

Listing 1

VMIRR:	CALL ADD25L PUSH HL CALL ADD1L POP DE LD BC,#5064	:VMIRROR rutina.Potprogram ADD25L nalazi adresu zadnje linije na ekranu.Rezultat je u HL,koji se sprema na stack.Zatim se potprogramom ADD1L nalazi adresa prve linije u HL regstar.Adresa zadnje linije se stavlja u DE registar.BC sadrži dimenziju ekrana (B=80 bajtova širina i C=100 linija poluvisina) Tako se može promijeti dimenzija ekrana za zrcaljenje.
VMRRlp:	CALL #B906 PUSH HL PUSH DE PUSH BC	:KL L ROM ENABLE.Donji ROM je neophodno otvoriti.Zbog brzine se rutine NEXT BYTE i NEXT LINE zovu direktno u ROM.Iako je u listingu navedena FIRMWARE adresa,u inicijalizaciji se to mijenja.Svi parametri na stack.
VMRRdo:	LD C,(HL) LD A,(DE) LD (HL),A LD A,C LD (DE),A CALL #BC20 EX DE,HL CALL #BC20 EX DE,HL DJNZ VMRRdo POP BC POP HL CALL #BC29 EX DE,HL POP HL CALL #BC26 DEC C JR NZ,VMRRlp JP #B909	HL registar prati gornju polovinu ekrana,a DE donju.Ovdje se sadržaji (HL) i (DE) zamjene preko pomoćnih varijabli A i C.Kada se sadržaji donje i gornje polovine razmjene,potrebno je ponavljati razmjenu sve dok se ne dođe do kraja pojedinih linija.U tu svrhu se :SCR NEXT BYTE; HL pomiče za jedno mjesto udesno,a isto tako i DE.Treba primjetiti da bi se pri izradi ultra :SCR NEXT BYTE; brzog pomaka ekrana trebalo upotrijebiti neku vrstu LDiRa,kao i u rutini USCROLL i DSCROLL.Operaciju treba ponavljati do kraja linije.Sada se sa stacka uzimaju adrese početaka linija i zatim se pozivaju potprogrami za prijelaz na sljedeću :SCR PREV LINE; (tj.prethodnu) liniju.
		:SCR NEXT LINE C registar je brojač linija.Operacija sa zamjenom sadržaja linija se ponavlja do kraja ekrana. :KL L ROM DISABLE;Na kraju se donji ROM zatvara.
HMIRR:	CALL ADD179 PUSH HL CALL ADD1L POP DE CALL GETMSK INC B PUSH BC POP IX LD BC,#28C8 CALL #B906	:HMIRROR rutina.ADD179 potprogram nalazi adresu zadnjeg bajta u prvoj liniji.Ta adresa se smješta na stack,da bi se poslije nalaženja adrese prve linije u HL registar,povratila sa stacka u DE registar.
HMRRlp:	PUSH HL PUSH DE PUSH BC LD YH,B	Ovaj potprogram nalazi u B broj točaka u bajtu-1,a u C masku za krajnju lijevu točku,u ovisnosti o modu rada.Ispravan broj točaka u bajtu,zajedno sa maskom ide na stack,da bi tu vrijednost pokupio IX registar.
HMRRdo:	LD A,(HL) CALL RM LD YL,A LD A,(DE) CALL RM	B=polovina širine (40 bajtova),a C=broj linija (200) :KL L ROM ENABLE; I ovdje se zbog brzine potprogrami NEXT BYTE,PREV BYTE i NEXT LINE zovu direktno u ROM,pa je potrebno otvoriti donji ROM.
		Svi parametri idu na stack.Brzina je neophodna,pa se koriste i sintetičke instrukcije.
	LD (HL),A LD A,YL LD (DE),A CALL #BC20 EX DE,HL CALL #BC23 EX DE,HL DEC YH JR NZ,HMRRdo POP BC POP HL CALL #BC26 EX DE,HL POP HL CALL #BC26 DEC C JR NZ,HMRRlp JP #B909	HL registar prati lijevu polovinu,a DE registar desnu.Preuzete bajtove nije dovoljno samo izmijeniti,potrebno ih je i promijeniti tako da se krajnja lijeva točka nađe na mjestu krajnje desne itd.Za to je zadužen potprogram RM.
		Medumemorija je registar YL.
HMRRc2:		Nakon jedne promjene,potrebno je povećati HL registar :SCR NEXT BYTE; na sljedeći bajt,a DE registar smanjiti na prethodni bajt u video memoriji.Za to nam, :SCR PREV BYTE; naravno,služe potprogrami u ROMu.
		Brojač polovine širine ekrana se smanjuje za jedan i tako sve dok jedna linija nije "izzrcaljena".Kada se završi sa jednom linijom,obnavlja se brojač polovine linije,prelazi se na sljedeću liniju:HL na :SCR NEXT LINE; početak,a DE na kraj.I sada nam za ova računanja pomažu potprogrami u ROMu.
		:SCR NEXT LINE Brojač linija se smanjuje za jedan i tako sve do kraja ekrana. :KL L ROM DISABLE; Na kraju se donji ROM ponovo otvori.
RM:	LD B,YH LD C,YL PUSH HL LD H,0 LD L,A	Potprogram za zrcaljenje jednog bajta.BC poprima vrijednosti iz registra IY,koji sadrži broj točaka u bajtu (s obzirom na mode) i masku za krajnju lijevu točku.HL se sprema na stack.H se briše.L poprima vrijednost originalnog bajta.

MIRR:	LD A,L AND C RRC H OR H LD H,A RLC L DJNZ MIRR LD A,H POP HL RET	A=trenutno stanje L, koji se "vrti" ulijevo. Krajnja lijeva točka se izdvaja, H registar za međumemoriju se "vrti" udesno, a izdvojena točka se nadovezuje sa H. Na kraju se L zarotira ulijevo. Na taj način točke koje "ispadnu" s lijeve strane "ulaze" sa desne. Ovu operaciju valja ponoviti sve dok se ne obrade sve točke u bajtu. Na izlazu iz potprograma, u H registru se nalaze sve nadovezane točke, tj. "izvršeni" bajt. Vrijednost HL registra sa ulaza u potprogram se obnavlja.
LSCR:	CALL DOT7 CALL SETG LD HL,M0L+1 CALL HRINIT LD A,C CPL LD B,A LD YL,200 CALL ADD179 CALL #B906	:LSCROLL rutina. Ova rutina je vrlo slična sa RSCROLL, pa objavljujemo samo LSCR sa razlikama. DOT7 potprogram vraća u C registru masku za krajnju desnu točku u bajtu. SETG postavlja varijablu GPAP, kojom se boji novi red točaka, odakle su pomaknute stare, u ovisnosti od maske C. HRINIT u C nalazi masku krajnje lijeve točke, a u IX adresu potprograma za pomak pojedinog bajta, u ovisnosti o HL reg. na ulazu i o ekranskom modu. Za pomak ekrana ulijevo, to su 7,3 ili 1 RLCA, a za pomak udesno, :KL L ROM ENABLE; to su 7,3 ili 1 RRCA. YL je broj linija (200). Za LSCR, HL počinje od zadnjeg bajta prve linije, a za RSCR od prvog bajta prve linije. Širina ekrana je u YH i uvijek je 80 bajtova. Početak se boji sa bojom grafičkog papera. Dalje D služi za prijenos prethodno istisnute točke. U B registru se čuva antimaska kojom se izdvaja ostatak točaka u bajtu. Ovdje se on pomiče ulijevo sa RLCA, ali kod RSCR, to je zapravo RRCA (pomak udesno). Tako izdvojene točke se spajaju sa onom koja je istisnuta iz prošlog bajta (ili na početku sa bojom grafičkog papera). Sada je taj bajt kompletiran i vraća se natrag u video memoriju. Još treba izdvojiti i točku koja će se upotrijebiti za spajanje u sljedećem prolazu. Ta točka se izdvaja sa maskom C i onda se pomiče udesno (kod RSCR ulijevo) dakle u :SCR PREV BYTE; obrnutom smjeru. HL prelazi na prethodni bajt (kod RSCR na sljedeći, sa SCR NEXT BYTE). Na kraju se brojač bajtova u liniji smanjuje za jedan i proces se ponavlja do kraja obrade linije. Vrijednost HL :SCR NEXT LINE; se uzima sa stacka, a to je početak (kraj) prethodne linije. Prelazi se na sljedeću, brojač se smanjuje za jedan, i tako do kraja ekrana. :KL L ROM DISABLE; Kraj-zatvaranje donjeg ROMa.
LHROLLr:	PUSH HL LD YH,80 LD A,(GPAP) LD D,A	
LhLOOP:	LD A,(HL) LD E,A AND B RLCA OR D LD (HL),A LD A,E AND C CALL CALLIX LD D,A CALL #BC23 DEC YH JR NZ,LhLOOP POP HL CALL #BC26 DEC YL JR NZ,LHROLL JP #B909	
CALLIX:	JP (IX)	Pomoću ovog potprograma se simulira CALL (IX)
SETG:	LD A,(#B339) AND C LD (GPAP),A RET	Ovaj potprogram uzima enkodirani grafički paper i izdvaja samo onu točku, čija je maska u C. Rezultat se smješta u GPAP memorijsku varijablu. Adresa #B339 vrijedi samo na CPC 464, ali se promijeni u inicijalizaciji, pa program radi i na CPC 6128.
HRINIT:	CALL GETMSK LD D,0 LD E,B OR A SEC HL,DE PUSH HL POP IX RET	Potprogram koji nalazi masku krajnje lijeve točke u C, i računa adresu potprograma za pomak izdvojene točke u suprotnu stranu. HL na ulazu mora sadržati adresu posljednje instrukcije potprograma (a to je RET). 7 pomaka = mode 2 3 pomaka = mode 1 1 pomak = mode 0 Rezultirajuća adresa je u IX registru.
DOT7:	LD DE,7 LD H,D LD L,D JP #BC1D	Potprogram nalazi masku krajnje desne točke. Tu masku možemo lako naći ako nađemo masku za točku u 7. koloni, što vrijedi za sve modove. :SCR DOT POS; C=maska, HL=adresa (X,Y)
USCR:	CALL ADD1L LD D,H LD E,L CALL #BC26 LD IX,#BC26	:USCROLL rutina. HL=adresa prve linije. Tu vrijednost zapravo trebamo u DE registru. U HL nam treba adresa druge linije, pa ćemo je naći uz :SCR NEXT LINE; pomoć potprograma u ROMu. :SCR NEXT LINE; IX registar sadrži adresu potprograma :KL L ROM ENABLE; za prelaz na novu liniju. Donji ROM treba otvoriti. B=199 linija koje se pomiču, a C=80 bajtova širine. Parametri se stavljaju na stack.
SCRLRT:	CALL #B906 LD BC,#C750	Poslije spremanja treba izvršiti i premještanje jedne linije na mjesto druge. To se najbrže izvodi sa LDIR, ali treba paziti i na OFFSET, pa se stvari zakompliciraju u
SLOOP:	PUSH BC PUSH HL PUSH DE CALL #EA4	

