd.

Pregled sistemskih podprogramov, ki so na voljo uporabniku

Podprogrami osnovnega krmilnega programa (monitorja) so uporabniku dosegljivi prek tabele skočnih instrukcij, ki se prične pri naslovu F000H. Vse posege v podprograme moramo narediti s to tabelo. Prenos parametrov poteka na dva načina. Znakovno usmerjeni vhodno-izhodni podprogrami uporabljajo register A, podprogrami za gibki disk pa prenašajo parametre v registrih HL in A.

Spominski prostor za sklad in spremenljivke zajema zgornjih 256 pomnilnih lokacij, FF00H - FFFFH, tabela za prekinitveni način 2 prvih 32 zlogov tega pomnilnega prostora, sklad pa je ob inicializaciji na samem vrhu pomnilnega prostora. Vmes so spremenljivke, ki jih uporablja monitor, rezidenčni vhodno-izhodni krmilniki in prekinitveni uporabni podprogrami.

Tabela skočnih instrukcij:

```
F000      ENT0: JP  INIT    ;hladni zagon
F003      ENT1: JP  PROMPT  ;topli zagon
F006      ENT2: JP  CONST   ;status vhodne naprave
F009      ENT3: JP  CONIN   ;sprejem znaka
F00C      ENT : JP  CONOUT  ;oddaja znaka
F00F      ENT4: JP  CRTOUT  ;oddaja znaka na vgrajen
                            ;slikovni prikazovalnik
F012      ENT5: JP  SIOST   ;SIO status - kanal B
F015      ENT6: JP  SIOIN   ;SIO sprejem znaka - kanal B
F018      ENT7: JP  SIOOUT  ;SIO oddaja znaka - kanal A
F01B      ENT9: JP  SELECT  ;izbor diskovnega pogona
F01E     ENT10: JP  HOME    ;pomik glave na stezo 0
F021     ENT11: JP  SEEK    ;pomik glave na stezo
F024     ENT12: JP  READ    ;prečitaj sektor v pomnilnik
F027     ENT13: JP  WRITE   ;piši sektor iz pomnilnika
```

Podroben opis funkcij

COLD vhodni parameter: nič
izhodni parameter: nič
opis: izvede hladen zagon osnovnega krmilnega programa (monitor) in inicializira ves sistem

WARM vhodni parameter: nič
izhodni parameter: nič
opis: izpis znaka pripravljenosti monitorja brez inicializacije

CONST vhodni parameter: nič
izhodni parameter: status v registru A
opis: pogleda, če je kak podatek v FIFO registru. Če je A=0, drugače je FFH.

CONIN vhodni parameter: nič
izhodni parameter: znak v registru A
opis: prečita znak iz registra FIFO in ga vrne v A. Če je FIFO prazen, čaka v zanki na znak.

CONOUT vhodni parameter: znak v A
izhodni parameter: nič
opis: znak, ki je v registru A, se vpiše v komunikacijsko enoto.

SIOST vhodni parameter: nič
izhodni parameter: status v A
opis: pogleda, če je v sprejemniku SIO kanala B kak znak. Če znak je, vpiše v A ničlo, drugače FFH.

SIOIN vhodni parameter: nič
izhodni parameter: znak v A
opis: vrne sprejeti znak iz SIO kanal B. Če znaka še ni, čaka nanj.

SIOOUT vhodni parameter: znak v A
izhodni parameter: nič
opis: pošlje znak v izhodni register SIO kanal B

SELECT vhodni parameter: številka pogona v C
izhodni parameter: status v A
opis: izbere disk, na katerem bodo tekle diskovne operacije. Če pogon ni pripravljen, teče naprej prejšnji pogon.

HOME vhodni parameter: nič
izhodni parameter: status v A
opis: pomik glave na sled 0 in verifikacija

SEEK vhodni parameter: številka sledi v C
izhodni parameter: status v registru A
opis: premik glave na želeno sled in verifikacija

READ vhodni parameter: številka sektorja v A
kazalec na začetek bloka v HL
izhodni parameter: status v A
opis: prebere želeni sektor v pomnilnik

WRITE vhodni parameter: številka sektorja v A
kazalec na začetek bloka v HL
izhodni parameter: status v A
opis: popiše želeni sektor z vsebino definiranega pomnilnika.

Opis dodeljenega prostora sistemskih spremenljivk

Tabela prekinitev

```
FF00   SIOV0: DEFS 2   ;SIO kanal B, oddajnik prazen
FF02   SIOV1: DEFS 2   ;SIO kanal B, zamenjava
FF04   SIOV2: DEFS 2   ;SIO kanal B, podatek na voljo
FF06   SIOV3: DEFS 2   ;SIO kanal B, poseben pogoj za  sprejem
FF08   SIOV4: DEFS 2   ;SIO kanal A, oddajnik prazen
FF0A   SIOV5: DEFS 2   ;SIO kanal A, zamenjava
FF0C   SIOV6: DEFS 2   ;SIO kanal A, podatek na voljo
FF0E   SIOV7: DEFS 2   ;SIO kanal A, poseben pogoj za sprejem
FF10   CTCV0: DEFS 2   ;CTC kanal 0  prekinitev
FF12   CTCV1: DEFS 2   ;CTC kanal 1  prekinitev
FF14   CTCV2: DEFS 2   ;CTC kanal 2  prekinitev
FF16   CTCV3: DEFS 2   ;CTC kanal 3  prekinitev

FF18   SYSVA: DEFS 2   ;sistemski PIO kanal A prekinitev
FF1A   SYSVB: DEFS 2   ;sistemski PIO kanal B prekinitev

FF1C   GENVA: DEFS 2   ;splošni PIO kanal A prekinitev
FF1E   GENVB: DEFS 2   ;splošni PIO kanal B prekinitev
```

Spremenljivke za paraleleno tipkovnico

```
FF20    FIFO:  DEFS 16  ;šestnajstzložni FIFO register
FF30  FIFCNT:  DEFS 1   ;število znakov v FIFO registru
FF33    LOCK:  DEFS 1   ;znak za alfamerično zaklenitev
```

Spremenljivke za realni čas

```
FF5D  TICKNT: DEFS 2   ;binarni enosekundni števec
FF5F  DAY:    DEFS 1   ;dan v tednu
FF60  MONTH:  DEFS 1   ;mesec
FF61  YEAR:   DEFS 1   ;leto
FF62  HRS:    DEFS 1   ;ura
FF63  MIN:    DEFS 1   ;minuta
FF64  SEC:    DEFS 1   ;sekunda
```

Vhodno izhodne spremenljivke za diskovne podprograme

```
FF6A  SPEED:  DEFS 1   ;hitrost premika glave
FF6B RECLEN:  DEFS 1   ;dolžina zapisa za čitanje/pisanje
FF6C  MOTOR:  DEFS 1   ;časovnik za aktivnost motorja
```

Spremenljivke za vdelani slikovni prikazovalnik

```
FF76 CSRCHR:  DEFS 1   ;znak za označevanje svetlobne točke
FF77   BASE:  DEFS 1   ;trenutna vsebina pomikalnega registra
```

KAKO RACUNALNIK DELUJE

CENTRALNI PROCESOR

generator urnih impulzov

Vsi sistemski impulzi, razen tistih za prikazovalnik in serijski prenos podatkov, nastanejo z delilno logiko in oscilatorjem 20Mhz (16Mhz). Ura 20Mhz je deljena s 5 z uporabo desetiškega števca U-97. Dobimo 4Mhz za delovanje separatorjev podatkov za gibke diske. Ura 4Mhz je deljena še z 2 s preostankom integriranega vezja U-96. Uro 2Mhz uporablja krmilnik za gibke diske.

Ura s taktom 2.5 Mhz je dobljena tako, da je 20Mhz signal deljen z 8, (binarni števec U-96). U-96 generira tudi vse potrebne impulze, če vzamemo namesto oscilatorja 20 MHz oscilator 16 Mhz. Delilna logika je zgrajena tako, da oscilator 20Mhz omogoča delovanje procesorja s taktnima frekvencama 2,5 in 5 Mhz, oscilator 16 Mhz pa s taktno frekvenco 4Mhz. Signal za poganjanje procesorja je speljan v logiko za oblikovanje impulzov, kot jih potrebuje procesor Z-80.

Signali za krmiljenje dinamičnih pomnilnikov nastajajo prek osnovnega oscilatorja. Ker je ta lahko 20 ali 16 Mhz, moramo pregledati podroben opis nastavitev. Pri posegu v pomnilnik (MREQ=0) in RFSH=1 logika generira krmilna signala CAS in MUX. Signal RFSH=0 pa onemogoči ta dva signala, saj ne more delovati pomikalni register U-76. Na ta način se poraba energije v dinamičnem pomnilnem polju precej zmanjša.

krmilnik za inicializacijo sistema

V sistem sta vgrajena dva načina inicializacije: vklop napajalne napetosti in pritisk na tipko START. Logika odkrije vklop napajalne napetosti tako, da uporabi del vezja U-101. Drugi del vezja rabi za detekcijo pritiska na tipko START. Pomnilni element U108 sinhronizira pritisk na tipko in strojni cikel M1 procesorja Z80. Izhod tega elementa aktivira monostabilni multivibrator U-106. Časovna konstanta je 12 mikrosekund. Oba načina inicializacije sta združena prek funkcije OR elementa U-81, nakar združeni signal še negira U-103 in ga pripelje v procesor. Družino pripadajočih vezij procesorja Z80 inicializirata inicializacijski impulz za Z80 in signal M1.

ojačevanje sistemskega vodila

Osembitni ojačevalec U-82 ojačuje vse krmilne signale sistema. Dva štiribitna ojačevalca U-78 in U-79 skrbita za prenos podatkov. Del vezja U-87 pravilno usmerja preklop pretoka podatkov. V fazi čitalnih ciklov procesor sprejema podatke, drugače pa se ojačevalci vedno usmerjajo k pisanju podatkov. Osembitni ojačevalec U-65 ojačuje nižje bite naslovnega vodila. Osembitni register U-83 opravlja dve funkciji. Ojačuje zgornjih osem naslovnih linij in zagotavlja stabilnost naslovnega vodila, dokler traja cikel MREQ. Procesor Z80 namreč ne zagotavlja stabilnosti naslovnega vodila na celotnem področju MREQ - naslovno vodilo se lahko spremeni. Posledica se pokaže pri

generiranju napačnih impulzov RAS, ki krmilijo delovni pomnilnik.

druga pomnilna banka

V sistem lahko vgradimo še 8 K dodatnega statičnega pomnilnega prostora.

U-67 (tip ROM) področje 0000H do 07FFH
U-68 (tip ROM) področje 0800H do 0FFFH
U-69 (tip ROM/RAM) področje 1000H do 17FFH
U-70 (tip ROM/RAM) področje 1800H do 1FFFH

Pomnilna elementa U-69 in U-70 sta lahko vrste RAM ali ROM.

Opis preklopa banke je naveden pri opisu, kako deluje dinamični pomnilnik.

dekodiranje vhodno izhodnega naslovnega prostora

Osembitni dekoder U-85 izbira ustrezne vhodno-izhodne elemente z uporabo naslovnih bitov A2, A3 in A4. Ko je A7 =0 , M1R =1 in IORQ =0, dekoder izbere enega izmed vhodno-izhodnih elementov ter omogoči vpis ali čitanje.

pregled naslovnega prostora

vrata 0 - 3 = izbor prenosne hitrosti za serijska kanala
vrata 4 = SIO A podatek
vrata 5 = SIO B podatek
vrata 6 = SIO A nadzor
vrata 7 = SIO B nadzor
vrata 8 = gp PIO kanal A
vrata 9 = gp PIO kanal A nadzor
vrata A = gp PIO kanal B podatek
vrata B = gp PIO kanal B nadzor
vrata C - F = kanal B prenosna hitrost
vrata 10 = 1771(2791) statusni/krmilni register
vrata 11 = 1771(2791) sled
vrata 12 = 1771(2791) sektor
vrata 13 = 1771(2791) podatek
vrata 14-17 = register za pomik na zaslonu
vrata 18 = CTC kanal 0
vrata 19 = CTC kanal 1
vrata 1A = CTC kanal 2
vrata 1B = CTC kanal 3
vrata 1C = sistemski podatkovni register
vrata 1D = sistemski nadzorni register
vrata 1E = podatek iz tipkovnice
vrata 1F = nadzor tipkovnice

Sinhronizacija prenosa podatkov pri gibkih diskih

Za uspešno in hitro prenašanje podatkov med diskovnim krmilnikom in procesorjem je uporabljena nemaskirna prekinitev NMI na procesorju. Pri čitanju in pisanju z diska se shrani podatek iz lokacije 66H. V to lokacijo se vpiše koda instrukcije RETURN. Ko preide procesor v položaj HALT, bo signal DRQ ali NMI povzročil na procesorju signal NMI. Procesor bo izvedel instrukcijo RETURN na lokaciji 66H in se vrnil k prenosu

podatkov. Ko je prenos vseh 128 zlogov končan, bo podatek na lokaciji 66H restavriran in procesor nadaljuje delo. Ta pripomoček v materialni opremi nadomesti potrebo po krmilniku DMA.

SLIKOVNI PRIKAZOVALNIK

generiranje slikovnih impulzov

Trije negatorji elementa U-11 so uporabljeni za generacijo ure slikovnih točk. Osnovni oscilator je frekvence 13.9776 Mhz. Če takšnega nimamo, lahko uporabimo kristal frekvence 14 Mhz (Spectrum). Ta frekvenca je deljena s 7. Dobimo znakovne impulze. Sinhronski binarni števec U-24 je krmiljen tako, da omogoča deljenje s številom 7. Ura za znakovne impulze je deljena s 128 prek osembitnega binarnega števca U-23. Tako dobimo skanirne impulze. Števec U-22 deli te impulze z 12, dobimo vrstične impulze in vertikalno komponento naslovnega vodila za znakovni generator. U-21 in del vezja U-9 določata vrstični naslov slikovnega pomnilnika. Vrstične impulze delita s 26 za generiranje ure 50 Hz. Druga polovica vezja U-21 deli 50 Hz s 16. Ta signal rabi za utripanje.

pomik slike na zaslonu

Za pomik slike skrbi materialna oprema. U-35 je register za pomik slike. Njegova vsebina je dodana po modulu 24 vrstičnemu naslovu s seštevalnikoma U-36 in U-34.

naslavljanje slikovnega pomnilnika

Multipleksorji U-47, U-48 in U-49 izberejo naslov za slikovni pomnilnik. Če ga piše ali čita procesor, bo signal CRTCE postal logična 0. Na pomnilnem polju se bo pojavil naslov, ki ga določa procesor. Sicer pa naslov slikovnega pomnilnika določa veriga števcev.

generiranje sinhronizacijskih impulzov

Horizontalni sinhronizacijski impulz generira dekodirna logika za osemdeseti znak v vrstici. Brž ko se impulz pojavi, aktivira monostabilni multivibrator U-51. Dolžino impulza lahko določimo po enačbi:

(Tw = 0.7 * Rt * Cext)

Rt je v komih, Cext v pf, Tw je v ns

Ko je konec impulza za horizontalno zakasnitev, se sproži monostabilni multivibrator za horizontalno sinhronizacijo. Enačba za določanje dolžine impulza je enaka kot za horizontalno zakasnitev.

nastanek slike

V prikazovalnem načinu delovanja se podatki iz slikovnega pomnilnika (ASCII zapis) in naslov iz desetiškega števca U-22 združijo v izbor pripadajočega vzorca za generator znakov U-73. Informacija o svetlobnih točkah se ujame v oktalni D spominski element U-74. Med posegom po noven vzorcu prikazuje na zaslonu prejšnji vzorec.

zatemnitev zaslona

Zaslon je zatemnjen pri vodoravnem in navpičnem vračanju žarka, pri posegu procesorja v slikovni pomnilnik in takrat, ko vrstični števec izkazuje vrednost 8 in 9. Zatemnitev je realizirana tako, da se onemogoči znakovni generator.

Pomnilna banka 64K in preklapljanje

Za krmiljenje naslovnih linij pomnilnega polja skrbita multipleksorja U-58 in U-59. Med posegom v pomnilnik se na pomnilnem polju najprej pokaže naslov vrstice. Ko je ta naslov stabilen, se aktivira eden od štirih možnih vrstičnih impulzov glede na vrednost naslovnih linij A15B in A14B. Nekaj trenutkov po nastanku vrstičnega impulza logika generira impulz MUXC, ki preklopi vrstični naslov v naslov stolpca. Ko je naslov stolpca stabilen, se generira impulz CAS. Če gre za čitalni cikel, se podatki iz pomnilnega polja pojavijo na izhodu ojačevalcev U-54 in U-57. Pri pisalnem ciklu pa ista ojačevalca poskrbita, da pomnilno polje sprejme podatke iz procesorja. Tiskano vezje mikroračunalnika omogoča nastavitev časa, ko naj traja aktivni cikel krmiljenja pomnilnega polja. S prevezami lahko nastavimo trenutek pojavitve signalov MUXC in CAS. To je pomembno, če se odločimo za hitrejše delovanje procesorja (ki nam mora to seveda omogočati) in če hočemo prilagoditi aktiven čas trajanja pomnilnega cikla novim časovnim razmeram, ki ji določa hitrejši procesor. Oglej si poglavje o nastavljanu krmiljenja pomnilnega polja.

osveževanje

V osveževalnem ciklu procesor Z80 postavi na spodnjih sedem naslovnih linij osveževalni naslov. Ko je ta stabilen, se aktivira signal RFSH. Ta signal sproži vse štiri signale RAS, ne moreta pa se pojaviti signala MUXC in CAS.

preklop pomnilnih bank

Bit 7 izhodnega prostora na naslovu 1CH nadzoruje preklop. Ko je izhod v logični enici, se v spodnjih 16K naslovnega prostora procesorja pojavita polje RAM/ROM in slikovni RAM. Drugače pa je vseh 64K dinamičnega pomnilnika na voljo procesorju.

KRMILNIK ZA GIBKE DISKE, SISTEMSKI PIO IN CTC

krmilnik za gibke diske

1771 (U-107) izvaja vse nadzorne funkcije za krmiljenje diskovnih pogonov. Edini dodatek je podatkovni separator, ojačevalci treh stanj, logika za pripravljenost glave in ojačevalci za povezavo z diskovnimi pogoni.

podatkovni separator

Nastavljivi števec U-98 je uporabljen kot digitalni monostabilni multivibrator, ki ga poganja ura 4Mhz. Če števec ne sprejme podatkovnega bita med dvema urnima bitoma, se števec izteče in dodeli krmilniku vrednost 0. Če pa se pojavi

podatkovni bit med dvema urnima impulzoma, je krmilniku dodeljena logična 1.

krmiljenje glave

Ko 1771 aktivira izhod HLD, vzbudi monostabilni multivibrator U-106. 771 vzorči vhod HLT, dokler ne odkrije logične 1. To pomeni: glava na disketni enoti je pritisnjena ob disketo in se ne premika več (problem mehanike). Čas impulza lahko poljubno spreminjamo, pač glede na različne vrste diskovnih pogonov.

notica

Vezje 1771 je po razporeditvi nožic zelo podobno vezjem družine 2791. To je krmilnik, ki je namenjen za krmiljenje 8 ali 5-palčnih diskov, enojne ali dvojne gostote . V primerjavi z 1771 potrebujejo za svoje delovanje še dve krmilni sponki, ki pokažeta, ali gre za 8 ali 5-palčne diske, enojno ali dvojno gostoto zapisa. Te sponke so na tiskanini predvidene - označene so z XYZ. Tako lahko z zelo preprostim tiskanim vezjem nadomestimo vezje 1771 z vezjem 2971. Na sistem lahko priključimo naslednjo kombinacijo gibkih diskov.

- 8'' enojna gostota in 5'' dvojna gostota
- samo 8'' dvojna gostota
- samo 5'' enojna gostota

Vse kombinacije so lahko eno ali dvostranske.

Sprememba programske opreme za priključitev različnih kombinacij terja samo pravilno nastavitev logične vrednosti sponk XY pred posegom v ustrezen diskovni pogon.

sistemski PIO

Sistemski PIO je na vhodno-izhodnem naslovnem področju od 1CH do 1FH. Stran A tega elementa je namenjena za nadzor in izbor aktivnega diskovnega pogona, selekcijo pomnilne banke in uporabniško definiranih sponk.

Biti 0, 1 in 2 so za izbor diskov in izklop motorja
bit 3 je za programsko ugotavljanje podatka iz tipkovnice (če nočemo uporabljati prekinitev)
bit 4 je uporabniško usmerjen
bit 5 je uporabniško usmerjen
bit 6 je uporabniško usmerjen
bit 7 je za preklop pomnilne banke

Stran B elementa PIO je namenjena za priključitev tipkovnice. Osmi bit tipkovnice (bit 7) mora biti ozemljen. Aktivacijski impulz s tipkovnice je lahko pozitiven ali negativen. Logično vrednost nastavljamo s prevezo. Oglej si poglavje o nastavitvah.

PIO SPLOSNEGA NAMENA IN SIO

PIO za splošno uporabo daje uporabniku na razpolago 16 poljubno definiranih bitov, ki so lahko po želji vhod ali izhod. Oglej si poglavje o nastavitvah.

Element SIO vsebuje dva celotna serijska kanala, ki sta v skladu s standardom RS232. Stran A lahko oddaja ali sprejema urne signale za sinhron način delovanja. Tako se je mogoče priključiti z računalnikom na sinhrono komunikacijsko mrežo. Nožica SYNC je pri kanalih A in B povezana z Rx, da je mogoče programsko ugotavljati prenosno hitrost. Če želimo uporabljati te nožice v kak drug namen, lahko povezavo brez posledic prekinemo.

Generator prenosne hitrosti

Element COM 8116 je uporabljen kot dvokanalni generator prenosne hitrosti za kanala A in B elementa SIO. Kanal A leži na naslovu 00H, kanal B pa na naslovu 0CH. V oba lahko samo vpisujemo. Ker utegnemo kupiti funkcionalno enake elemente, ki pa po napajanju niso v soglasju, je na tiskanem vezju predvidena povezava, prek katere je mogoče priključiti elemente z drugačnim načinom napajanja.

tabela prenosnih hitrosti:

00H = 50	bitov/sekundo
01H = 75	"
02H = 110	"
03H = 134.5	"
04H = 150	"
05H = 300	"
06H = 600	"
07H = 1200	"
08H = 1800	"
09H = 2000	"
0AH = 2400	"
0BH = 3600	"
0CH = 4800	"
0DH = 7200	"
0EH = 9600	"
0FH = 19200	bitov/sekundo

Prekinitvena struktura

Vsi vgrajeni elementi družine Z80 omogočajo prekintvene načine 0, 1 in 2. Vgrajen monitor uporablja način 2. Register I je inicializiran na 0FFH. Prioritetna veriga teče po naslednjem vrstnem redu:

SIO kanal A -- najvišja prioriteta
SIO kanal B
sistemski PIO kanal A
sistemski PIO kanal B
PIO splošni namen kanal A
PIO splošni namen kanal B
CTC kanal 0
CTC kanal 1
CTC kanal 2
CTC kanal 3 -- najnižja prioriteta

Zahteve za napajalno napetost

Sistem terja tri napajalne napetosti:
+5.0V --- 3A
+12V ---- 0.25A
-12V ---- 0.20A

Posebej priporočamo uporabo kakovostnih usmerikov, po možnosti z napetostno zaščito s faktorjem 1.25.

KAKO NASTAVLJAMO PREVEZE IN KAJ POMENIJO KONEKTORJI

Nastavljanje PIO za splošni namen (JB3) in konektorski priključek (J5)

J5

```
1  2
0  0   vrata A STROBE
0  0   vrata A READY
0  0   vrata A bit 0
0  0   vrata A bit 1
0  0   vrata A bit 2
0  0   vrata A bit 3
0  0   vrata A bit 4
0  0   vrata A bit 5
0  0   vrata A bit 6
0  0   vrata A bit 7
0  0   vrata B READY
0  0   vrata B STROBE
0  0   vrata B bit 0
0  0   vrata B bit 1
0  0   vrata B bit 2
0  0   vrata B bit 3
0  0   vrata B bit 4
0  0   vrata B bit 5
0  0   vrata B bit 6
0  0   vrata B bit 7
39 40
```

JB3

```
    ----- MASA ------
15  0 0 0 0 0 0 0 0 1
16  0 0 0 0 0 0 0 0 2
    ! ! ! ! ! ! ! !
    ! ! ! ! ! ! ! - vrata A READ polariteta
    ! ! ! ! ! ! --- vrata B STROBE polariteta
    ! ! ! ! ! ----- vrata B READY polariteta
    ! ! ! ! -------- vrata A STROBE polariteta
    ! ! ! ---------- vrata B smer 1/2
    ! ! ------------ vrata B smer 1/2
    ! -------------- vrata A smer 1/2
    !---------------- vrata A smer 1/2
```

Nastavljanje elementa CTC (JB2)

JB2

```
                2 1
sistemska ura   O O  CLOCK/TRIGGER 0
      ZC-TO0    O O  CLOCK/TRIGGER 1
      ZC-TO1    O O  CLOCK/TRIGGER 2
      ZC-TO2    O O  CLOCK/TRIGGER 3
                8 7
```

Konektor za slikovni krmilnik (J6)

```
   --- MASA ---
9  O O O O O 1
10 O O O O O 2
   ! ! ! ! !____ ločen slikovni signal
   ! ! ! !______ ločena vodoravna sinhronizacija
   ! ! !_______ ločena navpična sinhronizacija
   ! !_________ neuporabljeno
   !___________ združen slikovni signal
```

Nastavljanje slikovne logike (JB7)

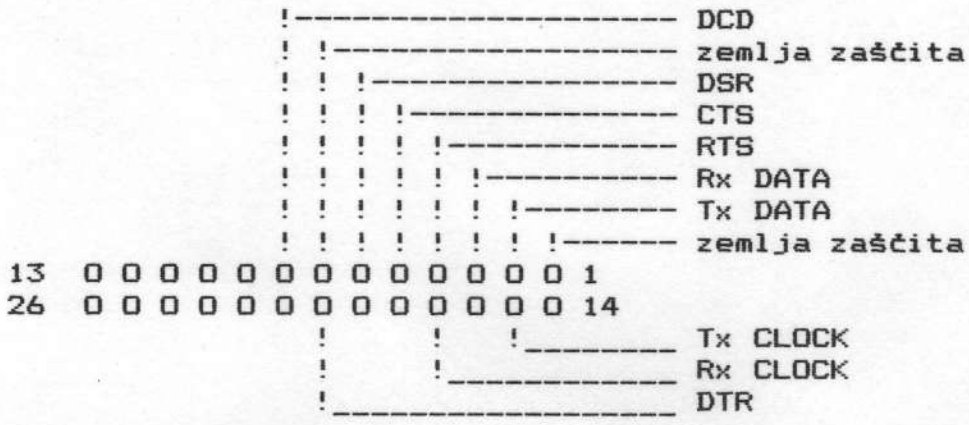
JB7

```
1  O  masa
2  O  nadzor negirane vodoravne sinhronizacije
3  O  vodoravna sinhronizacija - izhod
4  O  povezava za združen slikovni signal
5  O  povezava za združen slikovni signal
6  O  navpična sinhronizacija - izhod
7  O  masa
8  O  nadzor negirane navpične sinhronizacije
9  O  masa
10 O  nadzor negiranega slikovnega signala
11 O  masa
```

brez povezav imamo naslednjo konfiguracijo:

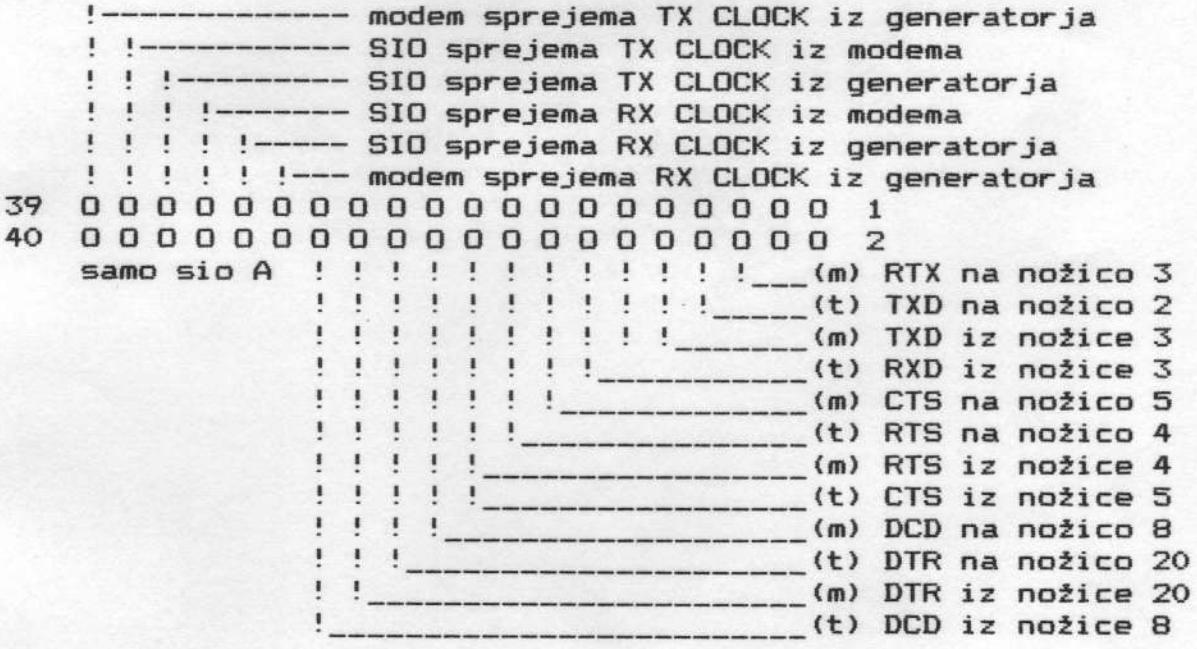
negiran slikovni signal
negativno navpično in vodoravno sinhronizacijo
za pozitivno horizontalno sinhronizacijo poveži 1 i 2
za pozitivno navpično sinhronizacijo poveži 8 in 9
za pozitiven slikovni signal poveži 10 in 11
za združen slikovni signal :
poveži 10 in 11, 3 in 4, 5 in 6
Vstavi R-21,R-25,R-26,Q-1,R-20 in 33pF.

Konektor za priključitev kanala A (J4) in B kanala (J3) sa SIO



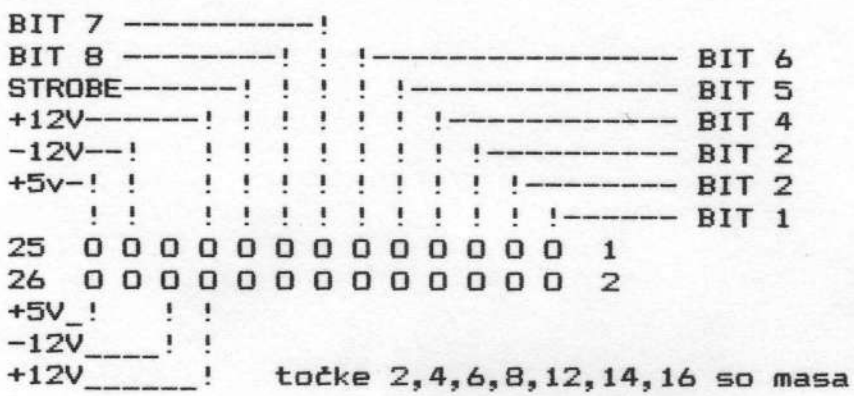
Določanje konfiguracije za serijska kanala (JB4=A JB5=B)

Samo kanal A lahko sprejema ali oddaja uro za določanje prenosne hitrosti.



Data Set Ready (DSR) je aktiven na obeh kanalih.
Legenda - (m) pomeni modem - naprava za komunikacijo
- (t) pomeni terminal - naprava, ki deluje kot terminal

Povezava konektorja za paralelno tipkovnico (J2)

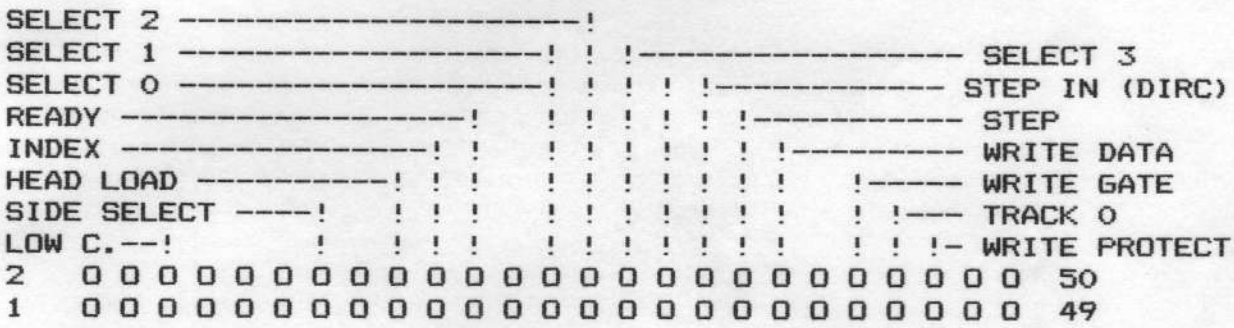


Določanje polaritete za krmilni signal iz tipkovnice

JB6

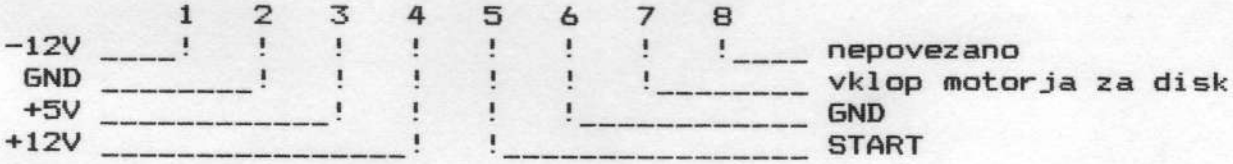
Povezava - sredina/levo - STROBE +
Povezava - sredina/desno - STROBE -

Priključni konektor za diskovne pogone (J1)



Vse lihe sponke so spojene z maso.

KONEKTOR ZA PRIKLJUCITEV NAPAJANJA (TB1)



ZAKLJUCEVANJE PREKINITVENE VERIGE

Če v računalnik MMS ne vgradimo vseh vezij družine Z80, ki jih tiskano vezje omogoča, morarmo poskrbeti za pravilno zaključitev prekinitvene verige po naslednjem ključu.

Če ne uporabljamo elementov CTC, GPIO in SIO, moramo povezati nožici 6 in 7 elementa U113.

Če dodamo SIO, moramo zgornjo povezavo odstraniti.

Če imamo SIO in CTC, nimamo pa GPIO, moramo povezati nožici 22 in 24 elementa U89.

Če od možnih dodatkov uporabimi samo CTC, moramo izvesti obe omenjeni povezavi.

Če uporabljamo samo GPIO, moramo povezati nožici 6 in 7 elementa U113.

Prioritetna veriga teče v naslednjem vrstnem redu:
SIO, PIO, GPIO in CTC.

Če uporabimo vse možne elemente, ne potrebujemo nobenih povezav!!!

DODATNE NASTAVITVE NA MMS

Hitrost delovanja procesorja

VAŽNO !!!

POGLAVJE O DODATNIH NASTAVITVAH PREBERI POTEM, KO RAČUNALNIK ŽE DALJ ČASA DELUJE BREZ NAPAKE. K REALIZACIJI PA PREIDI TAKRAT, KO BOŠ IMEL DOVOLJ IZKUŠENJ IZ MATERIALNE OPREME !!!! TO VELJA PREDVSEM ZA HITREJŠE DELOVANJA SISTEMA.

Delilnik osnovne frekvence je zasnovan tako, da je mogoče uporabljati tri različne hitrosti delovanja procesorja. S kremenčevim kristalom Y2 = 20Mhz lahko izbiramo med hitrostma 2.5 in 5 Mhz. Z uporabo kristala Y2 = 16 Mhz pa izberemo hitrost delovanja 4 Mhz. Uporabljena hitrost je predvsem odvisna od hitrosti pomnilnikov in od hitrosti družine Z80. Podrobno prouči, kakšna je ustrezna hitrost za različne vrste družine Z80. Za začetno oživljanje posebej priporočamo hitrost 2.5 Mhz. Ponavadi se izkaže, da je časovno najbolj kritičen element monitor ROM (2716). Zato priporočamo uporabo elementa 2732a. Za 4 Mhz mora imeti čas dostopa vsaj 333 ns in za 5 Mhz vsaj 258 ns. Seveda lahko koristno uporabimo samo zgornjo polovico tega elementa (pozicija U67) (A11=1).

Tabela za nastavljanje povezav pri določanju hitrosti delovanja

2.5 Mhz .... Y2 = 20 Mhz
P-Q
R-S
N-M

5 Mhz ..... Y2 = 20 Mhz
P-Q
R-S
O-M

4 Mhz ..... Y2 = 16 Mhz
O-M
O-S
N-Q

Krmiljenje dinamičnega pomnilnega polja

S prevezavami lahko določamo razmerje med časoma, ko je dinamično pomnilno polje aktivno, in samim aktivnim ciklom. Če imamo na voljo počasnejše pomnilnike, izberemo daljši aktivni in krajši neaktivni cikel. Če to ne pomaga, je naš pomnilnik (dinamični) prepočasen in moramo zmanjšati hitrost delovanja procesorja. Razmerje izbiramo na naslednji način:

Daljši čas za aktivni cikel, krajši čas med neaktivnimi cikli

K-H
J-I

Krajši čas za aktivne cikle, daljši čas med neaktivnimi cikli

L-H
K-I

Dodajanje pomnilnika RAM/ROM

Poziciji U69 in U70 lahko uporabimo za dodajanje dveh pomnilnikov kapacitete 2K x 8 vrste ROM ali RAM. Če povežemo B-A, je RAM, če pa povežemo B-C, je ROM.

Računalnik nam omogoča tudi širši nabor znakov, ki jih uporablja slikovni prikazovalnik. Za standarden nabor znakov povežemo U-V. Za razširjen nabor pa V-T; seveda pri tem uporabimo element 2732 namesto 2716, vsebino pa po želji sprogramiramo sami.

Uporaba funkcionalnega ekvivalenta za 8116.

Nekatere firme izdelujejo funkcionalni ekvivalent elementa 8116, vendar mora biti nožica 9 na +12V. To povezavo naredimo tako, da povežemo povezovana trna 1 in 2 v polju JB9.

VAŽNO!!!!

OMENJENE SPREMEMBE POSKUSAJ REALIZIRATI ZARES SAMO, ČE RAČUNALNIK ŽE DALJ ČASA DELUJE IN ČE IMAŠ DOVOLJ IZKUŠENJ Z MATERIALNO OPREMO. DRUGAČE SE RAJE NE DAJ ZAPELJATI.

PRIKLJUČEVANJE 5-PALČNIH DISKOV

Vezje WD1771 omogoča tudi priključevanje 5-palčnih diskov enojne gostote. Priključne sponke za 5-palčne diske niso v skladu s konektorjem, ki je v tiskanem vezju. Zato moramo konektor in priključne sponke skrbno proučiti in jih prilagoditi (zamenjava žic ploščatega priključevalnega kabla). Ko smo delo opravili, z merilnikom upornosti preverimo opravljeno delo.

Za krmiljenje 5-palčnih diskov potrebuje vezje 1771 frekvenco 1 MHz, podatkovni separator pa 2 Mhz. Signal 2Mhz spraviš na podatkovni separator tako, da spojiš povezovalna trna S in P (P je izvor, S pa ponor signala).

Signal 1 Mhz moramo dobiti iz signala 2 Mhz. Deliti ga moramo z 2. To storimo tako, da uporabimo prost register D na poziciji U108. Izvesti moramo naslednje povezave:

Sponka P - U108 nožica 11.
U108 nožica 8 - U108 nožica 12.
U108 nožica 12 - sponka Q.

Prevezave izpelji na spodnji strani tiskanega vezja.

Opravljeno delo preveri z osciloskopom. Signal na nožici 24 vezja 1771 mora nihati s frekvenco 1 Mhz, signal nožice 4 vezja U98 pa s frekvenco 2 Mhz.

NAVODILO ZA SESTAVLJANJE

( ) Skrbno preglej tiskano vezje z obeh strani. Pri proizvodnji tiskanih vezij se lahko pojavijo (sicer zelo redko) nezaželeni stiki ali pa prekinjene povezave. Nekaj minut aktivnega opazovanja lahko prihrani dolge ure muke polnega dela pri oživljanju računalnika.

( ) S pomočjo merilnika upornosti preveri,ali ni med napajalnimi povezavami stika. To storiš na konektorju za napajalne napetosti TB1.

( ) Pri naslednjih korakih si pomagaj s priloženo pozicijsko shemo.

OPOMBA ! Pri izdelavi računalnika priporočamo, naj vsa integrirana vezja ležijo v podnožjih. Posebej priporočamo profesionalna podnožja z okroglimi pozlačenimi kontakti. Plastika, v katero so kontakti pritrjeni naj bo oblikovana tako, da lahko vidimo tiskano vezje pod podnožjem. Če hočemo razširiti funkcionalne lastnosti računalnika, pa je uporaba profesionalnih podnožij obvezna za naslednje pozicije: U52-U59, U70, U80, U89 in U102.

( ) Vstavi in zaspajkaj 14-mestna podnožja na poziciji U9,10,11, 12,21,22,23,25,37,60,71 in 72. Pazi - nožica 1 je vedno obrnjena proti vrhu tiskanega vezja.

( ) Vstavi in zaspajkaj 14-mestna podnožja na poziciji U52,53, 56,66,76,77,78,79,81 in 87. (U52, 53 in 56 - v profesionalni izvedbi).

( ) Vstavi in zaspajkaj 14-mestna podnožja na poziciji U96,97, 99,100,101,103,104,105,108 in 110.

( ) Vstavi in zaspajkaj 14-mestna podnožja na poziciji U90,91, 92,93,94,95,112,114,115,116,117 in 118.

( ) Vstavi in zaspajkaj 16-mestna podnožja na poziciji U1-8, 13-20, 26-33 in 39-46.

( ) Vstavi in zaspajkaj profesionalna 16 mestna podnožja na pozicijah U54,55,57-59.

( ) Vstavi in zaspajkaj 16-mestna podnožja na pozicijah U24,34, 35,36,38,47,48,49,50,51,74 in 75.

( ) Vstavi in zaspajkaj 16-mestna podnožja na pozicijah U84,85, 86,98,106 in 109.

( ) Vstavi in zaspajkaj 18-mestna podnožja na pozicijah U61-64 in U107.

( ) Vstavi in zaspajkaj 20-mestna podnožja na pozicijah U65,82 in 83.

( ) Vstavi in zaspajkaj 24-mestna podnožja na pozicijah U67-70 in U73 (U70 - profesionalna izvedba)

( ) Vstavi in zaspajkaj 28-mestna podnožja na poziciji U88.

( ) Vstavi in zaspajkaj 40-mestna podnožja na lokacijah U80,89, 102,111 in 113 (U80,U89,U102 - profesionalna izvedba)

( ) Vstavi in zaspajkaj 11 povezovalnih trnov na poziciji JB7.

( ) Vstavi in zaspajkaj 3 povezovalne trne na poziciji JB6.

( ) Vstavi in zaspajkaj zaščitne kondenzatorje 0.1 mF na pozicijah: C1-C20, C25-C36, C39-C50, C55-C66, C70-C105, C107-C109, C111 in C116-C123.

( ) Vstavi in zaspajkaj kondenzatorje 2.2 mF (lahko večji) na pozicijah C21,22,37,38,51,52,67 in 68. Pazi na predznak.

( ) Vstavi in zaspajkaj 1.0 mF tantal na poziciji C115.

( ) Ponovno preveri polariteto vseh tantalovih oziroma elektrolitskih kondenzatorjev.

( ) Vstavi in zaspajkaj C23 - 10 nF keramični kondenzator.

( ) Vstavi in zaspajkaj C24 - 33 pF keramični kondenzator.

( ) Vstavi in zaspajkaj C106 - 47 pF kondenzator.

( ) Vstavi in zaspajkaj C110 - 100 pF kondenzator.

( ) Vstavi in zaspajkaj C113 - 33 pF kondenzator.

( ) Vstavi in zaspajkaj C114 - 3.3 nF kondenzator.

( ) Naslednji kondenzatorji so uporabljeni na združen slikovni signal (najpogosteje uporabljan). Če želimo porazdeljene slikovne signale, moramo pregledati poglavje o delovanju slikovnega krmilnika, da določimo prave vrednosti.

( ) Vstavi in zaspajkaj C142 - 33 pF kondenzator (samo za združen slikovni signal)

( ) Vstavi in zaspajkaj C53 - 100 pF kondenzator.

( ) Vstavi in zaspajkaj C STEP - 4.7 nF kondenzator

( ) Vstavi in zaspajkaj C69 - 180 pF kondenzator

( ) Vstavi in zaspajkaj R14-18, 4.7 K upore.

( ) Vstavi in zaspajkaj R22,35,36, 41-43 in 48,49, upori 10K.

( ) Na upor R48 zaspajkaj diodo BA520 ali podobno s katodo, obrnjeno v desno.

( ) Vstavi in zaspajkaj R37,38, upora 1 K.

( ) Vstavi 47 uF tantala na poziciji C140 in 141. Pazi na predznak.

( ) Vstavi in zaspajkaj R39,46,50; upori 220 ohmov.

( ) Vstavi in zaspajkaj R51; upor 220 ohmov 1/4 W.

( ) Vstavi in zaspajkaj R1 in R2; upora 820 ohmov.

( ) Vstavi in zaspajkaj R6-R13, R19; upori 33 ohmov.

( ) Vstavi in zaspajkaj RT = 3K, RY = 2.2K in RX = 33 ohmov.

( ) Vstavi in zaspajkaj R40, upor 1.2K.

( ) Vstavi in zaspajkaj R44, upor 100K.

( ) Vstavi in zaspajkaj R47, upor 330 ohmov.

( ) Vstavi in zaspajkaj R20, upor 75 ohmov. SAMO za združen slikovni signal.

( ) Vstavi in zaspajkaj R21, upor 1.5K. SAMO za združen slikovni signal.

( ) Vstavi in zaspajkaj R25, upor 3.9K. SAMO za združen slikovni signal.

( ) Vstavi in zaspajkaj R26, upor 4.3K. SAMO za združen slikovni signal.

( ) Vrednosti treh naslednjih uporov so namenjene za združen slikovni signal. Za porazdeljene vrednosti glej opis delovanja slikovnega prikazovalnika.

( ) Vstavi in zaspajkaj R3, upor 75K.

( ) Vstavi in zaspajkaj R4, upor 68K.

( ) Vstavi in zaspajkaj R5, upor 68K.

( ) Vstavi in zaspajkaj Q1, tranzistor 2N2222.

( ) Vstavi in zaspajkaj Q2, tranzistor 2N2907 (za delovanje 4 oz 5 Mhz vstavi MPS3640)

( ) Vstavi in zaspajkaj Y1, kremenčev kristal 13,9776 Mhz; lahko je tudi 14 Mhz.

( ) Vstavi in zaspajkaj Y2, kremenčev kristal 20 Mhz.

( ) Vstavi in zaspajkaj 26-polni konektor na poziciji J2. Spajkamo del konektorja, ki je krajši !!!

( ) Vstavi in zaspajkaj 50-polni konektor za priključitev gibkih diskov na poziciji J1.

( ) Vstavi in zaspajkaj dva sedemmestna (osempinska) uporovna stavka na pozicijah RP1 in RP2. Nožica 1 leži na levi strani pozicije za spajkanje. Če nimamo uporovnih stavkov, lahko po njihovem vzorcu zaspajkajmo navadne upore.

( ) Vstavi in zaspajkaj močnostni konektor TB1.

( ) Vstavi in zaspajkaj ZD1, 5.1V Zener diodo.

( ) Vstavi in zaspajkaj 10-polni konektor na poziciji J6. To je konektor za slikovni signal.

( ) Pred vstavljanjem integriranih vezij priključi na ploščo vse napajalne napetosti. S pomočjo natančnega voltmetra preveri napetost na nožici 1 podnožja U1. Mora biti -5V DC (5-odstotno odstopanje je dovoljeno).

( ) Preveri napajalno napetost na nožici 8 podnožja U1. Mora biti +12 V DC (5 odstotkov).

( ) Preveri napajalno napetost na pinu 22 konektorja J2. Mora biti -12 V DC (5-odstotkov).

( ) Preveri napajalno napetost na nožici 16 podnožja U1. Mora biti +5V (5-odstotno odstopanje).

( ) Napajalnik, ki ga nameravamo uporabiti za napajanje računalnika, mora biti kakovosten, po možnosti z napetostno in tokovno zaščito. Cenene izvedbe lahko postanejo zelo drage, če moramo zamenjati večjo količino okvarjenih integriranih vezij.

( ) IZKLJUČI NAPAJALNE NAPETOSTI !!!

( ) Vstavi povezovalne trne na pozicije P, Q, R, S, N, M.

( ) Poveži P-Q, R-S, N-M. Hitrost 2.5 Mhz.

( ) Vstavi in zaspajkaj povezovalne trne na pozicije K, H, J, L in I.

( ) Poveži L-I in K-H. Krajši čas za aktivni cikel v dinamičnem pomnilniku.

( ) Vstavi in zaspajkaj povezovalne trne na pozicije U, V in T.

( ) Poveži U-V. Generator znakov 2716.

( ) Vstavi 74LS00 na lokacijah U52 in 53. Nožice s številko 1 so obrnjene proti zgornjemu robu tiskanega vezja.

( ) Vstavi 74LS02 na lokaciji U87.

( ) Vstavi 74LS04 na lokacijah U66, 77, 103, 110.

( ) Vstavi 7404 na poziciji U11.

( ) Vstavi 7406 na lokaciji U105.

( ) Vstavi 74LS08 na lokacijah U37 in 104.

( ) Vstavi 7LS10 na lokaciji U10.

( ) Vstavi 74LS14 na lokacijah U101, 112, 114.

( ) Vstavi 74LS20 na lokaciji U56.

( ) Vstavi 74LS32 na lokacijah U25 in 81.

( ) Vstavi 7445 na lokaciji U109.

( ) Vstavi 74LS74 na lokacijah U9, 12, 60, 108.

( ) Vstavi 74LS123 na lokaciji U106.

( ) Vstavi 741S136 na lokaciji U94. (Za porazdeljen slikovni signal uporabi 74LS86.)

( ) Vstavi 74LS138 na lokacijah U55, 84, 85 in 86.

( ) Vstavi 74LS151 na lokaciji U75.

( ) Vstavi 74LS157 na lokacijah U47, 48, 49 in 50.

( ) Vstavi 74157 na lokacijah U58 in 59.

( ) Vstavi 74LS161 na lokaciji U24.

( ) Vstavi 74LS164 na lokaciji U76.

( ) Vstavi 74LS174 na lokacijah U35 in 74.

( ) Vstavi 74LS193 na lokaciji U98.

( ) Vstavi 74LS221 na lokacijah U38 in 51.

( ) Vstavi 74LS241 na lokacijah U65 in 82.

( ) Vstavi 74LS242 na lokacijah U99 in 100.

( ) Vstavi 74LS243 na lokacijah U71, 72, 78 in 79.

( ) Vstavi 74LS283 na lokacijah U34 in 36.

( ) Vstavi 74LS290 na lokaciji U97.

( ) Vstavi 74LS293 na lokacijah U96 in 22.

( ) Vstavi 74LS373 na lokaciji U83.

( ) Vstavi 74LS393 na lokacijah U21 in 23.

( ) Vstavi 8216 na lokaciji U54 in 57.

( ) Vstavi 2114 (250ns ali manj) na lokaciji U61-64.

( ) Vstavi EPROM generator znakov na lokacijo U73.

( ) Vstavi MIKRO-S EPROM na lokacijo U67.

( ) Vstavi Z80 na lokacijo U80.

( ) Vstavi PIO na lokacijo U111.

( ) Vstavi WD1771 na lokacijo U102.

( ) Vstavi 32 kosov 4116 na lokacije U1-8, 13-20, 26-33, 39-46. (Za procesor 2.5 Mhz morajo biti 300ns ali manj.)

( ) NE NADALJUJ, dokler nisi ponovno preveril, ali so vsi elementi na pravem mestu!

( ) Uporabi 26-polni kabel za priključitev paralelne tipkovnice ASCII. Preglej podatke o priključitvenih točkah. Prav tako tudi pravilno nastavi polariteto signala STROBE (glej navodilo o nastavitvah).

( ) Poveži povezovalne trne na konektorju JB7 tako kot za združen slikovni signal.

( ) Priključi slikovni prikazovalnik na točki 9 in 10 konektorja J6. Za povezavo uporabi koaksialni kabel 75 ohmov. (Za silo sta uporabni tudi navadni žici.)

( ) Na sponki 5 in 6 konektorja TB1 priključi tipko START.

( ) Pravilno zaključi prekinitveno verigo.

( ) Smo na koncu poti pred priključitvijo napajanja. Zdaj je zadnja priložnost za pregled doslej opravljenega dela .

( ) Vključi napajalne napetosti. HKRATI !!!

( ) Pritisni tipko START. Prikazovalnik mora ostati prazen.

( ) Pritisni tipko CR na tipkovnici. Osnovni kontrolni program (monitor) mora izpisati znak pripravljenosti. Zdaj je pravi trenutek, da proučiš, kako program deluje.

( ) Posebej priporočamo, da z uporabo vgrajenega testnega programa preverjaš delovanje pomnilnika vsaj 4 do 8 ur. Bolj ko boš testiral pomnilnik, bolj zanesljivo bo deloval. Delovanje testnega programa se morda zdi počasno, vendar program opravi ogromno število operacij na sekundo. Ugotovljeno je, da je vzrok za večino težav pri delovanju računalnika prav pomnilnik. Ker je monitor na naslovih F000H do FFFF, smemo testirati področje od 0000H do EFFFH. V nasprotnem primeru bomo povozili program in izhod iz tega je samo pritisk na tipko START.

( ) Ko smo preverili delovanje pomnilnika 60 K, zamenjamo med sabo integrirana vezja U1-8 in U39-46. Ponovno preverimo delovanje. Tako smo preskusili delovanje celotnega pomnilnega polja.

( ) Če slika na zaslonu ni čista, je lahko napaka v področju CPU ali pa v slikovni sekciji. Vezje za slikovno področje je samostojna celota in ga lahko testiramo posebej.

( ) Če se vam zdi, da računalnik deluje v redu, je čas, da priključimo diskovne pogone. Za priključevanje 5-palčnih diskov enojne gostote poglej poglavje o dodatnih nastavitvah.

( ) Domnevamo, da so pogoni usklajeni s standardom Shugart SA800 (8-palčni, enojna gostota z razporeditvijo priključnih sponk po vzoru stndarda firme Shugart). Eno stran 50-mestnega priključnega kabla spojimo z računalnikom (pazi na pravilno lego priključkov; če konektor vstavimo narobe, je možnost okvare precejšnja). Drugi del kabla pa spojimo z diskovnim(i) pogonom(i). Vsakemu pogonu moramo nastaviti svojo selekcijsko povezavo (prouči specifikacije, ki jih dobiš ob nakupu diskovnih pogonov).

( ) Priključi napajalno napetost.

( ) Pritisni tipko CR in preglej, ali računalnik še deluje.

( ) V pogon 0 vstavi KOPIJO (!!!!) sistemske diskete. Uporabi ukaz R00,01,01, da se prepričaš, ali pogon naloži glavo in ali 1771 sprejme podatke z diska. Če so napake, preglej razlago v poglavju o ukazu R.

( ) Uporabi ukaz R00,40,01 in preberi sektor na sledi 40H. Slišati moraš delovanje koračnega motorja, ko se premika. Če dobimo na zaslonu izpisano vsebino sektorja, vsa stvar najbrž deluje v redu. Če premik glave spremlja grozljivo škripanje, je najverjetneje hitrost za premik koračnega motorja prevelika. Po želji jo lahko spreminjaš tako, da v pomnilno lokacijo FF6AH vtipkaš pravo vrednost (podrobno si oglej priročnik o delovanju in krmiljenju elementa 1771.)

( ) Vstavi disketo s sistemom v pogon 0 in uporabi ukaz B.

( ) CPM* mora javiti znak pripravljenosti. Če se to ne zgodi (občutek, da je računalnik zaspal), potem vpisan operacijski sistem ne vsebuje pravega programa z imenom BIOS.

( ) Če se CPM* ne javi, lahko z monitorskim ukazom R preveriš, ali je v pomnilnik naložen program, ki leži na sledi 0, sektor 1.

( ) Ko se CPM* naposled javi, prični proučevati velike kupe literature in uporabniških programov za ta izredno popularni operacijski sistem.

DOPOLNITEV OSNOVNEGA NABORA FUNKCIJ

CTC

( ) Vstavi in zaspajkaj 8-polni konektor na poziciji JB2.

( ) Vstavi in zaspajkaj 28-polno podnožje na poziciji U88.

( ) Vstavi integrirano vezje CTC na poziciji U88.

( ) Preglej navodilo o naslavljanju funkcij CTC.

( ) Če hočeš, da CTC izključuje disk, poveži povezovalne točke 3 in 4 ter 7 in 8 na konektorju JB2.

SIO

( ) Vstavi in zaspajkaj 40-polno podnožje na poziciji U113.

( ) Vstavi in zaspajkaj štiri 14-polna podnožja na pozicijah U115-118.

( ) Vstavi in zaspajkaj 4,7 K upora na pozicijah R45 in 52.

( ) Vstavi in zaspajkaj 16 kondenzatorjev 390 pF na pozicijah C124-139.

( ) Vstavi in zaspajkaj dva 40-polna konektorja na pozicijah J3 in J4.

( ) Vstavi in zaspajkaj dva 26-polna konektorja na pozicijah J3 in J4.

( ) Vstavi dva MC1488 na pozicijah U117 in 118.

( ) Vstavi dva MC14899 na pozicijah U115 in 116.

( ) Vstavi SIO na pozicijo U113.

( ) Vstavi in zaspajkaj 18-polno podnožje na poziciji U107.

( ) Vstavi in zaspajkaj kristal na poziciji Y3. (5.0688 Mhz)

( ) Vstavi 8116 na pozicijo U107.

( ) Preglej poglavje o povezovanju serijskih naprav.

( ) Za priključitev tipične serijske naprave (Kopa, Paka,...) na kanal B poveži sponke 5-6 in 9-10 na konektorju JB5.

GPIO

( ) Vstavi in zaspajkaj R27-34, 10 K upore.

( ) Vstavi in zaspajkaj 40-polno podnožje na poziciji U89.

( ) Vstavi in zaspajkaj pet 14-polnih podnožij na pozicije U90-93 in 95.

( ) Vstavi in zaspajkaj 40-polni konektor na poziciji J5.

( ) Vstavi in zaspajkaj 16-polni konektor na poziciji JB3.

( ) Vstavi 74LS86 na poziciji U96.

( ) Vstavi štiri &$LS234 na pozicijah U90-93.

( ) Vstavi PIO na pozicijo U89.

( ) Preglej poglavje o priključevanju na GPIO.

PORAZDELJEN SLIKOVNI SIGNAL

( ) Preglej poglavje o povezovanju slikovne sekcije.

RAČUNALNIK NE DELUJE — kaj zdaj?

Statistično je dokazano, da računalnik MMS v enem od dvajsetih primerov ne bo deloval ob prvem poskusu. Kaj lahko storimo? Kot osnovo vzemimo, da deluje procesor s frekvenco 2.5 Mhz.

Najprej nekaj osnov

Ko priključimo napajalno napetost ali pritisnemo tipko START, začne procesor izvajati program na naslovu 0000H v čitalnem pomnilniku U67. Prvih 16 ukazov premakne ostanek sistemskega programa (monitor) na pomnilno področje z začetnim naslovom F000H. Po selitvi teče program na naslovu F000H. Če je bilo kopiranje monitorja izvedeno pravilno in če v računalniku ni bistvenih napak v tem delu vezja, bo monitor oživel. Predno pa izpiše znak pripravljenosti, mora opraviti kup dodatnih stvari:
Izklopiti mora bralni pomnilnik na naslovu 0000H, zbrisati slikovni pomnilnik, inicializirati serijski kanal, časovnik in krmilnik za gibke diske. Če je serijski kanal instaliran, prične v zanki otipavati serijski kanal in tipkovnico, kateri od njiju mu bo prvi poslal znak CR. To ciklično preverjanje lahko opazujemo kot negativne impulze na nožicah za izbor tipkovnice (U111 nožica 4) in serijskega kanala (U113 nožica 35).

Ali procesor deluje?

Prva stvar, ki jo moramo preveriti na nedelujočem računalniku, je delovanje centralnega procesorja (U80). Ugotoviti moramo, ali sploh deluje. To najlaže preverimo na nožici 27 (M1), ki mora nihati. Takt osilacij ni pomemben. Če ne niha, procesor ne izvaja ukazov. Preveriti moramo, ali ima procesor na voljo urne signale 2.5 Mhz (nožica 6), ali so nožice 26 (RST), 24 (WAIT) in 25 (BUSREQ) na nivoju logične enice (4.0V - 5V). Če odkrijemo napako, moramo preveriti pripadajoče komponente in povezave. Lahko smo zamenjali med seboj tranzistorja Q1 in Q2. Namen tranzistorja Q2 je prilagoditi obliko urnega signala procesorju. Q2 mora biti 2N2907 ali ekvivalent. Če hočemo, da procesor deluje s frekvenco 4 ali 5 Mhz, moramo Q2 zamenjati s hitrejšo verzijo tranzistorja, kot naprimer MPS 3460.

Ko procesor deluje, lahko ugotovimo, da je slikovni pomnilnik zelo priročna diagnostična naprava. Ponavljajoč vzorec kvadratov in dvopičij pomeni, da skladovni pomnilnik ne deluje; procesor sicer deluje, ne more pa izvajati sistemskega monitorja. Pogosto je vzrok za to zvita nožica v pomnilnem polju. Tako napako težko odkrijemo - zato pozor!! Morda smo tudi zamenjali elementa 74157 (U58 in U59) z elementom 74LS157. Probleme nam lahko povzročajo nezaspajkane nožice, stiki med nožicami in povezavami, okvarjen spominski element v zgornji vrstici (U1 - U8). U80 in U111 sta edini 40-mestni integrirani vezji, ki ju potrebujemo za izpis znaka, da je monitor pripravljen.

Problemi, ki jih lahko opazimo na zaslonu

Če dobimo na zaslonu več kot en znak pripravljenosti ali, če zaslon ni popolnoma zbrisan, je napaka največkrat pri elementih 74LS157. Ponovno preglejmo, ali ni kakšna nožica zvita, nezaspajkana, spojena s sosednjo. Ko nam ob vseh izkušnjah zmanjka idej, lahko posumimo o delovanju samega integriranega vezja.

Če zaslon počasi valovi, preverimo slikovni kristal. Ta mora biti 13.9776 Mhz, odlično pa se tudi obnese 14 Mhz. Takšno vrednost zagotavlja navpično sinhronizacijo 50 Hz. Včasih mora biti C24 10 pF ali R2 470 ohmov. Razmazane točke na slikovnem zaslonu govorijo o šumu v slikovnem signalu. Dodati moramo kak zaščitni elektrolit na napajalno napetost 5V v bližini področja za generiranje slike. Če znakom na zaslonu manjka prva polovica, smo verjetno pozabili vstaviti in priciniti upor R21.

Tipkovnica

Bit z najvišjo utežjo mora biti vedno logična 0, sicer bo računalnik na kak ukaz odgovoril z znakom začudenja. Tudi okvarjena elementa U112 in 114 lahko povzročata začudenje.

Problem gibkih diskov

Če smo pomotoma nastavili dva diskovna pogona kot pogona s številko 0, bomo gotovo uničili sistemske sledi na disketi (sled 0 in 1).

Kaj se nam še lahko pripeti?

Če zmanjka napajalne napetosti -12V, bo 1771 (U102) povročil, da bo glava na gibkem disku - če je slučajno pritisnjena ob disketo - popacala sled, nad katero je.

Če preide napajanje -5V potencial mase (prevzame pozitivno vrednost), lahko pričakujemo, da bo kateri od spominskih elementov okvarjen.

Test spominskih elementov, ki so vgrajeni, ni stoodstotno zanesljiv. Če se programi čudno obnašajo in je napajanje v redu, potem lahko sumimo pomnilno polje. Zamenjajmo elemente med seboj in poglejmo, ali so napake ostale enake.

Povedati moramo, da 9 računalnikov od desetih ne deluje zato, ker je morda nožica zvita ali pa ni zaspajkana.

<u>ŠE NEKAJ NE RAVNO NEPOMEMBNIH PODROBNOSTI</u>

Sistema CPM ne boste mogli naložiti ali prebrati zapisa na disketi z ukazom R, če diskovni pogoni niso priključeni.

NE TESTIRAJ pomnilnika nad EFFFH, ker boš ubil monitor.

Če se sistem prične obnašati čudno, takoj preveri napajalne napetosti !!!

Kondenzatorji, označeni z 101J, so 100 pF, označba 103 pa pomeni 0.01 uF.

Ojačevalci serijskih linij 1488 in 1489 vedno delujejo z rahlo vročino.

Če boš priključil kak konektor narobe, je precej verjetno, da bo nekaj elementov prešlo v okvarjeno agregatno stanje.

Slabo opravljeno spajkanje (lotanje, cinjenje, švasanje) lahko povzroči več problemov kot katerakoli druga stvar.

KOSOVNICA SESTAVNIH DELOV

POPIS INTEGRIRANIH VEZIJ PO POZICIJAH

74LS00	= U52,53
74LS02	= U87
74LS04	= U66,77,103,110
7404	= U11
7406	= U105
74LS08	= U37,104
74LS10	= U10
74LS14	= U101,112,114
74LS20	= U56
74LS32	= U25,81
7445	= U109
74LS74	= U9,12,60,108
74LS123	= 106
74LS136	= U94
74LS138	= U55,84,85,86
74LS151	= U75
74LS157	= U47,48,49,50
74157	= U58,59
74LS161	= U24
74164	= U76
74LS174	= U35,74
74LS193	= U98
74LS241	= U65,82
74LS242	= U99,100
74LS221	= U38,51
74LS243	= U71,72,78,79,90,91,92,93
74LS283	= U34,36
74LS290	= U97
74LS293	= U96,22
74LS373	= U83
74LS393	= U21,23
8216	= U54,57
2114	= U61,62,63,64
2716znk.gen.	= U73
2716mikro-s	= U67
Z80cpu	= U80
Z80PIO	= U111,89
WD1771	= U102
4116	= U1-8,13-20
Z80SIO	= U113
Z80CTC	= U88 26-33,39-46
13.9776MHZ	= Y1
20MHZ	= Y2
5.0688MHZ	= Y3
2N2222	= Q1(video)
2N2907	= Q2(cpu)
1N751	= ZD
DIODA BA520	= na uporu R48 (katoda desno)

OPIS POLPREVODNIŠKIH ELEMENTOV

element	količina	opis
74LS00	2	ŠTIRI NAND VRATA Z DVEMA VHODOMA
74LS02	1	ŠTIRI OR VRATA Z DVEMA VHODOMA
		ŠEST NEGATORJEV
7404	1	---
7406	1	ŠEST NEGATORJEV Z ODPRTIM KOLEKTORJEM
74LS08	2	ŠTIRI AND VRATA Z DVEMA VHODOMA
74LS10	1	TRI NAND VRATA S TREMI VHODI
74LS14	3	ŠEST NEGATORJEV S SCMITT T. VHODOM
74LS20	1	DVOJA NAND VRATA S ŠTIRIMI VHODI
74LS32	2	ŠTIRI NOR VRATA Z DVEMA VHODOMA
7445	1	BCD DEKODER IN OJACEVALEC
74LS74	4	DVOJNI DVOSTABILNI MULTIVIBRATOR
74LS86	1	ŠTIRI XOR VRATA Z DVEMA VHODOMA
74136	1	ŠTIRI XOR VRATA Z DVEMA VHODOMA
74LS123	1	DVOJNI MONOSTABILNI MULTIVIBRATOR
74LS138	4	DVOJIŠKI DEKODER
74LS151	1	MULTIPLEKSOR
74LS157	4	MULTIPLEKSOR
74157	2	MOCNOSTNA VERZIJA 74LS157
74LS161	1	SINHRON DVOJIŠKI ŠTEVEC
74LS164	1	8 MESTNI POMIKALNI REGISTER
74LS174	2	6 DVOSTABILNIH MULTIVIBRATORJEV
74LS193	1	SINHRON DVOJIŠKI ŠTEVEC
74LS221	2	DVOJNI MONOSTABILNI MULTIVIBRATOR
74LS241	2	OJACEVALEC TREH STANJ
74LS242	2	4 MESTNI OJACEVALEC TREH STANJ - NEGATOR
74LS243	8	4 MESTNI OJACEVALEC TREH STANJ
74LS290	1	BCD ŠTEVNIK
74LS283	2	ŠTIRI BITNI SEŠTEVALEC
74LS293	1	DVOJIŠKI ŠTEVNIK
74LS373	1	8 MESTNI DVOSTABILNI MULTIVIBRATOR
74LS393	2	DVOJNI ŠTIRI BITNI BINARNI ŠTEVNIK
MC1488	2	RS232 ODDAJA
MC1489	2	RS232 SPREJEM
COM 8116	1	DVOJNI GENERATOR PRENOSNE HITROSTI
8216	2	OJACEVALAC VODILA (intel/signetics/ nec)
2114	4	1 K x 4 STATICNI POMNILNIK
2716	1	GENERATOR ZNAKOV
2716	1	MIKRO - S monitor
Z80	1	CPU
Z80-PIO	2	PIO
Z80SIO	1	SIO
Z80CTC	1	CTC
FD 1771	1	KRMILNIK ZA GIBKE DISKE
4116	32	DINAMICNI POMNILNIK 16K
2N2907	1	PNP TRANSISTOR
2N2222	1	NPN TRANSISTOR
1N751	1	5.1 V ZENER DIODA 1/2 W
BA520	1	SILICIJEVA DIODA

UPOROVNI ELEMENTI

količina	vrednost
10	33 ohm 1/8 W
2	820 ohm 1/8 W
7	4.7Kohm 1/8 W
8	10 K 1/4 W
1	1.2 K 1/4 W
3	220 ohm 1/8 W
1	220 ohm 1/4 W
1	330 ohm 1/8 W
2	1.0 K 1/8 W
1	100 K 1/8 W
1	75 ohm 1/4 W
1	1.5 K 1/8 W
1	3.9 K 1/8 W
1	4.3 K 1/8 W
2	68 K 1/8 W
2	7-mestni uporovni stavek (8 pinski) nožica 1 je skupna, 1/8 W
1	3 kom 1/8 W
1	2.2 kom 1/8 W
1	75 kom 1/8 W

KONDENZATORJI

količina	vrednost	
104	10 - 100 nF	diskovni ali večslojni 16V
1	10 nF	keramika
3	33 pF	keramika
1	47 pF	keramika
1	3.3 nF	keramika
8	2.2 uF 20 V	tantal
2	47 uF 10 V	elektrolit
1	1.0 uF 10 V	elektrolit
2	100 pF	keramika
1	4.7 nF	
1	180 pF	

KREMENCEVI KRISTALI

količina	vrednost
1	13.9776 Mhz ali 14 Mhz
1	20 Mhz (2.5 in 5 Mhz verzija)
1	16 Mhz (4 Mhz verzija)
1	5.0688 Mhz

PODNOŽJA IN KONEKTORJI

količina	vrednost
3	26-polni konektor
1	50-polni konektor
44 (41+3)	14-mestno podnožje
55 (50+5)	16-mestno podnožje
5	18-mestno podnožje
3	20-mestno podnožje
5 (4+1)	24-mestno podnožje
5 (2+3)	40-mestno podnožje
1	28-mestno podnožje
1	8-mestna močnostna vrstična sponka

150 povezovalnih enovrstičnih trnov, zalitih v plastiko

Število za znakom + v oklepaju pomeni število profesionalnih podnožij.

POPIS UPOROVNIH ELEMENTOV PO POZICIJAH

pozicija	vrednost
R-1	820 ohmov
R-2	820 ohmov
R-3	75K (vodoravna sinhr.)
R-4	68K (navpična sinhr.)
R-5	68K (vodoravna zakasnitev)
R-6	33 ohmov
R-7	33 ohov
R-8	33 ohmov
R-9	33 ohmov
R-10	33 ohmov
R-11	33 ohmov
R-12	33 ohmov
R-13	33 ohmov
R-14	4.7K
R-15	4.7K
R-16	4.7K
R-17	4.7K
R-18	4.7K
R-19	33 ohmov
R-20	75 ohmov
R-21	1.5K
R-22	10K
R-25	3.9K
R-26	4.3K
R-27	10K
R-28	10K
R-29	10K
R-30	10K
R-31	10K
R-32	10K
R-33	10K
R-34	10K
R-35	10K

R-36	10K
R-37	1K
R-38	1K
R-39	220 ohmov
R-40	1.2K
R-41	10K
R-42	10K
R-43	10K
R-44	100K
R-45	4.7K
R-46	220 ohmov
R-47	330 ohmov
R-48	10K
R-49	10K
R-50	220 ohmov
R-51	220 ohmov
R-52	4.7K
RT	3.0 K
RX	33 ohmov
RY	2.2 K

POPIS KONDENZATORJEV PO POZICIJAH

pozicija	vrednost

opomba
Vsi kondenzatorji, ki niso na spodnjem spisku, so lahko 0.01 do 0.1 uF, 16V ali več.

C-21	22uF (ali več) 20V
C-22	2.2uF (ali več) 20V
C-23	10 nF
C-24	33 pF
C-37	2.2 uF (ali več) 20V
C-38	2.2 uF (ali več) 20V
C-51	2.2 uF (ali več) 20V
C-52	2.2 uF (ali več) 20V
C-53	100 pF (vodoravna sinhronizacija)
C-54	4700 pF (navpična sinhronizacija)
C-67	2.2 uF (ali več) 20V
C-68	2.2 uF (ali več) 20V
C-69	180 pF (vodoravna zakasnitev)
C-106,107	47 pF
C-110	100 pF
C-112	10 nF
C-113	33 pF
C-114	3.3 nF
C-115	1.0 uF 10V
C-124 do C-139	390 pF
C-140	47 uF 10V
C-141	47 uF 10V
C-142	33 pF

Za združen slikovni signal izberemo naslednje vrednosti

R-3	68 K
R-4	100 K
R-5	39 K
C-53	180 pF
C-54	4700 pF
C-69	470 pF

```
                        ;*********************************************************
                        ;*                                                       *
                        ;*  MICRO SLOVENIJA MONITOR ROM - NON RELOCATABLE VER.   *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                        ;
0000'                           ASEG
F000                    ROM     EQU     0F000H          ;START OF 2K ROM
FF00                    RAM     EQU     0FF00H          ;START OF 256 BYTE RAM
3000                    CRTMEM  EQU     3000H           ;BASE OF 4K CRT MEMORY
                        ;
                        ;
                                ORG     ROM
                    C           INCLUDE INIT.MAC
                    C       ;*********************************************************
                    C       ;*                                                       *
                    C       ;*      COLD START INITIALIZATION ROUTINE FOR            *
                    C       ;*      CONFIGURING THE SYSTEM AFTER A POWER-ON          *
                    C       ;*      OR PUSHBUTTON RESET.                             *
                    C       ;*                                                       *
                    C       ;*********************************************************
                    C       ;
                    C       ;
                    C       ;       -- MONITOR ENTRY POINT TABLE --
                    C       ;
F000    C3 F02A     C       COLD:   JP      INIT            ;MONITOR COLD ENTRY POINT
F003    C3 F12B     C       WARM:   JP      PROMPT          ;MONITOR WARM ENTRY POINT
F006    C3 F431     C       CONST:  JP      KBDST           ;CONSOLE STATUS VECTOR
F009    C3 F439     C       CONIN:  JP      KBDIN           ;CONSOLE INPUT VECTOR
F00C    C3 F520     C       CONOUT: JP      CRTOUT          ;CONSOLE OUTPUT VECTOR
F00F    C3 F520     C               JP      CRTOUT          ;CRT OUTPUT VECTOR
F012    C3 F4E8     C               JP      SIOST           ;SIO CHANEL B STATUS VECTOR
F015    C3 F4F0     C               JP      SIOIN           ;SIO CHANEL B INPUT VECTOR
F018    C3 F4FE     C               JP      SIOOUT          ;SIO CHANEL B OUTPUT VECTOR
F01B    C3 F6B3     C               JP      SELECT          ;DISK DRIVE SELECT
F01E    C3 F6EB     C               JP      HOME            ;HOME R/W HEAD
F021    C3 F6FD     C               JP      SEEK            ;SEEK TO TRACK
F024    C3 F72C     C               JP      READ            ;READ SECTOR
F027    C3 F721     C               JP      WRITE           ;WRITE SECTOR
                    C       ;
                    C       ;
                    C       ;
                    C       ;       DO A SHORT POST-RESET TIME DELAY. ALSO INITIALIZES THE
                    C       ;       STACK POINTER AND FILLS THE MONITOR SCRATCH RAM WITH ZEROS.
                    C       ;
F02A    F3          C       INIT:   DI
F02B    21 FF00     C               LD      HL,RAM          ;POINT TO START OF MONITOR RAM
F02E    36 00       C       INIT1:  LD      (HL),0          ;FILL 256 BYTE SPACE WITH ZEROS
F030    F9          C               LD      SP,HL           ;DO SOMETHING USEFUL TO ADD DELAY
F031    2C          C               INC     L
F032    20 FA       C               JR      NZ,INIT1        ;LOOP TAKES ABOUT 4 MILLISECONDS
                    C       ;
                    C       ;       INITIALIZE THE Z-80 FOR INTERRUPT MODE #2
                    C
F034    7C          C               LD      A,H
```

```
F035  ED 47     C           LD    I,A           ;LOAD I REG WITH MSB OF VECTOR TABLE
F037  ED 5E     C           IM    2             ; AND SELECT INTERRUPT MODE 2
                C   ;
F039  CD F5EC   C           CALL  CLRSCN        ;FILL THE CRT MEMORY WITH BLANKS
                C   ;
                C   ;       STORE ANY NON-ZERO VALUES FOR VARIABLES IN MEMORY
                C   ;
F03C  21 F0D3   C           LD    HL,INTAB      ;POINT TO DEFAULT VARIABLE TABLE
F03F  06 00     C   INIT2:  LD    B,0
F041  4E        C           LD    C,(HL)        ;BC=DATA BLOCK BYTECOUNT
F042  23        C           INC   HL
F043  5E        C           LD    E,(HL)        ;DE=DESTINATION FOR DATA
F044  23        C           INC   HL
F045  56        C           LD    D,(HL)
F046  23        C           INC   HL
F047  ED B0     C           LDIR                ;COPY DATA @ HL TO VARIABLES @ DE
F049  CB 7E     C           BIT   7,(HL)
F04B  28 F2     C           JR    Z,INIT2 ;LOOP AGAIN IF NOT AT END OF TABLE
                C   ;
                C   ;       INITIALIZE THE PROGRAMMABLE I/O DEVICES
                C   ;
F04D  23        C           INC   HL            ;POINT TO I/O INIT DATA TABLE
F04E  46        C   INIT3:  LD    B,(HL)        ;B=INIT LOOP BYTECOUNT
F04F  23        C           INC   HL
F050  4E        C           LD    C,(HL)        ;C=DEVICE CONTROL PORT#
F051  23        C           INC   HL
F052  ED B3     C           OTIR                ;SEND DATA @ HL TO PORT @ C
F054  CB 7E     C           BIT   7,(HL)        ;TEST FOR TABLE END MARKER
F056  28 F6     C           JR    Z,INIT3 ;LOOP AGAIN IF NOT AT END
                C   ;
                C   ;       DETERMINE IF CONSOLE I/O CONFIGURATION WILL BE FOR THE
                C   ;       ON-BOARD CRT AND KEYBOARD OR AN EXTERNAL SERIAL TERMINAL.
                C   ;
F058  ED 78     C           IN    A,(C)         ;TEST SIO READ REGISTER 2 TO DETERMINE
F05A  FE 06     C           CP    00000110B     ; IF THE SIO IS INSTALLED IN THE BOARD
F05C  20 12     C           JR    NZ,PARALL     ;SKIP CONFIGURATION TEST IF NO SIO
F05E  DB 1E     C           IN    A,(KBDDAT)    ;MAKE SURE KBD PIO 'READY' IS RESET
F060  06 10     C           LD    B,00010000B   ;B=RESET SIO EXTERNAL STATUS COMMAND
F062  ED 41     C   DECIDE: OUT   (C),B         ;TEST FOR ARRIVAL OF A SERIAL
F064  ED 78     C           IN    A,(C)         ; INPUT CHARACTER START BIT
F066  CB 67     C           BIT   4,A
F068  20 0E     C           JR    NZ,BAUD ;EXIT LOOP IF START BIT DETECTED
F06A  DB 1C     C           IN    A,(BITDAT)
F06C  CB 5F     C           BIT   3,A           ;TEST FOR DATA READY STROBE FROM
F06E  20 F2     C           JR    NZ,DECIDE     ; PARALLEL KBD AND LOOP IF INACTIVE
F070  DB 1E     C   PARALL: IN    A,(KBDDAT)    ;DISCARD FIRST KEYBOARD CHARACTER
F072  3E 83     C           LD    A,10000011B
F074  D3 1F     C           OUT   (KBDCTL),A    ;ENABLE INTERRUPTS FROM KBD PIO
F076  18 2D     C           JR    SIGNON  ;GO SIGNON WITH BULIT-IN CONSOLE I/O
                C   ;
                C   ;
                C   ;       AUTOMATIC BAUD RATE SETTING ROUTINE FOR SIO
                C   ;
F078  AF        C   BAUD:   XOR   A
F079  ED 41     C   BAUD1:  OUT   (C),B
F07B  ED 50     C           IN    D,(C)         ;READ SIO STATUS REGISTER
```

```
F07D  CB 62       C            BIT   4,D            ;TEST THE SYNC/HUNT BIT
F07F  28 F8       C            JR    Z,BAUD1 ;LOOP UNTIL IT CHANGES STATE
F081  3C          C    BAUD2:  INC   A
F082  ED 41       C            OUT   (C),B          ;RESET REGISTER #0 FLAGS AGAIN
F084  ED 50       C            IN    D,(C)          ; AND LOOP TIMING THE SYNC/HUNT BIT
F086  CB 62       C            BIT   4,D
F088  20 F7       C            JR    NZ,BAUD2       ;REPEAT UNTIL BIT CHANGES AGAIN
F08A  21 F0CA     C            LD    HL,RATES-1
F08D  23          C    BAUD3:  INC   HL             ;INDEX INTO BAUD RATE TABLE
F08E  17          C            RLA                  ; USING COUNT DERRIVED IN A
F08F  30 FC       C            JR    NC,BAUD3
F091  7E          C            LD    A,(HL)         ;GET BAUD RATE CONTROL BYTE FROM
F092  D3 0C       C            OUT   (BAUDB),A      ; TABLE AND OUTPUT TO COM-8116 TIMER
F094  CD F4F0     C            CALL  SIOIN          ;DISARD FIRST SERIAL INPUT CHARACTER
F097  3E 01       C            LD    A,1
F099  D3 07       C            OUT   (SIOCPB),A     ;RE-PROGRAM SIO B TO GENERATE
F09B  3E 1C       C            LD    A,00011100B    ; INTERRUPTS ON RECIEVED DATA,
F09D  D3 07       C            OUT   (SIOCPB),A     ; PARITY DOES NOT AFFECT VECTOR
F09F  21 F4FE     C            LD    HL,SIOOUT
F0A2  22 F00D     C            LD    (CONOUT+1),HL  ;RE-DIRECT CONSOLE OUTPUT TO SIO
                  C    ;
                  C    ;       PRINT SIGNON MESSAGE
                  C    ;
F0A5  FB          C    SIGNON: EI
F0A6  CD F3EC     C            CALL  PNEXT
F0A9  0D 0A       C            DEFB  CR,LF
F0AB  3C 2E 2E 2E C            DEFM  '<... MIKRO SLOVENIJA ....>'
F0AF  20 4D 49 4B C
F0B3  52 4F 20 53 C
F0B7  4C 4F 56 45 C
F0BB  4E 49 4A 41 C
F0BF  20 2E 2E 2E C
F0C3  2E 3E       C
F0C5  0D 0A       C            DEFB  CR,LF
F0C7  04          C            DEFB  EOT
F0C8  C3 F003     C            JP    WARM           ;GO ENTER MONITOR
                  C    ;
                  C    ;
                  C    ;
                  C    ;       BAUD RATE CONSTANTS FOR COM 8116 BAUD RATE GENERATOR
                  C    ;
F0CB  05          C    RATES:  DEFB  0101B          ;   300 BAUD
F0CC  06          C            DEFB  0110B          ;   600 BAUD
F0CD  07          C            DEFB  0111B          ;  1200 BAUD
F0CE  0A          C            DEFB  1010B          ;  2400 BAUD
F0CF  0C          C            DEFB  1100B          ;  4800 BAUD
F0D0  0E          C            DEFB  1110B          ;  9600 BAUD
F0D1  0F          C            DEFB  1111B          ; 19200 BAUD
F0D2  0F          C            DEFB  1111B          ; 19200 BAUD
                  C    ;
                  C    ;
F0D3              C    INTAB   EQU   $              ;INITIALIZATION DATA TABLES
                  C    ;
                  C    ;
                  C    ;       INITIALIZE THE Z-80 'I' REGISTER INTERRUPT VECTOR TABLE
                  C    ;
```

```
F0D3  02           C            DEFB   2
F0D4  FF1A         C            DEFW   SYSVEC+2
F0D6  F48C         C            DEFW   KEYSRV          ;PARALLEL KEYBOARD INTERRUPT VECTOR
                   C
F0D8  02           C            DEFB   2
F0D9  FF16         C            DEFW   CTCVEC+6
F0DB  F49F         C            DEFW   TIMER           ;ONE SECOND TIMER INTERRUPT VECTOR
                   C
F0DD  04           C            DEFB   4
F0DE  FF04         C            DEFW   SIOVEC+4
F0E0  F4AF         C            DEFW   SIOINT          ;SIO RECIEVE INTERRUPT VECTOR
F0E2  F4CF         C            DEFW   SIOERR          ;SIO PARITY, OVERRUN & FRAMING ERROR
                   C    ;
                   C    ;       INITIALIZE DISK I/O DRIVER VARIABLES
                   C    ;
F0E4  08           C            DEFB   8
F0E5  FF65         C            DEFW   UNIT
F0E7  FF           C            DEFB   255             ;FLAG ALL DRIVES AS DE-SELECTED
F0E8  FF FF FF FF  C            DEFB   255,255,255,255 ;CLEAR HEAD POSITION TABLE
F0EC  00           C            DEFB   00000000B       ;SELECT FASTEST SEEK SPEED
F0ED  80           C            DEFB   128             ;SELECT 128 BYTE SECTOR LENGTH
F0EE  1E           C            DEFB   30              ;SET MOTOR TURN-OFF TIMER
                   C    ;
                   C    ;       INITIALIZE THE CRT DISPLAY CURSOR
                   C    ;
F0EF  02           C            DEFB   2
F0F0  FF75         C            DEFW   CHRSAV
F0F2  20           C            DEFB   ' '
F0F3  5F           C            DEFB   '_'             ;USE NON-BLINKING UNDERSCORE
                   C    ;
                   C    ;       SET DEFAULT 'SOFTWARE' INTERRUPT VECTORS
                   C    ;
F0F4  06           C            DEFB   6
F0F5  FF57         C            DEFW   TIKVEC
F0F7  F480         C            DEFW   DSKTMR          ;POINT 'TIKVEC' TO DISK MOTOR TIMER
F0F9  F444         C            DEFW   STASH           ;POINT 'PINVEC' TO FIFO INPUT ROUTINE
F0FB  F444         C            DEFW   STASH           ;POINT 'SINVEC' TO FIFO INPUT ROUTINE
                   C    ;
                   C    ;       SET FREE MEMORY POINTER
                   C    ;
F0FD  02           C            DEFB   2
F0FE  FF7A         C            DEFW   FREPTR
F100  F7E8         C            DEFW   ROMEND          ;POINT TO FIRST LOCATION AFTER MONITOR
                   C    ;
                   C    ;
F102  FF           C            DEFB   -1              ;END OF VARIABLE INIT TABLE
                   C    ;
                   C    ;
                   C    ;
0000               C    BAUDA   EQU    00H             ;CHANEL A BAUD RATE GENETATOR
0004               C    SIO     EQU    04H             ;DUAL SERIAL I/O
0008               C    GENPIO  EQU    08H             ;GENERAL PURPOSE PARALLEL I/O
000C               C    BAUDB   EQU    0CH             ;CHANEL B BAUD RATE GENERATOR
0010               C    WD1771  EQU    10H             ;WESTERN DIGITAL DISK CONTROLLER
0014               C    SCROLL  EQU    14H             ;CRT SCROLL MEMORY SCROLL REGISTER
0018               C    CTC     EQU    18H             ;QUAD COUNTER/TIMER CIRCUIT
```

```
001C                C     SYSPIO  EQU     1CH             ;SYSTEM PARALLEL I/O
                    C     ;
                    C     ;       INITIALIZE SYSTEM PIO FOR USE AS BANK-SWITCH,
                    C     ;       DISK DRIVE SELECT AND PARALLEL KEYBOARD INPUT
                    C     ;
001C                C     BITDAT  EQU     SYSPIO+0
001D                C     BITCTL  EQU     SYSPIO+1
001E                C     KBDDAT  EQU     SYSPIO+2
001F                C     KBDCTL  EQU     SYSPIO+3
                    C
F103    03 1D       C             DEFB    3,BITCTL
F105    CF          C             DEFB    11001111B       ;PUT SYSTEM PIO IN BIT MODE
F106    18          C             DEFB    00011000B       ;MAKE BIT 4 3 BE INPUT
F107    40          C             DEFB    01000000B       ;DISABLE INTERRUPTS
                    C     ;
F108    01 1C       C             DEFB    1,BITDAT
F10A    00          C             DEFB    00000000B       ;DE-SELECT ROMS, ENABLE DRIVE 0
                    C                                     ;-ENMF=0, -DDEN=0, DSKAC=1,
                    C                                     ;SIDE SEL=1
                    C     ;
F10B    02 1F       C             DEFB    2,KBDCTL
F10D    4F          C             DEFB    01001111B       ;PUT KEYBOARD PORT IN INPUT MODE
F10E    1A          C             DEFB    SYSVEC+2        ;LOAD KEYBOARD INTERRUPT VECTOR
                    C     ;
                    C     ;
                    C     ;       INITIALIZE CHANELS 2 AND 3 OF THE CTC
                    C     ;       TO GENERATE ONE SECOND INTERRUPTS FROM CTC3
                    C     ;
0018                C     CTC0    EQU     CTC+0           ;CTC CHANEL 0 PORT#
0019                C     CTC1    EQU     CTC+1           ;CTC CHANEL 1
001A                C     CTC2    EQU     CTC+2           ;CTC CHANEL 2
001B                C     CTC3    EQU     CTC+3           ;CTC CHANEL 3
                    C
F10F    01 18       C             DEFB    1,CTC0
F111    10          C             DEFB    CTCVEC          ;BASE INTERRUPT VECTOR FOR CTC
                    C     ;
F112    02 1A       C             DEFB    2,CTC2
F114    27          C             DEFB    00100111B       ;PUT CTC2 IN TIMER MODE
F115    69          C             DEFB    105             ;CTC2 PERIOD=105*256*400 NANOSECONDS
                    C     ;
F116    02 1B       C             DEFB    2,CTC3
F118    C7          C             DEFB    11000111B       ;PUT CTC3 IN COUNTER MODE
F119    5D          C             DEFB    93              ;CTC3 PERIOD=999936 MICROSECONDS
                    C     ;
                    C     ;
                    C     ;       INITIALIZE SIO CHANEL B FOR ASYNCHRONOUS SERIAL
                    C     ;       INTERFACE TO PRINTER OR TERMINAL
                    C     ;
0004                C     SIODPA  EQU     SIO+0           ;SIO DATA PORT A
0005                C     SIODPB  EQU     SIO+1           ;SIO DATA PORT B
0006                C     SIOCPA  EQU     SIO+2           ;SIO CONTROL/STATUS PORT A
0007                C     SIOCPB  EQU     SIO+3           ;SIO CONTROL/STATUS PORT B
                    C
F11A    01 0C       C             DEFB    1,BAUDB
F11C    05          C             DEFB    0101B           ;SET COM 8116 FOR 300 BAUD DEFAULT
                    C
```

```
F11D  0B 07       C           DEFB   11,SIOCPB
F11F  04          C           DEFB   4              ;SELECT REGISTER #4
F120  45          C           DEFB   01000101B       ;16X CLOCK, 1 STOP BIT, ODD PARITY
F121  01          C           DEFB   1               ;SELECT REGISTER #1
F122  04          C           DEFB   00000100B       ;STATUS AFFECTS VECTOR, NO INTERRUPTS
F123  03          C           DEFB   3               ;SELECT REGISTER #3
F124  41          C           DEFB   01000001B       ;7 BITS/RX CHARACTER
F125  05          C           DEFB   5               ;SELECT REGISTER #5
F126  AA          C           DEFB   10101010B       ;7 BITS/TX CHARACTER, ASSERT DTR
F127  02          C           DEFB   2               ;SELECT REGISTER #2
F128  00          C           DEFB   SIOVEC          ;LOAD INTERRUPT VECTOR BASE
F129  02          C           DEFB   2               ;SELECT READ REG #2 FOR SIO TEST
                  C
F12A  FF          C           DEFB   -1              ;END-OF-TABLE
                  C     ;
                  C     ;     INIT DONE
                  C     ;
                  C     ;
                  C           INCLUDE MONITOR.MAC
                  C     ;*********************************************************
                  C     ;*                                                       *
                  C     ;*     BASIC HEX MONITOR FOR Z-80 PROCESSORS             *
                  C     ;*                                                       *
                  C     ;*********************************************************
                  C     ;
                  C     ;
                  C     ;
                  C     ;
F12B  CD F3EC     C     PROMPT: CALL   PNEXT
F12E  0D 0A       C             DEFB   CR,LF
F130  2A 20       C             DEFM   '* '
F132  04          C             DEFB   EOT
F133  21 FF88     C             LD     HL,LINBUF
F136  0E 20       C             LD     C,32
F138  CD F33B     C             CALL   GETLIN          ;INPUT A BUFERED CONSOLE LINE
F13B  38 35       C             JR     C,WHAT  ;PRINT 'WHAT ?' IF INPUT ERROR
                  C
F13D  AF          C             XOR    A
F13E  32 FF84     C             LD     (ESCFLG),A
F141  CD F3FC     C             CALL   CRLFS
F144  3A FF88     C             LD     A,(LINBUF)      ;GET FIRST CHARACTER IN LINE
F147  FE 0D       C             CP     CR
F149  28 E0       C             JR     Z,PROMPT        ;JUMP IF A NULL LINE
F14B  21 F182     C             LD     HL,CMDTAB       ;SEARCH FOR A MATCHING CHARACTER
F14E  01 000B     C             LD     BC,CMDSIZ/3     ; IN COMMAND SEARCH TABLE
F151  CD F360     C             CALL   SEARCH
F154  20 1C       C             JR     NZ,WHAT ;TRY AGAIN IF SEACRH FAILS
F156  C5          C             PUSH   BC
F157  FD 21 FF89  C             LD     IY,LINBUF+1
F15B  CD F36A     C             CALL   PARAMS          ;INPUT NUMERIC PARAMETERS FROM
F15E  DD E1       C             POP    IX              ; LINE BUFFER AND TEST IF ERROR
F160  38 10       C             JR     C,WHAT
F162  2A FF7C     C             LD     HL,(PARAM1)
F165  ED 5B FF7E  C             LD     DE,(PARAM2)
F169  ED 4B FF80  C             LD     BC,(PARAM3)
F16D  CD F180     C             CALL   CALLX           ;CALL SUBROUTINE @ IX
```

```
F170  30 B9        C          JR     NC,PROMPT      ;GO BACK TO PROMPT IF NO ERRORS
                   C
F172  CD F3EC      C    WHAT: CALL   PNEXT
F175  20 77 68 61  C          DEFM   ' what ?'
F179  74 20 3F     C
F17C  07           C          DEFB   'G'-64         ;SAY 'what ?' AND BEEP THE BELL
F17D  04           C          DEFB   EOT
F17E  18 AB        C          JR     PROMPT
                   C    ;
                   C    ;
F180  DD E9        C    CALLX: JP     (IX)           ;CALL SUBROUTINE @ IX
                   C    ;
                   C    ;
                   C    ;
F182  52           C    CMDTAB: DEFB  'R'
F183  4F           C          DEFB   'O'
F184  49           C          DEFB   'I'
F185  47           C          DEFB   'G'
F186  54           C          DEFB   'T'
F187  46           C          DEFB   'F'
F188  4D           C          DEFB   'M'
F189  43           C          DEFB   'C'
F18A  42           C          DEFB   'B'
F18B  44           C          DEFB   'D'
F18C  53           C          DEFB   'S'
                   C
F18D  F329         C          DEFW   SWITCH         ;SWITCH CONSOLE OUTPUT VECTOR
F18F  F205         C          DEFW   MEMDMP         ;DUMP MEMORY IN HEX/ASCII
F191  F1A3         C          DEFW   BOOT           ;BOOT UP CP/M
F193  F2E6         C          DEFW   BLOCK          ;MEMORY BLOCK MOVE
F195  F257         C          DEFW   VIEW           ;MEMORY EXAMINE/CHANGE
F197  F2D8         C          DEFW   FILL           ;FILL MEMORY
F199  F28C         C          DEFW   TEST           ;RAM DIAGNOSTIC
F19B  F281         C          DEFW   GOTO           ;JUMP TO MEMORY LOCATION
F19D  F2FE         C          DEFW   INCMD          ;READ FROM INPUT PORT
F19F  F320         C          DEFW   OUTCMD         ;WRITE TO OUTPUT PORT
F1A1  F1BD         C          DEFW   DSKCMD         ;DISPLAY DISK SECTOR DATA
                   C
0021               C    CMDSIZ EQU    $-CMDTAB
                   C    ;
                   C    ;
                   C    ;**********************************************************
                   C    ;*                                                        *
                   C    ;*     MONITOR COMMAND ACTION ROUTINES PACKAGE            *
                   C    ;*                                                        *
                   C    ;**********************************************************
                   C    ;
                   C    ;
                   C    ;
                   C    ;
                   C    ;
                   C    ;      -- DISK BOOT LOADER COMMAND --
                   C    ;
                   C    ; *   BOOT LOADER READS 128 BYTES FROM TRACK 0 / SECTOR 1
                   C    ;*    AND STARTS EXECUTION - LOADING THE CP/M OPERATING SYSTEM
                   C    ;*
```

```
F1A3  0E 00       C     BOOT:  LD    C,0          ;SELECT DRIVE 0 FOR BOOT LOAD
F1A5  CD F6B3     C            CALL  SELECT
F1A8  20 3D       C            JR    NZ,DSKERR
F1AA  CD F6EB     C            CALL  HOME         ;HOME HEAD TO TRACK 0
F1AD  20 38       C            JR    NZ,DSKERR    ;ERROR IF NOT READY OR AT TR0
F1AF  21 0080     C            LD    HL,0080H     ;POINT TO CP/M READ BUFFER
F1B2  0E 01       C            LD    C,1          ;SELECT SECTOR 1
F1B4  CD F72C     C            CALL  READ         ;READ TRACK 0/ SECTOR 1
F1B7  20 2E       C            JR    NZ,DSKERR
F1B9  F1          C            POP   AF           ;CLEAN UP STACK
F1BA  C3 0080     C            JP    0080H        ;GO EXECUTE LOADER
                  C     ;
                  C     ;
                  C     ;      -- DISK SECTOR READ COMMAND --
                  C     ;
F1BD  FE 03       C     DSKCMD: CP   3            ;CHECK PARAMETER COUNT
F1BF  37          C             SCF
F1C0  C0          C             RET   NZ
F1C1  4D          C             LD    C,L          ;USE FIRST ARG AS UNIT#
F1C2  CD F6B3     C             CALL  SELECT
F1C5  20 20       C             JR    NZ,DSKERR
F1C7  21 FF7E     C             LD    HL,PARAM2
F1CA  4E          C             LD    C,(HL)       ;USE SECOND ARG AS TRACK#
F1CB  CD F6FD     C             CALL  SEEK
F1CE  20 17       C             JR    NZ,DSKERR
F1D0  21 FF80     C             LD    HL,PARAM3
F1D3  4E          C             LD    C,(HL)       ;USE THIRD ARG AS SECTOR#
F1D4  21 0080     C             LD    HL,0080H
F1D7  CD F72C     C             CALL  READ
F1DA  CB C7       C             SET   0,A          ;MARK ERROR BYTE AS DUE TO READ
F1DC  20 09       C             JR    NZ,DSKERR
F1DE  21 0080     C             LD    HL,0080H
F1E1  11 0008     C             LD    DE,8
F1E4  C3 F227     C             JP    DUMP         ;DUMP DISK READ BUFFER AND RETURN
                  C     ;
                  C     ;
F1E7  4F          C     DSKERR: LD    C,A          ;SAVE 1771 STATUS
F1E8  CD F3EC     C             CALL  PNEXT
F1EB  64 69 73 6B C             DEFM  'disk error '
F1EF  20 65 72 72 C
F1F3  6F 72 20    C
F1F6  04          C             DEFB  EOT
F1F7  06 08       C             LD    B,8          ;PRINT 1771 ERROR BYTE IN BINARY
F1F9  AF          C     DSKR2:  XOR   A
F1FA  CB 11       C             RL    C
F1FC  CE 30       C             ADC   A,'0'        ;TRANSFORM A INTO ASCII '1' OR '0'
F1FE  CD F415     C             CALL  OUTPUT
F201  10 F6       C             DJNZ  DSKR2        ;REPEAT FOR 8 BITS
F203  B7          C             OR    A
F204  C9          C             RET
                  C     ;
                  C     ;
                  C     ;
                  C     ;      -- MEMORY DUMP COMMAND --
                  C     ;
F205  3D          C     MEMDMP: DEC   A            ;CHECK PARAMETER COUNT
```

```
F206  28 06      C           JR    Z,MDMP2
F208  3D         C           DEC   A
F209  28 08      C           JR    Z,MDMP3
F20B  2A FF86    C   MDMP1: LD    HL,(LAST)
F20E  11 0010    C   MDMP2: LD    DE,16
F211  18 0D      C           JR    MDMP3B
                 C
F213  EB         C   MDMP3: EX    DE,HL
F214  ED 52      C          SBC   HL,DE          ;DERRIVE BYTECOUNT FOR DUMP RANGE
F216  06 04      C           LD    B,4
F218  CB 3C      C   MDMP3A: SRL   H            ;DIVIDE BYTECOUNT BY 16
F21A  CB 1D      C           RR    L
F21C  10 FA      C           DJNZ  MDMP3A
F21E  23         C           INC   HL
F21F  EB         C           EX    DE,HL
F220  CD F227    C   MDMP3B: CALL  DUMP          ;DUMP DE*16 BYTES STRTING AT HL
F223  22 FF86    C           LD    (LAST),HL
F226  C9         C           RET
                 C   ;
                 C   ;
F227  E5         C   DUMP:  PUSH  HL            ;SAVE STARTING ADDRESS
F228  CD F3CD    C          CALL  PUT4HS         ;PRINT STARTING ADDRESS IN HEX
F22B  CD F402    C          CALL  SPACE
F22E  06 10      C          LD    B,16
F230  7E         C   DUMP2: LD    A,(HL)        ;GET A DATA BYTE @ HL
F231  23         C          INC   HL
F232  CD F3D2    C          CALL  PUT2HS         ;PRINT THE DATA IN HEX
F235  10 F9      C          DJNZ  DUMP2          ;REPEAT 16 TIMES
F237  E1         C          POP   HL             ;RESTORE STARTING ADDRESS
F238  06 10      C          LD    B,16
F23A  7E         C   DUMP3: LD    A,(HL)        ;GET BACK DATA BYTE @ HL
F23B  23         C          INC   HL
F23C  CB BF      C          RES   7,A
F23E  FE 20      C          CP    20H
F240  38 04      C          JR    C,DUMP4
F242  FE 7F      C          CP    7FH
F244  38 02      C          JR    C,DUMP5
F246  3E 2E      C   DUMP4: LD    A,'.'         ;PRINT A DOT IF DATA < 20 OR > 7F
F248  CD F415    C   DUMP5: CALL  OUTPUT        ;PRINT ASCII CHARACTER IN A
F24B  10 ED      C          DJNZ  DUMP3
F24D  CD F3FC    C          CALL  CRLFS
F250  C0         C          RET   NZ            ;EXIT IF ESCAPE REQUEST IS INDICATED
F251  1B         C          DEC   DE
F252  7A         C          LD    A,D
F253  B3         C          OR    E
F254  20 D1      C          JR    NZ,DUMP
F256  C9         C          RET
                 C   ;
                 C   ;
                 C   ;
                 C   ;
                 C   ;      -- MEMORY EXAMINE COMMAND --
                 C   ;
F257  CD F2CE    C   VIEW:  CALL  MDATA
F25A  CD F407    C          CALL  ECHO
F25D  FE 0D      C          CP    CR
```

```
F25F  28 1B     C           JR    Z,VIEW4
F261  FE 2D     C           CP    '-'
F263  28 19     C           JR    Z,VIEW5
F265  CD F3BD   C    VIEW2: CALL  ASCHEX
F268  3F        C           CCF
F269  D0        C           RET   NC
F26A  07        C           RLCA
F26B  07        C           RLCA
F26C  07        C           RLCA
F26D  07        C           RLCA
F26E  4F        C           LD    C,A
F26F  CD F407   C           CALL  ECHO
F272  CD F3BD   C           CALL  ASCHEX
F275  3F        C           CCF
F276  D0        C           RET   NC
F277  B1        C           OR    C
F278  77        C    VIEW3: LD    (HL),A
F279  CD F2B9   C           CALL  CHECK
F27C  23        C    VIEW4: INC   HL
F27D  23        C           INC   HL
F27E  2B        C    VIEW5: DEC   HL
F27F  18 D6     C           JR    VIEW
                C    ;
                C    ;
                C    ;
                C    ;      -- JUMP TO MEMORY LOCATION COMMAND --
                C    ;
F281  3D        C    GOTO:  DEC   A             ;CHECK PARAMETER COUNT
F282  37        C           SCF
F283  C0        C           RET   NZ
F284  E5        C           PUSH  HL
F285  DD E1     C           POP   IX
F287  CD F180   C           CALL  CALLX        ;CALL ADDRESS PASSED IN HL
F28A  B7        C           OR    A
F28B  C9        C           RET                ;RETURN IF WE GET BACK AGAIN
                C    ;
                C    ;
                C    ;
                C    ;      -- MEMORY READ/WRITE DIAGNOSTIC COMMAND --
                C    ;
F28C  FE 02     C    TEST:  CP    2             ;CHECK PARAMETER COUNT
F28E  37        C           SCF
F28F  C0        C           RET   NZ
F290  13        C           INC   DE
F291  5A        C           LD    E,D          ;GET ENDING PAGE ADDRESS INTO E
F292  54        C           LD    D,H          ;GET STARTING PAGE ADDRESS INTO D
F293  06 00     C           LD    B,0          ;INITIALIZE PASS COUNTER
F295  62        C    TEST1: LD    H,D          ;POINT HL TO START OF BLOCK
F296  2E 00     C           LD    L,0
F298  7D        C    TEST2: LD    A,L
F299  AC        C           XOR   H            ;GENERATE TEST BYTE
F29A  A8        C           XOR   B
F29B  77        C           LD    (HL),A       ;STORE BYTE IN RAM
F29C  23        C           INC   HL
F29D  7C        C           LD    A,H
F29E  BB        C           CP    E            ;CHECK FOR END OF TEST BLOCK
```

```
F29F  20 F7        C          JR     NZ,TEST2
                   C  ;                             NOW READ BACK EACH BYTE & COMPARE
F2A1  62           C          LD     H,D
F2A2  2E 00        C          LD     L,0            ;POINT HL BACK TO START
F2A4  7D           C  TEST3:  LD     A,L
F2A5  AC           C          XOR    H              ;RE-GENERATE TEST BYTE DATA
F2A6  A8           C          XOR    B
F2A7  CD F2B9      C          CALL   CHECK          ;VERIFY MEMORY DATA STILL GOOD
F2AA  C0           C          RET    NZ             ;EXIT IF ESCAPE REQUEST IS INDICATED
F2AB  23           C          INC    HL             ; ELSE GO ON TO NEXT BYTE
F2AC  7C           C          LD     A,H
F2AD  BB           C          CP     E              ;CHECK FOR END OF BLOCK
F2AE  20 F4        C          JR     NZ,TEST3
F2B0  04           C          INC    B              ;BUMP PASS COUNT
F2B1  3E 2B        C          LD     A,'+'
F2B3  CD F415      C          CALL   OUTPUT         ;PRINT '+' AND ALLOW FOR EXIT
F2B6  28 DD        C          JR     Z,TEST1 ;DO ANOTHER PASS IF NO ESCAPE
F2B8  C9           C          RET
                   C  ;
                   C  ;
                   C  ;
F2B9  BE           C  CHECK:  CP     (HL)
F2BA  C8           C          RET    Z              ;RETURN IF (HL)=A
F2BB  F5           C          PUSH   AF
F2BC  CD F2CE      C          CALL   MDATA          ;PRINT WHAT WAS ACTUALLY READ
F2BF  CD F3EC      C          CALL   PNEXT
F2C2  73 68 6F 75  C          DEFM   'should='
F2C6  6C 64 3D     C
F2C9  04           C          DEFB   EOT
F2CA  F1           C          POP    AF
F2CB  C3 F3D2      C          JP     PUT2HS         ;PRINT WHAT SHOULD HAVE BEEN READ
                   C  ;
                   C  ;
F2CE  CD F3FC      C  MDATA:  CALL   CRLFS
F2D1  CD F3CD      C          CALL   PUT4HS
F2D4  7E           C          LD     A,(HL)
F2D5  C3 F3D2      C          JP     PUT2HS
                   C  ;
                   C  ;
                   C  ;
                   C  ;       -- FILL MEMORY WITH CONSTANT COMMAND --
                   C  ;
F2D8  FE 03        C  FILL:   CP     3              ;CHECK IF PARAMETER COUNT=3
F2DA  37           C          SCF
F2DB  C0           C          RET    NZ
F2DC  71           C  FILL1:  LD     (HL),C
F2DD  E5           C          PUSH   HL
F2DE  B7           C          OR     A
F2DF  ED 52        C          SBC    HL,DE          ;COMPARE HL TO END ADDRESS IN DE
F2E1  E1           C          POP    HL
F2E2  23           C          INC    HL             ;ADVANCE POINTER AFTER COMPARISON
F2E3  38 F7        C          JR     C,FILL1
F2E5  C9           C          RET
                   C  ;
                   C  ;
                   C  ;
```

```
                        C     ;
                        C     ;       -- MEMORY BLOCK MOVE COMMAND --
                        C     ;
F2E6   FE 03            C     BLOCK:  CP      3               ;CHECK IF PARAMETER COUNT=3
F2E8   37               C             SCF
F2E9   C0               C             RET     NZ
F2EA   CD F2F3          C             CALL    BLOCAD
F2ED   79               C             LD      A,C
F2EE   B0               C             OR      B
F2EF   C8               C             RET     Z               ;EXIT NOW IF BC=0
F2F0   ED B0            C             LDIR
F2F2   C9               C             RET
                        C     ;
                        C     ;
                        C     ;
F2F3   EB               C     BLOCAD: EX      DE,HL
F2F4   B7               C             OR      A               ;CLEAR CARRY
F2F5   ED 52            C             SBC     HL,DE           ;GET DIFFRENCE BETWEEN
F2F7   EB               C             EX      DE,HL           ;HL & DE FOR BYTECOUNT
F2F8   D5               C             PUSH    DE
F2F9   C5               C             PUSH    BC
F2FA   D1               C             POP     DE              ;GET OLD BC INTO DE
F2FB   C1               C             POP     BC
F2FC   03               C             INC     BC              ;GET COUNT+1 INTO BC
F2FD   C9               C             RET
                        C     ;
                        C     ;
                        C     ;
                        C     ;
                        C     ;       -- READ FROM INPUT PORT COMMAND --
                        C     ;
F2FE   3D               C     INCMD:  DEC     A               ;CHECK IF PARAMETER COUNT=1
F2FF   37               C             SCF
F300   C0               C             RET     NZ
F301   4D               C             LD      C,L             ;POINT C TO INPUT PORT
F302   CD F3FC          C     IN1:    CALL    CRLFS
F305   79               C             LD      A,C
F306   CD F3D2          C             CALL    PUT2HS
F309   ED 78            C             IN      A,(C)
F30B   CD F3D2          C             CALL    PUT2HS
F30E   CD F407          C             CALL    ECHO
F311   FE 0D            C             CP      CR
F313   28 06            C             JR      Z,IN2
F315   FE 2D            C             CP      '-'
F317   28 04            C             JR      Z,IN3
F319   B7               C             OR      A
F31A   C9               C             RET
                        C
F31B   0C               C     IN2:    INC     C
F31C   0C               C             INC     C
F31D   0D               C     IN3:    DEC     C
F31E   18 E2            C             JR      IN1
                        C     ;
                        C     ;
                        C     ;
                        C     ;       -- WRITE TO OUTPUT PORT COMMAND --
```

```
                        C     ;
F320    FE 02           C     OUTCMD: CP      2               ;CHECK IF PARAMETER COUNT=2
F322    37              C             SCF
F323    C0              C             RET     NZ
F324    4D              C             LD      C,L             ;POINT C TO OUTPUT PORT
F325    ED 59           C             OUT     (C),E           ;OUTPUT DATA PASSED IN E
F327    B7              C             OR      A
F328    C9              C             RET
                        C     ;
                        C     ;
                        C     ;       -- SWITCH CONSOLE OUTPUT DEVICE COMMAND --
                        C     ;
F329    21 FF85         C     SWITCH: LD      HL,COFLAG
F32C    34              C             INC     (HL)            ;TOGGLE CONSOLE OUTPUT TYPE FLAG
F32D    CB 46           C             BIT     0,(HL)
F32F    21 F4FE         C             LD      HL,SIOOUT
F332    28 03           C             JR      Z,SWIT2 ;JUMP IF ZERO TO ONE TRANSITION
F334    21 F520         C             LD      HL,CRTOUT
F337    22 F00D         C     SWIT2:  LD      (CONOUT+1),HL   ;STORE NEW CONSOLE OUTPUT ADDRESS
F33A    C9              C             RET
                        C     ;
                        C     ;
                        C     ;*******************************************************
                        C     ;*                                                     *
                        C     ;*      CONSOLE I/O PACKAGE AND UTILITY ROUTINES       *
                        C     ;*                                                     *
                        C     ;*******************************************************
                        C     ;
                        C     ;
                        C     ;
F33B    41              C     GETLIN: LD      B,C             ;SAVE MAX LINE LENGTH PARAMETER IN B
F33C    CD F407         C     GLIN1:  CALL    ECHO            ;GET A CHARACTER FROM THE CONSOLE
F33F    FE 0D           C             CP      CR              ;CHECK FOR CARRIAGE RETURN
F341    28 0E           C             JR      Z,GLIN2
F343    FE 08           C             CP      'H'-64          ;CHECK FOR CTL-H BACKSPACE
F345    28 0C           C             JR      Z,GLIN4
F347    FE 20           C             CP      ' '
F349    D8              C             RET     C               ;OTHER CONTROL CHARACTERS ARE ILLEGAL
F34A    77              C             LD      (HL),A
F34B    23              C             INC     HL              ;STORE CHARACTER IN BUFFER
F34C    0D              C             DEC     C
F34D    20 ED           C             JR      NZ,GLIN1        ;GET ANOTHER IF THERE'S MORE ROOM
F34F    37              C             SCF
F350    C9              C             RET                     ;RETURN WITH CARRY=1 IF TOO
                        C                                     ;MANY CHARACTERS ARE ENTERED
F351    77              C     GLIN2:  LD      (HL),A          ;PUT CARRIAGE RETURN ON END OF LINE
F352    C9              C             RET                     ;RETURN WITH CARRY BIT=0
                        C
F353    2B              C     GLIN4:  DEC     HL              ;DELETE LAST CHARACTER FROM BUFFER
F354    CD F3EC         C             CALL    PNEXT
F357    20 08           C             DEFB    ' ','H'-64      ;PRINT A SPACE TO OVERWRITE THE
F359    04              C             DEFB    EOT             ; LAST CHARACTER, THEN DO A BACKSPACE
F35A    0C              C             INC     C
F35B    78              C             LD      A,B             ;MAKE SURE YOU'RE NOT TRYING TO
F35C    91              C             SUB     C               ;BACKSPACE PAST THE START OF THE LINE
F35D    30 DD           C             JR      NC,GLIN1
```

```
F35F  C9           C           RET
                   C   ;
                   C   ;
                   C   ;
F360  ED B1        C   SEARCH: CPIR                ;SEARCH TABLE @HL FOR MATCH WITH A
F362  C0           C           RET   NZ            ;EXIT NOW IF SEARCH FAILS
F363  09           C           ADD   HL,BC
F364  09           C           ADD   HL,BC         ;ADD RESIDUE FROM CPIR BYTECOUNT
F365  09           C           ADD   HL,BC         ; TO HL 3 TIMES TO GET POINTER
F366  4E           C           LD    C,(HL)        ; TO ADDRESS PART OF TABLE ENTRY
F367  23           C           INC   HL
F368  46           C           LD    B,(HL)
F369  C9           C           RET                 ;EXIT WITH Z=1 TO INDICATE MATCH
                   C   ;
                   C   ;
                   C   ;
                   C   ;
F36A  01 0000      C   PARAMS: LD    BC,0
F36D  FD 7E 00     C           LD    A,(IY+0)
F370  FE 0D        C           CP    CR            ;CHECK IF LINE TERMINATES
F372  20 08        C           JR    NZ,PARA2      ; IMMEDIATELY WITH A RETURN
F374  AF           C           XOR   A
F375  C9           C           RET                 ;RETURN WITH PARAM COUNT=0 IF SO
                   C
F376  0C           C   PARA1:  INC   C
F377  0C           C           INC   C
F378  CB 59        C           BIT   3,C
F37A  37           C           SCF
F37B  C0           C           RET   NZ            ;ERROR IF > 4 NUMBERS ENTERED
F37C  C5           C   PARA2:  PUSH  BC            ;SAVE PARAMETER COUNT
F37D  CD F39F      C           CALL  GETHEX        ;READ A NUMBER FROM LINE BUFFER
F380  C1           C           POP   BC
F381  D8           C   PARA4:  RET   C             ;ERROR IF RESULT OVER 16 BITS
F382  DD 21 FF7C   C           LD    IX,PARAM1     ;POINT TO PARAMETER STORAGE AREA
F386  DD 09        C           ADD   IX,BC         ;ADD PARAMETER COUNT IN BC
F388  DD 75 00     C           LD    (IX+0),L
F38B  DD 74 01     C           LD    (IX+1),H      ;STORE DATA RETURNED FROM 'GETHEX'
F38E  FE 20        C           CP    ' '
F390  28 E4        C           JR    Z,PARA1 ;GET ANOTHER ITEM IF SPACE
F392  FE 2C        C           CP    ','
F394  28 E0        C           JR    Z,PARA1 ;GET ANOTHER ITEM IF COMMA
F396  FE 0D        C           CP    CR
F398  37           C           SCF                 ;ELSE CHECK FOR CARRIAGE RETURN
F399  C0           C           RET   NZ            ; AND EXIT WITH CY=1 IF NOT
F39A  79           C   PAREND: LD    A,C
F39B  CB 3F        C           SRL   A             ;A=COUNT OF NUMBERS ENTERED
F39D  3C           C           INC   A
F39E  C9           C           RET
                   C   ;
                   C   ;       GETHEX CONVERTS ASCII TO BINARY AND DOES
                   C   ;       HIGH LIMIT CHECKS TO LESS THAN 17 BITS.
                   C   ;       CARRY SET ON ILLEGAL CONVERSION RESULT
                   C   ;       TERMINATING CHARACTER RETURNS IN A.
                   C   ;       HL RETURNS WITH 16 BIT BINARY INTEGER
                   C   ;
F39F  21 0000      C   GETHEX: LD    HL,0
```

```
F3A2  18 0B       C            JR     GNUM3
                  C
F3A4  06 04       C     GNUM1: LD     B,4
F3A6  29          C     GNUM2: ADD    HL,HL          ;MULTIPLY RESULT BY 16
F3A7  D8          C            RET    C              ;RETURN IF IT OVERFLOWS 16 BITS
F3A8  10 FC       C            DJNZ   GNUM2
F3AA  5F          C            LD     E,A            ;APPEND NEW LOW ORDER DIGIT
F3AB  16 00       C            LD     D,0            ;AND GET RESULT BACK INTO DE
F3AD  19          C            ADD    HL,DE
F3AE  D8          C            RET    C              ;RETURN IF OVERFLOW
F3AF  FD 7E 00    C     GNUM3: LD     A,(IY+0)       ;GET A CHARACTER FROM LINE INPUT
F3B2  FD 23       C            INC    IY             ; BUFFER @ IY AND BUMP IY
F3B4  4F          C            LD     C,A
F3B5  CD F3BD     C            CALL   ASCHEX         ;CONVERT ASCII TO NUMERIC
F3B8  30 EA       C            JR     NC,GNUM1
F3BA  79          C            LD     A,C
F3BB  B7          C            OR     A
F3BC  C9          C            RET
                  C     ;
                  C     ;
F3BD  D6 30       C     ASCHEX: SUB    '0'
F3BF  D8          C             RET    C
F3C0  FE 0A       C             CP     10
F3C2  3F          C             CCF
F3C3  D0          C             RET    NC
F3C4  D6 07       C             SUB    7
F3C6  FE 0A       C             CP     10
F3C8  D8          C             RET    C
F3C9  FE 10       C             CP     16
F3CB  3F          C             CCF
F3CC  C9          C             RET
                  C     ;
                  C     ;
                  C     ;
F3CD  7C          C     PUT4HS: LD     A,H
F3CE  CD F3D8     C             CALL   PUT2HX
F3D1  7D          C             LD     A,L
F3D2  CD F3D8     C     PUT2HS: CALL   PUT2HX
F3D5  C3 F402     C             JP     SPACE
                  C     ;
                  C     ;
F3D8  F5          C     PUT2HX: PUSH   AF
F3D9  1F          C             RRA
F3DA  1F          C             RRA
F3DB  1F          C             RRA
F3DC  1F          C             RRA
F3DD  CD F3E1     C             CALL   PUTNIB
F3E0  F1          C             POP    AF
F3E1  E6 0F       C     PUTNIB: AND    00001111B
F3E3  C6 90       C             ADD    A,90H
F3E5  27          C             DAA
F3E6  CE 40       C             ADC    A,40H
F3E8  27          C             DAA
F3E9  C3 F415     C             JP     OUTPUT
                  C     ;
                  C     ;
```

```
                        C     ;       PMSG PRINTS THE STRING OF ASCII CHARACTERS
                        C     ;       POINTED TO BY THE RELATIVE ADDRESS IN DE
                        C     ;       UNTIL AN EOT IS ENCOUNTERED IN THE STRING.
                        C     ;
0004                    C     EOT     EQU     04H
000D                    C     CR      EQU     0DH
000A                    C     LF      EQU     0AH
                        C     ;
                        C
F3EC   E3               C     PNEXT:  EX      (SP),HL
F3ED   CD F3F2          C             CALL    PMSG
F3F0   E3               C             EX      (SP),HL
F3F1   C9               C             RET
                        C     ;
F3F2   7E               C     PMSG:   LD      A,(HL)
F3F3   23               C             INC     HL
F3F4   FE 04            C             CP      EOT
F3F6   C8               C             RET     Z
F3F7   CD F415          C             CALL    OUTPUT
F3FA   18 F6            C             JR      PMSG
                        C     ;
                        C     ;
                        C     ;       CRLFS OUTPUTS A RETURN-LINEFEED-SPACE
                        C     ;       TO THE CONSOLE DEVICE
                        C     ;
F3FC   CD F3EC          C     CRLFS:  CALL    PNEXT
F3FF   0D 0A 04         C             DEFB    CR,LF,EOT
F402   3E 20            C     SPACE:  LD      A,' '
F404   C3 F415          C             JP      OUTPUT
                        C     ;
                        C     ;
                        C     ;
                        C     ;       ECHO INPUTS ONE CHARACTER FROM THE CONSOLE
                        C     ;       DEVICE, PRINTS IT ON THE CONSOLE OUTPUT AND
                        C     ;       THEN RETURNS IT IN REGISTER A WITH BIT 7 RESET
                        C     ;
                        C     ;       OUTPUT PRINTS THE CHARACTER IN REGISTER A ON
                        C     ;       THE CONSOLE OUTPUT DEVICE AND THEN DOES A CHECK
                        C     ;       FOR CONSOLE INPUT TO FREEZE OR ABORT OUTPUT.
                        C     ;
                        C
F407   CD F009          C     ECHO:   CALL    CONIN           ;INPUT A CHARACTER AND ECHO IT
F40A   F5               C             PUSH    AF
F40B   CD F00C          C             CALL    CONOUT
F40E   F1               C             POP     AF
F40F   FE 5B            C             CP      'Z'+1
F411   D8               C             RET     C
F412   D6 20            C             SUB     32              ;CONVERT UPPER CASE TO LOWER CASE
F414   C9               C             RET
                        C     ;
                        C     ;
                        C     ;
F415   CD F00C          C     OUTPUT: CALL    CONOUT
F418   CD F006          C             CALL    CONST           ;SEE IF CONSOLE INPUT IS PENDING
F41B   28 0F            C             JR      Z,OUTP2
F41D   CD F009          C             CALL    CONIN
```

```
F420  FE 0D      C          CP    CR            ;SEE IF CARRIAGE RETURN WAS TYPED
F422  28 05      C          JR    Z,OUTP1
F424  CD F009    C          CALL  CONIN         ;WAIT FOR ANOTHER INPUT CHARACTER
F427  18 03      C          JR    OUTP2         ; THEN RETURN TO CALLING ROUTINE
                 C
F429  32 FF84    C  OUTP1:  LD    (ESCFLG),A    ;SET ESCAPE FLAG TO NON-ZERO VALUE
F42C  3A FF84    C  OUTP2:  LD    A,(ESCFLG)
F42F  B7         C          OR    A             ;RETURN CURRENT STATUS OF ESCAPE
F430  C9         C          RET                 ; FLAG TO CALLING ROUTINE
                 C  ;
                 C  ;
                 C  ;
                 C          INCLUDE INTSRV.MAC
                 C  ;********************************************************
                 C  ;*                                                      *
                 C  ;*     INTERRUPT SERVICE ROUTINES FOR KEYBOARD          *
                 C  ;*     INPUT AND REAL-TIME CLOCK FUNCTIONS              *
                 C  ;*                                                      *
                 C  ;********************************************************
                 C  ;
                 C  ;
                 C  ;
                 C  ;
F431  3A FF30    C  KBDST:  LD    A,(FIFCNT)    ;GET INPUT FIFO BYTECOUNT
F434  B7         C          OR    A             ;TEST IF EQUAL ZERO
F435  C8         C          RET   Z             ;EXIT WITH A=0 IF QUEUE IS EMPTY
F436  3E FF      C          LD    A,255
F438  C9         C          RET                 ;ELSE SET A=255 TO INDICATE DATA READY
                 C  ;
                 C  ;
                 C  ;
F439  CD F431    C  KBDIN:  CALL  KBDST
F43C  28 FB      C          JR    Z,KBDIN ;LOOP UNTIL KEYBOARD INPUT READY
F43E  E5         C          PUSH  HL
F43F  CD F46D    C          CALL  REMOVE        ;GET CHARACTER FROM INPUT QUEUE
F442  E1         C          POP   HL
F443  C9         C          RET
                 C  ;
                 C  ;
                 C  ;
                 C  ;
                 C  ;
F444  21 FF33    C  STASH:  LD    HL,LOCK       ;POINT TO SHIFT LOCK VARIABLES
F447  BE         C          CP    (HL)          ;TEST IF A=SHIFT LOCK CHARACTER
F448  23         C          INC   HL            ; THEN POINT TO LOCK FLAG
F449  20 02      C          JR    NZ,STASH2     ;JUMP IF NOT THE SHIFT CHARACTER
F44B  34         C          INC   (HL)          ; ELSE COMPLIMENT THE SHIFT LOCK
F44C  C9         C          RET                 ; AND EXIT NOW
                 C
F44D  CB 46      C  STASH2: BIT   0,(HL)        ;TEST THE SHIFT LOCK FLAG
F44F  28 0A      C          JR    Z,STASH3      ;JUMP IF SHIFT LOCK NOT SET
F451  FE 40      C          CP    40H           ;ELSE CHECK FOR SHIFABLE CHARACTER
F453  38 06      C          JR    C,STASH3      ; AND JUMP IF NOT EQUAL OR GREATER
F455  FE 7F      C          CP    7FH           ; THAN '@' AND LESS THAN RUBOUT
F457  30 02      C          JR    NC,STASH3
F459  EE 20      C          XOR   00100000B     ;ELSE TOGGLE BIT 5 OF THE CHARACTER
```

```
F45B  4F          C     STASH3: LD    C,A
F45C  21 FF30     C             LD    HL,FIFCNT      ;BUMP INPUT FIFO CHARACTER COUNT
F45F  7E          C             LD    A,(HL)
F460  3C          C             INC   A
F461  FE 10       C             CP    16
F463  D0          C             RET   NC             ;EXIT NOW IF FIFO IS FULL
F464  77          C             LD    (HL),A         ; ELSE INCREMENT FIFO COUNT
F465  21 FF31     C             LD    HL,FIFIN       ;POINT HL TO FIFO INPUT OFFSET
F468  CD F474     C             CALL  INDEX
F46B  71          C             LD    (HL),C         ;STORE CHARACTER IN FIFO @ HL
F46C  C9          C             RET
                  C     ;
                  C     ;
                  C     ;
                  C     ;
F46D  21 FF30     C     REMOVE: LD    HL,FIFCNT
F470  35          C             DEC   (HL)
F471  21 FF32     C             LD    HL,FIFOUT      ;POINT HL TO FIFO OUTPUT OFFSET
F474  7E          C     INDEX:  LD    A,(HL)
F475  3C          C             INC   A
F476  E6 0F       C             AND   00001111B      ;INCREMENT FIFO POINTER
F478  77          C             LD    (HL),A         ; MODULO 16 AND REPLACE
F479  21 FF20     C             LD    HL,FIFO
F47C  85          C             ADD   A,L            ;INDEX INTO FIFO BY OFFSET IN A
F47D  6F          C             LD    L,A
F47E  7E          C             LD    A,(HL)
F47F  C9          C             RET
                  C     ;
                  C     ;
                  C     ;       SOFTWARE DISK MOTOR TURN-OFF TIMER ROUTINE
                  C     ;
F480  21 FF6C     C     DSKTMR: LD    HL,MOTOR       ;DECREMENT DISK TURN-OFF TIMER
F483  35          C             DEC   (HL)
F484  C0          C             RET   NZ             ;EXIT IF NOT TIMED OUT YET
F485  DB 1C       C             IN    A,(BITDAT)
F487  F6 04       C             OR    00000100B      ;DISABLE ALL DRIVE SELECTS AND
F489  D3 1C       C             OUT   (BITDAT),A     ; TURN OFF THE SPINDLE MOTORS
F48B  C9          C             RET
                  C     ;
                  C     ;
                  C     ;       -- INTERRUPT SERVICE ROUTINE FOR PARALLEL KEYBOARD --
                  C     ;
F48C  ED 73 FF35  C     KEYSRV: LD    (SPSAVE),SP    ;SAVE USER STACK POINTER AND
F490  31 FF57     C             LD    SP,TMPSTK+32   ; SWITCH TO LOCAL STACK
F493  E5          C             PUSH  HL
F494  D5          C             PUSH  DE
F495  C5          C             PUSH  BC
F496  F5          C             PUSH  AF             ;SAVE MACHINE STATE
F497  DB 1E       C             IN    A,(KBDDAT)     ;READ KEYBOARD INPUT PORT
F499  2F          C             CPL
F49A  2A FF59     C             LD    HL,(PINVEC)    ;GET KBD INTERRUPT ROUTINE VECTOR
F49D  18 22       C             JR    DSPTCH  ; AND JUMP TO DISPATCH POINT
                  C     ;
                  C     ;
                  C     ;
                  C     ;       -- INTERRUPT SERVICE ROUTINE FOR ONE SECOND TIMER --
```

```
                        C     ;
F49F    ED 73 FF35      C     TIMER:  LD      (SPSAVE),SP     ;SAVE USER STACK POINTER AND
F4A3    31 FF57         C             LD      SP,TMPSTK+32    ; SWITCH TO LOCAL STACK
F4A6    E5              C             PUSH    HL
F4A7    D5              C             PUSH    DE
F4A8    C5              C             PUSH    BC
F4A9    F5              C             PUSH    AF
F4AA    2A FF57         C             LD      HL,(TIKVEC)     ;GET CLOCK INTERRUPT ROUTINE VECTOR
F4AD    18 12           C             JR      DSPTCH  ; AND JUMP TO DISPATCH POINT
                        C     ;
                        C     ;
                        C     ;
                        C     ;       -- SERIAL INPUT INTERRUPT SERVICE ROUTINE FOR SIO --
                        C     ;
F4AF    ED 73 FF35      C     SIOINT: LD      (SPSAVE),SP     ;SAVE USER STACK POINTER AND
F4B3    31 FF57         C             LD      SP,TMPSTK+32    ; SWITCH TO LOCAL STACK
F4B6    E5              C             PUSH    HL
F4B7    D5              C             PUSH    DE
F4B8    C5              C             PUSH    BC
F4B9    F5              C             PUSH    AF              ;SAVE MACHINE STATE
F4BA    DB 05           C             IN      A,(SIODPB)      ;READ SIO DATA INPUT PORT
F4BC    E6 7F           C             AND     01111111B
F4BE    2A FF5B         C             LD      HL,(SINVEC)     ;GET SERIAL INPUT ROUTINE VECTOR
F4C1    CD F4E7         C     DSPTCH: CALL    CALLHL          ;CALL SUBROUTINE ADDRESSED BY H
F4C4    F1              C             POP     AF
F4C5    C1              C             POP     BC
F4C6    D1              C             POP     DE
F4C7    E1              C             POP     HL
F4C8    ED 7B FF35      C             LD      SP,(SPSAVE)
F4CC    FB              C             EI                      ;RE-ENABLE INTERRUPTS AND RETURN
F4CD    ED 4D           C             RETI
                        C     ;
                        C     ;
                        C     ;       -- RX ERROR INTERRUPT SERVICE ROUTINE FOR SIO --
                        C     ;
                        C     ;       ARRIVE HERE IF RECIEVE INTERRUPT FROM FRAMING, OVERRUN
                        C     ;       AND PARITY ERRORS. (PARITY CAN BE DISABLED)
                        C     ;
F4CF    ED 73 FF35      C     SIOERR: LD      (SPSAVE),SP     ;SAVE USER STACK POINTER AND
F4D3    31 FF57         C             LD      SP,TMPSTK+32    ; SWITCH TO LOCAL STACK
F4D6    F5              C             PUSH    AF
F4D7    CD F4F5         C             CALL    SIOIN2          ;CLEAR BAD CHARACTER FROM SIO
F4DA    3E 07           C             LD      A,'G'-64
F4DC    CD F515         C             CALL    SIOXMT          ;OUTPUT A CTL-G AS A WARNING
F4DF    F1              C             POP     AF
F4E0    ED 7B FF35      C             LD      SP,(SPSAVE)
F4E4    FB              C             EI
F4E5    ED 4D           C             RETI
                        C     ;
                        C     ;
F4E7    E9              C     CALLHL: JP      (HL)
                        C     ;
                        C     ;
                        C     ;
                        C     ;       POLLED MODE I/O ROUTINES FOR SIO CHANEL B
                        C     ;
```

```
F4E8  DB 07     C   SIOST:  IN      A,(SIOCPB)      ;GET SIO STATUS REGISTER
F4EA  E6 01     C           AND     00000001B
F4EC  C8        C           RET     Z               ;ACC=0 IF NO DATA AVAILABLE
F4ED  3E FF     C           LD      A,255
F4EF  C9        C           RET
                C   ;
                C   ;
F4F0  CD F4E8   C   SIOIN:  CALL    SIOST           ;TEST CONSOLE STATUS
F4F3  28 FB     C           JR      Z,SIOIN ;LOOP UNTIL DATA IS RECIEVED
F4F5  3E 30     C   SIOIN2: LD      A,00110000B     ;RESET STATUS BITS IN SIO FOR
F4F7  D3 07     C           OUT     (SIOCPB),A      ; PARITY/OVERRUN/FRAMING ERRORS,
F4F9  DB 05     C           IN      A,(SIODPB)      ; THEN GET THE INPUT CHARACTER
F4FB  E6 7F     C           AND     01111111B
F4FD  C9        C           RET
                C   ;
                C   ;
F4FE  FE 20     C   SIOOUT: CP      ' '             ;TEST FOR CONTROL CHARACTERS
F500  30 13     C           JR      NC,SIOXMT       ;JUMP IF PRINTABLE CHARACTER
F502  CD F515   C           CALL    SIOXMT          ; ELSE SEND CONTROL CHARACTER
F505  3A FF79   C           LD      A,(NULLS)       ; AND THEN SEND NULLS AS PADDING
F508  3C        C           INC     A               ;GET NULL PAD COUNT AND FIX SO
F509  18 06     C           JR      PAD1            ; THAT COUNT=0 SENDS NO NULLS
                C
F50B  F5        C   PAD:    PUSH    AF
F50C  AF        C           XOR     A
F50D  CD F515   C           CALL    SIOXMT          ;OUTPUT A NULL TO THE SIO
F510  F1        C           POP     AF
F511  3D        C   PAD1:   DEC     A
F512  20 F7     C           JR      NZ,PAD  ;LOOP SENDING NULLS TO SIO
F514  C9        C           RET
                C   ;
                C   ;
F515  F5        C   SIOXMT: PUSH    AF
F516  DB 07     C   SIOX1:  IN      A,(SIOCPB)
F518  E6 04     C           AND     00000100B       ;TEST TBE STATUS BIT
F51A  28 FA     C           JR      Z,SIOX1
F51C  F1        C           POP     AF
F51D  D3 05     C           OUT     (SIODPB),A      ;OUTPUT DATA TO SIO
F51F  C9        C           RET
                C   ;
                C   ;
                C   ;
                C   ;
                C           INCLUDE CRTOUT.MAC
                C   ;*******************************************************
                C   ;*                                                     *
                C   ;*      MEMORY-MAPPED CRT OUTPUT DRIVER                *
                C   ;*                                                     *
                C   ;*                                                     *
                C   ;*******************************************************
                C   ;
                C   ;
0030            C   CRTBAS  EQU     CRTMEM/256      ;STARTING PAGE# OF 3K CRT SPACE
003C            C   CRTTOP  EQU     (CRTMEM+3072)/256 ;ENDING PAGE# OF CRT SPACE
                C   ;
                C   ;
```

```
F520  E5           C     CRTOUT: PUSH   HL
F521  D5           C             PUSH   DE
F522  C5           C             PUSH   BC
F523  CB BF        C             RES    7,A
F525  4F           C             LD     C,A
F526  F3           C             DI                    ;KEEP THE WOLVES AWAY FOR A WHILE
F527  ED 73 FF35   C             LD     (SPSAVE),SP
F52B  31 FF57      C             LD     SP,TMPSTK+32   ;POINT SP TO TOP OF LOCAL STACK
F52E  DB 1C        C             IN     A,(BITDAT)
F530  CB FF        C             SET    7,A            ;SELECT ROM/CRT MEMORY BANK
F532  D3 1C        C             OUT    (BITDAT),A
                   C     ;
                   C     ;       FIRST REMOVE THE OLD CURSOR CHARACTER FROM THE SCREEN
                   C     ;
F534  21 FF75      C             LD     HL,CHRSAV      ;GET CHARACTER NOW OVERLAYED BY CURSOR
F537  46           C             LD     B,(HL)
F538  2A FF73      C             LD     HL,(CURSOR)    ;LOAD HL WITH CURSOR POINTER
F53B  7C           C             LD     A,H
F53C  E6 0F        C             AND    00001111B      ;A LITTLE INSURANCE THAT HL CAN'T
F53E  F6 30        C             OR     CRTBAS         ; EVER POINT OUTSIDE THE CRT MEMORY
F540  67           C             LD     H,A
F541  70           C             LD     (HL),B         ;REMOVE CURSOR BY RESTORING CHARACTER
                   C     ;
                   C     ;       PROCESS CHARACTER PASSED IN C
                   C     ;
F542  CD F565      C             CALL   OUTCH
                   C     ;
                   C     ;       NOW STORE A NEW CURSOR CHARACTER AT THE CURSOR LOCATION
                   C     ;
F545  7E           C             LD     A,(HL)         ;GET CHARACTER AT NEW CURSOR LOCATION
F546  32 FF75      C             LD     (CHRSAV),A     ;SAVE FOR NEXT TIME 'CRTOUT' IS CALLED
F549  FE 20        C             CP     ' '            ;TEST IF CHARACTER IS A SPACE
F54B  CB FF        C             SET    7,A            ;THEN TURN ON BIT 7 TO ENABLE BLINK
F54D  20 03        C             JR     NZ,CRT2 ;JUMP IF CHARACTER IS NON-BLANK
F54F  3A FF76      C             LD     A,(CSRCHR)     ; ELSE GET CHARACTER USED FOR CURSOR
F552  77           C     CRT2:   LD     (HL),A         ;STORE CHARACTER IN A AS CURSOR MARK
F553  22 FF73      C             LD     (CURSOR),HL    ;SAVE HL AS CURSOR POINTER
                   C
F556  ED 7B FF35   C             LD     SP,(SPSAVE)
F55A  DB 1C        C             IN     A,(BITDAT)
F55C  CB BF        C             RES    7,A            ;SWITCH BACK THE LOWER 16K OF RAM
F55E  D3 1C        C             OUT    (BITDAT),A
F560  FB           C             EI                    ;INTERRUPTS ARE SAFE AGAIN
F561  C1           C             POP    BC
F562  D1           C             POP    DE
F563  E1           C             POP    HL
F564  C9           C             RET
                   C     ;
                   C     ;
                   C     ;
F565  11 FF78      C     OUTCH:  LD     DE,LEADIN
F568  1A           C             LD     A,(DE)         ;GET LEAD-IN SEQUENCE STATE
F569  B7           C             OR     A
F56A  C2 F670      C             JP     NZ,MULTI       ;JUMP IF IN A LEAD-IN SEQUENCE
F56D  79           C             LD     A,C            ; ELSE PROCESS CHARACTER IN C
F56E  FE 20        C             CP     ' '
```

```
F570  38 0F       C            JR     C,CONTRL       ;JUMP IF A CONTROL CHARACTER
F572  71          C    DISPLA: LD     (HL),C         ; ELSE STORE DISPLAYABLE CHARACTER
F573  23          C            INC    HL             ; AND ADVANCE POINTER TO NEXT COLUMN
F574  7D          C            LD     A,L
F575  E6 7F       C            AND    01111111B      ;EXTRACT COLUMN# FROM HL
F577  FE 50       C            CP     80
F579  D8          C            RET    C              ;EXIT IF NOT PAST COLUMN 79
F57A  CD F5E7     C            CALL   RETURN         ; ELSE DO AUTOMATIC CARRIAGE RETURN
F57D  CD F642     C            CALL   LFEED          ; AND LINEFEED
F580  C9          C            RET
                  C    ;
                  C    ;
                  C    ;
F581  E5          C    CONTRL: PUSH   HL
F582  21 F58F     C            LD     HL,CTLTAB      ;SEARCH FOR CONTROL CHARACTER
F585  01 000D     C            LD     BC,CTLSIZ/3    ; HANDLING SUBROUTINE IN TABLE
F588  CD F360     C            CALL   SEARCH
F58B  E1          C            POP    HL
F58C  C0          C            RET    NZ             ;EXIT IF NOT IMPLEMENTED
F58D  C5          C            PUSH   BC
F58E  C9          C            RET                   ;DO SNEAKY JUMP TO PRESERVE REGISTERS
                  C
F58F  1F          C    CTLTAB: DEFB   '_'-64
F590  1E          C            DEFB   '^'-64
F591  1B          C            DEFB   '['-64
F592  1A          C            DEFB   'Z'-64
F593  18          C            DEFB   'X'-64
F594  11          C            DEFB   'Q'-64
F595  0D          C            DEFB   'M'-64
F596  0C          C            DEFB   'L'-64
F597  0B          C            DEFB   'K'-64
F598  0A          C            DEFB   'J'-64
F599  09          C            DEFB   'I'-64
F59A  08          C            DEFB   'H'-64
F59B  07          C            DEFB   'G'-64
                  C
F59C  F5E6        C            DEFW   BELL           ;CTL-G IS THE BELL
F59E  F5BE        C            DEFW   BAKSPC         ;CTL-H IS CURSOR LEFT
F5A0  F5CC        C            DEFW   TAB            ;CTL-I IS TAB
F5A2  F642        C            DEFW   LFEED          ;CTL-J IS CURSOR DOWN
F5A4  F62C        C            DEFW   UPCSR          ;CTL-K IS CURSOR UP
F5A6  F5C4        C            DEFW   FORSPC         ;CTL-L IS CURSOR RIGHT
F5A8  F5E7        C            DEFW   RETURN         ;CTL-M IS CARRIAGE RETURN
F5AA  F611        C            DEFW   CLREOS         ;CTL-Q IS CLEAR TO END-OF-SCREEN
F5AC  F603        C            DEFW   CLREOL         ;CTL-X IS CLEAR TO END-OF-LINE
F5AE  F5EC        C            DEFW   CLRSCN         ;CTL-Z IS CLEAR SCREEN
F5B0  F5B6        C            DEFW   ESCAPE         ;CTL-[ IS ESCAPE
F5B2  F66C        C            DEFW   HOMEUP         ;CTL-^ IS HOME UP
F5B4  F5BA        C            DEFW   STUFF          ;CTL-_ IS DISPLAY CONTROL CHARS
                  C
0027              C    CTLSIZ  EQU    $-CTLTAB
                  C    ;
                  C    ;
F5B6  3E 01       C    ESCAPE: LD     A,1
F5B8  12          C            LD     (DE),A         ;SET LEAD-IN SEQUENCE STATE
F5B9  C9          C            RET                   ; FOR XY CURSOR POSITIONING MODE
```

```
                        C     ;
                        C     ;
F5BA  3E 04             C     STUFF:  LD      A,4
F5BC  12                C             LD      (DE),A          ;SET LEAD-IN SEQUENCE STATE
F5BD  C9                C             RET                     ; FOR CONTROL CHAR OUTPUT MODE
                        C     ;
                        C     ;
F5BE  7D                C     BAKSPC: LD      A,L             ;CHECK FOR LEFT MARGIN
F5BF  E6 7F             C             AND     01111111B
F5C1  C8                C             RET     Z               ;ABORT IF IN LEFTMOST COLUMN
F5C2  2B                C             DEC     HL              ;BACK UP CURSOR POINTER
F5C3  C9                C             RET
                        C     ;
                        C     ;
F5C4  7D                C     FORSPC: LD      A,L             ;CHECK FOR RIGHTMOST COLUNM
F5C5  E6 7F             C             AND     01111111B
F5C7  FE 4F             C             CP      79
F5C9  D0                C             RET     NC              ;DO NOTHING IF ALREADY THERE
F5CA  23                C             INC     HL
F5CB  C9                C             RET                     ;ELSE ADVANCE THE CURSOR POINTER
                        C     ;
                        C     ;
F5CC  11 0008           C     TAB:    LD      DE,8            ;TABS ARE EVERY 8 COLUMNS
F5CF  7D                C             LD      A,L             ;GET COLUMN COMPONENT OF
F5D0  E6 78             C             AND     01111000B       ; PREVIOUS TAB POSITION
F5D2  83                C             ADD     A,E
F5D3  FE 50             C             CP      80              ;EXIT IF NEXT TAB COLUMN WOULD
F5D5  D0                C             RET     NC              ; BE PAST THE RIGHT MARGIN
F5D6  7D                C             LD      A,L
F5D7  E6 F8             C             AND     11111000B       ;ELSE INCREMENT THE CURSOR
F5D9  6F                C             LD      L,A             ; POINTER FOR REAL
F5DA  19                C             ADD     HL,DE
F5DB  C9                C             RET
                        C     ;
                        C     ;
F5DC  DB 1C             C     BELLX:  IN      A,(BITDAT)
F5DE  CB EF             C             SET     5,A             ;TOGGLE BIT 5 OF SYSTEM PIO TO
F5E0  D3 1C             C             OUT     (BITDAT),A      ; TRIGGER BELL HARDWARE TO SOUND
F5E2  CB AF             C             RES     5,A
F5E4  D3 1C             C             OUT     (BITDAT),A
F5E6  C9                C     BELL:   RET
                        C             ;BIT 5 OF PIO IS USED FOR -DDEN, YOU CAN
                        C             ;CHANGE THE PROGRAM IF YOU WANT TO INSTALL
                        C             ;THE HARDWARE BELL (EXCHANGE BELLX: WITH BELL:)
                        C     ;
                        C     ;
                        C     ;
F5E7  7D                C     RETURN: LD      A,L
                        C     ;
F5E8  E6 80             C             AND     10000000B
F5EA  6F                C             LD      L,A             ;MOVE CURSOR POINTER BACK
F5EB  C9                C             RET                     ; TO START OF LINE
                        C     ;
                        C     ;
F5EC  21 3000           C     CLRSCN: LD      HL,CRTMEM
F5EF  E5                C             PUSH    HL
```

```
F5F0  11 3001     C           LD    DE,CRTMEM+1
F5F3  01 0C00     C           LD    BC,24*128
F5F6  36 20       C           LD    (HL),' '
F5F8  ED B0       C           LDIR                ;FILL CRT MEMORY WITH SPACES
F5FA  E1          C           POP   HL            ;POINT TO HOME CURSOR POSITION
F5FB  3E 17       C           LD    A,23
F5FD  32 FF77     C           LD    (BASE),A      ;MAKE BASE LINE# BE 23 AND
F600  D3 14       C           OUT   (SCROLL),A    ; STORE IN SCROLL REGISTER
F602  C9          C           RET
                  C     ;
                  C     ;
F603  E5          C     CLREOL: PUSH  HL          ;SAVE CURSOR POINTER
F604  7D          C           LD    A,L
F605  E6 7F       C           AND   01111111B     ;GET COLUMN# COMPONENT OF
F607  4F          C           LD    C,A           ; CURSOR POINTER INTO C
F608  3E 50       C           LD    A,80          ;CALCULATE HOW MANY CHARACTERS
F60A  91          C           SUB   C             ; REMAIN ON CURRENT LINE
F60B  47          C           LD    B,A
F60C  CD F666     C           CALL  CLR           ;CLEAR REST OF LINE @ HL
F60F  E1          C           POP   HL
F610  C9          C           RET
                  C     ;
                  C     ;
F611  CD F603     C     CLREOS: CALL  CLREOL      ;CLEAR REMAINDER OF CURRENT ROW
F614  E5          C           PUSH  HL
F615  3A FF77     C           LD    A,(BASE)
F618  4F          C           LD    C,A           ;COPY BASE SCREEN ROW# TO C
F619  7D          C     CLRS1:  LD    A,L
F61A  17          C           RLA
F61B  7C          C           LD    A,H
F61C  17          C           RLA                 ;GET ROW# COMPONENT OF HL INTO A
F61D  E6 1F       C           AND   00011111B
F61F  B9          C           CP    C             ;SEE IF HL IS AT BOTTOM ROW OF SCREEN
F620  28 08       C           JR    Z,CLRS2 ; AND LEAVE CLEAR LOOP IF SO
F622  CD F637     C           CALL  DNCSR         ;ELSE POINT HL TO NEXT ROW DOWN
F625  CD F660     C           CALL  CLRLIN        ; AND FILL THAT LINE WITH SPACES
F628  18 EF       C           JR    CLRS1
                  C
F62A  E1          C     CLRS2:  POP   HL          ;RESTORE ORIGINAL CURSOR POINTER
F62B  C9          C           RET
                  C     ;
                  C     ;
F62C  11 FF80     C     UPCSR:  LD    DE,-128     ;SUBTRACT 1 FROM ROW# COMPONENT
F62F  19          C           ADD   HL,DE         ; OF CURSOR POINTER IN HL
F630  7C          C           LD    A,H
F631  FE 30       C           CP    CRTBAS        ;CHECK FOR UNDERFLOW OF POINTER
F633  D0          C           RET   NC
F634  26 3B       C           LD    H,CRTTOP-1    ;WRAP CURSOR AROUND MODULO 3K
F636  C9          C           RET
                  C     ;
                  C     ;
F637  11 0080     C     DNCSR:  LD    DE,128      ;ADD 1 TO ROW# COMPONENT
F63A  19          C           ADD   HL,DE         ; OF CURSOR POINTER IN HL
F63B  7C          C           LD    A,H
F63C  FE 3C       C           CP    CRTTOP        ;CHECK FOR OVERFLOW OF POINTER
F63E  D8          C           RET   C
```

```
F63F  26 30      C            LD     H,CRTBAS       ;RESET POINTER MODULO 128*24
F641  C9         C            RET
                 C    ;
                 C    ;
                 C    ;
F642  7D         C    LFEED:  LD     A,L
F643  17         C            RLA
F644  7C         C            LD     A,H
F645  17         C            RLA                   ;EXTRACT ROW# COMPONENT OF HL
F646  E6 1F      C            AND    00011111B
F648  4F         C            LD     C,A            ;COPY ROW# INTO C FOR SCROLL TEST
F649  CD F637    C            CALL   DNCSR          ;MOVE CURSOR TO NEXT ROW DOWN
F64C  3A FF77    C            LD     A,(BASE)       ;TEST IF CURSOR WAS ON BOTTOM ROW
F64F  B9         C            CP     C              ; OF SCREEN BEFORE MOVING DOWN
F650  C0         C            RET    NZ             ;EXIT IF NOT AT BOTTOM
                 C
F651  E5         C            PUSH   HL             ;ELSE PREP TO SCROLL SCREEN UP
F652  CD F660    C            CALL   CLRLIN         ;FILL NEW BOTTOM LINE WITH SPACES
F655  29         C            ADD    HL,HL
F656  7C         C            LD     A,H            ;GET ROW# COMPONENT OF HL INTO A
F657  E6 1F      C            AND    00011111B
F659  32 FF77    C            LD     (BASE),A       ;STORE NEW BASE LINE#
F65C  D3 14      C            OUT    (SCROLL),A     ;NOW SCROLL UP NEW BLANK BOTTOM LINE
F65E  E1         C            POP    HL
F65F  C9         C            RET
                 C    ;
                 C    ;
F660  7D         C    CLRLIN: LD     A,L
F661  E6 80      C            AND    10000000B      ;POINT HL TO FIRST COLUMN OF ROW
F663  6F         C            LD     L,A
F664  06 50      C            LD     B,80
F666  36 20      C    CLR:    LD     (HL),' '       ;STORE ASCII SPACES AT ADDRESS IN HL
F668  23         C            INC    HL             ; AND INCREMENT HL
F669  10 FB      C            DJNZ   CLR            ;REPEAT NUMBER OF TIMES GIVEN BY B
F66B  C9         C            RET
                 C    ;
                 C    ;
F66C  0E 20      C    HOMEUP: LD     C,' '          ;FAKE-OUT CURSOR ADDRESSING ROUTINE
F66E  18 17      C            JR     SETROW  ; TO DO HOMEUP ALMOST FOR FREE
                 C    ;
                 C    ;
F670  EB         C    MULTI:  EX     DE,HL          ;UNCONDITIONALLY RESET THE LEAD-IN
F671  36 00      C            LD     (HL),0         ; STATE TO ZERO BEFORE GOING ON
F673  EB         C            EX     DE,HL
F674  FE 01      C            CP     1
F676  20 08      C            JR     NZ,M2TST
F678  79         C    SETXY:  LD     A,C            ;GET SECOND CHAR OF SEQUENCE
F679  FE 3D      C            CP     '='
F67B  C0         C            RET    NZ             ;ABORT SEQUENCE IF NOT '='
F67C  3E 02      C            LD     A,2
F67E  12         C            LD     (DE),A         ;MAKE LEADIN=2 NEXT TIME
F67F  C9         C            RET
                 C
F680  FE 02      C    M2TST:  CP     2
F682  20 1B      C            JR     NZ,M3TST
F684  3E 03      C            LD     A,3
```

```
F686  12          C          LD    (DE),A        ;MAKE LEADIN=3 NEXT TIME
F687  3A FF77     C  SETROW: LD    A,(BASE)      ;ARRIVE HERE ON THIRD CHARACTER
F68A  81          C          ADD   A,C           ; OF ESC,'=',ROW,COL SEQUENCE
F68B  D6 1F       C          SUB   ' '-1
F68D  D6 18       C  SETR2:  SUB   24
F68F  30 FC       C          JR    NC,SETR2      ;MAKE SURE ROW# IS BETWEEN 0 AND 23
F691  C6 18       C          ADD   A,24
F693  F6 60       C          OR    CRTMEM/128    ;MERGE IN MSB'S OF CRT MEMORY
F695  F6 60       C          OR    60H
F697  67          C          LD    H,A
F698  2E 00       C          LD    L,0
F69A  CB 3C       C          SRL   H
F69C  CB 1D       C          RR    L
F69E  C9          C          RET
                  C
F69F  FE 03       C  M3TST:  CP    3
F6A1  20 0C       C          JR    NZ,M4TST
F6A3  79          C  SETCOL: LD    A,C           ;ARRIVE HERE ON FOURTH CHARACTER
F6A4  D6 20       C          SUB   ' '           ; OF ESC,'=',ROW,COL SEQUENCE
F6A6  D6 50       C  SETC2:  SUB   80
F6A8  30 FC       C          JR    NC,SETC2      ;MAKE SURE COL# IS BETWEEN 0 AND 79
F6AA  C6 50       C          ADD   A,80
F6AC  B5          C          OR    L             ;MERGE IN COL# WITH L
F6AD  6F          C          LD    L,A
F6AE  C9          C          RET
                  C
F6AF  CD F572     C  M4TST:  CALL  DISPLA        ;DISPLAY THE CONTROL CHARACTER
F6B2  C9          C          RET                 ; PASSED IN C
                  C  ;
                  C  ;
                  C  ;
                  C  ;
                  C          INCLUDE DISKIO.MAC
                  C  ;*******************************************************
                  C  ;*                                                     *
                  C  ;*     DISK INPUT/OUTPUT DRIVER SUBROUTINE PACKAGE     *
                  C  ;*     FOR WESTERN DIGITAL 1771 DISK CONTROLLER        *
                  C  ;*     AND FOR FAMILY 279X AS WELL                     *
                  C  ;*                                                     *
                  C  ;*                                                     *
                  C  ;*******************************************************
                  C  ;
                  C  ;
                  C  ;      EQUATES FOR DISK CONTROLLER PORTS AND COMMAND CODES
                  C  ;
0010              C  STSREG  EQU   WD1771+0      ;STATUS REGISTER
0010              C  CMDREG  EQU   WD1771+0      ;COMMAND REGISTER
0011              C  TRKREG  EQU   WD1771+1      ;TRACK REGISTER
0012              C  SECREG  EQU   WD1771+2      ;SECTOR REGISTER
0013              C  DATREG  EQU   WD1771+3      ;DATA REGISTER
                  C  ;
0088              C  RDCMD   EQU   10001000B     ;READ COMMAND
00A8              C  WRTCMD  EQU   10101000B     ;WRITE COMMAND
001C              C  SKCMD   EQU   00011100B     ;SEEK COMMAND
00D0              C  FINCMD  EQU   11010000B     ;FORCE INTR COMMAND
000C              C  RSTCMD  EQU   00001100B     ;RESTORE COMMAND
```

```
0004                C     HLOAD   EQU     00000100B       ;RD/WRT HEAD LOAD ENABLE
                    C     ;
00C9                C     RET     EQU     0C9H            ;SUBROUTINE RETURN INSTR OPCODE
0066                C     NMIVEC  EQU     0066H           ;THE NON-MASKABLE INTERRUPT IS
                    C                                     ;USED FOR DATA SYNCRONIZATION BETWEEN
                    C                                     ;THE Z-80 AND 1771 DISK CONTROLLER
                    C     ;
                    C     ;
                    C     ;
F6B3  79            C     SELECT: LD      A,C             ;GET UNIT# PASSED IN C AND
F6B4  FE 04         C             CP      4               ; CHECK FOR MAXIMUM VALID#
F6B6  D0            C             RET     NC              ;ERROR IF NUMBER > 3
F6B7  CD F7BA       C             CALL    TURNON          ;MAKE SURE DISKS ARE TURNED ON
F6BA  DB 1C         C             IN      A,(BITDAT)
F6BC  47            C             LD      B,A             ;SAVE CURRENT DRIVE SELECT DATA
F6BD  E6 F8         C             AND     11111000B       ;MERGE IN NEW DRIVE UNIT# IN C
F6BF  B1            C             OR      C               ; IN PLACE OF THE CURRENT ONE
F6C0  D3 1C         C             OUT     (BITDAT),A      ; TO SELECT THE NEW DISK DRIVE
F6C2  CD F7B0       C             CALL    FORCE           ;TEST NEW DRIVE'S READY STATUS
F6C5  28 06         C             JR      Z,SEL2  ; AND CONTINUE IF ITS READY
F6C7  78            C             LD      A,B
F6C8  D3 1C         C             OUT     (BITDAT),A      ;ELSE PUT BACK OLD DRIVE SELECT DATA
F6CA  3E 80         C             LD      A,10000000B     ; AND RETURN DRIVE-NOT-READY STATUS
F6CC  C9            C             RET
                    C
F6CD  21 FF65       C     SEL2:   LD      HL,UNIT         ;POINT HL TO DRIVE SELECT DATA
F6D0  7E            C             LD      A,(HL)          ;LOAD A WITH CURRENT UNIT#
F6D1  71            C             LD      (HL),C          ; AND STORE NEW UNIT# FROM C
F6D2  FE FF         C             CP      255             ;TEST IF NO DRIVE HAS BEEN SELECTED
F6D4  28 06         C             JR      Z,SEL3  ; YET AND SKIP NEXT SEGMENT IF SO
F6D6  23            C             INC     HL              ;POINT TO HEAD POSITION TABLE
F6D7  85            C             ADD     A,L             ; AND ADD IN NEW UNIT# AS INDEX
F6D8  6F            C             LD      L,A
F6D9  DB 11         C             IN      A,(TRKREG)      ;GET CURRENT HEAD POSITION
F6DB  77            C             LD      (HL),A          ; AND STORE IN TABLE @ HL
F6DC  21 FF66       C     SEL3:   LD      HL,TRKTAB
F6DF  7D            C             LD      A,L
F6E0  81            C             ADD     A,C             ;INDEX INTO TABLE TO GET
F6E1  6F            C             LD      L,A             ; HEAD POSITION OF NEW DRIVE
F6E2  7E            C             LD      A,(HL)
F6E3  FE FF         C             CP      255             ;TEST IF NEW DRIVE HAS EVER BEEN
F6E5  28 04         C             JR      Z,HOME  ; SELECTED AND DO A HOME IF NOT
F6E7  D3 11         C             OUT     (TRKREG),A      ;OUTPUT THE DRIVE'S CURRENT HEAD
F6E9  AF            C             XOR     A               ; POSITION TO THE TRACK REGISTER
F6EA  C9            C             RET
                    C     ;
                    C     ;
                    C     ;
F6EB  CD F7AD       C     HOME:   CALL    READY           ;CLEAR DISK CONTROLLER
F6EE  C0            C             RET     NZ              ;EXIT IF DRIVE NOT READY
F6EF  AF            C             XOR     A
F6F0  32 FF6D       C             LD      (TRACK),A       ;SET TRACK# IN MEM TO ZERO
F6F3  06 0C         C     RESTOR: LD      B,RSTCMD        ;LOAD B WITH A RESTORE COMMAND
F6F5  CD F795       C             CALL    STEP            ;EXECUTE HEAD MOVING OPERATION
F6F8  EE 04         C             XOR     00000100B       ;GET TRUE TRACK 0 STATUS
F6FA  E6 9C         C             AND     10011100B       ;MASK TO ERROR BITS
```

```
F6FC  C9          C            RET              ;RETURN 1771 STATUS IN A
                  C    ;
                  C    ;
                  C    ;
F6FD  CD F7AD     C    SEEK:   CALL   READY      ;CLEAR DISK CONTROLLER
F700  C0          C            RET    NZ        ;EXIT IF DRIVE NOT READY
F701  79          C            LD     A,C       ;GET TRACK# DATA FROM C AND
F702  FE 4D       C            CP     77        ; CHECK FOR MAXIMUM VALID#
F704  D0          C            RET    NC        ;FORGET IT IF TRACK# > 76
F705  32 FF6D     C            LD     (TRACK),A ; ELSE STORE TRACK# FOR SEEK
F708  D3 13       C            OUT    (DATREG),A ;OUTPUT TRACK # TO 1771
F70A  06 1C       C            LD     B,SKCMD   ;LOAD B WITH A SEEK COMMAND AND
F70C  CD F795     C            CALL   STEP      ; GO SEEK WITH PROPER STEP RATE
F70F  E6 98       C            AND    10011000B ;MASK TO READY,SEEK AND CRC ERROR
F711  C8          C            RET    Z         ; BITS AND RETURN IF ALL GOOD
                  C
F712  CD F6F3     C            CALL   RESTOR    ;ELSE TRY TO RE-CAILBRATE HEAD
F715  C0          C            RET    NZ        ;ERROR IF WE CAN'T FIND TRACK 0
F716  79          C            LD     A,C
F717  D3 13       C            OUT    (DATREG),A ;OUTPUT TRACK# TO 1771
F719  06 1C       C            LD     B,SKCMD
F71B  CD F795     C            CALL   STEP      ;TRY TO SEEK THE TRACK AGAIN
F71E  E6 98       C            AND    10011000B
F720  C9          C            RET              ;RETURN FINAL SEEK STATUS IN A
                  C    ;
                  C    ;
                  C    ;
F721  CD F7AD     C    WRITE:  CALL   READY      ;CLEAR THE DISK CONTROLLER
F724  C0          C            RET    NZ        ;EXIT IF DRIVE NOT READY
F725  CB 77       C            BIT    6,A
F727  C0          C            RET    NZ        ;EXIT IF DISK IS WRITE-PROTECTED
F728  06 A8       C            LD     B,WRTCMD
F72A  18 06       C            JR     RDWRT
                  C
F72C  CD F7AD     C    READ:   CALL   READY      ;CLEAR DISK CONTROLLER
F72F  C0          C            RET    NZ        ;EXIT IF DRIVE NOT READY
F730  06 88       C            LD     B,RDCMD
F732  22 FF71     C    RDWRT:  LD     (IOPTR),HL ;STORE DISK I/O DATA POINTER
F735  21 FF6E     C            LD     HL,SECTOR
F738  71          C            LD     (HL),C    ;STORE SECTOR# FOR READ/WRITE
F739  23          C            INC    HL
F73A  70          C            LD     (HL),B    ;SAVE READ/WRITE COMMAND BYTE
F73B  23          C            INC    HL
F73C  36 02       C            LD     (HL),2    ;SET DISK OPERATION RE-TRY COUNT
F73E  F3          C    RW1:    DI               ;NO INTERRUPTS DURING DISK I/O
F73F  21 0066     C            LD     HL,NMIVEC ;SAVE BYTE AT NMI VECTOR LOCATION
F742  56          C            LD     D,(HL)    ; IN D FOR DURATION OF READ/WRITE
F743  36 C9       C            LD     (HL),RET  ; LOOP AND REPLACE IT WITH A RET
F745  21 FF6B     C            LD     HL,RECLEN
F748  46          C            LD     B,(HL)    ;B=NUMBER OF BYTES/SECTOR
F749  0E 13       C            LD     C,DATREG  ;C=1771 DATA REGISTER PORT#
F74B  2A FF71     C            LD     HL,(IOPTR) ;HL=DISK READ/WRITE DATA POINTER
F74E  3A FF6E     C            LD     A,(SECTOR) ;GET SECTOR NUMBER
F751  D3 12       C            OUT    (SECREG),A ;OUTPUT SECTOR# TO 1771
F753  CD F7B0     C            CALL   FORCE     ;ISSUE A FORCE INTERRUPT COMMAND
F756  CB 6F       C            BIT    5,A       ; TO TEST CURRENT HEAD LOAD STATUS
```

```
F758  3A FF6F     C          LD    A,(CMDTYP)     ;GET READ OR WRITE COMMAND BYTE
F75B  20 02       C          JR    NZ,RW2  ;JUMP IF HEAD IS ALREADY LOADED
F75D  F6 04       C          OR    HLOAD          ; ELSE MERGE IN HLD BIT
F75F  CD F7A5     C    RW2:  CALL  CMDOUT         ;START THE 1771 DOING IT'S THING
F762  CB 6F       C          BIT   5,A            ;TEST IF COMMAND IS A READ OR WRITE
F764  20 0D       C          JR    NZ,WLOOP       ; AND JUMP TO THE CORRECT LOOP
F766  76          C    RLOOP: HALT
F767  ED A2       C          INI
F769  C2 F766     C          JP    NZ,RLOOP
F76C  CD F79E     C          CALL  BUSY           ;LOOP UNTIL 1771 COMES UN-BUSY
F76F  E6 9C       C          AND   10011100B      ;MASK OFF TO READY, NOT FOUND, CRC
F771  18 0B       C          JR    RW3            ; AND LOST DATA STATUS BITS
                  C
F773  76          C    WLOOP: HALT
F774  ED A3       C          OUTI
F776  C2 F773     C          JP    NZ,WLOOP
F779  CD F79E     C          CALL  BUSY
F77C  E6 BC       C          AND   10111100B      ;MASK OFF AS ABOVE PLUS WRITE FAULT
F77E  21 0066     C    RW3:  LD    HL,NMIVEC
F781  72          C          LD    (HL),D         ;RESTORE BYTE @ NMI VECTOR
F782  FB          C          EI
F783  C8          C          RET   Z              ;RETURN IF NO DISK I/O ERRORS
F784  21 FF70     C          LD    HL,RETRY
F787  35          C          DEC   (HL)           ;DECREMENT RE-TRY COUNT AND
F788  20 02       C          JR    NZ,RW4  ; EXECUTE COMAND AGAIN IF NOT=0
F78A  B7          C          OR    A
F78B  C9          C          RET                  ;ELSE RETURN 1771 ERROR STATUS
                  C
F78C  21 FF6D     C    RW4:  LD    HL,TRACK
F78F  4E          C          LD    C,(HL)         ;GET TRACK# FOR CURRENT OPERATION
F790  CD F6FD     C          CALL  SEEK           ;TRY TO RE-CAILBRATE THE HEAD
F793  18 A9       C          JR    RW1            ; BEFORE READING OR WRITING AGAIN
                  C    ;
                  C    ;
                  C    ;
F795  3A FF6A     C    STEP: LD    A,(SPEED)      ;GET STEP SPEED VARIABLE
F798  E6 03       C          AND   00000011B
F79A  B0          C          OR    B              ;MERGE WITH SEEK/HOME COMMAND IN B
F79B  CD F7A5     C          CALL  CMDOUT         ;OUTPUT COMMAND AND DELAY
F79E  DB 10       C    BUSY: IN    A,(STSREG)
F7A0  CB 47       C          BIT   0,A            ;TEST BUSY BIT FROM
F7A2  20 FA       C          JR    NZ,BUSY ; 1771 AND LOOP TILL=0
F7A4  C9          C          RET
                  C    ;
                  C    ;
                  C    ;
F7A5  D3 10       C    CMDOUT: OUT   (CMDREG),A     ;OUTPUT A COMMAND TO THE 1771
F7A7  CD F7AA     C          CALL  PAUSE          ;WASTE 44 MICROSECONDS
F7AA  E3          C    PAUSE: EX    (SP),HL
F7AB  E3          C          EX    (SP),HL
F7AC  C9          C          RET
                  C    ;
                  C    ;
                  C    ;
F7AD  CD F7BA     C    READY: CALL  TURNON         ;KEEP THOSE DISKS SPINING FOLKS
F7B0  3E D0       C    FORCE: LD    A,FINCMD       ;ISSUE A FORCE INTERRUPT COMMAND
```

```
F7B2  CD F7A5     C          CALL   CMDOUT
F7B5  DB 10       C          IN     A,(STSREG)      ;READ STATUS REGISTER CONTENTS
F7B7  CB 7F       C          BIT    7,A             ;TEST DRIVE NOT READY BIT
F7B9  C9          C          RET
                  C     ;
                  C     ;
                  C     ;
F7BA  3E 1E       C     TURNON: LD    A,30
F7BC  32 FF6C     C          LD     (MOTOR),A       ;RE-LOAD THE MOTOR TURN-OFF TIMER
F7BF  CD F7AA     C          CALL   PAUSE
F7C2  DB 1C       C          IN     A,(BITDAT)
F7C4  CB 57       C          BIT    2,A             ;TEST IF MOTORS HAVE BEEN STOPPED
F7C6  C8          C          RET    Z               ; AND EXIT IF STILL TURNED ON
F7C7  E6 FB       C          AND    11111011B       ;ELSE RE-ENABLE THE DRIVE SELECTS
F7C9  D3 1C       C          OUT    (BITDAT),A      ; AND ACTIVATE THE MOTOR RELAY
F7CB  C5          C          PUSH   BC
F7CC  06 00       C          LD     B,0             ;SET READY LOOP MAX TIMEOUT
F7CE  CD F7DE     C     TURN2: CALL   WAIT            ;WAIT 1/93 SECOND AND TEST READY
F7D1  28 02       C          JR     Z,TURN3 ;EXIT LOOP IF DRIVE READY
F7D3  10 F9       C          DJNZ   TURN2           ; ELSE TRY AGAIN UP TO 256 TIMES
F7D5  06 09       C     TURN3: LD     B,9
F7D7  CD F7DE     C     TURN4: CALL   WAIT            ;GIVE ABOUT 1/10 SECOND MORE DELAY
F7DA  10 FB       C          DJNZ   TURN4
F7DC  C1          C          POP    BC
F7DD  C9          C          RET
                  C     ;
                  C     ;
F7DE  DB 1B       C     WAIT:  IN     A,(CTC3)        ;GET CURRENT CTC3 COUNT VALUE
F7E0  4F          C          LD     C,A
F7E1  DB 1B       C     WAIT2: IN     A,(CTC3)
F7E3  B9          C          CP     C               ;TEST IF CTC3 HAS CHANGED BY 1 COUNT
F7E4  28 FB       C          JR     Z,WAIT2 ; AND LOOP UNTIL IT CHANGES
F7E6  18 C8       C          JR     FORCE           ; THEN GO TEST DRIVE READY STATUS
                  C     ;
                  C     ;
                  C     ;
                  C     ;
                        ;
F7E8  0000              ROMEND: DEFW  0               ;TAIL OF FREE MEMORY LINKED LIST
                        ;
                              ORG   RAM
                  C           INCLUDE MEMORY.MAC
                  C     ;*******************************************************
                  C     ;*                                                     *
                  C     ;*     STORAGE ALLOCATION FOR 256 BYTE SCRATCH RAM     *
                  C     ;*                                                     *
                  C     ;*******************************************************
                  C     ;
                  C     ;
                  C
FF00              C     VECTAB EQU    $               ;INTERRUPT VECTOR TABLE STARTS HERE
FF00              C     SIOVEC: DEFS  16              ;SPACE FOR 8 VECTORS FOR SIO
FF10              C     CTCVEC: DEFS  8               ;SPACE FOR 4 VECTORS FOR CTC
FF18              C     SYSVEC: DEFS  4               ;SPACE FOR 2 VECTORS FOR SYSTEM PIO
FF1C              C     GENVEC: DEFS  4               ;SPACE FOR 2 VECTORS FOR GENERAL PIO
                  C     ;
```

```
                        C     ;
                        C     ;     KEYBOARD DATA INPUT FIFO VARIABLES
                        C
FF20                    C     FIFO:   DEFS   16          ;CONSOLE INPUT FIFO
FF30                    C     FIFCNT: DEFS   1           ;FIFO DATA COUNTER
FF31                    C     FIFIN:  DEFS   1           ;FIFI INPUT POINTER
FF32                    C     FIFOUT: DEFS   1           ;FIFO OUTPUT POINTER
FF33                    C     LOCK:   DEFS   2           ;SHIFT LOCK CHARACTER+FLAG BYTE
                        C     ;
                        C     ;
                        C     ;     STACK POINTER SAVE AND LOCAL STACK FOR INTERRUPT ROUTINES
                        C
FF35                    C     SPSAVE: DEFS   2           ;USER STACK POINTER SAVE AREA
FF37                    C     TMPSTK: DEFS   32          ;LOCAL STACK FOR INTERRUPTS
                        C     ;
                        C     ;
                        C     ;     'SOFTWARE' VECTORS FOR INTERRUPT SERVICE ROUTINES
                        C
FF57                    C     TIKVEC: DEFS   2           ;ONE SECOND INTERRUPT ROUTINE VECTOR
FF59                    C     PINVEC: DEFS   2           ;PARALLEL CONSOLE INPUT VECTOR
FF5B                    C     SINVEC: DEFS   2           ;SERIAL CONSOLE INPUT VECTOR
                        C     ;
                        C     ;
                        C     ;     CLOCK-TIMER INTERRUPT VARIABLES
                        C
FF5D                    C     TIKCNT: DEFS   2           ;BINARY CLOCK TICK COUNTER
FF5F                    C     DAY:    DEFS   1           ;CALENDAR DAY
FF60                    C     MONTH:  DEFS   1           ;       MONTH
FF61                    C     YEAR:   DEFS   1           ;       YEAR
FF62                    C     HRS:    DEFS   1           ;CLOCK HOURS REGISTER
FF63                    C     MINS:   DEFS   1           ;     MINUTES RETISTER
FF64                    C     SECS:   DEFS   1           ;     SECONDS REGISTER
                        C     ;
                        C     ;
                        C     ;     DISK I/O DRIVER VARIABLES
                        C
FF65                    C     UNIT:   DEFS   1           ;CURRENTLY SELECTED DISK#
FF66                    C     TRKTAB: DEFS   4           ;4 DRIVE HEAD POSITION TABLE
FF6A                    C     SPEED:  DEFS   1           ;SEEK SPEED FOR 1771 COMMANDS
FF6B                    C     RECLEN: DEFS   1           ;SECTOR RECORD LENGTH VARIABLE
FF6C                    C     MOTOR:  DEFS   1           ;DRIVE MOTOR TURN-OFF TIMER
FF6D                    C     TRACK:  DEFS   1
FF6E                    C     SECTOR: DEFS   1
FF6F                    C     CMDTYP: DEFS   1           ;COMMAND BYTE FOR READS/WRITES
FF70                    C     RETRY:  DEFS   1           ;DISK OPERATION RE-TRY COUNT
FF71                    C     IOPTR:  DEFS   2           ;DISK I/O BUFFER POINTER
                        C     ;
                        C     ;
                        C     ;     CRT OUTPUT DRIVER VARIABLES
                        C
FF73                    C     CURSOR: DEFS   2           ;CURSOR POINTER
FF75                    C     CHRSAV: DEFS   1           ;CHARACTER OVERLAYED BY CURSOR
FF76                    C     CSRCHR: DEFS   1           ;CHARACTER USED FOR A CURSOR
FF77                    C     BASE:   DEFS   1           ;CURRENT CONTENTS OF SCROLL REGISTER
FF78                    C     LEADIN: DEFS   1           ;STATE OF LEAD-IN SEQUENCE HANDLER
```

```
                        C       ;
                        C       ;
                        C       ;       NULL PAD COUNT FOR SERIAL OUTPUT DELAY
                        C
FF79                    C       NULLS:  DEFS    1               ;# OF NULLS SENT AFTER CONTROL CHARS.
                        C       ;
                        C       ;
                        C       ;       LISTHEAD POINTER FOR DYNAMIC MEMORY ALLOCATION SCHEME
                        C
FF7A                    C       FREPTR: DEFS    2
                        C       ;
                        C       ;
                        C       ;       CONSOLE MONITOR PROGRAM VARIABLES
                        C
FF7C                    C       PARAM1: DEFS    2               ;STORAGE FOR NUMBERS READ
FF7E                    C       PARAM2: DEFS    2               ; FROM LINE INPUT BUFFER
FF80                    C       PARAM3: DEFS    2               ; BY 'PARAMS' SUBROUTINE
FF82                    C       PARAM4: DEFS    2
FF84                    C       ESCFLG: DEFS    1               ;CONSOLE ESCAPE FLAG
FF85                    C       COFLAG: DEFS    1               ;CONSOLE OUTPUT TOGGLE
FF86                    C       LAST:   DEFS    2               ;LAST ADDRESS USED BY 'MEMDMP'
FF88                    C       LINBUF: DEFS    64              ;CONSOLE LINE INPUT BUFFER
                        C       ;
                        C       ;

                                ;
                                END
```

Macros:

Symbols:

ASCHEX	F3BD	BAKSPC	F5BE	BASE	FF77	BAUD	F078
BAUD1	F079	BAUD2	F081	BAUD3	F08D	BAUDA	0000
BAUDB	000C	BELL	F5E6	BELLX	F5DC	BITCTL	001D
BITDAT	001C	BLOCAD	F2F3	BLOCK	F2E6	BOOT	F1A3
BUSY	F79E	CALLHL	F4E7	CALLX	F180	CHECK	F2B9
CHRSAV	FF75	CLR	F666	CLREOL	F603	CLREOS	F611
CLRLIN	F660	CLRS1	F619	CLRS2	F62A	CLRSCN	F5EC
CMDOUT	F7A5	CMDREG	0010	CMDSIZ	0021	CMDTAB	F182
CMDTYP	FF6F	COFLAG	FF85	COLD	F000	CONIN	F009
CONOUT	F00C	CONST	F006	CONTRL	F581	CR	000D
CRLFS	F3FC	CRT2	F552	CRTBAS	0030	CRTMEM	3000
CRTOUT	F520	CRTTOP	003C	CSRCHR	FF76	CTC	0018
CTC0	0018	CTC1	0019	CTC2	001A	CTC3	001B
CTCVEC	FF10	CTLSIZ	0027	CTLTAB	F58F	CURSOR	FF73
DATREG	0013	DAY	FF5F	DECIDE	F062	DISPLA	F572
DNCSR	F637	DSKCMD	F1BD	DSKERR	F1E7	DSKR2	F1F9
DSKTMR	F480	DSPTCH	F4C1	DUMP	F227	DUMP2	F230
DUMP3	F23A	DUMP4	F246	DUMP5	F248	ECHO	F407
EOT	0004	ESCAPE	F5B6	ESCFLG	FF84	FIFCNT	FF30
FIFIN	FF31	FIFO	FF20	FIFOUT	FF32	FILL	F2D8
FILL1	F2DC	FINCMD	00D0	FORCE	F7B0	FORSPC	F5C4
FREPTR	FF7A	GENPIO	0008	GENVEC	FF1C	GETHEX	F39F
GETLIN	F33B	GLIN1	F33C	GLIN2	F351	GLIN4	F353
GNUM1	F3A4	GNUM2	F3A6	GNUM3	F3AF	GOTO	F281
HLOAD	0004	HOME	F6EB	HOMEUP	F66C	HRS	FF62
IN1	F302	IN2	F31B	IN3	F31D	INCMD	F2FE
INDEX	F474	INIT	F02A	INIT1	F02E	INIT2	F03F
INIT3	F04E	INTAB	F0D3	IOPTR	FF71	KBDCTL	001F
KBDDAT	001E	KBDIN	F439	KBDST	F431	KEYSRV	F48C
LAST	FF86	LEADIN	FF78	LF	000A	LFEED	F642
LINBUF	FF88	LOCK	FF33	M2TST	F680	M3TST	F69F
M4TST	F6AF	MDATA	F2CE	MDMP1	F20B	MDMP2	F20E
MDMP3	F213	MDMP3A	F218	MDMP3B	F220	MEMDMP	F205
MINS	FF63	MONTH	FF60	MOTOR	FF6C	MULTI	F670
NMIVEC	0066	NULLS	FF79	OUTCH	F565	OUTCMD	F320
OUTP1	F429	OUTP2	F42C	OUTPUT	F415	PAD	F50B
PAD1	F511	PARA1	F376	PARA2	F37C	PARA4	F381
PARALL	F070	PARAM1	FF7C	PARAM2	FF7E	PARAM3	FF80
PARAM4	FF82	PARAMS	F36A	PAREND	F39A	PAUSE	F7AA
PINVEC	FF59	PMSG	F3F2	PNEXT	F3EC	PROMPT	F12B
PUT2HS	F3D2	PUT2HX	F3D8	PUT4HS	F3CD	PUTNIB	F3E1
RAM	FF00	RATES	F0CB	RDCMD	0088	RDWRT	F732
READ	F72C	READY	F7AD	RECLEN	FF6B	REMOVE	F46D
RESTOR	F6F3	RET	00C9	RETRY	FF70	RETURN	F5E7
RLOOP	F766	ROM	F000	ROMEND	F7E8	RSTCMD	000C
RW1	F73E	RW2	F75F	RW3	F77E	RW4	F78C
SCROLL	0014	SEARCH	F360	SECREG	0012	SECS	FF64
SECTOR	FF6E	SEEK	F6FD	SEL2	F6CD	SEL3	F6DC
SELECT	F6B3	SETC2	F6A6	SETCOL	F6A3	SETR2	F68D
SETROW	F687	SETXY	F678	SIGNON	F0A5	SINVEC	FF5B
SIO	0004	SIOCPA	0006	SIOCPB	0007	SIODPA	0004
SIODPB	0005	SIOERR	F4CF	SIOIN	F4F0	SIOIN2	F4F5
SIOINT	F4AF	SIOOUT	F4FE	SIOST	F4E8	SIOVEC	FF00
SIOX1	F516	SIOXMT	F515	SKCMD	001C	SPACE	F402

SPEED	FF6A	SPSAVE	FF35	STASH	F444	STASH2	F44D
STASH3	F45B	STEP	F795	STSREG	0010	STUFF	F5BA
SWIT2	F337	SWITCH	F329	SYSPIO	001C	SYSVEC	FF18
TAB	F5CC	TEST	F28C	TEST1	F295	TEST2	F298
TEST3	F2A4	TIKCNT	FF5D	TIKVEC	FF57	TIMER	F49F
TMPSTK	FF37	TRACK	FF6D	TRKREG	0011	TRKTAB	FF66
TURN2	F7CE	TURN3	F7D5	TURN4	F7D7	TURNON	F7BA
UNIT	FF65	UPCSR	F62C	VECTAB	FF00	VIEW	F257
VIEW2	F265	VIEW3	F278	VIEW4	F27C	VIEW5	F27E
WAIT	F7DE	WAIT2	F7E1	WARM	F003	WD1771	0010
WHAT	F172	WLOOP	F773	WRITE	F721	WRTCMD	00A8
YEAR	FF61						

No Fatal error(s)

o o
o o
o o
o

Institut Jožef Stefan
Odsek za računalništvo in informatiko
Jamova 39, 61000 Ljubljana
tel. (061) 263-261/528, tlx. 31296

Navodila za risanje predlog (layout) vezij, da so primerne za digitalizacijo.

1. Predloga naj bo narisana v merilu 2:1 ali večjem in po možnosti na "inčni" mreži (enota = 2.54mm).
   če boste mrežo oštevilčili, mora biti koordinatno izhodišče (0/0) v levem spodnjem vogalu, x-os v horizontalni smeri in y-os v vertikalni.

2. Pri dvostranskih vezjih smeta biti obe strani narisani z različnima barvama na enem listu, sicer naj bo označeno, kje je stran elementov in kje stran spajkanja. Praviloma je stran elementov narisana z modro ali črno barvo, stran spajkanja pa je rdeča.

3. Na predlogi se smejo pojavljati očesca, narisana kot krogi, velika sorazmerno z merilom, v katerem je narisana predloga (primer: standardna očesca s premerom 1.5 mm naj bodo v merilu 2:1 široka 3 mm.

4. Poleg očesc je mogoče narisati lomljene črte različnih debelin, poligone (poljubne mnogokotnike) in tekste, ki jih sestavljajo poljubno velike in poljubno debele velike tiskane črke, številke in standardni posebni znaki. Po želji je lahko nožica 1 integriranega vezja označena s kvadratkom.

5. Na predlogi naj bodo podane dimenzije posameznih izvrtin v mm.

6. Vezi na običajno gostih vezjih se rišejo na mrežo in se na mreži črte tudi lomijo. Pri večjih gostotah to ne zadošča, 0.3 mm debele povezave je mogoče spraviti na t.i. dvojno gostoto, tako da med dvema črtama na mreži potegnemo še eno vez. Pri tem je treba paziti na zadostno oddaljenost od očesc, ki so v bližini. Vse kočljive situacije je možno preračunati, upoštevati je treba le podatek, da je najmanjša debelina vezi 0.25 mm in najmanjši presledek (izolacija) prav tako 0.25 mm.
   Pri vezjih z večjo gostoto, zlati kadar uporabljate papir s tretjinskim rastrom, priporočamo, da ločite skoznike (metalizirane luknje, ki služijo izključno za to, da vezi prepeljemo skozi ploščo na drugo stran) od očesc, in jih na risbi tudi različno označite. Skozniki so namreč lahko nekoliko manjši od očesc, s čimer pridobimo na prostoru.
   Izjemoma je mogoče na zelo gostih vezjih potegniti tudi dve vezi med očescema premera 1.3mm, katerih središči sta razmaknjeni 0.1" oz. 2.54 mm.
   Priporočamo, da vse poševne črte zaradi estetskega videza rišete pod kotom 45 stopinj in se, če se le da, izogibate lomljenja črt pod pravim kotom.

7. Nekaj praktičnih napotkov:

- v sosednji ogliščí kvadratka, enote na mreži 2:1, ni dovoljeno postaviti očesc ne da bi se stikali

- če je očesce v vogalu kvadratka, ni dovoljeno narisati povezave po diagonali tega kvadratka

- narisati je mogoče dve očesci v nasprotnih ogliščih kvadratka

- priporočamo posebno pazljivost pri risanju debelejših povezav, kjer se boste najlaže izognili nezaželenim kratkim spojem, če bodo tudi debelejše povezave, tako kot očesca, narisane v merilu, ki je sorazmerno naravni velikosti

- med očescema, katerih središči sta razmaknjeni 0.1" (na predlogi dve enoti), je mogoče potegniti vez, debelo do 0.5mm.

Načrtovanje tiskanih vezij: Mitja Lasič in Albert Kolar
št. int. tel. 372