

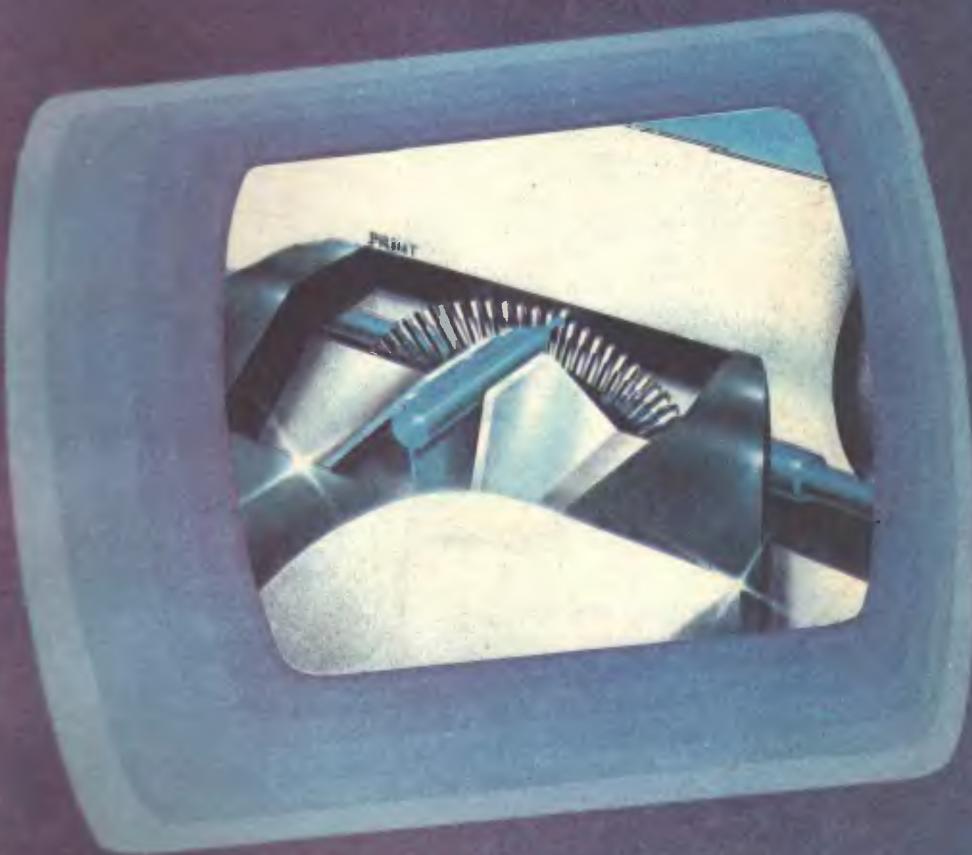
# Компютър

Издание на ЦК на ДКМС

# за вас

**5'86** Година Втора Цена 0,60 лв.

ISSN-0205 1893



СЛІСВА РЕДАКЦИЯ  
026/1893

*Скениране и обработка:*

*Антон Орущ*

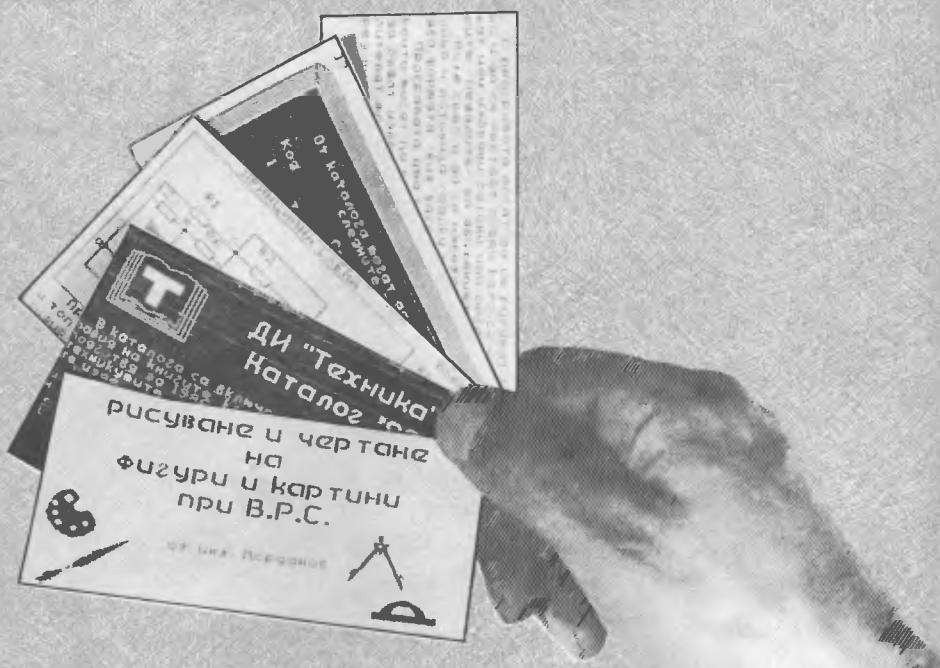
*[www.sandacite.net](http://www.sandacite.net)*

*deltichko@abv.bg*

*0896 625 803*



**ФОРУМ  
САНДАЦИТЕ**



# ко3

## НА ЧЕРТОЖНАТА МАСА

Графичен език за рисуване на картини, чертане на схеми и други изображения. Правец 82, 1 дискетно устройство. Системна дискета и ръководство за работа. Готов набор от елементи за изчертаване на електронни схеми. Цена 70 лева.

Разпространител: Сдружение за съвместна стопанска дейност "ППС - АВАНГАРД". София, "Бабуна планина" 4а, тел. 66-50-84.

## ДРАГИ ЧИТАТЕЛИ,

В писмата ви през април преобладаваха въпросите около новия домашен компютър Правец-8Д и това не ни изненада. Оформиха се и две основни групи писма — в едната ни питате кога, къде и как ще можете да купите компютъра, интересувате се от неговите характеристики и области на приложение. Другата група читатели, очевидно повече или по-малко запознати с компютърта, отправят никаки критични бележки и препоръки. Това, което обединява двете групи, е, че едните, и другите желаят да притежават Правец-8Д!

Най-често задаваният въпрос е: „Щом толкова го рекламирахте и хвалите, каквато сега когато най-сетне ще се появи на пазара и достатъчно количество?“ Върно е, че още минатала година във вестник „Направи сам“ публикувахме първите данни за облика на бъдещия домашен компютър, в „Компютър за вас“ 02. 86, поместихме подробен обзор на техническите характеристики и сфери на приложения на всички производители компютър. В тази статия се съдържат и отговорите на въпросите, които вълнуват мнозинството от читатели. Ценен пътеводител при овладяване на домашния компютър е аналитик на различната между Бейски-версията на Правец-82 и Правец-8Д, който можете да намерите в КВ 04. 86.

Но да не отбягваме въпроса: „Кога най-сетне?“. Без да ангажираме Комбината по микропроцесорна техника в Правец, ще кажем, че прави част на неговото ръководство, че временно спря продажбата на компютърта до окончателното отстраняване на някои дефекти в клавиатурата (производителите са от Габрово и Шумен). Така че при сегашните темпове за отпечатване на списанието, току-няж, компютрите да изпързват този брой.

Критичните бележки най-вече се свеждат до програмата несъвместимост между Правец-8Д и Правец-82, недостатъчната разделителна способност и най-вече, изтъкват читателите, без дисково устройство сериозна работа им може да се върши.

С това твърдение сме съгласни и име, но и никой не е очаквал това от Правец-8Д. Компютърът е от класа на домашните, т. е. домашния си и неговото предназначение, домашна е и цената му. Всъщност за цената си той притежава много добри технически показатели. И така домашният компютър е преди всичко машина за игри, за първоначално обучение и за решаване на сравнително ограничен кръг практическни задачи.

И все пак как би наглеждал Правец-8Д с дисково устройство? Знаем, че са правени успешни експерименти за свързването му към вносно 3,5-инчово дисково устройство. Но знаем също така, че цената на такова устройство надвишава стойността

на самия компютър. Ето защо, по наше мнение, едва ли си заслужава да се овладява специално производство у нас. Но за обучението в клубовете „Компютър“ комбинацията „Правец-8Д + дисково устройство за Правец-82“ предлага привлекателни възможности. Остава да се реши въпросът с интерфейса за свързване на двата прибора. Убедени сме, че този проблем е решим у нас, стига да се намери кой да подкрепи и да финансира разработката и производството.

И още една спрavedлив критика — компютър име, програми има? Така е, макар и без да разбираш защо. И производителите има компютъра, и много програми имащи са заинтересованы да произвеждат и да продават програмни продукти. Но до колкото знаем, готовност има само ИВСД „Азагард“, а в нашето списание от брой 6 започваме разговор до публикуване такива програми.

Сега да хвърлим бегъл поглед на бром, който ни предлагаме. В извежданиято на партийният конгрес в столицата бе открита поредната наложба „Изобретение в София“, която демонстрира постиженията на столичната научна и техническа мисъл, приноса ѝ в развитието на научно-технически прогрес и по-специално на неговите стратегически направления. Електрониката естествено заемаше централно място на наложбата и им сме се постаратели да им представим това участие.

Тук разширяваме присъствието на новите рубрики, чито основни положихме в първите три тагодишни броеве на списанието. Наред с поредиците „Пролог“, „Управляваща програма Монитор“, „Асемблер и машинен език“ не сме забравили и Бейски, като в две поредни статии ще разглеждаме по-подробно приложението на операторите РЕЕК, РОКЕ и CALL.

Заделими сме по-ново място за приложните програми и това не е случайно. Убедени сме, че след като държавата е илюстрирала огромни средства за масовото разпространение на компютърната техника, по-ефективно целият објект от града ще биде и нейното използване. Така че без да поддържаме значението на игрите (за съответната възрастова група), нуждата от приложни програми излаграме за по-актуална.

Случвало ли ви се е набраната на компютъра програма да тръгне от пръв път? Едва ли особено ако е по-дълга или съдържа по-обемиста върволица от цифри, включени в оператора DATA. Варваме, че оцените полезността на програмата „Контролна suma“ и намерението ни във фидеце към всяка програма да публикуваме и нейната контролна сума. Така лесно и бързо ще можете да локализирате и да отстраните допуснатите грешки.

Инж. ГЕОРГИ БАЛАНСКИ

Компютър за вас 1

За никого не е тайна, че по-нататъшното развитие на съвременното ни общество е немислимо без широкото използване на електронноизчислителната техника. Причините за това са повече от ясни – днес почти няма отрасъл на народното стопанство, където да не се използват компютри или микропроцесорни системи, независимо дали се прилагат за управлението на промишлени, или технологични процеси, в автоматизираните системи за управление, за решаването на технически, научни или икономически задачи. Всичко това доведе до бурно развитие на този съществен отрасъл.

Инж. ВАСИЛ ДИМИТРОВ

## ИЗОБРЕТЕНО В СОФИЯ '86

Също не е тайна, че икономиката на нашата страна създаде значителен потенциал за производство на електронноизчислителна техника. Възето партийно и правителствено решение в края на шестдесетте години за нейното развитие се оказа изключително далновидно и правилно прогнозирано. Днес нашата страна е един от най-големите производители на електронноизчислителна техника сред страните – членки на СИВ, а по производството ѝ на глава от населението е и на едно от първите места в света.

Електронизацията и най-вече компютризиацията на народното стопанство е здрав от основните фактори за съществуване на повсеместна интензификация, най-вече а онеzi производства, които имат решаващо значение за развитието на икономиката. Като доказателство на всичко, казано дотук, могат да се приемат редица експонати на проведената в началото на април изложба „Изобретено в София '86“. Да припомним само някои от тях.

### КОМПЮТРИ-МЕТАЛУРЗИ

Весьмънство става дума за програми на ноу-хау за изработка на лястови технологии на отливки от всички черни сплави и за топене на стомана в електро-дъгови пещи в режим на „съвет-

..Да се усъвършенствува структурата на електронноизчислителната техника, като се усвои производството на нови видове запомнящи устройства, на управляваща микропроцесорна техника и програмно-техническа система. Производството на електронноизчислителна техника да се увеличи около 2 пъти, а на микрокомпютърна та техника – над 2.5 пъти.

Ускорено да се развива програмната индустрия, която чрез прогресивни технологии да осигурява програмни продукти за автоматизация на технологичните процеси и на производството, на научните изследвания, проектно-конструктурската и управлческата дейност“

Из „Тезиси на XIII конгрес на БКП“

ник“ в реално време. На практика компютърът се превръща в технолог-леяр, работещ с максимална скорост, непознаващ трудности и нямащ проблеми с времето.

За контакти: инж. Валентин Иванов, тел. 65-29-69

### ПЕРСОНАЛЕН КОМПЮТЪР ИЗОТ 1036 С

Ератата на 8-битовите компютри отминава, за да отстъпи място на 16-битовите, които вече с право могат да носят прилагателното „профессионален“. При разработ-

ването на този модел са използвани четири изобретения, приложени в модулния принцип с възможности да се създават различни по състав и предназначение конфигурации, като основната се състои от процесорен блок, монитор и клавиатура. Този персонален професионален компютър е предназначен за обработка на научно-техническа и икономическа информация, за автоматизация на инженерния проектиране, на изследователската, учреджденската и развойната дейност. Освен това той с успех може да се използува при изграждането на локални мрежи като работна станция. От тази година започва серийният му производство в завод „Електроника“. ИЗОТ 1036 С е създаден от колектив с ръководител и. с. к. т. и. н.ж. Хр. Турлаков от Централния институт по изчислителна техника.

За контакти: инж. Ел. Русева, тел. 73-61/244

### ПРОГРАМИРУЕМ КОНТРОЛЕР „ИЗОМАТИК 1001“

Програмираме управляващи устройства (контролери) се използват за управление на различни технически обекти, изпълняващи операции с цикличен ха-

рантер, като роботи, агрегатни машини, подемно-транспортни и складови съоръжения, конвейери и прочие. Всъщност избройките сферите на приложение на конкретния контролер „ИЗОМАТИК 1001“, който е изпълнен на базата на 8-битов микропроцесор по така наречената концепция „черна кутия“. Чрез своите 40 оптоизолирани входа той може да взаимодействува с функционално свързани с управлявания обект устройства и сензори, което дава възможност за програмиране на оптимално поведение на обекта. Самото програмиране може да се извърши по два начини — предварително, без връзка на контролера с обекта, и при непосредствено изпълнение на операциите от обекта по време на програмирането, така наречените режими „оф лайн“ и „он лайн“.

Самото програмиране се разлиза чрез лесно усвоими команди с програматор, с който се записват програми върху пром. Едни програматор може да се използува за произволен брой контролери, които работят самостоятелно или в състава на сложни йерархични управляващи системи, включително и ГАПС. Разработката, която включва едно изобретение, е дело на колектив от ИТКР—БАН с ръководител ст. н. с. Г. Начев.

За контакти: инж. В. Пулев, тел. 71-401/384

### ПРЕНОСИМ ИНТЕЛIGЕНТЕН ВИДЕОТЕРМИНАЛ

Той осигурява връзката между микропроцесорните управляващи устройства и оператора както на ниво „букеенно-цифрова информация“, така и чрез графични изображения. Вграденият компютатор на Форт Х позволява лесно адаптиране на програмното осигуряване на терминала към конкретните нужди на потребителя. Предвидени са възможности за свързване на цифрови входове и изходи и опростенят интерфейс позволява на терминала да контактува със съответната периферия. Всичко това води до изграждането на управляващи устройства с малка и средна

сложност, като такива устройства могат да се свържат в мрежа до 32 броя на линии.

Клавиатурата е капацитивна и е във фолийно изпълнение, а мониторът е монохромен, като се разработват разширение за цветна графика, контролер за дискетни устройства и устройство за програмиране на епром от 16 до 256 Кбайта. Модулната конструкция позволява лесното интегриране на терминала като част на по-голяма система, а авторят му (в терминала е вложено едно изобретение) инж. В. Христозов от ИТКР — БАН, е подготвил виендрянският комбинат „Светлина“ — Сливен.

За контакти: инж. В. Пулев, тел. 71-401/384.

### МИКРОПРОЦЕСОРНА СИСТЕМА „ПЧЕЛА 05“

Информационната микропрограматорска система „Пчела 05“ е разработка на колектив с ръководител ст. н. с. к. т. н. Юл. Караваилов от ПНИЛ по РМСКОИП при ВМЕИ „Ленин“ и е създадена на базата на три изобретения.

Тя е предназначена за събиране на информация от аналогови или цифрови сензори, първичната ѝ обработка, временниот ѝ запомняне и предаването ѝ по линии за телекомуникации към централен компютър. Поради конструктивният си изпълнение (така наречените „евро“ модули) „Пчела 05“ е най-подходяща за вграждане към информационно-управляващи системи в промишлеността — химичната и хранително-вкусовата промишленост, енергетиката (АЕЦ, ТЕЦ и ВЕЦ) и прочие. Тя може да се използува и в научните изследвания, където се налага използването на точни, сигурни и бързи системи за измерване на аналогови сигнали от реалични промишлени сензори и възприемането на стандартизиирани цифрови сигнали.

„Пчела 05“ е изградена с елементна база от социалистически страни, като микропроцесорната ѝ част е изпълнена със българските интегрални схеми от серията СМ 600.

За контакти: П. Ангелов, тел. 63-61/29-33.

### МИНОГОПРОЦЕСОРНА ИЗЧИСЛИТЕЛНА СИСТЕМА

За да увеличат производителността на използваваната изчислителна техника, специалистите преминаха от едногипроцесорни към многогипроцесорни изчислителни системи с паралелни обработки на информацията. Такава е и разработката на главния консистент к. т. н. Христо Ангелов от ВМЕИ „Ленин“, като освен високата производителност притежава и още едно интересно качество — сравително ниска степен на сложност, което е от основно значение. Предлаганата система се характеризира с принципа нова организация на програмната и оперативната част, с нов и значително по-ефективен механизъм на взаимодействие на процесорите.

Сумирани, качествата на системата я издигат над известните системи от този клас (с общ па-мат).

За контакти: к. т. и. Хр. Ангелов, тел. 63-61/24-59.

### МИКРОКОМПЮТЪРНА СИСТЕМА ЗА ПРОФЕСИОНАЛНО ОРИЕНТИРАНЕ И ПОДБОР

Разработката на колектив от ВМЕИ „Ленин“ с ръководител н. с. к. т. н. Е. Панджов е създадена на базата на едно изобретение и претендира за приложение в социалната сфера — тя е предназначена за комплексно изследване на гама психологически параметри, изобщодими за целите на професионалното ориентиране и подбора при различните професии от операторен тип, чиято дейност се осъществява при затруднени условия и в опасни ситуации. Тя намира приложение в центровете за професионално ориентиране, в психологияческите лаборатории към Министерството на транспорта и други за изслед-



ване на поведението и реакциите на човека-оператор. За база на системата се използват персоналният микроКомпютър ИЗОТ 1031С и редица пултове, а програмирането е на Бейсик.

### РОБОТ С ПАРАЛЕЛНА ТОПОЛОГИЯ ТОРО

Роботиката е гребнокът на автоматизацията. Имайки предвид тази истини, колективът от ВМЕИ „Ленин“ с ръководител к. т. н. Др. Ненчев е разработил робот, който се характеризира със специална конструкция, изградена на базата на затворена пространствена кинематична верига. Това значително повишава възможностите за точно фиксиране на манипулатора на робота, висок кофициент на съотношението „полезен товар/собствено тегло“, както и поиздравява технологичност. ТОРО притежава 6 степени на свобода, осигурявачи му необходимата гъвкавост и универсалност при работа. Той може да се използува за роботизиране на широк кръг операции, предимно в радиоелектрониката и електротехническата промишленост — най-вече за извършването на сложни монтажни операции.

Освен това ТОРО е конструиран и като учебно-техническо средство за обучение по робототехника в средните и висшите учебни заведения. Той се управлява от персонален микроКомпютър Правец-82 и върху дисплея учащите се получават необходимия обем информация за текущото състояние на работата. За програмирането му се използува специално разработеният език ТОРОЛ, представляващ надстройка към Бейсик. Командите на ТОРОЛ позволяват управление „от точка в точка“. За системо програмни осигуряване се използува операционната система „Паралел“.

За контакти: Драгомир Ненчев, тел. 63-61/29-91.

Списъкът на изделия от високо ниво може да бъде продължен още, но и това кратко изброяване не е достатъчно красноречиво, за да се уверим, че столячините специалисти, а и не само те, уверено върят към подстъпите към следващия век.

Разработената в Института по техническа кибернетика и роботика при БАН от колектив с ръководител ст. н. с. к. н. Н. Илиев управляваща система МИК 2000 С с разпределена архитектура разширява концепцията за разпределение на функциите на управление върху нестационарни (преходни и фазови) непрекъснати процеси в различните промишлени производства. Системата е гъвка и с многострани възможности на приложение. Ето някои от тях:



# МИК 2000 С

- разпределено управление на технологични процеси с до 2000 входно-изходни точки към системата;
- централизирано наблюдение и водене на процеса чрез цветни мониторни станции;
- координация на всички функции чрез високоскоростна магистрала;
- удобно и бързо конфигуриране и параметриране в диалогов режим на контури, поддържащи технологичните параметри на процеса;
- проблемно ориентиран език за описание на динамиката на иестационарни процеси;
- ефективна система за обработка на аварийни състояния;
- стандартни показания на операторските мониторни станции, ориентирани към технологичния процес;
- пряк достъп на оператора до процесните величини чрез специализирана клавиатура;
- документиране на състоянието на процеса и действията на оператора;
- модулност на аппаратните средства;
- ефективна самодиагностика и прочие.

Едно от първите приложения на системата е внедряването ѝ в Завода за микробиални препарати — Пещера, СХК „Девня“ и „Нефтохим“ — Бургас, за управление на процеси в биохимичното производство.

За контакти: инж. В. Пулев, тел. 71-401/384

ОБРАЗОВАНИЕ

# КОМПЮТЪРЪТ ИЛИ ПЕДАГОГЪТ

Проблемите по използването на компютърната техника в образованието са в центъра на вниманието на нашата общественост. Във връзка с това нашата сътрудничка Виктория Белкова се срещна и разговаря с доцент к. и. н. ДИМИТЪР ПАВЛОВ от катедра „Теория и история на педагогиката“, ръководител на Проблемната научноизследователска лаборатория по компютърна техника в СУ „Климент Охридски“.

— Доцент Павлов, вие сте преподавател в СУ „Климент Охридски“ и като педагог същевременно се занимавате с проблема за използването на компютърната техника в образованието, ръководите студентска проблемна група и специализирана лаборатория по тези въпроси. Това са ли ваши лични интереси ли са?

— Научното си развитие получих под прякото ръководство на проф. д-р Сергей Иванович Архангелски — голям съветски учен с международен авторитет, който още края на 60-те и началото на 70-те години излагаше и обосноваваше своята педагогически концепция чрез математически модели, чрез теорията на системите, теорията на вероятностите, кибернетиката и т. н. Това, което усвоях от неговата педагогическа школа, от съветската педагогическа наука, се оказа решаващо за професионалната ми ориентация към компютърната техника.

Известно е също, че най-добри реформатори са тези, които засочват реформите от себе си. А педагогиката е един от науките, която съпътства развитието на човешката цивилизация от нейното зарождане до днес. Между науки, технически и социалния прогрес на човешкото общество и педагогическата наука действува законът за равновесното съответствие. Следователно интерес-

ът на всеки педагог към процеса на внедряване и използване на компютърната техника в образованието е закономерно явление, а не перфирно, индивидуално увлечение.

— Увеличават ли се привържениците на идеята за широкото нализане на компютъра в системата на образоването?

— Да. Много педагози вече работят активно върху тази проблематика. Например проф. д-р М. Андреев освен статиите, които публикува, е отдален самостоятелно място на компютъра в процеса на обучение във второто издание на своя фундаментален труд „Дидактика“. Интересни собствени педагогически изследвания, разработки и публикации в тази област имат още доц. Ив. Марев, ст. и. с. Р. Петров, гл. ас. Ж. Савова, гл. ас. Е. Василева и др. Към катедрата „Теория и история на педагогиката“ има аспиранти и специализанти, които разработват кандидатски дисертации и дипломни работи, създават и използват на компютърната техника в образованието, влиянието, което тя оказва върху учащите се, и т. н. Това са амбициозни млади хора, които имат по 6–8 публикации през последните две години, по 10–15 научни съобщения, лекции, доклади на различни научни форуми — конференции, семинари и др.

Що се отнася до студентската проблемна група, това действително е един интересен експеримент. Работата започна още когато нямахме нито един микрокомпютър. Благодарение на отзивчивостта на доц. Ст. Бъчваров, доц. Т. Боянов, и с. П. Павлов от Лабораторията по информатика и изчислителна техника при университета младите педагози намиха възможността да се запознаят с микрокомпютърът. И както при студентите от всички други специалности, един само задоволиха любопитството си, други научиха как могат да го използват в своята работа, трети отидоха по-нататък — започнаха самостоятелно да разработват програми, и по моя преценка съзряха своето професионално-педагогическо развитие с проблемите, предизвикани от компютърната техника в образоването.

— Доц. Павлов, как виждате основната цел на използването на компютрите в образоването?

— Основната цел на внедряването на микрокомпютъра от предучилищна възраст до втората степен на единното средно политехническо училище е многостранното развитие на личността, чрез което се създават разни възможности на асички деца за пълноценна реализация в живота. За много от тях микрокомпютърът ще бъде нещо подобно на телефона, магнитофон, радиопарата, телевизора, т. е. ще си служат с него, без да познават било конструкцията, било особеностите на работата му. Въпреки това всички трябва да се запознят с неговите принципи на действие, с



архитектурата му, с основните идни от теорията, програмирането и т. н., както всички деца са се научили да пишат, четят и смятат, независимо дали ще стават стругари, трактористи, актьори, лекари или писатели. Смисълът на тази подготовка не е да правим от всяко дете програмист, и то пък чрез нея да направим селекция на възможностите на децата, а преди всичко да изградим у бъдещите граждани нов стил на мислене, който е изобилен на човека в ерата на компютърната действителност.

— Вие твърдите, че компютърът ще появии върху начина на мислене на учащите се. А с какво според вас се отличава този нов стил на мислене?

— Преди всичко промените трябва да търсят в алгоритмизирането, моделирането, прогонзирането. Известно е, че децата са много любознателни. Те не само прилежно усвояват учебния материал в училище, но четат книги, списания, слушат радио, гледат филми, за да наслушат колкото е възможно повече знания, да си обяснят заобикалящата ги действителност. Една част от тях отиват по-нататък. Те търсят отговор на въпроса „Как?“. Тук вече е необходим алгоритъм, алгоритмично мислене. Наблюдените показват, че деца, които са работили с микрокомпютър и са способни да създават макар и елементарни програми, с лекота алгоритмизират всяка своя дейност. За тях решаващето на задачата по математика например е свързано преди всичко с построяването на логически върхи на последователност от операции, а не толкова на конкретна изчислителна работа.

За да може да намери адекватен отговор на въпроса „Задача?“, на детето трябва анализ и моделиране. Най-по-добро средство от микрокомпютъра в тази насока. Чрез него може бързо и лесно да се построи модел и кое то е още по-важно, да се провежди доколко той е правилен, от какви фактори зависи и т. н. По този път не е трудно да се правят предположения: „ако се увеличи двойно скоростта . . .“, „концентрацията . . .“, „налягането или гъстотата . . .“ и т. н., какво ще се получи, как ще се промени обектът, процесът, явление то.

Има много основания да се приеме, че дете, което още в училище се запознае с подобен начин на мислене, през целия си живот, независимо каква професия ще упражнява, винаги ще търси причинно-следствения характер на нещата и явленията, ще си служи с един или други хипотези и прогнози. Едва ли е необходимо да се доказва, че така формираната и изградена личност ще бъде полезна както на себе си, така и на обществото.

— Кои са според вас основните проблеми, свързани с внедряването и използването на компютърната техника в образованието?

— Най-кратко те могат да бъдат изразени чрез дад проблемни триъгълника. Първият обхваща въпросите „Къде?“, „С каква цел?“ и „С какви задачи?“. Вторият конкретизира „кога?“, „С какви функции?“ и „Как?“. От една страна, ако внимателно се анализират трудностите, свързани с използването на компютърната техника в образоването, веднага се откровява силуетът на единния или другия проблемен триъгълник. От друга страна, не е необходимо човек да бъде специалист, за да разбере, че именно това обикновено липсва на лавината от публикации, радио- и телевизионни представления, конференции, семинари и т. н., с които заливаме нашата общественост. По своята дълбочина същност тези въпроси са специфично педагогически. Решението им не могат да се намерят вън и независимо от педагогическата наука.

Процесът на обучение и възпитаване на младежта е коренно различен от процесите на съвременното материално производство, в сферата на услугите и т. н. Въпреки че има обмен и управление на различни видове информации, но те проптичат не толкова по законите на информатиката, колкото според изискванията на педагогиката.

Появата и използването на машината освободиха човека от тежки физически труд. Сегашният етап от научно-техническата революция е насочен към освобождаването му от стереотипно повторящи се операции, от механичния сбор и обработка

на данни, от контрола на стандартно противщи процеси и явления, от стандартни решения. Очевидно всяка аналогия с тези процеси е неправдоподобна. Учителят е изземан. Неговият труд по своята дълбочина същност е творчески. Той всяка година, всеки ден и час е различен. Децата, с които работим, са неповторими индивидуалности. Тук няма място за стереотипи, каквито често ни предлагат.

Като рожба на тези проблеми в областта на образоването през юли 1985 г. бе създадена и Проблемната научноизследователска лаборатория за изследване на педагогическите проблеми, свързани с внедряването и използването на компютърната техника в системата на образоването. Тя е съвместно научно звено на Софийския университет и Столичния съвет на народна просвета. След преодоляването на разнообразните трудности, съпътстващи създаването на десеспособен научен колектив, лабораторията системно и последователно ще изследва както тези, така и много други актуални проблеми. Веднага следва да добавя, че би било наивно да се мисли, че в тази област са възможни лесни и бързи решения.

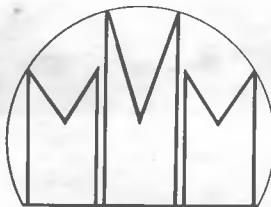
Ако за експеримент в хранителната промишленост са достатъчни от няколко часа до няколко дни, а за парфюмерийната до четири години, при педагогически експерименти са изобилни продължителни периоди. Това очевидно, че колкото по-рано започнем, толкова по-бързо ще можем да откриваме нови коризонти за развитие на педагогическата наука в тази област. И ако се вървя на сътенцията, че „няма нищо по-практично от една добра теория“, лесно може да се проследи влиянието на тази дейност и върху внедряването и използването на компютърната техника в системата на образоването.

Това въстъпство е пряко свързано с педагогиката като цяло и на всеки педагог поотделно в решаването на глобалните задачи, предизвикани от развитието на научно-техническата революция и пределно ясно формулирани на Февруарския пленум на ЦК на ВКП.

# ЗАПОЗНАЙТЕ СЕ

Вече близо едно десетилетие специалисти от различни области на научното познание — математици, философи, социолози, психолози, юристи, физици, икономисти, лингвисти, медици, кибернетици, изследват проблемите, свързани с трите „М“ — Методологията на Математическото Моделиране на природни и социални системи.

Проведените под този знак три години международни форуми а Международния дом на учениците (МДУ) „Ф. Ж. Кюри“ и Националният научно-практическа



## ПОД ЗНАКА НА ТРИТЕ „М“

### МЕТОДОЛОГИЯТА НА МАТЕМАТИ- ЧЕСКОТО МОДЕЛИРАНЕ

конференция в Благоевград популяризира проявите, организирани от междуусъдисциплинарна научноизследователска група по методология на математическото моделиране на природни и социални системи към Института по философия на БАН. Семинарите, еднодневните конференции, кръглите маси, националните конференции и международните симпозиуми дават възможност на учени с различни специалности и от различни институти, градове и страни да създават и да намерят нова научна общност, в чито рамки да задоволяват интереси със проблемите на математическото моделиране. Срещите, докладите, дори острите понякога спорове укрепват тая неофициална, но вече с традиционно наложило се постоянно ядро научна общност — резултат, не по-малко важен от съвместно постигнатите научни резултати.

Началото е през 1976 г., когато към Института по философия е създадена проблемна научноизследователска група по методология на математическото моделиране на природни и социални системи, известна по-късно у нас и в чужбина с краткото и лесно запомнящо се име — МММ.

През 1977 г. под ръководство на ст. н. с. Борис Чендов започва специализиран курс по математическо моделиране към Института по философия на БАН.

в курорта Дружба, за да събират на това място всяка четна година през първата половина на юни учени с интереси към методологичните или научно-практическите проблеми на математическото моделиране. В работата на конференцията участват с доклади учени от България, ГДР и СССР.

След конференцията групата активизира своята дейност, придобива популярност и признание и започва освен редовните семинари да провежда периодично еднодневни конференции, на

които се изнасят и обсъждат доклади, посветени на проблемите на математическото моделиране в определена научна област — физика, икономика, право и т. н.

През пролетта на 1982 г. в Благоевград се провежда Национална научно-практическа конференция по МММ, а през юни с. г. в МДУ — Първи международен симпозиум по МММ, който продължава делото на Тристранната конференция от 1980 г., но с много по-широко международно участие. Участват учени от почти всички страни — членки на СИВ. Разширен е не само кръгът на участниците, но и броят на секциите и сесиите.

Особено актнана е групата през 1983 година. Урежда еднодневни и двудневни конференции в София, някои от които с участието на чуждестранни учени, кръгла маса по МММ във Варна, по време на проведената в София конференция по вземане на решения (DECISION MAKING) бяха установени връзки с Международния институт по системен анализ (NASA).

Най-акушителната изява на групата досега е Вторият международен симпозиум по МММ през юни 1984 година. Участват 130 души, представители от всички страни от СИВ, Югославия и Финландия.



# ТЕОРИЯ

От Втория симпозиум досега групата продължава работата си в обичайните форми на едно-дневни конференции и семинари.

В тях като докладчици, гости и членове на консултативните съвети участват нашите академици Сава Гановски, Любомир Ильин, Благовест Сендов, Георги Бранков, Христо Христов, Азарий Попликаров и видни представители на съветската наука — академик Николски, чл.-кор. Г. Д. Постолов, д. ф. м. н. Чумаченко.

На групата активно съдействуват ръководството на Института по философия и на Единния център по философия и социология към БАН. Групата поддържа контакти и активно сътрудничи с много наши и чужди институти и организации, участвува в напълнението на задачи по теми на международното сътрудничество между академиите на науките на страните — членки на СИВ.

## КАПИТАНЕ, ДА ВДИГНЕМ ЛИ

# ФЛАГА?

Доц. ДИМИТЪР П. ШИШКОВ

В програмирането често се използва военното понятие **флаг** със съответните изрази „**вдигни флага**“ и „**svали флага**“. То отразява често срещаната се в реалния живот нужда да се посочи единото от две възможни състояния: има — има, плюс — минус, точка — тире (в азбуката на Море, в частност в морското дело, визуално се изобразяват с флагове), светла — угласва, иключено — иключено и т. и. Програмистите използват това понятие за разклоняване на изчислителния процес чрез избор на една от две възможности. В компютрите флагът се предства (моделира) чрез един двоичен разред (често вместо разред се казва бит, но с бит се означават три различни понятия: двоичен разред, единица мярка за количество информация от СИ и единица мярка за обем памет), който има две възможни състояния, условно наречени **нула** и **единица**. Когато флагът „сочи“ единица, се казва, че е **вдигнат**, а в противния случай — че е **svален**. Флаговете се делят на апаратни (флагове за состоянието на процесора, ФСП) и програмни (ПФ) в зависимост от местоположението им в първата им със системата от инструкции (команди) на компютъра.

### Флагове за състоянието на процесора

Тези **едноразредни индикатори** отразяват съкупност от състояния на елементите на централния процесор, а в иякои случаи — и на външни входни или изходни устройства. Различните компютри имат различни по брой и вид ФСП — по-рано имаше само по един-два флага (от икономически съображения), а напоследък — по няколко. Най-често срещаните ФСП са:

- **пренос** (С — сагту). С=1, ако е имало пренос от старшият бит на суматора след събиране в допълнителен код или **заем** — в случаи на изваждане на числа без знак при сравнение на по-малко с по-голямо число. При изместване или ротация наляво или надясно в него се изпраща съответно старшият или младшият бит на съдържанието на известваната клетка или дума. Използва се за добавяне на преноса от една дума към следващата при **моделиране на аритметика** с по-голяма от присъщата на компютъра точност, а също и за добавяне на 1 при получаване на допълнителен код на отрицателни

числа в такава моделирана аритметика.

• **нула** (Z — zero). Z=1, ако полученият в суматора резултат е пълна нула. Използува се за управление на цикли, в частност при получаване на контролна сума нула, също при търсене на данни с определена стойност — ако сравним една данна с еталон чрез логическата операция **изключващо или** и данната е равна на еталона, но резултатът е нула. Този флаг служи и при проверка на стойността на кода на е разред в оперативната памет (ОП).

• **отрицателен** (или отрицателно число, N — negative, в МП 6502 се нарича **знак**, S — sign). Флагът получава знака на числото — резултат от последната операция в суматора — 0, ако е положително, и 1, ако е отрицателно.

• **препълване** (V — overflow). V=1, ако е имало препълване. Такова може да се получи само при събиране на числа с еднакъв знак или при изваждане на числа с различни знаци. **Препълване има, ако събираемите са били положителни** (със старши битове 0), а **результатът е отрица-**

```

0   REM "ИВАН АНГЕЛОВ НИТЕ
  В" #БЫЛ ГЕОРГИ ДИН
  ИТРОВ-35" А" #СОФИЯ
  -1000"

10  FOR T = 1 TO 10
    : HOME
    : VTAB 12
    : PRINT TAB( 20 - T)
    :   TAB( 20 - T) "K"
    :   TAB( 20 + T) "P"
    : VTAB 12 - T
    : PRINT TAB( 20 ) "T"
    : VTAB 12 + T
    : PRINT TAB( 20 ) "N"
    : FOR P = 0 TO 5
    :   PEEK 49200
    : NEXT P, T
    : RUN

20  FOR T = 1 TO 10
    : HOME
    : VTAB 12
    : PRINT TAB( 10 + T)
    :   TAB( 10 + T) "K"
    :   TAB( 30 - T) "P"
    : VTAB 2 + T
    : PRINT TAB( 20 ) "T"
    : VTAB 22 - T
    : PRINT TAB( 20 ) "N"
    : FOR P = 0 TO 5
    :   Z = PEEK (49200)
    :   NEXT P, T
    : RUN

```

**СУПЕРМЪНІК**  
**БОНКУРС**  
**10-20\***

тлен (старши бит 1) или обратното — събирамите са били отрицателни, а резултатът е положителен. Флагът V на извиканите от флага C — иакои се заблуждават, като смятат, че при C=1 е имало превърнате, забравяйки, че се работи в допълнителен код.

• (забрана за прекъсване (I — interrupt disable). След получаване на прекъсване следващото прекъсване се забранява (Маскира), докато не се обработи текущото, след което отново се разрешават всични прекъсвания. В един МП I=1 е забрана за прекъсване, а I=0 — разрешение. В други МП е обратното. Чрез команда SET в българския микропроцесор CM-601, съвместим с Motorola 6800, както и в МП 6502 I=1, а чрез команда CLI I=0. Да обележим, че има и немаскируеми прекъсвания, каквато например може да бъде прекъсване при под на напрежението. Те настъпват независимо от стойността на I. При прекъсване начинът включване (RESET) централният процесор автоматично забранява прекъсванията. Програмистът може да забранява всични прекъсвания по време на критични линейни участъци или цикли. Ако компютърът има няколко входа или изхода на прекъсване, той може да има няколко флага за прекъсване.

• полулененос (H — half-carry). Н=1, ако при двоично събиране е имало пренос в единобайтов суматор от младшата половина (кварт) — младшите 4 бита, към старшата. Служи главно за нуждите на десетичната аритметика. В МП M6800 след двоично събиране на двуцифри числа в двоично-десетични (2—10) код е необходима десетична корекция чрез команда DAA, за да се получи правилен резултат в 2—10-код. В МП 6502 има режим десетична аритметика (флаг D=1), при който командите за събиране и изваждане правдилно обработват числата с 2—10-код. Ето защо в 6502 няма флаг H в PSW (вж. по-долу), макар че няякои флаг има, но той се използва в рамките на една аритметична инструкция и не е изведен в PSW.

• (режим) десетична аритметика (D — decimal mode). В МП 6502 това е режим, при който командите за двоично събиране и изваждане стават команди за десетична аритметика над числа с 2—10-код.

• четност (P — parity). P=1, ако броят на единиците в резултат от последната операция е четно (при контрол по четност) или нечетно число (при контрол

по нечетност). В съветския МП К580, съвместим с Intel 8080 и Z80, P=1 при четен брой единици в суматора.

Обикновено в МП присъстват C, N, Z и V, които се изменят от аритметичните команди, известните, ротациите и сравненията, а Z и N се изменят и от логическите операции.

Къде се намират ФСП? Те са разположени на фиксирано, определено от архитектурата на компютъра място, най-често обединени в: а) регистър на кода на усилвателя (CCR или PY в МП CM-601); б) регистър за състоянието на процесора (P в МП 6502, в който е поместена думата за състоянието на процесора PSW — processor status word); в) дума за състоянието на програмата (PSW — program status word, при EC и IBM, в думата флаговете заемат малка част, има и флагове за няколко съвместими двойки от противоположни режими). В редки случаи флагът е независим. Например в МП K580 няма флаг за забрана на външни прекъсвания, но чрез специални команди програмистът може да забрани или да разреши такива прекъсвания. Независим флаг обикновено се моделира чрез тригер.

Кой и как взаимодействува върху ФСП? Това става чрез инструкциите на компютъра, като взаимодействие е специфично: а) случайно — при извършване на операция над някакви операнди се получава резултат с определени качества, някои от които се инцирират в един или няколко ФСП. Например, ако събъртем 10000000(2)=—128 със себе си, ще получим резултат 0, като C=Z=V=1, а N=0; б) принудително — обикновено за иакои ФСП съществуват инструкции, с които се присвоява стойност 0 или 1 на съответния флаг. Казва се още нулиране на флаг, установяване на флаг нула или на флаг единица. Например в МП 6502 инструкциите CLC, CLD, CLI и CLV нулират съответно флаговете C, D, I и V, а SEC, SED и SEI — „вдигат“ флаговете C, D и I; в) независимо принудително — чрез избор на подходящи операции и операнди програмистът може да зададе програмно получаване на резултат, който да предизвика желана стойност на даден или дадени ФСП (ако има команди за задаване на стойности на съответните флагове); г) след пренос в ОП на регистъра или думата, съдържащи ФСП, програмистът може да оперира с получените стойности на ФСП, да ги изменя и да ги върне обратно. Това се използва рядко.

Кой и как използва ФСП?

а) програмистът ги използва за разпознаване на изчислителния процес чрез инструкциите за условно предаване на управлението (УПУ); б) иакои ФСП се използват от други инструкции например Н се използува от DAD, а D — от инструкциите за събиране и изваждане; в) управляващото устройство на процесора също използва иакои ФСП. Например системите за прекъсване следят флага I.

Да отбележим, че изпълнението на повечето инструкции на компютъра влиза върху поне един ФСП. Следователно флагът трябва да се използува от съответната инструкция, защото още следващата може да го измени. Ако това не може да стане веднага, стойността на флага трябва да се запази — обикновено в ОП.

#### Програмни флагове

Естествено е програмистът сам да може да създава флагове за нуждите на програмата си, като задава, изменя и използва стойностите им на определени места в програмния текст. Докато ФСП са няколко, много, често обединени в дълги поредици от битове — битови низове. Чрез тях могат да се разложват логически сложни процеси.

Къде се намират ПФ? а) в общи регистри; б) в разредите на ОП; в) при нужда — и във външната памет.

Кой и как взаимодействува върху ПФ? В съвременните компютри, в които има адресация на битовете в ОП, програмистът може по стойността на разреда (бит), съответен на даден флаг, да разклони изчислителния процес с подходящи инструкции за УПУ. При липса на битова адресация в ОП с подходяща маска (константа с единици в съответните на флага бит и нули в останалите битове) и логическо умножение може „да се извлече“ стойността на бита, т. е. флага, и да се съди за това, каква е тя чрез ФСП Z. С други подходящи маски и константи и с операциите логическо умножение и логически събиране програмистът може да зададе съответно стойност 0 или 1 на всеки бит, а значи и на всеки ПФ в ОП. А с посочената константа и операциите изключвачи (или събиране без пренос, събиране по модул две) стойността на всеки бит може да се сменя с противоположната ѝ. Така може да се получи флаг от типа магнита единица, при който 0 и 1 се сменят строго алтернативно. По описание начин може да се създаде и управлява всеки ПФ.

# САМОУЧИТЕЛ

Най-простите въпроси, които можем да зададем на системата, са за истинността на определен факт. Нека сме въвели посочената по-горе база от знания. Можем да попитаме системата: „Варно ли е, че ЛИЛИ СЛУЖИ в BT?“. Ето как се прави това:

? — СЛУЖИ (ЛИЛИ, BT).  
ДА  
?

Системата отговаря с ДА (потвърждава, че това е вярно) и отново продължава режим „въпрос“ (? —).

Ако поставим въпроса:

? — СЛУЖИ (БИСТРА, ЦИКС).  
Системата ще ни отговори с НЕ.

Внимание — това не означава, че БИСТРА не служи в ЦИКС. Истината е, че в базата от знания на компютъра не е регистриран подобен факт. Очевидно отговорът на въпроса се формира чрез претърсване на базата от знания, за да се установи присъствието на факта, формулиран във въпроса. Ако не се намери търсеният факт, отговорът е НЕ.

Обърнете внимание, че при внасяне на ново знание и при формулиране на въпрос използваме една и съща форма

СЛУЖИ (ЛИЛИ, BT).

но предхождана от различни комбинации от символи:

? — за въпрос  
> за внасяне на знания.

Можем да питаме не само за присъствие на определен факт в базата, но и за по-сложни ситуации. Можем да поставим въпроса: „Как служи в ЦИИТ?“ Подобни въпроси се поставят чрез формулиране на интересувация ни факт и поставяне на променлива вместо името на този, за когото питаме:

? — СЛУЖИ (S, ЦИИТ).

Отговорът на компютъра ще бъде: „ИВАН отговаря на въпроса“, и ще бъде представен под формата:

S=ИВАН RETURN  
?

## ПЪРВИ СТЪПКИ В ЛОГИЧЕСКОТО ПРОГРАМИРАНЕ

КЪНЧО ИВАНОВ  
кандидат на математическите  
науки

## ПРОГРАМИРАНЕ

# НА ПРОЛОГ

(продължава от KB. 02. 86.)

Когато отговорът е не просто ДА или НЕ, а изразява стойност на променлива, системата изчаква реакцията на потребителя, поставяйки мигащия курсор след отговора:

S=ИВАН █

Най-простата реакция за продължаване на диалога е да се натисне клавиша RETURN, с което системата преминава в режим въпрос.

Можем да проведем дналога по друг начин, като след отговора на компютъра:

S=ИВАН

го попитаме има ли още служител на ЦИИТ. За целта вместо RETURN след отговора натискаме „;“. Ето как ще протече дналогът в този случай:

? — СЛУЖИ (S, ЦИИТ).

S=ИВАН; (има ли още)  
НЕ  
?

Компютърът отговаря, че ИВАН СЛУЖИ в ЦИИТ, и на нашия допълнителен въпрос (;) той отговаря, че няма други факти, в които да е отразено, че някой служи в ЦИИТ (НЕ).

И така на въпроса:

? — СЛУЖИ (S, ЦИИТ).

системата отговаря, като претърсва базата и търси факти, в които СЛУЖИ и ЦИИТ съвпадат, като на променливата S се присвоява стойността, чиято позиция заема S (S=ИВАН). Стойността на S е отговор на въпроса. Когато след отговора S=ИВАН се въведе „;“, всъщност се иска от системата да продължи търсенето в базата за още факти, в които СЛУЖИ и ЦИИТ съвпадат с пръвната и третата стойност. Ето защо системата отговаря с НЕ — няма друг такъв факт.

Ако продължим дналога с:

? — УЧИ.  
>СЛУЖИ (ВАНИ, ЦИИТ).  
>СЛУЖИ (РУМЕН, ЦИИТ).  
>█.  
?

в базата ще бъдат добавени нови факти. Ако сега поставим отново въпроса:

? — СЛУЖИ (S, ЦИИТ).

диалогът ще протече, както следва:

S=ИВАН;  
S=ВАНИ;  
S=РУМЕН;

НЕ  
? —

Не забравяйте, че потребителят избира дали да прехъсне диалога с RETURN, или да го продължи с „;“ до пълно изчерпване на всички факти, удовлетворяващи въпроса.

По-сложна форма на въпрос можем да създадем чрез поставяне на променливи и за името, и за службата. Например:

? — СЛУЖИ (F, Q).

F=ИВАН

Q= ЦИНТ

F=ЛИЛИ

Q=BT;

F=ПЕГЪР

Q=КОРПОРАЦИЯ-ИНС RETURN

? —

Очевидно въпросът е: „Кой къде служи?“

Обърнете внимание на лекотата, с която се постигат съответните отговори. На програмиста на Бейсик предлагаме да помисли колко усилия би му струвало постигането на същия ефект с помощта на Бейсик. Но да продължим с внасянето на още знания за нашите познати. Например за месторождението:

? — УЧИ.

> РОДЕН (ИВАН, ВАРНА).

> РОДЕН (ЛИЛИ, ВАРНА).

Внимание — не бива да пишем:  
РОДЕНА (ЛИЛИ, ВАРНА).

макар че това е по-правилно от естествено езикови съображения. Принципът е, че системата ще третира като разнотипни релации РОДЕН и РОДЕНА.

> РОДЕН (РУМЕН, СОФИЯ).  
> РОДЕН (ВАНИЯ, ВАРНА).

Можем да поставим въпроси за месторождение:

> @.

? — РОДЕН (ИВАН, L).

L=VARNA RETURN

? —

При наличната база можем да задаваме и по-сложни въпроси. Например: „Къде е роден и къде служи РУМЕН?“

? — РОДЕН (РУМЕН, F), СЛУЖИ (РУМЕН, W).

F=СОФИЯ

W=ЦИНТ RETURN

? —

Употребихме предикатите РОДЕН и СЛУЖИ, обединявайки ги чрез „;“ в общ въпрос. Интересуваме се от съставен факт (композира от два отделни факта), в който РУМЕН е едновременно участващото лице. Променливите F и W получават стойности, удовлетворяващи съответните предикати. Лесно се вижда, че в напътната база искам от лицата са земляци. Иде лесно бихме могли да допълним базата, като добавим нови предикати:

? — УЧИ.

> ЗЕМЛЯЦИ (ИВАН, ЛИЛИ).

> ЗЕМЛЯЦИ (ИВАН, ВАНИЯ).

> ЗЕМЛЯЦИ (ЛИЛИ, ВАНИЯ).

Но това не е необходимо. Вместо това можем да въведем в базата „дъфиниция“ на ЗЕМЛЯЦИ.

Иначе казано, можем да въведем класуза, която определя предиката ЗЕМЛЯЦИ. Ето как става това:

? — УЧИ.

> ЗЕМЛЯЦИ (X, Y): — РОДЕН (X, Z), РОДЕН (Y, Z).

Записването трабва да се разбира по следния начин: ЗЕМЛЯЦИ са двама души X и Y, когато мястото им на раждане е едно и също Z. Сега можем да поставим въпрос:

> @.

? — ЗЕМЛЯЦИ (ИВАН, ЛИЛИ).

ДА

? — ЗЕМЛЯЦИ (ИВАН, РУМЕН).

НЕ

? — ЗЕМЛЯЦИ (ВАНИЯ, V).

V=ИВАН;  
V=ЛИЛИ;  
НЕ  
? — ЗЕМЛЯЦИ (Х, Y).

X=ИВАН

Y=ЛИЛИ;

X=ИВАН

Y=ВАНИЯ;

X=ЛИЛИ

Y=ВАНИЯ;

X=ВАНИЯ

Y=ИВАН;

X=ВАНИЯ

Y=ЛИЛИ;

НЕ

? —

Обърнете внимание, че землячество на всеки двама земляци се изразява по два начина:

X=ЛИЛИ

Y=ВАНИЯ

■

X=ВАНИЯ

Y=ЛИЛИ

Това е поради факта, че удовлетворяването на въпроса ЗЕМЛЯЦИ (Х, Y) съгласно класузата ЗЕМЛЯЦИ се постига със сметка на сложните факти:

РОДЕН (ЛИЛИ, ВАРНА), РОДЕН (ВАНИЯ, ВАРИА).

■

РОДЕН (ВАНИЯ, ВАРНА), РОДЕН (ЛИЛИ, ВАРНА).

Нали е лесно и приятно да се програмира на Пролог? Имайте предвид, че само поединци хме завесата. Големите възможности на езика ще се разкрият в следващите статии.

СЛЕДВА

# НЕПОСРЕДСТВЕН ДОСТЪП ДО ПАМЕТТА

Вътрешната памет на компютъра се състои от 65536 клетки, разделени в три основни групи:

- оперативна памет с произволен достъп (рам) — 48 Кбайта (с разширение до 64 Кбайта при включена платка от 16 Кбайта);

- постоянна памет (ром) — 12 Кбайта;
- входно-изходна памет — 4 Кбайта.

Всяка от тези клетки има собствен адрес, който отговаря на нейния пореден номер. Всичностадресите могат да се представят по два еквивалентни начина — положителен адрес от 0 до 65535 и отрицателен, който се получава, като от положителния адрес се извади числото 65536. Така например управляващата програма Монитор започва от адрес 65385 (\$FF69), който е еквивалентен на —151, и затова по-удобно е влизането в Монитора да става с команда **CALL—151**. Необходимо е още да се напомни, че съдържанието на клетките от рама може да се чете, но в тях (без специален програматор) не може да се записва информация. Останала това не всички клетки от рама са свободни за записване на произволна информация. В една част от тях, в т. нар. служебни клетки, с включването на компютъра се записват точно определени числа, от които зависи функционирането на компютъра. Промяната на никоя може да го направинеработоспособен (най-много до неговото изключване и повторно включване). В КВ. 02. 86. описахме действието на т. нар. "тайнистен параметър", поместен в клетка с адрес 214 (**POKE 214, 128**), който вместо всички бейсиков оператор или команда изпълнява **RUN**. Ще добавим още, че бейсиковите програми трябва да започват (в паметта!) с нулев байт, за което се грижат интерпретаторът, като при включване на компютъра записва на адрес 2048 (\$800) нулев байт. Така че ако запишете в тази клетка число, различно от нула, програмата няма да може да се стартира.

**POKE**

**PEEK**

**CALL**

## Инж. ГЕОРГИ БАЛАНСКИ

Съдържанието на всяка клетка на паметта се установява с функцията **PEEK** (израз), която изразява има стойност от —65535 до 65535. Функцията **PEEK** може да се използва в директен и в програмен режим. За да бъде изобразено на екрана съдържанието на изследваната клетка, функцията **PEEK** се използва заедно с оператора **PRINT**. Например в директен режим:

**PRINT PEEK (33)**  
40

Понеже клетката с адрес 33 е служебна, по-нататък ще видим какво означава числото 40.

С помощта на функцията **PEEK** може да се следи съдържанието на някоя от служебните клетки. Едно от често срещаните приложения в програмен режим е:

**K = PEEK (-16384)**

В случая се извършва проверка за натиснат клавиш от клавиатурата. Ако  $K > 127$ , това означава, че е бил натиснат някой клавиш, като K е кодът на въведенния знак.

Дадената тук малка програма демонстрира как „се прочита“ клавиатурата за натиснат клавиш. Ще допълним само, че за да може да се възприеме следващият подаден от клавиатурата знак, стробиращият сигнал трябва да се нулира чрез **POKE —16368,0**.

```
10 FOR I = 1 TO 2 STEP 0
20 PRINT "НАТИСНЕТЕ ПРОИЗВОЛЕН КЛАВИШ"
30 IF PEEK (-16384) > 127 THEN
40 I = 1
50 NEXT I
50 POKE -16368,0
```

Понеже в една клетка не може да се запише число, по-голямо от 255, адресите (от 0 до 65535) се записват в две последователни клетки — младша и старша, като се приема, че числата в старшата е 256 пъти по-голямо от същото число в младшата клетка. Ето защо, за да се установи адрес, който е записан в две последователни клетки, трябва да се прочете съдържанието и на двете.

Така например началният адрес за разполагане на бейсикови програми при включването на компютъра се записва от Монитора в двете последователни клетки 103 и 104 и можем да го прочетем с:

**Начален адрес → PRINT PEEK (103) + 256 ← PEEK (104).**  
Резултатът е 2049 (\$801).

Краен адрес — PRINT PEEK(175) + 256 \* PEEK(176) дава крайния адрес на бейсиковата програма, записан от Монитора в клетки 175 и 176.

По подобен начин можем да намерим още:

— Началния адрес, от който при заредена програма започват да се разполагат простиите променили (*LOMEM*). Възможно то е и краят на програмата, преди да бъде стартирана.

Начало на прости променили —> PRINT PEEK(105) + 256 \* PEEK(106).

— Последният достъпен адрес за бейсикови програми (*NIMEM*), от който започва разполагането на низовите променили, и крайният им адрес се намират чрез:

Начало на низове —> PRINT PEEK(115) + 256 \* PEEK(116).

Край на низове —> PRINT PEEK(111) + 256 \* PEEK(112). Аналогично може да се получи:

Начало на машинни —> PRINT PEEK(107) + 256 \* PEEK(108).

Край на машинни —> PRINT PEEK(109) + 256 \* PEEK(110).

Като се знаят начините за откриване на началните и крайните адреси, лесно може да се установят и застите области, например от низовите променили:

PRINT PEEK(115) + 256 \* PEEK(116) — PEEK(111) — 256 \* PEEK(112).

Полезно е да се знае как става определянето на началния адрес (*adr*) и дължината на машинната програма (*len*), без да се влизне в Монитора:

adr = PEEK(43634) + 256 \* PEEK(43635)

len = PEEK(43616) + 256 \* PEEK(43617)

Тази информация е необходима най-вече при записване на двоичен файл (машинна програма) върху диска — *BSAVE FILE*, A adr, L len

Ето още няколко често срещани приложения на оператора *PEEK*:

CH = PEEK(36) — отчита броя на позициите от левия край на полето за текст до текущата позиция на курсора;

CV = PEEK(37) — дава абсолютната стойност на вертикалната позиция на курсора;

X = PEEK(-16336) — включ-

ва високоговорителя и предизвиква кратък звуков сигнал;

X = PEEK(218) + PEEK(219) \* 256 — X получава стойност, равна на номера на програмния ред, който съдържа грешка, ако предварително е подадена команда *ONERR GOTO*;

Y = PEEK(222) — дава кода на допуснатата грешка след подаване на командата *ONERR GOTO*.

Операторът *POKE* (*адрес, стойност*) служи за записване в клетки от оперативната памет на десетични числа от 0 до 255.

Преди да продължиш напътък, още веднъж ще напомним казаното в началото за наличието на служебни клетки, чието съдържание не бива произволно да се промени. Ето защо използването на оператора *POKE* трябва да става само след подробно запознаване с разпределението на паметта и адресите на служебните клетки.

Действието на оператора *POKE* може да бъде визуализирано лесно, ако се върнем към таблицата за кодирали на знаците в ПК Правел-82, която публикувахме в миналия брой. Внимателният читател е разбрал, че директно от клавиатурата може да се въвеждат само знаците, които се изобразяват в режим *NORMAL*. Ако обаче запишем с *POKE* в съответни клетки на екранната памет кодовете на знаците в режимите *INVERSE* и *FLASH*, вече ще можем да ги видим на екрана. Тъкмо това прави следната единоредова програма, която в три клетки, разположени през една, записва кодовете на буквата M (латинско) в трите режима.

10 POKE 1500,13: POKE 1502,77: POKE  
1504,205

Ето и най-честите приложения на оператора *POKE*:

#### ОГРАНИЧИТЕЛИ ЗА ТЕКСТОВОТО ПОЛЕ

Нормално полето за текст на видеомонитора съдържа 40 клетки по хоризонтала и 24 по вертикална. Неговите размери могат да се променят в посочените граници, като се променят съдържанието на клетките 32, 33, 34 и 35 от нулевата страница на паметта, в която Мониторът е записал числата 0, 40, 0 и 24.

POKE 32,L — поставя левия

край на полето за текст на позиция L — от 0 до 39. Промяната се установява, когато курсорът се върне в левия край. Този оператор не променя широчината на полето за текст, понеже десният му край се отмества на същото число позиции надясно. За да не се унищожи бейсиковата програма в паметта или друга необходима за работата на компютъра информация, преди да се премести левият край на полето, съответно трябва да се нагласи неговата широчина така, че то да не излязва изън екрана (виж по-долу).

POKE 33,W — определя широчината на полето за текст на стойност W от 1 до 40. POKE 33,0 разрушава програмата в паметта! Ако W е по-малко от 33, третото табулаторно поле на оператора *PRINT* може да се окаже извън екрана.

POKE 34,T — поставя горния край на полето за текст на определена от T стойност в диапазона 0—23. Числото 0 определя мястото на най-горния ред. Горният край на полето за текст не може да се намира под долната му край, определен (в нашия случай) от B. Операторът *POKE 34,4* не позволява писането на текст на първите четири реда.

POKE 35,B — поставя долната край на полето за текст на позиция, определена от B. Параметърът B може да бъде в диапазона 0—24, като 24 определя най-долната ред. Долната граница не може да превишава горната!

Действието на тези четиринummerните е илюстрирано с малка демонстрационна програма.

```
10 HOME
20 VTAB 10: PRINT "ДОЗИ НАЛИЧИЕ Е"
30 VTAB 11: PRINT "ТЕКСТОВОТО ПОЛЕ"
40 POKE 32,20
50 POKE 33,16
60 POKE 34,8
70 POKE 35,16
80 HOME
90 PRINT "ИЗПОЛУЧИТЕЛНА Е 16"
100 PRINT "ВИДОВАТА Е 8"
110 FOR N = 1 TO 400: NEXT
120 GOTO 90
```



От тези ограничители най-често се използва РОКЕ 33,33 при внасяне на корекции в програмни редове. Обикновено това става така. Редът, в който искаме да внесем корекция, се извика с LIST или преглеждането на цялата програма се прекъсва с MK-C на желаното място. След натискане на ОСВ курсорът се придвижва с клавишите I, J, K, M в началото на реда и чрез движение с клавишите MK-U (евентуално и MK-H) отново се прочитат всички знаци, през които той премине. Тези, които не са необходими, се изтриват чрез натискане на интервалния клавиши или върху тях се написват новите.

Тук обаче има малка особеност. Писането на экрана става с 40 знака на ред, но след командата LIST компютърът поддържа знаците по 33 на ред. Така че остават празни 7 позиции, които курсорът обхожда и ако те се намират в текст (PRINT „XXXXXX“), се записват в паметта като празни полета. При стартиране на така коригираната програма тези празнини остават и текстът се накъсва.

Ето защо, когато се налага корекция в редове с повече от 33 знаци, съдържащи надписи, първо се подава команда РОКЕ 33,33 и програмата или редът се прочitat (листват), за да се установят на широчина на текстовото поле с 33 знака. След внасяне на корекциите по описанния начин нормалното текстово поле се възстановява с РОКЕ 33,40 или TEXT.

Необходими са и още някои пояснения относно използването на оператора РОКЕ за изменение на размерите на полето за текст. Текстът, който се появява на экрана, е записан в специална част на паметта (т. нар. екранна памет от адрес 1024 до 2039) и символите, чийто кодови са записани в тези клетки, се изобразяват на экрана. 1024 е адресът на точката, наимаща се най-горе вляво, а диаметрално разположената последна точка на экрана има адрес 2039. Ето защо всички промени в полето за текст трябва така да се правят, че като цяло да не излизат извън областта на екранната памет. Ако това стане, информа-

цията ще се запише в клетки, чиято съдържание не се изобразява на экрана, и може да бъде унищожена информацията, съдържаща се в други клетки. Затова трябва да се спазват посочените по-горе ограничения при избора на размерите на полето за текст.

### КЛЮЧ ЗА ГРАФИЧНИ И ТЕКСТОВИ СТРАНИЦИ

За изобразяване на текстова и графична информация в паметта на компютъра са отделени четири области, наречени още страници.

— Първа страница (\$400—\$07FF) за текстове и графики с ниска разделителна способност.

Достъпът до нея става обикновено с операторите TEXT или GR.

— Втора страница (\$800—\$0BFF) за текст и графика с ниска разделителна способност.

— Първа страница за графики с висока разделителна способност (\$2000—\$3FFF). Използва се с оператора HGR.

— Втора страница за графики с висока разделителна способност (\$4000—\$5FFF). Използва се с оператора HGR2.

Различните режими за работа се избират или със съответните текстови и графични оператори (вж. КБ. 06, 85, 01, 86 и 02, 86), или с РОКЕ-ключове, което покоят се предвижда (вж. „Кука за абордаж“, КБ. 02, 86).

### 1. РОКЕ — 16304,0

Превключва от текстов режим в режим на цветна графика, без да изчиства екрана до черно. В зависимост от състоянието на другите три ключа графичният режим може да бъде с ниска или висока разделителна способност (1 или 2 страница), графично изображение на цял экран или съдържащо и текст.

### РОКЕ — 16303,0

Превключва от режим на цветна графика в текстов режим, като в зависимост от ключа 2, се избира първа или втора страница.

### 2. РОКЕ — 16300,0

Превключва от страница 2 на страница 1 (текстова с ниска или с висока разделителна способност), без да изчиства екрана или да премества курсора. Това е необходимо например при преминаване от Разширен Бейсик към Целочислен Бейсик. В противен случай ще продължите да наблюдавате само втора страница.

### РОКЕ — 16299,0

Аналогично превключва от 1 на 2 страница.

### 3. РОКЕ — 16298,0

Превключва една страница за графика с висока разделителна способност към съответната текстова страница, без да изчиства екрана. Това се прави при преминаване към Целочислен Бейсик, възто то гава операторът GR погрешно ще покаже страница за графика с висока разделителна способност. Изборът на страниците зависи от ключ 2.

### РОКЕ — 16297,0

Подобна на предишната команда, но превключва от текстова към страница за графика с висока разделителна способност.

### 4. РОКЕ — 16301,0

Превключва от графики на цил екран към смесен режим „графика и текст“ — четири реда за текст в долната край на екрана. В зависимост от състоянието на останалите ключове в горната част на екрана може да има текст, графика с ниска или с висока разделителна способност. Изображените и на двете части на екрана идват от страници с единакъв номер — 1 или 2.

### РОКЕ — 16302,0

Превключва от смесен режим на графика и текст към режим на графика на цял екран. Изображенето на горната част на екрана може да бъда текст, графика с ниска или графика с висока разделителна способност.

## В СЛЕДВАЩИЯ БРОЙ:

- Смесени бейсикови и машинни програми.
- Формиране на блокове DATA

ПРОФЕСИОНАЛНО  
КРЪЩЕНИЕ

## АСЕМБЛЕР

и

## МАШИНЕН ЕЗИК

ОРЛИН ВЪЛЧЕВ  
БОРИСЛАВ ЗАХАРИЕВ

### 1 Инструкции за МП 6502 и съответните им оператори от асемблера

**Машинна инструкция** е битов низ с определен формат, който микропроцесорът възприема като единна конструкция, съдържаща пълна информация за изпълнението на една машинна операция.

**Операторът** е конструкция в символичен език за програмиране (от високо ниво или асемблер), с която на съответния транслейтор се дава пълна информация за изпълнение на операция или група операции.

При асемблерите и макроасемблерите на всяка машинна инструкция съответства оператор по асемблера, но има асемблерски оператори, които или не съответстват на машинна инструкция или съответстват на няколко машинни инструкции.

Навсякъде при описание на машинните инструкции ще бъде даван и формата на съответната оператор от асемблер. За именование на машинните инструкции ще бъдат използвани мнемоничните кодове на съответните им оператори от асемблера, като допълнително се уточнява и типът на адресиране.

В асемблера за 6502 всеки оператор има точно определен формат. Мнемоничните кодове на операциите са трисимволни и са получени от английските наименования на инструкциите. При писането на програма на асемблер трябва точно да се спазват формата и правописът на операторите.

При описание на инструкциите ще бъдат използвани следните означения и съкрашения:

**оп** – Операнд на инструкцията. Може да бъде число (при непосредствено адресиране), адрес или символично име;

**дълж** – дължината на инструкцията в байтове (1, 2 или 3);

**коп** – Код на Операция. Може да бъде мнемоничен код или машинен код.

Тук ще се употребява като машинен код на операцията – шестнайсетично число;

**цик** – Машинни Цикли; броят цикли на микропроцесора, за който той ще изпълни инструкцията. При първите си програми на асемблер няма да обръщате внимание на тази информация, но когато се пишат програми, работещи в реално време, се изисква познаване на точното време за изпълнение. То се получава като се сумират машинните цикли на всички инструкции, които ще се изпълняват, и след това броят им се умножи по времето за един машинен цикъл (приблизително 1 микросекунда при микропроцесор 6502).

Обрнете особено внимание на промяната на регистъра на състоянието след изпълнение на всяка една от инструкциите!

### 1.1 Инструкции за зареждане

#### 1.1.1 Инструкции за зареждане на регистри (Load)

Това са трите инструкции с мнемонични кодове LDA, LDX и LDY. На английски Load означава зареждам, а последната буква от мнемоничния код показва за кой регистър се отнася командалата – A (акумулатора), X (регистър X) или Y (регистър Y).

### 1.1.1.1 LDA (Load Accumulator from memory)

Зареждаме на акумулатора от паметта

Резултат:

(M) => A

Променя флагове N и Z.

P	*	-	*	*
---	---	---	---	---

N V B D I Z C

N - Negative (отрицателно)	D - Decimal (десетично)
V - Overflow (препълване)	I - Interrupt (прекъсване)
- (не се ползва)	Z - Zero (нула)
B - Break (прогр. прек.)	C - Carry (пренос)

Флагът Z получава стойност 1, ако зареденото в акумулатора число е нула. Флагът N се позиционира в 1, когато новата стойност на акумулатора е отрицателно число. Той показва дали резултатът е положителен или отрицателен. В допълнителен код старият бит е бит за знак - 0 при положителни числа и 1 при отрицателни. Така в един байт може да бъде записано число със знак от -128 до 127, в два байта - от -32768 до 32767 и т.н. На практика тук се копира знаковият бит (бит 7) от акумулатора в N. При резултат от \$00 до \$7F (0 до 127) N е нула, а при резултат от \$80 до \$FF (-128 до -1) N е единица.

Чрез тази инструкция акумулаторът може да бъде зареден с дадена стойност.

Тип адресиране	Асемблерски запис	Код	Дълж	Мн
Непосредствено	LDA #on	A9	2	2
Пряко в нулевата страница	LDA on	A5	2	3
В нул. стр. с инг. по X	LDA on,X	B5	2	4
Пряко	LDA on	AD	3	4
Пряко с индексиране по X	LDA on,X	BD	3	4
Пряко с индексиране по Y	LDA on,Y	B9	3	4
Косвено с предбр. индекс.	LDA (on,X)	A1	2	6
Косвено с посл. индекс.	LDA (on),Y	B1	2	5

**ПРИМЕР:** Нека клетката от паметта с адрес \$53FB съдържа стойността \$05 (по-нататък това ще бележим така:  $(\$53FB)=\$05$ ), а от адрес \$30A се изпълнява инструкцията LDA в режим пряко пълно адресиране.

Условие:

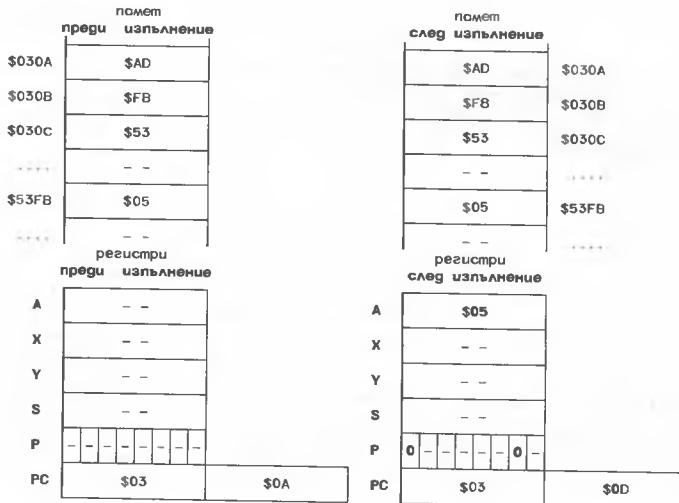
$(\$53FB)=\$05$

Оператор в асемблера:

LDA \$53FB

Машинна инструкция:

AD	F8	53
----	----	----



● Изпълнение на последователност от команди, въведени в един ред. Въведеният ред може да съдържа 255 знака. Ето един пример:

```
*300.305 300:18 69 01 I 300.302 RETURN
0300-02 02 00 00 00 00
0300-18 69 01 -> Извършен ща
на ънитите
```

Не е нужно командите L, I и N да бъдат отделяни с интервал. Ако при въвеждането е допусната синтаксична грешка, Мониторът ще изпълни всички команди до тази, в която е грешката, след което реагира със звуков сигнал.

● Дублирвани на зададена поредица от стойности от началния адрес на областта:

Въвежда се желаната поредица от стойности от началния адрес на областта:

```
*300:22 33 RETURN
```

В този случай се използува командала за "преместване M, но в по-особен вид:  
 [НАЧ. АДРЕС + N] <  
 [НАЧ. АДРЕС].  
 [КР. АДРЕС - N]. M

Не са броят на стойностите в поредицата, а [НАЧ. АДРЕС] и в двете части на командния ред е иначалният адрес на интересуващата ни област от паметта.

Най-напред командала M ще запише поредицата от стойности, следващи тези, в които е въведена поредицата, после в следващите ги и така, докато се запълни цялата зададена област на паметта.

```
*302<300..306M RETURN
```

```
*300..306 RETURN
```

```
0300-22 33 22 33 22 33 22 33
```

```
0306- 22
```

## УПРАВЛЯВАЩА ПРОГРАМА

# МОНИТОР

Инж. ДИМИТЪР  
ЕВСТАТИЕВ

(продължава от КВ. 01. 86.)

● Откриване на зададени поредици от стойности в област от паметта.

В този случай се използува командала за сравнение V във вида, в който използваме командала M за реализиране на предишната възможност. В този вид командала V служи за удобна проверка, дали всички клетки в определена област от паметта съдържат една и съща стойност.

Първоначално нулираме областта с начален адрес \$301 и краен адрес \$307:

```
*300:0 RETURN
```

```
*301<300..306M RETURN
```

При сравняването се оказва, че всички клетки от тази област съдържат една и съща стойност — нула:

```
*301<300..306V RETURN
```

При сравняване след промяна на съдържанието на клетка с адрес \$304 се получава:

```
*304:4 RETURN
*301<300..306V RETURN
0303-00 (04)
0304-04 (00)
```

● Създаване на собствени команда. Командата

MК—Y RETURN

осъществява преход към клетка от паметта с адрес \$3F8. В тази клетка можете да запишете машинния код [\$4C] на инструкцията безусловен преход [JMP]. В следващите две клетки [\$3F9 и \$3FA] можете да запишете адреса на ваша потребителска програма на машинен език. При това положение всеки път, когато подадете командала MК—Y, компютърът ще изпълнява вашата програма.

```
*300:20 58 FC 60 RETURN
```

```
*3F8:4C 0 3
```

В този пример при въвеждането на командала MК—Y програмата с начален адрес \$300 ще изчиства екрана. Мониторната подпрограмма HOME е с начален адрес \$FC58.

# ЗА И ПРОТИВ



Компетентният подход към въпроса изисква предварително да уточним за кой вид джойстик става дума. Най-общо джойстиките се делят на два вида — потенциометрични и с микропревключватели. Първите позволяват плавно движение на управляния обект на екрана на видеомонитора, а при вторите отклоняванието на посоката става рязко. За пълнота ще споменем и т. нар. „педали“, които представляват два механично независими един от друг (несъврзани) потенциометра (вж. КВ. № 85.), така че управляемият им движението става чрез едновременното въртене и на двата потенциометра с две ръце, към което трябва да се прибави и едновременното натискане на два бутона! Охвидно е, че подобно устройство е пригодено предимно за факири и в практиката се използва най-вече за работа с някои графични програми (не игри).

Но да се върнем към истинските джойстици. Проблемите при направата на потенциометричните джойстици идват не от електрическото съединение на потенциометрите, а от сложните механизми.

И двата потенциометра трябва да могат да се въртят едновременно чрез движението на един-

ствения лост и отгоре на всячко да има възможност за допълнително завъртане на тилото им около осите за justиране на устройството. Представа за сложността на конструкцията може да се добие от снимка 2, на която е показан разглобен джойстикът на СДМПБУ — Габрово и ИВСД „Авантгард“. Виждат се потенциометри с механизм за justиране (1), шарнирното лагеруване на оста (2), двете взаимноперпендикулярни дълги (3), в чиното прорези влизат краят на лоста (4). Дългите лагеруват в единия край на кутията, а в другия те са захванати неподвижно за осите на потенциометрите. Пружините (5) връщат дългите и потенциометрите в междинно (нейтрално) положение. Повече подробности за този джойстик и програми за неговото justиране можете да намерите във в. „Направи сам“ бр. 11. 85. И така отговорът на въпроса: „Струва ли си да правим сами потенциометричен джойстик?“, е вече повече от ясен и затова ще посто-

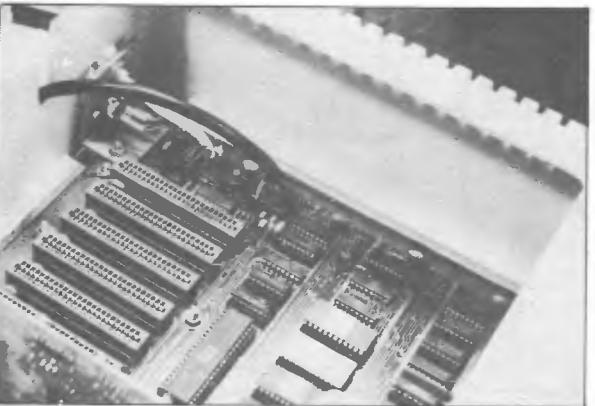
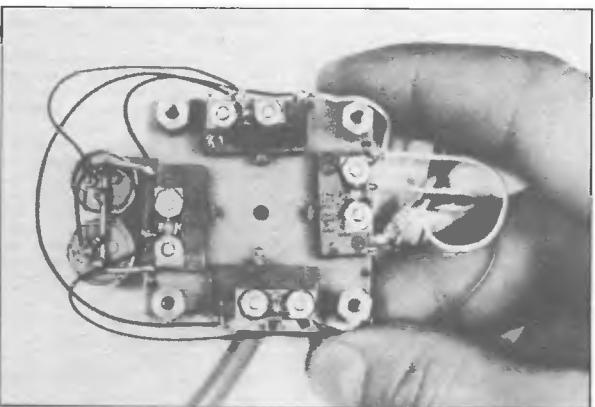
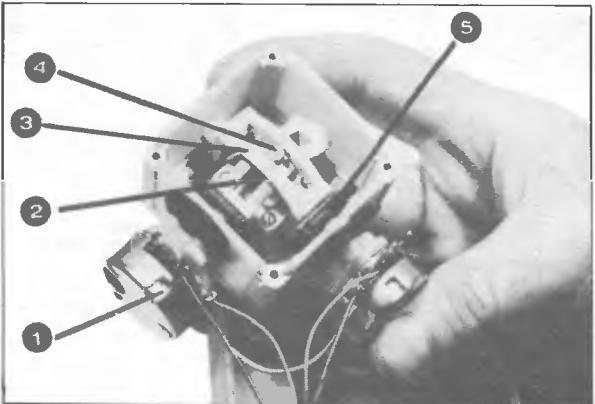
rim и тук адреса, от който можете да го получите:

София  
ул. Бабуна планина 2а  
ИВСД „Авантгард“  
тел. 66-50-84

По различен начин стои обаче въпросът с другия тип джойстик — този с микропревключвателите. Той все още не се произвежда у нас, но направата му е напълно по възможностите на всеки средно сърден любител, частите за него не са дефицитни, няма вътръщи и триещи се части и практически не се износва. Ще добавим още, че почитат игри изискват именно джойстик, който да позволява рязка промяна на посоката на движение на обектите, което не може да се постигне с потенциометричния джойстик. От своя страна пък малкото игри, пригодени за потенциометричен джойстик, напълно приемливо се управляват и с джойстик с микропревключватели.

Специално за читателите ни нашият автор и консултант инж. Румен Велев разработи оригинална конструкция на такъв джойстик (снимки 1 и 3), която публикуваме във в. „Направи сам“, бр. 03. 86.

На снимка 4 е показано включването на джойстика.



```

10  HOME
: DIM C$(35)
: FOR A = 0 TO 35
:   READ C$(A)
: NEXT
: INPUT "ВХОДНА СИ
: СТЕМА -";B
: INPUT "ИЗХОДНА С
: ИСТЕМА -";C
: INPUT "^ИСЛОТО -
: ";A$
: FOR D = LEN (A$)
:   TO 1 STEP -
:   1
:   B$ = MID$ (A$,D,1)
:   E = 0
:   FOR A = 0 TO B -
:     1
:     E = E + A *
:       (B$ = C$(A))
:   NEXT
:   F = F + INT (E *
:     B ^ (LEN
:     (A$) - D))
: NEXT D
: G = F

20  FOR H = 1 TO INT
:   ( LOG (F) /
:   LOG (C) ) +
:   1
:   I = INT (G / C
:   )
:   J = G - C * I
:   D$ = C$(J)
:   E$ = D$ + E$
:   G = I
: NEXT
: PRINT "ОТГОВОР -
: ";E$
: DATA 0,1,2,3,4,5
: ,6,7,8,9,A,B,
: C,D,E,F,G,H,I
: ,J,K,L,M,N,O,
: P,Q,R,S,T,U,V
: ,W,X,Y,Z
: GET LL$
: RUN

```

## **Анатомия**

# ПЪТЕВОДИТЕЛ

(Продолжение  
от №8, 02, 86)

# ИЗ ПРАВЕЦ-82

В този брой ще ви запознаем с основните режими на работа на микрокомпютъра Правец-82 и функционалната му схема — фиг. 1. В нея са дадени връзките и взаимодействието между отделните блокове.

За управлението на основните режими на работата на микропърсът генераторът изработва тактови импулси. Всички режими на работа на микропърсът са синхронизирани с тях. А микропроцесорът 6502 изработва сигнал четеене-запис, който управлява посоката на данните по информационната му магистрала.

## **1. ЗАПИС В ОПЕРАТИВНАТА ПАМЕТ (фиг.1)**

При активен сигнал за валидно (логическа 0 на шината чете-член) записът двупосочните буфери за данни са отворени в посока от МП 6502 към информационната магистрала. Микропроцесорът извежда през адресните буфери (единолосочни) адреса на клетката от оперативната памет, в която ще се валидизат данните от информационната магистрала. Този адрес избира определена клетка от оперативната памет и по време на управляващия сигнал за запис данните от магистралата се записват в нея.

След записването тези данни са достъпни за четене от МП 6502, за видеоблока или свързаното към микрокомпютъра аходно-изходно устройство.

**2.**  
**ЧЕТЕНЕ**  
**ОТ ОПЕРАТИВНАТА**  
**ПАМЕТ**  
**(фиг. 2)**

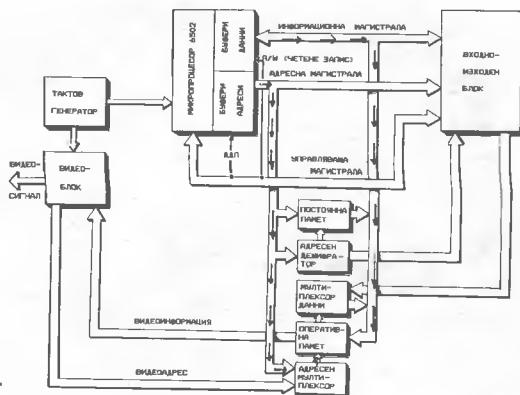
При активен сигнал за четене (логическа 1 на шината запис-чтение) даупосочните буфери за

ните буфери адреса на клетка от постоянната памет. Адресният декодер активира перото за избор (Chip Select) на адресната епром. При нормално разрешените находища за данни на епрома данните от съответната клетка се извеждат на информационната магистрала и през буферите за данни се пращат в МП 6502 за обработка.

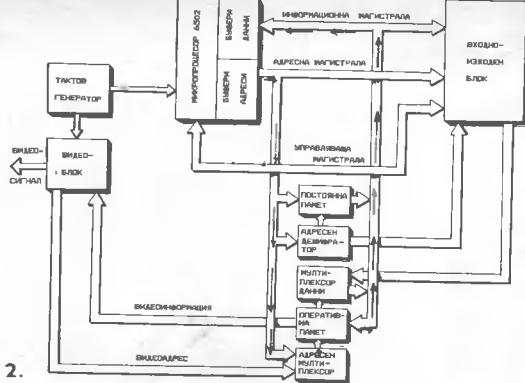
4.  
ЧЕТЕНЕ  
ОТ КЛАВИАТУРАТА  
(ФИГ. 4)

Клавнатурата на микрокомпютъра се разглежда като входно-изходно устройство и се адресира от входно-изходния блок. Специална подпрограма в управляващата програма Монитор следи за състоянието на клавнатурата. При натиснат бутон през мултиплексора за данни на информационната магистрала постъпват седем бита и през буферите за данни — в микропроцесора 6502. Ако осмият бит (17) е единица, има натиснат клавиш в това означава, че седемте бита данни съдържат буквено-цифрова информация. Те служат за адресиране на знаковия генератор на микрокомпютъра (също тип епром с капацитет 2 Кбайт). С адресирането и прочитането на съответната клетка от него и чрез последващо преобразуване на екрана на видеомонитора се изобразява знакът на натиснатия клавиш.

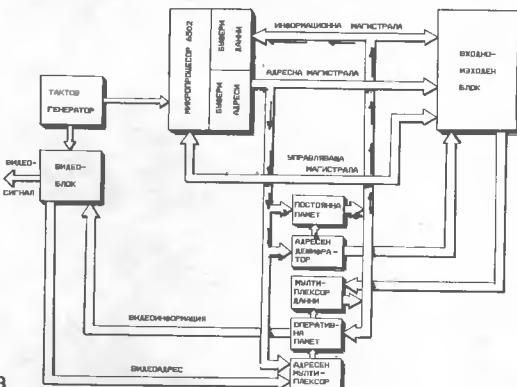
**Инж. БОРИС БАЧКОВ**  
**ИТКР — БАН**  
направление „Персонални  
компютри и системи“



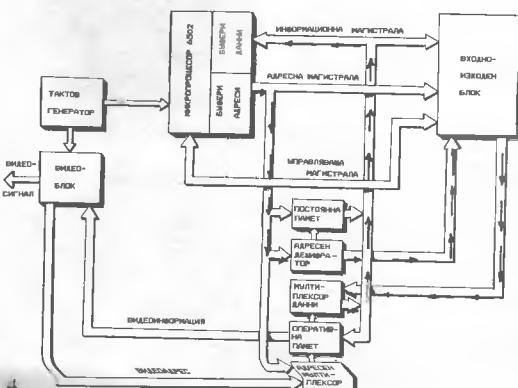
# СООТУЕР



2.



3.



4.

Една приятна новина, драги читатели. От този брой нататък всеки от вас придобива автоматичен имунитет спрямъчно на собствената си разселяност. След всяка програма, която публикуваме, ще отпечатваме специална парола. Тя представлява сума на байтовете с натрупване от началото на програмата до означения в таблицата ред. Въоръжени с нея, вие ще можете да работите много поспокойно. Само от време на време ще свирявате вашата сума с отпечатаната. Ако ли са единакви, продължавайте набирането. Ако ли не, воднага спирате и правите поправка, докато грешката е още гореща.

Идеята не е съвсем нова, но само няколко от най-реномираните списания в света са въвели тази практика.

Поради това умоляваме авторите отсега нататък да ни изпращат листингите на своите програми, придружени от таблица на контролните суми.

**КОМПЮТЪР ЗА ВАС**

# ЛЕКАРСТВО ПРОТИВ РАЗСЕЯНОСТ

## КОНТРОЛНА СУМА

ПЕТЪР РАИКОВ

Всеки програмен ред, поместен в оперативната памет на компютъра, се разполага по следния начин: двубайтов указател за началото на следващия ред, два байта за номер на реда, текст на реда в кодиран вид и един байт \$00 за край на реда. Програмата „Контролна сума“ намика къде е началото на тестваната програма (то може да е на адрес, различен от \$0801), изчислява суми от байтовете, съответни на отделните редове, и отпечатва на екран или принтер натрупавши се контролни суми през групи от 5 и 10 реда. Накрая отпечатва общата контролна сума на цялата програма. Извеждането на сумите може да бъде спирano с клавиши „интервал“ — при неговото натискане те се отпечатват поединично.

Редове 10 и 20 зареждат машинна програма на адреси от 37784 до 38398 непосредствено под областта, в която се разполага DOS. Ред 30 създава на дискаетата двоичен файл „КОНТРОЛНА СУМА. ОВД“. Той може да бъде стартериан с команда BRUN дори когато в паметта на компютъра вече е въведена тестваната програма. Тези, които нямат филодискови устройства, могат да запишат машинната програма на касетофон, като:

1. След въвеждане на програмата „Контролна сума“ (без ред 30) тя се стартериа с RUN.

2. Изчаква се около 3 секунди и се подава командата:

CALL — 151

за прамишване в Монитора.

3. Включва се касетофонът на запис и се подава 9398.95FEW

Зареждането става с командите:

9298.95FER 9398C

Зареждането на двоичната програма не се отразява на бейсиковата програма, въведена преди това в паметта на компютъра. Трябва да се има предвид, че автоматично се установяват HIMEM: 37886.

**ВАЖНО: ИЗПОЛЗУВАНЕТО НА MAXFILES ЦЕ УНИЩОЖИ ПРОГРАМАТА!**

Веднъж стартериана, тя може да се извика многократно с командите &5 или &10, съответствуващи на 5 или 10 програмни реда (независимо от тяхната номерация). Задаването на друга стъпка ще доведе до съобщение за грешка.

Трябва да се има предвид, че програмата включва в изчисляването на контролните суми и програмни редове, съдържащи оператора REM. Ето защо те не бива да бъдат пропускани съзнателно, за да се ускорява въвеждането.

```

0  REM *****
1  REM *
2  REM *      ПРОГРАМА   *
3  REM *      КОНТРОЛНА   *
4  REM *      СУМА    *
5  REM *
6  REM *****
7  REM
10 FOR ADDR = 37784 TO 38398
20 READ CODE: POKE ADDR, CODE: NEXT
30 PRINT CHR$(4); "BSAVE КОНТРО
ЛНА СУМА.ОВД", A$9398, L$0267"
40 END
100 DATA 169,76,141,245,3,169,25
5,141,246,3,169,147,141,247,
3,169,254,133,115,169,147,13
3,116,32,88,252,32,231,148,2
40
101 DATA 242,239,231,242,225,237
,225,244,225,160,245,243,244
,225,238,237,247,241,247,241
,160,200,201,205,197,205,186
,160,0,166,115
102 DATA 165,116,32,36,237,32,23
1,148,174,141,141,243,244,22
5,242,244,233,242,225,160,24
3,229,160,243,160,38,53,160,
233,236,233
103 DATA 160,38,49,48,174,141,14
1,0,76,3,224,201,53,240,29,2
01,49,240,3,76,201,222,32,17
7,0,201,48,208,246,32,177
104 DATA 0,201,0,208,239,169,10,
141,225,148,76,44,148,32,177
,0,201,0,208,224,169,5,141,2
25,148,160,5,167,0,153,225
105 DATA 148,136,208,250,165,103
,133,6,165,104,133,7,32,142,
253,32,231,148,243,244,249,2
40,235,225,160,0,174,225,148
,169,0
106 DATA 32,36,237,32,231,148,16
,0,242,229,228,225,174,141,14
1,0,32,227,149,32,231,148,16
0,239,244,160,160,160,160,16
1,160,160
107 DATA 228,239,160,160,160,160
,160,242,229,228,160,160
,161,160,163,160,242,229,228
,160,161,243,245
108 DATA 237,225,141,0,32,227,14
9,32,11,149,32,80,149,32,132
,149,238,226,148,169,1,205,2

```

85

## СТЪПКА 5 РЕДА.

ОТ	ДО	КОНТР.
# РЕД	# РЕД	СУМА
0	- 4	1565
5	- 20	1240
30	- 102	5460
103	- 107	7085
108	- 112	7076
113	- 117	6F2C
118	- 121	3810
КОНТР. СУМА		E 0449

```

26,148,208,12,32,112,149,169
,7
109 DATA 133,36,169,173,32,237,2
53,173,225,148,205,226,148,2
08,217,169,9,133,36,32,112,1
49,169,16,133,36,32,122,149,
32,142
110 DATA 253,169,0,141,227,148,1
41,228,148,141,226,148,32,94
,149,76,150,148,255,255,255,
255,255,255,255,104,141,
248,148,104
111 DATA 141,249,148,238,248,148
,208,3,238,249,148,173,255,2
55,240,6,32,237,253,76,239,1
48,173,249,148,72,173,248,14
8,72,96
112 DATA 160,0,177,6,133,8,32,40
,149,177,6,133,9,32,40,149,1
65,8,208,235,165,9,208,231,1
04,104,76,155,149,230,6
113 DATA 208,222,230,9,96,72,24,
109,227,148,141,227,148,169,
0,109,228,148,141,228,148,10
4,24,109,229,148,141,229,148
,169,0
114 DATA 109,230,148,141,230,148
,96,160,1,177,6,141,224,148,
136,177,6,141,223,148,96,173
,0,192,9,128,201,160,208,8,1
41
115 DATA 16,192,173,0,192,16,251
,96,174,223,148,173,224,148,
32,36,237,96,174,227,148,173
,228,148,32,65,249,96,160,0,
177
116 DATA 6,32,47,149,32,40,149,1
65,6,197,8,208,242,165,7,197
,9,208,236,96,169,0,205,226,
148,240,17,169,9,133,36
117 DATA 32,112,149,169,16,133,3
6,32,122,149,32,142,253,169,
16,133,36,32,231,148,173,173
,173,173,141,235,239,238,244
,242,174
118 DATA 160,243,245,237,225,160
,160,229,160,160,0,174,229,1
48,173,230,148,32,65,249,32,
142,253,32,142,253,141,16,19
2,96,32
119 DATA 231,148,173,173,173,173
,173,173,173,171,171,173,173
,173,173,173,173,171,173,173
,173,173,173,173,173,141,0,96
120 REM ПЕТЪР РАЙКОВ
121 REM СОФИЯ - 1986 Г.

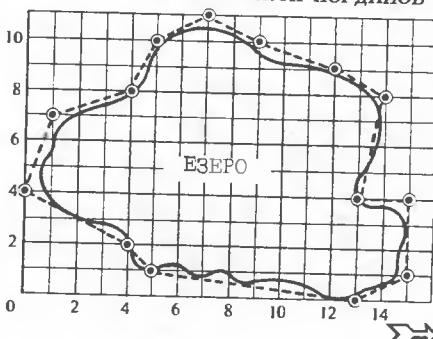
```

## ПЛОЩ НА ПОЛИГОН

Програмата изчислява лицето на неправилна равнинна фигура. За да се изпълни това, е необходимо съответният чертеж да бъде наложен върху мрежа с координатна система (като е показвано на фигуранта — взет е пример с езеро). Въвеждат се координатите A и B на всички върхове, като A<sub>1</sub>, ..., A<sub>n</sub> са координатите на съответните върхове по оста X, а B<sub>1</sub>, ..., B<sub>n</sub> — съответните координати по оста Y. Въвеждането на координатите трябва да става по последователност на върховете.

След изчисляване на площи програмата може да се регистрира чрез натискане на RETURN. Излизането от програмата става, като се въведе нула за брой на върховете.

Инж. АЛЕКСИ ИОРДАНOV



## ПЛОЩ НА ПОЛИГОН

БРОЙ НА ВЪРХОВЕТЕ (ВЪВЕДЕТЕ О ЗА КРАИ) : 14

КООРДИНАТИ НА ВРЪХ 1	(A,B) :	0,4
ВРЪХ 2	(A,B) :	1,7
ВРЪХ 3	(A,B) :	4,8
ВРЪХ 4	(A,B) :	5,10
ВРЪХ 5	(A,B) :	7,11
ВРЪХ 6	(A,B) :	9,10
ВРЪХ 7	(A,B) :	12,9
ВРЪХ 8	(A,B) :	14,8
ВРЪХ 9	(A,B) :	13,4
ВРЪХ 10	(A,B) :	15,4
ВРЪХ 11	(A,B) :	15,1
ВРЪХ 12	(A,B) :	13,0
ВРЪХ 13	(A,B) :	5,1
ВРЪХ 14	(A,B) :	4,2

ПЛОЩ = 108

```

10 HOME : PRINT TAB(9)" ПЛОЩ Н
   А ПОЛИГОН"
20 PRINT
30 INPUT "БРОЙ НА ВЪРХОВЕТЕ (ВЪВ
   ЕДЕТЕ О ЗА КРАИ) : ";N
39 REM КРАИ НА ПРОГРАМАТА ?
40 PRINT : IF N < 1 THEN 230
49 REM ТАБЛИЦА НА КООРДИНАТИТЕ
   ТРЕЯВА ДА СЪДЪРЖА БРОЯ ПОЗИ
   ЦИИ, РАВЕН НА БРОЯ НА ВЪРХОВ
   TE + 1
50 DIM X(N + 1),Y(N + 1)
69 REM ЦИКЪЛ ЗА ВЪВЕЖДАНЕ НА КО
   ОРДИНАТИТЕ ПО РЕДА НА ВЪРХОВ
   ЕТИ
70 FOR I = 1 TO N
80 IF I > 1 THEN 110
90 PRINT " КООРДИНАТИ НА ВРЪХ "
   ;I;
100 GOTO 120
110 PRINT "                   ВРЪХ
   ";I;
120 HTAB 25: INPUT "(A,B) : ";X
   (I),Y(I)
130 NEXT I
139 REM ПЪРВИЯТ ВРЪХ ИГРАЕ РОЛЯ
   И НА ПОСЛЕДЕН ВРЪХ
140 X(N + 1) = X(1)
150 Y(N + 1) = Y(1)
160 A = 0
169 REM ИЗЧИСЛЕНИЕ НА ПЛОЩТА; П
   ЕЧАТ
170 FOR I = 1 TO N
180 A = A + (X(I) + X(I + 1)) * (
   Y(I) - Y(I + 1))
190 NEXT I
200 PRINT : PRINT " ПЛОЩ = "; ABS
   (A) / 2
210 PRINT
219 REM ПРОГРАМАТА ДА СЕ ИЗПЪЛН
   И ОТНОВО
220 INPUT "НАТИСНЕТЕ 'RETURN', З
   А ДА ИЗПЪЛНИТЕ ПРОГРАМАТА О
   ТНОВО ";Z$: CLEAR : GOTO 10
230 END

```

## СТЪПКА 10 РЕДА.

ОТ	!	ДО	!КОНТР.
# РЕД	!	# РЕД	!СУМА
10	-	80	67C9
90	-	169	401B
170	-	230	40FB
			КОНТР. СУМА Е EBDC

**"рН-МЕТЪР"**

Целта на програмата е да пресмята рН-стойности на разтвори от протолити, като се използва математически метод с определено приближение. Ценно в случая е бързият и удобен начин за изчисление на рН за по-сложни по състав разтвори, съдържащи повече киселини и основи едновременно. По този метод могат да се моделират експерименти в аналитичната химия, без да се използват скъпи или дефицитни вещества.

Програмата работи с премества точност при изчисляване на рН на силни киселини при концентрации под 0,1 М, а за силини основи — под 0,001 М. Причината за намаляването на получаваната точност при по-големи концентрации е в използването математическия апарат (метод на Нютон), но това не е проблем, защото теоретичното изчисляване на рН на концентрирани силни киселини и основи е много просто и не изисква компютър.

При изчисляването на рН на слаби киселини и основи точността на методиката е от 0,1 до 0,01 рН-единици, което е напълно достатъчно, като се имат предвид неизбежните приближения, съществуващи в разтвора динамично равновесие и точността на повечето от използвани рН-метри. Трябва да се знае, че точност 0,01 рН-единици се постига само от модерни цифрови апарати, а обикновените имат точност 0,1 рН-единици.

Друго достойниство на програмата е възможността за изчисляване на рН на разтвори на соли, получени от слаби киселини и основи — а случаи точността също е 0,01 рН-единици.

Контролен пример:

Смесени са 25 куб. см оцетна киселина  $\text{CH}_3\text{COOH}$  с концентрация 0,1 М (една равновесна константа  $K_A = 1,75 \cdot 10^{-5}$ ) с 5 куб. см натриева основа  $\text{NaOH}$  (една равновесна константа  $K_B = 1,0$ ).

Инж. АНЕЛИЯ ЕРМЕНКОВА

Инж. ВАСИЛИ СТЕФАНОВ



RUN  
 \* \* \* ПРОГРАМА ЗА ПРЕСМЯТАНЕ НА РН \* \* \*  
 БРОЙ КИСЕЛИНИ ?  
 БРОЙ ОСНОВИ ?  
 БРОЙ РАВНОВЕСНИ КОНСТАНТИ ЗА КИСЕЛИНА #1?  
 KA1 = ?;1.75E-5  
 КОНЦЕНТРАЦИЯ = ? (МОЛ/ЛИТЪР) 0.1  
 ОБЕМ = ? (КУБ.СМ) 25  
 БРОЙ РАВНОВЕСНИ КОНСТАНТИ ЗА ОСНОВА # 1?  
 KB1 = ?  
 КОНЦЕНТРАЦИЯ = ? (МОЛ/ЛИТЪР) 0.1  
 ОБЕМ = ? (КУБ.СМ) 5  
 1 РАЗТВОР : РН = 4.14898775  
 ТОВА БЕШЕ ПОСЛЕДНОТО НАМЕРЕНО РЕШЕНИЕ  
 ЖЕЛАЕТЕ ЛИ ГРАФИКА? ДА/НЕ ?  
 НЕ

```

10 REM 'РН-МЕТЪР'
20 :
30 HOME : VTAB 5: PRINT "* * * П
РОГРАМА ЗА ПРЕСМЯТАНЕ НА РН
* * *": PRINT : PRINT
40 INPUT "БРОЙ КИСЕЛИНИ ?";NA: PRINT
50 INPUT "БРОЙ ОСНОВИ ?";NB: PRINT
60 DIM KA(NA,5),KB(NB,5),CA(NA,2
),CB(NB,2),VA(NA),VB(NB),FA(
256),XA(256):SA = 0:SB = 0
65 IF NA = 0 GOTO 135
70 FOR I = 1 TO NA
80 PRINT : PRINT "БРОЙ РАВНОВЕС
И КОНСТАНТИ ЗА КИСЕЛИНА #";I
;;: INPUT KA(I,0): PRINT
90 FOR J = 1 TO KA(I,0)
100 PRINT "KA";J;" = ";: INPUT K
A(I,J)
110 NEXT J
120 PRINT : INPUT "КОНЦЕНТРАЦИЯ
= ? (МОЛ/ЛИТЪР) ";CA(I,1): INPUT
"ОБЕМ = ? (КУБ.СМ) ";VA(I):SB
= BA + VA(I): PRINT
130 NEXT I
135 IF NB = 0 GOTO 205
140 FOR I = 1 TO NB
150 PRINT : PRINT "БРОЙ РАВНОВЕС
ИИ КОНСТАНТИ ЗА ОСНОВА # ";I
;;: INPUT KB(I,0): PRINT
160 FOR J = 1 TO KB(I,0)
170 PRINT "KB";J;" = ";: INPUT K
B(I,J)
180 NEXT J
190 PRINT : INPUT "КОНЦЕНТРАЦИЯ
= ? (МОЛ/ЛИТЪР) ";CB(I,1): INPUT
"ОБЕМ = ? (КУБ.СМ) ";VB(I):SB
= SB + VB(I): PRINT
200 NEXT I
205 IF NA = 0 GOTO 235
210 FOR I = 1 TO NA
220 CA(I,2) = CA(I,1) * VA(I) / (
SA + SB)
230 NEXT I
235 IF NB = 0 GOTO 300
240 FOR I = 1 TO NB
250 CB(I,2) = CB(I,1) * VB(I) / (
8A + SB)
    
```

```

260 NEXT I
270 :
280 REM РЕШЕНИЕ НА УРАВНЕНИЕТО
290 :
300 K = 0.5
310 XA = 1E - 15:XA(0) = XA:X = X
A:B = 0
320 GOSUB 1000
330 FA(0) = F:FA = F
340 IT = 16 / K
350 FOR I = 1 TO IT
360 XB = 1E - 15 * 10 ^ (I * K)
370 X = XB:XA(I) = X
380 GOSUB 1000
390 FA(I) = F:FB = F
400 IF FA * FB > 0 GOTO 470
410 X = (XA + XB) / 2:B = B + 1
420 DETA = XB - X
430 GOSUB 1000
440 X1 = X + DETA / (1 - FB / F)
450 PH = - LOG (X1) / LOG (10)
460 PRINT : PRINT B;" РАЗТВОР :
PH = ";PH
470 XA = XB:FA = FB
480 NEXT I
490 IF B = 0 THEN PRINT : PRINT
"НЯМА РЕШЕНИЕ": PRINT : BOTO
600
500 PRINT : PRINT "ТОВА БЕШЕ ПОС
ЛЕДНОТО НАМЕРЕНО РЕШЕНИЕ"
510 PRINT : PRINT "ЖЕЛАЕТЕ ЛИ ГР
АФИКА? ДА/НЕ ?": INPUT RS
520 IF RS < > "ДА" THEN BOTO 7
10
530 :
540 REM КРИВА НА ФУНКЦИЯТА
550 :
560 HOME : HGR : HCOLOR= 3: BOSUB
800: IF M9 < 0 THEN M9 = 0
570 DX = 279 / 16: IF M0 > 0 THEN
M0 = 0
580 DY = 159 / (M9 - M0):D = M9 *
DY:I1 = I1 - 1
590 HPLOT 0,D TO 279,D: HPLOT 0,
0 TO 0,159
600 HX = (1 - LOG (XA(1)) / LOG
(10)) * DX:HY = (M9 - FA(1))
* DY:IY = I1 + 1
610 FOR I = I1 TO I2
620 IX = (1 - LOG (XA(I)) / LOG
(10)) * DX:IY = (M9 - FA(I))
* DY
630 HPLOT HX,HY TO IX,IY:HX = IX
:HY = IY
640 NEXT I
650 FOR X = 0 TO 14
660 PX = (1 + X) * DX:D1 = D + 3:
DO = D - 3
670 HPLOT PX,D1 TO PX,DO: IF DO <
140 THEN HPLOT PX,154 TO PX
,158
680 IF (X < > 0) AND (X < > 7)
AND (X < > 14) THEN BOTO
700
690 PX = PX + 2: HPLOT PX,D1 TO P
X,DO: IF DO < 140 THEN HPLOT
PX,154 TO PX,158
700 NEXT
    
```



# КИРИЛИЦА

## ЗА ПРАВЕЦ-8Е

```

705 VTAB 21: HTAB 3: PRINT "1";:
HTAB 21: PRINT "7";: HTAB 3
8: PRINT "14": PRINT
710 BET TS: TEXT : END
800 MO = FA(1 / K):M9 = MO
810 I1 = 1 / K + 1:I2 = IT - I1 +
1
820 FOR I = I1 TO I2
830 Y = FA(I): IF MO > Y THEN MO =
Y
840 IF M9 < Y THEN M9 = Y
850 NEXT I: RETURN
1000 :
1010 REM ПРЕСМЯТАНЕ НА F(X)
1020 :
1030 S = 1E - 14 / X: IF NA = 0 GOTO
1130
1040 FOR M = 1 TO NA
1050 D = 1:NU = 1:JT = KA(M,0)
1060 FOR J = JT TO 1 STEP - 1
1070 KM = KA(M,J):J1 = J - 1: IF
J1 = 0 THEN J1 = 1
1080 D = 1 + KM * D / X
1090 NU = 1 + J / J1 * KM / X * N
U
1100 NEXT J
1120 S = S + CA(M,2) * (NU - 1) /
D
1125 NEXT M
1130 XP = X * 1E14: IF NB = 0 GOTO
1230
1140 FOR M = 1 TO NB
1150 D = 1:NU = 1:JT = KB(M,0)
1160 FOR J = JT TO 1 STEP - 1
1170 KM = KB(M,J):J1 = J - 1: IF
J1 = 0 THEN J1 = 1
1180 D = 1 + KM * D * XP
1190 NU = 1 + J / J1 * KM * NU *
XP
1200 NEXT J
1210 S = S - CB(M,2) * (NU - 1) /
D
1220 NEXT M
1230 F = S - X
1240 RETURN

```

СТЪПКА 10 РЕДА.

ОТ	АО	КОНТР.
# РЕД	# РЕД	СУМА
10	- 90	5845
100	- 180	4C0C
190	- 260	4314
270	- 360	292F
370	- 460	3684
470	- 560	4BE9
570	- 660	57D3
670	- 830	512D
840	- 1070	32BE
1080	- 1170	3A3C
1180	- 1240	1EF1
КОНТР.	СУМА	E C4EC

Персоналният компютър Правец-8Е е комплектуван фабрично със знаков генератор, в който не са включени буквите от кирилицата. Ето защо компютърът може да работи с кирилица само в режим на графика с висока разделятелна способност, като се използа например програмата HIPRINT.

Знаковият генератор на Правец-8Е е записан на програмируема постоянна памет (епром) от тип 2764 с организация 8 Кбита  $\times$  8. Освен символите, въвежданите от клавиатурата, той съдържа и кодирана информация, необходима за генериране на изображение в режими графика с висока и графика с ниска разделятелна способност.

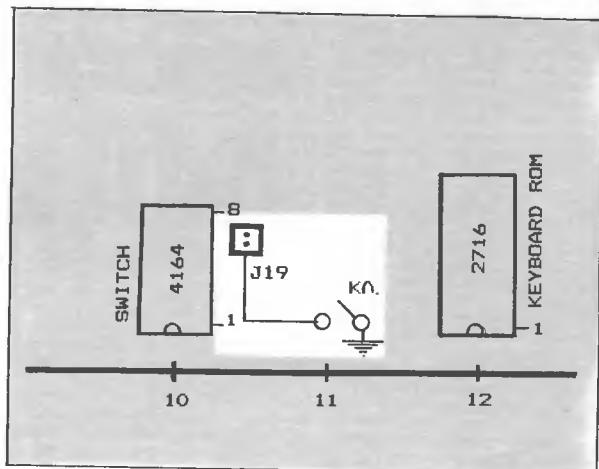
Анализът на записа в епром показва, че той съдържа два независими един от друг знакови генератори с еднакво съдържание. Затова има възможност така да се прекодира едната половина на епрома, че

в нея да се запищат графичните изображения на символите от кирилицата и двата знакови генератора да се преключват.

Това решение запазва възможностите на компютъра, а допълнително предлага съвместимост с програмите за Правец-82. За целта е необходимо да се промени съдържанието на определени области от епром (виж таблица 2), като в едната половина на знаковия генератор се записват графичните изображения на символите от таблица 1.

Двата знакови генератора се преключват с микропреключател, свързан към двете контактни плошкиадки, означени с надписа SWITCH J19 и наридащи се на позиция F11 на основната плата на компютъра (фиг. 1). Микропреключателят се монтира на удобно място на кутията или на дъното на компютъра.

**Инж. ПЕТЪР ПЕТРОВ**  
**ИТКР — БАН**  
**направление „Персонални**  
**компютри и системи“**



Фиг. 1

ТАБЛИЦА 1

ИНВЕРСНО				НОРМАЛНО				ИНВ.				НОРМАЛНО			
0	@	P	0	.	r	ю	п	ю	п	!	0	@	Р	ю	п
1	A	Q	1	,	q	в	р	я	я	!	1	А	Q	я	я
2	B	R	2	"	g	с	сп	б	р	"	2	В	R	с	р
3	C	S	#	3	s	т	м	ц	с	#	3	С	т	т	с
4	D	T	\$	4	d	е	у	а	д	\$	4	Д	т	е	т
5	E	U	%	5	e	т	ж	а	а	%	5	Е	у	т	ж
6	F	V	&	6	f	в	в	е	е	&	6	Ф	в	в	в
7	G	W	'	7	g	х	ж	и	и	'	7	Г	х	ж	и
8	H	X	(	8	h	х	ь	и	и	(	8	Х	х	ь	и
9	I	Y	)	9	i	у	з	и	и	)	9	И	у	з	и
A	J	Z	*	:	j	з	е	е	е	*	:	Я	з	е	е
B	K	C	+	:	k	к	л	и	и	+	:	Б	к	л	и
C	L	\	<	:	l	л	и	и	и	<	:	С	л	и	и
D	M	]	=	:	m	и	и	и	и	=	:	Д	и	и	и
E	N	^	>	:	n	о	и	и	и	>	:	Е	о	и	и
F	O	/	?	:	o	и	и	и	и	?	:	Ф	о	и	и

ТАБЛИЦА 2

200	-	FB	F7	E9	FF	E3	DF	C3	DD	C3	FF							
210	-	FD	FD	E1	DD	DD	DD	E1	FF	FF	FF	C3	FD	FD	C3	FF		
220	-	DF	CD	C3	DD	DD	DD	C3	FF	FF	FF	FB	E1	FB	FB	FF		
230	-	E7	DB	FB	E1	FB	FB	FB	FF	FF	FF	E3	DD	DD	C3	DF	E3	
240	-	FD	FD	E1	DD	DD	DD	DD	FF	F7	FF	F3	F7	F7	F7	E3	FF	
250	-	EF	FF	E7	EF	EF	ED	ED	F3	FD	FD	DD	ED	F1	ED	DD	FF	
260	-	F3	F7	F7	F7	F7	F7	E3	FF	FF	FF	E9	D5	D5	D5	D5	FF	
270	-	FF	FF	E1	DD	DD	DD	DD	FF	FF	FF	E3	DD	DD	E3	FF		
280	-	FF	FF	E1	DD	DD	E1	FD	FD	FF	FF	C3	DD	DD	C3	DF	DF	
290	-	FF	FF	C5	F9	FD	FD	FD	FF	FF	FF	C3	FD	E3	FD	E1	FF	
2A0	-	FB	FB	E1	FB	FB	DB	E7	FF	FF	FF	DD	DD	DD	CD	D3	FF	
2B0	-	FF	FF	DD	DD	EB	EB	F7	FF	FF	FF	'D	DD	DD	D5	C9	FF	
2C0	-	FF	FF	DD	EB	F7	EB	DD	FF	FF	FF	ED	ED	ED	E3	F7	F9	
2D0	-	FF	FF	C1	EF	F7	FB	C1	FF	E7	FB	FB	F9	FB	FB	E7	FF	
2E0	-	FB	F3	EF	EF	CF	EF	EF	F3	FF								
2F0	-	D3	E5	FF	FF	FF	FF	FF	FF	C0	DE	DE	DE	DE	DE	CO	FF	
300	-	FF	FF	ED	D5	D1	DS	ED	FF	FF	FF	F1	EF	E1	EE	D1	FF	
310	-	E3	FD	E1	DD	DD	DD	E3	FF	FF	FF	ED	ED	ED	ED	C1	DF	
320	-	FF	FF	C3	D7	D7	DB	C1	DD	FF	FF	E3	DD	C1	FD	E3	FF	
330	-	FF	F7	E3	D5	D5	DS	E3	F7	FF	FF	E1	FD	FD	FD	FF		
340	-	FF	FF	DD	EB	F7	EB	DD	FF	FF	FF	DD	CD	D5	DD	FF		
350	-	FF	F7	DD	CD	D5	D9	DD	FF	FF	FF	ED	F5	F9	F5	ED	FF	
360	-	FF	FF	E7	DB	DB	DB	D9	FF	FF	FF	C9	D5	D5	DD	DD	FF	
370	-	FF	FF	DD	DD	C1	DD	DD	FF	FF	FF	E3	DD	DD	D3	FF		
380	-	FF	FF	C1	DD	DD	DD	DD	FF	FF	FF	C3	DD	C3	D7	D9	FF	
390	-	FF	FF	E1	DD	DD	E1	FD	FD	FF	FF	C3	FD	FD	C3	FF		
3A0	-	FF	FF	C1	F7	F7	F7	F7	FF	FF	FF	ED	ED	ED	E3	F7	F9	
3B0	-	FF	FF	D5	D5	E3	D5	D5	FF	FF	FF	E1	DD	E1	DD	E1	FF	
3C0	-	FF	FF	FD	FD	E1	DD	E1	FF	FF	FF	FC	FD	E1	DD	E1	FF	
3D0	-	FF	FF	E3	DF	E7	DF	E3	FF	FF	FF	D5	D5	D5	C1	FF		
3E0	-	FF	FF	E3	DF	C7	DF	E3	FF	FF	FF	EA	EA	EA	EA	CO	DF	
3F0	-	FF	FF	DD	DD	C3	DF	DF	FF	FF	FF	EA	F5	EA	F5	EA	F5	
400	-	32	4A	4A	4E	4A	4A	32	00	OB	14	22	22	3E	22	22	00	
410	-	1E	02	02	1E	22	22	21	00	22	22	22	22	22	22	7E	40	
420	-	1B	24	24	24	24	24	24	TE	42	3E	02	02	1E	02	02	3E	00
430	-	08	3E	49	49	49	3E	OB	00	3E	02	02	02	02	02	02	00	
440	-	22	22	14	OB	14	22	22	00	22	22	32	2A	26	22	22	00	
450	-	2A	22	32	2A	26	22	22	00	22	12	0A	06	0A	12	22	00	
460	-	38	24	24	24	24	24	24	00	22	36	2A	2A	22	22	22	00	
470	-	22	22	22	3E	22	22	22	00	1C	22	22	22	22	22	1C	00	
480	-	3E	22	22	22	22	22	22	00	3C	22	22	3C	2B	24	22	00	
490	-	1E	22	22	1E	02	02	02	00	1C	22	02	02	02	22	1C	00	
4A0	-	3E	OB	OB	OB	OB	OB	OB	00	22	22	22	3C	20	20	1E	00	
4B0	-	2A	2A	1C	1C	2A	2A	2A	00	1E	22	22	1E	22	22	1E	00	
4C0	-	02	02	02	1E	22	22	21	00	03	02	02	1E	22	22	1E	00	
4D0	-	1E	20	20	1C	20	20	1E	00	2A	2A	2A	2A	2A	2A	3E	00	
4E0	-	1C	22	20	3C	20	22	1C	00	2A	2A	2A	2A	2A	2A	7E	40	
700	-	CD	B5	B5	B1	B5	B5	CD	FF	F7	EB	DD	DD	C1	DD	DD	FF	
710	-	E1	FD	FD	E1	DD	DD	E1	FF	DD	DD	DD	DD	DD	B1	BF		
720	-	E7	DB	DB	DB	DB	DB	B1	BD	C1	FD	FD	E1	FD	FD	C1	FF	
730	-	F7	C1	B6	B6	B6	C1	F7	FF	C1	FD	FD	FD	FD	FD	FD	FF	
740	-	DD	DD	EB	F7	E9	DD	DD	FF	DD	DD	CD	D5	D9	DD	DD	FF	
750	-	D5	DD	CD	D5	D9	DD	DD	FF	DD	ED	F5	F9	F5	ED	DD	FF	
760	-	C7	DB	DB	DB	DB	DB	DD	FF	DD	C9	D5	D5	DD	DD	PF		
770	-	DD	DD	DD	C1	DD	DD	DD	FF	E3	DD	DD	DD	DD	E3	FF		
780	-	C1	DD	DD	DD	DD	DD	DD	FF	C3	DD	DD	C3	D7	DB	DD	FF	
790	-	E1	DD	DD	E1	FD	FD	FD	FF	E3	DD	FD	FD	DD	E3	FF		
7A0	-	C1	F7	F7	F7	F7	F7	F7	FF	DD	DD	D3	DF	DF	E1	FF		
7B0	-	DS	D5	E3	E3	D5	D5	DS	FF	E1	DD	DD	E1	DD	DD	E1	FF	
7C0	-	FD	FD	FD	E1	DD	DD	E1	FF	FC	FD	FD	E1	DD	DD	E1	FF	
7D0	-	E1	DF	DF	E3	DF	DF	E1	FF	D5	DS	DS	DS	DS	C1	FF		
7E0	-	E3	DD	DD	C3	DF	DD	E3	FF	D5	DS	DS	DS	DS	B1	BF		
7F0	-	DD	DD	DD	C3	DF	DF	DF	FF	FF	D5	EB	DS	EB	DS	FF	FF	

# СЕКРЕТНОТО ОРЪЖИЕ

## НА ЕПЪЛ

Инж. ГЕОРГИ МИРЧЕВ

От средата на миналата година пътна слух, че през 1986 фирмата „Епъл“ се готви да нанесе своя голям удар и да укрепи, ако не и да възпроизведе позициите си на един от водещите производители на персонални компютри в света. Известно е, че основалите фирмата дамски младежи не само успяха да създадат под носа на гиганта в електрониката IBM първия персонален компютър, но и да отстоят правото на съществуване на „ябълките“. С времето ищата постепенно дойдоха на места си и сега приближните сътънотошение и силите на световния пазар за персонални компютри е около 41% за IBM спряму 11% за „Епъл“. Жизнеността на „Епъл“ се дължи предимно на много програмни продукти, създадени за нейните компютри — само официално регистрираните са над 20 000.

Но това очевидно не е достатъчно и затова фирмата усилено и в най-строга тайна подготвя своя хардверен удар с новите Макинтош® (в по-далечно бъдеще и Джонатан, отново сорт ябълки) и особено Епъл IIХ.

Макар и името му засега да съдържа тайнственото „Х“, в печата проникнаха някои подробности, по които ще се опитаме да реконструираме образа му.

Епъл IIХ е продължител и очевидно наследник на семейството Епъл II, но за разлика от него е вече 16-битов. Същевременно той ще може да работи на 100 процента съвместимост с всички програмни продукти за 8-битови си събрата. Така той ще обедини предимствата на новото, по-модерно схемно решение с вече съществуващото огромно софтуерно богатство.

устройство е съвместимо с 3,5-инчовите дисковетни устройства, с които фирмата вече близо година снабдява своите компютри.

Компютърът запазва отворена архитектура на фамилията II, която позволява към него да се включват множество допълнителни карти за разширяване на възможностите му. Предполага се, че ще могат да се използват и повечето от сега съществуващи карти.

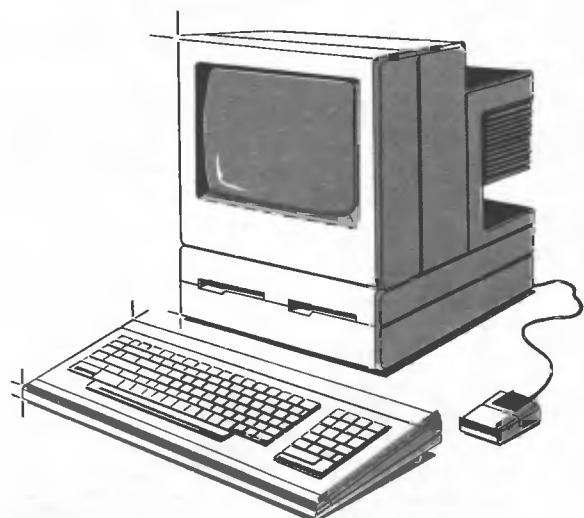
По подобие на Епъл Лайза и Макинтош той ще може да работи и с „мишка“.

Новият компютър скъсва и с постарелия дизайн на предците си — ще има ергономична подвижна клавиатура с регулируем наклон и вероятно с отделен блок цифрови клавиши, по новому ще бъде оформена кутията на компютъра и цветният видеомонитор.

В противовес на всички положителни страни на Епъл IIХ ще обобщим мнението на мнозина специалисти, които добавят:

— Колко жалко, че няма да бъде съвместим с IBM PC!

Нецо, което на Запад малко фирми се осмеляват да си разрешат.



## ПРИНТЕРЪТ РОБОТРОН

Заводите „Роботрон боромашиненверк“ (ГДР) започнаха производство на матричен принтер „Роботрон 6311“. Основни характеристики:

- Скорост на печтане — 100 символа в секунда
- Формат на растера на символите — 9x7
- В рам са записани два латински шрифта — стандарт и курсив, а в рам могат да се зареждат различни други национални шрифтове или професионални символи. Всички шрифтове могат да бъдат печтани в три формата на ширини — по 10, 12,5 и 15 знака на инч, като всеки от тях може да се удължи на ширината по желание (за заглавии например).

Принтерът се произвежда в следните варианти: с гумен валник и със зъбно подаване на хартията. Интерфейсна плата — Центроникс, паралелна (PC 232 C). Габаритни размери — 390 x 300 x 150 mm. Маса — около 10 кг. Консумация — 70 вата.



## СИСТЕМАТА „БЪТЕРФЛАЙ“

Английски специалисти са разработили компютър, който обединява 128 процесора и изпълнява 60 милиона операции в секунда. Той представлява паралелен процесор, мултипроцесор, мултикомпютър и същевременно нито едно от тях. Създателите му пръвначално са го характеризирали като паралелна микропроцесорна система с тясно взаимодействие, понеже всички процесори са обединяват от комутационна система, чиито възможности нарастват с увеличаване броя на процесорите.

Комутаторите са съръголемни интегрални схеми. Увеличението на процесорите води до линейно нарастване на производителността — всеки процесор я увеличава с 0,5 милион операции в секунда. Процесорите разполагат както със собствена памет, така и с общата памет на системата. Тенденцията в тази област е да се повишава броят на процесорите — сега са работи по системи с 16 000 процесора, като крайната цел е компютър, изграден от 65 000 процесоро-

## СИНХРОТРОН ЗА МИКРОЧИПОВЕ

Японски изследователи работят много настolioчко за прилагане на рентгенови лъчи при производството на интегрални схеми с изключено голяма степен на интеграция. С тази нова технология ще получат следователно крачка в миниатюризирането.

Сегашните най-компактни чипове са с капацитет 256 Кбайт са изградени от интегрални схеми, чиито проводящи ивици са широки по 2 микрона. Рентгеновото лъчение ще позволяи гази ширина да се намали до 0,5 микрона. В момента усилията на учениците са насочени към създаването на компактен синхротронен източник на рентгеновите лъчи, който ще има размери не повече от 10 метра в диаметър — същинско джудже в сравнение със съвременните ускорители на елементарни частици.

В проекта са инвестиирани няколко десетки милиона долара, а първите резултати се очакват след 4—5 години. Тази информация дава база за пренеща, кога могат да се очакват чипове с капацитет над един Мбайт.

## ГРАМОФОН ЗА КОМПУТРИ

Колкото и същършени да стават компютрите и запомнящите устройства, програмистите са все недоволни и настояват за повече и по-добра памет. Но и най-придирчивите изориги са омелели пред магнитното CDR-15025 (или четвъртата корница), която „Хитачи“ куся на пазара в края на мината година. И е имало зашо. През разメリ колкото две обикновени дискетни устройства, то може да съхранява 552 Мбайт информация, която чете със скорост 176 Кбайт/сек!

55 Униквестер-диски по 10 Мбайт, всичките поместени в една кутийка — това постижение наистина дава основание да се говори за ново направление в развитието на изящните запомнящи устройства. Образно казало, върху 12-санитристов дисък се побира информация, която, изписана на машинописни страници, би образувала стълб с височина 12 метра.

Към кредитите на устройството ще добавим, че за разлика от Униквестер-дискетите тук не е необходимо поддържането на специална газова среда, а дискът носи сълт на информацията, с изключително по-очувателен външен въздействия в сравнение с търдите дискове или гълъбите диски.

Но тъй като идеалното обикновено и с недостатъци, и повсяк супердиск си има недостатъци. За най-голямо от тях се смята фактът, че с него може само да се чете предварително записаната при фабрични условия информация. Тя може да се полузапомни, комбинира, промени, но не може да се записва обратно върху диска. За тази цел пак трябва да се прибегне до други традиционни устройства, които записват информацията върху диски с малгътко покритие.

Дискът, подобно на дисковете, може свободно да се сменя, като единственото изискване е да се използва от механично увреждане. За него не е страшно дори и да се ползва с кафе например, запод може да се измие с вода и сапу, без това да влоши качеството му.

Включването на дисковото устройство като стандартни компютри не представлява проблем и то се продава заедно с контролер за IBM PC и съвместим с него. Проблемите (засега) идват от дисковата операционна система, която не е пригодена да обслужва толкова големи обем памет, но това вече е проблем, с който програмистите скоро ще се справят.

## ЖЕСТОВЕ УПРАВЛЯВАТ КОМПЮТРИ

Най-често данните за управление и обработка с компютри се въвеждат от оператори чрез клавиатура. А специалистите от фирмата „Бел телефон“ предлагат за тази цел да се използват еластични ръкавици, в които са вградени датчици. Те са разположени на дланиите, на китките, по фалангите и по връховете на пръстите. Чрез тях различните жестове се преобразуват в комбинации от електрически сигнали. Като усвои езика на жестовете, операторът ще може да въвежда данните в паметта на компютъра приблизително по същия начин, както по втора програма на Българската телевизия се водят предавания за граждани с увреден слух.



## ЕКЗОТИЧНИ КОМПЮТРИ

В битката за оцеляване всеки се спасява кой както може. Докато един увеличават Мбайтовете на паметта на компютрите си, други разчитат на по-емоционални фактори. Една фирма например се стреми да съставя новата модерна техника с тежки стариен интерпрет на директорските кабинети, като поставя компютрите в кутии от махагон, палисадър и други екзотични дървесни видове. Оказало се, че броят на съдиментилите към миналото клиенти не бил малък.

## ПОРТАТИВНИТЕ „КРАЙДКЕЙЗ“

Семейството портативни компютри „Крайдкейз“ са с формени конструктивно в куфарчета от типа „дипломат“ и имат три вида екрани — на течни кристали, усъвършенстван в техни кристали и кърти плазмен. Масата им е около 5,5 килограма и могат да се захранват от батерии и от мрежата. Те имат вградени 82-милиметрови зампънища устройства на гънка и магнитен диск с капацитет 720 Кбайта. Към тях лесно могат да се възложат принтери, плодери и запомнящи устройства, както 133-милиметрови, така и 10-Мбайтови твърди дискове. Компютрите могат да влизат в локални глобални мрежи чрез специално устройство, наречено „Крайд скълер“, и са напълно съвместими с персоналните компютри PC, XT и AT на IBM. Екраните им имат формат 25 реда по 80 знака и диагонал 23 сантиметра с разделителна способност 640 на 200 точки.

## С ЕДНО ДОКОСВАНЕ

От тази година автомобилите „Бунк“ ще бъдат снабдявани с борден микрокомпютър. Той ще изпълнява повече от 100 контролни функции, създадени с управлението на автомобила. Тъй като шофьорът ги ползува предимно в движение, управлението на информационната система е извънредно опростено. Различните функции се избират по графичного меню само с едно докосване до екрана. Никоя от по-интересните възможности на компютъра (обем на паметта 8 Кбайта):

- управлява климатична инсталация;
- прави пеперудната диагностика на двигателя;
- прогнозира часъ на пристигане. Шофьорът извежда разстоянието, което му предстои да измине, а компютът изчислява през всеки 10–15 км средната скорост и чрез нея — часа на пристигане;
- от момента на запалване на двигателя компютърът следи моментния резход на гориво;
- управлява радиокасетофона.

Самата система се включва автоматично и се подготви за работа в момента, когато предиата линия връзка отвор, а екранът светва, когато шофьорът седне на мястото си и затвори вратата.

FLASH  
FLASH  
FLASH

● МЕГАПРОЕКЪТ. „Филипс“ и „Сименс“ съществат усилено съз за разработването на чип памет с капацитет един Мбайт, а в близко бъдеще и 4 Мбайта. Западноевропейското и холандското правителство ще поемат 40% от стойността на исследователската работа, която по предварителни прещеници ще излезе на 550 млн. долара. Едни политически сици: двете конкуриращи се фирми си подават ръка — просто написах на американските и японските концерни ги иригуждат да правят избор между да работят сътрудничество.

● ЛАЗЕРЕН ПРИИНТ. На големия професионален салон за компютърна техника „Комдекс“ ба представен лазерният принтер „Kics“, който струва с една трета по-евтино, отколкото останалите (2000 вместо 3000 долара). Неговият капацитет е напълно приемлив — шест страница, формат A4 за минута.

● ИНФОРМАТИКА 2005. Информатиката е единственият научноизследователски сектор, в който се очакват коренни промени през следващите двадесет години. Това са основните изводи, направени по време на колоквиума „Перспективи‘2005“, състоял се в края на миналата година в Париж. Пречистват се, че два фактора ще тласкат напред това развитие. Първият ще бъде появата на очакваните компютри от пето поколение, надарени с изкуствен интелект, вторият — създаването на микрочипове с ултратеска степен на интеграция. В количествени измерения това очаква, че до края на столетията степента на интеграция на електронните елементи ще бъде доведена до физическата им граница, която очаква, че отношението на интеграция към цена ще бъде повищено около 10 000 пъти. Една тогава може да се очаква, че шафетата на по-нататъшното развитие ще бъде поета от биочиповете.

# Съдържание

## ПОПРАВКА

В брой 4 в статията "Асемблер и машинен език" поради технически грехи са размени места на някои от вътрешните подзаглавия. Вирката последователност на всички подзаглавия е следната:

1. Адресиране с подразбиране на акумулатора.
  2. Адресиране с подразбиране на регистър.
  3. Непосредствено адресиране.
  4. При ко пълно адресиране.
  5. При ко адресиране в нулевата страница.
  6. При ко адресиране в нулевата страница с индексиране по X.
  7. При ко адресиране в нулевата страница с индексиране по Y.
  8. При ко пълно адресиране с индексиране по X.
  9. При ко пълно адресиране с индексиране по Y.
  10. Косвено пълно адресиране с предварително индексиране по X.
  11. Косвено пълно адресиране с посекдашко индексиране по Y.
  12. Косвено пълно адресиране.
  13. Относително адресиране спрямо РС.
- В статията "Повече за оператора IF" са разместили места на примерите за просто и съставно условие.
- Редакцията се извинява за допуснатите грехи.

ИЗОБРЕТЕНО В СОФИЯ*86 . . . . .	2
КОМПЮТЪРЪТ И/ИЛИ ПЕДАГОГЪТ . . . . .	5
ПОД ЗНАКА НА ТРИТЕ „М“ . . . . .	7
КАПИТАНЕ, ДА ВДИГНЕМ ЛИ ФЛАГА? . . . . .	8
ПРОГРАМИРАНЕ НА ПРОЛОГ . . . . .	10
НЕПОСРЕДСТВЕН ДОСТЪП ДО ПАМЕТТА . . . . .	12
МАШИНЕН ЕЗИК И АСЕМБЛЕР . . . . .	15
УПРАВЛЯВАЩА ПРОГРАМА МОНИТОР . . . . .	17
МОЖЕМ ЛИ ДА СИ НАПРАВИМ ДЖОИСТИК? . . . . .	18
ПЪТЕВОДИТЕЛ ИЗ ПРАВЕЦ-82 . . . . .	20
КОНТРОЛНА СУМА . . . . .	21
ПЛОЩ НА ПОЛИГОН . . . . .	23
РН-МЕТЪР . . . . .	24
КИРИЛИЦА ЗА ПРАВЕЦ-8Е . . . . .	26
СЕКРЕТНОТО ОРЪЖИЕ НА „ЕПЪЛ“ . . . . .	29
ПАНОРАМА . . . . .	30

## КОМПЮТЪР ЗА ВАС

Издание на ЦК на ДКМС

### СПИСВА РЕДАКЦИЯ „ОРБИТА“

Главен редактор  
в-р ДИМИТЪР ПЕЕВ 88-51-68



1000 София СОФИЯ  
БУЛ. „ТОЛБУХИН“ № 51 А  
ТЕЛ. 87-78-04

Приемни часове от 14 до 16 ч.

НЕПУБЛИКУВАНИ РЪКОПИСИ И ПРОГРАМИ НЕ СЕ ВРЪЦАТ.

РЕДАКЦИОНЕН СЪВЕТ: чл.-кор. Ангел Ангелов, проф. Ангел Писарев, ст.н.с. к.т.н. инж. Александър Александров, асистент Благой Сенцов, Веселин Стириданов, док. Димитър Шишков, инж. Иван Маранчов, инж. Петко Сураков, чл. кор. Петър Констанров, н.с. к.т.н. инж. Пламен Вачков, Рафаил Ангеловин инж. Иван Михайлов инж. Петър Петров

ЗАМ.-ГЛАВЕН РЕДАКТОР  
И ЗАВ. СПИСАНИЕТО  
инж. Георги Балански 87-09-14

ОТГОВОРЕН СЕКРЕТАР  
инж. Борис Айков 80-23-18

ДЕЖУРЕН РЕДАКТОР  
Анелия Ерменкова

ДИЗАЙНЕР  
Васил Пенев

ТЕХНИЧЕСКИ РЕДАКТОР  
Люба Каплакчиева

КОРЕКТОР  
Бисела Ботева

Предадено за печат  
16 април 1986 г.

Подписано за печат  
3 юли 1986 г.

Печатни коли 4

Формат 60/90/В

Тираж 20 000

Цена 0,60 лв.

Годишен абонамент 7,20 лв.

\*\*\*\*\*

ДП "Д. Благоев"

София, ул. "Ракитин" 2

Телефон 46-31

\*\*\*\*\*

Индекс 20 593

(4E-4F)	(78-79)	RNDL RNDH
*(50-51)	(81-82)	ACL ACH
*(52-53)	(83-84)	XTNDL XTNDH
*(54-55)	(85-86)	AUXL AUXH

Двубайтово псевдослучайно число, генерирано при всяко въвеждане от клавиатура чрез подпрограмата KEYIN - (RANDOM NUMBER; L - младши байт, H - старши байт).

Използват се като 16-битов псевдоакумулатор в подпрограмите за умножение и деление на двубайтови числа (L - младши байт, H - старши байт).

Използват се като разширение на акумулатора до 32 бита подпрограмите за умножение и деление на двубайтови числа (EXTENSION; L - младши байт, H - старши байт).

Използват се като допълнителен 16-битов регистър в подпрограмите за умножение и деление на двубайтови числа (AUXILIARY REGISTER; L - младши байт, H - старши байт).

\* Подпрограмите за умножение и деление на двубайтови числа не включени в Монитора на ПРАВЕЦ-82. Те ще бъдат разгледани самостоятелно като често се използват в програми на Асемблер и машинен език.

### ПОДПРОГРАМИ НА МОНИТОРА

НАЧАЛНИ АДРЕСИ	ИМЕ	ОПИСАНИЕ
ШЕСТНАЙ- СЕТИЧЕН	ДЕСЕТИ- ЧЕН	
F800	-2048 PLOT	Извежда точка в графичен режим с ниска разделителна способност. Преди извикването координатите се зареждат - X в регистър Y и Y в акумулатора. След изпълнението съдържанието на акумулатора се променя.
F819	-2023 HLINE	Извежда хоризонтална линия в графичен режим с ниска разделителна способност. Преди извикването координатите се зареждат - X1 в регистър Y, X2 в клетка с адрес \$2C и Y в акумулатора. След изпълнението съдържанието на акумулатора и регистър Y се променя.

# ГРАМОФОН ЗА КОМПУТРИ

А. Чан.



CDR-1502S работи на принципа на лазерните грамофони — фино фокусиран лазерен лъч оципва спираловидно очертаниите върху диска бразди, по които информацията е записана чрез микроскопични прогорени дупчици. Те отразяват лъча различно от гладките места и това се улавя от оптичен сензор, който подава информацията на електронния блок за разшифроване.