izdaje BIGZ OOUR „Duga"

računari

specijalno izdanje časopisa „Galaksija"

izlazi jedanput mesečno
cena 300 din
maj 1986.

15

uputstvo za upotrebu

profi asembler 64

periferijska oprema

novi matični štampači

ekskluzivno

svi „atarijevi" program

kad se zakon uspava

programe bez zaštite

umetak na 32 strane

dejan ristanović

mašinac za početnike

časopis za prave program

15

računari

Izlazi jednom mesečno
Izdaje BIGZ OOUR „Duga"

Cena 300 dinara / maj 1986.

Specijalno izdanje
časopisa „Galaksija"

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
Dobroslav Petrović

Direktor OOUR „Duga"
Bratoljub Babić

Glavni i odgovorni urednik
Gavrilo Vučković

Urednik izdanja
Jova Regasek

Tehnički urednik
Mirko Popov

Redakcija časopisa „Galaksija"
Tanasije Gavranović, pomoćnik glavnog i odgovornog urednika, Esad Jakupović, zamenik glavnog i odgovornog urednika, Aleksandar Milinković, urednik, Jova Regasek, urednik, Zorka Simović, sekretar redakcije, Srđan Stojančev, novinar, Gavrilo Vučković, glavni i odgovorni urednik

Stručna saradnja
Dejan Ristanović
Dušan Slavić
Nevenka Spalević
Anđelko Zgorelec

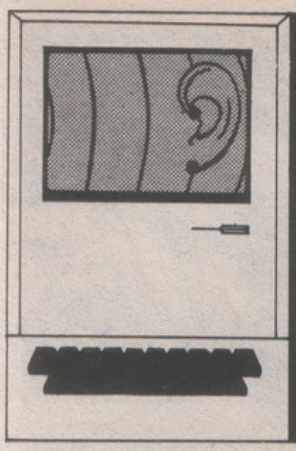
Stalni saradnici
Nada Aleksić, Ninoslav Čabrić, Branko Đaković, Donat Greber, Voja Gašić, Branko Hebrang, Đorđe Janković, Mihajlo Karapandžić, Vladimir Kostić, Vladimir Krstonošić, Saša D. Kovačević, Srđan Kosovac, Radomir A. Mihajlović, Zvonimir Makovec, Blažimir Miše, Dejan Muhamedagić, Ivan Nador, Radomir Nikolajev, Zoran Obradović, Miodrag Potkonjak, Dejan Ristanović, Jelena Rupnik, Dušan Slavić, Jovan Skuljan, Nevenka Spalević, Srđan Stakić, Zvonimir Vistrička, Anđelko Zgorelec, Zoran Životić

Fotografije
Vladimir Simović
Ilustracije
Miodrag Marković

Izdavački savet „Galaksije"
Dr Rudi Debijađi, prof. dr Branislav Dimitrijević (predsednik), Radovan Drašković, Tanasije Gavranović, Živorad Glišić, Esad Jakupović, Velizar Maslać, Nikola Pajić, Željko Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Milorad Teofilović, Vidojko Veličković, Velimir Vesović, Milivoje Vuković

Štampa
Beogradski izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17
Žiro-račun kod SDK 60802-833-2463
Devizni račun kod Beobanke 60811-620-6-82701-999-01066
Za inostranstvo cena dvostruka (400 D, 2,50 US$, 6,50 DM, 45 Sch, 5,50 Sfrs, 20 Ffrs)
Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika" broj 26/72, ovo izdanje oslobođeno je poreza na promet.

sadržaj



Šta ima novo

„Elektron" je Ljubišin

U nagradnoj igri „Elektron je vaš", koju je, specijalno za čitaoce „Računara", organizovao „Partizan" iz Čačka, zastupnik firme Acorn za Jugoslaviju, najviše sreće imao je ***Ljubiša Petrović*** *iz* ***Niša, Nikole Kopernika 13.*** *Ljubiša je, razume se, znao da su osnivači firme Acorn Hauzed i Kari, da procesor 65C12 ima više mašinskih instrukcija od procesora 6502 i da se bočni RAM kod računara iz familije BBC najviše koristi za upisivanje sadržaja aplikativnih programa koji se obično prodaju u ROM-u, pa je na tri umereno laka pitanja Dejana Ristanovića odgovorio sa b), c) i a).*

Ljubišu je, iz gomile od nekih 2000 pisama i dopisnica, izvukla Jelena Rupnik (ruka joj se pozlatila!) na prvoj sednici spoljne redakcije „Računara". Jelena je odbila da se slika čak i u ovoj prilici, pa čak i iza dopisnice, pa ovu lepu vest nismo mogli da propratimo i prigodnom fotografijom. Naš foto-reporter Vladimir Simović je, srećom, među devojkama koje je ovo varljivo proleće izmamilo na ulice, pronašao i jednog ljupkog ljubitelja Acorn računara i zamolio ga za jedan studijski snimak. Ljubiši Petroviću, dakle, „elektron" sa štampačem „šinva CP80", a ostalim znalcima BBC računara i učesnicima nagradne igre jedan pomalo zagonetan osmeh naše anonimne čitateljke.

Računari na granici

Početkom maja stupili su na snagu novi carinski propisi za uvoz kućnih računara. Prilikom povratka iz inostranstva sada se može uvesti kućni kompjuter u vrednosti od 90.000 dinara. U okviru ovog iznosa mogu se uneti i tzv. „posebne računarske jedinice" — periferijska oprema čiji je uvoz do sada bio veoma sporan. Poštom se mogu primati predmeti od 30.000 dinara.

Vrednost računara se obračunava prema tzv. statističkom deviznom kursu, koji se utvrđuje na početku godine i važi čitavih dvanaest meseci. Kurs za značajnije valute dat je u tabeli. Iz ovih odnosa lako je izračunati da je dozvoljen uvoz svih računara koji su jeftiniji od 340,2 američka dolara, ili 1.001,7 zapadnonemačkih maraka, ili 833,7 švajcarskih franaka, ili 650.289 italijanskih lira ili engleskih funti. Po ovom kursu obračunava se samo pravo na uvoz — carinske i ostale dažbine plaćaju se prema tekućem, dnevnom deviznom kursu.

Statistički kurs

američki dolar	=	264,53 din.
zapadnonemačka marka	=	89,84 din.
švajcarski franak	=	107,95 din.
funta sterlinga	=	343,30 din.
austrijski šiling	=	12,77 din.
francuski franak	=	29,42 din.
italijanska lira	=	0,13 din.

Iako još uvek prilično skromno odmeravaju jugoslovenske potrebe za kućnim računarima, novi propisi ipak obezbeđuju znatno povoljnije uslove od onih koji su važili do pre neki dan. Novi limit obuhvata kompletnu klasu 128K mašina, kompletnu MSX klasu, „komodor 128 D", najnoviju klasu NLQ matričnih štampača, najbolje disk-jedinice najkvalitetnije crno-bele monitore, a bračni par može, po pola, da uveze u delovima čak i nekog od IBM PC/XT klonova. Novi carinski propisi, dakle, više ne osuđuju Jugoslovene na „spektum" i druge mašine igračke. Od dobrih kompjutera i njihove korisne primene sada nas još jedino deli sve tragičniji YU standard.

Kompjuterski poliglota

Kada smo objavili vest o ruskom tekst procesoru, javilo se nekoliko zainteresovanih kojima je trebala adresa. Evo nečeg još zanimljivijeg: poliglotski tekst procesor koji radi na engleskom, ruskom, grčkom, hebrejskom i arapskom. Zove se „Multi Lingual Scribe" i može se dobiti od firme Gamma Productions, 710 Wilshire Blvd., suite 609, Santa Monica, California, USA. Da li ste primetili da su se u istom programu našli i hebrejski i arapski? Pravi internacionalizam! (*B.Đ.*)

Jedna vrsta zaštite

Večiti problem zaštite kompjuterskih sistema je toliko poznat da nema potrebe išta pričati o tome. Motorola je proizvela zaštitni uređaj za kompjuterske sisteme koji daje novu dimenziju zaštiti. Uređaj prima poziv, traži šifru i kada je dobije — prekida vezu. Tada u svojoj internoj telefonskoj beležnici traži telefonski broj koji ide uz šifru, naziva korisnika koji se ponovo predstavlja i tek se onda uspostavlja direktna veza sa računarom. Uređaj, osim ovoga, može da kontroliše i dužinu pojedinačnog rada nekog korisnika po unapred utvrđenom rasporedu. Osnovna verzija uređaja može da komunicira sa ukupno 400 korisnika. (*B.Đ.*)

Spoljna redakcija

Od broja koji upravo držite u rukama redakciji „Galaksije" u pripremanju specijalnog izdanja „Računari" pomaže i spoljna redakcija mahom stručnih saradnika: Branko Đaković, Dejan Ristanović, Jelena Rupnik, Jovan Skuljan, prof. dr Dušan Slavić, Nevenka Spalević i Zoran Životić.

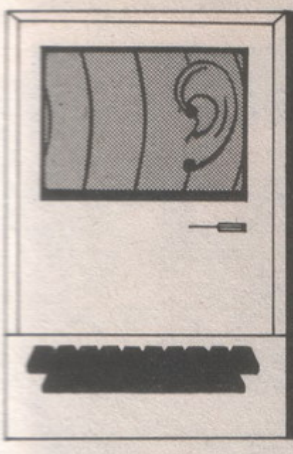
Nije sve tako crno

U poslednje vreme računari sve lakše nalaze put do naših škola. Nedavno je beogradskim školama kupljeno 100 kompjutera marke „orik nova", od ranije postoji dosta „lola" i „galaksija", a neke škole su se opremile i nekim većim mašinama. Ali, u isto vreme sve češće se čuju glasovi da se kompjuteri ne koriste na pravi način, da stoje zaključani, da se njima igraju deca profesora matematike, da učenici i ne znaju za njihovo postojanje.

Beogradska škola za usmereno obrazovanje „Beogradski bataljon", poznatija kao 13. gimnazija predstavlja sasvim suprotan primer. U računskom centru ove škole glavna zvezda je „amstrad 664" sa „epsonovim" štampačem, a pored njega tu su još jedan „spektrum", dve „lole" i dva „orika". U školi redovno radi kompjuterska sekcija, na kojoj svi učenici mogu da rade sa svim računarima, ili da nešto nauče od profesora i starijih kolega. Voleli bismo kada bi i ostale škole sledile ovaj primer i napravile svoje sekcije u saradnji sa profesorima, roditeljima i bivšim učenicima.

Drugi lep primer iz „Beogradskog bataljona" je primena računara, osim u obrazovanju, i u svakodnevnom životu i radu škole. U svoje slobodno vreme i na profesionalnoj praksi, učenici četvrte godine, u saradnji sa profesorom Miloradom Jovanovićem, napravili su nekoliko odličnih programa za školu, pomoću kojih se sređuju dnevnici, ocene, izostanci i sve što zahteva pedagoška administracija. Ovim je čitav postupak sređivanja dnevnika na kraju minulog polugodišta bio znatno olakšan i skraćen. Neka odeljenja su već dobila ocene sređene na računaru i odštampane na printeru. Osim ovog, napravljeni su programi za knjigovodstvo i proračun ličnih dohodaka. Primer „Beogradskog bataljona", verovatno, ne može bitno da ulepša inače dosta tužnu sliku u našem školstvu, ali ipak predstavlja veliko ohrabrenje.

V. K.



Šta ima novo

Flopi reklama

Oliveti je pronašao način da poboljša reklamu svojih proizvoda. Reklama će biti upakovana na flopi diskove i to tako što će se pod Olivetijevom oznakom prodavati diskete koje će na sebi imati demonstracione programe, a neće biti ništa skuplje od običnih disketa istog kvaliteta. Paket od deset upakovanih disketa sadrži pravu gomilu korisnog softvera, uključujući i tekst procesor, bazu podataka, supercalc program i slične poslastice. Ako ova akcija bude imala uspeha, izbor poklon softvera će verovatno biti proširen. To nas raduje, jel te? (*B.Đ.*)

Animator 1

Srećni vlasnici ***Artista*** *i* ***Art Studia****, čuvajte se! Na tržištu se pojavio novi program za crtanje koji obuhvata i sprajt-dizajner i program za animaciju. Ime mu je Animator 1 i proizvod je malo poznate softverske firme SoftCat. Samo crtanje je vrlo olakšano, a funkcije sa prozorima su izvanredno brze (****Art Studio,*** *stidi se!). Recimo, kod standardnih programa vi postavljate centar kruga, a zatim jednu tačku na obodu. Kod* ***Animatora 1*** *vam je sve vreme pred očima krstasti kurzor, koji se širi kako povećavate poluprečnik kruga, i tako pokazuje njegove buduće dimenzije.*

Jedanput nacrtanu sliku možete i animirati pretvaranjem pojedinih njenih delova u sprajtove. Inače, sam sprajt-dizajner je izuzetno dobar i jednostavan za korišćenje.

Sprajtovi se mogu snimiti na traku ili drajv i koristiti u bejziku ili mašinskim programima, što je sve detaljno objašnjeno u uputstvima za upotrebu.

Pliva heker preko Save, nosi sprajtić na vrh glave! ***(D.S.)***

Orao dobija krila

Varaždinski PEL nedavno je završio razvoj novog školskog računara, u koji je uloženo 15 milijardi dinara. Novi računar, izrastao iz „galeba" i „orla", dakle na procesoru 6502, ima profesionalnu tastaturu, profesionalni format ekrana (tekst 80×24, grafika 620×256 tačaka), 32 K ROM-a, 64 K RAM-a sa mogućnošću proširenja do 256 K, disk kontroler i dve disketne jedinice, a uskoro će biti opremljen i modulom za povezivanje u lokalnu mrežu. Računar je, praktično, prihvaćen kao standard za osnovne škole u Hrvatskoj, a preduzimljivi varaždinci smatraju da ima izvesnih šansi da bude uveden i u Srbiji i već se uveliko pripremaju da otvore jedan pogon u Beogradu. Za dalji razvoj ovog računara u Hrvatskoj je odvojeno 60 milijardi dinara — pola od toga za hardver a pola za softver — a sredstva će biti raspoređena preko jednog opštejugoslovenskog konkursa. „Nemamo ništa protiv da bilo ko, bez ikakvih posebnih uslova, proizvodi naš računar pod bilo kojim imenom," kažu u PEL-u, „i spremni smo da prihvatimo bilo čije rešenje kao svoje sopstveno ako je — bolje!"

Dolazi „amiga"

„Amiga" se, najzad, pojavila u Evropi. Možete je nabaviti u svim većim prodavnicama računara, a cena je onakva kakva je i bila obećana. Pojavilo se i nešto malo (nekoliko stotina) programa, što predstavlja dobar početak. Jedan deo programa je prerađen sa drugih računara, ali ima i dosta originalnih. Ohrabruje to što su se tog posla prihvatile sve veće softverske firme. Electronic Arts, jedna od najpoznatijih softverskih firmi, sada radi isključivo programe za „amigu", a pored nje tu su još i Activision, Synapse, Broderbund, Sublogic, Lattice, Mindscape, Metacomco i ostali, dok se za grafiku brine Islands Graphics. Pomenimo i nekoliko najinteresantnijih programa: C, dBASE III, „Turbo Pascal" (specijalna „amiga" verzija), „ABasic" (interpreter i kompajler), lisp, makroasembler, IBM emulator, „Unicalc" spredšit, „Textkraft" i još dosta toga. Pored ozbiljnijih programa, već su u prodaji programi za muziku, crtanje, kuvanje i, naravno, igranje! Prerađene su neke starije čuvene igre, ali postoje i mnoge nove, pisane upravo za „amigu". Ove potonje predstavljaju novi prodor u video igrama. Slike iz igara su prosto neverovatne za sve one koji imaju male kućne računare.

Tajnost niko ne garantuje

Ako ste mislili da niko ne može da sazna šta vi, zatvoreni u četiri zida, radite na svom ličnom kompjuteru, grdno ste se prevarili! Jedan holandski stručnjak usavršio je metod pomoću koga se iz blizine vaše zgrade može videti šta piše na ekranu vašeg monitora ili TV aparata. Potrebna je samo jedna direkciona antena i elektronske opreme u vrednosti od oko 15 dolara

Igrači(ce) „pakmena" i ostalih muškarčina o kojima smo pisali u prošlom broju, oprez! (*B.Đ.*)

Laserska jabuka

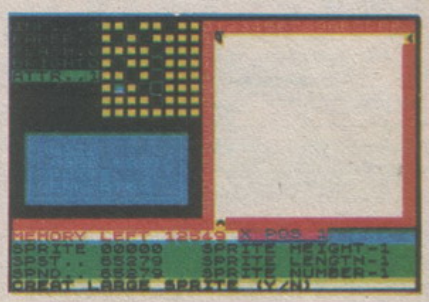
Oduvek ste želeli da imate "epl IIe" ili IIc ili bilo koji "epl"? Oduvek niste imali para za "epl"? Uobičajena stvar. Zato sada možete kupiti "laser". "Laser 128" je mašina na kojoj rade „skoro svi" programi napisani za "epl IIe" ili IIc. Košta samo 450 dolara i može se nabaviti kod Video Technology firme na 2633 Greenleaf Avenue, Elk Grove Village, Illinois, USA. Probajte. Možda rade baš „skoro svi" programi. (B.Đ.)

Laser Basic

Laser Basic *je nova varijanta jezika za „spektrum" okrenuta prvenstveno grafici i sprajtovima. Program zauzima oko 8 K na vrhu RAM-a i dodaje standardnom „spektrumovom" bejziku 138 novih komandi. Većina služi za kontrolu grafike, mada ima i vrlo korisnih — kao što su RENUMBER i TRACE rutina.*

Sprajtovi su standardne veličine 15×15 tačkica, ali mogu biti i veći. Unutar programa postoji i „skladište" već gotovih sprajtova, spremnih za upotrebu. Ima ih 109: od malih svemirskih brodova do velikih parnih lokomotiva.

Naravno, vaši programi obogaćeni grafikom iz ***„Laser Basica"*** *će raditi samo kada je on prisutan u memoriji. Zato su njegovi pisci izbacili na tržište Laser Compiler (posebno se plaća), koji jednom za svagda prevodi vaše programe u mašinac.*



Programerima koji su želeli da pišu svoje igre, a ***White Lighting*** *im je bio suviše komplikovan,* ***Laser Basic*** *će pomoći svojom jednostavnošću i efikasnošću.*

To piše i u sićušnom uputstvu od 75 stranica, koje naši pirati neće kopirati. ***(D.S.)***

Opus Discovery 1

Mnoge će iznenaditi kojom brzinom su pojedini proizvođači prateće opreme reagovali na novi „spektrum 128". To se, najpre, može pripisati činjenici da promene u samom „spektrumu" nisu korenite. Treba se samo prilagoditi dodatnom ROM-u i drugoj memorijskoj banci i — to je sve.

Firma Opus je poznata kao proizvođač najkompletnijeg periferijskog dodatka za „spektrum". To je 3.5 inčni disk drajv Discovery, koji u sebi sadrži još i paralelni interfejs za štampač, džojstik port, monitorski izlaz, konektor opšte namene i napajanje koje, ujedno, snabdeva i „spektrum".

Priključen na „običan" „spektrum", Discovery je otvarao mogućnost formiranja RAM-diska od gornjih 32 K njegove memorije, koji je tretiran kao drajv pet. Novi Discivery, priključen na „spektrum 128", utrostručuje ovu mogućnost, jer omogućava da gornju memorijsku banku (64 K) korisite kao drugi RAM-disk, koji će se tretirati kao drajv šest.

„Spektrum 128" poseduje jednu karakteristiku kojom baš ne može da se podiči: prelaskom u mod 128 iz moda 48, i obrnuto, briše se kompletan sadržaj memorije. Međutim, Discivery i to rešava. On, čak, omogućava i razmenu podataka između „manjeg" i „većeg" RAM-diska. Na žalost, ovako lepe mogućnosti moraju debelo da se plate: Opus Discovery 1 sa jednom jednostranim disk-drajvom kapaciteta 178 K košta 199,95 funti, a Discovery 2, sa dva drajva i ukupnim kapacitetom od 360 K, 329,95 funti.

Sic transit gloria mundi. (D.S.)

IBM opet prvi

IBM je proizveo čitavu novu klasu računara zasnovanih na novom 32-bitnom mikroprocesoru poznatom pod imenom RISC (kompjuter sa reduciranim setom instrukcija). Osnova te klase računara je IBM RT; RT je samo dvostruko skuplji od modela AT, ali je mnogo moćniji. Namenjen je CAD primenama u nauci i inženjerstvu. Novi procesor ima samo 40 za razliku od nekoliko stotina instrukcija kojima raspolažu procesori na velikim IBM-ovim računarima.

Operativni sistem RT-a je zasnovan na Unixu i zove se AIX. Računar je opremljen jednim hard diskom i jednim flopi disk drajvom.

Ovim svojim potezom IBM je pretekao Ejkorn koji je još prošle godine razvio svoj 32-bitni RISC procesor pod imenom ARM (Acorn RISC Machine), ali do sada nije uspeo da izbaci računar zasnovan na tom procesoru.

Smatra se da će ulazak IBM-a na tržište RISC procesora oživeti tržište i prokrčiti put za mnogo moćnije i mnogo jeftinije računare u bliskoj budućnosti.

Tvrda ploča

Utični hard disk iliti hard pločica stigao je i do Evrope. Posle intenzivne reklamne kampanje u SAD i dobre početne prodaje, prvi primerci obreli su se i u Velikoj Britaniji. Postoji varijanta za IBM PC (nije za čuđenje, zar ne?) koja košta samo 775 funti. Ko voli. . . (*B.Đ.*)

Kvantni skok u nepovrat

Početkom aprila srušeno je još jedno veliko računarsko carstvo — kompanija Sinclair Research prešla je u ruke Alana Šugera, ingenioznog šefa firme Amstrad. Šuger je odmah najavio obustavu proizvodnje kontroverznog QL-a i još kontroverznijeg mikrodrajva. Vlasnici „spektruma" ne moraju da brinu: „duga" će se i dalje proizvoditi kao mašina za — igranje. Iako je kupce svojih kompjutera često umeo da zavije u crno, Klajv Sinkler je odigrao jednu od ključnih uloga u računarskoj revoluciji. On je lansirao ideju o jeftinom računaru i pripremio put svima onima koji su ga ubrzo na njemu — pretekli!

Novi Tasword

Nedavno se u prodaji pojavila nova verzija starog, dobrog tekst-procesora za „spektrum" *Tasword Two.* Nova verzija se zove, naravno *Tasword Three (Tasword 3)* i isključivo je orijentisana na rad sa mikrodrajvom.

Pored svih standardnih funkcija koje ima i Tasword Two i brojnih dodataka i poboljšanja, novi tekst-procesor je, najzad, dobio i opciju koja mu je najviše nedostajala — MERGE za poštu. Ovoj mogućnosti će se obradovati oni koji često šalju veliki broj pisama sa istom sadržinom na različite adrese. Uz pomoć novog *Tasworda,* u procesor će samo ubaciti potrebne adrese i mogu da — odu na kafu!

Novosti su još i: potpuni insert mod (dosad je, valjda, bio delimičan), štampanje do 128 znaka u redu na printeru i bafer za tastaturu, koji će omogućiti brže kucanje (ne po gumicama).

Cena — prava sitnica: samo 16,90 funti. (D.S.)

Od Amstrada do PCa

Procesorsko proširenje zasnovano na 8088 koje omogućuje da na amstradu 6128 koristi softver namenjen IBM PC-ju sasvim je lep novitet za „amstradovce" (i za siromašne Pcjevce). Proširenje košta 229 funti, a to zaista nije puno za takvu mogućnost. (*B.Đ.*)

Više za mase nego za male mase

Amstrad se veoma striktno drži svoje filozofije. Dok je jeftin dotle je i zanimljiv. Naravno, tu su i nove mašine. Jeftino, naravno. Posle PCW 8256 pojavio se i PCW 8512. PCW8256 su neki napadali, mnogi mu se čudili, a izuzetno mnogo njih ga je kupilo. Sa 8512 priča će verovatno biti nekako ista takva, osim što će oni prvi još više napadati, oni drugi mu se još više čuditi, a oni treći ga još mnogo više kupovati. Novi PCW računar namenjen obradi teksta kao i 8256 ima i printer, i monitor i disk drajv, ali ima i, uz još jedan drajv, i dvostruko više memorije — 512 kilobajta. Cena je 499 funti bez britanskog poreza na promet, koji ne plaćate ako računar iznosite iz zemlje. Pravi biser za nekog prevodioca, novinara ili saradnika „Računara".

Atarijeva PC maska

Atari je uvek bio jak u najavama. Da se ostvarilo sve što su oni najavili, ko zna gde bismo sada bili. Najnovija najava je sitna, ali značajna. Atari je najavio da će uskoro jedna firma (znači ne Atari) izbaciti na tržište hardverski emulator koji ST-a pretvara u PC. Dodatak će koštati oko 300 funti i imaće ugrađen disk drajv od 5,25 inča. Navodno, moći će da se nabavi već ovog leta. To Atariju nije bilo dosta. Za malo dalju budućnost on najavljuje sličan emulator koji bi ST pretvarao u „meka". Cena se ne pominje. Zlobnici (često ih pominjemo, zar ne) pitaju se zar Atariju ne bi bilo jednostavnije da malo jače podrži TOS i da radi na programima za sam ST. Ko zna, možda su čak i u pravu.

IMB za siromašne

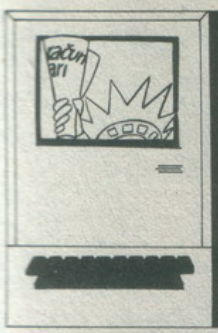
Sećate li se šuškanja o PC-ju II koja s vremena na vreme prostruje tržištem — poslednji put kod lansiranja PC JX-a u Australiji? Ponovo se intenzivno govori o novom IBM-ovom računaru koji bi odgovorio za pojačano interesovanje za sve one klonirane PC-je koji se proizvode na Dalekom istoku. Nova mašina bi trebalo da ima drajvove od tri i po inča i neverovatnu cenu od oko 400 funti. Strpimo se malo s „tajvancima". Skupljajmo pare.

Miševi u stroj!

U Velikoj Britaniji se pojavio još jedan miš za „spektrum". Ovoga puta dolazi od slavne firme Kempston, koja njime pokušava da opravda svoj stari (hardverski) renome.

Sam miš je kvalitetnije izrađen od onoga koji proizvodi firma AMX i ima dva nezavisna tastera na sebi. Interfejs je jednostavan — ne sadrži izlaz za printer, kao AMX-ov — i ne omogućuje da se programima dodaju bejzik komande za kontrolu miša. Jedino su data uputstva za očitavanje porta na koji je miš priključen.

Ali, Kempston ne bi bio ono što jeste da iznenađenje ne dolazi na kraju: uz pomenuti hardver dobijate i poznati izvanredni program za crtanje Art Studio, naravno, prilagođen za rad sa miš(onjom). Hej, MEK, kako zdravlje?! (D.S.)



Razglednica iz Njujorka

Četrdeset godina prvog elektronskog računara

"Enijak", prvi veliki digitalni računar generalne primene, sagrađen kompletno od elektronskih komponenti, sastavljen je u Murovoj školi Inženjerstva pri Univerzitetu Pensilvanija u Filadelfiji 1946. godine. Tzv. numerički integrator i računar su ispunjavali laboratoriju dimezija 10×20 metara. Težina čitavog hardvera je bila preko 30 tona, pa nije ni čudo da je upravo engleska reč hardware (gvožđurija) bila izabrana kao opisno ime za računarski sistem bez programa. "Enijak" je imao 70.000 otpornika, 10.000 kondenzatora i 18.000 elektrnskih cevi kao prekidačkih elemenata. Pri izgradnji računara bilo je potrebno zalemiti preko 500.000 spojeva.

Projekat ,,Enijak'' je započeo u tajnosti ratne 1943. godine sa namenom da se omogući automatsko izračunavanje tabela artiljeriskih parametara za precizno gađanje. Na žalost, ili na sreću, "enijak" nije kompletiran na vreme da bi poslužio u prvobitne svrhe. Da ironija bude veća, prvi problem rešavan na "enijaku" je bio sistem diferencijalnih jednačina kojima je bila modelirana hidrogenska bomba. Za nepuna dva sata "enijak" je bio u stanju da kompletira računarski posao ekvivalentan radu deset matematičara sa punim radnim vremenom u trajanju od deset godina.

Delovi praoca svih današnjih računara se danas nalaze izloženi u Smitsonovom institutu u Vašingtonu, Vojnoj akademiji Vest Point i na samom Univerzitetu Pensilavanija.

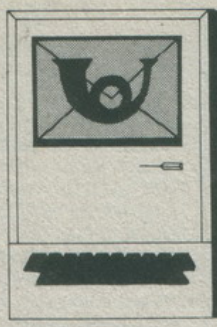
Ove godine Univerzitet Pensilvanija slavi 40-godišnjicu nastanka računara "enijak", 40-godišnjicu prvog predmeta iz oblasti računarskih nauka koji je ikada predavan na jednom univerzitetu, kao i 40-godišnjicu formiranja prve firme za proizvodnju računara. Prvu računarsku kompaniju su osnovali nosioci projekte "enijak" — Džona Mokli (John Mauchly) i Presper Ekert (Presper J. Eckert), koji su zajedno radili na "enijaku" sa Džonom Brajnardom (John Brainerd).

Murova škola Inženjerstva je održala komemorativnu svečanost u čast ,,enijaka'' u februaru ove godine,

a za mesec oktobar priprema se gala večera u čast trostruke 40-godišnjice. Očekuje se da će 22 studenta iz prvog razreda koji je ikada slušao jedan računarski predmet (toliko ih je još u životu) biti prisutno na večeri. Mnogi od studenata iz te generacije su danas poznati eksperti, političari i menadžeri. Dekan Murove škole, dr Džozef Bordonja (Jospeh Bordogna), kaže da se ta generacija od rastanka 1946. godine pa do ove godine nije sastajala.

Interesantna je činjenica da današnji mikroprocesori dimanzija santimetar kvadratni sa cenom od desetak dolara imaju performanse koje prevazilaze miliondolarski ,,enijak'' iz četrdeset šeste. U jubilarno vreme kao što je ova godina dok poredimo sadašnju situaciju u računarskoj industriji sa njenim počecima, pitamo se: ,,kakvim ćemo računarskim kapacitetima raspolagati kroz nekoliko godina?''

R.A. Mihajlović



Load dragi računari''

Svi su hakeri braća

Hej, Redakcijo! Jeste li našli maloj Andrei ,,dvicu''? Niste?! Ja ih slučajno imam dvije! Pod parolom ,,Imaš puno ,,Računara'': — daj malo'' javljam se na njeno toplo pismo i nudim joj svoju pomoć.

Doduše, nijedna ,,dvica'' nije kao iz štamparije, ali što se tu može? Naime, ćoškovi su malo ,,isfucani'' (zar da se osećam krivim što svaki broj okrenem deset puta u toku dana), ali su svi listovi na broju i neoštećeni.

Andrea, da li te interesuje ovaj predlog?

Tomislave, teši se da si sada prvi na listi čekanja. (Žao mi je što nemam ,,dvicu'' i za tebe).

Zrnić Željko
Fruškogorska 19
78000 Banjaluka

P.S. Andrea, hoćeš li i meni poslati pusu?
P.P.S. Predlažem i drugim ljubiteljima ,,Računara'' koji ima više retkih brojeva da pomognu i drugim Andreama i Tomislavima. Jer svi su hakeri braća! Hmm, da nije ovo uzviknuo Aristotel?

Malokalibarski računar za dame

Neću da trošim puno prostora pokazujući kako čitam ,,računare'' od prvog broja i da ih puno volim. Preći ću odmah na naslovne strane. Meni se puno sviđaju vaše naslovne strane — da ne pominjem ženske koje su na njima. To je jedna činjenica. Druga činjenica je to da te ženske polako istiskuju kompjutere sa naslovnih strana. U poslednjem broju, ,,Računarima 14'', imate jednu jako zanimljivu devojku na naslovnoj strani, ali je zato računar gotovo nestao. Ostalo je nešto pod imenom Casio.

Nemojte sad da se predomislite i da smanjite uticaj ženskog sveta na naslovne strane ,,Računara''. Samo nemojte ni da računari potpuno nestanu sa vaših naslovnih strana.

Goran Aćimović
Beograd

Ne boj se, računari neće nestati! Mi imamo dužnost da brinemo za sve vrste računara, pa je tako, ovoga puta, na naslovnoj strani, pored zanimljive devojke koju pominješ, prikazan i specijalan model damskog malokalibarskog računara. Sve na svome mestu.

Hamlet igra Elitu

Staus konzervativne avangarde dokazali ste poslednjom naslovnom stranom, zakopčanom do levog uha (košulja) mada dolazim do zaključka da vam ekipa u štampariji previše čita Bukovskog i suradnike. Onako nemoralan prikaz ženske podlaktice i to sa ruskim satom i paralelom u obliku značke komesara Police Departmenta ostavilo me sa trgovinskim deficitom od 300 dinara.

Kad biste malčice skratili one sažetke igara, a ne da meni ispali 6 sijalica od 150 kW dok dođem do pola . . . Dobro ima tu umjetničkih elemenata, ali batali. Nije važno.

Divno je kad naša hakerska opozicija pokaže humanu solidarnost sa Francuskom i opredeli se za ,,orica'' ili, u žargonu, ,,o-bika''. Jugo bikovi, uspaljeni erotičnošću crvenih tipki, nisu odoleli, pa smo sada, nakon Mongolije, i mi postali kompatibilni sa čitavom Gvinejom Bisao. Dobro sada, i nije tako lošo . . . očajan je.

Onaj ekranski editor vam je bomba. Ode mi dioptrija sa −1 na −9 i sad više ni kontaktne ne mogu nosit. Još i konjunktivitis! Sad se, vjerojatno, pitate gde se gasim. Odoh ja, društvo, ali vi ste još uvijek the best.

Goran Štimac
32. divizije 19
41000 Zagreb

Bili smo iskreno uvereni da će kad-tad početi da nam stižu i pesnička pisma. I, gle čuda, to se i dogodilo. Ovo je jedno od takvih poetskih pisama koja ćemo kroz koju godinu objaviti u zbirci pod naslovom ,,Umetnost resetovanja''. Reset.

Antibiotik za zvrkove

Javljam se povodom pisma Tomislava Vazdara. Ja sam također kupio ,,zvrkove''. Srećom, posjedovao sam Large Cargo Bay. Ubrzo sam primjetio što se događa i napustio sam spremište do kraja, čvrsto riješen da se zaletim u neku planetu i završim igru. Odlučio sam da kamikaze akciju sprovedem na maloj planeti koju do tada nisam bio vidjeo(?!). Krenuo sam hiperbrzinom i zaustavio se vrlo blizu. Od muke sam pogledao da li su zvrkovi već izašli. Nemalo sam se iznenadio kad sam vidjeo da su nestali. Prostor je bio slobodan za robu.

Radoslav Dejanović
V. Kovačića 5
Zagreb

Čovek se uči dok je živ. Ili bar dok je komander.

Randevu sa Jelenom

Čitam vaš i naš list od prvog broja do poslednjeg, koji je, trenutno, ispred mene, na stolu. Veoma mi se sviđaju rubrike ,,Load . .'', Peek&poke show'' i svi tekstovi MNOGOPOŠTOVANE Jelene Rupnik i malo manje poštovanih ostalih

članova redakcije (bez uvrede, naravno). Ne bih da šaljem Jeleni nikakve poljupce preko redakcije, već bih samo molio redakciju za informaciju o krugovima u kojima se J. R. kreće, kako bih, eventualno, mogao sa njom da razmenim mišljenje.

Voleo bih da saznam na koji način se dolazi do potrebne valute za nabavljanje naših mezimaca, jer, koliko je meni poznato, devizno tržište je otvoreno samo za radne organizacije. Istina, čuo sam za neke laserske štampače koji vrlo verno preslikavaju izvesno zeleno naslikane i ostale papirčiće (Nagradno pitanje: Postoji li interfejs pomoću kojeg pomenuti štampač mogu povezati sa „spektrumom"?).

Hteo bih nešto da poručim drugu Suadu B. koji tako slobodno izjavljuje da mrzi „spektrum" i njegove vlasnike. Hteo bih da konstatujem žalosnu činjenicu da poznajem par novopečenih „komodorista" koji su se do juče ponosili svojim „dugama", a sada ih sramežljivo teraju u zapećak, uz prezrivu tvrdnju da se gumice koriste samo za tegle sa slatkim . . . E, pa ja svim tim zlobnicima poručujem da se gumice mogu upotrebiti i u preventivne svrhe od prevelike navale raspusnih „komodorista". . .

Za kraj bih pozdravio sve članove uredništva i redakcije, uz molbu da u nekom broju objave onaj (barem meni) zagonetni način na koji se mogu u program ubaciti pokovi, šifre i ostala preko potrebna pomagala za svršetak igara. Jedino bih još mogao da zamolim za preporuku za brojeve sledećeg kola Lotoa — možda bih tako mogao da poboljšam svoje finansijske i kompjuterske pozicije.

Vanja Đurđević
Proleterskih brigada 6/9
11001 Beograd

P.S. Malo je trebalo da zaboravim glavno pitanje: Kako se postaje saradnik „Računara"? Naravno, honorarni. Šta ćete, svi smo mi materijalisti — velika je inflacija!

Da nismo, pre jedno godinu dana, nabavili jedan od tih laserskih štampača koji tako dobro preslikavaju zelene novčanice, teško da bismo mogli da nabavljamo sve one lepe mašine koje prikazujemo. Redakcija „Peek & poke showa" razmišlja o raspisivanju nagradnog konkursa na temu „Za šta sve mogu da služe gumice?" Da li bi bilo odziva? Saradnik „Računara" se postaje veoma lako. Prvo nam pošalješ nešto što si ti napisao ili isprogramirao. Onda mi pogledamo da li nam to odgovara. Ako nam odgovara, mi to najčešće objavimo. Ako objavimo, onda i platimo. I tako dalje.

Manje-više oštra kritika

Pored pohvala, vreme je da, najzad, dobijete i poneku manje-više oštru kritiku. Ne bojte se, nije toliko strašno! Drug Željko iz Banjaluke je sasvim u pravu! Uradite nešto sa tim ivicama! Devedeset i pet odsto računardžija najviše voli da se igra sa svojim mikro-ljubimcima. Zato ne dajte da izađe broj bez bar 5—6 podugačkih opisa najnovijih igara. Izaberite ponovo 50 najboljih za „spektruma" i „komodora", dodajte još par onako kvalitetnih strana sa lepim sličicama, kao u poslednjem broju. Taman me za nešto zainteresujete nekim kratkim opisom malog, novog hard-dodatka, najavite opširniji prikaz u sledećem broju i — ništa! Ja čekam!

Aleksandar Pantić
Dušana Popovića 3
11050 Beograd

Drago nam je kada nas kritikuju ljudi iz čijih pisama se vidi da nas i vole. Nije bilo strašno. Naprotiv. Neki delovi „Računara" će da se razvijaju upravo onako kako bi se tebi svideli. Naročito kad je reč o ivicama. Pričekaj.

Humanizam i renesansa

Samo da znate da me je vaša „trinaestica" (i naša kad rikneš tri konja) skoro stajala glave. Naime, umalo sam skočio kroz prozor od radosti kada sam pročitao da ćete objavljivati listinge ekranskog editora Vlade Kostića.

Šlogirao sam se kada sam pročitao odgovor pismu Suada Bejtovića. Kako možete da hvalite to? Ne shvatam kako se može mrzeti jedan mikroprocesor? Svaki procesor, a samim tim i računar, vredi onoliko koliko vredi njegov programer ili korisnik. Prema tome, 6502 druga Bejtovića je bezvredan koliko i njegov „komodor". Slažem se sa tim da fali Biblioteka u dvanaestici, samo mi nije jasno zašto se on uzrujava zbog toga (tamo neće naći mapu ni jedne igre). Zato predlažem drugu Bejtoviću da nastavi uništavati invejdere i da se ne meša u stvari za koje nije dorastao. Nadam se da će ljudi, napokon, shvatiti da računari (ne mislim na list) ne predstavljaju sredstvo za iskazivanje neprijateljstva prema okolini.

Zoltan Pastor
Laze Mančića 1/B
Novi Sad

Jako nam se svideo humanistički završetak tvog pisma, ali da li je prethodni deo pisma u skladu sa njim. Hajde, priznaj, ipak nisi tako mislio. „Računari" uče da kad ti neko podmetne bajt, ti treba da mu pružiš još jedan registar.

Suad i Suada

Opet onaj što ga Kelly pozdravlja. Tako je: Suad Bejtović iz Olimpijskog grada (ko je pomenuo Beograd?).

Momci iz Yugoslav Cracking Servicea su mi poslali pismo u kojem piše „128 is the best!!!" (u prevodu — 128 je najbolji!). To pismo ću da pokažem svima u RAK-u. AH, da — RAK. To je jedini Računarski amaterski klub u BiH Imamo QL, četiri „spektruma" i jednog C-64 sa diskom. Osim toga, imamo oko 600 članova. Od njih je 589 muških. I onda neki tamo kažu da je zabluda da kompjuteri nisu za žene. Spektrumovci iz RAK-a su hteli na krst da me prikucaju zbog nekih tamo izjava u kojima ih napadam. Tužna ironija je to da je 99% pametnih ljudi u klubu opredeljeno za „spektrum". Ja, iapk, volim pametne ljude i ispade da ne mrzim sve spektrumovce.

Neki kažu da je 13 nesretan broj, ali je vaš trinaesti broj ispunio sva očekivanja. Ja stvarno nemam ideju zašto se kupuju ostali računarski časopisi pored vas živih. „Razbarušeni sprajtovi" su super („spektrum" nema pokove). Ponavljam, vi ste najbolji računarski časopis u Kumovoj Slami — posle PCW-a.

Još nešto, gde nestade Jelena Rupnik u 13. broju? Na nju liči ono „Računari i žene", ali to nije potpisano. Osim toga, to piše udata žena, a sumnjam da je Jelena toliko nefer da se uda pored tolikih živih čitalaca „Računara". Uz put, nemojte da objavite njenu sliku. Neka svaki ubogi pirat i haker (spadam u obje kategorije) zamišlja svoju verziju Jelene Rupnik! Ne može ona da bude po volji **svim** hakerima i piratima i šta ste onda postigli (jesam dosadan). Živela Jelena Rupnik, boginja zaštitnica svih hakera i pirata.

Smatram da ovo pismo nije tako dobro kao prošlo, ali sjetite se onog oprobanog TV pravila: u pilot emisiji staviti sve najbolje da emisija krene, a onda kako bude.

Suad Bejtović
Palmira Toljatija 136/V
71000 Sarajevo

P.S. Imam dva iz rukopisa. A uopšte se ne primjeti. Ne vole me nastavnici.

Vlada kao turista

Primetio sam da je „Svet kompjutera nedavno objavio jedan kraći tekst, vašeg pera Vlade Kostića poznatijeg kao THE Haker. Pitam se da li to znači da je Vlada promenio boje kluba i počeo da piše i za druge ili je to, možda, neki drugi Vlada Kostić, ili ja nisam dobro video potpis. Vi, verovatno, znate šta je od toga istina, pa vas molim da mi objasnite. To nije neko značajno pitanje, ali me puno interesuje.

Saša Žeković
Zagreb

Stvar je dosta jednostavna. Vlada je naš eksluzivac i nije promenio boje. Tekst koji ste videli objavljen je s našim dopuštanjem.

Sedam gnevnih čitalaca

Upravo sam završio sa prvim pregledanjem „četrnaestice" i sa žaljenjem sam ustanovio da u njoj nema mog pisma. To me, ipak, nije pokolebalo i pišem ponovo. Možda će moje pismo delovati isuviše kritizerski, ali to je zbog toga jer sam primetio da se mnogo više osvrćete na kritikovanja nego na molbe.

— Kao prvu primedbu naveo bih moje mišljenje u vezi sa poskupljenjem lista. „Računari" sada koštaju lepih okruglih tri stotine dinara. Imaju raznovrsne teme koje su manje i više zanimljive (svaka vam čast na izboru) i sve to je smešteno na šezdeset i osam strana (sa koricama). „Galaksija" ništa manje dobar list sa obiljem zanimljivih tema — nema nezanimljivih — na istom obimu stranica i sa mnogo ozbiljnijim štampanjem, posebno fotografija, košta samo sto pedeset novih dinara — znači upola jeftinije.

Sve zaključke prepuštam čitalačkom auditorijumu, pošto vi obično nemate komentara za ovakve primedbe. Napominjem još jednom da „Računare" čitam od prvog broja, bez obzira na poskupljenja kojima smo sa vaše strane bili izlagani.

— Kao drugu primedbu bih izneo vaše lično razmetanje u nekim prilikama. Ja lično nemam ništa protiv bilo kog računara (osim par izuzetaka), ali mi jako smetaju omalovažavanja i hvaljenja od strane „komodorista" na račun nas „SPEKTRUMAŠA", da ostale i ne spominjem. Što se vas lično tiče rekao bih samo toliko da mi nije jasno kako ste mogli da se hvalite da cela redakcija poseduje BBC računare (kolike li su vam samo plate!!!). Verovatno svaki član redakcije ima par po dve tri tetke i pet šest ujaka na privremenom radu po Engleskoj i Nemačkoj, zar ne!!!

Treća stvar bi mogla da bude u vezi s načinom nabavke opreme. Na čast vam služi članak u najnovijem broju! Uostalom, kada već toliko piratujemo programe, što ne bismo švercali i računare (makar i u delovima). Sreća vaša a i naša što carinici čitaju samo inostrane cenovnike. Baš razmišljam kako sad da uvezem jedno deset pomenutih IBM klonova.

— Mislim da bi bilo vrlo zgodno da u nekom od brojeva pomenete kako se na neki instant način može doći do para za svu divotu iz rubrike „Šta ima novo". — Siguran sam da bi tu jako pomogla iskustva članova redakcije i njenih saradnika, posebno po pitanju deviza, jer ja nemam nikoga u inostranstvu (tetke, babe, ujne, stričeve . . .)

Čini mi se da SIV i narodna banka još ne vrše prodaju deviza na malo (kad počnu, obavestite me).

— Ima još par nepoznatih za koje smatram da mogu biti interesantne za sve nas, pa čak i veoma korisne za same „Računare".

Predlažem vam da osnujete tečaj za vaše ljude iz štamparske službe gde bi oni išli na praksu u štampariju koja štampa jedan ugledni i dobro poznati list za računare u bratskoj nam republici Sloveniji. Da bi odziv vaših radnika bio veći, može se organizovati izlet do dragog nam Trsta, besplatno naravno (sve u duhu stabilizacije). Pitam se, ne vas, samog sebe, kako ti Slovenci mogu samo da čine sve takve stvari, a vi ne možete, ccc!!!

Na kraju bih vas samo zamolio da mi oprostite na neuredno otkucanom tekstu, jer nisam bio u mogućnosti da ovaj tekst obradim preko „WORDSTARA" kako to radi drug Srđan Kosovac (neka bude IBM), čak je i ova mašina pozajmljena. Šta mi je?

Drugarski vas pozdravljm, s nadom da ćete kvalitet štampe poboljšati i pre 2000.-te godine.

Vanja Đurđević

Čitaoci „Galaksije" dobijaju svoj list na kioscima jeftinije nego što on košta u proizvodnji. To je rezultat svesnog ulaganja OOUR „Duga" u popularizaciju nauke. „Računari" izlaze na komercijalnoj osnovi — oni moraju da žive ili od onih zbog kojih i postoje, dakle od svojih čitalaca, ili od — oglasa. Da li bi bio srećniji da je vreme stalo sa računarima ZX 81 i „galaksijom"? Grafičari BIGZ-a cene umešnost svojih slovenačkih kolega. Donesi nam, ipak, koji tabak onog finog papira na kome rade Slovenci, pa ćeš videti da ni mi nismo ništa lošiji.

Periferijska oprema

štampači

Ovaj prikaz je, uglavnom, usmeren na štampače koji su danas interesantni kako za programere i hakere, tako i za one koji planiraju poslovnu primenu kompjutera: matrični štampači koji ispisuju 80 normalnih znakova u redu, poseduju NLQ opciju, koštaju 200—500 funti i nude danas praktično neophodnu „epson" kompatibilnost. Ukoliko za to bude interesovanja i mogućnosti, pripremićemo i prikaze ostalih tipova printera (štampači sa lepezom, električne pisaće mašine sa interfejsima, ink-jet printeri itd) za neki od sledećih brojeva našeg časopisa.

Pre nego što pogledate našu veliku tabelu, napomene o pojedinim modelima i njihove otiske, posvetićemo određenu pažnju karakteristikama štampača i pokušati da objasnimo termine koji se često sreću a retko razumeju. Svi štampači koje prikazujemo su **matrični,** što znači da se otisak na papiru formira udaranjem iglica u specijalnu traku. Obzirom da se znaci sastoje od tačkica, njihova čitljivost uvek teži da bude neprijatno slaba, pogotovu za ljude koji nisu navikli na ovakav način pisanja. Struktura će se, jasno, manje primećivati ako su tačke gušće raspoređene, što znači da štampači koji imaju više iglica nude kvalitetniji tekst. U našoj tabeli smo, međutim, odustali od navođenja veličine takozvane „matrice karaktera" jer je ona jednaka za sve modele koje pominjemo: 11*9. Razloge za ovakvu unificiranost nije preteško pogoditi: svi su modeli „epson" kompatibilni što znači da programi koji definišu karaktere na „epsonu" moraju da rade i na njima. Davno je, uz to, procenjeno, da je matrica 11*9 dovoljna za „normalne" primene; za specijalne primene su rezervisane i specijalne opcije.

Strogo kontrolisani štampači

Početni redovi tabele, osim imena štampača (1), navode i njegovu englesku cenu (2) koja uključuje i VAT od 15%. Zatim prelazimo na tehničke podatke: vrsta 3 nabraja brzinu štampača koju proizvođač navodi u specifikacijama. Ovaj je podatak potpuno beskoristan i verovatno nikome na svetu nije jasno kako se do njega dolazi: realne brzine su daleko manje. Zato smo usvojili nekoliko dodatnih testova. **Svi** štampači o kojima govorimo omogućavaju automatsko testiranje: dovoljno je da ubacite papir i traku i, uključujući štampač, pritisnete neki specijalni taster. Printer će tada, sve dok ga ne isključite, u sukcesivnim redovima ispisivati svoj set karaktera i tako vam omogućiti da kontrolišete ispravnost uređaja. Obzirom da je ovo štampanje izuzetno brzo, posvetili smo mu vrstu 4; primetićete da je čak i takva brzina daleko manja od one koju navodi proizvođač. Rezultati se, dalje, smanjuju kada dovedemo štampač u realne uslove: vrsta 5 daje brzinu rada na specijalno pripremljenom sintetičkom tekstu koji je dug 5000 znakova (oko 2,5 šlajfni). Tekst je *sintetički* utoliko što sadrži minimalan broj blanko simbola, što su redovi podjednakih dužina i što sadrži minimalan broj blanko simbola, što su redovi podjednakih dužina i što se piše samo jednom vrstom slova. Vrsta 6 daje rezultat koji će vam u praksi najviše značiti: brzina štampanja standardnog teksta sa novim redovima, podvlačenjima, prelascima na kurzivna *(italic)* slova, korišćenje istaknutih naslova i podnaslova. Tačnost merenja je diktirana tačnošću štoperice, pošto ne bi bilo fer koristiti časovnik ugrađen u računar: svaki štampač ima bafer određene veličine (21), tako da je računar slobodan mnogo pre nego što štampač završi posao.

Iako se još nismo detaljno bavili NLQ modom, verovatno znate da moderni matrični štampači nude daleko superiorniji otisak dobijen na račun brzine rada. Gubici u brzini rada su, kao što se vidi iz vrste 7, ogromni: standardan tekst od 10000 znakova se u standardnom *(draft)* modu ispisuje brzinom od stotinak karaktera u sekundu, dok isti tekst u NLQ modu zahteva sekundu za svega petnaestak znakova!

Svaki se štampač, jasno, nalazi pod kontrolom računara: umesto slova, kompjuter može da pošalje propisanu sekvencu kontrolnih kodova i tako izazove promenu tipova slova, prelazak na novi list i slične stvari. Kontrolisati štampač posredstvom računara nije, međutim, baš uvek komforno: često ćemo poželeti da preskočimo nekoliko redova ili pređemo na sledeći list, a da pri tom ne pamtimo komplikovane 'Escape' sekvenca i ne kucamo duge komandne linije. Zbog toga je na kutije svih modernih štampača ugrađeno po nekoliko tastera. Uz neizbežni *On-Off*, štampači koji slede Epsonove standarde imaju dirku *On Line,* pomoću koje se štampaču zabranjuje da prima dalje znakove od računara (zbog čega je to korisno? Ponekad će vam zatrebati da promenite list papira ili obavite telefonski razgovor u toku koga vas buka neće ometati), taster *Line Feed* pomoću koga štampač preskače jednu liniju i, najzad, taster *Form Feed* koji izaziva prelazak na prvi red sledeće stranice. Ni jedan od štampača koji pominjemo nema mogućnost da detektuje perforaciju između stranica, što je odlika nekih skupih profesionalnih modela: kraj strane se prepoznaje tako što je štampaču poznata njena dužina i time što pretpostavlja da se po svakom uključivanju nalazi na početku novog tabaka. Ukoliko vam se, dakle, dogodi da isključite štampač koji nije završio sa ispisivanjem stranice, moraćete da okrećete papir pomoću uvek prisutne ručice i da uključite printer tek kada perforacija prođe ispod glave.

Osim standardnih kontrola koje se podrazumevaju pa ih nije imalo smisla nabrajati u tabeli, neki štampači omogućavaju da sa kontrolnog panela biraju NLQ slova (8), pa čak i da se izabere mod i tip slova (9).

Modovi, tipovi i širine

Domaći ljubitelji računara (pa čak i kompjuterski profesionalci) često mešaju termine „mod" i „tip slova". *Tip* (ili, u stranoj literaturi, *font)* označava oblik slova, na primer uspravna *(plain),* kurzivna *(italic* — nikako nismo uspeli da saznamo da li su termini *kurziv* i *italic* sinonimi, ali u svakom slučaju jako liče) ili gotica. Reč *mod* označava način ispisivanja slova i daje predstavu o kvalitetu rezultujućeg teksta; uobičajeni su termini *normalni (draft), NLQ (Near Letter Quality)* i *LQ (Letter Quality) mod.* Tako možemo imati uspravna slova pisana u draft modu ili kurzivna slova u NLQ modu.

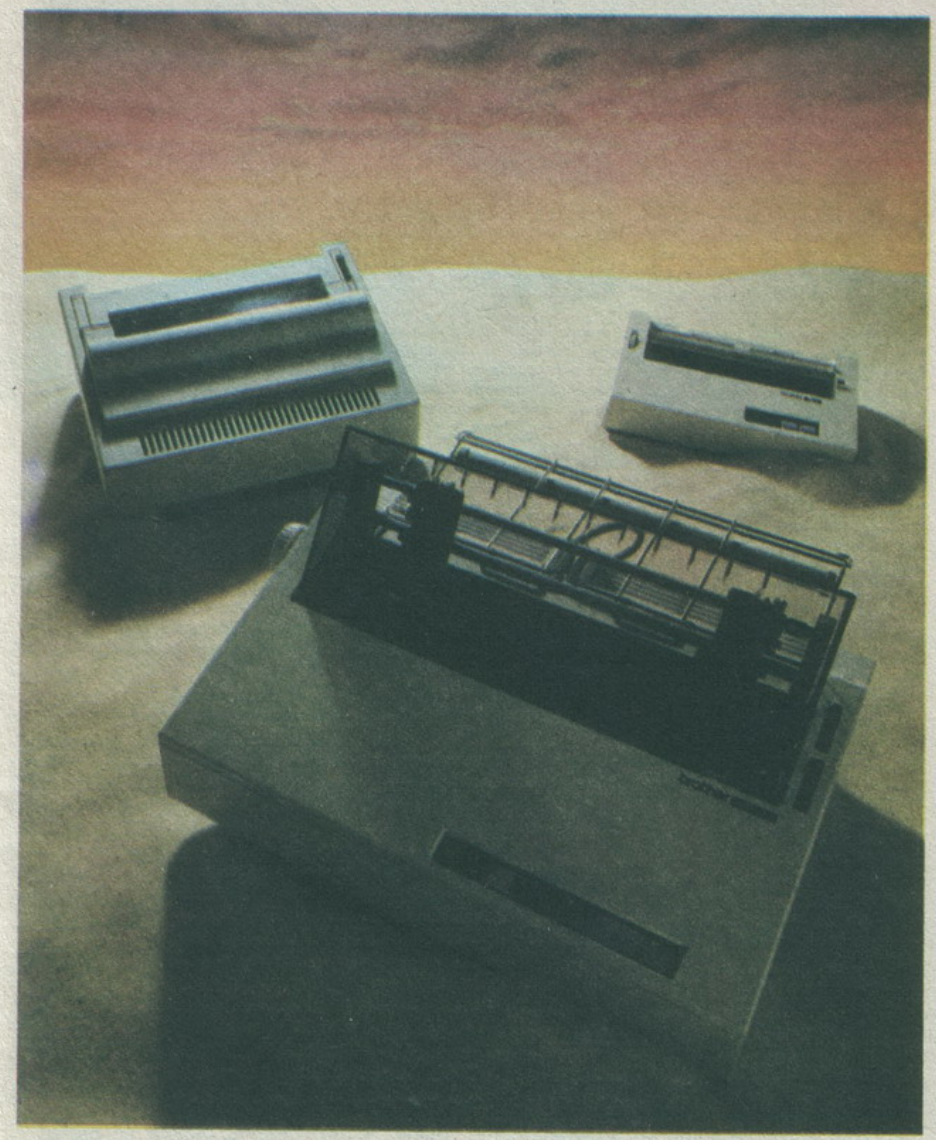
Da bi stvar bila još komplikovanija, uvodimo i termin *širina slova (pitch).* Standardne širine slova su *Pica* (10 znakova na inč širine papira), *Elite* (12 znakova po inču pri čemu, kod nekih štampača *Elite* slova imaju i drugačije oblike, što bi značilo da se radi o novom tipu), *Condensed* (17 karaktera po inču), *Enlarged* (5 znakova po inču) i *Condensed Enlarged* (8,5 znakova po inču). Neki štampači, uz standardni indikator koji pokazuje da li su „On Line", imaju i posebne LED diode koje prikazuju da li im je nestalo papira (10), u kom se modu nalaze (11) pa čak i zvučni alarm (12) koji se oglašava kada nešto nije u redu, na primer kada se papir umrsio.

Uz kontrole na kućištu, svi štampači poseduju i grupe mikroprekidača kojima možete da izaberete njihovo početno stanje. Ukoliko, na primer, uglavnom radite sa kondenzovanim slovima kako biste u svaki red mogli da smestite 132 znaka, jednom ćete odvrnuti nekoliko zavrtanja, skinuti kućište štampača i pomeriti mikroprekidač koji se odnosi na kondenzovani mod. Tako ćete izbeći potrebu da po svakom uključivanju štampača šaljete sekvencu kontrolnih kodova koji ga prebacuju na kondezovana slova, ali ćete, ako vam jednom zatrebaju normalna slova, morati da šaljete drugu sličnu sekvencu. Opšta karakteristika mikroprekidača je, dakle, da se stanje koje je njima postavljeno **može** poništiti ili promeniti softverski. Za jugoslovenske potencijalne vlasnike štampača su posebno važne kolone 13 i 14 naše tabele: kod većine printera se mikroprekidačima može izabrati dužina stranice (naš list ima 72 reda, a engleski i američki 66) i rastojanje između linija. **Svi** pobrojani štampači imaju mikroprekidač kojim se može zahtevati preskakanje perforacije na kraju svake stranice.

Vrste papira

Laicima je najnormalnije da štampači koriste običan papir A4 formata. Ovaj se papir, kao i na pisaćoj mašini, kreće oko

Ako se izuzmu prikazi ponekog popularnog modela, „Računari" se još od prvog broja nisu ozbiljnije bavili štampačima. Vremena kada su nastajali „Računari u vašoj kući" su sada daleko iza nas, pa nam čitanje tadašnjeg napisa izmamljuje podosta osmeha: o printeru se više ne sme govoriti kao o nedostižnoj spravi potrebnoj jedino onima koji primenjuju kompjutere u radnjama male privrede — oni danas predstavljaju nezaobilazan inventar svakog kućnog računskog sistema. Štampači izvanrednih karakteristika se na stranom tržištu nude pod izuzetno povoljnim uslovima. U ovim i sledećim „Računarima" dajemo detaljan prikaz najkvalitetnijih i najprivlačnijih modela, a pripremamo i nekoliko ekskluzivnih priloga za one koji ih već imaju — detaljno uputstvo za upotrebu štampača iz „epson" standarda i uputstvo za ubacivanje YU slova u njihove ROM-ove.



pokretnog valjka koji je ugrađen u standardnu opremu većine štampača iz srednje klase (16). Nije, međutim, baš prijatno koristiti A4 listove: štampaču je, da bi ispisao jedan takav list, potreban otprilike jedan minut, što znači da ćete svakoga minuta morati da ubacujete novi list! Posle izvesnog vremena ćete utvrditi da ubacivanje listova odnosi više vremena nego samo štampanje, pa ćete potražiti racionalnije rešenje. Jugoslovenski vlasnici računara obično koriste perforirani kompjuterski papir (u stranoj literaturi *fon-fold paper)* koji se obično isporučuje u sanducima od kojih svaki sadrži po 2000 međusobno povezanih listova papira proširenih perforacijom sa svake strane (sanduk od 2000 listova košta oko 5000 dinara i traje prilično dugo; autor ovoga teksta nikada nije naročito štedeo papir, a ipak mu svaki sanduk potraje po desetak meseci). Štampač koji treba da prima ovakav papir mora da bude opremljen takozvanim *traktorom* koji je obično (15) uključen u standardnu opremu. Širina listova koje prima traktor se može menjati pomeranjem vođica za papir, ali je ovo pomeranje kod nekih starijih modela prilično loše rešeno tako da štampač može prihvatati samo papir u uskom rasponu širina. Moderniji printeri (19) mogu da primaju papir svih širina do maksimalne (18), tako da se mogu ispisivati čak i uske nalepnice za koverte, što je od značaja za određene poslovne primene.

Najjeftinije rešenje je korišćenje rolne papira, za šta je, kod nekih printera, potrebno dokuptii poneki dodatak (17). Ovaj je papir prilično nekvalitetan i ima ga smisla koristiti samo ako se rezultati koriste za lične potrebe vlasnika računara. Jasno je da pre početka rada sa rolnama treba onemogućiti preskakanje perforacije, jer perforacija i ne postoji!

Cepanje papira je veoma bolna tačka za većinu štampača: u normalnom radu često je potrebno odcepiti upravo ispisani list papira bez potrebe da se izbacuje još jedan prazan i, po mogućnosti, bez potrebe da korisnik ustaje sa stolice. Pitanje cepanja papira je razrešeno na najrazličitije načine (20) i, ma kako banalno izgledalo, predstavlja jedan od veoma ozbiljnih parametara koji treba uzeti u obzir pri izboru štampača.

Tehnologija štampanja

Posvetimo malo pažnje stvarima koje se događaju od momenta kada računar pošalje jedan znak štampaču od momenta kada se taj znak pojavi na papiru. Veze računara i štampača se obično zasnivaju na principu takozvanog „hand shaking-a": osim linija za prenos podataka, računar i printer su povezani linijom koju zovemo *ready-busy.* Preko ove linije štampač signalizira računaru da je spreman da primi sledeći znak koji mu zatim biva i poslat. Taj znak putuje pravo u takozvani *bafer,* specijalni RAM koji je ugrađen u štampač. Kada se bafer popuni (njegova veličina (21) varira od modela do modela), računar mora da čeka da se neki znak odštampa da bi u baferu bilo mesta za sledeći. Veliki bafer omogućuje da se računar „oslobodi" za druge poslove mnogo pre nego što se štampanje završi; ukoliko uglavnom pišete kraća pisma i izveštaje, isplatiće vam se da investirate u veći bafer, jer će tako štampač moći da radi dok vi (ako vam ne smeta buka) pišete sledeće pismo. Na baferu se, sa druge strane, može dosta i uštedeti: računari sa pristojnim RAM-om koji su opremljeni fleksibilnim operativnim sistemom mogu da organizuju komunikaciju sa štampačem preko prekida (interapta) i tako paralelno obavljaju dva posla koristeći višak sopstvenog RAM-a kao bafer za štampač. Ovakva metodologija je kod Acornovih računara koji su dopunjeni „bočnim" RAM-om dovedena do savršenstva.

Pre nego što se oduševite veličinom bafera nekog od modela koji opisujemo, razmislite o jednom ograničenju: uobičajeno je da se bafer za štampač koristi i za definicije karaktera, tako da se pomeranjem specijalnog mikroprekidača bira jedna od ove dve namene. Kako će vam s vremena na vreme zatrebati naša latinična slova i kako će vas mrzeti da par puta nedeljno

UPOREDNE KARAKTERISTIKE ŠTAMPAČA

	Model	Epson LX80	Epson FX85	Mannesmann Tally 85	Kaga Taxan KP 810	Canon PW 1080A
2.	Orijentaciona cena (funti)	350	526	452	300	330
	BRZINA					
3.	Prema specifikaciji (CPS)	100	160	180	140	160
4.	Brzina self testa	77	112	121	96	112
5.	Sintetički tekst	81	111	131	93	102
6.	Realni tekst	57	79	117	71	98
7.	NLQ realni tekst	16	21	26	21	25
	KONTROLNI PANEL					
8.	Izbor NLQ moda	da	da	da	da	da
9.	Izbor tipa slova	ne	ne	ne	ne	ne
	INDIKATORI					
10.	Nestanak papira	da	da	da	da	da
11.	Izabran NLQ mod	ne	ne	da	ne	ne
12.	Zvučni alarm	da	da	da	da	da
	MIKROPREKIDAČI					
13.	Izbor dužine strane	da	ne	da	da	da
14.	Izbor rastojanja izmedju redova	da	ne	da	da	da
	PAPIR					
15.	Perforirani papir	opcija	da	da	da	da
16.	A4 listovi	da	da	da	da	da
17.	Rolna	da	da	opcija	da	da
18.	Max. širina (mm)	264	280	254	254	254
19.	Širina traktora se podešava?	da	ne	da	da	da
20.	Cepanje listova	slabo	dobro	slabo	lako	izvanredno
	KVALITET TEKSTA I KOMFOR					
21.	Bafer (KB)	1	8	3	3.5	3.5
22.	Broj specijalnih azbuka (stranih)	11	10	9	9	9
23.	Oblici slova	super	super	super	dobri	dobri
24.	Descenders	dobri	dobri	super	dobri	dobri
25.	Podvlačenje	OK	OK	dobro	dobro	super
26.	Tačkasta struktura (draft)	OK	OK	dobra	OK	dobra
27.	Tačkasta struktura (NLQ)	dobra	OK	super	dobra	odlična
28.	Broj karbon kopija	1	1-2	1	1	1
	ODRŽAVANJE					
29.	Umetanje listova papira	lako	OK	lako	lako	vrlo lako
30.	Umetanje perf. papira	-	teško	lako	uglavnom lako	lako
31.	Promena trake	vrlo laka	teška	vrlo laka	laka	laka
32.	Da li su ruke prljave?	ne	veoma	ne	ne	ni malo
33.	Uklanjanje zgužvanog lista	teško	teško	lako	OK	OK
	UPUTSTVO ZA UPOTREBU					
34.	Detalji	super	super	dosta	da	mnogo
35.	Sadržaj	da	da	da	odlican	odlican
36.	Indeks	super	super	super	ne	ne
	INTERFEJSI					
37.	Serijski	opcija	opcija	alternativa	opcija	opcija
38.	IBM kompatibilan	ne	da	da	opcija	ne
	BUKA					
39.	Koliko vrata?	2	2	1	1-2	1

rasklapate štampač, bafera ćete se najverovatnije zauvek odreći!

Pošto jedan znak iz bafera dođe na red, printer će ga analizirati i konstatovati da li se radi o običnom znaku ili o kontrolnom kodu. Ukoliko se radi o znaku, treba još razmisliti o načinu na koji će se on što brže ispisati. Svi štampači koje opisujemo imaju mogućnost štampanja u dva smera uz takozvano logičko traženje: glava će se, pre nego što počne da ispisuje red, pomeriti na onaj njegov kraj koji joj je bliži.

Nailazak na kontrolni karakter signalizira štampaču da iz bafera uzme i sledeće karaktere koji će zaokružiti takozvanu *Escape sekvencu.* Escape sekvenca je niz kontrolnih kodova koji počinju sa „Escape" (27) i koji kontrolišu štampač ili mu naređuju da buduće znakove ispisuje na neki specijalan način. A ti specijalni načini su, uz one koje smo već pomenuli, *Proportional, Emphasized* i *Double strike.* Proporcionalno razmicanje je lako razumeti: širine slova su različite tako da je slovo 'i' daleko uže od slova 'm'. Iako proporcionalno razmicanje znatno doprinosi dopadljivosti teksta, vlasnici računara ga **veoma** retko koriste pošto ga ne podržava praktično ni jedan tekst procesor. Svaki tekst procesor, naime, omogućava korisniku da uravna desnu ivicu teksta pri čemu, jasno, pretpostavlja da su sva slova jednako široka. Nemoguće je, dakle, uravnavati desnu ivicu i koristiti proporcionalno razmicanje. Ukoliko isključite uravnavanje, neki će redovi izgledati prekratko, pa će tekst biti pomalo smešan. Sve u svemu, proporcionalno razmicanje slova nije karakteristika štampača kojoj treba obraćati mnogo pažnje!

Izgleda da malo ko zna u čemu je **tačno** razlika između Double Strike, Emphasized i NLQ štampanja. Autor ovoga teksta mora da prizna da je i on do skora spadao u tu apsolutnu većinu i da se nije ni najmanje zabrinjavao zbog svoga neznanja. Nedavno smo, kroz uputstvo za printere firme *Micro Peripherals,* imali priliku da upoznamo sličnosti i razlike između ova tri načina štampanja: sličnost je, to svi znaju, u tome što glava u sva tri slučaja dva puta prelazi preko svakog reda. Kod *Emphasized* (ili, kako ga ponekad nazivaju, *Bold)* štampanja svaka se tačka ponavlja uz malo **horizontalno** pomeranje, tako da je horiznotalna linija slova T vrlo gusta (ne primećuje se njena tačkasta struktura), dok je vertikalna linija istog slova nešto deblja ali i dalje tačkasta. Kod *Double Strike* štampanja između dva prolaza dolazi do minimalnog **vertikalnog** pomeranja tako da je vertikalna crta slova T gusta, a horizontalna debela ali tačkasta. NLQ štampanje, najzad, podrazumeva **veoma** precizno pozicioniranje glave tako da i vertikalne i horizontalne linije daju utisak celovitosti. U NLQ modu se, osim toga, koriste posebni oblici slova definisani (obično) na matrici 23*16; slova su pripremana tako da imitiraju električne pisaće mašine i štampače sa lepezom. **Svi** štampači pomenuti u tabeli omogućavaju proporcionalno razmicanje, „italic", „emphasized", „double strike" i NLQ" štampanje, kao i pisanje indeksa i eksponenata, mada ne obavezno i u NLQ modu: videćemo, na primer, da uz Canon PQ1080A morate da kupite dodatni ROM da biste štampali kurzivna slova u NLQ modu. Svi štampači, osim toga, omogućavaju korisniku da definiše specijalne znake (na primer, naša

latinična slova), kako u običnom tako i u NLQ modu. Za definisanje NLQ karaktera je, međutim, potrebno dosta RAM-a, pa neke štampače treba dopuniti 6264 CMOS čipovima koji se utiču u predviđena podnožja.

Umesto da se mučite sa definisanjem slova možete da izvadite EPROM-e u koje su upisani oblici znakova (u štampač su obično ugrađena tri do četiri EPROM-a 2764 ili dva EPROM-a 27128; Epsonovi štampači koriste ROM-ove koje ne možete da brišete, ali možete da zamenite EPROM-ima) i da, umesto neke od specijalnih azbuka (22; obično se zamenjuju švedska slova) uvedete naša slova. Iako je za ovakvu hirurgiju potrebno određeno strpljenje, ona nije teška, pa ćemo joj posvetiti malo prostora u nekom od sledećih brojeva „Računara".

Kvalitet otiska

Kvalitet otiska je, verovatno, najvažnija odlika jednog štampača; ako kupite štampač sledeći logiku „nije važno kako dokument izgleda, samo da štampač nije skup", očekujte u bližoj budućnosti nove troškove za nov printer. Vrsta 23 govori o obliku slova, vrsta 24 o slovima koja se protežu ispod osnovne linije kao što su j, g, p, q i slična (ova se slova u stranoj literaturi označavaju kao *descenders)*, dok vrsta 25 opisuje kvalitet podvlačenja. Jasno je da je svaka od ocena u ove tri vrste plod subjektivnog utiska. Vrste 26 i 27 u tabeli, najzad, govore o tome koliko se tačkasta struktura slova primećuje u normalnom odnosno u NLQ modu.

Mnogi proizvođači tvrde da njihovi modeli mogu da proizvedu određen broj „karbon" kopija. Te se kopije, međutim, ne prave pomoću „običnog" indiga — treba kupiti poseban papir sa indigom ili papir kod koga su listovi pravljeni tako da se jedan automatski otiskuje na sledećem. Zato smo uveli vrstu 28 koja prikazuje broj običnih „karbon" kopija koje štampači prihvataju. Taj broj, kao što se vidi, retko prelazi jedan.

Upotreba i uputstvo

Za razliku od računara, štampači zahtevaju određeno održavanje: treba menjati papir, montirati traku, pomerati mikroprekidače, odstranjivati „konfete" koje preostaju u samom štampaču i raditi mnogo sličnih stvari. Vrsta 29 opisuje umetanje listova, a vrsta 30 perforiranog papira, dok vrste 31 i 32 opisuju promenu trake. Vrsta 33, najzad, opisuje situacije koje se neće dešavati previše retko: gužvanje papira. Ma koliko štampač bio dobro izrađen i ma koliko pažljivo sa njim rukovali, događaće vam se da se papir zaglavi. Kod nekih je modela ova operacija sasvim jednostavna, dok kod drugih podrazumeva rasklapanje dobrog dela uređaja.

Većina štampača ima **veoma** slaba uputstva za upotrebu: kontrolni kodovi su samo pobrojani bez detaljnih opisa (34), nema indeksa (36) ni dobrog sadržaja (35), primere treba tražiti svećom, štamparske greške su brojne, engleski toliko slab da i mi to primećujemo (knjige su pisali Japanci) . . . Teško je naći i drugu literaturu koja bi na popularan način upoznavala čitaoca sa funkcijama štampača, pogotovu ako se radi o nekim „specijalnim efektima" kao što je definisanje normalnih i NLQ slova i crtanje slika. Obzirom da u Jugoslaviji postoji sve više „epson" kompatibilnih štampača, „Računari" planiraju da objave kompletno uputstvo za njihovu upotrebu, prilagođeno kako početnicima tako i korisnicima kojima su potrebni dobri indeksi i tabele.

Interfejsi i kompatibilnost

Svi štampači koje prikazujemo imaju ugrađen paralelni (Centronics) interfejs, što znači da ih je lako povezati sa većinom danas popularnih računara koje vredi dopuniti štampačem (za ZX81, „spektrum", „elektron" i „galaksiju" treba dokupiti poseban interfejs, što i nije neobično; čudno je što se interfejs mora kupiti i za QL, računar koji pretenduje na poslovne primene). Većina štampača (37) može da se dopuni serijskim (RS232) interfejsom, koji je obično dopunjen dodatnim baferom. Pitanje je, međutim, koliko su takvi interfejsi potrebni: iako je uvrženo verovanje da serijski interfejs podrazumeva sporije štampanje, nije tačno („usko grlo" u paru računar—štampač **nije** komunikacija već brzina ispisivanja slova na papiru), nema nikakvog razloga da odustanete od paralelnog prenosa podataka, koji ne zahteva posebne troškove. Prodavci kompjuterske opreme su uobičajili da uz štampač besplatno daju i kabl za povezivanje sa vašim računarom, ali se na ovu konvenciju ne možete baš uvek osloniti: za kabl ćete ponekad morati da doplatite 10—20 funti.

Svi su štampači koje prikazujemo, kao što rekosmo na početku napisa, „epson" kompatibilni ali neki (38) omogućavaju direktno povezivanje i sa IBM-ovim kompjuterima. Ukoliko ste, dakle, vlasnik IBM PC i želite da koristite FX80, moraćete da kupite hardverski ili napišete softverski konvertor koda.

Mnogo buke . . .

Štampači proizvedeni pre pet-šest godina imaju jednu užasnu osobinu: prave toliku buku da vas nateruju da iskočite iz kože. Uz ovakav štampač je, pre svega, sasvim nemoguće raditi bilo šta osim nervirati se; njegova je primena, osim toga, iritirajuća kako za vaše ukućane tako i za (ni krive ni dužne) komšije. Obzirom da većina hakera ima običaj da radi noću, nabavka štampača koji neće nikoga buditi i nije tako loša investicija. Skoro bi se moglo reći da neki proizvođači namerno povećavaju buku koju njihovi modeli prave: štampači Star Gemini, na primer, aktiviraju svoju zujalicu kada god prelaze iz „Off Line" u „On Line" i tako standardnoj buci dodaju i stalno udaranje zvona.

Buka se, kao što znate, meri decibelima, ali je za takva merenja potrebno imati odgovarajuću opremu; podatak koji bismo naveli vam, osim toga, ne bi mnogo značio. Zato smo izabrali jednu novu jedinicu za merenje buke koja bi se teško probila u SI sistem: broj zatvorenih vrata. Poslednja vrsta naše tablice, naime, prikazuje broj vrata koja morate da postavite između sebe i štampača kako ne biste čuli njegov rad!

O opštim karakteristikama štampača smo, kao što ste i sami primetili, već previše pisali, pa ćemo preći na konkretnije stvari: u sledećim „Računarima" prikazujemo Epsonove modele FX80, LX80 i LX85, štampače Kaga Taxan i Canon, Mannesmann Tally 85, Smith Corona D200, Panasonic KX-P1092, Star SD10, Seikosha SP1000A, Samleco DX86, Citizen 120D i Brother 1509.

Dejan Ristanović

Naš test

Ploter „Fišer"

igračke za odrasle

Bilo je samo pitanje vremena kada će se neka firma setiti da kombinuje računarske primene i igre i tako stvori računarske igračke. Poslednjih godinu ili dve nekoliko proizvođača je izbacilo na tržište igračke koje se na ovaj ili onaj način dotiču računara. Firma „Fišer" je otišla i korak dalje. Počela je da proizvodi „Computing setove" — pakete delova za računarska pomagala. Najinteresantniji komplet sadrži delove za ploter-skener.

Centralni deo ploterskog kompleta je interfejs, koji i predstavlja osnovu svih Fišerovih „Computing setova". Uz interfejs se dobija i disketa na kojoj se nalaze programi potrebni za rad. Interfejs i softverska podrška se dosta razlikuju u kvalitetu. Dok je interfejs urađen dosta impresivno, programi za rad sa kompletom su prilično primitivni. Glavni mašinski program za upravljanje dat je u obliku DATA linija i programa svaki put mora ponovo da smešta podatke u memoriju, što usporava rad i prilično je nepraktično za korišćenje. Interfejsom se upravlja uz pomoć komandi SYS i USR — dakle direktnim pozivanjem mašinskih rutina. U svakom slučaju, ostaje vam ili da se snalazite sa onim što je dato, ili da sami napravite bolju programsku podršku, ako umete.

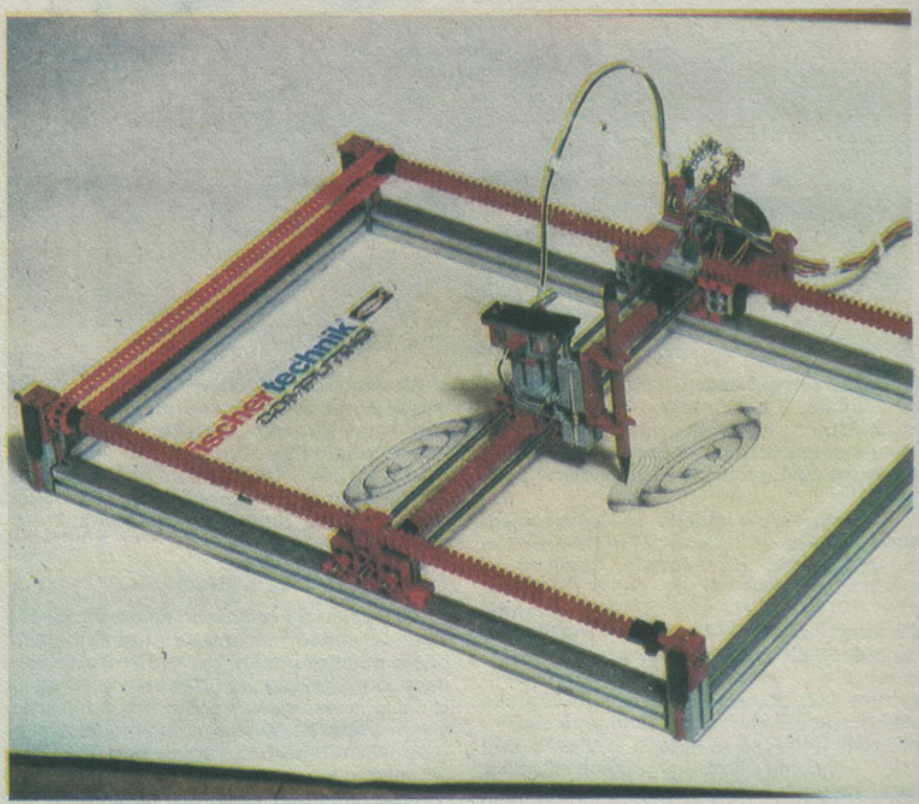
Sastavljanje plotera je, u najmanju ruku, — zanimljivo isto toliko koliko i rad sa njim. To i jeste ono što (pored relativno niske cene) predstavlja glavni kvalitet svih Fišerovih setova. Uz komplet se dobija detaljno i postupno uputstvo koje se veoma lako prati.

Ceo uređaj se nalazi na pleksiglas ploči formata A3. U uglu ploče je elektromotor zadužen za pomeranje po Y-osi a na pokretnoj osovini motor za pomeranje po X osi. Naravno, reč je o veoma kvalitetnim step-motorima, i prenosnim mehanizmima u obliku puža pa je pokretanje veoma precizno.

Glava plotera (ili skenera) je veoma jednostavna. Najvažniji deo je elektromagnet koji se nalazi na samom vrhu glave. Njegova dužnost je da diže i spušta olovku plotera. To je urađeno veoma jednostavno: nosač olovke stoji na dve male opruge koje drže olovku odvojenu od papira. Kada ploter dobije komadu koja nalaže pisanje, uključuje se elektromagnet koji privlači ceo držač olovke nadole. Kada se magnet isključi, opruge ponovo vraćaju olovku u prvobitni položaj.

Veza između interfejsa i plotera je izvedena preko kabla od 24 žice. U uputstvu je posebno naglašeno da sve žice kabla moraju biti tačno određene dužine — ako to ne uradite po uputstvu imaćete velikih problema sa „oživljavanjem" plotera. On, jednostavno, neće hteti da radi.

Pametnom upotrebom možete postići da ovaj „skrpljeni" ploter daje rezultate koji, naizgled, nimalo ne zaostaju za nekim mnogo puta skupljim profesionalnim ploterom. U tome vam pomaže i program koji je dat zajedno sa uputstvom. Pomoću njega možete crtati pravougaonike, krugove i elipse, kao i ogroman izbor slova. Moguće je dobiti slova svih veličina, iskrivljene na sve moguće načine. Slova se mogu i rotirati i pisati u svim pravcima. Nabrojane mogućnosti otvaraju dosta prostora za zaista zanimljiv rad sa Fišerovim Frankenštajn ploterom, ali se njegovi potencijali otkrivaju tek u toku primene, kada pustite mašti na volju. Pri tom se ne plašite igranja. Ploter je upravo za to i namenjen. Svi delovi su zamenljivi, a izvestan broj rezervnih delova, se dobija i u kompletu. Pojedini delovi su identični kao u nekim drugim setovima, tako da nije teško doći do njih kada ustreba.



Igračka ili korisno pomagalo: Ploter „Fišer" u akciji **Fotografija: Ljubodrag Simović**

Drugu primenu ovog uređaja — skener — obraditi nekom drugom prilikom. Transformacija plotera u skener nije komplikovana. Treba samo skinuti olovku i umesto nje staviti „čitač", promeniti softver i dovesti još dve žice do radne glave . . .

Fišerove „igračke za odrasle", kao što je ovaj ploter-skener, omogućuju mnogim ne preterano bogatim zaljubljenicima u računare da nabave i (po neki deo periferijske opreme za kojim odavno žude a nemaju ogromne sume novca da to i plate. Kvalitet opreme je sasvim na visini, ako se izuzmu detalji koji direktno zavise od cene. Tako ovaj Fišerov ploter nikada neće izgledati kao „pravi", iako tako, možda, radi. Jedna od stvari koja je mogla biti bolje rešena je i softverska podrška, ali bar tu imate mogućnosti da sami što-šta uradite, ako znate kako.

Bilo bi lepo kada bi i drugi proizvođači počeli da izbacuju sličnu opremu na tržište i da uskoro budemo suočeni sa mnoštvom sklopivih disk jedinica, štampača, svetlosnih olovaka i grafičkih tablica. Stisnimo palčeve.

Vladimir Krstonošić

„128" na sinklerov način

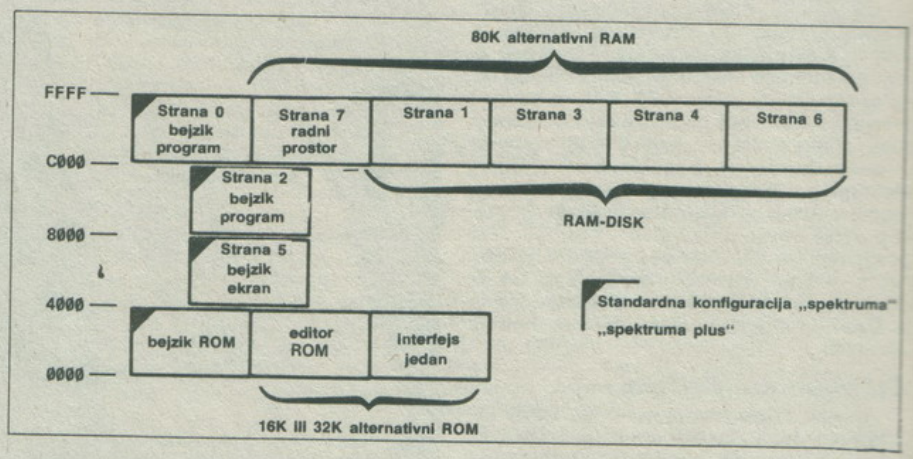
O „spektrumu 128" se sve češće govori i piše. Pri tome, glavnu metu interesovanja predstavlja organizacija memorije i večito pitanje: kako jedan osmobitni procesor izlazi na kraj sa tolikim silnim RAM-om? Odgovor je, kako izgleda, da se dodatnom memorijskom prostoru teško pristupa čak i iz mašinskog jezika.

Blokovi i stranice

Procesor Z80 normalno adresira lokacije između &000 i &FFFF, što ukupno iznosi 65536 bajtova. Taj prostor se može podeliti na četiri bloka od po 16K, pri čemu običan „spektrum" ili „spektrum +" najniži blok rezervišu za ROM, a preostala tri za RAM. „Spektrum 128" tu ne može biti nikakav izuzetak, jer se bazira na istom procesoru. U bilo kom trenutku rada, u sistem mogu biti uključena samo četiri bloka memorije od 16K.

Međutim, ukupan memorijski fond „spektruma 128" iznosi 160K, od čega 32K otpada na dva ROM-a, a 128K na osam blokova RAM-a od 16K. Memoriju tako možemo zamisliti kao knjigu sa deset stranica, pri čemu svaka strana sadrži 16K informacija. Procesor Z80 može istovremeno da čita samo četiri strane.

Prve dve strane (ROM) tretiraćemo posebno, a preostalih osam strana RAM-a numerisaćemo od 0 do 7.



Klip-klap

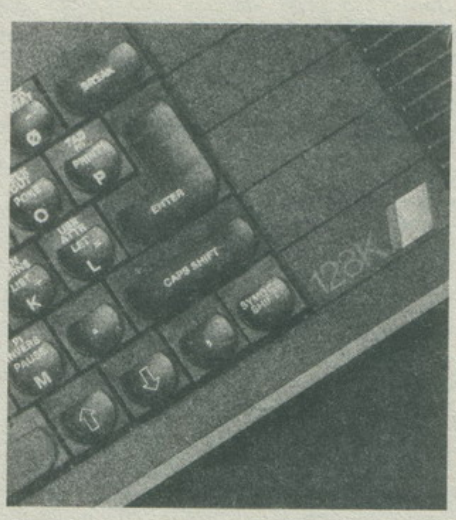
Jedan od ROM-ova je standardni bejzik ROM, dok drugi sadrži ekranski editor, rutine za rad sa zvukom itd. Po potrebi, procesor uključuje jedan ili drugi, ali uvek se ROM nalazi u najnižem delu memorijske mape, između adresa &0000 i &3FFF. Razume se, na isto mesto dolazi i ROM interfejsa jedan, ako je uključen u sistem. „Spektrum 128" prosto bira koji mu ROM kada treba, i sa njim onda radi.

Međutim, kako stoji stvar sa RAM-om? Memorijska mapa nudi tri bloka od 16K, počev od adresa '4000. Bilo bi logično očekivati da bilo koja od osam stranica RAM-a može doći u bilo koji od tri bloka mape. A nije tako. Blok između '4000 i '7FFF rezervisan je isključivo za RAM 5, a blok između '8000 i 'BFFF — samo za RAM 2. U vrh mape ('C000 – 'FFFF) može se smestiti bilo koja stranica RAM-a (pa čak ponovo RAM 2 ili RAM 5)! Obično je, međutim, na tom mestu RAM 0, što sve odgovara standardnoj konfiguraciji „spektruma" i „spektruma +".

Noviteti nastaju kada korisnik pozove ekranski editor. Editoru je potreban dodatni prostor za pamćenje sadržaja ekrana, a u tu svrhu se koristi RAM 7. Jednostavno, RAM 0 se ukloni iz vrha memorijske mape, i tu se dovede RAM 7. Naravno, čim korisnik napusti ekranski editor, RAM 0 će se vratiti na svoje mesto, bez gubitka sadržaja. Sva ova preklapanja vrši sam operativni sistem, bez bilo kakve intervencije korisnika.

Ram — Tape

Preostale su još stranice 1, 3, 4 i 6, koje, bar što se tiče operativnog sistema, kao da i ne postoje. Sinkler je tih 64K zamislio kao



„ram-disc", tj. kao medijum za brzo odlaganje i učitavanje podataka. Bolji naziv bi, međutim, bio „ram-tape", jer je pristup isključivo sekvencijalan: bajtovi se „snimaju" kao na kasetu, jedan za drugim, a onda se u istom tom redosledu moraju i čitati. Sve to, razume se, ekstremnom brzinom.

Zanimljivo je da se operacije sa ram-diskom obavljaju uobičajenim bejzik naredbama **SAVE** i **LOAD (VERIFY** je, jasno, nepotrebna). Korisnik jedino mora da otvori odgovarajući kanal za komunikaciju.

Ula — diktator

Jedna od koristi ram-diska bila bi čuvanje sadržaja ekrana i brzo smenjivanje slike. To „brzo" međutim, i nije tako brzo (u svakom slučaju ne trenutno). Bolje je tada zaobići neredbe **SAVE** i **LOAD,** i preslikavati blokove bajtova mašinskom naredbom **LDIR.**

Ipak, postoji i specijalan način zamene slike, dostupan jedino iz mašinskog jezika. Setovanjem posebnog indikatora, ULA se može navesti da čita sliku ne sa uobičajene adrese '4000, već iz najvišeg bloka memorije, tamo gde spada RAM 7. To je zaista trenutna operacija, ali krije u sebi i malu zamku. Vlasnici „spektruma" svakako znaju da ULA ne dopušta procesoru pristup bloku video memorije u fazi čitanja slike. Z80 je tu doslovce nemoćan, jer mu ULA prosto prekine klok, i tako ga efikasno onesposobi za neko vreme. To dosta usporava sve programe koji se izvršavaju iz najnižeg dela RAM-a u verzijama „spektrum" i „spektrum +". Sada, na „spektrumu 128", ULA „teroriše" i na višim adresama, što i nije baš najprivlačnije rešenje.

Čini se da je, ipak, najjača strana „spektruma 128" njegov muzički čip. Kad programeru dojade RAM-ovi, ROM-ovi i njihova „organizacija" može potražiti mir uz zvuke koje uspe da isprogramira. Bejzik mu, naravno, ni u tu svrhu neće biti od bog zna kakve koristi.

Jovan Skuljan

Odjeci

stolari i računari

Ko se boji domaće elektronske industrije?

Nakon naše vesti o proizvodnji serije računara „sperry PC 500" u „Računarima 12", u domaćoj medijskoj sili zaređala su mišljenja o nepotrebnim i potrebnim pokušajima „šrafciger" industrije. To je saradniku „Računara" Branku Hebrangu dalo povoda da se još jednom oglasi na istu temu.

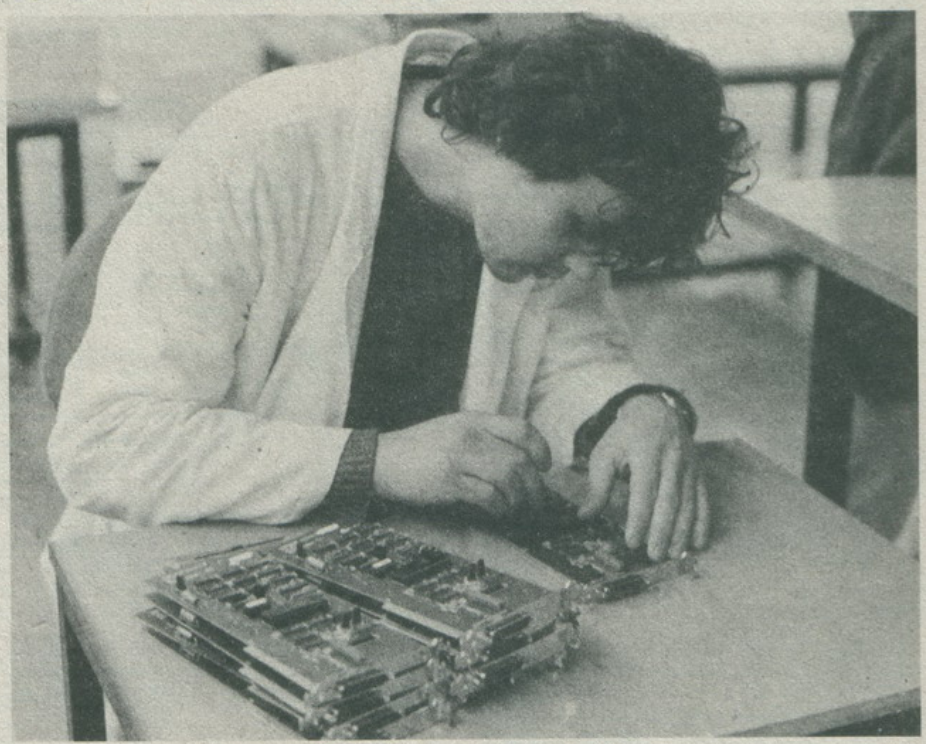
Iznenađujuće zanimanje domaćih medija izazvala je vijest o uključivanju Drvne industrije „Gaj" iz Podravske Slatine u tokove elektronske industrije. Pojedince je najviše začudilo sklapanje prve serije od 64 računala marke „sperry PC 500" u novoosnovanoj osnovnoj organizaciji udruženog rada „Elektronika"—„Gaj", a u suradnji sa „Sperry Univacom" i zagrebačkim „Infosistemom", o čemu su pisali neki listovi, a jugoslavenska javnost obaviještena je putem naše revije „Računari" broj 12. Taj posljednji naslov, čini se, pobudio je posebnu pažnju novinskih izvjestitelja, pa je „Gaj" završio na novinskim stupcima glasila širom zemlje, a ni televizija nije u tome zaostala.

Zabrinuti čipobrižnici

„Stolari prave kompjutore" zaključili su u jednom visokonakladnom listu, čudeći se kako je to moguće, zabrinuti za budućnost dotične tvornice, ali i domaće elektronske industrije. Bilo je to tipično reagiranje u nedostatku potpunijih informacija o mogućnostima proizvodnje računala, informatičke opreme i profesionalne elektronike u nas. Da su informatički znatiželjnici okrenuli list više u „Računarima 12", mogli su pročitati obrazloženje kibernetičara Branislava Bakića o mogućnostima proizvodnje računala u zemlji. Dakle, Bakić je ustvrdio da nam na raspolaganju stoje četiri načina: vlastiti razvoj, uključivanje u OEM tržište, kooperacija i licencna proizvodnja. Pri tome se rad s originalnim elektroničkim sklopovima (Original Equipment Manufacturer) kod nas poistovjećuje sa „šrafciger" industrijom.

Budući da kasnimo desetak godina za zbivanjima na tržištu malih računala danas ne možemo puno birati kako ćemo razvijati domaću proizvodnju računala. Postoje dva, tri veća proizvođača koja su kreirala vlastiti razvoj, no ni oni nisu usvojili masovnu proizvodnju akamoli tržište. Za to postoje razlozi. Uz te veće proizvođače razvijali su se manji, koji su se postepeno uključivali u svijet elektronike. Neki od njih će se održati, neki će ubrzo proizvodnju preorijentirati na nešto drugo. Slično je učinio i „Gaj" otvarajući vrata tržišta uz suradnju iskusnijih partnera, ali ne samo u proizvodnji računala. Primjerice, sada su u suradnji s partnerom iz NR Mađarske proizveli registracijske magnetofone, uređaje za zapisivanje telefonskih razgovora s nekoliko kombinacija ulaza i izlaza.

Jasno je da „Gaj" prvenstveno i dalje ostaje prerađivač drveta, među najvećima je u Hrvatskoj, a u proizvodnji namještaja koristi i NCR strojeve, čime se mogu podičiti rijetki domaći prerađivači drveta. U „Ekonomskoj politici" broj 1772 Dragiša Bošković, pišući o dilemama u domaćim medijima, zabilježio je: „Zato se ovde od sveg srca daje glas za „Gaj" iz Podravske Slatine kao ponuđača računara i unapred obećava da će taj glas biti dat svakome ko nađe načina da od dobavljenih delova nekako sklopi kompjuter i doturi ga jugoslavenskom kupcu. Činimo to, naravno, zato što nemamo mnogo izgleda da se problem reši kao što bi ga rešila zemlja doista usmerena da se preko dinara što slobodnije pretvorljivog u svetski novac i uz normalnu zaštitu na granici, otvori prema svetu. Zato ostaje samo taj okolišni put umnožavanja stotina, poželjno hiljada najraznovrsnijih ponuđača kompjutera koji kreću u posao željni zarade ali u međusobnom nadmetanju obaraju cene. Doslovno tako nešto, uostalom, upravo se zbiva u Indiji, na čijem je tržištu slično nadmetanje prepolovilo prošlogodišnje cene."



Bura u čaši vode: Sklopovi s OEM tržišta u podravskom „Gaju" koriste se i za proizvodnju profesionalne elektronike

Kad se slegne prašina

Kad se malo obriše podignuta prašina oko „stolara i računala"otkriva se istina da je osnovni cilj ipak kompjutorizacija proizvodnje. Tržišna ekspanzija malih računala treba podići informacijsko znanje na višu razinu kako ne bi trebali današnji školarci u tvornici još jednom učiti kako treba raditi. Dobro organizirana poduzeća lakše održavaju korak s konkurencijom, ulažu u kadar i razvoj, i s većom vjerojatnošću će uspješno organizirati proizvodnju i u 21. stoljeću.

Ilustrativno je da je 1972. godine na savjetovanju Informatika u SR Hrvatskoj i njezin daljnji razvoj konstatirano da treba pojačati ulaganja na tom području, ako ne želimo još više zaostati. Onda je istaknuto da u svijetu u organizaciji znanstvenih istraživanja i razvojnih djelatnosti troškovi za informaciono-dokumentaciono-komunikacijski sistem iznose 25 posto troškova od ukupnih ulaganja u istraživanje i razvoj. Kod nas se tada za te potrebe trošilo 50 puta manje.

Pet godina kasnije u „Impulsu", poslovno-interesnoj zajednici proizvođača telekomunikacija, elektronike, kabela i računara utvrdili su da su ulagana znatna sredstva, kroz proviziju, u razvoj tržišta opreme stranih proizvođača, dok je podrška domaćoj industriji gotovo u potpunosti

izostala. Domaći proizvođači su za uvoz repromaterijala plaćali visoke carine, a strani proizvođači imali su pogodniji carinski režim. Uvoz repromaterijala, pored deviznih teškoća, bio je znatno otežan ili onemogućen zbog nerazjašnjenih pojmova što je uopće repromaterijal za proizvodnju opreme za automatsku obradu podataka, pri čemu je uvoz sklopova i funkcionalnih jedinica, koje i strani proizvođači nabavljaju na OEM tržištu, bio praktički onemogućen. U takvim okolnostima domaći proizvođač je na domaćem tržištu bio u nepovoljnijem položaju od stranih proizvođača, što nigdje u svijetu nije slučaj, zapisano je tada u jednoj informaciji „Impulsa".

U to je vrijeme samo za četiri godine uvezena oprema za automatsku obradu podataka vrijedna 220 milijuna dolara, ondašnjih četiri milijarde dinara. U „Impulsu" su procijenili da smo za proviziju tada dali 1,2 do 1,6 milijardi dinara, a da smo domaće tržište praktično besplatno poklonili stranim proizvođačima.

Takva situacija na tržištu velikih kompjutorskih sistema nije mogla biti baza za uključivanje zemlje u svijet informatike, pa smo danas potpuno inferiorni na tržištu mikroelektronike.

Ipak se kreće

Velika računala ipak su nekako ulazila u urede u vrijeme investicijskog zamaha, a kada je prevladala minijaturizacija u elektronskoj industriji i kada je došlo vrijeme malih računala, demokratizacije informatičkog osnovnog sredstva, administracija je mala računala svrstala u red — luksuzne robe. Zakonom o privremenoj zabrani korištenja društvenih sredstava za financiranje neproizvodnih i neprivrednih investicija, informatička je oprema ubrojena u „administrativno-birokratsku uredsku opremu". To je onemogućilo nabavu te opreme, razvitak kompjutorske industrije, pa je počelo razdoblje šverca malih računala koje i danas traje. Tek promjenom propisa, potkraj 1984. godine, dozvoljen je uvoz malih računala s vrijednosnim ograničenjem u početku od 40.000 dinara, poslije 60.000 a sada 90.000 dinara. Uvoz dodatne opreme, pri tome, nije bio posebno reguliran.

Zatvorenost tržišta odrazila se i na školski sustav koji nije na vrijeme reagirao i uveo informatiku kao obvezatni predmet. Luk i pilica i danas su još didaktičko sredstvo osnovnoškolaca, a kompjutorsko obrazovanje zaposlenih u privredi nije ni u začetku.

No, i u takvim uvjetima razvila se kakva-takva produkcija računala. Spomenut ćemo najpoznatije domaće proizvođače.

Prvi su 1981. godine s računalima krenuli u varaždinskom PEL-u i beogradskom „Ivi Lola Ribaru". PEL je u konačnici napravio bolji posao i njegova računala tipa „orao" prihvaćena su za korištenje u školama u SR Hrvatskoj. Pravi boom na tržištu svojevremeno je napravljen prvim domaćim računalom u kitu, s „galaksijom", koja je dosad postigla, prema raspoloživim podacima, najveću seriju. „Školski servis" iz Zagreba proizvodi malo računalo „univerzus" kompatibilno s „apple II", a „Ivasim-Elektronika" iz Ivanić-Grada prodaje računalo „ultra", koje naziva univerzalnim kompatibilnim računalom. „Iskra" uz poslovne sisteme proizvodi i malo računalo HR-84, dok „Javor", trgovinska radna organizacija, OOUR „Informatika i elektronika" Bitolj proizvodi računalo „marta". Novosadski „Novokabel" proizvodi kompjutor ERA, a zagrebački tehničar „tera". Ima i drugih proizvođača, sve do privatnih, ukupno dvadesetak. Uz njihovu produkciju domaće tržište obogaćeno je i s nekoliko tipova računala proizvedenih u kooperaciji sa stranim tvrtkama, a putem zastupništva stranih proizvođača u nas se mogu kupiti i neka računala s konsignacije. Inače, svjetska ponuda malih računala obuhvaća više od 300 modela.

Ponuda opreme za računala već je slabija, a programska podrška ujedno je i najslabija točka svih domaćih proizvođača. Zaokupljeni usvajanjem hardvera, fizičkih dijelova računala, zapostavili su programsku podršku, pa poduzeća mogu kupiti male sisteme ali ne i programe za određene djelatnosti. Pri svemu tome treba znati da je naša industrija informatičke opreme među najmlađima u zemlji, a da su fundamentalna istraživanja na tom području vremenski dugotrajna i vrlo skupa. To je samo jedna dimenzija tog složenog problema.

Razbiti žabokrečinu

U Jugoslaviji zbog poznatih privrednih teškoća sve se manje ulaže u razvoj. Tako se 1978. godine za razvoj odvajalo 1,07 posto nacionalnog dohotka, a 1983. samo 0,91 posto. Znanstveni radnici u takvim prilikama nisu ni dovoljno plaćeni, zato se bave sivom ekonomijom, projektima i istraživanjima za honorar. To s razvojem nema nikakve veze.

U domaćim privrednim razvojnim jedinicama ima deset posto, u Sloveniji 20 posto, istraživača. U susjednoj Italiji ima 40 posto, a u SAD-u 70 posto od istraživača spomenutih zemalja.

Takva kretanja zavisna su od stupnja razvoja privrede i strukture faktora proizvodnje. U zanatskom načinu proizvodnje rad je zastupljen sa 70 posto, kapital s 20 a znanje sa skromnih deset posto. U industrijskoj proizvodnji rad je smanjen na 25 posto, udjel kapitala povećan je na 50 posto a znanja na 25. U informacijskom društvu, društvu sutrašnjice, rad će sudjelovati s pet do deset posto, kapital s 25 posto, dok će ostatak zauzimati znanje, informacije. U takvom postindustrijskom društvu čovjek bi trebao živjeti ugodnije, s više slobodnog vremena za zabavu, rekreaciju, učenje.

Za takvu budućnost vjerojatno danas u nas ne postoji dovoljan fond znanja. Ne samo u privredi već ni na visokoškolskim institucijama i institutima. Dok se ne stvori takva kritična masa znanja — učenjem, istraživanjima, transferom i usvajanjem tuđih spoznaja i iskustava — neće se moći ni organizirati proizvodnja s manjim udjelom rada. Proizvodi iz takvih tvornica teško će nalaziti kupca.

Treba unijeti nemir u naš mir stagnacije i žabokrečine, uskih horizonata, bavljenja samim sobom i raspodjelom postojećeg, svojevrsnog zatvaranja očiju i fatalizma pred budućnosti, zapisao je dr Drago Gorupić pišući o znanstveno-tehničkoj revoluciji i zadacima poslovodnog osoblja. Treba se otvoriti budućnosti, treba nam približiti budućnost informacijski, analitički i sukobom mišljenja, da bismo napokon počeli o njoj misliti i za nju se pripremati. Samo, to treba činiti brzo, jer nas svijet neće čekati.

Branko Hebrang

U sledećem broju

Novi projekat

TURBO DOS

Prvi nastavak kompletnog hardversko-softverskog projekta disk inrerfejsa za „spektrum" sa paralelnim interfejsom za štampač i džojstik i monitorskim izlazom. Interfejs je opremljen i sopstvenim ROM-om od 8 K koji radi „u senci" osnovnog ROM-a i potpuno je kompatibilan sa mikrodrajvom. Program od 32 K TURBO DOS učitava za samo sedam sekundi.

Nova serija

POLIKLINIKA C

Stručni saradnick „Računara" iz Njujorka prof. dr Radomir A. Mihajlović pripremio je opsežnu školu jezika C — polikliniku u kojoj će „bolesnike" od bejzika, paskala i fortrana pokušati da „izleči" ovim sve popularnijim programskim jezikom koji, de facto, postoje svetski standard.

Umetak

Dejan Ristanović

ŠTAMPAČI

Detaljan priručnik za upotrebu, programiranje i adaptaciju matričnih štampača na YU standarde

LOTO NA RAČUNARU

Žarko Vukosavljević, ekspert za loto sisteme, pripremio je za vlasnike „spektruma" program koji im utorkom uveče, sigurno, neće donositi milione, ali će im pomoći da racionalno ulažu svoj novac i igraju skraćene sisteme samo sa svojim srećnim brojevima.

HAKERSKI VODIČ MINHENA

Početkom maja Zoran Životić obišao je sve poznatije prodavce IBM PC/XT klonova u pojasu između Minhena i Frankfurta. Adrese, izbor, cene, uslovi nabavke i, konačno, kako na PC reaguju YU carinici.

Dejan Ristanović

Dejanove pitalice

Svi igraju Loto

Tvrdnju Jugoslovenske lutrije da je Loto sve popularniji ilustruje i naš šesti nagradni zadatak, koji je izazvao za sada najveće interesovanje; u predviđenom roku smo primili 327 odgovora, od toga svega 14 pogrešnih. Iako je ogromna većina rešavača koristila računar, nagrade su zaslužili čitaoci koji su pomalo zaposlili i svoj mozak!

Podsetićemo se, kao i obično, najpre zadatka: trebalo je pronaći Loto kombinaciju čija je oznaka 10.000.000 (10 miliona) kao i oznaku kombinacije 5, 8, 10, 11, 17, 25, 26. Zadatak se može rešiti prilično jednostavno primenom takozvane „grube sile": sedam koncentričnih FOR-NEXT petlji koje prebrojavaju kombinacije će, sa strane programera, biti sasvim dovoljne. Ovako je rešenje, sa druge strane, *vrlo* neprijatno za računar: ako program napišete na bejziku, „spektrum" će ga „žvakati" četiri do pet dana! Rešenja dobijena na ovaj način nas navode da ubuduće na kraju ove rubrike pišemo: *redakcija „Računara" ne snosi odgovornost za kvarove nastale u toku rešavanja ovoga zadatka.*

Neki su se rešavači dobro dosetili da upotrebe jedan od popularnih bejzik kompajlera; na taj je način vreme izvršavanja smanjeno neka 3—4 puta, ali je i dalje predugačko. Bolji programeri koji su operisani od matematike pisali su program na asembleru, dobijajući tako rutine koje daju rezultat prihvatljivo brzo — za „svega" par sati. Do rešenja se, međutim, može doći za isto toliko sekundi, primenom programa poput onoga koji objavljujemo.

Kako bismo ovaj zadatak rešili bez primene računara? Naučićemo najpre da izračunamo ukupan broj kombinacija u igri loto: sedam se brojeva može izdvojiti iz 39 na „39 nad 7" načina. „N nad K" računamo iz formule

$$\left|\begin{matrix}N\\K\end{matrix}\right| = \frac{N*(N-1)*\ldots*(N-K+1)}{1*2*3*4*\ldots*(K-1)*K}$$

Da vidimo koliko od ovih kombinacija počinje brojem 1: ovo se izračunavanje svodi na pitanje koliko se kombinacija može napraviti izborom 6 brojeva od 38, tj. na „38 nad 6" načina. Obzirom da je „38 nad 6" jednako 2.760.681, što je dosta manje od 10 miliona, kombinacija sa traženom oznakom ne počinje jedinicom. Ako bismo pretpostavili da kombinacija počinje dvojkom, njen bi redni broj mogao da bude najviše „37 nad 6" ili 2.324.784. Nastavljajući sličan postupak, nalazimo da je „35 nad 6" jednako 1.623.160 što, sabrano sa 2.324.784 i 1.947.792 daje 8.656.417, broj koji je manji od 10 miliona. Ako bismo, međutim, pretpostavili da dobitna kombinacija počinje šesticom, ovom bismo zbiru dodali „34 nad 6" tj. 1.344.904, dobivši 10.001.321, broj veći od 10 miliona. Tražena kombinacija, dakle, počinje brojem 5! Ponavljajući sličan postupak dobijamo da kombinacija sa oznakom 10.000.000 glasi *5, 27, 29, 30, 34, 35, 36,* što je ujedno i rešenje prvog dela zadatka.

Drugi deo zadatka može da se reši nešto brže, bez mnogo probanja. Kombinacija koje počinju brojevima 1, 2, 3 i 4 ima „39 nad 7" — „35 nad 7" ili 8.656.417 (ovaj smo broj već dobili, doduše na nešto drugačiji način). Zatim računamo koliko ima kombinacija koje počinju sa 5 dok im je drugi broj 6 ili 7; dobijamo „34 nad 6" — „32 nad 6" ili 438.712. Ponavljajući ovakav postupak i sabirajući međurezultate dobijamo da oznaka kombinacije 5, 8, 10, 11, 17, 25, 26 glasi *9124166.*

Pažljivo pregledavši prispele programe, pokušali smo da izaberemo najkraći i dodelimo mu prvu nagradu. U najužoj konkurenciji su se našla rešenja *Jelene Grujić* iz Subotice i *Romana Zoltana* iz Zrenjanina. Obzirom da je program drugarice Grujić bio nešto kraći, otkucali smo ga i detaljno testirali, primetivši mnogo sitnica koje bi bile interesantne za profesora Slavića i njegovu rubriku „To može i bolje": iako je program tačno generisao kombinaciju sa oznakom 10 miliona, kombinacije sa oznakom 1, 2, 3, . . . 30 su bile jednake zbog kumulativne računske greške; N nad K je, uz ostalo, računato na numerički neprihvatljiv način! Tako je prva nagrada u iznosu od 10.000 dinara pripala *Zoltanu Romanu* čiji (nešto prerađen) program dajemo na slici.

Preostale tačne odgovore smo ubacili u tradicionalni koverat iz koga smo izvukli pisma *Bojana Stojanovića* iz Smedereva i *Ivana Vrbovčana* iz Kapele kojima su, respektivno, pripale druge (5.000) i treća (3.000 dinara) nagrada.

```
 10 REM            L O T O
 20 REM
 30 REM Prema programu Zoltana Romana
 40 REM
 50 REM          "Racunari 15"
 60 REM
 70 DIM A(10)
 80 CLS
 90 PRINT "1. Oznaka po kombinaciji"
100 PRINT "2. Kombinacija po oznaci"
110 PRINT
120 INPUT A
130 ON A GOSUB 310,170 ELSE END
140 END
150 :
160 REM Kombinacija po oznaci
170 INPUT "Oznaka "B:M=39:G=0
180 FOR W=6 TO 1 STEP -1
190 H=7-W
200 G=G+1:Q=M-1
210 GOSUB 400
220 IF T<B THEN B=B-T:M=M-1:GOTO 200
230 M=M-1:A(H)=G
240 NEXT W:A(7)=G+B
250 PRINT "KOMBINACIJA: ";
260 FOR H=1 TO 7:PRINT A(H);:NEXT
270 PRINT
280 RETURN
290 :
300 REM Oznaka po kombinaciji
310 S=15380937
320 INPUT "Kombinacija "A(1),A(2),A(3),
    A(4),A(5),A(6),A(7)
330 M=39:W=7
340 FOR J=1 TO 7
350 Q=M-A(J):GOSUB 400:W=W-1:S=S-T
360 NEXT
370 PRINT "Oznaka za datu kombinaciju
    je ";S
380 RETURN
390 :
400 IF Q<W THEN T=0:RETURN
410 T=1:IF Q=W THEN RETURN
420 R=Q-W:K=1:IF R<W THEN R=W
430 FOR I=R+1 TO Q:T=T*I/K:K=K+1:NEXT
440 RETURN
```

Jedan programerski problem

Osmi nagradni zadatak ispunjava zahtev čitalaca koji stalno traže čisto programerske probleme. Promenljivima A, B, C, D i E su dodeljene vrednosti koje treba sortirati. Program koji će obaviti takvo sortiranje ***ne sme*** *da sadrži cikluse (eksplicitne ili implicitne), GOTO naredbe koje pokazuju „unazad" i bilo kakve strukture* ***osim*** *IF . . . THEN . . . ELSE naredbi. Ne smeju se, osim toga, dimenzionisati nikakvi nizovi niti koristiti skalarne promenljive* ***osim*** *pomoćne promenljive Z. Samo se po sebi razume da čitav program treba da bude napisan na nekom višem jeziku (na primer bejziku ili paskalu) i da ne sme koristiti naredbe PEEK, POKE i USR niti specifičnosti računara.*

Pošto napišete program, prebrojte IF naredbe i upišite njihov broj u kupon koji dajemo na kraju ove stranice. Kupon, zajedno sa programom, pošaljite na adresu „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd tako da pristigne pre ***15. juna 1986.*** *Među programima koji rešavaju traženi problem primenom minimalnog broja IF naredbi će biti izvučena prva (10.000), druga (5.000) i treća (3.000) novčana nagrada.*

Ispunićemo još jedan zahtev naših čitalaca: nije više neophodno slati originalne kupone. Ukoliko ne želite da oštetite svoj primerak „Računara", prepišite ***kompletan*** *kupon na početak prve stranice rešenja tako da bude uokviren i jasno uočljiv.*

Pet skalara moze da se sortira primenom najmanje _____ IF naredbi.

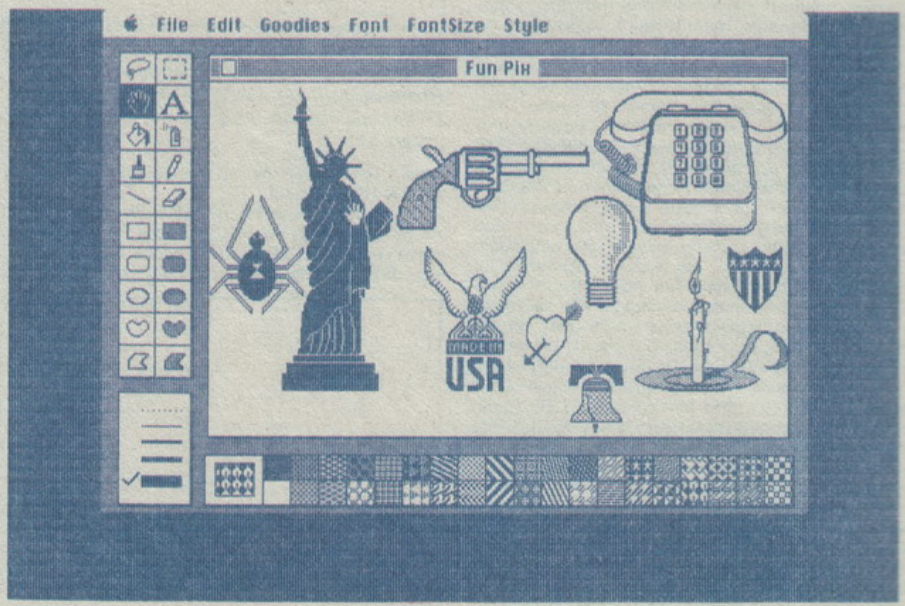
Ime i prezime _____

Adresa _____

Mesto _____

svi „atarijevi" programi

Na tržištu hardvera i softvera, zahvaljujući računarima iz serije „atari st", poslednjih meseci vlada neuobičajena živost. Novi naslovi uslužnih, sistemskih i jezičkih programskih paketa izlaze dosiovce svakoga dana i „atari" biblioteka je već narasla na gotovo 500 programa. Uz niz noviteta iz „atari" tabora, „Računari", verovatno prvi u svetu, objavljuju spisak svih „atari st" programa prema podacima koji su nam bili dostupni do sredine aprila.



Operacijski sistem TOS računala serije „atari st", kao što je poznato nije se dosad isporučivao u ROM-u, nego na tzv. sistematskoj disketi, čiji sadržaj je nakon svakog uključenja računala trebalo upisati u RAM memoriju. Sama firma Atari je ovaj potez objašnjavala potrebom da se OS prije „ugradnje" u ROM dobro istestira u praksi. Zlobnici (čitaj: konkurencija a la Commodore) su, pak, tvrdili da Atari ima problema zato što u operacijskom sistemu koristi grafički sistem GEM firme Digital Research. Ovaj grafički sistem je, po vanjskom izgledu (ali ne i po sastavu), vrlo sličan grafičkom sistemu kojeg koristi firma Apple na svojem računalu „mekintoš". Pronele su se glasine da je Apple zbog toga tužio Digital Research sudu i da se — pošto je vjerojatno da će tužbu i dobiti — TOS, koji koristi GEM, neće moći koristiti na računalima serije ST.

Mali lopovi — velika dobit

Istina je, vjerojatno, negdje u sredini. Točno je da je firma Apple prigovorila Digital Research-u zbog vanjske sličnosti GEM-a i njihovog „mekintoša", ali pošto i sama firma Apple namjerava kod Digital Research-a (kao najvećeg proizvođača sistemskih programa za mikrokompjutere, koji je stvorio, na primjer, i CP/M) naručiti neke sistematske programe za svoje nove tipove računala, vjerojatno je ocijenila da nije uputno baš previše „gnjaviti" novog partnera. Tako su se Aplle i Digital Research, na kraju, i bez suda dogovorili da se ipak izvrše neki „kozmetički" zahvati na GEM-u i ovaj više ne bude tako „napadno" sličan grafičkom sistemu „mekintoša".

Ovakav postupak firme Apple uzbudio je američku (i svjetsku) scenu mikroračunala (pa čak i šire, uključivo čitavu elektroniku i tehniku općenito). Naime, silan razvoj elektronike, a naročito mikroračunala posljednjih godina, bio je moguć zbog niza malih poboljšanja i unapređenja koja su razne konkurentske firme ugrađivale u svoje proizvode, ne bi li se nekako razlikovali od „susjedovih" i time lakše reklamirali i prodavali. Jasno je da svaka mala firma ne može svake godine smisliti neki novi, revolucionarni proizvod, kojim će zasjeniti svoje konkurente i ostvariti dovoljan prihod da financira razvoj slijedećeg novog, revolucionarnog proizvoda. Tako im ne preostaje ništa drugo nego da proizvodima svojih konkurenata dodaju neke male izmjene i poboljšanja. Ovakav „razvoj malim koracima" je prešutno prihvaćen čak i u takvom društvu kao što je američko, koje je i zasnovano na što većem profitu iz pojedinih proizvoda. Ako je i bilo nekih očitih kopiranja, nikome se nije isplatilo da rješava stvari sudom, zbog vrlo sporog, kompliciranog i skupog američkog sudskog sistema.

Međutim, sada je prvi puta neka velika firma (a Apple to jeste), sa velikim financijskim zaleđem, najavila da želi zaštititi ne samo svoje sistemske programe nego i sam njihov „vizuelni izgled". To je izazvalo uzbunu među mnogobrojnim malim firmama, koje su već užurbano radile na nekim poboljšanjima i sitnim izmjenama svojih grafičkih sistema. Svaka od njih se našla u situaciji da dobro promisli ima li smisla ulagati znatna financijska sredstva u razvoj sistemskih programa za koje više nije sigurno da li će se moći i smjeti prodavati. Veliki broj takvih firmi, na kojima, u stvari, i počiva snaga razvoja američke tehnologije, počeo je javno izražavati neslaganje s takvom politikom firme Apple, pa čak i otkazivati poslove s njom. Također, u mnogim stručnim časopisima pojavljivali su se uvodnici i pisma čitalaca, koji su zvonili na uzbunu „za spas inventivnog duha američke tehnologije", „za ugroženi razvoj tehnologije i općeniti prosperitet društva" itd. Neki od njih su „iščeprkali" i podatak da je i grafički sistem „mekintoša", u stvari, kopija grafičkog sistema razvijenog u laboratorijama firme XEROX.

Vjerojatno je i to utjecalo na odluku firme Apple da izgladi spor s Digital Research-om. Iako pojedinosti njihovog dogovora nisu sasvim poznate široj javnosti, izgleda da je Apple ipak uspio prisiliti Digital Research da poduzme određene „kozmetičke" zahvate na GEM-u. Međutim, kao što i svaka medalja ima dvije strane, Digital Research tvrdi da te izmjene u GEM-u ne moraju nužno biti korak unatrag u kvaliteti i korisničkoj prihvatljivosti programa. Navodno, sada su im širom otvorena vrata da znatno preprave i poboljšaju GEM i učine ga stvarno najboljim grafičkim sistemom — kao što su i reklamirali pri njegovom najavljivanju.

Najzad u ROM-u

Neupućeni čitalac će odmah pomisliti da ovakav razvoj događaja mora imati posljedice i za firmu Atari, koja koristi upravo spornu verziju GEM-a na svojim računalima serije ST. Međutim, firma Atari hladnokrvno izjavljuje da je od Digital Research-a kupila (i platila) GEM i da je se dalje ne tiču i ne obavezuju nikakvi dogovori (ili problemi) Digital Research-a s trećim licima. Da potkrijepi takav svoj hladnokrvni stav, upravo ovih dana pustila je u prodaju i ROM-verziju operativnog sistema TOS, koja sadrži originalni, neizmjenjeni GEM. ROM-verzija se prodaje u kompletu od 6 promova (Programmabile Read-Only Memory) po 32 K, ukupno 192 K. Cijena je, zajedno s besplatnom montažom u prodavnici Atari računala, 29 funti ili 99 DM.

To je smiješno niska cijena, čak manja nego cijena odgovarajućeg broja eproma koje su neki od korisnika računala serije ST već isprogramirali i ubacili u prazna podnožja predviđena za ROM-memoriju s operacijskim sistemom. Ovakav potez firme Atari naišao je na nepodijeljene simpatije svih korisnika računala serije ST, od kojih su neki bili i ogorčeni zbog toga što su kupili 520 ST po cijeni po kojoj se tri mjeseca kasnije prodavao 520 ST+.

Ugradnjom operacijskog sistema u ROM-u, računala tipa 260 ST imaju na raspolaganju za programa i podatke oko 478 K RAM memorije, pošto se od sveukupnih 512 K mora odbiti 32 K za video-memoriju i oko 2 K za sistemske varijable operacijskog sistema. To vrijedi i za dosad prodata računala tipa 520 ST, koja se više ne proizvode. Njih je zamijenio model 520 ST+, kod kojeg već i prije nije bilo problema s memorijom, budući da ima ugrađenih čitavih 1 MB (tj. 1024 K) RAM-a.

Zbog teškoća u proizvodnji i isporuci svojih kolor-monitora SM1224, firma Atari prihvatila je ponudu francuskog poduzeća Thomson, i u svojim kolor-kompletima prodaje njihov kolor-monitor CM-3638ZAR. On je, doduše, za oko 10% skuplji od originalnog Atari monitora, ali zato ima i nešto veću dijagonalu ekrana, oko 36 cm (14"). Neki trgovci u kompletu prodaju i

kolor-monitor SONY KX-14CP 1, koji je još nešto skuplji. Kvaliteta kolor-slike na svim kolor-monitorima je vrlo dobra, ali je opće uvjerenje da je za poslovne primjene, naročito obradu teksta ili podataka, mnogo prikladniji standardni Atari crno-bijeli monitor SM124.

Još jedan model

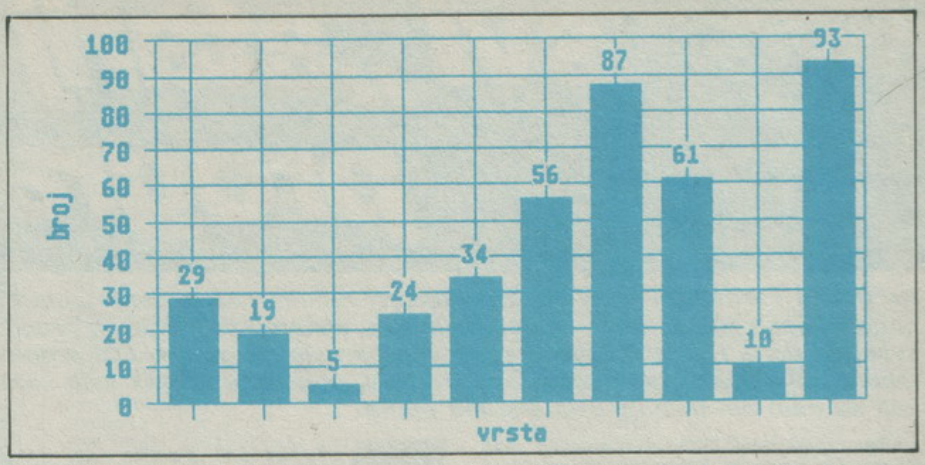
U Americi je nedavno otpočela prodaja novog modela računala iz serije ST — modela 1040ST. Taj model, također, ima ugrađen 1 MB RAM memorije, ali i već ugrađenu disk-jedinicu kapaciteta 720 K formatirano i ugrađeni ispravljač. Operacijski sistem TOS je, razume se, u ROM-u. Predviđena cijena novog modela, zajedno s crno-bijelim monitorom visoke razlučljivosti, iznosi 999$ ili oko 3000 DM. Prema toj informaciji, ne planiraju se novi modeli serije ST s još većom radnom memorijom, pošto je dostignuta gornja granica raširenja RAM memorije trenutno raspoloživim (jeftinim) memorijskim čipovima na tržištu.

Ali, već za sredinu ove godine najavljena je prodaja dodatnog modula za računala serije ST, koji će se s računalom povezivati preko DMA priključka (direct memory access). Dodatni modul će imati ugrađen procesor MC68020 i 2MB RAM-a (proširljive do 4 MB). Kao što je poznato, procesor MC68020 smatra se jednim od najjačih mikroprocesora na svijetu i ima potpunu 32-bitnu arhitekturu (i adresu sabirnicu i sabirnicu podataka). Ovaj dodatni modul će omogućiti računalima serije ST da rade s operacijskim sistemom UNIX — najjačim operacijskim sistemom na mikro i miniračunalima za više korisnika i više poslova istovremeno (multi-user, multitasking). Pošto za operacijski sistem UNIX postoji mnoštvo odličnih poslovnih i komercijalnih programa, ova najava se smatra za „objavu rata" firmama koje trenutno kontroliraju tržište malih poslovnih računala (uglavnom IBM i DEC). Očito firma Atari nastoji razbiti „famu" da je neozbiljna firma, koja se uglavnom bavi proizvodnjom neozbiljnih računala za igranje.

U izravnoj vezi s mogućnošću rada više korisnika i više poslova istovremeno je i potreba za tzv. „tvrdim diskom", tj. disk-jedinicom velikog kapaciteta i velike brzine prenosa podataka. Iako s proizvodnjom tih disk-jedinica firma Atari ima određenih problema, već je isporučeno stotinjak komada, i to uglavnom velikim programskim kućama, koje grozničavo prepravljaju svoje poslovne programe da mogu koristiti i „tvrdi disk". Ovaj model još nema službenu oznaku, a ima kapacitet od 10 MB. Njegova najavljena cijena je 600 funti ili 2000 DM.

Javno dobro

Bilo je nagađanja da programski jezik bejzik, isporučivan zajedno s računalima serije ST, a poznat po svojoj glomaznosti, nespretnosti i sporosti, nije konačna verzija. Firma Atari je ova nagađanja demantirala, izjavivši da sama ne misli raditi na daljim poboljšanjima bejzika. Međutim, već je najavljeno da će firma Metacomco ubrzo izbaciti na tržište svoj bejzik kompajler za ST. To će, vjerojatno, obradovati sve one kojima je bejzik drugi maternji jezik, pošto je firma Metacomco u posljednje vrijeme poznata po vrlo dobrim programima, a i vrlo usko surađuje s firmom Microsoft, tako da će, vjerojatno, njihov bejzik kompajler podržavati sintaksu (de fakto standarda) Microsoft. Prema najavi, Metacomco



Izjednačene snage: Iz dijagrama (dobijenih pomoću programa VIP Professional) ne može se izvući pouzdani zaključak koja je vrsta tržišta prihvatila „atari st"

bejzik kompajler će se prodavati i u ROM-u (128 K ROM-modul).

Od početka godine, uz računala serije ST više se ne isporučuje programski jezik Dr Logo, nego nova verzija SPEED-LOGO. Ova verzija je, navodno, oko tri puta brža od prijašnje. Svi vlasnici

ATARI-ST	programi
grafika	29
komunikacije	19
muzički	5
obrada podataka	24
obrada teksta	34
pomoćni	56
poslovni	87
programski jezik	61
obrazovanje	10
zabava	93
ukupno	418

računala serije ST mogu ovu verziju dobiti besplatno kod svojeg prodavača. Time se proširuje već ionako za računala takve klase i cijene prilično široka paleta tzv. public domain software, tj. programa koji su namijenjeni širokoj javnosti i besplatnom presnimavanju i širenju. Osim programskih jezika bejzik, Dr Logo i Speed-Logo, tu su još i grafički programi GEM-DRAW, Neochrome i DrDoodle, tekst-procesor GEM-WRITE i neki od programa iz razvojnog sistema (Command i Kermit). Među te besplatne programa spada i tzv. „Amiga-Ball", tj. program koji na kolor-monitoru računala serije ST imitira poznatu skakutajuću loptu, reklamni demo-programa firme Commodore za njihovo novo računalo „amiga". Zlobnici kažu da je vjernost i prirodnost boja i pokreta lopte mnogo bolja na računalima ST nego na originalu „amigi". Ali možda najznačajniji od svih public domain programa je CP/M-2.2 emulator, program koji otvara računalima serije ST vrata u svijet velikog broja CP/M programa (više o tome u prilogu „Atarijev povratak u budućnost".

Umjesto programskog paketa GEM-Draw+GEM-Write, koji Digital Research nije uspio dovršiti na vrijeme (u najavljenom obliku i kvaliteti), firma Atari sada isporučuje programski paket koji se sastoji iz programa First-Wort (tekst-procesor), DEGAS (grafički program) i DB-Master (database).

Programa kao pleve

Firma RDS-Software nudi komplet programa namijenjenih za sistemsko programiranje u programskom jeziku C. U kompletu su, osim samog C-compilera, koji je usput rečeno i združljiv (kompatabilan) s UNIX-sintaksom, a na izlazu daje već optimizirani program u strojnom kodu, još i C-Link, tzv. parser, koji provjerava sintaksu izvornih programa u jeziku C, i omogućava ispis unakrsnih referentnih listi (cross-reference list). Tu su još i tri pune diskete knjižnica (library) raznih C-funkcija, kao i C-Archive, program za pretraživanje knjižica funkcija, kao i stvaranje vlastitih knjižnica funkcija. Iako ne u izvesnoj vezi s programiranjem u jeziku C, u kompletu su još i uređivač sličica, oblika slova i aktivnost „miša" (icon, font i mouse editor). Neovisno o programskom paketu, može se dobiti i GEM-Disk-monitor, koji omogućuje „šaranje" po sadržaju disketa (pirati dobro znaju što se time sve može postići), kao i Proficopy za brzo presnimavanje sadržaja disketa, koji čak i kod 260ST presnimava čitav sadržaj diskete u jednom potezu.

Već spomenuta firma Metacomco upravo je dovršila Lattice-C-compiler, čije su knjižnice funkcija napravljene po uzoru na UNIX-ove. Već otprije je poznat njihov paskal kompajler, kao i razvojni paket za programiranje u strojnom jeziku (Assembler Development Kit). Upravo se dovršava i kompajler za programski jezik LISP (List Processing Language), koji se mnogo koristi u razvoju umjetne inteligencije (AI=Artificial Intelligence).

Njemačka firma Kingsoft (poznata po šah-programu Grandmaster za C-64) izdala je i prvu njemačku zabavnu igru za čitavu porodicu. Program postavlja razna tekstovna ili grafička pitanja i pritom svojim odgovorima zabavlja do 15 učesnika.

PROGRAMSKI JEZICI

4xFORTH	The Dragon Group
Assembler/Monitor	Computer One
BASIC Interpreter	Knupe GmbH
BASIC Interpreter	Philon
C Compiler	Computer One
C Compiler	GST Systems
C Compiler	Hisoft
C Compiler	Mark Williams Company
C Compiler	Philon
COBOL	Philon
COBOL Compiler	Knupe GmbH
Dev-Pac	Hitech
FAST-BASIC Compiler	Philon
Fast-C Compiler	Knupe GmbH
Fast-PASCAL Compiler	Knupe GmbH
FORTH	Computer One
FORTH Accelerator	The Dragon Group
FORTH ST	DATA Becker
FORTRAN Compiler	Knupe GmbH
FORTRAN-77	Philon
FORTRAN-77	Prospero Software
GFA-BASIC Interpreter	Integral Hydraulik
Haba Hippo-C	HABA Systems
Holmes & Duckworth FORTH	Mirage Concepts
ISO PASCAL	Prospero Software
K-Seka	KUMA Computers
Lattice-C	Antic Software
Lattice-C	Metacomco
LISP	Metacomco
M-BASIC Compiler	Knupe GmbH
Macro Assembler	Metacomco
Macro Assembler/Editor	Antic Software
Macro Assembler/Editor	GST Systems
Megamax-C Compiler	Megamax
Meta Pascal	Antic Software
Meta Pascal	Metacomco
Micro COBOL Compiler	BDS Software
MODULA-2	TDI Software
p-System	TDI Software
PASCAL	Computer One
PASCAL Compiler	Hisoft
PASCAL Compiler	Philon
Personal PASCAL	Optimized Systems Software
Personal PROLOG	Optimized Systems Software
PolyFORTH	Ind. Control Services
Profi-Assembler	RDS Software
Profi-Monitor	RDS Software
Profimat ST	DATA Becker
RPG	Philon
ST BASIC	ATARI
ST BASIC Compiler	ATARI
ST Development System	ATARI
ST LOGO	ATARI
ST PASCAL	CCD Software
TBC Compiler	RDS Software
TBC Lint	RDS Software
Turbo C	Borland International
Turbo Pascal	Borland International
UCSD BASIC	TDI Software

POMOCNI (UTILITIES)

ARST	RDS Software
BKS-Stop	BKS Software
BOS/5 Operating System	BOS Software
Busines Calculator	RDS Software
C-Diskeditor/GEM	C-Soft GmbH
SoftC-Libraries	RDS Software
Calendar	ATARI
Clock	ATARI
COCO	Michael Gehret
Control Panel	ATARI
CP/M-2.2 Emulator	Soft-Design
Directory print	RDS Software
Disc Library	Extended Software Company
Disc-Monitor	RDS Software
Disk Index	RDS Software
Disk-Monitor	G-Data
Drucker-Disk	RDS Software
Dump	ATARI
Editor-Disk	RDS Software
Fast-TOS	RDS Software
File-Copy	RDS Software
Find	ATARI
G-Format	G-Data
G-Multi	G-Data
G-Protect	G-Data
Hippo Dust	Hippopotamus Software
Hippo RAM-Disk	Hippopotamus Software
Hitchackers Guide to BIOS	ATARI
Holmes & Duckworth Tools	Mirage Concepts
IBM PC Cross Development Sys.	Metacomco
Joshua	Oliver Joppich
K-RAM	KUMA Computers
Kermit	ATARI
M-Copy	MichTron
M-Disk	MichTron
Mi-Dupe	MichTron
Micro-C Shell	Beckmeyer Development Tools
Mutil	MichTron
Object Size	ATARI
Personal Diskit	Optimized Systems Software
Profi-Copy	RDS Software
Programmers Calculator	RDS Software
Rhythm	Softechnics
RAM-Disk	G-Data
Resource Construction Set	ATARI
Softspool	MichTron
Source Labels (NM68)	ATARI
Strichcode	CDS-EDV-Service
Symbolic Instruction Debugger	ATARI
Trak-Grabber	Haase Computersysteme
Tramiel Operating System	ATARI
Transfer IBM/ST	MichTron
VT-52 Emulator	ATARI
Watch	ATARI
YU-TOS	Xenon
Yutility	Xenon

POSLOVNI

Z-key Accounting System	ATARI
Z-Key Accounting System	Supplemental
Administrator	Fydler
Appointmaster	Fydler
Bilanz-Analyze	Haus der Buchhaltung
Bosplanner	BOS Software
Buchhaltung	Haus der Buchhaltung
Business Administrator	Wienand KG
BS-Apris	Bavaria-Soft
BS-Assekura	Bavaria-Soft
BS-Manager	Bavaria-Soft
C-Auftrag	C-Soft GmbH
Calender	MichTron
Cash Manager	Toungate Software
Cash Trader	Quest International
Desk Diary	Paradox Software
Easy Record	MichTron
Electro Calender	Soft Logic Corp.
Faktura	Haus der Buchhaltung
Financial Cookbook	Electronic Arts
Fixed Assets	BOS Software
FIBU	Haus der Buchhaltung
Haba Calc	HABA Systems
Haba Checkminder	HABA Systems
Haba Dem	HABA Systems
Haba Dialer	HABA Systems
Haba Wills	HABA Systems
Haba Window Calculator	HABA Systems
Haba Window Dialer	HABA Systems
Hippo Concept	Hippopotamus Software
Home Account	Micro Ed.
HomePak	Batteries Inc.
Integrated-7	Mosaic
Inventory Controls	BOS Software
Invoicing	BOS Software
Isgur Portofolio System	Batteries Inc.
K-Calc	Antic Software
K-Spread	KUMA Computers
Kartei	Haus der Buchhaltung
Kunden	Bavaria-Soft
Lager	Bavaria-Soft
Lohn	Haus der Buchhaltung
Mi-Calendar	MichTron
Merkant	Vogler Software
Negotiator	BOS Software
Nominal Ledger	BOS Software
Nominal Ledgers	Chipsoft
Nominal/General Ledger	Quest International
Payroll	BOS Software
Personal Money Manager	MichTron
Project Leader	Datacode Systems
Project Manager	Planner
Purchase	Chipsoft
Purchase Ledger	BOS Software
Purchase/Creditor Ledger	Quest International
Rechnungen	Bavaria-Soft
Sales	Chipsoft
Sales/Debtors Ledger	Quest International
Sales Invoicing	Quest International
Sales Order Processing	Cheltek Systems
Schecks	Haus der Buchhaltung
Scheduler	Rising Star
Side-Click	RDS Software
SM Kunden	SM Software
SM Lager	SM Software
SM Rechnung	SM Software
SM Spreadsheet	SM Software
UCSD PASCAL	TDI Software
UCSD PASCAL-77	TDI Software

GRAFIKA

3-D Interiors	ATARI
Business Graphics	Rising Star
CAD-System	E&C Erlangen
CAD/CAM	Rising Star
DEGAS	Batteries Inc.
Dr. Doodle	ATARI
DraWrite	MiGraph Inc.
Easy Draw	MiGraph Inc.
FiG-GAL	Progressive Computer Appl.
Fleet Street Editor	Mirrorsoft
GEM-DRAW	ATARI
GEM-Paint	ATARI
Graphic Artist	Progressive Computer Appl.
Hippo Art I	Hippopotamus Software
Hippo Pixel	Hippopotamus Software
Mastergraph	Fydler
Megafont	Xlent Software
NEO-Chrome	ATARI
NEO-Images	ATARI
NEO-Slide	ATARI
Object Editor	Rugby Circle
Platine ST	DATA Becker
Print Shop	Broderbund
Print Shop Graphics Library	Broderbund
RubberStamp	Xlent Software
Shape & Icon Editor	Monarch Development Co.
ST Coloring Book	The Dragon Group
Val Draw	Rising Star
Val Paint	Rising Star

KOMUNIKACIJE

ASCII/Prestel Comms	Miracle Technology
ASCII Comms	Softworks
BBS	MichTron
Chat!	SST Systems
Fastcomm	AM Technology
Forem	Comnet
Haba Comm	HABA Systems
K-Comm	KUMA Computers
Local Area Networks	Software Punch
Mi-Term	MichTron
PC/Intercomm	Mark of the Unicorn
S-Term	Harald Zoschke
ST Talk	Quantum Microsistems
ST Term	Comnet
SWR	SST Systems
Terminal Emulation Software	Compel PLC
Val Mail	Rising Star
VICOM	ATARI
VT-100 Emulator	RDS Software

MUZICKI

Akkorde im Akkord	Steinberg Waldorf
Good Vibrations	Steiberg Waldorf
Music	ATARI
Music System	Island Logic
Music Writer	Crossbow Music

OBRADA PODATAKA

Adress-Perfect	RDS Software
Autoclerk	BOS Software
Autoindex	BOS Software
C-Adress/GEM	C-Soft GmbH
C-Video	C-Soft
dB One	Oxxi Inc.
Database	Talent Computer Systems
Datamat ST	DATA Becker
Datastate	Fydler
DB Master	Stoneware
Dral-Adressverwaltung	Luda Software
G-Datei	G-Data
H&D Base	Softline
Haba Dex Phone Book	HABA Systems
Hippo Simple	Hippopotamus Software
Holmes & Duckworth Base	Mirage Concepts
K-Data	KUMA Computers
Laserbase	Lasersoft
Manager	ATARI
Optimum	Datenbench
Superbase	Precision Software
Video Access Z	Guenther Software
VIP Consultant	VIP Technologies
Zoomracks	Quickview Systems

OBRADA TEKSTA

1ST_WORD	GST Systems
ATARI Word	Software Punch
BJ-Writer	ATARI
Boswriter	BOS Software
C-Text	C-Soft GmbH
Express Leter Processor	Mirage Concepts
Final Word	Mark of the Unicorn
GEM-Write	ATARI
Haba Business Letters	HABA Systems
Haba Mailmerge	HABA Systems
Haba Writer	HABA Systems
Hippo Spell	Hippopotamus Software
Hippo Word	Hippopotamus Software
K-Word	KUMA Computers
Lets Write	Mark Williams Company
Mailmate	Fydler
Metacomco Editor	Metacomco
Mighty Mail	MichTron
Mince Editor	Mark of the Unicorn
PaperClip	Batteries Inc.
Protext	Markt&Technik
Regent Word	Regent Software
Spelling Checker	Regent Software
Spellmate	Fydler
SM Manager	SM Software
SM Text	SM Software
ST Writer	ATARI
Superscript	Precision Software
Text-Designer ST	DATA Becker
Textomat ST	DATA Becker
VIP Freelance	VIP Technologies
Word for Word	Bay View Software
Wordmate	Fydler
WordWhiz	Oxxi Inc.
Spreadsheet	Rising Star
Stock Control System	Quest International
Taurus Accounts	Execon Software
Taurus Invoicing	Execon Software
Taurus Printers Package	Execon Software
Taurus Stock Control	Execon Software
Timelink	Softechnics
Twin	Mosaic
Val Docs 2	Rising Star
VIP Analysis	VIP Technologies
VIP Professional	VIP Technologies
Wasp Accounting Time Records	Whitendale
Wasp Accounts Diary Systems	Whitendale
Wasp Incomplete Records	Whitendale
Wasp Insurance	Whitendale
Wasp Legal Accounting	Whitendale
Wasp Legal Diary System	Whitendale
Wasp Membership Accounting	Whitendale
Wasp Printers Estimating	Whitendale
Wasp Time Recording	Whitendale

OBRAZOVNI

ATARI Planetarium	ATARI
Comword	Word of God Comm.
Hippo Almanac	Hippopotamus Software
Home Planetarium	Deltron
Homework Helper/Math	Spinnaker
Homework Helper/Word	Spinnaker
Maps and Legends	Antic Software
Master Type	Scarborough
Touch Typing Tutorial	Academy Software
Typesetter ST	Xlent Software

ZABAVA

3-D Chess	Intelligent
A Mind Forever Voyaging	Infocom
Amazon	Spinnaker
Asteroids	Crossbeam
Astrologie	SM Software
Baron	Bluechip Software
Battlezone	Crossbeam
Biorhytmus	Pahlen&Kraus
Bongo	Sven Krueppel
Borrowed Time	Activision Inc.
Brattacas	Psygnosis
Breakout	ATARI
C-Lotto/GEM	C-Soft GmbH
Caverns	Paradox Software
Colorspace	Antic Software
Colorspace	Llamasoft
Crimson Crown	Penguin Software
Cut Throats	Infocom
Deadline	Infocom
Delta Patrol	Antic Software
Dragon World	Spinnaker
Eidolon	Epyx
Electro Solitaire	Soft Logic Corp.
Enchanter	Infocom
Fahrenheit 451	Spinnaker
Flight Simulator	MichTron
Flight Simulator 2	Sublogic
Flipside	MichTron
Forbidden Quest	Priority Software
Gold Runner	MichTron
Goonies	Datasoft
Hacker	Activision Inc.
Halley Project	Mindscape
Hex	Mark of the Unicorn
Hippo Backgammon	Hippopotamus Software
Hippo Jokes & Quotes	Hippopotamus Software
Hitchickers Guide to Galaxy	Infocom
Hyperdrive	English Software
Hyphus	Penguin Software
Impossible Mission	Epyx
Infidel	Infocom
Jet	Sublogic
Joust	ATARI
Kings Quest 2	Sierra On-Line
Lands of Havoc	MichTron
Lost Kingdom of Zkul	Talent Computer Systems
Major Motion	MichTron
Marble Madness	Electronic Arts
Megaroids	Magamax
Millionaire	Bluechip Software
Millipede	Andromeda
Mindshadow	Activision Inc.
Missile Command	Andromeda
Mission Mouse	Antic Software
Mom and Me	Antic Software
Mudpids	MichTron
Murray and Me	Antic Software
My Chess 2	Datamost
Nine Princes of Amber	Spinnaker
Penstate	Penguin Software
Perry Mason: Mandarin Murder	Spinnaker
Planetfall	Infocom
Polyscan	Island Logic
Quiwi	Kingsoft
Quatrow	RDS Software
Seastalker	Infocom
Silent Service	Microprose
Sorcerer	Infocom
Spellbreaker	Infocom
Star Raiders	ATARI
Star Raiders	Crossbeam
Star Raiders II	Andromeda
Starcross	Infocom
Starglider	Firebird
Summer Games	Epyx
Summer Games 2	Epyx
Sundog: The Frozen Legacy	FTL Games/Software Heaven
Suspended	Infocom
Time Bandit	MichTron
Transylvania	Penguin Software
Treasure Island	Windham Classics
Tycoon	Bluechip Software
Ultima 2	Sierra On-Line
Universe 2	Omnitrend
West	Talent Computer Systems
Winnie the Pooh	Sierra On-Line
Winter Games	Epyx
Wishbringer	Infocom
Witness	Infocom
Wizard of Oz	Windham Classics
Zork	Infocom
Zorro	Datasoft

Računari i obrazovanje

Programirana nastava

škola sa osam programa

Računar u obrazovanju može da se koristi na više načina. Zahvaljujući jeftinim mikroračunarima i sve boljem obrazovnom softveru, oko 75% državnih škola u SAD koristi računare za podršku nastave, a samo desetak godina ranije ovo moćno nastavno sredstvo koristilo je svega 10% škola. Pri tome je u prvo vreme računar korišćen prevashodno u tzv. programiranoj nastavi, dok se danas sve više koristi za modeliranje i stimulaciju.

Nova vizija ili nova zabluda

Od svoje pojave u SAD pedesetih godina do danas programirana nastava je doživela značajan uspeh i podstakla razvoj i proizvodnju raznih tehničkih sredstava, programiranih udžbenika i softvera. Programirana nastava je vrsta nastave u kojoj se sadržaji na poseban način logički struktuiraju i daju učenicima u manjim ranije pripremljenim delovima. Ove celine oni uče samostalno i postupno, idući korak po korak sopstvenim ritmom, uz stalno proveravanje stepena usvojenosti izloženih sadržaja. Računari, kako veliki „tajmšering" sistemi tako i mikroračunari, predstavljaju idealno sredstvo za realizaciju programirane nastave.

Postoji više vrsta programirane nastave, a programi koji na ovaj način podržavaju učenje sastoje se obično iz četiri dela. Na početku se nalazi *tekstualni materijal* koji objašnjava o čemu se radi. On mora biti jednostavno i jasno napisan. Zatim se zadaju *tekstualna pitanja* koja zahtevaju odgovor učenika. Računar analizira odgovor. Ako je odgovor tačan, dobija se potvrda, a ako nije ukazuje na probleme na koje treba obratiti pažnju. Uvek su predviđeni i *alternativni tekstovi* na istu temu koje obično pišu razni profesori, tako da će verovatno svaki učenik naći pristup problemu koji njemu najviše odgovara. Na kraju se vrši *analiza uspeha* koja pokazuje koliko je tema savladana, daje procenat tačnih odgovora i sve druge pokazatelje koji su interesantni za nastavnika.

Tehnološko obrazovanje odraslih

Naš saradnik dr Radomir Mihajlović, vanredni profesor na Njujorškom institutu za tehnologiju, radio je u timu koji je za jednu firmu iz Kalifornije projektovao softverski paket namenjen obučavanju za korišćenje novih tehnologija. Projekat je kasnije otkupio IBM i danas je u širokoj upotrebi u svim njegovim fabrikama. Zamolili smo profesora Mihajlovića da nam kaže nešto više o njemu.

— Ovaj paket programa trebalo je da omogući da se i ljudi koji nemaju talenta za tehniku obuče za korišćenje i proizvodnju savremene tehnologije. Problem na Zapadu je taj da se tehnologije usavršavaju sve više

Kompjuterom za kompjutere: Prof. dr Radomir Mihajlović

i više, a da je sve manje talentovanih ljudi koji su u stanju da ih prate i koriste. Prema tome, u korišćenje nove tehnologije moraju se uključiti i oni sa nešto slabijim intelektualnim kvalitetima. Da bi se jedan čovek netalentovan za matematiku, elektroniku i programiranje obučio za produktivno korišćenje savremene tehnologije, potrebno mu je posvetiti proporcionalno znatno više pažnje nego nadarenom. On ne može sam da uči — neophodno mu je više tutorijalne interakcije. Međutim, ograničen je broj kvalitetnih profesora i oni fizički ne mogu da postignu da stotine sati posvete svakom svom studentu, kao što je Aristotel činio sa sinovima bogatih Atinjana. Bogati Grci slali su svoje sinove Aristotelu, on bi šetao s njima kroz park i postavljao bi im pitanja, a oni odgovarali. Veštim postavljanjem pitanja navodio bi ih na zaključke tako da su imali utisak da su sami došli do pravih odgovora. Ovaj projekat predstavlja pokušaj da se na računaru simulira Aristotelov metod.

Moj zadatak bio je da pripremim programe za učenje o mikroprocesorima. Od hardvera smo imali na raspolaganju računar spregnut sa video diskom čije je važno svojstvo da poseduje vizuelnu informaciju izuzetno velikog kapaciteta. Na ovaj način daje se mnogo efikasnija informacija, jer je video informacija mnogo efektnija od govorne. Po mojim proračunima na osnovu nekih formula iz teorije informacija, video informacija je čak 200.000 puta efikasnija nego govorna. Umesto da se zadaje lekcija pa domaći zadatak kao u tradicionalnim metodama obučavanja, ovde je predviđen niz sekvenci (sekvenca u programiranoj nastavi predstavlja deo nastavne teme koji je sadržajno i logički povezan, prim, autora) kojim se prolazi sopstvenim ritmom u zavisnosti od brzine savladivanja gradiva.

Sa video diskom sa direktnim pristupom, spregnutim sa računarom, programirana nastava realizovala se na sledeći način. Prvo bi se sa video diska preuzela jedna sekvenca po čijem se odigravanju ekran zamrzavao i postavljalo se pitanje kojim se proveravalo da li je usvojen sadržaj prezentiran sekvencom. Ako je odgovor bio korektan, išlo bi se dalje, a ako nije odigrala bi se nova sekvenca koja objašnjava u čemu je bila greška i ponovo bi se očekivao odgovor. Koristili smo dva zvučna i jedan video kanal. Na ovim kanalima su na različit način prezentirani isti sadržaji, tako da se za one kojima jedna priča nije dovoljna mogla odigrati i druga . . . Tu sam morao da predvidim sve moguće glupe odgovore i sekvencu posle koje bi trebalo da učenik bude u stanju da shvati u čemu je pogrešio i da dâ ispravan odgovor, što mi je, moram priznati, bilo teže od svega ostalog.

Uz malu pomoć nastavnika

Čini mi se da bez obzira na savršenu tehnologiju i neslućene mogućnosti video diska kao nosioca informacija ipak programirano učenje neće dati prave rezultate bez prisustva nastavnika. Bez obzira na maštovitost tvoraca softvera, ne verujem da se mogu predvideti svi mogući odgovori. Mislite li vi da ima smisla učitelja potpuno zameniti računarima, makar oni simulirali i Aristotela?

— Ne, nikako. Može se mnogo predvideti, ali ne sve, razume se na nivou snage računara sa kojima smo mi radili. Međutim, ja sam zastupao stanovište da svi ti terminali sa kojih se učilo moraju biti propraćeni jednim moderatorom — tako sam nazvao učitelja koji je trebalo da pomogne u situacijama koje nisu predviđene programom i pruži ljudski kontakt koji za učenika nije ništa manje važan od stručne pomoći u savladivanju gradiva. Međutim, što se tiče efikasnosti u manipulisanju informacijama, čovek je tu ipak znatno ispod mogućnosti računara. Zato smatram da treba vršiti simbiozu mogućnosti računara i nastavnika i tako na najbolji i najracionalniji način realizovati nastavu.

Iskustvo profesora Mihajlovića prilog je tezi koja se sve češće čuje da kompjuterski

Dva su viđenja korišćenja računara u obrazovanju — računar kao predmet izučavanja i računar kao nastavno sredstvo. Računari i njihova primena kao predmet proučavanja u nastavi često su bili razmatrani na našim stranicama. O drugoj vrsti njihove primene koja bi trebalo da bude mnogo češća — korišćenje računara kao nastavnog sredstva — još uvek retko ili skoro nikako ne govori. Kako je u našoj zemlji uvođenje računara u učionice još uvek u početnoj fazi, nemamo dovoljno praktičnih iskustava koja bi mogla da se uopšte i ponude univerzalni recept kako napraviti valjan edukativni program i kako ga koristiti u nastavi, ali među našim saradnicima ima i onih koji se profesionalno bave edukativnim softverom, pa bi njihova iskustva mogla biti korisna onima koji žele da na pravi način iskoriste svoje znanje programiranja, kao i nastavnicima i učenicima od kojih se očekuje da koriste gotove edukativne programe.

Matematičko modeliranje života: Mr Veljko Spasić

podržana programirana nastava uglavnom pomaže poboljšanju uspeha lošijih učenika, omogućavajući im da pristignu svoje vršnjake, ali da ima vrlo malo uticaja na razvijanje kreativnih sposobnosti. Posebno ako se zasniva na bihejvioristićkoj teoriji koja u središte programirane nastave stavlja odgovor (reakciju) učenika na moguće alternative rešenja i time sputava stvaralačko mišljenje. Uz to se ne sme izgubiti iz vida da se njenom primenom smanjuje mogućnost za razvijanje socijalnih odnosa i kolektivnih veza što, ako škole ne posmatramo samo kao obrazovne već i kao vaspitne ustanove, predstavlja značajnu slabost svake programirane nastave, pa i one podržane računarima.

Međutim, razumno i odmereno korišćenje računara kao sredstva za realizaciju programirane nastave omogućava znatno efektivniji rad. Na ovaj način znatno se poboljšava kvalitet organizacije nastavnog procesa. Što je još bitnije, omogućava se individualno usvajanje znanja sopstvenim ritmom, čime se učenici stimulišu za stalno i aktivno učenje. Mogućnosti programirane nastave nisu male. One su korisne naročito u matematici, prirodnim i tehničkim naukama. Primena programirane nastave daje dobre rezultate i u učenju gramatike, sintakse i elementarne pismenosti, kao i u mnogim društveno-naučnim predmetima. Kad god su u pitanju saznajne aktivnosti u procesu učenja, programirana nastava može znatno doprineti povećanju efikasnoti, pogotovu ako se u proces učenja uvode problemi koji zahtevaju stvaralačko rešavanje.

Učenje otkrivanjem

Modeliranje i simulacija na računaru u nastavi sve više dobijaju na značaju upravo stoga što kod učenika ne podržavaju mehaničko učenje, već podstiču aktivan misaoni rad. Naš saradnik, mr Veljko Spasić, matematičar koji radi kao istraživač u Centru za multidisciplinarne studije Beogradskog univerziteta, duže vreme je radio na Londonskom univerzitetu kao istraživač na naučnoistraživačkim projektima matematičkog modeliranja i simulacije. Paralelno sa tim radio je u centralnom timu CAL (Computer Assisted Learning) u okviru koga je izrdio više simulacionih programa od kojih su „Šećer u krvi čoveka" i „Simulacija populacione dinamike" izdati u Velikoj Britaniji (izdavač: Longman Publisher, London) i prevedeni u još desetak zemalja. Zamolili smo ga da nam ukratko objasni šta se podrazumeva pod modeliranjem i kako se ono može koristiti u računarski podržanoj nastavi.

— Modeliranje je, u širem smislu reči, postupak zamene jednog objekta drugim, koji mu je, u manjoj ili većoj meri, sličan. Matematičko modeliranje je moćno sredstvo u raznim oblastima istraživanja, a koristi se i kao dopuna drugih metoda.

U primeni računara u obrazovanju matematičko modeliranje je najpogodniji način „pakovanja" sistema koji se izučavaju u formi koja omogućava upotrebu računara. Jasno je da su neke oblasti više, a neke manje pogodne za ovakav pristup.

Matematički model omogućava pisanje simulacionog programa kojim se prikazuju ponašanje, odnosno osobine modela. Model se posmatra i analizira pomoću računara. Ovakvi simulacioni programi napisani u skladu sa dodatnim zahtevima koje nameće njihovo korišćenje u nastavi, obavezno su praćeni odgovarajućom dokumentacijom. U slučajevima kada je zadatak proučiti složenije, naročito dinamičke sisteme koji imaju komplikovanu strukturu, međuzavisnost sastavnih delova i složeno ponašanje, kao što je to veoma često slučaj u fizici, biologiji, elektronici itd., pristup preko simulacije predstavlja najpogodniji mogući vid prikaza ovih sadržaja. Simulacionim programima mogu se zamenjivati realni eksperimenti i laboratorijske vežbe koje iz određenih razloga ne mogu biti cena, dužina trajanja realnih eksperimenata, nepostojanje odgovarajuće opreme i slično.

Možete li nam opisati neki konkretan simulacioni program?

— Jedan veoma interesantan program engleskog Otvorenog univerziteta (Open University) simulira redove sa čekanjem. Osnovna ideja programa je da omogući lako i fleksibilno izvođenje eksperimenata sa funkcionisanjem uslužnog centra koju ste najpre sami definisali. Na raspolaganju stoje uslužna mesta čiji broj i potrebno prosečno vreme usluživanja zadajete sami. Takođe definišete vreme i trajanje pauze u radu uslužnih mesta, kao i maksimalno dozvoljen broj istovremeno zatvorenih. Korisnici usluga — kupci stižu na slučajan način, a vi određujete njihovu distribuciju. Kada je red prevelik, kupci se vraćaju i ostaju neusluženi. I ovu dužinu reda koja prevazilazi strpljenje kupaca određujete sami. Program, zatim, vrši simulaciju i na ekranu se vide kupci koji dolaze, bivaju usluženi ili odlaze radi gužve. Vidi se rad uslužnih mesta i njihovo privremeno zatvaranje radi pauze. Time su rezultati vašeg eksperimenta jasno prikazani. Na kraju se dobija izveštaj koji sadrži podatke o broju usluženih kupaca, broju vraćenih, troškovima rada, profitu, kao i o maksimalnom mogućem profitu. Program izvanredno prikazuje redove sa čekanjem koji se, inače, analiziraju složenim matematičkim aparatom i u vidu „eksperimentalne matematike" razvija se osećaj za ponašanje sistema masovnog opsluživanja i omogućava slobodno eksperimentisanje.

Kakvu formu treba da imaju edukativni programi? Da li se njihovoj zaštiti posvećuje pažnja kao zaštiti igara i drugih komercijalnih programa?

— Ovi programi nikako ne treba da budu zaštićeni. Jedna od važnih namena je da njihovi korisnici, učenici i nastavnici, mogu da uđu u njihovu strukturu i da ih menjaju. Oni nisu načinjeni da budu zamena već pomoć nastavnicima. Stoga su i propraćeni bogatom i jasnom dokumentacijom. Sem interne dokumentacije koja se bavi specifičnostima samog programa, edukativni programi treba da budu propraćeni posebnom dokumentacijom koja se obraća učenicima.

Kako budućnost školstva kao i budućnost mikroračunara u mnogome zavise upravo od edukativnih programa, uvereni smo da ima smisla posvetiti ovoj temi punu pažnju. Pred našim programerima i nastavnicima je ogroman i odgovoran posao kreiranja programske podrške nastave većine predmeta. Stoga pozivamo na saradnju sve one kojima je problematika korišćenja računara u nastavi bliska, izazovna ili bolna, jer ćemo zajedno svakako brže i bolje uvesti računare u učionice. To je, možda, pravi put ka — efikasnijoj školi.

Nevenka Spalević

Programski jezici

C (2)

nizak nivo a visoke grane

U slučaju pointera b=a+1, gde **a** i **b** pokazuju na podatak tipa long integer (4 bajta širok), adresa **b** je za 4 veća od adrese **a**. Aritmetika sa pointer promenljivima se obavlja u odnosu na specificirani tip (veličinu) adresiranih podataka. Korišćenje pointer-podataka podseća bejzik programere na stavove PEEK i POKE (slika 7).

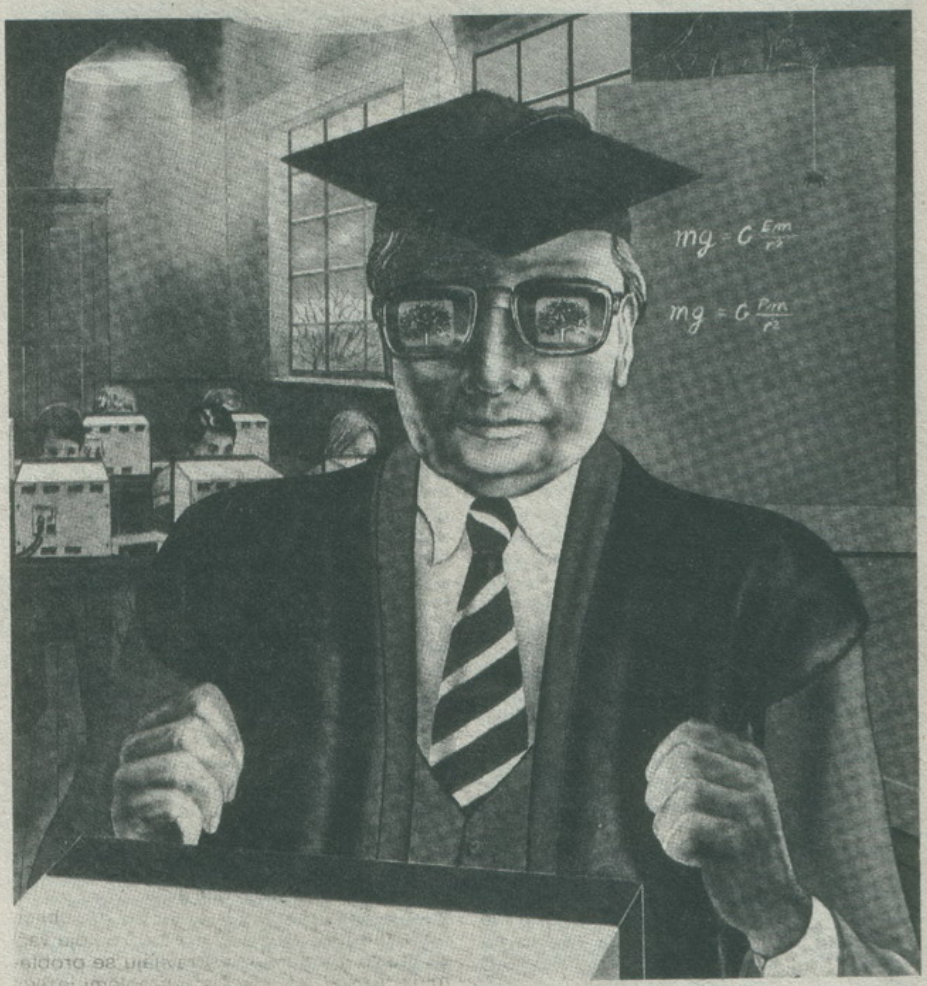
Kad je reč o dodeli memorijskih lokacija, C vrlo skromno nudi samo opciju statičke dodele. Tačne adrese moraju biti unapred specificirane i dodeljene u toku vremena prevođenja programa, a ne fleksibilno, zavisno od potrebe, u toku izvođenja programa (ALGOL-68).

Indirektno, veštim programiranjem moguće je manipulisati i pojedinačnim bitovima izolovanih memorijskih reči. U nameri da se približi mašinskom kodu što je moguće niže, Denis Riči je predložio više neuobičajenih operatora. Ovi operatori čine C programe pomalo teškim za čitanje, jer uz poznate +,=, * i / operatore, C programer ima na raspolaganju „asemblične" operatore kao što su: ++ (inkremet za 1), −− (dekrement za 1), << (pomak ulevo), >> (pomak udesno), & (I, engl. AND), '|' (ILI, engl. OR), i ~ (komplement 1, engl. NOT). Da bi bio u stanju da kontroliše ulazno-izlazne jedinice, ekran ili, pak, konkretnu memorijsku lokaciju, za sistemskog programera je vrlo važna mogućnost direktnog manipulisanja bitovima. C to u potpunosti podržava i zbog toga ga mnogi u šali nazivaju „asembli jezikom visokog nivoa". Uz povećane mogućnosti sistemskog programiranja, ovde se otkriva i tajna brzine izvršenja C programa. Tajna je delimično u eksplicitnosti i prirodnosti „dubinskih" operatora jezika C u odnosu na mašinu.

Jednostavni test program (eng. bechmark) za sabiranje svih celih brojeva po redu od 0 do 15.000 je pedesetak puta brzi od istog programa napisanog u bejziku. Sa izuzetkom specijalnih slučajeva, u većini situacija sa kvalitetnim C prevodiocem, razlika u brzini izvršenja istog programa u C i u asembli jeziku je gotovo neprimetna. Za računare sa različitim centralnim procesorskim jedinicama potrebno je naučiti različite asembli jezike. Pri prelasku na novi računar napori i cena ponovnog obučavanja mogu da dostignu neugodan nivo. Ljubitelji programskog jezika C tvrde da je C potrebno naučiti samo jednom i da je to beznačajna avantura u poređenju sa odisejom kroz uzburkane vode mnogobrojnih asembli jezika. Ako ni zbog čega drugog, u svakom slučaju zbog ovoga vredi naučiti C.

Protiv komplikovanja komplikovanog

Svi „paskalični", tj. „algolski" jezici (engl. Pascal-like/Algol-like), pa i jezik C, svrstavaju se u grupu jezika visokog nivoa. Programski jezik C je malo jezik na visokom nivou. To je ono što je lepo kod bilo kog jezika koji je potrebno naučiti brzo i bez većih problema.



C je, slično algolu, tipiziran i modularan jezik. C program je moguće razbiti na unutrašnje i spoljašnje blokove ili module (sl. 8). Svaki spoljašnji blok ima specijalnu namenu ili funkciju, pa otuda naziv spoljašnja funkcija. Funkcije prihvataju ulazne podatke (parametre) „po vrednosti". Preneseni parametri se stavljaju na raspolaganje potprogramu, uz očuvanje „rezervne kopije" u glavnom programu.

C ne dozvoljava ugnezdivost programa. Funkcija funkcije (deklarisanje nove funkcije u funkciji) ili dalje komplikovanje komplikovanog nije dozvoljeno. Ukratko, modularnost C programa uveliko olakšava lociranje moguće greške (engl. dubugging) i dalju dogradnju gotovih programa. Značaj modularnosti je eksponencijalno rastuća funkcija veličine programa. Ko to ne veruje, uveriće se brzo ako se prihvati iole većeg programerskog posla.

Uprkos činjenici da C ima manje ključnih ili rezervisanih reči od paskala, može se reći da poseduje daleko veće mogućnosti. C pokazuje svoju nadmoćnost kroz pažljivo uključivanje pogodnih kontrolnih struktura i tipova podataka, dozvoljavajući svojim korisnicima da praktično budu oslobođeni svih ograničenja.

C nudi samo neposredne i jednostavne instrukcije za kontrolu redosleda izvršenja: testove tipa ako-onda-inače, petlje za ponavljanje izvršenja nekih instrukcija, grupisanje instrukcija u blokove i potprograme. Za razliku od jezika ada i modula, kod većine verzija C jezika ne postoje kompleksne kontrolne instrukcije za paralelno izvođenje operacija kao što su to instrukcije za multiprogramiranje i sinhronizaciju ili korutine. (O novim trendovima u sistemskom programiranju i softverskom inženjerstvu uopšte biće reči u jednom od narednih brojeva.). C ne nudi operacije za direktnu odbranu kompozitivnih „objekata", kompleksnih struktura podataka, tipa lanaca

Pisanje softvera u tehnološkoj situaciji sa svakodnevnim iznenađenjima i novinama, kada se mašine smenjuju preko noći, predstavlja izuzetno složen poduhvat. Pažljivim izborom programskog jezika i ostalih razvojnih programskih pomagala na samom početku projekta, posao se može značajno pojednostaviti. Izbor programskog jezika, međutim, između nije nimalo jednostavan zadatak. Dopisnik „Računara" iz Njujorka dr Radomir A. Mihajlović piše o kontraverznom jeziku C, koji u višemilionskoj armiji američkih programera poslednjih godina predstavlja pravi hit. Ne bez razloga. Na svetu trenutno nema efikasnijeg, jednostavnijeg i prenosivijeg jezika od jezika C. Da stvar bude lepša, C je dostupan i domaćim programerima, čak i onima koji pišu na „spektrumu", „amstradu" ili „komodoru".

karaktera, skupova, lista ili matričnih višedimenzionalnih tabela označenih jednim zajedničkim simbolom (PL/I, paskal, itd.).

Operacije na složenim strukturama se mogu „ugraditi" u C prevodilac jednostavnim dodavanjem spoljašnjih rutina postojećoj C biblioteci. Kreiranjem sopstvenih funkcionalnih modula programer može udesiti da njegov C bude u stanju da čini čuda. Moguće je, praktično, definisati novi, privatni, jezik bez ičeg ekstra što programeru ne treba. Ovo znači da C jezici različitih korisnika mogu da poseduju različite mogućnosti, potpuno prilagođene pojedinačnim potrebama.

U svojoj jednostavnosti, primera radi, C nema ugrađene ulazno/izlazne (engl. Input-Output ili I/O) programske konstrukcije ili komplikovane programske kontrolere iznenadnih zahteva (engl. interrupt handlers). Za komplikovanije usluge C se obraća spoljašnjoj biblioteci rutina, koje mogu pripadati i prevodiocu jezika i operativnom sistemu. Ulaz i izlaz podataka u jeziku C kontrolišu lako uočljivi moduli spoljašnjih funkcija: **scanf**() ili **printf**().

Upotrebni programi (engl. utilities) i programski pretprocesor omogućavaju programeru da izoluje detalje određene datim računarima od glavnog koda C kompajlera. U uvodu svakog C programa dotični računar se predstavlja nizom specifičnih rutina, sabranim u standardnu ulazno-izlaznu datoteku u zaglavlju, **stdio.h** (u našoj verziji C jezika stdioz). Specifičnosti korišćenog sistema su problem programskog pretprocesora i datoteke stdio.h. Ulaz u C pretprocesor je izvorni C program, a izlaz „ispeglana" verzija izvornog programa (slika 8). Ovakav prikaz drastično olakšava ponovno definisanje Ca pri prelasku na novi sistem. Vrlo je jasno šta i gde treba menjati u sistemu programa i rutina C prevodioca da bi se C prevodilac za stari računar „prilagodio" na novi. Zaključak je da su minimalnost jezika C i njegova fizička nezavisnost, tj. labava veza sa spoljašnjom bibliotekom, prava tajna njegove jednostavne prenosivosti između računara najrazličitijih tipova.

Zahvaljujući svojim „malim dimenzijama", C je lako prenosiv, portabilan jezik. C program napisan za jedan računar može biti izvršen sa malim ili pak bez ikakvih izmena na drugom računaru sa C prevodiocem u sebi. C prevodilac je moguće napisati sa manje od 10.000 linija instrukcija u C jeziku. Korišćenjem savremenih programskih pomagala i uz skromnu veličinu C jezika, moguće je napisati kompaktan C prevodilac za novi računar i nepunih par meseci.

Uskoro C standard

Široka primena C jezika, slično „juniksu", dovela je do raslojavanja u načinu njegovog korišćenja. Interesantno, iako vezan za „juniks", više od 75% korišćenih C kompajlera je instalirano na mašinama bez „juniks" sistema. Masovnost i proizvoljnost upotrebe C jezika je neminovno donela na dnevni red pitanje standardizacije — šta bi trebalo a šta ne bi da bude deo osnovne verzije.

Juna 1983. godine ANSI (American National Standard Institute) je formirao grupu sa zadatkom da pripremi predlog standardne definicije C jezika. Grupa poznata kao X3J11 se redovno sastaje svakog tromesečja. Na sastancima — dostupnim publici, normalno uz cenu ulaznice od 75 koja važi za čitavu godinu — raspravljaju se problemi iz tri generalne oblasti: problemi jezika, biblioteke rutina i razvojne sistemske situacije. Predsedavajući na ovim sastancima su Lari Rosler (AT&T), P.J. Plauger (Whitesmiths, Ltd.) i Ralf Franer (Konsultant).

```
/* izvorni C program */
#include <stdio.h>

tip funkc_a(..);
{
 ... /* spoljna funkcija */
}

main()
{
 .
  {
   .. /* unutrasnji blok */
  }
 .
 .
 .
}

tip funkc_b(...);
{
 .. /* spoljna funkcija */
}
```

C Preprocesor ---> C Kompajler

Slika 8.

Grupa sarađuje sa mnogim zainteresovanim organizacijama iz softverske industrije i poznatim IEEE udruženjem. Glavni principi ravnanja grupe su da se nepotrebno ne uveća broj ključnih reči i time ne ugrozi minimalnost Ca, da mu se ne umanji prenosivost i da se ne raskine prirodna veza sa „iuniksom". Zvanično usvajanje C standarda očekuje se 1987. godine.

I pored nesumnjivih prednosti, C nije naročito popularan u Evropi. Na drugoj strani okeana, u Sjedinjenim Državama, u zemlji sa armijom od nekoliko miliona aktivnih programera, jezik C je svakako pravi „hit". Najkritičniji aspekt računarske industrije je relativno nizak nivo produktivnosti u softverskom inženjerstvu. Vreme potrebno za projektovanje i realizaciju hardverskog dela računarskog sistema u poređenju sa vremenom potrebnim za razvoj i uspešnu implementaciju softverskog dela je gotovo zanemarljivo.

Ma koliko se autori „modernih" programskih jezika treće i četvrte generacije ponosili svojim dostignućima, nesavršenost i nedorečenost trenutno raspoloživih softverskih oruđa je evidentna. Enormne dužine programa relativno ograničenih mogućnosti i podebeli softverski priručnici dovoljan su pokazatelj „ubojitosti" modernog softvera. Iz ove perspektive, u poređenju sa zatečenim jezicima, C je jezik uvećanih mogućnosti i povećane jednostavnosti upotrebe. Kao takav, C definitivno predstavlja krupan a istovremeno i neminovan korak unapred — korak ka savršenim programskim pomagalima budućih generacija programera.

Dr Radomir A. Mihajlović

Od sledećeg broja

Poliklinika C

Stručni saradnik „Računara" iz Njujorka prof. dr Radomir A. Mihajlović pripremio je opširnu školu jezika C — polikliniku u kojoj će „bolesnike" od bejzika, paskala i fortrana pokušati da „izleči" ovim sve popularnijim programskim jezikom koji, de facto, postaje svetski standard.

23/nizak nivo a visoke grane

Kad se zakon uspava

programeri bez zaštite

U Okružnom sudu u Beogradu uskoro će na optuženičku klupu sesti jedan neobičan svat — ovdašnji Gradski elektronski centar — da odgovori na optužbe dva mlada programera da je neovlašćeno preuzeo njihov program za kompjutersku obradu poreskih podataka. Kopiranje i preprodaju softvera u ovoj zemlji niko nikada nije smatrao ni prekršajem ni zločinom, niti je iko ikada zbog toga izašao pred lice pravde. Da li će ovo (za sada) jedinstveno suđenje promeniti naš odnos prema programerima i jednoj novoj, izuzetno važnoj oblasti intelektualnog stvaralaštva?

Zlatne osvajačke godine računara su prošle i nastupilo je vreme surove međusobne borbe za njihov opstanak na tržištu. Još pre desetak godina — sa serijskom proizvodnjom jevtinih računara — čovek se od pasivnog svedoka tehnološke revolucije pretvorio u njenog aktera: masovnog kupca i korisnika kompjutera. U ta konjunkturna vremena neprestano su nicale nove firme, novi tipovi računara i sve je imalo proću.

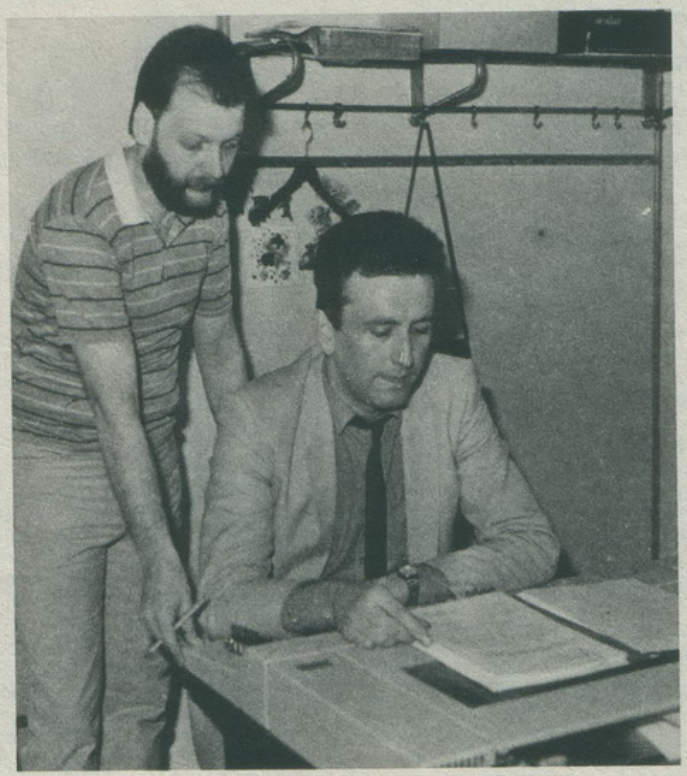
Međutim, publika je u međuvremenu postala znalačka i probirljivija, a konkurencija sve veća. Ona je terala proizvođače na neprestana, makar i mala, poboljšanja i unapređenja koja su ugrađivali u svoje proizvode. Ponekad je — u nedostatku revolucionarnih ideja — to bio samo puki estetski detalj, koji će odvući pažnju sa konkurentskog proizvoda. Kopiranje programa, tehničkih rešenja, pa i kompletnih proizvoda sa izmenjenom kozmetikom, postali su svakodnevna praksa u proizvodnji računara, koja su se solidarno i prećutno tolerisala.

Ali, idila je po svemu sudeći završena.

Računarski spor u Beogradu

Bilo je pitanje dana kada će se i kod nas — sa razvojem računarske kulture — desiti slični slučajevi. I desilo se: doduše, ne u proizvodnji, već u programiranju računara. Pred Okružnim sudom u Beogradu našla se tužba dvojice mlađih stručnjaka-istraživača protiv Gradskog elektronskog centra Zavoda za statistiku u Beogradu.

Rajko Cvetković, inženjer organizacije rada i **Miloš Panić,** ekonomista, smatraju da su oštećeni neovlašćenim korišćenjem njihovog programa za kompjutersku obradu poreskih podataka od strane GEC-a.



U nebranom računarskom grožđu: Miloš Panić i Rajko Cvetković

Tačnije, optužuju ovu gradsku instituciju da je plagirala, pozajmila i kopirala deo njihovog programa, prikazujući ga kao svoj!

Kako su se uopšte Cvetković i Panić našli u ovom nebranom računarskom grožđu?

U „inkriminisano" vreme obojica su radili u upravama društvenih prihoda beogradskih opština Zvezdara i Palilula. Još 1969. godine Gradski elektronski centar je počeo automatsku obradu podataka na svojim mašinama, da bi devet godina kasnije (1978) svih 16 opština dobilo računare NIXDORF 8820 za takozvano „obuhvatanje", knjiženje podataka, dok je njihova obrada i dalje ostala u nadležnosti GEC-a. Ideja je bila u tome da se elektronski „mozaik" Beograda rastereti bušenja kartica.

Posle dve godine rada na tim mašinama i sagledavanja njihovih mogućnosti, Panić i Cvetković sa još trojicom kolega iz opština Voždovac, Rakovica i Stari grad, samoinicijativno su uradili elaborat o modernizaciji poslovanja poreskog knjigovodstva u opštinskim službama. Gradski i opštinski oci su januara 1982. usvojili taj elaborat, da bi u sledećoj fazi njegovi autori uradili i programski paket (software) za samostalnu knjigovodstvenu obradu u službama.

— Sve smo to radili uz naš redovan posao, tvrdi Cvetković. — Sve u svemu, u sledeće dve godine taj je paket „proradio" u opštinskim službama, ali uz naš veliki angažman. Morali smo, naime, da koordiniramo primenu našeg programa, jer su opštine bile deficitarne sa stručnim kadrovima: uglavnom su to operativne službe sa bivšim daktilografkinjama i srednjoškolcima.

Prema tvrđenju Cvetkovića i Panića, maja 1984. Gradski elektronski centar je pozvao na sastanak predstavnike opštinskih službi, gde je odlučeno da počev od '85. prestane primena MOPOK-a (zlosrećnog programa naših sagovornika). Zauzvrat, GEC se obavezao da terminalski poveže opštinske mašine sa svojom velikom mašinom IBM 3031 i naknadno uradi novi, aplikativni softver, koji će sve to da podržava.

Izgleda da upravo od tog trenutka počinje da se dubi stručni i ljudski jaz između aktera koji će, eto, završiti na sudu.

— Već posle nekoliko meseci ispostavilo se da Gradski elektronski centar nije daleko odmakao u tom poslu, nastavlja Cvetković. — Krajem godine ponovo su nas pozvali i tražili od nas ponudu za dalji rad MOPOK-a. Januara 1985. sročili smo naš zahtev po kome je svaka opština, sem dve matične u kojima smo bili zaposleni, treba da nam isplati 80.000 dinara (14 puta 8 jednako 112 starih miliona — prim. autora). Posle dva meseca su nas odbili, ne baš učtivo, optuživši nas čak da smo — ucenjivači!

Bliži se i završnica ove računarske rašomonijade: Gradski elektronski centar je ponovo dobio u zadatak da što pre uradi projekat. Marta prošle godine GEC je opštinama distribuirao jedan deo programa za koji Cvetković i Panić tvrde da je čista kopija MOPOK-a! Na to su ih, kako vele, upozorile same opštinske uprave prihoda.

Tako se, na kraju, aprila 1985. optužba za povredu autorskih prava, koja tvorci programa prethodno nisu ni zaštitili, našla pred nadležnim Okružnim sudom u Beogradu.

Šta kaže druga strana?

Jeste — nije kopija

U gradskom elektronskom centru u Tiršovoj 1 nisu nas baš dočekali ozarena lica. Pre bi se reklo — kao nužno zlo. Direktor **Radivoj Gavrić** je na svoju ruku pojačao „prijemnu ekipu". Pored njega, tu su još i **Nevenko Antunac,** načelnik Gradske inspekcije društvenih prihoda, **Olivera Nikolić** i **Dragoljub Jakovljević,** projektanti u GEC-u. Ništa od one borbenosti, žara i ubedljivosti naših prethodnih sagovornika. Mukotrpno se kristališe tema.

Direktor Gavrić smatra za shodno da podvuče:

— Računari NIXDORF nabavljeni su za opštine s ciljem da se vremenom povežu sa IBM mašinom u našem centru. Međutim, od početka su upotrebljavani kao samostalna oprema. To se, jednostavno, dopalo opštinskim službenicima i tako je zanemaren osnovni cilj.

Ko je za to kriv?

— Inostrani proizvođač nam nije obezbedio softversku podršku za povezivanje mašina. Predstavništvo „Balkanija" je ukinuto, a s njim je prestala i ta vrsta obaveza.

Gradski inspektor Antonac praktično je vodio čitav posao koji je na kraju ispao sporan:

— Sa uvođenjem NIXDORF-a školovali su se i kadrovi. Nesumnjivo je da su drugovi koje pominjete dali neka bolja rešenja za uspešnije korišćenje tih računara. Drugim rečima, iz novog programa se mogao crpsti veći broj podataka. Zato je i prihvaćen. Međutim, ne radi se o revolucionarnim promenama: sistem poreskog knjigovodstva propisima je strogo regulisan i valja ih se pridržavati.

Antonac tvrdi da je među samim autorima programa došlo do nesuglasica i da su Cvetković i Panić jednostavno odstranili ostale saradnike: „Morao sam čak da smirujem i njih i njihove šefove koji su mi se žalili da redovan posao trpi zbog rada na MOPOK-u".

Nezadovoljni programeri su posebno kivni na Oliveru Nikolić iz GEC-a koja je i glavni potpisnik spornog, navodno plagiranog programa. Ona odlučno odbacuje većinu činjenica i optužbi Panića i Cvetkovića:

— Između njihovog i našeg programskog paketa uvek je postojao problem povezivanja. Odluka da odustanu je bila njihova. Po nalogu grada GEC je prihvatio da na svom velikom sistemu uradi deo poslova iz tog paketa. Deo na velikom računaru nije kopija, ali je činjenica da su u drugom delu morale biti zadržane njihove procedure. Da li je to kopija?

Poslednji odgovor na to pitanje daće sudija Vesna Kušić. Sud je već zatražio veštačenje jedne komisije Elektrotehničkog fakulteta koja treba da uporedi dokumentaciju „zaraćenih" strana.

Staromodni propisi

Da li će propisi pomoći ili odmoći u rešavanju ovog spora, dosad nezabeleženog u našoj računarskoj praksi?

Pravni savetnik u Autorskoj agenciji za Srbiju **Stanka Krstić** kaže da je prodor računara bio brži od zakonskih inovacija. Praktično, nijedan važeći propis ne pominje programiranje kompjutera kao autorsko delo. Nema ga u saveznom Zakonu o autorskim pravima iz 1978, a u Zakonu o zaštiti pronalazaka, tehničkih unapređenja i znakova raspoznavanja izričito je naglašeno da se računarski programi ne smatraju pronalaskom.

— Međutim, u praksi mi i računarske programe tretiramo po Zakonu o autorskom pravu, kaže Stanka Krstić. — Oslanjemo se na član 3. u kome stoji „da se autorskim delom smatra tvorevina iz oblasti književnosti, nauke, umetnosti i **drugih oblika stvaralaštva** . . ." U skladu s tim, vršimo i registraciju programa onih autora koji i na taj način žele da se obezbede od eventualnog plagijata. Pre svega se to odnosi na softverske programe."

Vesna Lazić iz iste agencije tvrdi da se najviše oslanjaju na tumačenja sarajevskog profesora **Pajić Vojislava,** eksperta za autorska prava, koji pominje i računarske programe kao intelektualno, autorsko delo. Hendikep je u tome što mnogi još smatraju da ovi programi „ne služe direktno čoveku poput tradicionalnih autorskih dela, već mašini, dakle tek posredno — čoveku".

U svakom slučaju, ako hoćemo da držimo korak sa tehnološkom revolucijom, programiranje računara zaslužuje bolji tretman od „drugih oblasti stvaralaštva". Time bi se ne samo izbegla nemila igra ljudske sujete, zavisti, emotivnog nadmetanja, krađe i prekrađe ideja, već bi se — što je važnije — ohrabrila generacija mladih stručnjaka programera da se, s osloncem na zakonske propise, upusti u nove poduhvate.

Utoliko je i opisani prvi računarski spor u nas ujedno i probni balon društvenog odnosa prema ovom novom obliku tehničke inteligencije.

Mihajlo Kovač

25/programeri bez zaštite

Akcije *Komodor 64*

Programi za kertridž

Prve domaće laste

Pokrećući akciju za pisanje programa za kertridž, nismo očekivali preveliki odziv čitalaca. Temu smo, ipak, smatrali dovoljno interesantnom, pa smo bili spremni da gotovo ceo posao preuzmemo na sebe. Na sreću, prevarili smo se. Priloga zaista nije stiglo mnogo, ali po svom kvalitetu i pristupu predstavljaju izuzetnu vrednost. Izgleda da smo ohrabrili jedan prilično slabo poznat segment korisnika računara koji poseduju izvanredno znanje, ali ga koriste isključivo za — svoje potrebe.

Nije nas iznenadilo što se svi prilozi bave grafičkim mogućnostima „komodora". Ovo je, kao što svi vlasnici znaju, Ahilova peta ovog popularnog računara. Zanimljivo je da nigde u radovima nije čak ni pomenuta reč sprajt. Dobre igre se očigledno ne pišu proširenim bejzicima. Naročito nas je iznenadilo što su programeri svoju pažnju usmerili na samo jedan grafički mod i to onaj sa najvišom rezolucijom. Multikolor se pojavljuje samo stidljivo — više kao demostracija mogućnosti da priložene rutine mogu da rade i na taj način. Iznenađenje je, svakako, prijatno, jer pokazuje da autori ovih programa grafiku koriste za ono za šta je i najvrednija — grafičku prezentaciju rezultata neke obrade, predstavljanje funkcija i slično, za šta mogućnost korišćenja više boja nema značaja. Dakle, sudeći po vašim reakcijama, bili smo potpuno u pravu kada smo tvrdili da nam nisu potrebna velika proširenja tipa „Simon's Basic", već kratki i efikasni dodaci koji samo ispravljaju ono što je Komodor propustio.

Posebno bismo hteli da istaknemo dva priloga koje smo dobili od Zoltana Pekića i Dušana Božina.

Zoltan Pekić nas je iznenadio izuzetnim pristupom pri pisanju svojih rutina. Posebno nas je oduševio originalni način razmišljanja — rutina koja na osnovu stanja registara video procesora izračunava adresu video memorije visoke rezolucije. Dakle, gde god se nalazila, na koji god način da je pozvana, pozicija slike visoke rezolucije će biti tačno izračunata i zatim kao parametar biti preneta u PLOT, LINE i slične rutine. Pošto ćemo u konačnom programu obavezno imati i naredbu za izbor režima rada video procesora, tačno će biti definisano, upotrebom nekoliko novih sistemskih varijabli, koji je režim izabran i gde se video memorija nalazi. Zato će, na žalost, ova rutina biti suvišna. Ostale rutine, PLOT I LINE, takođe su odlične. To se posebno odnosi na PLOT, koja je efektno rešena upotrebom tablica rasporeda bitova čime je, na račun nešto veće potrošnje memorije, dosta dobijeno na brzini. Za tako često pozivanu rutinu ovo je od primarnog značaja, pa ćemo u potpunosti prihvatiti njegovo rešenje.

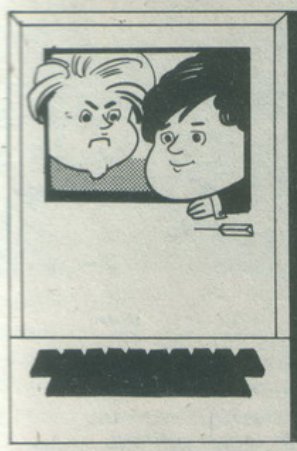
Druga Pekića smo zamolili da svoj prilog uobliči prema manjim sugestijama koje smo dali, pa njegov rad u celosti, očekujemo u nekim od sledećih brojeva „Računara".

Dušan Božin je poslao svoje rešenje PLOT rutine. Da je kojim slučajem Komodor njemu poverio pisanje programa za ROM-ove C64, sigurno je da bi potrebno područje za sistemske varijable bilo upola manje! Dušan na najbolji način ilustruje kako se veštim korišćenjem registara i steka pišu brze rutine bez čestog obraćanja memoriji. Njegov prilog, ipak, nešto odstupa od koncepcije koju smo na početku usvojili. Osnovno je da sve rutine koje pripadaju operativnom sistemu budu potpuno nezavisne od toga kako će biti primenjene u proširenjima bejzika. Ulazni parametri (konkretno X,Y koordinata i modalitet postavljanja tačke) treba da se prenose u rutinu na način primeran operativnom sistemu — dakle putem registara ili memorijskih lokacija, a ne na „bejzik" način uzimanjem parametara iz bejzik linije. Time PLOT postaje dostupan i programima u mašinskom jeziku, što je u drugom slučaju nemoguće. Dušan kaže da ima još i DRAW i TEXT rutine koje želi da doradi, pa će, ako nam se dopadnu ove koje je poslao, poslati i njih. Pozivamo ga da to obavezno učini, uz molbu da uvaži generalnu primedbu koju smo dali.

*Dušan je postavio i pitanje: kako će izgledati nove bejzik naredbe koje budemo uveli — da li će to biti pune reči ili samo slova uz neki zajednički prefiks? To ni nama još uvek nije do kraja jasno. Ako to budu pune reči, biće neophodno menjati rutine za tokenizaciju i listanje programa, što može odneti dosta memorije. Ako, pak, budu oblici tipa *P i slični, imaćemo više mesta za korisnije stvari, ali i slabiju razumljivost programa. Dakle, na vama je da donesete odluku.*

Sve one koji nameravaju da se uključe u akciju molimo da, ako su programi iole duži, priložе i kasetu sa izvornim programom, kako bismo ubrzali postupak oko provere priloga i njihovog objavljivanja. Oni koji nemaju štampač ne treba da brinu — kaseta će biti dovoljna, a listing ćemo napraviti mi.

Zoran Životić



Peek & poke show

Mikrotragedija

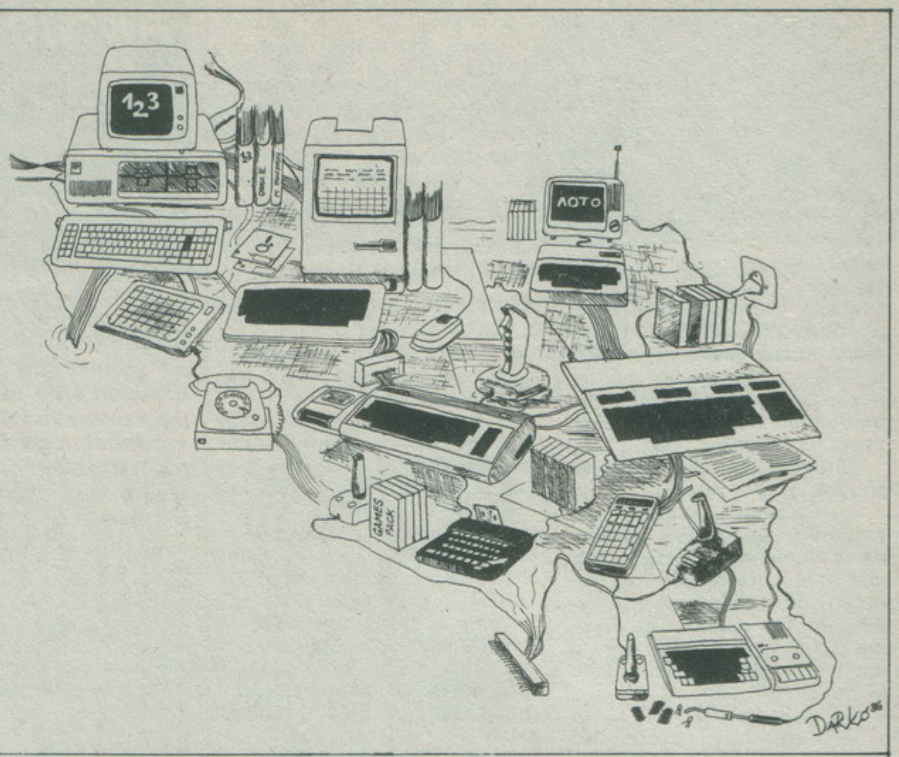
Dogodila se jedna tužna stvar. Hiljade novinarskih računardžija i računarskih novinara na ovom belom svetu neće više moći da kinje i maltretiraju svoju omiljenu žrtvenu figuru. Ser Klajv je napustio računare! Nije umro. Jednostavno je prodao svoju firmu nekom drugom i otišao da se bavi nekim drugim stvarima (telefonima i sličnim cakama). To je korak koji se može nazvati izdajstvom, ali i, kada se setimo kako mu je išlo u poslednjih godinu-dve merom zdravog razuma.

Onaj drugi, tačnije Onaj Drugi, koga smo pomenuli kao kupca nije niko drugi do sjajni i nepredvidivi Alan Šuger-Šećerko, uspešni i ponosni direktor Amstrada. Tako je obavljena direktna primopredaja — najsmotaniji najlukavijem.

Vlasnici malih vrućih crnih kutija i, uopšte, zaljubljenici u „spektrum" i slične mašine ne moraju da strahuju. Šuger je izjavio da neće biti drastičnijih promena u kompaniji. Osim što je sad mnogo verovatnije da će QL, a možda čak i „spektrum", dobiti nekakav disk drajv, recimo od 3 inča. Toliko o tužnim vestima. A sada i jedna dobra — Ser Klajv je napustio računare, ali mu je ostala plemićka titula.

Naš zajednički računar

Čitaocima koji pažljivije prate dnevnu štampu sigurno nije promaklo da je za potrebe Skupštine Jugoslavije nabavljen jedan oveći kompjuter sa mnoštvom monitora, štampača, disk jedinica i sličnih potrepština. „Peek & poke show" je oduševljen tim događajem i smatra da je ovo prvi put da je stvarnost pretekla našu Čip Pobodi Agency. Ima li ubedljivijeg načina da se pokaže da je zaista potrebno masovno uvoditi kompjutere u našu stvarnost od praktičnog primera? A šta je praktičnije nego postaviti jedan lep i moćan računar tako da ga vidi mnoštvo naših vodećih i rukovodećih ljudi. A gde je to? U Skupštini. „Peek & poke show" je uveren da je dalja taktika već razrađena i da će se u dogledno vreme mereno danima slični (malo manji i malo slabiji) računari, kao logičan korak, pojaviti i u skupštinama republika i pokrajina, nešto kasnije i u opštinama, a jednog dana, ko zna, možda čak i u mesnim zajednicama. Nakon toga će, nadamo se, na red doći i škole.



Karikatura: Darko Stanojević

Top lista „Crni biseri"

SUBGO, KEPO, TOGO, RINTP

Naučite kompjuter da govori šatrovački

U najnovijem broju jedne domaće revije (između ostalog i) za kompjutere pojavio se senzacionalan program, dat u vidu listinga, kojim ćete naučiti (nagovoriti?) svoj kompjuter da govori šatrovački. Da, dobro ste pročitali, šatrovački, odnosno sleng. Ovaj pravi biser domaćeg programerskog umeća, koji odražava trenutni (visok) nivo istog, trebalo bi da poseduje svaki pravi korisnik kompjutera.

Na žalost, program je dat samo za „komodor 64", pa će korisnici ostalih tipova računara biti vrlo uskraćeni. Ali, poznavajući agilnost i entuzijazam domaćih programera, ne sumnjamo da će se uskoro pojaviti prepevi i za ostale računare.

Predlažemo da ovaj program automatski skoči na prvo mesto naše nove top liste „Crni biseri", koja trenutno izgleda ovako:

1. *Šatrovački za C–64*
2. *Proračun malih trafoa*
3. *Memo*
4. *Proračun aktivnih filterskih skretnica*
5. *Štrumpfmatika*

Pozivamo čitaoce da nam šalju svoje predloge za ovu listu. Broj predloga nije ograničen, a najvrednijima za nagradu šaljemo knjigu „Prva šljapkanja po bejziku".

Veliži keriha!

Čip Pobodi Agency

Budući Oskar

Od svojih doušnika u svetu filma i gubljenja vremena saznajemo da poznati režiser Džon Karpenter, poznat po svojim horor remek-delima, priprema novi film po scenariju Andy MC Colucka pod radnim naslovom „Bio sam YU pirat i preživeo sam". Prema ovim istim doušnicima glavnu ulogu bi trebalo da igra Silvester Stalone, iako je on kao uslov za igranje zahtevao da se dopiše nekoliko scena tuče i razmene programa.

Jezici u sosu

Posle fortrana, prologa, bejzika i kobola, kao i drugih jezika koji su do sada korišćeni za eksperimente na srednjoškolcima, sve se glasnije čuju zahtevi da se uvede jedan od novih „User-friendly" jezika za obuku mladih. Jezik se zove „Boulderdash III" i čim čujemo njegove osnovne karakteristike javićemo vam.

Kvantni skok nadole

Nobelova fondacija je odlučila da proširi spektar nagrada koje svake godine dodeljuje. Od ove godine postojaće i Nobelova nagrada za stvaranje električnih tricikala i mrtvih tastatura. Tako će najbolji izumitelji sveta moći da budu zaista nagrađeni za svoj izuzetan trud.

Posle višegodišnjih pregovaranja firma Drp elektronics izbacuje na tržište napravu koja bi trebalo da omogući da ORIC emulira IBMa PC. Naprava je proizvedena po zahtevima jugoslovenskog tržišta i težiće oko 86 kilograma. Cena će joj biti između 70063 dolara i 857634547580 dinara. Očekuje se da će biti široko prihvaćena.

Primer 6:
Generator slučajnih brojeva

```
3600            50   ORG &3600
3600            60   OPT 7
35F0            70 SEED EQU &35F0
3600            80 RNDGEN
3600 2AF035     90   LD HL,(SEED)
3603 7C        100   LD A,H
3604 E680      110   AND &80
3606 57        120   LD D,A
3607 7C        130   LD A,H
3608 E608      140   AND &8
360A 07        150   RLCA
360B 07        160   RLCA
360C 07        170   RLCA
360D 07        180   RLCA
360E AA        190   XOR D
360F 5F        200   LD E,A
3610 7D        210   LD A,L
3611 E604      220   AND &4
3613 0F        230   RRCA
3614 0F        240   RRCA
3615 57        250   LD D,A
3616 7D        260   LD A,L
3617 E601      270   AND &1
3619 AA        280   XOR D
361A 0F        290   RRCA
361B AB        300   XOR E
361C 17        310   RLA
361D CB15      320   RL L
361F CB14      330   RL H
3621 22F035    340   LD (SEED),HL
3624 C9        350   RET
3625           360   >
```

Kada smo već kod igara, navedimo i jedan generator slučajnih brojeva. Program sa naše slike se poziva jednostavnim CALL RNDGEN i vraća u HL slučajan broj između 0 i &FFFF. Princip rada je prilično jednostavan (zato generator nije naročito dobar, ali je bar brz) pa ćemo vam ostaviti da ga sami otkrijete. Ako nećete previše da se zamarate, zapamtite jedino da pre *prvog* poziva RNDGEN treba u ćelije SEED i SEED+1 upisati bilo koji šesnaestobitni broj različit od nule.

Dejan Ristanović

Z 80 mašinac za početnike

Ovaj umetak, kao što mu i naslov govori, treba da upozna apsolutnog početnika sa programiranjem na mašinskom jeziku mikroprocesora Z80. Takav apsolutni početnik će se, čitajući ove redove, možda zapitati „šta je to mikroprocesor" ili „odakle oznaka Z80". Odgovore na ta i druga pitanja o značenjima osnovnih pojmova koje ćemo koristiti kroz čitav umetak treba da pruži ovo uvodno poglavlje.

Mikroprocesor je centralni deo svih kućnih računara i različite druge opreme koju nazivamo „inteligentnom". Ovaj termin označava čip koji može da se programira za procesiranje, tj. obradu nekakvih podataka. Ono *mikro* u njegovom imenu potiče od činjenice da se čitava komplikovana elektronika nalazi u jednom jedinom integrisanom kolu koje ćete, ako se odvažite da skinete poklopac vašeg kompjutera, obično prepoznati po tome što ima najveći broj nožica, tzv. pinova. Kao i kod većine reči koje počinju sa *mikro,* postojale su i postoje njihove verzije bez tog prefiksa: *procesorom* ili *procesorskom pločom* nazivamo deo nekog velikog kompjuterskog sistema koji obrađuje informacije.

Moderni kućni računari u sebi sadrže više čipova koji bi se mogli svrstati pod našu definiciju mikroprocesora: ukoliko, na primer, vaš računar povežete sa diskom, kontrolu ovoga uređaja će obavljati disk kontroler, tj. specijalizovani mikroprocesor. Ako pročitate da vaš kompjuter ima video kontroler, znaćete da se radi o još jednom mikroprocesoru koji se brine za generisanje slike; sličnih primera ima još dosta. Mi se, međutim, nećemo baviti specijalizovanim kontrolerima, već mikroprocesorom Z80 koji se nalazi „u centru" dobrog dela danas popularnih modela kućnih računara. Centralni mikroprocesor je zadužen za izvršavanje mašinskih programa koje korisnik startuje (dok, na primer, izvršavate bejzik program, traje izvršavanje bejzik interpretatora koji je upisan u ROM vašeg kompjutera) i za koordinaciju rada ostalih specijalizovanih kontrolera ugrađenih u računar. Korisnik obično nema nikakvih mogućnosti (a ni potreba) da komunicira sa bilo kojim hardverskim modulom svoga računara **osim** posredstvom centralnog mikroprocesora.

Z80, jasno, nije pretplaćen da uvek budu glavni: ako posetite neki računski centar, saznaćete da se za komunikaciju centralnog procesora (bez onoga *mikro)* sa terminalima brine jedan mikroprocesor, dok drugi neprekidno komunicira sa paketom diskova. U poslednje vreme se mogu pronaći i kućni računari iz srednje klase koji poseduju po dva ista ili različita mikroprocesora, od kojih se jedan brine za ulaz i izlaz podataka a drugi za opsluživanje zahteva korisnika. Za nas je, međutim, sasvim dovoljna pretpostavka da naš kompjuter ima samo jedan mikroprocesor.

Što se oznake Z80 tiče, u njoj ne treba tražiti neki poseban smisao. Z dolazi od imena firme *(Zilog)* koja proizvodi ovaj čip. Ono *80* verovatno treba da podseti da su Z80 proizveli „odbegli" Intelovi konstruktori koji su neposredno pre toga dizajnirali i 8080, mikroprocesor sa kojim Z80 ima dosta sličnosti i od koga je značajno moćniji. Moguće je da je u vaš računar ugrađen

mikroprocesor Z80A, Z80B ili neki sličan. To ne treba da vas zabrinjava: radi se o nekoj od verzija Z80 koja se od osnovne verzije razlikuje samo u nekim hardverskim detaljima; mašinski jezik je identičan onome koji mi obrađujemo.

Bit, bajt, reč

Čitajući kompjuterske časopise verovatno ste saznali da je Z80 **osmobitni** mikroprocesor, a da postoje i **šesnaestobitni** kao i nekakvi za koje nije baš jasno kakvi su (jedan takav je ugrađen u kontraverzni Sinclairov QL). Izgleda smešno da nekome ko već izvesno vreme poseduje računar i ko bar jednom nedeljno sa prijateljima raspravlja o tome koji model kompjutera ima više kilobajta memorije objašnjavamo šta je bit i šta je bajt. Iskustvo nam, međutim, pokazuje da se ovakvi razgovori zasnivaju na relativnom poređenju podataka: jasno nam je da je računar koji košta 400 australijskih dolara duplo jeftiniji od računara koji košta 800 australijskih dolara, ali nam to ne pomaže da znamo koliko *stvarno* košta svaki od ovih računara; da bismo to saznali treba da znamo i kurs australijskog dolara. Zato ćemo uzeti slobodu da, jednom za svagda, objasnimo pojmove **bit, bajt** i **kilobajt.**

Bit je najmanja jedinica za količinu informacija. Ako vas neko pita *da li si kupio „Računare 15"* možete da odgovorite sa **da** ili sa **ne** (ako odgovorite sa *ne,* a ipak čitate ovaj tekst . . . pa, kod nas se i onako smatra da krađa knjige i časopisa nije krađa). Uz malo dobre volje odgovoru **da** možete da pridružite broj 1, a odgoru **ne** broj 0. Za odgovor na pitanje je, dakle, dovoljan samo jedan bit informacija.

Postoje, naravno, i pitanja na koja se ne može dati jednobitni odgovor. Na pitanje *koliko koštaju „Računari 15"* ne možete da odgovorite sa **da** ili **ne;** biće potrebno da navedete neku brojku. Ta brojka se, videćemo u sledećem poglavlju, može pretvoriti u drugu brojku koja se sastoji od jedinica i nula, pa ćete tako možda reći da časopis koji čitate košta 100101100 dinara. Pošto ovaj broj ima 9 binarnih cifara (bit=binary digit ili binarna cifra), odgovor na pitanje o ceni ove knjige nosi 9 bita informacija.

Bit je, kao što vidimo, vrlo mala količina informacija: kao što nikome neće pasti na pamet da izražava rastojanje od Zemlje do Sunca u milimetrima, tako se i kapacitet memorija nekog kompjutera nikada ne izražava u bitovima. Obzirom da su stepeni dvojke (2°1,2°2,2°3 . . .) vrlo značajni za rad sa binarnim brojevima, uobičajeno je da se 2°3=8 bitova nazivaju bajt. Broj koji smo ranije naveli kao cenu našeg časopisa bismo sada napisali u obliku *1 00101100* i rekli da on nosi jedan bajt i još jedan bit inofrmacija (primetimo da je to isto što i 1*8+1=9 bita). Nije, međutim, uobičajeno izražavati količinu informacija na ovako „razlomljen" način, pa ćemo reći da su za izražavanje cene „Računara" potrebna dva bajta informacija; naših devet bitova ćemo, da bismo se uklopili u ovu definiciju, dopuniti vodećim nulama, pa ćemo tako reći da „Računari 15" koštaju *00000001 00101100* dinara. Uvođenjem vodećih nula smo, uzgred, ostavili dosta prostora ža buduća poskupljenja . . .

Osam bita ili jedan bajt je najmanja količina informacija kojoj mikroprocesor može da pristupi. To znači da će, na primer, naredba LOAD A, (20000) dovesti u akumulator (jedan od registara procesora; nije bitno ako ovaj termin još ne razumete) sadržaj memorijske ćelije čija je adresa 20000 i to svih osam bita koje sadrži ta ćelija. Ukoliko nam je potrebno da pojedinačno menjamo neke od tih bitova, moraćemo da dovedemo čitavu ćeliju u akumulator, modifikujemo je, a zatim je celu vratimo u memoriju.

Videli smo da mikroprocesor može (i mora) da u memoriju odjednom dovede **8** bita informacija, tj. jedan bajt. Zbog ove karakteristike mikroprocesor Z80 nazivamo **osmobitnim.** Danas pravo građanstva sve više stiču **šesnaestobitni** mikroprocesori koji, dakle, u jednom pristupanju memoriji mogu (i obično moraju) da prenesu 16 bita podataka. Na tržište se stidljivo probijaju i skupi **tridesetdvobitni** mikroprocesori dok neki veliki sistemi, osim tridesetdvobitnih koriste i **šezdesetčetvorobitne** procesore. U principu je (mikro)procesor koji može da obrađuje podatke koji se sastoje od većeg broja bitova brži i sposobniji za kontrolu veće operativne memorije ali je zato takav (mikro)procesor, kao i ostali čipovi koji ga prate, daleko skuplji.

Kada već govorimo o šesnaestobitnim procesorima, moramo da pomenemo još jednu definiciju bajta. Bajt smo, naime, definisali kao osam bita jer naš mikroprocesor može odjednom da ih pročita baš toliko. Ako radimo sa šesnaestobitnim procesorom imalo bi smisla definisati bajt kao 16 bita što su neki autori i radili. Bolje je, međutim, da za šesnaest bita usvojimo termin **reč (word).** Kako su se procesori razvijali, njihovi konstruktori su morali da usvoje i termine **double word** (32 bita), a onda i **long word** (64 bita); sledeća veća jedinica za količinu informacija **double long word** (128 bita) još nije zaživela u praksi. U terminologiji mikroprocesora Z80 **bajtovi** i **reči** će nam biti sasvim dovoljni.

U literaturi, posebno domaćoj, srećemo još jednu skrivenu definiciju reči bajt: često ćemo pročitati nešto poput „ova naredba stavlja broj iz akumulatora u pedeset hiljaditi bajt memorije". Znajući da je bajt jedinica za količinu informacija, gornja rečenica je sasvim besmislena: da li ste nekada čuli da je neko sipao flašu vode u deveti litar posude? Kada bi, međutim, pomenuta posuda bila na neki podeljena u šesnaest komora, a u flaši bio tačno litar vode, rečenica „sipao sam flašu vode u deveti litar poseude" bi mogla da se protumači kao „sipao sam jedan litar vode iz flaše u komoru posude čiji je redni broj 9"; vidimo da smo na prvi način uštedeli 15 reči. Pošto je memorija računara podeljena na ćelije od kojih u svaku staje po osam bita, rečenica „ova naredba stavlja broj iz akumulatora u pedeset hiljaditi bajt memorije" bi mogla da se prevede kao „ova


```
363E          520   ! JE ASCII KOD U A NA
363E          530   ! EKRANU.
363E E5       540   PUSH HL
363F C5       550   PUSH BC
3640 D5       560   PUSH DE
3641 CD2000 570   CALL #ROMAEKR
3644 D1       580   POP DE
3645 C1       590   POP BC
3646 E1       600   POP HL
3647 C9       610   RET
3648          620 ISPISHL
3648          630   ! ISPISIVANJE BROJA KOJI
3648          640   ! JE UPISAN U HL NA EKRAN.
3648 C5       650   PUSH BC
3649 D5       660   PUSH DE
364A E5       670   PUSH HL
364B CDF308 680   CALL ROMHLEKR
364E E1       690   POP HL
364F D1       700   POP DE
3650 C1       710   POP BC
3651 C9       720   RET
3652          730 ULAZ
3652          740   ! OCITAVANJE TASTATURE
3652 E5       750   PUSH HL
3653 C5       760   PUSH BC
3654 D5       770   PUSH DE
3655 CDF50C 780   CALL ROMTASTA
3658 D1       790   POP DE
3659 C1       800   POP BC
365A E1       810   POP HL
365B C9       820   RET
365C          830   ! TEKSTOVI PORUKA
365C 5D4FA7 840 TEKST1 TEXT "POGODIO SAM (OPET)!"
366F 0D0D   850   WORD &0D0D
3671 5AA14D 860 UPUT   TEXT "ZAMISLI BROJ IZMEDJU 1 I 1023"
368E 0D     870   BYTE &0D
368F 412C4A 880 TEXT "A JA CU GA POGODITI!"
36A3 0D0D   890   WORD &0D0D
36A5 504F53 900 TEXT "POSLE SVAKOG MOG POKUSAJA"
36BE 0D     910   BYTE &0D
36BF 505249 920 TEXT "PRITISNI:"
36C8 0D0D   930   WORD &0D0D
36CA 202D20 940 TEXT " - M AKO JE TVOJ BROJ MANJI;"
36E6 0D     950   BYTE &0D
36E7 202D20 960 TEXT " - V AKO JE TVOJ  BROJ VECI;"
3703 0D     970   BYTE &0D
3704 202D20 980 TEXT " - P AKO SAM POGODIO."
3719 0D0D   990   WORD &0D0D
371B 00     1000  BYTE 0
371C        1010  >
```

Možda ste se odlučili za upoznavanje mašinskog jezika da biste zatim pisali igre (i prodavali ih Englezima)? Evo jedne sasvim jednostavne igre primerene znanju koje smo do sada stekli: High-Low.

Zamislite broj između 0 i 1023 i startujte naš program. Računar će ispisati neki broj, a zatim ćete pritisnuti V ako je broj koji ste zamislili veći, M ako je manji a P ako je računar pogodio. Videćete da će svaki broj koji zamislite biti pogođen u najviše 10 pokušaja.

Igra je sasvim nezavisna od bejzika, što znači da ulaz i izlaz podataka obavlja sopstvenim potprogramima ISPIS, ISPIS_HL i ULAZ. Ovi potprogrami, sa svoje strane, pozivaju rutine iz ROM-a čije adrese možete da saznate konsultovanjem knjiga tipa „The Complete Spectrum ROM Disassembly" ili naših ranijih umetaka. Moraćete, dakle, da promenite adrese navedene u naredbama 50, 60 i 70.

Primetite da smo ulaz i izlaz podataka odvojili u posebne potprograme vodeći računa o činjenici da potprogrami iz ROM-a nekih računara ne čuvaju sadržaje opštih registara. Preporučljivo je slediti ovakvu metodologiju rada: ako naredbe zavisne od računara izdvojimo u potprograme i dobro dokumentujemo, naša će se remek dela lako prilagoditi drugim mašinama.

Što se algoritma na kome počiva program High Low tiče, nećemo mu obraćati mnogo pažnje: radi se o običnom binarnom pretraživanju. Ukoliko vam je termin „binarno pretraživanje" stran, pogledajte mali bejzik program koji je potpuno ekvivalentan bitnom delu našeg mašinca.

```
10 PRINT "Zamisli broj izmedju 1 i 1023!"
20 PRINT
30 LOW=0
40 HIGH=1024
50 BROJ=(LOW+HIGH)/2
60 PRINT "Da li je to broj "; BROJ;" (V,M,P)?";
70 INPUT "" X$
80 IF X$= "V" THEN LOW= BROJ: GOTO 50
90 IF X$= "M" THEN HIGH=BROJ: GOTO 50
00 IF X$<>"P" THEN GOTO 60
10 PRINT "Pogodio sam (opet!)."
20 PRINT
30 GOTO 10
```

```
3614 CD1E36 260   CALL NIBL ! ISPISE GA...
3617 F1    270   POP AF
3618 E60F  280   AND &0F    ! IZDVOJI NIZI NIBL...
361A CD1E36 290  CALL NIBL ! I ISPISF GA.
361D C9    300   RET
361E       310 NIBL
361E       320   ! UPISUJE ASCII PREDSTAVU
361E       330   ! NIBLA IZ A U CELIJU NA
361E       340   ! KOJU POKAZUJE DE.
361E FE0A  350   CP 10
3620 FA2536 360  JP M,CIFRA
3623       370   ! RADI SE O VREDNOSTI
3623       380   ! KOJA TREBA DA POSTANE
3623       390   ! SLOVO A-F.
3623 C607  400   ADD 7
3625       410 CIFRA
3625 C630  420   ADD "0"
3627 12    430   LD (DE),A
3628 13    440   INC DE
3629 C9    450   RET
362A       460   >
```

```
3600       50   ORG &3600
3600       60   OPT 7
3600       70 HLASC
3600 7C    80   LD A,H
3601 FE00  90   CP 0
3603 2803  100  JR Z,DVE  ! DA LI JE VECI OD 255?
3605 CD0D36 110 CALL BAJT ! ISPISE HIGH BAJT.
3608       120 DVE
3608 7D    130  LD A,L
3609 CD0D36 140 CALL BAJT ! ISPISE LOW BAJT.
360C C9    150  RET
360D       160 BAJT
360D       170  ! UPISUJE ASCII PREDSTAVU
360D       180  ! BAJTA IZ A U CELIJU NA
360D       190  ! KOJU POKAZUJE DE.
360D F5    200  PUSH AF
360E E6F0  210  AND &F0
3610 0F    220  RRCA
3611 0F    230  RRCA
3612 0F    240  RRCA
3613 0F    250  RRCA      ! IZDVOJI VISI  NIBL...
```

operaciju izvršiti još jednom izazivajući povratak u glavni program. Da smo umesto CALL NIBL: RET napisali JP NIBL (ili, još bolje, JR NIBL štedeći dva bajta), naredba RET bi bila nepotrebna, program bi brže radio i stek bi se manje opterećivao (znate li zašto?). U većini slučaja, međutim, nije vredno uštedeti par bajta i učiniti program nečitljivim: što je program slabije strukturiran i što više „prljavih trikova" koristi, to će u njemu biti više grešaka; te se greške, u krajnjoj instanci, lupaju o glavu vama ili nekom nesrećniku koji će docnije pokušati da modifikuje ili bar razume vaš program. Obzirom na činjenicu da je asembler po svojoj prirodi nepregledan, treba se truditi da se programi što propisnije pišu i što bolje komentarišu. Od ovoga možemo da odstupimo samo u (danas veoma retkim) situacijama u kojima su dragoceni bajtovi i mikrosekunde procesorskog rada.

Primer 5: Jedna jednostavna igra

```
3600       20   ORG &3600
3600       30   OPT 7
3600       40   ! ADRESE RUTINA IZ ROM-A:
0020       50 ROMAEKR EQU &20
03F3       60 ROMHLEKR EQU &3F3
0CF5       70 ROMTASTA EQU &CF5
3600 217136 80  LD HL,UPUT ! ISPISIVANJE UPUTSTAVA
3603       90 PISI
3603      100   ! ISPISUJE TEKST POCEVSI OD
3603      110   ! MEM. CELIJE NA KOJU HL
3603      120   ! POKAZUJE DO PRVOG 0 BAJTA.
3603 7E    130  LD A,(HL)
3604 A7    140  AND A      ! DA LI JE NULA?
3605 2806  150  JR Z,IGRA
3607 CD3E36 160 CALL ISPIS
360A 23    170  INC HL
360B 18F6  180  JR PISI
360D       190 IGRA         ! POCETAK PARTIJE
360D 010000 200  LD BC,0    !  LOW = 0
3610 110004 210  LD DE,1024 ! HIGH = 1024
3613       220 PROBAJ
3613 60    230  LD H,B
3614 69    240  LD L,C
3615 19    250  ADD HL,DE  ! HL = HIGH-LOW
3616 CB3C  260  SRL H
3618 CB1D  270  RR L        ! HL = HL/2
361A CD4836 280 CALL ISPISHL
361D 3E0D  290  LD A,&0D    ! NOVI RED
361F CD3E36 300 CALL ISPIS
3622       310 UZMI
3622 CD5236 320 CALL ULAZ
3625 FE4D  330  CP "M"
3627 2811  340  JR Z,MANJI
3629 FE56  350  CP "V"
362B 2809  360  JR Z,VECI
362D FE50  370  CP "P"
362F 20F1  380  JR NZ,UZMI ! NEISPRAVNO SLOVO
3631       390  ! BROJ JE POGODJEN!
3631 215C36 400 LD HL,TEKST1
3634 18CD  410  JR PISI
3636       420 VECI
3636 44    430  LD B,H      ! LOW  = HL
3637 4D    440  LD C,L
3638 18D9  450  JR PROBAJ
363A       460 MANJI
363A 54    470  LD D,H      ! HIGH = HL
363B 5D    480  LD E,L
363C 18D5  490  JR PROBAJ
363E       500 ISPIS
363E       510  ! ISPISIVANJE ZNAKA CIJI
```

naredba prenosi osmobitni broj iz akumulatora u memorijsku ćeliju čija je adresa (redni broj) 50 000". Kao i ranije, izražavali smo se manje precizno da bismo uštedeli nekoliko reči.

Memorijski registri i memorijske ćelije

Memorija se sastoji od ćelija u koje staje po osam bita i koje su numerisane po svojim rednim brojevima. Pretpostavimo da našem računaru treba da naredimo da u ćeliju čiji je broj 100 upiše zbir sadržaja memorijskih ćelija čije su adrese 101 i 102. Bilo bi logično da se za takvu naredbu upotrebi oznaka:

ADD (101), (102), (100)

ADD je skraćenica za naredbu „saberi" i prate je tri adrese: adresa memorijske ćelije u kojoj je prvi sabirak, adresa drugog sabirka i adresa ćelije u koju treba staviti rezultat (smisao zagrada ćemo razumeti docnije). Računar koji bi mogao da izvrši ovakvu naredbu bi se nazivao *troadresni računar.* Pokazalo se, međutim, da bi navođenje velikog broja adresa iza svake instrukcije predstavljalo neracionalan utrošak memorije, pa su se pojavili dvoadresni i jednoadresni računari. Jasno je da jedna adresa nije dovoljna da potpuno odredi sabiranje koje smo upravo naznačili. Zato će program ekvivalentan gornjoj naredbi na jednoadresnom računaru glasiti:

```
LOAD A, (101)  ; Prenesi u akumulator sadržaj ćelije 101.
ADD A, (102)   ; Dodaj na sadržaj akumulatora sadržaj ćelije 102;
               ; rezultat sabiranja smesti u akumulator.
LOAD (100),A   ; Prenesi sadržaj akumulatora u ćeliju 100.
```

Vidimo da umesto jedne naredve sada moramo da iskoristimo tri, ali da zato svaku od njih prati samo po jedna adresa. Uz adresu se, međutim, stalno pominje nekakvo slovo A, koje predstavlja ime jednog od registara procesora.

Operativna memorija se obično sastoji od velikog broja (npr. šezdesetak hiljada) ćelija. Osim nje, mikroprocesor ima i nekoliko (ili nekoliko desetina) bajtova svoje privatne memorije koji su grupisani u registre; flip-flopovi koji sačinjavaju te registre se nalaze unutar samoga mikroprocesora. Z80 ima četrnaest registara opšte i osam registara specijalne namene, kao i nekoliko internih registara koji se koriste za obavljanje nekih specijalnih operacije i koji su programeru nedostupni. Vidljivi registri se obeležavaju slovima A, B, C, D, E, H, L, F, I, SP, PC, IX, IY i R. Registar A nazivamo akumulatorom, jer se u njemu akumuliraju rezultati aritmetičkih operacija, dok ćemo imena i namene ostalih registara objasniti kada za to bude došlo vreme. Osmobitni registri A-F se, pod određenim uslovima, mogu kombinovati u šesnaestobitne, pa ćemo tako nailaziti na oznaku HL koja označava operaciju nad sadržajima registara H i L posmatranim zajedno.

Neke ćudne zastavice

Sledeći termin koji treba da upoznamo je **flag** (ako baš želite da koristite prevode, upotrebite „našu" reč indikator). Pretpostavljajući da ste razumeli šta je registar, reći ćemo da je flag jednobitni registar. Ovo, bar na prvi pogled, dolazi u kontradikciju sa našom ranijom tvrdnjom da mikroprocesor može da pristupa samo čitavim bajtovima memorije, ali je ta kontradikcija samo prividna: flagovi nisu deo memorije već deo samog procesora i imaju veliku ulogu za njegov normalan rad. Pretpostavimo da treba u ćeliju 100 upisati broj 255 ukoliko se u toj ćeliji ranije nalazila nula ili broj nula ukoliko se u toj ćeliji ranije nalazilo bilo šta drugo. Posmatrajmo program koji bi mogao da obavi tu operaciju:

```
LOAD A, (100)  ; Stari sadržaj ćelije 100 u akumulator.
COMPARE A,0    ; Uporedi sadržaj akumulatora sa nulom.
JUMP EQ,NULA   ; Ukoliko je jednak (EQ), idi do mesta
               ; u programu koje se zove 'NULA'.
LOAD A,0       ; Broj nula u akumulator . . .
LOAD (100), A  ; . . . a zatim i u ćeliju 100.
HALT           ; Kraj rada.

* NULA         ; Ovde se dolazi ako je sadržaj ćelije 100
               ; na početku bio 0.
LOAD A,255     ; Broj 255 u akumulator . . .
LOAD (100),A   ; . . . a zatim u ćeliju 100.
HALT           ; Kraj rada.
```

Vidimo da je instrukcija COMPARE A,0 bila iskorišćena da se testira sadržaj akumulatora, a zatim smo upotrebili naredbu JUMP EQ, NULA koja je predstavljala skok samo ako je odgovor na pitanje koje je postavila prethodna naredba bio potvrdan. Kako bi naredba JUMP „znala" kakav je rezultat izvršavanja prethodne naredbe? Jedino tako što je naredba COMPARE A,0 postavila neki flag mikroprocesora u stanje 0 ili 1. Taj flag će biti u stanju jedan ako je odgovor na pitanje „da li

je sadržaj akumulatra jednak nuli" potvrdan, a nula ukoliko je odrečan. Sada se naredba JUMP EQ, NULA može protumačiti rečima „idi na deo programa označen sa NULA ukoliko je flag jednak jedinici (ili, kako se to obično kaže, ukoliko je flag *setovan),* a produži sa izvršavanjem programa ako nije".

Z80 ima više flagova kojima su dodeljena specijalna značenja. O njima ćemo, jasno, opširno govoriti docnije; za sada je bitno da zamislimo flag kao zastavicu koja se podiže ako je odgovor na neko pitanje potvrdan, a spušta ako je odrečan; stanje te zastavice docnije možemo da testiramo i da menjamo tok izvršavanja programa u zavisnosti od njega. Svi flegovi su obično fiktivno povezani u jedan registar koji označavamo slovom F. Smisao ovoga grupisanja je mogućnost čuvanja vrednosti flegova na steku koji predstavlja sledeću oblast našeg interesovanja.

Privremena memorija

Obzirom da svaki mikroprocesor ima relativno malo registara, često će nam biti potrebno da neke od njih privremeno oslobodimo za obavljanje neke operacije da bismo, po njenom završetku, restaurirali njihov raniji sadržaj. U takvoj situaciji bismo mogli da se poslužimo idejom koja je izložena u programu:

```
LOAD (20000), A; Premešta se sadržaj akumulatora u mem. ćeliju 20000
...            ; Deo programa koji koristi akumulator.
LOAD A, (20000); Obnavljanje ranijeg sadržaja akumulatora.
```

Nevolja kod ovoga rešenja je što smo morali da „pamtimo" da je podatak koji smo sačuvali baš u ćeliji čija je adresa 20000; potrošili smo, osim toga, celih 6 bajtova za smeštanje LOAD instrukcija. Rešenje istog problema uz upotrebu steka bi glasilo:

```
PUSH A         ; Registar A se čuva na steku.
...            ; Deo programa koji koristi akumulator.
PULL A         ; Obnavlja sadržaj akumulatora.
```

Sadržaj registra A je, i u ovoj varijanti, prepisivan u neku memorijsku ćeliju, ali sada mi nismo morali da razmišljamo o njenoj adresi: mikroprocesor je opremljen registrom SP *(Stack pointer* ili *pokazivač steka)* koji sadrži adresu memorijske ćelije koju treba popuniti brojem koji se šalje na stek; sadržaj ovog registra se automatski povećava ili smanjuje po svakoj operaciji sa stekom, tako da primenom PULL instrukcije uvek dobijamo podatak koji smo „zapamtili" primenom prethodne PUSH naredbe. Stek se, osim toga, intenzivno koristi za realizaciju potprograma, i prekida i posvetićemo mu dužnu pažnju u jednom od sledećih poglavlja.

Predstavljanje brojeva

U uvodnom poglavlju smo sa naslovne strane ovih „Računara" prepisali cenu od 100101100 dinara, ali nismo objasnili način na koji smo došli do ovog niza brojeva; da vam je prodavac na kiosku pomenuo cenu od 100101100 dinara, našli biste se u velikom čudu! Pre nego što se pozabavimo ovim binarnim brojem, moraćemo da posvetimo pažnju običnim, dekadnim brojevima kojima se svakodnevno služimo.

Binarni i heksadekadni brojevi

Posmatrajmo, na primer, broj 1986. On je iskazan u pozicionom dekadnom sistemu. Ovo *dekadnom* potiče od toga što je baza sistema broj 10, što znači da oznaka 1986 predstavlja zbir:

$$1986 = 6*10^0 + 8*10^1 + 9*10^2 + 1*10^3$$
$$= 6*1 + 8*10 + 9*100 + 1*1000$$
$$= 6 + 80 + 900 + 1000$$

Što se broja 100101100 tiče, on je iskazan u pozicionom binarnom sistemu. Ovo *binarnom* potiče od toga što je baza sistema broj 2, što znači da oznaka 100101100 predstavlja zbir:

$$100101100$$
$$= 0*2^0 + 0*2^1 + 1*2^2 + 1*2^3 + 0*2^4 + 1*2^5 + 0*2^6 + 0*2^7 + 1*2^8$$
$$= 0*1 + 0*2 + 1*4 + 1*8 + 0*16 + 1*32 + 0*64 + 0*128 + 1*256$$
$$= 0 + 0 + 4 + 8 + 0 + 32 + 0 + 0 + 256$$
$$= 300$$

Vidimo da bionarni broj 100101100 i dekadni broj 300 predstavljaju istu stvar izraženu na dva različita načina. Prvi način predstavljanja je pogodniji za računar, a drugi za čoveka i postoje

U programu primećujemo jednu interesantnu konstrukciju koja se praksi često koristi: LD A,H: DR L. Ove dve naredbe testiraju vrednost registra HL i, u slučaju da je ona jednaka nuli, setuju Z fleg. Potpunije poređenje šesnaestobitnih registara HL i DE može da se obavi naredbama:

```
LD    A,H
CP    D
RET   NZ
LD    A,L
cP    E
RET
```

Kada izvršite ovaj potprogram, Z fleg će biti setovan ako je HL=DE. Ukoliko su ova dva registra različita, C fleg će reći koji je veći.

Primer 3: ASCII niz u heksadekadni broj

```
3600          20   ORG &3600
3600          30   OPT 7
3600          40 ASCHL
3600 210000   50   LD HL,0
3603          60 VRTI
3603 CD0E36   70   CALL CIFRA   ! SLEDECA CIFRA
3606          80   !            U A ILI KRAJ.
3606 29       90   ADD HL,HL
3607 29       100  ADD HL,HL
3608 29       110  ADD HL,HL
3609 29       120  ADD HL,HL   ! HL=16*HL...
360A 85       130  ADD A,L
360B 6F       140  LD L,A      ! ... + CIFRA.
360C 18F5     150  JR VRTI
360E          160 CIFRA
360E          170  ! PREPOZNAVANJE HEX CIFRE
360E          180  ! NA KOJU POKAZUJE DE.
360E 1A       190  LD A,(DE)
360F FE2F     200  CP "0"-1
3611 FA2B36   210  JP M,KRAJ
3614 D630     220  SUB "0"
3616 FE0A     230  CP 10
3618 F21D36   240  JP P,SLOVO
361B          250  ! PRONADJENA CIFRA 0-9:
361B          260  ! DE POKAZUJE NA SLEDECU.
361B 13       270  INC DE
361C C9       280  RET
361D          290 SLOVO
361D D607     300  SUB 7
361F FE0A     310  CP 10
3621 FA2B36   320  JP M,KRAJ
3624 FE10     330  CP 16
3626 F22B36   340  JP P,KRAJ
3629          350  ! PRONADJENO SLOVO A-F;
3629          360  ! DE POKAZUJE NA SLEDECE.
3629 13       370  INC DE
362A C9       380  RET
362B          390 KRAJ
362B          400  ! NAIDJENO NA ZNAK KOJI
362B          410  ! NIJE HEX CIFRA.
362B          420  ! TREBA IZBRISATI ADRESU
362B          430  ! POVRATKA SA STEKA I
362B          440  ! VRATITI SE U PROGRAM
362B          450  ! KOJI JE POZVAO ASCHL.
362B F1       460  POP AF
362C C9       470  RET
362D          480  >
```

Kada poželimo da pišemo mašinske programe koji se neće oslanjati na ulazne i izlazne naredbe bejzika, zatrebaće nam mnogo različitih konverzija. Često je, na primer, potrebno konvertovati niz ASCII znakova koji reprezentuju heksadekadni broj u vrednost pogodnu za računanje. Jedan od jednostavnijih (i ne previše racionalnih) načina da se ovaj problem reši pokazuje i naša slika.

Pre pozivanja potprograma ASC—HL treba u registar DE upisati adresu memorijske ćelije koja sadrži prvi znak ASCII niza. Program će konvertovati znakove sve dok ne naiđe na nešto što nije cifra niti veliko slovo A—F; kada će u HL registru biti vraćen odgovarajući broj. Da bi se očuvala jednostavnost, nije predviđeno kontrolisanje prekoračenja: ukoliko zahtevate od računara da konvertuje niz znakova 'ABCDEF' u HL registru će se naći broj &CDEF.

Primetimo da glavni program poziva potprogram CIFRA koji, kada naiđe na kraj ASCII niza, ne vraća kontrolu glavnom programu: naredbom POP AF je „zaboravljena" adresa povratka pa sledeće RET izaziva povratak u program koji je pozvao ASC—HL. Ovakvo rešenje nije struktuirano i može da izazove zabunu kod nekoga ko čita program, ali smo ga iskoristili da ilustrujemo tehniku koju smo u ranijem tekstu pominjali. Ljubiteljima struktuiranog programiranja namenjujemo Primer 4.

Dokažite da ste razumeli ovaj program pišući odgovarajuću rutinu koja u HL dovodi vrednost niza ASCII znakova koji su shvaćeni kao ceo dekadni broj. Trebaće vam jedino rutina koja množi HL sa 10 poput sledeće:

```
ADD   HL,HL
PUSH  HL
ADD   HL,HL
ADD   HL,HL
POP   DE
ADD   HL,DE
```

Primer 4: Broj u ASCII niz

Kao što nam je potrebno da pretvorimo niz znakova u broj tako će nam zatrebati da pretvorimo broj (upisan u HL) u niz ASCII znakova koje ćemo, na primer, ispisati na ekran. Ovu konverziju možete da poverite programu HL—ASC koji je dat na slici. Problem je rešavan „sa vrha na dole": najpre smo šesnaestobitni broj pretvorili u dva osmobitna i pozvali potprograme koji ih ispisuju. Ti su potprogrami pretvorili osmobitne brojeve u četvorobitne (četiri bita se ponekad zovu *nibl)* i za svaku od ovakvih grupa pozvali potprogram NIBL koji je na neki način inverzan potprogramu CIFRA iz prethodnog primera.

Obratimo pažnju na redove 290 i 300 koji glase: CALL NIBL:RET. Bolji poznavaoci Z80 bi primetili da se u ova dva reda mogu uštedeti dva bajta: CALL NIBL stavlja na stek adresu sledeće naredbe (RET) a zatim započinje izvršavanje potprograma. Kada se taj potprogram izvrši, njegovo će RET izazvati povratak na adresu koja je upisana na vrh steka tj. na naredbu RET koja će sličnu

Sve u svemu, posle CPIR indikator nule će biti setovan ako je podatak pronađen, a adresa tog podatka će biti upisana u HL. Ukoliko je BC došao do nule, a podatak nije pronađen, Z fleg će biti resetovan. Umesto CPIR možemo da koristimo CPI, pri čemu smo, kao i ranije, sami odgovorni za realizaciju ciklusa korišćenjem PV flega.

Postojanje naredbe CPIR ne ukida potrebu za CPDR (ComPare, Decrement&Repeat) i CPD: ponekad će nam biti neophodno da pretražujemo blok memorije počevši od njegovog kraja. Neka, na primer, želimo da pronađemo broj &0D u bloku memorije koji se završava na &3000 i pri čemu smo procenili da traženje u memorijskim ćelijama čije su adrese manje od &2000 nema smisla. Napisaćemo:

```
        LD      A,&0D
        LD      HL,&3000
        LD      BC,&1000
        CPDR
        JR      Z,found
        ...
```

Blokovi registara

Razmena sadržaja osnovnih i alternativnih registara spada, strogo uzevši, u „operacije sa blokovima", pa ćemo je još jednom objasniti. Prisećamo se da Z80, osim registarskih parova AF, HL, DE i BC ima i alternativne registarske parove AF', HL', DE' i BC'. Sa EXX vrlo brzo razmenjujemo HL sa HL'. BC sa BC' i DE sa DE' dok sa EX AF,AF' razmenjujemo sadržaje AF i AF'. Iako ovo izgleda sasvim jednostavno, nije lako organizovati program tako da se ovakvim razmenama nešto dobije: jeste da ste sa EXX sačuvali tri registarska para, ali nemate mogućnosti da dalje koristite njihove vrednosti (posle PUSH AF akumulator će, na primer, i dalje čuvati stari sadržaj koji je samo **prekopiran** na stek). U sledećem poglavlju ćemo videti da alternativni registri svoju pravu vrednost dobijaju tek ako je potrebno veoma brzo opsluživanje interapta.

Pomenimo instrukcije EX (SP),HL, EX (SP),IX i EX (SP),IY. Njihovo dejstvo je za nekoga ko je pročitao dosadašnji tekst prilično jasno: broj sa vrha steka se **razmenjuje** sa brojem koji je upisan u registar HL, IX odnosno IY.

Biblioteka programa

Primer 1: Program koji radi bilo šta

```
3100                    ORG     &3100
3100 3A0039             LD      A,(&3900)
3103 3C                 INC     A
3104 320039             LD      (&3900),A
3107 C9                 RET


10 INPUT "Unesi broj"; N
20 POKE &3900,N
30 A=USR (&3100)
40 PRINT "Broj+1= "; PEEK (&3100)
50 GOTO 10
```

Počećemo od sasvim jednostavnog primera: želimo da računar izvrši sasvim kratak mašinski program koji radi bilo šta — u našem slučaju uvećava vrednost neke memorijske ćelije za 1. Sa slike vidimo da bejzik program sa tastature uzima neki broj a onda ga, primenom naredbe POKE, upisuje u memorijsku ćeliju čija je adresa &3900. Zatim se, primenom naredbe USR, poziva mašinski program koji je smešten počevši od adrese &3100 (možete ga, jasno, smestiti i na mnoga druga mesta) i, na kraju, ispisuje nova vrednost bajta &3900. Isprobajte ovaj program — pomoći će vam da bolje upoznate vaš asembler i njegove veze sa bejzikom.

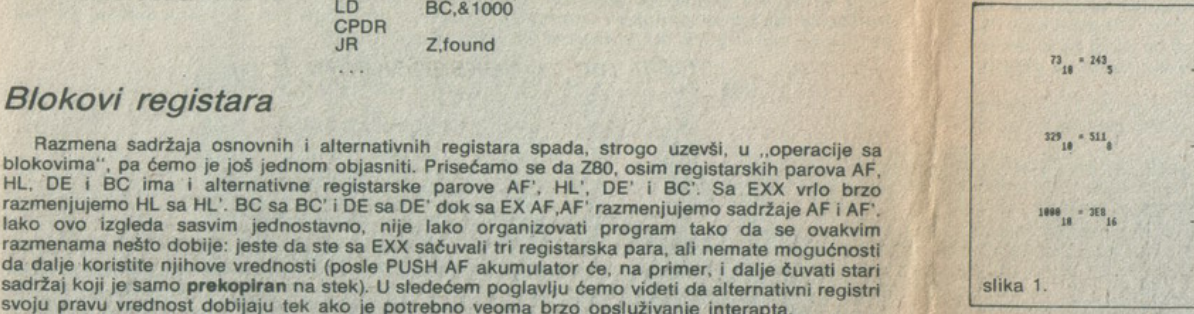
Primer 2: vremenska petlja

```
3100                    ORG     &3100
3100 0605       VREME   LD      B,5             ; ili nesto manje
3102 21FFFF     PETLJA1 LD      HL,&FFFF        ; ili nesto manje
3105 2B         PETLJA2 DEC     HL
3106 7C                 LD      A,H
3107 B5                 OR      L               ; HL=0 ?
3108 20FB               JR      NZ,PETLJA2
310A 10F6               DJNZ    PETLJA1
310C C9                 RET
```

Primer 2 predstavlja vremensku petlju — ako vam se ponekad učini da vaš mašinski program radi prebrzo, usporite ga ovakvim dodatkom. Trajanje vremenske petlje zavisi od početne vrednosti registra B — ukoliko u njega upišete neki veliki broj (na primer &FF), petlja će potrajati desetak minuta. Upisivanje jedinice u B će, sa druge strane, izazvati čekanje od svega par sekundi. Ukoliko želite još kraće petlje, postavljajte brojeve manje od &FFFF u HL.

postupci kojima se brojevi iz jednog sistema prevode u drugi. Postupak prevođenja binarnih brojeva u dekadne smo već upoznali, ali na pitanje *kako je dobijena brojka 100101100* još nismo odgovorili!

Podelimo, najpre, broj 300 sa dva. Dobićemo količnik 150 i ostatak **0**. Zatim delimo 150 sa 2 i dobijamo 75 i ostatak **0**. Sledeće deljenje daje 37 i ostatak **1**, zatim 18 i ostatak **1**, onda 9 i ostatak **0**, 4 i ostatak **1**, 2 i ostatak **0**, 1 i ostatak **0** i, nazad, pri deljenju jedinice sa 2 dobijamo količnik nula i ostak **1**. Prikupimo sve ostatke i napišimo ih obrnutim redom, počevši od poslednjeg. Dobijamo broj **100101100**, tj. predstavu broja 300 u binarnom sistemu. Radi vežbe možete da pogledate sliku 1 na kojoj je tabelarno prikazana ova i nekoliko drugih konverzija dekadnih brojeva u druge brojne sisteme.



slika 1.

Dekadno	Binarno	Heksadekadno
0	0000	& 0
1	0001	& 1
2	0010	& 2
3	0011	& 3
4	0100	& 4
5	0101	& 5
6	0110	& 6
7	0111	& 7
8	1000	& 8
9	1001	& 9
10	1010	& A
11	1011	& B
12	1100	& C
13	1101	& D
14	1110	& E
15	1111	& F

slika 2.

U poslednjem primeru sa slike 1 smo konvertovali broj 1000 u tzv. heksadekadni sistem, tj. brojni sistem čija je osnova 16. Dobili smo ostatke 3, 14 i 8, što znači da bi rezultujući broj trebalo napisati kao 3148. Ovo, naravno, ne bi valjalo, jer broj 3148 može da se sastavi i od ostataka 3, 1, 4 i 8. Zato bismo ovaj broj morali da napišemo kao *(3)(14)(8)* ili, ako želimo da uštedimo malo pisanja, kao *3E8*. Odakle se pojavilo slovo E? Za pisanje binarnih brojeva su dovoljne cifre 0 i 1, što znači da bi se neki vanzemaljac koji potiče iz kompjuterskog sveta u kome se operiše samo sa binarima veoma začudio kada bi video naše cifre 2, 3, . . . 8, 9. Za heksadekadne brojeve treba imati 16 cifara, tj. šest više nego za dekadne. Mogli smo, naravno, da smislimo neke sasvim nove znakove koji bi predstavljali ove cifre, ali bi takav postupak zahtevao predizajniranje tastatura i generatora karaktera naših računara. Daleko je jednostavnije zameniti cifre 10, 11, 12, 13, 14, i 15 slovima A, B, C, D i E prema tabeli sa slike 2.

Na slici 2 primećujemo i jednu interesantnu osobinu heksadekadnih brojeva. Posmatrajmo, na primer, broj 1000 napisan binarno i heksadekadno:

```
1000=0011 1110 1000
   = 3    E    8
```

Vidimo da, dok za pretvaranje dekadnih u binarne brojeve treba primenjivati relativno komplikovan algoritam, konverzija binarnih u heksadekadne brojeve i obratno predstavlja jednostavan postupak koji se brzo nauči napamet. To je upravo i razlog za uvođenje heksadekadnih brojeva — oni su izvanredna zamena za binarne (koji su, kao što rekosmo, jedini prihvatljivi za računar), a nisu predugački da bi ih imalo smisla pisati i pamtiti (mnogo je lakše zapamtiti da se bejzik program kod „galaksije" smešta u memoriju počevši od adrese 203A, nego pamtiti broj 0011110000111010!).

Ostalo je da razrešimo još jednu dilemu: kako ćemo prepoznati u kom je sistemu predstavljen neki broj. Binarni brojevi su obično toliko dugački da njihovo prepoznavanje ne predstavlja problem; nevolja je što, naišavši na broj 3000, ne znamo da li se radi o dekadnoj ili o heksadekadnoj konstanti. Matematičari su uobičajili da se iza svakog broja piše oznaka brojnog sistema kao indeks npr. 3000_{16} ili 3000_{10}. Ovakav način pisanja je za kompjuterske primene sasvim nepogodan, jer računari ispisuju slova u precizno formatiranim redovima. Zilog je zato uveo konvenciju da se iza broja ne piše nikakvo slovo ako se radi o dekadnoj konstanti, da se piše slovo B ako se radi o binarnom broju, a slovo H ako se radi o heksadekadnom; tako bismo mogli da napišemo 3E8H, 1000 ili 1111101000B. Iako je ovakav način označavanja pogodan za programiranje u mašinskom jeziku, autori bejzik interpretatora se trude da ga izbegnu jer može da dovede do zabune: da li, na primer, protumačiti naredbu:

```
IF A=2000H=0 kao
IF A=2000 THEN H=0 ili kao sintaksnu grešku:
IF A=2000 H THEN=0?
```

Kako se kod mnogih kućnih računara paralelno koristi bejzik i mašinski jezik, neprijatno je zahtevati od korisnika da u bejziku koristi jedne a u mašinskom (zapravo asemblerskom, o razlikama između ova dva pojma govorimo u poglavlju 4) druge oznake, prihvaćeno je da se ispred dekadnih brojeva ne stavlja nikakav prefiks, da se ispred heksadekadnih stavlja oznaka '&' a ispred

binarnih '%'. Na taj način bismo cenu ovih „Računara" mogli da napišemo kao 300, &120 ili %100101100 dinara. U „spektrumovim" asemblerima kao što je *Devpac* je uobičajeno da heksadekadnim brojevima prethodi oznaka '#' (number sign ili, u domaćem hakerskom žargonu, „taraba"). Obzirom da ova oznaka u mnemonici 6502 i nekih drugih mikroprocesora označava neposredno adresiranje, za nas je bilo apsolutno nemoguće da prihvatimo ovakvu konvenciju bez obzira na problem koji ona može da izazove kod dela čitalaca.

Operacije sa binarnim brojevima

U osnovnoj školi se uči sabiranje i oduzimanje dekadnih brojeva, pa je neophodno da se u ovoj osnovnoj školi mašinskog programiranja posvetimo operacijama sa binarnim veličinama. U osnovi, ako znate da sabirate dekadne brojeve, znate da sabirate i binarne, heksadekadne ili brojeve u bilo kom drugom sistemu. Činjenica da nešto znate, naravno, još ne znači da to radite tačno i brzo; ukoliko želite da steknete malo rutine (koja, inače, nije previše nužna za pisanje mašinskih programa), moraćete da posvetite određeno vreme uvežbavanju.

Izaberite neka dva relativno mala dekadna broja i saberite ih. Ako su to brojevi 7 i 11 dobićete zbir 18. Sada pretvorite 7 u binarni broj 111 i 11 u binarni broj 1011. Potpišite ova dva binarna broja i pokušajte da ih saberete.

Jedan i jedan su 2, ali se to dva piše kao 10 što znači da nulu pišete a jedan pamtite. Taj jedan što ste pamtili plus i iz broja 111 plus 1 iz 1011 daju 3 odnosno 11; pišete 1 i prenosite 1. Taj jedan plus 1 iz 111 plus 0 iz 1011 daju 2 tj. 10; pišete nulu i pamtite 1. Saberete, najzad, to jedan sa jedan iz 1011 i dobijate 2 tj. 10 koje napišete; rezultat je 10010. Pretvorite ovaj rezultat u dekadni broj:

$10010 = 0*2^0 + 1*2^1 + 0*2^2 + 0*2^3 + 1*2^4 = 18$

Rezultat je 18 što je, kao što ste verovatno i očekivali, isto što i 7+11. Na vama je da isprobate ono što ste upravo naučili na drugim primerima, što je, naravno, ujedno i lepa škola pretvaranja binarnih brojeva u dekadne i obratne operacije.

Oduzimanje biste mogli da obavljate na sličan način; umesto prenosa koristili biste „pozajmicu". Želimo, međutim, da ukažemo na jedan drugi način oduzimanja koji je daleko bliži mikroprocesorima sa kojima radimo. Poznato nam je da je 12−7 ISTO ŠTO I 12+(−7) što znači da se oduzimanje svodi na sabiranje sa negativnim brojem. Došlo je, dakle, vreme da upoznamo negativne binarne brojeve.

Broj −111 je, naravno, isto što i −7 ali je −111 broj vrlo nepogodan za pamćenje u memoriji računara: osim jedinica i nula u njemu se pojavljuje i neka crtica koja predstavlja negativan predznak. Ako pretpostavimo da se svaki broj upisuje u jedan bajt memorije, mogli bismo da se dogovora da prvi (ili poslednji ako, kao što je uobičajeno, bitove brojite sdesna) bit tog bajta u sebi kodira znak tako da će se broj 7 pamtiti kao 00000111 a broj −7 kao 10000111. Ovakvo predstavljanje brojeva se, međutim, pokazalo nepogodnim za mnoge primene, pa je izmišljeno nešto bolje — pamćenje komplementa broja. *Prvim komplementom* broja 7 (00000111) nazivamo binarni broj koji dobijamo kada umesto svake nule u tom broju stavimo jedinicu a umestu svake jedinice nulu. 00000111 tako postaje 11111000. Dodajući jedan na ovaj broj dobijamo 11111001 što je *drugi* ili *potpuni komplement* broja 7 i, ujedno, zgodan način da se predstavi broj −7.

Pokušajmo sada da izračunamo koliko je 12−7=12+(−7). Broj 12 binarno predstavljamo kao 00001100, a broj −7 kao 11111001. Sabiranjem ova dva broja dobijamo 100000101. Vidimo da rezultat ima 9 bita; kako u memorijsku ćeliju može da se smesti najviše 8, deveti bit će biti odbačen tako da će rezultat biti 00000101 ili, dekadno posmatrano, 5 što je, kao nekom čarolijom, isto što i 12−7. Iz ovog primera i svih primera koje ćete sami zamisliti da se zaključiti da je predstavljanje negativnih brojeva pomoću potpunog komplementa celishodno rešenje koje svodi oduzimanje na sabiranje sa negativnim brojem.

U prethodnom primeru smo, međutim, mirne duše odbacili poslednji (rekli smo da se bitovi broje sa desne strane) bit rezultata i dobili korektno rešenje; može se pretpostaviti da bi kod nekog drugog primera takvo nonšalantno ponašanje dovelo do greške. Da stvar bude još gora, greška je moguća čak i kada nema nikakvog prekoračenja. Da smo, na primer, pokušali da saberemo 121 i 93, tj. 01111001 i 01011101, dobili bismo:

```
  01111001
+ 01011101
----------
  11010110
```

Obzirom da je poslednji bit rezultata jedan, radi se o negativnom broju koji, pretvoren u dekadni, daje 11010110=−00101010=−42. Jasno je da −42 nikako ne može da bude rezultat sabiranje brojeva 121 i 93. Da nam se u budućnosti ne bi događale ovakve greške, naučićemo tri pravila:

1. U memorijsku ćeliju mogu da se smeštaju označeni ili neoznačeni brojevi. U jedan bajt (osam bita) mogu da se smeste označeni brojevi između −128 i +127 *ili* neoznačeni brojevi između 0 i 255.

ABRAKADABRA upotrebićemo čarobnu reč LDIR (LoaD, Increment&Repeat) i — ništa više! Nema labele, nema ciklusa, nema IF-a, nema potrebe za flegovima . . .

Kako radi instrukcija LDIR; Mogli biste da je zamenite sa:

```
cikl  LD    (DE),(HL)    ;  kada bi ovako nešto postojalo
      INC   HL
      INC   DE
      DEC   BC
      CP    BC,0         ;  kada bi ovakvo nešto postojalo
      JR    NZ, cikl
```

Već vas čujem kako kažete: baš je lepo iskoristiti neku instrukciju a onda u komentaru napisati 'kada bi ovakva instrukcija postojala'. Pokušajmo, vežbe radi, da napišemo isti program uz korišćenje isključivo postojećih instrukcija (zamena LDIR neće biti baš ekvivalentna, jer ova instrukcija utiče na PV, N i H flegove, ali koga za njih briga?)

```
      PUSH  AF           ;  čuva A i flegove
cikl  LD    A,(HL)       ;  sledi zamena za LD (DE),(HL)
      LD    (DE),A
      INC   HL           ;  modifikacija brojača
      INC   DE
      DEC   BC
      LD    A,B          ;  simulacija CP BC,0. Zašto?
      OR    C
      JR    NZ,cikl
      POP   AF
```

Upoznajmo sada instrukciju LDI koja je, kao što možete da pretpostavite, ekvivalentna sa LDIR ali ne izaziva ponavljanje operacije: najpre se izvrši nešto poput LD (DE),(HL), povećaju se HL i DE za po jedan a BC se umanji za 1. Tada se BC uporedi sa nulom pa se rezultat smesti u Parrity/Overiflow fleg (???) i to tako da je fleg resetovan ukoliko je BC postalo nula a setovan u svim ostalim slučajevima. Obzirom da se LDI retko koristi, nećemo joj posvećivati više pažnje.

Dejstvo instrukcije LDDR ćemo upoznati posmatrajući njenu simulaciju uz obavezno korišćenje nepostojećih instrukcija (tako je zabavnije, čini vam se da LDDR za nešto i služi!):

```
cikl  LD    (DE),(HL)
      DEC   HL
      DEC   DE
      DEC   BC
      CP    BC,0
      JR    NZ,cikl
```

Jedina razlika između LDDR i LDIR je što se primenom prve blok kopira od kraja ka početku, a kod druge od početka ka kraju. Da bismo, dakle, rešili problem sa početka ovoga poglavlja koristeći LDDR, morali bismo da napišemo.

```
      LD    HL,&30FF
      LD    DE,&32FF
      LD    BC,&100
      LDDR
```

Uz LDDR ćemo, jasno, pomenuti LDD, instrukciju koja bi trebala da vam bude jasna ako ste shvatili razliku između LDI i LDIR.

Zbog čega su konstruktori Z80 obezbedili LDD i LDDR kada su već postojale LDI i LDIR? Najsigurniji ćete odgovor dobiti ako ih pitate; ukoliko vam se ne putuje do Amerike, poverujte u naše nagađanje: ove instrukcije verovatno ne zauzimaju mnogo prostora u mikrokodu a, znate, procesori se reklamiraju i navođenjem broja instrukcija koje prepoznaju!

Pretraživanje blokova

LDI, LDIR, LDD i LDDR su instrukcije za pomeranje blokova memorije pa nam je ostalo da upoznamo instrukcije koje omogućavaju njeno pretraživanje. Cilj pretraživanja je, jasno, pronaći neki broj u jednoj od sukcesivnih memorijskih lokacija. Taj broj se smešta u akumulator, dok se u HL upisuje adresa početka bloka memorije koji se pretražuje. U BC treba upisati maksimalni broj bajtova koji se pretražuju; ako ste sasvim (ali baš sasvim) sigurni da će podatak pre ili kasnije biti pronađen, možete da napišete LD BC,&FFFF ili, uz malo smisla za humor, LD BC,0. Iza svega toga pišete instrukciju CPIR (ComPare, Increment&Repeat) koja bi mogla da se napiše i kao:

```
cikl  CP    A,(HL)
      JR    Z,izlaz
      INC   HL
      DEC   BC
      CP    BC,0
      JR    NZ,cikl
      RES   Z            ;  zar ne bi bilo lepo imati instrukciju
                         ;  koja resetuje indikator nule
```

adresu instrukcije na koju će se preći po nailasku na RET. U principu bismo mogli da napišemo LD HL,&3000: PUSH HL, a zatim RET: mikroprocesor bi se „vratio" na instrukciju koja je upisana u memorijsku ćeliju &3000 bez obzira na činjenicu da prethodno nije izvršeno nikakvo CALL; tako smo, premda to nema mnogo smisla, simulirali JP &3000!

Daleko je problematičnija situacija u kojoj smo u potprogramu koji je pozvan sa CALL nehotice upotrebili neko PUSH koje nije imalo svoje POP. Naišavši na RET, mikroprocesor slepo uzima dva bajta sa steka i smešta ih u PC, ne znajući da se ne radi ni o kakvoj povratnoj adresi već o podatku koji ima neki drugi smisao. Izvršavanje programa će se, u tom slučaju, nastaviti od nekog nepredviđenog mesta, što može da ima katastrofalne posledice po program u memoriji. Sve u svemu, **ako se planira povratak iz potprograma,** upotreba steka mora da bude „čista": koliko podataka stavimo na stek toliko moramo i da uzmemo i tako u svakoj grani potprograma!

Zašto smo rekli „ako se planira povratak"? Kada u bejziku izvršimo GOSUB, **moramo** da izvršimo i RETURN, inače se (nama nevidljiv) stek za pozive potprograma polako puni, pa će jednom biti prijavljena greška tipa 'Too many GOSUBs'. Ne bi imalo nikakvog smisla učiti mašinski jezik ako bi i on imao slična ograničenja: ako u nekoj grani potprograma konstatujemo da se nema smisla vraćati u glavni program, jednostavno POP AF će skloniti sa steka svaki trag da je neki potprogram uopšte pozivan. Novi sadržaj akumulatora i fleg registra u tom slučaju treba ignorisati.

Često je potrebno da se potprogramu prenesu neke vrednosti koje će obrađivati i da se u glavni program vrate rezultati te obrade. Argumente potprograma je najjednostavnije upisati u neke od registara; Z80 ih i onako ima dosta. Ukoliko je potrebno više argumenata, možete ih upisati u neke memorijske ćelije sa fiksnim adresama odakle će ih potprogram „pokupiti"; u tom slučaju potprogram **ne sme** da poziva samog sebe, ali su vam ovakve tzv. rekurzije za sada sasvim nepotrebne. Moguće je, najzad, smestiti sve argumente na stek, a onda izvršiti CALL. Potprogram tada najpre mora da prenese adresu povratka u neki od registara (npr. sa POP IX), zatim da obradi argumente postepeno ih skidajući sa steka, da smesti rezultate na stek i da konačno izvrši JP (IX). Sve u svemu, prilično komplikovano.

Specijalni slučaj instrukcije CALL je RST (ReSTart): dok CALL, zajedno sa adresom potprograma, zauzima uobičajena tri bajta, RST zajedno sa svim potrebnim podacima zauzima samo jedan. Kako je to moguće? Sa RST se može pozvati samo jedan od osam potprograma koji se nalaze na fiksnim memorijskim adresama &0, &8, &10, &18, &20, &28, &30, i &38. Sve se ove adrese nalaze u ROM-u i predstavljaju početke nekih vrlo važnih i često korišćenih potprograma. Umesto RST &10 možete, naravno, uvek da koristite i CALL &10; ukoliko vam utrošak dva bajta nije važan, instrukciju RST možete i da zaboravite.

U ROM-u vašeg računara se nalazi mnogo korisnih potprograma koji vas oslobađaju briga o pisanju po ekranu, skaniranju tastature, generisanju zvuka i slično: zar nije lakše napisati CALL KEY-SCAN i u akumulatoru dobiti kod pritisnutog tastera, nego ispitivati dirke jednu po jednu? Da biste, međutim, napisali CALL KEY-SCAN morate da znate gde se tačno u ROM-u nalazi rutina koja skanira tastaturu (ako je ona smeštena počevši od adrese &028F kao kod „spektruma", na početku vašeg programa ćete napisati KEY-SCAN EQU &028F), kako treba pripremiti njen poziv i gde su smešteni rezultati. Takve podatke možete da nađete u knjigama tipa „The Complete Spectrum ROM Disassembly". Nabavite, ako je to ikako moguće, knjigu u kojoj je opisan ROM vašeg računara i videćete da će ona postati najkoristnija referenca u praktičnom radu!

Operacije sa blokovima

Koristeći do sada naučene instrukcije ne bi vam bilo teško da, na primer, napišete program koji će prepisati segment memorije od &3000 do &30FF (zaključno) na memorijske adrese &3200—&32FF;

Program koji smo upravo napisali korektno rešava zadati problem i, kada ga poredimo sa odgovarajućom bejzik rutinom, predstavlja pravog Brzog Gonzalesa. Konstruktori Z80 su nam, međutim, omogućili da napišemo sasvim sličan program koji će se sastojati od samo četiri instrukcije, biti dosta brži i daleko jasniji na prvi pogled. Uvođenje novih (i u suštini nepotrebnih) instrukcija je, jasno, mač sa dve oštrice, jer je početniku često teško da odvoji bitno od nebitnog. Zato bismo vam savetovali da pri prvom čitanju ovog umetka jednostavno preskočite ostatak ovoga poglavlja zapamtivši jedino da postoje moćne instrukcije za pomeranje i pretraživanje blokova memorije; kada vaše poznavanje mašinskog jezika kroz praktičan rad bude uznapredovalo, neće biti kasno da naučite više o ovim „slatkišima".

Pomeranje blokova

Instrukcije za pomeranje memorijskih blokova koriste **sve** registare opšte namene osim, paradoksalno, akumulatora. Registar HL **uvek** sadrži adresu početka bloka, registar DE njegovo odredište ('destination') dok u BC treba upisati broj bajtova koji se prenose. U primeru koji smo dali na početku ovoga poglavlja podaci koji se premeštaju počinju od memorijske ćelije &3000, što znači da će naš mali program početi sa LD HL,&3000. Podaci se premeštaju u memorijski blok koji počinje od &3200: LD DE,&3200. Potrebno je premestiti &100 bajta dakle: LD BC,&100. Umesto

2. Pri sabiranju neoznačenih brojeva može da se pojavi prenos iz najvišeg razreda koji nazivamo *carry*. Pojava ovoga prenosa pri računanju implicira postojanje greške, tj. pokušaj da se saberu dva broja čiji je zbir veći od 255.

3. Pri sabiranju označenih brojeva prenos treba ignorisati. Moguće je, međutim, da se pojavi prekoračenje, tj. greška u računu koji nazivamo *overflow*. Mikroprocesori Z80 i 6502 omogućavaju programeru da proveri postojanje greške (koja čini rezultat u akumulatoru besmislenim) i preduzme neku akciju ukoliko je ona nastupila.

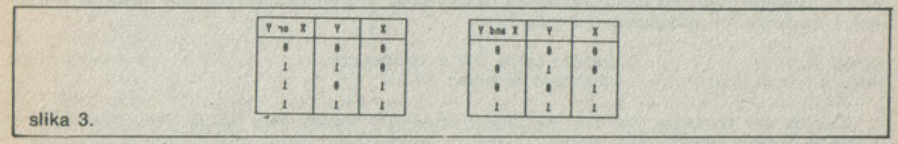
Binarno množenje se realizuje uz pomoć jedne nove operacije koju nazivamo *šiftovanje* i običnog sabiranja. Kako ćemo se na problem šiftovanja vratiti već u sledećem poglavlju, množenju sada nećemo posvećivati posebnu pažnju. Što se deljenja tiče, možete ga obavljati slično deljenju običnih brojeva.

Logičke operacije

Programeri koji svoja remek-dela isključivo na višim programskim jezicima možda smatraju da su sabiranje, oduzimanje, množenje, deljenje i, eventualno, stepenovanje jedine operacije koje računar može da obavi. Ukoliko se, međutim, upoznate sa nekim ko u životu piše jedino mašinske programe, verovatno ćete saznati da računari umeju da sabiraju (eventualno i da oduzimaju), da se sa množenjem slabo snalaze, da im deljenje predstavlja veliki problem, ali da su im operacije čudnih imena *konjunkcija, disjunkcija, negacija* i *šiftovanje* najjača strana.

Logičko "I"

Konjunkcija je latinski naziv za logičku 'i' funkciju koja je definisana sledećom tablicom:



X	Y	X i Y
0	0	0
1	0	0
0	1	0
1	1	1

X	Y	X ili Y
0	0	0
1	0	1
0	1	1
1	1	1

slika 3.

Pogledajmo smisao ove tablice na jednom primeru: neka je u memorijskoj ćeliji čiji je redni broj 100 upisana konstanta 120 (ili %1111000), a u ćeliju 121 konstanta 72 (%1001000). Naš računar izvršava program poput sledećeg:

```
LOAD A, (100)  ; U akumulator se dovodi broj %1111000.
AND A, (101)   ; Sadržaj akumulatora .AND. sadržaj ćelije 101;
               ; rezultat se smešta u akumulator.
LOAD (102), A  ; Vrednost akumulatora se premešta u ćeliju 102.
HALD           ; Kraj rada
```

Pitanje je, naravno, koji broj dolazi u memorijsku ćeliju 102. Potpišimo brojeve %1111000 i %100100 jedan ispod drugog i primenimo na njih tabelu sa slike 3, gledajući nezavisno svaki njihov bit. Dobijamo:

```
  1111000
AND 0100100
-----------
  0100000=32
```

Primećujete li vrlo bitnu razliku između sabiranja ili bilo koje aritmetičke operacije koju ste do sada poznavali i operacije 'i'? Kada sabirate višecifreni broj, to radite zdesna nalevo, vodeći računa o eventualnom postojanju prenosa u viši razred. Kod logičkog 'i' gledate svaki par potpisanih bitova bez obzira na poziciju u broju, pa biste operaciju mogli da obavite i sleva nadesno, pa čak i, ukoliko vam je takvo nešto potrebno, počevši iz sredine broja.

Uz prethodni primer i još nekoliko koje sami izmislite (pokažite, na primer, da je 12 AND 28=12 ili da je &7F+&80=%0) razumećete pravila uz pomoć kojih se obavlja logička operacija 'i', ali nećete shvatiti njenu svrhu. Pretpostavimo, na primer, da je u memorijskoj ćeliji čija je adresa 100 upisan neki broj i da nam je iz nekog razloga potrebno da sedmi (poslednji) bit tog broja postavimo na nulu, a da ostale bitove ne promenimo. Već nam je poznato da mikroprocesor može da pročita i u memoriju upiše samo čitav bajt što znači da bi, bez primene logičkih operacija, manipulacija pojedinim bitovima bila praktično nemoguća. Uz poznavanje funkcija 'i' napisaćemo jednostavan program:

```
LOAD A,(100)      ; Dovođenje sadržaja ćelije 100 u akumulator.
AND A,%1111111    ; Resetovanje sedmog bita.
LOAD (100),A      ; Smeštanje rezultata u memoriju.
```

Pre nego što vidimo kako radi ovaj program, primetimo razliku između njega i prethodnog primera u kome je druga linija glasila AND A,(101). Namerno smo istakli zagrade koje se ne pojavljuju u primeru koga upravo posmatramo. Dok AND a,(101) znači *primeni logičku operaciju AND ('i) na sadržaj akumulatora i* ***sadržaj memorijske ćelije*** *čiji je broj 100* dotle AND A,%01111111 znači *primeni logičku operaciju AND ('i') na sadržaj akumulatora i* ***broj %01111111.*** Zagrade, dakle, označavaju da je u okviru naredbe navedena adresa memorijske ćelije u kojoj se nalazi traženi podatak, dok njihovo izostavljanje označava da je u okviru naredbe naveden baš taj podatak.

Pretpostavimo da se u ćeliji 100 nalazio broj &9A, odnosno %10011010. On je najpre doveden u akumulator, a zatim je na njega i broj %01111111 primenjena operacija 'i'. Da vidimo šta se dobija:

```
  10011010
AND 01111111
------------
  00011010
```

Vidimo da se rezultat razlikuje od početnog podatka &9A samo u poslednjem bitu koji je, posle primene AND operacije, postao nula. Ovakav rezultat se mogao i očekivati: broj %01111111 se sastoji od samih jedinica od kojih svaka, kada je logički pomnožimo (operacija 'i' se zove i logičko množenje) sa bilo čim drugim daje kao rezultat to „nešto drugo". Jedino vodeća nula, kada je logički pomnožimo sa bilo čim, daje kao rezultat opet logičku nulu!

Osim postavljanja bitova na nulu, operacija AND ima još jednu značajnu ulogu koju nazivamo maskiranjem. Pretpostavimo da smo memorijsku ćeliju 100 namenili za smeštanje nekih podataka koji imaju samo po dva bita. To znači da desni („nulti") bit te ćelije kao i bit do njega („prvi bit") predstavljaju jedan podatak, sledeća dva bita drugi i tako dalje:

```
DDCCBBAA
```

Da bi ovakva koncepcija uopšte imala smisla, moramo da omogućimo čitanje i promenu bilo kog od podataka AA, BB, CC i DD. Što se postavljanja nekih bitova na nulu tiče, nema problema. Kako, međutim, da pročitamo podatak BB?

```
LOAD A,(100)        ; Sadržaj ćelije 100 u akumulator.
AND A,%00001100     ; Maskiraju se bitovi 2 i 3.
```

Ako se pre izvršenja ove dve naredbe u ćeloji 100 nalazio broj %01011001, posle njihove primene ćemo u akumulatoru dobiti:

```
  01011001
AND 00001100
------------
  00001000
```

Primenom takozvanog „maskiranja" smo, dakle, uspeli da izdvojimo tražena dva bita iz jednobajtnog broja i dovedemo ih u akumulator. Oni se, na žalost, ne nalaze baš na pravom mestu (odgovaralo bi nam da se u akumulatoru nalazi broj 0000010) ali ih tamo možemo dovesti primenom šiftovanja koje ćemo za trenutak upoznati. Imamo, međutim, još jedan problem: za sada nemamo načina da postavimo neki bit (ili neke bitove) u stanje logičke jedinice. Za to će nam poslužiti disjunkcija tj. logička operacija 'ili' koju često nazivamo *logičko sabiranje* i obeležavamo sa OR.

Logičko "ILI"

Na slici 4 je data tablica istinitosti disjunkcije koja je, kao što vidimo, sušta suprotnost konjunkciji koju smo već upoznali. Pogledajmo kako operacija 'ili' deluje na primeru:

```
  1111000
OR 0100100
-----------
  1111100=124
```

Par strana unazad smo, uzgred budi rečeno, videli da se primenom logičkog 'i' na ista dva broja dobija 32. Operacija 'i' daje logičku jedinicu u rezultatu samo na mestima gde se **u oba** operanda javlja logička jedinica, dok 'ili' daje jedinicu svude gde se **u bilo kom** operandu nalazi jedinica (pročitajte prethodnu rečenicu još par puta). Kao takva, operacija 'ili' je vrlo zgodna za postavljanje bitova u stanje 1.

Govoreći o operaciji 'i', napisali smo mali program koji postavlja poslednji bit memorijske ćelije 100 na nulu, pa ne bi bilo loše da ga dopunimo programom koji isti bit postavlja na jedinicu:

```
LOAD A,(100)      ; Dovođenje sadržaja ćelije 100 u akumulator.
OR   A,%10000000  ; Setovanje sedmog bita.
LOAD (100),A      ; Smeštanje rezultata u memoriju.
```


smeštajući nulu (sadržaj akumulatora) u memorijsku ćeliju koju pokazuje HL i umanjujući sadržaj ovog registra za 1 kako bi pokazivao sledeću ćeliju koju treba brisati. Sledi naredba DJNZ PETLJA koja, pri svakoom izvršavanju, umanjuje sadržaj B za jedan a onda izaziva skok na labelu PETLJA **samo ako je** sadržaj B različit od nule. Kada čitav segment memorije bude obrisan, B će doći do nule i DJNZ će biti preskočeno; mikroprocesor nailazi na RET i vraća kontrolu bejziku. Kako je instrukcija DJNZ predviđena za realizovanje ciklusa, ona uvek izaziva skok unazad, što znači da ciklusi mogu da budu dugi do 255 bajta, dok bi korišćenjem DEC B: JR NZ, CIKLUS morali da budu duplo kraći.

Potprogrami i stek

Došlo je vreme da se podrobnije upoznamo sa stekom, koji smo već više puta pominjali. Stek je, rekli smo, područje memorije koje se koristi za privremeno odlaganje podataka koji će u bliskoj budućnosti biti potrebni. Na stek, primenom instrukcije PUSH, možemo da smestimo bilo koji od **registarskih parova** AF, DE, HL i BC, kao i specijalne registre IX i IY. Pogledajmo, najpre, kako radi PUSH HL:

```
DEC     SP
LD      (SP),H
DEC     SP
LD      (SP),L
```

Ovaj program ne bi radio kada biste ga otkucali, jer Z80 ne poznaje instrukciju LD (SP), reg. I pored toga, on rečito pokazuje sve što se dešava: šesnaestobitni registar biva upisan u memorijske lokacije (SP-1) i (SP-2), dok sam SP biva umanjen za dva, tako da pokazuje na poslednji bajt na steku. Kada nam, nekoliko instrukcija docnije, bude potrebno da vratimo podatak sa steka u HL, izvršićemo POP HL odnosno:

```
LD      L,(SP)
INC     SP
LD      H,(SP)
INC     SP
```

Šta se dešava ako napišemo PUSH HL: PUSH DE? Na stek, najpre, biva smešten registarski par HL, a onda DE, što znači da za vraćanje podataka u registre treba koristiti POP DE: POP HL **a ne** POP HL: POP DE. Zbog toga se stek naziva i LIFO strukturom (LIFO=last in, first out): poslednji podatak koji stavimo na stek će biti prvi koji ćemo, primenom POP instrukcije, pročitati.

Jasno je da, izvršavajući POP DE, mikroprocesor ne može da zna da li su dva bajta koja pronađe na steku nekada bila u registru DE; oni su se tamo mogli naći i posle PUSH HL. To znači da kombinaciju PUSH HL: POP DE možemo da koristimo umesto LD DE, HL, instrukcije koju Z80 ne podržava. Nešto je, međutim, brže napisati LD D,H: LD E,L sa istim efektom.

Kada u bejziku napišete GOSUB 1000, računar će početi da izvršava potprogram koji počinje od linije 1000. Kada u daljem radu naiđe na RETURN, vratiće se izvršavanju segmenta programa koji se nalazi iza GOSUB 1000. Da bi to uradio, on na neki način mora da „zna" da je do linije 1000 stigao uz jedan poziv potprograma (inače će RETURN izazvati grešku 'RETURN without GOSUB') i da „zapamti" gde se nalazila naredba koja je pozvala potprogram. Ovi podaci se zaista i pamte, ali su za prosečnog programera nebitne njihove lokacije.

Z80 je opremljen instrukcijom CALL koja je potpuni ekvivalent GOSUB; razlika je jedino u tome što se iza GOSUB piše broj programske linije, a iza CALL memorijska adresa ćelije u koju je upisana prva instrukcija potprograma (uz korišćenje asemblera će se, jasno, iza CALL naći ime labele). Za povratak u glavni program koristi se RET, potpuni ekvivalent RETURN.

Da bi stvar bila još lepša, konstruktori Z80 su obezbedili uslovni CALL: CALL Z, PROBA će izvršiti potprogram označen labelom PROBA samo ako je Z(ero) fleg setovan, dok će se ako nije izvršiti sledeća instrukcija u programu. Umesto Z možete da koristite NZ, C, NC, PO, PE, P ili M, baš kao i kod uslovne JP instrukcije. Za povratak iz potprograma možete da koristite RET ali i RET C, RET NC, RET Z itd: ukoliko je uslov ispunjen, kontrola se vraća glavnom programu, a ako nije nastavlja se sa izvršavanjem naredbe iza RET (uslov).

Da biste znali sve o potprogramima, reći ćemo da nema prepreke da u toku izvršavanja nekog potprograma pozovete novi potprogram, a iz ovoga sledeći i tako do prilične dubine.

Kada bismo ovime zaključili poglavlje o potprogramima, znali biste koliko i bejzik programer koji mirno koristi GOSUB . . . RETURN ne razmišljajući o tome kako ova struktura funkcioniše. Mašinski programer **mora da zna više,** pa ćemo pokušati da prikažemo delovanje instrukcija CALL i RET.

Kada naiđe na RET, Z80 mora da zna gde da se vrati (ne bi bilo loše kada bi znao i da li je potprogram stvarno pozvan ili se do RET-a stiglo greškom, ali to nije moguće — mikroprocesor kao što je Z80 i onako ne bi mogao da prijavi (RETURN without GOSUB'). Pošto se povratak vrši na instrukciju koja je upisana u neku memorijsku ćeliju, pri nailasku na RET su potrebna samo dva bajta: adresa te ćelije. Gde bi ta dva bajta mogla da budu smeštena? Pa, jedini ozbiljan kandidat može da bude stek.

CALL nnmm je, otprilike, ekvivalentno sa PUSH PC: JP nnmm, a RET sa POP PC (jasno je da instrukcije PUSH PC i POP PC pod tim imenima ne postoje). Pri svakom pozivu potprograma na stek, dakle, bivaju smeštena dva bajta koja, posmatrana kao šesnaestovni broj, predstavljaju

Slično tome, C predstavlja situaciju u kojoj je indikator prenosa setovan, a NC situaciju u kojoj je restovan, P označava pozitivan a M negativan znak broja (N fleg resetovan odnosno setovan) dok PE i PO označava setovan odnosno resetovan indikator parnosti/prekoračenja. Za početak će nam, kao što vidimo na slici 13, biti dovoljni Z i N flegovi: program koji dajemo, naime, u memorijsku lokaciju REZ upisuje jedinicu ako je broj u ćeliji POD pozitivan, nulu ako je jednak nuli, a 255 ukoliko se radi o negativnom broju (niste zaboravili da mikroprocesor, kada se radi o jednobajtnim rečima, smatra brojeve 0—127 pozitivnim, a 128—255 negativnim uz korišćenje potpunog komplementa?).

```
3000                    ORG     &3000
3100            POD     EQU     &3100
3101            REZ     EQU     &3101
3000 3A0031             LD      A,(POD)
3003 FE00               CP      0
3005 2004               JR      NZ,NIJE_NULA
3007 320131             LD      (REZ),A
300A C9                 RET
300B            NIJE_NULA
300B F21430             JR      P,POZITIVAN
300E 3EFF               LD      A,255
3010 320131             LD      (REZ),A
3013 C9                 RET
3014            POZITIVAN
3014 3E01               LD      A,1
3016 320131             LD      (REZ),A
3019 C9                 RET
```

slika 13

Relativni skokovi

Govoreći o raznim vrstama adresiranja, naučili smo da Z80 podržava relativno adresiranje samo kod nekih instrukcija uslovnog i bezuslovnog skoka. Šta beše relativno adresiranje? Ponovo ćemo ga upoznati posmatrajući instrukciju JR PC+&20 (Jump Relative &20 bytes). Pretpostavimo da se ova instrukcija nalazila u memorijskim ćelijama &1000 i &1001 (videćemo da instrukcije relativnog skoka, za razliku od instrukcija apsolutnog, zauzimaju samo dva bajta memorije). Mikroprocesor je najpre iz memorije uzeo kod instrukcije JR (&18) i prepoznao ga, a zatim pročitao adresu &20. U tom trenutku registar PC pokazuje na memorijsku ćeliju &1002 u koju je upisana naredba koja bi bila izvršena da tok programa nije nasilno promenjen naredbom bezuslovnog skoka. Mikroprocesor će sabrati sadržaj registra PC (&1002) i broj &20, pa će tako dobijenu vrednost staviti u PC; izvršavanje programa će tako biti nastavljeno od instrukcije čija je adresa &1020.

Čemu bi mogla da posluži ova komplikacija? Program koji u celini napišemo korišćenjem isključivo relativnog adresiranja će korektno raditi ma u koji ga segment memorije učitali: JR PC+&20 izaziva preskakanje sledećih &20 bajta, bez obzira gde ova instrukcija bila smeštena. Treba, međutim, da kažemo da je skoro nemoguće a i sasvim nepotrebno pisati program koji radi gde god da ga smestite — daleko je lakše izmeniti ORG i asemblirati program na drugo mesto. Relativni skokovi se, zato, koriste isključivo radi uštede memorijskog prostora kod kratkih skokova koji su inače daleko češći od dugih: JP &2020 zauzima tri, a JR PC+&20 samo dva bajta. Za šta je onda nekome potrebno apsolutno adresiranje? Korišćenjem jednog jedinog bajta za „pamćenje" rastojanja omogućeni su samo skokovi dugi do 127 bajta (broj iz JR se posmatra kao označen što znači da su mogući i skokovi unazad) dok iza JP možemo da navedemo bilo koju adresu u čitavoj memorijskoj mapi.

Većina asemblera tretira relativne skokove kao i apsolutne: pisaćete JR DALJE baš kao što biste pisali i JP DALJE. Sam asembler će pronaći labelu DALJE i izračunati njeno rastojanje od lokacije instrukcije JR formirajući tako korektan adresni deo ove naredbe. Preporučili bismo vam da se ugledate na vaš asembler i ne pravite razlike između JP i JR: uvek koristite relativne skokove, a apsolutnim pribegnite samo ako vaš asembler prijavi grešku tipa JUMP TOO LONG.

Konstruktori Z80 su omogućili i relativne uslove skokova JR C i JR Z, odnosno JR NC i JR NZ. Što se indirektnog adresiranja tiče, instrukcije JP (HL), JP (IX) i JP (IY) omogućavaju skok na instrukciju koja je upisana u memorijsku ćeliju čiji je broj upisan u HL, IX odnosno IY. LD HL,&3000: JP (HL) je, na primer, ekvivalentno sa JP &3000.

Posebna instrukcija relativnog skoka je DJNZ, neka vrsta ekvivalenta FOR . . . NEXT petlje u bejziku. Nju ćemo upoznati posmatrajući primer na slici 14.

```
3100                    ORG     &3100
3100 219F30             LD      HL,&30A0-1
3103 3EA0               LD      A,&A0
3105 47                 LD      B,A
3106 3E00               LD      A,0
3108 77         PETLJA  LD      (HL),A
3109 2B                 DEC     HL
310A 10FC               DJNZ    PETLJA
310C C9                 RET
```

slika 14

Dati program upisuje nule u memorijske ćelije čije su adrese &3000, &3001, . . . &30A0. U početku upisujemo &30A0 u registar HL koji će uvek pokazivati na memorijsku ćeliju u koju treba upisati nulu. U registar B (deo registra BC koji, kao što znamo, predstavlja Byte Counter, tj. brojač bajtova) upisujemo &A0, ukupan broj nemorijskih ćelija koje treba izbrisati. Zatim ulazimo u petlju

Uzevši da se pre izvršavanja programa u ćeliji 100 nalazio broj &73 odnosno %01110011, novi sadržaj je:

```
  01110011
OR 10000000
----------
  11110011
```

Videvši da smo, jednostavno OR, setovali sedmi (broji se od nule, sećate se?) bit nekog bajta, dovoljno smo sposobni da sastavimo program koji će postaviti sadržaj BB bajta u DDCCBBAA na vrednost koju želimo. Pretpostavimo da se ta vrednost nalazi u akumulatoru i to na mestima 2 i 3 (kada upoznamo šiftovanje, moći ćemo tu i da je dovedemo!). Da promenimo vrednost tražena dva bita prvo ćemo ih postaviti na nulu (kako bismo uništili eventualnu raniju vrednost) a zatim ih, primenom operacije OR, postaviti na novu vrednost.

```
LOAD (101),A        ; Vrednost akumulatora privremeno smeštamo u 101.
LOAD A,(100)        ; Sadržaj ćelije 100 u akumulator.
AND A,%11110011     ; Resetuju se bitovi 2 i 3 (postavljaju na 0).
OR A,(101)          ; Bitovi se postavljaju u traženo stanje.
LD A,(100)          ; Rezultat u memoriju.
HALT
```

Neka je staro stanje ćelije 100 bilo %11010101 i neka se u akumulatoru nalazio broj %00001000 koji, prema našoj konvenciji, treba da promeni stanje ćelije 100 u &11011001. Proverimo da li će se to zaista i dogoditi:

```
  11010101
AND 11110011
----------
  11010001

  11010001
OR  00001000
----------
  11011001
```

X	not X
0	1
1	0

slika 5.

Osim konjunkcije i disjunkcije, koristićemo i logičku negaciju (NOT) čija je istinitosna tabela data na slici 5. Primena ove operacije je sasvim jednostavna: svaka jedinica se pretvara u nulu a svaka nula u jedinicu; NOT 11010101 je 00101010. Jedna od primena negacije je promena znaka: rekli smo da se *prvi komplement* broja dobija kada se svaka jedinica u tom broju pretvori u nulu, a svaka nula u jedinicu; sada znamo da se ovakvo „pretvaranje" naziva *negacijom broja.* Ukoliko nam je potreban *drugi* ili *potpuni komplement* broja, rezultatu negacije ćemo prosto dodati jedan.

```
not (not X) = X

X and 1 = X                 X and (Y or  Z) = (X and Y) or  (X and Z)
X and 0 = 0                 X or  (Y and Z) = (X or  Y) and (X or Z)
X or  0 = X
X or  1 = 1                 not (X and Y) = not X or  not Y
                            not (X or  Y) = not X and not Y
```

slika 6.

Na slici 6 je dato nekoliko formula koje povezuju konjunkciju, disjunkciju i negaciju. Nema naročite potrebe da se dublje upuštate u ove formule ni da ih pamtite, ali nije nemoguće da će vam neka od njih zatrebati u docnijem radu pa zato treba zapamtiti da one postoje. Ukoliko ste se, čitajući ovo, zainteresovali za matematičku logiku, preporučujemo vam neki srednjoškolski ili univerzitetski udžbenik iz koga ćete saznati mnogo više o Bulovoj algebri, tablicama istinitosti, valjanim formulama i sličnim zavrzlamama.

Pomeranje bitova

Posmatrajmo brojeve 115, 230 i 204 i pokušajmo da uočimo neku vezu između njih. Na prvi, drugi i treći pogled veze nema. Ukoliko imate malo smisla za enigmatiku, setićete se da govorimo o logičkim operacijama sa binarnim brojevima, pa ćete naše brojeve predstaviti u tom obliku i dobiti %01110011, %11100110 i %11001100. Veza postaje očigledna: drugi broj je dobijen kada smo odbacili prvu cifru prvog i dodali mu nulu na kraj; treći broj je dobijen kada smo isto to uradili sa drugim. Sledeći broj u nizu bi, dakle, bio %10011000 ili 152.

Tvrdnju *sledeći broj se dobija kada prethodnom dopišemo nulu na kraj i odbacimo njegovu prvu cifru* možemo da iskažemo i na drugi način: *sledeći broj se dobija kada sve cifre prethodnog pomerimo ulevo za jedno mesto, a na upražnjeno mesto poslednje cifre upišemo nulu.* Ovo je ujedno i najjednostavnija definicija šiftovanja ulevo.

Šiftovanje udesno je vrlo slična operacija, pri čemu bi se od broja %01110011 dobio najpre broj %00111001, a zatim broj %00011100. Šta, međutim, treba da se dobije šiftovanjem broja %11001000 udesno za jedno mesto? Bez mnogo razmišljanja bismo rekli da je rezultat %01100100 i tako pokazali da smo zaboravili ono o čemu je u prethodnom poglavlju bilo govora!

Pretvorimo broj 8 u binarni broj %00001000 i šiftujemo ga za jedno mesto udesno. Dobijemo %00000100 ili dekadno 4. Ako ponovimo operaciju dobićemo dekadno 2 i, pri novom ponavljanju, jedinicu. Vidimo da se šiftovanje udesno svodi na deljenje broja sa dva; slično tome bismo mogli da pokažemo da se šiftovanjem ulevo broj množi sa dva. Pošto smo ovo razumeli, vratimo se prethodnom primeru: broj %11001000 pretvoren u dekadni sistem daje 200, dok %01100100 predstavlja broj 100 ili polovinu broja 200. Za sada je sve u redu. Međutim, rekli smo da se u memorijske ćelije često smeštaju **označeni brojevi,** pri čemu se negativni brojevi pamte korišćenjem potpunog komplementa. Obzirom da je vodeći bit broja %11001000 jedinica, radi se o negativnom broju. Dobijamo da je %11001000=−%00111000=−56. S druge strane, vodeći bit broja %01100100 je nula, što znači da se radi o pozitivnom broju 100. Kako polovina od −56 nikako ne može da bude +100, operacija šiftovanja označenog broja je dala pogrešan rezultat!

U čemu smo pogrešili? Samo u tome što smo, šiftujući broj udesno, na mesto vodećeg (sedmog) bita stavili nulu i tako pokvarili znak rezultata. Ispravno bi bilo da smo na to mesto ponovo upisali bivši vodeći bit čime se znak broja, koji taj bit opisuje, ne bi promenio. Tako bi se šiftovanjem broja %11001000 udesno dobio broj %11100100=−%00011100=−28, što je upravo polovina od −56. Ovakvo šiftovanje, kod koga se vodi računa o znaku broja, naziva se *aritmetičkim,* dok se šiftovanje kojim se od %11001000 dobija %01100100 naziva *logičkim.* Mikroprocesori mogu da izvrše bilo koje od ovih šiftovanja; prvo ćete koristiti kada u memorijskim ćelijama čuvate označene brojeve, tj. kada pišete neku vrstu matematičkog programa, a drugo kada operišete sa bitovima u cilju njihovog pakovanja ili testiranja.

Z80 omogućava relativno složena logička šiftovanja brojeva o kojima ćemo još dosta govoriti. Za sada je važno samo znati da se kod tih šiftovanja vodeći ili krajnji bit ne gubi, već se prenosi u jedan od flegova nazvan indikator prenosa. Postoji mogućnost i da se bit iz indikatora prenosa vrati u šiftovani broj, a takođe i da se „izgubljeni" bit vrati na kraj broja: tako bismo od %10010001 dobili %00100011. Ovakve operacije se iz očiglednih razloga naziva rotiranjem ulevo.

Instrukcije i njihovo kodiranje

Kada biste bilo koji od programa koje smo do sada pisali pokušali da „saopštite" mikroprocesoru koji upoznajemo, rezultati bi bili nikakvi. Pre svega, za zadavanje instrukcija smo koristili neke reči: LOAD, COMPARE, AND, HALT, ali smo prećutkivali način na koje bi one mogle da se unesu u računar. Možda ih otkucate umesto bejzik programa? Pokušajte i pozdraviće vas 'Syntax error'. Ili, pošto ste načuli da se mašinski programi unose primenom instrukcije POKE, da otkucate POKE LOAD? Sličan pozdrav. Da bi računar mogao da izvrši mašinski program, on mu mora biti saopšten u prihvatljivom, kodiranom obliku!

Mnemoničke skraćenice

Instrukcije se kodiraju u binarne brojeve, pri čemu kod svake instrukcije obično zauzima jedan bajt. Osim koda koji bi, na primer, zamenjivao skraćenicu LOAD, računaru treba da saopštimo i adresni deo naše instrukcije (sećate se onoga LOAD *A,%11001100?)* za koji se obično koriste sledeći, sukcesivni bajtovi memorije. Različite instrukcije, zajedno sa svojim adresnim delom, mogu da zauzmu jedan, dva ili više bajtova, pri čemu ovaj broj varira od instrukcije do instrukcije.

Kada se već instrukcije moraju pretvarati u brojeve, zašto izmišljate reči koje te brojeve zamenjuju? Zato što je mašinski jezik i onako po prirodi težak i nečitljiv, pa postojanje naziva instrukcija (tzv. *mnemoničkih skraćenica)* olakšava kako pisanje programa tako i njegovo docnije prepravljanje. Samu transformaciju reči u binarne brojeve je u stanju da obavi kompjuter pomoću programa koji nazivamo *asembler* i kome posvećujemo naše sledeće poglavlje. Mnemoničke skraćenice su stvar konvencije: možemo da se odlučimo da instrukciju čiji je kod &A9 zovemo LD ili LOAD ili samo L i bilo koja od tih varijanti će rezultirati istim mašinskim programima. Slično je i u bejziku: naredba PRINT bi sasvim lepo mogla da se zove WRITE, DISPLAY ili TYPE. Konvencije su, međutim, jednom usvojene i ne možemo da ih menjamo, pa se na njih treba privići. Posle određenog broja sati sedenja uz računar, naučićete sve bitne skraćenice napamet i kucati ih bez mnogo razmišljanja.

Korišćenje mnemoničkih skraćenica krije u sebi veliku opasnost: tendenciju da se uobrazi postojanje nekih naredbi. Ukoliko, na primer, znate da postoji naredba JUMP (HL), moguće je da po analogiji upotrebite i naredbu CALL (HL) koja, na primer, ne postoji! Program koji smo nazvali asemblerom će vas, jasno, upozoriti na grešku, ali vam to neće mnogo pomoći ako ste čitav segment programa zasnovali na nepostojećoj naredbi. Setovi instrukcija šesnaestobitnih i još moćnijih procesora su daleko bogatiji i njihovi su se konstruktori potrudili da postignu veliku simetričnost: svaka instrukcija ima sve adresne modalitete i instrukcije slične prirode su konsistentne (ovakav set instrukcija nazivamo *ortogonalnim).* Osmobitni mikroprocesori, međutim, zbog uštede memorije i jednostavnosti konstrukcije, imaju razne modalitete adresiranja, koji su pristupačni samo kod nekih instrukcija, i za njihovo korišćenje je potrebno, bar u početku, neprestano konsultovati tabelu naredbi, koja je data u našem prethodnom umetku. Korišćenje te tabele ćete upoznavati kako ovaj tekst bude napredovalo; za sada samo bacite jedan pogled na nju i, ukoliko za to imate uslova, napravite par fotokopija — te će fotokopije postati redovan gost na vašem pisaćem stolu. Videćete da su u tabelama pobrojane skraćenice instrukcija, kratak opis koja se zove SRA i čije je dejstvo prikazano u tabeli instrukcija. Postoje i više nego čudne naredbe RLD i RRD koje razmenjuju polovinu bitova akumulatora sa memorijom i koriste se za rad sa BCD brojevima o kojima, kao što rekosmo, ukratko govorimo tek u budućim napisima.

Bezuslovni i uslovni skokovi

Kada bejzik intepretator vašeg računara izvrši neku instrukciju, on automatski prelazi na izvršavanje sledeće i tako dalje — sve dok ne naiđe na neko STOP ili END. Korišćenjem GOTO naredbe se, naravno, može zahtevati da se ovaj tok izvršavanja izmeni, dok se IF . . . THEN . . . ELSE konstrukcijom postiže izvršavanje sledeće naredbe samo ako su neki uslovi ispunjeni. Bez ovakvih naredbi ne bismo mogli da sastavimo čak ni najjednostavnije programe, što znači da nešto odgovarajuće mora da bude raspoloživo i na mašinskom jeziku.

Bezuslovni skokovi

Najjednostavnija naredba bezuslovnog skoka je JP — potpuni ekvivalent GOTO. Iza JP se nalazi adresa memorijske ćelije u koju je upisana naredba koju sledeću treba izvršavati, što znači da JP &3000, jasno, podrazumeva da se instrukcija na &3000 zaista može izvršiti: kako mnoge instrukcije zauzimaju dva, tri ili čak četiri bajta, može da se dogodi da se u memorijskoj ćeliji &3000 ne nalazi početak neke instrukcije, već njen drugi ili treći bajt. Dok bejzik interpretator u konfliktnim situacijama prijavljuje grešku, mikroprocesor ne poseduje apsolutno nikakav način da prosudi da li se na &3000 nalazi instrukcija koju ste vi planirali za izvršavanje; on će pokušati da izvršava ono što tamo nađe što će dovesti do nepredvidljivog toka događaja i, moguće, potpunog blokiranja računara.

Iako Z80 razume jedino instrukcije tipa JP &3000, za nas je ovakvo korišćenje apsolutnog adresiranja sasvim nepogodno. Pre svega, retko su nam potrebni skokovi na ovako „okrugle" lokacije — obično programiramo prelazak na neku instrukciju u programu. Pri pisanju tog programa nam je, jasno, poznata njegova početna adresa u memoriji (pišemo je iza ORG) ali nam nije poznata početna adresa svake njegove instrukcije: da bismo je našli moramo da izračunamo koliko bajta memorije zauzima svaka njoj predhodna instrukcija pa da tako dobijene brojeve saberemo i dodamo na vrednost ORG. Takvo računanje i sabiranje je neprijatan posao podložan greškama koji, da stvar bude još gora, moramo da ponavljamo kada god u programu napravimo bilo kakvu izmenu koja će produžiti ili skratiti neki njegov segment. Zbog toga svi iole dobri asembleri omogućavaju rad sa *labelama.*

Reč labela ('label') može da se prevede kao 'oznaka' — radi se o nekom simbolu kojim označavamo određeni segment programa. Asembleri obično omogućavaju da ime labele bude bilo koja reč od 3—6 slova sa izuzetkom mnemoničkih skraćenica naredbi: ako iskoristimo labelu JP računar je neće razlikovati od instrukcije JumP! Jasno je da ćemo ime labele birati tako da asocira na svrhu segmenta programa koji označavamo: česta imena labela su RADNI, RAČUN, DALJE, GREŠKA, KRAJ i slično.

```
                3000                    ORG     &3000
                3000 210000             LD      HL,0
                3003 7E         CIKLUS  LD      A,(HL)
                3004 77                 LD      (HL),A
                3005 23                 INC     HL
                3006 C30330             JP      CIKLUS
```

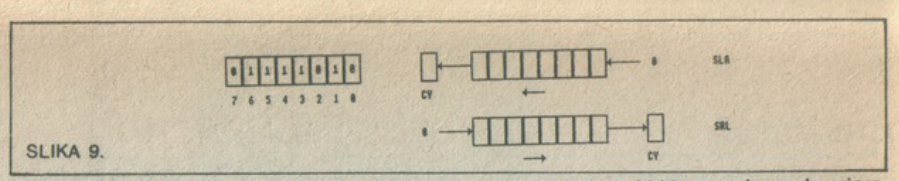
slika 12

Na slici 12 je prikazan program koji koristi naredbu JP da bi realizovao beskonačnu petlju: ovaj program upisuje u svaku memorijsku ćeliju njen prethodni sadržaj i tako u beskonačnost! Labela CIKLUS označava početak petlje, pa je JP CIKLUS ekvivalentno sa JP &3000 s tim što je JP CIKLUS daleko univerzalnije: jednostavnom promenom naredbe ORG ovaj program možemo da asembliramo u bilo koji segment memorije. Ukoliko se odlučite da otkucate i isprobate ovaj program, uzmite u obzir da će on eventualno izazvati krah sistema ukoliko vaš računar (što inače nije uobičajeno za Z80) koristi memorijski mapirane periferijske uređaje i da ćete moći da ga prekinete jedino pritiskom na RESET taster.

Labele, osim za skokove, možete da koristite i kao zamenu za brojeve. Na početku programa možete, na primer, da napišete RAMTOP EQU &5CB2, pa će docnije LD (RAMTOP), HL biti ekvivalentno sa LD (&5CB2), HL. U ovom slučaju korišćenje labela pomaže jedino boljoj čitljivosti programa jer ćete, kada ga docnije budete gledali, zaboraviti da je &5CB2 adresa sistemske promenljive RAMTOP pa ćete morati da konsultujete razne tabele da biste razumeli segment programa koji ste sami napisali.

Uslovni skokovi

Naredba JumP izaziva tzv. bezuslovni skok — naredba čija je adresa navedena iza JP se izvršava u svakom slučaju. U praksi ćemo daleko češće koristiti takozvane uslovne skokove, koji su omogućeni postojanjem flegova. Naredbe JP Z, JEDNAK će, na primer, izazvati skok na segment programa označen labelom JEDNAK **samo ako** je Z(ero) fleg setovan. Ukoliko želimo da se skok na labelu izvrši ukoliko je indikator nule resetovan, koristićemo JP NZ, JEDNAK (Jump if Not Zero).



SLIKA 9.

budemo želeli da zavisno od ulaznih veličina programa setujemo neki bit, moraćemo da pripremimo odgovarajuću masku pa da koristimo OR!

Ako imamo naredbu za setovanje nekog bita, verovatno postoji i naredba za njegovo resetovanje. Njen opšti oblik je RES b,t gde se o **b** i o t ne može reći ništa posebno novo. Ova naredba se, kao što smo videli u uvodnim poglavljima, može simulirati uz pomoć AND.

Testiranje bitova se obavlja primenom naredbe BIT b,t. Nju bismo mogli da prevedemo na srpskohrvatski kao *da li je bit broj b odredišta t setovan?* Ako je odgovor na ovo pitanje potvrdan, (bit je setovan) zero fleg će biti resetovan, a ako je odgovor odrečan (fleg je **re**setovan) zero fleg će biti setovan. Ovakva konvencija izgleda naopako, ali šta da se radi. Ukoliko, da rezimiramo, želimo da nastavimo program od mesta koje je označeno sa DALJE samo ukoliko je sedmi bit akumulatora setovan, izvršićemo:

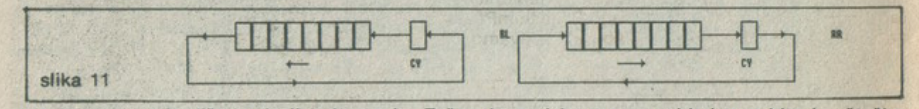
```
BIT 7,A          testiraj bit 7 akumulatora.
JP NZ, DALJE:    ukoliko zero fleg nije setovan,
                 bit 7 je setovan pa skoči na DALJE.
```

Šiftovanje i rotiranje

Govoreći o logičkim operacijama u uvodnim poglavljima, ukratko smo obradili šiftovanja i pomenuli razliku između logičkog i aritmetičkog šiftovanja i rotiranja. U tom vam je trenutku čitava stvar mogla izgledati komplikovano, ali je ona prava šala prema različitim vrstama šiftovanja i rotiranja koja Z80 podržava. Jasno je da svi tipovi šiftovanja nisu neophodni; oni su pripremljeni da vam olakšaju život. Dok ste, međutim, početnik mnogo različitih operacija vam umesto toga zagorčava život pa predlažemo da za početak upoznate samo najosnovnije. Kada vam neka od „specijalnih" operacija ovoga tipa bude zatrebala, lako ćete je razumeti.

Logičko šiftovanje ulevo i udesno je prikazano na slici 10. Vidimo da sa SLA t odnosno SRL t pomeramo bitove odredišta t na levo odnosno na desno za jedno mesto; umesto nultog (za SLA) odnosno sedmog bita (kod SRL) u odredište ulazi nula, dok bit koji „ispadne" usled pomeranja odlazi u indikator prenosa. Raniji sadržaj indikatora prenosa se, jednostavno, gubi.

Vidimo da smo primenom instrukcija SLA odnosno SRL izgubili bivše stanje indikatora



slika 11

prenosa a u odredište upisali jednu nulu. Zašto da ne izbegnemo gubitak tog bita i veštačko generisanje nule prepisavši prethodno CY na kraj odnosno početak odredišta? Tako dobijamo operacije RL t i RR t: rotaciju odredišta t i indikatora prenosa za jedno mesto ulevo, odnosno udesno. Obe ove operacije su prikazane na slici 11.

SLA, SRL, RL i RR su osnovna šiftovanja i rotacije; ostatak ovoga poglavlja slobodno možete da zanemarite. Uz određenu dozu hrabrosti možete, naravno, i da nastavite sa čitanjem.

Šta da radimo ako nam je potrebno da prosto rotiramo sadržaj nekog registra za jedno mesto ulevo? Možemo da izvršimo SCF, zatim RL t, a da onda, ako je indikator resetovan, izvršimo RES 0,t. Da vidimo kako to radi ako se u registru B nalazio broj %01010011. Posle SCF ovaj broj se nije promenio, ali je indikator prenosa setovan. Posle RL B vodeći bit broja (0) odlazi u CY, ostali bitovi se pomeraju za po jedno mesto ulevo, a bivše stanje CY flega (1) dolazi na kraj broja; dobili smo %10100111. Pošto je indikator prenosa setovan, izvršava se RES 0,B, pa %10100111 postaje %10100110; baš ono što smo želeli. Isti rezultat može, međutim, da se dobije i primenom naredbe RLC t (*rotate left circular* ili, u slobodnom prevodu uz malo pleonazama, *kružno rotiraj*). Dejstvo ove i odgovarajuće naredbe RRC t koja rotira odredište udesno je prikazano u tabeli instrukcija.

Kod šesnaestobitnih load instrukcija smo videli da su konstruktori Z80 ponekad bili raspoloženi da nam omoguuće štednju bajtova u programu, pripremajući specijalne slučajeve nekih instrukcija: postoji, na primer, naredba LD rp (nnmm), gde je **rp** bilo koji registarski par (pa i HL) a onda i instrukcija LD HL, (nnmm) koja zauzima bajt manje. Sve instrukcije za rotiranje i šiftovanje koje smo do sada upoznali rade sa bilo kojim registrom, memorijskom ćelijom na koju pokazuje HL ili neki od indeks registara, pa zauzimaju po dva, tri ili četiri bajta programa. Šiftovanja i rotiranja se, međutim, obično izvode nad sadržajem akumulatora, pa postoje jednobajtne naredbe RLCA, RLA, RRCA i RRA kao specijalni slučajevi RLC, RL, RRC i RR. Ukoliko, dakle, vašem asembleru kažete RR A, potrošićete dva bajta, dok ćete ako zaboravite blanko pa napišete RRA potrošiti samo jedan. Pa neka posle neko kaže da se ne isplati grešiti!

Z80 je opremljen naredbom za aritmetička šiftovanja udesno (deljenje označenog broja sa 2)

njihove funkcije, načini adresiranja koji ih prate, kodovi i podatak koji vas za sada neće interesovati: broj mašinskih ciklusa koji su potrebni da se instrukcija izvrši. Na osnovu ovog broja možete da izračunate koliko se vremena izvršava neka instrukcija: ako vaš računar radi na frekvenciji od 2 MHz, jedna mašinski ciklus traje 1/(2MHz)=0.5 mikrosekundi. Kada jednom budete programirali neki kontroler koji radi u realnom vremenu (npr. crta sliku na ekranu televizora), videćete da će vas vreme izvršavanja instrukcija naglo zainteresovati.

Memorija i adresiranje

Memoriju možete da zamislite kao niz ćelija sa slike 7. Svaka ćelija ima svoju adresu koja se kreće između 0 i 65535 (ili, ako više volite heksadekadne brojeve, između &0 i &FFFF). Svaka ćelija se sastoji od osam bitova koji su obeleženi brojevima 0—7.

65536 bajta iznosi 65536/1024=64 kilobajta i to je najveća memorija kojom može da bude opremljen računar za Z80, 6502 ili bilo kojim drugim osmobitnim mikroprocesorom (kod 16-bitnih mikroprocesora broj se penje do 16 Mb). Ovaj *adresni prostor* mora da bude podeljen na ROM u koji će biti smešten operativni sistem računara i interpreter (i) programskih jezika kao što je bejzik i RAM u koji će biti upisivani programi koje korisnik piše. Za sada ćemo reći da se ROM kod računara sa Z80 počinje od adrese &0000.

Većina instrukcija koje zadajemo mikroprocesoru se bavi manipulacijom (prenošenjem, sabiranjem, testiranjem . . .) brojeva koji su smešteni u neke memorijske ćelije, što znači da se uz kod instrukcije mora navesti i neka oznaka na osnovu koje će računar zaključiti gde se nalaze podaci koje treba da obrađuje. Ta oznaka se zove *adresni deo instrukcije.* U zavisnosti od načina na koji mikroprocesor transformiše adresni deo instrukcije u podatak sa kojim treba manipulisati, razlikujemo razne *modove adresiranja* od kojih ćemo neke upravo upoznati.

Implicitno adresiranje

Najjednostavniji slučaj adresiranja je izostavljanje adrese — adresa je tada implicirana smislom naredbe pa se ovakvo adresiranje naziva *impliciranim ('implied').* Posmatrajmo, na primer, naredbu SCF čiji je jedini smisao da setuje (postavi na jedinicu) indikator prenosa koji smo već pominjali. Iz koda ove naredbe se zna sve što je potrebno za njeno izvršavanje, pa bi bilo kakva adresa (npr. SSF carry) bila suvišna.

Registarsko adresiranje

Slično impliciranom je *registarsko adresiranje* koje podrazumeva da se argument nalazi u nekom od registara mikroprocesora. Instrukcija INC A, na primer, dodaje jedan sadržaju akumulatora.

Neposredno adresiranje

Malo je složeniji slučaj instrukcija LD A,25 koja u akumulator dovodi broj 25. Broj kojim treba manipuliati se ovde nalazi u okviru same instrukcije pa se tako adresiranje naziva *neposrednim ('immediate').* Instrukcije ovoga tipa ćemo koristiti kada želimo da dovodimo konstante u memoriju, smeštamo ih u registre ili na neki drugi način manipulišemo njima.

Apsolutno adresiranje

Došlo je, najzad, vreme da počnemo da radimo sa sadržajima memorijskih ćelija i upoznamo *apsolutno adresiranje:* posmatraćemo instrukciju LD A, (&1000) koja u akumulator dovodi osam bita ćelije čija je adresa &1000. Ova naredba je, u ovakvom obliku, sasvim jasna. No, šta da kažemo na LD HL, (&1000)? Iako još nismo govorili o šesnaestobitnim registrima, iz oznake ove instrukcije može da se pretpostavi da je **HL** naziv registra koji je nastao „slepljivanjem" registara H i L i koji, tako, ima 16 umesto 8 bita. Memorijska ćelija &1000 ima, kao što znamo, samo osam bita, što znači da bi pri doslovnom shvatanju navedene naredbe polovina registra HL ostala prazna. Konstruktori Z80 su odlučili da se ova instrukcija izvrši tako što se u HL smeste sadržaji memorijskih ćelija &1000 i &**1001** i to tako da memorijska ćelija &1000 ode u registar L a memorijska ćelija &1001 u registar H. Ako su, na primer, ćelije &1000 i &1001 sadržavale brojeve &20 i &3A, u registar HL će stići broj &3A20; primetimo da ćelija sa nižom adresom „čuva" manje značajan a ćelija sa višom adresom više značajan bajt šesnaestobitnog broja (znate li šta su „više" i „manje" značajne cifre? U broju 1234 cifra i označava hiljade i kao takva je daleko značajnija od cifre 4 koja označava jedinice: ako pri pisanju ovoga broja napišete 2 umesto 1, pogrešićete za 81% dok ćete, ako zamenite 4 sa 5, pogrešiti samo za 0.08%!). Ovo je konvencija koja je usvojena praktično kod svih procesora, kako osmobitnih tako i moćnijih.

Relativno adresiranje

Ako postoji *apsolutno adresiranje,* verovatno postoji i *relativno!* Upravo tako: kod apsolutnog adresiranja specificiramo memorijske ćelije navodeći njihovo tačno mesto u memorijskoj mapi dok kod relativnog adresiranja navodimo rastojanje podatka od instrukcije koja se izvršava. Prototip ovakve instrukcije bi bilo JR PC+&20. PC je skraćenica od *Program Counter;* to je ime

registra koji prati izvršavanje programa i stalno sadrži adresu instrukcije koja će se sledeća izvršavati. Ta adresa će biti sabrana sa konstantom &20, a zatim će biti izvršen skok na naredbu čiju smo ovim sabiranjem dobili. Za šta bi mogla da se iskoristi ovakva komplikacija? Ukoliko napišemo čitav program koristeći ovakvo adresiranje za sve naredbe (posebno naredbe skoka), on će se korektno izvršavati bez obzira na to u koji ga segment memorije upišemo. Kako ovakva mogućnost nije naročito koristna, relativno adresiranje je kod jednostavnih procesora kao što je Z80 zastupljeno samo kod malog broja instrukcija. Suprotno tome, kod velikih kompjuterskih sistema koji istovremeno opslužuje više korisnika relativno adresiranje predstavlja osnovni modalitet rada.

Indirektno adresiranje

Indirektno adresiranje je daleko značajnije; bez njega se teško mogu ostvariti iole složenije programerske zamisli. Verujući da je većina čitalaca ovog umetka dobro upoznata sa bejzikom, objasnićemo indirektno adresiranje na primeru naredbi PRINT A(100) i PRINT A(I). Obe naredbe treba da prikažu na ekranu sadržaj nekog elementa niza A. Kod prve (PRINT A(100) je broj tog elementa fiksiran u okviru naredbe pa ona predstavlja primer *apsolutnog* adresiranja. Naredba PRINT A(I), sa druge strane, naređuje bejzik interpretatoru da najpre proveri vrednost promenljive I **a onda** da ispiše vrednost elementa niza čiji broj ona čuva. To je *indirektno* adresiranje jer se u okviru naredbe ne navodi pozicija argumenta već mesto na kome može da se nađe ta pozicija. U mašinskom programiranju Z80 se umesto promenljive za indirektno adresiranje koriste registri: instrukcija LD A,(HL) će, ukoliko se u registru HL nalazi broj &1234, dovesti u akumulator sadržaj memorijske ćelije čija je adresa &1234.

Indeksno adresiranje

O *indeksiranom adresiranju ćemo* govoriti docnije, obzirom da je ono nešto komplikovanije od svih do sada pobrojanih. Ukratko: u okviru same instrukcije ili u nekom registru se nalazi tzv. *bazna adresa* a na nekom drugom mestu tzv. *ofset*. Računar sabira baznu adresu i ofset, a zatim pristupa ćeliji čija je adresa određena ovim zbirom. Ako se, na primer, u registru IX nalazio broj &2000, naredba LD A,(IX+&20) će u akumulator dovesti sadržaj ćelije čija je adresa &2000+&20=&2020. Indeksirano adresiranje je izuzetno značajno za većinu mikroprocesora, ali ga Z80 koristi u gotovo zanemarljivoj meri.

Asembleri

Ulogu asemblera (ili asembler programa) već poznajemo: oni treba da prevedu program pisan mnemoničkim skraćenicama (zvaćemo ga *izvorni program)* u *objektni kod* — pravi mašinski program koji će biti upisan negde u memoriju. Ukoliko nameravate da vam ovaj umetak posluži za bilo šta osim za ukras, ***nabavite asembler program za vaš kompjuter.*** To nije naročito velika investicija (ovakvi programi su kod preprodavaca jeftini, valjda zato što slabo koga interesuju), a predstavlja *neophodan* uslov za rad na mašinskom jeziku. Pretpostavljajući da asembler program (ili, kraće, asembler) već posedujete, nećemo se baviti prevođenjem instrukcija na pravi mašinski jezik; ukoliko dođete do toga da vam nešto takvo zatreba, verovatno ćete biti dovoljno dobar poznavalac mikroprocesora da se sami snađete. Ovo poglavlje je zato posvećeno isključivo upotrebi asemblera.

Asemblerske direktive

Asembler je program kao i svaki drugi: pravi se u mnogo verzija i svaka od njih treba da ima svoje karakteristike i svoje uputstvo za upotrebu. U „Računarima 12" i „Računarima 13" smo opširno opisali upotrebu najpoznatijeg asemblera za „spektrum" i „amstrad" koji se zove DEVPAC, dok smo se asemblerom za „galaksiji" bavili u okviru umetka „ROM 2" iz „Računara 3". Ukoliko, što nije naročito verovatno, koristite neki drugi program ili drugi računar, moraćete da pročitate originalno uputstvo, što vam u svakom slučaju savetujemo i kada se radi o DEVPAC-u. Videćete da se asembler sastoji iz *editora* koji omogućava da unosite, ispravljate i snimate program na traku ili disketu i *prevodioca* koji ovako formiran tekst (kažemo tekst, jer su mnemoničke skraćenice obične reči) prevodi u ojbektni kod i upisuje ga u memoriju. Editori se razlikuju od asemblera do asemblera, dok prevodioci rade manje-više jednako što znači da ćemo im posvetiti posebnu pažnju.

Da bi prevodilac mogao da smesti program u memoriju, treba mu saopštiti adresu od koje program počinje. Ta adresa se naziva *origin* i obeležava sa ORG. Ukoliko se na početku programa pisanom mnemoničkim skraćenicama nađe ORG &3000, rezultujući mašinski kod će biti upisan u memoriju počevši od adrese &3000.

Reč ORG nije mašinska instrukcija pošto se ona ne prevodi u binarni broj koji bi bio smešten u memoriju. Ona, umesto toga, daje prevodiocu uputstvo o tome šta da radi sa pravim instrukcijama, pa se naziva *asemblerskom direktivom*. Asemblerskih direktiva, jasno, ima mnogo pa ćemo ih

X	Y	X xor Y
0	0	0
0	1	1
1	0	1
1	1	0

slika 8.

Da vidimo za šta može da se iskoristi ova ekskluzivna operacija. XOR A zvuči prilično glupo: računamo A XOR A i rezultat opet šaljemo u A! Međutim, primetimo da se na taj način vrši ekskluzivno ili (tzv. *ilili)* nad jednakim bitovima. Kako je 1 XOR 1=0 i 0 XOR 0=0 u akumulatoru će se naći sve same nule, kao da smo napisali LD A,0! Mogli smo, naravno, da napišemo i ovo, ali bismo tada utrošili jedan bajt više i dobili program koji se sporije izvršava.

Došlo je vreme da objasnimo i smisao naredbe AND A koja briše sadržaj indikatora prenosa. Briše ga zato što u tabeli instrukcija piše da svaka od naredbi AND, OR i XOR stavlja nulu u CY. Treba da dodamo da će vrednost akumulatora ostati nepromenjena, jer će se svaki bit logički množiti sa samim sobom; jedinica ostaje jedinica jer je i AND 1 = 1, a nula ostaje nula jer je 0 OR 0=0. Mogli smo, sasvim ravnopravno, da upotrebimo i OR A, dok bi XOR A takođe obrisalo indikator, ali bi uzgred obrisalo i akumulator!

Još jedna lepa osobina naredbi AND, OR i XOR je da postavljaju indikator nule (Z) ukoliko se u akumulatoru dobija nula. Ako nam je, na primer, potrebno da ispitamo da li je u akumulatoru nula, napisaćemo AND A a zatim JP Z,NULA *(=JumP if Zero set,* skoči ako je zero fleg setovan). Sadržaj akumulatora, ako je različit od nule, neće biti izgubljen, jer smo već rekli da AND A utiče samo na flegove.

Šesnaestobitna aritmetika

Sve osmobitne aritmetičke operacije se obavljaju nad sadržajem akumulatora, dok ulogu šesnaestobitnog akumulatora izigrava registar HL. Druga razlika između osmobitnih i šesnaestobitnih aritmetičkih naredbi je što ove druge utiču na mali broj flegova — DEC A će, na primer, setovati Z(ero) fleg ako je sadržaj akumulatora postao jednak nuli, dok će posle DEC HL stanja svih flegova ostati nepromenjena. No, pođimo redom.

Naredbe ADD i ADC imaju isto dejstvo kao i njihovi osmobitni ekvivalenti, ali se primećuje razlika u mnemonici: piše se ADD B ali i ADD **HL**,BC a ne, kao što bi se dalo pomisliti, ADD BC; verovatno je Zilog želeo da se obezbedi od grešaka koje bi mogle da nastanu kada programer izostavi jedno slovo pa se ADD BC pretvori u ADD B. Bilo kao bilo, opšti oblik ADD i ADC instrukcija je ADD (ADC) HL, rp gde je **rp** bilo koji od registarskih parova BC, DE, HL i SP. Za šesnaestobitno oduzimanje se koristi SBC HL, rp dok naredba SUB ss **ne postoji;** ako vam je potrebno obično šesnaestobitno oduzimanje, moraćete da resetujete indikator prenosa pa da izvršite SBC. Što se indeksnih registara tiče na raspolaganju su vam naredbe ADD IX,pp i ADD IY,rr pri čemu se na sadržaj IX (IY) može dodavati BC, DE, SP i IX (IY). **Ne postoje** naredbe ADD, IX,IY ili ADD IY,IX, a ni naredba koju ste svakako očekivali: ADD IX,HL.

Naredba INC ss omogućava povećanje, a DEC ss umanjenje bilo kog registarskog para za jedan, pri čemu se stanje flegova ne menja; ako sa DEC smanjujete brojač u petlji, moraćete da dodate nekoliko naredbi koje će proveriti da li je on stigao do nule. Što se indeks registara tiče, poslužiće vam naredbe INC IX i INC IY, kao i odgovarajuće DEC instrukcije.

Operacije sa bitovima

Premda operiše sa bajtovima, Z80 je opremljen sa nekoliko veoma moćnih naredbi za rad sa njihovim sastavnim delićima — bitovima. Osim šiftovanja i rotiranja koja omogućavaju i svi drugi osmobitni mikroprocesori, Z80 omogućava i setovanje, resetovanje i testiranje bilo kog bita u registrima ili, pod određenim uslovima, u memoriji.

Postavljanje, brisanje, i testiranje

Iako smo nekoliko puta pominjali termine *nulti bit* ili *najznačajniji bit nekog bajta,* pogled na sliku 9 neće škoditi. Vidimo broj 123 ili &7B koji je upisan u neki registar; najniži ili nulti bit ovoga broja je, kao što vidimo, jedinica, a najviši, sedmi bit je nula. Ostali bitovi su numerisani brojevima 1—6.

Pretpostavimo da je broj &7B upisan u registar B i da Z80 izvrši naredbu SET 7,B. Obzirom da već dosta dugo gledamo razne instrukcije, dejstvo ove neće biti teško pogoditi: ona setuje sedmi (poslednji) bit registra B, pa broj %01111011 postaje %11111011=&FB. Da li smo ovaj rezultat mogli da dobijemo još nekako? Naravno da jesmo: naredbama LD A,B: OR &80: LD B,A možemo da simuliramo BIT 7,B uz utrošak četiri umesto dva bajta programa. Možda je značajnije što će nam SET 7,B, kada docnije budemo posmatrali program, olakšati njegovo razumevanje.

Naredbom SET ne možemo, na žalost, da zamenimo OR: njen opšti oblik je SET b, t gde je ***b*** broj između 0 i 7 a **t**, kao i ranije, bilo koji od registara (SET b, reg), memorijska ćelija na koju pokazuje HL (SET b,(HL)) ili neki od indeks registara (SET b,(IX+d) odnosno SET b,(IY+d)); **b** je, da ponovimo **broj između 0 i 7,** to jest broj koji mora da se zna u vreme pisanja programa; ako

Naredbe ADD i ADC utiču i na Z, P/V i S flegove: ukoliko je, na primer, rezultat sabiranja 0, biće setovan Z(ero) fleg. Kako može da se dobije nula sabiranjem dva pozitivna broja? Pokušajte, na primer, da saberete &F0 i &10 i dobićete &100; kako u akumulator može da stane svega osam bita rezultata, biće upisano 00 i setovan indikator nule. P/V fleg označava nekorektan rezultat sabiranja označenih brojeva o kome smo govorili u drugom poglavlju. Konstruktori Z80 su nas opremili naredbama za testiranje P/V flega, tako da u slučaju ovakve greške možemo da programiramo skok koji će prekinuti izvršavanje programa ili izdati neku poruku.

Pošto smo, na jedvite jade, shvatili kako se koriste naredbe ADD i ADC, treba da kažemo šta mogu da budu njihovi argumenti. Jedan od sabiraka je **uvek** u akumulatoru u koji se smešta i rezultat. Iza ADD (ADC) navodimo adresu drugog sabirka koji može da bude konstanta (ADD nn, npr. ADD &20), da se nalazi u nekom drugom osmobitnom registru (ADD reg) ili da bude u memoriji tako da na njega pokazuje registar HL (ADD (HL) ili neki od indeksnih registara (ADD (IX+dd) odnosno ADD (IY+dd). Verujemo da nema potrebe da ponovo objašnjavamo suštinu i primenu neposrednog, apsolutnog, indirektnog i indeksnog adresiranja koje se koristi kod ovih naredbi.

Oduzimanje osmobitnih brojeva

Oduzimanje je sasvim slično sabiranju: SUB s će od akumulatora oduzeti sadržaj **s** pri čemu **s** može da bude broj (SUB nn), registar (SUB reg), memorijska ćelijska ćelija na koju pokazuje HL (SUB (HL)) ili memorijska ćelija adresirana pomoću indeks registra (SUB (IX+dd) odnosno SUB (IY+dd)). SBC s je slična naredba koja ima iste adresne modove, ali se bitno razlikuje od SUB po tome što se u akumulator ne stavlja (A-s) nego (A-s-CY) gde je CY, kao i ranije, stanje indikatora prenosa. Primenom bilo koje od ovih instrukcija indikator prenosa će biti setovan ako je potrebna „pozajmica", tj. ako smo od manjeg broja oduzimali veći, dok će biti resetovan ako je operacija korektno izvršena. Na taj način, prostor zamenom ADD sa SUB i ADC sa SBC u prethodnom primeru, možemo da oduzimamo šesnaestobitne brojeve. I kod oduzimanja Z fleg biva setovan ako je rezultat u akumulatoru nula, dok se P/V setuje ako je nastupila greška. S(ign) fleg označava znak rezultata koji ima smisla samo ako operišemo sa označenim osmobitnim brojevima.

Specijalan slučaj sabiranja i oduzimanja su instrukcije INC t odnosno DEC t, gde **t** može da bude bilo koji od registara (reg) ili memorijska ćelija na koju pokazuje HL (DEC (HL)) ili neki od indeks registara (INC (IX+dd)) odnosno DEC (IY+dd)). Instrukcija INC (=increment, povećaj) će povećati sadržaj **t** za jedan dok će ga DEC **t** za isto toliko umanjiti. Videćemo da su ove naredbe vrlo značajne u petljama gde je potrebna česta promena sadržaja raznih brojača. INC i DEC ne deluju na indikator prenosa, što znači da će INC A dati nulu ako se u akumulatoru nalazilo &FF, ali se zato Z i S fleg postavljaju u slučaju da je rezultat nula, odnosno da je negativan.

Obzirom da smo pomenuli naredbu SCF koja setuje indikator prenosa promenimo i CCF *(change carry fleg)*, instrukciju koja menja njegovo stanje: ako je CY bilo jedan, postaće nula, a ako je bilo nula postaće jedan. Brisanje CY možemo, umesto sa AND A, da izvedemo sa SCF:CCF, ali tako trošimo jedan bajt više. U osmobitne aritmetičke instrukcije možemo da ubrojimo i NEG, jednostavnu naredbu koja nalazi drugi komplement sadržaja akumulatora, tj. menja znak broja u akumulatoru. Znajući da je A−B = A+(−B) mogli bismo da zamenimo SUB B sa:

```
PUSH    AF      ; čuvanje sadržaja skumulatora na steku.
LD      A,B
NEG
LD      B,A     ; promena znaka broja u B.
POP     AF      ; restauracija sadržaja akumulatora.
ADD     B       ; A=A+(−B)=A−B.
```

U tabeli ćete pronaći i naredbu DAA *(decimal adjust accumulator)* koja se koristi za rad sa BCD *(binary coded decimal)* brojevima. Kako se u ovom umetku bavimo samo osnovnim metodama predstavljanja brojeva, ovoj naredbi posvećujemo tekst u jednom od sledećih brojeva „Računara".

Logičke operacije

Došlo je vreme da se pozabavimo logičkim operacijama AND, OR, XOR i CPL. Prve dve smo upoznali na početku ove knjige, a poznajemo i poslednju samo pod drugim imenom: CPL je isto što i NOT, naredba koja nalazi prvi komplement akumulatora to jest svaku jedinicu u njemu pretvara u nulu a svaku nulu u jedinicu. AND, OR i XOR su, sa druge strane, binarne operacije koje imaju po dva argumenta pa ćemo im posvetiti malo više pažnje.

Jedan od argumenata bilo koje od ovih operacija se nalazi u akumulatoru u koji se smešta i rezultat. Drugi može da se nalazi u ovkiru same instrukcije (AND nn, neposredno adresiranje, npr. AND &F0), u nekom od registara (AND reg) ili u memoriji (AND (HL) odnosno AND (IX+dd)); možemo, sve u svemu, da kažemo da postoje instrukcije AND s, OR s i XOR s gde smo smisao slova **s** već ranije objasnili.

Nije teško pogoditi šta rade naredbe AND s i OR s: sadržaj akumulatora se, bit po bit, poredi sa **s** pa se formira rezultat prema tabelama sa slika 3 i 4. XOR s radi nešto sasvim slično pri čemu se nad bitovima izvodi ekskluzivna disjunkcija koja je definisana tabelom sa slike 8. Ekskluzivn disjunkcija, kao što joj i ime kaže, liči na disjunkciju (OR), ali se od nje malo i razlikuje: 1 OR 1=1 ali je i XOR 1=0!

upoznavati kako nam budu trebale. Neke od njih imaju sasvim banalno i sa aspekta posla koji treba da se obavi nebitno značenje: da li će se pri asembliranju listing prikazivati na ekranu ili će se ispisivati na štampaču, da li će računar prijavljivati sve greške ili samo one fatalne i slično. Neke druge direktive (kao što je ORG koju smo upravo upoznali) su fundamentalno važne.

Pseudoinstrukcije

Osim asemblerskih direktiva i mnemoničkih skraćenica, u asemblerskim listinzima mogu da se nađu i *pseudoinstrukcije.* Pseudoinstrukcije, iako ne predstavljaju nijednu od naredbi koju bi mikroprocesor razumeo, u rezultujući kod upisuju određeni sadržaj. Pretpostavimo, na primer, da naš program treba da ispiše tekst „ZDRAVO" na ekranu. Ispisivanje se, uzećemo najjednostavniji slučaj, obavlja tako što se poruka ZDRAVO, slovo po slovo, prepisuje u video memoriju, što znači da slova Z, D, R, A, V i O moraju da budu upisana negde u RAM. Program koji rešava problem bi mogao da izgleda otprilike ovako:

```
LD      HL, PORUKA      ; u registar HL se upisuje adresa početka poruke.
LD      DE, VIDEO       ; u registar DE se upisuje adresa početka
                        ; video memorije.
petlja
LD      A,(HL)          ; slovo poruke u akumulator.
CP      0               ; kraj poruke je nula bajt.
JP      Z,kraj          ; ako je u akumulatoru 0, ide se na deo
                        ; programa označen sa KRAJ.
LD      (DE),A          ; slovo se ispisuje na ekranu.
INC     HL              ; X se povećava za 1.
INC     DE              ; Y se povećava za 1.
JP      petlja          ; ispisuje se sledeće slovo.
kraj
HALT                    ; kraj rada.
poruka
DEFM    „ZDRAVO"
DEFB    0
```

Nije previše važno da shvatite kako tačno radi ovaj program; u globalu posmatrano, računar čita slovo po slovo poruke i ispisuje ga na ekranu. Obzirom da bi ispisivanje bilo „mrtva petlja", na kraj poruke je upisan nula bajt koga program detektuje (naredba COMPARE, CP) i prekida sa radom.

Obratimo pažnju na istaknute pseudonaredbe DEFM i DEFB. Prva od njih označava da prevodilac treba u memoriju da upiše tekst ZDRAVO, svako slovo u po jednu memorijsku ćeliju koristeći, naravno, ASCII kod. Slično tome, pseudonaredba DEFB zahteva od prevodioca da u memoriju upiše bajt 00. Ukoliko nekom prijatelju docnije budete dali **mašinski** program poput ovoga, on će perfektno raditi. Međutim, ako taj prijatelj bude pokušao da shvati kako je program napisan, tj. da ga pretvori u mnemoničke skraćenice, naći će se u velikom čudu kada bude pokušao da prevede (odnosno, kako se to stručno kaže, *disasemblira)* deo programa koji predstavlja poruku ZDRAVO. Naići će, naime, na bajtove od kojih neki predstavljaju instrukcije, dok su drugi nedefinisani. Čak i oni koji predstavljaju instrukcije će imati besmislene adresne delove i, uopšte, neće ličiti na program. U sličnoj situaciji bi se nalazio i sam mikroprocesor da smo zaboravili na naredbu HALT koja prekida njegov rad: pokušao bi da izvrši test ZDRAVO kao mašinski program i pri tom verovatno upao u neku vrstu beskonačne petlje.

Disasembleri

Iz čitave ove priče može da se izvede jedan izuzetno važan zaključak: od izvornog teksta programa se jednoznačno dobija objektni kod uz pomoć programa koji nazivamo asembler. Obrnuti proces, dobijanje mnemonički pisanog programa iz objektnog koda (disasembliranje) nije jednoznačan i, osim kod sasvim jednostavnih programa, nije ni malo lak. Vidimo, na primer, da smo u programu koji smo pisali označili tzv. *tabelama* (o njima docnije) tri mesta koja smo nazvali 'petlja', 'kraj' i 'poruka', pri čemu ti nazivi otprilike asociraju na namenu oznaka. Ukoliko neko disasemblira ovaj program, neće nikako moći da zna za ove naše oznake pa će mu praćenje čitave stvari biti daleko komplikovanije. Zato **uvek i obavezno** čuvajte na traci ili disketi izvorne verzije svih programa koje napišete; te verzije će vam biti dragocene ako jednom poželite da promenite neki svoj program ili da ga uklopite u nešto drugo.

Vrsta asemblera

Došlo je vreme da se pozabavimo vrstama asemblera koje ćete sresti kod kućnih računara. Najjednostavniji (i, paradoksalno, najkorisniji, ne samo za početnike) asembleri su uklopljeni u bejzik: asemblerske instrukcije pišete kao obične bejzik naredbe, a zatim ispred i iza njih navodite

nešto (npr. otvorenu i zatvorenu uglastu zagradu) što će „objasniti" računaru da treba da ih prevodi a ne da ih izvršava. Zbog čega je to zgodno? Pre svega, asembler je u ROM-u zajedno sa bejzik interpretatorom pa vam je stalno pri ruci. Ne morate, osim toga, da učite da upotrebljavate neki novi editor: mašinske programe kucate i ispravljate poput bejzika. Da stvar bude posebno lepa, možete lako da kombinujete bejzik program koji će da učitava podatke i štampa rezultate sa asemblerskim potprogramima koji će se pobrinuti da spora izvračunavanja postanu brza.

Druga mogućnost je asembler koji se, sa svojim editorom, upisuje u memoriju sa trake ili diska. Mana ove koncepcije je što, kada vam program krahira i izbriše memoriju računara (a to će vam se, verujte nam na reč, vrlo često dešavati, ne samo dok učite) morate da učitavate ne samo izvorni program već i čitav asembler koji je dug nekoliko kilobajta. Ovaj problem postaje posebno akutan ako sve treba da se učita sa trake! Za uzvrat, imate editor koji je posebno prilagođen radu sa asemblerom i koji, eventualno, poseduje mogućnosti makro ekspanzije (možete da upotrebljavate svoje „mašinske naredbe", koje se, pre asembliranja, automatski zamenjuju nizovima pravih mašinskih naredbi koje ste ranije definisali). Ukoliko se opremite ovakvim asemblerom moraćete, naravno, da naučite da koristite jedan nov editor što možda i nije tako loše.

Poslednji i, uslovno rečeno, najprofesionalniji asembler programi uzimaju izvornu verziju sa diska, a zatim prevedeni kod smeštaju na isti medij. Šta se time dobilo? Pre svega, treba da primetimo da se instrukcija LD A, (&1000) sastoji od petnaestak znakova koji se prevode u svega tri bajta objektnog koda. To znači da je izvorni zapis programa, pogotovu kada ga dopunimo komentarima, višestruko duži od konačnog programa koji pišemo. Ukoliko naš računar ima 48 Kb ROM-a od kojih 8 odu na sadržaj ekrana, sledeća četiri za program koji smo nazvali asemblerom i još par kilobajta za razne sistemske promenljive, ostaje nam tridesetak kilobajta. Ukoliko zaista želimo da napišemo toliko dugačak mašinski program, njegov izvorni oblik će biti dug stotinak kilobajta, što znači da nema nikakve šanse da stane u memoriju. Kapaciteti disketa su, sa druge strane, daleko veći, pa nema mnogo problema da se naš izvorni program rastegne na stotinak kilobajta! Naravno, ovakvi problemi, kao i asembleri koji ih rešavaju, prilično su daleko od čitalaca ovog umetka, pa ih zato nećemo podrobnije razmatrati.

Monitori

Osim asemblera, za učenje mašinskog jezika će vam dobro doći program koji se naziva *monitor* ili *debager*. Ovaj program vam omogućava da pretražujete memoriju i menjate njen sadržaj i, što je posebno važno, izvršavate mašinske programe instrukciju po instrukciju posmatrajući promene u registrima i memoriji. Iskusni programeri koriste ovakve programe za traženje grešaka i tako štede svoje vreme dok će za početnike svakako biti interesantna mogućnost da, u nekoj vrsti „usporenog filma", proučavaju delovanje naredbi koje upoznaju. Reći ćemo, ipak, da upotreba monitor programa obično nije sasvim jednostavna i da sa njima treba da se upoznate tek kada, bar pojmovno, shvatite osnove mašinskog programiranja.

Pre nego što pređete na čitanje sledećih poglavlja koja će se baviti naredbama mikroprocesora Z80, treba da rešite par praktičnih problema koji će proisteći iz činjenice da programi koje ćemo sastavljati mogu da se izvršavaju na računaru, što znači da je poželjno da neke od njih i otkucate. Pre toga morate da proverite na koje memorijske adrese treba da smeštate mašinske programe kako se oni ne bi „kosili" sa asemblerom i izvornim kodom (ako se, na primer, asembler nalazi u memorijskim ćelijama čije su adrese &3000—&4000, direktiva ORG &35000 će skoro sigurno izazvati krah čitavog sistema, jer će asembler, asemblirajući program, uništiti samoga sebe). Treba, osim toga, da proverite kako se mašinski programi pozivaju iz bejzika (za to obično služi naredba CALL ili funkcija USR) i kako se iz njih vraća u bejzik (obično mašinskom naredbom RET).

Arhitektura Z80

U uvodnom smo poglavlju objasnili pojmove *registar* i *fleg;* došlo je vreme da te pojmove upoznamo i na konkretnom primeru — arhitekturi mikroprocesora Z80.

Z80 je opremljen sa 14 opštih registara i osam registara specijalne namene; ovaj broj je, kada se radi o osmobitnom procesoru, prilično impresivan (6502, na primer, ima samo pet registara) što znači da vešt programer na Z80 može da izbegne prečesto pristupanje memoriji koje produžava i usporava program. Naravno, da bismo postali vešti programeri koji će postići to skraćenje i ubrzanje, treba dobro da upoznamo mnogobrojne registre koje nam je Zilog poklonio.

Registri specijalne namene

Upoznati svrhu registara opšte namene nije teško — u njih smeštamo podatke sa kojima operišemo. Jedan od tih registara (A) smo već upoznali — u njemu se obavlja većina aritmetičkih i logičkih operacija. Ostali registri opšte namene se obeležavaju slovima B, C, D, E, H i L. Svaki od njih može da sačuva po jedan osmobitni broj, pri čemu se, za neke instrukcije, ovi registri kombinuju u šesnaestobitne BC, DE i HL. Zanimljivo je reći da je Zilog uspeo da ovim šesnaestobitnim registrima, čija su imena dobijena prostimređanjem slova abecede, dodeli specijalna značenja: BC je skraćenica od Byte Counter, tj. brojač bajtova; nešto kao kontrolna

Sabiranje osmobitnih brojeva

Postoje dve naredbe za sabiranje osmobitnih brojeva: ADD i ADC. Obe imaju po jedan argument koji naznačava broj koji treba dodati sadržaju akumulatora, pri čemu se rezultat opet smešta u akumulator. U čemu je onda razlika između ove dve naredbe? ADD B će dodati sadržaj registra B na registar A, dok će ADC B na sadržaj registra A dodati registar B **i stanje C flega.** Kako se dodaje stanje nekog flega? Ništa lakše: ako je fleg setovan, vrednost mu je jedan, a ako je resetovan onda mu je vrednost nula. Posmatrajmo, dakle, sledeća dva programa:

```
                        SCF
LD      B,&20           LD      B,&20
LD      A,&40           LD      A,&40
ADD     B               ADC     B
```

Prve dve LD naredbe u oba programa treba da dovedu konstante &20 i &40 u registre B i A, a zatim sledi njihovo sabiranje. Pitanje je, naravno, šta će se po izvršavanju programa dobiti u akumulatoru. Što se prvog primera tiče, stvar je jednostavna: u akumulatoru će biti &20+&40+**CY** gde smo sa **CY** označili stanje indikatora prenosa (carry flega). Kakvo je to stanje? Obratimo pažnju na prvu naredbu drugog programa SCF; ova reč je skraćenica od *set carry flag,* što bi moglo da se prevede i kao *stavi 1 u CY (carry).* CY će, dakle, biti 1, pa će izvršavanjem drugog programa u akumulatoru biti &20+&40+1=&61!

Šta bi se desilo da smo izostavili naredbu SCF? Reklo bi se da je tada stanje indikatora prenosa nedefinisano, ali ono jednostavno ne može da bude takvo: fleg može da bude ili setovan ili resetovan; trećega nema! Zato će izvršavanjem prethodnog dela programa (ili radom bejzik interpretera ako je naš mašinski program upravo pozvan) biti formirana vrednost CY, pa će naredba ADC raditi korektno iako možda ne onako kako smo želeli. Zapamtite da, ako planirate korišćenje naredbi ADC i SBC (subtract, oduzimanje) **uvek** setujete indikator prenosa primenom SCF, ili ga resetujete naredbom AND A (zašto ova naredba resetuje indikator prenosa? Zato što tako piše u tablici. Zašto ne menja ništa drugo što bi bilo bitno? Malo strpljenja!)

Videli smo kako se izvršavaju naredbe ADD i ADC, ali nam je i dalje nejasno kakav je smisao kvarenja lepo dobijenog zbira dodavanjem indikatora prenosa. Zamislimo da treba da saberemo dva šesnaestobitna broja od kojih se jedan nalazi u memorijskim ćelijama &2000 i &2001, a drugi u &2002 i &2003. Zbir treba da smestimo u &200A i &200B. Obzirom da smo rekli da Z80 ima naredbe za šesnaestobitno sabiranje, problem se lako rešava. Te naredbe, međutim, još nismo upoznali pa nećemo ni da ih koristimo; one nam, uostalom, ne bi mnogo pomogle da treba da sabiramo trideset dvobitne ili veće brojeve. Kako da rešimo problem? Pokušaćemo da najpre saberemo sadržaje memorijskih ćelija &2000 i &2002 (niže bajtove oba broja) i rezultat smestimo u &200A, a da zatim saberemo sadržaje &2001 i &2003 i rezultat smestimo u &200B. Evo programa koji to radi:

```
LD      A,(&2002)
LD      B,A             ; pošto naredba LD B, (&2002) ne postoji.
LD      A,(&2000)
ADD     B
LD      (&200A),A       ; sabrali smo manje signifikatne delove brojeva.
LD      A,(&2003)
LD      B,A
LD      A,(&2001)
ADD     B
LD      (&200B),A       ; sabrali smo više signifikantne delove brojeva.
```

Isprobajmo ovaj program stavljajući u &2000 i &2001 broj &2030 (POKE &2000, &30: POKE &2001,&20), a u &2002 i &2003 broj &4050. Posle startovanja programa dobićemo da se u ćeliji &200A nalazi broj &80 (ako iskoristite PRINT PEEK (&200A) dobićete 128 što ne treba da vas iznenadi; 128=&80), a u ćeliji &200B broj &60, što znači da je zbir &6080; program, dakle, radi ispravno. No, šta ako u ćelije &2000, &2001, &2002 i &2003 upišemo resprektivno brojeve &20, &3A, &F0 i &10? Po startovanju programa će se dobiti zbir &4A10, dok je &3A20+&10F0 = &4B10. Program je, dakle, pogrešan!

Kako je nastupila greška? Z80 je najpre sabrao brojeve &20 i &F0 i dobio, ili bar trebao da dobije, &110. Broj &110 je, međutim, veći do &FF, pa se pojavio bit koji nije mogao da stane u akumulator. Onako kako smo ga napisali, program je zanemario taj bit, pa je sabrao brojeve &3A i &10 i dobio &4A; da je dodao prenos, dobio bi &4B što bi bio tačan rezultat.

Problem ćemo rešiti tako što ćemo drugu ADD instrukciju zameniti sa ADC B i sve će biti u redu. Kako je to moguće? Osim formiranja rezultata u akumulatoru, naredbe ADD i ADC menjaju sadržaje nekih flegova od kojih je jedan i CY. Ako se pojavi prenos, to jest ako je rezultat sabiranja veći od &FF, indikator prenosa biva setovan, dok je u suprotnom resetovan. Na taj način će ADC instrukcija uzeti u obzir prenos od prethodnog sabiranja i formirati korektan rezultat. Ostaje još da objasnimo razlog zbog koga nismo obe ADD instrukcije zamenili sa ADC. U tom slučaju bismo morali da resetujemo indikator prenosa kako bismo osigurali korektno obavljanje prvog sabiranja; ako bi CY bio setovan, dobili bismo za jedan veći rezultat od korektnog. Mogli smo, naravno, da napišemo jedno AND A (naredba kojom, kako rekosmo, resetujemo CY), ali bi to potrošilo bajt programa više!

Bilo je, kao i obično, više priče nego programa. Pogledajmo, ipak, kako sve to radi: ako je korisnik tražio da mu se prikaže sadržaj ćelije čija je adresa &3333, u registru HL će se, po povratku iz potprograma TAST, nalaziti broj &3333. LD C, (HL) je, jasno, specijalni slučaj instrukcije LD reg, (HL) gde *reg* predstavlja bilo koji registar. Primenom ove ćelije se *u registar C dovodi sadržaj memorijske ćelije čija je adresa u HL*, u našem slučaju ćelija &3333. Taj broj će, primenom potprograma ISPIS, biti napisan na ekranu.

Za indirektno adresiranje kod Z80 se uglavnom koristi registar HL koji smo jednom nazvali *šesnaestobitnim akumulatorom:* posedujemo čak i instrukciju LD (HL), nn kojom možemo da upisujemo konstante direktno u memoriju. Primetimo da instrukcija Ld (nnmm), nn ne postoji, što znači da *ne možemo* da napišemo nešto poput LD (&2000), 0; moraćemo da upotrebimo LD A,0:LD (&2000), A ili LD HL, (&2000):LD (HL),0, zavisno od toga da li nam je draže da obrišemo prethodni sadržaj akumulatora ili prethodni sadržaj registra HL. Indirektno smeštanje akumulatora u memoriju ili formiranje njegovog sadržaja možemo da postignemo i primenom instrukcija LD (BC),A odnosno LD (DE),A i njihovih „ogledalskih" simetrija LD A, (BC) odnosno LD A,(DE). Suvišno je reći da *ne postoje* instrukcije LD B,(DE), LD H, (BC) i slične. Sve instrukcije indirektnog adresiranja ili, kako bi ga puritanci nazvali, indirektnog *registarskog* adresiranja kod Z80 zauzimaju samo po jedan bajt.

Indeksirano adresiranje primenom registara IX i IY je daleko sporije i troši daleko više memorije pa se retko koristi. Posedujemo instrukcije LD (IX+dd), reg, LD reg, (IX+dd) i LD (IX+dd),nn kao i odgovarajuće instrukcije koje umesto IX koriste IY. Izvršavaju se tako što se sadržaj IX (ili IY) sabere sa brojem *dd* koji se nalazi u okviru same instrukcije i tako se dobije adresa ćelije kojoj treba pristupiti. Najveća vrednost za *dd* je 255.

U osmobitne load instrukcije spadaju i one koje prenose sadržaj akumulatora u specijalne registre I i R i koje omogućavaju njihovo čitanje tj. prenos vrednosti u akumulator. Za sada vam ove instrukcije neće biti naročito značajne.

Šesnaestobitne load instrukcije

Kao što smo sa *reg* obeležili bilo koji osmobitni registar opšte namene, sa *rp* ćemo obeležiti bilo koji od registarskih parova BC, DE, HL ili specijalni registar SP. Bilo koji od ovih registara možemo da popunimo proizvoljnom šesnaestobitnom konstantom koristeći instrukcije tipa LD rp, nnmm i trošeći tri bajta programske memorije; u prvi se smešta kod instrukcije, a u preostala dva šesnaestobitna konstanta *nnmm* upisana, kao i obično, tako da se na nižoj adresi nalazi niži bajt. Pogled na tabelu iz našeg prošlog umetka će pokazati da se LD HL,&2DCC kodira kao &21 &CC &2D. Za punjenje registara IX i IY konstantama smo snabdeveni naredbama LD IX, nnmm i LD IY, nnmm.

Potrebne su nam, naravno, i instrukcije za prenošenje sadržaja memorije u šesnaestobitne registre. Opšti oblik takve naredbe je LD rp, (nnmm); njenim izvršavanjem se u registar *rp* dovode sadržaji memorijskih ćelija čije su adrese *nnmm* i *nnmm+1*. Naredba LD BC, (&2DAA) će, na primer, dovesti u C sadržaj ćelije &2DAA a u B sadržaj ćelije &2DAB. Instrukcije ovoga tipa zapremaju čitava četiri bajta memorije: dva za njihov kod i dva za adresu ćelije. Konstruktori Z80 su obezbedili i specijalnu naredbu LD HL, (nnmm) koja zauzima samo tri bajta; jasno je da je LD HL, (&2000) specijalni slučaj naredbe LD rp, (nnmm) i da može da se kodira kao &ED &6B &00 &20 ali se ista naredba može kodirati i kao &2A &00 &20 zauzimajući bajt manje. Vaš asembler će, ako iša vredi, uvek automatski izabrati ovaj drugi oblik kodiranja.

Pošto ste polako počeli da razumevate strukturu LD instrukcija, neće vam biti teško da pogodite da je obezbeđena instrukcija LD (nnmm), rp i, kao njen specijalni slučaj, LD (nnmm), HL; ove se naredbe koriste za smeštanje sadržaja šesnaestobitnih registara u memoriju. Naredbe LD IX, (nnmm) i LD (nnmm), IX (iste takve postoje i za IY) se koriste za rad sa sadržajima indeksnih registara.

Na prvi je pogled dosta čudno što nas Zilog *nije* opremio instrukcijama tipa LD rp, rp' kojima bismo mogli da razmenjujemo sadržaje šesnaestobitnih registara (nemamo, na primer, naredbu LD HL, DE). Pokazuje se, međutim, da ove instrukcije nisu naročito potrebne, jer mogu da se simuliraju uz utrošak svega dva bajta programa: LD HL,DE može da se zameni sa LD H,D:LD L,E! Registar SP je izuzetak od ovoga pravila: kako ne možemo da pristupamo njegovim osmobitnim komponentama, snabdeveni smo naredbama LD SP,HL, LD SP,IX i LD SP,IY ali *ne* i naredbom LD HL,SP.

U šesnaestobitne load instrukcije spadaju strogo uzevši, i naredbe PUSH i POP kojima se operiše sa stekom, ali ćemo se njima pozabaviti u posebnom poglavlju.

Aritmetičke i logičke operacije

U drugom poglavlju smo upoznali operacije sa binarnim brojevima koje Z80 zna da izvrši: sabiranje i oduzimanje. Poput prenošenja podataka, sabiranje i oduzimanje može da bude osmobitno ili šesnaestobitno, dok se tek primenom malih mašinskih programa može operisati sa većim brojevima.

promenljiva u FOR-NEXT petlji bejzika. DE je skraćenica za Destination (odredište): ovaj registar, pri prenošenju blokova memorije, sadrži adresu početka memorijskog bloka koji će da primi podatke. Ime registra HL (HighLow) treba jednostavno da nas podseti da će se, posle upisivanja broja &20 u registar H i broja &AA u registar L u registarskom paru HL nalaziti broj &20AA, a ne broj &AA20. Uskoro ćemo videti da se najveći broj osmobitnih operacija obavlja nad registrom A, a najveći broj šesnaestobitnih nad registrom HL koji predstavlja neku vrstu šesnaestobitnog akumulatora.

Alternativni registri

Upoznali smo registre A, B, C, D, E, H I L; svega njih 7. Na početku ovog poglavlja smo, međutim, rekli da Z80 ima 14 registara opšte namene. Preostalih 7 registara nazivamo *alternativnim* i obeležavamo sa ' (prim) iza imena. Tako postoje registri A', B', C', D', E', H' i L' kao i registarski parovi BC', DE' i HL' (obratite pažnju da se registar DE' sastoji od registra D' i E' a ne registara D' i E'; pravilnije bi ga možda bilo zvati registar D'E'). Mikroprocesor u jednom trenutku može da opšti samo sa jednim setom registara, ali postoji naredba EXX koja će razmeniti sadržaje B, C, D, E, H i L sa sadržajima B', C', D', E', H' i L' i tako učiniti alternativne registre pristupačnima a osnovne nepristupačnima. Što se registara A i A' tiče, njih razmenjujemo naredbom EX AF, AF' pri čemu se razmenjuje i registar F koga smo ubrojili u registre specijalne namene i o kome ćemo govoriti malo docnije.

Za šta mogu da se iskoriste alternativni registri? Zamislimo da smo sastavili potprogram koji će se pozivati izuzetno često (npr. 50 puta u sekundi) i koji obavlja neku operaciju nezavisnu od glavnog programa kao što je crtanje slike na ekranu našeg monitora (videćemo da ove uslove zadovoljavaju programi za opsluživanje interapta). Tom programu je za rad neophodno nekoliko registara, a istovremeno ne smemo da dopustimo da se promene vrednosti koje je u te registre postavio glavni program pre poziva potprograma. Mogli bismo, naravno, da sačuvamo sadržaje registara na steku ili ih prenesemo u fiksne memorijske ćelije. Daleko je, međutim, kraće i brže da na početak potprograma i na njegov kraj stavimo instrukcije EXX koje će sačuvati odnosno restaurirati sadržaje opštih registara. Postojanje alternativnih registara nam praktično omogućava paralelno izvršavanje dva nezavisna programa.

Registri opšte namene

Registri specijalne namene se označavaju sa PC, SP, IX, IY, F, I i R. Svi registri čija se imena sastoje od po dva slova su, kao i do sada, šesnaestobitni, pri čemu u setu instrukcija nije ostavljena mogućnost pristupanja njihovim osmobitnim komponentama (ne možemo, na primer, da pristupimo registru X koji je deo registra IX; iz tog razloga nije opasno postojanje imena I, IX i IY jer se zajedničko I u njima ne odnosi na isti registar). Registri R i F nisu, kao što bi se moglo očekivati, osmobitni: registru R nedostaje sedmi, najviši bit, dok registru F nedostaju bitovi 3 i 5. Razloge za ove anomalije ćemo shvatiti u vrlo skoroj budućnosti, pošto obradimo „normalne" registre PC, IX, IY i SP.

Programski brojač

U setu instrukcija nećete naći ni jednu jedinu koja operiše sa sadržajem registra PC a ipak se taj sadržaj neprekidno menja. PC je skraćenica od *Program Counter* ili, u bukvalnom prevodu, programski brojač. U ovom se registru nalazi adresa instrukcije koju mikroprocesor treba da obradi tj. adresa bajta koji će, pri normalnom izvršavanju programa, biti sledeći pročitan. Kako se neke instrukcije sastoje od više bajtova, registar PC će u toku njihovog izvršavanja nekoliko puta menjati vrednost. Pretpostavimo da je na redu izvršavanje instrukcije LD A, (&4000) koja je kodirana kao &3A &00 &40 i, kao što vidimo, zauzima tri bajta memorije. &3A je oznaka instrukcije LD A, (nnmm) a &00 i &40 predstavljaju adresu &4000 koja se, kao i uvek, prikazuje sa nižim bajtom napred. Pre izvršavanja instrukcije u PC će biti upisana adresa memorijske ćelije koja sadrži broj &3A — početak instrukcije koja se obrađuje.

Mikroprocesor najpre uzima kod instrukcije (&3A) i smešta ga u registar naredbi koji je za nas nevidljiv. Istovremeno se registar PC povećava za i tako da pokazuje na bajt &00. Mikroprocesor prepoznaje instrukciju LD A,(nnmm) i primećuje da mu je potrebna šesnaestobitna adresa pa zahteva učitavanje njenog nižeg bajta na koji pokazuje PC. U registar adrese (ponovo nevidljiv za nas) dolazi broj &00, a PC se povećava za 1 tako da pokazuje na broj &40. Sledi još jedno pristupanje memorije i prenošenje broja &40 u registar adrese, pri čemu se PC ponovo povećava i pokazuje na sledeću instrukciju koju mikroprocesor treba da izvrši. Pre nego što dođe do nje, mikroprocesor će, jasno, morati još jednom da pristupa memoriji da bi pročitao sadržaj ćelije &4000 i smestio ga u akumulator.

Iako to u setu instrukcija nije izričito spomenuto, možemo lako da upisujemo brojeve u registar PC: instrukcija JP &3A00 (JUMP) bi slobodno mogla da se zove i LD PC,&3A00. Posle izvršavanja ove instrukcije mikroprocesor će nastaviti normalan rad, ali neće izvršavati sledeću naredbu već onu čiji je broj „nasilno" upisan u PC — u našem slučaju naredbu čiji je početak upisan u ćeliju &3A00. Omogućeno nam je i da dodajemo konstante sadržaju registra PC — takva dodavanja smo već upoznali pod imenom *relativni skokovi*.

Pokazivač steka

Registar SP *(Stack pointer* ili pokazivač steka) ima nekih sličnosti sa registrom PC: njega vrlo retko direktno menjamo (obično se takva operacija obavlja negde na logičnom početku ROM-a), ali su njegove implicitne promene vrlo česte. Ovaj registar pokazuje na neku memorijsku ćeliju ispod koje ima (uslovno rečeno) dovoljno prostora u RAM-u. Kada nam zatreba da privremeno sačuvamo sadržaj nekog **registarskog para** (Z80 je koncipiran tako da se na stek smeštaju samo parovi registara; ne pitajte nas zašto), upotrebićemo instrukciju poput PUSH HL. Tada će SP biti umanjena za 1, sadržaj registra H prekopiran u memorijsku ćeliju na koju pokazuje SP, zatim će SP biti ponovo umanjen za 1 i na kraju će sadržaj registra L biti prekopiran u memorijsku ćeliju na koju ukazuje SP. Potpuno suprotnu proceduru će izvršiti naredba POP HL koja će, kao krajnji rezultat, preneti u registar HL dva bajta sa vrha steka i povećati SP za dva.

Vidimo da je stek na neki način suprotan programu: registrar PC se povećava kada obradimo instrukciju, što znači da program „raste" od nižih ka višim adresama. Suprotno tome, sadržaj SP se smanjuje kada stavimo broj na stek što znači da ovaj „raste" ka nižim adresama. Zahvaljujući tome program možemo da upisujemo od „dna" memorije a za stek da rezervišemo njen „vrh". Kada SP pokaže na zadnju adresu programa, stek i program su se „sudarili" pa nema više memorije za normalan rad; računar bi tada trebao da prijavi grešku, ali je mnogo verovatnije da će nastupiti krah.

Indeksni registri

IX i IY su indeksni registri; neke instrukcije u adresnom delu mogu da imaju oznaku (IX+d) ili (IY+d), što znači da se adresa memorijske ćelije kojoj treba pristupiti dobija sabiranjem sadržaja registra IX (IY) i fiksnog ofseta *(displacement)* **d** koji se nalazi u okviru same instrukcije. Ovakvo indeksiranje je prilično beskorisno i njime se nećemo opširnije baviti; registre IX i IY možete da koristite i kao registre opšte namene (za to su sasvim dobri), ali prethodno proverite da takvom rešenju nisu pribegli konstruktori ROM-a vašeg računara; nije malo kompjutera kod kojih IX i IY imaju neku specijalnu funkciju (npr. pokazivači aritmetičkog steka), tako da ih mašinski programi ne smeju menjati ako planiraju regularan povratak u bejzik! Kod nekih računara (npr. „galaksije") indeksnim registrima možete da pristupate samo ako **dobro** znate šta radite.

Osvežavanje memorija

Registar I se koristi za pamćenje višeg bajta takozvanog *interapt vektora* i ima ulogu samo u interapt modu 2.

Registar R je zadužen za automatsko osvežavanje *(refresh)* dinamičkih memorija — karakteristiku koja Z80 stavlja iznad većine procesora koji za ovu neprekidno potrebnu operaciju zahtevaju dodatni hardver. Ovaj registar se automatski povećava posle izvršavanja svake instrukcije i nekoristan je za programera koji nema hardveraških ambicija.

Registar uslova

Došli smo, najzad, do registra F kome ćemo posvetiti dužnu pažnju. Već smo rekli da mikroprocesor poseduje određen broj flegova koji se koriste za kontrolu toka programa i neke aritmetičke i logičke operacije. Svi ovi flegovi su, fiktivno, svrstani u registar F kako bi njihovo stanje moglo da se sačuva na steku ili alternativnom registru F' (registar F je uparen sa registrom A pa tako postoje naredbe PUSH AF i POP AF kao i naredba EX AF, AF' koju smo već pominjali).

Bit nula F registra zove se indikator prenosa (Carry) i, naravno, obeležava slovom C koje ne treba mešati sa imenom jednog od registara. Ovaj bit se setuje kada se pri binarnom sabiranju pojavi prenos iz najstarijeg razreda (sabiramo, na primer, &33 i &FA; pokušajte) ili kada oduzimamo veći broj od manjeg. Postojanje ovoga flega, kao što ćemo videti, omogućava jednostavno sabiranje i oduzimanje brojeva koji su smešteni u nekoliko memorijskih ćelija. C fleg, isto tako, predstavlja deveti bit akumulatora pri šiftovanjima, pa omogućava prenošenje bitova iz jednog broja u drugi.

Bit jedan registra F se obeležava sa N i interno se koristi pri izvršavanju naredbe DAA. Za programera je nekoristan.

Bit dva — indikator parnosti/prekoračenja — se označava sa P/V (od *Parrity/Overflow)* i ima vrlo složenu funkciju. Kod logičkih operacija i IN instrukcije ovaj fleg je setovan ako je broj binarnih jedinica u rezultatu paran (Parrity). Nevezano sa tim, P/V će biti setovan ako je rezultat prethodnog sabiranja u akumulatoru nekorektan usred prekoračenja pogledajte drugo poglavlje i sabiranje označenih brojeva — P/V fleg u ovom slučaju praktično predstavlja deveti bit akumulatora), a resetovan ako je rezultat korektan (Overflow). P/V fleg se menja i kod instrukcija koje pomeraju blokove memorije, kao i pri promeni sadržaja i registra ali vas te finese, za sledeće duže vreme, neće interesovati.

Bit tri registra F se ne koristi.

Bit četiri — indikator poluprenosa — poznat je pod imenom *Half Carry* i označen sa H ali je za programera nekoristan. Mikroprocesor se služi ovim flegom pri izvršavanju već inkriminisane naredbe DAA.

Bit pet registra F se ne koristi.

Bit šest je, konačno, značajan: obeležavamo ga sa Z *(Zero)* i zovemo indikator nule. On se automatski setuje kada je rezultat neke operacije nula ili kada je test dao rezultat 'True' (tačno) i kao takav je neobično koristan za realizaciju struktura koje odgovaraju bejzik naredbi IF A=B THEN . . .

Znajući da bit sedam svakog bajta nosi znak broja, nećete se iznenaditi kada pročitate da se bit 7 registra F zove indikator znaka *(Sign fleg)* i obeležava sa S. On je setovan kada je rezultat neke operacije (npr. oduzimanja) negativan, tj. ako mu je najviši bit setovan. Ovaj fleg se koristi za realizaciju mašinskih struktura koje odgovaraju bejzik naredbi AF A < B THEN. . .

LD grupa

Prva grupa instrukcija mikroprocesoŕa Z80 kojom ćemo se baviti su LOAD instrukcije koje su zadužene za prenošenje podataka između memorije i registara. Neke od tih instrukcija smo već upoznali: u dosadašnjim primerima smo često koristili nešto poput LOAD A,&20, što je trebalo da znači *dovedi broj &20 u registar A.* Da sada budemo malo egzaktniji: u mnemonici Z80 naredba LOAD se piše kao *LD A,B* i znači *unesi (LOAD=napuni) u A sadržaj registra B.* Pročitajte ovu rečenicu još jednom. *Naredbom LD se sadržaj* ***drugog*** *argumenta prenosi u* ***prvi*** *argument.*

A i B mogu, jasno, da budu raznorazne stvari: osmobitni i šesnaestobitni registri, memorijske ćelije adresirane direktno, indirektno ili i indeksno i, naravno, obične konstante. Bitno je, međutim, da zapamtite da A i B *mogu* da budu svašta, ali *ne mogu* da budu bilo šta: neke instrukcije postoje a druge ne! Tako, na primer, možete da napišete LD A,(&2000) ali *ne možete* da napišete LD B,(&2000). Jasno je da su ove dve instrukcije prilično slične i da bi svaka od njih imala smisla; zašto u registar B ne bismo smeli da dovedemo sadržaj memorijske ćelije &2000? Samo zato što su konstruktori mikroprocesora Z80 bili upućeni na stabilizaciono ponašanje, pa u set instrukcija nisu uveli sve moguće varijante, već samo one koje su smatrali za najkorisnije. Zbog ove nesimetričnosti seta instrukcija opširno ćemo se baviti svim regularnim varijantama na koje će vas docnije podsećati i tabela iz našeg prethodnog umetka.

Osmobitne load instrukcije

Osmobitne load instrukcije se, kao što im ime i govori, bave prenošenjem bajtova (osmobitnih brojeva) između registara i memorije. U tabeli vidimo da je prva moguća naredba ovoga tipa LD reg, reg' gde je *reg* oznaka bilo kog od registara A, B, C, D, E, H i L. To znači da slobodno možemo da pišemo LD A,B, LD E,H, LD H,E i sve slične kombinacije koje nam zatrebaju. U principu bismo mogli da koristimo instrukcije tipa LD A,A ili LD B,B koje, jasno, nemaju nikakvog smisla ali su dozvoljene. Sve instrukcije tipa LD reg,reg' zauzimaju po jedan bajt memorije i ne utiču na stanja flegova kao, uostalom, i sve ostale LD naredbe koje ćemo koristiti.

Instrukcije LD reg,nn predstavljaju prototip neposrednog adresiranja koje smo već upoznali: u bilo koji od gore pomenutih 7 registara koje smo označili sa *reg* primenom ovakvih naredbi se dovodi osmobitna konstanta *nn.* Naredba LD A,&10 će, na primer, dovesti u registar A (akumulator) broj &10 tj. 16. Instrukcije ovoga tipa zauzimaju po dva bajta; u jedan se smešta kod, a u drugi konstanta *nn.*

Primer apsolutnog adresiranja je LD A, (nnmm). Radi se o instrukciji koja zauzima tri bajta memorije; u prvom se nalazi njen kod &3A, a u preostala dva šesnaestobitni broj *nnmm* koji predstavlja adresu neke memorijske ćelije. Dejstvo naredbe je lako objasniti: u akumulator se dovodi sadržaj memorijske ćelije čija je adresa *nnmm.* Protivteža ovoj naredbi je LD (nnmm), A čijom primenom smeštamo sadržaj akumulatora u memorijsku ćeliju čiji je broj *nnmm.* Primetimo da se ove instrukcije *ne navode* kao LD reg, (nnmm) ili LD (nnmm), reg, što znači da jedini registar sa kojim operišemo sme da bude akumulator; vaš asembler će prijaviti grešku ako pokušate da prevedete naredbu kao što je LD B, (&2000)!

Posle apsolutnog, upoznaćemo indirektno adresiranje. Konstruktori Z80 su nam obezbedili instrukcije LD reg, (HL), LD (HL), reg, LD (HL), nn i „specijalne" naredbe LD A, (BC), LD A, (DE) kao i njihove „ogledalske" oblike LD (BC),A i LD (DE),A. Iako smo se jednom bavili indirektnim adresiranjem, uzećemo slobodu da ga opišemo još jednom jer se radi o važnoj stvari koju svaki početnik ne mora odmah da razume. Nećemo se, naravno, baviti svim instrukcijama, već samo jednom: LD reg, (HL).

Pretpostavimo da treba da sastavimo program koji će omogućavati korisniku da pregleda sadržaj memorije. Korisnik našeg programa će, na primer, otkucati broj neke memorijske ćelije (između 0 i &FFFF), a naš će program na ekranu ispisati njen sadržaj. Za sada, jasno, ne umemo da napišemo potprogram za učitavanje adrese ćelije sa tastature i ispisivanje njenog sadržaja na ekranu, ali možemo da pokušamo da napišemo ono što se nalazi između poziva ta dva potprograma:

```
CALL TAST    ; Poziv potprograma koji sa tastature učitava'
             ; šesnaestobitni broj i smešta ga u registar HL.
LD C,(HL)    ; Prenošenje sadržaja tražene ćelije u C.
CALL ISPIS   ; Poziv potprograma koji ispisuje sadržaj C na ekran.
```

Mali oglasi

Ako ne možete da podnesete da drugi nemaju ono što vi imate, objavite svoj mali oglas u „Računarima".

Ako ne možete da podnesete da drugi imaju ono što vi nemate, javite se na neki od malih oglasa u „Računarima".

Ako ne volite da se dopisujete sa „Računarima", svoj mali oglas možete nam izdiktirati preko telefona 011/650-161 svakog radnog dana od 10—14 sati. Mi ćemo vam onda naknadno poslati ispunjenu uplatnicu.

Prva stvar koju treba da uradite je da se odlučite da li želite običan ili uokviren mali oglas.

CENA OBIČNOG MALOG OGLASA do dvadeset reči je 900 dinara. Svaka naredna reč košta još 60 dinara, s tim što ***oglas ne sme da ima više od 50 reči.*** *Adresa oglašivača se ne računa u cenu.*

CENA UOKVIRENOG MALOG OGLASA je 900 dinara po visinskom centimetru, s tim što se mogu zakupiti najmanje 32 slovna znaka. Ako se ne iskoristi čitav prostor u jednom redu, računa se broj redova a ne broj znakova. Za uokvirene oglase preko 5 cm cena je 1400 dinara po centimetru.

Poželjno je da vaš mali oglas počinje sa Prodajem, Kupujem, Držim časove, Menjam . . . ili nečim sličnim što ukratko ukazuje na sadržaj oglasa.

Da ne bi bilo zabune, obavezno naznačite da li želite običan ili uokvireni mali oglas, i zajedno sa tekstom vašeg malog oglasa pošaljite i priznanicu o uplati na adresu redakcije: GALAKSIJA, BULEVAR VOJVODE MIŠIĆA 17, BEOGRAD, sa naznakom „za male oglase u RAČUNARIMA".

Svi mali oglasi koji stignu u našu redakciju do 22. maja do 12 sati biće objavljeni u „Računarima" broj 16, koji izlazi iz štampe polovinom juna.

SPEKTRUM

□ Igre: Winter Games, Kung Fu, Beach Head, Willy 4, Movie, A.O. Yesod, Pyjamarama 4, Blade Runner, Summer Games, Elite, Gunfright, Hyperblaster . . . prodajem za 600 dinara, i drugi kompleti! Saša Radojković, 3. oktobar 166, 19210 Bor, tel. 030/38-182

□ Spektrumovci, za vašu dugu prodajem paket programa po ceni od 1000 din. bez kasete i to: B.C. Quest For Tires, Sir Fred, Winter Games, Arc Of Yesod, Zorro, Gunfright, Beach Head 3, Jet Set Willy 4, Pyjamarama 4 . . . Mojsin Zoran, Al. Spomenice 5/38, 19210 Bor, tel. 030/25-882

□ MOON CLUB — obratite se radi kupovine svih vrsta programa po vrlo povoljnim cenama. Usluga brza i tačna, a kvalitet zagarantovan. Tražite katalog većine programa. Ne navodimo igre jer novi hitovi stalno pristižu. Ne zaboravite, naša adresa je: Strika Jovan, Grčića Milenka 4/a stan 135, 11000 Beograd, tel. 011/444-5093

□ COMPUTER GAME SHOP vam predstavlja komplet ne baš super najnovijih ali najboljih i relativno najnovijih programa svih vremena za vaš spektrum. Komplet+kaseta+poštarina samo 850 din. Isporuku od 24 časa, kao i kvalitet snimka vam garantujemo 100%!! West Bank, Movie, Barry Mc Guigan, Beach Head 3, Blade Runner, Spitfire 40, Battle of Planets, Grumphy Supersleuth. Tel. 011/4887-575

□ Spektrumovci, Biblioteka Softvera svojim članovima omogućava: pristup u biblioteku od 500 programa za spektrum, mogućnost posudbe većine uputstava za programe, pretplatu na najnovije programe koje dostavljamo u roku od 15 dana od pojave na tržištu, pretplatu na klupski kompjuterski časopis New Bit, popuste pri nabavci hardvera i softvera kod najeminentnijih proizvođača . . . i još mnogo drugog. Tražite opširne besplatne prospekte. Branko Čurčić, pp 57, 47300 Ogulin, tel. 047/72-289 ili Milenko Savić, M. Tita 63/1, 31000 T. Užice, tel. 031/24-948

□ Prodajem ZX spektrum 48 K (izvrsno očuvan) sa pratećim priborom i uputstvima i nekoliko raznovrsnih programa. Tel. 034/212-009, Todorov Ratko, Đorđa Andrejevića Kuna 24, 34000 Kragujevac

□ Loto i Sportska prognoza. Tri programa za spektrum ukupno 1500 din. Iskoristite računar, uvećajte svoje šanse za dobitak. Žarko Vukosavljević, p. fah 65, 11070 N. Beograd, tel. 011/197-700

□ Spektrumovci! Mega-prilika! Movie (pravi detektivski film), Three Weeks In Paradise (Pyjamarama 4), Barry McGuigan Boxing, Gladiator (borba gladijatora), Tomahawk (helikopter), Forbidden Planet (čudo velikih majstora), Spellbound (budući klasik), Turbo Espirit (ludovanje po Njujorku), Arc Of Yesod (Nodes 2), Mugsy's Revenge (fantastično), za samo 800 din.+kaseta. Nazovite, pišite, jer mi radimo samo za vas. Marko Premzl, Rade Veina 5, 43260 Križevci, tel. 043/841-682

□ SPECTRUM RAINBOW SOFTWARE vam nudi: Needle, Satancopy 4, Satancopy 3, Turbotape 1, Turbotape 2, Supercopy 1, Superecopy 2, Mastercopy+40 drugih copy programa u jednom kompletu za samo 1000 din. Kompleti od 25 programa 800 din. Posedujemo i sve najbolje programe koji se trenutno nalaze u Jugoslaviji. Tražite besplatni katalog sa 2000 programa. Snimanje direktno iz računara po najnižoj ceni! Uverite se! Mihajlovski Kirčo, Moša Pijade 128, 91300 Kumanovo, tel. 0901/23-800

□ Zodiac Strip — po oceni stručne štampe najbolji erotski program za spektrum. Uzbudljiva igra sa deset devojaka. Cena 1000 din. Žarko Vukosavljević, p. fah 65, 11070 N. Beograd, tel. 011/197-700

□ Spektrumovci! GRUNF SOFT — prodajem najnovije i najbolje komplete za spektrum. Komplet 30: Nomad, Gladiator, Gunfright . . . Komplet 31: Saboteur, Commando, Eadet Porno . . . 12 programa+kazeta+poštarina 1200 din, a sa vašom kazetom 800 din. Za sve informacije obratite se na tel. 041/271-025 ili 430-185

prijem malih oglasa zaključujemo 22. maja u 12 časova

□ Prodajem ZX spektrum 48 K, tastaturu „ines", ZX interfejs 1, ZX mikrodrajv, štampač šinva CP-80, interfejs kempston-S. Tel. 063/855-390, 16-22h

□ Najnoviji programi za ZX spektrum u kompletu i pojedinačno: Movie, West Bank, Winter Games 1 i 2, Sky Fox, Pyjamarama 4, Mikie, Gunfright, Wham The Music Box. Tražite besplatan katalog. Gulabovski Goran, Lihnida 1/1 — 8, 91060 Skoplje, tel. 091/314-185 ili 317-008

□ COMPUTER STUDIO nudi:
Komplet 51: Rambo, Impossible Mission, Sex Mission . . .
Komplet 52: Elite, Mikie, Commando . . .
Uz to mnogo programa, literatura, hardverski dodaci, saveti, pomoć . . .
Četiri godine iskustva, nepovećavanje cena od 1983, preko 10.000 zadovoljnih mušterija iz zemlje i inostranstva, govore nama u prilog.
COMPUTER STUDIO, Starec Bojan, Kosančićev Venac 1A, 11000 Beograd, tel. 011/625-833

□ Rečnik, englesko-srpskohrvatski (oko 1400 najčešće upotrebljavanih reči)+kaseta+poštarina (1300 din). Tel. 011/497-662 od 17—19h, D. Marjanović, B. Jerković 123, 11000 Beograd

□ KRAPINKO SOFTWARE — najnoviji hitovi po najnižim cenama za ZX spektrum. Programi u kompletima ili pojedinačno. Rok isporuke 48 sati. Katalog besplatan. Kiralj Denis, Marka Oreškovića 1/7, 55000 Slavonski Brod, tel. 055/238-866.

□ Spektrumovci! Dosad neviđeni hitovi u kompletu 25: Tomahawk, Beach Head 3, Type Rope, Forbidden Planet, Battle Of Planets, Sky Fox, Barry McGuigan Box, Code Name Mat 2, West Bank, Movie, Blade Runner, Jetset Willy 4, Mega Fruit . . . Cena 700 din.+kaseta. Perić Nenad, Braće Miladinov 12, 37000 Kruševac, tel. 037/33-510

COMET SOFTWARE

Programi za ZX SPECTRUM
- video igre -
- namenski programi -
- uputstva za programe -

BETA BASIC 3.0
sa kasetom,
orginalnim uputstvom,
i poštarinom
cena 1000. din.

garancija za sve usluge

MILOVANOVIĆ LJUBIŠA
Petra Lekovića 57,11030 Beograd
tel. 011/558007 posle 17 h.

□ Prodajem pojedinačno snimljene programe za spektrum. Cena jednog programa je 100 din. Na svakih pet naručenih programa jedan poklanjam! Želite li imati programe snimljene direktno iz kompjutera, pišite za besplatan katalog! Janković Silard, Petefi Šandora 84, 25222 Telečka

□ Spektrumovci, Elite, Rambo, Hacker, Commando, XCL, Nomad, Saboteur, Impossible Mission. Cena jednog kompleta od 10 igara 800 din. Zoran Tomić, Fadila Španca 104/A, 23261 Lukićevo

□ Spektrumovci, veliki izbor programa. Cene 40, 60 i izuzetno 80 dinara. Besplatna uputstva za igre. Pokloni. Razmena. Tražite veliki besplatan katalog za igre. D-2 SOFT, 11420 Smed. Palanka, Pionirska 15 tel. 026/34-051

□ SPEKTRUM YU — SOFT ponovo sa vama. Jedini koji (još uvek) snima direktno iz spektruma, zato je snimak ispravan i posle nekoliko godina. Najnoviji programi, popust za pretplatnike, besplatan spisak. JEREMIĆ NEBOJŠA, RISANSKA 10, 11000 BEOGRAD, tel. 011/643-061

□ Super povoljno! Super programi po super niskim cenama. Super nagrade i super kompleti. Tražite besplatan

□ ROLEX SOFT vam nudi: najnovije programe po pristupačnim cenama u kompletima ili pojedinačno. Sigurno imamo: WS Kung Kong, Movie, Amazon Women i sve što drugi imaju. Najnoviji programi staju 200 a stariji 100 dinara. Spisak je besplatan, a katalog košta 100 din. Rosić Nebojša, Čelebićka 10/4, 11000 Beograd, tel. 011/591-631

□ ADICTIVE SOFTWARE vam nudi najnovije hitove u kompletu ili pojedinačno po najpovoljnijim cenama za ZX spektrum. The Way Of Tiger, Boughdek, F.A. Cup Football, Amazon Women, Green Bereth, Enkrevoball, Fireman, In Action, Yabba Dabba Doo, Ping-Pong, Visitors, Endurances, Beach Head 3, Summer Games 2. Cena kompleta 600 din. Ćirić Aleksandar, Crnotravska 13, 11000 Beograd, tel. 011/661-260

□ Svakog meseca najnoviji ZX spektrum programi u polusatnim kompletima (po 6 programa) za samo 200 din. Tražite katalog sa 1000 programa. Marko Marković, Žrtava fašizma 7/4, 71000 Sarajevo, tel. 525-212

□ Dosadio vam spektrum? Odvojte 1000 din. za kazetu sa TURBO TAPE-om i MEGA bejzikom i imaćete nov kompjuter. Saša Požgaj, Starih omladinaca 4, 41000 Zagreb, tel. 041/565-158

□ Prodajem komplet: DT Supertest, Fighting Warrior, Cokin, Kennedy Approach, Neverending Story, A.C.E. Fight Night, Hard Ball+kaseta (C-45(za 1100 din. Stojanović Branislav, Lipa 25, 11030 Beograd, tel. 011/556-807

□ Danas u vašem gradu svi najnoviji programi na jednom mestu: Ping-Pong, Movie, Summer Games 2, Amazon Women i puno drugih. Za spisak obratite se na adresu: Stevan Šaviјanski, Teodora Pavlovića 6, 21000 Novi Sad, tel. 021/618-590

□ COPY DE LUX — Kopira sve spektrum programe. Objedinjeni Monster i Super Copy u jedan program. Jednostavan rad za početnike i za pirate. Opširno uputstvo sa primerima. Sa kasetom i ptt 1200 din. Jeremić Nebojša, Risanska br. 10, 11000 Beograd, tel. 011/643-061

□ Prodajem komplet 2: Hyperblaster, Ping-Pong, Yabba Doo, West Bank, McGuigan, Gladioator... Komplet 3: Way of Tiger, Goonies, Visitors, Back to Future... Komplet 550 dinara. Raojković Saša, 3. oktobar 166, 19210 Bor, tel. 030/38-182

□ Samantha Fox Strip Poker, Back To Future (prema filmu), Green Beret (Imagine), Way of Tiger (3 programa), Ping Pong (Imagine), Bomb Jack (Elite), Pogo (Ocean), Shrinkin Fireman (Mastertronic), Amazon Women+kazeta+poštarina+uputstvo=3000 din. Šepić Miran, Rastočine Š-3, 51000 Rijeka, tel. 051/510-855

□ SPEKTRUM SOFTWARE STUDIO — izbor od preko 1000 programa — svaki program sa uputstvom — veliki izbor literature, knjiga i originalnih programskih uputstava na engleskom i srpskohrvatskom jeziku. Spisak programa je besplatan, za katalog sa opisom poslati 200 din. Brza i kvalitetna usluga — proverite. PAJNIĆ MIRKO, STRAHINJIĆA BANA 56, 11000 BEOGRAD, tel. 011/188-190 posle 15h

□ 1500 spektrumovih programa, među kojima i najnoviji superhitovi, snimam na vaše kazete (30XC 90) za 16000 din. Tražite katalog (sa naznakom — za megakoplete). Mario Mendeš, Jeretova 8, 58000 Split, tel. 058/553-506 iza 21 h

□ Sve najnovije programe koji se nalaze u Jugi kod nas možete jeftinije nabaviti! 11 programa po vašem izboru=800 din. Snimamo i proveravamo direktno iz spektruma! Katalog besplatan! COMPUTER ART, Vojvode Mišića 19, 21000 Novi Sad, tel. 021/56-542

□ Superkomplet — 18 najnovijih spektrumovih hitova sa prevedenim uputstvima, mapama, poukovima i kazetom za samo 1500 din. BEST BUY SOFTWARE, Mario Mendeš, Jeretova 8, Split, tel. 058/553-506 iza 21h

□ VRHUNSKI GRAĐEVINSKI PROGRAMI za ZX Spectrum: OKVIRI, REŠETKE, ROŠTILJI, PILOTI, DIMENZIONIRANJE, VODOVOD i mnogi drugi. Za radne organizacije i pojedince. Besplatan katalog. Gino Gracin, Kozala 17, 51000 RIJEKA, tel. 051/517-291.

□ BEST BUY SOFTWARE — najnoviji programi iz Engleske, Njemačke, Italije i Danske u kompletima od 90 minuta za samo 500 din. Violent Sex, Inferno, Athlete, Spitfire 40, Driller. Tražite katalog. Mario Mendeš, Jeretova 8, 58000 Split, tel. 058/553-506 iza 21h

□ C jezik 1.1, Blast, Logo 1,2, Beta Basic, Learn Chess i mnogo drugih uslužnih programa sa uputstvima prodajem. Tel. 011/182-101

□ SERVIS ZA KOMPJUTERE ZX SPEKTRUM I C-64. Brzi i kvalitetni popravci, ugradnja reseta i druge usluge. Kovačić Anđelko, VIII Vrbik 33a/6, 41000 Zagreb, tel. 041/539-277

□ Na mađarskom jeziku profesionalni prevodi, uputstva za spektrum: Blast, Devpac, Asembler, Disassembler, Paint Box, SCR 2D Machine, Tasword, Master File, Forth, Beta Basic i Leonardo. Tel. 011/182-101

□ Spektrumovci, niske cene, veliki popusti; najnovji programi. Rok isporuke 24 časa. Ako programe naručite telefonom, 12 časova. Tražite besplatan katalog. Robert Vahtler, Omladinska 39, 55000 Slavonski Brod, tel. 055/236-107

□ ROTKVA SOFT — predstavljamo vam više od 500 programa. Svi programi su snimljeni iz kompjutera i povereni. Popusti stalnim kupcima. Igre april 86: V-visitors, Taffy Turner, F.A. Cup Football, The Way of Tiger, Bomb Jack, Green Berets, Back to the Future itd. Navedene programe imamo! ROTKVA SOFT, Bul. bratstva i jedinstva 20/412, 21000 Novi Sad, tel. 021/398-454

□ TEST INTELIGENCIJE — kroz 180 pitanja dobićete potpunu sliku o svom koeficijentu inteligencije. Cena programa 800 din. Besmrtni POKE katalog — 260 poukova za 130 igara za spektrum. Pošaljite preporučeno 400 din. Tot Tibor, B. Dejanović 1, 21250 Sr. Karlovci, tel. 021/881-650

□ Spektrum — najnoviji i najbolji programi u kompletima. Komplet 54: Turbo Espirit, Spellbound, Jetset Willy 4, Forbidden Planet, Mugsys Revenge, Ping Pong, Spitfire 40, Amazon Women, V-visitors, The Way of Tiger 1—4. Komplet 53: Battle of Planets, Championship Boxing, International Rugby, Movie, Code Name Mat II, West Bank, Cylu, Video Olympics, Rapscallion, Frankenstein 20000, Sky Fox, Yabba Dabba Doo. Jedan komplet+kaseta+PTT 1500 din. Trtica Goran, Stevana Lukovića 9, 11000 Beograd, tel. 011/563-348

□ MACHINE CODE FOR BEGINNERS — sada i Vama data efikasnost mašinskog programiranja. Traka i opširno uputstvo (65 str.) — 2000 din. VRCA MILAN, Zarija Vujoševića 79, 11070 Novi Beograd

□ BLAST!!! — ubrzaće Vaše bejzik programe i do 40 puta (vidi RAČUNARE 14, strana 25). Traka i opširno uputstvo (35 str.) — 2000 din. VRCA MILAN, Zarija Vujoševića 79, 11070 Novi Beograd

□ SPEKTRUMOVCI! Novoformirani Kermitsoft od sada zauvek sa vama! najnoviji programi stalno stižu iz Engleske. Još uvek snimamo programe direktno iz računara i vršimo verifikaciju svakog programa. Požurite! Odavno imamo Ping Pong, Runestone, V-visitors, Green Beret, Knight Rider, Way of Tiger, Dyberun, Bomb Jack. Tražite besplatan spisak na adresu: Jerković Branko, Gročanska 47, 11000 Beograd ili na tel. 011/428-359

□ Spektrum — 12 programa za učenje engleskog+kaseta 1000 din. 25 radioamaterskih programa+kaseta 1000 din. 40 COPY programa+kaseta 1000 din. Tražite besplatan katalog sa preko 1000 programa. Trtica Goran, Stevana Lukovića 9, 11000 Beograd, tel. 011/563-348

□ DYNAMIC PROGRAMMING — specijalni efekti i više od 200 novih komandi u bejziku. Traka i opširno uputstvo (27 str.) — 2000 din. VRCA MILAN, Zarija Vujoševića 79, 11070 Novi Beograd

□ HAPPY SOFT nudi najnovije, najjeftinije programe i literaturu za komodor. Direktno sa engleskih top lista. Nagrade, popusti, besplatan katalog. Victor Karaba, Braće Dronjak 17/50, 21000 Novi Sad, tel. 021/395-257

□ BLAST!!!; DYNAMIC PROGRAMMING; MACHINE CODE FOR BEGINNERS: traka i uputstvo pojedinačno — 2000, za dva programa — 3000, sva tri — 4000 din. VRCA MILAN, Zarija Vujoševića 79, 11070 Novi Beograd

□ Spektrumovci, vama Kaj-soft nudi komplet K-2: Beach Head 3, Rock'N Wrestle, Ping-Pong, Tau Ceti, Cyberun, International Rugby, Back To Future, Dynasty. Komplet+kaseta+PTT=1000 dinara. Kajsoft, Diljska 20, 54000 Osijek

□ Kaj-soft nudi spektrumovcima komplet K-3: Movie, Battle of Planets, Goonies, The Way Of Tiger, Video Olympics, Zoids, Westbank, Barry McGuigan, Thunderbirds, Amazon Women, Strong Man, Cuberun. Kvaliteta je naš moto. Komplet+kaseta+PTT=1000 dinara. Kaj-soft, Diljska 20, 54000 Osijek

□ COMPUTER STUDIO nudi:
Komplet 53: BC Quest, Zorro, Jet Set Willy 3...
Komplet 54: Gunfright, Winter Games, Tau Ceti...
Uz to mnogo programa, literatura, hardverski dodaci, saveti, pomoć... Četiri godine iskustva, nepovećavanje cena od 1983., preko 10.000 zadovoljnih mušterija iz zemlje i inostranstva govore nama u prilog.
COMPUTER STUDIO, Starec Bojan, Kosančićev Venac 1A, 11000 Beograd, tel. 011/625-833

□ COMPUTER STUDIO nudi:
Komplet 55: Pyjamarama 4, Beach Head 2, Jet Set Willy 4, Boulder Dash 2...
Komplet 56: nastavak Hobbita, Thunder-Birds, International Rugby...
Uz to mnogo programa, literatura, hardverski dodaci, saveti, pomoć... Četiri godine iskustva, nepovećavane cene od 1983., preko 10.000 zadovoljnih mušterija iz zemlje i inostranstva govore nama u prilog.
COMPUTER STUDIO, Starec Bojan, Kosančićev Venac 1A, 11000 Beograd, tel. 011/625-833

□ SUN SOFTWARE CLUB — spektrum je tri godine sa vama, zahvaljujući kvalitetu i bogatom izboru više od 1500 programa. Programi su podeljeni u nekoliko grupa: uslužni, sistemski, copy, kao i najnovije igre. Pri naručivanju svi programi su snimljeni direktno iz računara, a svaki pojedinačno i proveren. Sve najnovije igre čije nazive možete naći u drugim oglasima, a koji su trenutno u Jugolsaviji, već posedujemo. Naročite povoljnosti uživaju članovi i stalni kupci. Posebni popusti za veće narudžbine. Kvalitetno i brzo! Veliki katalog na 18 strana je 200 din. a mali katalog programa je besplatan. Majdevec Ivan, Vojvode Mišića 2/5, 21000 Novi Sad, tel. 021/57-988

□ Veliki izbor kvalitetno pojedinačno snimljenih spektrum programa. Za besplatan katalog javite se na 011/436-137 do 15 časova. Miroslav Radosavljević, Braće Nedića 2, 11000 Beograd

□ COMPUTER STUDIO nudi:
Komplet 57! Blade Runner, Spitfire, Adrian Mole...
Komplet 58: Turbo S'Print, Viking, Yabba Dabba Doo...
Uz to mnogo programa, literatura, hardverski dodaci, saveti, pomoć... Četiri godine iskustva, nepovećane cene od 1983., preko 10.000 zadovoljnih mušterija iz zemlje i inostranstva govore nama u prilog.
COMPUTER STUDIO, Starec Bojan, Kosančićev Venac 1A, 11000 Beograd, tel 011/625-833

□ Spektrumovci !!! Komplet 21: YABBA DOO, SPELLBOUND, DUSTMAN, TURBO ESPRIT, MUGSY REVENGE, INFERNO, Alien SWARM, ATHLETE, PAR POSTMAN, TUNELES, DISASTER, STARSHIP ENTERPRISE, GALACTIC AUTOSTOPER, RETUR OF THINFS. Komplet 22: REAL YOU1,2, ADRIAN MOLE 1—4, SKY FOX, JET SET WILLY4, FRANKENSTEIN2000, CODE NAME MAT 2, MEGA FRUIT, CHOPPERX—1, SWORDS AND SORCERY. Cena jednog kompleta 800 din. Radović Branislav Sonje Marinković 14/4, 21000 Novi Sad, tel.: 021/28-682 ili 022/424-824 (vikendom)

□ Mc SOFTWARE! SPEKTRUMOVCI!
Hitovi aprila 1986. u kompletima od 14 programa za 800 din.+kaseta. Rok isporuke 1 dan. KOMPLET 32: Pyjamarama 4 (Three Weeks In Paradise) Winter Games 1,2 (US Gold), Thunderbirds, Riddlers dan, The Arc Of Yesod, Street Hawk (Ocean), Gladiator, Grumphy Grumphley, Gremlins, Zoids, Strong Man, Culy, International Rugby. KOMPLET 31: Zorro (US Gold), Gunfright (Ultimate), XCEL, Wham — The Music Box, Mr. Freeze, Rockman, Jason Gem, 2112AD, Tauceti, Juggernaut, Astro-Clone, Panzadrome, Zenji, Starquake (Wizards Lair 2). KOMPLET 30: Elite, Nomad, Mikie, Transformers (Ocean), SW World, Sub. Striker, Sir Fred, Metabolis Enigma force, Cos. Wartoad, Hyperblaster, Think, Bimbo, Tombola. Zoran Milošević, Pere Todorovića 10/38, 11030 Beograd, tel. 011/552-895

□ Spektrumovci, direktno iz Londona!!! Komplet 23: AMAZON WOMEN, PING PONG, McKENSIE, TITANIC, FRIDAY 13, FINEST HOUR, RUPERT, RUNESTONE, ASHKERON, CANNON BALL, OVERLORDS, FUZZIE WUZZIES, ALL OR NOTHING, SAM SPADE, Komplet 24:

WAY OF TIGAR 1—3, GREEN BARET, V-Visitors, F.A.FOOTBALL, TAFFY TURNER, BOMB JACK, UNCREDIBLE FIREMAN, SOUND MASTER, XENO2, AUSTERLITZ, YUSKOOL DAZE. Cena jednog kompleta 800 din. Radović Branislav Sonje Marinković 14/4, 21000 Novi Sad. tel.: 022/28-682 ili 022/424-824 (vikendom)

KOMODOR

□ KRAPINKO SOFTWARE — najnoviji hitovi po najnižim cenama za komodor 64. Programi u kompletima i pojedinačno. Rok isporuke 48 sati. Katalog besplatan. Kiralj Denis, Marka Oreškovića 1/7, 55000 Slvaonski Brod, tel. 055/238-866

□ Komodror 64 — paket programa: Nexus, Neverending Story, Revs, Kane Ping-Pong, Enigma Force, Staff 6+kazeta=1800 din. Moguća razmjena najnovijih programa. Lasić Hrvoje, Nike Katunara 6, 51000 Rijeka, tel. 051/442-656

□ KOMODOR HARDWARE — Novo iz P.N.P electronica. Dodaci sad i za komodor 64. ROM moduli sa mnogo dobrih programa. Nema više upisivanja sa kasetofona, jer modul postaje sastavni dio kompjutera. Lajt pen, kabel za monitor, EPROM programator, Sentroniks printer interfejs i još mnogo toga. POPRAVCI. Besplatni katalog. P.N.P. electronic, Jeretova 12, 58000 Split

□ TD SOFT — sve za C-64 na jednom mestu. Pišite, nazovite i tražite besplatan katalog. Marković Tomislav, Pačirski put 8, 24000 Subotica, tel. 024/28-888

□ Izuzetna prilika za komodor 64! Prodajem 80 vrhunskih igara za samo 2000 din. Kaseta besplatna. Prvih deset naručilaca dobijaju po 10 besplatnih programa na istoj kaseti. Ilić Mališa, 12307 Tabanovac

□ NOVO ZA KOMODOR 64!!! Profesionalni program za obradu teksta: srpskohrvatski karakteri, mogućnost štampanja svih grafičkih znakova, (lako konstruisanje tabele itd.), mogućnost definisanja 30 proizvoljnih karaktera, kompatibilnost sa Easy Script-om. Sve ovo za printere MPS 801, 802, 803 i VIC 1526. Olah tivadar, Bolmanska 96, 24220 Čantavir

□ Komodor 64 — stalan priliv najnovijih programa za kasete, u paketima ili pojedinačno. Informacije na tel. 011/102-914

□ Komodor 64 — komplet A: Robin of the Wood, Last V8, Yabba Dabba DOO, Mikie, Karateka, Podešavanje glave, Ping-Pong, Staff 2,3,4,5, Imhotep, Yie Ar Kung Fu, Kawasaki, William Wobbler. Komplet B: Kawasaki RR, Yie Ar Kung Fu, Who Dares Wins 3, Commando 1 i 2, Back To Skool, Dragon Skull, Mikie, Staff 2, 3 i 4, ACE, The Young Ones. Komplet+kazeta=1800 din. Svi programi su u TURBO TAPE-u. Ninić Goran Tomislavova 3, 43260 Kruževci, tel. 043/841-870

OBRADUJTE SVOJ RAČUNAR! OBJAVITE MU MALI OGLAS U „RAČUNARIMA"!

□ DALMATIA SOFTWARE vam predstavlja najbolje i najjeftinije programe za vaš Komodor 64: ELITE, OXFORD PASCAL, PIZZA 202, RAMBO II i još mnogo toga. Tražite besplatan katalog. DALMATIA SOFTWARE, Skojevaca 4/VI, 58300 Makarska, tel. 058/613-691

□ Komodor 64 — prodajem najnovije kazetne programe: Monty, Nodes Of Yesod, F.T. Supertest, Elite 2, Back To Future, Treasure Island 2, Staff 4 i 5, Commando 2, Robin Of the Wood, Desert Fox, Alien 8 trodimenzionalan, i ostale. Hajdina Damir, Štajnbergerova 42, 41040 Zagreb, tel. 041/253-338

commodore za sva vremena

BASIC i Simon's BASIC

Jasno, pregledno i potpuno izložen je osnovni jezik Komodora 64 sa preko 150 primera u knjizi COMMODORE ZA SVA VREMENA. I još mnogo toga!

336 strana, 16 × 23 cm, latinica
Cena: 3600 din.
Način nabavke: U svim knjižarama ili direktno od izdavača:
Mikro knjiga
P. O. Box 75, 11090 Rakovica – Beograd

□ Prodajem najbolje i najnovije programe za C-116: Grandmaster, Lawn Tennis, Big Attack, Canoe Slalom, Olympic Skier . . . Iznenađenja. Poklon programi. Mitrović Vukašin, Vojislava Ilića 90, 11000 Beograd, tel. 011/4899-166

□ EKSKLUZIVNO ZA C-64!!! Ako želite najbolje, najjeftinije (30 d), najkvalitetnije presnimljene programe, obratite se BIRC-SOFTU. Na 10 naručenih programa dobivate 1 besplatno. Katalog je besplatan. Grobenski Milivoj — Birc, F. Stareja 10, 42000 Varaždin

□ Komodor 64=hitovi maja: Run For Gold, Comic Bakey Shoping, Yie Ar Kung Fu, Mercenary, Who Dares Wins 3, Soccer 4, sa kasetom 1200 din. Pojedinačno Ping-Pong, Bounder, Rock'n Wrestle. Svi programi u kompletu su TURBO. Dejan Grubar, Stojana Jankovića 6A, 11000 Beograd, tel. 011/561-519

□ L — SOFT. Mnogo novih programa za komodor 64. Besplatan katalog sa najjednostavnijim načinom naručivanja. Levak Nenad, Kumičićeva 14, 42000 Varaždin, tel. 042/40--603

Komodor 64 — prodajem i menjam najnovije programe. Najveći hitovi u kompletu: Sky Fox, Mervenary, Neverending Story, DT Supertest, USA Road Race, Imhotep, Fight Night, Gremlins. Komplet+kaseta+poštarina 900 din. Možete poručiti i neke druge programe. Katalog besplatan. Teofilović Dragoljub, Vojvode Brane 7, 11000 Beograd, tel. 011/422-705 ili 752-753 Aca

□ Komodor 64 — poslovni programi:
— EASY SCRIPT — profesionalna obrada teksta
— WISAWRITE — prof. obrada teksta (datoteke samo na disku)
— PRACTICALC — obrada i sortiranje tabeli, raznih podataka, 22 matematičke operacije itd.
— Statistički paket programa
— HELP 64 — nezamenljiv za ozbiljno programiranje. Svi programi osim pod 4 su sa kvalitetnim srpskohrvatskim uputstvom. Jedan program 1500 din. svaki sledeći 1300 din. Milanović Mladen, Lenjinova 2C, 71000 Sarajevo, tel. 071/210-834

□ Boulder Dash 4, Yabba Dabba Doo, Superman 2, Turbo Pizza 202, Space Pilot 2, 3, American Road Racex, Back To The Future komplet+Kaseta=1350 din. Moguća i razmena. Perunović Vladimir, blok 6 Al, ulaz I, 81000 Titograd, tel. 081/11-379

□ Komodor 64 — preko 1700 prošlih, sadašnjih i budućih hitova koje možete nabaviti pojedinačno ili u kompletima uz niz popusta i niske cene. Komplet R: GOONIES, FOURTH PROTOCOL, LITTLE COMPUTER PEOPLE, PING PONG, ELITE II (Mercenary), MISTICAL MISSION, GYROSCOPE II, TIME TUNNEL, AMAZON WOMEN, PRO BOXING, TOMMY AND JERRY, SPIDER THE FLY . . . Komplet S: DORIATH, SATANS GROTTY CHRISTAMS, UHG, TALES OF THE CAT, COMPANU AND CO., WILLY TEX, LABIRINTO, INTHERTIANCE, TRIBLE JUMP, TAXI, METRANAUT, IRIDIUM, ROCK'N WRESTLE . . . Komplet T: STRONG MAN, ELECTRA GLIDE, SONG ELECTRA, TROOPER CROUPER, T.I.P.O., CAMELEON, NUDIST CAMP, THE BLACK CELEBRATION, LITTLE COMPUTER PEOPLE, SABOTEUR, POLE POSITION II, CASTLE WOLWENSTAIN II . . . Svaki komplet — 1500 din. Komplet svih 1700 programa (uključujući i gore navedene) možete dobiti po ceni od samo 200000 din.+kasete (vaše ili moje). Detaljna obaveštenja tražite na tel. 015/25-772 ili na adresu: Branko Vrhovac, Moše Pijade 4, 15000 Šabac

□ Komodor 20, 16, +4, 64, 128 programi. Tražite besplatan katalog. Đerman Šandor, Rade Konačra 23, 23000 Zrenjanin. RTTY, CW konačno i za C-64. Informacije na adresi: Kovač Jani, Klavniška 13, 69000 Murska Sobota

□ Turbo P.S. — novi programi za vaš C-64, korisnički programi, igre, Copy programi, upute i saveti za početnike. Katalog besplatan. Koštrun Marijan, Try I. Kukuljevića 14, 41000 Zagreb

□ Rasprodaja! Komodorci, jedan ogroman paket+kaseta, sve hitovi, staje samo 999 din. Može i posebno. Nazovite ili pišite: 021/812-309, Grozdić Dalibor, Žarka Zrenjanina 3, 21220 Bečej

□ Komodor 64 — komplet 40 najboljih korisničkih programa sa svih područja+kaseta+poštarina=3999 din! Pogodno za škole, radne organizacije i pojedince! Prodaja i zamena najboljih igara! Krstin Zoran, Prvomajska 2, 69000 Murska Sobota, tel. 069/24-526 (poslepodne)

□ Komodor 64 — komplet od 100 igara za samo 1000 din. Besplatan spisak. Exploading Fist, Commando, Skol Daze . . . Brza i kvalitetna isporuka. Sajčić Đorđe, Boška Savića 7, 37240 Trstenik

□ Komodor 64 komplet 1: Winter Games, Petak 13, Elite, Aligata, Superman, Rambo 2. Komplet 2: Commando, Rocky Horror Show, Rambo 2, Robin of Wood, Elite, Who Dares Wins 2. Komplet+kaseta 1000 din. ANE SOFT, Kosturska 77, 91400 Titov Veles, tel. 093/25-387, 20-334

□ HAPPY SOFT nudi najnovije, najjeftinije programe i literaturu za komodor. Direktno sa engleskih top lista. Nagrade, popusti, besplatan katalog. Victor Karaba, Braće Dronjak 17/50, 21000 Novi Sad, tel. 021/395-257

□ Najnovije igre za C-64 u kompletu. Komplet A; Robin of Wood, Yie Ar Kung Fu, Rally Afric, Madonna, Falkland's Thunderbirds Bounced (neviđena borba), Touch Foozball (ragbi), Castle Of Creet, Hypersport 2. Cena kompleta sa kasetom iznosi 1500 din. Programe možete naručiti na adresu: Petrović Milan Jovana Stojisavljevića 9/11, 11080 Zemun, tel. 011/195-758

commodore za sva vremena

MAŠINSKO PROGRAMIRANJE, MAPA MEMORIJE I ROM RUTINE

Kompletan kurs mašinskog programiranja sa detaljnim objašnjenjem svake memorijske lokacije dat je u knjizi COMMODORE ZA SVA VREMENA. I još mnogo toga!

336 strana, 16 × 23 cm, latinica
Cena: 3600 din.
Način nabavke: U svim knjižarama ili direktno od izdavača:
Mikro knjiga
P. O. Box 75, 11090 Rakovica – Beograd

□ Komodor 64 — najnoviji sportski programi: Winter Games (5 igara)+American Road Race+speed King+International Basketball+Slapshot Hockey+McGugan Champion Boxing+kaseta+poštarina=1300 din. Plaćanje pouzećem. Predrag Stjepčević, Bulevar revolucije 290/V, 11000 Beograd, tel. 011/417-064

□ ROMANTIC ROBOT SOFT — programi za C-64. Svi razbijeni. Pristupačne cene, besplatan katalog. SCOTCH kasete. COPY PROGRAM! — NPR. Turbo 250. Uz svaku narudžbu Turbo 1002. Spiskovi najboljih u Računarima 11, 12, 13, 14. najnovije: Human Race, Dynamite Dan, Kapriolen, McGuigans Box, Enigma Force, Nodes Of Yesod, Black Wuche, Wizardy, Fight Night, Underwurlde. Svaki mjesec i do 30 novih hitova. Žuljević Kemal, Braće Kosorića 13, 72220 Zavidovići, tel. 072/874-441

□ Komodor 64: za sve ljubitelje muzike komplet najboljih muzičkih programa (Funkyy Drummer, Synth Sample . . .). 7 programa+kaseta+poštarina=1000 din. Plaćanje pouzećem. Predrag Stjepčević, Bulevar revolucije 290. ., 11000 Beograd, tel. 011/417-064

□ Za komodor 64, VIC 20, C 128 prodajem konektore (priključke) za kasetofon, user-port, džojstik, te produžni kabl za džojstik, reset taster . . . Cijena pojedinačnog programa po izboru za C-64 samo 25 do 35 dinara. Jeftina literatura i uputstva za programe. Razmjena programa i literature. Besplatan katalog. Odgovaram svima. Tel. 074/832-832, Radovanović Milorad, Radojke Lakić 3, 74400 Derventa

□ Komodor 16, 116, +4. Veliki izbor programa. Novi hitovi: Bejzbol World Cup Fudbal, Berks, Rockman, katalog besplatno. Ljubisavljević Dragan, 3. oktobar 302/6, 19210 Bor, tel. 030/33-941

□ SERVIS ZA KOMPJUTERE ZX SPEKTRUM I C-64. Brzi i kvalitetni popravci, ugradnja reseta i druge usluge. Anđelko Kovačić, VIII Vrbik 33a/6, 41000 Zagreb, tel. 041/539-277

□ KOMODOR SOFT — biblioteka od oko 2000 programa. Tražite besplatan katalog. Bićete ugodno iznenađeni. Aušperger Nada, Vladimira Gortana 58, 42000 Varaždin

□ KOMDORCI! Najjeftinija i najbolja ponuda hardvera i softvera. Hardver: uređaj za direktno presnimavanje (2000), reset (500) i mnogo drugih uređaja po vašim i našim šemama. Softver: Fight Night, Back to Future, Commando 3, Night Shade, DT Supertest, Neverending Story, Staff 5, The Rats, Funky Drummer, Karateka 3D+kasete=1800 dinara i mnogo pojedinačnih programa. Lazić Miroslav, Slavka Rodića 14-b, 72000 Zenica, tel. 072/23-983 i 072/38-131

□ JOKER SOFWARE nudi za vaš C-64: Little computer peple, Transformers, Dragan Skull, Robin of the Wood, One Man Bans, Sys Droid. Naručite besplatan katalog na tel. 021/398-245, Jocić Dragan, Put bačkopalanačkog odreda 20/2 stan 21, 21000 Novi Sad.

□ Komodor 64: Back to Future 2, Enigma Forrce 2, Red Arrows, Who Dares Run for Gold, Yie Ar Kung Fu, Soccer 4, Flight Simulator 3, Soho Jumper. Programi+kaseta 1300 din. Pojedinačno: Supertest, Elite 2 . . . Mladen Jevtić, Crvenih hrastova 16/7, 11030 Beograd, tel. 011/544-428

commodore za sva vremena

HARDVER i KONSTRUKCIJE

Potpuna električna šema Komodora 64 sa objašnjenjima načina rada i uputstvima za gradnju interfejsa, modema, EPROM programatora i kartridža, data je u knjizi COMMODORE ZA SVA VREMENA. I još mnogo toga!

336 strana, 16 × 23cm, latinica
Cena: 3600 din.
Način nabavke: U svim knjižarama ili direktno od izdavača:
Mikro knjiga
P. O. Box 75, 11090 Rakovica – Beograd

□ CBM 64 IMPOSSIBLE AND MIRACLE SOFT predstavljaju vam komplet igara čije su recenzije upravo izašle u engleskim časopisima. Komplet 22: Karateka 2, Mercenary, 3 Elite (sa celokupnom opremom+30 hiljada kredita), Little Computer People, Nexus, Boulderdash, Zorro, Kung Fu Master, Nude girl (5 golih devojaka iz strip pokera). Komplet možete naručiti na adresu: Nenad Skendžić, Lenjinova 85/6, 21205 Sremski Karlovci, tel. 021/881-909 (Robert Ranković)

□ DX EUCALIPTUS vraća se sa najnovijim hitovima i super ludim paketima (Yie Ar Kung Fu, Captive, The Rats . . .) Sve to u besplatnom katalogu. Devčić Zoran, Nikole Tesle 12, 51440 Poreč, tel. 053/33-836

□ Izmene na štampačima KOMODOR 1526 i MPS 802 uz prateće programe omogućiće vašem KOMODOR 64/128
— kreiranje sopstvenih znakova i kompletnih setova slova
— proizvoljan raspored na tastaturi
— pisanje i štampanje teksta sa izborom seta po želji (ćirlica, nemački, matematički simboli itd.)
— hardcopy ekrana visoke rezolucije iz SIMON's BASICa
Tel. 011/331-306 i 426-422

□ BESMRTNI POKE KATALOG — 280 poukova za 138 igara. Komodorovci, otkrijte tajnu omiljenih igara. Za katalog poslati preporučeno 400 din. Tot Tibor, B. Dejanović 1, Sr. Karlovci, 21205, tel. 021/881-650

□ Komodor 64 — najnoviji programi za samo 50 din. (Nodes of Yesod, forth Protocol, Goonies) i mnogo drugih. Besplatan katalog. Moguća razmena. Šafranić Alan, Hrašće 206, 41020 Novi Zagreb

□ Za vaš komodor prodajem paket od 15 programa za 500 din. i to: Beach Head 2, Match Piont, Zorro, Hyper Sports, Fist, Basketball 2, Olympics . . . Mojsin Zoran, Al. spomenice 5/38, 19210 Bor, tel. 030/25-882

□ Spektrum — profesionalni prevodi: Napredni mašinski jezik 1500 d, ROM disassembley 1500 d, Mašinski jezik za apsolutne početnike 1300 d, Spektrum priručnik 800 d, 50 tajni bejzik programiranja 500 d, Devpac 3600 d, Devpac 3 na kaseti 500 d, Mega bejzik 500 d, Mega bejzik na kaseti 500 d. Tražite besplatan katalog. Trtica Goran, Stevana Lukovića 9, 11000 Beograd, tel. 011/563-348

□ Komodor 64 — komplet od 15 programa+kaseta+poštarina+pokloni=1000 din. Programi su: Commando, Elite, Hacker, Pyjamarama 1, 2 i 3, Boulderdash 1, 2 i 3, Skool Daze, Kremenko . . . Popović Tihomir, Vojvode Stepe 10, 32000 Čačak, tel. 032/33-69

□ Komodor: prodajem komplet: Hypersports, Hyper Olympics, Hacker, Exploding Fist, International Basketball 2, Squah, McGuigan, Robin, Beachhead 2 . . . i još 10 igara za 450 din. Radojković Saša, 3. oktobar 166, 19210 Bor, tel. 030/38-182

□ PIZZA SOFT komodor programi: Commando, Exploding Fist, Beach Head 2, Elite, Hacker, Zorro, Rambo 2 . . . Cene niske. Isporuka brza. Besplatan katalog. Rusovan Milan, Koste Racina 11, 21000 Novi Sad, tel. 021/316-895

□ Komodor 64 — profesionalni prevodi: Reference Guide 1700 d, Mašinski jezik 1300 d, Mašinski jezik za početnike 1400 d, Priručnik od C-64 1300 d, Kako da programirate C-64 800 d, Disk sistemi i štampači 900 d, Grafika i zvuk good, Matematika na C-64 1000 d. Simons basic 700 d, C-64 basic 700 d, Practicalc 800 d, Easy Script 400 d, Multidata 400 d, Vizawrite 400 d, Pascal 400 d, Help 400 d, Graf 400 d. Isporuka za 24 časa. Trtica Goran, Stevana Lukovića 9, 11000 Beograd, tel. 011/563-348

□ COMMODORE 64: i ovog meseca kompleti najnovijih programa. Komplet A: FOURTH PROTOCOL, GOONIES, STAFF 5, NEVERENDING STORY, NODES 2, DT SUPERTEST, NIGHT SHADE, SPACE PILOT 3, RATS, KARATEKA. Komplet B: WORLD CUP 2, SABOTEUR, LAST V8, TOCH FOOTBALL, YIE AIR KUNG FU, CASTLE WOLFENSTEIN, MERCENERY, FARM SONG, COMIC BAKERY, RUN FOR GOLD. Komplet+kaseta=1500 din. Katalog besplatan. Milan Stamenović, ul. Vladimira Nazora 77, 18300 Pirot, tel. 010/24-382.

□ KING'S CLUB vam nudi najbolje i NAJJEFTINIJE programe za C-64, ELITE, AMERICAN ROAD RACE, MC. GUIGAN BOXING, SPECTRUM SIMULATOR i mnoge druge. Katalog besplatan. Denis Paliska, Slaviše Vajnera Čiče 9, 51000 Rijeka, tel. 051/31-028

□ MPS-802: paket za profesionalnu obradu teksta: EPROM sa latinicom i prilagođeni program. Kesler Viktor, Rumenačka 106-1, 21000 Novi Sad, tel. 021/334-717

□ Razdelnik DVATASET za priključivanje dva DATASETA i presnimavanje programa. Novo: DVATASET+presnimavanje u oba pravca, verifikacija, ugrađen zvučnik, dva preklopnika, solidna konstrukcija. Kesler Viktor, Rumenačka 1061, 21000 Novi Sad, tel. 021/334-717

AMSTRAD

□ Amstrad — profesionalni prevodi: Priručnik CPC 464 1300 d, Mašinski jezik za početnike 1300 d, Locomotiv basic 1300 d, Devpac 700 d, Masterfile 700 d, Pascal 700 d, Tasword 400 d. Isporuka za 24 časa. Trtica Goran, Stevana Lukovića 9, 11000 Beograd, tel. 011/563-348

□ Komplet vrhunskih programa za amstrad: Formula 1, Arabian Nights, Relly 2, Match Point, Dam Darch, Project Future, Finders Keepers, Air Wolf, 3D Grand Prix, Beach Head. Kazeta+poštarina 1400 din: Marin Fulgosi, Savska 8, 41000 Zagreb, tel. 041/442-056

□ CAPTAIN SOFTWARE — najnoviji programi za amstrad CPC 464. Pojedinačno i kompleti uz niske cene. Tražite besplatan katalog. Captain Software, Margaretska 3, 41000 Zagreb, tel. 041/438-097 od 18—20h

□ Za amstrad prodajem paket od 10 programa za 2500 din. i to: Winter Games, Zorro, D.T. Supertest, Yie AR Kung Fu, Bruce Lee . . . Mojsin Zoran, Al. Spomenice 5/38, 19210 Bor, tel. 030/25-882

□ AMSOFT YU predstavlja najnovije CP/M programe. Jedino kod nas možete nabaviti Prolog, Lisp, Cobol, Algol, Wordstar, 3.30, Datastar, Multiplan, C-kompajler, Turbo Pascal. Najnovije igre: 3D Time Trek, The Key Paccor, The Frize, Sorcery plus, Highway Encounter, Myrodins flight Simulator. Posebna pogodnost: kompleti programa na kazetama ili disketama (10—20 programa na kazeti 2800 din.) Velika ponuda ostalih uslužnih programa na kazeti ili disketi. AMSOFT YU, Trg republike 4, 41000 Zagreb, tel. 041/315-478 ili 270-777

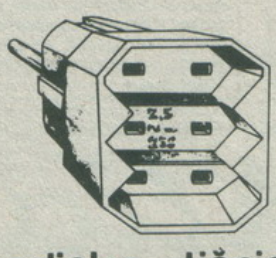
□ Amstradovci, veliki izbor jeftinih programa. Jedan komplet+kaseta 1400 din. Može i pojedinačno. Tražite besplatan katalog. Matić Vladan, Ive Lole Ribara 38, 11318 Miloševac

□ Kupujem programe CW, SSTV, STV, RTTY i sve iz radioamaterske oblasti za CPC 464. Ponude slati na adresu: Lazar Perišić, Narodnog Fronta 14, 21000 Novi Sad

ATARI

□ EMPELE ATARI CLUB — najnoviji i najbolji programi za atari 600/800 XL. Jedan bejzik program 100 din., a mašinski program 200 din. Igor Petrašeski, Emil Zola 7/23, 91000 Skopje, tel. 091/216-476

NOVO!



NOVO! NOVO! NOVO!
TRODJELNA UTICNICA

RJEŠAVA PRIKLJUČAK VIŠE MALIH POTROŠAČA SA PLOSNATIM UTIKAČEM SNAGE UREĐAJA DO 3x500 W. NPR: TV, VIDEO, HI-FI LINIJE, GRAMOFON, KOMPJUTORI, SUŠILO ZA KOSU, BRIJAĆI APARAT, FIGARO ITD. CIJENA 450 D/KOM.

TOMISLAV ŽIVKOVIĆ
41000 ZAGREB
BARČIĆEVA 6/2

trodjelna utičnica

▫ ATARI ST 520/+ kupujem, razmenjujem, prodajem programe i literaturu Nenad, tel. 011/673-733

HARDVER

▫ Prodajem spektrum 48 K sa tastaturom TREND, oroginalnom spektrum tastaturom, kempston interfejsom, džojstikom, pratećom literaturom i 25 kaseta sa programima. Može i pojedinačno. Prednost imaju kupci iz Sarajeva. Markušević Snežana, Lenjinova 67/IV, 71000 Sarajevo, tel. 071/612-438

▫ Prodajem galaksiju 8/6K sa originalnom kutijom i ispravljačem. Vidaković Dražen, Nodilova 5, 41000 Zagreb, tel. 041/416-120

▫ Prodajem komplet čipova za galaskiju 6+8K, memorijsko proširenje 32K (sa konektorom), programrator EPROM-a sa Textool podnožjem i konektorom, najboljem ponuđaču. Berberski Žarko, Drenovačka 2, 11060 Beograd tel. 011/772-203

▫ Prodajem komodor 64+dve palice+kasetofon+50 najnovijih programa za 12 starih miliona. Milojević Ivan, Staro naselje 6, 38218 Leposavić

▫ Prodajem amstrad 6128 sa zelenim monitorom i dodatnom disk jedinicom, i komodor 128 sa flopi diskom 1571. Novo i ocarinjeno. Tel. 011/699-011.

▫ SERVIS ZA KOMPJUTERE C-64 i ZX SPEKTRUM. Brzi i kvalitetni popravci, ugradnja reseta i druge usluge. Kovačić Anđelko, VIII Vrbik 33a/6, 41000 Zagreb, tel. 041/539-277

▫ Prodajem spektrum 48K, verzija 4B, sa profesionalnom tastaturom, reset prekidačem, zvukom na TV, video-monitor izlazom, ZX printer, programabilni kalkulator TI-59. Tel. 021/27-530 od 20 do 22 časa.

▫ ACORN BBC B prodajem. Standardni ISO-Pascal (korišćen u „Radionici logičkih igara"), mikro-Prolog, asembler i BASIC — svi u ROM-u, monohromatski monitor Philips, svi kablovi, kompletna originalna dokumentacija i svi priručnici za jezike, sve na engleskom. Samo zajedno 350.000 din. Jurkovič, Rozmanova 13, 61000 Ljubljana

▫ Prodajem kompjuter Spektra Video 328 sa 32K ROM i 80K RAM, Micro Soft BASIC, profesionalna tastatura. Uz kompjuter prodajem kasetofon, kertridž, kasete sa originalnim programima i literaturu. Čeda Dimitrijević, Medakovićeva 98, stan 82, 11000 Beograd, tel. 011/468-270

▫ Prodajem rubni konektor za spektrum (2800 din), komplet za proširenje RAM-a spektruma sa 16K na 48K (ili 80K 11000 din), te čipove: 4116, 4164 (1200 d), 4416 (4000 d), Z80A CPU, Z80A PIO, 8255, 6116, 2716, 2732 (2400 d), 2764 (2800 d), 27128 (3500 d), AY 38910, AY 38912 (5800 d), 6502, 6522 (4600 d), 1488, 1489 (1200 d), LM 1889 (4200 d), ABC 0804, ZN 427, ZN 428 (5600 d), TEXTOOL-28 (5900 d) i ostale CMOS i TTL čipove. Brzo i kvalitetno servisiram računare spektrum i komodor (C-64, C-16, C-116, C-4). N. Četković, J. Leskovara 1, 42000 Varaždin, tel. 042/38-56

▫ Matrični printer Seikoša GP 50s sa interfejsom za spektrum, prodajem ili menjam za mikrodrajv. Petković Branislav, Jurija Gagarina 122/44, 11070 Novi Beograd, tel. 011/160-694

▫ Prodajem amstrad 6128 sa zelenim monitorom i QL sa 64K RAM-a. Proširenje memorije od 512K za QL, novo, ocarinjeno. Miodrag Drenovčanin, Lenjinova 1/II, 22320 Indija, tel. 022/52-943

▫ SPECTRUM HARDWARE — KEMPSTON INTERFEJS (KEMPY), REDOSTIK PALICA (REDY). CIJENA KOMPLETA 15000 din. EPROM programer, Megarom, P.N.P. ROM (prepravljeni ROM), lajt pen, Centroniks interfejs, I/O port, A/D i D/A konvertori, jednostruki i dvostuki interfejsi za palice, audio pojačalo, izlaz za monitor, literatura, usluge printera, programiranje EPROM-a, savjeti i POPRAVCI. Besplatan katlog na adresu: P.N.P. electronic. Jeretova 12, 58000 Split

▫ Prodajem monohromatski (zeleni) monitor Filips (20 MHz, 820 linija) tonski, 40/80 znakova, neraspakovan, ocarinjen, Kvikšot II i diskete 5,25. Tel. 011/434-695.

▫ Prodajem Ejkorn elektron, povoljno. Tel. 011/328-957

▫ Prodajem Alphacome 32, printer za ZX spektrum sa jednom rolom termalno papira. Ivčević Zdravko, Vlaška 83, 41000 Zagreb, tel. 041/445-549 poslije 15h

▫ Prodajem komodor 64 kasetofon, palice, Turbo Tape kertridž, ili menjam za Casio PB 770 ili Sharp PC 1500. Jovanović Aleksandar, Georgija Jakšića 13, 15300 Loznica, tel. 015/82-406

▫ Prodajem komodor 64, kasetofon, džojstik, literaturu i programe. Omerbašić Dževad, S. Karića 14, 74470 Odžak, tel. 074/770-592

▫ Prodajem računar „galaksija", 8K ROM, 6K RAM, kvalitetno urađen, u tvorničkoj kutiji. Cijena 35000 din. Mandarić Branko, Kralja Tomislava 47,54400 Đakovo, tel. 054/844-756

▫ Komodor 64, 128 — uštedite novac! Umesto skupog komodorovog kasetofona kupite original interfejs za svaki običan kasetofon. Uređaj u minijaturnoj kutiji sa kablom, priključcima i uputstvom 3600 din. Profesionalni kvalitet! Garancija jedna godina. Slobodan Šćekić, Bulevar 23. oktobra 87, 21000 Novi Sad, tel. 021/59-573

▫ Komodor 64, 128 — naš masterfork omogućava priključenje dva kasetofona istovremeno? Presnimavanje svih i zaštićenih programa 70% brže! Moguće kopiranje cele kasete odmah! 2600 din. sa uputstvom. Garancija jedna godina! Slobodan Šćekić, Bulevar 23, oktobra 87, 21000 Novi Sad, tel. 021/59-573

▫ Povoljno prodajem VIC-20 sa programima. Tel. 018/52-708 posle 16h

BBC

▫ BBC-ELEKTRON! Velikiizbor ROM-ova, idealno za RAM proširenja. Pavlović Ilija, Vojvode Stepe 293/38, 11000 Beograd

prijem
malih oglasa
za broj 16
zaključujemo 22. maja
u 12 časova

Literatura

▫ KOMODOR 64 — najkvalitetniji profesionalni prevodi: Programmers Reference Guide — 1750 d. Mašinski jezik za početnike — 1550 d, Grafika i zvuk na C-64, Umjetnost grafike na C-64, Basic priručnik, Simons basic — sve po 1150 d. i Pascal — 800 din. Na višestruke narudžbe popust 10%! Duško Bjelotomić, Centar 1, 54550 Valpovo, tel. 054/82-665 ili 041/683-141

▫ AMSTRAD CPC 464 — odabrani profesionalni prevodi: Uputstvo za rad na amstradu — 1650 d. Locomotive basic — 1550 d, Mašinsko programiranje za početnike — 1450 d, Uputstva za Devpac i Tasword po 1350 d, Grafika i zvuk na CPC 464 — 1350 d. Na višestruke narudžbe popust 10%! Duško Bjelotomić, Centar 1, 54550 Valpovo, tel. 054/82-665 ili 041/683-141

"KOMPJUTER BIBLIOTEKA"

vam predstavlja novu biblioteku kompletnih uputstava za usluzne programe:

1. WORDSTAR CP/M 3.0 (2.000)
upoznajte ovaj fenomenalan program, koji vam omogucava vise nego sto ste ocekivali

2. SUPERBASE (1.200)

Na vas misli
i iznenadjuje vas:
KOMPJUTER BIBLIOTEKA
FILIPA FILIPOVICA 41,32000
CACAK, tel. 032-31-20

KOMODOR 128 / KOMODOR 128
izasao je iz stampe:
"PRIRUCNIK ZA KOMODOR 128"

Na jednom mestu je detaljno objasnjen rad u sva tri moda: C128, C64 i CP/M. Kvalitetna stampa, plastificirani povez.
------- CENA 2.500 din -------
Takodje se mozete pretplatiti na knjige koje su u pripremi:
1. "PROGRAMER'S GUIDE" (3.000)
2. "INTERN 128" (3.000)
3. "CP/M PLUS UPUTSTVO" (2.500)
4. "UPUTSTVO ZA 1571" (2.000)

KUPCI PRIRUCNIKA UZIVAJU POPUST NA BUDUCA IZDANJA OD 20%

"KOMPJUTER BIBLIOTEKA" FILIPA FILIPOVICA 41, 32000 CACAK
telefon 032 - 31 - 20

SPEKTRUM: Profesionalni prevodi za programere u masinskom kodu: MASINAC ZA POČETNIKE (1.200), DISASEMBLIRANI ROM (1.400), NAPREDNI MASINAC (1.400). U kompletu (3.600). Uputstva za uslužne programe: DEVPAC, MEGA, BASIC, BETA BASIC, ARTIST, MELBOURNE DRAV, MONITOR DISASEMBLER, EDITOR ASEMBLER, FIFT, QUILL, TASWORD i LEONARDO. Pojedinačno (500). U kompletu (34.000). Uputstvo za ELITE (900). Isporuka za 24 časa. „KOMPJUTER BIBLIOTEKA" FILIPA FILIPOVIĆA 41. 32000 ČAČAK, telefon 032-31-20.

RAZNO

▫ Projektujem sve vrste elektronskih uređaja, hardver i softver za mikroprocesore MC6800, Z80 i MC 68000. Zoran Kovačević, dipl. inž., Ž. Popovića 45, 15000 Šabac, tel. 27-617

▫ Zaštitite svoje računalo od praha, prekrivači od kvalitetnog skaja, po ugodnim cenama: C128 — 2000, C-64, amstrad — 1500 din, za ostali hardver navesti mjere. Isporuka brza. Peter Rotovnik, Aškerčeva 11, 63325 Šoštanj, tel. 063/881-146

▫ Prodajem printer border VC 1520, časopise Vunx Chan, E10, MT i Electronic po ceni od 450 din. po komadu. Meštrov Dragan, Rapska 26—1, 41000 Zagreb

▫ Prodajem literaturu za komodor 128, komodor 64: Sve o modemima, robotici, umjetnoj inteligenciji. Ujedno prodajem disk drajv, VIC 1541, te uređaj za zvuk Voice Master za C-64. Nikočević Miodrag, Matije Gupca 64, 41210 Zabok, tel. 049/21-272

▫ ŠARP PC 1500: programi, literatura o mašinskom jeziku (skripta 1, 2 i 3) i hardver. Eksterna memorija do 28 KB za bejzik. Kesler Viktor, Rumenačka 106-1, 21000 Novi Sad, tel. 021/334-717

▫ DŽOJSTICI
Vrlo kvalitetne džojstike sa originalnim džekom, 4+4 pravca, okidanje sa vrha palice, za komodor, amstrad, atari i spektrum, možete dobiti za svega 3500 din. Za spektrumovce posebna ponuda. Kako da priključite džojstik bez interfejsa i uštedite do milion starih dinara?! Spektrumovci, slušajte i vi zvuk svojim najomiljenijih igara preko televizora (kao na komodoru). Komplet već pripremljenih delova sa detaljnim uputstvom za ugradnju u spektrum — 1400 din., sa ugradnjom — 2000 din. ERF modulator 5 i 12 V 2000 din. Stojković Dušan i Goran, Trogirski trg 2, 37000 Kruševac, tel. 037/29-550

▫ Prodajem ZX81 1K sa pripadajućom opremom, 50 programa. Quicksave (20 puta ubrzava snimanje i tačnost). Prodajem nov elektronski voltmetar (1000 d), kupujem elektronske komponente. Đeneš Čaba, Ž. Zrenjanina 20, 23272 Novi Bečej

▫ Profesionalni mrežni filteri za otklanjanje svih električnih smetnji i zaštitu kompjutera. Profesionalni kompjuterski rečnici, srpskorvatski-engleski-nemački, i obrnuto, preko 3000 reči! Tel. 021/27-530 od 20 do 22 časa.

▫ Šeme i literaturu za sve kompjutere prodajem. Katalog besplatan. Zvonimir Vistrička, Svačićev trg 2, 41000 Zagreb, tel. 041/440-521

▫ Izrađujem profesionalne kutije, montaža pločice bez vijaka, Evropa format. Lim galvansko zaštićen i obojen plastik bojom. Više dimenzija prema katalogu. Izrađujem i druge profesionalne uređaje: termometri, signalizatori, dvopoložajni regulatori, laboratorijski ispravljači, ispravljači za stanice. Za sve aparate dajem garanciju. Informacije i besplatan katalog. Simonič Peter, Debro 41, 63270 Laško

▫ Servisiram računare KOMODOR 64 i ZX SPEKTRUM. Izrađujem EPROM programatore za iste računare. Programiram sve vrste EPROM-a. Barbarić Dinko, Diljski odred 21, 55000 Slavonski Brod, tel. 055/236-702

računari su uvek aktuelni

da li ste sigurni
da vam ono o čemu smo pisali juče
neće biti potrebno već sutra?

Računari 3

Računari u izlogu: AMSTRAD CPC 464, ELEKTRON, SPEKTRUM PLUS
Periferijska oprema: DISK JEDINICE
Programi koje treba imati: NEKI BOLJI BEJZICI
Majstorije na računaru: Spektrum — HALO, DA LI JE TO MAŠINAC? BBC/Elektron — ŠTA EJKORN NIJE REKAO?
Računar „galaksija": ROM 2 (umetak na 32 strane)
Katalog najboljih igara svih vremena za spektrum: PEDESET VELIČANSTVENIH
Generisanje naših slova u printerima „epson": ĆIRILICA NA „EPSONU"
Škola avanturističkih igara: UKLETI DVORAC I DRUGE BAJKE
Simulacije letenja: LETAČI BEZ DIPLOME
Računari u domaćoj radinosti: PROGRAMATOR EPROM-a (2)

Računari 4

Računari i njihove zamke: IMATI ILI UMRETI
Naš test: ATARI 800 XL
Uporedni tekst: „AMSTRAD" PROTIV „KOMODORA"
Računari u akciji: DISKETNA VEZA
Računari u poslovnoj primeni: MAŠINA SA BEZBROJ LICA
Računari u školi: HILJADU ZATO ZA RAČUNARE
Umetnost programiranja: MODERNE PROGRAMERSKE TEHNIKE
Majstorije na računaru: Spektrum — NOVE NAREDBE I FUNKCIJSKI TASTERI
Računari na brzinskom ispitu: SPRINTERI U KUĆICI PUŽA
Nova klasa mikroprocesora: DŽINOVI NA GLAVI ČIODE
Katalog najboljih igara za „komodor": PEDESET NAJBOLJIH
Programiranje na bejziku: NOVE AZBUKE NA „KOMODORU"; PROMENLJIVE BEZ TAJNI
Škola sistemskog softvera: PUT U SREDIŠTE ROM-a
Škola simulacija letenja: DOBAR LET, ELEKTRONSKA PTICA
Računari u domaćoj radinosti: GENERATOR TONA

Računari 5

Računari u izlogu: DVOBOJ DŽINOVA
Periferijska oprema: NEKI NOVI — RADOST CRTANJA
Periverijska oprema: NEKI NOVI „EPSONI"
Računari i obrada teksta: PISANJE BEZ MUKE
Sedam načina komuniciranja sa kompjuterom: TAKO ĆE GOVORITI RAČUNARI
Računari u akciji: STROGO KONTROLISANI DISKOVI
Škola sistemskog programiranja: RASPODELA MEMORIJE
Umetnost programiranja: Spektrum — INTERAPTI U BEJZIKU
Majstorije na računaru: Nove naredbe na spektrumu — NA SVOJU SLIKU I PRILIKU
Programiranje u bejziku: SPRAJTOVI NA „KOMODORU"
U svetu slučajnih brojeva: ZEC IZ ŠEŠIRA
Računari u domaćoj radinosti: PRIČA O FINOJ GRAFICI

Računari 6

Računari u izlogu: ENTERPRAJZ — PREDUZIMLJIVOST SA ZADRŠKOM
PROMAŠAJ NA JAPANSKI NAČIN: RAČUNARI IZLAZEĆEG SUNCA
Periferijska oprema: MONITORI
Programeri na ispitu: HAKERSKI ZAKONIK
Programi koje treba imati: SLIKARI BEZ KIČICE
Računari i obrada teksta: PERO OD OSAM BITA
Majstorije na računaru: komodor 64 — NOVE NAREDBE NA NOVI NAČIN
Umetnost programiranja: spektrum — NOVA KRUNA ZA STAROG KRALJA
Put u središte ROM-a: VELIKA VIDEO PREDSTAVA
Majstorije na računaru: spektrum — EKRAN POD PRESOM
Programiranje u bejziku: HAMLET U RAČUNARU — Kako kompjuteri donose odluke
Umetnost programiranja: SVIRKA NA „KOMODORU"
Računari i igre: BESMRTNOST I KAKO JE STEĆI
Računari u domaćoj radinosti: spektrum — ROM OD SEDAM MILJA; galaksija — FINI NARDVER ZA FINU GRAFIKU

Računari 7

Naš test: „KANON" PROTIV „EPSONA"
Računari u izlogu: komodor 128 — 0 „KOMODORU" SVE NAJLEPŠE
Periferijska oprema: SVETLOSNA PERA
Umetnost programiranja: HAKER NA USIJANOM LIMENOM KROVU
Računari u akciji: BAZE PODATAKA
Programi koje treba imati: NE DAJ SE, INES!
Računari u poslovnoj primeni: MALI RAČUNARI U VELIKOJ PRIVREDI
SVE „SPEKTRUMOVE"
RUTINE (umetak na 32 strane)
Test sa zadrškom: PREMA SVECU I DRAJV
Majstorije na računaru: komunikacija sa periferijskim uređajima — PUTOVANJE U PROVINCIJE
Programiranje u bejziku: FUNKCIJE, POTPROGRAMI, PROCEDURE
Grafika na računaru: KAKO TO RADI „AMSTRAD"
Programiranje na mašincu: spektrum — KODOVI IZ RUKAVA
Put u središte ROM-a: LIČNE STVARI BEJZIKA
Računari u domaćoj radinosti: ROM OD SEDAM MILJA (2); BRISAČ EPROM-a

Računari 8

Naš test: GALAKSIJA PLUS
Računari u izlogu: ATARI 520 ST
Periferijska oprema: modemi — SVET NA DLANU
Istorija računara: RAĐANJE PROGRAMSKIH JEZIKA
Akcije: EKRANSKI EDITOR I DRUGE BAJKE
SVE „KOMODOROVE" RUTINE (umetak na 32 strane)
Računari i matematika: spektrum — GRAFIČKO PREDSTAVLJANJE FUNKCIJA
Programiranje u bejziku: animacija ekrana (amstrad) — NA VRH BRDA VRBA MRDA
Majstorije na računaru: spektrum — MAŠINSKA VEZA
Škola logičkih igara (1): VOLITE LI PASKAL?
Računari u poslovnoj primeni: obrada ličnih dohodaka — PLAVI KOVERTI IZ RAČUNARA
Računari i umetnost: STRIPOTEKA NA KOMPJUTERU

Računari 9

Računari u izlogu: ŽIVELA AMIGA
Hakerski manifest: PRAVI PROGRAMERI NE GOVORE PASKAL
Periferijska oprema: PALICE ZA IGRU; LASERSKA VEZA
Programski jezici: KOMA ZBOG KOMALA
Tehnike programiranja: BACITE PROZORE KROZ PROZOR
Operativni sistemi: PRVI GEM ZA TRIPOS
Put u središte ROM-a: U SVETU EDITORA
Akcije: PRIČA O EDITORU
Makazama po ROM-u: spektrum — GENS IZ ROM-a
Majstorije na računaru: KVADRATNI KOREN (I)
Majstorije na računaru: spektrum — MAŠINSKA VEZA (2) — pozivanje mašinskih programa iz bejzika
Majstorije na računaru: komodor 64 — AUTOMATSKI START
Računari i nauka: STIMULANTI I MODELI
Škola logičkih igara (2): POSLEDNJI UVEK DOBIJA
Računari i matematika: NE DIRAJ MOJE KRUGOVE
Numerički metodi: INTERPOLACIJA — NUMERIČKI KRIVULJAR
Računari u poslovnoj primeni: Obrada podataka u RO NA RAČUNARU
Računari u domaćoj radinosti: DOPING ZA Z80

Računari 10

Računari u izlogu: AMSTRAD PCW 8256
Pisanje na računaru: MUKE SA TEHNOLOGIJOM
Računari u izlogu: SPEKTRUM 128
Naš test: Voice master 2 — SVOGA GLASA GOSPODAR
Računari i obrazovanje: NA BALKANU NIŠTA NOVO
Periferijska oprema: SA DISKOM ILI NA NJEMU
Razgovor sa računarom: MIŠEVI I SVECI
Organizacija računara i potprogrami iz ROM-a: SVE „AMSTRADOVE" RUTINE
Majstorije na računaru: komodor 64 — PROGRAMI IZ ROM-a
Majstorije na računaru: komodor 64 — TURBO KOJI MOŽE SVE
Matematički softver: EKSPONENCIJALNA FUNKCIJA
IGRE KOJE STE NAJVIŠE VOLELI
Softverska trpeza: DAN PO JUTRU, A JUTRO TMURNO
Put u središte ROM-a: JEZIČKI PROCESORI

Računari 11

Periferijska oprema: HARD DISKOVI
Kompjuterske mozgalice: NEUPISIVO UPISIVANJE
Naš test: VIDEO DIGITAJZER
Računari u izlogu: Superkompjuteri '86 — DŽINOVI U BOCI
Operativni sistemi: „AMSTRADOV" KERNAL
Naš test: PODMLAĐENI BBC
Umetnost programiranja: NAJKRAĆI PUT
Majstorije na računaru: komodor 64 — SRPSKOHRVATSKI BEJZIK
Interfejsi: komodor 64 — „KOMODOR" U MREŽI
Programiranje u bejziku: MATRICE I NIZOVI
Majstorije na računaru: spektrum — NOVI ROM
Numerički metodi: LAGRANŽOVI POLINOMI
Radionica logičkih igara: KAMENČIĆI NA KOMPJUTERU
Matematički softver: HIPERBOLIČKE FUNKCIJE
Računari u poslovnoj primeni: ZALIHE NA DISKETI
Udruženi programeri: PROFI ADRESAR

Računari 12

Pogled izbliza: LICEM U LICE SA „ATARIJEM"
Softverska trpeza: NOVINAR U SVOJOJ KUĆI
Istorija računara: IBM I SEDAM PATULJAKA
Umetnost programiranja: HOLANDSKA ZASTAVA
Operativni sistemi: CP/M
Programiranje na mašincu: SAŽIMANJE TEKSTA
Kako se koristi DEVPAC MONS
Umesto umetka: „AMSTRADOV" KALKULATOR
Majstorije na računaru: spektrum — OPERACIJA FILL
Majstorije na računaru: amstrad — KRALJEVSTVO ZA REGISTAR
Akkcije: komodor 64 programi za kertridž — SAM SVOJ PROGRAMER
Matematički softver: LOGARITAMSKA FUNKCIJA
Udruženi programeri: JEZIČKI PROCESORI U 16 LEKCIJA
Programiranje u bejziku: SVIRKA NA „AMSTRADU"
Put u središte ROM-a: PROCEDURE U AKCIJI
Hajde da se igramo: POKICE KOJE ŽIVOT ZNAČE; ELITNA GROZNICA
Spektrumovci u nevolji: SMRTONOSNA DIJAGNOSTIKA

Računari 13

Računari u izlogu: NEKI STARI GOSPODARI
Računari na sajmu: MIKROCOMPUTER '86
Računari u izlogu: SUPERPERSONALCI U VAŠOJ KUĆI
Operativni sistemi: MALI DIV VELIKOG SRCA
Računari na testu: Megaatari — DŽIN IZ PLASTIČNE KUTIJE
Udruženi programeri: ČASOPISI ZA HARDVERAŠE
Umesto umetka: DEVPAC GENS
Obrada ideja: MISLI KAO ŠTO PIŠEŠ
Akcije: komodor 64 — bejzik iz eprom-a
Matematički softver: AREA FUNKCIJE
Umetnost programiranja: TRIVIJALAN, NAJTRIVIJALNIJI, SKORO REŠIV ZADATAK
Tehnike programiranja: rekurzije — KAKO SE IZBAVITI IZ LAVIRINTA
Biblioteka programa: EKRANSKI EDITOR (1)
Radionica logičkih igara: KAKO POTKRESATI STABLO

računari 1 i 2 rasprodati

NARUDŽBENICA

Galaksija, Bulevar vojvode Mišića 17, 11000 Beograd

Molim vas da mi pouzećem pošaljete sledeće stare brojeve RAČUNARA: 3, 4, 5, 6 (za 20 dinara po primerku)
7, 8, 9, 10, 11 (za 250 din. po primerku).
12 (po ceni od 300 din.) — zaokružiti odgovarajuće brojeve.

Ime i prezime _____

Ulica i broj _____

Broj pošte i mesto _____

(Datum) (Potpis)

NAPOMENA: Ukoliko ne želite da isecanjem narudžbenice oštetite svoj primerak „Računara", molimo da potrebne podatke ispišete na dopisnici ili u pismu i pošaljete na navedenu adresu.

EXPOND®

BODY BUILDING SET

IZGRAĐUJUĆI SVOJE TIJELO IZGRAĐUJETE SVOJ DUH

novo

• SKINITE SUVIŠNE KILOGRAME •

VJEŽBAJTE S NAMA • OBLIKUJTE SVOJE TIJELO • SKINITE SUVIŠNE KILOGRAME

UTEZI „EKSPOND"

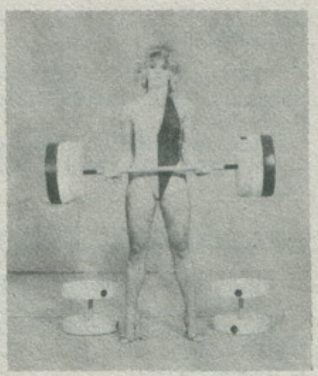
ŽIVEĆI u vremenu aerobika, džoginga i raznih dijeta, učinite i vi nešto za vaš izgled s pomoću Koplastove garniture plastičnih utega „EKSPOND", jedinstvenih na svetskom tržištu. ESTETSKI oblikovani, prosti za upotrebu i rukovanje, laki za održavanje, sklopivi i praktični za postizanje željenih težina, koja zavisi od primene punila.

UČINITE prvi korak ka zdravijem i zadovoljnijem životu upotrebom „EKSPOND" utega.

OSNOVNE KARAKTERISTIKE:

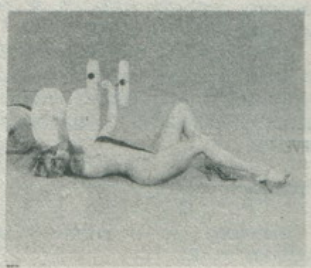
„EKSPOND" set se sastoji od tri metalne šipke; dvoručne i dve jednoručne, te šest plastičnih utega. Dvoručna šipka se sastoji od rukohvata i dva rukavca — nosača utega, koji se mogu teleskopski razmicati zavisno od potrebne širine, odnosno količine utega. JEDNORUČNE šipke su na istom principu, a namenjene su za sklapanje dvaju jednoručnih utega. ZAPREMINA jednog utega je 4,4 l, a težina praznog utega je 0,6 kg. UTEG napunjen vodom teži 5 kg, a peskom se težina povećava za otprilike 2,5 puta.

Vježba za razvoj mišića stražnjeg dijela nadlaktice. Raditi 3×10 ponavljanja.



Vježba za razvoj mišića prednjeg dijela nadlaktice. Raditi 3×10 ponavljanja, naizmjenično lijeva i desna ruka.

Vježba za razvoj mišića ramena. Utezi se dižu od bedara do prikazanog položaja. Raditi 3×10 ponavljanja.



Vježba za razvoj mišića grudi. Iz prikazanog položaja jednoručni utezi se spuštaju do tla i vraćaju u početni položaj. Raditi 3×10 ponavljanja.

ĐURO SALAJ KOPLAST

„EKSPOND" utezi se mogu naručiti u garnituri i pojedinačno u elementima, shodno nameni i potrebama korisnika. ŽELIMO vam dobre sportske rezultate.

EXPOND — CENE:

KOMPLET SET ------ 8 382 din
UTEG 1 kom ---.---- 847 din
DVORUČNA CEV----- 1 859 din
JEDNORUČNA CEV --- 1 012 din

NARUDŽBENICA – RACUNARI 15

Ovim neopozivo naručujem EXPOND komplet set ili delove-navesti koje _____ po ukupnoj ceni: _____ din.

Ovaj iznos uvećan za troškove poštarine platiću poštaru prilikom preuzimanja pošiljke.

Ime i prezime _____

Godina rođenja _____

Broj lične karte _____

Adresa _____

Poštanski broj i mesto _____

Čitko popunjene narudžbenice slati na adresu: Agencija DUGA — Bulevar Vojvode Mišića 17, 11000 Beograd sa naznakom: „EXPOND"

ekranski editor

Naredbe za podešavanja

FREEZE ili FR

Zamrzava dva gornja reda ekrana. To praktično znači da ih listanje programa i slične radnje neće obrisati.

Dva gornja zamrznuta reda su izuzetno praktična da u njih upišete, recimo, neku matematičku formulu koju često koristite u programu — kasnije, kad god vam zatreba, možete da je iskopirate. Ta dva reda možete, takođe, da iskoristite kao neku vrstu elektronskog bloka za čuvanje poruka poput „Pazi na liniju 200" ili „DF SZ = 23659".

UNFREEZE ili UNFR

Odmrzava dva gornja reda ekrana.

COLOR n,m ili CL n,m

*Parametri: **n** — boja pozadine (paper i border)*
__m__ — boja slova (ink)

Postavlja boju pozadine i slova.

REPEAT n,m ili REP n,m

*Parametri: **n** — brzina za sve tastere osim CAPS/5 i CAPS/8*
__m__ — brzina samo za CAPS/5 i CAPS/8

Podešava brzinu automatskog ponavljanja ako je taster stalno pritisnut.

Kada se učita editor, **n** ima vrednost 4, a **m**=2. Ako vam je to presporo, probajte REP 2,1. S druge strane, ako volite da čekate, otkucajte REP 6,4 ili REP 8,5.

Zašto REPEAT naredba ima dva parametra? Jednostavno zato što ekran ima 51 poziciju horizontalno, a samo 20 vertikalno. U takvim uslovima bilo bi vrlo neprirodno da se kursor pomera istom brzinom i vertikalno i horizontalno (otkucajte REP 1,1 pa se i sami uverite).

FTIME [n] ili FT [n]

*Parametri: **n** — vreme u sekundama; ako se ne navede vrednost = 60.*

Definiše vreme posle kojeg kursori počinju da fleširaju, ako se ne pritisne nijedan taster.

Kada se učita editor, FLASH TIME je postavljen na 60 — ako jedan minut ne pritisnete nijedan taster, kursori počinju da fleširaju. To je zato da ne biste morali pogledom da tražite kursore po ekranu, jer se pretpostavlja da ceo taj minut niste gledali u ekran (da ste gledali u ekran, sigurno biste i pritisli neki taster).

BEEP ili BP

Svaki pritisak na neki taster biva propraćen sa jednim „biiip".

UNBEEP ili UNBP

Poništava BEEP stanje (videti BEEP).

KB+

Predefiniše tastaturu da odgovara „spektrumu plus". Da bi se označilo to stanje, u status liniji, sasvim desno, pojaviće se jedno malo +. (Više detalja, na kraju serije, u tekstu „Organizacija tastature")

KB-

Vraća tastaturu da odgovara običnom spektrumu (videti KB+).

ZOP

ZERO OPTION. Predefiniše karakter set tako da cifra nula i slovo „O" izgledaju malo drugačije. Autoru se čini da je tako poboljšana čitljivost, naročito linijskih brojeva.

ZOFF

Poništava ZERO OPTION.

Ako ste iz radoznalosti otkucali ZOP, pa vam se rezultat nije svideo, onda je ovo prava naredba za vas.

Naredbe za pamćenje slike

STORE ili STO

STORE SCREEN. Naredba STORE nalaže editoru da zapamti sliku koja se trenutno nalazi na ekranu. Od tog trenutka, naredba HELP više neće pokazivati „HELP SCREEN" (videti HELP), već tu vašu sliku, a moguće je koristiti i RECALL naredbu.

Za šta ovo može da koristi? U mnogim situacijama vrlo je zgodno stalno imati pred očima listing nekog važnog programskog segmenta ili potprograma. Tada je kombinacija STORE-/HELP zlata vredna.

RECALL ili RCL

RECALL SCREEN. Vraća na ekran sliku koja je bila sačuvana pomoću STORE naredbe. Naravno, time slika neće biti izbrisana iz memorije.

Napomena: RECALL daje grešku „INCORRECT STATEMENT" ako nikakva slika nije bila upisana u memoriju pomoću STORE naredbe.

EXCHANGE ili EXC

EXCHANGE SCREENS. Slika sa ekrana ide u memoriju, a slika iz memorije se vraća na ekran.

Napomena: EXCHANGE DAJE GREŠKU „INCORRECT STATEMENT" ako nikakva slika nije bila upisana u memoriju pomoću STORE naredbe.

Naredbe za rad sa mikrodrajvom

MDF n

*Parametar: **n** — broj mikrodrajva; ako se ne navede vrednost = 1*

Za razliku od svih drugih naredbi, MDF deluje u bejziku, a ne u editoru. Dejstvuje tako što automatski ispisuje zvezdice, navodnike, tačka zareze i sve ostalo što ide iza SAVE, LOAD, VERIFY, FORMAT i EARSE naredbe kada se radi sa mikrodrajvom.

Otkucati MDF 0 je isto što otkucati MDFOFF — isključuje se MDF stanje.

MDFOFF

MICRODRIVE FACILITY OFF. Poništava MDF stanje. (Videti naredbu MDF)

```
51968 06 37 ED 52 D2 06 CD DD E5 E1 22 17 CD 11 0F 27  11
51984 37 ED 52 D2 FD CC 2A CC FC FD 21 3A 5C CD 6E 19  0B
52000 2A 53 5C AF ED 52 28 0D EB 56 23 5E 2A D0 FC 37  EB
52016 ED 52 DA FD CC 2A CE FC 23 CD 44 DA 30 09 2A 17  5E
52032 CD AF ED 42 D2 FD CC CD 97 CD 2A 17 CD CD E8 E6  20
52048 21 C8 FC 06 03 3E 30 BE 20 03 23 10 FA 04 21 00  8F
52064 00 E5 CD A1 CD E1 30 0E 0D 28 F6 90 30 F3 ED 44  4E
52080 16 00 5F 19 18 EB 24 E5 FD 21 3A 5C CD B0 16 CD  AE
52096 1A 1F AF 67 6F ED 42 D1 37 ED 52 DA 06 CD 3A 14  2F
52112 CD 1F 30 0F 3E 02 CD B7 D8 FE 20 CA 11 CD FE 4E  D9
52128 CA FA CC CD 97 CD CD A1 CD 30 55 0D 20 F8 CD E8  5B
52144 E6 21 14 CD CB 4E CB CE 20 1F CD 98 BD ED 4B B1  E4
52160 FC C5 CD D4 F0 EB 21 19 CD 01 18 00 79 ED B0 C1  34
52176 CD 02 EF 0E 14 ED 43 B1 FC 2A B1 FC 7D C6 05 FE  DA
52192 31 6F 22 B1 FC D4 98 BD ED 4B B1 FC C5 CD D4 F0  D3
52208 EB 21 C8 FC 01 04 00 79 ED B0 C1 CD 02 EF 18 A6  28
52224 3A 14 CD CB 4F 28 0F 3E 03 CD B7 D8 FE 20 CA 11  02
52240 CD FE 4E CA FA CC ED 4B 15 CD C5 2A D0 FC E5 2A  8D
52256 CC FC CD 44 DA ED 5B D2 FC DD 21 00 40 18 05 C5  E9
52272 E5 CD 6D DA E5 FD E1 DD 71 00 DD 70 01 E1 DD 75  8B
52288 02 DD 74 03 FD 74 00 FD 75 01 19 01 04 00 DD 09  3E
52304 C1 0B 78 B1 20 D9 2A 3D 5C E5 21 02 CD E5 ED 73  CB
52320 3D 5C CD 97 CD CD A1 CD D2 EC CC 0D 28 F7 C5 D5  55
52336 F5 11 00 40 1A 13 4F 1A 13 47 ED 53 8C CC ED 53  0E
52352 D6 CC 13 13 37 E5 ED 42 E1 30 E9 2A 00 00 CD E8  EC
52368 E6 21 C8 FC 06 03 3E 30 BE 20 03 23 10 FA 04 F1  45
52384 D1 D5 C5 E5 FD 21 3A 5C 90 28 1E 06 00 2A 00 00  0A
52400 38 0A 4F ED 42 E5 EB CD E8 19 18 09 ED 44 4F 09  08
52416 E5 EB CD 55 16 E1 22 00 00 E1 C1 D1 48 06 00 ED  B9
52432 B0 C1 0D 20 90 2A 00 00 EB 7E 23 FE 0E 20 FA AF  B9
52448 77 23 77 23 73 23 72 23 77 C3 65 CC CD DB F1 CD  30
52464 4B C4 CD 74 E5 E1 E1 22 3D 5C 3E 01 C9 21 64 CD  0C
52480 AF C9 E1 22 3D 5C 21 31 CD 22 54 FD 3E 01 32 E1  F8
52496 FC 3E 02 C9 00 00 00 00 00 43 41 4E 27 54 20 52  C4
52512 45 4E 55 4D 42 45 52 20 41 54 20 4C 49 4E 45 20  2B
52528 3A 20 20 20 4F 20 55 20 54 20 20 20 4F 20 46 20  07
52544 20 20 4D 20 45 20 4D 20 4F 20 52 20 59 20 20 20  19
52560 2D 20 20 43 41 4E 27 54 20 52 45 4E 55 4D 42 45  E8
52576 52 20 20 20 20 20 42 20 41 20 44 20 20 20 50 C9  C9
52592 20 41 20 52 20 41 20 4D 20 45 20 54 20 45 20 52  51
52608 20 20 20 2D 20 20 43 41 4E 27 54 20 52 45 4E 55  74
52624 4D 42 45 52 20 20 20 2A 53 5C CD E9 CE 22 F9 CE  CC
52640 C9 DD E5 DD 21 01 CF DD 70 00 DD 36 03 00 2A F9  DF
52656 CE 22 09 CF CD 05 CF FE 0D CA B8 CE FE 0E CA CE  68
52672 CE FE 22 CA D8 CE FE EC 28 20 FE ED 28 1C FE F7  B4
52688 28 18 FE F0 28 14 FE E1 28 10 FE E5 28 0C FE EA  80
52704 CA B1 CE FE CA 20 CD DD 34 03 2A 09 CF 22 F9 CE  FD
52720 CD 05 CF DD 36 01 00 FE B0 20 0E DD 34 01 CD 05  75
52736 CF FE 22 C2 AA CE CD 05 CF CD 1B 2D DA 93 CE 2A  44
52752 09 CF 22 FD CE DD 36 02 01 D6 30 26 00 6F CD 05  48
52768 CF CD 1B 2D 38 1A 11 E7 03 37 ED 52 30 7C ED 5A  9A
52784 DD 34 02 D6 30 4F 11 0A 00 CD A9 30 59 19 18 DE  91
52800 22 FF CE DD 35 01 20 08 FE 22 20 5E 0E 03 18 10  01
52816 FE 0E 20 56 2A 09 CF 11 05 00 19 22 09 CF 0E 02  BD
52832 CD 05 CF FE 3A 28 04 FE 0D 20 3F 2A FF CE ED 5B  AE
52848 CC FC AF ED 52 DA AA CD 19 ED 5B CE FC 37 ED 52  A8
52864 D2 AA CD ED 5A DD 7E 02 ED 5B FD CE 37 DD 46 00  5A
52880 DD E1 C9 DD 35 03 20 12 CD 8D 2C 30 0D CD 05 CF  32
52896 CD 88 2C 38 F8 FE 24 CA AA CD 0E 01 2A FB CE 18  2E
52912 DB CD 05 CF FE 0D 20 F9 2A 09 CF 23 CD E9 CE 22  6B
52928 09 CF ED 5B 4B 5C AF ED 52 DA B4 CD 18 BF 06 05  F2
52944 CD 23 CF 10 FB C3 B4 CD CD 05 CF FE 22 20 F9 CD  B5
52960 05 CF FE 22 28 F2 C3 B7 CD 56 23 5E 23 22 AE CC  EB
52976 22 C7 CC 23 ED 53 FB CE C9 00 00 00 00 00 00 00  AA
52992 00 00 00 00 00 CD 23 CF 3A 00 00 FE 20 28 F6 FE  33
53008 10 D8 FE 16 28 05 38 06 FE 17 C0 CD 23 CF CD 23  EB
53024 CF 18 E2 DD 34 08 C0 DD 34 09 C9 32 C4 CF AF 32  2B
53040 6C FD CD 01 D1 2A 3D 5C E5 21 52 D0 E5 ED 73 3D  75
53056 5C CD D2 ED 0E 00 CD 4E D1 3A C4 CF FE 02 C7  B9
53072 17 1B 2A 59 5C 22 5D 5C CD FB 19 3A C4 FC FE 02  C7
53088 20 04 ED 43 CC FC ED 43 49 5C 2A 5D 5C EB 2A 61  4A
53104 5C 37 ED 52 E5 60 69 CD 6E 19 20 06 CD B8 19 CD  65
53120 E8 19 C1 79 3D B0 28 35 C5 03 03 03 03 2B ED 5B  C9
53136 53 5C D5 CD 55 16 E1 22 53 5C C1 C5 13 2A 61 5C  EE
53152 2B 2B ED B8 2A 49 5C EB C1 70 2B 71 2B 73 2B 72  BD
53168 E1 E1 22 3D 5C CD AE E6 16 00 C3 2E D1 3E 01 32  27
53184 6C FD 18 EC 00 2A 59 5C 2B 22 5D 5C E7 CD 1B 2D  4E
53200 30 FA 18 01 E7 FE 0D CA 52 CF FE EA CA 52 CF FE  F1
53216 22 28 5E FE 2E 28 18 FE C4 28 14 0D 1B 2D 38 E4  43
53232 2A 5D 5C 2B 7E CB AF FE 41 38 04 FE 5B 38 CD 2A  09
53248 3D 5C E5 21 34 D0 E5 ED 73 3D 5C DF CD 9B 2C E1  D5
53264 E1 22 3D 5C 2A 5D 5C 01 06 00 CD 55 16 23 36 0E  25
53280 E5 CD F1 2B E1 23 77 23 73 23 72 23 71 23 70 22  BD
53296 5D 5C 18 A0 E1 22 3D 5C 3A 3A 5C FE 05 28 8D 18  AD
53312 93 E7 FE 0D CA 52 CF FE 22 20 F6 E7 FE 22 28 F1  C6
53328 18 83 E1 22 3D 5C 3A 3A 5C FE 03 20 07 3E 01 21  8F
53344 9B D0 18 2C E5 21 52 D0 E5 ED 73 3D 5C FD CB 00  7D
53360 7E C2 52 CF 2A 59 5C CD A7 11 0E 01 2A 5F 5C CD  86
53376 4E D1 ED 43 5C FD E1 E1 22 3D 5C 3E 02 21 CE D0  24
53392 16 01 32 E1 FC 22 54 FD C3 2E D1 20 20 4F 20 55  5F
53408 20 54 20 20 20 4F 20 46 20 20 20 4D 20 45 20 4D  08
53424 20 4F 20 52 20 59 20 21 20 20 20 4C 49 4E 45 20  43
53440 4E 4F 54 20 41 43 43 45 50 54 45 44 20 20 20 20  CA
53456 53 20 59 20 4E 20 54 20 41 20 58 20 20 20 45 20  4C
53472 52 20 52 20 4F 20 52 20 21 20 20 20 20 20 4C 49  1B
53488 4E 45 20 4E 4F 54 20 41 43 43 45 50 54 45 44 20  1D
53504 20 2A 52 FD 22 B1 FC CD 08 E0 ED 4B E9 FC 3E A9  21
53520 CD 22 D1 60 69 ED 4B EB FC 3E A7 A7 ED 42 20 02  85
53536 3E AA CD D4 F0 56 77 3E 01 CD E4 EE 72 C9 14 15  88
53552 20 07 3A 5E FD FE 07 38 F5 D9 CD E3 ED D9 ED 4B  75

53568 E9 FC 3E 01 CD E4 EE ED 4B EB FC C3 E4 EE ED 73  D7
53584 E8 D3 C5 22 E6 D3 FD 21 3A 5C CD B0 16 2A 59 5C  81
53600 01 00 01 CD 55 16 ED 4B E9 FC CD D4 F0 2B 79 3D  C9
53616 ED 5B EB FC 4B 1E 00 08 D9 C1 DD 2A 59 5C CD 04  C7
53632 D3 D2 ED D2 FE 22 CA CF D2 57 FE 3C 28 52 FE 3E  36
53648 28 6D CB AF FE 41 DA 1B D2 FE 5B 30 75 CB C9 06  AD
53664 00 21 84 D3 04 72 23 3E 0A B8 CA B4 D2 CD 04 D3  05
53680 D2 3F D2 57 CB AF FE 41 38 04 FE 5B 38 E6 7A FE  1E
53696 3A CA 48 D2 FE 2E CA 52 D2 FE 23 CA 63 D2 FE 24  7A
53712 CA 6A D2 FE 20 CA 75 D2 CD 29 D3 D4 A3 D3 18 14  74
53728 CD 04 D3 D2 EA D2 FE 3D 28 0D FE 3E 28 0D 57 3E  A8
53744 3C CD AE D3 7A 18 8D 3E C7 18 1A 3E C9 18 16 CD  E2
53760 04 D3 D2 EA D2 FE 3D 28 05 57 3E 3E 18 E3 3E C8  A1
53776 18 03 7A CB C9 CD AE D3 C3 7E D1 7A CB 49 20 F5  2C
53792 FE 30 38 0A FE 3A 30 EB CB 51 20 E7 18 0B FE 20  27
53808 20 E1 CB 51 CB D1 CA 7E D1 CD B1 D3 C3 7E D1 CD  02
53824 29 D3 D4 A3 D3 C3 ED D2 CD 29 D3 D4 A3 D3 3E 3A  53
53840 18 C3 CB E0 CD 29 D3 DA 7E D1 CB A0 CD A3 D3 3E  64
53856 2E 18 B2 CD 29 D3 3E 23 38 AB 04 77 CD 29 D3 D4  1D
53872 A3 D3 C3 7E D1 CD 29 D3 DA 7E D1 08 D9 F5 C5 E5  FA
53888 D9 08 04 36 20 23 16 03 CD 04 D3 30 1C 04 77 23  05
53904 CD 29 D3 38 19 15 20 F0 78 D6 04 47 08 D9 E1 C1  5B
53920 F1 D9 08 CD A3 D3 C3 7E D1 CD A3 D3 18 3F E1 E1  83
53936 E1 C3 7E D1 CD A3 D3 CD 04 D3 30 31 57 CB AF FE  0A
53952 41 DA F4 D1 FE 5B D2 F4 D1 7A CD B1 D3 18 E8 CD  68
53968 B1 D3 CD 04 D3 30 16 FE 22 20 F4 CD AE D3 CD 04  C1
53984 D3 30 0A FE 22 C2 89 D1 18 E5 CD AE D3 CB 41 C2  62
54000 CA D3 D9 DD E5 E1 06 00 4B 0C 0D 20 01 04 CD E8  5D
54016 19 C3 CD D3 08 D9 B9 28 0B 3C FE 33 28 11 23 08  1A
54032 7E 37 D9 C9 08 78 BA 28 03 08 18 ED D9 AF C9 3E  58
54048 05 04 23 23 23 23 18 E5 C5 D5 E5 04 04 21 84 E1  E1
54064 D3 7E CB AF D6 41 87 21 EA D3 16 00 5F 19 5E 23  56
54080 56 EB 16 00 4E 23 7E 5F B8 CC 57 D3 CB A3 19 0D  E7
54096 20 F4 E1 D1 C1 AF C9 C5 D5 E5 05 05 CB A0 11 83  87
54112 D3 13 23 1A BE 28 09 CB AF BE 28 04 E1 D1 C1 C9  B2
54128 10 EF 23 7E E1 E1 E1 E1 D1 C1 CD B1 D3 FE EA D0  3E
54144 28 16 37 C9 00 00 00 00 00 00 00 00 00 00 00 00  3E
54160 00 00 00 00 00 00 00 00 CD 04 D3 D2 ED D2 CD B1  B3
54176 D3 18 F5 21 84 D3 7E 23 CD AE D3 10 F9 C9 FE 20  37
54192 C8 DD 77 00 DD 23 CB 41 28 18 D5 E5 DD E5 E1 ED  B2
54208 5B E6 D3 A7 ED 52 E1 D1 20 08 08 D9 4F ED 7B E8  54
54224 D3 C9 D9 1D D9 C0 C5 D5 E5 DD E5 E1 01 00 01 CD  1C
54240 55 16 E1 D1 C1 C9 00 00 00 00 16 D4 41 D4 6E D4  E8
54256 DE D4 14 D5 29 D5 4F D5 00 00 82 D5 00 00 00 00  14
54272 BF D5 04 D6 1E D6 36 D6 64 D6 00 00 BA D6 02 D7  11
54288 47 D7 63 D7 6C D7 09 05 41 42 53 BD 05 41 43 53  18
54304 B6 05 41 4E 44 C6 05 41 53 4E B5 04 41 54 AC 05  3A
54320 41 54 4E B7 06 41 54 54 52 AB 14 41 54 AB 13 47  2E
54336 AB 08 06 42 45 45 50 D7 14 42 45 D7 13 42 D7 05  4F
54352 42 49 4E C4 08 42 4F 52 44 45 52 E7 15 42 4F 52  42
54368 E7 08 42 52 49 47 48 54 DC 15 42 52 49 DC 13 05  71
54384 43 41 54 CF 06 43 48 52 24 C2 14 43 24 C2 08 43  F8
54400 49 52 43 4C 45 D8 15 43 49 52 D8 07 43 4C 45 41  2E
54416 52 FD 15 43 4C 45 FD 13 43 FD 09 43 4C 4F 53 45  07
54432 20 23 D4 15 43 4C 4F D4 08 43 4C 4F 53 45 23 D4  53
54448 15 43 4C 4F D4 05 43 4C 53 FB 06 43 4F 44 45 4F  79
54464 14 43 4F 4F 0A 43 4F 4E 54 49 4E 55 45 E8 15 43  04
54480 4F 4E E8 06 43 4F 50 59 FF 05 43 4F 53 B3 0A 06  72
54496 44 41 54 41 E4 14 44 41 E4 13 44 E4 08 44 45 46  8D
54512 20 46 4E CE 15 44 45 46 CE 07 44 45 46 46 4E CE  6C
54528 15 44 45 46 CE 05 44 49 4D E9 06 44 52 41 57 FC  AA
54544 14 44 52 FC 04 07 45 52 41 53 45 D2 15 45 52 41  E0
54560 D2 13 45 D2 05 45 58 50 B9 07 07 46 4C 41 53 48  23
54576 DB 15 46 4C 41 DB 04 46 4E A8 05 46 4E 52 EB 13  C8
54592 46 EB 08 46 4F 52 4D 41 54 D0 15 46 4F 52 D0 0A  A8
54608 08 47 4F 20 53 55 42 ED 14 47 53 ED 07 47 4F 20  ED
54624 54 4F EC 14 47 54 EC 13 47 EC 07 47 4F 53 55 42  F7
54640 ED 14 47 53 ED 06 47 4F 54 4F EC 14 47 54 EC 13  61
54656 47 EC 0B 04 49 46 FA 04 49 4E BF 05 49 4E 4B D9  E5
54672 08 49 4E 4B 45 59 24 A6 15 49 4E 24 A6 07 49 4E  66
54688 50 55 54 EE 15 49 4E 50 EE 13 49 EE 05 49 4E 54  0B
54704 BA 09 49 4E 56 45 52 53 45 DD 15 49 4E 56 DD 0D  A8
54720 05 4C 45 4E B1 05 4C 45 54 F1 13 4C F1 06 4C 49  5B
54736 4E 45 CA 14 4C 49 CA 06 4C 49 53 54 F0 07 4C 4C  A1
54752 49 53 54 E1 15 4C 4C 49 E1 04 4C 4E B8 06 4C 4F  9F
54768 41 44 EF 14 4C 4F EF 08 4C 50 52 49 4E 54 E0 15  E8
54784 4C 50 52 E0 05 07 4D 45 52 47 45 D5 15 4D 45 52  18
54800 D5 13 4D D5 06 4D 4F 56 45 D1 14 4D 4F D1 05 05  A3
54816 4E 45 57 E6 06 4E 45 58 54 F3 14 4E 45 F3 13 4E  03
54832 F3 05 4E 4F 54 C3 09 08 4F 50 45 4E 20 23 D3 14  19
54848 4F 50 D3 07 4F 50 45 4E 23 D3 14 4F 50 D3 04 4F  7A
54864 52 C5 05 4F 55 54 DF 06 4F 56 45 52 DE 14 4F 56  CC
54880 DE 13 4F DE 10 07 50 41 50 45 52 DA 15 50 41 50  7D
54896 DA 07 50 41 55 53 45 F2 15 50 41 55 F2 06 50 45  D9
54912 45 4B BE 14 50 45 BE 04 50 49 A7 06 50 4C 4F 54  3E
54928 F6 14 50 4C F6 07 50 4F 49 4E 54 A9 15 50 4F 49  D3
54944 A9 06 50 4F 4B 45 F4 14 50 4F F4 07 50 52 49 4E  B9
54960 54 F5 15 50 52 49 F5 13 50 F5 0C 0B 52 41 4E 44  D2
54976 4F 4D 49 5A 45 F9 15 52 41 4E F9 06 52 45 41 44  8E
54992 E3 14 52 45 E3 13 52 E3 05 52 45 4D EA 09 52 45  2C
55008 53 54 4F 52 45 E5 15 52 45 53 E5 08 52 45 54 55  9E
55024 52 4E FE 15 52 45 54 FE 05 52 4E 44 A5 05 52 55  D6
55040 4E F7 0D 06 53 41 56 45 F8 14 53 41 F8 09 53 43  BE
55056 52 45 45 4E 24 AA 15 53 43 24 AA 05 53 47 4E BC  1A
55072 05 53 49 4E B2 05 53 51 52 BB 06 53 54 45 50 CD  66
55088 14 53 54 CD 06 53 54 4F 50 E2 13 53 E2 06 53 54  AB
55104 52 24 C1 14 53 24 C1 06 05 54 41 42 AD 05 54 41  AC
55120 4E B4 06 54 48 45 4E CB 14 54 48 CB 13 54 CB 04  B3
55136 54 4F CC 02 05 55 53 52 C0 13 55 C0 06 05 56 41  FA
55152 4C B0 06 56 41 4C 24 AE 14 56 24 AE 08 56 45 52  E8
```

Vladimir Kostić

Numerički metodi za mikroračunare

Autori: Nenad Mladenović, Veljko Spasić, Milorad Jovanović; recenzent dr Lav Ivanović, docent PMF; izdavač: NIRO „Tehnička knjiga" i Zavod za izdavanje udžbenika, Beograd, 1986. Strana: 185, cena: 1.850 dinara, tiraž: 4.000 primeraka

Autori mr Nenad Mladenović, matematičar koji radi kao asistent na Fakultetu organizacionih nauka u Beogradu, mr Veljko Spasić, matematičar koji radi kao istraživač u Centru za multidisciplinarne studije u Beogradu i Milorad Jovanović, matematičar koji radi kao profesor programiranja u Beogradu, izložili su u knjizi programe kojima se na mikroračunarima „komodor 64" i „spektrum" realizuju metode numeričke analize. N. Mladenović je jedan od koautora knjige „Kućni kompjuteri (algoritmi i programi)", a V. Spasić — „BASIC za mikroračunare", obe u izdanju Tehničke knjige, dok je M. Jovanoviću ovo autorski prvenac. Razume se, sam izbor računara na kojima su programi realizovani kazuje da ambicija autora i izdavača nije bila da objave naučno delo već da približe metode numeričke matematike širem krugu korisnika koji se susreću sa ovim problemima bilo u okviru školskih predmeta bilo što ih koriste kao sredstvo za obavljanje složenijih projekta i istraživanja. Stoga i ne dajemo naučno-stručni prikaz knjige, već izlažemo samo informacije koje bi mogle biti interesantne za potencijalne čitaoce ovako koncipirane knjige.

Na početku je u Bekusovoj notaciji dat pregled naredbi, komandi i funkcija bejzik jezika za „komodor" i „spektrum", a zatim su prikazane metode interpolacije, rešavanja sistema linearnih algebarskih jednačina, računanja karakterističnih vrednosti i rešavanja nelinearnih jednačina. Svaka od metoda izložena je u najkraćim crtama, objašnjen je alegoritam korišćen za njenu implementaciju na računaru i priloženi su test primer i listing programa za „komodor" ili „spektrum". Ova shema je dosledno korišćena za sve metode, ali se ipak može zapaziti da je knjigu radilo više autora koji nisu do detalja uskladili stil izlaganja. Kako su korišćene samo osnovne mogućnosti bejzika, programi se jednostavno mogu modifikovati za korišćenje na drugim računarima, a profesionalni način kreiranja programa dozvoljava da se sa lakoćom mogu vršiti i druge vrste izmena. Sve su ovo pogodnosti koje omogućavaju da se izloženi programi mogu koristiti ne samo u nastavi numeričke analize već i u nastavi programiranja.

Ipak, treba imati u vidu da knjiga, mada nosi naziv „Numeričke metode", pokriva samo njihov deo i da u programima nije dobro izabran kriterijum završetka izračunavanja. Grafička rešenja i kvalitet štampe su na zavidnom nivou.

Commodore za sva vremena

Autori: Dragan Tanaskoski, Stevan Milinković, Vladimir Janković; recenzent: Ladislav Rupnik; izdavač: Samostalno izdanje autora, Beograd, 1986. Strana: 336, cena: 3.600 dinara, tiraž: 6.000 primeraka

Autori Dragan Tanaskoski, mašinski inženjer koji radi u Institutu za fiziku i Stevan Milinković i Vladimir Janković, inženjeri elektronike koji se računarima profesionalno bave u Institutu „Boris Kidrič" u Vinči, pripremili su za vlasnike „komodora" knjigu koja im može pomoći da ovladaju većinom njegovih mogućnosti. D. Tanaskoskom i V. Jankoviću ovo nije prvo samostalno izdanje. Oni su koatori i izdavači uspešno predavane knjige „Spektrum priručnik", a Dragan Tanaskoski je našim čitaocima poznat i kao koautor programa „Eastenglish" nagrađenog na „Galaksijinom" konkursu za najbolje YU programere.

Počevši od osnovnih pojmova kao što su bit i bajt i načina uključivanja računara i korišćenje ekranskog editora, autori izlažu korisnicima „komodora" sve one informacije koje bi se inače morale tražiti u mnoštvu, za naše prilike, veoma skupih knjiga na stranim jezicima. Većina bitnih informacija iz nezaobilaznog Programmer's Reference Guidea (po ceni od 20$) može se, korektno interpretirana, naći u ovoj knjizi. Zoran Životić i Nevenka Spalević, koji su knjigu čitali, prepoznali su mnoge izvore koje su autori koristili, ovu knjigu treba, pre svega prihvati kao priručnik koji na jednom mestu iznosi informacije bitne za korišćenje popularnog „komodora", a ne kao iznošenje novih saznanja — utoliko pre što su svi izvori, uključujući i beleške sa predavanja profesora Dujmovića na Elektrotehničkom fakultetu, navedeni u literaturi. Istini za volju, mora se reći da stvari koje su ostale nedorečene u Reference Guideu i drugim izvorima nisu ni ovde pojašnjene.

Posebno poglavlje posvećeno je mogućnostima i korišćenju sajmons bejzika, a programiranju na mašinskom jeziku, kao i svim ostalim celinama, prethodi mali teoretski uvod koji objašnjava pojmove koji će se koristiti. Za ambicioznije čitaoce verovatno će posebno biti zanimljiva poglavlja posvećena organizaciji memorije i upotrebe ROM rutina u hardveru. Prezentirane su i konstrukcije centroniks interfejsa, RS 232 interfejsa, modema, eprom programatora i ROM modula.

Mada se načinu izlaganja materije mogu staviti i neke zamerke, one nisu suštinske i čitalac se, bez obzira na njih, u otkrivanju tajni „komodora" može osloniti na ovu knjigu. Očigledno je da su se velikog posla pisanja sveobuhvatnog priručnika, po meri jednog inženjera koji se prethodno nije bavio računarima, prihvatili znalci koji dobro poznaju materiju koju izlažu.

Sistemski softver „Amstrad"

ni na nebu, ni na zemlji

Operativni sistem „amstrada" ne prestaje da iznenađuje neumorne istraživače. Otkrivajući tajne „amstradovog" kalkulatora, ukazao se još jedan čudan stanovnik donjeg ROM-a. Radoznalim čitaocima sigurno nije promakao jedan (nedokumentovan) ulaz (BD3A) koji se nalazi između glavnog i kalkulatorovog džamp-bloka. Oni koji su otišli dalje (u ROM), verovatno su bili nepriјatno iznenađeni izuzetno razuđenim algoritmom. I, zaista, radi se o jednom od najkompleksnijih potprograma unutar operativnog sistema. Reč je o editoru.

Amstradov editor je, u osnovi, linijski, mada je ostavljena mogućnost „šetanja" po čitavom ekranu ili, preciznije, po trenutno važećem prozoru. To znači da se tekst može editovati na bilo kojoj poziciji („tekst" označava string dužine 255 znakova). Ali, ono po čemu se „amstradov" editor razlikuje od običnih linijskih editora je COPY opcija. Ona omogućava, na vrlo specifičan način (koji je dobro poznat vlasnicima BBC-ja), izlazak iz editovane linije. Kod običnih linijskih editora postoji jedan kursor, dok u našem slučaju postoje dva. Jedan služi za „pisanje" (označava poziciju na koju se smeštaju slova „očitana" sa taštature), a drugi za „čitanje". Drugi, ili COPY kursor, može se slobodno pomerati po čitavom prozoru. Pritiskom na specijalni taster COPY „čita" se slovo koje se nalazi ispod COPY kursora i upisuje na mesto kursora za pisanje. Dakle, postoje dva izvora teksta — tastatura i prozor u kom se vrši editovanje (na žalost, ovaj princip nije poštovan do kraja, tako da ne postoji mogućnost da disk ili kasetofon budu izvori). Ovakav editor je sasvim dovoljan za rad sa programskim jezicima, pri čemu je posebna pogodnost mogućnost editovanja direktnih komandi. Priložena je i tabela komandi koje editor razume (reč tekst označava liniju koja se edituje, a reč linija jednu fizičku liniju na ekranu).

Pogled na datu tabelu ne otkriva naročito bogatstvo naredbi, mada one omogućuju prilično komforan rad. Na žalost, „amstradov" editor je programski skoro potpuno zatvoren. To znači da mu se akcije mogu izmeniti samo na nekim neprirodnim mestima — u ovom slučaju to su linkovi za grafičko štampanje karaktera i uključivanje-isključivanje kursora. Ako neko bude želeo da prilagođava ovaj editor svojim potrebama ili dodaje nove naredbe, biće mnogo lakše da napiše novi.

Ugrađeni bejzik interpreter koristi prikazani editor za unošenje i ispravljanje programa. Na mikroračunarima je ustaljena praksa da se editori implementiraju u okviru programskih jezika, pa je prilično čudno što se ovaj nalazi u okviru operativnog sistema. Naravno, ovo je pogodna okolnost, jer editor može da se koristi i za neke druge programske jezike (šteta je što to HISOFT nije uradio za paskal i asembler — ili nisu imali potrebnu dokumentaciju za ovaj, već ugrađeni, ili neće da odustanu od svojih linijskih editora). A zgodno je imati i kalkulator pri ruci.

Dakle, sve je to lepo i krasno, ali pitamo se na šta je utrošeno punih 16K (do poslednjeg bajta) gornjeg ROM-a. Ono što nudi interpreter konstruktori „galaksije" bi, verovatno, smestili u neka 4 kilobajta. Po svemu sudeći, programeri Lokomotiv softvera se nisu baš pretrgli pišući bejzik za „amstrad", jer kako inače objasniti toliku razliku u kvalitetu ova dva programa, ugrađena u isti računar. Ostaje nam samo da čekamo dan kada će se proizvođači računara poštenije odnositi prema svojim kupcima.

Dejan Muhamedagić

OD CR (ENTER)	kraj editovanja i indikacija za prihvatanje izmena, tj. teksta
10 DLE (CLR)	briše znak ispod kursora, a tekst s desne strane se pomera ulevo
7F DEL (DELETE)	briše prvi znak s leve strane, a tekst s desne strane se zajedno sa kursorom pomera ulevo
EO COPY (COPY)	znak ispod kursora za čitanje se prepisuje u editovanu liniju
E1 INS (CTRL TAB)	flip-flop koji određuje insert ili overwrite mod
FO WUP (↑)	kursor za pisanje nagore
F1 WDN (↓)	kursor za pisanje nadole
F2 WLT (←)	kursor za pisanje ulevo
F3 WRT (→)	kursor za pisanje udesno
F4 RUP (SHIFT ↑)	kursor za čitanje nagore
F5 RDN (SHIFT ↓)	kursor za čitanje nadole
F6 RLT (SHIFT ←)	kursor za čitanje ulevo
F7 RRT (SHIFT →)	kursor za čitanje udesno
F8 BEG (CTRL ↑)	kursor za pisanje na početak teksta
F9 END (CTRL ↓)	kursor za pisanje na kraj teksta
FA STA (CTRL ←)	kurosr za pisanje na početak linije u kojoj se nalazi
FB FIN (CTRL →)	kursor za pisanje na kraj linije u kojoj se nalazi
FC BRK (ESCAPE)	kraj editovanja i indikacija da se korisnik odlučio da ulazni tekst ostane neizmenjen

Pri pokušaju da se izvede nešto nedozvoljeno (izlazak kursora za pisanje iz teksta koji se edituje, ili, recimo, pokušaj brisanja kada je linija prazna) javlja se zvučni signal. Ukoliko je editovani tekst „prazan", kursor za pisanje se može slobodno pomerati po čitavom prozoru.

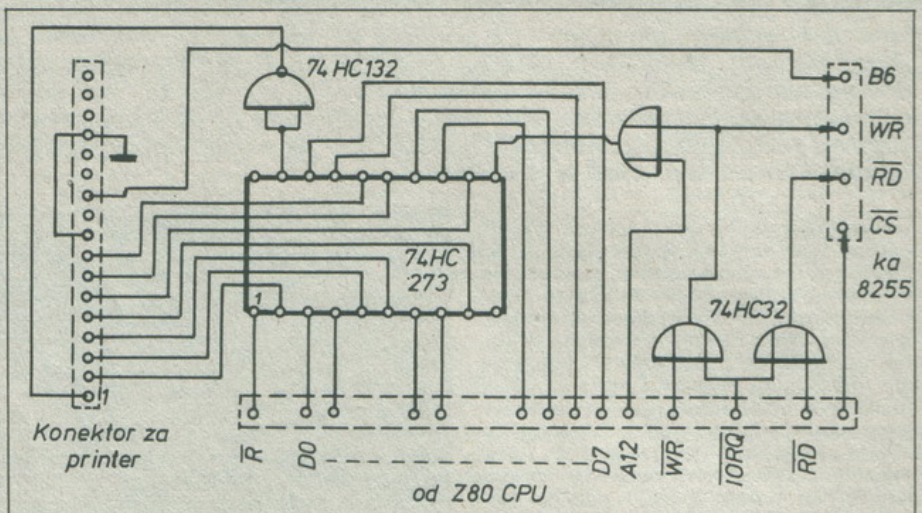
Računari u domaćoj radinosti
Amstrad/šnajder

paralelni interfejs za štampač

Rešenje paralelnog interfejsa koji je primenjen u računarima „amstrad" ima jedan neprijatan nedostatak koji u nekim slučajevima može u mnogome da ograniči rad: printeru se šalju samo sedam bita, dok je izvod na konektoru za osmi bit spojen na masu (logička nula). Kako su svi standardni alfanumerički znaci predstavljeni ASCII kôdom koji je manji od 128 (znači mogu se definisati sa sedam bita), to je za štampanje normalnog teksta ovakva veza sasvim zadovoljavajuća. Problem nastaje kad se žele štampati specijalni karakteri, čiji je ASICII kôd definisan sa osam bita, a koji se, pored standardnog seta znakova, takođe jalaze u ROM-u printera, ili se želi formirati sopstveni set znakova. Nedostatak osmog bita kod hard-copy programa prilično usporava rad.

Postoji više načina za otklanjanje ovog nedostatka, od kojih su neki tako jednostavni da je čudno da ih proizvođač nije ugradio. Kako verovatno ima veći broj vlasnika ovih računara kojima smeta, nedostatak osam bita ne izlazi za printer, ovde se prikazuju dva jačina kako se on može dovesti do printera.

Da bi izmene koje se navode bile jasnije, treba najpre videti kako je ovaj interfejs rešen u fabričkoj izradi računara. Mada je u računaru ugrađeno integralno kolo 8255 (paralelni ulazno-izlazni interfejs), ono je namenjeno za komunikaciju sa tastaturom, kasetofonom i generatorom zvuka. Samo se jedan bit (šesti) kanala B koristi za Centronics „busy" signal. Osnovu paralelnog interfejsa kod ovih računara sačinjava 8-bitni „latch" 74HC273, čijih je sedam bita iskorišćeno za podatak (kôd karaktera), a osmi služi za „strobe" signal (poruka printeru da ga na ulazu čeka podatak). Na slici 1. je prikazan deo električne sheme računara koji se odnosi na isterfejs. Propuštanje podatka na izlaz ovog leča vrši se aktiviranjem adrese &EF××, pri čemu se podatak sa magistrale podataka šalje na izlaz za prijter. Što se tiče programske rutine u ROM-u računara, ona je data na slici 2, a pozieva se iz bejzika sa PRINT#8.



Detalj električne sheme „omshoda" sa interfejsom za štampač Sl. 1

Novi interfejs . . .

Kako se iz programa vidi, omogućeno je da se ista naredba koristi i za startovanje programa koje korisnik sam piše, jer je veza ostvarena preko pozivne rutine operativnog sistema &BD2D. Iz istog listinga se, takođe, vidi (linije 210 i 260) da je ovim programom eliminisan osmi bit, što govori o tome da se ovaj deo programa ne može koristiti u našem slučaju bez izmene celog ROM-a, što je, sigurno, potpuno neprihvatljivo.

Jedan od načina da se na izlaz za printer izvedu svih osam bita je da se napravi poseban interfejs, koji bi se priključio na port za proširenje (za one koji ne žele da vrše nikakve ispravke u samom računaru), ili se u samom računaru ostvaruje povezivanje i smeštaj elemenata. Pri tom su potrebna tri integrisana kola — SN7406, SN74LS32 i 8255 — i tri elektrolita od 10 μ F, uz odgovarajuće konektore. Shema ovog dodatka je data na slici 3. Kao što se vidi, napajanje (+5 V) se uzima iz računara, tao da nije potreban dodatni stabilizovani ispravljač. Ovim rešenjem se dobija interfejs sa 24 ulazno-izlaznih kanala, kod kojih je samo 9 koriste za vezu sa printerom, dok se drugi mogu proizvoljno koristiti za druge namene. Sama činjenica da na raspolaganju ostaje 15 slobodnih kanala za razna upravljanja ili komunikaciju sa drugim uređajima opravdava investiciju i napor da se ovakav interfejs napravi.

Većinu gornjih adresa adresne magistrale (AB . . . A15) koristi sam računar. Korisniku za proširenja na raspolaganju ostaju samo A8, A9 i A10, odnosno adresa &FB, &FA i &FB. U konkretnom slučaju, je najbolje koristiti adresu &FB, jer se aktiviranje vrši samo logičkom nulom na A10. Kako je za komunikaciju sa izlazom potrebno više od jedne adrese, mora se u spoljašnjem interfejsu dekodirati i sadržaj registra C zajedno sa registrom B, odnosno neophodna je uvek 16-bitna port adresa. Iz raspoloživih adresa su pored A10, iskorišćene A0 i A1, (slika 3.), tako da će biti:

&FB00 — Kanal A ulaz/izlaz
&FB01 — Kanal B ulaz/izlaz
&FB02 — Kanal C ulaz/izlaz
&FB03 — upravljanje pomoću „upravljačkog bajta"

Upravljanje (određivanje stanja pojedi-

Bit 7: 1 — oznaka da se podatak koristi kao „upravljački bajt"
Bit 6: Izbor modusa grupe A: OO-modus 0; 01-modus 1;
Bit 5: 10 ili 11-modus 2
Bit 4: Kanal A 1-ulaz; 0-izlaz
Bit 3: Kanal C2 (gornje vrednosti) 1-ulaz; 0-izlaz
Bit 2: Izbor modusa grupe B: 0-modus 0; 1-modus 1
Bit 1: Kanal B 1-ulaz; 0-izlaz
Bit 0: Kanal C1 (donje vrednosti) 1-ulaz; 0-izlaz

```
Hisoft GENA3.1 Assembler. Page     1.

Pass 1 errors: 00

07E6                10          ORG  #07E6
07E6 21EC07         20 RESETP:  LD   HL,UL1
07E9 C3BA0A         30          JP   UL3
07EC 03             40 UL1:     DEFB 3
07ED F1BD           50          DEFW #BDF1
07EF C3FB07         60          JP   WAITPR
07F2 C5             70 PRINTC:  PUSH BC
07F3 CDFB07         80          CALL WAITPR
07F6 C1             90          POP  BC
07F7 C9             100         RET
07F8 013200         110 WAITPR: LD   BC,#32
07FB CD1B08         120 UL2:    CALL BUSYPR
07FE 3007           130         JR   NC,SENDPR
0800 10F9           140         DJNZ UL2
0802 0D             150         DEC  C
0803 20F6           160         JR   NZ,UL2
0805 B7             170         OR   A
0806 C9             180         RET
0807 C5             190 SENDPR: PUSH BC
0808 06EF           200         LD   B,#EF
080A E67F           210         AND  #7F
080C ED79           220         OUT  (C),A
080E F680           230         OR   #80
0810 F3             240         DI
0811 ED79           250         OUT  (C),A
0813 E67F           260         AND  #7F
0815 FB             270         EI
0816 ED79           280         OUT  (C),A
0818 C1             290         POP  BC
0819 37             300         SCF
081A C9             310         RET
081B C5             320 BUSYPR: PUSH BC
081C 4F             330         LD   C,A
081D 06F5           340         LD   B,#F5
081F ED78           350         IN   A,(C)
0821 17             360         RLA
0822 17             370         RLA
0823 79             380         LD   A,C
0824 C1             390         POP  BC
0825 C9             400         RET
                    410 ;
0A8A                420         ORG  #0A8A
0A8A 4E             430 UL3:    LD   C,(HL)
0A8B 0600           440         LD   B,0
0A8D 23             450         INC  HL
0A8E 5E             460         LD   E,(HL)
0A8F 23             470         INC  HL
0A90 56             480         LD   D,(HL)
0A91 23             490         INC  HL
0A92 EDB0           500         LDIR
0A94 C9             510         RET

Pass 2 errors: 00

Table used:   108  from   166
```

Sl. 2

Ugradnja paralelnog (Centronics) interfejsa u kućne računare „amstrad/šnajder" CPC omogućava direktno priključivanje većine printera, bez potrebe za ne baš jeftinim dodacima, koje zahtevaju drugi računari slične klase. Ovo je, sigurno, jedna od prednosti računara „amstrad/šnajder", jer se ozbiljnija primena ne može ni zamisliti bez printera. Na žalost, konstruktori ovog popularnog računara doneli su jednu čudnu odluku koja zagorčava život njegovim korisnicima. Sve se, međutim, može popraviti . . .

nih kanala) integrisanog kola 8255 vrši se pomoću „upravljačkog bajta" i to po sledećoj šemi:

U konkretnom slučaju „upravljački bajt" ima vrednost &83, što znači da je: kanal A predviđen za izlaz, kanal C1 za ulaz, kanal C2 za izlaz, kanal B za ulaz uz modus 0.

Opis same sheme interfejsa je jednostavan: signal za izbor čipa se dobija direktno sa adrese A10 mikroprocesora Z80, odnosno porta za proširenje. Takođe se direktno povezuju i adrese A0 i A1. Za razliku od upisivanjai čitanja memorije, ovde su neophodna oba signala: IOWR i IORD. Oni se dobijaju preko dva OR kola iz WR, RD i IORQ. Sistemski RESET procesora je aktivno nisko, tako da se mora invertovati za upravljanje kolom 8255. DO . . . D7 sa porta se pajaju direktno sa ulazima za podatke B255. Napajanje svakog integrisanog kola se mora blokirati letovanjem elektrlita na konektor za printer u računaru (ukoliko se interfejs ugrađuje u računar) ili na poseban konektor.

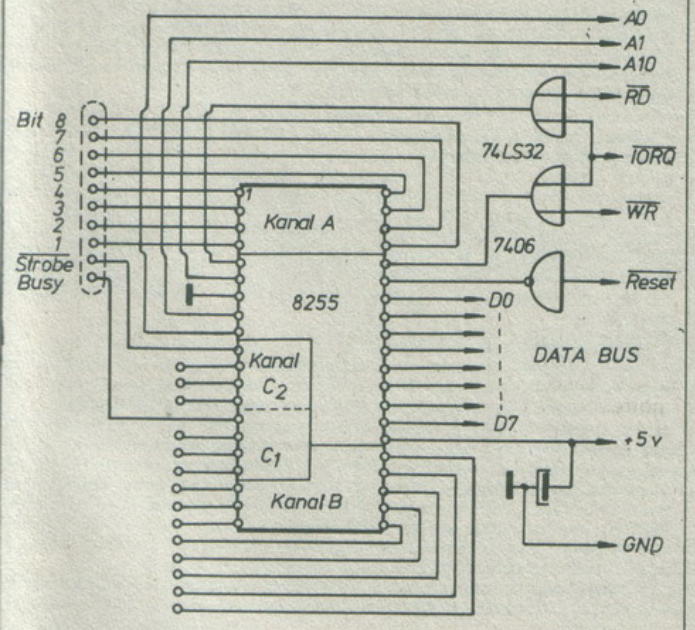
Pored ovog hardverskog dodatka, neophodan je i odgovarajući program kojim će se njim upravljati. Ovaj program je dat na sl. 4.

. . . ili mali rez

Drugi način je daleko jednostavniji. On se preporučuje svim onima koji se ne plaše da vrše izvesne dorade i izmene na samoj štampanoj ploči računara. Ove izmene su toliko male da ih mogu izvršiti i oni koji se nikad nisu bavili elektronikom. Važno je samo biti malo pažljiv pri letovanju, da se ne oštete drugi vodovi ili samo integrisano kolo. Za ovu izmenu je od dodatnog materijala potrebno samo nekoliko centimetara duga izolovana žica i, naravno, odgovarajuća električna lemilica (do 25 W).

Suština izmene se sastoji u sledećem: najpre se odviju 6 zavrtnjeva sa donje strane računara i odvoji gornji eo sa tastaturom i kasetofonom, rastavljajući dva konektora, ivod za osmi bit na konektoru za printer (izvod broj 9), koi je hardverski spojen sa masom, treba oštrim sečivom razdvojiti od mase (sl. 5) pri samom dnu. Zatim se izolovanom tankom žicom (najbolje licnastom) izvrši spajanje ovog izvoda (iznad prekida) sa vodom na štampanoj ploči koji ide od pina 12. integrisanog kola 8255 i to najbolje kod otvora gde se veza prenosi na drugu stranu. Pri letovanju treba paziti da se ne oštete veze na štampanoj ploči, kao i na to da se kalaj ne razlije po izvodu konektora i — izmena je gotova.

Pored ove hardverske izmene, potreban je i odgovarajući program koji šalje osmi bit na izlaz kada je to potrebno. Sistemska rutina u ROM-u računara tretira, naime, Centronics izlaz još uvek kao 7-bitni interfejs. Opisanom izmenom je osmi bit konektora printera spojen sa petim bitom kanala C integrisanog kola 8255, koji je sistemski predviđen za slanje podataka na kasetofon. Kako printer i kasetofon nikad ne rade istovremeno, moguće je odgovarajućom naredbom (iz bejzika: OUT &F600,32) ovaj bit setovati ili resetovati (OUT &600,0), u



Sl. 3

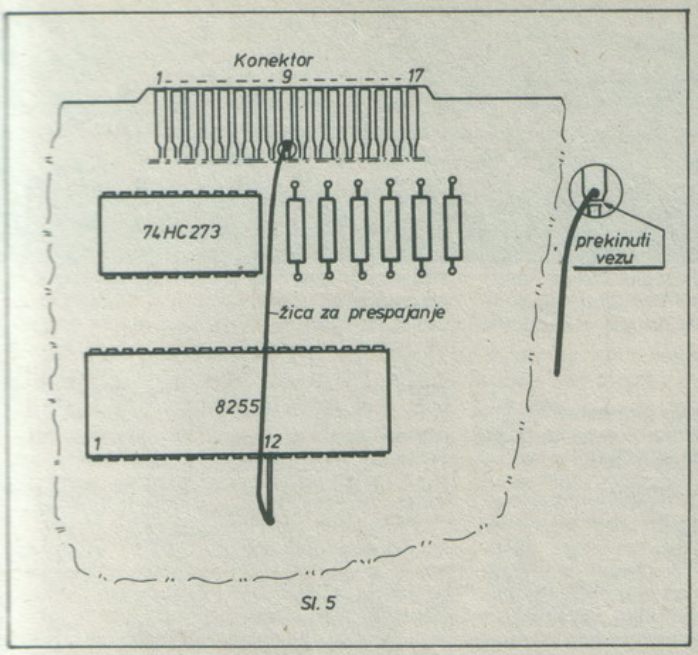
Električna shema novog interfejsa

Sl. 4

```
Hisoft GENAS.1 Assembler. Page     1.

Pass 1 errors: 00

                        10 ;       PROGRAM ZA 8-BITNI INTERFEJS
                        20 ;
                        30 ; Pre stratovanja  programa, neophodno  je da se
                        40 ; instrukcijom POKE izmene sadrzaji adresa i to:
                        50 ; BD29 sa &A037,  BD2C sa &A000,  BD2F sa &A02A
                        60 ; BD32 sa &A015  i  BDF2 sa &A006
                        70 ; i izvrsi initijalizacija sa: OUT &FB03,&83
                        80 ;
                        90 ;
A000                   100          ORG  #A000                ;Startna adresa
A000  C5               110 PRINTC: PUSH BC
A001  CD06A0           120          CALL WAITPR
A004  C1               130          POP  BC
A005  C9               140          RET
A006  013200           150 WAITPR: LD   BC,#32
A009  CD2AA0           160 UL2:    CALL BUSYPR
A00C  3007             170          JR   NC,SENDPR
A00E  10F9             180          DJNZ UL2
A010  0D               190          DEC  C
A011  20F6             200          JR   NZ,UL2
A013  B7               210          OR   A
A014  C9               220          RET
A015  C5               230 SENDPR: PUSH BC
A016  0100FB           240          LD   BC,#FB00             ;adresa 8-bitnog Dataport-a
A019  ED79             250          OUT  (C),A                ;slanje podatka na kanal A
A01B  0102FB           260          LD   BC,#FB02             ;adresa za "Handshake" (kanal C)
A01E  AF               270          XOR  A
A01F  F3               280          DI
A020  ED79             290          OUT  (C),A                ;Strobe "on"
A022  F680             300          OR   #80                  ;setovan 7.bit
A024  FB               310          EI
A025  ED79             320          OUT  (C),A                ;Strobe "off"
A027  C1               330          POP  BC
A028  37               340          SCF
A029  C9               350          RET
A02A  C5               360 BUSYPR: PUSH BC
A02B  D5               370          PUSH DE
A02C  57               380          LD   D,A
A02D  0102FB           390          LD   BC,#FB02             ;adresa za "Handshake" (kanal C)
A030  ED78             400          IN   A,(C)                ;Busy podatak (bit 0 kanala C)
A032  1F               410          RRA                       ;Busy podatak u CY (C-flag)
A033  7A               420          LD   A,D
A034  D1               430          POP  DE
A035  C1               440          POP  BC
A036  C9               450          RET
A037  2100A1           460 RESETP: LD   HL,UL1
A03A  4E               470          LD   C,(HL)
A03B  0600             480          LD   B,0
A03D  23               490          INC  HL
A03E  5E               500          LD   E,(HL)
A03F  23               510          INC  HL
A040  56               520          LD   D,(HL)
A041  23               530          INC  HL
A042  EDB0             540          LDIR
A044  C9               550          RET
                       560 ;
A100                   570          ORG  #A100
A100  03               580 UL1:    DEFB 3
A101  F1BD             590          DEFW #BDF1
A103  C306A0           600          JP   WAITPR
```



Sl. 5

Hardverska modifikacija „amstrada"

```
Hisoft GENA3.1 Assembler. Page      1.

Pass 1 errors: 00

                        10 ;       PROGRAM ZA 8-BITNI INTERFEJS-2
                        20 ;
                        30 ; Pre stratovanja  programa, neophodno  je da
                        40 ; se instrukcijom POKE izmeni sadrzaj adrese
                        50 ;  BDF1 sa  "C300A0"
                        60 ;
                        70 ;
A000                    80          ORG  #A000              ;Startna adresa
0B1B                    90 BUSYPR:  EQU  #0B1B
A000  013200           100 WAITPR:  LD   BC,#32
A003  CD1B0B           110 UL2:     CALL BUSYPR
A006  3007             120          JR   NC,SENDPR
A008  10F9             130          DJNZ UL2
A00A  0D               140          DEC  C
A00B  20F6             150          JR   NZ,UL2
A00D  B7               160          OR   A
A00E  C9               170          RET
A00F  C5               180 SENDPR:  PUSH BC
A010  06F6             190          LD   B,#F6              ; Kanal C IC 8255
A012  CB7F             200          BIT  7,A                ; karakter veci od 127?
A014  2B06             210          JR   Z,DALJE            ; ako nije, preskoci
A016  F5               220          PUSH AF
A017  3E20             230          LD   A,#20              ; 5. bit setovan
A019  ED79             240          OUT  (C),A              ; slanje podatka na kanal C
A01B  F1               250          POP  AF
A01C  01EF00           260 DALJE:   LD   BC,#EF
A01F  E67F             270          AND  #7F
A021  ED79             280          OUT  (C),A              ; podatak bez STROBE na izlaz
A023  F680             290          OR   #80
A025  F3               300          DI
A026  ED79             310          OUT  (C),A              ;Strobe "on"
A028  E67F             320          AND  #7F
A02A  FB               330          EI
A02B  ED79             340          OUT  (C),A              ;Strobe "off"
A02D  F5               350          PUSH AF
A02E  06F6             360          LD   B,#F6              ; Kanal C IC 8255
A030  3E00             370          LD   A,0
A032  ED79             380          OUT  (C),A              ; brisanje 5. bita kanala C
A034  F1               390          POP  AF
A035  C1               400          POP  BC
A036  37               410          SCF
A037  C9               420          RET

Pass 2 errors: 00

Table used:     74  from   198
```

Sl. 6

zavisno da li se šalje osmobitni ili sedmobitni podatak. Korišćenje bejzika je ne samo sporo već i prilično neudobno, pa je zato urađen program u „mašincu" koji se učitava u RAM na adresu &A000, a prikazan je na slici 6. Pritom se mora izmeniti i adresa skoka na adresi &BDF2 i to sa: poke &FD2,0 i poke 3BDF3, &AO.

Svetislav Zahar

Kako se unosi mašinski program u računar? Odgovor je vrlo jednostavan: na isti način kao što to radite sa bejzikom. Mnogi asembleri imaju svoj editor, pa odmah na početku treba savladati načine kako da se izlista grupa linija, kako da se ubaci ili izbaci neka od njih itd. Bejzik editorom smo uglavnom svi zadovoljni, na njega smo naviknuti pa nema potrebe da u tom delu tražimo nešto drugo. Dakle, linije se pišu sa linijskim brojem, program lista običnim LIST, snima se SAVE itd. Dovoljno je da napišete, na primer:

100 LDA #10

i dobili ste jednu liniju asembler programa. Da stvar bude lepša, ispred i iza nje se može nalaziti i običan bejzik program, što za sada može izgledati bez značaja, ali može i te kako biti korisno u praktičnom radu. Kako da se obeleži deo koji predstavlja asembler da bi ga PA preveo u niz mašinskih instrukcija? I ovo je vrlo jednostavno: ispred prve linije asembler programa ubacite SYS28672 (ako koristite kertriž verziju onda SYS32768). Najjednostavniji primer je dat na slici 1-P1.

Ako zadate RUN, bejzik će obaviti svoj posao u liniji 100, a zatim (110) asembler preuzima kontrolu i sledeće dve linije prevodi u niz brojeva: 169, 0, 141, 33, 208, 96, koji predstavljaju mašinski program. Ako hoćete da vidite šta ovaj program radi, ostaje da ga pozovete bejzik naredbom SYS. Sve je to jednostavno, ali gde je program u memoriji? Odgovor će vas iznenaditi: NIGDE. Zadatak dobrog asemblera nije samo prevođenje programa već i olakšavanje rada programeru. Jedna od takvih olakšica je i mogućnost da se prođe kroz program, proveri da li je sve u redu, ali da se smeštanje koda u memoriji ne obavi. Sada već nazirete da asembler može mnogo više, ali mu se mora staviti do znanja šta od njega očekujemo. Zato su uvedene posebne naredbe, takozvane pseudo instrukcije. Pišu se skraćeno sa tri slova, a da bi se razlikovale od asembler instrukcija mikroprocesora ispred se stavlja tačka.

Pseudonaredba .OPT

Osnovna naredba ovog tipa je OPTION koja se piše .OPT. Njenim parametrima se asembleru naređuje šta da uradi pri prevođenju: da li da samo proveri program, da li da kod smesti u memoriju itd. Kod PA je ovaj deo najkomplikovaniji, jer .OPT može imati do tri parametra u proizvoljnim kombinacijama, a neki od njih i više oblika. Zato ćemo, za početak, upoznati samo jedan — OO (dva slova oo). Ako se, dakle, negde u izvornom programu nađe .OPT OO, asembler će od tog trenutka započeti smeštanje mašinskog koda u memoriju. Primetite da smo upotrebili „negde u programu". Pseudo naredbe se mogu naći na proizvoljnom mestu i pojavljivati proizvoljan broj puta u jednom programu. Profi asembler ne postavlja nikakva ograničenja, pa je za sada dovoljno da zapamtite da je upotreba potpuno slobodna i da treba da bude prilagođena isključivo onome što vi od asemblera očekujete.

Izvršna adresa

Poznavanje ove opcije OPT naredbe (jasno, uz poznavanje samih mašinskih instrukcija) potpuno je dovoljno da napišete program koji radi. Njome smo obezbedili da se mašinski program nađe u memoriji. Gde će, međutim, on biti smešten? Ako se ne zada početna adresa, profi asembler će podrazumevati da treba da smesti kod od adrese 49152 ($COOO). Ako želite da počne od neke druge adrese, upotrebite naredbu koja jedina predstavlja izuzetak po načinu pisanja i izgleda ovako:

*=početna adresa

Znak * je specijalna varijabla kojoj znakom = dodeljujete početnu vrednost. Kada asembler započne sa prevođenjem, stalno će menjati njen sadržaj prema tome do koje adrese je stigao, pa ona predstavlja asemblerov PC (program counter). Sadržaj joj možete promeniti na bilo kom mestu u programu, pa različiti delovi mogu biti asemblirani na različite adrese. Znak „jednako" se generalno upotrebljava za dodelu vrednosti varijablama u programu. Specifičnost *= je u tome što ta dva znaka predstavljaju pseudo naredbu, pa se pišu **bez** razmaka — za sve ostale promenljive između naziva i znaka jednako obavezan je bar jedan razmak.

Promenljive u asembleru

Čemu, uopšte, promenjive u asembler programu? Druga upotreba je zamena nekih konstati varijablom kojoj se vrednost dodeljuje u početku, a zatim u celom toku programa koristi samo njen naziv. Na ovaj način se svaka izmena konstante u toku razvoja programa izvodi jednostavnom promenom vrednosti same varijable, a ne menjanjem konkretne vrednosti na svim mestima u programu gde se pojavljuje. Namerno smo rekli druga upotreba, jer se varijable primarno upotrebljavaju kao simboličke adrese. Svaki skok (JMP, JSR ili Branch instrukcija) zahteva adresu na koju će biti izveden. Izračunavanje konkretne vrednosti se može prepustiti asembleru ako se ispred instrukcije na koju se skok vrši postavi simbolička adresa, ili labela, kako se često naziva.

Primer programa koji će napuniti ekran kodovima od 32 do 128 i koji je već postao „klasika" za demonstriranje brzine mašinskog jezika dat je na slici 1 pod P2.

To je, otprilike, sve što je potrebno znati da bi započeli rad sa asemblerom. Međutim, prava priča o asembleru počinje tek sada.

Sistemi brojeva

Profi asembler raspoznaje tri sistema brojeva: heksadecimalni koji ima prefiks $, binarni sa prefiksom % i decimalni bez prefiksa. Moguće je zadati negativne brojeve i ASCII format. Nekoliko pravilnih oblika je dato na slici 1, P3.

Ni u jednom sistemu nije dozvoljeno prekoračenje maksimalne vrednosti koja se može predstaviti sa dva bajta ($FFFF ili 65535). Postoje i tri specijalna prefiksa sa sledećim značenjem:

< — niži bajt navedene vrednosti
> — viši bajt navedene vrednosti

profi asembler 64

kako to radi

Programski jezik u kome svaka instrukcija mikroprocesora ima svoj mnemonički ekvivalent naziva se asembler. Jedino ovaj programski jezik omogućava da se iskoristi apsolutno sve za šta je jedan mikroprocesor sposoban, ali je cena često prevelika — da bi se napisao program potrebno je uložiti enorman trud. Programeri na malim mašinama, bez obzira na sve teškoće, još uvek najradije posežu za ovim jezikom. Za vlasnike „komodora" pripremili smo uputstva za „Profi. ass 64", koji se može dobiti i preko „Eprom-servisa „Računara". Do ovog programa smo došli sasvim slučajno. Nije nam poznat ni autor, ni kuća koja ga je izdala, a, kako to obično biva kod nas, nismo uspeli da dođemo ni do originalnog uputstva. Verujući da se radi o dobrom programu, Zoran Životić je, jednostavno, izlistao ceo program i na osnovu toga izveo sopstveno upustvo, na kome je zasnovan i ovaj tekst.

! — forsiran dvobajtni format

Poslednji zahteva dodatno objašnjenje. Ako se u programu nađe instrukcija:

```
LDA 100
```

asembler će je prevesti u 165,100, što je naredba LDA ZERO PAGE. Ako želite instrukciju LDA ABSOLUTE, čiji bi kod za dati slučaj bio 173,100,0, onda asembler možete naterati da broj 100 smatra dvobajtnim, bez obzira što je viši bajt 0, upotrebom LDA !100.

Adresa ili vrednost može biti, kako smo već videli, zadata i promenljivom, ali i čitavim izrazom. Evo jednog „besmislenog" ali ilustrativnog primera koji će asembler sasvim korektno prevesti:

```
LDA #<(START+((BIT*16)!($FFFF ↑ FLG)+%01100000+„A") 3)
```

Aritmetičke i logičke operacije

Spisak dozvoljenih aritmetičkih i logičkih operacija dat je u tablici na slici 2. Dozvoljena je upotreba i običnih i uglastih zagrada — one imaju isti smisao pa izbor zavisi od vaših estetskih kriterija, ali se uglastim zagradama mogu izbeći nesporazumi jer se obične koriste i kao oznaka indirektnog adresiranja. Važno je znati da se izraz izračunava sleva nadesno bez ikakvih prioriteta operacija, osim onog uspostavljenog rasporedom zagrada. Sve operacije se obavljaju na standardan mikroprocesorski način, dakle po modulu 65636, pa će tako sabiranje, na primer, 65000+636 dati rezultat 0. Primećujete da se znaci ,< i ! koriste i kao operatori i kao prefiksi. Zato je važno da znate da se kao prefiks odnose na CEO izraz i moraju se nalaziti na početku. Čak i ako u prošlom primeru izbacite prvu i poslednju zagradu, asemler će kao rezultat dati vrednost nižeg bajta celog izraza, a neće uzeti samo niži bajt varijable START i zatim izvršiti ostatak računanja.

Promena dodeljenih vrednosti

Posebna promenjiva čija se vrednost može slobodno koristiti je već pomenuta *. Njeno korišćenje je ravnopravno sa ostalim koje sami definišete, a vrednost joj je jednaka adresi na kojoj će se naći instrukcija koja je koristi. Na primer:

```
LDA *+100
```

će, kada se prevede i startuje, smestiti u akumulator sadržaj memorijske lokacije udaljene 100 bajtova od same LDA instrukcije. Da bi ovo do kraja bilo jasno, reći ćemo da pojava labele u programu ima interni efekat:

```
LABELA=*
```

pa je gornji primer potpuno ekvivalentan sa:

```
LABELA LDA LABELA+100
```

Dodela vrednosti promenljivima znakom = se može vršiti samo jednom u programu. Ovo je razumljivo, jer se na taj način sprečava pojava dve iste varijable. Ako je potrebno da se vrednost promeni, za to se koristi specijalan znak dodele, strelica levo:

```
BROJAC=100
BROJAC <-BROJAC-1
```

Na prvi pogled, promena već dodeljene vrednosti nema smisla i mnogi asembleri je potpuno zabranjuju. Ipak, profi asembler poseduje nekoliko pseudo naredbi kojima mogućnost promene vrednosti varijabli znatno proširuje upotrebljivost.

Poruke i tablice

Tri pseudo naredbe, za razliku od ostalih, smeštaju neki kod u mašinski program kao da se radi o mikroprocesorskoj instrukciji. Ako je potrebno poruku ili tablicu reči smestiti u program, koristi se naredba sa dovoljno ilustrativnim nazivom .ASC. Asembler će u trenutku nailaska na nju tekst između znakova navoda koji sledi direktno smestiti u memoriju i to od mesta do kog je stigao u prevođenju (sadržaj varijable *). Jedan jednostavan primer je dat na slici 1, P4. Upotrebili smo mali trik — zadali smo da se rezultujući kod smesti od adrese 1024, što je početak video memorije, pa je način na koji .ASC naredba smešta kod u memoriju vidljiv

```
    SLIKA 1.

P1 100 PRINT"BEJZIK PROGRAM"
   110 SYS28672
   120 LDA #0
   130 STA 53281
   140 RTS

P2 100 SYS28672
   110 *= 10000
   120 .OPT OO
   130 LIMIT = 128
   140 START = 32
   150 LDY #START
   160 LDX #0
   170 NEXTCHR  TYA
   180 LOOP     STA 1024,X
   190 INX
   200 BNE LOOP
   210 INY
   220 CPY #LIMIT
   230 BNE NEXTCHR
   240 RTS
   RUN
   SYS10000

P3 $FF02
   #A
   %01101
   %0010110011011101
   23
   35287
   -12
   "A"
   itd.

P4 10 SYS28672
   20 *= 1024
   30 .OPT OO
   40 .ASC "TEKST....."

P5 10 SYS28672
   20 PASS = -1
   30 ..
   40 ..  PROGRAM 1.
   50 ..
   100 .PASS + PASS+1
   110 .IF PASS:.FIL dv,"naziv 2."

P6 10 SYS28672
   20 .OPT OO
   30 *= 820
   40 WAIT LDA 653
   50 BNE WAIT
   60 RTS

P7 10 OPEN 2,8,4,"naziv,p,w"
   20 SYS28672
   30 *= početna adresa
   30 .OPT O2
   40 .....

P8 330 COUNT + 1
   340 .BYT 32
   350 .IF COUNT-200:COUNT + COU
   NT+1:.GOT 340
```

odmah u toku asembliranja. Samo da napomenemo da se između znakova navoda može naći bilo šta što komodor inače podržava — dakle kontrolni karakteri, grafički karakteri i sl.

Za pravljenje tablica brojeva služe naredbe .BYT i .WOR od kojih prva smešta po jedan bajt, a druga celu reč (dva bajta) u standardnom formatu niži—viši bajt. Dve stvari čine ove naredbe vrlo korisnim: (1) mogućnost da se više parametara, odvojenih zarezom, unese u okviru jedne naredbe i (2) za parametre važe sva pravila o prefiksima, izrazima, brojnom sistemu koji su ranije pomenuti. Tako je sasvim uobičajen neki od ovih oblika:

```
.BYT <MSG1,<MSG2,...
.BYT >MSG1,>MSG2,....
```

ili

```
.WOR TEST+$FF,* — 10,.....
```

Listanje programa

Ovaj nivo korišćenja asemblera završićemo još jednom opcijom .OPT naredbe. Dokumentacija je jedna od važnih stvari pri radu u mašinskom jeziku. Bez obzira na kvalitet, bejzik editor se baš ne može pohvaliti preglednošću listinga. Kod asemblera je preglednost labela, adresa i instrukcija važna stavka, pa profi asembler poseduje svoj ispis. Ako u programu zadate .OPT P, na ekranu će biti ispisan listing programa, onako kako se prevodi; 40 znakova jedne linije je ipak malo da stanu svi podaci, pa je ispis razlomljen u dva reda i ponekad se teško snači. .OPT P je, zbog toga, prvenstveno namenjena za štampanje, ali za ovu priliku mora da ima i poseban oblik. Izabrano je elegantno rešenje koje omogućuje pravljenje odlične dokumentacije. Da bi izlistali program na printeru, treba da počnete ovako:

```
10 OPEN 11,4,0
20 SYS 28672
30 .OPT P11
```

Dakle, asembler će izlistati program preko bilo koje veze koju ste otvorili u bejziku i čiji logički broj zadate (bez razmaka) uz slovo P. Samim tim ispis se može uputiti i u neku datoteku na disku ili kasetofonu radi dalje obrade, na primer u tekst procesoru. Nakon upoznavanja sa P opcijom, vaši će programi najčešće započinjati naredbom:

.OPT OO,P

Programi iz delova

Svaki složeniji program se lakše piše razbijeno u više celina. Profi asembler u potpunosti podržava ovakav rad. Potrebno je imati mogućnost da se vrednost varijabli prenosi iz jedne celine u drugu i automatiku u učitavanju pojedinih delova. Prvi zadatak se može obaviti na više načina. Na raspolaganju su dve naredbe:

.SST dv,sa, "naziv" — snimi varijable

.LST dv,sa,,,naziv" — učitaj varijable

pri čemu je **dv** broj uređaja koji se koristi kao spoljna memorija, a **sa** sekundarna adresa. Kod ovih naredbi bi se mogla uputiti prva ozbiljnija zamerka asembleru. Naime, predviđen je rad preko serijskog interfejsa, praktično samo sa diskom, pa će **dv** uvek biti 8, sem kod retkih koji rade sa dva diska. Vlasnici kasetofona ne moraju previše da se uzbuđuju. Ima dovoljno načina da se ovaj posao ipak obavi. Naredbe .FIL i .END će nešto dužim putem izvršiti ovaj zadatak. Metod je jednostavan. Umesto da iz jedne celine snimite sve varijable i zatim ih upisujete sa LST u svim drugim delovima, moraćete svaki put da propuštate sve delove kroz asembler, ali će posao biti obavljen automatski, pa se jedino može govoriti o gubitku vremena. Pretpostavimo da jedan veliki program treba razbiti na manje delove, od kojih će prvi deo biti definisanje svih potrebnih konstanti u programu. Na primer:

```
10 SYS28672
   20 CHROUT=65490
   30 RANGE=65
   ITD.
```

Ako ovaj deo završite naredbom:

.END dv,,,naziv sledećeg dela"

asembler će proći kroz njega, interno formirati sve varijable koje se pojavljuju, upisati sledeći deo i asemblirati ga raspoznajući sve labele iz prvog dela. Očigledno da bez diska posao na razvijanju, recimo 4 dela, može biti jako komplikovan, pa se za rad sa kasetofonom ipak preporučuje što više smestiti u jednu celinu. Turbo tejpom se program može slobodno snimati, ali .END koristi isključivo C64 format, što zahteva da delove pišete gotovo bez greške, jer svaka ispravka i ponavljanje procesa asembliranja može da potraje.

Još jedna naredba ima potpuno istu sintaksu i zadatak, a to je .FIL. Male razlike ipak postoje. Originalno uputstvo za profi asembler bi, verovatno, razjasnilo razlog postojanja obe, iz samog listinga programa to nije bilo moguće. Jedina funkcionalna razlika je u tome što se .FIL može izvesti u prvom i drugom prolazu asembliranja, dok .END radi isključivo u prvom. Prvi i drugi prolaz?

Prvi i drugi prolaz(i)

Iako smo objašnjenje rada asemblera ostavili za kraj, ne znači da se radi o nečem komplikovanom. Programi se mogu pisati i bez detaljnog poznavanja rada profi asemblera, ali će to znanje omogućiti bolje korišćenje njegovih osobina.

Da bi asemblirao program, profi asembler mora da ga analizira dva puta. U prvom prolazu se samo proverava sintaksa naredbi i instrukcija (još se ne prevode u kod) i izvodi najvažniji deo — formiranje vrednosti varijabli i labela. Razlog je vrlo jasan. Ako svaka varijabla treba da se koristi bilo gde u programu, neophodno je imati sve njihove vrednosti u trenutku samog prevođenja instrukcija. U suprotnom bi bilo nemoguće obraćanje, na primer skok na labelu, unapred. Kada se ovaj posao obavi, u drugom prolazu se vrši prevođenje u mašinski kod. Šta će sa njim biti urađeno zavisi od izabranog parametra .OPT naredbe.

Što se tiče pseudonaredbi, one se izvode u zavisnosti od same instrukcije — neke u prvom, neke u drugom, a najčešće u oba prolaza. Poznavanje ove njihove osobine je presudno za razumevanje dejstva. Da bi to ilustrovali, vratićemo se opet na .END i .FIL. Ako se .FIL izvodi u oba prolaza, šta će se desiti kada profi asembler naiđe na nju? Prvo, to se dešava odmah u prvom prolazu, a do tada su formirane samo varijable, i to one koje se do tog trenutka pojavljuju. .FIL izvodi upisivanje sledećeg programa (dakle, prvi deo se gubi) i profi asembler počinje njegovo asembliranje od početka. Dakle, posledica je pomalo neprijatna, ako je niste očekivali. Iz prvog dela će biti postavljene varijable, ali ako se u njemu našla i neka instrukcija ona jednostavno neće biti prevedena (to se radi u drugom prolazu). Ujedno, početna adresa za asembliranje drugog dela, ako je niste zadali očekujući da se kod nastavi na prethodni, biće ista kao za prvi deo, pa labele drugog dela neće imati mnogo smisla, jer program ne počinje od prave adrese. .END obavlja isti zadatak kao .FIL, ali samo u prvom prolazu, dakle uvek na opisan način. Očigledno je potrebno naći put, ako je već dozvoljeno, da se FIL izvede u drugom prolazu.

Uslovno asembliranje

Postoji jedna važna osobina dodele vrednosti varijabli. Ako se to radi znakom =, onda se obavlja samo u prvom prolazu dok se promena vrednosti znakom dodele strelica levo obavlja u oba. Nije, dakle, neizvodljivo da primenom nekoliko linija kao:

```
150 PASS=1
160 PASS < PASS+1
```

GREŠKE U TOKU PREVOĐENJA PROGRAMA

0****SYNTAX	Nije dozvoljena samo labela bez instrukcije
1****SYNTAX	Nekorektan naziv labele
3****SYNTAX	U izrazu upotrebljen operator koji ne postoji
4****SYNTAX	Nekorektan raspored zagrada u izrazima
5****SYNTAX	Nekorektan prefiks (dozvoljeno !,<>,$,%) Nekorektan parametar .OPT naredbe (dozvoljeno O,P,N)
6****SYNTAX	Nekorektni parametri .SST, .LST, .FIL, .END Parametri .BYT i .WOR nisu odvojeni zarezom
7****SYNTAX	Nepostojeća pseudo-instrukcija
9****SYNTAX	Pogrešan način adresiranja. Upotrebljeni način adresiranja ne postoji na procesoru 6510.
*ILLEGAL QUANTITY	Zadati broj izlazi van opsega (bajt ili reč)
*OVERFLOW	Linija predugačka, suviše parametara u .BYT ili .WOR
*BRANCH OUT OF RANGE	Relativni skok (branch instrukcije) van opsega)
*REDEFINITION	Pojava dve iste labele (varijable)
*UNDEF'D STATEMENT	Adresiranje na labelu koja nije definisana
*!!SYM TABLE OVERFLOW	Tablica labela „silazi" ispod granice zadate sa .STM
*!!OUT OF MEMORY	Program i labele zahtevaju više memorije od raspoložive količine
*!!UNDEF'D STATEMENT	.GOT na nepostojeću liniju
*!!DEVICE NOT PRESENT	Periferni uređaj nije priključen
*!!IEEE	Greška u komunikaciji sa periferijom (ako STATUS <>0)
*!!DISK ERROR	Greška na disk jedinici. Ispisuje se poruka sa ERROR kanala.

Pojava greške ne zaustavlja proces prevođenja ali mašinski kod u najvećem broju slučajeva nije korektan. Ako je greška u adresnom delu, kod instrukcije će biti pravilno izveden ali će adresa imati vrednost 0. Loša labela obično prouzrokuje više grešaka, sve instrukcije koje je koriste takođe proizvode grešku.

Poruku o greški profi asembler ispisuje PRE linije na kojoj je nastala.

na osnovu stanja varijable PASS saznamo u kom prolazu je asembler. Sad još samo mogućnost uslovnog izvođenja naredbi i..... Ako ste pomislili na neku konstrukciju tipa IF, sasvim ste u pravu. Profi asembler zaista poseduje .IF, ali sa malo skraćenim oblikom: ispituje se (u oba prolaza) da li je vrednost izraza jednaka O ili ne. Dakle, nema poređenja tipa veće—manje i ostalih kombinacija. Namerno smo rekli izraza, a ne varijable, jer za vrednost koja se ispituje važe svi mogući oblici numeričkih, logičkih i ostalih izraza koje dopušta profi asembler. Ako je vrednost različita od nule, biće izvedene instrukcije iza .IF, u suprotnom se preskače cela linija. Sintaksa IF ne zahteva upotrebu THEN, potrebno je samo:

.IF izraz: naredba:naredba.....

Dve tačke iza izraza nisu nikakva specifičnost ove konstrukcije. Profi asembler dozvoljava, kao i bejzik, odvajanje naredbi ili instrukcija ovim znakom, pa u jednu liniju može stati dosta toga. Iz iskustva vam savetujemo da ovo ne zloupotrebljavate, bar ne u početku, jer sigurno vodi potpunoj nepreglednosti i teškom snalaženju pri ispravkama programa. Dovoljno će biti da se ograničite na jednu ili dve instrukcije u jednom redu. Dakle, deo koji će obezbediti potpuno povezivanje (i labela i koda) više programa treba da izgleda kao na slici 1, P5.

Ako umesto .FIL upotrebite .END, neće se desiti ništa jer, .END radi samo u prvom prolazu. U drugom prolazu će imati isti efekat kao i .END bez parametara, kojim se završava asembler program i vraća kontrola bejziku.

Pseudonaredba .IF ne bi bila kompletna bez GOTO, što je bilo jasno i autorima PA. Zato postoji .GOT naredba koja kao parametar zahteva linijski broj na koji od tog trenutka treba preći sa asembliranjem. .GOT se izvodi u oba prolaza, o čemu treba posebno voditi računa. Prva logična upotreba je u fazi izrade kada treba neki deo asembler programa privremeno izbaciti. Da ga ne bi brisali, pa kasnije ponovo upisivali, ispred dela zadajte .GOT na prvu liniju iza njega i to je sve. Daleko je interesantnija kombinacija .IF .GOT kojom se mogu lako ostvariti složeni zadaci. Pitanje: „Kako napraviti tablicu od 200 bajtova koji inicijalno treba da imaju vrednost 32? Odgovor „Upotrebiti dve stotine .BYT 32 instrukcija" se ne priznaje zbog nehumanog odnosa prema programeru. Ostaje elegantno rešenje sa slike 1, P8. U primeru su značajne dve stvari. Prvo, upotrebljena je dodela vrednosti znakom strelica levo. Tako će COUNT u liniji 330 dobiti vrednost 1 i pri drugom prolazu. Da je upotrebljeno = desilo bi se da nakon 350 linije, kada vrednost dostigne 200, varijabla COUNT ostane na toj vrednosti i u drugom prolazu uslov za ponavljanje .BYT instrukcije već u startu ne bude ispunjen, pa samim tim bajt 32 bude smešten u memoriju samo jednom. Treba, takođe, napomenuti da u tom slučaju ni eventualne labele nakon ovakvog dela ne bi u drugom prolazu bile korektne, jer im je vrednost formirana kao da će dve stotine bajtova

NAREDBE I SINTAKSE PROFI-ASS. 64.

LABELE — DO 8 znakova, prvo slovo
a zatim slovo ili broj
* — standardna labela PC

PREFIKSI, OPERATORI I OZNAKE

$	heksadecimalni	<	niži bajt	+	sabiranje
%	binarni	>	viši bajt	−	oduzimanje
"	ASCII	!	forsiran dvo-	·	množenje
bez p.	decimalni		bajtni format	&	AND
				!	OR
	; — komentar			↑	EOR
	= — dodela vrednosti varijabli			<	rotiranje levo
	←— dodela i promena vrednosti var.			>	rotiranje desno

naredba	prolaz	sintaksa	komentar
*=	1 2	*=adresa	dodela vrednosti PC (adresa za asembliranje)
. BYT	2	. BYT bajt, bajt,. .	tablica bajtova
. WOR	2	. WOR, reč, reč,. .	rablica reči (dva bajta)
. ASC	2	. ASC „ascii niz"	tablica ASCII karaktera
. OPT	1 2	. OPT O	smešta kod od adrese STREND (49)
. OPT	1 2	.OPT OO	smešta kod od adrese *=
. OPT	1 2	. OPT On	smešta kod u datoteku sa logič. brojem n
. OPT	1 2	. OPT O=adresa	za svaku instrukciju poziva maš. p. na adresi
. OPT	1 2	. OPT P	ispis listinga na ekran
. OPT	1 2	. OPT Pn	ispis u datoteku sa logičkim brojem n
. OPT	1 2	. OPT P=adresa	pre ispisa svakog karaktera poziva mašinski program
. OPT	1 2	. OPT N	poništava sve ranije. OPT
. OPT		. OPT N,OO,P	dozvoljeno više parametara odvojenih zarezom
. END	1 2	. END	kraj asembler programa
. END	1	. END dv, „naziv"	kraj programa i upis sledećeg
. FIL	1 2	. FIL dv, „naziv"	upis sledećeg asembler programa
. SST	1	. SST dv, sa, „naziv"	snima varijable (simboličke adrese)
. LST	1	. LST dv, sa, „naziv"	učitava varijable (simboličke adrese)
. STM	1 2	. STM adresa	ograničenje prostora za varijable do adrese n
. SYS	1 2	. SYS adresa	poziv mašinskog programa na adresi
. IF	1 2	. IF izraz:naredbe	ako izraz <>0 izvedi naredbe
. GOT	1 2	. GOT n	asembliranje prenesi na liniju n
. HTB	1 2	. HTB	izlazak iz asembliranja preko BWS vektora (bejzik warm start)

zaista biti u memoriji, dok će stvarno smeštanje koda početi odmah nakon samo jednog bajta 32. Druga stvar je ispitivanje dostignute granice. Pošto .IF registruje samo dva slučaja: jednako ili različito od nule, potrebno je izraz tako svesti da se uslov može ispitati ovim vrednostima. U primeru je to jednostavno izvedeno oduzimanjem 200, ali ponekad se ceo izraz može dosta iskomplikovati. Ne treba zaboraviti da vrlo korisno mogu da posluže logičke i shift operacije, ali tada morate da „razmišljate binarno".

Ako ste u stanju da primenite sve do sada rečeno, neće vam biti teško da shvatite ostatak naredbi iz tablice koju dajemo u prilog, a koje ovde nismo posebno objasnili. Ostaje nam samo da vas upoznamo sa nekim tehničkim detaljima koji mogu olakšati rad.

Finese naredbe .OPT

Treba znati da profi asembler koristi 35 bajtova memorijskog prostora iz sistemskog dela od adrese 57. Sadržaj ovih lokacija se privremeno čuva u INPUT baferu (od 512). Ako asembliranje prekinete sa STOP+RESTORE (što, inače, nije neophodno profi asembler podržava prekid samo STOP tasterom) originalne vrednosti neće biti vraćene pa izvođenje neke od bejzik funkcija, na primer PEEK, može blokirati računar. Tada vam ostaje da odmah snimite program, inicijalizujete računar i tek onda nastavite rad.

Varijable se smeštaju od najviše slobodne adrese naniže, kao što to bejzik radi sa stringovima. One formirane u asembleru se ne raspoznaju u bejziku, a na žalost, isto važi i obratno.

```
Slika 3

10 REM ** ISPRAVKA PROFI-ASS.64 **
20 RESTORE:FOR F=29762 TO 29809
30 READB:POKEF,B:NEXTF
40 DATA 6,76,38,77,176,48,101,76
50 DATA 170,169,0,101,77,176,39,72
60 DATA 138,72,162,2,6,76,38,77
70 DATA 176,28,202,208,247,104,101,76
80 DATA 133,76,104,101,77,176,15,133
90 DATA 77,234,234,234,234,234,234,234
```

Spisak svih parametara .OPT naredbe je dat u tablici, ali oblik .OPT P=adresa ili .OPT O=adresa zahteva dodatno objašnjenje. Radi se o tome da su autori profi asemblera, što je vrlo korektan potez, ostavili mogućnost da se vašim programom ubacite u proceduru ispisa ili asembliranja. Na primer, listanje izazvano opcijom P se odvija na standardan C64 način, dakle brzo uz malu mogućnost usporavanja CTRL tasterom. Za analizu listinga ovo je i dalje prebrzo, pa se može upotrebiti sledeći metod. Kreirajte mali program (slika 1, P6). Asemblirajte ga, obrišite i počnite sa radom na vašem programu, ali za ispis upotrebite opciju .OPT P=820. Sada će profi asembler pre svakog ispisanog znaka izvesti JSR 820 i time pozvati malopre kreirani program. Njegov zadatak je da se vrti u praznoj petlji ako je pritisnut neki od tastera SHIFT, C= ili CTRL, čime se listanje potpuno zaustavlja. Pri ulasku u ovu rutinu u akumulatoru se nalazi bajt koji se ispisuje, što ovde nije iskorišćeno, ali se može upotrebiti za posebna usmeravanja ispisa i slično.

Mehanizam koji se primenjuje na O=adresa se razlikuje utoliko što se skok na adresu vrši pre smeštanja celog koda jedne instrukcije i potpuno se preskače smeštanje od strane profi asemblera. Broj bajtova instrukcije se nalazi u lokaciji 78, sam kod počinje od 75, ali ako je duži od tri bajta (eventualno kod .WOR, .BYT i .ASC) ostatak počinje od adrese 347. Nije baš jednostavno ali se retko i koristi. Prvo što padne na pamet bi bilo relocirano asembliranje, ali je i ono podržano od strane .PA, pa ova opcija uglavnom završava na zadovoljstvu što postoji, ako ikad zatreba.

Relocirano asembliranje se izvodi opcijom .OPT O. Navođenje početne adrese se i dalje obavlja sa *= i program će biti asembliran kao da treba da radi od nje, ali će sam kod biti smešten od adrese na koju pokazuje sistemska varijabla bejzika STREND (49, 50). Ako koristite disk, elegantnije rešenje omogućava opcija .OPT On, gde je n logički broj veze otvorene u bejziku. Efekat je sličan opciji .OPT Pn, ali se sada ne usmerava listing u datoteku već sam rezultujući mašinski program (slika 1, P7). Na ovaj način se na disku dobija gotov program koji jednostavno treba upisivati sa LOAD „naziv",8,1. Početna adresa može imati bilo koju vrednost, jer program ne dodiruje memoriju u fazi asembliranja. Na žalost, ovako kreirana datoteka je sekvencijalnog tipa (a ne program), pa iako se na disku uz otvaranje sa ",P,W" lako prevodi u programsku, na kasetofonu je to nemoguće i kreirana datoteka neće moći da se upisuje naredbom LOAD.

Verujemo da su vam sada potpuno jasni razlozi zbog kojih ovaj program smatramo odličnim. Treba dodati i to da su autori uložili dosta truda da omoguće i neke „sitne" detalje kao što je upotreba bejzik službenih reči kao varijabli. Pošto se pri unošenju linija odmah vrši tokenizacija, na primer PRINT labela će biti predstavljena jednim bajtom, a ne nizom slova P R I N T. Ipak, profi asembler će se ponašati sasvim korektno, pa je upotreba svakog naziva dozvoljena, čak i pseudo naredbi ako ne počinju tačkom, na primer OPT=100.

Nije korektno prema autorima programa da na samom kraju spominjemo ozbiljnu grešku koju su napravili, ali nas tok ovog teksta na to prisiljava. Pri prevođenju niza decimalnih cifara u binarni broj, na jednom mestu je zaboravljeno resetovanje CARRY zastavice, pa se određeni brojevi prevode na vrednost za jedan veću od stvarne. U verziji prilagođenoj za kertridž kojim redakcija programira EPROME ova greška je ispravljena, ali možete isto uraditi i sa standardnom verzijom primenom programa sa slike 3. Upišite asembler, ukucajte i startujte dati program, i zatim snimite ispravnu verziju profi asemblera.

Zoran Životić

Matematički softver

To može i bolje (7)

arkus funkcije

Ovde su proučavane mogućnosti upotrebe pojedinih formula za izračunavanje arkus funkcija koje se preporučuju u literaturi ili su ugrađene u osnovnu programsku podršku računara. Dvanaest priloženih crteža nedvosmisleno pokazuju da se istraživanju valjanih algoritama za izračunavanje vrednosti matematičkih funkcija nije poklanjala potrebna pažnja. Uz analizu kako ne treba i zašto tako ne treba programirati ove funkcije, priloženi su programi koji pokazuju kako se mogu valjano izračunavati vrednosti arkus funkcija. Na apscisi priloženih crteža uvek je u pogodnoj razmeri argument, a na ordinati je tačnost izračunavanja merena brojem tačnih bitova mantise. Iako je razmatranje opštije, programi su napisani za računar koji ima 32 bita mantise (J=32), na primer „šarp MZ-700". Za „komodor 64" i „spektrum 48" u programe treba na odgovarajućim mestima ubaciti reči THEN i LET.

Arkussinus

Funkcija arkussinus arcsin(x) je definisana za abs(x)<=1. Funkcija arcsin(x) je rastuća od $-\pi/2$ do $\pi/2$. Za male vrednosti modula argumenta abs(x) može se uspešno aproksimirati formulom

$$(1)\quad \arcsin(x) \sum_{k=0}^{n} (2K)!x^{2k+1}/2^{2k}K!^2(2K+1).$$

Za razne vrednosti gornje granice sumiranja tačnost formule (1) predstavljena je na slici 1. Slika 1 pokazuje da je za 32 binarne cifre mantise i za argumente po modulu manje od 1/2 potrebno uzeti n=12. Primenom ekonomizacije ovaj broj n može se smanjiti uz izmenu vrednosti koeficijenata, ali to je posebna tema. Ako se nekome ne čini da je 14 množenja i sabiranja mnogo za 32 binarne cifre mantise (skoro 10 decimalne cifre) za abs(x)<=1/2 može koristiti formulu (1) sa n=12.

Naravno, ostaje teži deo problema: kako računati arcsin(x) za 1/2<abs(x)<=1. Pri krajevima intervala definisanosti izvod funkcije neograničeno raste pa treba očekivati da je u tim oblastima izračunavanje vrednosti funkcije otežano.

Poznatom formulom

$$(2)\quad \arcsin(x)=\pi/2-\arcsin(\mathrm{sqr}(1-x^2))$$

moguće je izračunati arcsin(x) za sqr(1/2)<x<=1 ako se zna ta funkcija za abs(x)<=sqr(1/2). To bi značilo da ranija aproksimacija (1) za n=12 nije dovoljna, već treba povećati n. Greška formule (2) predstavljena je na slici 2, gde je uočljivo da se skoro četvrtinu broja bitova gube na kvadriranje argumenta i oduzimanje tog kvadrata od jedinice. Povećanje greške

```
10 LABEL "ARCSIN"
11 IF ABS(X)<.00002772720598# Y=X:RETURN
12 Z=X
13 Y=SQR((.5-X+.5)*(.5+X+.5))+1D-36
14 X=X/Y
15 GOSUB 31
16 X=Z
17 RETURN

20 LABEL "ARCCOS"
21 IF ABS(X)<.00140844685#
   Y=1.57079632679#-X:RETURN
22 Z=X
23 X=SQR((.5-X+.5)/(.5+X+.5+1D-36))
24 GOSUB 31
25 Y=Y+Y
26 X=Z
27 RETURN

30 LABEL "ARCTG"
31 Y=ABS(X)
32 IF Y<.00002772720598#Y=X: RETURN
33 IF Y>2147483648#
   Y=1.57079632679#:GOTO 41
34 IF Y<.2305203097# U=0:V=0:GOTO 38
35 IF Y<.80814465159# U=.453125:
   V=.48691509945#:GOTO 38
36 IF Y<2.1352928688# U=.90625:
   V=1.2764618289B2#: GOTO 38
37 U=1.359375:V=4.65920712#
38 V=(Y-U)/(Y*V+1)
39 W=-V*V
40 Y=(((((.0736*W+.09090909)*W+
   .111111111#)*W+.142857429#)*W+
   .2)*W+.3333333333#)*W*V+V+U
41 IF X<0 Y=-Y
42 RETURN
```

izračunavanja preko sto puta je, naravno, nedopustivo.

Sledeća formula

$$(3)\quad \arcsin(x)=\pi/2-2\ \arcsin(\mathrm{sqr}((1-x)/2))$$

omogućava izračunavanje funkcije arcsin(x) za 1/2<x<=1 ako je funkcija poznata za 0<=x<1/2. Na slici 3 data je greška formule (3) koja potiče od nebrižljivog oduzimanja 1−x. Ovo oduzimanje se mora obaviti brižljivije, npr. sa 0.5−x+0.5. Na taj način moguće je izračunati funkciju arcsin(x) za 0<=x<= 1.Za −1<=x<0 funkcija se lako izračunava, jer je neparna

$$\arcsin(x)=-\arcsin(-x).$$

Rasprostranjena je zabluda da se arcsin(x) može računati pomoću arccos(x) formulom

$$(4)\qquad \arcsin(x)=\pi/2-\arccos(x),$$

naravno pod pretpostavkom da se funkcija arccos(x) valjano računa. Da je to zabluda upečatljivo pokazuje slika 4,jer je uočljivo da se za male vrednosti modula argumenta ne dobija nijedna tačna cifra rezultata. To je jedan od brojnih primera analitički tačne formule koja je numerički neupotrebljiva.

Računarski programi najčešće sadrže formulu

$$\arcsin(x)=\arctg(x/\mathrm{sqr}(1-x^2))$$

Greška ove formule je kao na slici 2; ona pretežno potiče od kvadriranja argumenta i od oduzimanja tog kvadrata od jedinice. Poslednja formula nije definisana za abs(x)=1, jer nije dozvoljeno deliti nulom. Za x=1 ili x=−1 treba uzeti da je

$$\arcsin(x)=(\pi/2)\mathrm{sgn}(x).$$

U „spektrum 48" firma je ugradila formulu

$$\arcsin(x)=2\arctg(x/(1+\mathrm{sqr}(1+\mathrm{sqr}(1-x^2))))$$

koja ima približno istu grešku kao formula (2). Kod te formule nema opasnosti od deljenja nulom za x=1 ili x=−1, ali je ona složenija od prethodne formule.

Sledećom formulom pokušava se da se smanji greška izračunavanja argumenta korena

$$\arcsin(x)=\arctg(x/\mathrm{sqr}((1-x)\ (1+x))).$$

greška je kao na slici 4. Treba imati u vidu da se čak do polovine bitova mantise izgubi samo zbog greške na poslednjem bitu u oduzimanju 1−x ili sabiranju 1+x. Da se smanji ogromna greška argumenta korena može se (1−x) (1+x) napisati tananije sa (.5−x+.5) (.5+x+.5). Da se izbegne testiranje argumenta x na 1 ili na −1, dovoljno je korenu fiktivno dodati 1E−36. Ovo sabiranje obično se ne vrši zbog velike razlike u vrednosti binarnih karakteristika sabiraka, pa je veoma brzo. Sabiranje se vrši samo za x=1 ili x=−1; tada je zbir matematički gledano pogrešan — ali šta to mari kad je krajnji rezultat tačan.

Prema tome, funkciju arkussinus treba računati formulom

$$\arcsin(x)=\arctg(x/(\mathrm{sqr}((.5-x+.5)(.5+x+.5))+1.E-36)).$$

Za male vrednosti modula argumenta, kada je $x^3/6$ zanemarljivo u odnosu na x, treba koristiti formulu arcsin(x)=x.

Za abs(x)>1 trebalo bi javiti grešku. To nije naročito potrebno uraditi iz ovog programa, jer u tom slučaju program za izračunavanje kvadratnog korena javlja grešku.

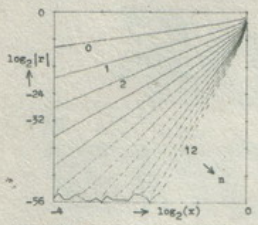
Arkuskosinus

Funkcija arkuskosinus arccos(x) definisana je za abs(x)<=1. Funkcija arccos(x) je opadajuća od π do 0, nije ni parna ni neparna. Kako važi formula

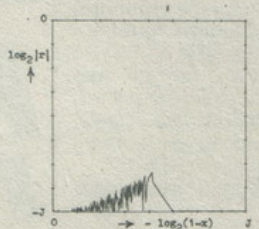
$$\arccos(x)=\pi-\arccos(-x),$$

sleduje da se za negativne argumente lako računa ako je poznata za pozitivne.

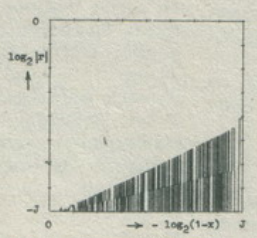
Arkus funkcije arcsin(x), arccos(x), arctg(x) pripadaju matematičkom softveru računara. Teško je naći računar kod kojeg su ove funkcije predstavljene korektnim programima. U seriji „To može i bolje" prof. dr Dušan Slavić izlaže detalje svog dvadeset godina dugog istraživanja u numeričkoj matematici.



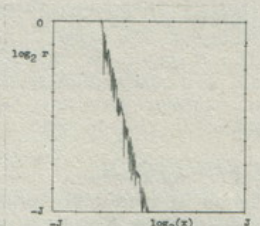
Slika 1. Relativna greška formule
$\arcsin(x) \approx \sum_{k=0}^{n} (2k)!x^{2k+1}/2^{2k}k!^2(2k+1)$
u funkciji argumenta x.



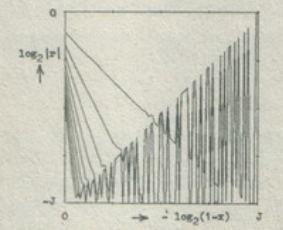
Slika 2. Relativna greška formule
$\arcsin(x) = \mathrm{arctg}(x/\mathrm{sqr}(1-x^2))$
u funkciji argumenta x.



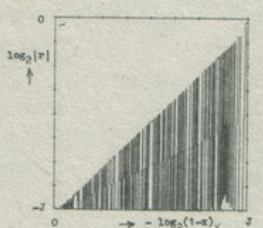
Slika 3. Relativna greška formule
$\arcsin(x) = \pi/2 - 2\arcsin(\mathrm{sqr}((1-x)/2))$
zbog nepreciznog oduzimanja 1-x.



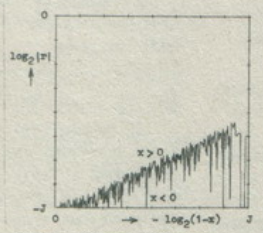
Slika 10. Relativna greška formule
$\mathrm{arctg}(x) = \arccos(1/\mathrm{sqr}(1+x^2))$ $(x \geq 0)$
ako se arccos i sqr računaju tačno.



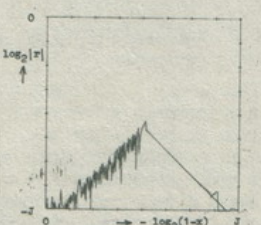
Slika 5. Relativna greška formule
$\arccos(x) \approx \mathrm{sqr}(2-2x)\sum_{k=0}^{n} \frac{(2k)!(1-x)^k}{2^{3k}\,k!^2\,(2k+1)}$
u funkciji argumenta x.



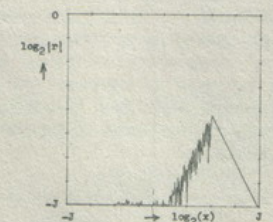
Slika 6. Relativna greška formule
$\arccos(x) = 2\arcsin(\mathrm{sqr}((1-x)/2))$
zbog nepreciznog oduzimanja 1-x.



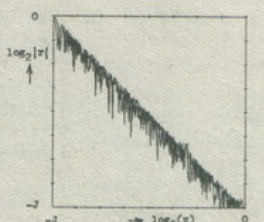
Slika 7. Relativna greška formule
$\arccos(x) = \pi/2 - t$
ako se t = arcsin(x) računa tačno.



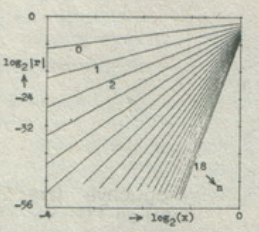
Slika 8. Relativna greška formule
$\arccos(x) = \mathrm{arctg}(\mathrm{sqr}(1-x^2)/x)$
ako se sqr i arctg računaju tačno.



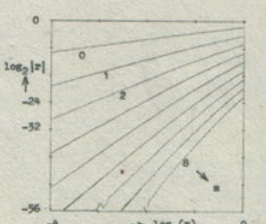
Slika 9. Relativna greška formule
$\mathrm{arctg}(x) = \arcsin(x/\mathrm{sqr}(1+x^2))$
ako se arcsin i sqr računaju tačno.



Slika 4. Relativna greška formule
$\arcsin(x) = \pi/2 - t$
ako se t=arccos(x) tačno računa.



Slika 11. Relativna greška formule
$\mathrm{arctg}(x) \approx \sum_{k=0}^{n} (-1)^k x^{2k+1}/(2k+1)$
u funkciji argumenta x.



Slika 12. Relativna greška formule
$\mathrm{arctg}(x) \approx \frac{x}{1} + \frac{1^2x^2}{3} + \frac{2^2x^2}{5} + \cdots + \frac{n^2x}{(2n+1)}$
u funkciji argumenta x.

Uopšteni potencijalni razvoj

(5) $\arccos(x)=\mathrm{sqr}(2-2x)\sum_{k=0}^{n} (2K)!(1-x)^k/2^{3k}k!^2(2k+1)$

pokazuje se kao numerički nestabilan; videti sliku 5. Kada x raste ka 1 tačnost opada na nula tačnih cifara mantise. To što analitička greška sa porastom broja sabiraka opada slaba je uteha. Ukupna greška je zbir analitičke i numeričke, a ona čini da formulu (5) moramo proglasiti za numerički neupotrebljivu.

Iz formula (1) i (5) sleduje formula

(6) $\arccos(x)=2\arcsin(\mathrm{sqr}((1-x)/2))$.

Pod pretpostavkom da su programi za kvadratni koren i arkussinus idealno dobro načinjeni, greška formule (6) predstavljena je na slici 6. Greška potiče od nebrižljivog oduzimanja $1-x$.

Kod većine računara izračunavanje funkcije arccos(x) svodi se na izračunavanje funkcije arcsin(x) pomoću formule

(7) $\arccos(x)=\pi/2-\arcsin(x)$.

Da je taj jednostavan metod pogrešan vidi se na slici 7: kada x raste od 0 ka 1 broj tačnih bitova mantise opada od J na J/2. Izgubiti polovinu tačnih cifara zbog jednostavne formule — to nije naročito mudro.

Kod nekih velikih računara, za $0<x<=1$, koristi se formula

(8) $\arccos(x)=\mathrm{arctg}(\mathrm{sqr}(1-x^2)/x)$.

Pod pretpostavkom da se funkcije kvadratni koren i arkustangens idealno tačno računaju greška formule (8) data je na slici 8. Greška potiče od kvadriranja argumenta i od oduzimanja tog kvadrata od jedinice.

Pa kako onda treba računati arkuskosinus? Treba, na primer, pomoću formula (6) i (8) dobiti formulu

$\arccos(x)=2\mathrm{arctg}((1-x)/(1+x))$
$(-1<x<=1)$

pa se pomoću sličnog eksperimenta uveriti da ni ona (iako brža od prethodnih) nije imuna od numeričke greške.

Zatim treba nepromišljeni izraz (1−x)/(1+x) zameniti boljim (.5−x+.5)/(.5+x+.5), čime se radikalno smanjuje greška.

Uslov da x nije −1 može se eliminisati ako se imeniocu doda neki mali broj, npr. 1E−36.

Valjana formula postaje

arccos(x)=2arctg(sqr((.5−x+.5)/(.5+x+.5+1E−36))).

U okolini tačke x=0 celishodno je usvojiti aproksimaciju arccos(x)=π/2−x, a ona važi sve dok je $x^3/6$ zanemarljivo u odnosu na π/2. Tim testom se neznatno usporava program ako uslov nije ispunjen, ali se znatno ubrzava program ako je uslov ispunjen.

Arkustangens

Funkcija arkustangens definisana je za sve realne vrednosti argumenta. Ona raste od −π/2 (za x teži minus beskonačno) do π/2 (za x teži plus beskonačno). Za velike vrednosti modula argumenta, veće od recipročne vrednosti elementarne promene mantise, može se usvojiti aproksimacija

arctg(x)=(π/2)sgn(x).

Funkcija arkustangens je neparna

arctg(x)=−arctg(−x),

pa je jednostavno vrednost funkcije negativnog argumenta svesti na vrednost funkcije pozitivnog argumenta.

Za male vrednosti modula argumenta, ako je $x^3/3$ zanemarljivo u odnosu na x, moguće je arctg(x) aproksimirati sa x. Time se ako je uslov ispunjen znatno ubrzava izvršavanje programa, a ako uslov nije ispunjen neznatno se usporava.

Gledano analitički moguće je funkciju arkustangens izračunati pomoću funkcije arkussinus, npr. sa

(9) arctg(x)=arcsin(x/sqr(1+x^2)).

Da to nije preporučljivo dokaz je slika 9: pod pretpostavkom da su programi za arkussinus i kvadratni koren idealno tačni, formulom (9) gubi se i do polovine binarnih cifara mantise. Najveća greška je za abs(x) $2^{i/2}$. Može se reći da je arkustangens moguće računati pomoću valjanih programa za arkussinus i kvadratni koren samo za abs(x)<=1.

Za x>1 vrednost arctg(x) može se izračunati pomoću vrednosti arkustangensa za argumente od 0 do 1 pomoću formule

arctg(x)=π/2−arcrg(1/x).

Uz neparnost funkcije arctg(x), rečeno je dovoljno za svođenje funkcije arkustangens na proste aproksimacije i funkcije arkussinus i kvadratni koren. Ispitivanje numeričke greške pokazuje da je ipak bolje izračunavanje vrednosti funkcije arkussinus svesti na arkustangens, nego obrnuto.

Pod pretpostavkom da se funkcije arkuskosinus i kvadratni koren idealno tačno računaju, na slici 10 predstavljena je tačnost formule

(10) arctg(x)=arccos(1/sqr(1+x^2)) (x>=0).

Slika pokazuje da je formula (10) izrazito loša za abs(x)<1, za abs(x)<$2^{i/2}$ ne dobija se nijedna tačna cifra rezultata. Formula (10) bi se mogla koristiti za računanje arctg(x) samo za abs(x)>1. Nevolja je jedino u tome što se (bar do sada) valjan program za arccos(x) ne može načiniti bez funkcije arctg(x), a ne obrnuto.

Očigledno funkciju arkustangens treba računati bez pozivanja funkcija arkussinus, arkuskosinus ili kvadratni koren.

Na slici 11 data je tačnost formule

(11) $$\text{arctg}(x) \sum_{k=a}^{n} (-1)^k x^{2k+1}/(2k+1) \quad (\text{abs}(x)<=1)$$

za razne vrednosti gornje granice sumiranja n. Za zadani interval primene formule i zadanu tačnost računara sa slike 11 čita se koliko treba uzeti sabiraka u razvoju. Ako najveća dozvoljena vrednost za abs(x) teži 1, broj sabiraka neograničeno raste. Dakle, potencijalni razvoj (11) može se koristiti samo za vrednosti abs(x) manje od 1.

Na slici 12 predstavljena je tačnost formule

(12)
arctg(x)=x/(1+1^2x^2/(3+2^2x^2/(5+
. . .m^2 x^2/(2m+1) . . .)))

koja je posebno dobra za veće tačnosti računara.

U literaturi se često funkcija arctg(x) aproksimira samo za abs(x)<=sqr(1/3), jer važi formula

arctg(x)=π/3+arctg((x−sqr(3))/(sqr(3)x+1))

gde je x>sqr(3).

Uz neparnost funkcije arctg(x) i pomenute aproksimacije, poslednja formula je dovoljna. U tom slučaju je n=13, odnosno m=7. Dakle, ako se želi brže izračunavanje ne sme se interval funkcije (−π/2, π/2) deliti samo na tri podintervala. Za abs(x) <=tan(29/128) i J=32 sa slika 11 i 12 dobija se n=6 i m=5; što odgovara podeli intervala funkcije na sedam podintervala. Naravno, ekonomizacijom se može uštedeti neka operacija množenja i sabiranja.

Svođenje proizvoljnog argumenta na osnovni interval obavlja se formulom

arctg(x)=arctg(a)+arctg((x−a)/(ax+1)).

Program sadrži izračunate vrednosti za arctg(a) i a. Ako je broj tih izračunatih vrednosti veliki, tj. broj podintervala veliki, onda je izračunavanje arctg(x) brzo — jedino je konstatovanje u kom se podintervalu nalazi argument x dugotrajno; to znači da ni sa brojem podintervala ne treba preterivati. Ako je broj podintervala mali, lako je svođenje na osnovni interval — ali je izračunavanje arkustangensa dugotrajno; ne treba preterivati ni sa štednjom izračunatih vrednosti za arctg(a) i a. Ova protivurečna situacija rešava se u skladu sa opredeljenjem da li treba više štediti na memoriji računara ili na brzini rada programa. Za tačnost od 32 binarne cifre mantise (skoro 10 značajnih cifara rezultata) celishodno je usvojiti da se interval funkcije (π/2, π/2) deli na sedam podintervala. Za veće tačnosti računara trebalo bi da broj podintervala bude veći.

U priloženim programima provedeni su rezultati ovde datih razmatranja. Funkcije arkussinus i arkuscosinus izračunavaju se pomoću funkcije arkustangens. Za vrednost arkustangensa u osnovnom intervalu primenjena je ekonomizacija, koja će biti obrađena drugom prilikom.

Dušan Slavić

Numerički metodi (3)

interpolacioni polinomi

U prethodnim nastavcima „Numeričkih metoda" razmatran je zadatak interpolacije u kome na osnovu poznatih

$$x_0, x_1, x_2, \ldots, x_n \qquad (1)$$

$$f(x_0)=y_0,\ f(x_1)=y_1,\ \ldots,\ f(x_n)=y_n \qquad (2)$$

najpre treba odrediti neku funkciju F(x) koja zadovoljava uslov

$$f(x_0)=F(x_0),\ \ldots\ f(x_n)=F(x_n) \qquad (3)$$

a zatim traženu nepoznatu vrednost f(x) za zadato x izračunati kao F(x). Opisana je linearna interpolacija (F je linearna funkcija), uveden opšti pristup polinomijalnoj interpolaciji (F je polinom), i dat metod, program i procena greške za Lagranžovu interpolaciju.

Njutnov interpolacioni polinom

Kao i u slučaju Lagranžovog interpolacionog polinoma, i u Njutnovoj metodi se polinom zapisuje u pogodnom obliku koji omogućava neposredno dobijanje koeficijenata, bez potrebe da se svaki put rešava pridruženi sistem jednačina. Neka su tačke (1) na jednakom rastojanju h. Njutnov polinom ima oblik:

$$y(x_0+th)=y_0+\frac{t}{1!}\Delta y_0+\frac{t(t-1)}{2!}\Delta^2 y_0+\ldots+\frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_0 \qquad (4)$$

gde su tzv. konačne razlike uvedene sa:

$$\Delta y_0=y_1-y_0,\ \Delta y_1=y_2-y_1,\ \ldots \qquad \Delta^2 y_0=y_1-y_0,\ \Delta^2 y_1=y_2-y_1,\ \ldots \qquad (5)$$

i tako dalje, a nova promenljiva t definisana vezom:

$$x=x_0+th \leftrightarrow t=\frac{x-x_0}{h}$$

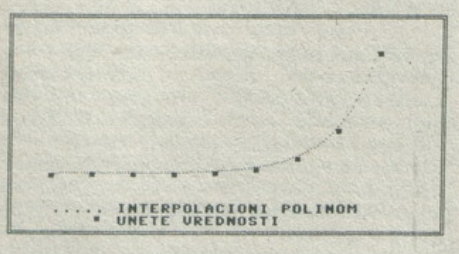
Čitaoci će lako sami napisati program koji izračunava vrednosti polinoma (4).

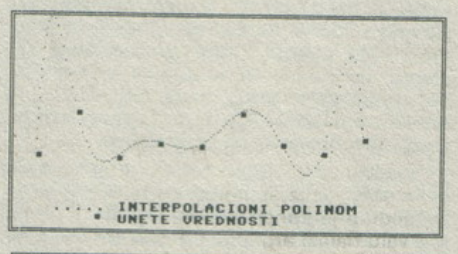
Aitkenov metod

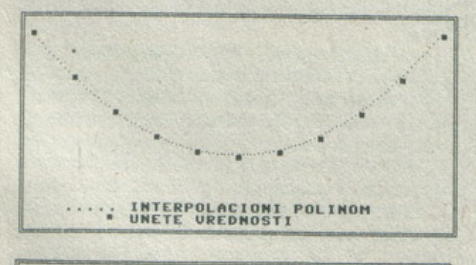
Navodimo program za polinomijalnu interpolaciju koji se zasniva na Aitkenovom postupku. U ovom postupku se primenjuje ideja iterirane interpolacije, odnosno uzastopnog interpoliranja, pri čemu se tačnost uvećava do tražene (ako je moguće). Pro-

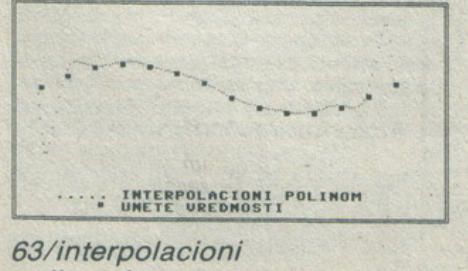
Mnogi konkretni problemi koji se rešavaju na računarima imaju u svojoj osnovi zadatke numeričke matematike. Numerički metodi poseduju veliku opštost i primenljivost, ali kako njihovo razumevanje i programiranje nije uvek sasvim jednostavno, to im upotreba, odnosno popularnost među vlasnicima mikroračunara, nije na nivou koji zaslužuju. Ova serija napisa ima za cilj da jednostavno, i samo u najvažnijim elementima, prikaže nekoliko metoda kojima se rešavaju neki osnovni zadaci numeričke matematike, kao i da ponudi programe koji su bazirani na tim metodama. Programi su pisani na bejziku za „komdor 64", ali se jednostavno mogu preneti na druge mikroračunare.

```
100 REM ****************************************
105 REM *                                      *
110 REM *              INTERPOLACIJA           *
115 REM *                                      *
120 REM *          METOD AITKEN-LAGRANGE       *
125 REM *                                      *
130 REM ****************************************
135 REM *          OPIS PROMENLJIVIH           *
140 REM ****************************************
145 REM *                                      *
150 REM * X-VREDNOST ARGUMENTA INTERPOLACIJE   *
155 REM *    (ULAZ)                            *
160 REM * A1-NIZ VREDNOSTI NEZAVISNO PROMENLJIVE *
165 REM *    (ULAZ)                            *
170 REM * A-PREUREDJEN NIZ A1                   *
175 REM * V1-ULAZNI NIZ VREDNOSTI FUNKCIJE      *
180 REM * V-PREUREDJEN NIZ V1                   *
185 REM * D-BROJ TACAKA U NIZOVIMA A I V (ULAZ) *
190 REM * T-GORNJA GRANICA APSOLUTNE GRESKE     *
195 REM *    (ULAZ)                            *
200 REM * Y-INTERPOLIRANA VREDNOST FUNKCIJE     *
205 REM *    (IZLAZ)                           *
210 REM *                                      *
215 REM ****************************************
220 PRINT CHR$(147)
225 PRINT" ************************************"
230 PRINT" * PROGRAM ZA INTERPOLACIJU FUNKCIJE *"
235 PRINT" *    JEDNE NEZAVISNO PROMENLJIVE   *"
240 PRINT" *                                  *"
245 PRINT" *       METOD AITKEN-LAGRANGE      *"
250 PRINT" ************************************"
255 M1=5
260 GOSUB775:REM IZDAVANJE PRAZNIH REDOVA
265 INPUT"  ( PRITISNITE RETURN DA NASTAVITE )";PR
270 PRINT CHR$(147)
275 DIM A1(100),V1(100),A(100),V(100),R(100)
280 REM ****************************************
285 REM *              ULAZNI PODACI             *
290 REM ****************************************
295 INPUT "UNESITE BROJ TACAKA";D:PRINT
300 PRINT"UNOSITE JEDAN ISPOD DRUGOG PAROVE"
305 PRINT"TACAKA  X(I),Y(I)":PRINT
310 FOR L=1 TO D
315 PRINT"    X(";L;"),Y(";L;") =";
320 INPUT A1(L),V1(L)
325 NEXT L
330 PRINT
335 PRINT"AKO JE I-TI RED POGRESAN, UNESITE I"
340 INPUT"AKO NEMA GREŠKE UNESITE 0";W
345 PRINT
350 IF W=0 THEN 380:REM TACNO UNETO, DALJE
355 IF W>D THEN 330:REM VECE OD DIMENZIJE, PONOVO
360 PRINT"UNESITE X(";W;"),Y(";W;") =";
365 INPUT A1(W),V1(W):REM UNOSENJE ISPRAVKE
370 GOTO 330
375 PRINT
380 INPUT "UNESITE TACNOST KOJU TRAZITE";T:PRINT
385 INPUT "UNESITE ARGUMENT INTERPOLACIJE";X
390 GOSUB 830:REM PREUREDJENJE A1 I V1
395 PRINT:PRINT
400 PRINT"********  UNOS PODATAKA ZAVRSEN ********
405 G=2:D2=0
410 IF D<1 THEN 525:REM DIMENZIJA 0
415 IF D=1 THEN 515:REM DIMENZIJA 1
420 REM ****************************************
425 REM *       POCETAK CIKLUSA ZA ITERACIJE     *
430 REM ****************************************
435 FOR J=2 TO D
440 D1=D2:K=J-1
445 FOR I=1 TO K
450 H=A(I)-A(J)
455 IF H=0 THEN 620
460 V(J)=(V(I)*(X-A(J))-V(J)*(X-A(I)))/H
465 NEXTI
470 D2=ABS(V(J)-V(K))
475 IF J<=2 THEN 495
480 IF D2 <= T THEN 570
485 IF J<5 THEN 495
490 IF D2 >= D1 THEN 590
495 NEXT J
500 REM ****************************************
505 REM *          ZAVRSETAK CIKLUSA           *
510 REM ****************************************
515 J=D
520 Y=V(J)
525 GOSUB 645:REM IZDAVANJE REZULTATA
530 M1=5
535 GOSUB 775
540 INPUT"(1)-NOVA INTERPOLACIJA, (2)-KRAJ";W
545 IF W<>1 THEN END
550 GOTO 375
555 REM ****************************************
560 REM *    DOVOLJNA TACNOST ZA D-1 ITERACIJU   *
565 REM ****************************************
570 G=0:GOTO 520
575 REM ****************************************
580 REM *  TEST VREDNOST D2 POCINJE DA OSCILUJE  *
585 REM ****************************************
590 G=1
595 J=K
600 GOTO 520
605 REM ****************************************
610 REM *       DVE ISTE VREDNOSTI U NIZU A      *
615 REM ****************************************
620 G=3
625 GOTO 595
630 REM ****************************************
635 REM *    IZDAVANJE REZULTATA I KOMENTARA   *
640 REM ****************************************
645 M1=5
650 GOSUB775
655 ON G+1 GOTO 660,675,695,735
660 PRINT"TACNUST DOSTIGNUTA"
665 GOSUB 795
670 RETURN
675 PRINT"TACNOST NIJE DOSTIGNUTA ZBOG GRESAKA"
680 PRINT"ZAOKRUZIVANJA."
685 GOSUB 795
690 RETURN
695 PRINT"BROJ TACAKA MANJI OD 3, I ZBOG TOGA"
700 PRINT"NIJE BILO MOGUCE PROVERITI TACNOST,"
705 PRINT
710 PRINT"ILI SE POMOCU DATIH VREDNOSTI"
715 PRINT"ZA X(I) I Y(I) TRAZENA TACNOST NE"
720 PRINT"MOZE DOSTICI (POVECATI BROJ TACAKA)"
725 GOSUB795
730 RETURN
735 PRINT"OTKRIVENE DVE ISTE VREDNOSTI"
740 PRINT"U NIZU X(I),"
745 PRINT
750 GOTO 880
755 REM
760 REM ****************************************
765 REM *       STAMPANJE M1 PRAZNIH REDOVA    *
770 REM ****************************************
775 FOR M=1 TO M1:PRINT:NEXT M:RETURN
780 REM ****************************************
785 REM *   IZDAVANJE INTERPOLIRANE VREDNOSTI  *
790 REM ****************************************
795 PRINT
800 PRINT"INTERPOLIRANA VREDNOST FUNKCIJE"
805 PRINT"Y =";Y:PRINT
810 RETURN
815 REM ****************************************
820 REM *   PREUREDJENJE ULAZNIH NIZOVA A1 I V1  *
825 REM ****************************************
830 FOR I=1 TO D:R(I)=ABS(X-A1(I)):NEXT I
835 FOR K=1 TO D
840 TE=1E38:I1=1
845 FOR I=1 TO D
850 IF R(I)>=TE THEN 860
855 TE=R(I):I1=I
860 NEXT I
865 R(I1)=1E38:A(K)=A1(I1):V(K)=V1(I1)
870 NEXT K
875 RETURN
880 END
```









63/interpolacioni polinomi

gram na ulazu zahteva vrednost za željenu tačnost. U programu su navedena značenja promenljivih.

Greške koje se čine kada se interpolacija vrši na bazi polinoma, mogu se procenjivati korišćenjem formula za grešku koje su date u prethodnim nastavcima. Formula za grešku kod Newtonovog polinoma ista je kao i kod Lagranžovog. Ali, da bi se dala i vizuelna ilustracija ponašanja interpolacionog polinoma, pogodno je dati grafički prikaz za tipične slučajeve.

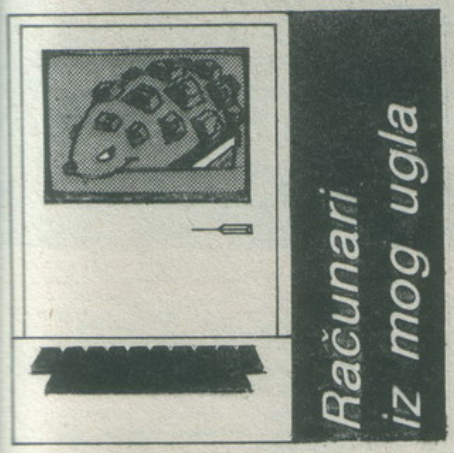
Na slici 1. prikazan je interpolacioni polinom (svejedno koji, Lagranžov ili Njutnov jer oni uzimaju iste vrednosti) za kvadratnu funkciju. U ovom slučaju greška je nula, jer je kvadratna funkcija specijalan slučaj polinoma, pa se aproksimira bez greške. (Napominjemo da su unete vrednosti predstavljene gornjim levim temenom kvadratića).

Na slici 2. je eksponencijalna funkcija koja nije polinom i kod koje se već javlja greška. Greška je veoma uočljiva na slici 3. gde je prikazana sinusna funkcija. Interpolacioni polinom znatno odstupa na krajevima intervala, što je tipično za ovu vrstu interpolacije.

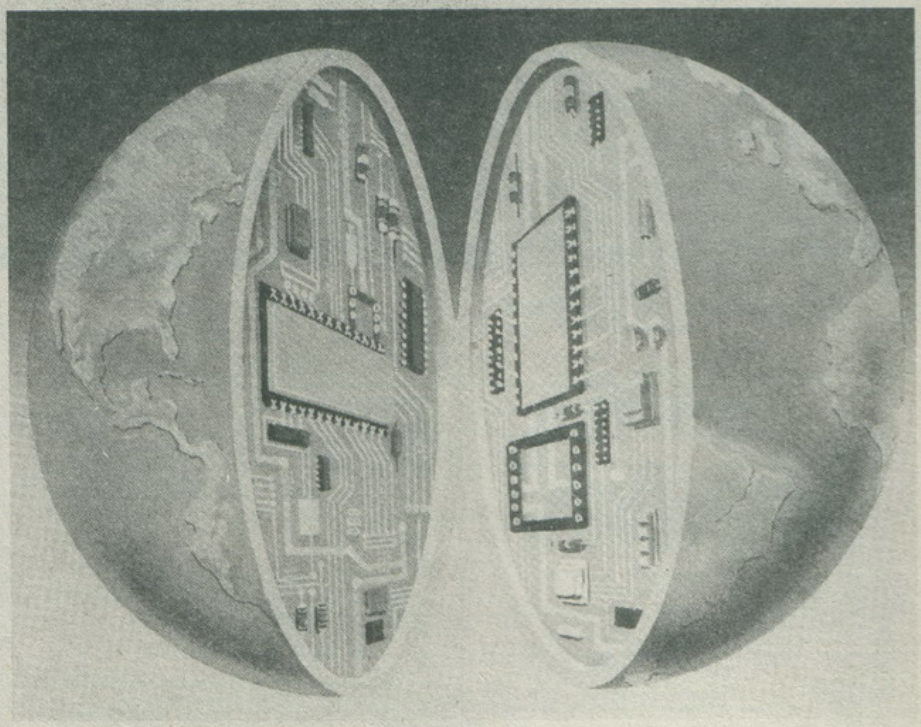
Na slici 4. je prikazan izmišljen eksperiment, tj. tačke su proizvoljno uzete. Tu se takođe, vidi oscilovanje interpolacionog polinoma na krajevima intervala. Treba istaći da pored polinomijalne interpolacije postoje i druge vrste interpolacija o kojima nije bilo reči. To su interpolacija racionalnom funkcijom, verižnim razlomkom, splajn interpolacionom funkcijom itd. Veoma često su ove druge vrste interpolacija pogodnije i daju bolje rezultate. Primera radi, na slici 5. su isti podaci sa slike 4. interpolirani splajn funkcijom (u poslednje vreme veoma popularan metod koji zaslužuje posebno razmatranje). Očigledno je da je aproksimacija „prirodnija".

Podvucimo još jednom da izbor interpolacionog metoda zavisi od tipa podataka na osnovu kojih se vrši interpolacija.

Mr Veljko Spasić



KOMPJUTER U NAŠOJ MALOJ FIRMI



Dvehiljadite godine svako će imati svoj računski centar.

Osnovni problem malih računara je u tome što su mali.

Veliki privredni promašaji posmatraju se sa ekonomsko-političkog stanovišta, a nikako da se shvati da su megalomanske investicije pre svega nečiji psihološki problem.

Od megalomanije obično oboljevaju ambiciozni, a nesposobni rukovodioci. Mučeni svojim ništavilom, vrebaju priliku za velike role. Pošto kompjuter ima, pre svega, statusnu vrednost, a tek onda upotrebnu, megalomanski direktor male firme će pokušati sebi da podigne značaj izgradnjom računskog centra. Mali računari ne dolaze u obzir. Sa malim se ne može biti zadovoljan čak ni ako vrši posao.

Kada mala firma počne da zida računski centar, to je jedan od pouzdanijih znakova da su stvari krenule loše. Uostalom, to ne mora biti nagoveštaj, nego i uzrok propasti, zajedno sa još nekom velikom investicijom. I sve to zato što direktor ima potrebu da sazida spomenik sebi za života.

Prodavci „domaćih" velikih računara svakako znaju za tu slabost nekih direktora — inače teško da bi uspeli nekom da uvale tako skupu robu. Oni znaju kako mali direktor Perica zamišlja računski centar: od računara se sve može saznati pritiskom na dugme. Ponudiće mu na fini način kompjuter kao sredstvo kontrole svih onih koji pokušavaju da ga skinu. Sve važne odluke u OOUR-ima moraće da se objedine pod njegovim mudrim rukovodstvom.

Ni sam direktor nije svestan svoje nesvesti. On bi najviše voleo neki IBM, makar polovan, prastari model, pa da može da prošeta poslovne prijatelje kroz 5—6 soba, a oni da blenu u lampice što svetlucaju i magnetne trake koje se vrte.

Naša surova realnost, gde veliki računari ne mogu da se uvoze, nateraće direktora da organizuje sastanak sa drugovima iz tvrtke DRP-DATA, našeg poznatog „proizvođača" računara.

Ovakvim sastancima treba da prisustvuju samo direktor i njegovi najbliži saradnici, a nikako neki mlad inženjer koji bi mogao da pita zašto sve to košta 5—10 puta više nego u inostranstvu. Iskusni žderači poslovnih ručkova nikada neće postaviti tako neumesna pitanja. Cene, performanse sistema, softver — to su sitnice koje mogu da obeshrabre drugove iz DRP-DATE, i dovedu u pitanje prijateljski poslovni ručak i uspešan nastavak dalje saradnje.

Sledeći vrlo važan korak je progurati celu stvar kroz samoupravnu proceduru. Srećom, postoje oprobane metode. Pitanje izgradnje računskog centra postaviće se na začelje dnevnog reda. Prvo bi usledila dugačka i žučna diskusija o rasipanju spajalica i drugog kancelarijskog materijala. Na kraju, kada su svi već umorni i ispražnjeni, ustaje šef računovodstva, koji sebe već vidi kao direktora računskog centra i predlaže investiciju od desetak milijardi. On vrlo opširno obrazlaže kako će taj svemogući računski centar „u sadašnjoj složenoj situaciji rešiti sve naše probleme do 2000-te godine". Pošto svi jedva čekaju da krenu kući, niko neće ni registrovati ovakve budalaste tvrdnje smotanog šefa računovodstva, koji ne zna ništa ni o svojim trenutnim problemima.

I tako računski centar 'ladno prolazi. Dok o regresu za godišnji odmor zna svako nešto da kaže, te milijarde su potpuno van spoznaje normalnog čoveka. I, uopšte, velike i male gluposti mnogo lakše prolaze nego srednje. Srednje gluposti može svako da vidi, dok ogromne i sićušne nije moguće sagledati bez posebnih pomagala. Na nesreću, one su i najopasnije: velike jer su velike, a male jer ih mnogo ima.

Pretpostavimo da je direktor male firme uspeo da završi računski centar, i da firma nije potonula. Računski centar neće povećati produktivnost, jer će se isključivo koristiti za administrativne poslove. Računar ne samo da neće smanjiti broj administrativnog osoblja, nego će ga čak i povećati. Na primer, bivši šef računovodstva, a sada direktor računskog centra, koji veze nema o računarima, moraće da zaposli minimalno dvojicu pomoćnika koji nešto znaju. Ako bi bio samo jedan, ovaj bi ga ubrzo izgurao iz fotelje.

Osim toga, razmnožavanje administracije obrnuto je srazmerno poslu koji treba obaviti. Računski centar, ma kako šugav i zastareo bio, daleko premašuje potrebe male firme. Da se računar ne bi dosađivao, baš kao i osoblje koje je vrlo brzo izgustiralo onih par idiotskih igara koje se na njemu mogu igrati, veštački se izmišlja posao. I svi su onda pretrpani poslom koji sami sebi prave, i nikako im nije jasno kako se nekada uopšte moglo raditi bez računskog centra.

Iz izloženog se vidi da je *potpuno ne prihvatljiva* svetska tendencija da se proizvode što manji, a što moćniji računari po niskim cenama. Mi bismo morali razvijati za naše uslove daleko primereniju koncepciju — Price Without Power (u slobodnom prevodu: um caruje snaga klade valja). Iako je na tom polju dosta učinjeno, očito je da se stalo na pola puta. Računski centar trebalo bi da preuzme funkciju koju je nekada imala crkva. Trebalo bi da narodu uliva strahopoštovanje, da bude što veći, sa prozirnim zidovima, da ljudi vide a da ne mogu da uđu. Osoblje bi obavezno moralo da nosi bele mantile kao sve viđenije profesije (lekari, arhitekte, nuklearni fizičari, mesari . . .). Oprema bi morala da bude futurističkog dizajna, sa što više svetlosnih efekata. Šta ima unutra nije osobito važno — bitno je da ne bude prazno.

Male računare bi trebalo zakonom zabraniti.

Jelena Rupnik

Razbarušeni sprajtovi

Usijani džojstik

Stiglo nam je nekoliko saveta za „Elitu" od izvesnog Tamina iz Pančeva: „Nakon skoka u hiper svemir potrebno je prići planeti, što često oduzima mnogo vremena. Da bi potrebno vreme bilo što kraće, nemojte „gledati" u planetu, već u neki od preostala tri pravca.

Ako želite da prvim laserskim pucnjem oštetite udaljeni neprijateljski brod, potrebno je da ubacite projektil, dovedete neprijateljski brod na mušicu vašeg lasera i u trenutku kada začujete zvučni signal, zapucate.

Na gotovo svakoj trećoj stanici možete zaraditi veliku količinu novca kupovinom i prodajom trećeg od nazad proizvoda čija se vrednost menja pritskom na taster 5 ili 6.

Toliko o Eliti. Marko Vukasović iz Zadra nam piše da je u igri „Commando" postigao skoro od 3075900 i da je „onaj ušljivi bunker" uništio 51 put. Smatra da je taj rezultat najbolji do sada postignut.

Ivan Stojković iz Niša javlja da je tako specijalizovao „Boulderdash II" da može čitavu igru da završi za 16 minuta (to mu je i rekord).

Cele iz Ljubljane nam kaže da je probao „Masterstroniksove" „budžetske" igre koje smo pominjali u prošlom broju i tvrdi da su sve osim „Spellbounda" izuzetno „ružne, glupe i dosadne". Dobro, onda preporučujemo „Spellbound".

Ismet Arslanović iz Sarajeva kaže da je od nekog druga čuo da postoji spravica koja omogućava da budete praktično besmrtni u igrama i da se zove Game Killer. Znate šta to znači. Kad se ona koristi, sprečava se registrovanje preklapanja (iterakcije) sprajtova, pa zato ne samo da ne možete biti ubijeni nego ne možete ni „ubiti". U nekim igrama, kao što je Elita, ovo ne pali. Spravica košta 15 funti.

Milan Pantelić iz Beograda se žali da je zbog našeg saveta nabavio igru „Three Weeks in Paradise" i da je sada zaglavio negde usred igre i ne može dalje. CCCccc.

Goran Kaperelić iz Beograda tvrdi da je igra „Flintstones" imbecilna i da mu je žao onih par hiljada koje je dao piratu za nju. Uz put tvrdi da je sagradio kuću za pola sata.

Za sam kraj evo jednog apela. Rašegorac Petar iz Novog Beograda moli da objavimo rešenje programa „Elite" za „Komodor" i završava pismo jednom izdvojenom rečenicom: VLASNICI COMMODORA VERUJU U DEJANA RISTANOVIĆA.

BESMRTNOST ZA „KOMODOR 64"

Po učitavanju igre treba uneti odgovarajući poke, i tek zatim startovati igru.

BANDITS	POKE 4759,169
GYRUSS	POKE 10399,234: POKE 10400,234
HYPER OLYMPIC	POKE 14041,173
EVERYONE'S WALLY	POKE 34461,157
DUMMY RUN	POKE 4306,165: POKE 4446,165
SPY HUNTER	POKE 15582,234: POKE 15583,234
QUASIMODO	POKE 13571,165
FRAK '64	POKE 22048,173
GROG'S REVENGE	POKE 23608,173
BOOTY	POKE 21003,165
BEACH HEAD	POKE 8567,173: POKE 28486,173 POKE 29768,173: POKE 34074,173
JET SET WILLY	POKE 14271,165
LAZY JONES	POKE 4251,173
ZAXXON	POKE 28400,173
BRUCE LEE	POKE 7462,165
TAPPER	POKE 15899,165
RAID OVER MOSCOW	POKE 4075,173: POKE 10057,173 POKE 21868,173: POKE 30950,173 (životi) POKE 23429,173: POKE 24314,173 (diskovi)
FLIP AND FLOP 2	POKE 19342,234: POKE 19343,234
BUMPING BUGGIES	POKE 10705,173
KUNG FU MASTER	POKE 38649,189
SKOOL DAZE	POKE 7553,165: POKE 7623,165 (lines)
COMMANDO	POKE 2409,173: POKE 4854,173
ARABIAN NIGHTS	Posle učitavanja prvog dela programa treba uneti: A=13190: POKE A, 169: POKE A+1,173: POKE A+2,141: POKE A+3,169: POKE A+4,89: POKE A+5,96
IMPOSSIBLE MISSION	Kada se učita loader, snimljen standardnom brzinom, treba otkucati: POKE 2305,141: POKE 2306,19: POKE 2307,103
BLAGGER	POKE 3560,8
FORT APOCALYPSE	POKE 36339,153
FROGGER	POKE 22341,173
HARD HAT MACH	POKE 16877,173
HUNCHBACK	POKE 9521,234: POKE 9522,234: POKE 9523,234
MOON BUGGY	POKE 24151,173
NEPTUNE'S DAUGHTERS	POKE 7870,255
POOYAN	POKE 20634,173
SNOKIE	POKE 33242,55

BESMRTNOST ZA „SPEKTRUM"

Impossible mission

Učitajte bejzik deo sa MERGE i dodajte:

15 POKE 45299,62: POKE 45300,53

startujte program sa RUN i nastavite učitavanje.

Dynamite dan

Učitajte bejzik loader sa MERGE i u liniji 10 dodajte ispred RANDOMIZE USR . . . POKE 51398,110:POKE 55755,0

Gyroscope

Premotajte traku iza uvodnog bejzika i naslovne slike i otkucajte sledeći program:

```
10 FOR N=23296 TO 23331
20 READ A:POKE N,A:NEXT N
30 RANDOMIZE USR 23296
40 DATA 49,240,93,221,33,0,94,17,0,
162,62,255,55,205,86,5,221,33,232,253,
17,24,2,62,255,55,205,86,5,175,50,162,
210,195,194,206
```

Program startujte sa RUN i nastavite učitavanje.

Commando

Postupak je isti kao i kod prethodnog programa, tj. ne treba učitavati bejzik loader i sliku.

```
10 FOR N=23296 TO 23322
20 READ A:POKE N,A:NEXT N
30 RANDOMIZE USR 23296
40 DATA 49,128,91,221,33,188,91,17,
128,162,62,85,55,205,86,5,49,0,98,
62,182,50,5,108,195,26,254
```

Gunfright

Traku premotajte iza bejzika i slike i otkucajte:

```
10 LOAD "" CODE:LOAD "" CODE
20 POKE 23457,201:RANDOMIZE USR 23424
30 POKE 41845,0:POKE 51049,233:POKE
49250,0:POKE 53888,201:PRINT USR
24064
```

Tau ceti

Preskočite bejzik i sliku na traci i otkucajte:

```
10 FOR N=20480 TO 20506
20 READ A:POKE N,A:NEXT N
30 PRINT USR 20480
40 DATA 49,31,64,221,33,0,91,17,0,
165,62,255,55,205,86,5,210,0,0,62,201,
50,120,194,195,0,91
```

B.C.'S quest for tires

Sve uraditi kao kod prethodnih igara, a program je:

```
10 FOR N=23296 TO 23319
20 READ A:POKE N,A:NEXT N
30 RANDOMIZE USR 23296
40 DATA 49,254,91,221,33,0,96,17,0,
160,62,255,55,205,86,5,62,182,50,86,
227,195,64,187
```

Hakerska priča

Sinopsis za vašu novu igru

Daš? Dam!

Cilj ove arkadne avanture je da uz pomoć 100 000 000 dolara, koje ste dobili od velike multinacionalne kompanije, omogućite sklapanje ugovora za izgradnju pet nuklearnih elektrana u našoj zemlji.

Na početku igre treba da izaberete zemlju u kojoj ćete izvršiti svoju humanističku misiju. Najbolje je da to bude neka zemlja koja već ima iskustva sa inostranim kreditima.

Nakon što ste to uradili, pred vama se pojavljuje ekran — plan grada na kome su obeležene sve značajnije ustanove. Najpametnije je da se odmah uputite u gradsku većnicu, gde se održava koktel kome prisustvuju svi viđeniji ljudi.

Na koktelu se igra poznata društvena igra — masne fote. Svetlo se neizmenično pali i gasi, a vi treba, dok je mrak, da umuvate neprimetno izvesnu količinu dolara u džepove raznih ministara. To nije lako, zato što se pored ministara oko vas muvaju i razni bezveznjaci, pa ih u mraku možete pobrkati. Kada se svetlo upali, ministar će promeniti boju; on će pocrveneti ili pozeleneti, zavisno od toga da li ste mu uvalili dovoljno love ili ne. U slučaju da je pozeleneo, moraćete ponovo da se jurite s njim po mraku, sve dok ne dobijete željenu nijansu. Dakle, na ovom nivou cilj je jednostavan, ako ne i vrlo jednostavan: treba da uvaljujete lovu ministrima sve dok ne pocrvene.

Ako su vam pocrveneli ministri za energetiku, finansije, zdravstvo, nauku i informisanje, kao i neki tipovi u uniformi, pred vama se pojavljuje sledeći ekran: banka u Švajcarskoj, gde ćete na svoj konto uložiti ostatak provizije i premiju koju ste dobili za uspešno obavljen posao. Time ste već stekli rejting „dangerous".

Sa novom provizijom odlazite opet u istu zemlju, tj. automatski se vraćate na prvi ekran na kome se nalazi plan grada. Možete ponovo da odete na koktel u gradsku većnicu, ali, u svakoj narednoj poseti teže ćete postići da vam ministri pocrvene.

Pretpostavimo da vam nije uspelo da stignete sve ministre. Ako ste, recimo, propustili da se izmirite sa ministrom za nauku i zdravstvo, u sledećem nivou moraćete da obiđete što više instituta, gde treba da skupite određen broj metra naučnih radova koji su razbacani po najskrovitijim mestima . . . Međutim, ne treba da uzimate bilo šta, jer ako greškom uzmete i one radove koji nisu pravilno orijentisani, moraćete da ih pojedete i tako im uništite svaki trag.

Najnezgodnije je ako vam izmakne ministar za informisanje; onda ćete morati da se rastrčite po redakcijama i da u svakoj kupite bar po jednog novinara. Takođe ćete morati da posetite biblioteku i tamo uništite sve tekstove o štetnosti radijacije, da spaljujete sve publikacije u kojima nema dovoljno optimizma prema izgradnji nuklearnih elektrana, itd.

Na poslednjem nivou, suočavate se sa javnošću putem televizijskih kamera. U TV studiju treba biti naročito spretan i brz, pa na vreme izmaći stolice svim protivnicima i preseći mikrofonske kablove onima koji bi mogli biti protiv.

Igra je završena kada na ovaj način uspete da jednoj zemlji uvalite 5 (pet) nuklearnih elektrana i izvučete živu glavu. Tada se na ekranu pojavljuje veliki natpis: „SVEGA IMA DA BUDE, SAMO NAS NEMA DA BUDE", i vi automatski stičete rejting „deadly".

Jelena Rupnik

Top lista

Spektrum
1. Movie
2. Winter Games
3. Spellbound
4. Barry McGuigan World Champions
5. Commando
6. Yie Ar Kung Fu
7. Rambo
8. Hypersports
9. Caves Of Doom
10. Gunfright

Komodor
1. Kung Fu Master
2. Yie Ar Kung Fu
3. Rock 'N' Wrestle
4. Eidolon
5. Mercenary
6. Kane
7. Desert Fox
8. Koronis Rift
10. Winter Games

Amstrad
1. Elite
2. Sky Fox
3. Yie Ar Kung Fu
4. Hypersports
5. They Sold a Million
6. Formula One Simulator
7. Caves Of Doom
8. Finders Keepers
9. Who Dares Wins 2
10. Spellbound

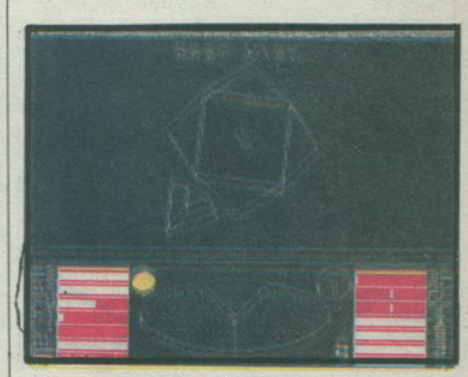
Igrajte „Elitu" bez varanja

PUCAJ, SINE!

Još jednom se (u istoriji) pokazalo da se od nepoštenja može sasvim lepo živeti. Ovde se to odnosi na moj status u igri „Elita", u kojoj sam tavorio sa dvestotinak kredita i malo opreme nekoliko prvih nedelja igranja. Pošto mi je dozlogrdilo da me napucava svaka šuša koja doleti iz mraka, odlučio sam da se privremeno odreknem časti i poštenja i na brzu ruku razbio celu flotu trgovačkih, piratskih i policijskih brodova. Švercovao sam narkotike, oružje i robove, jurilo me je desetak lovaca na ucene, ali su svi prošli isto (loše).

Sada mi je status, naravno, „Fugitive" i ne verujem da ću ga skoro izmeniti. Međutim, rejting mi je „Competent", a posada mi svaki čas više „Right on, Commander!", pa verujem da ću lepo napredovati u službi.

Još da napomenem da se nalazim u trećoj galaksiji, da sam do sada isprobao svu opremu koja se nudi na visokorazvije-

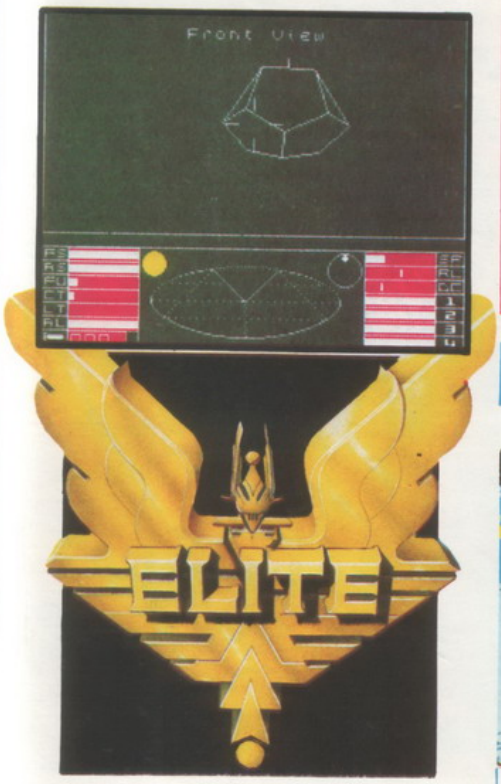


nim sistemima i, u vezi s tim, hoću da dam nekoliko saveta za igranje, koji su direktno proizišli iz dva meseca igranja.

1. Ne bacajte pare na „Military laser"! Prvo sam pojačao brod sa „Beam" (zračnim) laserima, ali nisam bio zadovoljan. Zatim sam kupio „Minning" (rudarski) laser, pa sam skupljao minerale i lepo napucavao okolo. Kada sam sakupio dovoljno love, nabavio sam taj vojni laser i utvrdio da guta mnogo energije,, tako da sam morao da kupim jaču energetsku jedinicu.
„Izuzetna probojna moć" vojnog lasera je važila samo za male brodove, ali se nijedan nije raspao bez bar tri pogotka. Još gore sam prošao kada sam upucavao velike „Python" trgovačke brodove, koji nisu hteli da puknu ni posle deset pogodaka! Ovo je bilo previše, jer sam potpuno isti efekat imao i sa „Beam" laserima, pa sam prodao vojni laser, kupio ponovo rudarski koji em što isti osao sređuje sa tri-četiri pogotka, em mi

omogućuje da skupljam minerale sa asteroida. Vrlo pozitivno na moral deluje kad vidite kako napadač (čitaj pirat), pogođen samo jedanput, iskače kapsulom, ostavljajući brod na milost i nemilost vama!

2. Kupite „Docking computer". Tako ćete pobeći od gomile pirata, kada ste već u sigurnosnoj zoni stanice, a oni vas napadnu u velikom broju, i uštedećete puno vremena, koje ste do sada gubili na sletanje. Bez kompjutera, u stanicu se ulazi dobro nacentriran na ulaz, bez rotiranja broda i pred sam ulaz sa maksimalnim gasom. Ovaj sistem retko kad omašuje, osim kada loše nacentrirate ulaz.
3. Nađite dva blisko smeštena različita sistema, recimo siromašan poljoprivredni



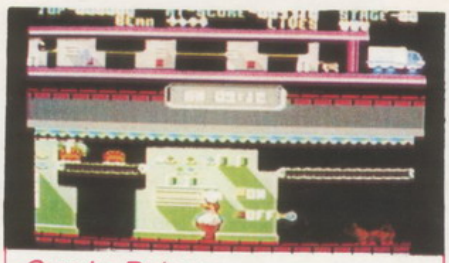
i bogat industrijski, i letite između njih sve dok ne sakupite dovoljno love za skok u novu galaksiju. Treća galaksija je vrlo interesantna.

4. Ako želite da igrate sa džojstikom, dovoljno je da pomerite palicu ulevo kada se na ekranu pojavi natpis: „PRESS SPACE, COMMANDER", Od sada će „ELITA" reagovati na svaki njegov pokret.
5. Snimajte svoj status u svakoj stanici. Da biste uštedeli trud, ne startujte traku, već snimite status „u prazno". Ako kasnije budete negde napucani, to će vam biti početno mesto za novu igru. Tek kada vam se prispava, snimite status stvarno na traku i isključite kompjuter.

P. S. Postao sam „Dangerous"!

Darko Stanojević

Spektrum



Comic Bakery

Smešna pekara

Ljudi iz redakcije „Računara" vole dobro da jedu. Svako ko je i jednom posetio redakciju zna da je to istina. Evo prave prilike da to i dokažemo. Ovo je prva igra koja se vrti oko jela još od prastarog Pacmena koje smo se dočepali. Igrali smo je jedno tri dana i uspeli da ispečemo gomile hleba (ali ne i da ga pojedemo). Priča je jednostavna. Vi ste pekar koga u procesu pečenja hleba ometaju rakuni (da, baš tako — rakuni) koji isključuju pećnice (da, baš tako — isključuju pećnice). Zato vi imate zračni pištolj koji smiruje rakune.

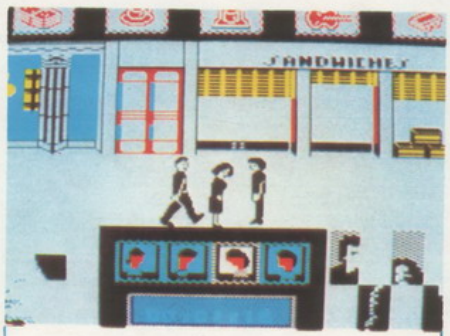
Grafika je ljupka, cilj je sladak, rakuni su simpatični. Igra malo dosadi kada više niste gladni, a nema ni slanih kifli. Ne izbegavajte ovu igru, ali je ne igrajte previše — osećate se kao da ste jeli previše testa — tromo.

The Eidolon

Eidolon

Sad se svi pitaju šta je to Eidolon. Eidolon je mašina koja omogućuje putovanje kroz vreme i maštu. Da se ne biste previše opustili u putovanju kroz vreme i vašu maštu, u hodnicima i pećinama kroz koje se putuje (da li te špilje predstavljaju maštu — mnogo je mračno) neprekidno nailazite na neverovatan izbor čudovišta. Čudovišta su najbolji deo ove igre. U njima ćete, sigurno, prepoznati pola svog komšiluka. Mnoga od čudovišta su pod uticajem jedne od sjajnih raznobojnih kugli koje se sreću u hodnicima. Crvene su opasne, ali vam mogu pomoći protiv čudovišta. Zlatne povećavaju energiju vaše mašine. Plave menjaju tok vremena, dok zelene t ansformišu jedno čudovište u drugo. Primetićete da se ne možete rešiti svih čudovišta na isti način. Svako od njih zahteva različiti tretman. Kao u pravom životu, jel te?

KOMODOR 64



Back to the future

Povratak u budućnost
Electric Dreams

Opet jedna igra rađena prema filmu. Ako ste ga gledali, biće vam mnogo lakše da shvatite o čemu se radi. Glavni junak, tj. vi, vraćate se u prošlost. Vaš najveći problem je u tome što je vaša buduća keva više zainteresovana za vas nego za vašeg budućeg ćaleta. Treba da razrešite tu edipovsku situaciju, inače vam se može desiti da uopšte ne postojite u budućnosti. Pored vaših dragih roditelja, tu se muvaju još neki tipovi. Od toga je jedan naročito neprijatan; stalno pokušava da vas nokautira, i tako vas ometa u nastojanju da organizujete susret svojih budućih roditelja, sve u nadi da će se oni zavoleti mladi. Koliko ste u tome uspešni, simbolično je predstavljeno fotografijama koje se nalaze u dnu ekrana. Prolazeći kroz razne prostorije, treba da pokupite pet različitih predmeta. Možete ih upotrebiti da vidite kakav efekat imaju na druge likove. Mada je grafika dosta dobro urađena, pametnije je da odete u bioskop i vidite kako to stvarno izgleda.

Scalextric

Leisure Genius

Još jedna igrica namenjena potencijalnim vozačima formule jedan. Pruža vam se mogućnost da birate između 17 svetskih „Gran pri" staza, a ukoliko ni jedna nije po vašem ukusu, možete i sami da dizajnirate stazu, a zatim je snimite na kasetofon ili disk. Igru mogu igrati dva igrača jedan protiv drugog. Menjanje brzina je automatsko, s tim što je maksimalna brzina 240 km/h. Na dnu ekrana možete videti svoj trenutni položaj na stazi, broj krugova koje treba preći, kao i brzinu kojom se krećete. Sva mudrost je otprilike u tome da ne ulećete pod punim gasom u krivine, jer vam se može desiti neprijatna stvar da izletite sa staze, što će vam drastično smanjiti brzinu, a vašem protivniku omogućiti da vas prestigne za tren oka. Grafički ugođaj je prilično dobar, mada nedostaju raznovrsni detalji pored staze. Prateći zvuci prilikom menjanja brzina, kao i škripanje guma u krivinama pomoći će vam da se maksimalno uživite u ulogu vozača formule jedan. Ako ste uživali igrajući Pitstop I i II, Pole Position, Revs i sl., pod hitno nabavite i ovu igru, da biste upotpunili svoju kolekciju.

• Ako „Računari" ne stižu do vašeg kioska . . .
• Ako ste nestrpljivi da svoj primerak dobijete što pre . . .
• Ako vodite računa o tome kako trošite novac . . .
• Ako želite besplatne usluge eprom-servisa . . .

PRETPLATITE SE NA ČASOPIS

računari

Pretplatom do kraja 1986. godine stičete nekoliko pogodnosti:

— uživate specijalni popust od 15%

— imate garantovanu cenu, bez obzira na poskupljenja

— ne možete ostati bez svog primerka

— omiljeni časopis dobijate na kuću

NARUDŽBENICA

Galaksija, Bulevar vojvode Mišića 17, 11000 Beograd

Želim da me pretplatite na časopis RAČUNARI od maja (broj 15) do kraja 1986. godine, (broj 21 po povlašćenoj ceni od 1.800 dinara.

DA NE (zaokružite odgovarajuću reč)

2. Molim vas da mi pouzećem pošaljite sledeće stare brojeve RAČUNARA: (1 i 2 su rasprodati) 3, 4, 5, 6 (za 200 dinara po primerku), 7, 8, 9, 10, 11 (za 250 D po primerku), 12 (po ceni od 300 D) — zaokružiti odgovarajuće brojeve.

Ime i prezime _____

Ulica i broj _____

Broj pošte i mesto _____

(Datum) (Potpis)

NAPOMENA: Ukoliko ne želite da isecanjem narudžbenice oštetite svoj primerak „Računara" molim da potrebne podatke ispišete na dopisnici ili u pismu i pošaljete na navedenu adresu.