

Specijalno izdanje časopisa „Galaksija“ Jul 1984. / Cena 200 D

računari

u vašoj kući

škola
akcionalih igara

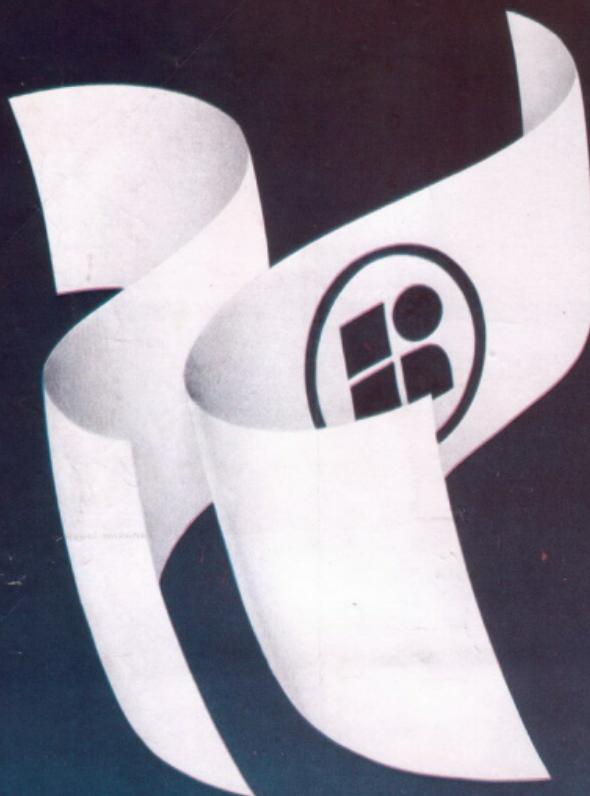
memorija
za „galaksiju“
48 K

QL • Electron • BBC B

2

Commodore 64





БЕОБАНКА
с буда са тана



БИКИН

Sadržaj

4/ ni golub na grani, ni vrabac na ruci

6/ na zapadu ništa novo

8/ u susret superinteligenciji

11/ era tehnoloških genija

16/ nemojte plakati za epлом

18/ računari u izlogu: QL, Elektron, BBC B, Commodore 64

32/ stampači u akciji

42/ moj prvi program

47/ biblioteka programa

55/ „galaksija“ bez tajni

66/ svi „spectrumovi“ restarti

68/ softverska veza

72/ škola akcionih igara

82/ programator eproma

88/ memorija za „galaksiju“

94/ „ubiću se, neće da radi!“

96/ konstruktorski hod po mukama

Na naslovnoj strani:

CRTEŽ SA KORICA KNIJGE „THE JOY OF COMPUTERS“, KOJA ĆE POD NASLOVOM „KOMPUTATOR U KUĆI“ IZAĆI OVE JESEN NI HRVATSKOSRPSKOM ODNOSENIO SLOVENAČKOM JEZIKU, U IZDANJU „CANKARJEVE ZALOŽBE“ (videti oglas na str. 38)

Izdaje

Beogradski izdavačko-grafički zavod
OOUR Novinska delatnost „Duga“
11000 Beograd
Bulevar vojvode Mišića 17

Telefoni

650-161 (redakcija)
650-528 (prodaja)
651-793 (propaganda)

Generalni direktor
Gojko Žecar

Direktor OOUR „Duga“
Zoran Milošević

Glavni i odgovorni urednik
Gavrilo Vučković
Urednik izdania
Jova Regasek

Likovna i grafička oprema
Dusan Mijatović

Redakcija časopisa „Galaksija“
Tanasić Gavranović, pomoćnik glavnog i odgovornog urednika
Esad Jakupović, zamenik glavnog i odgovornog urednika
Aleksandar Milinković, urednik
Jova Regasek, urednik
Zorka Simović, sekretar redakcije
Gavrilo Vučković, glavni i odgovorni urednik

Autori tekstova
Dejan Ristanović
Miša Hadži-Dorđević

Ivan Gerenčir
Nadejko Parezanović
Milan Pavićević
Borivoje Perzić
Jova Regasek
Slobodan Vujović
Andelko Zgorelec

Prevodioци

Domagoj Božić
Zdenko Dizdar
Gavrilo Vučković

Tehnička saradnja
Ljubiša Milovanović
Mira Todorović

Specijalna saradnja
Andelko Zgorelec, London

Izdavački savet „Galaksije“
Dr Rudi Debijadi, prof. dr Branislav Dimitrijević (predsednik), Radovan Drašković, Tanasić Gavranović, Živorad Glišić, Esad Jakupović, Velizar Maslač, Nikola Pajić, Željko Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Mihorad Teofilović, Vidojko Veličković, Velimir Vesović, Milivoj Vuković

Štampa

Beogradski izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17

Žiro-račun i kod SOKD

020-900-17125
Debitni račun kod Beobanke
60811-620-6-82701-999-01066

Za inostransku cenu dvostroška
(400 D. 3 US \$, 8 DM, 55 Sch, 7 Frs, 2 Lstg
Na osnovu miješanog Republičkog sekretarijata za kulturni
broj 413-77/72-03 i „Službenog glasnika“ broj 26/72,
ove izdanje ostvorenjeno je poreza na promet

ni golub, na grani ni vrabac u ruci

Zašto kasnimo?

Kada smo pre nepunih godinu dana u rednim godinama podeli da radimo izdanjima o jedinstvenim posebnim izdanjima koje bi u celini bilo posvećeno kućnim računari, nismo ni stališta šta se sve nalazi na drugom kraju filira koji smo time namerali da potpalimo. Računari u vajši kudi" i da se pojavili na tržištu prihvati. Danas januara, planuli su u tirazu od 50.000 primeraka, a mala „galaksija“ izrasta je preko noći u naš nacionalni računar. I ne samo to. Računari već mesecima ne silaze sa stranica štampe i programa radija i televizije, a poređ nekoliko računarskih nastava koji se već nalaze u izložima knjižara, izdavači najavljuju još slijest novih. Slovenci su pokrenuli već dva specijalizovana časopisa, a sasvim određene planove ima i redakcija „Galaksije“.

Računarski izazov probudio je, barem na rečima, i domaću elektronsku industriju, koja do nekoliko meseci, smatrali su da smo previše siromašni za to, o kućnim računarima nije želela čak ni da razmišlja. Uz „galaksiju“, „Iolu 8“ i varazdičkog „galeba“, originalne računare najavljuju i Elektronske Industrije, Iskra-Delta i Slovensales. I domaća kompjuterska kompanija nema odzivnije u Sil-4, koji, praktično, kućne računare još uvek drži pod zatvorom. To je razlog više da domaćoj industriji drštvo stisnemo palđave. Računarsku revoluciju, uostalom, možemo ostvariti samo ako budemo imali domaći računar.

I same redakcije je, ruku na srce, pomalo izgorela na vatri koju je zapaliće sopstvenim rukama. Akcija sa računaram „galaksija“ uvela je u redakciju vanredno stanje, koje traje već mesecima. Epromi, tastature, štampana kola, čipovi... neprekidno odvlače našu pažnju sa svakodnevnim poslova. To je malo usporilo pripreme „Računara 2“. Na ovom specijalnom izdanju, osim toga, radi veoma skromna ekipa ljudi

— mala redakcija i mala grupa spoljnih saradnika. Zbog toga nismo bili u stanju da se njim izdæmo, pred čitaoce, onda kada smo to želeli, ali smo redakcije poslove okončali znatno ranije nego što je naša štamparija, zbog nedostatka papira, bila u stanju to da predvratiti. To, naravno, ne znači da pred vas izlazimo, sa bajatim tekstovima. Većina medija njima ima trajnu vrednost, a oni kod kojih je aktuelnost posebno važna, dopunjavajući su do poslednjeg dana. Članak o računaru QL napisan je ponovo nakon što su u redakciju prispevi pri realni prikazi ne-predno pred ulazak izdanja u štampu.

Citav taj neplanirani vremenski po-mak uslovio je i izvezne promene u već najavljenoj konceptciji specijalnog izdanja. U obliku ideja i materijala i na ikak ogranichenom prostoru pojedine nagovestne rubrike i priloge morali smo da odfozimo za sledeći broj. „Računara“, izostao je, između ostalog, i najavljeni umetak: taj prostor, sa standardnim „prelomom“, ustupili smo izveznom broju tekstova koji će, verujemo, više odgovarati učenicima. Dodatno, nismo za sve ove promene bio je i povećan priliv oglasa, za koje se nadamo da nikako nisu „nužno zlo“ nego specifične informacije čiji značaj nije da podcenjuju.

Ako ovim izdanjem nismo u svemu zadovoljili svačiji ukus, molimo vas za malo strpljenja, pripreme „Računara 3“ već su uveliko u toku (nastojemo da iz štampe izđu krajem septembra) i u njima dele, verujemo, poređ ostalog nači i one priloge, koji su pri vešem mišljenju trebalo da nadu svoje mesto u publikaciji koju upravo čitate.

Redakcija

Ovogodišnji Sajam tehnike u Beogradu, koji je trajao od 22. do 26. maja, protekao je, više nego ljeden pre njega, u znaku kućnih i, naročito, malih izložbenih prostora. U hall 1, na ogromnom izložbenom prostoru, moglo je da se pogleda i proba obilje domaćih i stranih mini i mikroračunara, kojih, na žalost, i dalje nema u prodavnici. Sa posebnim uzbudnjem smo očekivali da vidimo da li je (i kako) domaća elektronska industrijia reagovala na „provokaciju“ zvanu računar „galaksija“. U nezvaničnim razgovorima sa zvaničnim licima na štandovima pokušali smo da saznamo nešto više o planovima domaćih proizvođača računara i modelima u koje se užala najveće nade.

Jugoslovenski računari, nije bilo teško zaključiti, nisu u pravom smislu „kućni“. Dimenzije su im, doduše, najčešće takve da sasvim lepo mogu da se smeste na vaš radni sto, mogućnosti dozvoljene za svu (pogoljivo kućnu), primene, ali sve to po ceni od koje vas odmahn zabilješki glava. Naši proizvođači su, dakle, najviše okretnuti firmama koje bi želele da se kompjuterizuju plaćajući ovaj tehnološki napredak u dinarima: maloprodaja interesuje slobodno koga (za maloprodaju, naravno, trebaju i posebne dozvole) i te najpre, zato što je u našim uslovima teško realizovati veliku i jefiniju seriju.

Šta sprema Ei

Obilazeći hall 1 u smeru kretanja kazaljke na časovniku, našli smo najpre na štand Tovarne Meril. Ona prodaje poznate Kopar tehnologije, koji su u poslednje vreme postali standard u mnogim računskim centrima. Ništa za vas, reklo bi se. Ali, na štandu je izložen i domaći računar PMP, koga se ne bismo postidelili ni na svetskom tržištu. Imali smo zadovoljstvo da porazgovaramo sa Marijanom Miletićem koji je, zajedno sa kolegom Brankom Jevtićem, konstruisao PMP, računar koji, kako ćemo videti, ima najmoderniju programsku podršku od svih kućnih modela. Ovaj kompjuter je, naime, zasnovan na mikroprocesoru TCT 11, koji predstavlja PDP 11 računar u malom. Operativni sistem je takav da svi programi pisani za PDP računare bez ikakvih problema rade i na PMP-ju. A poznato je da PDP računara ima ogroman broj, da se koriste bezmalo u svim računskim centrima u svetu, kao centralne ili periferijske jedinice i da, kako reče Marijan Miletić, on lično počinjan jedan primerek svoga računara onome ko navede jedan programski jezik koji im nije prilagođen! ROM računara PMP ima svega 2 KB i u njega je smješten samo kratki program koji učitava u RAM operativni sistem sa jedne od disketa. RAM u osnovnoj konfiguraciji ima 64 kilobajta, ali se, obzirom da je TCT 11 šesnaestobitni procesor, može praktično neograničeno proširivati.

Kako stoji stvar sa cennom? Centralna jedinica računara košta 150 hiljada dinara, a svaka disk-jedinica po sto. Slobodete se sa nama da ova cena, kada se uzme u obzir ono što se dobija, nije velika, i da ni strani računari, uz carine, ne bi koštali manje u odgovarajućoj konfiguraciji. Na žalost, postoji jedno ali: PMP ne može da se koristi bez terminala a terminal, zavisno od mogućnosti, može da košta 30—70 starih miliona! Utešila nas je jedino tvrdnja Tovarne Meril da rade na integralnom video delu za televizor i posebnoj tastaturi, koji će, ukoliko ispitivanje tržišta koje je u toku do povoljne rezultate, učiniti PMP pristupačnim širim krugovima korisnika.



Ne obrativši, možda i neopravdano, dovoljno pažnje Digitronom štandu, stigli smo da slediće izlagачa — El-Honeywell. Naša žurna nije bila neobična — „Politika“ je nedavno donela vest da će El uskoro izbaciti na tržiste prvi (svi koristimo taj epitet) kućni računar pristupan svakome. Čim smo videli da na štandu nema gužve, znali smo da na njemu nema ni tog računara. Tačno tako — El je izložio već dobro poznati Honeywell 6/10. Ovaj računar je zasnovan na Honeywell-ovom mikroprocesoru LS-6 koji se, radi kompatibilnosti, može dopuniti čuvenim Intelovim 8086. Korisnička memorija od 128 kilobajta može da se proširuje do 512 Kb, dok su na asinhroni komunikacioni port priključene dve disk-jedinice po 650 Kb. Oni koji žele da obraduju veće količine informacija mogu da dodaju Winchester masivne diskove po 20 Mb. Štampač se naknadno dopušta i može biti 132-kolonski matrični (100 karaktera u sekundi) ili sa lepezom — 30 karaktera u sekundi. Za cenu ovakve konfiguracije se nismo rasipitali — smatrali smo da vas neće interesovati.

Od „galeba“ do „orla“

Na sledećem štandu smo ugledali dobro poznati računar (Apple II) i dobro poznatog kolegu (Julija Makaneća mladeg). Došli smo, jasno, do štanda Velebita. Čitaocima „Galaksije“ je, svakako, poznato da su Varaždinci pre tačno godinu dana („Galaksija“ broj 135) izbacili na tržiste seriju od 40 računara „galeb“. Ovaj računar, zbog predugod razvoja i unekoliko zastarele concepcije koju je dopunila veoma visoka cena 90.000 dinara, nije zaživio na tržistu i, po svemu sudeći, pripada prošlosti. No, „galebov“ naslednik koji se zove „orao“ treba uskoro da uzleti i zauzme mesto na našim stolovima. „Orao“ je logičan nastavak „galeba“ i delo je istog konstruktora — Mire Kocijanca. Zasnovan je na mikroprocesoru 6502 i poseduje ROM od 16 i RAM od 6 Kb koji može da poveća i do 128 Kb (pozivajući se na mikroprocesor 6505 i razgovoru, što znači da neke od njih mogu i da se promene). Kao periferijske jedinice koriste se standardni crno-beli televizor na kome računar kontroliše finu grafiku sa 256x128 tačaka i kasetofon, ali postoji i RS 232 port za dalju proširenja, a u računar može da se ugradi i disk-interfejs. Bežik bi, obzirom na pristojan kapacitet ROM-a, trebao da bude vrlo dobar i dopunjene grafičkim rutinama (crtanje krugova itd.) ali, kako izgleda, ne i asemblerom koji bi omogućio jednostavno pisanje mašinskih programi.

„Orao“ će, izgleda, koštati koliko i „galeb“ — oko 9 starih miliona. Može li, kao takav, da konkuriše Spectrumu koji se prodaje preko malih oglasa i, za duplo manje novca, nude enormno programski podršku? Na to pitanje mi je teško odgovoriti; kakav god bio „orao“ je ipak naš računar. „Velebit“ jasno pokazuje da smatra da je njegov novi računar maksimalno

konkurentan — izrađeno je 1000 štampanih ploča, što je za našu prilike ogromna serija. Međutim, na sajamskom štandu još nema prototipa ni prospakata; izgleda da izgleda da razvoj „orla“ (pogotovo njegovog softvera) još nije završen!

очекujući „orla“, ostalo nam je samo da pogledamo Apple II računare. Velebit će već u trećem kvartalu ove godine prodavati poznati Apple IIe, koji će, sa jednom flopi-disk jedinicom, monitorm i štampačem Epson MX 100 ili FX 20 koštati nešto manje od 100 starih miliona. S obzirom na moćnu programsku podršku koju Apple ima, čini nam se da za mnoge škole predstavlja prilično zanimljivu alternativu.

„Partner“ i „lola 8“

Iskra Delta je i ove godine izložila poznati Partner. Radi se o poslovnom sistemu koji je zasnovan na mikroprocesoru Z80A, opremljen sa 128 Kb korisničke (RAM) memorije, jednom disk jedinicom od 1 Mb i Winchester masivnim diskom od 10 Mb. Tastatura je odvojena od centralne jedinice (veza se ostvaruje preko kabla), na kojoj se nalazi i crno-beli monitor koji omogućava ispisivanje 80 znakova u svakom od 24 reda. Pošto je ostvarena CP/M kompatibilnost, Partnerovim korisnicima su na raspolaganju mnogobrojni programski jezici, ali izgleda da su mnoge firme zainteresovane samo za računar koji će izvršavati odredene knjigovodstvene poslove. Za njih se Partner isporučuje bez programskog jezika i opremljen samo potrebnim programom. Cena računara u standardnoj konfiguraciji koja uračunava iTRS štampač i jedan programski jezik iznosi 330 starih miliona.

Završavajući krug, stigli smo do kompjuterskog štanda „lve Ribe“ sa desetak računara lola 8 na kojima su zainteresovani mogli slobodno da rade. Lola 8 je prisutan već dve godine i polako (nadajmo se i sigurno) kriči put ka tržistu. Operativni sistem računara je definitivno završen. Određen broj primera lola 8 je već našao svoje mesto u školama, a njima se pridružio i računar koji su na ovogodišnjem republičkom takmičenju u programiranju osvojili učenici beogradske Matematičke gimnazije „Veljko Vlahović“. Na sajmu je održano i Savezno takmičenje, koje se nije baš slavno završilo: učestvovalo su tri ekipe (Beograd, Novi Sad i Zagreb), kojima su nadležni na sajmu u 18 casova jednostavno isključili struju i tako obrisali programe koji su trebali da budu pisani još čitav čas!

Tri nove domaće kućne računare „galeb“, „galaksija“ i PMP — predstavljaju, bez sumnje, bogatu konstruktorsku žetu. Nove „prve domaće“ računare najavljuju i Elektronska Industrija, Iskra Delta i Slovenijales. Bilo bi nam draga da se porodica YU računara još više proširi — možda čemo po broju modela uspeti da prestignemo Englezef — ali bismo, iznad svega, voleli da se barem jedan od njih pojavи u knjizama, robnim kućama i, razume se, našim domovima. Plašimo se da je i ove godine, sa padom zavese na sajamsku priredbu, ne zaboravimo i na sve sjajne planove do sledećeg — sajma.

Dejan Ristanović

Pismo iz Slovenije

SLOVENAČKI RAČUNARSKI BUM

Ono što se sada dešava u Sloveniji mogli bismo da okvalifikujemo kao „kompjuterski bum“. Prošlo je više od godinu dana od kako je slovenački mesečnik SUPER-PIP počeo da objavljuje prve skromne priloge sa područja savremene elektronike. Ali svi se menjaju, potrebe rastu, a i želje mlađih za kompjuterskim znanjem sve su veće. Zato nije slučajnost što su tom problematikom počeli da se bave i drugi slovenački listovi i revije.

Neobuđajeno u svemu tome je što su se ovih dana na slovenačkoj novinskoj tržištu, koje i nije osobito veliko, pojavile jedna za drugom dve računarske revije: BIT, koji će od sada izlaziti jedanput mesečno, a izdaje ga „Ljubljanski dnevnik“, i posebno „Telekovo“ izdanje MOJ M-KRO. Tačku na „i“ dodali su „Radio študent“ i ZOTK (Savez organizacija za tehničku kulturu) Slovenije sa PROGRAMSKOM KASETOM, na kojoj se nalazi 10 programa za kućni računar Sinclair ZX SPECTRUM. Programi su isključivo rad domaćih autora, koji su se predstavili javnosti 6. juna ove godine na konferenciji za štampanje u Cankarjevom domu u Ljubljani.

Mlađi su to momci, mlađi kao što je mlađi i razvoj računarstva kod nas, i zato mislim da će biti pošteno da navedem i njihova imena: Žiga Turk, Matevž Kmet, Darko Vovk, Tomaž Iskra, Dragan Vidic, Primož Jakopin, Janez Kančić, Andrej Vlblešić i Dean Možetič. Ova programa su takozvane igre, a jedna od njih, KONTRABANT (avantura-krijumčarenje) ima sva obeležja domaćeg „folklor“. Cilj ove igre je nabaviti kućni računar, koji, na žalost, još uvek mora da se krijumčari. Kompjuterska igra KONTRABANT je, u sustini, putovanje po Jugoslaviji i zato je, kažu, dosta teška. Da bi se nagradio trud najboljih i najistrajnijih „igraća“, obezbeđeno je nekoliko primanjivih nagrada, među njima i INTERFACE i ZX Spectrum. Tačne propozicije će biti objavljene na kompjuterskim stranicama slovenačkih listova.

Možda ćemo ipak imati sreće, kažu mlađi iz Slovenije. Sreća da do kućnih računara ne moramo više da dolazimo – kršenjem zakona. Naime, Savez socijalističke omladine Slovenije (ZSMS) uputio je saveznom i republičkom Izvršnom vijeću predlog da se dozvoli uvoz kućnih i džepnih računara. Ako želimo da se oslobodimo kompjuterske nepismenosti, za šta se zauzima i ZSM Slovenije, potrebno je da svi mlađi, od

osnovne škole pa dalje, dobiju osnovna računarska znanja u vaspitnoobrazovnom procesu. U tom procesu trebalo bi da učestvuju zavodi za školstvo, komiteti za obrazovanje i vaspitanje i druge obrazovne institucije.

Jugoslavija je jedna od retkih zemalja u kojima je uvoz kućnih i džepnih računara faktički onemogućen. Kada bi se lični računari svrstali u predmete široke potrošnje, bilo bi omogućen uvoz i, što je još važnije, njihova konzignaciona prodaja. Ako tome dodamo i razvoj domaće proizvodnje ličnih računara (koji, doduše, ne mogu da konkurenči stranim računarima ni po ceni ni po snazi), to bi već bio ogroman korak napred. Savezni sekretarijat za spoljnu trgovinu obznanio je da je u toku rad na izmenama i dopunama Odluke o uslovima pod kojima fizička lica mogu uvoziti i primati određene predmete iz inostranstva. Da li će se nešto promeniti u tretmanu kućnih računara?

U Iskri obavještavaju da će se već ove jeseni na tržištu pojaviti prvi kućni računari HR-84 domaće proizvodnje – prvi 500 kompjutera. Iz serije od 2.500 komada. Do tada bi Iskra trebalo da okonča i povegore oko konzignacije Sinclair-ih računara.

Sasvim mali broj Sinclairovih računara ZX81 iznala je u maju na tržištu i Mlađinska knjiga iz Ljubljane. Međutim, to su računari koji imaju svega 1K RAM-a, što znači pola kucane stranice naredbi, pa zato stoji pod znakom pitanja njegova upotrijebljivosti za osnovne i srednje škole. Bilo kako bilo, možda su to ipak prve svetle tačke na horizontu računarske pismenosti kod nas. Ipak, jedno je sasvim sigurno: mlađi imaju i dovoljno energije i dovoljno znanja da iznesu kompjutersku revoluciju, ali, na žalost, za sada manje-više samo samonikli entuzijasti.

I baš zbog toga možemo da smatramo prvu slovenačku kasetu sa prvim domaćim programima za značajan događaj, za hrabar i pohvalan čin, utoliko pre što je softverska redakcija Radija-student mlađa – oformljena tek ove godine.

Tiraž kasete je 2.000 primeraka, cena 700 dinara (što je malo ako imamo u vidu da se u inostranstvu za jedan program plaća od 5–7 funti). Najvažnije u svemu je da oni za jesen pripremaju, kao što su obavestili na konferenciji za štampu, novu kasetu sa područja obrazovanja i saobraćaju. Dakle, prva slovenačka lasta – nije i poslednja.

Kristina Iskra

Sajam u Los Andelesu je osnovan još davne 1976. Njegov osnivač je Džim Voren (Jim Warren) – bivši hipik, urednik nekoliko kompjuterskih časopisa, nastavnik matematike, i, na kraju, doktor kompjuterskih nauka sa čuvenom Stanford univerzitetom blizu San Franciska. On je pošto je dobro poznao situaciju u Silicijumu dolini (južno od San Franciska), nekad čuvena po šljivicima i suvimi šljivama, a danas centar svetske moderne tehnologije, gde se 1976. puno toga događalo u vezi sa kućnim računarama, odlučio da organizuje skromnu izložbu. Ali, već pre godine interes je premašio sva očekivanja i mala izložba je prerasnula u sajam.

Džim je uvek bio široke ruke i nije odbijao nijednog izlagачa – uvek bi se i za najmanju firmu ili pojedinca našao neki kutak. Prća se, tako, da osnivač APPLE-a Steve Dzob (Steve Job) nije imao nekoliko stotina dolara da 1977. platí za izložbeni prostor, ali Džim ga nije odbio, već je pristao da čeka na novac. Tako je Apple predstavljen javnosti. Znamo šta će kasnije dogodilo sa ovom kompanijom. No, Džim je prošle godine prodao sajam, jer kako kaže, nema više onog zadovoljstva koje prvi dana.

Posetilaca sve više, ali nema varnica

Kako je ove godine bilo u San Francisku? Na žalost, nema više malih izlagачa, koji iznajaduju velikim idejama – sve same velike kompanije, a među njima gotovo polovina izdavači časopisa i knjiga. Kako piše novinar dnevnog lista San Franciske Kronikli (San Francisco Chronicle), „posetilaca sve više, ali nema varnica“. Naravno vodeći IBM i APPLE zauzimali su najveći prostor. Apple, koji ove godine troši 15 miliona dolara na promociju Mekintosa (kakva razlika u odnosu na onih nekoliko stotina dolara iz 1976). Izložje je detinjarski visoki „Mek“ na čijem su se velikom ekranu silikovito pokazivalo mogućnosti ovog računara. Bilo je i nekoliko starijih modela Apple II-e i Iise, ali na njih niko nije obraćao pažnju.

IBM-ov izložbeni prostor bio je najviše posvećen novom PC junioru, uobičajenom kompjuteru, ali je posebito ostavio prilično ravnnovrđenim! Dobri poznavaoci tržišta tvrde da se on slabo prodaje (vad dopisima je obišao više radnji u Kaliforniji i dobio je isti utisak). Od ostalih novosti sa sajma je to da je OSBORNE, posle bankrotstva, ponovo oživio (njegov osnivač Adam Osborne je smjenjen – sada vodi jednu softversku kompaniju). Interesantan je bio i štand japskog nacionalnog Mikao udruženja – oni su prikazali najnovije proizvode japskih kompanija.

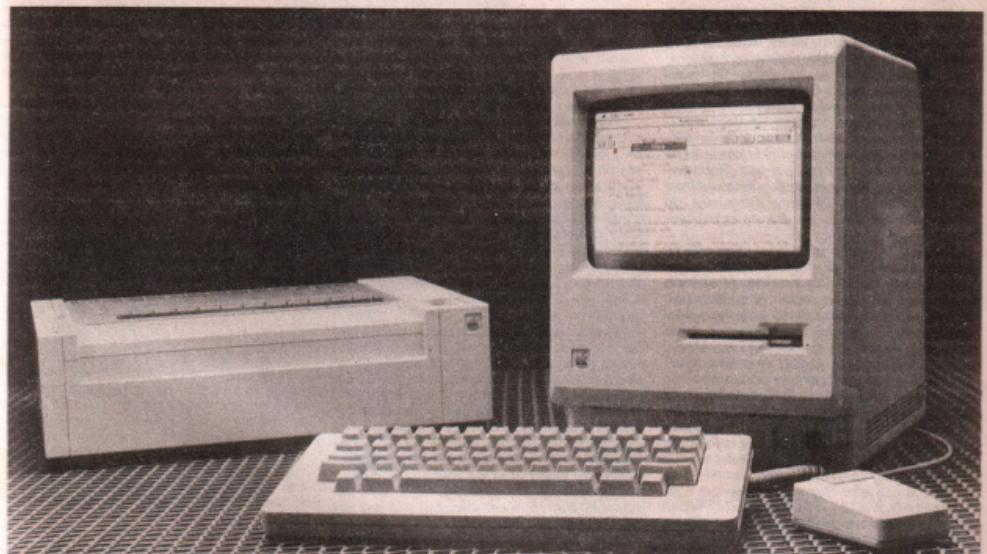
Sa knjigama i časopisima nastupilo je mnoštvo izdavača. Dosta interesovanja izazvala je i humoristička knjiga „Mrzni kompjutere“ (I hate computers). Svaki računar, kao i u Evropi, prati po nekoliko časopisa. „Mekintos“ ih sadrži i ima već dva (Macworld i St. Mac), a izdavači najavljuju još pet novih.

I pored visoke cene ulaznica (12 dolara) sajam je posetilo preko 40.000 entuzijasta. Na američkim sajmovima se, uzgred, uporedno održavaju i besplatni seminari, koji mogu pristupovati svakoj posetilac. Na njima se govorile najpoznatija imena iz ovog područja i može se naučiti dosta toga.

na zapadu ništa novo

Računari u sve

U proleće svake godine na zapadnoj obali Amerike održava se nekoliko sajmova za kućne računare na kojima se najbolje može videti koji vetrovi duvaju u ličnoj informatici. Vaš dopisnik već sedam godina napravi bar jedno putovanje preko Atlantika da vidi nove proizvode i trendove. Tako je bilo i ove godine. Posetili smo najstariji sajam te vrste, IX West Coast Computer Fair u San Francisku, i sajam Comdex (zatvorenog tipa) u Los Andelesu. Dok je prethodnih godina bilo iznenadenja i uzbudjenja na pretek, prošle, a pogotovo ove godine, ovi sajmovi su počeli da gube svoj tradicionalni šarm. Od kako su poslednjih godina velike multinacionalne kompanije zavladale ovom industrijom, nema više genijalaca koji će u svoj garaži napraviti novi računar ili u studentskoj sobici sastaviti softverski program koji će se prodati u 100.000 primeraka. Da li starimo ili smo previše romantični?



Na visokoj nozi: Tridesetdvobitni Macintosh, nova nada kompanije Apple i čitave američke lične informatike, košta svega oko 2.500 dolara

Računari, teleografi i . . .

Sajam u Los Andelesu je drugačijeg karaktera — trebalo bi da bude samo za ljude iz kompjuterske industrije. Međutim, ko god je bio raspoložen da plati 25 dolara, dobijao je bez sa napisom „gost izlagач“ koji mu je otvarao vrata u dvoranu. Uz oko 350 manjih izlagачa, bili su prisutni i svi računarski džinovi: IBM, APPLE, RADIO SHACK, DIGITAL. Uz dve britanske (Acorn i

Memotech), bilo je, takođe, i nekoliko japanskih kompanija. Najviše prostora zauzeila je sa svojim računarom džinovska kompanija ATT (American Telephone and Telegraph) koja ima skoro monopol u Americi na telefonsku i telegrafsku mrežu (za razliku od mnogih zemalja u svetu, ovde je to u privatnim rukama — samo je poštanska služba u državnim).

Pojava ove kompanije na tržištu računara predstavlja prvorazredno iznenadenje. Možda će od toga zaboljeti glava mnoge proizvođače — ne toliko zbog samog računara koliko zbog velikih finansijskih sredstava koje nameravaju da ulože u publicitet i propagandu. I poređ svih pokušaja, japanskoj industriji još uvek ne polazi za rukom

da zauzme iole značajniju ulogu na tržištu. Sada su uveli novu taktiku — neki njihovi računari se prodaju pod imenima RADIO SHACK, COMPAQ, jer ih oni proizvode jeftinije od Amerikanaca. Na tržištu periferijske opreme, posebno štampača i disk jedinica, Japanci vode glavnu reč. Posle pobune koju je ovdje doživeo Acorn BBC (ali to je prošlo), najpoznatija britanska kompanija ne postiže neki veći uspeh. Acorn se prodaje školskim institucijama, ali dosta slabije jer je sa cenom od oko 1.000 dolara previše skup.

One godine, dakle, dosta razočarenja. Sledće će, možda, biti bolje.

7/na zapadu ništa novo

Andelko Zgolec

Evolucija mozga od dinosaura do kompjutera *u susret superinteligenciji*

Zamislite tiranosaura rekса, najmoćnijeg među dinosaуrима, kako se osvrće oko sebe u bujnoj prašumi jednog davnog svestra. Visok je dva sprata; bedra mu se nadajuju od snage; u njegovim razijapljenim čeljustima bleska nekoliko redova zuba oštrih kara brijac; on je jedna moćna mašina za razraje. Ali u njegovoj ogromnoj lobanji smreštan je majušan mozak. On je glupažnica.

Dok se mrak spušta, jedno malo, dlakovo stvorenje izlazi iz svog skrovišta, gongeno ljuom gladi, i kreće oprezno u okolno špiraže po svom obroku crva i insekata. Tog trena ogromna telesina tiranosaуra zamraćuje nebo; njena senka pada na malu životinju i ova žurno beži tražeći spas.

To stvorenje je jedan od prvih sisara na Zemlji. Ono nije impozantna borbenja mašina, ali je intelektualni džin svoga vremena; ono je najinteligentniji oblik života koji se razvio na planeti do tog perioda. Srazmerno težini svog tela, ono je 5 puta pametnije od tiranosaуra, a 20 puta pametnije od biljnjadih dinosaуra kojima se tiranosaуr hrani.

Zašto je taj rani sisar bio toliko pametniji od dinosaуra? Niko nema pouzdani odgovor, ali verovatni razlog je taj što je on tada imao male šanse na uspeh, a pritisci pod kojima je živeo nametali su potrebu velike inteligencije u borbi za opstanak. Malo stvorenje mora da je bitisalo u stanju neprekidne strepnje, krijući se preko dana i tragajući noću za hranoj pod mučnim i neizvesnim okolnostima. Bespomoćno, moralo je da živi od svoja pameti.

Mada su se sisari i dinosaуri medusobno veoma razlikovali, oni su bili rođaci, jer su poticali od iste loze drevnih reptila. Drevni reptili, sa svoje strane, razvili su se iz amfibija, a amfibije su potekle od riba. Svi ti oblici života — ribe, amfibije i reptili — bili su relativno neinteligentni, a i njihovi potomci ostali su takvi sve do danas. Ali sisari su bili drukčiji; priča o evoluciji inteligencije u stvari je *njihova* priča. Pojava sisara označila je prvi veliki korak u evoluciji mozga.

Gospodari planete

U početku, sisari — ili, tačnije rečeno, reptili nalik na sisare, dakle jedan prelazni oblik — bili su svirepi, mnogobrojni i strahovito uspešni; u stvari, pre 250 miliona godina oni su bili dominantan oblik života na kopnu. Ali kasnije, iz nekog razloga, opali su u broju i veličini i potonuli u tamu, dok su ih dinosaуri nasledili kao gospodari planete.

Sisari su ostali potčinjeni vladajućim dinosaуrima više od 100 miliona godina, promenivši se vrlo malo tokom tog dugog intervala. A onda, otrplike 70 miliona godina unazad, klima na Zemlji počela je neglo da se menjaj, a dinosaуri, ne više u harmoniji sa svojom životnom sredinom, postepeno su isčezli. U roku od svega nekoliko miliona godina potpuno su nestali. Sisari, sada odjednom nesputnici, uselili su se u životne prostore koje su upraznili mrtvi dinosaуri i rasvetali se u čudesno mnoštvo raznovrsnih oblika. Neki su se opredelili za vazdušni i postali letaci; drugi su se vratili moru i postali kitovi i delfini; treći su ostali na šumskom tlu; a jedna grupa se uspentrala na drveće da bi postala preteča majmuna i čovekolikog majmuna.

Poput njihovih srodnika koji su istraživali i osavljali druge životne prostore, ovim životinjama nastanjenim na drveću bile su potrebne specijalne telesne odlike koje će ih opremiti za taj novi život: ruke za dohvatanje grana, na primer, i binokularni vid za procenjivanje razdaljine od grane do grane. Tokom vremena, te odlike su evoluirale. Ali stanovnik drveća morao je da ima veći i bolji mozak. Kad je skakao s jedne grane na drugu bio mu je potreban brz, precizan kompjuter u glavi za kombinovanje faktora razdaljine, brzine vetro, pokreta grane i ravnoteže tela. A izračunavanje je moralo da bude obavljeno u magnovremenu, jer bi svaka pogreška bila plaćena smrću ili teškom povredom. Tom prirodnom kompjutertu — mozgu — bili su potrebna prefinjena mentalna kola za obavljavanje neophodnih proračuna, a bila mu je potrebna i velika bavna memorija za skladištenje rezultata bivših istkustava u vazdušnim manevrima.

Drevni sisari, koji su ziveli na šumskom tlu, posedovali su jednostavniji mozak bez tih izvanrednih kvaliteta. Ali još jednom zakon opstanka najposposibilnog stupio je u dejstvo. Zahvaljujući malim varijacijama od jedinke do jedinke koje se dešavaju u svakoj populaciji, neke životinje su posedovale nešto bolji mozak od drugih; te životinje su imale veće šanse na opstanak, prenošiće svoju superiorniju moć mozga na potomstvo; dok su one koje su bile u manjoj meri obdarene tim mentalnim odlikama pokazivale sklonost da nestanu u ranom dobu. I njihovi geni su isčezli iz populacije.

Dejstvo selekcije

Pod dejstvom prirodnog odabiranja možak stanovnika drveća poboljšao se i u pogledu kvaliteta i u pogledu veličine; istovremeno, njegova manuelna spretnost i oštRNA ruka nastavili su da se razvijaju. Nakon 40 miliona godina neprekidnog poboljšavanja tela i mozga, potomci prvog stanovnika drveća postali su nova vrsta stvorenja. Ta nova stvorenja bila su ma-

mun — i, njegov rodak, čovekolikog majmuna.

Majmun se nije mnogo promenio od vremena do danas; njegova je priča bila završena. Ali evolucija čovekolikog majmuna se nastavila. Postao je veći i teži, i sišao je sa drveća da bi otišao novo bitisanje na šumskom tlu, vrativši se prebivalištu preko njegovih majušnih preci napustili 40 miliona godina pre toga.

Sada je bila postavljena pozornica za još bržu evoluciju inteligencije.

Pre otrplike 15 miliona godina, jedna napredna grupa čovekolikih majmuna — predaka šimpanza i gorile — nastanjivala je šume istočne Afrike. Još jednom, promena na tlu planeti skrenula je tok evolucije. Nastupila je era hladnjeg i suvijeg vremena, uslovivši pojavu otvorenih komada zemljišta za napasanje. Dok je suvo vreme nastavljalo da traje, pašnjaci su se sve više širili, i teren se transformisao u poznatu afričku savanu.

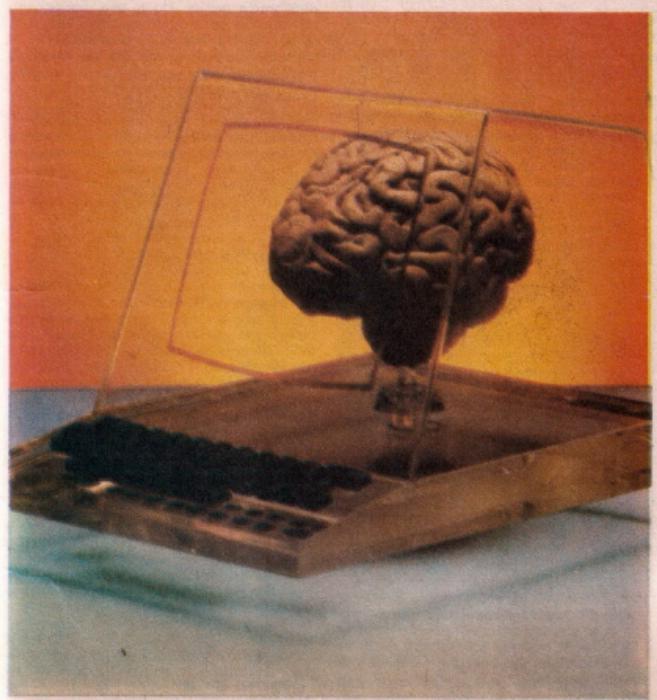
Zivotinja slična čoveku

Savana je imala privlačnu draž nepoznatog za čovekolike majmune. Postepeno, najsmeliji i najradoražniji među njima napustili su bezbednost drveća da bi istražili tu čudnu novu sredinu. Ne znamo šta se iza tog desilo, jer u fosilnim nalazima tada nastupa jedna praznina od nekoliko miliona godina. Znamo samo da se šumskim čovekolikim majmumima otvoreno zemljište dopalo i da su ostali tam, jer kada se fosilni zapis ponovo pojavljuje, on pokazuje da su se njihovi potomci uvelikoj odomaćili u savani. Ali potomci nisu više čovekoliki majmuni; oni su jedna nova životinja koja hoda uspravno, korakom nalik čovekovom. A ta nova životinja ima superioran mozak; u odnosu na telesnu veličinu, mozak je dva puta veći od mozga čovekolikog majmuna.

Priema fosilnim zapisima, ta dvojna, inteligentna životinja pojavila se u Africi pre tri do četiri miliona godina. Njeno ime je australopitek afarensis. Bio je sposoban lovac koji je jurio sa ostalim mesožderima svog vremena, nadmećući se sa lavom i džinovskom hijenom. Međutim, australopitek je bio slabšana, nejake grade, težak možda samo oko trideset pet kilograma, i bez oštRNA očnjaka ili drugog prirodnog oružja. Ali njegove fizičke slabosti bile su kompenzirane snagom njegovog intelekta.

Koje su okolnosti stvorile tu neobičnu moć mozga australopiteka? Odgovor nije jasan, ali proučavaoći ljudske evolucije ugovljujemo se slaju da je uspravno držanje bilo ključni faktor. Sa rukama određavno oslobodenim za nošenje hrane i oružja, a za izradu i korišćenje oruđa, australopitek je imao potencijal za stvaranje kulture. Ali kako je taj potencijal bio realizovan? Kako su te telesne odlike dovele do porasta mozga?

Ako je prošlost neki putokaz za budućnost, onda možemo očekivati nastanak jednog višeg oblika inteligencije. Kako će izgledati taj prefinjeni mozik? U čemu će se razlikovati od našeg? Da li elektronski računari, pa i ovi kućni, predstavljaju samo primitivni zametak buduće superparametra ili se radi o stranputici na putu za budućnost? O tome piše Robert Dzestrou (Jastrow), američki naučnik, publicista i eseista.



Teorija o prirodnom odabiru nudi odgovor. Prvi izradivači oružja u toj grupi dvonogih čovekolikih majmuna mora da su se razlikovali međusobno po stepenu dovitljivosti i ingenioznosti koji su upražnjavali u svom svakodnevnom životu. Oni koji su posedovali te osobine u najvećem stepenu bili su istinski junaci dana; oni su imali najviše izgleda da načine efikasna oružja za ubijanje divljaci i najbolju strategiju za lov. Oni su imali uspeha; oni i njihovi potomci su napredovali; njihovi geni su se sve više ukorenjivali u populaciji; odlike dovitljivosti i ingenioznosti intenzivirale su se među dvonogim čovekolikim majmunama.

Dovitljivost i ingenioznost — to su dva lica inteligencije, a njihovo sedište je u

mozgu. Ako se ovakve odlike intenziviraju, mozik mora da poraste. Na taj način je uspravno držanje, koje je dovelo do izrade i korišćenja oružja, doprinelo ubrzanjem porasta mozga u populaciji dvonogih čovekolikih majmuna.

Automatske poruke

Mozak koji je rezultirao nije bio veliki u apsolutnom smislu; imao je tek veličinu pesnice i težio jedva 450 grama; s druge strane, međutim, ni telo koje je kontrolisao nije bilo veliko. U procenjivanju inteligencije neke životinje po veličini njenog mozga, važno je izračunati proporciju veličine ili težine mozga u odnosu na telesnu težinu, zato što svaka životinja koristi jedan deo svog mozga za nesvesnu kontrolu tela. Taj segment mozga služi kao telefonska razvodna tabla, koja prima signale od čula i, sa svoje strane, odasliće automatske poruke. Beži od opasnosti! Približi se hrani! I

tako dalje. Samo višak sive materije, koja preostaje posle tih automatskih funkcija, stoji na raspolažanju za postupke koji mogu da dodu pod kontrolu inteligencije, kao što su pamćenje, planiranje i fleksibilno, neprogramirano reagovanje na nove izazove. Australopiteku, sa malim telom, bio je potreban relativno mali broj moždanih ćelija za telefonsku razvodnu u njegovoj glavi, pa je zato imao na raspolažanju veću količinu sive materije za pamćenje, planiranje i apstraktno mišljenje.

Australopitek je stekao tu moć preko dva ili tri miliona godina, nekoliko miliona godina nakon što su se njegovi drevni preci prvi put preselili u savanu. Izmenio se donekle tokom nekoliko sledećih miliona godina, evoluirajući uspešno u životinju poznatu pod imenom australopitek afrikansus, a zatim u telom krupnijeg australopiteka robustusa. Posle toga se promenio veoma malo, kako u pogledu mozga tako i u pogledu tela. Poput mnogih drugih oblika života u prošlosti, australopitek je došao u čorak. Pre sedamstotin pedeset hiljada godina potpuno je izumro.

Pre nego što će isčezenuti, australopitek je iznudio drugu inteligentnu životinju afričke savane. Novo stvorenje je nazvano homo eriktus — čovek koji stoji uspravno. Njegovo držanje bilo je uspravno, kao u australopiteka, ali plemenitije; njegovo telo i crte lica bili su više čovekolik a manje majmunoški; a njegov mozik bio je veći.

U početku, razlike u veličini mozga nije bila osetna, ali inteligencija australopiteka ostala je gotovo nepromenjena, dok je inteligencija homo eriktusa i dalje rasla. Najnovija etapa u evoluciji inteligencije na Zemlji tek što nije bila otpočela.

Neumitni trend evolucije

Sada nas je priča dovela do jednog vremena, pre otprilike milion godina, kada je australopitek upravo trebalo da isčezenje iz savane. Homo eriktus već je lovio sa zapanjujućom veštinsom; nijedna divljač nije bila suviše velika za njegovo junaštvo; a njegova tehnologija kamernog oružja bila je impresivna. Odjednom, rast mozga se ubrzao. Neko ne zna tačno zašto; smatra se da oružja, jezik i pritisici ledenerog doba spadaju među glavne razloge; ali, ma šta da je bilo uzrok, fosilni nalazi demonstriraju rezultate.

Blizu 450 grama sive materije bilo je pridodata ljudskoj lobanji za manje od milion godina. Veći deo bio je u prednjem delu mozga, koji je centar najviših domena apstrakcije i kreativnosti. Dok su se dodatni neuroni gomilali u lobanji ranog čoveka, njegovo čelo se ispuščavalo, lice je izgubilo svoj zverski izraz, i on je počeo da poprima izgled intelektualnog bića. Homo sapiens — čovek obdaren razumom — stigao na planetu.

Danas čovek predstavlja vrhunsku tvoreninu prirode na Zemlji. Kakva budućnost predstoji toj izuzetnoj životinji? Možda će i on izsečnuti, kao što je australopithecus izumro pre njega; devedeset odsto svih oblika života koji su postojali na Zemlji vremenom su izumrli. Ili će, možda, živeti neizmenjen sve do u daleku budućnost — jedan živi fosil poput školjke kameoice. Ta sudbina možda se već nadneće nad nas, jer ljudsko telo se izmenilo veoma malo tokom poslednjih milion godina, a ljudski mozak se nije izmenio, bar ne u ukupnoj veličini, tokom proteklih 100.000 godina. Organizacija mozga možda se poboljšala za to vreme, ali ograničene su količine informacija i uzajamnih poruka koje mogu da budu stranpe u lobanju fiksirane veličine. Činjenica da se mozak više ne širi, posle milion godina eksplozivnog rasta, sugerira da je priča o ljudskoj evoluciji možda već završena.

To ne znači da je završena i evolucija inteligencije. Naišme, razumno je pretpostaviti da ljudska bića ne predstavljaju poslednju reč u evoluciji na Zemlji, već samo izvornu lozu iz koje će se razviti jedan novi i viši oblik života, da bi se prevažila dosegnača australopiteka. Istorija života potvrđuje taj zaključak, jer pokazuju neumoljiv trend ka većoj inteligenciji kod viših oblika životinja. Očigledno, među svim odlikama nekog živog organizma, nijedna neće veću vrednost za opstanak od fleksibilnog, inovacijskog reagovanja na izmenjene okolnosti, reagovanja koje nazivamo inteligencijom. Izgleda neverovatno da će taj trend u evoluciji biti iznenadu okončan na jednom određenom nivou mentalnog doseganja koji nazivamo ljudskim. Ako je prošlost neki putokaz za budućnost, čovečanstvu je, izgleda, sudeno da dobije još inteligentnijeg naslednika.

Intelektualac

Kakav će izgled imati naš naslednik? Sudeći po skorašnjoj čovekovoj istoriji, novi oblik života ličiće na stari, ali će imati još veći mozak. Prema tom predviđanju, slediće vrsta inteligentnog života na Zemlji biće intelektualac, stvorenje slično nama ali sa većom lobanjom i slabijim mišićima.

Međutim, izvesni trendovi u modernoj tehnologiji sugeriraju sasvim drukčije videnje budućnosti. U dejstvu su moćne sile evolucije — prevažodno kulturne i ne biološke — koje mogu da dovedu do jednog egzotičnijeg oblika inteligentnog života — oblika koji će se razviti iz čoveka, ali pre kao da njegovog mozga nego kao da njegove slabine.

Prema tom videnju, novi oblik života se danas već stvara u laboratoriji kompjuter-skog naučnika. To je jedan veštacički život načinjen od silicijumskih čipova a ne iz neurona; a ipak, on misli, pamti, uči se na iskustvu i reaguje na nadražaje. Njegovo razmišljanje je još uvek priprsto; nije mnogo kreativno; ali evoluiru zapanjujućom brzinom.

Ideja izgleda apsurdna: kako može bogatstvo ljudske misli da bude upoređivano sa mehaničkim razmišljanjem jednog kompjutera? Pa ipak, iznenadujuće stvar kod kompjutera nije u tome što oni misle

manje dobro od čoveka, nego što uopšte misle. Tačno je da su elektronski mozgovi današnje primitivniji u poređenju sa ljudskim mozgom; u stvari, oni nam daju malo razloga da im se divimo, sem zbog svoje ogromne memorije i izvesnih matematičkih veština. Medutim, najnoviji modeli mogu biti uprogramirani tako da prate neku raspravu, postavljaju drčna pitanja, ili da pišu dopadljivu poeziju i muziku. Isto tako, oni mogu da obavljaju ponetu neobavezne konverzacije, toliko ubedljivo da njegovi ljudski partneri nisu ni svesni da razgovaraju sa mašinom.

Silicijumski mozak

To su dopadljivi kvaliteti kompjutera; on imitira život kao neki elektronski majmun. Kompjuter postaje kompleksniji, i imitacija postaje bolja. Na kraju, crta koja razdvaja original od kopije postaje zamaljena. Kroz sledećih petnaest godina — od 1995. prema tekucim trendovima — možemo videti kako se silicijumski mozak javlja kao jedan novi oblik života, koji može da se takmiči sa čovekom.

Pre nego što odbacimo tu viziјu, razmotrimo neka skorašnja dosegnača u kompjuterskoj industriji. Prvu istinski modernu generaciju kompjutera reprezentovala je mašina IBM 360 koja je puštena na tržište 1964. godine. Jedna mašina tog tipa obavljala je posao za Godardov institut za proučavanje svemira tokom poslednjih 15 godina. Koliko bi velika bila takva mašina ako bi je trebalo proširiti koliko da odgovara kapacitetu ljudskog mozga?

Mozak je težak približno 1.400 grama, troši električnu snagu otprilike 20 vata, i zauzima zapremenu tridesetog delu kubnog metra. Unutar tog malog prostora su sadrži memoriju od 10 milijardi informacionih jedinica. Jedan kompjuter IBM 360 tog istog kapaciteta trošio bi električnu energiju od milijardu vata, ispunjavajući čitav Svetiški trgovacki centar, i koštao bi mnogo milijardi dolara. Bila bi to nezgrapna imitacija možga.

Ali najnovije mašine na tržištu 1980. — samo 16 godina kasnije, pričaju drukčiju priču. Jedna od tih novih mašina, proširena toliko da odgovara kapacitetu mozga, mogla bi se smestiti u prostorije neke kancelarije, i trošila bi samo 100.000 vata električne energije. Bila bi još uvek nezgrapna, ali gotovo vredna izgradnje.

Portabl kompjuter

Dala poboljšanja, koja se sada odvijaju u laboratorijima za kompjutere, biće spremljena za tržište oko 1990. godine. U to vreme možde će se izgraditi kompjuter čovečjeg kapaciteta, koji bi mogao da se smesti u kofer i da operise pre utrošku električne energije od 1.000 vata.

Oko 1995. krivulja rasta silicijumskog mozga mogla bi da prede crtu ljudskih sposobnosti — 10 milijardi činjenica u jednoj aktenčasti, sa utroškom od svega 20 vata električne energije.

Naravno, silicijumskom mozgu koji odgovara ljudskom mozgu samo po smestaju način kapacitetu i moći grubog rasudivanja nedostajuće kreativna inspiracija i intuicija koje karakterišu više domena ljudske misli. Poput nekog novopovećenog doktora filozofije, on će imati ogromne mentalne moći, ali ne mudrost. Pa ipak, i tu kvalitativnu superiornost mozga nad kompjutrom sve

više podrivaju izvesna novija razvojna kretanja. Da bi se shvatio značaj tih kretanja, razmotrimo za časak arhitekturu ljudskog mozga. Svaka moždانا celija je direktno povezana sa mnogim drugim moždanim celijama, u nekim slučajevima čak sa 100.000. Kao rezultat toga, kad god posalješ neki svesni impuls dole u dobine naše memorije da prizovemo neku određenu informaciju, celije u kojoj je ta informacija pohranjena komuniciraju na podsvesnom nivou sa hiljadama drugih celija, i jedan talas mentalne aktivnosti prostruji kroz čitav mozak. Plodovi te podsvesne aktivnosti su intuitivna spoznaja, blesak percepcije i nadahnuta genija, što je sve omogućeno zahvaljujući bezbrojnim vezama među celijama ljudskog mozga. Te veze među celijama objašnjava mnogo šta od neobičnih možga.

Arhitektura kompjutera je sasvim drukčija. Kompjuterske memorije su kao jedna garnitura otvora na golubarniku poredanih u zid, bez povezanosti jednog otvora sa drugim. Informacija može da bude stavljenja u golubiju otvor ili izvadenja iz nje, ali golubiju otvori ne komuniciraju međusobno, niti razmisljavaju. Razmišljanje se obavlja na drugom mestu.

Integrisana kola

Najnoviji razvoj koji menja arhitekturu kompjutera baziran je na izumu koji nazivamo čipovima. U svom najnovijem obliku čipovi nose naziv integrisana kola. To su tanci četvrtastih pličice silicijuma, duge otprilike šesdeset milimetara po jednoj strani, koju su povezane međusobno provodnicima u velikom broju; one formiraju misleće i memorije sekcije elektronskog mozga. U starijim danima, oko 1960. ili 1970. svaki silicijumski čip mogao je da drži samo malo broj tranzistora i elektronskih elemenata. Rezultat toga bio je da je jedan čip mogao da liši, ili da pamti, ali ne oboje. Danas, međutim, vodeći proizvođači integrisanih kola u Sjedinjenim Državama i Japanu naučili su kako da natrpaju 100.000 tranzistora i elektronskih elemenata na jedan jedini čip. Sada jedan čip može i da misli i da pamti. Ako je svaki čip povezan sa mnogim drugim čipovima, kompjuter postaje pravi silicijumski mozak; popul ljudskog mozga, na funkcioniše kao jedna jedinica, obavljaju misaone procese u talasima unutrašnje mentalne aktivnosti na „podsvesnom“ nivou. Ovde se već približavamo onoj vrsti asocijativnog mišljenja koje predstavlja tako moćan element u ljudskom rasudivanju.

To su izgledi koji možemo predviđati za poslednju deceniju našeg veka. U to vreme silicijumski mozgovi će u saradnji sa našim najboljim umovima raditi na svim ozbiljnim problemima dana, u jednoj nepodbeboj kombinaciji moći sirovog rasudivanja i ljudske intuicije. A što će se zatim desiti? Šta u sledećem stoljeću? I šta u stoljeću iza toga? Ne mogu se predviđati granice rastuće krivulje silicijumske inteligencije. Jedan od vodećih ljudi u kompjuterskom istraživanju, profesor Marvin Minski sa MIT-a, veruje da se moći ostvariti mašina sa „opštom inteligencijom jednog prosečnog ljudskog bića... mašina će početi da obrazuje samu sebe... kroz nekoliko meseci će biti na nivou genija... a nekoliko meseci posle toga njena će moć biti neizračunljiva“. A onda, kaže Minski: „Ako nam sareća bude nakanjena, one će možda odlučiti da nas drže kao kućne ljubimice“.

era tehnoloških genija

Mikroelektronska revolucija, koja je, sa prilično zakašnjenja, najzad zaplijesnula i naše obale i za sobom povukla na hiljade mališana, zatekla je roditelje i pedagoge potpuno nespremne. Šta se to najednom dešava sa mladima? Da li su se oni naprečac okrenuli „korisnim“ stvarima ili u njihov novi hobi imaju svojih skrivenih zamki i opasnosti? Koje je stvarno mesto računara u obrazovanju, a koje u životu? Šta očekuje one koji se profesionalno opredeli za računarstvo, a šta one koji ostanu „kompjuterski nepismeni“? Ove stranice, koje prenosimo iz knjige „Računari i vaše dete“ engleskog sociologa Reja Hemonda (Ray Hammond), treba da pomognu (zbunjениm) roditeljima da bolje shvate odnos dece prema računarima i pravilnije usmere njihov razvoj.



Na muškoj teritoriji: Interesovanje za računare je strogo podvojeno na polove — ne zato što su devojčice za njih „glupe“ a dečaci „pametni“ nego isključivo iz socijalnih razloga

dete treba da „razume kompjuter“, ali ni dve osobe, izgleda, ne mogu da se slože kako da se to ostvari ili u čemu bi se to sastojalo.

Greška u koracima

Nezaposlenost je posebno ostra među onima koji završavaju školu. U nastojanju da mladima otvori veće mogućnosti zapošljavanja, većina srednjih škola pruža sada neki oblik kompjuterskog obrazovanja. To se najčešće radi tako da se deci predaje programiranje u okviru nekog kursa „kompjuterskih studija“. Na ovim kursevima najčešće se predaje bežik, ali mnogi nastavnici već razmišljaju i o alternativnim kompjuterskim jezicima, kao što su komal i algol.

Deca mlađa od 12 godina, u stvari, nisu u stanju da rade ni sa jednim od ovih apstraktnih algoritamskih jezika, i u mnogim srednjim školama obrazovanje u računarima ne počinje pre nego što dete ide u treću godinu (osmi ili deveti razred).

Postoji mišljenje da je pogrešno uvoditi mlađe u svet računara tako što će im se predavati programiranje. To je, smatra se, da se onome ko treba da nauči da vozi avion govorii o tome kako je on konstruisan. Odvajajući računarstvo u poseban studij i tražeći od učenika da dolazi u računarsku sobu (često lociranu u matematičkom kabinetu), škole odvajaju računarstvo od drugih „običnih“ predmeta i ističu njegovu matematičku ili naučnu osnovu. Ta se situacija često uporno odražava, jer je nastavnik matematike ili egzaktnih nauka prvi uveo računarstvo u školu, pa nastoji da zadrži ličnu vlast nad svojom „teritorijom“.

U nekim školama, računarski predmeti se slabo probijaju u osnovni nastavni program, pa se računarstvo još uvek predaje kao grana nauke. Po jednom interesantnom tumačenju, računarska nauka je drugim predmetima, kao što su ekologija i fizičko obrazovanje, izbačena iz nastavnih programa jer su ambiciozni nastavnici pažnju usmerili na predmete koji im donose više prihoda i daju više vlasti u njihovim institucijama. Postojeći nastavni programi u srednjim školama „ispitno su usmereni“, sposobnost kako učenika tako i nastavnika određuje se samo uspehom na ispitima. Ugleđ, unapređenja i novac idu nastavnicima koji predaju teorijske predmete koji se najlakše mogu uključiti u ispitni format. Izgleda da ima doista dokaza u prilog ovog mišljenja: kao neposredno posledica ovog i drugih prilisa, u mnogim srednjim školama računar ostaje zaključan u matematičkom ili naučnom kabinetu.

Rezervat za dečake

Ovo odvajanje računara ima i drugu tešku posledicu. Među decom između 11 i 18 godina, računari i računarstvo smatraju se prvenstveno „muškim“ predmetom. To je, delimično, stoga što se računar često čuva u „muškim“ odjeljenjima — naučnim ili matematičkim — pa je njegovo korišćenje još uvek u vlasti muškaraca, a delom zbog načina na koji se računar koristi u nastavi.

Razlike u intelektualnom razvoju polova dugo su predmet istraživanja i suprotnih gledišta; postoje neki površni dokazi da su u toku mlađalačkih godina dečaci u matematici uspešniji od devojčica. Ovakvi rezultati se moraju primati s rezervom, jer mogu biti

posledica načina na koji se devojčicama predaje matematika. Uticajna studija britanske vlade „Socijalni trendovi“, koja se objavljuje svake godine, pokazuje da su devojčice u celini uzev, uspešnije od dečaka, da 32% devojčica koje završe srednju školu odlaži na dalje školovanje, prema 22% kod dečaka. U najboljim grupama, međutim, dečaka je nešto više od devojčica.

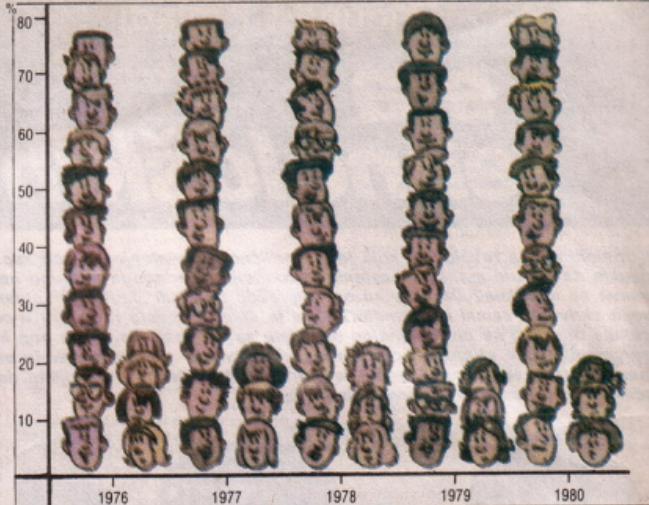
One što je, međutim, jasno, jeste da računari pripadaju dečacima — bar u ovom trenutku. Zahtevi tinejdžera (od kojih su mnogi ludi za vaskonskim igrama) priskrivaju mnoge porodice da nabave računare. Proizvođači su svesni ogromne snage tog tržišta i koriste je, podstičući postojće trendove. U ranom mlađačkom dobu, seksualne uloge su polarizovane. Kada dečaci i devojčice počnu da otkrivaju svoju seksualnost, oni se povlače skup u svoj ugao kako bi razmotrili situaciju. Na tom tržištu, oglaši i izložbe kućnih računara uglavnom počivaju na igrama koje se nepogrešivo odnose na divlje ratove u kosmosu i herojska dela — sva muška po pristupu. Proizvođači se ne usuđuju da tinejdžerima njihovo interesovanje za računare prikažu kao „detinjstvo“ — radi se o grupi nedovoljno odrasloj da spozna i kontroliše svoj divlji separatistički šovinizam. Tako se računari svrstava u kategoriju kartinga i fudbala, a devojčice koje nemaju sreće da u njemu otkriju ono što ih interesuje, celu stvar tako i klasificuju.

Kućni računari se masovno kupuju na tinejdžerskom tržištu. Pre dve decenije tako je bilo sa električnim gitaram. I kada štav se šezdesetih godina roditelji nadali da će njihovo dete napredovati i stići do „pravog“ instrumenta, tako se današnji roditelji nadaju da će „osvajači kosmosa“ dovesti do konstruktivnijeg pristupa primeni računara. Verovatno je da će oni u tom svom nadanju imati više sreće nego što su je imali njihovi roditelji.

Akcijske igre su izvanredna atrakcija za dečake (mada ih vole i mnoge devojčice). Pravilno usmereni, dečaci i devojčice će napredovati ka kreativnijem korišćenju računara. Izgleda da neka deca postaju veoma vezana za „pokmena“ ili njegov ekvivalent, ali iako se često koristi reč „zavisan“ (kao u slučaju droge), nema dokaza da dolazi do tako ozbiljnog stanja.

Veliki igrač

Veoma se mnogo istražuju motivi koji navode mlade da troše novac i slobodno vreme na akcijske igre. Kod pakmena, na primer, svaka uništena tačka predstavlja dostignute, trenutni uspeh, a ako naša igra nije vrhunска, protivnik — mašina — nije u mogućnosti da nam se podsmeju. Igre daju učešnicima mogućnost da se istaknu i veoma veštii igrači mogu osvojiti zavidan „igracički ugled“. Da bi se dalje shvatila privlačnost akcijskih igara, treba razmotriti šta one stvarno daju. Jasno je da igrači bivaju potpuno apsorbovani — osećanje koje bi se teško moglo ostvariti u mnogim drugim aktivnostima. O toj apsorbovanosti se govori kao o posebnom duševnom stanju u kome se osoba koncentriše na trenutku, zadatak, isključujući sve druge unutarnje ili spoljne stimulante. Akcija i svest se



Status quo: Devojčice u Engleskoj se znatno redje odlučuju za računarske studije od dečaka i taj odnos se već godinama ne menja

stapaju tako da osoba prosti radi ono što se mora uraditi, bez kritičke, dvojne perspektive sopstvenih akcija. Nastoji se da ciljevi budu jasni, sredstva se koordiniraju sa ciljevima, i povratna spreka sa akcijom je neposredna i jednoznačna. U takvoj situaciji osoba ima jak osećaj kontrole — ili lične motivisanosti — pa ipak, što je parodatsko, uključivanje ega je slabu ili nikakvo, tako da se doživljjava osećaj sopstvene transcendencije, ponekad osećaj jedinstva sa okolinom. Izgleda kao da je proticanje vremena poremećeno: neki dogadjaji, kao da traju nesaznerno dugo, ali časovi, uposte uzev, prolaze u minutima.

Najbolje poređenje stvarnog života sa doživljajem akcijske igre verovatno je borba ili izvršenje nekog izuzetno hrabrog dela. Duh i telo su stoljeni u jedinstveni „veličanstveni“ cilj, liseni svega drugog. Radost koja se doživljava, zajednička je simularnim i stvarnim akcijama. Beskonačno vežbanje i dresiranje vojnika upravljeno je ka postizanju tog „stapanja svesti“ u trenutku kad se suoči sa neprijateljem. Još nije utvrđeno da li to zamenjuje potrebu mladih da nasilničkim delovanjem.

Programiranje na stramputici

Kada su se pojavili 1975. godine, kućni računari (i veći deo školskih mikroračunara) mogli su da se uporede sa kristalnim detektorima koje su u ranijim generacijama konstruisali mnogi (ali vrlo malo) devojčići. To je bio hobi u kome su teško konstruisanja bile sastavni deo privlačnosti. Ali računarski jezici su se usavršavali sa usavršavanjem računara i verovatno je u vrlo bliskoj budućnosti nećemo morati razumeti računarski jezik da bismo mogli programirati računare — kao što danas ne moramo biti u stanju da podesimo oscilator

da bismo mogli da izaberemo radio-stanicu. Za nekoliko godina, računarski jezici, kao što su bežik, algol, fortran i kobil biće zastareli za normalno programiranje i biće samo od akademskog značaja. Sa porastom kapaciteta računara i inteligencije softvera, detaljno programiranje će postajati sve manje nužno. Već postoje programi, kao što je „Poslednji“, koji sam pišu programne za korisnika, i nagoštavajući vreme kada nam neće biti potreban nijedan kompjuterski jezik, lako je to obećanje u ovom trenutku pomoći optimističko, jasno je da su dan „peeking“ i „poking“ prošli. (Mislim da te dve reči iz programerskog žargona sumiraju sve što je najgorje u mikroračunarenu. One opisuju postupke programiranja koji su već pedesetih godina smatrani primitivnim, ali za kojih i današnji poklonici „kristalnih detektora“ pokazuju najveće interesovanje.) Ono što će biti potrebno, jeste potpuno razumevanje načina na koji radi kompjuterska logika. Bez toga se neće moći koristiti programi kao što je „Poslednji“, koji samo automatizuju pisanje programa.

Uprkos predočoj pojavi računara koji će sami sebe programirati, škole nastavljaju da predaju programiranje, odvajajući računarsku logiku, ali mučenje nad bilo kojim posebnim računarskim jezikom prevaziđeno je i treba ga što pre napustiti. Samo neznačatan broj deca nastavice školovanje sa ciljem da se obrazuju kao profesionalni računarski programeri, kada jednom dodu na univerzitet ili na neki profesionalni kurs, oni će obaviti ciju svu veštinsku u bežiku ili nekom drugom višem jeziku i počeće da uče kako se sa mikroprocesorom direktno saobraća korisnjem mašinskog koda ili asemblerškog jezika. Temeljno poznavanje računarske logike biće ovim dacoma korisnije od učenja nekog posebnog višeg programskog jezika.

Fanatici koda

Nekoj deci, posebno inteligentnijoj, programiranje računara može postati neka

vrsta droge, čak mnogo više nego akcione igre koje su ih, možda, prve privukle računaru. Često socijalno nedorasti, oni fanatici koda razvijaju sklonost ka računarskoj inteligenciji, hrabreni roditeljima koji smatraju da je njihovo inteligentno dete na putu da postane tehnološka zvezda: oni se povlače iz realnog sveta u svet spačave slobodnog, nezgramnih zastarelim računarskim jezicima. Ta situacija postaje nezdrava kad dete počne da piše programe u potpunosti za sopstveno zadovoljstvo. Privatni svet spačave slobodnog postaje svetlost u kome je dete bog, a računar pruža detetu i roditeljima opravdavanje za potpuno iracionalno i štetno ponasanje.

Nepoznavanje računarske materije često dovodi do situacije u kojoj se deca iskreno, ali pogrešno, smatraju zvezdama. Nastavnici ne mogu da idu u korak sa detetom kada se ono dublje upusti u mogućnosti veštacke inteligencije i potpuno su nesposobni da procene stvarnu vrednost dečjih dostignuća. Često dolazi do velikih zloupotreba programiranja i dete nije u stanju ili nije voljno da priveli nikakav praktičan savet u vezi sa svojim radom na programiranju. Za trenutak dete impresionira svoje vršnjake, roditelje i nastavnike složenim programima sa bleštavim ekranima i zvučnim efektima. U stvari, dete je pobeglo iz realnog sveta i navikava se na programiranje u potpunom akademskom vakuumu.

Pod pretpostavkom da ovi daci ostanu dovoljno konformisti da bi zadovoljili uslove prijema, univerziteti obično moraju da ih reprogramiraju pre nego što ih usmere na studij kompjutera.

Postoji mišljenje da nabavka kućnog računara može ponekad da ima loš efekt na učenje mladog čoveka i čak na izglede za njegovo zapošljavanje. Prema tom mišljenju, davanje računara mladima u osjetljivom periodu njihovog života, u nadi da se time postavlja osnova za uzbudljivu karijeru, može, u stvari, imati suprotni efekat. Da bi na računarama radili koji treba, programeri moraju da misle na jeziku koji koriste, kao što je i za tečan francuski ili nemacki jezik potrebno misliti na tim jezicima. Većina jezika koji su danas dostupni za kućne računare zastarela je i pretrpana svim i svatim. Ako se mlada osoba duboko uživi u njih, bice joj otežano učenje novog jezika.

Jedna mogućnost da se ta leškoča izbegne, jeste proširiti pristup računarama. Deca pre treba da uče šta oni mogu da urade, nego kako da ih programiraju. To znači, kad je reč o kući, da roditelji treba da obezbede što svestranije korišćenje računara. Sto se škole tice, računar treba izvući iz muškog naučnog odjeljenja i uključiti ga u ceo nastavni program, tako da bude od koristi svim predmetima. On mora postati učilo koje će se normalno koristiti, ravno pravno sa globalnom i videorederima.

Nastavnici računarskih predmeta retko stvarno poznavaju predmet koji predaju. To je javno izjavila jedna britanska komisija koja je našla da su nedavni ispit na kompjuterskim studijama A-nivoa otkrili zaprepašćujući nedostatak razumevanja kod učenika kako se pripremiti za ispitne programe i kako se obezbediti od kvara hardvera.

Računarske sveznalice

Jedna britanska studija je nastojala da ustanovi od kolike je važnosti kompjutersko predznanje kada učenik oda na univerzitet ili stupi u industriju. Istraživači su prikupili mišljenja iz obe oblasti i otkrili da akademiske obrazovne institucije više voli da imaju dake sa zdravom osnovom u matematici i fizici nego u kompjuterskim naukama, jer su daci sa računarskom osnovom verovatno usvojili i druge navike. Industrija, sa svoje strane, malo značaja pridaje kvalifikaciju u kompjuterskim naukama. Računarstvo se, uz to, tako brzo menja da prikupljeno znanje zastari pre nego što daci dođe na diplome.

Decu opterećuju složeni problemi: ako prihvativamo da je učenje programiranja pregrešan pristup, kako prekvalifikovati naše nastavnike da daju šire, opštije obrazovanje zasnovano na računaru, u kojem bi mašina bila sredstvo a ne cilj. Svaka nasa da se nešto brzo može promeniti, nije opravdani. Međutim, ako slegnemo ramenima i odlučimo da iz lošeg posla izvučemo najviše što možemo prihvatanjem „kompjuterskih studija“, suočeni smo sa situacijom da nastavnici ne razumeju predmet dovoljno da bi ga mogli uspešno predavati. Kako onda možemo da pomognemo deci?

Ponovno se odgovor mora naći u kući. Roditelji koji su svesni opasnosti izolacije mladih mogu da urade mnogo da spreče pretvaranje svog deteta u fanatika koda. Oni mogu nastojati i na svestranijem odnosu svog deteta prema računaru, iznalažeći primene računara van školskog nastavnog programa. Pri tome se mogu suočiti sa teškim zadatkom. Programiranje je u mnogim školama postalo snobovski predmet, i dečaci koji se pripremaju za upis na univerzitet, sa kompjuterskim naukama visoko na svojoj listi akademskih dostignuća, ubrajaju korišćenje računara (nasanrop nijehovom programiranju) u „glupavoj“ računarenje. Ovaj tip omalovlažavanja podmukao je, i teško je izaci s njim na kraj. Ipak, dok je dete zadovoljan što koristi „glupavoj“ računarenje, može biti u boljem položaju da shvati ulogu koju će veštacka inteligencija imati kasnije u životu.

Mladi mozgovi

Pošto su istaknute opasnosti od podučavanja dece programiranju računara, treba reći i to da stvaralačka maštta i sposobnost učenja, aka su usmereni kako treba, mogu kod tinejdžera da izbace neke odlične programere koji su u stanju da nizno zaraduju i po nego što završe školu. Mnoge zemlje imaju sada neku vrstu takmičenja „mladih računarskih mozgov godine“, kada se stvarno sposobniji deci dešavaju da pokažu što mogu. Pobednici na ovim takmičenjima usmerili su se bez izuzetka na određenim primenama računara i dobijaju nagrade jer su razvili programe koji pomažu hendikepiranim osobama, olakšavaju otkrivanje kriminalaca ili su od neke druge direktnote koristi za društvo. Neka decu tako vešta u pisanju programa (najčešće igara) koji privlače drugu decu, da proizvedu mikroračunara kupuju njihove programe po visokoj ceni, odobravajući im i određeni procenat od prodaje. Poznati su mnogi slučajevi 15-godišnjaka koji su zarađali mnogo novaca u svom slobodnom vremenu.

Ta obdarena deca uspela su, međutim, da postanu „uravnoteženi“ programeri, sposobna da sagledaju potrebu društva za

specifičnim tipom programa. Ta je sposobnost isto tako značajna, ako ne i značajnija, od ciste sposobnosti da se pišu računarski programi. Ta deca mogu biti potpuno apsorbovana u programiranju, ali je malo verovatno da bi tu aktivnost koristila kao psihološki oslonac ili kao odbranu od spoljnih svela.

Mnogi dečaci i devojčice sa zdravim pristupom programiranju, završavaju školu petkom i, namesto da se subotom angažuju u lokalnoj prodavnicu poput njihovih roditelja, opredeljuju se za lokalnu softver kompaniju (ili svoju spačavu sobu) da bi nastavili sa razvojem određenog komercijalnog projekta. Treba naglasiti da ta deca predstavljaju neznačan manjinu i obično spadaju u kategoriju „obdarene dece“. Njihov uspeh počiva na sposobnosti da otkriju takvu primenu računara koja će ili naći tržište ili na neki način pomoći društvu. Njihova sposobnost da stvaraju programe manje je važna i malo je verovatno da će se napredovanjem u karijeri smanjiti raditi na fizičkom programiranju. Verovatno je da će oni raditi u tri faze:

1. identifikaciju određenog problema koji traži rešenje
2. analizi problema
3. skiciranju programa koji će dati rešenje.

Stvarni detaljan rad na razvoju program biće prepušten drugim, manje kreativnim programerima.

Zbog ovoga, za većinu dece računarstvo treba da bude sastavni deo njihovog obrazovanja; na kraju tog obrazovanja, obrazovanje računara za pomoć u istorijskom istraživanju ne bi predstavljalo ništa više od odlaska u biblioteku i uzimanja knjige sa police. Deca bi bila upućena u to kako računari rade, šta mašine mogu a šta ne mogu, kao i u najbolji način korišćenja raznih tipova hardvera. To bi bilo idealno stanje i do njega će se vremenom doći. Problem je jedino ako važe dele treba da završi školu već za sledeće dve ili tri godine.

Mnoge škole koriste računare samo u nastavi matematike i nauka. Mnogo rede se oni sreću u nastavi engleskog i francuskog jezika, geografije ili istorije, da i ne govorimo o muzici i umetnosti. Postoje za to dva glavna razloga: nedostatak sredstava i nedostatak nastavnike stručnosti. Sa padom cene mikroračunara, nedostatak stručnosti pokazuje se kao najteži problem koji treba savladati.

Računari i nauka

Dobre strane korišćenja računara u nastavi su očigledne. Ispitivanje životnog ciklusa stanovnika nekog jezera predmet je koji se lako uključuje u računarski program. Program Edel Lake koji je razvila MECC (Minnesota Educational Computing Consortium), jedna od najboljih obrazovnih računarskih organizacija u svetu, omogućuje daci da stvore model jezera i ispituju ih ishranom lanac u njemu. Program je deo jednog šireg programa pod nazivom „Razumevanje lanca ishrane“; svaki korisnik može da planira svoje sopstveno jezero i da uvedi razne oblike životinja u njega. Program trenutno razaznaje koji su oblici života inkompatibilni — na primer, pastrmica i bela riba — i o tome obaveštava učenika. U nastojanju da stvoru unavrežnije jezere, učenik će morati da prihvati neke gubitke u kolicićima bele ribe i da razmotri načine kako da reguliše broj snažnijih grabiljivica.

Ova vrsta interakcionog programa sti- muliše učenika da traži sve više informacija o ulozi raznih oblika života u vodenoj sredini i, u zavisnosti od obimnosti programa, treba da je sposoban da primi mnogo informacija direktno iz računara. Ovaj tip programa pokazao se odličnim u osvajanju i držanju pažnje učenika i praktički osloba- da učenika potrebe za nastavnikom dok stvara svoj akvatični svet. Na kraju projekta, učenik stvara više ili manje uravnoteženo jezero, sa svim važnim informacijama o vodenici i zapremini, biljnom, ribljem i životinjskom svetu dobijenim u štampanom obliku.

Programi, kao što je *Quake*, koji takođe potiču sa MECC, omogućuju učeniku da igra ulogu seismologa i pomažu mu da odredi epicentar zemljotresa posle sekun- darmog udarnog talasa. Programi se osla- njuju na raznu dokumentaciju i sastavljeni su tako da utiču i na bogaćenje dječkog rečnika i bolje razumevanje geometrije i srodnih nauka.

Nastava geografije se danas dobrim de- lom bavi ljudima koji žive u drugim krajevi- ma sveta. Postoje mnogi programi koji pomažu dječima da učebežavaju u mape kretanje stanovništva i određene socijalne i ekonomski promene, kao što je industrijski razvoj. Sa visokokvalitetnim grafičkim dis- plejem, daci mogu da prikazuju razne dešće Afrike, na primer, i da ih dopune sencirom u boji o padavinama, žetvenima prinosima, gustum stanovništva itd. Pri ruci je računar koji omogućuje da se učenici u svakom trenutku prebacuju sa mape na informacioni displej i tako dove do što tačnije predstave o jednoj oblasti. Tako moći programi omogućuju da se izbrišu granice koje je obrazovanje vještacki postavio između predmeta. Dok ispituje topografiju jedne oblasti u Zimbabveu, učenik će moći, ako želi, da doda i sencirom koje će pokazivali industrijsku proizvodnju, prosečan dohodak, zdravstvenu statistiku ili geološke in- formacije.

Računari i matematika

Daci su u stanju da najpre u matematiči stvaraju modelle problema. Matematika je možda jedna od najapstraktnijih disciplina i pruža velike mogućnosti za dobijanje grafičkih i matematičkih modela. U svojoj aritmetičkoj ulozi, računar je izuzetno koris- tan kad odgovara na pitanje „šta ako?”, jer otkriva interakcije različitih matematičkih događaja. Koristeći popularni komercijalni program kao što je „Visical”, da- može da osnuje imaginarni biznis upotpunj- en sa svim fiksnim i proizvodnim troško- vima i tome doda podatke o prodaji sa odgovarajućim izdaciama. Pošto je taj moći program potpuno interaktivni, dok samo treba da menjaj jedan faktor — broj artikala prodatih za mesec dana, na primer — da bi saznao efekt koji bi takva jedna promena u poslu imala na zaradu i na finije elemente, kao što je ideo fiksnih troškova koji otpada na amortizaciju vozila. Moguće je ovakve modelle sačiniti i ručno, ali svaki bi zahteva- vao mnogo časova rada. Računarski program omogućuje daku da se beskonačno „igrat” sa modelom svog biznisa i tako stiže temeljito razumevanje vitalnih elemenata koji zajedno proizvode dobit — ili gubitak.



U svetu mašte: Igre na računarama, koje su, po pravilu, često veoma agresivne, pokazati zamjenjuju potrebu mladih za nasilničkim delovanjem

Instinktivni osećaj za biznis koji se ovim dobija posle nekoliko nedelja igranja, često je mnogo sigurniji od instinkta koji imaju mnogi profesionalni biznismeni koji nisu imali mogućnosti da izučavaju modelle na bazi računara.

Računarske nauke kao odvojeni pred- met sada su sasvim jasno definisane u većini škola i pokrivaju oblasti kao što su razumevanje elektronike, programiranje i primena. Vrlo malo škola (za razliku od koledža u daljem obrazovanju) bavi se uti- cajem računara na društvo.

U drugim naukama — hemiji i fizici, na primer — računari su dobro uvedeni i dostupni su mnogi provereni programi. Van oblasti nauka stvar stoji drukčije.

Britanska organizacija MUSE (Micro- computer Users in Secondary Education — Korisnici mikroračunara u srednjem obrazovanju) trenutno ima spisak od 120 pro- gramova namenjenih školama, od kojih je više od 100 orijentisano ka matematički i naučna- ma. Postoje mnogi programi za umetničke predmete za osnovce, ali malo broj za decu između 11 i 18 godina.

Muzika i umetnost predmeti su u kojima mikroračunari imaju izuzetno značajnu i posebnu ulogu. Ipak, uprkos mnogim danas dostupnim pogodnostima, mali broj škola koristi računare u tim oblastima.

Istraživački projekti

Sake godine firma Commodore prikazuje kako njeni Pet mikroračunari kontroli- šu jedno teško putovanje dece kroz pod- mukle Jorkširske močvare u severnoj En-

gleskoj. Uz nešto nadzora, deca unose u program potrebne podatke: rutu, stanje okoline, broj i imena učesnika itd. a na komandnom mestu se, zahvaljujući računu- ru, veoma pažljivo prati kretanje. To je idealan tip primene koji pokazuje da uloga računara nije ograničena na jedan ugao naučne laboratorije.

Jedna škola u San Dijegu, Kalifornija, beleži na mikroračunaru godišnji tok migra- cije kitova, tako se ovo može smatrati čisto naučnom primenom, ona je obuhva- ta ništa manje nego 150 daka, koji su se jednog dana, posle mnogo uloženog truda, ukrcaju na brod da bi isplivili u Pacifik i na licu mesta videli da li njihova računarska simulacija ove najduže migracije sisara u svetu, olakšava pronađenje kitova. Šteta je da veći broj škola ne može da priuštiti ovaj tip stvaralaštva u rešavanju problema po- motu računara.

Ali računarstvo se brzo razvija i, mada je situacija trenutno daleko od zadovoljavaju- ce, za nekoliko godina mogla bi se ipak izmeniti tako da će deca koja uče stolarski zanat smatrati prirodnim da se obrate raču- naru stolarske radnje da programira ruku robota da delove koji se ponavljaju.

Šta je, međutim, sa decom koja završi- vaju školu? Neka su možda naučila nešto bez ikiga, druga su možda predmet pratila detaljnije, dok je najviše njih verovatno gurnuto u grupu koja je imala da sluša usmena objašnjenja i radi praktički nekoliko minuta u nedelji.

Oni koji završavaju školu ozbiljno se interesujući za računare, mogli bi nastaviti da predmet izučavaju na kursu u nekom

koledžu ili da se direktno zaposle na računaru. Mnogi koledži i univerziteti nude kurseve sa računarskom kvalifikacijom na kraju. Studenti koji sa uspehom završe takav kurs, verovatno će biti u stanju da u mnoštvu ponuda izabere ono što im najviše odgovara. Bilo je posljednji godine nekih diskusija u novinama o nezaposlenosti koja pogodača računarsku industriju (koja se sama često smatra prvenstvenim uzrokom nezaposlenosti). Radi se, međutim, o računarskom osoblju nizih kvalifikacija — kao što su programeri i operatori — dok se nastavlja (i verovatno će se nastaviti) jaka potražnja za inženjerima, sistemskim analitičarima i projektantima.

Interesantna mogućnost koja upravo postaje realna, a pruža se studentu računarskog, jeste da se vrati u školu i predaje računstvo. Do sada je ono kao specifičan predmet bilo rezervisano za univerzitete i koledže. Problemi je u tome da nastavnici treba da imaju daleko širi pogled na računarstvo od ograničenog pogleda razvijenog pri radu u računarskoj industriji. Jasno je da je obrazovanje očajno oskudno u obrazovanom kompjuterskom osoblju, i takva karijera bi se nešumljivo veoma isplatala.

Poslovi u računarstvu

U računarskoj industriji postoji nekoliko nivoa hijerarhije i, mada je moguće napredovati u sistemu, oni koji dolaze sa univerziteta ili sličnih institucija, mogu očekivati da će napredovati brže.

Početni stepen glavnih hijerarhijskih nivoa u kompjuterskoj industriji. Najniži stepen mogao bi se nazvati „kvalifikovani korisnik“. Ta kategorija obuhvata radnike koji su naučili da na računarama rade specijalne poslove, kao što su obrada reči ili knjigovodstvo. Često se ovaj kader regrutuje iz drugih zanimanja — na primer, sekretarijata ili kancelarijske administracije — ali on uključuje i funkcije specifične za računarsku industriju, kao što su operater na terminalu, operatör na bušaću kartica i slično. Angažovanje na tim poslovinama zahteva osnovne školske kvalifikacije i sposobnosti jasnog mišljenja. Znanja, kao što je rad na tastaturi, normalno se uče na samom poslu.

Sledeći viši stepen jeste operater. Operatori kontrolisu rad na računaru, bilo na trakama ili na magnetnim diskovima. Operatori moraju razumeti osnovu na kojoj počiva rad računara i taj tip posla često podrazumeva rad u smeri, pošto mnogi veliki računarski centri rade 24 časa dnevno. Poslodavci zahtevaju dobro opštje obrazovanje, i izvesno poznavanje rada računara smatraju određenom prednošću. Postali operatori na računaru dobra je odskočna daska za posao programera.

Programer unosi u računar uputstva na jeziku koji računar razume, kao što su kabal ili fortran. Kvalifikacije potrebne da se postane programer obuhvataju jasno metodičko mišljenje i sposobnost koncentrisanja i na najsjajnijim detaljima. Vešt programer je sasvim poseban tip osobe, pa iako su za prijem potrebne odgovarajuće školske kvalifikacije, priroda sposobnosti da se radi sa logikom mašine često ima veću težinu. Mnogi programeri čak ulaze u industriju bez formalnih kvalifikacija.

Programiranje je razdvojeno u dve grupe: primjenjeno programiranje i sistemsko programiranje. Primjenjeni programer pišu programe da bi rešili neki specifični zadatak. Sistemski programeri pišu programe da bi držali u redu računarski sistem — takav je, na primer, program koji ustanovljava grčke računare. Primjenjeni programer je u situaciji da sreće lude van računarske sobe — na primer, stranke koji koriste računar — i da radi sa tim programera na razvijanju programa "za određeni zadatak". Sistemski programeri su specijalizovani i skloni su ka samostalnom radu. Oni su, verovatno, najbliži direktnom komuniciranju sa inteligencijom mašine.

Poslovi programiranja, industrija računara veštacki poviša granicu iza koju je pristup dozvoljen samo najsjajnijim programerima i najboljim diplomiranim studentima. To je čarstvo sistem-analitičara i projektantata.

Sistemski analitičari

Sistem-analitičari razmatraju probleme i odlučuju kako se oni mogu rešiti pomoću računara. Neka naftna kompanija, na primer, otkriće je načinu na koji se uveličaju nafta i naftni naftni rezervi, neki delovi naftnotonog polja daju sirovu naftu pogodnu za proizvodnju benzina, dok drugi delovi daju naftu iskoristivu jedino u proizvodnji plastičnih masa. Treba odlučiti da li vredni investirati milione dolara da u eksploataciju polja. Odluka će zavisiti od predviđanja prilika na međunarodnom tržištu nafta u toku radnog veka polja (rečimo za sledećih 20 godina). Kompanija, isto tako, mora da odluči u kom će delu polja napraviti būšiti. Pošto su investicije ogromne, kompanija smatra razumno da potraži savet od računara i pred mu problem da ga on analizira.

Problem razmatra analitičar. On najpre treba da prikupi informacije o ranjenju stanju naftnog tržišta, zatim da se obavesti o potražnji raznih tipova sirove nafte i da konsultuje vodeće ekonomsko stručnjake o njihovom sagledavanju budučnosti različitih tržišta u raznim krajevima sveta. Prikupljena su i sva dostupna meraenja sa naftnotonog polja da bi se u radio računarski model koji će omogućiti simuliranje raznih rešenja za bušenje i eksploataciju polja. Analitičar, sa ekonomistima i projektantima programi, oblikuje potrebe računske programe. Konačni rezultat će se verovatno sastojati od velikog broja programa (koji možda uključuju grafičku predstavu strukture polja), koji pažljivo oblikovanjem i dedukcijom mogu pružiti logične zaključke o mogućnostima eksploatacije polja. Izvršni organi naftne kompanije biće u stanju da se na ovom modelu igraju pitanja „šta ako?“ otkrivajući kako bi razne odluke o ceni, tehnikama rafiniranja i tržišnim pristupima uticale na čitavu stvar. Dolazeći tako do svih informacija koje su im potrebne, oni će moći da donesu konačnu odluku o najboljem načinu eksploatacije svog naftnotonog polja.

Ovo je uprošćeni primer. Stvarni projekt može zahtevati nekoliko godina rada, angažovanje velikog računarskog tima i zasnovanost na detaljnim računarskim predviđanjima producije polja i perspektiva svetske privrede. Iz primera se, ipak, može videti da je uloga analitičara i projektantata programa vitalna, jer računari mogu da rade samo u okviru unapred propisanog metoda i na osnovu primljenih informacija. Analitičar je odlučujući faktor u dobrom korišćenju ra-

čunara, kada računar stvarno postaje dopuna i proizvodnje čovekovanog mozga.

Ima i nekih drugih važnih poslova u industriji računara, mada ih je malo koji se tako visoko cene kao posao analitičara. Izuzetak je, možda, projektovanje hardvera. Postoje dobiti izgledi za elektronske inženjere svih nivoa, od centra za popravku autoputeva do istraživačkih odjeljenja, ali oblasti razvoja proizvoda i čistog istraživanja otvorene su samo onima sa najvišim kvalifikacijama u elektronskom inženjerstvu.

Mnogi analitičari i projektanti mašina i softvera napreduju do rukovodećih i konzultantskih položaja, ali te titule često znače samo to da pojedinci imaju više vlasti, vrlo često radeći u sopstvenoj režiji. Radni sadržaj posla često nije promjen.

Ako industrija računara cvetla, mnoge druge industrije podležu recessiji; jedan OECD izveštaj objavljen u krajnjem 1982. ukazuje na to da će i nekoliko sledećih godina te nevolja prilikati većinu zapadnih zemalja — sa značajnim izuzetkom SAD i Japana.

Mikroričunarski centri

Neke vlade, posebno britanska, smatraju da mikroelektronika može pomoći da se reši neki kratkoročni problemi nezaposlenosti. Po jednoj shemi koju je ustanovila britanska vlada da bi omogućila sticanje izvešnjeg radnog iskustva velikom broju nezaposlenih mladih (pre nego postanu nezaposleni), nudi se mesto u 40 mikroričunarskih centara u Britaniji. U tim centrima, mlađi koji su završili školu, mogu pohađati kurseve o raznim aspektima mikroričunarstva, primajući už to malo novčanu naknadu. Ta shema pruža dovostruka korist: onima koji su prošli kroz školu bez dodira sa računaram (jer su napustili školu pre nego što je računar stigao, ili nisu bili „izabrani“ da ga koriste) pruža mogućnost da se s njim upoznaju, a poboljšava i njihove izglede da nadu neki posao.

Može se postaviti pitanje, kako osposobljavanje jednog završenog data da dobije zaposlenje koje počiva na računaru (čija je uloga verovatno da zameni nekoliko manuelnih radnika), pomaže problemu nezaposlenosti. Dvosmislenost o ulozi računara u zaposljavanju podstaknuta je izvestajima koji predviđaju masovnu nezaposlenost kao direktni posledici računarske automatizacije. Jedan nedavni izveštaj kazuje da će nezaposlenost izazvana automatizacijom doći do 16% u Zapadnoj Evropi. Za one koji završavaju školu bez ikakve računarske pismenosti, izgledi da se zaposle zaslužna su ništavni. Mnogi stručnjaci smatraju da mikrotehnologija pre svečeva jaz između onih koji imaju i onih koji nemaju, nego što dovodi do veće jednakosti, kao što neki futurolozi predviđaju.

Bez obzira na svoje sposobnosti i interese, deca koja se približavaju dobu kada završavaju školu, zaslužuju naše simpatije. U obrazovanju, u sustini jednom informaciono-prelaznom procesu, oni su se našla zahvaćena informacionom revolucionom na koji su mnogo lakše adaptirajući od svojih učitelja. Po završetku obrazovanja, oni se suočavaju sa jednim svetom koji se sam u prelaznom periodu između pred- i posla-računarskog doba našao u metežu. Ona moraju biti spremna da prežive u tom svetu, a jedna od bitnih kvalifikacija te spremnosti jeste sposobnost da vladaju kompjuterskom inteligencijom i da je koriste.

Računari iz drugog ugla **nemojte plakati za epлом!**

Najzamršnija i najuzbudljivija kompjuterska igra koja se trenutno igra zove se kupovina računara. Ona je obavijena mistrijom, opasnostima, intrigama i pršnim pozivima. Kao i u svakoj pustolovnoj igri, glavni igrači — bez obzira da li traže izgubljeno Blago, Malteškog Sokola ili Pravi Kompjuter — pitaju se: Kome mogu da verujem? To je pitanje o kome zaista vredi razmisliti.

Kupiti ili umreti

Pre svega, ne verujte nikome ko tvrdi da vam je računar potreban *odmah*. Fanatici kompjutera su usvojili nešto žara — i neke od izreka — religioznih zanesenjaka. „Računajte Kraj je blizu!“

U stvari, nema potrebe za žurbom. Živeli ste dugo bez računara i moći ćete sasvim fino još malo. Jedno od prvih pravila u kupovini kućnog računara jeste da se ne žuri. Nema potrebe da kupite računar još ove nedelje. Kupovina kućnog računara zahteva dosta strpljenja, izdržljivosti — i vremena.

Ne verujte oglašima za računare. Oglašavanje se pretvorilo od nemamerne samoparodije u sedmdesetih godina u nadrealizam u osmdesetim. Jedan tekući oglas koristi slogan: „Daleko od očiju — daleko od pameti!“

Ne verujte, takođe, ni časopisima. Neki specijalizovani časopisi za računare svakim danom postaju sve sličniji haj-faj magazinima. Jeste li ikad procitali loš izveštaj o velikom proizvođaču (čitaj — oglašivaču) stereofonskih uređaja u nekom od haj-faj časopisa? Prikazi su toliko laskavi i toliko puni obzira da su potpuno beskorisni.

Kao i haj-faj časopisi, i magazini za računare zavise od prihoda od oglasa za iste proizvode o kojima objavljivaju prikaze. Ako su suviše često iskreni — doviđenja oglašivači i, na kraju, zbogom časopis! „Popular Computing“ je odbio da objavi jedan moj prilog o tome zašto je Apple ile slabe vrednost. Razlog? Prema izjavi urednika: „Desilo se da ja volim Apple Computer korporaciju!“ Mislim da je dozvoljeno zamisliti da on, takođe, voli i mnogo strana Apple-ovih oglasa u boji svakog meseca.



Karakatura: Miroslav Aržovac

Vapaj u divljini

Dakle, kome verovati? Prijateljima koji imaju računare? Ne obavezno! Ljubitelji računara su obično skloni da se podaju onoj vrsti kompjutera koji poseduju bez obzira da li stvarno vredi ili ne. Sem toga, nakon što potroše ne baš zanemarljivo

sumu novca da bi kupili računar, malo ljudi će reći: „Pogrešio sam. Trebalo je da uzmem nešto drugo“. Više ljudi se kune u svoj „ševojet“ nego što ga proklinje.

I dok smo kod teme kola: prodavci računara možete verovati otpitliki isto toliko koliko i prodavcima polovnih automobila. Ponekad razmišljam da su se svi nezaposleni prodavci automobila prijavili i odmah dobili zaposlenje u prodačnicama računara od jedne do druge obale (misli se na Atlantski i Tihii ocean).

Ima, doduše, i nekoliko dobrih prodavača računara. Oni znaju isto tako dobro kao i

ja da većina daje lošu reputaciju manjini. Budimo fer — isuviše stvari se dešava previše brzo da bi se bilo ko održao, a da se ne zapusti. Ja se, možda, ne mogu zapustiti, ali ja ne moram da čekam na mušterije osam časova na dan.

Zato, za sve na svetu, ne uzimajte moju reč ni za šta. Ja sam samo glas koji vapi u divljini (Los Angeles). Misli iznesene u ovom članku nisu ništa drugo nego moja subjektivna, pristrasna, izuzetno lična mišljenja.

Kome, dakle, verovati? Ako želite da imate pravi računar za pravu cijenu —

Jugoslovenski ljubitelji računara okrenuti su, uglavnom, engleskom i nemačkom tržištu. U domovini kućnih računara, Americi, na čitav fenomen i, rekli bismo bizi, gleda se sa znatno manje neroze. Koje je njihovo vidjenje kućnih računara i pojedinih zvezda na računarskom nebū? Članak Pitera Viljama (Peter Williams) o najpopularnijim modelima kućnih računara „sa druge strane velike bare“ radikaljan je čak i za američke prilike, ali može da pomogne i našim ljubiteljima računara da raščiste sa nekim predrasudama. U svom članku Viljams propušta kroz sito i rešeto četrdesetak modela računara — mi smo izbor suzili samo na one koji su poznati i u našoj zemlji.

imajte poverenja u sebe! Uronite, učite koliko možete, razgovarajte sa što je moguće više sveta i na kraju ćete razaznati ono što će vam biti korisno.

TIMEX SINCLAIR 1000

Timex Sinclair je poznat kao prvi svetski računar za jednokratnu upotrebu. Kupite jedan za oko \$ 49.95, odnesete ga kući, koristite nekoliko dana i odlučite: (A) računari se podapaju i želite da dobijete neki bolji ili (B) računari vam se ne dopadaju i ne želite da imate više ništa sa njima. U oba slučaja oslobodajte se Timex Sinclaira sa otrplike isto toliko ceremonije kao kada praznите mišolovku.

COMMODORE

VIC 20 i Commodore 64 predstavljaju izuzetnu vrednost među jeftinim kućnim računarima. Za manje od 100 dolara VIC 20 je mnogo bolji računar za jednokratnu upotrebu nego Timex Sinclair. Za oko 200 dolara Commodore 64 ima skoro sve hardverske uređaje kao i Apple II, koji samo 1395 dolara. U grupi kućnih računara sa cenom ispod 500 dolara Commodore 64 je jasni pobednik.

TEXAS INSTRUMENTS 99/4A

Gde li su samo našli ime kao što je 99/4A? Izgleda da je čak i Bill Cosby (Bill Cosby) bio nesposoban da spase ovu mašinu. Texas Instruments ju je početkom ove godine (1983) prodavao uz gubitak, nadajući se da će razliku nadoknadi prodajom programa. Plan nije proradio. Iskazan je veliki gubitak. Akcije su pale. Ova situacija nije mnogo iznenadila one koji prate računare. Texas Instruments je obeshrabrio sve druge da rade softver za 99/4A. To je jednako pametno kao da obeshrabrite Standard Oil da proizvodi benzin za vaš automobil. Ako T.I. ne promeni tu politiku, i to brzo, neće još dugo biti na tržištu kućnih računara.

ATARI

Uzduž i popreko ATARI zemlje, ljudi hodaju okolo sa staklastim pogledom, mrmljujući „Šta se to dogodilo?“ Trenutak je izgledao da je Atari zagospodario tržište kućnih računara. Koštao je manje od Apple-a. Imao je „pamena“, imao je primamljiv slogan („Jeste li igrali Atari da-

nas?“) Šta se desilo? Desila se konkurenčija. Coleco je ponudio bolje igre — Commodore je ponudio jeftiniji računari. Ako Atari ne izvuče nešto magično iz svog šešira, i to uskoro, može završiti na sve većoj listi kalifornijskih ugroženih vrsta.

APPLE IIe

Apple računar ne predstavlja dobru vrednost već čitave dve godine. I Apple II i Apple III su preskupi, u poređenju sa slično opremljenim ačunarima, još od 1980. Apple II (Apple kaže da „e“ znači poboljšani-enhanced; ja kažem da to znači skup — expensive) dobio je još nekoliko znakova, malo slova, malo više memorije i priključak za štampač; smanjen mu je broj čipova (smanjenje ne predstavlja vrednost za kupca, ali čini računar jeftinijim za proizvodnju) i povećana cena. Čak i ja, koji sam imao malo poverenja u Apple, mislio sam da će ponuditi više a naplaćivati manje.

Glavni razlog zašto ljudi kupuju Apple II je softver i periferijska oprema koju proizvode drugi proizvođači. Većina toga potiče iz kasnih sedamdesetih godina kada je Apple bio jedini računars u boji na tržištu. Ali Apple gubi te prednosti. Postoji više lepih računara koji rade sa softverom Apple-a.

U stvari, ako pogledate na spisak od 16 000 programa koji stoje na raspolaganju za Apple, i koji je formata telefonskog imenika, videćete da je njihova veličina većina bezvredna. Izgledaju kao prilozi na konkursu za upis u višu školu. („Imate tri sata na raspolaganju da napišete program za računar. Počnite.“)

RADIO SHACK

Radio Shack je bila jedna od prvi kompanija (druge su Commodore i Apple) koja je ponudila lične kućne računare. Nakon nekoliko doasnih godina u kojima je izgledalo da se Radio Shack, kao i Apple, odmara na lovorkama (Apple se više ne odmara — on se ucrvljao) izšao je sa nekoliko računara koji su dobri i jedinim koji je vrlo dobar.

Dobre veste kod Radio Shacka je model 100. Možda ne biste zeli da vodite poslovanje svoje kompanije na njemu, ali biste mogli poželeti da ga stavite u svoju ručnu torbu. Trastura je puno većine i ima veliku preglednost. Ekran je tečni kristal, kao kod džepnih računara i ispisuje, osam redova od po 40 karaktera. Nije ono što bi nazvali mašina za obradu teksta — veće je mašina za zapise teksta. Sve što pohranite u model 100 može se preneti na drugi računar za nekaknovo izdavanje ili reviziju. Podaci se smestaju u vrstu RAM koja nikad ne zaboravlja. Nakon što se podaci prenesu ili iskoriste, memorija se može obrisati za naredne zadatke.

IBM PERSONAL COMPUTER

IBM je sagradio izvanredan kućni računar. Umesto da „sklapi“ mali računar košćenjem malo od ove IBM mašine a malo od one, IBM je sastavio grupu inženjera, dizajnera, programera i, da, nekoliko zabilježnika u kućne računare, i izdvojio ih u zadatku da projektuju i izrade IBM PC.

Među proizvođačima malih računara postoje je strah da će IBM progutati tržište kućnih računara. Umesto toga, izgleda da je IBM stvorio čitavu novu industriju: Svako je dobio deset učeća: proizvođači periferijske opreme, proizvođači programske opreme, prodavci programi, firme za prodaju putem pošte i preprodavci. IBM ispoljava neobičan stepen susretljivosti prema ljudima koji izrađuju stvari koje se ugradjuju u, ili priključuju u IBM PC. Pre svega, politika IBM-ja je bila: ako to nije napravio IBM, ne treba ni da se koristi za IBM. Sada je IBM čak objavio shemu PC-a i nudi svu moguću pomoć.

Čak, kao čudo nad čudima, IBM prodaje proizvod druge kompanije pod imenom IBM. To je jedan Epson, proizveden u Japanu, na koji se dodaje pločica IBM-ja. To je toliko radikalno kao da Steinway stavlja svoje ime na klavire proizvedene u Japanu. S druge strane, IBM/Epson izgleda malo star uz nove Epson štampe.

Tastatura je jedinstvena. Neki ljudi je vole, a neki mrze. IBM tastatura je dizajnirana sa onim što je poznato pod nazivom „audio taktilni řidbek“. Kara pritisnute tastere, vi čujete i osetite klik.

Drugi čudan element tastature je smještaj SHIFT tipke. Godinama, proizvođači malih kompjutera, želeći da s ponosom istaknu kvalitet svojih tastatura, navodili su da su „seletic-tipa“. Tipke su bile na istim pozicijama, kao kod IBM Selectric-a. IBM, iz razloga poznatih jedino stvaraocu ili stvaraocima, nije sledio seletic format... Ali, Keytronics je već predstavio tastaturu koja se može priključiti na IBM i ispraviti sve te, uh, ekscentričnosti. To je lepa stvar sa IBM-om: on će najverovatnije postati liki računara sa najviše podriške u istoriji. U stvari, on to već sad jeste.

Kao i kod Apple-ile, za sve što želite da radite sa sopstvenim računaram, bez sumnje ćete naći periferijski uređaj ili softver koji će vam to omogućiti. Apple IIe, ipak, to radi na nivou više škole, a IBM za — odrasle.

Da biste verovali, čak i pre godinu dana, da će IBM ikada proizvesti proizvod koji se zove XT? Zvuči kao ime sportskog automobil ili poboljšanog uređaja za pranje usta. Kako brzo veliki momci dozvoljavaju da im izmakne ugleđ kada treba „napraviti“ novac. I tako to ide. Dobre vesti i loše vesti o svetu kućnih računara su da se on brzo menjaju. Niko ne može biti stalno u toku. Sad umrite malo slabodrogog vremena da izadete napolje i dodirnete robu.

čekajući QL-a

Računari u izlogu



Mašina na koju se mora računati: QL

Nikada, verovatno, nećemo sazнати šta je Klijav Sinkler navelo da istriš pred rudu sa svojim novim računarem — preterano poverenje u svoje razvojne inženjerje, pritisak javnosti koja je čitavih četraest meseci sa nestripljenjem isčekivala njegov novi potез, želja da na samom početku godine preduhitri konkurenčiju koja je početa da najavljuje nove mašine, ili se već pojavila sa njima, ili, zapravo, poslovno nesmotrenost — niti je to za čitavu priču previše bitno. Tek, pred nama se, iako je od njegove januarske promocije prošlo više od pet meseci, još uvek nalazi samo desetak istih fotografija i morske nagadanja iz engleske štampe o računaru čiji razvoj, po svemu sudeći, neće biti okončan još nekoliko meseci. Zato se ovaj tekst neće previše baviti detaljima računara QL — može se dogoditi da se oni promene pre nego što se na našem izdanju osuši štamparska boja.

Više „hop“ nego skok

Pošto je na januarskoj promociji zašao medju publiku i napsao svoj šešir čekovima na po 400 miliona, Klijav Sinkler je prestat da se pojavi u javnosti. Za to je imao dobrih razloga — isporuka prvih mašina se toliko otegla da su čak i Englezzi, koji su spremni da se zakunu na ZX znak, počeli da sumnjaju da će se ovaj računar ikada pojavit na tržištu. Početak isporuke, sa zakašnjenjem od gotovo četiri meseca, malo je umirio računarske duhove, ali nije ni izdaleka otklonio sve dileme. Računar se isporučuje sa jednom ružnom izraslinom na zadnjoj strani kutije, koja treba da otkloni neke konstruktorske zablude iz prvih dana, i nedovršenim operativnim sistemom. Šta, uostalom, misliti o računaru koji nakon

nekoliko meseci upotrebe treba vratiti fabriči na doradu?

Za nagli skok na polju kućnih računara, koji, po mišljenju njegovih tvoraca, donosi QL, mogao je da bude primeren samo neki egzotičan termin poput „quantum leap“ iz kvantne fizike (kvantni ili nagli skok — otrilike kao svetionik na sred beskrajnog, pustog okeana). Za ovaj skok trebalo bi najviše da bude zaslužan mikroprocesor 68008 firme Motorola, koji Sinklerovi stručnjaci za propagandu deklarišu kao tridesetdvobitnu mašinu. To je, praktično, bila prva stvar na kojoj je Sinkler bio uvhaćen i koja je nagovestila potonje komplikacije sa QL-om. Iako inženjeri ne uspevaju da se slože što je to mikroprocesor od 32 bita, sasvim je sigurno da 68008 to nije — njegova unutrašnja arhitektura je, istina, u formatu od 32 bita, ali magistrala za podatke ima samo osam linija. To, u suštini, znači da on operiše sa podacima kao i bilo koji osmibitni procesor. Na račun Sinklerove pretenčnosti Englezzi zbijaju šalu kako QL, zapravo, ima samo dva bita — jedan spred i jedan pozadji tako se radi o veoma brzim i veoma moćnom procesoru, od nalog skoka je, izgleda, ostalo samo jedno „hop“.

Ko mnogo ima...

Oko drugog procesora iz QL ima znatno manje dilema. Radi se o mikrokompjuteru na jednom čipu, sa oznakom Intel 8049, u kojoj je, pored procesora, ugrađena i sistemski i radna memorija. Njemu su povezani trivijalni ali česti zadaci — kontrola tastature, generisanje zvuka i prijem serijskog signala — čime je glavni procesor rasteraćen jedne velike i „dosadne“ briže. U svome radu glavni procesor se dosta oslanja i na dva specijalno projektovana čipa — jedan kontroliše ekran i memoriju, a drugi mikrodravojave, časovnik, lokalnu mrežu i emitovanje serijskih signala — koji preuzimaju dosta veliki teret, štedeći snagu „glavnog gazde“ za udarne zadatke. I to je, kad je reč o operativnoj i kontrolnoj logici, sve!

Za sistemsku memoriju u prvobitnoj verziji računara bilo je odvojeno 32 K, ali se pokazalo da je operativnom sistemu na tom prostoru previše tesno. Prvi primerci isporučeni su sa dodatkom od još 16 K ROM-a. U konačnoj verziji, ovaj dodatak će biti ugrađen u računar, čime će se QL po sistemskim bajtovima približiti američkim mašinama, čiji su konstruktori u ovom detalju oduvek bili široke ruke. Pošto će, uz čip više koji nije ukalpljan u cenu računara, dobiti i neku neplaniranu pogodnost, ovaj previd vlasnicima QL ne bi trebalo previše da kvari raspolaženje.

Prilikom dimenzionisanja radne memorije konstruktori QL su, reklo bi se, bili znatno manje skrči — 128 kilobajta, u koje odjednom staje devedeset stranica teksta, mogu sa vlasnicima računara „galaksija“ učiniti „dovoljnim za čitav Beograd“. Pokazuju se, međutim, da „ko mnogo ima mnogo i troši“ da bi razvio svoju punu snagu, operativnom sistemu QL-a nije dovoljan postojeći RAM i moraće, po svemu sudeći, da sačeka na memorisko proširenje od 500 kilobajta tako se konstruktörima QL može uputiti zamerka da neusklađenost mašinskog i programskog dela računara, potreba za obilnom memorijom pre predstavlja vrlinu nego nedostatak. To je, u suštini, jedini način da se ostvari novi kvalitet.

Ipak nešto drugo

O tastaturi računara QL nije se, praktično, znao ništa sve dok prve mašine nisu isporučene kupcima. Stručnjaci za propagandu su, nedorečenim podacima, vešt podgrevali uši da se Sinkler ovoga puta opredeli za mehaničke tastere i profesionalne performanse. Pokazalo se, međutim, da je i ova tastatura zasnovana na principu membrana (!), ali i da je to urađeno dovoljno solidno za profesionalnu primenu računara — brz i tačan unos podataka u mašinu. Prilikom nedavne posete „Galaksiji“ Andelko Zagorec se žalio da je tastatura previše meka, osetljiva i „neposlušna“.

Januarska prašina nad računaram QL je već odavno pala, ali se još uvek ne vidi ništa. Mikroprocesor sa tridesetdvobitnom arhitekturom, 128 kilobajta radne memorije, novi bežik koji sa arhitečnim mikrosoftovskim varijantama ima zajedničko samo ime, novi operativni sistem koji pretenduje da postane standard za tridesetdvobitne procesore, dva ugradena mikrodrajva, četiri besplatna programa i ostale pogodnosti za svega 400 funti dovele su u iskušenje čak i najbolje poznavaoce računara, koji počeli javno da se održu svojih skupocenih računarskih sistema. Jutro je, međutim, pametnije od večeri — vreme je počelo da pokazuje da je u QL ugradeno više lepih želja njegovih tvoraca nego čipova i sistemskih bajtova.



Tačka na „i“ ili kamen oko vrata: Ugradivanje mikrodrayova u sam računar predstavlja hrabar i rizičan potez koji se još uvek nije srećno završio

dok su kritičari časopisa „Your Computer“ došli do zaključka da se na njoj radi sa lakoćom. U svakom slučaju, na njoj se, uz sve standardne znake i široku razmaknicu, nalazi pet tastera opšte namene kojima se mogu dodeliti proizvoljne funkcije. Naredbe se, kao i na svim ozbiljnim računarna, ne unose preko tastera sa dodjeljenim funkcijama (tzv. keyword), nego se kucaju slovo po slovo, kao i bilo koja druga reč.

Fina grafika . . .

Najčiglednija iznenadenja, međutim, vezana su za ekran QL-a. Zamišljen veoma ambiciozno, on je organizovan tako da vlasnicima nove Sinklerove mašine obezbedi pogodnosti znatno skupljih sistema. Pored fine grafike 256×256 tačaka sa osam boja, QL može da radi sa ultrafinom grafikom 512×256 tačaka u četiri boje. Grafički modaliteti se postavljaju automatski — u zavisnosti od toga da li se nakon uključivanja računara odbere rad sa televizorom ili rad sa monitorom. Korišćenjem pogodnosti za crtanje tačkama — vertikalnih i horizontalnih pruga u velikim i malih tačaka — na ekranu se može proizvesti 255 različitih boja! Boje i oblici slova su, kažu, veoma stabilni u oba grafička modaliteta — ni malik lelujanju kod računara Spectrum.

Ekran je — prvo u nizu iznenadenja — podeljen na tri dela. Dno ekrana — zeleni slova na crnoj pozadini — rezervisano je za četiri programske linije. Lava polovina ekrana — bela pozadina sa crvenim slovima — predviđena je za listing programa, a

Procesor

Motorola 68008 na 7,5 MHz (unutrašnja arhitektura 32 bita, magistrala za podatke 8 bita, direktno adresira 1 mebibajt radne memorije)

Sistemski memorija

32+16 K za operativni sistem QDOS i superbežikom — može se proširiti do 64 K

Radna memorija

128 K od čega je 32 K zauzeto za ekran — može se proširiti do 640 K

Operativni sistem

QDOS sa sledećim prednostima: multiprogramski rad, sistematsko poznavanje zadataka po prioritetu, proroci na ekranu, rad sa periferijskom opremom bezavistan od vrste uređaja

Jezik

Superbežik sa sledećim prednostima: strukturirano programiranje, jednostavno uvođenje novih naredbi (uključujući i sintaksu), brzina izvršenja nezavisna od dužine programa, jednostavno pozivanje mašinskih programa, jednake pogodnosti za reči i izzove, ekranски editor

Ekran

Ultrafina grafika na crno-belom ili monitoru u boji u dva modaliteta — 512×256 (četiri boje) i 256×256 (osam boja) — i 25 redova

desna — crvena pozadina sa belim slovima — za njegovo izvršavanje. Unošenje programa i naknadnih izmena je olakšano ekranškim editorom, a testiranje dovedeno do savršenstva — dok se preko leve strane ekrana kreće listing, na desnoj se može pratiti njegovo dejstvo. Na žalost, ove pogodnosti nisu izbrisane do kraja. Kada korisnik pokuša da dvaput izlazi neki program — a jedan program se u toku razvoja lista bezbroj put — u belom prozoru dolazi do „zagubu“. Osim toga, podela ekrana na tri dela predstavlja uobičajeni način rada QL-a, u „monitorskom“ modalitetu i ne može se ni na koji način preskočiti osim sistemskim komandama. Da bi ekran dobio uobičajeni izgled, treba otkucati najmanje dve naredbe, što, očigledno, nije previše elegantno.

... i softverske smicalice

Mogućnost da se na ekran postavi više prozora, pri čemu se u svakom mogu dešavati različite stvari, predstavlja jednu od najudarnijih pogodnosti novog Sinklerovog operativnog sistema. Časopis „Your Spectrum“ je, međutim, naziva običnom šalom i softverskom — smicalicom. Umesto „čvrstog“ sistema, kakav, na primer, koristi Apple, Digital Research ili Microsoft, tvorci QL-a su, naprosto, po ekranu iscrtili okvire. Za svaki okvir, doduše, obezbedena je programska kontrola sa svim parametrima

teksta sa do 85 (monitor), odnosno 40 do 60 znakova u svakom redu (TV)

Tastatura

QWERTY punog formata sa 65 tastera, širokom razmaknicom, levim i desnim šift tasterom, četiri tastera za kontrolu kurzora i pet tastera opšte namene

Mikrodrajv

Dva ugradena mikrodrajva minimalnog kapaciteta 2×100 K, prosečnim vremenom pristupa podatku od 3,5 sekundi i brzinom učitavanja od 15 K u sekundi

Proširenje

Pored priključaka za napajanje, TV i monitor, obezbeđeno je i devet portova za periferijske uređaje: radna memorija, mikrodrajv, sistemski memorija, serinski interfejs (2) lokalna mreža (2) i palice za igru (2)

Serijski interfejs

Dva standardna RS232C interfejsa za printer, modem itd. sa podešivom brzinom prenosa između 75 i 19200 bita u sekundi za dvostruki dvosmerni prenos

Dimenzije

138×472 mm

Masa

1388 G

Cena

399 funti

kao i za čitav ekran, ali ona ne dejstvuje pouzdanije: kada ne može da stane u jedan, tekst „ladno“ prelazi i u drugi prozor — brišanjem jednog prozora tako se može obrisati i onaj do njega. Slični efekti, zaključuje „Your Spectrum“, mogu se postići i na bilo kom drugom računaru, uključujući i četiri putne jetfline Spectrum.

„Dva časa nasamo“

Ovim se sličnosti između QL i Spectruma ni izdaleka ne iscrpljuju — njihovi operativni sistemi i njihovi bežici imaju, očigledno, zajedničke temelje, koji bi učinili računarskog „arheologa“ verovatno odvele sve do računara ZX 80. Nije, razume se, nimalo jednostavno suditi o programskom jeziku jednog računara na osnovu sličnih naredbi i funkcija i šturi tudi utisaka tipa „dva časa nasamo sa QL-om“. Programski jezik QL-a, nazvan superbežik, pisan je očigledno sa velikim ambicijama i snažnom željom da nadmaši sve svoje prethodnike. Mada pomalo pati od „kemičkih“ brojčanih akademizma — „dugačke, nezgrapne naredbe i šture, jezgrovite poruke“ — superbežik, izgleda, ima sve odlike zrelog programskog jezika: viselinjske funkcije, procedure sa lokalnim promenljivim, promenljive sa dugim imenima, formirano listanje i ekranški editor. Arhitečne naredbe tipa GOTO i GOSUB uvršćene su, izgleda, samo iz nostalgije, a FOR-NEXT pete lične na klasičnu samo po imenu.

Da bi se otkrile dobre i slabe strane superbežika potrebno je barem nekoliko meseci rada sa QL-om, ali je već sada jasno da mu nedostaje malo više logike i finoće u detaljima. Uočena su, takođe, i neka veoma uspešna i moćna rešenja. Procedure se mogu izvršavati i potpuno nezavisno – izvan svog matičnog programa. Pošto svaka procedura može imati i svoje parametre, uvođenje novih naredbi u superbežik predstavlja pravu dečiju psemicu. Fleksibilnost, očigledno, dini jednu od jačih strana ovog jezika. Na osnovu hardverskih specifikacija moglo bi se pomisliti da se programi izvršavaju „kosmičkom brzinom“. Prvi utisak, međutim, govore da je QL samo dvaput brži od Spectruma, a u FOR-NEXT petljama ni toliko.

Biser u školjci...

Konstruktorma QL-a, po svemu sudići, najviše problema zadaje operativni sistem nazvan QDOS. tako je prošlo već više od pet meseci od najave računara, on ni do danas nije završen, niti ima izgleda da se to dogodi u nekoliko narednih meseci. Zamisljen možda previše pretentaciono – da postane standard za Motoroline tridesetdvobitne procesore – QDOS trenutno najviše hraniće upravo u onim osobinama koje treba da predstavljaju njegovu najjaču udarnu snagu: promena parametra u toku izvršavanja programa, na primer vrednosti neke sistemске promenljive, paralelan rad više programa u memoriji i na ekranu i udoban i brz rad sa mikrodrajvovima.

Nedorečenost, verovatno, predstavlja osnovni razlog koji je naveo tvorce QL-a da operativni sistem zatvore za korisnika. QDOS se, na žalost, ne može koristiti direktno u sopstvenim programima – za sada se iz superbežika ne može prići ni jednoj od njegovih udarnih pogodnosti. Deo za rad sa diskovima ima dovoljno snage za dva mikrodrajva sa po desetak fajlova na svakom, ali se može pokazati potpuno nemodernim kada se jednoga dana na QL prikući tvrdi disk.

Rad sa tastaturom, ekranom i periferijskim uređajima zasnovan je na sistemu „struja“ i „kanala“ koji je razrađen na računaru ZX Spectrum (vidi str. 68–71). Ovaj sistem omogućuje veoma jednostavnu i elegantnu softversku vezu sa periferijskim jedinicama – takvu da ne priključeni uređaj QL gleda kao na organski deo svoje sopstvene mašine. Povezivanje određene struje u mašini sa određenim kanalom (periferijski uređaj) može se izvršiti sa čitavim spiskom parametara koji se, na primer, odnose na vrstu štampača i modalitet rada. Otvaranjem struje u memoriji se automatski rezerviši prostor za 16 K sistemskog programa za pogon priključenog uređaja. Sistem je, čak, u stanju da prepozna da li je periferijski uređaj sa suksesivnim (štampaci, tastatura, mreža) ili slučajnim pristupom (mikrodrajv, tvrdi disk) podaćima. Vreme će, možda, pokazati da su struje i kanali ubožljite QL-ovo oruđe od multiprogramskog rada.

... a školjka na dnu mora

Mogućnost paralelnog rada više programa u memoriji – dok čekate da na printeru

Rečnik QL-a						
Procedure						
ARC	OPEN	NEW	PAPER	INT	IF THEN ELSE	
ELLIPSE	OPEN-IN	RETRY	RECOL	LN	DEFine	
LINE	OPEN-NEW	SAVE	SCROLL	LOG 10	PROCedure	
POINT	POKE	SBYTES	STRIP	SIN	RETURN	
SCALE	POKE-W	SEXEC	UNDER	SORT	DATA	
BEEP	POKE-L	STOP	WINDOW	TAN	END FOR	
INPUT	RANDOMISE	AT	Funkcije	PI	END IF	
PAUSE	CLEAR	BLOCK	BEEPING	CHR\$	END REPeat	
PRINT	CONTINUE	BORDER	INKEY\$	CODE	END DEFine	
BAUD	EXEC	CLS	KEYROW	PEEK	END SElect	
CLOSE	EXEC-W	CSIZE	ACOS	PEEK-W	EXIT	
COPY	LBYTES	CURSOR	RND	PEEK-L	NEXT	
COPY-N	LIST	FLASH				
DELETE	LOAD	INK				
DIR	LRUN	OVER				
FORMAT	MERGE	PAN				

izliza vaš najnoviji program, računar vas može zabavljati lepotom partijama baha – sa svim tokovima na ekranu trebalo bi da predstavlja biser QDOS operativnog sistema. Nevolja je, jedino, u tome što je ta školjka još uvek duboko na dnu mora. Paralelni rad, uostalom, nije neobičan čak ni za najeffiniji računari. Svaki operativni sistem radi paralelno na različitim zadacima – očitavanju tastature, prikazivanju znaka na ekranu, generisanju zvuka – ali je obično toliko spor da izgleda kao da se ovi poslovi odvijaju jedan za drugim. Oni se, da stvar bude lepša, tako i odvijaju. U čemu je onda štos kod QL-a? On ima dva procesora. Prelazak sa jednog programa na drugi je, osim toga, dovoljno brz da se to ne primeti. Multiprogramski rad značajno povećava kvalitet obrade podataka. Voleli bismo, zato, da tvorići QL-a istraju u svojim namearama. Time bi se u kućne računare preseila još jedna pogodnost velikih računarskih sistema.

Kamen oko vrata

Nijedan hardverski detalj na QL-u nije izazvao toliko protivničare reakciju kod mikrodrajvova. Dok jedini njihov ugradnjem u sam računar smatraju mudrim, klijunjem potezom, drugi su ubedeni da oni predstavljaju QL-u kamen oko vrata i da će Sinkler kad-tad morati da potrazi druga rešenja. Mnogi su još i sada ubedeni da će u konačnu verziju QL-a biti ugradeno nešto sasvim drugo – minidiskov firmi Hitachi. Zagovornici mikrodrajvova smatraju ovaj Sinklerov izum dosta elegantnim sredstvom za trajno skidalište podataka, a protivnici neozbiljnijom igračkom koja ni po čemu nije primerena mogućnostima QL-a.

Verovati u mikrodrajve nije, ne pozakuj se da on zasad predstavlja najveću kočnicu razvoja novog Sinklerovog računara. Da bi se na traci pronašlo određeni podatak, računar radi „u prazno“ 3,5 sekunde, a učitavanje jednog od besplatnih programa traje sedamdesetak sekundi – trenutak u poređenju sa kasetom, većnost u poređenju sa disk jedinicom! Preterano sporo su, u zadnji čas, postali i svešni i tvorci QL-a, koji grožnjavaju tragaju za efikasnijim operativnim sistemom. Tragajući za podatkom na traci, operativni sistem ide od jednog do drugog bloka – ako ono sto se traži nije u prvom bloku, prelazi se na drugi, treći i tako sve do kraja. Direktni pristup blokovima će, verovatno, ubrzati učitavanje podataka, ali ni to, po svemu sudići, neće biti dovoljno da se spase postojeća koncepcija.

Neko to voli drugčije

Na zadnjoj strani QL-a nalazi se nekoliko priklikuća — računska mreža, monitor, TV, serijski interfejs, palice za igru, spoljni ROM i spoljni RAM. Ma kako to zvuči neverovatno, u mašinu koja je, između ostalog, namenjena i za obradu teksta, nije ugrađen paralelni interfejs za štampač niti se našlo mesta za port opšte namene! Sinkler, koji se, očigledno, upinje iz petinih žila da nametne svoje standarde računarskom svetu, pokazuje, izgleda, veliku obođljivost kada se radi o standardima koje su izmisili drugi.

Uz račun QL isporučuje se i četiri ozbiljna „profi“ programa koja dolaze iz radionicne firme Psion — program za obradu teksta, program za obradu podataka, knjigovodstvo i poslovnu grafiku. U zapadnom svetu, gde je odnos prema programima značio drugačijeg nego kod nas, jedan ovakav softverski paket predstavlja sam za sebe dovoljan razlog da se neko odluči za kupovinu QL-a. Mada imaju veoma efektan izgled ekranu, oni su pisani u trenutku kada je operativni sistem QL-a bio u embrionalnom stadijumu i, sasvim sigurno, ne mogu da demonstriraju najvažnije mogućnosti računara. Ne treba, međutim, sumnjati da će vrlo brzo na tržište biti izbačene unapredene verzije.

Performanse računara QL objavljene su šest meseci ranije nego što je trebalo i konstruktori sada imaju velikih problema da održe datu reč. U tome, možda, leži i najveći nedostatak QL-a – od njega se očekuje znatno više nego što može da pruži. A brzopletost, kao i svuda, ima svoju cenu – jedan inače i savremeno i zrelo koncipiran računar izlazi iz pred svoju kupe sa neizbrusenim bežikom, nedovršenim operativnim sistemom i nekoliko otvorenih pitanja oko konačne verzije mašine. Isporuča računara je, izgleda, bila isto toliko brzopletka kao i njegova promocija – pored svih nevolja Sinkler sada ima za petama i nezadovoljne vlasnike čije mašine ne rade kako treba.

„Ko igra na očtrici tehnološkog noža, može i da se poseće“ — mudruju Englezzi. A klijaju Sinkleru preostaje da veruju u izreku da se ipak najslade smjeje onaj koji to čini poslednji. Sudjeći prema prvim očenama da „QL, sasvim zavređuje svoju cenu“ i „da će izvršiti snažan uticaj na razvoj kućnih računara, sličan onome koji su ostvarili Apple, IBM PC i Spectrum“, možda će ipak on biti taj.

Elektron

Ejkornu nije bilo nimalo lako da koncipira kompjuter koji bi se uklonio u liniju koju su činili Atom, BBC A i BBC B: novi računar je morao da bude toliko jeftin da ugrozi modele iz ekonomskog klase, toliko moćan da konkuriše Commodoru 64 i drugim računarima iz srednje klase, a ipak ne toliko moćan da udrima pozicije BBC modela A i B. Svi ovi zahtevi, jasno, nisu mogli da budu ispunjeni, pa je Ejkorn odlučio da žrtvuje BBC A i izbaciti na tržište računar koji će u mnogome kopirati BBC, ali koji će biti lišen nekih skupih delova i tako biti na raspolaženju onima koji za kompjuter nisu spremni da izdvoje više od 200 funti. Electron je ispunio očekivanja iz našeg prvog specijalnog izdanja: došao je na sam vrh tabele najtraženijih modela na evropskom tržištu!

Dlaka u jajetu

Electron ima 32 Kb ROM-a i isto toliko RAM-a. ROM je, kao i kod BBC B, podešen na operativni sistem (ekvivalentan sa verzijom 1.20) i bežik (BASIC II, naravno). Umesto bežika, jasno, može da se prikupi EPROM sa nekim drugim programskim jezikom (za sada je na raspolaženju fort), procesor teksta ili nešto treće. I pored ovih očiglednih sličnosti, hardver Electrona je sasvim originalan i nema većinu dodirnih tacaka sa BBC-jem. Upotrebljivanje je manje komponenti i ostavljanje manje praznina podnožja za proširenja; računar je, tako, postao manji, jeftiniji i, na žalost, sporiji.

Glavna zamerka koju korisnici BBC-ja obično upućuju Electronu je nedostatak teletekst kompatibilnosti. Nije nikakva tragedija što računar nije kompatibilan sa teletekst mrežom, već što mu nedostaje MODE 7 koji većinu korisnika BBC (računajući i autora ovoga teksta) upotrebljava otprilike 90% vremena rada. Zašto je mod 7 važan? Jednostavno zato što računar, dok radimo u njemu, rezerviše samo 1 Kb RAM-a za video-memoriiju, pa je 31 Kb slobodan za naš program i podatke. Korisnici Electrona obično rade u modu 6, koji za ekran rezerviše čitavih 8 Kb.

Electron ima generator tonova koji u svemu imitira BBC-jev osim u jednoj „sintetiči“; u jednom momentu možemo da generišemo samo jedan ton što, na prvi pogled, značajno smanjuje njegovu vrednost. Čini nam se da zaključne ne treba donositi preuranjeno: treba biti pravi majstor da bi se synchronizovao rad BBC-jevih kanala i dobila „orkestersku“ muziku. U najvećem broju programa (čak i izvanrednih akcioneih igara poput Snapper) i BBC B koristili samo jedan kanal za sintezu tona!

Razlike bez značaja

Ostale razlike između BBC-ja i Electrona su primetno praktično na prvi pogled: tastatura je, da počnemo od nje, manja i

sastoji se od znatno manje tastera (55 prema 74). Korisnici Electrona su na prvi pogled ostali bez tastera kojima mogu da dodeljuju proizvoljne literale. Kažemo na prvi pogled, jer se istovremeno pritiskom na CAPS LOCK i neki od brojeva dobita funkcija ekvivalentna pritisku na neki od BBC-jevih narančastih tastera. Kao posebna pogodnost, taster BREAK takođe može da dobiti dodatne funkcije.

Taster BREAK je jedna od velikih BBC-jevih i Electronovih pogodnosti. Ma šta računar radio u nekom trenutku (izvršavajući bežik ili mašinskog programa, potpuna blokada usled neke greške, traženje programa sa kasete ili disk-a ili nešto četvrtog), pritisak na BREAK ga vrati u normalno stanje i reinicIALIZUJE bežik. Program koji se nalazio u računaru biva privremeno izgubljen i vratio se tek kada odlukujemo OLD. Zašto bismo kucali OLD kada god pritisnemo BREAK? Lakše je, pre početka rada, otukati KEY 10 OLD CTRL M i taster BREAK će, pored uobičajene funkcije, izvršavati OLD po reinicIALIZaciji računara.

Loša strana Electrona je što praktično nema priključenih interfejsa. Na njegovoj zadnjoj strani su samo priključci za kasetofon, televizor i kolor odnosno crno-beli monitor, dok je na donjoj strani samo običan port za ekspanziju. Svi oni koji žele da kupuju periferijske uređaje moraju, dakle, da nabavljaju razne dodatke kojima ćemo posvetiti sledeće redove.

U prodaji je takozvani teletekst adapter koji Electronu dodaje mod 7 sa svim opcijama — izgleda da je teletekst mod podržan programom u ROM-u i da je potreban samo hardver kojim bi se realizovalo. Programi za podršku štampača su takođe u ROM-u, dokaz je što su naredbe VDU — VDU 3, kod BBC-ja namenjene kontroli štampača, označene kao „Reserved“ u uputstvu za upotrebu. Da bismo, dakle, povezali Electron sa štampačem, treba da kupimo Centronics interfeks koji će nas koštati tridesetak funti. Povezivanje sa diskovima je takođe moguće, ali je potrebno kupiti disk interfeks koji se, prema neprverenim informacijama, smesta u posebnu kutiju i povezuje sa portom za ekspanziju. Čak je i za dodavanje palica za igru potrebno dokupiti A D konvertor. Jasno je da svi ovi dodaci

značajno povećavaju cenu Electrona, koja postaje sve približenija BBC-jevoj.

Razmislite još jednom

Programska podrška je jedna od jačih strana Ejkornovog novog kompjutera. Praktično svi programi pisani za BBC B (a takvih je, kao što smo rekli, dosta na tržištu) mogu da se pružaju na Electronu obratno. Jedini problem je mod 7 koji je zbog mogućnosti pisanja velikih slova, koširčen za „naslovne strane“ mnogih BBC-jevih igara. Izvršavanje ovakvih programa na Electronu je, naravno, moguće ako se lišimo prizora koji obično traje dok se program učitava sa trake. I kod nas su u modu zajednički aranžmani kupovine stranih programa koje ravnoopravno i u međusobnoj saradnji nabavljaju vlasnici BBC-ja i Electrona. Vlasnicima Electrona je, osim Acorn Usera i A&B Computing, na raspolaženju i izvanredni časopis Elbug koji izdaje klub Beebug.

Sve u svemu, Ejkorn Electron predstavlja izvanredan izbor. Nije lako naći računara koji koštaju 200 funti sa moćnim i bitnim strukturiranim bežikom, grafikom 640×256, mogućnošću sinteza tonova (eventualno) glasa i visoko profesionalnom tastaturom. Preporučili bismo, ipak, onima koji razmisljavaju o kupovini Electrona da realno procene svoje potrebe za ekspanziju sistema. Ukoliko shvatite sva ograničenja računara i zaključujete da vam ona ne smetaju, kupite Electron što pre. Ako, sa druge strane, želite proširenja, bide svakako bolje da razmotrite mogućnosti nabavke duple skupljeg BBC B — ne samo što će procijeniti nego ćete imati manjih problema sa uvozom. Filozofija kupovine osnovne verzije računara i postupenih proširenja prema potrebama je za strance preporučljiva, ali kod nas vladaju drugi zakoni.

CPU: 6502
Klok: 1,79 MHz
Memoria: 32 K ROM-a

32 RAM-a

do 80 znakowa, 8 boja, 120 delniczych i 266

127/255

korisnik
TV.

kompozitní video, RGB

videlicet, *klassenlos*,
systematisch bas

Jezici: bejzik sa
ugradenim asemblерom

Tastatura: 56
mehaničkih tastera.

opciones unos naranjas
y otros amarillos.

legimi tastera 28

• Dodatajivanje funkcija Dokumentacija:

Izvanredna sa dobrim
odisom operativnog

sistema i obiljem

Detalja za programiranje na

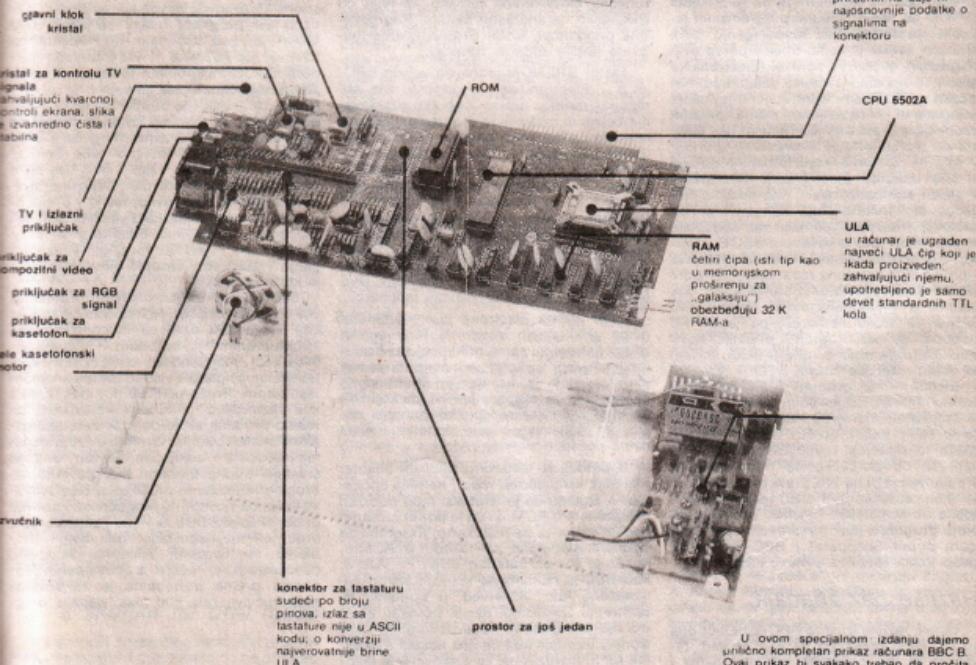
jeziku i mašinskom

Dimenzijs: 340 x 160 x 65 mm

Cena: 199 funti



fastatura



prostor za još jedan

konektor za tastaturu
sudeći po broju
pinova, izlaz sa
tastature nije u ASCII
kodu; o konverziji
najverovatnije brine
I_{EL}A

U ovom specijalnom izdanju dajemo unično kompletan prikaz računara BBC B. Ovaj prikaz bi svakako trebao da pročita svakog ko namerava da nabavi Electron: dva računara su izuzetno slična i gotovo sasvim kompatibilna, pa čemo se u glavnom usredstvili na razlike.

U specijalnom izdanju „Računari u vašoj kući“ BBC mikrokompjuter je ocenjen kao najkompletniji stoni računar koji se može naći na tržištu. Ta konstatacija je izrečena posle opsežne konsultacije stranih časopisa, ali se autor ovoga teksta u nju do kraja uverio tek kada je dobio priliku da intenzivno radi sa BBC-jem i upozna njegove tajne. Šta je, dakle, to što BBC B (model A se praktično više i ne proizvodi) čini najboljim?



Računar bez premeta BBC B

Prvi utisak o BBC-ju mora da bude povoljan: na (za jedan stoni kompjuter) velikoj ali veoma lepo oblikovanoj kutiji dominira visoko profesionalna tastatura, na kojoj su, osim standardnih alfanumerika, zastupljeni mnogi specijalni znaci (računajući čak i velike i srednje zagrade, ali, začudo, ne i znak copyright koji ostaje privilegija vlasnika Spectruma). Utisak o kompletnosti tastature dopunjava deset narančastih tastera kojima možemo da dodeljujemo proizvoljne literale. Imamo utisak da jedino tastature Hewlett-Packardovih kompjutera mogu da se mere i (verovatno) nadmašte BBC-jevu.

Ekranски editor

Po uključenju računara nalazimo se u takozvanom „modu 7“. Još smo u „Računaru 1“, naime, rekli da je kod BBC kompjutera napravljen kompromis između potrebe za visokom rezolucijom i velikim slobodnim RAM-om tako što su uvedeni

razni grafički modovi — ukoliko želimo visoku rezoluciju, moramo da se ponišrimo sa time da ćemo imati malo RAM-a i obratno. Mod 7 na ekran troši samo 1 kilobajt, pa je, verovatno, zato i izabran za početak rada. To je dobro, ali se pojavljuje jedan drugi, po našoj oceni za početnike vrlo ozbiljan, problem: mod 7 je kompatibilan sa takozvanim „telefekstom“, dok ostali sadrže karaktere oblikovane po ASCII standardima. To znači da, na primer, taster na kom je ispisana srednja zagrada u modu 7 daje levu strelicu što, u prvi mah, može itekako da zbuti. Zato bismo početniku svakako savetovali da, pre nego što započne „razgovor“ sa računaram, izabere MODE 6.

Čim počnu da se pišu prvi bežik programi, ukazuju se potreba za upoznavanjem editora koji omogućava ispravljanje grešaka u programu. BBC može da se podiže neobičnim ali izuzetno moćnim i fleksibilnim ekranškim editorom. Pre nego što ga opisemo, neće smetati da pomenemo razlike između ekranških i linijskih (linijski editori se koriste kod velikog broja računara iz ekonomski i srednje klase) editora: da bismo ispravili neku naredbu treba da otuknemo EDIT (ili neku odgovarajuću naredbu) i broj linije. Zatim pomoću leve i desne

strelice pomeramo kurzor i, odgovarajućim komandama, vršimo ispravke, posle kojih pritisnemo ENTER. Kod ekranških editora su u upotrebi i druge dve strelice: pomoću njih pozicioniramo kurzor na neku poziciju ekran-a i vršimo ispravke „na licu mesta“. Samo se po sebi razume da su ekranški editori daleko fleksibilniji, i moćniji, ali i oni imaju veliku manu: ako smo, na primer, kucali neku dužu komandu (naredbu bez linijskog broja) i pri tom pogrešili, ostaje nam jedino da je ponovo kucamo osim ako želimo da se pomirimo sa neredom i komfuzijom na ekranu.

BBC-jev editor je, u osnovi, ekranški, ali otklanja ovaj nedostatak: čim pritisnemo neki od četiri tastera na kojima se nalaze strelice, standardni kurzor se pretvara u beli kvadrat i pojavljuje se drugi kurzor koji se, daljim pritiscima na strelice, pozicionira na proizvoljnu poziciju ekran-a. Pritiskom na taster COPY sa te pozicije kopiramo tekst na poziciju kurzora-kvadrata (taster COPY, kao i svi drugi tasteri, ima funkciju autorepete ako ga duže pritisnemo). Na taj način se može kopirati proizvoljna grupa slova — nijedan tekst sa ekran-a nije „mrtav“: ako je kompjuter, na primer, nede ispisao „Mistake“, možemo slobodno da tu reč uklonimo u neku našu naredbu ili

komentar. Samo se po sebi razume da je moguće kopirati i proizvoljne komande, što postaje posebno korisno kada poželimo da snimimo isti program nekoliko puta.

Strukturirano programiranje

Po upoznavanju editora možemo da počnemo sa pisanjem prvih bežik programa i upoznavanjem BBC-jevog programskog jezika. Uzbroj uvidimo koliko je on kompleksniji i moćniji od ostalih verzija bežika. BBC bežik je, naime, sakupilo u sebi mnoga dobra svojstva drugih programskih jezika, ponajviše paskala. Tu, pre svega, mislimo na velike potencijale za pisanje strukturiranih programa. Strukturirano programiranje omogućava ne samo jednostavnije pisanje programa sa manje grešaka nego i jednostavnije docnije modifikacije programa i njegovo prilagodjavanje drugim kompjuterima.

Strukturirani program se piše po takozvanim stepenima složenosti: početni problem se deli na nekoliko etapa, a zatim se svaka ta etapa posebno programira daljim rešavanjem na jednostavnije probleme. Postupak se nastavlja sve dok se ne dođe do krajnje jednostavnih problema koji se rešavaju u nekoliko programskih linija. Svaka etapa rešavanja problema se izdvaja u poseban potprogram koji je u principu nezavisan od ostalih i ima svoje ulazne i izlazne varijeline. Pošto je taj program (nazivamo ga i „modul“) razvijen i potpuno testiran, snimamo ga na traku i prelazimo na sledeći. Na kraju posla jednostavno učitavamo module i povezujemo ih u celini. Kada je, na primer, docnije potrebno da prilagodimo već postojeći program drugom računaru, bice dovoljno da, bez ikakvih intervencija u glavnom programu, preradimo module koji operišu sa tastaturom i ekranom.

Da bi se na racionalan način pisali strukturirani programi, treba imati mogućnost pozivanja potprograma sa prenošenjem argumenta i definisanjem lokalnih promenljivih. Ovakva mogućnost je BBC-ju data preko naredbi PROC, FN i LOCAL. U glavnom programu se, na primer, može naći naredba PROCsuma (A, B, C). Kada računar nađe na nju, potražiće naredbu DEFPROCsuma (X, Y, Z) (umesto X, Y, Z mogu, jasno, da se koriste i bilo koja druga slova ili reči) i promenljivoj X dodeliti vrednosti promenljive A, a promenljivoj Y i Z, respektivno, vrednosti promenljivih B i C. Zatim se izvršavaju naredbe iz DEFPROCsuma. Ukoliko je prva od njih naredba LOCAL A, u potprogramu će biti nezavisna od promenljive A u glavnom programu, što znači da će potprogram biti savsim autonoman. Umesto procedure, može da se koristi i funkcionalni potprogram koji počinje sa DEFNsuma (A, B) a poziva se, na primer, sa: X=2+FNsuma (total, sa-birak)+32.

Ništa nije savršeno

Iako su konstruktori BBC-jevog bežika očigledno uložili veliki napor da omoguće pisanje savremenih, strukturiranih progr-

INSTRUKCIJE RACUNARA BBC I ELECTRON

ABS	ACS	ADVAL	AND	ASC	ASN	ATH	AUTO
BGET#	BPUT#	CALL	CHR#	CLEAR	CLOSE#	CLG	
CLS	COLOUR	COS	COUNT	DATA	DEF	DEG	DELETE
DIM	DIV	DRW	ELSE	END	ENDPROC	ENVELOPE	EOF#
EOR	ERL	ERR	EVRL	EXP	EXT#	FALSE	FN
FOR	GCOL	GET	GET\$	GOSUB	GOTO	HIMEM	IF
INKEY	INKEY\$	INPUT	INPUT#	INSTR	INT	LEFT\$	LEN
LET	LIST	LISTO	LN	LORD	LOCAL	LOG	LOMEM
MID\$	MOD	MODE	MOVE	NEW	NEXT	NOT	OLD
ON	OPENIN	OPENOUT	OPT	OR	PAGE	PI	PLOT
POINT	POS	PRINT	PRINT#	PROC	PTR#	RAD	RERD
REM	RENUMBER	REPORT	REPORT	RESTORE	RETURN	RIGHT\$	RND
RUN	SAVE	SIGN	SIN	SOUND	SPC	SQR	STEP
STOP	STR\$	STRING\$	TAB	TRN	THEN	TIME	TO
TOP	TRCE	TRUE	UNTIL	USR	VAL	VDU	VPOS
WIDTH							

ma, očigledno su propustili da posao doveđu do kraja. Pre svega, prenošenje matrica i nizova kao argumenta nije jednostavno i, štavi se, nije dovoljno. Tako se uvidom u stručne časopise ulazi u ovu tajnu i tako pisanje potprograma za rad sa matricama i sistemima jednačina (školski primer korisnosti strukturiranog programiranja) postaje moguće. No, izgleda da nizakvimi trikovima nije moguće sintetisati lokalnu ON ERROR naredbu; naredbu pomoći koje bismo bili u stanju da kontrolišemo eventualne greške u proceduri a da ne remetimo tretiranje grešaka u drugim procedurama i glavnom programu.

Nedostaje, osim toga, lokalna RESTORE naredba, kao i mogućnost da se jednostavno „sačuva“ status računara (sadržaji svih sistemskih promenljivih koje su, inače, prilično razasute po memoriji) koji bi se „povratio“ po nailasku na standardnu naredbu ENDPROC. Ove nedostatke 90% korisnika neće ni zapaziti, ali nikome, sigurno, neće biti pravo što, u obilju naredbi, MERGE nije napisano mesto. Ova naredba se, naravno, može jednostavno sintetisati ali, pogotovo za početnika, to nije isto kao da je ona u ROM-u. Pri povezivanju modula neobično je korisno što računar ima vrlo fleksibilnu i izuzetno brzu naredbu RE-NUMBER.

Ultravisoka rezolucija . . .

Rezolucija 640×256, sasvim jasno, pokazuje da su Ejkornovi inženjeri želeli da BBC ima moćnu grafiku, pa su se potrođuli da obezbeduju kompletan set komandi za kontrolu displeja. Lepa je pogodnost što je mapa ekranu ista u svim modovima: jasno je, na primer, da u modu 1 ima duplo manje tačaka koje mogu da se kontrolisu nego u modu 0, ali je tačka sa koordinatama 1279, 1024 u levom gornjem ugлу pa će program planirati za mod 0 raditi i u modu 1 tako da, jasno, slika biti grublja.

Za crtanje se koriste naredbe MOVE i DRAW, čija funkcija postaje jasna kada pročitamo njihove nazive. No, mnogo je korisnika naredba PLOT koja ima tri argumenta: prva dva govore o funkciji koju treba izvršiti (PLOT 4 je, na primer, isto što i MOVE), a preostala dva daju koordinate tačke. Naredba PLOT obavlja neke poslove koji bi, inače, mučili programera: crtanje isprekidane linije, bojenje kvadrata, trouglova i mnogouglova i slično. Autoru ovoga teksta se posebno dopala paralelna mogućnost apsolutnog i relativnog (prema adresi zadnje crtanje tačke) adresiranja grafičkih jedinica: svaki od ovih metoda ima svoje prednosti pa je lepo ako su obe prisutna.

U prospektima se obično navodi da BBC poznaje 16 boja, što je samo delično

tačno: različitih boja ima, u stvari, 8, dok se preostalih osam dobija kombinacijom osnovnih. Osim toga, nema smisla raditi u modu 2 (preveliko zauzeće RAM-a) — boje se opredeliti za mod 4 ili mod 5 i imati na raspolaganju 4 boje, lepu rezoluciju i više slobodnog memorijskog prostora. Ni u ovim modovima, međutim, nismo ograničeni na boje koje je Ejkorn izabrao, što je jedna od najlepših finesa na BBC kompjuteru: primenom naredbe VDU može se bilo koja od četiri standardne boje predefinisati u neku drugu. U modu 0, na primer, raspoložemo samo crnom i belom bojom, ali posle VDU 19, 1, 4, 0. bela boja postaje zelenja!

... i kontrola ekrana

Naredba VDU je neobično moćna, pa se oni koji saznavaju da je VDU X isto što i PRINT CHR\$(X), obično nemalo iznenade. Zaista, VDU je uvedeno jedino zbog jednostavnijeg i bržeg pisanja i uvek može da se zameni sa PRINT CHR\$. Kodovi 0—31 su rezervisani za specijalne namene od kojih je jedan (promena početnih boja) već potomenuta. Pomoću VDU se, da pomereni samo neke primene, jednostavno definisu „prozori“ za tekst i grafiku (na taj način korisnik može da radi na jednom delu ekrana i time ne kvari izgled ostatka na kom je, recimo, neka slika), menjaju grafički koordinatni početak, uključujući i isključuju kurzor, menjaju njegov oblik, aktivira i isključuje štampač... VDU 25, na primer, potpuno imitira već dovoljno moćnu naredbu PLOT, C, X, Y.

Mikroprocesor: 6502A

Klok: 2 MHz

Memorija: 32K ROM-a sa bežikom i operativnim sistemom i 32K RAM-a

Ekran: osam različitih grafičkih modova sa maksimalnom rezolucijom 640×256

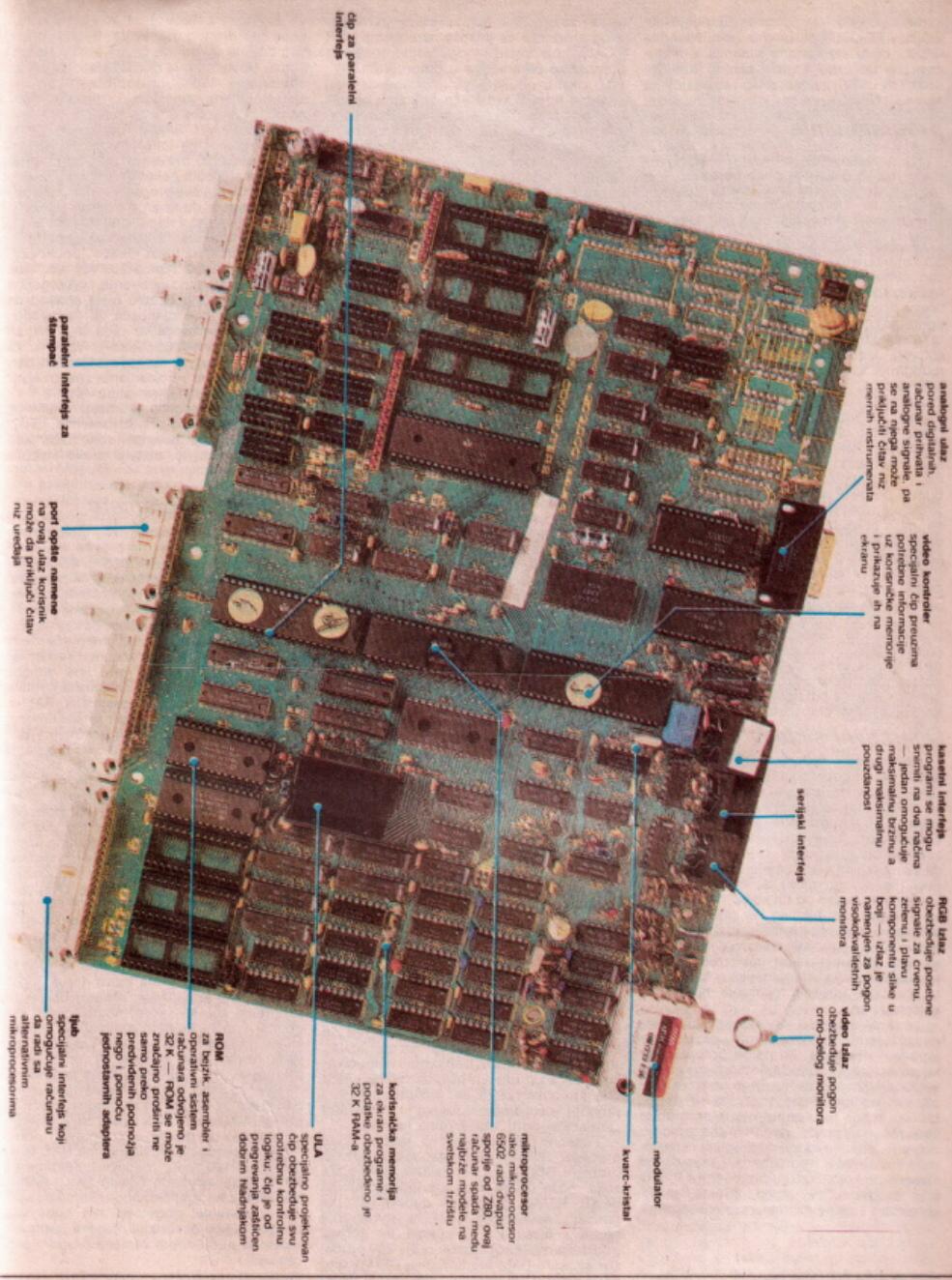
tačaka, do 32 reda sa po osamdeset znakova i do 16 boja

Interfejs: televizor, crno-beli i monitor u boji, disk-jedinica, printer, analogni ulaz, port opšte namene, port za vezu sa drugim mikroprocesorima

Ježici: bežik i asembler

Ježici na raspolaganju: lisp, fort, logo. Tastatura: 74 mehanička tastera (64 standardne namene i 10 za dodjeljivanje funkcija)

Dokumentacija: odlična, sa detaljnim prikazom rada i programiranja mikroprocesora 6502, ali previše napredna za početnike



Set naredbi za crtanje dopunjavaju GCOL i COLOUR za biranje boja. Naredba GCOL, osim standardne funkcije, omogućava da se kreće logički sabira (funkcije AND, OR i XOR) sa onima što već postoji na ekranu i tako formiraju vrio složene slike. Za igre i obradu teksta je korisno i definisanje karaktera koje se jednostavno postavi jednom od opcija naredbe VDU. Što je najlepše, predefinirani karakteri mogu da budu i na tastaturi pa se, na primer, srednja zagrada može zameniti našim slovom. t.

Sintesajzer bez tastature

Generalisanje tonova je jedna od najjačih karakteristika BBC-ja. Četiri kanala (jedan za generisanje buke) omogućavaju kako sintezator komplikovane muzike tako i proizvodjenje glasa (mada se u BBC ugraduje posebna opcija u vidu sintetizatora glasa koja uglavnom dodaje potreban softver). Jednostavne note se dobijaju na redovni SOUND koji ima četiri parametra: broj kanala, visinu note, trajanje i jačinu zvuka: SOUND 1—10, 101, 10 daje, na primer, četvrtinu note C u „Moderato“ ritmu. Pravi efekti se, međutim, dobijaju primenom naredbe ENVELOPE, pomoći koje definisemo odvojnicu tona koji najpre može da se pojavlja (modulacija može da bude i po frenkenciji, a ne po amplitudi), smanjuje, opet pojavlja i tako ponavlja dok se ne sinhronizuje sa tonom na nekom drugom kanalu ili, jednostavno, dok korisnik ne pritisne taster Escape.

Dok se naredbama, za generisanje slike zaista ne mogu uputiti nikakve zamerke, naredbe za generisanje tona su izuzetno moćne, ali je autor ovoga teksta stekao utisak da su neverovatno složene i da najveći deo korisnika (računajući i njega) nikada neće uspeti da iskoristi sve njihove potencijale. Trebalо je da budu formirane naredbe koje će biti prilagođene potrebama korisnika, a ne čipu 76489.

Operativni sistem

Konstruktori računara BBC su težili i uspeli da on bude neobično brz i efikasan. BBC B, premda ima star osmobilni procesor 6502, radi daleko brže čak i od većine šesnaestobilnih kompjutera, što samo potvrđuje tezu o ravnopravnosti hardvera i softvera pri konstrukciji računara. Jasno je da su morali da budu učinjeni neki kompromisi (ne mnogo njih!) da bi se dobilo u efikasnosti. Jedan od tih kompromisa je rad sa stringovima: ukoliko na početku rada program je definisao maksimalnu dužinu alfanumerika, utrošak memorije će biti minimalni. Računar neće imati ništa protiv produžavanja tih alfanumerika u toku rada, ali će tada utrošak memorije biti skoro katastrofalan.

Izkušnji korisnici će, svakako, poželeti da rade sa mašinskim jezikom i da iskoriste ugrađeni asembler. Firma se potrudila da taj posao što više olakša dobrom dokumentacijom. Ugradeni asembler je podržan bežikovim editorom, što znači da se asembleriske rutine pišu poput bežik programa: čim računar naide na otvorenu srednju zagrdu, počinje sa asembleriranjem i smješta dobijeni sadžaj u memoriju počevši od lokacije na koju ukazuju promenljive P%. U sors programu se, jasno, koriste labele, komentari i slične asembleriske naredbe, pa

je rad sasvim komforan. Pozivanje mašinskog programa se postiže naredbama USR i CALL koje imaju mnogobrojne opcije za prenošenje parametara u potprogram.

U okviru uputstva za upotrebu date su početne adrese svih važnijih rutina operativnog sistema. Tako je onaj ko piše mašinske programe pošteden brije u crtanju i pisanju po ekranu, radu sa kasetofonom, generisanju tonova... Namerno smo rekli „adresa rutina operativnog sistema“ a ne, adresna rutina bežik interpretatora — ova dve programa su, naime, sasvim odvojena!

Operativni sistem zauzima 16 Kb i obuhvata sve rutine neophodne za kontrolu računara. Bilo je mnogo verzija operativnog sistema, što stvara nebrojne probleme sa kompatibilnošću. Ako kupujete BBC B, obavezno provjerite da li je u njega ugrađena verzija operativnog sistema 1.20! Ukoliko računar, kada otukate „HELP“, odgovori nekim drugim brojem, slobodno potražite drugog prodavca!

BBC-jev bežik takođe zauzima 16 Kb i smesten je u jedan EPROM 27128 (sada se u računare stavljuju i ROM-ovi što, verovatno, znači da je razvoj operativnog sistema okončan). Neposredno pored ovoga EPROM-a se nalaze još tri podnožja — u njih korisnik može da smesti druge EPROM ili ROM čipove. Programme u njima, jasno, možemo i sami da razvijamo, ali će se većina korisnika opredeliti za gotove. Ukoliko, na primer, kupimo ROM sa paskal kompjajlerom, otukacemo „PASCAL“ po uključivanju računara i bežik će biti isključen, ne zauzimajući nikakav adresni prostor. U podnožju ne moraju da se stavljuju samo interpretatori novih programskih jezika — mnogi koristi procesor teksla ili neki drugi uslužni program kao alternativu bežiku. Ovi programi, pogotovo ako imate BASIC II (proverite to kako što ćete pritisnuti BREAK i otukati REPORT, ako računar ispisne (C) Acorn 1983, imate drugu verziju bežika i vaš računar poznaje mnoge naredbe koje nisu spomenute čak ni u vašem uputstvu za upotrebu!), mogu da pozivaju programi iz bežik interpretatora, premda proizvođač nije ponudio potrebnu dokumentaciju.

Drugi procesor

BBC B ima ogromne mogućnosti ekspanzije. Na njegovoj zadnjoj strani su, poređ standardnih priključaka za televizore, RGB i crno-beli monitor i kasetofon ugrađeni dvanaestobilni AD konvertor, RS423 interfejs (verzija čuvenog RS232 serijskog interfejsa) i priključak opcionog Econet interfejsa (dokupljuje se kao posebni dodatak i služi za povezivanje više BBC računara u mrežu). Sa donje strane kutije su portovi za printer, disk-jedinice (disk interfejs se posebno dokupljuje ali nije mnogo skup), „tube“ interfejs (za izuzetno brzu komunikaciju 6502 sa drugim procesorom), — 1 MHz bus za povezivanje sa laboratorijskom i drugom specijalnom opremom i standardni „korisnički port“ univerzalne namene.

Posebno je interesantno dodati drugi procesor obzirom da se tako rešava najveći problem BBC-ja — nedostatak RAM-a. Za sada može da se kupi još jedan 6502 sa 64 KB RAM-a (njegovim dodavanjem se gotovo neverovatno ubrzava izvršavanje bežik i mašinskih programa jer softver podržava multiprocesorski rad) i Z80 procesor sa istim RAM-om koji daje CP/M kompatibilnost. Čini nam se, ipak, da u ovom

momentu ne treba trošiti nekih 200 funti za kupovinu drugog procesora. Boće je sačekati do leta u nadi da će se, prema obećanju, pojaviti nova opcija: National semiconductors 16032 mikroprocesor sa šesnaestobilnim basovinom i tridesetdvobitnom arhitekturom koji će moći da ospušta 16 megabajta RAM-a. Opremljen njime, BBC B će postati toliko moćan kompjuter da će teško biti prevaziđen bilo čime osim cenom u sledećih nekoliko godina i, svakako, dovoljan za sve programe koji poželite!

Mogućnost dodavanja drugog procesora ima, na žalost, i negativnih posledica. Da bi programi bili kompatibilni sa prošireni kompjuterima, ne treba nikako „poukavati“ direktno u video memoriju, već se treba opredeliti za stalno pozivanje potprograma iz ROM-a. To ne bi bilo ništa strašno da firma Ejkorn nije zaključila da će najbolje spreći korisnike da to rade ako uskrati potrebne informacije. Tako u upustu za upotrebu nalazimo adrese sistemskih potprograma ali ne i objašnjenje „pokovanja“ video memorije, tačnu mapu RAM-a i sistemskih promenljivih, podršku direktnog pristupa portovima... Korisnicima ostaje jedino da sami pribave potrebne informacije što se, naravno, obavlja preko klubova programera.

Klubovi programera

Pominjanjem klubova programera dolazimo do poslednjeg (nikako ne najmanje važnog) poglavljaju: ove priče o BBC B — programske podrške. Za BBC je napisano zaista mnogo izvanrednih uslužnih, sistemskih i, naravno, zabavnih programa. Igre su neobično efektne i teško da se izgled ekranra igara za druge računare može meriti sa BBC-jevom. Za ovaj računar se najviše pišu akcione igre (brz je i ima odličnu grafiku i ton), dok je avantura znatno manje, najviše zbog nedostatka RAM-a. Logičkih igara takođe ima i to izvanrednih — šahovski program White Knight (Beli konj) je jedan od najboljih sličnih programa za stone računare uopšte! Od akcionih igara posebno vredi nabaviti Snappera (virfouzno urađena verzija pakmena), a od avantura — Snowball i, naravno, Hobbita.

Možemo da vam preporučimo dva komercijalna časopisa: A&B Computing (izlazi šest puta godišnje) i Acorn User, naročito ovaj drugi. Ipač, prava stvar je tek učlanjenje u kompjuterski klub Beebuz. Ovaj klub može da se podiže sa preko 30000 članova i predstavlja najveću sličnu organizaciju u Evropi. Izdaje časopis koji, u poređenju sa komercijalnim, izgleda sasvim neugledno, ali u njemu nema reklama a tekst iz koga se ne može naučiti ništa novo treba tražiti svećom. Što je najlepše, časopis objavljuje programe koji se skoro mogu oceniti kao odlični i ravnici komercijalnim. Sa drugim sličnim klubom koji se zove Laserburg do sada nismo imali kontakt ali može da se prepostavi da deluje na sličnim osnovama.

Šve u svemu, BBC B predstavlja izvanredan izbor za onoga ko za računar može da potroši 400 funti. Treba zaista biti kritički raspoložen da mu se nadu mane. I poređ toga, preporučujemo vlasnicima BBC B da ulože još 100 funti i nabave proširenje do 20 Kb, koje oslobada čitavih 32 Kb RAM-a bez obzira na grafički mod u kome se radi. Ovo proširenje, pogolovo ako bude više programa koji bi koristili njegove „usluge“, može da bude dobra zamena drugom procesoru!

Dejan Ristanović

commodore 64 računari

Commodore 64 se pojavio na tržištu gotovo bez imalo galame, toliko uobičajene za promocije novih računara, ali je na njemu odmah stekao veoma čvrste pozicije. Iako je praksa iznala na videlo nekoliko njegovih dosta neprljatnih slabosti, dobro znacići računarske tehnike smatraju ovaj računar jednim od najboljih modela u tzv. srednjoj klasi. Prava snaga ovog računara, ipak, dolazi do izražaja tek kada radi u sistemu sa svojim štampacem i svojom disk jedinicom i kada mu se dokupe neka proširenja. Među njih, na žalost, spada i malo bolji — bežzik.

Poznavalač računara odmah zapaža nedostatak kablova za povezivanje sa kasetofonom. COMMODORE 64 je konstruisan tako da traži svoj originalni kasetofon, kome može sam da uključuje i isključuje motor. Moguće je i povezivanje sa normalnim kasetofonima, što traži malo poznavanje elektronike, ali se tada moramo odreći programskih kontrola rada motora.

Kasetofon koji prodaje COMMODORE spaja se posebnim višežilnim kablom na računar. Standardnog je kvaliteta, sa brojačem. Brzina komunikacije računara sa kasetofonom je vrlo mala, otprilike kao kod ZX-81, ali je pouzdanost izuzetna. Svaki bajt se na traku snima dva puta i pri učitavanju se njihovim poređenjem konstatiše greška.

„Rastegljiva“ memorija

Po uključenju se inicijalizuju boje slova, ekranu i okoline ekranu redom na belu, svetlo plavu i tamno plavu, što se pokazalo kao vrlo iritirajuća kombinacija za oči, pa je redovan postupak da se ta kombinacija promeni recimo na zelenu, crvenu i crnu. Kvalitet slike je vrlo dobar, bez uobičajenog prelivanja koje se javlja kod spectruma.

Na ekranu se pojavljuje poruka: 64K RAM SYSTEM 38911 BAJTA SLOBODNO. Nije štamarska greška! Računar zališta ima 64K RAMA, ali je za bežzik dostupno svega 38911. Razlog ovome leži u konцепciji računara. U njega je ugradeno 8K bežzik ROM-a, 8K ROM-a u kome su rutine za rad sa diskom i printerom i 4K ROM u kome se nalaze 4 karakter seta: po jedan za velika i mala slova i dva graficka seta.

Na nekim mestima se ROM i RAM preklapaju. Tada je ROM kao važniji jedino dostupan. Za prosečnog korisnika, dakle, COMMODORE 64 ima pristupačno relativno malo memorije prema onoj količini koja je u njega ugradena. Ostatak memorije se može iskoristiti samo u mašinskim programima. Tada se ROM može programski isključiti i tako prći memoriji koja se nalazi „pod“ njim.

Uputstvo za programiranje je najslabiji deo ovog kompleta. COMMODORE je poznat po lošim uputstvima koja ne sadrže ni deo potrebnih i neophodnih informacija.



Nikoga ne ostavlja ravnodušnim: Commodore 64

Nameće se poređenje sa Spectrumovim uputstvom, koje je i po količini informacija i po kvalitetu daleko ispred COMMODORE-ovog. Korisniku koji želi da programira neophodna je knjiga „COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE“ u kojoj se nalazi neuporedivo više informacija o računaru. Cena je 10 funti Engleskoj.

Sam računar izgleda impozantno sa pravom tastaturom od 66 tipki, uključujući i izvođena 4 funkcionalna tastera sa desne strane, koji nemaju praktično nikakvu namenu osim što se programski može registrirati pritisak na njih. Sa desne strane računara nalaze se dva konektora za priključenje komandnih palica ili svetlosnolovke. Na zadnjoj strani nalaze se konektori za dodatni ROM, TV prikličak, video izlaz, Hi-Fi izlaz i ulaz (za ulaz nigde nema ni reći kako ga iskoristiti), za vezu sa diskom i printerom, za kasetofon i takozvani „USER“ port na koji se verovatno mogu priključiti neki dodatni uređaji i povezati paralelni printeri.

Najime, veza računara sa diskom i printrom je serijska, što znači da se prenosi bit po bit informacije, čime je izmena podataka između računara i spoljnih uređaja relativno spora. Program od 20K, na primer, učitava se oko 50 sekundi sa diska. U poređenju sa paralelnim diskovima, to je prava večnost. No, kako disk ne služi samo za snimanje i učitavanje programa, već i za rad sa podacima (RANDOM FILES, na primer) kada se može vrlo brzo prći bilo kom podatku na disku bez učitavanja prethodnih, ne treba ga se unapred odreći.

COMMODORE 64 je baziran na procesoru 6510 — nadgradnja poznatog procesora 6502. — iako 6510 ima nekoliko novih mogućnosti u odnosu na 6502, programiranje na njima je identično. Ovi procesori su jednostavniji od poznatog Z80 i mašinsko programiranje na njima je relativno lako. Za displej se brine poseban čip koji sa određenim lokacijama u memoriji uzima podatke i generiše sliku. Postoje dva načina rada ovog čipa: u niskoj i visokoj rezoluciji.

port za proširenja
omogućuje
priključivanje čitavog
niza uređaja — pinovi
se mogu programirati
pojedinačno da rade
kao ulaz ili kao izlaz

konektor za tastaturu

korisnička memorija
osim nivoa
obvezujuće 64 K RAM-a,
ali niv. sive
dostupno korisniku —
prikljuk inicijalizacije
računara „rezervne“
dobar deo za sebe

miokroprocesor 6510

generator zvuka
svi zvučni potencijali
smešteni su u jedan
jedini čip sa oznakom
6581

priključak za
kasetofon
port je prilagođen
specifičnoj
kasetofonu i na njega
se ne mogu direktno,
bez izmena, priključiti
klasični kasetofoni

sistemske program
i pored čitava trinčip
koja su odvojena za
njega, bežik ovom
računaru nije najjača
strana

serijski priključak
obezbeđuje serijsku
komunikaciju sa
periferijskim uređajima

priključak za televizor
istovremeno sa video
signalom prenosi se i
zvuk

kuditel za karticu
privata sistemske
programme, igre
koje se nalaze u
spoljašnjem ROM-u

prekidač
za napajanje

komandni portovi
primaju čitavu niz
uredaja od
svetlosne olovke do
paličica za igre

Procesor: 6510

Klok: 1 MHz

Memorija: 64 K — 20 K
ROM-a sa operativnim
sistemom i bežikom i
44 K RAM-a; kada se ne
koristi bežik, interpretator
dostupno je maksimalno
54 K RAM-a

Ekran: 25 redova sa po

četredeset znakova; visoka
rezolucija sa 320 x 200
tačaka; 16 boja
Interfejs: kasetofon, TV,
kerfild, monitor 232,
korisnički port
Jezik: bežik
Jezici na raspolaganju:
fort, komal, pilot, logo,
USCD, paskal i druge

verzije bežika
Tastatura: 62 mehanička
tastera standardne
namene i 4 višenamenske
Dokumentacija: slaba,
nerazumljiva i nedovoljna
Dimenzije:
404 x 216 x 75 mm
Cena: 229 funti

Grafika bez podrške

Niska rezolucija ili takozvani TEXT mod
je „normalnog“ načina rada. Tada su na
ekranu 25 linija sa 40 karaktera na svakoj. U
visokoj rezoluciji na ekranu su 320 linija sa
200 kolona. Treba odmah naglasiti da BA-
SIC 2.0 koji je ugraden u računar ne
podržava visoku rezoluciju, odnosno ne
postoje komande za crtanje tačaka, linija,
krugova i elipsa. Za to će se potruditi

SIMON'S BASIC koji se prodaje kao prošireni
ROM-a ili na disku, a u zadnje vreme
može se naći i na kaseti, po ceni od 190
DM. Osim tih, ovaj prošireni BASIC 2.0 ima
još mnoštvo korisnih novih komandi. Izme-
du ostalog, podržava muzičke mogućnosti
koje je osnovni bežik takođe zapostavio.

Korišćenje visoke rezolucije i muzičkih
pogodnosti moguće je, doduše, i na BASIC-
u 2.0, ali isključivo POKE naredbom, što je i
dosta nepraktično i dosta spor.

Jedna od dobrih mogućnosti BASIC-a
2.0 je kontrola (opet POKE naredbom)
takozvanih sprajtova (SPRITE). Sprajti je
„karakter“ veličine 24 sa 21 tačkice. Kada
se negde u memoriji nalaze 63 boja koja
određuju jedan sprajt, dovoljno je postaviti
jedan bit na 1 da bi se taj sprajt pojavio na
ekranu. Njegova X i Y koordinata se takođe
nalaze u memoriji i mogu se promeniti
POKE naredbom, čime se sprajti sele
na jednog na drugo mesto brzinom kojom se
izvršava POKE. Svaki sprajt može biti u
jednoj od 16 mogućih boja.

U svakom trenutku se može kontrolisati
do 8 sprajtova na ekranu. Sprajtovi mogu
biti povećani po X ili Y osi 2 puta, pri
čemu im se samo veličina udvostručuje.
Dok rezulutacija ostaje ista.

Za svaki sprajt se može odrediti prioritet
po kome se prikazuju kako za pozadinu
i za ostale sprajtove. Sprajti se mogu
stajati iza slova u TEXT modu, a može biti i
preko njih. U slučaju kada se dva sprajta
preklopne delom ili potpuno, sprajti se višim

prioritetom će biti prikazan, a onaj sa nižim će biti zaključen. Video čip će se sam javiti na pogodan način setovanjem odredenog broja ita do došlo da dodata dva spraja, što se može lepo iskoristiti u igrama.

Sprajtovi su normalno u dve boje, ali se, ako se žrtvuje rezolucija po horizontali, mogu praviti i u četiri. Grafika u visokoj rezoluciji je takođe samo u dve boje, ali i za nju vredi mogućnost 4 boje uz smanjenje rezolucija sa 200 na 160.

Za muziku se brine poseban čip koji, kao i video čip, sa određenih lokacija u memoriji uzima podatke za tonove. Tato je i moguće programiranje tih čipova običnom POKE komandom. Mogućnosti zvučnog čipa su upravo izvanredne i mogu se meriti sa mogućnostima skromnih sintezatora. Moguće je istovremeno korištenje tri zvučna kanala. Na svakom od njih za svaki ton se može odrediti njegov oblik pomoću 4 parametra (ADSR), a može se regulisati i jačina. Povrh svega, može se menjati i kvalitet tona određivanjem talasnog oblika od koga se sastoji zvuk. I na kraju, zvuk se šalje kroz antenski priključak u televizor, tako da mu kvalitet zavisi od kvaliteta zvučnika u televizoru. Hi-fi izlaz je mono, ali se može jednostavnim elektronskim kompjutrom simulirati i stereo.

Bledunjavi bejzik

BASIC 2.0 predstavlja klasičnu varijantu bejzika, uz vrlo mala odstupanja, i zato se programer, koji je navikao na komotonosti koju pruža Sinclairov bejzik u Spectrumu, u njemu sreću snalazi. To se naročito odnosi na rad sa slovnim varijablama, sa kojima se operacije vrše pomoću LEEFT \$, MID \$ i RIGHT \$.

Programe pisane na COMMODORE 64 je lakše prevesti na Sinclairov bejzik nego obratno. Dosta je nezgodna strana BASIC-a 2.0 što su nazivi varijabli ograničeni na svega dva karaktera i što se programska linija ne može sastojati od više od 80 karaktera uključujući i broj linije. Dobra strana mu je takođe „ekranSKI editor“ čime je ispravljanje programske linija znatno olakšano. Ako se zeli neka izmena u programskom listingu, dovoljno je dati LIST komandu, dotorati cursor na odgovarajuće mesto, izvršiti izmenu (obrisati, zameniti ili umetnuti) i pritisnuti RETURN, čime će u programske listing uместо stare biti unesena nova varijantna linija.

COMMODORE 64 ima nekoliko „nezgodnih“ osobina koje mogu često da zasmetaju. Jedna je da brojevi programskih linija na koje „skakuju“ GOTO i GOSUB komande moraju obavezno da postoje u programu — i računar ne podrazumeva sledeću liniju ako nedostaje ona na koju se „skakuje“. Druga osobina je da se svakim unosom bilo koje programske linije koja počinje brojem gube sadržaji svih varijabli. Ta osobina je naročito neugodna pri testiranju programa: primitate da program pogrešno radi, izvršite izmenu (izmenite liniju i pritisnite RETURN) i ne možete nastaviti rad programa sa CONTINUE jer ste upravo izgubili sadržaj svih varijabli. Jedino rešenje je ponovo RUN. U slučaju greške na kraju dužeg programa (kome treba, recimo, 10 minuta da stigne do nje) biće vam jasna važnost ove osobine.

BASIC 2.0 dopušta rad sa numeričkim varijablama čime imate mora završiti znakom %. Osepg brojeva se kreće između — 32767 do 32767.

Osim bejzika, postoje još neki programski jezici koji mogu raditi na COMMODORE 64. To su, pre svega, pilot, logo, fort, pascal i komal, pri čemu su, bez sumnje, najznačajnija poslednja dva: pascal je kompajler sa vrlo dobrim mogućnostima, a komal je donekle sličan bejziku ali ima mnogo moćnije komande.

Disk jedinica

Najpristupačniji (najjeftiniji) disk za COMMODORE 64 je VIC 1541 koji u Nemačkoj košta oko 700 DM. Njegov kapacitet na jednoj disketu je 170K i do 144 fajla odnosno programa. Za ozbiljno korišćenje računara neophodno je posedovanje bar jednog diska od četiri koliko se može istovremeno priključiti. Kasetofon je, jednostavno, suviše spor. Možda je za prosečnog korisnika dovoljno da nabavi program pomoću koga kasetofon snima i reprodukuje binarnom disku, ali ostaje činjenica da to još nije disk u pogledu ostalih mogućnosti.

VIC 1541 se sa računaram povezuje šestostolnim „DIN“ kablom koji se dobija uz disk. Na mrežu se povezuje svojim kablom. Diskete koje se uključuju u disk su standardne, veličina 5,25 inča (13 cm). Uz disk se dobija uputstvo koje je i ovog puta vrlo šturo i sa malo neophodnih primera. Uz to, ima i dosta štamparških grasača i nedorečenosti, što sve zajedno stvara loš utisak.

Na demonstracionoj disketti, koja se takođe dobija uz disk, nalazi se desetak programa koji su samo donekle demonstracioni a više su korisni u kasnijem radu sa diskom, i za to COMMODORE treba pohvaliti.

Njihovom analizom se može saznačiti mnogo onoga što nije rečeno u uputstvu. Na disketu se nalazi i WEDGE koji služi da bi se izbegla dosta glomazne komande koje su neophodne pri radu sa diskom. DOS (DISK OPERATING SYSTEM) je ugrađen u ROM računara i to predstavlja značajnu prednost u odnosu na druge sisteme, kod kojih se DOS nalazi na disketu i zauzima njen veći deo, a takođe zauzima i RAM u računaru kada se u njega učita.

U disku se nalaze procesor 6502, 16K ROM i 2K RAM. Dakle disk je dosta „inteligentan“. Tehnički je izvodljivo spoljno programiranje procesora u disku, čime on prelazi na mašinski program koji se nalazi u ta 2K RAM-a.

Matrični štampači

Od matričnih printerova koji se mogu povezati na COMMODORE 64 neophodno je pomenuti dva najbolja modela koji istovremeno imaju i dosta pristupačnu cenu.

Prije je VIC 1525 koji je, u stvari, prepravljeni SEIKOSHA GP100, u koji je ugrađen COMMODORE-ov interfejs. On se svojom cenu od oko 700 DM predstavlja vrlo dobar izbor. Poseduje nešto lošiji karakter set kada su u pitanju mala slova, ali je to sve vrlo čitljivo. Brzina štampanja mu je oko 50 karaktera u sekundi. Sirina papira koju može da prihvati je 10 lnča, što odgovara formatu A4. Papir mora biti perforiran po ravnama. Može se nabaviti i kod nas.

Drugi je VIC 1526 koji, verovatno, dolazi iz COMMODORE-ovog razvojnog biroa. Cena mu je oko 900 DM. Karakter set

Naredbe i funkcije

ABS	INPUT	RESTORE
AND	INPUT#	RETURN
ASC	INT	RIGHTS
ATN	LEFT\$	RND
CHR\$	LEN	RUN
CLOSE	LIST	SAVE
CLR	LOG	SGN
CMD	MID\$	SIN
CONT	NEW	SPC
COS	NEXT	SQR
DATA	NOT	STEP
DEF	ON	STOP
DIM	OPEN	STR\$
END	OR	SYS
EXP	PEEK	TAB
FN	POKE	TAN
FOR	POS	THEN
FRE	PRINT	TO
GET	PRINT#	USR
GET#	READ	VAL
GOSUB	REM	VERIFY
GOTO	RETRN	WAIT

mu je izvanredan, a brzina štampanja, zahvaljujući radu u oba pravca, nešto veća nego kod 1525. Sirina papira je ista kao kod 1525, ali ne mora biti perforiran. Može se štampati i na običnom A4 papiru, što predstavlja značajnu prednost.

Ova oba printerata rade sa 80 karaktera u redu i imaju mogućnost grafike. SIMON'S BASIC ima COPY komandu koja služi da sliku u visokoj rezoluciji iskopira na printer. U kombinaciji sa 1525 svu lepo radi, a sa 1526 dolazi do „raspadanja sistema“ — pritiski i računar se zaglavare pa se moraju isključiti oba COMMODORE-ja, svojevremenno, povukavši iz prodaje celu seriju 1526 jer je greška u ROM-u koji se nalazi u printeru. Najavljenja je nova serija sa istom označkom 1526 kod koje će greška biti ispravljena. Zato treba obavezno isprobati printer 1526 sa SIMON'S BASIC-om prilikom kupovine.

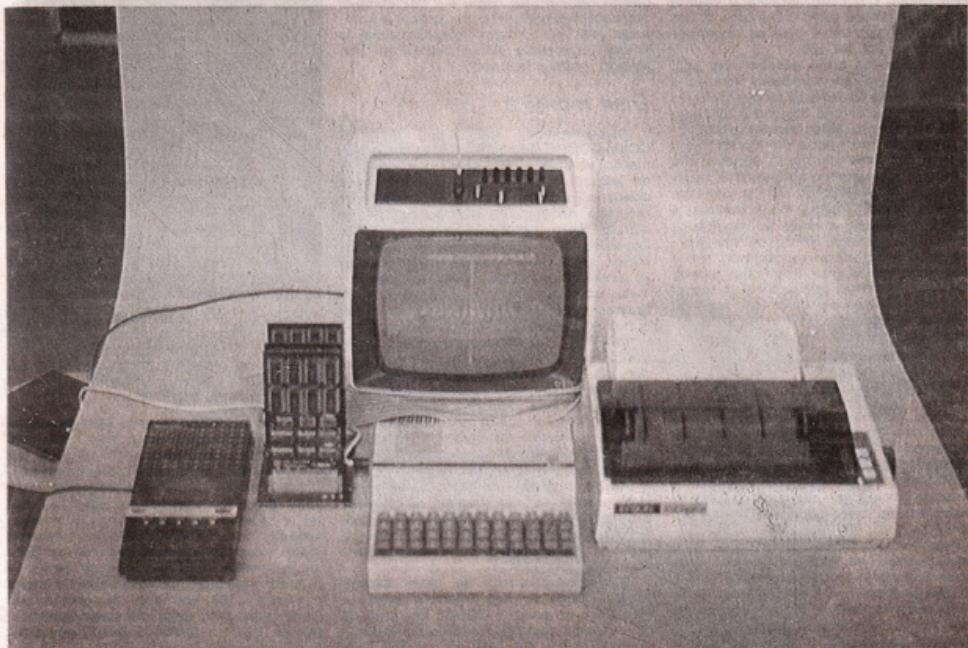
Programska podrška

Programska podrška je dosta solidna, ali uglavnom preovlađuju igre. Pošto je COMMODORE 64 sa diskom i printerom dosta pogodan za obradu teksta ili neki manji biznis, ima i takvih programa. Naravno, oni cenu dosta odskoču od programa za igru, čije su cene od 5 do 15 funti. Za sada je najuskljnij program SIMPLEX 64 koji je namenjen za unos i obradu dnevnih bilansne neke manje firme sa cenom od 172 funte.

Pošto je nekoliko programa za obradu teksta: WORDCRAFT 40, EASY SCRIPT, VIZAWRITE. Programi su profesionalno urađeni i imaju odlične karakteristike. Tekst se može držati na disku, izmene se mogu vršiti po želji itd.

Sa cenom u Nemačkoj od oko 700 DM, COMMODORE 64 je vrlo pristupljivo i našem džepu. Ako se kupujete kod nas preko oglasa, može se očekivati dupla ili veća cena. COMMODORE 64 ne preporučujem početnicima, za njega je bolji Spectrum. Za ozbiljnijeg korisnika, koji namerava da investira u disk i printer, COMMODORE 64 je vrlo dobar izbor. Ako ne želi da programira, imate podršku dobrobit (i skupit) programa. Ako želi da piše svoje programe, u tome neće daleko otiti bez SIMON'S BASIC-a.

Ivan Gerencir



U prvom izdanju „Galaksije“ „Računar u vašoj kući“ upoznali smo vas sa računarem **LOLA 8** i njegovim osnovnim mogućnostima. U međuvremenu on se dalje razvijao i široj oblasti primene i kroz ljubitelja.

LOLA 8 je nastao u računarskoj laboratoriji **LOLE RIBARA** jer se pojavila potreba za malim računarem koji bi se jednostavno programirao, a pored standardnih mogućnosti personalnih računara, uz dodatne module, koristio za obradu poslovnih i tehničko-tehnoloških informacija, po potrebi i u realnom vremenu.

BASIC je izabran kao najpriступačniji jezik za ovakvu vrstu računara. Ipak je realizovan veoma moćan set instrukcija, naročito u oblasti matematičkih operacija, sa tačnošću do 10 decimala i velikom brzinom izvršenja.

LOLA 8 je, i bez posebne reklame, stigao u mnoge škole, fabrike, institute, projektnе biroe, laboratorije, domove zdravlja, kancelarije. Veličko interesovanje i mogućnosti primene u raznim oblastima kao i iskustva u korišćenju računara **LOLA 8** su uslovili dalji rad na njegovom razvoju.

Pokazalo se da su 64 reči RAM memorije nedovoljni za obzišnjije programe na **BASIC**-u. RAM memorija se proširuje modulima od po 16K. Poredjenja radi, u 6K reči RAM memorije može da stane program od oko 150 linija (dužine 1 reča ekran), 650 različitih promenljivih, 100 vektora dimenzija 10 elemenata ili pak

230 nizova prosečne dužine 40 karaktera. Dodavanjem samo jednog modula od 16K dužina programa može da bude i do 500-600 linija; čak 2500 promenljivih, 330 vektora dimenzija 10 ili 850 nizova.

Većina korisnika personalnih računara zna da je njihova slaba tačka nepouzdanoć zapisa na kasetofonu, koja zavisi od kvaliteta i tipa korisnog kasetofona, čistote magnetne glave, kasete i mnogih drugih faktora. Rešenje sprege za kasetofon **LOLE 8** omogućuje pouzdano snimanje i čitanje programa za gotovo sve kasetofone u komercijalnoj upotrebni. Novost su i 3 nove BASIC komande:

DSAVE, **DLOAD** i **DVERIFY** koje omogućavaju snimanje i čitanje vrednosti promenljivih na kasetofon, a vezane su za primene u poslovnoj obradi informacija.

Logičan nastavak razvoja računara **LOLA 8** je modul za priključenje štampača. Programi pisani rukom su često nepregledni i nečitljivi, pa se zbog mnogobrojnih izmena dešava da mislimo da smo uneli jedno, a u memoriji računara je nešto drugo. Stampađanjem sadržaja programa ovi problemi nestaju, ali je još značajnija mogućnost štampanja rezultata obrade informacija.

Većina štampača koje srećemo na inostranom tržištu ima paralelni ulaz/izlaz (tzv. Centronics standard) i svi oni mogu da se priključe na **LOLA 8**. Situacija na domaćem tržištu štampača je dosta loša. Postoji nekoliko domaćih

proizvođača (TERA, TRS), ali su njihove cene za sada previsoke.

Zato je razvijen modul koji omogućava da se električne mašine za pisanje mogu koristiti umesto štampača. Postoji nekoliko tipova ovih mašina na domaćem tržištu **OLIMPIJA**, **ROBOTRON-EI Niš** ... Njihova cena je znatno pristupačnija, a veliki broj firmi ih već posedeju. Naravno ove pišaće mašine ne mogu da pruže sve pogodnosti kao savremeni štampači, ali su za većinu primena viši nego dovoljne.

LOLA 8, sa dodatnim modulima, omogućava izgradnju malih sistema obrade informacija za različite primene: evidencije, knjigovodstvo ili praćenje i upravljanje jednostavnim procesima.

Nadamo se da vas ovim poslednjim nismo prepričali. Hteli smo da vam ukazemo na još neke mogućnosti primene **LOLE 8**. U **LOLI RIBARU** svakog dana preko 30 učenika III razreda OC „Beogradski bataljon“ radi na računaru **LOLA 8**, radaju se ideje za nove igre, svira Bahova svita u H moli ... drugim rečima stišava nova generacija koja zna što je računar.

O saradnji **LOLE 8** sa srednjim i osnovnim školama i drugim mogućnostima računara **LOLA 8**, recimo ton generatora, u nekom od sledećih nastavaka Galaksijinog časopisa „Računar u vašoj kući“. Dofle, za svu obaveštenja obratite se na tel. 570-227 ili pišom na adresu industrija mašina IVO LOLA RIBAR, za računara **LOLA 8**, 11250 BEograd-ŽELEZNIK.

LOLA RAČUNARSKI SISTEMI

Industrija IVO LOLA RIBAR jedan je od najznačajnijih jugoslovenskih proizvođača računarskih sistema namenjenih upravljanju u industriji. Familija LOLA računara, izgrađena na osnovi standardizovanih komponenti, odlikuje se fleksibilnošću, pouzdanosti u radu, kompaktnom dizajnu i jednostavnostu eksploatacije koja ne zahteva posebno obrazovanje u oblasti računara.

Svi predstavnici LOLA familije industrijskih računara se mogu preko standardnog modula spregnuti sa centralnim računarcem, omo-

gućujući tako izgradnju distribuiranih sistema upravljanja za sve proizvodne celine i centrale fabrike.

Izkušeno više generacija stručnjaka u projektovanju i razvoju upravljačkih sistema i koncenciji najavšnjemstvenih i najnovijih tehnologija garantira su kvaliteti ILR proizvoda u oblasti industrijske elektronike. IVO LOLA RIBAR danas raspolaže timovima stručnjaka koji rade na razvoju novih računarskih sistema, obuhvaćajući tako svu liniju polja automatizacije u industriji.

LOLA SISTEMI NUMERIČKOG UPRAVLJANJA

U ožujku programskog upravljanja LOLA RIBAR predstavlja sistem za upravljanje numeričkog upravljanja LOLA 30 (za trutnice, glodalište i centne obrade), LOLA 305 (za trugove) i LOLA 42 (za upravljanje industrijskim robotima). Svi sistemi su prevedeni za upravljanje složenim kretanjima: max. 6 rotacionih ili linearnih osa simultano.

LOLA 30 CNC (computer numerical control) obezbeđuje sve standardne funkcije najmodernejših alatnih mašina:

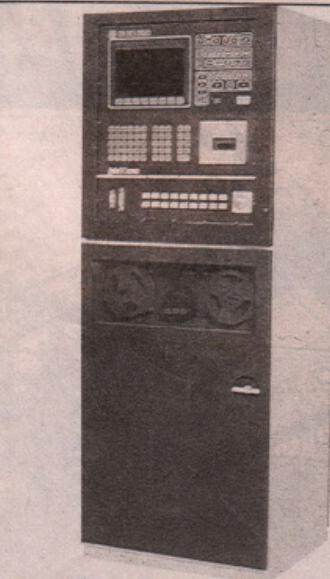
- Prilagodjenje na različite tipove mernih sistema (sa rezolucijom 0.001 ili 0.01 mm)
- 3D linearna interpolacija • kružni interpolacije (bez drugih ograničenja) • unutarnje podatci preko slatkih butonika • optičko-fotonske jedinice ili alfanumeričke tastature
- Tabele korektura za max 96 alata • Fikani začeci, začeci i trugove • Gledanje i rezanje načina • Asistencija, interaktivno i mešovito programiranje • Parametarsko programiranje •

Eksterna jedinica (31 cm) omogućava prikazivanje aktuelne koordinatne pozicije svih osa, aktivnosti i četiri sledeća bloka programa, kao i svih potrebnih informacija, upozorenja i uputstava za operatore.

Integralni dio LOLA 30 sistema je programabilni automatski PA 512, na bazi koga je realizovana fleksibilna sprege sa mašinom.

Specifičnost LOLA 42 CRC (computer robot control) sistema je za upravljanje industrijskim robotima, namenjenim za programiranje u režimu obučavanja. Operator dovršava osnu po osu robota u željenu poziciju koja se automatski memorira. Ovakvo uneti program se slijedi i obavljati u automatskom pojeđinacno u cikličnom režimu.

Osim više tipova instrukcija kretanja, upravljački sistem prima i specijalne instrukcije synchronizacije radi radnog na temeljnom okruženju. U tom smislu na raspolaženju konzum stoji 32 ulazno-izlazni signala kojima se robot povezuje sa drugim robotima ili mašinama.



LOLA PROGRAMABILNI AUTOMATI

Najstariji primjeri u svim granama privrede (eksploatacija i prečišća ruda i nafta, pregradnja i proizvodnja hrane, uspeće) naša je gama LOLA programabilnih automata. Podela u **PROGRAMABILNIH AUTOMATIMA** izaziva revoluciju u upravljanju mašinama i procesima. I pored sve raznolikosti primene, programabilni automati su u svim granama privrede konkretnim upravljačkim zadatku i „raslu“ zajedno sa njegovim eventualnim proširenjem.

Obvezujuća realizacija „stepne“ logike sa novom savremenom tehnologijom, brojčanog i vremenske funkcije, kao i obradu numeričkih podataka. Za rešavanje problema regulacije i upravljanja razvijen je specijalan modul PID regulacije koji prima standardne algoritme obradbe do 32 regulacione pete.

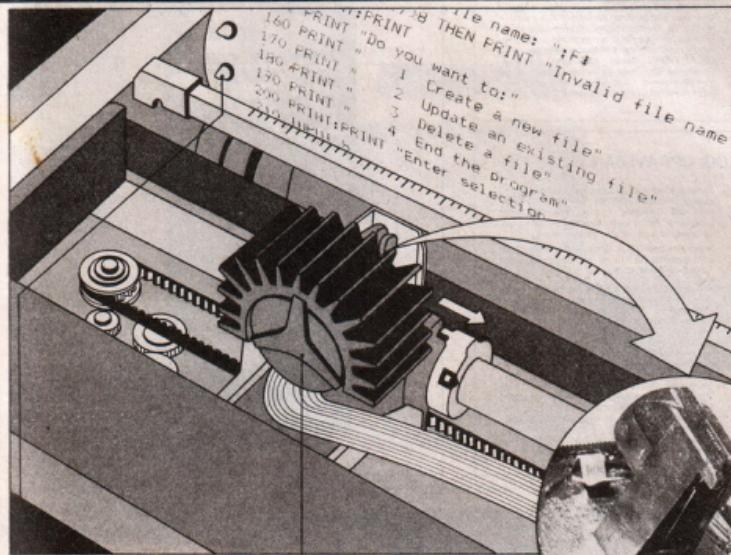
Poseban uređaj za razvoj i testiranje upravljačkih sistema na bazi programabilnih automata — PROG PA omogućava unotenje

upravljačke logike u vidu standardne, stepenice funkcije sa uključenom prikazivanjem na ekranu. Uneti program se može sačuvati u EPROM memoriji, na magnitnoj kaseti, ili se, prikupljajući, stampa može dobiti u formi dokumenta. Pomoću specijalne prikazivanje omogućava operatoru da u svakom trenutku prati na ekranu odvijanje procesa u realnom vremenu. Svi aktivirani elementi „relejnih“ funkcija prikazani su inverzno. Time se vrlo poslovno da se otkriju nepravilnosti u radu uređaja svodi na minimum.

LOLA programabilni automati (mini PA, PA 256, PA 512 i MPA) koncipirani su tako da se opremanjem elektronskih efekata zadovolje širok opseg kategorije primen u različitim konfiguracijama sa max. 32 ulaza/izlaza: do složenih upravljačkih zadataka koji podrazumevaju ne samo veliki broj ulazno/izlaznih signala (po nekoliko hiljada), već i visok nivo numeričkog procesiranja.

štampači u akciji

Periferijska
oprema



Matrični štampač: Za oblikovanje karaktera sistem koristi matricu sa tačkama; štampanje slova se vrši kolonu po kolonu, pomeranjem glave sa vertikalnim nizom iglica: tačkasta struktura otisnutih slova ne može se sakriti čak ni kod najkvalitetnijih štampača

U specijalnom izdanju „Računari u vašoj kući“ posvetili smo dosta prostora karakteristikama štampača i kriterijumima kojima se treba rukovoditi pri njihovom izboru. Zbog nedostatka prostora (da, i 100 strana nije mnogo) nismo mogli da prikažemo detaljnije nijedan model i tako, na praktičnim primerima, ilustrujemo savete koje smo davali. Ovoga puta ispravljavamo taj propust i opširnije prikazujemo modele za koje smatramo da su vredni pažnje domaćih ljubitelja računara.

Seikosha GP100

Seikosha GP100 je veoma popularan štampač sa najboljim odnosom mogućnosti/cena. Radi se o matričnom tipu koji proizvodi slova na matrići 6×7 (zapravo 5×7 , jer poslednja kolona mora da ostane slobodna zbog razmaka između slova). Ova matrična je, u osnovi, mala i nije dovoljno za definisanje naših latiničnih slova, posebno velikog „S“. Ipak, uz malo „umetničkog“

dara i malo više mašte, postiže se zadovoljavajući kvalitet teksta.

Štampač piše na kompjuterskom tabulirajućem papiru (ima ga u prodaji i kod nas) čija je širina promenljiva (maksimalna 10 inča, ili oko 25 centimetara). Traka je, na žalost, uvozna, ali ne košta više od 5 funti (900 dinara — što znači da se bez problema naručuje preko pošte) i veoma je kvalitetna i trajna — dobar deo rukopisa ovog specijalnog izdanja je uraden na Seikosha štampaču, a traka je i dalje kao nova.

Štampač ima četiri različita seta karaktera koji se biraju pomoću mikroprekidača na zadnjoj strani njegove štampane ploče. Ako, na primer, želimo da koristimo nemačka slova umesto engleskih, jednostavno ćemo odvrnuti četiri zavrtnja; skinuti kutiju i pomeriti prvi mikroprekidač u položaj ON. Osim nemačkog, možemo da izaberemo američki, engleski ili švedski set karaktera; pošto se štampači ne prodaju u Jugoslaviji. Seiko (firma koja proizvodi štampače Seikosha; pomnimo je zato što mnogi, računajući i autora ovoga teksta, teško uočavaju u vezi između ova dva imena dok im se ne skrene pažnja) očigledno nije smatralo za

potrebu da uvrsti i naš set karaktera u standardnu opremu.

Slova su normalno visoka 2,82 mm i široka 2,11 mm, ali se mogu odabratи i dvostruko veća — tada u jedan red staje samo 40 slova (red ima 80 normalnih slova), što je prilično pogodno za naslove i podnaslove. U jednom redu se može naći kombinovani tekst pisani normalnim i uvećanim

silovima.

Oznaka štampača, GP100, predstavlja skraćeniku od Graphic Printer što, naravno, znači da postoji mogućnost crtanja slika. Korisnik ima pristup svakoj od 480 tačaka u redu (80 karaktera po 6 tačaka) definisnjem sopstvenih kolona. Pretpostavimo da, na primer, želimo da štamparamo karakter poput onoga na slici. Za svaku njegovo kolonu sabiramo brojeve koji predstavljaju svaku tačku. Za prvu tačku dodajemo $2^1 = 1$, za drugu $2^2 = 2$, za treću $2^3 = 4$, za četvrtu $2^4 = 8$ i tako dalje. Za poslednju tačku u koloni dodajemo $2^6 = 64$. U našem primeru brojevi koji predstavljaju karakter bi bili 7, 42, 112 i 0. Njih ćemo, naipre, povećati za 128 (jedna od specifičnosti GP100a), a za tim saopštiti računaru, jedan po jedan. U

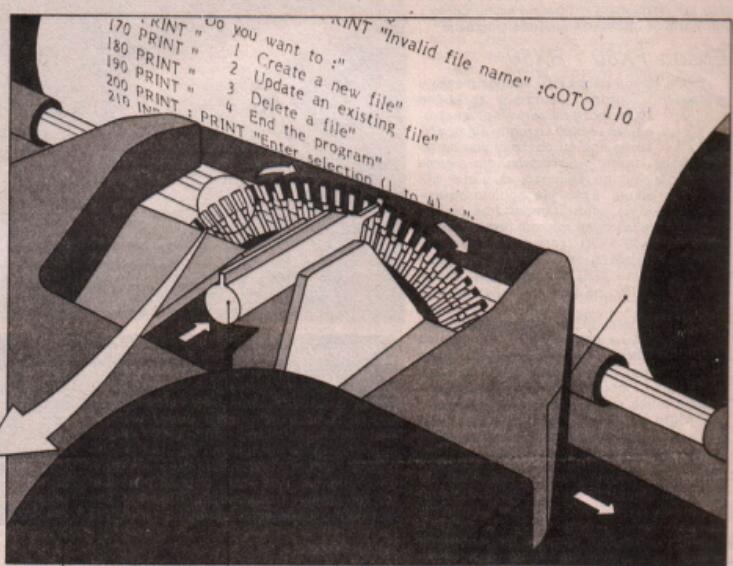
Stampač, bez sumnje, predstavlja najvažniju periferijsku opremu svakog kućnog računara koji svome vlasniku predostavlja nešto više od igre. Bez njega se ne može zamisliti obrada teksta, rad sa datotekama, pa čak ni iole ozbiljnije programiranje. Na tržištu se trenutno mogu naići dva tipa štampača — maticni (dot-matrix) i sa lepezom (daisywheel). Prvi omogućuju definisanje posebnih karaktera (kao što su naša latinična slova) i crtanje i rade samostalno i veoma brzo. Drugi su previše spori, zahtevaju mehaničke intervencije na lepezi da bi se uvela i naša slova i stalno opsluživanje papirom, ali imaju izvanredan otisak. Kvalitet otiska maticnih štampača je dovoljan za ličnu upotrebu — listinzi, datoteke, dokumentacija, evidencije — ali na njemu ne možete napisati tekst za novine, seminarški ili diplomski rad, pa čak ni poslovno pismo. Tu postoje samo dve mogućnosti: štampač sa lepezom ili — ručni rad na klasičnoj pisacoj mašini.

Štampač sa lepezom:
Lepezu na čijim se
špicama nalazi po jedno
slovo okreće veoma
precizan motor; kada dođe
u poziciju za štampanje,
slovo otključuje na hartiju
elektromagnetski udarač;
otiskuje slovo na hartiju
mašina i štampača sa
lepezom nema značajnijih
razlika u kvalitetu otiska,
ali je rad sa štampačem
neupoređivo udobniji, jer
je pre štampanja moguće
sve ispravke na tekstu
izvršiti elektronskim putem



okviru jednog reda možemo slobodno da
mešamo obične, dvostrukе i karaktere sači-
njene od kolona koje smo definisali bez
ikakvih ograničenja, što nam omogućava
pisanje normalnog latiničnog teksta, pa i
grčkih slova potrebnih za neko naučno
štivo.

Selkosa GP100 ne spada u brže štam-
pače — ispisuje samo 50 karaktera u
sekundi, što bi bilo dovoljno da je
štampač „parametan“: da piše i dok se glava
vraca u početni položaj. To, na žalost, nije
slučaj, pa ispisivanje jedne strane teksta ume-
da potraje prilično dugo. Za većinu primena
ovo nije velika smetnja. Posebno ograniči-
enje u brzini predstavlja okretanje doboša
koje je sporije nego kod mnogih drugih
printera — prelazak na sledeću stranu ume-
da, i da potraje. GP100A je, uz to, i
prilično bučan štampač, što neće biti previ-
še omiljeno u vašem domu.



Autoru ovoga teksta ne smeta ni jedno
od ovih ograničenja jer mu je jasno da je
ono rezultat težnje da štampač bude jeftin.
Pri izradi softvera koji je smешten u ROM
se, međutim, skrilašlo daleko više nego što je dopustivo. GP100A je, tako, ostao veoma
„neinteligentan“ uređaj: ne zna kada nala-
zi kraj strane, pa će pisati po perforaciji
ukoliko se programer ne pobrine da pre-
broji ispisane redove, ne omogućava sof-
tversko smanjenje broja slova u redu niti
promenu razmaka između redova, pravi
neobično mnogo problema, kada programer
poželi da piše dva puta preko istog
teksta (radi podvlačenja važnijih poruka, na
primer) i, što je posebno zamorno, ima
nepriyatno dugo pamćenje: ako u jednom
trenutku nesmetnostno pošaljemo nekoliko
karaktera u njegov buffer, ne postoji regula-
ran način da ih odatle obrišemo pre nego
što red bude štampan (isključivanje štam-
pača iz napajanja, jasno, ne smatram
regularnim rešenjem, jer uvek može da
izazove probleme kod drugih istovremeno
prikupljenih uređaja). Ne bi bio nikakav
problem da je firma upotrebljava neki od
ionako neiskorišćenih kontrolnih karaktera
za reinitializovanje štampača; ovakav pro-
ust, jednostavno, navodi na pomicanje da
firmi nije želela da ugrozi prodaju svojih

daleko skupljih modela i da je zato GP100A
ostao bez boljeg ROM-a.

Uz štampač se dobija knjižica od 30
strana kao jedina dokumentacija. Za isku-
sivog korisnika računara dokumentacija je
sasvim dovoljna, a obuhvata čak i sve što je
potrebno za gradnju interfejsa za lične
potrebe. Za početnika je knjižica, međutim,
previše lakonski pisana: čini nam se da će
neko kome je GP100A prvi štampač utrošiti
časove, a možda i dane da shvati ono što je
moglo da bude jednostavno objašnjeno.
Žalosna je činjenica da oni koji pišu uputstvo
za korišćenje računara izbegavaju da
se pozabave radom sa štampačima (svaka-
ko zato što ima mnogo nekompatibilnih
tipova), računajući da će ovaj najpopularniji
periferijski uređaj biti posebno dokumentovan,
a oni koji pišu uputstva za upotrebu
štampača računaju sa tim da je onome ko
posedi računar dovoljna dokumentacija
koju već ima. Malo više standardizacije i
sadržine postigelo bi početnike mnogih
glavobolja.

Uve u svemu, Selkosa GP100A je ume-
reno dobar izbor i verovatno može da se
pohvali najboljim odnosom mogućnosti/cena.
Začudo, nije postigao preveliku popu-
larnost (možda će uskoro biti bolje?), što se
odrazilo na slabiju softversku podršku. Sof-

Prvičlan za hobiste: tako nema baš najbolji kvalitet otiska, štampač Seikosha 100, ipak, u odnosu na cenu, predstavlja veoma dobar izbor.

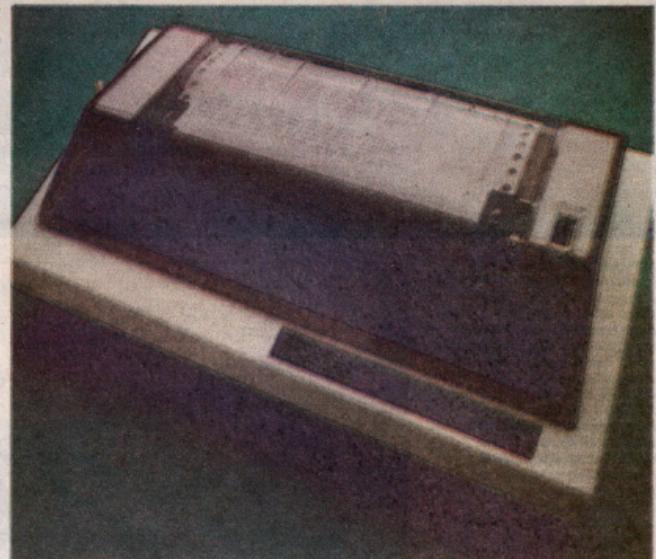
verska podrška za štampanje? I ona ponekada može da bude važna: ako imate BBC B ili Electron, na tržištu i po časopisima ćete naći na gomilu programa koji omogućavaju obradu teksta na Epsonu, kopiranje sadržaja ekranu na Epsonu, stvaranje uticaja o bojama na Epsonu i mnogo što-šta slično. Ukoliko ne nameravate da sami pišete ovakve programe (nije teško ali oduzima vreme), zaboravite na Seikoshu i počnite da razmišljate o Epsonu kome ćemo i mi, naravno, posvetiti sledeće poglavije.

Epson FX80 i RX80

Firma Epson danas drži vodeće mesto u proizvodnji i prodaji štampača za kućne računare. Za to može da zahvali prvenstveno svome modelu MX80 i drugim štampačima iz MX serije koji su postavili standarde za mnoge druge proizvođače. No, vreme čini svoje, pa su štampači iz serije MX postali preskupi. Epson je odlučio da pripremi dovoljno jeftin štampač koji će dobrostojno reprezentovati renome firme i nastali su FX80 i RX80. FX80, zapravo, ima sve dobre osobine štampača iz MX serije, a košta samo 380 funti (320 bez interfejsa), što je sasvim dovoljno da vam ga najtoplje preporučimo.

FX80 je, naravno, matrični štampač koji formira slova na matrići 6 x 8, što znači da slov g, i slična ne bivaju deformisana kao kod Seikoshe. Set karaktera je više nego univerzalan i obuhvata preko 250 simbola. Odakle tolika slova? FX80, osim standardnih ASCII karaktera, ima i nekoliko vrsta slova, kao i karaktere od interesa za razne jezike (naših slova, na žalost, ni ovde nema). Šta se vrsta slova tiče, na raspolažanju su normalna (80 u redu), povećana (40 u redu), kondenzovana (137), kondenzovana-povećana (68), elitna (96) i elitna-povećana (48 slova u redu). Sve ove kategorije slova su, uglavnom, objašnjene samim imenom osim jednih — elitnih. Elitni slovi bi trebala da se što više približe kvalitetu koji se postiže na pisacu mašini, odnosno da ulepšaju izgled dokumenta. Zašto bi neko onda korisio bilo koja druga slova? Jednostavno zato što mnogi vole da dokument ispisani na štampaču izgleda u pravom smislu kompjuterski.

Epson FX80 je jedan od vrlo retkih štampača koji omogućavaju automatsko podvlačenje teksta i takozvani „double strike“ mod. O čemu se radi? Ponekad nam je potrebno da posebno istaknemo deo teksta ili, jednostavno, dobijemo kontrastniji dokument koji će ići u štampu. Tada ćemo izabrati opciju koja štampaču nalaže da svaki red štampa po dva puta, pri čemu će drugi trag biti za pola tačke pomeren u odnosu na prvi (sve labele u Računarama 1 su, na primer, ovako kucane). U tom slučaju štampač, zbog povećane preciznosti, ne štampa u dva smera, što značajno umanjuje brzinu koja u normalnim uslovima dostiže više nego dovoljnih 160 karaktera u sekundi. Postoji još jedan način da sa razlogom prepolovimo ovu brzinu: ukoliko se odluči-



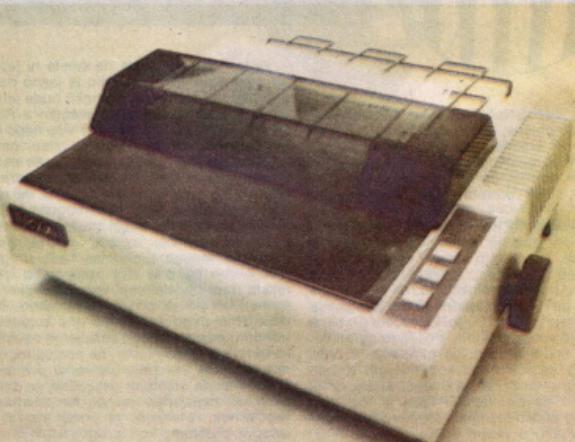
Seikosha GP 100A

Tip: matrični
Matriča: 5 x 7
Interfejs: centronics
Memorija: 256 znakova
Linija po redu: 6
Karakter po mreži: 6-12 znakova

Brzina štampanja: 50
znakova u sekundi
Karakter u redu: 80
Broj kopija: 1
Podvlačenje: ne
Isticanje: da
Proporcionalni razmak: ne
Karakter po mreži: 6-12 znakova

Dvosmerno štampanje: ne
Logično traženje: ne
Maksimalna širina papira: 10
Pomeranje papira: traktor
Dimenzije: 42,7 x 23,8 cm
Masa: 4,5 kg
Cena: 247-funti

Blok grafika: ne
Grafika visoke rezolucije: da



„Kraljevska loza“ među matričnim štampačima: Epson MX 80 može da zadovolji potrebe i najrazmaženijeg ljubitelja kućnog računara ali je za njega potreban više nego dubok džep.

Epson FX 80

Tip: matrični
Matrič: 9 x 9
Interfejs: centronics
Memorijska: 256 karaktera
Linija po inču: 6,8
Karakter po inču: 5, 8,2,
5, 10, 16, 5

Brzina štampanja: 80
znakova u sekundi
Karakter u redu:
maksimalno 132
Broj kopija: 3
Podvlačenje: da
Isticanje: da
Proporcionalni razmak:
opcionalni
Blok grafika: ne

Grafika visoke rezolucije:
da
Dvosmerno štampanje: da
Logično traženje: da
Maksimalna širina papira:
25 cm
Pomeranje papira: traktor
Dimenzije: 40,6 x 38,1 cm
Masa: 5,5 kg
Cena: 400 funti

mo za 80 karaktera u sekundi, štampač će pisati daleko tiše, što je korisno ako radiš u nekoj kancelariji ili, jednostavno, ako nam ukucavši ne trpe buku.

Za one koji su navikli na skupe pisaće mašine, kod kojih su slova proporcionalna razmaka („m.“, na primer, ne zauzima onoliko mesta koliko „i.“), FX80 nudi opciju koja omogućava čak i takvu pogodnost. Da stvar bude još lepsi, bilo koja druga opcija (npr. veličina slova) može da bude izabrana zajedno sa proporcionalnim razdvajanjem slova.

Onima za koje ni ovoliko slova nije dovoljno, na raspolažanje su naredbe za definisanje karaktera. Možemo, na primer, da predefinisemo neki nepotreban ASCII karakter i dobijemo naše slovo „Č“ koje će, samim tim, biti dodeljeno nekom tasteru (ako imamo BBC ili Electron, efekat ćemo kompletirati predefinisanjem tog karaktera u setu slova računara, pa ćemo na ekranu videti baš ono što treba da se pojavi na štampaču). Interesantno je, međutim, zapaziti još nešto: posle malo eksperimentisanja sa definisanjem karaktera, primičemo da naša slova izgledaju ružnije od originalnih. Da li to znači da imamo manje dizajnerskog talenta? Možda, ali se i Epson pobrinuo da tako bude: kod slova iz ROM-a se ne adresiraju samo tačke već i prostori između njih (po „polu tačke“) što, jasno, poboljšava izgled dokumenta. Ta mogućnost, na žalost, nije ostavljena i korisnicima. RX80 ne omogućava definisanje karaktera.

Epson FX80 je izvanredan za one koji žele da crtaju. Moguće je, naravno, kontrolisati svaku tačku u redu, ali koliko ima takvih tačaka? Skoro da bismo mogli da kažemo „koliko poželite“. U normalnoj rezoluciji na svakih 8 inča (20 cm) ima 480 tačaka, u dvostruko rezoluciji na taj prostor stane dvostruko više (960), ali je na raspolažanju i takozvana „četverosrstruka tačnost“ koja nudi 1920 tačaka! Ovakva rezolucija se približava ploterima i omogućava, na primer, generisanje bar koda koji bi „razumeo“ HP41C ili neki drugi računari opremljeni optičkim čitacem. Osim ovih modelova, na raspolažanju su i CRT grafiki I i II (640 odnosno 720 tačaka na 20 cm) koje su idealne za vlasnike BBC računara, jer se podudaraju sa maksimalnom rezolucijom na ekranu računara. Za šta oni mogu da koriste veće rezolucije? Vešti programeri su već sastavili rutine koje senče slike, čineći tako da se stekne ulask u bojama koje su bile zastupljene na ekranu koji je dampaovan.

Za crtanje je vrlo korisno što može da se izabere razmak između redova i to sa velikom preciznošću (od šestine do dvesta sedesetina inča). Može, osim toga, lako da se izabere položaj leve i desne marge (samim tim i broj slova u redu), pomeranje

početka teksta od vrha i dna strane i mnogo šta-šta drugo. Vidimo da, za razliku od Seikošes, Epson FX80 prepoznaće kraj strane, ali on radi i više od toga: prepoznaće i zvučno signalizira trenutak kada je vaša zaliha papira pri kraju. U istom trenutku štampač „koci“ kompjuter i čeka da stavite još papira. Jednostavan pritisk na taster ON LINE će, zatim, naložiti računaru da nastavi sa slanjem karaktera. Osim tastera ON LINE, na kućištu printera su i tasteri LF (pomeranje papira za jedan red) i FF (pomeranje na početak sledeće strane), koji imaju i sekundarnu ulogu: ako pritisnete LF dok uključujete štampač, bice obavljena automatska dijagnostika (korisno kada kupujete štampač), a ako u tom trenutku držite pritisnute FF i LF, svi podaci koje kompjuter bude slao bice štampani kao heksadekadne vrednosti (čak i kontrolni karakteri).

Osim tastera na kućištu, na samoj štampačnoj ploči printera postoji čitava serija mikroprekidača. U njihovu pomoć možemo da biramo stanje u kome će naš štampač biti kada ga uključimo. Ako, na primer, stalno koristimo „elitna“ slova, nema nikakvog smisla da posle svakog uključivanja sačemo čitavu seriju kontrolnih karaktera – jednostavno ćemo pomeriti jedan preklopnik. Slične se odnosi i na druge opcije koje se u toku radi softverski menjaju, kao i na setove slova koja biramo (RAM u kome se nalaze karakteri koje smo mi definisali biva, jasno, izbrisani kad god isključimo štampanje).

Krajnje je vreme da kažemo da Epson RX80 piše na tabulir hartiji, ali da njegova nešto skupljija verzija, FX80, piše na standardnom papiru A4 formata. Dobra investicija? S vremenom na vreme svakako vredi imati ovako nešto, ali nam veruješ na reč da nije nimalo lako stalno nadzirati štampač i svaki minut ga opsluživati novim listom papira. Zato je za duže tekstove bolje nabaviti bank post tabulir papir (ima ga ponekad i kod nas).

Epson FX80 u standardnoj verziji ima Centronics paralelni interfejs, ali se za njega može dokupiti i RS232C. Obzirom da bater ovog štampača ima čitava dva kilobajta, Centronics interfejs će omogućiti brži prenos i manji gubitak računarevog vremena na beskorisno čekanje. Ovaj bater, na žalost, ne može da se koristi ukoliko definisemo neke karaktere (RAM je ograničen), što znači da je za većinu primena u

kojima su potrebna slova Č, Č, Š i računar čekati da bude ispisano svaki slovo, da će i RS232 biti više nego dovoljno.

Treba stvarno biti veliki probaći da bi pokušamo, ipak, i to. Pre svega, uputstvo prilično opširno, ali ne i naročito napisano nedostaju upravo one informacije koje ljudstvo računara koji je kupio svoj prvi štampač bile od nekih koristi. Teškoćama snažalačem potpomaže i nekoliko, do sada u operativnom sistemu štampača: neki kodovi u nekim slučajevima (kada smo, na primer, odlučili da pišemo nekim posebnim slovima) jednostavno ne rade ono što bi trebalo da rade, dok drugi izazivaju neprajepitne efekte. Moguće je, međutim, da se „bagovi“ već otklonjeni ili da će uskoro biti.

Sve u svemu, Epson FX80 predstavlja izvanredan izbor kako za one kojima je potreban brz printer sa velikom grafičkom rezolucijom tako i za one koji žele kvalitetan izlazni dokument. Kod njega, međutim, treba obratiti pažnju na jednu stvar: štampanje je set naredbi koje podržavaju grafičke veoma moćan. Starijim računarima (TRS 80, na primer) i novim računarima sa skromnom grafikom (mislimo, naravno, na računar „galaksija“) može da zasmetala činjenica da FX80 ne podržava blok grafika, pa će težava direktno prenosa sadržaja ekran-a na štampač. Blok-grafika se, jasno, može zameniti definisanim karakterima, ali takvo rešenje zahteva učitavanje posebnog programa u računar posle svakog uključivanja štampača. Osim toga, njegov ROM od 16 Kb je kompaktan (EPROM 27182), pa nije lako pronaći način da se u njega upišu neka naša slova. Oba ova nedostatka oslobođen je treći štampač o kome ćemo govoriti: Okidata Microline 80. Microline 80 je stariji, sporiji i daleko slabiji štampač od FX80 ali, kao što ćemo videti, predstavlja prihvatljivu alternativu za mnoge prime.

Oki Microline 80

Microline 80 je, da to kažemo na samom početku, prilično star štampač, što, u kompjuterskom svetu, svakako nije naročito preporuka. Iako se, jasno, radi o štampanju opšte poznate koje može da bude opremljen nekim od standardnih interfejsa. Microline 80 (u daljem tekstu, jedino radi kratčeg pisanja, MC80) specijalno je pripreman za računar TRS 80 (odatle verovatno oznaka). Za potencijalne vlasnike štampača u Jugoslaviji MC80 ima nekoliko velikih „plusova“: piše, pre svega, na običnom kompjuterskom papiru (o čemu što smo rekli), mogu da se nabave kod nas, ali je uvek dostupnije tražiti standardni papir. Koristi običnu traku za pisacu mašinu i, najzad, njegova „blok grafika“ je kao stvarno naša računara „galaksija“.

Set karaktera MC80 je više nego kompletan: pored velikih i malih slova, brojeva i standardnih ASCII specijalnih simbola, tu

Oki Microline 80

Tip: matrični
Matrič: 5 x 7
Interfejs: centronics
Memorijska: 256 znakova
Linija po inču: 6, 8
Karakter po inču: 5, 10,
16, 5
Brzina štampanja: 80
znakova u sekundi

Karakter u redu:
maksimalno 132
Broj kopija: 1
Podvlačenje: ne
Isticanje: ne
Proporcionalni razmak:
ne
Blok grafika: da
Grafika visoke rezolucije:
ne

Dvosmerno štampanje: ne
Logično traženje: ne
Maksimalna širina papira:
9,5 inča
Pomeranje papira: valjak,
traktor, frikciono
Dimenzije: 34,2 x 24,5 cm
Masa: 6,5 kg
Cena: 242 funte

Izvaničan kvalitet otiska: Štampač sa lepezom Juki 6100 predstavlja jedan od najboljih modela u svojoj klasi.

su 64 simbola od kojih je neke vrlo teško protumačiti (gleđajući ih, autor ovoga teksta dolazi na smešnu pomisao da su dizajneri zeleli da omoguće igru „vešanje“ (hangman), pa su se potrudili da pripreme karaktere u vidu čovečuljka kome nedostaje pola tela, samo ruka... kao i karakter koji bi lepo izigravao scenu sa kraja igre). Niko se, naravno, neće buniti zato što mu je set karaktera povećan, ali vlasnici MC80, za razliku od vlasnika GP100A i FX80, nemaju mogućnost da se dopune sopstvenim karakterima, što je ozbiljan nedostatak: šta ako su nam potrebna naša latinična slova? Odgovor na ovo pitanje možemo da damo samo onima koji poseduju programator EPROM-a: dovoljno je da izmenite deo štampačevog generatora karaktera, kodiranje je vrlo jednostavno, praktično kao kod računara „galaksija“ (pogledajte „Računare 1“ i tekst „Prvih nekoliko mikrosekundi“).

Pored kompletног seta karaktera, MC80 ima i nekoliko veličina slova: u redu možemo da štampamo 132, 80 ili 40 karaktera (tako zvane „duge linije“) odnosno 105, 64 ili 32 karaktera („kratke linije“). Da li je ovakva raznovrsnost potrebna? Možda i nije, ali su na ovaj način pomireni mnogi standardi: štampanje 132 ili 80 karaktera u redu, kopiranje TRS-ovog ekranra (64 slova u redu) i, naravno, listanje „galaksijnih“ „Spectrumu“ itd. programa (u doba kada je MC80 projektovan, bilo je relativno malo računara koji imaju 32 karaktera u redu, ali se dalekovidost firme Oki danas pokazuju u punom sjaju). Brzina pisanja retko prelazi 50 karaktera u sekundi (zavisi od broja slova u redu i razmaka redova i može da bude najviše 80 CPS) što je umereno spor ali privihatljivo.

MC80 nema grafiku visoke rezolucije, što znači da korisnik jedino može da kontrolise grafičke blokove koji su veliki koliko šestina svakog karaktera. On je, dakle, više nego neuspodoran za vlasnike BBC-ja, Electrona, Commodora 64 i za one vlasnike Spectruma kojima štampač treba da služi za kopiranje slika sa ekranra.

I pogorje toga što mu nedostaju kodovi za kontrolu grafičke visoke rezolucije, MC80 ima više kontrolnih kodova od Seikosha GP100A, koji je u ovom prikazu već karakterisan kao „neinteligentan štampač“. Znati li to da je MC80 „pametniji“? Izgleda da jeste: iako ni on ne može da prepozna kraj strane (zar je toliko teško smestiti u RAM štampanja jedan brojač redova?), MC80 ne-pogrešivo detektuje trenutak kada je list običnog papira pri kraju (tada za njim ne sledi još jedan list) kao i trenutak kada rotira kompjuterskog papira spadne na malu debeljinu. Osim toga, na njemu se nalazi taster pomoću koga štampač prelazi u mod OFF LINE, tj. kada kompjuter mora da čeka na našu intervenciju da bi nastavio da štampa. Tačka mogućnost nedostaje Seikoshi — kada je baš neophodno da štampač „zakoči“ računar, pribegava se njegovom isključivanju čime se, jasno, gubi sve iz internog baterija.

Uz MC80 se dobija dobro uputstvo na 56 strana; u njemu se relevantne činjenice



Juki 6100

Tip: lepeza
Matrica: —
Interfejs: centronics
Memorija: 2000 znakova
Linija po inču: 6.8
Karakter po inču: 10. 12.
15
Brzina štampanja: 18
karaktera

Karakter u redu:
maksimalno 165
Broj kopija: 4
Podvlačenje: da
Isticanje: da
Proporcionalni razmak:
da
Blok grafika: ne
Grafika visoke rezolucije:
ne

Dvosmerno štampanje: da
Logično traženje: da
Maksimalna širina papira:
Pomeranje papira:
frikcionalno
Dimenzije: 454 x 520 x 151
Masa: 14 kg
Cena: 400 funti

ponavljaju i po nekoliko puta, ali je to po našem mišljenju, velika pogodnost za početnike kojima je često teško da prate terminologiju, pogolomu kada je tekst pisan na stranom (engleskom) jeziku.

Sve u svemu, stekli smo utisak da je Microline 80 dobar štampač, ali to ne znači da vam ga prepotro preporučujemo: njegova cena od oko 270 funti nije toliko manja od Epsonovih 320 funti (obe cene bez uračunatog interfejsa) da bi se čovek oprimio da ga tako slabiji uredaj. Došlo je vreme računara sa visokom grafičkom rezolucijom i šteta je da tu rezoluciju ne prati i printer!

Štampači sa lepezom

Štampači sa lepezom (daisywheel) do skora su bili preskupa alternativa brzim i pogodnim matrčnim modelima. U poslednje vreme su razvijeni jeftiniji „margarita štampači“ (popularni naziv za štampače koji proizvode tekst okretanjem specijalne lepeze sa slovima) koji daju izvanredan izlazni dokument bolji nego u električnoj pisačkoj mašini. Ovi štampači danas koštaju svega 300—500 funti, što nije mnogo viša cena od one koju placamo za neki matrčni štampač. Pitanje je, dakle, za kakav se printer odlučiti?

Prije nego što pokušamo da odgovorimo na njega, prikazaćemo, bar ukratko, nekoliko štampača sa lepezom: Brother HR15, Silver Reed EX44, Juki 6100 i Dataline 1200 DL.

Pomožalo je parodoksalno reći da najbolji kvalitet pisanih teksta daje najjeftiniji od pobrojanih štampača — Silver Reed EX44 koji košta 300 funti. Za ovaj novac se, zapravo, dobija pisača mašina koja je opremljena interfejsom za povezivanje sa računarnom, što je idealno za one koji se tek privikavaju na korišćenje tekst-procesora.

EX44 je, na žalost, prilično spor štampač — njegovih 12 karaktera u sekundi postiže i preosećan daktilograf. Ovako mala brzina je rezultat kompromisa mogućnosti/cena: EX44 ima bafer od samo 15 bafta, štampa samo u jednom smeru i nema mogućnosti logičkog traženja. Sve u svemu, odličan štampač za one kojima se mnogo ne žuri!

Juki 6100, kao i mnogi dobiti i jeftini uređaji, dolazi iz Japana. Njegova cena, kada je opremljen interfejsom i potrebnim kablovima, nešto je manja od 400 funti. Štampač je brži od prethodnog (18 znakova u sekundi) što je dopunjeno mogućnošću dvosmernog štampanja i interne baferom od 2 kilobajta. Osim ispisivanja normalnih slova, Juki 6100 omogućava proporcionalno razmicanje slova i slične pogodnosti koje imaju daleko skuplji štampači. Na žalost, mehanizam koji okreće traku nije najbrže rešen, pa se mastiljava traka ponekad zamotava oko njega. Druga loša strana Jukija je što je neprijetno bučan, što će tek delimično rešava poklopcom koji se uz njega dobija.

Cini nam se da je najbolji izbor Dataline 1200 DL. Očekuje se da će njegova cena biti znatno povoljnija (sada 450 funti) što bi.

U pripremi

RAČUNARI 3

u vašoj kući

• HAKERSKI MANIFEST

šta piše u jednom papiru koji tajno kruži među programerima sveta — šta jednom programeru dolikuje a šta ne i kako prepoznati pravog programera

• RAČUNARI U IZLOGU

noviteti na evropskom tržištu — šta mogu, koliko koštaju a koliko vrede, kako se nabavljaju

• RAČUNARI U AKCIJI — DISK-JEDINICE

disk jedinice su postale dovoljno jeftinje i za kućnu primenu — kako izbor primeriti svojim potrebam a mogućnostima računara — karakteristike, kvalitet, cene, izbor, nabavka

• INTERFEJSI

izbor i povezivanje interfejsa za štampač sa računarem ponekad nije nimalo jednostavna stvar — kako se pišu softverski interfejsi — prikaz svih interfejsa za računar Spectrum

• NEKE NOVE IGRE

prikaz najnovijih igara za računare Spectrum, Commodore 64 i Electron

• SOFTVERSKA TRPEZA

upredni tabelarni prikaz sistemskog i uslužnog softvera za računare

Spectrum, Commodore 64, Electron i BBC B

• SPEKTROTEKE

kako koristiti kućni računar za obradu podataka — prikaz svih datoteka za Spectrum — koja datoteka u kojoj prilici

• ŠKOLA AVANTURISTIČKIH IGARA

kako se igraju a kako pišu igre-avanture

kombinovano sa dobrim karakteristikama ovoga štampača, moglo da mu pomogne da postane bestseler. Maksimalna brzina štampanja je pristojnih 25 karaktera u sekundi, pri čemu štampač piše u oba smere sa mogućnošću logičkog traženja. Bafer od 512 bajtova i Centronics interfejs omogućavaju povezivanje sa većinom postojećih računara: na štampanoj placi se nalazi određeni broj prekidača koji omogućavaju izbor brzine prenosa.

Čak i onima koji planiraju nešto da crtaju, 2000 DL može da posluži umesto matričnih štampača: njegova rezolucija od 165 karaktera na 11 inča (oko 28 cm) i minimalni razmak linija od samo 1/48 inča (0,021 cm) nisu karakteristike koje se tako lako mogu zanemariti.

Pre nego što odlučite na kupovinu štampača sa lepezom, morate ozbiljno da razmislite o jednom problemu: retko ko koristi štampač samo za pisanje teksta na stranom jeziku, što znači da printer bez slova Č, C, S i Ž nema mnogo smisla. Kod ovih štampača slova ne mogu da se definisu, što znači da je potrebna „jugoslovenska lepeza“ koju, koliko znamo, niko ne može da vam isporuči. Jedan servis za precizniju mehaniku u Beogradu je kráće vreme pouzdao da prepravi lepeze za naše uslove, ali su rezultati bili nezadovoljavajući. Nepriznatno, zar ne?

Zamislite, osim toga, da ste napisali neki tekst od stotinak strana, uredili ga pomoću

računara i poželedi da se ispišete na štampanju. Štampač sa lepezom, jasno, pišu na običnoj hartiji, što znači da će vam biti potrebno nekoliko časova sedenja poremeđi i menjanje listova svaka dva minuta! Ništa, međutim, nije savršeno: štampači sa lepezom jedini imaju poseban kvalitet otiska za profesionalnu primenu.

Hewlett-Packard 2225

HP 2225 predstavlja prvi komercijalno najavljeni „inkjet“ štampač. Odakle mu ovo ime? Radi se o potpuno novom metodu štampanja teksta koji se zasniva na izbacivanju kapljice mastila na papir. Svakog slovo je, naime, predstavljeno maticom od 11×12 tačaka. U glavi štampača nema pokretnih delova: to je jednostavno rešetlo od 132 otvora iza kojih se nalaze mikro rezervoari sa mastilom i grejači. Kada štampač treba da ispiše neko slovo, neki od grejača zagrevaju mastilo koje se širi i kapljica izleće. Zatim nastupa praktično trenutno hlađenje, iz većeg rezervoara u glavi biva povučeno novo mastilo, a glava se, za to vreme, pomeri do sledeće pozicije na kojoj treba pisati.

U čemu je prednost ovakvog metoda? Pre svega, u glavi nema pokretnih delova, pa je štampač jeftin (500 dollara, što je neverovatno niska cena za firmu kao što je Hewlett-Packard) i, što je možda još važnije, praktično nečujan. Ostvarena je, osim toga, i prenosivost — dimenzije HP 2225 su 8,9×29,2×20,6 cm, što znači da lako staje čak i u manju tašnu. Napajanje je na struju ili baterije, a firma tvrdi da jedan set baterija omogućava štampanje oko 200 (!) strana, što znači da je potrošnja više nego mala.

• OPERATIVNI SISTEM/ADRESNI DEKODERI

prvi nastavak softversko-hardverske škole za buduće graditelje računara koja vas korak po korak praktično uводи u hardversku i programsku konцепциju kućnog kompjutera

• MAŠINAC ZA POČETNIKE

paralelna škola mašinskog programiranja u stripu za procesore Z80 i 6502 sa ilustracijama, rečnikom menonika

• NAŠI PROJEKTI

— trotonski generator zvuka za računar „galaksija“

— uređaj za automatsko učitavanje EPROM-a prilikom uključivanja računara ili nakon hardverskog reset-a

• COMMODORE 64

specijalan prilog za vlasnike računara Commodore 64 koji se i kod nas polako probija ka vrhu rang-liste najpopularnijih računara

• BIBLIOTEKA PROGRAMA

novi programi za računare „galaksija“, Spectrum, Commodore 64 i BBC/Electron

• NIZ NOVIH TEKSTOVA O RAČUNARIMA „GALAKSIJA“ I SPECTRUM

Ukoliko želite da vod sada obezbedite svoj primerak, Računara 3“ možete naručiti na adresu: „Galaksija“, 11000 Beograd, Bulevar vojvode Mišića 17 — isporuka pouzećem

HP 2225 može da se pohvali i mnogim drugim dobrim karakteristikama: karakteri mogu da se definisu na matici 11×12, u jedan red stajaju 40, 71, 80 ili 142 slova, što znači da je graficka rezolucija 192×96 tačaka po inču (=2,5 cm) ili 1296 tačaka u redu (može, dakle, da se stampa slika sa standardnog televizora — ako se raspolaže pogodnom opremom). Set karaktera obuhvata preko 150 znakova zajedno sa slovima iz danske, škotske, engleske, finske, francuske, nemacke, italijanske, norveške, španске i švedske abecede.

Kada su svojevremeno najavljeni u časopisima, mnogi su smatrali da „inkjet“ štampači nemaju sanse jer im je potreban skup „reprematerial“! Hewlett-Packard je pokazao da nije baš tako: HP2225 piše na specijalnom ili običnom papiru (u oba slučaja kvalitet je sasvim zadovoljavajući, s tim što se na specijalizovanom eliminira raziljanje mastila, pa je kvalitet uporediv ne samo sa matričnim nego i sa štampačima sa lepezom), dok se umesto trake menjaju glave kada se mastilo iz nje istroši. To se događa posle ispisanih 500 strana (rukopis solidne knjige), a jedna glava košta svega 8 dollara — ne skuplje nego traka za matrični štampač!

Cini nam se da HP2225 predstavlja izvanredan izbor, naročito za one koji su odlučili da kupe prenosiv i moćan računar kao što su HP75 ili novi HP71B (HP2225 može da se nabavi sa Centronics, HP IL ili HP IB interfejsom koji je učinut u cenu od 500 dollara). Kada u bliskoj budućnosti neka druga firma proizvede sličan štampač, njihove cene će opasti, pa „inkjet“ modelima predviđamo veliku budućnost!

Dejan Ristanović

**Knjiga koja vas popularno uvodi
u svijet kompjutora!**

Peter Laurie

KOMPJUTOR U KUĆI

priručna enciklopedija

**Najtraženija engleska knjiga o kompjutorima
— sada i na našem jeziku!**

Iz sadržaja: — OPĆENITO O KOMPJUTORU — PROGRAMIRANJE — PRIMJENA U PRAKSI: Poslovni programi, kompjutor i slikarstvo, simulacije, manipulacija slike, kompjutorska animacija, kompjutor koji govorи, kompjutorska glazba, robotika, roboti u industriji, programiranje prilagođeno za kućne kompjutore SINCLAIR itd. — POGLED U BUDUĆNOST — RJEČNIK POJMOVA — SHEMATSKE SKICE I UPUTE itd.

Temeljni priručnik za svakog koji namjerava nabaviti kompjutor za kuću, školu ili za poslovne potrebe

Bitne osobine:
jasnoća i preglednost teksta,
stručna pouzdanost objašnjenja
i bogatstvo korisnog
ilustrativnog materijala

Preplatna cijena: 2.200.- dinara
Prodajna cijena: oko 3.000.- dinara
Izlazi iz tiska: 24. oktobra 1984.

Narudžbenica Računari II

izdavačkoj jedinici „Cankarjeve založbe“
Ilica 26, 41000 Zagreb

Ovime naručujem primjeraka knjige KOMPJUTOR U KUĆI (izdanje na srpskohrvatskom) po preplatnoj cijeni od 2.200.- dinara. Što ću platiti u pet mješjevnih rata po 440.- dinara.

Knjigu ću primiti po izlasku iz tiska i uplati cijelokupne preplatne cijene.

Ime i prezime naručitelja

Adresa

Vlastoručni potpis

Napomena: preplatiti se možete i na telefonski broj (041) — 432-325



PARTNER - SARADNIK U SAVREMENOM POSLOVANJU

ISKRA DELTA uvodi na jugoslovensko i strano tržište mikroračunar **PARTNER**, rezultat sopstvenog razvoja. Ne nudimo vam samo mašinsku opremu, nego i savremeno integrisano rešenje vašeg finansijskog, skladišnog i materijalnog poslovanja. Međutim, i pored svega toga, primena je tako jednostavna da možete da je savladate u jednom danu.



galaksija

u školi

Računari
i obrazovanje

... u rukama Mandušića Vuka
svaka puška biće ubojita!
(Njegoš: Gorski vijenac)

Brojni podaci

Obrazovanje u oblasti programiranja može biti zasnovano na izradi programa koji rade sa različitim objektima: brojevima, tekstom ili grafičkim objektima. Međutim, ipak se najčešće, za ovu svrhu, koriste brojevi. U bežiku za „Galaksiju“ brojevi se u internom kodu registruju u pokretnom zarezu. Međutim, za cele brojeve iz intervala –999999, 999999 korisnik ima utisak da radi sa celim brojevima, jer ih unosi kao cele, a računar ih izdaje u istom obliku. Za ostale brojeve koristi se pozicionalni ili eksponentijalni zapis brojeva.

Na ovaj način „Galaksija“ omogućuje izučavanje svih elementarnih struktura, nad brojnim podacima, koje se javljaju u programu: linjske, razgranate i cikličke, kao i korišćenje potprograma. Posebno treba istaći da razgranata struktura, koja često u bežiku dovodi do loše čitljivosti programa, ovdje u narednoj IF ima opciju ELSE, što daje bolje strukturirane programe. Tako, program

```
10 INPUT X  
20 IF X < 0 PRINT X * X: ELSE PRINT X  
30 STOP
```

izdaje kvadrat unetog broja ako je uneti broj manji od nule, odnosno izdaje uneti broj ako ovaj nije manji od nule.

Od strukturiranih brojnih podataka postoji samo jednodimenzionalni niz, čiji broj elemenata zavisi od rasploživog memorijskog prostora. Dakle, posebno se ne di- mensioniše. Za obuku u programiranju ovo se ne može uzeti kao neki bitan nedostatak, jednostavno iz razloga što je jednodimenzionalna struktura podataka, zapravo, prirodno najbliza fizičkoj strukturi memorije. Višedimenzionalne strukture, kao što su tabele i matrice, mogu se obrazovati nad jednodimenzionalnim nizom i predstavljaju lepe zadatke za vežbe iz programiranja.

Od elementarnih funkcija postoji samo celobrojna funkcija (INT), dok se za trigonometrijske, eksponentijalnu i logaritamsku funkciju mogu napisati odgovarajući potprogrami na bežiku. Pisanje ovih potprograma na raznim obrazovnim nivoima može imati različit smisao, od programiranja potprograma i izgradnje biblioteke potprograma, do numeričkih problema aproksimacije algebarskim izrazima u kojima se javljaju samo aritmetičke operacije sabiranja, oduzimanja, množenja i deljenja.

Primena računara na rešavanje numeričkih problema može se u školama lako

demonstrirati od najnižeg do najvišeg obrazovnog nivoa. Ovdje se kao poseban izazov javlja traženje efikasnijih numeričkih algoritama, što „galaksija“ omogućuje merenjem vremena rada programa, korišćenjem programske kontrolisanog časovnika. Evo jednostavnog primera: sastavite program koji određuje broj šestocifrenih prirodnih brojeva koji zadovoljavaju uslov da su im zbirovi tri cifre veće i tri cifre manje težine jednak. Ako program zasnujete na ideji da se ispituju svi šestocifreni prirodni brojevi tražeći na računaru oko 15 časova. Međutim, ostavljamo čitaocu da nade efikasniji algoritam koji će se na računaru izvršiti za samo 75 sekundu!

Ukratko, bežiku na računaru „galaksija“ pruža mogućnosti obuke u programiranju, kao i u pripremi računara u oblasti jednostavnijih naučno-tehničkih izračunavanja.

Rad sa tekstom

Praktično, pri svakoj upotrebi računara korisnik ima potrebu da u manjoj ili većoj meri radi sa tekstom. Naravno, postoje takve primene računara u kojima se isključivo radi sa tekstom, kao što je izučavanje prirodnog jezika, prevođenje sa jednog jezika na drugi, u velikom broju informacionih sistema i sl. Rad sa tekstom treba da omogući: • prirodniji dijalog korisnika sa računaram, • uredanje teksta, • pretraživanje teksta i • unošenje i izdavanje teksta.

U interaktivnom radu sa računaram korisnik najčešće odgovara na poruke koje izdaje računar. U ovom slučaju, tekst služi da podrži komunikaciju računara sa korisnikom. Na primer, u jednom programu za obračun kamate računar izdaje pitanje:

KAMATA U PROCENTIMA? 30.

a korisnik unosi vrednosti kamate, u ovom primeru 30%. Izdavanjem pitanja korisniku se olakšava korišćenje programa, jer bi bez pitanja mogao da pogreši u redosledu unošenja podataka, što bi dovelo da pogrešni rezultati u programu. Naravno, isto tako je važno, a možda još značajnije, da računar uz brojeve, koji su rezultati rada programa, izda i tekst koji objašnjava rezultate. Na primer, već pomenuti program za obračun kamate može izdati rezultate u obliku:

```
PRETHODNO STANJE: 20000  
KAMATA: 6000  
NOVO STANJE: 26000
```

Ovo je očigledno znatno čitljivije nego da su izdata samo tri broja 20000, 6000 i 26000. „Galaksija“ u potpunosti omogućuje unošenje i izdavanje teksta u smislu podrške dijalogu između korisnika i računara.

Neki mikroračunarski sistemi su orijentirani na rad sa tekstrom, pa omogućuju

uredjenje i pretraživanje veće količine teksta. „Galaksija“ nije takav sistem iz dva razloga: • u jednom redu na ekranu se izdaju 32 znaka, i • ne postoje mala slova.

Izborom rešenja sa 32 znaku u redu postignuta je dobra čitljivost brojeva i teksta na ekranu bilo kojeg televizora. Ovo je onemogućilo ozbiljniji rad sa tekstom, pa nije imalo smisla uvoditi mala slova, jer bi to samo nepotrebno poskupilo sistem. Međutim, „galaksija“ ipak pruža izvesne mogućnosti koje su značajne u obuci programiranja, a odnose se na rad sa tekstom. Tako se može vršiti dopisivanje teksta (konkatenacija) i ispitivanje na jednakost. Neke dalje mogućnosti mogu biti ostvarene izgradnjom biblioteke potprograma. Tako, sledeći potprogram vrši poređenje dva ažbučna (znakovna) podataka koji su susedni elementi ažbučnog niza:

```
3000 FOR K=0 TO 15  
3010 A=BYTE (PTR X$(I)+K)  
3020 B=BYTE (PTR X$(I+1)+K)  
3030 IF A+B ELSE K=0:RETURN  
3040 IF A < B K=1:RETURN  
3050 IF A < B K=1:RETURN  
3060 NEXT K=0:RETURN
```

Rezultat poređenja se postavlja kao vrednost promenljive K na sledeći način: K dobija vrednost –1, 0 ili 1, prema tome da li je X\$(I) redom manje, jednak ili veće od X\$(I+1). Gornji potprogram se može uspešno koristiti za uredjenje ažbučnih nizova podataka u ažbučni redosled, što svakako treba da je prisutno u svakom kursu iz programiranja.

Grafika na računaru

Većina mikroračunarskih sistema ima određene grafičke mogućnosti, koje se saštovaju u crtanju na standardnom ekranu crno-belog ili kolor televizora. Crtanje se postiže tako što se izvestan broj tačaka na ekranu nalazi pod programskom kontrolom. Prema tome koliki je ovaj broj tačaka, grafika se deli na grubu i finu grafiku. U gruboj grafici, programski se kontroliše nekoliko hiljada tačaka, a u finoj nekoliko desetina hiljada tačaka, pa i više. „Galaksija“ raspolaže grubom grafikom sa $64 \times 48 = 3.072$ tačke na ekranu. Iako su ovo vrlo skromne grafičke mogućnosti, njihovo postojanje čini „galaksiju“ znatno interesantnijom i u obrazovanju iz sledećih razloga: • daje mogućnost upoznavanja učenika sa grafičkim mogućnostima računara, • znatno proširuje fond zadataka interesar- nih za programiranje, • daje mogućnost programiranju nekih video igara, što posebno doprinosi interesovanju mlađih za programiranje.

Da bismo ilustrovali kako i vrlo jednostavna rešenja mogu da proizvedu intere-

Mikroračunari nalaze sve veću primenu u raznim oblastima. Jedna od vrlo značajnih primena mikroračunara jeste i obrazovanje. Na svetskom tržištu postoji bogat izbor „obrazovnih“ paketa, ali je cena ovih uređaja veoma značajna, posebno u našoj sadašnjoj ekonomskoj situaciji. Može li, međutim, biti kakve koristi u školama i od mikroračunara minimalnih mogućnosti? Ovoj klasi pripada i mikroračunar „galaksija“. Iako skromnih hardverskih potencijala, „galaksija“ omogućuje rad sa brojevima, tekstom i grafikom. Koje su njene stvarne mogućnosti kao učila za programiranje, a koje kao sredstva za unapredjenje nastave?



sanine crteže na ekranu, posmatrajmo sledeći program:

```

100 IDEMONSTRACIJA GRAFIKE
110 HOME
120 X=INT(32*RND)
130 Y=INT(24*RND)
140 FOR I=X TO 63-X
150 FOR J=Y TO 47-Y
160 IF DOT I, J UNDOT I,J:ELSE DOT I,J
170 NEXT J:NEXT I
180 FOR I=1 TO 200
190 IF KEY(4) GOTO 220
200 NEXT I
210 GOTO 120
220 FOR X=0 TO 63
230 FOR Y=0 TO 47
240 IF DOT X, Y UNDOT X,Y:ELSE DOT X,Y
250 NEXT Y:NEXT X
260 GOTO 260

```

Program bira, slučajnim izborom, tačku u levoj gornjoj četvrtini ekranra, a zatim, koristišći ovu tačku kao jedno teme pravougaonika, crta centralni pravougaonik na ekranu. Pravougaonik ispunjava tako da „pali“ ugašene tačke, a „gas“ upaljene tačke. Posle iscrtanog svakog pravougaonika, program čeka 18 sekundi, a onda bira novu slučajnu tačku, itd. Ako korisnik zeli da zadrži sliku, u pauzi između crtanja pravougaonika treba da pritisne taster D, posle čega nastaje završna obrada slike i program se završava. Pri svakom izvršavanju programa dobija se drukčija slika na ekranu.

Mašinski jezik

Profesionalno obrazovanje u oblasti računarstva ne može se u potpunosti ostvariti bez pristupu mašinskog nivoa programiranja. Zapravo, tek upoznavanjem mašinskog jezika i arhitekture konkretnih računara na

Prva lasta: Sa računaram „galaksija“ počinje da se ostvaruje ideja o uvođenju računara i u naše škole

pravi način se sagledavaju i mogućnosti računara. U slučaju „Galaksije“ može se upoznati mašinski jezik mikroprocesora Z80A, koji je, inače, često korišćen osmo bitni mikroprocesor. Manji programi na mašinskom jeziku mogu biti pisani u heksadekadnom obliku. Unošenje ovakvih programi najbolje je organizovani u obliku programske datoteke. Na ovaj način će se lako vršiti ispravke u programu. Ilustrujemo ovu sledećim primerom:

```

10 FOR I=&2C3A TO &2C48
20 TAKE A
30 BYTE I,A
40 NEXT I
50 INPUT X$ 
60 BYTE &2C41, BYTE(PTR X$)
70 X=USR (&2C3A)
80 GOTO 80
90 &01, &00, &02, &21, &00, &28, &36,
&41
95 &23, &0B, &CB, &78, &28, &F8, &C9

```

Pre unošenja programa u memoriju treba primeniti komandu NEW 15 da bi se oslobođio prostor za smeštaj potprograma na mašinskom jeziku. Startovanjem programa vrši se prenošenje mašinskog potprograma iz programske datoteke u memorijski prostor od adrese &2C3A do &2C48, a zatim uneti znak sa tastature (naredba 50) postavi se u adresu &2C41 i prene na izvršavanje mašinskog potprograma koji upiše uneti znak tako da se izda u svim pozicijama na ekranu.

Pisanje ozbiljnijih programa na nivou mašinskog jezika zahteva simbolički jezik, a za ovo je neophodno raspolagati odgovarajućim asemblerom za „galaksiju“. Razvoj ovakvog asimblera nije veći programski problem, pa će, nesumnjivo, uskoro biti na raspolaganju i za „galaksiju“.

Proširenja „Galaksije“

Većina mikroračunarskih sistema je otvorena prema raznim vrstama softvera, skog i hardverskog proširenja pa i računar

„galaksija“. Osnovna varijanta „galaksije“ može se programskom nadgradnjom učiniti moćnijom za razne primene. Ovo se lako postiže izgradnjom biblioteka potprograma koje podržavaju neke specifičnosti pojedinih primena, kao što je bibliotske lementarnih matematičkih funkcija, rad sa tekstrom, graficki rad, kao i izrada najraznovrsnijih aplikacionih programa, posebno za potrebe obrazovanja.

Hardverska proširenja treba orijentisati, pre svega, na one delove koji baziraju na elektronskim komponentama, kao što je proširenje RAM-memorije, fina grafika, ton i sl. Veza računara sa spoljnim svetom, koja omogućuje primenu u manjim eksperimentalnim situacijama, bila bi takođe vrlo značajna u nekim obrazovnim sadržajima. Posebno bi, svakako, bilo korisno povezivanje električne pišaće mašine sa „galaksijom“, pre svega za potrebe štampanja programi.

Prvi korak

Osnovna varijanta „galaksije“ sa 4 KB ROM i 4 KB RAM-memorije predstavlja mikroračunar koji u obrazovanju obezbeđuje:

- zbiljavanje učenika sa računaram.
- obuku u programiranju na bežiku,
- upoznavanje grafičkih mogućnosti računara na nivou grube grafike,
- upoznavanje mašinskog nivoa programiranja,
- obuku u programskim tehnikama koje su od značaja za navedene nivoje programiranja.

Prema tome, svi kursevi informativnog karaktera iz oblasti računarstva, bez obzira na obrazovni nivo, mogu se zasnovati na „galaksiji“. Kada učenik savlada programiranje na „galaksiji“, može pisati programe za prenošenje stičenih znanja iz raznih predmeta na računar. Ako bi profesori i bili u stanju da postave ovakve programske zadatke učenicima i da nadgledaju njihov rad, to bi bio izvanredan način za kreativnu i kritičku primenu stičenih znanja.

Autor ovog teksta veruje da bi ovakva prima računara u školama doprinela motivisanosti učenika za sticanje novih znanja, a time i unapredjenju nastave iz opštih i stručno-obrazovnih predmeta. Pisanje programa je kreativan posao u kojem projektant programa unosi sva svoja znanja o određenoj temi i rezultate svog rada daje na korišćenje drugima. To je intelektualna aktivnost čija je obrazovna vrednost najveća za one koji pišu programe, a znatno manja, često i nikakva, za one koji samo koriste rezultate rada tudiših programera. „Galaksija“ pruža mogućnost da se prvi korak u primeni računara u nastavi učini, najzad, i u našim školama.

Prof. dr Nedeljko Parezanović

moj Programiranje za početnike

prvi program

Škola programiranja u 100 lekcija

Šta, najpre, znači „znati bežik“? Pod ovim terminom se podrazumeva poznавање određenog broja naredbi i njihovog dejstva, razumevanje strukture programa i potprograma i, uopšte, sposobnost da se dati algoritam pretvoriti u program za neki računar. No, programiranje je nauka koja se nikako ne može svesti na prevaranje gotovog algoritma u program — vеstina je napraviti algoritam ili, kada dalje napredujemo, pronaći problem koji može algoritamski da se реши, pa ga onda реши. Ova „Škola programiranja“ ne može, obzirom na prostor koji zauzima, da zameni čitave biblioteke knjiga posvećenim onima koji uče programiranje, profesionalno ili amaterski. Njen zadatak je, kroz sastavljanje jednog programa koji će prolaziti kroz nekoliko etapa, da vas podstakne na dalji samostalni rad i istraživanje. Ne treba, kao i obično, da očekujete da ćete program koji sastavljate moći koristiti da upotrebiti — koristan program može da bude toliko složen da nije pogodan za početnika. No, ukoliko budez pažljivo čitati sledeće redove, verovatno ćete se prisetiti nekog problema iz vašeg svakodnevnog života koji može da se реши primenom računara i skupiti hrabrosti da se posvetite njegovom rešavanju. Ukoliko se to zaista dogodi, ova „Škola“ je potpuno ispunila svoj cilj.

Programi koje budemo pisali biće nameđeni računaru „galaksiju“, pa će njeni vlasnici imati lepu privilegiju da ih ukucaju bez ikakvih izmena. No, „Škola“ je u suštini nezavisna od računara; pa ćete moći da je pratite i ako ste vlasnik Spectruma, ZX81, pa čak i (premda uz malo problema) nekog džepnog računara sa mogućnošću programiranja. Ukoliko imate problema sa razumevanjem neke naredbe koja vam izgleda kao „specijalitet računara ‘galakciju’“, uzmete upstušto za njegovu upotrebu (jedan od razloga što smo se opredelili za ovaj računar je u tome što svi posedujete njegovu dokumentaciju) i učiteš se time da se prevodenjem programa mnogo što-šta nauči.

Ukoliko smatrate da ste dobro upućeni u programiranje, možda ćete ipak pozeleti da pročitate ovu „Školu“. Algoritmi i tehnike koju smo koristili su za početnike dobar jer su jednostavni. U mnogim slučajevima postoje bolja rešenja — na vama je da ih pronađete.

Čas 001 — Algoritmi i programi

Pre nego što počnemo sa pisanjem prvog programa, moraćemo da pogledamo

kako sve to izgleda na daleko jednostavnijim primerima. Tako će, doduše, naš prvi program biti naš treći program, ali — nigrde ne žurimo.

Reč „algoritam“ potiče od imena uzbeskog (autor ovog teksta mora da prizna da mu reč „ubeskog“ ne znači baš mnogo) matematičara Al Horezmija koji je živeo u IX veku naše ere. Ovaj matematičar neobičnog imera je definisao pravila za izvršavanje četiri osnovne računske radnje (pravilo za deljenje je bilo znatno primitivnije od onoga koja danas poznaju učenici nižih razreda osnovne škole!) i tako postao začetnik jedne danas neobično popularne grane matematike koja se naziva „teorija algoritama“.

Verovatno je za većinu naših čitalaca potpuno besmislena egzaktna definicija algoritma koju matematičari danas usvajaju (samo potpunoši radi, najmanje „strasna“ definicija algoritma je „skup instrukcija koja može da izvrši Postova mašina“). Ipak, pojam algoritma je veoma lako shvatljiv: svakoga dana izvršavamo hiljadu algoritama a da toga uopšte nismo ni svesni. Jeste li, na primer, nekada čekali autobus 84? Ako jeste, evo algoritma koji ste tada primenjivali:

1. Dodi na stanicu.
2. Da li je na stanicu neki autobus? Ako jeste, predi na korak 3 a ako nije ponovi korak 2.
3. Pogledaj broj autobusa.
4. Ako je taj broj 84, predi na korak 5. U protivnog vrati se na korak 2.
5. Udi u autobus.
6. Kupi kartu.
7. Poništi kartu.
8. Da li se autobus kreće? Ukoliko se kreće, vrati se na korak 8 a ukoliko miruje — predi na korak 9.
9. Da li autobus stoji na stanicu? Ukoliko je odgovor da, predi na korak 10; u protivnom, vrati se na korak 8.
10. Da li je to stаница na kojoj treba da izadeš? Ako nije, vrati se na korak 8. a ako jeste — izadi iz autobusa.
11. Stigao si na cilj.

Kao što vidimo, algoritmi se formulišu „po tаčkama“ (koracima). Svakoj od njih je obavezno da broj na koji se drugi koraci, ako je to potrebno, pozivaju. U okviru svake tačke dato je neko precizno upstušto koje treba izvršiti. Za formulisavanje algoritma neobično je važno da poznajemo naredbe koje onaj kome je upstušto namenjeno (računar, robot, drug kome objašnjavamo gde stanjumu...) — uvek da izvrši. U gornjem primeru korak 7 glasi: poništi kartu. No, moguće je da onaj kome je algoritam namenjen, jednostavno, ne zna šta to znači. Njemu bi bilo nužno objasniti kako da pronađe automat da poništava kartu, kako da okrene kartu pre nego što će staviti u puste automata, koliko treba da čeka dok se automat ne oglasi zvonom, šta

da radi ako automat ne funkcioniše... Obzirom da nas najviše zanima pisanje programa za računare, algoritamski koraci će najčešće biti ekvivalentni naredbama nekog programskega jezika, u našem slučaju bežika.

Vreme je da napišemo i mali program. Obzirom da nas najviše zanima pisanje programa za računare, algoritamski koraci će najčešće biti ekvivalentni naredbama nekog programskega jezika, u našem slučaju bežika.

Vreme je da napišemo i mali program. Obzirom da nas najviše zanima pisanje programa za računare, algoritamski koraci će najčešće biti ekvivalentni naredbama nekog programskega jezika, u našem slučaju bežika.

Evov najpre algoritma koji je, za početak, napisan samo rečima. Uzalna veličina je bilo koji prirođen broj n.

1. Da li je broj na paran? Ako jeste, predi na korak 3, a ukoliko nije na korak 2.

2. Pomoži n s tri i dodaj jedan rezultat. Zbir nazovi n (time je stara vrednost n „zaboravljena“). Predi na korak 1.

3. Podeli n s dva i količnik ponovи nazovi n.

4. Da li je n jednak jedan? Ukoliko nije, predi na korak 1, a ukoliko jeste završi sa radom.

Algoritam je prilično jasan, pa reč „nagadanje“, izgleda, nije pogodna za njegovo ime. No, tako nije. Vidi se da se polazi od broja n i da se on neprekidno deli sa 2 ili množi sa tri, pa se rezultat dodaje jedan) sve dok se ne dobije broj 1. Pitanje je da li će, bez obzira na broj n od koga smo počeli, uvek na kraju da se dobije 1, ili postoji neki broj n na koji jedinica nikada neće biti dostignuta. Poljski profesor matematike Ulam je pretpostavio da se, za svako polazno n, na kraju dobija 1, ali mu nije poslož za rukom da to i dokaze što, bar za sada, nije uspešio u jednom drugom matematičaru (za egzaktno rešenje problema je i dalje raspisana privlačna novčana nagrada) pa je ostalo samo da se — nagada. Sastavimo, dakle, program koji transformise neko n u broj 1 i poškupajmo da pronađemo neko n na koje se to neće dogoditi!

Na početku rada treba da zatražimo od korisnika da unese neko n. Za to nam, jasno, služi naredba INPUT. Zatim treba da ispitamo da li je n deljivo sa dva (broj je paran ako je deljiv sa dva). To ćemo učiniti tako što ćemo u najpre podeliti sa dva i naći ceo deo rezultata koji ćemo nazvati m. Zatim ćemo ispitati da li je m jednak n/2. Ukoliko je odgovor na ovo pitanje potvrdan, n je paran broj (na primer: INT(4/2)=4/2-2), a ako je određen — n je neparan (npr. 2=INT(5/2)»5/2-2.5). Ostatak algoritma je prilično jasan — posudebito bežicu naredbe koje odgovaraju pojedinim algoritamskim koracima.

10 INPUT N

20 M=INT (N/2

30 IF M=N/2 GOTO 60

40 N=3*N+1

50 GOTO 20

„Osnovna škola bejzika u stripu“, koju smo objavili u našem prvom specijalnom izdanju o računarima, bila je namenjena onima koji u programiranju čine prve korake. Iako takvih sigurno ima dosta, još je više onih koji su upućeni u osnove bejzika ali ne znaju šta bi dalje radili sa njim. Ovo poglavlje je namenjeno njima.

$$\begin{array}{l} 12 \times 9 = 108 \\ 11 \times 9 = 99 \\ 10 \times 9 = 90 \\ 9 \times 9 = 81 \\ 8 \times 9 = 72 \\ 7 \times 9 = 63 \\ 6 \times 9 = 54 \\ 5 \times 9 = 45 \\ 4 \times 9 = 36 \\ 3 \times 9 = 27 \\ 2 \times 9 = 18 \\ 1 \times 9 = 9 \end{array}$$

60 N=N/2

70 IF N=1 PRINT „DOSTIGNUT BROJ 1“
ELSE GOTO 20

Čim startujemo program uobičajenim RUN, računar će ispisati znak pitanja tražeći od nas da unesemo neki broj. Učinimo to i pritisnimo ENTER (ili RET — zavisi od tastature koju imamo) i, nekoliko trenutaka posle toga, primetimo da je računar završio posao i na to nas upozorio odgovarajućim komentartom. Ovim je, ujedno, završen i čas 001. Na kraju svakoga časa u pravoj školi su dopuštena pitanja, Mi, jasno, ne možemo da predvidimo sva pitanja koja će čitačima ove „Škole“ „pasti na pamet“, ali možemo da odgovorimo na ona koja se nameću sama od sebe:

— Ja ne verujem računarima. Sve što ovaj program radi je da, posle dužeg ili kraćeg vremena, prikaže neki komentar na ekranu. Da li bi moguće modifikovati program tako da računar prikazuje brojeve u nizu, sve dok ne prikaže broj 1?

• Nema većih problema: dovoljno je dodati naredbu 25 PRINT N i računar će u svakom prolasku kroz ciklus prikazivati broj iz niza.

— Da probamo... Radi, ali brojevi prebrzo prolaze — i dalje ne mogu da proverim da li računar dobro radi.

* To može da se reši i to na dva načina: ukoliko, u toku izvršavanja programa pritisnemo i držimo DEL, sve će se „zamrznuti“. Ako ne želimo da se mučimo na ovaj način, dodaćemo naredbu 26 FOR I=0 TO 1000:NEXT I i računar će čekati posle 1000:NEXT vrativ svakog broja. Od interesa bi, međutim, bilo i prebrojati koliko je prolazaka kroz ciklus bilo potrebno da se dođe do kraja posla. Kako bi to moglo da se ostvari?

— Trebalo bi dodati jednu promenljivu koju bi se povećavala za jedan na svakom prolasku kroz ciklus. Neka to bude promenljiva P: dodajmo, dakle, naredbu 27 P=P+1 i na kraju programa 85 PRINT P. Neka, N bude 5... Dobili smo da je računar prošao kroz ciklus 4.5 puta! Da li je možda poslednji prolazak računat polovicno?

* Naravno da ne. Jednostavno, promenljiva P na početku rada nije imala vrednost 0 već 0.5 (kod drugih računara bi možda bila prijavljena greška „No such variable at line 27“). Grešku lako ispravljamo dodajući naredbu 5 P=0.

— Da li je neko već eksperimentisao sa većim brojevima? Da li se uvek dobija rezultat 1?

* Pomoću velikih kompjutera isprobani su brojevi sa deset, petnaest i, u poslednje vreme, dvadeset cifara. Pokazalo se da svi

oni, posle duže ili kraće konvergencije, dolaze do broja 1. To, naravno, ne znači da jednoga dana neće biti pronađen broj koji se ne transformiše u 1 na opisani način, ali daje realne osnove za pretpostavku da je nagađanje prof. Ulama tačno.

— Da li naš program radi za tako velike brojeve (do 1 E38)?

* Veliki broj je, možda, neparan. U tom slučaju računar ga možni sa 3 i rezultatu dodaje 1. Ukoliko je zbir opet neparan, operacija se ponavlja, što znači da je program već udesetstvušen. Može, dakle, tako da se desa da broj izade iz opsega u kome računar radi i da se pojavi greška pri zaokrugljivanju. Zato broj koji kucamo treba da bude bar 20 puta manji od maksimalnog celog broja koji „galaksija“ može da prihvati (vama ostavljamo da pronađete koji je to broj).

* Znači li to da računar „galaksija“ ne može da nam pomogne ako želimo da ispitujemo veće brojeve?

* Naravno da ne znači. Mogli bismo da smeštamo svaku cifru broja u element numeričke matrice A(I) i da sāmi sintetišemo računske operacije. No, to je problem kojim ćemo se zabaviti nekom drugom prilikom. Shvatili smo da su program i algoritam, pa smo spremni da se uhvatimo u koštač sa problemom koji ćemo rešavati u ovoj „Školi“.

Čas 010 — Tablica množenja

Prethodni problem je bio jednostavan zato što nam je algoritam bio unapred dati i treba ga samo pretvoriti u program. Došlo je vreme da se oprobamo i u sastavljanju algoritma: treba da napišemo program pomocu koga računar testira korisnikovo poznavanje tablice množenja. Ovaj program, naravno, nije namenjen nama koji ćemo ga pisati — znanje bežika podrazumeva i neka najosnovnija matematička znanja. No, za osnove iz naše okoline ovaj program može da bude zabavan i, što je još važnije, koristan.

Pre svega, treba jasno da definisemo problem i odlučimo šta računar treba da radi. Testiranje tablice množenja se obavlja tako što računar zadaje probleme (problemima su tipa 2*7=?), a zatim očekuje od korisnika da ih reši. Korisnik kuca odgovor, računar proverava njegovu ispravnost i izdaje odgovarajući izvestaj. Proveravanje rezultata ne bi trebalo da predstavlja poseban problem — svaki računar zna da množi, a naredba IF je kao stvorena za testiranje rezultata. Pravi problem je, dakle, postavljanje problema. Kako računar može da odluči koje će brojeve prikazati korisniku?

Prvo što ćemo pomislići je da se u memoriju računara stave svi mogući parovi brojeva. Obzirom da će naš program testi-

rati poznavanje tablice množenja 10x10, „pamćenje“ 100 parova brojeva bi odneло 800 bajta memorije, što i nije previše. U memoriju bismo upisali i redosled postavljanja problema: prvo, na primer, računar treba da prikaže osmi par brojeva, zatim pedeset šesti i tako dalje. To bi oduzeo još izvestan prostor, ali bi utrošak memorije i dalje bio prihvatljiv. No, računar bi zadavao probleme uvek istim redom, što znači da bi test, posle izveznog vremena, postao nepouzdan. U takvom trenutku bismo morali da menjamo čitav program ugradjući u njega novi redosled brojeva. To, po mogućству, treba da izbegavamo — jednom napisani program bi morao da bude u stanju da pouzdano radi bez ikakvih modifikacija.

Ideja koju smo imali, dakle, nije dobra. Kako da dođemo do dobre ideje? Najbolje je da razmislimo o načinu na koji bi čovek rešavao isti problem. Kako, na primer, učitelj testira poznavanje tablice množenja? Tako što kaže neka dva broja i od učenika zahteva da ih pomnože. Kako on dolazi do tva dva broja? U većini slučajeva kaže prve brojeve koji mu „padnu na pamet“ — ti brojevi su, dakle, slučajni.

Pojam „slučajan broj“ se vrlo teško može definisati. Što se nizova slučajnih brojeva tiče, njihova definicija je prilično jednostavna: to je grupa brojeva koji nisu povezani nikakvim međusobnim relacijama. Ukoliko ste proučili uputstvo za upotrebu vašeg računara i našu „Osnovnu školu bežika“, svakako ste premetili da je i računar „galaksija“ snabdevan naredbom RND koja služi za generisanje slučajnih brojeva. Kako je uopšte moguće da računar generiše slučajne brojeve kada su svih rezultati koje on daje stroge funkcija ulaznih veličina?

Slučajni brojevi koje računar generiše su, u stvari, pseudoslučajni. Među pseudoslučajnim brojevima postoje neke „tajne veze“, ali su one toliko složene da korisnik na osnovu prethodnih ne može nikako da pogodi dalje članove niza — za njega su, dakle, pseudoslučajni bojevi „dovoljno slučajni“. Formula koja povezuje članove niza pseudoslučajnih brojeva je najčešće tako konstruisana da se stalno dobijaju brojevi između 0 i 1 i to tako da je njihova srednja vrednost praktično 0,5 (što više brojeva generiše, to će njihova srednja vrednost biti bliža ovoj cifri). Za formiranje ovoga niza potreban nam je njegov prvi član da bi iz njega bio formiran drugi, iz njega treći i tako dalje. Taj prvi član niza se u stranoj literaturi sličkovito naziva „seed“ (seed=seme) i generiše tako što računar, u nekom trenutku, analizira sadržaj svog internog časovnika ili nekog drugog registra koji se često menja. Računar „galaksiju“ nije nikakav izuzetak — svaki put kada ga uključite, PRINT RND će dati drugi rezultat.

Slučajne brojeve u programu, dakle, generiševmo naredbom RND. Možemo da napišemo A=RND i promenljiva A će dobiti neku slučajnu vrednost između 0 i 1. Naša tablica množenja treba, međutim, da testira množenje celih jedinocifrenih brojeva. Zato ćemo dobiti slučajni broj pomoći sa devet, ueti ceo deo rezultata (tako dobijamo ceo slučajan broj između 0 i 8) i dodati mu 1 (broj je sada između 1 i 9), na isti način ćemo da generišemo i drugi slučajni broj, a zatim ćemo, pomoću naredbe

```

10 ! TABLICA MNÖZENJA
20 !
30 T=0
40 N=0
50 P=0
60 HOME
70 PRINT AT 256, "TACNI"; AT 265, "NETACNI"; AT 276, "BEZ ODGOVORA"
90 PRINT AT 320, "T"; AT 331, N; AT 345, P
100 PRINT AT 0, "KOLIKO JE"
110 A=INT(RND*99+1); B=INT(RND*99+1); C=A*B:PRINT
120 PRINT AT 9, A, " * "; B, " ? "
130 INPUT X#
140 IF VAL(X#)=C GOTO 300
150 N=N+1:PRINT AT 0, "NIJE TACNO!"
240 PRINT AT 32; A, " * "; B, " = "; C
250 FOR I=1 TO 500:NEXT I
260 GOTO 90
300 PRINT RT 0, "B R A V O ! "
310 T=T+1
320 GOTO 250

```

PRINT, ispisati problem na ekranu. Evo i programa:

```

10 A=INT(RND)*9+1
20 B=INT(RND)*9+1
30 PRINT "KOLIKO JE":A," * ";B," ? "
40 INPUT C
50 IF C=A*B GOTO 90
60 PRINT "ODGOVOR NIJE TAÇAN"
70 PRINT "POKUŠAJ PONOVO"
80 GOTO 30
90 PRINT "BRAVO! TAÇAN ODGOVOR"
100 GOTO 10

```

Ovaj program se neprekidno izvršava: računar zadaće probleme i daje informacije o tačnosti odgovora. Kako sve a naročito uvežbavanje tablice množenja mora na kraju da došadi, treba znati kako se prekida izvršavanje programa. Dovoljno je pritisniti taster BRK (BREAK) i računar će ispisati uobičajeno READY. Nešto radikalnije rešenje predstavlja pritisak tastera RESET (posledice su iste) ili isključivanje računara iz mreže za napajanje, čime će i program biti izgubljen. I nije neki program, reći ćete, ali ga ipak nemotje brisati — radičemo još na njemu. Pitanja?

— U programu se dva puta ponavlja naredba A (ili B)=INT (RND 9)+1. Da li može da se upotrebi potprogram?

— Ne može da se upotrebi potprogram koji bi racionalno trošio prostor. Neki računari omogućavaju definisanje funkcija, što znači da bi naredbe 10 i 20 mogle da se zamene sa

```

10 A=FNRNDN
20 B=FNRNDN; trebalo bi, jasno, dodati i 1000 DEFNRNDN=INT (RND 9)+1, ušteti u memoriju, dakle, ne bi bila značajna, a program ne bi bio ništa razumljiviji.

```

— Čini mi se da program, ovako kako je napisan, nije dovoljno animirajući za date i da ne daje realne rezultate koji bi govorili o njegovom znanju. Program, pre svega, postavlja više problema an ne prebrojava tačne i pogrešne odgovore. Osim toga, vreme za razmišljanje nije nikako ograničeno, pa onaj ko sedi pred računaram ne mora ni da odgovara na pitanja — ne postoji način da se ustanojni koliko je problema bilo postavljeno!

— Tačno. Međutim, ovaj program predstavlja samo početak. Pri pisanju nekog složenog programa možemo da se opredelim za jedan od dva pristupa: da pišemo onakav program kakav on na kraju i treba da bude ili da ga razvijamo „korak po korak“. Prvi pristup primenjuju iskusniji programeri koji cene svoje vreme. Drugi je pristupač korisnicima koji tek počinju:

razvijajući program, oni imaju priliku da rešavaju probleme na koje nailaze jedan po jedan kako nailaze na njih i da se, kada ne mogu dalje da napreduju, zaustave na nekoj verziji programa i proglaše je za konačnu. Radicemo, dakle, dalje na programu za testiranje poznavanja tablice množenja rešavajući probleme iz postavljenog pitanja obrnutim redom.

Čas 011 -- kako ispisati tabelu?

Poslednja primedba je bila da program ne formira tabelu tačnih i netačnih odgovora i rubriku koja govori o vremenu razmišljanja korisnika. Tu tabelu ćemo brzo napraviti, ali moramo da se odlučimo na koji deo ekranu da je smestimo i kako da je formiramo. Najjednostavnije je da se u prvom deo ekranu nalazi samo problem koji dete treba da rešava, u drugom odgovor koji dete kucu a, nešto ispod loga, tabela koja ilustruje njegovo znanje. Neka ona počinje od devetog reda na čiji početak, kao što vidimo sa mape ekrana (strana 11 uputstva za upotrebu „galaksije“), dolazi sa PRINT AT 256.

Na samom početku programa moramo da, kako bi to programeri rekli, inicijalizujemo brojače. To će jednostavno biti promenljive T, N i P. Promenljiva T će brojati tačne odgovore, promenljiva N tačne a promenljiva P problemu koje dete nije rešilo u predviđenom vremenu. U početku problema nije ni bilo, pa će sve tri promenljive dobiti vrednost nula za što su nam potrebne tri naredbe . . .

— Da li su te naredbe zaista neophodne i onima koji nemaju „galaksiju“? Zar promenljive T, N i P, obzirom da nisu korisčene, već nemaju vrednost nula?

— Kod „galaksije“ ne sma da se računa sa početnim vrednostima promenljivih (iskusniji korisnici će primetiti da sve promenljive na samom početku rada imaju vrednost 0,5 da se ona ne menjaju naredbom RUN). Neki računari (TRS 80, na primer) automatski dodeljuju vrednost nula svakoj promenljivoj kojoj nije dodeljena vrednost. Čak i ako posedujete računar kod promenljivoj kojoj nije dodeljena vrednost. Čak i ako posedujete računar kod koga dodeljivanje vrednosti promenljivima na početku rada nije neophodno, programerski „bon-ton“ zahteva da ga izvršite, jer će tako onaj ko analizira program na samom početku znati koliko promenljivih ima, koliko će one memorije zauzeti i, uopšte, što da

```

30 T=0
40 N=0
50 P=0
60 HOME
70 PRINT AT 256, "TACNI"; AT 265, "NETACNI"; AT 276, "BEZ ODGOVORA"
80 FOR L=1 TO 20
90 PRINT AT 320, T; AT 331, N; AT 345, P
100 PRINT AT 0, "KOLIKO JE"
110 A=INT(RND*99+1): B=INT(RND*99+1): C=A*B: PRINT
120 PRINT AT 9; A; " * "; B;
130 Y#=00000000: "DOT"
140 INPUT X#
200 S=VAL(PTR Y#+6)
210 IF S<25 GOTO 270
220 PRINT AT 0, "PREDUGO RAZMISLJAS!"
230 P=P+1
240 PRINT AT 32; A; " * "; B; "="; C
250 FOR I=1 TO 500: NEXT I: NEXT L
260 GOTO 260
270 IF VAL(PTR X$)=C GOTO 300
280 N=N+1: PRINT AT 0, "NIJE TACNO!"
290 GOTO 240
300 PRINT AT 0, "B R A V O !"
310 T=T+1
320 GOTO 250

```

radi ako njegov računar ima „strožiji bezik“.

Da nastavimo. Svaki put kada dete da tačan odgovor, računar treba da izvrši naredbu T=T+1 i tako poveća sadržaj promenljive T za jedan. Kada dete pogreši, računar će povećati sadržaj promenljive N, a kada prekorači predviđeno vreme — promenljive P. Posle toga računar će izvršiti naredbe kojima će stari sadržaj tablice biti zamenjen novim koji je aktuelan.

— Da li je potrebno da najpre izberimo stari sadržaj tablice?

— Nije. Koristimo naredbu PRINT AT koja štampa ono što nalazimo na proizvoljnoj poziciji ekranra, ne obraćajući pažnju na to što se ovim ispisivanjem menjaju sadržaji koji se na taj poziciji već nalazio. U ovom slučaju ne može da se javi problem koji bi u nekim sličnim programima bio nepriyatniji: obziru da se broj tačnih, pogrešnih i zakasnih odgovora nikada ne smanjuje, ne može da se desa da od prethodnog sadržaja preostane neko slovo koje bi, zajedno sa novoispisanim, promenilo smisao poruke (ako preko broja 10 štamparamo broj 9, dobitemo 90 a ne 9).

Program sa prve slike omogućava, osim svega što i prethodni, delimično ispisivanje tabele. Broj zakasnelih odgovora je, međutim, uvek nula, jer još nismo ni govorili o naredbama kojima bismo obezbedili da „galaksiju“ mjeri utrošeno vreme.

— To bar nije problem. Pa „galaksiju“ poseduje ugrađeni časovnik...

„Galaksija“ zaista poseduje veoma tačan časovnik koji čemo u ovu prilici svakako koristiti. Da se podsetimo: časovnik je usko povezan sa promenljivom Y\$ kojom, na početku, treba dodeliti vrednost „00.00:00“. Da aktiviramo časovnik, koristimo naredbu „DOT“, a da ga zastavimo „UNDOT“. Ove dve naredbe ne treba nikako mešati sa naredbama za čitanje po ekranu — zvezdica na kraju im je sasvim izmenila smisao. Sve, bar za sada, izgleda

vrlo jednostavno: na početku aktiviramo časovnik i dodelimo mu nullu vrednost, a zatim, posto dete da odgovor, proveravamo da li je prošlo više od, na primer, 25 sekundi. Problem, međutim, leži baš u ovom poređenju: kako da uporadimo broj proteklih sekundi sa 25?

— Ništa lakše — upotrebimo naredbu IF Y\$25 P=P+1...

■ Taman posao! Pre svega, Y\$ je alfanumerički, a 25 je broj, pa nema nikakvog smisla poređiti ih. Mogli bismo, naravno, da napišemo i IF Y\$>„25“ ali to ne bi bilo nista bolje. Pre svega, u Y\$ nije smешten samo broj sekundi već i broj časova i minuta u formatu koji smo maločas opisali. Čak ni IF Y\$>„00:00:25“ ne bi mnogo pogodio, jer „galaksija“ ne omogućava poređenje alfanumeričkih sa pomoću simbola > — moguće je samo proveravati da li su sadržaji dve alfanumeričke promenljive jednak i to pomoću funkcije EQ.

— Problem, dakle, ne može da se реши...

■ Naravno da može. „Galaksija“, rečeno, ne može da poređi stringove, ali sasvim lepo poredi brojeve. Potrebno je, dakle, da pretvorimo sadržaj Y\$ u broj sekundi...

— Pa da! Za to služi naredba TIME opisana u poglavljaju koje je namenjeno iskusnijim vlasnicima „galaksije“. Samo...

— Samo što se radi o programu na mašinskom jeziku koji čak ni onima koji čitaju to poglavje nije baš sasvim jasan! Ne, mi ćemo problem rešiti koristeći „galaksiju“ bezik. On poseduje naredbu VAL koja pretvara alfanumerik u konstantu. Ukoliko, kako piše u uputstvu za upotrebu, upotrebimo naredbu A=VAL(PTR Y\$) nećemo uraditi ništa naročito. Promenljiva A će zaista dobiti vrednost konvertovanog stringa u Y\$, ali samo do simbola : koji, za „galaksiju“, služi kao oznaka kraja izraza. Dobjemo, dakle, broj sati koji će uglasnom uvek biti nula. Nas interesuju sekundi koji su sedmi i osmi simbol promenljive Y\$. Zato ćemo upotrebiti VAL(PTR Y\$+7). . .

— Zar smemo da sabiramo Y\$ (alfanumerik) sa 7 (broj)?

■ Naravno da ne smemo. Međutim, ispred Y\$ se nalazi naredba PTR koja, kao

što znamo, daje adresu promenljive u memoriji (u našem slučaju, to je 10880, što bismo videli iz mape sistemskih promenljivih da znamo da pretvorimo broj 2A80 u dekadni zapis) tj. broj. Tom broju dodajemo sedam, što je sasvim regularno, a onda izračunavamo vrednost izraza koji je na tu adresu smesten.

— Mogli smo, dakle, da upotrebimo i naredbu A=VAL (10887)!

■ Mogli smo, ali to nećemo da učinimo. Ovakvom izmenom bismo, doduše, uštedeli dva bajta, ali tada niko ne bi znao šta takva naredba radi a da ne računamo što bi, čak i opremijen mapom sistemskih promenljivih, pomicišo da se nalazi na pogrešnom putu, jer ni jedna sistemска promenljiva na počinje od 10887. Ovakvo je svakome ko se upoznao sa „malim tajnima“ naredbe PTR sve sasvim jasno.

Program sa slike 2 ilustruje ono što smo do sada uradili. Treba obratiti pažnju da se program ovoga puta ne izvršava neprekidno, već se, zahvaljujući naredbi broj 80, postavlja samo 20 problema. Posle prikazivanja poslednjeg „galaksija“ nailazi na naredbu 260 koju neprekidno izvršava ne kvarči tako izgled tabele. Rezultate koje je postiglo dete, jasno, ne može da „uništi“, bar ne po pravilima „fer-pleja“.

— Ništa lakše — upotrebimo naredbu IF Y\$25 P=P+1...

■ Pod fer-plejom ne podrazumevamo način koga bi se setilo čak i dete koje je do 20 pogrešnih odgovora — isključivanje računara iz napajanja. Osim toga, pritišak na BRK prekida izvršavanje programa. Čak i ako prepostavimo da ovaj taster nije naročito upadljiv, taster sa RESET na zadnjoj strani prosto mami da bude pritisnut. . .

■ Testirajmo program.

— Pokušajmo, najpre, da damo tačan odgovor. — Dobro je, kaže BRAVO. A poštašen? i to je u redu. Da sačekamo više od 25 sekundi. . . Nikada 25 sekundi nije ovoljko trajalo... — Prošao je i minut i po i još ništa. Izgleda da program ne radi!

■ Pokušajmo sada da damo tačan odgovor. Evo, kaže da smo predugo razmišljali!

— U pravo vreme. Zašto to nije ispisao ranije?

■ Zato što mu nismo rekli da to uradi. Koristili smo naredbu INPUT koja nalaže računaru da čeka sve dok korisnik ne otkuca traženi podatak i da onda nastavi sa radom. Vreme, u međuvremenu, teče, što interni časovnik lepo registruje, ali to ni na koji način ne možemo da registrujemo dok računari ne nastavi sa radom. Pre nego što počnemo da radimo na ovom problemu, treba da primite još jedan: pokušajmo da sačekamo minut i deset sekundi.

— Računar kaže da je odgovor dobar, iako je vreme prekoračeno više nego dvostruko!!!

■ Problem je u tome što smo, ispitujući vreme, testirali samo sekunde. Kada minut prode, sekunde ponovo počinju od nule. Mogli bismo, jasno, da proveravamo i minute, ali i časove, ako hoćemo da budemo sasvim dosledni, ali se ovim nećemo baviti. Ovaj problem će se rešiti sam od sebe kada rešimo onaj mnogo veći — kako da registrujemo sam trenutak kada vreme istekne. O tome, naravno, sledećeg časa.

Čas 100 — Naredba KEY

Od naredbe INPUT, na žalost, moramo da odustanemo, jer ne možemo da je promenimo tako da vraća kontrolu našem programu da bismo kontrolisali vreme.

— Naredba INPUT je previše zgodna! Zar problem kontrole vremena ne može baš nikako da se reši?

* Može, ali ne na nivou kojim se bavimo (iskusnijim korisnicima koji čitaju ovu školu a žele da provere svoja znanja o interplu predlažemo da pokušaju da savstave odgovarajući mašinski program za modifikovano INPUT). Moraćemo, kao što rekosmo, da se odrekнемo INPUT-a i proučimo jedini preostalu naredbu za testiranje tastature — naredbu KEY. Ona ima veliku prednost nad naredbom INPUT utoliko što se izvršava samo jedan trenutak — kada naide na naredbu IF KEY (1) GOTO 1000 računar proverava da li je taster A pritisnut iako jeste ide na liniju 1000, a ako nije na sledeću: čekanja, ni u kom slučaju, nema.

Moramo, pre svega, da razmotrimo koju grupu tastera treba testirati. Dete, po prirodi stvari, treba da otkuca neki broj, što znači da tastere sa slovima i specijalnim znacima možemo sasvim lepo da ignorisemo. Ostaje da treba testirati samo tastere 0-9, odnosno, prema tabeli sa strane 21 upisatva. KEY (32) — KEY (41). Njih čemo najbolje da testiramo u petlji, i to jedan po jedan, na primer ovako:

```
10 X$=" "
20 FOR I=32 TO 41
30 IF KEY (I) ELSE NEXT I : GOTO 20
40 X$=X$+CHR$ (I+16) : GOTO 20
```

Najteže je shvatiti liniju 40. Na početku smo „promenljivoj X\$ dodelili vrednost „praznog stringa”, tj. alfanumerika koji ne sadrži nijedan simbol. Čim otkrijemo da je dete pritisnuo neko slovo, treba da ga „prilepimo” u taj string koji će tako, iz trenutka u trenutak, postajati sve duži. Ovo dodavanje slova je, zapravo, sabiranje alfanumerika. Ako, na primer, X\$ ima vrednost „GALAKSI”, Y\$ vrednost „JA” a mi izvršimo X\$=X\$+Y\$, PRINTX\$ će dati reč GALAKSIJA koja je dobijena nedovezivanjem JA na GALAKSI.

Nedovezivanjem alfanumerika „0” na X\$ bi moglo da se realizuje sa X\$=X\$+„0” ili, prema tabeli sa strane 18, X\$=X\$+CHR\$ (48). Primetimo da, za testiranje tastera na kome je nula, koristimo IF KEY (32) i da je 48=32+16. U ciklusu koji smo maločas dali, promenljiva i ima ulogu brojača koji ispituje tastere koji su pritisnuti. Čim naide na neki od njih, „galaksija” na promenljivoj X\$ nadovezuje CHR\$ (16) što predstavlja upravo ASCII kod pritisnutog tastera. Moguće da neće biti loše da ovo sami isprobate na nekoliko primera i uverite se da program korektno radi. Da biste, međutim, ovo uopšte mogli da uradite, treba da otkrijete trenutak u kome je korisnik zavrsio sa kucanjem broja. Signal za to je, naravno, pritisak na LENTER, pa čemo program do puniti naredbama:

```
30 IF KEY (I) ELSE NEXT I : IF KEY (48)
GOTO 100 ELSE GOTO 20
100 PRINT X$ : GOTO 10
```

Startovaćemo program u kucatim prizvoljne brojeve. Po pritisku na ENTER, računar će prikazati broj koji smo otkucali i početi

— Zašto ovaj program treba da bude ovolikog „temeljanja”? Da bi neka cista bila registrovana, treba pritisnuti neki taster i poduze da držati pritisnutim. Sa druge strane, ovu dugu pritiskanje može da ima i nepriyatnu posledicu da računar neki taster

10 !	TABLICA MNOŽENJA	
20 !		
30 T=0		
40 N=0		
50 P=0		
60 HOME		
70 PRINT AT 256, "TRCNI"; AT 265, "NETACNI"; AT 276, "BEZ ODGOVORA"		
80 FOR L=1 TO 20		
90 PRINT AT 320, T; AT 331, N; AT 345, P		
100 PRINT AT 0, "KOLIKO JE"		
110 R=INT(RND*99+1); B=INT(RND*99+1); C=R*B; PRINT		
120 PRINT AT 9, R; " * "; B; " =? "		
130 M=0; X\$=""; Y\$="00 00 00": DOT\$		
140 IF KEY(48)<>M>0) GOTO 270		
150 IF KEY(29)<>M>0) GOTO 330		
160 FOR I=32 TO 41		
170 IF KEY(I) ELSE NEXT I: GOTO 200		
180 X\$=X\$+CHR\$(I+16): M=M+1: IF M>10 GOTO 220		
190 PRINT AT 19, X\$		
195 NEXT I		
200 S=VAL(PTR Y\$+6)		
210 IF S<25 GOTO 140		
220 PRINT AT 0, "PREDUGO RAZMISLJARS"		
230 P=P+1		
240 PRINT AT 32; R; " * "; B; " ="; C		
250 FOR I=1 TO 500: NEXT I: NEXT L		
260 GOTO 260		
270 IF VAL(PTR X\$)=C GOTO 300		
280 N=N+1: PRINT AT 0, "NIJE TACNO!"		
290 GOTO 240		
300 PRINT AT 0, "B R A V O ! "		
310 T=T+1		
320 GOTO 250		
330 M=M-1		
340 BYTE PTR X\$+M, 0		
350 PRINT AT 19, X\$		
360 GOTO 200		

registruje dva puta, a mogućnost da stvara korigujemo pritiskom na — izgleda, ne postoji.

■ Problem je u tome što „galaksija” nije mnogo brz računar, a bežik nije mnogo brz jezik. Dok „Galaksija” ne prode kroz čitav ciklus i ne naide na pritisnut taster, može da prode doista vremena. Rešenje ovoga problema ne postoji — mogao bi jedino da se napiše poseban program na „mašincu” koji, kao što rekosmo, izlazi iz okvira ove škole. Što se tiče korigovanja — to pomoći ima. Treba, najpre, da testiramo taster — (KEY (29)). Ako je on pritisnut, treba da promenljivoj X\$ „oduzmemos” jedno slovo, što i nije tako jednostavno kako na prvi pogled izgleda. Treba, pre svega, da naučimo da „galaksija” smetiš alfanumerike u standardnim ASCII, zapisu na čemu se, izas poslednjeg slova, nalazi 00 koja označava njegov kraj. Cilj je da tu nulu pomerimo za jedno mesto ulevu.

Da pristupimo memoriji treba nam, jasno, naredba BYTE i promenljivi koja će nam govoriti koliko smo slova do tog momenta otkucali. To će biti promenljiva M koja će na početku dobiti vrednost 0 i koja će, zatim, biti povećavana kad god na X\$ nadovežemo neko slovo. Kada pozelimo da „odešemos” poslednje slovo, izvršćemo M=M-1: BYTE PTR X\$+M, 0 i tako skratiti X\$.

— Ali ovom naredbom mi nismo „po merili” nulu za jedno mesto ulevu! Jedno stavno smo na to mesto upisali nulu, ali se ona i dalje nalazi i na sledećem mestu — kraj stringa, tako, označavaju dva nula-bjata!

■ Tačno, ali to ovde nije mnogo bitno. Prvi nula bajt je za „galaksiju” oznaka kraja stringa, dok drugi ni u kojoj prilici ne

biva tretiran, jer, jednostavno, nije deo korisnog sadržaja promenljive X\$.

Na trećoj slici je prikazan program koji ćemo nazvati konačnim. U njemu ima još nekoliko mesta koja vrede prokomentarisati. Promenljivu M smo, osim za redakture, upotrebili za neke druge sitinice: „galaksija” najpre, proverava da li je otkucani string duži od deset slova. Ukoliko bi se dete igralo kucajući broj duži od 16 slova (po prirodi stvari, besmislen kao rezultat množenja dva dvočifrenra broja) kolika je maksimalna dužina „galaksijinoj” alfanumerici, mogli bi da nastupe razni problemi. Uz ovaj test računara će svaku ovakvu igru priglasiti sa „predugo razmišljanje” i preći na sledeći problem.

Testirao je, osim toga, da li je taster ENTER ili — pritisnut pre nego što je otkucan neki broj. Ukoliko jeste, pritisak ovog tastera biva ignorisan, jer bi mogao da izazove nove probleme. To smo učinili logičkim množenjem iskaza koje je opisan u 22 poglaviji Uputstva za upotrebu „galaksije” (strane 22-23).

I poslednje pitanje:

— Ovih sto lekcija je nekako mnogo brzo prošlo. Čini mi se da je njihovo numerisanje bilo u najmanju ruku čudno...

■ Lekcija nije bilo sto, nego 100. A binarni sistem programeri često koriste...

BIBLIOTEKA PROGRAMA

Spectrum

Kako postići besmr- tnost

Sve video igre, na ovaj ili onaj način, ograničavaju broj „života“ glavnog junaka. Često je to i istovremeno ograničavanje raspoložive količine energije, kiseonika, hrane i vremena. Posledicu znamo: igrač veliku većinu vremena provodi u prvih nekoliko nivoa (koji mu ubrzaju postaju nezanimljivi). Tek što uspe da pređe u sledeću (najčešće nepoznatu i težu) fazu igre, igrač izgubi nekoliko života, ne stigavaš da uoči sve prepreke i razradi novu takтику.

(Istini za volju, neke video igre, kao što je „Penetrator“, pored uobičajenog moda za igru, pružaju i mogućnost uvezivanja (training mode) igrača — igrač uvezava pojedine faze igre, ili, jednostavno, bira samo ono što voli.)

Naravno, bez ovih otežavajućih okolnosti većina igara bi izgubili svaku draž. Ipak, da bi ispravili ovu „nepravdu“, vredljubitelji igara su upornim radom otkrili ključne lokacije u programu koje je potrebno izmeniti da bi igrač postao „besmrtni“. Iako oduzimaju mnogo vremena, ovakvi poduhvati mogućuju sticanje novih značaja iz oblasti programiranja.

Za ovu priliku smo odabrali nekoliko popularnih profesionalnih igara, o čijim višim nivoima mnogi iscrpili spekturu, mašti još uvek maštaju. Navešćemo lokacije koje je potrebno izmeniti naredbom POKE neposredno pre startovanja programa. Većinu ovih lokacija je otkrio Elisa Kabilio iz Beograda. Za program Pyramid zaslužan

Program	Adresa	Sadržaj	Posledica
COOKIE	28698	0	bezbroj života
PSSST	24984	0	bezbroj života
Manic Miner	35136	0	bezbroj života
	34269	n	n = broj života, n < 32
Pyramid	44685	0	neograničena energija
Arcadia	25776	0	bezbroj života
Hunchback	26888	0	bezbroj života
Transversion	26020	0	bezbroj života
Jet Set Willy	34483	195	eliminisanje zaštite
	35899	0	bezbroj života
ZIP-ZAP	53750	0	bezbroj života
	53751	0	bezbroj života
	53752	0	bezbroj života
	53753	0	bezbroj života
	54141	0	bezbroj života
	54142	0	bezbroj života
	54143	0	bezbroj života
	54144	0	bezbroj života
	54038	0	bezbroj života

je Zoran Štramberger, a za program Arcadia Slobodan Vujnović, oboljica iz Zemuna.

Iz tabele se vidi da Manic miner može da se prekroji i tako da igrač ima fiksani broj života, ali ne veći od 32, jer se u tom slučaju događaju čudne stvari na ekranu. Na primer, ako želite da igrate sa deset života, kućajte POKE 34269,10.

Unošenje ovih izmena (u žargonu: patch, zakrpa) vrši se na sledeći način:

- 1 Učita se bežijk program koji učitava ekran u mašinski deo, ili delove programa. To se postiže komandom „MERGE“.
- 2 Pronalazi se linija u kojoj se izvršava naredba RANDOMIZE USR xxxx ili PRINT USR xxxx, ili nešto slično. Ovom nardbom bežijk nepovratno skače u mašinski program na adresu xxxx, pa se izmene moraju uneti pr njenog izvršenja.

- 3 Liniju ćemo izmeniti tako što ćemo neposredno pre ove naredbe ukucati POKE adresu, sadržaj iz tabele.
- 4 Sada možemo da kažemo RUN ili GOTO ... da bi se nastavilo normalno učitavanje ostalih delova programa. Još je praktičnije sačuvati ovaj izmenjeni program sa, na primer, SAVE „NOVI“ LINE - - i pomoći njega kasnije vršiti učitavanje programa.

Za izmenu programa Jumping Jack potrebno je ubaciti tri nove linije i program sačuvati sa SAVE „JJ-EI“ LINE 10:

```
10 POKE 26034.0: POKE 26035.91: RESTORE 30: FOR a=23296 TO 23309
20 READ d: POKE a,d: NEXT a: GO TO 2
30 DATA 33.0,0.34,141.38,24,34,143,117,195,66,111
```

Nedavno su u literaturi pojavilo još nekoliko zahvata koji omogućavaju ulazak u bilo koji nivo Manic Miner. Postupak je prilično čudan: učitati normal-

no program, zatim pritisnuti ENTER i sledeći niz brojeva: 6031769. Na dnu ekana bi trebalo da se pojavi cipela. Pritisakujuci ISTOVREMENO brojeve iz tabele koje slijedi, ulazimo u odgovarajući nivo igre.

nivo	Kombinacija
1	6
2	61
3	62
4	621
5	63
6	631
7	632
8	6321
9	64
10	641
11	642
12	6421
13	643
14	6431
15	6432
16	64321
17	65
18	651
19	652
20	6521

S. Vujnović

Spectrum

64 znaka u redu

Ovaj program vam omogućava da na računaru ZX Spectrum, oštvrte ispis teksta na ekranu u širini od 64 slova u jednom redu. Program je u potpunosti urađen u bežiku. Zbog ograničenja koja nameće rezolucija na ekranu, slova su nešto manje čitljiva nego originalna, ali uz malo podešavanja osvetljivanja i kontrasta na vašem TV aparatu, to ne bi trebalo da predstavlja problem.

Da bi se razumelo kako program radi, potrebno je upoznati se sa značenjem sistema varijabli CHARS. Ta varijabla stoji na adresi 23606 i 23607 i

pokazuje Spectrumovom operativnom sistemu gde treba da traži definiciju oblika slova koje ispisuje. Pribitna vrednost za CHARS je 23606 = 0 i 23607 = 60 i ona pokazuje na ROM (fiksnu memoriju), pa stoga i dobijamo slova odmah po uključenju računara. Ta sistemska varijabla se može postaviti na bilo koju drugu vrednost u okviru RAM-a. Ako se na novoj adresi nalazi neka druga, nova definicija slova (na primer, cirilica). Spectrum će je prihvati kao svoju sopstvenu.

Da bi ta osobina mogla da se iskoristi za dobitanje 64 slova u redu na ekranu, potrebno je definisati dva nova seta znakova. Oba ova seta će sadržavati ista slova, samo što će se u jednom setu slovo nalaziti u levoj polovini Spectrumovog standardnog znaka, a u drugom setu u desnoj polovini. Sada je još potrebno da navedemo računar da ta dva znaka odstampe na jednoj standardnoj poziciji na ekranu, što se postiže na redbom OVER 1, koja, inače, ima zadatak da preštampa dva slova jedno preko drugog.

Niz znakova može imati do 64 slova i smešta se u niz BS, koji može biti dimenzioniran na novu širinu ekranu. Ta dimenzija je može biti veća od 64 i ne manja od 1. Program će prihvati niz slova bilo koje dužine (do 64).

Potrebno je takođe, objasniti funkciju promenjive F. Ona služi programu da za svako parno slovo zadrži poziciju za pisanje na ekranu i tako omogući da, preko OVER 1, budu napisana dva slova umesto jednog u jednoj standardnoj slovnoj poziciji.

Ako vam u pojedinim slučajem program krahira sa novim setom znakova, povratite na standardni set se vrši sa: GO TO 9999

Program omogućava štampanje na bilo kom mestu na ekranu, i to na način koji je sličan Spectrumovoj PRINT AT instrukciji. To znači da pre poziva podprograma treba postaviti dve promenjive: „i“ za red „p“ za kolonu u kojoj želite da stampate. Takođe treba biti obaviz da dužina teksta koji stampate ne pređe krajinu desne kolonu jer će se tada javiti greška.

Sam program je veoma kratak i obuhvata samo linije 9000 do 9999. Ceo polprogram se nalazi u liniji 9000. Linija 9998 vrši automatsko snimanje potprograma i novog seta znakova. Linija 9990 je za automatsko učitavanje seta znakova. Sama, definicija dva nova seta znakova je prilično obima i zahteva 1540 bajtova na kraju memorije (iznad RAMTOP-a).

BIBLIOTEKA PROGRAMA

asetu. Od sada pa na daleko možete ga učitavati sa LOAD 454 slova/1" ili samo LOAD".

Osnovna vrednost ovog programa je što umesto 704 slovna (koja isključimo donje u sistemске linije), omogućava korišćenje 1408 slovnih mesta na ekranu, što znači i dva puta više informacija u isto
vreme.

Jedan predlog bi bio da se tako formirani ekran smesti kao CREENS datoteka na traku, a

Da bi se olakšalo ukucavanje, broj bajtova je sažet na 1/4 prvobitnog, tako da treba ukucati samo 384 broja u DATA linijama. To sažimanje je zahtevalo jednu malu rutinu koja će izvršiti potrebne transformacije i premetstiti karakter seta na njegovo pravo mesto. Rutina za premetstavljanje seta znakova je u linijama 60 do 110, a definicija znakova je u DATA linijama 130 do 360. Linija 380 služi za demonstraciju rada programa.

Posle ukucavanja programa i njegovog startovanja, mogu se izbrisati sve DATA linije, kao i rutina za premeštanje seta znakova.

Pošto ukucate ceo program, otukajte RUN da izvršite definisanje novog seta znakova. Program će tada izvršiti malu demonstraciju rada. Zatim otukajte GO TO 9999 i izvršite linije 10 do 390 jer vam više nisu potrebne. Sa GO TO 9988 izvršite snimanje podprograma i definisano seta znakova na

re nekog vašeg programa, tako da u toku učitavanja programa na ekranu će imate sve informacije i uputstva za taj program. Ista tehnika se može koristiti da smestite sadržaj uake na njen početak. Učitavanje datoteke SCREEN\$ traje oko 40 sekundi. Dovoljno je, da, toksto ostavite na početku svake trake, pa da kasnije snimite njen sadržaj koji je napisali uz pomoć ovog programa. Datoteka tipa SCREEN\$ se učita sa LOAD "SCREEN\$".

korivoj Perzić, dipl. ing.

„galaksija“
**master-
majnd**

Masternajnd je poznata ločka igra na specijalnoj tabli a raznobojnim pribadacama. edan od igrača sa zadnje strane tabele postavlja četiri proizvoljne pribadace u proizvoljnu poziciju, postoji šest boja

```

100 PRINT "KOMPUTERSKA IGRA: POKLONI SMIJU"
200 GOTO 1028
2000 PRINT AT 362,"V A R A S !"
818 WK=1
828 S.1829
828 HOME
818 PRINT " * M A S T E R M A J N
*:1
828 PRINT " P R A V I L A I G R E S U ,
ADAMO SE."
828 PRINT " O P S T E P O Z N A T A , O V A J
D R O G A M "
828 PRINT " O M O G U C A V A P A R A L E N U
I G R E "
828 PRINT " T I V R A C U N A R A , M O Z E
I L I D A P O D A "
828 PRINT " B E D I S T " ; P
828 PRINT " I G R A J U C I F R E 0-6
A C E T I R "
828 PRINT " M E S T A , S V I P O D A C I S
K U C A J U "
828 PRINT " N O R M A L N O K O M A D A I
K U C A J U "
818 PRINT AT 499,"P R I T I S N I R E T
128 IF KEY(48) ELSE G.4128

```

ibadača, a neki od otvora mogu da ostanu i prazni (sed-
i)ci. Pošto "galaksija" nema kolor grafiku, boje su zamjenjene brojevima 0–6. Da bi igra bila zanimljiva, igrač i računar paralelno pokušavaju da pogode kombinaciju koju je protivnik zamislio — onaj kome to pre uspe dobija pobjednički poen.

Najpre treba, sa NEW 512, rezervisati prostor za mačijski

Zadatka drugog igrača je da godi zadatu kombinaciju. U tom cilju, on u red tab stavlja. Antici, pribaveš da je boja).

Zadatka drugog igrača je da godi zadatu kombinaciju. U tom cilju, on u red tab stavlja. Antici, pribaveš da je boja).

Zadatka drugog igrača je da godi zadatu kombinaciju. U tom cilju, on u red tab stavlja. Antici, pribaveš da je boja).

na kasetu sa SAVE, a zatim startovati program sa RUN. Potom zaglavljati, računari će zatražiti da se predstavljaju, a zatim će početi igra. U toku predstavljanja treba da zamislimo broj koji će računari pogadati; to je, kako rečemo, četvorocifreni broj, u kome nijeda cifra ne sma da bude veća od 6, dok se cifre, naravno, smiju ponavljati. Kada je na nasi red, treba da otkucamo pretpostavku (npr. 1234) i pritisnemo RETURN (ENTER) i računari će prikazati koliko bismo crnii, odnosno belli pribadača dobili (21 oznjava dve crne i jednu belu), a zatim će on poškodi da pogodi našu kombinaciju. Odgovaramo mu na isti način – kucajući 21, 03 ili neki sličan broj i pritisnući ENTER. Igra se nastavlja sve dok neko ne pogodi protivničku kombinaciju (skor 40); računari tada ispisuju trenutni skor i igra se nastavlja.

Dejan Ristanović

„galaksija“ **Evolucija**

„Evolucija“ („Life“) je pozata simulacija života. Na nekom stanju korisnik zadaje početni organizam koji se sastoji od određenog broja živih ćelija, dok su preostala polja „mrta“. Taj organizam se razvija u strogim evolucionim uslovima – usamljene ćelije i ćelije koje žive u prenajeljenim područjima brzo umiru, dok se u umereno naseljenijem predelu ma iz casa u cas radaju nove ćelije. Neki početni organizmi, posle određenog broja generacija umiru dok drugi prelaze u neke stabilne forme. Za mnoge početne organizme se, verovatno ili ne, posle određenog broja generacija javlja struktura koja, čak i bez mnogo mašteta, podseća na čoveka.

Da biste otkucali ovaj program, treba najpre da rezervirate prostor za njegov mašinski deo — to činite tako što kucate NEW 454. Posle toga treba da unesete mašinski deo pomoću UTM-a ili nekog drugog programa. Ponovo otkucajte NEW 4544 da obrišete UTM, pa unesite bezijk koji, jednostavno, štampa uputstva. Sve to snimite na kasetu sa SAVE. Posle toga ćete program uvek startovati jednostavno RUN i slediti uputstva koja računar ispisuje.

Dejan Ristanović

BBC Electron
***Kako
presnimiti
program?***

četiri umesto dva bajta, što ne bi trebalo posebno da nas zabrine — oni koji kupe dodatni procesor sa 64 K (ili pola megabajta) RAM-a shvatice da program može da se učitava i u ovaj novi memorijski prostor, koji na neki način treba adresirati. Drugo započanje će biti još jednostavnije i korisnije: posto znamo da program počinje, koliko bajtova ima i koja mu je izvršna adresa, moći ćemo da ga snimimo na kasetu bez mnogo problema — dovoljno je da otkucamo "SAVE „ime“ FFFFFFFF + nnnn FFFFeeee, pri čemu nnnn označava dužinu programa u bajtovima, xxxx adresu od koje se on učitava a xxxx izvršnu adresu.

Na žalost, ne mogu svi programi da se prenesme na ovaj način — u cilju njihove zaštite adresa od koje se ovi programi učitavaju je odabranu tako da, po učitavanju, kontrola ne može da se vrati bežijk interpretatoru. Te programe je neophodno učitati na neko drugo mesto, a zatim snimiti tako da kopija bude identična originalu, što možemo samo koristeći sistemski potprogram OSFILE.

Posle čitanja programa sa trake računar će u svoj radni prostor smestiti parametre programa. Adresa od koje se program učitavaće će biti smještena od &3B1 do &3C1, izvršna adresa od &3C2 do &3C5, dužina u &3C6 (MSB) i &3C8 (LSB), a ime programa počevši od &3D2 pa do prvog nula bajta koji predstavlja terminator. Treba primjetiti da dužina nije uvek tačna — ako je njen bajt manje težine (LSB) nula, bajt veće težine (MSB) treba veštacki povećati za jedan da bi se dobio tačan podatak o dužini. Uz pomoć ovih podataka, knjil se nude ne-

Najjednostavniji način da prenimo program je da, npr., otkucamo „OPT 1.2, a zatim „LOAD““. Učitavanje će započeti, potrajeti izvesno vreme, a zatim će računar, izima programu, ispisati njegovu dužinu u bajtovima, kao i „load“ i „execution“ adresu programa. Adresa od koje se program učita, i izvraća adresu imaju na

čini postavljanja (no), jer će se po prvi put javno objavljivati) može da se pripremi kontrolni blok za snimanje programa koji omogućava pravljenje kopija. Čitava ova teorija je sažeta u program PREPIS koji objavljujemo. Asemblirajte ga i snimite na traku (pre kucanja sors programa otukajte PAGE = &3000, a posle asembleriranja — "SAVE PREPIS", EOF + &R0). Pro-

PREPIS, na žalost, ima ograničeno dejstvo: pomoču njega ne možete da učitate programe Hopper, Starship, Command i još nekoliko drugih. Osim toga, program „The Hobbit“ je toliko dugačak da ga PREPIS vrlo teško prepišije. O načinu na koga se prenimači i najbolje zaključani (kada se „LOAD“ poskušamo da pročitamo posebno zaščiten program, dobicemo poruku „Locked“) dugački programi govorimo, ako za to bude interes, u Računarima 3. Tada ćemo objaviti i poboljšanu verziju programa PREPIS, koja se zove DCOPY.

Toga dana sve vam je islo savršeno — učitati ste svoju najomiljeniju igru i, iz prvog pokusaja, postigli svoj apsolutni rekord — došli ste do 12. ekrana! Igra te je dalje, nadate se da ćete stići i do sledećeg kad čujete zvono na vratilima (na telefon se, naravno, ne biste obazirali). Šta da radite? Pominjajte sve lepe reči kojih se setite, pritisnite BREAK i odlazite do vrata da biste ustanovili da neko želi da pita gde je stara porodice... Vas najbolji skočni svih vremena je, naravno, otisao u nepovrat.

Mora li da bude baš tako? Naravno da ne mora ako poseduje program „Univerzalna pauza“. Učitajte ga pre učitavanja bilo kog (obziljno mislimo bilo kog) drugog programa i kada želite pauzu, pritisnite tastu **(„biznis a“ ili, kako ga naši programeri nazivaju, „majmunski znak“).** Ekran će se potpuno „zamrznuti“ — jedino će se (eventualno) čuti kontinuirani ton. Kada poželite da nastavite, pritisnite bilo koju tastu (na primer, onaj koju trebate da oznaci vašu slediće akciju) i produžite sa igrom.

Kako „Univerzalna pauza“ može da bude univerzalna? Principi ovog programa su prirođeno jednostavniji, ali češće pomisili da oni odlaže preduzimanje u „mračne vode“ operativnog sistema. Väs računar radi u režimu stalnih prekida (interrupti) – svaki put kada, na primer, pritisnete neki tastir mikroprocesor dobije maskiranu interupt da bi ga registrovao. Poseban vid prekida su takođe zvani dogadjaji (events) koji su uobičajeno u informativne prirode u kojima računar u normalnoj situaciji ignorise. Pomoću „FX 14“ (ili kako je urađeno u ovom programu, odgovarajućeg OS/2 YTE poziva) neki od dogadjaja mogu da budu omogućeni, a oni tom slučaju mikroprocesor kompjutera.

risti EVNT vektor da odredi adresu na koju će „skočiti“ kada dogodaj nastupi. U našem slučaju omogućen je dogodaj 2 koji, kako piše u uputstvu, nastupa kada je pritisnut neki taster. U tom slučaju, izvršava se PROG; u akumulator je u broj 2 (broj dogodaja), a u registru ASCII kod pritisnutog tastera. U okviru tretranja dogodaja potrebno je, najpre, sačuvati sve procesorske registre na steku. Posle toga nastupi pauza koja je neophodna da bi korisnik otpustio taster @. Zatim računar skanira tastaturu bez korišćenja rutine operativnog sistema — za vreme „dogodaja“ interpret je onemogućen pa one rutine ne funkcionišu.

šu. Ovo skaniranje je omogućeno testiranjem područja SHELLA koje je veoma nekompletno obrađeno u okviru uputstva za upotrebu. Ukoliko je detektovan pritisak na bilo koji tastir, računar sa steke skida potrebne registre i, pomoću RTS, vraća kontrolu programu koji se izvršava pre pritiska na @.

Program možete da otčukate kao i svaki bežič. Posle inicijalizacije program je neprekidno u računaru sa kojim možete da radite sasvim normalno; program će biti obrisan jedino pritiskom na CTRL BREAK, dok je posle BREAK potrebna nova inicijalizacija koja se postiže sa CALL & BOO

OLIVETTI M20

Fabrika Olivetti ima dugu tradiciju u proizvodnji lichenih profesionalnih računara. Tako je još pre dvadesetak godina bila prva sa „programom“ P101 računarom, koji je imao 4 K memorije, zasnovane na trizoliji opruge. Nastavljajući tradiciju u svojoj filijali u Cupertino u Kaliforniji, Olivetti je 1979. godine otpočeo razvoj novog lichenog profesionalnog računara. Rezultat, M20 zvanično je predstavljen u proljeće 1982. godine. Proizvodi se u Italiji, a u nas ga zastupa RO DINARA iz Beograda.

HARDWARE:

M20 je zasnovan na Zilog-ovom potpuno 16-bitnom mikroprocesoru, Z 8001, koji radi na frekvenciji 4 MHz. Za njega je napisan poseban operativni sistem PCOS (Professional Computer Operating System).

Mikroprocesori iz serije Z 8000 omogućavaju: segmentirano adresiranje do 128 segmenta po 64 KB (odnosno 8 MB), virtuelnu memoriju, 32-bitnu aritmetiku, dinamičku dodelu memorije, multi-programiranje, multi procesorski rad i još dosta drugih svojstava do nedavno korišćenih samo kod većih sistema. Zajedno sa M20 paralelno je razvijena celu seriju od M10 do M60 zasnovana na novoj tehnologiji.

Taj razvoj se i dalje nastavlja i gotovo svakodnevno svojstva i mogućnosti sistema rastu. Osnovni memoriski kapacitet je 128 KB RAM-a, a moguća su proširenja sa priručnjima po 32 KB ili 128 KB. Ukupno za sada M20 može imati do 0,5 MB, odnosno korisničkih 421 KB neto. Disketne jedinice od 5 1/4 inča, mogu da budu od 160, 320 ili 640 KB neformatovano.

Winchester disk sa 11,7 MB neformatovano, moguće je priključiti kao dodatnu periferijsku jedinicu ili ugraditi jedne diskete. Ekran od 12 inča može da bude crne beli sa standardnim ili zelenim fosforom ili u boji sa 4 ili 8 boja. Grafika je sa 512×256 tačkica i 16 prozora. Struktura broja linija i znakova u redu je softverski promenljiva i može da bude 16×64 ili 25×80. Tastatura ima poseban numerički deo a



funktionski tasteri zamjenjeni su kombinovanjem tastera, pa je moguće generisati do 256 znakova ili komandi. Postoje naši znaci i tastatura u skladu sa JUS-om.

Osnovna verzija M20 uključuje 1 paralelni priključak za štampač i jedan serijski RS 232. Kao dodaci mogu da budu ugrađeni: IEEE488 sa svim najviše 14 periferikama, kao i još najviše 4 RS 232 priključka, odnosno 4 strujne petlje. LAN (Local Area Network) priključak omogućava grupno korišćenje

kapaciteta, na primer: datoteka na disku, za najviše 64 računara M20 povezanih u mrežu.

Od velikog broja raznih periferijskih uređaja štampača, plotera, traka, kasete, bušača čitača, itd., koji je moguće priključiti na M20, posebno treba izdvojiti matrični štampač PR 15 sa 120 znk/sa širinom od 8 inča 80 do 132 znaka u redu sa gustošćom 10, 12 i 16 znk/inč ploterskim mogućnostima otiskom povećane vernošću i paralelnim Centronics ili serijskim RS 232 priključenjem.

Za povezivanje sa drugim sistemima na raspolaženju su protokoli: asinhroni npr.: TTY33, sinhroni BSC npr. IBM 2780, bit orijentisani npr.: IBM 3275 i X25.

SOFTWARE:

Radi kompatibilnosti sa softverom drugih proizvođača, M20 koristi proširenje, elektronsku ploču sa INTEL 8086, 8 MHz, a postoje i verzije sa oba procesora na osnovnoj ploči. Tako je omogućeno korišćenje i operativnih sistema: MS/DOS (kao IBM PC), CPM/80 (kao i APPLE), CPM/86 i UCSD-p.

Sistemske Software pod PCOS-om obuhvata Assembler, Basic (Microsoft, interpreterski, za oko 50% brž od IBM PC-a), Pascal i Fort. Pod ostalim operativnim sistemima mogu da se koriste mnogi jezici, uključujući FORTRAN, kao i BASIC sa prevodećim.

Proizvođač nudi mnogo gotovih paketa od kojih posebno treba istaći:

MULTIPLAN — usavršena verzija VISICALC-A (MICROSOFT) za analizu i modelovanje

OLIWORD — obrada teksta (postoji verzija prevedena na srpskohrvatski)

OLICHART — grafički prikazi rezultata Multiplan analize

OLISORT — sortiranje

OLITERM — komunikacija asinhrona

OLICOM — komunikacija sinhrona

OLEINTRY — unos podataka i obrada

OLISTAT — statistički paket

OLINUM — numerička analiza

OLIS275 — komunikacija RS 3275

Applikacioni software pored korisnika radi i oko 400 specijalizovanih softver-skih kuća, za koje Olivetti periodično publikuje katalog, a u nas je urađen jedan iz interaktivne grafike.

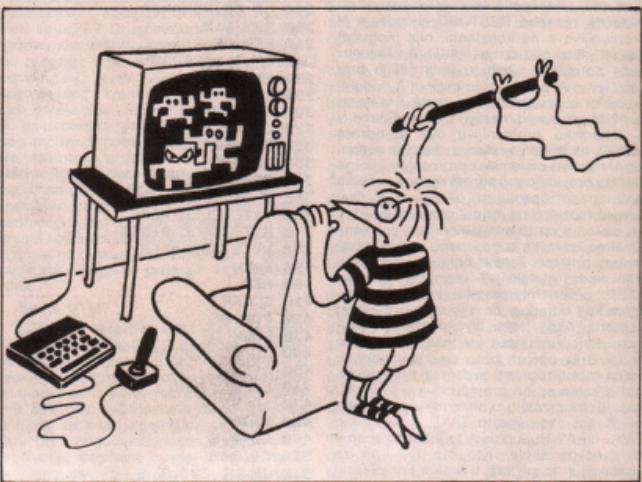
Dokumentacija za M20 u ovom trenutku kod zastupnika obuhvata biblioteku od 56 knjiga, koja raste po stopi od 2-3 mesečno, pa ovaj prikaz lichenog profesionalnog računara M20 treba shvatiti kao presek jednog vrlo dinamičnog procesa.

„galaksija“ bez tajni

Majstorije
na računaru

Uputstvo za upotrebu računara „galaksija“ koje smo objavili kao podlistak specijalnog izdanja „Računari u vašoj kući“ bilo je namenjeno prevashodno početnicima. Osim njih, „galaksiju“ su napravili i mnogi koji su „prevazišli bejzik“ i poželeti da pišu i mašinske programe. Njima, osim korisne mere sistemskih promenljivih, uputstvo nije pružalo ništa, nismo mogli da damo čak ni adresu korisnih rutina u ROM-u jednostavno zašto što one nisu bile fiksirane kada je uputstvo pisano. Za njih smo ovoga puta pripremili obilje programske finesa za napredno programiranje računara „galaksija“.

Čim je razvoj ROM-a završen, ponudili smo svima zainteresovanim njegov disasembliрani listing (i dalje možete da ga dobijate ako na žito račun „Galaksiju“ uplatite 200 din i pošaljete nam potvrdu o uplati uz napomenu da želite program GALA005052). On je praktično trenutno postao bestseler, ali smo ipak bili svesni da svi oni koji su ga naručili neće moći da izvuku mnogo korisnika od njega. Da biste razumeli operativni sistem i bejzik interpretator jednog računara (čak i ako, kao u našem slučaju, taj program ima samo 4 Kb), morate da poznajete mašinski jezik izuzetno dobro i užito mnogo rada, strpljenja i, što je najvažnije, dosta vremena. Zato smo odlučili da vam u „Računarama 2“ omogućimo da podete precicom: izdvojili smo sistemске adrese koje su nam se činile najkorisnijim, obradili kompletну aritmetiku pokretnog zarezra, proučili da vas korišćenje linkova za naredbe, video i (do sada nedokumentovanog) linka u interrupt i sve to sumirali u sledećih nekoliko strana. Obzirom da je sada pisanje mašinskih programi za „galaksiju“ postalo moguće na jedan od tri načina (o njima nešto docnije), nadamo se da će podaci koje dajemo biti iskorisćeni na najbolji mogući način: za pisanje sistemskih i uslužnih programa i, naravno, efektnih igara.



Karakture Milos Artovic

nalazi iole duži bejzik program, nije ni malo jednostavno pronaći neko određeno slovo u njemu da bismo ga, pomoću BYTE, promenili. Ako pre toga slova stavimo jednostavno PRINT PTR i izvršimo program, direktno ćemo dobiti traženu poziciju (ne treba zaboraviti da će se ta adresa smanjiti za 9 bajtova aka izbrišemo naredbu PRINT PTR koja, računajući i blanko simbol, ima isto toliko slova).

Navedimo jedan primer upotrebe naredbe PTR. Dok lista program, računar „galaksiju“ ne očekuje da naide na liniski broj 0. Ukoliko ipak nađe na njega, računar smatra da je listanje završeno i vraca se u komandni mod prikazavši READY. Kako da kreiramo liniski broj 0? Ako želimo da prva naredba programa ima taj broj, ništa lakše: otkučamo WORD WORD(&2C36). 0, pa će „galaksija“ listati samo prvu liniju programa u kojoj može da piše ime njegovog autora ili, možda, A=USR(&2C3A) što bi stvorilo utisak da je program pisan na mašinskom jeziku. No, šta ako želimo da se listanje programa prekine posle neke druge linije? U prethodnu liniju ćemo, na samom njenom kraju, dodati naredbu PRINT PTR, a

zatim je i izvršiti. Dobićemo adresu prvog bajta iza slova R (od PTR). To je bajt &D koji označava kraj programske linije. Iza njega, kao što ćemo videti, stoji liniski broj sledeće naredbe koji možemo da promenimo u 0 primenom naredbe WORD. Ne treba da zaboravimo da obrišemo PRINT PTR, naredbu koja je obavila svoju funkciju. Ukoliko bismo ovu naredbu obrisali pre nego što promenimo liniski broj slediće, morali bismo ponovo da preračunavamo adresu jer se sadržaj memorije pomerio.

Naredbu PTR ćete mnogo više koristiti (opisani metod je prilično „opasan“ jer „galaksiju“ sistemi traženja liniskih brojeva kod GOTO i CALL naredbi očekuju da ovi brojevi rastu) za pisanje programa koji menjaju samog sebe. Jedan od takvih je i „Telefonski imenik“, ilustracija Interfejsa 1, koji je objavljen u marsovskoj „Galaksiji“. Ako se odlučite na sličnu „avanturu“, vodite računa da adresu početka i kraja bejzik programa mora stalno da bude smještena u sistemskim promenljivim &2C36 i &2C38. Odstupanje od ovoga može da ima nepriyatne posledice po GOTO naredbe i, posebno, ispravke pojedinih linija.

PTR

Dvadeset šesto poglavje Uputstva za upotrebu računara „galaksija“ bilo je posvećeno naredbi PTR. Ona, da podsetimo, daje memorisku adresu neke numeričke ili alfaniumeričke promenljive (PRINT PTR X\$, na primer, daje rezultat 10864, što je isto što i &2A70 prema mapi sistemskih promenljivih). No naredba PTR može da se koristi i bez adresnog dela, npr. A=PTR ili PRINT PTR (poslednja modifikacija „galaksijinog“ ROM-a pred njegovom „zaključivanje“ 3. januara 84). U tom slučaju, naredba PTR daje memorisku adresu na kojoj je sama smještena. Šta će nam to? Ako se u memoriji

TAKE

Devetaesto poglavje Uputstva govori o naredbi TAKE koja je mnogim vlasnicima drugih računara poznata pod imenom READ. Radi uštеде prostora u ROM-u, u bežiku uobičajena naredba RESTORE je zastupljena samo kao poseban slučaj TAKE. Ovoj mogućnosti smo do sada posvetili samo jedan pasus, propuštajući da spomenemo jedan interesantan trik koji je zamislen još kada je TAKE koncipirana. Ako se između TAKE nalazi neki broj n (npr. TAKE A,B,100,C) „galaksija“ će potražiti liniju sa tim brojem i postaviti pointer naredbe TAKE na njen početak. Ukoliko se u toj naredbi nalazi (DATA) LISTA, SLEDEĆE TAKE će uzeti podatak iz nje. Ukoliko se u naredbi uočiće ne naredbi lista, računar će potražiti sledeću naredbu, zatim sledeću...

Ovakva struktura naredbe TAKE je prilično fleksibilna i omogućava uštědu RAM-a: sve se dešava u istoj listi. Na prvi pogled, koncepcija ima veliku manu: simulacija naredbe RESTORE N, gde je N promenljiva u ne konstanta, nije moguća! Zaista, ako otkucamo TAKE A, računar će potražiti listu u liniji čiji je broj smešten u A, već će promenljivo A dodeliti vrednost iz prve sledeće liste. Šta da se radi? Mogli bismo, naravno, da smestimo N u sistemsku promenljivu čiju je adresa &2A9D, ali je takvo rešenje „leb sa sedam kora“: u ovu sistemsku promenljivu se ne smesta broj linije već adresu njenog početka posmatrano absolutno u memoriju. Ovu adresu možemo da odredimo pomoću PTR, ali moramo da je menjamo svaki put kada pravimo ispravke u programu, što nije baš mnogo prijatno. Zato je bolje primeniti ovaj trik: kada „galaksija“ razmatra naredbu TAKE, bežik interpretator koristi potprogram koji određuje da li je sledeći element numerički. Ako jeste, biva pozvan potprogram koji izračunava vrednost izraza (ako je taj izraz običan broj, vrednost koja se vrati iz potprograma je sam taj broj, naravno) a zatim se on preračunava u vrednost koju treba smestiti u sistemsku promenljivu. A šta ako umesto TAKE A napišemo TAKE 0+A? Nula plus A je, kao što znamo iz osnovne škole, isto što i A, ali za „galaksiju“ to ne važi u svakoj konstellaciji dogadjaja. Pri analizi naredbe računar naiđe na broj nula koji prepoznaće kao numerik, podrazumevajući da se radi o simularnoj naredbi RESTORE. Potprogram koji izračunava vrednost izraza, međutim, nije tako lako prevariti i on će svakako razumeti da je 0+A isto što i A, pa ćemo dobiti naredbu RESTORE A!

```

10 !
20 !           CLEAR
30 !
40 !           DAJE VREDNOST NULA
50 !           NUMERICKIM PROMEN-  
60 !           LJIVIMA A-Z
70 !
80 <
90 ORG &3000
100 OPT 3
110 LINK EQU &2BA9
120 KOMANDA EQU &75B
130 PREPOZNAJ EQU &39A
140 DFILE EQU &2800
150 NUMVAR EQU &2A00
160 LD A,&C3 ! &C3 = JUMP
170 LD (LINK),A
180 LD HL,PROG
190 LD (LINK+1),HL
200 RET
210 PROG
220 EX (SP),HL
230 PUSH DE
240 LD DE,KOMANDA
250 RST &10
260 POP DE
270 JR Z,NAREDBA
280 EX (SP),HL
290 RET
300 NAREDBA
310 LD HL,TABLICA-1
320 JP PREPOZNAJ
330 TABLICA
340 TEXT "CLEAR"
350 BYTE CLEAR>B+&B0
360 BYTE CLEAR#&FF
370 BYTE BACK>B+&B0
380 BYTE BACK#&FF
390 BACK
400 RET
410 CLEAR
420 POP AF
430 PUSH DE
440 LD DE,NUMVAR
450 CYCLE
460 LD HL,ZERO
470 LD BC,4
480 LDIR
490 LD A,E
500 CP &68
510 JR NZ,CYCLE
520 POP DE
530 RST &30
540 ZERO
550 WORD &0018
560 WORD &4000
570 >
580 A=USR (&3000)

```

KEY

Odmah iza naredbe TAKE ćemo sada o naredbi KEY uz odgovarajuću tabelu u kojoj su se nalazila dva provokativna prazna mesta pa su se mnogi verovatno zapitali šta znači KEY(49) i KEY(51). Pripremajući tabelu, imali smo puno razloga da izostavimo pomenuta dva koda jer u bežiku mogu samo da izazovu zabunu, ali ćemo sada govoriti o njima pošto mogu da budu od velikog značaja pri pisanju mašinskih programi.

KEY(49) testira taster BRK, a KEY(51) taster DEL. BRK nema mnogo smisla testirati jer pritisak na njega prekida izvršavanje programa (otkucajuće 10 PRINT KEY (49):GOTO 10 i startujte program pokušavajući da pritisnete BRK samo na kratko,

toliko kratko da ga KEY registruje, ali da se program ne prekine. Nije lako ali je moguće!

Testiranje DEL je još beznadežniji slučaj: pritisak na njega prekida izvršavanje programa i taj prekid traje sve dok korisnik ne otpusti DEL. U mašinskom programu, međutim, svakom od ovih tastera možete da dodelite proizvoljnu funkciju ukoliko ne koristite potprogram za testiranje tastature iz ROM-a. U mašinskim programima, stavlješ, mogu da se testiraju i pritisci na SHIFT BRK (npr. u „Zamku“ i „Jumping Jack-u“), programima koji su emitovani preko Videlatora, ova kombinacija se koristi za prekidanje igre i novi start) i slične „nemoguće“ kombinacije.

Naredba KEY(0) koristi potprogram za rad sa tastaturom iz ROM-a, što znači da se javljaju isti problemi (sasvim očekivani, uzgred rečeno) sa testiranjem BRK i DEL, ali i jedna nova zagonetka: ako slučano pritisnemo STOP/LIST dok se izvršava KEY (0), počće listanje programa i, samim tim, prestati njegovo izvršavanje. Dalje ispitivanje bi nas uverilo da sličan artefakt postoji i kod naredbe INPUT. Ništa ozbiljno, ali o tome treba voditi računa u foku pisanja bežik programa.

INPUT

Kada smo već kod naredbe INPUT, potrebno je da odgovorimo na jedno često postavljano pitanje: ako koristimo naredbu INPUT X\$, tekst koji kucamo ne mora da bude pod novadnicima. Kada su, zapravo, novadnici obavezni? Najpre u \uparrow listama (što se vidi i iz spiska naredbi sa primerima iz uputstva za upotrebu), a zatim i u naredbama za dodeljivanje koje bi mogle da izazovu zabunu. Možemo, dakle, da napišemo X\$ = GALAKSJA, ali ako u X\$ želimo da smestimo slovo Y i \$, moraćemo da iskoristimo X\$ = „Y\$“ jer X\$ = Y\$ ima drugi smisao! Kod „numeričkog INPUT-a“, imamo puno pravo da otkucamo neki aritmetički izraz umesto obične konstante „galaksija“ će ga izračunati i dodeliti promenljivoj koja je navedena u INPUT-u. U tekstu koji kucamo mogu da se koriste i promenljive i funkcije, ali tako da sintaksu bude ispravna; u protivnom, „galaksija“ će ispisati WHAT? ili HOW? (zavisno od prirode greške) i prestati sa radom. U toku INPUT-a, dakle, biva izvršena i naredba VAL.

VAL

Naredba VAL je neobično moćna i treba je koristiti kad god je to moguće. Potreban nam je, na primer, program koji pretvara heksadekadne brojeve u dekadne. Možemo, naravno, da izdvajamo slovo po slovo iz promenljive X\$, pretvaramo ga u broj i primenjujući uobičajeni algoritam množenja sa 16, ali postoji i daleko jednostavniji način: otkucaćemo:

```

10 INPUT X$  
20 Y$ = „.„+X$  
30 PRINT VAL(PTR Y$) i — to je sve. Ako u odgovoru na INPUT otkucamo 2AO, primenjujući Y$ će dobiti vrednost &2AO a na ekranu će biti stampan broj 522, prevod 2AO u dekadni sistem. Ako pišete program koji će vam omogućiti da unosite mašinski program kao „.hex-dump“ u memoriju, program poput ovog će vam uslediti čitavu trecinu kucanja podataka!
```

Postoji još mnogina načina da lukav

iskoristite polprograme operativnog sistema bez potrebe da se pozavabite mašinski skim jezikom. Za pripremanje pomenutih heks-dampova koji bi bili pogodni za listanje i objavljuvanje u „Galaksiji“, bilo je neophodno da sastavimo program koji će dati mašinski program pretvarati u naredbe. Za to je bilo potrebno pretvarati numerike dobijene pomoću BYTE-ov ASCII vrednosti i od njih kreirati listu (program koji samoge se menja). Za ostvarivanje loga postupili smo se malim trikom: obrisemo ekran sa HOME, štamparamo vrednost koju treba pretvoriti u ASCII karaktere a zatim iz same video memorije (od &2801 pošto se na &2800 nalazi znak koji je u neftnosti koje su nam potrebne (od 255) uvek blanko jer su brojevi pozitivni) čitamo jednu po jednu cifru običnim BYTE-om.

Korišćenje rutina operativnog sistema, naravno, nosi sa sobom i odredene opasnosti da će nas računar „pogrešno shvatiti“. Svi oni koji su napisali više mašinskih programa počeli su da „misle heksadekadno“, tj. da koriste heksadekadni sistem za predstavljanje svih programerskih problema. Za njih će, dakle, biti sasvim prirođeno da umesto NEW 1024 otkucaju NEW &400 rezervišući kilobajt za mašinski i program koji će biti smesten neposredno ispred bežika. Računar „galaksija“ ih, na žalost, neće razumeti! Iza NEW i OLD se, naime, mora nalaziti numerik, a ne izraz NEW &400 čit biti shvaćeno kao obično NEW i računar, da stvar bude gora, neće to prijaviti ni na kakav način. Doinje može da dode do mešanja bežika i mašinskog programa sa katastrofnim posledicama po oba. Ako, dakle, koristimo neku opciju koja nije dokumentovana, moraćemo napraviti da preverimo njenje dejstvo.

Minsko polje

Skijaše-početnike uče da se zaustave pre nego što nauče da se krenu niz padinu, pa cemo i mi, pre nego što počnemo da govorimo o mašinskim programima izložiti koje memoriske zone i sistemske registre treba po svaku cenu čuvati ili, bar, vratiti u prvobitno stanje.

Postoje dvije vrste mašinskih programa: jedni se koriste u bežiku kao polprogrami, što znači da, posle njihovog izvršavanja, „galaksija“ treba da se vrati u bežik. Da bi se takva kooperacija ostvarila, potrebno je, očigledno, dobro čuvati sve ono što je interpretatorov radni prostor. Postoje, sa druge strane, i mašinski programi sasvim nezavisni od bežika i ROM-a koji, na primer, predstavljaju interpretatore drugih programskih jezika ili (mnogo verovatnije) kompletne igre. Reklo bi se da ovi programi ne moraju ništa da čuvaju i da mogu proizvoljno da se prostiru u memoriju. No, odmah ćemo videti da nije baš tako.

Treba, pre svega, da razumemo da računar ne izvršava naš program sve vreme. Sveke pedesetinke sekunde procesor prekida rad na našem programu i pomaze video-stepenu da generiše sliku na ekranu monitora. Ovu rečenicu pišemo već po ko zna koji put ali za to imamo mnogo razloga: iz liničnog iskustva nam je poznato da je vrlo teško, naci grešku koja je nastala, kao posledica interalta koji programer često

KAKO SE UNOSE MASINSKI PROGRAMI

U okviru poglavljia koje je posvećeno malim tajnama računara „galaksije“ dajemo dosta primera koji treba da ilustruju opisivane tehnike. Neki od tih primera, međutim, mogu da budu interesantni i za uslužni programi koje biste rado uneli u računar i koštili u raznim prilikama (ovde najpre mislimo na programe RSAVE i TOME). Za to unošenje će poslužiti hex-dump svakog od tih programi a kratka rutina UTM koju ovde donosimo.

Mašinski program se u računar unosi na jedan od tri načina: pomoću UTM-a ili nekog sličnog programa, pomoću asemblera ili ilikog krajnja soluciјa, preko asemblera nekog drugog računara koji je povezan sa „galaksijom“. Kada ćemo koristiti koju od ovih soluciјa? Drugi računar je neobično zgodan ako pišemo komplikovanij sistemski program (npr. ROM 2) koji će, u toku razvoja, više puta „krahirati“. Posle svakog ovakvog „krahira“, moraćemo da brišemo „galaksijinu“ memoriju pri čemu izvorni, asembleri, program u memoriji drugega računara neće biti izgubljen. Korišćenje asemblera programa je zgodno kada pišemo igre ili neke druge uslužne programe, i, naročito, kada posedujemo ovakav asembler. Program UTM ostaje kao rešenje neobično zgodno za unošenje gotovih programa koji su objavljeni u nekom časopisu i čije nas modifikacije ne zanimaju — želimo samo da otkucamo program i da ga korišćimo.

Šta je UTM?

UTM je, naravno, Ultra Tiny Monitor. Kada pogledate koliko je dugaćak, verovatno ćete pomislići da ni ono „Ultra Tiny“ u nazivu ne ilustruje dovoljno njegove slabosti u odnosu na prave monitor programe. Ali nije baš tako — videćete da je UTM, za primene za koje je napisan, sasvim dovoljan. On omogućava unošenje programa u proizvoljni segment memorije, testiranje i promenu sadržaja proizvoljne memoriske lokacije i kontrolu sadržaja većih segmenata.

```

5 CALL 248
10 HRR$K4)
20 Y$=""
30 I=0
40 X$=""
50 H=I:CALL 160
58 PRINT "";
59 H=BYTE(I):CALL 160
68 PRINT " ";
69 INPUT X$;
70 IF E0 X$,Y$ I=I+1:GOTO 50
71 IF (BYTE(PTR X$+1)=0)*(BYTE(PTR X$+2)=0) CALL 130-BYTE I,H:X$="":GOTO 100
72 CALL 130-I:H:GOTO 40
73 X$()="":X$()
74 H=VAL(PTR X$())
75 RETURN
76 FOR K=4 TO 1 STEP -1
77 G=H
78 H=INT(H/16)
79 G=G-16*H
80 IF G<10 X$()=CHR$(G+48):ELSE X$()=CHR$(G+55)
81 NEXT K
82 FOR K=1 TO 4 PRINT X$():NEXT K
83 RETURN
84 HOME
85 PRINT AT 400;"GALAKSIJA"
86 PRINT
87 PRINT "UTM .VERSION 1.0"
88 PRINT
89 RETURN

```

ta memorije. Evo uputstva za njegovu upotrebu.

Po startovanju programa, „galaksija“ ispisuje njegovo zagлавje i upitnik. Kada god se pojavi upitnik možete da izaberete jednu od sledećih opcija:

Ukoliko otkucavate neki četvorocifren heksadekadni broj, kome ne smete da prethodi oznaka &, računar će pokazati sadržaj memoriske celije čija je adresa datim brojem i proglašiti tu adresu za „tekuću“.

Ukoliko otkucavate neki jedno ili dvocifren heksadekadni broj, kome ne smete da prethodi oznaka &, računar će taj broj smestiti u memorisku celiju čija je adresa u tom momentu označena kao „tekuća“. Zatim će automatski biti izvršena sledeća opcija.

Ukoliko jednostavno pritisnete RET odnosno ENTER, tekuća vrednost će biti počevana za jedan, pa će na ekranu biti prikazan sadržaj te memoriske celije koju zatim slobodno možete da menjate primenom opcije 2.

Kako da otkucate neki mašinski program pomoću UTM-a? Startujte UTM sa RUN (ne smete da zaboravite da, ukoliko mašinski program počinje od &2C3A, treba da pomerite UTM kucajući NEW NNNN pre kucajanja UTM-a ili NEW NNNN OLD NNNN WORD &2C38, NNNN +&2E3E pre njegovog unošenja sa kasete; NNNN je bar za jedan veći broj od dužine mašinskog programa, pri čemu treba zaboraviti da NNNN mora da bude dekadni broj), otkucajte početnu adresu programa a zatim, bajt po bajt, sadržaje svake memoriske lokacije. Koristite stalne heksadekadni sistem, koji je uobičajen kod programera. Posle unošenja poslednjeg bajta, možete da pritisnete BRK i tako prekinete rad ili ponovite početnu adresu i, uzastopnim pritiscima tastera ENTER, proverite sadržaj memorije koji treba da odgovara hex-dampu. Ukoliko pronađete grešku, lako ćete je ispraviti kucanjem novog sadržaja.

potpuno zaboravlja jer se dešava van njegovog učijaja. Da bi računar korektno izvršio interapt, nužno je da ni po koju cenu ne menjamo sadržaj procesorskog Y registra. Ukoliko to učinimo (pa makar se restaurirali samo nekoliko naredbi docnije), rizikujemo da računar posle interapta počne da izvršava program na nekom drugom mestu u RAM-u pa čemo, pogolovo ako je prostor sistemskih promenljivih poremećen, izazvati siguran krah pratećim reinicijalizacijom računara i gubitkom programa.

Ma koliko se trudili da naš program bude nezavistan od bežizka, za njegovo testiranje će biti korisno da se ipak oslonimo na njega. Nismo toga ko će napisati sistemski program tako da proradi „iz prve“. Ukoliko se program razvija na „galaksiji“, posle svake probe čemo izgubiti program ukoliko ne obezbedimo mogućnost da bežizak interpretator preuzeme kontrolu. Čak i ako program razvijemo na nekom drugom računaru, požećemo da koristimo potprograme iz ROM-a, a zahtevaju odredene predostrožnosti.

Stek mora da vam bude „najveća svetinja“. Ukoliko vaš program ima, u nekoj grani, više naredbi PUSH nego naredbi POP, povratak u bežizak će biti nemoguć, jer RET nalazi adresu za povratak na vrhu steka. Isto tako, ako se stek stalno puni, računar će se garantovano „ugušiti“ posle izvesnog vremena. Simptomi ove greške su programi koji jedno vreme ispravno rade, a onda nastupa „krah“ pratećim brišanjem čitave memorije.

Dakle, svaka operacija sa stekom, a naročito direktna izmena registra SP, treba da bude pod plaćivog planiranja (ovim, naravno, nikako ne želimo da kažemo da ne treba koristiti stek — bez njega se ne pišu ni najkraci programi)koje će imati za cilj da sve bude vraćeno u prvobitno stanje. Vodite, na primer, računa o tome da poziv potprograma iz koga se ne vraćamo dodaje dva broja na stek. Kao i svako pravilo, i ovo ima izuzetaka: ako se ne vratace u bežizak pomoću RTN već pomoću JP & 66, ne morate previše da brinete o steku: posle ovog skoka „galaksija“ ispisuje READY i reinicijalizuje registar SP (u programerskom žargonu ova adresa se zove „farmu“, pa bi zagrizeniji hakeri predhodnu rečenicu izgovorili kao „ne brini o steku ako računar posle ide na farmu“).

Osim steka adresiranog registrom SP, „galaksija“ ima još jedan stek adresiran registrom IX — aritmetički stek čiju ćemo strukturu izložiti kada budemo govorili o aritmetici u pokretnom zarezu. Ovaj stek, ukratko, sadrži argumente aritmetičkih operacija i ne treba ga previše remetiti. Ponekad će nam, naravno, zatrebati registar IX (pošto Y i tako ne smemo da koristimo), pa čemo ga najpre sačuvati (sa PUSH IX), onda koristiti i najzad vratiti u početno stanje (POP IX).

Sistemski promenljive

Što se područja sistemskih promenljivih tiče, ne treba menjati poziciju kurzora u memoriji tako da se on nađe van područja &2800—&2A00, poziciju kraja memorije (osim ako ne ugradujemo proširenje memorije), bežizak pointer, poziciju tekuće linije, linkove za naredbe i video (osim na

```

&3000: 3E C3 32 A9 2B 21 0C 30
&3000: 22 AA 2B C9 E3 D5 11 77
&3010: 07 D7 D1 2B 02 E3 C9 21
&3018: 1C 30 C3 9A 03 44 49 53
&3020: 50 4C 41 59 B0 29 B0 28
&3028: C9 F1 CD 6A 0A D5 11 00
&3030: 2B 19 D1 7E 6F 26 00 C3
&3038: BC 0A

```

```

10 !
20 ! DISPLAY (PPP)
30 !
40 ! DAJE ASCII KOD KARAKTERA
50 ! U VIDEO MEMORIJI CIJA JE
60 ! ADRESA <PPP> PREMA MAPI
70 !
80 <
90 ORG &3000
100 OPT 3
110 LINK EQU &2BA9
120 FUNKC EQU &777
130 PREPOZNAY EQU &39A
140 DFILLE EQU &2B00
150 KRAJHL EQU &ABC
160 IZRAZ EQU &A6A
170 LD A,&C3 ! &C3 = JUMP
180 LD (LINK),A
190 LD HL,PROG
200 LD (LINK+1),HL
210 RET
220 PROG
230 EX (SP),HL
240 PUSH DE
250 LD DE,FUNKC
260 RST &1 ! POREDI HL I DE
270 POP DE
280 JR Z,NAREDBA
290 EX (SP),HL
300 RET
310 NAREDBA
320 LD HL,TABLICA-1
330 JP PREPOZNAY
340 TABLICA
350 TEXT "DISPLAY"
360 BYTE DISPLAY#&FF00>B+&B0
370 BYTE DISPLAY#&FF
380 BYTE NEGDE#&FF00>B+&B0
390 BYTE NEGDE#&FF
400 NEGDE
410 RET
420 DISPLAY
430 POP AF
440 CALL IZRAZ
450 PUSH DE
460 LD DE,DFILE
470 ADD HL,DE
480 POP DE
490 LD A,(HL)
500 LD L,A
510 LD H,0
520 JP KRAJHL
530 >

```

sasvim specijalan način, o kome ćemo govoriti nešto docnije) i pointere početka i kraja bežizka. Promena drugih sistemskih promenljivih ne može da ima katastrofalne posledice, ali može da izazove male nepriznatosti (ako, na primer, promenite sadržaj sistemskih promenljivih &2BA8, slika će se pomeriti što će teško ispraviti).

Naime zeleli da pominjete da je najbolje ne dirati sistemski promenljive: pisanje mašinskih programi, pogolovo ako se radi o rutinama sistemskih prirode, ne može ni da se zamisli bez intervencija u ovom bloku memorije. Hoćemo samo da vas upozorimo da svaka promena neke od sistemskih

promenljivih mora da bude rezultat razumevanja njene funkcije i situacija kada se funkcija ostvaruje.

Pažljiv pogled na tabelu sistemskih promenljivih olfkira da u njemu ima neiskorišćenog prostora — bajtovi &2A97, &2A98, &2AAA i &2A90 nisu korišćeni, pa bi mogli lepo da se iskoriste za smestaj nekih podataka koji su našem programu potrebiti. Da li je pametno koristiti ih? Nismo sigurni — računar „galaksija“ će biti i dalje proširivan, pa je sasvim moguće da će neka od varijanti ROM-a ili iskoristiti prazan sistemski prostor. U tom slučaju, moraćete da preradujete vaše programe kako bi ostali kompatibilni i sa proširenjem računara. Teško da će vaš program biti baš toliko dugog da mu zatafne četiri bajta, pa vam savetujemo da pristor za podatke potražite na nekom drugom mestu. A jedno od takvih mesta je već poznato — bajfer koji se prostire od &2BB6 do &2B36. Ovih 126 bajtova su prostor koji se u toku izvršavanja programa svakako ne koristi, pa može zgodno da posluži kao zona u koju bi se privremeno smestili podaci. Zašto kazemo privremeno? Ukoliko se izvršavanje programa prekine, svakako će biti potrebno da korisnik otkuca nešto da bi program nastavio da radi. To nešto biva smesteno u bajfer, čime jedan deo njegovog sadržaja nepotvrđeno stvara. Treba, takođe, biti vrlo oprezan sa upotreboom bajfera ako se koristi naredba INPUT za primanje podataka.

Mašinski programe i podatke koji su na njih potrebni možete slobodno da smestate i na kraj RAM-a, ali je korisno pre tog promeniti sadržaj sistemskih promenljivih &2A6A. Područje iznad RAMTOP-a je bezbedno i na njega ne utiču ni naredbe za editovanje programa ni operacije sa numeričkim i alfanumeričkim nizom. Potrebno je, ipak, da obratimo pažnju na probleme koji mogu da nastupe ako jednoga dana proširišemo memoriju „galaksije“ ili ako program damo nekom prijatelju koji je to već učinio.

Nove naredbe

ROM računara „galaksija“ je koncipiran tako da omogućava relativno jednostavno dodavanje novih naredbi i funkcija. Da bismo dodali novu naredbu ili funkciju nije jasno, neophodno da razumemo čitavu strukturu bežizak interpretatora (dovoljan je „recept“ koji ćemo, korak po korak, dati u okviru ovoga poglavljija), ali nije loše da učimo nekoliko osnovnih stvari.

Pri analizi bežizak programa računara „galaksija“ u registru DE ćuva adresu bajta koji se trenutno obradjuje. Kada pišemo običan mašinski potprogram koji se poziva sa USR, registar DE će biti automatski sačuvan na steku. Ako, pak, želimo da dodamo novu naredbu, potrebno je da se uklonimo u sam interpretator, tj. da po svaku cenu čuvamo sadržaj DE registra. Ukoliko naša naredba ima i neki adresni deo, potprogram koji je podržava mora da primi i taj deo. Vodeći računa da registar DE, po povratku u bežizak, „pokazuje“ na prvi neobrađen bajta.

Kao i DE, registar IX permanentno pokaže stanje aritmetičkih akumulatora i treba ga, ukoliko nije vršeno računanje koje vraća rezultat bežizku, ostaviti neprimenjen. Naš program mora da se pobrini i da sve moguće sintaksne greške. Ako pogledate listing ROM-a, videćete da je dosta prostora posvećeno proveri sintakske i rutinama za tretiranje grešaka. Tako, jednostavno, mora da bude. Ukoliko interpretator

VAŽNE ADRESE „GALAKSIJINOG“ OPERATIVNOG SISTEMA

Funkcija

a) Aritmetika pokretnog zareza

Izraz adresiran sa DE · (IX)

HL · (IX)

(HL) · (IX)

(IX) · HL

(IX) · (HL)

Izraz u zagradi adresiran sa DE · (IX)

Sabiranje

Oduzimanje

Množenje

Delenje

Poređenje elemenata na vrhu aritm. steka

RND · (IX)

b) Adrese bežik interpretatora

Poziv: CALL &

AB2

ABC

A45

A6D

73B

B32

B1E

AE6

AF7

B10

C8F

LD (&2BA9), A
LD HL,&3000
LD (&2BA), HL

Program koji počinje od &3000 treba da prepozna da li je vreme da se naide na naredbu ili na funkciju. U čemu je razlika? Jednostavno rečeno funkcije se razlikuju od naredbi po tome što se nalaze desno od znaka jednakosti. Tako je, na primer, PRINT naredba a INT funkcija. Po stoje neki reči koje mogu da budu i naredbe i funkcije (BYTE, WORD, DOT), ali to ne treba da vam zvoni: pogled na listing ROM-a će vam uveriti da je svaka od ovih reči u memoriji smeštena po dva puta i da se potpuno razlikuju tretira u zavisnosti od toga da li se trenutno traži naredba ili funkcija. Mi ćemo, najpre, pokušati da uvedemo jednu naredbu.

Tabela naredbi

Kada očekuje da prepozna naredbu „galaksija“ će na link doći pomoću naredbe CALL i to sa adresom &75B. Obzirom na to, na vrhu steka će se u tom slučaju nalaziti bajtovi &07 i &5B. Pošto nam je i tako potrebno da sačuvamo HL registar, prva naredba našeg programa će biti EX (SP). HL ćime smo u HL registar doveli adresu sa koje je dođeno na link. Sada treba da provjerimo da li se u HL registru nalazi broj &75B. To ćemo uraditi tako što ćemo najpre sačuvati registar DE (PUSH DE), zatim u njega uneti ovu konstantu (LD DE,&75B) a onda izvršiti RST &10. Time je požvan sistemski potprogram koji uporeduje registre HL i DE upotrebov svezog jednog bajta. Po povratku sa ovoga potprograma Z i C flagovi su setovani u zavisnosti od rezultata poređenja. Ukoliko je, dakle, Z flag resetovan registri HL i DE nisu bili jednak, pa treba vratiti kontrolu bežik interpretatoru. To radimo tako što, najpre, restauramo DE i HL i izvršimo jednostavno RET.

Ukoliko je Z setovan, treba da pokušamo da prepoznamo naredbu. Najpre ćemo, za svaki slučaj, vratiti DE u normalno stanje (POP DE), a zatim ćemo u HL smestiti adresu tablice umanjenu za jedan-preći na sistemski program čija je ulazna adresa &39A:

LD HL, TABLICA-1
JP &39A.

Potrebno je još da formiramo tablicu koja se sastoji od naziva bežik naredbi koje dodajemo, pri čemu se iz svake od njih

Izraz adresiran sa DE (celobrojno) · HL
Izraz u zagradi adresiran sa DE · HL
Preskakanje blankova adresiranih sa DE
Ako DE adresira zarez, kao RST &8 inače WHAT?

KEY(O) uz podršku BRK, DEL i LIST

Karakter iz A na ekran uz &D, &C i &1D

Štampanje HL na ekranu kao ASCII niz

Štampanje alfanumerika adresiranog sa DE; terminatori je OO (nema novog reda) ili &D (izdaje se novi red)

Licitaj varijablu čije ime pokazuje DE. U HL se smetiš PTR; ako je CT setovan nije varijabla; ako je Z setovan numerička varijabla a u protivnom alfanumerička

Prepoznavanje sledeće naredbe

Stanje aritmetičkog steka

Slobodna memorija (od ARR\$ do bežika) · HL

Shranjevanje bajta na kasetu. Bajt je u A

Bajt sa kasete · C

008;RST &8

A6A

104

5

CF5

20;RST & 320

8FB

937

125

30;RST &30

8F6

183

E68

EDD

nalazi ulazna adresa mašinskog programa koji je ostvaruje. Tekst se, jednostavno, smešta u memoriju slovo po slovu, ali je ulazna adresa priličan problem. Ona se odnosi na neki program koji se nalazi bilo gde u prvih 16 Kb memorije, pa su za njeno smeštanje potrebna dva bajta. Prvi od njih daje značajniji bajt adrese (veoma je važno da učite da ovo odstupa od standardne Zilogove mnemonike kod koje najpre dolazi manje značajniji bajt adresa), pri čemu mu je najznačajniji bit setovan. Ovo setovanje je neophodno da bi računar razlikovao kraj imena od početka ulazne adrese.

Posebno drugog, manje značajnog bajta adrese, sledi im sledeće naredbe koju dodajemo, zatim sledeće i tako dalje. Posle poslednje naredbe stavljam još jednu ulaznu adresu kojoj ne prethodi ni jedna reč. Ta adresu pokazuje bilo koji bajt prvih 16 Kb memorije na kome se nalazi naredba JP &75B. Potrebno je, dakle, da računar nekako prepozna kraj tabele kako bi znao da nema više novih naredbi.

Naredbu koju dodajemo treba da započne jednim POP AF koje će rešiti problem steka koji je poremećen CALL-om na link koji nije završen sa RET. Ovaj POP treba da zaboravimo čim smo ga stavili u program; ne treba ga kompenzovati nikakvim daljim PUSH, niti, niti, smi njim u AF registre doveli neke podatke od začetka.

Slede instrukcije koje izvršavaju naredbu i u njen završetak sa RST &30. Posle ovoga bežik interpretator će pokušati da prepozna sledeću naredbu koja je adresirana DE registrom.

Koje korisne sistemske programe još možemo da upotrebimo? Potrebno je, pre svega, da prenesemo neke argumente nove naredbe. Ukoliko je naredbi potrebno samo jedan argument koji sledi iz nje u zagradi i po prirodi predstavlja ceo broj, stvar je vrlo jednostavna: u okviru tabele ćemo setovati bit 6 prvog bajta ulazne adrese (sada znamo da je sva mora nalaziti u prvih 16 Kb — najslarja dva bita ne služe za adresiranje). Izraz u zagradi će biti izračunat, računar će naci ceo broj tog izraza i poslati ga našem programu kao sadržaj registra HL. Registr DE će u tom slučaju pokazivati prvi bajt iz zagrada.

Ukoliko iz nekih razloga ne želimo da setujemo bit 6 u tabeli (to će se, na primer, dogoditi ako realizujemo naredbu koja ima različite moguće tipove adresnog dela), možemo da započnemo naredbu sa CALL &A6A, što ima istu funkciju. Ukoliko se argument ne nalazi u zagradi, iskoristimo

RST &8 koji će izračunati vrednost izraza i njen deo smestiti u HL registar. Posle toga, naravno, može da sledi zarez koji odvaja prvi argument od drugog. Za detektovanje toga služi CALL &5. Ako DE pokazuje zarez, on će biti preskočen i sledeći izraz vraćen kao sadržaj registra HL, a kako ne pokazuje biće prikazano WHAT? prema pravilima koja smo ranije definisali.

Možda ćemo poželeti da jednostavno preskočimo blankove i dodemo do sledeće značajne bežijk konstrukcije. To radimo sa CALL &104. Ova rutina će nam koristiti jedino ako želimo samo da detektujemo ono što se nalazi iza naše naredbe; u protivnom, do sada pomenutim potprogramima obaveće i ovo zaobilazeњe blanka simbola.

Pošto dodajemo naredbu, nije nam dovoljno samo da prenesemo argumente — naša nova naredba treba da ima i neko dejstvo. To će retko biti neko jednostavno pomeranje memorije — obično treba kontaktirati se ekranom ili lastaturom. I tu mogu mnogo da pomognu sistemski pozivi.

RST &20 stampa na ekranu karakter čiji je kod smešten u registrar A. Pri tome će se, ako je taj bajt &D, preći u novi red, obrisati čitav ekran ako je poslat bajt &C, a obrisati poslednji karakter ako je u A &1D.

Ako se ne zadovoljavamo štampanjem jedne jedine cifre, poželetimo da koristimo CALL &937. Ovaj poziv na ekranu stampa čitav alfumerički adresiran DE registrom i to sve do koda &D (izdaje se novi red na ekranu) ili do koda &80 koji ne izaziva novi red. Debogovanju programa će pomoći i mogućnost štampanja vrednosti registra HL kao ASCII niza slova — to se postiže sa CALL &FB.

Očitavanje tastature

Tastaturu možemo da testiramo na više načina. Filozofiji mašinskog jezika najviše odgovara da se poslužimo memorijskim prostorom &2000 – &2037. Svakom bajtu ovoga prostora dodeljen je po jedan taster prema tablici KEY(N). Najmanji signifikantan bit ovoga bajta je setovan ako je taster pritisnut, a resetovan ako nije. Ako, na primer, želimo da detektujemo da li je taster A (kod 1) pritisnut, izvršćemo program:

LD A,&2000+1

AND 1

JP Z,PRITISNUT.

Mnogo je, međutim, komforntnije koristiti potprogramme za tastaturu. CALL &CFS, na primer, vraća u A registr kód zadnjeg pritisnutog tastera (kao KEY(0)) pri čemu, ukoliko je BRK pritisnut, računar prikazuje READY a ako je pritisnuto DEL privremeno prekida sa radom. Najnepričajnije je to sa STOP/LIST započinje u takvoj prilici najverovatnije neželjeno listanje programa. Sistemski potprogram &CFS podzadržava, pored ostalog, i naredbu REPT za ponavljanje teksta.

Ukoliko želimo samo da računar prekine sa radom ako je pritisnuto BRK, a malo pričeka ako je DEL u akciji, izvršćemo samo CALL &2FF.

Interakcija bežika sa mašinskim programima može da se odvija i preko promenljivih A-Z, X\$, Y\$, A\$(I), i X\$(I). Za to je najbolje upotrebiti CALL &125, što locira varijablu na čije ime pokazuje DE. Po završetku rada

```

10 !
20 ! REC (RECIPROCNA VREDNOST)
30 !
40 ! ILUSTRACIJA FLOATING
50 ! POINT ARITMETIKE
60 !
70 <
80 ORG &3000
90 OPT 3
100 LINK EQU &2BA9
110 FUNK EQU &777
120 PREPOZNAJ
130 NUMIX EQU &A45
140 IZUIX EQU &781
150 DELI EQU &AF7
160 LD A,&C3 ! &C3 = JUMP
170 LD (LINK),A
180 LD HL,PROG
190 LD (LINK+1),HL
200 RET
210 PROG
220 EX (SP),HL
230 PUSH DE
240 LD DE,FUNKC
250 RST &10
260 POP DE
270 JR Z,NAREDBA
280 EX (SP),HL
290 RET
300 NAREDBA
310 LD HL,TABLICA-1
320 JP PREPOZNAJ
330 TABLICA
340 TEXT "REC"
350 BYTE REC>8+&80
360 BYTE REC#&FF
370 BYTE NEGDE>B+&B0
380 BYTE NEGDE#&FF
390 NEGDE
400 RET
410 REC
420 POP AF
430 LD HL,JEDAN
440 CALL NUMIX
450 CALL IZUIX
460 CALL DELI
470 RET
480 JEDAN
490 BYTE 0 ! BROJ 1
500 BYTE 0
510 BYTE &B0
520 BYTE 0
530 >
540 A=USR (&3000)

```

```

&3000: 3E C3 32 A9 2B 21 0C 30
&300B: 22 AA 2B C9 E3 D5 11 77
&3010: 07 D7 D1 2B 02 E3 C9 21
&3018: 1C 30 C3 9A 03 52 45 43
&3020: B0 25 B0 24 C9 F1 21 33
&3028: 30 CD 45 0A CD B1 07 CD
&3030: F7 0A C9 00 00 00 00

```

u HL registru se nalazi apsolutna pozicija promenljive (poput naredbe PTH u bežiku). Carry flag je setovan ako DE ne pokazuje na nešto što bi moglo da bude promenljiva. Ukoliko je C flag resetovan, Z flag govori o vrsti promenljive, setovan Z označava numeričku, a resetovan — alfamumeričku promenljivu.

Sve što smo govorili o dodavanju naredbi ilustruje i program na slici 1. Unesite ga u računar (pomoću hex-loadera) i startujte, pa će set naredbi važećeg računara biti dopunjeno naredbom CLEAR. Ona dodeljuje svim numeričkim promenljivama (A—Z) vrednost nula, što je neobično kori-

sno pri inicijalizaciji programa (kada normalno okucate RUN, promenljive zadržavaju vrednosti koje su dobile u prethodnim programima ili, a od ukључivanja računara nisu korišćene, sadrže broj 0).

Zašto smo se odlučili za naredbu koja ne opšti sa ekranom, ne prima ništa sa tastature niti zahteva neki argument? Jednostavno smo želeli da ilustrujemo sam princip dodavanja naredbi dok za prenošenje argumenta imamo vremena — njega ćemo ilustrovati u sledećim redovima.

Nove funkcije

Funkcije se, kao što rekosmo, uvek nalaze sa desne strane znaka jednakosti, kao deo aritmetičkog izraza. Najveći deo funkcija (ovde nije reč samo o računaru „galaksija“) zahteva jedan argument koji se da bi se poštovala logika sintakse bežika, obavezno stavlja u zagradu (kod nekih novijih kompjutera ovaj argument može ali ne mora da se stavi u zagradu, ali se njenim korišćenjem eliminise mogućnost zabune u slučajevima kada je argument čitav izraz). Kada nađe na funkciju, dakle, računar naprje izračunava vrednost izraza u zagradi, zatim na tu vrednost primenjuje funkciju i dobijenim rezultatom dalje operiše u aritmetičkom izrazu.

Program za dodavanje funkcija je u mnogo čemu sličan programu za dodavanje naredbi, mada postoji i nekoliko razlika. Prva i najvažnija razlika je činjenica da se na ovaj program ne stiže sa lokacije &75B već sa lokacije &777. Na početku programa, dakle, ne treba u DE registar smestiti &75B već &777 pre nego što se izvrši RST &10. Isto tako, na kraju tabele ne treba ići na lokaciju RAM-a na kojoj se nalazi JP &75B, već negde treba formirati JP &777, pa se sa kraja tabele uputiti na to mesto.

Možda je najvažnije da se program koji izvršava funkciju završava sa RET ili JP &ABC, ne sa RST &30. Kada koristimo RET a kada JP &ABC? Kada je rezultat neki deo broj koji se nalazi u HL registru, koristimo JP &ABC da bi taj broj bio smešten na aritmetički stek. Ukoliko smo već računali sa brojevima u pokretnom zarezu tako da se rezultat nalazi na aritmetičkom steku, upotrebimo obično RET. Kako sa sada znamo da radimo samo sa celim brojevima, naše funkcije će se završavati sa JP &ABC.

Program na slici 2 će, nadamo se, učiniti da vam dodavanje funkcija bude znatno jasnije. Odlučili smo se za naredbu DISPLAY koja ima jedan argument koji daje poziciju na ekranu. Naredba DISPLAY tada vrši ASCII kod karaktera koji se nalazi na toj poziciji. Ovo, jasno, uvek možemo da zamenišmo sa BYTE (&2800+I) ali nam, kao što znamo, nije cilj da po svaku cenu dodajemo korisne naredbe koje bi bile složene. Kada razumete princip, moći ćete da se okušate i na nekoj drugoj funkciji.

Za sada, dakle, znamo da dodajemo samo funkcije čiji su argumenti i rezultati celi brojevi. Ovakve funkcije mogu da budu korisne u igrama i sličnim primenama, ali su matematički nastrojeni čitaoći ovih redova svakako poželeti da svoj računar dopune nekom trigonometrijskom ili eksponentijalnom funkcijom koja im mnogo nedostaje. To nije jednostavno kako se u prvi mah čini, ali verujemo da će ovakve funkcije veoma brzo opravdati trud uložen u razumevanje principa njihovog dodavanja. Taj trud će najvećim delom biti utrošen na razumevanje tzv. obrnute poljske notacije.

Obrnuta poljska notacija

Obrnuta poljska notacija, Lukasijevićeva notacija ili, jednostavno, RPN je sistem računara koji na prvi pogled izgleda savšim neprirođeno (odatle valjda i ono „obrnuta“ u nazivu), ali koji je sa aspektom računara i posebnim mašinskih jezika nešto sasvim normalno i, skoro bi se moglo reći, jedino prihvatljivo. Oni koji su posedovali ili poseduju neki Hewlett-Packardov džepni računar (bio on programabilan ili ne) svakako su upoznati sa principima RPN-a pa mogu slobodno da predu na čitanje sledećeg poglavija. Preostalima savetujemo da posećete nekog prijatelja koji ima ovakav računar i u praksi provere koliko će im ovo poglavljivo biti jasno. Probe sa džepnim kalkulatorom je daleko manje naporna od pokušaja da se na mašinskom jeziku (pogotovo ako ne posedujem programer) nauči RPN.

Za RPN notaciju je stek neobično važan pojam. Stek može da zamislite kao orman sa mnogo polica koje se nalaze jedna iznad druge. Na svakoj polici može da se nađe samo jedan predmet (u našem slučaju samo jedan broj). Zato kada god želimo da stavimo broj na donju policu broj koji se u njoj nalazio mora da bude prenesen na prethodnu, broj iz prethodne na policu iznad nje i tako dalje. U teoriji bi stek morao da bude beskonačan, što znači da se dodavanjem broja ništa ne gubi iz njega. Računari teško mogu da posluju sa beskonačnom memorijom koja bi morala da bude upotrebljena da se sačuvaju beskonačni stek, pa naš orman ima gornju policu. Kada stavimo neki broj na stek, broj iz prethodne police će preći u poslednju, dok će broj iz poslednje police biti za vreme izgubljen.

Najniže dve police u steku su posebno privilegovane jer se u njih smještaju argumenti aritmetičkih operacija. Ako, na primer, treba da saberemo brojeve 2 i 3, najpre ćemo na stek staviti broj 2 a zatim broj 3 (ime će, naravno, broj 2 biti pomaknut u sledeću policu), a zatim pozvati program za sabiranje. Da li će on izračunati zbir i staviti ga na stek? Stvar, na žalost, nije tako jednostavna: po sabiranju nam je potreban zbir, a ne i argumenti koji bi na ovaj način ostali zauvek na steku. Zato će program za sabiranje najpre spustiti stek za jedno mesto (u gornju policu upisati nulu, sadržaj gornje police preneti na policu ispod nje itd. sve dok sadržaj druge police ne pomeri u prvu a sadržaj prve ne skloni sa stek), sabrati argumente koji su se ranije nalazili na najniže dve police i smestiti zbir na najnižu policu „zaboravljajući“ njen prethodni sadržaj. Ako su se dakle, na steku nalazili brojevi 2, 3 i 7 (tim redom; 2 je na najnižoj polici), posle sabiranja će na steku biti brojevi 5, 4, 7 i 0.

O množenju teško vredi reći išta novo, ali oduzimanju i deljenju zahtevaju nekoliko reči objašnjenja. Kod ova dve operacije, naime, nije svejedno da li će se računati X-Y (X/Y) ili Y-X (Y/X). Zato je u RPN-u uvedena konvencija koja kaže da je broj koji se stavlja na najnižoj polici steku uvek umanjujući (delilac), a da je broj iznad njega uvek umanjujući (deljenik). Ako se, kao u malopredašnjem primeru, na steku nalaze brojevi 2, 3, 4 i 7 posle izvršavanja oduzimanja će on dobiti oblik 1, 4, 7 i 0.

&2C3A: 3E C3 32 AC 2B 21 46 2C	10 !	
&2C42: 22 AD 2B C9 F5 FE 00 10	20 !	ZAMENA READY
&2C4A: 2B 7C FE 00 20 FA F1 C9	30 !	
	40 !	ILUSTRACIJA LINKA ZA
	50 !	VIDEO
	60 !	
	70 !	
	80 <	
	90 ORG &2C3A	
	100 OPT 3	
	110 LINK EQU &2BAC	
	120 LD A,&C3 ! &C3 = JP	
	130 LD (LINK),A	
	140 LD HL,PROG	
	150 LD (LINK+1),HL	
	160 RET	
	170 PROG	
	180 PUSH AF	
	190 CP 64	
	200 JR Z,MIPRO	
	210 LD A,(RADNI)	
	220 CP 0	
	230 JR Z,NOPRINT	
	240 DEC A	
	250 LD (RADNI),A	
	260 BACK	
	270 POP AF	
	280 POP HL	
	290 POP HL	
	300 RET	
	310 NOPRINT	
	320 POP AF	
	330 RET	
	340 MIPRO	
	350 LD A,6	
	360 LD (RADNI),A	
	370 PUSH DE	
	380 LD HL,PORUKA	
	390 LD DE,(&2A6B)	
	400 LD BC,11	
	410 LDIR	
	420 LD (&2A6B),DE	
	430 POP DE	
	440 JR BACK	
	450 PORUKA	
	460 BYTE 64	
	470 BYTE 39	
	480 TEXT "GALAKSIJA"	
	490 RADNI WORD 0000	
	500 >	
	510 A=USR(&2C3A)	

Broj u kondenzovanom formatu

Do sada smo smatrali da se na svakoj polici steka u jednom trenutku može naći samo jedan broj, što nam je u mnogome olakšalo čitavo rezonovanje. Svaki vlastinski „galaksija“, međutim, zna da svaki broj u pokretnom zarezu zauzima četiri bajta (32 bita) memorije nalazeći se u takozvanom „kondenzovanom formatu“. Kondenzovani format, međutim, nije pogodan za računanje, jer je eksponent broja pomešan sa njegovom mantisom. Zato je pre svake aritmetičke operacije neophodno broj iz kondenzovanog prevesti u normalizovani format. Takav broj će zauzimati 40 bitova od kojih će osam biti neiskorišćeno. Ako po

&2C3A: 3E C3 32 AC 2B 21 46 2C	
&2C42: 22 AD 2B C9 F5 FE 40 2B	
&2C4A: 11 3A 80 2C FE 00 28 0B	
&2C52: 3D 32 80 2C F1 E1 E1 C9	
&2C5A: F1 C9 3E 06 32 80 2C D5	
&2C62: 21 75 2C ED 5B 6B 2A 01	
&2C6A: 0B 0D ED 80 ED 53 6B 2A	
&2C72: D1 1B E1 40 27 47 41 4C	
&2C7A: 41 4B 53 49 4A 41 00 00	

završetku rada treba „zapamtit“ neki normalizovani broj, pretvoricom ga ponovo u kondenzovani format i smestiti u memoriju. Čemu ova komplikacija kada su svi brojevi mogli da se čuvaju u normalizovanom formatu? Jednostavno zbog uštede skućenog memorijskog prostora.

Za programera bi bilo previše komplikovano da stalno misli na to koliko bajtova zauzima koji broj na steku, koliko je ovaj popunjen, šta pokazuje register IX i slično. Zato su obezbeđeni sistemski programi koji omogućavaju da stek zamislimo kao orman sa policama opisan u prethodnom poglavju. Ukoliko se služite njima, činiće vam se

da se na svakoj polici nalazi po jedan broj, nečete misliti na to koliko on ima bajtova, u kom je formatu i tome slično. Zato ćemo sledeće redove posvetiti sistemskim programima za operacije sa stekom.

Najčešće će procesorski register HL pokazivati na neku memoriju lokaciju počevši od koje je smesteno neka konstanta koja zauzima četiri bajta. CALL &A45 će ovu konstantu prevesti u normalizovani format i smestiti je na aritmetički stek uz njegovu obavezno pomeranje nagore. Inverznu funkciju (smeštanje broja sa dne steka u četiri bajta memorije na koju pokazuje HL) obavlja CALL &A6D.

Kako se, međutim, konstante uopšte stavljuju u memoriju? Ako se radi o celobrojnoj konstanti, možemo da je smestimo u HL register i izvršimo CALL &ABC. Ovaj poziv će sadržaj registra HL pretvoriti u broj u normalizovanom obliku i smestiti ga na aritmetički stek uz uobičajeno pomeranje. Inverznu funkciju (smeštanje celog dela broja sa dne aritmetičkog steka u HL uz spuštanje steka) obavlja CALL &A6D.

Kako, međutim, da smestimo konstantu koja nije cela tako da je dočinje preneseemo na stek naredbom CALL &A45? Mi namerno nismo trošili prostor na objašnjenje smeštanja brojeva u kondenzovanim formata (možete li da dedukujete način na koji „galaksija“ smešta brojeve? — nagradicemo najbolje obrazložene odgovore) jer ovo prevođenje može za nas da obavi operativni sistem primenom jednog lepog trika. Potrebno nam je, na primer, da ustanovimo kako se u memoriju smešta broj 1 (on je ceo, ali na sasvim isti način možemo da postupimo i ako nije). Tada ćemo najpre izvršiti A=1, a zatim: FOR I=0 TO 3: PRINT BYTE (PTR A+I): NEXT I. Tako ćemo, bajt po bajt, videti da se broj 1 smešta u memoriju kao (posle prevođenja u uvek pogodnije heksadekadne brojeve): &0, &0, &0 i &0.

Krajnje još vreme da damo i adresne sistemskim rutinama koja obavljaju osnovne operacije. CALL &B2 sabira, CALL &B1E oduzima, CALL &A6E množi, CALL &A7P deli, a CALL &B10 poredi brojeve (rezultat poređenja biva smešten u Z i C flagove u uobičajenim načinima). Da ne ostanemo dužni onima koji žeđe da pišu programe za igre, sistemski program koji na stek stavlja slučajni broj između 0 i 1 može da se pozove sa CALL &C8F. Ukoliko nam je, najzad, potrebno da izračunamo vrednost čitavog izraza i smestimo je na stek, postavćemo DE register tako da pokazuju na početak izraza i izvršiti CALL &A2B. Ukoliko je taj izraz u zagradi (npr. ako je argument neke funkcije), upotrebimo CALL &781.

Ukoliko, najzad, želimo samo da utičemo na znak broja, setovaćemo ili resetovljemo bit 7 u celliji na koju pokazuje IX-1 u zavisnosti od toga da li je broj negativan ili pozitivan.

Kao ilustraciju svega ovoga i još jedno podsećanje na mogućnosti dodavanja funkcija dajemo program sa slike 3. On dodaje računaru funkciju REC koju prati argument u zagradi. Funkcija izračunava, kada što joj ime i govor, recipročnu vrednost ovog argumenta, što i nije nešto naročito korisno ali zato odlično ilustruje rad sa aritmetičkim stekom. Namerno se

10 ! PERMANENTNI SAT
30 ! ILUSTRACIJA LINKA U
50 ! INTERAPTU
60 !
70 <
80 ORG &2C3A
90 OPT 3
100 DFILE EQU &2B00
110 INTER EQU &FD
120 Y EQU &2A80
130 HALT
140 LD IY, ENTRY
150 RET
160 ENTRY
170 LD A, (&2B00)
180 CP 0
190 JR NZ, BACK
200 LD A, (Y+2)
210 CP " "
220 JR NZ, BACK
230 LD A, (Y+5)
240 CP " "
250 BACK
260 JP NZ, INTER
270 LD DE, DFILE+20
280 LD HL, Y
290 DISP
300 LD A, (HL)
310 AND A
320 JP Z, INTER
330 LD (DE), A
340 INC HL
350 INC DE
360 JR DISP
370 >
380 A=USR (&2C3A)

&2C3A: 76 FD 21 40 2C C9 3A B0
&2C42: 2B FE 00 20 0C 3A 82 2A
&2C44: FE 3A 20 05 3A 85 2A FE
&2C52: 2E C2 FD 00 11 14 2B 21
&2C5A: B0 2A 7E A7 CA FD 00 12
&2C62: 23 13 18 F6

nismo poslužili sistemskim programom &ABC koji je mogao da prenese jedinicu iz HL na stek — smeštanje četvorobitne konstante u memoriju i njen prenošenje na stek pomoću CALL &A45 je mnogo instruktivnije.

Link za video

Pogled na mapu sistemskih promenljivih pokazuje da „galaksija“, osim linka za naredbe, ima i link za video na adresama &2BAC, &2BD i &2BAE. Pre nego što će na ekranu ispisati bilo koje slovo (podrazumeva se da to slovo treba da bude ispisano sistemskim pozivom RST &20 a ne direktno smešteno na ekran pomoću LD ili BYT) računar izvršava CALL &2BAC. Ukoliko po pozivu naide na RET, računar će se vrati u ROM i ispisati karakter. Ukoliko, međutim, na ovim adresama nade neki JP, naš program će preuzeti kontrolu i karakter će biti ispisani tek posle naše obrade, kada računar nade RET na kraju našeg programa u RAM-u.

Link za video je, u stvari, obezbeđen da bi uopšte bilo moguće podržati rad sa stampaćem. Ukoliko je ovaj priklučen, trebalo bi sastaviti program koji, kada je neki flag setovan, prenosi svako slovo koje treba

da bude ispisano na ekranu na papir preštevom porta i štampača koji je sa njim povezan. Videćemo, međutim, da se sa ovim linkom mogu postići i neke druge stvari koje će interesovati i one koji ne poseduju štampač. Početno, naravno, od najjednostavnije i ne naročito korisne primene koja će nam pomoći da shvatimo drugi primer programa.

Da li vam se nekada učinilo da vaša „galaksija“ radi prebrzo? Naravno da nije (to se ne dešava čak ni u vlasnicima daleko bržih kompjutera), ali ćemo vam prikazati program koji će usporiti ispisivanje poruka. Ideja je u tome da nije ni malo teško sastaviti bežik program koji će na ekranu usporeni ispisivati neki tekst da bi se, na primer, realizovala neka EPP poruka. Ipak, retki su računari kod kojih se može intervensisati tako da se uspori ispisivanje poruka bežik interpretatora. Pokažite program sa slike 4 nekom vlasniku ZX81 ili Spectrumu — računaru koji ne omogućavaju nikakav izlazak iz ROM-a!

Kako radi ovaj program? Ništa lakše — pre ispisivanja svakog karaktera „galaksija“ izvesno vreme izvršava „mrtvu petlju“ koju možete da skraćujete ili proizvadate menjanjem broja koji se smeti u HL register na početku njenog radnog dela. Po završetku „mrtve petlje“ jedno RET vraća kontrolu nad računaronom bežik interpretatoru u ROM-u. Primitete da je bilo neophodno sačuvati register AF.

Program sa slike 5 predstavlja daleko složeniju primenu linka za video i znatno veće posezanje u rad operativnog sistema. Umetni READY, računar će ispisivati GALAKSIJA sve dok je ovaj program u memoriji. Kako smo to postigli? Pre svakog READY „galaksija“ ispisuje znak koji se sastoji od dva karaktera čiji su kodovi 64 i 39 (označeni u tabeli na strani 18 kao spec). Mašinski program najpre ispisuje da li se u registru A nalazi kod 64 koji označava da upravo treba da bude štampana leva polovina znaka. Ukoliko nije stigao kod 64, izvršava se obično RET i dozvoljava ispisivanje karaktera. Ukoliko, međutim, program prepozna karakter čiji je kod 64, biva proglašeno „vanredno stanje“: u jednu memoriju lokaciju se upisuju podatci da sledećih 6 bajtova ne treba štampati a zatim se, na sasvim standardan način, na ekranu ispisuje tekst GALAKSIJA.

Najlošenije je bilo „zabraniti“ računaru da štampa karaktere. Da bi ovo bilo moguće, proučili smo rutinu za štampanje karaktera u ROM-u i utvrdili da „galaksija“, posle RST &20, poziva dve nivoa potprograma od kojih je drugi linija. Zato naša rutina skida se prva četiri bajta koja predstavljaju dve adrese za povratak sa potprograma (trebalo bi da svi znamo da adresa za povratak sa potprograma ima 2 bajta, a da isto toliko sa steka „sklanja“ jedno POP HL). Računar, dakle, na završetku našeg programa nailazi na RTN, ali sa ono ne vraća na mesto na kome se nalazio CALL koji je pozvao link već u rutinu koja je pozvala RST &20; karakter, tako, nije ni štampan što smo i zeleli.

Link za video može da se koristi i za mnogo što-šta drugo (säčekajte samo malo pa ćete naći na programi koji se automatski startuje) pa i za jednostavnu intervensiju u video igrama. Ima, na primer, igara kod kojih se igračevo „broj života“ ispisuje samo na ekranu, pa bi u principu bilo moguće beskonačno proizvadati igru zabranjujući brisanje karaktera koji ga predstavljaju. Čini nam se, ipak, da je ovakva primena

sljednih linkova daleko zahvalnija kod drugih računara, kod kojih je struktura ekranra komplikovanija (link za video kod Spectruma bi mogao divno da se koristi da postoji). Autori igara za „galaksiju“ će, svakako, smatrati da je lakše prosti „poukovati“ nešto u ekran nego konstituti sistemski i program koji to radi!

Link u interaptu

Na početku poglavljia koje je posvećeno računaru „galaksija“ rekli smo da nikako ne treba koristiti registar IY, jer je on korišten kao link u interaptu. Pošto smo do sada, nadamo se, stekli sva potrebna znanja o pojmu linka, posvetimo se podrobnej ulozi koja je dodjeljena registru IY. Moramo, međutim, da upozorimo da su linkovi koje smo pomijenili opasni prema redosledu izlaganja, što znaci da je najopasnije baviti se ovim poslednjim pre startovanja svakog programa treba „sore faji“ (izvorni program) snimiti na kasetu. Kada pogrešno završen interpret „ugusi“ računar, RESET najverovatnije neće pomagati — bice potrebo da isključite i ponovo ga uključite.

Pošto je pomogao video stepenu da generiše sliku i povećao brojčanik sekundi u časovniku (ako je ovaj uključen, naravno), mikroprocesor je pri kraju interapta — treba još samo učeti sistemski registre sa steka i izvršiti RETI. Pre nego što lo uradi, međutim, mikroprocesor nalazi na naredbi JP (IY). U registru IY je u toku inicijalizacije računara smešten broj &00FD, pa mikroprocesor izvršava naredbe koje se tamo nalaze. To su, naravno, četiri POP naredbe i RETI.

Mi, međutim, možemo da promenimo sadržaj registra IY tako da on pokazuje neku lokaciju u RAM-u. Tamo treba da se nalazi program koji će se izvršavati svake pedesetinke sekunde i koji će se završavati sa JP &DE. Pri izradi takvog programa, međutim, treba da budemo neobično pažljivi: on, pre svega, mora da se izvršava vrlo kratko jer svaki mašinski ciklus njegovog izvršavanja skraćuje iskorišćenje mikroprocesora za izvršavanje redovnog programa. Osim toga, za vreme interapta ne sme nikako da se desi da mikroprocesor dobije novi interpret, ako ništa drugo zbog toga što bi se stek tako stalno punio. Da bi to bilo sprećeno na samom početku interapta, mikroprocesor automatski izvršava DI. Ukoliko naš program koji se izvršava u interaptu, samostalno ili nehotičnim pozivom neke rutine iz ROM-a, izvrši El, vrlo je verovatno da će nastupiti veliki problemi. Ako se ovo ne desi, ali se interpret ne omoguci u razumnom vremenu, slika se više neće videti i jedini spas će biti isključenje računara.

Osim kratkoče, savetovali bismo vam da u okviru programa koji se izvršava u interaptu ne pozivate potprograme iz ROM-a, jer većina njih u nekim slučajevima može da poremeti normalan tok stvari (odlaskom na rutinu za grešku, na primer). Ako, dakle, želite da testirate neke tastere u interaptu, moraćete da testirate celiće &2000-&2037, a ne da koristite CALL &CFS.

Na slici 6 dajemo primer korišćenja linka u interaptu. Radi se o rutini koja se nekada nalazi u ROM-u, ali je morala da bude

```

10 <
20 ORG &3F00
30 OPT 3
40 LINK EQU &2BA9
50 Z6B3 EQU &75B
60 X343 EQU &39A
70 DFILE EQU &2800
80 NUVAR EQU &2A00
90 BYTE EQU &668
100 WORDH EQU &E63
110 WORDZ EQU 5
120 LD A,&C3 ! &C3 = JUMP
130 LD (LINK),A
140 LD HL,PROG
150 LD (LINK+1),HL
160 RET
170 PUSH DE
180 EX (SP),HL
190 PUSH DE
200 LD DE,Z6B3
210 RST &10
220 POP DE
230 JR Z,NAREDBA
240 EX (SP),HL
250 RET
260 NAREDBA
270 LD HL,TABLICA-1
280 JP X343
290 TABLICA
300 TEXT "RSAVE"
310 BYTE RSUB>&600
320 BYTE RSUB>&600
330 BYTE BACK>&600
340 BYTE BACK>&600
350 BACK
360 RET
370 RSUB
380 POP AF
390 LD HL,&66
400 PUSH HL
410 RST B
420 PUSH HL
430 CALL ZAREZ
440 INC HL
450 PUSH HL
460 CALL ZAREZ
470 LD B,H
480 LD C,L
490 POP HL
500 POP DE
510 PUSH BC
520 LD B,&60
530 DI
540 LEADER
550 XOR A
560 CALL BYTEA
570 DJNZ LEADER
580 LD A,&A5
590 CALL BYTEA
600 POP BC
610 PUSH HL
620 PUSH BC
630 LD H,D
640 LD L,E
650 ADD HL,BC
660 LD B,&A5
670 CALL WORDHIL
680 LD A,B
690 POP BC
700 PUSH HL
710 PUSH HL
720 ADD HL,BC
730 LD B,A
740 CALL WORDHIL
750 POP HL
760 JP &E57
770 >
780 A=USR (&3F00)

```

```

&3F00: 3E C3 32 A9 2B 21 0C 3F
&3F01: 22 AA 2B C9 05 D5 11 5B
&3F10: D7 D1 2B 02 E3 C9 21
&3F19: 10 3F C3 9A 03 B2 53 41
&3F20: 54 45 2B 27 B6 26 C9 F1
&3F29: 21 40 80 E7 F5 00 E5 CD 05
&3F30: 00 23 E5 CD 00 00 44 4D
&3F39: E1 D1 C5 06 68 F3 AF CD
&3F40: 68 00 10 FA 3E 45 CD 68
&3F49: BE C1 E5 C5 62 68 09 06
&3F50: AS CD 63 0E 78 C1 E1 E5
&3F59: 07 47 CD 63 0E E1 C3 57
&3F60: 0E

```

žrtvovana zbog nekih rešenja rada sa alfumericima: neprekidnom prikazivanju vremena. Kada otkucamo i izvršimo ovu rutinu, neće se, bar u prvi mah, ništa dogoditi. Da bismo vidieli njenje dejstvo, otkucacemo Y\$=12:10,15:10“DOT“ (zapazite da su minute i sekundi odvojeni tačkom, a ne uobičajenim dve tačke). Od toga momenta u desnom gornjem ugлу ekranra videćemo tekuće vreme. Bez obzira da li listamo program ili brišemo ekran, vreme će biti prikazivano sve dok, na primer, ne otkucamo Y\$=12:10,15:10“.

Kako je ovaj efekat posignut? Prilično jednostavno: svake pedesetinke sekunde računar prenosi sadržaj promenljive Y\$ na ekran ukoliko su ispunjena dva uslova. Prvi je da se u Y\$ uopšte nalazi vreme (treći simbol treba da budu dve tačke), a drugi je da peti simbol stringa Y\$ tačka. U ovom programu se javlja i naredba HALT koju obično prenебрегавamo u školama programiranja na mašincu, pa smo odlučili da joj ovde posvetimo par reči.

Naredba HALT izaziva prekid rada mikroprocesora, što obično izaziva pomisao da je ona destruktivna. Mikroprocesor, međutim, neće zadužiti stajati: čim naide prvi interpret, njegova aktivnost će se nastaviti. To znaci da se HALT može prevesti kao „sačekaj prvi interpret, izvrši ga, pa nastavi sa radom“. Kao takva, HALT je neobično korisna naredba: ukoliko želimo da radimo nešto sa IY registrom ili, čak, da prenosimo veće blokove ekranra, vrlo je korisno upotrebiti naredbu HALT kao neku vrstu sigurnosnog ventila. Ukoliko to ne učinimo, moguće je da program počne da greši s vremenom na vreme u naoko besmislenim trenucima, a iskustvo nam govori da se takve greške najčešće otkrivaju.

Šta bi se desilo da, u momentu kada je mikroprocesor naišao na HALT interpret bio onemogućen (DI)? U tom slučaju bi HALT zaista blokirao računar, ali bi pritisak na RESET povratio sve u normalno stanje bez štete po sadržaju RAM-a. Dolazak na „farmu“ bi, ujedno, i omogućio interpret, pa bi se pojavila i slika koja do tega momenta nije bila.

Obzirom da govorimo o interaptu, neophodno je da sa par reči obradimo NMI — nemaskirani interpret. On nastupa kada korisnik pritisne dirku RESET koja se nalazi sa zadnje strane „galaksije“. Nemaskirani interpret se razlikuje od maskiranog po tomo što računar, kada ga dobije, u istom trenutku prestaje sa izvršavanjem naredbe koja je trajala i prelazi na program koji se nalazi u ROM-u počevši od &66. Tamo najprije biva reinicijalizovan stek pointer, a zatim ispisano READY. Posle nemaskiranog interpret, međutim, „galaksija“ neće reinicijalizovati sistemski promenljive, linkove i sadržaj IY registra, što za praktičnu posledicu ima činjenicu da računar može ali i ne mora da se vrati u radno stanje — ako je, na primer, interpret pogrešno organizovan, slike neće biti ni posle njega.

Kako se „pakuje“ bejzik

Došlo je vreme da se posvetimo periferijskoj jedinici koju svr vlasnici „galaksije“ redovno koriste — kasetofonu. Pre nego što objasnimo rad sa njom i navedemo adrese dva sistema polprograma koja će nam omogućiti daleko više od naredbi OLD i SAVE, posvetimo pažnju načinu na koji „galaksija“ pakuje bejzik program u svome RAM-u.

Kod „galaksije“ nije zastupljeno tako-zvanou tokenizovanje naredbi, što ima dobre

loške strane. Pošto naredba nije pretvorena u poseban kod (kod Spectruma se, na primer, PRINT pretvara u &F5), tako je i jednostavno analizirati program bajt po bajt, ali se zato njegovo izvršavanje unekoliko usporava. Program je, dakle, u memoriji smješten kakav kakov je, slovo po slovu, u dva izuzetka.

Linjski broj se kodira kao dva bajta i to u obliku koji proizvodi naredbu WORD. Ukoliko je, na primer, prva naredba bežik programa 1500 GOTO 1550, PRINT WORD (&2C3A) će dati rezultat 1500. Blanko koji računar prikazuje prilikom listanja programa između linjskog broja i prvog slova naredbe nije, u stvari, upisan u RAM-u (tako se stodi memorija), dok se ostali blankovi u okviru linije kodiraju kao i svaki drugi ASCII karakter.

Pošte poslednjeg znaka neke programske linije u memoriji je obavezno smješten bajt &D (13 dekadno), koji predstavlja terminator linije (znak da je ona završena) i pokazuje računaru da slijedi dva bajta broj slijedeće linije. Ukoliko se radi o posljednjoj liniji bežik programa, iz njenog &D se ne stavlja ništa posebno (kod drugih računara kraj bežika označava jedna do tri nule ili neki drugi specijalni kodovi). Kod „galaksije“ bi ovakav terminator bio sasvim nepotreban: sistemski promjenljivač čija je adresa &2C38 u svakom trenutku pokazuje gde se nalazi prvi slobodan bajt iza bežik programa. Ukoliko smo, na primer, u memoriju stavili samo bežik naredbu 10 PRINT A, memoriske čelije počevši od &2C3A imaju redom sadržaje 0,10,80,82,73,78,84,32,65,13, a promjenljivač &2C38 ima vrednost &2C44.

Dobar primer svega ovoga je program „Telefonski imenik“, koji je objavljen u „Galaksiji“ broj 143. Ovaj program modifikuje samoga sebe dodajući nove naredbe i kodirajući ih na odgovarajući način. On, naravno, stalno vodi računa o sistemskoj promjenljivoj &2C38 koju povećava posle dodavanja svakog bajta. Pri njegovoj analizi treba jedino imati u vidu da bežik program ne počinje od &2C3A (tu je smješten mašinski potprogram), već da je sa NEW 60 početak bežika pomeren za 60 bajtova. Računar to registruje promenom sistemskog promjenjiveča &2C38.

Snimanje na kasetu

Dolazimo, najzad, do snimanja programa i podataka na kasetu. Da bi računar mogao korektno da učita program, moraju mu budu saopštene informacije o njegovoj dužini i (kada nije reč o bežik programu) adresi od koja ga treba upisivati. Zato se ispred svakog programa nalazi takozvani „header“ (zaglavljiv) u okviru koga su upisane ove informacije. Pre hedera na traku biva snimljen takozvani „lider“ od 96×8 nula bajtova koji nemaju nikakvu posebnu ulogu — oni jednostavno saopštavaju računaru da sledi program i omogućavaju da se kasetofoni koji imaju automatsku kontrolu nivoa reprodukcije prilagode specijalnim tonovima. Računar pri učitavanju ne provjerava da li je učitano baš 768 „praznih“ bajtova po učitavanju programa ne morate da počnete sa trakom prematom načinu na mesto na kome se nalazi početak „liderske“ tolerancije, jasno, nije velika.

&3BAC: 00 00 00 00 00 00 00 00 00
&3BB4: 00 00 21 FF 29 36 20 2B
&3BBC: BC 5C 20 F9 21 CD 2B 11
&3BC4: E7 28 01 11 00 ED 0B 10
&3BCC: FE 41 55 54 4F 53 54 41
&3BD4: 52 54 20 50 52 4F 47 52
&3BCD: 41 4D

Posle „lidersa“ nalazi sinhronizacioni bajt &A5 koji se takođe ne unosi u memoriju: on, jednostavno, govorи računaru da je došao kraj „lidersa“ i da sledi „header“. Header ima samo četiri bajta: prva dva označavaju adresu od koje program treba da bude smješten u memoriju, a druga dva adresu prvog bajta koji će biti slobodan posle uvođenja programa. Sledi sam program, bajt po bajt. Na kraju snimka nalazi takozvan „čeksum“: jedan bajt koji omogućava računaru da sa velikom sigurnošću (ova sigurnost ni kod jednog kompjutera ne može da bude stoprocenata) utvrdi da li je program ispravno upisan u memoriju ili je u toku učitavanja neki bajt ispušten ili deformiran. Šta je, dakle, „čeksum“? U toku upisa programa u memoriju, „galaksija“ sabira pročitane bajtove i to po modulu 256 (100+20=120, 120+111=231, 231+30=50=261=256) i tame slično), a čeksum ovaj zbir popunjava do 00 (u vrijeme primjera čeksum bi bio 256-5=251).

Ukoliko bi se po učitavanju pokazalo da zbir, dopunjeno čeksumom, nema vrednost nula, bilo bi sasvim sigurno da je program pogrešno upisan, pa bi računar ispisao WHAT? a zatim i READY. U protivnom, vrlo je verovatno (moguće je, naravno, da je neki bajt povećan za jedan, a neki drugi smanjen za 1, što se u konačnom zbiru ne primjeće) da je program ispravno prepisan u memoriju, pa će „galaksija“ ispisati samo READY.

Čak i kada je program pogrešno upisan sa kasete, recunar „galaksija“, zahvaljujući originalnom metodu kodiranja bitova, olakšava njegovu „vraćanje na normalu“ uz minimalne intervencije.

Na kasetu se upisuju četvrtasti impulsi koji predstavljaju binarne jedinice i nule. Svi impulsi se sastoje od pozitivne i negativne poluperioda od kojih svaka traje po 0.22 ms. Ukoliko se na kasetu upisuje binarna nula sledi pauza duga 2.46 ms, a ukoliko se upisuje jedinica — kratka pauza od 1.1 ms. Iza ovakvog „dvostrokovog“ impulsa koji odgovara jednom bitu sledi drugi, treći i ostali bitovi bajta, koji su kodirani na isti način. Po završetku čitavog bajta sledi dodatna pauza od 5.36 ms koja predstavlja značajno osiguranje: ako se desi da je računaru, zbog oštećenja trake ili nekog drugog problema ispušti neki bajt, pauza od 5.36 ms će se „podsetiti“ da je bajt završen i on će biti spreman da primi sledeći. Tako će biti ispušteno jedno slovo, ali će sva dalja bita ispravna; ukoliko nije bilo daljih oštećenja. Računar će, jasno, po učitavanju ispisati WHAT? pa će obično RUN verovatno naći sintaksnu grešku koju ćemo lako ispraviti. Postoji verovatnoća (mada vrlo mala) da program posle greške u učitavanju bude sintaksno ispravan, ali da nekorektne radi. U tom je slučaju mnogo lakše pokušati novi upis sa kasete nego tražiti grešku u imenu premetljive ili konstante.

Da biste sami upisali bajt na traku treba, najpre, da mašinskim programom isključite interpet (DI), a zatim da pozovete sistemski potprogram čija je ulazna adresa &E68. Na traku će biti poslat bajt iz registra A.

Učitavanje bajta sa kasete se postiže na prilično sličan način — CALL &EDD će navesti računara da sačeka sledeći bajt („galaksija“ neće imati ništa protiv ako do naiaska tog bajta protekne daleko više od 5,36 ms) i da ga prenese u registar C.

Program sa slike 7 je jedan od primera rada sa kasetofonom: on omogućava relo-cirano snimanje programa. Rutina sa slike 7 se smješta na sam kraj memorije kako bi bilo moguće upisivati programe koji se nalaze na njenom početku. Kada izvršite program, vaš računar će posedovati novu naredbu RSAVE koja ima tri argumenta: prva dva odgovaraju argumentima naredbe SAVE koja omogućava snimanje segmenta memorije, a treći da je adresu počevši od koje će se na kasetu upisani program dočinje upisivati. Naredba RSAVE &3000, &3500, &5000 će snimiti sadržaj memorijskih čelija &3000+&3500 na traku, ali će se ovaj program ubuduće učitavati počevši od &3500 (= &3000+&500) kada upotrebimo očnicu OLD.

Ovaj program predstavlja dobar uvod za sledeće poglavije. Želeli smo da navedemo dve primene koje bi trebalo da budu ne samo poučne nego i korisne. Prva od njih će biti automatsko startovanje programa!

Automatsko startovanje programa

Sinclairovi računari omogućavaju da program koji se učitava sa kasete počne da se izvršava bez ikakvih daljih intervencija korisnika. Obrazloženje potreba za ovim automatskim startovanjem je obično tvrdnja da je za početnike teško da pamti naredbu RUN što i nije neki razlog (ako već mora da pamti naredbu LOAD... „RUN valjda ne predstavlja toliki problem). Da su oni koji propovedaju automatsko startovanje iskreni, reči bi da ono predstavlja dobar način zaštite od neovlašćenog presnimavanja programa. „Galaksija“ nikada nije bila pristalica nekih zaštita programa, najpre zbog želje da softver što više kruži među njenim čitatcima. Dakle, razlog zbog koga navodimo informacije vezane za auto-start programa nije naša želja da stimulisemo čitatce da na mnoštu zaštićenih programa (na kraju poglavija ćemo reći i kako se ovakve zaštite „probijaju“), već verovanje da je autostart lepa „cak“ koja ilustruje mnogo stvari o kojima smo do sada govorili.

Ideja je prilično jednostavna: kada računar „galaksija“, primenom naredbe OLD, učita program, on će počekati da ispiše READY. Ukoliko je u link za video upisan poziv nekog programa, kontrola će biti predata tom programu koji će se, dakle, startovati odmah po učitavanju. Ostalo je još pitanje kako da postignemo da se sadržaj bajtova &2BAC-&BAE promene u toku učitavanja programa. Samo na jedan način — naš program treba da počne da se učitava od adresi &2BAC, a njegova ulazna adresa treba da bude na samom početku. Posle ovakve intervencije nećemo, jasno, više moći da se oslanjamo na RST &20, sistemsko skrolovanje slike i slične pomoći operativnog sistema. Zauvrat, čak i pri prelasku na RESET neće prekidati naš program — i posle nemaskiranog interpet-a „galaksija“ će počekati da ispiše READY i ponovo započeti izvršavanje našeg programa.

Sve ovo ilustruje program sa slike 8 da kao hex-dump. Unesite ga, u memoriju a zatim (pod uslovom da se u memoriji već nalazi startovan program sa slike 7), otkačite RSAVE &3BAC, &3BDD. — &1000.

isključite i uključite računar pa otkucajte OLD i startujte kasetu na koju ste maločas snimili program. Po njegovom učitavanju, na ekrantu ćete videti jednu kratku poruku koju će RESET ponavljati.

Zamislite da vam je neko dao program sličan ovome i to na kaseti. Kako da ga presnimite ako ne želite da se upustite u metod koji uvek daje dobre rezultate pre-snimanje uz "pomoć dva kasetofona. Ne brinite, nije mnogo teško. Najpre utvrdite njegovu početnu (MMMM) i krajnju (NNNN) adresu. Otkucajte OLD 200 i učitajte program koji će ovim biti relociran za 200 bajtova. Zatim, koristeći program sa slike 7, otkucajte RSAVE MMMM+200, NNNN+200, -200 i mirno snimite program na novu kasetu.

Može li da se napravi bolja zaštita? Može ako sastavite program koji će biti toliko dugacka da za njegovog relociranja upisivanje nema mesta (npr. program koji bi se prostirao od &2BAC do &3FFF) što možete da učinite jedino ako posedujete memorisko proširenje. Takav program će biti prilično dobro zaštićen od onih koji nemaju toliku memoriju, ali oni sa hardverom poput vašeg i dalje neće imati problema, a snaci će se i jedan deo „lukavijih“ sa RAM-om od 6 Kb: sastavice program za prepisivanje koji će počinjati od &2A00 i smestiti ga u područje numeričkih varijabli u koju u toku učitavanja programa nije moglo bitno.

Ukoliko, najzad, sastavljate program uz pomoć nekog drugog računara sa mnogo memorije ili disk jedinicama, nema prepreke da napišete rutinu dugu nekih 54 kilobajta koja će, samom svojom dužinom, biti prilično dobro zaštićena (možete li da sastavite program koji bi presimao i ovakvo nešto? Ne zaboravite da iz svakog bajta na kaseti može da postoji i pauza duži od 5.36 ms), ali takva rutina neće biti baš mnogo komercijalna jer će njen učitavanje trajati.

Funkcija TIME

U akcionim igrama na kraju se igrači obično rangiraju po osvojenim poenima koji su, pored ostalih parametara, funkcija utrošenog vremena. Računar „galaksije“, kao što znamo, ima časovnik koji je vrlo pogodan za prikazivanje vremena, ali ne toliko pogodan za merenje broja sekundi koje su protekli između dva događaja. Program sa slike 9 dodaje vašem računaru funkciju TIME koja pretvara vreme iz \$Y u sekunde. Ovaj program ilustruje skoro sve o čemu smo govorili u poslednjem poglavljiju: dodavanje naredbi, prijavljivanje grešaka (ako sadržaj promenljive \$Y nije vreme, biće prijavljeno HOW?) rad sa brojevima u pokretnom zarezu uključujući i smeštanje konstanti...

Svi „galaksijini“ bagovi

Redovni čitaoci naših napisu u „Galaksiji“ su se u poslednje vreme upoznali sa pojmom „bag“, ali neće biti loše da, u cilju osnateg kompjuterskog obrazovanja, još jednom kažemo nekoliko reči o njemu.

Bag je greška u operativnim sistemima računara koja se ispoljava u nekim specijal-

nim situacijama kao reakcija na neku ne-predviđenu akciju korisnika. Ne treba misliti da je bag trenutak kada računar ispiše SYNTAX ERROR ili WHAT? — tada je sve pod potpunom kontrolom operativnog sistema. Posledice baga su obično daleko teže: gubitak linije, programa ili sadržaja čitave memorije, pogrešan ispis na ekrantu ili gubitak njegovog sadržaja, remećenje bitnih sistemskih promenljivih...

Svi računari imaju bagove koji predstavljaju manju ili veću smetnju njihovim korisnicima. Izuzetno, kod nekih računara sa „zatvorenim“ operativnim sistemom (bez PEEK i POK) bagovi mogu da budu korsnici, jer otvaraju pristup području memorije koju su inače tabu za korisnika. Proučavanjem i sistematsičnjem bagova računara HP41C članova američkog PPC kluba su razvili sintetičko programiranje koje im je omogućilo kompletan kontrol ovog čuve-nog džepnog računara.

Što se kućnih kompjutera tiče, njihovi bagovi prave samo probleme: BBC mikroračunari i dalje imaju bagova (najzadniji od njih praktično onemogućava korišćenje reza u naredbi PRINT), premda njegov operativni sistem predstavlja dva-nestuveriju istog programa. Spectrum (pogotovo stariji modeli) ima pregršt bagova od kojih je, na svu sreću, malo „pogubnih“, naredbe za editovanje kod TRS 80 izazivaju bezbrojne probleme kod neiskusnih korisnika koji teže da primene sve što su našli u

uputstvu za upotrebu... Nije ništa bolje nego drugih kompjutera.

Iako smo užilo dosta truda da pronađemo i elimišemo sve „galaksijine“ bagove, nismo se nikada zavarivali nadom da ih neće biti. Teši naš jedino činjenica da su sive „galaksijine“ bagove pronašli oni koji su radili na njegovom razvoju — sudeći po vašim pismima, nijedan čitalac „Galaksije“ nije našao na neki bag svoga računara. Mi ćemo, naravno, navesti pronađene bagove da bi se čitaoci ovih redova čuvali situacija koje mogu da izazovu probleme.

BUG 1: Otkucajte punu programsku liniju (obeležje 10 i slova A sve dok računare bafer ne bude pun, na primer). Zatim otkucajte EDIT 10 i, kada se linija pojavi na ekrantu, pritisnite DEL nekoliko puta a zatim i ENTER. Pritisnite LIST — ko je rekao da se u memoriji „galaksije“ nalazi neki program?

BUG 2: Otkucajte liniju koja će glasiti 10 A= zatim tridesetak otvorenih zagrada, pa neki broj, pa isto toliko zatvorenih zagrada, a zatim startujte program. Posledice su iste kao kod baga 1. Napominjemo da će „normalan“ broj od dvadesetak otvorenih zagrada izazvati „normalnu“ reakciju HOW?

BUG 3: Ovaj bag je toliko dobro sakriven da još nismo uspeli da ga lociramo. U nekim retkim situacijama (uzroci, da ponovimo, još nisu poznati) naredbe za rad sa numeričkim nizom A(I) će izazvati poruku SORRY. PRINT MEM daje neki negativan broj. Problem se rešava kucanjem naredbe ARR(\$-1), posle koje sve postaje normalno.

Svi ovi bagovi bi, jasno, mogli da se isprave odgovarajućim promenama ROM-a, ali ovi promeni verovatno neće nikada biti izvršene. Odgovor na pitanje zašto? je sasvim jednostavan: ako promenimo ROM adresu svih sistemskih potprograma (pa i onih koji su pomenuti u ovom poglavljiju) će se promeniti, što znači da programi neće biti kompatibilni sa obe verzije ROM-a. Tako ćemo najverovatnije ostati kod 28 verzije ROM-a koju smo proglašili konačnom (PRINT BYTE (55) će vas uveriti da vaš računar ima 28 verziju ROM-a).

I uputstvo za upotrebu „galaksije“ nije imuno na bagove: u mapi sistemskih promenljivih na strani 26 postojje dve sistemске promenljive na adresi &2B11, ali će svakako moje i olažljivo čitaocu biti jasno da „register za REPT“ zauzima ćeliju &2B44. Osim toga, „horizontalna pozicija teksta“ zaista ima inicijalnu vrednost 11, ali dekadno, a ne heksadekadno. Za autora ovoga teksta je najboljnije da kaže da program sa dana stranu 7 ima grešku i to ne štampansku nego logičku: on dobro radi za vrednost iz prve, ali čim bi se putovanje odužilo preko jednog časa... Obzirom da nam se čini da svaki program početnik (baš tako) može da napiše ovakav program tako da radi ispravno, ovde ne dajemo novu verziju.

Time smo došli do kraja ovoga prilično opširnog poglavljija koje je posvećeno računaru „galaksiji“. Autor ovoga teksta duguje veliku zahvalnost Vojni Antoniću, konstruktoru „galaksije“, za pomoć pri izboru tema koje će biti obradene, sarađujući na pripremi prve upotrebe raznih tehniki, i, posebno, za sve podatke o ROM-u koje niko drugi nije ni mogao da zna. Zahvaljujući ponajviše ovim podacima, naša „galaksija“ je, čini nam se, postala jedan od vrlo relikvija računara koji, kako rekosmo i u naslovu poglavlja, „nema tajni“!

Dejan Ristanović

ČETRNEAEST GODINA ELEKTRONIKE

Zagrebačka Tvornica računskih strojeva jedan je od jugoslovenskih pionira u primjeni mikroelektroničkih logičkih sklopova u biroelektroničkim i AOP uređajima. Već 1969. god. razvijen je na bazi TTL int. sklopova prvi elektronički stolni kalkulator; razvojni modeli koristili su feritnu memoriju, ali u proizvodnoj seriji ugrađivani su već, tada još savim novi, MOS shift-registri. Razvoj novih tipova uvijek je pratit najnoviju kretanja na svjetskom tržištu komponenti i gotovih proizvoda. Među prvim firmama u Evropi i TRS je počeo ispitivati mogućnosti mikroprocesora preve generacije; kao rezultat takvog rada pojavio se i fakturimi stroj TRS-701 (kanjić 711) sa INTEL-ovim 4-bitnim mikroprocesorom. Na području 8-bitnih procesora u ranoj je fazi sagledana važnost „single-chip“ tipova za masovnu primjenu u biro-opremi, pa je jedan niz serijskih proizvoda, gotovo cijeli kalkulatorski program, zasnovan na FAIRCHILD-ovom „F-8“, kojeg su za TRS-ove potrebe u velikim količinama izrađivala u posebnim varijantama 3 svjetske proizvođača (sigurno jedinstveni pothvat u Jugoslaviji).

Po polovicom sedamdesetih godina nabavljeni su razvojni sistemi za 6800-familiju i za Z80, a rezultat rada s tim, još i danas aktuelnim mikroprocesorima, su računski sistemi TRS-702 i 703, kao i periferni uredaji: matrični štampači TRS-825/830 i video-terminali TRS-826/836. Danas mogućnosti 8-bitnih procesora još su izdaleku ništa iscrpljene i mnogi novi TRS-ovi proizvodi koriste hardware i sistemski software za njih razvijen tokom vremena, ali već je započeo i rad na 16-bitnim sistemima, jer su u okviru srednjoročnih razvojnih planova definirani i proizvodi, u kojima će procesna „snaga“ 16-bitnih mikroprocesora doći do punog izražaja. Paralelno s razvojem elektroničkih sklopova odvija se razvoj programske podrške; grupa za podršku u okviru teh. sektora brojil dvadesetak ljudi i trajno širi područja primjene naše AOP-opreme.

U vezi s novim malim sistemima TRS-703 i 901 donesena je (većim dijelom već i realizirana) odluka da se izrade vlastiti sistemski programi. Operacioni sistemi za „703“ i „901“ su već gotovi, a sada se radi na PL-1 kompilatoru. Na ovo vrijeme smo u odlučili i sami zaoblikovali devizu, koje bi mogli trošiti za potrebu u svijetu kompjuternih OP-sistema CP/M ili MS-DOS i pripadajućih kompilatora, nego i u želji da korisnicima pružimo sistemski software optimiran za naše male sisteme i za profesionalne aplikacije.

U skorijoj budućnosti treba pristupiti realizaciji DBM (Data Base Management) software-a, imajući u vidu i neke specifične aplikacije i način viziju daljnog razvoja računarske tehnike u smjeru distribuirane obrade i lokalnim mrežama.

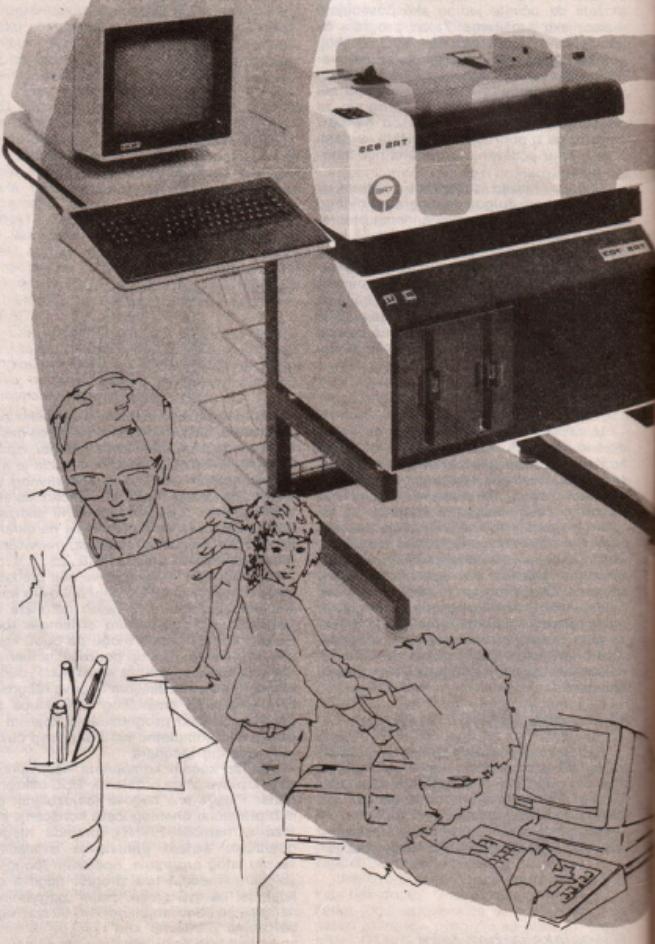
Potpuna modularnost na nivou hardware-a i sistemske i programske podrške omogućava TRS-u da svoju liniju proizvoda prilagodi potrebama narucilaca, a u izgradnji je i takva servisna organizacija koja će na želju profesionalan način moći održavati isporučene uredaje u cijeloj zemlji. Pored dva glavna pravca, "malih računarskih sistema i perifernih uredaja, TRS će i nadalje proizvoditi uredsku opremu (kalkulator), a s mnogo ambicija ulazi je i u područje elektromedicinske opreme (za sad proizvodom „TRS-GLUKOTEST“ — uredajem za mjerjenje koncentracije lukoze u krvi). S posebnim ponašom u TRS-u se gleda na činjenicu da su svi naši proizvodi rezultat vlastitog razvoja; smršljena i dosljedno provođena kadrovska politika omogućava TRS-u da trajno okuplja mlade stručnjake i da ostane jedan od rasadnika stručnih kadrova u zagrebačkoj regiji. Ironi-

ja je našeg vremena da se donedavno vlastiti razvoj manje isplatio od kupnje licenci ili puškog sastavljača uvezenih pod sistema. U TRS-u se nadaju da će sadašnji napori za stabilizaciju naše privrede stimirati one proizvođače koji s vlastitim resursima dolaze dc novih proizvoda i tehnoloških rješenja.

Na tom putu — uzimajući u obzir i činjenicu da bi autorija bila društveno neprivatljiva i štetna — TRS će i dalje suradivati s naučnim ustanovama i proizvodnim organizacijama u

SRH i šire, podupirući napore da se između proizvođača elektroničkih uredaja, a posebno AOP opreme, postignu dogovori o usklađenju proizvodnih programa i da se što više ovih tehnički i tehnološki vrlo složenih uredaja realizira kao zajednički proizvod. U tome i kroz

TRS 7



kooperaciju s odabranim stranim partnerima dugoročno vidimo jednu šansu za plasman na inozemno tržište i zadovoljenje potreba za devizama. Učešće uvozne repromaterijala u TRS-ovim proizvodima je oko 10% (po vrijednosti), a budući da se opravdano može govoriti o supstituciji uvoza, trebalo bi o toj činjenici voditi računa u gremijima u kojima se raspodjeljuju strana sredstva plaćanja.

Najatraktivniji proizvod Tvornice računskih strojeva svakako je:

MALI POSLOVNI SISTEM

Preteća je cijene obitelji suvremениh matičnih kompjutera opće namjene. Hardware je modularen i zasnovan na 8-bitnom mikroprocesoru Z80 s nizanim periferijskim upravljačkim jedinicama. U osnovnoj konfiguraciji su on pred 64 KB RAM ima i dvije diskovne jedinice (Floppy-diskete) s kapacitetom od 2,2 MByte. Bitna svjedočta tog kompjutera određuju njegova sistemска i aplikativna programska podrška: za „703“ razvijeni je i vlastiti operacioni sistem (svakako jedinstveni slučaj u Jugoslaviji), a donjeva se i prevodlaci (compiler) za PL-1 visi programski jezik. U asortimanu konsolinskih programa najznačajniji su programi za vodenje materijalnog i finansijskog knjigovodstva, osobnih dohodaka i salda-konta, kao i programski paket za organiziranje i praćenje proizvodnje i skladišnog poslovanja. Tu su zatim programi za obradu teksta i za umuljiranje nekih IBM-ovih synchronih terminala. U osnovnoj konfiguraciji mali sistem „703“ obuhvaća centralnu jedinicu s oba diska, jedan stampač TRS-85 i video-terminal TRS-88. Stišprema je model „901“ sa četiri upisna mesta „multi-task/multi-user“ načinom rada.

TRS 835

prijski matrični štampač TRS-835 je izlazna
linica za ispis podataka. Predviđen je za rad
s terminalom — terminal isključivo u prije-
vom modu, odnosno kao RO štampač (RE-
CEIVE ONLY).

Sve su funkcije štampača mikroprocesorski upravljane i nadzirane radi postizanja i održavanja zadanih tehničkih karakteristika. Kod projektiranja i proizvodnje serijskog matročnog štampača TRS-835 korisena su suvremena teorijska saznanja i najmodernoji tehnološki dostignuća, što štampaču osigurava vrhunsku kvalitetu, veliku trajnost, sigurnost i pouzdanošću u radu.

TEHNIČKI PODACI

Brzina ispisu	180 zn/sek
Broj znakova u retku	132 znaka
Gustina ispisu	2,54 mm/zn
Ispis u oba smjera s optimiziranjem puta	
Matrica ispisu	9 x 9
Veličina znaka	2,54 x 1,97 mm
Inkrementalni vertikalni pomak formulara	0,529 mm (u oba smjera)
Brzina pomaka formulara	101,6 mm/sek
Gustina ispisu redaka	6 redaka na 25,4 mm ili 6 redaka/inču
Fond znakova	96 zn (JUS ili ASCII)
Vrpca velike trajnosti u kaseti	
Veličina prihvativne memorije	274 zn (do 4 K opcionalno)
Interface prema izvoru podataka:	
— CENTRONICS paralelni	
— V 24 (RS 232 c) serijski	
— DEC LA 160 paralelni	

REFERENCE

Štampač TRS-815, TRS-825, TRS-8250 J i TRS-835, proizvedeni u Tvornici računskih strojeva provjereni su i ispitani s uspehom u radu sa sistemima slijedećih proizvođača: **ISKRA DELTA, IBM, OLIVETTI**.

TRS 838

ASINHRONI VIDEO TERMINAL

TRS-838 je suvremeno oblikovani univerzalni video-terminal koji se zahvaljujući primjeni modernih mikroprocesorskih komponenti i konceptu, po komu se sve funkcije uspostavljaju u jednom terminalu, a ne u kruto ožičenom logikom, manje primjenjivim u svim aplikacijama u kojima se traži univerzalni asinhroni terminal. Pored ostalih interesantnih svojstava, posebno je treba istaknuti,

- mogućnost prikupljača hard-copy štampača (preko RS 232 C medusklipsa)
 - mogućnost povećanja video RAM-a do 12 K znakova (6 potpunih stranica) opcija
 - mogućnosti prikazivanja tzv. "business graphics" (ne u lokalnom modul)
 - mogućnost emuliranja drugih rasprostranjenih asinhronih terminala (IBM 3101 standardno, DEC VT 100 po posebnoj narudžbi)
 - mogućnost izravnog povezivanja s hostom udaljenim do 1200 m preko RS 422/s medusklipsa

Terminal se sastoji od jedinice s ASCII tastaturom u ekranso/lošiće jedinice s kadaonim clevi dijagonale 30 cm (12 inča) sa zelenim fosforom srednje persistencije. Raspored znakova u tastaturi izmjenjiv je u dogovoru s naručiteljem. TRS-638 s različitim sistemima komunikacija preko asinhronog međusokopa po RS 422 ili RS 232 C za izravnu ili udaljenu (remota) vezu, odnosno preko 20 mA strujne petlje za izravnu vezu.

Ostala svojstva TRS-838:

- radi u znakovnom ili blok modu
 - brzina prijenosa podataka je od 110—9600 Bits/sek
 - korisnik može odabratи lokalni mod rada bez intervencije sistema; to se može koristiti za učenje operatera, kreiranje formata ili editiranje podataka na ekranu, adresiranje cursor-a funkcije brisanja, samostalni pomak na novi red, scrolling-up, prijenos bloka podataka i formatiranje polja
 - atributi formiranih polja mogu biti:
 - a) zaštićeno / nezaštićeno
 - b) vidljivo / nevidljivo
 - c) treptajuće / netreptajuće
 - d) pojačani intenzitet / normalni intenzitet
 - e) modificirano / nemodificirano
 - znakovi su u matrići 7x9, 24 reda s po 80 znakovima; 25. red je za poruke operateru o radu terminala (sistemski red)
 - tastatura je ergonomski optimalno riješena, odvojena od sistema i operater je može prilagoditi svom položaju
 - ekran se može zaokreti oko horizontalne osi i tako postaviti u najugodniji položaj za operatera
 - ESCAPE sekvencu za izvršavanje specifičnih funkcija koje se ne mogu aktivirati preko kontrolnih znakova.

Terminal svojim funkcionalnim karakteristikama emulira IBM 3101 (model 2x) terminal i može ga u potpunosti zamijeniti.

Terminal je razvijen za rad s IBM S/370 (uključujući 303×), 4300, 8100 i S/1 programskom podrškom.

TVORNICA RAČUNSKIH STROJEVA - ZAGREB

41000 ZAGREB — Braće Kavurića 21 — p. p. 02-846 — Brzopis: TRS-Zagreb
Telex: 21434 — Centrala: 447-111 447-001 447-692 — Prodaja: 411-362

Telex: 21434 — Centralna: 447-111, 447-001, 447-592 — Prodaja: 411-302
Sektor podrške: Poslovnička Zagreb, Šeferova 2, 41000 Zagreb, tel. 212-900, 212-911, — Poslovnička Beograd, 27. marta 39, 11000 Beograd, tel. 327-747, — Poslovnička Ljubljana, Linhartova 11a, 61000 Ljubljana, tel. 327-470, — Poslovnička Rijeka, Hotel „Royal“, Opatija, Dražice 3, tel. 721-544.

Komponente za memorijsko proširenje

Konstruktorski hod po mukama

Za konstruktors i hobiste, čini nam se, nikada nisu bila teža vremena — čipove je, čak i na svetskom tržištu, sve manje, a cene vrtoglavije rastu. Neslavni rekord u sklopovima cena drži Intelova čip 8255 koji je sa neka tri funte za dva meseca skočio na 19 funti! On, na žalost, igra ključnu ulogu u našem programatoru EPROM-a, a vrlo brzo ćemo mu se, kada budemo radili paralelni interfejs za stampač, vratići još jednom. Utešno je jedino što su cene dinamičkih memorija stabilne već mesecima, tako da postupno dobri izgledi da se do kraja neće menjati.

Iako u ovim uslovima nije bilo nimalo jednostavnije sklopiti aranžman za organizovanju nabavku delova iz inostranstva — u trenutku kada svojim kupcima mere čipove „na gram“, strani trgovci nisu bili previše raspoloženi za naše devizne propise i „štoseve“ sa paketicima od 1500 dinara — redakcija je ipak pošlo za rukom da postigne odgovarajuće sporazume sa putem smrža u organizatoru Londona za ovaj podlogu. Poviši je učinkovitosti čipova 4416/16nren mogu naći jedino u Engleskoj (ne sumnjamo da će ih u uskoro nabaviti i trgovci iz Austrije i Italije), a drugi je da je naš doprinos Andrejko Zgorelec občasno puni pomoć, po čak i kontroli uspori.

Iz organizacije akcije sa računaron „galaksija“ mogli smo puno toga da naučimo, ali, na žalost, malo što možemo da učemo o tome da se u inostranstvu, nimalo u ulicama na domaćim privrednim tokovima, a pogotovo ne na kretanju na svetskom tržištu čipova. Zato se naši uslovi, verovatno, neće dopasti konstruktora koji gore od nestreljivosti da povećaju snagu svoje „galaksije“! Mi, ovoga puta, ne možemo da garantujemo da same cenu nego — ruk isporuke. Možemo jedino da obećamo komponente — štampana kola i čipove — i to samo za one koji poštaju narudžbinice do 10. avgusta.

Takođe, učitac koji zadržava memoriju, nećemo niti organizovati proizvodnju štampanih kola, odnosno nekavnu čipova. To, praktično, znači da isporuka ne može početi pre kraj septembra, ali i to da će biti okončana u relativno kratkom roku. Naravno, ako nas i ovoga puta ne zatrepa 4400 narudžbenica, kao za računar „galaksiju“. Prilikom isporuke će biti poštovan redosled prijavljivanja — ovoga puta nije bilo preliminarnih narudžbenica, pa ne može doći ni do zabune. Ljubitelji računara, govorite slobodno iz januarske akcije da je jedan dan ranije u stanju narudžbenice zatražiti i do tri meseća čekanja višešta.

Citaoči koji su dobro upoznali mehanizme akcije sa računaron „galaksija“ — znamo da su oni, mimo naše volje i uticaja, mnogima dobro „sameli“ žive — primetite dve novine: znatno nepovoljnije ali realnije rokove isporuke i nešto drugačiju organizaciju. Isporuka štampanih kola vršiće se, najverovatnije, iz redakcije „galaksije“. Time ćemo postići da se učitaci ujedno u dinamiku, a citaoči će lako doći do potrebnih obaveštaj. Isporuka čipova biće, kao i do sada, vršena iz inostranstva.

Cene koje navodimo za štampana kola i komplete čipova nisu, razum se, konačne. To, međutim, ne bi trebalo previše da brine ljubitelje računara. Tačne cene za štampana kola obјavljemo petnaestak dana pre početka akcije. Oni će, takođe, biti ujedno i u dinamiku, a citaoči će lako doći do potrebnih obaveštaj. Tačna cena komplata čipova biće navedena u predračunu koji će svaki interesent dobiti iz inostranstva. Isporuka čipova biće vršena u skladu sa našim carinskim propisima.

Nijedan časopis ne može da preuzeme na sebe sve konstruktorske probleme svojih čitalaca, a većina, najčešće, nije raspoložena da preuzeme i redakciju. Pravim, reč je o nešto predviđajući najveću vrednost koja ćemo može ponuditi citaočima. U našim uslovima, međutim, veće nesto drugačiji zakoni — provereni projekti ne znače ništa ako se ne obezbedi uslov za njihovo realizaciju. Računar „galaksija“ bi, bez obzira na sve svoje kvalitete, najverovatnije ostao samo „mrtove slova na papiru“ da nije organizovan servis za nabavku delova. U njemu nije uvek sve funkcionalisalo kako treba.

— rokovi su se razvilačili kao gladna godina, a cene skakale kada smo se tome najmanje nadali — ali sada ipak ima u pogonu najmanje 1500 „galaksija“ i svakoga dana ih je sve više. Redakcija se još uvek nije umorila balansirajući između vatre svojih čitalaca i vatre svojih saradnika u svojoj akciji. Nadamo se da niste ni vi. Strpljenje je, uostalom, majka mudrosti. Ako nam ono ne pomogne, zastava ne znamo šta će.

Koliko šta košta

1. Štampano kolo za memorijsko proširenje (dvoslučna štampa na vitroplastu) sa konstruktorskom plošćicom od 1300 dinara

2. Štampano kolo za programator EPROM-a (jednoslučna štampa na vitroplastu) od 1000 dinara

3. Komplet čipova za memorijsko proširenje 48 K (čipovi, podnožja, konektor, poštarna)

48 K oko 40 funti
32 K oko 30 funti
16 K oko 20 funti

4. Komplet delova za programator EPROM-a (integrirana kola, podnožje od 40 nožica i ZIF podnožje od 28 nožica, konektor)

oko 35 funti

Važi do 10. avgusta

Narudžbenica (štampana kola)

Molim vas da mi pošaljete pouzećem:

1) štampano kolo za memorijsko proširenje (vitroplast, dvoslučna štampa) sa konstruktorskom plošćicom po orijentacionoj ceni od 1300 dinara

2) štampano kolo za programator EPROM-a (vitroplast, jednoslučna štampa) po orijentacionoj ceni od 1000 dinara

Mesto

Ulica i

broj

Ako mi konačna cena ne bude dogovarala, zadržavam pravo da otkažem svoju narudžbinu deset dana pre pošetke isporuke. Štampana kola će plasti poštaru prilikom preuzimanja vaše pošiljke. Narudžbenicu (original ili fotokopiju) dostaviti na adresu: Redakcija „Galaksije“, 11000 Beograd, Bulevar vojvode Mišića 17, ZA STAMPANA KOLA

Narudžbenica (komplet integriranih kola)

Molim vas da mi pošaljete predračun sa uslovima isporuke za sledeći komplet delova:

- memorija za „galaksiju“ 16 K
- memorija za „galaksiju“ 32 K
- memorija za „galaksiju“ 48 K
- programator EPROM-a za „galaksiju“
- programator EPROM-a za ZX Spectrum

Mesto

Ulica i

broj

Narudžbenicu (original ili fotokopiju) dostaviti na adresu: INTRAPRESS, 16 Garway Road, LONDON W2 4NH, England

MOĆ
RAČUNARA
I LJUDSKI UM

Džozef
Vajzenbaum:

MOĆ
RAČUNARA
I LJUDSKI
UM

Knjiga koja na izuzetan način osvjetljava neke aspekte jednog od najlošijih fenomena sa vremena tehnologije — kompjuteralizaciju. Pisana popularno i duhovito, ali bez narušavanja tehničke i naučne discipline, poseduje i tu nevakidašnju vrednost što pruža odgovore na neke često nepoznate probleme, a istraživalac postavlja nezaobilaznu pitanja.

Format 14 x 20. Str. 307. Broširano. Latinica. Cena..... 350,-

RAD — IZDAVAČKA RADNA ORGANIZACIJA
11000 Beograd, Motel, Pijade 12, Tel.: 404-765 i
422-517

NARUDŽBENICA — RAČ/105

Ovim ne pozivaju narudžbenik knjigu:
MOĆ RAČUNARA I LJUDSKI UM,
po ceni od 350,-

Knjigu će uplatiti pouzećem, prilikom preuzimanja od pošte.

(Poručilac)

(Broj pošte, mesto

i adresa stanja)

(Datum)

(Potpis poručilca)

RAD

poziv
na
preplatu

hemij tehnološk priročnik (u 6 knjiga)



HEMIJSKO-TEHNOLOŠKI PRIROČNIK pokriva specifičnu području hemije i hemijske tehnologije. Zahtijevajući konceptciju ovog višestomog izdanja, sve pojedinačne knjige čine celinu, čime poprimaju trajnu vrednost i postaju stalni saradnik hemičara u njihovom radu.

Prva knjiga

HEMIJSKI I FIZIČKI PODACI I VELIČINE

Autori: M. Krmanović, M. Obradović, S. Đorđević, G. Đurić, P. Todorović, D. Šeća, P. Bojović, K. Subotić, N. Radovićević, C. Milićević, V. Dražić, O. Tatić-Janić, B. Đorđević, V. Vučić, S. Šumanović, Z. Đorđević. **Redaktori:** Z. Đorđević, V. Vučić, S. Šumanović, Z. Đorđević. **Knjiga** sadrži uži izbor podataka iz područja hemije i tehnologije, u kojima se računaju koniste u laboratorijima istraživačkih i industrijskih organizacija. Dati su numeričke vrednosti veličina sa odnosom na fiziko-hemiske osobine elemenata, jedinjenja i tehničkih materijala.

Iz sadržaja:

- Matematika
- Definicije pojema i jedinini fizike hemije
- Međunarodni sistem mernih jedinica i preporuke
- Fizičke osobine elemenata i jedinjenja
- Fizičkohemijske tablice
- Osobine tehničkih materijala
- Opšti, trivijalni i komercijalni nazivi materijala i hemikalija
- Podaci za proračun doza i zaštite od zračenja radioaktivnih izvora

Druga knjiga

ANALITIKA

Autori: J. Čiro, D. Stojanović, V. Vajgand, D. Pešić, J. Dobrošavljević, V. Antonijević, S. Kočić, M. Kobiljarov, G. Hrđaković, Z. Knežević, A. Gogalić. **Redaktor:** J. Čiro. Ova knjiga je obrađena i sistematizovana tako da daje ne samo teorijske znanosti, već i praktične primjene, odnosno analitičku hemiju nego i postupak u praktičnoj primjeni. Obuhvaća se osnovne klasične i suvremene metode organske i neorganske analize. Metodologija je sistematizovan veliki broj analitičkih podataka, koji su neophodni za svakodnevno izvođenje pojedinačnih analiza.

Iz sadržaja:

- Ravnopravne u ravnomjeru elektrota
- Kvalitativna neorganska analiza
- Gravimetrija i volumetrija
- Elektroanalitičke metode
- Spektrohemijska analiza
- Radio-hemski analiza
- Organska analiza
- Spektrografska analiza organskih jedinjenja

- Knjige će izlaziti iz štampe sukladno, do kraja 1984.
- Mogu se pretplatiti u najviše 10 mesečnih rata.
Po izlasku iz štampe cena će biti znatno veća
- Pretplatnicima koji jedinom uplate cenu iznos pretplate odobravamo POPUST 20%

1949
1984

rad

Treća knjiga

FIZIČKOHEMIJSKE METODE

Autori: S. Đorđević, I. Donatovićević, D. Pešić, A. Gogalić, M. Marinović, M. Pešić, A. Muš, S. Ribićević, A. Antić-Jovanović, M. Jerević, N. Jakovljević-Hala, M. Roguljić, I. Draganić, Z. Draganić, K. Zubakov, G. Hrđaković, N. Ovetićanin, I. Ovetićanin, S. Đorđević, S. Đorđević i D. Pešić. **Knjiga** treća uži izbor fizičkohemijske metode merenja koje se najčešće koriste u laboratorijima istraživačkih i industrijskih organizacija. Dati su numeričke vrednosti veličina sa odnosom na fiziko-hemiske osobine elemenata, jedinjenja i tehničkih materijala.

Iz sadržaja:

- Optičke fizičkohemijske metode
- Elektrohemijske metode
- Optičke metode
- Razne metode

Cetvrtu knjigu

ŠTETNE I OPASNE MATERIJE

Autori: D. Stojanović, N. Stojanović, D. Kosoanović. **Redaktori:** D. Stojanović. Knjiga sadrži tablarni pregled preko 3.000 raznovrsnih hemijskih i neorganskih materijala i toksičnim karakteristikama, kao i pronosom opće opasnosti i mogućih učinkova. Pored ovih karakteristika navedene su i određene neophodne fizičke osobine za istu jedinjenja, kao što su temperatura topljenja i kijenja, gustina, indeks refrakcije, napon pare i gustina plina.

Iz sadržaja:

- Direktno dejstvo štetnih materija
- Indirektno dejstvo štetnih materija
- Transport opasnih materija
- Skladištenje i pakovanje opasnih materija
- Rad sa opasnim materijama i štetnim
- Mere zaštite na radu
- Opbrane otpadne materije
- Uputstvo za prvu pomoć
- Tablarni pregled karakteristika opasnih materija

RAD - IZDAVAČKA RADNA ORGANIZACIJA
11000 Beograd, Moše Pijade 12.
tel: 404-765 - 422-517

NARUDŽBENICA — RAČUNARI/103

Ovim se neopozivo pretpostavlja da
HEMIJSKO-TEHNOLOŠKI PRIROČNIK (u 6 knjiga),
po prepl. cani od 12.600,-
Uplatu duži:
ODJEDNOM — Po prijemu „Postava“ uplatnice, sa
POPUSTOM od 20% (10.080,- din.,
U — RATA — Upisati broj (najviše deset), po
prijemu uplatnice na RADA
(Nepotrebno precrati).

U službu sporazudjevanje je Peti opštinski sud u
Beogradu.

(Poručilac)

(Broj pošte, mesto)

(adresa stanja)

(M.P.)

(Datum)

(Potpis poručilca)

Peta knjiga

HEMIJSKO INŽENJERSTVO

Autori: G. Popović, B. Đorđević, V. Vasiljević, S. Šarčanović, F. Đenčić, G. Vujanović-Naković, S. Kelenar-Durđević, Z. Gavrilović, A. Tešić, A. Mihajlović, S. Joksimović-Tepjačin, S. Pejanović, A. Tušović, D. Šimović, D. Hadžimajlović, A. Tošić, A. Groz, R. Radovićević, M. Mitrović, S. Pribićević, S. Ovsišević, S. Šimović, M. Gamber, B. Bošković, A. Desimir, Lj. Milišević, M. Jevtić, S. Šimović. **Redaktori:** S. Končar-Durđević, D. Šimović, J. Mićić.

U ovoj knjizi dati su teorijske osnove, kao i podaci koji će koristiti poslovne u radu na tehnologiji i drugim projektanjskim poslovima u gospodarstvu i naštrenjanim i delatnostim, dati su primjeri koji boje ilustruju smenu materija.

Iz sadržaja:

- Primjena matematika u hemijskom inženjerstvu
- Termodinamika

Principi i projektovanje tehnoloških operacija

A. Operacija prenosa količina kretanja

Fluidi u mirovanju. Fluidi u kretanju. Strujanje u cjevima. Melanja. Disperzivni sistemi. Disperzija faza.

B. Operacija mehaničkog tretiranja čvrstih materija

Trenutak, transport čvrstog materijala. Smjerenje materijala. Prosejavanje.

C. Operacija prenosa topline

Konvekcija topline. Zrajenje topline. Toplotni blansir. Cevni raznjenjivo topline. Ukuvanje. Kondenzacija i kondenzatori. Rashladni procesi. Goriva i savorevanje.

D. Operacija prenosa mase

Osnove principa prenosa mase. Konstantni utređaj za prenos mase sistema gas-tehn. Apsorpcija. Destilacija. Ekstrakcija tečno-tečno. Luženje. Palomjerija i kondicijoniranje. Suhlenje i sušnice. Apsorpcija i razmaka zravama. Katalizacija. Ostale direktne operacije.

E. Osnove teorije i projektovanja hemijskih reakcija

1. Kataliza i katalizatori

Dinamika sistema. Instrumentacija i automatska regulacija u hemijskom inženjerstvu

2. Osnove projektovanja proizvodnih sistema hemijske (procesne) industrije

3. Hemijsko-inženjerska ekonomika

Sesta knjiga

KOROZIJA I ZAŠTITA MATERIJALA

Autori: S. Mladenović, M. Petrović, G. Riković. **Redaktori:** O. Tatić-Janić. Knjiga obrađuje teorijske osnove korozije i tehničko-tehnoloske principne zaštite materijala.

Iz sadržaja:

- Elektrohemijska i hemijska korozija metala i legura
- Korozija materijala neorganskog i organskog porekla
- Koroziona agresivnost raznih sredina
- Zaštita materijala od korozije

Knjige će izlaziti u štampanoj latincu, formata 15,5x22,5 cm, povezane u celo platno sa zlatotiskom, na oko 5.000 strana.

Cena u pretplati 12.600,-

Majstori na računaru

softverska veza

Struje i kanali — najmoćnije
spectrumovo tajno oružje

Dobro je o periferijskom uređaju razmisljati tako što će se on podeliti u dva dela: jedan odgovara softveru a drugi hardveru koji prima ili generiše podatke. Kod ZX bežičke softverske komponente naziva se „struja“, a hardverska komponenta „kanal“. Ključna razlika je u tome što je struja bežičan protok podataka u program ili iz njega, dok kanal odgovara određenom uređaju, kao što je, na primer, ZX štampač. Struju treba shvatiti kao zbir podataka na njihovom putu ka nekom komadu hardvera ili od njega.

Struja...

Struja je određena nekim brojem od 0 do 15; njene osnovne operacije su čitanje i pisanje podataka. Naredba:

INPUT #s; „ulazna lista“: čitaće podatke iz struje „s“ i prebacivati ih u promenljivu koju smo nazvali „ulazna lista“. Na primer:

INPUT #0; A; B; A\$ čitaće podatke iz struje 0 i skladištiti ih u promenljive A, B i A\$. Na isti način, naredba:

PRINT #s; „print lista“: slaće podatke strujom „s“ iz promenljivih u „print listi“. Na primer:

PRINT #0; TOTAL; A\$ Obratite pažnju na to da se i INPUT i PRINT mogu koristiti na potpuno isti način kao i obične naredbe INPUT i PRINT. Svaki parametar koji se može koristiti kao i deo PRINT i INPUT može se uključiti i kao deo struje. Na primer, u važnosti su obe donje linije:

PRINT #2; „RAČUNAR“; TAB (10); „GA-LAKSIJA“

INPUT #0; „Kako se zoveš?“; NS Naredba PRINT salje strujom 2 slovni niz „RAČUNAR“, zatim nekoliko praznina i, najzad, slovni niz „GALAKSIJA“. Obratite pažnju na to da podatak koji je poslat strujom odgovara podatku koji bi bio poslat ekranu. Naredba INPUT je nešto složenija, utoliko što ne zahteva samo podatak iz struje 0, već i šešire podatak u obliku slovnog niza „Kako se zoveš?“. Svaki broj struje je, u stvari, vezan sa dve struje podataka: jednu ulaznu i jednu izlaznu struju. Podatak upisan u struju bilo naredbom PRINT ili naredbom INPUT, salje se izlaznom delu struje, a svaki podatak koji se čita iz struje dobija se iz njenog ulaznog dela.

Sa „nerazvijenim“ (tj. bez mikrodravja) Spectrumom mogu se koristiti još jedino naredbe LIST i INKEY\$. Potpuni oblik naredbe LIST je:



LIST #s.n
gde je „s“ broj struje kojom se program mora izlistati, a „n“ je broj linije od koje počinje listanje. Na primer:

LIST 1
izlistati će program pomoću struje 1.

Druga naredba orijentisana ka struji, INKEY\$, može se koristiti za prihvatanje pojedinačnog bajta iz bilo koje struje koja je vezana sa uređajem koji podržava ulaz.

Naredba:

INKEY\$ s

vratice kod jednog karaktera iz struje „s“.

... i kanali

Lako je shvatiti ideju o struji podataka, ali se postavlja pitanje kako se određena struja povezuje sa određenim periferijskim uređajem. Odgovor je: da bi se neki podatak poslao ili primio određenom strujom, ona mora biti najpre otvorena. Otvaranje struje ima dva cilja: ono povezuje potrebnu struju sa određenim periferijskim uređajem i, u isto vreme, signalizira koji će uređaj biti upotrijebljen. Često, uz signaliziranje da će jedan uređaj neposredno biti korišćen, otvaranje struje obuhvata i uputstvo instalisanom uređaju kako da dode u stanje u kome će se koristiti; ovo upućivanje, međutim, vrlo mnogo zavisi od samog uređaja.

Otvaranje struje ostvaruje se naredbom:

OPEN #s, c

gde je „s“ broj struje koja se otvara, a „c“ slovni niz koji određuje kanal sa kojim je ona povezana. Prema toj naredbi, odredite

svakog podatka poslatog strujom „s“, biće kanal „c“, koji će, isto tako, biti i izvor svakog podatka pročitanog iz te struje. Pre nego što se može dati praktičan primer korišćenja naredbe OPEN, treba znati kojim kanalima SPECTRUM raspolaze.

Spectrum bez mikrobrojeva raspoznaće samo tri različita kanala:

K — kanal tastature
S — kanal ekранa
P — kanal šrampača

Tako

OPEN #5, „K“
otvara struju 5 i vezuje je sa tastaturom. Na osnovu naredbe:

INPUT #5; A; B
dobiće se podaci sa tastature na isti način kao i pri normalnoj naredbi INPUT. Međutim, naredba:

PRINT #5; „GALAKSIJA“
šalje sada podatke izlaznoj strani struje 5, koja je povezana sa dnom ekranu. Na taj način, slovni niz „GALAKSIJA“ upisuje se u donji deo ekranu koji je normalno rezervisan za INPUT poruke. Ako to pokušate, verovatno da će moći videti niz, jer se donji deo ekranu briše čim se program zauzavi ili čim računar nađe na naredbu INPUT. Ako hoćete da vidite efekat slanja podataka u „ulaznu oblast“ ekranu, pokušajte sa:

10 OPEN #5; „K“
20 PRINT #5; RND
30 GOTO 20

Videteću na ekranu nasumične brojeve, koji beže odozdo navise. Program će se završiti sa porukom o greći OUT OF SCREEN.

Prvi put kada poželite da na svoj Spectrum vežete neki periferijski uređaj, sudaricete se sa problemom kako mu slati ili kako od njega primati podatke. Obično se misli da je lakše konstruisati hardversku vezu sa Spectrumom nego „softversku vezu“ sa ZX bejzikom. Međutim, Spectrum ima veoma elastičnu i sofisticiranu metodu rukovanja periferijskim uređajima, zasnovanu na korišćenju „struja“ i „kanala“, pa je razmerno lako povezati periferijske uređaje sa ZX bejzikom tako da ih računar tretira kao svoj sopstveni hardver. Koristeci veoma jednostavan softver u mašinskom jeziku moguće je slati podatke svakom uređaju naredbama PRINT i LIST i primati podatke od svakog uređaja naredbama INPUT i INKEY. To praktično, znači da se pirmjenjeni programi mogu pisati u standardnom bejziku prećestog pozivanja mašinskog koda. Čudno je da se, uprkos tome što standardni Spectrum koristi tako elastičnu tehniku, ona jedva i pominje u priručniku!

Mapa kanalskih zapisa

kanal za tastaturu

CHANS — adresa print rutine za donji deo ekran-a
 +2 — adresa input rutine za tastaturu
 +4 — K kod kanala K
 kanal za ekran
 +5 — adresa print rutine za ekran
 +7 — adresa rutine za tretiranje grešaka
 +9 — S kod kanala S
 kanal za redaktorski baf
 +10 — adresa bafera input rutine
 +12 — adresa rutine za tretiranje grešaka
 +14 — R kod kanala R
 kanal za ZX printer
 +15 — adresa rutine za ZX printer
 +17 — adresa rutine za tretiranje grešaka
 +19 — P kod kanala P

Mada u principu svaka struja ima ulaznu i izlaznu stranu, u praksi jedini kanal koji može da primi i ulaz i izlaz jeste kanal tastature. Druga dva kanala, ekran i štampač, samo su izlazni kanali i svaki pokušaj da se sa njih očitavaju podaci završće prijavljivanjem greške J. Treba primetiti da je ovo ograničenje isključivo karakteristika hardvera za koji je vezana struja.

Za jedan kanal je moguće vezati više od jedne struje, ali ako se želi promeniti kanal sa kojim je struja vezana, veza se najpre mora ukloniti sa CLOSE. Dakle:

CLOSE „s“ ukloniće svaku postojeću vezu između struje „s“ i kanala. U ovom smislu, CLOSE je za kanal suprotno od OPEN. Zatvaranje struje može se koristiti i zato da se komponenta hardvera jednog kanala obavesti da struji više nije potrebna.

Formati kanala

Cetiri struje, 0 do 3, automatski se otvaraju prilikom inicijalizacije računara:

struja	kanal
0	K
1	K
2	S
3	P

Tako će, čak i bez OPEN, naredba:

PRINT # 2; „GALAKSIJA“

ispisati tekst na ekranu. Spectrum koristi ove struje za upućivanje programskih podataka u odgovarajući uređaj. Na primer, LPRINT šalje podatke ka struji 3. Ovi odnosi struja i kanala mogu se promeniti korišćen-

njem naredbe OPEN, ali same struje se ne mogu zatvoriti. Pokušaj da se zatvori jedna od ovih struja završava njenim ponovnim otvaranjem ka svom prvobitnom kanalu.

Ako struje i kanali treba da se koriste za povezivanje dodatnih uređaja, tada je očigledno važno znati umutrašnje formate koji su korišćeni za skladištenje informacija o kanalima, kao i koji je kanal vezan sa kojom strujom.

Informacija koja definije svaki kanal uskladištena je u zoni kanalnih informacija polazeći od CHANS i završavajući sa PROG-2 (gde su i CHANS i PROG sistemske promenljive). Svaki kanal ima poseban „kanalni zapis“ sledećeg formata:

adresa veličina
 n — 2 bajta adresa izlaznog programa
 n+2 — 2 bajta adresa ulaznog programa
 n+4 — 1 bajt kod kanala

gde su ulazni i izlazni programi sabrunitne mašinskog koda. Izlazni program mora da prihvati Spectrumove kodove karaktera koji mu se „dotvaraju“ preko A registra. Ulazni program mora da vrati podatke u obliku Spectrumovih kodova karaktera i da signalizira da su podaci dostupni selovanjem tzv. flega za prenos (carry flag). Ako nikakvi podaci nisu dostupni, to se signalizira resevojanjem Z i C flega. Ako kanal ne može da prihvati podatak sa ulaza ili izlaza, o tome treba da bude obavešten program koji tretira greške. Standardni način tretiranja grešaka u ZX bejziku jeste putem jednog restart poziva na adresi 8. Na mašinskom jeziku to izgleda ovako:

ERROR RST 0008
 DEFB errocode
 gde je „errocode“ numerički kod poruke o grešci koja se daje korisniku.

Kanalni zapisi za tri standardna Spectrumova kanala i jedan dodatni kanal koji do sada nije opisan prikazani su u Tabeli 1.

Obratite pažnju na to da je svaki kanalni zapis standardnog formata i da jedino K kanal može da prihvati i ulaz i izlaz. Novi, R kanal Spectrum koristi interna za slanje podataka redaktorskog bafera. Naredba OPEN neće dozvoliti korisniku da poveže struju sa R kanalom, pa je tako njegovo korišćenje ograničeno.

Važna primedba: format jednog kanalnog zapisu drukčiji je kada se koriste mikrodrajv i interfejsi.

Tablica struja

Veza struja sa kanalima uskladištena je u tablici struja, u oblasti sistemskih promenljivih, na prostoru od 38 bajta, polazeći od STRMS (adresa 23568). Tablica struja i svaki par bajtova u toj tablici nose jedan broj „x“ koji predstavlja adresu početka jednog kanalnog zapisu. Naredstvo prostog skladištenja adresa kanalnog zapisu, „x“ je „rastojanje“ na kome se kanalni zapis načini od početka zone kanalnih informacija:

početna adresa = adresa kanala + x - 1
 zone kanalnih zapisa

Pošto postoji maksimalno 16 struja, moglo bi se pomisliti da su dovoljna 32 bajta (tj. jedna kanalna adresa po struci) za skladištenje svih asocijacijalnih kanala i struja. U stvari, dodatnih 6 bajta koriste se za skladištenje kanalnih informacija za tri unutrašnje struje koje odgovaraju brojevima struje 255, 254 i 253. Te tri unutrašnje struje su automatski povezane sa kanalima R, S, odnosno K i nisu dostupne iz bejzika. Međutim, prisustvuju te tri unutrašnje struje mora se uzeti u obzir kad se pokušava naći adresu kanala koji odgovara na kojoj od struja 0 do 15. Prva tri ulaza u tablicu struja odnose se na unutrašnje struje 253 do 255, a četvrti ulaz daje adresu kanala koji će se koristiti sa strujom 0 id. To znači da je početak tablice struja, što se spoljnih struja tiče, na adresi:

STRMS + 6 ili 23574

Adresa kanala povezanog sa strujom s (s je u okviru 0 do 15) skladišti se u dva memo-rijaka mesta sa početkom na

23574 + S/2

Kada se jedna struja otvoriti za jedan određeni kanal, ispituje se tablica struja da bi se našla adresu kanala. Kad je struja zatvorena, njeni adresi za ulaz u tablicu struja podešena je na nullu. Tako se nulli ulaz u tablici struja koristi za otkrivanje pokušaja da se koristi struja koja još nije bila otvorena.

Ovaj sistem kanalnih zapisova i tablice struja proširen je i na korišćenje mikrodrajva, ali njegove bitne karakteristike ostaju iste. Svaki kanal je opisan kanalnim zapisom, a struje su povezane sa kanalnim zapisima pomoću tablice struja.

Pre nego što predremo na razmatranje kako praktično iskoristiti poznavanje struja i kanala, vredni pomenuti jednu sistemsku promenljivu: CURCHL. U njoj se čuva adresa tekćeg kanala. Na taj način, nakon naredbe kao što je PRINT # S, CURCHL sadrži adresu kanala povezanog sa strujom „s“.

Kanali po meri

Bejzik naredbe IN i OUT predstavljaju najjednostavniji način za slanje, odnosno primanje podataka direktno sa odgovarajućeg ulazno-izlaznog porta. Na primer, ako je generatoru zvuka dodeljen port 31 za kontrolu njegove grevkencije, tada će

OUT 31,f

slati podatke (f = 0 do 255) koji će direktno uticati na visinu tona. Za jednostavne uređaje i uređaje koji se kontrolisu pojedinačnim bitovima, IN i OUT su vrlo pogodne naredbe. Međutim, ako je uređaj „znakovno orijentisan“, tj. ako prima i odasila podatke u obliku znakova, tada su IN i OUT neponedost. Na primer, paralelni štampač i modem su znakovno orijentisani uređaji, i najpode-

```

10 DATA 01,00,254,237,121,201
20 FOR a=23297 TO 23301
30 READ d
40 POKE a,d
50 NEXT a
100 GO SUB 1000
110 FOR i=0 TO 7
120 LPRINT i;
130 NEXT i
140 GO TO 110
1000 LET c=PEEK 23631+256*PEEK 2
3632
1010 LET c=c+15
1020 POKE c,23296-INT (23296/256
)
1030 POKE c+1, INT (23296/256)
1040 RETURN

```

sniji način da se sa njima radi jeste putem običnih PRINT i INPUT naredbi. Jasno je da najbolji put vodi preko struja i kanala.

Uključivanje novog periferijskog uređaja u sistem struja i kanala može se ostvariti na jedan od dva načina: ili promenom adresa uskladištenih u nekom postojećem kanalnom zapisu ili stvaranjem jednog potpuno novog kanalnog zapisu.

Privi metode se sastoji u ubacivanju (POKE) novih adresa u postojeći kanalni zapis. Kanal će, tako, biti usmeren na vaše sopstvene programe na mašinskom jeziku, smeštene negde u memoriji. Na primer, pretpostavimo da želite da povežete standardni štampač umesto ZX štampača. Umetno postojeće adrese na prva dva mesta kanalnog zapisu ZX štampača, (CHANS + 15), treba da postavite adresu sopstvenog programa za pogon štampača, i naredbe LPRINT i LLIST, kao i svaka druga naredba koja se odnosi na struje otvorene ka kanalu P, slade podatke novom štampaču.

Pisanje novog programa za upravljanje štampačem je, u principu, lako: sve što treba da se uradi jeste dovesti ASCII kodeve u A registar i poslati ih u štampač. Međutim, Spectrumov set znakova obuhvata i mnoge kodove koje standardni ASCII set karaktera ne poznaje. Pre nego što se pošalju preko štampača, ove kodove treba konvertovati. Na primer, svih kontrolnih kodo- u unutar PRINT naredbe biće poslati prema Dodatku A Spectrumovog priručnika. To znači da LPRINT TAB (10) šalje ASCII kodove 23, 10 i 0; 23 je Spectrumov kontrolni kod za TAB, a slediće dva koda su najmanji i najvažniji bajt parametra TAB funkcije. Ovi kodovi, međutim, za štampač mogu da znače nešto sasvim drugo. Isto tako, koodovi za bežik naredbe ne znače da štampač ama beš ništa. Oni se, zato, moraju prevesti u niz ASCII znakova, tako da se postigne efekat kao kada se svaka naredba kuca slovo po slovo.

Prije ovog metoda povezivanja nekog novog periferijskog uređaja predstavlja sledeći program koji umesto izlazne adrese uskladište u P kanalu postavlja adresu mašinskog programa koji je uskladišten u beferu za štampač. Ovaj sistem funkcioniše samo stoga što je ZX štampač van upotrebe! Nov izlazni program ne radi ništa korisno sa podacima, osim što ih šalje ka portu 254 koji kontroliše zvučnik i ivičnu

Tabela 2

23296	outdr	LD	BC, 254
23299		OUT	(C), A
23301		RET	

— unesi u BC registar 254
— pošalji sadržaj registra
A na port 254
— vrati se u bežik

Jednostavni program za opsluživanje periferijskih uređaja u mašinskom jeziku

boju; to obezbeđuje da se njegovi efekti vide i čuju: Rutina kojom se postavi ovi efekti data je u tabeli 2. Ona je korišćena u bežik listingu prikazanom u Programu 1.

Mašinski sistem izlaznog programa skladišti se u DATA naredbi u liniji 10 i unosi u memoriju pomoću linija 20 do 50 (23296 = početna adresa befera za štampač). Subrmina 1000 menjaj adresu u kanalnom zapisu za kanal P. Linija 1000 dobija adresu početka kanalne oblasti u c (CHANS), a zatim linija 1010 nalazi početak kanalnog zapisu za kanal P(CHANS + 15). Linije 1020 i 1030 ubacuju (POKE) adresu novog izlaznog programa u prva dva mesta u kanalnom zapisu. Ako startujete ovaj program, videćete ivični blesak i veoma divljiv promenju. Ako zavistavite program, dobijete daljji dokaz da novi izlazni program šalje podatke kontrolnom portu za border i izlistavanjem (LLIST) programa. (Pažnja: pre primene ovog programa treba odvojiti ZX štampača).

Novi kanali

Menjanje adresa uskladištenih u postojećim kanalnim zapisima predstavlja lak metod za povezivanje novih uređaja, ali, na žalost, on onemogućuje jedan od Spectrumbnih postojećih I/O uređaja. U praksi je nemoguće modifikovati kanal K, jer se njegove adrese ponovno uspostavljaju kad god se izvrši naredba INPUT. Kanali S i P ostaju, tako, kao jedini kandidati za modifi-

PROGRAM 2

```

10 DATA 0,91,11,91,69,1,0,254,
237,121,201,207,18
20 FOR a=23296 TO 23308
30 READ d
40 POKE a,d
50 NEXT a
100 LET s=5: GO SUB 1000
110 PRINT #1,RND;
120 GO TO 110
1000 LET a=23574+2*k
1010 LET c=PEEK 23631+256*PEEK 2
3632
1020 LET r=23296-c+1
1030 POKE a,r-INT (r/256)*256
1040 POKE a+1, INT (r/256)
1050 RETURN

```

23296	chanrec	BEFB	5
23297		DEFB	91
23298		DEFB	11
23299		DEFB	91
23300		DEFB	..E..
23301	outdrv	LD	BC, 254
23304		OUT	(C), A
23306		RET	
23307	indriv	RST 8	
		DEFB	18

kaciju. Pošto je kanal S isuviše dragocen, jedini stvarni kandidat je P. To je lepo sve dok ne osetite potrebu da istovremeno koristite više od jednog periferijskog uređaja.

Da bi se omogućilo istovremeno povezivanje nekoliko periferijskih uređaja, potrebno je urediti nove kanale. Dodavanje jednog novog kanala izgleda vrlo lako, ali treba urediti u obzir nekoliko sitnijih detalja. Pre svega, kanalni zapis je moguće stvoriti bilo gde u memoriji, ne samo u zoni kanalnih informacija; međutim, ako je kanalni zapis uskladišten iznad INPUT radne oblasti (počinje kod WORKSP), sistemski promenjivi CURCHL, u kojim se čuva adresa tekucog kanala, biće izmjenjen nakon svake naredbe INPUT. To, naravno, znači da će adresa kanala biti izgubljena i Spectrum će otkazati. Međutim, ako je kanalni zapis uskladišten ispod INPUT radne oblasti, tada sve funkcioniše kako treba. U našem primeru korišćen je befer štampača za skladištenje novog kanalnog zapisu i novih izlazno-ulaznih rutina. Druga teškoća je u tome što naredbe OPEN i CLOSE funkcionišu samo sa standardnim kanalnim zapisima za K, S i P, to znači da je potrebno obezbediti i potprogram za povezivanje kanala sa nekom strujom i, ako je potrebno, potprogram za njegovo zatvaranje. Sve to zajedno obezbeđuje mašinsku rutinu prikazanu u Tabeli 3.

Prihv pet bejtova čine nov kanalni zapis. Program koji počinje od adrese 23301 je izlazni program i on, jednostavno, šalje kod u registru A u izlazni port 254 — port zvučnika i ivične boje. Program koji počinje od adrese 23307 je ulazni program — on obaveštava o grešci da bi ukazao da se ovim kanalom ulazi nije dozvoljen. Naravno, u stvarnoj primeni svaki od ovih programa može da bude mnogo složeniji. Program 2 predstavlja bežik program za unošenje i aktiviranje mašinske rutine.

Linije 10 i 50 unose nov kanalni zapis i ulazno-izlazne rutine u befer štampača. Potprogram 1000 otvara struju s na nov kanal. Drugim rečima, ona je ekvivalentna naredbi OPEN # „E“. Linija 1000 izračunava tačnu adresu za struju s u tablici struja. Linije 1010 i 1020 izračunavaju tzv. offset preskak za novi kanalni zapis, a linije 1030 i 1040 skladište njegovu adresu u tablici struja. Linija 1000 koristi podprogram 1000 za otvaranje struje 5 ka ureduju E, a

- bajt manje težine u izlaznoj adresi
- bajt veće težine u izlaznoj adresi $(91 \times 256 + 5 + 2301)$
- bajt manje težine u ulaznoj adresi
- bajt veće težine u ulaznoj adresi $(91 \times 256 + 11 + 23307)$
- kod novog kanala („E“ = 69)
- unesi u registar BC 254
- pošalji sadržaj akumulatora na port 254
- vrati se u bežik
- pozovi rutinu za greške
- kod za pokretu „Invalid device“

Rutine za uvođenje novog kanala i opsluživanje periferijskih uređaja preko njega



linije 110 do 120 šalju kodove koji odgovaraju nasumičnim brojevima mazučnom i nivčnom kontrolnom portu. Zaustavljanjem programa i unošenjem naredbe LIST # 5 može se proizvesti i blesak boje i zvuka, što znači da je program izlistan ka portu 254! Ako promenite liniju 110:

110 INPUT # 5, i

dobijete poruku o grešci, što znači da se taj kanal ne može koristiti za ulaz. Naravno, umesto 254 može se postaviti adresa nekog štampača, programatora EPROM-a ili gene-

ratora zvuka.

Ovim što se moraju pisati obuhvatniji i specijalizovani programi, nema nikakvih teškoća u dodavanju novih kanalnih zapisa. Jedino treba voditi računa da gornji program neće funkcionišati ako su spojeni mikrodržavoj.

Uzlazni kanal

O problemima pisanja izlaznog dajvera već je ranije bilo reči. Na kraju, treba

Mali oglasi

- Spectrumovci** — veliki izbor najprograma po najnižoj ceni uz novi popusti: hobit, penetrator, horace, passi i drugi. Cena svih programa je 200 ND. Tel. 345-480
- Disk-drive za Apple (II, II+III), potpuno nov, prodajem. Tel. 011/4443-351

Software club

Garantujemo najniže cene u zemlji! Bilo koji program po vašem izboru za 100 din. — dešet programa za samo 600 dinara

- Manic Miner
- Penetrator
- Hobit.

mnogi drugi programi koji stizu iz Engleske. Svi programi su snimljeni i verifikovani direktno iz Spectruma. Pišite ili se javite telefonom: Andrija Dejan, Dantova 64, 11060 Beograd, tel. 011/784-041

- GALAKSIJA** — komplet čipova samo 7500 din. (2K verzija). Svačih dodatnih 2K samo 1750 din. Isporuča odmah. SHARP 1500 / printer 80.000 din. Sezović Javor. Pive Karamatićevića 31, Beograd. Tel. 011/496-665

ZX SPECTRUM 16/48K
i dalje najinffiniji software u gradu SAMO 50 DINARA PC PROGRAMU. Puno novih programi za 16 i 48K.
Skripte za mašinac.
Super brza usluga.
Specijalni programi po pružibinji.
TRŽITI NOVI BESPLATNI KATALOG.
Košić Vladimir, Varvarinska 35, 11000 Beograd, tel. 011/400-823

COMMODORE 64
Programski jezici
kompjajleri
assembleri
word procesori
najnovije igre
kučni programi
poslovni programi
muzički sintesažeri
sekvenci, kompzeri
ekdukativni programi
kurs BASIC-a 130 KB
govor-voice generator
super grafika
sprei idsheet
razmena programa
Veselin Milišavljević, 11000 Beograd, Vitanovačka 42, stan 45, tel. 011/462-659 (po podne)

I dalje sa Vama COMPUTER-LAND
Naša specijalnost je ZX SPECTRUM
Sistemski programi Namenski progr ami igre
Besplatni blic-katalog (na... i prog. r. cene) Katalog sa opisima svih progr. i cennama 100.- din
COMPUTERLAND J. Gagarina 120/62 11070 Novi Beograd TEL. (011) 162-774

ZX-Spectrum programi na kasetama. Za 80 din. nudim pas-cali, forth, devpac, arcadia, penetrator, time-gate, jet-pac, flight simulation, football manager, 3D-tunnel, hobbit, cyrus chess, manic miner, monopoly, kong, mr. wimpy, pool, paintbox, ant-attack, highnoon, bridge, battle 1917 i još 250 i drugih. Za katalog poslatite 20 dinara. Razmenjujem VHS-video materijal. Marko Marković, Dl. Bijedica 27/XI, 71000 Sarajevo

Za comodore-64 VIC-20 preko 200 fabričkih programa prodajem. Besplatni katalog? Možete razmeniti? Jelton? Andrija Količić, Vojvode Branki 31/II, 11000 Beograd, tel. 011/424-325

Pod velom tajni: Iako se smatra da se o njemu već odavno sve zna. Spectrum ne prestaže da iznenadjuje programere

pomenuti dodatne zahteve za ulazni dajver. Ako jedan ulazni kanal treba, poput osmobiltnog A-D konvertora, da prihvata pojedinačni znakovni kod, tada je najbolja bežik naredba INKEYS, # koja će vratiti pojedinačni znak. Međutim, ako planirate da koristite INPUT da biste učitali čitav znakovni niz, tada morate biti svesni dve stvari. Prvo, naredba INPUT ima i izlaznu struju, i nije dovoljno zameniti izlazni program programom za tretiranje greške; mora se obratiti svaki podatak koji šalje naredbu INPUT, čak i u slučaju da se ona pristo zanemaruje Drugo, naredba INPUT prihvata podatke kao da su otuknani na tastaturi. To, praktično, znači: ako koristite INPUT # s: B, da biste učitali neki broj u promenljivu B, program mora da šalje nešto ASCII kodova koji odgovaraju brojevima i da se završi sa jednim ENTH kodom, kao da broj ulazi sa tastature. Na kraju, treba obratiti pažnju i na ovo: čak kada se podaci očitavaju i sa specijalnog komada hardvera, naredba INPUT će tačno interpretirati sve kodove — one uvek funkcioniše kao da prima struju znakovnih kodova koji odgovaraju dirkama koje su pritisnute na tastaturi.

Spectrumov sistem struja i kanala predstavlja neочекivano premiju za programere na ZX bežikiju. Korisan umutar programa, on obezbeđuje nezavisnost uređaja i uopšte povećanje elastičnosti bez ikakvih negativnih strane. Programeru na mašinskom jeziku struje i kanali pružaju idealnu alatku za softverske interfejsse sa novim uređajima. Primeri dati u članku predstavljaju polaznu tačku za sve realne softverske interfejsse koje biste možda želeli da napišete.

(Electronics and Computing Monthly)



Sve za ZX SPECTRUM

- video igre
- igre
- namenski programi
- knjige
- SPEC MALONO!!**

Ako vam trebaju usluge na printeru obratite nam se.
VRŠIMO:

- programski listing
- as embriks listing
- kopiranje crteža u vizueloj rezoluciji.

Iskorištite LETNI POPUST!! MILOVANOVIC LJUBIŠA
Petar Leković 57
11000 BEOGRAD
tel: 011/558 007

Skripte za ZX81 i spectrum na srpskohrvatskom

1. program (800 N. din.)
2. mašinski jezik (800 N. din.)

3. najbolji i najbrži listinski programi iz stranih časopisa (800 N. din.) 50 str.

4. šeme i uputstva za hardver dodatku u samogradnji (A/D konvertor, 1/0 port, RS232 interfejs, light pen i mnogi drugi...) na (800 N. din.)

Vlasnici kompjutera Commodore 64 i commodore VIC 20

Javite se radi razmene programa i literaturu. Količić Andrija, Vojvode Branki 31/II, 11000 Beograd, tel. 011/424-325

škola akcionalih igara

Igre na računaru

Cilj škole je da vas, kroz dosta primera, uvede u neke programerske veštine koje će vam pomoći da kreirate sopstvene akcione igre. Neki od primera programa predstavljaju interesantne igre koje, doduše, ne mogu da se mere sa komercijalnim programima, ali koje, obzirom da čete ih sami otkucati i shvatiti njihovu realizaciju, mogu da vam pruže mnogo časova zabave, naročito ako spadate u one vlasnike računara koji ne žele samo da se igraju.

Svi znamo da su dobre akcione igre moguće samo na mašinskom jeziku, a ipak smo ova školu proprijalj bežiš primerima. Zašto? Čini nam se, pre svega, da je nedovoljan broj čitalaca ovih redova dobro upoznat sa mašinskim jezikom, pa bi mašinski programi odvlačili pažnju od onoga što program treba da ilustruje. Kada savladate mašinac, pogolovo, ako posedujete knjigu „Disasembliarni Spectrumov ROM“, neće vam biti teško da primere bukvvalno prevedete na mašinski jezik, ili da, u programima koje ćete sastavljati, koristite tehničke naučene u ovoj školi.

Škola je, u osnovi, namenjena vlasnicima Spectruma, koji će imati privilegiju da sve date primere isprobaju na svom računaru bez ikakvih prepravki. A za vlasnike drugih računara, međutim, može da bude korisno da je pročitaju: tehnike koje ćemo objasniti su univerzalne, dok naredbe, jasno, mogu da budu izmenjene. Dovoljno je da držite uputstvo za upotrebu vašeg računara na dohvatu ruke dok čitate ove redove i sigurno ćete se snaći bez većih problema.

„Pritisni bilo koji taster...“

Nekada su računari bili namenjeni uglavnom obradbi brojeva koji su se unosili primenom naredbi koje su vrlo slične dobro nam poznatoj naredbi INPUT. Jasno je da na ovaj način (korisniku je ostavljeno skoro beskonačno vreme za razmišljanje i prijem podataka kvari izgled ekranata) ne mogu da se primaju podaci potrebitni za akcione igre, pa je u bezijk uvedena naredba INKEY\$, koja daje kod pritisnutog tastera. Ako se, na primer, u programu nalazi naredba 5 LET XS = INKEY\$, promenljivoj XS će biti dodeljeno slovo a ako je igrač pritisnuo a, a ako je pritisnuo SHIFT A i u tome slično. Kod Spectruma je važno primeti da naredba INKEY\$ ne registruje trenutak kada je taster pritisnut, kao kod mnogih drugih kompjutera. Ukoliko, na primer, računar neprekidno izvršava naredbu 5, a korisnik stalno drži taster A pritisnut, X će neprekidno dobijati vrednost „A“.

Naredba INKEY\$, i pored ogromnih prednosti nad INPUT, nije pogodna za akcione igre. Zašto? U mnogim igrama je neophodno da se korisniku omogući istovremeno pritiskanje dva ili više tastera (krejanje i puštanje, na primer). Zato Spectrum poseduje moćniju i, na žalost, komplikovaniju naredbu IN. Ona ne daje kao rezultat alfumerički nego broj. Ova naredba, osim toga, ne testira jedan nego čitavih pet tastera u isto vreme; tih pet tastera ćemo nazivati blok.

Svaki blok tastera ima svoj broj koji nalazimo u tabeli 1 i stavljamo iz IN naredbe (ako, na primer, želimo da testiramo tastere 1, 2, 3, 4, i 5, koristimo naredbu LET A = IN 63486). Ukoliko nije jedan od tastera iz bloka nije pritisnut, IN će vratiti broj 255 (uz jednu malu ogragu ali o njoj nešto docnije). Svaki pritisnuti taster smanjuje ovaj broj za 2 dignuto na stepen n, gde se u takode čita iz tabele. Neka su, na primer, pritisnuti tasteri 1, 2 i 4. PRINT IN 63486 će tada dati broj $255 - 20 - 21 + 2 = 244$. Oni malo bolje upoznati sa binarnom logikom, svakako će zapaziti da je svakome tasteru dodeljen po jedan bit koji je „setovan“ ako je taster otputišen „resetovan“ ako je on pritisnut. Zato je sasvim jednostavno ispitati da li su, na primer pritisnuti istovremeno tasteri 1 i 2, pri čemu se ne obraća pažnja na stanje preostala tri i naredba kojom se to postiže: JE (IN 63486 AND 3) = 0 THEN PRINT pritisnuti su 1 i 2“.

Čin nam se da će vam biti od koristi da uložiti određeni napor u razumevanje ove naredbe, jer ćete slične imati prilike da koristite u mnogim drugim prilikama. IN 63486 daje broj između 0 i 255 koji možemo da zamislimo kao niz od 8 binarnih jedinica i nula. Desna dva bita odgovaraju tasterima 1 i 2 respektivno ukoliko je neki od njih 0, odgovarajući taster je pritisnut. Preostali bitovi, međutim, imaju sadržaj koji zavisi od toga da li su tasteri 3, 4 i 5 pritisnuti, što nas, kao što rekosmo, ne interesuje. Treba, dakle, da pronađemo način da zanemarimo vodećih 6 bitova, dobijenog broja odnosno da ih pretvorimo u nulu. Upravo je tome namenjena konstrukcija... AND 3. Zapazimo, najpre, da se broj 3 prikazuje binarno kao 00000011. Logička funkcija AND („i“) nalaže računaru da proredi brojeve IN 63486 i 00000011, bit po bit. Ako su obe bita koja se porede jedinice, rezultat će takođe biti 1, a protivnovo rezultat će biti nula. Obzirom da broj 3 ima šest vodećih nula, prvi šest bitova razuljata će takođe biti nula, što smo i zeli. Poslednja dva bita neće promeniti vrednost jer ih logički množimo (AND se ponekad naziva logičko množenje a OR logičko sabiranje) jedinicom. Ukoliko su tasteri 1 i 2 pritisnuti, ta dva bita (i, samim tim, čitav rezultat) će biti jednakni nuli, dok će u protivnom rezultat uvek biti različit od

nule. Ovakvo „maskiranje“ je vrlo često u upotrebi, pogotovo kada pišemo mašinske programe jer predstavlja najcešćodniji način izdvajanja nekoliko bitova i jednog bita.

Kao primer svega izloženog, dajemo jednostavni program za pisanje i čitanje po ekranu. Iako to nije akcionalna igra, u svakoj akcionaloj igri mora da se čita po ekranu zavisno od onoga što korisnik kuca, pa će tehnike naučene analizom ovoga programa svakako biti od koristi. Primelite da se u programu na više mesta koristi naredba INKEY\$ — pokazuje se da je ona jednostavnija od IN, pa zato ovu drugu treba koristiti samo u slučajevima kada postoji mogućnost da su pritisnuta dva tastera istovremeno — u našem primeru u naredbama 50 i 60.

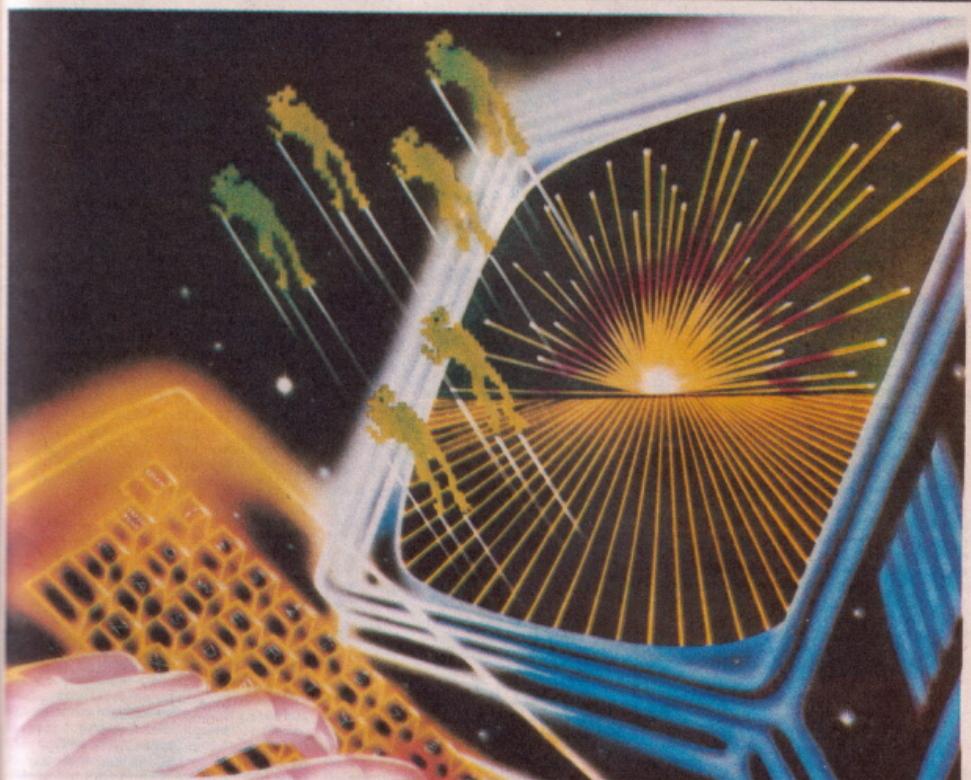
Obratimo, najzad, pažnju na liniju 20 u kojoj se skriva naredba BEEP koja, naravno, ne smeta, ali nije ni mnogo potrebna. Došlo je, dakle, vreme da izložimo „ogrudu“ koju smo pomenuli: Spectruman operativni sistem ne mora uvek brojewe pritisnutih tastera oduzimati od 255 — u upotrebi su i druge vrednosti. U nekim slučajevima, međutim, računar se sigurno vraća u „normalizovanu“ stanju i usvaja ovu konstantu. Najjednostavniji recept za ovu „normalizaciju“ je izvršiti BEEP pre ispitivanja tastera pomoću IN. Ovo objašnjenje mora, na žalost, da primite kao i mi — zdravo za gotovo. Naredba IN, jednostavno, ima mnoge druge namene osim ispitivanja tastature, pa se pojavljuju i neki artefakti.

Marsovcii dolaze

Sve akcione igre mogu da se kreiraju korišćenjem naredbe PLOT, tj. crtanjem figure svaki put kad se ukaže potreba tačku po tačku. Ova način je mukotran i, što je mnogo važnije, ne daje dobre rezultate kada je potrebna animacija. Često su oblici koje treba prikazivati relativno mali (figura čoveka u „pakmenu“, svemirski brod u „invejdinsima“, voćka kod „Horacija“...), pa možemo da ih svedemo na veličinu jednog karaktera, koji ćemo, jednostavno, prikazivati na ekranu god je god je potrebljano. Karakteri se kod Spectruma (i kod većine drugih sličnih kompjutera) definisu na matrići 8×8 kao što ćemo videti iz sledećeg jednostavnog primera.

Pretpostavimo da želimo da definisemo karakter u obliku figure čoveka sa slike. Najpre smo ga, kao što se vidi, upisali u matričnu kvadratiku tako da su neka polja crna a neka bela (često je lako ukloniti neki složeniji oblik u prilično malu matriču — za ovu umetnost bivaju angažovani i profesionalci kada se pišu komercijalni programi). Zatim ga pretvaramo u brojeve i to red po red (vlasnici drugih računara će uštedeti sestra napora ako provere da se to na njihovom kompjuteru ne radi kolonu po

Svaki vlasnik Spectruma je sigurno nabavio neki komercijalni program za akcionu igru (neki ih imaju na desetine, a možda i na stotine) i, upotrebljavajući ga, pozavideo njegovom autoru: kako je moguće da je igra ovako dobro zamišljena, da je ekran ovako efektan, da su zvučni efekti ovako dobri... Dobru akcionu igru, zaista, nije lako sastaviti — ne toliko zbog toga što je teško napisati program koliko zato što nije jednostavno zamsiliti dobar i originalan scenario koga se neko drugi već nije setio. Kada zamislite scenario, realizacija programa je stvar tehnike, koja, kao što rekosmo, nije teška, ali je lakša ako sami ne morate da je otkrivate od samog početka. To je i smisao ove škole akcionih igara.



kolonu, što je vrlo čest slučaj) pri čemu "zamenjeno" polje zamenjujemo jedinicom, a ono koje to nije — nulom. Konstruktori Spectruma su nas snabdeli funkcijom BIN koja ovako dobijene binarne brojove prenosi u dekadne. U našem primeru dobivamo:

red 1: BIN 00001000
 red 2: BIN 00010100
 red 3: BIN 00001000
 red 4: BIN 00111110
 red 5: BIN 00001000

red 6: BIN 00011100

red 7: BIN 00100010

red 8: BIN 00100010

Najbolje je da, kada smo ispisali ove brojeve, uposlimo računar na njihovom prevođenju u dekadne. Za prvi red otkucamo, jednostavno, PRINT BIN 00001000 i dobijemo broj 8, za drugi broj 20 i tako dalje. Zatim unosimo jednostavan program kao što je sledeći, startujemo ga i, od tog momenta, karakter „A“ postaje čovečuljaka slike.

Da bismo vam olakšali pisanje sopstvenih akcionih igara, dajemo tabelu sa nekoliko predloženih karaktera. Tu ćete naći sadržaje koje treba uneti u DATA linije ha kraju programa da biste imali definisane

obilike koji se često koriste. Jasno je, naravno, da korišćenje karaktera koje će poznavati i hiljade drugih nije neka prevelika originalnost, ali će lako, promenivši neke detalje, dobiti mnogo originalnije oblike. Kada se radi na maloj matrici (u ovom slučaju 8×8), čak i promena jednog polja iz temelja menja čitav utisak o karakteru. Dajte, dakle, na volju vašoj umetničkoj mašti!

Zvučni efekti

Nijedna komercijalna igra ne vredi mnogo bez interesantnih zvučnih efekata. Konstruktori Spectruma su, na žalost, ostavili jedino ne naročito fleksibilnu naredbu BEEP, koja ima dva argumenta razvođenja

```

IN63486 : keys 1,2,3,4,5
IN64510 : keys Q,W,E,R,T
IN65022 : keys A,S,D,F,G
IN65278 : keys Caps Shift,Z,X,C,V
IN61438 : keys 6,7,8,9,0
IN57342 : keys Y,U,I,O,P
IN49198 : keys H,J,K,L,ENTER
IN32766 : keys B,N,M,Symbol
Shift,Break/Space

```

SLIKA 2

```

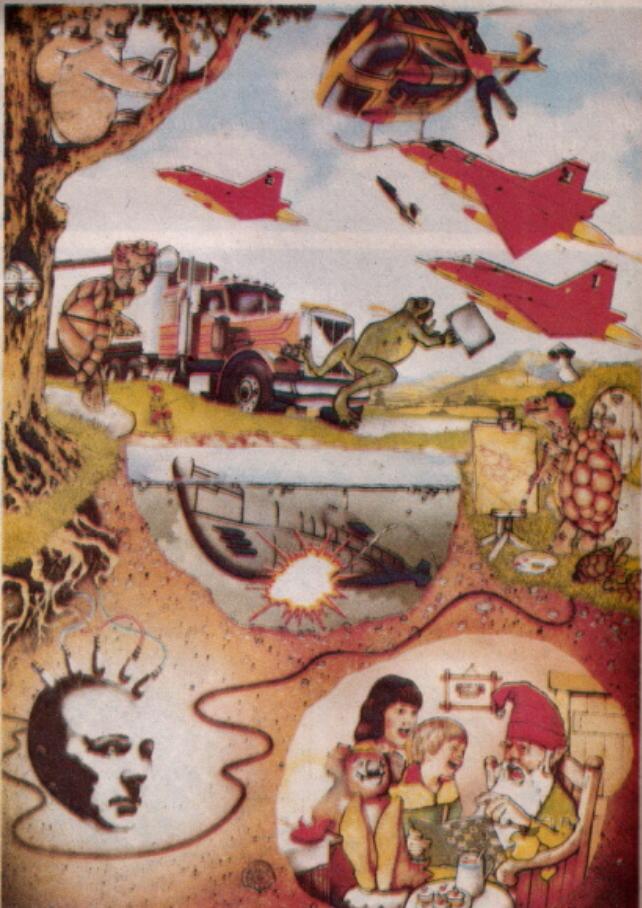
10 GO SUB 1000
15
20 BEEP .1,10
25
30 IF m=1 THEN OVER 1
40 PLOT INK ink;x,y: LET a==:
LET b==y
50 LET x==x+(IN 65022=251 AND x<254)-(IN 65022=254 AND x>1)
60 LET y==y+(IN 64510=253 AND y<166)-(IN 65278=251 AND y>1)
70 PLOT INVERSE (n=2): INK ink
;a,b
80 LET a==INKEY$:
90 IF a=="0" AND a<="7" THEN
LET ink==VAL a$:
100 IF a=="8" THEN PRINT AT 0,m
#11: FLASH 0: OVER 1: INVERSE 0;
" " " : LET m=m+1<3 AND m=2: PRINT AT 0,m#11: FLASH 1: 0
VER 1: INVERSE 0;
" " "
110 OVER 0: GO TO 30

120
1000 LET x==127: LET y==87: LET in
k=0: LET m=0
1010 BORDER 0: PAPER 7: INK 0: C
LS
1020 PRINT "Kretanje: "" W-laz
e", "X-dole", "A-levo", "D-desno"
1030 PRINT " " + dijagonalne komb
inacije"
1035 PRINT " B-Promena opcije"
1040 PRINT " " " Pritisni ne
ki taster."
1050 IF INKEY$>" " THEN GO TO 10
50
1060 IF INKEY$="" THEN GO TO 106
0
1070 CLS
1080 PRINT FLASH 1;"CRTANJE.": F
LASH 0;" POMERANJE, BRISANJE,
"
1090 PLOT 0,167: DRAW 255,0
1100 RETURN

```

zarezima: prvi određuje trajanje tona, a drugi frekvenciju. Kao što vidimo, ne posjeduju mogućnost da se ton, u toku „izvođenja“, menjaju po amplitudi ili po frekvenciji. No, to je tek sporedni problem sa naredbom BEEP; glavni je okolnost da računar ne može ništa da radi dok svira. Sa jedne strane, što je igra zanimljivija u njoj ima više zvučnih efekata, ali je, sa druge strane, i manje dinamična jer računar gubi vreme na nešto što je, ipak, sporedno.

Kada pišemo bežijk program sa zvučnim efektima, najbolje je da koristimo READ i DATA naredbe koje nam omogućavaju da



Važnije od tehnike: Maštovitost i inventivnost predstavljaju ključ za dobru akcionu igru — sve ostalo se postiže gotovo samo po sebi

uz minimum memoriskog utroška proizvedemo druge melodije. Na slici je dat jedan jednostavan primer zvučnog efekta koji bi mogli da stvaraju napadači iz vasiione u toku spuštanja.

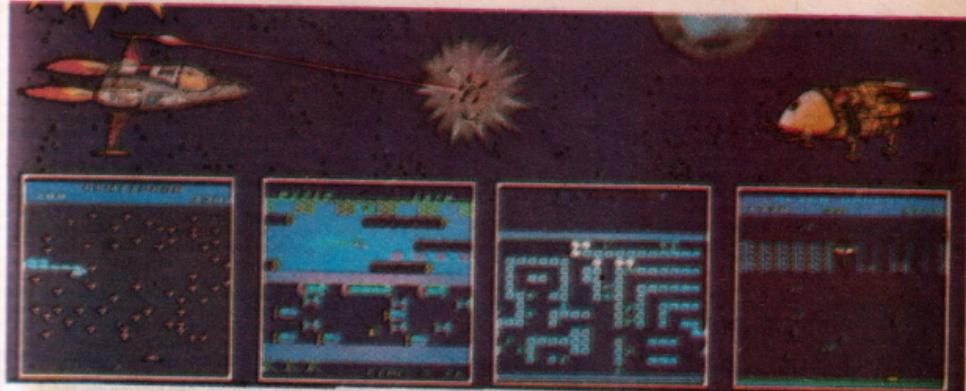
Vidimo da za svaki ton moramo da navedemo dva člana DATA liste, što nije samo veliki utrošak memorije već i gubitak vremena (READ nije ni izdaleka jedna od najbržih Spectrumovih naredbi). Zato je dobro odrediti se promjenljive dužine tonova i nadoknadići ih kompozitorskom vještinom, kao u drugom primeru.

Na kraju poglavljia o tonovima jedan praktičan savet koji je više hardverske prirode. U prodaji se nalaze takozvane zvučne kutijice za Spectruma („Sound boxes“). Njihova uloga je da pojačaju tone koje računar generiše, čim zvučni efekti postaju

interesantniji ili, bar, privlače pažnju šireg auditorijuma. Ovakve kutije možete sasvim uspešno zamjeniti bez ikakvog posebnog troška. Povežite, jednostavno, MIC III EAR na Spectrum sa MIC na kasetofonu, pritisnite REC i PLAY (bez umetnute kasete) — to je sve. Zbog ekonomičnosti nekih rešenja Spectrumbog hardvera, kasetni ulaz i izlaz i generator tonova su tems povezani, što omogućava da se kasetofon koristi kao pojačalo.

Crtač u računaru

Svi znamo kako u bioskopu vidimo „po-kretne slike“: svake sekunde preko ekran-a „poleteće“ 24 nepomične slike, ali mi, zbog tromosti oka, imamo utisak kretanja. Jasno je da nešto slično možemo da pokušamo kod računara, ali jednostavni eksperiment pokazuju da su sanse samo teorijske: računar, pre svega, ne može da u jednoj sekundi prepiše 10 (jedan manje 24) sadržaja ekran-a koji, inače, zaprema 8 Kb RAM-a. Osim



SLIKA 4

```

9900 RESTORE 9100
9920 FOR n=0 TO 7
9930 READ a
9940 POKE USR "a"+n, a
9950 NEXT n
9970 RETURN
9100 DATA 8,28,8,62,8,28,34,34

```

SLIKA 5

```

3000 RESTORE
3010 FOR i=1 TO 7
3020 READ duzina,nota
3030 BEEP duzina,nota
3040 NEXT i
3050 DATA .2,7,.1,2,.1,2,.2,4,.4
2,2,6,2,2,7

```

Izgled memorije od 48 Kb može da primi samo šest ekranu, što nije dovoljno ni za pol sekunde „televizije“. Pa, kako je onda kompjuterska animacija uopšte moguća? Tajna je, naravno, u tome što najveći deo ekranu ostaje statičan, a pokreću se samo mali delovi (na primer, ranije definisani karakteri) i to ne prizvoljno nego tako da se njihov položaj u svakom trenutku može izmjeniti iz prethodnog položaja. Programer treba da bude toliko vešt da, promenom boja, istakne delove ekranu koji se pomenu, a neutrališe statičnu pozadinu. To se mora uvek da bude mnogo teško iako, nekako, zahteva određeno iskustvo. Mi ćemo početi od najnedostavljivijeg: karakteri koji se kreće po ekranu.

Posmatrajmo primer sa prve slike koji predstavlja najnedostavljiviji primenjujuću ideju. Ako otkucate program, videćete samo tačku koja nepriljivo treperi. Šta? Naredba 30 se izvršava odmah posle naredbe 20 što znači da kvadratični biva osman praktično trenutno. Sa druge strane, da bi kvadratični bio ponovo ispisani, računar treba da obavi jedno sabiranje i učin GO TO naredbu što, naravno, traje besno vreme. Tačka je, dakle, mnogo lže isključena nego što je uključena i to izazvodi nepričutan efekat.

Zadovoljstvo bez premcu: Za mladog programera nema uživšenijeg trenutka od momenta kada na ekranu oživi njegova sopstvena igra

Ako već ne možemo da ubrzamo sabiranje i GO TO naredbu, možemo da produžimo interval koji tačka provodi na ekranu dodavši jednu „mrtvu petlju“. Stvar možemo da začinimo i sa malo muzike:

```

24 FOR N=1 TO 5 : NEXT N
27 BEEP .01,RND 20

```

Sada je već mnogo bolje, ali se ispoljava novi problem: čim kompjuter pokuša da štampa karakter izvan ekranu (na to ne treba tako dugo čekati), biva prijavljena greška. Prva ideja koju bismo mogli da primenimo je IF naredba koja ispituje da li se stiglo do kraja ekranu. No, Spectrumska IF naredba je spora i bolje bi bilo iskoristiti FOR-NEXT petlju (9 i slično od 1 do 31), ali postoji i elegantnije rešenje: da karakter koji se kreće, kada dođe do jednog kraja ekranu, nestane i pojavi se na suprotnom. Na sledećoj slici je dat program koji to omogućava. Početne linije izgledaju sasvim jasno sve dok ne stignemo do linije 40. Tamo nisu očekujete neobična konstrukcija: LET $x=x+(y=31)$. Šta je to?

Malo bolji poznavaoči Spectruma znaju da se i logički izrazi izražavaju brojevima: ($y=1$) je logički izraz koji može da bude tačan ili lažan. Ako je tačan, računar će mu dodeliti broj 1, a ako je ne tačan — broj nula (kod mnogih drugih računarskih situacija je potpuno ista osnova što se tačnom iskazu dodeljuje vrednost -1; ovo ćete najlakše ispitati ako otkucate PRINT (3=3) i pogledate rezultat). Kada, dakle, u postane 31, x će biti povećano za jedan, što znači da će računar početi da piše od početka sledeg reda.

Drugi deo naredbe 40 izgleda još zagonetnije jer je ponovo iskoriscena naredba AND. Smisao ove naredbe je, da najpre to raščistimo, da računar ne prijavi grešku kada tačka, liniju po liniju, stigne do kraja ekranu. U tom slučaju bi trebalo da x ponovo postane nula (j. da se tačka pojavi u gornjem ugлу ekranu). Da vidimo šta se dešava u oba kritična slučaja: kada je x jednak 22 i kada nije.

Ako je x jednak 22, onda i $x=22$ (obratite pažnju na redosled operacija) ima vrednost 1 koja, logički pomnožena sa 22, daje oper 22, što smanjuje x na nulu. Ako, pak, x nije 22, $x=22$ će dobiti vrednost nula, pa će nula logički pomnožena sa 22

dati ponovo nulu koja neće menjati vrednost X. Ako ste shvatili ovu logiku, ne bi trebalo da imate problema sa razumevanjem linije 50 koja proverava da li je stiglo do kraja ekranu. Tu liniju ćemo vam, dakle, ostaviti za vežbu.

Glavom o zid

Tako smo, na relativno zadovoljavajući način, rešili problem kretanja tačke po jednoj liniji. Jasno je, međutim, da nam to nije dovoljno — potrebna nam je tačka koja će se mnogo slobodnije kretati po ekranu i odbijati se od drugih objekata na njemu. Za to moramo uvesti promenljivu koja će govoriti o pravcu kretanja. Za slučaj da se kretanje obavlja po jednoj pravoj, ta promenljiva će se posle svakog štampanja i brišanja dodati trenutnoj vrednosti ikse (ako promenljiva ima vrednost 1, karakter će se kretati po jednoj polje udesno), dok se promena smera postavi jednostavnom promenom znaka promenljive. Ovaj slučaj je, međutim, previše jednostavan, pa ćemo se pozabaviti kretanjem u svim smerovima. Njega karaktere promenljivu d, koja će imati vrednost 1 za kretanje dole, 3 za gore, 0 za desno i 2 za kretanje levo (za trenutak ćemo videti zašto su brojevi tako raspoređeni), a x i y će, kao i do sada, označavati položaj tačke na ekranu:

```

1000 LET X=X+(d=1 OR d=2)-(d=0 OR d=3)

```

1010 LET y=y+(d=2)-(d=1)

Ukoliko karakter „putuje“ preko ekranu i udari u horizontalni zid, d treba da promeni vrednost. To, na primer, možemo da postignemo naredbom:

```

2000 LET d=d+1-(2 AND (d=3 OR d=1))

```

Ako ste početnik, ove naredbe mogu da vas zaplaše. No, ne brinite, nije teško razumeti njihovo dejstvo: dovoljno je da se odlučite za neke konkretnе vrednosti x, y i d i ručno, korak po korak, izračunate kako će se one promeniti posle izvršavanja ovakih naredbi. Posle najviše dva prve sve će biti sasvim jasno!

Krajnje je vreme da izložimo nekoliko načina na koje može da se dođe da li je došlo do sudara dva objekta koja se kreću po ekranu. U nekim slučajevima možemo, jednostavnim IF, uporediti promenljive u koje su upisane njihove pozicije. U mnogim slučajevima, međutim, nije ekonomično trošiti memoriski prostor na koordinatne svakog objekta (zamislimo samo da pišemo

program u kome igrač treba da prođe kroz „prebukirani“ pojas asteroida) kada je on već upisan na ekrani, odakle ga, uz malo veštine, možemo pročitati.

Prvi način da to učinimo je naredba SCREEN (x, y) gde su x i y koordinate pozicije na ekrani koju želimo da „pročitamo“. Naredba će vrati karakter koji se nalazi na tom mestu uvađa ograničenja koja su veoma ozbiljna: predefinisani i inverzni karakteri se nekorektno detektuju.

Drugi način je naredba ATTR koja, istina, ne daje obaveštaja o karakteru na nekom mestu, ali zato govoriti sve o njegovoj boji. Ukoliko uspemo da svaku vrstu objekata obojimo na određeni način ATTR će nam pomoći da se orijentisemo. Kao i SCREEN, ATTR ima dva argumenta koji određuju mesto na ekrani. Kao rezultat se dobija jedan bajt „spakovani“ kao zbir sledećih sabiraka:

128 ako karakter blinkuje
64 ako je karakter posebno svetao
8 puta boja „papira“ (PAPER)
1 puta boja „masti“ (INK) kojim je karakter „pisan“.

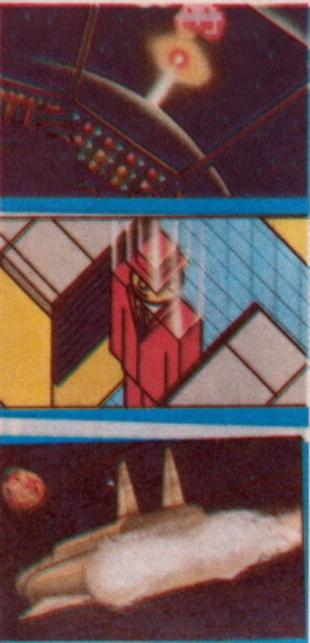
ATTR ne obraća pažnju na to da li je karakter standardan ili ga je korisnik sam definisao iz jednostavnog razloga što i ne analizira taj karakter — kao što joj i ne govorii. ATTR analizira atribute karaktera (o atributima smo dosta govorili u „Računari-ma“ 1. kada smo objašnjavali kako Spectrum kodira sadržaj ekrana).

Poslednji način predstavlja naredba POINT koja nije u velikoj upotrebi iako je neobično moćna. Prate je dva argumenta koja daju poziciju na ekrani, ali ne poziciju karaktera nego poziciju tačke, kao kod naredbe PLOT x, y. POINT vraca broj 1 ili 0 — jedan ako je tačka čije su koordinate dosta osvetljena, a nula ako je zatamnjena. Zašto je ova moćna naredba kojom možemo da ispitamo status bilo koje tačke na ekrani manje u upotrebi? Jednostavno zato što su bežički programi spori i ne mogu se da dozvole lukuš rad sa tačkama — radi se isključivo sa karakterima. Primenom POINT možemo, jasno, da ispitamo svaki karakter tačku po tačku, ali takvo ispitivanje oduzima previše vremena da bismo mogli da ga preporučimo.

Da ilustrijuamo sve ovo, objavljujemo program „Lopta i reket“. Ozbirno da se kreću samo dva objekta (lopta i reket), igra je prilično brza i dinamična. Kao i obično, ne treba samo da se zabavljate njima (za zabavu im i mnogo boljih igara), nego da pokušate da ga analizirate, u čemu će, nadamo se, pomoći i komentari uključeni u REM liniji.

Skrolovanje

Skrolovanje je trenutak kada se svi objekti (slova, specijalni karakteri i crteži) pomeraju nagore za jedan red da bi se oslobođio prostor za novu liniju. Za normalan rad i pisanje programa (posebno kod kompjutera kod kojih su naredbe kucaju po čitavom ekranu: kod Spectruma su za naredbe odvojeni samo poslednji redovi), skrolovanje je svakako neophodno. Spectrum, međutim, posle ispisivanja svake 22 linije očekuje od korisnika da potvrđuju odgovor na pitanje Scroll? (za polvrdan odgovor se smatra pritisak na bilo koji



Iz jedne avanture u drugu. Prizori iz igračkog života

SLIKA 6

```
2000 FOR n=10 TO -10 STEP -.5
2010 BEEP .02,n
2020 NEXT n
```

SLIKA 7

```
10 LET y=0
20 PRINT AT 10,y;""
25 PAUSE 1
30 PRINT AT 10,y;""
40 LET y=y+1
45 IF y>31 THEN GO TO 10
50 GO TO 20
```

SLIKA 8

```
10 LET x=0: LET y=0
20 LET a=x: LET b=y
30 PRINT AT x,y;""
40 LET x=x+(y>31): LET x=x-22
AND x=22
50 LET y=y+1<32 AND y=31>
60 PRINT AT a,b;""
70 GO TO 20
```

taster osim na „N“ i BREAK), što može da bude vrlo neprigodno za akcione igre. Jedan od načina da izbegnemo ovu sistematsku poruku je da, na samom početku programa, upotrebito POKE 23692. — 1.

Skrolovanje je vrlo pogodno ako se neki objekti kreće prema vrhu ekrana uz unereno pokretanje po x osi. Jedan od najjedno-

stavnijih načina da isprobate skrolovanje je da štampatе upravo definisani karakter u obliku malog svemirskog broda u poslednjem redu, a zatim koristite „prazne“ PRINT naredbe koje će ga pomeriti prema vrhu ekrana. Videćete da je pokretanje broda daleko brže nego u ranijim primerima gde smo koristili PRINT AT. Razlog je u tome što skrolovanje obavljaju mašinski program koji je mnogo brži od svakog bežička.

Kada smo se uverili u korisnost skrolovanja, požećemo da ga upotrebimo i po x dimenziji (sleva na desno ili sdesna nalevo). Na žalost, mašinski potprogrami koji bi ovo omogućili ne nalaze se u ROM-u, a svaki bežički koji treba da pomeri 8 K video memorije bi bio toliko spor da ga ne treba ni pominjati, ipak, kao vežbu, pokušajte da napišete bežički program koji pomeri ekran za jedno mesto uлево. Ako uspete, zaslužujete čestitke i kompliment da je krajnje vreme da definitivno pređeta na mašinac.

Ako nemate volje za pisanje mašinskih programa, dajemo vam dva koja će vam lepo poslužiti: prvi omogućava levi, a drugi desni skrol. Kao što se vidi, ova dva programa ukupno imaju samo 46 bajtova, što znači da su toliko optimizovani da se početnički neviđaju upustiti u njihovu analizu: zato i ne dajemo disasemblišani listing.

Glavna nevolja sa skrolovanjem je što se ne pokreće samo ono što bismo mi želeli. Na ekranu se, naime, obično nalazi i neki broj brodova, maksimalni skor, abstraktne uputstva za upotrebu ili nešto slično. Sukcesivno pomeranje ekrana izaziva, jasno, i gubljenje ovoga teksta, što je prilično neprijetljivo. U komercijalnim igrama se često koriste mašinski potprogrami koji skroliju samo deo ekrana ali se, u najvećem broju slučajeva, može proći i bez njih: jednostavno koncipirajte sadržaj ekrana tako da su informacije koje stalno treba prikazivati u prvom ili eventualno prva dve reda, a zatim ih ponovo štampanje svaki put kada iskoristite PRINT da bi ekran skroloval. Na taj način, ujedno, rešavate i problem promene skora u toku igre, a Spectrum je dovoljno brz da se ovo stalno ispisivanje mnogo ne primeti.

Strgo kontrolisani bajtovi

Izgleda da smo se približili mašinskom jeziku — došli smo do naredbi kojima možemo da pristupamo čitavoj memoriji našeg Spectruma. One su neobično moćne ali imaju i loše strane: dok se do sada operativni sistem našeg računara brišu s svim greškama i čuvaju naš program, PEEK i, posebno, POKE su naredbe koje mogu da bude destruktivne. Zato svaki program koji ih koristi treba, pre startovanja, svakako snimiti na traku — to je mera preostrožnosti koja nikada nije usvršena.

Nećemo govoriti o sintaksi i funkcijama ovih naredbi (objašnjene su na samom početku „Malih tajni velikih majstora programiranja“ u „Računari-ma“ 1.), ali moramo da kažemo nekoliko reči o specifičnostima Spectrume memorijeske mapе. Treba, naime, da nademo memorijeske prostore u kojima možemo slobodno da smestamo sadržaj ne plašći se da će naš program biti oštećen. U nekim programima čete naći na korišćenje REM linija u tu svrhu što je direktni znak da su autori tih programa, pre Spectruma, imali ZX81. Iako vam ne bismo mnogo preporučili ovaj način, rečemo nekoliko reči o njemu pre nego što predemo na naredbu CLEAR.

Obrnište program koji se nalazi u raču-

naru i otkucajte prvu naredbu i REM AAA a zatim POKE 23760, 127. Listanje programa ce otkriti da se prvo slovo A pretvorilo u znak za kopiraj!! Pogled na tabelu ASCII karaktera otkriva da je 127 kod log znaka, sto znači da smo pronašli mesto na koju je smešteno slovo A i promenili ga pomoću POKE! Adresa 23760 odgovara prvom karakteru iz REM naredbe (podrazumevamo se da ispred REM-a nema drugih naredbi). Ne moramo da se ogranicimo na jedan karakter: prva REM linija može da ima mnogo slova i tako postući kao akumulator za čitav mašinski program ili tabelu podataka.

Zašto smo rekli da ne preporučujemo ovaj metod? Kada jednom počnete da listate program, pre linije će biti puni svakakvih karaktera. Rešenje ovog „estetskog“ problema bi moglo da bude smeštanje mašinaca u poslednje linije programa, ali je njih mnogo teže pronaći u memoriji računara i, što je još gore, njihova početna adresa se menja svaki put kada se program produži ili skraći.

Najnadostavljeniji način da rezervišeš prostor u memoriji je naredba CLEAR. Za Spectrum 16 Kb CLEAR n ce rezervisati prostor od adrese (n+1) do 32599 za mašinski program i podatke: ukoliko je u veću od 32600, neki i svii karakteri koje ste definisali mogu da budu izgubljeni. Sve ovo se odnosi i na Spectrum 48 Kb, s tim što su sve cifre veće za 32768. U memoriskom prostoru rezervisanu na ovaj način cete, za sada, smeštati same sadržaje delova ekranu koje treba brzo i dinamično razmenjivati da bi se dobila animacija, ali cete ubroj naučiti da je mnogo rentabilnije izbeći korišćenje matrica i nizova kada su svi njihovi elementi između 0 i 255. Takve brojeve je bolje smeštati u rezervisani deo memorije, pri čemu svaki od njih zauzima po jedan bajt. Vreme utrošeno za pisanje programa koji će, po potrebi, prečaravati poziciju nekog bajta i pozivati ga će se višestruko isplati — najpre z bog utroška memorije, a zatim, što je još važnije, z bog mnogo bržeg izvršavanja programa.

Razbijanje nizova na bajtove je samo jedna (i to marginalna) upotreba naredbe PEEK i POKE. Mnogo je zanimljivije koristiti ih za promenu sadržaja sistemskih promenljivih. Govorimo samo o onima koje su od posebnog značaja za akcione igre.

Već smo videli da smeštanje broja različitog od 1 u memorisku celiju 23692 rezultuje da ispisuje Scroll? kada se ekran popuni. Slično tome, POKE 23609,100 (probajte i druge brojeve umesto 100!) daje kratak ton po pritisku svakog tastera umesto uobičajenog klika, što je zgodno u toku INPUT-a u nekoj igri.

Promena sadržaja memoriskje celije 23624 utiče na boju kojom se poruke ispisuju na ekranu. Ova memoriska celija, zapravo, čuće atribut slova ovih poruka na način koji smo upoznali kod naredbe ATTR. Slično tome, memoriska celija 23693 je „odgovorna“ za sadržaj viših redova ekranu i attribute tog sadržaja, ali se koristi dosta teže: pokusajte, na primer, da izvršite POKE 23659,1 a zatim i CLS.

Celije 23677 i 23678 su korisne za one koji koriste naredbu DRAW koja je, kao što znamo, „relativna“ — pomeranje se obavlja u zavisnosti od položaja poslednje „pome-

9 REM Inicijalizacija
10 GO SUB 5000
11 REM Stanjanje lopte, cuvanje starih vrednosti
28 PLOT x,y: LET z=xx: LET bwy
30 LET y=yy: LET bwy
39 REM Provera položaja lopte
48 IF y<6 THEN GO SUB 2000
50 IF x<6 OR x>248 THEN GO SUB
3000

58 IF y>100 THEN GO SUB 4000
69 REM Pomeranje reketka
70 PLOT p,6: DRAW 20,8
80 LET p=p+(5 AND INKEY\$="8" AND
p>235)-(5 AND INKEY\$="5" AND
p<235)

90 PLOT p,6: DRAW 20,8
99 REM Obrisni loptu
100 PLOT a,b
110 IF true THEN GO TO 20
119 REM Lopta prouzila reket
120 LET ball=ball-1: LET z=z-1

130 PRINT RT 0,10-LEN STR\$ z: 0
VER 0,\$
139 REM Da li je kraj?
140 BEEP 1,0: IF ball=0 THEN GO
TO 8000
199 REM Priprema za sledeću
loptu

200 PLOT p,6: DRAW 20,8
210 GO SUB 5000
220 GO TO 20

1999 REM Posodjeni reket
2000 LET dx=RND\$*SGN dx#+SGN dx#
2

2010 LET dy=RND\$*4+2
2020 LET true=x>p AND x<p+21
2060 LET y=?

2070 LET ss=z+1: PRINT RT 0,10-LE
N STR\$ z: OVER 0,\$

2100 BEEP .01,20
2200 RETURN
2999 REM Posodjeni zid

3000 LET dx=-RND\$*SGN dx#+SGN d
x#2

3010 LET dy=RND\$*SGN dy#+SGN dy#
2

3020 LET x=6+(242 AND x>248)

3100 BEEP .01,8

3200 RETURN
3999 REM Posodjeni vrh

4000 LET dy=RND\$*4-2
4010 LET dx=RND\$*SGN dx#+SGN dx#
2

4020 PLOT 6,h: DRAW 243,0

4030 LET h=h-2 AND h>80

4040 PLOT 6,h: DRAW 243,0

4050 LET y=h-1

4100 BEEP .01,40

4200 RETURN
4999 REM Potprogram za

inicijalizaciju

5000 LET ball=3: LET z=0

5010 LET h=165

5100 BORDER 4: PAPER 1: INK 7: C
LS

5300 LET x=127: LET y=h-1

5310 LET dx=RND\$*10-5: LET dy=-(R
ND\$*3+2)

5320 LET p=115

5330 LET true=1

5600 GO SUB 6000

5700 RETURN
5999 REM Crtanje ekrana

6000 OVER 0

6010 PRINT RT 0,1;"SKOR=000

0\$TRDLO LOPTI 000"

6020 PRINT RT 0,10-LEN STR\$ z:\$

PRINT AT 0,31-ball: FLASH 1;"0"

6100 PLOT 5,0: DRAW 0,167: DRAW

245,8: DRAW 0,-167

6150 PLOT p,6: DRAW 20,8

6200 OVER 1

6400 RETURN
7999 REM *** KRAJ IGRE ***

8000 PRINT RT 10,11; FLASH 1;"KR
AI IGRE"

8010 BEEP 1,-12: BEEP 2,-24

8020 OVER 0: BORDER 7: INK 9

8030 STOP

5 REM prostor za "masinac"

10 CLEAR 31999

11 REM 64999 umesto 31999

za 48K

20 GO SUB 8900: REM "masinac"

8900 RESTORE 8100: FOR i=32000 T

0 32645: READ a: POKE i,a: NEXT

i: RETURN

8009

8010 REM SCROLL levo

8019 REM RANDOMIZE USR 32000-

8020 REM 65000 umesto 32000

za 48K

8040 REM SCROLL desno

8049 REM RANDOMIZE USR 32023-

8050 REM 65023 umesto 32023

za 48K

8100 DATA 1,255,2,17,0,88,33,1,8

8,237,176,1,0,24,17,0,64,33,1,64

,237,176,281

8118 DATA 1,255,2,17,255,90,33,2

54,90,237,184,1,0,24,17,255,87,3

3,254,87,237,184,201

SLIKA 11

1 REM Za 16K SPECTRUM

2 CLEAR 32592: REM

(65380 za 48K)

3 LET num chars=1

4 LET bytes reserve=num chars

#8

5 LET n=32595-bytes reserve:

REM (65367 za 48K)

6 LET char before first=CODE

"#"-32

7 LET byte before first=char

before first#8

8 LET char=n+1-byte before fi

rst#256

9 LET chr in 23606=char-256#I

NT (char/256)

10 LET chr in 23607=INT (char/

256)

20 FOR i=1+1 TO n+bytes reserv

e: READ a: POKE i,a: NEXT i

30 DATA 255,36,36,24,24,36,36,

255,0

50 POKE 23606,chr in 23606

60 POKE 23607,chr in 23607

70 PRINT RT 11,15;"#"

80 POKE 23606,0

90 POKE 23607,50

nute“ tačke. Ako, na primer, najpre izvršimo PLOT 100,80 a zatim DRAW 20,60, liniju koju dobijemo se neće završavati u tački 20,60 nego u 120,140! Promeniti sistemski promenljivih 23677 i 23678 možemo da simuliramo apsolutni DRAW. U prethodnom primjeru bilo je dovoljno zamjeniti naredbu DRAW 20,60 naredbom: DRAW 120-PEEK 23677, 140-PEEK 23678 ili, alternativno, zamjeniti ovo sa dve naredbe, od kojih bi jedna izračunavala druga crtala, što je način koji će početnici uvek radije primeniti.

Peščani sat

U mnogim igrama je potrebno meriti vreme koje je igrač bio potreban da obavi neki zadatak. Poneko to radi tako što u pravilnim intervalima, dodaje 1 nekoliko promenljivoj, ali se ubroj pokazuje da je ovakav sat više nego netočan, što izaziva ne malo začepašenje. U čemu je stvar? Vaš Spectrum ne radi neprekidno posao koji ste mu zadali; vrlo često (50 puta u sekundi) on

```

10 REM **** RELI ****
9998 REM Definisanje karaktera
9999
9000 CLEAR 32399: RESTORE 9200
9020 PRINT RT 11,18: FLASH 1;"SR
CEKRU"
9030 FOR i=32400 TO 32599
9040 READ a: POKE i,a: NEXT i
9050 FOR i=1 TO 18: FOR n=0 TO 7
9060 READ a: POKE USR CHR$(i+14
3+i,n): NEXT n: NEXT i
9070 CLS : PRINT RT 9,7;"STRATUJ
TRKU"
9090 LOAD ""
9199 REM Znaci za definisanje
karaktera
9200 DATA 0,24,36,36,126,102,102
102
9210 DATA 0,124,68,68,126,70,70,
126
9220 DATA 0,124,64,64,96,96,96,1
26
9230 DATA 0,126,66,66,98,98,98,1
26
9240 DATA 0,126,64,64,128,96,96,
126
9250 DATA 255,129,189,165,165,18
9,129,255
9260 DATA 0,126,64,64,96,102,98,
126
9270 DATA 0,68,68,68,124,76,76,7
6
9280 DATA 0,16,16,16,24,24,24,24
9290 DATA 284,284,51,51,284,284,
51,51
9300 DATA 0,0,0,0,0,0,0,0
9310 DATA 0,32,32,32,48,48,48,62
9320 DATA 0,126,86,86,86,86,86,8
6
9330 DATA 0,124,76,76,76,76,76,7
6
9340 DATA 0,126,66,66,98,98,98,1
26
9350 DATA 0,124,68,68,124,64,64,
64
9360 DATA 0,0,0,0,0,0,0,0
9370 DATA 0,124,66,68,126,70,70,
78
9380 DATA 0,124,64,64,124,12,12,
124
9390 DATA 0,126,16,16,24,24,24,2
4
9400 DATA 0,68,68,68,68,68,68,12
4
9410 DATA 0,126,66,2,39,16,0,16
9420 DATA 0,86,86,86,86,86,86,12
6
9430 DATA 0,24,24,6,8,24,24,0
9440 DATA 0,68,68,124,16,16,16,1
4
9499 REM Podaci za "kola"
9500 DATA 8,73,73,93,127,93,65,6
5
9510 DATA 33,66,188,60,60,29,2,4
9520 DATA 254,16,56,63,56,16,254
0
9530 DATA 4,2,29,60,68,188,66,33
9540 DATA 65,65,93,127,93,73,73,
8
9550 DATA 32,64,184,68,68,61,66,
132
9560 DATA 127,8,28,252,28,8,127,
0
9570 DATA 132,66,61,60,60,184,64
22
9593 REM Podaci za brojeve
9600 DATA 60,66,66,98,98,98,98,68
9610 DATA 0,8,8,12,12,12,12,12
9620 DATA 0,124,2,2,68,96,95,62
9630 DATA 0,124,2,2,68,6,6,126
9640 DATA 0,64,36,108,108,126,4,
4
9650 DATA 8,62,64,64,68,6,6,126
9660 DATA 8,52,64,64,124,70,70,1
26
9670 DATA 0,128,8,8,12,12,12,12
9680 DATA 0,60,66,66,60,60,70,70,60
9690 DATA 0,68,65,65,50,6,6,126
9999 POKE 23606,0: POKE 23607,0
: RETURN
5 REM **** RELI II ****
10 GO TO 3000
100 LET al=IN 64510-255: LET d1
=IN 65022-255: LET mv=IN 57342
120 LET dr=dmdr+(mv-255)-(8 AND
dr=8 AND mv=255)
140 LET x=-dr:0: dr=1:0: dr
=?>(dr>2) AND (dr<6): LET y=-(dr
)>4 AND dr=?>(dr<4)
160 PRINT RT x,y;CHR$(144+dr)
180 IF al OR dr THEN PLOT 4+*#8
,56: DRAW 0,5: LET a=-(al AND a
)>1 AND a1: PLOT 4+*#8,56:
DRAW 0,5
200 FOR i=1 TO 5-a: BEEP .02,RN
2010-2040: NEXT i
220 PRINT RT x,y;CHR$(144+dr)
240 GO TO LOOP<400 AND ATTR <x
,y>57>
500 GO TO END
997
998 REM Alternative:
999
1000 POKE 23606,136: POKE 23607,
123: RETURN
1497
1498 REM Normalno
1499
1500 POKE 23606,0: POKE 23607,0
: RETURN
1997
1998 REM Stampanje vremena
1999
2000 GO SUB ALTER
2010 LET s=<0> AND <10>*STR$-
m<0> AND <10>*STR$=
2020 PRINT RT s,0: LET z=3:CHR$(C
ODE s*>104):CHR$(CODE s*>2)*
104)
2030 PRINT RT a,m+3:CHR$(CODE s
*>3+104):CHR$(CODE s*>4)+104)
2040 GO SUB NORM
2050 RETURN
2051
4997
4998 REM ***INICIJALIZACIJA***#
4999
5000 LET n=1E39
5010 LET LOOP=100
5020 LET ALTER=1000: LET NORM=15
5030
5030 LET TIME=2000
5040 LET START=S3000: LET NEXT CR
R=5400
5050 LET NEW GAME=9000: LET BYE=
9000
5060 LET END=7000
5070 LET CRASH=7500: LET FINISH=
9000
5080
5100 POKE 23693,33: POKE 23659,1
: CLS : INPUT "
5110 GO SUB ALTER
5120 DIM b$(2,5): LET b$(1)=CHR$(1
6+CHR$ 2+CHR$ 17+CHR$ 6+CHR$ 1
02: LET b$(2)=CHR$(16+CHR$ 1+CHR$ 1
7+CHR$ 7+CHR$ 107)
5130 BRIGHT 0: OVER 0: FLASH 0:
INVERSE 0: BORDER 5: PAPER 7: IN
K 1
5140 PRINT RT 0,0: RESTORE 6500:
LET n=1
5150 FOR i=1 TO 189: READ a: LET
n=n-nHOT n: FOR v=1 TO 18: PRIN
T b$(n+1):: NEXT v: NEXT i
5160 PRINT RT 19,19: INK 0;"J"; IN
K 2,10;"J"; INK 1,2; INK 3;"PRV
Okvvvv"; INK 6,5;"VREME"; INK 1,14;
INK 1;"SPECTRUM*SPEEDWAY"; INK 13
,1;"oar"
5170 FOR i=1 TO 4: PLOT 4+*#8,62
: NEXT i: INK 8
5180 GO SUB ALTER: PRINT RT 8,6:
INK 3;"IIxII"
5190 LET c=3: FOR i=1 TO c: PRIN
T RT 16+i,1;"C": NEXT i: PLOT 8,
63: DRAW 31,0
5200 PRINT #0: INK 2: PAPER 6;"f
fffffSPECTRUM*SPEEDWAYfffff"
5330 LET n=0: LET s=0
5399 REM Sledeci kola
5400 LET x=20: LET y=10: LET dr=
6: LET g=1
5410 PLOT 12,66: DRAW 0,5
5420 OVER 1: GO SUB NORM
5430 POKE 23672,0: POKE 23673,0:
BEEP 1,0
5490 GO TO LOOP
6499 REM Podaci za ekran
6500 DATA 34,10,2,17,96,11,1,5,3
,4,7,12
6510 DATA 1,2,2,5,1,2,6,4,7,2,1,
2,8,2,6,3,2,5,1,2,1
6520 DATA 2,5,2,5,2,5,2,9,2,1,2,
1,3,1,2,2,3,4,2,3,5
6530 DATA 1,2,1,2,1,3,1,2,3,2,4
6540 DATA 8,3,2,1,2,2,2,2,2,3,2,
4,8,1
6550 DATA 2,3,2,1,3,1,3,2,2,8,4,
1,2,3,2,1,3,2,2
6560 DATA 2,2,3,4,3,2,1,3,1,3,2,
2,2,2,2,2,3
6570 DATA 4,2,3,1,6,2,3,2,2,2,3,
7,3,3,4,2,4
6580 DATA 1,3,2,3,6,3,9,4,2,3,3,
2,1,2,3,2
6590 DATA 3,3,3,4,2,2,1,2,3,7,4,
3,4,5,2,2,1,2,13,4,5,1,4,6,11,6,
8,34
6997
7000 PLOT 4+*#8,56: DRAW 0,5
7010 OVER 0: LET s=>INT ((PEEK
23634/256)+PEEK 23672)>50)
7020 LET n=n+INT ((s/68): LET s=s
-68: LET a#=0: LET b#=6: GO SUB
NORM
7030 GO TO CRASH<FINISH-CRASH R
HD ATTR <x,y>56>
7497
7498 REM *****CRASH*****#
7499
7500 FOR i=0 TO 25: LET dr=dr+1-
(8 AND dr=7): PRINT RT x,y;CHR$(1
44+dr): BEEP .02,1: NEXT i
7510 GO SUB ALTER: PRINT RT x,y;
b#1: GO SUB NORM
7520 PRINT RT 16+c,1;" " : LET c=
c-1
7530 IF c>0 THEN GO TO NEXT CAR
7540 GO TO NEW GAME
7997
7998 REM *****FINISH*****#
7999
8000 PRINT RT x,y;CHR$(144+dr):
LET t=60#n+#
8010 GO SUB ALTER
8020 INPUT "
8030 PRINT AT 1,0:#0;"J|||||J|KBR
R0..-USPELI STEK||||||": PRUSE 2
80
8040 INPUT "
8050 IF t<0 THEN LET h#=1: LET a#
1: LET b#=7: GO SUB TIME: GO SUB
ALTER: PRINT RT 1,0:#0;"FFFFFKI T
0 JE NAJBOLJE VREMEFFFF": GO TO
8070
8060 PRINT RT 1,0:#0;"ALI VREME
NIJE NAJBOLJE"
8070 PAUSE 200
8080 PRINT RT x,y;"J"
8090 GO SUB NORM
8097
8098 REM ***NOVRA IGRA***#
8099
8100 INPUT "JOS JEDNOM <D/N>": L
INE z#
8101 IF <z#> "X">=N" THEN GO
TO BYE
8120 INPUT "Pritisni ENTER": LI
NE z#
8130 GO TO START
8180 BORDER 1: PAPER 1: INK 7: C
LS
8190 PRINT RT 10,13: FLASH 1: IN
K 6: PAPER 2;"ZDRAY000"
8199 STOP

```



dobija takozvani interrupt koji mu nalaže da prede na izvršavanje jednog programa u ROM-u koji će, pre nego što računar nastavi sa radom, obaviti neku vrstu "generalnog spremanja". Bilo kakav vremenska petlja koju stavite u program će, dakle, biti često prekidana i taj broj prekida nećeće moći lako da izračunate.

Zato je za merenje vremena najbolje iskoristiti sistemski časovnik koji računar izvjećava za jedan u toku svakog interrupta (to je jedna od komponenti pomenutog "generalnog spremanja"). Za upotrebu ovog časovnika treba upoznati memorije, čelije, cije su adrese 23674, 23673 i 23672. Svake pedesetinice sekunde sadržaj čelije 23672 biva uvećan za jedan. Kada se on poveća na 255, računar povećava sadržaj čelije 23673 dok se nula smješta u 23672. Slična operacija se ponavlja i kada sadržaj 23673 postane 255, samo što se tada sadržaj 23674 povećava za jedan. Sazivim je, prema tome, prosto izračunato vreme u sekundama od kada su sva tri ova brojčića bila nula (možete ih sami postaviti na nulu pomoću POKE na početku igre): LET sekundi = (65536*PEEK 23674+256 *PEEK 23673+PEEK 23672)/50.

Na žalost, Spectrumom sat je daleko od savršenstva. Pre svega, vreme biva izgubljeno svaki put kada se izvrši BEEP (do sada je ponovno inicijalizovanje nekih sistemskih promjenljivih posle BEEP bilo od korist), ili kada se koristi neka periferijska jedinica (npr. štampač). Pored toga, ponekad izračunato vreme može da bude netočno: možda se sadržaj 23674 nije bio promenio dok je računar izvršavao prvo PEEK. Zato se u on povećan i sadržaj 23673 anulirat pre nego što je izvršeno drugo PEEK. Zato se uputstvu za upotrebu savetuje da se uvek izračunaju dva broja sekundi, pa da se već uzmre za tačan.

Tenk na reliju

Jeste li znali da možete da redefinišete više od 21 karaktera (čak do 96) i tako povećate raznovršnost vaše flote napadača iz svermija? Evo kako se to radi na najlakši mogući način:

Najpre iscrtajte sve nove karaktere i izračunate dekadne vrednosti njihovih linija.

— Odlučite koji originalni karakter treba da budu predefinisani i na koji način.

— Upotrebite CLEAR da rezervišete prostor za definicije. Za svaki karakter koji treba definisati rezervište osam bajta

— Prebrojte koliko ima karaktera prenoga koji treba redefinisati (ako, na primer, redefinišete znak „.„ koji ima kod 42, čete da pre njega ima 42—32=10 karaktera). Zatim pomožite taj broj sa 8 izvršite naredbe:

POKE 23606, m-256 INT (m/256)

POKE 23607, INT (m/256)

— Okončajte definiciju karaktera na uobičajeni način.

Sve će ovo postati mnogo jasnije kada pogledate primer koji predefiniše karakter u grafični blok.

Gotovo sve što smo do sada prešli sadržano je u igri „Relli“. Treba, naravno, voziti automobil po kružnoj stazi. Automobil, doduše, više liči na tenk, ali je to zato što je potrebno da se na neki način vidi kuda se krećemo iz oblika pokretnog predmeta: tenk je za to neobično pogodan zborog „cevi“. Ukoliko vam se ovaj oblik ne sviđa, ili ste miroljubivo raspoloženi, možete da ga promenite bez ikavkih problema.

Program koristi IN (naredbu opisanu na početku teksta) da omogući prilisak dva tastera u isto vreme, na primer lastere za ubrizgaj i promenu pravca. Automobil, nai-m, ima četiri brzine. Pritisak na O vas ubacuje u sledeću, a pritisak na A u prethodnu brzinu. Automobil se kreće u svim

pravcima; taster P je namenjen skretanju u pravcu kazaljke na časovniku, a O skretanju u obrnutom pravcu.

U igri je korišćeno merenje vremena, ali se vreme prikazuje tek kada nastradate ili kada obidete čitav krug — igra bi inače bila prespora. Za ispunjenje zadatka imate tri automobila: svaki put kada stradate, računar vas vraca na početak, što je dodatno otežanje.

Glavna karakteristika igre je predefini-sani set karaktera: praktično su sva slova i brojevi predefinisani uz korišćenje ideja iz prethodnog poglavljija. To je, na žalost, učinilo program predugačkim, pa je morao da bude podešen u dva dela: prvi definisće karaktere a zatim učitava drugi koji predstavlja pravu igru. Ovakva podela programa, jasno, ne odgovara vlasnicima Spectruma 48 Kb koji mogu, uz malo veštine, ujediniti programme u jednu celinu.

Ukoliko kucate program iz dva dela, otkačajte najpre prvi, snimite ga sa SAVE „Relli“ LINE 10, verifikujte snimak i zatim otkačajte drugi deo programa i snimite ga sa SAVE „C“ line 10.

Sedam zlatnih pravila

Ovim smo se približili kraju ove škole akcijskih igara na Spectrumu. Ipak, pre nego što završimo poslednji čas i pozovemo vas da sami prionte na posao, nekoliko saveta koji će nadamo se, pomoći da vaše buduće video-igre budu dinamičnije.

Što više humoru u komentariima koje računar ispisuje. Neka korisnik programa vidi da igru ne treba shvatiti baš previše ozbiljno. Ipak, treba paziti da se še sačuvaju simpatiju prema kompjuteru.

Ukoliko se pojavljuje situacija u kojoj je kompjuter potrebovao duže vreme da se pripremi za sledeće akcije (npr. da „zamisliti“ labyrin na početku igre), pokušajte da stvorite utisak da kompjuter razmišlja. Neka se, na primer, na ekranu s vremenom na vreme smenjuju komentari u stilu „Samo što nisam sprem“... „Hmmm... nije baš lako“ i slično. Ovi komentari nipošto ne smiju da budu isti pred svaku partiju, jer će postati dosadni. Pripremite više poruka nego što je potrebno, pa ih slučajno birajte pomoću RND.

Uvek koristite karaktere koje ste sami definisali umesto zvezdica ili nečeg sličnog. Sigurni smo da igraču neće ni malo smetati što je bomba velika koliko i avion ili Mesec koliko i Sunce ako su istaknuti bitni detalji iako se na prvi pogled vidi šta se dešava.

Eksperimentište što više sa bojama; koristite čak slučajne boje koje, ipak, moraju da budu kontrolisane u izvesnoj meri (ako PAPER i INK budu isto obojeni, neće se, videti šta se dešava). Promena boje pozadine neverovatno menja utisak o prizoru, a „košta“ samo jednu kratku naredbu.

Eksplozije su neobično efektne i ne treba štedeti trud na njihovu realizaciju. Treba samo, paziti da eksplozije ne traju predugo jer one naješće blokiraju sve druge događaje.

Pošto ste završili igru, utrošite još pola sata da smislite nešto što će biti originalno. Predložili bismo da uvedete neki događaj koji je izuzetno malo verovatno i dešava se na primer, jednom u 200. partiju. Za korisnika će biti izuzetno prijatno iznenadjenje da otkrije da igra koju mu je možda skoro dosadila ima još malih tajni.

Iako ste završili igru, utrošite još pola sata da smislite nešto što će biti originalno. Predložili bismo da uvedete neki događaj koji je izuzetno malo verovatno i dešava se na primer, jednom u 200. partiju. Za korisnika će biti izuzetno prijatno iznenadjenje da otkrije da igra koju mu je možda skoro dosadila ima još malih tajni.

Na mački jezik!

Dejan Ristanović

Servis računara

„Galaktičari“ u nevolji

Pre nego što vas obuzme panika, provode još jednom štampane veze od video izlaza preko integralnih kola 74LS38, 74LS00, 74LS123, CD 4017, CD 4040 do 74LS93 i 74LS04. Uključite ponovo računar i pokušajte da na ekranu registrujete bar neki signal. Možda će to biti slika bez ikakvih znakova (slova ili brojeva), ali sa karakterističnim rasterom. Ako se i sada na ekranu ništa ne pojavljuje, ostaje vam mogućnost „hladnog lema“ (postoji vizuelni ali ne i električni kontakt) na jednom od gore poimenutih integrisanih kola. Zagrijte lemljicu i osvežite lemove na ovim kolima. Ako i to ne pomogne, biće vam potrebna logička sonda da medu ovim integralicama pronađete krvica.

Lociranje i otklanjanje greške je mnogo jednostavnije ako se na ekranu pojavi bar nekakva slika. Za brzo lociranje kvara potrošiće vam tabelu 1, sa pregledom integrisanih kola u čijoj okolini treba tražiti kvar.

Pročitajte još jednom

Ako još uvek niste dobili tastaturu ili kompletirali sve delove, pročitajte sledeće savete pre početka rada:

- dok ne „ozivite“ računar, dovoljno je da ugradite samo jednu memoriju 6116; kada računar proradi, lako ćete ugraditi i ostale, a ako ste u nečemu drastično pogrešili, bar ćete sačuvati skupu memoriju.
- koristite RF modulator firme ASTEC UM 1111 E 36; ovaj se modulator napaja sa 12 volti, a raspored nožiće je isti kao na štampanoj placi; R16 zamjeni kratkospojnikom, a otpornik R15 izostavite; za druge tipove modulatora potrebno je eksperimentalno odrediti vrednosti ovih otpornika.
- vaš računar može, u nevolji da radi i bez kondenzatora za blokadu Cn; ako ih još niste nabavili, probajte da „ozivite“ svoj računar i bez njih

- posebnu pažnju obratite lemljenju najveći broj kvarova potiče upravo odatle; prilikom lemljenja integrisanih kola, napravite pauzu posle svake nožiće dok se kolo ne ohladi. kada se na njemu pojave nečistoće, vrh lemljice čistite vlažnom kromom
- masku za tastaturu pre upotrebe treba malo obraditi; ivice svih otvora za tastere malo sastružite turpijom i prvo postavite tastare na uglovima, a zatim po jedan u sredini svakog reda.

Ako ste pažljivo sklapali svoj računar i ako su vam sve komponente ispravne, onda ćete prilikom prvog uključivanja imati sliku na ekranu sa dugo željenim READY. Međutim, može se desiti da je slika iskrivljena i

SIMPTOM	MOGUĆI UZROK
1. Ekran bez teksta i rastera	80A, 6116, 2732, 74LS04, 74LS93, 74LS74, 74LS123
2. Raster postoji ali nema teksta	2716, 74LS04, 74LS74, 74LS156
3. Slika jako iskošena	C3, C4, R12, R13
4. Ne dekoduju se određeni karakteri	74LS156, 74LS251
5. Karakteri se ispisuju ali nedostaju neki njihovi delovi ili su deformisani	Z80A, 2732, 6116, 74LS174
6. Paralelno se ispisuju prvi i peti red istim karakterima	74LS174
7. Svi se karakteri ispisuju ali neki delovi nedostaju	74LS166
8. Slova presećena po vertikali a na BRK se javljaju crte na desnoj strani	74LS174
9. Nevezani karakteri na raznim delovima ekrana	Z80A, 74LS32, 74LS156
10. Pirinčani raster po celom ekranu	Z80A, 4LS156
11. Primenjivji raster po celom ekranu	Z80A, 74LS156
12. Dijagonalni raster sa jednom vertikalnom štrafatom	Z80A, 74LS156
13. Nevezani tekst, ne prima komande	74LS38, 4LS74
14. Vertikalne slike bez teksta	74LS04, 74LS38
15. Crte po celom ekranu	74LS38
16. Pored READY povremeno se javlja nevezani tekst	74LS38
17. READY na desnoj polovini ekrana zatim karakteri po celom ekranu	Z80A, 74LS74
18. Slika se kreće po vertikali	74LS04, 74LS123
19. Slika jako iskošena i kreće se po vertikali	C3, R12, 74LS123
20. Kosi tačkasti karakteri	74LS123
21. Par belih kosih crta	74LS123
22. Vertikalno postavljeni ili iskošeni karakteri	74LS93, CD4040
23. Vertikalno kretanje slike bez teksta	74LS93
24. Na PRINT MEM pokazuje manje memorije	6116, 74LS156

da se slova pojavljuju u više iskošenih redova. Pokušajte prvo da povecate kondenzator C3 na oko 10 nF do 30 nF i otpornik R12 na 1 do 2 koma. Slika će poticati da se ispravlja. Početak reda možete pomerati smanjivanjem kondenzatora C4 do 50 nF i R1 do 15 koma.

Nepotuni karakteri

Umirili ste sliku i sada proveravate da li se svako slovo koje otkucate pojavljuje na ekranu. Ako neki karakter ne reaguje, odmah proverite njegovog suseda sa vertikalne i horizontalne linije na shemi. Ako ne reaguje čitav red, proverite veze ka kolima 74LS251 i 74LS156. Možda je ta linija u prekidu, ili nožica kola nije dobro zategnjena. Posle ove intervencije, tastatura će reagovati, ili je neispravno kolo ka kome vodi linija koja ne reaguje na komande sa tastature.

Na ekranu se mogu pojaviti i nepotupni karakteri (recimo, slovo A bez horizontalne crte po sredini). Grešku treba tražiti na nožicama 4 i 5 kola 74LS174 ili na nožicama 2, 3, 4, 5, 10, 11, 12, i 14 kola 74LS166, tj. na njihovom putu ka epromu 2716. U slučaju da se slika dvostruka, proverite

nožice 6 i 7 kola 74LS174, kao i kola koje imaju ovu funkciju. Loše zategnjene nožice 2 i 3 ovog kola daju deformisane i neprepoznatljive karaktere, a pritisakom na tastu BRK na desnoj strani ekrana pojavljuju se vertikalne linije. Ako su karakteri presećeni po horizontali, proverite nožice 10 i 11, a ako se karakteri ispisuju istovremeno u prvom i petom redu, odgovorne su nožice 14 i 15 kola 74LS174.

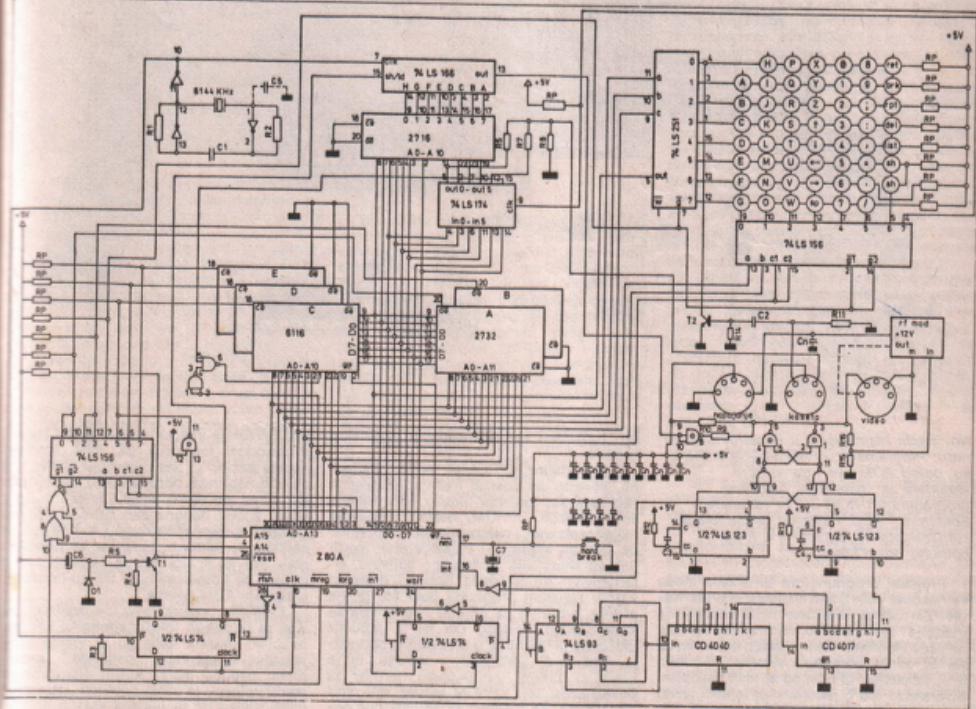
Neispravno kolo 74LS32 ili 74LS156 (u adresnom dekoderu) prouzrokuje sliku pirinčanog rastera ili će se na raznim delovima ekrana pojaviti nevezani karakteri koji mogu biti i nestabilni. Ako se pojavi dijagonalni raster sa jednom vertikalnom štrafatom, nožica 15 kola 74LS156 ima kratak spoj sa masom. Isto se simptomi javljaju i kad je neispravan mikroprocesor Z80A.

Nevezani tekst

Nevezani karakteri, ali sa ispisanim READY na početku prvog reda, javljaju se ako nožica 11 kola 74LS38 ima spoj sa masom. Ovaj deo kola, kada je ono neispravno, prouzrokuje vertikalne slike po ekranu ili nevezani tekst. Potpuno jednako

Najzad je sve gotovo! Posle svih muka sa nabavkom delova, čekanjem tastature i čipova, vaš računar je, najzad, sklopljen. Uključi ste ispaljač, povezali kable i počeli da pretražujete frekventne opsege na svom prijemniku, a na ekranu se ne pojavljuje ama baš ništa. Baš nemate sreću! Ne treba, medutim, gubiti glavu — to nije najgora stvar koja je mogla da vam se desi. Do sada je, osim toga, proradio oko dve hiljade „galaksija“, mnoge i u znatno neveštijim rukama nego što su vaše. Zašto biste baš vi bili izuzetak?

Računar „galaksija“ odlikuje visoka pouzdanost u radu — do sada se redakciji javilo samo nekoliko čitalaca, u svakom slučaju manje od deset koji nisu mogli sami da izadu na kraj sa svojim računaron, tako da praktično nemamo nikakvih serviserskih iskustava. Zato smo jedan od redakcijskih primeraka stavili „na muke“, izazivajući na njemu, naravno po određenom sistemu, sve kvarove koji su mogli da nam padnu na pamet. Računar je ovu torturu glatko preživeo, a rezultati su sumirani u tabeli i u ovom tekstu. Ako vam ni on ne pomogne da se izvučete iz nevolje, javite se redakciji. Vaš računar, budite sigurni u to, neće ostati samo gomila skupljih čipova.



Električna shema računara „galaksija“: ne shemi nema nikakvih izmena — sedno su obeleženi izvodi na integrisanim polima, što znatno olakšava fraganje za servisima.

se manifestuje prekid signala za osvežavanje memorije (koji inače ovde ima sasvim drugu ulogu), nožica 28 mikroprocesora, ako je neispravan Z80A, nožice 3 i 4 kola 74LS04 ili nožica 13 kola 74LS74.

Može se desiti da, neposredno posle uključivanja računara, na desnom delu ekrana dobijete READY, a da se odmah zatim pojave nevezani karakteri po celom ekranu. Grešku potražite na nožicama 2, 3 ili 5 kolica 74LS74 ili na nožicama 20, 24 ili 27 mikroprocesora.

Vertikalno kretanje slike izaziva neku nepravilnost u lancu deliteљa učestanosti kristala — kola 74LS93, CD4040, CD 4017 ili sam monostabilni multivibrator 74LS123. Tačne učestanosti signala na ovim kolima ne možete izmeriti bez specijalnih instrumenata (osiloskopom ili frekvencmetra), pa vam jedino preostaje da kola zamenite novim.

Kad računar potpuno proradi, ugradite preostale memorije 6116 i oktucajte PRIN MEM. Ako računar pokaza da nemate onoliko memorije koliko ste ugradili integrisanih kola, proverite nožice 4 i 5 kola 74LS 156 ili promenite mesta memorijama i opet ištessiterajte kapacitet memorije.

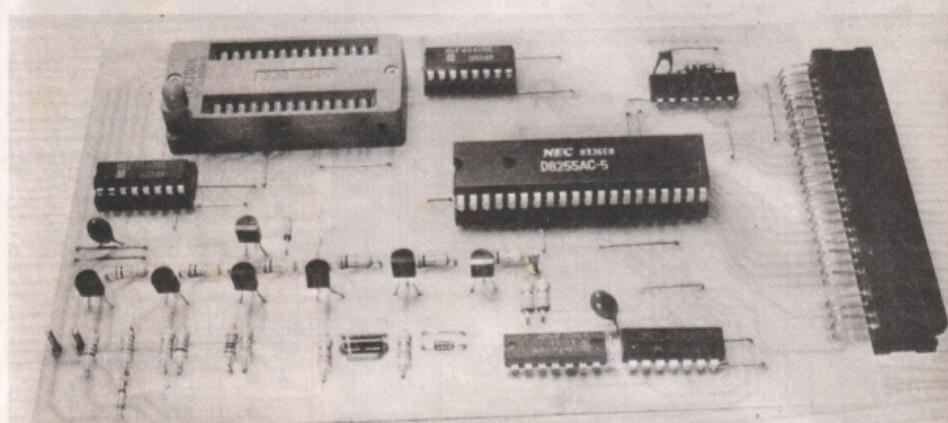
Sve navedene kvarove mnogo lakše ćete otkloniti ako imate mogućnosti da koristite diretan ideo ulaz u svoj televizor, ili vam je na raspolaganju monitor. U tom slučaju izbegavate greške koje unosi RF modulator.

Tekst: Miča Hadži Đorđević

1881. Miss Hauzi D.

programator eproma

računari
u domaćoj radinosti



Nov način razmišljanja u konstruktorskom radu: Programator EPROM-a, na kome su, pored ROM-ova i generatora karaktera za „galaksiju“, uspešno programirani i ROM-ovi za računare Spectrum i BBC, može da uvede mlade konstruktore i programere u krug potpuno novih ideja.

Projekat programatora EPROM-a može se nekome učiniti kao svojevrsno preterivanje. Zar zaista ne postoji ništa zanimljivije i važnije? Nel! Naravno, za one koji ne žele da sede skrštenim rukom i čekaju šta će im za samogradnju ponuditi najčešće „Računare 3“, „Galaksija“ ili neki od stranih časopisa. Stvaralački rad u računarskoj tehničkoj teško je ostvariv bez jedne ovakve naprave. Mikroprocesor je potpuno bespomoćan bez uputstava za rad koja dobita od ROM-a, a ROM možete imati jedino ako ga sami napravite. Uredaji za povezivanje računara sa periferijskom opremom, interfejsi za procesno upravljanje, višekanalni generatori tonova, uređaji za sintezu glasa i sve ostale računarske sitnice koje život znače rade znatno jednostavnije ako je operativni program u nekom ROM-u nego ako se učitava sa kasete. I ne samo to! Kako je korist od neke nove rutine ili naredbe koja povećava snagu računara ako ona prilikom svake upotrebe mraza iznova da se učitava sa trake? Šta tek da se radi kada se u biblioteci programa nakupi tridesetak novih naredbi.

Bi i funkcija i stotinak korisnih mašinskih rutina?

Programator EPROM-a može, očigledno, korisno da posluži i onim ljubiteljima računara koji nemaju nimalo konstruktorskih ambicija. EPROM, zaista, predstavlja najjednostavniji, najpouzdaniji, najtrajniji, pa, sasvim sigurno, i najlegantniji medijum za smještanje sistemskog softvera, alternativnih programskih jezika i izuzetno važnih uslužnih programa (kao što je, na primer, program za obradu teksta) koji se svakodnevno koriste. On se u računaru uključuje ili direktno, u predviđena podnožja, kao kod „galaksije“ i BBC-ja, ili je dovoljan sasvim jednostavan hardver za povezivanje, tzv. EPROM loader, kao kod Spectruma. Za njegovo aktiviranje nije potreban nikakav dodatni softver (BBC) ili je dovoljno nekoliko prostih mašinskih rutina (Spectrum). Pristup programu je, povrh svega, trenutan i, što je najvažnije, on se nalazi u računaru čim ga uključite ili čim, odredenom naredbom, pozovete potreban program.

Kako radi

Rad sa programima iz EPROM-a je komplikovan čak i onda kada imate disk-jedinicu! Na ovom svetu, mediumu, ništa nije savršeno, pa nisu ni EPROM-i: brišanje i upisivanje podataka u njih se ne vrši u računaru nego u specijalnim uređajima, kao što je ovaj čiju shemu objavljujemo, a čitava procedura, naročito brišanje, može da potraje i nekoliko časova. Pošto se programira jednput a koristi bezbroj puta, ovaj nedosta-

tač EPROM-a ima samo akademski značaj. Proizvođači memorija ga, ipak, shvataju veoma ozbiljno i već navajaju čipove koje se pod napomenu ponušaju kao RAM (može se, dakle, po njima pisati i brišati do misije) a bez napona kad EPROM (čuva podatke beskonačno dugo i to one koji se zateknu u njima u trenutku isključenja). Ipak, sasvim je sigurno da naredne dve godine na komercijalnom tržištu pripadaju klasičnim EPROM-ima.

EPROM-i, sigurni smo, ne predstavljaju više ni izdaleku takvu zagotonku kao pre nekoliko meseci kada su, u vreme preteklostva, u računaru „galaksije“, izazvali pravu uzbunu među našim čitacima. U međuvremenu je svaki graditelj „galaksije“ imao priliku da ih upozna sasvim dobro uveri se da se ne radi o nimalo neznačajnim čipovima. Malo teorije, ipak, neće biti na odmet.

EPROM predstavlja skraćenici engleskog izraza „Electrically Programmable Read Only Memory“ i označava memoriju koja se programira električnim putem i kojoj se podaci mogu samo čitati. Procedura programiranja EPROM-a je gotovo beznajno jednostavna: svaki bajt se upisuje pod „visokim naponom“ TTL impulsom kojeg traje 50 milisekundi. Ovaj „visoki napon“ kreće se između 25 V (kod 2716) i 13 V (kod 27256). Programiranje ne mora da bude uspešno — može se programirati bilo koja memorijska lokacija u bilo koje vremenu — pojedinačno jedna za drugom ili na presek. Jednput programiran, EPROM će sačuvati upisani sadržaj najmanje deset godina — ako se ne izloži sunčevu svjetlosti.

Računar „galaksija“, bez sumnje, predstavlja veliki podsticaj i za programere i za samograditelje računara. On razoboljuje kućni računar kao složen tehnički proizvod i ohrabruje ljubitelje računara da i sami ugrade ponекu cijelu u tu građevinu. Ozbiljniji programerski i konstruktorski rad na razvoju i, naročito, „doradama“ računara (najomiljenija sportska disciplina domaćih hakerâ) ne može se, međutim, ni zamisliti bez alatke koja se naziva programator EPROM-a i koja omogućuje jednostavnu produkciju ROM-ova. Zato smo za treću samogradnju posle „galaksije“ pripremili projekt univerzalnog programatora EPROM-a (adaptacija i dorada sheme objavljene u nemackom časopisu MC) za računare „galaksija“ i ZX Spectrum. Zašto baš za njih? Zato što su to dva najpopularnija računara na domaćem tržištu i zato što u njima ima najveće prostora za dorade. Uredaj programira najznačajnije savremene tipove EPROM-a — 2516, 2716, 2732, 2732A, 2764 i 27128 — i radi pod punom programskom kontrolom korisnika. Pomoći njega se može iskopirati ROM bilo kog kućnog računara — od „galaksije“ do BBC-ja — ili isprogramirati alternativni ROM sa nekim komercijalnim ili svojim sopstvenim programom.

Lična karta EPROM-a

tip	(kapacitet kilobajta)	potrošnja mirno/akt.	brzina odziva
2716	2 Kb	25/100 mA	350 ns
2732A	4 Kb	35/125 mA	200 ns
2764	8 Kb	40/100 mA	200 ns
27128	16 Kb	45/150 mA	200 ns

Intel

sti ili nekom još intenzivnijem izvoru ultra-lijubičastih zraka. Broj ciklusa pisanje (impulzom) — brisanje (ultraljubičasti zraci) praktično je neograđen.

I pored sve jednostavnosti, procedura programiranja i brisanja EPROM-a može manje iškusnim konstruktorima izgledati krajnje opasno: šta se dešava kada se kroz nežnu strukturu čipa propusti 25 V ili kada se ona izloži razornim UV zracima? Očigledno, upravo ono što je potrebno!

Memorijsku celiju u EPROM-u predstavlja MOS tranzistor sa dva gejta: jedan je slobodan za prijem signala, a drugi obložen košulicom silicijum-dioksidom pošto na njega nije vezano, najčešće se naziva plivajući gejt. Silicijum-dioksid je izvanredan izolator — elektroni ne mogu ni da izdu napole iz košulice na plivajućem gejtu niti da uđu unutra. Kada se, međutim, na dren dovede visok napon a na slobodan gejt nizak, tako da potencijalna razlika između njih iznosi 20 do 25 V, elektroni dobjaju dovoljno energije da probiju plavajući gejt. U ovoj „košulici“ ostaju zaboravljeni praktično zauvek — u svakom slučaju najmanje deset godina. Celija sa nanelektrisanim plivajućim gejtom ponaša se kao prazna memorijска lokacija.

Ultraljubičasto zračenje slab izolaciona svojstva silicijum-dioksid. Ako je dovoljno intenzivno, „košulica“ će se toliko „razmekšati“ da će elektroni „iscuriti“ kao da im nikada nije bilo — EPROM će biti obrisan i spremjan za upis novog sadržaja. Celija sa razelektrisanim plivajućim gejtom ponaša se kao prazna memorijска celija. Zato se u praznom EPROM-u u svakoj celiji nalazi logičko „0“, odnosno na svakoj adresnoj lokaciji broj 255 (heksadekadno FF). Prilikom programiranja u EPROM se, dakle, upisuju samo logičke nule, a prilikom brisanja te nule se vraćaju na jedinice.

Ultraljubičasto zračenje slab izolaciona svojstva silicijum-dioksid. Ako je dovoljno intenzivno, „košulica“ će se toliko „razmekšati“ da će elektroni „iscuriti“ kao da im nikada nije bilo — EPROM će biti obrisan i spremjan za upis novog sadržaja. Celija sa razelektrisanim plivajućim gejtom ponaša se kao prazna memorijска celija. Zato se u praznom EPROM-u u svakoj celiji nalazi logičko „0“, odnosno na svakoj adresnoj lokaciji broj 255 (heksadekadno FF). Prilikom programiranja u EPROM se, dakle, upisuju samo logičke nule, a prilikom brisanja te nule se vraćaju na jedinice.

Familije

Na tržištu se, trenutno, mogu naći dve nekompatibilne familije EPROM-a: jedna nosi oznaku 27XXX i dolazi iz proizvodnih hala čuvenog Intelâ, a druga 25XX i pripada Texas Instruments. Kao i sa mnogim drugim stvarima, Texas Instruments ni ovde nije imao sreće: Intelove stranarde i Intelovu familiju, koja, povrh svega, ima i nešto više članova, privatno je znatno više japanskih i evropskih proizvođača memorija i TI je ostao gotovo usamijen. Jedan tip EPROM-a (2516) kompatibilan je, doduše, sa Intelovim 2716, ali su zato TMS 2532 i TMS 2564, sa drugačijim rasporedom pinova i nešto drugačijom procedurom programiranja, ostali sasvim izvan „serije“. Njihova uloga se, izgleda, svela na to da plaže kupce visokom cenu i da besvesni komplikuju univerzalne uredaje za programiranje EPROM-a!

Nas programator EPROM-a uraden je prema Intelovim standardima. On može da programira sve članove Intelove familije — 2716, 2732, 2732A, 2764 i 27128 — osim najnovijeg 27256, jer se on pojavio kada je uredaj već bio završen. Iako, u principu, ne bi bilo naročitih teškoća da se uredaj sposobi da programira i EPROM-e iz familije 25XX, ova pogodnost ipak nije ugradnja, jer bi programator previsi iskomplikoval i softverski i hardverski. Pošto su čipovi po Intelovom standardu jeftiniji i, u to, pristupačniji od EPROM-a TI, ovakva komplikacija zaista nije bila potrebna. Za one koji ne mogu bez familije 25XX, dajemo uprednu tabelu sa rasporedom izvoda (sl. 1) i procedurom programiranja (sl. 2) za EPROM-e iz obe familije. Iz njih se može videti kako se na „galaksijinom“ programatoru EPROM-a mogu, uz malo veštine i nekoliko hardverskih rezova, programirati i TI čipovi.

Svaki EPROM ima četiri kontrolne funkcije — CS = izbor čipa, OE = dozvoljava izlazu, VPP = programski napon i PG = programski impuls — i, u zavisnosti od logičkih stanja na kontrolnim ulazima, pet različitih modova rada: mirovanje (stand by), čitanje, programiranje, provera i zadrška (inhibit). Procedura za mirovanje, čitanje i proveru je krajnje jednostavna i,

reklo bi se, spontana, i otuda izbor konceptije programatora treba da vodi računa prvenstveno o tome kako da se na najbolji način zadovolje kriterijumi za proceduru programiranja. (Zadrška se koristi samo kod istovremenog programiranja više EPROM-a različitih sadržaja.)

Procedura programiranja ne bi, takođe, bila vredna pažnje da upis jednog bajta, sa stanovišta mikroprocesora, ne traže čitavu većnost: mikroprocesor drži adresu i podatak koji joj pripada stabilnim svegu nekoliko stotina nanosekundi i odmah prelazi na sledeću, dok je EPROM-u za prihvatanje podatka potrebno čitavih 50 milisekundi. Zato okončnu svakog programatora čini sklop koji „zaustavlja“ adresu i podatak dok se ovaj ne upiše u EPROM i, razume se, obezbeđuje programski impuls u trajanju od 50 ms. Klasična konceptija programatora EPROM-a zasniva se na tzv. osmobilnim „lećevima“ (latch) — primitive memorijama koje su u stanju da „upamte“ jedan jedini bajt. Njihov zadatak je da „zamrznu“ adresu i podatak u trajanju od 50 ms. Uz izlazne bafere, adresni dekoder i još nešto malo hardvera, ovakva rešenja obavežno kralji i višeplinski preklopnik za izbor tipa EPROM-a.

PIO 8255

Svojstva koja se očekuju od ovakvih sklopova objedinjena su u jednoj posebnoj vrsti čipa poznatoj pod imenom PIO (Parallel Input Output Port — grubo: paralelni ulazno-izlazna kapija). Ova vrsta čipa je specijalno projektovana za dvosmernu paralelnu komunikaciju mikroprocesora sa periferijskim uređajima, pri čemu se njihova funkcija određuje isključivo programskim kontrolom. Programator EPROM-a, bez sumnje, predstavlja najjednostavniji uredaj koji pomoći PIO čipu može da se spregne sa mikroprocesorom. Zašto se ta pogodnost ne bi iskoristila da se samo sa nekoliko čipova napravi uredaj profesionalnih mogućnosti i — karakteristika?

Svaka porodica mikroprocesora mora, neizostavno, da ima i svoj PIO čip. On se u Zilogovoj familiji Z80 zove, jednostavno, Z80 PIO. Mada je namenjen za spregu sa računarima izgrađenim oko procesora Z80, u ovom programatoru EPROM-a je ipak upotrijebljen Intelov čip iz familije 8085 sa oznakom 8255AC-5. Zašto? Pored toga što je njegova i mašinska i programska spregičak i sa „tudim“ procesorima jednostavnija od spregre Z80 PIO sa svojim sopstvenim procesorom, Intelov PIO čip ima čitav jedan port više, što u ovoj primeni predstavlja presudnu prednost. Da budemo jednočasno jasni: Z80 PIO se ne može koristiti u ovoj shemi bez ozbiljnijih hardverskih izmena jer nema dovoljno izlaza da se ostvare sve potrebne funkcije.

Sprega PIO čipa 8255 sa mikroproceso-

rom (čitač: kućnim računaram) je veoma jednostavna. Pa ipak, ovom čipom je u Intelovom katalogu posvećeno tridesetak strana, što sasvim reči govori o njegovoj fleksibilnosti i univerzalnosti primene. Iako bi detaljnije upoznavanje sa ovim čipom bilo bez sumnje, od velike koristi, jer je nezamjenljiv u interfajnsima paralelnoj tlocrti, ovo ga puta se, silom prilika, moramo ograničiti samo na najelementarnije stvari — dakle samo na ono što je od bitne važnosti za razumevanje rada programatora EPROM-a. Pažljiv konstruktör će, verujemo, u takvog opisa izvući dovoljno praktičnih podataka za mnoge druge primene.

za mnoge moguće primjene. PIO čip 8255 sastoji se od tri porta — A, B i C — za dvosmjerni prenos podataka kontrolnog registra koji određuje način rada svakog od njih. Za upravljanje čitavim čipom dovoljno je svega nekoliko kontrolnih impulsa: CS poziva PIO čip i omogućuje komunikaciju sa mikroprocesorom; RD nareduje PIO čipu da postavi podatak na DATA BUS mikroprocesora tako da ga ovaj prihvati i pročita, a WR da upamtí podatak koji preko DATA BUS-a upravo prima. Preko ulaza AO i AI pozivaju se pojedini portovi (vidi tabelu). Ako su, na primer, obe ulaze niske, bice aktivan PORT A, a ako su obe visoka kontrolni registar.

A1	A0	PORT
0	0	A
0	1	B
1	0	C
1	1	kontrol registra

PIO čip 8255 može da se koristi u tri različite vrste rada:

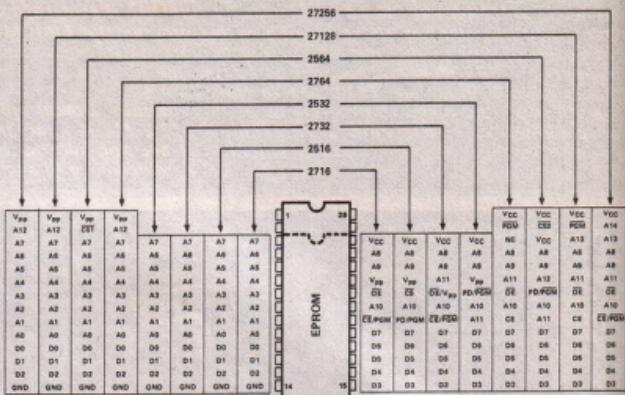
modalitet 0 svi portovi i ulaz i izlaz
modalitet 1 interaktivni ulaz i izlaz
modalitet 2 dvostruki bas

Vrstom rada se upravlja potpuno programski. Za to su dovoljne samo dve instrukcije – podaci se upisuju pomoću mašinske ili bežijk naredbe OUT, a čitaju pomoću mašinske ili bežijk naredbe IN. Da bi se odredila vrsta rada čitavog cipa i svakog porta u njemu, potrebno je pre svega prozvati kontrolni registar naredbom OUT i u njega upisati tzv. kontrolni bajt. Svaki bit u kontrolnom bajtu ima za PIO čip određeno značenje:

bit 6 i bit 5
 izbor modalitet
 00 = modalitet 0 bit 1
 01 = modalitet 1 PORT B
 10 = modalitet 2 1 = ulaz
 0 = izlaz

bit 4
PORT A
1 = ulaz
0 = izlaz

0 = izlaz	PORT C (gornji)
bit 3	1 = ulaz
PORT C (gornji)	0 = izlaz
1 = ulaz	
0 = izlaz	



Svi EPROM-i sveta: Uporedni raspored izvoda na svim tipovima EPROM-a koji se mogu naći na tržištu: naš programator, razume se, nije dovoljno univerzalan da prihvati svaki od njih, ali to nimalo ne umanjuje njegovu vrednost

2716		2732		2732A		2764		27128		2516		2532		2564	
RD	PGM	RD	PGM	RD	PGM	RD	PGM	RD	PGM	RD	PGM	RD	PGM	RD	PGM
NC		NC		NC		A12		A12		NC		NC		CS1	
*	*	*	*	*	*	#	#	#	#	*	*	*	*	"0"	"0"
CE/PGM	CE/PGM	CE/PGM	CE/PGM	CE	CE	PD/PGM		A11		A11		PD/PGM		PD/PGM	
"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"
OE	OE/V _{pp}	OE/V _{pp}	OE/V _{pp}	OE	OE	CS		PD/PGM		PD/PGM		PD/PGM		PD/PGM	
"0"	"1"	"0"	"25	"0"	"21	"0"	"1"	"0"	"1"	"0"	"1"	"0"	"0"	"0"	"0"
V _{pp}	A11	A11	A11	A11	A11	V _{pp}		V _{pp}		V _{pp}		V _{pp}		A12	
"1"	+25	#	#	#	#	"1"	+25	"1"	+25	"1"	+25	#	#	"1"	
V _{CC}	V _{CC}	V _{CC}	V _{CC}	N.C.	A13	V _{CC}									
+5	+5	+5	+5	+5	+5	+5	+5	+5	+5	+5	+5	+5	+5	+5	+5
NC	NC	NC	NC	PGM		PGM		NC		NC		NC		CS2	
*	*	*	*	*	*	"1"	"0"	"1"	"0"	*	*	*	*	"0"	"0"
25 V	25 V	21 V	21 V	21 V	21 V	25 V									

„nije važno
„dinamički ulaz

Procedura programiranja i čitanja EPROM-a: Uporedni raspored izvoda sa kontrolnim impulsima i logičkim nivoima za različite tipove EPROM-a; kod EPROM-a sa 24 izvoda broj nožice je za dva manji od onoga koji je naveden (nožica 1 dolazi na nožicu 3, a nožica 24 na nožicu 26 u podnožju).

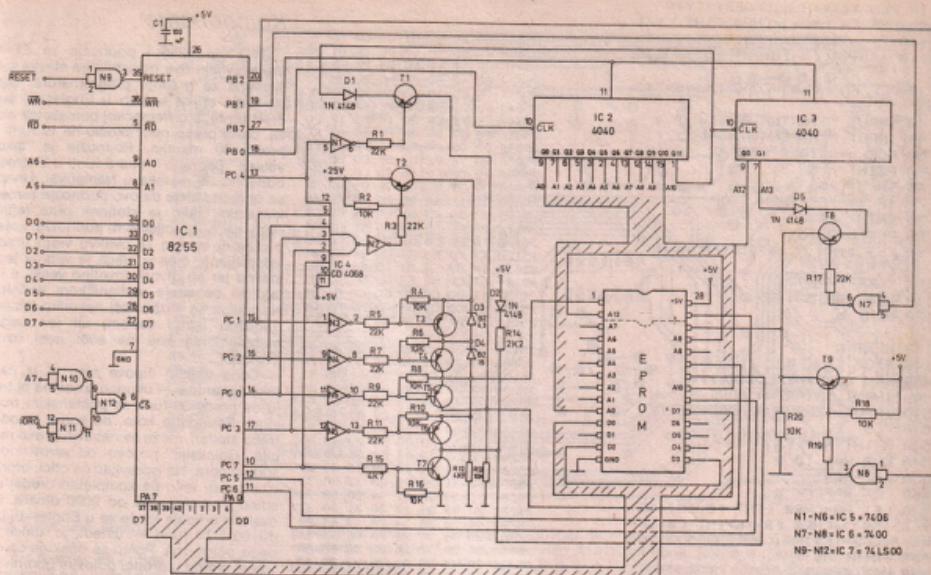
Upisivanje kontrolnog bajta u kontrolni register naziva se inicijalizacija PIO čipa, ona se ne može izbeći — bez nje PIO čip naprsto ne zna šta treba da radi! Nakon ovoga, PIO čip ostaje u određenom modalitetu, a portovi u određenoj vrsti rada, sve dok se ponovo ne pozove kontrolni register i u nešta ne upiše novi kontrolni bajt.

Pretpostavljamo da želimo daPIO čip radi u modalitetu 0 i da sva tri porta funkcionišu kao izlazna. Ako su kontrolni ulazi $A0 = 1$ i $A1 = 0$ vezani na najniže adresne linije, a CS na liniju $A2$, port A će imati adresu 0 (binarno $000 - A2=0, A1=0$ i $A0=0$), port B — 1

registar 3 (bin 011). Kontrolni bajt treba da ima vrednost 128 (bin 10000000).

Bejzjk naredba OUT 3, 128 (= prozor kontrolni registar i upisi u njega broj 128) izvršće inicijalizaciju čipa i odrediti vrstvu rada - setovati flag za modalitet (bit 7 = 1), „uključiti“ modalitet 0 (bit 6 = 0, bit 5 = 0 i bit 3 = 0) i postaviti PORT A (bit 4 = 0), PORT B (bit 1 = 0) i PORT C (bit 3 = 0 i bit 0 = 0) u izlazni modalitet.

Mini-program koji bi nečim uposlio svaki od portova mogao, na primer, da izgleda ovako:



```

10 REM *****HEXLOADER*****
20 DEF FN h(b$)=CHR$( INT (a/16)+48+7*(a>159))+CHR$( a-16*INT (a/16)+48+7*(a-16*INT (a/16))>9
30 DEF FN h(b$)=CODE b$-48-7*(b$>9)
40 INPUT "STARTNA ADRESA:":s
45 INPUT "ZAVRSNA ADRESA:":f
50 FOR n#=T0 F STEP 8
55 PRINT n;" "
60 PRINT a$;
65 INPUT a$;
70 IF a$="S" THEN STOP
75 PRINT a$;
80 FOR b#=T0 TO 7
85 LET z=FN h(a$)+16*FN h(a$)2
90 LET z=z+1
95 NEXT b
100
105 NEXT n
110 STOP
115 LET a$=a$(4 TO 7)
120 NEXT b
125
130 NEXT n
140 STOP
499 REM *****HEXLIST*****
500 INPUT "STARTNA ADRESA:":X
510 INPUT "ZAVRSNA ADRESA:":Y
520 CLS
530 PRINT X;" ";
540 FOR A#=1 TO 8
550 LET A$=" "
560 LET B$=PEEK X
570 LET A$=CHR$( INT (B/16)+48+7*(B>159))+CHR$( B-16*INT (B/16)+48+7*(B-16*INT (B/16))>9)
580 PRINT A$;" ";
590 LET X=X+1
600 IF Y=X THEN STOP
610 NEXT A
620 PRINT
630 GO TO 530

```

Program 1

Program za unositiči masinskog koda (tzv. hex-loader): Bajtovi se ne unose jedan po jedan, kako je to uobičajeno, nego red po red — **ENTER**, dakle, treba pritisnuti tek na kraju reda

Program 2

Program u bejziku ispisuje meni i prosledjuje parametre mašinskom programu — ovo rešenje je prihvaćeno da bi se izbeglo pisanje novog editora

Program 3

Mašinski kod

Komponente

PIO čip 8255 i podnožje za EPROM predstavljaju dve najozbiljnije stavke u troškovima za gradnju programatora. Čip je do kraja aprila koštao u Engleskoj svega 3-4 funte, a u Nemačkoj petnaestak maraka, da bi preko noći skočio na 19 funti (!) i preko 30 maraka. Podnožje je, takođe, veoma „paprenje“ oko 8 funti u Engleskoj, odnosno 35 maraka u Nemačkoj. Unapred se odrećite ideje da ovo podnožje zamjenite klasičnim, iako je stolnik puta jeftinije. Radi se o specijalnom podnožju koje se proizvodi upravo za ovakvu vrstu uređaja. Postavljanje čipa u njega je veoma jednostavno jer su otvori dovoljno veliki, a kontakt se ostvaruje mehaničkom kočnicom. Na tržištu se može naći nekoliko tipova sa približno istom cenom, ali je ubedljivo najbolji onaj koji na sebi nosi oznaku „Textool“.

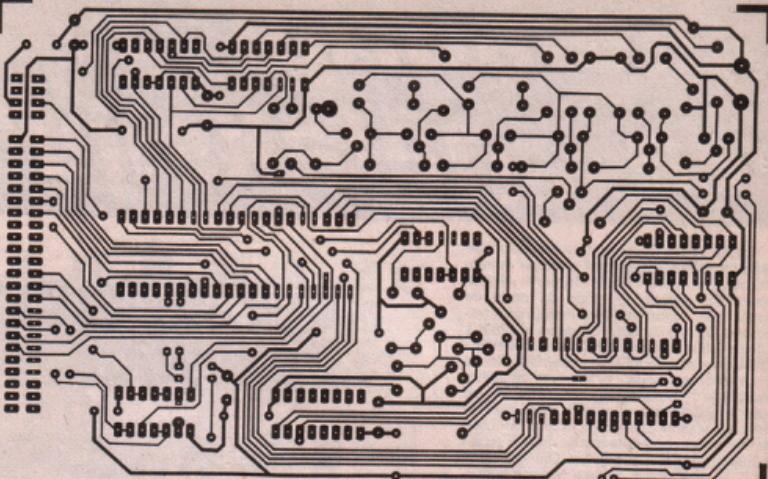
Cene ostalih čipova nisu niži od daleko šokantnije — ukupan iznos ne bi trebalo da pređe 35 funti sa poštanskim troškovima. Stampano kolo, na kome takođe ne treba štedeti, može se nabaviti preko redakcije „Galaksije“ po ceni od *verovatno oko 1000 dinara*. Na ispravljalici će otiti, otprilike, ista suma, tako da kompletan uredaj ne bi smeo da koštira više od 9000 dinara. Cena ovakvih uređaja kreće se u Engleskoj između 60 i 70 funti — ušteda je, dakle, više nego odgledina. Posto se očekuje da će se tržište smiriti u drugoj polovini godine i cene krenuti blago nizbrdo — veruje se da one više nikada neće vratiti na staru meru — čitava kalkulacija može da se pokaže i značajno povoljnijom.

OUT 0, 255 — uključi sve izlaze na portu A
 (255 = 11111111)
 OUT 1, 0 — isključi sve izlaze na portu B (0 = 00000000)
 OUT 2, 129 — uključi prvi i poslednji izlaz na portu C (129 = 10000001)

Portovi, a sуштина јесте у тиме, неće променити стање сваког микропроцесор новом OUT инструкцијом на упути нови захтев и нови — податак. Port, другим речима, може да остане „замрзнут“ бесконачно дugo, а да то никад не омете микропроцесор — уместо да чека да периферијски уредјај заврши послати ви у мртвој постели, он може да ради на другим задацима.

Adrese

U našem projektu za povezivanje PIO čipa i aktiviranje portova primjenjene su nešto drugačije adrese. PIO čipu je, razumje se, svejedno koja ga adresa poziva (koje će adresne linije biti vezane na CS, A0 i A1), ali ne i mikroprocesoru, pa ni računaru „galaksiju“. I u Spectrumu, iz jednostavnog razloga što ta adresa ne smije biti u memorijskoj, već se mora nalaziti u tzv. ulazno-izlaznoj (I/O) adresnoj mapi. Iz te mape se uzimaju adrese za rad sa svim periferijskim uređajima, pa se mora uzeti i adresa za programator EPROM-a. Procesor Z80 u ovoj mapi ima samo 256 adresu (A0—A7), a toliko ih ima i računar „galaksija“. Spectrum, doduše, i ovde ima 65536 adresu, ali je najbolje koristiti adresne linije A5, A6 i A7, jer ih sam računar ni za šta ne koristi. Ove adresne linije, dakle, odgovaraju u svakom slučaju. Kod Spectruma valjaju zadovoljiti još jedan uslov: adresne linije A0—A3 moraju biti na visokom nivou da bi se



zaštitele neke već postojeće adrese u I/O mapi.

Za prozivanje PIO čipa odabrana je adresna linija A7 i to za uslov kada se nalazi na niskom nivou. Ona, sama za sebe, međutim nije dovoljna. PIO čip sma da se aktivira samo kad se mikroprocesor obraća svojoj I/O mapi. On to čini jedino kada dobije IN ili OUT instrukciju. Da bi periferijski uređaj uopšte znao da mu se obraća, mikroprocesor mora u tom času da spusti nivo na svom izlazu IORQ (=tebi kažem). Jednostavan sklop sa NI kapijama omogućuje mikroprocesoru da prozove PIO čip samo ako su A7 i IORQ u isto vreme na niskom logičkom nivou. Ova situacija je moguća u 128 slučajeva (sve adrese od 0 do 127), ali to nije ni od kakvog značaja, jer u tom času ne radi nijedan drugi periferijski uređaj i ne može da dode ni do kakve zasebne. Apsolutno prozivanje PIO čipa, jednom jedinstvenom adresom, samo bi nepotrebno iskomplikovalo uređaj. Na kontrolne ulaze A0 i A1 vezane su adresе A6 i A5 (A0 = A6, A1 = A5 — logičnije bi bilo obrnuto ali ovo malo pojednostavljuje ispravu). Pošto sa prozivanjem bilo kog porta mora istovremeno da bude prozvan i PIO čip, u adresu mora da uđe i taj uslov.

A5 A5 A4 A3 A2 A1 A0 dekadno adresa

0 0 1 1 1 1 1 31 PORTA

0 0 1 1 1 1 1 63 PORT C

1 0 0 1 1 1 1 93 PORT B

1 1 1 1 1 1 1 127 kontrolni

register

Pošto između programatora EPROM-a i mikroprocesora u računaru ne postoji nijedna interakcija („handshaking“ — signali kojima mikroprocesor i PIO čip razmjenjuju informacije da li su završili određenu operaciju ili ne), PIO čip radi u modalitetu 0. Preko porta A prenose se podaci koji se čitaju u EPROM ili čitaju iz njega, preko porta C impulsni i naponski nivoi koji brinu o proceduri programiranja, verifikacije i čitanja za različite tipove EPROM-a, a preko porta B impulsni za resetovanje i okidanje programskog brojača, kao i za elektronski

preklopnik koji na nožicu 24 dovodi ili napajanje (do 2764) ili adresnu liniju A13 (27128). Očigledno je da portovi B i C uvek rade u izlaznom, a port A prilikom programiranja u izlaznom, a prilikom čitanja u ulaznom modalitetu. Potrebna su, dakle, samo dva kontrolna bajta:

128 — modalitet 0, port A izlaz, port B izlaz, port C izlaz

144 — modalitet 0, port A ulaz, port B izlaz, port C izlaz.

Adresiranje memorijskih lokacija u EPROM-u prilikom čitanja i programiranja izvedeno je, takođe, na krajnje neobičan način. Bilo bi logično da se taj posao poveri mikroprocesoru — adresiranje bi se tada

odvijalo automatski. Tada bi, međutim, bila potrebna još dva „leća“ za privremeno „zamrzavanje“ adresa. Stampano kolo bi se toliko iskomplikovalo da bi, najverovatnije, bila potrebna dvostruka štampa, a programiranje postalo znatno složenije. Zato je u programator ugrađen poseban programski brojač sa dva čipa 4040 (brojač sa modulom 16 — deli sa 4096). Ovakva konceptacija je elegantna i u mašinskom i u programskom smislu, ali ima i jednu slabu tačku: ako se EPROM ne programira suksesivno, od početka do kraja, nego na preskok, treba dopisati petlje za okidanje brojača u prazno uvek kada se preskaci neka adresa.

Hardverska konceptacija programatora EPROM-a omogućava potpunu programsku kontrolu svake funkcije prilikom čitanja, testiranja, programiranja i verifikacije EPROM-a i, samim tim, ostavlja korisniku potpuno određene ruke da odredi funkcionalnu konceptiju i prilagodi uređaj svojim videnjima i svojim potrebama. Opisan hardverski rešenje nedostaje, možda, samo jedna stvar: sopstveni ROM sa programom za rad uređaja koji bi se „aktivirao“ čim se programator priključi na računar. Time bi se izbeglo dosadno učitavanje programa sa trake svaki put kada treba isprogramirati neki EPROM, ali bi se programator ipak nepotrebno iskomplikovao. Rad sa programatorom je, inače, u svemu ostalom veoma komoran: nakon startovanja programa dovoljno je uneti tip EPROM-a i početnu adresu u RAM-u. Sve ostalo se dešava gotovo samo od sebe.

Programator EPROM-a, mada mu cena nije mala, spada u onu vrstu uređaja koja ne isplaćuje praktično čim proradi i po tome je u kompjuterskom svetu potpuno jedinstven. Razlike u ceni programa na kaseti i u EPROM-u su toliko drastične (prazan EPROM 27128 košta oko 25, a napunjeni i do 60 funtili), da se programator ponekad isplaćuje samo jednim programanjem! Ovakvi uređaji, međutim, imaju i jednu mnogo dragoceniju vrednost: oni u programerskom i konstruktorskom radu otvaraju vrata za sasvim nove nove ideje.

Jova Regasek

Ovo nije sve!
**U „RAČUNARIMA 3“
SOFTVERSKA KONCEPCIJA I
PRAKTIČNA REALIZACIJA PLUS**
• Kompletan program za
računar „galaksija“ sa sledećim
opcijama
— testiranje eproma i...
— parcijalno testiranje između
dve adrese
— kopiranje EPROM-a u RAM
i...
— ... parcijalno kopiranje
između dve adrese
— programiranje EPROM-a iz
RAM-a i...
— ... parcijalno programiranje
između dve adrese
— verifikacija i...
— ... verifikacija između dve
adrese
• usavršeni program za računar
Spectrum sa opcijama za
testiranje, kopiranje,
programiranje i verifikaciju bit
po bit

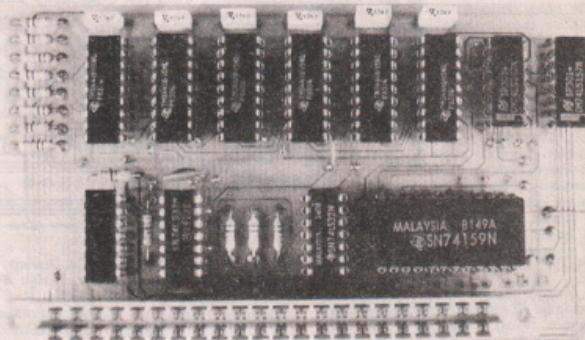
memorija za „galaksiju“

„Galaksija
48 K“

Da bi jedan računar funkcijonisao kao skladna celina, svi njegovi skloovi moraju da budu primereni jedni drugima. Moćan operativni sistem ne može da dode do izražaja sa malom radnom memorijom (računaru QL nije, izgleda, dovoljno ni 128 K), niti će velika memorija biti dobro upošljena sa skromnim operativnim sistemom. Kolika je memorija, onda, primerena računara „galaksija“? Skeptici bi rekli da je on već sada imao dovoljno, ali u to ih može razuvjeti svaki vlasnik „galaksije“ koji je pokušao da na svom računaru napiše barem jedan ozbiljniji program u bežizku. Oni koji tvrde da će se „galaksija“ sa 54 K radne memorije, „osećati“ kao u tudem delu koje joj je, u to, za tri broja veće“ verovatno su nešto viši u pravu. Učitavanje programa u toliku memoriju, slazemo se, može da potraje čitavu večnost. Nije, međutim, sve ni u brzini učitavanja cak i ako ona ne bi mogla da se poboljša.

Obilna memorija možda izgleda nezgrano uz računar „galaksiju“, ali ona u mnogome može da nadoknadi skromniju osobinu operativnog sistema. Domaći ljubitelji računara, koji siliom prilike nabavljaju mašine koje su znatno ispod njihovih potreba, u doradama operativnog sistema često vide jedini izlaz. Operativni sistem „galaksije“ je savršeno otvoren za ovakve dorade — računaru nedostaje samo malo više memorije. Veća memorija, pored toga, predstavlja osnovni ulog za hardversku poboljšanje računara, kao što je, na primer, finija grafika I, naročito, za programski zamah. Ko će se privatiti da napiše „obziljan“ program za „galaksiju“ kad se sa 6 K redne memorije ne mogu pisati takvi programi?

lako trenutno potrebe, po pravilu, ne mogu da budu pouzданo merilo za veličinu memorije jednog računara — nju će diktirati razvoj i hardvera i softvera koji se sada ne može da predviđa — svaki vlasnik „galaksije“ će, očigledno, morati sam da odredi svoju mjeru. Uz one kileabajte koji već postaju u računaru: 16 K, sasvim sigurno, predstavljaju najrazumniju količinu. Ako vreme pokaže (verovatno i hoće, ali ne tako brzo) da je i to nedovoljno, memorija se kasnije može jednostavno, uz dodatak samo dva ili cetiri čipa, proširiti za još 16 ili 32 K. Od potrošačke groznice nikada nije imao koristi, pa nećete ni vi i nemojte se plasiti da kasnije nećete naći potrebne čipove — doći ćete do njih sigurno, i to, verovatno, znatno jeftinije. Realno, 32 K predstavlja gornju razumnu granicu.



„Galaksija 48“: Memorisko proširenje povećava radnu memoriju računara „galaksija“ u zavisnosti od broja ugrađenih čipova, za 16, 32 ili 48 K; zahvaljujući fleksibilnom adresnom dekoderu, proširenje se može pomerati po memoriskoj mapi u skokovima od po 4 K.

Statičke ili dinamičke?

Ako je, u trenutku kada je koncipiran računar „galaksija“ i moglo da bude dilema oko tipa čipa koji će se koristiti u radnoj memoriji, sada nedoumici nema ni najmanje mesta. Statičke memorije, koje se ugrađuju u „galaksiju“ toliko su poskušane na svetskom tržištu da ih preporučavamo samo u izuzetnim prilikama — i mikronskim količinama. Prema trenutnom paritetu, dinamički kileabajt je gotovo osam (!) puta jeftiniji od statičkog, što praktično znači da se za „galaksijinu“ 6K u osnovnoj verziji može dobiti gotovo 48 K u „dinamičkim“ bajtovima! Na izbor statičkih memorija za osnovnu verziju „galaksije“ najviše je uticaj bojazan da dinamičke memorije mogu konstruktorma da zagođaraju život više nego što je to nužno. Kakva je razlika između statičkih i dinamičkih memorija i da li je zebnja konstruktora „galaksije“ bila opravdana?

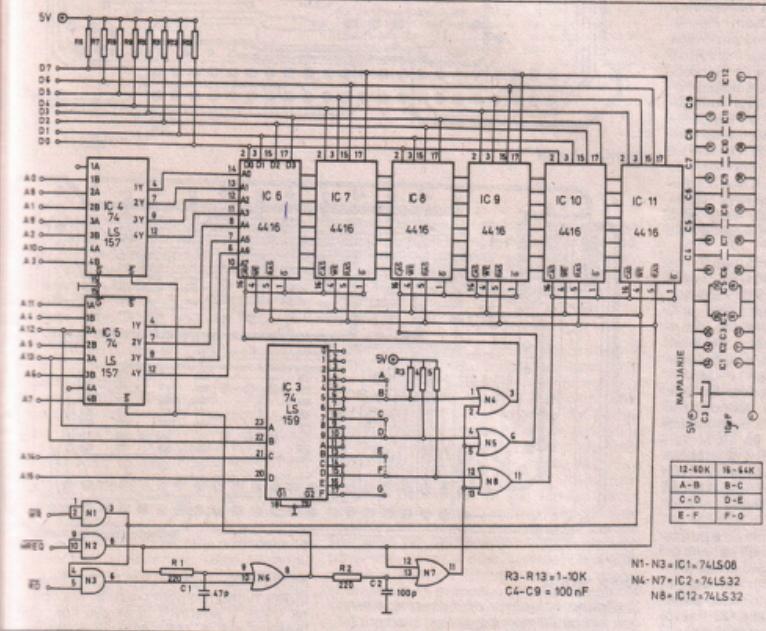
Memorijska celija u statičkoj memoriji izgrađena je na bazi tzv. flip-flopa — dosta složenog elektronskog sklopa koji se ostvaruje sa pet-šest tranzistora. Jednom upisan, sadržaj ostaje u ovakvoj celiji sve dok se preko njega ne upiše novi ili dok se memorija ne isključi iz struje. Svi sklopopi potrebni za rad sa ovom vrstom memorije ugrađeni su u sam čip, pa je procedura upisivanja i čitanja podataka potpuno spontana. Statičke memorije su, međutim, veoma glogazne — to je značajnija memorijска konfiguracija zahteva: bi mnogo čipova i ogromno štampano kolo. Proizvodnja je do sada posloša za rukom da u jedan čip

smestee najviše 8192×8 celija (osam kilobajta), ali takav čip košta 50 dolara!

Kod dinamičkih memorija celija je izgrađena na bazi kondenzatora. Pošto zauzimaju manji prostor od celije sa tranzistorima, kondenzatori obezbeđuju znatno veću gustinu pakovanja i — manju cenu po bajtu. Mikrometarske dimenzije, međutim, povlače za sobom i „mikrometarski“ kapacitet — pet do deset stotina delova pikofarad! — pa se celija brzo prazni i njen sadržaj se nepovratno gubi. Najbolji tipovi dinamičkih memorija mogu da pamte podatak najviše četiri milisekunde. Zato se njihov sadržaj stalno obnavljati — to se postiže takozvanim „osvežavanjem“. Osvežavanje dinamičkih memorija, koje utjeruje strah kod neobaveštenih ljubiteljima računara, predstavlja trivijalan proces koji procesor 7208 obavlja automatski na kraju svakog mašinskiog ciklusa. Osvežavanje se postiže proizvajanjem memorijskih celija „na rasprodoru“.

Druga specifičnost dinamičkih memorija, međutim, nije toliko bezazlena i zahteva i nešto malo dodatnog hardvera i određenu proceduru za upisivanje i isčitavanje podataka. Memorijске celije u dinamičkoj memoriji su organizovane u redove i kolone, ali je direktni pristup omogućen samo redovima. Zato se adresa ne saopštava memoriji odjednom nego — kao i kod mobilnih procesora — u dvaput. U memoriju se najpre upiše adresa za redove a potom — u istom mašinskiom ciklusu! — za kolone. Celija tada postaje sprema za prim ili oda svoj sadržaj. Nešto malo komplikovanija procedura rada i zavisnost

Radna memorija, bez sumnje, predstavlja jedno od osnovnih merila snage jednog računara. Bez obzira na snagu i na kapacitet njihovog operativnog sistema, u savremenim kućnim računarima se za memorisanje, po pravilu, koristi sav prostor koji dopušta mikroprocesor. Takva koncepcija se, razume se, nije mogla očekivati i od računara „galaksiju“, ali nju mogu (i, rekli bismo, moraju) da sproveđu svi oni koji nameravaju da naš računar koriste iole ozbiljnije. Iako su za hobiste, zbog oskudice na svetskom tržištu čipova, došli veoma teški dani, u redakciji smo razvili memorisko proširenje „galaksiju 48“, koje „galaksiju“ značajno približava „ozbiljnom“ računarskom društvu. Proširenje omogućava povećavanje memorije u blokovima od po 16 K, a može se priključiti na verzije računara sa 2 i 6 K.



KOMPONENTE

Integrirana kola
 74LS416 2, 4 ili 6
 74LS161 2
 74LS08 1
 74LS32 1 ili 2
 74158 1

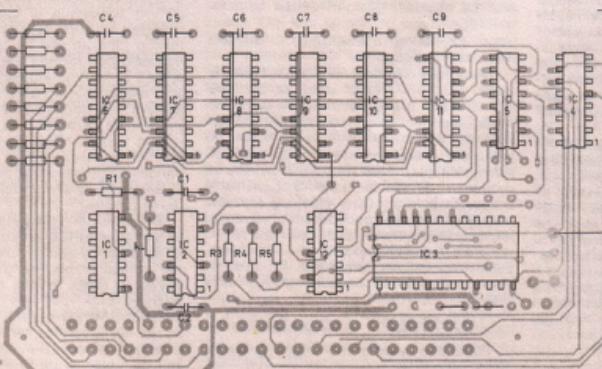
OPTORNICI
 R1, R2 220 ohm
 R3—R13 1—10 ohm

Kondenzatori
 C1 56 pF
 C2 100 pF
 C3—C7 100 nF
 C8 10—47 uF (tantali)

Podnožka
 18 notica 2, 4 ili 6
 16 notica 2
 14 notica 2 ili 3
 24 notica 1

Konektor
 S x 22 razmak 1
 3.95 mm

Montažna shema proširenja „galaksija 48“: Iako je primjenjena dvostruka štampa, nekoliko kratkospojnika se nije moglo izbjeći; kratkospojnici koji prolaze ispod memoriskih čipova služe da skrate vezu kondenzatora za blokodator sa masom, čime se povećava njihova efikasnost u „pepljanju“ napajanja za dinamičke čipove; kratkospojnici ispod memoriskih čipova služe za programiranje adresnog dekodera — na slici je prikazana situacija kada se u „galaksiji“ nalazi 6 K RAM-a (kada su ugradena sva tri memoriska čipa).



Električna shema proširenja „galaksija 48“
 memorija je podjeljena u tri nezavisna blokova od po 16 K

od splošnijeg hardvera predstavlja, praktično, jedinu komplikaciju u radu sa dinamičkim memorijama. Ova komplikacija je, međutim, prilično trivijalna i nije vredna pažnje ni sa stanovišta cene, ni sa stanovišta složenosti, ni sa stanovišta pouzdanosti rada uredaja. Dinamičke memorije, uz to, rade doslovce u milionima primeraka kućnih računara — nema nikakvog razloga zašto ne bi mogle i u „galaksiji“!

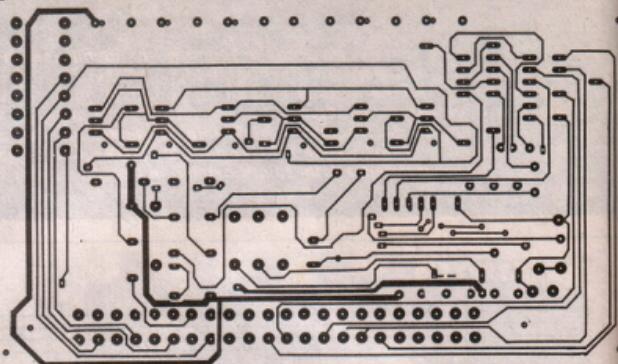
I sve i nešto

Na tržištu postoji četiri tipa dinamičkih memorija. Prvi među njima, 4416, organizacija 16384 bita $\times 1$, predstavlja najjeftiniji i, možda, najlošiji izbor. Osam čipova, koliko je potrebno za 16 K, koštaju svega oko 2000 dinara. Ove memorije, na žalost, zahtevaju trostruko napajanje, a za veće kapacitete strahovito složeno štampano koliko i mnoštvo — čipova! Po svojoj konfiguraciji i tehničkim rešenjima, ove memorije predstavljaju prevaziđeni tip. Memorije dvostruko većeg kapaciteta, 32768 $\times 1$, koje se ugrađuju u računar ZX Spectrum, mogu da budu znatno privlačniji izbor. Ovaj tip memorije — dugo je trebalo da se to sazna! — ne proizvodi se nigde u svetu. Radi se o memorijama 4164 (65536 $\times 1$) kojima nije ispravna jedna sekcija!

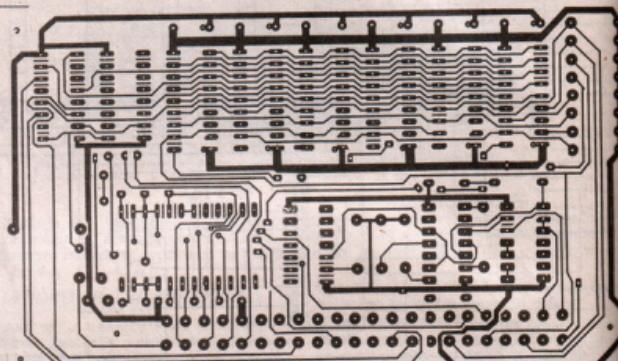
Tako, praktično, za izbor ostaju samo dva tipa. Osam čipova tipa 4164, za četredesetak funti, obezbeđuju memoriju kapaciteta 64 K! Iako se u „galaksiji“, kao ni u jednom računaru, ne može direktno iskoristiti čitav adresni prostor (nešto se mora ostaviti i za ROM, mada RAM, pod određenim uslovima, može da radi i „ispod“ ROM-a), ova konfiguracija predstavlja najlastičnije „rešenje“. Memorija jednostavno pokriva sve prazne prostore u memorijskoj mapi, uključujući i prostor za ROM 2, i uspešno zamjenjuje jedan ili dva statička RAM-a (D i E). Oni vlasnici „galaksije“ koji još nisu nabavili ove čipove mogu, tako, za isti novac da kupe gotovo potrebu memorijskog proširenja.

Za ljubitelje potpunih rešenja — a potpuna rešenja su uvek i elegantrnija i jeftinija od polovičnih — ova varijanta predstavlja jedini mogući izbor. U redakciji je razvijen prototip ovog rešenja koji se odlično pokazao i koji će dobro postupiti za razvoj sistemskog softvera, ali smo se ipak odlučili za drugu varijantu. Zašto? Ovo rešenje ima jedan doista ozbiljan nedostatak filozofsko-finansijske prirode — ono pred vlasnicima „galaksije“ postavlja dilemu „sve ili ništa“. Plaćamo se da bi većina ljubitelja računara pre bila prisiljena da izabere „ništa“ nego „sve“. Za njih bi to bila „supa za zdraće“ — lepo ali skupo i nepristupačno rešenje.

Cetvrti tip memorije, 4416, organizacije 16384 $\times 4$ bita, možda nije toliko fleksibilan u popunjavanju rupa u memorijskoj mapi, ali objedinjuje prednost potpunih i polovičnih rešenja. Memorija se, takođe, može proširiti po punog formata, ali se to ne mora uraditi odjednom — moguće je i u triput u blokovima od po 16 K. Jedan ovačak blok, koga čine dva čipa 4416, nije, čak ni danas, skuplji od jednog jednog čipa 6116! Ovaj tip memorije je relativno nov i malo se teže nalazi na tržištu — drže ga.



gornja strana



donja strana

Štampano kolo: Iako objavljujemo crteže obe strane štampanog kola u razmeri 1:1, preporučujemo vlasnicima „galaksije“ da se strepe do septembra i sačekaju fabričku verziju.

praktično, jedino u Engleskoj — ali tako neće ostati zadugo.

Memorijski čip 4416 ima osam adresnih ulaza, AO—A7, i četiri ulaza za podatke, DO—D3, što, praktično, znači da jedan čip može da pamti samo pola bajta (četiri bita). Zato se za svaku polovinu bajta koristi po jedan čip. Preko kontrolnih ulaza WE (upisivanje/čitanje), RAS (upisivanje adrese za redove i osežavanje), CAS (upisivanje adrese za kolone i izbor čipa) i G (dozvola izlaza) upravlja se procedurom upisivanja i isčitavanja podataka. Čip može da radi u nekoliko različitih modaliteta i veoma složenim memorijskim shemama. Mi smo, razume se, odabrali najjednostavniji način rada.

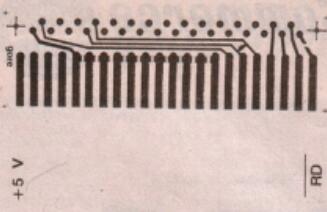
„Dinamička“ procedura

Električna shema memorije „galaksija“ 48* izvedena je klasično za ovaku vrstu sklopova, bez ikakvih ekstravagantnih zahvata i komplikovanih vremenskih kola. Adrese za redove i kolone obezbeđuju dva

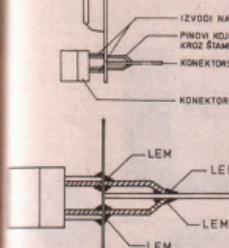
multiplexer tipa 74LS157. U zavisnosti od stanja na kontrolnom ulazu S, multiplexer na izlaze Y postavlja ili adresu koju dobija preko ulaza A (A8—A13) ili adresu koju dobija preko ulaza B (AO—A7). Na početku ciklusa upisivanja ili čitanja multiplexer postavlja na adresne ulaze AO—A7 memorijskog čipa adresu za redove, a stotinu nanosekundi kasnije na ulaze A1—A6 adresu za kolone. Adresa za redove upisuje se u memorijski čip silaznom ivicom impulsa RAS (row address select — izbor adres za redove), a adresa za kolone silaznom ivicom impulsa CAS (column address select — izbor adrese za kolone). Stotinu nanosekundi nakon kompletiranja adresne aktiviranje se izlazni baferi i podatak može da se pročita.

Procedurom upisivanja adrese upravlja se signalom MREQ preko nekoliko ILI kapica da se odgovarajućim kolima za kašnjenje. Na adresnim ulazima memorijskog čipa stalno se nalazi adresa za redove i oni se pozivaju svaki put kad se mikroprocesor signalom MREQ obraća memoriji — bez obzira da li želi da upiše podatak, da ga pročita ili da osveži memorijsku celiju. Ako da upiše ili da pročita, WR signal će pripremiti memorijski čip, a RD ili WR otvoriti ILI kapiju N6. Pošto je već upisao čip adresu za redove, MREQ će, nakon

Konektor i konektorska pločica. Sredviš-sistem montaže konektora omogućuje da se na računar „galaksija“ istovremeno sa memorijskim proširenjem priključi još neki uređaj, na primer programator EPROM-a ili generator zvuka, raspored izvođa na konektorskoj pločici na proširenju dopunjeno je novim signalima (RD, CS1, CS2 i CS3 i +3V).



Konektorska pločica na računaru „galaksija“. Da bi se omogućilo jednostavnije povezivanje periferijskih uređaja preko porta za proširenja, konektorska pločica mora biti priključena na računar, istovremeno, sve izmene se vrše sa gornje strane pločice, a za njih su dovoljni skupovi temelica i dva parčeta zice.



izvesnog kašnjenja (R1, C1, N6) „prekopčati“ multiplexer i dovesti na A1—A6 adresu za kolone. Nakon što se stabilizuje ova adresa, na izlazu kola za kašnjenje oko N7 (R2, C2) će se pojaviti signal CAS i upisati u memoriski čip. Prilikom upisivanja podataka čipu se istovremeno sa adresom za kolone saopštava i podatak (tzv. early write modalitet — upisivanje impulsonog CAS). Prilikom čitanja podataka se pojavljuje otinak nanosekundi nakon upisivanja adrese za kolone.

Signal CAS, istovremeno, upravlja i ulaznim izlaznim baferima — podatak se ne može ni upisati ni pročitati ako ulaz kojim se upravlja nije na niskom nivou. To, praktično, znači da se on može koristiti i za izbor određenog memoriskog bloka. Adresni dekoder 74159 određuje prema komu će bloku ovaj signal biti usmeren. Dekoder pokriva čitav memoriski prostor u blokovima od po 4 K. Pošto se radi o modelu sa tvrdom, otvorenim kolektorima, izlazi se mogu sačijati paralelno bez ikakvog ograničenja. Za blok od 16 K moraju se spojiti četiri izlaza. Shema proširenja „galaksija 48“ dopušta dve mogućnosti — nadovezivanje na postojeća 2 K radne memorije (spojeći A—B, C—D i E—F) i 6 K (B—C, D—E—G). Šta da urade oni koji su u „galaksiji“ ugradili srednju varijantu od 4 K? Moraju da dokupe još jednu statičku memoriju 2116 ili da presečanju voda CS koji vodi prema memoriji D, isključe ovaj memoriski blok. U prvom slučaju memorija se može proširiti do 64 K, a u drugom „samo“ do 32 K.

Upotrebljeni dekoder se nekome može učiniti previše glomazan za ovakvu primenu. To je, međutim, bio jedini način da se postigne izvesna fleksibilnost sistema. Od nega, osim toga, mogu imati koristi čak i oni koji ne nameravaju da proširuju memoriju do kraja — na konektor su izvučeni signali CS za poslednjih 12 odnosno 16 K. Oni se elegantno, bez potrebe za novim dekoderom, mogu iskoristiti u nekom drugom proširenju — na primer proširenju

ROM-a u kombinacijama 1×16 K, 2×8 K ili 4×4 K! Oni koji se unapred odriču poslednjih 16 K radne memorije, treba na štampanom kolu da presekut veze između nožica 14 i 15 i 15 i 16 i time razdvoje CS signale u poslednjih 16 K adresnog prostora.

Štampano kolo...

Štampano kolo je, zbog dosta složene strukture, uradeno u dvostrukoj štampi. To će, znamo, razluzititi one hobičke koji sve vole da rade sami, naročito ako na ono što i sami mogu da urade moraju da čekaju dva-tri meseca. Ovakvi sklopovi, međutim, podrazumevaju štampano kolo profesionalne izrade — inače se može desiti da nikada ne prorade. Za one koji imaju razumevanja za domaće prilike i znaju da se neke stvari moraju i čekati, pripremili smo servis za nabavku štampanog kola. Za one koji gore od nestreljivosti i koji imaju više poverenja u sebe objavljujemo crtež obe strane u razmeri 1:1. Sa jedne na drugu stranu veza se prenosi u svim tačkama koje se nalaze na gornjoj strani — najčešće preko pinova na integriranim kolima ali ima i posebnih prenosnika. U tim tačkama (treba li to reći?) elementi se leme sa obe strane. Oni koji se odluče za samogradnjom štampanog kola moraju da nabave specijalne pinove za štampana kola, poput onih na našim fotografijama, ili da zamele cipo direktno. Ovo, naravno, važi samo za štampana kola iz amaterske radionice. Na štampanom kolu koje obezbeđuje redakciju sve rupe će biti metalizirane i hobisti neće morati da brinu gde se veza prenosi, a gde ne. Na ovakvom štampanom kolu lemi se, naravno, samo da donje strane.

... i štampani konektor

Na štampanom konektoru na računaru „galaksija“ nisu, iz neobjašnivih razloga, izvedena dva priključka bez kojih memoriski proširenje ne može da radi — priključak za +5 V za napajanje proširenja iz računara i RD priključak sa signalom za isčitavanje memorije. Zato se, pre nego što se proširenje priključi, na konektoru moraju izvršiti izvesne dorede: napajanje, komadom zice, treba dovesti na priključak 1, koji je, inače, prazan, a RD signal, direktno sa nožice 21 na mikroprocesoru,

na priključak 22; ovaj priključak je, inače, vezan sa masom računara i na ta veza, na konektorskoj pločici, mora preseći. Da bi se oslobodio prostor za CS signale, treba, takođe, preseći i sve veze između priključka 6 i 15. Sa zadnje strane memoriskog proširenja postavljena je konektorska pločica. Ona omogućuje da na računaru, istovremeno sa proširenjem, bude priključen još jedan dodatak — na primer, programator eproma, generator tonova, interfejs za štampač ili — nešto treće.

Memorijski bagovi

Testiranje sagradenog proširenja je krajnje jednostavno — ovo testiranje, zapravo, „galaksija“ obavlja automatski prilikom svake inicijalizacije — i rezultat je vidljiv čim se ukliči računar. Dovoljno je, dakle, otkucati naredbu PRINT MEM, koja izdaje količinu slobodne memorije, i na ekranu će se, u zavisnosti od broja ugradenih blokova, pojaviti brojka od koje će se konstrukturu zavrtiti u glavi. Na žalost, nije tako. Ma koliko to čudno zvučalo, operativni sistem „galaksije“ ne obezbeđuje potpunu podršku za memoriski proširenje — u njemu se potkralo nekoliko bagova koji u izvesnim situacijama otežavaju rad sa spoljašnjom memorijom. Najozbiljniji bag se odnosi na pun memoriski format. Umetao da u takvoj situaciji računa sa 54 K slobodne memorije, koliko stvarno i ima, „galaksija“ u promenljivu „kraj memorije“ upisuje nulu i ponaša se kao da nema ni jednog slobodnog bajta — odjiba da radi u programskom modalitetu! Kako je to moguće? Prilikom testiranja memorije, „galaksija“ postavlja promenljivu „kraj memorije“ na prvu lokaciju sa kojom nema odziva — umesto na poslednju sa kojoj ga ima — a prva lokacija posle FFFF je 0000!

Ova teškoća će zagorčavati život samo onima koji na postojećih 6 K dograde još 4 K radne memorije. Rešenje je jednostavno ali neeligibilno i dosta zamorno: ako se pre početka rada u promenljivu „kraj memorije“ (82A6A) upiše FFF0 „galaksija“ će ponovo postati poslušna do sledećeg — uključenja. Ručno postavljanje tzv. „ramtopa“ predstavlja u ovu situaciju, na žalost, jedino rešenje. U svim ostalim situacijama „kraj memorije“ se postavlja kako treba, ali, praktično, znači da se moramo odreći čitavog memoriskog bloka (kako „odsediš“ samo snaesnat bajtova?) koji, u našem slučaju, iznosi 4 K.

Naredba PRINT MEM, da stvar bude teža, sa više memorije počinje da izdaje sve čudnije rezultate. Ona ispravno funkcioniše samo do 22 K radne memorije, odnosno do adrese 32767. Nakon toga daje negativan broj, iz tog se ne može izvući ami baš nikavog logičan zaključak o stvarnoj količini slobodne memorije. Sa proširenjem od 16 K PRINT MEM daje rezultate 21445, od 32 K — 27706, a od 48 K — 11322! Ispitivanje promenljive „kraj memorije“ — PRINT WORD (82A6A) — daje takođe negativne brojove — 32768 (16 K), — 16384 (32 K) i 0 (48 K). U prostoru iznad 32 K RAM-a „galaksija“ u naredbama tipa BYTE i WORD prihvata samo heksadekadne ali ne i decimalne pozitivne brojove. Sva ova ograničenja, međutim, nemaju nikavog uticaja na uobičajeni rad sa memorijom — ona potpuno normalno prihvata i bežik i mašinske programe.

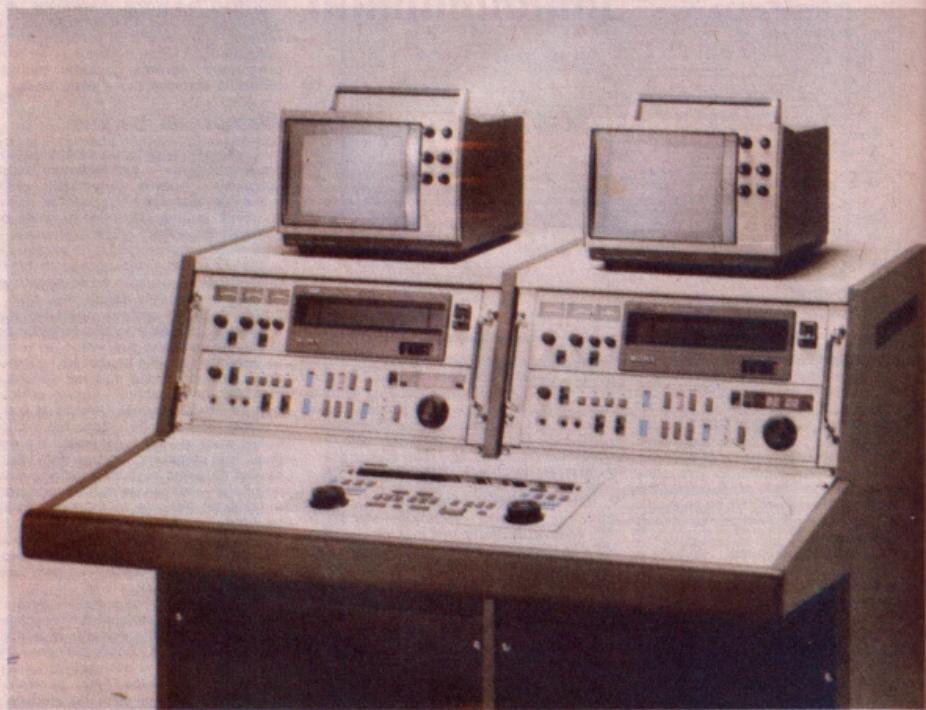
Projekat i tekst: Jova Regasek

Crtzi: Mira Todorović

Yugoslavia Commerce



SONY



SONY



Yugoslavia Commerce

„YUGOSLAVIA COMMERCE“ je preko 20 godina generalni zastupnik „SONY“-a za našu zemlju. Predmet zastupništva je celokupni proizvodni program SONY-a iz oblasti Hi-Fi i video opreme, kao i kompletan program U-MATIC opreme. Kompletan servis je obezbeđen u Beogradu, Sarajevskoj broj 5.

Od marta ove godine u Sarajevskoj br.3 otvoren je izložbeni salon SONY-a. Za sada, „salon“ nije kompletiran, ali će do kraja godine dobiti svoju konačnu fizičnost. Trenutno je izložena Hi-Fi i video oprema. To je tek prva raka na putu pronalaženja stalne koncepcije izložbenog salona.

U drugoj fazi očekujemo krajem maja, bice izložena U-MATIC video oprema. U-MATIC oprema bice posebno interesantna za RO iz oblasti turističke privrede, mada je njeni primeni daleko širi: u bankarstvu, u osiguranju, poslovima, institucijama, ... Na primeru turističke RO objasnimo vam jedan od načina primene U-MATIC video opreme. U-MATIC je pogodan:

— Za snimanje i prezentiranje kompletne turističke ponude na turističkim berzama i sajmovima. Poslovnim partnerima moćiće da prezentuju dinamiku turističke sezone tokom cele godine. Vaše mogućnosti nisu više ograničene sezonom.

U turističkim agencijama, pogotovo u tome koje se bave plasmanom turističkih aranžmana i prodajom izleta. Obogatite vašu ponudu video spotovima. Rezultati sigurno neće zastati.

— Za projekciju filmova, muzičkih, sportskih i drugih emisija. Ponudite vašim gostima video, oni su na to već navikli. Možda im baš to nedostaje. Loše vreme ne smi da vas iznenadi.

— Za organizovanje interne televizije

U trećoj fazi ustanavljanja SONY-evog salona bice predstavljeno uvođenje kompjutera u video sistem. Ova faza očekuje se krajem novembra '84 god. Kompletiranjem sve tri faze izložbeni salon mogao bi da postane ogledni centar jedne banke podataka o trenutno najavremenijejšoj audio, video i kompjuterskoj opremi.

Ako vam naš salon nije usputan, posetite nas na jednom od sledećih sajmova:

— Sajam tehnike u Beogradu: od 22. — 26. V. '84. god.

— Sajam elektronike u Ljubljani: od 3. — 10. X. '84. god.

— Interdisk u Zagrebu: 18. — 24. IV. '84. god. Da vas ukrašto upoznamo sa našim poslovnim partnerom: Kompanijom SONY.

Malo je firmi u svetu koju su poput SONY-a, za tako kratko vreme, izmenile naše svakodnevne navike. Još je manje firme koje su to uradile tako „mušnjevito“.

Ali krenimo od samog početka.

Firma SONY osnovana je u maju 1946. godine. MAJARU IBUKA i AKIRO MORITA, sa početnim kapitalom od samo 500 (USA) dolara, osnivaljaju malu firmu-radiotricicu: TOKYO TELECOMMUNICATION ENGINEERING CORPORATION, jednu od mnogobrojnih, koje su nastale odmah posle II svetskog rata, u Japunu. Za neupoređenih deset godina SONY je prevaleo put od male radionice do svetskog giganta. U prvoj deceniji lansiran je prvi japanski tranzistor (po američkoj licenciji), a ubrzo prvi na svetu SONY izbacuje tranzistorski radio prijemnik. Tako je započela SONY-eva svetska ekspanzija. Kasnije su još bila jednostavnije. Pokrenuta je lavina pronalažaca koji će lagano osvajati fabrike, institute, domove, ... Ostalo su učinili sposobni inženjeri-elektroničari i spretni menadžeri.

SONY svake godine lansira na svetsko tržište nove proizvode koji postaju sastavni deo života savremenog čoveka. Kao ilustraciju nabrojamo samo neke SONY-eve proizvode:

— pored spomenutog tranzistorskog radio prijemnika, za SONY je vezana i pojava prvog tranzistorskog TV prijemnika.

— 1958 prvi tranzistorski video rekorder.

— 1969 prvi tranzistorski portabil video rekorder.

— 1964 prvi kućni video rekorder.

— 1969 razvijen U-MATIC sistem (3/4 inča).

— 1975 razvijen BETAMAX sistem (1/2 inča).

— 1976 razvijen OMEGA sistem (1 inč).

Pored ovog ne možemo a da ne spomenu:

— TV sistem trinitron.

— usavršavanje video i audio traka (Fe i Cr trake u SONY-ov izumu).

— MAVICU (prije elektronski fotoaparat).

— COMPAKT DISK (prije digitalni gramofon u saradnji sa PHILIPS-om).

— BETAMUSIC (prvi video-kamera rekorder).

— WATCHMAN (prije džepni televizor).

— BETA HI-FI (prije video rekorder sa Hi-Fi zvučnikom).

SONY-evi izuzetan doprinos razvoju Hi-Fi i video tehnike ovim nije iscrpljen. Spomenimo i trenutno najmanju kameru na svetu: SONY CCD-65. Očekuje se njena svetska premijera.

Organizacionu strukturu kompanije SONY čine tri njeni dela:

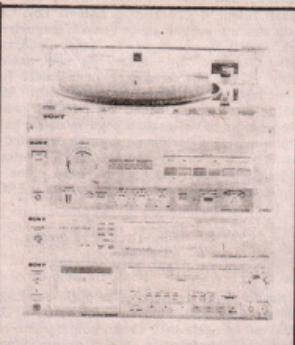
Prvi deo, koji je i bio osnovan za razvoj, ostala dva, radi na razvoju opreme široke potrošnje. Tu se proizvodi, Hi-Fi i video oprema.

Drugi deo, osnovan 1965. godine, je takozvani „INSTITUTIONAL VIDEO“ baziran na sistemu U-MATIC.

I na kraju treći deo firme „SONY-BROADCAST“ nastao je tek sedamdesetih godina razvojem video-rekordera od 1 inča.

SONY-evi ciljevi do 1990. godine su da se zadrži pozicija vodećeg video kompanije u svetskim razmerama. Stratešku točku je na razvoju novih media, kao što su: teletekst, videotekst, kabloska televizija, sateletska televizija.

U svojoj orijentaciji SONY nije zapostavio ni razvoj mikrokompjutera. Krajem godine bice predstavljen SONY-ev personalni kompjuter (uglavnom za poslovnu primenu) sa kompletom perifernom opremom. I u ovoj deceniji može se očekivati dinamičan razvoj, posebno ako se zna da SONY učaze 5-6% godišnjeg profita u istraživanje i osvajanje novih proizvoda.



Yugoslavia Commerce

OUR YUCOM TRADE
Kneza MILOŠA 60/11000 Beograd
011/688-166

IZLOŽBENI
SALON

SONY

11000 Beograd
Sarajevska br. 3
011/646-344

svi spektrumov restarti

Umetnost programiranja

Pri mašinskom programiranju procesora Z80 pozivanje potprograma se vrši instrukcijom CALL. No, ovaj procesor obezbeđuje još jednu instrukciju kojom se može izazvati izvršenje određenog potprograma. To je tzv. restart (RST). Prednost ove instrukcije je što u programu zauzima svega 1 bajt (u poređenju sa CALL — 3 bajta), pa samim tim zahteva manje vremena za njeni izvršenje. Ali, ovom instrukcijom je moguće pozvati samo potprogram koji počinje na jednoj od sledećih adresi: @/H/BB, 1/H/1gh, 2/H, 28H ili 38H. Odmah premećujemo da se ove adrese odnose na najniži memoriski prostor i to na tzv. nullu stranicu (zero page) memorije, tačnije prvih 256 bajta. Pogled na memorisku mapu Spektruma uveriće nas da u ovom računaru svih 8 restart adresa „pripadaju“ ROM-u. Tako na njima počinju neki potprogrami operativnog sistema. Naravno, prostor između samih restart adres je svište mali za 10le ozbiljniji potprogram, pa obično ovi potprogrami započinju skokom ili pozivom glavnog dela potprograma koji se nalazi na nekom drugom mestu u memoriji.

Pri kreiranju jednog složenog operativnog sistema, kao što je u Spektrumu, vodi se računa o tome da se na restart adresama postavite potprogrami koji se u toku rada najčešće pozivaju. To omogućava da se u glavnem delu operativnog sistema vrlo jednostavno, trošći jedan bajt memorije, dake RST instrukciju, pozove neki od tih potprograma. Štaž ovaj lukuš je ne bismo ponekada priuštili i sebi pribuši neki svoj mašinski program?

Pre svega, moramo upoznati tačnu funkciju svakog od ovih restart potprograma. Mi ne možemo menjati postojeći program u ROM-u (osim ako neko nije raspolozio da piše sopstveni operativni sistem), pa se stoga, ukoliko želimo da koristimo njegove potprograme, moramo prilagoditi njegovim „ubicicama“.

Pogledajmo, stoga, redom, šta i kako rade pojedini restart potprogrami u Spektrumu.

RST 0: počni od početka

Poziv ovoga dela operativnog sistema ne postoji nigde u ROM-u. Razlog je jednostavan: po uključenju računara, procesor započinje izvršavanje instrukcija upravo od adrese @ — posle provere RAM-a i inicijalizacije sistema prestaje sa radom dajući na ekranu svima poznatu copyright poruku. To znači da bi instrukcija RST @ u nekom programu imala absolutno isti efekat kao da smo isključili i ponovo uključili računar. Stoga ova instrukcija ima smisla jedino ako je upotrebljena na kraju nekog programa, a zelimo da on bude uništen i sistem reinicijalizovan.

RST 8: prijavi grešku

Potprogram koji započinje na adresi @ ima punu ruku posla dok preko tastature unosite neki

bezik program. On je, naime, zadužen da izvrši normalizaciju steka i nekih sistemskih promenljivih svaki put kada interpretator nađe na grešku u vašem programu. Osim toga, on na pogodan način „doturi“ interpretator kod greške, tako da se kasnije ona i fiksualno prikaže.

U čemu je značaj normalizacije steka? Ukoliko pogledamo bilo koji složeniji mašinski program, vidimo da se on sastoji od manje ili više potprograma koji se pozivaju iz jednog glavnog dela. Ovi potprogrami mogu, takođe, da pozivaju svoje potprograme, itd. Ako radimo mašinski program koji, recimo, briše ekran ili tenu slično, nećemo imati potrebe da prijavljimo grešku. Međutim, ako pišemo program koji treba da prepozna određeno komande ili koji radi interaktivno, tj. zahteva od korisnika unesenje nekih podataka, moramo kontrolisati te podatke pri unosu i javljati određenom putom da je došlo do greške jer, recimo, sintaksu ili podaci ne odgovaraju. Šta da radimo ako greška nastupi u nekom potprogramu koji poziva drugi potprogram, a njega, opet, poziva. pretpostavimo, glavni program? Pri svakom uzasnoplom pozivu na stek se ostavljaju povratne adrese i jednostavno naredbom za povratak iz potprograma (RET) ne možemo da se tako lako vrati u mesto gde hoćemo. Ako vršimo direktni skok, (JP ili JR) u potprogram za „regulisanje“ grešaka posle svake greške imaćemo sve više pozicija na steku, (ostavljenih usled CALL instrukcija) i oni će se, naprosto, zagušiti, a program krahira.

Svih ovih i nekih drugih briga može naš, vrlo elegantno, osloboditi instrukcija RST @. Ona će pozvati rutinu iz operativnog sistema koja će vratiti stek u prvočitno stanje i izvršiti indirektni skok u naš potprogram koji će, dalje, regulisati grešku. Naravno, mi moramo na neki način „javiti“ ROM-u gde se naša rutina nalazi. To činimo tako što u sistemsku promenljivu ERR SP (23613) ubacujemo, recimo na početku rada našeg programa, adresu na kojoj se nalazi naš potprogram za prijavljivanje grešaka. Najjednostavniji način da se ovo uradi je da se na početku programa na stek ubaci adresa naše rutine i sadrži tek pointer (SP) odmah sačuvući u ERR SP. To bi, onda, izgledalo ovako:

```
LD HL, ERRENT    adresa rutine za greške
PUSH HL          idu na stek
LD (ERR SP), SP  sačuvaj stek pointer
```

Nakon ovoga dovoljno je da posete detekcije greške u programu jednostavno izvršimo RST @ i dalje rad se nastavlja u našem potprogramu za prijavljivanje grešaka, sa stekom na prvočitno adresi.

Ako u programu treba da prijavljujemo više različitih grešaka, onda iza instrukcije RST @ posavistimo jedan bajt koji će predstavljati kod određene greške. Posle izvršenja, ovaj kod će se, zahvaljujući potprogramu iz ROM-a, naći u sistemskoj promenljivoj ERR NR (23610). Naša rutina za greške će, tada, jednostavno preuzeti ovaj kod, prouanalizirati ga i prijaviti odgovarajuću grešku. Veoma je važno da po izvršenju ovog potprograma, ili na njegovom početku, vratimo

na stek njegovu adresu (PUSH instrukcijom) i bi RST @ i dalje mogao funkcionišati ispravno još nešto: ako hoćemo da po izvršenju mašinskog programa nastavimo rad u bezik sistemu koristimo RST @ na opisani način, onda ga ikakvog postavljanja u ERR SP treba sadržati u sistemskim promenljivim na adresi 23728, a pre povratka bezika vratiš joj originalni sadržaj. Ukoliko smo direlirali ERR NR, pre povratka u bezik sistem treba u ovu lokaciju ubacimo vrednost 255 @FF.

Za sve ove probleme, naravno, postoje mnoga druga rešenja, ali je verovatno najbolje koristiti potprograme iz ROM-a kad god je moguće. To će, pre svega, smanjiti veličinu našeg programa, čemu uvek treba težiti.

RST 10: ispiši karakter

Od svih ovih restart potprograma u ROM-programeru će, verovatno, najviše koristiti on koji započinje na adresi 10H. Njegova funkcija da izvrši ispisivanje znaka, dočurenog u akumulatoru procesora, na ekran ili štampač. Pod kom se ovde podrazumejuvaju ne samo alfaničari i grafički karakteri, već i svii kontrolni kodovi koji obezbeđuju Spectrumov set karaktera. Ovaj potprogram radi nešto sportivije, upravo zbog to što obuhvata sve ove znakove. Stoga, ako zahavamo maksimalno brzo ispisivanje na ekran, recimo u igrama, onda nam jedino preostaje da pišemo svoje rutine za ispisivanje. No, ako vremensko ispisivanje nije kritično, onda će naš RST 10 rešiti dosta problema, naročito oko Spectrumove „nezgodne“ video memorije.

Kao što smo već rekli, da bi smo ispisali jedan karakter, potrebno je da u akumulator stavimo ASCI kód karaktera i izvršimo RST 10H. Ali, u praksi stvari zahtevaju malo više pažnje i pribrojno da. Pre svega, pre bilo kakvog ispisivanja, ROM-u treba javiti gde vršimo ispisivanje, glavni deo ekранa, donje dve linije ekran-a štampač. To činimo tako što u akumulator ubacimo 2 za glavni deo ekran-a, 1 za donji deo ili 3 ispis na štampač, te zatim pozovemo potprogram na adresi 1601H. Ovo je potrebno uraditi da bi se zoni kanalskih informacija (channel information) u RAM-u postavila određena izlazna adresa za ispisivanje. Nakon toga, ispisivanje će se vrati gde smo odredili, sve dok ponovno pozovemo rutinu na adresi 1601H ne promenimo postječe stanje.

Kao što je već rečeno, na raspolaženju nam suvi kontrolni kodovi (AT, TAB, kodovi za boje, poziciju kurzora, itd.), kako je predstavljeno u uputstvu za Spectruma na stranici 183. Po kontrolnim kodova za INK i PAPER, treba da se koduje @—7, a za FLASH, INVERSE, BRIGHT i OVER 1 ili @, zavisno od toga da li „uključimo“ ili „isključimo“ određenu funkciju. Iza ih i TAB moramo dati dva parametra: za AT istovno kao u beziku, dok TAB kontrola zahteva da sledeći podatak bude stvarni TAB parametar, da je drugi podatak nebitan, ali ga moramo „splatiti“.

Mašinsko programiranje na računaru ne može se ni zamisliti bez dobrog poznавања operativnog sistema. Vecina rešenja koja su ljubitelju računara potrebna čak i za najjednostavniji program već se nalazi u ROM-u — na programeru je samo da ih „pronađe“ i iskoristi na svoj način. Teško je naći kućni računar koga prati tako obimna, raznovrsna i detaljna dokumentacija kao ZX Spectrum. Pa ipak, njegov operativni sistem je još uvek pun tajni i one koji se spremaju da zarone u njega sasvim sigurno očekuju prijatna iznenadenja. Za prikaz Spectrumovog ROM-a i opis svih rutina u njemu teško da bi bilo dovoljno i čitavo ovo izdanje. Zato prikaz ograničavamo na osam najvažnijih rutina, koje u ROM-u zauzimaju sasvim posebno mesto i koje računar u toku rada najčešće priziva.

Zbog greške u ovom potprogramu ROM-a, naročito moramo obratiti pažnju da ne dozvolimo „ispis“ kontrolnog koda 8 (cursor left) u momentu kada su koordinate za ispis podešene na 0/0 (gornji lev ugađ ekranu), jer će se desiti vrlo čudno stvari: autor ovog teksta se dešavalo da bežijk program bude oštećen, a moguć je čak i krah programa. Na ostalim pozicijama nema nikakvih problema. Kao i za svaki pojedini znak, za svaki kontrolni kód i njegove parametre potreban je poseban RST 10H. Naravno, u programu čemo to ostvariti pomoću petlji ili sličnih rešenja.

Za one koji su zainteresovani za „izlazak“ iz ROM-a, ovo se pruža šansa za tako nešto. Pošto RST 10H je pruža u samu rutinu za ispis karaktera traži njenu adresu u RAM-u, mi bismo mogli na to mesto u RAM-u (channel information) „povući“ adresu naše rutine i na taj način preuzeći daljnju kontrolu. Tako bi smo sasvim moguće na mesto neiskorišćenih kontrolnih koda dodata neke nove, na primer za brišanje ekranu, automatsku tabulaciju, i slično. Posle provere kóda, ukoliko nije „naš“, izvršili bi smo skok u rutinu u ROM-u (počinje na 0xF4H) na odgovarajuće mesto, pa bi ona dajše dovršila posao. Ovo je sasvim izvodljivo ako pozivamo RST 10H iz našeg mašinskog programa. Stvari postaju bitno komplikovanije ako hoćemo da na RST 10H iskačemo u našu rutinu za vreme normalnog rada bežijk interpretera. Razlog za to je što jedna od rutina za brišanje ekranu u ROM-u direktno postavlja u RAM adresu potprograma za ispis (0xF4H). Znaci, ako mi izmenimo adresu na koju prelazi RST 10H, već prvo brišanje ekranu uništice našu adresu. Protiv ovoga se, donekle, može „boriti“ tehnikom interupta, o čemu će, ako bude interesovanje, biti reči drugom prilikom.

RST 18 i RST 20: uzmi sledeći karakter

Ova kratka potprograma na adresama 18H i 20H bežijk interpter intenzivno koristi za vreme izvršavanja bežijk programa. Funkcija im je da, shodno sistemskoj promenljivoj CH ADD (3645), u kojoj se čuva adresa do koje je interpter stigao analizirajući liniju bežijk programa, vrati interperetu kód karaktera koji je adresiran preko CH ADD (RST 18H), odnosno kód prvog sledećeg karaktera (RST 20H). Eventualna primena ovih restart rutina bi mogla biti u mašinskim programima koji na neki način „nadziru“ bežijk interpter, ili u eventualna proširenja postojećeg bežjika u Spectrumu.

RST 28: počni da računaš

Na adresi 28H počinje Spectrumov kalkulator. Kao što je poznato, on omogućuje rad sa pokret-

nim zarezom i obezbeđuje Spectrumu sasvim pristojan set matematičkih funkcija. Rutine kalkulatora zauzimaju dobar deo ROM-a, no uobičajeno je da se pozovi kalkulatora vrši instrukcijom RST 28H. Iza ove instrukcije treba da sledi suksesivno bajti koji će odrediti kalkulatoru „šta da radi“, sve do bajta (38H) koji određuje kraj rada kalkulatora. Izvršavanje programa se dalje nastavlja od sledeće instrukcije.

Sve funkcije koje su na raspolažanju u bežijk programima. Dobro upoznavanje ovog kalkulatora može biti od ogromne koristi svima koji su zainteresovani za rešavanje složenijih matematičkih problema u mašinskim programima. Kalkulator raspolaže sa 66 operativnih funkcija i 5 konstanti. Izračunavanje nekog matematičkog problema se svodi na to da se on „razbije“ na takve delove koji odgovaraju operativnim funkcijama kalkulatora i zatim reši u suksesivnom izvršavanju ovih funkcija. Table i objašnjenja u ovih funkcija i konstanti bi nam odnеле previše prostora u ovom tekstu, te preporučujem citacima koji su zainteresovani za ove probleme sledeće knjige: „Understanding your Spectrum“ od Dr. Iana Logana i „The Complete Spectrum Rom Disassembly“ od istog autora.

RST 30: napravi malo mesta

Namena ovog kratkog potprograma, koji se inače vrlo često poziva iz operativnog sistema, jeste da u određenom prostoru u RAM-u (tzv. work space) „napravi mesta“ za onoliko bajta koliko mu se doturi u BC paru registara. S obzirom da funkcioniše preko određenih sistemskih promenljivih, čini mi se da ovaj potprogram sašda nema neku „upotrebnu vrednost.“

RST 38: prekini i očitaj tastaturu

Restart potprogram na adresi 38H se bitno razlikuje od svih do sada spomenutih: specifičnost lokacije 38H je u tome što procesor, ukoliko mu je naredeno da radi u režimu interpara (interrupt — prekid) i to u modu 1 (IM1), svakih 20 milisekundi (dakle 50 puta u sekundi) prekida izvršavanje programa. „zapamtiti“ gde je stao i prelazi upravo na adresu 38H, da odatle izvrši potprogram koji na njoj započinje. Kada ga završi, vratiće se na glavni program i to tačno na mesto na kom je bio prekinut. Iz ovoga proizlazi da, u stvari, radi dva programne paralelno: glavni, koji interperira bežijk, itd., i drugi na adresi 38H. Postavlja se pitanje šta to radi taj potprogram kad mu je već dodeljena „čas“ da radi upredo s bežijk interperetom? Odgovor je: očitava tastaturu i „pokreće“ Spectrumov interni časovnik. No, svakako, glavni zadatak mu je da očita tastaturu i u sistemskoj promenljivoj LASTK (2356) ubaci kod tipke koja je eventualno pritisnuta. Očitavanje tastature se ne vrši pomoću ovog potprograma za vreme izvršavanja bežjika

naredbi LOAD, SAVE, i BEEP zbog isključenja interpara, mada SAVE i LOAD imaju svoje instrukcije za direktno očitavanje tipke BREAK.

Što će naših mašinskih programa koji koriste tastaturu, sve dok ne naredimo isključenje interpara instrukcijom DI (disable interrupts), u sistemskoj promenljivoj LASTK čemo dobiti kod pritisnute tipke uz prethodnu proveru kontrolnog bita u sistemskoj promenljivoj FLAG5 (2361) — ukoliko je bit 5 jednak nuli nikakva tipka nije pritisnuta, a ako je jedanak jedinicama možemo da preuzmemo kod tipke iz sistemskoj promenljivoj LASTK. Jedno takvo tipično očitavanje bi izgledalo ovako:

WAIT HALT: sačekaj interpar
BIT 5, (Y+1); tipka pritisnuta? (Y=5C3AH)
JR Z, WAIT: ne — skok nazad
RES 5, (Y+1); mala preostrožnost
LD A, (LASTK); uzmi kod tipke
RET; izlaz.

(Naredbe HALT se može izostaviti. Labela WAIT tada ide na BIT 5, (Y+1).)

Ako nam je za rad programa neophodno isključenje interpara (naredbom DI), moraćemo sami da očitavamo tastaturu. Ukoliko se radi o igri, gde se očitavaju 3—4 tipke, to ćemo uraditi direktnim očitavanjem sa „porta“ pomoću IN instrukcija. Lepo objašnjenje ovog postupka je dat u knjizi „Spectrum machine code for absolute beginner“. Međutim, ako nam je potrebna kompletna tastatura, možemo pozivati, kad je to potrebno, potprogram iz ROM-a na adresi 02BFH ili, još zgodnije, sa RST 38H, s tim da odmah sledi naredba DI, pošto pri izlazu iz rutine na adresi 38H postoji naredba EI (Enable Interrupts) za omogućenje prekida. Posle ovakog očitavanja, kod eventualno pritisnute tipke biće postavljen u LASTK, a bit 5 u FLAG5 će javiti da je tipka zainteznuta. Evo primera očitavanja tastature u tom slučaju:

READ RES, (Y+1); mala preostrožnost (Y=5C3AH)
CALL 02BFH; poziv potprograma iz ROM-a
BIT 5, (Y+1); dalje je neka tipka pritisnuta?
JR Z, READ: ne — skok nazad
LD A, (LASTK); uzmi kod tipke
RET; izlaz.

•CALL 02BFH može biti zamjenjen sa RST 38H s zatim DI.

U slučaju da pišemo program koji radi u režimu interpara u modu 2, možemo u servisnu rutinu, koja se izvršava po nastupu interpara, ubaciti RST 38H i na taj način imati očitavanje tastature — kao u modu 1 (normalan rad bežijk interperetator i operativnog sistema), pošto interpar u modu 2 nastupa u istim vremenskim intervalima kao u modu 1 (svakih 20 ms).

Ovo bi bilo, u kratkim crtama, ono osnovno što moramo znati o „nultoj stranici“ Spectrumovog ROM-a i njegovim restart potprogramima da smo ih uspešno koristili u našim programima. Za detaljniju analizu neophodan je disassembling ROM-a, a od velike pomoći mogu biti i knjige na ovom temu već spomenute u tekstu.

Milan Pavićević

UNIS**UNIS-RO promet birosredstava (p.o.)-Sarajevo****Olympia International**

INFORMATIONS-UND KOMMUNIKATIONS-SYSTEME

**olivetti****ELEKTRONSKI SISTEMI ZA PISANJE**

su na raspolaganju za oba, jednostrani ili dvostrani tok podataka.

Kontrolna elektronika: Kontrolna elektronika sastoji se od tri pod-montaže: 1. Kontrolor stampača
2. Kontrolor modaliteta
3. Interface**Interface:** IEC bus (слушаč)
(kako izaberete) IEEE 488/ANSI MC 1.1.

Centronics kompatibilni, 7(8) bits

• EIA RS 232 C (CCITT V 24)

TTY 20 ma current 100p, 50-19.200 Bauda

Snabdjevanje energijom:(kako izaberete) 220V 50Hz 110/220V 60Hz
240V 50Hz 110...127V 60Hz
110/220V/50Hz 120V 60Hz
100V 50Hz 100V 60Hz

Sekundarno: UMG = +36V

UMK = +9V

Preko regulatora volatage: UDD = +5V

Osigurači: Primarni: 1 termički osigurač (115 st. C)
Sekundarni: 1 x 1.6 amp sporo-eksplodirajući
1 x 4.0 amp sporo-eksplodirajući**Potrošnja struje:** U radu — 70 watti

U pripremi — 35 watti

Sigurnosna stipulacija: * Odgovara Evropskoj
sigurnosnoj stipulaciji BSI,
VDE, SEV, NEMKO itd.** Odgovara internacionalnoj
sigurnosnoj stipulaciji,
npr. IEC 380*** Odgovara prekomorskoj
sigurnosnoj stipu
lacijsi CSA, UL.

Stampač:

Stampač sa lepe
zom (daisy wheel), 96
znakova, pozicioniranje šta
mpača i lepeze pomoću koraka
motora. Brzina štampanja 17 zna
kova/sekunda.**Metoda štampanja:** Automatsko naprijed/na
zad, automatsko potiskivanje funkcije medu
prostora.**Stilovi štampanja:** Masni tip, prostorno kucanje, dupli udar**Razmak slova (pitch):** Standardni stilovi 1/10", 1/12", 1/15",
proporcionalni pismo 1/60", osnovno kre
tanje 0.42 mm.**Dužina linije:** 141 znakova sa razmakom 10 (pitch)
169 znakova sa razmakom 12
212 znakova sa razmakom 15**Lista konstrukcija:** ASCII standard i sekvenca izbegavanja za
kontrolu stampača**Mehanizam trake za pisanje:** Specijalne kaseteRučni izbor priprema indige i karbon trake,
automatski izbor multi-udarne karbon trake,
štampanje u dva nivoa.Standardna karbon traka, lift-off karbon
traka, dužina 130 m, širina 13 mm, broj
udara 100.000**Tehnički podaci**

Tip mašine: Tip 103 je mikroprocesor
sorom kontrolisana output pisača mašina sa
interface-om. Dodatno funkcijama pisače nisu
ne, ova mašina može biti kao output ili input/ou
put uredaj u on-line modalitetu. Ukupno 3 interface-a

**ELECTRONSKA
PISAČA MAŠINA
OLYMPIA ESW 103**

CELARIJE BUDUCNOSTI

Multi-karbon traka, dužina 115 m, širina 13 mm, broj udara 328.000
 Tkanena traka, crna, dužina 9 m, širina 13 mm, broj udara 1.080.000
 Tkanena traka, crno/crvena, dužina 9 m, širina 13 mm, broj udara 270.000

Tastatura: Tastatura sa 48 tastera tip TUMBLER. Svi tasti sa funkcijom ponavljanja osim tastora akcenata. Tasteri funkcija: povrat štampača sa i bez prelaza na sledeći red, prostorni taster, polukorak, povrat za jedno mesto, ispravka, taster za pomeranje i fiksno pomeranje.

Kontrole desnog dela tastature: tasteri za desnu i levu marginu, taster ON/OFF-LINE tastor tabulatora, prostor medju redovima i taster za razmak (pitch) slova, tasteri za napred/nazad vertikalno pozicioniranje valjka polukorak, preko funkcije ponavljanja.

Kontrole levog dela tastature: oslobađanje margine, taster tabulatora (tabuliranje unazad), taster šifre (code), taster ponavljanja.

Kontrole u kombinaciji sa tastorom šifre: mašna slova, prostorno pisanje, dupli udar.

Indikator lampa: ON/OFF-LINE, greška, masna slova, prostorno pisanje, dupli udar.

Valjak: Dimenzije valjka: dužina 43,5 cm prečnik 44,5 mm

Buffer: 2K i 4K byte buffer podataka u ON-LINE operacijama

Memorija ispravki: 256 byte (1 linija) u OFF-LINE operacijama

ELEKTRONSKI SISTEM ZA PISANJE ETS 1010

Ukoliko već posedujete elektronsku pisanu mašinu Olivetti već ste načinili prvi korak u elektronski kancelarijski sistem Olivetti. Vašu elektronsku pisanu mašinu možete koristiti sa savremenim uređajem sa ekranom.

ETS 1010 ima CRT display sa 25 linija, dve pogonske diskete sa 360 K-bajta memorije i mogućnošću povezivanja sa E121.

Za upis informacije jednostavno otkucati tekst (tastatura ima tradicionalni izgled). Informacija se prikazuje na displeju. (U ovoj fazi teksta na mašini se ne kuca), a kada se završi upis ili izmena teksta, štampa se končan tekst brzinom od 20 znakova u sekundi. Ono što vidite na ekranu-doterani tekst, masni slog i podvučena slova sada dobijete otkucano na papiru.

Lako je naučiti i raditi sa ETS 1010.

ETS 1010 olakšava prelaz na upotrebu automatske uključivanjem u elektronsku pisanu mašinu OLIVETTI i održava blisku i komandnu vezu sa korisnikom. Mašina je još uvek najefikasnije sredstvo za pisanje kratkih obavesti, indeksnih kartica, novinara itd.

ETS koristi tehniku laganog pomeranja teksta. Kada se na displeju pomeri teksta ne vrše se preskoci od jednog reda u

određenom vremenskom periodu. Umesto toga, kreće se polako i omogućava jednostavno čitanje. Da bi se olakšala upotreba ETS ima funkcije UNDO i HELP. Ukoliko nešto radite (npr. umetanje paragrafa na pogrešnom mestu), a zatim

učete grešku nemojte brinuti, jednostavno ponistite to (UNDO). ETS 1010 će sve dovesti u red.

Ukoliko niste sigurni u vezi sa funkcijom ili šta treba da iđe da radite tražite pomoc (HELP). 1010 će vas provesti kroz sve funkcije. ETS 1010 takođe poseduje široki opseg novih funkcija za pomoć u upravljanju.

Vanserske funkcije.

Jedinstvena sposobnost vizuelnog sistema u ETS 1010 predstavlja novu standardnu odličku za lakšu obradu faksata, naslova i brojeva kao predmetaka kao i za prenošenje informacija od jednog dokumenta na drugi.

Vizuelni sistem omogućava da se ekran podeli na dva ili više površina sa protok informacija. Veći dokumenti (duži od 80 kolona ekranâ) se obraduju u mnogim sistemima horizontalnim pomerjanjem. Jedini problem je u tome što se gubi tekst. Upotrebom kapaciteta prosledivanja u sistemu ETS 1010, leva kolona se može „zamrznuti“, dok se preostali deo dokumenta kreće.

ETS 1010 pomoć u administrativnom rukovodstvu.

ETS 1010 poseduje „ladiće za dokumentaciju“ i „registre“ za vaše dokumente. Možete staviti dokument u arhiv ili jednostavno referencirati prema položaju dokumenta. Tehniku vodenja dokumentacije je lako razumeti jer ona obezbeđuje brzu dostupnost listi svih ladića za dokumentaciju, registrima i daktiografima kao i obaveštenje o datumu nastanka, autoru, daktiografu i određenom klijenu. Identifikacioni sistem za automatsku reviziju štedi vreme administriranja što ubrzava korekturu. Ukoliko je potrebno obaviti ovu funkciju ETS 1010 može emitovati ili štampati izbrisane reči precrteane linijom, ili dodate reči sa crtom iznad njih.

Sa ostalim vanserskim elementima ETS 1010 će uskoro pomagati u proračunima, planiranju budžeta i periodičnim proračunima i pripremi izveštaja.

ETS 1010 za pouzdanost

Kvalitet i pouzdanost su oznake izvedbe ETS 1010. ETS 1010 može podneti pražnjenje statičkog elektriciteta koji se često javlja zbog suve klime i zagrejanih prostorija. Ukoliko je pad napona ili nestanak struje u radnim prostorijama čest, za zaštitu od uništenja informacija služi baterija (po izboru kupca).



SVE DETALJNE INFORMACIJE KOD ZASTUPNIKA ZA JUGOSLAVIJU

UNIS

UNIS-RO promet birosredstava (p.o.)-Sarajevo

UDRUŽENA METALNA INDUSTRija
SARAJEVO

Direktorija 45 d, 71000 Sarajevo
telefon: 071/ 516-355
39-342

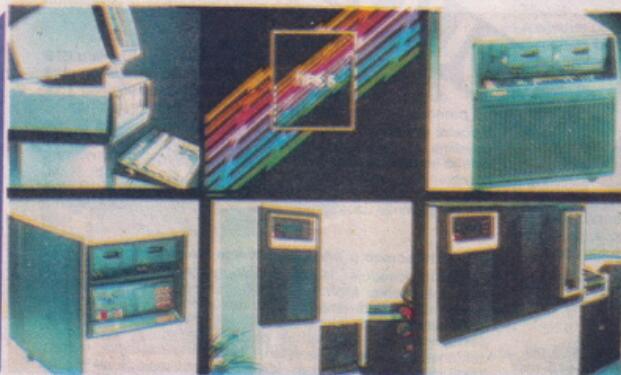


Honeywell

INFORMACIONI SISTEMI



PROGRE



Porodica računara „EI-HONEYWELL”

„EI-HONEYWELL” NUDI OPREMU PO POTREBAMA I MOGUĆ-
NOSTIMA KORISNIKA, ZATO, POTRAŽIMO REŠENJE ZAJEDNO!



Računski centar sa sistemom „EI-HONEYWELL” DPS 6/95 Karakteristike tog modela su:

- najpoznatiji model iz serije EI-HONEYWELL DPS6
- 32 — bitni procesor visokih performansi za komercijalne i naučne obrade
- memorija kapaciteta 2 MB do 16 MB
- mogućnost priključenja do 112 terminalskih radnih jedinica i/ili komunikacionih linija
- kapacitet diskova do 3,6 GB (giga bajta)

ELEKTRONSKA INDUSTRIJA NIŠ
RO „EI-RAČUNARI”
OUR „EI-HONEYWELL”
INFORMACIONI SISTEMI
18000 NIŠ, Srbija, Tasmajdanska 85-82
tel. 018/332-342, 334-098
telex: 16296 YU EI-HIS

Ei Honeywell
INFORMACIONI SISTEMI
SEKTOR MARKETINGA
11000 BEOGRAD, Terazije 14
tel. 011/345-442, 345-443
telex: 11837 YU EI-HIS

OUR „PROGRES-INFORMATIKA”
11000 BEOGRAD, Mašinistska 5/2/VII
tel. 011/345-442, 345-443
tele: 11820 YU PROG
Zavod za razvoj i razvoj
41000 ZAGREB, Štefana 4
tel. 041/44-7478
S1000 LUBLJANA, Slovenia, Trg 4
tel. 061/224-783

**EI — HONEYWELL
INFORMACIONI SISTEMI**

„EI-HONEYWELL” informacioni sistem je preduzeće sa zajedničkim ulaganjem (JOINT VENTURE) američke firme „HONEYWELL INFORMATION SYSTEM” Inc. sa sedištem u Mineapolisu, poznatog svetskog proizvođača računara, **ELEKTRONSKA INDUSTRije**, Niš jednog od najpoznatijih preduzeća u oblasti elektronike i digitalne tehnike u Jugoslaviji i „PROGRESA”, OUR „PROGRES-INFORMATIKA”, Beograd, kao generalnog zastupnika i distributera opreme HONEYWELL, za koju pruža korisnicima svu stručnu pomoć, i kao takvo preduzeće bazirano na uspešnoj saradnji 3 partnera, obavlja sledeće delatnosti:

— **Proizvodnja** DPS6 modela mini, srednjih i velikih računara, video-terminala i serijskih štampača iz proizvodnog programa Honeywell-a

— **Plasman** mini, srednjih i velikih računara na jugoslovenskom tržištu

— **Export** mini, srednjih i velikih računara u više od 50 zemalja sveta, posebno u nesvrstane zemlje i zemlje u razvoju

— **Projektovanje** informacionih sistema

— **Razvoj** aplikativnih paketa

— **Distribucija** sistemskog softvera i aplikativnih paketa

— **Instaliranje**, održavanje i poslove field-engineering-a

— **Obuku** korisnika računarskih sistema.

Svi računari se proizvode pod znakom „EI-HONEYWELL”, koji garantuje kvalitet, proizvoda na svetskom nivou, sa automatskim prenetim novim znanjima i otkrićima samog „HONEYWELL”-a

Danas na 5-godišnjicu potpisivanja JOINT VENTURE Ugovora, možemo se pohvaliti da smo sa otvaranjem nove fabrike u Nišu (1. maj 1981) uspeli da zadovoljimo velike potrebe jugoslovenskog tržišta, i dosada plasiramo preko 300 sistema, koji su dobili mnoge pohvale i priznanja od strane zadovoljnih korisnika.

NOVO U IZDANJU
IRO „VUK KARADŽIĆ“

ELEKTRONSKI RAČUNAR

Most u budućnost

SVE ŠTO STE
HTELI DA ZNATE
O RAČUNARIMA,
PROCITAJTE
U OVOJ KNJIZI

Elektronska obrada podataka, kao polje aktivnosti savremenog čoveka, i elektronski računar, kao tehnološki proizvod koji omogućava

tu aktivnost, poslednjih godina se izuzetno brzo razvijaju i šire.

U svakodnevnom životu su sve prisutniji i imaju sve veću ulogu.

Literatura iz ove oblasti je u svetu veoma brojna i bogata. Kod nas, na žalost, to nije slučaj. Postoji samo manji broj udžbenika i priručnika okrenutih prvenstveno programskim

jezicima i teorijskim osnovama računara namenjenih studentima.

Nedostaju tekstovi koji daju celovit prikaz računara i njegovih mogućnosti na popularan, iako dosledno stručan način dostupan svima čije je poznavanje matematike i tehnike na elementarnom nivou.

**OVA KNJIGA TREBALO BI DA
POPUNI OVU PRAZNINU**

Latinica,
130 strana,
format: 165 x 22cm,
60 fotografija i ilustracija,
tvrd, plastificirani povez.

Cena: 900 din.

Stanko, Popović

elektronski računar

most u budućnost



Vuk Karadžić

Izdavačka radna organizacija „Vuk Karadžić
Beograd, Kraljevića Marka 9, tel. 620-371

NARUDŽBENICA RAČUNARI II

Nekopivo! Naručujem knjigu ELEKTRONSKI RAČUNAR po ceni od 900— dinara. ZA GOTOVO — sa popustom od 20%. Plaćanje poštara primena prijema knjige. NA OPTPLATO u 2 meseće rate. Rate će otpisivati redovno svakog meseča, prema rata i važećim zakonskim propisima.

(Nepotrebno prečišći)

U slučaju sporu priznajem nadležnost Opštinskog suda u Beogradu.

Prezime, očevce ime i ime

Mesto i godina rođenja

Zanimanje — Podt. broj i mesto stanovanja

Ulica i broj

telefon

Naziv i adresa radne organizacije

telefon

Overa o zapošljenu za kupce na otpлатu

Potpis kupca

Br. i Kartice izdate od SUP-a

Pensionom prilaže pretposlednji ček od penzije



Iskra Delta



računalniški sistemi delta



RO ISKRA DELTA je proizvođač kompletnih računarskih sistema sa uhodanim razvojem i proizvodnjom mašinske opreme, sistemske i aplikativne programske opreme, sa organizovanom prodajom, školovanjem, održavanjem i inženjeringom.

Osnovna konцепција RO ISKRA DELTA temelji se na proizvodnji računarskih sistema i perifernih jedinica sa maksimalnom primenom domaće tehnologije i znanje, a na osnovu najnovijih svetskih dostignuća u ovoj oblasti.

Iskra Delta proizvodnja računarskih sistema i inženjering LJUBLJANA, Parmova 41
Telefon 061/312-988 h.c.
Telex 31366 YU DELTA